# CMPE 492
# Analysis of Market Dynamics of Crypto Exchanges: A Comparative Study of CEX and DEX Markets

Yusuf Akin
Halil Utku Çelik
Cenk Yilmaz
**Advisor**: Can Özturan

November 2025

# Contents

**1 INTRODUCTION** **1**

1.1 Background: Centralized Exchange Market Structure . . . . . 1

1.1.1 Traditional CEX Orderbook Mechanics . . . . . . . . . 1

1.1.2 Binance and Modern CEX Infrastructure . . . . . . . . 3

1.1.3 On-Chain Orderbook Implementations . . . . . . . . . 7

1.1.4 Market Volume and Depth: CEX vs DEX . . . . . . . 13

1.1.5 Maximal Extractable Value (MEV) . . . . . . . . . . . 15

1.2 Broad Impact . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

1.3 Ethical Considerations . . . . . . . . . . . . . . . . . . . . . . 16

**2 PROJECT DEFINITION AND PLANNING** **18**

2.1 Project Definition . . . . . . . . . . . . . . . . . . . . . . . . . 18

2.2 Project Planning . . . . . . . . . . . . . . . . . . . . . . . . . 20

2.2.1 Project Time and Resource Estimation . . . . . . . . . 20

2.2.2 Success Criteria . . . . . . . . . . . . . . . . . . . . . . 21

2.2.3 Risk Analysis . . . . . . . . . . . . . . . . . . . . . . . 21

2.2.4 Team Work . . . . . . . . . . . . . . . . . . . . . . . . 21

**3 RELATED WORK** **23**

3.1 Price Discovery Between CEX and DEX . . . . . . . . . . . . 23

3.2 DEX Market Structure and Liquidity . . . . . . . . . . . . . . 23

3.3 Liquidity and Slippage Analysis . . . . . . . . . . . . . . . . . 24

3.4 MEV and Transaction Costs . . . . . . . . . . . . . . . . . . . 24

3.5 Existing Tools and Platforms . . . . . . . . . . . . . . . . . . 25

3.6 Gap in Current Research . . . . . . . . . . . . . . . . . . . . . 25

**4 METHODOLOGY** **26**

4.1 CEX Price Index Calculation . . . . . . . . . . . . . . . . . . 26

4.2 DEX Trade Monitoring . . . . . . . . . . . . . . . . . . . . . . 27

4.3 Price Deviation Measurement . . . . . . . . . . . . . . . . . . 28

4.4 Lead-Lag Correlation Analysis . . . . . . . . . . . . . . . . . . 28

4.5 Slippage Modeling . . . . . . . . . . . . . . . . . . . . . . . . 29
4.6 Predictive Models for Liquidity Rebalancing . . . . . . . . . . 30
4.7 Liquidity Depth Comparison . . . . . . . . . . . . . . . . . . . 32
4.8 Statistical Validation . . . . . . . . . . . . . . . . . . . . . . . 33

**5 REQUIREMENTS SPECIFICATION 34**
5.1 Functional Requirements . . . . . . . . . . . . . . . . . . . . . 34
5.1.1 FR1: CEX Data Collection . . . . . . . . . . . . . . . . 34
5.1.2 FR2: Price Index Calculation . . . . . . . . . . . . . . . 34
5.1.3 FR3: DEX Monitoring . . . . . . . . . . . . . . . . . . . 34
5.1.4 FR4: Price Deviation Analysis . . . . . . . . . . . . . . . 35
5.1.5 FR5: Lead-Lag Analysis . . . . . . . . . . . . . . . . . . 35
5.1.6 FR6: Slippage Analysis . . . . . . . . . . . . . . . . . . . 35
5.1.7 FR7: Liquidity Analysis . . . . . . . . . . . . . . . . . . 36
5.1.8 FR8: Data Storage . . . . . . . . . . . . . . . . . . . . . 36
5.1.9 FR9: Dashboard Visualization . . . . . . . . . . . . . . . 36
5.2 Non-Functional Requirements . . . . . . . . . . . . . . . . . . 36
5.2.1 NFR1: Performance . . . . . . . . . . . . . . . . . . . . 36
5.2.2 NFR2: Scalability . . . . . . . . . . . . . . . . . . . . . . 37
5.3 Use Cases . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
5.3.1 UC1: Monitor Real-Time Price Deviation . . . . . . . . 37
5.3.2 UC2: Analyze Historical Lead-Lag Relationship . . . . 37
5.3.3 UC3: Estimate Slippage for Planned Trade . . . . . . . 38
5.3.4 UC4: Compare CEX vs DEX Liquidity . . . . . . . . . 38

**6 DESIGN 39**
6.1 Information Structure . . . . . . . . . . . . . . . . . . . . . . . 39
6.1.1 Entity-Relationship Model . . . . . . . . . . . . . . . . . 39
6.2 Information Flow . . . . . . . . . . . . . . . . . . . . . . . . . 42
6.2.1 Activity Diagram: Real-Time Price Deviation Monitoring 42
6.2.2 Sequence Diagram: Price Index Calculation . . . . . . . 42
6.2.3 Sequence Diagram: DEX Trade Processing . . . . . . . 43
6.3 System Design . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
6.3.1 High-Level Architecture . . . . . . . . . . . . . . . . . . 43
6.3.2 Component Details . . . . . . . . . . . . . . . . . . . . . 45
6.3.3 Data Flow Patterns . . . . . . . . . . . . . . . . . . . . . 46
6.4 User Interface Design . . . . . . . . . . . . . . . . . . . . . . . 46
6.4.1 Dashboard Layout . . . . . . . . . . . . . . . . . . . . . 46
6.4.2 Interaction Patterns . . . . . . . . . . . . . . . . . . . . 47
6.4.3 Visual Design Principles . . . . . . . . . . . . . . . . . . 48

7 IMPLEMENTATION AND TESTING 49
7.1 Implementation . . . . . 49
7.1.1 Completed Components . . . . . 49
7.1.2 Components in Development . . . . . 50
7.2 Testing . . . . . 50
7.2.1 Unit Testing . . . . . 50
7.2.2 Integration Testing . . . . . 50
7.2.3 System Testing . . . . . 51
7.2.4 Manual QA . . . . . 51
7.3 Deployment . . . . . 51
7.3.1 Infrastructure Requirements . . . . . 51
7.3.2 Deployment Configuration . . . . . 51
7.3.3 Documentation . . . . . 52

8 RESULTS 53
8.0.1 Price Discovery Analysis . . . . . 53
8.0.2 Volume Comparison . . . . . 53
8.0.3 Liquidity Analysis . . . . . 53
8.0.4 Slippage Characterization . . . . . 54
8.0.5 Price Deviation Measurements . . . . . 54
8.0.6 Predictive Modeling and Liquidity Rebalancing . . . . . 54

9 CONCLUSION 55
9.0.1 Summary of Work Completed . . . . . 55
9.0.2 Key Contributions . . . . . 55
9.0.3 Next Steps . . . . . 56
9.0.4 Expected Outcomes . . . . . 56

A SAMPLE APPENDIX 57

# Chapter 1

# INTRODUCTION

## 1.1 Background: Centralized Exchange Market Structure

### 1.1.1 Traditional CEX Orderbook Mechanics

Centralized exchanges (CEXs) such as Binance, Coinbase, and Bybit operate using a traditional **orderbook model** that has been the foundation of financial markets for decades. Understanding this model is essential for comparing CEX and DEX market dynamics.

**Orderbook Structure:**

An orderbook is a real-time electronic list of buy (bid) and sell (ask) orders for a specific trading pair, organized by price level. Each entry contains:

- **Price**: The limit price at which a trader is willing to buy or sell
- **Quantity**: The amount of the asset available at that price
- **Side**: Whether the order is a bid (buy) or ask (sell)
- **Timestamp**: Order arrival time for time-priority matching

**Order Matching Engine:**

The matching engine operates on a **price-time priority** algorithm:

(i) **Price Priority**: Orders with better prices are matched first

  - For bids: Higher prices have priority
  - For asks: Lower prices have priority

(ii) **Time Priority**: Among orders at the same price, earlier orders are filled first

When a market order arrives, it immediately executes against the best available limit orders on the opposite side. For example, a market buy order will match with the lowest ask prices until the order is completely filled.

**Liquidity and Market Depth:**

Market depth refers to the orderbook's ability to absorb large orders without significant price impact. Deep markets have:

- Large volumes of orders at multiple price levels
- Tight bid-ask spreads (small difference between best bid and best ask)
- High liquidity concentration near the current market price

CEXs achieve deep liquidity through:

- Professional market makers providing continuous two-sided quotes
- High trading volumes attracting more participants
- Low latency infrastructure (sub-millisecond order execution)
- Institutional-grade matching engines processing millions of orders per second

**Advantages of CEX Orderbooks:**

- **Efficient Price Discovery**: Continuous order flow enables real-time price formation
- **Minimal Slippage**: Deep liquidity reduces price impact for large trades
- **Advanced Order Types**: Support for limit orders, stop-loss, take-profit, iceberg orders
- **High Throughput**: Capable of handling hundreds of thousands of orders per second

### 1.1.2 Binance and Modern CEX Infrastructure

Binance, as the world's largest cryptocurrency exchange by trading volume, exemplifies the sophistication of modern CEX infrastructure. Understanding its operational model provides insight into how centralized exchanges achieve their performance characteristics.

