User profile and mobile number portability for beyond 5G: blockchain-based solution

Fariba Ghaffari, Emmanuel Bertin, Noel Crespi

User Profile and Mobile Number Portability for Beyond 5G: Blockchain-based Solution

[1, 2] Fariba Ghaffari, [1, 2] Emmanuel Bertin, *Senior Member, IEEE*, [2] Noel Crespi, *Senior Member, IEEE*
[1] *Orange Innovation, 14000 Caen, France*
[2] *SAMOVAR, Telecom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France*
{fariba.ghaffari, emmanuel.bertin}@orange.com, and noel.crespi@it-sudparis.eu

***Abstract*—Mobile Number Portability (MNP) as a regulatory requirement for Mobile Network Operators (MNO), allows users to easily switch their operators while keeping their number. User subscription, profile management, and MNP procedures in current MNOs are handled in centralized systems and databases that rely on trusted third parties. Having a single point of failure, low scalability, high latency, imposed porting fees, data leakage in the central database, low availability, and need for re-subscription while porting procedure are the most highlighted defects of the conventional systems. Addressing these issues, we propose a Blockchain-based system to 1) manage user subscription and user profile/identity in a distributed database, and 2) mobile number and profile porting procedure. This solution eliminates any central point via managing the processes using smart contracts. Moreover, it decreases IT complexity, increases automation, provides low latency, and high confidentiality. The experiments confirm that this solution can provide fast and scalable profile management and porting solution.**

***Index Terms*—User profile management, subscription, Mobile number porting, IPFS, Blockchain, smart contract.**

I. Introduction

Due to the ever-growing number of mobile subscribers in Mobile Network Operators (MNO), and the variety of services, coverage, price, etc., it is the user's right to be able to switch among MNOs based on their needs and preference. One crucial requirement of freely porting among MNOs is the capability of keeping the previous number. This process is called Mobile Number Porting (MNP) which allows the subscribers to keep their phone number in the process of porting from one MNO to another [1], [2]. MNP is a regulatory requirement for Mobile Network Operators (MNO) in over 100 countries worldwide that enhances the competitiveness in the cellular network market, decreases the imposed prices and services, helps to improve innovation among MNOs, and generally can increase user welfare and satisfaction [3]. The 40% rate of mobile number porting in 2019, indicates the importance of this mechanism.

Currently, Mobile number portability Clearance House (MCH), is a centralized operator that manages the whole MNP procedure. This centralized MCH not only can be a single point of failure for availability and a bottleneck for the performance of the porting procedure but also MNOs need to trust MCH which can pose a threat to user data protection [4]. Moreover, the MNOs and users need to execute some porting procedures manually and some of them are repetitive processes. For instance, since while MNP, the user's profile is not ported from Donor Network Operator (DNO, i.e., the source MNO) to Recipient Network Operator(RNO, i.e., the target MNO), the whole user subscription procedure needs to be done from scratch. Furthermore, the key management procedure of storing the user's data in centralized storage in MNOs is another challenging issue that can increase the processing load of the MNOs [5]. In this regard, an alternative solution would be a game-changer, if it can help the procedure to be executed faster, more transparent, automated, and secure.

Benefiting from Blockchain technology [6] and its extension smart contracts [7], [8], in this paper we propose a *profile and mobile number porting* system for beyond 5G cellular networks that aims to: 1) eliminate the central points in the MNP and profile management procedures to decrease the inherited threats and the operational complexity, 2) provide a faster, more transparent, and secure method for porting, 3) increase the MNP automation, 4) decrease the MNP cost (note that in some countries users need to pat MNP fee), and 5) increase the security of user's profile management via storing the data in a distributed database and using hybrid (symmetric-asymmetric) cryptography system. The main ***contributions*** of this paper are as follows:

1) *Outsourcing the user profile management* in cellular networks to a distributed system based on Blockchain and eliminating any trusted party such as MCH;
2) Introducing very *fast mobile number porting* procedure;
3) *Porting the user's profile* (along with mobile number) to decrease the repetitive user-side process.
4) The system provides *forward and backward secrecy* regarding the user's stored data in a distributed database (i.e., when the user switches between MNOs, DNO/RNO can not have any access to the future/ previous keys/data, respectively.)
5) Provide high user *privacy and data confidentiality* using hybrid cryptography system.
6) *Removing the DNO from the porting procedure* can increase the porting speed.
7) *Providing automation* in profile management and porting procedure using smart contracts as secure, immutable, and distributed entities.

Paper organization: Section II provides the preliminaries regarding MNP, Blockchain, smart contracts and current profile management in cellular network. Then, III briefly surveys

state of the arts in porting and user profile management methods. In Section IV we outline the problems of the existing methods and presents our proposed solution. The detailed design and construction of our proposed method is provided in Section V, followed by the experiment in section VI. Section VII provides the conclusions about the proposed method as well as some open challenges for future research directions.

