SoK: Unified Blockchain Data Structure

Natkamon Tovanich, Madalina I Sas, Christophe Lebrun, Munthitra Thadthapong, William J Knottenbelt, Arnaud Gaudinat, Marco Mattavelli

SoK: Unified Blockchain Data Structure

Natkamon Tovanich[1]*, Madalina I. Sas[2]*, Christophe Lebrun[3]*, Munthitra Thadthapong[1]
William J. Knottenbelt[2], Arnaud Gaudinat[3], and Marco Mattavelli[4]
[1]CREST, CNRS, École Polytechnique, Institut Polytechnique de Paris, France
[2]Department of Computing, Imperial College London, UK
[3]Haute École de Gestion de Genève / HES-SO, Geneva, Switzerland
[4]École Polytechnique Fédérale de Lausanne, Switzerland
Email: *faironchain@listes.epfl.ch*
* These authors contributed equally to this work.

***Abstract*—Over the last 15 years, blockchain technologies have evolved into a heterogeneous ecosystem with numerous practical applications. The diversity of blockchain data models – ranging from UTxO-based to account-based systems and evolving towards smart contract platforms – has led to fragmentation in how blockchain data is stored, accessed, and analysed. This heterogeneity hinders interoperability, complicates cross-chain analytics, and limits reproducibility in blockchain research. In this paper, we present a comprehensive study of blockchain data unification, covering a wide range of academic and commercial approaches. We examine their underlying data structures and identify key limitations in existing solutions. Building on these insights, we propose a unified data model that abstracts and flattens token transfer data while capturing the semantics of both UTxO and account-based paradigms. We conclude by outlining future research directions to further validate and extend this model across diverse blockchain ecosystems.**

***Index Terms*—public blockchain, blockchain data, distributed ledger technology, standardisation, data structure**

I. Introduction

Since Satoshi Nakamoto published the seminal Bitcoin whitepaper [1], *distributed ledger technologies (DLTs)* have emerged as a disruptive innovation in several fields spanning across finance, logistics, healthcare, cybersecurity, and entertainment. Over the past decade, the original concept has evolved across a broad range of novel implementations, overcoming various initial limitations of Bitcoin, such as privacy protection, inefficiencies of consensus protocols, excessive energy consumption, transaction throughput constraints, and smart contracts support.

The blockchain ecosystem has evolved into a complex taxonomy of DLTs, each defined by specific properties, consensus mechanisms, and user interactions. Moreover, integrating smart contract functionality on selected blockchains has enabled virtually limitless classes of applications, broadening the scope of the initial financial use case for blockchains. These systems now support a diverse portfolio of assets – from native cryptocurrencies to various other tokens – tailored for specific use cases while serving as a fertile ground for research into transaction networks, cryptocurrency economics, and forensic analyses of illicit activities.

Each blockchain generates publicly accessible data encapsulating *transactions*, *addresses*, *metadata*, and complex constructs such as *smart contracts* and *internal transactions*. These *smart contracts* enable *decentralised finance (DeFi)* applications, which extend traditional financial services on decentralised networks. Data from DeFi applications is stored on blockchains in granular detail – publicly available while user identities remain pseudonymized – making it ideal for comprehensive analysis. Beyond this, inter-blockchain operations facilitated by *bridges* add another layer of complexity.

Although available, distributed ledger data is not straightforward to browse and analyse because it lacks consistent, indexed, user-friendly data structures, representations, and associated APIs. Each blockchain tailors its architecture to specific use cases, making data access and interpretation challenging. Whereas emerging commercial entities compete to provide unified blockchain data services, their offers are often prohibitively expensive for most potential users and academic research. The absence of open, standardised, and unified data structures, representations, and APIs forces researchers to waste a considerable amount of their efforts in data collection, indexing, and organisation, rendering blockchain research costly, time-consuming, difficult to replicate, and ultimately inefficient.

This paper explores the challenges of a unified blockchain data structure, reviews the academic and commercial state-of-the-art, and proposes a framework for standardising digital asset or *token* transfers within a unified data structure. We make several contributions to the field of blockchain research:

1) We identify and analyse the primary challenges of identifying unified blockchain data structures, particularly focusing on issues such as data representation, standardisation, modularity, and reusability;
2) We provide a comprehensive review of theoretical and practical tools that address blockchain data standardisation, highlighting their strengths and limitations;
3) We introduce a novel abstract data structure for token transfers designed to bridge the gap between academic or theoretical insights and current commercial practices.

This work was supported by the CHIST-ERA grant *FairOnChain: Fair and modular blockchain data infrastructure for open science and society*, by Agence Nationale de la Recherche (ANR-23-CHRO-0002), by the Engineering and Physical Sciences Research Council (EP/Y036247/1), and by Swiss National Science Foundation (217526).

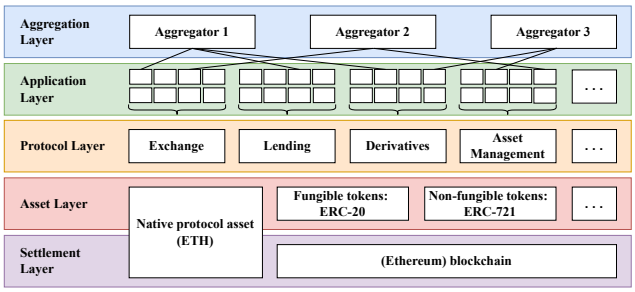

Fig. 1. The Decentralised Finance Stack (reproduced from Schär (2021) [2])

The rest of this paper is structured as follows: Section II discusses the challenges of blockchain data unification. Theoretical and practical tools aimed at blockchain standardisation, including academic and commercial solutions, are reviewed in Section III and Section IV, respectively. Section V introduces a unified abstract data structure for token transfers. To contextualise the contributions in this paper, Section VI presents relevant blockchain data and analysis surveys, as well as limitations of the current endeavour and possible further directions for blockchain data unification. Finally, Section VII presents our conclusions.

II. Challenges to Blockchain Data Unification

In this section, we analyse the challenges associated with harmonising blockchain data models and access by looking at differences in system architecture, issues associated with assets and tokens, interoperability, and data duplication. We conclude with a reflection on the heterogeneity of settlement layers and the evolution of token standards in order to propose a theoretical basis for future standardisation solutions.

Unifying blockchain data models presents significant challenges, both in incorporating the appropriate abstraction elements across various layers of blockchain systems [2] and in accessing data from different blockchain platforms. As illustrated in Figure 1, the foundation lies in the *settlement layer*, responsible for achieving consensus and maintaining the blockchain ledger. Above it lies the *asset layer*, which encompasses both native assets and tokens issued via smart contracts on the underlying blockchain. Building on these is the *protocol layer*, which introduces interoperable standards to compose complex smart contract-based activities. Each layer introduces unique structures and semantics, making consistent modelling across blockchains a non-trivial task. Although the diagram also includes the *Applications layer* and the *Aggregations layer*, these operate off-chain. Accordingly, we have chosen not to focus on these layers in this paper.

Implementing a practical solution requires striking a balance between the availability of blockchain data and the efficiency of storage and querying for specific use cases. This paper focuses on establishing a standardised and unified data framework for token transfers – a targeted approach that underpins a broad array of frequent research use cases. Achieving this objective first requires identifying the abstract processes common to all blockchains – especially at the blockchain and asset layers.

A. Differences in Operationalization and Architecture

One of the primary challenges lies in the variation in data organisation and supported operations across different blockchain architectures. At the heart of this complexity is the accountancy system of the public ledger, which typically falls into two distinct categories: *account-based* and *unspent transaction output (UTxO-based)* blockchains. The key difference is in how transactions are processed: account-based blockchains, such as Ethereum, store user balances in accounts, while UTxO-based blockchains, like Bitcoin, trace unspent transaction outputs, akin to tracking change from cash transactions.

Furthermore, *smart contracts (SCs)* introduce an additional layer of complexity. SCs are self-executing programs deployed on the blockchain that manage and store state variables – such as user balances for a given token – which are updated through internal transactions. They enable the creation of complex protocols – such as DeFi – and digital assets – such as *fungible* ERC-20 tokens and *non-fungible* ERC-721 tokens (NFTs) – to be issued, transferred, or deleted under specified conditions. Unlike native token transfers, which are recorded directly on the ledger, *internal transactions* occur within the smart contract logic. This makes them inherently less transparent and more challenging to track.

As a result, extracting and analysing SC data often requires additional decoding steps to accurately interpret internal state changes, since these operations do not appear as explicit asset transactions on the ledger. Although traditional UTxO-based cryptocurrencies did not support SCs, newer systems such as Cardano [3], [4] have decoupled the core accountancy mechanism of the ledger from its SC functionalities. This separation enables the integration of stateful contract logic within the UTxO framework without altering its fundamental transaction model.

B. Native Assets and Smart Contract Tokens

Initially, most blockchain ledgers were designed to track the *native protocol asset* of each blockchain, such as BTC for Bitcoin or XMR for Monero. These assets are transferred between accounts or transaction outputs through regular transactions. Native assets are generated through the blockchain's validation process (e.g. mining in Bitcoin) and are integral to the blockchain's economic system.

The advent of SC platforms, such as Ethereum, introduced token issuance: where additional digital assets, known as SC tokens, are created and managed by SCs on the blockchain [2]. These platforms facilitate token issuance beyond the capabilities of native assets, with operations such as minting, burning, or freezing providing greater flexibility. However, this increased functionality also complicates the standardisation and analysis of these assets across different blockchain systems. This evolution underlines the importance of studying the asset layer in blockchain systems.

C. Interoperability and Data Duplication

Scaling Solutions (Layer-1, Layer-2, and Sidechains): Layer-1 (L1) refers to the base blockchain – such as Ethereum or Bitcoin – that maintains the original consensus mechanism and ledger. By contrast, Layer-2 (L2) solutions are built on top of L1 protocols to enhance scalability by processing transactions off the main chain while inheriting its security guarantees. Examples include Optimism and Arbitrum in the Ethereum ecosystem. Sidechains, like Polygon, are independent blockchains operating parallel to L1; they interact with the parent chain but follow separate consensus rules. These differing architectures pose challenges for standardising data due to their distinct transaction processing and validation mechanisms.

Cross-Chain Interoperability and Data Duplication: Bridges enable interoperability by facilitating asset and data transfers across otherwise isolated blockchain networks. However, this process can result in the replication of data across distinct chains, contributing to data duplication. In parallel, permanent chain forks introduce additional challenges in ensuring consistent data across split ledgers.

