Received: 2 August 2023 | Revised: 3 February 2024 | Accepted: 22 February 2024

DOI: 10.1002/nem.2266

SPECIAL ISSUE PAPER

WILEY

Athena: Smart order routing on centralized crypto exchanges using a unified order book

Robert Henker[1] | **Daniel Atzberger**[2] | **Jan Ole Vollmer** | **Willy Scheibel**[2] | **Jürgen Döllner**[2] | **Markus Bick**[3]

[1]XU Exponential University, Potsdam, Germany

[2]Hasso Plattner Institute, Digital Engineering Faculty, University of Potsdam, Potsdam, Germany

[3]ESCP Business School, Berlin, Germany

Correspondence
Robert Henker, XU Exponential University, Potsdam, Germany.
Email: mail@roberthenker.com

Summary

Most cryptocurrency spot trading occurs on centralized crypto exchanges, where offers for buying and selling are organized via an order book. In liquid markets, the price achieved for buying and selling deviates only slightly from the assumed reference price, that is, trading is associated with low implicit costs. However, compared to traditional finance, crypto markets are still illiquid, and consequently, the reduction of implicit costs is crucial for any trading strategy and of high interest, especially for institutional investors. This paper describes the design and implementation of Athena, a system that automatically splits orders across multiple exchanges to minimize implicit costs. For this purpose, order books are collected from several centralized crypto exchanges and merged into an internal unified order book. In addition to price and quantity, the entries in the unified order book are enriched with information about the exchange. This enables a smart order routing algorithm to split an order into several slices and execute these on several exchanges to reduce implicit costs and achieve a better price. An extensive evaluation shows the savings of using the smart order routing algorithm.

1 | INTRODUCTION

Presently, the majority of cryptocurrency trading occurs on *centralized crypto exchanges* (CEXs), constituting a substantial 73.2% of the overall trading volume as of June 2023.[1] Acting as intermediaries between buyers and sellers, CEXs differ from their counterparts, decentralized crypto exchanges (DEXs), which rely on smart contracts for facilitating trades.[2] Notable examples of CEXs include *Binance*, *Coinbase*, and *Kraken*, while popular DEXs encompass *Uniswap*, *Sushiswap*, and *PancakeSwap*. One key advantage of CEXs is their acceptance of traditional payment methods, allowing users to acquire cryptocurrencies using fiat currencies. In contrast, DEXs require other cryptocurrencies or stablecoins for trading.[3]

On CEXs, the process of buying and selling a target currency, such as *Bitcoin*, against a base currency, like USD or other cryptocurrencies, is facilitated through an order book. This order book continuously aggregates unfulfilled orders due to the absence of corresponding counteroffers.[4] The bid side comprises open buy offers, while the ask side consists of open sell offers. Each side is organized into levels, represented as tuples indicating price (in the base currency) and size (in the target currency). The bid side is sorted in descending order, and the ask side in ascending order. For instance, the first eight levels for both sides are illustrated in Figure 1. Taker orders, lacking price limits, execute instantly at the best available price. They consume volumes from the first levels of the opposing side until the order is fulfilled. For example, consider a buy order without limit of size one BTC in the example of Figure 1. The execution of this order would fill all offers of the first seven levels and parts of the eighth level of the ask side. The average price would then be given by

$$\begin{aligned}&0.0007 \cdot \$28,870.00 + 0.0007 \cdot \$28,880.00 + 0.4717 \cdot \$28,882.00 + 0.0007 \cdot \$28,890.00 + 0.0950 \cdot \$28,894.50 + 0.0345\\ &\quad \cdot \$28,895.00 + 0.0165 \cdot \$28,896.00 + 0.3802 \cdot \$28,899.50\\ &\quad = \$28,890.52\end{aligned} \tag{1}$$

which deviates from the shown reference price ($28,869.50) by approximately 0.073%. Such price deviations due to limited liquidity are called implicit costs or slippage.[5] Although this deviation may seem small, it is particularly significant for institutional investors whose investment strategies require a large number of trades, for example, with only one trade per day, the price distortion would add up to 365·0.073% = 26.65%. In traditional finance, it is known that those implicit costs significantly impact the overall return more than explicit costs, for example, trading fees.[6] However, in most research papers, implicit costs that arise from the order book are not considered; the analysis primarily relies on a reference price, for example, the average between best ask and best bid or tick data, for example, price–volume pairs that are added to a list every time a trade is executed, as shown in Figure 1. The main reason for this is the complicated collection and processing of order book data.

Smart Order Routing Algorithms (SOR) aim to reduce implicit costs by splitting an order into smaller orders (also called order slices) and placing them on several exchanges or temporally offset.[7] This requires not only the tracking of multiple order books across different exchanges but also their accurate handling.

This paper builds upon the groundwork laid by Henker et al in their previous work introducing *Hephaistos*.[8] Notably, Hephaistos amalgamates order books for a given currency pair across multiple exchanges, creating a *Unified Order Book* (UOB). In our endeavor to minimize implicit costs, we introduce a novel component, *Athena*. Unlike Hephaistos, Athena goes beyond mere data aggregation by applying a price mechanism to the UOB, which results in a splitting of orders. By seamlessly integrating Athena into the existing infrastructure, our research extends the initial contributions made by Henker et al in the following ways:

1. We provide a descriptive analysis of existing order book data from nine exchanges concerning volume, variety, velocity, and veracity. We conclude that order book data exemplifies the *Big Data* phenomenon.[9]

FIGURE 1 Visualizations for tick data and order book for BTC/USDT from the exchange *Bitmex*. (Left) Extract from the order book. The ask side is marked in red, and the bid side is marked in green. (Middle) Candlestick chart showing price range of recent trades on an hourly basis: depth chart displaying the aggregated volume for the bid and ask side with the ask side on the right colored in red and the bid side on the left colored in green. (Right) Tick data on recent trades, i.e., every time a trade is executed the corresponding tick is added to the list together with a time stamp.

2. We evaluate the performance of Athena. We compare the *Xetra Liquidity Measure* (XLM) for three currency pairs on six exchanges with the respective UOB and calculate the effect of the SOR on implicit costs.

The remaining part of this work is structured as follows: Section 2 presents existing approaches for SOR and other works originating from the cryptocurrency domain. In Section 3, we introduce liquidity measures and give details on the UOB and our SOR. The architecture of the underlying infrastructure is presented in Section 4. Based on a benchmark dataset, we evaluate the effectiveness of our SOR algorithm for reducing implicit costs; the experimental setup and the results together with threats to validity are discussed in Section 6. We conclude this work and point out directions for future work in Section 7.

2 | RELATED WORK

When trading cryptocurrencies, one of the most relevant circumstances that needs to be considered is slippage, which "measures the difference between the expected trading price and the actual trading price."[10] However, most studies ignore the slippage effect and only consider reference prices, as shown in the comprehensive survey of Fang et al.[11] Different liquidity measures were introduced to capture the structure of an order book and relate it to implicit costs. In the following, we will present existing approaches for collecting order books from CEX and measuring liquidity. We will furthermore discuss existing techniques and services for SOR, which aim to reduce the slippage. Work that addresses fraud detection[12,13] or price prediction[14–17] is neglected in our presentation of the related work because it does not include order books to date.

2.1 | Collecting order books from CEX

Most studies that analyze order book data rely on external services, for example, *CryptoTick*.

Dilbagi introduced a comprehensive processing pipeline for gathering order book data from a single exchange, *Binance*, focusing on detecting market manipulations through visualization.[18]

Burján and Gyires-Tóth proposed a GPU pipeline for processing large amounts of order book data for subsequent machine learning tasks.[19] Their system collects four currency pairs from (*Coinbase*) through its API following a preprocessing and data normalization step.

So far, Henker et al have presented the most powerful system for collecting order book data from CEX and a subsequent provisional algorithm for SOR. Their system *Hephaistos* serves as a component within a comprehensive infrastructure technology suite of an institutional grade digital asset analytics and low-latency trading engine. It collects and processes order book data from 22 exchanges, with 55 currency pairs in real time covering roughly 34% of the daily spot market. As this work mainly builds up on the ideas of Henker et al, we will detail their technical considerations in the subsequent sections.

Huang and Polak developed a similar architecture for collecting order books.[20] Their system *LOBSTER* consists of three types of modules, so-called readers, processors, and writers. In their presentation, the authors demonstrate the application of the system to order books of *Nasdaq*-traded stocks.

2.2 | Liquidity analysis for CEX

For retail investors, the bid–ask spread, that is, the difference between the lowest selling price and the highest purchase price, is the most relevant liquidity measure.[21] Dyhrberg et al studied three CEXs and showed that the spread for BTC/USD is smaller than spreads in traditional equity exchanges. They also demonstrated that retail investors dominate the market by analysing the trading activity. Similar results were shown by the analysis of Brauneis et al.[22] Furthermore, the authors showed, in a detailed analysis of intraday patterns in trading activity, volatility, and liquidity, the absence of institutional frictions, such as overnight trading, indicating retail investors' dominance.[23]

Angerer et al presented the most extensive study on liquidity on CEXs.[10] The authors compared the spread, the order book depth, and the order book imbalance on more than 500 currency pairs from four exchanges covering

9 months in 2019. Surprisingly, their results revealed that lower spreads and imbalances positively correlate with higher implicit costs, which contradicts the idea of liquidity. Our work mainly focuses on an institutional trader's perspective by measuring the implicit costs in a low-latency infrastructure.

