

An AI-Driven Legacy and Privacy Management Framework Using Blockchain Technology

Akshara B

714522243005

*Final Year, Department of AIDS,
United Institute of Technology
Coimbatore***Dharsini K M**

714522243012

*Final Year, Department of AIDS,
United Institute of Technology
Coimbatore***Kabilan A**

714522243023

*Final Year, Department of
AIDS,
United Institute of Technology
Coimbatore***Kamals Rosan S R**

714522243302

*Final Year, Department of AIDS,
United Institute of Technology
Coimbatore***Ms.Saranya G**

AP/AI&DS,

*United Institute of Technology
Coimbatore*

Abstract - The exponential growth of digital platforms has led individuals to accumulate vast volumes of sensitive data, including personal documents, financial credentials, digital identities, and online account access. While modern cybersecurity infrastructure effectively mitigates external threats, it provides no mechanism for managing digital asset ownership continuity, privacy preservation, or authorized inheritance in the event of user incapacitation or death. This paper presents a novel AI-driven and blockchain-based framework for secure digital legacy and privacy management. The proposed system integrates an Artificial Intelligence (AI) monitoring engine that continuously tracks user activity and detects inactivity or emergency conditions, with Ethereum-based smart contracts that automate conditional access transfer to pre-verified nominees. Digital assets are stored using the InterPlanetary File System (IPFS) for decentralized and tamper-resistant file persistence, while all access events are permanently recorded on an immutable blockchain ledger. A multi-layer security architecture combining JWT-based authentication, bcrypt password hashing, Role-Based Access Control (RBAC), and OTP-based two-factor verification ensures that only authorized parties can interact with the system.

Key Words: Digital Legacy Management, Blockchain, Ethereum Smart Contracts, IPFS, AI Inactivity Monitoring, JWT Authentication, Nominee Management, Decentralized Storage, Digital Inheritance.

1. INTRODUCTION

In today's rapidly evolving digital landscape, individuals generate and store vast amounts of sensitive information across online platforms, cloud services, banking systems, healthcare portals, and social media networks. This data — encompassing personal documents, login credentials, financial records, digital assets, and private communications — collectively constitutes an individual's digital identity. While existing cybersecurity mechanisms are sophisticated in protecting data from unauthorized access, hacking, and external threats, they fundamentally overlook three equally critical aspects: data ownership continuity, long-term privacy management, and structured digital legacy planning.

The consequences of this oversight are significant. When a user becomes incapacitated, critically ill, or deceased, their family members and legal representatives frequently encounter

insurmountable barriers in accessing essential digital information. A 2023 digital asset management survey estimated that over 3.7 million Bitcoin — valued at hundreds of billions of dollars — have been permanently lost due to deceased owners leaving no access recovery mechanism. Beyond financial assets, social media profiles, email archives, and cloud-stored documents representing a lifetime of personal history are similarly at risk.

This paper presents a comprehensive AI-driven and blockchain-based framework that directly addresses these limitations. The proposed system, termed the AI Legacy and Privacy Guardian (ALPG), combines five tightly integrated modules: a secure user management and authentication layer, an encrypted digital asset and nominee management system, a Python-based AI monitoring engine for inactivity and behavioral anomaly detection, an Ethereum smart contract execution layer for automated and verifiable asset transfer, and a nominee notification and secure data retrieval subsystem.

The key contributions of this work are:

- A unified AI and blockchain framework for digital legacy management that automates inheritance transfer without dependence on centralized authorities or manual legal procedures.
- An AI monitoring engine employing behavioral analysis and exponential activity scoring to detect user inactivity with high accuracy, triggering smart contract execution only under verified conditions.
- Ethereum-based smart contracts that encode user-defined access rules and execute nominee transfers in a tamper-proof, transparent, and immutable manner.
- IPFS-based decentralized file storage ensuring that digital assets remain accessible even in the event of individual server failures, combined with symmetric encryption for content confidentiality.
- A multi-layer security architecture integrating JWT authentication, bcrypt password hashing, Role-Based Access Control, and OTP-based two-factor verification.

2. LITERATURE SURVEY

A systematic review of existing research in blockchain-based access control, AI-driven behavioral monitoring, digital legacy management, decentralized storage, and smart contract automation has been conducted. The following studies represent the most relevant prior work, and their limitations collectively motivate the design of the proposed ALPG framework.

