

Disclaimer (Cover Page) — v1.1 Technical Specification

This v1.1 “Fiet Technical Specification (Pre-Whitepaper)” is a **historical draft**. **Most of the maths and implementation details in v1.1 should be treated as deprecated**, however the **high-level concepts remain valid**.

What remains valid (high level)

At a conceptual level, the system flow is still:

- **Verified Reserve Liquidity (VRL)** → **Commit** → **Position** (mints LCCs) → **Market** ← **Swaps**

What is deprecated (details)

- **Maths and mechanism details** in v1.1 (formulas, indexing choices, accounting flows, and edge-case handling) have evolved and are **not authoritative**.
- **Contract/module names, call flows, and invariants** may differ materially from what is described in v1.1.

What supersedes v1.1

The `agents/` directory in this repository provides the **current research notes**, implementation reasoning, and evolving specifications that **supersede v1.1**:

- <https://github.com/usherlabs/fiet-protocol/tree/main/agents>

Most of the research notes under `agents/` are intended to be incorporated into the **work-in-progress v1.2 specification**.

Practical reading guidance

- Treat v1.1 as **context**, not as a spec to implement against.
- For up-to-date behaviour and rationale, prefer the `agents/` research notes (and the codebase).



Fiet Technical Specification (Pre-Whitepaper)

Version 1.1 — Ryan Soury, Usher Labs, 20th August 2025

Fiet is a decentralised liquidity commitments protocol that enables institutional market makers to reuse market-neutral, actively-managed reserve liquidity to facilitate markets on automated market makers (AMMs) across blockchains. Held in centralised exchanges, custodial wallets, banks, or traditional finance systems, these reserves are verified via zero-knowledge proofs and committed to AMMs, allowing market-driven pricing algorithms to determine liquidity delivery based on trader demand.

*Pronounced: **fee-yet***, Fiet stands for “Fiat et al.” (Fiat and others), reflecting its mission to bridge liquidity from TradFi and CEXs into decentralised finance (DeFi). This fosters deeply liquid markets on blockchains, delivering enhanced trading experiences, reduced costs, and access to new assets and markets.

Introduction

Foreign exchange fees have yet to be significantly affected by blockchains, primarily due to the capital management constraints of decentralised exchanges (DEXs). Capital must be locked on-chain to enable trades, yet much of the world’s fiat currency remains in bank accounts rather than deployed to blockchains.

Automated Market Makers (AMMs), which are smart contracts facilitating token swaps on decentralised exchanges (DEXs), offer an efficient model for DeFi trading. A blockchain is a distributed ledger that records transactions securely across a network, while DEXs enable peer-to-peer trading without intermediaries. However, AMMs often face liquidity constraints, as locking capital on-chain represents an opportunity cost for market makers, limiting market efficiency and forcing reliance on passive or retail liquidity provision. This illiquidity is particularly pronounced for non-USD stablecoins, real-world assets (RWAs), emerging tokens and markets on emerging blockchains, leading to higher slippage / spreads and trading costs for users.




Fiet addresses this challenge by enabling market makers to commit both on-chain and off-chain liquidity to AMMs without locking funds, allowing dynamic and actively managed reserve liquidity to power DeFi markets. Commitments made today settle tomorrow, facilitating capital efficiencies and deepening market liquidity.

Built initially on Arbitrum, Fiet integrates with existing AMMs like Uniswap, using zero-knowledge (ZK) technology to verify the availability of reserve liquidity and on-chain incentives to ensure reliable settlements. This approach lowers costs for traders and unlocks markets for token issuers. With institutional liquidity from trading firms to support on-chain foreign exchange (FX) markets and emerging tokens with reduced currency risks or opportunity costs, Fiet unlocks efficiency in a \$6 trillion annual sector encompassing crypto-to-crypto trades and cryptocurrency-powered remittances.

Fiet is developed by [Usher Labs](#), leveraging their zkTLS-based cryptography infrastructure to verify financial data sources in a privacy-preserving manner. The protocol operates under a dual license (BUSL-1.1 and GPLv2), [available here](#).

Protocol Mechanics

Fiet's operation involves three core steps:

1.  **Verified Reserves:** Market makers commit liquidity (e.g., \$1M in USD or USDC) from on- or off-chain sources, verified securely using zkTLS proofs to ensure funds are available without compromising privacy.
2.  **Liquidity Commitment Certificates (LCCs):** These commitments are encapsulated as **Liquidity Commitment Certificates** (e.g., lcc-USDC, lcc-USDT), which are non-transferable, protocol-bound instruments. Comparable to a standby letter of credit in traditional finance, LCCs guarantee liquidity and are tradable only within Fiet's integrated DEXs, mirroring the settlement currency (e.g., USDC).
3.  **On-Demand Settlement:** Traders purchase LCCs to access deep liquidity and can redeem them for the underlying asset (e.g., USDC), similar to redeeming a warehouse receipt. If a market maker fails to settle, **Settlement Guarantors** step in, fulfilling obligations and claiming liquidity provider (LP) positions to ensure market continuity.

Key Features

Fiet's approach to dynamic liquidity in DeFi markets includes:

- **Collateralisation:** Market makers provide a small upfront capital contribution in the underlying currency to secure participation and seed markets, ensuring a seamless trading experience.
- **Risk Management:** Settlement mechanisms, guided by Value-to-Signal (VTS) ratios, dynamically adjust liquidity requirements based on market demand, optimising capital efficiency.
- **Regulatory Compliance:** LCCs are non-transferable, protocol-bound units, designed to align with regulatory frameworks and distinct from stablecoins or securities.
- **Preserved DeFi Functionality:** Integrated AMMs retain standard features, including exchange fees for market makers and liquidity providers.
- **Open Participation:** The protocol is accessible to anyone wishing to make markets or trade.
- **Non-Custodial Design:** Users retain full control of their assets, aligning with DeFi's ethos of self-sovereignty.

Learn More

[Why Fiet?](#)

[Participants and Roles](#)

[Concepts](#)

[Cryptography Infrastructure](#)

[FIET — Protocol Native Token](#)

[Glossary](#)

Get Involved

[Join the conversation on Discord](#) to share your interest in Fiet.

Why Fiet?

Cryptographic advancements enable blockchains to track and verify liquidity across diverse sources, identifying its location, ownership, currency denomination, deployment, and availability for rehypothecation (the reuse of collateral for additional financial obligations). This eliminates the need for liquidity to be locked on-chain for immediate use in decentralized finance (DeFi) markets, such as those powered by automated market makers (AMMs).

By verifying the existence and eventual settlement of off-chain liquidity, Fiet orchestrates dynamic liquidity flows between participants, minimising capital lockup and maximising market efficiency. This approach unlocks new opportunities in the cryptocurrency ecosystem.

Context

Fiet addresses inefficiencies in DeFi markets, originating from problems identified in non-USD fiat-backed stablecoins, real-world assets (RWAs), and emerging blockchains. These inefficiencies stem from the requirement to lock liquidity in AMM pools, which imposes significant opportunity costs and risks for market participants.

Challenges with AMM Liquidity

AMMs require liquidity providers (LPs) to lock capital in pools to facilitate trading, constraining capital efficiency. This lockup presents several challenges:

- **Bootstrapping Liquidity:** Attracting sufficient liquidity to AMM pools is challenging without substantial incentives. Passive LPs face trade-offs and opportunity costs when comparing staking opportunities across a growing array of DeFi protocols, reducing their willingness to allocate capital.
- **Impermanent Loss (IL):** AMM mechanics expose LPs to impermanent loss, where price movements rebalance pools unfavourably (e.g., towards stablecoins in bull markets or volatile assets in bear markets), skewing risk-reward dynamics and limiting risk management flexibility.
- **Opportunity Costs for Market Makers:** Institutional market makers (MMs), critical for healthy trading volumes on central limit order book (CLOB) exchanges, are deterred by locked liquidity in AMMs. CLOBs, which match buy and sell orders based on price and time priority, offer greater flexibility for MMs. Off-chain algorithmic trading strategies typically yield higher annual percentage yields (APYs) than AMM liquidity provision, and the lack of asset price-based risk management in AMMs further exacerbates these opportunity costs.

Local Stablecoins

Local stablecoins, pegged to non-USD fiat currencies (e.g., AUD, BRL, IDR), face acute liquidity challenges. Fiat currencies are increasingly used for payments and debt rather than investment, given their inflationary nature compared to hard assets like real estate or Bitcoin. Locking these currencies in AMM pools is unattractive, as yields are often offset by inflation. Consequently, local stablecoin markets suffer from shallow liquidity, high slippage, and elevated trading costs. This limits their use in DeFi and hinders cost-efficient on-chain foreign exchange (FX) or remittance, which could compete with traditional inter-bank FX rates if liquidity were accessible permissionlessly.

Real-World Assets (RWAs)

Tokenised RWAs, such as digitised real estate or commodities, face conflicting incentives. While tokenisation maximises capitalisation, integrating RWAs into DeFi requires liquid AMM markets to enable functions like collateralisation in lending protocols. However, bootstrapping AMM liquidity requires LPs to fractionalise their RWA investments, reducing overall capitalisation and deterring participation.

Emerging Blockchains

Emerging blockchains face a "chicken-and-egg" problem: DeFi requires total value locked (TVL) in AMM pools to ensure low-slippage trading, but attracting TVL is challenging without established markets. High slippage costs deter traders, stalling DeFi adoption on these chains.

Epiphany

The most efficient way to integrate off-chain liquidity into on-chain decentralised finance (DeFi) markets is through tokenisation, the process of representing real-world assets as digital tokens on a blockchain. This is evident in the widespread adoption of fiat-backed stablecoins, such as Tether (USDT) and Circle (USDC), which exceed \$200 billion in market valuation, and asset-backed stablecoins, like those from Ethena Labs, which leverage government bonds or yield-bearing, risk-adjusted strategies.

However, a common misconception is that on-chain liquidity representation requires *formal* tokenisation — issuing freely transferable cryptocurrencies that directly represent ownership of underlying assets. This is not the case. By verifying and tracking off-chain liquidity — such as funds in bank accounts or centralised exchange reserves — using cryptographic proofs (e.g., zero-knowledge proofs), DeFi protocols can dynamically adjust market mechanics based on committed liquidity without immediate on-chain settlement or formal token issuance. Within AMMs, this committed liquidity is termed “virtual” liquidity.

This approach enables DeFi to reflect real-world liquidity dynamics, akin to how forward contracts in traditional finance (TradFi) commit to future delivery without immediate asset transfer. By bridging off-chain and on-chain systems, Fiet redefines DeFi’s interaction with global liquidity, fostering deeper, more efficient markets.

Innovation

Fiet redefines DeFi liquidity by enabling AMMs to account for “virtual” liquidity — off-chain reserves verified but not locked on-chain. Using zero-knowledge proofs (i.e., zkTLS), Fiet verifies liquidity held in centralised exchanges, custodial wallets, banks, or other TradFi systems, allowing market makers to commit these reserves to AMMs without immediate capital lockup. This is akin to an oracle, a system that feeds external data to blockchains, but tailored for liquidity verification.

Fiet’s key innovations include:

- **Liquidity Commitments:** Market makers commit verified reserves, which AMMs use to facilitate trading based on market demand, settling only when required. This minimises opportunity costs and impermanent loss.
- **Collateral and Incentives:** Small upfront collateral ensures market maker participation, while settlement guarantors seize collateral if MMs fail to deliver, ensuring trust and continuity.
- **Regulatory Alignment:** Liquidity Commitment Certificates (LCCs), non-transferable protocol-bound instruments, facilitate trading without constituting securities, aligning with regulatory frameworks.

Use Cases

On-Chain Foreign Exchange (FX)

Consider an AMM pool pairing a local stablecoin, NGNC (pegged to the Nigerian Naira), with USDC (pegged to USD). Traditional AMM mechanics result in imbalanced liquidity, as demand for USDC exceeds NGNC due to USD’s global reserve status and NGN’s inflationary pressures. This leads to high slippage and costly trading. With Fiet:

- Market makers commit USD-denominated reserves (verified via zkTLS) to the NGNC/USDC pool without locking NGNC.
- The AMM prices trades based on market demand, and MMs settle NGNC only when traders demand it (e.g., via OTC conversion from USD to NGN and transfer to an NGNC issuer).
- This enables low-cost, on-chain FX for local stablecoins (e.g., AUD, PHP, CAD), competing with traditional FX systems.

DEX Token Listings

For an AMM pool pairing USDC with an RWA (e.g., tokenised real estate, HOUSE), Fiet allows RWA issuers to lend assets to institutional MMs, who commit their USDC reserves to facilitate the market. This eliminates the need for LPs to fractionalise RWA investments, preserving capitalisation while enabling DeFi integration.

Lower-Cost High-Volume Markets

Consider an AMM pool pairing USDC and ETH on Arbitrum, two highly sought-after assets due to their widespread integration across DeFi protocols. Unlike traditional AMMs, which require liquidity providers to lock capital, Fiet enables market makers to commit the same verified reserve liquidity to multiple markets simultaneously, provided their collateralisation and settlement obligations are met. This capital efficiency allows MMs to earn fees across various markets, even with lower per-market fees, as the same liquidity supports multiple trading pairs.

Traders benefit from reduced fees, as they no longer compensate LPs for capital lockup, resulting in deeper, lower-cost markets for high-volume assets like USDC/ETH.

Cross-Chain DeFi

Emerging blockchains can host liquid AMM markets without requiring immediate TVL. Fiet’s virtual liquidity enables low-slippage trading, fostering DeFi adoption and supporting new assets like RWAs in cross-chain ecosystems.

