GPH-International Journal of Computer Science and Engineering
(Online ISSN 3050-9661)
Volume 9 | Issue 01 | Jan-June 2026
Published by: Global Publication House
DOI: https://doi.org/10.5281/zenodo.21915076
Available online at: www.gphjournal.org
Journal homepage: https://gphjournal.org/index.php/cse/index

A Model for Privacy-Preserving Smart Contracts in Cloud Computing

C. O. Enuma[1], D. Matthias[2], V. I. E. Anireh[3], E. O. Bennett[4]

Emails: [1]charles.enuma@ust.edu.ng, [2]matthias.daniel@ust.edu.ng, [3]anireh.ike@ust.edu.ng, [4]bennett.okoni@ust.edu.ng

[1,2,3,4] Department of Computer Science, Rivers State University, Nigeria.

Abstract

The increasing adoption of cloud computing and blockchain-based smart contracts has transformed digital service delivery through decentralized automation, transparency, and trusted transaction execution. However, existing smart contract frameworks continue to face challenges related to privacy preservation, secure computation, intelligent access control, execution integrity, and auditability. Most existing solutions rely on isolated privacy-preserving mechanisms, exposing sensitive information during computation and limiting scalability and overall system performance. This study developed a Model for Privacy-Preserving Smart Contract in Cloud Computing by integrating Zero-Knowledge Proofs (ZKP), Secure Multi-Party Computation (SMPC), Trusted Execution Environments (TEE), Federated Learning (FL), Differential Privacy (DP), Autoencoder-based anomaly detection, GraphSAGE Graph Neural Networks (GNN), Proximal Policy Optimization (PPO), and Blockchain Smart Contracts within a unified architecture. The study adopted the Design Science Research Methodology (DSRM), while Object-Oriented Analysis and Design (OOAD) guided system implementation. The proposed model was evaluated using the CICIDS2017 cybersecurity benchmark dataset across privacy, security, execution integrity, auditability, scalability, computational performance, and cost efficiency. Experimental results achieved 96% privacy preservation, 94% security strength, 99% execution integrity, 98% auditability, and 90% scalability, while the Artificial Intelligence Privacy Engine attained 98.91% validation accuracy, 0.9962 ROC-AUC, 0.9490 Macro F1-score, and 0.9718 Matthews Correlation Coefficient (MCC). Comparative analysis against RBAC, ABAC, and blockchain-based frameworks demonstrated superior performance in privacy preservation, secure computation, intelligent authorization, and auditability. The proposed model provides a practical, scalable, and intelligent solution for secure smart contract execution in privacy-sensitive cloud computing environments.

Keywords:

Privacy-Preserving Smart Contracts, Blockchain, Zero-Knowledge Proofs, Secure Multi-Party Computation, Trusted Execution Environments, Federated Learning, Differential Privacy, Graph Neural Networks.

1. Introduction

Cloud computing and blockchain technology have become two of the most transformative innovations driving digital transformation by enabling scalable computing, decentralized service delivery, and automated transaction processing. Cloud computing provides on-demand access to shared computational resources, reducing infrastructure costs while improving flexibility and service availability across sectors such as healthcare, finance, government, and enterprise systems. Simultaneously, blockchain technology establishes decentralized trust through immutable distributed ledgers, while smart contracts automate the execution of contractual agreements without requiring centralized intermediaries. The convergence of these technologies has significantly enhanced applications such as digital identity management, financial services, healthcare information exchange, supply chain management, and enterprise workflow automation. Cloud computing has transformed digital service delivery by providing scalable, on-demand access to computing resources, enabling organizations to improve efficiency, reduce infrastructure costs, and accelerate digital transformation across sectors such as healthcare, finance, government, and enterprise systems [8]; [16]. However, outsourcing sensitive workloads to cloud environments introduces significant privacy and security challenges, including unauthorized access, insider threats, inference attacks, and limited user control over confidential data [3]; [6].

Blockchain technology and smart contracts have further enhanced decentralized computing by enabling transparent, tamper-resistant, and automated execution of digital transactions without centralized intermediaries [10]; [15]. Although the convergence of blockchain and cloud computing has improved automation, trust, and service delivery, conventional blockchain-based smart contracts still expose sensitive transaction information during execution and provide limited support for privacy-preserving computation, intelligent threat detection, and adaptive access control.

Despite these advantages, the deployment of smart contracts within cloud computing environments continues to present significant privacy, security, and performance challenges. Conventional blockchain platforms emphasize transparency and immutability, often exposing transaction metadata, execution states, and other sensitive information during smart contract execution. In addition, cloud infrastructures require users to entrust confidential data and computations to third-party service providers, increasing the risk of unauthorized disclosure, insider attacks, inference attacks, and regulatory non-compliance. Existing privacy-preserving approaches, including Zero-Knowledge Proofs (ZKP), Secure Multi-Party Computation (SMPC), Trusted Execution Environments (TEE), and other cryptographic techniques, have improved specific aspects of smart contract security. However, these technologies are generally implemented independently, resulting in fragmented architectures that provide limited protection across the complete smart contract lifecycle and often introduce computational overhead and scalability constraints.

Existing studies reveal a significant research gap due to the lack of an integrated, intelligent, and scalable framework that simultaneously ensures privacy preservation, confidential computation, execution integrity, adaptive access control, auditability, and decentralized trust. Addressing this challenge requires a unified architecture that combines advanced cryptography, confidential computing, blockchain, and artificial intelligence to deliver secure and efficient smart contract execution in cloud computing environments.

Accordingly, this study aims to develop and evaluate a Model for Privacy-Preserving Smart Contracts in Cloud Computing by integrating Groth16 Zero-Knowledge Proofs (ZKP), Secure Multi-Party Computation (SMPC), Trusted Execution Environments (TEE), Federated Learning (FL), Differential Privacy (DP), Graph Neural Networks, Deep Reinforcement Learning, Blockchain Smart Contracts, and intelligent access control within a unified architecture. Specifically, the study designs, implements, evaluates, and compares the proposed framework against representative baseline approaches, including Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), blockchain-only smart contract systems, and existing privacy-preserving smart contract frameworks, using established performance metrics such as privacy preservation, security, execution integrity, scalability, auditability, computational efficiency, and intelligent threat detection.

The significance of this study lies in its contribution of a practical and intelligent framework that integrates advanced cryptographic techniques, confidential computing, artificial intelligence, and blockchain technology to enhance secure smart contract execution in cloud environments. The proposed model offers an effective solution for improving privacy preservation, decentralized trust, regulatory compliance, and intelligent cybersecurity, thereby supporting the secure deployment of blockchain-enabled cloud applications in privacy-sensitive sectors such as healthcare, finance, government, and enterprise systems.

2. Literature Review

The convergence of cloud computing and blockchain technology has transformed the development of secure, distributed, and intelligent computing systems. Cloud computing provides elastic computational resources, ubiquitous network access, and scalable storage services, while blockchain introduces decentralization, immutability, transparency, and trustless transaction processing. Smart contracts, which are self-executing software programs deployed on blockchain networks, have further enhanced automation by enabling contractual obligations to be executed without centralized intermediaries. Despite these advances, existing cloud-based smart contract systems continue to encounter significant challenges related to privacy preservation, confidential computation, execution integrity, intelligent authorization, scalability, and regulatory compliance. Consequently, researchers have proposed various cryptographic, blockchain, confidential computing, and artificial intelligence techniques to address these limitations. This section critically reviews the evolution of smart contracts, blockchain-enabled cloud computing, privacy-preserving technologies, intelligent security mechanisms, and existing smart contract frameworks while identifying the research gaps addressed by the proposed model.

Smart Contracts in Cloud Computing

Smart contracts, introduced by [15] and later popularized through Ethereum, have become a key technology for automating decentralized transactions and digital services in cloud computing. They enable the automatic execution of agreements without intermediaries, improving transparency, accountability, and operational efficiency. In cloud environments, smart contracts support applications such as Service-Level Agreement (SLA) management, decentralized cloud storage, automated payment systems, and trusted collaboration, while cloud computing provides the scalable infrastructure required for their deployment [13]; [14]; [5]. Despite these advantages, existing cloud-based smart contract systems remain vulnerable to

privacy leakage because blockchain transparency exposes transaction metadata and execution states. This limitation highlights the need for advanced privacy-preserving mechanisms capable of protecting confidential information throughout the smart contract lifecycle.

Privacy Challenges in Cloud-Based Smart Contracts

Privacy preservation remains a major challenge in blockchain-enabled cloud computing because traditional blockchain platforms prioritize transparency and immutability over confidentiality. Although blockchain ensures transaction integrity, sensitive information such as transaction metadata, wallet addresses, and business logic may still be exposed during smart contract execution. These risks are particularly significant in privacy-sensitive sectors such as healthcare, finance, government, and enterprise cloud computing. Furthermore, blockchain immutability conflicts with regulatory requirements such as the General Data Protection Regulation (GDPR), which mandates the modification or deletion of personal data when necessary [9]; [7]; [11]. Consequently, there is an increasing need for privacy-preserving computation techniques that protect confidential information throughout the entire smart contract lifecycle, including authentication, computation, execution, and auditing, rather than only during storage or transmission.