II. Background

A. Mobile Number Porting (MNP)

Mobile number porting is implemented in two general ways across the globe: Recipient-led and Donor-led [9]. In the first solution, the Recipient Network Operator(RNO) is the one who arranges the required process with the Donor Network Operator (DNO), while in the latter solution, the subscribers need to contact the DNO to obtain a Porting Authorization Code (PAC), which they need to give it to the RNO for the further porting process. The Recipient-led solution is the most dominant porting method. Regardless of the MNP approach, the porting procedure has four main steps:

1) *Request:* The subscriber would request RNO to start the porting procedure (or request DNO for PAC).
2) *Validation:* The RNO verifies the request by sending a validation request to DNO. Note that this request can be sent directly, or through another trusted party.
3) *Clearance:* DNO manages the legal clearance from a legal authority (Mobile Clearance House-MCH) to assure that the number doesn't have any legal issues.
4) *Activation:* Once, the RNO receives the clearance notification, asks the DNO to remove the subscribers from its users and add them as the user of RNO.

B. User profile management in MNOs

Unified Data Repository (UDR) [10] in 5G Service-Based Architecture (SBA) serves as a centralized data repository in each MNO for the subscription data, user profile, policies, and application data. Other network functions are connected to this component using standard APIs to fetch, update, and delete data. One of the main challenges regarding UDR is its centralized nature, the possibility of being attacked to obtain unauthorized access to sensitive data on other network functions, loss of availability, and loss of integrity [11].

C. Blockchain and smart contract

Distributed ledger technology (DLT) is a general term for technologies that utilize replicated, shared, and synchronized digital data among the users of distributed computers located on multiple sites providing immutability, distributed network, consensus, transparency, and non-repudiation. Blockchain, as the first extension of DLT, was introduced as a distributed ledger of transactions fitted into blocks and connected as a chain. Firstly, the Blockchain was introduced to only support P2P financial transactions. The concept of smart contracts has been put into practice by Ethereum in 2015. Smart contracts are some pieces of code, executed on top of Blockchain in a distributed manner, that would be run only when the predefined conditions are satisfied. These contracts minimize exceptions and eliminate the need for trusted intermediaries. Due to their unique features, these technologies can bring many unprecedented opportunities in cellular networks, healthcare, IoT, cloud computing, etc.

TABLE I: Comparison of state of the arts

Features	***Existing***	[4]	[9]	[15]	[14]	[12]
Central point elimination	No	No	No	No	No	No
Confidentiality	Yes	No	NA*	No	NA	Yes
Distributed Database	No	Yes	No	No	No	No
Fast porting	No	No	NA	No	No	Yes
Forward/backward secrecy	NA	No	NA	No	No	NA
Performance analysis	NA	Yes	No	No	No	Yes
Porting user profile	No	No	No	No	No	Yes

* Not Applicable

III. Related works

In our previous work [12], we proposed a similar MNP solution. This method works on top of the existing centralized system which makes it vulnerable to DoS/DDoS attacks. Moreover, DNO/RNO validation for porting the user profile increases the latency in the whole procedure and decreases the automation level. Shah et al. [4], proposed a Blockchain-based MNP scheme on top of Ethereum. This method provides transparency, cost and time efficiency, and a secure call routing mechanism using deployed smart contracts. Moreover, Krishnaswamy et al. [9] proposed a Blockchain-based framework for mobile number porting in private Hyperledger Fabric [13]. This method suffers from having a single point of failure and the non-availability of performance analysis. Apart from the Blockchain-based solutions, a significant number of studies provided MNP in a decentralized manner. For instance, Chen et al. [14] proposed a call routing mechanism to support enum-based mobile number porting. Moreover, Odii et al. [15] proposed a hybrid solution to support MNP and call routing.

Table I compares the state of the arts, based on their general features (e.g., removing the single point of failure, data confidentiality, distributed database, fast-porting, forward secrecy, backward secrecy, performance analysis, etc.). The proposed method addresses these issues by removing the single point of failure, proposing distributed system interacting with a distributed database, and providing confidentiality, integrity, forward/backward secrecy, and a fast porting procedure.

IV. Problem statement and system overview

To explain the defects in the existing user profile management and porting processes, let's assume that a user (u) wants to subscribe to MNO and then switch to RNO. Currently, this process is done in the following centralized manner:

1) *User Subscription*
   - The SIM-Card activation of u in MNO (that is required to record the user's profile in MNO), is handled by central servers, and generally, in a real-world scenario, it takes several hours for activation.
   - While the activation process, the user's encrypted data would be stored in a centralized database in

MNO. In this step, MNO executes a key management procedure to create, store, update, and revoke the data encryption keys.
- Once the user's data is recorded in the MNO's database, the user's subscription procedure is finished and she can profit from services.

2) *porting:*
   a) in the *Request* phase, u requests DNO to start the porting procedure; for the successful requests, a verification code would be sent to u.
   b) In the *Validation* step, RNO validates the user and the request by sending the validation request to DNO through a centralized party (e.g., MCH).
   c) For *Clearance*, DNO manage the legal clearance from a legal authority.
   d) Finally, for *Activation*, Once, MCH receives the clearance notification, it asks DNO to revoke the user's subscription and asks RNO to add new user.

We have identified several drawbacks in the existing model (i.e., with/without MNP applications), as follows:

- The subscription and porting procedures are highly time-consuming in current cellular networks.
- User profile would not be ported to the RNO, so, the user needs to repeat all subscription procedures in RNO.
- Centralized servers to manage the user's request, key management, and the centralized database for the user's identity can be a single point of failure (e.g., the risk of losing the user's data in a centralized server).
- whole this procedure can have a high processing load on MNO as a central point.
- MNOs need to trust a third party (e.g., MCH), which can pose a threat to data protection and brings trust issues.
- In the majority of countries the users need to pay a porting fee (to MNO or third party).