Outcomes

Fiet bridges TradFi and DeFi, coupling cryptography and smart contracts with regulated partners (e.g., stablecoin issuers) and institutional MMs. Key outcomes include:

- **DeFi Yield on Reserves:** MMs earn DeFi yield by rehypothecating reserves held in CEXs, banks, or brokerage accounts.
- **Cost-Efficient FX:** Fiet markets enable low-cost exchange for local stablecoins, capturing a share of the \$903 billion annual remittance-based FX market (compared to crypto's \$810 million).
- **On/Off-Ramps:** Local stablecoin issuers become DeFi-powered on/off-ramps, offering cheaper USD access than traditional services.
- **New Markets:** RWAs and emerging assets integrate into DeFi without liquidity bootstrapping overhead.
- **Cross-Chain Liquidity:** Blockchains host liquid markets without locked TVL, enhancing DeFi accessibility.

Fiet's technology enables DeFi markets to dynamically reflect real-world liquidity, delivering deeper markets and new financial opportunities.

Participants and Roles

Market Makers (MMs)

In the Fiet Protocol, market makers (MMs) enable dynamic liquidity provision for AMMs by committing actively managed reserve liquidity, rather than locking capital as required in traditional AMM models.

MMs in Fiet perform a role similar to market makers on central limit order book (CLOB) exchanges, which facilitate trading by maintaining bid and ask orders. However, instead of building an order stack, Fiet MMs undertake the following steps:

1. **Connect Financial Accounts:** MMs link their off-chain or on-chain accounts (e.g., centralised exchanges, custodial wallets, or banks) to the protocol.
2. **Signal and Verify Reserves:** Using cryptographic zero-knowledge proofs (e.g., zkTLS), MMs verify the availability of their reserve liquidity, ensuring trust and privacy.
3. **Commit Liquidity to AMMs:** MMs commit verified reserves to an AMM market, enabling trading without capital lockup.
4. **Settle Based on Demand:** MMs deliver liquidity to the AMM's smart contracts when trader demand for one side of the market increases.
5. **Withdraw Excess Liquidity:** As demand for the counterparty asset decreases, MMs can withdraw liquidity proportional to their committed position, preserving flexibility.

While MMs can employ strategies to optimise liquidity distribution within the AMM to maximise trading fee yield, their control is less granular than in CLOB-based exchanges. Nevertheless, Fiet allows MMs to retain active management of their reserves, settling liquidity only in response to on-chain market demand, thus minimising opportunity costs and impermanent loss.

MMs can also execute direct market or limit orders (swaps) within the AMM to exploit arbitrage or order flow opportunities, enhancing their yield potential.

Integration Partners

Integration Partners, or Integrators, are third-party systems and interfaces that facilitate interactions between traders (retail users) and the Fiet Protocol.

Fiet provides dynamic liquidity for automated market makers (AMMs) by integrating at the token layer. Traders interact with Fiet markets via a wrap/unwrap mechanism, where committed liquidity is wrapped into non-transferable Liquidity Commitment Certificates (LCCs). Based on the integrated DEX protocol and available on-chain liquidity, LCCs may be unwrapped immediately for the underlying cryptocurrency (e.g., USDC) or held and manually redeemed, akin to a TradFi financial instrument.

Integration Partners use Fiet's libraries to abstract this wrap/unwrap process, enabling token swaps (e.g., from token A to token B) without exposing traders to protocol mechanics. Potential Integration Partners include:

1. **Fintechs:** Platforms incorporating foreign exchange (FX) capabilities.
2. **Wallets:** Offering direct access to Fiet's DeFi markets.
3. **On/Off-Ramps:** Services converting fiat to cryptocurrency with reduced costs.
4. **DeFi Interfaces and Aggregators:** DEXs or platforms like 1inch integrating Fiet for cost-effective order routing.

Fiet's integration libraries support fiat-to-crypto swaps through interactions with stablecoin issuers. These abstractions provide compliant onboarding, enabling users to convert non-USD fiat currency (e.g., via bank transfer) into stablecoins like USDC at low cost. Stablecoin issuers function as prisms, consolidating and distributing fiat liquidity among traders, market makers (MMs), and AMMs, which ensure accurate token exchange accounting.

Integration Partners facilitate liquidity flows between traders, stablecoin issuers, and MMs, while MMs manage liquidity based on their strategies.

Traders

Traders are users who interact with markets through interfaces to execute market orders (swaps).

Traders can perform the following actions:

- **Crypto-to-Crypto Swap:** Deposit token A to receive token B, executed within the AMM.

- **On-Ramp (Fiat to USDC):** Deposit fiat with a stablecoin issuer to receive a local stablecoin, then swap it for USDC in the AMM, following compliant onboarding processes.
- **Off-Ramp (USDC to Fiat):** Send USDC to the AMM to receive a local stablecoin, which is redeemed for fiat via the stablecoin issuer.

Direct Liquidity Providers (LPs)

Direct Liquidity Providers (LPs) are users who supply liquidity to automated market maker (AMM) markets by locking capital, following the traditional AMM model. Fiet permits any user to act as a Direct LP, maintaining open access to liquidity provision. However, Direct LPs face increased competition for yield, as they operate alongside institutional MMs who leverage verified reserve liquidity without capital lockup.

Guarantors

Settlement Guarantors are entities, including market makers (MMs) or automated bots, that monitor AMM market states and value-to-signal (VTS) ratios to identify failing MMs. After a Request for Settlement grace period expires, incentives activate for Guarantors to settle on behalf of failing MMs, akin to a clearinghouse in traditional finance (TradFi) ensuring transaction completion.

In exchange for settling the required amount, Guarantors acquire all or part of the MM's liquidity position, including settled liquidity or at least the MM's collateral committed to the market. Non-MM Guarantors are expected to liquidate the seized position immediately to realise profit. Guarantors may also compensate Provers for attestations to reserve liquidity sources, enabling early detection of an MM's overcommitted reserves relative to their signalled liquidity.

Provers

Provers are entities that assist market makers (MMs) in generating cryptographic proofs to verify the correctness and availability of their reserve liquidity. Fiet leverages Usher Labs' Verity zkTLS stack for this purpose. MMs select a trusted Prover to share their data with, compensating them in FIET tokens, or opt to act as their own Prover to maintain full data privacy.

At launch, Usher Labs will serve as the sole Prover, with additional Provers introduced as MM demand and protocol decentralisation evolve.

Concepts

Markets

Verified Reserve Liquidity

Liquidity Commitment Certificates (LCCs)

Value-to-Signal Model

Oracle

Settlements

Rewards

Markets

A Fiet Market extends existing decentralised finance (DeFi) markets, such as those built on Uniswap, by enabling market makers (MMs) to commit dynamic, actively managed reserve liquidity to facilitate trading. Fiet integrates with automated market maker (AMM) protocols at the token layer. This integration allows immediate liquidity commitments, with settlements triggered only when market demand for a specific token increases.

Fiet Markets inherit AMM accounting principles. Demand for one token (e.g., ETH in a USDC/ETH pool) increases the supply of the counterparty token (e.g., USDC). This enables MMs to settle the high-demand token and withdraw excess liquidity from the low-demand token, minimising capital lockup. Each market is isolated, immutable, and persists as long as its blockchain remains active. Such design ensures contained risks and predictable behaviour. Market creation is permissionless.

Accounting Model

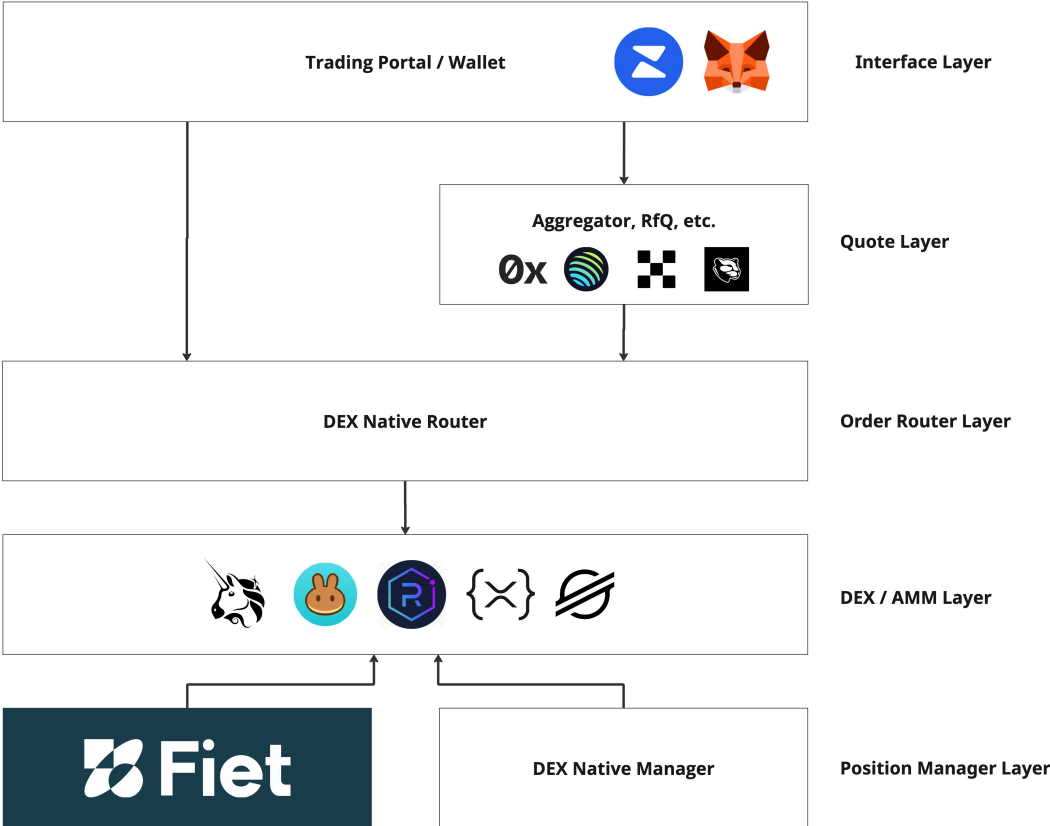
Fiet inherits the integrated AMM price curve. Most AMMs adopt or are based on the constant product price curve, defined as $x \cdot y = k$, where x and y represent the quantities of two assets in the pool, and k is a constant.

AMM price curves ensure symmetry between assets, with slippage (price impact) determined by the pool's liquidity depth relative to trade size. For example, a \$100,000 swap in a \$200,000 pool (50/50 balance) shifts the pool to a 75/25 ratio, increasing slippage. In contrast, a \$1,000,000 pool depth maintains lower slippage for the same trade.

Fiet extends these models by incorporating virtual liquidity in the form of synthetic assets — Liquidity Commitment Certificates (LCCs). These encapsulate both settled on-chain liquidity and verified reserve liquidity (VRL). Unlike traditional AMMs, where all liquidity is locked, Fiet treats VRL as available for trading. Verification occurs via zero-knowledge proofs (e.g., zkTLS), backed by MM collateral. This enables deeper liquidity pools, reducing slippage for traders. Arbitrage incentives rebalance markets to maintain equilibrium, while collateral requirements guarantee eventual settlement of VRL. Settlements align with the position-specific Value-to-Signal (VTS) model, ensuring demand-driven adjustments per token and range.

DEX Integration

From a technical perspective, the Fiet Protocol resides at the liquidity provision layer in the market stack. It serves as a position management facility within decentralised exchanges (DEXs).



Traders engaging the order routing layer (via Uniswap, PancakeSwap, or other DEX aggregators or Request for Quote (RfQ) technologies) may seamlessly interface with Fiet Markets. Deep integration enables order routers to recognise unique Fiet functionality, such as LCCs. Without such integration, default behaviour

results in the order routing layer recognising only settled on-chain liquidity. Fiet Markets will still receive routed orders in this case, but not at a volume that leverages the full capability of Fiet.

To assist in this integration, Fiet provides libraries and associated components that frontends and smart contracts can use to directly interface with Fiet Markets.

Market Chain

Fiet operates across multiple blockchains, starting with Arbitrum One. Each blockchain hosting Fiet Markets is termed a Market Chain. This isolates market management to ensure scalability and independence.

Market Parameters

Markets are defined by the following parameters, configured at deployment and immutable thereafter:

Parameter	Description	Configurability
Market Pair	Consists of two distinct crypto assets (e.g., USDC/AUDD) compatible with the underlying AMM protocol. On EVM blockchains, these are ERC20-compliant assets. These are wrapped as Liquidity Commitment Certificates (LCCs) within the AMM pool, forming a market pair such as lcc-USDC/lcc-AUDD.	Permissionless, set at creation.
Proxy Pool	A reference pool, supported by AMMs with hook or extension mechanics (e.g., Uniswap v4, PancakeSwap), that proxies trades from a standard asset pair (e.g., USDC/AUDD) to the core LCC-based pool (e.g., lcc-USDC/lcc-AUDD). This automates the wrap/unwrap process of LCCs, enabling integration with DeFi interfaces, aggregators, and other order routing technologies.	Enabled for hook-compatible AMMs; automatic if supported.
Base Value-to-Signal (VTS) Target Rate	Specifies the minimum collateral market makers (MMs) must settle upfront to commit Verified Reserve Liquidity (VRL). Collateralisation seeds the market for small trades and incentivises settlement guarantees by ensuring MMs have capital at risk.	Per-token, set at deployment (e.g., 0.02 for USDC).
Value-to-Signal (VTS) Parameters	The VTS model involves parameters immutable after market deployment that impact the emphasis of market demand projections, and the rate at which these projections decay over time, recognised as the scaling parameter α and the decay rate λ .	Immutable post-deployment; defaults as per protocol (e.g., $\alpha = 1.5$, $\lambda \approx 0.0001927$).
Settlement Grace Period & Maximum Time	Establishes a period where MMs are initially safe from collateralised liquidity position seizure. Once the grace period ends, a second period gradually unlocks their positions for seizure until it's completely available.	Per-token, set at deployment (e.g., grace 24 hours, max 3600 seconds).
Seizure Alpha	The sensitivity parameter α in the seizure formula, controlling the impact of an MM's RfS exposure on the seizure amount. This parameter escalates seizure for higher exposures, balancing intervention incentives with MM fairness.	Set at deployment (e.g., 1.5); immutable thereafter.

