

1INCH AQUA

SHARED LIQUIDITY LAYER

Anton Bukov
k06aaa@gmail.com

Sergej Kunz
info@deacix.de

Gleb Alekseev
alekseev.gleb@gmail.com

Sergey Prilutskiy
sergey@gmail.com

ABSTRACT

Aqua is a shared liquidity layer protocol that utilizes a liquidity provider-centric model. This model allows liquidity to remain in LP-controlled wallets, where capital is managed by the protocol through strategies with shared balances. In addition to consolidating order flow and provisioning liquidity across multiple strategies, Aqua is built around two complementary properties: shared liquidity and unlocked capital. Shared liquidity enhances capital efficiency by allowing the same wallet equity to back multiple strategies, while unlocked capital preserves custody and enables higher utility efficiency by keeping assets available for other DeFi uses such as money markets, governance, and staking. For strategy builders, Aqua separates custody and balance accounting from trading logic, allowing them to concentrate solely on the trading logic and mathematical aspects.

Version: Developer Preview 1.0

Repository: <https://github.com/1inch/aqua>

1 Current state of DeFi

In today's DeFi market, capital is fragmented not only across protocols, but also across execution models. First, on-chain AMMs pool liquidity into shared smart-contract accounts: all funds are locked in a common vault, and traders (Takers) interact with an aggregated balance rather than with individual liquidity providers (Makers). This is convenient for takers and aggregators, because it reduces liquidity fragmentation consolidating liquidity in a single target.

At the same time, models with off-chain order creation and discovery, such as limit orders, RFQ workflows, and PMM strategies, keep the LP's capital on their own account: funds are moved on-chain when a taker executes the trade. For LPs, AMMs are capital-inefficient: liquidity must be locked into specific pools with predefined parameters (e.g. price ranges, fees), while the same capital in PMM-like strategies can simultaneously support multiple markets and configurations. Off-chain PMM setups are more capital-efficient for LPs, provided they can manage their trading strategies effectively.

2 Problem Statement

AMMs have become the dominant liquidity primitive in DeFi, but their design creates several structural problems for LPs.

First, AMM liquidity is highly fragmented from the maker's perspective: capital is spread across many pools, fee tiers, and price ranges, often with uneven depth be-

tween "blue chip" pairs (e.g., major stablecoins and ETH pairs) and the long tail of assets. Popular pools attract huge reserves, while pools for smaller or newer assets struggle to accumulate enough liquidity to be usable.

Second, AMMs are built around TVL as locked capital: this means that capital assigned to one pool cannot be reused elsewhere, even if market conditions shift and other opportunities become more attractive.

Third, these properties combine into lower capital efficiency for makers, who must manually decide "where to park liquidity," constantly rebalance between pools, and accept the operational and security risks of moving funds across multiple protocols and deployments.

Aqua addresses these problems by introducing a new concept: a shared liquidity layer protocol — a dedicated layer for managing access to LP's liquidity. In this model, a single token balance in a maker-controlled wallet can back multiple strategies, fee tiers, and even entirely different types of AMM at the same time. Strategies request and consume liquidity through Aqua's shared accounting, activating capital only when and where it is needed. From the maker's perspective, this reduces fragmentation, eliminates the need to pre-allocate and lock capital into many separate pools, and simplifies the operational problem of managing liquidity in a complex AMM landscape.

In order to achieve this, Aqua handles the accounting and permissioning of balances available to strategies, while the strategies themselves focus solely on their trading logic and math. Instead of implementing their own deposit and withdrawal flows, strategies integrate with

