



RESEARCH ARTICLE

ENHANCING THE EFFICIENCY OF ETHEREUM BLOCKCHAIN OPERATIONS BASED ON ONTOLOGY

Waddah Ahmed Munassar^{1,*}, and Noora Nasser Humssan¹¹ Dept. of Information Technology, Faculty of Engineering, University of Aden, Yemen*Corresponding author: Waddah Ahmed Munassar; E-mail: wmunassar@yahoo.com & noorhumssan1990@yahoo.com

Received: 04 March 2026 / Accepted: 25 March 2026 / Published online: 31 March 2026

Abstract

This study presents an ontology-driven framework, EthOn, aimed at enhancing the operational efficiency of the Ethereum blockchain. Despite Ethereum's prominence as a decentralized platform for executing smart contracts, it still faces significant limitations such as high gas fees, low throughput, and limited semantic processing capabilities. These challenges primarily stem from the platform's reliance on syntactic data structures and the absence of context-aware mechanisms in smart contracts. The proposed framework integrates semantic ontologies to structure blockchain data and enhance smart contract logic through context-driven reasoning. By embedding domain-specific ontologies into the Ethereum layer, EthOn enables intelligent data interpretation, optimized contract execution, and improved interoperability for decentralized applications (dApps). A prototype developed using Protégé, Node.js, Web3.js, Ganache Network and Apache Jena demonstrated notable advancements in both execution speed and gas efficiency. The results demonstrated remarkable improvements: Ethereum required an average block creation time of 10–15 seconds with gas consumption fluctuating between 9.6 million and 44.4 million units, while EthOn achieved block generation within 1–2 seconds and maintained gas usage between 423,630 and 3.7 million units. This represents an overall gas reduction of nearly 90% and a substantial improvement in execution speed. These results highlight the transformative potential of semantic technologies in reshaping blockchain infrastructure, paving the way for more scalable, efficient, and intelligent decentralized systems.

Keywords: Blockchain; Ethereum; Ontology; EthOn; Gas price; Semantic web; Transaction execution.

1. Introduction

Blockchain technology has emerged as a transformative innovation that redefines how data is recorded, verified, and exchanged within distributed environments. By enabling decentralized, transparent, and tamper-proof ledgers, blockchain eliminates the need for centralized intermediaries, thus reshaping digital trust [1]. Among the prominent blockchain platforms, Ethereum stands out as a second-generation blockchain that introduced programmable smart contracts and support for decentralized applications (dApps), fostering innovation across diverse sectors such as finance, healthcare, supply chains, and identity management [2][3]. Despite Ethereum's functional richness and widespread adoption, the platform suffers from key performance limitations that hinder its scalability. These include high gas fees, limited throughput, and latency in transaction finality, particularly under high network load [4].

Additionally, Ethereum's underlying data architecture is syntactic in nature and lacks semantic expressiveness, making it difficult to model, interpret, or reason over complex and heterogeneous data [5]. Such limitations constrain the development of intelligent, data-aware blockchain applications capable of dynamic reasoning and automation. In contrast, Semantic Web technologies—particularly ontologies—offer a structured, machine-interpretable representation of domain knowledge that promotes interoperability, contextual awareness, and logical inference [6][7]. Ontologies formally define domain-specific vocabularies and relationships, enabling semantic annotation and advanced query capabilities. Integrating ontologies into blockchain platforms enhances the ability to understand and reason over blockchain data and smart contract logic, thereby improving automation, conflict resolution, and adaptability [8]. While existing research has primarily

focused on using blockchain to enhance the integrity and traceability of semantic data, limited attention has been paid to how semantic technologies—specifically ontologies—can improve blockchain performance and smart contract efficiency [9][10].

This study aims to bridge that gap by proposing a novel ontology-driven framework called ETHON, designed to enhance the operational efficiency of Ethereum. The framework utilizes domain ontologies to semantically structure transactional data and guide smart contract behavior. EthOn was implemented using the Protégé tool for ontology modeling and integrated with Web3.js libraries to interact with Ethereum test networks. Experimental evaluations demonstrate that the framework leads to measurable improvements in execution time, computational efficiency, and gas consumption, presenting a viable approach for building semantically enhanced blockchain systems.

2. Literature Review

2.1 Related Works

Over the past decade, numerous studies have addressed the performance limitations of the Ethereum blockchain, particularly regarding gas consumption and transaction throughput. Early research by [12], the original architect of Ethereum, acknowledged that the Ethereum Virtual Machine (EVM) trades performance for generality, which limits transaction speed and increases computation costs. Subsequent works attempted to mitigate these limitations through protocol-level optimizations and off-chain computation. For instance, [13] introduced sharding as a scalability solution, enabling parallel processing of transactions across subnetworks, although its implementation within Ethereum 2.0 remains ongoing and complex. Another line of research focused on optimizing gas efficiency at the smart contract level. [14] proposed Oyente, an analysis tool that detects costly patterns in smart contracts and identifies gas-consuming vulnerabilities. Similarly, [15] developed a compiler-based optimization framework that rewrites Solidity code to reduce redundant operations and lower gas usage without altering contract semantics. More recently, [16] suggested leveraging off-chain computation via Layer-2 solutions like Optimistic Rollups and zk-Rollups to reduce on-chain load and improve throughput, though such models introduce new trust assumptions and interoperability challenges. Despite these efforts, most proposed enhancements are either confined to protocol-level modifications—requiring consensus changes and wide adoption—or rely on external infrastructure, thereby introducing complexity and potential centralization risks. There remains a critical gap in research concerning the use of semantic enrichment or ontology-based reasoning as a lightweight yet powerful

alternative to optimize Ethereum's performance without altering its core architecture.

2.2 Integration of Ontologies with Blockchain

The integration of ontologies with blockchain technology has garnered increasing attention in recent years, as researchers aim to enhance data interoperability, contextual awareness, and intelligent automation within decentralized systems. Ontologies, defined as formal representations of knowledge structures using concepts, properties, and relationships, enable machines to reason semantically over complex datasets. When applied to blockchain, this semantic enrichment allows decentralized applications to transition from purely syntactic operations to knowledge-driven interactions.[17] pioneered the conceptual fusion of semantic web technologies with smart contracts, arguing that ontologies can standardize data across decentralized networks and support dynamic contract logic through reasoning. They proposed a semantic middleware that enables Ethereum contracts to access and interpret RDF-based knowledge graphs in real-time. Similarly, [18] presented EduCTX, a blockchain-based higher education credit system enhanced by ontologies to unify credential semantics across international institutions, demonstrating how shared vocabularies can solve interoperability challenges. Later, [19] introduced a semantic architecture for blockchain that employs OWL ontologies and SPARQL endpoints to provide semantically enriched queries over blockchain data. This architecture facilitates enhanced discoverability, complex querying, and machine reasoning over decentralized records. Another significant work by [20] examined the role of ontologies in multi-chain interoperability, asserting that semantic layers can abstract heterogeneity across blockchains and act as translators between incompatible data formats. Despite these advances, most existing implementations remain limited to specific domains or are constrained by performance overheads introduced by semantic processing. Moreover, few studies explore the integration of ontologies directly within the transaction execution layer of smart contracts—an area that this study aims to address through the proposed EthOn framework, which embeds semantic reasoning into the heart of Ethereum's operational stack.

2.3 Research Gaps and Contributions of This Study

While numerous studies have explored methods to enhance Ethereum's performance or to integrate semantic technologies into distributed systems, there remains a noticeable lack of unified approaches that address both concerns simultaneously. Existing performance-enhancement solutions—such as protocol-level improvements (e.g., sharding, Ethereum 2.0), Layer-2 scaling (e.g., Optimistic Rollups), or optimized smart contract compilers—often focus exclusively on

scalability and efficiency, without incorporating semantic reasoning or domain knowledge into the execution layer. As a result, current Ethereum-based systems continue to operate in syntactic and deterministic environments that lack contextual intelligence. Conversely, most research in semantic blockchain integration has concentrated on metadata annotation, knowledge graph linking, or off-chain reasoning, which does not directly impact smart contract execution. For instance, [21] proposed a semantic model to classify blockchain data, but the framework operates at the data indexing level without influencing the logic of contracts themselves. Similarly, [22] surveyed blockchain-ontology integrations but noted that most existing approaches remain domain-specific, exploratory, or suffer from performance bottlenecks due to external reasoning components. To date, few frameworks embed ontological models directly into the core transaction lifecycle of blockchain networks—particularly within Ethereum's smart contract execution. This gap leaves room for novel architectures that can semantically interpret blockchain interactions in real time, enforce rule-based constraints, and optimize resource usage without relying on protocol modification or centralized middleware. The EthOn framework proposed in this study aims to fill this void by embedding domain-specific ontologies into Ethereum's smart contract logic and transaction handling. Unlike previous works that separate semantics from execution, EthOn incorporates semantic reasoning at the operational level using OWL ontologies, Apache Jena for inference, and Web3.js for blockchain interaction. This integrated approach not only enhances gas efficiency and transaction speed, but also introduces machine-readable contextual logic into decentralized environments—contributing both practically and theoretically to the advancement of intelligent blockchain systems.

3. Methodology



This study employs an implementation-driven methodology designed to evaluate the impact of semantic

ontologies on the performance of the Ethereum blockchain. The approach is divided into four structured phases: ontology modeling, system integration, semantic execution, and performance evaluation.