Minimising Impermanent Loss

Impermanent loss occurs when the value of deposited tokens in an AMM position diverges from holding them outside the pool due to price fluctuations. In Fiet Markets, MMs mitigate this through dynamic settlements via the Value-to-Signal (VTS) model, which adjusts obligations per position and token based on demand.

Since settlements are triggered only for high-demand tokens in active or soon-to-be active positions, MMs avoid locking capital across the full range. For instance, in a USDC/ETH market with volatile ETH prices, MMs settle USDC when demand rises (e.g., bearish ETH dumps), withdrawing excess ETH equivalents as the position rebalances. This reduces exposure to price shifts within ranges, as commitments remain virtual until needed.

The protocol's position-specific settlement formula enables withdrawals that allow MMs to reallocate capital actively. Combined with Verified Reserve Liquidity (VRL), this approach limits impermanent loss to settled portions only, which includes collateral (base VTS rate). Direct LPs, settling fully upfront, face standard AMM impermanent loss, but Fiet's model offers MMs a more capital-efficient alternative.

For example, an MM committing \$1,000,000 to a USDC/ETH market may settle 2% (\$20,000) on-chain as lcc-USDC/lcc-ETH, exposed to impermanent loss (IL). The remaining \$980,000 VRL, held as USD, USDC or ETH, is managed dynamically, free from IL.

Verified Reserve Liquidity

Verified Reserve Liquidity (VRL) is liquidity cryptographically verified as available to the Fiet Protocol but held outside its control. Reserve liquidity can reside off-chain in bank accounts, or on-chain in custodial wallets. It can even be actively managed and deployed in centralised exchanges (CEXs). VRL enables market makers (MMs) to commit dynamic liquidity to Fiet Markets without immediate on-chain settlement.

Liquidity Signals

Verified Reserve Liquidity (VRL) comprises liquidity signals, each representing the state of a market maker's (MM's) reserve liquidity sources. MMs specify an amount of reserve liquidity to make available to the Fiet Protocol, using cryptographic zero-knowledge proofs (zkTLS) to verify solvency and availability while preserving privacy.

To generate signals, MMs must:

1. Connect financial data sources (e.g., centralised exchanges, bank accounts, or custodial wallets) via APIs.
2. Produce zkTLS proofs confirming:
 - The location of the liquidity.
 - The asset denomination.
 - Solvency for the specified amount.
3. Engage a Prover (e.g., Usher Labs) to facilitate proof generation based on shared data, or act as their own Prover to maintain data privacy. Only the verified parameters (e.g., liquidity amount) are recorded on-chain.

Verified signals enable MMs to commit liquidity to Fiet Markets. The VRL state is maintained in a verifiable compute environment with cryptographic data portability for validation across Market Chains. This source of truth for the verified state is read by MMs or authorised parties to produce transactions sent to the `SpokeReceiver` contract on a Market Chain. This mechanic is necessary to securely commit the signal to Fiet Markets.

Fiet Markets then account for the total liquidity using the AMM accounting model. As reserve liquidity is settled to committed markets, the VRL amount in the source of truth may decrease, but the receiving market recognises the sum of settled and reserve liquidity. Market making across multiple Fiet Markets simultaneously (rehypothecating liquidity) requires MMs to manage solvency and signal uptime carefully for each committed market.

Liquidity Sources

Liquidity sources are financial data sources where market makers' (MMs') Verified Reserve Liquidity (VRL) resides. These sources include:

- **Centralised Exchanges:** Platforms such as Binance, Kraken, or CoinJar holding cryptocurrency reserves.
- **Wallets (EOA):** Externally owned accounts, such as Ethereum wallet addresses, containing on-chain funds.
- **Brokerages:** Services like Robinhood or CMC Markets managing investment accounts.
- **Bank Accounts:** Regulated financial institutions holding fiat currency.
- **Stablecoin Issuers:** Licensed entities, such as Circle, that issue fiat-backed cryptocurrencies, receive or remit fiat, and custody liquidity. Some issuers act as custodians, while others function as over-the-counter (OTC) liquidity providers.
- **Custodians:** Providers like Finoa or Fireblocks offering wallet addresses, which may distinguish individual users within their platforms or use shared addresses.
- **Smart Contracts:** Smart and/or abstracted wallets such as Safe, or other multi-signature arrangements, where ownership is verifiable and the contract serves custodial purposes.

The Fiet Protocol verifies the solvency and availability of liquidity in these sources using zero-knowledge proofs (zkTLS), ensuring MMs' commitments to Fiet Markets are backed by accessible reserves.

Signal Uptime

Verified liquidity signals must be maintained to prevent expiration, which can lead to the seizure of settled assets in Fiet Markets. Market makers (MMs) ensure signal uptime through recurring proof generation.

Fiet Protocol will deploy with an initial expiry time of 60 minutes.

Within this expiry window, MMs and Settlement Guarantors can engage Provers to:

1. Access a privacy-preserving data stream of attestations to liquidity sources for all MMs.
2. Advance the Verified Reserve Liquidity (VRL) state via zero-knowledge proofs to syndicate the latest VRL state to relevant Market Chains.

These actions are taken to detect insolvency and pending MM failure. Expedition of penalties subsequently leads to rewards.

If the signal shows reduced available liquidity but commitments remain within the total signalled amount, no action is required. Committed amounts to markets must remain within the total liquidity signal.

If the committed amount's value exceeds the total signalled liquidity, indicating the MM is no longer solvent, or the signal expires due to source data retrieval failures, three outcomes are possible:

1. The MM identifies this discrepancy and reveals additional reserve liquidity to ensure solvency.
2. A Settlement Guarantor (e.g., another MM) requests an expedited solvency check and VRL state advancement from the Prover. If the discrepancy persists, the Guarantor is allocated the seized liquidity position proportional to the difference between the MM's commitment and signalled value.
3. The MM's liquidity position is eventually seized if solvency is not restored.

MM solvency is verified during commitment and decommitment from a Fiet Market. If the MM is insolvent at decommitment, their liquidity position can be seized.

MMs are advised to maintain a 10% buffer between signalled and committed liquidity, particularly when reserve currencies differ from market currencies, to account for price fluctuations, fees, and spreads during settlement.

Technicalities

The Fiet Protocol employs cryptographic technologies to verify and manage Verified Reserve Liquidity (VRL). The process is structured as follows:

```
use serde::{Deserialize, Serialize};
use std::collections::BTreeMap;

#[derive(Serialize, Deserialize, Clone, CandidType)]
pub struct MMState {
    total: (String, u64), // Tuple of total reserve denominated in (e.g., "USD") to amount
    reserves: BTreeMap<String, u64>, // Asset ID (e.g., "ETH", "SOL") to amount
    sources: [String], // Hash array of sources and ownership correlation
    prover: String, // Address of Prover for this MMState
    nonce: u64, // For state freshness
}

#[derive(Serialize, Deserialize, Default)]
pub struct State {
    market_makers: BTreeMap<String, MMState>, // MM ID to State
    current_root: [u8; 32], // Latest Merkle root
}
```

1. **zkTLS Proof Generation:** Usher Labs' Verity zkTLS stack verifies financial data sources:
 - RiscZero's zkVM rolls up and verifies private facets of zkTLS proofs.
 - The Internet Computer (IC) rolls up and verifies public facets, cross-referencing RiscZero STARK proof verification. The IC, a replicated state machine, supports advanced computation and cryptographic data portability via multi-party computation (MPC) threshold signatures.
2. **VRL State Advancement:** Each liquidity signal rollup produces a new VRL state, represented as a Merkle tree of MMState.
3. **State Verification:** The IC's Verifier validates the VRL state, storing it with a threshold-ECDSA signature of the Merkle tree root hash.

4. **Cross-Chain Validation:** The verified VRL state is validated on Market Chains capable of ECDSA and Merkle proof verification, enabling MMs to commit liquidity to Fiet Markets.
5. **Market Commitment:** Committing VRL requires collateralisation per the Value-to-Signal Model, ensuring markets recognise the total liquidity.

This framework ensures privacy-preserving verification of VRL, with cryptographic integrity maintained across Market Chains.

Rehypothecation

Verified Reserve Liquidity (VRL) can be dynamic, comprising highly liquid assets such as cryptocurrencies, fiat currencies, or bonds that are actively traded on centralised exchanges or held in bank accounts, mutual funds, or cash management accounts.

Market makers (MMs) can rehypothecate VRL by committing the same liquidity to multiple Fiet Markets or other platforms (e.g., Binance order stacks) simultaneously. When market demand for a settlement token exceeds the Value-to-Signal (VTS) target, MMs must settle liquidity for that token. Conversely, as demand for the counterparty asset decreases, MMs can withdraw excess liquidity. This mechanism enables MMs to maintain high-velocity liquidity across markets, requiring active management of solvency and signal uptime to meet settlement obligations.

Liquidity Commitment Certificates (LCCs)

Market makers (MMs) partition their Verified Reserve Liquidity (VRL) into commitments to Fiet Markets across blockchains. These commitments are represented as Liquidity Commitment Certificates (LCCs).

What is a Liquidity Commitment Certificate?

Liquidity Commitment Certificates (LCCs) are synthetic assets in the Fiet Protocol that represent settled in-scope liquidity and out-of-scope Verified Reserve Liquidity (VRL) committed to Fiet Markets.

Verified via zero-knowledge proofs (zkTLS), LCCs ensure:

1. Traders can engage the decentralised exchange (DEX) with live price action that accounts for virtual liquidity.
2. MMs can deliver settlement tokens (e.g., USDC, ETH) held in reserves, such as bank accounts or centralised exchanges, without immediate on-chain lockup.

LCCs are non-transferable, protocol-bound assets traded exclusively on Fiet's integrated DEX, mirroring the settlement token. Traders can exercise LCCs to redeem the underlying token. The Value-to-Signal (VTS) ratio for each token in a position dynamically adjusts collateral requirements, ensuring MMs maintain on-chain capital to support settlement obligations. This design distinguishes LCCs from stablecoins or securities, aligning with regulatory compliance.

Traditional Finance Analogy

Fiet's **Liquidity Commitment Certificates (LCCs)** are like bank-backed guarantees, ensuring verified liquidity with the trust of a letter of credit, but in a decentralised, transparent system.

Imagine a **LCC** as a hybrid of a **standby letter of credit** and a **tradeable warehouse receipt** used in commodity markets. In traditional finance, a standby letter of credit is a bank's guarantee that a seller can deliver goods or funds when needed, providing confidence to buyers without immediately moving the assets. Similarly, our protocol allows **Market Makers** to commit liquidity, like US dollars or crypto, held in verified reserves, such as bank accounts or centralised exchange balances. Using **zkTLS proofs**, we verify these reserves on-chain, ensuring the liquidity is real without requiring it to move upfront.

This commitment is wrapped into a **certificate** — think of it like a warehouse receipt that proves ownership of goods, like grain or oil, stored elsewhere. In our case, the 'goods' are the liquidity (e.g., USDC or ETH). Traders can buy and sell these certificates on integrated decentralised exchanges, much like warehouse receipts are traded in commodity markets, allowing them to access deeper liquidity pools without the funds being locked on-chain.

When the market demands it — say, during a surge in trading — the certificate can be 'exercised' to deliver the actual asset, like redeeming a warehouse receipt for physical goods. But unlike a typical token or stablecoin, these certificates are **locked to our platform**, ensuring they're used only for trading and settlement within our ecosystem, reducing risks of misclassification. Our **Value-to-Signal ratio** acts like a dynamic collateral requirement, ensuring providers always have some skin in the game on-chain, while flexibly adjusting to market needs.

In short, our Liquidity Commitment Certificate lets traders tap into massive, verified liquidity pools with the confidence of a bank-backed guarantee, but with the flexibility and efficiency of a decentralised exchange.

Example

A market maker (MM) commits \$1,000,000 in USD-denominated Verified Reserve Liquidity (VRL) to a USDC/ETH Fiet Market. The market requires 2% collateral (\$20,000), allocated between USDC and ETH according to the automated market makers' (AMM) current state and liquidity mathematics, ensuring a balanced initial position. This commitment is represented by Liquidity Commitment Certificates (LCCs), lcc-USDC and lcc-ETH, which encapsulate the settled collateral (\$20,000) and the remaining VRL (\$980,000) verified via zero-knowledge proofs (zkTLS), enabling trading on the Fiet Market's AMM with settlements triggered by market demand.

Key Features

1. **ERC20 Compliance:** LCCs adhere to the ERC20 standard, enabling compatibility with Ethereum-based AMMs.
2. **Non-Transferable:** LCCs are protocol-bound bookkeeping units, restricted to Fiet's integrated decentralised exchange (DEX), ensuring controlled liquidity commitments.