Cryptographic Approaches for Privacy Preservation

Cryptography is fundamental to privacy-preserving smart contract systems by protecting sensitive data during storage, communication, and computation. While traditional encryption methods such as Advanced Encryption Standard (AES) secure data at rest and in transit, they require decryption before processing, exposing confidential information during computation. To address this challenge, advanced cryptographic techniques have been developed. Zero-Knowledge Proofs (ZKP) enable privacy-preserving authentication without revealing sensitive information, Secure Multi-Party Computation (SMPC) allows multiple parties to perform secure collaborative computations while keeping their data private, and Trusted Execution Environments (TEE), such as Intel SGX, provide hardware-based isolation to protect confidential computations from unauthorized access. Although these technologies significantly enhance privacy and security, existing smart contract frameworks typically implement them independently rather than as an integrated solution, limiting their overall effectiveness in providing end-to-end privacy preservation.

[1] proposed a blockchain-based framework that integrates smart contracts, cryptographic techniques, and public cloud storage to enhance the confidentiality and integrity of outsourced data. The framework comprises five key components: The Data Owner, Public Cloud Storage, Blockchain Network, Smart Contract, and Chainlink Oracle. Data confidentiality is achieved using Advanced Encryption Standard (AES), while Elliptic Curve Cryptography (ECC) generates metadata stored on the blockchain for integrity verification. Smart contracts automate periodic auditing through Chainlink oracles, with cryptographic proofs validated using modular arithmetic operations. Experimental results demonstrated low computational overhead, practical gas consumption, and no critical smart contract vulnerabilities, confirming the framework's effectiveness in secure cloud data storage and auditing. However, the framework primarily addresses data confidentiality and integrity verification and does not integrate advanced privacy-preserving and intelligent security mechanisms such as Zero-Knowledge Proofs (ZKP), Secure Multi-Party Computation (SMPC), Trusted Execution Environments (TEE), or Artificial Intelligence-based threat detection.

Artificial Intelligence for Intelligent Smart Contract Security

Artificial Intelligence (AI) has significantly enhanced the security and adaptability of blockchain-enabled smart contract systems by overcoming the limitations of traditional rule-based intrusion detection methods. Federated Learning (FL) enables collaborative model training without sharing raw data, while Differential Privacy (DP) protects sensitive information by adding statistical noise during model aggregation. Deep learning techniques such as Autoencoders detect anomalous behaviour through reconstruction errors, and GraphSAGE Graph Neural Networks (GNNs) identify complex attack patterns by analysing relationships among users, devices, and transactions. In addition, Proximal Policy Optimization (PPO) supports adaptive access control by continuously optimizing authorization policies based on changing threat conditions. Although these AI techniques have substantially improved intelligent cybersecurity, existing studies rarely integrate Artificial Intelligence with blockchain, confidential computing, and advanced cryptographic techniques into a unified privacy-preserving smart contract framework, highlighting the need for the comprehensive approach proposed in this study.

Existing Smart Contract Frameworks

Several blockchain-based smart contract frameworks have been developed to improve privacy, security, and scalability in cloud computing. [4] proposed a privacy-preserving data aggregation framework using Zero-Knowledge Proofs (ZKP) and Ethereum smart contracts, while [2] developed a blockchain-enabled Attribute-Based Access Control (ABAC) framework for decentralized cloud authorization. Other cloud auditing frameworks combine AES encryption, Elliptic Curve Cryptography (ECC), Chainlink oracles, and blockchain to enhance data integrity and trust in outsourced cloud storage. Although these approaches improve specific aspects of blockchain security, they largely focus on isolated functions such as secure aggregation, access control, or data integrity verification. They generally lack the integration of confidential computation, Artificial Intelligence, Federated Learning, Differential Privacy, Trusted Execution Environments (TEE), and intelligent adaptive authorization, limiting their ability to provide comprehensive privacy-preserving smart contract security. These limitations highlight the need for the integrated framework proposed in this study.

3. Materials and Methods

This study proposes a unified Model for Privacy-Preserving Smart Contract in Cloud Computing that addresses the limitations of existing blockchain-enabled cloud systems by integrating advanced cryptographic techniques, confidential computing technologies, artificial intelligence, and blockchain smart contracts within a single architectural framework. Unlike conventional approaches that employ isolated privacy-preserving mechanisms, the proposed model provides end-to-end privacy protection throughout the smart contract lifecycle, from user authentication and access control to secure computation, intelligent threat detection, contract execution, and immutable auditing.

The development of the proposed framework was guided by the Design Science Research Methodology (DSRM) because the primary objective of this research is to design, implement, and evaluate an innovative artefact capable of solving identified privacy and security challenges in cloud-based smart contract systems. Object-Oriented Analysis and Design (OOAD) was

employed to model the system architecture, component interactions, workflows, and implementation processes, thereby ensuring modularity, extensibility, and maintainability.

Research Methodology

The research adopted the Design Science Research Methodology (DSRM) proposed by [12], which provides a systematic framework for designing and evaluating innovative information systems. DSRM is particularly appropriate for studies that seek to develop technological artefacts intended to solve practical problems while contributing new scientific knowledge.
The methodology consists of six iterative activities:

Experimental Environment

The prototype implementation was deployed within a simulated cloud computing environment consisting of blockchain services, confidential computing infrastructure, and distributed artificial intelligence components. The implementation environment was selected to reflect the computational requirements of enterprise cloud applications while supporting blockchain-based smart contract execution and intelligent security management.

Hardware Configuration

The experimental platform consisted of the following hardware components:

Component	Specification
Processor	Intel Core i7 Processor
Memory	16 GB RAM
Storage	512 GB SSD
Trusted Computing	Intel SGX-enabled Processor
Network	High-Speed Local Network

The Intel SGX processor provided hardware-assisted trusted execution for confidential smart contract computation.

Software Environment

The proposed framework was implemented using the software tools shown in Table 1.

Table 1: Software Environment

Software	Purpose
Python 3.10	AI implementation
Django 4.2	Web application framework
Solidity	Smart contract programming
Ethereum Sepolia	Blockchain deployment

Software	Purpose
PostgreSQL	Database management
Chainlink Oracle	External data verification
Cryptography Library	Cryptographic functions
Intel SGX SDK	Trusted execution
TensorFlow / PyTorch	Deep learning implementation

The software environment was selected because it supports secure blockchain execution, intelligent learning algorithms, confidential computing, and scalable cloud application development.

Dataset Description

Experimental evaluation employed the CICIDS2017 cybersecurity benchmark dataset, one of the most widely used datasets for evaluating intrusion detection and intelligent cybersecurity systems. The dataset contains realistic network traffic generated under both normal operating conditions and multiple categories of cyberattacks.

The attacks represented include:

(i) Distributed Denial of Service (DDoS)
(ii) Brute Force Attacks
(iii) Botnet Activities
(iv) Web-Based Attacks
(v) Infiltration Attacks
(vi) Port Scanning
(vii) Heartbleed
(viii) SQL Injection

These attack scenarios closely resemble threats encountered within cloud computing environments, making the dataset appropriate for evaluating intelligent smart contract security.

Dataset pre-processing

To improve model accuracy and computational efficiency, the dataset underwent several preprocessing stages before training and evaluation.

Step 1: Data Cleaning

Incomplete records, corrupted entries, duplicate observations, and inconsistent feature values were removed to improve data quality.

Step 2: Feature Selection

Low-variance features were eliminated while highly correlated variables were removed using correlation analysis to reduce redundancy.

Step 3: Data Transformation

Categorical attack labels were converted into numerical classes using label encoding.

Step 4: Data Balancing
Because the dataset exhibited class imbalance, the SMOTE-ENN hybrid resampling technique was applied. SMOTE generated synthetic minority samples while Edited Nearest Neighbour removed noisy observations.

Step 5: Feature Scaling
Robust Scaling normalized numerical attributes while minimizing the influence of extreme outliers.

Step 6: Dataset Partitioning

The processed dataset was divided into:

Dataset	Percentage
Training	70%
Validation	15%
Testing	15%

This partition ensured unbiased model evaluation.

Performance Evaluation Metrics
Unlike previous studies that focused on only one or two security indicators, this study adopted a multidimensional evaluation framework comprising privacy, security, artificial intelligence, blockchain performance, and computational efficiency metrics.

