# A Tutorial on the Implementation of a Hyperledger Fabric-based Security Architecture for IoMT

Filippos Pelekoudas-Oikonomou
*Evotel Informática S.A.,* Spain
*Faculty of Engineering and Science,*
*University of Greenwich*
Chatham Maritime, UK
filippos@evotel-info.com

Jose Ribeiro
*Evotel Informática S.A.,* Spain
jose@evotel-info.com

Georgios Mantas
*Faculty of Engineering and Science,*
*University of Greenwich*
Chatham Maritime, UK
g.mantas@greenwich.ac.uk

Firooz Bashashi
*Evotel Informática S.A.,* Spain
firooz@evotel-info.com

Georgia Sakellari
*Faculty of Engineering and Science,*
*University of Greenwich*
Chatham Maritime, UK
g.sakellari@greenwich.ac.uk

Jonathan Gonzalez
*Evotel Informática S.A.,* Spain
jgonzalez@evotel-info.com

***Abstract*— Internet of Medical Things (IoMT) have improved individuals' quality of life by enabling IoMT-based healthcare monitoring systems to grow dramatically in recent years. Therefore, cutting-edge security techniques are needed to address the security risks of IoMT networks effectively and in a timely manner. On the other hand, blockchain technology has the potential to play a significant role in both securing IoMT devices and preventing unauthorized access during data transmission and it has been anticipated by the industry and the research community to be a disruptive technology that can be incorporated into novel security solutions for IoMT networks. In this regard, the goal of this research work is to demonstrate the integration of blockchain technology into novel security solutions for IoMT networks and to deploy a Hyperledger Fabric-based blockchain security architecture for IoMT-based healthcare monitoring systems by utilizing the features of the Hyperledger Fabric Platform, its utilities, and its lightweight consensus nature in order to: i) improve security in IoMT-based healthcare monitoring systems, ii) provide secure data storage in a decentralized way, and iii) eliminate single point of failure.**

**Keywords— Internet of Things, Internet of Medical Things, Hyperledger Fabric, Blockchain**

## I. INTRODUCTION

The Internet of Medical Things (IoMT), in which medical Internet of Things (IoT) devices are interconnected so that anyone may have access to them from anywhere and at any time, has been emerged in recent years thanks to IoT technology. This technology has transformed the healthcare industry and brought significant benefits to the healthcare sector [1]. The development and expansion of IoMT can significantly contribute to improving people's quality of life by enabling IoMT-based healthcare monitoring systems to provide personalized and user-centric healthcare services despite time and location constraints [2], [3]. However, the wide range of communication technologies (e.g., WLANs, Bluetooth, and Zigbee) and IoMT device types (i.e., bio-sensors, actuators) used in IoMT-based healthcare monitoring systems as well as the fact that personal and confidential healthcare information (i.e., patient's personal details and vital signs) is transmitted between patients and healthcare providers via the Internet are factors that raise many security concerns [4]–[7]. Therefore, for IoMT-based healthcare monitoring systems to be accepted and widely adopted in the upcoming years, security solutions that satisfy the fundamental security requirements (i.e., authentication, authorization/access control, data integrity, data confidentiality, and availability) are essential [8].

However, IoMT edge devices, which are fundamental elements of IoMT-based healthcare monitoring systems, are resource-constrained and cannot support the high resource requirements of complex traditional security solutions [9], [10]. Furthermore, owing to the problem of single point of failure, the centralized approach, commonly upheld by contemporary security solutions, is unsuitable for the case of IoMT networks [11], [12]. Hence, before IoMT edge networks can earn the trust of all involved parties and provide their full potential in the healthcare industry, innovative security solutions are urgently needed to overcome the compelling security concerns they face [3].

Blockchain technology has been foreseen, by industry and academia alike, to be a disruptive technology that can be integrated into novel security solutions for IoMT networks since it has the ability to play a significant role in: a) securing IoMT devices; and b) eliminating the single point of failure [17,18]. Researchers are focusing their efforts on the distributed features of IoT networks in order to discover a means to combine the topology of these networks with the distributed structure of blockchain in order to make IoT networks more secure [19]. Despite the significant benefits that blockchain technology can bring to IoMT-based health monitoring systems, in order to address their security challenges, these resource-constrained IoMT devices are unable to afford complex and energy inefficient operations (for example, the mining process in Proof of Work (PoW)) due to their limited processing power, storage capacity, and battery life [9].

Towards this direction, our research is focused on integrating blockchain technology in IoMT networks using more energy efficient blockchain frameworks such as the Hyperledger Fabric (HF) [13], [14], [15], [16]. In the present

This This research work has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876487. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Finland, Spain, Italy, Germany, Czech Republic, Belgium, Netherlands". Author José Ribeiro would like to acknowledge the REACT project, which has received funding from European Union's Horizon Europe research and innovation programme under the Marie Sklodowska-Curie grant agreement No 101069053 (REACT).