3. **Permissionless Creation:** LCCs can be created by any MM committing Verified Reserve Liquidity (VRL) to a Fiet Market.
4. **Fungibility:** LCCs are fungible for the same settlement token (e.g., all USDC-based LCCs are lcc-USDC), facilitating uniform trading within the DEX.
5. **Regulatory Alignment:** LCCs are designed as non-transferable assets distinct from stablecoins or securities, complying with regulatory frameworks.

Parameters

LCCs in the Fiet Protocol are defined by deployment parameters set at market creation and order parameters passed during trades to the Proxy Pool. Deployment parameters are immutable and configure the core functionality, while order parameters allow dynamic behaviour for specific trades, such as handling excess LCCs.

Deployment Parameters

1. **DEX Pool Manager:** References the smart contract(s) managing liquidity flow in the AMM. LCCs use this address to evaluate inflows to or outflows from the DEX. These addresses are whitelisted, permitting LCC settlements to these contracts and preventing user-to-user transfers.
2. **Settlement Token:** Specifies the ERC20 token address mirrored by the LCC (e.g., USDC on Arbitrum at `0xaf88d065e77c8cc2239327c5edb3a432268e5831`).
3. **Oracle:** Maps liquidity signal reserve tickers (e.g., "usd") to smart contracts implementing the [IOracle interface](#) adopted from the Morpho Blue protocol. These contracts evaluate the price of the signal reserve currency relative to the settlement token.

Order Parameters

LCCs may change behaviour based on parameters passed along with market trade orders specifically to Proxy Pools. Proxy Pools pair native assets (e.g., ARB/USDT) and proxy orders to a fixed LCC-based core pool (e.g., lcc-ARB/lcc-USDT). These parameters change how LCCs are managed within the order logic. If these parameters are not present, then the Proxy Pool will restrict the available trade size to the amount of liquidity available and settled to the market. For example, an order of \$50,000 USDT → ARB, without parameters present, against a market where only \$20,000 ARB in USD value is realised (settled on-chain), will restrict the trade size to \$20,000. This ensures smooth predictable behaviour for traders engaging the Proxy Pool without awareness of underlying Fiet functionality. However, if these parameters are present, it is presumed the trader or integrating smart contracts are aware of Fiet protocol logic, and therefore the trade size is not restricted and instead the full order is fulfilled. Excess LCCs received from the trade that cannot be immediately unwrapped for native assets will be handled as per the provided parameters.

Recipient: A `recipient` address indicates to the Proxy Pool where to transfer LCCs received. Occurs specifically when the market is incapable of immediate unwrap and settlement, due to insufficient atomic liquidity. Allows the recipient to receive whatever available liquidity there is, and LCCs as excess. Any excess LCC received will be automatically replaced by underlying settlement tokens once MMs settle accordingly in a future blockchain transaction.

Constraints

For Traders

To engage a Fiet Market (e.g., USDC/ETH), traders must wrap assets into LCCs, as the core AMM pool is an lcc-USDC/lcc-ETH pair. Alongside the core LCC-based pool, Fiet incorporates mechanics and integration with DEXs to abstract this, presenting the market as a USDC/ETH pair that routes trades to the core lcc-USDC/lcc-ETH pool. In extensible AMMs such as Uniswap v4, this is conducted with corresponding proxy pools.

Traders can only:

1. Use LCCs on Fiet's integrated DEX, mirroring the settlement token.
2. Exercise LCCs to redeem the underlying settlement token.

LCCs cannot be transferred between wallets or users. This constraint ensures regulatory compliance by distinguishing LCCs from stablecoins or other crypto assets and enables accurate Value-to-Signal (VTS) ratio tracking by differentiating traders' DEX liquidity from wrapped liquidity. VTS ratios are tracked per token at the position level, not within LCCs.

For Market Makers

Market makers (MMs) committing VRL to a Fiet Market will simultaneously and automatically create a liquidity position in the AMM. VRL cannot be committed to generate LCCs without this position, preventing arbitrary

LCC management outside market operations. The split of this commitment between the paired tokens is determined by the underlying AMM's liquidity mathematics, which calculates the allocation based on the current market price and pool state, ensuring alignment with the protocol's trading dynamics.

Fiet's smart contracts hold AMM liquidity position receipts on behalf of MMs, issuing a transferable receipt non-fungible token (NFT) to the MM, referencing their commitment. This NFT allows flexibility in managing position parameters and settlement obligations across wallets. Fiet proxies AMM liquidity position management functions, retaining all default configurability.

To decommit from a market, MMs:

1. Burn the Fiet Market Position NFT.
2. Liquidate the AMM liquidity position via the protocol.
3. Withdraw tokens, including fees, from the protocol.
4. Drop the VRL signal.
5. Update VRL state across Market Chains with a dropped signal message, incurring no penalties if no commitments remain.

MMs cannot decommit liquidity subject to an open Request for Settlement (RfS). Such liquidity and associated collateral remain locked until settled by the MM or a Settlement Guarantor.

Settlement Queue

As LCCs are received from Fiet Markets, a condition applies to this subset amount of LCC, which is whether it is placed into a settlement queue.

The settlement queue addresses scenarios where immediate unwrapping of LCCs to their underlying settlement tokens is not possible due to insufficient settled liquidity at the time of a trade. This mechanism is necessary to maintain traceability and fairness in liquidity allocation, ensuring that traders receive the tokens they are entitled to without mixing settlements from unrelated sources. It solves the challenge of handling pending obligations in a decentralised system, where market makers' settlements may lag behind trader demand, while preventing disruptions to trading flow.

When a trader swaps directly with the LCC-based Core Pool, the inflow token's underlying liquidity moves "in-market" to support the pool. For the outflow token, the protocol attempts to allocate settled liquidity to the LCC for unwrapping. If insufficient, the shortfall is recorded as pending, queued chronologically for resolution. Pending items are traced to specific markets and users, based on how the LCC was acquired (e.g., from a particular swap), to ensure settlements are directed appropriately — traders acquiring LCCs from a market expect liquidity tied to that market's activity.

Market makers' settlements clear the queue, prioritising outstanding pending items before allocating excess to the market. Traders with Direct Liquidity Provider (LP) positions can unwrap immediately, as their interactions do not accrue pending items. If a trader takes further action with queued LCCs (e.g., in another swap), the pending item is cleared to reflect the updated state. This queue enables full order fulfilment without restricting trade sizes, abstracting complexity for traders while upholding protocol integrity.

Compared with Leverage or Margin?

Fiet's Liquidity Commitment Certificates (LCCs) may appear akin to leveraged positions, as committing a small collateral (e.g., 2% or \$20,000 USDC) facilitates deployment of \$1,000,000 in liquidity as lcc-USDC. However, LCCs involve neither leverage nor margin, as the protocol entails no borrowing, debt creation, or amplified exposure to price movements. Instead, LCCs encapsulate Verified Reserve Liquidity (VRL) owned by the market maker, attested via zkTLS proofs.

Distinctions include:

- **Absence of Borrowing:** Unlike perpetual futures platforms, where collateral (e.g., \$2,000) supports borrowing \$98,000 for 50x leverage, Fiet requires no loans. MMs facilitate liquidity from their verified reserves, without third-party funding.
- **Operational Risk Focus:** Seizure arises from failure to settle, an operational lapse, rather than price-driven liquidations. While price fluctuations can indirectly contribute — if a mismatch exists between the signal reserve currency (e.g., USD) and settlement token (e.g., ETH), such as an ETH price spike rendering the committed amount insolvent relative to the signalled value — this triggers seizure only through resultant non-delivery, not automated margin calls.
- **Protocol-Bound Synthetic:** lcc-USDC functions as a non-transferable token bound to Fiet's ecosystem, backed by attested reserves, not a derivative position betting on asset prices.

- Collateral Function: The initial base Value-to-Signal rate (eg., 2%) serves to anchor the commitment and incentivise guarantors, not as margin against borrowed funds.
- Demand-Driven Settlement: MMs deliver tokens from reserves in response to trader activity, without repayment obligations. Settlement Guarantors intervene for non-fulfilment, claiming proportional position shares, distinct from closing leveraged trades.

This framework supports substantial liquidity facilitation without introducing debt or speculative amplification, prioritising verified commitments over financial gearing.

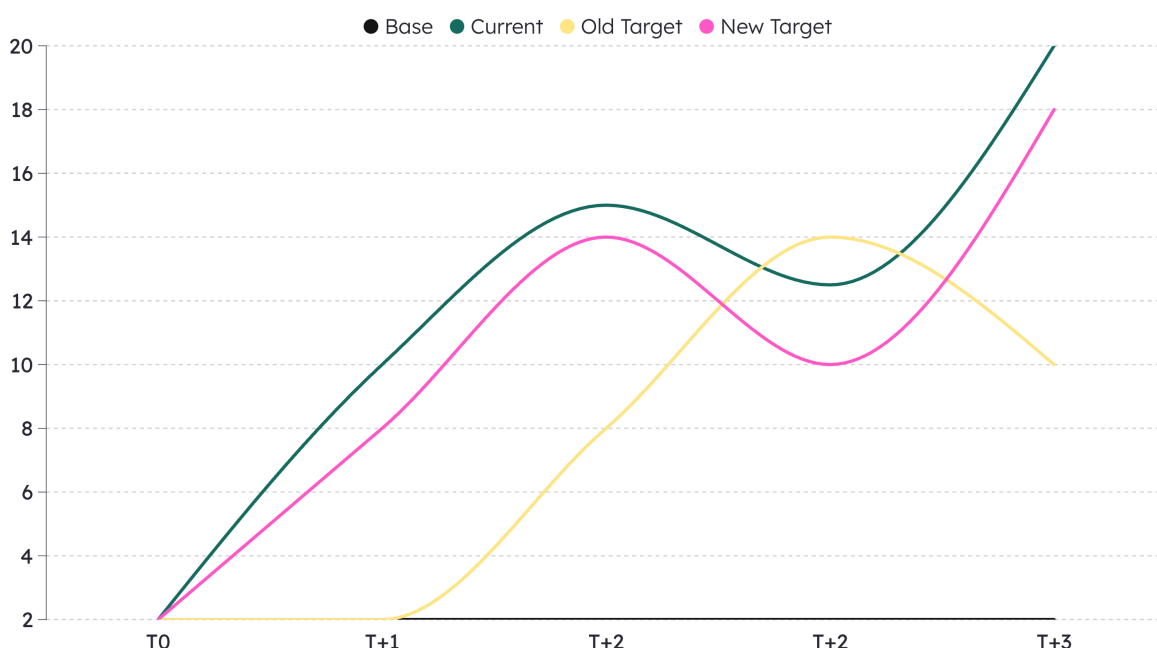
Value-to-Signal Model

The Value-to-Signal (VTS) model governs the settlement of Verified Reserve Liquidity (VRL) in the Fiet Protocol, ensuring that Market Makers (MMs) settle liquidity only when their specific liquidity positions are directly affected by swaps in an Automated Market Maker (AMM) pool. Each MM's position, defined within a price range $[i_l, i_u]$ as in [Uniswap v4's Concentrated Liquidity Market Maker \(CLMM\) framework](#), maintains its own VTS ratio, calculated as the proportion of settled liquidity S to committed liquidity C for each token in the position.

In a Fiet Market, MMs commit VRL, represented as Liquidity Commitment Certificates (LCCs), which encapsulate both settled on-chain liquidity and verified off-chain reserves. The VTS model uses pool-wide indicators, such as swap volume over a time window, to assess demand for each token (e.g., token0 or token1) within active positions. When a swap consumes liquidity from a position's tick range, the VTS model calculates the required settlement to ensure traders receive the full outflow amount in native tokens, not LCCs. This approach significantly minimises capital lockup, aligns with Uniswap v4's concentrated liquidity mechanics, and supports risk management by tying settlements to specific swap activity.

The VTS model underpins Fiet's settlement and risk management mechanisms, enabling efficient liquidity provision while maintaining market integrity by accounting for verified liquidity across systems.

What is the Value-to-Signal Ratio?



The **current Value-to-Signal ratio** ($VTS_{current}$) measures the proportion of settled liquidity to committed liquidity for each token in an MM's position within a Fiet Market. Each position, specified by a price range, maintains separate $VTS_{current}$ ratios for `token0` and `token1`, reflecting the liquidity dynamics of the AMM pool.

As MMs and traders interact with the market via LCCs, which encapsulate both settled and committed liquidity to mirror the pool's effective liquidity, the $VTS_{current}$ for each token in a position adjusts based on market activity:

1. Deposits of a token into the AMM pool, such as trader swaps or MM settlements, increase the corresponding $VTS_{current}$.
2. Withdrawals of a token from the AMM pool, when excess liquidity is removed, decrease the corresponding $VTS_{current}$.

LCCs held outside the market do not affect $VTS_{current}$. The $VTS_{current}$ is enforced at the market level, ensuring that adjustment settled liquidity levels align with demand for each token.

