

Machine Management

Computer Networks – Term Project

December 10th, 2022

Juan A. (armijosj@myumanitoba.ca)

Contents

The Idea	3
Problem.....	3
Background.....	3
Relevance.....	3
Objective.....	3
Development:	4
Technologies employed:.....	4
Solidity	4
Truffle	4
Ganache	4
MetaMask	4
NodeJS.....	4
Ethers.js	4
Putting it together	4
How to run it:.....	5
1. Deploy the blockchain	5
2. Run the application	6

The Idea

Background

Machine maintenance software is a system employed in industries that allows supervisors and operators to plan maintenance tasks in an organized and systematic way. Operators enter a request for a machine, and maintenance workers fulfil those requests by doing the work, logging the information in the software, and closing the request. There are tasks that require immediate action, such as fixing, and therefore the scheduled tasks are often left unfinished.

Problem

Industries frequently lack maintenance workers. Therefore, the software gets manipulated by closing requests when, in reality, they have not yet been fulfilled. There is neither accountability nor transparency when revisiting the requests.

Relevance

When maintenance tasks are skipped or faked, there is a great chance of machines getting broken. The machine breaking means stopping maintenance for a fix, which blocks further maintenance. Generating an endless loop of unproductivity.

In a highly productive industry, a machine stoppage, even for a couple of hours, results in a loss of thousands of dollars. There is an even more significant concern about potential harm when a machine breaks. It could endanger employees' safety, resulting in a lawsuit or even the death of one employee.

Objective

This project seeks to address this issue by offering cutting-edge technology and solutions. Smart Contracts and Blockchain will help to achieve this by providing the business with transparent and responsible software. The proposed smart contract will allow all employees to oversee every action and spot any attempt at manipulation.

Development:

Technologies employed:

Solidity

Link: <https://docs.soliditylang.org/>

Object-oriented, high-level language for developing smart contracts.

Truffle

Link: <https://trufflesuite.com/>

Framework for managing smart contract development.

Ganache

Link: <https://trufflesuite.com/ganache/>

Free to use Ethereum blockchain deployer. Owned by Truffle. Acts like a personal blockchain.

MetaMask

Link: <https://metamask.io/>

A crypto wallet & gateway to blockchain apps. It is the communication media between the DApp (distributed application) and the smart contract.

NodeJS

Link: <https://nodejs.org/en/>

Used for web application development.

Ethers.js

Link: <https://docs.ethers.org/v5/>

JavaScript library for interacting with the Ethereum Blockchain and its ecosystem.

Putting it together

The smart contract is written in Solidity. Truffle then deploys the Smart Contract onto a personal Ethereum blockchain running in Ganache. MetaMask is the communication channel between the smart contract on the blockchain and the web application. The web application runs with the NodeJS framework and uses the Ether.js library to call the smart contract.

Prerequisites to run:

- Download Ganache.
- Download Truffle Suite.
- Download NPM.
- Download NodeJS.
- Create a MetaMask Account and add the extension to Chrome.
- Clone the GitHub repository

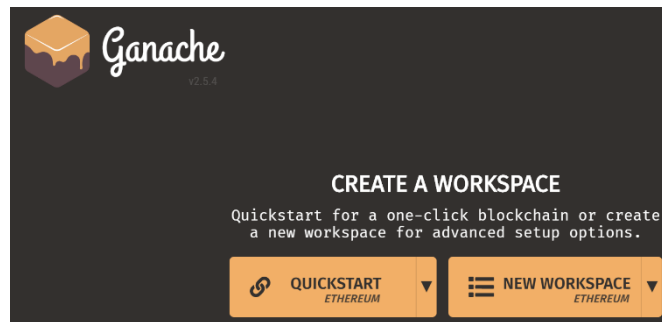
How to run it:

Please be sure to have completed all prerequisites before starting the following steps.

These are the steps that are followed in the demo video.

Deploy the blockchain

1. Open Ganache and run a personal Blockchain by clicking on QUICKSTART



2. Open a command line in the repository folder and type the following command:
truffle migrate
3. The Smart Contract MachineManagement is now deployed. We now need to copy the address
4. In the command line type: **truffle console** and press enter.
5. Then type: **myvar = await MachineManagement.deployed()**
6. Next, type: **myvar.address**
7. Copy the address.
8. Press 'Ctrl+c' two times to exit the truffle console.

