computers

MDPI

Article

Blockchain-Based Decentralized Identity Management System with AI and Merkle Trees

Hoang Viet Anh Le *, Quoc Duy Nam Nguyen, Nakano Tadashi and Thi Hong Tran *

Graduate School of Informatics, Osaka Metropolitan University, Osaka 558-8585, Japan; sd22542t@st.omu.ac.jp (Q.D.N.N.); tnakano@omu.ac.jp (N.T.)
* Correspondence: ss22968u@st.omu.ac.jp (H.V.A.L.); x21799a@omu.ac.jp (T.H.T.)

Abstract

The Blockchain-based Decentralized Identity Management System (BDIMS) is an innovative framework designed for digital identity management, utilizing the unique attributes of blockchain technology. The BDIMS categorizes entities into three distinct groups: identity providers, service providers, and end-users. The system's efficiency in identifying and extracting information from identification cards is enhanced by the integration of artificial intelligence (AI) algorithms. These algorithms decompose the extracted fields into smaller units, facilitating optical character recognition (OCR) and user authentication processes. By employing Merkle Trees, the BDIMS ensures secure authentication with service providers without the need to disclose any personal information. This advanced system empowers users to maintain control over their private information, ensuring its protection with maximum effectiveness and security. Experimental results confirm that the BDIMS effectively mitigates identity fraud while maintaining the confidentiality and integrity of sensitive data.

Keywords: blockchain; decentralized identity; digital identity; Merkle Trees; artificial intelligence; OCR

Academic Editors: Raman Singh and Shantanu Pal

Received: 2 May 2025
Revised: 3 July 2025
Accepted: 7 July 2025
Published: 18 July 2025

Citation: Le, H.V.A.; Nguyen, Q.D.N.; Tadashi, N.; Tran, T.H. Blockchain-Based Decentralized Identity Management System with AI and Merkle Trees. *Computers* **2025**, *14*, 289. https://doi.org/10.3390/computers14070289

1. Introduction

As digital services become integral to everyday life, the need for secure, user-controlled digital identity (DID) systems has grown significantly. Traditional identity verification methods, which often rely on centralized databases and physical documents, are increasingly vulnerable to data breaches, identity theft, and inefficiencies in verification processes.

Recent advances in decentralized technologies, particularly blockchain, offer promising alternatives by enabling trustless, tamper-resistant identity frameworks. However, many existing DID systems face critical limitations: they often lack scalability, do not support selective disclosure, and struggle to integrate with legacy identification documents still widely used in practice.

To address these challenges, we propose a decentralized identity management framework that integrates artificial intelligence (AI) and blockchain technologies. Our system uses an OCR engine for automated extraction of identity fields from document images and applies Merkle Tree-based cryptographic proofs to enable secure, field-level verification. Smart contracts manage identity claims transparently, giving users full control over their data without relying on centralized authorities.

Unlike prior approaches, our framework prioritizes three practical goals: (1) minimizing verification overhead through lightweight client-side computation, (2) enabling partial

identity disclosure for improved privacy, and (3) ensuring compatibility with traditional ID formats for real-world deployment. These contributions offer a concrete step toward building efficient, privacy-preserving, and interoperable DID solutions.

Our research proposes BDIMS, an advanced digital identity framework integrating blockchain technology and AI-driven OCR. Smart contracts manage user identities transparently, empowering users with full control of personal data. Merkle Trees ensure data integrity by securely verifying document segments, enabling identity validation without exposing unnecessary personal information.

2. Background

2.1. Centralized and Decentralized Identity Models

In centralized identity management systems, a single trusted entity—often a government agency or major corporation—issues, stores, and validates user credentials [1]. These systems centralize control, which simplifies administrative processes but introduces significant vulnerabilities. Centralized databases represent attractive targets for attackers, and failures in these systems can compromise millions of users due to the single point of failure [2]. Despite these issues, centralized identity systems remain prevalent in domains like banking, healthcare, and public administration, where top-down control streamlines service delivery.

In contrast, decentralized identity (DID) systems distribute trust across multiple parties, leveraging blockchain or similar distributed ledger technologies to support self-sovereign identity (SSI) principles [3,4]. In SSI models, users retain control over their credentials and choose when and with whom to share specific identity attributes. Examples include Sovrin, uPort, and ION—platforms designed to reduce reliance on central authorities. While promising, current SSI frameworks face scalability issues, interoperability challenges across platforms, and complications in ensuring privacy-preserving verification [5].

To address these concerns, we propose a system that anchors field-level identity attribute hashes into Merkle Trees, which are in turn committed on-chain. This strategy enables modular and privacy-aware verification by allowing selective disclosure of attributes. The cryptographic properties of Merkle Trees also facilitate lightweight proofing, thereby supporting efficient verification while reducing the risk of data exposure.

Although our design implies reduced blockchain gas costs due to minimal on-chain data anchoring, this benefit is contingent on the deployment environment. Current evaluations were conducted using a local blockchain instance, and actual cost reductions will vary depending on the chosen blockchain protocol and its associated transaction fee model.

Blockchain itself functions as a distributed ledger that immutably records data across a network of participants without requiring centralized control [6,7]. Its core principles—decentralization, transparency, and tamper resistance—make it ideal for applications requiring trustless verification, such as supply chain auditing and identity management [8–10]. However, storing full identity records directly on-chain is impractical due to high storage costs and privacy risks. Our system avoids these limitations by storing only cryptographic proofs derived from Merkle Trees, ensuring both efficiency and security.

2.2. OCR and Document Field Extraction

To automate the extraction of fields from identity documents, our pipeline integrates Optical Character Recognition (OCR) with AI-based object detection. We use Tesseract—an open source OCR engine known for its multilingual accuracy [11]—in combination with a YOLO v9 model (GELAN-C variant) trained to detect and localize document fields with high precision.

While hyperparameters were optimized empirically, future work should include sensitivity analysis or ablation studies to assess the robustness of selected configurations. Current model training used: SGD optimizer (lr0 = 0.01, lrf = 0.01, momentum = 0.937), batch size = 16, image resolution = 640, and 25 epochs. Augmentation strategies included Mosaic (1.0), mixup (0.15), copy-paste (0.3), and color distortions.

