# Enhancing Security of Cellular IoT with Identity Federation

Bernardo Santos[1], Bruno Dzogovic[1], Boning Feng[1], Van Thuan Do[2,1], Niels Jacot[2], Thanh Van Do[3,1]

[1] Oslo Metropolitan Univeristy, Pilestredet 35, 0167 Oslo, Norway
[2] Wolffia AS, Haugerudvn. 40, 0673 Oslo, Norway
[3] Telenor ASA. Snarøyveien 30 1331 Fornebu, Norway
{bersan, bruno.dzogovic, boning.feng}@oslomet.no
{vt.do,n.jacot}@wolffia.net
thanh-van.do@telenor.com

**Abstract.** This paper presents a Cellular Identity Federation solution which both strengthens and simplifies the authentication of Internet of Things (IoT) devices and applications by providing single sign-on between the network layer and IoT applications. They are hence relieved of the burden of authentication and identity management, which could be both technically and economically challenging. The paper aims at clarifying how IoT authentication can be skipped without compromising security. The proposed solution is described thoroughly, and the authentication process is depicted step by step. Last but not least is the comprehensive description of the proof-of-concept which shows the feasibility of the Cellular Identity Federation.

**Keywords:** Cellular IoT, NB-IoT, Internet of Things, M2M, 5G, Lower Power wide area network.

## 1 Introduction

Cellular Internet of Things is a rather new concept which emerged due to the demands for mobility and extended coverage of the Internet of Things. Briefly, the concept means that the mobile network is used as network layer for **IoT** applications. In fact, the cellular mobile network is originally intended for mobile phones which are on always online and require higher data rates compared to **IoT** devices which quite often have limited power and only communicate occasionally using a few bytes. Consequently, to use the 3G/4G mobile networks for **IoT** are neither cost efficient nor sustainable. To accommodate **IoT** devices, the mobile community has proposed a few wireless access technologies such as Extended Coverage GSM for Internet of Things (EC-GSM-IoT), Long Term Evolution Machine Type Communications Category M1 (LTE MTC Cat M1, also referred to as LTE-M) and Narrowband IoT (NB-IoT) [1]. Most importantly, the next mobile generation mobile network, **5G** is aiming at supporting a variety of **IoT** applications with diversified requirements in terms of mobility, bandwidth, latency and reliability by making use of the concept of network slicing [2].

This is a post-peer-review, pre-copyedit version of a chapter published in INCoS 2019: Advances in Intelligent Networking and Collaborative Systems pp 257-268, that is part of the Advances in Intelligent Systems and Computing book series (AISC, volume 1035).
The final authenticated version is available online at: https://dx.doi.org/10.1007/978-3-030-29035-1_25

Unfortunately, although better security is offered to **IoT** applications due to the inherent stronger security at the network level compared to other wireless technologies, it is not sufficient and **IoT** applications are left to themselves to ensure appropriate security.

This paper introduces an innovative solution which aims at providing higher level of security at the same time as relieving the administrative burden of the **IoT** applications. The solution makes use of the concept of identity federation which is used in the world wide web to provide single sign-on i.e. the user can just sign in once at one web site and move around to other ones without having to sign in again. In our case the device authentication at the network level is re-used on the **IoT** application to achieve single sign-on and higher level of security. The paper starts with a concise description of related works followed by a short introduction to identity federation. The main objective of the paper is to clarify how identity federation can both enhance and simplify security of cellular IoT applications. We choose to use a rather formal description using Unified Modelling Language (**UML**) [3] diagrams, being the central part of the paper the description of the proposed Cellular Identity Federation solution. The proof of concept implementation at the Secure 5G4IoT Lab[1] is also presented in a comprehensive way. The paper concludes with some suggestions for future works.

## 2 Related Works

As mentioned in [4] there are currently no known activities aiming at simplifying authentication of IoT devices by providing single sign-on with the cellular network authentication. However, there are a few works focusing on extending the usage of the SIM (Subscriber Identity Module) authentication for other applications on smart phones such as Internet browsing, Web mail, Social networks, financial services, etc.

The Generic Bootstrapping Architecture (**GBA**) [5][6] is a standard specified by the 3rd Generation Partnership Project (**3GPP**), which achieves the mentioned objective by introducing in the mobile network a new network element called Bootstrapping Server Function (**BSF**), responsible for retrieving authentication vector from the Home Subscriber Server (**HSS**) and carrying out a mutual authentication of the mobile phone aka User Equipment (**UE**). The **BSF** provides the mobile Internet application aka Network Application Function (**NAF**) with encryption key *Ks_NAF* for the session between the **NAF** and the **UE**. The most serious limitation of this solution lies on the fact that **GBA** requires the presence of the **GBA** client on the mobile phone, which is quite difficult because handset manufacturers do not have the incentive to implement it.

