CIT. Journal of Computing and Information Technology, Vol. 34, No. 1, March 2026, 23–44
doi: 10.20532/cit.2026.1006072

VICBO – An Ontology-Enhanced Blockchain for Efficient and Adaptive Vehicle Insurance Claim Management

Minh Duc Nguyen

Department of Economic Information Systems, University of Economics, Hue University, Hue, Vietnam

Vehicle insurance systems face ongoing challenges related to transparency, fraud prevention, and the efficient handling of claims, particularly in contexts involving jurisdictional variability and diverse documentation requirements. This paper introduces VICBO (Vehicle Insurance Claim based on Blockchain and Ontology), a hybrid architecture designed to support transparent, automated, and context-aware vehicle insurance claim management. The system integrates blockchain-based smart contracts with a domain ontology named VICOn, developed using the Web Ontology Language (OWL), which models insurance entities, processes, and policies. Semantic rules, expressed in Semantic Web Rule Language (SWRL), enable the evaluation of dynamic conditions such as document requirements, eligibility constraints, and jurisdiction-specific rules. The architecture also leverages the InterPlanetary File System (IPFS) for decentralized off-chain storage and integrates semantic reasoning with blockchain execution to guide smart contract decisions. To validate VICBO, a prototype was implemented using Hyperledger Fabric as the underlying blockchain platform. While the VICOn ontology was evaluated through scenario-based testing using SPARQL queries, the VICBO system was tested under varying transaction rates. Results demonstrate that VICBO achieves both semantic adaptability and operational efficiency, highlighting the promise of combining ontological reasoning with blockchain infrastructure in next-generation vehicle insurance systems.

ACM CCS (2012) Classification: Information systems → Data management systems → Information integration → Mediators and data integration

Keywords: vehicle insurance claim, blockchain, smart contract, ontology, Hyperledger Fabric

1. Introduction

The global vehicle insurance industry is undergoing a significant transformation, driven by the convergence of digital technologies, regulatory reforms, and increasing demands for transparency, automation, and personalization. Traditional vehicle insurance workflows, ranging from policy issuance and accident reporting to claim verification and compensation, are often plagued by inefficiencies, information asymmetry, and susceptibility to fraud [1], [2]. Moreover, these processes frequently rely on fragmented and siloed data sources, making it difficult for insurers to obtain a complete and trustworthy view of each claim. These challenges are further compounded by fragmented data sources, manual adjudication processes, and inconsistent compliance with legal and procedural standards across jurisdictions. As a result, current systems struggle to deliver timely, consistent, and auditably fair outcomes, especially in cases requiring jurisdiction-specific validation or multi-party coordination [3], [4].

In recent years, blockchain technology has emerged as a promising infrastructure for building decentralized, secure, and auditable insurance platforms. Through its core features of immutability, consensus, and smart contract automation, blockchain enables stakeholders to interact in a tamper-resistant and transparent manner [5], [6]. Several studies have proposed blockchain-based systems that record policy terms on-chain, automate claims through smart contracts, and reduce fraudulent behavior through shared ledgers and cryptographic evidence. Hybrid architectures that combine blockchain with decentralized storage networks such as InterPlanetary File System (IPFS) have also been explored to support secure yet scalable document and metadata management [7], [8]. However, despite these advances, existing blockchain-based solutions focus primarily on automating transactions and recording data, offering limited support for interpreting contextual information or reasoning about nuanced insurance rules. Smart contracts alone cannot adapt to jurisdictional differences, dynamically changing policy conditions, or complex eligibility criteria that depend on semantic relationships between incidents, vehicles, documents, and regulatory entities. In this regard, ontologies have been recognized as powerful enablers of semantic reasoning and adaptive insurance logic by providing a formal, machine-interpretable model of domain knowledge [9], [10].

This paper introduces VICBO (Vehicle Insurance Claim based on Blockchain and Ontology), a hybrid architecture that combines smart contracts for secure execution with VICOn, a Web Ontology Language (OWL)-based ontology that models entities, processes, and rules within the vehicle insurance domain. VICBO enables semantic-aware, transparent, and partially automated management of insurance claims by utilizing smart contracts and semantic reasoning. It also employs decentralized document storage via IPFS and evaluates complex conditions, such as jurisdiction-based document requirements and eligibility criteria based on vehicle class hierarchies. The contributions of this study are threefold:

- Design and implementation of a blockchain-based vehicle insurance workflow using Hyperledger Fabric that supports contract agreement, document referencing, claim verification, and compensation payment.
- Development of VICOn, a domain ontology for modeling vehicle insurance processes, and propose Semantic Web Rule Language (SWRL) rules for context-aware evaluation.
- Validation of the system through SPARQL-based scenario testing and performance experiments to measure throughput and latency under various transaction loads.

The remainder of this paper is organized as follows: Section 2 reviews related work on blockchain-enabled vehicle insurance systems and semantic technologies. Section 3 introduces the system architecture and ontology design. Section 4 presents scenario-based ontology validation and system performance evaluation. Section 5 concludes with key findings and directions for future research.

2. Literature Review

2.1. Blockchain and Its Ecosystem

Blockchain technology has evolved from its initial application in cryptocurrency to become a foundational infrastructure for decentralized, tamper-resistant, and transparent systems. At its core, a blockchain is a distributed ledger that maintains a sequence of cryptographically linked blocks, each representing a batch of transactions [11]. The decentralized and consensus-driven nature of blockchain ensures that no single entity can alter past records without detection, making it an attractive solution for domains that require trust, auditability, and integrity. One of the most transformative components of blockchain systems is the smart contract. Introduced conceptually by Szabo [12], a smart contract is a self-executing code deployed on a blockchain that autonomously enforces business logic when predefined conditions are met. In the context of insurance, smart contracts can encode policy terms and automate actions such as claim approval and fund disbursement, thereby eliminating manual intervention and reducing processing delays [2]. Ethereum has been a pioneer in general-purpose smart contracts, while platforms like Hyperledger Fabric offer permissioned alternatives more suited for enterprise collaboration, such as in consortiums of insurers [13].

Moreover, blockchain is able to support token-based mechanisms for financial transactions. In insurance systems, stablecoins or utility tokens can be used for premium payments, claim compensation, or incentivizing safe behavior [14]. While Ethereum and public chains typically use ERC-based tokens, permissioned frameworks like Fabric can implement token transfers using chaincode logic and state-based endorsement policies, allowing for flexible control over value flows between stakeholders [5].

Privacy and access control are additional concerns addressed by the blockchain ecosystem. Techniques such as Zero-Knowledge Proofs (ZKP), Decentralized Identity (DID), and selective disclosure protocols are being integrated to ensure that sensitive user data (*e.g.*, behavioral patterns or medical records) can be validated without being disclosed [14], [15]. These advances are especially relevant for insurance applications where compliance with privacy laws (*e.g.*, General Data Protection Regulation – GDPR) is essential.

In sum, the blockchain ecosystem offers a versatile and modular foundation for building robust vehicle insurance systems. Its components including smart contracts, tokenization, and privacy mechanisms can be combined to deliver transparent, efficient, and adaptive insurance workflows.

2.2. Blockchain-enabled Vehicle Insurance Systems

The vehicle insurance sector has increasingly adopted blockchain technology as a means of addressing longstanding challenges such as fraud, lack of transparency, and inefficiencies in claims processing. Blockchain's core features, including immutability, decentralization, and automation through smart contracts, are especially well-suited for insurance scenarios that demand accountability and auditability [2], [6].

Smart contracts have been employed to automate insurance workflows, ranging from policy issuance to claims validation, and compensation. For instance, Hassan *et al.* [6] designed an Ethereum-based insurance framework in which predefined rules encoded into smart contracts automatically reject invalid claims based on policy conditions. Similarly, Yadav *et al.* [5] leveraged Hyperledger Fabric to store vehicle, owner, and insurance data, streamlining accident reporting and claims processing through automated validation mechanisms, with the potential to reduce errors and mitigate fraudulent or duplicate claims. Several real-world prototypes have demonstrated blockchain's effectiveness in vehicle insurance contexts. Lamberti *et al.* [16] developed an on-demand insurance system combining IoT sensors with blockchain-based smart contracts, allowing users to toggle coverage dynamically, with all interactions immutably recorded on-chain. Liu *et al.* [17] proposed a blockchain-based auto insurance data-sharing scheme among stakeholders to enhance privacy protection and improve claims processing efficiency, leveraging proxy re-encryption for secure data exchange. Blockchain also supports the integration of telematics data for usage-based insurance. Qi *et al.* [14] introduced a decentralized framework that calculates premiums based on driving behavior while preserving driver privacy through zero-knowledge proofs. Huang *et al.* [18] advanced this approach by leveraging smart contracts on a consortium blockchain to ensure transparent, privacy-preserving evaluation of driving behavior, enabling insurance companies to adjust premiums based on encrypted telematics data. Fraud mitigation is another domain where blockchain demonstrates considerable promise. Roriz and Pereira [4] discussed how blockchain and smart contracts can be integrated to prevent specific types of fraud in vehicle insurance, emphasizing the potential of automated and transparent processes to reduce fraudulent activities. Building on the broader application of blockchain in insurance operations, Bhadra *et al.* [19] introduced a blockchain-based solution for insurance subrogation, leveraging smart contracts to securely and transparently streamline reimbursement and coordination among insurers and other stakeholders involved in the post-settlement process. To handle large-scale and real-world data, blockchain-based systems often adopt hybrid architectures. Nizamuddin and Abugabah [7], as well as Chen *et al.* [8], implemented frameworks that combine blockchain with IPFS to ensure data integrity, reduce storage costs, and enhance scalability in auto insurance processes. Both studies emphasize securing insurance data through decentralized storage and cryptographic techniques to address challenges such as fraud, inefficiency, and high administrative overhead.