**System Architecture:**

Modern CEXs like Binance employ a highly optimized multi-tier architecture:

(i) **Order Gateway Layer**:

- Receives orders via REST API and WebSocket connections
- Performs initial validation (balance checks, rate limiting, authentication)
- Distributes load across multiple matching engine instances
- Latency: Typically 1-10 milliseconds for order acceptance

(ii) **Matching Engine Core**:

- Maintains in-memory orderbook for each trading pair
- Executes price-time priority matching algorithm
- Processes 1.4 million orders per second (Binance's reported capacity)
- Written in low-latency languages (C++, Java with GC optimization)
- Uses lock-free data structures for concurrent access

(iii) **Market Data Distribution**:

- Broadcasts orderbook updates via WebSocket to millions of subscribers
- Publishes trade data with microsecond timestamps
- Maintains historical data for charting and analysis
- Supports multiple data feed levels (aggregated vs. raw tick data)

(iv) **Settlement and Custody Layer**:

- Updates user balances in real-time post-trade
- Manages hot and cold wallet segregation for security

- Handles deposits and withdrawals to blockchain networks
- Implements multi-signature schemes and security protocols

**Market Making Ecosystem:**
CEXs rely heavily on professional market makers to provide liquidity:

- **Designated Market Makers**: Firms with formal agreements to maintain tight spreads
  - Typical spread requirement: 0.05-0.1% for major pairs
  - Minimum quote size requirements (e.g., $10k-100k per side)
  - Receive fee rebates or discounts (maker fees often near zero or negative)
- **High-Frequency Trading Firms**: Algorithmic traders providing passive liquidity
  - Co-located servers for minimum latency (sub-millisecond response)
  - Sophisticated inventory management and risk controls
  - Cross-exchange arbitrage to maintain price alignment
- **Retail Limit Orders**: Individual traders placing limit orders
  - Contribute to overall market depth
  - Often provide liquidity at wider spreads

**Fee Structure and Incentives:**
Binance's fee model encourages liquidity provision through a tiered VIP system:

| VIP Level | 30-Day Volume (USD) | BNB Balance | Maker Fee | Taker Fee |
|---|---:|---:|---:|---:|
| Regular User | $<$ 1M | $\geq$ 0 BNB | 0.1000% | 0.1000% |
| VIP 1 | $\geq$ 1M | $\geq$ 25 BNB | 0.0900% | 0.1000% |
| VIP 2 | $\geq$ 5M | $\geq$ 100 BNB | 0.0800% | 0.1000% |
| VIP 3 | $\geq$ 20M | $\geq$ 250 BNB | 0.0400% | 0.0600% |
| VIP 4 | $\geq$ 75M | $\geq$ 500 BNB | 0.0400% | 0.0520% |
| VIP 5 | $\geq$ 150M | $\geq$ 1,000 BNB | 0.0250% | 0.0310% |
| VIP 6 | $\geq$ 400M | $\geq$ 1,750 BNB | 0.0200% | 0.0290% |
| VIP 7 | $\geq$ 800M | $\geq$ 3,000 BNB | 0.0190% | 0.0280% |
| VIP 8 | $\geq$ 2B | $\geq$ 4,500 BNB | 0.0160% | 0.0250% |
| VIP 9 | $\geq$ 4B | $\geq$ 5,500 BNB | 0.0110% | 0.0230% |

Table 1.1: Binance VIP fee structure (standard spot trading)

**Spot Maker Program:**
For professional market makers, Binance offers additional incentives:
The maker-taker model incentivizes liquidity provision:

| **Tier** | **Maker Volume %** | **Weekly Volume (USD)** | **Maker Fee** | **Taker Fee** |
|---|---:|---:|---:|---:|
| Tier 1 | 0.05% | Or 25M | 0.0000% | Standard VIP |
| Tier 2 | 0.15% | - | -0.0040% (rebate) | Standard VIP |
| Tier 3 | 0.50% | - | -0.0060% (rebate) | Standard VIP |
| Tier 4 | 1.00% | - | -0.0080% (rebate) | Standard VIP |

Table 1.2: Binance Spot Maker Program (negative fees = rebates earned)

- **Makers** add liquidity by placing limit orders that rest in the orderbook
- **Takers** remove liquidity by executing against existing orders
- Lower maker fees (or rebates) encourage traders to provide liquidity
- This creates competitive spreads and deep orderbooks

**Order Types and Advanced Features:**
Professional trading requires sophisticated order types:

- **Limit Orders**: Buy/sell at specified price or better
- **Market Orders**: Execute immediately at best available price
- **Stop-Loss / Take-Profit**: Trigger orders at specific price levels
- **Iceberg Orders**: Display only partial quantity, hiding total size
- **Fill-or-Kill (FOK)**: Execute entire order immediately or cancel
- **Immediate-or-Cancel (IOC)**: Execute partial order immediately, cancel remainder
- **Post-Only**: Ensure order adds liquidity (cancel if would match immediately)
- **Trailing Stop**: Dynamic stop price that follows market movement

**API Infrastructure:**
Modern CEXs provide extensive API access [**?**]:

- **REST API**:
  - Query account information, balances, order history
  - Place, cancel, and modify orders

  - Rate limits: typically 1,200-6,000 requests per minute

- **WebSocket Streams**:
  - Real-time orderbook updates (full depth or top N levels)
  - Trade stream with microsecond timestamps
  - User data stream for private order/balance updates
  - Aggregate trade data for lower-frequency consumers

- **FIX Protocol** (for institutional clients):
  - Industry-standard financial protocol
  - Lower latency than REST/WebSocket
  - Dedicated connections for high-frequency traders

**Security and Risk Management:**
CEXs implement multiple layers of security:

- **Custody Security**:
  - Cold wallet storage for 95%+ of assets
  - Multi-signature authorization for withdrawals
  - Hardware security modules (HSMs) for key management
  - Regular security audits and penetration testing

- **Trading Risk Controls**:
  - Position limits to prevent excessive concentration
  - Auto-deleveraging in perpetual futures markets
  - Circuit breakers to halt trading during extreme volatility
  - Margin call and liquidation systems

- **Compliance and KYC**:
  - Know Your Customer (KYC) verification requirements
  - Anti-Money Laundering (AML) transaction monitoring
  - Withdrawal limits based on verification level
  - Geographic restrictions based on regulatory requirements

**Performance Metrics:**
Binance's publicly reported statistics demonstrate CEX capabilities:

- **Peak Capacity**: 1.4 million orders per second
- **Average Latency**: 5-10ms from order submission to confirmation
- **Orderbook Depth**: $50M+ within 0.5% for BTC/USDT
- **Daily Volume**: $50-100B across all pairs
- **Available Pairs**: 1,500+ trading pairs
- **Uptime**: 99.95%+ (excluding scheduled maintenance)

**Centralization Trade-offs:**
While CEXs offer superior performance, they require trust in the platform:

- **Custody Risk**: Users must trust exchange with asset custody
- **Counterparty Risk**: Exchange insolvency affects all users (e.g., FTX collapse)
- **Censorship Risk**: Exchanges can freeze accounts or restrict trading
- **Privacy Concerns**: KYC requirements and transaction surveillance
- **Single Point of Failure**: Technical issues or attacks affect all users

These trade-offs motivate the development of decentralized alternatives, though DEXs face their own challenges in matching CEX performance and user experience.

### 1.1.3 On-Chain Orderbook Implementations

While traditional DEXs like Uniswap use Automated Market Makers (AMMs) with liquidity pools, recent innovations have brought orderbook-based trading to blockchain systems. **Hyperliquid** represents a breakthrough in this space, demonstrating that on-chain orderbooks can achieve performance comparable to centralized exchanges.

### Hyperliquid: A Deep Dive

Hyperliquid is a Layer-1 blockchain purpose-built for high-performance decentralized trading [?,?]. Founded by Jeff Yan and Iliensinc (Harvard alumni with backgrounds at Google and high-frequency trading firms), Hyperliquid addresses fundamental limitations of existing DeFi platforms while maintaining full decentralization and transparency.

