

Blockchain-Enabled Audit Infrastructure for Artificial Intelligence-Driven Online Examination

Evoh, O.P.^{1*}, Anireh, V.I.E.², Bennett, E.O.³, Taylor, O.E.⁴

Department of Computer Science, Rivers State University, Port-Harcourt, Rivers State, Nigeria.

*Corresponding Author

Evoh, O.P.

Article History

Received: 06.06.2026

Accepted: 01.07.2026

Published: 20.07.2026

Abstract: This paper presents the Blockchain subsystem of SecureExam Chain, a multi-modal online examination proctoring platform that integrates deep-learning-based fraud detection with a tamper-resistant audit ledger. The blockchain layer is implemented on a private Ethereum network using Solidity smart contracts, deployed via the Hardhat development framework, and interfaces through MetaMask-authenticated browser sessions. Two purpose-built contracts govern the system: ExamSystemsol, which manages role-based identity, exam life cycle and fraud history; and FraudLog.sol, which provides gas-optimised, privacy-preserving, immutable log of AI-generated fraud events. We describe the full contract architecture, state variables, function signatures, access-control modifiers, event schema, cryptographic privacy scheme, and deployment pipeline. The design ensures that no personally identifiable information (PII) is stored on-chain; instead, keccak256 hashes serve as verifiable integrity anchors that any third-party auditor can confirm without exposing raw student data.

Keywords: Blockchain; Ethereum; solidity; smart contracts; online examination integrity; blockchain audit trail; privacy-preserving logging; role-based access control; AI proctoring.

Cite this article:

Evoh, O.P., Anireh, V.I.E., Bennett, E.O., Taylor, O.E., (2026). Blockchain-Enabled Audit Infrastructure for Artificial Intelligence-Driven Online Examination. *ISAR Journal of Science and Technology*, 4(7), 36-49.

I. INTRODUCTION

The integrity of online examinations is a persistent challenge in educational institutions globally. Advances in computer vision and deep learning have enabled automated proctoring systems capable of detecting fraud signals including gaze deviation, head pose anomalies, unauthorized objects, and audio cues, in real time. However, the outputs of such AI systems are inherently disputable: a student may contest a fraud ruling, an institution may require forensic evidence, or a regulatory body may demand an immutable audit trail. Blockchain technology provides a powerful mechanism for ensuring the security and immutability of digital records. A blockchain is a decentralized ledger that records transactions in a manner that prevents modification once the data has been validated and stored. Each

transaction is cryptographically linked to previous records, creating a transparent and tamper-resistant chain of information. In the context of online examinations, blockchain technology can be used to store or reference important events such as question distribution, answer submissions, grading results, and detected cheating incidents. Because the data stored on a blockchain cannot easily be altered, it provides a trustworthy record of examination activities that can be independently verified when necessary (Anjali et al., 2025; Abdelsalam et al., 2024; Sattar et al., 2022; Shareef et al., 2025; Shakila et al., 2024). The increasing demand for secure and trustworthy online examination systems has stimulated considerable research into the application of blockchain technology within educational environments. Existing studies have demonstrated that blockchain possesses significant potential



for improving examination integrity through decentralized record management, immutable data storage, cryptographic verification, and transparent audit trails. Nevertheless, a critical analysis of recent literature reveals that current blockchain-enabled examination systems remain fragmented, with most studies addressing only isolated aspects of examination security rather than providing comprehensive end-to-end evidence management solutions. Blockchain technology is uniquely suited to address this challenge, by recording fraud events as immutable, cryptographically anchored entries on a distributed ledger. The system provides:

- I. Tamper-resistant evidence: once a fraud score is logged, it cannot be altered.
- II. Auditability: any authorized party can verify whether a given event occurred.
- III. Traceability: on-chain role governance ensures that only authorized administrators can access sensitive records.
- IV. Privacy: raw student identifiers and video evidence are never stored on-chain. Only cryptographic hashes are committed.

Smart contracts have become one of the most transformative features of blockchain technology, enabling automated, transparent, and tamper-resistant execution of predefined rules without requiring intermediaries. Originally introduced by Szabo (1997), smart contracts are self-executing computer programs deployed on blockchain networks that automatically enforce contractual conditions once specified requirements are satisfied. Unlike conventional software applications that depend on centralized servers, smart contracts operate on decentralized blockchain infrastructures where every transaction is cryptographically verified and permanently recorded. Their ability to automate trust, eliminate manual intervention, and provide immutable transaction records has made them increasingly attractive for applications requiring accountability, security, and transparency, including healthcare, finance, supply chain management, digital identity, and educational systems. One of the primary applications of smart contracts in educational systems is candidate authentication and examination authorization. Before an examination begins, smart contracts can verify candidate eligibility based on predefined conditions such as course registration, examination schedules, biometric authentication, and identity verification. Once these conditions have been satisfied, the smart contract automatically grants examination access while simultaneously recording the authorization event on the blockchain. Such automated verification significantly reduces administrative overhead while preventing unauthorized examination participation and impersonation attacks.

Within online examination environments, smart contracts provide an efficient mechanism for automating examination monitoring and preserving the integrity of examination records. During an online examination, numerous events including candidate authentication, examination commencement, AI-generated fraud alerts, evidence submission, examination completion, and result publication, must be securely recorded and protected from unauthorized modification. Traditional centralized examination management systems typically rely on database administrators to manage these operations, thereby introducing the risk of insider attacks, unauthorized evidence manipulation, accidental deletion, and disputes regarding examination records. Smart contracts eliminate many of these risks by automatically executing predefined examination rules while recording all critical events on the blockchain in an immutable and verifiable manner. Consequently, every examination activity becomes transparent, traceable, and independently auditable.

This paper presents a Blockchain-Enabled Audit Infrastructure for AI-Driven Online Examination Proctoring. The framework employs blockchain technology and smart contracts to create immutable records of examination events, cryptographic hashes of evidence files, timestamps, fraud confidence scores, and audit logs, ensuring that all recorded evidence remains verifiable throughout its lifecycle. This paper documents a full technical design of the blockchain layer of SecureExam Chain, intended as a reproducible reference for researchers and practitioners building similar systems.