- **Ontology Modeling:** In the initial phase, a domain-specific ontology named EthOn was developed using Protégé. This ontology captures the structural and transactional elements of the Ethereum blockchain, including blocks, transactions, gas costs, and smart contracts. The ontology aims to semantically enrich Ethereum data structures, enabling context-aware reasoning and facilitating semantic queries. The ontology was formalized in OWL format, ensuring compatibility with semantic web technologies.
- **System Integration:** A technical environment was established to enable communication between the Ethereum blockchain and the ontology system. The environment was built using Node.js as the execution platform, Web3.js to interact with the Ethereum test network (Ganache was used for simulation), Apache Jena Fuseki server to host and query the ontology using SPARQL.
- **Semantic Execution:** In this phase, the EthOn ontology was integrated with a simulated Ethereum environment. Smart contract transactions were annotated with semantic metadata. Custom scripts extracted real-time execution data and stored it in RDF format. Semantic queries (via SPARQL) were used to analyze performance-related metrics, such as: Gas usage per transaction, Block creation latency. The analysis demonstrated that transaction execution was faster and gas consumption was significantly lower when using the ontology-enhanced model. These findings underscore the potential of semantic technologies, such as ontologies, to improve the efficiency and performance of blockchain systems.

Table 1: Comparative Analysis of Related Works and EthOn Framework

Study / Framework	Focus Area	Ontology Integration	Impact on Smart Contract Execution	Performance Optimization	Limitations / Gaps
[14]	Gas analysis and vulnerability	No	Not integrated	Partial	Lacks semantic interpretation
[15]	Code optimization	No	Indirect via code rewriting	Yes	Static logic
[17]	Metadata and trust architecture	Partial (RDF only)	Off-chain interpretation	No	No effect on logic
[19]	Semantic querying	Yes (OWL + SPARQL)	Off-chain only	Limited	Not tied to execution
[20]	Cross-chain interoperability	Partial (abstract layer)	External semantics	Not focused	No Ethereum-specific logic
EthOn (This Study)	Smart contract execution	Full (OWL integrated)	Embedded at runtime	Significant	Validated via implementation

Block Height:	23001623 < >
Status:	Unfinalized
Timestamp:	3 mins ago (Jul-26-2025 06:47:35 AM +UTC)
Gas Used:	34,427,961 (76.51%)  +53% Gas Target
Gas Limit:	44,999,871
Base Fee Per Gas:	0.000000000298391307 ETH (0.298391307 Gwei)
Block Height:	23001624 < >
Status:	Finalized
Timestamp:	20 mins ago (Jul-26-2025 06:47:47 AM +UTC)
Gas Used:	15,009,990 (33.36%)  -33% Gas Target
Gas Limit:	45,000,000
Base Fee Per Gas:	0.000000000318164804 ETH (0.318164804 Gwei)

4. An Analytical and Exploratory Platform for the Ethereum Blockchain

Etherscan is one of the most prominent block explorers dedicated to the Ethereum blockchain, offering a web-based interface that enables users to explore and analyze all blockchain activities, including transactions, blocks, and smart contracts. It is widely regarded as a fundamental tool for understanding the internal structure and operational behavior of the Ethereum network, owing to its transparent and publicly accessible data. Etherscan serves as a reliable research tool in the academic study of Ethereum. It allows researchers to analyze historical data, audit smart contract behavior, and evaluate network resource consumption such as gas. Moreover, it is a valuable educational resource for understanding the internal mechanics of blockchain systems, particularly within the context of Web3 technologies and decentralized finance (DeFi). Its comprehensive data accessibility and transparency make it an indispensable component of blockchain research methodologies. The below displays Detailed Block Data from Block Number 23001623 and 23001624 on the Ethereum Network via Etherscan Platform.

By displaying the results obtained from blocks 23,001,623 to 23,001,631 on the Ethereum network indicate that the block creation process required a relatively notable amount of time. The timestamps ranged between 06:47:35 a.m. and 06:49:11 a.m. on July

26, 2025, with the average block creation time estimated at approximately 12 to 15 seconds. This reflects a relative delay in execution when compared to other, more efficient systems. On the other hand, the gas consumption data exhibited significant variation, ranging from 9,603,890 units to 44,409,521 units, highlighting considerable differences in transaction volume and the complexity of operations within the blocks. These findings emphasize that, although the Ethereum network demonstrates stability, it still faces challenges in terms of execution speed and resource efficiency, particularly under high workloads or when processing complex smart contracts.

5. Proposed of Ethereum Ontology Model (EthOn)

The primary purpose of the Ethereum Ontology (EthOn) framework is to provide a formal, semantic representation of the core components and relationships within the Ethereum blockchain ecosystem. EthOn leverages established semantic web technologies, such as RDF and OWL, to structure blockchain data in a machine-readable and interoperable manner. This ontological framework aims to enable consistent interpretation of Ethereum data across diverse systems and platforms, thereby enhancing interoperability and integration with external datasets. Moreover, by representing Ethereum concepts—such as accounts, smart contracts, transactions, and blocks—in an explicit

and structured way, EthOn facilitates knowledge sharing, automated reasoning, and semantic querying. These capabilities are critical for improving transparency, enabling intelligent analytics, and supporting applications that rely on a precise understanding of blockchain data. EthOn also contributes to bridging the gap between blockchain technologies and the Semantic Web, making Ethereum data more accessible and usable in linked data environments[22].

5.1 EthOn ontology Representation

To semantically model the core components of the Ethereum blockchain, this study adopts the EthOn ontology, which provides a formal, machine-readable representation of Ethereum’s data structures and operational semantics. Developed by ConsenSys, EthOn is grounded in the Ethereum Yellow Paper and is implemented using RDF and OWL standards to capture entities such as blocks, transactions, accounts, and world states in a structured and interoperable format. EthOn enables a clear distinction between syntactic constructs (e.g., raw data fields) and semantic abstractions (e.g., state changes and account ownership), thereby facilitating knowledge reasoning and data integration. Its extensible design includes dedicated modules such as the Contracts Extension and the ERC-20 Extension, which support the modeling of smart contracts, events, and token standards. By integrating EthOn into this research framework, the study ensures semantic precision, enhances the transparency of transaction representation, and supports intelligent querying and reasoning over blockchain data.

5.1.1 EthOn Account Concept Representation

Figure 3 illustrates the EthOn conceptual model of the Ethereum account structure, showcasing the ontological relationships between key entities and attributes. At the core of this model is the Account class, which is subclassed into three distinct types: External Account, Contract Account, and Protocol Account. Each account is uniquely identified by an address (hexBinary), and external accounts are associated with an accountPublicKey, while contract accounts store both accountCodeHash and accountCode. The Account class is linked to an Account State, representing the dynamic aspects of the account, including accountNonce and accountBalance, both defined as integers. The state also connects to an Account Storage component through the hasAccountStorage property, which stores a storageRoot (hexBinary) value. Furthermore, the Account State is associated with the World State as part of Ethereum’s global ledger, establishing the current state context through hasCurrentState and partOf relations. Additionally, the Protocol Account links to Protocol State via the hasState property, indicating special system-level accounts. The External Actor, subclassed from the FOAF ontology’s Agent, exercises control over the External Account using the controls property, thereby embedding social semantics into the blockchain framework. This ontological representation enhances semantic clarity and formal reasoning within Ethereum’s ecosystem.

