

# Arcadia Finance [Draft]

November 2023

Thomas Smets  
Arcadia Finance  
Brussels, Belgium  
thomas@arcadia.finance

## ABSTRACT

(To Do) Arcadia Finance introduces a novel on-chain architecture for collateral management of margined positions.

The overarching vision of Arcadia Finance is to build standardized on-chain infrastructure for collateral management, unlocking the capital in diverse DeFi assets, and establishing a comprehensive platform for managing collateralized positions.

As we anticipate the continued growth of on-chain tokenized assets and financial primitives, the importance of standardized and efficient collateral management becomes increasingly paramount. Without it, the full potential of these assets and financial primitives remains unrealized, hindered by protocol inflexibility and fragmentation.

Arcadia strives to enable higher capital efficiency for users, transparent risk assessment, faster go-to-market time for new protocols without compromising on security, self-custody and decentralisation.

## KEYWORDS

Collateral management, DeFi, EVM

## 1 INTRODUCTION (TO DO)

This paper will explain the rationale why we are building the Arcadia Finance Technology Stack, and go over its technical implementation.

The Arcadia Financial Technology Stack is a set of on-chain, inter-linked financial protocols, centered around collateral management. They facilitate non-trusting peers to close financial contracts without the need of intermediaries.

It consists out of three protocols, each with distinct responsibilities and permissions:

- The Arcadia Registry.
- The Arcadia Accounts.
- The Generalised Creditor.

## 2 TERMINOLOGY

**DEBTORS AND CREDITORS:** We will often use the terms Debtors and Creditors throughout this paper, our definition is broader than the commonly accepted context within traditional debt arrangements. We use the term Debtor for any holder of a financial instrument that results in liabilities on the balance sheet (borrowers, option writers, payers of cash-settled futures contract or swaps...). While the Creditor refers to the entity to whom the liability is owed.

**OPEN POSITION:** The size of the liability that a specific Debtor has with a specific Creditors.

**COLLATERAL:** The assets pledged by a Debtor to a Creditor to cover the credit risk in case the Debtor would default.

**MARGIN:** The value of collateral assets that a Debtor must hold in a Margin Account to cover the credit risk of its Creditor(s). The margin requirements, are set by the Creditor. In general there are two types of margin requirements: Initial Margin (minimum value of collateral that must be held in the margin account to open a new position) and Maintenance Margin (minimum value of collateral that must be held in the margin account to keep an existing position open).

**MARGIN ACCOUNT:** Our definition of a Margin Account extends beyond the conventional usage within Brokerage Accounts. The holder of a Margin Account not only has the ability to utilize and transact with the assets within the account (as is the case with a cash account or your standard crypto wallet), but can also incur liabilities against those assets. The margin requirements (initial and maintenance margin) are set by the Creditor.

## 3 HISTORY OF COLLATERAL MANAGEMENT

Withing the financial system, Collateral serves a primary function in mitigating the counterparty risks borne by Creditors, in case Debtors would fail to fulfill their financial liabilities.

### 3.1 Origin

Evidence of collateralized loans, also known as secured loans, can be traced back to at least 400 BCE in Ancient Greece [7]. In these early transactions, borrowers provided tangible assets as collateral to secure loans, laying the foundation for the concept of collateralization.

### 3.2 Current State

Over the past decades we have witnessed significant advancements in collateral management practices.

With the rise of complex financial instruments and the globalization of markets, the role of collateral has expanded beyond traditional lending. Collateral is now used to secure a wide range of financial instruments, including derivatives trading, margin accounts, and other sophisticated financial contracts.

The digitalisation of the financial system greatly improved the efficiency of the exchange and settlement of collateral. Reporting, margining and reconciliation processes could be executed on a daily basis instead of on a weekly or even monthly basis [9].

The emergence of blockchain technology and decentralized finance (DeFi) led to a new wave of innovation for collateral management. Blockchain has some excellent properties with regards to collateral management:

- With its immutable and transparent ledger, collateral can be audited 24/7.
- Smart contracts enforce margin requirements real time on a 24/7 continuous basis.
- The atomic execution of transactions avoid expensive reconciliation processes and eliminates settlement risk.
- The atomic execution also enables optimistic execution of transactions (e.g. flash loans) which can greatly improve processes such as refinancing loans or even enable completely new financial use-cases.
- The permissionless nature of blockchain and Dapps (Decentralised Applications), combined with a shared state between all those applications, make different financial applications and assets composable and inter-operable by default.

