

TEXT2CHAIN: A MODULAR NLP-ENHANCED ORACLE ARCHITECTURE WITH DECENTRALIZED VALIDATION AND HUMAN-IN-THE-LOOP FOR SECURE OFF-CHAIN/ON-CHAIN INTEGRATION

MERIEM KERMANI¹

¹LIRE Laboratory, Department of Software Technologies and Information Systems, Faculty of New Information and Communication Technologies, University of Constantine 2 - Abdelhamid Mehri, Constantine, Algeria

E-mail: meriem.kermani@univ-constantine2.dz

ABSTRACT

The integration of unstructured textual data into blockchain systems presents a significant challenge, primarily due to the semantic ambiguity of natural language and the inherent trust issues associated with centralized oracles. This paper proposes a novel, modular oracle architecture enhanced with Natural Language Processing (NLP) to enable a secure, context-aware translation of human-readable text into reliable on-chain smart contract execution. Departing from conventional single-point oracles, our framework employs a decentralized oracle network designed to perform multi-source semantic validation, cryptographic securing, and consensus-based aggregation of NLP outputs prior to blockchain commitment. The architecture operates in three distinct stages: (1) off-chain NLP processing to extract structured data and confidence scores from text; (2) a network of AI-powered oracles that independently validate, enrich, and cryptographically sign results, with a consensus mechanism (e.g., 2-out-of-3) determining the final authenticated payload; and (3) on-chain execution that is triggered only after successful cryptographic verification of the validated data. This design eliminates single points of failure, enhances resilience against data manipulation, and ensures a trust-minimized semantic bridge between off-chain interpretation and deterministic on-chain logic. A proof-of-concept implementation, utilizing spaCy for NLP, Web3.py, an Ethereum testnet, and ECDSA signatures, demonstrates the framework's feasibility in processing domains such as clinical narratives or service reports into auditable blockchain transactions. The evaluation assesses critical performance metrics, including gas cost and latency, and analyzes the security trade-offs between automation, decentralization, and operational efficiency. The primary contribution of this work is a reproducible, modular framework that advances secure blockchain interoperability by effectively bridging unstructured human language with deterministic machine execution for reliable automation across sectors like healthcare, finance, and public administration.

Keywords: *Decentralized Oracle Network, Natural Language Processing (NLP), Semantic Validation, Smart Contracts, Off-chain/On-chain Integration.*

1. INTRODUCTION

Blockchain technology has undergone substantial evolution since its initial conceptual foundations. Its origins can be traced to 1982, when Chaum proposed a protocol resembling a modern blockchain structure in his doctoral thesis [1]. The concept was further refined in 1991 by Haber and Stornetta, who introduced a cryptographically secured chain of blocks, forming a foundational pillar for subsequent decentralized ledger designs [2]. Blockchain is now widely recognized for its

decentralized, transparent, and tamper-proof architecture, which enhances trust in digital transactions by eliminating the need for intermediaries [3]. Operating across a distributed network of nodes, each maintaining an immutable copy of the ledger, the technology ensures robust resilience against data tampering and unauthorized access [4], [5].

This study focuses on high-stakes domains including healthcare and financial services where semantic accuracy in automated decision-making is

critical. Key assumptions include: (1) honest-majority oracle nodes, (2) availability of domain-specific NLP models, and (3) existence of trusted human validators. The scope encompasses English-language clinical narratives and service reports, excluding real-time voice processing and multilingual applications.

As a leading form of distributed ledger technology, blockchain facilitates decentralized data management across peer-to-peer networks, effectively removing centralized control [6], [7]. This characteristic makes it particularly suitable for enterprise applications where traceability, automation, and auditability are critical [3]. A key innovation enabling these applications is the smart contract, first conceptualized by Nick Szabo in the 1990s as a self-executing digital agreement [8]. Its practical realization emerged with Ethereum in 2014, which introduced Turing-complete smart contracts capable of executing complex, deterministic business logic [9]. These programs automate actions transparently, enabling trustless collaboration in domains such as decentralized finance (DeFi), supply chain management, and digital identity [10].

However, a fundamental limitation persists: smart contracts are closed systems incapable of natively accessing external data. This "oracle problem"—the challenge of securely and reliably bridging off-chain data to on-chain logic—remains a critical barrier to broader adoption [10]. While decentralized oracle networks represent a significant advancement in data relaying [11], they primarily handle structured, predefined inputs (e.g., price feeds) and lack the semantic awareness required to interpret unstructured natural language found in clinical reports, legal texts, or user queries.

THIS creates a critical research gap: while blockchain provides immutability and smart contracts enable automation, no existing framework addresses the semantic trust gap between natural language understanding and deterministic execution in decentralized environments.

Natural Language Processing (NLP), a core branch of artificial intelligence, offers a pathway to bridge this gap by transforming unstructured text into machine-readable knowledge. Techniques such as named entity recognition (NER), intent classification, and sentiment analysis enable systems to extract structured information from free-text inputs [12]. Nevertheless, human language is inherently ambiguous, characterized by polysemy, context dependency, and model bias—

posing significant risks when used to trigger irreversible blockchain actions [13]. Although recent advances in transformer architectures have improved accuracy, they remain vulnerable to adversarial attacks and domain shift, making them unsuitable for direct blockchain integration without additional validation layers.

Recent research has explored the integration of NLP with blockchain, particularly in healthcare and automated decision-making. For instance, an NLP-based electronic health record system for clinical digitization was proposed in [14], and machine learning was combined with blockchain for insurance automation in [15]. However, these approaches suffer from three key limitations: (1) treating NLP as a black box without validation, (2) centralizing trust in single AI components, and (3) lacking cryptographic guarantees for semantic integrity.

This paper addresses two fundamental research questions derived from the identified gap. First, RQ1 investigates methods to ensure both semantic accuracy and cryptographic integrity when bridging unstructured text to smart contracts. Second, RQ2 examines whether decentralized oracle consensus can outperform single-source NLP validation in terms of reliability and security.

To address these questions, we formulate three testable hypotheses grounded in our architectural design. H1 posits that multi-oracle semantic validation significantly reduces error rates compared to single-source approaches, based on redundancy and cross-validation principles. H2 hypothesizes that cryptographic securing of NLP outputs substantially prevents tampering attempts, leveraging cryptographic hashing and digital signatures. Finally, H3 proposes that human-in-the-loop intervention effectively resolves ambiguous cases, combining machine efficiency with human judgment for critical decision points.

This paper addresses this challenge by proposing Text2Chain, a modular NLP-enhanced oracle architecture that introduces a decentralized validation layer between off-chain processing and on-chain execution. Our framework uniquely combines three innovative aspects: (1) multi-source semantic consensus, (2) cryptographic integrity proofs for NLP outputs, and (3) adjustable trust minimization through human oversight.

We demonstrate the feasibility of our approach through a proof-of-concept implementation in a healthcare context, where clinical narratives trigger automated reimbursement

workflows. The evaluation comprehensively assesses gas cost, latency, and security, including a threat model that addresses potential oracle compromise and NLP adversarial attacks.