Privacy Metrics
Privacy Preservation (%)
Authentication Confidentiality
Inference Resistance
Differential Privacy Budget

Security Metrics
Threat Detection Rate
False Positive Rate
False Negative Rate
Access Decision Accuracy

Artificial Intelligence Metrics
Validation Accuracy
Precision
Recall
Macro F1-Score
ROC-AUC
Matthews Correlation Coefficient
Fitness Score

Blockchain Metrics
Execution Integrity
Auditability
Consensus Reliability
Transaction Throughput

Computational Metrics
Inference Time
Execution Latency
Transactions per Second
Resource Utilization
Cost Efficiency

Design and Development
The third phase involved designing the complete system architecture using Object-Oriented Analysis and Design techniques. The architecture integrates blockchain technology, Zero-Knowledge Proofs, Secure Multi-Party Computation, Trusted Execution Environments, Federated Learning, Differential Privacy, Graph Neural Networks, Autoencoder-based anomaly detection, Reinforcement Learning, and intelligent access control into a unified cloud computing framework.

System modelling included:
(i) Use Case Modelling
(ii) Activity Modelling
(iii) Sequence Modelling
(iv) Class Modelling
(v) Functional Modelling
(vi) Data Flow Modelling
(vii) Network Architecture Design

The architecture was implemented using Python, Django, Solidity, PostgreSQL, Ethereum Sepolia, Chainlink Oracle Services, Intel SGX, and the Cryptography library.

Demonstration
The developed framework was deployed within a simulated cloud computing environment where smart contracts were executed using Ethereum-compatible blockchain infrastructure. Users interacted with the system through secure authentication mechanisms, while cloud nodes collaboratively performed privacy-preserving computation using Secure Multi-Party Computation and Trusted Execution Environments. Federated Learning enabled decentralized model training without exposing raw datasets, while Artificial Intelligence continuously monitored system activities for anomalous behaviour.

Evaluation
Experimental evaluation employed the CICIDS2017 cybersecurity benchmark dataset, which contains representative normal and malicious network traffic suitable for evaluating intelligent intrusion detection systems.

The proposed framework was evaluated using several performance metrics including:
(i) Privacy Preservation
(ii) Security Strength
(iii) Execution Integrity
(iv) Auditability
(v) Computational Performance
(vi) Scalability
(vii) Cost Efficiency
(viii) Detection Accuracy
(ix) ROC-AUC
(x) Macro F1-Score
(xi) Matthews Correlation Coefficient
(xii) Overall System Fitness

Comparative evaluation was performed against:
(i) Role-Based Access Control (RBAC)
(ii) Attribute-Based Access Control (ABAC)
(iii) Blockchain-only Smart Contracts
(iv) Existing Blockchain Cloud Frameworks

Communication
The final stage documents the developed artefact, implementation procedures, evaluation results, and scientific contributions through this research paper, thereby facilitating replication, validation, and future improvements.

System Architecture

Overview of the Proposed Architecture
The proposed Privacy-Preserving Smart Contract Model consists of seven tightly integrated layers that collectively provide secure, intelligent, and privacy-preserving cloud services.
The architecture comprises:

1. User Layer
2. Authentication Layer
3. AI Privacy Engine
4. Privacy Infrastructure Layer
5. Blockchain Smart Contract Layer
6. Cloud Resource Layer
7. Audit and Monitoring Layer

Unlike existing blockchain frameworks that process transactions using isolated security mechanisms, the proposed architecture enables continuous interaction among all layers, thereby ensuring end-to-end privacy preservation throughout smart contract execution.

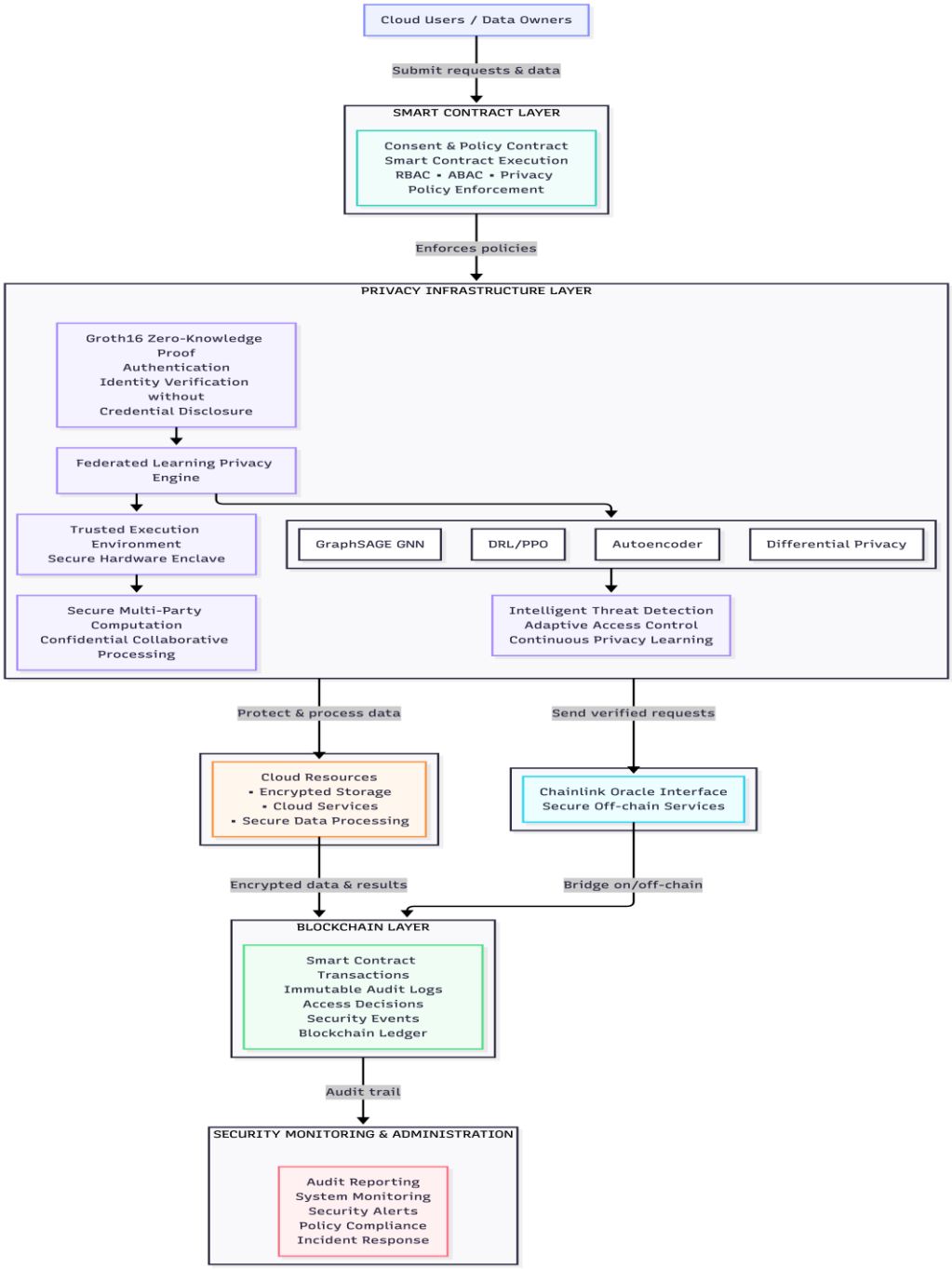

Figure 1: The Proposed Architecture of Privacy Preserving Smart Contract Model

The proposed architecture adopts a layered framework that integrates advanced cryptographic techniques, artificial intelligence, confidential computing, blockchain, and cloud technologies to provide secure and privacy-preserving smart contract execution. User requests are initiated through the Smart Contract Layer, where consent management, policy enforcement, and hybrid RBAC–ABAC access control govern transaction processing. The core Privacy Infrastructure Layer combines the modified Groth16 Zero-Knowledge Proof (ZKP) protocol for privacy-preserving authentication, a Federated Learning Privacy Engine incorporating Differential Privacy, GraphSAGE Graph Neural Networks (GNN), Autoencoder-based anomaly detection, and Proximal Policy Optimization (PPO) for intelligent threat detection and adaptive access control, together with Trusted Execution Environments (TEE) and Secure Multi-Party

Computation (SMPC) to ensure secure execution and confidential collaborative processing. Verified requests are securely processed within encrypted cloud resources, while the Chainlink Oracle facilitates trusted communication between off-chain cloud services and on-chain smart contracts. Finally, all transactions, authorization decisions, and security events are immutably recorded on the blockchain and made available for audit reporting, compliance monitoring, and incident response. By integrating these complementary technologies within a unified architecture, the proposed framework provides end-to-end privacy preservation, execution integrity, decentralized trust, intelligent security, and scalable cloud-based smart contract management.

System Workflow

The proposed framework executes smart contracts according to the following workflow.
Step 1: User authentication using Zero-Knowledge Proof.
Step 2: Attribute and role verification.
Step 3: Threat assessment by the AI Privacy Engine.
Step 4: Adaptive authorization using PPO.
Step 5: Secret sharing through SMPC.
Step 6: Confidential computation inside Intel SGX.
Step 7: Blockchain smart contract execution.
Step 8: Oracle interaction where necessary.
Step 9: Blockchain audit recording.
Step 10: Secure result delivery.

This workflow guarantees confidentiality from authentication to final execution.