**Core Architecture:**

Hyperliquid's technology stack consists of two main layers:

(i) **HyperCore (L1 Blockchain)**:

- Custom blockchain designed specifically for trading operations
- HyperBFT consensus algorithm (Byzantine Fault Tolerant variant)
- Optimized for order matching rather than general computation
- Processes over 100,000 orders per second
- Median block time: 0.2 seconds (sub-second finality)
- Deterministic order execution at consensus level

(ii) **HyperEVM (Execution Layer)**:

- Ethereum Virtual Machine compatible smart contract platform
- Allows developers to build DApps on Hyperliquid
- Seamlessly integrates with the trading engine
- Enables composability with trading primitives
- Lower gas fees compared to Ethereum mainnet

**On-Chain Orderbook Design:**

Unlike traditional DEXs where transactions are broadcast to a mempool and subject to MEV (Maximal Extractable Value) exploitation, Hyperliquid's orderbook operates fundamentally differently:

(i) **Order Submission**:

- Orders sent directly to validators via authenticated API
- No public mempool exposure (eliminates front-running)
- Orders included in the next block ( 200ms)
- No need to sign each individual transaction after initial setup

(ii) **Consensus-Level Matching**:

- Matching engine runs as part of block validation
- All validators execute identical matching logic
- Deterministic execution ensures consensus
- Price-time priority strictly enforced on-chain

(iii) **State Management**:

- Full orderbook state maintained on-chain
- Every order, trade, and cancellation permanently recorded
- Complete transparency and auditability
- Historical data queryable via blockchain explorer

(iv) **One-Click Trading Experience**:

- Initial wallet signature authorizes trading session
- Subsequent orders submitted without per-transaction signatures
- User experience comparable to centralized exchanges
- Security maintained through session management

**Technical Innovations:**
Several key innovations enable Hyperliquid's performance:

- **Zero Gas Fees for Trading**:
  - Users pay only trading fees (0.01% maker, 0.035% taker)
  - No gas fees for orders, cancellations, or trades
  - Protocol subsidizes validator costs through trading revenue
  - Removes friction for high-frequency trading strategies
- **Efficient State Representation**:
  - Compressed orderbook encoding to minimize storage
  - Incremental state updates rather than full snapshots
  - Pruning of filled/cancelled orders
  - Optimized data structures for fast order matching
- **MEV Protection**:
  - No public mempool eliminates traditional sandwich attacks

  - Consensus-level matching prevents validator manipulation
  - Fair ordering based on arrival time at validators
  - Transparent execution visible post-trade

- **Cross-Chain Bridge**:
  - Native bridge to Arbitrum (Ethereum L2)
  - Supports deposits of USDC, BTC, ETH, SOL
  - $1 flat withdrawal fee (no gas costs)
  - Bridge processed by validator set for security

**Trading Features:**
Hyperliquid offers professional-grade trading capabilities:

- **Perpetual Futures**:
  - Primary trading product with 100+ markets
  - Leverage up to 50x (varies by asset)
  - USDC-margined positions
  - Funding rate mechanism for price anchoring
  - Liquidation engine with insurance fund

- **Spot Trading**:
  - Direct buy/sell of cryptocurrencies
  - Native support for multiple assets
  - Shared liquidity with perpetual markets
  - Settlement in traded asset

- **Advanced Order Types**:
  - Limit orders (post-only, reduce-only options)
  - Market orders with slippage protection
  - Stop-loss and take-profit orders
  - Trailing stops for dynamic risk management
  - Time-in-force options (GTC, IOC, FOK)

- **Portfolio Margin**:

  - Cross-margining across positions
  - Offsetting long/short exposure
  - More capital efficient than isolated margin
  - Real-time margin calculation

**Liquidity Mechanisms:**
Hyperliquid employs multiple mechanisms to ensure deep liquidity:

(i) **HLP Vault (Hyperliquidity Provider)**:

- Protocol-owned market making vault
- Community-owned with no management fees
- Provides liquidity across all trading pairs
- Earns from spreads and trading fees
- 46% of protocol revenue allocated to HLP
- 4-day withdrawal lockup period

(ii) **User Vaults**:

- Individual traders can create trading vaults
- Other users deposit funds to follow strategies
- Vault creators earn 10% of profits
- 1-day withdrawal lockup
- Transparent performance metrics

(iii) **Direct Market Making**:

- Professional market makers can connect via API
- Earn maker rebates (negative fees)
- Compete with HLP vault for best spreads
- Contribute to overall market depth

**Tokenomics and Governance:**
The HYPE token plays a central role in the protocol:

- **Total Supply**: 1 billion HYPE tokens (fixed)
- **Distribution**:

  - 38.9%: Future emissions and community rewards
  - 31.0%: Genesis airdrop to early users (fully circulating)
  - 23.8%: Core contributors (locked until 2027-2028)
  - 6.0%: Hyper Foundation
  - 0.3%: Community grants

- **Utility**:
  - Staking for network security ( 2.5% APY)
  - Governance voting on protocol upgrades
  - Fee payments on HyperEVM
  - Value accrual through buyback mechanism

- **Revenue Distribution**:
  - 46%: HLP vault participants
  - 54%: Assistance Fund for HYPE buybacks
  - $1M+ daily revenue at peak
  - Creates buy pressure for token

**Validator Network:**
Hyperliquid's security relies on a validator set:

- Currently 16 active validators
- Proof-of-Stake with HYPE staking
- 1-day delegation period, 8-day unstaking (1+7 queue)
- Validators earn portion of transaction fees
- Slashing for byzantine behavior or downtime
- Planned expansion of validator set over time

**Performance Comparison with CEXs:**
Hyperliquid achieves performance metrics approaching centralized exchanges:

**Comparison with Traditional DEXs:**

**Challenges of On-Chain Orderbooks:**
Despite advantages, on-chain orderbooks face technical challenges:

| **Metric** | **Binance (CEX)** | **Hyperliquid (DEX)** |
|---|---:|---:|
| Order Latency | 5-10ms | 200ms (median) |
| Throughput | 1.4M orders/sec | 100K orders/sec |
| Trading Fees (Taker) | 0.10% | 0.035% |
| Gas Fees | N/A | $0 |
| Custody | Centralized | Self-custody |
| KYC Required | Yes | No |
| Transparency | Opaque | Fully on-chain |
| Uptime | 99.95% | 99.9%+ |

Table 1.3: Hyperliquid vs. traditional CEX performance

| **Characteristic** | **AMM DEXs (Uniswap)** | **On-Chain Orderbook (Hyperliquid)** |
|---|---|---|
| Liquidity Model | Pooled (AMM) | Orderbook with individual orders |
| Price Discovery | Algorithmic (x*y=k) | Continuous bid/ask matching |
| Execution Speed | 12+ seconds (Ethereum) | 0.2 seconds median |
| Slippage | Proportional to pool depth | Depends on orderbook depth |
| Order Types | Market swaps only | Limit, market, stop-loss, etc. |
| MEV Exposure | High (sandwich attacks) | Low (consensus-level matching) |
| Transaction Fees | High gas + trading fee | Zero gas, low trading fee |

Table 1.4: Comparison of DEX liquidity models

- **State Growth**: Orderbooks generate massive state updates requiring efficient storage
- **Validator Requirements**: High-performance nodes needed for fast order matching
- **Decentralization Trade-offs**: Currently only  16 validators secure Hyperliquid (compared to 1000s for Ethereum)
- **Network Effect**: Requires critical mass of traders to achieve competitive liquidity

### 1.1.4 Market Volume and Depth: CEX vs DEX

Understanding the scale difference between CEX and DEX markets is crucial for contextualizing this research.

**Volume Comparison:**

According to recent market data (2024):

- **CEX Monthly Volume**: Approximately $3-4 trillion across major exchanges
- **DEX Monthly Volume**: Approximately $90-120 billion (3-4% of CEX volume)
- **Historical Peak**: DEX volume reached 10% of CEX volume during DeFi peak in 2020-2021

**Volume Distribution by Exchange Type:**

- Top 5 CEXs (Binance, Coinbase, Bybit, OKX, Kraken): 80% of total CEX volume
- Top 5 DEXs (Uniswap, PancakeSwap, Curve, dYdX, Sushiswap): 75% of total DEX volume
- Long tail: Hundreds of smaller exchanges with minimal liquidity

**Asset-Specific Patterns:**
Market share varies dramatically by asset type:

| **Asset Type** | **CEX Share** | **DEX Share** |
|---|---:|---:|
| Major Pairs (BTC/USDT, ETH/USDT) | 97% | 3% |
| Stablecoins (DAI, USDC swaps) | 20% | 80% |
| Long-tail Altcoins | 40-60% | 40-60% |
| New Token Launches | 10% | 90% |

Table 1.5: Market share by asset category (approximate)

Key observations:

- DEXs dominate trading for decentralized stablecoins like DAI
- New tokens launch on DEXs first, often migrating to CEXs after gaining traction
- When tokens get CEX listings, volume typically increases 70x while DEX volume decreases
- Major assets remain heavily CEX-dominated due to deeper liquidity

**Liquidity Depth Analysis:**
For highly liquid pairs like ETH-USDT:

- **Binance Orderbook**: Typically maintains $20-50M liquidity within ±0.5% of mid-price
- **Uniswap V3 Single Pool**: Typically $5-15M liquidity within ±0.5% (single fee tier)
- **Uniswap V3 All Pools**: Combining multiple fee tiers brings total to $15-30M
- **Liquidity Ratio**: CEXs maintain approximately 2-4x deeper liquidity than DEXs for major pairs

**Implications for This Research:**
These volume and liquidity disparities have important implications:

(i) **Price Discovery**: CEXs likely lead price formation for major assets due to higher volume

(ii) **Execution Costs**: DEX trades should experience higher slippage on average

(iii) **Market Efficiency**: Larger CEX-DEX price deviations expected for less liquid pairs

(iv) **Arbitrage Opportunities**: Persistent inefficiencies may exist where DEX liquidity is thin

Our research aims to quantify these relationships and provide empirical evidence for market microstructure differences between exchange types.

### 1.1.5 Maximal Extractable Value (MEV)

Maximal Extractable Value (MEV) refers to the profit that validators, miners, or specialized actors (searchers) can extract by strategically ordering, including, or excluding transactions within blocks. On blockchain networks like Ethereum, transactions submitted to the mempool are visible to all participants before being included in a block, creating opportunities for exploitation. MEV extraction manifests in several forms: front-running (placing a transaction ahead of a pending transaction to profit from known price movements), back-running (placing a transaction immediately after another to capitalize on resulting state changes), sandwich attacks (surrounding a victim's transaction with both a front-run and back-run to extract value), and liquidations (competing to be first to liquidate under-collateralized positions in DeFi protocols).

