Demo: Blockchain-based Secured and Federated Slice Broker (SFSBroker)

Tharaka Hewa*, Nisita Weerasinghe*, Pawani Porambage*, Madhusanka Liyanage*‡, Mika Ylianttila*
*Centre for Wireless Communications, University of Oulu, Finland
†School of Computer Science, University College Dublin, Ireland
Email: *[firstname.lastname]@oulu.fi, †madhusanka@ucd.ie

NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium | 978-1-6654-0601-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/NOMS54207.2022.9789764

***Abstract*—Network slicing is a versatile and distinguishing capability of the 5th and 6th Generation (5G & 6G) mobile networks. Network slicing enables the consumers to deliver individualized and customized telecommunication services on the commonly shared infrastructure. Slice brokering is the dedicated service to facilitate the tenants and resource providers in slice allocation process. We proposed SFSBroker (Secured and Federated Slice Broker) to leverage the slice brokering process with the application of game theory. We formulated the optimal slice selection problem into the Stackelberg game model. The computational logic has been incorporated on smart contracts. Furthermore, the proposed architecture includes Security Service Blockchain (SSB) which has been integrated for Denial of Service (DoS) attack prevention. We implemented the end to end setup including tenant request to slice creation using Hyperledger Fabric blockchain, Katana slice manager, and OpenStack. In this demonstration, the system provisions to obtain hands-on experience on the end to end workflow of slice brokering process. *Index Terms*—Blockchain, Network slicing, Game theory, Hyperledger fabric**

I. Introduction

The evolution of the telecommunication technologies in to 5th and 6th Generation(5G & 6G) leverage more sophisticated consumer use cases in different industrial contexts. The connectivity business models have been revamped towards more consumer-centric and diversified service delivery capabilities. The modern telecommunication business models seamlessly extend towards service oriented models such as infrastructure, secured computation, data storage, and data sharing by bridging the gaps between consumers and service infrastructure.

Network slicing is one of the distinguishing strength of the 5G and beyond telecommunication technology. Network slicing enables the consumers to share the common infrastructure for different requirements. Ideally, network slicing enables the 5G infrastructure providers to deliver more individualized service provisions on top of the sole infrastructure. Each network slice comprises of consumer-specific service functions which are ideally shaped towards the different applications. The network slices operate independently on top of the shared infrastructure. Especially, the variety of business network slicing varieties extend with the evolution of local 5G operators, which function as regional and localized resource providers for different industry verticals.

Network slice brokering is a dedicated service to leverage the slice selection operation efficiently. In general, slice brokering service interconnects the consumers and resource providers in order to facilitate optimal slice. There are different efforts to improve the slice selection process.

Game theory [1], [2] has been applied to improve the slice selection process. The selection operation has been incorporated into game models. The players of such game models consist of tenants and resource providers who have well-defined objectives in terms of profit and usability. In contrast, network slicing in multi-operator and multi-tenant scenarios require on-demand federation of resource providers per request basis along with low latency selection operation and high scalability to handle massive consumer groups.

Distributed Ledger Technology (DLT) is one of the most blessing technology with many potential strengths in the future telecommunications [3] industries. The core principle of DLT is the cryptographically verified distributive storage of the entire database of records (i.e., digital ledger) at all the nodes in a network. Thus, DLT aims to eliminate the use of a centralized server and brings in place a decentralized services encoded as smart contracts. As the most popular DLT, blockchain comprises immutable and timestamped blocks containing validated transactions and connected using hash-based chain and timestamps. The transactions will be approved upon the approval within the members of blockchain network, which is known as the consensus process.

In this paper, we demonstrate the end to end functional application of the game theory-based selection algorithm in a near realistic implementation. We encoded the selection algorithm into the smart contracts to leverage the selection operation with decentralization to ensure lower latency and consumer scalability. Furthermore, we incorporated the Security Service Blockchain (SSB) that verifies the input and output transaction request are secured against Denial of Service (DoS) attacks. We used Hyperledger Fabric blockchain platform as the blockchain service for SSB and SFSBroker. Furthermore, the infrastructure provider simulated using OpenStack with Katana slice manager integration. The implementation setup provisions ideally demonstrates the end to end slice creation process as well as tolerance of the SFSBroker with simulated DoS attacks.

