Integrating Blockchain Smart Contracts and Ontology to Ensure Transparency and Integrity in Administration

Oluseyi Ayodeji Oyedeji
Department of Computer Science,
University of Ibadan
Ibadan, Nigeria

Ibiyinka Temilola Ayorinde
Department of Computer Science,
University of Ibadan
Ibadan, Nigeria

ABSTRACT
This paper presents an ontology-driven administrative monitoring system that integrates blockchain smart contracts to ensure transparency, accountability, and integrity in organizational processes. This tracks resources and manages administrative workflows in an open and decentralised manner, addressing long-standing governance challenges such as opacity and unaccountability. Many important administrative processes involve the movement of resources from one point to another; these resources and processes can be modelled similarly to the movement of goods in a supply chain. Motivated by the need to improve institutional governance, particularly in contexts where individual actions often undermine fairness, this study combines ontology and blockchain to formalize administrative processes and enhance traceability. A pre-created ontology from the author's previous work, developed for postgraduate administration at the University of Ibadan, was adopted. The ontology defines classes and activities which were translated into blockchain entities and smart contracts, implemented in Solidity, and deployed on the Ethereum test network. Test cases derived from ontology competency questions validate the functional correctness of the smart contracts. The results confirm that administrative activities can be monitored transparently and immutably, providing a foundation for broader applications in public administration, education, and corporate governance.

Keywords
Blockchain, Ontology, Smart Contracts, Information Systems, Administrative Processes

1. INTRODUCTION
The level of productivity within an organisation is directly linked to the quality of its administration. Technology has undoubtedly transformed how most organisational processes are conducted [1]. It remains one of the major enablers of innovation, helping establishments adopt methods that drive growth and efficiency [2].

Most administrative processes involve the movement of digital assets such as memos, requests, approvals, access rights, and documents from one office, department, or stakeholder (source) to another (destination). This can be conceptualised similarly to the movement of goods in a supply chain, as illustrated in Figure 1. Hence, a concept called the administrative chain can be used to describe the flow of administrative resources from one point to another, enabling the application of blockchain consensus and monitoring mechanisms to optimise such processes.

Blockchain is a data structure with a distributed architecture that is replicated and shared among all members of a network [3]. When data is stored on a distributed architecture such as the blockchain, it extends far beyond its common use in cryptocurrency. One of its most significant features is the smart contract, which allows for trusted, automatic interactions between entities [4].

Another foundational principle of blockchain is decentralisation, which involves removing unnecessary control from any single player within a system. If a process can operate without centralised authority, control should be distributed. Additional features such as trust, tamper-proof storage, openness, and transparency make blockchain an ideal technology for building transparent administrative systems.

The administrative structure of the Postgraduate College serves as a prototype of a typical institutional administration and can benefit from combining ontology and blockchain technologies. Ontology and blockchain can be applied separately yet complementarily to improve the efficiency of the administrative workflow. Currently, there is a dearth of research exploring their joint application to model and explicitly represent operations within administrative systems. This gap has led to inefficiencies in the flow of operational data, difficulties in identifying problems, and limited visibility into document and request movements.

This paper presents the development of a transparent administrative monitoring system that integrates ontology and blockchain technology. The specific objectives are to:

1) Identify and adopt a relevant administrative ontology.
2) Implement blockchain smart contracts that enhance transparency based on the activities defined within the ontology using the Ethereum Virtual Machine; and
3) Validate the implemented blockchain system through test cases derived from ontology competency questions.

This work provides a standardised and reusable knowledge structure along with an immutable and transparent monitoring system for administrative domains. It enables timely identification of bottlenecks, conflicts, delays, and deadlocks by tracking where digital assets are located at every point in time.

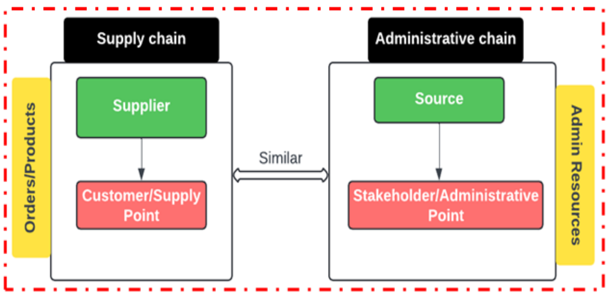

Figure 1: Conceptualisation of Administration as chain of events

2. LITERATURE REVIEW

2.1 Distributed Systems

Distributed systems are systems in which components or resources are spread across multiple networked computers and communicate with each other only by-passing messages, while appearing to users as a single coherent system [5]. The distributed approach reduces dependency on the physical location of systems, lowers operational costs, increases reliability, and improves fault tolerance.

In earlier computing paradigms, systems were often built using sequential programming. However, as software complexity increased and hardware evolved, the shift to parallel programming enabled the exploitation of multicore processors. The increasing computational requirements of modern applications soon exceeded the capabilities of a single node, leading to a move toward distributed computing [6]. Distributed systems thus provide a way to aggregate computational power across many interconnected machines, creating the foundation for decentralized and fault-tolerant architectures. This concept is critical to blockchain technology, which leverages distribution to maintain data integrity and consensus across nodes.

2.2 Decentralised Systems and Blockchain Technology

Decentralised systems emerged as an alternative to the limitations of centralised architectures, where a single point of control or failure can compromise the reliability of an entire system. In decentralised systems, decision-making and validation are distributed among participants, allowing for greater transparency and resilience. This paradigm is at the core of blockchain and related technologies such as the Interplanetary File System (IPFS), which have been applied across diverse fields including finance, supply chains, education, and governance [7]. These systems exploit blockchain affordances such as transparency, tokenisation, codification of trust, and distributed infrastructure [8].

