The “Smart Will” – Blockchain for Afterlife Asset Transfer

The **Smart Will** is a decentralized inheritance management solution using **blockchain**. It ensures secure, transparent, and automated execution of wills without intermediaries. The system comprises two main contracts:

1. DigitalWillFactory Contract (Factory for Deploying Wills)

Purpose:

The **DigitalWillFactory** contract is responsible for deploying and managing multiple instances of the **DigitalWill** contract. Instead of creating a separate DigitalWill contract manually, users deploy them through this factory contract, ensuring scalability and easy tracking.

Key Functionalities and Flow:

1. **Deploying a Will Contract:**
    - A user calls deployDigitalWill(), which creates a new DigitalWill instance.
    - The address of the new DigitalWill contract is stored in deployedDigitalWills.
    - An event DigitalWillDeployed is emitted with the deployer’s address.

2. **Creating a Will on an Existing DigitalWill Contract:**
    - Users call createWillOnDigitalWill() by specifying:
        - **Beneficiaries** (who will inherit assets).
        - **Shares** (percentage of inheritance).
        - **Assets** (physical/digital assets tied to the will).
        - **Oracles** (trusted verifiers who confirm the owner’s death).
        - **Aadhaar numbers** (for identity verification).
    - The function forwards the request to the createWill() function inside the DigitalWill contract.

3. **Verifying Death via Factory Contract:**
    - When the owner dies, oracles call verifyDeathViaFactory() to confirm the owner’s passing.
    - This forwards the request to oracleVerifyDeath() in DigitalWill.
    - An event DeathVerifiedViaFactory is emitted.

4. **Revoking a Will:**
    - The owner can revoke their will by calling revokeWillViaFactory(), which delegates the call to revokeWill() in DigitalWill.
    - An event WillRevokedViaFactory is emitted.

5. **Fetching Will Details for Users & Frontend Integration:**
    - Functions getWillDetailsFromDigitalWill(), getOracleCountFromDigitalWill() and getOwnerWillIds() allow retrieval of will data for UI display.

6. **Admin Management:**
    - The admin can update the contract owner using updateAdminOnDigitalWill().

2. DigitalWill Contract (Core Inheritance Contract)

Purpose:

The **DigitalWill** contract manages individual wills, ensuring proper beneficiary allocation, death verification, and asset distribution.

Key Functionalities and Flow:

1. **Creating a Will (createWill)**
    - The user registers a will by specifying:
        - **Beneficiaries and their shares** (who gets what percentage).
        - **Assets** (details of the assets included in the will).
        - **Oracles** (trusted verifiers responsible for confirming the owner's death).
    - A unique **will ID** is generated and stored on-chain.

2. **Modifying a Will (modifyWill)**
    - The owner can update their will by modifying beneficiaries, assets, or assigned oracles.

3. **Revoking a Will (revokeWill)**
    - The owner can cancel their will at any time before their death.

4. **Death Verification (oracleVerifyDeath)**
    - Oracles assigned to the will verify the owner's death.
    - Once a predefined number of oracles confirm, the will is **activated** for execution.

5. **Asset Distribution (executeWill)**
    - Once death is confirmed, the contract releases the assets to the beneficiaries **automatically** based on the predefined shares.

6. **Retrieving Will Data:**
    - Functions like getWillDetails(), getWillOracles(), and ownerWills() allow users to retrieve complete will details, including assets and verification status.