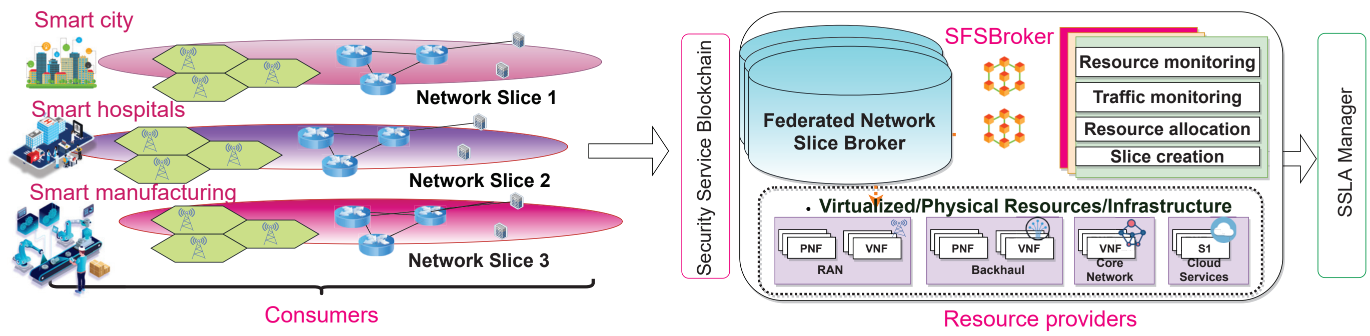

Figure 1. A use case scenario for SFSBroker that serves multiple tenants in different application domains.

II. System Architecture

IoT tenants, SSB, SFSBroker and the resource providers are the three distinguishing stakeholders in the proposed architecture. IoT tenants represent the consumer end. Resource providers represent the local 5G operators and MNOs that offer the resources to trade within the brokering process, SFSBroker is the core part of the proposal, which has been integrated with the IoT tenants and resource providers to facilitate brokering process. The selection algorithm has been encoded as the smart contracts on blockchain. Figure 1 reflects the SFSBroker architecture. The three distinguishing components of the SFSBroker are as follows.

A. SFSBroker core components

Security Service Blockchain: Security service blockchain validates the resource requests and resource offers are within the legitimate range. SSB is based on the original work [4]. SSB stores individual profiles of tenants and resource providers which define the valid range of resource providers and resource offers.

Resource unit price database: Stores the individual resources and the unit prices of each resource provided by the corresponding MNOs. The resource unit price database is synchronized with the pricing information updated by the respective MNOs whenever required.

Prime mover: A smart contract which handles the consumer slice requests, formulate the network slice blueprint according to the requirement and forwards to the mediator.

Mediator: A smart contract which retrieves the unit prices of resources from the storage database and compute the best matching MNOs for the federated slice formulation. The slice selection operation is performed by a game theory based selection algorithm encoded in the smart contract.

SSLA Manager: Secured service level agreement manager establishes the service level agreement between the IoT tenant and resource providers for the requested slice.

Global slice manager: This interacts with the network slice managers (i.e., Katana slice manager) deployed in each MNOs for the federated network slice creation. The North Bound Interface(NBI) REST API of Katana slice manager is integrated with the slice manager smart contract of SFSBroker [5]. The global slice manager is not restricted to Katana and can be customized to integrate with slice manager.

B. SFSBroker initialization

The SFSBroker requires timely synchronization with the resource availability with the resource providers. The resource provider availability is recorded in a database. SSB validates the resource offers prior to include in the unit price database. The database is being accessed by the SFSBroker service at the computation of optimal resource requests to deliver consumer resource request. Furthermore, blockchain service requires instantiation with the encoded smart contract with SFSBroker logic.

C. SFSBroker workflow

SFSBroker workflow includes the end-to-end processes of the slice creation process. The key steps are as follows.

*1) **Resource request initiated from the tenant**:* The tenant generates the resource request for the slice creation. The resource request comprises of the quantities of each units for the slice. SSB validates resource requests for the valid range.

*2) **Slice blueprint creation**:* Prime mover collects the resource requests coming from the consumer and forwards *Mediator* upon creation of the blueprint.

*3) **Slice selection** :* Mediator runs the slice selection algorithm by taking the updates of resource availability and price from the resource unit price database, creating the Network Slice Template (NST), and sending NST to *Global Slice Manager*. Finally, the *Global Slice Manager* accumulates the slice with the coordination of slice managers and SSLA manager, and send the slice to consumers.

*4) **Slice instantiation**:* The slice instantiation includes invocation of the individual slice managers of the resource providers. In general, slice managers provide APIs which can be invoked for slice instantiation.

III. Prototype Setup

The prototype setup facilitates the end-to-end slice selection process. The demonstration setup implemented all components on the proposed architecture using different software modules. Each and every components have been deployed on appropriate infrastructure as per the requirement. Table I reflects the summary of software components and infrastructure placement in the demo setup. Entire system is adapted to the microservices architecture. Thus, most of the services are deployable as Docker containers. Figure 2 illustrates the workflow of significant steps in the proposed architecture.

