Mobile Theft Detection

Arkadip Biswas
Dept. Of Computer Science & Engg.
23CSM1R26
ab23csm1r26@student.nitw.ac.in

Anik Das
Dept. Of Computer Science & Engg.
23CSM1S01
ad23csm1s01@student.nitw.ac.in

I. Introduction

Mobile theft is a pervasive issue in today's digital age, providing substantial challenges for individuals, corporations, and law enforcement agencies around the world. As smartphones and other mobile devices have become a vital part of daily life, they have also become profitable targets for theft and unlawful access. The Mobile Theft Detection & Identification project is focused on developing a robust system to detect and identify stolen mobile devices while closely monitoring related events. This involves a set of measures need to implement identity and security to the mobile devices to prevent from theft or unauthorized access. This includes implementation of several functionalities like Location Tracking, Biometric Authorization, PIN / Password Verification, Real-Time Device Tracking etc. Services like “Find My Device “, “Find My iPhone” are good examples of Theft Prevention Applications that serves as a crucial tool for locating and securing lost or stolen mobile devices, primarily smartphones and tablets. The Find My Device feature enables users to pinpoint the precise location of their device on a map, remotely lock it, play a sound to help locate it, and even erase its data to prevent unauthorized access.

II. Project Overview

In this project, Mobile Theft Detection System is primarily built on The Blockchain Platform. IoT modules are effortlessly integrated inside the mobile devices, providing with a variety of data such as Device Location, User Activities and Device Condition. The Device Identity Management System is built upon Ethereum Blockchain System providing with high level of security and reliability. Once a Device is registered with its IMEI Number, it cannot be changed or modified. When a device is reported stolen, the device will be tracked in real-time and its location will be updated at specific time intervals. Options for remote locking and wiping personal data can be incorporated.

III. Preliminaries

A. Blockchain

Blockchain is implemented to protect the Device Identity. Once the device is registered with its unique IMEI Number and it is appended to the Blockchain, it cannot be modified. All the participants in the ledger will be involved in the decision-making. Hence, subsequent registration with the same IMEI number will result in a failure thus providing transparency, reliability, and security of the information. Moreover, all the transactions are encrypted, and validated by miners in the blockchain network, making it tamper-proof.

B. Smart Contracts

Smart Contracts are Digital Crypto-Contracts which follow certain “if/when. . . then. . . ” criterion, and executes certain actions, when pre-configured conditions are met. In this work, Smart Contracts are implemented to maintain a digital ledger that contains all the transaction details that get executed inside the network. Smart Contracts are immutable, hence they add to the security of the application and state of the ledger, reducing the possibility of fraud.

IV. IWF Architecture for PBL

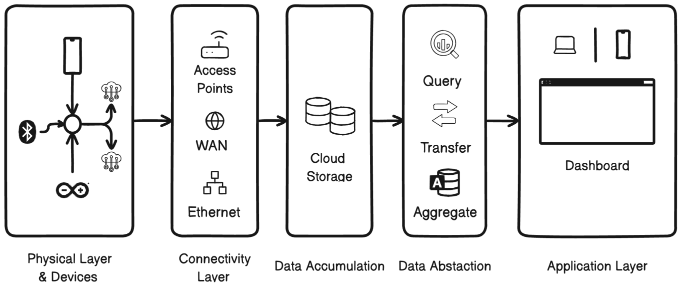


Fig. 1. IWF Model for PBL

The proposed architecture is described below :

- **Physical Devices & Controllers:** The L1 layer of the IWF Architecture consists of the sensor devices that generate data to be passed on to the higher levels. Applications can interact with the devices to query data requests. Our L1 Devices consist of GPS sensors integrated inside of several Mobile Devices that fetch the current location of the Mobile Devices.

- **Connectivity Layer:** Connectivity Layer primarily focuses on the communication between L1 Devices and provides reliable delivery over unreliable Networks. As we are implementing the Entire Network on our localhost, the System itself is acting as an Access Point providing communication between L1 Devices.