Despite these innovations, challenges remain. Transaction throughput, data privacy, legal interoperability, and integration with existing insurance systems require further development [3], [15]. Nonetheless, the emerging body of work confirms that blockchain-enabled systems can offer significant improvements in transparency, automation, and trust across the vehicle insurance lifecycle.

2.3. Synergizing Ontology and Blockchain

While blockchain ensures integrity, immutability, and automation through smart contracts, it lacks semantic interpretability and contextual awareness. In complex domains such as vehicle insurance, where claim decisions often depend on nuanced factors (*e.g.*, driving context, regulatory rules, supporting documents), the integration of ontology-based knowledge models with blockchain is gaining attention as a promising solution [10], [20]. Ontologies provide a formal and machine-readable representation of domain knowledge, allowing the system to define relationships between key concepts. This structured vocabulary enables semantic interoperability, ensuring that data exchanged across systems and stakeholders can be interpreted consistently [21]. When used alongside semantic rules written in languages such as SWRL, ontologies empower automated reasoning about policy terms, eligibility criteria or document requirements [9].

Several works have demonstrated how this synergy enhances smart contract capabilities. Chondrogiannis *et al.* [10] applied semantic web technologies to model health insurance terms and integrated them with Ethereum-based smart contracts to support fine-grained, policy-driven evaluation of healthcare insurance agreements. Woensel and Seneviratne [22] proposed a hybrid approach for automatically generating smart contracts, written in languages such as Solidity, from semantic knowledge graphs. This method allows policy logic, modeled using OWL ontologies and N3 rules, to be maintained at a high semantic level and translated into efficient executable code. Their approach was demonstrated in health insurance scenarios, where coverage decisions are automated based on semantically defined regulations and patient data, ensuring transparency and consistency on blockchain platforms. Moreover, the use of ontologies improves trust in data inputs. For example, Kim and Laskowski [23] demonstrated how ontologies can be integrated with blockchain to enhance provenance tracking by translating semantic traceability constraints into smart contracts, ensuring data integrity and traceability across complex supply chains.

In summary, the synergy between ontology and blockchain enables context-aware, dynamic, and legally compliant claim processing. While blockchain provides execution guarantees, ontologies provide the reasoning framework that supports interpretation and adaptability. This layered architecture aligns well with the needs of vehicle insurance, where each claim may vary based on jurisdiction, incident type or driver history. The growing body of research points to hybrid architectures as a key enabler of next-generation insurance systems.

3. The System Design of VICBO

This section presents the design of the VICBO system, focusing on its architecture and operational workflow, the development of the ontological model, and the construction of semantic rules.

3.1. System Architecture and Workflow of VICBO

Figure 1 illustrates the system architecture of VICBO designed for vehicle insurance claim management. The architecture is organized into four functional layers that interact to ensure a transparent, traceable, and semantically rich claim processing workflow.

The off-chain storage layer is responsible for storing large and sensitive claim-related documents (*e.g.*, accident images, police reports, and repair estimates). A cryptographic hash of this data is generated and passed to the smart contract layer, which serves as the core logic controller of the system. This smart contract layer manages contract deployment, claim registration, policy enforcement, and the initiation of payments based on validated inputs. The knowledge layer, composed of a domain ontology and semantic rules, supports the system's reasoning and validation functions, and its outputs are incorporated into the workflow before being submitted to the blockchain. Finally, all critical events and transactions are immutably recorded in the blockchain layer, ensuring verifiability and auditability across the system. Together, these four layers form a modular and interoperable architecture that supports the end-to-end functionality of the VICBO system.

Figure 2 provides the workflow of VICBO involving four primary actors who interact

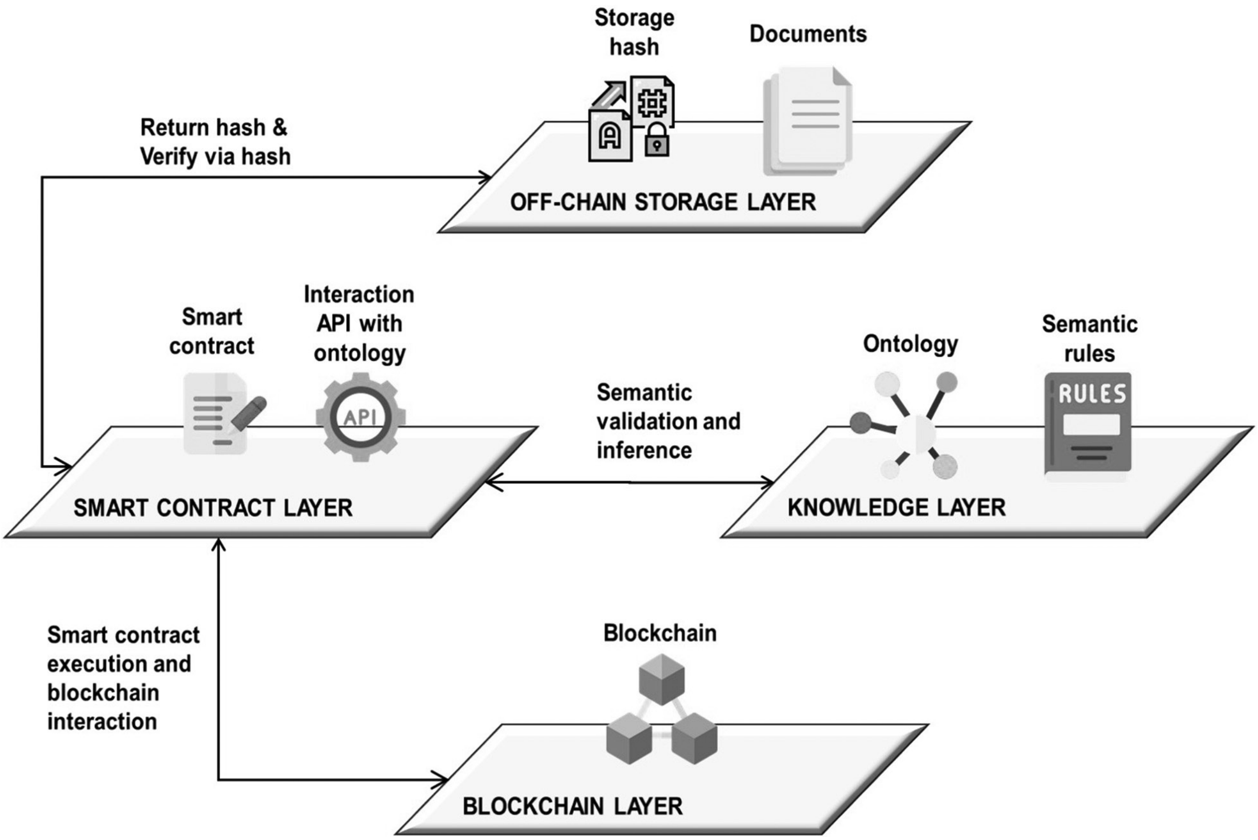


Figure 1. The system architecture of VICBO.

throughout the vehicle insurance claim process. The vehicle owner initiates the process by signing the insurance contract and submitting a claim along with relevant evidence following an incident. The insurance company is responsible for reviewing the claim, evaluating the reasoning output, and making the final approval or rejection decision. The repair shop carries out the necessary vehicle repairs and receives compensation upon claim approval. The competent authority, such as the traffic police or regulatory body, plays a supporting role by issuing official documents (*e.g.*, accident or incident reports) that the vehicle owner must include when filing a claim.

This workflow begins when a vehicle owner and an insurance company establish a contractual agreement that defines the terms of coverage, eligibility conditions, and compensation thresholds. Upon mutual agreement, the policy is encoded and deployed as a smart contract on the blockchain. From this point onward, the term smart contract refers specifically to this immutable, blockchain-based version. This deployment guarantees immutability and transparency, ensuring that all parties adhere to the predefined terms without the possibility of post hoc modification.