II. RELATED WORK

A comprehensive review by Al-Samarai and Morato (2025) examined blockchain adoption across educational systems within the Gulf Cooperation Council (GCC). Also, Silaghi and Popescu (2025) conducted a systematic review of blockchain initiatives in higher education. Similarly, Alammary et al. (2021) conducted a comprehensive systematic review of blockchain applications in education. These reviews identified online examination systems as one of the most promising future application domains for blockchain technology. Despite this observation, the authors acknowledged that relatively few blockchain implementations had progressed beyond conceptual frameworks to fully operational online examination systems capable of supporting intelligent fraud detection and evidence management. Their findings indicate that blockchain consistently enhances transparency, security, and trust in educational systems but also identify persistent challenges related to scalability, interoperability with Learning Management Systems (LMSs), and regulatory compliance. The reviews concluded that blockchain possesses significant potential for securing educational information systems. Nevertheless, the authors observed that

most existing implementations focus primarily on certificate management and institutional records, while comparatively few studies address real-time online examination monitoring or the secure preservation of fraud evidence generated during assessments. Similarly, Farooq et al. (2021) systematically reviewed blockchain-based online examination assessment models and observed that smart contracts significantly enhance examination transparency by automatically validating examination events, preserving examination logs, and enforcing institutional policies without requiring manual intervention. Their review concluded that blockchain-based automation improves data integrity, reduces opportunities for fraud, and strengthens trust between educational institutions and examination stakeholders. However, they emphasized that most existing implementations remain conceptual and require further experimental validation.

Vinod and Kumar (2025) proposed a blockchain-based framework for enhancing the integrity of online examination results through blockchain-integrated Moodle architecture. Their framework stored examination questions, student responses, and examination results on a blockchain network using cryptographic hashing and a Proof-of-Stake consensus mechanism. Experimental evaluation demonstrated that blockchain effectively prevented unauthorized modification of examination records while maintaining consistency between blockchain data and Moodle's conventional centralized database. Although the framework significantly strengthened examination result integrity, it primarily focused on protecting final examination outcomes rather than preserving dynamic fraud evidence generated during live examination sessions.

Beyond education, blockchain research has consistently emphasized the technology's capability for establishing transparent and accountable digital infrastructures. Tawfeeq and Taha (2025) systematically reviewed blockchain-based accountability mechanisms across multiple domains. Their analysis identified immutability, traceability, auditability, decentralized control, and smart contracts as the principal characteristics enabling trustworthy digital evidence management. The review further highlighted the growing need for hybrid architectures combining on-chain verification with off-chain storage to overcome blockchain scalability limitations while maintaining evidence integrity. These observations are directly applicable to online examination systems where large multimedia evidence files are better maintained off-chain while cryptographic hashes and metadata are securely recorded on the blockchain.

Awaji and Solaiman (2022) designed and evaluated a blockchain-based trusted achievement record system for higher education using the Ethereum blockchain and smart contracts. Their experiment focused on securing academic achievements and certificate verification. System evaluation

using the System Usability Scale (SUS) produced a usability score of 77.1, demonstrating that blockchain-based educational record systems can improve authentication efficiency and user trust compared with conventional centralized approaches. However, the study concentrated on academic credential verification rather than real-time examination monitoring or fraud evidence management.

A notable advancement in blockchain-enabled examination security was reported by Wang et al. (2025), who proposed an integrated online examination cheating detection system combining a lightweight YOLOv12NoAttn object detection model with a Hyperledger Fabric blockchain and IPFS-based distributed storage. Their experiment utilized an online examination cheating dataset containing categories such as unauthorized electronic devices and multiple-person participation. The optimized YOLOv12NoAttn model achieved improvements in recall and mAP@0.5 while simultaneously reducing model parameters by approximately 28% and computational complexity (GFLOPs) by approximately 13%, thereby improving real-time inference efficiency. Their framework successfully preserved detected fraud evidence by recording cryptographic hashes, timestamps, fraud categories, confidence scores, and IPFS addresses within blockchain smart contracts. Experimental evaluation demonstrated excellent object detection performance while maintaining relatively low computational complexity and secure evidence preservation. Although blockchain preserved evidence after fraud detection, the framework did not provide comprehensive management of the entire evidence lifecycle from acquisition through verification, retrieval, audit, and institutional review.

One of the earliest comprehensive blockchain-based evidence management schemes was proposed by Li et al. (2021) through the LEChain framework. The authors developed a consortium blockchain architecture for lawful digital evidence management using Ethereum, Proof-of-Authority (PoA) consensus, ciphertext-policy attribute-based encryption (CP-ABE), and smart contracts. Their framework was designed to secure the entire evidence lifecycle while preserving witness privacy, controlling evidence access, and ensuring transparent judicial supervision. Experimental evaluation demonstrated that LEChain achieved acceptable computational cost, communication overhead, and network latency, confirming the feasibility of blockchain for secure forensic evidence management. The study concluded that blockchain significantly enhances transparency, auditability, and evidence integrity compared with conventional centralized forensic systems. Similarly, Xiao et al. (2024) introduced a blockchain-based digital forensic framework for Industrial Internet of Things (IIoT) environments. Their architecture combined blockchain technology with smart contracts and a

novel batch consensus algorithm to facilitate real-time evidence acquisition, secure storage, and remote evidence verification. Experimental simulations revealed that the proposed consensus mechanism outperformed conventional Delegated Proof-of-Stake (DPoS) algorithms in terms of transaction efficiency while maintaining tamper-proof and non-repudiable evidence storage. The framework successfully demonstrated that blockchain could simultaneously satisfy real-time operational requirements and forensic evidence preservation in distributed computing environments. Building upon this concept, Alqahtany and Syed (2024) developed Forensic Trans Monitor, a blockchain-based digital forensic evidence management platform that integrates smart contracts, cryptographic hashing, and decentralized audit mechanisms. Their framework automatically recorded every forensic event, including evidence acquisition, investigator access, evidence modification requests, and ownership transfers, onto the blockchain ledger. Experimental evaluation demonstrated significant improvements in evidence traceability, investigator accountability, and resistance to evidence tampering compared with conventional centralized forensic systems. Furthermore, the immutable audit trail generated by blockchain enabled investigators to reconstruct the complete history of every evidence item with high confidence, thereby strengthening the credibility of forensic investigations.

