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

January 2026

Contents

Chapter 1

INTRODUCTION

1.1 Background and Motivation

Centralized exchanges (CEXs) and decentralized exchanges (DEXs) represent two fundamentally different market structures for cryptocurrency trading. CEXs such as Binance, Coinbase and OKX rely on high-throughput off-chain matching engines and orderbook-based price discovery, offering low-latency execution and deep liquidity. In contrast, DEXs such as Uniswap execute trades on-chain through automated market makers (AMMs), enabling self-custody and transparency at the cost of higher latency and execution frictions driven by pool depth, blockchain congestion and Maximal Extractable Value (MEV).

These structural differences create measurable gaps in execution quality and information flow between venues. For liquid pairs, CEXs often dominate price discovery due to their substantially larger trading volume, while DEX prices typically track and converge through arbitrage. For less liquid regimes or during volatile periods, deviations can persist longer, slippage grows non-linearly with trade size, and on-chain transaction ordering can introduce additional costs beyond theoretical AMM price impact.

This project builds an end-to-end monitoring and analysis system to quantify these dynamics using real-time data streams from both CEX and DEX venues.

1.1.1 CEX Orderbook Mechanics

Most major CEX spot markets use a central limit order book (CLOB). Participants submit *limit orders* (resting bids/asks at specified prices) and *market orders* (immediate execution against the best available opposite-side liquidity). The matching engine typically enforces **price–time priority**: better prices match first; for equal prices, earlier orders match first. The resulting best bid/ask defines the spread, and the mid-price (average of best bid and best ask) provides a simple reference for fair price estimation.

Orderbook **market depth** (available quantity at/near the current price) largely determines execution quality: deeper books and tighter spreads yield lower slippage for a given trade size. In practice, this depth is sustained by a combination of professional market makers and broad participation, supported by low-latency infrastructure and high-throughput market data distribution [3, 9].

1.1.2 MEV and Transaction Ordering Effects

DEX trades are executed via blockchain transactions, which introduces an additional layer of execution cost not present in off-chain CEX matching: **transaction ordering**. On chains with a public mempool, pending swaps can be observed before inclusion in a block. Specialized actors (MEV "searchers") may reorder, insert, or bundle transactions to extract profit, for example through front-running/back-running and sandwich attacks [14].

For AMM-based DEXs, this matters because the swap price is a deterministic function of pool state and trade size. If an attacker trades just before a victim swap (moving the pool price) and then trades back after, the victim experiences worse execution than predicted by the AMM price impact alone. Empirically, this appears as additional slippage and heavier tails in execution-cost distributions, especially around volatile periods or large trades. In our system, we treat these effects as part of the observed execution outcome and discuss how they can widen CEX–DEX deviations and distort measured slippage beyond theoretical curves.

1.1.3 On-Chain Orderbooks: Hyperliquid

While many DEXs rely on AMMs, some modern systems implement **on-chain orderbooks** to approximate CEX-style trading while retaining on-chain settlement. Hyperliquid is a prominent example: it provides rapid block times (sub-second) and integrates order matching into the chain's execution/consensus pipeline, enabling limit-order trading with low user friction [10, 15].

This design changes the microstructure of decentralized trading. Compared with AMMs, execution quality depends more directly on orderbook depth rather than pool liquidity, and the system can reduce some forms of mempool-based MEV by altering transaction exposure and matching semantics. We include Hyperliquid as context for the broader design space of DEX market structures and to motivate why comparing “DEX” venues requires distinguishing AMM-based exchanges from orderbook-based decentralized trading systems.

1.2 Problem Statement and Objectives

The primary objective is to measure and explain cross-venue market dynamics between CEX and DEX markets for major cryptocurrency pairs. Specifically, we aim to:

- compute a robust CEX reference price using a multi-exchange price index,
- capture DEX swap executions on Uniswap V2 and Uniswap V3 across Ethereum and BSC,
- quantify price deviation between DEX execution prices and the CEX reference price,
- measure lead–lag relationships to study price discovery and information flow,
- characterize slippage as a function of trade size using both theoretical and empirical approaches.

