Authentico: A Blockchain-Powered Document Verification Platform

Date: July 5, 2025

Abstract

Authentico is a pioneering document verification platform that leverages blockchain technology to establish a secure, transparent, and immutable system for managing and verifying digital documents. By eliminating reliance on centralized authorities, Authentico significantly reduces fraud, streamlines verification processes, and enhances document privacy through end-to-end encryption. This whitepaper details the platform's core functionalities, technical architecture, security measures, market relevance, and future vision.

1. Introduction

In an increasingly digital world, the integrity and authenticity of documents are paramount. However, traditional paper-based and centralized digital verification systems are plagued by inefficiencies, susceptibility to fraud, and a lack of transparency. Manual processes lead to significant delays, while centralized databases are vulnerable to data breaches and single points of failure.

Authentico addresses these critical challenges by harnessing the power of blockchain technology. Our platform provides a trustless, decentralized solution that ensures the immutability and verifiability of documents, offering a robust alternative to conventional methods.

2. Problem Statement

The current landscape of document verification is fraught with significant pain points:

- **Slow, Manual Verification Processes:** As highlighted by the case of a small business owner in Francistown, Botswana, manual processing of licenses and documents can take weeks, leading to missed opportunities and damaged credibility. This paper-heavy approach stifles growth and efficiency.
- **Fraudulent Certificates & Forged Documents:** The rise of digital documents has unfortunately been accompanied by an increase in fraudulent certificates and forged documents. A 2025 Smile ID report indicated a jump in Southern Africa's document fraud rejection rates from 9% in 2023 to 21% in 2024, costing businesses billions annually.
- **Lost or Inaccessible Records:** Centralized storage systems are prone to data

loss, corruption, or inaccessibility due to system failures or malicious attacks, leading to significant operational disruptions.
- **Lack of Digital Trust:** The absence of a universally trusted and transparent mechanism for digital document verification erodes confidence in online interactions and transactions.

3. Solution: Authentico's Vision and Mission

Authentico's core objective is to revolutionize document verification by establishing a secure and transparent trust layer using blockchain technology.

Vision: To foster a global digital ecosystem where trust is the foundation of every interaction.

Mission: To revolutionize document verification by empowering individuals and organizations worldwide with secure, efficient, and user-centric blockchain solutions.

Authentico achieves this by:

- **Reducing verification time from weeks to seconds:** By digitizing and decentralizing the process, Authentico eliminates bottlenecks.
- **Streamlining organization workflows:** Verified organizations can efficiently review and approve documents.
- **Giving users full control over their credentials:** Users manage their encrypted documents and sharing permissions.
- **Safely preserving documents:** Immutable blockchain records and decentralized storage ensure document longevity and integrity.

4. Key Features and Value Proposition

Authentico offers a comprehensive suite of features designed to provide a secure, efficient, and user-friendly document verification experience:

- **Secure Document Upload:** Users can upload documents that are encrypted using AES-256 before being stored on IPFS via Pinata. This ensures document privacy and data integrity.
- **Blockchain Anchoring:** Document metadata and unique hashes are immutably anchored on the Ethereum Sepolia testnet. This provides an unalterable record of the document's existence and its verification status.
- **Document Verification:** Verified organizations can securely review and verify documents, with the verification status recorded on-chain.
- **Organization Verification:** Organizations can apply for and achieve a verified status, enabling them to act as official document verifiers within the Authentico

network.
- **Secure Document Sharing:** Users can share verified documents via secure links and QR codes, allowing recipients to verify the document's authenticity without direct access to the encrypted content.
- **Document Viewing:** Authorized users can securely view encrypted documents, with the platform handling decryption on-the-fly.
- **Intuitive Dashboard:** Provides a user-friendly interface for individuals and organizations to manage their documents and verification requests.
- **Advanced Document Lookup:** Enables efficient searching and retrieval of verified documents.
- **Analytics for Organizational Insight:** Provides organizations with data on verification trends and activities.

5. Technical Architecture

Authentico is built as a monorepo leveraging npm workspaces, comprising three core components: a Next.js 14 frontend, a Node.js/Express backend, and Ethereum smart contracts.