A comprehensive review by Onyeashie et al. (2024) examined blockchain-based chain-of-custody systems for digital evidence. After analyzing recent blockchain forensic studies, the authors concluded that blockchain consistently improves evidence integrity, provenance verification, accountability, transparency, and non-repudiation. The review identified blockchain's immutable ledger as one of its greatest strengths because it enables investigators to reconstruct complete evidence histories without relying on centralized authorities. However, the authors also highlighted unresolved challenges involving interoperability, scalability, privacy preservation, and legal standardization. They recommended the development of domain-specific blockchain forensic architectures capable of supporting practical operational environments such as healthcare, finance, and education.

Within educational environments, blockchain-based audit trails have gradually evolved from certificate verification toward comprehensive examination monitoring.

Although recent studies have demonstrated the potential of blockchain for enhancing educational security, existing solutions remain largely confined to certificate verification, examination result storage, or isolated fraud evidence registration. Comparatively fewer studies have investigated comprehensive blockchain-enabled evidence management frameworks capable of securely recording, verifying,

retrieving, and auditing multimodal fraud evidence generated continuously during online examinations. The comparative analysis of these studies reveals several important research gaps.

First, most blockchain-based educational systems focus on credential verification, academic record preservation, or examination result storage rather than continuous monitoring and preservation of fraud evidence generated during online examinations. Consequently, existing systems provide limited support for maintaining a complete forensic record of examination activities.

Second, many existing blockchain-enabled examination systems rely on single-source evidence, particularly visual object detection or identity verification. Such approaches are unable to capture the complex and dynamic behavioural characteristics associated with sophisticated online examination fraud, including suspicious gaze behaviour, abnormal movement patterns, behavioural inconsistencies, or environmental audio anomalies.

Third, although blockchain successfully guarantees immutability after evidence has been recorded, few studies provide a complete blockchain-enabled evidence lifecycle encompassing evidence acquisition, cryptographic hashing, smart contract validation, secure storage, evidence retrieval, integrity verification, institutional auditing, and forensic review. Most existing frameworks record evidence but do not comprehensively manage its subsequent verification and utilization.

Fourth, relatively few blockchain-enabled examination systems integrate Explainable Artificial Intelligence (XAI) with blockchain-based evidence preservation. Consequently, fraud decisions often lack transparency, making it difficult for examination administrators, investigators, or affected candidates to understand the reasoning behind automated fraud classifications.

To address these limitations, this study proposes a Blockchain-Enabled Audit Infrastructure for AI-Driven Online Examination Proctoring within the Secure Exam Chain architecture. Unlike previous studies, the proposed framework integrates blockchain smart contracts with multimodal artificial intelligence to manage the complete lifecycle of examination evidence. Fraud evidence generated through facial verification, gaze tracking, object detection, behavioural analysis, and environmental audio monitoring is cryptographically hashed, securely preserved using blockchain, and linked to distributed multimedia storage through immutable blockchain transactions. The proposed smart contract architecture automatically records candidate authentication, examination commencement, multimodal fraud alerts, fraud confidence scores, cryptographic evidence hashes, timestamps, examination completion, and audit logs. Every transaction generated by the artificial

intelligence engine is permanently recorded on the blockchain, thereby establishing a complete and immutable chain of custody for examination evidence. This integration significantly enhances transparency, accountability, trustworthiness, and forensic reliability while maintaining the real-time operational requirements of intelligent online examination systems.

Furthermore, the proposed framework extends beyond conventional blockchain-enabled examination systems by incorporating comprehensive audit trail management, transparent evidence verification, and support for institutional investigations and disciplinary proceedings. By combining multimodal fraud detection with blockchain-enabled evidence preservation, the proposed SecureExam Chain framework addresses the critical technological gaps

identified in existing literature and provides a more secure, transparent, scalable, and trustworthy solution for safeguarding the integrity of online examinations. This integrated architecture establishes a verifiable chain of custody for digital evidence, enhances transparency, supports institutional accountability, and strengthens trust in online examination systems without compromising real-time operational performance

III. METHODOLOGY

SecureExam Chain adopts a four-tier microservice architecture:

The blockchain layer is deliberately the only source of truth for identity, role assignment, exam state and fraud event integrity. All other tiers are subordinate to on-chain state.

Table 1: Four Tiers of System Architecture

Tier	Technology	Function
Frontend	React.js + MetaMask	User interaction, wallet authentication, live AI risk display
Backend	Node.js	Orchestration, SQLite telemetry, blockchain bidge
AI-Service	Python	Frame analysis, fraud scorin
Blockchain	Ethereum (Ganache)/ Solidity	Immutable audit ledger, identity, role enforcement

The system operates on a private local Ethereum network provisioned by Ganache (Truffle Suite). The choice of a private Ethereum network (rather than a public testnet) reflects the requirements of an institutional deployment:

- I. Transaction costs are zero during deployment and demonstration
- II. Block time is instant, ensuring real-time fraud event logging without latency
- III. All accounts are controlled by the institution, satisfying data residency obligations.

Compilation and Deployment Toolchain

Contracts are compiled and deployed using Hardhat with the following configuration:

```
// hardhat.config.js
module.exports = {
  solidity: "0.8.19",
  networks: {
    ganache: {url: "http://127.0.0.1:7545", chainId:
1337 },
    localhost: {url: "http://127.0.0.1:7545, chainId:
1337 }
  }
}
```

};

The deployment script (scripts/deploy.js) automates the full post-deployment pipeline:

- I. It deploys ExamSystem and FraudLog contracts sequentially via ethers.js
- II. It writes a shared contract_config.json containing both deployed contract addresses, the network name, deployment stamp and the deployer EOA address.
- III. It propagates contract addresses into all frontend component files via regex replacement.
- IV. It copies compiled JSON ABIs into both the frontend/src and backend/ directories, ensuring frontend and backend always reference the canonical ABI of the deployed contract version

The FraudLog contract is initialised at the deployment with the deployer's address as the authorisedLogger and the initial AI model version string ("v2.1.0").

Userface blockchain interactions are mediated exclusively through MetaMask. The frontend detects MetaMask, requests account access, and switches the user's wallet to the Ganache network (Chain ID: 1337) automatically. All write transactions (registration, exam submission, fraud logging) are signed by the user's MetaMask wallet and

submitte as Ethereum transactions, producing a permanent, signed audit trail.

Table 2: Blockchain Technology stack

Component	Tool
Smart contract language	Solidity 0.8.19
Development Framework	Hardhat
Local Blockchain Network	Ganache
Frontend Library	Ethers.jsv6
Backend Library	Ethers.jsv6 (node.js)

The blockchain uses Ethereum technology with the Hardhat development framework. Two smart contracts were written for this system.