It is no coincidence that some of the first blockchain applications with product market fit, were secured lending protocols. These early pioneers like Maker[10], Aave[11] or Compound[6] rely on collateralization as the only way to mitigate counter party risks and work with over-collateralized loans. As such they enable peer-to-peer or even peer-to-contract loans, without the need of any trusted intermediaries.

### 3.3 Collateral management in crisis

Collateralization, both in the traditional financial world and in DeFi, is not without its problems. Recent crises in both industries highlight the need for better collateral management infrastructure.

Bad collateral management, rehypothecation of collateral, opaque accounting practices of collateral, and outright fraud with collateralized assets, are all cited as root causes[4] behind the 2008 financial crisis. Following the crisis, a variety of major regulations were introduced (e.g. EMIR in the European Union, or Dodd-Frank in the USA). While these packages were successful in stabilizing the financial system, they have clear centralization tendencies[3]. The major beneficiaries of said regulations are the major established institutions, mandatory intermediaries for central clearing, central banks and the regulatory bodies themselves. This might paradoxically cement even greater systemic risks into the financial system.

In 2022 DeFi experienced its own financial crisis. In a few months time, many of the ecosystem's key (albeit mostly centralised) players collapsed, and removed all liquid with them. Again bad collateral management, rehypothecation of collateral, opaque protocol mechanisms, and outright fraud with collateralized assets were at the root of the problem.

A recurring trend with protocols that imploded in 2021 and 2022 is that the lack of collateralization was obfuscated behind complex protocol designs and narratives. Some of the notorious examples are:

- Synthetic stable-tokens such as Gaia-USDF, Iron-Titan and Luna-Terra.
- Olympus and its forks relying on the (3,3) model.

- Protocols where the only utility of a token is to receive more of the token itself (animal yield farms, reflective tokens...).
- Unsecured loans by market makers and CeFi players.

DeFi protocols should focus more on collateralization, and less on Complex tokenomics that obfuscate who ends up paying when things go bad.

### 3.4 The Future?

The over-collateralized DeFi protocols however, operated remarkably well during the aforementioned crisis, even better than their traditional counterparts. This is well illustrated by some of the collapsed centralised entities, active in the space until 2022, that had both on- and off-chain liabilities. All on-chain Creditors were repaid in a timely and orderly fashion, as enforced by their smart contracts. While the lawsuits for the many off-chain Creditors are still ongoing (and probably will be for many years).

It is for these types of crises that liabilities are secured in the first place. These events further strengthened our vision that blockchain-based protocols have the potential to become the dominant infrastructure for collateral management. But before we get there, a number of Inefficiencies of current DeFi protocols must be overcome, as we will discuss in Section 4.2.

## 4 COLLATERAL MANAGEMENT IN DEFI

### 4.1 General Principles

### 4.2 Inefficiencies

In the next sections we will highlight a number of inefficiencies, which need to be overcome before DeFi can be broadly adopted as the go-to infrastructure for collateral management.

**4.2.1 Hacks and exploits.** The major problem in DeFi today is the number of hacks and exploits, the estimated amount of funds lost in 2023 ranges from \$400M to \$1B. There are many security related practices the sector as a whole should improve on, but we want to highlight one problem.

Almost all DeFi protocols use over-collateralization in some way of form to manage counterparty risks between different user-groups. Lending protocols, perpetual protocol, Prime brokerage protocols, option protocols etc. all require some or all users to deposit collateral. While the nature of the financial contracts may be very different for each of these protocols, they share a great amount of common logic:

- Pricing of collateralized assets.
- Management (depositing, withdrawing...) of collateralized assets.
- asset-liability specific risk parameters and calculations.
- Margin calls and liquidations of risky positions.
- Settling bad debt.

