Financial Document Authentication and Verification Using Hierarchical Tokenization on Permissioned Blockchains

Chialuka Ilechukwu * , Sung-Chul Hong , Barin Nag

Posted Date: 25 February 2026

doi: 10.20944/preprints202602.1535.v1

Keywords: permissioned blockchain; document authentication; document verification; hierarchical tokenization; provenance; non-transferable tokens; smart contracts; ERC-6150

 doi:10.20944/preprints202602.1535.v1

Article

Financial Document Authentication and Verification Using Hierarchical Tokenization on Permissioned Blockchains

Chialuka Ilechukwu [1,*], Sung-Chul Hong [1] and Barin Nag [2]

[1] Department of Computer and Information Sciences, Towson University, Towson, MD 21252, USA
[2] Department of Business Analytics and Technology Management, Towson University, Towson, MD 21252, USA
* Correspondence: cilechu1@students.towson.edu

Abstract

Document authentication remains a pressing challenge in various domains, including financial services, academic credentialing, healthcare, and supply chain management. Existing centralized verification systems are vulnerable to manipulation, inefficiency, and limited transparency. Blockchain technology, with its immutability and tamper-resistant capabilities, offers a strong decentralized alternative; however, many current implementations lack structured, issuer-bound relationships for documents. This paper proposes a blockchain-based model that leverages a hierarchical token structure to authenticate and trace the provenance of high-value digital documents, with a focus in financial records. The model introduces the concept of an issuer-bound parent token and document-linked child tokens, enforcing a structured trust relationship between a legitimate institution and the documents it issues. By combining on-chain cryptographic hashing with off-chain file references, the approach is designed to balance verifiability with scalability. We implement a proof-of-concept using Ethereum-compatible smart contracts on a permissioned blockchain and evaluate it in a consortium-style financial setting. Our functional analyses demonstrate the model's ability to ensure document integrity, provenance, and resistance to document fraud. This work offers a practical and extensible foundation for secure digital document authentication and verification in financial and other trust-sensitive settings.

Keywords: permissioned blockchain; document authentication; document verification; hierarchical tokenization; provenance; non-transferable tokens; smart contracts; ERC-6150

1. Introduction

1.1. Background and Motivation

The proliferation of forged documents in finance, education, government, healthcare, and other domains undermines trust in digital workflows. Although familiar, centralized verification suffers from single points of failure, low transparency, and easy tampering. As processes digitize, we need mechanisms that ensure document authenticity without re-centralizing trust. Manipulation is now prevalent with consumer tools: high-resolution scans, editable PDFs, and convincing templates. In forensic contexts, image/spectral detection methods are used; however, image-processing approaches are often complex and time-consuming, while spectroscopy can be more accurate but harder to implement, so no single method is at once simple, accurate, automatable, inexpensive, and non-destructive (Al-Ameri et al., 2023). However, most verification occurs outside specialized forensic settings, requiring methods that integrate into everyday operational workflows. This gap becomes costly at scale.

 doi:10.20944/preprints202602.1535.v1

The financial impact of document fraud is material: Europol categorizes document fraud as one of the 'engines of organized crime', citing financial scams as one of the broad spectrum of criminal activities it is linked to; Switzerland has averaged over 4,200 seized fraudulent IDs annually since 2007; Frontex recorded over 8100 fraudulent documents and 6,700 implicated travelers per year in 2017-2018 (Baechler, 2020). Online supply scales the problem: a longitudinal dark-web study reported over AUD 1.8 million in ID-related sales over 2.5 years and found that over 80% of ID-related listings were digital products (e.g., scans/images/templates of identity documents) (Devlin et al., 2024). Beyond financial contexts, falsified vaccine cards show how authentic-looking records can undermine policy and public health decision-making (Ali et al., 2024).

Classical integrity safeguards like public key infrastructure (PKI) and cryptographic hashing are useful but often impractical for everyday users; therefore, user-friendly schemes use secure QR codes that link verification to an issuer-controlled service/database, improving usability but reintroducing reliance on the issuer's infrastructure (Szyjewski, 2023). Blockchain technology popularized by Bitcoin (Nakamoto, 2008) uses a peer-to-peer network and a hash-linked, time-stamped ledger that becomes computationally impractical to alter, allowing participants to share a consistent history without a trusted intermediary. By distributing records across a network of nodes and securing them with cryptographic mechanisms, blockchains can protect the integrity of valuable information against counterfeiting, unauthorized modifications, and other threats. Blockcerts, an open standard for issuing and verifying blockchain-based official records (including academic credentials), enables recipients to hold and share credentials and supports verification via blockchain proofs (Blockcerts, n.d.). In cross-border trade, where blockchain is used for trade documents, a systematic literature review finds that blockchain is best framed as an infrastructural trust layer rather than a universal remedy, especially when integrated with complementary systems (Böhmecke-Schwafert, 2024), and in multi-organization processes, its core contribution is not just immutability but cross-organizational capabilities—shared visibility, aggregation, validation, automation, and resilience, supporting auditable records and coordinated workflows (Yerpude et al., 2022). Blockchain-based traceability research shows that provenance/traceability information (e.g., origin and handling history) is critical alongside content integrity for reducing counterfeit risk (H. Lee & Yeon, 2021). These applications demonstrate that blockchain can support tamper-resistant recordkeeping and verification, strengthening defenses against document fraud in various contexts.

The work by H. Lee & Yeon (2021) also highlights the harm done to brand trust and marketplace credibility, making issuer/source provenance worth paying attention to. In credentialing and identity systems, issuer provenance is often formalized via the Self-Sovereign Identity (SSI) paradigm. SSI structures issuer-credential relationships using Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs). Credentials carry an issuer signature and can be checked against public identifiers/registries, enabling independent validation without contacting the issuer (Nita & Mihailescu, 2024; Satybaldy et al., 2022). Privacy-preserving identity wallet methods (e.g., selective disclosure, Zero-Knowledge Proofs (ZKPs)) further reduce the exposure of data during verification (Podda et al., 2025). We do not implement SSI (DIDs/VCs) or ZKPs in this proof-of-concept; we reference SSI as established context for issuer provenance. Our divergence is to enforce issuer-document lineage and non-transferability, complementing SSI-based models.

Tokenizing documents involves representing documents as non-fungible tokens (NFTs) on a blockchain, allowing the encoding of metadata and ownership information for each document. Existing blockchain document systems typically rely on two separable steps: (i) anchoring document hashes/off-chain pointers on-chain—used in both untokenized and tokenized designs—and, for tokenized designs specifically, (ii) minting flat ERC-721 (non-fungible token standard) tokens that require verifiers to infer issuer authority primarily from the legitimate issuer/organizer smart contract address (Susik et al., 2023) (often obtained from an official website or curated list), rather than from an explicit on-chain hierarchy. Since NFTs were made to be tradable or transferable (which is undesirable for credentials/documents), a malicious actor can decide to transfer a document token without authorization. Soulbound tokens (SBTs) have been proposed as non-transferable tokens held by a wallet ('Soul'), representing

 doi:10.20944/preprints202602.1535.v1

commitments, credentials, and affiliations, and potentially revocable by the issuer (Weyl et al., 2022), and Rejectable SBTs extend SBTs by enabling holder acceptance/rejection and producing on-chain non-repudiation evidence of issuance and reception (Pericàs-Gornals et al., 2024). However, these approaches do not encode a formal issuer-document hierarchy. Similarly, academic credential verification prototypes typically register and verify each credential as a standalone record (e.g., degrees verifiable via QR codes and hash-based queries) (Quispe & Pacheco, 2025), rather than enforcing an on-chain issuer-credential hierarchy at the token layer. The missing piece is an on-chain structure that ties each document token to its issuing authority, with non-transferability enforced by smart contract logic. We do not implement soulbound logic in full; we only focus on the non-transferability aspect.

We adopt hierarchical tokenization: a hierarchical, non-transferable token model in which an issuer's identity is a parent Organization Token (OT) and each document is a child Document Verification Token (DVT), under that parent. Non-transferability blocks post-issuance sale or reassignment of tokens; an on-chain hash anchors the integrity of an off-chain payload; and the system runs on a permissioned blockchain. Compared with hash anchoring and flat ERC-721 NFTs, hierarchical tokenization (a) binds issuer-document at the contract layer, (b) eliminates ad-hoc issuer inference, and (c) creates traceable issuer-document lineage, reducing false-acceptance risk for financial documents (Know-Your-Customer (KYC) files, loan packages, trade/claims evidence) while remaining domain-agnostic.

Because the model targets institutional verification workflows, the deployment environment matters. Deployment note: Public blockchains introduce concerns regarding transaction costs, privacy, and performance. A permissioned blockchain can mitigate these issues by limiting the number of participants, offering configurable fees/throughput, and utilizing more efficient consensus protocols. We therefore use a permissioned network under role-based governance for token issuance. As an example of how permissioned blockchains operationalize governance, Zhai et al. (2023) show that permissioned document workflows use role-separated peers, certificate-based membership/authority control, and private channels, enabling traceable document state changes within a consortium of known participants. Precht et al. (2026) report that blockchain-based document management systems frequently externalize payloads off-chain (about 68%, commonly via IPFS) while anchoring verifiable on-chain references (e.g., hashes/pointers) for scalable integrity checks, and further shows that implementations span both public and enterprise platforms, with permissioned frameworks such as Hyperledger Fabric and private Ethereum deployments among the common enterprise platforms in this space. From an accounting/audit perspective, Georgiou et al. (2024) emphasize that in private/permissioned settings, assurance depends on governance, and internal controls around the blockchain; auditors shift from testing transactions directly to evaluating blockchain controls and governance (e.g., code quality, protocol changes, and power allocation among peers), since blockchain use alone does not necessarily assure reporting reliability. Our model aligns with that view. The implementation details—node roles, consensus mechanism, network configuration, and storage mechanics—appear in a later section.

1.2. Problem Statement

In high-value contexts, trust must extend beyond the file integrity to provenance—the document's origin under an issuing authority. Common designs lack a natively enforced issuer-document linkage; verification often traces addresses or registries rather than reading a formal on-chain relationship. Also, document NFTs are flat and transferable (unsuitable for personal and specific records), and on-chain storage is costly. We design a system for an institutional context, finance-first but domain-agnostic, that (i) cryptographically binds each document to its issuer with tokens, (ii) prevents post-issuance sale or transfer of document tokens, (iii) keeps verification simple and traceable, and (iv) fits role-based, permissioned deployments.

The high-level research question is "Can a hierarchical token model be used to achieve efficient, verifiable document authentication on a permissioned blockchain, improving provenance assurance, tamper evidence, and traceability for financial documents, while remaining applicable across

 doi:10.20944/preprints202602.1535.v1

domains?" This paper aims to design and implement a system that addresses all the stated requirements in unison.

1.3. Research Objectives

By achieving the following objectives, this paper aims to address the above challenges:

- Design a hierarchical tokenization model that encodes a structured issuer-document (parent-child) relationship.
- Implement smart contracts using ERC-721 and ERC-6150 (hierarchical token standard) with necessary features to support the parent-child token logic and non-transferability.
- Integrate an off-chain storage solution for the document payloads while maintaining verification integrity through on-chain cryptographic hashes.
- Deploy the proposed system on a permissioned, Ethereum-compatible multi-node network with governance and access control, mirroring a controlled institutional environment.
- Simulate issuance/verification workflows and conduct a functional evaluation to show that the system preserves document trust and provenance without incurring excessive gas costs or computational overhead, confirming practical feasibility.

1.4. Research Contribution

This work is a proof-of-concept implementation that contributes the following to the state-of-the-art blockchain-based document authentication:

- A theory-led hierarchical tokenization model that enforces issuer-document lineage on-chain and prohibits token transfer.
- A smart contract realization of hierarchical NFTs (ERC-6150), with a form of soulbound behavior (non-transferability), tailored to document verification.
- An on-chain/off-chain strategy that preserves scalability and recomputable proofs in the context of the proposed model.
- A four-node, permissioned deployment that demonstrates feasibility and mirrors institutional governance and controlled access.
- A domain-agnostic verification high-value document verification framework, prioritizing financial documents as the initial use case.

1.5. Paper Outline

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 (Materials and Methods) details the hierarchical model and its components, presents the system architecture and network configuration, and algorithms, while describing the authentication and verification workflow in a financial scenario. Section 4 (Implementation & Results) reports execution evidence and screenshots, a verifier-user interface walkthrough, functional evaluation of the PoC, and basic/security tampering tests. Section 5 (Discussion) interprets the findings, outlining the limitations, and broader implications. Section 6 concludes with key findings and directions for future work.

2. Related Work

2.1. Digital Document Authentication Methods

Traditional digital document authentication involves using cryptographic techniques and watermarking to protect the integrity and originality of electronic or physical copies of documents, such as financial statements, audit reports, and other high-value records. Digital signatures are key in modern authentication; analogous to handwritten signatures, they use public-key cryptography to authenticate the signer and help detect tampering (Subramanya & Yi, 2006). For instance, PDF documents can be signed and later verified using the Elliptic Curve Digital Signature Algorithm (ECDSA) (Ramadhan et al., 2023). Digital signatures have been integrated with QR-code-based

 doi:10.20944/preprints202602.1535.v1

symbols to authenticate the issuer and detect forgery/tampering (Teraura et al., 2020). This approach was applied by embedding an ECDSA-generated digital signature in a QR code on documents to facilitate secure authentication and detect forged signatures (Wellem et al., 2022) and was later extended by adopting the Edwards-curve Digital Signature Algorithm (EdDSA) in a QR-code-based approach to support secure document authentication (Walidaniy et al., 2023).

