

# Getting started

Alchemix Finance is a future-yield-backed synthetic asset protocol and community DAO. The protocol gives you advances on various yield farming strategies via a synthetic token. The token represents a fungible claim on the underlying collateral in the Alchemix protocol, where the claim has to be made by a depositor of that collateral.

The DAO will focus on funding projects that will help the Alchemix ecosystem grow, as well as the greater Ethereum community.

## What Does Alchemix Do?

Imagine a bank. You deposit money, and the bank pays you interest on your deposit. There's a credit card attached to the account. The card allows you to spend up to 50% of the asset you have deposited. In order to access this debt, you sacrifice a small percentage of its value upfront. There's no interest on the debt. There are no monthly payments to make. Your debt is denominated in the asset you deposit, so there are no liquidations. Instead, the interest you earn on your total initial deposit pays off any debt you have, automatically. Alchemix offers this product, in DeFi form: Alchemix is Self-Paying, Interest-Free, Non-Liquidating Loans.

For more detailed information and to learn how it works see [Components](#) and [Guides and Explainers](#)

## How do I use it?

Alchemix currently offers aUSD to borrow against select USD-pegged stablecoins, and aETH to borrow against ETH.

1. **Deposit:** Select a yield strategy and then add collateral (Stablecoins or ETH) to that strategy.
2. **Borrow:** Borrow up to the acceptable collateralization ratio and receive a synthetic aAsset representing future yield equivalent to the borrowed amount.
3. **Convert:** Swap the aAsset to any other token via a DEX or DEX Aggregator (check the current aAsset price before doing so). aAssets can also be used directly on some DeFi protocols.
4. **Spend:** Do anything with the loan (buy more crypto, take a vacation, cash out to savings, and many more options!)
5. **Wait:** The user's chosen yield strategy will go to work earning interest on the full initial deposit - the harvested yield will pay down their debt automatically over time.
6. **Withdraw, Borrow, Repay, or Self-Liquidate:** At any time, users can withdraw principal or borrow more, up to the collateralization ratio, or self-liquidate to gain their collateral minus any outstanding debt. Users may also repay their loans if they wish (with the respective aAsset or underlying token). Alchemix always views 1 aAsset as equivalent to 1 underlying asset (i.e., 1 aUSD = 1 DAI) for the purpose of borrowing, self-liquidating, and repaying loans.

For more detailed instructions see [How to](#)

# Components

**Alchemist** - Alchemists act as the smart contract hub to generate yield and yield advances. They share many similarities to collateralized lending platforms, such as MakerDAO and AAVE. The currently accepted collateral types are ETH, DAI, USDC, USDT, and FRAX, as well as their respective yield-bearing tokens for each strategy. Users take loans in the form of synthetic aAssets: aETH can be borrowed against ETH-denominated yield token deposits, and aUSD can be borrowed against DAI, USDC, USDT, and FRAX-denominated yield token deposits. Alchemix takes 10% of all generated yield as a service fee. This service fee is sent to the DAO treasury.

**Transmuter** - Alchemix incentivizes liquidity pools for the synthetic aAssets so that users can swap their synthetic assets for the underlying asset immediately upon receipt. If the liquidity pool exchange rate value of the synthetic asset is significantly below the value of the underlying, the Transmuter is a price stability module that allows users to stake their synthetic aAssets to have them be converted into their corresponding base asset on a 1:1 basis over time. Whenever an Alchemist performs a harvest, or whenever a user repays their debt with a base asset or self-liquidates, those tokens are sent to a Transmuter buffer contract. Those tokens are then gradually released to be redeemed by their corresponding Transmuter according to a time-based formula. For this reason, the Transmuter is considered a backstop to the redeemability of synthetic aAssets, and not the preferred route for users to swap their synthetic aAssets. Liquidity for instant swaps of aAssets can be found on Curve Finance, as well as other decentralized exchanges / AMMs.

**aAssets** - aAssets are the synthetic debt assets that represent a user's future yield. Users must be able to sell aAssets in order to be able to utilize them for on-chain and real-world purchases. Users may wish to buy aAssets to repay a loan early, or to convert to the underlying over time via the Transmuter. For these reasons, aAssets are typically expected to have a market price at some discount from the underlying (e.g., 1 aETH will typically be worth less than 1 ETH). This discount can be viewed as the cost for depositors to access future yield early. The Transmuter, Elixir AMO, and DAO Treasury all work to minimize the discount for each aAsset through governance defined frameworks.

**Elixir AMO** -The Transmuter has a set amount of aAssets that are allowed to be converted to the underlying collateral. This creates the potential for a surplus of assets in the Transmuter. When these surpluses occur, they are used to provide liquidity for their corresponding aAssets. This is done through the DAO-controlled Elixir AMO. The Elixir deposits to the liquidity pools increase the price of the aAsset and earn yield. This liquidity can be withdrawn single-sided as aAssets in order to mimic the effect of the Transmuter.

**Alchemix DAO** - The governance of Alchemix Finance is managed by the DAO with the liquid governance token, ALCX.

**ALCX Token** - The Alchemix governance token (ALCX) initial supply and emissions are split among the developers, the DAO itself, and the community.

# Alchemist

:

Alchemists are the smart contract center of the Alchemix protocol. The AlchemistV2.sol contract powers the Alchemist. Alchemists accept yield-bearing assets as collateral (and can accept underlying collateral as deposits as well, which are converted to yield tokens). An Alchemist may contain multiple yield strategies, but may only issue one alAsset (i.e., the alUSD alchemist issues alUSD, but has yield strategies and accepts yield bearing tokens for USDC, USDT, and DAI). Below is an example of how it works with the stablecoin DAI and the Alchemix synthetic stablecoin, alUSD, using the yearn.finance DAI yVault vault. The system works the same for other accepted stablecoins, as well as ETH (and associated yield tokens) and alETH:

1. User deposits yvDAI into the Alchemist (if user deposits DAI, the Alchemist will first deposit the DAI in Yearn to obtain yvDAI, then accept the deposit.)
2. User borrows alUSD worth up to 50% of the amount of deposited collateral. Loans will have an absolute minimum 200% collateralization ratio. This means for every 2 DAI a user deposits, the user may borrow up to 1 alUSD.
3. The yield that accrues to the yvDAI is periodically harvested to repay the debts of depositors. With each harvest, the user's account will be credited a proportional share of the harvest, thus reducing user's debt. If you have deposited DAI, but have not borrowed alUSD, the harvest will increase your alUSD borrow limit.
4. As yield is harvested, it is transferred to the [Transmuter](#) contract.
5. As the protocol pays down the user's debt, the user can withdraw increasing amounts of DAI from the Alchemist, or the user can re-up their alUSD loan, while always maintaining a minimum 200% collateralization ratio.
6. At any time the user can repay any portion or all of their debt in order to unlock their collateral. DAI, USDC, and USDT are alUSD are treated as 1:1 by the Alchemist for repayment and self-liquidation. As such, alUSD debt can be repaid with alUSD and/or DAI, USDC, and USDT. Repaying debt with alUSD is also a price-restoring mechanism, because when alUSD price is under \$1, users can buy it from AMMs and pay off their debt at a discount.
7. At any time, you can liquidate a portion or all of your collateral. The contract will repay your alUSD debt using the DAI from your yvDAI collateral.

Essentially, the Alchemists give users a flexible line of credit for their future yield. Users can enter and exit anytime without committing to long lockups. There will never be a liquidation of a user's collateral unless they do it themselves because their debt will only ever go down.

Alchemix value the safety of users' deposits more highly than anything else. Alchemix has undergone multiple [audits](#) and carries ongoing bug bounties. While Alchemix does undergo security reviews and audits, there is no guarantee that something bad will not happen. Two additional security measures exist to protect the Alchemical synthetic tokens.

Firstly, there is a limit on how much of each accepted underlying collateral can be deposited into Alchemix to mint alAssets. This limit depends on the level of technical, market, and legal risk for a given asset. Secondly, the Alchemist limits the damage that can be done through the `Maximum Loss`, `Repay Cap`, and `Liquidate Cap` parameters.

The `maximumLoss` looks at the price of a yield token in units of the underlying token. If that falls below the specified `maximumLoss` (typically between 1 and 25 bps), then various functions associated with the yield token are shut down - specifically `liquidate`, `deposit`, `depositUnderlying`, `withdrawUnderlying`, `withdrawUnderlyingFrom` and `harvest`. Note that in this scenario, the user is still able to withdraw the yield token - just not the underlying collateral. For example, the user can withdraw yvDAI, but not DAI.

The `Repay Cap` and `Liquidate Cap` are in place to prevent significant arbitrage movements if an underlying collateral asset were to depeg. This gives sentinels time to react to market events. The limits are time-based:

- **Repay Cap** - The quantity of debt that can be repaid over a set duration of time. It could specify for example that "10k ETH can be repaid every 10 minutes."
- **Liquidate Cap** - The quantity of an asset that can be liquidated over a set duration of time. It could specify for example that "10k ETH can be liquidated every 10 minutes."

Repay and Liquidate caps are currently only set for stablecoin collateral, as they are designed to be stably pegged to USD. They are not required for ETH, as ETH is a standalone asset with a floating price.

These security measures will not completely prevent the damage from a stablecoin losing its peg or becoming devalued, but they will protect much of the Alchemist synthetic token's value and backing.

# Transmuter

⋮

The Transmuter is the primary price restoring mechanism for the alchemical synthetic tokens (alAssets). The Transmuter.sol contract powers the Transmuter. Harvested yield, as well as liquidation, will flow directly to the TransmuterBuffer. The TransmuterBuffer will send funds to the Transmuter following a rate and limit set by governance, thus ensuring that Transmuter depositors are guaranteed a 1:1 redemption of alAssets for their underlying collateral.

Users deposit alUSD into the Transmuter. As the yield, liquidations, and repayments come in, it will credit users (for example) DAI proportional to the amount of alUSD they have staked, relative to the total amount of alUSD staked. When a user chooses to withdraw the converted DAI, an equal amount of alUSD will be burned.

The rate at which funds can be converted to their underlying collateral is dependent on the flow rate of the TransmuterBuffer. More information can be found in [📄 The Transmuter, Elaborated](#). This limit ensures that the Transmuter will not be drained for tiny (0.1%) arbitragers, and also creates a front-stop (i.e. excess) pool of funds that would remain idle until eventually allowed to be transmuted. Instead of these funds remaining idle, the Transmuter was upgraded to send significant excess funds to the [📄 Elixir AMO](#).

Many users wish to estimate the conversion rate of the transmuter. Assuming there is sufficient available flow, funds will enter the transmuter whenever there is a yield harvest. Therefore, the minimum flow to the transmuter may be estimated by the average yield being earned by all depositors for the corresponding alAsset. The yield is shared with all transmuter depositors proportionally. Additional funds will be sent to the transmuter whenever any user repays or liquidates their loan, which is why using the average yield results in a minimum estimate of flow to the transmuter and transmutation rate.

# alAssets

:

## Synthetic Debt Tokens representing Future Yield

alAssets are the synthetic debt assets that represent a user's future yield. Each alAsset has at least one corresponding underlying asset that backs it (ETH for alETH, various stablecoins for alUSD).

Anyone with debt in Alchemix may repay debt with an alAsset or the underlying asset. Thus, the value of an alAsset to someone with a loan is equivalent to the value of the underlying (1:1).

In order to utilize Alchemix loans for on-chain and real world purchases, users must be able to sell them for other assets. For someone without an Alchemix debt position, an alAsset is equivalent to the underlying asset eventually - due to the [Transmuter](#). Alternatively, an alAsset may be sold to someone with an Alchemix loan that wishes to repay their debt. For these reasons, alAssets will typically be valued by the market at some discount from 1:1 with the underlying. For example, a 3rd party may purchase an alAsset from someone with debt at some value less than 1:1 because they expect to receive a 1:1 conversion from the Transmuter over a short enough time period that the discount is worthwhile.

The market price discount from 1:1 can be viewed as the price depositors pay to access their future yield up-front. The Transmuter, Elixir AMO, as well as treasury operations, help maintain sufficient alAsset liquidity and work to minimize the discount for each alAsset within frameworks outline by governance.

# Elixir AMO

## Alchemix Algorithmic Market Operator

The Elixir AMO (Algorithmic Market Operator) is a tool that makes use of the front-stop that can build up in the [Transmuter](#). When large enough, the front-stop can be deployed to the Elixir for each aAsset. This deployment results in the collateral being deposited to the corresponding Curve liquidity pool. This has a few effects:

1. The initial deposit helps rebalance the pool, thus increasing the price of the aAsset.
2. The deposit earns yield in the form of CRV and CVX, that can be sold to cover protocol expenses, kept and locked to vote for increased incentives to the pools, or sold to speed up the debt repayment of Alchemist depositors.
3. At any time the AMO can withdraw the aAsset (instead of the underlying collateral) and remove it from circulation (effectively matching the Transmuter effect of converting aUSD into collateral assets), thus acting as a second instance of increasing the price of the aAsset.

The Elixir offers faster-acting control of aAsset pricing as well as an additional revenue stream for the DAO, all while being able to match the Transmuter by shrinking the aAsset supply when necessary.

In accordance with [\[AIP-80\]](#), farming rewards from the AMO go to the following three sources, split equally into 1/3rds:

1. 1/3rd farmed assets will be relocked or sold to purchase other gauge-controlling assets at the discretion of the Alchemix BizGov SubDAO. The allocation strategy will focus on growing existing positions and ensuring new positions taken are sizable enough to be meaningful. A non-exhaustive list current assets that could be accumulated and locked includes veCRV, sdCRV, SDT, vICVX, veBAL, vIAURA, veVELO, sdBAL, among others.
2. 1/3rd of farmed assets will be sold to ETH and Stablecoins, with the exact breakdown at the discretion of the BizGov SubDAO, to fund the treasury. The treasury is allocated a budget of \$450k each quarter in accordance with [\[AIP-40\]](#).
3. 1/3rd of farmed assets will be sold to purchase ALCX. The purchased ALCX will be added to existing ALCX bribe programs, with the exact programs at the discretion of the multisig. The purpose of the allocation strategy will be to maximize aAsset liquidity and AMO revenue.



# Contracts

Verified contract address list

## ALCX

- ALCX `0xdBdb4d16EdA451D0503b854CF79D55697F90c8DF`
- gALCX `0x93dede06ae3b5590af1d4c111bc54c3f717e4b35`

## aIUSD

- aIUSD [Contract](#) `0xBC6DA0FE9aD5f3b0d58160288917AA56653660E9`
- Alchemist
  - Contract `0x5C6374a2ac4EBC38DeA0Fc1F8716e5Ea1AdD94dd`
  - Implementation `0xf547b87Cd37607bDdAbAFd9bF1EA4587a0F4aCFb`
- Transmuter (DAI)
  - Contract `0xA840C73a004026710471F727252a9a2800a5197F`
  - Implementation `0xAD2A6C1C6025bE8C703930dCd921a2Fa25220298`
- Transmuter (USDC)
  - Contract `0x49930AD9eBbcc0EB120CCF1a318c3aE5Bb24Df55`
  - Implementation `0xAD2A6C1C6025bE8C703930dCd921a2Fa25220298`
- Transmuter (USDT)
  - Contract `0xfc30820ba6d045b95D13a5B8dF4fb0E6B5bdF5b9`
  - Implementation `0xAD2A6C1C6025bE8C703930dCd921a2Fa25220298`
- Transmuter (FRAX)
  - Contract `0xE107Fa35D775C77924926C0292a9ec1FC14262b2`
  - Implementation `0xe04b5b4de60fa2fba69a93ade13a8b3b569d5b4`
- TransmuterBuffer
  - Contract `0x1EEd2DbeB9fc23Ab483F447F38F289cA15f79Bac`
  - Implementation `0x4370675aB54fc710A27e29Ea2BB5563468b5948f`
- YearnTokenAdapter
  - yDAI `0xA7AA5BE408B817A516b40Daea7a919664f13f193`
  - yUSDC `0x1D28D426e4e20B9d43130C23252b8e06F9cB388E`
  - yDAI `0x5951f159eF502f0571A5D7e136a580DcadEa42Eb`
- AaveTokenAdapter
  - aDAI `0xd5D43A11668881D17c9ADEEDF26fC06Fc26D3c67`
  - aUSDC `0xb7D9356b0D5b2367E730b583A2b140269A647b55`
  - aUSDT `0x65Bc963576D3831741F75C3Eae43ac28Fc7B4aFB`
  - aFRAX `0xc5fE32e46fD226364BFf7A035e8Ca2aBE390a68f`
- VesperTokenAdapter
  - vaDAI `0x36033E1fd88470a5192692e01150C3593847FCA5`
  - vaUSDC `0x54aeD38aBE51Bb39C8e75AC1D9d9335D6136B6a1`
  - vaFRAX `0xbc8003aA713DF9c4887700a7A42f26F06C810819`
- Whitelist
  - Alchemist `0x78537a6CeBa16f412E123a90472C6E0e9A8F1132`
  - Transmuter (DAI) `0xdd8AC2d5A739Bb4a591C5b0c7e613B810fE83ff1`
  - Transmuter (USDC) `0x35b2c16de6F283Ab0949964d90cCF930f378ade6`
  - Transmuter (USDT) `0x46f992D00C2Dfb6FbbbB64d69Ab353c2fc435ACE`
- AMO
  - AMO (aIUSD) `0x9735f7d3ea56b454b24ffd74c58e9bd85cfad31b`
  - AMO (aIUSDFRAXBP) `0x06378717d86b8cd2dba58c87383da1eda92d3495`