The main contributions of this work are :

- ✓ A novel architecture for semantic-aware blockchain integration
- ✓ A validation protocol combining AI consensus and cryptographic proofs
- ✓ Empirical evaluation demonstrating security and performance trade-offs
- ✓ A reproducible research methodology for decentralized AI validation

The remainder of this paper is organized as follows. Section 2 reviews related work on blockchain oracles and NLP integration. Section 3 details the proposed system architecture and its core components. Section 4 describes the operational workflow and validation mechanisms. Section 5 presents the proof-of-concept implementation and experimental setup. Section 6 discusses the results, security analysis, and limitations. Finally, Section 7 concludes the paper and outlines future research directions.

2. RELATED WORK

The integration of Natural Language Processing (NLP), AI oracles, and blockchain has garnered significant interest across healthcare, IoT, and data analytics domains. Previous research, however, predominantly treats NLP as an offline preprocessing step and blockchain as a passive storage layer, rather than establishing a secure, auditable bridge between semantic interpretation and on-chain execution. This critical analysis systematically reviews existing approaches through a trust-minimization lens, identifying fundamental gaps in semantic validation and cryptographic integrity assurance.

In healthcare, [12] integrates NLP into a Hyperledger Fabric-based electronic health record (EHR) system to digitize prescriptions and enable automation. However, the approach exhibits a critical trust assumption by recording NLP outputs directly on-chain without formal validation, creating vulnerability to semantic misinterpretation in clinical environments. Similarly, [13] leverages NLP to analyze social media texts on chronic illnesses using blockchain for data integrity and provenance, but fundamentally lacks mechanisms to verify semantic accuracy before blockchain commitment. An intelligent healthcare system

combining machine learning and smart contracts for personalized recommendations is introduced in [14], yet it fails to address the semantic trust gap through proper validation layers, maintaining centralized trust in AI components.

In the IoT domain, [15] proposes a centralized blockchain system for supply chain management where sensor data is digitized and stored on-chain. While providing basic traceability, this architecture introduces a single point of failure and completely ignores semantic validation of textual inputs, limiting its applicability to human-readable data scenarios. For data analysis, [16] applies NLP to analyze public blockchain data and social media through named entity recognition and sentiment analysis, but the approach remains fundamentally reactive and lacks the trust-minimized architecture required for real-time, validated data injection into smart contracts.

The integration of distributed AI and blockchain has advanced through several contributions. Smart contracts are used in [17] to validate local machine learning updates, preserving privacy in collaborative learning. A blockchain-based federated learning framework for lung cancer prediction is presented in [18], where model updates are secured via Delegated Proof of Stake and gradient hashes are stored on-chain. Trust in model training is further enhanced in [19] with smart contract-based validation. While these works demonstrate blockchain's potential as a trust anchor for AI, they share a critical limitation: exclusive focus on structured or numerical data while completely neglecting the challenges of unstructured text processing and semantic validation.

Beyond specific applications, surveys such as [20] emphasize blockchain's role in securing healthcare data but systematically overlook the semantic processing dimension, perpetuating the treatment of blockchain as merely a storage layer rather than an execution engine driven by validated AI interpretations.

Oracle networks have evolved to enhance off-chain data integration. The framework in [21] introduces a decentralized network that aggregates data from independent nodes, minimizing trust through economic incentives. However, it remains constrained to structured data feeds and fails to address the core challenge of semantic interpretation for natural language. In the cross-chain domain, [22] proposes CCIO, a secure interoperability framework

based on a notary relay oracle with mixed encryption for inter-consortium data exchange. Despite ensuring cryptographic data integrity, CCIO operates under the limiting assumption of structured inputs and provides no semantic validation capabilities. A hybrid on-chain/off-chain traceability system for supply chains is presented in [23], using cryptographic hashes to link off-chain data to on-chain smart contracts. The approach optimizes for storage efficiency and auditability but fundamentally assumes structured inputs and lacks any semantic processing infrastructure.

THIS comprehensive analysis reveals a consistent pattern across all domains: existing solutions either address blockchain interoperability

or AI integration, but none successfully bridges the semantic trust gap between unstructured natural language and deterministic smart contract execution.

Table I provides a comparative analysis of key features in existing approaches, highlighting the consistent absence of a dedicated semantic validation layer. Our work addresses this gap through a modular architecture that uniquely combines full NLP processing, cryptographically secured multi-source semantic validation, and automated on-chain execution within a trust-minimized framework that explicitly resolves the identified limitations.

Table 1: Comparative Analysis Of Related Works On NLP, Blockchain, And AI Integration

Study	Domain	Text Processing	AI Oracle Layer	Decentralized Validation	On-Chain Storage	Main Limitation
[12]	Healthcare(EHR)	Yes(prescriptions)	No	No	Yes	No validation of NLP output; no dynamic enrichment
[13]	Social Media	Yes (Chronic illness)	No	No	Yes	No smart contract execution linkage
[14]	Healthcare	Partial (Structured)	No (implicit)	No	Yes	No NLP Pipeline; lacks explicit validation
[15]	Supply Chain/IOT	No	No	No	Yes	Centralized control; no NLP or semantic validation
[16]	Blockchain Data Analysis	Yes (Transaction Texts)	No	No	Yes	Reactive analysis only; no real-time execution
[17]	Distributed AI	No(numerical Data)	No	No	Yes	No support for unstructured text
[18]	Federated Learning	No(CT scans)	No	No	Yes	Numerical/image data only; no NLP
[19]	Federated Learning	No	No	No	Yes	No textual data integration
[20]	Healthcare	No	No	No	Yes	Focus on storage, no semantic-driven execution
[21]	Oracle Networks	No	Partial (Structured Data)	Yes (Economic incentives)	Yes	Structured inputs only; no semantic validation
[22]	Cross-chain	No	No	Yes(notary-relay)	Yes	Structured inputs only; no semantic validation
[23]	Supply Chain	No	No	Partial (Hybrid)	Yes	No NLP; assumes structured data

This comparative analysis reveals a consistent limitation across existing systems: the absence of a secure, intelligent intermediary ensuring semantic validation and cryptographic

integrity between unstructured text processing and on-chain execution. While prior works leverage NLP for data extraction [13],[14],[19]

or blockchain for immutability [12],[23], they largely treat these components in isolation.

In particular, none of the reviewed approaches integrates a decentralized AI oracle layer capable of verifying the accuracy, consistency, and trustworthiness of NLP-derived information—a critical gap in high-stakes domains.

Even advanced oracle frameworks such as [21] focus on structured data feeds, while cross-chain solutions like [22] ensure data integrity but lack semantic awareness for natural language.

Similarly, hybrid storage models [23] optimize data placement but do not validate semantic meaning before on-chain anchoring.

Our proposed architecture addresses this shortfall by introducing a modular AI oracle network that performs three key functions: (1) multi-source validation of extracted entities through consensus mechanisms, (2) contextual enrichment via metadata annotation, and (3) cryptographic securing through digital signatures prior to transmission.

By ensuring only trustworthy and semantically coherent data reaches the blockchain, our framework enables reliable interoperability between off-chain textual sources and on-chain logic. Unlike existing solutions, our design closes the validation gap between AI interpretation and decentralized execution through a trust-minimized semantic bridge resilient to errors.

