2025, 22nd Sept – Dublin, Republic of Ireland

Index Maker

Component-Level Consensus

A Blueprint for Distributed Financial Services

author: Sonia Sadhbh Kolasinska

Codebase Location

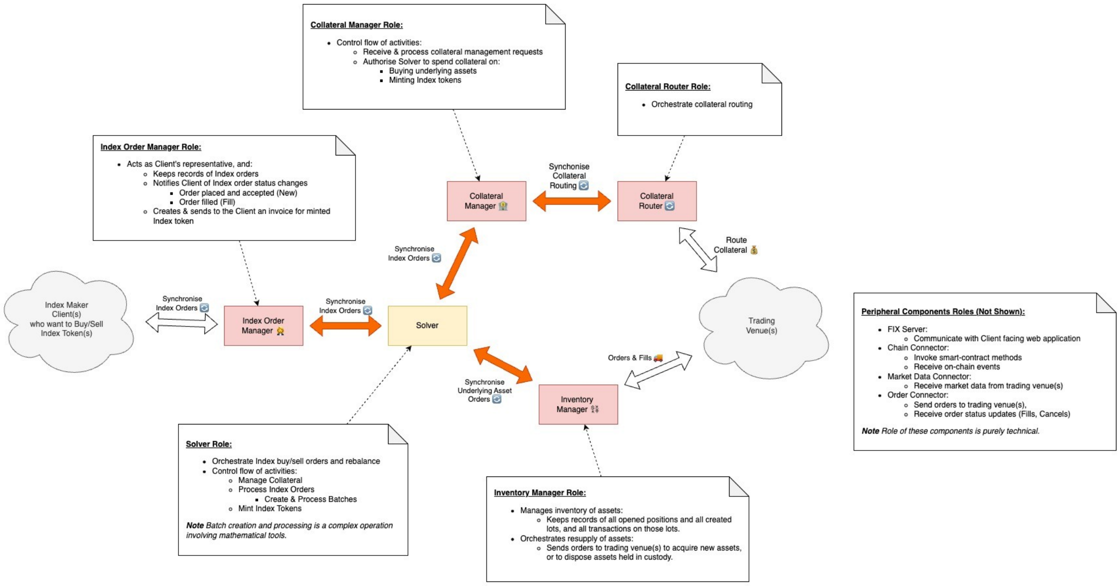
https://github.com/IndexMaker/index-maker

Abstract

This paper presents a formal analysis of the inter-component consensus protocol governing a sophisticated financial trading system. We examine how a distributed network of specialized components— including the Server, Chain Connector (CC), Index Order Manager (IOM), Solver, Strategy, Batch Manager (BM), Collateral Manager (CM), Collateral Router (CR), Collateral Bridge (CB), Inventory Manager (IM), and Order Tracker (OT) — collaborates to ensure atomic transaction finality. The protocol, built on a series of nested consensus phases and underpinned by comprehensive lot-based accounting, guarantees immutability and auditable proof of consensus. This blueprint provides a foundational understanding for a future where these messaging primitives are transitioned onto a public blockchain, thereby extending trustless guarantees to the component level.

1. Introduction and Component Roles

The design of modern financial systems increasingly prioritizes modularity and resilience. Our system is a complex, distributed model that leverages specialized components to manage every phase of a transaction lifecycle. While some components may be tightly coupled for performance, their communication is governed by a well-defined protocol that ensures data integrity and transactional atomicity. This document serves as a blueprint for that protocol, detailing the role of each component and the consensus flow that underpins the entire system.