Jobst et al computed several liquidity measures, including the bid–ask spread and the XLM, on three CEXs for three currency pairs throughout 2022.[5] Their results revealed that the events around the *FTX collapse*[24] and *Terra Luna crash*[25] significantly caused changes in liquidity. In a similar work, Chortane and Naoui studied the impact of the Covid-19 crisis on crypto market liquidity.[26] Their results showed that, except for a few cryptocurrencies, for example, *Cardano*, no impact could be detected in liquidity's short-term and long-term development.

Complementary to investigating the impact of macroeconomic events on cryptocurrency liquidity, Brauneis et al examined the relationship with the activity on the underlying blockchain.[27] Their results showed that liquidity on crypto markets is unrelated to the broader financial market and mainly driven by the activity on the blockchain.

2.3 | Smart order routing algorithms and existing services

In traditional financial markets, the use of SOR is a de facto standard to encounter fragmented markets and assure efficiency.[28] Especially after the introduction of the *MiFID* (Markets in Financial Instruments Directive) framework by the European Union in 2004 and its updated version in 2018, which introduced best execution. This required financial institutions to take all reasonable steps to obtain the best possible result for their clients' orders, taking into account price, costs, speed, likelihood of execution and settlement, size, nature, or any other consideration relevant to order execution.[29] This consequently increased competition among market infrastructure providers, and markets became more fragmented and thus requiring SOR.[30] The impact of SOR for reducing implicit costs has been investigated by Ende et al on order books of stocks contained in the *EURO STOXX 50*.[31]

Jeon et al studied the impact of the effects on liquidity when trading across multiple exchanges through a UOB.[32] Thus, their research question is similar to our work but limited to one crypto currency Bitcoin. Another relevant difference is that their dataset is based on CryptoTick, and no infrastructure covering the entire processing pipeline and ensuring quality, accuracy, and validity required for not only calculation but according execution is presented. The analyses by Jeon et al show similar results to our experiments and demonstrate the effectiveness of an SOR based on the UOB. The UOB shows that the BTC/USD market is highly fragmented. Although this results in higher slippages on single exchanges, it not only supports the idea of SOR but also creates arbitrage opportunities, as shown by Al-Yahyaee et al,[33] especially in market phases of low liquidity and high volatility.

Following to the state of research, we present companies that claim to offer comparable SOR services and try to examine scope and actual availability based on the limited available sources.

In 2020 *LCX*, a Liechtenstein-based and regulated CEX introduced the *LCX Terminal* as an SOR service.[34] The service was limited to selected exchanges where LCX provided the integration through trading APIs of the exchanges without publishing information on latency or execution quality. The service, however, is no longer available due to a lack of demand, as stated by LCX.[35]

Shrimpy provides a crypto analytics tool and trading bot. In 2019, it first announced plans to offer an SOR service.[36] However, the service is also limited to selected exchanges and pairs and requires the user to connect his personal exchange accounts. In June 2023, Shrimpy announced that they were ceasing operations and closing their business. Although it claimed to have almost 100,000 customers and a trading volume of $14bn, the company did not concretized how much of that originated to SOR and stated that market conditions are unsustainable due to low trading volumes following the FTX crash in November 2022.[37]

SwissBorg, a centralized CEX, announced building an SOR and liquidity aggregation service first in 2021 as part of their initial coin offering (ICO).[38] According to SwissBorg, only five exchanges (*Binance*, *HitBTC*, *LMAX*, *Kraken*, and *Bitfinex*) were available. When writing (Q3 2023), Binance represents the most liquid exchange, and the five exchanges seem sufficient for retail investors. However, this might change following the recent regulatory issues faced by Binance. Certainly, it is not enough for institutional trading firms trading higher volumes, thus requiring more exchanges for placing large-volume orders.

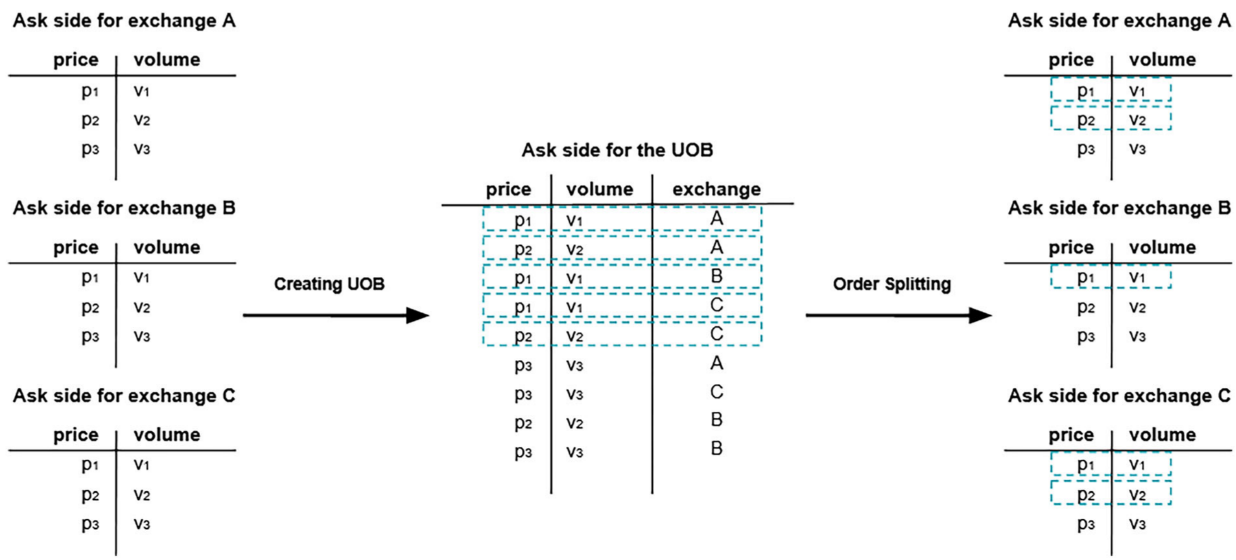

FIGURE 2 Procedure of the SOR. In the first step, the UOB is created by aggregating the order books from the single exchanges. The figure exemplifies this on the ask side for three exchanges. The resulting UOB, therefore, contains three columns, with the third column denoting the exchange. In the second step, the order is executed on the UOB, which results in a splitting of the order over the three exchanges.

3 | ATHENA: SOR USING A UOB

Our system, Athena, implements an SOR algorithm, following the methodology proposed by Henker et al.[8] The algorithm comprises two main steps: (1) the creation of a UOB and (2) the splitting of orders for execution, as depicted in Figure 2.

In the first step, order books from multiple exchanges are aggregated into a single UOB. Each level in the UOB is represented as a triple (p, v, e), denoting the price p, volume v, and originating exchange e. The levels are sorted by price, with the ask side arranged in ascending order and the bid side in descending order. The UOB thus diverges from traditional representations in that it is only monotonically in/decreasing, but not strictly monotonically, that is, if two different exchanges e_a and e_b contain two levels (p, v_a) and (p, v_b) with the same price p, they will occur as separate entries (p, v_a, e_a) and (p, v_b, e_b) within the UOB. Notably, the UOB may contain overlapping bid and ask sides, reflecting instances where the best bid and ask prices originate from different exchanges. The UOB construction facilitates price improvement opportunities, as demonstrated in Table A1. By consolidating order book data from multiple exchanges, Athena optimizes the execution process and reduces implicit costs associated with price deviations.

In the second step, orders to buy or sell are executed based on the UOB. Partial orders are executed on individual exchanges, ensuring optimal buy or sell prices are achieved at each point in time. However, it is important to note that sufficient funds must be deposited at each exchange for order execution, as fund transfers are constrained by exchange transaction speeds. While SOR algorithms like Athena offer significant advantages in optimizing trade execution, they also face limitations related to fund transfer speeds and market efficiency. As such, ongoing refinement and adaptation are necessary to address evolving market dynamics and optimize trading strategies.

4 | ARCHITECTURE OF HEPHAISTOS

Our system Hephaistos collects order book data from the exchanges and aggregates them into a UOB, which is the input for Athena. Figure 3 delineates the elements of our system and their dispersion among servers. The implementation of the order book reconstruction and archiving components is in Python, and we employ two distinct PostgreSQL databases located on separate servers to ensure redundancy.