Digital watermarking introduced a new line of defense; it embeds hidden information, such as text or images, into document content to verify authenticity later. It can be used for both digital and printed documents. Afrakhteh et al. (2010) propose embedding an imperceptible watermark in official printed documents; during verification, the watermark is then extracted to confirm the document's validity, with the extraction process designed to handle print-scan (PS) distortions. Digital watermarks can also be combined with QR codes to make documents resilient to tampering. A study by Mohsin Arkah et al. (2020) authenticated color documents by combining QR codes with an invisible watermark encoded as a color map, reporting visual transparency and enabling detection of tampered regions.

There is no doubt that these conventional authentication techniques laid the groundwork for ensuring document integrity and provenance in many domains, including financial reporting and assurance. These traditional methods, however, often require centralized authorities or trusted third parties (e.g., certificate authorities for digital signatures) to issue and manage credentials, making them susceptible to single points of failure and administrative bottlenecks. These and other potential limitations have paved the way for decentralized solutions, such as blockchain technology, which is discussed in the next subsection.

2.2. Blockchain-based Document Authentication

Blockchain technology has introduced decentralized document verification frameworks, eliminating the need for central authorities. By leveraging an immutable ledger, documents or their digital fingerprints can be recorded on a blockchain, enabling independent verification of authenticity and issuance. This technology has been applied to document authentication in various ways. Usha et al. (2023) did a comprehensive review of blockchain-based approaches for document authentication, highlighting the use of cryptographic hashing, smart contracts, and consensus mechanisms. Their work evaluates platforms like Ethereum and Hyperledger, pointing out their capabilities in enhancing transparency, immutability, and security for digital document management. The review surveys existing implementations that leverage distributed ledger technology to secure educational certificates, professional credentials, and legal documents. Although the surveyed applications span other domains, the same requirements for integrity, auditability, and tamper evidence also apply to high-value financial documents such as audit reports, statements, and regulatory filings. However, the review highlights recurring gaps: manual or human-in-the-loop verification steps, scalability/performance constraints in some implementations, and missing integrated document-generation components—motivating more structured and end-to-end verification models.

Blockchain-based document authentication has been used across various application areas, with strong representation in academic and professional credentials. MIT Media Lab's Blockcerts, already introduced, is a Bitcoin-based credentialing system for issuing verifiable digital certificates, where authenticity is checked against an on-chain hash record and verification can be performed independently by third parties; this approach has been piloted for diplomas, professional certifications, and other educational records, providing a decentralized and secure infrastructure in digital credentialing (Jirgensons & Kapenieks, 2018). While Blockcerts focuses on public blockchain integration and open-access verification, its core principles—immutability, document hashing, and verifier autonomy still resonate with this research's objectives of a permissioned, hierarchical token-based model for document authentication. Cardenas-Quispe & Pacheco (2025) present a Python/Docker prototype for academic degree verification on a hybrid blockchain network (six Docker nodes) that records credential hashes and issues QR-code-verifiable degrees, using an

 doi:10.20944/preprints202602.1535.v1

associative signature process based on Byzantine consensus to validate stored records before issuance. Their approach maintains decentralization and data integrity but does not incorporate tokenization; verification of credentials is done through hashed records, rather than digital tokens. Pu & Lam (2023) analyze multiple blockchain-based digital certificate platforms across industries and show that blockchain records are used to enable independent verification, reducing time and effort spent on manual verification. Silaghi & Popescu (2025) systematically review blockchain-based initiatives for academic certificate management and observe that many proposals focus on recording credentials (and, where supported, revocation status) on-chain, while challenges around governance, interoperability, and real-world adoption remain.

Beyond education verification has been used for supply chain documents and provenance records. In complex supply chains like those for food, pharmaceuticals, and luxury goods, blockchain's immutable and transparent ledger can record each step and important documentation to ensure end-to-end traceability. For example, in the jewelry industry, blockchain has been used to create a secure, tamper-proof log of the journey of gemstones and precious metals from mine to retailer, which verifies the product's authenticity and traces the origin of materials, and can help reduce fraud and strengthen consumer trust (Patel et al., 2025). In the financial sector, blockchain has been explored for sensitive documents. A blockchain-based model for securing financial documents was proposed using digital signatures, cryptographic hashing, and a PKI-managed key infrastructure; the model stores signed document data (e.g., hashes/signatures) and related metadata on-chain and includes a key cancellation mechanism to invalidate compromised signing keys, supporting integrity checks and tamper-evidence in financial document exchange (Mohammad Saeidia et al., 2025).

These use cases show the value of blockchain as it builds upon earlier digital signature methods by providing decentralized trust and persistence. Verification can be done independent of any single provider. A stakeholder can validate the authenticity of a document by comparing it with the trusted blockchain record. In summary, blockchain-based document verification enhances robustness against potential fraud and removes friction from the verification process, although most existing systems still treat documents as hashed records or certificates written directly on-chain rather than as explicit, issuer-bound tokens. This gap motivates Section 2.3, which examines how tokenization of identities and documents can provide a richer foundation for more structured verification models later in the paper.

2.3. Tokenization of Digital Assets: Identity and Document Assurance

While recording document hashes on a blockchain preserves integrity, a new development is tokenizing documents as digital assets. Tokenization refers to representing the ownership or rights of an asset (digital or physical) as a token on the blockchain. Surveys of Non-Fungible Tokens (NFTs) outline established standards (e.g., ERC-721/1155), minting workflows, metadata and URI (Uniform Resource Identifier) references, smart contract mechanics, and recurring challenges around interoperability, lack of standardization, scalability, and storage; they also examine NFT applications across sectors, including healthcare, supply chain management, education, agriculture, gaming, and identity verification, and domains such as digital twins, metaverse, and luxury asset authentication (Hammi et al., 2023; Razi et al., 2024). A complementary systematic review synthesizes how blockchain, NFTs, and digital certificates can be combined for authenticity and provenance without prescribing a single heavy architecture, which reinforces the applicability of tokenized verification (Ramirez Lopez & Morillo Ledezma, 2025). We consider two focal areas in this light: tokenized identity (who issues) and tokenized documents (what is being verified). Accordingly, Section 2.3.1 reviews identity-based issuer binding (SSI/DIDs and tokenized credentials), and Section 2.3.2 reviews tokenized documents (metadata and off-chain anchoring).

2.3.1. Digital Identity: From SSI/DIDs to Tokenized Credentials

SSI and DIDs for issuer-document binding (without tokens)

 doi:10.20944/preprints202602.1535.v1

Self-sovereign identity (SSI) utilizes Decentralized Identifiers (DIDs) and standard cryptographic proofs to bind a digital document (or claim) to an issuing entity, while allowing selective disclosures. The identity layer helps address the primary question in document handling: "Did this file truly originate from the specified issuer?". Nita & Mihailescu (2024) formalized a Web3-oriented flow that clarifies request, issuance, presentation, and verification. It stresses user-controlled disclosure with verifier-side cryptographic checks. Their work outlines an operational sequence that includes key creation, credential issuance, proof of possession, and verification of the issuer's signature to establish a binding relationship between the issuer and the document using DID/VC mechanisms. Complementing this, Satybaldy et al. (2022) motivates SSI for online document verification (illustrated with loan processing) to reduce reliance on intermediaries, improve interoperability, and preserve privacy.

In practice, DIDs and cryptographic proofs can be used to authenticate the lineage of financial documents to an authoritative issuer without revealing unnecessary information, allowing for a direct and clean mapping to KYC file checks, income attestations, and evidential trails for lenders and insurers.

Extending identity with tokenized credentials (NFTs/SBTs)

Some approaches integrate SSI with credential tokens, while others use tokens solely as credentials. Recent designs combine SSI (DIDs) with non-transferable or constrained tokens, so that credentials remain bound to a subject and enforce explicit acceptance/usage policies, and record lifecycle events (issue/accept/reject/revoke) on-chain. In a DID→SBT gating approach, Kim & Ryou (2023) issue an SBT following proof verification, and downstream services grant access by verifying the possession of the SBT rather than repeatedly collecting personal data. For consent evidence and lifecycle control, rejectable SBTs (RejSBTs) add accept/reject semantics and non-repudiation evidence for reception and terms acceptance, with low reported on-chain overheads—useful where both consent evidence and operational cost matter (Pericàs-Gornals et al., 2024). A workshop prototype issues vaccination SBTs to citizen wallets and demonstrates how tokenized status facilitates risk assessment without disclosing full identity across on-chain contracts and off-chain UI/data layers (Lunesu et al., 2023). This shows the end-to-end viability of issuer-attested, subject-bound tokens. Beyond specific prototypes, a broader review synthesizes compliance-relevant identity concerns such as revocation, linkability/monitoring, and the need for standardization (Banaeian Far & Hosseini Bamakan, 2023), while an identity-focused design emphasizes privacy and regulatory controls by anchoring proofs/attestations on-chain and keeping sensitive identity attributes off-chain (Anwar & Gill, 2025). An additional identity-oriented implementation uses non-transferable NFTs for identity verification and document traceability, and stores credential data off-chain (e.g., the InterPlanetary File System (IPFS)) to reduce reliance on centralized repositories (Eltuhami et al., 2022). Another related design uploads encrypted credentials to IPFS and supports selective disclosure, allowing verifiers to validate only the required portions of a credential (Singh et al., 2024).

2.3.2. Tokenized Documents: Metadata and Off-chain Anchoring

Once issuer and holder anchors are established, the artifacts themselves can be represented as tokens. Tokenizing documents means creating their token equivalent (typically an NFT) on a blockchain, governed by smart contract standards. This approach transforms a traditionally static record into an interactive, standard-compliant asset that can harness the full potential of a blockchain ecosystem. For instance, an academic diploma issued as a unique token, assigned to a specific recipient, goes beyond the benefit of a simple blockchain entry. It becomes a verifiable asset that carries embedded metadata, such as the issuer and verification hash, using widely adopted standards like ERC-721. These tokenized documents can then be programmatically validated and, where appropriate, transferred under predefined conditions, enabling secure, automated, and auditable workflows for documents.

Standards-compliant NFTs encode a token's unique identifier and metadata via smart contracts (e.g., ERC-721 and ERC-1155 (multi-token standard)) (Hammi et al., 2023; Razi et al., 2024), and in

 doi:10.20944/preprints202602.1535.v1

document settings, they are commonly paired with cryptographic hashes and off-chain storage pointers so integrity and provenance can be verified without storing the full file on-chain (Kumar et al., 2022). Prototypes across platforms demonstrate issuance and later authenticity checks using on-chain certificate records plus off-chain file hosting—e.g., Solana e-certificate NFTs with Phantom wallet connection and NFT metadata, while the underlying certificate file is hosted off-chain (Artha et al., 2022), as well as Ethereum + IPFS designs that represent educational assets as NFTs after institutional verification, storing the original file on IPFS and using the resulting content identifier (CID), stored/hashed on-chain, as the retrieval and integrity reference (Kumar et al., 2022). Following the same approach, a related study presents an NFT-enabled evidence system that combines fog computing, IPFS, and blockchain to hash evidentiary artifacts and record IPFS hashes/CIDs in NFT metadata for later verification, reinforcing the same integrity-anchoring logic (Peelam et al., 2025). In finance-related settings, the same principle can be applied to statements, invoices, collateral files, and mandated disclosures: provenance is explicit, integrity is checkable, status changes are transparent, and sensitive content remains off-chain under institutional controls.

A fundamental note about storage is that many implementations adopt off-chain storage options to minimize costs and increase scalability. The blockchain stores hashes or essential metadata, and the complete document is stored externally. Off-chain storage is not addressed as a distinct sub-theme in this section. Given its widespread adoption in recent blockchain-based document authentication approaches, we acknowledge it as foundational to this work and discuss implementation details in a later section.

The transition from simply recording hashes on-chain to full tokenization can introduce new functionality, including enhanced interoperability and automated compliance checks, and can facilitate integration with decentralized identity systems. Tokenized documents become not just proof of authenticity but active components within broader digital ecosystems. It also raises operational issues (revocation, metadata sensitivity, and identity linkage) that newer SBT-style designs mitigate by enforcing non-transferability and providing issuer-controlled lifecycle actions (e.g., audit/update or revocation) (Kim & Ryou, 2023; Lunesu et al., 2023), with some designs also requiring explicit holder acceptance of the credential binding (Pericàs-Gornals et al., 2024).

While SSI (DID/VC) frameworks and tokenized credentials address identity and provenance, they typically remain flat and do not encode hierarchical issuer-document relationships, motivating the hierarchical token models discussed in the next Section 2.4.

2.4. The Need for Hierarchical Token Models

As document tokenization progresses, there is a need to model more structured relationships between documents and their constituent parts. For example, real-world records are rarely flat: a financial report aggregates schedules and notes; a legal dossier links master contracts to amendments and exhibits; academic credentials bundle course completions under the final award; and a professional certification may span prerequisite levels. Traditional NFTs, such as ERC-721, treat each token as an independent asset and cannot natively capture parent-child relationships or roll-ups. In practice, an issuer-anchored parent that ties to document-level children makes provenance and supporting evidence traversable and verifiable on-chain, hence the need for an approach that defines these relationships at the contract layer rather than depending on ad-hoc presentation logic.

To address this limitation, Ethereum Improvement Proposal 6150 (EIP-6150; ERC-6150) (K. Lee et al., 2022) introduces Hierarchical NFTs as an extension of ERC-721, standardizing on-chain parent–child relationships through interfaces for querying a token's parent, listing children, and checking root/leaf status—thereby enabling traversal of multi-level trees within a single contract—so composite records (e.g., a consolidated report as the parent and schedules/notes as children) can be represented and navigated as a verifiable hierarchy while remaining compatible with ERC-721 tooling.