## aIETH

- aIETH [Contract](#) `0x0100546F2cD4C9D97f798FFC9755E47865FF7Ee6`
- Alchemist
  - Contract `0x062Bf725dC4cDF947aa79Ca2aaCCD4F385b13b5c`
  - Implementation `0xf547b87Cd37607bDdAbAFd9bF1EA4587a0F4aCFb`
- Transmuter (WETH)
  - Contract `0x03323143a5f0D0679026C2a9fB6b0391e4D64811`
  - Implementation `0xAD2A6C1C6025bE8C703930dCd921a2Fa25220298`
- TransmuterBuffer
  - Contract `0xbc2FB245594a68c927C930FBE2d00680A8C90B9e`
  - Implementation `0x4370675aB54fc710A27e29Ea2BB5563468b5948f`
- YearnTokenAdapter
  - yWETH `0x7E30fC3411AFD4c0381a4ec6e6ba09e19B9Edb5B`
- AaveTokenAdapter
  - aWETH `0x53Bff8132A9dcf65814b713542Cf658927536C8e`
- VesperTokenAdapter
  - vaETH `0xFCD619923456E20EAe298B35E3606277b391BBB4`
- wstETHTokenAdapter
  - wstETH `0x3c2c6FA5398CDD933465F6fD04F3575f7201BCF8`
- rETHTokenAdapter
  - rETH `0x7A172816C0560964E6aD54EF54AFABB035614047`
- sfrxETHTokenAdapter
  - sfrxETH `0xfa6A5D33e18CB0d52991536ab15750fB13119E45`
- Whitelist
  - Alchemist `0xA3dfCcbad1333DC69997Da28C961FF8B2879e653`
  - Transmuter (WETH) `0x211C74DB951c161c5A379363716EbDca5125EF59`
  - WETHGateway `0x3f950f06064A43d42a8e7C199BB77b8E3A4b2096`
- WETHGateway
  - Contract `0xA22a7ec2d82A471B1DAcC4B37345Cf428E76D67A`
- AMO
  - AMO (aIETH) `0xe761bf731a06fe8259fee05897b2687d56933110`

## Treasury

- 24H Timelock Multisig `0x8392f6669292fa56123f71949b52d883ae57e225`
- Developer Multisig `0x9e2b6378ee8ad2a4a95fe481d63caba8fb0ebb9`

## Fantom

- aIUSD `0xB67FA6deFCe4042070Eb1ae1511Dcd6dcc6a532E`
- gALCX `0x70F9fd19f857411b089977E7916c05A0fc477Ac9`
- AlchemistUSD `0x76b2E3c5a183970AAAD2A48cF6Ae79E3e16D3A0E`
- TransmuterBuffer aIUSD `0x5a07d36D1f543960EE7806d35827E995539Fe5CF`
- TransmuterDAI `0x486FCC9385dCd16fE9ac21a959B072dcB58912e0`
- TransmuterUSDC `0xaE653176d1AF6A68B5ce57481427a065E1baC49f`
- TransmuterUSDT `0x53F05426D48E667c6a131F17db1b6f7AC535aBC6`
- YeamAdapterDAI `0xa6f7e861fccD07aF4a5e94d050C2BE3D9FF75D9A`
- YeamAdapterUSDC `0xE68E8a676d496851BB054E10ff3915C056293102`
- YeamAdapterUSDT `0x99E439401A57D67251431B1277d4CE7BE8E3fC1`

## Optimism

- aIUSD `0xCB8FA9a76b8e203D8C3797bF438d8FB81Ea3326A`
- aIETH `0x3E29D3A9316dAB217754d13b28646B76607c5f04`
- gALCX `0x870d36B8AD33919Cc57FFE17Bb5D3b84F3aDee4f`
- Alchemist `0x1Bce0aca8B0E4139e4390Cf1A7A6eb644000A2F0`
  - ETH Alchemist Proxy `0xe04Bb5B4de60FA2fBa69a93adE13A8B3B569d5B4`
  - USD Alchemist Proxy `0x10294d57A419C8eb78C648372c5bAA27fD1484af`
- Transmuter Buffer Code - `0x36E8D12227f8C8eF38BB3aEaBf816681b5440be5`
  - Transmuter Buffer aIETH - `0x7f50923EE8E2BC3596a63998495baf2948a28f68`
  - Transmuter Buffer aIUSD - `0xe99a9A717c60F9639B235ede422c27d60FBEB3b9`
- Transmuter Code - `0x54aeD38aBE51Bb39C8e75AC1D9d9335D6136B6a1`
  - Transmuter DAI - `0xFCD619923456E20EAe298B35E3606277b391BBB4`
  - Transmuter USDC - `0xA7ea9ef9E2b5e15971040230F5d6b75C68Aab723`
  - Transmuter USDT - `0x4e7d2115E4FeEcD802c96E77B8e03D98104415fa`
  - Transmuter ETH - `0xb7C4250f83289ff3Ea9f21f01AAD0b02fb19491a`
- StaticAToken DAI - `0x43A502D7e947c8A2eBBaf7627E104Ddcc253aBc6`
- StaticAToken USDC - `0x4186Eb285b1efdf372AC5896a08C346c7E373cC4`
- StaticAToken USDT - `0x2680b58945A31602E4B6122C965c2849Eb76Dd3B`
- StaticAToken WETH - `0x337B4B933d60F40CB57DD19AE834Af103F049810`
- AAVE V3 Adapter DAI - `0x58846EFaDc0819d6F1D45a71D36b48A2ea79fcC5`
- AAVE V3 Adapter USDC - `0xF2b581F0Ef5c54384aCdA512D400E3C67a0eD692`
- AAVE V3 Adapter USDT - `0xa5f4786eb1fD403a92086841515b71610d34259b`
- AAVE V3 Adapter WETH - `0xB4294E2931b4C3cf77C9377F9A62d3D81ad82B86`
- Optimism Multisig - `0xC224bf25Dcc99236F00843c7D8C4194abE8AA94a`
- WETH Gateway - `0xDB3fE4Da32c2A79654D98e5a41B22173a0AF3933`

# Resources

## Useful links

**Website:** [www.alchemix.fi](http://www.alchemix.fi)

**Discord:** <https://discord.com/invite/alchemix>

**Twitter:** <https://twitter.com/AlchemixFi>

**Stats and data:** <https://alchemix-stats.com/>

**Developer Docs:** <https://alchemix-finance.gitbook.io/v2/>

**Front End Repository:** <https://github.com/alchemix-finance/alchemix-v2-frontend>

**Newsletter:** <https://alchemixfi.substack.com/>

**Medium:** <https://alchemixfi.medium.com/>

**YouTube:** <https://www.youtube.com/c/AlchemixFinance>

**v1 Whitepaper:** <https://alchemix-finance.gitbook.io/alchemix-finance/>



# Marketing Material



Our logos for use in promo and marketing materials. Please consult the brand book for more information. For any questions, please contact @metalface in the official Discord.

## Brand book

An overview of how to use our brand in your material



AlchemixBrandBook\_01.pdf 514KB  
PDF

## The Alchemix logo

### Bronze



### White

### Black




# Alcx logo - Bronze



Our master logo for use in all situations unless contrast is an issue.


## SVG

 **ALCX Full Std logo.svg** 5KB  
Image


Our standard horizontal logo


 **ALCX Stacked Full Std logo.svg** 4KB  
Image

Our vertically stacked variant


 **ALCX Std logo.svg** 1KB  
Image

The Alchemix logo solo


 **ALCX Std Filled logo.svg** 1KB  
Image


 **ALCX Std logo\_Thick.svg** 1KB  
Image

For smaller uses where legibility is an issue


 **ALCX Std Filled logo\_Thick.svg** 1KB  
Image

## PNG

 **ALCX-Full-Std-logo.png** 23KB  
Image

 **ALCX-Stacked-Full-Std-logo.png** 23KB  
Image

 **ALCX-Std-logo\_Thick.png** 11KB  
Image

 **ALCX-Std-logo.png** 11KB  
Image

## JPG

 **ALCX-Full-Std-logo\_Blackbg.jpg** 63KB  
Image

# Alcx logo - White



Our white logo for use where contrast is an issue.

## SVG



**ALCX Full Std logo.svg** 5KB  
Image

Our standard horizontal logo



**ALCX Stacked Full Std logo.svg** 5KB  
Image

Our vertically stacked variant



**ALCX Std logo.svg** 1KB  
Image

The Alchemix logo solo



**ALCX Std logo\_Thick.svg** 1KB  
Image

For smaller uses where legibility is an issue

## PNG



**ALCX-Full-Std-logo.png** 20KB  
Image



**ALCX-Stacked-Full-Std-logo.png** 19KB  
Image



**ALCX-Std-logo.png** 8KB  
Image



**ALCX-Std-logo\_Thick.png** 9KB  
Image

# Alcx logo - Black

Our black logo for use where contrast is an issue.

## SVG



**ALCX Full Std logo.svg** 5KB  
Image

Our standard horizontal logo



**ALCX Stacked Full Std logo.svg** 5KB  
Image

Our vertically stacked variant



**ALCX Std logo.svg** 1KB  
Image

The Alchemix logo solo



**ALCX Std logo\_Thick.svg** 1KB  
Image

For smaller uses where legibility is an issue

## PNG



**ALCX-Full-Std-logo.png** 18KB  
Image



**ALCX-Stacked-Full-Std-logo.png** 17KB  
Image



**ALCX-Std-logo\_Thick.png** 8KB  
Image



**ALCX-Std-logo.png** 8KB  
Image

# alAssets



## alETH



**alETH.svg** 934B  
Image



**alETH\_thick.svg** 939B  
Image



**alETH.png** 12KB  
Image



**alETH\_thick.png** 12KB  
Image

## alUSD



**alUSD.svg** 4KB  
Image



**alUSD\_thick.svg** 4KB  
Image



**alUSD.png** 11KB  
Image



**alUSD\_thick.png** 11KB  
Image

# Financial reports

Quarterly financial reports.

## 2023

Q1 - 2023



Alchemix\_Q1\_2023\_Financial\_Report.pdf 2MB  
PDF

## 2022

Q4 - 2022



Alchemix\_Q4\_2022\_Financial\_Report.pdf 3MB  
PDF

Q3 - 2022



Alchemix\_Q3\_2022\_Financial\_Report.pdf 2MB  
PDF

Q2 - 2022



Alchemix\_Q2\_2022\_Financial\_Report.pdf 2MB  
PDF

Q1 - 2022.



Alchemix\_Q1\_2022\_Financial\_Report.pdf 1MB  
PDF

# Multisig Admin Rights

:

## Access Control

### Alchemist Contracts

Alchemist contracts for aUSD and aETH allow the user to create a collateralized debt position by depositing tokens as collateral and taking debt against the collateral by minting aUSD or aETH.

Each Alchemist Contract has the following privileged roles: Admin, Sentinel, and Keeper, with the following privileges:

#### 1. Admin

- Add tokens to the list of underlying and yield tokens supported by the Alchemist
- Enable and disable tokens from this list
- Add and remove sentinels and keepers
- Transfer the admin role to a different address (must be accepted by the new admin)
- Configure the Alchemist's parameters, including limits, fees, and the addresses of the transmuter and protocol fee reciever
- Reset ("snap") the expected value of a yield token to the current value. Since deposits, withdrawals of the underlying, and liquidations are blocked if the value of a yield token suddenly drops significantly below its expected value, this can prevent the contract from becoming unusable if the yield token doesn't recover, or takes too long to recover.
- Disable or enable whitelisting requirements

#### 2. Sentinel

- Sentinels can disable (ie, pause) underlying and yield tokens. `pauseUnderlyingToken()` will disable `deposit()`, `depositUnderlying()`, `repay()`, and `liquidate()` functionality for the given underlying token. `pauseYieldToken()` will disable `deposit()`, `depositUnderlying()`, `withdraw()`, `withdrawUnderlying()`, `liquidate()` and `harvest()` for the given yield token (ie, given yield strategy). See **#Pause Control** below for more information.

#### 3. Keeper

- Keepers can trigger harvests of the yield tokens.

## Upgradeability

One major design choice of note in Alchemix v2 is that of upgradeability. All 3 major contracts (AlchemistV2, TransmuterV2, and TransmuterBuffer) are built to be used via upgradeable proxies. This entrusts the Alchemix DAO with the ability and responsibility to upgrade the functionality of these pieces as it sees fit.

## Pause Control

Sentinels have the ability to pause yield tokens should there be an issue. Admins may then unpause the tokens. When an underlying token is disabled, it should be noted that the `withdraw()`, `withdrawUnderlying()`, `repay()`, `mint()`, and `burn()` (ie, repay debt with aAssets) functions can still be called - allowing users to settle their debt and withdraw the yield token or underlying token.

Each accepted yield token has a configured maximum amount of loss that it can experience and still function normally. If the yield strategy loses more than the specified `maximumLoss`, then the yield strategy is paused automatically, meaning users may not make any deposits, may not liquidate or repay, and may not take a new loan with these strategies. Harvests will also be disabled. Lastly, users will be unable to withdraw collateral in the form of the underlying asset. Users will still be able to repay their loan and withdraw the yield token, however. For example if the `maximumLoss` was exceeded, a user could not withdraw DAI from a strategy that uses yDAI, but they would still be able to repay their loan with DAI to withdraw their yDAI collateral).

Sentinels also have the ability to pause underlying tokens should there be an issue. This would only be applicable to aAssets with multiple underlying tokens, such as aUSD. If an underlying token is paused, the `deposit()`, `depositUnderlying()`, `liquidate()`, and `repay()` functions would be disabled for that token. Notably, debts may still be paid down by harvests and users may repay debt with aAssets or other underlying tokens and withdraw their funds.

## Timelock

The Alchemix Timelock Multisig acts as the admin for the Alchemix contracts. The timelock is 24h, to allow for a delay between queueing changes and changes being implemented (changes being contract upgrades or exercising of admin powers). Alchemix also maintains a developer multisig with discretionary funds to be used for transactions that are infeasible with a 24h multisig. 24h was chosen to allow adequate time to review while accommodating the fast-moving nature of DeFi.

## More Information

For more in-depth information on admin controls and contract features, see the [v2 audit](#) and [developer docs](#).



# Audits



- Alchemix v2 was audited by Runtime Verification. Alchemix continues to engage Runtime Verificaiton for additional audits on protocol changes. The v2 audit report can be found here: [https://github.com/runtimeverification/publications/blob/main/reports/smart-contracts/Alchemix\\_v2.pdf](https://github.com/runtimeverification/publications/blob/main/reports/smart-contracts/Alchemix_v2.pdf)
- Alchemix offers a bug bounty program through ImmuneFI. The program can be found here: <https://immunefi.com/bounty/alchemix/>
- Alchemix ran a one-off code4rena contest. The contest can be found here: <https://code4rena.com/contests/2022-05-alchemix-contest>

# Alchemix DAO

## Purpose of the DAO

The Alchemix DAO will receive income from the Alchemix protocol. The treasury will pay developers and permanent staff to maintain and expand the protocol. In addition, treasury funds will pay for audits of the protocol and any of its future products.

The Alchemix DAO will fund projects that build on Alchemix or use the Alchemist synthetic tokens in their applications. Reach out to the developers and mods on the community discord if you have an idea for something to build. If the community likes your idea and approves it via a token vote, then you will receive funding.

Alchemix believes in supporting the greater Ethereum community and plans to donate a portion of treasury income to gitcoin grants and other charitable programs.

ALCX token holders can signal a desire to make changes to the protocol via governance voting.

## Bug Bounty Program

While the Alchemix team is confident in the security of the smart contracts, it cannot be assumed that there is not an exploitable attack vector or a hidden bug in the protocol. If you have found a vulnerability, please discreetly contact the development team or discord mods. A significant amount of the ALCX supply is reserved to paying bug bounties, and we will pay handsomely to protect the protocol. The Alchemix team have observed many teams being stingy with bug bounty rewards. Those who disclose vulnerabilities will not be shortchanged because protecting users' funds is prioritized above all else.