ISBN 978-3-903176-57-7© 2023 IFIP

research work, an HF-based architecture for securing IoMT-based healthcare monitoring systems is proposed in regard to the previously addressed challenges, and an HF network that corresponds to this architecture is deployed. The main motivation of this study relies on the lack of blockchain-based security architectures for IoMT networks, notably for IoMT-based healthcare monitoring systems.

Following this introduction, we organise the paper as follows. In section II, we provide a literature review of HF implementations for securing IoT networks. In section III, we briefly explain the main components and functionalities of the HF platform. In section IV, we present our proposed HF-based blockchain security architecture for IoMT-based healthcare monitoring systems. In section V, we present the deployment of a HF network that corresponds to our proposed architecture. Finally, section VI concludes the paper.

## II. Related work

In this section we present a literature review regarding HF implementations for securing IoT networks.

Ning Lu et al. in [14], describe HF-Audit, a proposed decentralized data integrity auditing solution that employs HF to construct two distinct communication channels for User-TPA(third-party auditor)-CSP (cloud service provider). The scheme publishes TPA credit information through the crediting channel and auditing information through the auditing channel. Shohei Kakei et al. in [15], propose a distributed authentication infrastructure that distributes trust points across several service providers and connects them via cross-certification. The proposed solution is presented by a cross-certification method that employs a smart contract on HF and a sophisticated Public Key Infrastructure (PKI). In [16], Siris V. et al. investigate interledger protocols for securely integrating transactions between two separate blockchains, one private or permissioned, Hyperledger Fabric, and one public, Rinkeby, to reduce execution cost and delay as compared to using a single public chain.

The authors of [17] suggest a system for sharing sensitive company data on the Industrial Internet of Things Hyperledger Fabric (IIoT). Raw data gathered by IIoT businesses are encrypted using the Advanced Encryption Standard (AES) and saved on the peer-to-peer file system Inter-Planetary File System (IPFS). Han Liu et al. in [18], propose the design and implementation of fabic-iot, a HF-based access control system for IoT. (ABAC). Fabic-iot is able to track records, provide dynamic access control management, and solve the access control problem in IoT by simplifying the sharing and storing of device resources by utilizing a distributed architecture.

The authors in [19], offer a novel strategy to enhance IIoT security by utilizing blockchain technology for access control. This paper provides a unique architecture based on smart contracts and attribute-based access control to limit the risk of man-in-the-middle attacks, device hijacking, distributed denial-of-service attacks, and permanent denial of service assaults. Authors in [20], proposes yet another access control solution for the Internet of Things that makes use of the features of HF. Authors establish the separation of people and devices with the suggested blockchain-based access control system based on Hyperledger Fabric by enforcing rules and programmatic access management in the chaincode.

Houshyar Pajooh et al. [21] propose a HF-based blockchain-based edge computing method for IoT networks. The suggested architecture enables safe communication and data sharing via channels and is intended to accommodate scalable IoT applications. Mutual authentication and authorization procedures assure the security of the networked devices. The authors of [22] propose and implement a secure data protection approach, Hyperledger Fabric lightweight group management (H-LGM), to safeguard critical data from unauthorized access, as well as lightweight rekeying to extend the lifetime of a network comprised of resource-constrained IoT devices.

## III. An Overview of Hyperledger Fabric

### A. Hyperledger Fabric

HF is an open-source, business-oriented, permissioned, and built for use in enterprise contexts distributed ledger technology (DLT) platform proposed initially by E. Androulaki et al. [13] and established under Linux Foundation [23]. HF has a highly flexible and extensible design that allows for innovation, versatility, and optimization across a wide range of industry use cases. The HF platform is permissioned, which implies that the participants know each other rather than being anonymous, as in the case of a public blockchain, which does not require permission to access its resources and is thus completely untrusted. This means that even if the participants do not entirely trust one another (for example, competitors in the same market or business), an HF blockchain can nevertheless be operated by a governance model (e.g., transaction endorsement policies or chaincode) based on existing trust between participants. One of the HF platform's most notable functionalities is its support for pluggable consensus protocols. This capability allows the platform to be more successfully tailored to individual use cases and trust models. When deployed within a single organization or governed by a trusted authority, fully byzantine fault-tolerant consensus, for example, may be redundant and an unnecessary load on performance and throughput. In such cases, a crash fault-tolerant (CFT) consensus protocol may be sufficient; however, in a multi-party, decentralized use case, a byzantine fault-tolerant (BFT) consensus protocol may be required [24]. For all of these reasons, Hyperledger Fabric could be a feasible blockchain platform on which to build lightweight blockchain-based security solutions for IoMT networks [9], [10].