Early blockchain applications were primarily financial, but subsequent projects extended its use to diverse domains including scientific publication [9] [10]. Blockchain can be described as a distributed ledger, a type of distributed database maintained across a network of peers through a synchronisation mechanism [11]. Like the Internet, public blockchains form open infrastructures that are not owned or controlled by any single entity. Each node maintains a copy of the ledger, enabling transparency and resistance to tampering.

The blockchain ledger consists of a sequence of blocks, each containing a set of validated transactions. Every block references its predecessor through a cryptographic hash known as the parent block, thereby forming a chain of immutable records [12]. The first block in a blockchain is called the genesis block, which has no parent. This chained architecture ensures that any attempt to alter data in one block would require modification of all subsequent blocks which is an infeasible task due to computational and network constraints.

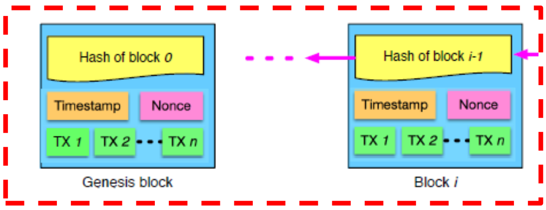

Figure 2: Blockchain Architecture [11]

2.2.1 Characteristics of Blockchain Networks

Previous works identified several key properties that distinguish blockchain networks [11]:

1. Decentralisation: Unlike centralised transaction systems that rely on a trusted intermediary (e.g., a central bank), blockchain transactions occur directly between peers (P2P), eliminating bottlenecks and reducing operational costs.
2. Persistency: Each transaction must be validated by nodes and recorded across the entire network, making data tampering extremely difficult.
3. Anonymity: Users interact through pseudonymous addresses without centralised identity management, preserving privacy while ensuring accountability.
4. Auditability: Transactions are timestamped and traceable, enhancing transparency and allowing any user to verify historical records.

2.2.2 Types of Blockchain Networks

Blockchain networks can be classified into three main types. They are public, private, and consortium, each with different levels of permission and centralisation [11].

1. In public blockchains, all nodes can participate in consensus and view transactions.
2. Private blockchains are controlled by a single organisation, which determines participation and consensus mechanisms.
3. Consortium blockchains fall in between, with pre-selected validators from multiple institutions managing consensus.

These network types differ in efficiency, scalability, and immutability. Public blockchains offer the highest transparency but lower transaction throughput, while private and consortium blockchains trade openness for efficiency [11] [13].

The rise of blockchain represents a paradigm shift in distributed computing, providing mechanisms for secure, transparent, and tamper-proof transactions without reliance on a central authority [14].

2.3 Blockchain for Governance

Initially popularised through cryptocurrencies, blockchain technology is now increasingly applied to governance systems. Its core properties, immutability, traceability, and decentralization, align with the values of accountable and transparent governance. Blockchain's potential for reforming administrative processes lies in its ability to record decisions, transactions, and approvals in a verifiable, tamper-proof ledger.

A notable evolution in blockchain governance is the Decentralised Autonomous Organisation (DAO), which uses smart contracts to automate decision-making and enforce transparent rules without central authority [14]. These organisations exemplify how blockchain can operationalise trust and transparency, offering models for governance structures that are less prone to corruption or bias.

2.4 Ethereum Blockchain and Smart Contracts

Ethereum extends blockchain beyond value transfer to general-purpose computation [15]. It enables small programs, called smart contracts to be stored and executed on-chain. These contracts, written in high-level languages such as Solidity, define deterministic rules that are automatically executed when specified conditions are met. Once deployed, they are immutable and transparent to all network participants [16].

In the Ethereum network, all nodes execute the same contract code to maintain consensus about state changes in the distributed ledger. This architecture makes Ethereum a decentralised computing platform rather than a mere transaction ledger. Smart contracts thus facilitate the creation of decentralised applications (dApps) for prediction markets [17] [18], social networks [19], and more recently, administrative systems.

The ability to encode administrative logic within smart contracts presents an opportunity to re-engineer traditional bureaucratic systems for greater efficiency and accountability. This study leverages Ethereum's capabilities to implement smart contracts derived from an ontology of administrative activities.

2.5 Related Works

2.5.1 Dynamic Ontology for Knowledge Representation in Establishments

Previous work [20] proposed a dynamic ontology for representing organisational structures and operations. Their research addressed the representational problems involving occurrents such as states, events, and processes, suggesting that states be represented using temporal arguments and events through Davidson-style reification. Implemented in Prolog using first-order logic, their approach contributed to the formal understanding of organisational knowledge structures.

However, their model lacked transparency and was not distributed, limiting its applicability for real-time, multi-agent administrative systems. Furthermore, the ontology could not be directly expressed in Web Ontology Language (OWL) due to its logic-based implementation. This present work addresses these limitations by integrating ontology-based knowledge representation with blockchain technology, achieving both transparency and distribution through smart contracts.

2.5.2 Statistical Complexity in Blockchain Ecosystem

A previous work [21] examined blockchain from the perspective of information theory, analysing it using Crutchfield's Statistical Complexity measure. The findings indicated that while blockchain is algorithmically sophisticated, it is not a chaotic system and exhibits relatively low complexity. This theoretical work is significant because it provides a formal basis for understanding blockchain behaviour, though it does not directly address its application in administrative or semantic domains.