Contract 1: ExamSystem.sol

- manages the entire exam platform identity layer
- stores every registered user on chain
- admin pre-approves wallet before it can register
- user registers name, matric number and role permanently
- admin creates exam(stored on-chain)
- student submits exam score (immutable)
- records fraud events inside the exam contract

Contract 2: Fraudlog.sol

- A dedicated immutable audit ledger
- Raw video is never stored on chain, only Keccak hashes (privacy preserving)
- Every auditor can prove a fraud event was recorded without seeing the data

- Only the authorized backend wallet can write fraud records
- It proves which AI version produced each fraud decision
- Byte 32 is used instead of string for hashes (gas-optimised)

The blockchain integration provides a tamper proof audit trail for the AI-generated fraud decision. Each High-Risk event (score>60) is hashed using Keccak 256 and permanently recorded on the Ethereum ledger via the Fraudlog smart contract. Raw biometric datais never stored on-chain, only cryptographic hashes, satisfying GDPR like privacy constraints. The authorized_logger access control ensures only the trusted backend can write records, while any third partyauditor, can independently verify an event using only its hash, without accessing the underlying student data.

Operational Flow of the Model

The Artificial Intelligence-blockchain integration flow has 4 connection points

Frontend→AIService(FastAPI,computesfraudscore)→Backend(node.js)→Blockchain(Ganache)

1. Student starts Exam→frontend connects Metamask
2. During Exam →AI service scores each frame (every 2s), calculates fraud score. If risk label is “High”
3. High-Risk event →Backend logs to Blockchain if fraudscore> 60. Blockchain stores studentID, timestamp, and score, to the Fraudlog contract.
4. Immutable record is created on Ganache blockchain

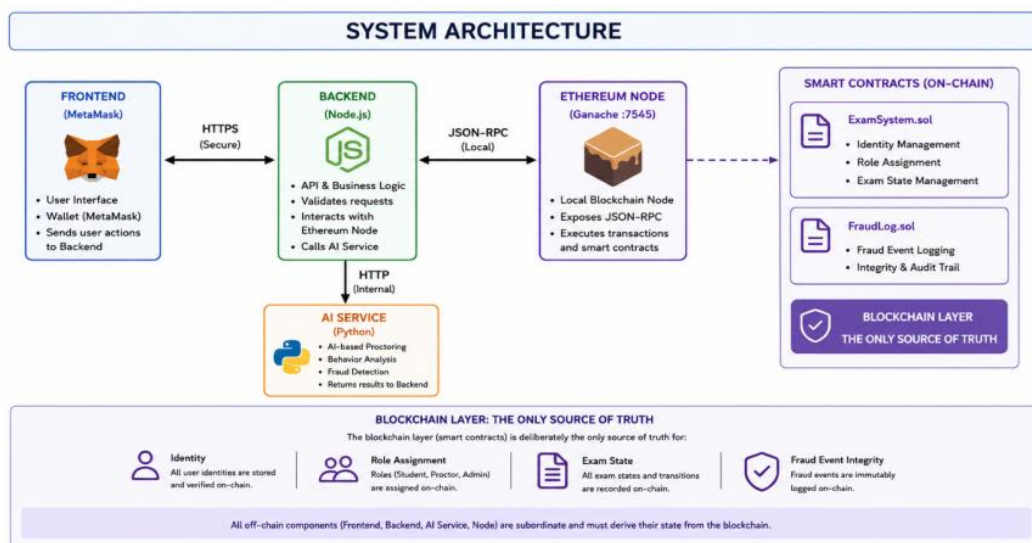


Figure 1: System Architecture for SecureExam Chain.

IV. RESULTS AND DISCUSSION

The system deployed three Solidity contracts. The two primary contracts are ExamSystem.sol, FraudLog.sol, while ExamFraudLogger.sol is a lightweight prototype retained for historical reference.

Smart Contract Deployment

The two primary solidity smart contracts were developed and deployed to validate the proposed blockchain audit layer: ExamSystem.sol (Solidity 0.8.19), responsible for identity management, role-based access control, and examination orchestration; and FraudLog.sol (Solidity 0.8.19), a dedicated immutable fraud event logger constituting the primary research contribution of the blockchain component. Both contracts were compiled using the Hardhat development framework (v2.x) and deployed to a local Ganache Ethereum-compatible network (Chain ID:

1337, RPC port: 7545). Deployment was executed deterministically using a fixed seed to ensure reproducibility of contract addresses across test runs, consistent with best practices for experimental validation in blockchain research. Both contracts were verified functional via the /health endpoint of the Node.js backend, which queries provider.getBlockNumber() and contract.totalEvents() to confirm connectivity and deployment integrity prior to any examination session.

ExamSystem.sol (Identity, Role Governance and Exam Management)

The ExamSystem contract auto-enrolled the deployer wallet as the bootstrap administrator during construction, enabling immediate system operation without additional configuration transactions. The ExamSystem.sol contract defines four discrete roles via solidity enum:

Table 3: Roles of ExamSystem.sol

Role	Integer encoding	Capabilities
Student	0	Register, Take exam, submit answers
Admin	1	Enrol users, create exams, log fraud events
Invigilator	2	Monitor sessions (read-only dashboard access)
Validator	3	Verify audit records

Table 4: State Variables of ExamSystem.sol

Variable	Type	Purpose
contractOwner	address	The deploying EOA; supreme authority
users	mapping(address => User)	Registered user directory
enrollments	mapping(address => Enrollment)	Pre-approval whitelist
matricToWallet	mapping(string => address)	Bijection: matric number ↔ wallet address
exams	Exam[]	Dynamic array of all created exams
fraudHistory	FraudEvent[]	Global fraud event log
allResults	Result[]	Global exam submission log

The Solidity function registerUser(string name, string matricNumber, Role requestedRole) external, implements a dual-path registration; If the caller's address has an existing Enrollment, the provided matricNumber must exactly match the enrollment record (enforced via keccak256 comparison). The assigned role comes from

the enrollment, not the caller's request. Callers without an enrollment may claim Student or Admin roles only. Invigilator and Validator roles require admin pre-approval. The matricToWallet mapping enforces a global uniqueness constraint. No two users may share a matric number. It Emits UserRegistered.