A. AICyberChain: Combining AI and Blockchain for Improved Cybersecurity

Authors: Zia Ullah et al. | IEEE Access, 2024. This paper proposed an integrated architecture combining AI-based anomaly detection with blockchain-based secure data sharing and smart contract enforcement on the Ethereum test network. However, the scope is strictly limited to external cybersecurity threat detection and does not extend to personal digital asset management, ownership inheritance, or automated access control for authorized heirs.

B. Blockchain-Based Secure Data Sharing for Healthcare

Authors: Xu et al. | IEEE Transactions on Industrial Informatics. Xu et al. proposed a blockchain-based framework for secure and transparent sharing of medical records using smart contracts. While the technical mechanisms are well-developed, the framework is domain-specific and cannot be generalized to personal digital legacy management. It provides no mechanism for AI-based inactivity detection or conditional asset transfer based on user-defined life events.

C. Federated Learning for Privacy-Preserving Data Sharing

Authors: Montasari | IEEE Security and Privacy. This study employed a federated learning paradigm where ML models are trained locally without centralizing raw data. However, the work does not address digital ownership transfer, post-mortem asset management, or any form of legacy planning. No nominee notification workflow or blockchain-based immutable logging is present.

D. Blockchain-Based Access Control System

Authors: Schlatt et al. | IEEE Internet Computing. Schlatt et al. presented a decentralized identity and access management system using blockchain. However, it does not incorporate any form of AI-based automation, inactivity detection, or conditional trigger logic. All access decisions require active user participation, making it unsuitable for scenarios where the owner is incapacitated or deceased.

E. Smart Contract-Based Data Ownership Model

Authors: Khademi and Honavar | IEEE Blockchain Conference. This work demonstrated the use of Ethereum smart contracts to enforce data ownership rights. However, the system lacks integration with any AI monitoring component, does not support IPFS-based decentralized file storage, and provides no complete lifecycle from user registration through inactivity detection to nominee access.

Table -1: Literature Survey Comparison

S.No	Paper Title & Authors	Methodology	Limitations
1	AICyberChain (Zia Ullah et al., IEEE Access, 2024)	AI threat detection + blockchain on Ethereum testnet.	Lacks digital inheritance and nominee management.
2	Blockchain Secure Data Sharing for Healthcare (Xu et al.)	Blockchain framework for medical data using smart contracts.	No personal asset ownership or inactivity-triggered transfer.

3	Federated Learning for Privacy (Montasari)	Federated learning without centralizing raw data.	No AI-based inactivity monitoring or nominee workflow.
4	Blockchain Access Control (Schlatt et al.)	Decentralized identity management using blockchain.	No inactivity detection or automated inheritance trigger.
5	Smart Contract Data Ownership (Khademi & Honavar)	Smart contracts enforce data ownership and access rights.	No complete legacy lifecycle from monitoring to transfer.

3. EXISTING SYSTEM AND LIMITATIONS

The most closely related existing system is AICyberChain (Zia Ullah et al., IEEE Access, 2024), which combines AI-powered threat detection with blockchain-based secure data sharing and smart contract enforcement on the Ethereum network. While AICyberChain represents a meaningful technical contribution, its design scope is fundamentally limited to external cybersecurity applications.

A detailed analysis reveals the following critical limitations with respect to digital legacy management: (1) No Personal Asset Vault — no mechanism for users to store and categorize personal digital assets for future inheritance; (2) No Nominee Management — no concept of designated beneficiaries or user-defined conditional access rules for authorized heirs; (3) No Inactivity-Based Triggering — all smart contract executions are triggered by cybersecurity threats, not user inactivity; (4) No Decentralized File Storage — no IPFS layer, meaning stored data is subject to centralized failure risks; (5) No Posthumous Access Mechanism — no framework for controlled data access after the owner's death or incapacitation.

4. PROPOSED SYSTEM METHODOLOGY

The proposed ALPG framework employs a seven-phase methodology spanning the complete digital legacy lifecycle, from initial user onboarding to final nominee asset retrieval. Each phase is independently verifiable, cryptographically secure, and integrated with adjacent phases through well-defined API interfaces.

Phase 1: User Registration and Secure Authentication

New users register by providing their full name, email address, and a chosen password. Passwords are immediately hashed using bcrypt (cost factor 12) prior to database storage. Login sessions are managed using JSON Web Tokens (JWT) with configurable expiry periods. An OTP-based two-factor verification step is enforced at login, with a six-digit code delivered to the user's registered email via SMTP and valid for ten minutes.