Alchemix also has an Immunefi program where you can submit bug reports here - <https://immunefi.com/bounty/alchemix/>

## Governance Process

The Alchemix DAO is currently run through a developer multisig with signaling through the Snapshot app. ALCX Tokens give holders governance rights over the direction of the protocol and use of the treasury. Community members may create proposals by following the [Community Governance Process](#).

## The Future

After the Alchemix DAO has matured, the development team will propose their vision for full on-chain governance, giving complete control to the community for a number of protocol parameters. The development of the DAO will be an open process. There are multitudes of brilliant and savvy people in the crypto space, and the Alchemix development team takes community feedback seriously. Thus, the community's input may heavily influence the direction and design of the DAO.

# ALCX Token Distribution

:

The Alchemix team believes in the values of fair launch. The Alchemix protocol was developed independently, and the team did not take any money or tokens from outside investors for the entire period up until launch. We believe the majority of the token recipients should be those who help to contribute to the protocol. Strategic sales of tokens would be considered if it assists further development of the protocol.

ALCX is a token that grants governance rights in Alchemix DAO. There are no guarantees it has any value, whatsoever. ALCX will not have a hard-cap, but does have a carefully crafted **emissions schedule**. For the purpose of the token distribution, we are basing the numbers below on the total expected supply after 3 years.

The token distribution is as follows:

- The Alchemix DAO received a premine of 15% of the projected supply after three years. Usage of these tokens and proceeds from them is completely up to the community's discretion.
- The Alchemix DAO has an additional reserve of 5% of the projected three-year supply for bug bounties.
- The remaining 80% of the tokens can be obtained by staking certain tokens and liquidity pool tokens in the **Staking Pools** contract.
  - The Alchemix founders, onboarded developers, and community developers who build on Alchemix will have access to an exclusive staking pool, which will receive 20% of the ALCX block reward. This equates to 16% of the supply after 3 years.
  - Stakers and liquidity providers are eligible to obtain 80% of the ALCX block reward, equating to 64% of the supply after 3 years. Currently, a portion of these emissions are being sent to the treasury.

This token distribution allocates a large majority of tokens to those who contribute to the Alchemix protocol. The token distribution assures that no one from the development team will have enough tokens to control the protocol while rewarding them for their work and incentivizing them to continue working on the protocol.

# Staking Pools

:

Alchemical synthetic tokens require deep liquidity in order to be maximally effective. The StakingPools.sol contract powers Staking Pools. Its primary purposes are distributing ALCX tokens to the community and providing liquid pairs for tokens in the Alchemix ecosystem.

At launch, there were four pools, one for aUSD, one for ALCX, one for aUSD/DAI SLP (Sushi Liquidity Pair) tokens, and one for ALCX/ETH SLP tokens. Since then aETH has also been added as a synthetic asset option. The emissions distribution is modified through governance proposals. Below are the types of pools, how they are incentivized, and their reasoning for existing:

- **aUSD and aETH LP tokens.** Establishing a peg as close to \$1/1 ETH as possible and providing deep liquidity for aUSD and aETH is important for the value proposition of Alchemix as well as liquidity provider confidence. Doing so will give real utility for aUSD as an advance on yield farming income as it can easily be converted to DAI, USDC, or USDT with low slippage. Incentivization is carried out through protocol owned sdCRV and vICVX, as well as ALCX emissions going to staking pools and vote bribing platforms.
- **ALCX/ETH SLP tokens.** This pair exists to provide liquid markets for the ALCX governance token.
- **ALCX tokens.** This pool is to reward holders of ALCX who may be too risk-averse to join in the ALCX/ETH pool, and serves as an anti-dilutive measure for ALCX holders. The longevity of this pool will be determined by the community.

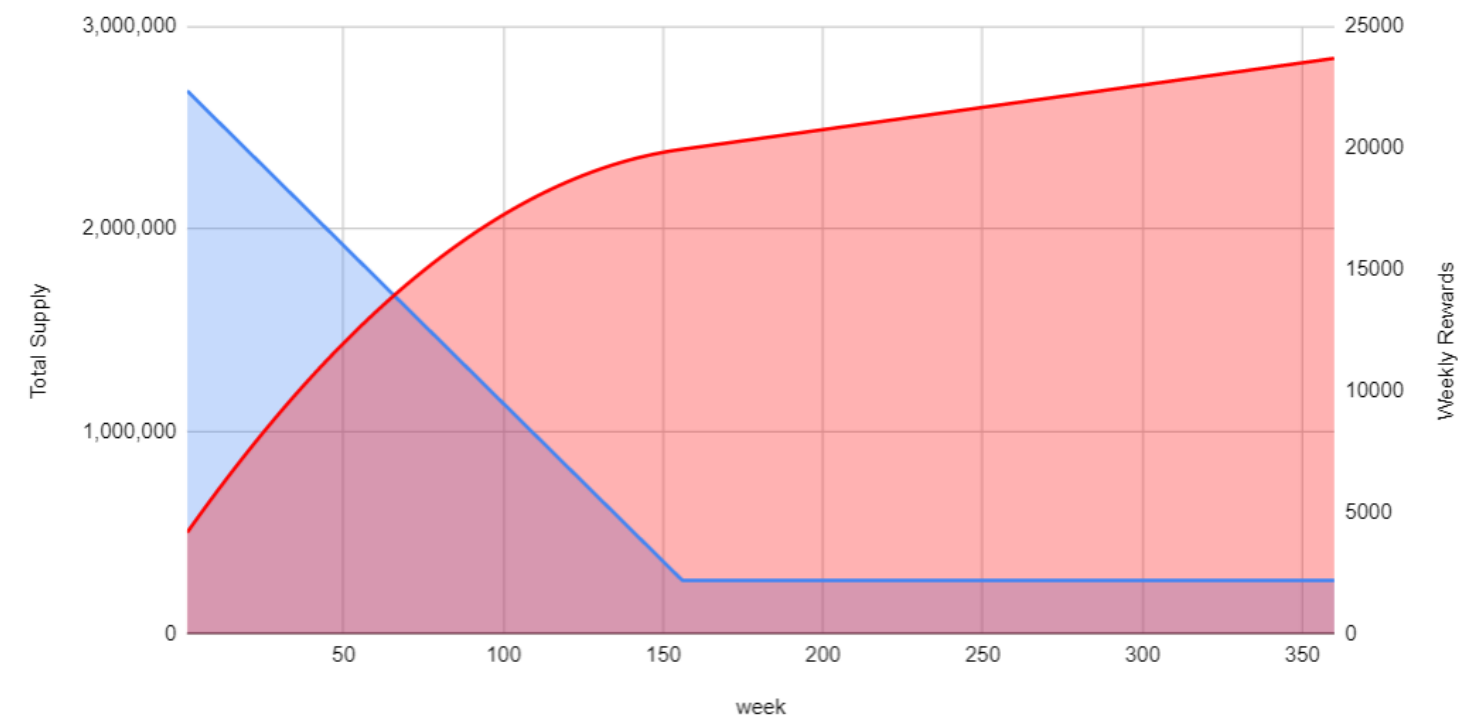
These pools and the weights thereof will be adjusted as Alchemix brings more alchemical synthetic tokens to market. The priority will be incentivizing synthetic pairs with their base asset.

# ALCX Emissions Schedule

The ALCX governance token's distribution is carefully crafted to gradually reduce issuance over the course of three years, continuing with a long tail of fixed weekly emissions. Emissions are the rate at which new ALCX tokens are minted.

There will be an initial pre-mine of 478,612 ALCX tokens. This calculation is derived from an estimated token supply of 2,393,060 ALCX tokens in circulation after 3 years, with 15% of the supply going to the DAO treasury, and 5% to the bug bounty program. 358,959 ALCX will be allocated to the treasury for use as determined by the community. 119,653 ALCX will be held by the treasury and is reserved for bug bounties.

**ALCX: Weekly Rewards and Supply Over Time**



Approximately 22,344 ALCX tokens will be distributed from Staking Pools in week one of staking, and the amount will decrease by approximately 130 ALCX per week for three years. Because ALCX rewards are calculated per block, network conditions could slightly affect the emission schedule, but the effect should be negligible. After three years, there will be a flat 2200 ALCX weekly emission, increasing the total supply by 114,400 ALCX annually. At the three year point, there will be approximately 4.5% annual inflation of supply, and it will gradually decrease over time.

# Community Governance Process

This page outlines the process for governance proposals that originate in the community.

If you are interested in submitting a proposal to the Alchemix DAO, then you will need to follow the process below.

## Step 0

- ✔ It is recommended you gather sentiment for your proposal idea through discussion with the community in the discord server, fireside chats, DMs with other community members, and/or writing a draft document to share.

## Step 1

- ✔ Post your proposal to <https://forum.alchemix.fi/public/> with the tag "Ecosystem Development".

## Step 2

- ✔ Share the Alchemix forum link to your proposal in the #governance-proposals channel in the Alchemix discord server at [discord.com/invite/alchemix](https://discord.com/invite/alchemix).

## Step 3

- ✔ After a minimum of 5 days have passed since completing Step 1 AND Step 2, you may post your proposal to the "Alchemixed Opinions" Snapshot (the platform for Alchemix governance temperature checks). Snapshot link: <https://snapshot.org/#/alchemixedopinions.eth>

1. A minimum of 50 ALCX is necessary to post a proposal. If you do not have 50 ALCX, you can ask someone with 50 ALCX to post on your behalf.
2. Proposals will be live for 3 days.
3. Voting options must only be For, Against, or Abstain.

## Step 4

- ✔ If greater than 50% of the non-abstain vote is "For", and a quorum of 5k ALCX voting is met, the proposal will move to the official Alchemix Proposal Snapshot as an AIP (Alchemix Improvement Proposal) at <https://snapshot.org/#/alchemixstakers.eth>

## Step 5

- ✔ Official AIPs will have a quorum of 35k ALCX. If quorum is met, then the multisig will be directed to execute the most popular voting option unless directed otherwise by the voting parameters.

## Supplementary Information

### Common Proposal Types

1. A change to the protocol treasury and/or how it is utilized.
2. A change to protocol parameters (such as transmuter flow rate).
3. A grant, donation, or funding request.
4. Deploying a new protocol-related contract or contract upgrade (ie, deploying a new yield strategy or deploying on a new chain).

### What Makes a Good Proposal?

1. The first item in the proposal should summarize only the action items on what exactly is being proposed. It should not include any justification in order to be as succinct as possible.
2. The introduction of the proposal should be an introduction giving context to the reason for the proposal and what the body of the proposal contains.
3. The body of the proposal should include in-depth analysis and justification for what is being proposed.
4. The end of the proposal should state the voting options that will appear on snapshot and what they mean (For, Against, Abstain).
5. Any proposal that requires a high level of effort (new features, new deployments, grant requests etc) should identify who is going to do the work. For example, a poorly written proposal would simply suggest a new feature. A well written proposal would suggest a new feature and provide verification that Core team is willing and able to build the feature, OR it should state who is building the feature. If a grant request is included, then there is more onus on the proposer to justify why they will be able to build the feature. Treasury actions and changing protocol parameters will typically only take a handful of multisig transactions - in these cases it can be assumed the multisig will carry out the proposal if passed.

# Technical Contributions

:

These pages describe guidelines for technical contributions to the Alchemix ecosystem.

Technical contributions to the Alchemix ecosystem can come in many forms, including:

- Token adapters for new yield sources
- Integrations with other DeFi protocols
- Alchemix-specific tools and utilities

However, in order to integrate any new smart contracts into an AlchemixV2 debt system, some governance actions will be required. Contributors should use the following procedures to guide them as they prepare to build and integrate code into the Alchemix ecosystem.



# Token Adapters

The following details the steps necessary for integrating a new adapter into the Alchemix V2 protocol.

## Token Adapter Governance Process

There are three steps, including 2 separate AIPs (Alchemix Improvement Proposals), needed to get an adapter approved and connected in an Alchemix V2 debt system. The first AIP is technically optional, as both AIPs could be condensed into a single AIP if the integration developer is comfortable putting in the development work upfront without pre-approval.

The [Community Governance Process](#) details the general governance steps that should be followed for each AIP.

### Step 1

- ✔ Propose the new yield source for integration, and request grant funding.

The purpose of this step is for the integrator/proposer to verify that the Alchemix DAO wants to integrate the proposed yield strategy. Additionally, the integrator/proposer can request a pre-approved grant of ALCX tokens, to be paid out when the Adapter is deployed in step 2.

A template for Step 1 proposals will be provided in the future.

### Step 2

- ✔ Write, deploy and verify the ITokenAdapter compliant adapter. See [Technical Requirements](#) below.

### Step 3

- ✔ Propose integration of the new yield source using the new adapter.

The following parameters need to be approved in at least one of the two AIPs:

- Target network (eg. ETH Mainnet, Optimism, etc...)
- Yield bearing asset name & address (include etherscan & github links)
- Collateral asset name & address (include etherscan & github links)
- [Maximum Loss](#) expressed in basis-points, eg. 50 for 0.5% [more info](#)
- Deposit cap (expressed in units of underlying collateral) [more info](#)
- Credit unlock blocks (how long after a harvest does it take for the yield to be distributed to depositors) [more info](#)

The following needs to be approved as well, once development and deployment are complete:

- Adapter name & address
  - include etherscan link
  - include github link to solidity code in the [v2-foundry repo](#)
  - include github link to deployment artifacts in [deployments repo](#)
- Multisig transaction details that should be executed by the Alchemix dev multisig, detailed [here](#)

- ⓘ NOTE: To be clear, all of the above bullet points only need to be approved ONCE by governance. It is up to the the builder whether or not they want pre-approval before creating and deploying the new adapter, or if they want to make a single AIP for approval once the adapter is built, deployed, and verified.

## Technical Requirements

1. Build a token adapter that is compliant with the [ITokenAdapter interface](#), along with a set of unit & integration tests, and make a pr against the master branch of the [Alchemix V2 Repo](#).
2. Once the pr is approved and merged by the core team, deploy the contract to the target network.
3. Make a pr against the master branch of the [deployments repo](#) that includes the artifacts from the deployment (json file containing, at a minimum, the **abi & address** of the deployed adapter).

## Dev Multisig Transactions

Relevant addresses for already-deployed Alchemix contracts can be found in the [deployments repo](#).

### Enable a new adapter

1. TARGET\_ALCHEMIST\_ADDRESS.addYieldToken(YIELD\_TOKEN\_ADDRESS, (ADAPTER\_ADDRESS, MAXIMUM\_LOSS, MAXIMUM\_EXPECTED\_VALUE, CREDIT\_UNLOCK\_BLOCKS));
  1. YIELD\_TOKEN\_ADDRESS = the address of the yield token being integrated
  2. ADAPTER\_ADDRESS = the address of the newly deployed adapter
  3. MAXIMUM\_LOSS = the maximum loss value (in bps) from the AIP
  4. MAXIMUM\_EXPECTED\_VALUE = the deposit cap value (in units of underlying collateral) from the AIP
  5. CREDIT\_UNLOCK\_BLOCKS = the credit unlock blocks value from the AIP
2. TARGET\_ALCHEMIST\_ADDRESS.setYieldTokenEnabled(YIELD\_TOKEN\_ADDRESS, true);
  1. YIELD\_TOKEN\_ADDRESS = the address of the yield token being integrated

### Upgrade an adapter

(If the newly deployed adapter is an upgraded adapter for an existing yield token)

1. TARGET\_ALCHEMIST\_ADDRESS.setTokenAdapter(YIELD\_TOKEN\_ADDRESS, ADAPTER\_ADDRESS);
  1. YIELD\_TOKEN\_ADDRESS = the address of the yield token being integrated
  2. ADAPTER\_ADDRESS = the address of the newly deployed adapter

### Create a harvest job for the Alchemix Keeper

1. HARVEST\_RESOLVER\_ADDRESS.addHarvestJob(true, YIELD\_TOKEN\_ADDRESS, ALCHEMIST\_ADDRESS, MINIMUM\_HARVEST\_AMOUNT, MINIMUM\_DELAY, SLIPPAGE\_BPS);
  1. details on these parameters can be found [here](#)
  2. MINIMUM\_HARVEST\_AMOUNT should be set to a value that can be expected to be harvested every 1-2 days
  3. MINIMUM\_DELAY should be set to 1-2 days

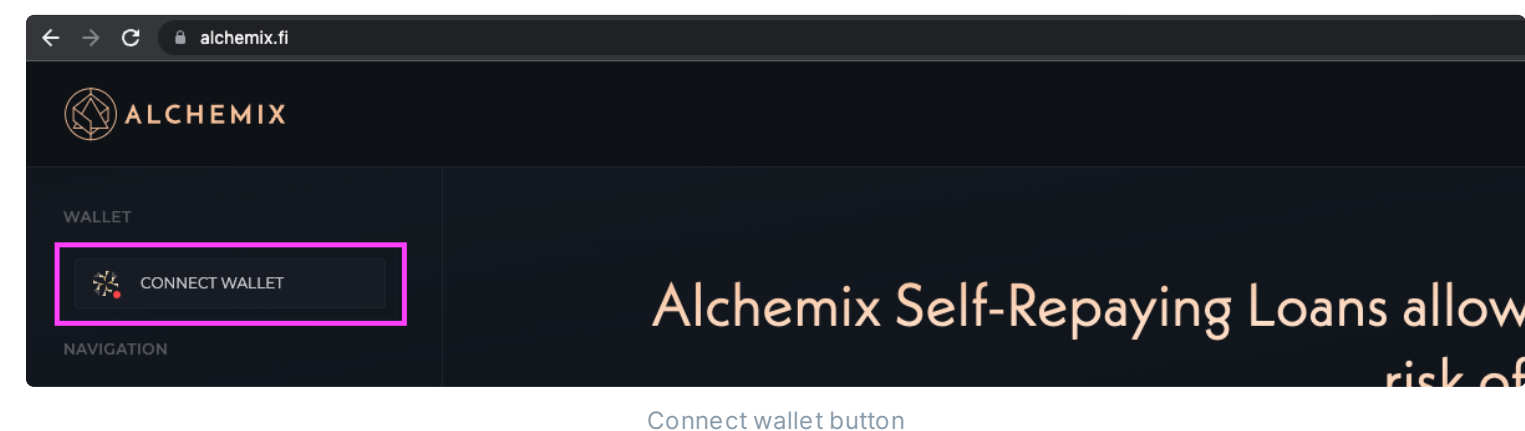
# Site overview

Fellow Alchemists, welcome to the Alchemix v2 user interface.