Target Rate

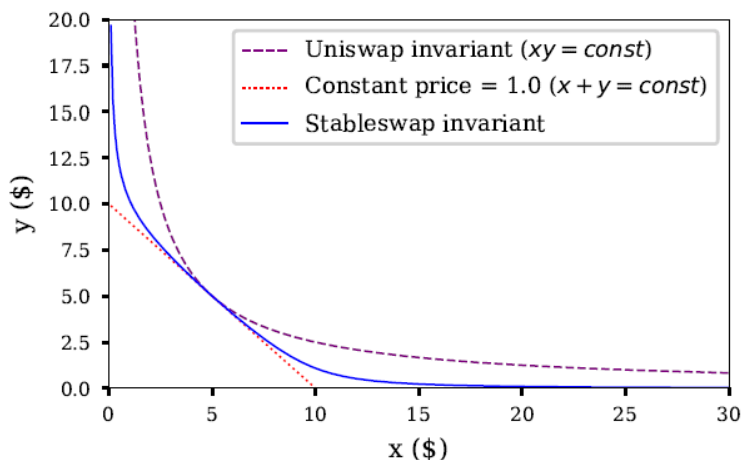
A Fiet Market monitors capitalisation needs through a **target Value-to-Signal ratio** (VTS_{target}), which dynamically adjusts to ensure:

$$VTS_{current} \geq VTS_{target}$$

The (VTS_{target}) represents the required settled liquidity level to meet market demand.

At market launch, (VTS_{target}) is set to a default base rate per token

As market conditions evolve through participant interactions, (VTS_{target}) increases with rising demand for a token and decreases over time as demand declines. AMM accounting allows MMs to settle liquidity for one token and withdraw excess liquidity from the other token (`token0` or `token1`), as demand for one inversely affects the other, following the AMM's symmetrical price curve.



As liquidity increases on the Y axis, liquidity decreases on the X axis, and vice versa.

Each MM's VTS obligation is proportional to their committed liquidity. Larger commitments incur greater settlement requirements but allow larger withdrawals when $VTS_{current} > VTS_{target}$.

New MMs joining an imbalanced market must settle liquidity to meet both tokens' (VTS_{target}), at minimum collateralising their position to the base (VTS_{target}). Existing excess liquidity is proportionally allocated to new MMs upon commitment. If prior MMs withdraw this excess before new commitments, trader swaps cover (VTS_{target}) requirements of the new MM. MMs can time commitments based on market conditions to optimise capital management.

Base Target Rate

The **base Value-to-Signal target ratio** (VTS_{base}) is a fixed rate set per token at market launch, akin to a loan-to-value ratio in lending protocols. The (VTS_{base}) establishes a minimum VTS_{target} rate and ensures collateralisation, seeding the market and incentivising Settlement Guarantors to intervene if MMs fail to settle.

The (VTS_{base}) is determined to account for:

1. Settlement time for the token (e.g., blockchain block time or bank transfer duration).
2. Asset volatility.
3. Liquidation availability across exchanges and corridors.

For example:

- ckBTC, settling in 20 minutes, may have a (VTS_{base}) of 5%, higher than USDC at 2%, which settles faster based on blockchain block time.
- Local stablecoins (e.g., AUD) have a (VTS_{base}) of 3.5%, reflecting bank transfer times balanced by lower volatility.

MMs, typically high-frequency traders, are expected to prepare reserves for settlement requirements in advance.

Atomic State Updates

Every trade in a Fiet Market triggers a state update for pool-wide **Market Demand Indicators**, measuring demand for `token0` and `token1` to determine the liquidity needed to meet that demand. The target VTS rate relies on these indicators for its calculation.

Each modification of liquidity position parameters in a Fiet Market, by MMs or Direct Liquidity Providers (Direct LPs), updates the state managing these positions. For MMs, the Fiet Protocol serves as a position management facility within DEXs, associating VTS rates with each position to ensure precise distribution of obligations based on liquidity utilised by traders. Direct LPs, using default interfaces to the CLMM, including third-party smart contracts, have their positions managed with the same processes as MM positions.

Fiet prioritises integration with hook-enabled AMMs. On the AMM pool addressing a market between the LCCs (e.g., lcc-USDT/lcc-ARB) is modified through a `CoreHook`. After each trade, the `afterSwap` hook updates the state based on the trade's impact on liquidity demand, enabling real-time adjustments to VTS_{target} . The `CoreHook` inherits the underlying accounting model built into the CLMM, without hook-adjusted outflows or deltas. After liquidity position modifications, the `afterAddLiquidity` or `afterRemoveLiquidity` hooks are invoked, allowing atomic updates to the protocol's position awareness and pre-calculation of committed liquidity for each token.

Every core pool has a corresponding proxy pool modified through the `ProxyHook`. This proxy pool establishes a market between native assets (e.g., USDT/ARB) underlying the LCCs, such that traders can engage this particular market without any awareness of the LCCs and Fiet functionality under the hood. In essence, this proxy pool functions as an order proxy to the core pool, abstracting the Fiet-specific logic. The `ProxyHook` proxies trade orders to the `CoreHook` by overriding the `beforeSwapReturnDelta`, piggybacking off the core pool's delta results. Additionally, if there is insufficient liquidity in the market to meet a trade size against the proxy pool, and no `recipient` address has been provided to receive the excess LCCs which cannot be unwrapped into native tokens until an MM settles, then swap simulation caps the trade size to meet what available underlying liquidity there is. The `ProxyHook` inherits functions of a `MarketVault`, logic associated with managing "in-market" liquidity which comprises liquidity deposited via trades, settlement by MMs, or deposit by Direct LPs.

For AMMs without hook functionality, Fiet uses a standardised integration pattern, employing a `SwapRouter` to track swap dynamics and a `PositionManagerProxy` to manage positions for MMs and Direct LPs. The protocol hooks into the LCC `transfer` method to track market interactions, restricting iterations to specific smart contracts. A check tuple distinguishes actions:

- Trade: One token's `transfer` call has `to` as the `SwapRouter` (input token), while the other has `msg.sender` as the `SwapRouter` (output token).
- LP Deposit: Both tokens' `transfer` calls specify `to` as the `PositionManagerProxy`.
- LP Withdrawal: Both tokens' `transfer` calls have `msg.sender` as the `PositionManagerProxy`.

This approach ensures accurate tracking of market activity for the VTS model while maintaining compatibility with diverse AMM architectures.

Fixed Commitments and Dynamic Effective Liquidity

When an MM commits a total liquidity value (e.g., \$1,000,000 USD-denominated VRL) to a Fiet Market pairing two tokens (e.g., ETH and USDC), this total commitment $C(r)$ for position (r) is cloned into $C_0(r)$ and $C_1(r)$, representing the **maximum potential amounts** of `token0` and `token1` across the position's price range.

The total commitment is fixed in value denominated in a common base currency (e.g., USD):

$$C(r) = \frac{1}{2} (V_0 \cdot C_0(r) + V_1 \cdot C_1(r))$$

where V_0 and V_1 are the current market values (in the base currency) of one unit of token0 and token1, respectively. For example, in a USDC/ETH market with USDC valued at 1 USD and ETH at 2,500 USD, $V_1 = 1$ (USDC) and $V_0 = 2500$ (ETH).

LCCs, minted during commitment and deposited into the AMM, encapsulate settled and committed liquidity, functioning as effective liquidity ($x(r)$, $y(r)$) of the position, as described in Uniswap v3 Whitepaper (Section 2). The sum of effective liquidity in USD, ($x(r) + y(r)$), equals ($C(r)$) for the position.

For example, at a price where 1 ETH (`token0`) equals 2000 USDC (`token1`), a \$1,000,000 commitment will establish:

- $C_0(r) = 500$ ETH, valued at \$1,000,000.
- $C_1(r) = 1,000,000$ USDC.

The LCCs allocated to the pool (e.g., $LCC_0(r) = 250$ ETH, $LCC_1(r) = 500,000$ USDC) depend on the current tick and price range, adjusted by prior trade activity.

Due to the AMM's price curve dynamics, the effective liquidity $x(r)$, $y(r)$ cannot reach the maximum potential of both tokens simultaneously. If one token's effective liquidity reaches $C_0(r)$ or $C_1(r)$, the other token's liquidity is completely exhausted, rendering the position out-of-range. Post-commitment, $C_0(r)$ and $C_1(r)$ remain constant unless the MM decommits or adjusts the position, while the effective liquidity ($LCC_0(r)$, $LCC_1(r)$) or ($x(r)$, $y(r)$) shifts dynamically with the current tick.

Model

The Value-to-Signal (VTS) model is a fundamental mechanism in the Fiet Protocol, governing liquidity commitments and settlement requirements for each token in a Fiet Market. The model relies on two key

metrics: the **current VTS rate** (VTS_{current}) and the **target VTS rate** (VTS_{target}). These metrics work together to ensure that market makers (MMs) settle liquidity in response to market demand while optimising capital efficiency.

MMs commit VRL via LCCs, settling and withdrawing liquidity at their discretion, incentivised by the target VTS rate, while Direct LPs fully settle liquidity upfront, using Uniswap v4's default position management. The model defines current, required, and target VTS rates per position, ensuring settlements align with swap-driven demand.

VTS_{current}

The current VTS rate measures the proportion of settled liquidity to committed liquidity for a given position r (with range $[i_l, i_u]$ and liquidity $L(r)$) and token A . It is defined as:

$$VTS_{\text{current}}(r, A) = \frac{S_A(r)}{C_A(r)}$$

Where:

- $S_A(r)$: The settled liquidity for token A in position r , including liquidity settled by MMs (via settlements), traders (via swaps), or Direct LPs (effective liquidity provided upfront).
- $C_A(r)$: The committed liquidity for token A in position r , representing the maximum potential amount of token A across the position's price range:

$$C_0(r) = L(r) \cdot \left(\frac{1}{\sqrt{p(i_l)}} - \frac{1}{\sqrt{p(i_u)}} \right), \quad C_1(r) = L(r) \cdot \left(\sqrt{p(i_u)} - \sqrt{p(i_l)} \right)$$

Where $(p(i_l) = 1.0001^{i_l})$, $(p(i_u) = 1.0001^{i_u})$, and $L(r)$ is the liquidity parameter for position r .

- A : Represents the token where $A = 0$ for `token0`, e.g., ETH, and $A = 1$ for `token1`, e.g., USDC.

For all positions, $VTS_{\text{current}}(r, A) \leq 1$, as $S_A(r) \leq C_A(r)$ reflecting partial settlement based on market demand.

The rates are calculated independently for each token, therefore:

$$VTS_{\text{current}}(r, 0) + VTS_{\text{current}}(r, 1) \leq 2$$

However, for Direct LPs, $S_A(r)$ liquidity is fixed and settled upfront. Therefore, $S_A(r)$ where position r is in-range $(i_l \leq i_c < i_u)$ can be calculated as $S_0(r) = \Delta x$ for `token0`, $S_1(r) = \Delta y$ for `token1`, where Δx , Δy are derived from the [Uniswap v3 Whitepaper](#) (Page 9, Equations 6.29, 6.30).

This ratio provides a real-time measure of the liquidity currently available in the market relative to the commitments made.

What is Square Root Price?

In the context of concentrated liquidity market makers like Uniswap v3 and v4, $\sqrt{p(i)}$ and \sqrt{P} are related but distinct concepts, both representing square-root prices for computational efficiency in the constant product formula.

To clarify:

- $\sqrt{p(i)}$ refers to the square-root price at a specific tick boundary i . Ticks are discrete points on the price curve, where the price at tick i is defined as $p(i) = 1.0001^i$ (with 1.0001 being the tick spacing factor). This makes $p(i)$ the fixed square-root value at that tick's lower or upper bound, used for position ranges and liquidity calculations at boundaries.
- \sqrt{P} , on the other hand, is the exact current square-root price of the pool, which can lie anywhere between the square-root prices of the current tick i_c and the next tick $i_c + 1$ (i.e., within $[\sqrt{p(i_c)}, \sqrt{p(i_c + 1)}]$). During a swap, P updates continuously as liquidity is depleted within the tick, even if no tick boundary is crossed.

This distinction allows for precise tracking of intra-tick price movements during swaps, while tick-based $\sqrt{p(i)}$ provides efficient discretisation for position bounds and bitmap storage. For example, outflow calculations in a swap use \sqrt{P} for exact deltas, falling back to tick-bound $\sqrt{p(i)}$ when approximating at boundaries.

VTS_{target}

The target VTS rate dynamically adjusts the required liquidity settlement based on market demand and position characteristics. It sets the target level of settled liquidity that an MM should aim to achieve for their position r .

VTS_{target} is a complex calculation that depends on various sub-formulas.

Market Demand Indicators

Market demand is evaluated through pool-wide indicators. This calculation is based on activity that impacts the pool at large, and therefore functions as a parameter within each position's target VTS rate calculation.

The demand indicators $I_A(t)$ project future liquidity needs:

1. Total token A outflow over time window ($[t - T, t]$):

$$\Delta O_0 = \sum_{\text{swaps in } [t-T, t]} |\Delta x|, \quad \Delta O_1 = \sum_{\text{swaps in } [t-T, t]} |\Delta y|$$

Where:

- t : Is the current time
 - T : Is a fixed time in configured at market deployment
 - $\Delta x, \Delta y$: Derived from the integrated CLMM
2. A **boost term** to project additional liquidity required beyond immediate demands, ensuring a smooth trader experience through pre-settled liquidity. The aim is to completely abstract interactions directly from LCCs from traders. While not completely possible, as sufficiently large trades will absorb all of the pre-settled liquidity, for the majority of trades, the boost term fulfils this goal.