Phase 2: Digital Asset Upload and Encrypted Storage

Authenticated users upload digital assets through the secure asset management interface. Uploaded files are encrypted on the server using AES-256 symmetric encryption before being transmitted to IPFS nodes via the IPFS HTTP API. IPFS returns a unique content identifier (CID) for each stored file. The CID, along with file metadata, is stored in MongoDB. A SHA-256

hash of the original file content is computed and stored alongside the CID for integrity verification.

Phase 3: Nominee Assignment and Smart Contract Configuration

Users designate one or more nominees through the nominee management interface. Nominee records undergo an email verification step. Once verified, the user configures smart contract parameters for each nominee — specifying the inactivity threshold (in days) that should trigger asset access, and the specific assets the nominee is authorized to access. These parameters are encoded into an Ethereum smart contract deployed by the system on behalf of the user.

Phase 4: AI Activity Monitoring and Behavioral Analysis

The AI monitoring engine, implemented as a Python-based microservice, runs on a scheduled 24-hour interval. It retrieves the last_login timestamp for each registered user from MongoDB and computes the elapsed inactivity period. An exponential moving average is applied to each user's historical login interval data to establish a personalized activity baseline. A dual-threshold detection model classifies users as: Active, At-Risk (approaching the configured inactivity threshold), or Inactive (exceeding the threshold).

Phase 5: Smart Contract Condition Evaluation and Execution

When the AI monitoring engine flags a user as Inactive, the system dispatches a verification signal to the Ethereum smart contract associated with that user's account. The smart contract autonomously evaluates all encoded conditions. If satisfied, the smart contract executes the access transfer transaction, recording the event immutably on the Ethereum blockchain. The transaction hash, timestamp, and nominee identifiers are all stored permanently on-chain, providing a legally verifiable audit trail.

Phase 6: Nominee Notification and Secure Data Retrieval

Upon successful smart contract execution, the Node.js backend detects the contract event via Ethereum event listeners and triggers the notification workflow. Each eligible nominee receives an email notification containing a secure, time-limited access token (valid for 72 hours by default). This token grants the nominee access to a secure retrieval interface where they can download their designated encrypted assets from IPFS, decrypted on the fly using the user-specific AES-256 key released by the smart contract.

Phase 7: Audit Logging and Compliance

Every significant system event — including user logins, asset uploads, nominee verifications, smart contract deployments, contract executions, and asset retrievals — is recorded both in MongoDB and on the Ethereum blockchain. The blockchain audit log provides a tamper-proof, chronologically ordered record of the complete digital legacy lifecycle, independently verifiable by nominees, legal representatives, or regulatory authorities.

5. SYSTEM ARCHITECTURE AND MODULE DESIGN

The ALPG system follows a microservices-inspired layered architecture comprising five primary modules interconnected through secure internal APIs and event-driven blockchain listeners. The modular design ensures that each component can be independently developed, tested, scaled, and updated without disrupting the overall system operation.

Module 1: User Management — Authentication and Authorization

This module serves as the secure entry point for all system interactions. It manages user registration, login, profile updates, and session lifecycle. JWT tokens govern session state with configurable expiry. An OTP-based two-factor authentication layer adds a second verification step at every login. Role-Based Access Control (RBAC) enforces a two-tier permission model: Data Owners and Nominees — ensuring that nominees cannot access assets prior to verified contract execution.

Module 2: Asset and Nominee Management

This module handles the complete lifecycle of digital asset storage and nominee configuration. Files uploaded by the user are AES-256 encrypted on the server before transmission to IPFS, ensuring that raw file content is never exposed to the storage layer. Nominee records are subject to mandatory email verification before activation. The module provides a dashboard interface for users to view, update, and audit all stored assets and active nominee configurations.

Module 3: AI Monitoring and Inactivity Detection

The AI monitoring engine is a Python Flask microservice that runs on a 24-hour schedule. It retrieves user activity data from MongoDB, computes inactivity scores using exponential moving averages, and classifies users into activity tiers. The dual-threshold model applies a configurable absolute inactivity threshold (default: 90 days) and a 1.5x personal baseline multiplier to reduce false positives from inherently infrequent users.

