International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

COPY RIGHT

2026 IJIEMR. Personal use of this material is permitted. Permission from IJIEMR must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. No Reprint should be done to this paper; all copy right is authenticated to Paper Authors

IJIEMR Transactions, online available on 7th June 2026. Link
https://ijiemr.org/downloads.php?vol=Volume-15& issue=issue06

DOI: 10.48047/IJIEMR/V15/ISSUE 06/179

Title Blockchain Ontology for Streamlining Application Development using Semantic Web Rule Language
Volume 15, ISSUE 06, Pages: – 1772 - 1786

Paper Authors
V Sitharamulu[1], Bhuvan Unhelkar[2], Prasun Chakrabarti[3], Mohammed Ali Hussain[4]

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper as Per UGC Guidelines We Are Providing A Electronic Bar code

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Blockchain Ontology for Streamlining Application Development using Semantic Web Rule Language

V Sitharamulu[1], Bhuvan Unhelkar[2], Prasun Chakrabarti[3], Mohammed Ali Hussain[4]
[1,2]University of South Florida
[3]Department of Computer Science and Engineering, Sir Padampat Singhania University, Udaipur Rajasthan lndia.
[4]Professor, Dept. of CSE, Sreenidhi Institute of Science and Technology, Hyderabad, India. Email: vsitaramu.1234@gmail.com, bunhelkar@usf.edu, drprasun.cse@gmail.com, alihussain.phd@gmail.com

Abstract

Decentralised applications (DApps) allow for extensive and trusted functions via blockchain infrastructure. These applications have various services like decentralized storage, transaction scalability protocols and distributed computing solutions. To develop reliable tools by integration of DApps, the primary aim is to propose a novel ontology of the blockchain based on EthOn. This paper introduces basic concepts on creation of the ontology that are associated with the DApp. Then elucidates about the development of links that are associated with them. The work provides the literature on existing application with the architecture of blockchain through blockchain patterns. To meet the primary aim of the work, Semantic Web Rule language (SWRL) is utilised that showcase rules which act as restrictions to address the integration of DApps. Thus, using an inference engine, the work determines novel restrictions on properties like cost and other characteristic metrics. For instance, the work illustrates constraint inference of the concept between Ethereum and polygon, which is a sidechain of Ethereum. In conclusion, this work contributes on developing applications on video games which are built on blockchain technology. This work also helps in understanding the ontology in modelling DApps. It has potential contributions in applications with several related areas in future works.

KEYWORDS: Ontology, DApps, Decentralized applications, Blockchain, Semantic Web Rule language

Introduction

Blockchains are considered as a revolutionary technology that support Decentralized Applications (DApps) which are characterized by many features such as transaction openness, application traceability, and impossibility of censorship [1]. They have wide applications in different sectors like financial area [2], health [3], video games [4]. The emerging domains of newer applications have also changed drastically from 2008 when Bitcoin started.

However, current research focus light on some challenges like interim interoperability of blockchains [5]. Interoperability could not be more important for the blockchain industry for many organizations, platforms, projects, and services through blockchain integration.

For this reason, blockchain interoperability is categorized in three main types. The first is mind-introduction or the relationship between two or more blockchain systems or

platforms- just like the relationship between the Bitcoin and Ethereum networks, whose objective is to allow different systems to pass information and value to one another. The second is the internal interoperability which can be defined as projects that are aimed at one system of block chain. For example, projects based on Ethereum may require the use of some general interfaces and protocols and simultaneously the Ethereum Requests for Comments system to improve integration. The third form is the interoperability of services between decentralized applications or commonly known as DApps. Many decentralized applications have different components with distributed storage and scalability solutions. To meet the desired purpose, all the components in the distributed storage need to communicate each other. Proven solutions are required to enhance the blockchain sector in today market.[6].

Since the concept of DApps and their interactions remain unstructured and undefined, let the paper demonstrate below the proposed ontology, which categorizes the various elements of DApps. This involves the establishment of a structure corresponding to the ideas regarding the types of services which defines the operations of DApps. Based on the literature, the ontology developed for this study extends from the blockchain ontology commonly known as EthOn [7].

The ontology is enhanced by applying it for various common blockchain designs found in decentralized applications. These designs include NFT management apps, NFT trading platforms, DAOs, blockchain oracles, supply chain management, and non-blocking user interfaces.

- The idea utilizes SWRL rubrics to ensure the coherence of our DApps ontology. These rules govern the concepts within the ontology duly used to infer new axioms.
- Demonstrate how the DApps ontology and SWRL rules can be used to improve interaction between two blockchain systems.
- Based on these contributions, the paper illustrates the Light_Trail_Rush (LTR) manufacturing video game.

Initially, the literature reviews on blockchain ontologies and then skin to the work to be done. The work presents a new type of ontology which has been specially developed to provide the formal definition of DApps and express blockchain patterns [8]. Moreover, the approach the annotation rules in the SWRL. Finally, we discuss the use of objectives of my study in the video game industry and present the outcomes. Lastly, there is a recap of the conclusion as well as the discussion of the outlook for our research.

Latest Advances and Innovations A. Present State of Semantic Interoperability:

Interoperability within the blockchain industry remains challenging due to differing definitions of key concepts such as "finality." For instance, in PoW (used by Bitcoin), finality is probabilistic, leading to potential invalidation of blocks, known as eventual certainty. On the other hand, Proof of Authority (PoA) ensures OP Consistency validating validated blocks irreversible. These differences complicate interoperability, though some stochastic approaches allow limited interaction between blockchains with varying consensus mechanisms.

Smart contracts also play a significant role in this context. Projects like BLONDiE and Ethereum

Ontology aim to create comprehensive blockchain ontologies, but these efforts are still tentative and do not fully support semantically compatible blockchains. **Figure 1** pictures a basic blockchain ontology, modelled using OWL to define concepts and relationships in Ethereum. The Sandra platform by Ever dream soft uses OWL to manage NFTs and game assets, facilitating the design of blockchain ontologies.

