Final Year Project: Interim Submission Report

Student Name: Tianxing Fan	**Student ID:** 20100035
Student Course: CGD	**Type of Project:** Artefact
Project Title: Blockchain-Powered Backend for the Strategic Research Nexus	
Supervisor: Dr. Salim Saay	**2nd Reader:** Dr. Tiziana Margaria

Table of Contents

1. Introduction

The Strategic Research Nexus (SRN) is a global academic platform designed to connect Afghan researchers, fostering evidence-based collaboration and supporting sustainable development initiatives through interdisciplinary knowledge exchange (SRN, 2025). At present, the platform operates primarily as a static website with limited dynamic functionality, relying on external tools such as Google Drive for publication sharing and Google Calendar for event coordination. While this configuration enables basic operation, it introduces challenges related to fragmented data management, inconsistency, and the absence of verifiable authenticity mechanisms, which can undermine confidence in research outputs hosted on the platform.

To address these challenges, this project proposes the design and implementation of a "Hybrid Blockchain Database Architecture." Unlike pure blockchain systems, which often suffer from the "Blockchain Trilemma" (Buterin, 2017), thereby sacrificing scalability in order to maintain decentralization and security, this project strategically couples a high-performance relational database (PostgreSQL) for data retrieval with a decentralized public ledger (Ethereum) for integrity verification. This approach is motivated by the specific infrastructural constraints of the SRN, which requires high availability and low-cost storage for large research artifacts—requirements that purely on-chain solutions cannot meet efficiently. By adopting this hybrid model, supported by recent research into ledger databases (Ge *et al.*, 2022), the system ensures that the platform retains the user experience standards of modern web applications while inheriting the tamper-resistant trust guarantees of blockchain technology.

This distinct architectural choice also fills a gap in current research. While existing blockchain solutions often assume large-scale institutional contexts with substantial technical resources, SRN represents a resource-constrained platform where accessibility is critical. Accordingly, this project investigates whether selective blockchain integration—restricted to hashing and verification rather than full data storage—can provide robust integrity assurances without the performance penalties of a fully decentralized system.

The overall goal of the project is to design a scalable backend architecture that unifies data management and enhances trust within the SRN platform. The intended deliverables of the project include:

1. A functional prototype backend developed using ASP.NET Core, PostgreSQL, and Solidity smart contracts.
2. Middleware integration via the Nethereum library to support blockchain interactions.
3. An evaluation of performance, trust, and usability characteristics of the proposed hybrid architecture.
4. Comprehensive technical documentation to support future deployment within SRN's existing hosting environment.
5. A demonstrable prototype deployed in a test environment to assess real-world feasibility.

It is important to clarify that this project focuses solely on technical artefact development and backend system architecture. Throughout this report, any reference to 'users' strictly denotes internal system administrators or authorized maintenance staff interacting with the backend interface. No external participants, public end-users, or human subjects are involved in the design, testing, or evaluation of this system. The project's evaluation methodology is therefore confined strictly to technical performance benchmarking and expert heuristic interface inspection (Nielsen, 1994).

2. Literature Review

The application of blockchain technology in academia has evolved from simple credentialing pilots to complex, hybrid systems designed to secure research ecosystems. This review examines the progression from institutional-scale solutions to lightweight architectures suitable for resource-constrained environments like the Strategic Research Nexus (SRN).

2.1 Blockchain in Academic Credentialing and Verification

Early implementations of academic blockchain systems, such as MIT's digital diploma pilot (Durant and Trachy, 2017), focused on issuing immutable certificates using the Bitcoin blockchain. While pioneering, these systems relied on "full-node" verification, which requires significant technical overhead. More recent research has shifted towards solutions tailored for developing nations where institutional trust may be fragile. For instance, Farabi *et al.* (2025) propose a blockchain-powered academic credential verification system for Bangladesh. Similar to the context of the SRN in Afghanistan, ShikkhaChain utilizes Ethereum smart contracts to combat credential fraud in a region with limited digital infrastructure. However, unlike ShikkhaChain, which focuses primarily on static degree certificates, the SRN project addresses the more dynamic challenge of verifying continuous research outputs (publications and datasets), necessitating a more flexible metadata schema. Additionally, Cardenas-Quispe and Pacheco (2025) recently demonstrated a degree verification prototype emphasising Byzantine consensus mechanisms, further validating the use of distributed ledgers for academic integrity.

