A Project Report

On

# Blockchain technology for IOT access control

BY

**Vashi Chaudhary**

**2018A7PS0243H**

Under the supervision of

**Prof. G Geetha kumari**

**SUBMITTED IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS OF**

**CS F376: DESIGN PROJECT**

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)**

**HYDERABAD CAMPUS**

**(December 2020)**

# ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my DOP faculty prof G Geethakumari for her continous help and support. Her understanding, guidance and enthusiasm has helped me throughout this course, and significantly aided me towards the completion of this report.

Secondly, I would like to thank Ms. Rashmi Sahay for her constant help. Her utmost support helped me overcome every obstacle that I faced during the project and imparted me with all the knowledge required for the project.

Last but not the least, I would like to thank the institution, BITS Pilani, for arranging this DOP course for its students, and giving us the opportunity to boost our industrial exposure and develop various technical as well as soft skills.

**Birla Institute of Technology and Science-Pilani,**

**Hyderabad Campus**

**Certificate**

This is to certify that the project report entitled "**Blockchain technology for IOT access control**" submitted by Ms. VASHI CHAUDHARY (ID No. 2018A7PS0243H) in partial fulfillment of the requirements of the course CS F376, Design Project Course, embodies the work done by her under my supervision and guidance.

**Date:** **(PROF G. GEETHAKUMARI)**

BITS- Pilani, Hyderabad Campus

# ABSTRACT

As the IoT system architecture expands and grows in the coming years, the data increases exponentially. With the increase of hardware devices in an IoT system, the wireless intercommunication between them becomes unsecure. What data should be allowed to be accessed by which device depends on the access control mechanism of the IoT architecture. The data interchange between IoT devices is completely wireless and there is a possibility of data leak or data tampering in the process of transmission. The data that is being transmitted over IoT devices can contain private data and thus, the transmission needs to be secure. The traditional methods of access control become useless as the amount of data grows. The new technique of involving blockchain technology can be used to solve this problem. Blockchain technology is a decentralized peer-to-peer mechanism which can be used between the devices of IoT. Blockchain is currently being used in domains like cryptocurrency, healthcare systems, supply chain management etc. When data is being transferred between two IoT devices, the access control mechanism can be layered with a blockchain layer where the blockchain layer ensures that the devices involved in the transaction are correct,  using encrypted public and private keys. There are various researches going on in this field, some of which have been presented in the report.

The topic of this report is “Blockchain technology for IOT access control” and it presents a detailed introduction about Internet of things, its use, and blockchain technology and how blockchain can be used for IOT access control. A literature survey is presented about the past techniques available in the domain after that. The literature survey is followed by a model implementation of the proposed idea for access control using smart contract and web3 library of python.

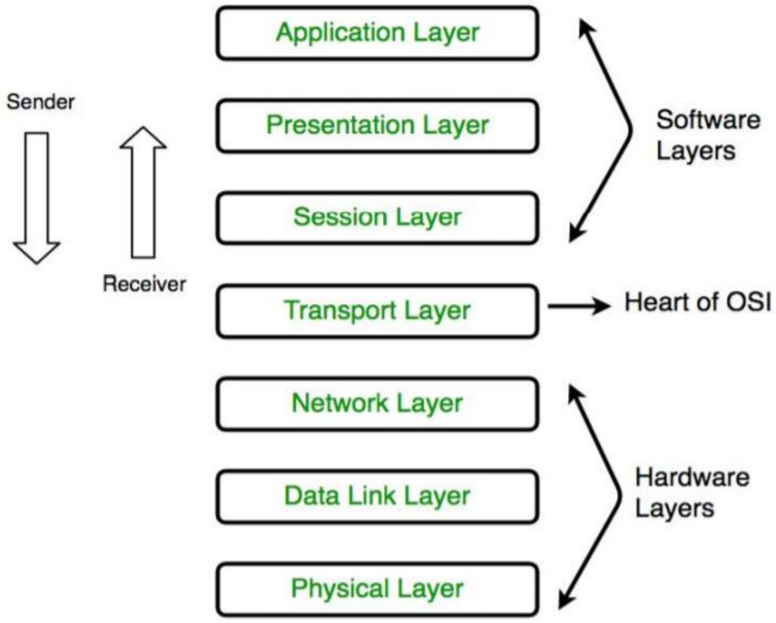
# CONTENTS

# INTODUCTION

## INTERNET OF THINGS (IoT):

The IoT is a network of physical objects that are embedded with sensors, softwares, actuators and other technologies for connecting and interchanging data with other devices in the system. The IoT architecture is involved in various tasks like machine learning, real-time analytics, embedded systems etc. The IoT architecture is being extensively used in healthcare management systems and home automation. It is estimated that there will be around 41.6 billion connected devices by 2025.

For understanding IoT to its core, we need to understand the 7 layered Open System Interconnection (OSI) model. This model uses layers to define what is actually going on in a particular network system. The 7 layers are – Application, Presentation, session, transport, network, data link and physical. The application layer is the topmost layer. It is the layer that the users see and use to interact with the Iot system. The presentation layer is used for preparation or translation of application format to network format. The session layer is used as a medium to coordinate setup and terminate applications at the end of each session. The transport layer is used for data transfer between the system and hosts. It decides what data to send, at what speed etc. The TCP/IP is a part of this layer. The network layer is used for packet forwarding and routing through different routers. The data link layer is used for node-to-node transfer and also handles error correction from physical layer. The physical layer is the bottom-most layer of the OSI model. The layer is used for the electrical and physical representation of the system. The electrical cables, switches etc. are a part of this layer.



# INTERNET OF THINGS- ARCHITECTURE

IoT is a new model that contains all wireless networks, actuators and mobile networks where each component has a unique address and they communicate amongst each other using radio-Frequency identification devices (RFID). This communication is void of human interaction and has become a prevalent part of our lives.

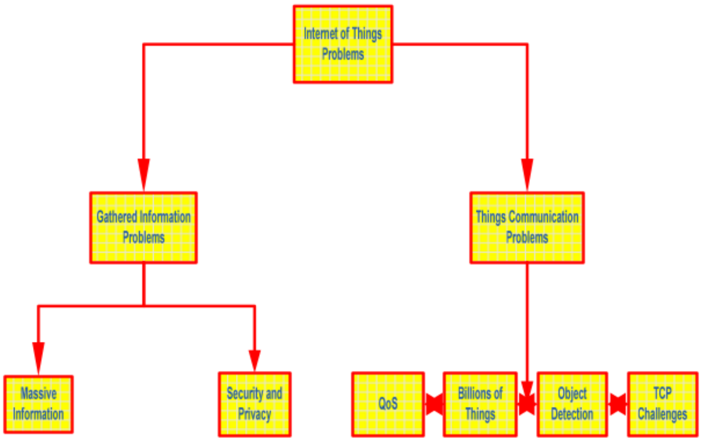
There are two proposed architectures of IoT.

1. 3-layer architecture
2. 5-layer architecture

**3-layer architecture:** This architecture contains 3 layers- perception, network and application. The perception layer is used to identify each device of the system using its IP address using RFID tags, sensors and cameras, etc. The network layer is used to transmit the information gathered by the perception layer. The application layer is used to converge between IoT social needs and industrial technology.