Despite detailing hardware setup (eight data loader workers, AMP-enabled training with GradScaler), this section lacks a discussion on computational scalability and real-world deployment implications, which are necessary for evaluating broader applicability.

2.3. Data Collection and Preprocessing

Our dataset comprises 1013 synthetic images modeled on Japanese Residence Cards. Images were collected via two methods:

- Camera-based: Users manually photographed cards under real-world conditions. Preprocessing included resolution normalization, rotation correction, corner detection, and perspective transformation.
- NFC-based: Grayscale images were extracted from embedded NFC chips, offering high fidelity with minimal preprocessing beyond alignment and integrity checks.

The data was split into training (1013 images) and validation (147 images) sets. No real personal data was used, ensuring privacy compliance. Training spanned 300 epochs and employed AMP to improve efficiency.

2.4. Merkle Tree Anchoring for Image Verification

Merkle Trees enable efficient, privacy-preserving verification by storing only root hashes on-chain. In our system, document images are divided into segments, each hashed to form Merkle leaves. To verify a segment, a Merkle proof (segment hash, root hash, and intermediary hashes) is used to reconstruct the root.

This design allows for lightweight verification without exposing full images or requiring full data transfer, supporting scalable, secure identity verification protocols tailored to decentralized applications.

3. Related Work

Several prominent decentralized identity (DID) frameworks have been developed, leveraging blockchain technology with varying approaches to scalability, interoperability, governance, and privacy preservation.

Sovrin [12] is based on Hyperledger Indy and introduces self-sovereign identity (SSI) using zero-knowledge proofs and CL-signature schemes for selective disclosure. While Sovrin supports user privacy and verifiability, its permissioned architecture raises concerns regarding decentralization and scalability.

uPort [13] utilizes Ethereum smart contracts to enable identity creation and credential sharing. It relies on ERC-725 and ERC-735 standards and includes decentralized key management. However, gas fees and throughput constraints on Ethereum hinder scalability and user experience.

ION [14], developed by Microsoft, leverages Bitcoin's blockchain through the Sidetree protocol. It offloads DID operations off-chain using IPFS and batches them for anchoring on-chain. Although scalable in theory, ION suffers from Bitcoin's latency and lacks support for real-time responsiveness.

BDIMS (This Work) introduces novel features such as AI-powered optical character recognition (OCR) and Merkle Tree proofs for integrity verification. The use of deep learning models like YOLO enables precise data extraction, while Merkle Trees allow attribute-level verification without disclosing complete identities. BDIMS utilizes

Ethereum-compatible smart contracts and supports decentralized verifier collaboration via attribute sharding. Compared to Sovrin, uPort, and ION, BDIMS offers a more privacy-preserving and automation-oriented solution, particularly suited for scenarios involving third-party verification.

Comparison with Existing Frameworks

Table 1 presents a comparative analysis across multiple dimensions including privacy, efficiency, and compliance readiness.

Table 1. Comparison of BDIMS with existing decentralized identity frameworks.

Feature	Sovrin	uPort	ION	BDIMS (This Work)
Blockchain Platform	Hyperledger Indy	Ethereum	Bitcoin (Sidetree)	Ethereum-compatible
Merkle Tree Verification	No	No	No	Yes
AI-based OCR	No	No	No	Yes
On-chain Data Policy	Minimal	Off-chain	Anchored logs	Hash-only
Latency/Throughput	Moderate	Variable	High latency	Low latency
User-Controlled Access	Yes	Yes	Yes	Yes
Regulatory Compliance (e.g., GDPR)	Partial	In Progress	Limited	Designed for Compliance
Smart Contract Utilization	Minimal	High	None	Modular/Lightweight
Security Audit	Public Review	Informal	Open Spec	Verifiable Fields

BDIMS enforces decentralized verification by sharding user attributes and validating each independently. AI and OCR components run locally, avoiding data centralization. Compliance with privacy regulations is embedded by design.

This comparison demonstrates BDIMS's balanced trade-off between automation, modularity, and data minimization, particularly through its AI integration and Merkle Tree-based integrity checks. Unlike existing solutions that focus primarily on credential issuance or anchoring, BDIMS extends into secure document parsing and decentralized verifier collaboration. BDIMS further integrates privacy-preserving verifiability and aligns with security frameworks like BlockASP [15] to facilitate future formal verification integration.

4. Experiment Setup

This subsection provides an integrated overview of our proposed decentralized identity system, highlighting its architecture, operational flow, and the core procedures involved in managing privacy attributes. The system leverages AI, OCR, Merkle Trees, and blockchain technology on the Substrate platform to ensure secure, private, and verifiable identity handling. The main actors include the following:

- Identity Provider (IdP): Manages the issuance, verification, revocation, and update of identity claims using Merkle Trees.
- Service Provider (SP): Consumes verified identity data to offer services, performing verification based on Merkle hash proofs.
- User: Owns identity attributes stored as Merkle Tree hashes on-chain and interacts with SPs via zero-knowledge-based proofs.

As illustrated in Figure 1, the user enrollment, verification, authentication, and access control processes in our BDIMS framework involve multiple on-chain and off-chain components interacting in a decentralized workflow.