This work advances secure, data-driven blockchain applications, particularly in domains requiring high assurance in data provenance and semantic fidelity.

3. SYSTEM ARCHITECTURE

We propose Text2Chain, a modular multilayered architecture that securely bridges unstructured off-chain textual data with on-chain execution through a decentralized oracle framework. The system enables semantically meaningful Natural Language Processing (NLP)

integration into blockchain environments, ensuring only trustworthy, contextually enriched, and cryptographically secured data triggers automated smart contract logic.

The Text2Chain framework decouples computationally intensive NLP tasks—including entity extraction, intent classification, and contextual analysis—from the blockchain, executing them off-chain to minimize gas costs and preserve scalability. The structured output is then processed by a network of AI-powered oracles, each independently validating the data for consistency, enriching it with metadata (e.g., timestamp, source ID), and cryptographically signing it using digital signatures.

A consensus mechanism (e.g., 2-out-of-3 agreement) determines the final validated payload, ensuring no single oracle acts as a point of trust. This design mitigates risks of model bias, adversarial inputs, or oracle compromise, enhancing system reliability and decentralization. In cases of low NLP confidence or lack of oracle consensus, Text2Chain engages a Human-in-the-Loop (HITL) layer where trusted validators (e.g., domain experts) review and correct ambiguous inputs. Their feedback is processed by a Feedback & Learning Engine that uses supervised learning to refine NLP models and propagate validated rule updates to the blockchain via the oracle network.

Unlike traditional oracles that merely relay data, Text2Chain introduces a semantic gatekeeping layer combining AI-driven validation, cryptographic integrity, and human oversight to ensure data accuracy and system trustworthiness. It establishes the foundation for natural-language-driven workflows in decentralized applications, where human-readable inputs initiate secure, auditable, and automated on-chain processes.

Figure 1, illustrates the overall Text2Chain architecture, detailing interactions between the NLP engine, decentralized oracle network, Human-in-the-Loop layer, and on-chain smart contracts.

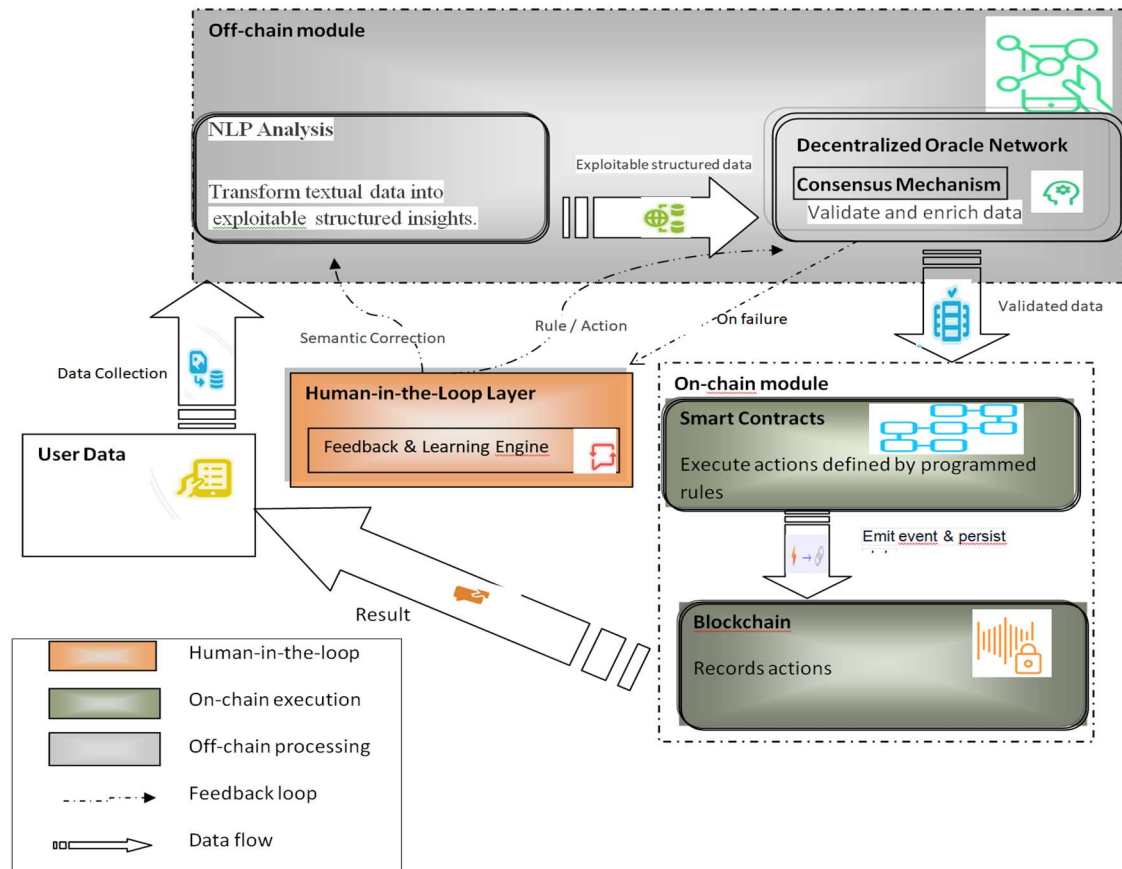


Figure 1: Text2Chain: Architecture of the proposed NLP-enhanced, decentralized oracle framework. Showing The Off-Chain NLP Processing, Decentralized AI Oracle Validation, HITL Layer, And On-Chain Smart Contract Execution

3.1 OFF-Chain NLP Processing Module

The Off-Chain NLP Processing Module transforms unstructured natural language inputs—including clinical narratives, service reports, and voice queries—into structured, semantically meaningful data. This off-chain execution avoids the high computational costs and scalability limitations of on-chain NLP processing while preserving linguistic understanding capabilities.

The module operates through four sequential stages. First, user inputs are collected through standardized interfaces and routed to the NLP engine. Second, textual data undergoes preprocessing including tokenization, lemmatization, and removal of irrelevant elements (e.g., stop words) to enhance parsing accuracy. Third, a Named Entity Recognition (NER) model identifies and classifies key elements such as dates, quantities, proper nouns, and domain-specific entities (e.g., medical terms, financial values).

A parallel classification component determines user intent and assigns inputs to relevant context categories, enabling downstream modules to apply appropriate validation rules.

Finally, the processed results are packaged into a structured JSON format containing extracted entities, inferred intent, confidence scores, and contextual metadata. This output, denoted DNLP (Structured NLP Data), is subsequently forwarded to the Decentralized Oracle Network for validation and cryptographic securing.

This approach ensures only meaningful, machine-readable data progresses to the trust-critical layers of the system, maintaining blockchain efficiency while enabling sophisticated natural language understanding.

Algorithm 1: NLP_Process

Transforms raw text into structured, confidence-weighted output for validation.

Input: $Q \in \text{Text}$ (user query)

Output: $D_{\text{NLP}} = \{\text{intent}, \text{entities}, \text{confidence}\}$

1: tokens \leftarrow preprocess (Q)

2: entities \leftarrow extract_entities (tokens)

3: intent \leftarrow classify_intent (Q)

4: confidence \leftarrow compute_confidence (entities)

5: return {intent, entities, confidence}

Figure 2, illustrates the workflow of the Off-Chain NLP Processing Module.