2.2 Theoretical Basis for Hybrid Blockchain Database Systems

A critical debate in current literature concerns the trade-off between "full decentralization" and "system performance." While traditional relational databases are often critiqued in blockchain discourse for their centralized nature, recent scholarship by Ge *et al.* (2022) on Hybrid Blockchain Database Systems highlights their indispensability for high-volume data management. They argue that for data-intensive applications, a pure blockchain architecture is often infeasible due to the "Blockchain Trilemma" (balancing scalability, security, and decentralization). Storing large PDF artifacts directly on-chain leads to "blockchain bloat" and prohibitive gas fees. Ge *et al.* (2022) further classify systems into "Permissioned Blockchains," "Ledger Databases," and "Hybrid Systems." They demonstrate that hybrid models—where a relational database handles state management (high throughput) and a blockchain handles state verification (high trust)—significantly outperform pure decentralized applications (dApps) in query latency while maintaining tamper-evidence. This project adopts this "Hybrid" theoretical framework, using the database as a "Performance Layer" and the blockchain as a "Verification Layer."

2.3 Lightweight Protocols for Resource-Constrained Environments

For small-scale platforms like SRN, minimizing computational cost is paramount. Mahmoud *et al.* (2023) and Dorri *et al.* (2017) discuss "Lightweight Blockchain Protocols" in the context of IoT and edge computing. They argue that resource-constrained entities (like NGOs or small research groups) should not operate as full nodes but rather utilize "Gateway Architectures." In this model, a trusted backend acts as an oracle to the blockchain network. Mandinyenya and Malele (2025) support this approach in their framework for personal data sharing, recommending Off-Chain Storage (such as local databases or IPFS) combined with on-chain cryptographic proofs to ensure privacy and scalability. Synthesising these studies, it is evident that a gap exists in the literature regarding lightweight, post-publication verification tools for decentralized research networks in conflict zones. This project addresses this gap by implementing the "Hybrid" pattern to create a cost-effective, tamper-evident backend that does not require the end-user to possess specialized crypto-wallets or technical expertise.

3. Methodology

This project adopts the Design Science Research Methodology (DSRM) as proposed by Peffers *et al.* (2007), which is well suited to artefact-based research aimed at designing and evaluating IT solutions for organisational problems. DSRM emphasises the structured progression of research through six phases—problem identification and motivation, definition of objectives, design and development, demonstration, evaluation, and communication—providing a rigorous framework for developing and validating IT artefacts within real-world constraints.

In this project, DSRM is used as a guiding framework rather than a rigid checklist. Given the interim nature of this submission, the methodology focuses primarily on the early and mid-stage phases of DSRM, namely problem identification, objective definition, and system design and development, while later phases such as demonstration, evaluation, and communication are planned for the final stage of the project.

3.1 Problem Identification and Requirements Analysis (Completed)

An initial analysis of SRN's existing workflows was conducted through site inspection and document review to identify critical trust points, including publication uploads, metadata management, and event logging. This analysis revealed key requirements such as the need for unified data management to reduce fragmentation, cryptographic verification to enhance authenticity, and seamless system integration to minimise disruption for users.

Several constraints were identified, including limited technical resources and the absence of dedicated servers. These constraints informed the selection of open-source technologies such as PostgreSQL for data storage and Ethereum test networks for blockchain experimentation. Ethical considerations at this stage include data privacy and responsible handling of user information; these are addressed by ensuring that only cryptographic hashes, rather than raw content, are anchored on the blockchain. Potential points of failure, such as delays in blockchain transaction confirmation, were identified early and are addressed through the design of asynchronous processing mechanisms and fallback storage strategies.

3.2 System Design and Development (In Progress)

Based on the identified requirements and the theoretical "Hybrid" framework, the system architecture has been designed to bridge the gap between SRN's existing static infrastructure and dynamic blockchain verification (see Appendix A). The system employs an ASP.NET Core application for API services, Entity Framework Core for relational data management using PostgreSQL, and Ethereum smart contracts written in Solidity for anchoring verification data (see Appendix D).

To address the challenge of integrating with SRN's existing Google Sites frontend (which does not support server-side code), this project employs a "Headless Architecture." The ASP.NET Core backend is designed as a decoupled, standalone RESTful API hosted on a separate cloud service (e.g., Azure App Service). Crucially, the backend specification mandates strict Cross-Origin Resource Sharing (CORS) policies, specifically designed to whitelist the srn.ie domain. This architectural decision explicitly supports future integration via standard client-side JavaScript (Fetch API) calls from the static frontend, without requiring the backend to manage the presentation layer. This API-First approach ensures that the sophisticated cryptographic logic remains isolated in the backend while maintaining compatibility with the existing "live" website infrastructure.