- **Data Accumulation:** Data generated from L1 Devices gets stored on this layer. On our PBL, we used Blockchain as our Data Storage application. Each credential of the Device user gets stored in the Blockchain in the form of transactions.

- **Data Abstraction:** In this layer, accumulated data is aggregated and formatted using different techniques such as Filtration, Selection, Reformatting, etc. Data is also protected using appropriate authentication and authorization. The Blockchain handles the part of authentication and Authorization. It utilizes cryptographic keys and digital signatures to authenticate users, devices, or systems, preventing identity theft and fraud. Moreover, Authentication and Authorization can be automated using smart contracts. These self-executing contracts are written in code, allowing verified users to gain access to specific resources or perform actions without manual permission from a central authority.

- **Application Layer:** The Frontend UI of our project acts as an Application Layer that interacts with the Blockchain network to send and receive stored data in form of transactions and helps to generate insight reports. We have :
  1) **Dashboard :** A Dashboard showing the Current State of All Registered Devices (Active / Lost)
  2) **Register Page :** A Registration Page through which users can register their Mobile Devices into the Blockchain Network.
  3) **Login Page :** A Login Page through which registered users be able to Login and change their Device State.

V. Methodology

The chosen PBL would be mainly implemented with the help of an IOT Device, like Android devices, which would be acting as the edge devices. Apart from devices, another important part of the architecture is the blockchain implementation. Figure VIII-B shows the IWF Model which is analogous to our chosen PBL.

- **Device Registration:** Each Device is registered using the Device's IMEI(International Mobile Equipment Identity) Number, Username and Password. The Device Identity is maintained in the Blockchain Network. Once the Device is registered, it cannot be altered or deleted, ensuring the integrity and reliability of the information.

- **Module 1: Alert propagation**
  - **Device Status:** Once a device is stolen, the user can update the Device Status to ”Lost/Stolen” by allowing the user to log in using the same registered credentials onto any other Device. Once marked as ”Stolen”, all the registered users in the Blockchain Network will be notified through the Dashboard.

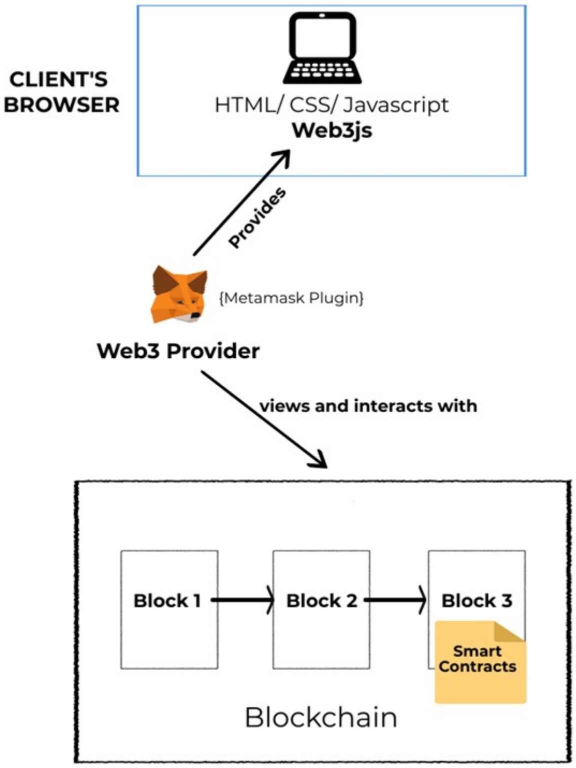


Fig. 2. Blockchain Architecture

  - **IMEI Banning:** If a user tries to register using the ”Stolen” IMEI Number, registration is forbidden, and subsequent actions are taken. Smart Contract in the Blockchain will provide the Identification & Validation of Registered IMEI.