When a claim is initiated, the vehicle owner submits all relevant information and supporting documents through the VICBO system interface. These may include photographs of the damaged vehicle, repair estimates, police reports, and other evidence of the incident. Instead of storing these files directly on the blockchain, which would be costly and inefficient, the system leverages off-chain storage solutions (such as IPFS or a secured institutional database) to manage large or sensitive files. Once the evidence is uploaded, a cryptographic hash representing the content is generated and submitted to the smart contract. This hash serves as a verifiable reference to the original off-chain data, ensuring its integrity without compromising privacy.

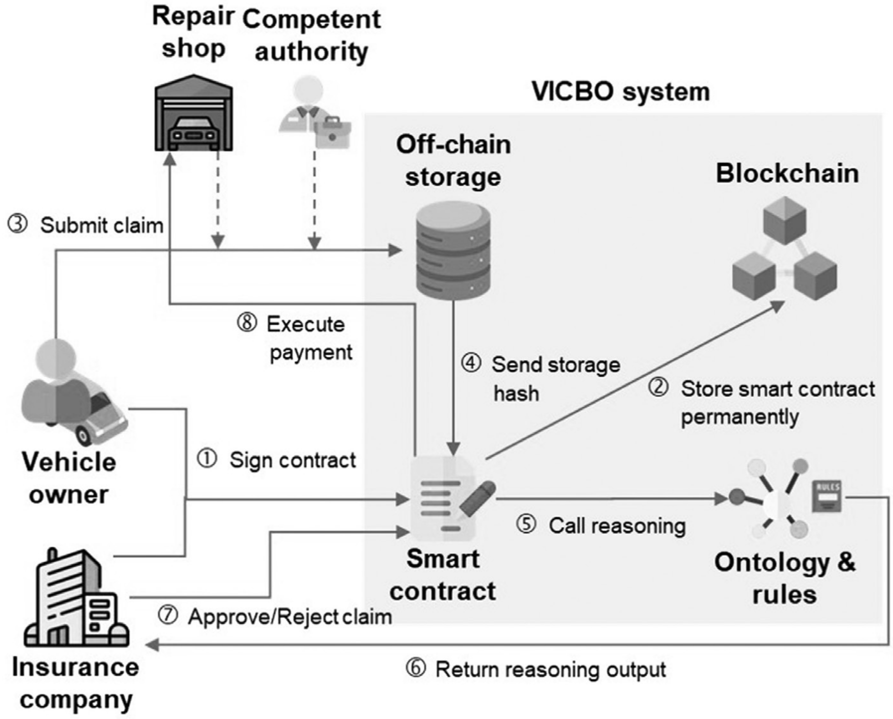


Figure 2. The workflow of VICBO.

Following the receipt of the claim and associated hash, the smart contract interacts with the ontology-based reasoning module. This semantic layer is built on an OWL ontology enriched with domain knowledge and regulatory logic, and it is complemented by a set of rules encoded using the SWRL. The reasoning engine analyzes the claim in relation to policy coverage, historical claims, and predefined conditions to infer recommendations.

The outcome of the reasoning process is transmitted to the insurance company for final review. While VICBO automates much of the initial evaluation, the human decision-maker retains ultimate authority, especially for complex or borderline cases. Along with the recommendation, the reasoning log is also delivered to the insurer, enabling transparency in how the conclusion was derived. Once the decision is made, the insurer updates the smart contract with the approval or rejection outcome and submits a hash or compressed representation of the reasoning trace. This ensures that the logic behind the decision is auditable and verifiable on-chain, even if the full log remains off-chain.

Upon approval, the smart contract executes the compensation by transferring funds directly to the designated repair facility or service provider. This automated disbursement eliminates manual payment handling, reduces settlement time and mitigates risks related to fraud or administrative errors. All critical events (*e.g.*, claim submission, decision logging, and payment execution) are recorded immutably on the blockchain, ensuring end-to-end traceability and accountability throughout the insurance claim lifecycle.

3.2. GDPR Compliance Strategy for European Union Deployment

Deploying blockchain-based claim management systems within the European Union requires strict adherence to the GDPR. However, GDPR introduces several challenges for distributed ledger technologies due to immutability, decentralised governance, and the difficulty of modifying or deleting on-chain information. These tensions have been extensively analyzed by the European Parliament's Scientific Foresight Unit (STOA) in its study [24]. Following the recommendations highlighted in this report, VICBO adopts a privacy-preserving architecture that integrates off-chain storage, encryption mechanisms, and cryptographic erasure to ensure GDPR compliance.

A key recommendation from the STOA study is to keep personal data off-chain and store only hash pointers on the blockchain to facilitate compliance with Articles 16 and 17 of the GDPR [24]. In VICBO, documents such as photographs, police reports, and invoices are therefore not written to the blockchain. Instead, they are encrypted and uploaded to Web3.Storage, an IPFS-Filecoin-based storage service. The blockchain contains only a cryptographic hash and a content identifier (CID), which serve as verifiable references but do not reveal personal information. This design enables the modification or deletion of personal data without interacting with the immutable ledger, thereby aligning with GDPR's data minimisation requirements.

Technical measures are central to GDPR compliance, particularly for ensuring confidentiality and controlled access. As emphasised by the STOA report, GDPR compliance in distributed ledger environments requires clearly defined governance and access-control mechanisms [24]. VICBO adopts a hybrid encryption strategy in which each document is encrypted using a symmetric AES-256 key. This key is then encrypted using the public keys of authorised parties (*e.g.*, insurer, vehicle owner, competent authority). Smart contracts store only encrypted key metadata and hash references, ensuring the blockchain does not contain raw personal data. This model ensures confidentiality, role-based access, and controlled distribution of sensitive information.

GDPR's right to erasure (Article 17) poses a significant challenge for blockchain systems, since data stored on-chain cannot be removed or retroactively modified. The STOA report stresses the inherent difficulty of reconciling Article 17 with blockchain's immutability [24]. To address this requirement, VICBO implements a cryptographic erasure mechanism. When a deletion request is made, the system permanently removes the encrypted off-chain document and destroys the corresponding decryption keys. As highlighted in the STOA study, such approaches, where data is rendered inaccessible rather than physically destroyed, can satisfy GDPR erasure requirements when destruction is technically infeasible [24]. After key deletion, the content becomes unrecoverable even though the hash remains on-chain; because this hash no longer enables identification, it is no longer classified as personal data.

Article 25 of the GDPR requires data protection by design and by default. VICBO satisfies this principle by minimising on-chain data, restricting identifiable information to encrypted off-chain stores, and operating on a permissioned blockchain architecture that clearly defines roles and data controllership. This aligns with the STOA study's recommendation that blockchain deployments adopt governance structures that enable accountability and ensure that data controllers can fulfil GDPR obligations [24]. Semantic reasoning in VICBO operates on abstract identifiers rather than raw personal information, further reducing exposure risks and strengthening compliance.

By integrating off-chain encrypted storage, hybrid key management, cryptographic erasure, and clearly defined data governance structures, VICBO addresses the major GDPR challenges identified in the STOA report. This ensures that the system can be safely deployed within the European Union while maintaining the transparency, auditability, and automation benefits of blockchain technology.

3.3. Modeling Policy Management with Ontological Reasoning

In VICBO, policy management is semantically modeled using VICOn. This ontology serves as the formal knowledge base for reasoning over claims and policies in a machine-interpretable manner, providing the semantic structure that underpins automated eligibility assessments, evidence validation, and process coordination. The development of VICOn follows the NeOn methodology [25], one of the most widely adopted approaches in ontology engineering due to its understandability, scenario orientation, and comprehensive documentation support. The ontology was constructed through a collaborative and iterative process involving both ontology engineers and domain experts in insurance and blockchain technologies.

In the first phase, we invited stakeholders from both technical and domain backgrounds to work collaboratively using Protégé[1] and GitHub[2]. This setting enabled real-time modeling, issue tracking and version control throughout the development lifecycle. In the second phase, a set of specifications was formalized using an Ontology Requirements Specification Document (ORSD) to capture the key knowledge areas, competency questions (CQs), and coverage expectations that the ontology needed to fulfill. These requirements were derived from real-world vehicle insurance workflows and policy rules.

The third phase emphasized reusing existing ontological resources where applicable. Foundational ontologies such as the Dublin Core[3], the Time Ontology[4], and the Event Ontology[5] were reused to promote semantic interoperability and reduce development effort. Specifically, Dublin Core terms were adopted for representing metadata associated with insurance-related documents such as claims, reports, and repair estimates; the Time Ontology was used to model temporal information such as policy durations, incident timestamps, and claim submission dates; and the Event Ontology provided a conceptual framework for capturing the structure and relationships of vehicle-related incidents. By leveraging these existing ontologies, VICOn ensures alignment with widely accepted standards while allowing the development team to focus on domain-specific elements of the vehicle insurance claim process. The development process was repeated iteratively until consensus was achieved across all project members. As a result, VICOn embodies both practical relevance and formal consistency, enabling semantic reasoning in a way that is adaptable to evolving insurance regulations and jurisdictional requirements.