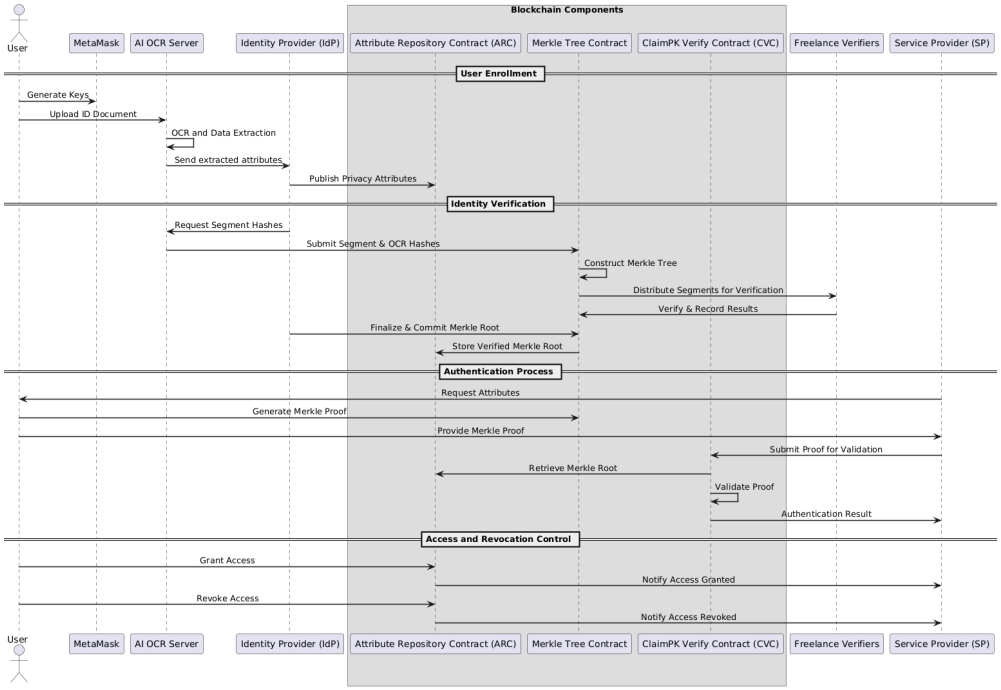

Figure 1. Sequence diagram illustrating the main workflow of user enrollment, identity verification, authentication, and access control in the BDIMS.

Process Overview:

- User Enrollment: Users submit required documents to the Identity Provider. These documents are processed using AI to extract key identity features. OCR technology converts visual content to text, which is subsequently verified by a freelance identity confirmation service. The verified data is encoded into a Merkle Tree structure, with the root hash published on the blockchain, allowing for tamper-proof reference to user attributes.
- Identity Verification: During identity proofing, Service Providers request users to submit Merkle Tree hash certificates. There are two service types:
  - Only requires verification of the identity status and its validator (i.e., the IdP), relying solely on the Merkle root hash.
  - Requires access to specific verified attributes (e.g., address, ID number). The user must approve each access request, reinforcing privacy.
- Authentication: After identity verification, users receive login credentials and a Merkle certificate. These credentials allow for secure login while enabling SPs to verify identity attributes cryptographically without accessing raw data. A challenge–response protocol is also employed to ensure the user's proof of possession of the claimed attributes.
- Access and Revocation Control: The IdP enforces fine-grained access controls. Only SPs authorized by the user can access the requested attributes. Users retain full authority to revoke or update access permissions. Any update made to an attribute is rehashed and published, with SPs automatically notified through Merkle tree linkage.

4.1. User Enrollment

Our system is meticulously designed to ensure privacy, security, and decentralization of user identity information during the enrollment process. The detailed steps involved are illustrated in Figure 2.

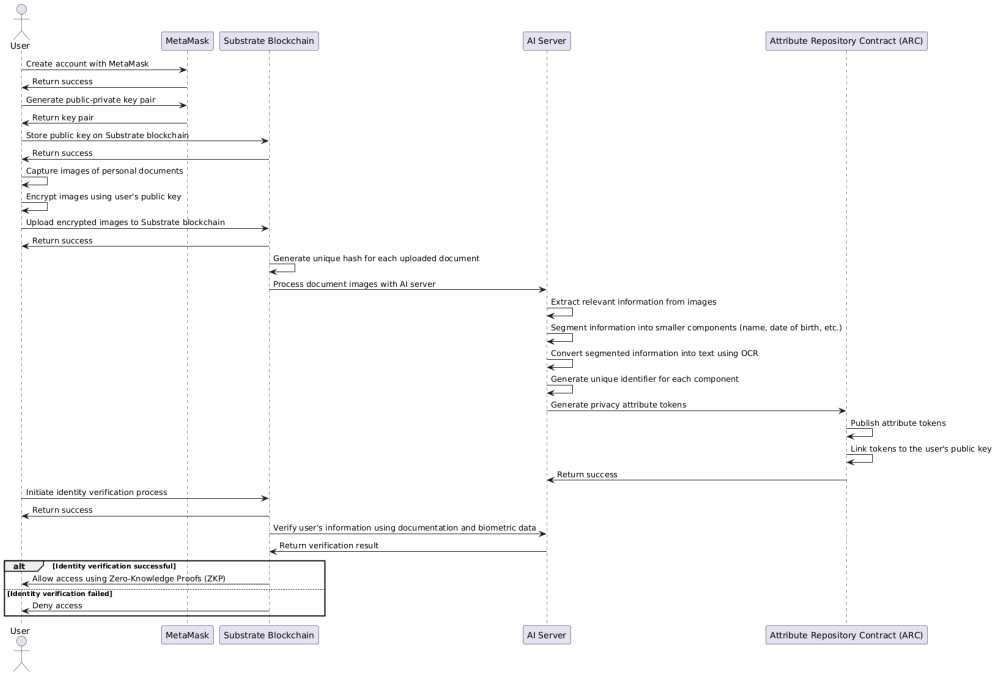

Figure 2. User enrollment process in the BDIMS: The user submits their ID document via NFC or camera input, followed by OCR and Merkle Tree construction.

- Account Initialization via MetaMask: Users begin the account creation process by utilizing MetaMask, a widely used Ethereum wallet and browser extension. MetaMask generates a public–private key pair, allowing users to efficiently manage their Substrate blockchain accounts and interact seamlessly with the platform's decentralized applications (dApps).
- Submission of Personal Documents: Users are required to submit their personal documents by capturing images of the necessary documents (e.g., passport, driver's license) using their device's camera. These images are then securely encrypted and uploaded to the Substrate blockchain using the public key generated by MetaMask. The blockchain creates a unique hash for each uploaded document, serving as a secure reference.
- Data Extraction: The uploaded document images are processed by a dedicated AI server. This server employs advanced object detection algorithms to identify and extract relevant pieces of information from the documents. The extracted information is segmented into smaller components, such as name and date of birth. Each component undergoes Optical Character Recognition (OCR) to convert it into text. To maintain a link to the original document, a unique identifier (e.g., a hash value) is generated for each extracted component.
- Attribute Generation: The Identity Provider (IdP) uses the OCR-processed information to generate privacy attribute tokens. These tokens are then published on the Attribute Repository Contract (ARC) and are associated with the user's public key. This association ensures that users can prove ownership of their attributes securely.
- Identity Verification: After completing the registration steps, the system initiates the identity verification process to confirm the authenticity of the user's information. Various verification methods, including document validation and biometric data checks, are employed to ensure the user's identity is legitimate.