Testnets: Testnets replicate the live blockchain environment for development and testing purposes. Examples include Göerli, Sepolia and Holesky in the Ethereum ecosystem. Although testnets follow the same protocols as their corresponding mainnets, they operate on distinct ledgers, and the assets within them hold no financial value. This separation allows developers to experiment with new features without risking real assets and avoids complications related to data duplication in a production environment.

D. Heterogeneity of Settlement Layers

Even at the settlement layer – where operations and tokens are generally standardised – significant heterogeneity remains a challenge for data unification. While UTxO-based and account-based blockchains represent the two major paradigms, many DLTs deviate from these models or introduce novel features that alter their data structures. Even within ecosystems featuring multiple layers and compatible tokens, like Ethereum, there is a diverse range of blockchain technologies in active use, each with its own settlement protocols. For example, of the 100 tokens with the highest market cap in early 2023, 46 were native tokens used directly by the community (i.e. associated with L1 chains) [5].

Many other L1 blockchains exhibit completely novel operations and lack compatibility with Bitcoin or Ethereum. For example, NANO [6] employs a unique structure in which each account maintains its own chain, while Solana [7] processes transactions containing numerous instructions that can be signed by multiple users. Furthermore, Cardano [8], [9], which uses a proof-of-stake settlement mechanism, extends the UTxO model to support SCs through its extended UTxO (eUTxO) model [3], [4].

Despite these differences, most settlement-layer operations fundamentally involve transferring and recording information on a public ledger. Conceptualizing blockchain operations as flows – the movement of data or digital assets between source and destination – provides a useful abstraction that enables comparison across these heterogeneous systems.

E. Token Standards and Decentralised Applications (dApps)

Ethereum tokens adhere to well-defined tokenization standards that ensure consistency and interoperability. *Fungible tokens* follow the ERC-20 standard [10], while *non-fungible tokens (NFTs)*, which are unique and traceable, are governed by the ERC-721 standard [11]. In addition, ERC-1155 [12] facilitates the management of multiple token types within a single contract. These standards define common data fields and functions, promoting both intra-standard consistency (e.g. within ERC-20 tokens) and cross-standard interoperability; for instance, ERC-20 can interoperate with BRC-20 [13], a standard that enables the issuance of non-fungible tokens on the Bitcoin blockchain, thereby expanding cross-chain interoperability [14].

Building on these token standards, a wide range of *decentralised applications (dApps)* and *decentralised finance (DeFi)* protocols have emerged. These systems leverage SCs to facilitate peer-to-peer transactions with enhanced transparency and security. DeFi platforms – such as decentralised exchanges (DEXs), lending protocols, and yield farming platforms – have rapidly gained popularity by offering users access to financial services without relying on traditional banks. The programmability of SCs also enables the creation of customisable financial products, including synthetic assets, derivatives, staking (and related operations), and other innovative financial instruments, further expanding the possibilities of decentralised ecosystems.

Moreover, governance tokens have become a fundamental element of decentralised autonomous organisations (DAOs), granting token holders voting rights on key decisions – ranging from protocol upgrades and fee structures to governance proposals. By allowing stakeholders to participate directly in decision making, governance tokens promote a more transparent, inclusive, and democratic system, ensuring that control remains decentralised and within the community rather than centralised in a single authority.

The growing adoption of DLTs has resulted in a fragmented ecosystem, where diverse architectures and data structures pose significant challenges to interoperability, analysis, and unified data processing. In the following sections, we examine key theoretical and practical approaches aimed at bridging the gaps between these disparate models.

III. Theoretical and academic solutions

Theoretical and academic solutions to blockchain unification focus on developing abstract models, formal methods, and conceptual frameworks that can bridge the gaps between different blockchain architectures. Table I summarises theoretical and academic solutions, detailing their focus, covered blockchains, key contributions, and limitations.

TABLE I
COMPARISON OF THEORETICAL AND ACADEMIC SOLUTIONS

Paper	Focus	Blockchain Covered	Key Contribution	Limitations
Unified Data Structure				
A Unified Data Model for Blockchains [15]	Schema-matching algorithm for blockchain data unification	Ethereum (account-based), NANO (block-lattice)	Identifies semantically similar fields and maps them for cross-chain integration	Limited to Ethereum and NANO, does not generalize to all blockchain types
Chimeric Ledgers [16]	Unified ledger model bridging UTxO and account-based systems	UTxO (Bitcoin-like), Account-based (Ethereum-like)	Defines abstract data types and conversion functions, leading to the Extended UTxO model in Cardano	Focuses on theoretical abstraction; lacks real-world implementation for multiple blockchains
Blockchain Networks [17]	Network representations of different blockchain transaction models	UTxO (Bitcoin, Litecoin, Monero), Account-based (Ethereum, Ripple), DAG-based (IOTA)	Categorises blockchain transaction models and highlights challenges in interoperability	Does not propose a unified model or solution for cross-chain interactions
Data Processing & Analytics Framework				
A General Framework for Blockchain Analytics [18]	General-purpose blockchain analytics	Bitcoin, Ethereum	Provides a flexible, SQL/NoSQL-based framework for structured blockchain analysis	No unified support for both blockchain models; lacks real-time network analytics
BlockSci [19]	High-performance blockchain analysis platform	UTxO-based (Bitcoin, Litecoin, Dash, Namecoin)	Efficiently parses and analyses blockchain transaction graphs with address clustering	Does not support Ethereum or Monero due to different transaction paradigms
GraphSense [20]	Cryptoasset analytics for transaction tracking	UTxO-based (Bitcoin, Litecoin, Bitcoin Cash, Zcash)	Provides a modular, extensible framework with interactive dashboards and API access	Limited to UTxO blockchains; lacks support for Ethereum and real-time updates

A. Unified Data Structure Proposals

A Unified Data Model for Blockchains: Meyer et al. (2022) [15] propose a clustering-based schema-matching algorithm to unify the data models of multiple blockchains. This method identifies semantically similar fields in different blockchain structures and maps them together. The study focuses on Ethereum and NANO. Ethereum follows an account-based model, where a single state ledger records account balances. In contrast, NANO operates on a block-lattice structure, where each account has its blockchain, and each transaction is represented as a single block. Despite these differences, the study finds that both blockchains share fundamental concepts, such as the movement of tokens between accounts, which enables meaningful cross-chain data mapping despite their architectural differences.

Chimeric Ledgers: Zahnentferner (2018) [16] proposes a unified ledger model that bridges the gap between UTxO-based and account-based blockchain systems. Based on the UTxO model introduced by Atzei et al. (2017) [21], this approach defines abstract data types for both accounting styles and provides conversion functions between them.

For UTxO-based blockchains, transactions reference unspent outputs from previous transactions, and the ledger is modelled as a list of transactions. Balance computation requires traversal of the entire ledger. The abstract data types for the UTxO-based model are:

$$
\begin{aligned}
\mathsf{Ledger} &:= \mathsf{List}[\mathsf{Transaction}] \\
\mathsf{UTxOTx} &:= (\mathsf{inputs} : \mathsf{Set}[\mathsf{Input}], \mathsf{outputs} : \mathsf{Set}[\mathsf{Output}], \\
&\quad\ \mathsf{forge} : \mathsf{Value}, \mathsf{fee} : \mathsf{Value}) \\
\mathsf{Output} &:= (\mathsf{address} : \mathsf{Address}, \mathsf{value} : \mathsf{Value}) \\
\mathsf{Input} &:= (\mathsf{tx} : \mathsf{Id}, \mathsf{index} : \mathsf{Int})
\end{aligned}
$$

where address represents the recipient, tx refers to the transaction identifier, and index denotes the specific output being referenced.

For account-based blockchains, transactions directly modify account balances. A nonce field is required to prevent replay attacks, and transactions explicitly reference sender and receiver addresses. The abstract data type is defined as:

$$
\begin{aligned}
\mathsf{AccTx} := \ & (\mathsf{sender} : \mathsf{Option}[\mathsf{Address}], \\
& \ \mathsf{receiver} : \mathsf{Option}[\mathsf{Address}], \mathsf{value} : \mathsf{Value}, \\
& \ \mathsf{forge} : \mathsf{Value}, \mathsf{fee} : \mathsf{Value}, \mathsf{nonce} : \mathsf{Int})
\end{aligned}
$$

The proposed translation mechanism defines an equivalence relation between UTxO-based and account-based transactions, considering two transactions equivalent if they share the same forged amount, fees, and net balance effect. To facilitate the conversion, the author introduces the HybridTx type, which represents transactions by mapping addresses to values rather than treating them as distinct entities.

$$
\begin{aligned}
\mathsf{HybridTx} := \ & (\mathsf{inputs} : \mathsf{Map}[\mathsf{Address}, \mathsf{Value}], \\
& \ \mathsf{outputs} : \mathsf{Map}[\mathsf{Address}, \mathsf{Value}], \\
& \ \mathsf{forge} : \mathsf{Value}, \mathsf{fee} : \mathsf{Value}, \mathsf{nonce} : \mathsf{Int}) \\
\mathsf{DepTx} := \ & (\mathsf{inputs} : \mathsf{Set}[\mathsf{Input}], \mathsf{depositor} : \mathsf{Option}[\mathsf{Address}], \\
& \ \mathsf{forge} : \mathsf{Value}, \mathsf{fee} : \mathsf{Value}) \\
\mathsf{WithTx} := \ & (\mathsf{withdrawer} : \mathsf{Address}, \mathsf{outputs} : \mathsf{List}[\mathsf{Output}], \\
& \ \mathsf{forge} : \mathsf{Value}, \mathsf{fee} : \mathsf{Value}, \mathsf{nonce} : \mathsf{Int})
\end{aligned}
$$

This approach bridges UTxO and account-based models by decomposing transactions into deposit and withdrawal operations, effectively linking balances across both frameworks.

Building on this work, the Extended UTxO model (EUTxO) has been formalised, incorporating these concepts into the UTxO model with smart contracts developed by the Cardano

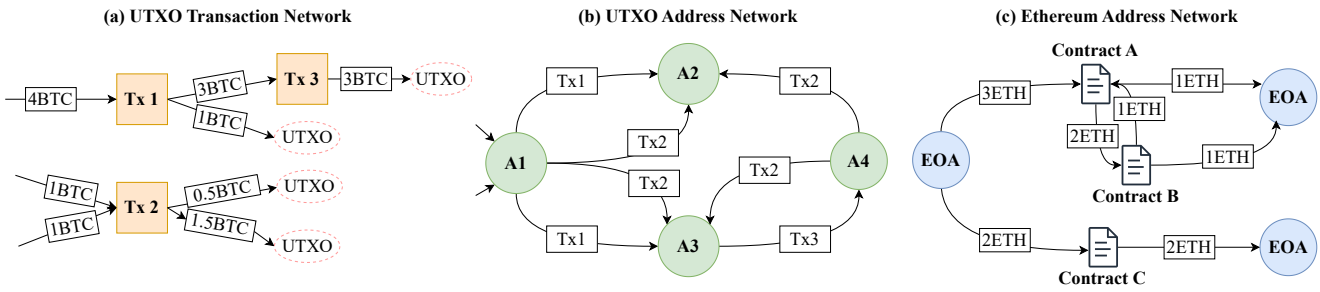

Fig. 2. Three Main Blockchain Network Models (reproduced from Wu et al. (2021) [22])

research team [3]. An implementation in the functional programming language Agda is available in [23].