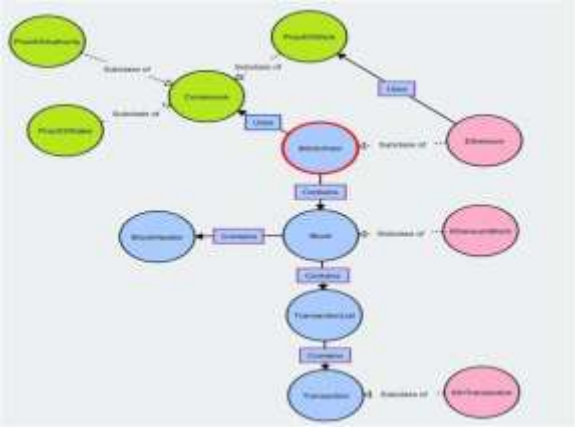


Figure 1. A blockchain ontology Example.

B. Blockchain Services (BS):

Blockchain Services are enhancements developed and implemented as separate blockchain projects, aimed at improving functionalities such as either transaction support or data storage.

Scaling Factor: Several approaches exist so far to enhance the scalability of blockchain systems and often implemented as Layer 2 solutions that work on top of other existing blockchains.

State-Channels: Under event disputes, statechannels specifically allow peer-to-peer transactions and are conducted in off-chain mode with the option to post the final state on-chain. This method reduces the load on the blockchain for increased scalability. Fun Fair, for instance, uses state-channels between two parties (as depicted in **Figure 2**). Broader applications include the Lightning Network for Bitcoin, Raiden Network for Ethereum, and Counterfactual (**Figure 3**), though further studies are needed to determine the most effective approach for payment transactions.

Rollups: Rollups are another scalability solution that aggregates blockchain transactions by batching them or storing data off-chain while maintaining some on-chain elements. Figure 4 illustrates the two primary categories of rollups: Zero-Knowledge Rollups (ZK-Rollups) and Optimistic Rollups.

- **ZK-Rollups:** Zero-Knowledge Proofs (ZKP) emerged to compress transaction data. The transaction data is also verified during the compression process. Applications like Loopring and ZKSync employ ZK-Rollups to enhance the throughput of decentralized exchanges and DApps.
- **Optimistic Rollups:** These assume all transactions are valid unless challenged, In contrast to ZK-Rollups, transactions are invalid until proven valid.

Validium: StarkEx, a ZKP-based scalability solution developed by StarkWare, operates in both on-chain and off-chain modes are used to avoid data availability issues through Data Availability Committee console. Competitors like DeversiFi use Validium opposing ZKRollups.

Sidechains: Sidechains are separate set of blockchains connected to the mainchain via a two-way peg. These are often used in specific applications such as video games with notable examples including Loom Network and Polygon (**Figure 5**). Sidechains can freely scale due to their loose coupling with the mainchain.

Polygon: Polygon formerly known as Matic. Polygon is a bypassing technique for handling fraud proofs connected to Ethereum. The process

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

is being managed by issuance of token based system to efficiently handle them.

Loom: Loom is initially related to Ethereum and virtual video games. Loom has expanded into a multi-chain platform in due course of time. It has proven as a significant sidechain solution within the Ethereum ecosystem.

Multi-Chain Ecosystems: Multi-chain ecosystems enable interaction among different blockchains. These include Polkadot and Cosmos that work different validation models. Polkadot utilizes a Relay Chain model for synchronized validation across shards, while Cosmos use a system of hubs to manage shared communication.

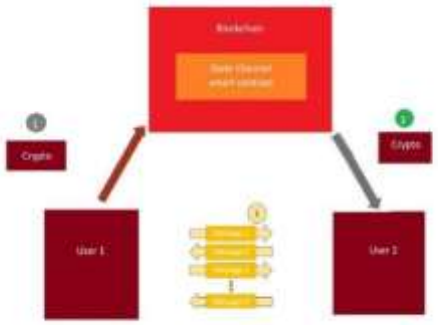


Figure 2. state channel's function

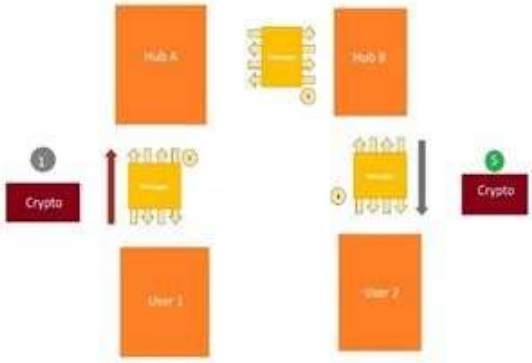


Figure 3. Lightning Network and the Raiden Network operate

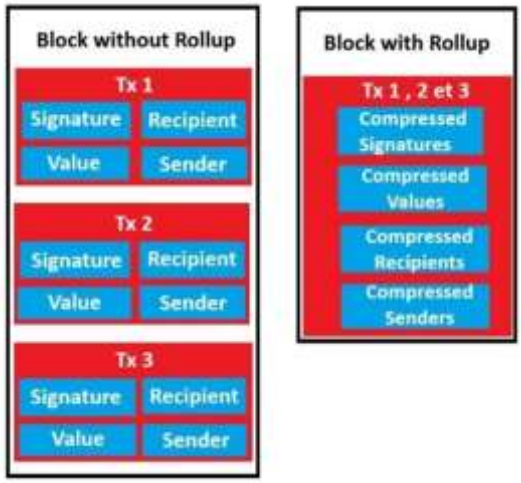


Figure 4. Rollups function

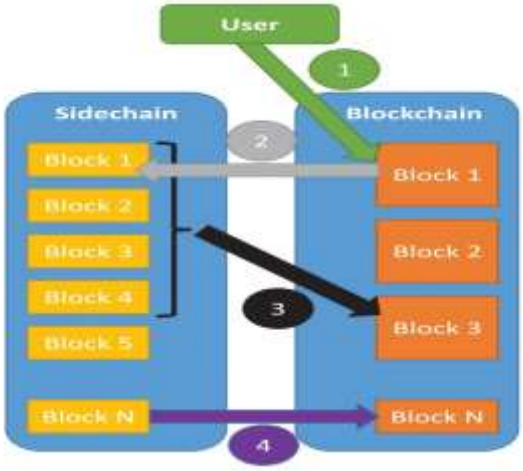


Figure 5. Sidechains operation module.

