



# Decentralized Notary Service dApp

## Immutable Proof of Existence

Leveraging Web3 for Transparent Digital Document Notarization. This presentation outlines the architecture and next steps for our dApp.

### Project Goal

Build a functional Web3 application for immutable, transparent proof of existence and integrity for digital documents.

# The Shift from Centralized to Decentralized Notarization

## The Problem: Centralized Bottlenecks

### Slow & Costly

Traditional notarization is often a bureaucratic process that is time-consuming and expensive.

### Lack of Integrity

Guaranteeing the permanent integrity of digital documents against tampering is challenging in centralized systems.

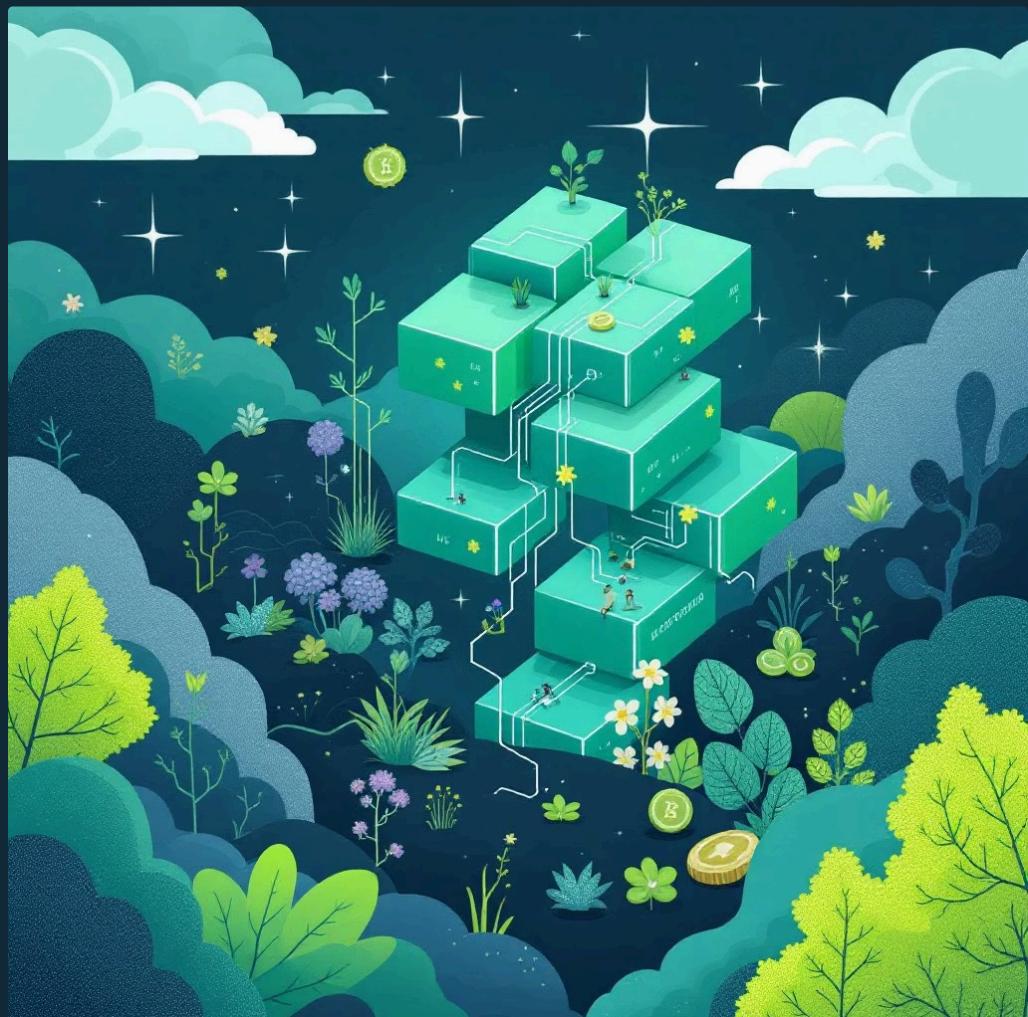
## The Solution: Immutable Blockchain Records

### Decentralized Notary dApp

Uses a smart contract to create an unchangeable record of a document's cryptographic hash (fingerprint) on a public blockchain.

### Core Objective

Develop a functional Web3 application for **immutable, transparent** document notarization and verification.



# Technical Foundation: Contract and Frontend Architecture

Our initial deployment establishes a robust foundation for secure, transparent document proof-of-existence.



## Deployment Network

Deployed on the **Sepolia Test Network**.

Contract Address:

Oxd184b5eA12FD46e847CdDa76bbAeF1  
299977386D



## Core Functionality

Stores a document's **SHA-256 hash** and the notarizer's address using a public mapping. Events are logged via DocumentNotarized.



## Frontend Stack

UI/UX developed with responsive **HTML / Tailwind CSS**. Web3 interaction handled by **Ethers.js v6** for reliable connectivity.



Current features include explicit wallet connection and display of complete historical notarization logs for the connected user.

# Roadmap: State Transition to Enterprise Readiness

We have identified critical upgrades required to move the dApp from a functional prototype to a robust, integrated service.

User Control/UX	Explicit "Connect Wallet" button functionality.	Implement a <b>Wallet Connect</b> button for <b>easy switching/disconnection</b> from within the dApp.
Smart Contract Logic	Single-tier access; <b>any wallet</b> can notarize, which is insecure.	Implement <b>Multi-Client Role-Based Access Control (RBAC)</b> : Only whitelisted company addresses can notarize; public users retain <b>verification</b> access.
Application Integration	Standalone web application.	Convert dApp to a <b>Silent API Service</b> (via postMessage protocol) for seamless integration with a backend application ("AI Interviewer").
Deployment	Local hosting via http-server.	Prepare for <b>public deployment</b> via <b>GitHub Pages</b> .



# Next Steps: Phase 2 for Secure, Scalable Adoption



## Security: Implement RBAC

Apply strict **Role-Based Access Control (RBAC)** within the Smart Contract to secure notarization rights.

## UX: Wallet Management

Enhance **Wallet Management** features, including connection switching and improved user experience flows.

## Integration: Silent API

Develop the **Silent API** integration layer to enable seamless use by enterprise backend systems without requiring a full dApp UI.

## Go-Live: Public Deployment

Finalize the **GitHub Pages** deployment process for public access and testing.

- ❑ The decentralized notary dApp is technically complete and is now ready for the crucial **security and integration phase** to enable secure enterprise and public use.