As a result, Figure 3 presents an excerpt of the VICOn ontology, visualized using the OntoGraf plugin of Protégé. This diagram illustrates key classes and their semantic relationships. Object properties are used to connect these classes, with each property defined by a domain (source class) and a range (target class). To complement this, Table 1 summarizes typical object properties used in VICOn, demonstrating how the ontology models real-world policy structures and actor interactions through formalized semantic links.

In summary, the development of the VICOn ontology provides a formal and extensible foundation for representing policy-related knowledge in the VICBO system. The final ontology comprises 44 classes, 61 object properties, 18 data properties, and 210 individuals, supported by over 212 logical axioms, reflecting a well-structured and semantically rich knowledge model. From the 44 concepts identified in the ORSD, 16 concepts (36.4%) were successfully mapped to existing ontologies, including 10 concepts reused from Dublin Core, 3 concepts from the Time Ontology, and 3 concepts from the Event Ontology. The remaining concepts were modeled as VICOn-specific classes or properties where no suitable standard term existed. These reuse ratios demonstrate that the ontology design is aligned with established semantic standards while introducing new domain-specific constructs only when necessary. This ontology-driven approach enables VICBO to manage insurance policies and claims in a structured, transparent, and machine-understandable manner.

3.4. Design of Semantic Rules for Adaptive and Context-aware Claim Processing

In VICBO, semantic rules are developed not only for basic policy evaluation but also to support adaptive and context-aware reasoning in insurance claim processing. These rules are defined using the SWRL, which enables the inference of new knowledge from existing facts in the VICOn ontology. This approach allows the system to dynamically classify claims, validate conditions, and recommend actions based on contextual factors such as jurisdictional policies, incident types, document availability, and vehicle usage.

---

[1] https://protege.stanford.edu

[2] https://github.com

[3] https://www.dublincore.org

[4] https://www.w3.org/TR/owl-time

[5] https://w3id.org/MON/event.owl

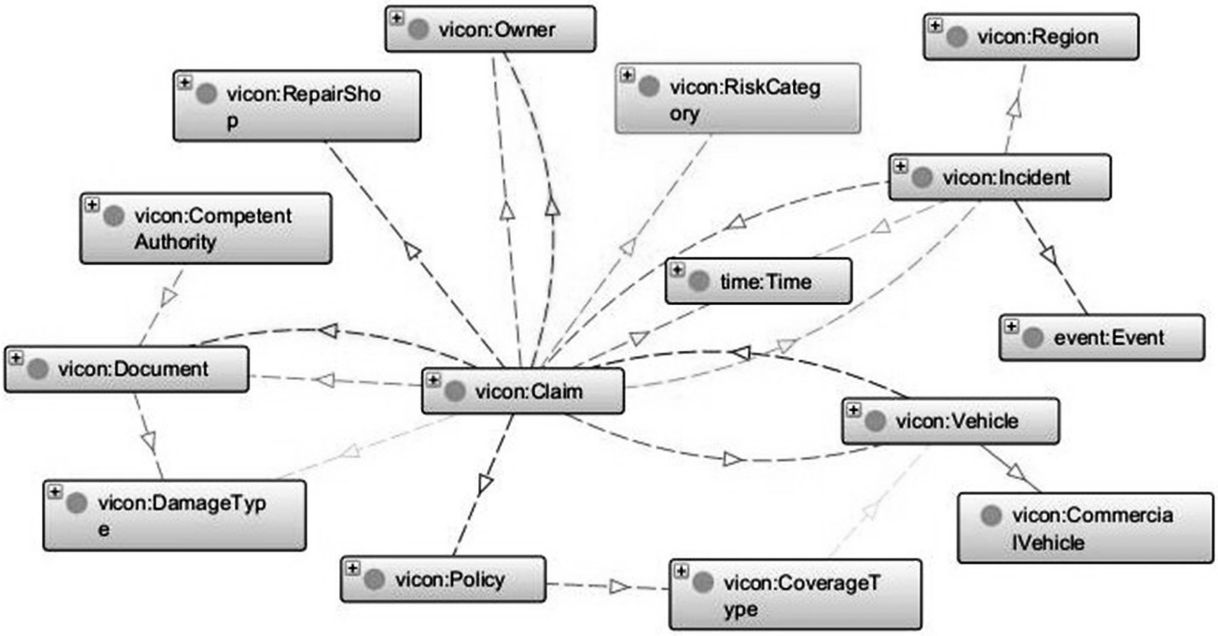


Figure 3. An excerpt from the VICOn ontology.

Table 1. Typical properties used in the VICOn ontology.

Property	**Domain**	**Range**
relatedToIncident	Claim	Incident
involvesVehicle	Claim	Vehicle
occursInRegion	Incident	Region
requiresDocument	Claim	Document
submittedBy	Claim	Owner
hasDamageType	Claim	DamageType
hasCoverageType	Policy	CoverageType
hasIncidentDate	Claim	Time
relatedToClaim	Incident	Claim
relatedToClaim	Vehicle	Claim
relatedToPolicy	Claim	Policy

Each SWRL rule in VICBO is defined as a logical expression composed of an antecedent (also referred to as the rule body) and a consequent (also referred to as the rule head). The antecedent specifies a set of conditions that must be satisfied for the rule to be activated, while the consequent defines the new knowledge that is inferred once those conditions are met. These rules are constructed using OWL classes, object properties, data properties, and individuals, along with built-in SWRL functions for operations such as comparisons, datatype matching, and logical conjunctions. For the execution and reasoning process, VICBO employs the Pellet reasoner [26], integrated with the Protégé ontology development environment. Pellet supports both OWL DL reasoning and SWRL rule execution, allowing VICBO to perform consistency checking and rule-based inference over policy and claim data.

Unlike smart contracts, which are fixed and procedural, semantic rules allow VICBO to adapt its behavior based on semantic context. For example, semantic rules allow the system to respond to local regulations, interpret variations in required documentation and flag risk-sensitive scenarios. Table 2 presents three representative SWRL rules that demonstrate VICBO's adaptive reasoning capabilities where namespace prefixes are omitted for legibility.

These rules in Table 2 demonstrate the ability of VICBO to perform fine-grained, explainable, and configurable reasoning. For example, if insurance policy rules vary by province or country, new regions and their regulatory status can be added as individuals in the ontology, and corresponding rules can be updated or extended. Similarly, changing risk classifications or document requirements can be handled simply by modifying or adding rules, rather than modifying the software or blockchain contracts. This rule-based reasoning feature significantly increases the adaptability, maintainability, and transparency of VICBO. It enables the system to evolve in response to regulatory changes or institutional policy updates while retaining formal semantic rigor and traceability.

4. Evaluation of the VICBO System

This section presents a comprehensive evaluation of the proposed VICBO system. The focus is placed on assessing both the semantic reasoning capabilities enabled by the VICOn ontology and the operational performance of the integrated blockchain-based architecture.

Table 2. Examples of SWRL rules for adaptive claim processing in VICBO.

No.	**SWRL rule**	**Rule description**	**Context dependency**
1	Claim(?c) ∧ hasRegion(?c, RegionA) ∧ requiresStricterReview(RegionA, true) → ManualReviewRequired(?c)	If a claim originates from a region with stricter insurance regulations, it requires manual review.	Jurisdictional policy differences
2	Incident(?i) ∧ CollisionWithInjury(?i) ∧ Claim(?c) ∧ relatedToClaim(?i, ?c) → requiresDocument(?c, MedicalReport)	If the incident is a "Collision with Injury", a medical report is required in the claim submission.	Incident type and document rules
3	Vehicle(?v) ∧ hasUseType(?v, HazardousTransport) ∧ isCommercialVehicle(?v, true) ∧ Claim(?c) ∧ relatedToClaim(?v, ?c) → HighRiskClaim(?c)	If the damaged vehicle is a commercial truck used for hazardous transport, the claim is high risk.	Vehicle type and usage context

4.1. Practical scenario-based testing of the VICOn ontology

Before conducting scenario-based validation, the VICOn ontology was checked for structural soundness using the Pellet reasoner in Protégé. The ontology conforms to the OWL 2 DL profile with an expressivity level of ALCHIQ(D). Pellet reported no inconsistent axioms and zero unsatisfiable classes, confirming the coherence of the class hierarchy and the correctness of domain and range specifications after refinement. This validation also included all alignment axioms and imported ontology terms (*e.g.*, Dublin Core, Time, Event), ensuring that the reuse of external vocabularies does not introduce inconsistencies. All property constraints, class restrictions, and imported vocabularies were successfully classified without warnings. This reasoner health check demonstrates that VICOn meets the foundational requirements for consistency and logical validity prior to competency question validation. To evaluate the practical utility and reasoning capabilities of the VICOn ontology, we conducted scenario-based testing using complex and realistic vehicle insurance claim situations. Each scenario involves adaptive and context-sensitive conditions that require semantic reasoning such as jurisdiction-specific document requirements, subclass-based vehicle coverage, and behavioral risk detection based on claim history. The test cases were executed over an instance-based version of VICOn populated with synthetic data that simulates real-world insurance environments. For each scenario, a SPARQL query was designed to extract or validate inferred knowledge based on ontology assertions and reasoning outputs.