**5-layer architecture:** this has 5 layers called - perception, transport, processing, application and business. The business layer is used to define the IoT applications and charge and management. This layer is also responsible for user's privacy and research related to IoT applications. The applications layer is used to determine the type of applications of IoT and is used to develop intelligent authentic and safer IoT applications. The processing layer is used to handle(store and analyze) the information gathered by the perception layer. The transport layer is similar to the network layer in the 3-layer architecture. It acts as an intermediate between the perception and processing layers. The perception layer is used to define the physical meaning of IoT architecture.

Other special-purpose architectures include - media aware traffic control security architecture, clock synchronization network and mankind neural network. There are a lot of challenges faced by an Iot system. The challenges can be classified as gathered information problems and Things communication problems.



- ❖ INFORMATION GATHERING PROBLEM -

  1. Massive gathered problem - Due to the presence of high number of IOT devices, the information gathered is huge. So, there is a problem is of storing, transmission and processing of big data.
  2. Security and privacy - The data is transferred through a wireless medium and hence the chances of tampering of data are huge. The Iot system needs to be resilient to attacks, data authentic and private.

- ❖ THINGS COMMUNICATION PROBLEMS -

  1. Billions of IoT things - Considering the hugeness of IoT architecture, the hardware and precision required in the transmission of data pose a huge problem.
  2. IoT and TCP challenge - Hugeness and privacy are the issues involved.
  3. Real-time object detection - The information received by the IoT system needs to be processed in real-time ie. the processing needs to be fast, but the data is huge.

The suggested IoT database architecture contains six layers -

1. The IoT layer - contains sensors, PDA, desktops, cameras, actuators etc.
2. Data collection layer- this layer collects the objects' data. Different collection, reading and correcting strategies are used here.
3. Data warehousing layers - this contains object identification, data abstraction and compression.
4. Event processing layer - Used to analyze events in IoT effectively. This contains event-based queries or conducting event analysis.
5. Data mining service layer - This depends on the data warehousing layer and event processing layer. This layer is a service-oriented data mining layer.
6. Application layer- this contains three layers- local manager sub-layer, general manager sub-layer, inelegant applications sub-layer.

# Radio Frequency Identification (RFID)

An RFID system is used to automatically sense an object. An RFID system has two parts.

1. Reader - A reader sends out and receives radio signals to and from the RFID tag.

2. Tag - An RFID tag is attached to the object. Whenever the object is in the range of the RFID reader, the tag transmits a feedback signal to the reader. The tag is of three types -

a) Passive - does not have its own power supply and is dependent on the signal from the reader as the source of energy.

b) Semi-passive - has a power supply but also depends on reader signals for energy to transmit the signal back.

c) Active - completely runs on its own power supply and does not consume the energy of the reader signals.

## Wireless Sensor Networks (WSNs)

A Wireless sensor network can be defined as a network of devices that can communicate the information gathered from a monitored field through wireless links. The data is forwarded through multiple nodes, and with a gateway, the data is connected to other networks like wireless Ethernet.

All motes are composed of five main elements:

1. Processor - A processor works in three modes of operation

    a) Sleep - to save power

    b) Idle - data can arrive from other motes

    c) Active - Data is sent to other motes

2. Power Source - eg rechargeable battery, Solar panels, capacitors. 3. Memory - A memory is used to store programs and data

4. Radio - Low rate(10-100 kbps), short-range(<100m)

5. Sensors - Can be physical, chemical or biological. Used for sensing temperature, humidity, light, pressure, noise, acceleration, soil moisture, etc

## Internet Protocol version 6 (IPv6)

IPv6, developed by the Internet Engineering Task Force (IETF), deals with the problem of IPv4's incapabilty of handling a million devices.. IPv6 is a 128-bits address having an address space of 2^128, which is way bigger than IPv4. In IPv6 we use Colon-Hexa representation. There are 8 groups and each group represents 2 Bytes.

## 6LoWPAN

6LoWPAN provides a means of carrying packet data in the form of IPv6 over IEEE 802.15.4 and other networks. It provides end-to-end IPv6 and as such, it is able to provide direct connectivity to a huge variety of networks including direct connectivity to the Internet.

## Applications of IoT

1. Transportation and logistics domain - Assisted driving, Monitoring environmental parameters, Augmented maps etc.

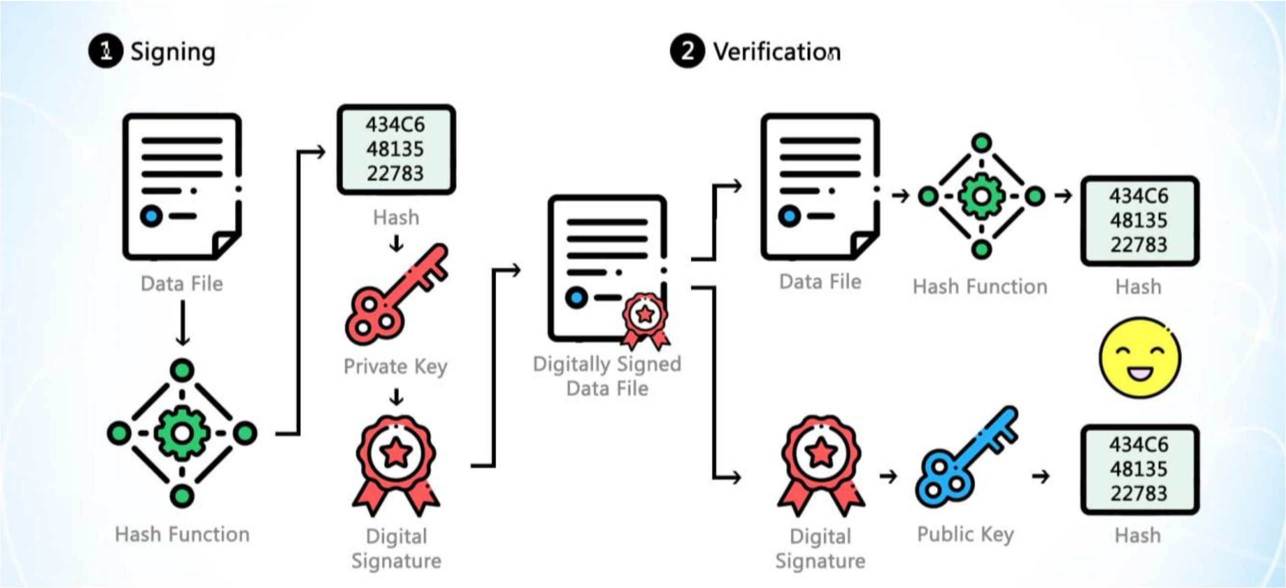
2. Healthcare domain - tracking, authentication, data collection, sensing

3. Smart environment domain - comfortable homes/offices, smart museums and gyms

4. Personal and social domain - social networking, losses thefts 5. Logistics, Assisted driving, Mobile ticketing, Augmented maps.