Addressing these constraints, this paper proposes a new user profile management and mobile number and profile porting solution on top of Blockchain and smart contracts. The proposed method relies on the Blockchain addresses as an identifier, the user's key pair in Blockchain for authentication and key management, smart contracts for process automation, and distributed database to manage the user's profile. It is important to mention that we assume having a governance body that validated the identity of the MNOs before inserting them into the system (similar to the existing real-world scenario in which MNOs need the allowance of a governance body to operate).

The subscription porting and termination procedures in the proposed system are as follows:

1) *User subscription:*
   a) The user sends its request to the Blockchain to deploy a unique smart contract.
   b) The user fills out the off-chain subscription form.
   c) MNO creates a file of identity and phone number.
   d) MNO requests the Blockchain to obtain ownership of the user's data.
   e) MNO stores the user's encrypted data in IPFS, and stores its link in the user smart contract.
2) *Porting:*
   a) The user sends her port request to the Blockchain which redirects her to RNO to submit the request.
   b) RNO verifies the request using `port` contract.
   c) RNO requests the user to decrypt her profile data.
   d) RNO requests `port` contract to delegate the ownership of user data.
   e) RNO stores the user's encrypted data in IPFS, and stores its link in the user smart contract.
   f) After updating the user's profile, porting smart contract updates the specific contract of DNO and RNO to remove/add the user from/to their list.
3) *Revocation of user subscription:* u sends the request to the Blockchain. Then, porting a smart contract would destroy the user's contract and remove it from the list of its associated MNO.

V. System design

The proposed method consists of three main steps: 1) User subscription, 2) porting MNO, and 3) Subscription termination. It is important to mention that the general assumptions of the proposed method are as follows:

- Off-chain connections (i.e., the connections outside of the Blockchain) are secure.
- User equipment supports e-SIM in which the user's Blockchain address (Ad_u) and public/private key pair (Pub_u, Pr_u) are hard-coded.
- regulatory body is responsible authenticate the MNOs, and `subscription` and `port` smart contracts.
- Regulatory bodies and MNOs participate in Blockchain's consensus procedure.

A. Designed smart contracts

The designed smart contracts are as follows:

1) *Address book (SC_{AB})* stores the addresses of the subscription, Port management, MNO list, and User list smart contracts, to make their collaboration secure. This contract maps the names of contracts to their addresses:
$$AddressBook \xleftarrow{name_{SC}} Ad_{SC}$$
where $name_{SC}$ are predefined names for single contracts (e.g., "sub" for subscription), and Ad_{SC} is its address in Blockchain. Note that the purposes of designing this contract are 1) avoiding using hard-coded addresses to evade maintainability defects of smart contracts [16], 2) having a list of predefined addresses to manage the function execution capability, based on different caller smart contracts, and 3) avoiding data falsification in subscription/porting procedures by forged smart contracts advertised to the users by an attacker. Moreover, using this smart contract makes it impossible to calling the distrusted external smart contracts,so, it helps protecting the system against Reentrancy attack [17].

2) *User smart contract (SC_U)* is a specific contract for the users which stores, at least, the following attributes:

$$Attr_u \xleftarrow{Ad_u} [Number_u, CID_{EN^M_{K_s}}, EN^{K_s}_{Pub_u}, EN^{K_s}_{Pub_{MNO}}, Hash(M)]$$

where $Number_u$ is the user's phone number, $CID_{EN^M_{K_s}}$ is the access identifier of IPFS storage (more details in Section V-B). Here, M is the user's identity, wrapped into a file, that can contain the user's Personally Identifiable Information (PII) (e.g., name/family, address, IMSI). K_s is a symmetric key generated by MNO to encrypt user data. $EN^M_{K_s}$ is M encrypted by K_s. $Hash(M)$ is the hash of plain-text M. $EN^{K_s}_{Pub_u}$ and $EN^{K_s}_{Pub_{MNO}}$ are the K_s encrypted by the user's and MNO's public key, respectively.

3) *User List smart contract (SC_{UL})* stores the list of subscribed users with the following structure:

$$UserList \xleftarrow{Ad_u} [Ad_{SC_u}, Code_{MNO}]$$

where $Code_{MNO}$ is a unique code dedicated for each MNO and indicates the user's current MNO.

4) MNO smart contract (SC_{MNO}) is a unique specific contract for each MNO, deployed by a regulatory body, at the time of its subscription in the system. This contract stores, at least, MNO's subscribers and the list of user subscription/port requests.

5) MNO list smart contract (SC_{MNOL}) is a single smart contract, owned by the regulatory body, to keep the list of trusted and validated MNOs. This contract keeps the MNO information in the following structure:

$$MNOL \xleftarrow{Code_{MNO}} [Ad_{MNO}, Ad_{SC_{MNO}}]$$

6) *Subscription smart contract (SC_{sub})* handles the user subscription procedure by deploying a unique SC_u for that particular user and adding her in SC_{UL}, and SC_{MNO}. Moreover, setting the ownership of the user data to the MNO is handled by this contract.