4.1.1. Scenario 1: Adaptive Document Requirements Based on Incident Type and Jurisdiction

This scenario demonstrates the capability of the VICOn ontology to support adaptive claim validation based on the semantic combination of incident type and jurisdiction. Specifically, it tests whether the ontology can infer that a claim associated with a *Collision* with *Injury* incident in a region requiring stricter regulatory oversight (*e.g.*, Region A) must include both a police report and a medical report. The document requirements are not globally fixed but depend on regional attributes captured in the ontology through object and data properties (*e.g.*, *occursInRegion* and *requiresStricterReview*). Table 3 presents the SPARQL query and its result for this scenario.

The result confirms that for the given claim, the ontology correctly infers the need for both a police report and a medical report, satisfying context-specific business rules. To further enhance transparency, Figure 4 provides an explanation tree that traces how scenario 1's inference was derived.

Table 3. SPARQL query and result for scenario 1.

```
PREFIX vicon: <https://w3id.org/vicon#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?claim ?requiredDoc WHERE {
    ?claim a vicon:Claim ;
           vicon:requiresDocument ?requiredDoc .
    ?incident a vicon:CollisionWithInjury ;
              vicon:relatedToClaim ?claim ;
              vicon:occursInRegion ?region .
    ?region a vicon:Region ;
    ?region vicon:requiresStricterReview ''true''^^xsd:boolean .
}
```

```
claim                      requiredDoc
vicon:Claim122             vicon:PoliceReport
vicon:Claim122             vicon:MedicalReport
```

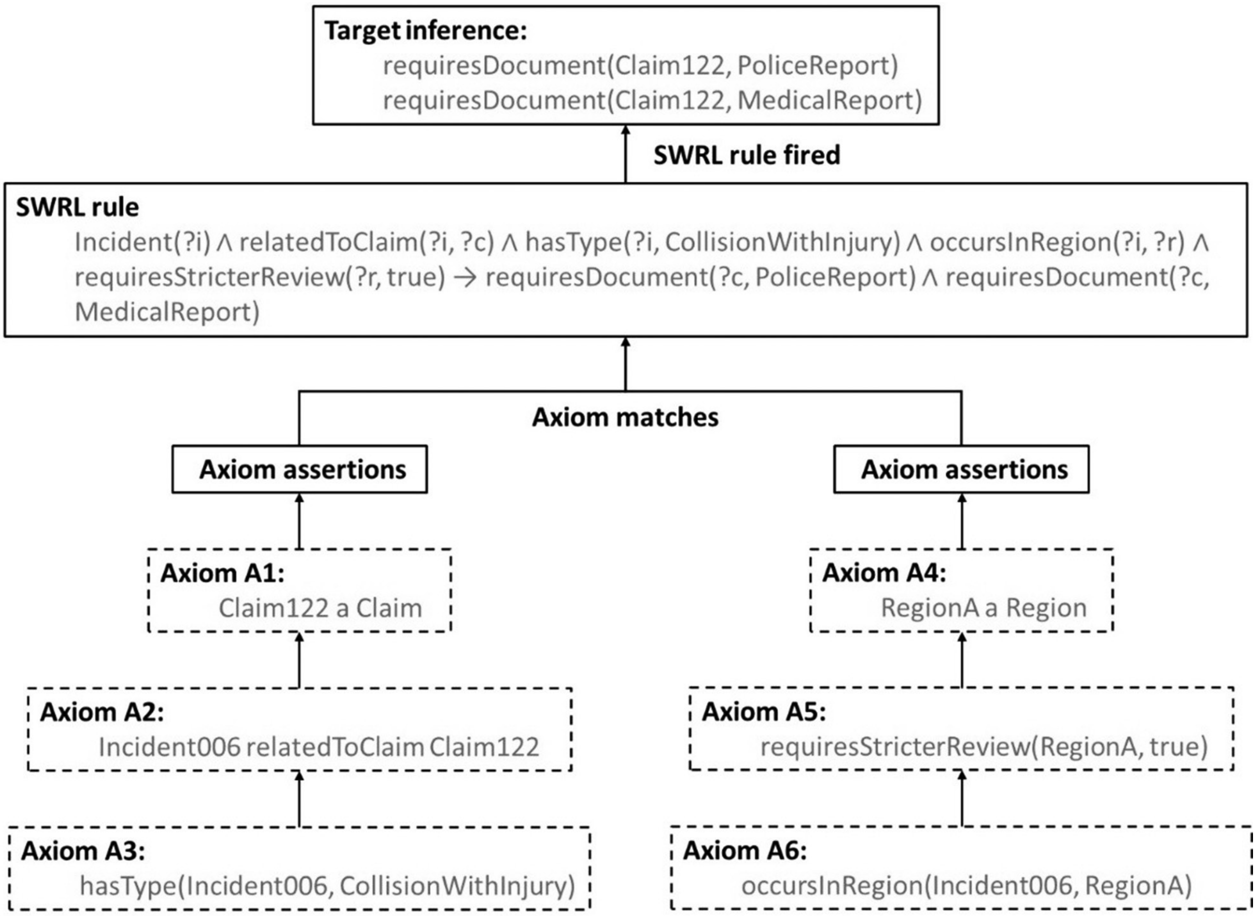


Figure 4. Reasoning explanation tree for scenario 1.

Figure 4 visualizes the SWRL rule firing together with the supporting ontology axioms, illustrating the precise chain of semantic matches that led the reasoner to conclude that Claim122 requires both a police report and a medical report. This reflects VICOn's ability to integrate regulatory semantics and incident classifications, offering adaptive and explainable validation of claim requirements.

4.1.2. Scenario 2: Determining Eligibility Based on Class Hierarchies of Vehicle Type

This scenario demonstrates the importance of subclass reasoning within VICOn when determining a claim's eligibility based on policy coverage. In this case, a policy covers vehicles classified under the category `ElectricVehicle`, while the damaged vehicle is of type `TeslaModelY`, which is a subclass of `ElectricVehicle`. The goal is to verify whether VICOn correctly infers the eligibility of the claim by resolving the semantic hierarchy between the specific vehicle model and the general class specified in the policy. This scenario reflects real-world situations where policies are written at a generic level (*e.g.*, covering all electric vehicles), but claim data references specific vehicle types. Table 4 presents the SPARQL query and the corresponding result for scenario 2.

The result in Table 4 confirms that the claim is considered eligible because the involved vehicle is semantically classified under the coverage type specified in the policy through subclass reasoning. This illustrates the essential role of ontological hierarchies in representing and inferring policy-claim compatibility. Without reasoning over the class structure, such a match would require explicit type assertions or hardcoded mappings, limiting the system's flexibility and accuracy.

Table 4. SPARQL query and result for scenario 2.

```
PREFIX vicon: <https://w3id.org/vicon#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?claim WHERE {
   ?claim a vicon:Claim ;
          vicon:relatedToPolicy ?policy ;
          vicon:relatedToIncident ?incident ;
          vicon:involvesVehicle ?vehicle .
   ?policy vicon:hasCoverageType vicon:ElectricVehicle .
   ?vehicle a ?vehicleType .
   ?vehicleType rdfs:subClassOf vicon:ElectricVehicle .
}
```

```
claim
vicon:Claim323
```

4.1.3. Scenario 3: Detecting Fraud Risk Based on Cross-claim Patterns

In this scenario, VICOn is demonstrated to support behavioral risk assessment by detecting patterns across multiple claims submitted by the same vehicle owner. Specifically, the goal is to identify claims that may be classified as suspicious based on both claim frequency and damage severity. In this case, if a vehicle owner has submitted more than three claims within the past year and the current claim involves minor damage, the system should flag the current claim as potentially fraudulent. Traditional smart contracts or flat data systems are not well suited for this kind of historical pattern detection, whereas an ontology can support semantic rules and queries that span multiple related instances across time. The query in Table 5 identifies claimants who have submitted more than three claims within the past year and retrieves their most recent claim involving minor damage.

The result in Table 5 confirms that `vicon:Claim921`, involving minor damage, was submitted by a vehicle owner who has submitted more than three claims in the past year. This finding aligns with business logic that flags such patterns as potentially fraudulent. Without the use of semantic modeling and aggregation, this type of cross-instance pattern detection would be difficult to express declaratively.