Blockchain Networks: Akcora, Gel, and Kantarcioglu (2022) [17] review the network representations of transaction data in six major blockchains. The authors categorise network models based on the transaction models. Figure 2 illustrates the three primary blockchain network models identified in their study, reproduced from Wu et al. (2021) [22].

1) *UTxO-based Blockchains* (e.g. Bitcoin, Litecoin, Monero, Zcash), where transactions consume unspent outputs from previous transactions and generate one or more outputs. This structure forms a transaction graph, where nodes represent transactions and edges connect inputs to outputs. Two network representations can be derived:
   - The *Transaction network* omits addresses, connecting transactions directly, useful for *anti-money laundering (AML)* analysis (Figure 2 (a)).
   - The *Address network*, the most commonly used graph model for UTxO networks, creates edges between addresses (Figure 2 (b)).

   To infer ownership, multiple addresses can be clustered based on heuristics such as multi-input and change address detection [24]–[26]. Privacy-focused blockchains, such as Monero and Zcash, introduce cryptographic enhancements to obscure transaction details.
2) *Account-based Blockchains* (e.g. Ethereum, Ripple) modify account balances rather than generate new outputs, simplifying transaction networks. Ethereum supports SCs and consists of three node types: *externally owned accounts (EOAs)*, *smart contract accounts (CAs)*, and the `NULL` address. Based on these, three types of networks can be constructed:
   - The *Transaction network* captures native currency (Ether) transfers between accounts, similar to the UTxO-based Address network. Edges represent the movement of native tokens (Figure 2 (c)).
   - The *Token transaction network* represents token trade through internal SC transactions (e.g. ERC-20 and NFTs).
   - The *Trace network* represents a sequence of contract calls, capturing interactions between CAs and EOAs within the transaction.

   Ripple, a permissioned blockchain, uses a credit network where trust lines indicate that the source address trusts the target address.
3) *DAG-based Blockchains* (e.g. IOTA) introduce a Directed Acyclic Graph (DAG) structure known as the Tangle rather than a linear chain to improve scalability. Two models can be derived:
   - The *Tangle network* is a DAG where nodes represent transactions, and edges indicate approvals of previous transactions.
   - The *Transaction network* represents the money flow network adopted from the UTxO transaction model.

While these models provide efficient mechanisms for transaction handling, they lack a unified framework for cross-model interoperability. Challenges persist in integrating Layer-2 scaling solutions and enabling seamless cross-chain transactions.

B. Data Processing and Analytics Frameworks

A General Framework for Blockchain Analytics: Bartoletti et al. (2017) [18] present a general-purpose blockchain analytics framework that supports both Bitcoin and Ethereum, implemented as an open-source Scala library. The framework constructs data views stored in SQL or NoSQL databases and provides modular APIs representing blockchain entities, such as blocks, transactions, and metadata. It also integrates with external data sources, such as cryptocurrency exchange rates and address tags. The authors evaluate the framework through various use cases, including protocol-specific analysis, transaction fee dynamics, and address tagging.

While the framework offers greater flexibility than other blockchain parsers, supporting multiple databases and external data integration, it lacks capabilities for analysing network-level events (e.g. information propagation, forks, and attacks), which require a full node. The framework is scalable and adaptable for blockchain data analysis. However, it does not unify the data structures of Bitcoin and Ethereum, meaning each blockchain requires accessing separate data sources.

BlockSci: A Blockchain Analysis Platform: BlockSci [19] is an open-source blockchain analytics platform written in C++ that parses raw blockchain data into transaction graphs, indexes, and scripts for querying and visualisation through Jupyter notebooks. It supports UTxO blockchains, including

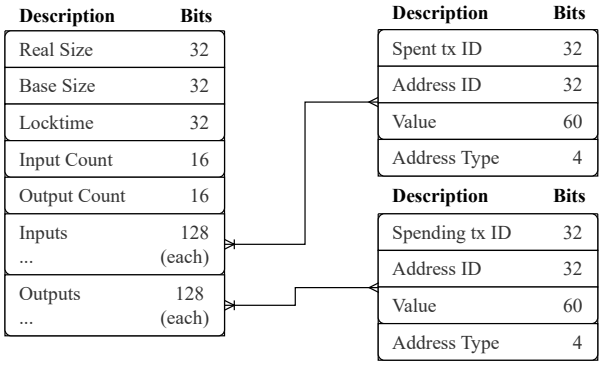

Fig. 3. The Blocksci Transaction Structure (reproduced from Kalodner et al. (2020) [19])

Bitcoin, Bitcoin Cash, Litecoin, Namecoin, and Dash. It does not support Monero (due to its non-DAG transaction structure), Ethereum (due to its complex scripting), or Zcash (due to the use of zero-knowledge proofs that obfuscate transaction details).

BlockSci's architecture enables fast analysis by loading transaction graphs into memory and using a custom parser to extract transaction data into a unified intermediate format. It constructs a transaction graph while exposing only the longest chain, omitting orphaned and stale blocks. Many analytical operations use *MapReduce* to compute transaction values and fee distributions. For address clustering, it employs heuristics such as multi-input and change address detection [27], except in cases of CoinJoin transactions (a mixing service) [28].

Figure 3 displays the entity-relationship (ER) diagram of the BlockSci transaction structure. Each transaction entry in BlockSci's core transaction graph consists of a fixed-size header followed by variable-length input and output arrays. The header includes metadata fields such as the *real size* (32 bits), *base size* (32 bits), *locktime* (32 bits), *input count* (16 bits), and *output count* (16 bits). Each input occupies 128 bits and includes the *spent transaction ID* (32 bits), *address ID* (32 bits), *value* (60 bits), and *address type* (4 bits). Similarly, each output also occupies 128 bits and stores the *spending transaction ID* (32 bits), *address ID* (32 bits), *value* (60 bits), and *address type* (4 bits). This compact and uniform structure allows for memory-efficient storage and fast sequential iteration, which is crucial for analyses that benefit from the spatial locality of reference.

While BlockSci offers a structured approach to blockchain analytics, it is limited to UTxO-based blockchains and lacks support for account-based models like Ethereum. Additionally, the absence of built-in address labelling or tagging restricts its ability to link addresses to real-world entities, making it less suitable for certain kinds of analysis.

GraphSense: A General-Purpose Cryptoasset Analytics Platform: GraphSense [20] is an open-source cryptocurrency analytics platform designed for in-depth investigation of money flows and customised analytics. It is built as a modular and extensible analytics pipeline, managed using Docker and Docker Compose, and consists of the following key components:

- *Data Aggregation:* Collects data from UTxO-model blockchains (e.g. Bitcoin, Bitcoin Cash) using BlockSci [19], as well as external sources like TagPacks (attribution tags) [29] and cryptocurrency exchange rates.
- *Data Transformation and Storage:* Performs the following tasks: (1) Ingests blockchain and external data into Apache Cassandra, a NoSQL backend, for efficient processing; (2) Transforms data to compute transaction statistics and cluster addresses; and (3) Generate structured address and entity graphs.
- *User Interfaces:* GraphSense offers a REST API and Apache Spark integration, along with a visual dashboard for interactive data exploration and analysis.

GraphSense offers higher-level graph abstractions and collaborative address tagging (TagPacks) compared to other blockchain analytics tools. However, its key limitations include support only for UTxO-based blockchains and the lack of real-time transaction updates, as updating address clusters and recomputing graph abstractions require significant computational resources.

C. Discussion

Theoretical and academic studies mainly tackle the challenges of unifying blockchain data structures and outline analytical frameworks. On the data unification front, prior works such as Meyer et al. [15] and Zahnentferner [16] propose schema-matching algorithms and abstract transaction models to reconcile architectural differences between blockchains – particularly between UTxO and account-based paradigms. These approaches lay important theoretical groundwork but remain limited in scope, either focusing on a narrow set of blockchains or lacking real-world deployment.

Meanwhile, analytics frameworks like BlockSci [19], GraphSense [20], and Bartoletti et al. [30] provide pipelines for blockchain data processing, primarily targeting UTxO systems. However, they did not propose a unified analytics framework across transaction models. With the partial exception of BlockSci, these efforts do not demonstrate scalable deployment in practice. While these works highlight foundational progress, a scalable, practical, and cross-model solution for unified blockchain data analysis remains needed.

IV. Commercial Platforms and Practical Tools

The increasing demand for blockchain data access has led to the development of various practical tools, some commercial and some open-source, aiming to offer structured, scalable, and user-friendly access to blockchain data, eliminating the need to operate full nodes. While some services focus on raw data retrieval, others offer analytics, indexing, and visualisation capabilities. Table II compares these data tools based on data availability (raw, decoded, off-chain), query endpoints, and backend databases.