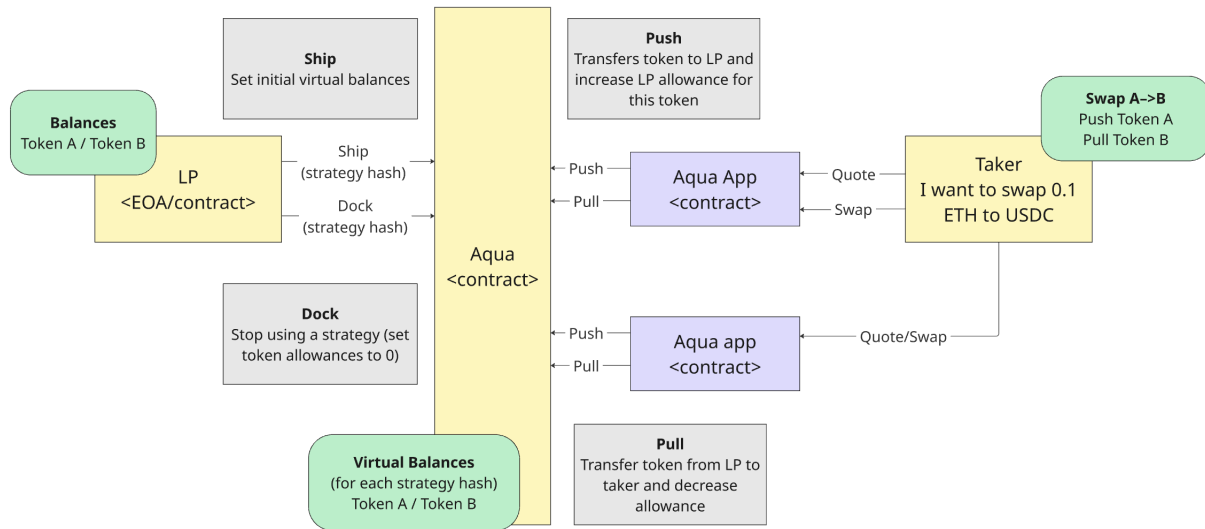


Figure 1: Aqua protocol interactions

Aqua. Liquidity providers grant structured access to their balances, and Aqua tracks which portions of those balances are reserved, available, or in use across multiple strategies, without actually locking funds inside each strategy’s smart contract. From the builder’s perspective, strategy development becomes cleaner: teams can focus on designing better pricing curves, fee schedules, and risk models, while relying on Aqua’s shared liquidity layer for secure and efficient balance management.

Unlike traditional AMMs, where funds are locked in pool contracts, Aqua keeps liquidity under the user’s control.

3 Protocol design

Aqua introduces the concept of virtual balances. where each token balance represents an allowance granted by a Liquidity Provider (LP), or Maker, enabling specific Strategies to utilize those tokens. The Strategies represent the trading logic within the Aqua application, along with a set of associated parameters (such as token pairs, fees, etc.) that define how the logic is applied.

A key design principle of Aqua is that it is permissionless for strategy developers. Aqua allows anyone to build and launch Strategies on top of the shared liquidity layer without requiring centralized approval. Conceptually, these Strategies are AMMs: they implement pricing formulas, inventory management, and quoting logic, but instead of being tied to a dedicated on-chain pool, they operate as AMM strategies consuming liquidity from Aqua.

As shown on Figure 1, the basic Aqua operations are `ship()` and `dock()`. The `ship()` operation registers an Aqua App and its Strategy (a set of parameters) and allocates initial token balances. Conversely, the `dock()` op-

eration, also initiated by the LP, revokes Aqua balances from the specific strategy.

Aqua’s core operational functions are `push()` and `pull()`. These operations retrieve assets from a Maker or send assets to a Maker, adjusting Aqua virtual balances accordingly.

A Strategy uses actual tokens and performs transfers, which will fail if insufficient funds are present at the actual addresses and changes Aqua balances accordingly.

The actual transfer of tokens is performed only if all strategy logic is executed correctly. Transfers happen only at the end and only if the Maker has a sufficient balance. In this case, an Aqua AMM can implement any DEX logic (standard constant product, stable swap, concentrated liquidity, etc.) and add any tweaks such as progressive fees, spread decay, and many other features.