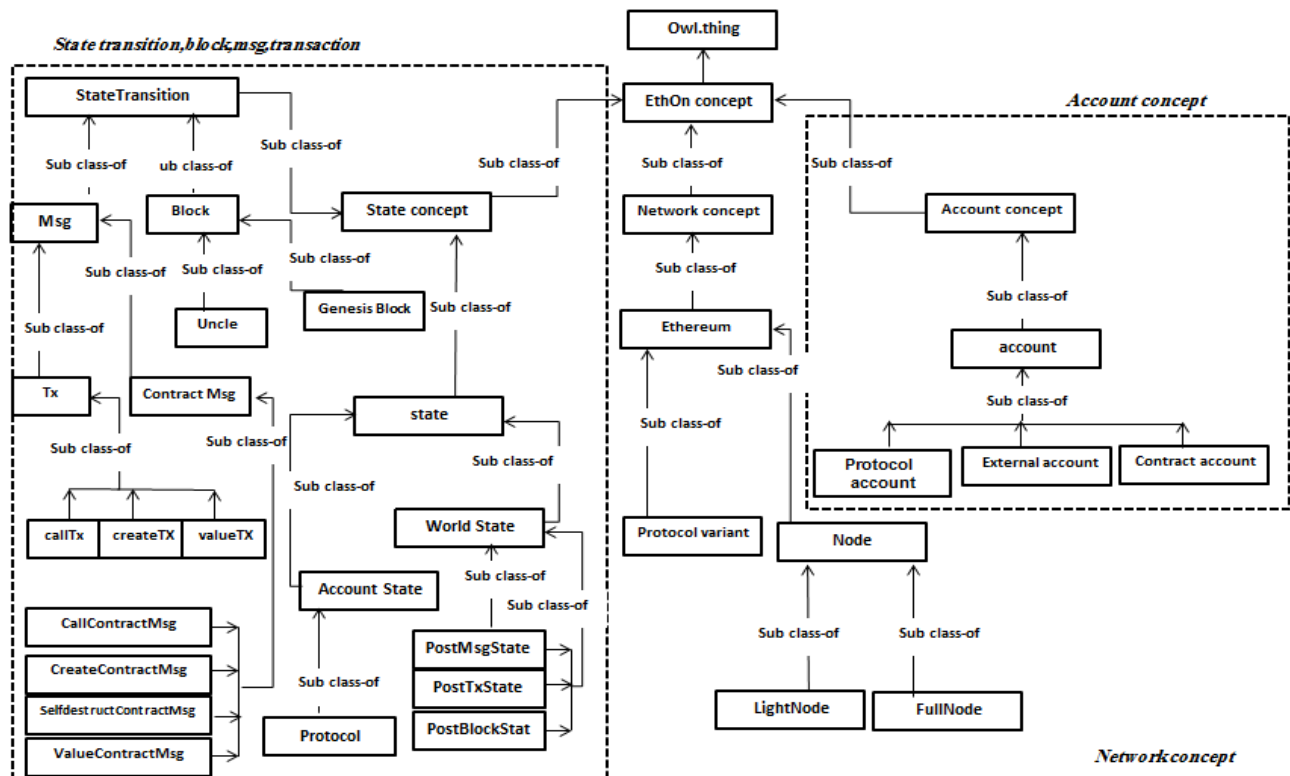


Fig. 1: Illustrate EthON Ethereum Ontology

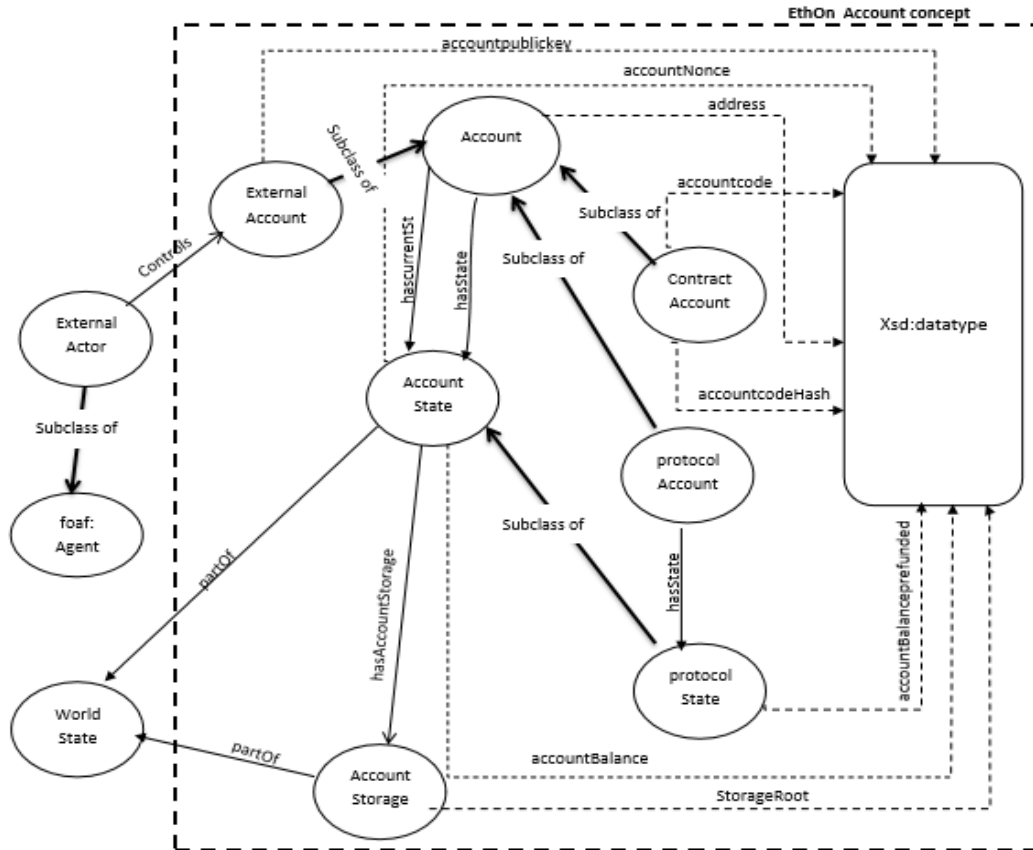


Fig. 2: illustrates the EthON account structure

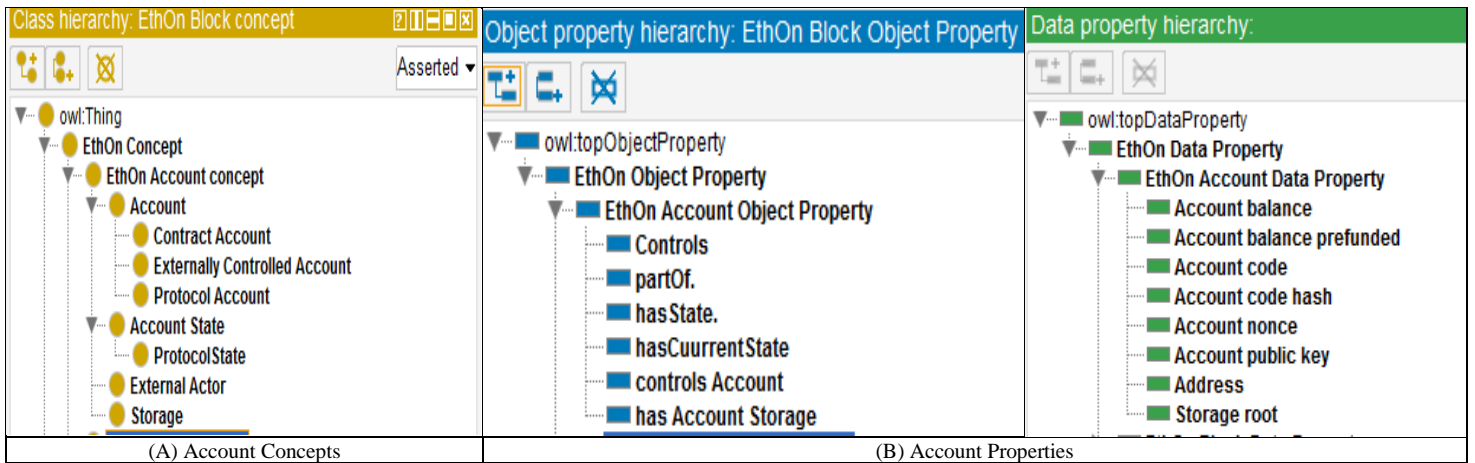


Fig. 3: illustrates the Entities of account Ontology

The figure 3 illustrates the structure of the EthON Account Ontology. (A) Account Concepts: Shows the class hierarchy where EthON Account concept is a main class containing subclasses such as Account, Contract Account, Externally Controlled Account, Protocol Account, Account State, ProtocolState, External Actor, and Storage. This represents how account-related concepts are organized in the ontology. (B) Account Properties: Object Property Hierarchy (middle): Defines relationships between classes.

5.1.2 State Transition concept and modeling scheme

Figure 4. illustrates a formal ontology-based model—EthON—designed to represent state transitions at varying levels of granularity within the Ethereum blockchain. It delineates how messages, transactions, and blocks each generate new states that collectively evolve the global state space (WorldState).

