Trading Terminal Architecture Diagram Review: Feedback Synthesis

The architecture diagram needs to be refined to serve as a comprehensive blueprint for the Trading Terminal. The primary goal is clarity regarding core user value propositions, critical data dependencies, and explicit architectural trade-offs.

I. Core Interaction Flows (Focus: User Value)

The diagram must visually articulate the complete lifecycle of a user's intent within the terminal, from data consumption to on-chain execution.

Flow Component	Key Message	Visual Representation Priority
Token Discovery & Browsing	Off-chain data indexing enables fast, comprehensive searching.	High: Show interaction with Indexer/Data Pipeline.
Market Data Viewing	Real-time state derived from off-chain aggregation (OHLC, Trades, Holders).	High: Detail the read path from Indexer to Frontend.
Swap Execution	Non-custodial, router-based transaction submission to on-chain liquidity.	Critical: Show Wallet -> Router -> CPI -> Liquidity Program.
Watchlist Management	User-specific state management, likely leveraging off-chain user database.	Medium: Illustrate persistence mechanism.

II. Critical External Integrations (Focus: Data & Execution Dependency)

The architecture's performance is intrinsically linked to its external data and execution partners. These must be the most prominent non-terminal components.

Integration Type	Specific Examples/Role	Dependency Level
Off-chain Indexers/Data Pipelines	Real-time token indexing, OHLC aggregation, trade history, holder analytics.	**Critical:** Data backbone for all read operations.
Solana RPC Infrastructure	Transaction submission, pre-execution simulation, confirmation checks.	**Essential:** On-chain communication layer.
Liquidity Programs (e.g., Pump, Meteora, Raydium)	Execution venues for swaps (target for Cross-Program Invocations - CPI).	**Critical:** Final execution layer.

Integration Type	Specific Examples/Role	Dependency Level
Oracles (e.g., Pyth)	Reference pricing, not used for transaction execution.	Ancillary/Reference only.

III. Edge Case Representation and Readability

Avoid complexity through exhaustive error modeling. Focus on showcasing awareness of common, high-impact execution failures within the core swap flow.

Failure Type	Representation Method	Rationale
Slippage Exceeded	Single decision point/branch after simulation/pre-execution checks.	Demonstrates risk management mechanism.
Insufficient Liquidity	Single decision point/branch at the router/execution layer.	Demonstrates router intelligence and execution venue limitations.
Transaction Failure (RPC/On-chain)	General failure path post-submission to RPC.	Covers network/program execution errors.

IV. Architectural Trade-offs & Design Decisions

The diagram should implicitly or explicitly communicate the strategic choices made during development, which define the terminal's competitive advantage and operational profile.

Design Decision	Trade-off/Rationale	Terminal Differentiation
Non-Custodial Execution	Maximize decentralization/user control; potentially higher transaction friction.	Differs from centralized exchange models.
Heavy Off-chain Data Reliance	Maximize speed and UX; increase dependency on centralized indexer infrastructure.	Enables "real-time" token discovery and fast OHLC rendering.
Router-based Multi-DEX Execution	Maximize execution price/liquidity; introduce complexity in transaction construction (CPIs).	Improved pricing vs. single-DEX platforms.
Atomic Transaction Guarantees	Maximize execution reliability; potential for larger/more complex transactions.	Reliable execution over multiple steps.

https://excalidraw.com/#room=d83f64d87c486822d19b,l9O-BJhjro2aoBbtoivywg