5.1 System Architecture

The platform's architecture is designed for scalability, security, and modularity:

```
graph TD
  subgraph "Frontend (Next.js)"
    A[Landing Page] --> B[Authentication]
    B --> C[Individual Dashboard]
    B --> D[Organization Dashboard]
    B --> E[Admin Dashboard]
    C --> F[Document Upload]
    C --> G[Document Viewing]
    C --> H[Document Sharing]
    D --> I[Verification Queue]
    D --> J[Organization Profile]
    E --> K[User Management]
    E --> L[Organization Applications]
  end

  subgraph "Backend (Node.js/Express)"
    M[Auth Routes] --> N[Firebase Admin]
    O[Document Routes] --> P[Encryption Service]
```

```
    O --> Q[Storage Service]
    O --> R[Blockchain Service]
    S[Organization Routes] --> N
    S --> R
  end

  subgraph "External Services"
    T[Firebase Auth]
    U[Firebase Firestore]
    V[Pinata/IPFS]
    W[Ethereum Blockchain]
  end

  subgraph "Smart Contracts"
    X[DocumentNFT Contract]
  end

  B <--> M
  F --> O
  I --> O
  L --> S
  N <--> T
  N <--> U
  P --> Q
  Q <--> V
  R <--> X
  X <--> W
```

5.2 Key Components

- **Frontend (Next.js 14 with App Router):**
  - Built with React and TypeScript, providing a responsive and dynamic user interface.
  - Utilizes **Thirdweb SDK** for seamless wallet connection (social login and embedded wallets) and blockchain interactions.
  - Integrates **Firebase Client SDK** for user authentication and real-time database access (Firestore).
  - Styled with **Tailwind CSS** for a modern, neubrutalism design theme, enhanced with **Framer Motion** for animations.

- **Backend (Node.js/Express):**
  - A robust API service handling core business logic.
  - Manages **Authentication** using Firebase Admin SDK.
  - Facilitates **Document Processing**, including encryption, secure IPFS storage via Pinata, and blockchain anchoring.
  - Oversees **Organization Management**, handling verification applications and approvals.
  - Manages **Blockchain Interactions** using ethers.js and a dedicated sponsor wallet for transaction signing.
- **Smart Contracts (Solidity):**
  - Deployed on the **Ethereum Sepolia testnet**.
  - The DocumentNFT contract represents verified documents as non-fungible tokens, recording verification status and managing ownership.
  - Ensures immutable and transparent record-keeping of document authenticity.

5.3 Third-Party Services

Authentico's robust functionality is powered by strategic integrations with leading third-party services:

- **Firebase:**
  - **Authentication:** For secure user registration and login.
  - **Firestore:** A NoSQL cloud database for storing user profiles, document metadata (excluding sensitive content), and organization details.
  - **Security Rules:** Enforces granular access control for data stored in Firestore.
- **Pinata/IPFS:**
  - **Document Storage:** Encrypted documents are stored on the InterPlanetary File System (IPFS) via Pinata, ensuring decentralized and censorship-resistant storage.
  - **Pinning Service:** Pinata's service ensures that documents remain persistently available on the IPFS network.
- **Thirdweb:**
  - **Wallet Connection:** Simplifies the process of connecting user wallets, supporting various Web3 wallets and social logins.
  - **Blockchain Interaction:** Provides a streamlined interface for interacting with blockchain operations.
- **Ethereum Blockchain (Sepolia Testnet):**
  - **Document Anchoring:** Serves as the immutable ledger for recording document hashes and metadata.
  - **Verification Status:** Provides an unalterable, publicly verifiable record of

document verification.

6. Security Features

Authentico is built with a security-first mindset, incorporating multiple layers of protection to ensure the integrity, confidentiality, and availability of documents and user data.

6.1 Encryption

- **AES-256-GCM Encryption:** All documents are encrypted using the Advanced Encryption Standard with 256-bit keys in Galois/Counter Mode (GCM) before being uploaded to IPFS. This provides strong confidentiality and authenticated encryption.
- **Secure Key Management:**
  - An envelope encryption scheme is employed, where each document is encrypted with a unique Data Encryption Key (DEK).
  - The DEK itself is then encrypted with a master key.
  - While the current implementation uses a master key derived from an environment variable, a production-grade solution would integrate with a Key Management Service (KMS) for enhanced security and key rotation policies.
- **End-to-End Encryption:** Documents remain encrypted throughout their lifecycle, from upload to storage and retrieval, ensuring that only authorized users with the correct decryption keys can access the original content.