This guide will help you familiarize yourself with the platform.

First and foremost the v2 website is a place where you'll be able to setup and manage your Alchemix loans, farm with your tokens, use the transmuter and vote on governance proposals. The first time you arrive at the site, you'll see the introduction page that tells you about the product and shows you what base assets are supported along with their current loan to values. This list will grow continuously as new assets are added. If you ever want to return here just click on the Alchemix logo at the top left of the page or the Intro button in the footer.

First you'll need to connect your wallet. Click the 'Connect wallet' button and follow the prompts to select your wallet provider.



Connect wallet button

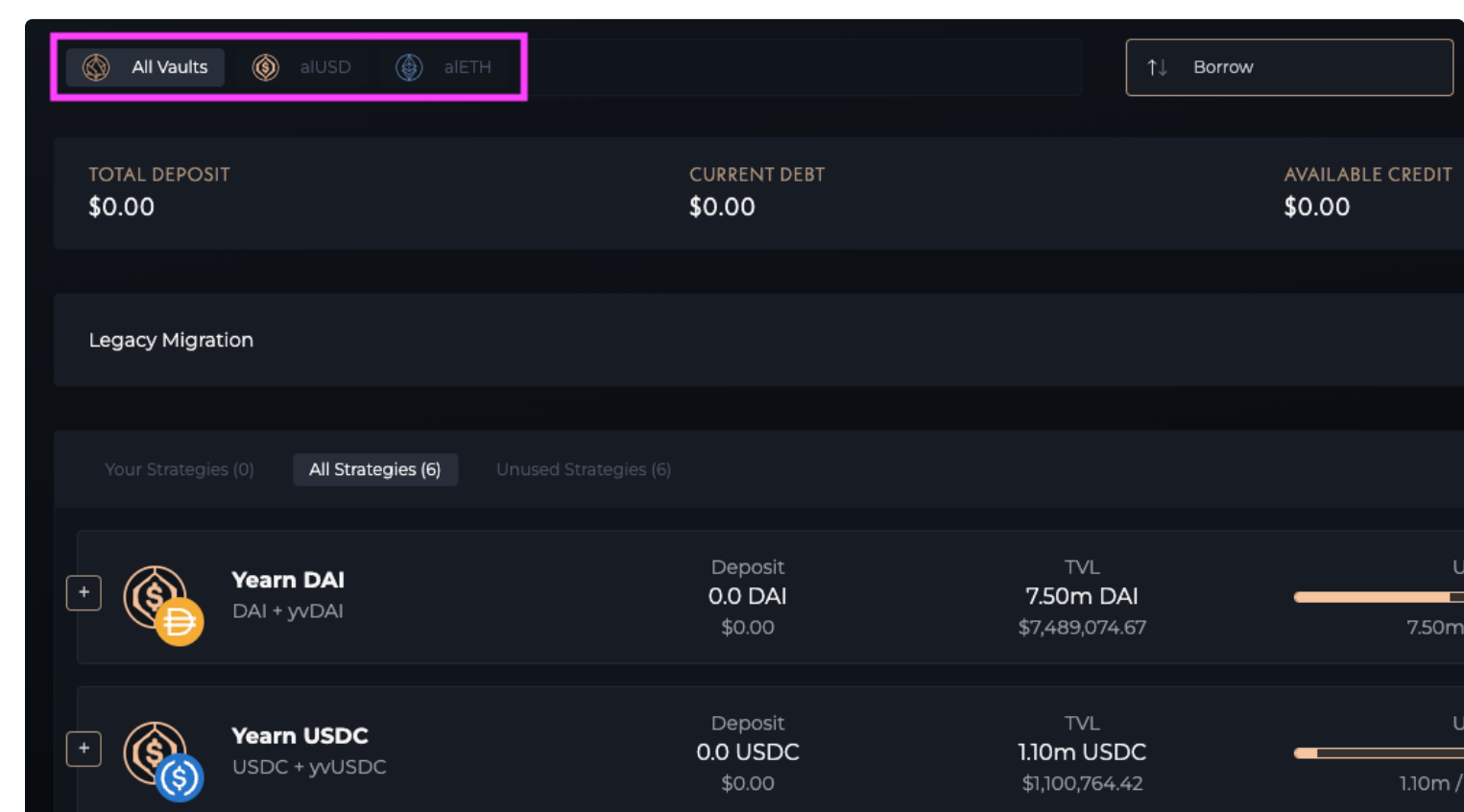
Once you've connected to your chosen wallet, you'll be taken to the Vaults page that's associated with your connected address.

aAssets are a fundamental concept to understand when using Alchemix. Whenever you take out a loan against your deposit it will always be issued in the form of an aAsset. For example aUSD is the asset you'd borrow against all stable coin assets you've deposited as collateral. aAssets are always pegged 1:1 to the assets you deposit. The stable coin collateral you've deposited could be made of a variety of stable coins, for example you may have deposited a mixture of Dai, USDC and Tether. Since these are all dollar pegged assets you'll only need to borrow one asset, aUSD.

Let's take a look at the vaults page.

Whether you have an existing loan or not the vaults page will display the current available strategies. You can take advantage of any of these, they cover the full range of assets that we currently support. This list will grow longer over time. To help manage what's visible, you can use the toggles to filter for the aAssets you're most interested in.

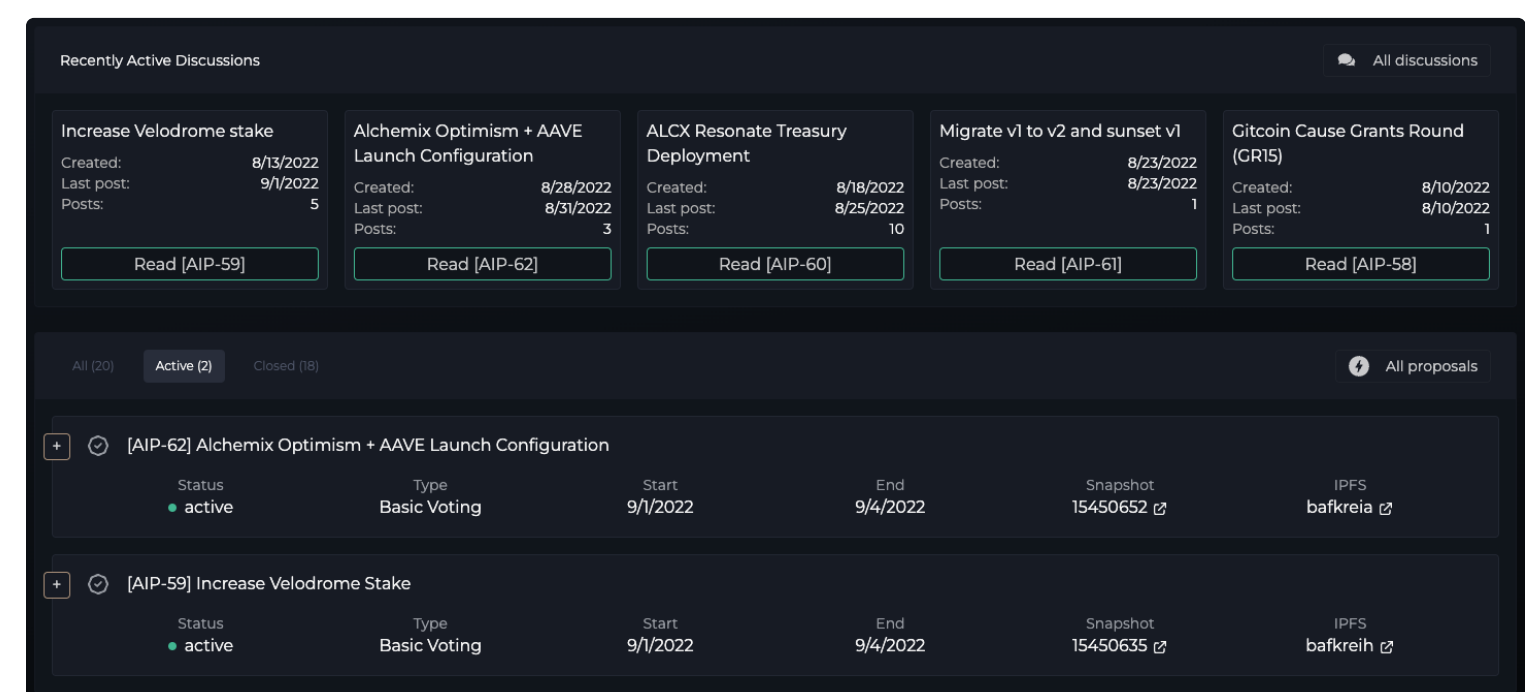
Each vault listed will show the collateral assets you can deposit and the current APY.



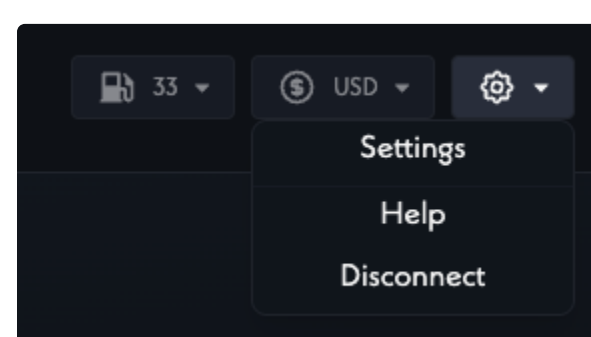
Moving on, let's check out the Transmuter page. This is where you can swap your aAssets at a guaranteed 1:1 ratio for equal value tokens. Unlike the open market where pegged assets can fluctuate in value by small amounts, the transmuter ensures you won't suffer any loss or slippage. The drawback however is that it takes time to swap your assets. So if you need to exchange immediately we recommend you use other exchanges. We've linked a few external swap providers at the top of the page.

Next, let's look at the Farms page. Here you can stake LP tokens and single sided ALCX to earn rewards. You can also wrap your ALCX into gALCX which automatically stakes your ALCX into the single sided pool and auto-compounding rewards. gALCX can also be used cross chain and on layer 2 networks as we roll out our multi-chain strategy.

Moving on to Governance. Here you can see a list of all the previous and active Alchemix improvement proposals or AIPs. Holders of ALCX are eligible to vote on any active proposal.



One last thing to draw your attention to. The website will store some user preferences on your computer's local storage, should you wish to set your preferred display currency, gas defaults and language. You can set them either at the top of any page or navigate to the settings page.



If you have any support queries, please contact our team in the official [Discord channel](#)

In the next guide, we'll look at how to deposit funds.

# Deposit funds

The unique thing about Alchemix loans is that they allow you to leverage your wealth without risk of liquidation. Another way to put this is that Alchemix allows you to borrow against an asset without carrying the risk of losing your collateral in the event of a market crash.

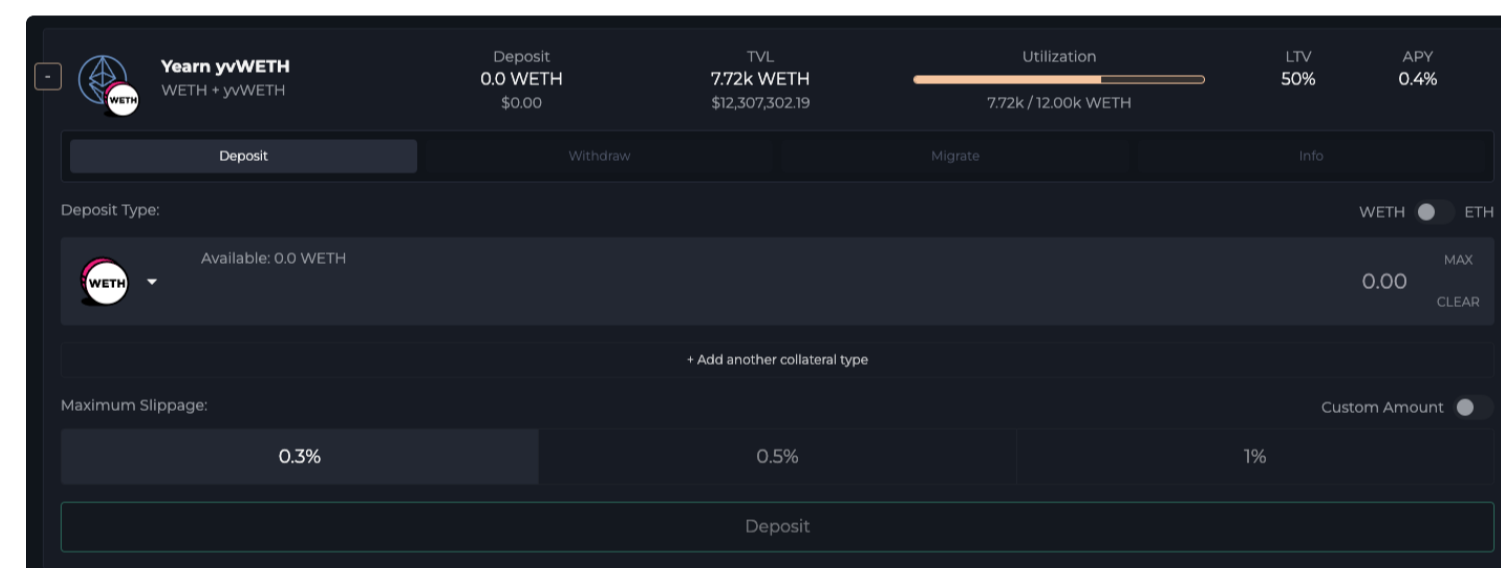
In order to setup a new Alchemix self-repaying loan you'll need to visit the Vaults page. Here you'll be presented with a list of available vaults where you can deposit any of the currently supported collateral assets.

 <b>Yearn yvDAI</b> DAI + yvDAI	Deposit 0.0 DAI \$0.00	TVL 6.97m DAI \$6,969,032.91	Utilization 6.97m / 25.00m DAI	LTV 50%	APY 1.3%
 <b>Yearn yvUSDC</b> USDC + yvUSDC	Deposit 0.0 USDC \$0.00	TVL 997.55k USDC \$997,552.44	Utilization 997.55k / 15.00m USDC	LTV 50%	APY 0.21%
 <b>Yearn yvUSDT</b> USDT + yvUSDT	Deposit 0.0 USDT \$0.00	TVL 10.69k USDT \$10,677.22	Utilization 10.69k / 15.00m USDT	LTV 50%	APY 0.12%
 <b>AAVE aDAI</b> DAI + aDAI	Deposit 0.0 DAI \$0.00	TVL 7.91k DAI \$7,902.39	Utilization 7.91k / 5.00m DAI	LTV 50%	APR 0.74%
 <b>AAVE aUSDC</b> USDC + aUSDC	Deposit 0.0 USDC \$0.00	TVL 17.65k USDC \$17,647.61	Utilization 17.65k / 5.00m USDC	LTV 50%	APR 0.3%
 <b>AAVE aUSDT</b> USDT + aUSDT	Deposit 0.0 USDT \$0.00	TVL 399.45k USDT \$399.13	Utilization 399.45 / 5.00m USDT	LTV 50%	APR 1.15%
 <b>Yearn yvUSDT</b> USDT + yvUSDT	Deposit 0.0 USDT \$0.00	TVL 0.0 USDT \$0.00	Utilization 0.00 / 15.00m USDT	LTV 50%	APY 0.53%
 <b>Yearn yvWETH</b> WETH + yvWETH	Deposit 0.0 WETH \$0.00	TVL 7.72k WETH \$12,307,302.19	Utilization 7.72k / 12.00k WETH	LTV 50%	APY 0.4%
 <b>Lido wstETH</b> WETH + wstETH	Deposit 0.0 WETH \$0.00	TVL 1.55k WETH \$2,463,494.56	Utilization Vault Full	LTV 50%	APR 3.89%
 <b>Rocket rETH</b> WETH + rETH	Deposit 0.0 WETH \$0.00	TVL 387.46 WETH \$617,611.17	Utilization Vault Full	LTV 50%	APR 3%
 <b>AAVE aWETH</b> WETH + aWETH	Deposit 0.0 WETH \$0.00	TVL 1.87k WETH \$2,976,963.04	Utilization 1.87k / 3.00k WETH	LTV 50%	APR 2.98%

Under each vault listed you'll see which tokens can be deposited into that vault.