DEX users are particularly vulnerable to MEV attacks due to the transparent nature of AMM pricing mechanisms and the deterministic execution of trades. In a sandwich attack—the most common form of MEV targeting DEXs—an attacker observes a large pending swap in the mempool, then submits two transactions: one that trades in the same direction as the victim (pushing the price unfavorably), and another that trades in the opposite direction after the victim's transaction executes (profiting from the price movement). This attack directly increases the victim's slippage beyond what the AMM formula predicts, effectively stealing value that would otherwise go to liquidity providers or remain with the trader.

## 1.2 Broad Impact

This project provides a rigorous empirical analysis of decentralized exchange market microstructure in comparison with centralized exchanges. The DeFi space, despite significant growth in total value locked and trading volume, lacks comprehensive real-time analysis tools that compare operational characteristics between CEX and DEX venues.

**Technical Contributions:**

- **Empirical Market Analysis**: Systematic measurement of price discovery mechanisms, liquidity characteristics, and execution quality across exchange types
- **Comparative Framework**: Direct comparison of orderbook-based CEX markets versus AMM-based DEX markets using consistent metrics
- **Real-time Monitoring Infrastructure**: Development of data collection and analysis pipeline for continuous market observation

**Research Value**: Understanding how DEX markets function relative to established CEX infrastructure is essential for anyone working in crypto trading, liquidity provision, or protocol development. This analysis provides quantitative data on questions that are currently answered mostly through intuition or limited sampling.

## 1.3 Ethical Considerations

**Research Transparency**: Our analysis may reveal exploitable price discrepancies between venues. We view this as acceptable because:

- Price inefficiencies in public markets are discoverable by anyone with sufficient technical capability
- Arbitrage activity improves price alignment across venues, benefiting all market participants
- Publishing research findings contributes to understanding of market structure

**Market Impact**: We acknowledge that systematic arbitrage can affect DEX liquidity providers through adverse selection. However, understanding these dynamics is necessary for protocol improvement and informed participation in DeFi markets.

# Chapter 2

# PROJECT DEFINITION AND PLANNING

## 2.1 Project Definition

**Research Objectives:**

This project conducts a systematic analysis of DeFi exchange markets, specifically examining:

(i) **Price Discovery Dynamics**:

- Measure temporal relationship between CEX and DEX price movements
- Quantify lag times using cross-correlation analysis
- Determine conditions under which DEX prices lead or lag CEX prices

(ii) **Volume Distribution**:

- Compare absolute trading volumes across CEX and DEX venues
- Analyze volume distribution by trade size
- Characterize market share across different asset pairs

(iii) **Liquidity Structure**:

- Measure available liquidity at various price levels on CEXs (orderbook depth)
- Calculate effective liquidity in AMM pools (considering pool reserves and concentrated liquidity)

- Compare capital efficiency between exchange types

(iv) **Execution Cost Analysis**:

- Model slippage as a function of trade size for DEXs
- Calculate empirical slippage distributions from historical trades
- Compare execution costs between CEX and DEX for equivalent trade sizes

(v) **Infrastructure Development**:

- Build data collection pipeline for real-time CEX and DEX monitoring
- Implement price index calculation from multiple CEX sources
- Create analysis and visualization dashboard

(vi) **Predictive Analytics for Liquidity Management**:

- Develop empirical models for predicting price movements across Uniswap V3 tick ranges
- Build probability distributions from historical trade data to estimate range transition likelihood
- Integrate CEX price index as predictive signal for slippage estimation
- Design automated liquidity rebalancing strategies based on predicted price movements
- Evaluate performance of active rebalancing vs. passive positioning strategies

**Technical Scope:**

- **CEXs**: 5-10 major exchanges for price index (Binance, Coinbase, Bybit, OKX, Gate.io, HTX, KuCoin, MEXC)
- **DEXs**: Uniswap V2/V3 primary focus; potential expansion to Sushiswap, PancakeSwap, Hyperliquid
- **Pairs**: BTC/USDT, ETH/USDT initially; expand to additional liquid pairs
- **Networks**: Ethereum mainnet, BSC

## 2.2 Project Planning

### 2.2.1 Project Time and Resource Estimation

**Development Timeline:**

| **Phase** | **Weeks** | **Deliverables** | **Status** |
|---|---|---|---|
| Infrastructure Setup | 1-4 | CEX websocket consumers, DEX transaction listeners | Complete |
| Data Pipeline | 5-8 | Price index calculation, data storage, stream processing | In Progress |
| Analysis Implementation | 9-12 | Statistical analysis, backtesting, slippage modeling | Planned |
| Visualization | 13-15 | Dashboard development (Streamlit), real-time monitoring | Planned |
| Documentation | 16 | Final report, system documentation | Planned |

Table 2.1: Project timeline and deliverables

**Technical Resources:**

- RPC access (Alchemy/Infura/QuickNode) for blockchain data
- WebSocket connections to 8+ CEX APIs
- Database/filesystem for OHLC aggregates
- The Graph API for historical DEX pool data
- Computing resources for stream processing

**Estimated Effort**: 15-20 hours per team member per week

| **Criterion** | **Metric** | **Target** |
|---|---|---|
| Data Coverage | CEX sources monitored | $\geq$ 5 exchanges |
| DEX Monitoring | Chains supported | $\geq$ 2 (Ethereum, BSC) |
| Price Index | Update frequency | $\leq$ 1 second lag |
| Historical Data | Analysis period | $\geq$ 30 days continuous |
| Price Deviation | Measurement precision | $\leq$ 0.01% accuracy |
| Slippage Model | Prediction accuracy | $R^2 \geq 0.8$ for major pairs |
| System Uptime | Monitoring availability | $\geq$ 95% during test period |
| Statistical Significance | Sample size per pair | $\geq$ 10,000 DEX trades |

Table 2.2: Project success criteria and targets

### 2.2.2 Success Criteria

### 2.2.3 Risk Analysis

| **Risk** | **Impact** | **Likelihood** | **Mitigation Strategy** |
|---|---|---|---|
| RPC Rate Limits | High | Medium | Multiple providers, request optimization, local node backup |
| CEX API Downtime | Medium | Medium | 8+ redundant sources, fallback logic, data validation |
| Data Quality Issues | Medium | Medium | Outlier detection, cross-validation, manual QA |
| Blockchain Congestion | Low | Low | Archive node queries for gap filling |
| Insufficient Data Volume | Low | Low | Extended collection period, multiple pairs |
| Technical Complexity | Medium | High | MVP approach, modular design, clear milestones |
| Infrastructure Costs | Low | Low | Free tier maximization, efficient queries |

Table 2.3: Risk analysis and mitigation strategies

### 2.2.4 Team Work

**Team Structure** (3 members):

**Division of Responsibilities:**

- **CEX Infrastructure**: WebSocket management, orderbook aggregation, price index calculation, volume analysis

- **DEX Infrastructure**: On-chain monitoring, The Graph integration, transaction parsing, pool state tracking
- **Analysis & Visualization**: Statistical analysis, backtesting framework, slippage modeling, dashboard development

**Collaboration Methods:**

- Shared Git repository with defined module interfaces
- Daily standups for blocking issues
- Code review for critical data processing logic
- Shared documentation for data schemas and API contracts

# Chapter 3

# RELATED WORK

## 3.1 Price Discovery Between CEX and DEX

Recent research has examined the price formation mechanisms between centralized and decentralized exchanges. Alexander et al. (2025) analyzed price discovery and efficiency between Uniswap liquidity pools and major centralized exchanges, finding that DEXs play a role in price formation rather than simply following CEX prices, though their efficiency varies by trading conditions [?]. The study revealed that informed traders adjust their DEX usage based on market uncertainty, switching between different fee tiers and pool versions.

Work on CEX-DEX arbitrage by Wu et al. (2025) highlights how arbitrageurs capitalize on temporary price discrepancies arising from asynchronous price discovery across venues [?]. Centralized exchanges provide high liquidity and near-instantaneous execution while decentralized exchanges experience inherent latency due to blockchain consensus mechanisms [?]. This temporal asymmetry creates systematic arbitrage opportunities.

## 3.2 DEX Market Structure and Liquidity

Research by Lehar and Parlour (2021) demonstrates that while DEXs trade significantly more unique tokens than major CEXs, they handle substantially lower volumes for established assets [?]. Their findings show that when tokens migrate from DEX-only trading to CEX listing, trading volume increases dramatically (approximately 70x) while DEX volume drops, indicating clear market segmentation between the two venue types.

Market analysis from Kaiko Research (2024) reveals that DEX monthly trade volume represents approximately 3% of CEX volume in recent periods, down from historical peaks of 10% during peak DeFi enthusiasm in 2020 [?]. However, for specific tokens—particularly stablecoins like DAI—DEXs account for over 80% of trading volume, demonstrating that market share varies significantly by asset type.

