Blockchain and IoT for the Creation of a Private Ethereum Network to Control Sensors in Smart Home Environments

Simona Franci[1]

[1]Leonardo Company Spa

Abstract

In recent years, technological innovation has allowed a rapid development of Internet services used as the basis for numerous applications in the professional and personal fields. The development of home automation and the devices associated with it has brought about great changes in terms of efficiency, simplicity and energy saving: with a simple click today it is possible to turn the light, heating, television and more on or off. The purpose of this paper is to provide performance evaluations, at a simulation and experimental level, of blockchain and IoT technologies and their interactions, creating a truly decentralized system that can allow the control of temperature and air conditioning inside a home with the use of a smart contract.

Keywords

Blockchain, Smart Contract, Internet Of Things,

1. Introduction

In recent years, technological innovation has allowed a rapid development of Internet services used as the basis for numerous applications in the professional and personal fields. For the individual, the possibility of managing a set of everyday devices without manual intervention, but through systems connected to the internet is fundamental [1, 2, 3, 4, 5]. The development of home automation and the devices associated with it has brought about great changes in terms of efficiency, simplicity and energy saving: with a simple click today it is possible to turn the light, heating, television, and more [6, 7, 8]. The evolved electronic devices are also crucial for this purposes [9, 10, 11]. Furthermore, artificial intelligence can be effectively used for energy efficiency optimization [12, 13, 14].

The aim of this paper is to provide performance evaluations, at a simulation and experimental level, of blockchain and IoT technologies and their interactions, creating a truly decentralized system that can allow the control of temperature and of air conditioning inside a home with the use of a smart contract. The blockchain, a decentralized and secure system that allows the exchange of transactions, was initially created for use in the economic sector - the exchange of Bitcoin - and used in a second phase with Ethereum [15]. The blockchain is the protagonist of an exponential development which over the last few years has led to the creation of various public or private networks for the management of countless projects [16, 17, 18]. The fulcrum of this new technology is that it is highly innovative in terms of security, transparency and immutability. IoT services represent a set of technologies that allow any type of device - sensors, mobile phones and more - to be connected and controlled via the Internet [19, 20], also including automotive vehicles. In the digital age, the development of home automation has allowed the simplification of numerous individual activities, the improvement of the design, management and maintenance of electronic devices, significantly increasing daily safety [21, 22, 23**?**]. The study then turns its attention to the technologies necessary for the creation of a private Ethereum blockchain in the LAN environment, showing a general framework regarding its use and functioning. The support of these technologies has allowed us to develop the work project which aims to create an "intelligent" house, controlled through the interaction between the blockchain and a smart contract capable of managing the entire air conditioning and heating system interior of "Paolo's" home. Whether it is a simple temperature detection, or a more complex management of data from the interconnection between multiple sensors within the smart home, it was necessary to implement a highly secure, private and autonomous system in order to avoid the third party access.

2. Use case and system design for temperature control

The project underlying this work falls within the context of the creation of an "intelligent" and safe home with particular reference to applications for temperature control and management of the air conditioning system. The problem can be formulated as follows. The owner of the house, Paolo, intends to create a blockchain system governed by a smart contract capable of detecting the temperature data sent by a sensor positioned on the thermostat, to send an alarm message in case the temperature

SYSYEM 2024: 10th Scholar's Yearly Symposium of Technology, Engineering and Mathematics, Rome, December 2-5, 2024

simona.franci21@gmail.com (S. Franci)