2.5.3 Blockchain Technology and Supply Chain

Another work [22] explored blockchain applications within the supply chain and logistics sectors, identifying how blockchain can address inefficiencies through transparency, traceability, and decentralisation. The study outlined how blockchain enhances trust among network participants, ensuring that every transaction is recorded and verifiable.

The analogy between supply chains and administrative workflows is particularly relevant here: administrative processes can be seen as the movement of digital assets (e.g., requests, memos, approvals) through different nodes (departments or offices). This concept forms the basis for the administrative chain proposed in the current research.

2.5.4 Blockchain Types and Challenges

A previous work [13] presented a systematic review of blockchain technologies, identifying challenges such as scalability, security, and interoperability. Similarly, another work [23] discussed typical consensus algorithms and technical bottlenecks. Both studies concluded that blockchain adoption would grow gradually, with many startups failing before widespread institutional integration. These studies underscore the importance of domain-specific prototypes that demonstrate blockchain's practical utility beyond cryptocurrencies.

2.5.5 Enhancing Blockchain with Ontologies

Previous research [24] has argued that ontologies can significantly improve blockchain design by formalising traceability concepts and encoding them into smart contracts. Their research translated elements of a traceability ontology into Ethereum smart contracts, demonstrating how semantic constructs can enhance provenance tracking. However, they acknowledged the need for more systematic conversion processes between semantic web representations (e.g., OWL, RDF) and blockchain code.

Similarly, another work [25] proposed a governance ontology embedded as metadata tags within smart contracts to facilitate interoperability between blockchain protocols. They urged greater collaboration between the blockchain and ontology communities to bridge semantic and protocol-level gaps. It was also theorised that blockchain could be the missing component for realising the semantic web, enabling reasoning over decentralised data [26].

These works collectively indicate a growing research interest in integrating ontology and blockchain, though practical implementations in governance or administrative systems remain limited. The present study contributes by implementing an ontology-driven blockchain prototype for institutional administration.

2.5.6 Blockchain and Governance Applications

Previous works [27] have proposed applying blockchain to e-government interoperability. Their system, Ontology Blockchain, used Python-based service orchestration and semantic policy analysis to ensure consistency across governmental services. Similarly, it has been examined [28] how blockchain could support sustainable governance by promoting transparency and reducing corruption. These studies reinforce blockchain's potential to transform administrative systems, though most remain conceptual. This work

extends this trajectory through a concrete implementation validated with ontology-derived test cases.

2.5.7 Summary of Literature Gap

The reviewed literature reveals substantial progress in blockchain theory, ontology design, and decentralised governance. However, few studies have combined ontology-based knowledge representation with blockchain's immutability and decentralisation to model and automate administrative processes. Existing ontology-based administrative systems [20] lack distributed trust, while blockchain-based governance systems [27] [28] lack semantic depth.

This research bridges that gap by developing an ontology-driven blockchain system that enables transparent, traceable, and accountable administration. By translating ontology-defined activities into blockchain smart contracts, the admin monitoring system demonstrates how semantic structures can be operationalised for governance reform and institutional transparency.

3. METHODOLOGY

3.1 System Architecture

The overall architecture of the administrative monitoring system is illustrated in Figure 3. The architecture shows how processes and activities defined in the Administrative Process Ontology are extracted, translated, and implemented within a blockchain environment [29] [30] [1]. These processes are modelled as blockchain entities and executed as smart contracts on the Ethereum Virtual Machine (EVM) using the Solidity programming language.

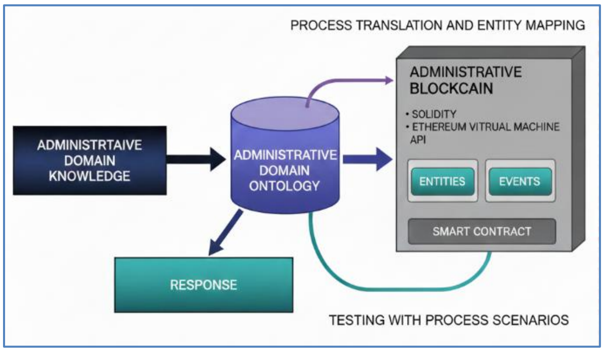

Figure 3: Overall System Architecture

The architecture consists of three primary layers:

1. Ontology Layer: where administrative knowledge is represented through classes, properties, and relationships that capture real-world administrative processes.

2. Integration Layer: which maps ontology entities to blockchain components, performing entity translation and semantic alignment.

3. Blockchain Layer: where smart contracts execute administrative operations, manage transactions, and record immutable events.

The entire design ensures that semantic accuracy from the ontology is preserved while gaining transparency and immutability from the blockchain. Requests are issued to smart contracts through the integration layer, and responses are retrieved through event triggers logged in the blockchain.

3.2 Mapping Ontology Concepts to Blockchain Entities

The Administrative Process Ontology developed in earlier research [29] represents the knowledge structure of a university postgraduate administration. Entities such as ApplicantStudent, EligibleStudent, Supervisor, RecordOfficer, FinanceOfficer, and Dean were modelled as classes, while activities such as verification, recommendation, approval, and clearance were captured as object properties linking these classes.

Figure 4 depicts the transformation of these ontology concepts into blockchain entities.