7) *Port management smart contract (SC_{port})* is a single smart contract dedicated to handling the porting process or termination of the user's subscription. To port the user, after validating the user's request, this contract removes the user from SC_{DNO} and adds her into SC_{RNO}. Moreover, this contract delegates the ownership of the user's data to RNO. In the termination procedure, this smart contract removes the user from SC_{MNO} and SC_{UL} and destroys SC_u.

B. User subscription

In this phase, the user aims to subscribe to one of the pre-validated MNOs. Following is the process of user (u) subscription in MNO (See Fig. 1):

1) u sends her subscription request by sending the following transaction to SC_{sub}:

$$< Code_{MNO}, Hash(nonce) >,$$

where $nonce$ is a random number generated by user, and $Hash(nonce)$ is the hash amount of the $nonce$ calculated by $Keccak-256$ [18] algorithm.

2) SC_{sub} receives the request and verifies that Ad_u is not recorded as subscribed user in SC_{UL}. Then, it verifies the $Code_{MNO}$ from SC_{MNOL} to ensure that the MNO is approved by the regulatory. If all conditions are passed, SC_{sub} deploys a SC_U for the user.

3) SC_{sub} inserts $Data^U_{sub}$ (See Fig. 1) in the $UserList$ in SC_{UL}. Moreover, it inserts the summary of the user's request in SC_{MNO} using $Data^{MNO}_{sub}$ (See Fig. 1). In $Data_{sub_{MNO}}$, St_{req} shows the progress of the request; It can be 1 if the request is demanded, 2 when request is validated, and 3 when the request is terminated. Note that, inserting subscription, and port requests to SC_{MNO} are restricted to SC_{sub} and SC_{port}, respectively.

4) Ad_{SC_U} would be transferred to the user as the result request. Then u is redirected to the subscription page of the MNO. To continue the procedure, the user sends $< Ad_u, nonce >$ to the MNO.

5) After receiving the user's request, MNO needs to assure that the request is valid and that the user has just been subscribed in the system. To do so, MNO asks SC_{MNO} to validate the request by sending $< Ad_u, nonce >$. We can claim the request is valid, if the following requirements pass:

   - $H'(nonce) == Hash(nonce)$, where $H'(nonce)$ is the hash of $nonce$ which has just been sent by u, and $Hash(nonce)$ is stored in SC_{MNO} in ***Step 3***.
   - $St_{req} == 1$

   If all conditions are passed, SC_{MNO} confirms the request and changes St_{req} to 2. Note that, from the security perspective, the user's request to the MNO is not recoverable by the potential attackers. More precisely, in ***Step 1***, the user sends the $Hash(nonce)$ to the Blockchain, and in ***Step 3***, she sends the $nonce$ itself. Since the hash function is a one-way function, we can claim that only the user can send this request.

6) User is redirected to a web page to fill in her data.

7) MNO receives the user's identity data (M). As mentioned before, we need to strictly limit the data access to MNO and u. To do so, we used a hybrid cryptosystem for a multi-user environment. The hybrid cryptosystem is a technique of combining symmetric and asymmetric cryptography algorithms. To apply this method we execute the following steps:

   - MNO generates symmetric key K_s;
   - MNO encrypt K_s using Pub_u and Pub_{MNO} and gets $EN^{K_s}_{Pub_u}$ and $EN^{K_s}_{Pub_{MNO}}$
   - MNO encrypts M with K_s to get $EN^M_{K_s}$

8) MNO stores $EN^M_{K_s}$ in IFPS as a distributed database. After storing the data in IPFS, it would be indexed by a cryptographic hash function, which results in returning its unique content identifier (CID) to MNO. The CID (let's call it $CID_{EN^M_{K_s}}$) can be used for further access

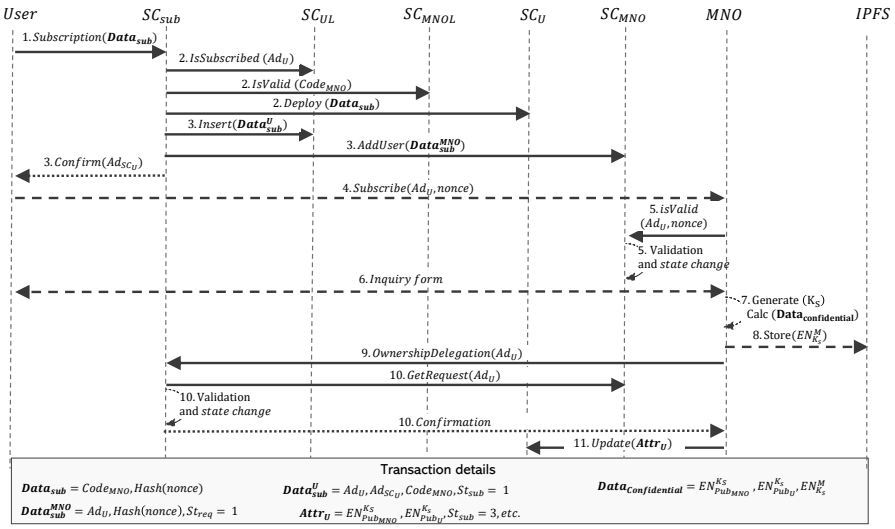


Fig. 1: User subscription procedure

to the data in IPFS.

