

Developer Documentation: Blockchain-Based Energy Trading in AI-Driven Smart Appliance Control System

Project Title:

Blockchain-Based Energy Trading in AI-Driven Smart Appliance Control System

Objective:

To develop a secure, decentralized system that integrates blockchain technology for P2P energy trading with an AI engine that optimizes appliance control based on energy availability, cost, and user behavior.

1. System Architecture Overview

- Frontend (Web/Mobile Dashboard): User interaction, energy monitoring, trading interface.
- Backend (Server/API): AI decision engine, blockchain integration, database.
- Smart Appliances / IoT Devices: Controlled based on AI logic.
- Blockchain Network (e.g., Ethereum, Hyperledger): For energy transactions and smart contracts.

2. Modules & Development Tasks

A. AI Engine Development

- Language: Python
- Develop models to predict appliance usage patterns.
- Integrate sensor data (e.g., solar output, grid availability).
- Train ML models to suggest when to use/store/sell energy.
- Use libraries: TensorFlow, scikit-learn, pandas.

B. Smart Appliance Control

- Language: Python, C++, or embedded C (depending on device)
- Create APIs for device communication (MQTT/HTTP).
- Build firmware to receive control commands from the server.
- Implement fail-safe modes and manual override.

C. Energy Monitoring System

- Language: Python, Node.js
- Collect data from smart meters (voltage, current, kWh).
- Store data in real-time DB (e.g., InfluxDB, Firebase).
- Push data to AI engine and frontend.

D. Blockchain Integration

- Blockchain: Ethereum (using Web3.js or Web3.py)
- Write Smart Contracts for energy trading logic.
- Connect blockchain wallet (e.g., MetaMask, TrustWallet).
- Develop transaction and trade execution logic.
- Use libraries: Web3.js, Solidity, Truffle, Ganache for local testing.

3. Backend Development

- Language: Node.js / Python (Flask, Django)
- Manage user profiles, preferences, wallet keys.
- Interface with AI, blockchain, and IoT modules.
- Expose REST APIs for frontend and devices.

4. Frontend Development

- Tech Stack: React.js / Flutter (for mobile)
- Display energy usage and AI suggestions.
- Show trading options and transaction history.
- Enable wallet management and contract execution.