# BLOCKCHAIN AND ACCESS CONTROL

Blockchain is a peer-to-peer decentralized public ledger. It contains a list of records, called blocks which are linked to each other using encryption. Each bloack has a hash value from the previous block, a timestamp and the transaction value/data. The data in any of the blocks can’t be changed easily. Tampering needs to have a minimum 51% support in the blockchain. Once a block is altered, it will need to change all the further vlocks ie. Altering any data is next to impossible. Blockchain is being used in various domains like healthcare, supply chains and vank replacements etc.

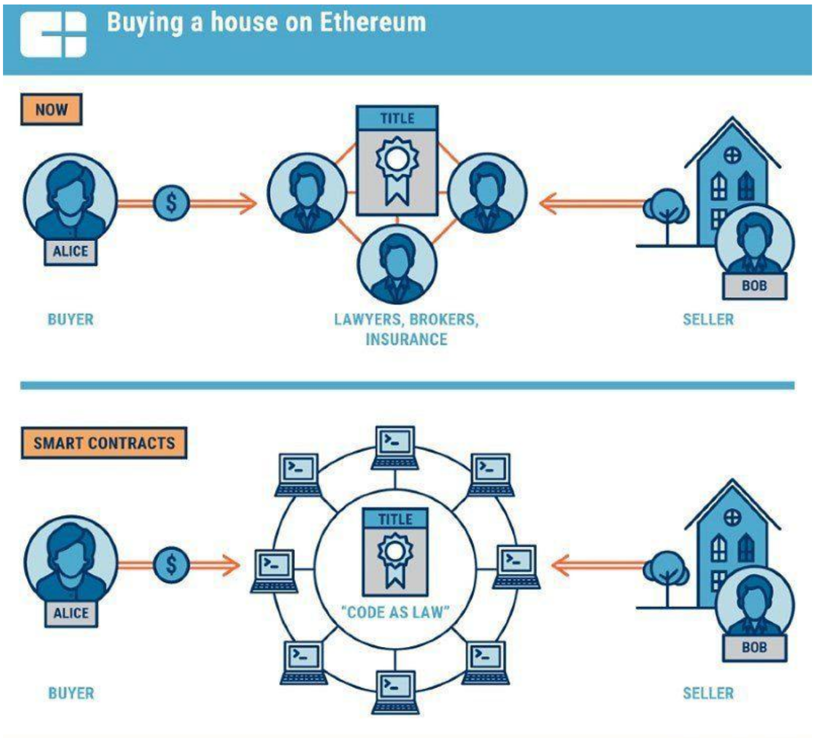


Access control can be achieved using an edge device that can determine whether the incoming queries are from a device that is authorized to exercise the privilege of sending these commands and the edge device has the ability to confirm that authority. The idea is that these edge-based devices can create unforgeable capabilities that they can issue to specific interlocutors. This enables safe communication between edge devices and controllers. A mapping from domain name to IP address can be validated when an edge device needs to communicate with the controller. Whenever the controller wants to communicate with the controlled device, it must return the nonce, encrypted for the device public and private key. The same process can be carried out by any two devices.

Blockchain can play an important role when it comes to access control in IoT systems. The transmission of data in between two IoT devices can be secured using Blockchain technology. Each device in the Blockchain can be given a public/private key. Whenever a transaction has to take place, the public and private keys of the sender and recipient devices are checked before each transaction. Only if the verification gives positive result, the transaction is fulfiled. The data being carried in the transaction can also be encrypted to bring more security. This process is completely decentralized and keeps the data and transactions safe and secure.

# ETHEREUM AND SMART CONTRACTS

**Ethereum** is a decentralized platform that can run smart contracts i.e. it can run programms exactly as programmed without any possibility of downtime, censorship, fraud or third party. Ethereum is open software based on blockchain technology. Ethereum is open access to digital money and data-friendly services for everyone – no matter your background or location. It's a community-built technology behind the cryptocurrency ether (ETH) and thousands of applications you can use today. Ethereum lets you send cryptocurrency to anyone with a small amount of fee. It is a development of bitcoin which is programmable. Ether is the cryptocurrency used on the ethereum platform. Ethereum can be used to deploy smart contracts on the main net, test nets or even private blockchains.



**Smart Contracts** are some computer programs or transaction protocols which execute automatically, control or document legal events and actions according to the contract. Smart contracts are used to reduced intermediates, frauds, losses etc. The best example of a smart contract is a vending machine. As soon as you input money and a enter a valid product code, it dispenses the desired product. A smart contract can be deployed on a blockchain by sending a transaction from a blockchain wallet. The transaction uses a compiled code of the smart contract and special receiver address. Our project uses smart contracts deployed on ethereum for the sake of this project.

# LITERATURE SURVEY

[1] **The Internet of Things: A Review of Enabled Technologies and Future Challenges**

IKRAM UD DIN et al. (2018) have presented a detailed summary of the existing IoT technologies and their future challenges.IoT is an emerging classical model, containing millions of interconnected devices for finding state-of-art findings to real-world glitches. IoT has become an inevitable part of our lives and is being used for everything around us. IoT is being used in smart cities for data management and security issues, Heterogeneous IoT for data aggregation and privacy protection, Fog computing for caching, information forwarding, Data mining, WSN-based data-centric IoT, IoT based cellular communications, context awareness-based IoT, Virtual object-based IoT and real-time analytics using IoT. IoT has tried to bridge the gap between humans and Information technology. IoT models assume a cyber-physical world where everything is originated, driven, intermixed and facilitated for emergence of any feasible association.

[2] **A roadmap for security challenges in the Internet of Things**

Arbia Riahi Sfar et al. (2018) proposed a roadmap for security challenges in IoT. The first problem lies in the fact that there are billions of IoT devices that interact together in a complex manner. Secondly, IoT devices have different operational backgrounds and environments with limited computational powers. Thirdly, IoT interactions with thousands of nodes pose a big security problem. These questions need to be addressed before IoT becomes a reality otherwise the privacy of many will be at stake. The paper defines IoT as a tetrahedron relationship. The nodes in this tetrahedron are person, process, intelligent objects and technological ecosystem. The edges are represented by privacy, Truest, access control, readability, safety, auto-immunity and responsibility. Access control; is an edge between a person and intelligent nodes, which emphasizes the means to establish connections among entities and retrieve them easily using identifiers. As correct identification is important for any task in IoT, The access control task needs to be safe and secure.

[3] **An Access Control Scheme based on Blockchain Technology**

M. Laurent et al.(2018) have proposed an access control scheme based on block chain technology. This approach comes as a result of data security and privacy in IoT driven systems. These domains pose a big challenge in the adoption of remote data storage applications, majorly because of loss of data control. And the possibility of third-party data storage providers. The paper implements two approaches based on permission less blockchains. The first approach uses bitcoin as a base framework and modifies it for the purpose. The other approach builds a blockchain from scratch with all the desired features. Blockchain is used for the purpose because of the very basic nature of a blockchain. Blockchain is used as an access moderator that permits to distribute authorization tokens and only authorized signatures are given access tokens. This approach also allows to dynamically delete the granted privileges when needed. This process is used for security and privacy in IoT systems.