TABLE II
COMPARISON OF BLOCKCHAIN DATA PLATFORMS

		Raw Data			Decoded Data						Off-Chain Data			
Tool	**# Chains**	**UTxO**	**EVM**	**Layer 2**	**Event Logs**	**ERC-20 Transfers**	**NFT Trades**	**DEX Trades**	**Bridges**	**Protocol-specific**	**Token Price**	**Entity Tagging**	**Query Endpoint**	**Backend DB**
Node-as-a-Service														
Infura	19		✓	✓									JSON-RPC	Unknown
QuickNode	60	✓	✓	✓		✓	✓	C			C		JSON-RPC	Unknown
Alchemy	38		✓	✓		C	C						JSON-RPC	Unknown
Moralis	18		✓	✓	C	C	C	C		C	C	✓	JSON-RPC	Unknown
Data Processing														
Blockchain ETL	18	✓	✓	✓		✓	✓						SQL	Google BigQuery
Cryo	6		✓	✓		C	C						Command Line, Python library	Raw table files
Data Provider														
Blockchair	17	✓	✓			✓							JSON-RPC	Unknown
BitQuery	10		✓	✓		C	C	C					GraphQL	Unknown
GoldRush	75	✓	✓	✓	C	C	C				C		JSON-RPC	Unknown
Transpose	12	✓	✓	✓		✓	✓	✓	✓	✓	✓		JSON-RPC, SQL	Unknown
The Graph (and Messari Subgraphs)	71	✓	✓	✓		C	C	C	C	C			GraphQL	Firehose
Data Query & Analytics Tool														
Blockscout	47		✓	✓		✓	✓					✓	JSON-RPC, GraphQL	PostgreSQL
Dune Analytics	47	✓	✓	✓	✓	C	C	C	C	✓	C	✓	SQL, JSON-RPC	DuneSQL
Flipside Crypto	30	✓	✓	✓	✓	✓	✓	C	C	C	C	✓	SQL, JSON-RPC	dbt, Snowflake
Footprint Analytics	31		✓	✓		✓	C	C	C	✓	✓	✓	SQL, JSON-RPC	SQL
Allium	62	✓	✓	✓	✓	✓	C	C	C	C	✓		SQL, JSON-RPC	PostgreSQL, Snowflake, DataBricks, Bigquery, S3, GCS

Note:

- ✓: The tool/platform provides an endpoint or access to this data.
- C: The tool/platform provides a unified endpoint for multiple blockchains, with documentation demonstrating unification (e.g. using the same endpoint or offering a single documentation for multiple chains).

A. Node-as-a-Service

Infura [31], in collaboration with MetaMask, provides scalable blockchain infrastructure, allowing developers to interact with various blockchains through JSON-RPC APIs without the need to run their own nodes. It supports Ethereum, Polygon, Optimism, Avalanche, and IPFS, featuring multi-chain access, archive data querying, and a decentralised infrastructure network (DIN) for reliability.

QuickNode [32] provides blockchain infrastructure with JSON-RPC APIs for accessing 60+ blockchain nodes. It also provides additional APIs for enhanced functionality, including streams (for real-time data), rollups (for deploying Layer 2 or 3), IPFS storage, and Token/NFT APIs.

Alchemy [33] provides Chain APIs (for blockchain node access, similar to Infura and QuickNode) and enhanced APIs for NFT tracking, token analytics, and price data. The *Transfers API* is its key feature, consolidating token transfer records for native tokens, ERC-20, ERC-721 (NFT), and ERC-1155, simplifying wallet and on-chain activity tracking.

Moralis [34] is a Web3 development platform that offers its own APIs for accessing real-time blockchain data, user authentication, and multi-chain support. Moralis includes a *DEX Trades API*, which enables developers to track decentralised exchange transactions, and a *DeFi API*, which allows users to view their DeFi positions per wallet, including liquidity pools, staking, and yield farming positions across various DeFi protocols. Moralis also supports IPFS storage via its *File API*.

B. Data Processing

Blockchain ETL [35] is an open-source Extract, Transform, Load (ETL) framework for processing and analysing blockchain data. It supports UTxO chains (Bitcoin, Litecoin, Zcash), EVM-compatible blockchains (Ethereum, Polygon, Avalanche, Arbitrum), and non-EVM chains (Solana, Polkadot, Tezos).

While public datasets are accessible via Google Cloud BigQuery, the open-source ETL framework allows users to run data extraction on their own nodes. The data schema varies based on blockchain type:

- *UTxO Blockchains* use a standardised schema with tables for blocks, transactions, inputs, and outputs. However, significant data redundancy exists, as the transactions table contains the most relevant information, including block metadata and input/output details.

- *EVM-Compatible Blockchains* follow a five-table schema, i.e. blocks, transactions, logs, receipts, decoded_events, to capture contract interactions and transaction outcomes. Ethereum datasets also include additional tables for account state, token transfers, and traces, which capture internal contract execution.

These datasets provide a structured and comprehensive resource for blockchain researchers and analysts, enabling the processing and exploration of large-scale data. However, they do not inherently unify UTxO and account-based models, as the ETL framework and datasets are stored separately.

Cryo [36] is an open-source blockchain data extraction tool compatible with Ethereum and EVM chains, such as Optimism, Polygon, and Avalanche. It offers command-line and Python integration, allowing users to extract, filter, and format blockchain data into tabular formats, such as Parquet, CSV, JSON, or Python DataFrames. Cryo supports data filtering based on block ranges, contracts, transactions, logs, traces, and state changes.

C. Data Providers

Blockchair [37] provides aggregated statistics and raw data across 17 blockchains, including Bitcoin-like, Ethereum, and other blockchain types (Ripple, Stellar, Monero, Cardano, Mixin). It supports JSON-RPC API requests, SQL-like queries for filtering and aggregating data, and daily data dumps for large-scale analysis.

GoldRush [38] (formerly Covalent's Unified API) offers structured multi-chain data APIs for wallets, NFTs, and transaction tracking over 75 blockchains. Its APIs include:

- The *Wallet API* to trace token balances, transfers, prices;
- The *Transactions API* for historical transaction data with readable log events;
- The *NFT API* for metadata and trading history;
- The *Cross-Chain API* to fetch cross-chain wallet activity and transactions;
- The *Security API* for token allowances and potential risk;
- The *Blockchain API* to access raw blockchain data, including token prices.

Transpose [39] provides a structured blockchain data API with an integrated SQL Analytics API. For each blockchain supported, it organises data based on four hierarchical layers:

- The *Settlement Layer* captures raw on-chain data such as blocks, transactions, logs, and traces;
- The *Asset Layer* tracks token balances, ownership, transfers, and NFT metadata;
- The *Protocol Layer* aggregates data from on-chain protocols including DEXs, lending markets, liquidity pools, and bridges;
- The *Prices Layer* provides token pricing data, including real-time values and historical price series (e.g. Open, High, Low, Close (OHLC) price analysis over specific time periods).

The Graph [40] is a decentralised protocol designed to efficiently index and query blockchain data. It introduces *subgraphs*, structured APIs that allow developers to index specific SCs and events for specific DeFi protocols. The *Graph Node*, a core component of the system, continuously scans blockchain transactions, extracts relevant event data, and updates indexed records. Users can query indexed subgraphs via GraphQL, enabling decentralised applications to retrieve structured blockchain data efficiently and effectively.

Since June 13, 2024, The Graph has fully migrated to a decentralised network of Graph Nodes, creating a permissionless and incentive-driven ecosystem [41]. It supports both account-based models (e.g. Ethereum) and UTxO-based models (e.g. Bitcoin), making it a versatile solution for standardising blockchain data structures across different ledger models.

Messari Subgraphs [42]: Messari, a crypto market intelligence company, is a leading subgraph developer within The Graph ecosystem. It offers a unified method for indexing DeFi protocols, such as decentralised exchanges (DEXs) and lending platforms, using consistent data schemas. Their data structures include a generic schema for network-wide use, along with specialised schemas for decentralised finance applications, including tokens, bridges, DEXs, NFT marketplaces, lending platforms, and derivatives.

Compared to other blockchain data solutions, Messari subgraphs provide a more structured and category-specific indexing approach, ensuring interoperability and efficient data retrieval for DeFi applications.

D. Data Query and Analytics Platforms

BlockScout [43] is an open-source, customisable blockchain explorer for EVM-compatible chains. It supports real-time transaction tracking, contract verification, and enriched on-chain interaction analysis. Unlike closed-source explorers like Etherscan, BlockScout offers full transparency, customisation, and adaptability, making it ideal for developers and blockchain researchers.

The platform provides a comprehensive database that includes [44]:

- *Raw Data:* Blocks, transactions, event logs, and internal contract calls.
- *Aggregated Data:* Network statistics, including transaction counts, gas fees, and market history.
- *Decoded Data:* ERC-20 and ERC-721 token transfers, contract metadata, and verified contract source code.

BlockScout offers both GraphQL and JSON-RPC APIs for querying data, as well as the feature to export blockchain data to CSV for further analysis. While it lacks advanced clustering and forensic analytics, BlockScout excels in real-time blockchain monitoring, contract execution, and multi-chain compatibility. However, its design remains focused on EVM-compatible blockchains, limiting its ability to unify UTxO and account-based models.

Dune Analytics [45] is a SQL-based blockchain analytics platform that allows users to query, visualise, and analyse blockchain data. Dune features SQL-based queries, a user-friendly interface for querying and visualising data, and customisable dashboards. It also allows users to share, fork, and

build upon public queries and dashboards. Dune organises data into four categories:

- *Raw Data:* Unprocessed blockchain data, including transactions, blocks, logs, and traces, directly extracted from blockchain nodes.
- *Decoded Data:* Translated SC interactions using Application Binary Interfaces (ABIs), converting event logs and function calls into structured, human-readable tables.
- *Curated Data:* Pre-processed datasets, such as DEX, NFT trades, and address tags, are categorised for specific use cases across various blockchains.
- *Community Data:* Protocol-based datasets contributed by the community through Spellbook [46], including governance, DeFi protocols, and other blockchain applications.

Dune utilises its own Dune SQL engine for fast and distributed data retrieval, replacing PostgreSQL and Spark SQL in earlier versions.

Flipside Crypto [47] is a blockchain analytics platform that transforms raw on-chain data into structured, human-readable formats across multiple blockchains. It enhances analysis with curated labels, including wallet tags, token prices, and DeFi protocol identifiers. Flipside models data using a Star schema, featuring a central fact table linked to dimension tables containing additional data, which makes querying more efficient at the cost of data redundancy. It categorises tables into three types:

- *Fact tables* store raw blockchain events (transactions, transfers, logs) and enable the summarisation of on-chain activity;
- *Dimension tables* store metadata and context (token names, DEX platforms, addresses) useful for filtering and grouping;
- *Curated tables* (called "ez") combine fact and dimension data and are optimised for easy querying and aggregation.

Flipside uses *dbt* to process and transform low-level tables, making data accessible via *Snowflake SQL* through *Data Studio*, a web-based SQL query interface for analysts, and *LiveQuery*, an API for real-time data access and application integration.

Footprint Analytics [48] is a blockchain data analytics platform that aggregates, models, and visualises on-chain data. It enables users to analyse DeFi protocols, NFT markets, cross-chain bridges, and GameFi ecosystems through structured datasets and interactive dashboards. Footprint Analytics processes raw blockchain data through the following structured pipeline:

- *Bronze:* Ingests, normalises, and cleans raw on-chain data into basic data structures (e.g. blocks, transactions, event logs).
- *Silver:* Enriches data with contextual labels (e.g. cross-chain, DeFi, GameFi, and NFTs) and links SCs to known entities.
- *Gold:* Aggregates data into cross-chain metrics (e.g. total value locked, trading volumes, user behaviour, and market caps) for business-level insights.