### B. Hyperledger Fabric Components

In this section we present the main components and functionalities of HF as well as the reasons why it constitutes a suitable platform for deploying blockchain-based security architectures for IoMT-based healthcare monitoring systems.

*Blockchain Network:* It is a P2P network where nodes share a distributed ledger and follow a consensus protocol. The HF network includes chaincodes for generating transactions which are recorded immutably on each peer node's copy of the ledger.

*Peer:* It is an essential component of the blockchain network that hosts the distributed ledger and chaincode. Peers can also offer SDK and APIs for user interaction. Two types of peers exist: anchor peers and endorsement peers. Anchor peers distribute blocks to endorsement peers who endorse invoked chaincode on behalf of clients. Chaincode specifies endorsement policies that dictate the required number of peers to execute and endorse it.

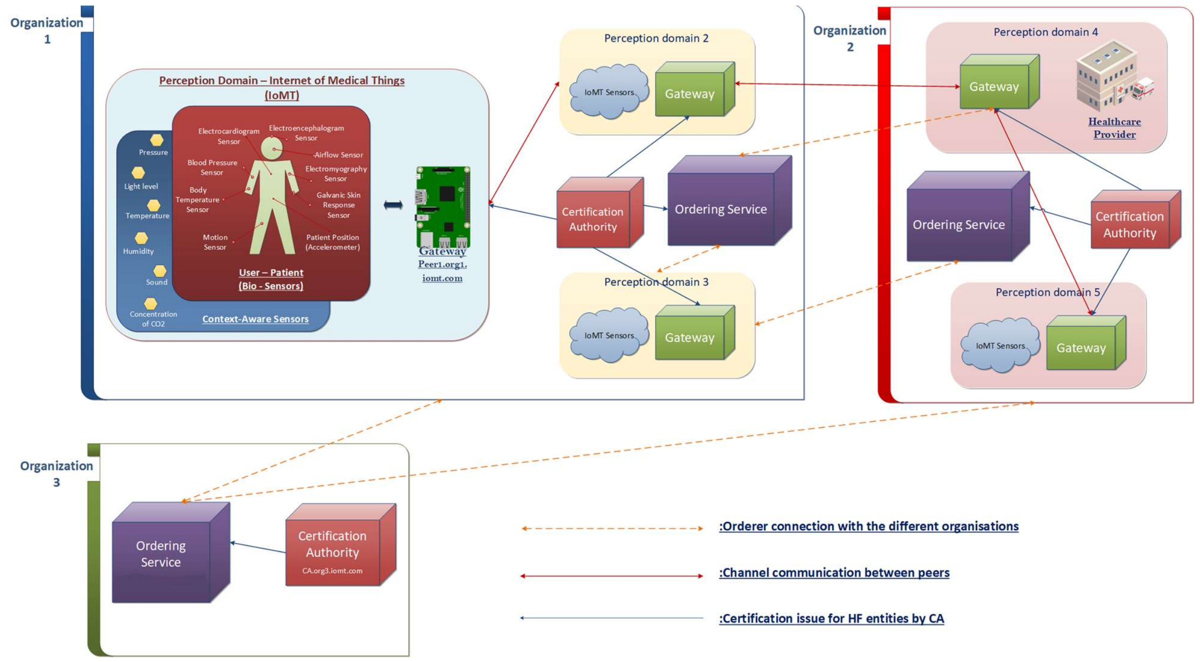


Fig. 1 A generic Hyperledger Fabric architecture for IoMT healthcare systems

*Ordering Service:* In contrast to permissionless blockchains such as Bitcoin and Ethereum, which rely on probabilistic processes to achieve consensus, HF employs an ordering node to define the order of transactions. This node is part of the ordering service, a group of nodes responsible for transaction ordering before the deterministic consensus process takes place. Ordering is distinct from endorsement, which occurs on peers. HF offers three ordering service consensus protocols: Solo, Kafka, and Raft. Solo has one ordering node and is suitable for testing but not for production. Kafka uses a leader-follower approach and Zookeeper coordination, while Raft uses the Raft protocol and is easier to set up than Kafka [25].

*Certificate Authority (CA):* CA generates X.509 certificates to identify admins, users, peers, orderers, and applications on a blockchain network [26]. These certificates also specify the network privileges of each entity.

*Chaincode:* As chaincode can be defined the code that functions as an application and offers capabilities to the blockchain network, packaged within a Docker container. While node.js is used in this implementation, other languages like Go or Java can be used to write chaincode.