4 Economic considerations

4.1 Capital efficiency in Aqua and traditional AMMs

The diagram 2 contrasts pooled custody with shared allowances. On the left, LPs deposit into a single AMM contract, funds sit locked in the pool, and takers swap directly against that aggregated balance. On the right, LP funds remain in Maker wallets while Aqua sits between Makers and multiple applications’ strategies: takers still swap at the strategy level, but strategies source and return inventory via Aqua’s accounting. Since these strategies execute asynchronously, the same token balance can compound its utility across multiple opportunities, thereby improving capital efficiency.

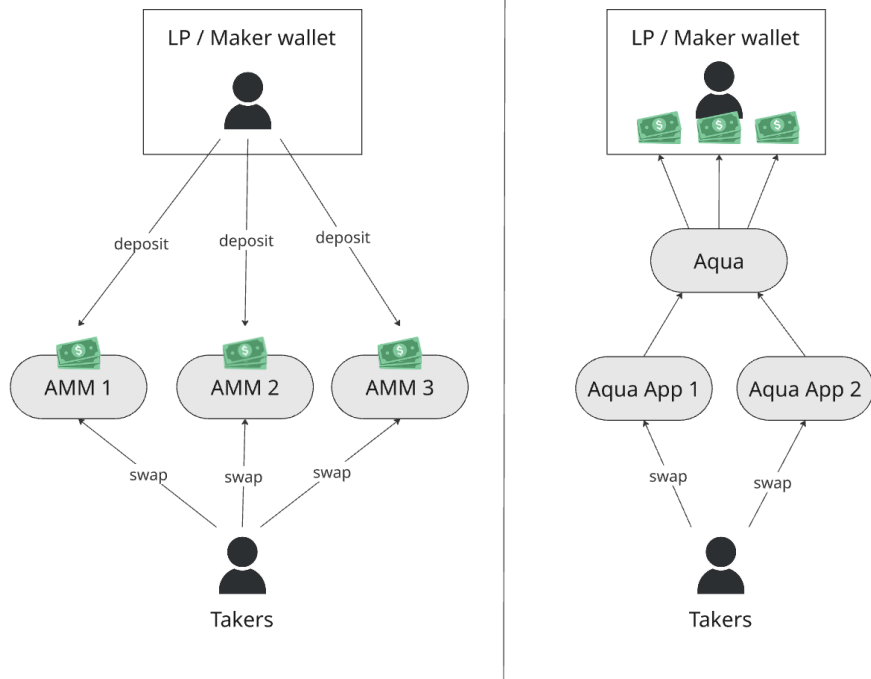


Figure 2: Traditional AMM model vs Aqua

4.2 Specialization over Homogenization

Traditional AMMs homogenize LP behavior through a single invariant and fee tier. Aqua maximizes specialization: Makers choose which Strategies to authorize and how to configure them, while each Strategy implements bespoke pricing (constant product, stable, concentrated, inventory-skewed, decay spreads, etc.) without rebuilding custody. Depth and pricing become a function of skill and configuration, not forced by pool templates.

4.3 Fair and Unfair Flow

In traditional AMMs, LPs see all incoming trades as one big stream. They cannot tell fair volumes (flow the pool “deserves” at its own prices, given its liquidity and formulas) from unfair volumes (flow that only comes when the pool is mispriced vs. the market). Arbitrage, MEV strategies, and informed takers can hit AMM prices when they are most misaligned with the broader market. For LPs, this manifests as persistent adverse selection: even if volumes and fees look attractive, a share of realized PnL is transferred to faster or better-informed counterparties through impermanent loss and unfavorable inventory shifts.

Aqua turns adverse selection into priced, bounded risk. Strategies adapt spreads to inventory and volatility, Makers set Strategy-specific and global caps, and cross-Strategy hedging nets exposures in real time.

4.4 Shared vs unlocked liquidity: capital and utility efficiency

Aqua’s design combines two core properties that are usually mutually exclusive in AMM-style systems:

- **Shared liquidity.** The same wallet equity can be provisioned to multiple Strategies through Aqua’s virtual balances and risk checks;
- **Unlocked capital.** Underlying tokens remain in the Maker’s own wallet, rather than being locked inside pool contracts.