To maintain integrity between the off-chain PostgreSQL database and the on-chain Ethereum ledger, the system architecture incorporates a "Dual-Write" consistency strategy (see Appendix B). As defined in the system specifications, the proposed workflow is as follows:

1. The file metadata and hash are immediately stored in PostgreSQL with a status of "Pending Verification."
2. A background worker service (utilizing the Nethereum library) is configured to pick up pending records and broadcast the hash to the Sepolia testnet.
3. The service is designed to listen for the specific DocumentAnchored event emitted by the smart contract. Upon confirmation, it updates the local database record to "Verified" and appends the transaction hash. This mechanism is designed to ensure that the centralized database remains a reliable "cache" of the on-chain truth, mitigating the risk of data tampering identified in the threat model.

The database schema (see Appendix C) has been designed and normalised to the Third Normal Form (3NF) to ensure data integrity and reduce redundancy. Core entities include

tables for research artefacts and verification records, enabling a clear association between local records and corresponding blockchain transactions. These design decisions aim to support later evaluation of system performance and verification reliability.

Additionally, a preliminary security assessment has been conducted using the STRIDE threat modelling framework (Hernan *et al.*, 2006) to systematically identify potential vulnerabilities (see Appendix E). The analysis addresses risks across all six categories (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege). This structured approach ensures that security considerations are integral to the architectural design phase.

3.3 Evaluation Strategy (Planned)

The evaluation phase is planned for the later stage of the project and will assess the proposed system across performance, trust, and usability dimensions. Performance evaluation will focus on metrics such as transaction latency and system throughput, measured using industry-standard benchmarking tools such as Apache JMeter or k6 to simulate concurrent user loads. Trust will be assessed through scenario-based analysis, including simulated SQL injection or direct database tampering attempts to evaluate whether the system correctly flags discrepancies between the stored data and the on-chain hash. Usability will be examined using heuristic evaluation methods and limited user feedback to assess accessibility for non-technical users. This mixed-methods evaluation approach is intended to provide both quantitative and qualitative insights into the feasibility and effectiveness of the hybrid architecture, while remaining appropriate to the scope and constraints of an undergraduate final year project.

3.4 Project Plan and Milestones

The project implementation roadmap aligns with the phased schedule detailed in Appendix F. Work on the core ASP.NET Core API and smart contracts commences immediately during the Winter Break. February is reserved for the critical integration of Nethereum and the dual-write background service, ensuring the hybrid verification logic is functional. The focus shifts in March to system evaluation and security testing, culminating in the Draft Report submission. Finally, April is allocated for code freezing, bug fixing, and documentation refinement to meet the Product Submission deadline. Development will leverage the Ethereum Sepolia test network to validate smart contract interactions without incurring mainnet costs.

4. Conclusions

This interim report has outlined the progress made toward designing a hybrid backend architecture to enhance data integrity and trust within the Strategic Research Nexus platform. To date, the work completed aligns with the proposed timetable, with the problem analysis and core system design phases successfully concluded. These activities have established a clear understanding of SRN's technical constraints and trust-related requirements, forming a solid foundation for subsequent implementation.