Recent academic work reinforces the value of hierarchical tokenization. Ongwook Bae et al. (2024) introduce a hierarchical NFT model that establishes explicit parent-token relationships to address flat ERC-721 limitations in fractional ownership and complex asset structuring. A parent token can mint child tokens with assigned ownership shares, and those child tokens can recursively

 doi:10.20944/preprints202602.1535.v1

mint descendants—enabling fractional ownership and multi-level structuring of the underlying asset. Their platform supports fractional trading, metadata referencing through IPFS, and hierarchical relationship tracking within a smart contract ("CWNFT") implemented on the Klaytn blockchain (KIP-17 standard). Compared to flat NFT standards like ERC-721, their design better accommodated layered asset relationships, but lacks standardization, raising concerns about interoperability. Overall, their work validates the rationale behind adopting hierarchical structures where traceable linkage and compositional logic are required. Their parent-child linkage aligns with our approach toward structured document verification, even though they focus on asset fractionalization rather than secure document authentication.

Bhagat et al. (2025) extend the parent-child NFT model with dynamic mechanisms for splitting and merging based on ERC-6150, which provides standardized support for traceable parent-child relationships. Their smart contracts enable NFTs to be hierarchically split into child tokens and later merged, while retaining proportionate share distributions and recursive ownership mapping. This illustrates how ERC-6150-style parent–child linkage, combined with split/merge functions (burning selected children and minting a new token with the combined share under the same parent), can preserve hierarchical lineage while keeping ownership-share accounting consistent across the tree. Although motivated by fractionalized digital assets, their protocol's capabilities can be applied to documents (e.g., sub-reports, appendices, and amendments linked under a master record), with lineage retained for audit, revocation, or re-issuance. In our case, such decomposition is quite valuable, but it is secondary to the primary issuer-anchored linkage described in this paper.

Other domains complement the need for structured token relationships: López-Pimentel et al. (2025) propose a government-anchored digital identity ecosystem in which a state-issued, unique non-replicable identity token sits at the core, and entities attach certified/uncertified tokens (e.g., from educational and financial institutions) to form a verifiable token-based identity tree, illustrating how a trusted parent identity can bind subordinate records for accountable, traceable linkage at scale.

This survey of NFTs for digital twins highlights architectures where tokens encode identity and provenance-oriented traceability, reflect lifecycle changes via updatable (dynamic) metadata, and combine on-chain token records with off-chain storage/metadata references (Hasan et al., 2024). While not focused on documents, the same idea of linking components and tracking evolution over time maps directly to document stacks (e.g., financial statements evolving across quarters with linked schedules and notes), motivating structured token relationships that verifiers can traverse. Kuznetsov et al. (2024) propose an NFT-based global digital registry that maps standard identifiers (e.g., ISBN/ISSN/DOI) to ERC-721 tokens, supports metadata updates with version history and nested (parent–child) relationships, and uses IPFS-backed off-chain metadata referenced on-chain for lookup and verification. The same identifier-and-linkage pattern applies to documents: section- or module-level identifiers should roll up into a parent record with machine-readable links for search validation, and policy checks—validating the need for an on-chain hierarchy (nested/parent-child NFTs) rather than unrelated, flat NFTs. Guidi & Michienzi (2023) review the shift from NFT 1.0 (largely static) to NFT 2.0 (dynamic/interactive assets with enhanced metadata), situate this evolution within common standards (ERC-721/1155), discuss composability (including ERC-998, the Composable NFT standard) and upgrade/migration considerations (e.g., proxy-based updates), and note that NFT 2.0 introduces nested NFTs—parent NFTs that can contain child NFTs—providing a direct mechanism for representing structured, multi-part assets. This naturally points towards hierarchical models for document use cases, where issuance, approvals, amendments, expiries, and dependencies can be represented on-chain. An applied certification and traceability system for refurbished medical devices uses a parent–child token hierarchy to link a device (parent) with replacement parts and certificate documents and uses non-transferable NFTs to anchor refurbishment certificates; metadata is stored via IPFS, with a working DApp and security analysis (Gebreab et al., 2023). Although adapted to physical assets, the issuer-document linkage and traceable lineage concept can also be mapped to document tokenization: a parent record can reference children

 doi:10.20944/preprints202602.1535.v1

documentation (work orders, certificates), with lifecycle-related updates reflected through dynamic NFTs and on-chain references for inspection and compliance.

For completeness, ERC-998 by Lockyer et al. (2018) specifies composable NFTs, allowing one token to own other NFTs or fungible tokens as a bundling mechanism (e.g., a dossier token holding a set of document tokens). While beneficial for moving sets together, ERC-998 does not ascribe hierarchical semantics to those relationships (e.g., prerequisite order, issuer-document lineage). In contrast, ERC-6150 provides explicit parent-child linkage that better matches document structures centered on the issuing entity. These hierarchies also map cleanly onto permissioned networks, where access control and off-chain hash-checks for the underlying files operate in tandem with on-chain relationships to enable verifiable yet privacy-preserving workflows.

Taken together, these works establish hierarchical token models as the next practical step beyond flat NFTs, making issuer–document linkage, dependencies, and verification automatically enforced through smart-contract logic while preserving an auditable chain of evidence. Building on this foundation, section 3 introduces a simple, issuer-anchored hierarchy in which a parent identity token links to document-level tokens with off-chain hash proofs, prioritizing clarity and interoperability over unnecessary complexity, while leaving fine-grained document decomposition (e.g., sub-reports, appendices) for future work.

3. Materials and Methods

The section describes the hierarchical tokenization method used to implement an issuer-bound document authentication prototype on a permissioned blockchain. The method digitally represents an issuing organization and their authenticated documents using a two-tier token system deployed as smart contracts. An Organizational Token (OT) and Document Verification Tokens (DVTs) are arranged in a parent–child relationship, with smart contracts enforcing minting rules, storing minimal metadata, and linking on-chain state to off-chain document storage for integrity and scalability enhancement.

3.1. Overview of the Hierarchical Model

The hierarchical model, illustrated in Figure 1, introduces a two-level token structure. At the core is the parent Organization Token (OT), a non-transferable NFT representing the identity of a verified institution (e.g., a financial institution, regulator, or certifying body). Once issued, the OT holder is the only entity permitted to mint children, non-transferable Document Verification Tokens (DVTs). Each DVT represents a verified document (e.g., an audited financial statement or audit report) and is cryptographically linked to its parent OT. DVTs are 'lightweight' in the sense that they store only essential metadata on-chain, such as a document identifier, cryptographic hash, and storage pointer, while the full document remains off-chain.

This structure serves three main objectives:

1. **Issuer-bound verification.** DVTs cannot exist independently; they are hierarchically linked to a legitimate OT, anchoring each document to its issuing institution.
2. **Scalability and efficiency.** On-chain data is minimized to hashes and references, while large document content is stored off-chain.
3. **Governance and control.** Only OT holders are authorized to mint DVTs, aligning token issuance with real-world institutional hierarchies and internal controls.

 doi:10.20944/preprints202602.1535.v1

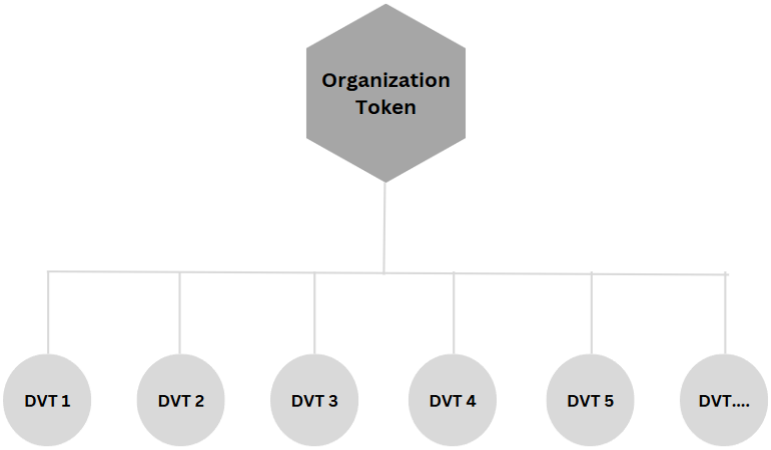


Figure 1. The proposed hierarchical token structure linking an Organization Token (OT) to its Document Verification Tokens (DVTs).

3.2. Model Components

The hierarchical tokenization method comprises a permissioned blockchain network, Organization Tokens (OT), Document Verification Tokens (DVTs), their parent–child relationship, an off-chain storage and integrity layer, token-standard choices, and smart-contract logic.

3.2.1. Permissioned Blockchain Network

The model is deployed on a permissioned blockchain, where participation is restricted to authorized institutions and verifiers. A legitimate organization can mint an Organization Token (OT) and associated Document Verification Tokens (DVTs), its approved members can participate in consensus, and only authorized verifiers access the verification interface. Unlike public blockchains, a permissioned network provides fine-grained access control, institutional governance, and predictable performance, all of which are essential for document authentication in regulated financial and institutional contexts. Vetting members and controlling verifier access ensures that users query immutable, tamper-evident records from trusted nodes.

3.2.2. Organization Token (OT)

Each OT is a unique, non-transferable NFT representing an institution's digital identity within the network. It serves as the parent anchor for all DVTs issued by that institution. OT metadata stored on-chain includes the token ID, token name, and a URI pointing to off-chain organizational metadata. Only the authorized organization node (admin) is permitted to mint an OT, ensuring a controlled and auditable issuance process. Non-transferability prevents impersonation or trading of institutional identity and guarantees that only a verified institution can issue document tokens.

3.2.3. Document Verification Token (DVT) and Document Requirements

A DVT is a non-transferable child token linked to a parent OT. It represents a specific document authenticated by the issuing organization, such as an audited financial statement, regulatory filing, license, or academic transcript. On-chain DVT metadata includes the DVT token ID, the parent OT ID, a cryptographic hash of the document and a URI pointing to the off-chain file. Only the owner of the OT can mint DVTs associated with it, preserving issuer authenticity. Non-transferability maintains the permanent link between each document and its issuing organization, preventing resale or misattribution of verified documents. Once minted, a DVT serves as durable, tamper-evident proof that a particular document instance was issued by the institution.

To ensure reliable hash-based verification, the model assumes that documents used for DVTs satisfy the following requirements:

- **Static and finalized.** Documents are in their final form at issuance and are not edited afterward.

 doi:10.20944/preprints202602.1535.v1

- **Deterministically hashable.** Files use formats that yield stable hashes (e.g., PDF, XML) without dependence on external resources.
- **Self-contained and time-independent.** Content and structure do not change based on access time, environment, or external links.
- **Institutionally approved.** Each document reflects the institution's official, authoritative version at the time of issuance.
- **Material significance.** Only documents whose authenticity materially matter (e.g., certifications, contracts, high-value records) are tokenized.
- **Verifiable access.** Documents remain accessible for verification, either publicly or through controlled access mechanisms.

These assumptions ensure that the DVT mechanism can be securely applied across domains while preserving document integrity and long-term trust.

3.2.4. Parent–Child Relationship

The model enforces a strict hierarchical relationship between the OT and DVTs. Every DVT must reference an existing OT via the parent token's ID. This relationship is encoded in the smart contract and is fully verifiable on-chain. The linkage guarantees the origin traceability, prevents orphaned or unauthenticated tokens, and enables efficient hierarchical queries, such as retrieving all DVTs issued under a particular OT or resolving the parent OT for a given DVT.

3.2.5. Off-Chain Storage and Integrity Proofs

The Documents themselves are not stored on-chain due to storage limitations and gas costs. Instead, organizations use secure off-chain storage, such as IPFS/Pinata or institutional document management systems. The blockchain records a Keccak-256 hash of each document alongside a URI or link used to retrieve the file. OT metadata JSON files can also be stored off-chain, with their URIs referenced from the OT. This design balances scalability and trust: large documents remain off-chain, while integrity is enforced through cryptographic linkage between the on-chain hash and the retrieved file.

Figure 2 illustrates the combined metadata architecture of the OT and DVTs within the hierarchical model. Core metadata is stored on a permissioned blockchain—DVT ID, parent OT ID, parent token name, document hash, and URIs—while the underlying files reside off-chain. This separation enables decentralized, tamper-evident verification without incurring excessive on-chain storage costs.

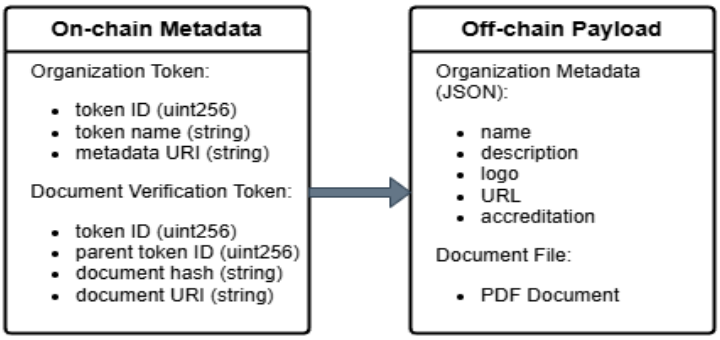


Figure 2. On-chain metadata and off-chain payload in the proposed model.

3.2.6. Token Standards Compliance

The model builds on NFT token standards. ERC-721 provides the basic non-fungible token interface but lacks native support for hierarchical relationships. Implementing parent–child links solely with ERC-721 requires additional manual mappings in the smart contract.

 doi:10.20944/preprints202602.1535.v1

To model issuer–document hierarchies more naturally, the design adopts concepts from ERC-6150, which introduces native parent–child token logic. At the contract level, key elements include:

- Implementing *parentOf(tokenId)* and *childrenOf(parentTokenId*) functions.
- Overriding the *_transfer* function to enforce non-transferability for the OT and DVTs.
- Internally mapping each DVT to its OT via a stored *parentTokenId.*

Table 1 summarizes how ERC-721 and ERC-6150 compare for this hierarchical document-token use case.