6.2 Blockchain Security

- **Immutable Records:** Document hashes and metadata are permanently recorded on the Ethereum blockchain, making them tamper-proof and providing undeniable proof of existence and integrity.
- **Verification Status:** The verification status of each document is recorded on-chain, providing a transparent and auditable history.
- **Transaction Signing:** All blockchain transactions are securely signed using a dedicated sponsor wallet, ensuring authenticity and preventing unauthorized modifications.
- **Smart Contract Security:** Smart contracts are designed with security best practices, including reentrancy guards, and are intended for thorough auditing.

6.3 API Security

- **JWT Authentication:** All API endpoints are secured using JSON Web Tokens (JWTs) for user authentication and authorization. Tokens are validated on both the frontend and backend.

- **Role-Based Access Control (RBAC):** The platform implements a robust RBAC system with Individual, Organization, and Admin roles. Access to specific routes and functionalities is strictly enforced based on the user's assigned role.
- **Rate Limiting:** API endpoints are protected against abuse and Denial-of-Service (DoS) attacks through rate limiting mechanisms.
- **Input Validation:** All user inputs are rigorously validated on both the frontend and backend to prevent common web vulnerabilities such as injection attacks (e.g., SQL injection, NoSQL injection) and cross-site scripting (XSS).
- **Session Management:** Session management utilizes HTTP-only cookies for token storage and session fingerprinting to protect against session hijacking.

7. User Flows

7.1 User Document Upload Process

1. **User Authentication:** User connects their wallet (e.g., MetaMask) via Thirdweb, establishing authentication with Firebase.
2. **Document Selection & Metadata:** User selects a document file and provides essential metadata (name, type). They also select a verified organization responsible for verification.
3. **Upload & Processing:** The frontend securely transmits the document to the backend. The backend:
    - Calculates the original document's SHA-256 hash.
    - Generates a unique Data Encryption Key (DEK).
    - Encrypts the DEK with the master key.
    - Encrypts the document with the DEK.
    - Uploads the encrypted document to IPFS via Pinata, obtaining an IPFS Content Identifier (CID).
    - Stores document metadata, encrypted DEK, and IPFS CID in Firestore.
4. **Blockchain Anchoring:** The backend asynchronously mints an NFT on the Ethereum blockchain with the document's hash and metadata. The document's status is updated to "Pending Verification," and the chosen verifying organization is notified.

```
sequenceDiagram
    participant User
    participant Frontend
    participant Backend
    participant Firebase
    participant Pinata
    participant Blockchain
```

```
    User->>Frontend: Upload document & select verifier
    Frontend->>Backend: POST /api/documents/upload
    Backend->>Backend: Generate hash & encrypt document
    Backend->>Pinata: Upload encrypted document
    Pinata-->>Backend: Return IPFS CID
    Backend->>Firebase: Store document metadata
    Firebase-->>Backend: Confirm storage
    Backend-->>Frontend: Return initial success
    Frontend-->>User: Show "Processing" status
    Backend->>Blockchain: Mint document NFT
    Blockchain-->>Backend: Return transaction hash
    Backend->>Firebase: Update with blockchain details
    Backend->>Firebase: Create notification for verifier
    Firebase-->>User: Status update notification
```

7.2 Document Verification Process

1. **Organization Dashboard:** A verified organization logs in and accesses their dashboard to view pending verification requests.
2. **Document Review:** The organization selects a document. The backend securely retrieves the encrypted document from IPFS, decrypts it using the DEK (which is first decrypted with the master key), and streams the decrypted content to the organization for review.
3. **Verification Decision:** The organization approves or rejects the document, providing a reason for rejection if applicable.
4. **Blockchain Confirmation:** The backend updates the document's status in Firestore and calls the smart contract to record the verification decision on the blockchain. The document owner receives a notification of the verification result.

```
sequenceDiagram
    participant Organization
    participant Frontend
    participant Backend
    participant Firebase
    participant Pinata
    participant Blockchain

    Organization->>Frontend: Access verification queue
```

```
    Frontend->>Backend: GET /api/documents/pending
    Backend->>Firebase: Query pending documents
    Firebase-->>Backend: Return pending documents
    Backend-->>Frontend: Display document list
    Organization->>Frontend: Select document to review
    Frontend->>Backend: GET /api/documents/:id/secure-details
    Backend->>Firebase: Get document metadata
    Firebase-->>Backend: Return metadata with encrypted DEK
    Backend->>Backend: Decrypt DEK with master key
    Backend->>Pinata: Retrieve encrypted document
    Pinata-->>Backend: Return encrypted document
    Backend->>Backend: Decrypt document with DEK
    Backend-->>Frontend: Return decrypted document
    Frontend-->>Organization: Display document for review
    Organization->>Frontend: Verify or reject document
    Frontend->>Backend: POST /api/documents/:id/verify
    Backend->>Firebase: Update document status
    Backend->>Blockchain: Call verifyDocument() or rejectDocument()
    Blockchain-->>Backend: Return transaction hash
    Backend->>Firebase: Update with verification details
    Backend->>Firebase: Create notification for document owner
    Firebase-->>Organization: Confirmation of verification
```