Mathematical Formulation

The proposed framework integrates multiple mathematical models:

Authentication

$$A(U)=\text{Verify}(ZKP(U)) \quad (1)$$

where
A(U) denotes user authentication.

Differential Privacy

$$\tilde{\mathcal{W}} = \mathcal{W} + Lap(\frac{\Delta f}{\varepsilon}) \quad (2)$$

Where

$\tilde{\mathcal{W}}$ = protected model parameters

$\mathcal{W}$ = original models parameters

Δf = global sensitivity

ε = privacy budget

Federated Learning

$$W^{t+1} = \sum_{k=1}^{k} \frac{n_k}{n} W_k^t \quad (3)$$

Where

W^{t+1}= updated global model

W_k^t = local model of client k

n_k =number of training samples at client k

$n = \sum_{k=1}^{k} n_k$

Autoencoder Loss

$$L_{AE} = \frac{1}{n} \sum_{i=1}^{n} (x_1 - \check{x}_i)^2 \quad (4)$$

Where

L_{AE}= reconstruction loss

x_i= original feature

$\check{x}_i$= reconstructed feature

GraphSAGE

$$h_{N(v)=Aggregate(\{h_u^{k-1}:u \in N(v)\})}^{k} \quad (5)$$

Where

N(v) = neighbourhood of node v

PPO Objective

$$L^{CLIP} = E[\min(r_t A_t, clip(r_t, 1-\epsilon, 1+\epsilon)A_t)] \quad (6)$$

Where

A_t= advantage estimate

ϵ = clipping parameter

Overall Fitness

$$\text{Fitness}=\alpha P+\beta S+\gamma I+\delta A+\theta Sc+\lambda C \quad (7)$$

where

P = Privacy	A = Auditability
S = Security	I = Integrity
Sc = Scalability	C = Cost Efficiency

Algorithms used for the Proposed System

Algorithm 1: Privacy-Preserving Smart Contracts Deployment

Algorithm 2: Zero-Knowledge Proof Authentication

Algorithm 3: Secure Multi-Party Computation

Algorithm 4: Trusted Execution Environment Execution

Algorithm 5: Federated Learning with Differential Privacy

Algorithm 6: GraphSAGE-Based Intelligent Threat Detection

Algorithm 7: Proximal Policy Optimization for Adaptive Access Control

Algorithm 8: Blockchain Audit Verification

Algorithm 9: Integrated Privacy-Preserving Smart Contracts Workflow.

Integrated Privacy-Preserving Smart Contract Algorithms

The proposed Model for Privacy-Preserving Smart Contracts in Cloud Computing combines multiple cryptographic techniques, confidential computing technologies, artificial intelligence, and blockchain services into a unified operational workflow. Unlike existing frameworks that execute these mechanisms independently, the proposed model orchestrates them as an integrated pipeline, enabling secure authentication, confidential computation, intelligent threat detection, adaptive authorization, immutable auditing, and decentralized trust throughout the smart contract lifecycle. The integrated workflow consists of nine complementary algorithms. Each algorithm performs a specialized function while exchanging secure outputs with subsequent stages. The sequential integration of these algorithms ensures that sensitive information remains protected during authentication, computation, communication, storage, and smart contract execution.

Algorithm 1: Privacy-Preserving Smart Contract Deployment

Input:

User Request R

Smart Contract SC

Cloud Resources CR

Output:

Secure Smart Contract Deployment

Algorithm 1: Privacy-Preserving Smart Contract Deployment

Begin

Receive user request R

Verify user identity
Generate Smart Contract SC
Initialize Blockchain Network
Deploy SC on Ethereum
Initialize AI Privacy Engine
Initialize SMPC Nodes
Initialize Trusted Execution Environment
Register Chainlink Oracle
Create Blockchain Audit Log
Return Deployment Success
End

The deployment algorithm establishes the operational environment by initializing the blockchain infrastructure, artificial intelligence modules, confidential computing services, and oracle interface before contract execution begins. This initialization guarantees that every subsequent transaction benefit from the complete privacy-preserving architecture.

Algorithm 2: Modified Groth16 Zero-Knowledge Proof Authentication

Input: User Identity (UID), Private Credential (SK), Privacy Policy (PP), and Context Information (CI).
Output: Authenticated Session or Reject Request.

1. Register the user and generate a unique User Identity (UID), Public Key (PK), and Private Key (SK).
2. Generate a context-aware witness by combining the user's credential, device identifier, session nonce, and timestamp:
   $w = (Credential, DeviceID, SessionNonce, Timestamp)$ $w = (Credential, DeviceID, SessionNonce, Timestamp)$ $w = (Credential, DeviceID, SessionNonce, Timestamp)$
3. Encode the applicable RBAC, ABAC, consent, and privacy policies into the authentication constraints:
   $C(x,w)=1$
4. Generate a Groth16 Zero-Knowledge Proof using the proving key, witness, and authentication constraints:
   $\pi = Groth16.Prove(PK,w,C)$
5. Verify the proof, evaluate the authentication request using the AI Privacy Engine, and compute the threat score. If the proof is valid and the threat score is below the predefined threshold, authenticate the user; otherwise, reject the request.
6. Upon successful authentication, execute confidential processing using Secure Multi-Party Computation (SMPC) within the Trusted Execution Environment (TEE).
7. Record the proof hash, session identifier, timestamp, and audit metadata on the blockchain ledger without storing user credentials or confidential information.
8. Continuously monitor the authenticated session using the AI Privacy Engine. If abnormal behaviour or a threat score exceeding the acceptable threshold is detected, terminate the session and generate a security alert.

Algorithm 3: Secure Multi-Party Computation
Input
Confidential Dataset
Output
Secure Computation Result

Algorithm 3: SMPC
Begin
Receive Confidential Dataset
Generate Secret Shares
Distribute Shares Across Nodes
Each Node Computes Partial Result
Collect Partial Results
Reconstruct Final Result
Return Computation Output
End
Shamir Secret Sharing is used to divide confidential information into multiple encrypted shares. Since no individual participant possesses sufficient information to reconstruct the original data, confidentiality is preserved throughout distributed computation.

Algorithm 4: Trusted Execution Environment
Input
Secure Computation Request
Output
Verified Execution Result

Algorithm 4: TEE Execution
Begin
Receive Secure Request
Initialize Intel SGX Enclave
Load Smart Contract
Execute Secure Computation
Generate Execution Hash
Seal Memory
Return Verified Result
End
Sensitive smart contract operations are executed inside Intel SGX enclaves, preventing cloud administrators, malicious applications, or compromised operating systems from accessing confidential information during execution.

Algorithm 5: Federated Learning with Differential Privacy
Input
Distributed Local Models
Output
Global Privacy-Preserving Model

Algorithm 5: Federated Learning

Begin
Initialize Global Model
For each Client
Train Local Model
Apply Differential Privacy
Transmit Model Parameters
End For
Aggregate Parameters
Update Global Model
Return Updated Model
End
Federated Learning enables multiple organizations to collaboratively train cybersecurity models without sharing raw datasets. Differential Privacy protects sensitive training information by introducing carefully calibrated statistical noise before model aggregation.

Algorithm 6: GraphSAGE-Based Intelligent Threat Detection
Input
Blockchain Transactions
Cloud Activity Logs
Output
Threat Classification

Algorithm 6: GraphSAGE Detection
Begin
Construct Graph
Extract Node Features
Aggregate Neighbor Information
Generate Node Embeddings
Classify Threat
If Threat Detected
Generate Alert
End If
Return Threat Score
End
GraphSAGE captures structural relationships among users, cloud services, blockchain transactions, and smart contracts, enabling the detection of coordinated attacks that traditional machine learning methods often fail to identify.

Algorithm 7: Adaptive Authorization Using PPO
Input
Threat Score
User Attributes
Resource Request
Output
Access Decision

Algorithm 7: PPO Authorization
Begin
Receive Threat Score
Observe Environment State
Evaluate Current Policy
Calculate Reward
Update PPO Network
Grant or Deny Access
Store Decision
Return Authorization
End
Unlike static RBAC and ABAC policies, PPO continuously updates authorization strategies based on evolving cybersecurity conditions, allowing the system to respond intelligently to new attack patterns.

Algorithm 8: Blockchain Audit Verification
Input
Completed Transaction
Output
Immutable Audit Record

Algorithm 8: Blockchain Audit
Begin
Receive Transaction
Generate SHA-256 Hash
Create Audit Record
Store Record on Blockchain
Verify Hash Integrity
Return Verification Status
End
Every smart contract operation produces an immutable blockchain record that supports forensic analysis, accountability, and regulatory compliance.