Table 1. *Comparison of ERC-721 and ERC-6150 for hierarchical document-token modeling.*

Feature	ERC-721	ERC-6150
Parent–child token relationship	Not natively supported; requires manual mappings	Natively supported via *parentOf()* and related functions
Multi-level hierarchy	Flat, single-level ownership	Supports tree-like multi-level hierarchies
Transferability	Transferable by default; must be overridden for non-transfer	Transferable by default; allows hierarchy-aware transfer restrictions
Suitability for document issuance	Feasible with workarounds (e.g., external mappings)	Well-suited to issuer–document hierarchies
Customizability for the DVT use case	Requires extensive custom logic for hierarchy	Designed for hierarchical assets; aligns more naturally with the DVT model.

3.2.7. Role of Smart Contracts

Smart contracts form the enforcement layer of the hierarchical token model. They define:

- Which address is permitted to mint an OT
- How DVTs are created and linked to their parent OT
- The rules that disable token transfers.
- How document hashes, URIs, and other metadata are stored immutably on-chain.

These rules ensure that only verified institutions issue authenticated document tokens, preserving the trust model without centralized oversight. The logic is implemented in Solidity contracts deployed on the permissioned network and provides a transparent, auditable mechanism for token creation, hierarchy management, and metadata integrity.

3.3. System Architecture

The system architecture instantiates the hierarchical OT–DVT model (Section 3.2) on a permissioned Ethereum-compatible blockchain and integrates it with off-chain storage and verification tools. Figure 3 shows the logical architecture, while Figure 4 depicts the four-node Hyperledger Besu topology.

3.3.1. Logical Architecture

As illustrated in Figure 3, the system comprises the following components:

- **Minting interface (Remix IDE + MetaMask) -** The issuing institution uses Remix IDE, connected to MetaMask, as an administrative interface to deploy the Solidity contracts and to invoke minting functions for the Organization Token (OT) and Document Verification Tokens (DVTs) during the proof-of-concept.
- **Permissioned Besu network -** A Hyperledger Besu network, configured with Clique Proof-of-Authority (PoA), hosts the OT–DVT contracts and maintains the ledger state. All token operations (minting, metadata queries) are recorded as transactions in this network
- **Hierarchical token smart contracts** - Custom Solidity contracts implement ERC-6150-style parent–child logic for the OT and DVTs. On-chain metadata is kept intentionally lightweight

 doi:10.20944/preprints202602.1535.v1

and includes the DVT token ID, parent OT ID, the Keccak-256 hash of the underlying document, and a URI pointing to the off-chain payload. Contract logic also enforces access control and non-transferability for the OT and DVTs and kept extensible for optional revocation behavior.

- **Off-chain storage and IPFS integration** - To optimize storage, documents and auxiliary JSON metadata are stored off-chain using IPFS, with a pinning service (Pinata) ensuring persistence. When a document is uploaded, IPFS returns a content identifier (CID), which can be expressed as a content-address URI. This URI, together with a Keccak-256 hash of the original file, is then submitted to the *mintDVT() function*. The smart contract stores the *documentHash* and *documentURI* as the DVT's verification fingerprint and lookup pointer; no document content is stored on-chain.

Verifier web interface and hashing logic - A lightweight HTML/JavaScript front end, implemented with Ethers.js, connects to the verifier node to retrieve DVT metadata. The verifier web interface connects to a read-only RPC endpoint on the dedicated verifier node described in Section 3.3.2. Given a DVT ID, the interface:

- Queries the contract for the parent OT ID, document hash, and URI.
- Downloads the referenced file from IPFS.
- Recomputes the Keccak-256 hash in the browser (using the same logic as the Python-based hashing script used at minting time).
- Compares the recomputed hash with the on-chain hash to return an authenticity verdict to the user.

This architecture provides a full path from token issuance to independent verification, while keeping on-chain storage minimal and preserving institutional control over token minting.

3.3.2. Besu Topology, Deployment, and Permissioning.

Figure 4 shows the four-node Hyperledger Besu topology used to implement the permissioned network configured with the Clique Proof-of-Authority (PoA) consensus protocol:

- **Organization node -** Participates in consensus and is the minting authority for the OT and DVTs. It is the node connected to the administrative (Remix) interface, and its Ethereum address is whitelisted to submit state-changing transactions.
- **Two consensus-only nodes** - Participate in consensus but do not mint tokens. They improve network resilience, availability, and fault tolerance by maintaining additional replicas of the ledger.
- **Verifier node** - Maintains a synchronized copy of the blockchain but does not participate in consensus. It exposes a read-only RPC endpoint used by the verifier web interface for metadata queries, ensuring that verifiers cannot alter network state.

All four nodes run in isolated Docker containers orchestrated with Docker Compose on a shared bridge network. The Compose configuration specifies RPC ports, peer-to-peer networking, and custom genesis settings (Clique PoA). Static peer discovery and node-level permissioning restrict participation to the four authorized enode identities, while account-level permissioning ensures that only the organization node's address can submit state-changing transactions.

This deployment confirms that the hierarchical token model can operate on a private, multi-node, permission-controlled Ethereum-compatible network, with clear separation among node roles: minting authority, consensus-only (validator-only), and read-only verification.

 doi:10.20944/preprints202602.1535.v1

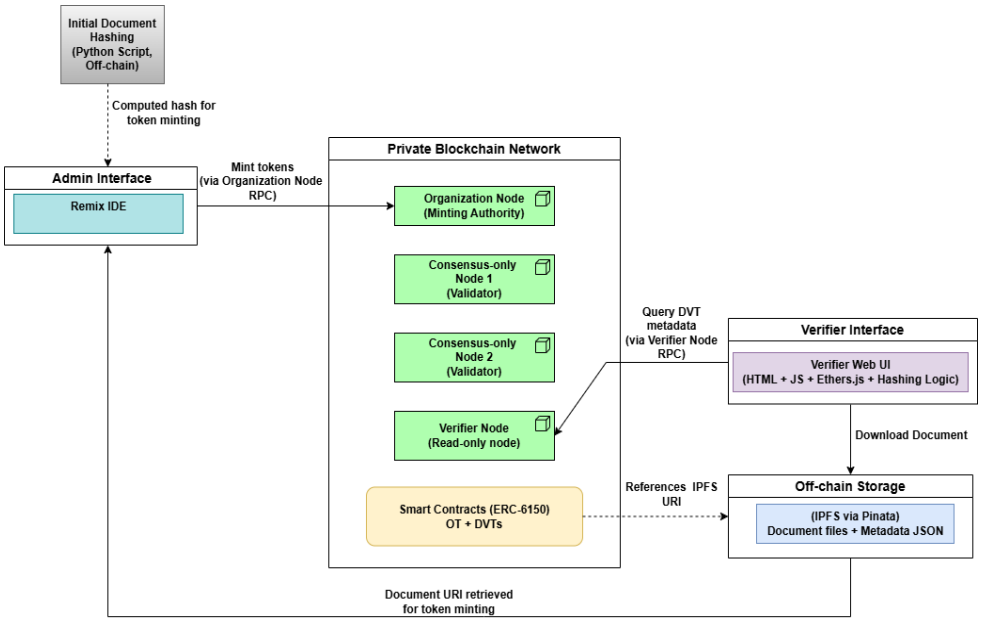


Figure 3. Logical system architecture linking the minting interface, Besu permissioned network, off-chain IPFS storage, and verifier web interface.

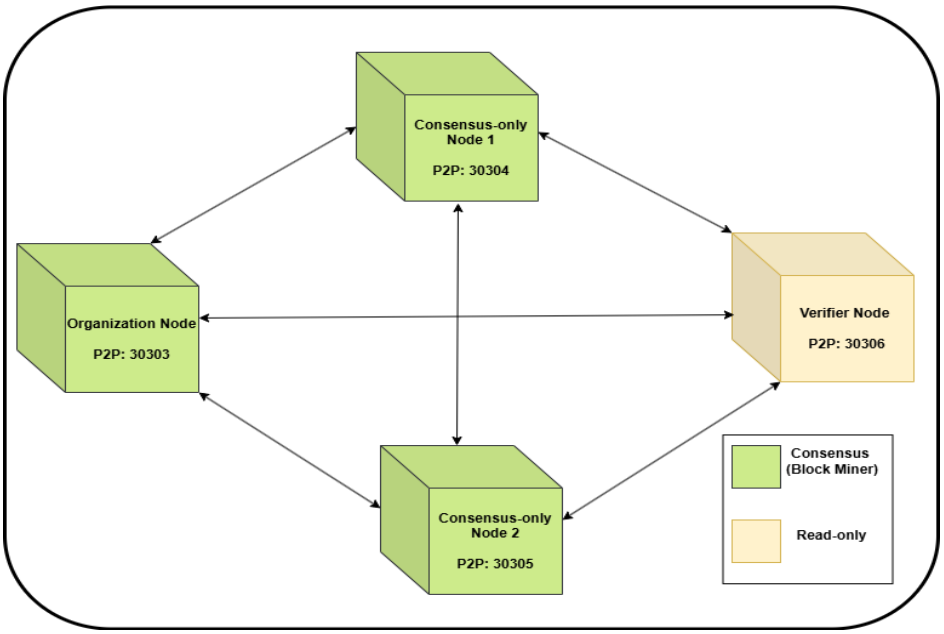


Figure 4. Four-node Hyperledger Besu topology showing the organization node, two consensus nodes, and a read-only verifier node.

3.4. Blockchain Platform Selection and Justification

Building on the requirements outlined in Sections 3.2 and 3.3, the blockchain platform must support hierarchical NFTs for document authentication in a permissioned, consortium-style environment. In such settings, institutions require tamper-resistant ledgers and strong provenance guarantees, but also practical control over membership, governance, and operating costs.

Permissioned blockchains are therefore preferable to permissionless networks such as Bitcoin or public Ethereum because membership can be restricted to identified entities, transaction validation can be limited to authorized validators, and governance can be tailored to audit and compliance requirements (Faccia et al., 2022)

For this hierarchical token-based document authentication system, the platform must in particular:

 doi:10.20944/preprints202602.1535.v1

- Allow the authenticity of documents to be determined solely by authorized institutional actors, without external intermediaries.
- Provide an immutable, auditable record of token issuance and verification, visible to all authorized participants.
- Restrict participation to known, identifiable members of the consortium, with configurable read and write permissions.
- Support high-volume transaction processing without significant performance degradation.
- Enable automation of issuance and verification workflows through smart contracts.
- Protect sensitive document data by keeping only hashes and metadata on-chain and relying on permissioned access and off-chain storage, rather than advanced privacy schemes such as zero-knowledge proofs or fully private contracts.

These requirements lead naturally to an Ethereum-compatible, permissioned platform that can support hierarchical NFTs and standard Ethereum tooling.

3.4.1. Selection Criteria.

Based on the above, the following criteria were defined for choosing the blockchain platform:

- **Ethereum compatibility -** Full support for the Ethereum Virtual Machine and standard token interfaces (e.g., ERC-721 and ERC-6150-style hierarchies), enabling flexible smart-contract implementation of OT–DVT relationships.
- **Consensus flexibility for permissioned networks -** Support for configurable Proof-of-Authority consensus mechanisms (such as Clique, IBFT 2.0, or QBFT), allowing organizations to trade off finality, validator count, and fault tolerance.
- **Permissioning and governance -** Native support for node-level and account-level permissioning so that only authorized entities can participate in consensus, mint tokens, or access network data, consistent with consortium governance models.
- **Deployment flexibility -** Support for Docker-based on-premise or cloud deployment, with the ability to customize network parameters, node configurations, and access-control policies to match organizational security requirements
- **Enterprise readiness and tool integration -** Compatibility with standard Ethereum tools (Remix, MetaMask, Ethers.js) for contract development, deployment, and front-end integration, and strong auditability through smart contracts.
- **Scalability and performance -** Demonstrated ability to support substantial transaction volumes and document-token issuance in permissioned, multi-node deployments (Khan et al., 2025; Pierro et al., 2024).
- **Pragmatic data protection -** Emphasis on permissioned membership, role-based access control, and off-chain storage of payloads, rather than reliance on advanced privacy mechanisms that are not implemented in this proof-of-concept

These platform capabilities (permissioning and PoA options) are documented in Hyperledger Besu's official documentation (Hyperledger Foundation, 2025).

These criteria ensure that the chosen platform can support hierarchical tokenization, controlled governance, and efficient verification workflows in a realistic enterprise setting.

3.4.2. Choice of Blockchain Platform

Four platforms were considered: Hyperledger Fabric, Quorum (GoQuorum), private Ethereum using Geth, and Hyperledger Besu (Khan et al., 2025; Pierro et al., 2024).

- **Hyperledger Fabric** is a mature permissioned framework widely adopted for enterprise workflows. However, it does not natively support Ethereum token standards; tokenization must be implemented as custom chaincode that mimics ERC-20 or ERC-721 semantics. Implementing and maintaining hierarchical NFTs in Fabric would therefore require additional development effort and

 doi:10.20944/preprints202602.1535.v1

separate tooling, making it less aligned with an Ethereum-based NFT ecosystem (Hyperledger Foundation, 2023).

- **Quorum (GoQuorum)** is an enterprise Ethereum fork that provides advanced privacy features such as private transactions, encrypted payloads, and transaction managers like Tessera (ConsenSys, 2024). By contrast, the hierarchical token model in this paper emphasizes transparency within a permissioned consortium: all authorized participants should observe the same OT–DVT hierarchy and hash anchors. Quorum's privacy mechanisms are not required for this design and would introduce additional infrastructure and operational complexity.
- **Private Ethereum with Geth** as commonly used in research prototypes (e.g., (Khan et al., 2025)), offers full EVM compatibility and supports private-network configuration for permissioned deployments (Ethereum Foundation, 2024). However, Geth is primarily geared towards public Ethereum and does not provide rich, built-in permissioning or consortium-management features; these must be added externally, increasing the engineering overhead for a research-grade, multi-node permissioned deployment.
- **Hyperledger Besu** is an enterprise Ethereum client designed for both public and private permissioned networks, with comprehensive node- and account-level permissioning and multiple PoA consensus algorithms (Clique, IBFT 2.0, QBFT) (Hyperledger Foundation, 2025). As an EVM-compatible client, Besu natively supports smart contracts implementing ERC-721, ERC-1155, and ERC-6150-style interfaces, allowing the OT–DVT model to be realized without modifying the client. Besu integrates well with standard Ethereum tooling (Remix, MetaMask, Ethers.js) and has been empirically evaluated in token-centric PoA deployments, including comparative studies with Quorum in energy-token scenarios (Pierro et al., 2024).