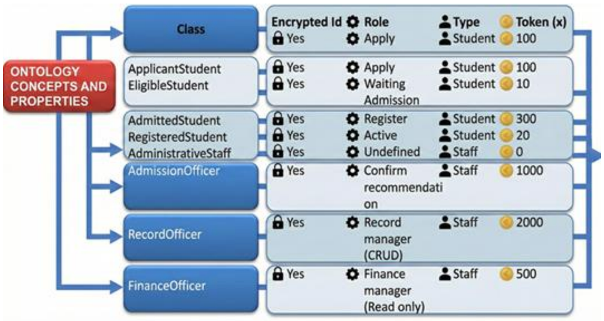

Figure 4: Transformation of Ontology Concepts into Blockchain Entities

The mapping process involved the following steps:

1. **Entity Identification**: Classes in the ontology were identified as potential smart contracts.
2. **Property Translation**: Object properties were mapped to contract variables or functions representing relationships between contracts.
3. **Role Assignment**: Each actor (e.g., Student, Supervisor, FinanceOfficer) was assigned a blockchain address and cryptographic key.
4. **Privacy Enforcement**: Each entity was given an Encrypted Public Address generated within the Ethereum test network to ensure secure identity representation.
5. **Tokenisation**: Actions such as submission, review, or approval were represented as token-based transactions. These tokens symbolise digital proof of participation within the administrative process.

This mapping ensures that every ontology relationship is operationalised within the blockchain while preserving the semantics of the administrative domain.

3.3 Ontology and Blockchain Integration

Ontology provides a formalised representation of domain knowledge, enabling machines to interpret and reason about organisational concepts. In this work, ontology serves as the semantic foundation for all administrative activities. It ensures standardisation, reusability, and interoperability across systems.

Blockchain, on the other hand, provides a distributed, tamper-resistant execution layer that guarantees the integrity and auditability of all operations. The combination of ontology and blockchain thus establishes a dual structure: ontology for knowledge representation and blockchain for execution integrity.

The integration process involved:

- **Semantic Extraction**: extracting all relevant entities, relationships, and axioms from the ontology.
- **Contract Modelling**: designing Solidity smart contracts that mirror ontology classes and relationships.
- **Function Design**: creating functions that implement administrative activities (e.g., submitApplication(), verifyRecord(), approveAdmission()).
- **Event Logging**: defining events such as onSubmission, onApproval, and onCompletion to record every state change on the blockchain.

The Ethereum blockchain was selected because it supports Turing-complete smart contracts and provides an Application

Programming Interface (API) for programmatic interactions, including:

- creation of blockchain addresses,
- retrieval of transaction histories,
- monitoring of contract events,
- querying of token balances, and
- retrieval of mined block information.

These features make Ethereum suitable for modelling complex workflows such as administrative processes that involve multiple interdependent participants.

3.4 Implementation Environment

The system was implemented and tested using the following tools and technologies as seen in Table 1:

Table 1: Implementation information

Component	Description
Blockchain Platform	Ethereum Test Network (Ropsten / Goerli)
Smart Contract Language	Solidity v0.8.x
Development IDE	Remix IDE (browser-based environment)
Blockchain Client	Ganache (for local blockchain simulation)
Wallet and Transaction Interface	MetaMask browser extension
API Interaction	Web3.js for blockchain–frontend communication
Testing Token	Virtual Ether (simulated alt-tokens, no real cryptocurrency)

The implementation process followed these steps:

1. The ontology file (OWL format) was parsed to extract entity names, roles, and properties.
2. Corresponding Solidity smart contracts were designed, with functions representing ontology activities.
3. Contracts were compiled and deployed using Remix IDE connected to MetaMask configured for the Goerli Test Network.
4. Ganache was used to simulate local transactions for early testing.
5. Web3.js scripts were employed to interact with the contracts, send transactions, and retrieve logs.

Each smart contract was programmed to emit events upon successful execution, thereby serving as a proof of activity and enabling traceability across the blockchain ledger.

3.5 Validation Using Ontology-Derived Scenarios

Validation was achieved using Competency Questions (CQs) that were originally designed to verify the correctness of the Administrative Process Ontology. These CQs were repurposed to test the blockchain implementation, thereby ensuring semantic equivalence between the ontology and the deployed system.

Examples of validation questions include:

- Which student applications are pending approval?
- Who is the officer responsible for this request?
- Has a particular student completed financial clearance?

Each question was translated into a blockchain query and executed as a smart-contract call. For instance:

- `getPendingApprovals()` retrieves all application addresses awaiting approval.
- `getLastActionByOfficer(address officer)` verifies the latest transaction associated with an officer.
- `isFinanciallyCleared(address student)` checks the clearance status of a student.

Figure 5 shows how these competency questions were implemented within the testing process.

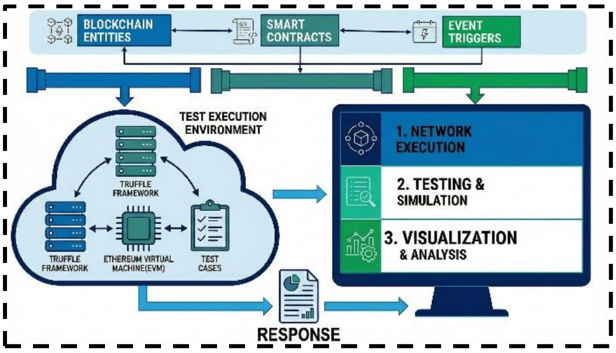

Figure 5: Testing The Blockchain with Competency Questions

Successful execution of these queries confirmed that the blockchain produced responses identical to those expected from the ontology reasoning layer. This demonstrates that the mapping and translation mechanisms preserved both semantics and logic. The result is a transparent, immutable, and verifiable administrative monitoring system.