Shared liquidity primarily improves capital efficiency: the same unit of equity can support multiple, asynchronous Strategies and markets, increasing aggregate notional exposure without increasing the amount of capital that must be parked in pools at any given moment. In contrast, unlocked capital primarily improves what we call utility efficiency: the ability of that same unit of capital to simultaneously participate in other DeFi protocols and rights (collateral, governance, staking, etc.) while it is being used as liquidity.

Formally, capital efficiency in this context measures how much notional liquidity a given wallet equity can support under Aqua’s netting and risk limits. Utility efficiency measures how many independent “uses” the same collateral can serve at once: for example, acting as LP inventory, serving as money-market collateral, and carrying governance or voting power. In traditional AMMs, once tokens are deposited into a pool, their utility efficiency for

the LP drops nearly to zero: they cannot be used as collateral, cannot vote in gauges, and cannot carry protocol-specific rights that require custody of the tokens in the LP's own account.

Because Aqua leaves custody with the Maker and only operates via allowances and virtual balances, the same assets can remain fully composable with the rest of DeFi. A Maker can, for instance, deposit 1,000 USD worth of tokens into a money market, borrow additional assets against this collateral, and route the resulting portfolio through Aqua. Suppose that 1,000 USD of equity in a lending protocol is levered into 3,000 USD of collateral and 2,000 USD of debt. Those 3,000 USD worth of assets can then be authorized to several Aqua Strategies; with shared liquidity, each token can participate in multiple positions (for example, each asset appearing in two different pairs or Strategies), so that aggregate notional liquidity may reach on the order of 6,000 USD while the underlying equity is still 1,000 USD and the debt is explicitly tracked by the money market. Aqua's risk limits and allowances bound how much of this stacked exposure can actually be consumed at any moment.

Unlocked liquidity also improves utility efficiency beyond leverage. Since tokens remain in Maker-controlled wallets, they can still be used wherever the underlying protocol recognizes on-chain balances: for example, participating in governance voting, gauge or ve-style voting systems, or other staking and rewards programs, provided that these systems only require holding or locking tokens at the wallet level. In a traditional AMM, tokens locked in pool contracts cannot vote or signal preferences; under Aqua, the same tokens can provision liquidity to multiple Strategies and continue to contribute to governance or external reward mechanisms, as long as those mechanisms are compatible with the Maker's custody and risk constraints.

In other words, shared liquidity and unlocked capital are complementary. Shared liquidity multiplies the capital efficiency of a given equity base inside Aqua; unlocked capital multiplies the utility efficiency of that equity across DeFi as a whole. Together, they allow LPs to treat their balance sheet not just as "TVL parked in pools," but as a composable, multi-purpose resource that can be allocated across Strategies, money markets, and governance without redundant locking and unlocking cycles.

5 Liquidity Strategy Lifecycle

Makers assemble portfolios of Strategies, activate them against their wallet balances, let them earn fees, and then retire them as conditions change. Funds remain in Maker-controlled wallets throughout.

Assembly and activation. In the assembly phase, an LP chooses a set of Strategies that match their risk and market view. Each Strategy fully defines how assets will be managed (pricing, spreads, fees, etc.). Aqua itself does not

guarantee profitability; that is the responsibility of Strategy builders. Once the portfolio is chosen, the Maker calls `ship()` for each strategy to activate them and assign virtual balances (per-token allowances) without transferring custody; unlike traditional AMMs, activation is configuration rather than a deposit. Example of strategies composition is on Figure 3. Each line defines a separate strategy between two tokens.

Utilization: `pull()` / `push()`. When Takers trade against a Strategy, Aqua's `pull()` and `push()` are used to consume and return liquidity. `pull()` draws tokens from the Maker's wallet up to the allowed virtual balance and `push()` sends tokens back.