Module 4: Blockchain Smart Contract Execution

This module manages the deployment and execution of Ethereum smart contracts that govern digital asset inheritance. Each user's contract is uniquely parameterized with their configured inactivity threshold, nominee wallet addresses, IPFS CIDs of designated assets, and AES decryption key references. The contract enforces a strict verification protocol: only nominees who completed email verification and whose wallet addresses match the contract's encoded nominees are eligible for execution.

Module 5: Nominee Notification and Data Retrieval

Upon detecting a successful contract execution event, this module initiates the nominee notification workflow. Verified nominees receive email notifications containing time-limited secure access tokens. The retrieval interface authenticates nominees using these tokens, retrieves encrypted files from IPFS using the CIDs released by the smart contract, decrypts the files, and delivers them through a secure download interface.

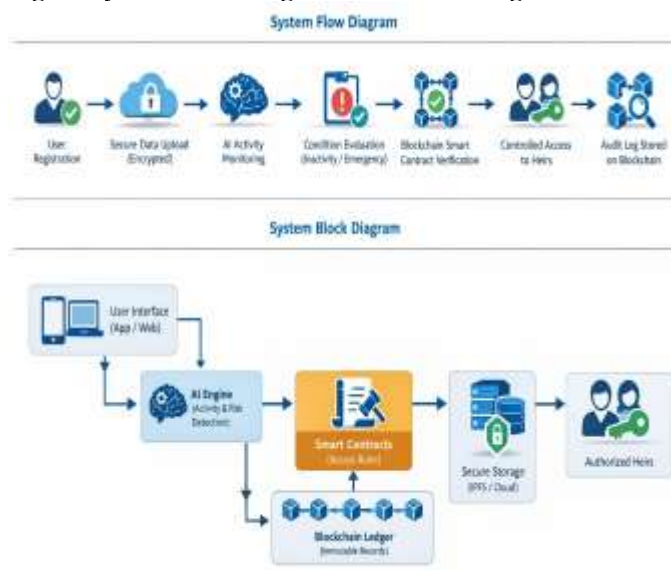
Table -2: Module Summary

Module	Name	Key Functions
Module 1	User Management	Registration, login, JWT session management, bcrypt hashing, RBAC enforcement
Module 2	Asset and Nominee Management	IPFS encrypted file storage, nominee assignment, access rule definition
Module 3	AI Monitoring and Inactivity Detection	Login pattern analysis, behavioral scoring, inactivity threshold triggering
Module 4	Blockchain Smart Contract Execution	Ethereum contract verification, nominee eligibility check, immutable logging
Module 5	Nominee Notification and Data Retrieval	Email alerts, time-limited secure access tokens, encrypted IPFS retrieval

6. SYSTEM FLOW AND ARCHITECTURE DESCRIPTION

The complete system workflow proceeds through six sequential stages. The system flow diagram (Fig. 1) illustrates the end-to-end operational sequence from user registration to nominee asset retrieval, while the system block diagram depicts the internal component architecture and inter-module communication pathways.

Fig -1: System Flow Diagram and Block Diagram



Stage 1: User Registration and Secure Asset Upload

The user accesses the platform through the web interface and completes registration with email and bcrypt-hashed password. Following OTP-based two-factor authentication, the user uploads digital assets through the asset management dashboard. Uploaded files are encrypted using AES-256 on the

server and transmitted to IPFS nodes, with the resulting CID and SHA-256 content hash stored in MongoDB. This stage establishes the user's digital vault and records the initial activity timestamp for the AI monitoring baseline.

Stage 2: AI Activity Monitoring

The AI monitoring microservice continuously tracks user login events and computes personalized inactivity scores. The service classifies each user into Active, At-Risk, or Inactive status based on the dual-threshold model. For At-Risk users, warning emails are dispatched to prompt a login event. The AI engine operates transparently in the background without disrupting the user experience during normal platform usage.

Stage 3: Condition Evaluation and Smart Contract Triggering

When the AI engine classifies a user as Inactive, it dispatches a condition evaluation signal to the Ethereum smart contract associated with that user's account. The smart contract independently verifies the inactivity duration, nominee verification status, and all additional user-configured conditions. This independent verification ensures that the transfer decision is not reliant on the application server alone, providing a trustless execution guarantee.