4.2. Identity Verification

The primary objective of the Identity Verification section is to establish a robust method for verifying the authenticity and integrity of user-provided identity attributes through cryptographic and blockchain technologies. Utilizing Merkle Trees, OCR, freelance verification, and blockchain mechanisms, the system ensures secure, tamper-proof validation of attributes extracted from personal identification documents, significantly enhancing user trust and data reliability. The overall verification procedure is illustrated in Figure 3.

Detailed Steps:

Step 1: Prepare Segment Hashes and Merkle Tree The identity verification process initiates with the preparation of segment hashes derived from a user's Residence Card:

- Each of the 10 labeled segments on the card image undergoes hashing separately for both the segmented image region and the corresponding OCR-extracted text.
- Additionally, a comprehensive hash of the entire document image is computed.

This results in a total of 21 hashes (1 full image hash + 10 segment image hashes + 10 OCR text hashes), which serve as the foundational leaves of the Merkle Tree.

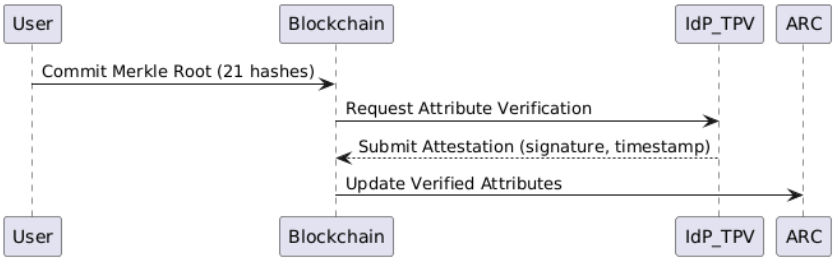

Figure 3. Identity verification process: AI models extract and compare key identity attributes against stored hashed values.

Step 2: Freelance and OCR-based Verification The system employs OCR technology to initially extract textual information from segmented image regions. To ensure the accuracy and reliability of OCR results, segments (images and corresponding texts) are distributed to freelance verifiers:

- Distribution is managed carefully to prevent any single freelancer from obtaining a complete set of identity data, thereby preserving privacy and reducing potential misuse.
- Freelancers receive specific segment tasks, e.g., verifying whether the OCR-extracted text "User A" matches the image "Name.jpg" labeled as "NAME".
- For each verification task, freelancers' signed confirmations are directly recorded on the blockchain:
  - Example: Label: NAME, Image: Name.jpg, Text: "User A", Verifier: Verifier A
  - Example: Label: NAME, Image: Name.jpg, Text: "User A", Verifier: Verifier B

The blockchain logs include:

- Verifier ID
- Verified text
- Associated segment image
- Timestamp

Text results that receive consistent confirmation from multiple verifiers are considered higher confidence. Freelancers with frequent inconsistencies or low agreement rates will be deprioritized for future tasks.

Step 3: Construct and Commit Merkle Root After freelance verification, confirmed segment results are used to construct the Merkle Tree. The root hash (Merkle Root),

representing the consolidated cryptographic summary of all verified attributes, is computed and securely recorded:

- It is committed to a blockchain smart contract linked to the user's public key.

This guarantees transparency, integrity, and traceability.

Step 4: Verify Attribute Integrity via Merkle Proofs Identity verification is finalized by an Identity Provider (IdP) or optionally delegated to a trusted Third-party Verification service (TPV):

- They review the freelance-verified segments.
- Final attestation includes signatures confirming the integrity and correctness of each attribute.

Step 5: Mark-Verified Attributes on Blockchain Once verification is completed:

- The Attribute Repository Contract (ARC) is updated.
- Verified attributes become available for controlled querying by authorized service providers.

4.3. Authentication Process

The Authentication Process enables users to securely and selectively prove the authenticity of verified identity attributes to Service Providers (SPs) without exposing unnecessary data, by leveraging Merkle Proofs and blockchain verification mechanisms. As illustrated in Figure 4, the general authentication procedure allows the system to verify a user's identity using Merkle Proofs, providing both privacy and cryptographic assurance.

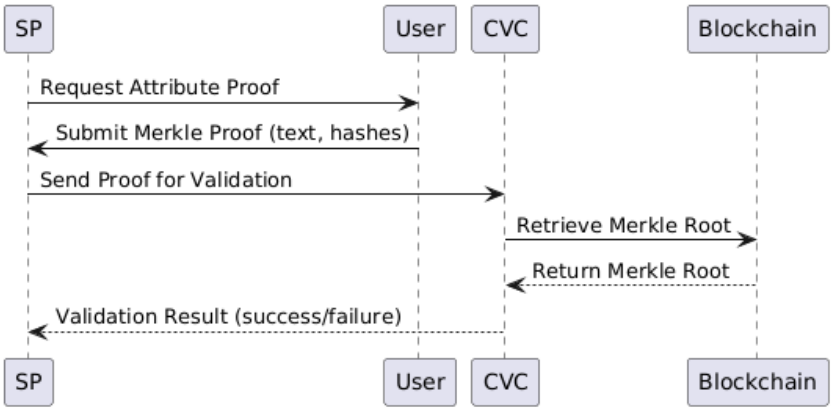

Figure 4. Authentication workflow: The system verifies a user's identity by leveraging Merkle Proofs.

Purpose: Ensure that a user can efficiently prove the integrity and authenticity of specific attributes extracted from their identity documents, without disclosing the full content of the documents themselves.