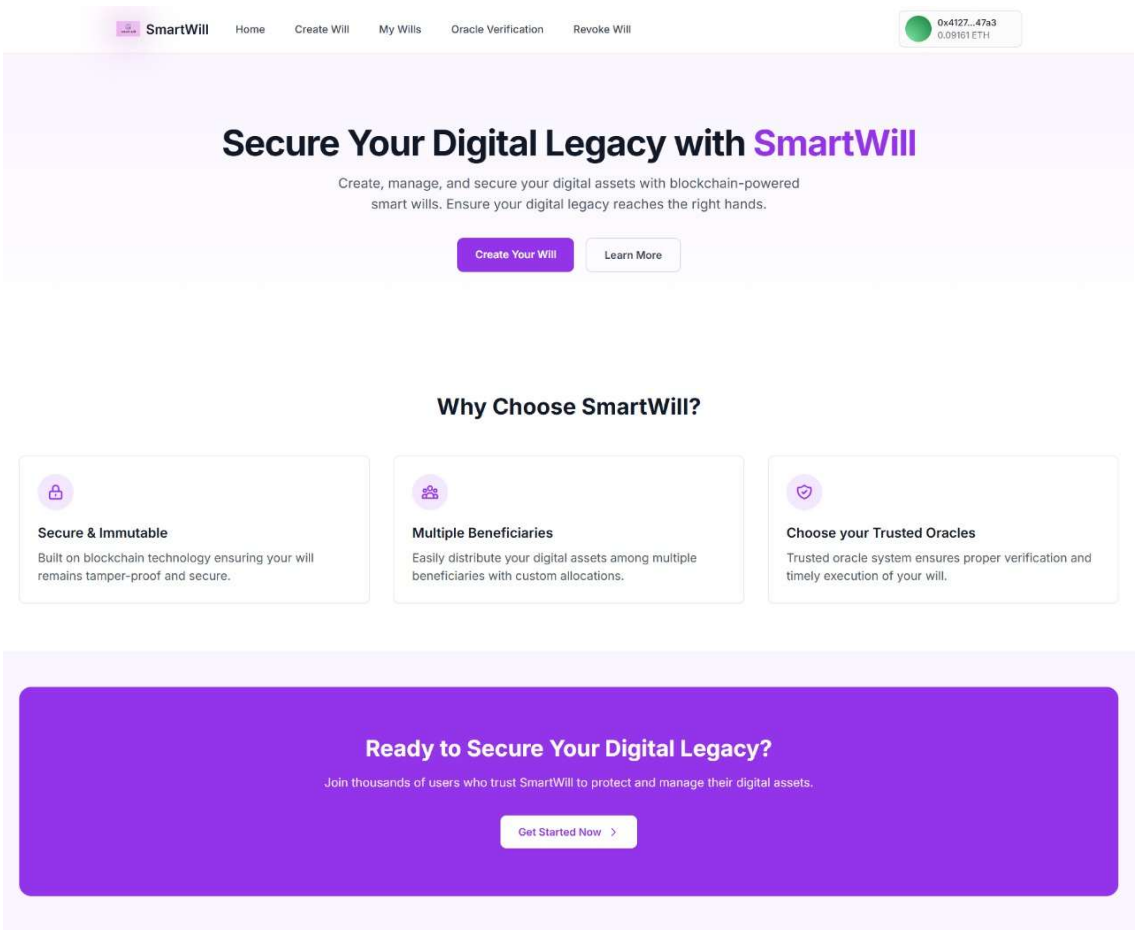
Project Execution Flow

1. **Deploy DigitalWillFactory Contract** (initial setup by the platform).
2. **User Deploys a DigitalWill Contract** via the factory.
3. **User Creates a Will** on their DigitalWill contract (assigns beneficiaries, assets, oracles).
4. **Oracles Verify the Death** (trusted individuals confirm the owner's passing).
5. **Assets are Automatically Transferred** to the Beneficiaries.

User Interface

1. User Registration & Authentication

- New users create an account by providing meta mask login.
- The system links the user to a **smart contract** that will store their digital will.