Footprint offers two data output paths: *REST APIs* for developers to access structured data via SQL and the *Footprint Web App*, a no-code platform for analysts to explore and visualise blockchain data for non-technical users.

Allium [49] is an enterprise-grade blockchain data platform with Online Analytical Processing (OLAP) capabilities and real-time access to raw and enriched on-chain data via SQL APIs. It enables low-latency querying (50–100ms), making it ideal for high-performance applications. Like other analytics tools, Allium offers Allium Explorer, a web-based platform for querying, analysing, and visualising blockchain data.

E. Discussion

Commercial platforms and practical tools offer solutions for blockchain data access, processing, and analysis. Services like Infura [31] and QuickNode [32] enable easy node access and real-time data retrieval, while Blockchain ETL [35] and Cryo [36] provide open-source tools facilitating advanced data extraction and transformation. Nonetheless, these solutions are often fragmented and blockchain-specific, meaning they do not provide integrated solutions that unify data across different blockchains in a single structure or query endpoint, limiting their ability to offer a holistic view of blockchain data.

Commercial data providers such as GoldRush [38] and Transpose [39] provide the data structure and query endpoint for blockchain data across multiple chains. These platforms offer access to raw and decoded data, as well as advanced features like cross-chain transaction tracking, wallet activity, and security monitoring. Platforms like The Graph [40] stand out by providing structured and indexed data to query specific SCs and DeFi protocols efficiently. While these platforms reduce the complexity of accessing diverse blockchain data and offer some level of integration across blockchains, it is unclear how they handle the unification of UTxO and account-based transaction data. Moreover, commercial platforms often involve subscription fees or data access limitations, which could present barriers for smaller projects or researchers working within tight budgets.

On the analytics front, platforms like BlockScout [43], Dune Analytics [45], Flipside Crypto [47], and Footprint Analytics [48] offer powerful query and visualisation capabilities. While commercial services like these provide robust infrastructure and ease of use, they often come with a trade-off between user-friendliness and the level of control over data processing. In contrast, open-source platforms like Blockchain ETL and The Graph offer more control over data extraction but require higher technical expertise for effective implementation.

To move forward, the next step for blockchain data platforms is to bridge existing gaps by offering more unified, cross-blockchain solutions within a single, granular data structure. This would allow for integrating different blockchain architectures while maintaining open-source principles and transparent data processing. Such an approach would not only simplify access and analysis but also promote data ownership, control, and reproducibility, which are critical factors for both developers and researchers.

V. Unified Token Transfer Data Structure

The heterogeneity of blockchain architectures demands flexible data representations tailored to specific applications. Despite these differences, most blockchain activities ultimately involve the movement of tokens, which can be abstracted as sequential token transfers. Building on this insight, we propose a unified data structure centred on token transfers to represent the discrete movement of value within and across blockchain networks.

Instead of modelling entire transactions in their native, often complex formats, our approach focuses on token transfer activities as the core unit of analysis. This abstraction captures the essential elements required for balance tracking, behavioural analysis, and forensic investigations. This simplification not only promotes standardisation and interoperability but also streamlines data analysis by focusing on the most critical element: token movement. However, it may obscure finer details – such as transaction lineage or internal contract logic – especially in blockchains with more complex operations.

Standardising token transfer data is essential for enabling robust cross-chain analysis and for seamlessly correlating data from both centralised and decentralised exchanges. Our unified model is designed to accommodate native assets as well as SC-generated tokens, including both fungible tokens and NFTs. This approach is applicable across diverse blockchain paradigms, including account-based systems (such as Ethereum) and UTxO-based systems (such as Bitcoin). To mitigate data redundancy and ensure consistency across datasets, our design leverages normalised tables for key entities such as ledgers, blocks, tokens, transactions, and transfers.

A. Flattened Transactions Approach

Building on the Chimeric Ledgers model [16] and solutions such as the GoldRush API [38], our approach "flattens" complex transactions into discrete entries representing token transfers. In this model, each token transfer is recorded as an individual entry. For instance, in a UTxO-based transaction with multiple inputs and outputs, the model records one transfer per input and one per output. For account-based blockchains, each transfer is derived from token transfer events recorded in transaction logs (e.g. ERC-20 tokens) and includes native token transfers (such as Ether on the Ethereum blockchain) embedded directly within transactions.

A key design decision in our unified data representation is the simplification of UTxO data. Instead of preserving the full UTxO reference – which typically includes details like the originating transaction hash and output index – we abstract this information by representing UTxO entries solely by their associated addresses. Specifically, a UTxO input is represented by a `spender_address` (the address spending the input), while a UTxO output is represented by a `receiver_address` (the address receiving the output). This abstraction reduces granularity but aligns well with use cases focused on address-level balance tracking and token flow analysis.

To further clarify, we represent each token transfer by recording its effect on specific addresses. In our model, a token transfer may manifest as a deposit, a withdrawal, or both – depending on the underlying transaction type:

- *Deposits:* In UTxO-based transactions, each output credits tokens to an address, thereby constituting a deposit. Similarly, in account-based transactions, the receipt of tokens by an address is regarded as a deposit. This classification clearly represents the inflow of funds.
- *Withdrawals:* Conversely, UTxO inputs remove tokens from an address and are thus recorded as withdrawals. In account-based systems, when tokens leave an address, it corresponds to a withdrawal. This categorisation allows us to capture the net effect on address balances across various blockchain architectures.

This unified view – abstracting UTxO inputs and outputs into withdrawals and deposits, respectively – enables a consistent representation of token flows regardless of the underlying transaction model. Although this approach omits some low-level details, it offers a pragmatic balance between simplicity and the granularity required for effective analytics and forensic investigations.

B. Proposed Data Model

Our proposed unified blockchain data model (Figure 4) is centred on the `AbstractTokenTransfer` table, which serves as the core schema for representing token movements across UTxO-based and account-based blockchains, as well as across native tokens and smart contract-based assets (e.g. ERC-20, NFTs). The unified token transfer table comprises the following fields:

- `tx_sid`: A unique identifier (surrogate key) of the transaction containing the token transfer. This field is an external key linked to the `AbstractTransaction` table.
- `transfer_index`: An index that orders token transfers within a transaction. For smart contract tokens, this corresponds to the transfer event log index; for UTxO models, it identifies the specific input or output referenced by the token transfer. Together with `tx_sid`, it uniquely identifies each token transfer.
- `token_sid`: A unique identifier of the token involved in the transfer. This field is an external key to the `AbstractToken` table.
- `spender_address`: In account-based models, this represents the address of the token owner from whom the tokens are transferred. For UTxO-based models, it denotes the signer of the transaction that spends the UTxO input. This field is set to `null` when SC tokens are minted or in UTxO-based deposits.
- `receiver_address`: In account-based models, this denotes the recipient's address where tokens are deposited. In UTxO-based models, it represents the owner of the UTxO output. This field is set to null when SC tokens are burned or in UTxO-based withdrawals.

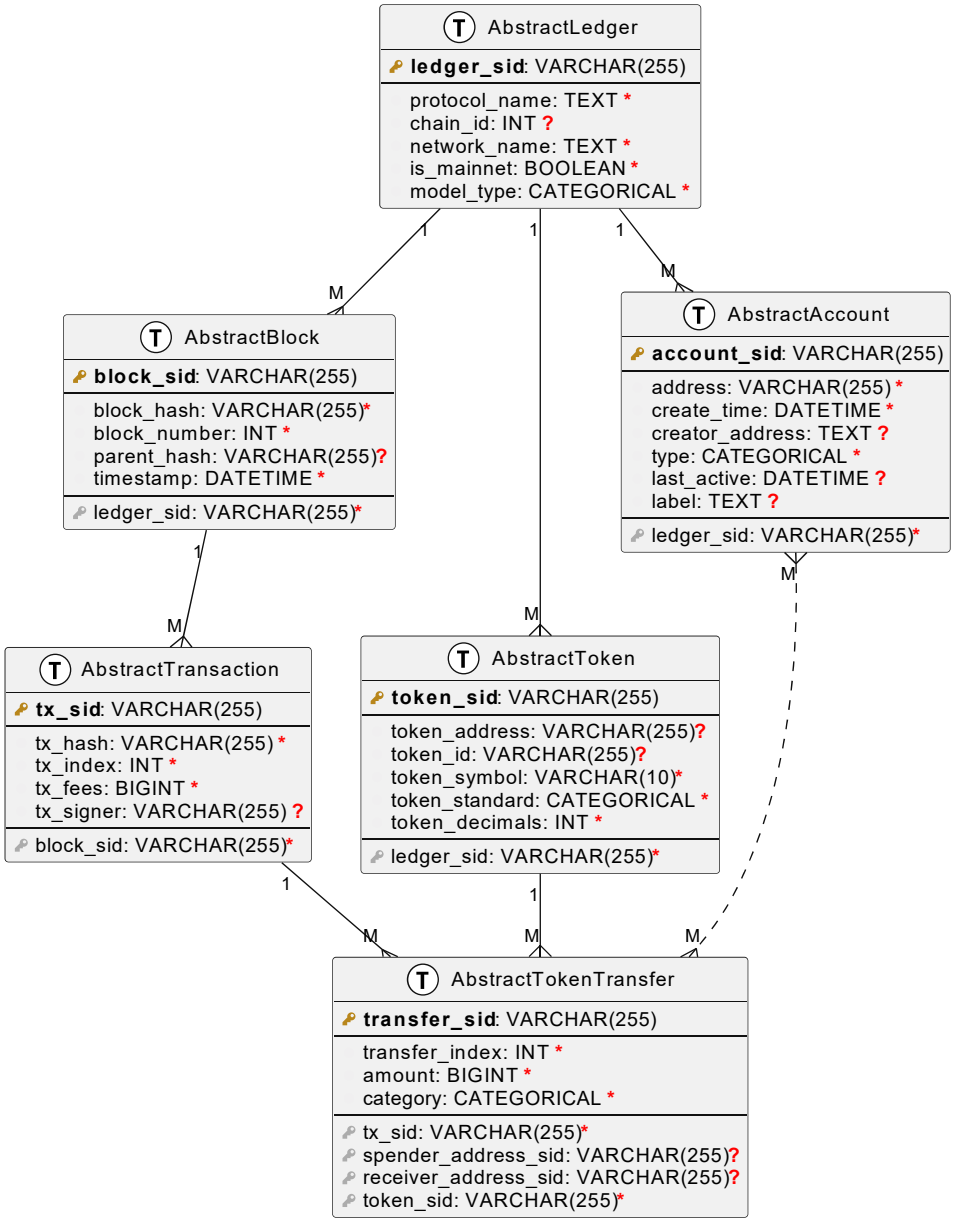

Fig. 4. **ER diagram of the proposed Unified Data Structure**. Each ledger supported by the model is described in the `AbstractLedger` table. Blocks, characterised by their number, hash, and parent, are stored in the `AbstractBlock` table. Accounts, along with optional token balance, label, and activity information, are maintained in the `AbstractAccount` table. Native and non-native tokens (such as NFTs) are stored in the `AbstractToken` table and referenced by transfers. Each transaction recorded in the `AbstractTransaction` table is decomposed into multiple transfers, which are stored in the `AbstractTokenTransfer` table. This structure supports UTxO transactions with multiple inputs and outputs (recorded as individual transfers) as well as smart contract transactions that transfer multiple assets within a single script.
Note: A star (*) indicates a compulsory field, while a question mark (?) denotes an optional one.