`push()` introduces a new concept of auto-increasing virtual balances. Whenever assets are sent back to a Maker via `push()` Aqua automatically credits the corresponding Strategy's virtual balances. In other words, any tokens that flow back expand the Strategy's usable assets, thereby compounding potential utility without requiring a separate deposit or configuration step.

Retirement: `dock()`. To stop using a Strategy, the Maker calls `dock()`, which revokes its Aqua virtual balances and prevents further consumption of liquidity, while all funds remain in the Maker's wallet. In practice, a Maker can deactivate underperforming Strategies, reassigning the same capital to new Strategies without withdrawing from pool contracts or unwinding complex positions.

6 Security Model and Considerations

Aqua's security model is built around three pillars: Maker-controlled custody, allowance-based access, and strict balance invariants. The protocol never takes unrestricted custody of tokens: at any moment, the maximum capital at risk is bounded by the allowances the Maker has granted.

6.1 Custody and approvals

Before any Strategy can consume liquidity, the Maker must approve the Aqua contract for spending a specific token. This ERC-20 approval defines a hard upper bound on how much of that token Aqua can ever pull from the Maker's wallet, regardless of how many Strategies and allowances are configured.

6.2 Balance invariants and illiquidity

Every `pull()` is ultimately limited by the real on-chain balance of the Maker's wallet: if there are not enough actual tokens to satisfy the requested transfer, the operation reverts and the swap is not executed. This creates a situation of illiquidity: there may be sufficient virtual balances from Aqua's perspective (Strategies appear "funded"), but if the underlying wallet does not hold enough tokens at the

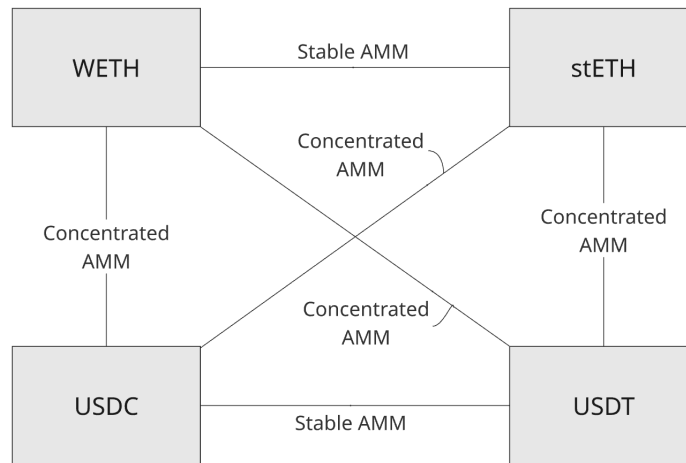


Figure 3: Strategy composition example

moment of settlement, trades simply fail rather than partially executing.

6.3 Impermanent loss under illiquidity

While a Strategy is illiquid (virtual balances exist, but real tokens are missing), its quotes may continue to move with the market, but no trades can actually settle because `pull()` cannot succeed. The risk appears when liquidity returns: if prices have moved unfavorably during the illiquid period, the first trades executed after funds reappear may immediately realize what is effectively an “instant impermanent loss” relative to the earlier price path.

6.4 Dutch auctions and path-dependent losses

A special case arises when a Strategy implements a Dutch auction or similar mechanism that gradually walks the price down in search of liquidity while the Maker’s wallet is illiquid. If the real balance is insufficient, the auction may continue decreasing the price until it reaches unfavorable levels. When liquidity finally appears, the accumulated demand at those extreme prices can be filled at once, effectively selling the asset near the bottom. In such cases, even if the market later reverts, the loss is already realized and cannot be reversed by future price movements because the inventory has been sold or acquired at the auction’s terminal price.

6.5 Operational guidelines for Makers

Within this model, Aqua provides strong guarantees about custody and invariants, but economic safety remains the LP’s responsibility. LP’s are encouraged to:

- keep ERC-20 approvals to Aqua at reasonable, strategy-aligned levels;

- monitor that real wallet balances remain in line with Strategy virtual balances to avoid prolonged illiquidity;
- promptly deactivate Strategies that become structurally illiquid or economically unviable.