9) To modify $Attr_u$ (See mapping provided in 2), MNO requires to have the ownership of updating function in SC_U; So, it requests SC_{sub} to execute the delegation procedure by sending the Ad_u to it.
10) SC_{sub} retrieves the user request from SC_{MNO} and makes sure that the user requested for this subscription. To do so, it checks the request of Ad_u and verifies that $St_{req} == 2$. If the validation is successful, the ownership of the update function in SC_U will be given to MNO. Note that, the address of the SC_{sub} is immutably written in SC_{AB}. So, SC_U can be assured that SC_{sub} is an eligible contract to change ownership. After changing the ownership, SC_{sub} updates the St_{req} in SC_{MNO} from 2 to 3 and inserts $Number_u$ in the list of the MNOs active users.
11) After initiating the ownership, MNO can store $Attr_u$ into SC_U (See the mapping provided in 2, Section V-A).

Using this procedure, the user's data can be retrieved either by the user or the MNO. The other MNOs even if they achieve to download the data, won't be able to open it.

C. Porting procedure

In this phase, we assume that u has already been subscribed to DNO, and aims to change her MNO to RNO while keeping $Number_u$. The procedure is as follows (Fig. 2):

1) u sends the porting request to SC_{port} by creating a transaction in the Blockchain, and sending:
$$< Code_{RNO}, Pub_U >,$$
2) SC_{port} verifies the user's record in the list of subscribed users, and if the user was subscribed, SC_{port} inserts the summary of the user's request in SC_{RNO}), using the following data:
$$< Ad_u, Pub_U, Add_{SC_U}, St_{port} = 1 >$$
3) The request result would be sent to u, which redirects her to the port request page of RNO. User sends $< Ad_u >$ to RNO and can select her proffered plan in the new operator.
4) To validate the user request, RNO asks SC_{RNO} to confirm the user's request (i.e., Ad_U is already stored there), and verify $St_{req} == 1$. If these confitions passed, RNO will authenticate the user using her Pub_U that is stored in SC_{RNO}. This step can be done by sending a challenge (encrypted by Pub_U) to u and ask her to decrypt and resend it. This authentication would assure RNO that the eligible user is requesting for porting. If all conditions passed, SC_{RNO} changes the St_{req} to 2.
5) Since the user data is stored in IPFS, and only DNO and the user can have access to that, RNO asks the user to send the decrypted data. User retrieves $EN^{M_{IPFS}}_{K_s}$ from InterPlanetary File System (IPFS), using $CID_{EN^M_{K_s}}$. She executes:
   - Retrieves $EN^{K_s}_{Pub_u}$ from SC_U;
   - Decrypts it with Pr_u and retrieves K_s;
   - Decrypts $EN^{M_{IPFS}}_{K_s}$ using K_s and retrieves M_{IPFS}

   Then, user sends the data, lets call it M_U, to RNO. Note that, since the user is not eligible to modify her identity, RNO needs to verify the authenticity and originality of the received data.
6) RNO receives M_U and validate its integrity with the previous version which is validated by DNO. To do so, SC_{RNO} retrieves $Hash(M_{IPFS})$ from SC_U. Then validates that $Hash(M_{IPFS}) == Hash(M_U)$. After successful validation, RNO generates new symmetric key, K_{s_2}, and calculates $EN^M_{K_{s_2}}$, $EN^{K_{s_2}}_{Pub_u}$ and $EN^{K_{s_2}}_{Pub_{MNO}}$. RNO stores $EN^M_{K_{s_2}}$ in IPFS and gets $CID_{EN^M_{K_{s_2}}}$).
7) RNO requests SC_{port} to delegate the ownership of update function of SC_U to RNO. SC_{port} gets the record of user request and verifies that $St_{req} == 2$. If the validation is successful, the ownership will be delegated to RNO. Then, RNO stores $Attr_u$ into SC_U.
8) SC_{port} sends a transaction to SC_{DNO} to remove the

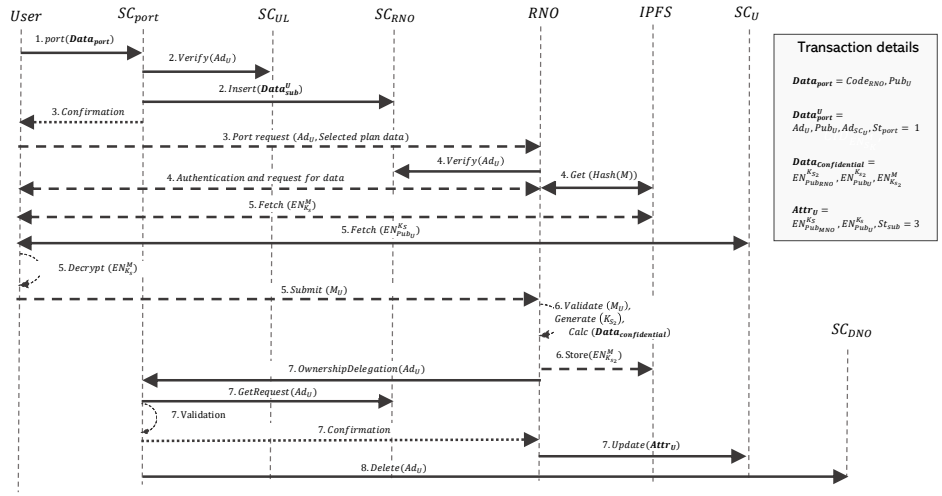


Fig. 2: Mobile number and Profile Porting procedure

$< Ad_U, Number_u >$ from the list of its active users.