To complement the positive reasoning scenarios described above, VICOn was also evaluated using adversarial tests to verify that undesired inferences are not produced. For scenario 1, which checks jurisdiction- and incident-dependent document requirements, a counter-scenario was created in which a claim involved a `CollisionWithoutInjury` incident occurring in a region without stricter review requirements. As expected, the SPARQL query for required documents returned an empty result set, confirming that the ontology does not incorrectly infer the need for a medical report or additional documents. For scenario 2, which evaluates eligibility based on class hierarchies, a `DieselVan` (explicitly modeled as disjoint from `ElectricVehicle`) was tested against the same eligibility condition. The corresponding query produced no matches, demonstrating that the ontology does not infer eligibility for non-covered vehicle types. For scenario 3, which identifies high-risk behavioral patterns based on claim frequency, a counter-scenario was constructed in which a vehicle owner had only one recent claim involving `MinorDamage`. As expected, no suspicious claim was inferred. These adversarial tests collectively confirm that the ontology maintains high precision and avoids false-positive inferences across all tested reasoning situations.

Table 5. SPARQL query and result for scenario 3.

```
PREFIX vicon: <https://w3id.org/vicon#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?currentClaim WHERE {
  ?currentClaim a vicon:Claim ;
                vicon:submittedBy ?owner ;
                vicon:hasDamageType vicon:MinorDamage ;
                vicon:hasIncidentDate ?currentDate .

  {
    SELECT ?owner (COUNT(DISTINCT ?pastClaim) AS ?claimCount) WHERE {
      ?pastClaim a vicon:Claim ;
                 vicon:submittedBy ?owner ;
                 vicon:hasIncidentDate ?d .
      FILTER (?d >= (NOW() - ''P365D''^^xsd:dayTimeDuration))
    }
    GROUP BY ?owner
    HAVING (?claimCount > 3)
  }
}
```

```
currentClaim
vicon:Claim921
```

The three testing scenarios collectively demonstrate the practical capabilities of the VICOn ontology to support adaptive, context-aware, and semantically rich decision-making in vehicle insurance claim management. By enabling reasoning over class hierarchies, contextual policy rules, and behavioral patterns across multiple claims, VICOn provides a flexible framework that extends far beyond static rule-checking or procedural logic. The SPARQL queries evaluated against the ontology validate its structural soundness, semantic expressiveness, and effectiveness in modeling real-world insurance operations. These results highlight VICOn's critical role in enhancing transparency, adaptability, and automation within the VICBO system.

4.2. Reasoning Performance and Scalability of VICOn

In addition to scenario-based validation, it is essential to assess the reasoning performance of the VICOn ontology to ensure that it remains scalable and efficient when deployed in large-scale vehicle insurance environments. Since semantic reasoning can become computationally expensive as the number of ontology individuals increases, this subsection reports the execution time, memory consumption, and scalability of VICOn under progressively larger datasets. To evaluate scalability, synthetic datasets containing 1,000, 10,000, 50,000, and 100,000 individuals were generated. These individuals represent realistic combinations of claims, incidents, vehicles, policy instances, stakeholders, and document metadata. The Pellet reasoner was employed to execute three core reasoning tasks including (1) ontology classification, (2) global consistency checking, and (3) execution of SWRL rules defined for workflow reasoning and policy compliance. Peak memory usage was recorded for each test to assess resource requirements. Table 6 presents the results.

As the dataset size increases, reasoning time grows in a predictable and monotonic pattern consistent with known performance characteristics of Pellet. Classification time increases moderately up to 10,000 individuals and then

Table 6. Reasoning time and memory usage of VICOn.

No. of individuals	Classification time (ms)	Consistency check (ms)	SWRL rule execution (ms)	Peak memory (MB)
1,000	210	95	180	310
10,000	1,120	430	1,020	480
50,000	5,850	2,380	5,120	860
100,000	12,900	5,800	11,400	1,420

more sharply at 50,000 and 100,000 individuals due to the increased number of class assertions and rule-triggering conditions. Consistency checking remains the fastest operation across all datasets, whereas SWRL rule execution constitutes the primary overhead, reflecting the expressive rule-based constraints embedded in VICOn. Memory consumption scales linearly with dataset size and remains within feasible operational limits for typical server configurations. Overall, the results indicate that VICOn maintains highly acceptable reasoning performance for datasets up to 50,000 individuals, with performance remaining manageable even at 100,000 individuals. These findings confirm that VICOn can be deployed in real-world insurance scenarios with large knowledge bases while preserving reasoning accuracy and stability.

4.3. Competency Question Coverage

To evaluate whether the VICOn ontology adequately satisfies the requirements specified during the ORSD phase, we conducted a CQ coverage assessment. During ontology development, a set of CQs was derived to capture essential information needs across key functional areas of vehicle insurance operations, including policy management, claim processing, workflow reasoning, stakeholder roles, document handling, and temporal constraints. These CQs articulate the questions that the ontology must be capable of answering through SPARQL queries and semantic inference. In this validation step, the CQs were grouped according to their corresponding knowledge domains, and each group was linked to the number of SPARQL queries implemented to assess coverage. For each group, we calculated the proportion of CQs that the ontology can successfully answer, providing a transparent view of coverage at both group and overall levels. Table 7 summarizes the CQ groups, their associated questions, the number of validation queries, and coverage percentages.

Across all groups, a total of 29 SPARQL queries were implemented to evaluate the ontology against the defined CQs. As shown in Table 7, the VICOn ontology achieves an average coverage score of 93.1%, indicating that it can satisfactorily answer the vast majority of questions derived from real-world vehicle insurance requirements. This demonstrates strong semantic completeness and confirms that the modeled concepts, properties, and reasoning rules sufficiently support the intended decision-making and information retrieval tasks within the VICBO system.

Table 7. Summary of CQs and coverage.

Group	Competency questions	No. of SPARQL queries	Coverage (%)
Policy & contract management	– Which policies are associated with a specific policyholder? – Was the policy active at the time of an incident? – What coverage types are included in a given policy?	5	80
Claim registration & incident information	– Which claims were submitted by a given policyholder? – Which claims are associated with a specific incident type (*e.g.*, collision)? – Which claims involve severe or total vehicle damage? – Which claims relate to a specific incident event?	4	100
Rule-based workflow & decision making	– Is a claim eligible based on the policy coverage? – Which claims require investigation due to missing documents? – Which policyholders submitted more than three claims in the past year? – What is the next workflow stage for a specific claim?	5	100
Vehicle information & damage assessment	– Which damage classifications are associated with a specific vehicle? – Which vehicles have multiple claims within the past year? – Is a vehicle repairable based on the documented damage?	3	100
Stakeholder roles & authorization	– Which stakeholders are involved in a specific claim? – Who is authorized to access a specific document?	2	100
Document & evidence management	– Which documents are linked to a specific claim? – Which claims are missing required evidence (*e.g.*, a police report)? – Which photos or repair estimates are associated with a claim?	4	75
Temporal & context-aware reasoning	– Did a claim occur within the policy's valid period? – Which policyholders have more than three claims in the past year? – Which claims were submitted after the policy expiration date? – Which incidents occurred outside the permissible reporting window?	6	100
		Sum = 29	Avg. = 93.1

4.4. Performance Experimentation on the VICBO System

4.4.1. Experimental Configurations

Hyperledger Fabric is a permissioned blockchain platform developed under the Linux Foundation's Hyperledger project[6]. It is designed to support enterprise-grade applications requiring modularity, scalability, and privacy. Unlike public blockchain platforms, Fabric introduces a flexible architecture that separates transaction execution, ordering, and validation processes. This separation enables better performance, confidentiality, and fine-grained control over access and data sharing among network participants. Fabric uses chaincode to define application logic, and it organizes participant organizations into peers, clients, and a centralized ordering service. The consensus mechanism is pluggable, allowing enterprises to adopt appropriate ordering protocols such as Solo, Kafka (deprecated), or Raft. Fabric also supports channels, which are private sub-networks allowing a subset of participants to share data privately. Figure 5 presents the transaction flow in Hyperledger Fabric, depicting the interaction between four core components: the client, endorsing peer, ordering service, and committing peer.

The process begins with the client preparing a transaction proposal and sending it to an endorsing peer. The endorsing peer simulates the execution of the transaction by invoking the chaincode and returns an endorsement response. The client collects the necessary endorsements and submits the transaction to the ordering service, which packages transactions into a block and broadcasts it to all relevant peers. The committing peers receive the block, validate endorsement policies and other criteria, and finally write the transaction to the ledger in the validate and commit phase. This workflow ensures that transaction execution is separated from ordering and committing, enhancing the modularity and scalability of the platform. As a result, Hyperledger Fabric is particularly well-suited for business process applications such as vehicle insurance.