7.3 Organization Verification Application

1. **Application Submission:** An organization registers with a wallet and completes a verification application form, which is submitted to the backend.
2. **Admin Review:** An Authentico administrator reviews the application via the admin dashboard, approving or rejecting it.
3. **Verification Status:** Upon approval, the organization's status is updated to "verified" in Firestore, enabling them to verify documents. The organization receives a notification of the decision.

```
sequenceDiagram
    participant Organization
    participant Frontend
    participant Backend
    participant Firebase
    participant Admin
```

```
    Organization->>Frontend: Complete application form
    Frontend->>Backend: POST /api/organizations/apply
    Backend->>Firebase: Store application
    Firebase-->>Backend: Confirm storage
    Backend-->>Frontend: Application submitted
    Frontend-->>Organization: Show confirmation
    Admin->>Frontend: Access admin dashboard
    Frontend->>Backend: GET /api/organizations/applications
    Backend->>Firebase: Query pending applications
    Firebase-->>Backend: Return applications
    Backend-->>Frontend: Display applications
    Admin->>Frontend: Review and decide
    Frontend->>Backend: PUT /api/organizations/applications/:id
    Backend->>Firebase: Update application status
    Backend->>Firebase: Update organization status
    Backend->>Firebase: Create notification
    Firebase-->>Organization: Notification of decision
```

7.4 Document Sharing

Users can securely share verified documents via:

1. **Sharing Methods:** Generating a unique verification link or a QR code embedding the link.
2. **Verification Page:** Recipients accessing the link are directed to a public verification page displaying document metadata, verification status, and blockchain transaction details. Crucially, the original document content is *not* displayed on this page, maintaining privacy.
3. **Security Measures:** The verification links only reveal metadata. The blockchain transaction can be independently verified on Etherscan, and the document hash can be checked for integrity, ensuring trust without exposing sensitive data.

```
sequenceDiagram
    participant Owner
    participant Recipient
    participant Frontend
    participant Backend
    participant Firebase
    participant Blockchain
```

```
Owner->>Frontend: Generate sharing link/QR
Frontend->>Owner: Display link/QR code
Owner->>Recipient: Share link/QR code
Recipient->>Frontend: Access verification page
Frontend->>Backend: GET /api/verify/:documentId
Backend->>Firebase: Get document metadata
Firebase-->>Backend: Return metadata
Backend->>Blockchain: Verify on-chain status
Blockchain-->>Backend: Return verification status
Backend-->>Frontend: Return verification details
Frontend-->>Recipient: Display verification page
```

8. Market Analysis and Target Customers

The market for secure document verification is rapidly expanding due to increasing digital transformation and the persistent threat of fraud.

8.1 Market Validation

The escalating rates of document fraud in Southern Africa, as evidenced by the Smile ID report, underscore the urgent need for a robust solution like Authentico. The report's findings of a significant jump in fraud rejection rates and substantial financial losses highlight a critical market pain point that Authentico directly addresses.

Key market trends driving demand for Authentico include:

- **Growing demand for efficient KYC (Know Your Customer) processes:** Businesses across sectors require faster, more reliable identity and document verification.
- **Increasing focus on data privacy and security:** Regulations and public awareness are driving demand for platforms that prioritize data protection.
- **Rising global adoption of blockchain technology:** The inherent security and transparency of blockchain are increasingly recognized for various applications, including document verification.