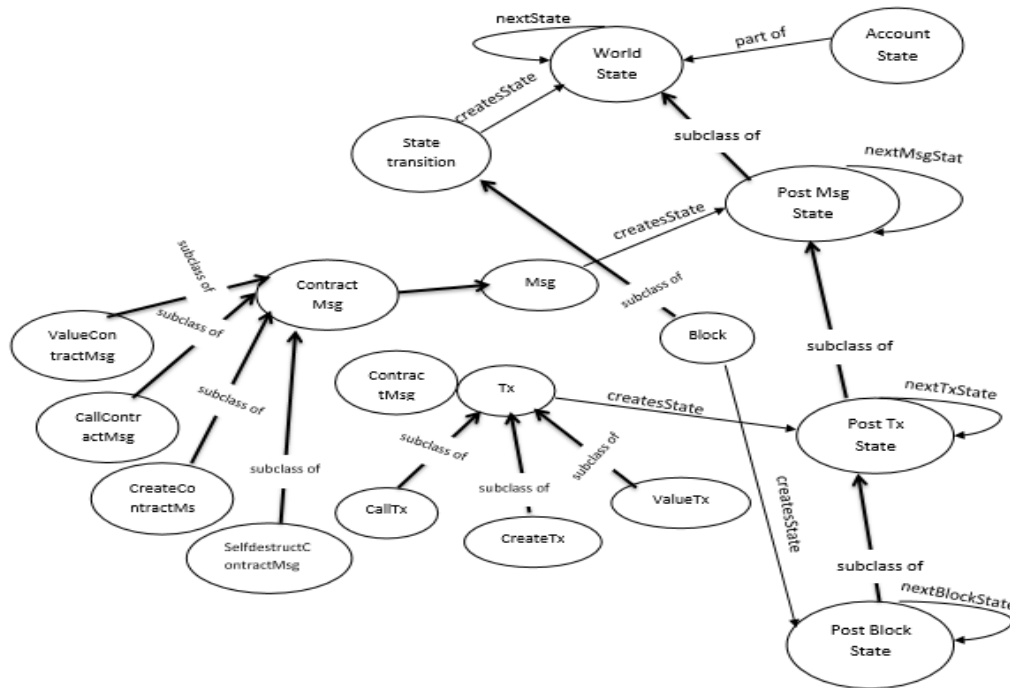


Fig. 4: illustrates the EthOn State Transition structure

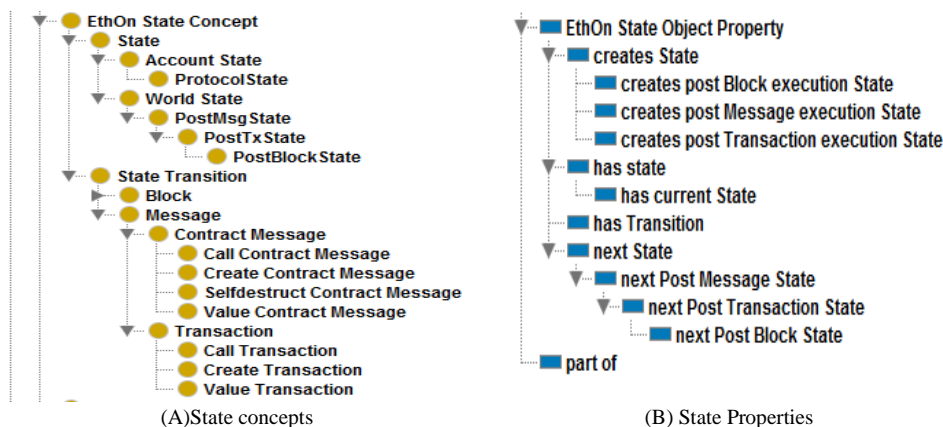


Fig. 5: illustrates the Entities of state transition Ontology

The figure 5 illustration the structure of the ethon state ontology: Part (A) represents the main classes such as State and State Transition, along with their subclasses. For example, State includes subclasses like Account State, World State, PostBlockState, etc. Similarly, Message includes a subclass Contract Message, which further branches into types like Call, Create, Selfdestruct, and Value Contract Message. Part (B) shows the object properties that define relationships between the entities, such as: has state, creates state, has transition, next state, etc. This structure helps in understanding how states and transitions are semantically modeled in Ethereum’s execution process.

5.1.3 EthOn block Concept Representation

A block in the Ethereum network is the fundamental data unit used to aggregate a group of validated transactions along with essential metadata related to network state and

execution history. Each block contains critical information such as the block number, timestamp, hash of the previous block, and Merkle roots for the transaction, receipt, and state tries (e.g., stateRoot, txRoot). It also includes the list of executed transactions during that block’s lifetime. Blocks form a chronological and cryptographically linked sequence known as the blockchain, where each block references its predecessor to ensure data integrity and ordering. Additionally, blocks record miner rewards and may include information about uncle blocks if applicable. Within the EthOn ontology, a block is modeled as a central entity connected to a set of semantic properties and structural relations, enabling automated reasoning and systematic analysis of blockchain data. The figure 6 represents a semantic (ontological) model of the block structure within the Ethereum network, as described by the EthOn ontology framework.

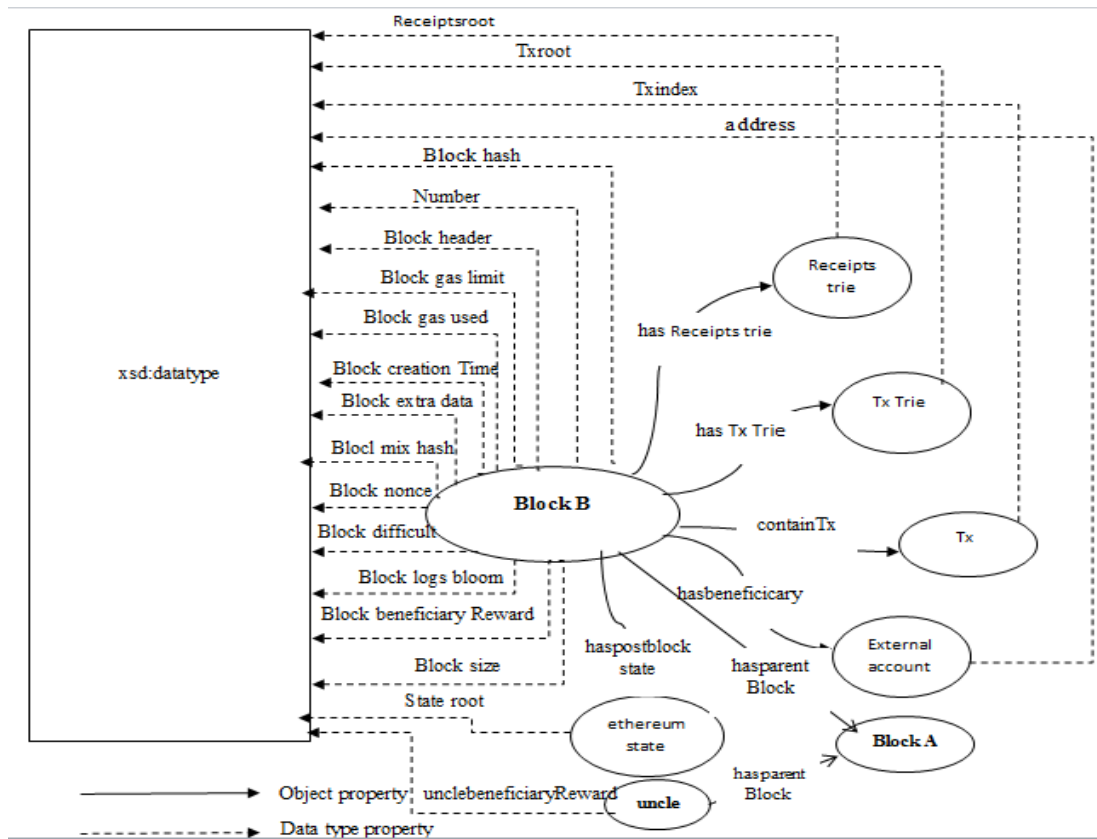


Fig. 6: illustrates the EthOn Block structure

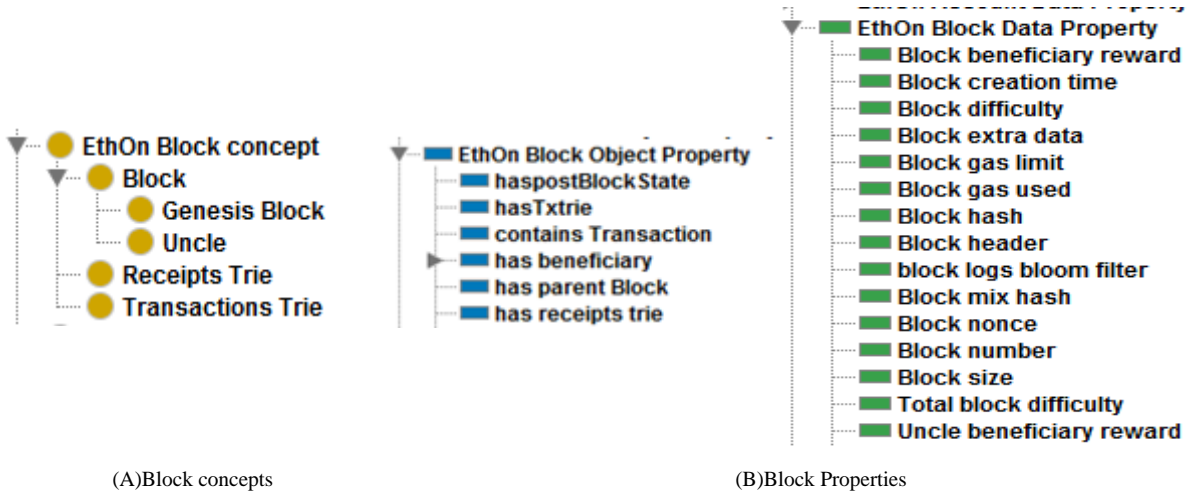


Fig. 7: illustrates the Entities of state transition Ontology

5.1.4 Message, Transaction Receipt and log concepts.

The Figure 8 represents a formal conceptual model from the EthOn Ontology, which semantically describes the structure and behavior of Ethereum blockchain components. It specifically focuses on messages, transactions, and account interactions, using RDF/OWL-based ontology modeling. The visualization adheres to semantic web principles, capturing relationships between core entities in Ethereum.

This figure 9 illustrates the Entities of Message Ontology in Ethereum. In section A the main class is Message which includes Contract Message with

subclasses like Call Contract Message, Create Contract Message, Selfdestruct Contract Message, and Value Contract Message, as well as Transaction which includes Call Transaction, Create Transaction, and Value Transaction, and also Receipts Trie which contains Transaction Receipt. In section B the properties are divided into Object Properties which define relationships such as hasreceipt, containstx, controls, signs, triggersmsg, from, to, and creates, and Data Properties which include MsgGasUsed, value, and payload. This structure shows how messages and transactions are represented and linked within the Ethereum ontology model using defined object and data properties.