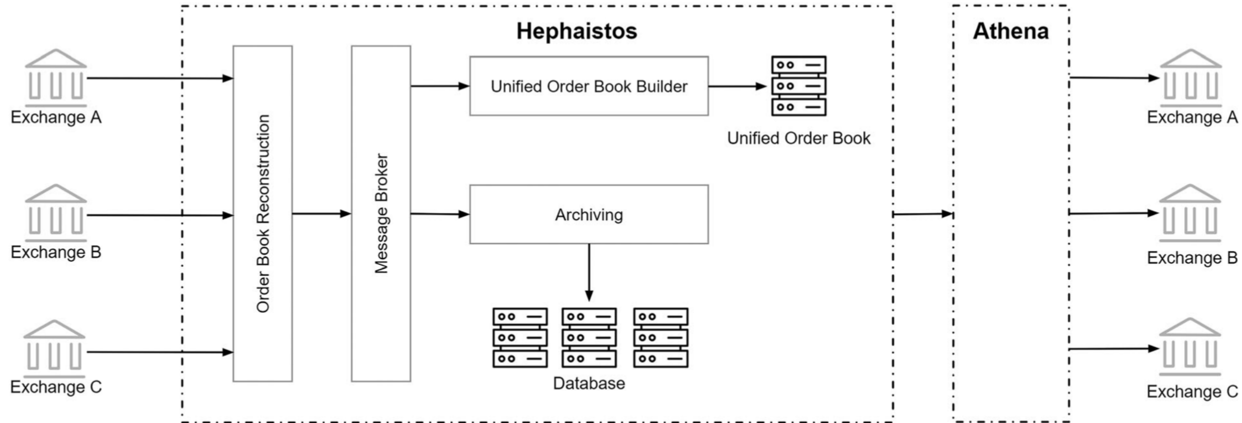

FIGURE 3 System overview. Hephaistos reconstructs the order books from different exchanges and archives them in a database and aggregates the UOB. Athena relies on the UOB for its SOR algorithm, which splits an order into several orders that are then placed on the exchanges.

4.1 | Order book reconstruction

Many exchanges provide a WebSocket or FIX API for receiving incremental updates to the order book. Typically, this process begins with fetching or receiving a snapshot of the entire order book, followed by subscribing to a stream of incremental updates. Each update contains only the price levels that have changed since the previous update. Updates can fall into three categories: (1) adding a new price level, (2) altering the volume of an existing price level, or (3) deleting a price level. Often, a (p,v)-tuple representation is used for all three cases, with a volume of 0 indicating the deletion of a level.

The client then reconstructs the order book accurately by applying each update to its local state, ensuring synchronization with the source exchange's order book. Various options exist for verifying the integrity of the reconstructed order book, but their availability varies widely among exchanges.

Update timestamps serve as the most fundamental variant, enabling verification that updates have been received in the correct sequence.

Update sequence numbers can be viewed as an extension of update timestamps. They are anticipated to be strictly incremental, thereby facilitating the detection of any missing updates.

Checksums represent the most comprehensive approach. They not only verify the accurate reception of updates but also validate the correctness of the reconstructed order book. These checksums are computed over the initial n levels of the order book subsequent to applying the respective update.

Our system incorporates all three integrity checks, leveraging the options provided by different exchanges. The reconstruction component of our system encapsulates exchange-specific message parsing and API behavior. This includes handling maximum order book depth. Some exchanges send explicit deletions for levels that exceed the maximum depth due to intermediately inserted levels, while others expect the client to discover and manage this. The reconstruction component produces streams of fully reconstructed order book snapshots in a standardized format. To distribute the computational workload across multiple CPU cores, our system runs one or more reconstruction processes per connected exchange. All resulting snapshots are then fed into a central message broker.

4.2 | Message broker

We implemented a central *Apache Kafka* message broker, adhering to the publisher–subscriber model. This architecture enables flexible scaling-out and distribution across servers on both ends. By adopting this approach, we eliminate the need for n:m connections between publishers (responsible for order book reconstruction) and subscribers (such as archiving). Additionally, it prevents redundant reconstruction of the same order book across different processes.

Apache Kafka offers sufficiently low end-to-end latency and provides intermediate storage with guaranteed message ordering. This feature serves as both a cache and a means to bridge load spikes and outages in the database and archiving services, thereby ensuring the completeness of archived data.

To optimize performance, we utilize separate Kafka topics for each trading pair on every exchange. This strategy enables consumers to selectively subscribe to the required markets only, thereby reducing the overall message load on the broker and enhancing consumer efficiency.

4.3 | Archiving

We archive all received order book data without any aggregation or reduction, preserving it for later analysis.

The archiving component subscribes to all Kafka topics and transforms the received data into the database storage format detailed in Section 4.4. This process involves deriving incremental changes from the stream of snapshots to minimize required storage space. The archiver maintains the current state in memory and only generates a new database entry once a level has been removed or replaced, thus determining the *valid_until* timestamp. This approach reduces the number of write operations to the database and ensures consistency compared to saving every level upon arrival and updating the *valid_until* timestamp later. However, this method may result in a delay until the full snapshot for a particular point in time becomes available in the database, especially for levels further down in the order book that change less frequently.

To optimize transaction processing overhead in the database, the resulting table rows are committed in batches. With each batch, the archiving service stores the latest message id whose data is included, for every Kafka topic, in the same transaction. This mechanism ensures that in the event of a crash or restart, the archiving service can resume from the exact point in the order book stream where archiving halted, without any loss of data or integrity violations.

The archiving workload can be efficiently distributed over multiple processes across different Kafka topics since encoding and database writes for distinct streams do not conflict.

4.4 | Database storage

Our database storage concept integrates three key aspects to optimize space efficiency while ensuring acceptable speed for typical workloads.

4.4.1 | Validity timestamps

Figure 4 illustrates a simplified example of an order book and its evolving levels over time. Each entry for a specific price level includes a corresponding volume and persists for a defined duration until it is either substituted with a

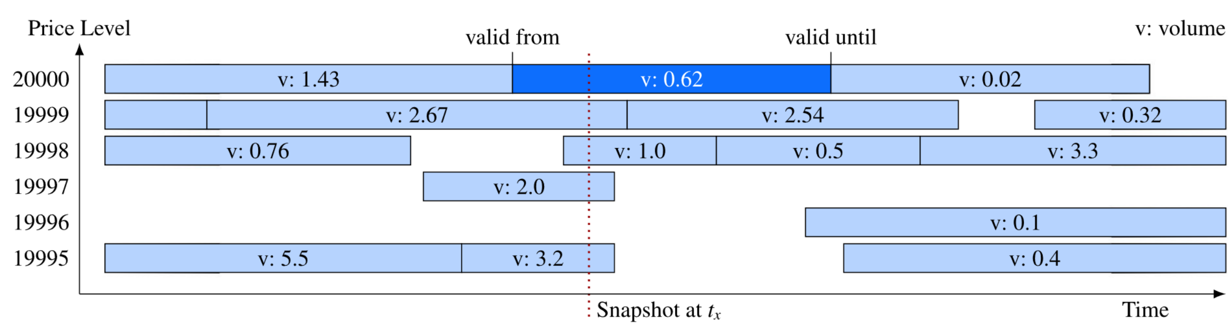

FIGURE 4 Simplified illustration of the order book storage format showcasing evolving levels over time. The figure demonstrates the order book's structure, where each entry represents a specific price level accompanied by its corresponding volume. Entries persist until replaced or deleted, denoted by the timestamps *valid_from* and *valid_until*. The complete snapshot at any time t_x encompasses entries with *valid_from* $\leq t_x$ and *valid_until* $> t_x$.

different volume or deleted. We characterize the lifespan of such an entry using two timestamps: *valid_from* marks the entry's initial inclusion in the order book, while *valid_until* indicates when the entry is replaced or removed. The complete snapshot of the order book at any given time t_x comprises all entries with *valid_from* $\leq t_x$ and *valid_until* $> t_x$.

The cornerstone of our storage format lies in storing each entry as an individual row, incorporating the *valid_until* timestamp. This approach facilitates direct retrieval of the order book snapshot for any given point in time without necessitating an initial snapshot and the subsequent replaying of incremental updates received from the exchanges. Moreover, it maintains space efficiency compared to storing full order book snapshots after every update.

4.4.2 | Table partitioning

Based on observations of various use cases and queries executed on the data, we have determined that the majority of queries involve data from only a small subset of markets, often just one, and seldom merge bids and asks. This characteristic enables the utilization of *table partitioning* to enhance query response time.[39]

When employing magnetic hard disks, disk response time significantly influences the total time required to filter a large table. To mitigate the amount of data that needs to be searched and subsequently loaded from disk, we adopt separate tables for bids and asks. Each of these tables utilizes Postgres' built-in table partitioning mechanism to physically store data from different markets at distinct locations on the disk. Queries can directly access the required partition or utilize the parent table with a filter for the market, allowing the query planner to automatically determine the appropriate partition(s). In both scenarios, the amount of data needing to be loaded from disk is significantly reduced, thereby reducing the query response time accordingly.