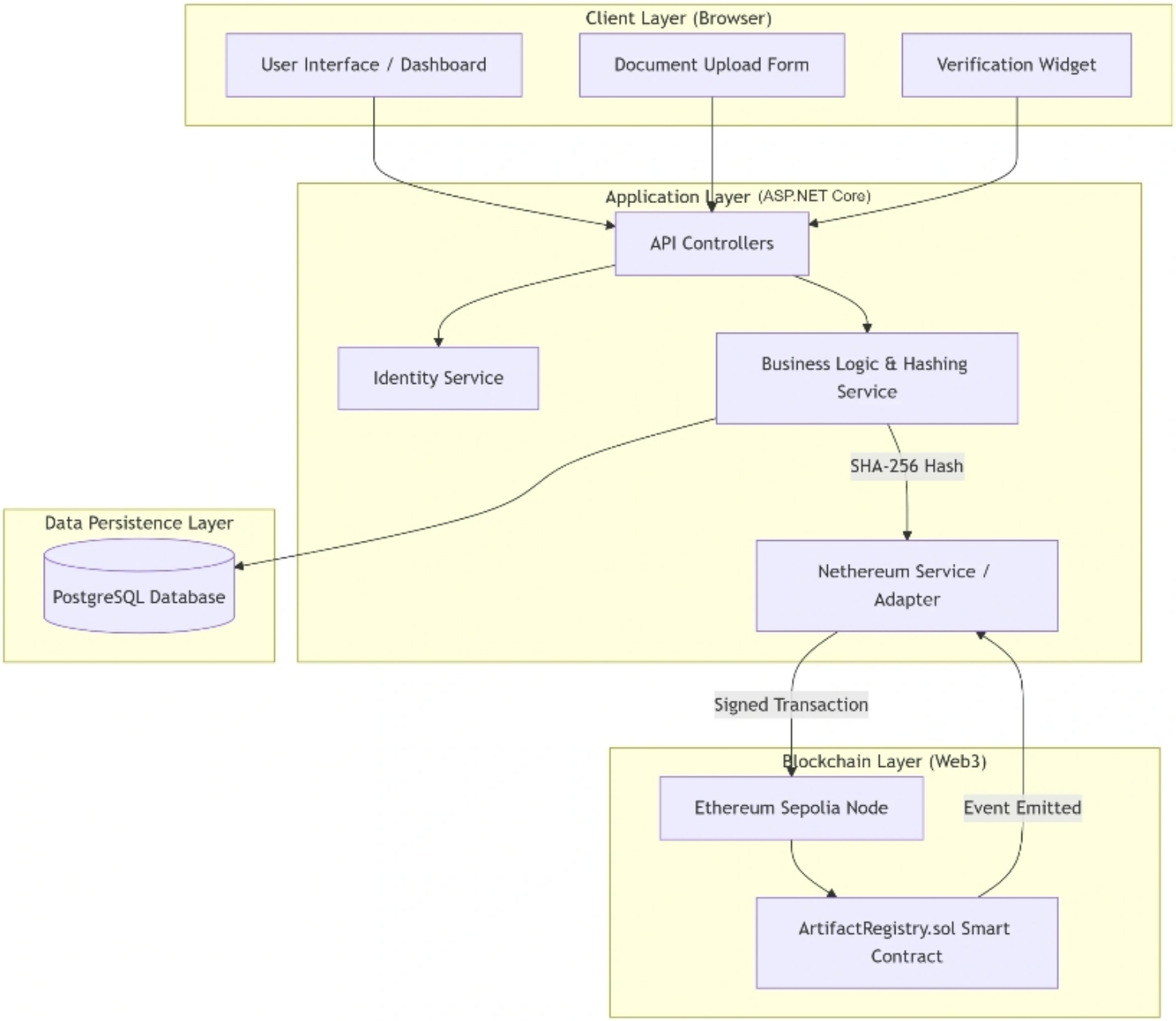
Relative to the project objectives, the work completed so far has focused on architectural feasibility rather than full system validation. The hybrid approach has been specified at the design level, addressing issues of data fragmentation through unified database modelling and incorporating blockchain-based verification through selective hashing. Initial proof-of-concept activities, such as successful hash anchoring on a blockchain test network, indicate the technical viability of the proposed approach, although comprehensive evaluation remains part of the planned future work.

No major unanticipated obstacles have emerged during the interim phase. However, practical considerations such as blockchain transaction costs and confirmation latency were identified early in the process. These constraints have informed the decision to rely on test networks and asynchronous processing mechanisms during development, and they will be further examined during the evaluation phase to assess their impact on usability and performance.