D. Termination phase

If the user aims to terminate her subscription with all MNOs it is required to remove user's contract and the MNO's privilege to update her data. The following steps can be followed:

1) u sends the termination request to SC_{port};
2) SC_{port} removes the user from SC_{MNO} and SC_{UL}.
3) SC_{port} destroys SC_U,so, no one can have write/read access to user data unless the previously downloaded versions.

VI. Evaluation

We simulated the whole procedure in a private Ethereum Blockchain. The smart contracts are written in Solidity [19]. The system is implemented in a computer with Intel i7 Dual-core 1.6GHz and 16GB RAM. The Blockchain part runs Ganache-cli 6.12.2 to simulate the consortium Ethereum. We used Solc V.0.8.2 to compile the smart contracts and Web3j v.1.4.1 to interact with them. ' Following, we provided the performance analysis of the proposed method by evaluating the scalability of the system in terms of the increasing number of concurrent requests. Scalability can be defined as changes in latency or throughput when altering a parameter [20]. We assess the latency and throughput [21] as:

$$Latency = \frac{t_f - t_s}{|Tx|}, Throughput = \frac{|Tx|}{t}$$

where Tx is the set of transactions, $|Tx|$ is the number of transactions, t is the total time of execution, t_f is the ending time of execution that we received the transaction receipt for all sent transactions, and t_s is the time in which we started sending the concurrent requests.

The following parameters are adjusted in our assessment :

- Block size (BS): Number of transactions fit into a block.
- Block time (BT): The difficulty of consensus puzzle resulting in the extraction of blocks in predefined time.
- Concurrent requests (C): Number of users sending requests concurrently.

Fig. 3 (a-c) depicts the latency of the system for the aforementioned configurations BS and BT. Based on the definition of scalability, if the latency/throughput stays almost stable regarding the alteration of parameters, we can say that the system is scalable [22], [23]. As shown in the figure, system latency is almost stable for $C \geq 200$. Therefore, we can claim that **the system is scalable** and can maintain adjustable and low latency in a large-scale request environment for user subscription, porting, and termination procedures.

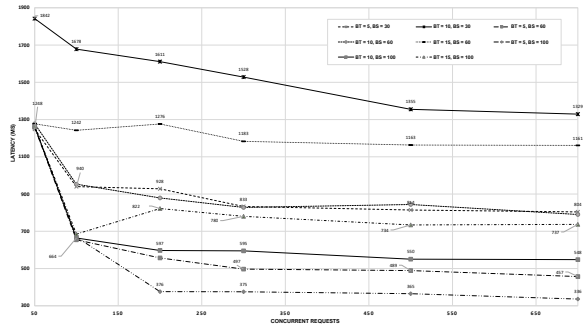
Moreover, Table II provides the throughput of the system in different Blockchain configurations. As shown in the table, increasing the BS and decreasing the BT can positively affect the performance by increasing the overall throughput. For instance, compare the throughput for $BT = 10, BS = 30$ (i.e., the highest complexity of consensus due to minimum trust in the network, and lowest number of transactions fit in each block) and $BT = 5, BS = 100$ (i.e., highest trust and highest number of transactions in each block). Note that, an important issue to select the configuration is the trust level among participants in the network (i.e., decreasing the block time results in an easier consensus puzzle, which can bring the risk of integrity violation in the system).

VII. Discussion and Future direction

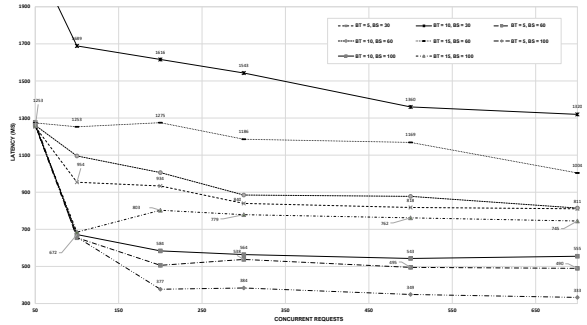
We introduced a novel method for user subscription, user profile, and mobile number portability as a clean-slate solution for beyond 5G cellular networks using Blockchain and smart contracts. This method brings high availability, integrity, scalability, and transparency. Moreover, it decreases the IT complexity on the MNO side, eliminates porting fees, decreases the process latency, and delivers high confidentiality and privacy. We discuss several important aspects of the proposed method and provide some future directions as follows.