Let's see how you can borrow against some of your ETH holdings with a new aETH loan.

First click on the + button on the Yearn WETH vault. This will open the deposit section which offers you several deposit options.



Let's go from the top. The LTV tells you how much you're able to borrow against your deposit. 50% means you'll be able to borrow a maximum of half the deposited amount.

In this example the vault accepts WETH or yvWETH. Luckily there's a handy conversion tool built into the WETH vault that allows you to convert your ETH to WETH during the deposit.




All you need to do is flick the toggle and the input box allows you to input the ETH amount you'd like to deposit.

Let's input a value of 1 ETH.

Next we have slippage. With certain vaults, (like the Yearn vaults) when you deposit collateral Alchemix will convert it into the Yearn equivalent. This is how Alchemix is able to earn yield on your deposit in the background. In this case it will exchange WETH to yvWETH for you. Because exchange rates fluctuate, your vault will receive a slightly different amount of yvWETH in return. This is usually minimal and not something you need to worry about. In order to protect our users, you're able to set your own slippage limits using the buttons provided, or specify a different amount here.

Now you can press 'deposit' and authorize the transaction in your wallet. If it's the first time you've deposited into one of our vaults, two transactions will need to be confirmed. The first is the token approval and the second will be the actual deposit.

Once your transaction has completed you'll be able to see how much you've deposited in the vault.

 <b>Yearn yvUSDT</b> USDT + yvUSDT	Deposit 0.0 USDT \$0.00	TVL 174.36k USDT \$174,069.23	Utilization 174.36k / 20.00m USDT	LTV 50%	APY 0.01%
 <b>Yearn WETH</b> WETH + yvWETH	Deposit 0.0 WETH \$0.00	TVL 8.20k WETH \$13,101,883.54	Utilization 8.20k / 12.00k WETH	LTV 50%	APY 0.17%
 <b>Lido wstETH</b> WETH + wstETH	Deposit 1.00 WETH \$1,593.20	TVL 1.60k WETH \$2,567,143.20	Utilization Vault Full (160k WETH)	LTV 50%	APR 4.41%

If you have any support queries, please contact our team in the official [Discord channel](#)

Now let's look at how to take a Self-Repaying Loan

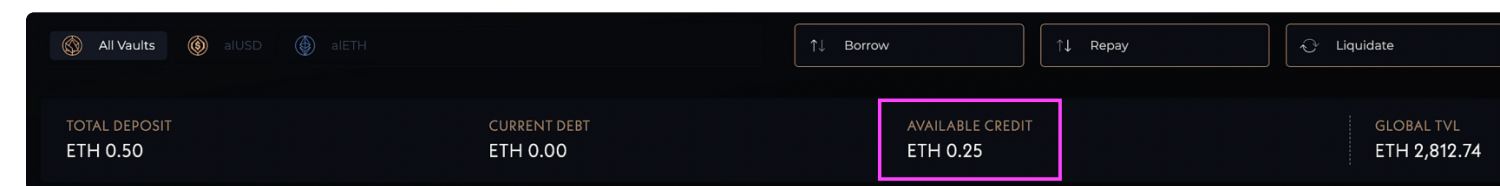


# Take a Self-Repaying Loan

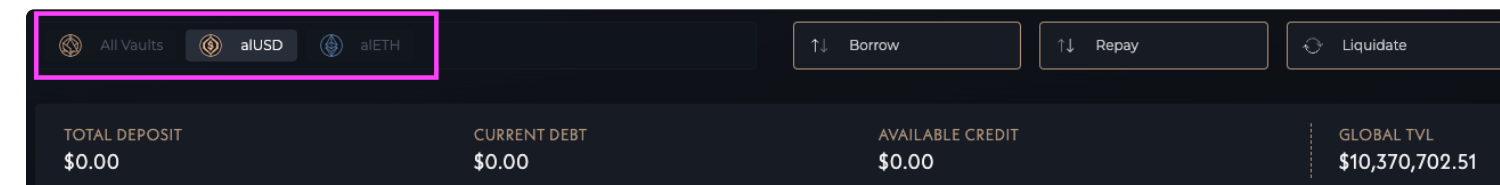
Now you've made a deposit into Alchemix. Your deposit will automatically start earning yield. You can choose to borrow against it or you can leave it to work earning credit.

We're going to borrow against our deposit. The amount you can borrow depends on the value of the tokens you have deposited.

You can see how much you have available to borrow by clicking the Vaults page, and looking at the 'Available Credit' amount.



If you have positions across several vaults, you can use the filters to see individual credit available on each vault.



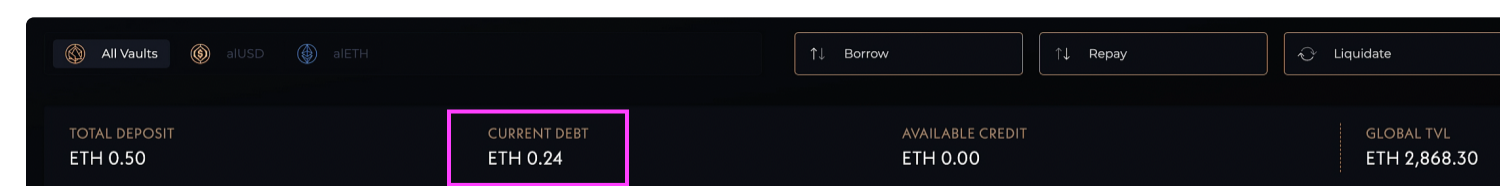
alAsset filters

Total deposit is the total amount of collateral you have deposited into Alchemix. This may be across one, or multiple vaults depending on how many you have set up. Debt is the total amount you have borrowed against your deposited collateral. Interest shows how much you've earned which adds to the amount of available credit you have access to. The difference between interest and debt is that interest taken as credit will not need to be repaid.

The debt limit indicates the total amount you'll be able to borrow. Debt will need to be repaid either automatically over time, or sooner by manual repayment using the repay function.

The amount you can borrow against your deposit depends on the collateralization ratio of the tokens you deposited. You can borrow anything up to the total amount of debt your assets allow.

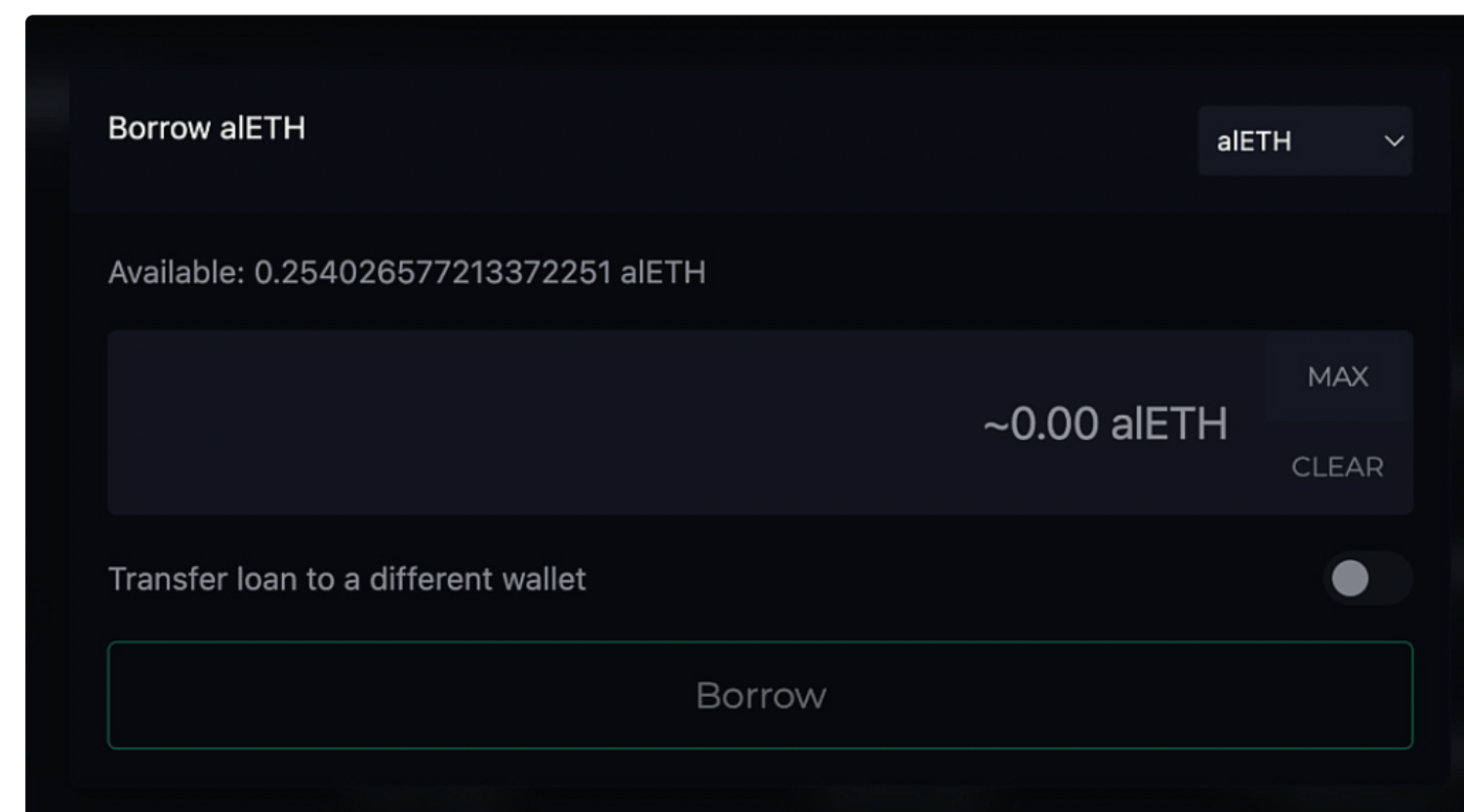
On this page you'll see more detail about how much debt you've currently taken and what the debt limits are per vault.



Debt limit

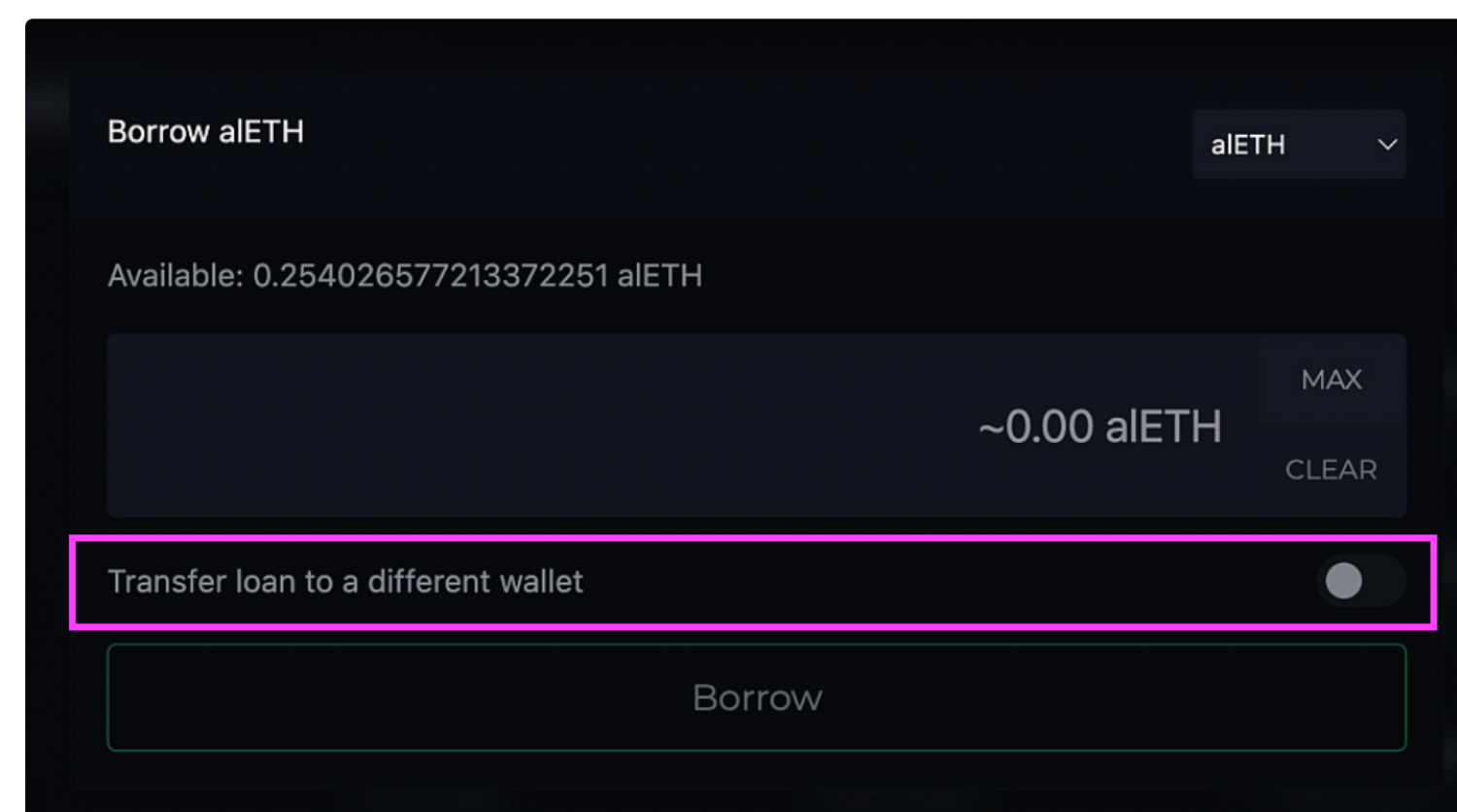
To take your first self-repaying loan, click the 'borrow' button at the top of the page.

The borrow dialogue will open up allowing you to specify which alAssets you have access to borrow. In our case we've only deposited WETH so we only have access to borrow alETH.



The Borrow modal

For our users convenience we've added the ability to transfer the proceeds of the loan to another wallet address. To do this simply flick the toggle and input the recipient's address.



Type in the amount you'd like to borrow, or click the 'Max' button to borrow the maximum amount possible. When you're happy with your loan request click borrow to initiate the transaction.

Follow the prompts in your wallet to authorise the transaction.

Once you've completed the borrow, you'll see your balance update in the wallet, unless you sent your proceeds to another address.

Now you'll be able to see that your debt has increased at the top of the Vaults page.

## alAssets

Now you have your alAssets, you can find out the most efficient ways to use them at <https://alchemix-stats.com/eam>

You can also swap them using one of the suggested links at the top of our swap page - <https://alchemix.fi/swap>, manually using <https://curve.fi>, or using any other supported DEX.

If you have any support queries, please contact our team in the official [Discord channel](#)

Next let's look at how to repay your loan.

# Repay your loan

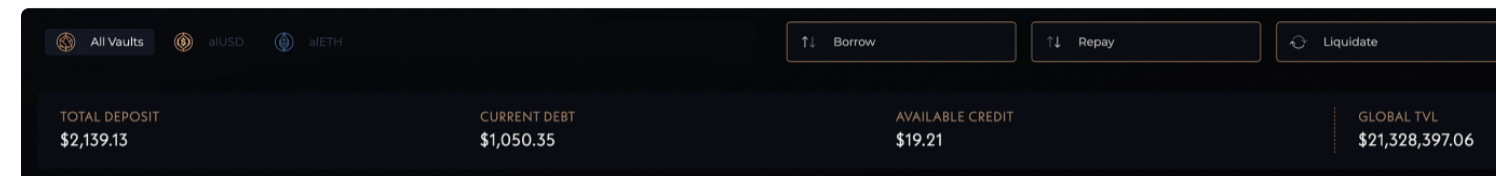


Alchemix loans are fully flexible. Your funds are never locked and there are always options that allow you to release your deposits.

