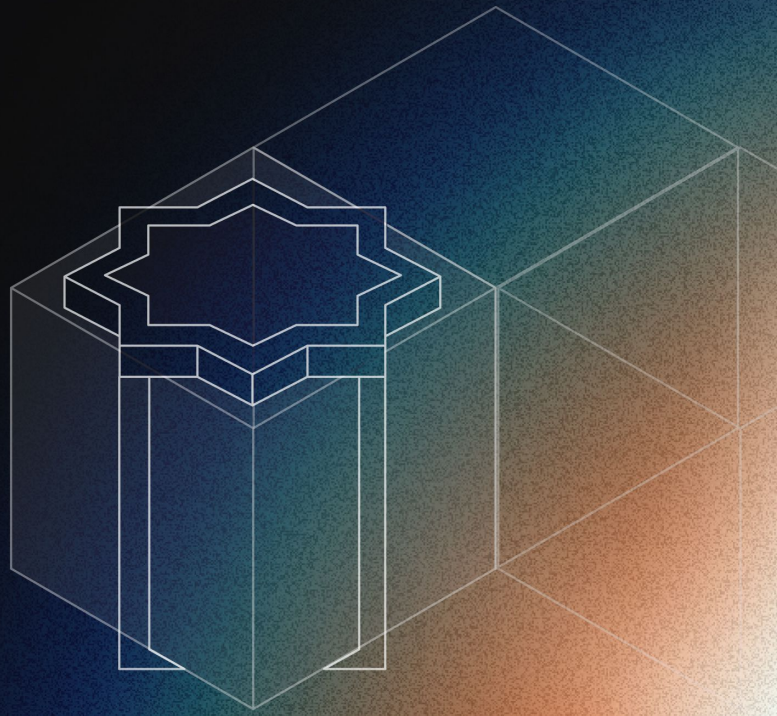
WORKSHOP: BUILDING DECENTRALIZED APPLICATIONS ON HAQQ NETWORK

Introduction

- Welcome to the Workshop
- Importance of Decentralized Technology
- Focus on Ethical Finance



Why DeFi For Social Impact?

- Transforming Traditional Finance Systems
- Creating a Fairer and Inclusive Financial World
- Role of DeFi in Social Impact

Key Challenge Objectives & Themes

- Ethical Finance
- ESG-Powered Solutions
- Interoperability
- Inclusivity & Accessibility
- Adherence to Shariah-Compliant Principles

HAQQ Network

HAQQ is built using Evmos which in turn is based on Cosmos SDK. High throughput and instant finality are inherited from the Tendermint Core. IBC – from Cosmos SDK and Ethereum Compatibility – from Evmos. Two important Tendermint consensus engine characteristics are:

- Security – network works even if up to ⅓ of nodes fail or act maliciously.
- Consistency – every non-faulty node sees the same transaction log and computes the same state.

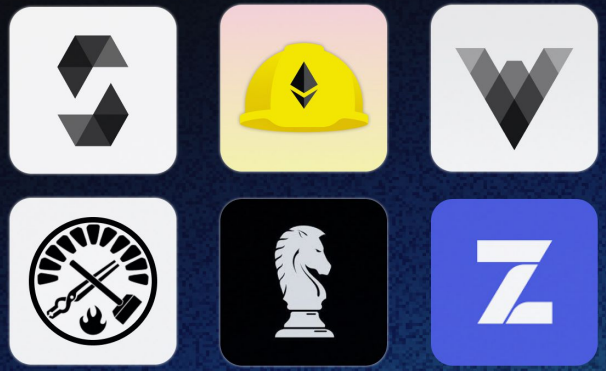
Evmos

COSMOS

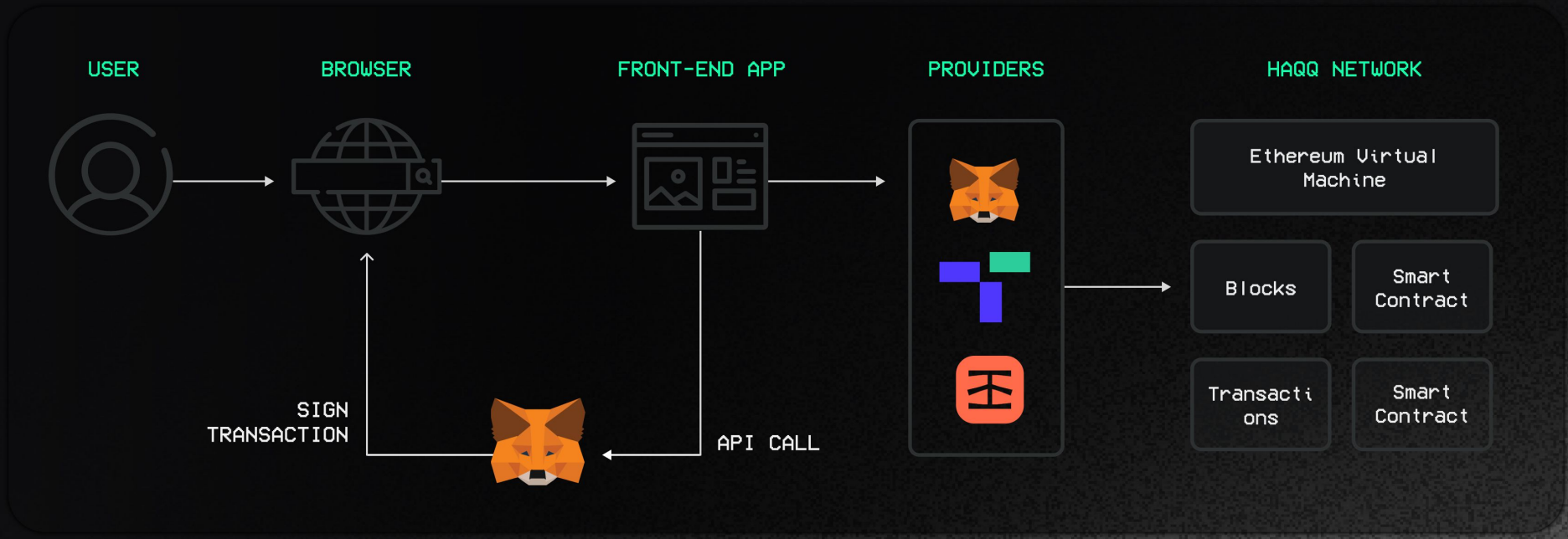
As result scalability and rich development experience is enabled from the beginning.