Pricing of assets, management of assets and liquidation logic is complex, error prone and most mistakes easily lead to severe user losses. Today all protocols implement these redundantly, even for new versions of the same protocol. Not only is this very costly in development time, the core logic is rewritten and re-deployed time and time again, and with each new deployment, new bugs can be introduced.

Being able to build on top of battle tested code would benefit smart contract developers, protocols, users and the overall ecosystem.

**4.2.2 Fragmented and isolated collateral.** The average DeFi-user has collateralized assets fragmented and isolated across a multitude of protocols (do we have any numbers on this?), and many quality assets are sitting idle. While isolated margin positions have their benefits (they isolate risks), they are not capital efficient. Having the ability to use a portfolio of assets as collateral improves the capital efficiency for a number of reasons:

- The volatility of a portfolio of assets is always equal to, or lower than the weighted volatility of the individual assets. Or put different, assets losing in value can be compensated with assets increasing in value.
- Having less positions, with lower volatility, reduces the number of transactions.
- Depending on the correlation between collateralized assets, the Creditor might use less strict margin requirement.
- Debtors can use negatively correlated assets to hedge positions and limit liquidation risks.

Having a global shared state across applications and assets is often cited as one of the key advantages of blockchain over centralised financial infrastructures[8]. This should open up a whole new solution space for Debtors to manage collateralized positions, and might even enable a single Debtor to share its margin between non-trusting Creditors without intermediaries. Hence it is quite ironic that traditional brokers and clearing houses offer more advanced capabilities for cross- and portfolio-margined positions, compared to the state-of-the-art DeFi protocol.

Again there are again multiple underlying root causes why collateral in DeFi is still fragmented and under-utilized:

- As mentioned in Section 4.2.1, there is no shared collateral management layer, each protocol with collateral has their own non-standardized implementation.
- Protocols are build around specific assets types (eg. lending protocols for simple ERC20 or for AMM LPs, or for NFTs). Different asset types cannot be used within the same protocol to back a single position and emergence of new primitives/token standards requires migrations/new protocols.
- Blockchain is still a young technology, core non-financial infrastructure required for on-chain portfolio management (think Account Abstraction, intents or cross-chain messaging) is only recently developed.

Fragmentation of assets not only results in capital inefficiencies, it also contributes to a challenging user experience, more on that in the next Section (4.2.3).

**4.2.3 End user complexity.** End-user do not "want" to manage collateral, they want to optimize portfolio's to achieve a certain objective (e.g. increase yield, delta hedging, diversify exposure to different protocols...). Collateral management is a means to an end, it is not an activity most users enjoy doing.

End-user adoption will only increase outside of a niche bubble of tech enthusiasts if the technical complexities around collateral management are be abstracted away. Important note, abstracting away technical complexities should never come at a cost of hiding

risks or relying on centralised and custodial solutions (as is to often the case).

As mentioned in the previous Section 4.2.2, the bad user experience is partly due to the fragmentation of assets and positions. Rebalancing portfolio's often require multiple transactions per asset and per position. Today the user has to execute multiple sequential transactions per asset, instead a single transaction that rebalances the complete portfolio. Let's take as example a user that wants to earn yield from his collateralized assets and get exposure to Liquid Staking Tokens (LSTs). Since LST protocols introduce an additional layer of risk, he wants to diversify risks over 5 different LST service providers. Assuming our user has wrapped Ethereum, he would need to do 10 transactions via multiple platforms (5 approvals and 5 swaps or 5 approvals and 5 staking actions). Rebalancing said portfolio, or depositing it as collateral, would require another 5 to 10 transactions. Not only is this time consuming, it also introduces additional operational risks, to name a few:

- Herstatt risk (settlement risk) due to changing markets before each leg of the action is settled.
- Increased risk of manual mistakes (fat fingers, bad slippage settings).
- Increased risk of falling victim phishing attacks on one of the platforms..

A second abstraction is to let users define what they want, and provide them with the information (the calldata) how to do it. Since most collateral management actions require interactions with multiple assets/protocols, this abstraction can only be achieved after portfolio's as a whole can be managed with a single atomic transaction.