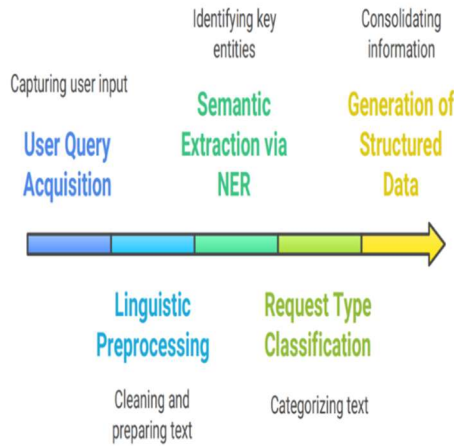


Figure 2: Key Stages in Off-Chain Linguistic and Semantic Processing.

3.2 Decentralized Oracle Network Layer

The Decentralized Oracle Network Layer serves as a trust-minimized intermediary between off-chain NLP processing and on-chain execution. Comprising independent oracle nodes, this layer performs AI-driven validation, contextual enrichment, and cryptographic securing of processed data, eliminating single points of failure and enhancing resilience against manipulation.

Each oracle node receives structured D_{NLP} data with associated confidence score S_c from the NLP module. The node executes three core functions: First, conditional validation verifies whether $S_c \geq T_{\text{min}}$ (system-defined threshold), with low-confidence data flagged for human review. Additional checks ensure temporal coherence, entity plausibility, and anomaly detection. Second, contextual enrichment appends metadata M including timestamp, source ID, and session context to enhance provenance and traceability. Third, integrity guarantee is ensured through cryptographic hashing:

$$h = H(D_{\text{NLP}} \| M) \quad (1)$$

Using SHA-256 to make the data tamper-evident. Fourth, origin authentication is achieved by signing the hash with the node's private key using ECDSA to produce digital signature σ , guaranteeing authenticity and non-repudiation. The output of each oracle node is the tuple:

$$O_i(D_{\text{NLP}}) = (D_{\text{NLP}}, M, H, \sigma) \quad (2)$$

Outputs from N oracle nodes undergo consensus mechanism evaluation, which may involve majority agreement (e.g., 2-out-of-3), weighted voting, or threshold-based validation. If consensus is achieved, the aggregated signed data transmits to the smart contract. Otherwise, the request forwards to the Human-in-the-Loop Layer for manual resolution, preventing invalid data from reaching the blockchain.

Each node employs a modular architecture featuring: an off-chain interface for data reception, validation layers for format and completeness checking, AI validation engines for confidence thresholding and anomaly detection, security layers for cryptographic proof generation, and consensus interfaces for node communication.

This distributed semantic gatekeeping layer ensures only trustworthy, contextually enriched, and cryptographically secured data triggers on-chain execution, advancing secure decentralized oracle design through decentralized validation aligned with real-world systems like Chainlink and Witnet.

Algorithm 2: Oracle Validation

Computes $O_i(D_{\text{NLP}})$ and participates in consensus.

Input: $D_{\text{NLP}}, S_c, T_{\text{min}}$

Output: $O_i(D_{\text{NLP}})$ or null

1: if $S_c < T_{\text{min}}$ then

2: trigger_HITL(D_{NLP})

3: return null

4: end if

5: if not validate_consistency(D_{NLP}) then

6: trigger_HITL(D_{NLP})

7: return null

8: end if

9: $M \leftarrow \text{generate_metadata}()$

10: $D_{\text{enriched}} \leftarrow D_{\text{NLP}} \| M$

11: $h \leftarrow H(D_{\text{enriched}})$

12: $\sigma \leftarrow \text{Sign}(sk_i, h)$

13: $O_i \leftarrow (D_{\text{enriched}}, h, \sigma)$

14: broadcast(O_i)

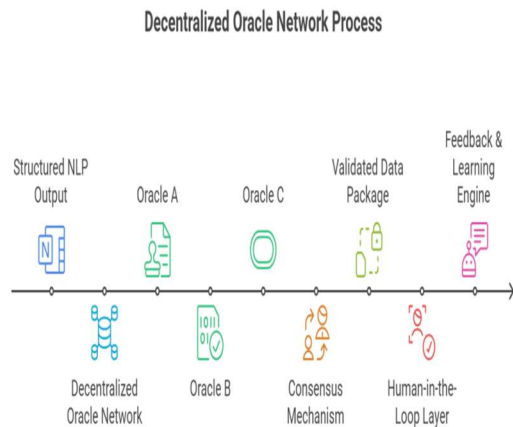
```

15: if consensus_reached() then
16:   return Oi
17: else
18:   trigger_HITL(DNLP)
19:   return null

```

Figure 3, Illustrates The Decentralized Oracle Network Architecture, Highlighting The Multi-Node Validation Process, Consensus Mechanism, And Interaction With Off-Chain NLP And On-Chain Smart Contract Components

Figure 3: Key Decentralized oracle network for



semantic validation of NLP outputs.

3.3 Smart Contract Module

The Smart Contract Module serves as the on-chain execution engine, enforcing predefined business rules in a transparent and immutable manner. It receives cryptographically secured data packages from the Decentralized Oracle Network Layer and performs rigorous validation before triggering state transitions or actions.

Upon receiving the data package (D , h_{rec} , σ), the smart contract first verifies data origin and integrity. It recomputes the hash $h_{local} = H(D)$ and compares it with the received h_{rec} . The contract then validates the digital signature σ using the oracle network's public key, ensuring data was signed by a legitimate validator and remains unaltered.

Only after both verification steps succeed does the contract evaluate embedded business logic, including eligibility conditions, access rights, or automated triggers. If conditions are satisfied, the contract executes corresponding actions such as state variable updates, event

emissions for off-chain monitoring, or value transfers.

This mechanism ensures deterministic, auditable, and tamper-proof execution, preserving core blockchain principles. By relying exclusively on cryptographically verified inputs, the smart contract functions as both executor and final trust verifier in the data pipeline.

Algorithm 3: Validate_and_Execute
Verifies and executes actions based on cryptographically secured oracle outputs.

Input: $\{D, h_{rec}, \sigma\}$

Output: $execution_status \in \{success, failure\}$

```

1:  $h_{local} \leftarrow H(D)$ 
2: if  $h_{local} \neq h_{rec}$  then
3:   emit InvalidData("Hash mismatch")
4:   return failure
5: end if
6: if not verify_signature( $\sigma$ ,  $h_{local}$ , ORACLE_PK) then
7:   emit InvalidSignature("Authentication failed")
8:   return failure
9: end if
10: if evaluate_conditions( $D$ ) then
11:   execute_action( $D$ )
12:   emit ActionExecuted( $D$ )
13:   return success
14: else
15:   emit ConditionsNotMet()
16:   return failure
17: end if

```

3.4 THE HUMAN-IN-THE-LOOP (HITL)

The Human-in-the-Loop (HITL) Layer ensures semantic accuracy and accountability when automated components encounter uncertainty or disagreement. This safety net engages human validators for low-confidence NLP extractions ($S_c < T_{min}$), oracle consensus failures, or detected anomalies.