Our analysis focuses on the pairs BTC/USDT, ETH/USDT and BNB/USDT, using CEX feeds from Coinbase, MEXC, Binance, Gate.io, OKX, Bybit, HTX and KuCoin.

1.3 Contributions

This work makes the following contributions:

- **Real-time multi-venue data collection:** A pipeline that streams ticker data from multiple CEX WebSocket APIs and swap events from Uniswap V2/V3 pools on Ethereum and BSC.
- **CEX price index:** A volume-weighted reference price constructed from multiple exchanges to reduce single-venue noise and anomalies.
- **Deviation and lead–lag analysis:** Metrics and analysis modules to quantify CEX–DEX deviation distributions and temporal relationships between venues.

- **Slippage characterization:** A framework combining theoretical AMM price impact with empirical slippage distributions derived from observed trades.
- **Storage and reporting:** A database-backed design supporting both real-time monitoring and retrospective statistical analysis.

1.4 Broad Impact

This project provides a structured empirical framework for comparing orderbook-based and AMM-based markets using consistent metrics. The results are relevant to researchers studying market microstructure, practitioners designing execution strategies, and liquidity providers evaluating the trade-offs between fee revenue and adverse execution costs. By quantifying deviation dynamics and lead–lag effects, the system supports evidence-based reasoning about where price discovery occurs and how quickly on-chain markets converge to off-chain reference prices.

1.5 Ethical Considerations

Measuring CEX–DEX deviations may highlight temporary inefficiencies that can be exploited via arbitrage. We treat this as ethically acceptable within the context of research transparency: such inefficiencies exist in public markets and are discoverable by capable participants. Moreover, arbitrage activity generally improves price alignment across venues, benefiting market participants through tighter effective spreads and better execution quality over time. We avoid providing prescriptive guidance for exploitation and instead focus on documenting aggregate dynamics and structural causes, including the role of MEV and transaction ordering on on-chain execution costs.

Chapter 2

PROJECT DEFINITION AND PLANNING

2.1 Project Definition

Research objectives. The project conducts a comparative analysis of cryptocurrency market microstructure across centralized and decentralized venues. Our core research dimensions are:

- **Price discovery dynamics:** quantify lead–lag relationships between CEX and DEX price movements via correlation analysis.
- **Volume distribution:** compare trading activity across venues for the monitored pairs.
- **Liquidity structure:** compare CEX orderbook depth to DEX pool liquidity and concentrated liquidity structure.
- **Execution costs:** model and measure slippage as a function of trade size for Uniswap pools.
- **Infrastructure development:** build a real-time monitoring pipeline enabling continuous data collection and retrospective analysis.

Technical scope. The final system monitors CEX tickers from Coinbase, MEXC, Binance, Gate.io, OKX, Bybit, HTX and KuCoin. On the DEX side, we monitor Uniswap V2 and Uniswap V3 swap events. The DEX collector is designed as a multi-chain listener and is configurable to support additional EVM networks (e.g., Polygon, Arbitrum, Optimism, Base, Avalanche, Fantom, Cronos, zkSync, Polygon zkEVM, Celo) via `dex-prices/config.json`; however, our empirical analysis focuses on Ethereum and BSC. We analyze BTC/USDT, ETH/USDT and BNB/USDT.

