Multi-Signature Wallet using Blockchain

Secure Decentralized Fund Management with Ethereum, Hardhat & React

Student: Sushant Tulasi

Institution: VESIT, 4th Year Engineering

Tools: Solidity, Hardhat, React.js, AWS

Network: Sepolia Testnet

Abstract

This project implements a **Multi-Signature Wallet (MultiSig Wallet)** on the Ethereum blockchain to enhance the security of digital fund management.

It ensures that no single owner can execute a transaction alone — instead, a minimum number of approvals are required.

Core Technology Stack

Built using **Solidity** for smart contracts.

Hardhat for deployment.

React.js for frontend interaction.

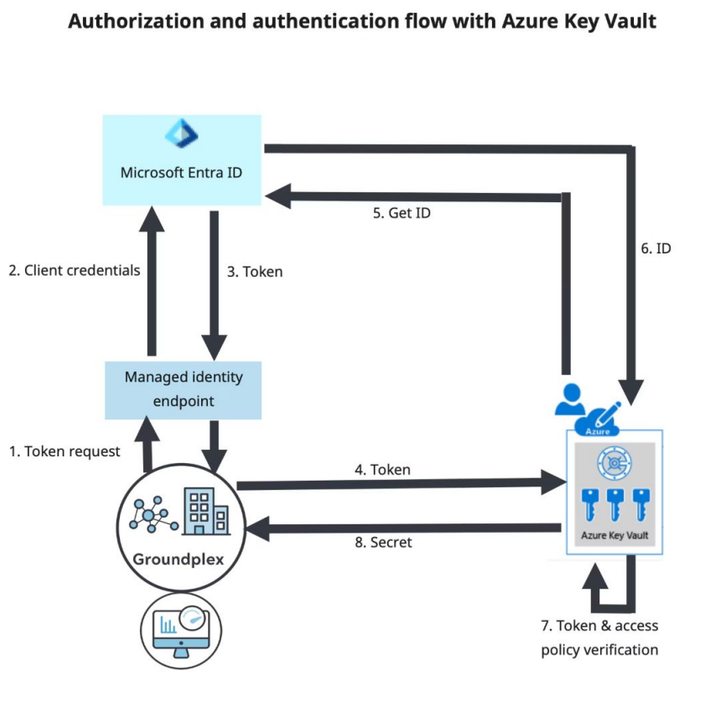
AWS for cloud hosting.

Introduction & Motivation

Traditional wallets allow a single private key to control funds — a single point of failure.

The Problem

Organizations or shared groups need a **trustless mechanism** where multiple users approve before funds move.



The Solution

MultiSig Wallets solve this by distributing control among multiple owners.

Motivation: Build a decentralized, transparent, and tamper-proof wallet to demonstrate real-world blockchain use.

Objectives

The project focused on achieving five key goals to deliver a functional and secure decentralized application.

1 Secure Smart Contract Design

Design a secure Ethereum smart contract supporting multiple owners.

2 Enforce Approval Threshold

Enforce a minimum number of approvals before fund release.

3 User-Friendly Frontend

Develop a clean frontend using React and MetaMask.

4 Public Accessibility

Deploy the system on AWS for public accessibility.

5 Decentralized Fund Management

Demonstrate decentralized fund management for group ownership.

System Architecture

The system is structured in distinct layers, ensuring separation of concerns and robust interaction between the user interface and the blockchain.

Components:

Frontend

(React + Ethers.js): User interaction and MetaMask integration.

Smart Contract

(Solidity): Logic for ownership, approval, and transaction control.