Index Maker Client(s) who want to Buy/Sell Index Token(s)
Synchronise Index Orders 🔄
Index Order Manager
Synchronise Index Orders 🔄
Solver
Synchronise Index Orders 🔄
Collateral Manager
Synchonise Collateral Routing 🔄
Collateral Router 🔄
Route Collateral 💰
Trading Venue(s)
Orders & Fills
Synchronise Underlying Asset Orders 🔄
Inventory Manager

Collateral Manager Role:

- Control flow of activities:
  - Receive & process collateral management requests
  - Authorise Solver to spend collateral on:
    - Buying underlying assets
    - Minting Index tokens

Collateral Router Role:

- Orchestrate collateral routing

Index Order Manager Role:

- Acts as Client's representative, and:
  - Keeps records of Index orders
  - Notifies Client of Index order status changes
    - Order placed and accepted (New)
    - Order filled (Fill)
  - Creates & sends to the Client an invoice for minted Index token

Peripheral Components Roles (Not Shown):

- FIX Server:
  - Communicate with Client facing web application
- Chain Connector:
  - Invoke smart-contract methods
  - Receive on-chain events
- Market Data Connector:
  - Receive market data from trading venue(s)
- Order Connector:
  - Send orders to trading venue(s),
  - Receive order status updates (Fills, Cancels)

Note *Role of these components is purely technical.*

Solver Role:

- Orchestrate Index buy/sell orders and rebalance
- Control flow of activities:
  - Manage Collateral
  - Process Index Orders
    - Create & Process Batches
  - Mint Index Tokens

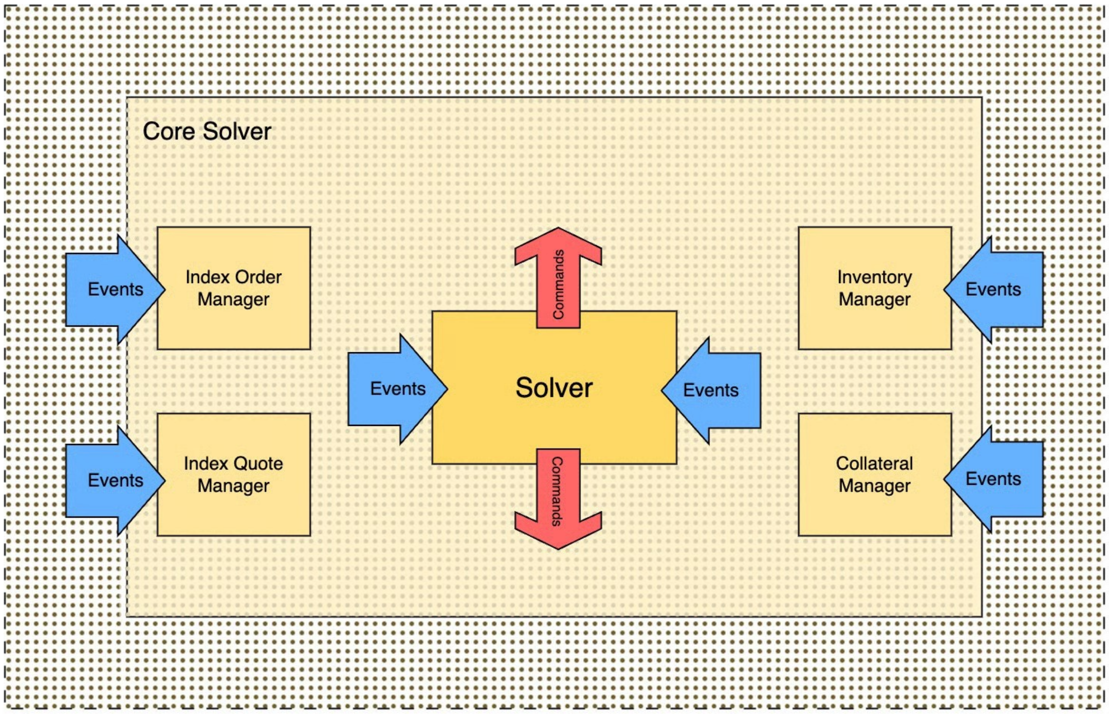
Note *Batch creation and processing is a complex operation involving mathematical tools.*

Inventory Manager Role:

- Manages inventory of assets:
  - Keeps records of all opened positions and all created lots, and all transactions on those lots.
- Orchestrates resupply of assets:
  - Sends orders to trading venue(s) to acquire new assets, or to dispose assets held in custody.