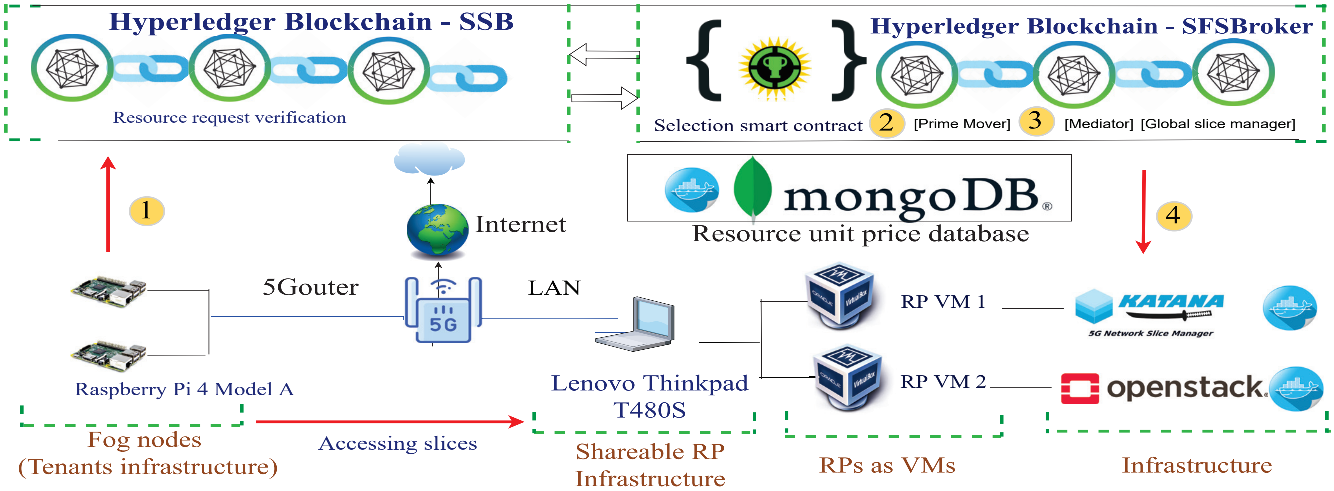

Figure 2. Implementation setup

Table I
SUMMARY OF THE IMPLEMENTATION AND DEPLOYMENT

Component	**Software**	**Infrastructure**
IoT tenants	Java request generator	Raspberry Pi Model 4A
Resource unit price database	Mongo DB	Cloud server
SFSBroker service	Hyperledger	Cloud server
SSB service	Hyperledger	Cloud server
SSLA manager	SpringBoot API simulator	Cloud server
Shareable infrastructure	OpenStack	Cloud server
Slice manager	Katana	Cloud server

IV. EXPERIMENTS IN THE DEMONSTRATION

A. End to end slice creation demonstration

This demonstrates the end to end slice creation operation. The users can trigger the resource request through the web application. The users can also enable/disable the resource request verification to realize the impact on SSB. The proposed setup reflects the individual latencies on significant steps of the end to end slice creation process.

B. Manage the created slice through Katana

The created slice can be viewed through the Katana command line interface. Katana command line provisions the management of network slice. The demonstration setup facilitates the user to interact with Katana slice manager to interact hands-on with Katana slice manager.

C. Impact of the SSB on DoS prevention

The proposed experimental setup can be used to demonstrate the impact of SSB in securing SFSBroker for the simulated DoS attacks.

D. End to end latency with $BlockTime$ configuration

The proposed demonstration setup implemented with consortium blockchain, which enables the configuration of block mining time($BlockTime$). The implementation setup facilitates the users to realize the impact of block mining time for the end to end slice creation latency.

V. TECHNICAL REQUIREMENTS

The proposed architecture includes SSB and SFSBroker, which are two blockchain-based services. Therefore, two virtual machines required to deploy each blockchain instance. The summary of technical requirements for the virtual machines included in Table II.

Table II
VIRTUAL SYSTEM SPECIFICATIONS.

Operating system	Ubuntu 18.04
Docker Version	19.03.12
TCP ports	Selected ports open
Connectivity	Access through public IP

VI. ACKNOWLEDGEMENT

This work has been performed under 6Genesis Flagship (grant 318927) project. The research leading to these results partly received funding from European Union's Horizon 2020 research and innovation program under grant agreement no 871808 (5G PPP project INSPIRE-5Gplus). The paper reflects only the authors' views. The Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] M. Vincenzi, A. Antonopoulos, E. Kartsakli, J. Vardakas, L. Alonso, and C. Verikoukis, "Multi-tenant Slicing for Spectrum Management on the Road to 5G," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 118–125, 2017.
[2] N. Zhao, H. Wu, and Y. Chen, "Coalition game-based computation resource allocation for wireless blockchain networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8507–8518, 2019.
[3] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for 5G and beyond networks: A state of the art survey," *Journal of Network and Computer Applications*, p. 102693, 2020.
[4] T. Hewa, A. Kalla, P. Porambage, M. Liyanage, and M. Ylianttila, "How dos attacks can be mounted on network slice broker and can they be mitigated using blockchain?" in *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2021, pp. 1525–1531.
[5] "Katana Slice Manager," https://github.com/medianetlab/katana-slice-manager/wiki, Accessed on 24.02.2021.