[4**] Access Control Scheme Based on Combination of Blockchain and XOR-Coding for ICN**

Xiaobin Tan et. al(2018) present a new approach that combines blockchain and older access control schemes together. This research comes as a response to address-centric networking issues. This research proposes novel network architecture, referred to as Information-centric. Networking. The system architecture consists of control providers, Internet service providers and users. The architecture allows each user to obtain the data required by sending interest to the network. The content provider returns the data. This data is stored on middle routers. When another user wants this data, the nearest router would return the data. This data is stored in a blockchain. The data cannot be accessed on an end-to-end basis, it happens on the basis of the content prefix. Each user has its own public/private key and one authorized user can't share the published data with another user. The architecture involves 4 steps. One, the users should register to content providers and the content providers give them public/private key in return. Second, the content providers generate data decryption information. Third, the user can get the latest blockchain from ICN by sending interest. Fourth, the user searches for a decryption key using their private key. A level of XOR based encoding and decoding can be applied to this entire system.

[5] **Blockchain-based access control**

Damiano Di Francesco Maesa(2018) have presented a blockchain based approach for access control. Access control is a dynamic process, where a device is given permission/denied to access certain data based on certain context. This is real-time process and it happens each time a devices asks for access. This paper presents a novel approach that uses blockchain for access control and distributed transfer of these rights amongst users. These policies and rights are publicly visible on a blockchain. And hence, any user can access that policy at any time and the devices that have an access in that context. This is a distributed, fraud-free system for accepting/ denying access rights of devices using a set policy. The implementation of this research is based on XACML policies and it used the bitcoin blockchain framework to achieve the same. This research presents a safe and secure way of data transmission in IoT architectures. It can be deployed on various use cases.

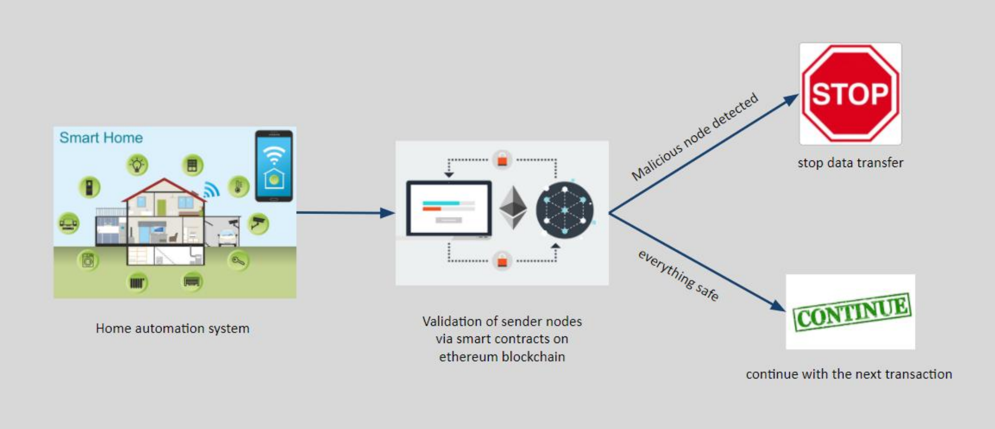
# IMPLEMENTATION

## PROPOSED MODEL

The project “IoT access control using Blockchain” demands us to bring up a new approach to securing access control using blockchain. The proposed model builds an IoT architecture and extracts the radiolog files from the same. A smart contract is then created and deployed on a private blockchain. This smart contract allows the owner of the smart contract to add valid nodes to the whitelist on the blockchain. The data retrievers can check if a data sending node is a valid node or not using the check method in the smart contract. If a malicious node is found( the function check outputs a 0), then the data transfer can be stopped and it can be assumed that some sort of access control breach has taken place.

The implementation of the project contains four parts:

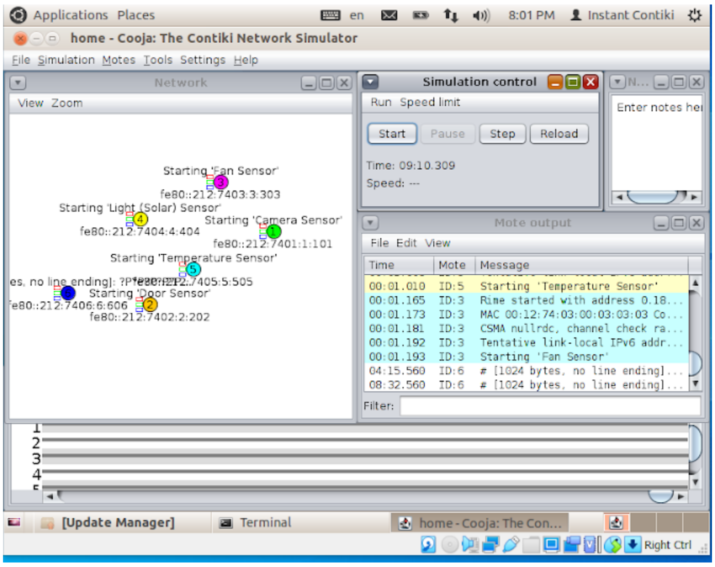
1. Setting up a home automation system in IoT
2. Setting up a private ethereum blockchain
3. Building a smart contract
4. Deploying the smart contract using web3 on the private blockchain


Smart Home
Home automation system
Validation of sender nodes via smart contracts on ethereum blockchain
Malicious node detected
STOP
stop data transfer
everything safe
CONTINUE
continue with the next transaction

# HOME AUTOMATION SYSTEM

A home automation system containing 5 sender sensor nodes and 1 receiver router node was set up. This set up was run on cooja simulator on the contiki OS

**Contiki** is an operating system used for network based and memory constrained systems which focus on low powered wireless IoT devices. Contiki can be used to develop street lighting systems, smart homes, alarms and monitoring systems etc. It is a multitasking OS with a built in IP protocol. **Cooja** is a Contiki OS network simulator. It allows large and small networks of IoT. The project uses cooja to simulate 5 sensor nodes and 1 router node as sky motes.[6]



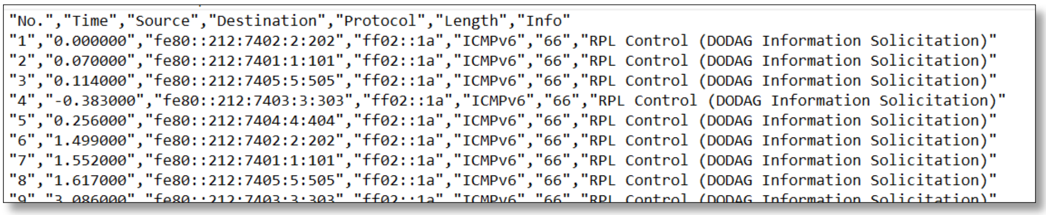
```
Time       Mote  Message
00:01.010  ID:5  Starting 'Temperature Sensor'
00:01.165  ID:3  Rime started with address 0.18...
00:01.173  ID:3  MAC 00:12:74:03:00:03:03:03 Co...
00:01.181  ID:3  CSMA nullrdc, channel check ra...
00:01.192  ID:3  Tentative link-local IPv6 addr...
00:01.193  ID:3  Starting 'Fan Sensor'
04:15.560  ID:6  # [1024 bytes, no line ending]...
08:32.560  ID:6  # [1024 bytes, no line ending]...
```

