Scanopy AMM PRD

1. Feature Introduction

Overview

The AMM is a Canopy-routed liquidity system that enables seamless trading across all Nested Chains using CNPY as the universal hub asset. It provides automated price discovery, liquidity provision incentives, and cross-chain swap routing through a single unified interface.

Problem Statement

Current multi-chain environments force users to manage liquidity across isolated pools, navigate complex bridging mechanisms, and suffer from fragmented liquidity that results in high slippage. Liquidity providers struggle to identify optimal yield opportunities across chains, while traders face poor user experience when executing cross-chain swaps.

2. Principles & Objectives

Goals

- Universal Liquidity: Establish CNPY as the hub asset for all trading pairs
- Cross-Chain Simplicity: Enable single-interface trading across all Nested Chains
- Optimal Routing: Automatically find best execution paths for swaps
- LP Incentivization: Maximize yields for liquidity providers through fee distribution
- Price Efficiency: Minimize slippage through aggregated liquidity pools

Success Metrics (KPIs)

- Total Value Locked (TVL) > $10M within 6 months
- Daily trading volume > $1M
- Average slippage < 3% for trades under $10,000
- Number of active liquidity providers > 1,000
- Cross-chain swap success rate > 99%

Non-Goals

- Supporting non-Canopy ecosystem tokens initially
- Providing lending/borrowing functionality
- Implementing complex derivatives trading
- Managing user funds custody

3. Users

Primary Personas

1. Retail Trader

- Background: Individual seeking to swap tokens across chains
- Technical Level: Low to medium
- Needs: Simple interface, best prices, fast execution
- Pain Points: Complex routing, high fees, failed transactions

2. Liquidity Provider

- Background: Yield-seeking investor with idle capital
- Technical Level: Medium to high
- Needs: High yields, risk management, easy position management
- Pain Points: Impermanent loss, complex reward claiming, low yields

3. Arbitrage Trader

- Background: Professional trader exploiting price differences
- Technical Level: High
- Needs: Low latency, deep liquidity, API access
- Pain Points: MEV competition, execution delays, limited liquidity

Key User Stories

1. As a trader, I want to swap between any assets across Nested Chains without managing multiple chains manually so that I can focus on trading rather than infrastructure.
2. As a liquidity provider, I want to maximize yields by providing liquidity across multiple chains so that I can earn optimal returns on my capital.
3. As an arbitrage trader, I want programmatic access to liquidity pools so that I can execute profitable trades efficiently.

4. Requirements

P0 - Must Have

CNPY Hub Model

- Implement all pools with CNPY as base pair
- Support automatic CNPY routing for any-to-any swaps
- Handle slippage protection

Pool Management

- Create pools automatically for graduated launchpad tokens
- Support standard constant product (x*y=k) formula
- Implement base trading fee (fee tbd, may be 0 at launch)
- Track pool metrics (TVL, volume, APY)
- Handle liquidity addition/removal

Swap Interface

- Single-page swap interface
- Real-time price quotes
- Slippage tolerance settings
- Transaction preview with fees
- Cross-chain routing visualization

LP Interface

- Position dashboard showing all LP holdings
- Add/remove liquidity workflows
- Reward claiming mechanism
- Historical performance tracking

Fee Distribution

- To liquidity providers (83.3%)
- To CNPY treasury/burn (16.7%)
- Automatic fee compounding option
- Real-time fee accrual tracking

P1 - Great to Have

Advanced Routing

- Priority fee adjustment

LP Enhancements

- Single-sided liquidity provision
- Auto-rebalancing strategies
- LP NFT positions
- Yield farming boost mechanics

Analytics Dashboard

- Pool performance metrics
- Volume/TVL charts
- Top traders leaderboard

- Liquidity depth visualization
- Price impact calculator

P2 - Nice to Have

Advanced Order Types

- Limit orders
- Stop-loss orders

Professional Tools

- API for programmatic trading
- WebSocket price feeds (?)
- Historical data export
- Custom trading strategies
- Backtesting environment
- Impermanent loss calculator