3.6 Security and Integrity Considerations

To maintain integrity within the test environment, several safeguards were implemented:

- All user identities were encrypted using public-key cryptography.
- Transactions were digitally signed to ensure authenticity.
- Event logs were timestamped automatically by the Ethereum network, providing verifiable traceability.
- Each function was associated with a specific role, restricting access and preventing unauthorised execution.
- Data redundancy was ensured by the distributed nature of the blockchain ledger.

These measures ensure that administrative operations cannot be altered or repudiated after execution, supporting trustless verification across participating stakeholders.

3.7 Methodology Summary

The methodology bridges semantic modelling and decentralised execution. Ontology defines administrative processes formally and consistently, while blockchain ensures that these processes are executed transparently and immutably. The combination results in the system, a blockchain-enabled administrative monitoring system capable of detecting process bottlenecks, verifying responsibilities, and improving institutional accountability.

4. RESULTS AND DISCUSSIONS

The developed framework is structured into five major categories derived directly from the administrative ontology:

1. General-information smart contracts
2. Document smart contracts

3. Stakeholder smart contracts
4. Activity smart contracts
5. Event smart contracts

All contracts were implemented using Solidity (v0.8.x), compiled and tested on Remix IDE, deployed to the Ethereum Virtual Machine (EVM), and executed on the Goerli Testnet using MetaMask and faucet-provided test ETH. The ontology specification [29] served as the semantic foundation for the translation of concepts and activities into blockchain entities.

4.1 General Information Contracts

General information contracts define foundational administrative metadata such as aliases, duties, and conceptual equivalence relations. They correspond to high-level ontology constructs and support shared meaning across the system.

4.1.1 Sample Contract 1: PgAdminAlias

Table 2 summarises the PgAdminAlias contract, responsible for creating and retrieving equivalence terms.

Table 2. Postgraduate Concept Alias Contract

Description	Concept Alias
Ontology Reference	Equivalence
Name	PgAdminAlias { }
Functions	CreateAlias(), UpdateAlias() etc.

4.1.2 Sample Contract 2: PgDuties

Table 3 summarises the PgDuties smart contract, responsible for managing information about the duties of the organisation.

Table 3. Postgraduate Duties Contract

Description	Information about Postgraduate College Duties
Ontology Reference	Postgraduate Duty (Class)
Name	PgDuties { }
Functions	CreateDuty(), UpdateDuty() etc.

4.1.3 Validation of Sample General Information Contracts

Validation of the general information layer is carried out by testing the smart contracts in Tables 2 and 3. The objective here is to ensure that the ontology rules are correctly translated into blockchain smart contracts. Hence, the competency questions from the ontology validation stage are used to formulate test cases for validating smart contracts. The returned results from the test cases are compared with the answers from the ontology competency questions (CQ).

Test Case 1

CQ: *What is another name for the concept University within the context of the Postgraduate Administration Ontology?*

- Step a: Connect to contract PgAdminAlias { }
- Step b: Execute CreateAlias ('University', 'Central University')
- Step c: Execute GetAlias('University')

Result: *Central University*

This test confirms that semantic equivalence defined in the ontology is correctly preserved and retrievable through the alias management smart contract.

Test Case 2

CQ: *Who manages the postgraduate branch of the University?*

- Step a: Connect to contract PgDuties {}
- Step b: Execute CreateDuty ('Postgraduate Branch Management', 'Postgraduate College')
- Step c: Execute GetDutyCoordinator ('Postgraduate Branch Management')

Result: *Postgraduate College*

This validation verifies that administrative responsibilities defined in the ontology are accurately encoded and enforced through the postgraduate duties smart contract.

4.2 Document Smart Contracts

Document smart contracts manage the movement, location, auditing, and lifecycle of administrative documents. They implement the ontology definitions for Document, DocumentTrail, and associated relations.

4.2.1 Sample Contract 3: PgAdminDocument

Table 5 summarises the Document smart contract, responsible for managing the documents moving around the organisation.

Table 5. Postgraduate Document Contract

Description	Documents used in Postgraduate Administration
Ontology Reference	Document (Class)
Name	PgAdminDocument {}
Functions	InitiateDocument (), MoveDocument () etc.

4.2.2 Validation of Document Contract

The document smart contract is validated by converting document-related competency questions identified during the ontology development stage into corresponding smart contract test cases, after which the generated responses are evaluated.

Test Case 3

CQ: *What is the current location of document RQ101123?*

- Step a: Connect to the PgAdminDocument {} smart contract
- Step b: Execute GetDocumentCurrentLocation(RQ101123)

Result: Finance Section, Postgraduate College, University of Ibadan

This test confirms that the smart contract correctly tracks and retrieves the current location of an administrative document, demonstrating transparent document movement across administrative units.

Test Case 4

CQ: *What is the previous location of document INV24533?*

- Step a: Connect to the PgAdminDocument {} smart contract
- Step b: Execute GetDocumentPreviousLocation(INV24533)

Result: Admission Section, Postgraduate College, University of Ibadan

This validation verifies that the historical trail of a document is preserved on the blockchain, thereby enabling reliable auditing and provenance tracking.