Given the selection criteria in Section 3.4.1, Hyperledger Besu provides the most balanced option. It:

- Enables transparent, issuer-anchored token hierarchies using standard Ethereum smart contracts.
- Supports permissioned, multi-node deployments with configurable PoA consensus (Clique in this proof-of-concept) for low-cost, predictable finality.
- Offers native node- and account-level permissioning without requiring additional privacy layers that are not used in this work.

Hyperledger Besu therefore aligns closely with the requirements of a hierarchical document-authentication system: Ethereum compatibility, flexible permissioning, and a pragmatic privacy model based on permissioned access and off-chain storage, rather than advanced cryptographic privacy schemes.

3.4.3. Network Parameters and Toolchain

The permissioned network was instantiated as a private Hyperledger Besu chain configured with Clique Proof-of-Authority via a custom genesis file. The genesis configuration specifies the validator set, private chain identifier, block period, and gas limits suitable for token and metadata transactions, and pre-funds the four nodes described in Section 3.3.2 to cover smart-contract deployment and minting costs.

The implementation used Hyperledger Besu (v25.2.0), Solidity compiler (Remix IDE online, v0.8.31), Docker (v28.0.1) and Docker Compose (v2.33.1-desktop.1), Node.js (v16.20.2), and standard Ethereum tools (Remix IDE, MetaMask, Ethers.js) for contract deployment and verifier interface integration. The full genesis file, Docker Compose configuration, and permissioning TOML files (accounts and nodes) are provided in the project's online repository/supplementary materials for exact replication.

3.5. Algorithms

This section summarizes the core logic of the hierarchical token model as language-agnostic pseudocode. Algorithms 1 and 2 describe the minting of the Organization Token (OT) and Document

 doi:10.20944/preprints202602.1535.v1

Verification Tokens (DVTs). Algorithm 3 captures the hierarchical queries between parents and children. Algorithm 4 outlines access-control checks for minting. Algorithm 5 describes the off-chain verification process that recomputes the document hash and compares it to the on-chain anchor.

Algorithm 1: MintOrganizationToken (OT minting)

1: **Input:** admin address a, orgId, orgName, orgURI.
2: Require that a is the authorised admin and has not previously minted an OT.
3: Require that orgId has not been used.
4: Mint a non-fungible token with ID orgId to a.
5: Store organisation metadata (orgName, orgURI) under orgId.
6: Record that a has an OT and link $a \rightarrow orgId$.
7: Mark the OT as non-transferable (soulbound).
8: **Output:** OT ID orgId. *(used later as parentId for DVTs)*

Algorithm 2: MintDVT (document token under an OT)

1: **Input:** issuer address a, document hash h, document URI uri.
2: On-chain: fetch parentId = orgTokenIdOf[a]; require parentId ≠ 0 and ownerOf(parentId) = a.
3:Require that h and uri have not been used by any previous DVT.
4: Allocate a new DVT ID dvtId from an auto-increment counter.
5: Mint DVT dvtId to a and set its parent to parentId.
6: Store DVT metadata (parentId, h, uri) and mark hash and URI as used.
7: Mark the DVT as non-transferable (soulbound).
8: **Output:** DVT ID dvtId

Algorithm 3: Hierarchical Queries (parent–child)

1: **Input (A):** OT ID parentId.
2: Verify that parentId exists.
3: Return the list of DVT IDs childrenOf(parentId).
4: **Input (B):** DVT ID dvtId.
5: Verify that dvtId exists.
6: Return the parent OT ID parentOf(dvtId).

Algorithm 4: Access-Control Check for Minting

1: **Input:** caller address a, action ∈ {MintOT, MintDVT}.
2: If action = MintOT:
3: Require that a is the contract owner/admin and has not minted an OT.
4: If true, authorise; otherwise, reject.
5: If action = MintDVT:
6: Require that orgTokenIdOf[a] ≠ 0 and ownerOf(orgTokenIdOf[a]) = a.
7: If true, authorise; otherwise, reject.
8: **Output:** allow / reject

Algorithm 5: Off-Chain Document Verification (hash comparison)

1: **Input:** DVT ID dvtId.
2: On-chain: via the verifier node, call getParentTokenId(dvtId) to obtain parentId; if parentId does not correspond to a valid Organization Token, classify the document as Not authentic and abort.
3: On-chain: read DVT metadata for dvtId and obtain stored hash h_stored and document URI uri.

 doi:10.20944/preprints202602.1535.v1

4: Off-chain: fetch the document bytes from uri (IPFS or HTTPS).
5: Compute h_computed = Keccak-256 of the fetched bytes.
6: Normalise h_stored and h_computed to the same hex format.
7: If h_computed = h_stored, classify the document as **Authentic**; otherwise classify as **Not authentic**.
8: **Output:** verification result (Authentic / Not authentic)

3.6. Document Authentication and Verification Workflow (financial-document scenario)

This practical sequence for deploying the smart contract and issuing tokens in a financial institution setting (e.g., banks, audit firms) is outlined below for relevant financial documents such as account statements, audit reports, and loan approval letters.

(a) Smart-contract deployment

1. A regulated financial institution (or its designated administrator) deploys the hierarchical-token smart contract to the permissioned blockchain.
2. This initial deployment instantiates the document-authentication logic and the rules governing the issuance of the OT and DVTs.

(b) Document issuance process

i. **Document creation.** - The issuing organization generates a digital financial document and stores it off-chain (e.g., IPFS or a secure internal document server).

ii. **Document hashing -** Before minting, the institution computes a cryptographic Keccak-256 hash of the final document content using a trusted tool, script, or UI component. The resulting hash is retained for tokenization.

iii. **Organization Token minting -** The institution mints its non-transferable OT (e.g., via mintOrganizationToken), establishing its verified on-chain identity. This is a one-time action that anchors all subsequent DVTs.

iv. **Initiating document issuance -** To issue a document on-chain, the institution calls the DVT-minting function (e.g., mintDVT) on the deployed contract, supplying the pre-computed document hash and the URI pointing to the off-chain file.

v. **Smart-contract validation -** Before minting a DVT, the contract performs two key checks:

- **OT ownership verification -** It verifies that the caller address (*msg.sender*) is the owner of a valid OT, confirming the legitimacy of the issuing financial institution.
- **Parent–child consistency -** It confirms that the referenced OT exists and can serve as the parent in the hierarchy.

vi. **DVT minting -** If the checks pass, the contract mints a new DVT as a child of the OT, assigns it a unique ID from the DVT counter, and stores the document hash and URI in the DVT metadata.

These steps constitute the **authentication phase** of the system and are represented in the left-hand side of Figure 5.

(c) Verification process

The verification logic enables decentralized authentication of financial documents without requiring any blockchain transactions from verifiers. It relies on on-chain metadata and off-chain storage to validate integrity and provenance.

Verifier workflow and validation logic

i. **Verifier access and DVT query -** A verifier (e.g., employer, regulator, auditor, counterparty, credit bureau) is granted read access to the permissioned blockchain, typically via a verifier node. The verifier supplies a DVT ID to query the smart contract.

ii. **Parent token retrieval -** The verifier calls *getParentTokenId(dvtId)* to obtain the OT ID associated with the DVT, confirming the link to a specific financial institution.

iii. **DVT metadata retrieval -** The verifier calls *getDVTMetadata(dvtId)* to retrieve the parent OT ID (for cross-check), the stored document hash, and the document URI (IPFS link or storage URL).

iv. **Parent-token match check.**

The verifier compares the parent OT ID returned by *getParentTokenId* with the parent ID embedded

 doi:10.20944/preprints202602.1535.v1

in the DVT metadata. If they do not match, the process is terminated, and the document is treated as untrusted.

v. **Document retrieval.**

If the parent IDs are consistent, the verifier fetches the document from the off-chain document URI provided in the metadata.

vi. **Hash recalculation.**

The verifier recomputes the document hash using the same Keccak-256 algorithm on the fetched bytes.

vii. **Hash comparison.**

The recomputed hash is compared to the hash stored in the DVT metadata. If they match, the document is considered authentic; if not, it is treated as altered or replaced.

Verification outcome

- **Match:** the financial document is authentic and issued by the correct organization
- **No match:** the document is not authentic.

Because verifiers only perform **read-only queries** and off-chain hashing, no additional blockchain transactions are required during verification, which saves gas and supports scalability. These steps constitute the **verification phase** of the system and are represented on the right-hand side of Figure 5.

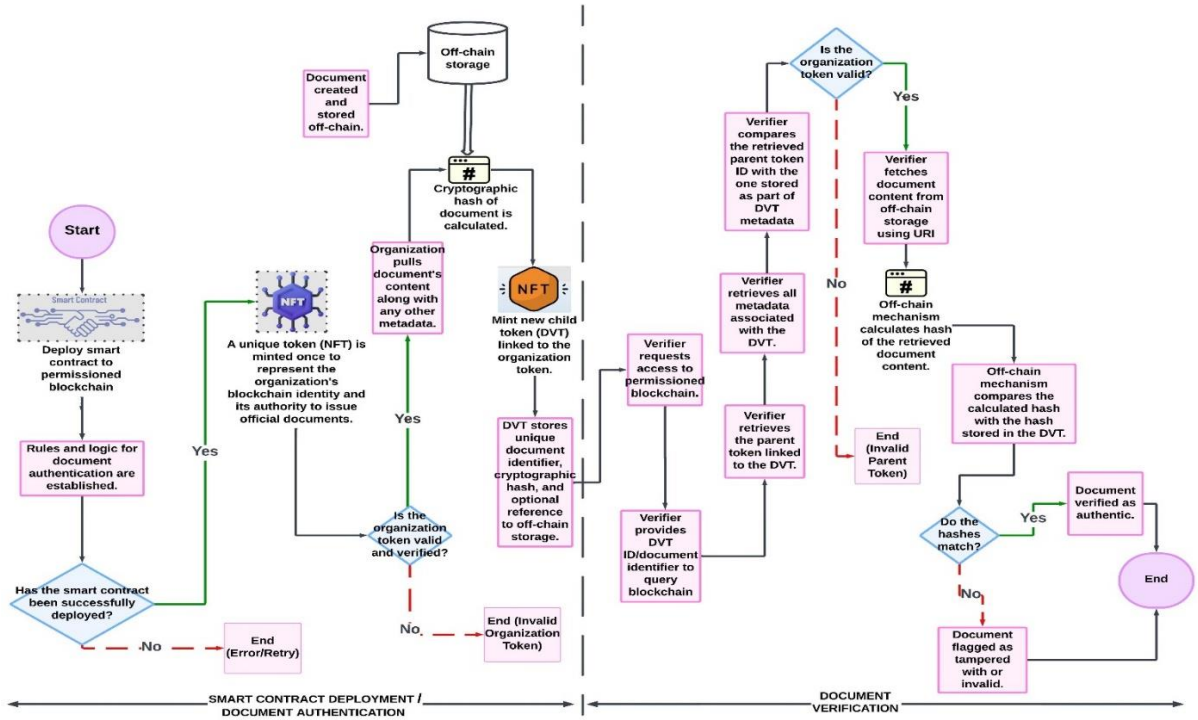


Figure 5. End-to-end process flow of the hierarchical tokenization approach.

The diagram illustrates the full lifecycle of the system, from smart-contract deployment and OT/DVT minting (left) to hash-based validation and verification outcome (right).

4. Implementation and Results.

4.1. Smart-Contract Execution Evidence

This section reports execution evidence that the hierarchical token contracts compiled, deployed, and behaved as intended on the four-node Hyperledger Besu Clique network. All transactions were initiated via the Remix IDE and signed in MetaMask using the organization node's administrator account. Token minting and policy enforcement were observed through Remix transaction receipts, and emitted Solidity events. All four Besu nodes were successfully launched and remained

 doi:10.20944/preprints202602.1535.v1

synchronized during testing; the full Docker Compose configuration is provided in the online repository (Supplementary Materials). Token minting was validated via transaction logs and wallet confirmations in MetaMask, confirming that each operation was mined into a finalized block.

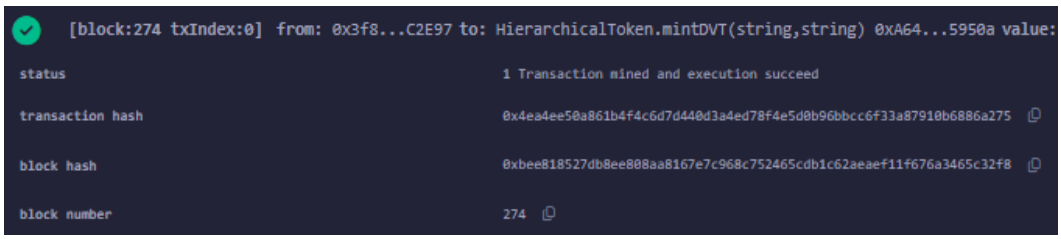
4.1.1. Organization Token minting (Figure 6).