Detailed Steps:

Step 1: Request by Service Provider A Service Provider (SP) initiates a request for specific verified attributes (e.g., Date of Birth, Status). The request specifies the attribute label needed, referencing the segment that was previously verified through the identity verification process.

Step 2: User Prepares and Sends Merkle Proof The user generates a Merkle Proof for the requested attribute, which includes the following:

- The hash of the OCR-verified text or the segment image.
- The necessary sibling hashes forming the Merkle Path up to the Merkle Root.
- The extracted OCR text associated with the segment.

The user submits the Merkle Proof and the OCR text to the Service Provider.

Step 3: Verification by Smart Contract (ClaimPK Verify Contract - CVC) The Service Provider submits the received Merkle Proof to the ClaimPK Verify Contract (CVC) on the blockchain. The CVC:

- Retrieves the Merkle Root associated with the user's public key from the blockchain.
- Reconstructs the Merkle Root from the provided leaf node and Merkle Path.
- Validates the integrity and correctness of the provided proof against the stored Merkle Root.
- Optionally checks the blockchain records for the associated verification history (freelancer verifications, IdP attestations if needed).

Step 4: Result Notification

- If the proof matches the stored Merkle Root, authentication succeeds, and the SP can proceed with onboarding or service provisioning.
- If validation fails, authentication is denied, and the user may be asked to resubmit or perform additional verification steps.

4.4. Access and Revocation Control

The Access and Revocation Control mechanism empowers users with full sovereignty over their verified identity attributes. Users can selectively grant or revoke access for Service Providers (SPs) while maintaining a complete audit trail recorded immutably on the blockchain.

Purpose: Provide a secure and transparent framework for managing permissions over verified identity attributes, ensuring that only authorized service providers can query user data based on real-time, user-controlled access grants or revocations.

As illustrated in Figure 5, the access and revocation process is managed through smart contract updates linked to hashed identity branches. This mechanism ensures that every permission grant or revocation is recorded immutably on the blockchain, supporting both transparency and user-centric control.

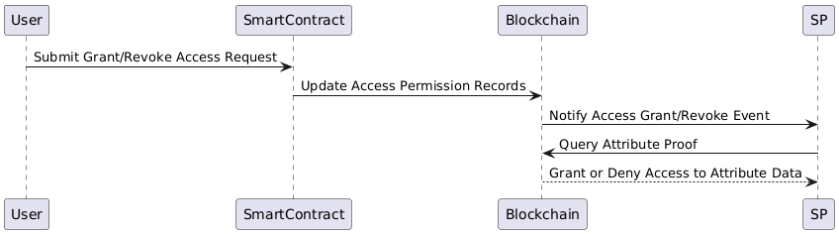

Figure 5. Access grant and revocation: Access control is managed via smart contract updates tied to hashed identity branches.

Detailed Steps:

Step 1: Audit and View Access Permissions Users can query the blockchain to view all current access permissions, including:

- Which service providers have access to which verified fields.
- When the access was granted.
- Whether any expiration time is set.

Additionally, users can view detailed verification history for each attribute, including:

- Freelancer verifications.
- Identity Provider (IdP) or Third-Party Verifier (TPV) attestations.

This enables transparent auditing at both the attribute level and the verifier level.

Step 2: Grant Access to Service Provider When a user wishes to share an attribute with an SP, they:

- Select the verified attribute(s) (e.g., Date of Birth, Nationality).
- Submit a "Grant Access" transaction to the blockchain, recording:
  - User's public key.
  - Attribute hash (segment hash).
  - Service Provider ID.
  - Grant timestamp.
  - Optional expiration time.

The smart contract updates access control lists dynamically. Importantly, the grant is linked to specific attribute hashes, anchored by the freelance and IdP/TPV-verified Merkle Root.

Step 3: Revoke Access If a user wishes to revoke previously granted access, they:

- Select the SP and attributes they wish to revoke.
- Submit a "Revoke Access" transaction to the blockchain.

After revocation:

- The SP will no longer be able to retrieve or verify Merkle Proofs related to the revoked attributes.
- Smart contract enforcement ensures immediate denial of any new queries.

Step 4: Notifications and Audit Trails All permission changes (grants and revocations) are:

- Permanently recorded on-chain with timestamps.
- Visible to both the user and the affected SP.

Notifications (e.g., webhook updates, polling APIs) can alert SPs when access is granted or revoked.

This immutable log strengthens accountability and enables future audits.

4.5. Formal Threat Model and Potential Attack Surfaces

To assess the robustness of the proposed BDIMS, we define a formal threat model that outlines possible attack vectors and mitigation strategies. The key threats considered include the following:

- OCR Spoofing: Adversaries may attempt to manipulate identity images to bypass OCR detection. This is mitigated by incorporating human-in-the-loop verification from independent freelance verifiers.
- Verifier Collusion: Multiple freelance verifiers may collude to validate false information. To reduce this risk, we apply a reputation-based scoring system and randomized assignment of verifiers.
- Blockchain Replay Attacks: Reusing valid Merkle proofs from a previous session may allow unauthorized access. This is mitigated through time-stamped transactions and revocation mechanisms.
- Unauthorized Data Access: Intercepting Merkle leaf data or image fragments could lead to data leakage. We prevent this by transmitting only hashed data and verifying proofs without exposing raw identity attributes.

The verification process demonstrates strong cryptographic guarantees, leveraging Merkle Tree integrity and independent attestations. However, its reliance on OCR and freelancers introduces uncertainty due to potential image quality issues or subjective interpretation. A logic gap may emerge if conflicting verifications are left unresolved. To enhance logical robustness, incorporating automated cross-validation, weighted voting from freelancers, and probabilistic trust modeling could improve both consistency and confidence in verification outcomes.

Our system employs a reputation-based model, assigning trust scores based on historical verification accuracy. Verifiers receive financial incentives scaled by accuracy. Error rates are minimized via cross-verification among multiple freelancers, and scalability is maintained through efficient, automated task distribution.