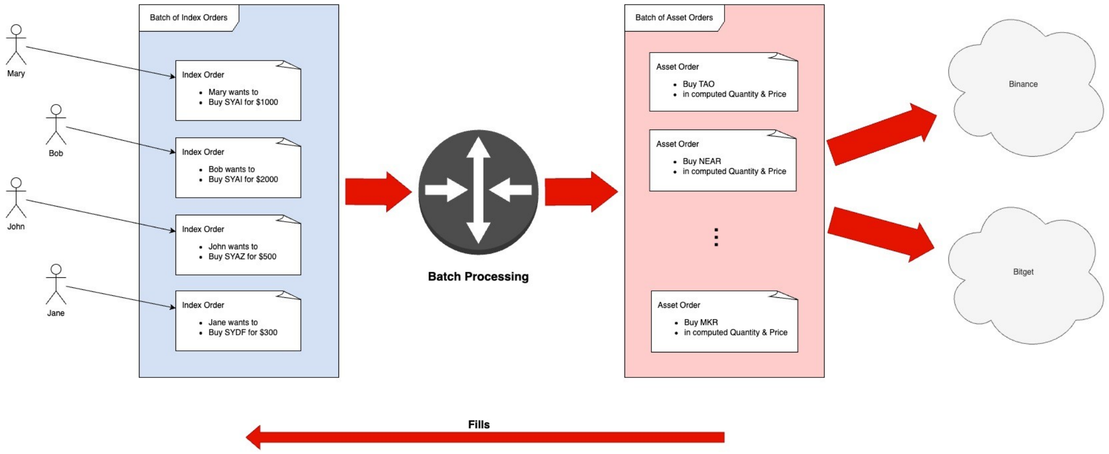
- **Server:** Initiates the process by publishing a "New Index Order" event.
- **Chain Connector (CC):** Responsible for on-chain interactions, including the final token minting request.
- **Index Order Manager (IOM):** Manages the lifecycle and status of each Index order and confirms market-readiness.
- **Solver:** The computational core; orchestrates the entire process, computes strategies, and synchronizes components.
- **Strategy:** A module used by the Solver to compute optimal trading strategies and create order batches.

- **Batch Manager (BM):** Processes batches of orders, manages fills, and determines batch completion.
- **Collateral Manager (CM):** A critical component for ensuring and managing collateral, from initial requests to final confirmation.
- **Collateral Router (CR):** Analyzes trading strategies to find the optimal path for collateral movement.
- **Collateral Bridge (CB):** Publishes "Hop Complete" events each time collateral is moved, a crucial part of the CR's routing.
- **Inventory Manager (IM):** Forwards orders to the Order Tracker for execution on trading venues.
- **Order Tracker (OT):** Sends orders to and receives fills from external trading venues.