At the time of writing this paper, the required infrastructure to enable both abstractions, is heavily debated within the broader ecosystem and multiple proposals are launched to standardize the infrastructure. Most notable are the proposals for Account Abstraction (e.g. EIP-3074 and EIP-4337) and for Intent based architectures (e.g. EIP-7521)

Both abstractions are already successfully applied (albeit in a somewhat non-standardized form) within DeFi by Decentralised Exchanges, NFT-marketplaces and their aggregators.

Our philosophy is that users should express what they want to do, not how they do it.

## 4.3 New Architectures

During the last months many solutions were proposed by different protocols and stakeholders on how to build a more robust decentralised financial system. Notable recent papers are from Uniswap V4[1], Morpho[2] and Euler[5]. All mention similar concepts as the distinction between product and protocol, modularization of the protocol, oracle agnostic implementations and explicitly separating the logic of the protocol in different layers, where layers with different complexity evolve at different speeds.

Especially the last concept, to separate the logic in different layers, resonates with the long term vision of the Arcadia Finance team. Different Logic layers of the financial stack, with different complexities should have different life-cycles and innovation timelines. With the lowest most core component the slowest moving,

while the upper user facing layers must evolve and adapt quick in response ever changing markets.

A good practical example is uniswap V4, where Uniswap Labs implements the underlying core mathematical logic for CLAMMs (Concentrated Liquidity Automated Market Makers). Other teams can build in a permissionless manner on top of the Uniswap V4 protocol and develop feature rich and fast innovating new DEXs, without the need and risks of redundantly implementing the complex maths.

Also over-collateralized protocols would benefit from a similar architecture. A shared standardized and permissionless layer with battle tested logic for collateral management, on top of which Creditors build their application specific financial contracts.

Bringing back pristine focus to collateralization is exactly what we are doing with the Arcadia Financial Technology Stack. We believe that taking a collateral management first approach results in overall better DeFi protocols.

## 5 THE ARCADIA FINANCIAL TECHNOLOGY STACK

the Arcadia Financial Technology Stack is implemented for the Ethereum Virtual Machine (EVM) and consists out of three protocols:

- The Arcadia Registry.
- The Arcadia Accounts.
- The Generalised Creditor.

## 6 ARCADIA REGISTRY

The Arcadia Registry is non opinionated on-chain infrastructure with two main functionalities:

- (1) Standardize pricing logic to denominate a portfolio of on-chain assets in a given numeraire (unit of account).
- (2) Standardize logic to store, manage and process asset related risk parameters/variables.

### 6.1 Modular pricing logic

The Arcadia Registry is not a monolithic smart contract containing all logic. It consists of a main coordinating smart contract (also referred to as The Registry) and multiple Modules.

Each Module is a separate smart contract with the pricing logic for specific Oracle implementations or Asset types. The Registry keeps mappings per Creditor of which Module to use for a certain asset.

It is the Creditor itself that sets these mappings, the Creditor decides which Modules to use for each asset they allow their Debtors to use as collateral. Different Creditors can use different Modules to price the same asset, or a Creditor can choose to not allow certain assets at all.

Modules can only be appended to the Registry, not removed or overwritten. This ensures that Pricing logic is immutable, but it still gives flexibility to add new assets, or implement more efficient Pricing logic over time.

Working with this modular architecture has a number of advantages:

- Modules are re-usable by different Creditors, instead of redundant monolithic Pricing Contracts per Creditor, battle tested Pricing/Risk logic can be reused.
- Different Modules can call each other recursively, creating Pricing/Risk logic that is as composable as tokenized assets.
- Logic for new assets, or even asset types, can be added without the need of upgradable Proxy contracts and their inherent trust assumptions.

There are two main categories of modules: Oracle Modules and Asset Modules.

### 6.2 Oracle Modules

The Oracle Module implements the logic to convert oracle-rate of different oracle technologies into a standardized format. Each different oracle implementation (e.g. Chainlink oracles, Pyth oracles, Uniswap TWAPs...) has its own Oracle Module.