Here is an image of how it should look like:

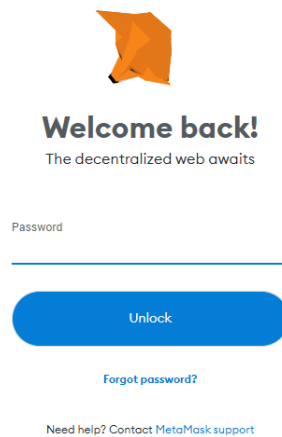
```
C:\Users\Juan A\Desktop\Personal Projects\comp-4300-project>truffle console
truffle(development)> myvar= await MachineManagement.deployed()
undefined
truffle(development)> myvar.address
'0xb27efc32DaA574146383A50B48fe5c655f3d1bc6'
```

1. Run the application

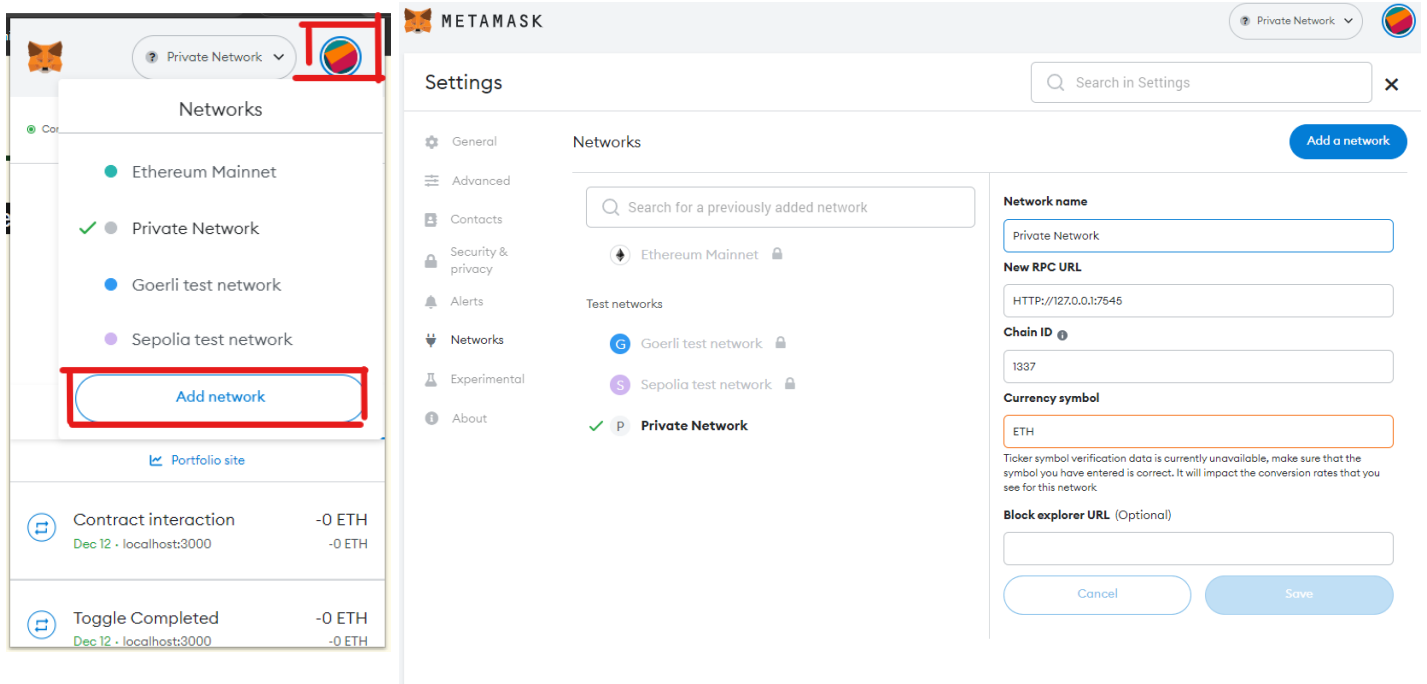
1. Paste the address into the first line of the App.js file under the folder: src.

```
src > JS app.js > ...  
1  const myContractAddress = "0x127f1b918723e704c2b5Fa902E5c5833AEC1FB97"; // <- paste address here  
2  
3  window.myContract = null;
```