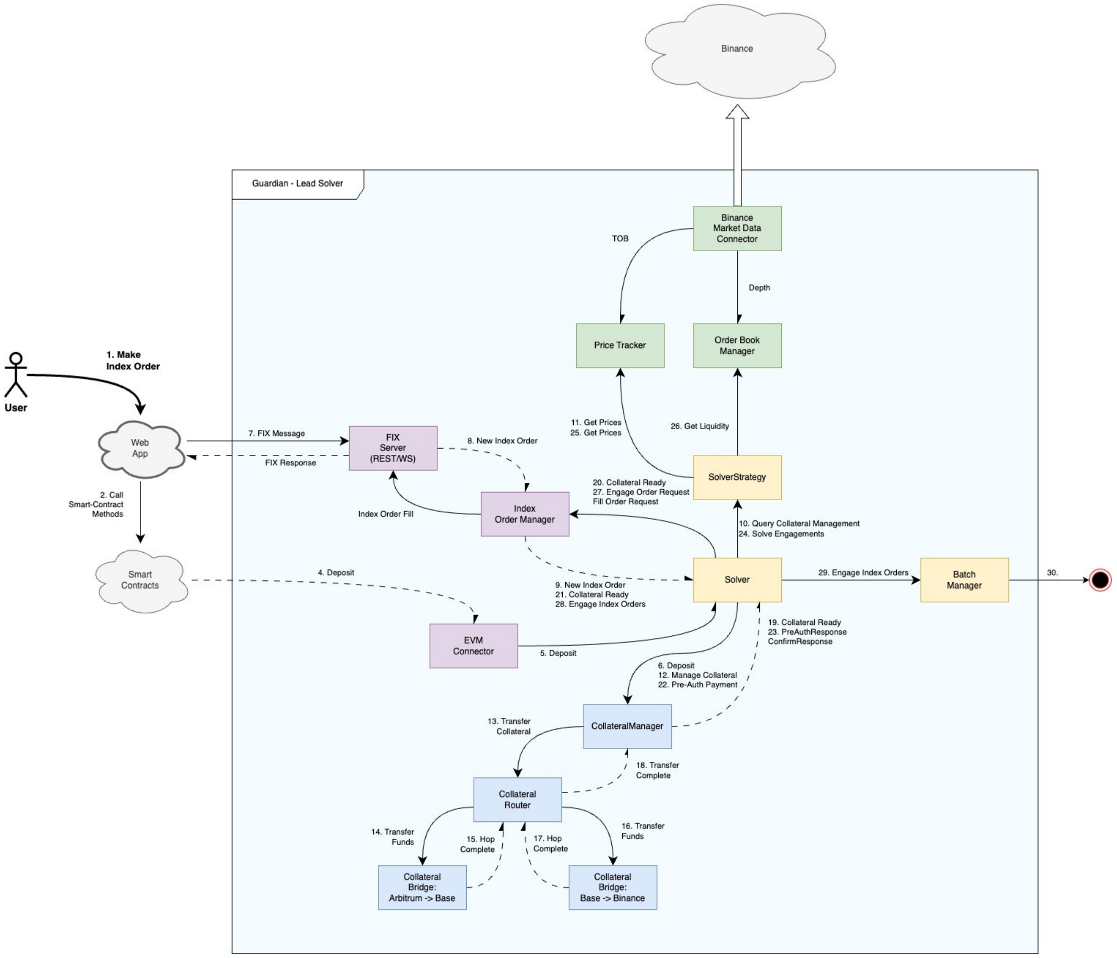


2. Phase 1: Batch Creation and Initial Collateral Management

The protocol begins with the creation of a new order and the meticulous management of collateral.



1. **Order Initiation:** The Server publishes a "New Index Order" event. The IOM can either reject the order or publish an event confirming the order is "managed."
2. **Strategy Computation:** The Solver receives the IOM's event, places the order in a queue, and then asks the Strategy module to compute a trading strategy. This strategy is crucial as it dictates the trading venues and assets to be used as collateral.
3. **Collateral Request and Position Check:** The Solver sends a collateral management request to the CM, passing in the trading strategy. The CM accepts the request and, asynchronously, checks the user's Collateral Position to ensure sufficient deposited collateral. The CM's state is updated by "Deposit" events.
4. **Collateral Routing:** Once the collateral position is confirmed, the CM sends a collateral routing request to the Collateral Router (CR). The CR analyzes the trading strategy and determines the optimal routing path, which is composed of multiple Collateral Bridges (CB). Each time a CB moves the collateral, it publishes a "Hop Complete" event.
5. **Collateral Ready and Synchronization:** When the CR completes the routing, it publishes a "Collateral Transferred" event. The CM updates the user's Collateral Position, marking the routed balance as "Ready," and publishes a "Collateral Ready" event.
6. **Pre-Commitment Consensus:** The Solver picks up the "Collateral Ready" event and initiates a synchronization phase. It sends a "PreAuth" request to the CM and a "Confirm Collateral Ready" request to the IOM. Upon receiving the CM's "Authorised" response and the IOM's "Order Status Update," a critical two-party consensus is reached.