Data Storage:

a. **Inter Planetary File System (IPFS):** IPFS is an open-source distributed file system permitting users to self-host nodes and sharing of data is noticed by unique hashes. **Figure 6** elucidates how IPFS users request resources using hash values and forming a mesh network to locate and transmit the desired data. While IPFS is free and secure, it has fewer limitations, such as the potential unavailability of resources due to the lack of financial incentives for sharing storage and high latency.

b. **File coin:** File coin is built on top of IPFS and offers financial incentives to participants who lease out their decentralized storage space on the blockchain.

c. **Storj:** Storj is an alternative decentralized storage solution used as a distributed cloud. Unlike IPFS or File coin, Storj replicates, clusters, and encrypts data for enhanced security and availability.

Figure 6. Representation of multichain operation.

Figure 7. Functioning module of IPFS.

PROPOSED ONTOLOGIES

The paper proposed the DApp concepts within an ontology. The observation includes the introduction of generic blockchain ontology, which is specialized for DApps.
The work implements the use of Decentralized Blockchain Applications (DBA) to distinguish them from other decentralized apps.

A. High-level overview of the ontology

Figure. 8 illustrates part of the generated ontology using the OntoGraph Plugin for Protégé. The user signed messages attached on the blocks are considered as unique transactions over all block chain elements. Transactions can be in process (verifying) or completed (verified or unverified). Once validated, reversing or deleting of transactions is a risky approach. The approach depicts blockchain concepts, inherited by Polygon and Ethereum.

The work focussed on the existing two blockchain ontologies: BLONDiE and EthOn. The new ontology development requires extending of these existing models.

EthOn encovers Ethereum blockchain constructs, transactions, blocks, contract calls, data structures and their periodic updates. The scope of work also extended EthOn to include essential concepts for representing DApps.

BLONDiE, on the other hand, incorporates blocks from Ethereum, Bitcoin, and Hyperledger Fabric and lacks details on transaction implementation for each of individual blockchain. Although it includes Ethereum smart contract calls and creation, but could not specify data structures. The limitation in BLONDiE is it cannot store transactions and state changes. Thus, our the scope of our work is not extended to BLONDiE and used it to analyze integrating concepts from various blockchains into a single ontology. Some of the notations used in the scope of study is:

- **ethon:** Ontology imported from EthOn.
- **gbo:** Systematizes general blockchain concepts. generic part of the ontology, developed using OntoGraph in Protégé.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

- **sbo:** Formalizes concepts for the DApp context. DApp-specific segment.
- **bpo:** Codifies blockchain patterns and Figure. 9 illustrates various blockchain patterns with usage examples.

Figure 8. Transactions and Blockchains proposed in the study

Figure 9. Blockchain Patterns in Proposed Ontology.

B. Classes

This subsection provides an overview of the major classes used in the ontology for the proposed work. Each class relates to blockchain or DApps, and is defined by annotations, constraints, and examples. Constraints are associated with Ontology concepts using object properties (ObjectProperties) or data properties (Data Properties). Below are examples of concepts and annotations formalized in the proposed blockchain ontology extension.

1 GBO: Blockchain - Per NIST, a blockchain is an append-only, shared record with strong immutability control of transactional information [9]. Constraints include Consensus Method (gbo:AConsensusMechanism), Blocks (gbo:Block), average block time in seconds, block size (bytes), Smart Contracts (Absent/Present), termination time (seconds) for confirmation and complexity for logistics are most commonly observed.

2 GBO: Transaction – All Awaited and validated transactions are grouped as a block and reside in a memory buffer called mempool. Examples include Coinbase Transaction- pays the miner for each block and allows money creation, and Polygon Checkpoints a sidechain checkpoint on Ethereum that secures Polygon transactions [10].

3 GBO: Blockchain User - A blockchain user is an account holder or participant in the ecosystem, accessed via platforms like Meta mask or as a node. Users can be regular, DApp users, cryptocurrency owners, or miners. PoS stakers are compensated, while miners generate new blocks in PoW chains.

4 GBO: Consensus Mechanism - A consensus mechanism solves the Byzantine Generals Problem [11], defining block and transaction validity

in the network. Examples include Proof of Work, Proof of Stake, and Proof of Authority.

5 GBO: Smart Contract- A smart contract, used by DApps, facilitates decentralized value transfer and can act as a record-keeping or ID verification system. As described by Szabo in 1997 and formalized by Buterin in 2014, smart contracts are authoritative and immutable. Examples include Ethereum and Hyperledger Fabric [12], with other DLTs like Tangle and Hash graph supporting similar features [13].

6 SBO: Decentralized Blockchain Application - A DApp operates on blockchain principles and may require additional services such as an interface engine, storage, or scaling solutions. SWRL rules define DApp properties based on employed technologies. Constraints include using a gbo:Blockchain, at least one gbo:Blueprint, an engine (sbo:Department of engines), a sbo:LayerTwo solution, a sbo:Solution for Storage, transaction volume, running costs, operating latency, and factors affecting data transmission.

SBO: Engine- Engines are frameworks for building DApp interface and can be web pages, game engines, or application frameworks.

SBO: Layer Two - Examples include ZeroKnowledge Rollups, Polygon and Validium.

SBO: Storage - Examples include IPFS, Central Storage, and File coin.

GBO: Polygon Checkpoint- A Polygon checkpoint is an Ethereum transaction confirming Polygon actions. Checkpoints are submitted for every thirty minutes to sync Polygon with Ethereum. Constraints include an array of Polygon transactions approved by the checkpoint ID from the transaction's hash. Classes in our defined ontology are detailed in the appendix section.