- amount: The quantity of tokens transferred, stored as the highest precision decimal representing a large integer to avoid precision loss.
- category: The type of transfer operation (e.g. transfer, mint, burn, deposit, withdrawal). Missing spender or receiver values indicate minting or burning events.

To enrich this core representation, the model is supported by contextual tables that describe essential entities, including blocks, transactions, accounts, tokens, and blockchain ledgers. Each table in the schema adopts a surrogate key convention to ensure uniqueness and facilitate robust data integration across multiple blockchain networks and forks. These surrogate keys, denoted as {table_name}_sid, are constructed by combining the chain_id with the appropriate native identifiers. This design choice avoids conflicts arising from identical hashes or indices across chains. The surrogate keys also serve as primary keys for their respective tables and enable semantic joins.

AbstractTransaction stores transaction data in a format that is compatible with different kinds of ledgers:

- tx_sid: A unique identifier for each transaction, derived by combining chain_id and tx_hash.
- tx_hash: Transaction hash or ID, stored as a hash with appropriate prefixes. Uniquely identifies the transaction.
- tx_index: The order of the transaction within its block.
- tx_fees: The fee paid for the transaction.
- tx_signer: The address of the entity that signed the transaction.
- block_sid: The external key to the AbstractBlock table.

AbstractToken stores data about tokens, including symbol and denomination details for native assets, fungible tokens, and NFTs:

- token_sid: A unique token identifier composed of chain_id and token_address
- token_address: The address of the token contract (for smart contract fungible and non-fungible tokens); null for native tokens.
- token_id: Uniquely identifies NFT within a collection (that is identified by the token_address field). This field is set to null for fungible tokens.
- token_symbol: The symbol or ticker of the token (e.g. BTC, ETH, USDT).
- token_standard: The token standard (e.g. native, ERC-20, ERC-721, ERC-1155).
- token_decimals: The number of decimals for the token's smallest denomination.
- ledger_sid: External key to the AbstractLedger table.

AbstractAccount represents an individual entity identified by its public key address. It serves as a generalised abstraction of both externally owned and smart contracts accounts across different blockchain architectures.

- account_sid: A unique identifier for each address, created by concatenating chain_id and the raw blockchain address.
- address: The hash or unique identifier for the account, smart contract, or transaction input/output address.
- create_time: The UTC timestamp when the account, smart contract, or address was created or first used.
- creator_address: For smart contract addresses, the address of the account that created the contract; null for standard account addresses.
- type: The type of address (e.g. UTxO input/output, Externally Owned Account (EOA), or Smart Contract (SC)).
- last_active: The UTC timestamp indicates when the account was last active when the smart contract was executed.
- label: A human-readable label or description for the address (e.g. an alias or categorisation).
- ledger_sid: External key to the AbstractLedger table.

AbstractBlock stores block-level data, offering a unified representation of blocks regardless of the originating ledger:

- block_sid: A unique block identifier formed by combining chain_id and block_number.
- block_hash: Uniquely identifies the block, stored as a hexadecimal string with appropriate prefixes (i.e. 0x).
- block_number: The block height or numerical identifier. Note that different blockchains or forks may have blocks with the same height.
- parent_hash: The hash of the parent block; null for genesis blocks (i.e. block number 0).
- timestamp: The UTC timestamp when the block was included in the chain.
- ledger_sid: External key to the AbstractLedger table.

Finally, AbstractLedger tracks different blockchain ledgers and their variants, including mainnets, forks, sidechains, and testnets.

- ledger_sid: A unique identifier for blockchain ledger
- protocol_name: The name of the protocol that implements the ledger.
- chain_id: The identifier for the specific chain (e.g. 1 for Ethereum Mainnet, 17000 for Ethereum Holesky Testnet).
- network_name: The name of the specific chain (e.g. btc-main, eth-main, eth-gorli, polygon, etc.).
- is_mainnet: Indicates whether the ledger is a main network or a testnet.
- model_type: The underlying model type (e.g. UTxO-based, account-based).

C. Considerations and Trade-offs

This unified and simplified token transfer data structure facilitates efficient cross-chain analysis by standardising diverse transaction formats. It supports forensic and analytical tasks while avoiding extraneous complexity. However, this design involves several trade-offs, which are outlined below:

Granularity vs. Simplicity: Substituting detailed UTxO references with `spender_address` and `receiver_address` eliminates the ability to reconstruct exact UTxO lineages. Despite this loss of granularity, the substitution considerably simplifies the schema for use cases centred on net token flows and balance analysis – common requirements in forensic and behavioural analysis.

Unified View Across Models: The flattened structure provides a consistent representation across UTxO- and account-based blockchains, as well as native and SC tokens. It facilitates integration and comparison across different chains and token types. The model incorporates internal transactions (e.g. SC events) alongside native token transfers, simplifying queries and enabling coherent joins with other blockchain data such as block metadata and transaction logs.

Applicability: Our simplified schema intentionally abstracts low-level details, such as full UTxO references and trace-level execution, to prioritise scalability, interpretability, and storage efficiency. This design is well-suited for analytics and forensic applications focused on token flows and balance changes over time rather than reconstructing full transaction lineages or contract execution paths. By flattening transactions into token-level transfers, the model reduces data complexity and enables more efficient querying across heterogeneous blockchains.

Nonetheless, this abstraction may introduce limitations for use cases that depend on fine-grained provenance data. For instance, identifying *maximal extractable value (MEV)* transactions often requires precise trace-level information to analyse transaction ordering and internal contract calls, which may be obscured in our abstracted model. Likewise, tracing Anti-Money Laundering (AML) flows based on UTxO lineage or multi-input address clustering may lose resolution when such structures are simplified. Despite these trade-offs, the unified token-level view enhances cross-chain compatibility and supports behavioural clustering, making it a pragmatic choice for many blockchain analytics tasks.

Anonymity: By abstracting away low-level details, particularly in UTxO models, the unified model may limit the visibility of privacy-preserving techniques (e.g. CoinJoin and stealth addresses). This abstraction could hinder fine-grained forensic investigations that rely on transaction lineage or script-level execution data. Inadvertently, centralising diverse blockchain data into a common structure may facilitate de-anonymisation when combined with off-chain identifiers.

Protocol Compatibility: As blockchains evolve and introduce new transaction types, token standards, or execution models, maintaining a unified schema becomes more complex. Protocol-specific data requires custom parsing logic, leading to potential fragmentation within the unification framework. Moreover, reconciling semantic differences (e.g. handling zero-value transfers, gasless meta-transactions, or state transitions) may challenge long-term schema consistency. While blockchain data abstraction supports interoperability, it should be designed with flexibility and extensibility in mind to accommodate emerging blockchain paradigms.

VI. Discussion

This section reviews prior surveys on blockchain data management and analytics, highlighting their key contributions while positioning our work within this landscape. We highlight how our approach differs, particularly in addressing data structure integration, and identify open challenges and research directions to advance cross-chain analytics and interoperability.

A. Related Work

Apart from generic descriptions and reviews of blockchain technology (e.g. [50]–[52]), existing blockchain data surveys focus primarily on data management (performance and storage) or data analytics (analysing transactions and detecting fraud). Most studies address only one side and do not explore blockchain data structures that connect both aspects. Wei et al. [53] classify blockchain systems into standard (e.g. Bitcoin), hybrid (e.g. Fabric), and DAG-based (e.g. IOTA), describing optimisation techniques for architecture, data structures, and storage engines. Paik et al. [54] examine trade-offs between on-chain and off-chain storage while exploring blockchain analytics for fraud detection and financial tracking. Dinh et al. [55] compare blockchain platforms and introduce benchmarks to assess the data processing performance of Ethereum, Parity, and Hyperledger. Loghin [56] categorise blockchain databases into permissioned, hybrid, and ledger databases and compare their scalability and transaction throughput.

Beyond data management, surveys on blockchain analytics focus on extracting insights from transaction data. Akcora et al. [57] review analytics methods, tools, and applications in finance, identifying gaps in multi-blockchain compatibility and large-scale data visualisation. Liu et al. [58] explore data mining approaches in blockchain analytics, categorising tasks into transaction traceability and address linking, collective user behaviour, and individual user behavior. Balaskas and Franqueira [59] evaluate tools for transaction tracking and fraud detection, identify challenges such as data volume, pseudonymity, and real-time processing, and highlight the need for improved cross-chain analytics tools. Tovanich et al. [60] examine visualisation techniques for blockchain data, highlighting the lack of visualisation tools that support smart contract execution and addressed the specialised needs of blockchain analysts and domain experts.

Two particular surveys focus on network analysis of blockchain data. Wu et al. [22] provide an overview of network analysis pipelines, covering network modelling, statistical profiling, and analysis tasks such as illicit activity detection and entity recognition. Khan [61] surveys network analysis methods applied to Ethereum, categorising them into network modelling, extracted network properties, machine learning techniques, and target applications. Azad, Akcora, and Khan [62] review machine learning techniques with a focus on graph-based and temporal methods. The study demonstrates how machine learning enhances fraud detection, market behaviour analysis, and the security of smart contracts.

Most analytics studies in the surveyed literature focus on a single blockchain. Emerging research proposes methodologies for detecting and analysing cross-chain transactions, including [63]–[66]. An increasing body of research addresses Maximal Extractable Value (MEV) – the profit miners or validators gain by inserting, reordering, or censoring transactions within blocks [67], [68]. Several studies detect and quantify MEV, especially in private mempools [69]–[71]. Recent surveys provide overviews of MEV techniques [72], [73], while others focus on mitigation strategies, including consensus changes, transaction ordering policies, and privacy mechanisms [73]–[75]. To make MEV extraction more transparent, Ethereum introduced block-building auctions through its proposer-builder separation (PBS) model [76], [77].