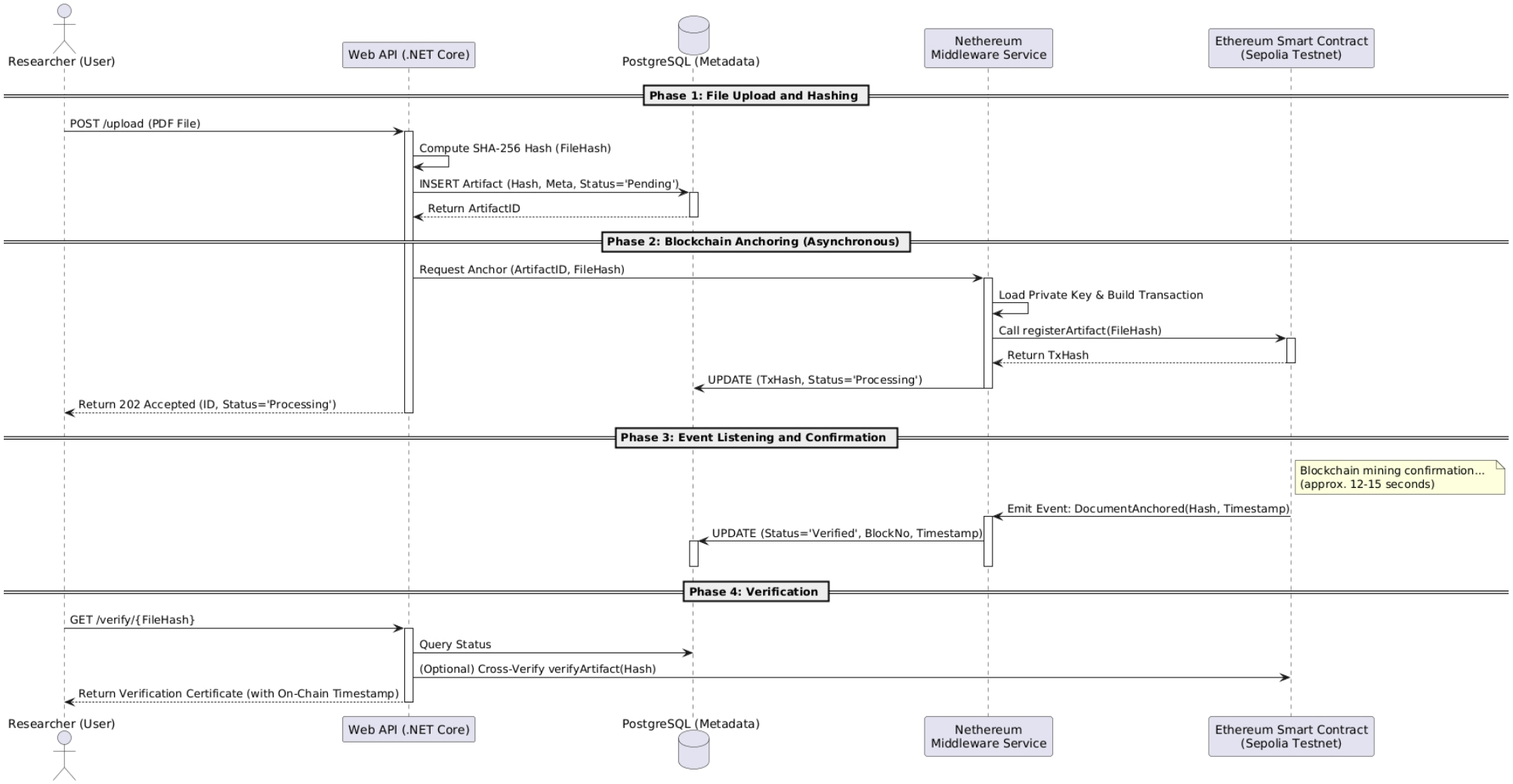
As of December 2025, the project environment has been fully prepared for the implementation stage (see Appendix G). The ASP.NET Core solution structure has been established, the PostgreSQL database schema and smart contract interfaces have been designed and documented, and the development environment—including containerisation and blockchain test network connectivity—has been configured. The next phase of the project will focus on implementing the designed components, integrating the hybrid verification workflow, and conducting systematic evaluation in line with the project methodology.