Furthermore, this approach expedites sequential accesses of specific tables, such as during data export for a specific time range or during replaying for backtesting, due to the improved spatial locality on disk.

4.4.3 | BRIN indexes

The search for a specific point in time or time range using the *valid_from* and *valid_until* fields serves as a common filter criterion in our queries. While traditional b-tree indexes can expedite the search of a table for a particular value, they typically require as much disk space as the target column(s), resulting in significant space overhead for tables with numerous rows but few columns.

Given the insert-only nature of the tables facilitated by the archiver, where data are solely inserted and never modified or deleted, the order of rows inserted into the tables, and consequently their arrangement on disk, aligns with the *valid_from* and *valid_until* timestamps. The *Block Range Index (BRIN)* index emerges as a highly specialized index suitable for cases where the target column correlates with the disk's order.[40] BRIN divides the table rows into blocks and retains the value range (minimum and maximum) for the target columns in each block. The lookup involves a linear search of block summaries for blocks whose value range encompasses the value in question, followed by linear searches of every identified block for exact matches. This indexing mechanism delivers sufficient speed-up for our use case with minimal space overhead for storing the index data.

4.5 | UOB builder

The primary concept behind a UOB involves aggregating price levels and associated volumes from multiple exchanges that list a given pair. Each price level within the UOB includes the originating exchange, along with price and volume information. If multiple exchanges offer the same price level, they are not merged but included separately to facilitate subsequent separation. An illustrative example of a UOB derived from two order books is presented in Table 1.

The construction of the UOB is implemented as a distinct component—a stream processor. In this setup, a separate process receives individual order book updates from Kafka, constructs the UOB, and then returns the result to Kafka. This approach offers the advantage of performing the calculation only once, enabling multiple engine instances to utilize the same result. However, it does come with the drawback of increased latency and overall computational effort if only one instance utilizes the result.

TABLE 1 Spot exchanges considered in our experiments. The data are provided by *CoinMarketCap* (https://coinmarketcap.com/rankings/exchanges/)—which means that the data were not available. The score and average liquidity are computed according to an internal weighting scheme of CoinMarketCap.

Exchange	Score	Trading vol (24 h) in Mio.	Avg. liquidity	Weekly visits in Mio.	# markets	# coins	# fiat supported
Binance	9.9	14,464	829	11.8	1518	390	12
Bequant	—	—	—	—	—	—	—
Bitmex	5.3	0.23	461	0.15	101	58	—
Bitstamp	7.3	233	603	0.26	175	79	3
CEXIO	4.4	8	479	0.12	292	118	3
Huobi	—	—	—	—	—	—	—
Kraken	8.4	1354	732	1.18	767	241	7

5 | EXPERIMENTAL SETUP

In the following, we present details on the dataset used in our experiments and the liquidity measures used to evaluate the performance of the SOR. To guarantee reproducibility and provide a benchmark for further research, we made our data publicly available as a Zenodo repository.[1]

5.1 | Data

Our underlying system boasts support for over 20 exchanges, yet for the sake of performance in our experiments, we focus solely on order books from six prominent centralized exchanges (CEX)—namely, Binance, Bitstamp, Bequant, CEXIO, Huobi, and Kraken—spanning the year 2022. Details on the exchanges is provided in Table 1.

Our evaluation centers on three key currency pairs: Bitcoin (BTC), Ethereum (ETH), and Ripple (XRP), each paired with USD as the base currency. The order books, presented in NDJSON format, provide order book snapshots encapsulated within individual JSON structures, facilitating seamless loading and storing within a Python environment.

To ensure consistency, we confine our evaluation to hourly snapshots of order books, capturing the first 20 levels of both the bid and ask sides. Although our system supports up to 100 levels, we justify this restriction based on volume considerations. The volume vol_{20} of the first 20 levels of both sides of the order book is given as the sum:

$$vol_{20} = \sum_{i=1}^{20} p_{ask,i} \cdot v_{ask,i} + \sum_{i=1}^{20} p_{bid,i} \cdot v_{bid,i}, \tag{2}$$

measured in the base currency, that is, USD.

We depict the distribution of volumes across exchanges for the year 2022 using a boxplot (Figure 5), showcasing essential statistics such as the minimum, maximum, median, and first and third quartiles. Notably, the plot illustrates a broad spectrum of volume ranges across exchanges, with BTC/USD volume on Binance, for instance, ranging from below 10,000 to over 1,000,000.

5.2 | Evaluation measure

Various liquidity measures exist, each quantifying specific aspects of an order book and associated implicit costs.[41] For retail investors, the most relevant liquidity measure is the bid–ask spread, that is, the difference between the best ask and the best bid price. In our considerations, we rely on the XLM measure, as it captures the idea of implicit costs that arise as price deviations from a reference price.[6] Given a snapshot of an order book, where $l_{ask,i} = (p_{ask,i}, v_{ask,i})$ and

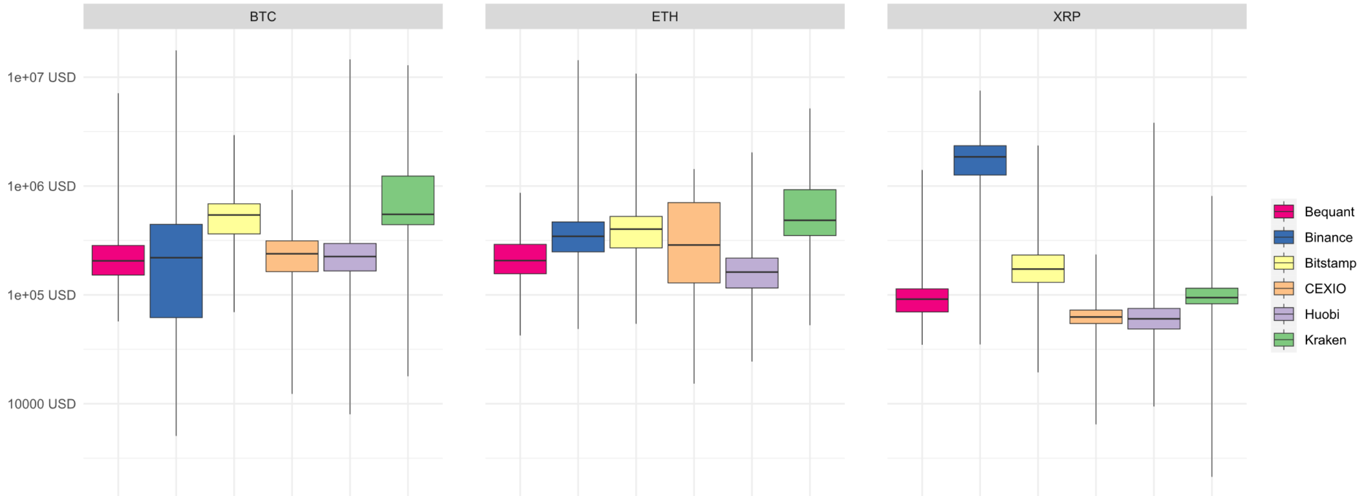

FIGURE 5 Boxplot showing the five-number summary for each order book for each combination of exchange and currency pair for 2022. The values on the vertical axis are given in USD. The lower and upper bounds of each rectangle denote the first and third quartiles, the line between the median. The verticals start and end at the minimum and the maximum, respectively.

$l_{bid,i} = (p_{bid,i}, v_{bid,i})$ specify the ith level of the ask and bid side, respectively, the reference price p_{ref} is given as the mean of the best ask $p_{ask,1}$ and best bid $p_{bid,1}$, that is,

$$p_{ref} = \frac{p_{ask,1} + p_{bid,1}}{2}. \tag{3}$$

Given an amount vol, measured in base currency, for which a target currency is to be acquired, it requires level $l_{ask,1} = (p_{ask,1}, v_{ask,1}), ..., l_{ask,k-1} = (p_{ask,k-1}, v_{ask,k-1})$ and a fraction of the kth level $(p_{ask,k}, \widetilde{v}_{ask,k})$ of the ask side are for the complete execution of the buy order. The deviation $\Delta_{ask}(vol)$ of the obtained average price from the reference price is therefore given by

$$\Delta_{ask}(vol) = \frac{vol}{\widetilde{v}_{ask,k} + \sum_{i=1}^{k-1} v_{ask,i}} - p_{ref}. \tag{4}$$

The XLM for the ask side is then defined as

$$XLM_{ask}(vol) = \frac{\Delta_{ask}(vol)}{p_{ref}}. \tag{5}$$

The XLM_{bid} for the bid side is defined analogously. Their sum yields the total XLM, that is,

$$XLM(vol) = XLM_{ask}(vol) + XLM_{bid}(vol). \tag{6}$$

Table 2 contains the maximal value for the XLM_{ask} for the ask side in the case of an invested volume of $1,000,000. For Bitcoin, the implicit costs are low and would only become relevant with a high frequency of trades. For Ethereum and Ripple however, the implicit costs when buying a large volume could become a major issue depending on the exchange.