To avoid the need for the BSF the Eureka Mobicome project has been proposing some solutions called Subscriber Identity Module (**SIM**) strong authentication that provides strong authentication from a regular browser on a regular mobile phone carrying a **SIM/USIM** [7][8]. However, these solutions do not address IoT, in which devices are communicating without the intervention of human beings. ETSI did promote the use of **GBA** in their Machine-To-Machine (**M2M**) functional architecture but they focus only

[1] The Secure 5G4IoT lab results from the collaboration between OsloMet, Telenor and Wolffia within the scope of H2020 Concordia project: http://5g4iot.vlab.cs.hioa.no/

on using the strong authentication of the **SIM** card. Indeed, they do not provide a comprehensive and flexible cellular **IoT** identity and access management, which enables both easy inclusion of **IoT** devices and strong authentication and confidentiality.

# 3 Identity Federation

Identity federation provides the means to share or to link users' identities between partners. Such identities are federated between partners when there is an agreement between the providers on a set of identifiers or identity attributes [4]. To share information about a user, partners must be able to identify the user, even though they may use different identifiers for the same user.

When two domains are federated, the user can authenticate to one domain and then access resources in the other domain without having to perform a separate login process.

Single Sign-On (**SSO**) is an important component of identity federation, but it is not the same as identity federation.

Identity federation involves a large set of user-to-user, user-to-application and application-to-application use cases at the browser tier, as well as the service-oriented architecture tier.

## 3.1 Most popular Identity Federation solutions

Although there are multiple Identity Federation solutions only a few popular ones are briefly introduced in this section.

### A. Liberty Alliance

To alleviate the identity burden of both the users and the Service Providers, the Liberty Alliance Project, established in 2001 and overtaken by the Kantara Initiative in 2009 introduced the notion of Federated Network Identity [9]. A new actor called Identity Provider is responsible of authenticating the users and federating user accounts at Service Providers that join the Identity Provider's Circle of Trust.

### B. OAuth2.0

OAuth 2.0 [10] is a scalable delegation protocol that allows a certain application to do certain tasks on behalf of a user. It establishes the concept of an authorization token, providing information on which services on a server the application is authorized to access to, not overriding the access control decision that the server may take.

### C. OpenID Connect

OpenID Connect [11] is a standard built upon OAuth that goes one step further to offer single sign-on and identity provision on the internet. It enables client applications to verify the identity of the user based on the authentication performed by an OpenID Provider, as well as to obtain basic profile information about the user in an interoperable

and REST-like manner. OpenID Connect specifies a RESTful HTTP API, using JSON as a data format. Client apps receive the user's identity encoded in a secure JSON Web Token (**JWT**) called ID token.

Currently OpenID Connect is definitely the most popular Identity Management which is used by several identity provider such as Facebook, Google, Twitter and even mobile operators in their Mobile Connect, a secure log-in solution using mobile phones. It is worth noting that OpenID Connect does not provide identity federation, i.e. federation of the **SP**'s identity with the **IDP**'s identity but promotes the usage of the **IDP**'s identity at the service provider. Further, it is not used in identity management for cellular **IoT**.

# 4 Identity Federation for Cellular IoT

## 4.1 Current IoT systems

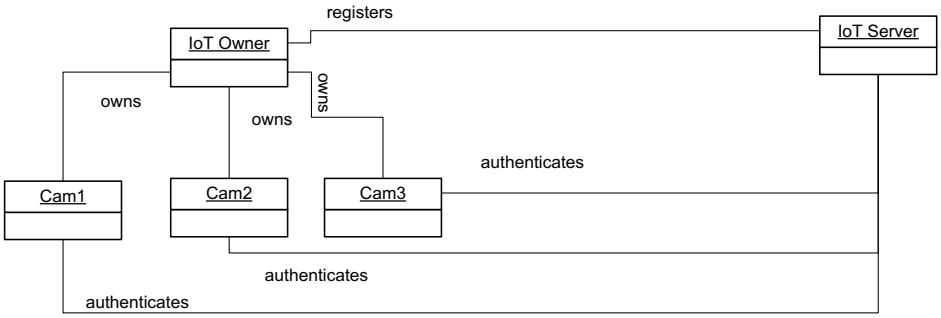


**Fig. 1** A typical IoT System Object Diagram

Let us consider a current typical **IoT** system, for example a home surveillance camera system with 3 **IoT** devices and one **IoT** Server as shown in **Fig. 1**. The **IoT** Server can be in-house or in the cloud operated by the **IoT** manufacturer.