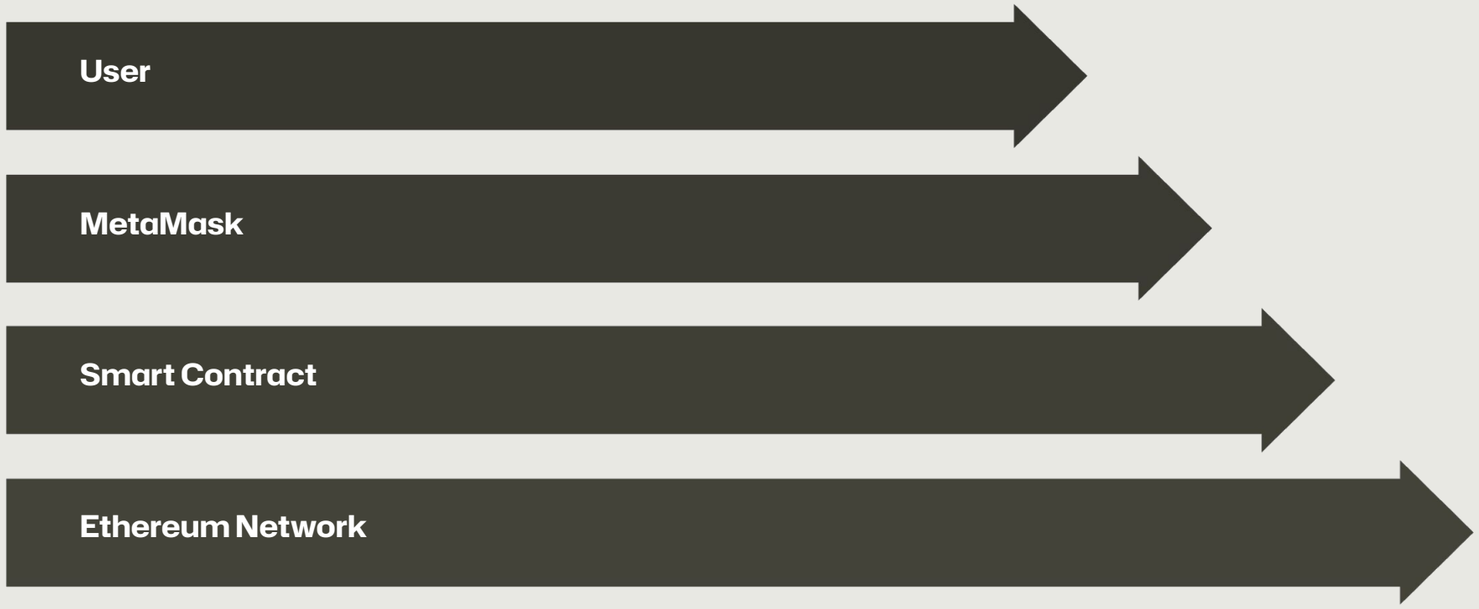
Blockchain

(Ethereum / Sepolia): Decentralized ledger for data and fund storage.

Deployment

(AWS): Hosting the user interface.

Flow:



Smart Contract Design

The core logic is implemented in **Solidity (v0.8.x)**, focusing on secure, multi-party transaction management.

Main Features:

- Multiple owners assigned at deployment.
- Threshold approval logic (M-of-N).
- Events for Submit, Approve, Revoke, and Execute.
- Each transaction stored on-chain.

Key Functions:

submit

```
submit(address to, uint value, bytes data)
```

approve

```
approve(uint txId)
```

revoke

```
revoke(uint txId)
```

execute

execute(uint txId)

Frontend & MetaMask Integration

The user interface, built with **React.js + Ethers.js**, provides a seamless experience for interacting with the MultiSig contract.

Features:

- Connect wallet button using MetaMask
- Display connected account and ETH balance
- Show list of pending transactions
- Approve or execute transactions directly from UI

Interaction Flow:

1

Connect

User connects MetaMask

2

Fetch State

React fetches balance & contract state

3

Action

User submits or approves a transaction

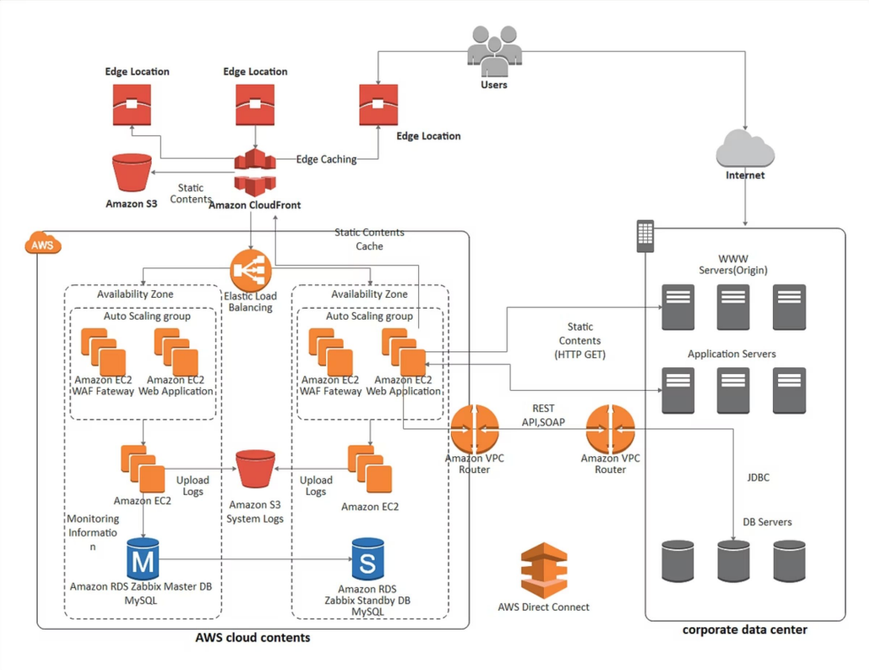
4

Update

Contract updates on blockchain

AWS Deployment

Hosting Setup:



Build the React app using `npm run build`.

Deploy the build folder to **AWS S3 (Static Website Hosting)** or **EC2 instance**.

Use **AWS CloudFront** for HTTPS and caching.

4. Optional: Add a custom domain name.

Benefits:

→ High availability

→ Global reach via CloudFront

→ Easy scalability and updates

Results & Future Enhancements

Results:

- Successfully deployed MultiSig Wallet on Sepolia Testnet.
- Multiple users could submit, approve, revoke, and execute transactions securely.
- Fully decentralized — no central authority.
- Integrated React UI + MetaMask successfully.

Future Enhancements:

- Add mobile support via WalletConnect.
- Integrate notifications for approvals.
- Use Polygon or Optimism for lower gas fees.
- Add analytics dashboard for transaction history.

Conclusion

The **Multi-Signature Wallet** project demonstrates the power of blockchain in creating secure, collaborative financial systems. It ensures transparency, eliminates single-user control, and provides a real-world decentralized application model.

Through Solidity, Hardhat, React, and AWS, this project bridges blockchain logic with cloud-hosted usability.

> # Key Takeaway:
>
> ## Decentralization + Multi-Approval = Trustworthy Financial Security