Finally, Vo, Kundu, and Mohania [78] highlight key data management and analytics challenges, including scalability and privacy. The study emphasises the need to integrate on-chain and off-chain data to ensure interoperability between different blockchain networks. Deepa et al. [79] outline research challenges in integrating blockchain with big data applications, concerning security, privacy, computational overhead, and the need for standardisation across diverse domains.

Despite these contributions, none of the reviewed studies discusses blockchain data structures and modelling, particularly in the context of comparing UTxO-based and account-based models. Our work bridges the gap between surveys on data management and analytics, addressing the challenges of unifying blockchain data structures to facilitate blockchain data analytics tasks, particularly for cross-chain transactions, rather than focusing solely on data management performance.

B. Limitations and Future Work

Our extensive review of academic solutions (Section III), as well as open-source and commercial practical tools (Section IV), focuses on analysing relevant research and technical documentation rather than deploying tools or benchmarking performance (such as BlockSci [19]). As part of future work, performance evaluation will be essential to determine suitable backend architectures for implementing our proposed unified data model to enable standardised and reproducible blockchain research.

For commercial platforms, we did not systematically create accounts or directly test public APIs. Instead, our analysis relies on a detailed study of documentation, which proved essential for understanding key data structures, API endpoints, and the intended analytical use cases. Although this paper does not list or compare the internal data models of each platform, these insights were instrumental in identifying commonalities and gaps relevant to our unified data model.

While we compare blockchain data platforms, deeper technical benchmarking is limited by the lack of standard evaluation metrics across tools. Many academic and open-source solutions either use different evaluation strategies or do not report performance. Benchmarking blockchain systems is the focus of other works, including [55], [80]–[83].

Our data model design involves trade-offs guided by the technical challenges outlined in Section II. However, these decisions need to be validated in practice. Our future work includes implementing this unified schema on public blockchains, testing scalability, querying performance, and integration with analytics platforms. Quantitative metrics include query efficiency, storage overhead, and scalability across chains, while qualitative assessments focus on usability, interpretability, and support for practical applications such as MEV detection, AML heuristics, and cross-chain analysis.

Given the rapid evolution of the blockchain landscape, our model will need to adapt to emerging paradigms. While we have focused on account-based and UTxO-based architectures, we also considered hybrid approaches such as Cardano's extended UTxO and NANO's block-lattice design. Expanding blockchain use cases – including DeFi, DAOs, and supply chain tracking – will produce increasingly diverse data, spanning multiple chains and abstraction layers. A flexible, abstract data model supporting unified analysis across such ecosystems will be vital for advancing future blockchain research.

Finally, ethical and legal concerns arise in standardising blockchain data, particularly regarding user privacy. While our model retains pseudonymity (address-based), the risk of de-anonymisation increases when on-chain data is combined with external sources such as KYC databases, address clustering, or IP addresses [27], [84], [85]. Regulatory frameworks such as the *General Data Protection Regulation (GDPR)* may constrain how unified models can be deployed, stored, or shared [86], [87].

VII. Conclusion

This paper addressed the challenges of blockchain data unification by analysing existing fragmentation in data structures across blockchain platforms. We reviewed prior efforts, both academic and practical solutions, to standardise blockchain data and identified their limitations in handling diverse models. Our study highlighted the need for a unified approach that integrates both UTxO-based and account-based structures while maintaining flexibility for various blockchain architectures.

To bridge this gap, we proposed a novel data model that abstracts blockchain data, unifies addresses, and "flattens" transactions into simpler token transfers, providing a standardised framework for analytics and research. This model enhances interoperability across different blockchain ecosystems, facilitating cross-chain transaction analysis and improving data accessibility for both researchers and developers.

Future research should focus on implementing and benchmarking this model across multiple blockchains to validate its scalability and efficiency. It will be crucial to explore ways to extend this model for emerging blockchain paradigms, such as modular and cross-chain protocols. Practical applications include its integration into blockchain analytics platforms, DeFi risk assessment tools, and fraud detection systems, ensuring more reliable and reproducible blockchain data analysis.

ACKNOWLEDEGMENTS

We thank Ünsal Öztürk, Stéphane Augusto, and Christian Russo for their thoughtful suggestions to improve our work and for their generous help with proofreading the manuscript.

REFERENCES

[1] S. Nakamoto, "Bitcoin: a Peer-to-Peer Electronic Cash System," White Paper, 10 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] F. Schär, "Decentralized Finance: On Blockchain- and Smart Contract-based Financial Markets," *SSRN Electronic Journal*, 2020.

[3] Cardano.org, "Extended UTXO model — Cardano Docs," Cardano.org. [Online]. Available: https://docs.cardano.org/about-cardano/learn/eutxo-explainer/

[4] J. Greene, *Cardano for the Masses*, 2022.

[5] M. Kondo, "EMURGO Africa 2023 Q1 Report: Coin Market and Tokenized Projects," EMURGO Africa, 05 2023. [Online]. Available: https://www.blog.emurgo.africa/emurgo-africa-2023-q1-report

[6] C. Lemahieu, "Nano: A Feeless Distributed Cryptocurrency Network," White Paper, 2018. [Online]. Available: https://content.nano.org/whitepaper/Nano_Whitepaper_en.pdf

[7] A. Yakovenko, "Solana: A new architecture for a high performance blockchain ," Solana.com, 11 2017. [Online]. Available: https://solana.com/solana-whitepaper.pdf

[8] C. Badertscher, P. Gaži, A. Kiayias, A. Russell, and V. Zikas, "Ouroboros Genesis: Composable Proof-of-Stake Blockchains with Dynamic Availability," *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 01 2018. [Online]. Available: https://eprint.iacr.org/2018/378.pdf

[9] C. Badertscher, P. Gaži, A. Kiayias, A. Russell, and V. Zikas, "Ouroboros Chronos: Permissionless Clock Synchronization via Proof-of-Stake," 2019. [Online]. Available: https://eprint.iacr.org/2019/838

[10] F. Vogelsteller and V. Buterin, "EIP 20: ERC-20 Token Standard," White Paper, 11 2015. [Online]. Available: https://eips.ethereum.org/EIPS/eip-20

[11] W. Entriken, D. Shirley, J. Evans, and N. Sachs, "EIP 721: ERC-721 Non-Fungible Token Standard," White Paper, 01 2018. [Online]. Available: https://eips.ethereum.org/EIPS/eip-721

[12] W. Radomski, A. Cooke, P. Castonguay, J. Therien, E. Binet, and R. Sandford, "EIP 1155: ERC-1155 Multi Token Standard," White Paper, 06 2018. [Online]. Available: https://eips.ethereum.org/EIPS/eip-1155

[13] Q. Wang, G. Yu, and S. Chen, "Bridging BRC-20 to Ethereum," *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, vol. 21, p. 264–272, May 2024. [Online]. Available: https://arxiv.org/pdf/2310.10065v2

[14] Q. Wang and Y. Guo, "Understanding BRC-20: Hope or Hype," Oct 2023.

[15] J. ao Vicente Meyer and S. Mello, "An Analysis of Data Modelling for Blockchain," *Lecture Notes in Computer Science*, p. 31–44, Jan 2022.

[16] J. Zahnentferner, "Chimeric Ledgers: Translating and Unifying UTXO-based and Account-based Cryptocurrencies," *Cryptology ePrint Archive*, 2018. [Online]. Available: https://eprint.iacr.org/2018/262

[17] C. G. Akcora, Y. R. Gel, and M. Kantarcioglu, "Blockchain networks: Data structures of Bitcoin, Monero, Zcash, Ethereum, Ripple, and Iota," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 12, no. 1, p. e1436, 2022.

[18] M. Bartoletti, S. Lande, L. Pompianu, and A. Bracciali, "A general framework for blockchain analytics," in *Proceedings of the 1st Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers*, 2017, pp. 1–6.

[19] H. Kalodner, M. Möser, K. Lee, S. Goldfeder, M. Plattner, A. Chator, and A. Narayanan, "BlockSci: Design and applications of a blockchain analysis platform," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 2721–2738.

[20] B. Haslhofer, R. Stütz, M. Romiti, and R. King, "GraphSense: A general-purpose cryptoasset analytics platform," *arXiv preprint arXiv:2102.13613*, 2021.

[21] N. Atzei, M. Bartoletti, S. Lande, and R. Zunino, "A formal model of Bitcoin transactions," *Cryptology ePrint Archive*, 2017. [Online]. Available: https://eprint.iacr.org/2017/1124.pdf

[22] J. Wu, J. Liu, Y. Zhao, and Z. Zheng, "Analysis of Cryptocurrency Transactions from a Network Perspective: An Overview," *Journal of Network and Computer Applications*, vol. 190, p. 103139, 2021.

[23] O. Melkonian, "GitHub - omelkonian/formal-utxo: Formalization of the UTxO abstract model for (bitcoin-style) blockchain transactions." 2018. [Online]. Available: https://github.com/omelkonian/formal-utxo

[24] M. Harrigan and C. Fretter, "The Unreasonable Effectiveness of Address Clustering," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, 2016, pp. 368–373.

[25] Y. Zhang, J. Wang, and J. Luo, "Heuristic-Based Address Clustering in Bitcoin," *IEEE Access*, vol. 8, pp. 210 582–210 591, 2020.

[26] M. Möser and A. Narayanan, "Resurrecting Address Clustering in Bitcoin," in *International Conference on Financial Cryptography and Data Security*. Springer, 2022, pp. 386–403.

[27] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A Fistful of Bitcoins: Characterizing Payments Among Men with No Names," *Communications of the ACM*, vol. 59, pp. 86–93, 03 2016.

[28] S. Goldfeder, H. Kalodner, D. Reisman, and A. Narayanan, "When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, pp. 179–199, 10 2018.

[29] GraphSense, "GitHub - graphsense/graphsense-tagpacks: A collection of public TagPacks," 2024. [Online]. Available: https://github.com/graphsense/graphsense-tagpacks

[30] M. Bartoletti, S. Lande, L. Pompianu, and A. Bracciali, "A general framework for blockchain analytics," *Workshop Scalable and Resilient Infrastructures for Distributed Ledgers*, 12 2017.