TABLE II: System throughput with different parameters. BT (s), Throughput (transaction per second (tps))

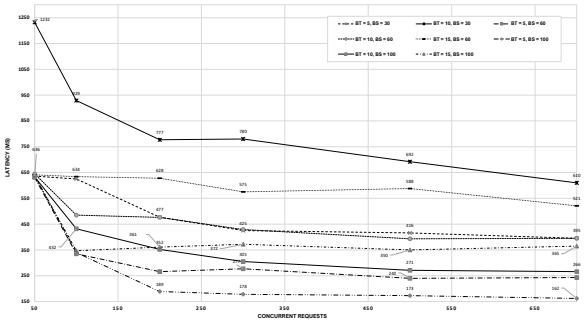
P	***Subscription***								***Porting***								***Termination***							
BS	*30*		*60*			*100*			*30*		*60*			*100*			*30*		*60*			*100*		
BTC	5	10	5	10	15	5	10	15	5	10	5	10	15	5	10	15	5	10	5	10	15	5	10	15
50	0.8	0.54	0.79	0.78	0.79	0.79	0.79	0.79	0.79	0.46	0.78	0.79	0.78	0.79	0.79	0.79	1.57	0.81	1.58	1.55	1.55	1.56	1.57	1.56
100	1.06	0.59	1.52	1.04	0.8	1.49	1.5	1.45	1.04	0.59	1.52	0.91	0.79	1.51	1.48	1.45	1.59	1.07	2.97	2.05	1.57	2.94	2.31	2.87
200	1.07	0.62	1.79	1.13	0.78	2.65	1.67	1.21	1.06	0.61	1.97	0.99	0.78	**2.65**	1.7	1.24	2.09	1.28	3.74	2.1	1.59	**5.28**	2.84	2.76
300	1.2	0.65	2.01	1.2	0.84	2.66	1.67	1.28	1.19	0.64	1.85	1.12	0.84	**2.59**	1.8	1.28	2.34	1.28	3.6	2.32	1.73	**5.6**	3.27	2.68
500	1.22	0.73	2.04	1.18	0.85	2.73	1.81	1.36	1.22	0.73	2.01	1.13	0.85	**2.86**	1.84	1.31	2.4	1.44	4.15	2.53	1.69	**5.75**	3.68	2.85
700	1.24	1.03	2.18	1.26	1.01	2.96	1.82	1.35	1.23	1.01	2.03	1.22	0.99	**2.99**	1.79	1.34	2.52	1.63	4.1	2.52	1.91	**6.13**	3.75	2.73



(a)



(b)



(c)

Fig. 3: System latency with different configuration of BT and BS in several concurrent requests for user subscription (a), porting (b), and subscription termination (c).

A. Discussion on the deployment in real-world scenario

To implement the proposed method in a real-world scenario, some prerequisites must be well-defined, for instance:

- Who are the owners of the Blockchain?
- Where is the geographic area for the authority of the regulatory body?
- Which entities participate in securing the Blockchain?
- How the trust among MNOs and in MNO-UE relationships would be addressed?

The three main actors of the proposed system are users, MNOs, and regulatory bodies. Today, to address the legal issues to allow an enterprise to be able to register as MNO, there is a regulatory body in each country. In our proposed method we also consider this entity. Moreover, In the proposed method the underlying Blockchain can be implemented as a consortium among MNOs and regulatory bodies for each country. So, based on the local regulatory rules in each country, its configuration and requirements can be shaped.

We propose to have the following setting for the implementation of the method in an operational scenario. The Blockchain itself should be a consortium between MNOs and the regulatory body. But some smart contracts such as SC_{MNOL}, SC_{AB} that have the role of regulation, should be deployed and owned by a regulatory body. Moreover, due to the difference among rules in countries, we propose to limit the authority area to the countries. In addition, as the Blockchain is a consortium among MNOs, they are the entities that participate in consensus and keep the latest ledger.

Regarding trust in the system, all MNOs are approved by regulations. So, we can claim that a minimal level of trust exists among MNOs. Moreover, in the MNO-User relationship, all subscription, and the porting procedure is done by smart contracts as a distributed trusted party.

B. Discussion on security concerns and solutions

In this section we provide a brief discussion of the security of the proposed system regarding several threat scenarios:

1) Single point of failure (DoS): Existing a centralized point in the subscription or porting procedure.

Analysis: As shown in Fig. 1 and Fig. 2, subscription and porting procedures are handled by smart contracts. So, to manage the user's requests there are several Blockchain nodes in the system. So, the failure of a single node does not have a significant effect on the functionality of the system. Moreover, data is not registered in a centralized database.

2) Adversary subscription/porting (MitM): Adversary $\mathcal{A}$ aims to eavesdrop user's connection with SC_{port} or SC_{sub} to falsify the user's request or modify her data.

Analysis: As shown in *Step 1*, *Step 4* of Fig. 1, when the user sends the request to the smart contract, she sends

$Hash(nonce)$, in which $nonce$ is a random number selected by the user, and in *Step 4* she sends the $nonce$ itself in off-chain connection. So, if $\mathcal{A}$ succeeds in retrieving $Hash(nonce)$, there is no way for them to find $nonce$. So, the user's request is not recoverable. This problem is resolved by authentication in *Step 4*, of Fig. 2 for porting procedure.

3) User data confidentiality: Adversary $\mathcal{A}$ aims to retrieve user data from a shared database, using the CID in SC_{UE}.

Analysis: To provide data confidentiality, we used a hybrid cryptosystem by which only those who have the user's or dedicated MNO's private key, can have access to data (see Section V-B, *Step 7*). Moreover, when the user switches its MNO, the RNO, generates a new symmetric key, and the ownership of the user's data is delegated to RNO; so, DNO is not able to change user's data or have access on updated data. Based on this fact, we can say that the proposed method provides forward and backward secrecy.

4) Maintainability challenge: The non-database contracts (e.g., SC_{sub} , SC_{port}), needed to be updated.