Technical Considerations

- Solidity Supported version up to v0.8.19
- Hardhat, Foundry, Vyper, Huff, etc
- Block gas limit: 40,000,000 in comparing with Ethereum mainnet 30m



Web3 DApp Scaffold



Demo - Building A Simple DApp

- Install wallet: Metamask, Rabby, Coinbase
- Add Haqq mainnet and testnet to the wallet
- Request tokens from faucet

```
!faucet test 0x7E2266F3aB3Eedfde02CCeE4b41e5CAEA185509a
```

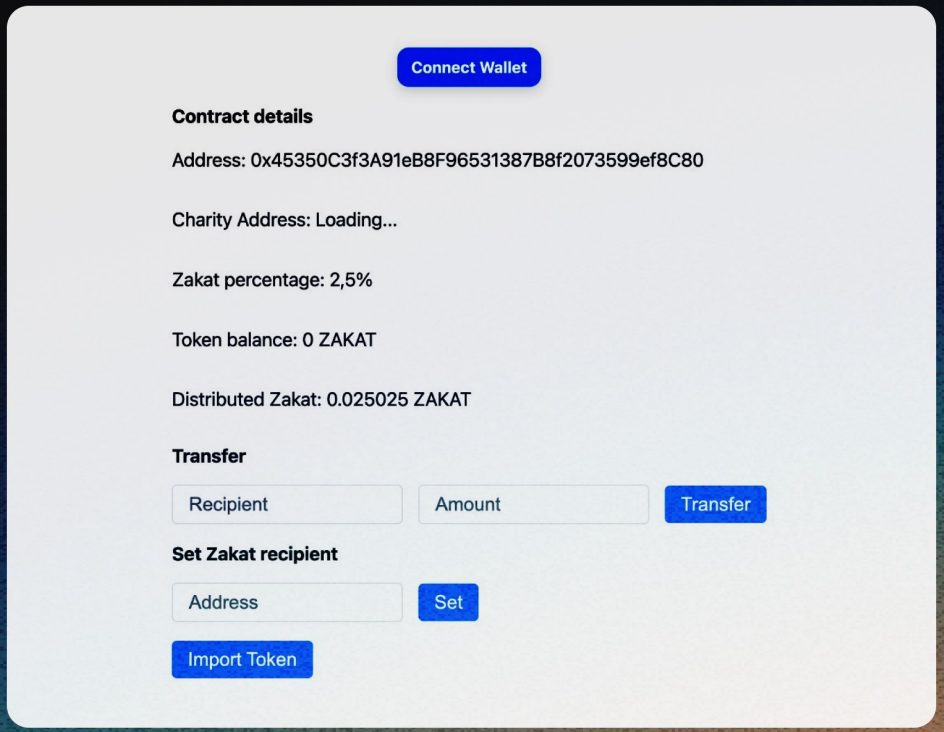
Demo - Solidity Contract

- Solidity Contract
- Deployment with Hardhat

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.19;

import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Permit.sol";

/// @title ZakatToken - A token contract that supports zakat (charity) distribution
/// @notice This contract implements a token with an optional zakat (charity) distribution feature.
/// Users can send transactions with a portion of the amount being automatically sent to a zakat recipient.
contract ZakatToken is ERC20, ERC20Permit, Ownable {
    // Constants
    uint256 private constant PERCENTAGE_DENOMINATOR = 1000; // Denominator for percentage calculation

    // State variables
    uint256 public zakatPercentage = 25; // Represents 2.5% (as before, divided by 1000 later)
    uint256 public distributedZakat; // Total tokens distributed as zakat

    // Default zakat address for users who haven't set their own
    address public defaultZakatAddress;
    mapping(address => address) private userZakatRecipients;

    /// @dev Initializes the contract with the provided token name, symbol, and default zakat address.
    /// @param name The name of the token.
    /// @param symbol The symbol of the token.
    /// @param _defaultZakatAddress The default zakat recipient address.
    constructor(string memory name, string memory symbol, address _defaultZakatAddress)
```

Demo - Web Integration

- Add network to wallet
- Sign In button
- Interaction with Smart Contracts read/write
- Deploying Web Application on Vercel



Tatum

- Explore HAQQ RPC Documentation
- Visit: https://docs.tatum.io/
- Scan QR Code and get up to 6 months of free use!

Contact Information

Shariah Oracle

HAQQ

Islamic Coin

Grants

HAQQ x IslamicCoin

Thank You

HAQQ x IslamicCoin