4.3 Stakeholder Smart Contracts

These contracts model organisational actors, departments, and hierarchical structures within the administrative domain.

4.3.1 Sample Contract 4: UniversityDepartment

Table 6 summarises the Department smart contract, responsible for managing information about the departments.

Table 6. University Department Contract

Description	University Department linked with PG Administration
Ontology Reference	University Department (Class)
Name	UniversityDepartment {}
Functions	AddDepartment (), ModifyDepartment () etc.

4.3.2 Validation of Department Contract

Validation is performed by executing ontology-derived stakeholder competency questions as smart contract test cases and verifying the results.

Test Case 5

CQ: *What departments are affiliated with the Postgraduate College?*

- Step a: Connect to the UniversityDepartment {} smart contract
- Step b: Execute GetAllDepartment ()

Result: Twelve departments are returned including Computer Science, Civil Engineering, Economics, and others.

This test confirms that departmental entities defined in the ontology are correctly instantiated and retrievable through the blockchain smart contract.

Test Case 6

CQ: Who is the Head of Department of Computer Science at the University of Ibadan?

- Step a: Connect to the UniversityDepartment {} smart contract
- Step b: Execute GetHeadOfDepartment("Computer Science", "University of Ibadan")

Result: Professor Ashley Cross

This validation demonstrates that role-specific stakeholder relationships defined in the ontology are accurately preserved and enforced within the blockchain system.

4.4 Activity Smart Contracts

These contracts formalise postgraduate administrative activities such as admissions, billing, registration, complaints, and certification.

4.4.1 Sample Contract 5: PgAdminActivity

Table 7 summarises the Activity smart contract, responsible for managing organisational activities

Table 7. Postgraduate Activity Contract

Description	Activities in Postgraduate Administration
Ontology Reference	Activity (Class)
Name	PgAdminActivity {}
Functions	AddActivity (), ModifyActivity () etc.

4.4.2 Validation of Activity Contract

Validation is performed by executing activity-related competency questions as smart contract test cases and verifying that the outputs align with the ontology-defined activity structure.

Test Case 7

CQ: *What activities are defined within the context of the Postgraduate Administration Ontology?*

- Step a: Connect to the `PgAdminActivity {}` smart contract
- Step b: Execute `GetAllActivity()`

Result: [“Admission Process”, “Application Process”, “Billing Process”, “Certificate Process”, “Complaint Process”, “Program Deferment Process”, “Program Reactivation Process”, “Student Clearance Process”, “Student Registration Process”]

This result confirms that all ontology-defined administrative activities are correctly stored and retrievable from the blockchain.

Test Case 8

CQ: What is another name for the concept Activity within the ontology?

- Step a: Connect to the `PgAdminAlias {} ` smart contract
- Step b: Execute `GetAlias(‘Activity’) `

Result: Process

This validation confirms that semantic equivalence between ontology concepts and their aliases is preserved during smart contract execution.

4.5 Event Smart Contracts

Event contracts represent atomic occurrences within activities such as announcements, submissions, approvals, and notifications.

4.5.1 Sample Contract 6: PgAdminEvent

Table 8 summarises the Event smart contract, responsible for managing organisational events

Table 8. Postgraduate Event Contract

Description	Events involved in Postgraduate Administration
Ontology Reference	Event (Class)
Name	PgAdminEvent {}
Functions	AddEvent (), AddEventInstance () etc.

4.5.2 Validation of Event Contract

Event contract validation is performed by translating ontology competency questions into executable test cases and verifying the outputs.

Test Case 9

CQ: *What events take place within the Application Process in the Postgraduate Administration Ontology?*

- Step a: Connect to the PgAdminEvent {} smart contract
- Step b: Execute GetAllEvent('Application Process')

Result: 13 events returned

This result confirms that all ontology-defined events associated with the application process are correctly stored and retrievable from the blockchain.

Test Case 10

CQ: *Who is the performer of the event linked to document FM87866?*

- Step a: Connect to the PgAdminEvent {} smart contract
- Step b: Execute GetEventWithDocument(FM87866)
- Step c: Execute GetEventInfo()

Result: "Mr Jaime Chase"

4.6 Deployment of Smart Contracts

All smart contracts were compiled within Remix and deployed to the Ethereum Virtual Machine (EVM) using the Goerli Testnet.

- Deployment transactions consumed gas fees paid using faucet ETH.
- MetaMask served as the blockchain client interface.

4.7 Discussion

The results confirm that ontology concepts can be reliably translated into blockchain-based representations without semantic loss. Each contract type successfully implemented:

- ontology classes (e.g., Document, Department, Activity, Event)
- ontology relations (e.g., previousLocation, performerOf, producesDocument)
- ontology-derived competency questions

The validations and verification demonstrate that the blockchain version of the system reproduces the exact behaviour expected from the ontology model. For every competency question, blockchain execution returned results fully consistent with the ontology's definitions.

The results highlight three major outcomes:

1. Semantic Correctness – blockchain operations preserved ontology meaning.
2. Transparent Provenance – every interaction created an immutable, timestamped trail.
3. Process Accountability – documents, actors, and activities were traceable end-to-end.

Beyond functional correctness, the test results demonstrate that the blockchain-based implementation enforces deterministic execution of administrative activities. Unlike manual or centrally managed systems, each activity and event execution is recorded immutably and can be independently verified. This ensures that once an administrative process is initiated, its progression cannot be altered without detection. The successful execution of competency-question-based tests confirms that the ontology-derived rules are faithfully preserved in the smart contracts, thereby strengthening transparency and administrative accountability.