## 3.3 Liquidity and Slippage Analysis

Empirica's liquidity analysis framework provides a methodology for ranking Uniswap pools by slippage and market depth metrics [?]. Their research shows that only pools with sufficient liquidity concentrated around current prices can support meaningful trading without excessive price impact. The concentration level—the share of Total Value Locked within a narrow price range—emerges as a critical metric for assessing pool quality.

Comparative analysis between Uniswap V3 and major CEXs by Kaiko Research demonstrates that concentrated liquidity DEXs can be modeled similarly to orderbooks, with liquidity distributed across discrete price ranges [?,?]. However, for highly liquid pairs like ETH-USDT, Binance typically maintains 4x deeper liquidity at most price levels compared to individual Uniswap V3 pools, though combining multiple Uniswap pools with different fee tiers narrows this gap.

## 3.4 MEV and Transaction Costs

Capponi et al. (2024) provide comprehensive analysis of transaction costs on Uniswap V3, breaking down slippage into benign and adversarial components [?]. Their findings reveal that cost composition varies dramatically with trade characteristics: gas costs dominate for small swaps (under $1,000), while price impact and slippage account for the majority of costs on large swaps (over $100,000). The research introduces the concept of "reordering slippage" to quantify costs from adversarial transaction ordering.

Recent work by Wu et al. (2025) on CEX-DEX extracted value shows increasing centralization in arbitrage markets, with three major searchers affiliated with top block builders dominating CEX-DEX arbitrage opportunities [?]. Exclusive searcher-builder arrangements amplify centralization pressures both downstream and upstream of the MEV supply chain, raising concerns about Ethereum's decentralization guarantees.

## 3.5 Existing Tools and Platforms

**Industry Analytics Platforms:**

- **Kaiko**: Provides comprehensive market depth data across CEXs and DEXs through unified API, enabling direct comparison of liquidity metrics [?]
- **Dune Analytics**: Offers SQL-based interface for on-chain analytics, allowing custom queries on DEX activity
- **DefiLlama**: Aggregates total value locked and volume data across DeFi protocols [?]

**Open-Source Tools:**

GitHub repositories such as solidquant's CEX-DEX arbitrage template demonstrate technical feasibility of real-time data streaming from multiple exchanges and orderbook aggregation techniques [?]. These tools provide baseline implementations for WebSocket management and multi-venue data collection.

## 3.6 Gap in Current Research

While existing research examines price efficiency and arbitrage opportunities, there is limited work providing **real-time, comprehensive comparison** of market microstructure between CEXs and DEXs with:

- Live price index calculation from multiple CEXs with volume weighting
- On-chain DEX monitoring at individual transaction level
- Empirical slippage modeling calibrated to actual trade size distributions
- Integrated dashboard for continuous monitoring and analysis
- Statistical testing of lead-lag relationships across trading conditions

Our project addresses this gap by building an end-to-end system that combines real-time data collection, rigorous statistical analysis, and accessible visualization for comprehensive market dynamics analysis. The system enables researchers and practitioners to move beyond retrospective studies to continuous market monitoring and hypothesis testing.

# Chapter 4

# METHODOLOGY

## 4.1 CEX Price Index Calculation

**Objective**: Derive a robust reference price from multiple CEX sources that represents consensus market price.

**Approach**:

**Exchange Selection**: Query top N exchanges (N=5-10) for each trading pair based on 24-hour volume. Use REST APIs to retrieve volume rankings and filter exchanges with $> 1\%$ market share for the pair.

**Price Collection**: Subscribe to WebSocket ticker streams for each selected exchange, collecting bid/ask prices with timestamps. Maintain real-time orderbook snapshots (top 5 levels) for validation.

**Price Index Formula**: For each pair at time $t$, calculate volume-weighted average:

$$\text{price\_index}(t) = \frac{\sum_{i=1}^{N} \text{mid\_price}_i(t) \times \text{volume}_i}{\sum_{i=1}^{N} \text{volume}_i} \tag{4.1}$$

where:

- $\text{mid\_price}_i(t) = \frac{\text{bid}_i(t)+\text{ask}_i(t)}{2}$
- $\text{volume}_i$ = 24-hour trading volume on exchange $i$
- Sum over all selected exchanges

**Data Validation**:

- Reject prices deviating $> 5\%$ from median across exchanges (outlier detection)
- Require minimum 3 valid exchange prices for price index calculation

- Log anomalies for investigation

**Update Frequency**: Recalculate on every ticker update from any exchange. Typical latency: 100-500ms from exchange timestamp.

## 4.2 DEX Trade Monitoring

**Objective**: Capture all DEX trades at transaction level with accurate execution prices.

**Uniswap V2/V3 Monitoring**:

**Data Source**: Primary real-time RPC connection to Ethereum/BSC nodes, subscribing to Swap events from target pool contracts. The Graph API provides backup for historical data and gap filling.

**Event Parsing**: Extract parameters from Swap events:

- `amount0In`, `amount1In`: Input token amounts
- `amount0Out`, `amount1Out`: Output token amounts
- `sender`: Transaction initiator
- `to`: Recipient address
- Block timestamp, transaction hash

**Execution Price Calculation**: For a swap from token0 to token1:

$$\text{execution\_price} = \frac{\text{amount1Out}}{\text{amount0In}} \tag{4.2}$$

Convert to USD terms using token prices from CEX price index.

**Pool State Tracking**: Maintain reserve balances $(R_0, R_1)$ for each pool. For Uniswap V3, track current tick and liquidity within active price range. Calculate theoretical slippage from pool state.

**Trade Classification**:

- Size bins: <\$1k, \$1k-\$10k, \$10k-\$100k, >\$100k
- Direction: Buy vs Sell relative to quote token
- MEV identification: Check if part of sandwich/arbitrage bundle

## 4.3 Price Deviation Measurement

**Objective**: Quantify deviation between DEX execution prices and CEX price index.

**Per-Trade Deviation**:

$$\text{deviation}_i = \frac{\text{DEX\_price}_i - \text{CEX\_price\_index}_i}{\text{CEX\_price\_index}_i} \times 100\% \tag{4.3}$$

where DEX_price$_i$ is the execution price of trade $i$ on DEX, and CEX_price_index$_i$ is the price index at trade $i$ timestamp (±1 second window).

**Aggregate Metrics**:

**Mean Absolute Deviation (MAD)**:

$$\text{MAD} = \frac{1}{N}\sum_{i=1}^{N} |\text{deviation}_i| \tag{4.4}$$

**Standard Deviation**:

$$\sigma_{\text{deviation}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\text{deviation}_i - \mu)^2} \tag{4.5}$$

**Percentile Analysis**: Compute P50 (median), P90, P95, P99 deviations. Separate analysis for buy/sell directions.

**Time-Series Analysis**: Calculate rolling window statistics (1min, 5min, 15min intervals) to identify periods of persistent deviation.

## 4.4 Lead-Lag Correlation Analysis

**Objective**: Determine temporal relationship between CEX and DEX price movements.

**Method**: Cross-Correlation Function (CCF)

**Price Return Series**:

$$\text{CEX\_return}(t) = \log(\text{CEX\_price\_index}(t)) - \log(\text{CEX\_price\_index}(t-1)) \tag{4.6}$$

$$\text{DEX\_return}(t) = \log(\text{DEX\_price}(t)) - \log(\text{DEX\_price}(t-1)) \tag{4.7}$$

**Cross-Correlation**:

$$\text{CCF}(\tau) = \text{Corr}(\text{CEX\_return}(t), \text{DEX\_return}(t+\tau)) \tag{4.8}$$

for lag $\tau \in [-60s, +60s]$ with 1-second intervals.

**Interpretation**:

- If max CCF at $\tau < 0$: CEX leads DEX by $|\tau|$ seconds
- If max CCF at $\tau > 0$: DEX leads CEX by $\tau$ seconds
- If max CCF at $\tau = 0$: Synchronous movement

**Statistical Testing**: Granger causality test to determine if CEX price changes help predict DEX price changes [?]. Null hypothesis: CEX price changes do not Granger-cause DEX price changes. Significance level: $\alpha = 0.05$.

## 4.5 Slippage Modeling

**Objective**: Predict slippage for arbitrary trade sizes based on pool state and empirical distributions.

**Two-Component Model**:

**Component 1: Theoretical AMM Slippage**

For Uniswap V2 constant product AMM with reserves $(x, y)$ and input amount $\Delta x$:

$$\Delta y = y - \frac{xy}{x + \Delta x} \tag{4.9}$$

$$\text{slippage}_{\text{theoretical}} = \left| 1 - \frac{\Delta y / \Delta x}{y / x} \right| \tag{4.10}$$

For Uniswap V3 concentrated liquidity: Use SDK or on-chain quoter contract to integrate liquidity across active tick ranges.

**Component 2: Empirical Slippage Distribution**

(i) Collect $N$ trades ($N >$ 10,000) over analysis period

(ii) Calculate actual slippage for each trade

(iii) Group by trade size bins

(iv) Build probability distribution: $P(\text{slippage} \mid \text{trade\_size})$

**Slippage Prediction**: For a hypothetical trade of size $S$:

$$\text{predicted\_slippage} = \alpha \times \text{slippage}_{\text{theoretical}}(S) + \beta \times \text{slippage}_{\text{empirical}}(S) \tag{4.11}$$

where $\alpha, \beta$ are weights fit via regression on historical data.