An oracle has a quote-asset and a base-asset, the oracle-rate  $R_{oracle}$  reflects how much tokens of the quote-asset are required to buy 1 token of the base-asset. The precision  $S_{oracle}$  of oracles is variable and can even be a binary fixed-point number. Since all pricing logic within the Arcadia Protocol uses fixed-point numbers with 18 decimals precision, a correction from the oracle precision to a precision of 18 decimals is required for the standardized oracle rate  $R_{ba \rightarrow qa}$ , returned by the Oracle Module.

$$R_{ba \rightarrow qa} = R_{oracle} \frac{10^{18}}{S_{oracle}} \quad (1)$$

### 6.3 Asset Modules

Just as Oracle Modules implement the logic to return oracle-rates in a standardized format, Asset Modules will return Asset values in a standardized format. And just as each oracle implementation has its own oracle Module, each asset type has its own Asset Module.

Next to pricing logic, Asset Modules also store and manage asset specific risk parameters, which we will discuss in more detail in Paragraph 6.5.

Asset Modules can be further divided into two distinct groups: Primary Asset Modules and Derived Asset Modules.

**6.3.1 Primary Asset Modules.** Primary assets are defined as assets that are not composed of other assets. Primary Assets must be priced using one or more on-chain oracles, for which the rates  $R_{ba \rightarrow qa}$  are fetched from the Oracle Modules via the Registry.

Each asset of a Primary asset is priced in USD with 18 decimals precision. Since the asset amount,  $A_{asset}$  can have a variable precision (number of decimals), a correction  $S_{asset}$  is again applied to bring the usd-value of the asset  $V_{asset}^{usd}$  to 18 decimals precision.

Depending of the direction of the oracle, the value of the Primary Asset is calculated different. When the asset is the base-asset of the oracle:

$$V^{usd}(A_{asset}) = R_{asset \rightarrow usd} \frac{A_{asset}}{S_{asset}} \quad (2)$$

When the asset is the quote-asset of the oracle:

$$V^{usd}(A_{asset}) = \frac{1}{R_{usd \rightarrow asset}} \frac{A_{asset}}{S_{asset}} \quad (3)$$

For some assets, there might not be a direct oracle to USD. For these assets multiple oracles can be chained together:

$$V^{USD}(A_{asset}) = R_{asset \rightarrow x} R_{x \rightarrow USD} \frac{A_{asset}}{S_{asset}} \quad (4)$$

$$V^{USD}(A_{asset}) = \frac{R_{y \rightarrow USD}}{R_{y \rightarrow asset}} \frac{A_{asset}}{S_{asset}} \quad (5)$$

**6.3.2 Derived Asset Modules.** Derived assets are defined as assets that are composed of one or more underlying assets (which can be primary assets or other derived assets). Examples of derived assets are liquidity positions of AMMs, staked assets, yield bearing tokens.

The derived Asset Modules will calculate the USD-value in three steps. It will first calculate the amounts of the underlying assets  $A_u$ , given the amount of the asset to be priced  $A_{asset}$ .

$$A_u = f_u(A_{asset}) \quad (6)$$

Where  $f_u(A)$  is a flashloan-resistant and protocol specific function. For example for a Uniswap V2 Liquidity Positions it can be derived from the reserves of the pool, or for ERC4626 tokens it can be calculated by calling `convertToAssets(shares)`.

In a second step the Derived Asset Module will recursively fetch for the USD-value of each of the underlying assets  $V^{USD}(A_u)$  from the Registry, using the underlying asset amounts  $A_u$  from Equation 6.

$$V^{USD}(A_u) = V^{USD}(f_u(A_{asset})) \quad (7)$$

The underlying assets of a derived assets can be derived assets themselves (e.g. staked LP tokens, or liquidity positions of AMMs where one of the tokens is a yield bearing asset). In this case the underlying asset will in turn fetch the USD-values of their underlying asset(s). The stop conditions of the recursion are when the underlying asset is a Primary asset, or if a maximum recursion depth (set by the Creditor) is reached. By working with modular and recursive Asset Modules, the pricing logic of the Registry fully captures the benefits of composability withing DeFi.

Lastly, the total USD value of the asset,  $V^{USD}(A_{asset})$ , is calculated as the sum of the USD values of its underlying assets. Since all underlying assets are valued with 18 decimals precision, no additional precision correction is required.

$$V^{USD}(A_{asset}) = \sum_u V^{USD}(A_u) = \sum_u V^{USD}(f_u(A_{asset})) \quad (8)$$

## 6.4 Portfolio Pricing

Section 6.1 explains how the Registry can calculate the value of assets in USD. Building further on this logic, the Registry can price any combination of assets in any numeraire (the asset used as unit of account).