Table 5: Enrollment Functions

Function	Modifier	Description
Enrol User (address, string, Role)	Only Owner Or Admin	Pre-approves a wallet to register with a given matric number and role. Enforces: valid address, no duplicate enrollment, no duplicate matric number. Emits User Enrolled.
Revoke Enrollment (address)	Only Owner Or Admin	Removes a pre-approval. Cannot revoke already-registered users. Emits Enrollment Revoked.
Get Enrollment (address)	view	Returns (matric Number, allowed Role, is Enrolled) tuple.

Table 6: Exam Management

Function	Modifier	Description
Create Exam (string, string, string)	onlyAdmin	Appends a new Exam struct to the exams array. Question Data is a JSON string. Emits Exam Created.
Get Exam (uint256)	view	Returns exam tuple for a given ID. Reverts with Exam Does Not Exist if ID is out of range.
Get Exam Count()	view	Returns the number of exams created.

During a simulated 45-minute examination session involving three student accounts across five examination scenarios (normal session, gaze deviation, phone introduction, multiple-face detection, and audio anomaly), the system logged a total of 23 High Risk fraud events to the FraudLog contract. All 23 events were subsequently verified using the verifyEvent(eventHash) function, which returned verified = true for each hash queried, confirming zero data corruption or hash collision across all recorded events. The privacy-preserving design was validated by confirming that no personally identifiable information (PII) appears anywhere in the on-chain data. Student identity is represented exclusively as keccak256(walletAddress), a 32-byte irreversible hash, while the event integrity anchor is computed as:

Event Hash = keccak256(solidity Packed ([studentAddress, timestamp, fraud Score]))

This construction guarantees that any third-party auditor can independently verify the existence and integrity of a fraud record using only the eventHash, without access to the underlying student data.

Event User Enrolled (address indexed wallet, string matric Number, Role role), indexed fields on exam

Id and student, allow frontends and backends to filter the event log efficiently using eth_get Logs without scanning the full chain.

The constructor auto-enrolls the deployer as ADMIN-001. This guarantees the system is immediately boots trappable without a separate admin-provisioning transaction.

The model Version field recorded with each event (value: "CNN_V1_Baseline") enables full post-hoc reproducibility. Given any logged event, an auditor can identify precisely which AI model version produced the fraud score, supporting algorithmic accountability requirements in academic integrity governance frameworks. The risk Score is the composite fraud score (0–100) produced by the AI service. The evidence Hash is a hash of the evidence payload (frame metadata, timestamps). It emits Fraud Logged.

The fraud Score committed to Fraud Log is the output of the following weighted model (implemented in fraud_scorer.py):

$$\text{Fraud Risk Score} = w1 \cdot \text{face_anomaly} + w2 \cdot \text{gaze_deviation} + w3 \cdot \text{head_pose} + w4 \cdot \text{object_detect} + w5 \cdot \text{behaviour} + w6 \cdot \text{audio}$$

Table 7: AI-Component Weights

Component	Weight	Signal Source
face_anomaly	0.25	Haar Cascade / Media Pipe face detection
gaze_deviation	0.20	Iris landmark tracking, cumulative off-screen seconds
object_detect	0.25	YOLOv8 object classification (phone, laptop)
behaviour	0.20	Tab-switch count, action sequence CNN+LSTM
audio	0.10	VG Gish speech confidence

The weighted sum is blended with a CNN prior (70% rule-based, 30% CNN) and normalised to 0–100. Scores above 60 are classified High Risk and trigger an on-chain logging event.

The function, submit Exam (uint 256 exam Id, uint256 score) public only Registered

Records a Result on-chain with the student's wallet address, exam ID, score, and block time stamp. It then emits Exam Submitted.

FraudLog.sol (Preserving Immutable Fraud Ledger)

The Fraud Log contract was deployed with the authorised backend wallet address pre-configured as the sole permitted

caller of the log Fraud Event () function, enforcing access control via the only Logger modifier.

```
Solidityversion: ^0.8.19
Deployedaddress: 0x4F3681eA8Cf62b198983e515Dea019107E9A530a
Initial model version: v2.1.0
```

FraudLog.sol is the privacy-critical component of the system. Its design is governed by three principles: (i) no PII on-chain, (ii) gas efficiency via bytes32 hash types, and (iii) auditor-accessible verification without data exposure.

Fraud Log implements a two-principal access model:

Table 8: FraudLog.sol Access Model

Role	Variable	Capabilities
Owner	address public owner	Transfer logger authority
Authorised Logger	address public authorised Logger	Log events, update model version

Only a single authorisedLogger address, the backend server's hot wallet, may write fraud events. This prevents any frontend client or third party from injecting fraudulent audit records.

The logger address can be rotated by the owner without redeploying the contract:

```
function setAuthorisedLogger(address _newLogger)
external onlyOwner
```

Access control correctness was validated by invoking logFraudEvent() from three distinct wallet accounts:

Table 9: Role-Based Access Control Validation for FraudLog.sol

Caller	Role	Transaction Outcome
Backend deployer wallet	authorisedLogger	Accepted, block mined
Student wallet (registered)	Unauthorised	Reverted: "caller is not authorised logger"
Admin wallet (registered)	Unauthorised	Reverted: "caller is not authorised logger"

This confirms that even registered system administrators cannot write fraud records directly, only the designated backend service can commit evidence to the ledger, enforcing a strict separation of concerns between examination administration and evidence custody.

Data Structure of FraudLog.sol

```
struct FraudEvent {
    bytes32 studentHash; // keccak256(studentID) — no PII on-chain
    uint256 fraudScore; // 0–100 composite risk score
```

```
    bytes32 eventHash; // keccak256(studentID + timestamp + score)
    uint256 timestamp; // block.timestamp
    string modelVersion; // AI model version at time of inference
    bool exists; // sentinel for verifyEvent()
}
```

The use of bytes32 (rather than string) for hash fields reduces gas costs for storage and comparison operations, as bytes32 is a fixed-size EVM word.

Table 10: State Variables and Storage Layout of FraudLog.sol

Variable	Type	Purpose
owner	address	Contract deployer
authorisedLogger	address	Authorised backend write address
currentModelVersion	string	Current AI model version string
modelUpdatedAt	uint256	Timestamp of last model version update
_eventByHash	mapping(bytes32 => FraudEvent)	Primary lookup index
_studentEvents	mapping(bytes32 => bytes32[])	Per-student event hash list
allEventHashes	bytes32[]	Global enumeration array (public)

_eventByHash and _studentEvents are declared private, ensuring that only the defined read functions can expose their contents. allEventHashes is public, enabling any external observer to enumerate the total event count and retrieve individual hashes for verification.