- **Module 2: Theft Detection and Event Monitoring**
  - **Real-Time Location Update:** Once a Device is marked ”Stolen”, device will be tracked in real-time and location will be updated on the User Dashboard in frequent intervals of time.
  - **Data Recovery:** Data Recovery is ensured by regular data backups once the Device is marked as ”Stolen”.

VI. Technologies Used

A. Ganache

Ganache is a private Ethereum blockchain environment that allows developers to create a local network to test smart contracts. In actuality, Ganache is a local development environment for Ethereum blockchain development. You may customize how the chain functions and use it to conduct tests, issue commands, and check the state of your personal blockchain. When creating and testing decentralized apps (DApps) on the Ethereum network, developers frequently utilize Ganache. Before releasing DApps onto the main Ethereum blockchain, it offers a straightforward interface for setting up and maintaining local Ethereum networks, which may be very helpful for development and testing. A brief overview of the system architecture of a Blockchain Network implemented using Ganache is shown in Figure 2. Here's an overview of its key features:

- **Local Blockchain**: Ganache provides a local blockchain environment that runs on your machine, allowing you to interact with it without needing to connect to the Ethereum main-net or a test network like Ropsten or Rinkeby.
- **Ethereum Node**: It acts as a fully functioning Ethereum node, allowing you to send transactions, deploy smart contracts, and interact with the blockchain just like you would on the mainnet.
- **Account Management**: Ganache automatically generates a set of Ethereum accounts for you to use in your development environment. These accounts come preloaded with Ether for testing purposes.
- **Gas Control**: Ganache allows you to control gas settings, such as gas price and gas limit, to simulate different network conditions and transaction fees.

B. Truffle

A complete development platform and toolkit for creating decentralized apps (dapps) on the Ethereum blockchain is called the Truffle Suite. It provides an array of potent tools that optimize the whole development process, from the design and testing of smart contracts to their deployment and asset administration. The mainstay of the Truffle Suite, Truffle, offers developers an easy-to-use environment for creating, assembling, and implementing smart contracts with efficiency. Truffle has its own SDK for building, testing, and deploying contracts in Solidity, JavaScript, and TypeScript.

C. Metamask

Metamask is a decentralized software cryptocurrency wallet that allows users to interact through the local blockchain using a web browser or a mobile app. Metamask provides the Digital Signature for every transaction before getting added to the Blockchain network.

D. Solidity

Solidity in an object-oriented programming, statically-typed language created specifically for Ethereum Network for constructing and developing Smart Contracts.
Smart contracts are high-level computer codes that are posted on the Ethereum blockchain for further execution after being compiled to EVM byte code. It enables us to carry out reliable transactions without the intervention of a third party; these transactions are irreversible and traceable.

E. Web3

Web3 is often described as a series of open-source and interconnected decentralized applications powered by blockchain computing architecture. Web3 aims to decentralize the internet by leveraging blockchain technology, smart contracts, and decentralized protocols. Web3 is used to fetch/write the data from the blockchain through Smart Contracts and display the result on the Browser page.

F. ReactJS

Developed and maintained by Facebook, ReactJS, is an open-source JavaScript library. It's primarily used for building User Interfaces (UIs) and Frontend Logic for web applications. Here are some key features and concepts associated with ReactJS:

- **Component-Based Architecture**: React allows developers to build UIs using reusable components. Each component represents a piece of the UI, encapsulating its own logic and rendering output. Components can be composed together to create complex user interfaces.
- **JSX (JavaScript XML)**: JSX is a syntax extension for JavaScript that allows developers to write HTML-like code within JavaScript. JSX makes it easier to define UI components and their structure directly within the JavaScript code.
- **Declarative Syntax**: React promotes a declarative programming style, where developers describe the desired UI state and React takes care of updating the DOM to match that state. This approach simplifies UI development and makes code easier to understand and maintain.
- **Auth and Route Protection**: React comes with it's routing protocol suite, React-Router-Dom, which makes it easy to declare private and public routes within an application. React provides the functionality of authentication based routing to access private routes.