- a. For **token1** in, **token0** out:

$$B_0 = \alpha \cdot \frac{|\Delta x|}{\sum_{\text{active } r} L(r) \cdot \left| \frac{1}{\sqrt{P_{\text{after}}}} - \frac{1}{\sqrt{P_{\text{before}}}} \right|}$$

- b. For **token0** in, **token1** out:

$$B_1 = \alpha \cdot \frac{|\Delta y|}{\sum_{\text{active } r} L(r) \cdot \left| \sqrt{P_{\text{after}}} - \sqrt{P_{\text{before}}} \right|}$$

Where:

- α : A scaling parameter (e.g., 0.1 to 2)
- $L(r)$: Liquidity in the position
- $\sum_{\text{active } r} L(r)$: Sum of liquidity in positions where position r with range $[i_l, i_u]$ is active and directly affected during a swap, if its range intersects the swap's tick range
 - For a swap increasing ticks (token1 in, token0 out): $i_l \leq i_{c,\text{after}}$, and $i_u > i_{c,\text{before}}$
 - For a swap decreasing ticks (token0 in, token1 out): $i_l \leq i_{c,\text{before}}$ and $i_u > i_{c,\text{after}}$
 - This includes positions where $i_{c,\text{before}} > i_l$ and $i_{c,\text{after}} > i_u$, which are active during the swap but become out-of-range post-swap.
- $\sqrt{P_{\text{before}}}$: Square-root price before the swap.
- $\sqrt{P_{\text{after}}}$: Square-root price after the swap (updated continuously, even within ticks).
- $\Delta x, \Delta y$: Represent the total outflow for a specific token from an individual swap, allowing the term to quantify the demand intensity of each swap event in isolation.
- A **decay term** that ensures the target VTS rate returns to its base rate (VTS_{base}) if demand for a token A wanes:

$$D = e^{-\lambda(t-t_{\text{last}})}$$

Where: λ is the decay rate (e.g., $\frac{\ln(2)}{3600} \approx 0.0001927$)

Therefore, the Indicator is defined as:

$$I_0(t) = D \cdot I_0(t_{\text{last}}) + (1 - D) \cdot B_0, \quad I_1(t) = D \cdot I_1(t_{\text{last}}) + (1 - D) \cdot B_1$$

Where: $I_0(t), I_1(t)$ are stateful indicators updated after each swap.

Required VTS Rate ($VTS_{required}$)

To ensure that the VTS_{target} is sufficiently liquid, such that traders receive the exact expected amount of native token per their swap activity, there must be a minimum threshold of liquidity required to be settled to at least meet the direct requirements of the swap activity. Establishing this baseline constitutes sufficient liquidity. Anything in excess covers a projection based on future demand.

The time-window required VTS rate, $VTS_{required}(t, r, A)$, measures the proportion of committed liquidity needed to cover cumulative outflows over the time window $[t - T, t]$:

1. Sum Outflows:

$$\Delta O_0(r) = \sum_{\text{swaps in } [t-T, t]} |\Delta x(r)|, \quad \Delta O_1(r) = \sum_{\text{swaps in } [t-T, t]} |\Delta y(r)|$$

Where:

- $\Delta x(r), \Delta y(r)$: Outflow amounts for position r in a swap:

- **token1** in, **token0** out:

$$\Delta x(r) = \begin{cases} L(r) \cdot \left(\frac{1}{\sqrt{P_{\text{after}}}} - \frac{1}{\sqrt{P_{\text{before}}}} \right) & \text{if position active in swap} \\ 0 & \text{otherwise} \end{cases}$$

- **token0** in, **token1** out:

$$\Delta y(r) = \begin{cases} L(r) \cdot \left(\sqrt{P_{\text{after}}} - \sqrt{P_{\text{before}}} \right) & \text{if position active in swap} \\ 0 & \text{otherwise} \end{cases}$$

- For no exact price delta (fallback proportional allocation):

- **token1** in, **token0** out:

$$\Delta x(r) = \frac{\Delta y \cdot L(r)}{\sum_{\text{in-range } r} L(r)}$$

- **token0** in, **token1** out:

$$\Delta y(r) = \frac{\Delta x \cdot L(r)}{\sum_{\text{in-range } r} L(r)}$$

- $\sqrt{P_{\text{before}}}$: The square-root price at the start of the swap's impact on position r (e.g., $\max(\sqrt{p(i_l)}, \text{initial } \sqrt{P})$ within the tick).
- $\sqrt{P_{\text{after}}}$: The square-root price at the end of the swap's impact (e.g., $\min(\sqrt{p(i_u)}, \text{final } \sqrt{P})$ after depletion).
- "Position active in swap": The position's range $[i_l, i_u]$ intersects the traversed square-root price interval during the swap.
- Other variables as previously defined (e.g., $L(r)$, Δy , Δx , etc.).

2. In-Range Calculation:

$$VTS_{required}(t, r, A) = \min \left(1, \frac{\Delta O_A(r)}{C_A(r)} \right)$$

where:

- A : The token where $A = 0$ for **token0**, e.g., ETH, and $A = 1$ for **token1**, e.g., USDC.
- $\min(1, \dots)$ wrapper to prevent ratios >1 in high-outflow scenarios, ensuring full settlement for position r at most.

The allocation of pool-wide swap outflows (ΔO_A) to position-specific outflows ($\Delta O_A(r)$) is achieved by determining each position's contribution to the total outflow during individual swaps. This process relies on the concentrated liquidity mechanics, where outflows are distributed based on the liquidity $L(r)$ of each active position relative to the total liquidity in the affected price range.

For each swap, compute the outflow delta per position r ($\Delta x(r)$ or $\Delta y(r)$) using the position's liquidity and the square-root price change. Then, sum these deltas over the time window $[t - T, t]$ to obtain $\Delta O_A(r)$.

This ensures position-specific granularity: positions with higher $L(r)$ in the traversed range contribute more to the outflow, while inactive positions contribute zero. The pool-wide ΔO_A is simply the sum across all positions, but the per-position breakdown drives the VTS calculations.

Target Rate Definition

The target Value-to-Signal (VTS) rate, $VTS_{\text{target}}(r, A)$, determines the proportion of committed liquidity for token A (where $A = 0$ for `token0`, e.g., ETH, and $A = 1$ for `token1`, e.g., USDC) that Market Makers (MMs) should settle for a position r with range $[i_l, i_u]$ to meet projected market demand in a Fiat Market. The calculation differs based on whether the position is in-range ($i_l \leq i_c < i_u$) or soon to be in-range, defined as out-of-range positions close to the current tick i_c that are likely to become active based on recent swap activity.

In-Range Positions:

For an in-range position r ($i_l \leq i_c < i_u$), the target VTS rate ensures sufficient liquidity to cover past and projected demand, using the time-window-based required VTS rate as a minimum threshold:

$$VTS_{\text{target}}(r, A) = \min \left(1, \max \left(VTS_{\text{required}}(t, r, A), VTS_{\text{base},A} + I_A(t) \cdot \frac{L(r)}{\sum_{\text{in-range } r} L(r)} \right) \right)$$

Where:

- t : Current time.
- A : Token index ($A = 0$ for `token0`, $A = 1$ for `token1`).
- $L(r)$: Liquidity parameter for position r , defining its contribution to the pool's liquidity (Uniswap v3 Whitepaper, Page 5, Section 6.2.1).
- $VTS_{\text{base},A}$: Base VTS rate set at market launch (e.g., 0.02 for USDC, 0.05 for ckBTC)
- $I_A(t)$: Pool-wide market demand indicator for token A , capturing recent trading activity over the time window $[t - T, t]$.
- $\sum_{\text{in-range } r} L(r)$: Total liquidity of in-range positions, used to weight the demand indicator.
- $\min(1, \dots)$: Prevents settling beyond the size of each in-range position. The Excess Liquidity in Soon-to-Be In-Range Positions is the difference between the Sum of Uncapped to Capped projections, i.e., Excess is all the projections remaining after we cap against in-range positions.

Soon-to-Be In-Range Positions:

For soon-to-be in-range positions (out-of-range but near i_c), allocate excess liquidity requirements across positions likely to activate, using tick-based iteration with proximity decay adjusted for recent tick velocity to prioritise closer ranges.

- For `token0`: Iterate forward over initialised ticks $i > i_c$ in the TickBitmap.
- For `token1`: Iterate backward over initialised ticks $i \leq i_c$
- For each tick i :
 - Define:

$$N_A(i) = \{r : i_l = i\} \text{ (token0)} \quad \text{or} \quad \{r : i_u = i\} \text{ (token1)}$$

This defines the set of positions starting (for token0) or ending (for token1) at the current tick, grouping them for allocation.

$$L_i = \sum_{r \in N_A(i)} L(r)$$

This calculates the total liquidity of positions in the set at the current tick, used for proportional weighting.

- Initialise:

$$\begin{aligned} E_A &= \text{Excess Required Liquidity}_A \text{ (for } VTS_{\text{required}}), \\ E_A &= \text{Excess Liquidity}_A \text{ (for } VTS_{\text{target}}), \\ N_A &= \emptyset \end{aligned}$$

This sets up the excess liquidity to allocate and an empty set for accumulating processed positions.

- For each $r \in N_A(i)$

- Compute for $VTS_{\text{required}}(t, r, A)$:

$$VTS_{\text{potential}}(t, r, 0) = \frac{E_0}{\sum_{r' \in N_0 \cup N_0(i)} C_0(r')} \cdot \frac{L(r)}{L_i} \cdot e^{-\kappa_v |i_i - i_c|}$$

$$VTS_{\text{potential}}(t, r, 1) = \frac{E_1}{\sum_{r' \in N_1 \cup N_1(i)} C_1(r')} \cdot \frac{L(r)}{L_i} \cdot e^{-\kappa_v |i_u - i_c|}$$

Computes a potential required VTS rate for token0 or token1, weighting excess by committed liquidity, position liquidity share, and proximity decay.

- Compute for $VTS_{\text{target}}(r, A)$

$$VTS_{\text{potential}}(r, 0) = VTS_{\text{base},0} + \frac{E_0}{\sum_{r' \in N_0 \cup N_0(i)} C_0(r')} \cdot \frac{L(r)}{L_i} \cdot e^{-\kappa_v |i_i - i_c|}$$

$$VTS_{\text{potential}}(r, 1) = VTS_{\text{base},1} + \frac{E_1}{\sum_{r' \in N_1 \cup N_1(i)} C_1(r')} \cdot \frac{L(r)}{L_i} \cdot e^{-\kappa_v |i_u - i_c|}$$

This computes a potential target VTS rate for token0 or token1, adding the base rate to the weighted excess allocation with decay.

- For both:

- If $VTS_{\text{potential}}(t, r, A) \leq 1$ or $VTS_{\text{potential}}(r, A) \leq 1$:

- Set $VTS_{\text{required}}(t, r, A) = VTS_{\text{potential}}(t, r, A)$ or $VTS_{\text{target}}(r, A) = VTS_{\text{potential}}(r, A)$
- Update $E_A = E_A - VTS_{\text{required/target}}(r, A) \cdot C_A(r)$.

This assigns the potential rate if within bounds and reduces excess by the allocated amount.

- If $VTS_{\text{potential}}(t, r, A) > 1$ or $VTS_{\text{potential}}(r, A) > 1$:

- Set $VTS_{\text{required}}(t, r, A) = 1$ or $VTS_{\text{target}}(r, A) = 1$.
- Update $E_A = E_A - C_A(r)$.

This caps the rate at full settlement if potential exceeds 1 and reduces excess by the full commitment.

- Add $N_A(i)$ to N_A .

This accumulates the processed position set for the next iteration's summation.

- Stop when $E_A \leq 0$ or no more initialised ticks.

This halts allocation once excess is depleted or all relevant ticks are covered.

- Parameters: $\kappa_v = 0.01 \cdot v$ (proximity decay factor, where v is recent tick velocity, e.g., average ticks crossed per swap over the last hour).

This adjusts the decay based on market volatility to prioritise nearer positions in active conditions.

Breakdown of Symbols:

Consider the $VTS_{\text{potential}}(t, r, 1)$. The expression calculates the total committed liquidity (for token1, in this case) across a combined set of liquidity positions. It sums the maximum commitment values $C_1(r')$ for all positions r' that belong to the union of two sets: the accumulated set of previously processed positions N_1 and the set of positions at the current tick ($N_1(i)$). This total is then used to normalise or weight the allocation of excess liquidity requirements in the VTS model, ensuring settlements are proportional to the commitments of nearby positions likely to become active soon.

In the Fiet Protocol, this supports market makers in facilitating liquidity by dynamically adjusting settlement obligations based on anticipated demand, particularly for positions near the current price tick in concentrated liquidity market makers (CLMMs) like Uniswap v3 or v4.

Here is a step-by-step explanation of the symbols in the expression $\sum_{r' \in N_1 \cup N_1(i)}$:

- \sum : This is the summation operator (sigma). It indicates that you add up the values of the term that follows (in this case, implicitly $C_1(r')$ or a similar quantity) for every element in the specified set.
- r' : This is a dummy variable (often called an index or iterator). It represents each individual liquidity position in the set being summed over. The prime (') distinguishes it from the main position r in the

broader formula. In the Fiet context, r' iterates over positions similar to r , which are bounded price ranges where market makers commit liquidity.

- \in : This symbol means "is an element of" or "belongs to." It specifies that r' must be a member of the set that follows.
- N_1 : This is the accumulated set of positions for token1. It starts as an empty set ($N_1 = \emptyset$) and grows by adding groups of positions ($N_1(i)$) as the algorithm iterates over ticks. Positions in N_1 are those that have already been processed in previous ticks during the allocation.
- \cup : This is the set union operator. It combines two sets into one, including all unique elements from both without duplicates. Here, it merges the accumulated positions N_1 with the current tick's positions $N_1(i)$.
- $N_1(i)$: This is the set of positions for token1 that end at the specific tick i (defined as $N_1(i) = r : i_u = i$, where i_u is the upper tick bound of position r). The subscript "1" indicates token1, and (i) denotes the current tick being evaluated in the iteration.