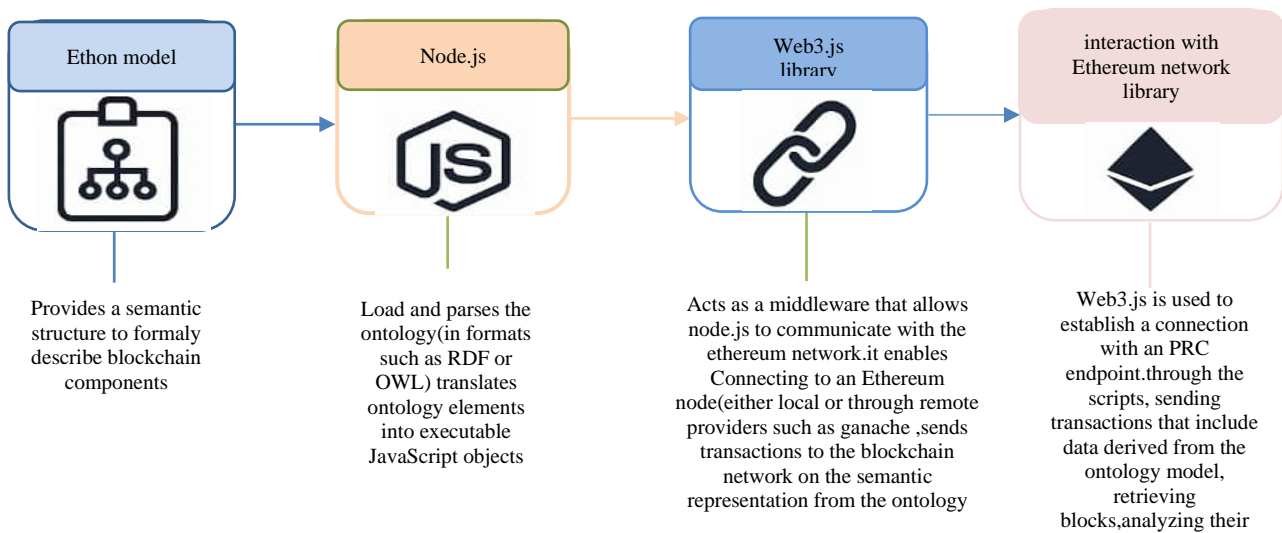


Fig. 10: illustrates the workflow for integrating the EthOn model with the Ethereum network using the Node.js environment and the web3.js library

6. Ontology-based integration of EthOn with Ethereum using Node.js and Web3.js

This section presents a detailed ontology-driven framework for integrating the EthOn ontology with the Ethereum blockchain using Node.js and Web3.js. EthOn serves as a semantic model that describes the core entities and relationships in Ethereum, including accounts, transactions, blocks, and smart contracts. The goal is to bridge semantic modeling with decentralized infrastructures, thereby enhancing data interpretability and interaction intelligence within blockchain-based applications. The integration process involves four main components: the EthOn ontology model, Node.js as the execution environment, Web3.js as the Ethereum interface library, and the Ethereum blockchain itself. The workflow enables semantically enriched blockchain interactions and intelligent transaction processing based on ontology concepts.

Figure 10 illustrates the workflow of integrating the EthOn ontology model with the Ethereum blockchain using Node.js and Web3.js. This process highlights how ontology-based semantic representations can be effectively utilized to enhance blockchain interaction and transaction processing.

- **EthOn Model:** The EthOn model provides a semantic structure that formally describes Ethereum blockchain components. By relying on ontology principles, it ensures a well-defined and standardized representation of elements such as transactions, blocks, accounts, and smart contracts. This semantic foundation enables a higher level of abstraction that facilitates reasoning, interoperability, and reusability across blockchain-based applications.
- **Node.js:** Acting as the execution environment, Node.js is responsible for loading and parsing the

ontology (commonly encoded in RDF or OWL formats). Through its JavaScript runtime, it translates ontology concepts into executable JavaScript objects, making them operable within blockchain applications. Node.js thus bridges the gap between the abstract semantic definitions of EthOn and the practical implementation of logic required for blockchain transactions.

- **Web3.js:** Web3.js functions as a middleware layer that enables Node.js applications to interact seamlessly with the Ethereum network. It establishes the communication channel between the local execution environment and Ethereum nodes (either local instances such as Ganache or remote providers). By doing so, it supports sending ontology-driven transactions to the blockchain, thereby ensuring that semantic representations are effectively translated into actionable operations on the Ethereum ledger.
- **Interaction with the Ethereum Network:** Finally, Web3.js facilitates direct interaction with the Ethereum blockchain by connecting to RPC endpoints. This includes sending transactions derived from ontology-based data, retrieving blockchain states such as blocks and transaction receipts, and analyzing their contents. The ontology-driven integration ensures that all data exchanged aligns with the semantic definitions provided by EthOn, resulting in a more structured and meaningful interaction with the network. The Local Ethereum Blockchain Simulation environment was initialized Using Ganache CLI, In the experimental environment of this research, a local Ethereum blockchain network was instantiated using Ganache CLI (version 7.9.2). Ganache is a widely used personal blockchain simulator that enables developers and researchers to emulate the Ethereum network locally for the purposes of testing,

debugging, and performance evaluation, without incurring transaction fees or interacting with the public Ethereum mainnet.

6.1 Ontology-Based Representation of Ethereum Transactions in EthOn

Within the EthOn model, Ethereum transactions are semantically represented through ontology concepts that capture their structural and operational properties. In the figure 11 show demonstrates how each transaction is linked to its corresponding block using the relation ethon:containsTx, while attributes such as ethon:txHash, ethon:txNonce, ethon:gasPrice, and ethon:gasUsed define its computational and economic characteristics. By formalizing these elements into machine-readable knowledge entities, EthOn enables more than just raw storage—it facilitates advanced reasoning, consistency validation, and cross-domain integration. This semantic enrichment also supports performance improvements: transactions can be classified, queried, and optimized based on their ontology-defined features, thereby reducing execution time and minimizing gas consumption. The EthOn-based representation transforms Ethereum from a purely technical ledger into

a knowledge-driven system, paving the way for more intelligent blockchain infrastructures that enhance both efficiency and scalability.

6.2 Ontology-Based Representation of Ethereum Blocks in EthOn

The semantic description of Ethereum blocks through the EthOn ontology allows for a structured and knowledge-driven representation of blockchain data. In the figure 12, a block is explicitly defined as ethon:Block, while its essential attributes—such as ethon:number, ethon:blockHash, ethon:stateRoot, ethon:transactionsRoot, and ethon:hasParentBlock—are represented as machine-readable properties. This formalization transforms a block from being just a raw record on the blockchain into a knowledge entity with explicit semantics. Such a model supports automated reasoning, facilitates consistency checks, and enhances interoperability with external datasets. Moreover, the ontology-based approach allows for more advanced querying and intelligent analytics, enabling researchers and developers to uncover insights about Ethereum’s state transitions and transaction structures in a more efficient and meaningful way.

```

ibb:1 ethon:containsTx tx:f6d954908a0025efa6b631f7afde2981ae4522edeb6d5e6abb885c288528ba99 .
tx:f6d954908a0025efa6b631f7afde2981ae4522edeb6d5e6abb885c288528ba99 ethon:cumulativeGasUsed "21000"^^xsd:integer .
tx:f6d954908a0025efa6b631f7afde2981ae4522edeb6d5e6abb885c288528ba99 ethon:txHash "f6d954908a0025efa6b631f7afde2981ae4522e
tx:f6d954908a0025efa6b631f7afde2981ae4522edeb6d5e6abb885c288528ba99 ethon:txNonce "65628"^^xsd:integer .
tx:f6d954908a0025efa6b631f7afde2981ae4522edeb6d5e6abb885c288528ba99 ethon:txIndex "0"^^xsd:integer .
tx:f6d954908a0025efa6b631f7afde2981ae4522edeb6d5e6abb885c288528ba99 ethon:txGasPrice "100000000"^^xsd:integer .
tx:f6d954908a0025efa6b631f7afde2981ae4522edeb6d5e6abb885c288528ba99 ethon:txGasUsed "11044"^^xsd:integer .
    
```

Fig. 11: Snapshot of the semantic representation of transactions data

```

ibb:1 a ethon:Block.
ibb:1 ethon:number "1"^^xsd:integer .
ibb:1 ethon:blockHash "cebe6edeb7b1f38c730d2a1e0724cc82aae74ae78cbb82976d31efb9766eeeba"^^xsd:hexBinary .
ibb:1 ethon:hasParentBlock ibb:59516bf9df41d021b8e1b0744091aa5e534ff11b84c0824caabf12abe63bbc86 .
ibb:1 ethon:block_nonce "af164dc005769243"^^xsd:hexBinary .
ibb:1 ethon:knowsOfUncle ibu:1dccc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347 .
ibb:1 ethon:blockLogsBloom "80000038200000000000000040040000000004083380402c08203010000000012000401100004100400011000000
ibb:1 ethon:hasTxTrie ibtx:ff01d8e941bf8e09b73d1a8f3b71ae6ed2537f875229e749890fa21acedea366 .
ibb:1 ethon:hasPostBlockState ibs:62807c4799c883a27893e9f7dae54c83e7b882ac736b83c5abd194ca6a01882c .
ibs:62807c4799c883a27893e9f7dae54c83e7b882ac736b83c5abd194ca6a01882c ethon:stateRoot "62807c4799c883a27893e9f7dae54c83e7b88
ibb:1 ethon:blockDifficulty "2162920971648890"^^xsd:hexBinary .
ibb:1 ethon:blockSize "31256"^^xsd:integer .
ibb:1 ethon:blockExtraData "e4b883e5bda9e7a59ee4bb99e9b1bc"^^xsd:hexBinary .
ibb:1 ethon:blockGasLimit "8003730"^^xsd:integer .
ibb:1 ethon:blockGasUsed "709104"^^xsd:integer .
ibb:1 ethon:blockCreationTime "2025-4-22T13:37:00".
    
```

Fig. 12: Snapshot of the semantic representation of blocks data


```

===== Summary =====
Total Gas Used: 212040
Total Gas Cost (ETH): 0.012828
Average Gas Price (Gwei): 60.50
Min Gas Price (Gwei): 32
Max Gas Price (Gwei): 90