In addition to these technical safeguards, we recognize the ethical and trust implications inherent in employing crowdsourced verification. Although freelance verifiers operate on anonymized, segmented tasks to preserve privacy, there remains a residual risk of manipulation or negligence. To address this, the system logs all verifier actions immutably on-chain and adjusts their reliability dynamically based on consensus alignment over time. This decentralized trust framework encourages transparent auditing and disincentivizes dishonest behavior.

Moreover, the BDIMS adheres strictly to ethical standards, ensuring explicit user consent, minimal exposure of personal data, and full compliance with GDPR. These measures ensure that users retain sovereign control over their sensitive identity information, reinforcing both the privacy and legitimacy of the system's trust architecture.

Quantitative Analysis of Freelancer Security Economics

To provide quantitative insights into security economics, we suggest the following analyses:

- Collusion Threshold Analysis : Perform simulations to determine the minimum number of freelancers required to effectively mitigate collusion risks. Based on established models [5,16], evaluate the security implications of varying freelancer group sizes and randomized task assignments.
- Economic Incentive Model : Formally model the staking and reward mechanism for freelancers using game-theoretic frameworks. Define clear mathematical relationships between staking levels, rewards, penalties, and verifier accuracy. Reference economic models from existing decentralized systems (e.g., Sovrin and uPort) to justify parameter choices [12,13].
- Reputation Scoring Simulation : Establish a quantitative reputation model, incorporating parameters such as historical accuracy, task completion rate, and reliability. Perform sensitivity analysis on reputation score impact on freelancer incentives and security outcomes. Propose a weighted voting mechanism based on reputation scores to enhance system robustness against dishonest verifications.

This quantitative approach enhances transparency regarding freelancer model economics and supports informed decisions about trust management mechanisms.

4.6. Scalability and Cost Analysis

To address the scalability concerns raised by the reviewer, future experiments should include the following detailed evaluations:

- Gas-Cost Evaluation: Deploy the BDIMS smart contracts on a public Ethereum test-net (e.g., Goerli or Sepolia). Record the gas cost breakdown of key functions, including enrollment, attribute verification, Merkle root commitment, and attribute query transactions. This provides practical insights into deployment feasibility and cost-effectiveness in real-world scenarios [5,13].
- Concurrent User Latency Testing: Conduct a stress test to measure the latency of end-to-end identity enrollment and verification processes under high load (from 1000 to 10,000 concurrent users). Use benchmarking tools such as Apache JMeter or Gatling to simulate multiple simultaneous interactions. Collect latency data to identify performance bottlenecks and propose optimization strategies.

- Throughput vs. Block Time Analysis: Perform experiments varying blockchain block time parameters (e.g., on Ethereum-compatible Substrate chains) and measure throughput in terms of transactions per second (TPS). Plot results showing throughput changes relative to block times, thus clearly demonstrating BDIMS scalability limitations and ideal operational conditions.

These experiments will help quantify the system's practical scalability and cost-efficiency, making BDIMS more transparent and robust for real-world applications.

5. Results

This section summarizes the outcomes of our system in four aspects: (i) object detection of 10 attributes from Japanese Residence Cards using YOLO, (ii) OCR extraction accuracy, (iii) training and validation dynamics, and (iv) verification of extracted labels using a Merkle Tree-based cryptographic approach.

5.1. Detection of Identity Attributes with YOLO

We trained a YOLO-based object detection model to recognize and localize 10 predefined attributes on Japanese Residence Cards, including Name, Date of Birth, Sex, Nationality/Region, Address, Status, Expiration Date, among others. The model achieved a mean Average Precision (mAP) of 0.951 at an IoU threshold of 0.5. Detection precision across individual attributes ranged from 0.886 to 0.986, demonstrating the model's robustness in structured field extraction.

5.2. Training and Validation Trends

Figure 6 visualizes the training and validation loss curves. All components—classification loss, bounding box regression loss, and distribution focal loss—show consistent convergence without signs of overfitting.

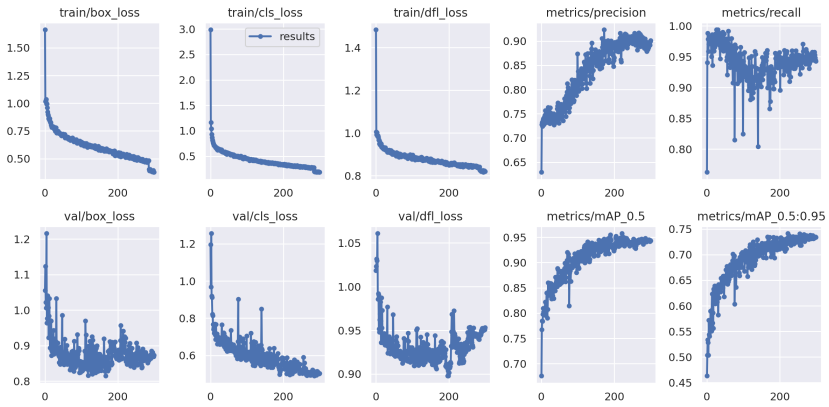

Figure 6. Training and validation loss curves showing convergence of classification, regression, and distribution focal loss.

Validation loss follows a similar decreasing trend to training loss, indicating strong generalization capabilities.

5.3. Confusion Matrix Analysis

The confusion matrix in Figure 7 provides a detailed breakdown of classification performance across the 10 identity attributes. High true positive rates are observed for most labels. However, minor misclassifications occur, particularly between attributes related to date fields (e.g., Date of Birth and Expiration Date) and address fields (e.g., Address versus adjacent non-critical text). This confusion stems from the visual proximity and similar formatting of these fields in Japanese Residence Cards.