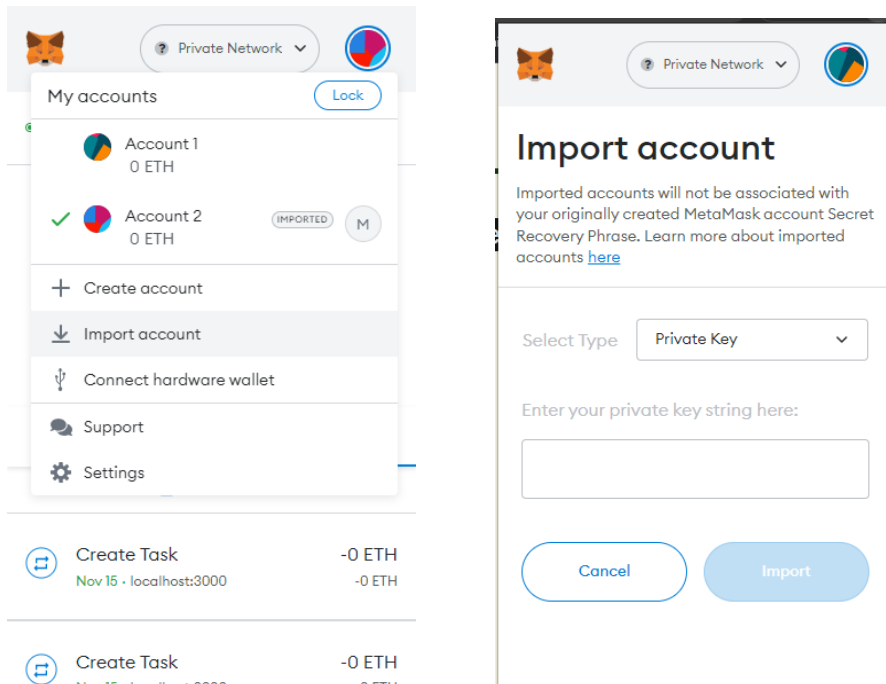
2. Open the console and type: **npm run dev**
3. The web application will appear on the default browser (developed and tested in chrome).
4. Then the MetaMask extension window will appear asking for credentials.



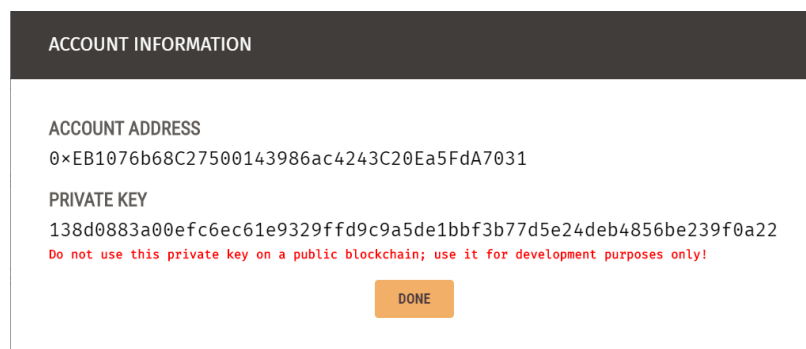
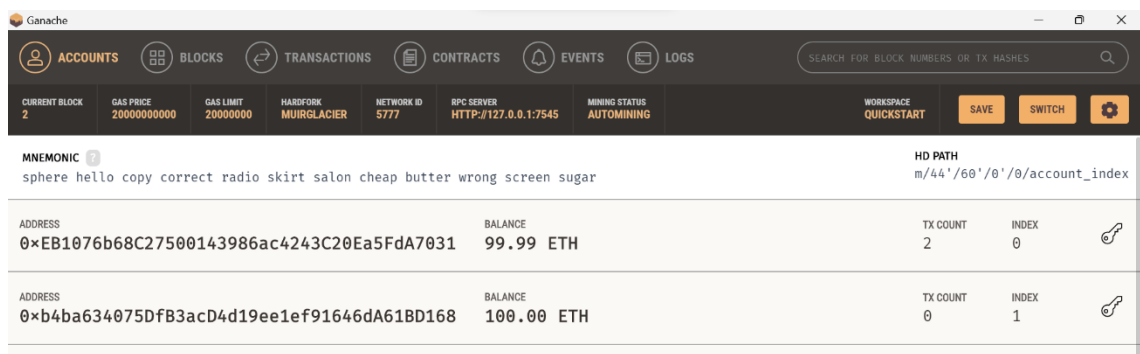
5. Add the private network with the following settings (if not already there):
 - Click on the top right corner icon, then on Add network.
 - On the settings page make sure host and port are: [HTTP://127.0.0.1:7545](http://127.0.0.1:7545) and Chain ID: 1337