For completeness, while not part of the summation itself, the summed term (e.g., $C_1(r')$) refers to the maximum committed liquidity for token1 in position r' , calculated as $C_1(r') = L(r') \cdot (\sqrt{p(i_u)} - \sqrt{p(i_l)})$, where $L(r')$ is the liquidity parameter, and $\sqrt{p(\cdot)}$ is the square-root price at the tick bounds.

This notation draws from standard set theory and summation in mathematical modelling, adapted to the Fiet Protocol's approach to liquidity commitments in CLMMs.

Oracle

Oracles are smart contracts that deliver external data, particularly price information, to the Fiet Protocol. They provide price data to determine the relative value of currencies, such as calculating how many USDC one BTC is worth at a given moment.

Oracles in Fiet

Oracles in the Fiet Protocol provide price data to:

1. Calculate the value of signal currencies relative to committed tokens in VTS ratio computations.
2. Assess MM solvency for settlement obligations.
3. Trigger incentives for Settlement Guarantors in risk management processes.

When the signal currency matches the settlement token, oracles are bypassed, reducing external data dependencies.

Implementation

The Fiet Protocol adopts an oracle-agnostic approach, allowing LCC creators to select price feed mechanisms tailored to specific market requirements. Each LCC specifies its oracle via parameters, ensuring flexibility in implementation.

Oracles in Fiet Markets implement the [IOracle interface](#), adopted from the Morpho Blue protocol, defined as:

```
function price() external view returns (uint256);
```

This function returns the price of one unit of signal currency quoted in the settlement token, scaled to account for decimal differences between currencies.

Types of Oracles Compatible

Fiet Markets support various oracle implementations:

1. **Price Feed Oracles:** Leverage external price feeds from providers like Chainlink, Redstone, ChainSight, or Pyth to compute asset exchange rates.
2. **Exchange Rate Oracles:** Designed for wrapped or rebasing tokens (e.g., wstETH/stETH) with deterministic exchange rates.
3. **Fixed-Price Oracles:** Applied to assets with stable or predefined exchange rates, such as stablecoins pegged to the same value.

Key Oracle Characteristics

1. **Purpose-Built:** Each oracle delivers the exchange rate between a signal currency and a settlement token for a specific market.
2. **Immutable:** Oracle addresses are fixed upon market deployment, ensuring data source consistency.
3. **Independent:** Oracles operate autonomously, using distinct pricing sources to enhance reliability.
4. **Flexible Implementation:** Market creators can select varied data sources while adhering to the standardised interface.

Settlements

Settlements in the Fiet Protocol enable Market Makers (MMs) to fulfil liquidity commitments in response to trader demand within specific price ranges. Using the Value-to-Signal (VTS) model, the protocol calculates a Request for Settlement (RfS) per position and token, determining the amount MMs must deposit or may withdraw to align settled liquidity with market needs. This approach supports efficient market facilitation in concentrated liquidity automated market makers (AMMs), where settlements are triggered by outflows in active or soon-to-be active positions, ensuring traders access native tokens without unnecessary capital lockup.

What is a Request for Settlement?

A Request for Settlement (RfS) is a position-specific and token-specific directive generated by the Fiet Protocol, instructing an MM on the net liquidity adjustment required for a given position and token A (where $A = 0$ for `token0`, $A = 1$ for `token1`).

$$a_A(r) = VTS_{\text{target}}(r, A) \cdot C_A(r) - S_A(r)$$

Where:

- $VTS_{\text{target}}(r, A)$: The target VTS rate for position r and token A , derived from market demand indicators and required rates (as detailed in the VTS model). This is computed statelessly using pool-wide indicators updated post-swap.
- $C_A(r)$: The maximum committed liquidity for token A in position r , based on the liquidity parameter $L(r)$ and range bounds (e.g., $C_0(r) = L(r) \cdot \left(\frac{1}{\sqrt{p(i_l)}} - \frac{1}{\sqrt{p(i_u)}} \right)$).
- $S_A(r)$: The current settled liquidity for token A in position r , tracked as deposited native tokens (including trader inflows).

For out-of-range positions, $a_A(r)$ applies to the single active token (e.g., full `token0` if $i_c \geq i_u$). For soon-to-be in-range positions, excess demand is allocated via tick-based iteration with proximity decay. RfS is exposed as a public view function, allowing MMs to query or recalculate off-chain for efficiency, without storing per-position state per swap. Direct LPs, settling fully upfront, have $VTS_{\text{current}}(r, A) = 1$, yielding $a_A(r) \leq 0$.

This formula enables MMs to facilitate markets by settling only as needed, optimising capital while ensuring liquidity meets demand in affected positions.

MMs are incentivised to settle liquidity when the RfS is open, i.e., $a_A(r) > 0$. This positive value signals an obligation to deposit additional native tokens into the position to align with the target VTS rate, supporting market continuity and earning exchange fees proportional to their facilitated liquidity.

Withdrawals are permitted when $a_A(r) < 0$, allowing MMs to retrieve excess settled liquidity for the specific token in the position. This mechanism promotes capital efficiency by enabling the reallocation of funds no longer required to meet current demand.

RfS enforcement is targeted: only positions actively utilised by traders during swaps, whether in-range or projected to activate, incur adjustments. Settlement obligations therefore align to the fees accrued from that utilisation due to concentration.

Fiet Markets pair two LCCs in an AMM pool, where the AMM accounting model reduces demand for one token as demand for the other increases. MMs settling liquidity for the high-demand token can later withdraw excess liquidity from the low-demand token, with amounts determined by their exposure to $a_A(r)$.

MMs are advised to maintain a settlement buffer, ensuring $a_A(r) \leq 0$, to mitigate the risk of liquidity position seizure.

Incentives and Seizure

When MMs commit to a Fiet Market, they must collateralise their commitment to meet the base $VTS_{\text{base},A}$, a fixed percentage of their total committed liquidity configured per token A at market launch. If an MM fails to settle liquidity for a token when a Request for Settlement (RfS) is open, their liquidity position becomes available for seizure by other parties after a grace period, functioning similarly to liquidations in money market protocols. The collateralised liquidity position incentivises intervention by other parties to settle on behalf of the failing MM.

The intervening party, a Settlement Guarantor, can be another MM or a dedicated bot. Guarantors settling the required amount acquire all or part of the failing MM's liquidity position, including settled liquidity or at least

the collateral committed to the market. Non-MM Guarantors are expected to liquidate the seized position immediately to realise profit.

Seizure Process

1. When an RfS opens ($a_A(r) > 0$), a grace period begins, allowing MMs to settle their obligations without penalty.
2. After the grace period, unsettled MMs' liquidity positions become available for seizure, with the amount calculated on a linear scale based on time elapsed since the period's end, increasing until the full position is seizable.
3. Guarantors may wait until the seizure is sufficiently lucrative before settling on behalf of the failing MM.
4. Upon settlement:
 - If the Guarantor is an MM with verified reserve liquidity, they receive a new Fiet Market position NFT, assuming the failing MM's obligations.
 - If not an MM, the protocol liquidates the seized position, allowing the Guarantor to withdraw the underlying settled liquidity for liquidation on third-party exchanges.

Grace Period

- A fixed window per token, set at market deployment, allows MMs to settle liquidity before seizure begins, based on typical settlement durations for the token (e.g., longer for bank transfers in jurisdictions with slower financial systems than for stablecoin withdrawals from a centralised exchange).
- Tracked by block number range from RfS opening to the period's end.

Fiet assumes MMs are sophisticated and capable of preparing their reserve liquidity for settlement as the RfS conditions near 'open'. The grace period primarily serves to prevent MEV operations from seizing liquidity positions of MMs generally acting in good faith.

This structure ensures settlement obligations are met, with collateralisation and seizure mechanisms safeguarding market liquidity.

Seizure Amount

When a Request for Settlement (RfS) remains open after the grace period, the liquidity position of a failing MM becomes available for seizure on a linear scale based on time. The seizure amount is calculated as:

$$s(r) = C(r) \cdot \min \left(1, \frac{t}{t_{\max,A}} \cdot \left(1 + \alpha \cdot \frac{a_A(r)}{C_A(r)} \right) \right)$$

Where:

- $s(r)$: The portion of the position r available for seizure (applied to the entire position, encapsulating both tokens).
- $C(r)$: The MM's total committed liquidity for position r , normalised by averaging the commitments across both tokens after converting them to a common base currency (e.g., USD) based on the current market value.
- t : Time elapsed since the grace period's end (in seconds).
- $t_{\max,A}$: The fixed maximum time per token until the entire position is seizable (e.g., 3600 seconds).
- α : A sensitivity parameter controlling the impact of the MM's RfS exposure (e.g., 1.5).
- $a_A(r)$: The MM's exposure to the RfS amount for token A in position r .
- $C_A(r)$: The MM's committed liquidity for the intervened token A in position r .

The formula scales the seizure rate linearly with time, adjusted by the MM's exposure to the RfS amount relative to their committed liquidity for the specific token ($\frac{a_A(r)}{C_A(r)}$). Seizure occurs at the position level (r) encapsulating both tokens; however, the trigger is token-specific (failure on $a_A(r) > 0$). MMs with larger exposures for the intervened token face faster seizures, reflecting their greater responsibility to settle. The $\min(1, \cdot)$ function caps the seizure at the full position $C(r)$, ensuring completion within $t_{\max,A}$.

Market Makers (MMs) facilitate liquidity across diverse assets, and per-token timelines encourage participation in markets with mixed settlement speeds (e.g., crypto-fiat pairs). It also aligns guarantor incentives: Shorter deadlines for fast-settling tokens prompt quicker interventions, while longer ones for slower assets allow MMs more leeway without assuming bad faith.

This mechanism balances fairness for failing MMs needing time to settle, Guarantors incentivised to intervene, and Traders requiring timely settlements to maintain market liquidity. Upon seizure, the guarantor

claims the proportional share of the entire position, including both tokens' settled liquidity and future obligations.

Looping

To mitigate settlement risks and guarantee liquidity for traders, the Fiet Protocol allows non-MM Settlement Guarantors to participate, complementing the privileges granted to good-faith MMs. Typically, institutional MMs hold significantly larger liquidity reserves than bot operators in cryptocurrency markets.

Seizure mechanics permit partial settlements based on the liquidity a Guarantor can facilitate in a single transaction. A Guarantor can settle part of a failing MM's obligation for a specific token in a position and seize a proportional fraction of the available liquidity position. By removing assets from the Fiet Market and liquidating them on third-party exchanges, the Guarantor can reacquire the native token (settlement currency) to continue settling further portions of the obligation, seizing additional fractions of the position. This iterative process, termed looping, enables Guarantors with smaller reserves to fully guarantee settlements and seize the entire liquidity position of a failing MM in stages.

Proof of Settlement for Grace Period Extension

The Fiet Protocol accounts for external variables affecting settlement times, such as bank transfer delays, cross-chain issues, or centralised exchange withdrawal times. MMs can compensate Provers to generate cryptographic proofs of their intent to settle, such as a pending withdrawal from a centralised exchange to a registered custodial wallet or a settlement smart contract on the Fiet Market Chains.

Submitting a valid Proof of Settlement to the Fiet Market renews the grace period for the MM if verified. If the grace period has expired, verification initiates a new grace period. However, any liquidity position already available for seizure due to a late submission remains seizable. This mechanism allows MMs to extend their grace period during an RfS or minimise seizure risks if delayed, ensuring good-faith efforts are protected while maintaining market integrity.

Guarantor Incentive Structure

The following scenario offers a breakdown of the incentive structure that guarantees settlements to Fiet Markets.

Context

In an AMM pool with virtual and realised liquidity, MMs deposit a base collateral equal to 2% of their committed liquidity per token.

Consider a USDC/ETH pool, with $C_{total} = 1,000,000$ USD (normalised across tokens), at a current tick $i_c = 2000$. Fiet assumes a concentrated liquidity model, with an aggregated position r (in-range $[1900, 2100]$, $L(r) = 2,500,000$) to align with the normalised commitment. Square-root prices: $\sqrt{p(1900)} = 1.4$, $\sqrt{p(2100)} = 1.6$, current $\sqrt{P} = 1.5$. Assume ETH is valued at \$4,000 USD per ETH.

The pool initially holds:

- Realised Liquidity: 10,000 USDC and ETH valued at 10,000 USD (2.5 ETH).
- Virtual Liquidity: 980,000 USD (1,000,000 total from MMs' commitments minus realised liquidity)

A trader swaps ETH for 100,000 USDC (token1 out), requiring 100,000 USDC, but the pool initially holds only 10,000 USDC in realised liquidity, leaving a shortfall of 90,000 USDC that MMs must settle proportionally to their commitments in the affected position.

If MM A holds 50% of the commitments ($C(r) = 500,000$ USD normalised, with $C_1(r) = 500,000$ USDC max normalised to 500,000 USD, $C_0(r) = 125$ ETH max normalised to 500,000 USD), their settlement obligation for USDC in r is 45,000 USDC ($a_1(r) = VTS_{target}(r, 1) \cdot C_1(r) - S_1(r)$, assuming target 0.2 and settled 5,000). If MM A fails to settle, another MM (e.g., MM B) can settle the 45,000 USDC on MM A's behalf and seize MM A's liquidity position.