Unlike loans in the traditional banking system, there are no fees or penalties for early repayments. Not only that, Alchemix loans automatically repay themselves over time.

You can also choose to repay your loan manually. Let's look at how to do this.

On the vaults page you can see that this account has a debt of \$1050.35.

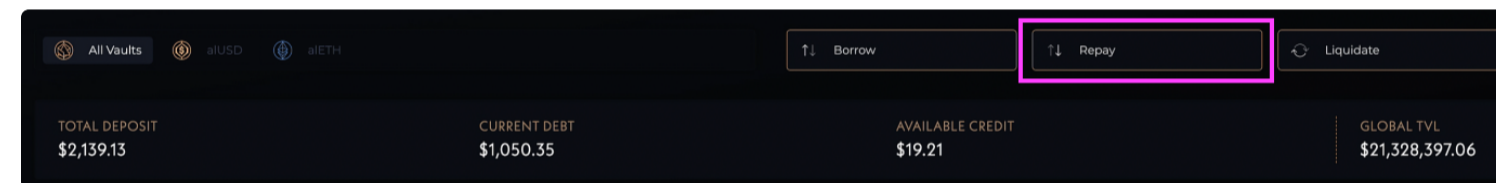


Since both aUSD and aETH have been borrowed, we need to decide which loan to repay. By using the filters we can display the debt owed for each aAsset type.

Let's pay back the aUSD loan. Looking at the active vaults we can see that we've made a deposit in USDC in order to borrow aUSD. Since aUSD is a dollar-pegged asset we can use any acceptable stable coin to repay the loan.

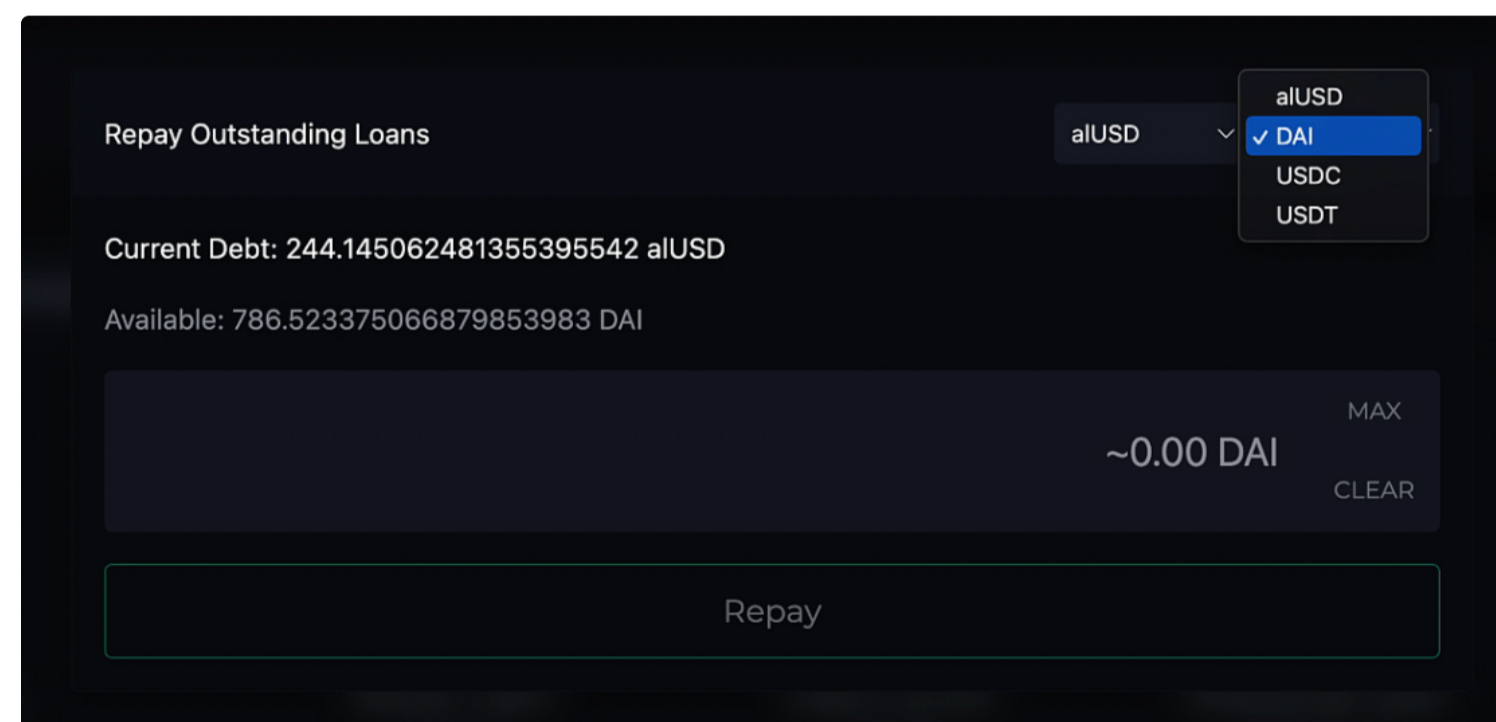
We only have Dai available in our wallet, but since Alchemix accepts Dai as a collateral type, that's no problem.

Click on the repay button which opens the repay dialog.



Repay button

Select aUSD from the first dropdown, then select Dai from the next dropdown. It's worth noting that the aUSD loan can be repaid in aUSD, Dai, USDC or USDT, regardless of which vault originated the loan.



We want to repay all of the aUSD loan so we'll click 'max' in the input field.

Now press 'repay' and complete the transaction in your web3 wallet.

And, we're done.

If you have any support queries, please contact our team in the official [Discord channel](#)

Next let's look at how to repay our aETH loan with our deposited collateral using the liquidate function.

# Liquidate your loan

One of the great advantages Alchemix loans bring is that there is always a way to exit your position, even if you don't have funds in your wallet to make a repayment.

The liquidate function allows you to use your deposited collateral as a source of funds to repay your debt. Alchemix loans are always overcollateralized, meaning there are always enough funds in your account to cover your debt and allow you to exit.

To liquidate the aETH loan on this account we'll open the liquidate dialog and choose the aETH from the first drop down. Since our deposit was made in WETH we'll accept the default WETH in the second drop down.

Liquidate Your Debt

Available: ~0.0 DAI

~0.00 DAI

Maximum Slippage: 0.3% 0.5% 1%

I understand that liquidating will use my deposited collateral to repay the outstanding debt

Liquidate

Select the debt you'd like to pay off

We want to liquidate all of the loan so let's choose 'max' in the input box.

As aETH is pegged to ETH there will be minor discrepancy in their value. To enable users to limit the effect of any slippage you can choose your preferred slippage tolerance here.

Liquidate Your Debt

Available: ~0.0 DAI

~0.00 DAI

Maximum Slippage: 0.3% 0.5% 1%

I understand that liquidating will use my deposited collateral to repay the outstanding debt

Liquidate

Slippage options

Once you're happy with the liquidation parameters press liquidate.

If you have any support queries, please contact our team in the official [Discord channel](#)

Now that we've liquidated our loan, our deposit is available to withdraw so let's look at that in the next video.

# Withdraw funds



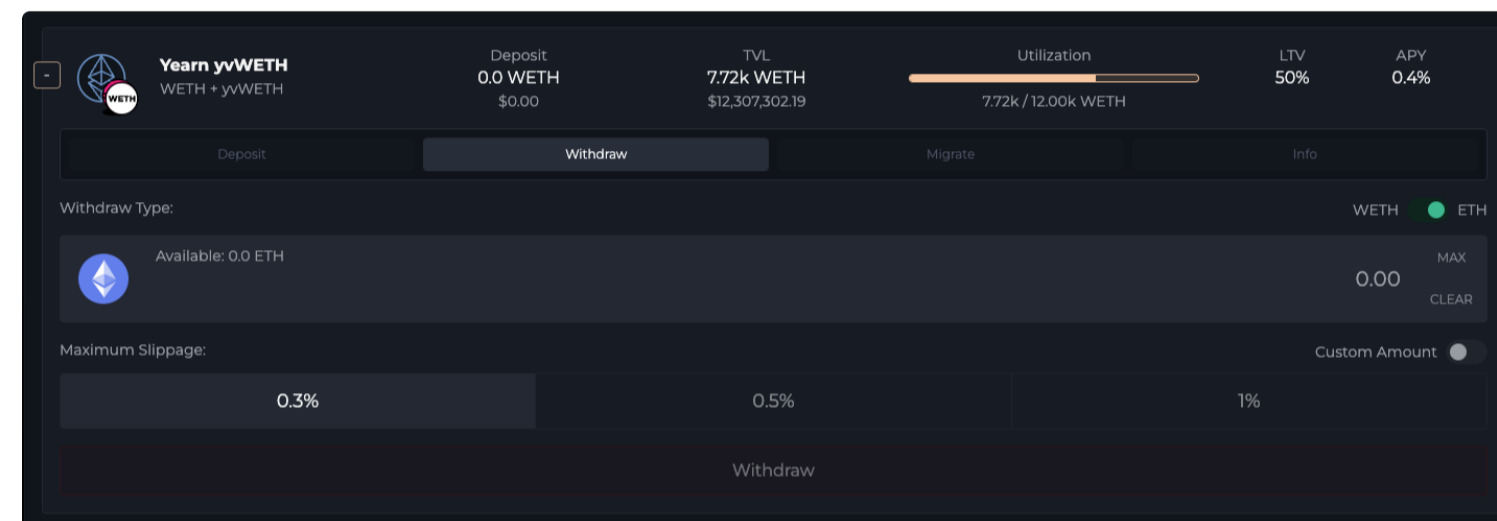
When you want to move your funds back into your wallet you'll need to call the 'withdraw' function.

Something to bear in mind is that you can only withdraw funds that exceed the collateralization requirements of any outstanding loans you have.

If for example you've borrowed 50% of your deposit of a loan that has a 50% loan-to-value ratio then you won't be able to withdraw any funds unless you either repay your debt or liquidate it with your deposit.

In the previous guide, we liquidated our loan which releases the funds to be withdrawn.

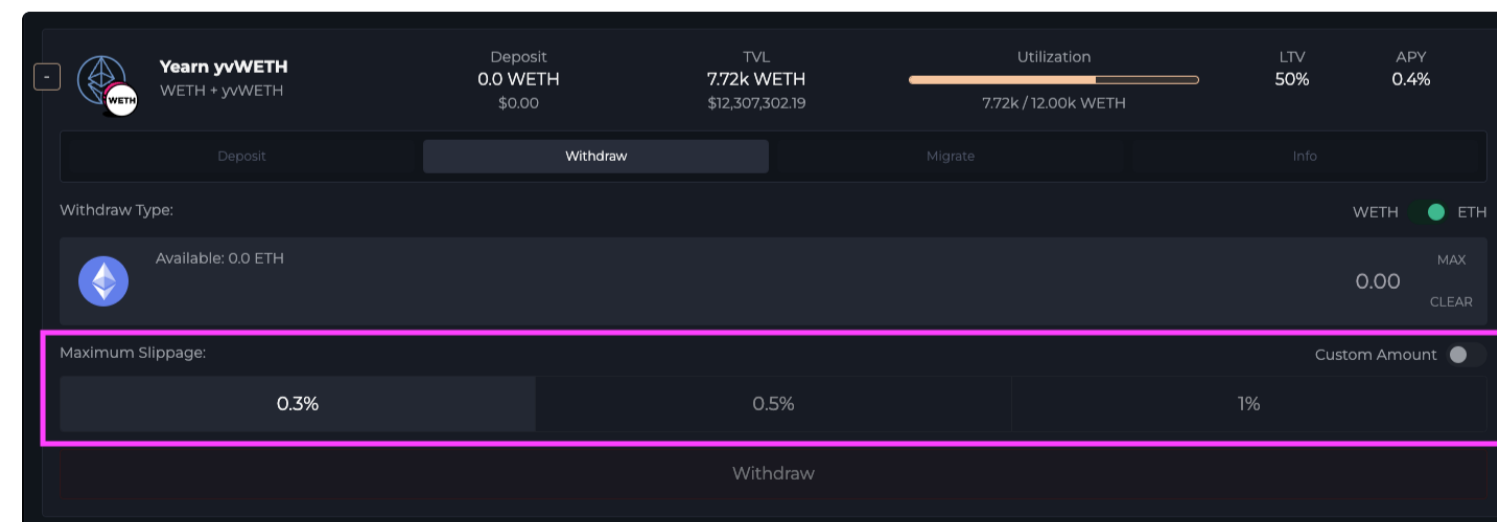
Click the + button next to the vault you want to withdraw from and click the 'Withdraw' tab. You'll notice you've got the choice to receive the standard or yield bearing asset, in this case WETH or yvWETH.



Since we want ETH in our wallet following the withdrawal, we'll toggle the WETH/ETH switch. The system will conveniently withdraw WETH and convert it to ETH for us.

In this case we'll choose 'max' since we want to completely exit the vault.

Like the liquidate and deposit functions, withdraw adds slippage protection control to allow users to limit the slippage as the system converts assets to unwind your position.



Click 'Withdraw' and confirm the transactions in your wallet.

When you've successfully withdrawn you'll see your updated balance reflected in your wallet.

If you have any support queries, please contact our team in the official [Discord channel](#)



# Migrate between vaults

Now that Alchemix supports multiple strategies per collateral type, you can use the Migrate tool to easily move funds between them in the most efficient way. Different strategies offer different APYs which regularly change.

To migrate your funds to a different strategy using the Migration tool, open the position you want to migrate using the '+' button.

Vault Name	Deposit	TVL	Utilization	LTV	APY/APR
Yearn yvDAI	0.0 DAI	6.97m DAI	6.97m / 25.00m DAI	50%	1.3%
Yearn yvUSDC	0.0 USDC	997.55k USDC	997.55k / 15.00m USDC	50%	0.21%
Yearn yvUSDT	0.0 USDT	10.69k USDT	10.69k / 15.00m USDT	50%	0.12%
AAVE aDAI	0.0 DAI	7.91k DAI	7.91k / 5.00m DAI	50%	0.74%
AAVE aUSDC	0.0 USDC	17.65k USDC	17.65k / 5.00m USDC	50%	0.3%
AAVE aUSDT	0.0 USDT	399.45 USDT	399.45 / 5.00m USDT	50%	1.15%
Yearn yvUSDT	0.0 USDT	0.0 USDT	0.00 / 15.00m USDT	50%	0.53%
Yearn yvWETH	0.0 WETH	7.72k WETH	7.72k / 12.00k WETH	50%	0.4%
Lido wstETH	0.0 WETH	1.55k WETH	Vault Full	50%	3.89%
Rocket rETH	0.0 WETH	387.46 WETH	Vault Full	50%	3%
AAVE aWETH	0.0 WETH	1.87k WETH	1.87k / 3.00k WETH	50%	2.98%

Vault overview

From here, select the 'Migrate' tab.

Deposit Withdraw **Migrate** Info

Target Vault: AAVE INTEREST BEARING DAI

Available: 0.0 Shares 0.00 MAX CLEAR

Migrate

Migrate tab

Click the 'Target Vault' dropdown and select the vault you want to migrate your funds into.