2.2 Project Planning

2.2.1 Project Time and Resource Estimation

Phase	Weeks	Deliverables	Status
Infrastructure Setup	1–3	CEX WebSocket consumers, DEX listeners	Complete
Data Pipeline	4–6	Price index, storage, stream processing	Complete
Analysis Implementation	7–9	Deviation, correlation, slippage modules	Complete
Visualization	10–11	Dashboard (Streamlit)	Complete
Documentation	12	Final report and system docs	Complete

Table 2.1: Project timeline and deliverables

2.2.2 Success Criteria

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
Sample Size	Trades per pair	$\geq$ 10,000 DEX trades

Table 2.2: Project success criteria and targets

2.2.3 Risk Analysis

Risk	**Impact**	**Likelihood**	**Mitigation Strategy**
RPC rate limits	High	Medium	Multiple providers, request batching, caching
CEX API downtime	Medium	Medium	Redundant sources, reconnect logic, validation
Data quality issues	Medium	Medium	Outlier detection, cross-venue checks, logging
Blockchain congestion	Medium	Medium	Backfilling, time-window matching for indices
Technical complexity	Medium	High	Modular design, incremental milestones

Table 2.3: Risk analysis and mitigation strategies

2.2.4 Team Work (if applicable)

Team structure. The three-member team divided responsibilities across three primary areas: CEX infrastructure (WebSocket connections, aggregation and price index), DEX infrastructure (on-chain monitoring, event parsing and pool tracking), and analysis/visualization (statistical modules and dashboard). We collaborated through a shared repository, consistent interfaces between modules, and iterative integration to ensure that collected data was usable for downstream analysis.

Chapter 3

RELATED WORK

3.1 Price Discovery Between CEX and DEX

Prior work has examined whether price discovery occurs primarily on CEX venues or whether DEX venues contribute meaningfully. Alexander et al. study price discovery and efficiency between Uniswap pools and major centralized exchanges and find that DEXs can contribute to price formation under certain conditions, with behavior varying across pool versions and fee tiers [2]. Wu et al. emphasize that cross-venue latency and market fragmentation create recurring short-lived discrepancies that are resolved by arbitrage activity [4, 14].

3.2 DEX Market Structure and Liquidity

DEX market structure differs from CEXs not only in execution mechanism but also in asset coverage and volume concentration. Lehar and Parlour show that DEXs list many more tokens than major CEXs but handle substantially lower volume for established pairs, and that CEX listings often shift volume away from DEX-only trading [12]. Market reports similarly quantify persistent volume and liquidity gaps between CEX and DEX venues and provide practical depth and liquidity reference metrics [11].

3.3 Liquidity and Slippage Analysis

Execution costs on AMM-based DEXs depend on local liquidity around the current price and increase nonlinearly with trade size. Empirica's framework highlights how concentrated liquidity affects effective depth and slippage, and motivates comparing Uniswap V3 liquidity distribution to an orderbook-like structure [7]. Related analyses note that for major pairs, CEX books often exhibit deeper liquidity at common depth thresholds, while aggregating multiple DEX pools and fee tiers can narrow the gap [1, 11].

3.4 MEV and Transaction Costs

Transaction costs on DEXs include both benign price impact and blockchain-specific frictions. Capponi et al. decompose costs on Uniswap V3 and show that adversarial ordering effects can materially affect realized execution, especially for larger swaps [5]. Recent work on CEX–DEX extracted value also highlights increasing centralization in the MEV supply chain (searchers/builders), which can affect who captures cross-venue arbitrage profits and under what conditions [14].

3.5 Existing Tools and Platforms

Commercial platforms (e.g., Kaiko) offer unified access to depth and market data across venues, while on-chain analytics platforms enable custom DEX queries and capital-flow summaries (e.g., DefiLlama) [6, 11]. Open-source tooling demonstrates feasibility of multi-exchange streaming and arbitrage templates, but often targets execution rather than systematic measurement and reporting [13].

3.6 Gap in Current Research

Despite a growing literature on price efficiency and arbitrage, there is limited work that integrates (i) a robust multi-venue CEX reference price, (ii) transaction-level DEX execution monitoring, and (iii) consistent metrics for deviation, lead–lag, and slippage in an end-to-end system suitable for continuous monitoring and reproducible retrospective analysis. Our project targets this gap by combining real-time data collection, database-backed storage, and a modular analysis engine that produces report-ready outputs.

Chapter 4

METHODOLOGY

4.1 CEX Price Index Calculation

Objective: derive a robust reference price from multiple CEX sources that represents the consensus market price.