C. OBJECT PROPERTIES

Object Properties define relationships between concepts in the ontology, focusing on blockchain data structures and the functional use of actors and services.

GBO: CONTAINS	This relationship indicates how a blockchain is composed of blocks, and blocks are composed of transactions. It describes the hierarchical structure from blocks to transactions
GBO: INCLUDEDIN	The inverse of gbo: Contains, showing that when a block is created, it is incorporated into a blockchain.
GBO: USES	A many-to-many relationship indicating that actors or technologies depend on each other. For instance, a blockchain user uses a blockchain, and a DApp relies on a blockchain for its operation.
GBO: NEXTBLOCK	Points to the subsequent block being mined in a blockchain at any given time.
GBO: LASTCHECKPOINT	Refers to the latest checkpoint in a blockchain system, such as Ethereum Polygon, which updates checkpoints on Ethereum periodically
GBO: MINESFOR	Connects a blockchain user to the blockchain they mine, applicable only to blockchains using Proof of Work (PoW).
GBO: HASBENEFICIARY	Links a block in a blockchain to the miner who created it, identifying the recipient of the block's benefits.

D. Individuals by Class

In this paper, the authors specify the individuals necessary to complete a DApp for applying SWRL rules, and some of them are Data Administrator, State Manager, FORM Manager, and Rule Enforcer.

GBO:BLOCKCHAIN	
SBO:_POLYGONMAINNET	Represents the primary Polygon blockchain, distinct from the Polygon POS chain.
SBO:_ETHEREUMMAINNET	This individual represents the main Ethereum blockchain, in its current version (ETH 1.0).
SBO:_ETHEREUMBRANCHNTENTIONNET	A test version of Ethereum (Ropsten), which utilizes PoW behavior but runs non-Passthrough Ropsten ETH for testing DApp behavior.
GBO:LAYERTWO	
SBO:_POLYGONLAYERTWO	A scalability solution for Ethereum with distinct properties
SBO:_NULLLAYERTWO	Representing a DApp lacking Layer Two, used in SWRL rules
SBO:DECENTRALIZEDBLOCKCHAINAPPLICATIONS	
SBO:_DBA1	This individual represents a DApp using only Ethereum, with no scalability solution in Layer Two.
SBO:_DBA2	This individual represents a DApp using Ethereum, as well as its sidechain Polygon.
SBO:_DBA1	which utilizes Polygon alone, differing from its use as a sidechain to Ethereum.
SBO:POLYGON TRANSACTION	We have also denoted two individuals that represent two successive Polygon checkpoints.
SBO:_POLYGONCHECKPOINT1	This transaction is submitted to the Ethereum blockchain
SBO:_POLYGONCHECKPOINT2	

E. Data Properties:

Data properties connect classes with simple data like integers, strings and dates, essential for SWRL rules in section V. They are

categorized into costs, transaction counts, data quantities, and latency. Costs include general properties related to developer user costs, price per megabyte of storage, transaction costs, and costs for token transfers and smart contract deployments. Transaction counts track the number of transactions needed, distinguishing between developer and user operations. Data quantities reflect the amount of data required for applications differentiating between developer and user storage and on-chain versus off-chain storage. The Object Property section shows how to implement use cases using design patterns.

SIMULATING DESIGN PATTERNS OF BLOCKCHAIN

To better appreciate and understand ontology across numerous use cases, the authors modelled common blockchain design patterns. This approach follows Xu et al. [14], applying blockchain patterns to analyse design patterns in the industry.

A. NFT Management DApp

Numerous DApps are developed to manage NFTs, typically represented by ERC-721 tokens. For example, the NFT management DApp built by B2Expand creates new NFTs from a particular contract's state. Projects like Mintable [15] go further by offering Gasless minting, where buyers cover contract state changes, thus eliminating the cost burden on sellers. Mintable also provides a contract for instant sales, which, though more costly per use, offers flexible selling periods. An example of such a DApp is CryptoKitties where players interact with virtual cats as NFTs that can be traded, sold, or cloned, with each attributes affecting its value. A notable issue in NFT DApps is the centralized storage of images linked to NFTs. CryptoKitties, an NFT management application, uses a centralized system for metadata storage and its storage solution is applied to all DApps and is further represented by **sbo.**

DeployContractForEachNFT	Defines whether to deploya smart contract for each NFT or not
GaslessMinting	Defines whether the creation of NFTs is free or not.

B. NFT Marketplace

NFT marketplaces, such as OpenSea [16], facilitate NFT transactions between buyers and sellers. These platforms often involve sellers who are companies or secondary market users, resulting in high NFT creation and sale volumes. To distinguish users, we categorize them as buyers or sellers. The Data Properties specific to NFT marketplaces are:

User Buyer	
NbTxPerUserBuyer	Number of transactions each buyer performs.
NbTxPerUserBuyerOnMainchain	Transactions each buyer performs on the main blockchain
NbTxPerUserBuyerOnLayerTwo	Transactions each buyer performs on Layer Two
NbMbPerUserBuyer	Amount of data each buyer needs to store.
NbMbPerUserBuyerOnMainchain	Data stored by each buyer on the main blockchain.
NbMbPerUserBuyerOnLayerTwo	Data stored by each buyer on Layer Two.
NbMbPerUserBuyerOnStorage	Data stored by each buyer on the storage service.
User Seller	
NbTxPerUserSeller	Number of transactions each seller performs
NbTxPerUserSellerOnMainchain	Transactions each seller performs on the main blockchain
NbTxPerUserSellerOnLayerTwo	Transactions each seller performs on Layer Two.
NbMbPerUserSeller	Amount of data each seller needs to store.
NbMbPerUserSellerOnMainchain	Data stored by each seller on the main blockchain
NbMbPerUserSellerOnLayerTwo	Data stored by each seller on Layer Two
NbMbPerUserSellerOnStorage	Data stored by each seller on the storage service

These Data Properties with detail transaction costs and data storage from Section III, are tailored for the differing interactions of buyers and sellers with the application.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

C. Decentralized Autonomous Organizations (DAOs)

DAOs [17] operate like companies where tokens act as shares, granting holders governance rights over the DAO. This governance includes updates to the application and alterations to its properties. Maker DAO is a notable example, managing the Dai [18] stablecoin. Through the Maker application, token holders can vote on various parameters, such as adjusting loan rates and maintaining protocol stability. Figure 10 illustrates Maker DAO's governance interactions, showing how MKR token holders influence different modules of the DApp, including system shutdowns in case of faults or market distortions.