*Channels:* They facilitate communication between nodes in a blockchain network, consisting of organizations, member peers, the distributed ledger, and chaincode. Transactions are proposed and processed within channels. In Hyperledger Fabric, a node can join multiple channels, allowing for private transmission of data and information.

*Endorsement policies:* They determine the minimum number of channel peers required to execute and endorse a transaction's chaincode for it to be considered valid. During transaction validation, endorsing peers confirm that the transaction has the required number of endorsements.

*Membership Service Provider (MSP):* HF abstracts membership operations, separating cryptographic processes such as certificate issuance, validation, and authentication. It enables peers to authenticate incoming transaction requests from clients and sign transaction outcomes. MSP defines its own identity and rules for identity management, while CA generates certificates for MSP operations.

All in all, HF is a scalable platform designed for permissioned blockchains, with changeable trust assumptions that can support various industrial use cases, including healthcare. Its lightweight nature and features make it a suitable choice for blockchain-based security architectures for IoMT-based healthcare monitoring systems, considering the design needs and resource constraints of IoMT nodes.

## IV. Proposed Architecture

In this current research work, we have expanded upon our previous works [9], [10] and further improved our initial design and proceed with an HF-based blockchain security architecture for IoMT-based healthcare monitoring systems. This architecture is based on the initial system model which refers to the perception domain (i.e., edge network) described in [10]. The proposed architecture is generic and can include multiple organizations, depending on the parties involved or the needs of a given scenario, and each organization can have a number of orderers and/or peers that transfer data within the blockchain network. In Fig. 1, we present an example topology consisting of three organizations.

**Organization 1 (Org1)**: contains 3 peers, each belonging to a corresponding perception domain (i.e., edge network). These peers are responsible for the transactions initiated in their own perception domain, while also responsible for holding the ledger of the channels that they are involved in.
**Organization 2 (Org2)**: contains 2 peers and a corresponding CA. As an example, we can define that one peer belongs to a healthcare provider (i.e., Peer1.org2.iomt.com) while the other to a patient (i.e., Peer1.org2.iomt.com).

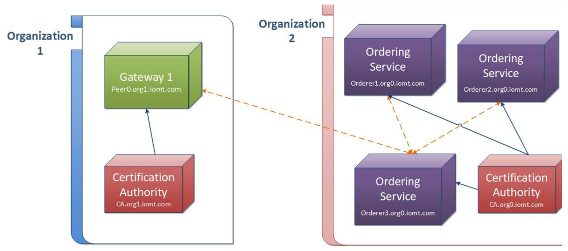


Fig. 2 The deployed HF network for the proposed scenario

**Organization 3 (Org3)**: contains only an orderer. Each organization does not have obligatory components and the number of peers/orderers can be adjusted to the needs of the organization.

## V. Implementation

In order to implement our architecture in an actual application, we have taken the initial step of constructing an HF network, as shown in Fig. 2, corresponding to the proposed HF-based blockchain security architecture for IoMT-based healthcare monitoring systems. This section details the process we undertook to create the network as a tutorial. In table I. we present the parameters of the VM in which we have developed the network.

TABLE I. Virtual Environment Parameters

| Feature | Specifications |
|---|---|
| Operating System | Ubuntu 20.04.1 AMD 64 |
| RAM | 4096 MB |
| Hyperledger Fabric | 1.4.4 |
| Docker | 18.09.1 built 4c52b90 |
| Node.js | v.10.24.0 |

The logic behind the selected network topology, shown in Fig. 2, is that there is an established organisation (i.e., Organization 2) that contains the orderers that will be used for the ordering of transactions in the network, and another organisation (i.e., Organization 1) of the network that provides just a peer (i.e. gateway). As already mentioned, it is not required for every organisation to have the exact same components in order to be part of the network. The number of nodes inside the organisation are corresponding to its needs and functionalities. In particular, the deployed HF network for the proposed scenario includes the following components:

- Certificate Authority (CA);
- a Transport Layer Security (TLS) CA for client and node certificates for secure communication;
- the organization "org1" that contains the peer node, and;
- the organization "org2" that hosts three orderers.

The process of the creation of the network is described in the following steps:

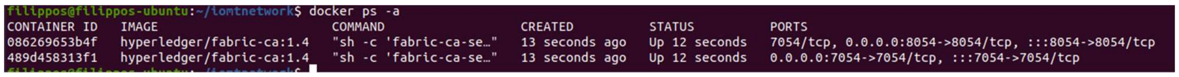
1) To set up the network, we first launch the Fabric CA servers, which are provided by the HF, in order to generate all the necessary certificates for the orderers and peers. We use two Fabric CA servers: one for the CA and one for the TLS-CA. We launch these servers using Docker containers, with the CA server running on port 7054 and the TLS-CA running on port 8054, as shown in Fig. 3.