Approach: for each monitored pair, we maintain real-time best bid/ask prices and top-of-book quantities from multiple exchanges via WebSocket connections. At time t, we compute the mid-price for exchange i as:

$$\text{mid\_price}_i(t) = \frac{\text{bid}_i(t) + \text{ask}_i(t)}{2} \tag{4.1}$$

We then compute a volume-weighted index:

$$\text{price\_index}(t) = \frac{\sum_{i=1}^{N} \text{mid\_price}_i(t) \times \text{volume}_i}{\sum_{i=1}^{N} \text{volume}_i} \tag{4.2}$$

where volume_i denotes an exchange weight. In the implementation, weights are adapted using an exponential moving average (EMA) of observed displayed liquidity share, and side-specific VWAPs and total quantities are computed from aggregated bid/ask notionals and quantities.

4.2 DEX Trade Monitoring

Objective: capture DEX trades at transaction level with accurate execution prices.

Data source: we monitor Swap events from Uniswap V2 and Uniswap V3 pools using a configurable multi-chain listener. While the listener supports multiple blockchains through configuration (`dex-prices/config.json`), the analysis in this report uses Ethereum and BSC, extracting token amounts and timestamps from on-chain logs.

Execution price: for a swap from token0 to token1, execution price is computed from the realized amounts:

$$\text{execution\_price} = \frac{\text{amount1Out}}{\text{amount0In}} \tag{4.3}$$

4.3 Price Deviation Measurement

Objective: quantify deviation between DEX execution prices and the CEX price index.

Per-trade deviation:

$$\text{deviation}_i = \frac{\text{DEX\_price}_i - \text{CEX\_price\_index}_i}{\text{CEX\_price\_index}_i} \times 100\% \tag{4.4}$$

We match each DEX trade to the closest CEX price index value within a small time window around the trade timestamp.

4.4 Lead–Lag Correlation Analysis

Objective: determine the temporal relationship between CEX and DEX price movements.

We compute log return series and evaluate the cross-correlation function across positive and negative lags to identify whether CEX prices lead DEX prices or vice versa [8].

4.5 Slippage Modeling

Objective: characterize and predict slippage for varying trade sizes.

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

$$\Delta y = y - \frac{xy}{x + \Delta x} \tag{4.5}$$

$$\text{slippage}_{\text{theoretical}} = \left| 1 - \frac{\Delta y / \Delta x}{y/x} \right| \tag{4.6}$$

We complement theoretical slippage with empirical distributions derived from observed swaps to capture real execution outcomes, including the effects of on-chain congestion and transaction ordering.

Chapter 5

REQUIREMENTS SPECIFICATION

This chapter specifies the functional and non-functional requirements for the CMPE492 CEX–DEX market analysis system. Requirements are stated at the system level and mapped to implemented components in Chapter 7.

5.1 Functional Requirements

FR1: CEX data collection. The system shall maintain real-time WebSocket connections to multiple CEX APIs and ingest best bid/ask prices and quantities for the monitored pairs. It shall automatically reconnect on disconnections.

FR2: Price index calculation. The system shall compute a multi-exchange reference price per symbol using a volume-weighted mid-price aggregation. The index shall update whenever fresh ticker data arrives.

FR3: DEX swap monitoring. The system shall monitor Uniswap V2 and Uniswap V3 Swap events on Ethereum and BSC, and extract execution amounts, pool address, transaction hash, block number, and timestamps.

FR4: Deviation measurement. For each DEX swap, the system shall match the trade timestamp to the closest CEX price index within a small time window and compute CEX–DEX deviation.

FR5: Lead–lag analysis. The system shall provide a lead–lag analysis module that can compute cross-correlation across configurable lags and run causality tests when appropriate.

FR6: Slippage characterization. The system shall compute theoretical AMM price impact (V2 constant product; V3 concentrated liquidity when available) and build empirical slippage distributions from historical swaps.