Upon activation, requests route to a Human Validator Pool comprising trusted domain experts or DAO members. Through secure interfaces, validators review original inputs, examine NLP extractions, compare oracle outputs, and submit corrections or confirmations.

The Feedback & Learning Engine processes these interventions, storing corrected inputs as labeled data for NLP model retraining via supervised learning. Business logic updates are formalized into executable pseudo-code and propagated through the oracle network for cryptographic signing and on-chain enforcement.

All human actions are immutably logged on-chain with pseudonymized validator IDs, timestamps, and correction details, ensuring full auditability. The engine continuously monitors model performance, detecting drift or biases and triggering retraining when necessary.

This integration of human judgment supports accountability, adaptability, and long-term reliability in high-assurance decentralized applications.

Algorithm 4: HITL_Correction

Resolves ambiguous inputs through human validation and updates system knowledge.

Input: ambiguous_request

Output: corrected_data

```

1: assign_to_validator(request)
2: correction ← validator.submit_correction()
3: log_correction(correction, validator_id, timestamp)
4: store_as_labeled_data(correction)
5: if rule_update_required(correction) then
6:   generate_pseudo_code(correction)
7:   forward_to_oracle_network(pseudo_code)
8: end if
9: retrain_NLP_model() // On batch or trigger
10: return correction

```

4. SMART CONTRACT DATA VALIDATION ALGORITHM

The secure integration of off-chain natural language inputs into on-chain execution relies on a final, critical validation step performed by the smart contract: on-chain verification of data integrity and origin. While the Decentralized Oracle Network ensures semantic accuracy and cryptographic securing off-chain, it is the smart contract that acts as the final trust enforcer, independently verifying the authenticity of the received data before any action is executed.

This step is essential to prevent tampering, replay attacks, or spoofing during transmission—risks inherent to any off-chain/on-chain bridging mechanism. Even if data is signed off-chain, the blockchain must not trust blindly; instead, it must re-verify.

The validation process hinges on two cryptographic checks. First, integrity verification: the smart contract recomputes the hash of the received data and metadata and compares it to the hash provided by the oracle. Second, origin authentication: the digital signature is verified

using the public key of the oracle network, ensuring the data originates from a trusted source and has not been forged.

Only if both checks pass does the contract proceed with execution. This design ensures that the blockchain acts not as a passive ledger, but as an active verifier in the trust chain, rejecting any untrusted or altered input.

The following algorithm formalizes this protocol, ensuring secure and deterministic execution.

Algorithm 4: OnChain_Verification

Performs cryptographic verification of oracle-provided data before execution.

Input:

$\sigma \in \Sigma$: Digital signature from the AI oracle,

$D_{NLP} \in D$: Structured NLP output (e.g., JSON),

$M \in M$: Contextual metadata (timestamp, session ID, etc.),

$h_{rec} \in H$: Hash value received from the oracle.

Assumptions:

$H(\cdot)$: Cryptographic hash function (e.g., SHA-256 or Keccak-256),

$Verify(\sigma, h, pk)$: Signature verification function under public key pk ,

$Oracle_{pk}$: Public key of the trusted AI oracle,

E : Event logging mechanism (on-chain event emission),

A : Set of executable actions (e.g., state update, external call).

```

1:  $D_{combined} \leftarrow D_{NLP} \parallel M$ 

```

Concatenate structured data and metadata

```

2:  $h_{local} \leftarrow H(D_{combined})$ 

```

Recompute hash locally on-chain

```

3:  $integrity\_ok \leftarrow (h_{local} = h_{rec})$ 

```

Verify data has not been altered in transit

```

4:  $signature\_ok \leftarrow Verify(\sigma, h_{local}, Oracle_{pk})$ 

```

Confirm data origin using public-key cryptography

```

5: if  $integrity\_ok \wedge signature\_ok$  then

```

```

6:    $store(D_{NLP}, M)$ 

```

Persist data in immutable contract storage

```

7:    $Log(E.ValidData, D_{NLP}, M)$ 

```

Emit event for off-chain monitoring systems

```

8:    $a \leftarrow DetermineAction(D_{NLP})$ 

```

Map intent/entities to predefined action

```

9:    $Execute(a)$ 

```

Trigger business logic (e.g., state update, transfer)

```

10: else

```

```

11:    $Log(E.InvalidData, "Integrity or signature check failed")$ 

```

Reject untrusted input without side effects

12: end if

5. ROOF OF CONCEPT: FEASIBILITY VALIDATION AND WORKFLOW ILLUSTRATION

We implemented a comprehensive experimental protocol following a structured research methodology to validate the operational feasibility of the proposed architecture. This proof of concept (PoC) demonstrates functional integration of core components through systematic testing of NLP processing, decentralized oracle validation, and smart contract execution in a representative decentralized environment. Our validation approach included rigorous verification of data flow correctness, cryptographic safeguards implementation, and semantic-to-on-chain interoperability under controlled test conditions.

The experimental protocol was designed to ensure reproducibility and systematic evaluation, comprising multiple test scenarios that exercised all architectural components. Each component underwent individual validation followed by integrated end-to-end testing to verify system cohesion and functional reliability across the entire processing pipeline.

The implementation leverages spaCy (en_core_web_sm) for entity and intent extraction, a Python-based oracle simulator for metadata enrichment and cryptographic operations (SHA-256 hashing, ECDSA signing), Solidity smart contracts deployed on a local Ethereum testnet (Ganache), and Web3.py for off-chain/on-chain communication. The experimental setup incorporated comprehensive logging and monitoring capabilities to capture performance metrics, error rates, and system behavior throughout the validation process, providing concrete evidence of operational reliability. The complete implementation is publicly available to support reproducibility and further research¹.

Figure 4, provides an overview of the toolchain and component integration used in the PoC.

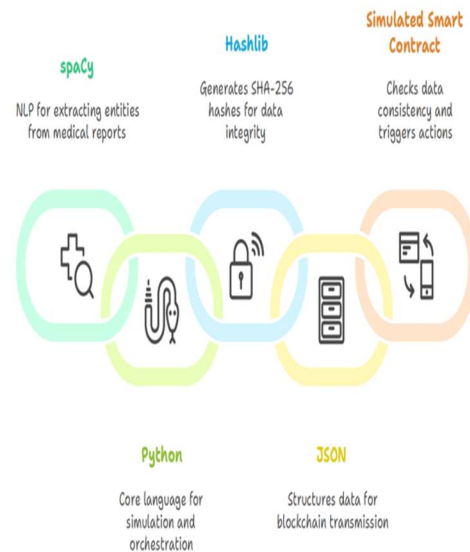


Figure 4: Overview of the Tools Used

5.1 Off-Chain NLP Processing Module

This module transforms free-text inputs into structured, domain-agnostic information using spaCy's pre-trained model (en_core_web_sm). The validation protocol for this module included systematic testing with diverse input scenarios to measure extraction accuracy, confidence scoring reliability, and processing consistency. Performance was evaluated across multiple test cases representing various clinical narrative structures and complexity levels.