2) After setting up the Fabric CA servers, we proceed to generate the TLS and MSP certificates. We use the official HF implementation to create a similar MSP structure for peers and orderers. According to the HF documentation [27], the following folders are essential components of the org1 and org2 MSP structure: admincerts, cacerts, signcerts, and keystore. We create these folders and generate an admin certificate for the CA server, which will be used to register the peer and orderers as shown in Fig. 4.

3) As a next step we generate the MSP and TLS certificates for the orderers, the peer, and each organization. The generation of both types of certificates is implemented by the fabric/binary provided by the HF. The process for generating TLS certificates is similar to the process for generating MSP certificates, with the main difference being that we use a TLS-CA server instead of a CA server.

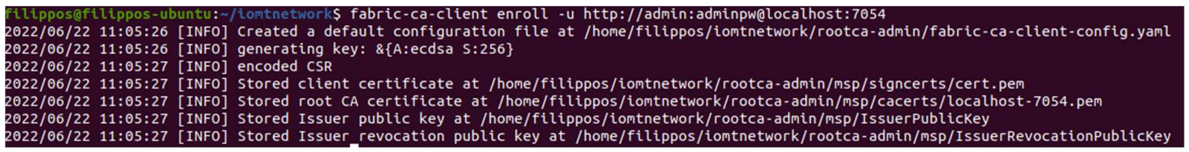
4) After generating the certificates, the next step is to initiate the orderers. To do this, we need to create a genesis block that contains information about the network, including the MSPs and organizations that will be able to join the network as show in Fig. 5. To generate the genesis block, we use the files we created for the MSPs of org1 and org2 in step 3 and follow the following sub-steps:

- we create the configtx.yaml file that contains the necessary configuration,
- we create a folder channel-artifacts to store the channel's genesis block file, and
- we generate the genesis block and a channel using the configtxgen binary.



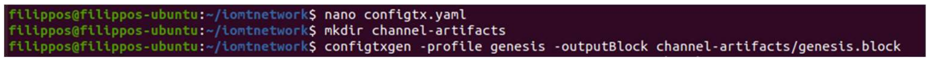
```
filippos@filippos-ubuntu:~/iomtnetwork$ docker ps -a
CONTAINER ID   IMAGE                      COMMAND                  CREATED          STATUS          PORTS
086269653b4f   hyperledger/fabric-ca:1.4   "sh -c 'fabric-ca-se…"   13 seconds ago   Up 12 seconds   7054/tcp, 0.0.0.0:8054->8054/tcp, :::8054->8054/tcp
489d458313f1   hyperledger/fabric-ca:1.4   "sh -c 'fabric-ca-se…"   13 seconds ago   Up 12 seconds   0.0.0.0:7054->7054/tcp, :::7054->7054/tcp
```

Fig. 3 The Fabric CA servers containers (step 1)



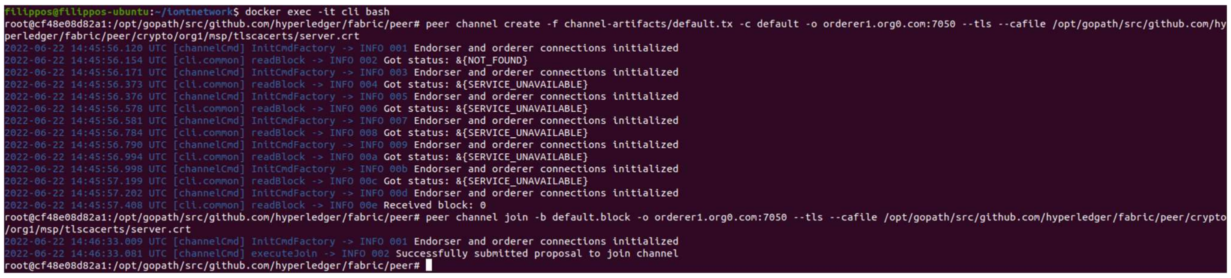
```
filippos@filippos-ubuntu:~/iomtnetwork$ fabric-ca-client enroll -u http://admin:adminpw@localhost:7054
2022/06/22 11:05:26 [INFO] Created a default configuration file at /home/filippos/iomtnetwork/rootca-admin/fabric-ca-client-config.yaml
2022/06/22 11:05:26 [INFO] generating key: &{A:ecdsa S:256}
2022/06/22 11:05:27 [INFO] encoded CSR
2022/06/22 11:05:27 [INFO] Stored client certificate at /home/filippos/iomtnetwork/rootca-admin/msp/signcerts/cert.pem
2022/06/22 11:05:27 [INFO] Stored root CA certificate at /home/filippos/iomtnetwork/rootca-admin/msp/cacerts/localhost-7054.pem
2022/06/22 11:05:27 [INFO] Stored Issuer public key at /home/filippos/iomtnetwork/rootca-admin/msp/IssuerPublicKey
2022/06/22 11:05:27 [INFO] Stored Issuer revocation public key at /home/filippos/iomtnetwork/rootca-admin/msp/IssuerRevocationPublicKey
```