Our ontology treats DAOs similarly to NFT Marketplaces by categorizing users into two types: Owners- who make key project decisions, and Investors- who use the DAO's services without direct involvement.

For DAOs, the Data Properties include for governance users and target users.

NbTxPerUserForGovernance	Number of transactions each governance user needs to execute
NbTxPerUserForGovernanceOnMainchain	Number of transactions each governance user needs to execute on the main blockchain
NbTxPerUserForGovernanceOnLayerTwo	Number of transactions each governance user needs to execute on the Layer Two platform
NbMbPerUserForGovernance	Amount of data/amount of data each governance user needs to store
NbMbPerUserForGovernanceOnMainchain	Amount of data each governance user needs to store on the main blockchain
NbMbPerUserForGovernanceOnLayerTwo	Amount of data each governance user needs to store on the Layer Two platform
NbMbPerUserForGovernanceOnStorage	Amount of data each governance user needs to store on the Layer Two platform
TargetUser: Properties related to a target user of theDAO	
NbTxPerTargetUser	Number of transactions each target user needs to execute
NbTxPerTargetUserOnMainchain	Number of transactions each target user needs to execute on the main blockchain
NbTxPerTargetUserOnLayerTwo	Number of transactions each target user needs to execute on the Layer Two platform
NbMbPerTargetUser	Amount of data each target user needs
NbMbPerTargetUserOnMainchain	Amount of data each target user needs to store on the main blockchain
NbMbPerTargetUserOnLayerTwo	Amount of data each target user needs to store on the Layer Two platform
NbMbPerTargetUserOnStorage	Amount of data each target user needs to store on the storage service

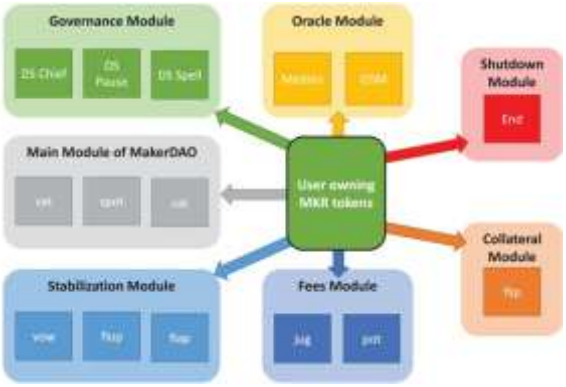


Figure 10. Governance user's influencing various MakerDAO services.

D Oracles

Oracles are essential for integrating external data into blockchains. They ensure that decentralized applications (DApps) have reliable data sources, which are crucial for maintaining security and functionality. For example, MakerDAO relies on oracles to provide up-to-date information on exchange rates which is critical for managing loans and collaterals. Figure 11 shows Omnia Relay and oracle service collects real-time price data from various exchanges via APIs. Then transmits this information to MakerDAO contracts. Access to Omnia's functions is controlled through a whitelist of addresses, and smart contracts use these inputs to determine final price values. Although the proposed ontology models define interactions between DApps and oracles, it does not address the associated costs and processing times.

E Supply Chains

Blockchain technology provides substantial advantages for managing supply chains, through transaction traceability. This is useful for tracking products down to individual items, with each item having a unique reference number. For instance, a furniture retailer might have many items with the same model but each with a distinct serial

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

number. The Data Properties for supply chain DApps include the following:

NbTxPerProduct	Number of transactions for each product
NbTxPerProductOnMainchain	Number of transactions for each product on the main blockchain
NbTxPerProductOnLayerTwo	Number of transactions for each product on the Layer Two platform.
NbMBPerProduct	Account of data to store for each product
NbMBPerProductOnMainchain	Account of data to store for each product on the main blockchain
NbMBPerProductOnLayerTwo	Account of data to store for each product on the Layer Two platform
NbMBPerProductOnStorage	Account of data to store for each product on the storage service

Reference:

As in the case of supply chain DApps, the pattern is very close to section 3. However, instead of the user type, the usage of the application is considered depending on whether there is only one reference for many products or many references for fewer products. For example, a seller having 100,000 products under a single reference will naturally have different requirement than the seller with 50 products in 25 references. The latter requires advanced search and filtering capabilities in their smart contracts to allow Interaction with buyers.

F. Non-Blocking User Interfaces

Event-based interaction as pointed out by Ojha [19] is widely used in blockchain. This pattern makes sure that a user can be able to perform multiple actions and the completion of one action does not have to wait for the other.

Implementation in Ontology

In relation to the actions of the system, the authors defined the ontology based on parallel activities or sequences. For instance, in a DApp for creating and selling NFTs, it is possible to upload features in terms of an image file and defining changes in the description of the NFT at single point of time. However, once another user wishes to purchase this NFT, they need to wait for the authorization before they use it, so the purchase becomes a blocking operation.

The Data Properties specific to DApps with non-blocking user interfaces are:

ActionQueueSize	Total number of options that a user can perform
ActionCriticalQueueSize	This is the maximum sequence of user actions that are allowed at a specific instance.

G. Synthesis on: Blockchain Patterns

Concerning each of the blockchain design patterns described above, the Data Properties useful for their modelling have been determined. Therefore, makes it possible for the development of a model that would fit any application within our ontology with these Data Properties. The authors could not find themselves in a necessity to introduce new abstractions of the kind of Object Properties or introduce new, broader concepts to accommodate the top-level patterns of blockchain.