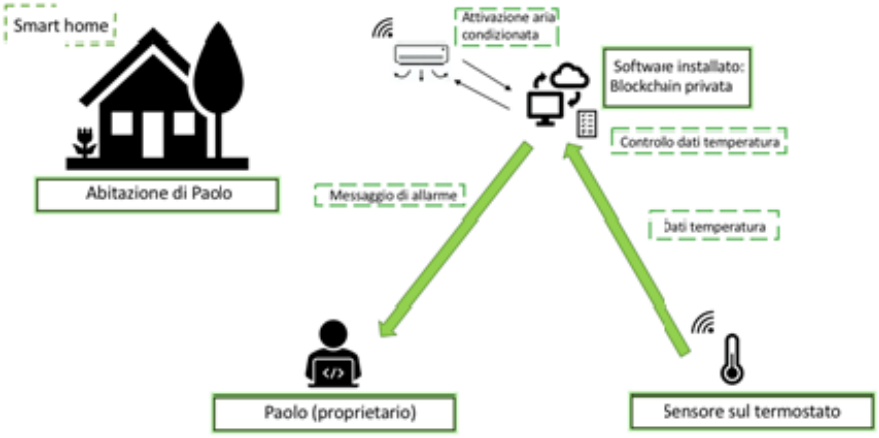

Figure 1: Reference architecture

is lower or higher than a minimum value (lower bound) and a maximum value (upper bound) respectively and to turn the air conditioning on or off. The scheme is shown in Figure 1.

3. Architectural design

In order to create the system in Figure 1, it is necessary to create a private blockchain network and define the architecture of the systems for climate and temperature control [24]. To this end, a certain number of electronic technologies and software systems were taken into consideration and integrated after establishing the interaction methods to achieve the ultimate goal of the work concerning the creation of the system in Figure 1. The proposed solution is also suitable for the automotive sector [16]. The technologies considered for the creation of each individual subsystem are listed below:

- Blockchain subsystem: Go ethereum, MetaMask, Remix;
- Sensor subsystem: thermostat and air conditioning activation. The sensors send real-time temperature data to the blockchain which is controlled through the smart contract;;
- Subsystem for sensor-end user communication: smart contract. The latter checks the temperature data sent by the sensor and sends an alarm message to the end user if the values are not within the range decided by the user himself. If the temperature value is higher than the upper bound, the smart contract passes from a control phase to an event creation phase: the air conditioning will be activated, the event will be stored within the blockchain in the form of transaction and Ether will be consumed.

The three subsystems have been integrated and exchange information through radio communication technologies in the local area according to standard protocols. For each of the subsystems listed, the implementation choices made are described in greater detail below.

3.1. Go Ethereum implementation

The software that creates the Ethereum-based blockchain runs on Go Ethereum. The latter, once installed, is used via the terminal of the device in question. Go Ethereum can be downloaded for various operating systems, and in the case of MacOs it requires an additional tool called Homebrew for managing packages that are absent in it. The intention of the work project is to create a private network made up of two different LAN nodes in which 300,000 Ether will then be allocated in a first account and 400,000 Ether in the second. The allocation of Ether to a particular account must be done within the genesis file. In order for the network to be installed and the nodes to run, it is necessary to have a solid Internet connection and that the devices are connected to the same router.

3.2. MetaMask implementation

In the implementation of the private network it is assumed that the MetaMask software is a Chrome extension, does not require any special installation software and that Paolo is equipped with a computer on which he can access his wallet and check his Ether balance. The use of MetaMask, as can easily be understood, requires an Internet connection. In the work project this particular technology is used to check the remaining balance following the various transactions that are carried out.

3.3. Implementation of Remix

Remix was used to develop and test the smart contract code. The creation of the smart contract was preceded by a series of considerations on the intention it should have and how it would be possible to transform an idea into a practical act. The first phase of contract analysis involved the study of writing a smart contract and the creation of a guide for it (see Appendix - writing a smart contract). The second phase concerned the creation and implementation of the same. The objective of the work project was to generate a contract that was able to control the temperature values sent by the thermostat sensor in real time. The expectation of the smart contract is the following: the data is read at any time from the contract which is activated only when a value is lower or higher than a particular range. It was established that the value of the upper bound was 25 degrees and the lower bound was 19 degrees. A temperature below 19 degrees leads to the creation of an event: an alarm message is sent to the devices of the private Ethereum network and Paolo is notified. The event is stored. Similarly, a temperature above 25 degrees leads to the generation of an air conditioning activation event. In this case the smart contract interacts with both sensors: with the thermostat sensor for detecting the temperature and with the air conditioner sensor for its activation. An event is created which is also stored within the network.

4. System implementation

4.1. System configuration

The components used for the practical implementation of the system illustrated in the previous chapter 3 are illustrated below. To create the system described in the use case (Figure 1), the following components were used:

- The “Simona” computer with MacOs High Sierra 10.13 operating system, IP 2.239.82.138 and port 30303;
- The “Paolo” computer with MacOs High Sierra 10.13 operating system and IP 2.239.82.138 and port 30303;
- A sensor positioned on the thermostat;
- A sensor positioned on the air conditioning.