Fig. 4 Generating admin certificates for CA (step 2)



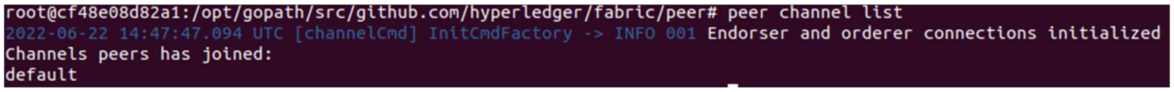
```
filippos@filippos-ubuntu:~/iomtnetwork$ nano configtx.yaml
filippos@filippos-ubuntu:~/iomtnetwork$ mkdir channel-artifacts
filippos@filippos-ubuntu:~/iomtnetwork$ configtxgen -profile genesis -outputBlock channel-artifacts/genesis.block
```

Fig. 5 Genesis block creation (step 4)



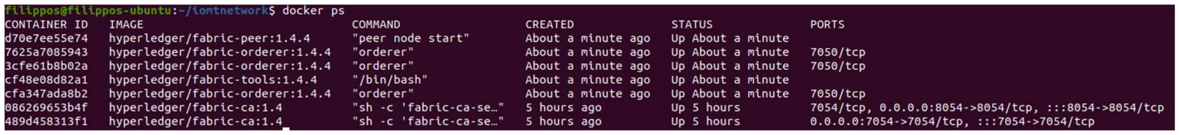
```
filippos@filippos-ubuntu:~/iomtnetwork$ docker exec -it cli bash
root@cf48e08d82a1:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer channel create -f channel-artifacts/default.tx -c default -o orderer1.org0.com:7050 --tls --cafile /opt/gopath/src/github.com/hy
perledger/fabric/peer/crypto/org1/msp/tlscacerts/server.crt
2022-06-22 14:45:56.120 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-06-22 14:45:56.154 UTC [cli.common] readBlock -> INFO 002 Got status: &{NOT_FOUND}
2022-06-22 14:45:56.171 UTC [channelCmd] InitCmdFactory -> INFO 003 Endorser and orderer connections initialized
2022-06-22 14:45:56.373 UTC [cli.common] readBlock -> INFO 004 Got status: &{SERVICE_UNAVAILABLE}
2022-06-22 14:45:56.376 UTC [channelCmd] InitCmdFactory -> INFO 005 Endorser and orderer connections initialized
2022-06-22 14:45:56.578 UTC [cli.common] readBlock -> INFO 006 Got status: &{SERVICE_UNAVAILABLE}
2022-06-22 14:45:56.581 UTC [channelCmd] InitCmdFactory -> INFO 007 Endorser and orderer connections initialized
2022-06-22 14:45:56.784 UTC [cli.common] readBlock -> INFO 008 Got status: &{SERVICE_UNAVAILABLE}
2022-06-22 14:45:56.790 UTC [channelCmd] InitCmdFactory -> INFO 009 Endorser and orderer connections initialized
2022-06-22 14:45:56.994 UTC [cli.common] readBlock -> INFO 00a Got status: &{SERVICE_UNAVAILABLE}
2022-06-22 14:45:56.998 UTC [channelCmd] InitCmdFactory -> INFO 00b Endorser and orderer connections initialized
2022-06-22 14:45:57.199 UTC [cli.common] readBlock -> INFO 00c Got status: &{SERVICE_UNAVAILABLE}
2022-06-22 14:45:57.202 UTC [channelCmd] InitCmdFactory -> INFO 00d Endorser and orderer connections initialized
2022-06-22 14:45:57.408 UTC [cli.common] readBlock -> INFO 00e Received block: 0
root@cf48e08d82a1:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer channel join -b default.block -o orderer1.org0.com:7050 --tls --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto
/org1/msp/tlscacerts/server.crt
2022-06-22 14:46:33.009 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
2022-06-22 14:46:33.081 UTC [channelCmd] executeJoin -> INFO 002 Successfully submitted proposal to join channel
root@cf48e08d82a1:/opt/gopath/src/github.com/hyperledger/fabric/peer# 
```

Fig. 6 Access to peer cli and channel connection (step 6)



```
root@cf48e08d82a1:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer channel list
2022-06-22 14:47:47.094 UTC [channelCmd] InitCmdFactory -> INFO 001 Endorser and orderer connections initialized
Channels peers has joined:
default
```