At the first-time installation all the devices are blank, and the owner will have to carry out a configuration which, although simple for technology professionals might be challenging for common people. First, the owner will have to get his/her devices connected to the wireless local area network and thereafter to Internet. If Wireless LAN [12] or Zigbee [13] is the wireless technology used in the local area network, the owner will have to supply a password or a link key which can be used by devices in the authentication towards the wireless home gateway to get granted connection.

After that the network connection is established, the **IoT** owner will have to personalise the **IoT** Server by registering his/her name, user name and a password which is used for authentication and access control for later sign-on sessions. Next, the owner shall register all devices and define passwords both the ones to be used upon access to the devices and the ones to be used by the devices upon authentication to the **IoT** Server. Indeed, in order to get granted access the **IoT** devices will have to be authenticated towards the **IoT** Server.

### 4.2 Current Cellular IoT Systems

Let us now consider a cellular **IoT** system. The cellular mobile network is in this case used to provide Internet connectivity **IoT** devices. The **IoT** owner is also a mobile subscriber and acquires from mobile operator a number of subscriptions corresponding to the number of devices which have to be inserted in the devices as shown in **Fig. 2**. Upon power on, the **IoT** devices carry out authentication towards the mobile Home Subscriber Server (**HSS**) using their **SIM** cards. The **SIM** authentication is a strong authentication in the mobile network, which ensures that the **IoT** device uses a legal International Mobile Subscription Identity (**IMSI**) belonging to the mobile subscriber i.e. the **IoT** owner. Mobile operators in many countries also query and register the International Mobile Equipment Identity or **IMEI**, which uniquely identifies the device, in order to ban stolen devices.

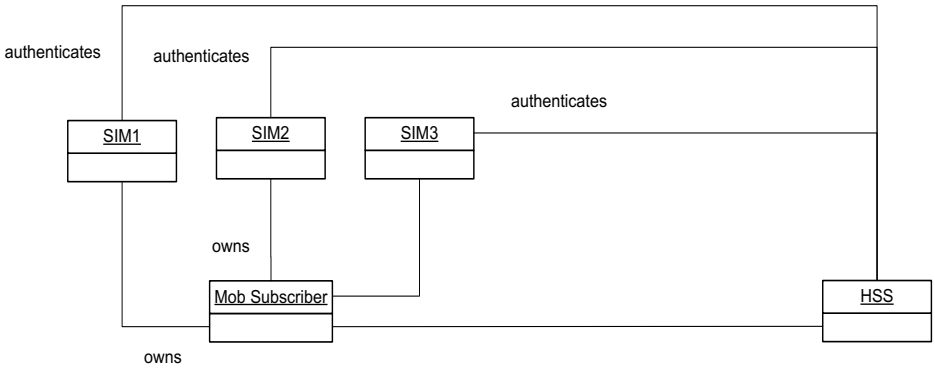


**Fig. 2** Mobile Connection Object Diagram

After successful authentication the **IoT** devices are allowed to get connected to the Internet. However, to communicate with the **IoT** server, the **IoT** devices will have to be personalised such that proper authentication with the **IoT** server can be performed as previously described.

### 4.3 Federation of Mobile and IoT identities

The main finding from the previous sections is that prior to any communication between **IoT** devices and **IoT** server two authentication procedures must be done as follows:

- *Authentication of the mobile devices via the **SIM*** by the mobile network to ensure that both the mobile devices belong to the mobile subscriber to get granted access to the mobile network.
- *Authentication of **IoT** devices by the **IoT** server* to ensure that the **IoT** devices do belong to the **IoT** Owner to get granted access to the **IoT** server.