(**a**)

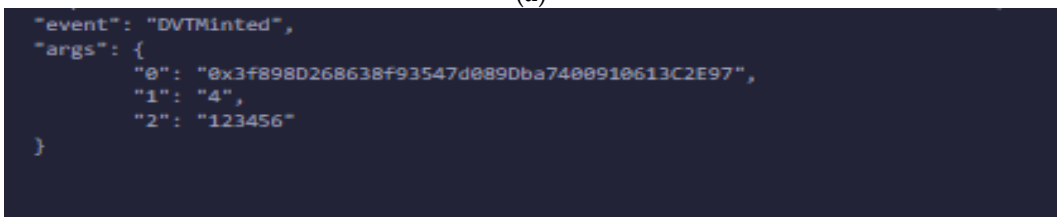
(**b**)

Figure 6 shows the successful execution of mintOrganizationToken. Panel (a) presents the transaction receipt for the OT with a specified orgId, confirming that the transaction was mined and gas was consumed as expected. Panel (b) shows the corresponding OrganizationMinted event, including the admin address and OT identifier, demonstrating that the organization's non-transferable anchor token was recorded on-chain and linked to its metadata

4.1.2. Document Verification Token minting (Figure 7).



(**a**)

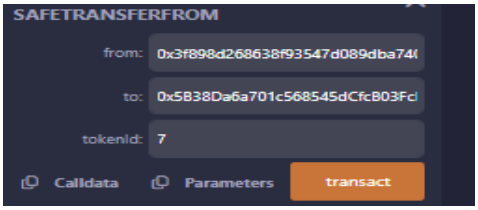


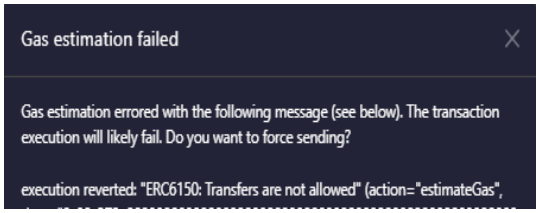
(**b**)

Figure 7. provides evidence of DVT creation under the previously minted OT. Panel (a) displays the transaction receipt for mintDVT, where a new DVT ID is auto-incremented and written to the ledger. Panel (b) shows the DVTMinted event, which logs the DVT ID and its parent OT ID, confirming that each document token is correctly attached to its issuing organization in the on-chain hierarchy.

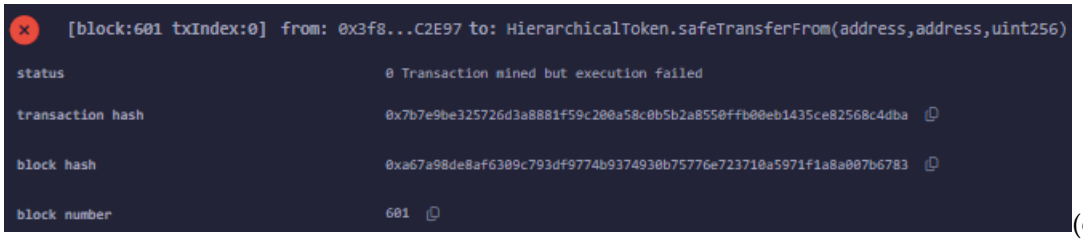
 doi:10.20944/preprints202602.1535.v1

4.1.3. Non-transferability enforcement(Figure 8).





(**a**) (**b**)



(**c**)

Figure 8. illustrates the enforcement of the soulbound policy for both OT and DVTs. Panel (a) shows the selection of an existing DVT ID; panel (b) displays Remix's "transaction likely to fail" estimation when a transfer is attempted; panel (c) shows the final transaction revert with the custom error message. Together, these panels confirm that token transfers and burns are blocked at the contract level, ensuring that issued financial documents remain permanently custodied by the organization.

4.1.4. Child token retrieval under the parent (Figure 9)

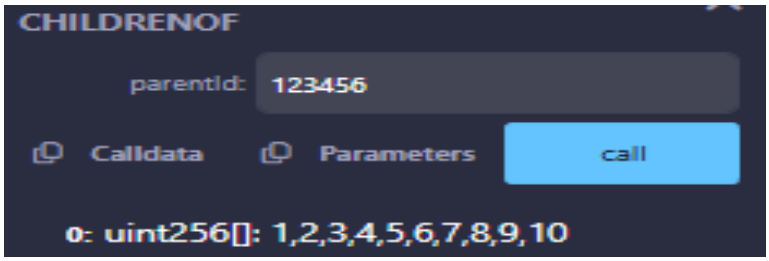
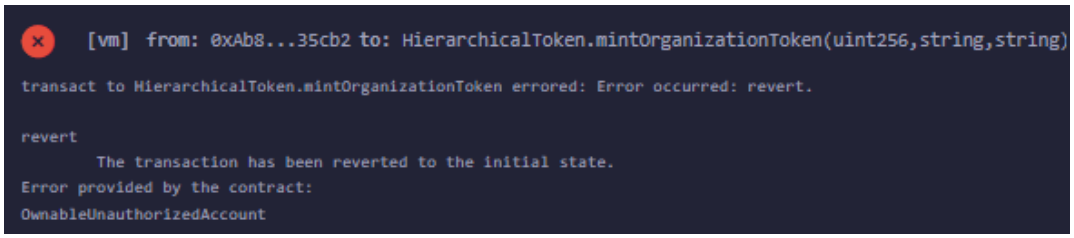


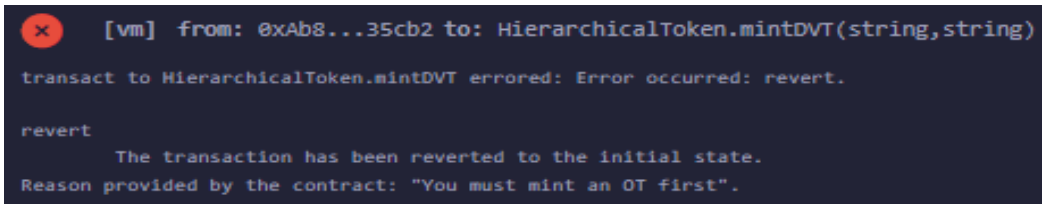
Figure 9 reports the result of querying the OT→DVT relationship using the read-only 'getDVTsByOrganization/childrenOf' call. The output lists the set of DVT IDs associated with a given OT (e.g., IDs 1–10 under OT 123456), demonstrating that the contract maintains a consistent parent–child structure and that verifiers or auditors can programmatically enumerate all document tokens issued by a particular financial institution.

4.1.5. Access control checks (Figure 10)

 doi:10.20944/preprints202602.1535.v1



(**a**)



(**b**)

Figure 10. summarizes access control behavior for OT and DVT minting. Panel (a) shows a failed OT mint attempt from a non-admin account, which is rejected by the *onlyOwner* modifier. Panel (b) shows a failed DVT mint attempt from an address that does not own an OT, confirming the requirement that only OT holders can issue child DVTs.

All these results align with the role-based restrictions specified in Section 3.2 and Algorithms 1–4, and demonstrate that unauthorized token issuance is blocked at the smart-contract layer.

4.2. Verifier User Interface

A lightweight web-based Verifier User Interface (UI) was implemented in HTML, JavaScript, and Ethers.js to realize the off-chain verification flow described in Section 3.6 and Algorithm 5. The UI connects to the read-only verifier node via JSON-RPC and encapsulates the required contract calls (getParentTokenId, getDVTMetadata) and hash comparison logic.

In a typical session, the verifier enters a DVT ID, and the UI automatically retrieves the parent OT ID and DVT metadata, fetches the referenced document from IPFS or HTTPS, recomputes its Keccak-256 hash in the browser, and displays an **Authentic / Not authentic** result together with the underlying hash values. This allows employers, auditors, and regulators to verify financial documents without interacting directly with the blockchain client or command-line tools.

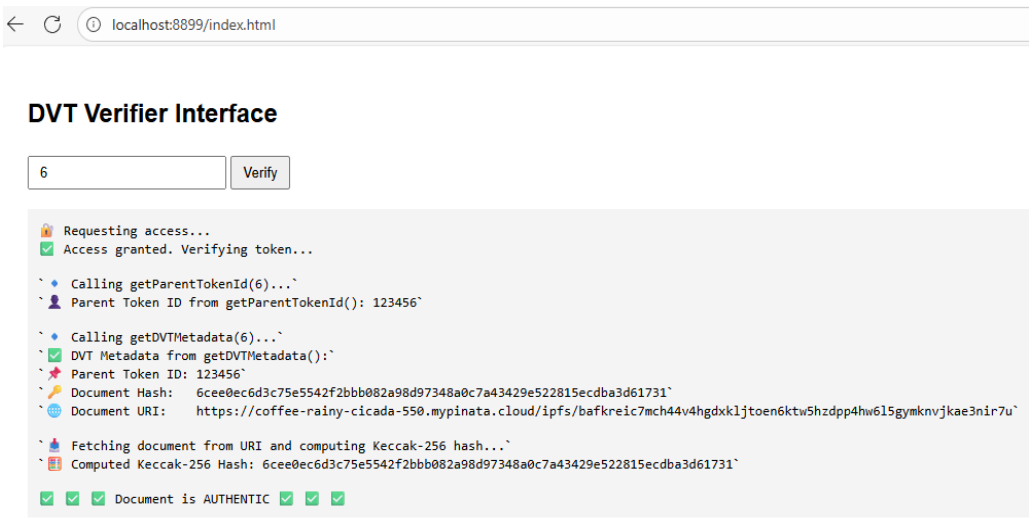


Figure 11. Verification simulation of a document using DVT ID 6 on the Verifier Interface.

4.3. Functional Evaluation

 doi:10.20944/preprints202602.1535.v1

This subsection evaluates whether the implemented system meets the core functional requirements derived from the model in Section 3. The focus is on correctness of behavior rather than throughput benchmarking. Each requirement is mapped to concrete evidence (figures, sections, or algorithms) and a qualitative pass/fail outcome.

Table 2. Functional requirements, evaluation evidence, and outcome.

ID	Functional Requirement	Evidence (section / figure /algorithm)	Outcome
R1	Only the designated organization node can mint an OT	Contract logic in Section 3.2.7; Algorithms 1 & 4; access-control results in Figure 10 (non-admin OT mint rejected)	Pass
R2	Only OT holders can mint DVTs under their own OT	Contract logic in Section 3.2.7; Algorithms 2 & 4; Figure 10 (failed DVT mint from non-OT address)	Pass
R3	OT and DVTs are non-transferable (soulbound semantics)	Non-transferability design in Section 3.2.6; Algorithms 1 & 2 ; Figure 8 (transfer attempt revert)	Pass
R4	Hierarchical OT→DVT linkage is correctly maintained and queryable	Hierarchy model in Section 3.2.4; Algorithm 3; Figure 9 (list of child DVT IDs under an OT)	Pass
R5	Each document is uniquely represented by its hash and URI	DVT metadata design in Section 3.2.3; Algorithm 2; contract checks on usedDocumentHashes and usedDocumentURIs	Pass
R6	Off-chain verification of documents without new blockchain transactions for verifiers	Verification workflow in Section 3.6; Algorithm 5; verifier UI in Section 4.2 and Figure 11	Pass
R7	Permissioned network with separated roles (organization, consensus-only nodes, verifier)	System architecture in Section 3.3; smart-contract execution evidence in Section 4.1	Pass
R8	Acceptable responsiveness for typical minting and verification operations	Qualitative observations from test runs in Section 4.1 and verifier UI interactions in Section 4.2	Pass

Overall, the implementation satisfied all functional requirements considered in this proof-of-concept. Role-based minting, enforced non-transferability, and consistent parent–child linkage were confirmed through on-chain events and failed transaction attempts. The verifier interface demonstrated that third parties can perform hash-based verification in real time against the permissioned Besu network without initiating new transactions, supporting the intended separation between issuance and verification.

4.4. Basic Security Considerations

The implementation incorporates several security measures aimed at preserving authenticity, integrity, and controlled access, consistent with the design in Section 3.

- **Layered access control -** Access restrictions are enforced at both the smart-contract level and the network level. Contract functions use role-based checks (e.g., *onlyOwner*, OT-ownership checks) to limit OT minting to a single authorized administrator address and DVT minting to the OT holder, while Besu's node- and account-level permissioning restricts who can connect to the network and submit transactions, respectively. This dual layer reduces the risk of unauthorized token issuance from both internal misuse and external intrusion.
- **Hash-based integrity protection -** Each document is anchored by a Keccak-256 hash stored in the DVT metadata. Verification recomputes the hash over the retrieved document bytes and compares it with the on-chain value (Algorithm 5). Any modification to the document content or substitution of a different file causes a hash mismatch, signaling loss of integrity.
- **Issuer-bound, non-transferable credentials -** OT and DVTs are intentionally non-transferable: transfer and approval functions revert, and ERC-5192-style "locked" semantics (soulbound token interface) are signaled. This prevents tokens from being sold, reassigned, or moved

 doi:10.20944/preprints202602.1535.v1

to other accounts, reducing impersonation risk and preserving a stable, auditable issuer–document relationship over time.

- **Controlled exposure of document data -** The blockchain only stores hashes and URIs; full document payloads remain off-chain (Section 3.2.5). This limits on-chain exposure of sensitive financial information and allows institutions to apply their own access controls at the storage layer (e.g., IPFS with pinning, or internal repositories), while still enabling authorized (semi-public) integrity checks.
- **Read-only verification path -** Verifiers interact with a dedicated read-only node that exposes JSON-RPC for contract queries but does not participate in consensus or minting. This reduces the attack surface on critical validator nodes and confines verifier activity to non-transactional reads.

These considerations provide a baseline security posture appropriate for a proof-of-concept. A more detailed threat model and controlled tampering experiment with mutable off-chain storage is shown in Section 4.5.