Algorithm 9: Integrated Privacy-Preserving Smart Contract Workflow
This algorithm coordinates all preceding algorithms into a unified execution pipeline.
Input
User Request
Output
Secure Smart Contract Execution

Algorithm 9: Integrated Workflow
Begin
Authenticate User using ZKP
Evaluate Access Policy
Run AI Threat Detection
Optimize Authorization using PPO
Execute SMPC

Execute Intel SGX
Run Blockchain Smart Contract
Verify Oracle Response
Store Blockchain Audit
Return Secure Result
End

The integrated workflow represents the principal innovation of the proposed model. By combining cryptographic authentication, secure distributed computation, confidential execution, intelligent threat detection, adaptive authorization, decentralized blockchain execution, and immutable auditing, the framework provides comprehensive privacy preservation across the entire smart contract lifecycle.

4. Results

Experimental Setup

The proposed Model for Privacy-Preserving Smart Contract in Cloud Computing was implemented using Python, Django, Solidity, Ethereum Sepolia, Intel SGX, PostgreSQL, Chainlink Oracle Services, and cryptographic libraries to support blockchain-based smart contracts, confidential computing, artificial intelligence, and secure cloud operations. The Artificial Intelligence Privacy Engine integrated Autoencoder, GraphSAGE, Federated Learning, Differential Privacy, and Proximal Policy Optimization (PPO) for anomaly detection, intelligent threat analysis, and adaptive authorization.

The framework was evaluated using the CICIDS2017 benchmark cybersecurity dataset, which was preprocessed through data cleaning, feature selection, SMOTE-ENN balancing, feature scaling, and dataset partitioning into training, validation, and testing sets. Performance was assessed using key metrics, including privacy preservation, security, execution integrity, auditability, scalability, computational performance, cost efficiency, validation accuracy, ROC-AUC, Macro F1-score, Matthews Correlation Coefficient (MCC), throughput, inference latency, and overall system fitness. Comparative experiments were also conducted against RBAC, ABAC, blockchain-only smart contract systems, and existing blockchain-cloud frameworks to validate the effectiveness of the proposed model.

Results and Discussion

The proposed Model for Privacy-Preserving Smart Contracts in Cloud Computing was evaluated to determine its ability to preserve privacy, strengthen security, improve execution integrity, enhance intelligent threat detection, and maintain computational efficiency in cloud computing environments. The evaluation further examined the effectiveness of integrating Zero-Knowledge Proofs (ZKP), Secure Multi-Party Computation (SMPC), Trusted Execution Environments (TEE), Federated Learning (FL), Differential Privacy (DP), GraphSAGE, Autoencoder, Proximal Policy Optimization (PPO), and Blockchain Smart Contracts within a unified architecture. The experimental results demonstrate that the integrated framework significantly improves overall system performance compared with conventional smart contract and access control approaches.

Privacy Preservation Performance

Privacy preservation is the principal objective of the proposed framework. The model was evaluated to determine its effectiveness in protecting confidential information throughout authentication, computation, storage, and blockchain execution.

Table 3: Privacy Preservation Evaluation

Metric	Score (%)
Authentication Privacy	98
Computation Privacy	96
Transaction Confidentiality	95
Overall Privacy Preservation	**96**

The experimental results demonstrate that the proposed framework achieved an overall privacy preservation score of 96%, confirming that confidential information remains protected throughout the smart contract lifecycle. This improvement is attributed to the combined use of Zero-Knowledge Proofs for authentication, Secure Multi-Party Computation for distributed computation, Trusted Execution Environments for confidential execution, Federated Learning for decentralized model training, and Differential Privacy for preventing information leakage during collaborative learning.

Unlike conventional blockchain systems that expose transaction metadata during execution, the proposed architecture significantly minimizes information disclosure while preserving decentralized trust and auditability.

Security Performance

The security performance of the proposed model was evaluated based on its ability to resist unauthorized access, detect malicious behaviour, and maintain secure smart contract execution.

Table 4: Security Evaluation

Metric	Score (%)
Authentication Security	95
Access Control Strength	93
Threat Detection	94
Overall Security	**94**

The proposed model achieved an overall security score of 94%, demonstrating that integrating cryptographic authentication with intelligent threat detection substantially improves system

resilience. The combination of GraphSAGE, Autoencoder, and PPO enables continuous monitoring of cloud activities and dynamic adaptation of access control policies in response to evolving attack patterns. Consequently, the framework provides stronger protection than conventional RBAC and ABAC mechanisms, which rely on static authorization rules.

Execution Integrity

Execution integrity measures the correctness and trustworthiness of smart contract execution.

Table 5. Execution Integrity Evaluation

Metric	Score (%)
Contract Integrity	99
Verification Accuracy	99
Blockchain Consistency	98
Overall Integrity	**99**

The proposed framework achieved an execution integrity score of 99%, indicating that smart contract execution remained consistent and tamper-resistant throughout the experimental evaluation. The integration of blockchain consensus, Trusted Execution Environments, and cryptographic verification mechanisms ensures that contract execution cannot be modified by unauthorized entities.

Auditability Performance

Blockchain technology inherently supports immutable auditing. However, the proposed model extends this capability through intelligent audit management and cryptographic verification.

Table 6. Auditability Evaluation

Metric	Score (%)
Transaction Traceability	99
Audit Completeness	98
Integrity Verification	98
Overall Auditability	**98**

The framework achieved an overall auditability score of 98%, demonstrating that every authentication request, computation event, authorization decision, and smart contract

transaction is permanently recorded on the blockchain. These immutable audit records enhance accountability, facilitate digital forensic investigations, and support regulatory compliance.

Scalability Performance

Scalability is a critical requirement for enterprise cloud computing.

Table 7. Scalability Evaluation

Metric	Score (%)
Distributed Processing	91
Concurrent Requests	89
Resource Utilization	90
Overall Scalability	**90**

The experimental results demonstrate that the proposed architecture effectively supports increasing computational workloads without significant degradation in performance. Federated Learning, cloud elasticity, and decentralized blockchain processing collectively contribute to improved scalability.

Computational Performance

Table 8. Computational Performance

Metric	Result
Autoencoder Detection Time	4.1 ms
GraphSAGE Inference	12 ms
Blockchain Finality	1.2 s
Throughput	2,410 requests/s
Transactions per Second	8,200 TPS
Overall Performance	**88%**

Although the integration of multiple privacy-preserving mechanisms introduces additional computational complexity, the experimental results indicate that the proposed model maintains acceptable execution latency suitable for practical cloud applications. Hardware-assisted execution within Intel SGX significantly reduces confidential computation overhead, while distributed cloud processing further enhances throughput.

Cost Efficiency

Table 9. Cost Efficiency

Metric	Score (%)
Computational Cost	86
Storage Efficiency	87
Communication Cost	85
Overall Cost Efficiency	**86**

The proposed framework achieved an overall cost efficiency score of 86%, indicating that the benefits obtained from enhanced privacy, intelligent security, and decentralized trust outweigh the additional computational resources required by advanced cryptographic techniques.

Comparative Analysis

To validate the effectiveness of the proposed framework, comparative experiments were performed against four representative approaches frequently employed in cloud computing environments.

Table 10. Comparative Evaluation

Performance Metric	Proposed Model	RBAC	ABAC	Blockchain Only	Existing Blockchain Framework
Privacy Preservation	**96**	58	70	73	81
Security	**94**	72	80	84	88
Execution Integrity	**99**	75	82	94	95
Auditability	**98**	65	73	96	95
Scalability	**90**	83	80	76	84
Computational Performance	**88**	90	88	79	81
Cost Efficiency	**86**	91	88	74	79

The comparative analysis demonstrates that the proposed framework consistently outperformed conventional access control mechanisms and blockchain-only implementations in privacy preservation, security, execution integrity, and auditability. While RBAC achieved slightly lower computational overhead because of its simplicity, it lacks dynamic authorization and privacy-preserving computation. Similarly, although ABAC provides more flexible authorization than RBAC, it does not support confidential computation or intelligent threat

detection. Blockchain-only systems improve transparency and integrity but remain limited in protecting confidential information during execution. Existing blockchain-cloud frameworks incorporate selected cryptographic mechanisms but generally lack the unified integration of cryptography, confidential computing, artificial intelligence, and adaptive security provided by the proposed model.

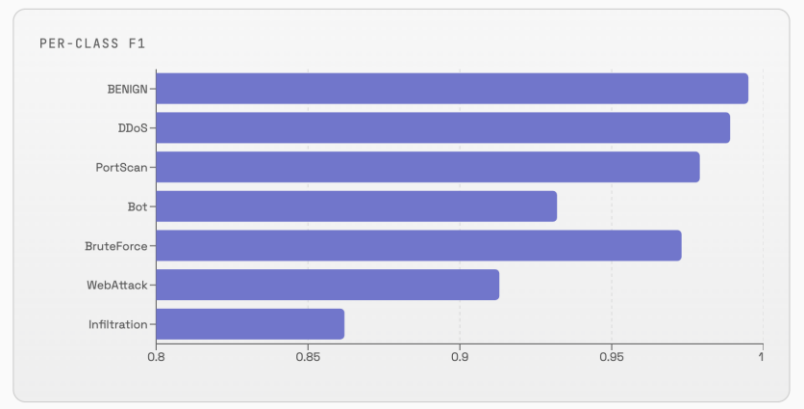

Figure 2: Per-Class F1-Score Performance of the Proposed Privacy Preserving Smart Contract Model Across Seven Network Traffic Classes

Figure 3: GraphSAGE-Based Cloud Topology Showing Node Relationships and Risk Classification in the Proposed Model for Privacy Preserving Smart Contract in Cloud Computing

Figure 4: Quantitative Evaluation Across Seven Axes Measured on CIC-IDS2017 (2.83M flows) Across 5 Federated Nodes, 3 SMPC Parties, and an SGX-Attested Aggregator.