```

Fig. 14: Snapshot of the contract Execution result

```

start scan the transaction (Polling)...
new block: #1
new transaction:from: 0xfba71691847b2359aad5ba6a81bb27d7aff8934a
to : 0xbc6f39013e069d65089fd75ac07c7bb45c34974a
the value: 20 ETH
Hash: 0xe61167ae5b52f164a9f78ee9cbb51c2d66c4c7369d513ac1aa65cf3f1a5faa7f
new block: #2
new transaction:from: 0xfba71691847b2359aad5ba6a81bb27d7aff8934a
to : 0xbc6f39013e069d65089fd75ac07c7bb45c34974a
the value: 14 ETH
Hash: 0xf6c1493558db2545671145ba9dcb153df4a8cffb9e6a131965839bb31e6cafc8
new block: #3
new transaction:from: 0xfba71691847b2359aad5ba6a81bb27d7aff8934a
to : 0xbc6f39013e069d65089fd75ac07c7bb45c34974a
the value: 34 ETH
Hash: 0x27fe0f48b02eddbc9759f6a632dedbe44a6fbcce1c7853ee7bca1c14bee7dc1
new block: #4
new transaction:from: 0xfba71691847b2359aad5ba6a81bb27d7aff8934a
to : 0xbc6f39013e069d65089fd75ac07c7bb45c34974a
the value: 55 ETH
Hash: 0x304d6ca12fc0ffb4a229ce06166252d7dc5661ee9072d0e73f47918fd11e63dd
new block: #5
new transaction:from: 0x4b57f5c51ed6603a8e443d93ebc225e6747fdc24
to : 0x95aa57a3fca61164a71c51c865e47f27e8360150
the value: 0 ETH
Hash: 0x8ebd0e034ddb95b54c5712586f045ad98b002159b87e5e7460cdf80c12fe54c6
new block: #6
new transaction:from: 0xdc02c725b4d8122e1f8bed38ec45fd95c55e38b6
to : 0x95aa57a3fca61164a71c51c865e47f27e8360150
the value: 0 ETH
Hash: 0x937e3f7e657c45a0d7ed0fa07871fd07c57499f9e51866e38ed1896687b00c05
new block: #7
new transaction:from: 0x811d795967051ac1e7e9372be449d5c61d4960c8
to : 0x95aa57a3fca61164a71c51c865e47f27e8360150
the value: 0 ETH
Hash: 0xdae861ee57c3ec1f2a45c1e4f2d3c2cf921aacc7689c08a6a1abe6dc269d5eab

```

Fig.15: Snapshot of the monitor Execution result

7.3 Blockchain Transaction and Block Monitoring via Web3.js Polling Mechanism

The primary purpose of the script is to continuously monitor newly mined blocks and extract detailed information about the transactions they contain. This is achieved through a polling mechanism that queries the blockchain node at regular intervals. This script demonstrates an effective polling-based blockchain monitoring approach using Web3.js. Such a mechanism is valuable in blockchain analytics, fraud detection systems, and transaction auditing tools.

7.4 Ontology-Driven Data Retrieval in Ethereum Using SPARQL Queries

Based on the outputs illustrated in Figures 16 and 17, the use of the SPARQL interface enabled the retrieval of precise data from the ontology model linked to the Ethereum network. Figure 16 presents the results of a

query designed to extract both the gas used and gas price for each transaction, providing a practical indicator for assessing resource efficiency during the execution of smart contracts and transactions. In contrast, Figure 17 shows the outcome of another query that focuses on retrieving block creation time, the number of blocks, and the corresponding gas consumption, which facilitates the analysis of the network's temporal performance and supports comparisons between different environments such as the Ethereum Mainnet and the proposed EthOn model. These results demonstrate that integrating ontology with SPARQL significantly enhances the ability to conduct advanced analyses of both resource utilization and execution time, while ensuring access to well-structured and reliable data that supports rigorous academic evaluation of blockchain network performance.

```

1 PREFIX ibb: <http://example.org/block#>
2 PREFIX tx: <http://example.org/transaction#>
3 PREFIX ethon: <http://example.org/ethon#>
4
5 SELECT ?tx ?gasUsed ?gasPrice
6 WHERE {
7   ?block ethon:containsTx ?tx .
8   ?tx ethon:txGasPrice ?gasPrice ;
9     ethon:txGasUsed ?gasUsed .
10
11 }
    
```

tx	gasUsed	gasPrice
tx:8381c1180a42ef43b7caab19905642a3ff2d8232098e4c32ecd0ec199b6acea8	100000	800000000
tx:fd954908a0025efa6b631f7afde2981ae4522edeb6d5e6abb885c288528ba99	11044	100000000
tx:f1d5e70c9769f079412e8d79cbb2e1dda7a956719bc01e15b4143adb91d5660b	90000	800000000
tx:f1d5e70c9769f079412e8d79cbb2e1dda7a956719bc01e15b4143adb91d5660b	86000	800000000
tx:beab23c13faa52b0ec3bc4ee5dd6167209de184c6d7376ebb5cedba8641cebae	21000	100000000
tx:5c0de8e962dcf32af76ba148823a10ae9e389120d01f0a8172b2c4b678cd187	100000	800000000
tx:2d8e7a83ae8be09146fb8afb2a488642964b0d4c434acd8745288fb61ce40e47	21000	400000000
tx:9820d31b16fe320fa04a750afd04eeda7437e14a8b3e2b856ee68e0f2e56ac9e	100000	800000000
tx:84e9a77de601a8dde98eeda25524c43741ef07623d7ae35f09b88fe423392a94	100000	800000000
tx:380bdd0cf1a7ceb355e9a85cd807f611455fa13e5004197a057fe1d37f8e42	900000	800000000

Fig. 16: sparql Interfaces Result of gas cost

```

1 prefix ethon: <http://ethon.consensus.net/>
2
3 prefix ibb: <http://knowledge.interition.net/ethereum/block#>
4 prefix ibs: <http://knowledge.interition.net/ethereum/uncle#>
5 prefix ibu: <http://knowledge.interition.net/ethereum/state#>
6 prefix ibtx: <http://knowledge.interition.net/ethereum/block_tx_root#>
7
8 prefix xsd: <http://www.w3.org/2001/XMLSchema#>
9
10 SELECT ?block ?number ?blockCreationTime ?blockGasUsed
11 WHERE {
12   ?block ethon:number ?number ;
13     ethon:blockHash ?blockHash ;
14     ethon:blockGasUsed ?blockGasUsed;
15     ethon:blockCreationTime ?blockCreationTime .
    
```

block	number	blockCreationTime	blockGasUsed
ibb:8	8	"2025-4-22T13:37:22"	3697660
ibb:5	5	"2025-4-22T13:37:12"	132560
ibb:9	9	"2025-4-22T13:37:23"	1458570
ibb:4	4	"2025-4-22T13:37:09"	2773990
ibb:6	6	"2025-4-22T13:37:13"	245670
ibb:1	1	"2025-4-22T13:37:00"	709104
ibb:3	3	"2025-4-22T13:37:05"	585040
ibb:7	7	"2025-4-22T13:37:15"	2148760
ibb:2	2	"2025-4-22T13:37:01"	423630

Fig. 17: sparql Interfaces Result of block creation time- blockgasused

8. Evaluation

8.1 Reduced Gas Cost

The table 3 illustrates the gas usage per transaction. The results obtained through SPARQL Query 2 show a marked reduction in gas costs, with consumption ranging from 1,000 to 100,000 gas units per transaction. This is a significant improvement when compared to traditional Ethereum transactions, which often consume several 200,000 gas units. The efficiency gain is attributed to the semantically enriched design of smart contracts, which helped eliminate unnecessary computational steps and redundant logic.

8.2 Improved Transaction Execution Speed

As shown in table 4 the ETHON model significantly enhanced the speed of block creation. Based on SPARQL Query 1, which retrieved block creation times using semantic annotations, the average execution time per block was reduced approximately of 1 to 7 second. This marks a substantial improvement over the standard Ethereum network, where block generation typically takes around 10 seconds. The acceleration is primarily attributed to ETHON’s ontology-driven optimization of the transaction flow and block structure.

The table 4 presents the experimental results obtained from executing the EthOn model on the Ethereum test network. It displays a sequence of nine consecutively generated blocks, documenting both the exact timestamp of each block's creation and the amount of gas consumed during its processing.

8.3 Evaluation of Results from EthOn Model versus Ethereum Main Network

In table 5, Block creation time and gas consumption are among the key performance metrics in blockchain systems, as they directly influence transaction throughput and execution efficiency. In this study, we compare the performance between the Ethereum mainnet

and the proposed ontology-based framework EthOn using nine experimental blocks.

- **Block Creation Time Analysis**

The results presents in table 6 show that the proposed EthOn framework achieved a significant reduction in block creation time compared to the Ethereum mainnet. The block creation time in EthOn remains nearly constant at a few seconds, while the mainnet shows substantially longer intervals.

This improvement leads to reduced waiting time for transaction confirmation and enhanced responsiveness for decentralized applications.