In order to use these properties for the definition of concrete application's operating characteristics, it is essential to write SWRL rules. While it is difficult to write SWRL rules that cover each pattern due to the specific nature of the applications, the set rules in the ontology may be modified by researchers to include their specific application, thus giving a well-covered modelling solution.

Figure 11. Example of an Oracle, inspired from https://docs.makerdao.com/smart-contractmodules/oracle-module.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

SEMANTIC WEB RULE LANGUAGE (SWRL)

A. SWRL Axioms Definitions SWRL [20] is a language used for defining rules on OWL ontologies, allowing inference machines to derive new axioms from existing ones. These rules specify conditions on individuals and their relationships, with axioms in the antecedent implying those in the consequent. Our blockchain ontology includes 17 SWRL rules. Rules 1, 1bis, and 2 focus on blockchain fundamentals, ensuring proper formalization. Rules 3 to 6 address interactions between Ethereum and its sidechain, Polygon. The remaining ten rules (A to D4) specify DApp requirements, including cost and latency characteristics based on DApp properties and services.

a. Rule 1

Any transaction included in a valid block is itself valid. Listing defines this with three individuals: a transaction (? x), list of transactions (? list), and a valid block (? b). Lines 4 and 5 describe the inclusion of ? x in? list within? b. Line 7 specifies that if all antecedent axioms are valid, then the transaction? x is valid. The rule is illustrated using the Protégé plugin Aided OWL Notation (AOWLN) [21] in Figure. 12.

b. Rule 1BIS

If a block contains an invalid transaction, the block is considered invalid. This rule complements Rule 1. Lines 1 to 3 define a failed transaction (? x), a transaction list (? list), and a block (? b). Lines 4 and 5 describe their inclusion. Line 7 concludes that the block is invalid if it contains? x. Figure. 13 shows the rule's graphical representation via AOWLN.

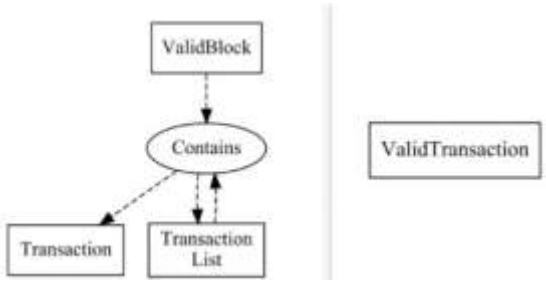


Figure 12.
Rule 1, Antecedent on the left and the consequent on the right

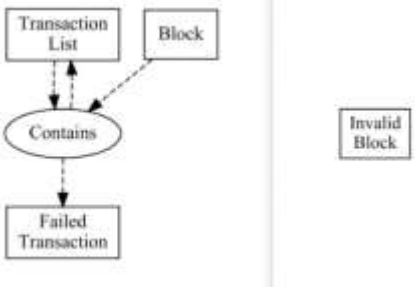


Figure 13.

Rule 1bis: the sequence of properties for the antecedent is shown on the left, and the consequent is on the right.

c. Rule 2

A valid block in a PoW blockchain must be mined by a participant on that blockchain. Lines 1 to 4 define a blockchain (? blockchain), a valid block (? b), proof of work (?pow), and a blockchain account (?x). Line 5 confirms PoW usage, while lines 6 and 7 state that the block (? b) is mined by account (? x) on the blockchain (? blockchain).

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

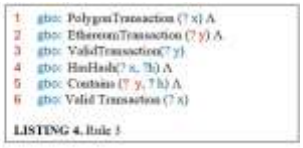
d. Rule 3

If a Polygon transaction is included in a valid Ethereum transaction by its hash, it is also valid. Lines 1 to 3 define Polygon and Ethereum transactions and their validity. Line 4 requires that an Ethereum transaction contains the hash of a Polygon transaction.

Line 7 infers the validity of the transaction. **e. Rule 4**

A Polygon transaction is valid or invalid based on its timing relative to a one-week period. Lines 1 to 4 describe a time-bound Polygon transaction. Lines 5 to 10 use SWRL built-ins to compare the submission time of? x with the last Polygon block time (? z) and a one-week delay. Lines 8 to 12 establish validity based on this time difference.

LISTING 3. Rule 2



LISTING 4. Rule 3

f. Rule 5

The Polygon sidechain updates the Ethereum mainchain periodically by adding Merkel roots of new blocks. This rule suggests including a Polygon checkpoint in the next Ethereum block, though it's optional. The initial lines describe the Polygon blockchain and the timing of checkpoints. The following lines compare many times to decide if a new checkpoint is necessary. If the last checkpoint is older than 30 minutes, a new one is added.

g. Rule 6

A Polygon transaction submitted within 30 minutes is included in the Polygon checkpoint. Listing states that Polygon transactions approved by smart contracts are part of Ethereum checkpoints. The initial lines define the Polygon transaction (? tx) and its timestamp (? x). The following lines ensure the transaction is not older than 30 minutes and describe its relationship with the checkpoint.