P3 - If We Have Time

Social Features

- Copy trading functionality
- LP strategy sharing
- Trading competitions

Advanced Order types

- DCA (Dollar Cost Averaging) automation
- TWAP (Time-Weighted Average Price) execution
- Conditional orders

5. Technical Architecture

System Components

Services

- LP token
- Fee distributor

Backend Services

- Price calculation engine
- Analytics aggregator

Frontend

- Trading interface built into explorer
- Real-time price updates via WebSocket
- Mobile-responsive design
- Multi-wallet support (metamask?)

Integration Requirements

Launchpad Integration

- Automatic pool creation for graduated tokens
- Initial liquidity from bonding curve
- Launch token price feeds
- Reward distribution

Explorer Integration

- Pool metrics display
- LP position tracking
- Trading history
- Yield opportunity ranking

Wallet Integration

- Multi-chain transaction signing
- Balance aggregation
- Approval management
- Position tracking

On/Off Ramp Integration

- Atomic swap order book system
- Sell-only order books (CNPY for USDC/USDT)
- Cross-chain support (Ethereum/Solana)
- Order aggregation and fulfillment
- Oracle-based architecture

6. User Experience Design

Swap Flow

1. Navigate to swap interface or chain view
2. Select input token and chain (if not prefilled)
3. Select output token and chain (if not prefilled)
4. Enter swap amount
5. View estimated price impact

6. Confirm transaction
7. Track swap progress
8. Receive confirmation

LP Flow

1. Select pool to provide liquidity
2. Choose balanced or single-sided
3. Enter amounts
4. Review pool share and fees
5. Add liquidity
6. Receive LP tokens
7. Monitor position

Order Book Interface (On/Off Ramp)

1. View real-time order book
2. See available liquidity at price levels
3. Select multiple orders for fulfillment
4. Execute atomic swap
5. Track order status

Key UX Principles

- Simplicity First: Hide complexity unless requested
- Transparent Pricing: Always show fees and slippage
- Fast Feedback: Real-time updates on all actions
- Error Prevention: Validate inputs before execution
- Mobile-First: Optimize for mobile trading

7. Metrics and Analytics

Trading Metrics

- Daily Volume: Total USD traded per day
- Unique Traders: Daily/monthly active users
- Average Trade Size: Median and mean trade values
- Cross-Chain Volume: Percentage of cross-chain swaps

Liquidity Metrics

- Total Value Locked: USD (or CNPY) value in all pools
- Pool Distribution: TVL concentration across pools
- LP Count: Active liquidity providers
- Fee APY: Average yields for LPs

- Liquidity Utilization: Volume to TVL ratio

Performance Metrics

- Slippage: Average price impact by trade size
- Execution Time: Time from initiation to completion
- Gas Efficiency: Average gas cost per swap
- System Uptime: Availability percentage

On/Off Ramp Metrics

- Open Interest: Total value in order books
- Order Fill Rate: Percentage of orders completed
- Average Spread: Difference between bid/ask
- Cross-Chain Volume: USDC/USDT to CNPY flow

8. Open Questions / Next Steps

Pending Decisions

- Fee tier structure (single vs multiple tiers)
- Impermanent loss protection mechanisms
- Liquidity mining reward distribution (claimable?)

Considered but Deferred

- Order Book Integration: Hybrid AMM/order book model
- Flash Loans: Uncollateralized lending for arbitrage
- Options Trading: Automated options market maker
- Prediction Markets: Binary outcome trading
- Privacy Features: Anonymous swap functionality
- Partial Order Fills: Support for partially filled orders in order book
- Concentrated liquidity ranges
- Lending market integration
- Insurance protocols

Next Steps

1. Complete development and testing
2. Design liquidity bootstrapping program
3. Develop routing algorithm optimization
4. Create LP incentive structure
5. Build analytics and monitoring infrastructure
6. Implement atomic swap order book system
7. Plan mainnet launch sequence