Given a portfolio  $P$  consisting of assets with amount  $A_i$ , the value of the portfolio denominated in a certain numeraire  $V_{spot}^{num}(P)$  with precision  $S_{num}$  is given as:

$$V_{spot}^{num}(P) = \frac{\sum_i V^{USD}(A_i) S_{num}}{R_{num \rightarrow USD} 10^{18}} \quad (9)$$

The result of equation 9 is the spot or MtM (mark-to-market) value of the portfolio. Next to the MtM value of the portfolio, the Registry can also calculate risk adjusted portfolios value, where assets are discounted with a asset-numeraire specific risk factor  $RF_i^C$ ,

set by each Creditor. The risk adjusted portfolio value  $V_{risk}^{num}(P)$  is then given as:

$$V_{risk}^{num}(P) = \frac{\sum_i RF_i^C \cdot V^{USD}(A_i) S_{num}}{R_{num \rightarrow USD} 10^{18}} \quad (10)$$

## 6.5 Creditor Specific Risk Parameters

Each Creditor can set a number of risk parameters in the Registry to protect themselves against different risks. These risk parameters are taking into account when depositing assets, or when risk-adjusted portfolio values are calculated.

The risk parameters can be asset specific, protocol specific or a single value per creditor.

**6.5.1 Collateral, Liquidation and Risk Factors.** As is described in detail in Paragraph 7.4, collateral factors and liquidation factors are used to discount asset values to account for market, liquidity and smart-contract risks.

Each asset  $i$  that can be priced by the Registry has a Creditor specific collateral factor  $CF_i^C$  and liquidation  $LF_i^C$  factor. For Primary Assets, the factors are set directly by Creditor.

For Derived Assets, the risk factors are a function of the risk factors of its underlying assets  $u$ , and optionally an additional protocol specific risk factor  $RF_p^C$ , to account for the additional complexity smart contract risk of derived assets:

$$CF_i^C = f(\sqrt{CF_u^C}, RF_p^C) \quad (11)$$

The relation between the risk factor and the risk factors of the underlying assets will depend on the specific nature of the derived asset. It can e.g. be the minimum of each risk factor of the underlying assets, a weighted average or something else.

For derived assets with a single underlying asset the risk factor becomes straightforward and is just the multiplication of the risk factor of the underlying asset and the protocol specific risk factor. As an example, take a protocol ( $p$ ) with yield bearing assets, which has an *ETH* and a *USDC* vault. The collateral factors of the corresponding  $pETH$  and  $pUSDC$  tokens are:

$$CF_{pETH}^C = RF_p^C \cdot CF_{ETH}^C \quad (12)$$

$$CF_{pUSDC}^C = RF_p^C \cdot CF_{USDC}^C \quad (13)$$

**6.5.2 Maximum Exposure.** The maximum exposures give Creditors a tool to set an upper boundary against the overall exposure they can have against certain primary assets or protocols. Creditors can even choose to not be exposed to certain primary assets or protocols (via derived assets) by setting the corresponding maximum exposure to 0.

There is again a difference how the maximum exposure is defined for primary assets and for derived assets.

For primary assets, the maximum exposure is defined per primary asset and per creditor. It limits the total amount (in the unit of the underlying asset) of exposure that a single Creditor can have against the primary asset. As such assets with low on-chain liquidity or small market caps can be used as collateral, while market manipulation risks, such as the Mango Market Exploit[12] can be mitigated.

The total exposure of a primary asset takes into account both direct deposits of the primary asset, as indirect exposures via derived

assets. If a maximum exposure is reached, no additional amount of the primary asset can be deposited, nor can any derived asset be deposited that is composed of the primary asset.

For derived assets, there are two mechanisms that provide the Creditor to an upper bound for certain risks. As mentioned before, none of the primary assets of which the derived asset is composed can exceed their maximum exposure to mitigate risks of market manipulation and low liquidity. Secondly, each creditor can set an upper limit to the total USD exposure of all derived assets of a certain protocol (e.g. the sum of all liquidity position of Uniswap V2). As such the Creditor can limit potential losses due to exploits in the protocol.