From an administrative perspective, the blockchain system improves process monitoring by enabling real-time visibility of document movement, responsible actors, and event sequences. This is particularly important in multi-stage administrative workflows where delays, misrouting, or undocumented interventions are common. By encoding administrative logic as smart contracts, the system reduces reliance on discretionary human intervention, thereby minimising ambiguity and operational inconsistencies.

When compared conceptually with traditional administrative systems, the blockchain-based approach exhibits notable advantages. Manual administrative systems are prone to document loss, lack of traceability, and limited transparency. Centralised digital systems improve efficiency but still rely on trusted intermediaries and remain vulnerable to unilateral modifications. In contrast, the proposed blockchain implementation ensures decentralised verification, tamper resistance, and verifiable provenance of administrative actions, making it particularly suitable for environments requiring high integrity and accountability.

This work demonstrates that ontology-driven administrative systems can be operationalized through smart contracts, enabling fair, auditable, and tamper-proof governance.

5. CONCLUSION AND FUTURE WORK

This paper presented an ontology-driven blockchain framework designed to enhance transparency, accountability, and integrity within organisational administration. By translating ontology classes, properties, and competency questions into smart contracts deployed on the Ethereum Virtual Machine, the system demonstrates how institutional processes can be formalised, automated, and validated using decentralised technologies. The developed administrative monitoring system successfully models documents, stakeholders, activities, and events, while ensuring that every administrative interaction is recorded as an immutable and auditable trace on the blockchain.

The results confirm that the blockchain implementation preserved the semantic fidelity of the ontology: each competency question derived from the original administrative model was accurately answered through smart contract execution. This demonstrates the viability of using blockchain as a provenance infrastructure for administrative processes, especially in environments where opacity, manual bottlenecks, and unchecked discretion undermine fairness. This approach provides a foundation for institutions to achieve trusted governance by combining the conceptual clarity of ontologies with the tamper-proof guarantees of blockchain.

Although this evaluation focuses on postgraduate administration, the same validation approach can be applied in other administrative domains. Once an ontology and corresponding competency questions are available, these can be translated into smart contracts and test cases in a similar way. This makes the proposed design a reusable blueprint for other institutions that require transparent, auditable, and formally verifiable administrative processes.

Despite these promising results, several opportunities exist for significant advancement. First, while the current implementation focuses on functional correctness, further research is needed to evaluate performance under varying loads, transaction throughput, and gas-cost optimisation. Second, integration with user-facing interfaces and identity management systems (such as decentralised identifiers or role-based access layers) would be necessary for real-world deployment. Third, although the ontology-to-smart-contract mapping was successful, the process remains partially manual; future work should explore automated translation pipelines capable of converting OWL/RDF knowledge models directly into blockchain-compliant artefacts. Finally, a more comprehensive benchmarking framework that uses novel metrics proposed by the author is under development and will be published separately to evaluate system robustness, alignment, and governance efficiency.

In summary, this research demonstrates that blockchain smart contracts, when guided by formal ontologies, can introduce verifiable structure, provenance, and accountability into administrative systems. The framework serves as a basis for future extensions into broader public-sector governance, education management, corporate compliance, and multi-stakeholder coordination. The combination of semantic modelling and decentralised execution represents a promising path toward transparent and equitable institutional processes.

6. REFERENCES

[1] Cascio, W.F. and Montealegre, R. (2016) ‘How technology is changing work and organizations’, *Annual Review of Organizational Psychology and Organizational Behavior*, 3(1), pp. 349–375. doi:10.1146/annurev-orgpsych-041015-062352.

[2] Mgunda, M.I. (2019) ‘The impacts information technology on business’, Journal of International Conference Proceedings, 2(3), pp. 149–156. doi:10.32535/jicp.v2i3.656.

[3] Bremer, J., & Lehnhoff, S. (2017). Decentralized Coalition Formation with Agent-based Combinatorial Heuristics. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, 6, 29. https://doi.org/10.14201/ADCAIJ2017632944

[4] Taherdoost, H. (2023). Smart Contracts in Blockchain Technology: A Critical Review. Information, 14, 117. https://doi.org/10.3390/info14020117

[5] Alshammari, M. (2020). Evolution of Issues in Distributed Systems: A Systematic Review. 2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), 33–37. https://api.semanticscholar.org/CorpusID:228094241

[6] Ramon-Cortes, C., Alvarez, P., Lordan, F., Alvarez, J., Ejarque, J., & Badia, R. M. (2021). A survey on the Distributed Computing stack. Computer Science Review, 42,100422. doi: 10.1016/j.cosrev.2021.100422

[7] Hassan, S., & de Filippi, P. (2021). Decentralized Autonomous Organization. Internet Policy Review, 10. https://doi.org/10.14763/2021.2.1556

[8] Rozas, D., Tenorio-Fornés, A., Díaz-Molina, S., & Hassan, S. (2021). When Ostrom Meets Blockchain: Exploring the Potentials of Blockchain for Commons Governance. SAGE Open, 11(1). https://doi.org/10.1177/21582440211002526

[9] Bartling, S. (2019). Blockchain for Science and Knowledge Creation: Perspektiven zur Digitalisierung im Gesundheitswesen (pp. 159–180). https://doi.org/10.1007/978-3-662-57611-3_10

[10] Leible, S., Schlager, S., Schubotz, M., & Gipp, B. (2019). A Review on Blockchain Technology and Blockchain Projects Fostering Open Science. 2, 1–28. https://doi.org/10.3389/fbloc.2019.00016

[11] Zheng, Z., Xie, S., Dai, H.-N., Chen, X., & Wang, H. (2018). Blockchain challenges and opportunities: A survey. International Journal of Web and Grid Services, 14, 352. https://doi.org/10.1504/IJWGS.2018.095647

[12] Chuen, L.E.D. (2015) Handbook of Digital Currency,.1st edition., Elsevier (Ed). 361 – 381.