Appendices

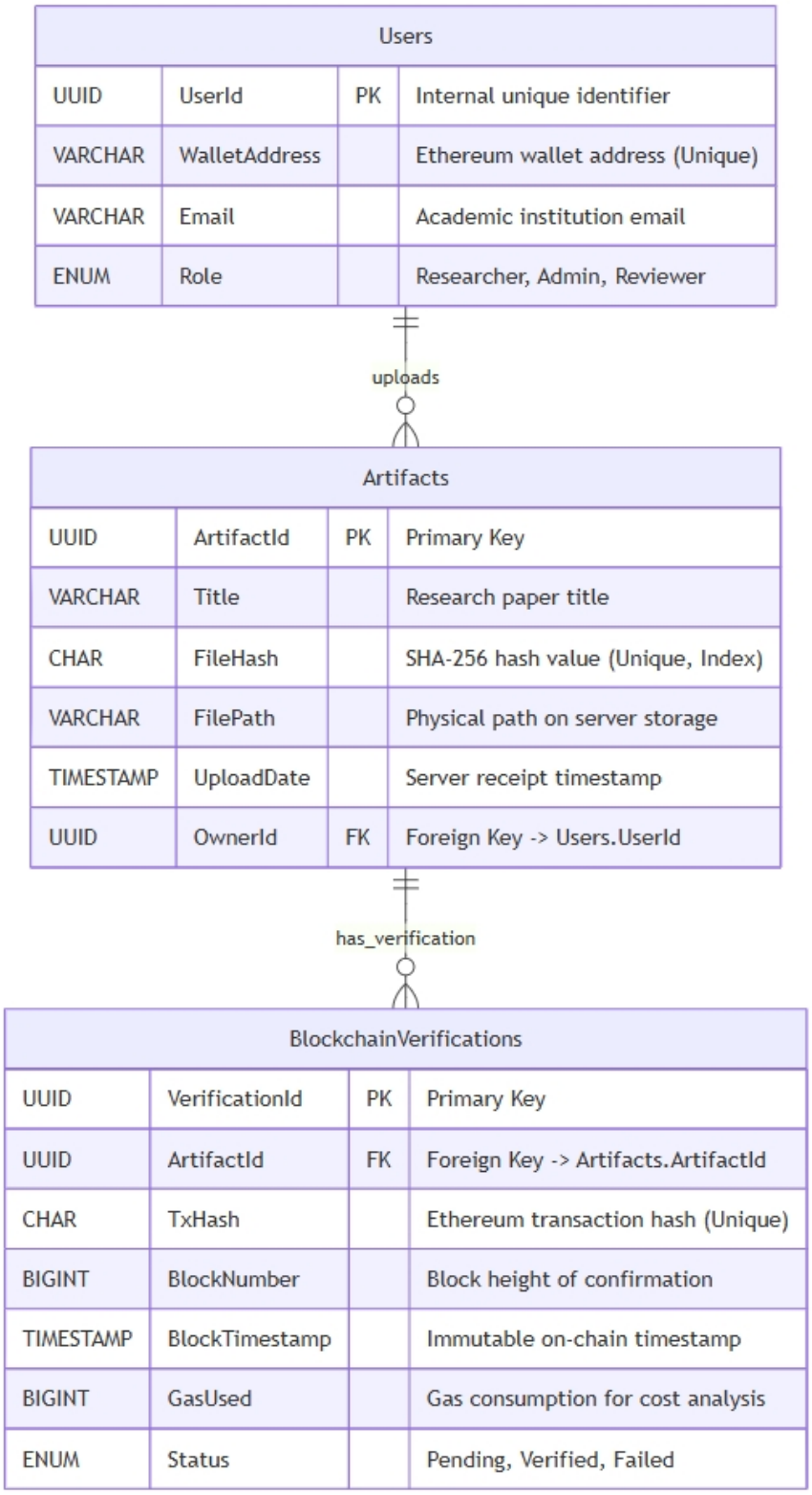
Appendix A: System Architecture Diagram



Appendix B: UML Sequence Diagram



Appendix C: Database ER Diagram



Appendix D: Smart Contract Interface

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

/**
 * @title IArtifactRegistry
 * @dev Core interface definition for SRN decentralized verification system
 *      Implements Proof of Existence logic
 */
interface IArtifactRegistry {

    /**
     * @dev Core data structure: Pack verification information for optimized SLOAD
operations
     */
    struct Record {
        bytes32 documentHash; // File's SHA-256 fingerprint (32 bytes)
        address owner;        // Uploader's wallet address
        uint256 timestamp;    // Block timestamp (authoritative time)
        bool isRegistered;    // Flag to prevent duplicate registrations
    }

    /**
     * @dev Event: Document anchoring notification
     *      indexed keyword allows backend Nethereum service to efficiently filter
and listen for specific hashes
     */
    event DocumentAnchored(
        bytes32 indexed docHash,
        address indexed owner,
        uint256 timestamp
    );

    /**
     * @dev Core function: Register a new document hash
     * @param _hash Document's SHA-256 hash value
     * Requires access control: Only allowed for authorized middleware accounts
     */
    function registerArtifact(bytes32 _hash) external;

