



# Reach17 DAO

Sistema di governance DAO  
basato sugli obiettivi dell'Agenda  
2030 delle Nazioni Unite

# Obiettivo del Progetto:

- Realizzare una DAO basata sugli obiettivi dell'Agenda ONU 2030 per gestire le attività della società Reach17.
- Il modello di governance adottato combina democrazia diretta e rappresentativa, includendo meccanismi di voto ponderato e accesso alla governance tramite acquisto di token ERC-20

# Libreria Viem e Scelte Tecniche

- **Viem**: Una libreria moderna in TypeScript che facilita l'interazione con la blockchain in maniera più modulare e orientata agli strumenti di sviluppo odierni. È una scelta strategica per evitare librerie più datate come Ethers, offrendo un'esperienza di sviluppo coerente con TypeScript e maggiore sicurezza nel typing.
- Favorisce un controllo più preciso su chiamate e transazioni, semplifica il test delle interazioni con gli smart contract e assicura una gestione accurata dei tipi per chiamate R/W negli smart contract

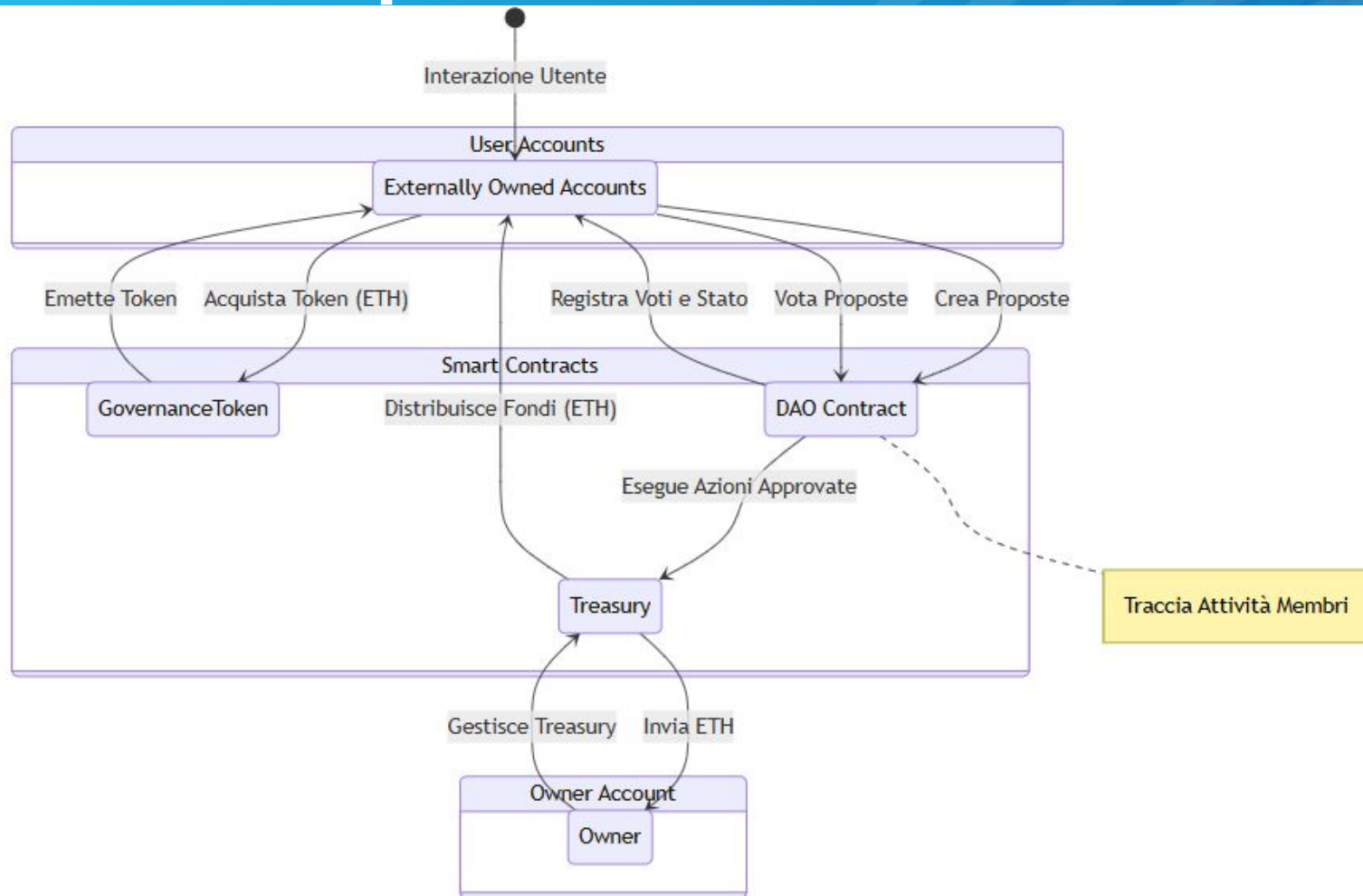
# Hardhat e Logica del Framework

- **Hardhat è la scelta ideale per costruire e testare l'intero ciclo di sviluppo del progetto, dall'implementazione allo scripting fino ai test automatici.**
- **Supportando TypeScript, rende semplice l'integrazione con Viem e mantiene il progetto fortemente tipizzato**
- **Sfruttando i moduli di testing di Hardhat, ogni funzione della DAO (dall'acquisto di token al sistema di proposta e voto) è stata rigorosamente validata tramite test unitari e di integrazione.**