[13] Patil, Mr & Jamdar, Mr & Bhosale, Jaykumar & Katkar, Smita. (2023). An Overview of Blockchain Technology.

[14] Faqir-Rhazoui, Y., Arroyo, J., & Hassan, S. (2021). A comparative analysis of the platforms for decentralized autonomous organizations in the Ethereum blockchain. Journal of Internet Services and Applications, 12. https://doi.org/10.1186/s13174-021-00139-6

[15] Buterin, V (2013). Ethereum White Paper. A Next Generation Smart Contract & Decentralized Application Platform. https://ethereum.org/en/whitepaper/

[16] De Filippi, P. & Hassan, S. (2016). Blockchain Technology as a Regulatory Technology: From Code is Law to Law is Code. First Monday. 21. 10.5210/fm.v21i12.7113.

[17] García J.V., Calvo, A., Hassan, S., & Sánchez-Ruiz, A. (2016). Betfunding: A Distributed Bounty-Based Crowdfunding Platform over Ethereum (pp. 403–411). https://doi.org/10.1007/978-3-319-40162-1_44

[18] Peterson, J., Krug, J., Zoltu, M., Williams, A., & Alexander, S. (2018). Augur: A Decentralized Oracle and Prediction Market Platform. https://doi.org/10.13140/2.1.1431.4563

[19] Xu, Q., Song, Z., Goh, R., & Li, Y. (2018). Building an Ethereum and IPFS-Based Decentralized Social Network System. 1–6. https://doi.org/10.1109/PADSW.2018.8645058

[20] Ayorinde, I. T. and Akinkunmi, B. O. (2013). A Dynamic Ontology for Shared Representation of Knowledge of Establishments. Journal of Science Research, Special Edition Vol. 12: 275- 283.

[21] Dos Santos, R. (2017). On the Philosophy of Bitcoin/Blockchain Technology: Is It a Chaotic, Complex System? Metaphilosophy, 48, 620–633. https://doi.org/10.1111/meta.12266

[22] Sadouskaya, K. (2017). Adoption of Blockchain Technology in Supply Chain and Logistics. https://core.ac.uk/download/pdf/84794328.pdf

[23] Zheng, Z., Xie, S., Dai, H.-N., Chen, X., & Wang, H. (2017, June). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. https://doi.org/10.1109/BigDataCongress.2017.85

[24] Kim, H.M. and Laskowski, M. (2021). Towards an Ontology-Driven Blockchain Design for Supply Chain Provenance. Schulich School of Business, York

University, 4700 Keele St.Toronto, Ontario Canada. Intell Sys Acc Fin Mgmt. 2018; 25: 18–27. https://doi.org/10.1002/isaf.1424.

[25] Kim, H.M., Laskowski, M and Nan, N. (2020) A First Step in the Co Evolution of Blockchain and Ontologies: Towards Engineering an Ontology of Governance at the Blockchain Protocol Level. Schulich School of Business, York University, 4700 Keele St., Toronto, Ontario Canada. Energy Engineering (Energy) eJournal.1-10

[26] Héctor, E. and Ugarte, R. (2017) A more pragmatic Web 3.0: Linked Blockchain Data. [“SEMANTIC BLOCKCHAIN” - STARTING POINT] Rheinische Friedrich-Wilhelms-Universität Bonn – Germany. 10.13140/RG.2.2.10304.12807/1.

[27] El-dosuky, M., & Awad ElAdl, G. (2019). GIZAChain: e-Government Interoperability Zone Alignment, based on blockchain technology. https://doi.org/10.7287/peerj.preprints.27477v1

[28] Schulz, K. A., Gstrein, O, J. and Zwitter, A.J. (2020). Exploring the governance and implementation of sustainable development initiatives through blockchain technology. Chair Group Governance & Innovation, Campus Fryslân, University of Groningen, Wirdumerdijk 34, 8911 CE Leeuwarden, the Netherlands

[29] Oyedeji, O. A. and Ayorinde, I. T. (2023). Development of Administrative Processes Ontology: A Case Study of Postgraduate Education Domain. International Journal of Computer Applications (IJCA) Vol. 185. No. 34: 23-30. http://dx.doi.org/10.5120/ijca2023923119

[30] Ayorinde, I. T., Oyedeji, O. A. and Oyedeji, O. I. (2024). Simplistic Graphic User Interface Design for Rendering Blockchain Data (A Case Study of Educational Administrative Domain). Journal of Innovation in Science and Technology. March 2024. Vol. 3. No. 2: 1-13. http://dx.doi.org/10.56397/IST.2024.03.01.

[31] Oluseyi Ayodeji Oyedeji, Ibiyinka Temilola Ayorinde. (2024). Implementing Formalised Provenances to Enhance Administrative Ontologies. In IJSRED-International Journal of Scientific Research and Engineering Development (Vol. 7, Number 1, pp. 550–560). https://doi.org/10.5281/zenodo.10667678