On/Off Ramp Order Book PRD

1. Feature Introduction

Overview

The Scanopy On/Off Ramp Order Book is an atomic swap system that enables large-scale trading between CNPY and stablecoins (USDC/USDT) across Ethereum and Solana networks. Unlike the AMM's automated market maker model, this system uses sell-only order books where makers list CNPY for sale at their desired prices, and takers can aggregate multiple orders to fulfill large trades with minimal slippage.

Problem Statement

The AMM model, while efficient for most trades, creates significant slippage for large orders due to the constant product formula. Institutional traders and large holders need a mechanism to execute size without massive price impact. Additionally, users need efficient bridges between external chains (Ethereum/Solana) and the Canopy ecosystem without relying solely on liquidity pools that may have insufficient depth for large transactions.

2. Principles & Objectives

Goals

- Large Order Execution: Enable trades > $100,000 with minimal slippage
- Price Discovery: Market-driven pricing through competitive order placement
- Cross-Chain Efficiency: Direct atomic swaps between Ethereum/Solana and Canopy
- Maker Protection: Ensure makers can set and forget orders safely
- Taker Convenience: One-click aggregation across multiple orders

Success Metrics (KPIs)

- Daily volume through order books > $100,000
- Average slippage for $100k+ orders < 5%
- Order fill rate > 80%
- Cross-chain swap success rate > 99%
- Time to settlement < 3 minutes
- Supporting all token pairs (focus on CNPY/stablecoin) - *liquidity concentration is important on CNPY/USDC/T*

Non-Goals

- Providing lending or leverage
- Implementing complex order types initially
- Acting as custodian for user funds
- Competing with AMM for small trades
- Partial fills

3. Users

Primary Personas

1. Large Holder (Maker)

- Background: CNPY whale looking to exit positions
- Technical Level: Medium to High
- Needs: Sell large amounts without crashing price, set specific rates
- Pain Points: AMM slippage, finding counterparties, order management

2. Institutional Buyer (Taker)

- Background: Fund or large trader entering CNPY positions
- Technical Level: High
- Needs: Acquire size quickly, minimal price impact, cross-chain access
- Pain Points: Fragmented liquidity, high slippage, slow OTC deals, OTC trust

3. Arbitrage Trader

- Background: Professional seeking price discrepancies
- Technical Level: High
- Needs: Fast execution, API access, real-time data
- Pain Points: Latency, gas costs, partial orders

Key User Stories

1. As a large holder, I want to list my CNPY for sale at specific prices and have them automatically execute when buyers appear, so I can exit positions without actively managing trades.
2. As an institutional buyer, I want to purchase large amounts of CNPY by aggregating multiple sell orders in a single transaction, so I can build positions efficiently.
3. As an arbitrage trader, I want programmatic access to order book data and execution, so I can profit from price differences between the order book and AMM.

4. Requirements

P0 - Must Have

Order Book Core

- Sell-only order placement for CNPY
- Order pricing in USDC/USDT
- Order size limits (min/max)
- Order expiration options
- Order cancellation mechanism

Supported Trading Pairs

- Ethereum USDC → CNPY
- Ethereum USDT → CNPY
- Solana USDC → CNPY
- Solana USDT → CNPY

Atomic Swap Engine

- Atomic execution guarantee
- No partial fills initially
- Automatic settlement

Order Aggregation

- Multi-order selection interface
- Combined order execution
- Total cost calculation
- Slippage estimation
- Single transaction workflow (if possible)

Maker Interface

- Create sell orders with custom pricing
- View active orders and status
- Edit order prices (cancel and recreate)
- Track order history
- Withdraw proceeds automatically

Taker Interface

- Real-time order book display
- Price level aggregation
- Liquidity depth visualization
- One-click multi-order buying

- Transaction tracking

P1 - Great to Have

Advanced Order Features

- Programmatic orders
- Bulk order management (create more than one sell)

Analytics Dashboard

- Historical order book depth
- Price charts and spreads
- Volume analytics
- Maker/taker ratios
- Cross-chain flow analysis

API Access

- REST API for order book data
- WebSocket real-time feeds
- Order placement API
- Historical data endpoints
- Rate limiting controls

P2 - Nice to Have

Enhanced Trading

- Partial fill support (might not be technically viable)
- Advanced order matching
- Priority gas auctions
- Order Expiration

Integration Features

- AMM arbitrage tools
- Professional trading UI
- Mobile app support
- Telegram/Discord bots
- Price alerts

P3 - If We Have Time

Market Making Tools

- Automated market making bots (provisioning liquidity)

5. Technical Architecture

System Components

Order Book Engine

- Order management system
- Price-time priority matching
- State management
- Order validation

Cross-Chain Infrastructure

- Oracle price feeds (if we want USD values)

Settlement System

- Atomic swap coordinator
- Lock and release mechanism
- Timeout handling
- Dispute resolution
- Fee collection

Backend Services

- Order book aggregator
- Price feed service
- WebSocket server
- API gateway
- Database layer