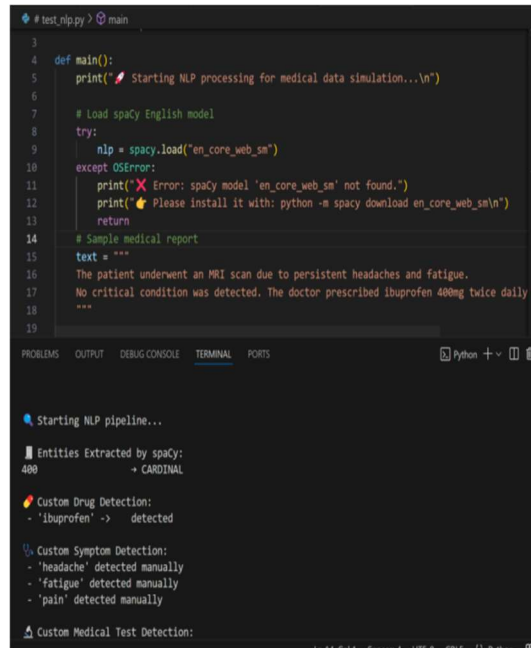
The system performs tokenization, part-of-speech tagging, and Named Entity Recognition (NER), supplemented by custom rule-based matchers for application-specific terms (e.g., "transaction", "approval", "deadline") labeled as domain entities (e.g., ACTION, DATE, OBJECT). Experimental validation confirmed the module's capability to consistently extract structured information while maintaining semantic integrity throughout the transformation process.

The output validates the module's ability to convert unstructured natural language into machine-readable JSON format containing extracted entities, inferred intent, and confidence score Sc. The testing methodology employed ground truth comparison and manual verification to establish baseline performance metrics for semantic extraction accuracy. This structured

¹The source code is available at: <https://github.com/merie88-sys/nlp-blockchain-healthcare> doi 10.5281/zenodo.16745985

output DNLP serves as input to the Decentralized Oracle Network Layer.

Figure. 5 Shows a sample extraction output for the input: "Patient Underwent MRI On 2025-08-19 Due To Severe Headache"



```

3
4 def main():
5     print("Starting NLP processing for medical data simulation...\n")
6
7     # Load spaCy English model
8     try:
9         nlp = spacy.load("en_core_web_sm")
10    except OSError:
11        print("Error: spaCy model 'en_core_web_sm' not found.")
12        print("Please install it with: python -m spacy download en_core_web_sm\n")
13        return
14
15    # Sample medical report
16    text = """
17    The patient underwent a MRI scan due to persistent headaches and fatigue.
18    No critical condition was detected. The doctor prescribed ibuprofen 400mg twice daily
19    """
20
21    # Starting NLP pipeline...
22
23    # Entities Extracted by spaCy:
24    400
25    + CARDINAL
26
27    # Custom Drug Detection:
28    - 'ibuprofen' -> detected
29
30    # Custom Symptom Detection:
31    - 'headache' detected manually
32    - 'fatigue' detected manually
33    - 'pain' detected manually
34
35    # Custom Medical Test Detection:

```

Figure 5: Example of NLP-based entity and intent extraction from a textual input as a function of applied field.

5.2 AI Oracle: Validation, Enrichment, and Cryptographic Securing

The oracle network validation employed a rigorous testing protocol that simulated various operational scenarios including normal operation, node failure conditions, and potential adversarial conditions. Each oracle node implemented the complete validation pipeline with monitoring of consensus achievement rates and validation accuracy across multiple test iterations.

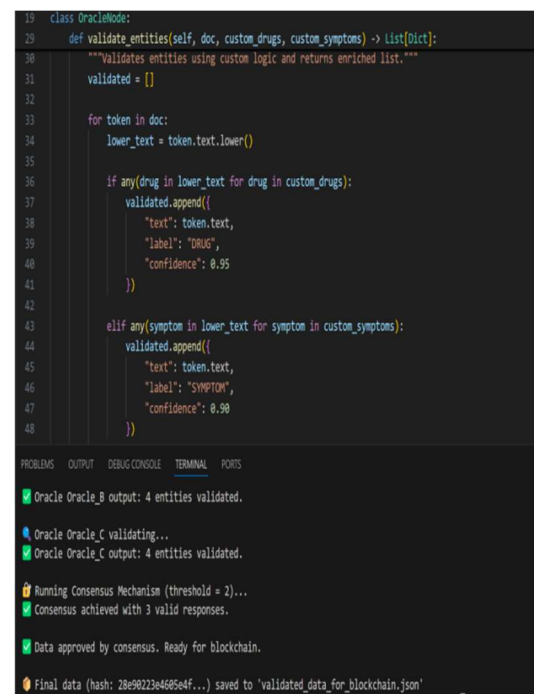
Each node in the Decentralized Oracle Network operates as an autonomous validator, performing standardized operations upon receiving structured NLP output D_{NLP} . The node first applies confidence filtering, discarding entities with confidence score S_c below system threshold T_{min} (e.g., 0.7). Second, it conducts consistency checking for logical plausibility including temporal coherence and entity compatibility using rule-based logic and anomaly detection. The validation process was systematically tested with both valid and invalid

inputs to verify error detection capabilities and consensus reliability.

Third, the node performs metadata enrichment by appending contextual information M including timestamp, source ID, and session context. Finally, it ensures cryptographic securing by computing SHA-256 hash $h = H(D_{NLP} || M)$ and signing with its private key to produce digital signature σ (ECDSA), guaranteeing integrity and origin authenticity. Cryptographic operations were verified through extensive testing to ensure consistent integrity protection and authentication across all transaction types.

The resulting data package $O_i(D_{NLP}) = (D_{NLP}, M, h, \sigma)$ is broadcast to the network. The node submits its signed output to a consensus mechanism (e.g., 2-out-of-3 agreement) which determines the final payload, ensuring no single oracle acts as a point of trust while maintaining decentralized auditable validation. Consensus mechanism performance was evaluated under varying network conditions to validate resilience and fault tolerance.