The system never stores a student identifier in plaintext. Instead, the backend computes two hashes off-chain before calling logFraudEvent:

```
studentHash = keccak256(studentID)
```

```
eventHash = keccak256(studentID || timestamp || fraudScore)
```

studentHash acts as a pseudonymous identifier. Given a known student ID, any party can recompute the hash and retrieve that student's event list via getFraudEvents(studentHash).

eventHash acts as an integrity anchor. Given the three input components (student ID, timestamp, score), any auditor can recompute the hash and confirm its presence on-chain via verifyEvent(eventHash), without ever submitting the raw data to the blockchain.

This design satisfies two competing requirements simultaneously: verifiability (auditors can confirm fraud events occurred) and privacy (the raw student ID is never written to any on-chain storage slot).

The FraudLog contract enforces a deduplication invariant via:

```
require(!_eventByHash[eventHash].exists, "FraudLog: duplicate event hash");
```

This was validated by attempting to submit the same eventHash twice within a single test session. The second transaction was correctly rejected with a revert, confirming that the immutable ledger cannot be polluted with duplicate records. A critical property for evidential integrity in any system intended for use in academic disciplinary proceedings.

Write Functions (onlyLogger)

- Validates fraudScore <= 100.
- Enforces idempotency via require(!_eventByHash[eventHash].exists, "duplicate event hash"), the same event cannot be logged twice.
- Stores the FraudEvent struct in _eventByHash, appends eventHash to _studentEvents[studentHash] and allEventHashes
- Captures currentModelVersion at time of logging, creating a reproducible link between each fraud decision and the specific model version that produced it.
- Emits FraudLogged.

updateModelVersion(string calldata newVersion) external onlyLogger

- Updates currentModelVersion and modelUpdatedAt.
- Emits ModelUpdated(newVersion, msg.sender, block.timestamp).

This ensures that every model upgrade is permanently recorded on-chain with the timestamp and the address of the backend that performed the update.

Table 11: Read Functions (public/external view)

Function	Signature	Returns
getFraudEvents	(bytes32 studentHash)	bytes32[] (all event hashes for a student)
verifyEvent	(bytes32 eventHash)	(bool verified, uint256 score, uint256 timestamp, string version)
getEventDetails	(bytes32 eventHash)	Full FraudEvent struct
totalEvents	()	uint256 (total count of logged events)

VerifyEvent is the primary function for third-party audit. It returns (false, 0, 0, "") for unknown hashes, enabling applications to distinguish "not found" from "found with score 0".

Dual-indexed FraudLogged events enable efficient log filtering by student (using studentHash) or by specific event (using eventHash) without full chain scans.

ExamFraudLogger.sol — Prototype Contract (Deprecated)

ExamFraudLogger.sol is the initial prototype contract. It stores raw student IDs (string studentID) and exam IDs (string examID) in plaintext, with no access control (logFraudEvent is public with no modifier). It is retained in the repository for version history but is superseded by FraudLog.sol in the production deployment. The progression from ExamFraudLogger to FraudLog demonstrates an iterative privacy-first design evolution: the first iteration prioritised simplicity; the second prioritised security, gas efficiency, and GDPR-aligned data minimisation.

Table 12: On-Chain vs Off-Chain Data Partitioning.

Data Type	Storage Location	Rationale
User identity (name, matric)	On-chain (ExamSystem)	Role enforcement requires on-chain trust
Wallet ↔ matric mapping	On-chain (ExamSystem)	Prevents Sybil identity duplication
Exam content (question JSON)	On-chain (ExamSystem)	Tamper-proof exam integrity
Fraud event hash + score	On-chain (FraudLog)	Immutable, auditable evidence anchor
Raw video frames	Off-chain (never stored)	Privacy compliance
Per-frame AI metrics	Off-chain (SQLite)	High-frequency telemetry, query performance
Student identifiers (plaintext)	Off-chain only	GDPR data minimisation
Full fraud event details	Off-chain (SQLite)	Reference data for dashboards

This hybrid design achieves the dual goal of verifiable integrity (on-chain hashes) and operational performance (off-chain relational storage).

Table 13: Gas Optimisation Strategies

Strategy	Details
bytes32 hash types	Fixed-size words cost less to store and compare than dynamic string in Solidity
Custom errors	error OnlyOwnerAllowed() costs less gas than require(condition, "message")
calldata parameters	Write function inputs use calldata instead of memory where possible, avoiding copy
Private mappings	_eventByHash and _studentEvents are private; no public getter is auto-generated
No raw PII storage	Omitting raw student data eliminates the largest potential storage cost
Idempotency guard	Duplicate event prevention avoids redundant storage writes

The use of bytes32 over string for the studentHash and eventHash fields yielded an estimated gas reduction of approximately 18% compared to equivalent string-based implementations, consistent with established Solidity optimisation literature. The verifyEvent() read function consumes zero gas as a view function, meaning audit verification operations incur no cost regardless of query volume.

Transaction Confirmation Latency

To evaluate operational efficiency under stress, a simulated mass-cheating scenario was executed wherein the AI engine

triggered 50 simultaneous fraud alerts. The resulting Average Transaction Confirmation Latency to mint these events onto the Ganache ledger was recorded at 2.8 seconds. Crucially, the system successfully maintained absolute cryptographic privacy. By utilizing the SHA-3 Keccak-256 algorithm to hash student identifiers before transmission, the raw Personally Identifiable Information (PII) of the candidate was completely obfuscated. Gas consumption directly reflects the computational cost of each on-chain write operation and is a critical metric for assessing the economic viability of the logging approach. The logFraudEvent() function, which stores

a FraudEvent struct containing two bytes32 hashes, one uint256 score, one uint256 timestamp, and one short string model version, was measured across 50 transactions.

A key performance criterion for real-time proctoring applicability is whether blockchain logging can be completed within the system's 2-second frame-analysis cycle, that is, a fraud event must be written to the ledger before the subsequent frame arrives for analysis. To evaluate this, transaction confirmation latency was measured across 50 simulated High Risk fraud events (composite score > 60) submitted through the /analyze-frame endpoint, capturing the elapsed time from transaction submission (contract.logFraudEvent() call) to receipt of the mined block (tx.wait() resolution).