## 4.6 Predictive Models for Liquidity Rebalancing

**Objective**: Utilize empirical trade distributions to predict the probability of price movements across Uniswap V3 concentrated liquidity ranges, enabling automated liquidity rebalancing strategies.

**Approach**:

**Empirical Trade Distribution Analysis**:

Collect historical DEX trades and build probability distributions for trade characteristics:

(i) **Trade Size Distribution**: For each pool, compute the empirical distribution of trade sizes:

$$P(\text{trade\_size} = s) = \frac{\text{count(trades with size } s)}{N_{\text{total}}} \tag{4.12}$$

(ii) **Inter-Trade Time Distribution**: Model the time between consecutive trades:

$$P(\Delta t) = \text{empirical distribution of time gaps} \tag{4.13}$$

(iii) **Price Impact Distribution**: For each trade size bin, measure the distribution of price movements:

$$P(\Delta p \mid \text{trade\_size} = s) = \text{empirical distribution of price changes} \tag{4.14}$$

**Probability of Range Transition**:

For Uniswap V3 pools with concentrated liquidity positions, calculate the probability that the next trade will move the price from the current tick range to a different liquidity concentration level:

$$P(\text{tick}_{\text{new}} \in [\text{tick}_a, \text{tick}_b] \mid \text{tick}_{\text{current}}) = \sum_s P(\text{trade\_size} = s) \times P(\Delta p(s) \text{ crosses boundary}) \tag{4.15}$$

where the boundary crossing probability is determined by:

- Current pool reserves and liquidity distribution
- Required trade size to move price by $\Delta p$
- Historical frequency of such trades

**Slippage Prediction with Price Index Integration**:

Enhance the slippage prediction model by incorporating the CEX price index as an external signal:

$$\text{predicted\_slippage}(s,t) = f(\text{pool\_state}(t), \text{trade\_size} = s, \text{CEX\_DEX\_deviation}(t)) \tag{4.16}$$

where:

- pool_state($t$): Current reserves and liquidity distribution
- CEX_DEX_deviation($t$) $= \frac{\text{DEX\_price}(t) - \text{CEX\_price\_index}(t)}{\text{CEX\_price\_index}(t)}$
- When $|\text{CEX\_DEX\_deviation}| > \theta$ (threshold), expect increased arbitrage activity

The model can predict higher slippage when:

(i) DEX price deviates significantly from CEX price index (arbitrage opportunity)

(ii) Recent trade velocity is high (estimated from inter-trade time distribution)

(iii) Current price is near concentrated liquidity boundaries

**Automated Liquidity Rebalancing Logic**:

Liquidity providers can use these predictions to optimize position management:

**Algorithm 1** Automated Liquidity Rebalancing

```
Input: Current position [$\text{tick}_{\text{low}}, \text{tick}_{\text{high}}$], threshold $p_{\text{threshold}}$
Compute $P$(price exits range within $\Delta t$) using empirical distributions
if $P(\text{exit}) > p_{\text{threshold}}$ then
    Identify new optimal range [$\text{tick}_{\text{new\_low}}, \text{tick}_{\text{new\_high}}$] based on:
        - Predicted price target from CEX-DEX deviation
        - Trade size distribution (concentration around new range)
        - Expected fee revenue vs. gas costs
    Execute rebalancing transaction
end if
Update position tracking
```

**Risk-Adjusted Positioning**:

The model enables liquidity providers to make informed decisions:

- **Wide Range Strategy**: When trade size distribution shows high variance and frequent large trades, use wider ranges to avoid frequent rebalancing
- **Narrow Range Strategy**: When trades are small and predictable, concentrate liquidity tightly around current price for higher fee capture
- **CEX-Aligned Strategy**: Monitor CEX price index deviation and proactively reposition before arbitrageurs force price convergence

**Performance Metrics**:
Evaluate rebalancing strategy performance:

$$\text{Sharpe Ratio}_{\text{LP}} = \frac{\text{Fee Revenue} - \text{Gas Costs} - \text{Impermanent Loss}}{\sigma(\text{Returns})} \tag{4.17}$$

Compare passive (no rebalancing) vs. active (predicted rebalancing) strategies over the analysis period.

**Implementation Considerations**:

- **Gas Cost Modeling**: Factor in Ethereum gas prices when deciding rebalancing frequency
- **Slippage on Rebalancing**: Account for execution costs of position adjustments
- **Market Impact**: Consider that rebalancing transactions themselves affect pool state
- **Real-time Updates**: Model parameters should be updated continuously as new trades occur

## 4.7 Liquidity Depth Comparison

**CEX Orderbook Depth**:

$$\text{depth}_{\text{CEX}}(\Delta p) = \sum_{\text{prices within } \Delta p} \text{volume at price level} \tag{4.18}$$

Example: depth within ±0.5% = sum of bid/ask volumes between current_price ×0.995 and current_price ×1.005.

**DEX Pool Depth**:

For V2 pools: Calculate the amount that can be traded to achieve $X\%$ price impact using the constant product formula.

For V3 pools:

$$\text{depth}_{\text{DEX}}(\Delta p) = \sum_{\text{ticks within } \Delta p} L_{\text{tick}} \times \Delta p_{\text{tick}} \tag{4.19}$$

where $L_{\text{tick}}$ is liquidity in each tick and $\Delta p_{\text{tick}}$ is the price difference per tick.

**Comparison Metric**:

$$\text{liquidity\_ratio}(\pm X\%) = \frac{\text{depth}_{\text{DEX}}(\pm X\%)}{\sum_i \text{depth}_{\text{CEX}_i}(\pm X\%)} \tag{4.20}$$

## 4.8 Statistical Validation

**Backtesting Framework**:

(i) **Historical Simulation**: Use collected historical data to simulate price index calculation as it would have occurred in real-time. Compare predicted vs actual deviations.

(ii) **Cross-Validation**: Train slippage models on 70% of data, test on held-out 30%. Report $R^2$, MAE (Mean Absolute Error), RMSE (Root Mean Square Error).

(iii) **Robustness Checks**:

- Analyze performance across different market conditions (high/low volatility)
- Test sensitivity to exchange selection for price index
- Evaluate impact of missing data

# Chapter 5

# REQUIREMENTS SPECIFICATION

## 5.1 Functional Requirements

### 5.1.1 FR1: CEX Data Collection

- **FR1.1**: System shall connect to minimum 5 CEX WebSocket APIs simultaneously
- **FR1.2**: System shall collect bid/ask prices with $< 1$ second latency
- **FR1.3**: System shall sync 24h volume data via REST APIs
- **FR1.4**: System shall handle WebSocket disconnections and reconnect automatically

### 5.1.2 FR2: Price Index Calculation

- **FR2.1**: System shall calculate volume-weighted price index from selected CEX'es
- **FR2.2**: Price index shall update within 1ms of receiving new ticker data
- **FR2.3**: System shall require minimum 3 valid exchanges for price index

### 5.1.3 FR3: DEX Monitoring

- **FR3.1**: System shall monitor Uniswap V2/V3 swap events on Ethereum and BSC

- **FR3.2**: System shall parse swap events and extract trade parameters
- **FR3.3**: System shall calculate execution price for each trade in USD terms
- **FR3.4**: System shall classify trades by size, direction, and type
- **FR3.5**: System shall maintain pool state (reserves, liquidity) in real-time

### 5.1.4 FR4: Price Deviation Analysis

- **FR4.1**: System shall calculate per-trade deviation between DEX and CEX prices
- **FR4.2**: System shall compute aggregate statistics (mean, std, percentiles)
- **FR4.3**: System shall generate time-series data with configurable windows
- **FR4.4**: System shall alert when deviation exceeds configurable threshold

### 5.1.5 FR5: Lead-Lag Analysis

- **FR5.1**: System shall compute cross-correlation function for price returns
- **FR5.2**: System shall identify lag time of maximum correlation
- **FR5.3**: System shall perform Granger causality tests
- **FR5.4**: System shall report statistical significance (p-values)

### 5.1.6 FR6: Slippage Analysis

- **FR6.1**: System shall calculate theoretical slippage from pool state
- **FR6.2**: System shall build empirical slippage distributions from historical trades
- **FR6.3**: System shall predict slippage for user-specified trade sizes
- **FR6.4**: System shall compare actual vs predicted slippage for validation

### 5.1.7 FR7: Liquidity Analysis

- **FR7.1**: System shall measure orderbook depth on CEXs at multiple price levels
- **FR7.2**: System shall calculate available liquidity in DEX pools
- **FR7.3**: System shall compute liquidity ratios between CEX and DEX
- **FR7.4**: System shall track liquidity changes over time

### 5.1.8 FR8: Data Storage

- **FR8.1**: System shall store OHLC data at 24h intervals
- **FR8.2**: System shall store individual DEX trades with pool size, amounts, and timestamp
- **FR8.3**: System shall store price indexs as it changes 2bps or more as index and timestamp
- **FR8.4**: System shall support queries for historical data analysis

### 5.1.9 FR9: Dashboard Visualization

- **FR9.1**: Dashboard shall display real-time price deviation charts
- **FR9.2**: Dashboard shall show volume comparison between CEX and DEX
- **FR9.3**: Dashboard shall visualize liquidity depth for selected pairs
- **FR9.4**: Dashboard shall display slippage curves for different trade sizes
- **FR9.5**: Dashboard shall show lead-lag correlation results
- **FR9.6**: Dashboard shall allow selection of trading pairs and time ranges