Table 3: illustrates the results of gas usage per transaction

TX	Gas used
tx:8381c1180a42ef43b7caab19905642a3ff2d8232098e4c32ecd0ec199b6acea8	11044
tx:f6d954908a0025efa6b631f7afde2981ae4522edeb6d5e6abb885c288528ba99	21000
tx:f1d5e70c9769f079412e8d79cbb2e1dda7a956719bc01e15b4143adb91d5660b	100000
tx:f1d5e70c9769f079412e8d79cbb2e1dda7a956719bc01e15b4143adb91d5660b	100000
tx:beab23c13faa52b0ec3bc4ee5dd6167209de184c6d7376ebb5cedba8641cebae	21000
tx:5c0de8e962dfc32af76ba148823a10ae9e389120d01f0a8172b2c4b678cd187	110000
tx:2dbe7a83ae8be09146fb8afb2a488642964b0d4c434acd8745288fb61ce40e47	90000

Table 4: illustrates the results of generated blocks in ganache using ETHON model

Block No.	Block creation Time	Block Gas Used
Block 1	2025-4-22T13:37:00	709,104
Block 2	2025-4-22T13:37:01	423,630
Block 3	2025-4-22T13:37:05	585,040
Block 4	2025-4-22T13:37:09	2,773,990
Block 5	2025-4-22T13:37:12	132,560
Block 6	2025-4-22T13:37:13	245,670
Block 7	2025-4-22T13:37:15	2,148,760
Block 8	2025-4-22T13:37:22	3,697,660
Block 9	2025-4-22T13:37:23	1,458,570

Table 5: illustrate Block Gas, creation time Consumption:EthOn vs Ethereum mainnet

Block No.	Ethereum gas used(Ethon)	Ethereum mainnet gas used	Block creation Time (EthOn)	Block creation Time (Ethereum mainnet)
Block 1	709,104	34,427,961	2025-4-22T13:37:00	26-2025 06:47:35 AM
Block 2	423,630	15,009,990	2025-4-22T13:37:01	26-2025 06:47:47 AM
Block 3	585,040	17,935,450	2025-4-22T13:37:05	26-2025 06:47:59 AM
Block 4	2,773,990	18,260,575	2025-4-22T13:37:09	26-2025 06:48:11 AM
Block 5	132,560	18,101,948	2025-4-22T13:37:12	26-2025 06:48:23 AM
Block 6	245,670	21,589,234	2025-4-22T13:37:13	26-2025 06:48:35 AM
Block 7	2,148,760	15,758,011	2025-4-22T13:37:15	26-2025 06:48:47 AM
Block 8	3,697,660	44,409,524	2025-4-22T13:37:22	26-2025 06:48:59 AM
Block 9	1,458,570	9,603,890	2025-4-22T13:37:23	26-2025 06:49:11 AM

Table 6: Comparison of Block Creation Time.

Metric	Ethereum Mainnet	Current Study (EthOn)
Block Creation Time	Several seconds up to noticeable delay	Consistently a few seconds
Blocks Created per Minute	Lower due to latency	Much higher frequency

• *Gas Consumption Analysis*

Gas consumption is a vital economic performance indicator, representing the computational resources required to execute transactions and smart contracts. The data obtained from nine blocks demonstrates a substantial reduction in gas usage by EthOn compared to the Ethereum mainnet:

Minimum Gas Consumption in EthOn: 132,560 units

Maximum Gas Consumption in EthOn: 3,697,660 units

Minimum Gas Consumption in Mainnet: 9,603,890 units

Maximum Gas Consumption in Mainnet: 44,409,524 units

Table 7: Comparison of Gas Consumption

Metric	Ethereum Mainnet	Current Study (EthOn)
Minimum Gas Used	9,603,890 units	132,560 units (~98.6% improvement)
Maximum Gas Used	44,409,524 units	3,697,660 units (~91.7% improvement)

• *Average Gas Consumption Calculation*

The average gas consumption for each environment is computed using the midpoint formula:

$$\text{Average Gas} = (\text{Minimum Value} + \text{Maximum Value}) / 2$$

$$\text{Ethereum Mainnet: } (9,603,890 + 44,409,524) / 2 = 27,006,707$$

$$\text{EthOn: } (132,560 + 3,697,660) / 2 = 1,915,110$$

$$\text{Efficiency Gain: } ((27,006,707 - 1,915,110) / 27,006,707) \times 100 \approx 92.91\%$$

The results indicate that the proposed EthOn framework achieved: A significant improvement in block creation time, ensuring faster transaction processing and better responsiveness for decentralized applications, A drastic reduction in gas consumption (approximately 93%), leading to lower transaction costs and improved economic efficiency. These findings validate the effectiveness of integrating ontology-based enhancements into blockchain design, reinforcing the scalability and sustainability of smart contract execution.

It is important to note that the number of transactions and smart contracts also differs between the two environments: In EthOn: Each block contains approximately 10 to 20 transactions and includes two smart contracts. In Ethereum Mainnet: Each block contains approximately 50 to 100 transactions and

includes between one and two smart contracts. The evaluation highlights that EthOn achieves remarkable improvements in gas efficiency and block creation speed compared to the Ethereum mainnet. While results from Table 4 indicate the theoretical performance potential of ontology-based optimization, the real-world data in Table 4 demonstrate the practical challenges of scalability and network congestion. Bridging this gap requires deploying EthOn in more extensive testbeds or layer-2 environments to validate its applicability under realistic blockchain conditions.

Figure 18 illustrates the comparison of block creation time between the proposed EthOn framework and the Ethereum mainnet across nine consecutive blocks. The plotted lines demonstrate that EthOn maintains a consistently stable block creation time within a narrow range of seconds, highlighting its efficiency and reliability. In contrast, the Ethereum mainnet shows a noticeable increase in block creation time, with values progressing from 06:47:35 to 06:49:11, reflecting longer confirmation delays. This clear divergence indicates that EthOn achieves faster and more predictable block generation, which directly contributes to higher transaction throughput and reduced latency for decentralized applications. Figure 19 illustrates a comparative analysis of gas consumption between the EthOn model and the Ethereum Mainnet across multiple blocks. The red curve demonstrates that gas usage in EthOn is relatively low and stable, ranging from approximately 132,560 to 3,697,660 gas units, reflecting the framework’s efficiency in minimizing the computational resources required for executing transactions and smart contracts. In contrast, the green curve representing the Ethereum Mainnet shows significant fluctuations and substantially higher values, ranging between 9,603,890 and 44,409,524 gas units, highlighting the high execution cost in the main network. This visual disparity between the two curves underscores the effectiveness of EthOn in improving the economic efficiency of blockchain execution and substantially reducing transaction costs compared to the conventional Ethereum network. Consequently, the findings reinforce the importance of integrating ontology into blockchain architecture, as it enhances scalability through optimized resource utilization, while simultaneously ensuring cost reduction, thereby providing a more sustainable and practical foundation for large-scale decentralized applications.