    /**
     * @dev Core function: Verify document existence
     * @param _hash Hash to verify
     * @return registered Whether registered
     * @return timestamp Registration time
     * @return owner Owner address
     */
    function verifyArtifact(bytes32 _hash) external view returns (
        bool registered,
        uint256 timestamp,
        address owner
    );
}
```

Appendix E: Threat Model

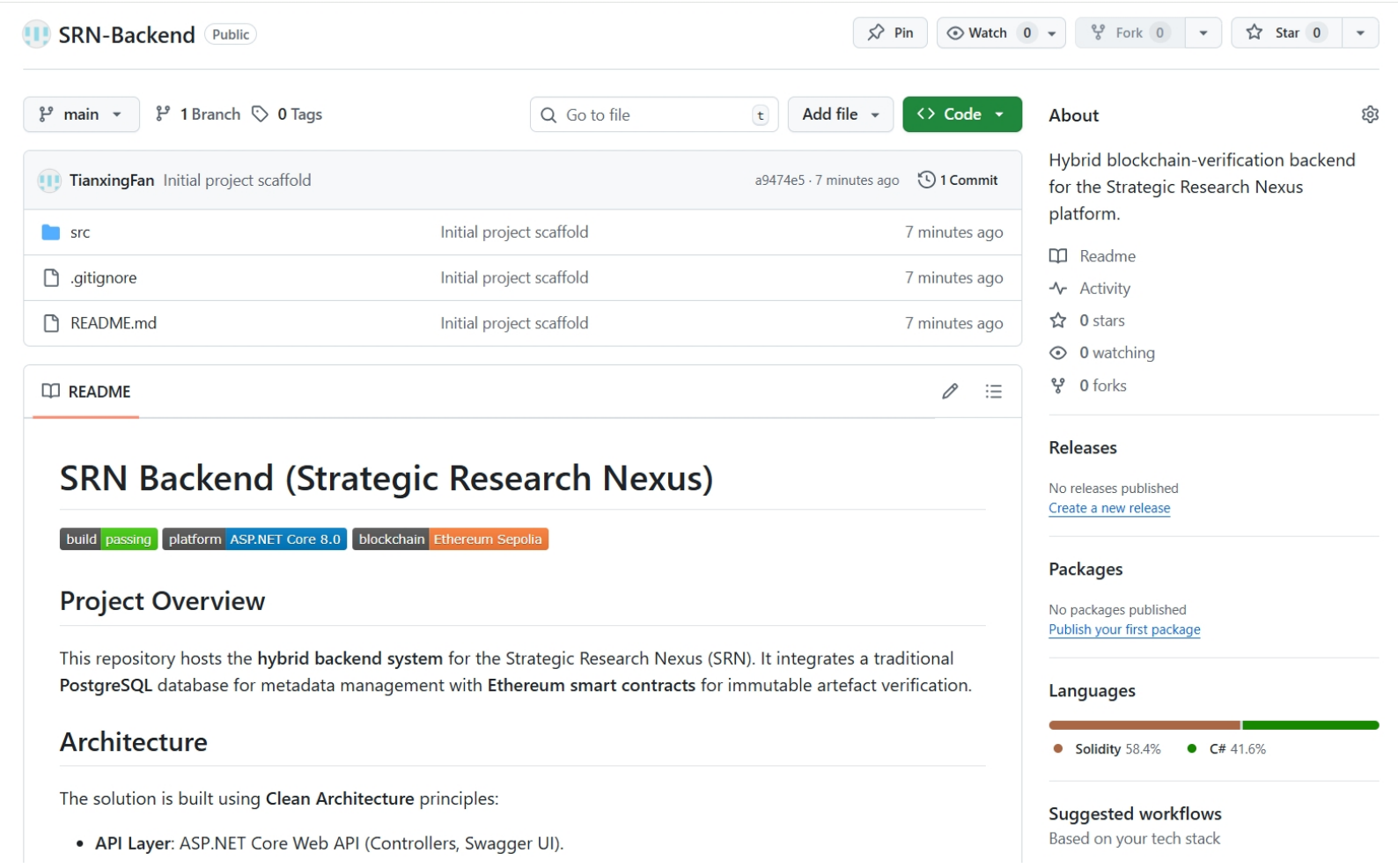
Threat Category	Potential Vulnerability Scenario	Mitigation Strategy in Architecture Design
Spoofing	Attacker impersonates server private key to send false attestation to contract.	Private keys are not stored in source code or config files. They are dynamically injected at runtime via secure secret management (e.g., User Secrets or Environment Variables). Smart contract uses AccessControl to restrict write permissions.
Tampering	Insider directly modifies UploadDate or metadata in the PostgreSQL database.	Verification logic recalculates the file hash and queries the smart contract. If DB data is tampered with, its hash will not match the immutable on-chain record, causing verification failure.
Repudiation	Author claims they never uploaded a controversial paper.	System requires user to sign the file hash with their Web3 wallet (e.g., MetaMask) during upload. Server stores this signature as non-repudiable evidence of authorship.
Information Disclosure	Unpublished sensitive research data viewed on public blockchain.	Only the 32-byte SHA-256 hash is stored on-chain. As a one-way function, original paper content cannot be derived from the on-chain data.
Denial of Service	Malicious user uploads large files at high frequency to exhaust server's ETH Gas fees.	Implement IP rate limiting at API gateway; enforce daily upload quotas per user; backend monitors Gas balance and implements an automatic circuit breaker.
Elevation of Privilege	Ordinary user attempts to call admin verification interfaces.	ASP.NET Core Identity implements strict Role-Based Access Control, combined with smart contract onlyRole modifiers for dual-layer permission verification.

Appendix F: Project Timeline

Phase	Duration & Dates	Key Activities	Deliverables
Foundation & Scaffolding	Up to Dec 23, 2025 (Completed)	Finalize local development environment. Design API & Database Schema. Establish Project Scaffolding & Git Repo. Draft Smart Contract Interfaces.	Interim Report, GitHub Repository (Scaffold), Dev Environment Ready.
Core Implementation (Winter Break)	Dec 24 - Jan 25, 2026 (In progress)	Commence Coding: Implement ASP.NET Core API Endpoints. Write and Unit-Test Smart Contracts. Setup Swagger UI for API testing. Connect PostgreSQL Database.	Functional Web API, Tested Smart Contracts, API Documentation (Swagger).
Blockchain Integration	Weeks 1 - 4 (Jan 26 - Feb 22, 2026)	Integrate Nethereum for transaction signing, Implement "Dual-write" background service, Deploy contracts to Sepolia Testnet, System Integration Testing.	Fully Integrated Hybrid Backend, "On-chain" Verification Working.