## 5.2 Non-Functional Requirements

### 5.2.1 NFR1: Performance

- **NFR1.1**: System shall process 100+ trades per second
- **NFR1.2**: Dashboard shall update visualizations within 500 ms of new data

### 5.2.2 NFR2: Scalability

- **NFR3.1**: System shall support addition of new trading pairs without code changes
- **NFR3.2**: System shall support addition of new CEXs via configuration
- **NFR3.3**: System shall support addition of new blockchain networks

## 5.3 Use Cases

### 5.3.1 UC1: Monitor Real-Time Price Deviation

**Primary Actor**: Research Analyst

**Preconditions**: System is running and collecting data

**Main Flow**:

(i) Analyst opens dashboard

(ii) Analyst selects trading pair (e.g., ETH/USDT)

(iii) System displays real-time price deviation chart

(iv) System shows current CEX price index and latest DEX trade price

(v) System displays deviation statistics (current, 5min avg, 1hr avg)

(vi) Analyst observes deviation patterns

**Postconditions**: Analyst understands current market state

### 5.3.2 UC2: Analyze Historical Lead-Lag Relationship

**Primary Actor**: Research Analyst

**Preconditions**: Minimum 7 days of historical data collected

**Main Flow**:

(i) Analyst selects "Lead-Lag Analysis" module

(ii) Analyst specifies trading pair and date range

(iii) System computes cross-correlation function

(iv) System displays CCF plot with lag times on x-axis

(v) System highlights maximum correlation and corresponding lag

(vi) System displays Granger causality test results

(vii) Analyst interprets whether CEX leads or lags DEX

**Postconditions**: Analyst has quantitative measure of price discovery dynamics

### 5.3.3 UC3: Estimate Slippage for Planned Trade

**Primary Actor**: DEX Trader / Bot Designer

**Preconditions**: Slippage model trained on historical data

**Main Flow**:

(i) User enters trading pair and trade size

(ii) User specifies pool (e.g., Uniswap V3 ETH/USDT 0.05%)

(iii) System retrieves current pool state

(iv) System calculates theoretical slippage from AMM formula

(v) System retrieves empirical slippage distribution for similar trade sizes

(vi) System displays predicted slippage range (P50, P90, P99)

(vii) User decides whether to execute trade

**Postconditions**: User has informed estimate of execution cost

### 5.3.4 UC4: Compare CEX vs DEX Liquidity

**Primary Actor**: Research Analyst / Liquidity Provider

**Preconditions**: System collecting orderbook and pool data

**Main Flow**:

(i) Analyst selects "Liquidity Comparison" module

(ii) Analyst specifies trading pair

(iii) System displays combined CEX orderbook depth chart

(iv) System displays DEX pool liquidity distribution

(v) System calculates and displays liquidity ratios at ±0.5%, ±1%, ±2%

(vi) System shows historical liquidity trends

(vii) Analyst compares capital efficiency between venues

**Postconditions**: Analyst understands relative liquidity availability

# Chapter 6

# DESIGN

## 6.1 Information Structure

### 6.1.1 Entity-Relationship Model

**Core Entities**:

**Exchange**: Stores information about trading venues

- `exchange_id` (PK): Unique identifier
- `exchange_name`: Binance, Coinbase, etc.
- `exchange_type`: CEX or DEX
- `api_endpoint`: WebSocket/REST URL
- `status`: Active/Inactive

**TradingPair**: Represents tradable asset pairs

- `pair_id` (PK): Unique identifier
- `base_token`: e.g., ETH
- `quote_token`: e.g., USDT
- `pair_symbol`: ETH/USDT
- `active`: Boolean

**CEXTicker**: Real-time price data from centralized exchanges

- `ticker_id` (PK): Unique identifier

- `exchange_id` (FK): Reference to Exchange
- `pair_id` (FK): Reference to TradingPair
- `timestamp`: UTC millisecond precision
- `bid_price`, `ask_price`: Best bid/ask
- `volume_24h`: 24-hour volume

**PriceIndex**: Calculated reference price

- `price_index_id` (PK): Unique identifier
- `pair_id` (FK): Reference to TradingPair
- `timestamp`: UTC millisecond precision
- `price`: Calculated price index
- `num_exchanges`: Number of exchanges used
- `exchanges_used`: Array of exchange_ids

**DEXPool**: Liquidity pool information

- `pool_id` (PK): Pool contract address
- `pair_id` (FK): Reference to TradingPair
- `dex_protocol`: Uniswap-V2, Uniswap-V3, etc.
- `blockchain`: Ethereum, BSC, etc.
- `fee_tier`: 0.05%, 0.30%, 1.00%
- `reserve0`, `reserve1`: Token reserves

**DEXTrade**: Individual DEX transactions

- `trade_id` (PK): Transaction hash + log index
- `pool_id` (FK): Reference to DEXPool
- `timestamp`: Block timestamp
- `block_number`: Ethereum block
- `sender`: Address initiating swap

- `amount0_in`, `amount1_in`: Input amounts
- `amount0_out`, `amount1_out`: Output amounts
- `execution_price`: Calculated price
- `trade_size_usd`: USD value
- `trade_direction`: Buy/Sell

**PriceDeviation**: Deviation measurements

- `deviation_id` (PK): Unique identifier
- `trade_id` (FK): Reference to DEXTrade
- `price_index_id` (FK): Reference to PriceIndex
- `deviation_pct`: Percentage deviation
- `absolute_deviation`: Absolute price difference

**SlippageModel**: Empirical slippage statistics

- `model_id` (PK): Unique identifier
- `pool_id` (FK): Reference to DEXPool
- `trade_size_bin`: <1k, 1k-10k, etc.
- `mean_slippage`, `std_slippage`: Statistics
- `p50`, `p90`, `p99`: Percentile values
- `sample_size`: Number of trades

**Relationships**:

- Exchange (1) → (*) CEXTicker
- TradingPair (1) → (*) CEXTicker, PriceIndex, DEXPool
- DEXPool (1) → (*) DEXTrade, SlippageModel
- DEXTrade (1) → (1) PriceDeviation

## 6.2 Information Flow

### 6.2.1 Activity Diagram: Real-Time Price Deviation Monitoring

The system continuously monitors both CEX and DEX venues in parallel:

**CEX Stream**:

(i) CEX Collectors receive ticker updates via WebSocket

(ii) Query 24h volumes every 5 minutes via REST API

(iii) Update PriceIndex calculation using volume-weighted average

(iv) Store PriceIndex in database with timestamp

**DEX Stream** (concurrent):

(i) DEX Listener receives Swap event from blockchain

(ii) Parse trade parameters (amounts, addresses, timestamp)

(iii) Calculate execution price from swap amounts

(iv) Query PriceIndex at trade timestamp (±1s window)

(v) Calculate deviation percentage

(vi) Store DEXTrade and PriceDeviation records

(vii) Update real-time statistics

(viii) Push update to Dashboard for visualization

### 6.2.2 Sequence Diagram: Price Index Calculation

The price index calculation involves coordination between multiple components:

(i) Dashboard requests price index for ETH/USDT

(ii) PriceIndexService queries VolumeUpdater for 24h volume data

(iii) VolumeUpdater returns volume ranking across exchanges

(iv) PriceIndexService queries CEXCollector for latest tickers

(v) Database returns recent ticker data from all exchanges

(vi) PriceIndexService filters outliers ($> 5\sigma$ from median)

(vii) Calculate volume-weighted average price (VWAP)

(viii) Store calculated price index in database

(ix) Return price index to Dashboard for display

### 6.2.3 Sequence Diagram: DEX Trade Processing

Processing a DEX trade involves multiple stages:

(i) Blockchain emits Swap event

(ii) DEXListener captures and parses event

(iii) Query PoolStateTracker for current pool reserves

(iv) Calculate execution price from swap amounts

(v) Query Database for price index at trade timestamp

(vi) DeviationCalculator computes price deviation

(vii) Store trade record in Database

(viii) Store deviation record in Database

(ix) Trigger analytics update

## 6.3 System Design

### 6.3.1 High-Level Architecture

The system follows a layered architecture with four main tiers:

**Presentation Layer**:

- Streamlit Dashboard
- Price Deviation Charts
- Volume Comparison Visualizations
- Liquidity Analysis Interface

- Lead-Lag Correlation Plots
- Slippage Prediction Tool

**Application Layer**:

- **Analysis Engine**: Statistical computations and aggregations
- **Price Index Service**: VWAP calculation from multiple CEXs
- **Deviation Calculator**: Per-trade deviation measurement
- **Lead-Lag Analyzer**: Cross-correlation and Granger causality
- **Slippage Model**: Empirical and theoretical predictions
- **Liquidity Analyzer**: Depth comparison across venues

**Data Collection Layer**:
CEX Data Collector:

- WebSocket connections to 8+ exchanges (Binance, Coinbase, Bybit, OKX, etc.)
- REST API for volume data
- Ticker handler and price aggregator

DEX Data Collector:

- Ethereum and BSC RPC connections
- Event parser for Swap events
- Pool state tracker for reserves
- The Graph API client for historical data