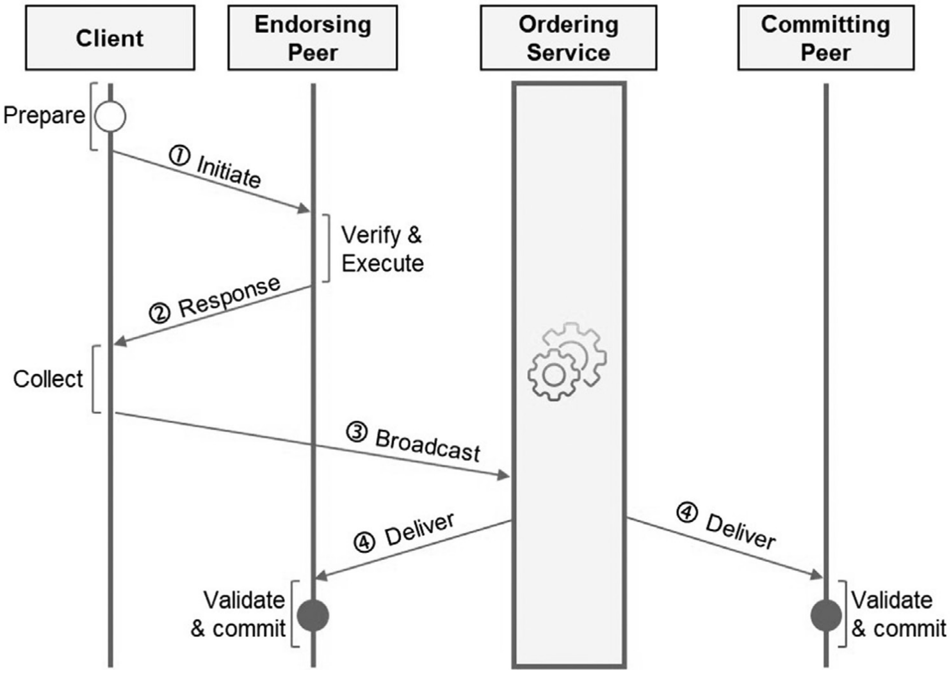


Figure 5. Transaction flow in Hyperledger Fabric.

---

[6] https://hyperledger-fabric.readthedocs.io/en/release-2.5/

The VICBO system is implemented following multi-layered architecture as introduced in Section 3 that integrates blockchain, smart contracts, knowledge, and off-chain storage. The system is deployed on a local machine running Ubuntu 22.04 LTS, with an Intel Core i7 processor, 32 GB of RAM, and 1 TB SSD. Hyperledger Fabric version 2.5 is used as the core blockchain platform, configured with two organizations, each hosting two peer nodes and one certificate authority. A Raft-based ordering service is used to achieve transaction consensus, and CouchDB is selected as the state database to enable flexible queries on ledger data. Chaincode, written in Go, implements essential insurance logic including policy creation, claim submission, status tracking, and conditional payment execution. The blockchain network is containerized using Docker Compose to simplify orchestration and scaling. Moreover, smart contracts are deployed through Fabric's chaincode framework and expose a set of transaction functions accessible via client applications through the Fabric SDK. All semantic verification steps, including ontology-based reasoning and document-hash extraction, are executed entirely off-chain within the client-side middleware. The middleware processes evidence, performs reasoning over the VICOn ontology, and submits the resulting outcomes (*e.g.*, compliance verdicts and document-hash proofs) as part of the transaction proposal to the blockchain network. Smart contracts then validate these inputs and record the final results on the ledger in a deterministic manner. RESTful endpoints facilitate communication between the client interface and peer nodes. To support adaptive reasoning, the system incorporates a semantic knowledge layer based on the VICOn ontology. The ontology is developed using Protégé 5.6.1 and follows the OWL 2 DL specification while the Pellet reasoner is employed for reasoning. To address scalability and privacy concerns in evidence management, VICBO integrates an off-chain storage layer utilizing Web3.Storage[7], a service built on IPFS and Filecoin. Documents such as police reports, medical evaluations, and repair invoices are encrypted and uploaded to Web3.Storage, which generates a unique CID to ensure document integrity. These CIDs are referenced within smart contracts and semantically linked to their corresponding claims through properties defined in the ontology.

4.4.2. Experimental Results

To evaluate the performance of the VICBO system, we focus on two key operational metrics: throughput and latency. These metrics are widely adopted in performance evaluations of blockchain and Hyperledger Fabric-based systems [18], [20] and are used to assess the responsiveness and efficiency of the system under varying transaction loads. Throughput measures how many transactions the system can successfully handle per second, and is defined as:

$$Throughput(TPS) = \frac{N_{successful}}{T_{total}} \qquad (1)$$

where $N_{successful}$ is the number of successfully processed transactions, and T_{total} is the total time period in seconds during which those transactions were submitted and confirmed. This metric helps evaluate the system's scalability under different workloads.

Latency quantifies the average delay between when a transaction is submitted and when it is confirmed on the blockchain. It is calculated as:

$$Latency(s) = \frac{1}{N}\sum_{i=1}^{N}(T_i^{commit} - T_i^{submit}) \qquad (2)$$

Here, T_i^{submit} and T_i^{commit} are the timestamps for the submission and confirmation of the i-th transaction, respectively. This metric provides insight into how quickly the system responds to user actions and confirms events.

We conducted experiments on two major operations within VICBO: claim verification and compensation payment. In the claim verification operation, the measured time comprises several components. First is the on-chain ex-

---

[7] https://web3.storage/

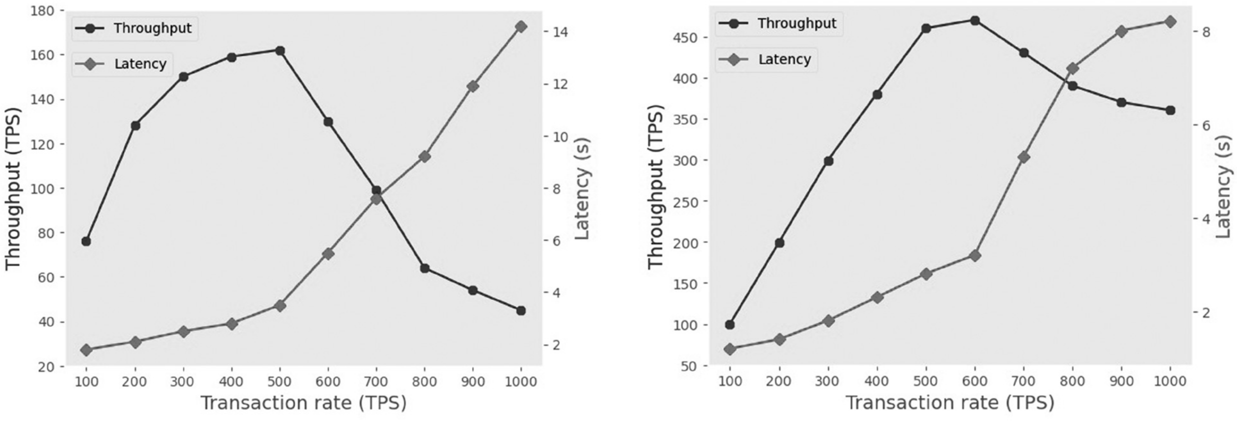
ecution time required to record the claim and submit associated hash values to the blockchain. This is followed by the semantic reasoning phase, which includes the execution of SWRL rules and ontology-based inference using the Pellet reasoner. As part of this reasoning process, the system also performs off-chain data retrieval, specifically fetching and verifying document hashes stored in IPFS. In contrast, the compensation payment operation consists solely of on-chain smart contract execution, where the system transfers tokens directly to the repair shop once a claim is approved. This operation does not invoke any semantic reasoning or access external data sources, resulting in a more streamlined execution path. For each operation, throughput and latency were measured under increasing transaction rates to simulate realistic usage scenarios. Each simulation was executed 1,000 times to ensure statistical reliability and consistency in the performance measurements.

Figure 6 illustrates the performance evaluation of claim verification and compensation payment operations in VICBO, showing throughput and latency trends under varying transaction rates.

In Figure 6(a), the claim verification operation demonstrates an initial increase in throughput as the transaction rate rises, peaking at approximately 160 TPS around the 500 TPS input rate. Beyond this point, throughput begins to decline, indicating that the semantic reasoning and IPFS-based document verification processes have become bottlenecks under high load. Correspondingly, latency remains relatively low and stable at lower rates but increases sharply beyond 500 TPS, reaching over 14 seconds at the highest input rate. This trend reflects the cumulative overhead introduced by the integrated ontology reasoning and off-chain data access during claim processing.

In Figure 6(b), the compensation payment operation shows a more scalable performance profile. Throughput increases nearly linearly up to around 450 TPS and begins to plateau only slightly at higher input rates. Since this operation relies solely on on-chain smart contract execution without invoking semantic or off-chain components, it maintains higher throughput and lower latency across the board. Latency increases gradually from under 2 seconds at 100 TPS to approximately 8 seconds at 1000 TPS, primarily due to block confirmation time and smart contract execution queuing within the Hyperledger Fabric framework.

Overall, Figure 6 demonstrates that VICBO maintains acceptable performance for both operations under moderate to high transaction rates, with claim verification being more sensitive to computational and input/output overhead, and compensation payment showing stronger scalability under increasing load.



(a) Claim verification.

(b) Compensation payment.