Fig. 7 Peer has joined the channel



```
filippos@filippos-ubuntu:~/iomtnetwork$ docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED              STATUS              PORTS
d70e7ee55e74   hyperledger/fabric-peer:1.4.4      "peer node start"        About a minute ago   Up About a minute
7625a7085943   hyperledger/fabric-orderer:1.4.4   "orderer"                About a minute ago   Up About a minute   7050/tcp
3cfe61b8b02a   hyperledger/fabric-orderer:1.4.4   "orderer"                About a minute ago   Up About a minute   7050/tcp
cf48e08d82a1   hyperledger/fabric-tools:1.4.4     "/bin/bash"              About a minute ago   Up About a minute
cfa347ada8b2   hyperledger/fabric-orderer:1.4.4   "orderer"                About a minute ago   Up About a minute   7050/tcp
086269653b4f   hyperledger/fabric-ca:1.4          "sh -c 'fabric-ca-se…"   5 hours ago          Up 5 hours          7054/tcp, 0.0.0.0:8054->8054/tcp, :::8054->8054/tcp
489d458313f1   hyperledger/fabric-ca:1.4          "sh -c 'fabric-ca-se…"   5 hours ago          Up 5 hours          0.0.0.0:7054->7054/tcp, :::7054->7054/tcp
```

Fig. 8 The entitites of the network in Docker containers

5) As a next step we configure the peer. To achieve this, we create a docker-compose.yaml file that contains the necessary configuration of the peer.

6) As a final step, we access the peer cli using the command docker exec -it cli. Through the peer cli we initiate the connection of the peer to the channel, as shown in Fig. 6.

Fig. 7 depicts the successful connection of the peer to the channel, while in Fig. 8 we present the entities that have been developed in the process as Docker containers.

The aforementioned HF architecture can be secure and effective against eavesdropping, spoofing and masquerading attacks due to the deployed Certificate Authority that provides TLS and MSP certificates to the nodes (peers, orderers) connected to the network. These certificates are necessary in order for a node to be able to securely communicate, transmit or receive information in the channel. Therefore, a non-certified, by these entities, node is unable to participate in the network.

## VI. Conclusion and Future Work

In this research work we have demonstrated the deployment of a HF network as a part of a an HF-based architecture for securing IoMT-based healthcare monitoring systems. We have identified the suitability of HF, as a permissioned blockchain, for IoMT networks and proceeded with the deployment of the network on a virtual environment. As future work, following the creation of the secured channel, as described above, to complete the functionality of the proposed HF architecture, the architecture should be further deployed via the proper chaincode on the setup HF network and be evaluated in terms of performance metrics such as transaction throughput, resource consumption, network use and latency.

## References

[1] G. Zachos, I. Essop, G. Mantas, K. Porfyrakis, and J. C. Ribeiro, “An Anomaly-Based Intrusion Detection System for Internet of,” pp. 1–25, 2021.

[2] E. Karavatselou, M. A. Fengou, G. Mantas, and D. Lymberopoulos, “Profile management system in ubiquitous healthcare cloud computing environment,” *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST*, vol. 263, pp. 105–114, 2019, doi: 10.1007/978-3-030-05195-2_11.

[3] M. Papaioannou *et al.*, “A Survey on Security Threats and Countermeasures in Internet of Medical Things (IoMT),” *Trans. Emerg. Telecommun. Technol.*, 2020, doi: 10.1002/ett.4049.

[4] P. Gope and T. Hwang, “BSN-Care: A Secure IoT-Based Modern Healthcare System Using Body Sensor Network,” *IEEE Sens. J.*, 2016, doi: 10.1109/JSEN.2015.2502401.

[5] S. Khezr, M. Moniruzzaman, A. Yassine, and R. Benlamri, “Blockchain technology in healthcare: A comprehensive review and directions for future research,” *Appl. Sci.*, 2019, doi: 10.3390/app9091736.

[6] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, “Anatomy of Threats to the Internet of Things,” *IEEE Commun. Surv. Tutorials*, vol. 21, no. 2, pp. 1636–1675, 2019, doi: 10.1109/COMST.2018.2874978.

[7] N. Neshenko *et al.*, “Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations,” *IEEE Commun. Surv. Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019, doi: 10.1109/COMST.2019.2910750.

[8] F. Pelekoudas-oikonomou *et al.*, “Blockchain-Based Security Mechanisms for IoMT Edge Networks in IoMT-Based Healthcare Monitoring Systems,” 2022.

