Internship - 1

Smart contract

1. Solidity - High level programming language
2. Ethereum virtual machine (EVM) - a lightweight operating system that is created to run smart contract
3. Solc compiler – compile solidity, translate solidity code into bytecode (which EVM can understand)
4. Bytecode – a low-level programming language which is compiled from a high level programming language
5. Application Binary Interface (ABI) – an interface to interact with EVM bytecode and Javascript code

Deploy and Run Transaction

Environment:

1. JavaScript VM: All transactions will be executed in a sandbox blockchain in the browser. Nothing will be persisted .
2. Injected Web3: Mist and Metamask are example of provides that inject web3
3. Web3 provider: Remix will connect to a remote node. You will need to provide the URL address to the selected provider (geth, parity, Granache)

Web3

- Is a collection of libraries which allow you to interact with a local or remote Ethereum node, using http or IPC connection.
- Interacts with the Ethereum blockchain
- Use to retrieve user accounts, send transactions, interact with smart contract, etc

Migrations

- Javascript files that help you deploy contracts to the Ethereum network.
- Truffle requires you to have a Migrations contract in order to use the Migrations feature

Visual studio code

Command:

```
npm run dev
```

Tasks

1. Obtain data from serial port, and store data in blockchain
2. Setup DAPP to not use Metamask plugin, but sign transaction at backend
3. Obtain transaction hash without the need of additional mutator function to store the transaction hash
4. Wifi scheduler for IoT devices

- Payment Form
- Archive
- Presentation and Next meeting (13.07.2020, 10am)