h. Specific Rule A

Rule A calculates developer costs based on transactions, data storage, and technologies used. defining a DApp, its blockchain (? blockchain), transaction scalability solution (?l2), and storage solution (?storage), the DApp's properties and cost calculation, the formula is:

$$\begin{aligned}&\text{CostDevPerUser}\\&= \text{NbTxDevPerUserOnMainchain} \times \text{CostTx}_{\text{blockchain}}\\&+ \text{NbTxDevPerUserOnLayerTwo} \times \text{CostTx}_{\text{LayerTwo}}\\&+ \text{NbMbDevPerUserOnMainchain} \times \text{CostPerMb}_{\text{blockchain}}\\&+ \text{NbMbDevPerUserOnLayerTwo} \times \text{CostPerMb}_{\text{LayerTwo}}\\&+ \text{NbMbDevPerUserOnStorage} \times \text{CostPerMb}_{\text{Storage}}\end{aligned}$$

i. Specific Rule B

Rule B calculates user costs similarly to Rule A but focuses on user costs. Lines 1 to 7 define the DApp and its technologies. Lines 8 to 23 entails the DApp's characteristics and calculate user costs. The formula is:

$$\begin{aligned}&\text{CostPerUser}\\&= \text{NbTxPerUserOnMainchain} \times \text{CostTx}_{\text{Blockchain}}\\&+ \text{NbTxPerUserOnLayerTwo} \times \text{CostTx}_{\text{LayerTwo}}\\&+ \text{NbMbPerUserOnMainchain} \times \text{CostPerMb}_{\text{Blockchain}}\\&+ \text{NbMbPerUserOnLayerTwo} \times \text{CostPerMb}_{\text{LayerTwo}}\\&+ \text{NbMbPerUserOnStorage} \times \text{CostPerMb}_{\text{Storage}}\end{aligned}$$

j. Specific Rules C

Rules C1 to C4 determine the read and write latency of DApp transactions. Rule C1 calculates read latency with scalability services. Rule C2 specifies that DApp may not use scalability services. Rule C3 calculate

write latency with scalability services. Rule C4 indicate that DApp may not use scalability services.

k. Specific Rules D

Rules D1 to D3 determine read and write latency for DApp data storage. Rule D1 calculates read latency with a storage service. Rule D2 specifies that a DApp may not use a storage service. Rule D3 calculates write latency with a storage service. Rule D4 calculates write latency for DApps without a storage service.

2) Open World Assumption Issues

OWL adheres to the Open World Assumption (OWA), suggesting that non-entailments are potentially true. This poses challenges for generalization in decentralized blockchain environments, as some data may be unverifiable. For example, Rule 5 discussed the CheckpointPolygon concept without complete data on Polygon blocks.

3) Inferences Found

We used Drools [22] with SWRLTab to infer new axioms from the ontology. This process, completed approximately in 750ms and generated 469 new axioms, including 261 related to data with object properties. Examples include inferred axioms for writing latency and checkpoint transactions. Listing 17 provide details on inferred costs and latencies further representing differences between user and developer costs while polygon is being implemented.

4) Conclusion of the Analysis

Our proposed ontology classifies the three DApps, evaluating how technologies impact application attributes. The key findings include: Ethereum-related solutions which incur high costs for developers and users, while Polygon significantly reduces user costs and no developer costs. Security constraints were not considered, and Polygon offers cost savings. Its DPoS consensus is less secure than Ethereum's PoW. Polygon's use as a sidechain with Ethereum facilitating added security. Some of the alternatives like rollups could also be considered for scalability issues.

RESEARCH APPLICATIONS IN THE VIDEO GAME INDUSTRY

A. Motivations

Our research also focuses on applying the advertising DNA and ontology to the blockchain video game industry. Specifically, we implement the ontology to describe B2Expand's multiplayer real-time game, LTR.

B. Application of Our Research The proposed ontology that helps elucidating the technical nuances of transitioning between blockchains, such as Ethereum and Polygon. By defining these blockchains and specifying operational restrictions through SWRL rules, we determine that Polygon is a suitable scalability solution for Ethereum.

We developed SWRL rules (Rules 3 to 6) to illustrate interactions between Ethereum and Polygon, addressing transactions, checkpoints, and transaction hashes. Through these rules, we explored semantic differences between the two blockchains, though further refinements might be needed as some modelled concepts may be too general.

Our ontology suggests th at users need a Polygon blockchain account, which is compatible with Ethereum standards. To

integrate Polygon within LTR, we propose the following steps:

1. Game Integration: Implement Polygon in Unity-based games instead of Ethereum since Polygon operates within the Ethereum ecosystem.
2. NFT Management DApp: Users should connect to Polygon with the same tools used for Ethereum.
3. Asset Minting: Assets will be minted on Ethereum and remain unchanged.
4. Asset Transfers: Asset transactions between players will occur on Polygon, reducing costs.

Figure.12, illustrates a Business Process Model and Notation (BPMN) for an LTR asset's lifecycle on Polygon as an Ethereum sidechain. B2Expand mints the asset on Ethereum, transfers it to Polygon's master contract address, and users can transfer between them. If a user needs to use the asset on a platform that does not support Polygon, they can request a withdrawal to Ethereum and wait for the processing time.

EVALUATIONS OF THE PROPOSED BLOCKCHAIN ONTOLOGY

1. Evolution: The ontology must adapt to the rapid evolution of blockchain technology. It should support scalability, enabling updates across the blockchain environment by defining the update process based on the changes of the ecosystem. It must also accommodate the add-ons of new concepts without distorting the already established models.

2. Ease of Use of the Ontology: For widespread utility, the ontology should be easily extensible. Key concepts observed in DApps are described below, and guidelines for formalizing them assisting the ontology experts are funished. This ensures the ontology remain user-friendly and adaptable to various needs.

3. Synthesis: The overall evaluation criteria are summarized in Table 1.

TABLE 1: Evaluating Our Ontology

Criteria	Operationalization
Scalability	Resilience to the appearance of new blockchain services
Ease of use ontology	knowledge level to have in order to formalize a new DApp

B. Results of the Evaluation of the Blockchain Ontology

1. Evolutivity: Our ontology shows scalability. It can integrate new blockchain services and their latency and cost characteristics. This modular approach allows for specifying components of a DApp within a consolidated model. New SWRL rules can be developed to define complex restrictions on blockchain services. Researchers can extend the ontology to address new concepts, such as economic issues affecting various actors.

2. Ease of Use of the Ontology: The ontology combines basic blockchain patterns, simplifying the formalization of new cases. More complex constraints require knowledge of SWRL. Despite SWRL's limitations, it enables the creation of new rules and extensions to address complex issues effectively.

3. Synthesis: The evaluation findings are entailed in Table 2. The table provides a summary of the assessment results, capturing

the effectiveness and applicability of the ontology based on the criteria outlined.