Deposit Withdraw Migrate Info

Target Vault: AAVE INTEREST BEARING DAI

Available: 0.0 Shares 0.00 MAX CLEAR

Migrate

Target vault selector

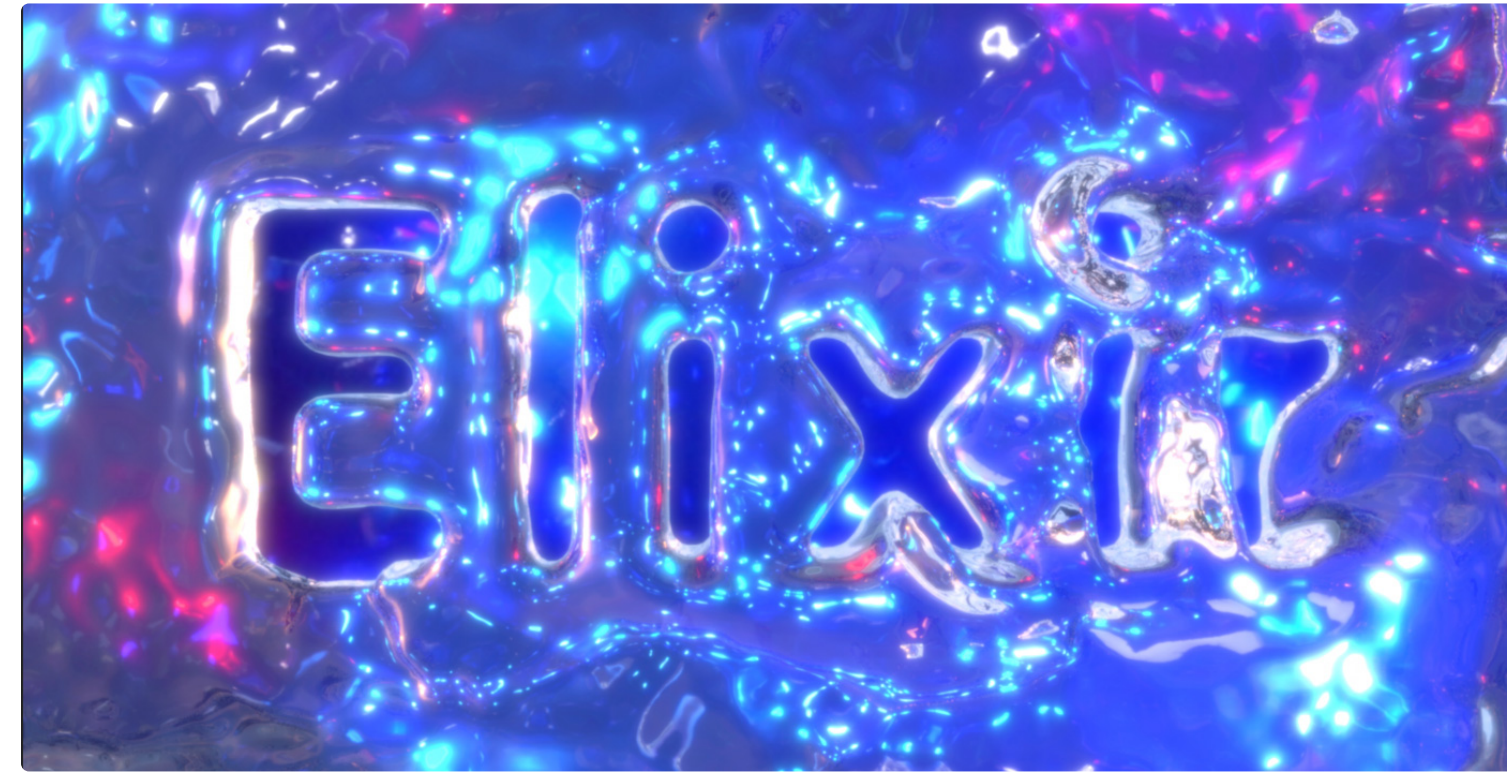
Next, input the amount of funds you'd like to migrate. Click 'All' if you want to migrate everything. Finally, click 'Migrate' and follow the prompts in your wallet. Your funds will now automatically move to your chosen target strategy.

Note: You can also choose to manually unwind and restart your position in another strategy should you choose to.

If you have any support queries, please contact our team in the official [Discord channel](#)

# The AMO: The Elixir

The Alchemix Algorithmic Market Operator



When Alchemix was originally launched, it was never anticipated that the peg stability module, the Transmuter, would build up a significant backstop of funds. To take advantage of this, in the V1 deployment of Alchemix, reserves were deployed in Yearn. The yield was passed from these deposits to DAI and ETH depositors in Alchemix. This enabled us to have a killer feature — boosted yield, which at times, doubled the amount of interest paid to Alchemix depositors.

While building V2, it dawned on the Alchemix team that these DAI and ETH reserves could be more intelligently deployed in order to better benefit the Alchemix ecosystem. Instead of these assets passively making money elsewhere in DeFi, it makes much more sense to use these funds actively in the market to earn the protocol income and to better manage the prices of the aAssets.

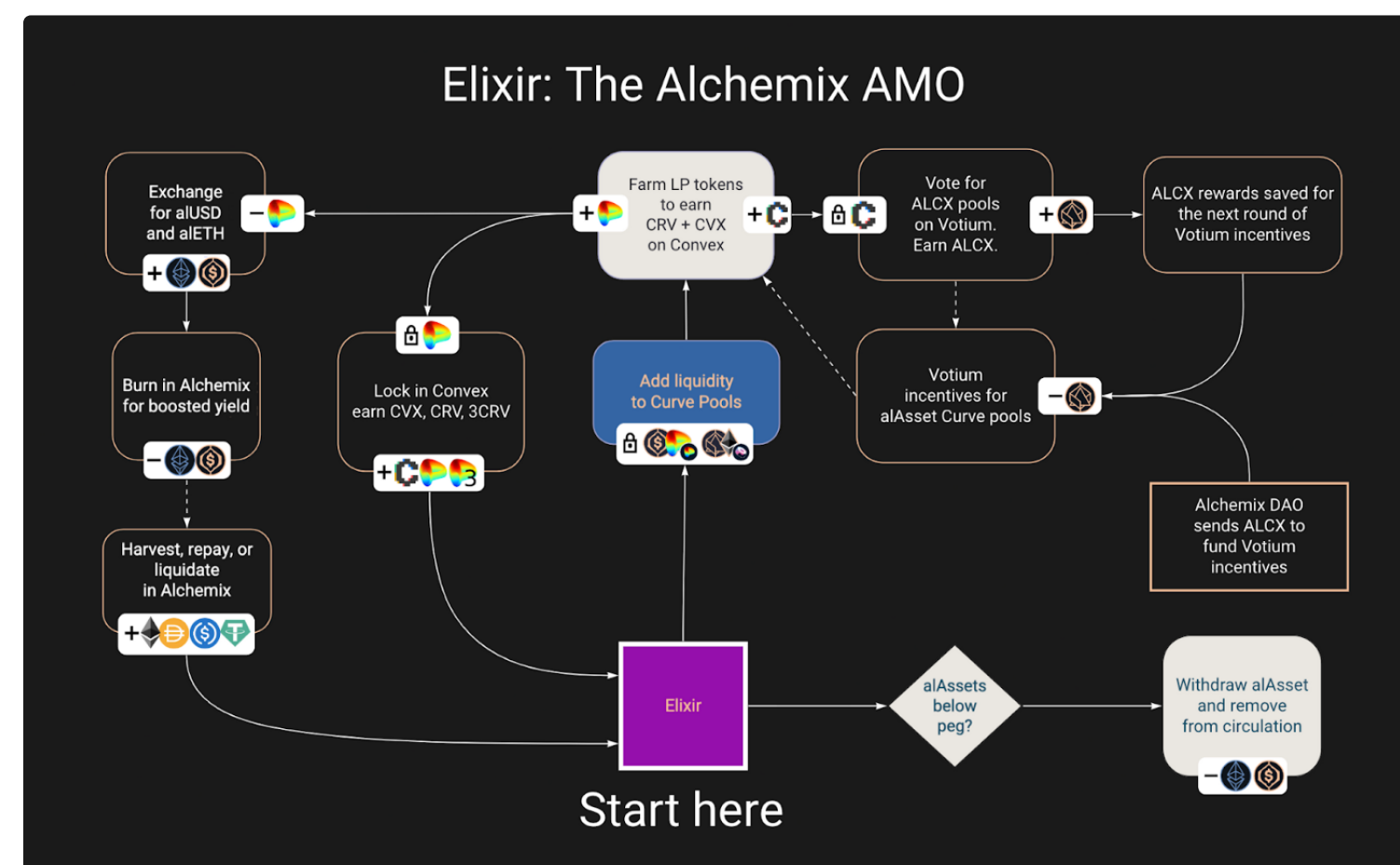
## Introducing the Alchemix Elixir

### *Elixir: The Alchemix AMO*

The Alchemix Elixir is a contract inspired by FRAX's Algorithmic Market Operator (AMO). Their AMO allows them to expand and contract the supply of FRAX in LP pools, with FRAX3CRV LP being the most predominant. They mint and deposit FRAX when the token price is above their peg, and withdraw and burn FRAX when the token price is below their peg. They also farm with the LP in Convex, earning the protocol income in the process.

Through its own automations, the Alchemix Elixir takes a similar approach to market operations, with the exception that **Alchemix cannot mint aUSD into the LP pools** (thus maintaining the overcollateralized nature of aAssets).

See below for a diagram that shows how funds flow through the market.



The Elixir was jump-started by migrating the v1 Transmuter TVL to it. From there, the additional will receive additional funds only when the Transmuters build up surpluses. The Elixir contract will supply liquidity in the primary Curve liquidity pools for aUSD and aETH. By depositing excess DAI, USDC, USDT, and ETH into Elixir, liquidity is deepened and the prices of aAssets are made more stable.

Curve allows for single sided withdrawals and deposits, and bases the exchange price on the relative balance between the tokens in the LP pool. If a pool is overbalanced with aUSD or aETH, it means we are below 1 USD for aUSD and 1 ETH for aETH. Alchemix can increase the price of the aAsset by single-sided withdrawals of aUSD or aETH, thus rebalancing the pool and increasing the aAsset price closer to 1:1. These withdrawn aUSD and aETH tokens would be removed from circulation, with the potential to be redeployed to their Curve pool should the price of the aAsset increase to the point where it is able to support the addition of aUSD with a negligible effect on the price of the asset.

The next function of Elixir is to generate revenue and build long-term liquidity for the protocol. The Elixir will do this by making a liquidity-driving asset accumulation strategy. Liquidity-driving assets, such as CVX and CRV, gives the DAO power to direct rewards from the corresponding protocols. The more liquidity driving assets are owned, the more Alchemix can sustainably incentivize the primary aAsset liquidity pools. Alchemix also typically bribes voters to vote for emissions to these pools - where typically every \$1 input results in greater than \$1 emitted to the liquidity pool. Given the Elixir tends to own a significant share of the liquidity pools, this can result in a significant amount of value returned to the DAO in the form of CRV, CVX, and other assets. The more CVX Alchemix votes with, the more ALCX is returned as a rebate for voting for the aAsset pools. So between this multiple and the Yotium rebates, it greatly enhances the efficiency and longevity of ALCX emissions.

*The ancient tomes of alchemy describe a mysterious fluid known as "Elixir". It was thought to have the power to turn base metals into gold and even grant immortality.*

When Alchemix stakes its own Curve LPs on Convex, it receives CRV and CVX rewards. The Elixir is able to use these rewards to benefit the DAO, with the current approach explained [here](#).

The Elixir is a significant upgrade to our peg stability module. The concentrated management of our protocol-controlled value aligns it more closely to our interests.

The ancient tomes of alchemy describe a mysterious fluid known as "Elixir". It was thought to have the power to turn base metals into gold and even grant immortality. In that sense, the Alchemix Elixir is true to its name, with the peg-stability mechanisms and CVX flywheel bringing long-term price stability and sustainability to Alchemix aUSD and aETH. It's a new era for Alchemix, and we're happy to be bringing magic to DeFi yet again.

## Contracts:

aUSD Elixir: **0x9735f7d3ea56b454b24ffd74c58e9bd85cfad31b**

aETH Elixir: **0xe761bf731A06fE8259FeE05897B2687D56933110**

# Bridging assets to other chains

## Introduction to bridging

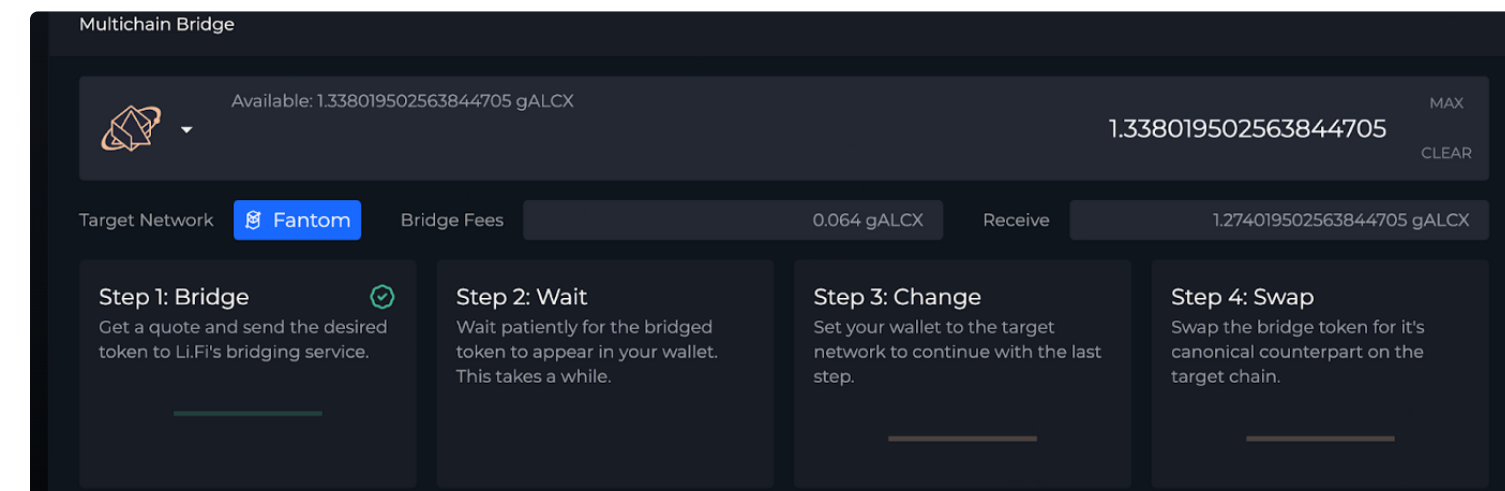
Users can send assets to wallets on other networks directly from a centralised exchange like Binance, or users can use a cross-chain bridge to perform the bridging transaction manually on-chain.

To send tokens from a centralised exchange, withdraw your supported token, ensuring that you specify the correct wallet address **and** blockchain.

To perform the bridge manually visit your favorite cross-chain bridge. You could use <https://zapper.fi/bridge>, <https://synapseprotocol.com/> or any other bridge you trust.

## Bridging with Alchemix

You can also specifically bridge aAssets and gALCX easily on Alchemix.fi. First, navigate to the bridge page. In the 'From' field, input the amount and select the token from the dropdown list. In the 'To' field, select your target network from the dropdown. The numeric field will display how many tokens you'll receive when the funds arrive on your target network.



The screenshot shows the 'Multichain Bridge' interface. At the top, it displays 'Available: 1.338019502563844705 gALCX' and a 'MAX' button. Below this, there's a 'Target Network' dropdown set to 'Fantom', 'Bridge Fees' of '0.064 gALCX', and a 'Receive' field showing '1.274019502563844705 gALCX'. The interface is divided into four steps: 'Step 1: Bridge' (Get a quote and send the desired token to Li.Fi's bridging service.), 'Step 2: Wait' (Wait patiently for the bridged token to appear in your wallet. This takes a while.), 'Step 3: Change' (Set your wallet to the target network to continue with the last step.), and 'Step 4: Swap' (Swap the bridge token for its canonical counterpart on the target chain.).

Next you'll need to approve the asset of your choice, then press the swap button to send the funds to the bridge.

The bridge transaction should take between 10-30 mins to complete. After that the funds will appear in your target wallet.

Lastly, you will need to swap your bridge tokens for the canonical token using step #4. Until you complete this task you will not be able to use your swapped tokens.

Now that you have your assets on the target network you can deposit, earn and take self-repaying loans on the new chain.



# Risk and Counterparties

A guide to the various counterparties that make up the Alchemix system and the risk each counterparty takes on.

The purpose of this article is to go over the risks that various types of users of the Alchemix system take on when the system is functioning as intended. This list is not intended to be exhaustive, as certain risks are inherent to DeFi and Crypto in general, such as smart contract risk. For information about other types of risks, see [🔗 Multisig Admin Rights](#), [🔗 Audits](#), and [🔗 Vault Losses and Collateral De-pegging](#).

## Depositors (Borrowers)

Depositors provide collateral to the yield strategies in the Alchemists in order to take aAsset loans. Unless a depositor also wishes to act as a liquidity provider, they will typically swap their aAsset to another asset soon after taking the loan. Therefore, they are not exposed to the price of aAssets over time.

The primary risk a depositor takes on is risk of having funds deposited in the underlying yield strategy, through Alchemix. If the strategy they deposit to experiences a loss that exceeds the `maximumLoss` (a variable set by governance), then the yield strategy will be paused. This means users may not make any deposits, may not liquidate or repay, and may not take a new loan with this strategy. Harvests will also be disabled. Lastly, users will be unable to withdraw collateral in the form of the underlying asset. Users will still be able to repay their loan and withdraw the yield token, however. For example if the `maximumLoss` was exceeded, a user could not withdraw DAI from a strategy that uses yDAI, but they would still be able to repay their loan with DAI to withdraw their yDAI collateral).

In the scenario of an underlying strategy suffering a majority loss of funds (ie, greater than 50% of the strategy), then the user would actually have bad debt with Alchemix (the value of their debt would exceed the value of their collateral). In this scenario, the user actually suffered less of a loss by using Alchemix.

Note some yield strategies may require selling yielded tokens to harvest yield. In this scenario, a temporary depeg of the value of the harvested token would result in the user experiencing reduced yield for the period of time the token remains depegged.

For more an example of the protocol handles collateral depeg events and vault losses, see [🔗 Vault Losses and Collateral De-pegging](#).