TABLE 2 Maximal values for XLM_{ask} (slippage) in the case of a volume of \$1,000,000 together with the reference price and the average buy price. The symbol ⊥ indicates that the volume on the ask side never exceeded \$1,000,000.

	BTC			ETH			XRP		
Exchange	**Slippage**	**Ref. price**	**Avg. price**	**Slippage**	**Ref. price**	**Avg. price**	**Slippage**	**Ref. price**	**Avg. price**
Binance	2.446%	\$43,486.04	\$44,549.58	0.040%	\$1041.73	\$1042.15	0.445%	\$0.296	\$0.297
Bequant	0.074%	\$19,485.48	\$19,499.91	⊥	⊥	⊥	1074%	\$0.419 \$	\$0.424
Bitstamp	1106%	\$39,173.49	\$39,606.80	441,500%	\$1409.70	\$7633.53	1803.047%	\$0.455	\$8657
CEXIO	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
Huobi	0.040%	\$17,255.00	\$17,261.93	0.041%	\$1226.99	\$1227.50	0.094%	\$0.622	\$0.623
Kraken	0.732%	\$28,102.15	\$28,307.95	0.412%	\$1915.93	\$1923.83	⊥	⊥	⊥

TABLE 3 Order book data metrics.

Metric name	Unit	Description
Order book update messages	Updates per second	Total number of order book updates per second received by the data collection services. This value correlates with market activity, though a value of 0 usually indicates an exchange being offline
Latency exchange to data service	ms	Latency from exchange to data service
Latency data service to broker	ms	Latency from data service to broker
Latency broker to bot	ms	Latency from broker to bot
Data processing latency	ms	Internal processing latency of the data collection and harmonization component
Message broker cache size	TB per second	Size of the message log on disk of our central message broker for collected order book data

6 | RESULTS

In the following, we examine the performance of Hephaistos and Athena. To do so, we look at various data collected by Hephaistos that show the difficulties in handling order book data.

6.1 | Performance of Hephaistos

Our system, Hephaistos, serves as the cornerstone of a high-frequency trading engine, with each component intricately linked to the system's stability and monitoring infrastructure, which analyzes, measures, and alerts on specific critical performance metrics, or key performance indicators (KPIs).

Our system stability monitoring is built on *Prometheus* and *Grafana*. Prometheus is an open-source toolkit for system monitoring and alerting, while Grafana is a versatile, open-source analytics and visualization web application. Grafana offers a rich array of charts, graphs, and alerts when connected to supported data sources. Prometheus and Grafana are both tailored for time-series data, with Prometheus primarily handling data collection and Grafana specializing in reporting. Prometheus collects extensive metrics and features a robust querying language, while Grafana transforms these metrics into insightful visualizations.

We classify metrics into two categories: order book data metrics and order execution metrics, as summarized in Tables 3 and 4, respectively. Through these KPIs, we can observe that order book data exemplifies the Big Data phenomenon. For instance, as depicted in Figure 6, order book data are generated rapidly, showcasing the velocity property. In September 2022 alone, the 22 exchanges covered by the system generated 32,043,446,130 order book price updates, which corresponds to 1.7 TB volume stored and processed. Over 3 years, we have amassed order books totaling

TABLE 4 Order execution metrics.

Metric name	Unit	Description
Order placement latency	ms	Time between sending the placement message and the order acceptance timestamp reported by the exchange
Exchange round trip time	ms	Time between sending the placement message and receiving acknowledgment of order acceptance from the exchange
Order execution latency	ms	Time between sending the placement message and receiving the full execution report
Internal engine latency	ms	Time between receiving an order book update from Kafka and sending the order placement message
Internal system latency	ms	Time between receiving an order book update at data service and sending out the resulting (first) order

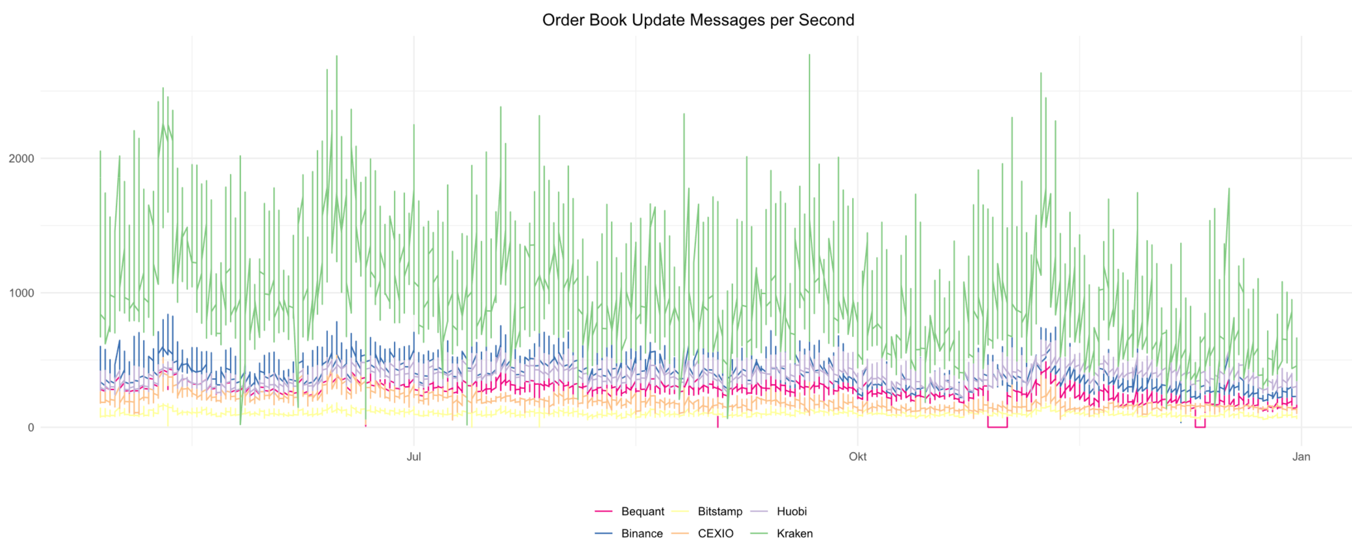

FIGURE 6 Order book update messages per second within 9 months from the six exchanges covered in our experiments.

55 TB in volume. Occasionally, some entries may contain nonsensical information, necessitating detection and filtering (veracity). Moreover, certain exchanges provide supplementary data beyond the basic order book parameters, such as notional volume, entry time, last change time, or order type, highlighting the variety inherent in the data. To extract value from the vast pool of Big Data, our system prioritizes low latency.

6.2 | Performance of Athena

We now want to evaluate the effectiveness of Athena and show to what extent it can reduce implicit costs.

6.2.1 | Impact of the invested volume and number of exchanges

First, we evaluate the impact of the number of exchanges on the implicit costs. We compare the maximum slippage when investing a volume varying between \$10,000 and \$70,000 when two order books (among all possible combinations of two order books from the six available) are combined and when six order books are combined. The results for the three currency pairs are shown in Figure 7.

The construction of the UOB and XLM shows that implicit costs increase as investment volumes increase because higher levels cause higher prices to be incorporated into the average pricing. Both the total implicit costs measured in