4.2. Local blockchain network creation

The preliminary steps to the reaction of the Ethereum LAN network were:

- Creation of two folders, one on each device involved in the system configuration. The folder served as a reference path throughout the work project;
- Download of “go Ethereum for Mac” from the official website;
- Download Homebrew, a free and open source software package management system that simplifies the installation of a tool on the Apple MacOS operating system. For the Ethereum Homebrew network it is necessary to create the “bootstrap” node.

4.3. Creating a new account

The creation of a new account occurred through the command “geth account new <datadir>”. A password is required to generate a public/private key pair kept within the “keystore”. Once the password has been entered and stored, the account will be created. This operation was carried out on both computers set up for the creation of the Ethereum network in LAN in order to obtain two different accounts on the device renamed “Simona” and an account on “Paolo” (Figure 3).

A fundamental step in creating a private network is defining your own genesis block. The file, in .json format, must be saved in the datadir in each peer on the network. Contains the following information: - ChainID: provides the value of the ID that characterizes the private network. In this project ID 15 was used; - Homesteadblock: defines the first Ethereum network release; - Eip150Block/ Eip155Block/ Eip158Block: EIP media 150, 155, and 158 describe the standards, core protocol specifications, and client APIs for the Ethereum platform; - ByzantiumBlock/ConstantinopleBlock/petersburgBlock: define releases after Homestead; - Ethash: represents the consensus algorithm; - Difficulty: defines the mining difficulty; - GasLimit: establishes the maximum value of gas that can be used in transactions; - Alloc: allows you to pre-allocate a certain number of Ether to one or more accounts. Two different accounts were inserted in the genesis block in question, one belonging to “Simona” and one to “Paolo” (Figure 3).

Once the file was created, it was possible to initialize the private network through the command “geth —datadir <path> init <path-genesis-file>”.

4.4. Network setup and mining procedure

Once the node was configured to the desired genesis state, the next step was to configure the peer-to-peer network. It was appropriate to use a single node - the “bootstrap” node - as a meeting point for the other peers. The latter must know its IP address and enode identifier to allow connection to other users of the network. It was then necessary to verify that the firewall configuration could allow UDP and TCP traffic on port 30303. To control the open ports, telnet was downloaded using the commands “brew tap theeternalsw0rd/telnet brew install telnet

curl http://ftp.gnu.org/gnu/inetutils/inetutils- 1.9.4.tar.gz -o inetutils-1.9.4.tar.gz tar xvzf inetutils-1.9.4.tar.gz cd inetutils-1.9.4 ./configure make sudo make install". Port 30303 was found to be closed. The reference port has been added to the router settings.

The enode is defined following the generation of a key using the commands "–genkey=boot.key" and "–nodekey=boot.key" (Figure 6). The IP address displayed in the identifier must be replaced with the externally accessible IP in order to obtain the URL that can be used to connect the other peers. We chose to use the account "0x6106520baB278022355948516DC2bAfA2FC56a48" on the "Simona" device as the bootnode. The "bootstrap" node is online. The mining procedure was started with the command "geth –mine –minerthreads 1 –rpc –rpcaddr 0.0.0.0 –rpcport "8545" –port "30303" –rpccorsdomain "*" –networkid 15 - -datadir <path> –allow-insecure-unlock". The information entered in this command refers to the characteristics of the private network previously illustrated. The extraction has begun.

To connect the geth console, a new shell was opened and the command "geth –datadir <path> attach ipc:<path> /geth.ipc" was entered. Once the connection was completed, it was possible to check the available accounts through "eth.accounts" and the bootnode balance with "eth.getBalance("")" Second node connection to the LAN network

To connect the node on "Paolo's" computer to the bootnode, the first step was to initialize the previously created account with the same genesis file used for "Simona". The command to use is the same as the network initialization step, with the appropriate changes to the path. Once the node was started, the "admin.nodeInfo" command was executed on "Paolo's" JavaScript console to find the identifier. The latter was then inserted into the "admin.addPeer("enode")" command on the "Simona" console. The passage returned the value "true", confirming the connection of the two nodes to each other.