8.2 Competitive Advantage

Authentico distinguishes itself from competitors (e.g., Entrust, Sumsub, Persona) primarily through its foundational reliance on **blockchain technology**. While competitors often leverage AI/Biometrics for verification, Authentico's key

differentiators include:

- **Digital Wallet Integration:** Seamless Web3 wallet connectivity for user authentication and interaction.
- **Decentralized and Immutable Records:** The use of blockchain ensures that document verification records are unalterable and transparent, a core advantage over centralized systems.
- **Enhanced Trust:** By eliminating centralized authorities, Authentico fosters a trustless environment, reducing reliance on intermediaries.

8.3 Target Customers

Authentico is designed to serve a broad range of organizations and individuals who require secure and verifiable document management:

- **Financial Institutions:** Banks and fintech companies seeking to streamline KYC processes, reduce fraud, and enhance compliance.
- **Educational Organizations:** Universities, colleges, and e-learning platforms that need to verify academic credentials, transcripts, and certifications securely.
- **Human Resources Departments:** Businesses looking to improve employee onboarding, verify professional credentials, and manage internal documentation with greater security and efficiency.
- **Government Agencies:** For issuing and verifying official documents, licenses, and permits, reducing bureaucratic delays and combating fraud.
- **Healthcare Providers:** For secure management and verification of patient records and medical certifications.

8.4 Revenue Model

At its initial stage, Authentico prioritizes trust-building and platform adoption. Our revenue model is designed to ensure sustainability while encouraging widespread use:

- **Organization Verification Fee:** Organizations are required to pay a modest verification fee of BWP 1000.00 (approximately $75.00 USD) to activate their issuer status and begin verifying documents. This fee ensures commitment and helps sustain platform operations.

As the platform matures and expands, additional revenue streams could include premium features for organizations, API access for enterprise integrations, or transaction-based fees for advanced services.

9. Future Development and Roadmap

Authentico is committed to continuous improvement and expansion. Our future

roadmap includes:

9.1 High Priority Recommendations (from Security Analysis)

- **Implement Key Management Service (KMS):** Transition from environment variable storage of master keys and private keys to a dedicated KMS solution for enhanced security and key rotation.
- **Enhance Token Security:** Implement token rotation on sensitive operations and replace the global token blacklist with a scalable solution (e.g., Redis) for more robust token revocation.
- **Improve Session Security:** Implement CSRF protection for all state-changing operations and enhance session fingerprinting beyond IP address to include more robust client-side characteristics.
- **Fix Critical Test Failures:** Address and resolve all identified failing tests, particularly within the BlockchainService, and implement missing security tests to increase overall test coverage.

9.2 Medium Priority Recommendations

- **Enhance Role-Based Access Control:** Develop more granular permissions and implement resource-based access control to allow for finer-tuned authorization policies.
- **Improve Error Handling:** Standardize error handling across all components, implement sophisticated error recovery strategies, and optimize logging practices for production environments.
- **Strengthen Input Validation:** Ensure consistent and comprehensive input validation across both frontend and backend, including protection against NoSQL injection and implementation of a Content Security Policy (CSP).
- **Enhance Blockchain Security:** Implement transaction simulation before submission to prevent unexpected outcomes and integrate advanced gas price management.

9.3 Low Priority Recommendations

- **Improve Documentation:** Further enhance API documentation, provide detailed documentation of security features and assumptions, and develop a comprehensive security incident response plan.
- **Refactor Code for Consistency:** Standardize on TypeScript across all files, implement consistent logging practices, and reduce code duplication for improved maintainability.
- **Enhance Monitoring and Alerting:** Implement advanced security event monitoring, add automated alerts for suspicious activities, and create a

centralized dashboard for security metrics.
- **Improve Developer Experience:** Enhance error messages for developers, integrate security-focused linting rules, and create security checklists for code reviews.

10. Conclusion

Authentico represents a significant leap forward in digital document verification. By leveraging blockchain technology, decentralized storage, and robust encryption, we offer a solution that is not only secure and transparent but also highly efficient and user-centric.

While the platform demonstrates a solid foundation with strong security features, our commitment to continuous improvement, guided by the recommendations from our security analysis, will ensure Authentico evolves to meet enterprise-grade security standards.

We believe Authentico will play a pivotal role in reshaping the future of digital trust and document verification on a global scale, fostering a more secure and efficient digital world for individuals and organizations alike.