Stage 4: Blockchain Smart Contract Verification and Execution

Upon condition satisfaction, the Ethereum smart contract executes the access transfer transaction. It emits a structured blockchain event containing the nominee identifiers, transferred asset CIDs, and decryption key references. This event is permanently recorded on the Ethereum ledger, establishing an immutable, time-stamped proof of the inheritance transfer that can be independently verified by any party with access to the blockchain.

Stage 5: Controlled Nominee Access

The Node.js backend detects the smart contract execution event through Ethereum event listeners. It generates time-limited, cryptographically signed access tokens for each eligible nominee and dispatches notification emails with secure retrieval links. Nominees access the retrieval portal, authenticate using their tokens, and download their designated assets decrypted and streamed through HTTPS.

Stage 6: Audit Log Stored on Blockchain

Every significant event — including contract deployment, inactivity detection, condition evaluation, contract execution, and nominee retrieval — is recorded both in the application's MongoDB audit collection and on the Ethereum blockchain. The blockchain audit log is immutable and independently verifiable, providing a legally credible record of the complete digital legacy transfer process.

Fig -2: LegacyGuardian Dashboard - User Interface



7. CONCLUSIONS

This paper presented the AI Legacy and Privacy Guardian (ALPG), a novel framework that combines artificial intelligence, Ethereum blockchain smart contracts, IPFS decentralized storage, and multi-layer security mechanisms to provide a comprehensive, automated, and trustworthy solution for digital legacy and privacy management. The proposed system directly addresses the critical limitations of existing approaches — including their inability to act autonomously upon user inactivity, their dependence on centralized authorities, and their lack of purpose-built inheritance workflows — by integrating five tightly coupled modules that collectively manage the complete digital legacy lifecycle.

The AI monitoring engine provides proactive inactivity detection using behavioral baseline analysis, reducing false positives while ensuring timely nominee notification. Ethereum smart contracts provide a trustless, transparent, and immutable execution layer that eliminates dependence on any single party for inheritance decisions. IPFS ensures that digital assets remain accessible and censorship-resistant across distributed storage nodes.

Future work will focus on: (1) enhancing the AI monitoring module with deep learning models; (2) adding biometric authentication support; (3) developing a mobile application with push notification support; (4) evaluating Ethereum layer-2 scaling solutions to reduce gas costs; and (5) conducting a formal legal and regulatory compliance review for compatibility with GDPR and the Digital Assets Act.

ACKNOWLEDGEMENT

The authors sincerely thank the Department of AI and Data Science, United Institute of Technology, Coimbatore, for providing the necessary resources, infrastructure, and academic support throughout the development of this research work. Special gratitude is extended to Ms. Saranya G (AP/AI&DS) for her invaluable mentorship, continuous guidance, and encouragement during the preparation of this manuscript.

REFERENCES

- [1] C. Chen et al., "A Traceable Online Will System Based on Blockchain and Smart Contract Technology," *Symmetry*, vol. 13, no. 6, pp. 1-19, 2021.
- [2] Z. Zhang et al., "Decentralized Digital Inheritance Management System Using Blockchain," *Preprints.org*, 2025. DOI: 10.20944/preprints202501.0001.v1.

- [3] J. Goldston et al., "Digital Inheritance in Web3: A Case Study," *arXiv preprint arXiv:2304.11293*, 2023.
- [4] M. Alharby and A. van Moorsel, "Blockchain-Based Smart Contracts: A Systematic Mapping Study," *arXiv preprint arXiv:1710.06372*, 2017.
- [5] Y. Lei et al., "AI Afterlife as Digital Legacy: Perceptions and Concerns," *arXiv preprint arXiv:2501.12345*, 2025.
- [6] Z. Ullah et al., "AICyberChain: Combining AI and Blockchain for Improved Cybersecurity," *IEEE Access*, vol. 12, pp. 45321-45338, 2024.
- [7] Xu et al., "Blockchain-Based Secure Data Sharing for Healthcare," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1234-1245, 2022.
- [8] Montasari, "Federated Learning for Privacy-Preserving Data Sharing," *IEEE Security and Privacy*, vol. 20, no. 4, pp. 55-63, 2022.
- [9] V. Schlatt et al., "Blockchain-Based Access Control System," *IEEE Internet Computing*, vol. 26, no. 3, pp. 34-41, 2022.
- [10] A. Khademi and V. Honavar, "Smart Contract-Based Data Ownership Model," in *Proc. IEEE International Conference on Blockchain*, pp. 112-119, 2021.