7 Conclusion

Aqua re-defines DeFi liquidity from the paradigm of total value locked to the paradigm of total value allocated, making the whole DeFi market more competitive. Instead of treating capital as something that must be immobilized inside isolated pools, Aqua treats it as a shared resource that remains under LPs’ custody and is selectively allocated to Strategies via explicit approvals and virtual balances. In this model, the key metric is no longer how much value is locked in contracts, but how much value is actively provisioned to productive strategies at any moment in time. By combining shared liquidity with unlocked capital, Aqua increases not only the capital efficiency of LPs’ balance sheets, but also their utility efficiency across DeFi.

When Strategies are designed and managed effectively, this architecture creates an opportunity to multiply capital efficiency for LPs. The same wallet equity can simultaneously support multiple execution venues and pricing models, subject to well-defined risk limits. Capital is not merely parked; it is continuously reallocated and reused across a portfolio of Strategies. At the same time, because assets remain in LP-controlled wallets, they can still serve additional roles — for instance as money-market collateral, governance or gauge voting power, or staking positions — instead of being inert inside pool contracts. Liquidity becomes both more productive inside Aqua and more useful in the broader DeFi ecosystem.

For builders, Aqua acts as a complete infrastructure layer for custody, accounting, approvals, and lifecycle management. Strategy builders do not need to implement de-

posits, withdrawals, or bespoke balance tracking. Instead, they can focus entirely on trading logic, pricing math, and risk models, while Aqua provides shared liquidity, unlocked capital, and a consistent interface for accessing and allocating LP equity.

A Terms and definitions

Aqua Application A smart contract that implements trading strategy logic and interacts with the Aqua protocol.

Strategy builders Developers or teams who design and implement trading Strategies on top of Aqua. Strategy builders focus on pricing logic, execution algorithms and risk management. They use Aqua's primitives (e.g. ship, dock, pull, push) to consume provisioned liquidity and expose their Strategies as reusable components that Makers and Aqua Applications can opt into.

AMM (Automated Market Maker) A class of on-chain exchange protocols where liquidity is pooled into smart contracts, and prices are determined by a predefined formula (e.g. constant product). Traders interact with a shared pool balance instead of posting individual bids/asks.

PMM (Private Market Maker) A market maker that runs its pricing and risk logic off-chain (or in proprietary infrastructure) and quotes prices to venues (RFQ, limit order books, auctions, etc.). PMMs typically manage their own inventory and risk models, and interact with on-chain liquidity via orders or RFQs instead of depositing funds into AMM pools.

LP (Liquidity Provider) An entity (user, fund, market maker) that allocates capital to a protocol or

strategy so that others can trade against it. In AMMs, an LP deposits tokens into pools; in Aqua, an LP/Maker authorizes Aqua to provision liquidity from its wallet to multiple Strategies.

Maker A participant who provides liquidity or quotes (prices) to the market. Makers stand ready to buy and sell, earning fees or spread in return for taking inventory and price risk. In Aqua, a Maker is an LP whose balances are shared across Strategies via virtual provisioning.

Taker A participant who consumes liquidity at the prices offered by Makers. Takers submit swaps, market orders, or RFQ requests and accept the quoted price or pool price, paying fees and implicitly compensating Makers for providing liquidity and bearing risk.

Capital efficiency A measure of how much notional liquidity or trading exposure a given amount of wallet equity can support under Aqua's risk checks and virtual balance model. Higher capital efficiency means that the same unit of equity can back more Strategies, markets, or order flow without requiring additional locked collateral in pools.

Utility efficiency A measure of how many independent "uses" the same assets can serve at the same time across DeFi. In Aqua, unlocked capital allows tokens to provision liquidity to multiple Strategies while still being usable as money-market collateral, governance or gauge voting power, staking positions, or other protocol-specific utilities, instead of being immobilized inside pool contracts.