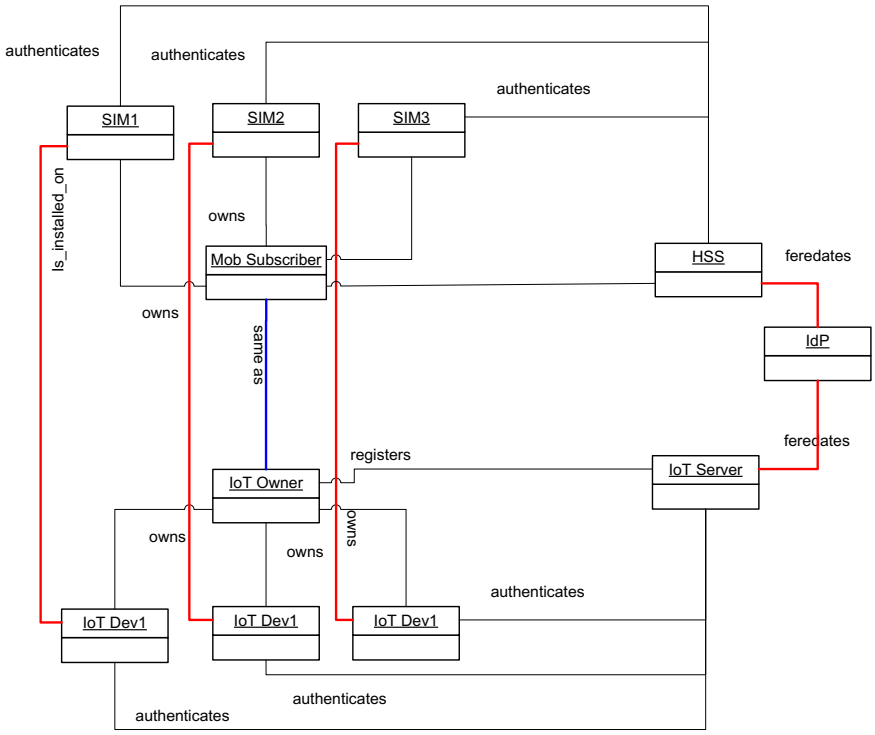


**Fig. 3** Federation between IoT and mobile devices

The two authentications are necessary because the **IoT** system and mobile network are completely separate systems that do not know anything about each other.

To remove the later authentication on the **IoT** layer, the concept of federation of identities can come to help. Indeed, as shown in **Fig. 3**, federation provides the means to specify that the **Mobile_Subscriber** is **same as** the **IoT_Owner** who knows that the **SIM$_x$** is installed on **IoT_Dev$_x$**. Indeed, when the **SIM$_x$** is successfully authenticated by the **HSS**, if the **IoT_Dev$_x$** can prove that it carries this **SIM$_x$** then it is the **IoT_Dev$_x$** belonging to the **IoT_Owner** and can get granted access to the **IoT_Server**. This proof can be in a form of a unique one-time token generated by the HSS and passed through the mobile network to the **IoT_Dev$_x$** which presents to the **IoT_Server**.

In order to let the **HSS** and the **IoT_Server** know that the **SIM$_x$** containing an **IMSI$_x$** is installed on the **IoT_Dev$_x$**, an Identity Federation table similar to the one shown in **Table 1** has to be installed at both the **IoT_Server** and **HSS**.

**Table 1.** Identity Federation Table

| ***Device_ID*** | ***IMSI*** |
|---|---|
| *Camera_Front_Door* | $IMSI_1$ |
| *Camera_Backyard* | $IMSI_2$ |
| *Camera_Garage* | $IMSI_3$ |
| *Smoke_Detector* | $IMSI_4$ |
| *Contact_Front_Door* | $IMSI_5$ |
| *Contact_Balcony_Door* | $IMSI_6$ |
| *Contact_Back_Door* | $IMSI_7$ |

Further, it is necessary to define and implement an authentication protocol between the **IoT_Server** and **HSS**, which enables the **IoT_Server** to ask the **HSS** for authorization upon login request from an **IoT** device. As to make it both simpler and more standardized, we propose to use OpenID Connect and hence to introduce an additional functional entity called **Identity Provider** or **OpenID Provider**, which acts as a middleman managing the identity federation table on behalf of both the **IoT_Server** and **HSS**. In addition, it will on behalf of the **IoT_Server** perform authorization of the **IoT** devices.

Before continuing with the explanation of the authentication procedure let us now present the overall architecture of the solution.