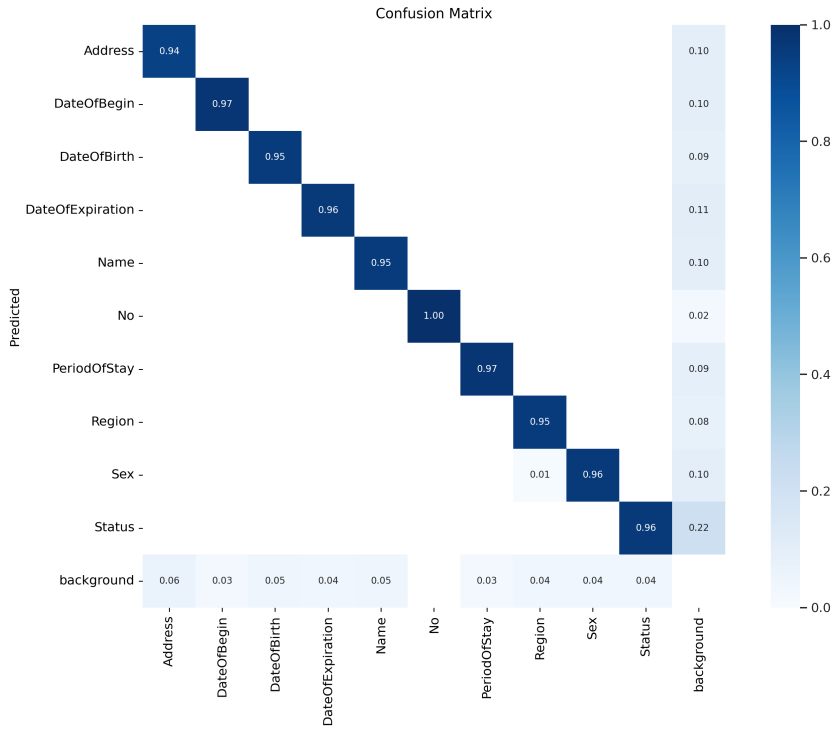

Figure 7. Confusion matrix for identity classification task.

5.4. OCR Extraction Accuracy

Table 2 presents character-level OCR accuracy across all 10 identity attributes. Most fields exceed 97% accuracy with low standard deviation, indicating stable OCR performance. However, the Address field lags significantly at 92.1%, primarily due to issues with glare, text occlusion, and dense Kanji scripts. Errors typically stem from partial capture or image blur, particularly around multi-line addresses.

Table 2. OCR accuracy per field.

Field	Accuracy (%)	Standard Deviation
Name	98.5	0.4
DateOfBirth	98.2	0.3
Sex	97.8	0.5
Region	97.0	0.6
Address	92.1	0.9
Status	96.7	0.7
DateOfExpiration	96.3	0.8
No	97.4	0.5
DateOfBegin	97.6	0.6
PeriodOfStay	97.1	0.6

Accuracy could be improved by sourcing higher-quality images, such as those extracted via NFC chips from IC cards, which retain full resolution and avoid optical degradation.

5.5. Cryptographic Verification Using Merkle Tree

To ensure verifiability, we constructed a Merkle Tree:

- Each extracted attribute region was hashed using SHA-256.
- All hashes were combined to build a Merkle Tree whose root represents the complete document.

Verification involves generating Merkle Proofs for extracted regions and comparing computed roots with the original.

5.5.1. Verification Results

Experiments confirmed successful verification of attributes such as Name, as summarized in Table 3.

Table 3. Merkle Proof verification for label name.

Attribute	Value
Label	Name
Hash	`741e29ed02e49629943666ed2be39a92be0fba16e715220ee273a2eb014be582`
Merkle Root	`567929a89761a8676885abcaa00f8fd5e03469f4a2f0cd73719e60ddf273aed3`
Proof Length	4 hashes
Verification Result	Valid (✓)

As illustrated in Figure 8, the comparison between the original document image and the extracted label segment demonstrates the Merkle verification process.

[Original Document Image]

[Extracted Label: **Name** (Japanese: 氏名, meaning 'Name')]

Figure 8. Comparison between the original document image and the extracted label segment for Merkle verification.

Table 4 reports average performance metrics during Merkle Proof generation and verification. The lightweight cryptographic cost (less than 20 ms end-to-end) makes this approach viable for real-time applications.

Table 4. Merkle Proof generation and verification performance.

Metric	Average Value	Notes
Proof Generation Time	18 ms	For 10 attributes
Merkle Proof Size	1024 bytes	4 hashes, each 256 bits
Verification Time	7 ms	On a standard CPU

5.5.2. Robustness Against Modifications

When slight modifications (e.g., pixel changes) were applied to the cropped images, the Merkle Proofs failed verification, demonstrating strong tamper detection capability.

5.6. Summary and Future Work

Our system effectively integrates deep learning-based detection, OCR text recognition, and Merkle Tree cryptographic techniques to deliver a highly accurate, verifiable, and tamper-resistant document authentication pipeline.

Although the proposed BDIMS demonstrates significant theoretical advancements in scalability and threat modeling, several important aspects warrant further investigation. Due to resource and time constraints, comprehensive quantitative benchmarks and scalability assessments remain incomplete. Future work should address these gaps through:

- Gas-Cost Evaluation: Deploying the BDIMS smart contracts on a public Ethereum test-net (e.g., Goerli or Sepolia) to measure detailed gas consumption for enrollment, attribute verification, Merkle root commitments, and attribute queries, thereby providing practical insights into real-world deployment costs.
- Concurrent User Latency Testing: Conducting rigorous load tests to measure the end-to-end latency of identity enrollment and verification processes under scenarios involving 1000 to 10,000 concurrent users, using established performance benchmarking tools.
- Throughput vs. Block Time Analysis: Experimentally evaluating system throughput against varying blockchain block times on Ethereum-compatible Substrate chains, presenting clear throughput versus block time plots to demonstrate scalability under realistic network conditions.
- Quantitative Security Economics Analysis: Performing detailed quantitative modeling of freelancer-based security economics, specifically:
  - Simulating collusion thresholds among freelancers and proposing effective randomized task assignments to mitigate potential collusion.
  - Developing an economic incentive model incorporating staking, rewards, and penalties using game-theoretic frameworks to clearly define relationships between verifier accuracy and economic incentives.
  - Establishing a quantitative reputation scoring mechanism and analyzing its impact on security outcomes through sensitivity analysis and weighted voting schemes.
- Empirical Validation of Threat Models: Conducting extensive adversarial simulations and robustness testing, particularly targeting sophisticated attack scenarios including freelancer collusion, deepfake generation, and advanced OCR spoofing.
- LLM Integration for Enhanced OCR Accuracy : Integrating large language models (LLMs) for post-OCR text correction and semantic validation, particularly beneficial for handling noise, irregular layouts, and multilingual content, to further enhance extraction accuracy.