Evaluation & Draft Writing	Weeks 5 - 8 (Feb 23 - Mar 22, 2026)	Performance Testing (JMeter/Postman), Security Testing (SQLi/Tampering), Write Draft Report.	Performance Data Logs, Draft Report.
Refinement & Submission	Weeks 9 - 13 (Mar 23 - Apr 26, 2026)	Refine API Error Handling & Logs, Code Freeze & Bug Fixing, Final Report & Presentation Prep.	Product Submission, Final Report.

Appendix G: Environment Setup Screenshot

GitHub Link: https://github.com/TianxingFan/SRN-Backend



References

Buterin, V. (2017) Sharding FAQ, Vitalik Buterin's Website, available: https://vitalik.eth.limo/general/2017/12/31/sharding_faq.html [accessed 23 Dec 2025].

Cardenas-Quispe, M.A. and Pacheco, A. (2025) 'Blockchain ensuring academic integrity with a degree verification prototype', *Scientific Reports*, 15, Article 9281, available: https://www.nature.com/articles/s41598-025-93913-6 [accessed 23 Dec 2025].

Dorri, A., Kanhere, S.S., Jurdak, R. and Gauravaram, P. (2017) 'Towards an optimized blockchain for IoT', *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, 173-178.

Durant, E. and Trachy, A. (2017) 'Digital Diploma Debuts at MIT', *MIT News*, 17 Oct, available: https://news.mit.edu/2017/mit-debuts-secure-digital-diploma-using-bitcoin-blockchain-technology-1017 [accessed 23 Dec 2025].

Farabi, A., Khandaker, I., Ahsan, J., Shanto, I.K., Jahan, N. and Khan, M.J. (2025) 'ShikkhaChain: A Blockchain-Powered Academic Credential Verification System for Bangladesh', *arXiv preprint* arXiv:2508.05334, available: https://doi.org/10.48550/arXiv.2508.05334.

Ge, C., Loghin, D., Ooi, B.C., Wu, P. and Zhang, J. (2022) 'Hybrid Blockchain Database Systems: Design and Performance', *Proceedings of the VLDB Endowment*, 15(13), 1092-1105.

Hernan, S., Lambert, S., Ostwald, T. and Shostack, A. (2006) 'Uncover Security Design Flaws Using The STRIDE Approach', *MSDN Magazine*, available: https://learn.microsoft.com/en-us/archive/msdn-magazine/2006/november/uncover-security-design-flaws-using-the-stride-approach [accessed 23 Dec 2025].

Mahmoud, M.A., Gurunathan, M., Ramli, R., Mya, K.T., Chandrasekaran, K. and Kurniawan, A. (2023) 'Review and Development of a Scalable Lightweight Blockchain Integrated Model (LightBlock) for IoT Applications', *Electronics*, 12(4), Article 1025, available: https://www.mdpi.com/2079-9292/12/4/1025 [accessed 23 Dec 2025].

Mandinyenya, B. and Malele, V. (2025) 'A Hybrid Framework for Enhancing Privacy in Blockchain-Based Personal Data Sharing using Off-Chain Storage', *Journal of Information Security*, 12(1), 45-58.

Nielsen, J. (1994) 'Heuristic Evaluation', in Nielsen, J. and Mack, R.L., eds., *Usability inspection methods*, New York: John Wiley & Sons, 25-62.

Peffers, K., Tuunanen, T., Rothenberger, M.A. and Chatterjee, S. (2007) 'A Design Science Research Methodology for Information Systems Research', *Journal of Management Information Systems*, 24(3), 45-77, available: https://www.researchgate.net/publication/284503626_A_design_science_research_methodology_for_information_systems_research [accessed 23 Dec 2025].

SRN (2025) Strategic Research Nexus, available: https://www.srn.ie/ [accessed 23 Dec 2025].