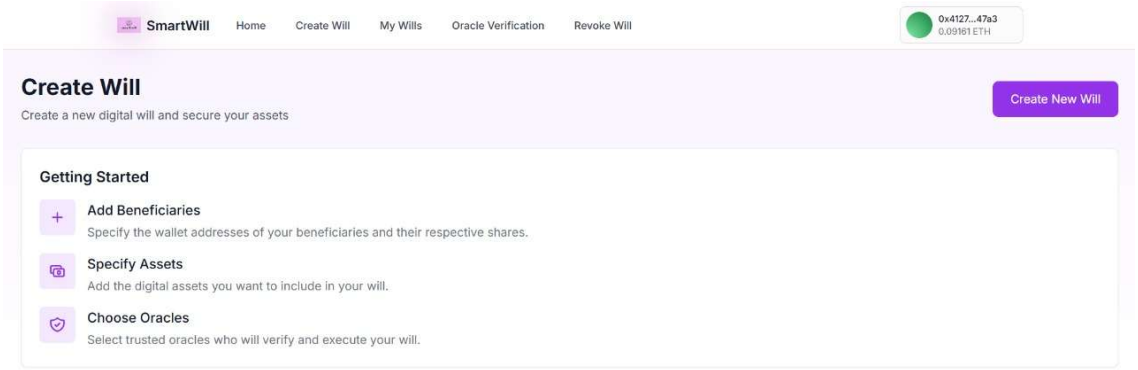


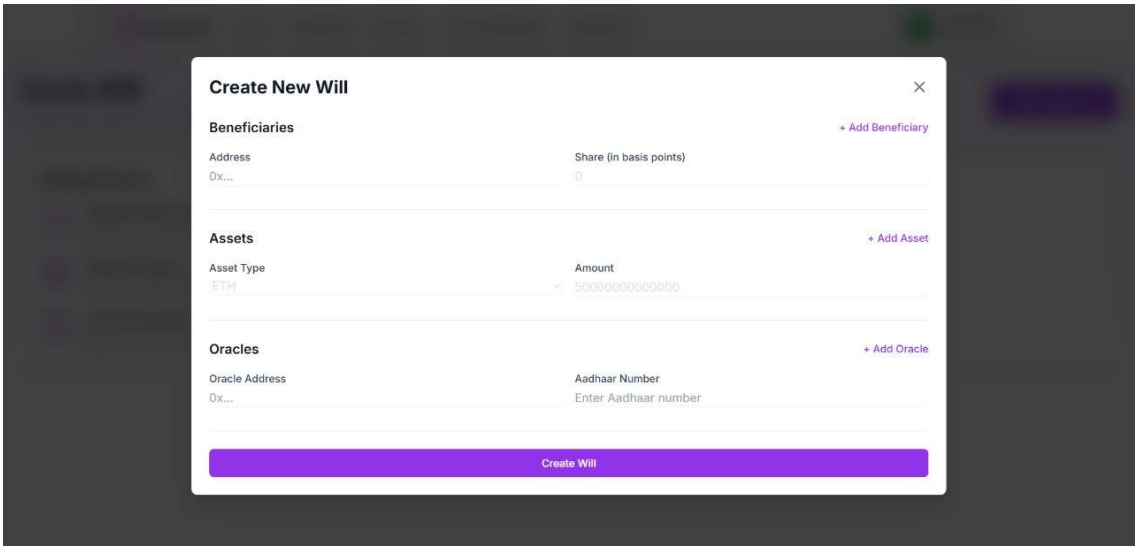
Tech Involved:

- **Blockchain Wallet Integration:** Users must connect their crypto wallet (e.g., MetaMask).

2. Creating a Digital Will

- The user inputs information about their assets (crypto, digital documents, etc.).
- They specify beneficiaries and the percentage of assets allocated to each.
- A **smart contract** is automatically generated with these details.



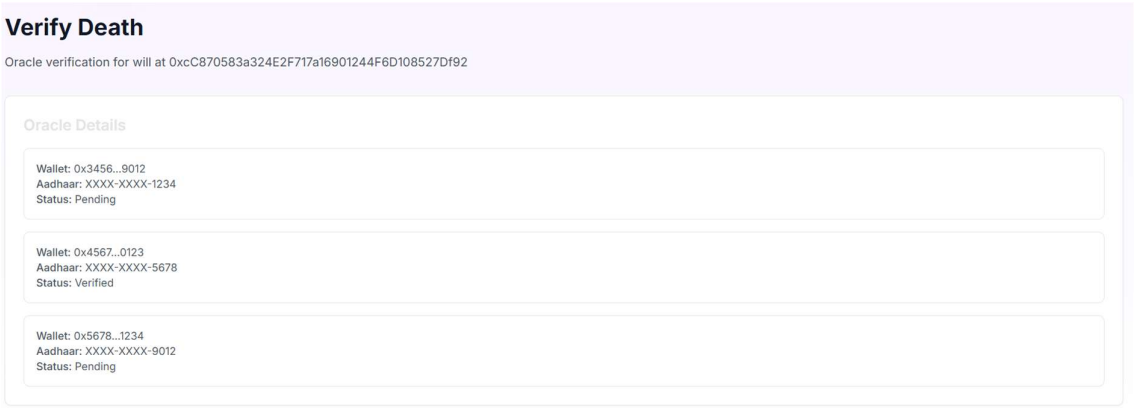


Tech Involved:

- **Smart Contract Deployment:** A self-executing contract is written to the blockchain.
- **Decentralized Storage:** IPFS or a similar system is used to store will documents.