**Data Storage Layer**:

- **TimeSeries DB**: OHLC data, tick data (InfluxDB or similar)
- **File Cache**: Raw data buffering
- **PostgreSQL**: Metadata, configuration, aggregated statistics

### 6.3.2 Component Details

**CEX Data Collector**:

- **Technology**: Python asyncio with websockets library
- **Components**:
  - ExchangeConnector: Base class for WebSocket connections
  - TickerHandler: Processes incoming ticker messages
  - VolumeUpdater: Periodic REST API calls for volume
  - PriceIndexCalculator: Computes VWAP
- **Threading**: One async task per exchange connection
- **Error Handling**: Exponential backoff for reconnections

**DEX Data Collector**:

- **Technology**: Python web3.py for RPC, async event filtering
- **Components**:
  - BlockchainListener: Subscribes to new blocks
  - EventParser: Decodes Swap events from logs
  - PoolStateManager: Maintains reserve state
  - TheGraphClient: Historical data queries
- **Optimization**: Batch RPC requests, cache static pool data

**Analysis Engine**:

- **Technology**: Python NumPy/Pandas for numerical computation
- **Components**:
  - DeviationAnalyzer: Per-trade and aggregate deviations
  - LeadLagCalculator: Cross-correlation and Granger tests
  - SlippagePredictor: Empirical + theoretical fusion
  - LiquidityMeasurer: Orderbook and pool depth
- **Scheduling**: Runs on configurable intervals (e.g., every 5 minutes)

**Dashboard**:

- **Technology**: Streamlit for rapid development
- **Components**:
  - RealTimeCharts: Plotly interactive visualizations
  - DataFetcher: Database queries for historical data
  - WebSocketClient: Receives live updates
  - ControlPanel: User inputs for selections

### 6.3.3 Data Flow Patterns

**Stream Processing**:
CEX Ticker → [Validation] → [Price Index Calc] → [Cache + DB]
DEX Trade → [Parse] → [Price Calc] → [Match Price Index] → [Deviation] → [DB] → [Analytics Queue] → [Dashboard]
**Batch Processing** (for historical analysis):
[Load Historical Data] → [Compute Statistics] → [Generate Report] → [Store Results]

## 6.4 User Interface Design

### 6.4.1 Dashboard Layout

**Header Section**:

- Trading pair selector (dropdown)
- Time range selector (1h, 6h, 24h, 7d, 30d, custom)
- Network selector (Ethereum, BSC)
- Status indicators (CEX connections, DEX listener, last update)

**Main Content Area** (Tabbed Interface):
**Tab 1: Real-Time Monitoring**

- Top row: Current price index, latest DEX trade, current deviation
- Middle: Price deviation time series chart
- Bottom: Volume comparison bar chart (CEX vs DEX)

**Tab 2: Lead-Lag Analysis**

- Cross-correlation function plot
- Summary statistics (optimal lag, correlation, p-value)
- Interpretation text

**Tab 3: Liquidity Analysis**

- Side-by-side charts: CEX depth vs DEX liquidity
- Liquidity ratio table at different price levels
- Historical liquidity trend

**Tab 4: Slippage Calculator**

- Input fields: Trade size, direction, pool selection
- Output: Predicted slippage (P50/P90/P99)
- Slippage curve visualization

**Tab 5: Volume Analysis**

- Daily volume time series (CEX vs DEX)
- Trade size distribution histograms
- Market share pie chart

### 6.4.2 Interaction Patterns

- **Hovering**: Tooltips show exact values
- **Clicking**: Selects time point, shows related data
- **Dragging**: Zooms into time range
- **Export**: Download data as CSV or charts as PNG
- **Auto-refresh**: Toggle for real-time updates (default: ON, 5s interval)

### 6.4.3 Visual Design Principles

- **Color scheme**: CEX data in blue, DEX in green, deviations in red/orange
- **Typography**: Monospace for numbers, sans-serif for text
- **Responsive layout**: Adapts to different screen sizes
- **Minimal clutter**: Hide advanced options behind expandable sections

# Chapter 7

# IMPLEMENTATION AND TESTING

## 7.1 Implementation

The implementation phase is currently in progress. The following components have been completed:

### 7.1.1 Completed Components

**CEX Data Collection Infrastructure**:

- WebSocket connections to 8+ major centralized exchanges
- Real-time ticker data streaming
- Automatic reconnection handling
- Data validation and outlier detection

**DEX Transaction Monitoring**:

- RPC connections to Ethereum and BSC networks
- Swap event listeners for Uniswap V2 and V3
- Transaction parsing and data extraction
- Pool state tracking

**Data Storage**:

- File-based caching system for raw data

- Database schema implementation
- Data persistence layer

### 7.1.2 Components in Development

- Price index calculation engine
- Price deviation measurement system
- Statistical analysis modules
- Dashboard interface

## 7.2 Testing

Testing strategy encompasses multiple levels:

### 7.2.1 Unit Testing

- Test individual components in isolation
- Validate data parsing logic
- Verify mathematical calculations
- Test edge cases and error conditions

### 7.2.2 Integration Testing

- Test component interactions
- Validate data flow between modules
- Test database operations
- Verify WebSocket and RPC connections

### 7.2.3 System Testing

- End-to-end workflow validation
- Performance testing under load
- Stress testing with high-frequency data
- Failover and recovery testing

### 7.2.4 Manual QA

- Validation against known market events
- Cross-reference with established platforms
- Leverage team's HFT experience for anomaly detection
- Visual inspection of results

## 7.3 Deployment

Deployment architecture is planned as follows:

### 7.3.1 Infrastructure Requirements

- Cloud compute instances for data collection
- Database server for persistent storage
- Web server for dashboard hosting
- RPC node access or local blockchain nodes

### 7.3.2 Deployment Configuration

- Containerization using Docker (if needed)
- Configuration management for exchange endpoints
- Environment-specific settings
- Monitoring and logging setup

### 7.3.3 Documentation

- System architecture documentation
- API documentation
- Configuration guide
- User manual for dashboard
- Troubleshooting guide

# Chapter 8

# RESULTS

Results will be presented upon completion of the implementation and data collection phases. The analysis will include:

### 8.0.1 Price Discovery Analysis

- Lead-lag correlation results between CEX and DEX
- Statistical significance of temporal relationships
- Variation across different market conditions
- Comparison across multiple trading pairs

### 8.0.2 Volume Comparison

- CEX vs DEX volume ratios
- Volume distribution by trade size
- Temporal trends in market share
- Pair-specific analysis

### 8.0.3 Liquidity Analysis

- Orderbook depth comparison
- Pool liquidity measurements
- Capital efficiency metrics
- Liquidity concentration analysis

### 8.0.4 Slippage Characterization

- Empirical slippage distributions
- Model validation results
- Comparison with theoretical predictions
- Trade size impact analysis

### 8.0.5 Price Deviation Measurements

- Statistical summary of deviations
- Temporal patterns
- Market condition dependencies
- Outlier analysis

### 8.0.6 Predictive Modeling and Liquidity Rebalancing

- Empirical probability distributions for trade sizes and price impacts
- Range transition probabilities for Uniswap V3 concentrated liquidity positions
- Accuracy of slippage predictions incorporating CEX price index deviations
- Performance comparison: passive vs. active rebalancing strategies
- Risk-adjusted returns (Sharpe ratios) for different liquidity positioning strategies
- Optimal rebalancing thresholds considering gas costs and expected fee revenue
- Case studies of rebalancing decisions during high CEX-DEX deviation periods

# Chapter 9

# CONCLUSION

This midterm report presents the design and planning for a comprehensive analysis of cryptocurrency exchange market dynamics, comparing centralized and decentralized venues.

### 9.0.1 Summary of Work Completed

We have successfully:

- Defined clear research objectives focusing on price discovery, liquidity, volume, and execution costs
- Developed detailed methodology for price index calculation, deviation measurement, and lead-lag analysis
- Specified comprehensive functional and non-functional requirements
- Designed system architecture with clear separation of concerns
- Implemented core data collection infrastructure for both CEX and DEX monitoring

### 9.0.2 Key Contributions

Our project addresses a gap in current DeFi research by providing:

- Real-time comparative analysis framework
- Integrated price index calculation from multiple CEX sources
- Empirical slippage modeling based on actual trade distributions
- Comprehensive monitoring and visualization system

### 9.0.3 Next Steps

For the remainder of the semester, we will:

(i) Complete price index calculation engine implementation

(ii) Implement statistical analysis modules (lead-lag, deviation measurement)

(iii) Develop slippage prediction model

(iv) Build and deploy dashboard interface

(v) Collect sufficient data for statistically significant analysis

(vi) Conduct comprehensive testing and validation

(vii) Generate final results and analysis

### 9.0.4 Expected Outcomes

Upon project completion, we expect to:

- Quantify the extent to which DEX prices lag CEX prices
- Characterize liquidity differences between exchange types
- Provide empirical slippage predictions for DEX trades
- Develop predictive models for price movement probabilities across Uniswap V3 concentrated liquidity ranges
- Demonstrate automated liquidity rebalancing strategies informed by CEX price index deviations and empirical trade distributions
- Deliver a functional monitoring system for continued research

This work contributes to better understanding of DeFi market microstructure and provides practical tools for traders, researchers, and protocol developers working in the cryptocurrency space.

# Appendix A

# SAMPLE APPENDIX

Contents of the appendix.