FR7: Storage and querying. The system shall store time-series data for the price index, DEX swaps, and derived analytics into a database optimized for time-series queries, supporting both real-time visualization and retrospective analysis.

FR8: Visualization. The system shall provide a dashboard that displays real-time price index values and supports visualization of deviations, lead–lag results, and slippage curves as data becomes available.

5.2 Non-Functional Requirements

NFR1: Performance. The system shall process continuous real-time data streams without falling behind. Database writes should be batched where appropriate and queries should use time-bucketed access patterns.

NFR2: Reliability. Collectors shall tolerate external failures (temporary exchange downtime, RPC instability) through retries, reconnection, and redundancy.

NFR3: Extensibility. Adding a new symbol or exchange should require configuration changes rather than a full redesign. Multi-chain monitoring should be configurable per chain.

NFR4: Reproducibility. The system shall support reproducible analysis by persisting raw observations and providing deterministic analysis scripts for generating results.

5.3 Use Cases

This section summarizes representative end-to-end use cases supported by the system.

5.3.1 UC1: Monitor Real-Time Price Deviation

Primary actor: Research analyst. **Preconditions**: collectors and database are running. The user selects a trading pair and time range in the dashboard. The system displays the current CEX price index, the latest DEX execution price, and the current deviation, together with short-horizon aggregates (e.g., 5-minute and 1-hour averages) to contextualize transient spikes. **Postconditions**: the user can observe real-time alignment between venues and identify abnormal divergence periods.

5.3.2 UC2: Analyze Historical Lead–Lag Relationship

Primary actor: Research analyst. **Preconditions**: sufficient historical data is available. The user selects a pair and a historical window. The system computes a cross-correlation function over returns and reports the lag at which correlation peaks; when enabled, it also reports statistical testing outputs (e.g., Granger causality p-values) to support interpretation. **Postconditions**: the user obtains a quantitative estimate of information flow direction and characteristic lag.

5.3.3 UC3: Estimate Slippage for a Planned Trade

Primary actor: DEX trader / bot designer. **Preconditions**: slippage statistics or models have been generated for the target pool/pair. The user inputs trade direction and size and selects a pool. The system returns theoretical AMM price impact from current pool state and empirical slippage ranges from historical trades (e.g., P50/P90/P99) for comparable sizes. **Postconditions**: the user can estimate execution cost before submitting an on-chain transaction.

5.3.4 UC4: Compare CEX vs DEX Liquidity

Primary actor: Research analyst / liquidity provider. **Preconditions**: required liquidity data sources are available. The user selects a pair and compares CEX depth (aggregated across exchanges) against DEX effective liquidity around the current price. Liquidity ratios are summarized at standardized thresholds (e.g., ±0.5%, ±1%, ±2%) to make the comparison comparable across venues and time. **Postconditions**: the user can evaluate relative execution quality and capital efficiency across venue types.

Chapter 6

DESIGN

6.1 Information Structure

6.1.1 Database Schema Overview

The system stores market data in PostgreSQL with TimescaleDB hypertables for time-series efficiency. The core time-series tables are:

- `price_index`: aggregated reference price per symbol with metadata such as number of exchanges, side VWAPs, and aggregated bid/ask quantities.
- `dex_swaps`: decoded on-chain swaps with execution price and trade metadata.
- `dex_pool_state`: pool state snapshots (e.g., Uniswap V3 tick, sqrtPriceX96, liquidity) captured as historical time-series.
- `price_deviations`: deviation measurements between DEX execution price and CEX reference price.
- `slippage_analysis`, `correlation_analysis`: derived analytics tables used by the analysis engine.

Metadata tables store token and pool information:

- `tokens`: token addresses, symbols, decimals, chain.
- `pools`: pool address, chain, DEX identifier, token0/token1, fee tier.

Derived views and aggregates. For convenience and performance, the database includes a `dex_trades` view that normalizes swaps into a buy/sell trade format and a continuous aggregate (`dex_volume_1h`) for hourly swap volume rollups.