3. Verification Mechanisms

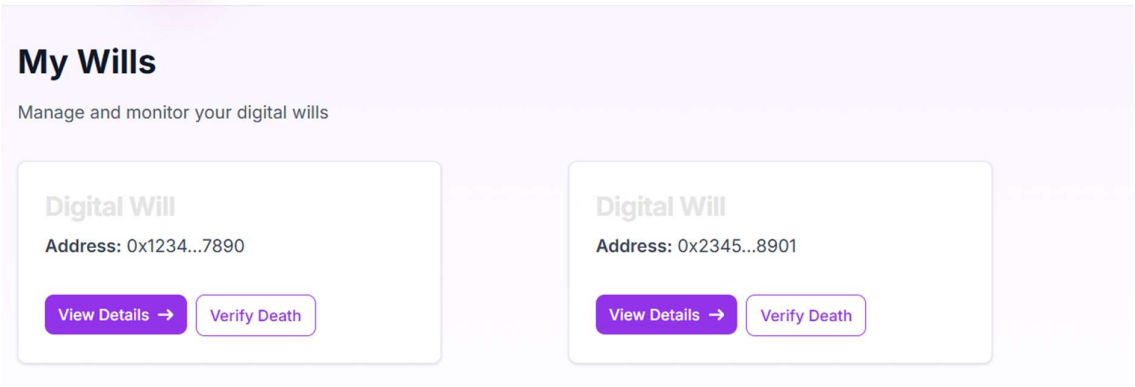
- Users must designate an **Oracle Service** that confirms their death.
- Trusted individuals confirm the owner's passing



4. Will Storage & Security

Process:

- The will is encrypted and stored securely.
- Only the **executor** (smart contract) can trigger its release upon verification.
- The user can modify or update the will if needed



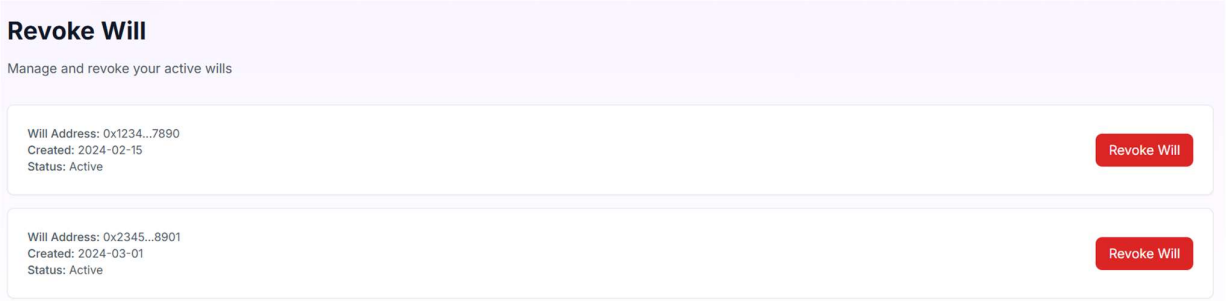
Tech Involved:

- **Encryption (AES, RSA, etc.)** ensures the document remains private.
- **Decentralized File Storage** prevents a single point of failure.

5. Execution of the Will (Upon Death Confirmation)

Process:

- The Oracle confirms the user's death.
- The smart contract **automatically distributes** assets to beneficiaries.



Tech Involved:

- **Automated Asset Transfer:** Smart contracts execute predefined transactions.
- **Decentralized Decision Making:** DAO or multi-signature verification is used.

6. Beneficiary Access & Claiming Assets

Process:

- Beneficiaries must authenticate and provide necessary details to claim assets.
- Once verified, funds are directly transferred to their wallets.

Tech Involved:

- **Multi-Signature Transactions:** Ensures security in asset transfers.