TABLE 2 Results Evaluating the Research Contributions

Criteria	Results
Evolutivity	The ontology is evolutive
Ease of use ontology	The ontology guides users with blockchain patterns

CONCLUSION

The DApps-focused blockchain ontology was developed to provide a formal framework for decentralized applications (DApps). Unlike existing ontologies that primarily model blockchain systems, this approach is specifically designed to support
DApp development by modeling the blockchain services that can be integrated within DApps. The work was validated in the video game industry, using the LTR game as a case study, and SWRL rules were defined to impose constraints on DApp concepts. By identifying blockchain patterns, this ontology framework can be easily adapted to various use cases. Looking ahead, additional SWRL rules can be introduced to impose new constraints, such as translating concepts across different blockchain systems. Another potential improvement would be extending the ontology to formalize solutions for crosschain interoperability.

REFERENCES

[1] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia, ``A survey on blockchain interoperability: Past, present, and future trends," ACM Comput. Surveys, vol. 54, no. 8, pp. 1-41, Nov. 2022.

[2] L. Besancon, C. F. D. Silva, and P. Ghodous, ``Towards blockchain interoperability: improving video games data exchange," in Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC), May 2019, pp. 8185.

[3] J. Pfeffer, ``EthOn Introducing semantic Ethereum," ConsenSys Media, New York, NY, USA, Tech. Rep., 2017. Accessed: May 8, 2022. [Online]. Available: https://media.consensys.net/ethonintroducing-emanticethereum-15f1f0696986

[4] N. Six, N. Herbaut, and C. Salinesi, ``Blockchain software patterns for the design of decentralized applications: A systematic literature review," Blockchain, Res. Appl., vol. 3, no. 2, Jun. 2022, Art. no. 00061.

[5] R. Materese, "Blockchain", Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep., Sep. 2019. Accessed: May 8, 2022. [Online]. Available: https://www.nist.gov/blockchain

[6] M. Zichichi, S. Ferretti, and G. D'Angelo, "On the efficiency of decentralized file storage for personal information management systems," in Proc. IEEE Symp. Comput. Commun. (ISCC), Jul. 2020, pp. 1-6.

[7] L. Lamport, R. Shostak, and M. Pease, ``The Byzantine general's problem,"in Concurrency, Works Leslie Lamport. New York, NY, USA: Association for Computing Machinery, 2019, pp. 203-226.

[8] V. Buterin, `" A next-generation smart contract and decentralized application platform", Tech. Rep., 2014. Accessed: May 8, 2022. [Online]. Available: https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_eneration_smart_contract_and_decentralized_application_platformvitalik-buterin.pdf

[9] N. E. Ioini and C. Pahl, ``A review of distributed ledger technologies," in Proc. Move Meaningful Internet Syst. (OTM) Conf., H. Panetto, C. Debruyne, H. A. Proper, C. A. Ardagna, D. Roman, and R. Meersman,

ds. Cham, Switzerland: Springer, 2018, pp. 277-288.

[10] X. Xu, I. Weber, and M. Staples, ``Blockchain patterns," in Architecture for Blockchain Applications. Cham, Switzerland: Springer, 2019, pp. 113-148.

[11] Blockchain Solutions for Gaming. Whitepaper V2 Draft, Funfair Technologies, Dublin, Ireland, 2018. Accessed: May 8, 2022. [Online]. Available: https://funfair.io/wpcontent/uploads/FunFair-commercial-WhitePaper-v2-draft.pdf

[12] OpenSea, the Largest NFT Marketplace, OpenSea, New York, NY, USA, Sep. 2021.

[13] C. Jentzsch, ``Decentralized autonomous organization to automate governance," Slock.it, Berlin, Germany, White Paper, Nov. 2016. [Online]. Available: https://lawofthelevel.lexblogplatformthree.com/wp-content/uploads/sites/187/2017/07/WhitePaper.pdf

[14] The Dai Stablecoin System, Maker, Loretto, KY, USA, 2017.

[15] V. Ojha. (2020). GitHub_IBM/BlockchainDevelopmentDesignPatterns.IBM, Armonk, NY, USA. Accessed: May 8, 2022. [Online]. Available: https://github.com/IBM/BlockchainDevelopmentDesignPatterns

[16] I. Horrocks, `" SWRL: A semantic web rule language combining OWL and RuleML'', W3C Member Submission, vol. 21, no. 79, pp. 1-31,2004.

[17] M. Proctor, ``Drools: A rule engine for complex event processing," in Proc. Int. Symp. Appl. Graph Transformations with Ind. Relevance. Cham, Switzerland: Springer, 2011, p. 2.

[18] O. Choudhury, M. Dhuliawala, N. Fay, N. Rudolph, I. Sylla, N. Fairoza, D. Gruen, and A. Das, ``Auto-translation of regulatory documents into smart contracts," in Proc. IEEE Blockchain Initiative, Sep. 2018, pp. 15.

[19] K. Karantias, ``SoK: A taxonomy of cryptocurrency wallets," Cryptol. ePrint Arch., Tech. Rep. 2020/868, 2020. Accessed: May 8, 2022. [Online].Available: https://eprint.iacr.org/2020/868

[20] W. Warren and A. Bandeali. (2017). 0x: An Open Protocol for Decentralized Exchange on the Ethereum Blockchain. [Online]. Available: https://github.com/0xProject/whitepaper

[21] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis,A. D. Caro, D. Enyeart, C. Ferris, G. Laventman , Y. Manevich, and S. Muralidharan, ``Hyperledger fabric: A distributed operating system for permissioned blockchains," in Proc. 13th EuroSys Conf., Apr. 2018, pp. 1-15.

[22] V. Buterin, A next-generation smart contract and decentralized application platform," Tech. Rep., 2014. Accessed: May 8, 2022. [Online]. Available: https://blockchainlab.com/pdf/Ethereum_white_paper_next_generation_smart_contract_and_decentralized_application_platformvitalik-buterin.pdf