4.5. Network connection with MetaMask

The MetaMask connection to the Ethereum private network was made with the node running. The first step was taken by creating a custom network in the Chrome extension, in particular by specifying the host name and port used. At a later stage, the network accounts were entered using the private key, or by uploading the genesis file and the password relating to the account itself. At this point, the MetaMask wallet is ready to be used: Paolo can then manage his identity and examine the account balance and transactions before approving or rejecting them.

4.6. Creation of smart contracts for the use case

The smart contract was created to allow the recovery of the temperature data sent by the sensor placed on the thermostat. If the home temperature value is lower than the lower bound defined by Paolo then the contract will issue an alarm message to the homeowner. If, otherwise, the value of the room is higher than the value of the upper bound, an alarm message is sent to Paolo and the air conditioning of the room in question is activated:

- the constructor function is executed first within the contract and is unique;
- "onlyOwner" is the name of the modifier, i.e. the one who changes the behavior of a function. In the case in question it is Paolo;
- require declares prerequisites for the function and checks that the conditions are verified before starting the subsequent lines of code;
- "msg.sender" defines the address of the account that invoked the function;
- set I only allow Paolo to set the value of the function;
- it also guarantees that the blockchain is unchangeable;
- get allows you to read the data set by the owner from the other nodes in the network.

4.7. Smart contract validation

In order to validate the deployment of the contract, the terminal sends a "warning" message due to the blocking of the account on which the transaction is being carried out. Before the smart contract can be validated, the bootnode account must be unlocked with the "web3.personal.unlockAccount("", "password", duration) command. The command must return "true". The contract has been deployed: a new block in the private network has been created. The validation of the transaction cost a quantity of gas paid by the account connected to the Remix network.

4.8. Evaluation of the case study

The functioning of the implemented system was finally tested and validated. The following figure shows an event corresponding to the expenditure of Ether following the deployment of the contract and therefore the creation of a new block. If the sensor detects a temperature of 26 degrees, i.e. a temperature higher than the maximum limit set by Paolo, the smart contract creates an event and subsequently a transaction which is stored within the blockchain. We have the correct functioning of the smart contract and of the system created in this work work.

5. Conclusions

The interaction between blockchain and IoT today represents an emerging technology given the vastness of the application fields to which it lends itself. From the tests conducted during the writing of this work it emerged that, in the smart home context, the use of a decentralized system capable of controlling the sensors positioned on electronic devices, such as the thermostat and the air conditioner, finds positive results in temperature management in a house. In the project it was fundamental to define a smart contract for the real-time control of temperature values: an alteration above or below a certain limit value set and decided by the owner of the "smart" house, determines the activation or deactivation of the air conditioning system without any manual intervention. The simplification and monitoring of connected devices is therefore evident. If from an application point of view the creation of the private Ethereum network in the LAN and the system connected to it has produced positive results, the reduction of energy costs and a better management of Paolo's time and quality of private life, all that remains is set the physical connection of sensors as a future objective by setting up a Raspberry Pi3 as an interlocutor with the blockchain.

References

[1] R. Modi, Introduction to blockchain, ethereum and smart chapter 1. retrieved from coinmonks: https://medium. com/coinmonks/https-medium-com-riteshmodi-solidity-chapter1-63dfaff08a11, 2018.

[2] F. Fiani, S. Russo, C. Napoli, An advanced solution based on machine learning for remote emdr therapy, Technologies 11 (2023). doi:10.3390/technologies11060172.

[3] R. Garavaglia, Tutto su blockchain: capire la tecnologia e le nuove opportunità (2018).

[4] G. De Magistris, S. Russo, P. Roma, J. T. Starczewski, C. Napoli, An explainable fake news detector based on named entity recognition and stance classification applied to covid-19, Information (Switzerland) 13 (2022). doi:10.3390/info13030137.

[5] C. Napoli, G. Pappalardo, E. Tramontana, Improving files availability for bittorrent using a diffusion model, in: Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE, 2014, p. 191 – 196. doi:10.1109/WETICE.2014.65.

[6] R. Giuliano, F. Mazzenga, A. Vizzarri, Integration of broadcaster and telco access networks for real time/live events, IEEE Transactions on Broadcasting 66 (2020) 667–675.

[7] I. E. Tibermacine, A. Tibermacine, W. Guettala, C. Napoli, S. Russo, Enhancing sentiment analysis on seed-iv dataset with vision transformers: A comparative study, in: ACM International Conference Proceeding Series, 2023, p. 238 – 246. doi:10.1145/3638985.3639024.