The home automation system contains:

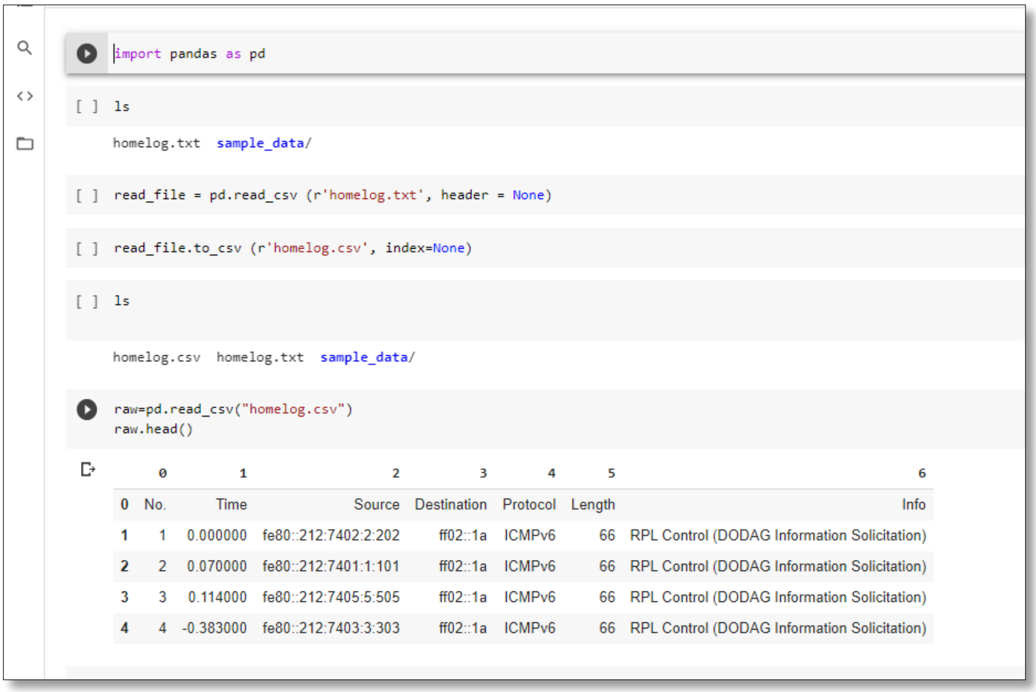
- Camera sensor
- Door sensor
- solar sensor
- fan sensor
- Temperature sensor[7]

All these sensor are sender nodes.

When a simulation is run on cooja using sender and receiver nodes on a 6LoWPAN network. The data transfer between the sender and receiver nodes are saved as PCAP radio log files. These files can be converted to CSV files using wireshark. Wireshark gives comma separated text data files. These files can be converted to a usable CSV format using the CSV library of python. The next step involved converting the source IP addresses to an integer. This was done just for the sake of this project. The type of the IP address(string/integer) will directly change the white list creation process.



```
"No.","Time","Source","Destination","Protocol","Length","Info"
"1","0.000000","fe80::212:7402:2:202","ff02::1a","ICMPv6","66","RPL Control (DODAG Information Solicitation)"
"2","0.070000","fe80::212:7401:1:101","ff02::1a","ICMPv6","66","RPL Control (DODAG Information Solicitation)"
"3","0.114000","fe80::212:7405:5:505","ff02::1a","ICMPv6","66","RPL Control (DODAG Information Solicitation)"
"4","-0.383000","fe80::212:7403:3:303","ff02::1a","ICMPv6","66","RPL Control (DODAG Information Solicitation)"
"5","0.256000","fe80::212:7404:4:404","ff02::1a","ICMPv6","66","RPL Control (DODAG Information Solicitation)"
"6","1.499000","fe80::212:7402:2:202","ff02::1a","ICMPv6","66","RPL Control (DODAG Information Solicitation)"
"7","1.552000","fe80::212:7401:1:101","ff02::1a","ICMPv6","66","RPL Control (DODAG Information Solicitation)"
"8","1.617000","fe80::212:7405:5:505","ff02::1a","ICMPv6","66","RPL Control (DODAG Information Solicitation)"
"9","3.086000","fe80::212:7403:3:303","ff02::1a","ICMPv6","66","RPL Control (DODAG Information Solicitation)"
```



```
import pandas as pd
```

```
[ ] ls
```

```
homelog.txt   sample_data/
```

```
[ ] read_file = pd.read_csv (r'homelog.txt', header = None)
```

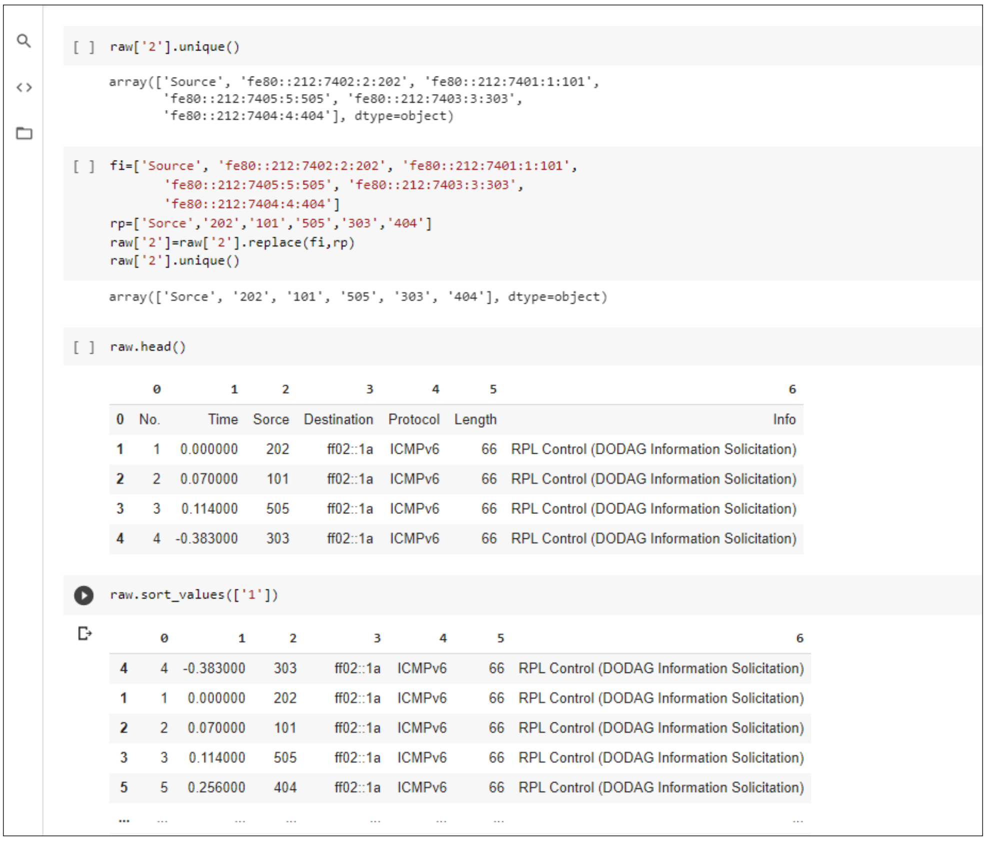
```
[ ] read_file.to_csv (r'homelog.csv', index=None)
```

```
[ ] ls
```

```
homelog.csv   homelog.txt   sample_data/
```

```
raw=pd.read_csv("homelog.csv")
raw.head()
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 0 | No. | Time | Source | Destination | Protocol | Length | Info |
| 1 | 1 | 0.000000 | fe80::212:7402:2:202 | ff02::1a | ICMPv6 | 66 | RPL Control (DODAG Information Solicitation) |
| 2 | 2 | 0.070000 | fe80::212:7401:1:101 | ff02::1a | ICMPv6 | 66 | RPL Control (DODAG Information Solicitation) |
| 3 | 3 | 0.114000 | fe80::212:7405:5:505 | ff02::1a | ICMPv6 | 66 | RPL Control (DODAG Information Solicitation) |
| 4 | 4 | -0.383000 | fe80::212:7403:3:303 | ff02::1a | ICMPv6 | 66 | RPL Control (DODAG Information Solicitation) |