6.1.2 Entity-Relationship Diagram

The ER diagram describes relationships between pools, tokens, swaps and derived analytics.

6.2 Information Flow

6.2.1 Real-Time Data Flow

The system runs two primary real-time ingestion streams in parallel:

- **CEX stream:** WebSocket tickers from multiple exchanges are aggregated into a per-symbol price index and stored for downstream matching.
- **DEX stream:** Swap events are captured from EVM nodes and persisted with execution price information and metadata.

Deviation calculation and analysis jobs operate on the persisted time-series to generate results for visualization and reporting.

6.2.2 Activity Diagram: Real-Time Price Deviation Monitoring

The CEX stream continuously ingests best bid/ask updates from multiple venues and updates the reference price index whenever fresh data arrives. The DEX stream ingests on-chain Swap events as new blocks are produced and computes an execution price per trade. For each swap, the system queries the nearest contemporaneous CEX price index value in a narrow time window around the trade timestamp, computes deviation, persists both raw and derived records, and updates aggregate statistics used by the dashboard.

6.2.3 Sequence Diagram: Price Index Calculation

For a given symbol, the price index module collects per-exchange ticker snapshots, filters stale or anomalous inputs, and computes a volume-weighted reference price using mid-prices. The computed index value is persisted to the database with metadata (e.g., number of contributing venues) and optionally published for real-time visualization.

6.2.4 Sequence Diagram: DEX Trade Processing

When a swap is observed on-chain, the listener parses event fields, normalizes token amounts, and computes execution price. The system then matches the trade time to the closest CEX index observation, computes deviation, and stores the resulting record. Downstream analysis modules consume these stored observations to compute distributions, correlations, and slippage summaries.

6.3 System Design

6.3.1 Component Architecture

The system is structured as modular components:

- **CEX Collector** (`price-index/`, Go): maintains multiple WebSocket connections and computes the multi-exchange price index.
- **DEX Collector** (`dex-prices/`, Go): subscribes to new blocks and parses Uniswap V2/V3 swap logs on Ethereum and BSC.
- **Database Layer** (`database/`): PostgreSQL + TimescaleDB schema and helper clients.
- **Analysis Engine** (`analysis/`, Python): correlation/lead–lag, deviation statistics, slippage analysis, and report data generation.
- **Dashboard** (`price-index/dashboard.py`, Streamlit): visualization of real-time and historical metrics.

6.3.2 Interfaces and Data Contracts

All components communicate through the database schema (and optionally a shared-memory buffer for live dashboard display). This enables independent operation and simplifies recovery: collectors can restart without breaking analysis jobs as long as data is persisted.

6.4 User Interface Design (if applicable)

The system provides a Streamlit dashboard for monitoring the price index and inspecting analysis outputs interactively.

6.4.1 Dashboard Layout

The dashboard header includes trading pair selection, time range selection, and network/protocol filters where applicable. The main area is organized into tabs: (i) real-time monitoring (index vs latest DEX trade and deviation), (ii) lead–lag analysis (CCF plot and summary statistics), (iii) liquidity comparison (depth/liquidity views and ratios), (iv) slippage analysis (slippage curves and percentiles), and (v) volume analysis (time-series and distribution plots).

6.4.2 Interaction Patterns

Charts support hover tooltips for exact values and timestamps, zooming/drag selection for focusing on sub-periods, and export of plots/data (e.g., CSV, PNG) for reporting. Auto-refresh can be enabled for real-time monitoring while keeping historical analysis views stable.

6.4.3 Visual Design Principles

Visual encoding is kept consistent across views: CEX series use one primary color, DEX series use another, and deviations are highlighted to emphasize divergence periods. Numerical displays use aligned formatting for readability, and secondary controls are grouped to reduce clutter.