VII. Implementation

A. Setting Up Environment

1) Ganache is installed that will act as the local blockchain network for this decentralized Application.
2) Truffle for Smart Contract Construction, Management and Deployment.

B. Smart Contract Development

Smart Contracts are deployed using Solidity programming language. Smart Contract contains the functionalities:

1) Structure to store IMEI, UserID, Password and LostStatus of the Registered Device.
2) Register Function that adds Device Information to the BlockChain.
3) Login Function that checks for UserName Password match in the registered devices. Returns boolean value.
4) getAllRegisteredPhones() that returns the IMEI values of all registered devices along with its lost status.
5) reportLost() that changes the lost status of the given IMEI numbered Device when triggered.

C. UI Development

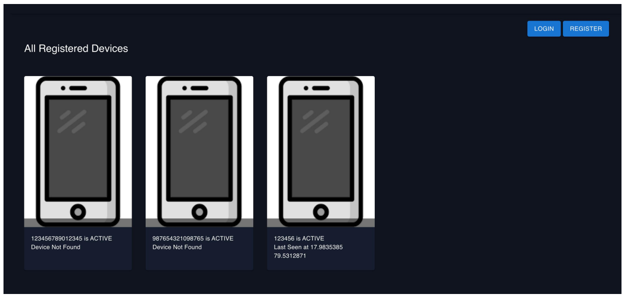
ReactJS has been used for UI Development. A number of other NPM modules were used for Authentication and communicating with the smart-contracts.

1) Initialise a React boilerplate code using create-react-app CLI.
2) Install modules like react-router-dom(for routing), web3(for communicating with smart contracts), material-ui(for styling purposes).
3) Initialise the Login/Register, Homepage and Dashboard Routes by declaring private and public routes logic.
4) Add Content in each page and stylise using MaterialUI.
5) Initialise a web3 variable and declare the function to interact with the smart-contracts.
6) Write tests to check whether the functions are returning proper output or not.
7) Finally integrate all components and run the project on localhost:3000 for demonstration purpose.

VIII. Visual Representation

A. Public Dashboard

Public Dashboard contains all the registered Devices along with their Current Status (Active/Lost) and their Current Location if lost. This page is for visualization purpose only, infromation cannot be modified here.



All Registered Devices

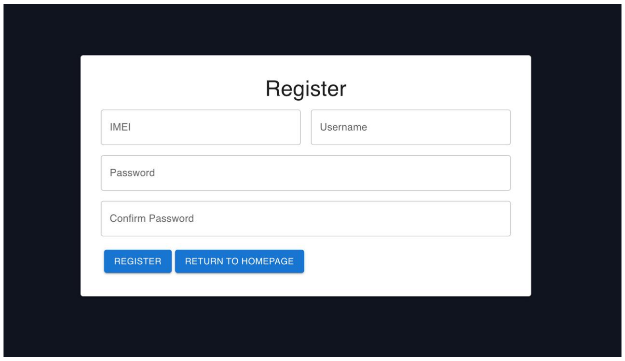
123456789012345 is ACTIVE
Device Not Found

987654321098765 is ACTIVE
Device Not Found

123456 is ACTIVE
Last Seen at 17.9835385
79.5312871

B. Registration Page

Users have to register their devices by providing their Unique Device IMEI along with preferred username and password. Registration with an already registered IMEI will result in a failure.