4.5. Tampering Scenario: Mutable Off-Chain Storage

To evaluate behavior under mutable storage, a PDF document as in the baseline scenario was hashed (Keccak-256) and its hash stored on-chain as part of the DVT metadata, while the file itself was hosted on Google Drive. After anchoring the hash, the PDF was edited and re-saved at the same Drive location, so the document URI remained unchanged.

During verification, the system fetched the file from the stored URI, recomputed its hash, and compared it to the on-chain value. As shown in Figure 12, the mismatch caused the verifier interface to report an authentication failure. This confirms that even subtle edits to a document on a mutable platform cannot pass verification: any change to the underlying bytes produces a hash mismatch. The experiment also illustrates that practical integrity depends on both immutable on-chain hashes and careful operational controls over off-chain storage.

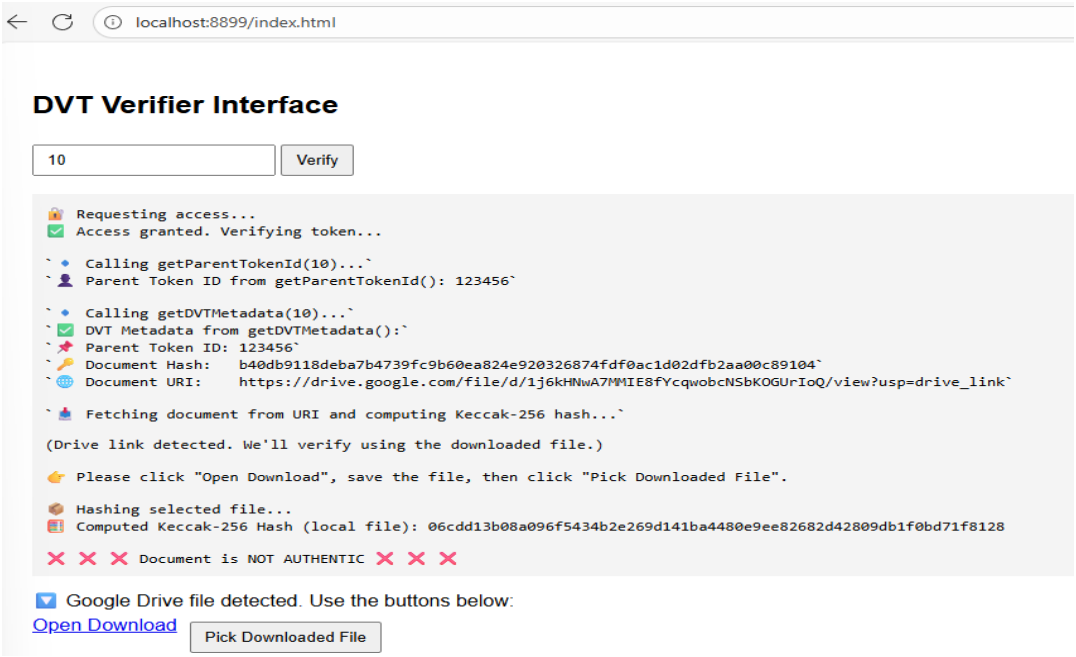


Figure 12. Verification failure due to hash mismatch after an update to a Google Drive–hosted PDF.

5. Discussion

5.1. Known Limitations

As a proof-of-concept, the implementation deliberately focuses on issuer-anchored integrity and provenance for high-value financial documents, rather than reproducing every advanced feature

 doi:10.20944/preprints202602.1535.v1

discussed in the literature. Accordingly, the prototype is intentionally scoped and has several limitations:

- **Controlled, small-scale environment -** All tests were conducted on a local permissioned Besu network with a modest number of nodes and tokens. The model has not yet been exercised in a multi-institution, production financial setting or anchored to a public chain.
- **No formal scalability or fault-tolerance evaluation -** Beyond qualitative responsiveness, there is no systematic benchmarking under high transaction volumes, adverse network conditions, or validator failures. Performance and resilience remain to be quantified separately.
- **No lifecycle controls for documents -** DVTs are permanently active in this prototype. The contract does not yet support explicit revocation, expiry, or versioning of documents, even though such mechanisms are important for financial records that may be corrected or rescinded. The prototype also does not track dynamic metadata changes or support "dynamic NFT" behavior; each token is treated as a static anchor for a single document version.
- **Off-chain storage assumptions -** The model relies on IPFS (with pinning) and, in one scenario, Google Drive. Persistence, retention, and version control are not enforced by the contract; they depend on institutional storage policies and external services.
- **Scope of document types –** The design targets high-value, relatively low-frequency financial documents (e.g., audited reports or formal statements), rather than everyday transactional records where per-document tokenization may be operationally excessive.
- **No built-in compliance or advanced identity layer -** The system assumes that compliant financial documents already exist and does not implement jurisdiction-specific regulation, automated document generation, or a full SSI/VC stack. These aspects are left for integration or future work.

These constraints are consistent with a proof-of-concept focused on integrity and provenance rather than full-scale deployment.

5.1. Positioning and Implications

A purely hash-based verification scheme can show that a file has not changed since its hash was first recorded, but it cannot guarantee that the initial hash corresponded to a legitimate financial document. A forged statement can be hashed once and then "verified" forever against that false baseline. What such schemes lack is institutional provenance—assurance that the first recorded hash was issued under the authority of a regulated institution.

The hierarchical model addresses this gap by introducing the Organization Token (OT) as an issuer-anchored root. The OT serves as a non-transferable, on-chain identity for the institution, and all DVTs must be minted as its children. Minting rights are tied to the issuer's address; as a result, each document hash is embedded in an auditable lineage of issuance, not treated as a free-floating checksum. For high-value financial documents—such as account statements, audited reports, loan approvals, or trade confirmations—this provides a stronger basis for trust than hash anchoring alone.

ERC-6150-style semantics make this linkage transparent and queryable. Auditors, regulators, counterparties, or employers can inspect parent–child relationships, enumerate all DVTs under a given OT, or confirm that a purported tokenized statement actually descends from the proper institutional root. Deploying this structure on a permissioned blockchain further reinforces trust by adding governed membership, accountable validators, and a ledger of actions aligned with institutional risk management. By relying on standard Ethereum-compatible interfaces, the model remains interoperable with existing toolchains and can be extended without altering the core ledger design.

The design also clarifies the relationship to other credential paradigms. While it shares non-transferability traits with Soulbound Token proposals, the binding here is deliberately asymmetric: the OT is bound to the issuer; DVTs are not yet bound to recipients' wallets. The focus is on authenticity and provenance of financial documents, not on self-custodied credentials. Likewise, the prototype does not implement an SSI framework (DID/VCs) or credentials wallets; instead, it offers an issuer-anchored ledger that could later be combined with such frameworks if desired.

 doi:10.20944/preprints202602.1535.v1

Finally, the current implementation uses a simplified two-tier hierarchy (issuer → documents). Real-world financial organizations often have multi-level business units and approval chains. The same hierarchical logic can be extended to deeper trees and delegated minting rights—e.g., business-line-level issuance under a group-level OT, or policies that apply across sets of documents. These richer structures are beyond the scope of the present prototype but align naturally with the model's intent.

Together, these results indicate that the proposed model delivers issuer-anchored provenance, tamper-evident verification, and gas-free validation for high-value financial documents in a permissioned setting.

6. Conclusion and Future Work

6.1. Summary of Findings

This study proposed and implemented a hierarchical tokenization model for authenticating high-value financial documents on a permissioned Ethereum-compatible blockchain. An issuer-anchored OT represents the institution; non-transferable DVTs represent individual documents and carry cryptographic hashes and references to off-chain storage.

The main findings are:

- The prototype successfully bound documents to issuers via an OT–DVT parent–child structure, with the hierarchy behaving as intended on a live Besu Clique network.
- Hash-based verification with off-chain storage detected tampering: when document bytes were modified, the recomputed hash no longer matched the on-chain value and verification failed.
- Role-based minting and a verifier UI operated correctly in practice: only the organization admin could mint an OT and DVTs, while external verifiers used a read-only node and web interface to perform decentralized, transaction-free checks.
- A permissioned Hyperledger Besu network proved operationally suitable for this use case, offering governance, configurable PoA consensus, and predictable behavior under the tested conditions.
- In a mutable storage scenario (Google Drive-hosted PDF updated without link change), the system correctly flagged a mismatch between the recomputed and stored hash, demonstrating robustness when off-chain locations are editable.

Taken as a whole, these results show that a hierarchical, issuer-anchored token model on a permissioned Ethereum client can provide strong integrity and provenance guarantees for high-value financial documents, while keeping verification lightweight for external stakeholders.

6.1. Future Work

Future work will focus on deepening, rather than radically changing, the proposed model:

- **Token lifecycle controls -** Introduce explicit mechanisms for revocation, expiry, and controlled updates of DVTs, allowing institutions to reflect the full lifecycle of financial documents (e.g., corrected statements, rescinded approvals) while preserving audit trails.
- **Richer use of hierarchy -** Extend the issuer-anchored structure beyond a simple two-tier model to support more nuanced organizational and document groupings (for example, business units or document sets), while leaving detailed decomposition strategies and policies to later work.
- **Integration with identity frameworks -** Explore optional integration with decentralized identity or Verifiable Credentials, where hierarchical tokens continue to guarantee issuer authenticity and integrity anchoring, and an external identity layer handles recipient and verifier identities in a privacy-aware manner, including the design of explicit acceptance or opt-out flows if DVTs are later bound to recipients' wallets, similar in spirit to SBT-style accept/reject patterns.
- **Privacy-preserving verification -** Explore lightweight zero-knowledge or related techniques that would allow verifiers to confirm authenticity or selected attributes of financial

 doi:10.20944/preprints202602.1535.v1

documents without revealing full content or sensitive metadata, while keeping the core issuer-anchored model unchanged.

- **Performance and deployment patterns -** Conduct a separate evaluation of performance characteristics and deployment options for larger consortia, without committing to specific consensus experiments or parameter choices in this paper.
- **Application-layer extensions -** Investigate controlled forms of metadata-driven or template-based document handling, including dynamic metadata generation for documents whose content is derived from on-chain state, without turning the blockchain into a document store, and examine how metadata sensitivity and regulatory and legal requirements in specific financial jurisdictions can be layered onto the framework.

These directions preserve the core contribution—issuer-anchored, hierarchical tokenization for financial documents on a permissioned chain—while leaving room to refine lifecycle management, identity, and performance in subsequent work and follow-on studies.

Supplementary Materials: The following supporting information can be downloaded at the website of this paper posted on Preprints.org. Supplementary File S1 (ZIP) contains the smart contract source code (Solidity); verifier web interface (HTML/JavaScript with ethers.js, including hashing logic); Python hashing script; sample PDF documents (original and altered); Docker Compose and blockchain configuration files (genesis and permissioning configuration); and a README with reproduction steps.

Author Contributions: Conceptualization, C.I. and S.-C.H.; methodology, C.I. and B.N.; software, C.I.; validation, C.I., S.-C.H. and B.N.; formal analysis, C.I.; investigation, C.I.; resources, C.I., S.-C.H. and B.N.; data curation, C.I.; writing—original draft preparation, C.I.; writing—review and editing, C.I., S.-C.H. and B.N.; visualization, C.I.; supervision, S.-C.H. and B.N.; project administration, S.-C.H. and B.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data and code supporting the reported results are provided in Supplementary File S1.

Acknowledgments: During the preparation of this manuscript, the authors used generative artificial intelligence (ChatGPT-5, OpenAI) as an interactive technical assistant for drafting support, language refinement, and exploratory code prototyping under author supervision. All implementation, testing, validation, and scholarly analysis were conducted by the authors, who take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Afrakhteh, M., Ibrahim, S., & Salleh, M. (2010). Printed Document Authentication Using Watermarking Technique. 2010 Second International Conference on Computational Intelligence, Modelling and Simulation, 367–370. https://doi.org/10.1109/CIMSiM.2010.70
2. Al-Ameri, M. A. A., Mahmood, B., Ciylan, B., & Amged, A. (2023). Unsupervised Forgery Detection of Documents: A Network-Inspired Approach. Electronics, 12(7), 1682. https://doi.org/10.3390/electronics12071682
3. Ali, V. E., Asika, M. O., Elebesunu, E. E., Agbo, C., & Antwi, M. H. (2024). Cognizance and mitigation of falsified immunization documentation: Analyzing the consequences for public health in Nigeria, with a focus on counterfeited COVID-19 vaccination cards: A case report. Health Science Reports, 7(2), e1885. https://doi.org/10.1002/hsr2.1885
4. Anwar, M. J., & Gill, A. Q. (2025). NFTs enabled federated digital identity data representation and management. Discover Data, 3(1), 19. https://doi.org/10.1007/s44248-025-00058-y
5. Artha, K. A. R., Zain, S. N., Alkautsar, A. A., & Widianto, M. H. (2022). Implementation of Smart Contracts for E-Certificate as Non-Fungible Token using Solana Network. Proceedings of the 2022 IEEE 7th

 doi:10.20944/preprints202602.1535.v1

International Conference on Information Technology and Digital Applications, ICITDA 2022. https://doi.org/10.1109/ICITDA55840.2022.9971423