Completing these experiments and analyses will substantially enhance the BDIMS's robustness, credibility, and real-world applicability, establishing a strong foundation for practical deployment in secure digital identity management systems.

6. Conclusions

This paper introduced a blockchain-based decentralized identity management system that combines AI-enhanced OCR, Merkle Tree cryptography, and privacy-preserving identity proofs. Our contributions include: (1) a modular architecture supporting secure and verifiable identity claims, (2) an OCR pipeline augmented with AI techniques for enhanced data extraction from identity documents, and (3) a Merkle-based commitment model for selective disclosure and revocation. Experimental evaluation showed effective mitigation of identity fraud while preserving confidentiality and integrity.

To enhance extraction accuracy—especially in documents with noise, irregular layouts, or multilingual content—we propose incorporating large language models (Future Works) for post-OCR correction. LLMs can contextually infer intended text sequences, improving robustness without human intervention. This integration offers semantic validation of critical identity fields, reducing errors from visual distortions or typographical anomalies.

Future work will prioritize measurable improvements, including latency reduction in identity proof generation, bandwidth-optimized deployment for constrained networks, and benchmarking OCR+LLM pipelines against standard datasets. We also aim to enhance scalability through optimized Merkle Proof computation and transaction batching for high-throughput settings.

While the proposed system is validated in a controlled environment, broader adoption necessitates legal and institutional feasibility studies. Real-world integration—such as with government-issued eIDs or banking KYC systems—requires regulatory alignment and trusted issuer partnerships. A hybrid model is envisioned: certified authorities handle root issuance, while users retain control over attribute disclosure.

We plan to assess compliance with global privacy regulations (e.g., GDPR, APPI) by refining data minimization protocols, consent capture, and storage mechanisms. Institutional Review Board (IRB) approval will be sought for pilot deployments involving personal data.

In conclusion, this work demonstrates a viable pathway for secure, user-centric identity verification using decentralized infrastructure. Future research will anchor technical innovations in measurable performance metrics and practical deployment models.

Author Contributions: Conceptualization, H.V.A.L. and Q.D.N.N.; methodology, H.V.A.L.; software, H.V.A.L.; validation, H.V.A.L., Q.D.N.N. and T.H.T.; formal analysis, H.V.A.L.; investigation, H.V.A.L.; resources, H.V.A.L.; data curation, H.V.A.L.; writing—original draft preparation, H.V.A.L.; writing—review and editing, H.V.A.L. and T.H.T.; visualization, Q.D.N.N.; supervision, T.H.T.; project administration, T.H.T. and N.T.; funding acquisition, T.H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Figure 8 is a sample image for illustrative purposes only and does not contain any personal data or identifiable information of real individuals.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Bradley, J.; Hill, B.; Sakimura, N. Identity Management Using a Centralized Authority. In In Proceedings of the 2014 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA), Hong Kong, 14–18 July 2014; pp. 1–8.
2. Ziegeldorf, J.H.; Morchon, O.G.; Wehrle, K. Privacy in the Internet of Things: Threats and Challenges. *Secur. Commun. Netw.* **2014**, *7*, 2728–2742. [CrossRef]
3. Callas, J. Decentralized Identity: A New Approach to Identity Management. *IEEE Secur. Priv.* **2021**, *19*, 12–18.
4. Allen, C. The Path to Self-Sovereign Identity. Life with Alacrity, 25 April 2016. Available online: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html (accessed on 4 January 2024).
5. Ravidas, S.; Nguyen, K.; Oualha, N. Decentralized identity: A survey on emerging trends and challenges. *IEEE Access* **2022**, *10*, 14038–14060. [CrossRef]
6. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 4 May 2024).
7. Mougayar, W. *The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology*; Wiley: Hoboken, NJ, USA, 2016.
8. Kouhizadeh, M.; Sarkis, J. Blockchain Practices, Potentials, and Perspectives in Greening Supply Chains. *Sustainability* **2017**, *9*, 3652. [CrossRef]
9. Hardjono, T.; Lipton, A.; Pentland, A. Towards an Interoperability Architecture Blockchain Autonomous Systems. *IEEE Trans. Eng. Manag.* **2020**, *67*, 1296–1306. [CrossRef]
10. Reed, D.; Sporny, M.; Longley, D.; Allen, C.; Sabadello, M.; Chadwick, D. Decentralized Identifiers (DIDs)v1.0. World Wide Web Consortium (W3C), Working Draft, 2021. Available online: https://www.w3.org/TR/did-core/ (accessed on 4 October 2024).
11. Smith, R. An Overview of the Tesseract OCR Engine. In: Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Parana, Brazil, 23–26 September 2007; Volume 2, pp. 629–633. [CrossRef]
12. Sovrin Foundation. Sovrin: A Protocol and Token for Self-Sovereign Identity and Decentralized Trust. 2021. Available online: https://sovrin.org/library/sovrin-protocol-and-token-white-paper/ (accessed on 4 February 2025).
13. Veramo. Veramo: Modular Framework for Decentralized Identity. 2024. Available online: https://veramo.io (accessed on 4 February 2025).
14. Microsoft. ION: A Decentralized Identifier Network Built on Bitcoin. 2022. Available online: https://identity.foundation/ion/ (accessed on accessed on 4 February 2025).
15. Alsobeh, A.M.R.; Magableh, A.A. BlockASP: A framework for AOP-based model checking in blockchain systems. *IEEE Access* **2023**, *11*, 115062–115075. [CrossRef]
16. Abuhasan, F.; Ashqar, H.I.; Alsobeh, A.M.R.; Darwish, O. Blockchain-based national digital identity framework—Case of Palestine. In Proceedings of the International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS2024), Dubrovnik, Croatia, 24–27 September 2024. [CrossRef]