```
[ ] raw['2'].unique()
```

```
array(['Source', 'fe80::212:7402:2:202', 'fe80::212:7401:1:101',
       'fe80::212:7405:5:505', 'fe80::212:7403:3:303',
       'fe80::212:7404:4:404'], dtype=object)
```

```
[ ] fi=['Source', 'fe80::212:7402:2:202', 'fe80::212:7401:1:101',
       'fe80::212:7405:5:505', 'fe80::212:7403:3:303',
       'fe80::212:7404:4:404']
    rp=['Sorce','202','101','505','303','404']
    raw['2']=raw['2'].replace(fi,rp)
    raw['2'].unique()
```

```
array(['Sorce', '202', '101', '505', '303', '404'], dtype=object)
```

```
[ ] raw.head()
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 0 | No. | Time | Sorce | Destination | Protocol | Length | Info |
| 1 | 1 | 0.000000 | 202 | ff02::1a | ICMPv6 | 66 | RPL Control (DODAG Information Solicitation) |
| 2 | 2 | 0.070000 | 101 | ff02::1a | ICMPv6 | 66 | RPL Control (DODAG Information Solicitation) |
| 3 | 3 | 0.114000 | 505 | ff02::1a | ICMPv6 | 66 | RPL Control (DODAG Information Solicitation) |
| 4 | 4 | -0.383000 | 303 | ff02::1a | ICMPv6 | 66 | RPL Control (DODAG Information Solicitation) |

```
raw.sort_values(['1'])
```

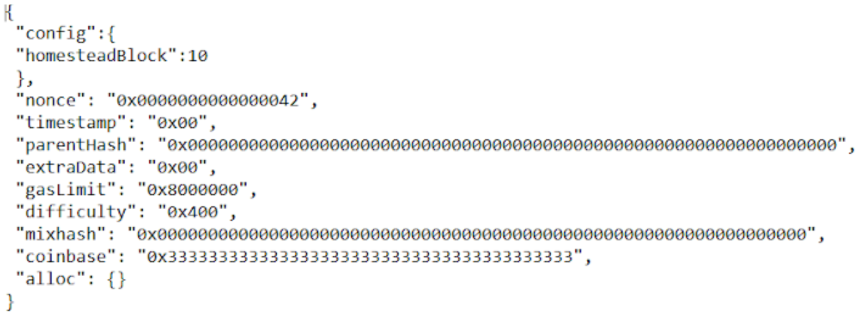
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 4 | 4 | -0.383000 | 303 | ff02::1a | ICMPv6 | 66 | RPL Control (DODAG Information Solicitation) |
| 1 | 1 | 0.000000 | 202 | ff02::1a | ICMPv6 | 66 | RPL Control (DODAG Information Solicitation) |
| 2 | 2 | 0.070000 | 101 | ff02::1a | ICMPv6 | 66 | RPL Control (DODAG Information Solicitation) |
| 3 | 3 | 0.114000 | 505 | ff02::1a | ICMPv6 | 66 | RPL Control (DODAG Information Solicitation) |
| 5 | 5 | 0.256000 | 404 | ff02::1a | ICMPv6 | 66 | RPL Control (DODAG Information Solicitation) |
| ... | ... | ... | ... | ... | ... | ... | ... |

# SETTING UP A PRIVATE ETHEREUM

Private ethereum Network is a private blockchain network isolated from the main network. These private blockchains are usually created to restrict read permissions of the blockchain. The nodes with the correct permissions are used to access the blockchain. These nodes are however, not connected to the main network. Private blockchains are used by organizations which need to keep the data secure within the organization. It is also used for testing purpose. As, we were also deploying the contract for testing purpose, we decided to use private blockchain for the same.

For the sake of implementation, geth( go ethereum) was used. A genesis file was created which is initialized as the first block of the blockchain with a nonce value, coinbase, gas limit, difficulty etc. The gas limit is set high and the difficulty is set low. After initializing the genesis file, the node1 and node 2 are initialized. Accounts are created. Both these nodes are connected by adding peers. Balance is added into these accounts by starting mining.

This process sets up a private blockchain on the system. We can now interact with this blockchain using command line arguments or web3 library of blockchain.



```
{
 "config":{
 "homesteadBlock":10
 },
 "nonce": "0x0000000000000042",
 "timestamp": "0x00",
 "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
 "extraData": "0x00",
 "gasLimit": "0x8000000",
 "difficulty": "0x400",
 "mixhash": "0x0000000000000000000000000000000000000000000000000000000000000000",
 "coinbase": "0x3333333333333333333333333333333333333333",
 "alloc": {}
}
```

# BUILDING A SMART CONTRACT

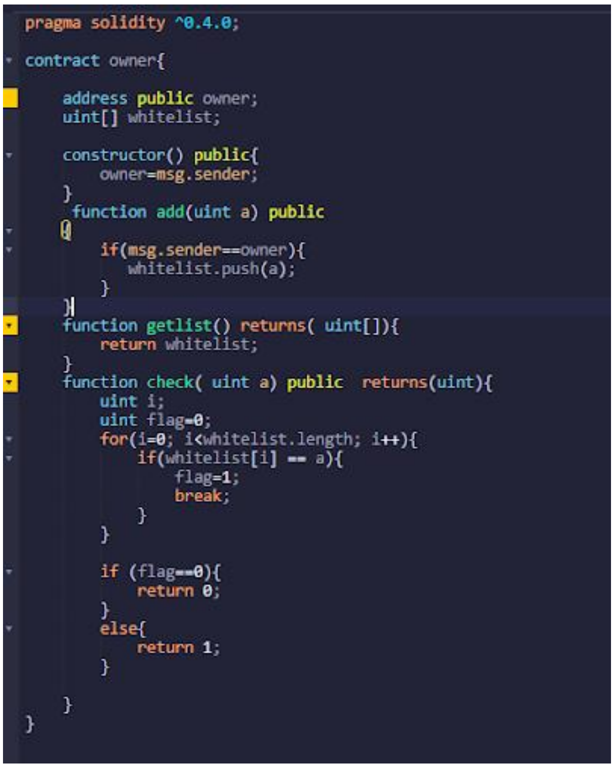
## Why do we need a smart contract?

