Project Design Phase-I

Solution Architecture

Date	16 October 2023
Team ID	NM2023TMID09517
Project Name	ClimateTrackSmart using block chain
Maximum Marks	4 Marks

INTRODUCTION

Solution architecture is a multifaceted process that serves as the crucial link between business challenges and technological solutions. Its primary objectives encompass finding the optimal technology solutions to address existing business problems, describing the software's structure and characteristics, defining features and development phases, and providing the necessary specifications for solution management and delivery. In the context of this document, we will outline the solution architecture for a project that aims to create and deploy a smart contract on the Ethereal block chain. The project's goal is to leverage block chain technology for a specific application. Here, we will provide a step-by-step guide to ensure successful project completion.

PREREQUISITES

Before diving into the solution architecture, ensure you have the following prerequisites in place:

Node.js:

Node.js is a JavaScript runtime that will be used for various tasks during the project.

Visual Studio Code (VS Code):

VS Code is a popular code editor that will facilitate code development and management.

MetaMask:

MetaMask is an Ethereal wallet and gateway to the Ethereal block chain. It will be used for smart contract deployment.

SOLUTION ARCHITECTURE STEPS

Step 1: Project Setup

1. **Obtain the Project Files**: Download the project files in a compressed ZIP format. Extract all files from the ZIP archive.

2. **Open Visual Studio Code (VS Code):** Launch VS Code. In the top left corner, select "Open Folder. “Choose the folder where you extracted the project files and open it.

3. **Copy Smart Contract Code:** Locate and open the project name. Sol file within your project folder. Copy the smart contract code from this file.

4. **Remix IDE**: Open the Remix IDE platform (an online Solidity development environment). Create a new file in Remix and name it projectname.sol. Paste the smart contract code copied from VS Code into this new file.

5. **Compile the Smart Contract:** In Remix, navigate to the Solidity compiler section. Select projectname.sol and click the "Compile" button to compile the smart contract.

6. **Deploy the Smart Contract:** In Remix, choose the "Deploy & Run Transactions" section. Select the injected provider -MetaMask - as the deployment environment. Click "Deploy." MetaMask will open, and you should confirm the deployment by clicking "OK."Once deployed, copy the address of the deployed contract.

7. **Update Connector.js**: In your project directory, search for the connector.js file. In the connector.js file, paste the address of the deployed smart contract at the bottom of the code within the export cost address variable. Save the code.

STEP 2: FRONTEND SETUP

1. **Open File Explorer:** Access your file explorer.

2. **Navigate to the Project Folder**: Open the folder where you extracted the project files.

3. **Access Frontend Files**: In the project folder, locate the "src" directory.

4. **Open the Command Prompt (Cmd):** Press "Alt + A" to select all the files and directories within the "src" folder. In the search bar, type "cmd" and press Enter.

5. **Install Dependencies**: In the Command Prompt, enter the following commands:
   - **npm install**
   - **npm bootstrap**
   - **npm start**

6. **Access the Frontend:** Once the installation is complete, a local development server will be started.
7. Copy the {LOCALHOST IP ADDRESS} provided, open it in Google Chrome (or your preferred browser), and explore the frontend of your project.

CONCLUSION

This document provides a clear and systematic approach to setting up, developing, and deploying a smart contract on the Ethereal block chain. By following these steps, you will be well-equipped to create and manage your block chain-based project effectively. Make sure to adapt the instructions to your specific project requirements and stay organized throughout the process for a successful outcome.