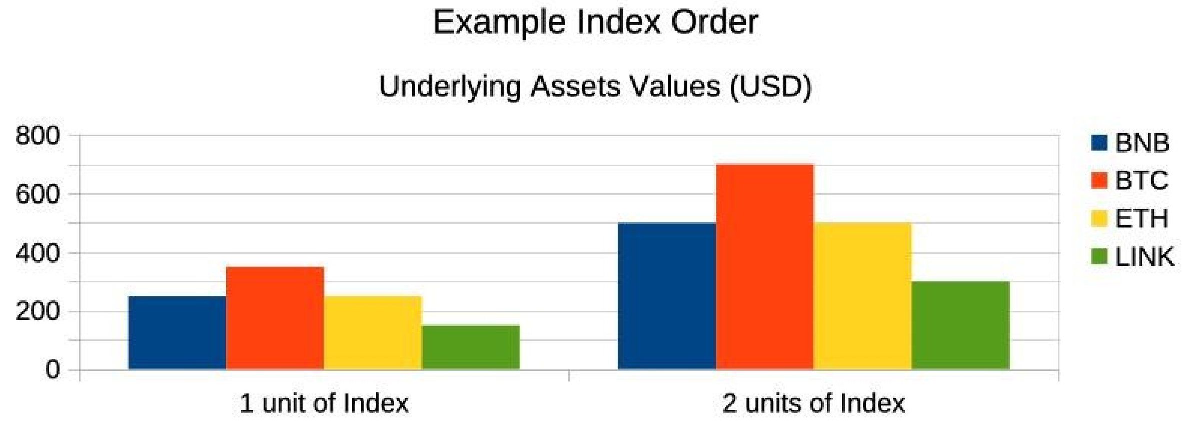
3. Comprehensive Lot Transaction Accounting

The integrity of the entire system relies on a detailed lot-based accounting system that tracks every asset and collateral movement with immutable precision.

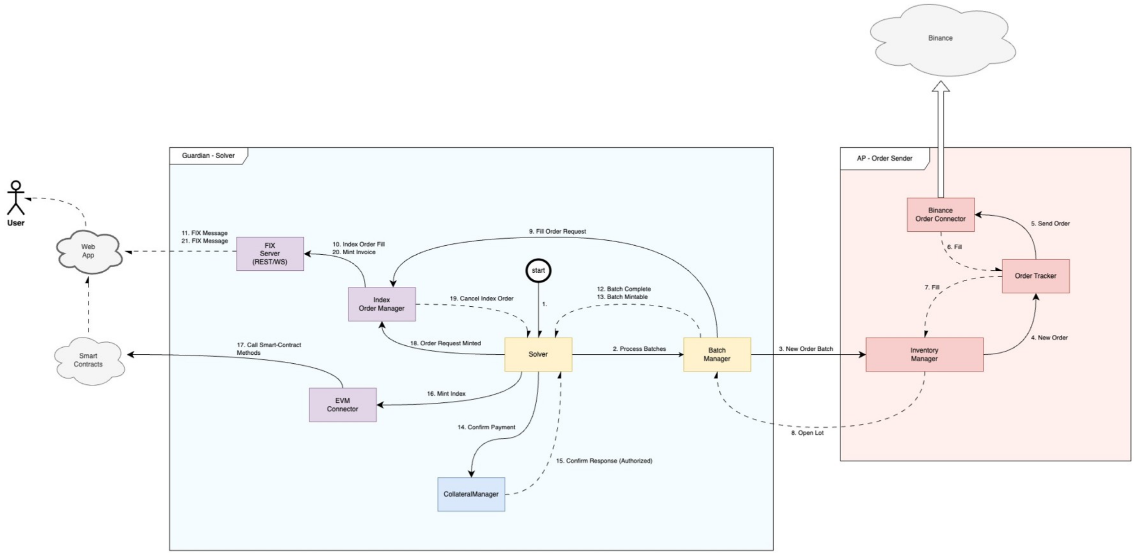
- **Collateral Manager (CM):** Each deposit received by the CM not only updates the user's Collateral Position but also creates a specific **lot** with a creation date and amount. When collateral is marked as "ready," "pre-authorized," or "spent," this corresponding lot is updated. For spends, special **spend lots** are created, providing a verifiable match between the spent collateral, the deposits, the minted Index tokens, and the Index orders.
- **Inventory Manager (IM):** The IM tracks the detailed positions of all underlying assets. Like the CM, it stores lots for each asset, updating, opening, and closing them as