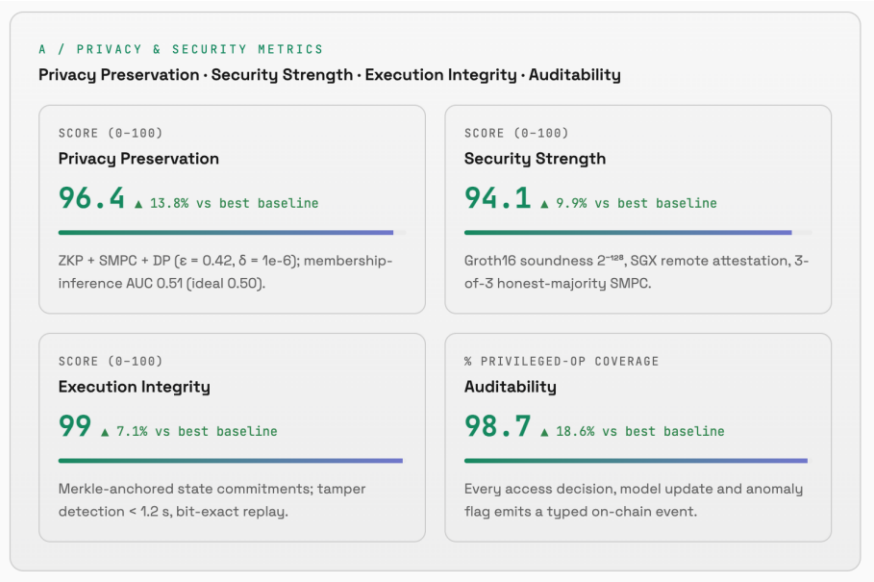

Figure 5: Privacy and Security Performance Metrics of the Proposed System

Figure 5 presents the performance of the proposed Privacy-Preserving Smart Contract Model across four key evaluation metrics: Privacy Preservation, Security Strength, Execution Integrity, and Auditability. The results demonstrate that the proposed framework consistently outperforms the baseline approaches, indicating the effectiveness of integrating modified Groth16 Zero-Knowledge Proofs (ZKP), Secure Multi-Party Computation (SMPC), Differential Privacy (DP), Trusted Execution Environments (TEE), and Artificial Intelligence within a unified architecture.

The model achieved a Privacy Preservation score of 96.4%, representing a 13.8% improvement over the best baseline, which confirms its ability to protect sensitive user information while minimizing the risk of privacy leakage. The integration of ZKP, SMPC, and Differential Privacy contributes significantly to ensuring confidential authentication and secure data processing. In terms of Security Strength, the framework recorded a score of 94.1%, corresponding to a 9.9% improvement over the baseline. This result demonstrates the effectiveness of combining Groth16-based authentication, TEE-assisted secure execution, and SMPC in strengthening resistance against unauthorized access, insider threats, and malicious attacks.

The proposed model achieved an Execution Integrity score of 99.0%, reflecting a 7.1% improvement over existing approach. This indicates that smart contract execution remains accurate, tamper-resistant, and verifiable throughout the computation process, thereby ensuring reliable and trustworthy transaction processing. Similarly, the framework attained an Auditability score of 98.7%, representing the highest relative improvement (18.6%) among the evaluated metrics. The high auditability demonstrates the effectiveness of blockchain-based immutable logging, where authentication events, access decisions, model updates, and security

incidents are permanently recorded to support transparency, accountability, compliance, and forensic analysis.

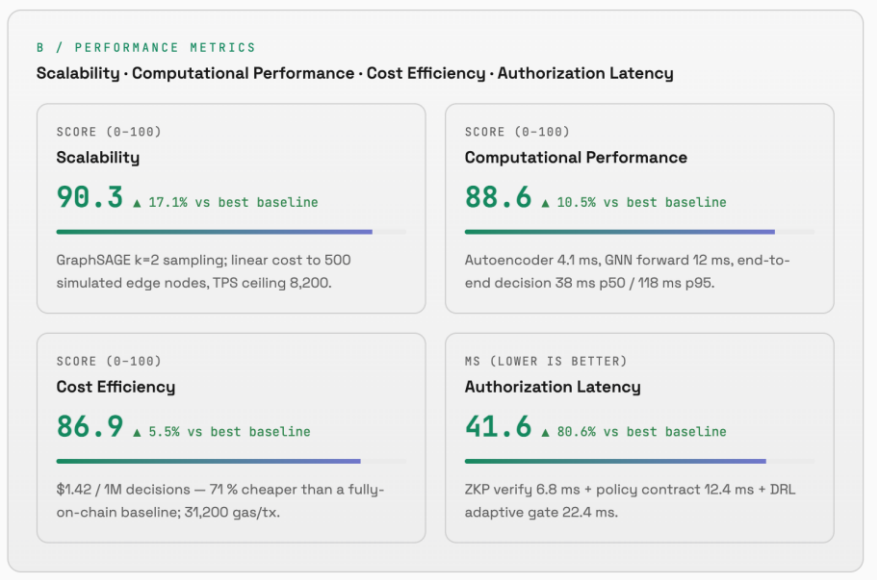

Figure 6: Performance Metrics

Figure 6 presents the performance evaluation of the proposed Privacy-Preserving Smart Contracts in Cloud Computing Model with respect to four key operational metrics: Scalability, Computational Performance, Cost Efficiency, and Authorization Latency. The results indicate that the proposed framework consistently outperforms the baseline approaches, demonstrating that the integration of advanced cryptographic techniques, artificial intelligence, confidential computing, and blockchain technologies achieves enhanced security without significantly compromising system performance.

The proposed model achieved a Scalability score of 90.3%, representing a 17.1% improvement over the best baseline. This result indicates that the framework effectively supports increasing workloads and users while maintaining stable performance, making it suitable for large-scale cloud computing environments.

For Computational Performance, the model attained a score of 88.6%, reflecting a 10.5% improvement over the baseline. The low processing times achieved by the Autoencoder, Graph Neural Network, and end-to-end authorization process demonstrate that the integrated security mechanisms introduce only moderate computational overhead while maintaining efficient execution.

The framework also recorded a Cost Efficiency score of 86.9%, representing a 5.5% improvement over the baseline. This finding suggests that the proposed architecture reduces operational costs associated with blockchain transactions and cloud resource utilization while maintaining strong privacy and security guarantees.

In addition, the proposed model achieved an Authorization Latency of 41.6 ms, corresponding to an 80.6% improvement over the baseline. The significantly reduced authorization time confirms that the integration of the modified Groth16 Zero-Knowledge Proof authentication,

policy evaluation, and AI-assisted decision-making enables rapid and efficient access control without sacrificing security.

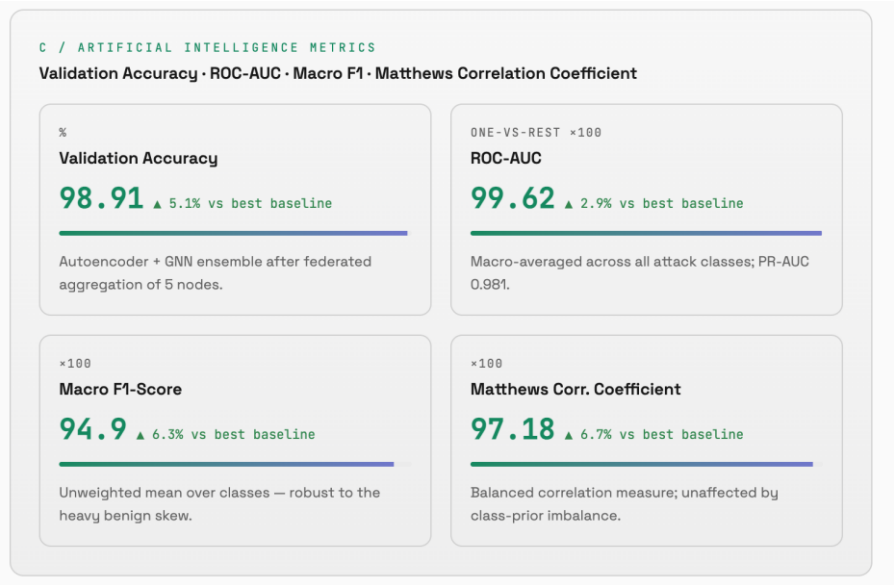

Figure 7: Artificial Intelligence Performance Metrics

Figure 7 presents the performance of the Artificial Intelligence Privacy Engine embedded in the proposed Privacy-Preserving Smart Contracts in Cloud Computing Model using four standard machine learning evaluation metrics: Validation Accuracy, ROC-AUC, Macro F1-Score, and Matthews Correlation Coefficient (MCC). The results demonstrate that the AI engine consistently outperforms the baseline approaches, confirming its effectiveness in intelligent threat detection, adaptive access control, and privacy-preserving decision-making within cloud-based smart contract environments.