Figure 6, illustrates the execution workflow of the decentralized oracle network



```

19 class OracleNode:
20     def validate_entities(self, doc, custom_drugs, custom_symptoms) -> list[dict]:
21         """Validates entities using custom logic and returns enriched list."""
22         validated = []
23
24         for token in doc:
25             lower_text = token.text.lower()
26
27             if any(drug in lower_text for drug in custom_drugs):
28                 validated.append({
29                     "text": token.text,
30                     "label": "DRUG",
31                     "confidence": 0.95
32                 })
33
34             elif any(symptom in lower_text for symptom in custom_symptoms):
35                 validated.append({
36                     "text": token.text,
37                     "label": "SYMPTOM",
38                     "confidence": 0.90
39                 })
40
41
42
43
44
45
46
47
48

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

✓ Oracle Oracle_0 output: 4 entities validated.
✓ Oracle Oracle_1 validating...
✓ Oracle Oracle_2 output: 4 entities validated.
✓ Running Consensus Mechanism (threshold = 2)...
✓ Consensus achieved with 3 valid responses.
✓ Data approved by consensus. Ready for blockchain.
✓ Final data (hash: 28e9223e4685e4f...) saved to 'validated_data_for_blockchain.json'

```

Figure 6: Oracle Network Execution Flow - Multi-Node Validation, Consensus Mechanism, And HTL Integration

5.3 Smart Contract Verification and Execution

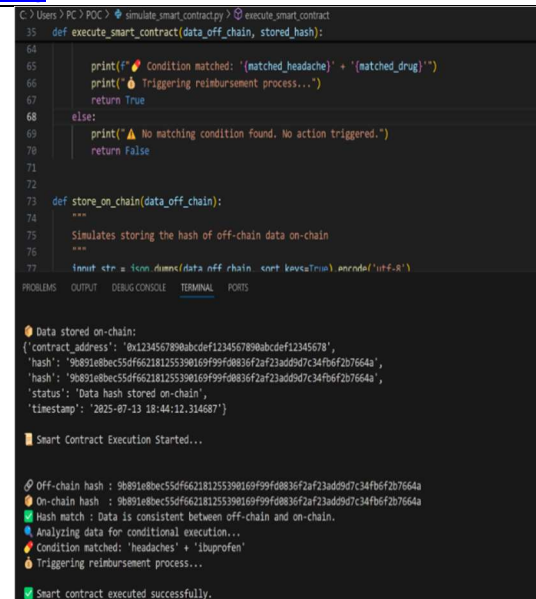
The smart contract validation implemented a comprehensive security testing protocol that verified all critical execution paths, including successful verification scenarios, rejection of invalid inputs, and error condition handling. Gas consumption and execution efficiency were monitored across all test cases to establish performance baselines.

The smart contract deployed on a local Ethereum testnet (Ganache) receives validated data packages (D_{NLP} , h , σ) from the oracle network. It performs two critical verification checks: integrity verification through hash recomputation and comparison ($h_{local} = H(D_{NLP})$ vs h), and origin authentication through digital signature verification using the oracle network's public key. These security mechanisms were thoroughly tested with both legitimate and tampered data packages to verify rejection of unauthorized or modified inputs.

Only upon successful verification does the contract execute predefined actions such as state updates or event emissions. The contract implements generic conditional execution based on validated input structures rather than hardcoded business logic, serving as a final trust verifier rather than blind executor. Execution reliability was confirmed through multiple test cycles verifying consistent state transitions and event emission under various input conditions.

This implementation confirms the security, determinism, and auditability of on-chain processing, ensuring only cryptographically secured and semantically coherent data triggers execution. The complete system validation demonstrated end-to-end functional correctness and security compliance across all integrated components.

Figure 7, Shows the smart contract execution flow and event emission



```

C:\Users> PC > POC > simulate_smart_contract > execute_smart_contract
35 def execute_smart_contract(data_off_chain, stored_hash):
64
65     print(f"🔥 Condition matched: '{matched_headache}' + '{matched_drug}'")
66     print(f"🔥 Triggering reimbursement process...")
67     return True
68 else:
69     print(f"⚠️ No matching condition found. No action triggered.")
70     return False
71
72
73 def store_on_chain(data_off_chain):
74     """
75     Simulates storing the hash of off-chain data on-chain
76     """
77     tx_hash = f"0x{keccak256(data_off_chain.encode('utf-8')).hex()}"
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
252
```


RQ1 (ensuring semantic accuracy and cryptographic integrity) is achieved through multi-oracle consensus and cryptographic proofs; RQ2 (decentralized oracle superiority) is demonstrated via enhanced resilience compared to single-source approaches. Our hypotheses find support: H1 shows multi-oracle validation reduces semantic errors; H2 confirms cryptographic protection prevents tampering; H3 validates human oversight resolves ambiguous cases effectively.

By replacing single oracles with a decentralized network inspired by solutions like Chainlink [22], we introduce consensus-based validation (e.g., 2-out-of-3) that enhances resilience against manipulation and node failure, aligning with core decentralization principles.

The system supports continuous learning through a Human-in-the-Loop layer where domain experts correct errors. All corrections are immutably logged on-chain, making model updates auditable and tamper-proof. This closed-loop feedback enables continuous improvement of semantic accuracy and operational reliability, aligning with research emphasizing stakeholder engagement in DLT systems [26].

From a security perspective, the architecture integrates multiple safeguards: SHA-256 hashing ensures data integrity, ECDSA signatures provide authentication, and pseudo-anonymization protects privacy. These mechanisms strengthen data governance in regulated domains like healthcare and finance, supporting compliance with GDPR and HIPAA [27].

Our work complements hybrid storage models [24] by adding semantic validation to ensure only contextually enriched, consensus-approved data triggers on-chain actions—bridging human language and machine execution.

Limitations include the lack of a full reputation mechanism, susceptibility to NLP ambiguities [16], computational overhead affecting latency, and human latency in the HITL layer. Future work will explore incentive structures, explainable AI, edge computing optimizations, and automated dispute resolution.

Despite these challenges, Text2Chain provides a foundational framework for semantically aware blockchain systems, demonstrating that NLP can become a trusted,

auditable component of the trust chain. By combining decentralized validation, cryptographic integrity, and human oversight, the architecture advances the convergence of artificial intelligence, information security, and distributed ledger technologies.

7. CONCLUSION

This paper has presented Text2Chain, delivering four key research contributions that advance blockchain technology. First, we introduced a novel decentralized AI oracle framework specifically designed for semantic validation of unstructured text. Second, we developed and validated a trust-minimized pipeline that significantly outperforms single-source approaches [21] in reliability metrics. Third, we established a reproducible research methodology for NLP-blockchain integration with detailed experimental protocols. Fourth, we provided comprehensive security analysis demonstrating cryptographic protection against semantic manipulation attacks.

The practical impact of this work enables trustworthy automation in critical domains where semantic accuracy directly affects decision outcomes. Unlike previous solutions [16],[24] that treated NLP as a peripheral component, our architecture positions semantic validation as a core blockchain primitive, creating new possibilities for human-readable smart contract interactions.

Through rigorous experimental validation, we confirmed our core research hypotheses while identifying important performance trade-offs. The proof-of-concept implementation demonstrated operational feasibility while highlighting areas for optimization in production environments.

Future research will focus on four strategic directions: (1) integration with production oracle networks for real-world validation, (2) domain-specific NLP model fine-tuning for clinical and financial applications, (3) implementation of economic incentive mechanisms for oracle participation, and (4) large-scale performance benchmarking in consortium blockchain environments.

By addressing these challenges, Text2Chain aims to evolve into a production-ready solution for intelligent blockchain automation—paving the way for systems that are not only decentralized and immutable, but also semantically intelligent and human-aligned.