**6.5.3 Minimum Margin.** Liquidating unhealthy positions has a considerable gas cost, which liquidators take into account when deciding to liquidate a position or not. This gas cost is independent of the size of the open position. Not taking into account the size independent gas cost could lead to very small positions that are never profitable to liquidate. Or even worse, liquidations that are successfully started, but never profitable to end.

To avoid both problems, accounts must have a Creditor specific minimum value of collateral, the minimum margin  $MM^C$ , before they can open a position. The minimum margin can not be used to back liabilities and can be seen as an amount of collateral that is locked to at least cover gas costs of Liquidators.

## 7 ARCADIA ACCOUNTS

### 7.1 Multi-collateral

By allowing multiple types of assets to be deposited within a single Account, users, institutions and protocols can build diversified portfolio's. The list of asset(type)s that can be deposited in Arcadia Accounts is continuously being expanded. Our goal is for users to unlock capital of any quality, price-able DeFi primitive.

Each Creditor can determine themselves which asset(type)s are to be allowed as collateral in Accounts of their Debtors.

### 7.2 Composable

Arcadia Accounts themselves are according to the ERC-721 standard and can thus be represented as a single asset. This means that Accounts are fully composable with existing infrastructure and are straightforward to integrate. Other benefits are that Accounts can be tracked and monitored through existing DeFi dashboards and it opens up use-cases where Accounts can be sold in their entirety (both assets and liabilities) on existing NFT marketplaces.

### 7.3 Token Standard Agnostic

### 7.4 Margin Accounts

### 7.5 Account Abstraction ready

### 7.6 Flash Actions

Flash Actions (or optimistic Actions or flash accounting) expands on the concept of flashloans, and are only possible thanks to a unique property of smart contracts: atomicity[13]. Just as with flashloans, each step of the Flash Action must be successful or the transaction as a whole fails. The final health check of the Account (assets can cover all liabilities) is done at the very end of the transaction. This

allows the Account Owner to temporarily bring the Account in an undercollateralized (or even non collateralized state) without the risk on bad debt for any Creditor. Since if the Account is not brought back into a healthy state during the transaction, the final health check fails and thanks to atomicity the whole transaction fails.

This gives Account Owners unprecedented flexibility to manage assets and liabilities. In a fully permissionless way they can chain the following together (= do a flash action):

- A margin Account can be opened for a new Creditor, if the new Creditor is approved by the Account Owner.
- The Creditor can execute arbitrary logic (e.g. give a flashloan).
- The Account owner can optimistically withdraw assets from the Account.
- The Account owner can transfer assets from his own wallet to the Account or external logic.
- The Account owner can execute external logic, where he uses the assets (loaned, withdrawn or transferred) to interact with multiple DeFi protocols to swap, stake, claim...
- The Account owner can deposit recipient tokens back into the Account.

The only requirement is that the Account is in a healthy state at the very end of the flash action.

Flash Actions are a very powerful tool that has no equal in traditional finance. We will list a few examples how flash actions from an Arcadia Account can be used out of the box (the list is far from exhaustive). And keep in mind that all of these actions can be done, even if the assets are used to secure liabilities.

- Rebalance whole portfolio's, swapping a portfolio of n different assets directly to a new portfolio of m different assets.
- Refinance liabilities (change to a different Creditors), without the need to sell any collateral.
- Stake or provide liquidity on approved DeFi protocols. External protocols used in this context will need to provide a receipt token which must be allowed as collateral as well within the Arcadia protocol. Examples can be providing liquidity on Aave (receiving approved aTokens), depositing assets on Yearn (receiving approved yTokens), ...
- Change ranges for Uniswap V3 LP. Contrary to Uni V2 and similar AMMs, Uni V3 positions are meant to be more actively managed in terms of liquidity ranges. Users that deposit Uni V3 positions in their Arcadia Accounts will have the ability to change those liquidity ranges without having to withdraw their tokens first.
- Claim airdrops that depend on address-owned tokens. Arcadia Accounts will feature "flash withdrawals". This feature can be used by the Account owner to claim airdrops using assets under collateral within their Arcadia Account, without having to close DeFi positions to withdraw their tokens first.
- ...