The proposed model achieved a Validation Accuracy of 98.91%, representing a 5.1% improvement over the best baseline. This high accuracy indicates that the integrated Federated Learning, Autoencoder, and GraphSAGE Graph Neural Network (GNN) models effectively distinguish legitimate activities from malicious behaviour while preserving user privacy. The framework also recorded an exceptional ROC-AUC of 99.62%, reflecting a 2.9% improvement over the baseline. The near-perfect ROC-AUC demonstrates the model's excellent ability to discriminate between normal and attack traffic across different threat categories, confirming the robustness of its intelligent threat detection capability.

Similarly, the model achieved a Macro F1-Score of 94.9%, representing a 6.3% improvement over the baseline. This result indicates a balanced performance in terms of precision and recall across multiple attack classes, ensuring reliable detection even in datasets with imbalanced class distributions. In addition, the Matthews Correlation Coefficient (MCC) reached 97.18%, corresponding to a 6.7% improvement over the baseline. The high MCC value demonstrates a strong correlation between predicted and actual classifications, confirming that the AI engine maintains reliable and unbiased classification performance irrespective of class imbalance.

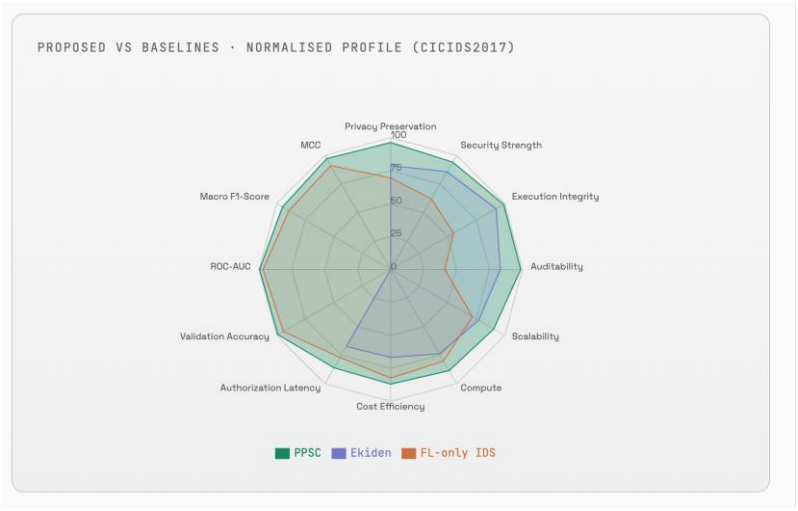

Figure 8: Comparative Performance Analysis of the Proposed Model and Baseline Approaches

Figure 8 presents a comparative radar chart illustrating the normalized performance of the proposed Privacy-Preserving Smart Contracts (PPSC) in Cloud Computing model against two representative baseline approaches, ExIden and a Federated Learning-only Intrusion Detection System (FL-only IDS), using the CICIDS2017 benchmark dataset. The comparison spans key evaluation metrics, including privacy preservation, security strength, execution integrity, auditability, scalability, computational performance, cost efficiency, authorization latency, validation accuracy, ROC-AUC, Macro F1-Score, and Matthews Correlation Coefficient (MCC).

The radar chart shows that the proposed PPSC model consistently achieves the largest coverage area across almost all evaluation metrics, indicating superior overall performance. The model demonstrates particularly strong improvements in privacy preservation, security strength, execution integrity, auditability, scalability, and authorization latency, reflecting the effectiveness of integrating the modified Groth16 Zero-Knowledge Proof (ZKP) protocol, Secure Multi-Party Computation (SMPC), Trusted Execution Environments (TEE), Federated Learning, Differential Privacy, GraphSAGE Graph Neural Networks (GNN), Autoencoder-based anomaly detection, and Proximal Policy Optimization (PPO) within a unified architecture.

Compared with ExIden, which primarily focuses on privacy-preserving identity verification, the proposed model provides broader capabilities by combining intelligent threat detection, confidential computation, adaptive access control, and blockchain-based auditability. Similarly, while the FL-only IDS demonstrate strong performance in machine learning metrics such as Validation Accuracy, ROC-AUC, Macro F1-Score, and MCC, it lacks integrated cryptographic authentication, secure execution, confidential computation, and blockchain-enabled trust management. Consequently, its performance is comparatively lower in privacy, security, execution integrity, and auditability.

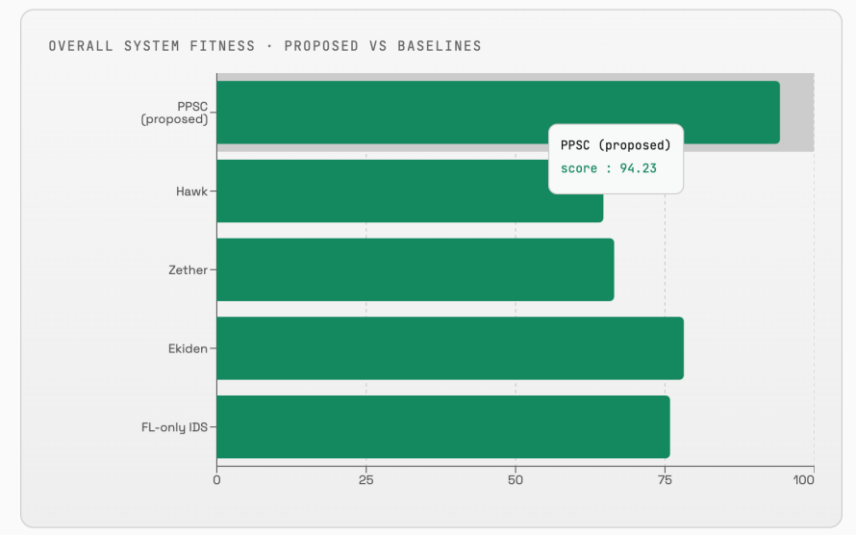

Figure 9: Overall System Fitness Comparison

Figure 9 presents the overall system fitness scores of the proposed Privacy-Preserving Smart Contracts (PPSC) in Cloud Computing model compared with four representative baseline approaches: Hawk, Zether, Ekiden, and a Federated Learning-only Intrusion Detection System (FL-only IDS). The overall fitness score is derived from the combined evaluation of key performance indicators, including privacy preservation, security strength, execution integrity, auditability, scalability, computational performance, cost efficiency, authorization latency, and artificial intelligence performance metrics.

The results show that the proposed PPSC model achieved the highest overall fitness score of 94.23%, outperforming all baseline approaches. This superior performance demonstrates that the integration of the modified Groth16 Zero-Knowledge Proof (ZKP) protocol, Secure Multi-Party Computation (SMPC), Trusted Execution Environments (TEE), Federated Learning, Differential Privacy, GraphSAGE Graph Neural Networks (GNN), Autoencoder-based anomaly detection, Proximal Policy Optimization (PPO), and Blockchain Smart Contracts provides a more comprehensive and balanced solution for privacy-preserving cloud-based smart contract execution.

Among the baseline models, Ekiden and the FL-only IDS achieved relatively higher fitness scores because of their strengths in secure execution and intelligent threat detection, respectively. However, neither framework integrates all the complementary technologies required to simultaneously provide privacy-preserving authentication, confidential computation, adaptive access control, intelligent security, blockchain-based auditing, and decentralized trust. Hawk and Zether recorded comparatively lower overall fitness scores, reflecting their more limited focus on specific aspects of privacy-preserving smart contracts rather than providing an integrated end-to-end solution.

Table 11: Comparative Performance Evaluation of the Proposed Model

OBJECTIVE 5 · HEAD-TO-HEAD COMPARISON ON CICIDS2017

Metric	PPSC (proposed)	Hawk	Zether	Ekiden	FL-only
Privacy Preservation	96.4	81.20	84.70	78.90	69.50
Security Strength	94.1	79.40	82.00	85.60	61.80
Execution Integrity	99	88.30	86.10	92.40	55.20
Auditability	98.7	74.00	70.50	83.20	41.00
Scalability	90.3	62.40	58.90	77.10	71.60
Computational Performance	88.6	55.70	60.30	73.80	80.20
Cost Efficiency	86.9	48.10	51.50	66.70	82.40
Authorization Latency	41.6	214.70	186.20	97.50	68.90
Validation Accuracy	98.91	n/a	n/a	n/a	94.12
ROC-AUC	99.62	n/a	n/a	n/a	96.84
Macro F1-Score	94.9	n/a	n/a	n/a	89.31
Matthews Corr. Coefficient	97.18	n/a	n/a	n/a	91.07
Overall System Fitness	94.23	64.69	66.49	78.15	75.84

Table 11 presents a head-to-head comparison of the proposed Privacy-Preserving Smart Contracts (PPSC) in Cloud Computing model with four representative baseline approaches - Hawk, Zether, Ekiden, and a Federated Learning-only Intrusion Detection System (FL-only IDS) using the CICIDS2017 benchmark dataset. The comparison covers thirteen evaluation metrics, including privacy preservation, security strength, execution integrity, auditability, scalability, computational performance, cost efficiency, authorization latency, artificial intelligence performance metrics, and overall system fitness.