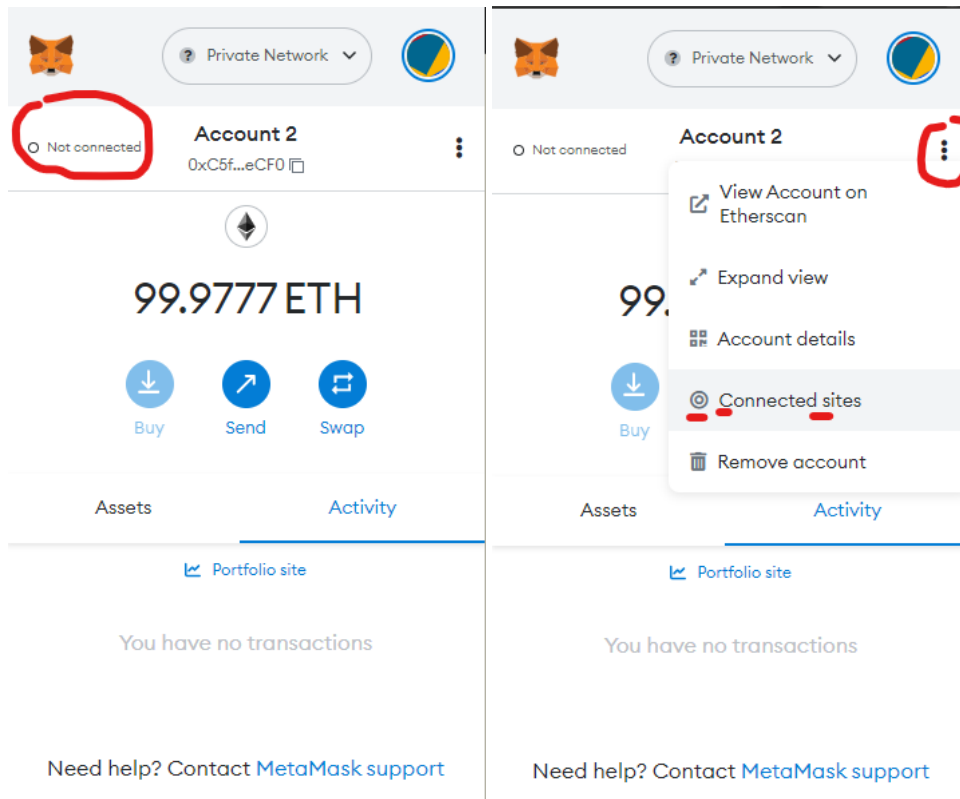
6. The next step is to import a new account using a private key.



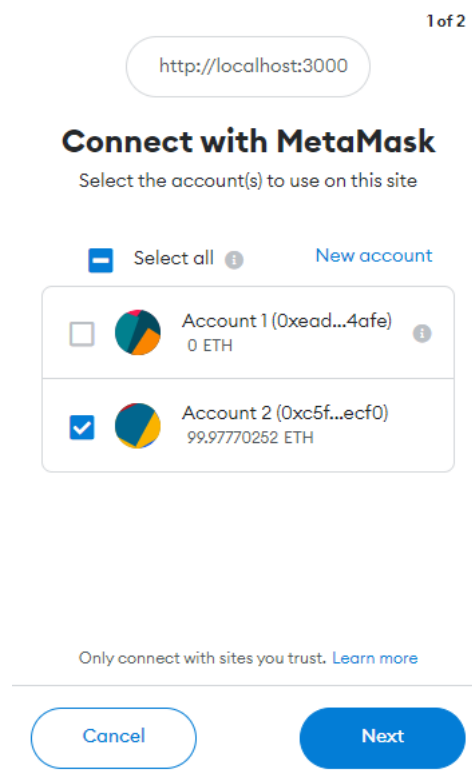
7. The private key is given by Ganache. Click on the key icon to the left of the account that 99.99 ETH.



9. Then Paste into MetaMask. Make sure it shows connected. If not, click on the connect button. After all the steps the MetaMask window should look similar like this:



10. Select Manually Connect to this Site and then click on Next and finally Connect:



How to use it:

1. Click on 'Create' and then fill out the form and click 'Submit'.

Machine Management

Tasks: Refresh Create

Type:

Machine:

Reason:

Submit Cancel

2. Click on 'Confirm' on MetaMask.

MetaMask Notification

Private Network

Account 3 → 0x9D6...8B24

New address detected! Click here to add to your address book.

http://localhost:3000

0x9D6...8B24 : CONTRACT INTERACTION ⓘ

DETAILS DATA HEX

EDIT

Estimated gas fee ⓘ

0.00000233

0.000002 ETH

Site suggested

Max fee: 0.00000233 ETH

Total

0.00000233

0.00000233 ETH

Amount + gas fee

Max amount: 0.00000233 ETH

Reject Confirm

3. Check the checkbox for completing a task. And Click Confirm on MetaMask again.

Machine Management

Tasks: Refresh Create

ID: 0	Type:Fix	Machine: Welding	Reason:Broken Pipe	<input checked="" type="checkbox"/>
-------	----------	------------------	--------------------	-------------------------------------

Conclusions and Final Remarks:

The proposed idea increases trust, security, transparency, and the traceability of data shared across the business network. As we can see, we get a history of activity that cannot be modified. This results in a better sense of accountability for employees. Additionally, it will build a sense of community and support since not only supervisors have control over the data and the tools, but everybody in the company will feel as important as the rest.