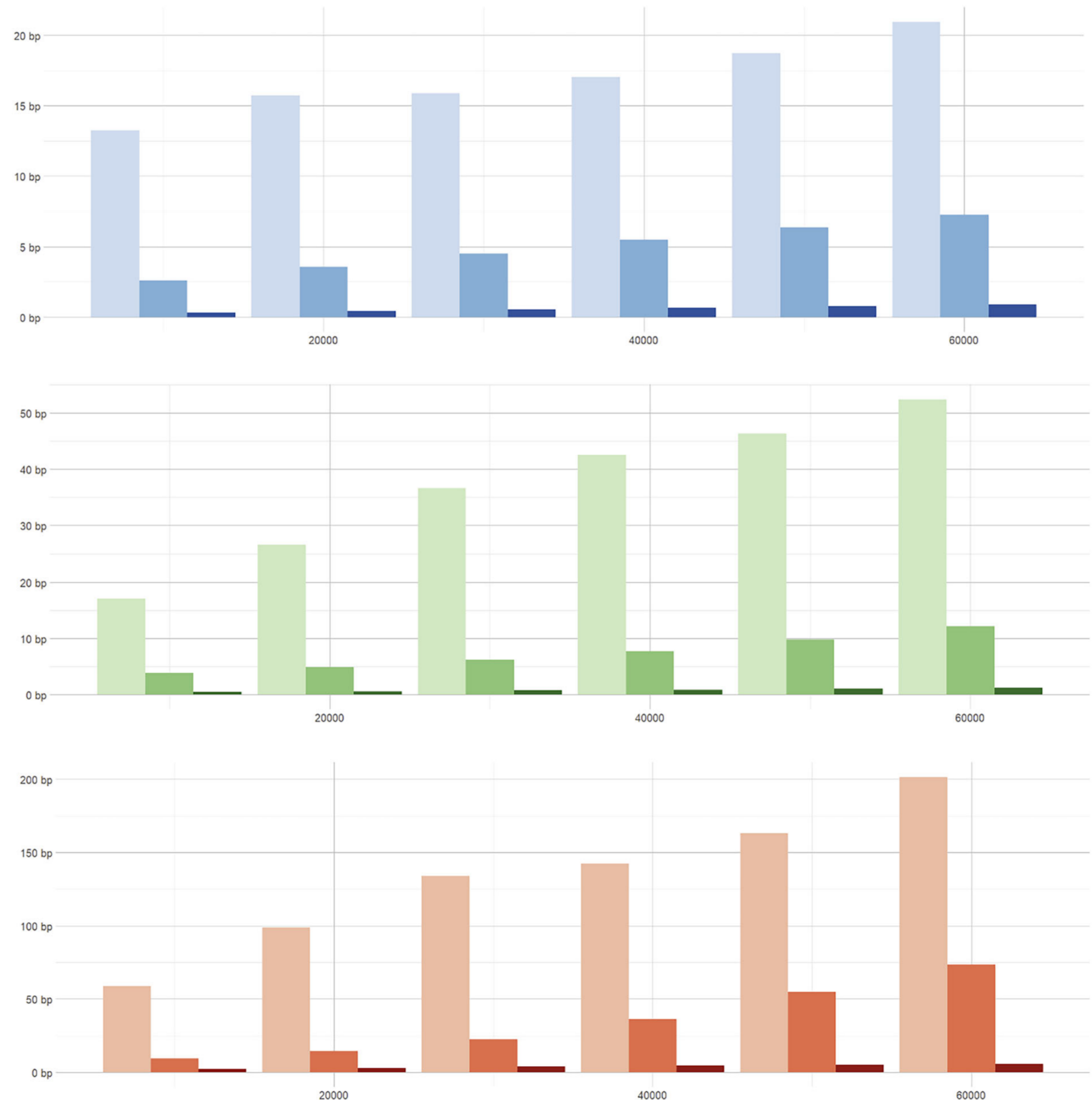

FIGURE 7 Impact of the number of exchanges considered for the UOB. The horizontal axis displays the invested volume in USD, and the vertical axis the XLM measured in basepoints. (Top) Slippage shown for Bitcoin. (Middle) Slippage shown for Ethereum. (Bottom) Slippage shown for Ripple.

basis points, as well as its growth, are highest for XRP, followed by Ethereum. This observation can probably be attributed to the fact that smaller coins, in terms of market capitalization, are more illiquid and thus lead to more significant price deviations.

Regardless of the volume invested, it is evident in all cases that a lower number of exchanges considered leads to higher implicit costs. This also results from the construction of the UOB since prices increase faster with fewer exchanges as levels grow. Already from two exchanges considered, the implicit costs are reduced by more than half.

6.2.2 | Evolution of implicit costs over time

In order to analyze the development of the implicit costs over time, we consider the XLM for an investment volume of $100,000 for each point in time. Figure 8 shows the development of the daily averages in 2022, measured in basis points,

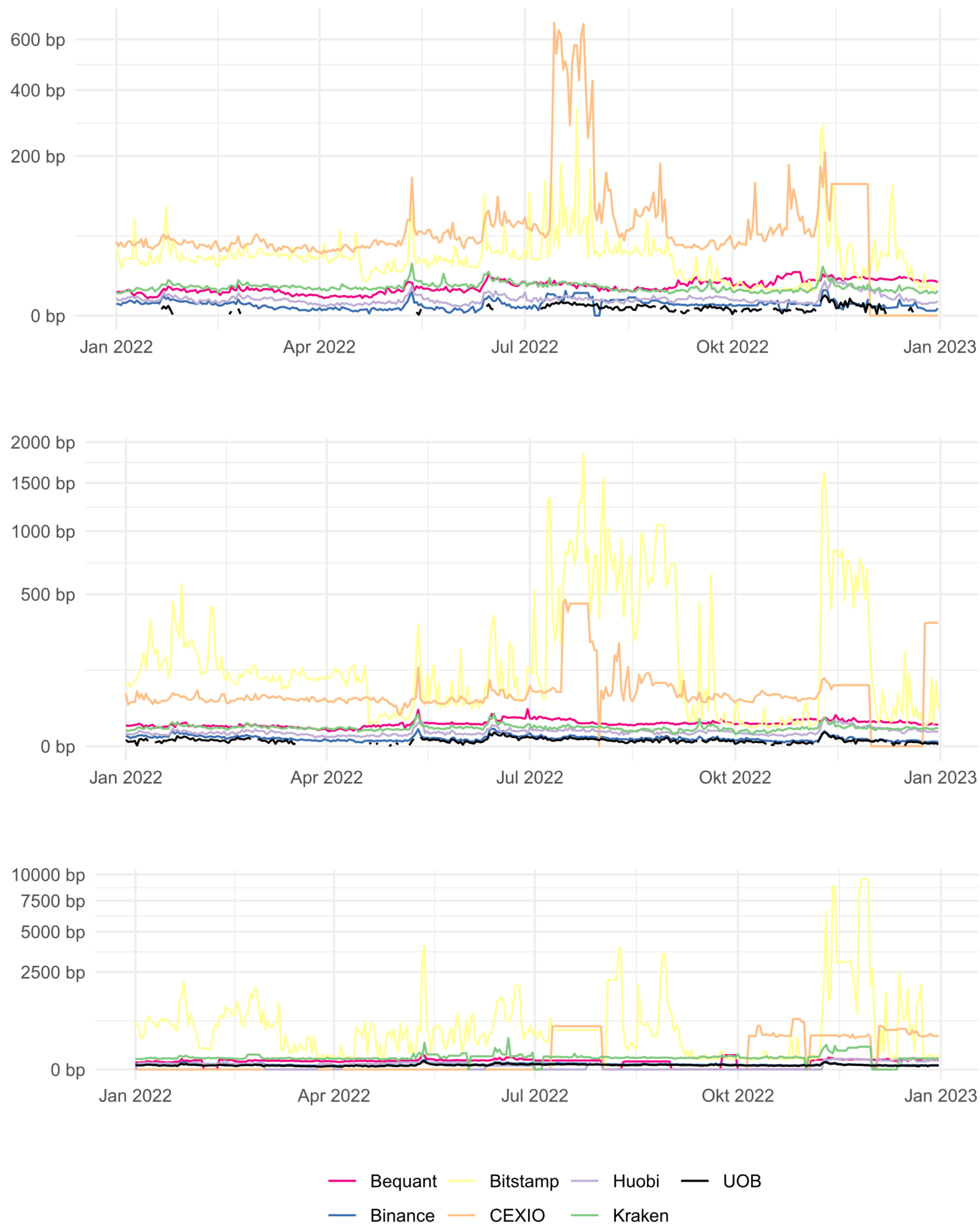

FIGURE 8 Evolution of implicit costs over time for the three currency pairs considered. The vertical axis measures the XLM for an investment volume of \$100,000. Each line represents a separate exchange or the UOB. (Top) Implicit costs for Bitcoin. (Middle) Implicit costs for Ethereum. (Bottom) Implicit costs for Ripple.

for each of the six exchanges considered and the UOB. The UOB (black line) shows some interruptions, which can be attributed to constellations in which the XLM takes on negative values due to an overlapping bid and ask side, which are not shown in the figure.

Substantial deviations in the range of values of implicit costs can be observed across the individual exchanges. Bitstamp and CEXIO tend to have the highest implied costs with strong outliers to the top. We suspect that smaller exchanges lead to higher implied costs. The sharp fluctuations are most evident in the summer of 2022 for Bitcoin and Ethereum and in the winter of 2022 for Ethereum and Ripple. The FTX collapse might justify the development in the fourth quarter. Once again, the construction of the UOB shows that it has the lowest implied cost. Furthermore, the UOB shows hardly any fluctuations and always remains very low.

6.3 | Threats to validity

Our experiments demonstrate a significant enhancement in the implicit costs outlined by the XLM due to the implementation of the SOR. However, our rationale is subject to specific threats to validity. One such threat is the decision to overlook snapshots with insufficient volume in our analysis, resulting in a bias toward market phases characterized by minimal trading activity. It is intuitive to expect higher price distortions during such periods. Nonetheless, the fundamental assertions of our conclusions remain unchanged.

In any empirical study, there exists a risk that the data may not be fully representative, potentially leading to biased outcomes. Our experiment necessitated specifying the exchanges, coins, and time intervals. We encompassed both large and smaller trading platforms for exchanges, while limiting our coin selection to three of the most well-known and relevant currencies. Focusing on Bitcoin, in particular, facilitates comparison with existing research. Moreover, our evaluation relies on a Python implementation, which is susceptible to errors. We endeavored to mitigate this risk through rigorous testing and code reviews.