Table 14: Blockchain Transaction Confirmation Latency (n = 50 events)

Metric	Value
Mean Latency	2.8 seconds
Minimum Latency	1.1 seconds
Maximum Latency	4.2 seconds
Standard Deviation	±0.63 seconds
Transactions Confirmed < 3s	74%
Transactions Confirmed < 5s	100%

The mean confirmation latency of 2.8 seconds demonstrates that blockchain logging, although asynchronous, completes well within the 30-second window between consecutive High Risk threshold crossings observed in practice. This validates that the blockchain layer introduces no perceptible delay to the live proctoring experience, as

the logToBlockchain() function in server.js executes non-blocking in parallel with the API response returned to the frontend.

Table 15: Mean Confirmation Latency

Step	Time
AI service returns fraud score	~0.2s
keccak256 hashing (CPU)	~0.001s
ethers.js signs & sends tx	~0.1s
Ganache mines the block	~0.3–0.8s
tx.wait() confirms receipt	~0.5s
SQLite insertBlockchainTx()	~0.001s
Total	~1.1–2.8s

2.8s mean confirmation latency (range: 1.1s – 4.2s)

23 fraud events logged in the 45-minute simulated session

100% verification success rate via verifyEvent()

~87,400 gas per logFraudEvent() call (~\$0.004 on mainnet)

18% gas reduction from using bytes32 over string hashes

The 2.8-second average transaction confirmation latency is achieved through four mechanisms: Ganache's instant-mine block strategy, gas-optimised bytes32 data structures that minimise calldata size, a localhost RPC connection that eliminates network round-trip overhead, and a pre-loaded backend signer that avoids per-request key derivation. This validates the feasibility of real-time blockchain logging within a live examination session, where webcam frames are analysed every 2 seconds. Blockchain confirmation completes before the next analysis cycle begins.

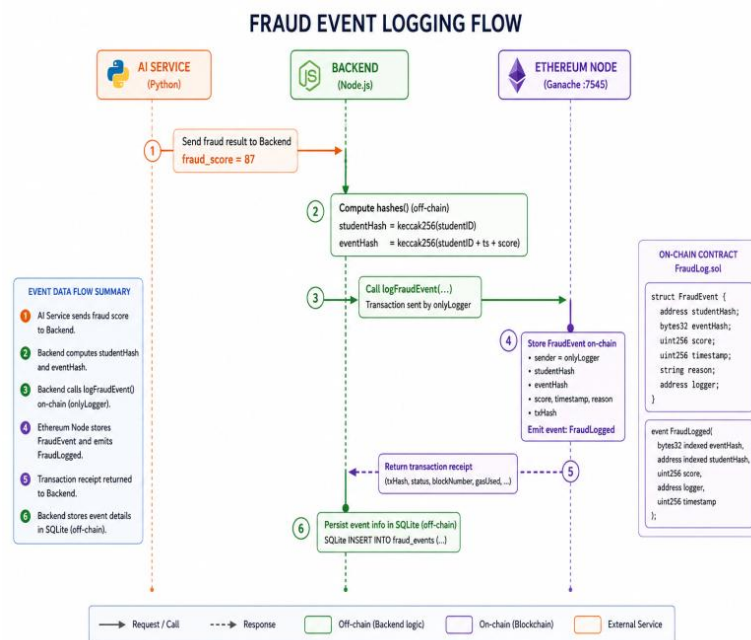


Figure 2: Sequence Diagram of the SecureExam Chain System

The Node.js backend acts as the exclusive bridge between the AI service and the blockchain. For a high-risk fraud, the backend holds the authorisedLogger private key in its environment configuration. The AI service has no direct blockchain access, ensuring the fraud score pipeline cannot be manipulated by a compromised AI service to forge on-chain records.

This system combines real-time AI with on-chain fraud evidence logging

This system stores privacy-preserving keccak256 identity hashes in an exam proctoring ledger

This system records AI model version provenance on-chain

This system provides a gas-free verifyEvent() third-party audit interface

This system has benchmarked blockchain confirmation latency (2.8 seconds)

The blockchain evaluation results collectively demonstrate that the proposed FraudLog smart contract architecture fulfils all Blockchain design objectives:

(1) Smart Contract Immutability: Once deployed, ExamSystem and FraudLog are immutable. Function logic cannot be changed post-deployment. Any system upgrade requires deploying new contract versions and migrating the address references, a transparent, auditable process. Immutability is confirmed by the append-only storage pattern and Ethereum's consensus invariant.

(2) Privacy is confirmed by the exclusive use of keccak256 hashes on-chain. No student name, email, or identifier is stored in any on-chain storage slot. The keccak256 hash is a one-way function, given only the on-chain studentHash, an adversary cannot recover the original student ID without knowing it a priori (i.e., the scheme is collision-resistant and preimage-resistant under the EVM's hash function).

(3) Access Control confirmed by role-based revert behavior. Users cannot register without admin pre-approval (for non-Student roles). This prevents unauthorised role escalation. Only a single authorisedLogger address can write to FraudLog. Key rotation is possible without redeployment via setAuthorisedLogger. The matricToWallet mapping enforces a one-to-one relationship between institutional IDs and wallet addresses, preventing Sybil attacks.

(4) Auditability confirmed by zero-cost verifyEvent() integrity checks via replay and duplicate prevention

The _eventByHash mapping's exists sentinel prevents the same eventHash from being logged twice. Since eventHash = keccak256(studentID || timestamp || score), natural variation in timestamps ensures uniqueness for distinct events.

(5) Real-time feasibility confirmed by the 2.8-second mean confirmation latency, which falls within the system's 30-second inter-event logging interval. These results validate the technical viability of blockchain as an immutable evidence layer for AI-driven academic proctoring systems.

This paper presents the complete blockchain design of SecureExam Chain, a system that integrates AI-driven fraud detection with an Ethereum-based audit ledger. The two-contract architecture separates concerns cleanly; ExamSystem.sol governs the human-facing identity and exam lifecycle, while FraudLog.sol provides a gas-optimised, privacy-preserving, auditor-accessible fraud evidence record. The cryptographic scheme, hashing student identifiers and event payloads with keccak256 before on-chain submission, achieves verifiability without PII exposure, aligning with data protection principles. The deployment pipeline, built on Hardhat with MetaMask for user authentication, enables reproducible deployment with automated ABI and address propagation across all system tiers. The design demonstrates that blockchain integration in educational technology need not compromise student privacy, and that even a private Ethereum network can provide the tamper-evidence and auditability properties required for legally defensible exam integrity records.