Register

IMEI

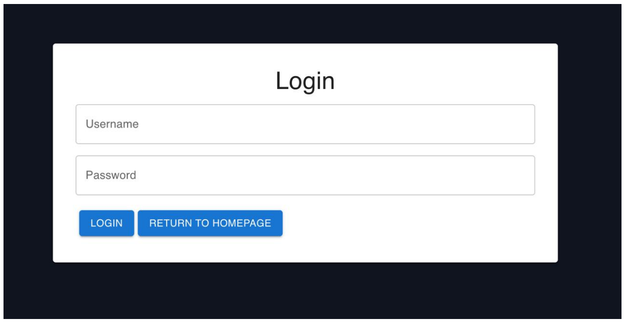
Username

Password

Confirm Password

C. Login Page

Users will enter their registered Username along with their password. This will be verified with the data entered into the blockchain and will result in success or failure.

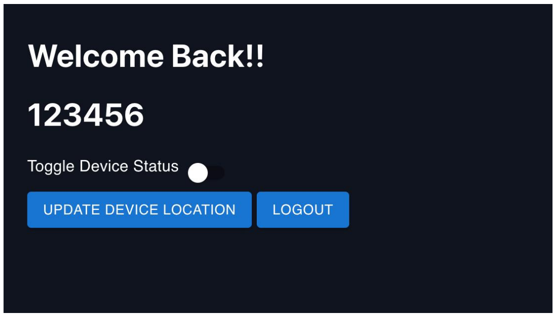

Login

Username

Password

D. User Dashboard Page

Users have to Login to change their Current Device status and to get Updated location co-ordinates that will be displayed in the Common Dashboard and will be visible to all users.

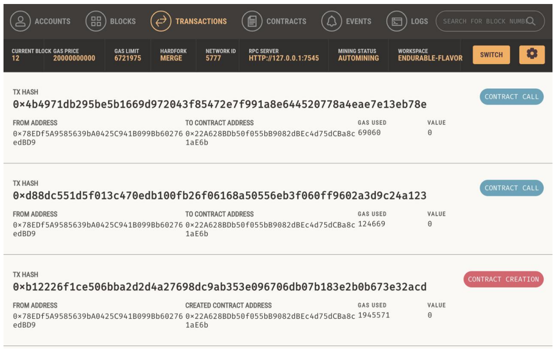

Welcome Back!!

123456

Toggle Device Status

E. BlockChain - Ganache

Every transaction, from Contract Deployement to Contract calls are getting updated and visalized in the Blockchain Network.



```
CURRENT BLOCK  GAS PRICE    GAS LIMIT  HARDFORK  NETWORK ID  RPC SERVER              MINING STATUS  WORKSPACE
12             20000000000  6721975    MERGE     5777        HTTP://127.0.0.1:7545   AUTOMINING     ENDURABLE-FLAVOR

TX HASH
0x4b4971db295be5b1669d972043f85472e7f991a8e644520778a4eae7e13eb78e
FROM ADDRESS                                 TO CONTRACT ADDRESS                          GAS USED  VALUE
0x78EDf5A9585639bA0425C941B099Bb60276        0x22A628BDb50f055bB9082dBEc4d75dCBa8c        69060     0
edBD9                                        1aE6b

TX HASH
0xd88dc551d5f013c470edb100fb26f06168a50556eb3f060ff9602a3d9c24a123
FROM ADDRESS                                 TO CONTRACT ADDRESS                          GAS USED  VALUE
0x78EDf5A9585639bA0425C941B099Bb60276        0x22A628BDb50f055bB9082dBEc4d75dCBa8c        124669    0
edBD9                                        1aE6b

TX HASH
0xb12226f1ce506bba2d2d4a27698dc9ab353e096706db07b183e2b0b673e32acd
FROM ADDRESS                                 CREATED CONTRACT ADDRESS                     GAS USED  VALUE
0x78EDf5A9585639bA0425C941B099Bb60276        0x22A628BDb50f055bB9082dBEc4d75dCBa8c        1945571   0
edBD9                                        1aE6b
```

The implementation video has been uploaded in GDrive, which can be viewed at Link :MobileTheft.mp4.

IX. Modular Objectives

1) **Module 1:**
2) **Module 2:**