## Liquidity Providers / Transmuter Users

aAsset liquidity providers are exposed to price fluctuations of aAssets. They create the liquidity for users to sell their aAssets for other tokens. aAssets can only be redeemed for underlying collateral in three ways:

1. Loan repayment (instant, 1 aAsset = 1 asset)
2. Selling through a liquidity pool (instant, price will fluctuate)
3. Depositing in the transmuter (timeline is uncertain, 1 aAsset = 1 asset)

A liquidity provider / aUSD holder that does not have an Alchemix position in a yield strategy does not have option 1 at their disposal. A liquidity provider has three primary steps to consider when providing liquidity:

1. Price / balance of liquidity pool when entering the pool
2. Yield earned from providing liquidity over the life of the liquidity provision
3. Price / balance of liquidity pool when leaving the pool

If the balance of the pool moves favorably for the LPer over time, they can earn yield as well as a net positive slippage from the difference between their exit and entry position. If the balance of the pool moves unfavorably, then the net negative slippage would be subtracted from the yield earned during their liquidity provision over time.

A user can hedge this exposure by using the transmuter, or by being a depositor within Alchemix. If the aAsset pool shifts less favorably for the depositor/LPer, they can withdraw aUSD instead of stablecoins for a bonus positive slippage and repay their debt. They could also use the same approach with the transmuter, at the opportunity cost of waiting for the collateral to flow into the transmuter. See [🔗 Transmuter](#) and [🔗 The Transmuter, Elaborated](#) for more information on how the transmuter distributes collateral to aAsset stakers.

Lastly, aAssets could become undercollateralized if a large enough loss of funds of an underlying yield strategy occurred, as detailed in the [# Depositors \(Borrowers\)](#) section.

## Alchemix DAO and ALCX Holders

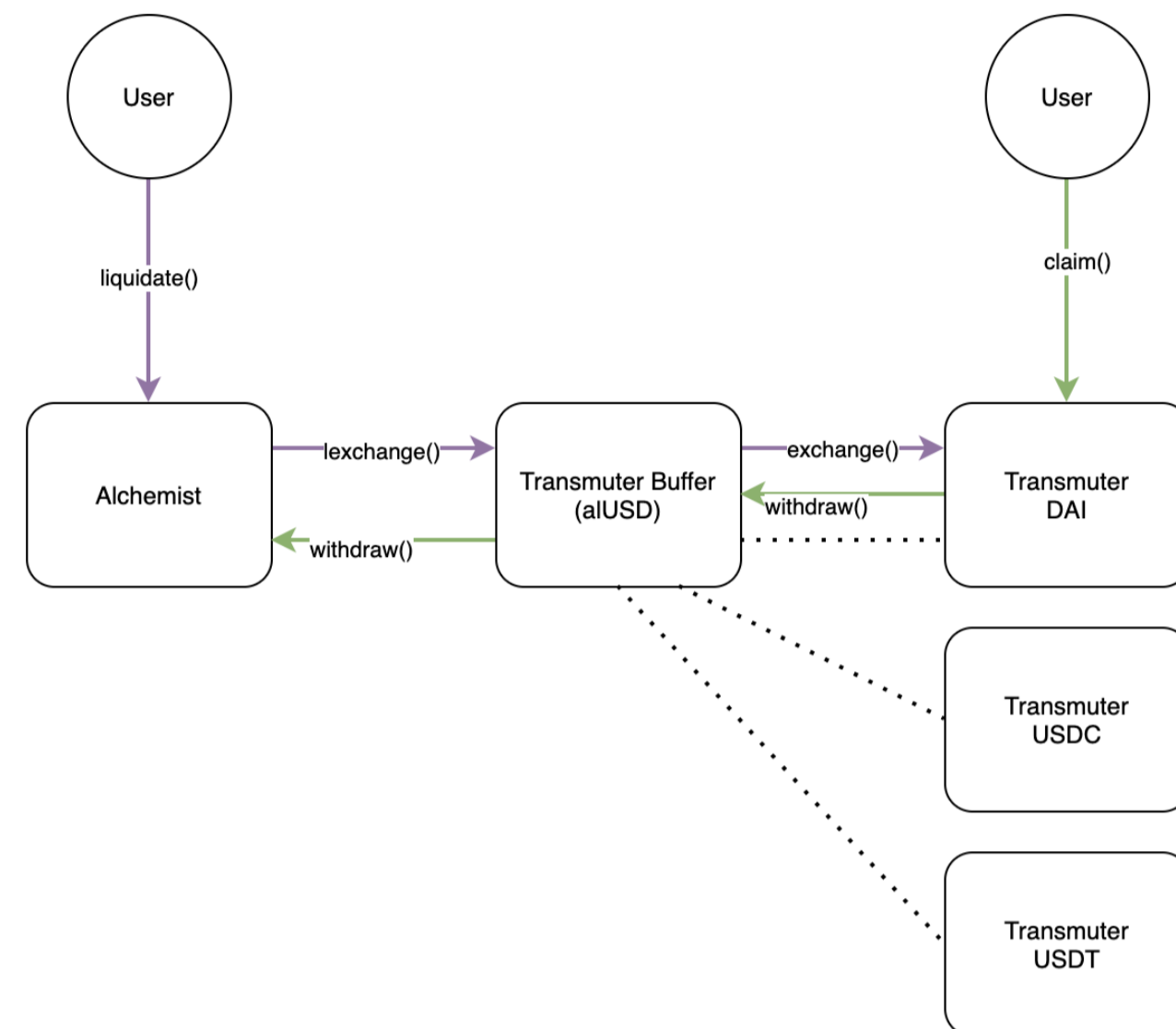
As mentioned above, it is possible for bad debt to exist in Alchemix if a yield strategy suffers a significant loss of funds. Because ALCX liquidity and single staking pools are not locked, Alchemix cannot currently slash stakers to make the protocol whole. In the scenario of a full yield strategy loss above, the treasury could sell ALCX or other assets from the treasury to make the protocol whole if decided by governance, which would dilute ALCX holders.

# The Transmuter, Elaborated

A deeper dive into how the Transmuter functions

The Transmuter is actually composed of two separate components: The TransmuterBuffer and the Transmuter.

When funds enter the Transmuter, assuming there is at least a matching amount of the corresponding aAsset they will be immediately be claimable by users with aAssets staked in the transmuter. The TransmuterBuffer sits between the Alchemist and Transmuter, limiting the available funds that are accessible for transmutation. The goal here is to delay the transmutability of funds so that the massive front-stop (ie, excess funds in the Transmuter buffer) cannot immediately be used to take advantage of extremely small (< 0.1%) arbitrage opportunities, thus burning protocol value for tiny gains. The longer the system can hold onto the front-stop, the longer it can supply liquidity and earn revenue through the Elixir/AMO, and the more Alchemist depositors Alchemix can sustain.



Transmuter Flowchart

In the Transmuter, user exchanged and un-exchanged balances are updated in a step-wise manner, only when the `exchange()` function is called. The `exchange()` function sends the underlying asset (USDC, DAI, or USDT) to the transmuter in exchange for the aUSD burned from the transmuter. The TransmuterBuffer receives a call to its `exchange()` function whenever `alchemist.harvest()`, `alchemist.liquidate()`, or `alchemist.repay()` are called - ie, whenever a yield harvest occurs, or when a user liquidates or repays their loan. `TransmuterBuffer.exchange()` will update the available amount of flow that is theoretically accessible by the transmuter, and subsequently call `Transmuter.exchange()` with the marginal amount of funds that need to be exchanged. Each Transmuter handles a single collateral type. Each TransmuterBuffer handles a single synthetic type, and all collateral types underlying that synthetic.

## Flow Rate

The flow rate is set by governance and is a per-second MAXIMUM rate of flow for funds to be sent from the TransmuterBuffer to the Transmuter. The main features of the flow are:

1. Flow-rate is constant and linear.
2. The flow-rate (measured in underlying collateral token per one second, ie 1 DAI/second) will continuously add to the available-flow (measured in the underlying collateral token, ie 1 DAI).
3. The available-flow is a measure of how much underlying collateral will immediately flow from the TransmuterBuffer to the Transmuter, upon a deposit to the TransmuterBuffer (ie, a call of the `exchange()` function). This means the available-flow can build up over time if the transmuter flow-rate is being underutilized. A build-up of available flow makes it possible for the effective flow-rate over a period of time to exceed the flow-rate, thus ensuring the set flow-rate is acting as more of an average over time, rather than a hard cap.
4. Each underlying-token has its own flow-rate. The available-flow for a given underlying-token can exceed the total amount of funds denominated in that underlying-token (across all strategies) held by the transmuter-buffer in the Alchemist. However, when this is the case, the Transmuter will only be able to access the actual funds held by the transmuter-buffer in the the Alchemist (see Invariants 1 and 2)
  1. Figure 1 shows a scenario where available flow has exceeded the total buffered amount (total amount of underlying token controlled by the transmuter buffer across all strategies in the alchemist).
  2. Figure 2 shows a scenario where total buffered amount has exceed the available flow.
  3. In both scenarios, the total amount exchanged to the transmuter cannot surpass the lesser of the two values in question.
  4. In Figure 1, there will be an excess of available-flow. Should the flow of the underlying asset to the transmuter increase beyond the defined flow-rate, the excess of available-flow would be used to absorb the faster rate (as described in Item 2 above).

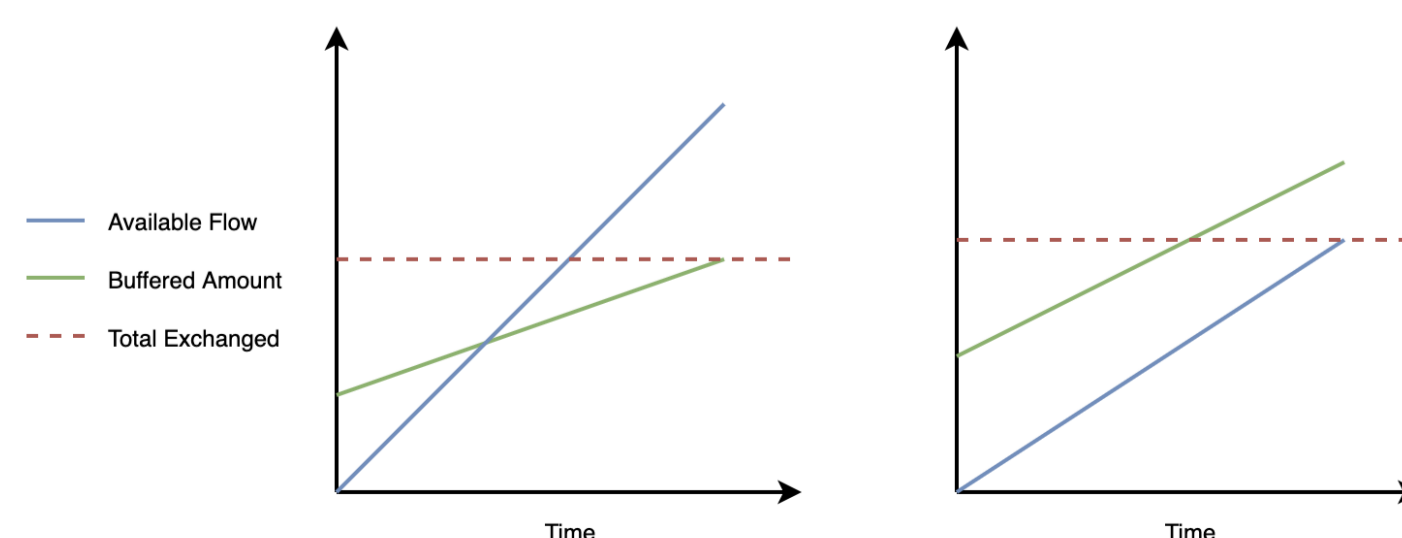


Figure 1

Figure 2

# Vault Losses and Collateral De-pegging

Vault losses and collateral de-pegging events are two scenarios that can create a loss of aAsset backing, thus jeopardizing the health of the protocol. Vault losses are caused by the underlying yield strategy returning less of the underlying token than expected - for example, a strategy that is meant to earn 10% APR on ETH suddenly only being worth 0.9 ETH per 1 ETH deposited. Collateral de-pegging is caused by the underlying collateral being worth less than its expected value. This is only applicable to aUSD as it is the only alchemist that accepts multiple collateral types. For example, if USDT were to be worth \$0.9 relative to DAI and USDC each being worth \$1. This is not relevant to ETH as 1 ETH will always be worth 1 ETH.

Vault losses are handled automatically through the `maxLoss()` parameter. Collateral de-pegging is handled manually through sentinels' ability to pause tokens (see [Multisig Admin Rights](#)). See below for an example scenario where a yield-bearing asset experiences a 10% loss in underlying collateral, and another scenario where DAI drops to \$0.80 relative to USDC and USDT.

## Vault Loss Scenario

**Scenario:** A yield-bearing asset experiences a loss in the underlying collateral. For this example, we will assume a 10% loss of DAI from the yvDAI vault that is unrecoverable.

After the transaction that causes the loss is committed to the chain, the `maxLoss()` is triggered and the following happens:

- The following yvDAI Alchemist functions are automatically disabled:
  - `deposit()`
  - `depositUnderlying()`
  - `withdrawUnderlying()`
  - `withdrawUnderlyingFrom()`
  - `liquidate()`
  - `harvest()`
- The following yvDAI Alchemist functions are still useable:
  - `withdraw()`
  - `withdrawFrom()`
  - `repay()`
  - `mint()`
  - `burn()`

**Resolution:** In order to re-enable the disabled functions, the following needs to happen:

1. A proposal is created to call `snap()` on the Alchemist, targeting the yvDAI vault.
2. A vote takes place (we assume it passes)
3. `snap()` is called on the Alchemist, which accepts the 10% loss and resets the expected value of those yield tokens held in the Alchemist

**Damage:** The maximum damage is the total amount of funds lost from the vault. aUSD will still be overcollateralized and depositors will experience the loss, the same way they would experience the loss if they held the tokens outside of Alchemix or if they used `withdraw()` after `maxLoss()` was triggered. The effective rate of yield flow to the transmuter buffer would also be slightly slower, as there would be slightly less collateral in the system earning yield relative to the aAsset supply. Note that if a loss of >50% were realized, this could lead to a loss in backing for aUSD.

While the risk of a vault losing collateral is low, the damage is still significant. However, Alchemix does not control the operations of 3rd party vaults, so the only way to minimize the risk is to carefully consider which yield-bearing strategies are added.

**Necessary Response Time:** The response time for this scenario does not need to be necessarily fast, because the functions that are affected by a vault loss will be automatically disabled. The team and the DAO should assess the loss to make sure that it is unrecoverable before taking the governance steps to remedy the situation by calling `snap()`.

## Collateral De-pegging

**Scenario:** One collateral token used by the Alchemist experiences a severe de-pegging against other collaterals. For this example, we will assume DAI drops to 80 cents vs USDC & USDT. aUSD maintains its peg against USDC & USDT.

This de-pegging presents multiple arbitrage opportunities:

1. Users can buy DAI off the market, deposit it into the Alchemist, take a loan, and repeat this loop until the minting cap is reached.
2. Users can buy DAI off the market and use it to repay their loans until the repay cap is reached.
3. Users can liquidate their current yDAI position (paying off their outstanding debt at a discount), buy more DAI with their loan, deposit it into the Alchemist, take a loan, and repeat until the liquidation cap is reached.

These arbitrage opportunities will likely result in one or more of the mint / repay / liquidate caps being met.

**Resolution:** The only resolution that matters is getting DAI to reach peg again. This can take multiple avenues.

- The peg could re-stabilize on its own without any intervention.
- The collateral in question can be disabled by a sentinel or admin, buying the peg more time to re-stabilize. This would disable the following functions:
  - `deposit()`
  - `depositUnderlying()`
  - `repay()`
  - `liquidate()`

**Damage:** The de-pegging of DAI results in the price of the aAsset dropping towards the de-pegged asset. The sentinels exist to disable underlying tokens as soon as they experience a de-pegging event. The repay, liquidate, and mint caps are in place to limit the amount of de-pegging of the synthetic asset that can occur prior to sentinel action.

**Risk:** Given the interconnected nature of underlying collateral tokens and DeFi at large, there will likely be consistent, small arbitrage opportunities between collaterals and their pegged synthetics. In times of high volatility, these arbitrage opportunities can get exasperated as assets experience larger and longer de-pegging events. As a result, there is some risk that the repay/liquidate/mint caps get reached.

Note that if the peg does not restabilize, DAI would remain paused there will be a loss in the backing of aUSD. The DAO will need to determine how to proceed in this scenario.

**Necessary Response Time:** The faster a sentinel can respond by disabling the de-pegged underlying token, the less the price of aAsset will be arbitrated down.