Chapter 7

IMPLEMENTATION AND TESTING

7.1 Implementation

7.1.1 CEX Collector (Go)

The CEX data collector is implemented in `price-index/`. It maintains concurrent WebSocket connections to multiple exchanges, continuously updates an in-memory shared-memory layout for live visualization, and computes a per-symbol reference price index. The index values are persisted to TimescaleDB at a configurable interval. The collector targets exchanges including Coinbase, MEXC, Binance, Gate.io, OKX, Bybit, HTX and KuCoin.

7.1.2 DEX Collector (Go)

The DEX listener is implemented in `dex-prices/`. It connects to EVM chains using RPC endpoints, subscribes to new blocks and swap logs, and parses Uniswap V2 and Uniswap V3 Swap events. Supported chains are configured in `dex-prices/config.json`; multiple networks are available, and the analysis in this report uses Ethereum and BSC. The listener persists swap events to `dex_swaps`, upserts token and pool metadata, and records pool state snapshots to `dex_pool_state` for historical analysis. Token and pool metadata are cached to reduce repetitive on-chain calls.

7.1.3 Database (PostgreSQL + TimescaleDB)

The database schema is located in `database/`. TimescaleDB hypertables store tick-level observations efficiently and enable time-window queries. Adminer provides a lightweight web interface for inspecting tables during development.

7.1.4 Analysis Engine (Python)

The analysis modules in `analysis/` provide statistical computation over stored data, including price deviation analysis (`analysis/deviation_calculator.py`), lead–lag correlation (`analysis/correlation.py`), slippage analysis (`analysis/slippage.py`), and volume analysis (`analysis/volume_analysis.py`). The script `analysis/run_analysis.py` orchestrates end-to-end runs and stores outputs in `price_deviations`, `correlation_analysis`, and `slippage_analysis`.

7.2 Testing

7.2.1 Data Validation and Sanity Checks

Testing focuses on verifying correctness of ingestion and analytics:

- **Connectivity tests:** WebSocket and RPC connectivity, reconnection logic.
- **Schema tests:** database schema validation using `database/test_schema.py`.
- **Data sanity:** range checks for prices and volumes, timestamp recency, and cross-venue consistency checks.

7.3 Deployment

The system runs locally in development and can be deployed to a server for continuous collection. The database runs via Docker Compose, while collectors and analysis jobs run as long-lived processes or scheduled jobs. The appendix includes operational notes and deployment details.

Chapter 8

RESULTS

8.1 Overview

This chapter presents results on CEX–DEX price deviations, lead–lag relationships, and slippage behavior for BTC/USDT, ETH/USDT and BNB/USDT across Ethereum and BSC.

8.2 Dataset Summary

This section summarizes the collection period, number of CEX ticker updates, number of DEX swaps captured, exchange and chain coverage, and any missing-data intervals. Although the DEX listener supports additional networks via configuration, the reported dataset covers Ethereum and BSC.

Item	**Value (TBD)**
Collection period	TBD
Symbols	BTC/USDT, ETH/USDT, BNB/USDT
CEX exchanges	Coinbase, MEXC, Binance, Gate.io, OKX, Bybit, HTX, KuCoin
DEX protocols	Uniswap V2, Uniswap V3
Chains	Ethereum, BSC
#CEX ticker records	TBD
#DEX swaps	TBD

Table 8.1: Dataset summary (placeholder)

8.3 CEX–DEX Price Deviation

We report deviation distributions (mean, standard deviation, MAD, and percentiles) per symbol, separated by chain and protocol where applicable. We also include time-series plots highlighting periods of persistent deviation and major outliers.

Placeholder: deviation time-series plot (CEX price index vs. DEX execution price and deviation %).

Figure 8.1: CEX–DEX deviation over time (placeholder)

8.4 Lead–Lag Analysis

We present cross-correlation function (CCF) plots and the lag value at which correlation peaks, indicating whether CEX prices lead DEX prices and by how much. Statistical testing results (e.g., Granger causality) are also reported where applicable.

Placeholder: lead–lag cross-correlation plot (CCF) per symbol.