Another constraint in our considerations pertains to practical circumstances. For instance, in evaluating the SOR, we assume the stability of the UOB between aggregation and order execution. Despite our system facilitating low-latency trading, maintaining the constancy of the UOB is not guaranteed at all times. Additionally, we presume that sufficient funds are consistently deposited at each exchange for order execution. However, achieving this during operations proves challenging, as it necessitates an excessive amount of stablecoins on each exchange, ultimately impacting the overall return of trading strategies.

7 | CONCLUSIONS

In a professional trading environment, the most relevant costs are implicit costs arising from insufficient liquidity. In fragmented and inefficient markets, SOR services are therefore essential for institutional traders. This paper describes Hephaistos, a system that collects order books from several CEX and prepares them to apply a SOR. The SOR aggregates the individual order books into a common UOB, whose levels store the price, volume, and exchange. The price determination based on the UOB results in a split of a single order across multiple exchanges, which reduces implicit costs. In a representative evaluation using six exchanges and three currency pairs, we empirically demonstrated the effectiveness of the SOR by measuring the XLM for different investment volumes. This showed that with each additional exchange, the implied cost is roughly halved, and the cost of trading the UOB fluctuates less and is only in a smaller range of values.

We see different promising directions for future work. We plan to extend our experimental setup to address our current threats to validity, for example, by evaluating more exchanges and currency pairs or different liquidity measures. In addition to looking at spot markets, analysis of other financial markets, for example, futures and options, would also be relevant. Especially due to the FTX crash, spot markets were drained of liquidity, and trade volumes decreased, flowing into crypto derivatives markets. In addition, it would be interesting to see to what extent the implicit costs affect the returns of specific trading strategies and what additional returns can be generated by the SOR.

Our infrastructure is used within a low-latency trading environment. As the analysis of the UOB shows, the bid and ask sides often overlap across several exchanges, generating an opportunity for arbitrage trading. In subsequent work, we plan to present actual results from spot arbitrage trading and evaluate how many cases the theoretical arbitrage constellation could be used. This requires a continuation of the description of our infrastructure with an additional trading component.

CONFLICT OF INTEREST STATEMENT

Robert Henker declares a potential conflict of interest through his position as CEO and shareholder of German Deep Tech Quantum GmbH. The remaining authors declare no potential conflict of interest.

ACKNOWLEDGEMENTS

Open Access funding enabled and organized by Projekt DEAL.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in Order Book Data from Six Crypto-Exchanges at https://doi.org/10.5281/zenodo.10600373.

ORCID

Willy Scheibel https://orcid.org/0000-0002-7885-9857

ENDNOTE

Archive available at DOI: 10.5281/zenodo.10600374.

REFERENCES

1. The Block. DEX to CEX spot trade volume (%). 2023. https://www.theblock.co/data/decentralized-finance/dex-non-custodial/dex-to-cex-spot-trade-volume.
2. Makridis CA, Fröwis M, Sridhar K, Böhme R. The rise of decentralized cryptocurrency exchanges: evaluating the role of airdrops and governance tokens. *Finance*. 2023;79:102358. doi:10.1016/j.jcorpfin.2023.102358
3. Aspris A, Foley S, Svec J, Wang L. Decentralized exchanges: the "wild west" of cryptocurrency trading. *Int Rev Financ Anal*. 2021;77:101845. doi:10.1016/j.irfa.2021.101845
4. Gould MD, Porter MA, Williams S, McDonald M, Fenn DJ, Howison SD. Limit order books. *Quant Finance*. 2013;13(11):1709-1742. doi:10.1080/14697688.2013.803148
5. Jobst A, Atzberger D, Henker R, Vollmer JO, Scheibel W, Döllner J. Examining liquidity of exchanges and assets and the impact of external events in centralized crypto markets: a 2022 study. In: CryptoEx'23. IEEE. IEEE 2023. in press.
6. Gomber P, Schweickert U. The market impact-liquidity measure in electronic securities trading. *Die Bank*. 2002;7(1):485-489.
7. Pujol G, Brueckner A. Smart order routing and best execution. *AMCIS 2009 Proceedings*. 2009:155.
8. Henker R, Atzberger D, Vollmer JO, Scheibel W, Döllner J, Bick M. Hephaistos: a management system for massive order book data from multiple centralized crypto exchanges with an internal unified order book. In: CryptoEx'23. IEEE. IEEE 2023.
9. Jin X, Wah BW, Cheng X, Wang Y. Significance and challenges of big data research. *Big Data Research*. 2015;2(2):59-64. doi:10.1016/j.bdr.2015.01.006
10. Angerer M, Gramlich M, Hanke M. Order book liquidity on crypto exchanges. In: CAL'21. Crypto Asset Lab. Crypto Asset Lab 2021.
11. Fang F, Ventre C, Basios M, et al. Cryptocurrency trading: a comprehensive survey. *Financ Innov*. 2022;8(1):13. doi:10.1186/s40854-021-00321-6
12. Victor F, Weintraud AM. Detecting and quantifying wash trading on decentralized cryptocurrency exchanges. In: WWW'21 ACM ACM 2021:23–32.
13. Bartoletti M, Pes B, Serusi S. Data mining for detecting bitcoin ponzi schemes. In: CVCBT'18. IEEE. IEEE 2018:75–84.
14. McNally S, Roche J, Caton S. Predicting the price of bitcoin using machine learning. In: PDP'18. IEEE. IEEE 2018:339–343.
15. Abraham J, Higdon D, Nelson J, Ibarra J. Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Sci Rev*. 2018;1(3):1.
16. Mohapatra S, Ahmed N, Alencar P. KryptoOracle: a real-time cryptocurrency price prediction platform using Twitter sentiments. In: BigData'19. IEEE. IEEE 2019:5544–5551.
17. Ho KH, Chan TT, Pan H, Li C. Do candlestick patterns work in cryptocurrency trading? 2021:4566–4569.
18. Dilbagi A. Infrastructure and techniques to collect data and detect market manipulation on crypto-exchanges. Master's thesis. Operations Research and Financial Engineering, Princeton University, New Jersey, USA. 2021.
19. Burján V, Gyires-Tóth B. GPU accelerated data preparation for limit order book modeling. In: *Machine Learning, Optimization, and Data Science*. Springer; 2020:385-397.
20. Huang R, Polak T. Lobster: limit order book reconstruction system. *Available at SSRN 1977207*. 2011.
21. Dyhrberg AH, Foley S, Svec J. How investible is bitcoin? Analyzing the liquidity and transaction costs of bitcoin markets. *Econom Lett*. 2018;171:140-143. doi:10.1016/j.econlet.2018.07.032
22. Brauneis A, Mestel R, Riordan R, Theissen E. A high-frequency analysis of bitcoin markets. In: EUROFIDAI - ESSEC. 2019, doi:10.2139/ssrn.3478514.
23. Brauneis A, Mestel R, Theissen E. The crypto world trades at tea time. Intraday evidence from centralized exchanges across the globe. 2023.
24. Huang K. Why did FTX collapse? Here's what to know. 2022. https://www.nytimes.com/2022/11/10/technology/ftx-binance-crypto-explained.html.
25. Krisztian Sandor EG. The fall of terra: a timeline of the meteoric rise and crash of UST and LUNA. 2022. https://www.coindesk.com/learn/the-fall-of-terra-a-timeline-of-the-meteoric-rise-and-crash-of-ust-and-luna/.
26. Chortane SG, Naoui K. The impact of the Covid-19 crisis on the liquidity of cryptocurrencies. In: IEEE 2022:730–734.
27. Brauneis A, Mestel R, Riordan R, Theissen E. Bitcoin unchained: determinants of cryptocurrency exchange liquidity. *J Empir Financ*. 2022;69:106-122. doi:10.1016/j.jempfin.2022.08.004

28. Wang X, Huan TL. BNP paribas: equity smart order router. Electionic Projects Collection. 2010.
29. European Commission. Implementing and delegated acts - MiFID II. 2023. https://finance.ec.europa.eu/regulation-and-supervision/financial-services-legislation/implementing-and-delegated-acts/markets-financial-instruments-directive-ii_en.
30. Delaney A. Smart order routing: the route to liquidity access & best execution. 2009. www.smart-trade.net/wp-content/uploads/2016/12/Smart_Order_Routing_The_Route_to_Liquidity_Access_and_Best_Execution.pdf.
31. Ende B, Gomber P, Lutat M. Smart order routing technology in the new European equity trading landscape. In: *Software Services for e-Business and e-Society*. Springer; 2009:197-209.
32. Jeon Y, Samarbakhsh L, Hewitt K. Fragmentation in the bitcoin market: evidence from multiple coexisting order books. *Financ Res Lett*. 2021;39:101654. doi:10.1016/j.frl.2020.101654
33. Al-Yahyaee KH, Mensi W, Ko HU, Yoon SM, Kang SH. Why cryptocurrency markets are inefficient: the impact of liquidity and volatility. *North am J Econ Finance*. 2020;52:101168. doi:10.1016/j.najef.2020.101168
34. LCX. LCX smart order 2023. https://www.lcx.com/smart-order/.
35. LCX. LCX terminal. 2023. https://terminal.lcx.com.
36. Shrimpy. Crypto API for smart order routing. 2019. https://blog.shrimpy.io/blog/crypto-api-for-smart-order-routing.
37. McNickel D. Crypto bots in trouble - shrimpy calls it quits. 2023. https://bravenewcoin.com/insights/crypto-trading-bots-in-trouble.
38. SwissBorg. Go from fiat to crypto in an instant! 2023. https://swissborg.com/smart-engine.
39. PostgreSQL. Documentation table partitioning. 2023. postgresql.org/docs/12/ddl-partitioning.html.
40. PostgreSQL. Documentation BRIN indexes. 2023. www.postgresql.org/docs/12/brin.html.
41. Aitken M, Comerton-Forde C. How should liquidity be measured? *Pac-Basin Fin J*. 2003;11(1):45-59. doi:10.1016/S0927-538X(02)00093-8

How to cite this article: Henker R, Atzberger D, Vollmer JO, Scheibel W, Döllner J, Bick M. Athena: Smart order routing on centralized crypto exchanges using a unified order book. *Int J Network Mgmt*. 2024;e2266. doi:10.1002/nem.2266

APPENDIX A.