## 7.7 Intent ready

## 7.8 Automated Asset Management

## 7.9 Upgradeability

Each deployed Account is linked to a certain Account logic contract. Account logic will include the features Account owners can benefit from, for example, flash withdrawals, active collateral management, authorization delegation, ... The Account logic is upgradable, but the user has full control if and when they want to upgrade to new logic. Should a user wish to use features newly introduced in an upcoming Account version, it will be up to them to upgrade the linked Account logic. Users will not have to migrate assets or close DeFi positions when doing so.

Creditors can determine which Account logic versions are allowed to be used by their Debtors. As such, highly-customized Account logic can be implemented for protocol-specific versions should this be required. The Arcadia protocol can in no way change the version of a user-owned deployed Account, or change any functionality in existing Account logic.

## 8 ARCADIA CREDITOR

### 8.1 Generalised Creditor

### 8.2 Composable

### 8.3 Risk Management

### 8.4 Accounting Liabilities

### 8.5 Liquidations

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

### 8.6 Arcadia Lending Pools

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

## 9 SUMMARY

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris



Figure 1: Example of an image.

felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

## REFERENCES

- [1] H. Adams, M. Salem, N. Zinsmeister, and et al. 2023. Uniswap v4 Core [Draft]. Draft.
- [2] Mathis Gontier Delaunay, Paul Frambot, Quentin Garchery, and Matthieu Lesbre. 2023. Morpho Blue Whitepaper. (2023). <https://github.com/morpho-org/morpho-blue/blob/main/morpho-blue-whitepaper.pdf>
- [3] J. Gregory. 2014. *Central Counterparties: Mandatory Central Clearing and Initial Margin Requirements for OTC Derivatives*. Wiley. <https://books.google.be/books?id=DpXqAwAAQBAJ>
- [4] Martin Hellwig. 2008. The causes of the financial crisis. In *CESifo Forum*, Vol. 9. München: ifo Institut für Wirtschaftsforschung an der Universität München, 12–21.
- [5] Euler Labs. 2023. Protocols vs Products. (2023). <https://www.euler.finance/blog/protocols-vs-products>
- [6] Robert Leshner and Geoffrey Hayes. 2019. Compound: The money market protocol. *White Paper* (2019).
- [7] P. Millett. 2002. *Lending and Borrowing in Ancient Athens*. Cambridge University Press. <https://books.google.be/books?id=0IriOXQSe5QC>
- [8] Fabian Schär. 2021. Decentralized finance: On blockchain-and smart contract-based financial markets. *FRB of St. Louis Review* (2021).
- [9] M. Simmons. 2019. *Collateral Management: A Guide to Mitigating Counterparty Risk*. Wiley. <https://books.google.be/books?id=DMOKDwAAQBAJ>
- [10] Maker Team. 2017. The dai stablecoin system. (2017). <https://makerdao.com/whitepaper/DaiDec17WP.pdf>
- [11] Harry Thornburg. 2020. Aave protocol whitepaper v1. (2020). [https://github.com/aave/aave-protocol/blob/master/docs/Aave\\_Protocol\\_Whitepaper\\_v1\\_0.pdf](https://github.com/aave/aave-protocol/blob/master/docs/Aave_Protocol_Whitepaper_v1_0.pdf)
- [12] Fran Velasquez. [n. d.]. DeFi Exchange Mango's \$114M Exploit Was 'Market Manipulation,' Not a Hack, Ex-FBI Special Agent Says — coindesk.com. <https://www.coindesk.com/tech/2022/10/20/defi-exchange-mangos-114m-exploit-was-market-manipulation-not-a-hack-ex-fbi-special-agent-says/>. [Accessed 30-01-2024].
- [13] Yining Xie, Xin Kang, Tieyan Li, Cheng-Kang Chu, and Haiguang Wang. 2022. Towards Secure and Trustworthy Flash Loans: A Blockchain-Based Trust Management Approach. In *International Conference on Network and System Security*. Springer, 499–513.

## DISCLAIMER

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi.

Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit

amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.