Figure 6. Throughput and latency of claim verification and compensation payment under varying transaction rates.

5. Conclusion

This paper presents VICBO, an integrated system for vehicle insurance claim management that leverages the immutability and transparency of blockchain technology alongside the adaptability and context-awareness enabled by semantic reasoning. The proposed architecture combines four key layers, including blockchain, smart contract, knowledge, and off-chain storage to support trustworthy, automated, and semantically rich claim processing. By modeling insurance policy knowledge through the VICOn ontology and encoding eligibility and compliance logic using SWRL rules, the system enables dynamic, automated claim evaluations that surpass the capabilities of static smart contracts. A detailed workflow was proposed to demonstrate how VICBO processes insurance claims from contract agreement to compensation payment, incorporating semantic validation and off-chain document handling via IPFS. The VICOn ontology was engineered using the NeOn methodology and evaluated through SPARQL-based scenario testing, showcasing its ability to handle complex requirements such as jurisdiction-dependent documentation, class-based vehicle eligibility, and behavioral risk detection. Performance evaluation results confirm the system's practicality. While the integration of reasoning and off-chain data introduces increased latency in claim verification under high transaction loads, compensation payments remained efficient due to their purely on-chain execution. The overall throughput and latency remained within acceptable thresholds, demonstrating VICBO's readiness for real-world deployment in enterprise-level insurance environments.

Overall, this study's main contribution lies in demonstrating how a hybrid blockchain-ontology architecture can provide not only verifiability and automation, but also semantic adaptability, an essential capability missing from existing blockchain-based insurance frameworks. VICBO shows that integrating reasoning with decentralized execution can significantly enhance transparency, consistency, and decision support in complex claim management scenarios.

Future work will explore the optimization of reasoning workflows, such as pre-filtering and incremental inference, and the integration with real regulatory frameworks and support for multi-party governance. Additionally, the application of learning-based techniques to enhance fraud detection will be investigated.

Acknowledgment

The author gratefully acknowledges the valuable contributions of the domain experts and ontology engineers who participated in and supported this study.

Declaration of Competing Interests

The author declares no conflict of interest.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

The VICBO ontology developed in this study is available upon request from the author.

References

[1] A. Shetty *et al.*, "Blockchain Application in Insurance Services: A Systematic Review of the Evidence", *Sage Open*, vol. 12, no. 1, 2022. http://dx.doi.org/10.1177/21582440221079877

[2] V. Gatteschi *et al.*, "Blockchain and Smart Contracts for Insurance: Is the Technology Mature Enough?", *Future Internet*, vol. 10, no. 2, 2018. http://dx.doi.org/10.3390/fi10020020

[3] M. Demir *et al.*, "Blockchain Based Transparent Vehicle Insurance Management", in *Proc. of the 6th International Conference on Software Defined Systems (SDS)*, 2019, pp. 213–220. http://dx.doi.org/10.1109/SDS.2019.8768669

[4] R. Roriz and J. L. Pereira, "Avoiding Insurance Fraud: A Blockchain-based Solution for the Vehicle Sector", *Procedia Computer Science*, vol. 164, pp. 211–218, 2019. http://dx.doi.org/10.1016/j.procs.2019.12.174

[5] A. S. Yadav *et al.*, "Blockchain-based Secure Privacy-Preserving Vehicle Accident and Insurance Registration", *Expert Systems with Applications*, vol. 230, p. 120651, 2023. http://dx.doi.org/10.1016/j.eswa.2023.120651

[6] A. Hassan *et al.*, "Secured Insurance Framework Using Blockchain and Smart Contract", *Scientific Programming*, vol. 2021, no. 1, p. 6787406, 2021. http://dx.doi.org/10.1155/2021/6787406

[7] N. Nizamuddin and A. Abugabah, "Blockchain for Automotive: An Insight Towards the IPFS Blockchain-based Auto Insurance Sector", *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, 2021.

[8] C.-L. Chen *et al.*, "A Blockchain and IPFS-Based Anticounterfeit Traceable Functionality of Car Insurance Claims System", *Sensors*, vol. 23, no. 23, 2023. http://dx.doi.org/10.3390/s23239577

[9] M. Al-Thawadi *et al.*, "A-SHIP: Ontology-Based Adaptive Sustainable Healthcare Insurance Policy", *Sustainability*, vol. 14, no. 3, 2022. http://dx.doi.org/10.3390/su14031917

[10] E. Chondrogiannis *et al.*, "Using Blockchain and Semantic Web Technologies for the Implementation of Smart Contracts Between Individuals and Health Insurance Organizations", *Blockchain: Research and Applications*, vol. 3, no. 2, p. 100049, 2022. http://dx.doi.org/10.1016/j.bcra.2021.100049

[11] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", *Decentralized Business Review*, p. 21260, 2008.

[12] N. Szabo, "The Idea of Smart Contracts", *Nick Szabo's Papers and Concise Tutorials*, vol. 6, no. 1, 1997.

[13] E. Androulaki *et al.*, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains", in *Proceedings of the 13th EuroSys Conference (EuroSys '18)*, New York, NY, USA: Association for Computing Machinery, 2018. http://dx.doi.org/10.1145/3190508.3190538

[14] H. Qi *et al.*, "Scalable Decentralized Privacy-Preserving Usage-Based Insurance for Vehicles", *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4472–4484, 2021. http://dx.doi.org/10.1109/JIOT.2020.3028014

[15] H. Zheng *et al.*, "A Novel Insurance Claim Blockchain Scheme Based on Zero-Knowledge Proof Technology", *Computer Communications*, vol. 195, pp. 207–216, 2022. http://dx.doi.org/10.1016/j.comcom.2022.08.007

[16] F. Lamberti *et al.*, "Blockchains Can Work for Car Insurance: Using Smart Contracts and Sensors to Provide On-Demand Coverage", *IEEE Consumer Electronics Magazine*, vol. 7, no. 4, pp. 72–81, 2018. http://dx.doi.org/10.1109/MCE.2018.2816247

[17] X. Liu *et al.*, "A Blockchain-Based Auto Insurance Data Sharing Scheme", *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, p. 3707906, 2021. http://dx.doi.org/10.1155/2021/3707906

[18] C. Huang *et al.*, "Blockchain-Assisted Personalized Car Insurance With Privacy Preservation and Fraud Resistance", *IEEE Transactions on Vehicular Technology*, vol. 72, no. 3, pp. 3777–3792, 2023. http://dx.doi.org/10.1109/TVT.2022.3215811

[19] O. Bhadra *et al.*, "Decentralized Insurance Subrogation Using Blockchain", in *Proceedings of the 2022 5th International Conference on Blockchain Technology and Applications (ICBTA '22)*, New York, NY, USA: Association for Computing Machinery, 2023, pp. 1–9. http://dx.doi.org/10.1145/3581971.3581972

[20] M. D. Nguyen *et al.*, "BOCA: A Novel Semantic Blockchain-Based Authentication System of Educational Certificates", *International Journal of Computers and Applications*, vol. 44, no. 11, pp. 1074–1082, 2022. http://dx.doi.org/10.1080/1206212X.2022.2111509

[21] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications", *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993. http://dx.doi.org/10.1006/knac.1993.1008

[22] W. Van Woensel and O. Seneviratne, "Semantic Interoperability on Blockchain by Generating Smart Contracts Based on Knowledge Graphs", arXiv preprint arXiv:2409.12171, 2024.

[23] H. M. Kim and M. Laskowski, "Toward an Ontology-Driven Blockchain Design for Supply-Chain Provenance", *Intelligent Systems in Accounting, Finance and Management*, vol. 25, no. 1, pp. 18–27, 2018. http://dx.doi.org/10.1002/isaf.1424

[24] M. Finck, *Blockchain and the General Data Protection Regulation: Can Distributed Ledgers Be Squared With European Data Protection Law?* Luxembourg: Publications Office of the European Union, 2019.

[25] M. C. Suárez-Figueroa *et al.*, "The NeOn Methodology Framework: A Scenario-Based Methodology for Ontology Development", *Applied Ontology*, vol. 10, no. 2, pp. 107–145, 2015.

[26] E. Sirin *et al.*, "Pellet: A Practical OWL-DL Reasoner", *Journal of Web Semantics*, vol. 5, no. 2, pp. 51–53, 2007. http://dx.doi.org/10.1016/J.WEBSEM.2007.03.004

Received: July 2025
Revised: January 2026
Accepted: February 2026

Contact addresses:
Minh Duc Nguyen
Department of Economic Information Systems
University of Economics
Hue University
Hue
Vietnam
e-mail: nguyenminhduc@hueuni.edu.vn

MINH DUC NGUYEN received his M.Eng. and PhD degrees in Information Systems from Inje University, Korea, in 2017 and 2021, respectively. He is currently a lecturer at the Department of Economic Information Systems, University of Economics, Hue University, Hue, Vietnam. His research interests include data mining, ontology engineering, semantic technologies, knowledge management, and blockchain systems.