[9] F. Pelekoudas Oikonomou, J. Ribeiro, G. Mantas, J. Bastos, and J. Rodriguez, “A Hyperledger Fabric-based Blockchain Architecture to Secure IoT-based Health Monitoring Systems,” 2021.

[10] F. P. Oikonomou, G. Mantas, P. Cox, F. Bashashi, F. Gil-Castineira, and J. Gonzalez, “A Blockchain-based Architecture for Secure IoT-based Health Monitoring Systems,” pp. 1–6, Dec. 2021, doi: 10.1109/CAMAD52502.2021.9617803.

[11] M. Seliem and K. Elgazzar, “BIoMT: Blockchain for the internet of medical things,” 2019, doi: 10.1109/BlackSeaCom.2019.8812784.

[12] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in Internet of things: The road ahead,” *Computer Networks*. 2015, doi: 10.1016/j.comnet.2014.11.008.

[13] E. Androulaki *et al.*, “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains,” *Proc. 13th EuroSys Conf. EuroSys 2018*, vol. 2018-January, Apr. 2018, doi: 10.1145/3190508.3190538.

[14] N. Lu, Y. Zhang, W. Shi, S. Kumari, and K. K. R. Choo, “A secure and scalable data integrity auditing scheme based on hyperledger fabric,” *Comput. Secur.*, vol. 92, p. 101741, 2020, doi: 10.1016/j.cose.2020.101741.

[15] S. Kakei, Y. Shiraishi, M. Mohri, T. Nakamura, M. Hashimoto, and S. Saito, “Cross-Certification towards Distributed Authentication Infrastructure: A Case of Hyperledger Fabric,” *IEEE Access*, vol. 8, pp. 135742–135757, 2020, doi: 10.1109/ACCESS.2020.3011137.

[16] V. A. Siris, D. Dimopoulos, N. Fotiou, S. Voulgaris, and G. C. Polyzos, “Decentralized authorization in constrained IoT environments exploiting interledger mechanisms,” *Comput. Commun.*, vol. 152, no. September 2019, pp. 243–251, 2020, doi: 10.1016/j.comcom.2020.01.030.

[17] C. Chen, J. Yang, W. J. Tsaur, W. Weng, C. Wu, and X. Wei, “Enterprise Data Sharing with Privacy-Preserved Based on Hyperledger Fabric Blockchain in IIOT’s Application,” *Sensors*, vol. 22, no. 3, pp. 1–23, 2022, doi: 10.3390/s22031146.

[18] H. Liu, D. Han, and D. Li, “Fabric-iot: A Blockchain-Based Access Control System in IoT,” *IEEE Access*, vol. 8, pp. 18207–18218, 2020, doi: 10.1109/ACCESS.2020.2968492.

[19] D. H. Shih, T. W. Wu, M. H. Shih, G. W. Chen, and D. C. Yen, “Hyperledger Fabric Access Control for Industrial Internet of Things,” *Appl. Sci.*, vol. 12, no. 6, 2022, doi: 10.3390/app12063125.

[20] A. Iftekhar, X. Cui, Q. Tao, and C. Zheng, “Hyperledger fabric access control system for internet of things layer in blockchain-based applications,” *Entropy*, vol. 23, no. 8, 2021, doi: 10.3390/e23081054.

[21] H. H. Pajooh, M. Rashid, F. Alam, and S. Demidenko, “Hyperledger fabric blockchain for securing the edge internet of things,” *Sensors (Switzerland)*, vol. 21, no. 2, pp. 1–29, 2021, doi: 10.3390/s21020359.

[22] J. Maeng, Y. Heo, and I. Joe, “Hyperledger Fabric-Based Lightweight Group Management (H-LGM) for IoT Devices,” *IEEE Access*, vol. 10, pp. 56401–56409, 2022, doi: 10.1109/access.2022.3177270.

[23] “Projects - Linux Foundation.” https://www.linuxfoundation.org/projects/ (accessed Jun. 08, 2022).

[24] “No Title.” https://hyperledger-fabric.readthedocs.io/en/release-2.2/.

[25] S. Shalaby, A. A. Abdellatif, A. Al-Ali, A. Mohamed, A. Erbad, and M. Guizani, “Performance Evaluation of Hyperledger Fabric,” *2020 IEEE Int. Conf. Informatics, IoT, Enabling Technol. ICIoT 2020*, pp. 608–613, 2020, doi: 10.1109/ICIoT48696.2020.9089614.

[26] “X.509 : Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks.” https://www.itu.int/rec/T-REC-X.509-201910-I/en (accessed Jun. 08, 2022).

[27] “Membership Service Providers (MSP) — hyperledger-fabricdocs main documentation.” https://hyperledger-fabric.readthedocs.io/en/release-2.2/msp.html (accessed Jun. 22, 2022).