[31] MetaMask, Inc., "Infura," 2025. [Online]. Available: https://docs.metamask.io/services/

[32] Quicknode, Inc., "Quick Node," 2025. [Online]. Available: https://www.quicknode.com/docs/welcome

[33] Alchemy, Inc., "Alchemy," 2025. [Online]. Available: https://docs.alchemy.com/reference/transfers-api-quickstart#2-erc20-transfers

[34] Moralis Developers, "Moralis," 2023. [Online]. Available: https://docs.moralis.com/web3-data-api/evm

[35] Blockchain ETL, "GitHub - blockchain-etl/public-datasets: The list of public blockchain datasets in BigQuery," 2023. [Online]. Available: https://github.com/blockchain-etl/public-datasets

[36] Paradigm, "GitHub - paradigmxyz/cryo: cryo is the easiest way to extract blockchain data to parquet, csv, json, or python dataframes," 2023. [Online]. Available: https://github.com/paradigmxyz/cryo

[37] Blockchair, "Blockchain API Documentation," Blockchair.com, 2024. [Online]. Available: https://blockchair.com/api/docs#link

[38] Goldrush, Inc., "Goldrush," 2024. [Online]. Available: https://goldrush.mintlify.app/overview

[39] Chainalysis Inc., "Overview - Transpose Documentation," Transpose.io, 2024. [Online]. Available: https://docs.transpose.io/data/overview/

[40] The Graph Foundation, "The Graph Docs," 2025. [Online]. Available: https://thegraph.com/docs/en/

[41] Edge & Node, "6,000+ Subgraphs Such as AAVE, Balancer, ENS & SushiSwap Live on Network: Sunrise is Here," 2024. [Online]. Available: https://thegraph.com/blog/sunrise-here-the-graph-network/

[42] Messari, Inc., "GitHub - messari/subgraphs: Standardized subgraphs for blockchain data," 2025. [Online]. Available: https://github.com/messari/subgraphs/tree/master

[43] Blockscout Limited, "Blockscout Blockchain Explorer," 2025. [Online]. Available: https://docs.blockscout.com/

[44] ——, "Blockscout Database Schema," 2025. [Online]. Available: https://docs.blockscout.com/setup/db-schema

[45] D. Analytics, "Data Catalog - Dune Docs," Dune.com, 2023. [Online]. Available: https://docs.dune.com/data-catalog

[46] Dune Analytics, "GitHub - duneanalytics/spellbook: SQL views for Dune," 2025. [Online]. Available: https://github.com/duneanalytics/spellbook

[47] F. Crypto, "Table Docs by Chain — Flipside Docs," Flipsidecrypto.xyz, 10 2024. [Online]. Available: https://docs.flipsidecrypto.xyz/data/flipside-data/data-table-documentation

[48] Footprint Analytics, "Footprint Analytics Documentation: Design Concept," 2025. [Online]. Available: https://docs.footprint.network/docs/designconcept

[49] Allium, "Allium Documentation," 2025. [Online]. Available: https://docs.allium.so/

[50] W. Gao, W. G. Hatcher, and W. Yu, "A Survey of Blockchain: Techniques, Applications, and Challenges," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2018, pp. 1–11.

[51] H. Guo and X. Yu, "A Survey on Blockchain Technology and its Security," *Blockchain: Research and Applications*, vol. 3, no. 2, p. 100067, 2022.

[52] M. Xu, Y. Guo, C. Liu, Q. Hu, D. Yu, Z. Xiong, D. Niyato, and X. Cheng, "Exploring Blockchain Technology through a Modular Lens: A Survey," *ACM Computing Surveys*, vol. 56, no. 9, pp. 1–39, 2024.

[53] Q. Wei, B. Li, W. Chang, Z. Jia, Z. Shen, and Z. Shao, "A Survey of Blockchain Data Management Systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 21, no. 3, pp. 1–28, 2022.

[54] H.-Y. Paik, X. Xu, H. D. Bandara, S. U. Lee, and S. K. Lo, "Analysis of Data Management in Blockchain-Based Systems: From Architecture to Governance," *IEEE Access*, vol. 7, pp. 186 091–186 107, 2019.

[55] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling Blockchain: A Data Processing View of Blockchain Systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1366–1385, 2018.

[56] D. Loghin, "The Anatomy of Blockchain Database Systems," *Data Engineering*, p. 48, 2022.

[57] C. G. Akcora, M. F. Dixon, Y. R. Gel, and M. Kantarcioglu, "Blockchain Data Analytics," *Intelligent Informatics*, vol. 4, no. 6, 2018.

[58] X. F. Liu, X.-J. Jiang, S.-H. Liu, and C. K. Tse, "Knowledge Discovery in Cryptocurrency Transactions: A Survey," *IEEE Access*, vol. 9, pp. 37 229–37 254, 2021.

[59] A. Balaskas and V. N. Franqueira, "Analytical Tools for Blockchain: Review, Taxonomy and Open Challenges," in *2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. IEEE, 2018, pp. 1–8.

[60] N. Tovanich, N. Heulot, J.-D. Fekete, and P. Isenberg, "Visualization of Blockchain Data: A Systematic Review," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 7, pp. 3135–3152, 2019.

[61] A. Khan, "Graph Analysis of the Ethereum Blockchain Data: A Survey of Datasets, Methods, and Future Work," in *2022 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2022, pp. 250–257.

[62] P. Azad, C. G. Akcora, and A. Khan, "Machine Learning for Blockchain Data Analysis: Progress and Opportunities," *arXiv preprint arXiv:2404.18251*, 2024.

[63] R. Belchior, P. Somogyvari, J. Pfannschmidt, A. Vasconcelos, and M. Correia, "Hephaestus: Modeling, Analysis, and Performance Evaluation of Cross-Chain Transactions," *IEEE Transactions on Reliability*, vol. 73, no. 2, pp. 1132–1146, 2023.

[64] Z. Zheng, J. Wu, D. Lin, Q. Li, and N. Ruan, "XSema: A Novel Framework for Semantic Extraction of Cross-chain Transactions," *arXiv preprint arXiv:2412.18129*, 2024.

[65] A. Augusto, A. Vasconcelos, M. Correia, and L. Zhang, "XChainDataGen: A Cross-Chain Dataset Generation Framework," *arXiv preprint arXiv:2503.13637*, 2025.

[66] D. Lin, Z. Zheng, J. Wu, J. Yang, K. Lin, H. Xiao, B. Song, and Z. Zheng, "Track and Trace: Automatically Uncovering Cross-chain Transactions in the Multi-blockchain Ecosystems," *arXiv preprint arXiv:2504.01822*, 2025.

[67] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 910–927.

[68] K. Qin, L. Zhou, and A. Gervais, "Quantifying Blockchain Extractable Value: How dark is the forest?" in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 198–214.

[69] B. Weintraub, C. F. Torres, C. Nita-Rotaru, and R. State, "A flash (bot) in the pan: measuring maximal extractable value in private pools," in *Proceedings of the 22nd ACM Internet Measurement Conference*, 2022, pp. 458–471.

[70] Z. Li, J. Li, Z. He, X. Luo, T. Wang, X. Ni, W. Yang, X. Chen, and T. Chen, "Demystifying defi MEV Activities in Flashbots Bundle," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 165–179.

[71] T. Chi, N. He, X. Hu, and H. Wang, "Remeasuring the arbitrage and sandwich attacks of maximal extractable value in Ethereum," *arXiv preprint arXiv:2405.17944*, 2024.

[72] V. Gramlich, D. Jelito, and J. Sedlmeir, "Maximal extractable value: Current understanding, categorization, and open research questions," *Electronic Markets*, vol. 34, no. 1, p. 49, 2024.

[73] H. Materwala, S. M. Naik, A. Taha, T. A. Abed, and D. Svetinovic, "Maximal Extractable Value in Decentralized Finance: Taxonomy, Detection, and Mitigation," *arXiv preprint arXiv:2411.03327*, 2024.

[74] Z. Alipanahloo, A. S. Hafid, and K. Zhang, "Maximum Extractable Value (MEV) Mitigation Approaches in Ethereum and Layer-2 Chains: A Comprehensive Survey," *IEEE Access*, 2024.

[75] L. Heimbach and R. Wattenhofer, "SoK: Preventing Transaction Reordering Manipulations in Decentralized Finance," in *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*, 2022, pp. 47–60.

[76] L. Heimbach, L. Kiffer, C. Ferreira Torres, and R. Wattenhofer, "Ethereum's Proposer-Builder Separation: Promises and Realities," in *Proceedings of the 2023 ACM on Internet Measurement Conference*, 2023, pp. 406–420.

[77] B. Öz, D. Sui, T. Thiery, and F. Matthes, "Who wins Ethereum Block building Auctions and Why?" *arXiv preprint arXiv:2407.13931*, 2024.

[78] H. T. Vo, A. Kundu, and M. K. Mohania, "Research Directions in Blockchain Data Management and Analytics." in *EDBT*, 2018, pp. 445–448.

[79] N. Deepa, Q.-V. Pham, D. C. Nguyen, S. Bhattacharya, B. Prabadevi, T. R. Gadekallu, P. K. R. Maddikunta, F. Fang, and P. N. Pathirana, "A survey on blockchain for big data: Approaches, opportunities, and future directions," *Future Generation Computer Systems*, vol. 131, pp. 209–226, 2022.

[80] C. Fan, S. Ghaemi, H. Khazaei, and P. Musilek, "Performance evaluation of blockchain systems: A systematic survey," *IEEE Access*, vol. 8, pp. 126 927–126 950, 2020.

[81] G. Wang, Y. Zhang, C. Ying, X. Lit, and G. Yu, "Hammer: A general blockchain evaluation framework," in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2024, pp. 391–402.

[82] D. Saingre, T. Ledoux, and J.-M. Menaud, "BCTMark: a framework for benchmarking blockchain technologies," in *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2020, pp. 1–8.

[83] B. Nasrulin, M. De Vos, G. Ishmaev, and J. Pouwelse, "Gromit: Benchmarking the performance and scalability of blockchain systems," in *2022 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*. IEEE, 2022, pp. 56–63.

[84] G. Kappos, H. Yousaf, M. Maller, and S. Meiklejohn, "An empirical analysis of anonymity in Zcash," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 463–477.

[85] A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of clients in Bitcoin P2P network," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 15–29.

[86] M. Finck, "Blockchain and the general data protection regulation: Can distributed ledgers be squared with european data protection law?" European Parliamentary Research Service, Tech. Rep. PE 634.445, 2019. [Online]. Available: https://www.europarl.europa.eu/RegData/etudes/STUD/2019/634445/EPRS_STU(2019)634445_EN.pdf

[87] European Data Protection Board, "Guidelines on the processing of personal data in the context of the use of blockchain technology," https://www.edpb.europa.eu/our-work-tools/documents/public-consultations/2025/guidelines-022025-processing-personal-data_en, Feb 2025.