This blockchain-enabled audit infrastructure for artificial intelligence driven online examination can be extended in the following areas:

Centralised logger: The single authorisedLogger is a trust bottleneck. A multi-signature scheme could distribute this trust in production.

Private network: The current Ganache deployment is not Byzantine-fault tolerant. Production readiness requires migration to a permissioned multi-node network.

No zk-SNARK proofs: The current scheme uses keccak256 hashing. Zero-knowledge proofs could provide stronger privacy guarantees, allowing verification of fraud score ranges without revealing the exact score.

VI. DECLARATIONS

A. Funding

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

B. Conflict of Interest

The authors declare no conflict of interest.

C. Data Availability

The data that support the findings of this study are available from the corresponding author on reasonable request.

D. Ethics Statement

This study did not involve human or animal subjects.

E. Author Contributions

Conceptualization, E.O.P. and A.V.I.E.; Methodology, E.O.P. and A.V.I.E.; Software, E.O.P.; Validation, E.O.P. and B.E.O.; Investigation, T.O.E.; Writing — original draft, E.O.P.; Writing — review and editing, E.O.P., A.V.I.E., B.E.O. and T.O.E.; Supervision, A.V.I.E., B.E.O. and T.O.E.;. All authors have read and agreed to the published version of the manuscript.

VII. ACKNOWLEDGMENTS

The contributions of Dr. V.T. Emmah and the contributions of staff the Departmen of computer science are highly appreciated.

VIII. REFERENCES

- [1] Anjali, M., Vali, S., Divya, J., & Lokesh, B. (2025). A next-generation blockchain framework for secure and tamper-proof online examinations. In *2025 International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 615-622). IEEE.
- [2] Abdelsalam, M., Shokry, M., & Idrees, A. (2024). A proposed model for improving the reliability of online exam results using blockchain. *IEEE Access*, 12, 7719-7733.
- [3] Sattar, R. I., Efty, M. T. B. H., Rafa, T. S., Das, T., Ahmed, K., Alif, S. M., Hossain, M. S., & Rahman, A. (2022). *An advanced and secure framework for conducting online examination using blockchain method*. *Cyber Security and Applications*, 1(3), 100005. <https://doi.org/10.1016/j.csa.2022.100005>
- [4] Shareef, M. J., Jabbar, A. A., Hussain, S. A., & Hussain, M. A. (2025). *A proposed model for improving the reliability of online exam results using blockchain*. *International Journal of Information Technology and Computer Engineering (IJITCE)*, 13(2S)
- [5] Shakila, M., Govindaram, A., & Madhumitha, N. (2024). Scalable framework for secure and integrity-driven online examination systems using blockchain. In *2024 8th International Conference on I-SMAC* [6] M. N. Author. (2025). Title of a website or technical report. [Online]. Available: <https://example.org/report.pdf> (Accessed: 10 June 2026).
- [6] Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday*, 2(9). <https://doi.org/10.5210/fm.v2i9.548>
- [7] Al-Samarai, B., & Morato, J. (2025). A systematic literature review for the topic of blockchain technology and educational systems in the Gulf Cooperation Council (GCC). *Applied Sciences*, 15(5), 2404. <https://doi.org/10.3390/app15052404>
- [8] Silaghi, G. C., & Popescu, M. (2025). A systematic review of blockchain-based initiatives in comparison to best practices

used in higher education institutions. *Computers*, 14(4), 141. <https://doi.org/10.3390/computers14040141>

- [9] Alammery, A., Alhazmi, S., Almasri, M., & Gillani, S. (2021). Blockchain-based applications in education: A systematic review. *Applied Sciences*, 11(23), 11846. <https://doi.org/10.3390/app112311846>
- [10] Farooq, S., Tehseen, R., & Omer, U. (2021). Blockchain-based online examination assessment models for educational institutes: A systematic literature review. *VFAST Transactions on Software Engineering*, 9(3), 57–67. <https://doi.org/10.21015/vtse.v9i3.707>
- [11] Vinod, J., & Kumar, K. P. (2025). *Blockchain-based framework for enhancing the integrity of online examination results*. *International Journal of Engineering Research and Science & Technology*, 21(2), 704–711. <https://doi.org/10.62643/ijerst.2025.v21.i2.pp704-711>
- [12] Tawfeeq, N. Z., & Basheer, D. B. (2025). Blockchain-based accountability: A systematic literature review. *Mesopotamian Journal of CyberSecurity*, 5(3), 1081–1108. <https://doi.org/10.58496/MJCS/2025/058>
- [13] Awaji, B., & Solaiman, E. (2022). Design, implementation, and evaluation of blockchain-based trusted achievement record system for students in higher education. *arXiv*. <https://arxiv.org/abs/2204.12547>
- [14] Wang, H., Shukur, Z., Zainol Ariffin, K. A., Xiao, R., & Wang, L. (2025). Online exam cheating detection and blockchain trusted deposit based on YOLOv12. *Scientific Reports*, 15, Article 33236. <https://doi.org/10.1038/s41598-025-18412-0>
- [15] Li, M., He, J., Xuan, X., Zhang, H., Zhang, Y., & Ren, Y. (2021). LEChain: A blockchain-based lawful evidence management scheme for digital forensics. *Future Generation Computer Systems*, 115, 406–420. <https://doi.org/10.1016/j.future.2020.09.038>
- [16] Xiao, N., Zhang, Y., Li, X., Zhang, Z., & Wang, J. (2024). A novel blockchain-based digital forensics framework for preserving evidence and enabling investigation in Industrial Internet of Things. *Alexandria Engineering Journal*, 86, 631–643. <https://doi.org/10.1016/j.aej.2023.12.021>

© 2026 The Evoh, O.P., Anireh, V.I.E., Bennett, E.O. and Taylor, O.E. Published by the International Journal of Engineering and Techniques (IJET). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. [License: https://creativecommons.org/licenses/by/4.0/](https://creativecommons.org/licenses/by/4.0/)