Calculations

Trader's Action: Deposits ETH equivalent to 100,000 USD (25 ETH at \$4,000 per ETH), withdraws 100,000 USDC.

Pool's Realised Liquidity Before Settlement:

- USDC: $10,000 - 10,000 = 0$ (pool provides its 10,000 USDC to trader, shortfall remains 90,000 USDC)

- ETH: ETH valued at 10,000 USD + inflow (adjusted for price shift, equivalent to 100,000 USD total inflow) = 110,000 USD

MM Settlement: MMs collectively settle 90,000 USDC to cover the shortfall.

Pool's Realised Liquidity After Settlement:

- USDC: $0 + 90,000 = 90,000$, but trader withdraws the full 100,000 USDC (initial 10,000 + settled 90,000), resulting in 0 USDC remaining in the pool.
- ETH: 110,000 USD
- To Trader: 100,000 USDC (fulfilled by initial reserves + settlements)

MM A's Liquidity Position

MM A's 50% share of the pool's realised liquidity post-swap in r is 55,000 USD (value across tokens, post-inflow ETH equivalent to 55,000 USD).

Guarantor's Profit

If MM B settles MM A's 45,000 USDC obligation in r :

- MM B seizes MM A's position: 55,000 USD value.
- Profit = $55,000 \text{ USD} - 45,000 \text{ USD} = 10,000 \text{ USD}$.

This 10,000 USD profit incentivises MM B's intervention, reflecting the base collateral and position value.

Incentive Mechanism

The base collateral acts as an incentive for guarantors (e.g., MM B) to settle on behalf of a failing MM (e.g., MM A), as the seized liquidity position's value (55,000 USD) exceeds the settlement cost (45,000 USD), yielding a profit proportional to the failing MM's exposure. This ensures market liquidity and operational stability.

Rewards

What are Rewards?

Rewards in the Fiet Protocol are incentives distributed to users to encourage behaviours such as market making, trading, or proving. These rewards, typically in the form of FIET tokens or other assets, are provided by:

- The Fiet DAO.
- Market creators incentivising MMs.
- Token issuers promoting market usage.

External incentives and points programs, managed outside the protocol in external applications, are not part of Fiet's rewards. Their eligibility, computation, and distribution occur externally, and the Fiet interface may display them for informational purposes only.

Types of Reward Programs

Market Programs

These programs encourage specific activities within individual markets:

- **Maker Rewards:** Earned by MMs for facilitating liquidity in a market.
- **Taker Rewards:** Earned by users for executing trades in a market.

Rewards are distributed linearly over a defined period, with fixed amounts allocated for each activity type.

Uniform Rate Programs

Administered by the Fiet DAO for FIET token distribution, these programs apply a consistent reward rate per dollar exchanged across eligible markets:

- All users receive the same base rate up to a predetermined supply limit.
- If the total supply exceeds this limit, the rate adjusts to maintain a fixed daily distribution.
- Multipliers or divisors may apply to specific tokens, detailed in the [Fiet forum](#).

These programs ensure equitable and predictable reward distribution across market activities.

How Rewards Work

The Fiet Protocol enables users to automatically earn rewards by participating in incentivised markets, such as through market making via settlements or trading. These activities are recorded on-chain, and reward amounts are calculated off-chain using this data. The Fiet DAO will elect a representative organisation to manage this computation process, submitting rewards on-chain weekly for verification and distribution.

Rewards are made claimable approximately weekly through the Universal Rewards Distributor (URD). Users can claim their earned rewards via the Fiet interface without a deadline or directly through on-chain transactions on the URD using alternative methods. This structure ensures secure, transparent, and accessible reward distribution.

Cryptography Infrastructure

The Fiet Protocol leverages Verity, an advanced composable cryptographic infrastructure developed by Usher Labs, to enable secure and private verification of liquidity and settlement data. Fiet adopts the Verity zero-knowledge Transport Layer Security (zkTLS) stack to create a Prover of data flows — integrating sensitive financial data from traditional systems, such as centralised exchanges and banks, onto the blockchain without exposing private information.

With Verity, the Prover is designed to:

1. produce high-frequency MPC-TLS proofs
2. that rollup into STARK-based zero-knowledge proofs, powered by RiscZero, then
3. verify in a public replicated and verifiable compute environment, the Internet Computer, where,
4. further public computation and state can be managed, before
5. a succinct **Threshold-ECDSA Signature** over a hash of state allows for VRL verification and cross-chain state syndication

This cryptography infrastructure ensures the integrity and transparency of on-chain processes while upholding stringent privacy standards, supporting critical Fiet operations.

Cryptographic Components

The system encompasses several cryptographic technologies to support Fiet's operations:

- **zkTLS Proofs:** These zero-knowledge proofs, incorporating multi-party computation and STARK-based verification, securely validate data from trusted financial institutions, such as reserve liquidity amounts, while preserving confidentiality. They enable Fiet to confirm MM solvency and VRL commitments without disclosing sensitive account details.
- **Merkle Trees:** Organise the VRL state within a verifiable compute environment, facilitating efficient validation of liquidity signals across Market Chains.
- **Threshold-ECDSA Signatures:** Provide secure, decentralised signing of VRL state updates, enabling data portability and syndication to Market Chains for cross-chain verification.

These components, unified under the Fiet Prover, powered by Verity, ensure Fiet's ability to manage private data securely in a decentralised ecosystem.

Integration with Fiet Protocol

The system integrates with key Fiet Protocol features:

- **Verified Reserve Liquidity (VRL):** zkTLS proofs verify off-chain liquidity (e.g., bank accounts, exchange wallets) for VRL commitments, allowing MMs to supply liquidity without immediate on-chain settlement.
- **Value-to-Signal (VTS) Model:** Validate signalled versus settled liquidity, ensuring accurate VTS ratio calculations and settlement triggers.
- **Settlements:** zkTLS proofs of settlement intent, enable MMs to extend grace periods during RfS processes, mitigating seizure risks.
- **Custom Price Oracles:** Proofs of external price data feeds, maintaining market stability.

This integration supports privacy-preserving verification, enhancing trust and efficiency across Fiet's operations.

Security Guarantees

Verity's cryptographic infrastructure provides truth and security for Fiet Markets:

- **Data Integrity:** zkTLS proofs and Merkle trees ensure financial data remains accurate and tamper-proof during verification.
- **Privacy Protection:** Sensitive information is never exposed on-chain, complying with regulatory and institutional standards.
- **Attack Resistance:** Threshold-ECDSA signatures and decentralised verification reduce single-point-of-failure risks, protecting against malicious actors.
- **Auditability:** Cryptographic proofs enable transparent validation of protocol actions, maintaining user confidence.

These guarantees ensure Fiet's operations are secure, reliable, and compliant.

About Verity zkTLS

For detailed technical specifications of Verity, explore the documentation here:

<p>Welcome to Verity's Documentation - Usher Labs' Verity Documentation</p> <p>The home for developers and cryptographic experts exploring secure data integration with blockchains.</p> <p>▼ ▼ https://docs.verity.usher.so/</p>	
---	--

About Usher Labs

Usher Labs develops enterprise-grade data security and integration solutions for Web3 projects. The Verity infrastructure establishes verifiable data pipelines between traditional financial systems and blockchain environments, enabling Fiet to connect private data with decentralised ecosystems while ensuring trust, privacy, and compliance.

FIET — Protocol Native Token

The Fiet Protocol will launch its initial version without the FIET native token.

This token, planned for future release, will coordinate governance and incentives as the protocol becomes more decentralised. FIET will enhance the ecosystem by incentivising roles such as Provers and MMs, which start more centralised but will progressively decentralise. Once a decentralisation threshold is reached, the Fiet DAO, FIET token, and associated structures will be introduced to support a fully decentralised protocol.

The FIET token is not required for the protocol's core functionality at launch. Its introduction will significantly enhance the ecosystem's governance and incentive mechanisms once the protocol generates real value, ensuring sustainable decentralisation and participant engagement.

Design

The roles of the FIET token — governing technical mechanics, securing liquidity signals, and rewarding participants — prioritise operational utility over speculative gain. Rewards are tied to participant actions, such as trading, market making, and proving, rather than Vault economics. Slashing and emissions are managed by fixed rules to ensure predictability and network stability.

Governance

Holders of the FIET token will vote on technical parameters, such as protocol fees, token-specific base (VTS_{target}), and grace periods for settlements. Governance excludes economic controls over rewards pools, ensuring decisions focus on operational mechanics rather than profit motives. Voting requires staking FIET, with proposals needing a supermajority to pass, promoting secure and consensus-driven protocol management.

Protocol Fee

The Fiet Protocol will introduce a protocol fee, applied to the yield earned by MMs and captured during liquidity settlement. Initially set at 0%, the fee will only be adjusted through DAO governance after the Fiet DAO and FIET token launch. All fees acquired will be allocated to the Fiet DAO treasury, used to buy back FIET tokens via Fiet Markets to support protocol management, incentives, and growth. This ensures value capture occurs only when the protocol is sufficiently decentralised, enhancing ecosystem sustainability.

Prover Network

Provers in the Fiet Protocol incur costs associated with computation for cryptographic proof generation, necessitating compensation. As the protocol decentralises its Prover network, enabling infrastructure participation and allowing MMs to self-operate Provers for enhanced data privacy, a marketplace model will emerge. The marketplace will operate in an evolutionary model from Proof of Staked Authority to Proof of Stake, where model structure evolves through demands from the DAO, Prover Operators and Market Makers, whereby failure to fulfil Prover duties will result in slashing/expulsion from the protocol. MMs delegating proof generation to Provers must compensate them based on their proof of work, fostering a trust-based relationship. Failure to advance VRL state through zero-knowledge proofs results in stale on-chain data, potentially dropping liquidity signals and negatively impacting dependent MMs, with settlement guarantees further incentivising Prover reliability.

Post-launch of the Fiet DAO and FIET token, the DAO treasury will allocate a significant portion of FIET tokens to compensate Provers on behalf of MMs. This incentive scheme will reduce barriers to onboarding institutional liquidity and market makers and ensure sufficient MM participation to support protocol-level fees.

Glossary

To enhance accessibility, key terms used in the Fiet Protocol documentation are defined below:

- **Automated Market Maker (AMM):** A smart contract facilitating token swaps in a decentralised exchange using predefined price curve algorithms, supported by liquidity pools.
- **Blockchain:** A decentralised, tamper-resistant digital ledger for recording transactions across a distributed network, ensuring security and transparency.
- **Cryptographic Proofs:** Verifiable computations, such as zero-knowledge proofs or MPC-TLS proofs, ensuring data integrity and privacy in Fiet's operations, like VRL verification or settlement intent.
- **Data Portability:** The ability to securely transfer verified data, such as VRL state, across Market Chains using cryptographic signatures (e.g., tECDSA).
- **Decentralised Exchange (DEX):** A peer-to-peer trading platform for digital assets, operating without intermediaries and typically powered by AMMs.
- **Fiet DAO:** The planned decentralised governance body for the Fiet Protocol, managing technical parameters and treasury post-(F I E T) token launch.
- **Fiet Markets:** AMM-based trading pools in the Fiet Protocol, pairing LCCs to facilitate secure and efficient token swaps.
- **Fiet Prover:** A zkTLS Prover, built with Verity, that generates zkTLS proofs for data verification, supporting VRL, VTS, and settlements.
- **Impermanent Loss:** Losses faced by liquidity providers in AMM pools due to price volatility and pool rebalancing, impacting returns.
- **Liquidity Commitment Certificate (LCC):** A synthetic, non-transferable asset in Fiet Markets representing MMs' VRL commitments, traded exclusively on Fiet's DEX.
- **Market Maker (MM):** A participant facilitating liquidity for Fiet Markets, committing VRL and managing settlements, incentivised by rewards and fees.
- **Market Chain:** A blockchain network integrated with Fiet Markets, where VRL state is syndicated and verified for cross-chain operations.
- **Merkle Tree:** A cryptographic data structure organising VRL state in a verifiable compute environment, enabling efficient validation across Market Chains.
- **Proof of Settlement:** A cryptographic proof demonstrating an MM's intent to settle liquidity, used to extend grace periods during RfS processes.
- **Request for Settlement (RfS):** A condition triggered when additional liquidity is needed, managed by Fiet's VTS manager.
- **Rehypothecation:** The reuse of committed liquidity across multiple markets or platforms, adapted in Fiet to optimise capital efficiency while managing settlement risks.
- **Settlement Guarantor:** A participant (MM or bot) settling an MM's RfS obligation, seizing their liquidity position for profit, incentivised by collateral.
- **Threshold-ECDSA (tECDSA) Signatures:** Decentralised signatures in zkTLS system, signing VRL state updates for secure cross-chain syndication.
- **Universal Rewards Distributor (URD):** A Fiet Protocol component enabling weekly reward claims for user activities (e.g., market making, trading).
- **Value-to-Signal (VTS) Model:** A mechanism calculating the ratio of settled to committed liquidity, driving settlement triggers.
- **Verifiable Compute Environment:** A secure platform, such as the Internet Computer, processing zkTLS proofs and managing VRL state in Fiet.
- **Verified Reserve Liquidity (VRL):** Off-chain liquidity committed by MMs, verified via Verity's zkTLS system, enabling capital-efficient market making.
- **Zero-Knowledge Proofs:** Cryptographic methods allowing data verification (e.g., reserve liquidity) without revealing sensitive details, ensuring privacy and security.
- **zkTLS:** Zero-knowledge Transport Layer Security proofs, verifying private financial data at for Fiet's operations.