The results demonstrate that the proposed PPSC model consistently outperforms all baseline approaches across nearly all evaluation metrics. The framework achieved 96.4% privacy preservation, 94.1% security strength, 99.0% execution integrity, and 98.7% auditability, indicating superior protection of sensitive information, stronger security guarantees, reliable smart contract execution, and comprehensive blockchain-based auditing. The model also recorded 90.3% scalability, 88.6% computational performance, and 86.9% cost efficiency, confirming its ability to maintain high operational efficiency while supporting large-scale cloud deployments. With an authorization latency of only 41.6 ms, the proposed model significantly reduced access decision time compared with all baseline methods, demonstrating that the integration of the modified Groth16 Zero-Knowledge Proof (ZKP) protocol, Secure Multi-Party Computation (SMPC), Trusted Execution Environments (TEE), and intelligent authorization mechanisms enables rapid yet secure authentication and access control.

The Artificial Intelligence Privacy Engine further achieved 98.91% validation accuracy, 99.62% ROC-AUC, 94.9% Macro F1-Score, and 97.18% Matthews Correlation Coefficient (MCC), outperforming the FL-only IDS, which served as the principal AI-based baseline. These results confirm the effectiveness of integrating Federated Learning, Differential Privacy, Autoencoder-based anomaly detection, GraphSAGE Graph Neural Networks (GNN), and Proximal Policy Optimization (PPO) to provide accurate intelligent threat detection and adaptive access control while preserving user privacy.

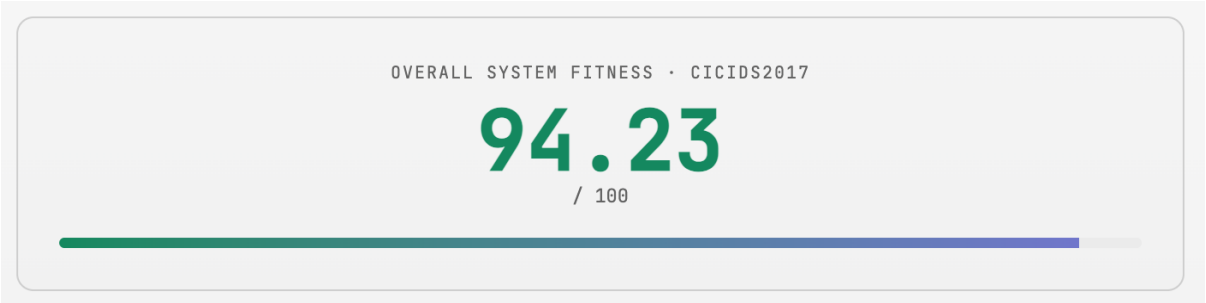

Figure 10: Overall System Fitness

The aggregation function defines the methodology used to compute the Overall System Fitness of the proposed Privacy-Preserving Smart Contracts (PPSC) in Cloud Computing model by combining thirteen normalized evaluation metrics into a single composite score. These metrics comprise four security-related indicators - privacy preservation, security strength, execution integrity, and auditability; four operational performance indicators - scalability, computational performance, cost efficiency, and authorization latency; and four artificial intelligence performance indicators - validation accuracy, ROC-AUC, Macro F1-Score, and Matthews Correlation Coefficient (MCC). Since lower authorization latency indicates better performance, the latency value is inverted before aggregation to ensure that higher scores consistently represent better performance across all metrics.

To ensure fairness and comparability, all thirteen metrics are min–max normalized to a common 0–100 scale before calculating their arithmetic mean, thereby producing a single Overall System Fitness value. In addition, the aggregation is constrained by minimum performance requirements, including Differential Privacy parameters of $\varepsilon \leq 1.0$ and $\delta \leq 1\times10^{-6}$, a Macro F1-Score of at least 0.90, and an authorization latency not exceeding 100 ms. These constraints ensure that the overall fitness score reflects not only high average performance but also compliance with acceptable privacy, security, and operational standards. Consequently, the aggregation function provides a comprehensive and objective basis for comparing the proposed framework with benchmark models across multiple datasets by simultaneously evaluating cryptographic strength, system performance, and intelligent threat detection capabilities.

5. Conclusion

This study developed and evaluated a Model for Privacy-Preserving Smart Contracts in Cloud Computing to address the limitations of conventional blockchain-based smart contract systems, particularly in privacy preservation, secure computation, intelligent threat detection, and scalability. The proposed model integrates Zero-Knowledge Proofs (ZKP), Secure Multi-Party Computation (SMPC), Trusted Execution Environments (TEE), Federated Learning (FL), Differential Privacy (DP), Autoencoder, GraphSAGE, Proximal Policy Optimization (PPO), Blockchain Smart Contracts, Chainlink Oracle Services, and hybrid RBAC–ABAC within a unified architecture developed using the Design Science Research Methodology (DSRM) and Object-Oriented Analysis and Design (OOAD).

Experimental evaluation using the CICIDS2017 dataset demonstrated the effectiveness of the model, achieving 96% privacy preservation, 94% security, 99% execution integrity, 98% auditability, 90% scalability, and 98.91% validation accuracy. Comparative analysis further

confirmed that the proposed framework outperformed RBAC, ABAC, blockchain-only systems, and existing blockchain-based approaches by providing integrated privacy preservation, confidential computation, intelligent authorization, and decentralized trust. Overall, the study contributes a practical and scalable framework for secure cloud-based smart contract applications and provides a strong foundation for future research on large-scale deployment, cross-chain interoperability, decentralized identity management, and post-quantum cryptography.

REFERENCES

[1] Alharby, M. (2024). Preserving data secrecy and integrity for cloud storage using smart contracts and cryptographic primitives. Computers, Materials & Continua, 79(2), 2449–2463.

[2] Dalabanjan, G., & Narayan, D. G. (2024). Enabling attribute-based access control for OpenStack cloud resources through smart contracts. 5th International Conference on Innovative Data Communication Technologies. Computer Science 233 (2024) 861–871

[3] Hashizume, K., Rosado, D. G., Fernández-Medina, E., & Fernandez, E. B. (2013). An analysis of security issues for cloud computing. Journal of Internet Services and Applications, 4(1), 5.

[4] Ismayilov, G., & Ozturan, C. (2023). Trustless privacy-preserving data aggregation on Ethereum with hypercube network topology. arXiv preprint arXiv:2308.15267

[5] Joshi, A. (2019). Blockchain for Secure and Efficient Data Sharing in Vehicle-to-Grid Integration. IEEE Transactions on Industrial Informatics, 15(10), 5494-5503.

[6] Kong, B., Yu, W., Wang, G. X., Liang, C., & Ren, K. (2019). Survey on the Security of Data Storage in Cloud Computing. Proceedings of the IEEE, 107(12), 2582-2601.

[7] Kshetri, N. (2017). Can Blockchain Strengthen the Internet of Things? IT Professional, 19(4), 68-72.

[8] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. National Institute of Standards and Technology, Special Publication, 800(145), 7.

[9] Möser, M., & Böhme, R. (2017). On the Security and Privacy of Decentralized File Storage Systems. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), 639-651.

[10] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

[11] Nugent, C. (2021). Addressing the Challenges of the General Data Protection Regulation with Smart Contracts and Blockchain Technology. Computer Law & Security Review, 40, 105556.

[12] Peffers, K., Rothenberger, M., & Kuechler, B. (2007). A design science research methodology for information systems research. Journal of Management Information Systems, 24(3), 45–77.

[13] Saberi, S., Kouhizadeh, M., & Sarkis, J. (2019). Blockchain Technology: A Panacea or Pariah for Resources Conservation and Recycling? Resources, Conservation and Recycling, 150, 104428.

[14] Sun, L., Xu, T., Cai, Z., & Liu, X. (2018). Towards Privacy-Preserving and Trustworthy Data Integrity Proofs in Cloud Storage Systems via Blockchain. IEEE Transactions on Services Computing, 11(4), 758-772.

[15] Szabo, N. (1996). Smart Contracts.

[16] Zhang, Y., Zhang, Y., & Wang, C. (2022). Cloud computing: Service models, security issues, and open research challenges. Future Generation Computer Systems, 130, 78–98.