TABLE A1 Example for the first 20 levels for five order books and the resulting UOB.

Ask side												
Binance		Bitstamp		Bequant		Huobi		Kraken		Unified order book		
Price	Volume	Price	Volume	Price	Volume	Price	Volume	Price	Volume	Price	Volume	Exchange
$46,216.93	0.684	$46,240.66	0.372	$46,225.10	0.011	$46,172.92	0.000	$46,205.80	0.281	$46,172.92	0.000	Huobi
$46,216.94	0.299	$46,247.10	0.040	$46,225.11	0.261	$46,178.27	0.000	$46,220.30	0.062	$46,178.27	0.000	Huobi
$46,220.83	0.204	$46,248.59	0.032	$46,225.42	0.011	$46,191.10	0.000	$46,220.40	0.065	$46,191.10	0.000	Huobi
$46,221.64	0.005	$46,262.16	0.540	$46,232.06	0.003	$46,215.97	0.069	$46,220.50	0.100	$46,205.80	0.281	Kraken
$46,223.52	0.111	$46,267.09	0.032	$46,235.40	0.328	$46,215.98	0.005	$46,221.20	0.100	$46,215.97	0.069	Huobi
$46,224.00	0.241	$46,318.04	1.080	$46,237.56	0.250	$46,218.86	0.100	$46,226.10	0.100	$46,215.98	0.005	Huobi
$46,224.01	0.320	$46,361.69	0.202	$46,239.23	0.058	$46,219.00	0.002	$46,235.50	0.300	$46,216.93	0.684	Binance
$46,224.62	0.749	$46,386.72	2.160	$46,239.46	0.018	$46,219.99	0.000	$46,239.40	0.033	$46,216.94	0.299	Binance
$46,224.85	0.088	$46,471.18	0.441	$46,240.37	0.014	$46,220.00	0.002	$46,243.00	0.045	$46,218.86	0.100	Huobi
$46,225.32	0.359	$46,727.56	0.105	$46,241.02	0.002	$46,221.00	0.002	$46,249.40	0.056	$46,219.00	0.002	Huobi
$46,226.12	0.230	$46,727.57	0.002	$46,241.32	0.060	$46,221.64	0.005	$46,252.50	0.300	$46,219.99	0.000	Huobi
$46,226.29	0.215	$46,792.62	0.002	$46,242.38	0.035	$46,221.99	0.004	$46,256.40	0.048	$46,220.00	0.002	Huobi
$46,227.10	0.100	$46,857.76	0.002	$46,242.75	0.288	$46,223.00	0.400	$46,258.00	0.010	$46,220.30	0.062	Kraken
$46,227.97	0.008	$46,923.00	0.002	$46,242.81	0.034	$46,225.32	0.200	$46,258.60	0.023	$46,220.40	0.065	Kraken
$46,229.07	0.577	$46,988.32	0.002	$46,244.62	0.013	$46,225.50	0.747	$46,259.80	0.032	$46,220.50	0.100	Kraken
$46,229.59	0.781	$47,053.74	0.002	$46,244.63	0.228	$46,225.69	0.019	$46,266.00	0.027	$46,220.83	0.204	Binance
$46,229.61	0.781	$47,119.25	0,002	$46,244.73	0.230	$46,226.88	0.393	$46,273.30	0.074	$46,221.00	0.002	Huobi
$46,230.00	8.135	$47,184.85	0.002	$46,244.74	0.679	$46,228.00	0.030	$46,282.50	0.120	$46,221.20	0.100	Kraken
$46,231.11	0.242	$47,232.45	0.047	$46,244.93	0.222	$46,228.10	0.030	$46,285.20	0.024	46,221.64	0.005	Binance
$46,231.56	0.990	$47,250.54	0.002	$46,245.19	0.293	$46,228.60	0.023	$46,295.10	1.971	$46,221.64	0.005	Huobi
Bid side												
Binance		Bitstamp		Bequant		Huobi		Kraken		Unified order book		
Price	Volume	Price	Volume	Price	Volume	Price	Volume	Price	Volume	Price	Volume	Exchange
$46,216.92	0.064	$46,161.75	0.540	$46,213.46	0.139	$46,215.96	0.303	$46,204.30	0.175	$46,216.92	0.064	Binance
$46,214.01	0.056	$46,155.89	0.415	$46,213.45	0.011	$46,211.99	0.005	$46,193.00	0.065	$46,215.96	0.303	Huobi
$46,212.30	0.031	$46,152,39	0.033	$46,213.44	0.011	$46,211.98	0.006	$46,185.90	0.067	$46,214.01	0.056	Binance

TABLE A1 (Continued)

Bid side												
Binance		Bitstamp		Bequant		Huobi		Kraken		Unified order book		
Price	Volume	Price	Volume	Price	Volume	Price	Volume	Price	Volume	Price	Volume	Exchange
$46,212.29	0.087	$46,133,92	0.033	$46,210.49	0.335	$46,207.69	0.038	$46,185.00	0.021	$46,213.46	0.139	Bequant
$46,211.31	0.017	$46,088,99	1.080	$46,208.93	0.250	$46,207.68	0.200	$46,175.20	0.023	$46,213.45	0.011	Bequant
$46,208.36	0.041	$46,081.34	0.220	$46,207.92	0.000	$46,207.58	0.043	$46,175.00	0.033	$46,213.44	0.011	Bequant
$46,208.35	0.040	$45,991.92	0.460	$46,207.01	0.022	$46,205.89	0.002	$46,165.50	0.010	$46,212.30	0.031	Binance
$46,207.96	0.018	$45,990.77	2.160	$46,201.97	0.017	$46,205.88	0.427	$46,165.10	0.300	$46,212.29	0.087	Binance
$46,207.62	0.059	$45,200.01	1.430	$46,201.90	0.024	$46,205.11	0.002	$46,165.00	0.169	$46,211.9	0.005	Huobi
$46,205.80	0.284	$45,200.00	0.018	$46,201.42	0.002	$46,205.10	0.200	$46,163.60	0.056	$46,211.98	0.006	Huobi
$46,205.79	0.134	$45,050.00	0.001	$46,199.66	0.058	$46,205.07	0.100	$46,163.40	0.030	$46,211.31	0.017	Binance
$46,205.78	0.084	$45,015.00	0.020	$46,197.17	0.200	$46,204.76	0.022	$46,157.10	0.023	$46,210.49	0.335	Bequant
$46,205.27	0.235	$45,000.00	0.103	$46,197.16	0.365	$46,204.03	0.520	$46,154.70	0.045	$46,208.93	0.250	Bequant
$46,205.00	0.298	$44,324.00	0.020	$46,192.76	0.014	$46,204.02	0.237	$46,151.30	0.032	$46,208.36	0.041	Binance
$46,203.80	0.302	$44,300.00	0.010	$46,191.80	0.083	$46,200.66	0.005	$46,143.00	0.185	$46,208.35	0.040	Binance
$46,203.79	0.301	$44,000.01	0.001	$46,188.76	0.149	$46,200.49	0.200	$46,134.30	0.220	$46,207.96	0.018	Binance
$46,203.78	0.211	$44,000.00	0.000	46,188.16	0.760	46,200.06	0.032	$46,134.20	0.048	$46,207.92	0.000	Bequant
$46,201.06	0.072	$43,886.31	0.001	$46,187.78	0.093	$46,199.82	0.220	$46,110.10	0.006	$46,207.69	0.038	Huobi
$46,200.20	0.011	$43,500.00	0.010	$46,187.14	0.585	$46,198.93	0.041	$46,107.20	0.001	$46,207.68	0.200	Huobi
$46,200.19	0.084	$43,422.17	0.081	$46,186.82	0.059	$46,198.91	0.005	$46,103.20	0.000	$46,207.62	0.059	Binance