A smart contract is a small piece of code that enables the client and owner to implement some necessary tasks on blockchain. As a part of the project, we need to check if the source node in Iot is a valid IP address or not. If the IP address is a valid IP address, we continue with the data transfer, otherwise we go ahead and delete the source node.

## How do we implement the smart contract?

The smart contract  is implemented in solidity using remix ide. The bytecode and abi of the compiled code are used when we deploy the contract with web3 library.

## Code of the smart contract



```
pragma solidity ^0.4.0;

contract owner{

    address public owner;
    uint[] whitelist;

    constructor() public{
        owner=msg.sender;
    }
     function add(uint a) public
    {
        if(msg.sender==owner){
           whitelist.push(a);
        }
    }
    function getlist() returns( uint[]){
        return whitelist;
    }
    function check( uint a) public  returns(uint){
        uint i;
        uint flag=0;
        for(i=0; i<whitelist.length; i++){
            if(whitelist[i] == a){
                flag=1;
                break;
            }
        }

        if (flag==0){
            return 0;
        }
        else{
            return 1;
        }

    }
}
```

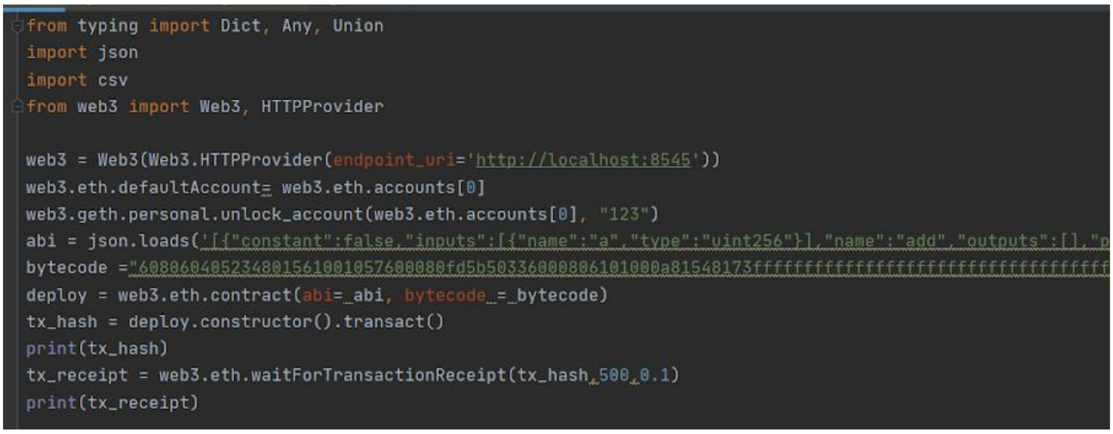
## The functions

- Contract variables: there are two contract variables. Public address owner and Integer array whitelist.
- Constructor: The constructor is used to set the deployer address in owner.
- Add: the add function is used to add an IP address to the whitelist, only if the sender is the owner.
- Check: this function returns 1 if the passed IP address is present in the list and 0 if the passes IP address is not there in the list.
- Getlist : this function returns the whitelist at that current moment.

# DEPLOYING THE SMART CONTRACT

The smart contract is deployed on the private blockchain using a python script and web3 library.

## Part 1:



```
from typing import Dict, Any, Union
import json
import csv
from web3 import Web3, HTTPProvider

web3 = Web3(Web3.HTTPProvider(endpoint_uri='http://localhost:8545'))
web3.eth.defaultAccount= web3.eth.accounts[0]
web3.geth.personal.unlock_account(web3.eth.accounts[0], "123")
abi = json.loads('[{"constant":false,"inputs":[{"name":"a","type":"uint256"}],"name":"add","outputs":[],"p
bytecode ="608060405234801561001057600080fd5b50336000806101000a81548173fffffffffffffffffffffffffffffffff
deploy = web3.eth.contract(abi= abi, bytecode = bytecode)
tx_hash = deploy.constructor().transact()
print(tx_hash)
tx_receipt = web3.eth.waitForTransactionReceipt(tx_hash,500,0.1)
print(tx_receipt)
```

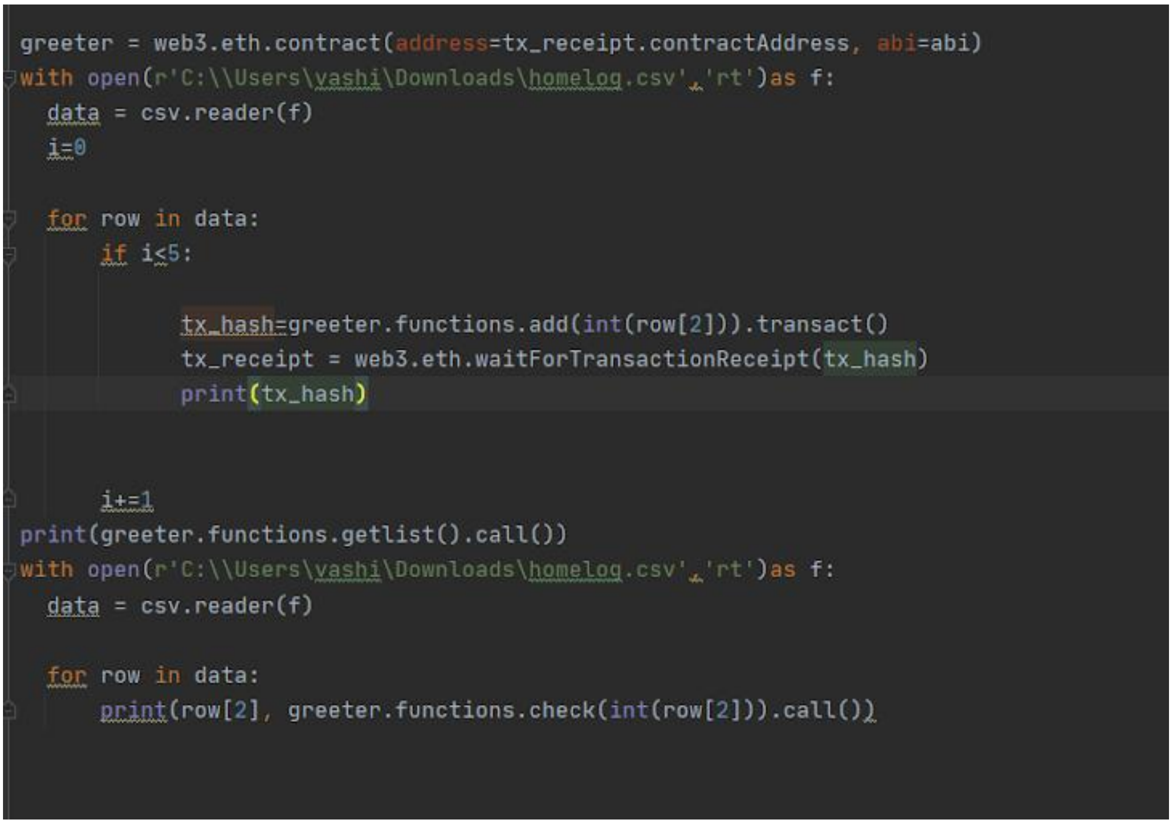
The above code uses the HTTPProvider to connect to the blockchain running on the local host port 8545. It then defines the default account and unlocks it. The abi and bytecode generated after compiling the solidity code are used to deploy the contract on the transaction using the transact method call on the constructor.