Atomic Swap Flow

```
None
1. MAKER FLOW
2. ├── Connect Wallet
3. ├── Approve CNPY
4. ├── Create Sell Order
5. |    ├── Set Amount
6. |    ├── Set Price
7. |    └── Set Expiration
8. ├── Lock CNPY in Contract
9. └── Wait for Execution
```

```
10.                  ↓
11. TAKER FLOW
12. ├── View Order Book
13. ├── Select Orders
14. ├── Calculate Total
15. ├── Approve USDC/USDT
16. ├── Initiate Swap
17. └── Confirm Transaction
18.                  ↓
19. SETTLEMENT PROCESS
20. ├── Verify Oracle Prices
21. ├── Lock Taker Funds
22. ├── Tx Verification
23. ├── Release CNPY to Taker
24. ├── Release USDC to Maker
    └── Emit Events
```

Security Considerations

Oracle Security (USD pricing)

- Multiple oracle sources
- Price deviation checks
- Manipulation protection
- Heartbeat monitoring
- Fallback mechanisms

Integration Requirements

AMM Integration

- Price comparison display
- Arbitrage opportunity alerts
- Combined liquidity metrics
- Routing optimization

Wallet Integration

- Multi-chain connections
- Transaction signing

- Balance checking
- Approval management

- Order book metrics
- Trading volume tracking
- Historical data
- Maker/taker analytics

6. User Experience Design

Maker Flow

1. Order Creation
   - Connect wallet with CNPY
   - Select amount to sell
   - Set price per CNPY
   - Choose expiration (optional)
   - Review gas costs
   - Confirm and sign
2. Order Management
   - View active orders
   - Monitor fill status
   - Cancel if needed
   - Track proceeds
   - Claim stablecoins

Taker Flow

1. Order Discovery
   - View order book depth
   - Sort by price/size
   - Filter by chain
   - Check total liquidity
   - Compare to AMM prices
2. Order Execution
   - Select multiple orders
   - Review aggregated price
   - Approve stablecoin spend
   - Execute atomic swap
   - Monitor settlement

Interface Design

```
25. ORDER BOOK VIEW
26. ┌─────────────────────────────────┐
27. │ CNPY/USDC Order Book            │
28. ├─────────────────────────────────┤
29. │ SELLS                           │
30. │ Price      Size      Total      │
31. │ $0.52      50,000    $26,000    │
32. │ $0.51      100,000   $51,000    │
33. │ $0.50      200,000   $100,000   │
34. ├─────────────────────────────────┤
35. │ Spread: $0.01                   │
36. │ Total Depth: 350,000 CNPY       │
37. └─────────────────────────────────┘
38.
39. AGGREGATION INTERFACE
40. ┌─────────────────────────────────┐
41. │ Build Your Order                │
42. ├─────────────────────────────────┤
43. │ ☑ 50,000 @ $0.50  = $25,000     │
44. │ ☑ 100,000 @ $0.51 = $51,000     │
45. │ ☐ 200,000 @ $0.52 = $104,000    │
46. ├─────────────────────────────────┤
47. │ Total: 150,000 CNPY             │
48. │ Cost: $76,000 USDC              │
49. │ Avg Price: $0.506               │
50. │ [Execute Swap]                  │
    └─────────────────────────────────┘
```

Key UX Principles

- Transparency First: Clear pricing and fees
- Aggregation Simplicity: Easy multi-order selection
- Price Comparison: Always show AMM alternative (for swaps with AMM pair)
- Reduce Friction: Batch where possible
- Status Visibility: Real-time order updates

7. Metrics and Analytics

Liquidity Metrics

- Order Book Depth: Total value at each price level
- Spread Analysis: Bid-ask spread over time
- Time to Fill: Average order lifetime
- Cancel Rate: Percentage of cancelled orders

Volume Metrics

- Daily Volume: USD traded through order book
- Cross-Chain Distribution: Volume per chain
- Average Trade Size: Mean and median
- Maker/Taker Split: Origin of liquidity
- Peak Trading Times: Hourly volume patterns

Efficiency Metrics

- Settlement Time: Cross-chain completion
- Gas Costs: Per trade average
- Failed Transaction Rate: Technical failures

User Metrics

- Active Makers: Unique order creators
- Active Takers: Unique buyers
- Repeat Usage: Returning user rate
- Order Sizes: Distribution analysis
- Chain Preferences: Most used bridges

8. Open Questions / Next Steps

Pending Decisions

- Minimum order size requirements
- Fee structure (maker/taker fees) *if any*
- Oracle provider selection *if required*
- Emergency pause conditions *if desired*
- Supported stablecoin expansion timeline

Considered but Deferred

- Two-Sided Order Book: Allow buy orders for CNPY
- Partial Fills: Complex order splitting

- Advanced Order Types: Stop-loss, trailing stops
- Margin Trading: Leveraged positions
- Fiat Integration: Direct bank transfers
- Dark Pools: Hidden liquidity pools
- Clearing House: Centralized settlement

Next Steps

1. Select and integrate oracle providers
2. Develop cross-chain message protocol
3. Build order matching engine
4. Design API specifications
5. Plan liquidity bootstrapping strategy
6. Develop monitoring and alert systems
7. Create user documentation and tutorials