[8] R. Giuliano, F. Mazzenga, E. Innocenti, A. Vizzarri, Integration of video and radio technologies for social distancing, IEEE Communications Magazine 59 (2021) 30–35.

[9] G. C. Cardarilli, G. M. Khanal, L. Di Nunzio, M. Re, R. Fazzolari, R. Kumar, Memristive and memory impedance behavior in a photo-annealed zno–rgo thin-film device, Electronics 9 (2020) 287.

[10] F. Bonanno, G. Capizzi, G. L. Sciuto, C. Napoli, Wavelet recurrent neural network with semi-parametric input data preprocessing for micro-wind power forecasting in integrated generation systems, in: 5th International Conference on Clean Electrical Power: Renewable Energy Resources Impact, ICCEP 2015, 2015, p. 602 – 609. doi:10.1109/ICCEP.2015.7177554.

[11] S. Acciarito, A. Cristini, L. Di Nunzio, G. M. Khanal, G. Susi, An a vlsi driving circuit for memristor-based stdp, in: 2016 12th Conference on Ph. D. Research in Microelectronics and Electronics (PRIME), IEEE, 2016, pp. 1–4.

[12] G. Lo Sciuto, G. Capizzi, S. Coco, R. Shikler, Geometric shape optimization of organic solar cells for efficiency enhancement by neural networks, in: Advances on Mechanics, Design Engineering and Manufacturing: Proceedings of the International Joint Conference on Mechanics, Design Engineering & Advanced Manufacturing (JCM 2016), 14-16 September, 2016, Catania, Italy, Springer, 2017, pp. 789–796.

[13] G. Lo Sciuto, G. Capizzi, R. Shikler, C. Napoli, Organic solar cells defects classification by using a new feature extraction algorithm and an ebnn with an innovative pruning algorithm, International Journal of Intelligent Systems 36 (2021) 2443–2464.

[14] G. Lo Sciuto, G. Susi, G. Cammarata, G. Capizzi, A spiking neural network-based model for anaerobic digestion process, in: 2016 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), IEEE, 2016, pp. 996–1003.

[15] G. Wood, et al., Ethereum: A secure decentralised generalised transaction ledger, Ethereum project yellow paper 151 (2014) 1–32.

[16] L. Cotugno, F. Mazzenga, A. Vizzarri, R. Giuliano, The major opportunities of blockchain for automotive industry: a review, in: 2021 AEIT International Conference on Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE), IEEE,

2021, pp. 1–6.
[17] N. Boutarfaia, S. Russo, A. Tibermacine, I. E. Tibermacine, Deep learning for eeg-based motor imagery classification: Towards enhanced human-machine interaction and assistive robotics, in: CEUR Workshop Proceedings, volume 3695, 2023, p. 68 – 74.
[18] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of things (iot): A vision, architectural elements, and future directions, Future generation computer systems 29 (2013) 1645–1660.
[19] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications, IEEE communications surveys & tutorials 17 (2015) 2347–2376.
[20] G. Capizzi, F. Bonanno, C. Napoli, Hybrid neural networks architectures for soc and voltage prediction of new generation batteries storage, in: 3rd International Conference on Clean Electrical Power: Renewable Energy Resources Impact, ICCEP 2011, 2011, p. 341 – 344. doi:10.1109/ICCEP.2011.6036301.
[21] A. Reyna, C. Martín, J. Chen, E. Soler, M. Díaz, On blockchain and its integration with iot. challenges and opportunities, Future generation computer systems 88 (2018) 173–190.
[22] C. Napoli, F. Bonanno, G. Capizzi, An hybrid neuro-wavelet approach for long-term prediction of solar wind, in: Proceedings of the International Astronomical Union, volume 6, 2010, p. 153 – 155. doi:10.1017/S174392131100679X.
[23] G. Capizzi, C. Napoli, L. Paternò, An innovative hybrid neuro-wavelet method for reconstruction of missing data in astronomical photometric surveys, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 7267 LNAI, 2012, p. 21 – 29. doi:10.1007/978-3-642-29347-4_3.
[24] C. Dannen, Introducing Ethereum and solidity, volume 1, Springer, 2017.