## Part 2:

The second part of the code reads the CSV file generated through home automation simulation and applies some contract functions on the same.

The add function is used to add the valid IP addresses in a whitelist. This method is called using transact() as it is meant to make changes to the blockchain and add nodes to it.

The check function can be called by the owner and client both. This function is used to detect malicious nodes. For the sake of this project, we have just printed the return value.



```
greeter = web3.eth.contract(address=tx_receipt.contractAddress, abi=abi)
with open(r'C:\\Users\vashi\Downloads\homelog.csv','rt')as f:
  data = csv.reader(f)
  i=0

  for row in data:
      if i<5:

            tx_hash=greeter.functions.add(int(row[2])).transact()
            tx_receipt = web3.eth.waitForTransactionReceipt(tx_hash)
            print(tx_hash)


      i+=1
print(greeter.functions.getlist().call())
with open(r'C:\\Users\vashi\Downloads\homelog.csv','rt')as f:
  data = csv.reader(f)

  for row in data:
      print(row[2], greeter.functions.check(int(row[2])).call())
```

# Results

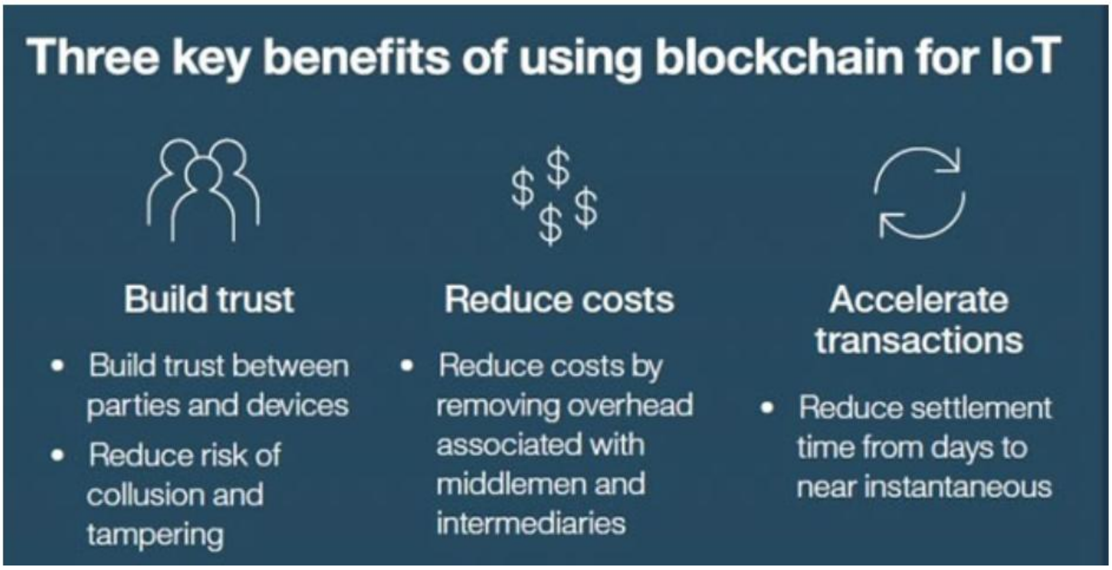
- The home automation system was set up successfully
- The blockchain was set up on local host
- The smart contract was compiled successfully
- The whitelist creation process was carried on with the csv files generated from the home automation system.
- The check function successfully returned 0 on detecting a malicious node.

# Future prospects

As the use of IoT in our lives is continuously increasing, handling important data through the current IoT architecture can be risky. The current IoT architecture allows transaction between any two objects on the basis of their IP address. Our scheme uses blockchain to secure this data transfer by authenticating the devices using Blockchain.This project is a very small demonstration of the real capabilities that it can explore. The project can be deployed on a real IoT platform and The main etherum network. For doing this, we need to take dynamic CSV files from the IoT network. The smart contract can be modified according to access rights requirement of the model. The model can be set up on a private blockchain of an organisation or even the main ethereum network for public use. This project will help in eliminating attacks on any IoT system. This is a decentralized mechanism and it is very easy to deploy.

# CONCLUSION

There is a constant increase in the demand and expansion of ideas where IoT can be used for a faster and better way of transmission. As the system becomes more dense, the data involved increases exponentially and hence, the security attacks and privacy issue become a danger to the system. To solve a basic problem of access control in IoT devices, Blockchain can be used. Blockchain is a distributed peer-to-peer system. Breaking into a blockchain and changing values without being caught is next to impossible. The report presents a detailed explanation of how Blockchain can be integrated with IoT to achieve a very secure network. Access control is the control that the IoT system has on granting access to devices for data transmission between two devices. This can be controlled using blockchain in many ways. Blockchain can be handled by the devices, or ICN architecture, or defined polices etc. This is a novel method which is being developed currently. New methods are being developed and being researched for better security results.



Three key benefits of using blockchain for IoT

**Build trust**

- Build trust between parties and devices
- Reduce risk of collusion and tampering

**Reduce costs**

- Reduce costs by removing overhead associated with middlemen and intermediaries

**Accelerate transactions**

- Reduce settlement time from days to near instantaneous

# REFERENCES

[1] Din, I. U., Guizani, M., Hassan, S., Kim, B. S., Khan, M. K., Atiquzzaman, M., & Ahmed, S. H. (2018). The Internet of Things: A review of enabled technologies and future challenges. *Ieee Access*, *7*, 7606-7640.

[2] Sfar, A. R., Natalizio, E., Challal, Y., & Chtourou, Z. (2018). A roadmap for security challenges in the Internet of Things. *Digital Communications and Networks*, *4*(2), 118-137.

[3] Laurent, M., Kaaniche, N., Le, C., & Vander Plaetse, M. (2018, July). A blockchain-based access control scheme.

[4] Tan, X., Huang, C., & Ji, L. (2018, June). Access Control Scheme Based on Combination of Blockchain and XOR-Coding for ICN. In *2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)* (pp. 160-165). IEEE.

[5] Maesa, D. D. F., Mori, P., & Ricci, L. (2017, June). Blockchain based access control. In *IFIP international conference on distributed applications and interoperable systems* (pp. 206-220). Springer, Cham.

[6] https://en.wikipedia.org/wiki/Contiki

[7] https://github.com/kanika2296/home-automation-contiki