# Sviluppo di Contratti e Struttura Logica

- **Ogni smart contract è stato strutturato per facilitare le operazioni principali della DAO, con una logica chiara per la gestione dei token, la creazione delle proposte e il voto.**
- **Il codice segue uno stile modulare e leggibile, basato su separazione delle responsabilità: un contratto per la gestione dei token, uno per la DAO e una logica di voto accuratamente modellata.**

# Sinottico implementazione Dao





# Funzionalità della DAO 1/2

## Acquisto Token e Accesso alla Governance

- *I membri acquistano token ERC-20 come strumento di partecipazione alla governance*
- *Voto ponderato in base alla quantità di token posseduti, offre maggiore influenza a chi ha un maggiore interesse economico nella DAO*

## Sistema Proposte/Votazioni

- *I membri possono proporre nuove idee, finanziamenti o cambiamenti aziendali. Ogni proposta contiene titolo, descrizione, destinatario e importo.*
- *Modalità di Voto: In favore, contro e astensione, per esprimere in modo sfaccettato il parere dei membri.*

# Funzionalità della DAO 2/2

- **Esecuzione delle Proposte Approvate:** *gestita direttamente dall'owner, include l'invio di fondi o l'aggiornamento di stato delle proposte.*
- **Protezione da attacchi:** *previsto un controllo anti-rientro per evitare exploit, specialmente in azioni critiche come l'esecuzione di fondi.*
- **Chiusura delle proposte**



# Testing con Hardhat 1/2

## GovernanceToken

- *Test per verificare acquisto token, aggiornamento supply, ritiro ETH e accesso esclusivo del proprietario.*

## DAO e Proposte

- *Test per assicurare la creazione corretta delle proposte, il sistema di votazione e l'esecuzione protetta delle stesse*

## Scenari di voto e chiusura

- *Validati per garantire che solo i membri possano partecipare e che l'esecuzione e la chiusura delle proposte seguano le logiche prestabilite*

# Testing con Hardhat 2/2

## Execute Test: *"npx hardhat test"*

### DAO Governance Token Tests

#### Proposal Voting and Execution with Token Transfers

- ✓ Should allow a token-holding account to vote on a proposal (119
- ✓ Should execute proposal and transfer tokens when majority is re
- ✓ Should not execute proposal when majority is not reached

#### Proposal Expiration and Voting Period

- ✓ Should handle proposal expiration correctly
- ✓ Should allow multiple proposal recovery at once
- ✓ Should track proposal timing correctly

### GovernanceToken

#### Token Purchase

- ✓ Should allow users to buy tokens with correct ETH amount
- ✓ Should update total supply after token purchase
- ✓ Should not allow purchase with 0 ETH

#### Withdrawal

- ✓ Should allow owner to withdraw accumulated ETH
- ✓ Should not allow non-owner to withdraw

#### Token Balance Management

- ✓ Should accurately track multiple purchases from same address
- ✓ Should protect against reentrancy attacks

### DAO Contract Tests

#### Proposal Creation

✓ GovernanceToken Deployment completed successfully

25 passing (2s)

# Deploy SmartContract con Hardhat

## Deploy su rete locale:

*“npx hardhat run scripts/deployContracts.ts”*

## Deploy su rete pubblica di Test:

*“npx hardhat run scripts/deployContracts.ts  
--network arbitrumSepolia”*



DAO Address:

[0xc2106bf7edf220b33e775242e8d786db3ca7d9cf](#)



GovernanceToken Address

[0xe49288d0dc39921099a60d4de445490b08c2272b](#)



Owner Address:

[0x6Db728e8A98b89C421CBb0cF23A3e9975a943B27](#)

# Conclusioni

- **La DAO consente una governance decentralizzata, conforme alle esigenze dell'azienda e agli obiettivi ONU 2030.**
- **Strumento solido per la governance partecipativa, che può evolversi nel tempo grazie all'architettura modulare**
- **Testing rigoroso e protezioni contro attacchi garantiscono affidabilità, contribuendo a una gestione finanziaria sicura e trasparente.**



# Deploy e versioning GitHub del codice

- La fase di sviluppo del progetto è stata facilitata dall'uso del IDE Visual Studio Code pushando le modifiche sul repository GITHUB prima sul branch “development” per poi eseguire il merge sul branch principale “main”, il codice è disponibile sulla repository:

<https://github.com/antopat1/ProgettoSmartContractonSolidityAdvancedDIAntoninoPaterno>