Analysis: In the proposed system, two contracts SC_{sub} and SC_{port}, are updatable. They do not store hardcoded parameters and use the stored addresses in SC_{AB} to retrieve the other contracts and access them. Based on [16], the maintainability can be addressed by assigning the variables dynamically.

C. Discussion on storage

Due to its append-only nature, storage consumption is one of the main concerns in the application of Blockchain in telecommunication use cases. In this paper, we proposed to use IPFS. This solution can decrease storage consumption because there is no need to record the user's data in the Blockchain; so MNOs are not obliged to record a huge amount of the data of the users of other MNOs in their ledger. They only need to download the part of data that they are allowed.

D. Future directions

Some future directions are proposed as follows:

- Proposing the incentivization method for the participants to make the network more secure.
- To migrate from today's existing centralized procedure to a distributed one, we need to provide an interaction between centralized databases, the proposed Blockchain, and the IPFS. So, for future direction, we can target providing the required data for the porting procedure, not only from the users (for new users) but also from the MNO database.
- A similar procedure in a consortium Blockchain can be utilized for roaming usecases.
- Finally, more assessments can provide stronger proof of the feasibility of the method. Evaluation of the method's performance on different Blockchain implementations such as Hyperledger Fabric, Quorum, etc., and several consensus models can be a practical step.

REFERENCES

[1] S. Bühler and J. Haucap, "Mobile number portability," *Journal of Industry, Competition and Trade*, vol. 4, no. 3, pp. 223–238, 2004.

[2] Y.-B. Lin, I. Chlamtac, and H.-C. Yu, "Mobile number portability," *IEEE network*, vol. 17, no. 5, pp. 8–16, 2003.

[3] Dong Hee Shin, "A study of mobile number portability effects in the united states," *Telematics and Informatics*, vol. 24, no. 1, pp. 1–14, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736585305000626

[4] J. Shah, S. Agarwal, A. Shukla, S. Tanwar, S. Tyagi, and N. Kumar, "Blockchain-based scheme for the mobile number portability," *Journal of Information Security and Applications*, vol. 58, p. 102764, 2021.

[5] 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*, vol. 166, p. 102693, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804520301673

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[7] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[8] N. Szabo, "Secure property titles with owner authority," *Online at http://szabo. best. vwh. net/securetitle. html*, 1998.

[9] D. Krishnaswamy, K. Chauhan, A. Bhatnagar, S. Jha, S. Srivastava, D. Bhamrah, and M. Prasad, "The design of a mobile number portability system on a permissioned private blockchain platform," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 2019, pp. 90–94.

[10] "5g; 5g system; usage of the unified data repository services for subscription data," 3GPP, Tech. Rep., 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/129500_129599/129505/16.03.00_60/ts_129505v160300p.pdf

[11] H. C. Rudolph, A. Kunz, L. L. Iacono, and H. V. Nguyen, "Security challenges of the 3gpp 5g service based architecture," *IEEE Communications Standards Magazine*, vol. 3, no. 1, pp. 60–65, 2019.

[12] F. Ghaffari, E. Bertin, and N. Crespi, "Blockchain-based user profile and mobile number portability for beyond 5g mobile communication networks," in *2022 4th Conference on Blockchain Research Applications for Innovative Networks and Services (BRAINS)*, 2022, pp. 75–78.

[13] "Hyperledger Fabric – Hyperledger Foundation." [Online]. Available: https://www.hyperledger.org/use/fabric

[14] W.-E. Chen and Y.-L. Ciou, "Enum-based number portability for mobile communication networks," *Journal of Internet Technology*, vol. 20, no. 1, pp. 135–145, 2019.

[15] J. Odii, M. Onyesolu, and C. Onukwugha, "A hybrid call routing framework for mobile number portability in nigeria," *Int Res J Comput Sci (IRJCS)*, vol. 3, pp. 9–17, 2014.

[16] J. Chen, X. Xia, D. Lo, J. Grundy, X. Luo, and T. Chen, "Defining smart contract defects on ethereum," *IEEE Transactions on Software Engineering*, 2020.

[17] N. F. Samreen and M. H. Alalfi, "Reentrancy vulnerability identification in ethereum smart contracts," in *2020 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*. IEEE, 2020, pp. 22–29.

[18] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "The keccak sha-3 submission," STMicroelectronics, 2NXP Semiconductors, Tech. Rep., 2011. [Online]. Available: https://keccak.team/index.html

[19] C. Dannen, *Introducing Ethereum and solidity*. Springer, 2017, vol. 1.

[20] M. Schäffer, M. d. Angelo, and G. Salzer, "Performance and scalability of private ethereum blockchains," in *International Conference on Business Process Management*. Springer, 2019, pp. 103–118.

[21] F. Ghaffari, E. Bertin, N. Crespi, S. Behrad, and J. Hatin, "A novel access control method via smart contracts for internet-based service provisioning," *IEEE Access*, vol. 9, pp. 81 253–81 273, 2021.

[22] P. W. Eklund and R. Beck, "Factors that impact blockchain scalability," in *Proceedings of the 11th international conference on management of digital ecosystems*, 2019, pp. 126–133.

[23] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "A survey on the scalability of blockchain systems," *IEEE Network*, vol. 33, no. 5, pp. 166–173, 2019.