6. Baechler, S. (2020). Document Fraud: Will Your Identity Be Secure in the Twenty-first Century? European Journal on Criminal Policy and Research, 26(3), 379–398. https://doi.org/10.1007/s10610-020-09441-8
7. Banaeian Far, S., & Hosseini Bamakan, S. M. (2023). NFT-based identity management in metaverses: challenges and opportunities. SN Applied Sciences, 5(10), 260. https://doi.org/10.1007/s42452-023-05487-5
8. Bhagat, N., Bae, J., & Lee, S.-H. (2025). Dynamic Management of Hierarchical NFTs: Efficient Splitting and Merging. Journal of the Korea Society of Computer and Information, 30(2), 73–82. https://doi.org/10.9708/jksci.2025.30.02.073
9. Blockcerts. (n.d.). Blockcerts: The open standard for blockchain credentials. Retrieved November 20, 2025, from https://www.blockcerts.org
10. Böhmecke-Schwafert, M. (2024). The role of blockchain for trade in global value chains: A systematic literature review and guidance for future research. Telecommunications Policy, 48(9), 102835. https://doi.org/10.1016/j.telpol.2024.102835
11. ConsenSys, Inc. (2024). ConsenSys GoQuorum documentation. https://docs.goquorum.consensys.io/
12. Devlin, C., Chadwick, S., Moret, S., Baechler, S., Rossy, Q., & Morelato, M. (2024). Illuminating the dark web market of fraudulent identity documents and personal information: An international and Australian perspective. Forensic Science International, 363, 112203. https://doi.org/10.1016/j.forsciint.2024.112203
13. Eltuhami, M., Abdullah, M., & Talip, B. A. (2022). Identity Verification and Document Traceability in Digital Identity Systems using Non-Transferable Non-Fungible Tokens. 2022 International Visualization, Informatics and Technology Conference (IVIT), 136–142. https://doi.org/10.1109/IVIT55443.2022.10033362
14. Ethereum Foundation. (2024, June 16). Private networks. Go-Ethereum Documentation. https://geth.ethereum.org/docs/fundamentals/private-network
15. Faccia, A., Pandey, V., & Banga, C. (2022). Is Permissioned Blockchain the Key to Support the External Audit Shift to Entirely Open Innovation Paradigm? Journal of Open Innovation: Technology, Market, and Complexity, 8(2), 85. https://doi.org/10.3390/joitmc8020085
16. Gebreab, S. A., Salah, K., Jayaraman, R., & Zemerly, J. (2023). Trusted Traceability and Certification of Refurbished Medical Devices Using Dynamic Composable NFTs. IEEE Access, 11, 30373–30389. https://doi.org/10.1109/ACCESS.2023.3261555
17. Georgiou, I., Sapuric, S., Lois, P., & Thrassou, A. (2024). Blockchain for Accounting and Auditing—Accounting and Auditing for Cryptocurrencies: A Systematic Literature Review and Future Research Directions. Journal of Risk and Financial Management, 17(7), 276. https://doi.org/10.3390/jrfm17070276
18. Guidi, B., & Michienzi, A. (2023). From NFT 1.0 to NFT 2.0: A Review of the Evolution of Non-Fungible Tokens. Future Internet, 15(6), 189. https://doi.org/10.3390/fi15060189
19. Hammi, B., Zeadally, S., & Perez, A. J. (2023). Non-Fungible Tokens: A Review. IEEE Internet of Things Magazine, 6(1), 46–50. https://doi.org/10.1109/IOTM.001.2200244
20. Hasan, H. R., Madine, M., Musamih, A., Jayaraman, R., Salah, K., Yaqoob, I., & Omar, M. (2024). Non-fungible tokens (NFTs) for digital twins in the industrial metaverse: Overview, use cases, and open challenges. Computers & Industrial Engineering, 193, 110315. https://doi.org/10.1016/j.cie.2024.110315
21. Hyperledger Foundation. (2023). Hyperledger Fabric documentation (release 2.5). https://hyperledger-fabric.readthedocs.io/en/release-2.5/
22. Hyperledger Foundation. (2025, December 19). Besu for private (permissioned) networks. https://besu.hyperledger.org/private-networks
23. Jirgensons, M., & Kapenieks, J. (2018). Blockchain and the Future of Digital Learning Credential Assessment and Management. Journal of Teacher Education for Sustainability, 20(1), 145–156. https://doi.org/10.2478/jtes-2018-0009
24. Khan, M. M., Khan, F. S., Nadeem, M., Khan, T. H., Haider, S., & Daas, D. (2025). Scalability and Efficiency Analysis of Hyperledger Fabric and Private Ethereum in Smart Contract Execution. Computers, 14(4), 132. https://doi.org/10.3390/computers14040132
25. Kim, G., & Ryou, J. (2023). Digital Authentication System in Avatar Using DID and SBT. Mathematics, 11(20), 4387. https://doi.org/10.3390/math11204387

 doi:10.20944/preprints202602.1535.v1

26. Kumar, N. N., Kumar, R. S., Basale, R. R., & Saffath, M. (2022). Decentralized Storage Of Educational Assets Using NFTs And Blockchain Technology. 2022 International Conference on Smart Systems and Inventive Technology (ICSSIT), 260–266. https://doi.org/10.1109/icssit53264.2022.9716362
27. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Shevchuk, R., & Karpinski, M. (2024). NFT Technology for Enhanced Global Digital Registers: A Novel Approach to Tokenization. Future Internet, 16(7), 252. https://doi.org/10.3390/fi16070252
28. Lee, H., & Yeon, C. (2021). Blockchain-Based Traceability for Anti-Counterfeit in Cross-Border E-Commerce Transactions. Sustainability, 13(19), 11057. https://doi.org/10.3390/su131911057
29. Lee, K., msfew, Kartin, & qizhou. (2022, December 15). ERC-6150: Hierarchical NFTs. Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-6150
30. Lockyer, M., Mudge, N., Schalm, J., Echeverry, S., & Zhou, Z. V. (2018, July 7). ERC-998: Composable non-fungible token. Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-998
31. López-Pimentel, J. C., Gonzalez-Sanchez, J., & Morales-Rosales, L. A. (2025). A Digital Identity Blockchain Ecosystem: Linking Government-Certified and Uncertified Tokenized Objects. Applied Sciences, 15(15), 8577. https://doi.org/10.3390/app15158577
32. Lunesu, M. I., Tonelli, R., Pinna, A., & Sansoni, S. (2023). Soulbound Token for Covid-19 Vaccination Certification. 2023 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops), 243–248. https://doi.org/10.1109/PerComWorkshops56833.2023.10150304
33. Mohammad Saeidia, F., Zahedi, M. H., & Farahani, E. (2025). A Secure and Reliable Model for Financial Documents Using Digital Signature and Blockchain Technology. AI and Tech in Behavioral and Social Sciences, 3(1), 23–33. https://doi.org/10.61838/kman.aitech.3.1.3
34. Mohsin Arkah, Z., Alzubaidi, L., Ali, A. A., & Abdulameer, A. T. (2020). Digital Color Documents Authentication Using QR Code Based on Digital Watermarking. In Advances in Intelligent Systems and Computing (Vol. 940, pp. 1093–1101). Springer Verlag. https://doi.org/10.1007/978-3-030-16657-1_102
35. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf
36. Nita, S. L., & Mihailescu, M. I. (2024). A Novel Authentication Scheme Based on Verifiable Credentials Using Digital Identity in the Context of Web 3.0. Electronics, 13(6), 1137. https://doi.org/10.3390/electronics13061137
37. Weyl, E. G., Ohlhaver, P., & Buterin, V. (2022). Decentralized Society: Finding Web3's Soul. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4105763
38. Ongwook Bae, J., Bhagat, N., Lee, S.-H., & Bae, J. (2024). Hierarchical NFT using Parent-Child Structure. Journal of The Korea Society of Computer and Information, 29(2), 127–136. https://doi.org/10.9708/jksci.2024.29.02.127
39. Patel, A., Sai, S., Daiya, A., Akolekar, H., & Chamola, V. (2025). Blockchain enabled traceability in the jewel supply chain. Scientific Reports, 15(1), 3837. https://doi.org/10.1038/s41598-025-88245-4
40. Peelam, M. S., Chamola, V., Sharma, A. K., & Chaurasia, B. K. (2025). Decentralized Trust: NFT and Blockchain-Enabled Evidence System using Fog Computing. Blockchain: Research and Applications, 100321. https://doi.org/10.1016/j.bcra.2025.100321
41. Pericàs-Gornals, R., Mut-Puigserver, M., Payeras-Capellá, M. M., Cabot-Nadal, M. Á., & Ramis-Bibiloni, J. (2024). Digital credentials management system using rejectable soulbound tokens. Annals of Telecommunications, 79(11–12), 843–855. https://doi.org/10.1007/s12243-024-01032-6
42. Pierro, G. A., Cocco, L., & Tonelli, R. (2024). Besu vs. Quorum: Comparative analysis in the context of simulated energy communities. In M. Bartoletti, C. Schifanella, & A. Vitaletti (Eds.), Proceedings of the Sixth Distributed Ledger Technology Workshop (DLT 2024) (CEUR Workshop Proceedings, Vol. 3791) (pp. 1–16). CEUR-WS.org. https://ceur-ws.org/Vol-3791/paper24.pdf
43. Podda, E., Hölzmer, P., Amard, A., Sedlmeir, J., & Fridgen, G. (2025). The impact of zero-knowledge proofs on data minimisation compliance of digital identity wallets. Internet Policy Review, 14(3). https://doi.org/10.14763/2025.3.2019

 doi:10.20944/preprints202602.1535.v1

44. Precht, H., Hüllmann, J. A., & Marx Gómez, J. (2026). Paperless Everything: A Systematic Literature Review for the Design of Blockchain-based Document Management Systems. Distributed Ledger Technologies: Research and Practice, 5(2), 1–48. https://doi.org/10.1145/3737296
45. Pu, S., & Lam, J. S. L. (2023). The benefits of blockchain for digital certificates: A multiple case study analysis. Technology in Society, 72, 102176. https://doi.org/10.1016/j.techsoc.2022.102176
46. Cardenas-Quispe, M. A., & Pacheco, A. (2025). Blockchain ensuring academic integrity with a degree verification prototype. Scientific Reports, 15(1), 9281. https://doi.org/10.1038/s41598-025-93913-6
47. Ramadhan, M. R., Mandala, S., & Yulianto, F. A. (2023). Analysis and Implementation of Digital Signature Algorithm in PDF Document. 2023 11th International Conference on Information and Communication Technology (ICoICT), 2023-August, 11–16. https://doi.org/10.1109/ICoICT58202.2023.10262708
48. Ramirez Lopez, L. J., & Morillo Ledezma, G. G. (2025). Employing Blockchain, NFTs, and Digital Certificates for Unparalleled Authenticity and Data Protection in Source Code: A Systematic Review. Computers, 14(4), 131. https://doi.org/10.3390/computers14040131
49. Razi, Q., Devrani, A., Abhyankar, H., Chalapathi, G. S. S., Hassija, V., & Guizani, M. (2024). Non-Fungible Tokens (NFTs)—Survey of Current Applications, Evolution, and Future Directions. IEEE Open Journal of the Communications Society, 5, 2765–2791. https://doi.org/10.1109/OJCOMS.2023.3343926
50. Satybaldy, A., Subedi, A., & Nowostawski, M. (2022). A Framework for Online Document Verification Using Self-Sovereign Identity Technology. Sensors, 22(21), 8408. https://doi.org/10.3390/s22218408
51. Silaghi, D. L., & Popescu, D. E. (2025). A Systematic Review of Blockchain-Based Initiatives in Comparison to Best Practices Used in Higher Education Institutions. Computers, 14(4), 141. https://doi.org/10.3390/computers14040141
52. Singh, P., Sagar, S., Singh, S., Alshahrani, H. M., Getahun, M., & Soufiene, B. O. (2024). Blockchain-enabled verification of medical records using soul-bound tokens and cloud computing. Scientific Reports, 14(1), 24830. https://doi.org/10.1038/s41598-024-75708-3
53. Subramanya, S. R., & Yi, B. K. (2006). Digital signatures. IEEE Potentials, 25(2), 5–8. https://doi.org/10.1109/MP.2006.1649003
54. Susik, R., Nowotniak, R., & Kulczycki, E. (2023). Blockchain-based certification of research outputs and academic achievements: A case of scientific conference. Communication Papers of the 18th Conference on Computer Science and Intelligence Systems, 37, 329–333. https://doi.org/10.15439/2023F9620
55. Szyjewski, G. (2023). Securing Digital Copies of the Documents to Ensure Documents' Integrity. EUROPEAN RESEARCH STUDIES JOURNAL, XXVI(Issue 4), 718–726. https://doi.org/10.35808/ersj/3321
56. Teraura, N., Echizen, I., & Iwamura, K. (2020). A QR Symbol with ECDSA for Both Public and Secret Areas using Rhombic Sub-cells. 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 1392–1399.
57. Usha, B. A., Monish, S., Murali Manohara Hegde, A. S., Kumar, N., Aditya, P., & Manjunath, A. (2023). Blockchain Technology in Document Authentication: A Comprehensive Literature Review. 2023 4th International Conference on Communication, Computing and Industry 6.0 (C216), 1–5. https://doi.org/10.1109/C2I659362.2023.10431235
58. Walidaniy, W. D., Yuliana, M., & Darwito, H. A. (2023). Enhancing Document Authenticity with QR Codes and ECC-Based Digital Signatures. 2023 International Electronics Symposium (IES), 238–243. https://doi.org/10.1109/IES59143.2023.10242576
59. Wellem, T., Nataliani, Y., & Iriani, A. (2022). Academic Document Authentication using Elliptic Curve Digital Signature Algorithm and QR Code. JOIV : International Journal on Informatics Visualization, 6(3), 667. https://doi.org/10.30630/joiv.6.2.872
60. Yerpude, S., Sood, K., & Grima, S. (2022). Blockchain-Augmented Digital Supply Chain Management: A Way to Sustainable Business. Journal of Risk and Financial Management, 16(1), 7. https://doi.org/10.3390/jrfm16010007
61. Zhai, X., Pang, S., Wang, M., Qiao, S., & Lv, Z. (2023). TVS: a trusted verification scheme for office documents based on blockchain. Complex & Intelligent Systems, 9(3), 2865–2877. https://doi.org/10.1007/s40747-021-00617-1

 doi:10.20944/preprints202602.1535.v1