trades occur. The IM is also responsible for "flipping" lots and positions from Long to Short, maintaining a precise accounting of all inventory states.

- **Index Order Manager (IOM):** The IOM manages Index Orders using **order update lots**. All new orders received from the Server either add to or reduce an existing lot associated with a specific Index Order, providing a granular, event-driven history of each order's lifecycle.



4. Phase 2: Engagement and Order Execution



With collateral confirmed and accounted for, the system moves to the execution phase.

1. **Order Engagement:** The Solver updates the order status to "Ready" and places it in a

ready queue. It then picks up the order and once again asks the Strategy module to generate a realizable order batch, or "Engagement."

2. **Parallel Broadcast:** The Solver sends the Engagement in parallel to the IOM and the Batch Manager (BM).
   - **IOM:** The IOM must lock the order for execution and confirm with an "Order Engaged" event.
   - **BM:** The BM, being tightly coupled with the Solver, simply updates its internal state without sending a confirmation.
3. **Execution and Fills:** After the IOM confirms the order is engaged, the Solver passes a message to the BM, which then relays the Engagement for processing. The BM sends the orders by matching any lots carried over from previous batches and sending requests to the Inventory Manager (IM), which forwards them to the Order Tracker (OT) for execution on trading venues. The BM receives fills (either internal or external) and computes the fill percentage for each Index order.
4. **Reporting and Batch Completion:** The BM sends the fills to the IOM via the Solver, and the IOM confirms the fills to the Server, which generates execution reports for the users. The BM completes a batch when all underlying asset orders are fully filled or cancelled.

5. Phase 3: Post-Execution and Token Minting

The final phase involves determining which orders are complete and initiating the token minting process.

1. **Batch Analysis:** Upon batch completion, the BM analyzes all Index orders within the batch and determines which are ready for minting.
2. **Event Publication:** The BM publishes two events:
   - **Batch Complete:** Includes Index orders that require more processing in a future iteration.
   - **Batch Mintable:** Includes Index orders for which all underlying assets have been acquired, making them ready for token minting.
3. **Iterative Processing (Feedback Loop):** The Solver picks up the "Batch Complete" orders and, as part of a continuous feedback loop, mixes them with new, fresh orders and sends them to the Strategy to produce the next complete "Engagement" batch, restarting the entire process for both the continued and new orders.
4. **Minting Consensus:** The Solver picks up the "Batch Mintable" orders and sends a

request to the Collateral Manager (CM) to confirm payment for the minting. The CM responds with an "Authorised" or "Not-Auth" event.

5. **Final Minting and Invoice:** Once authorized, the Solver requests minting from the Chain Connector (CC). The Solver then informs the IOM that the Index Token was minted. The IOM produces a detailed invoice containing all information about the order, the assigned lots, and the collateral used.

6. Transition to a Distributed, On-Chain Model

The described protocol is perfectly suited for a blockchain migration. By transposing the internal messaging to on-chain transactions, each event becomes an immutable, cryptographically verifiable record. The smart contract would act as the state machine, receiving intents and authorizations and enforcing the same logical constraints. This transition would imbue the system with the trustless, immutable, and provable guarantees that are foundational to the future of decentralized financial services.