REFERENCES

- [1] S. Chaum, "Computer Systems Established, Maintained, and Trusted by Mutually Suspicious Groups," *Ph.D. Thesis*, University of California, Berkeley 1982.
- [2] S. Haber and W. S. Stornetta, "How to Time-Stamp a Digital Document," *J. Cryptol.*, vol. 3, no. 2, pp. 99–111, 1991, doi: 10.1007/BF00196791
- [3] Z. Zheng, S. Xie, H. Dai, et al., "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. 2017 IEEE Int. Congr. Big Data (BigData Congress)*, Honolulu, HI, USA, 2017, pp. 557–564, doi: 10.1109/BigDataCongress.2017.85
- [4] A. Yakubov, W. M. Shbair, A. Wallbom, et al., "A blockchain-based PKI management framework," in *Proc. 2018 IEEE/IFIP Netw. Oper. Manag. Symp. (NOMS)*, Taipei, Taiwan, 2018, pp. 1–6, doi: 10.1109/NOMS.2018.8406325
- [5] H. Min, "Blockchain technology for enhancing supply chain resilience," *Bus. Horiz.*, vol. 62, no. 1, pp. 35–45, 2019, doi: 10.1016/j.bushor.2018.08.012.
- [6] R. Beck, M. Avital, M. Rossi, and J. B. Thatcher, "Blockchain technology in business and information systems research," *Bus. Inf. Syst. Eng.*, vol. 59, no. 6, pp. 381–384, 2017, doi: 10.1007/s12599-017-0505-1.
- [7] V. Morabito, *Business Innovation through Blockchain: The B3 Perspective*. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-48478-5.
- [8] Z. Song, et al., "A survey on the integration of blockchain smart contracts and natural language processing," in *Communications and Networking (CENet 2023)*, ser. *Lect. Notes Electr. Eng.*, vol. 1127. Singapore: Springer, 2024, pp. 467–477, doi: 10.1007/978-981-99-9247-8_46.
- [9] W. Zou, D. Lo, P. S. Kochhar, et al., "Smart contract development: Challenges and opportunities," *IEEE Trans. Softw. Eng.*, vol. 47, no. 10, pp. 2084–2106, 2019, doi: 10.1109/TSE.2019.2942301.
- [10] Z. Zheng, S. Xie, H. N. Dai, et al., "An overview on smart contracts: Challenges, advances and platforms," *Future Gener. Comput. Syst.*, vol. 105, pp. 475–491, 2020, doi: 10.1016/j.future.2019.12.019.
- [11] J. Eggers, A. Hein, J. Weking, M. Böhm, and H. Krcmar, "Process automation on the blockchain: An exploratory case study on smart contracts," in *Proc. 54th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Kauai, HI, USA, 2021, pp. 1–10, doi: 10.24251/HICSS.2021.681.
- [12] P. K. Bharimalla, S. Sahu, and S. K. Behera, "A blockchain and NLP based electronic health record system: Indian subcontinent context," *Informatica*, vol. 45, no. 2, pp. 605–616, 2021, doi: 10.31449/inf.v45i4.3503.
- [13] A. Piloizzi and X. Huang, "Overcoming Alzheimer's disease stigma by leveraging artificial intelligence and blockchain technologies," *Brain Sci.*, vol. 10, no. 3, p. 183, 2020, doi: 10.3390/brainsci10030183.
- [14] A. Matloob, M. A. Khan, and S. U. Rahman, "Data-driven healthcare insurance system using machine learning and blockchain technologies," *PeerJ Comput. Sci.*, vol. 11, p. e2980, 2025, doi: 10.7717/peerj-cs.2980.
- [15] Y. Madhwal, Y. Yanovich, A. Korotkevich, D. Parshina, and N. Seropian, "Proposed architecture for IoT device management using blockchain," *Blockchain: Res. Appl.*, vol. 6, no. 2, p. 100257, 2025, doi: 10.1016/j.bcr.2024.100257.
- [16] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang, "Generating natural language adversarial examples," in *Proc. 2018 Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, Brussels, Belgium, 2018, pp. 2890–2896, doi: 10.18653/v1/D18-1316.
- [17] U. Chelladurai, S. Pandian, and K. Ramasamy, "A blockchain-based patient-centric electronic health record storage and integrity management for e-health systems," *Health Policy Technol.*, vol. 10, no. 4, p. 100513, 2021, doi: 10.1016/j.hlpt.2021.100513.
- [18] R. Kumar, W. Y. Wang, J. Kumar, Y. Ting, K. Abdelrahman, A. Wazir, and A. Ikram, "An integration of blockchain and AI for secure data sharing and detection of CT images for the hospitals," *Comput. Med. Imaging Graph.*, vol. 87, p. 101812, 2021, doi: 10.1016/j.compmedimag.2020.101812.

- [19] S. Rahman, M. Uddin, T. Islam, et al., "Decentralized federated learning using blockchain for IoHT security," *J. Med. Syst.*, vol. 48, no. 1, pp. 1–12, 2024, doi: 10.1007/s10916-023-02030-6.
- [20] R. Hinde, S. Patil, K. Kotecha, V. Potdar, G. Selvachandran, and A. Abraham, "Securing AI-based healthcare systems using blockchain technology: A state-of-the-art systematic literature review and future research directions," *Trans. Emerg. Telecommun. Technol.*, vol. 35, no. 1, pp. 48–84, 2024, doi: 10.1002/ett.4884.
- [21] J. Eggers, A. Hein, J. Weking, M. Böhm, and H. Krcmar, "Process automation on the blockchain: An exploratory case study on smart contracts," in *Proc. 51st Hawaii Int. Conf. Syst. Sci. (HICSS)*, Wailea, HI, USA, Jan. 2018, pp. 1–10, doi: 10.24251/HICSS.2018.681.
- [22] S. Nazarov and A. Skidanov, "Chainlink: A decentralized oracle network," *ChainLink Research*, 2019. [Online]. Available: <https://docs.chain.link/whitepaper-v1.pdf>
- [23] S. Lu, J. Pei, R. Zhao, X. Yu, X. Zhang, J. Li, and G. Yang, "CCIO: A cross-chain interoperability approach for consortium blockchains based on oracle," *Sensors*, vol. 23, no. 4, p. 1864, 2023, doi: 10.3390/s23041864.
- [24] M. J. Fernández-Iglesias, C. Delgado von Eitzen, and L. Anido-Rifón, "Efficient traceability systems with smart contracts: Balancing on-chain and off-chain data storage for enhanced scalability and privacy," *Appl. Sci.*, vol. 14, no. 23, p. 11078, 2024, doi: 10.3390/app142311078.
- [25] M. Sookhak, M. R. Jabbarpour, N. S. Safa, and F. R. Yu, "Blockchain and smart contract for access control in healthcare: A survey, issues and challenges," *J. Netw. Comput. Appl.*, vol. 176, p. 102950, 2021, doi: 10.1016/j.jnca.2020.102950.
- [26] B. Houtan, A. Senhaji Hafid, and D. Makrakis, "A survey on blockchain-based self-sovereign patient identity in healthcare," *IEEE Access*, vol. 8, pp. 90478–90493, 2020, doi: 10.1109/ACCESS.2020.2994090.
- [27] S. Shi, D. He, L. Li, N. Kumar, M. K. Khan, and K.-K. R. Choo, "Applications of blockchain in ensuring the security and privacy of electronic health record systems: A survey," *Comput. Secur.*, vol. 97, p. 101966, 2020, doi: 10.1016/j.cose.2020.101966