Figure 8.2: Lead–lag cross-correlation between CEX and DEX returns (placeholder)

8.5 Slippage and Execution Cost

We report empirical slippage distributions by trade size bin and compare them with theoretical AMM price impact estimates. Results are summarized with slippage curves (median and tail percentiles).

Placeholder: slippage vs trade size curve (P50/P90/P99).

Figure 8.3: Slippage as a function of trade size (placeholder)

8.6 Liquidity Comparison

We compare CEX depth at ±0.5%, ±1%, and ±2% to the effective DEX liquidity around the current price, and report liquidity ratios per symbol.

8.7 Dashboard Snapshot

We include dashboard screenshots and describe the interaction flow (pair selection, time-range selection, and displayed metrics).

Chapter 9

CONCLUSION

This report presented an end-to-end system for monitoring and analyzing market microstructure differences between centralized and decentralized cryptocurrency exchanges. By combining a multi-exchange CEX price index with on-chain Uniswap V2/V3 trade monitoring across Ethereum and BSC, we enable consistent measurement of price deviations, lead–lag effects, and execution costs for BTC/USDT, ETH/USDT and BNB/USDT.

As final plots and tables are incorporated, the report will quantify how closely DEX execution tracks the CEX reference price, how lead–lag behavior varies across market conditions, and how slippage scales with trade size under real-world conditions.

9.1 Limitations and Future Work

Key limitations include external dependency reliability (exchange APIs and RPC endpoints), incomplete coverage of on-chain transaction ordering effects, and the need for longer continuous datasets for stronger statistical significance. Future work includes improved MEV classification, broader DEX coverage, deeper liquidity measurement (CEX orderbook snapshots and Uniswap V3 liquidity distribution), and more robust model validation across regimes.

Bibliography

[1] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 core. *Uniswap Labs*, 2021. Available at https://uniswap.org/whitepaper-v3.pdf.

[2] Carol Alexander, Jun Deng, and Jiawei Feng. Price discovery and efficiency in uniswap liquidity pools. *Journal of Financial Markets*, 2025. Forthcoming.

[3] Binance. Binance api documentation. API Documentation, 2024. Available at https://binance-docs.github.io/apidocs.

[4] Vitalik Buterin. Ethereum: A next-generation smart contract and decentralized application platform. *Ethereum White Paper*, 2014. Available at https://ethereum.org/en/whitepaper.

[5] Agostino Capponi, Ruizhe Jia, et al. Transaction costs on decentralized exchanges: The case of uniswap v3. *Management Science*, 2024. Forthcoming.

[6] DefiLlama. Defi total value locked dashboard. Web Platform, 2024. Available at https://defillama.com.

[7] Empirica Labs. Uniswap v3 liquidity analysis framework. White Paper, 2024. Available at https://empirica.io.

[8] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

[9] Joel Hasbrouck. *Empirical Market Microstructure: The Institutions, Economics, and Econometrics of Securities Trading*. Oxford University Press, New York, 2007.

[10] Hyperliquid Labs. Hyperliquid: High-performance decentralized exchange. Technical Documentation, 2024. Available at https://hyperliquid.gitbook.io.

[11] Kaiko Research. Dex market structure report 2024. Technical report, Kaiko, 2024. Available at https://www.kaiko.com.

[12] Alfred Lehar and Christine A. Parlour. Decentralized exchange markets: Design choices and market performance. *Journal of Financial Economics*, 141(3):951–977, 2021.

[13] solidquant. Cex-dex arbitrage template. GitHub Repository, 2024. Available at https://github.com/solidquant/cex-dex-arb.

[14] Kai Wu, Yuxuan Zhang, and Ming Chen. Cex-dex arbitrage and maximal extractable value. *Review of Financial Studies*, 2025. Forthcoming.

[15] Jeff Yan and Iliensinc. Building a high-performance on-chain order book. *ArXiv preprint*, 2024. arXiv:2024.xxxxx.

Appendix A

SAMPLE APPENDIX

Contents of the appendix.