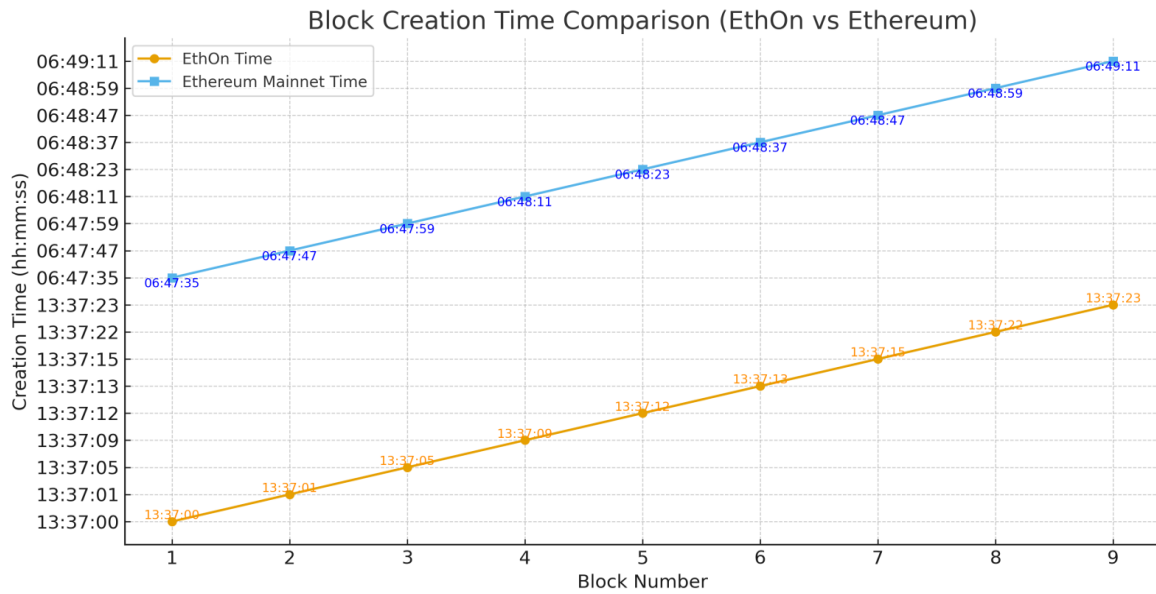


Fig. 18: illustrate block creation time comparison:EthOn vs Ethereum mainnet

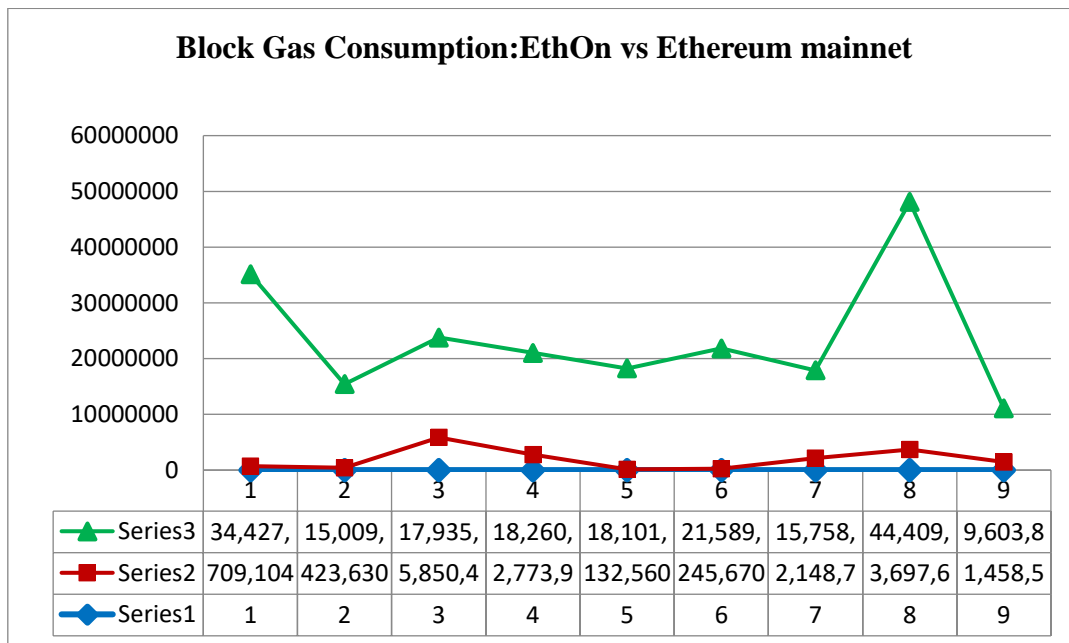


Fig. 19: illustrate Block Gas Consumption:EthOn vs Ethereum mainnet

9. Conclusion and Recommendation

This study has demonstrated that the proposed model significantly improves the performance of the Ethereum network by leveraging the semantic representation of core blockchain elements. Through the integration of ontologies into the smart contract environment, the model enabled faster transaction execution and reduced gas consumption—addressing some of the most persistent inefficiencies in current blockchain infrastructures. In addition, the model exhibited notable flexibility and scalability, indicating its potential for adaptation to other blockchain platforms with comparable architectural needs. In light of these results, it is recommended that ontologies be applied to a broader

range of blockchain-based applications, including digital identity systems, electronic voting mechanisms, and supply chain monitoring. Such domains can greatly benefit from the enhanced semantic clarity and interoperability offered by ontological modeling. To facilitate wider adoption, there is also a clear need for the development of supportive tools and frameworks that streamline the integration of ontologies into smart contract development environments. Finally, the study encourages further academic and technical inquiry into the convergence of semantic web technologies and blockchain systems, as this emerging field holds considerable potential for building more intelligent, efficient, and adaptable decentralized systems.

References

- [1] V. Buterin, "Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform," 2014.
- [2] G. Wood, Ethereum: A Secure Decentralised Generalised Transaction Ledger, Ethereum Yellow Paper, 2019.
- [3] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," *Telecommunication Systems*, vol. 70, no. 3, pp. 325–374, 2019.
- [4] W. Wang et al., "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.
- [5] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.
- [6] G. Antoniou and F. van Harmelen, *A Semantic Web Primer*. MIT Press, 2004.
- [7] N. Shadbolt, W. Hall, and T. Berners-Lee, "The Semantic Web revisited," *IEEE Intelligent Systems*, vol. 21, no. 3, pp. 96–101, 2006.
- [8] M. A. Musen, "The Protégé project: A look back and a look forward," *AI Matters*, vol. 1, no. 4, pp. 4–12, 2015.
- [9] J. J. Carroll, C. Bizer, P. Hayes, and P. Stickler, "Named graphs," *Journal of Web Semantics*, vol. 3, no. 4, pp. 247–267, 2004.
- [10] T. Nakamura, S. Fujimura, and F. Ishikawa, "Ontology-based smart contract framework for cross-domain applications," *Proc. IEEE Int. Conf. Blockchain*, 2020, pp. 123–130.
- [11] S. Zander, J. Barzen, and F. Leymann, "Semantic blockchain: Combining blockchain and semantic web technologies for improved data interoperability," *Future Generation Computer Systems*, vol. 115, pp. 607–618, 2021.
- [12] G. Wood, Ethereum Project Yellow Paper, 2014.
- [13] V. Buterin, "Ethereum 2.0 and sharding roadmap," Ethereum Research Forum, 2017.
- [14] L. Luu et al., "Making smart contracts smarter," *Proc. ACM SIGSAC Conf. Computer and Communications Security*, 2016, pp. 254–269.
- [15] T. Chen, X. Li, X. Luo, and X. Zhang, "Gas reduction for Ethereum smart contracts," *IEEE Trans. Software Eng.*, vol. 48, no. 3, pp. 732–747, 2020.
- [16] S. Wang, Y. Zhang, J. Xie, H. Wang, and X. Lin, "A Survey of Layer-2 Scaling Solutions for Ethereum," *IEEE Access*, vol. 9, pp. 161526–161545, 2021.
- [17] S. Grüner, A. Mühle, and C. Meinel, "A semantic blockchain architecture for data quality assessment and trustworthiness," *Proc. IEEE Int. Conf. Internet of Things (iThings)*, 2018, pp. 1472–1479.
- [18] M. Turkanović, M. Hölbl, and K. Košič, "EduCTX: A blockchain-based higher education credit platform," *IEEE Access*, vol. 6, pp. 5112–5127, 2018.
- [19] T. Rakhmawati, N. Sabouret, and D. Benslimane, "Semantic blockchains: Architecture and application challenges," *Future Generation Computer Systems*, vol. 108, pp. 781–792, 2020.
- [20] E. Rodríguez-Tello, J. L. Sánchez-Cervantes, and J. Cervantes, "Blockchain interoperability: A comprehensive review," *ACM Computing Surveys*, vol. 54, no. 11, pp. 1–34, 2021.
- [21] Y. El Faqir, A. Abou El Kalam, and M. El Marraki, "Ontology-based data classification model for blockchain," *Int. J. Network Security*, vol. 22, no. 4, pp. 645–653, 2020.
- [22] A. Rejeb, J. G. Keogh, and H. Treiblmaier, "Leveraging the power of blockchain and semantic technologies in supply chain management," *J. Theor. Appl. Electron. Commerce Res.*, vol. 16, no. 3, pp. 1002–1024, 2021.
- [23] J. Pfeffer, "EthOn — introducing semantic Ethereum," *ConsenSys Media*, Medium, 2018.

تعزيز كفاءة عمليات سلسلة كتل إيثيريوم بالاعتماد على الأنطولوجيا

وضاح أحمد منصر^{1*}، و نورا ناصر حمصان¹¹ قسم تقنية المعلومات، كلية الهندسة، جامعة عدن، اليمن* الباحث الممثل: وضاح أحمد منصر؛ البريد الإلكتروني: wmunassar@yahoo.com & noonhumssan1990@yahoo.com

استلم في: 04 مارس 2026 / قبل في: 25 مارس 2026 / نشر في 31 مارس 2026

المُلخَص

تقدم هذه الدراسة إطار عمل قائمًا على الأنطولوجيا، يُعرف باسم EthOn، يهدف إلى تعزيز الكفاءة التشغيلية لسلسلة كتل إيثيريوم. وعلى الرغم من بروز إيثيريوم كمنصة لامركزية لتنفيذ العقود الذكية، إلا أنه لا يزال يواجه قيودًا كبيرة مثل ارتفاع رسوم الغاز، وانخفاض معدل المعالجة، ومحدودية قدرات المعالجة الدلالية. تنشأ هذه التحديات بشكل أساسي من اعتماد المنصة على هياكل بيانات تركيبيّة (نحوية) وغياب الآليات واعية بالسياق داخل العقود الذكية. يقوم الإطار المقترح بدمج الأنطولوجيات الدلالية لتنظيم بيانات سلسلة الكتل وتعزيز منطق العقود الذكية من خلال الاستدلال المعتمد على السياق. ومن خلال تضمين أنطولوجيات خاصة بالمجال ضمن طبقة إيثيريوم، يتيح EthOn تفسيرًا ذكيًا للبيانات، وتنفيذًا مُحسنًا للعقود، وتحسين قابلية التشغيل البيئي للتطبيقات اللامركزية (dApps). تم تطوير نموذج أولي باستخدام أدوات Protégé وNode.js وWeb3.js وشبكة Ganache وApache Jena، حيث أظهر تقدمًا ملحوظًا في كل من سرعة التنفيذ وكفاءة استهلاك الغاز. وأظهرت النتائج تحسينات كبيرة: حيث تطلب إيثيريوم زمنًا متوسطًا لإنشاء الكتلة يتراوح بين 10-15 ثانية مع استهلاك غاز يتذبذب بين 9.6 مليون و44.4 مليون وحدة، بينما حقق EthOn توليد الكتل خلال 1-2 ثانية مع الحفاظ على استهلاك الغاز بين 3.7 و423,630 مليون وحدة. ويمثل ذلك انخفاضًا إجماليًا في استهلاك الغاز يقارب 90% وتحسنًا كبيرًا في سرعة التنفيذ. تُبرز هذه النتائج الإمكانيات التحويلية للتقنيات الدلالية في إعادة تشكيل بنية سلاسل الكتل، مما يمهد الطريق نحو أنظمة لامركزية أكثر قابلية للتوسع وكفاءة وذكاءً.

الكلمات المفتاحية: سلسلة الكتل؛ إيثيريوم؛ الأنطولوجيا؛ EthOn؛ سعر الغاز؛ الويب الدلالي؛ تنفيذ المعاملات.

How to cite this article:

W. A. Munassar, and N. N. Humssan, "ENHANCING THE EFFICIENCY OF ETHEREUM BLOCKCHAIN OPERATIONS BASED ON ONTOLOGY", *Electron. J. Univ. Aden Basic Appl. Sci.*, vol. 7, no. 1, pp. 90-108, Mar. 2026. DOI: <https://doi.org/10.47372/ejua-ba.2026.1.503>



Copyright © 2026 by the Author(s). Licensee EJUA, Aden, Yemen. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY-NC 4.0) license.