### 4.4 The proposed Cellular Identity Federation solution

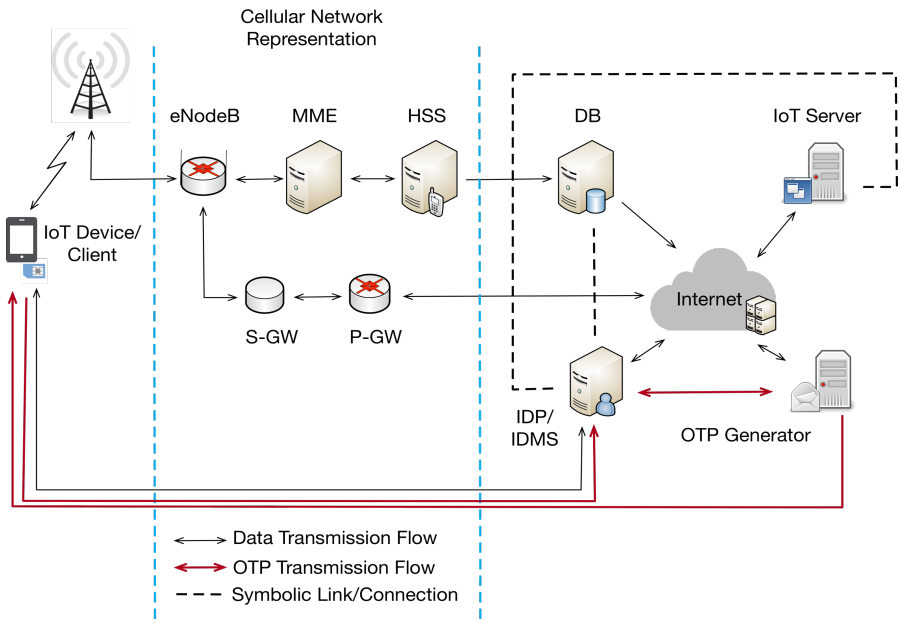


**Fig. 4** Overall architecture of the Cellular Identity Federation solution

As shown in **Fig. 4**, in the proposed Cellular Identity Federation solution, the current 4G mobile network consisting of standardized network elements such as **eNodeB**s (base stations), **MME** (Mobility Management Entity), **HSS** (Home Subscriber Server) on the control plane and **S-GW**(Serving Gateway), **P-GW** (Packet Data Network Gateway) on the user plane) is now interfaced with the **IoT** Server such that information about successfully authenticated **IMSI**s can be shared with the **IoT** Server. Three new entities are introduced in the solution as follows:

- **The Identity Provider (IDP/IDMS):** makes use of the authentication information from the HSS to carry out authentication of the IoT devices on behalf of the **IoT_server**. In our solution, OpenID Connect is selected and used because it is by far the most popular and simple standard.
- **The Secure Database (DB):** To keep the security protection of the **HSS** at the same level as before, instead of introducing an IP interface on the **HSS** and allowing direct interactions with it from the IDP, we introduce a partially mirrored database, which gets transferred from the HSS only relevant parameters about successfully authenticated devices such as **IMSI**, **IMEI** (International Mobile Equipment Identity), **MSISDN** (Mobile Subscriber ISDN Number), **PDP** (Packet Data Protocol) type, **PDP** address (IP address), **APN** (Access Point Name), etc. These parameters will again be sent to the **IDP** for storage and use in the authentication of the IoT devices.
- **The OTP Generator:** To ensure that an **IoT_Dev$_x$** is really the one belonging to the **IoT_Owner,** it has to prove that it actually is the device carrying a **SIM$_x$** that has been successfully authenticated by the **HSS**. For that, it is not sufficient that it presents its IMSI or IMEI to the **IoT_Server** upon login because both **IMSI** and **IMEI** can be easily sniffed for replay. A more secure token is required. Upon receipt of authentication parameters of an **IoT_Dev$_x$** from the **DB,** the **IDP** will request the **OTP Generator** to produce a one-time password and send it to the **IoT_Dev$_x$** using the **PDP** address such that it can use it to login onto the **IoT_Server**.

### 4.5 The authentication process

To clarify how an IoT device is authenticated let us now consider an **IoT_Dev$_1$** hosting a **SIM$_1$**. The authentication process is as follows:

1. At power on, the **SIM$_1$** on **IoT_Dev$_1$** participates to the authentication of User Equipment
2. Upon successful authentication, the **HSS** stores the state of **IoT_Dev$_1$** as registered and notifies the Replica Database that initiates the duplication and transfer of the parameters of the **IoT_Dev$_1$** to the **IDP**.
3. The **IDP** stores the data and send a request for the generation of a one-time password to be sent to the **PDP** address of the **IoT_Dev$_1$**.
4. The **OTP Generator** generates an **OTP** and sends it to both the **IDP** and **IoT_Dev$_1$**.
5. The **IoT_Client** on the **IoT_Dev$_1$** fetches the **OTP** and presents it to the **IoT_Server** upon login.
6. The **IoT_Server** redirects the **IoT_Client** to the **IDP**.

7. The **IDP** compares the presented **OTP** and if it matches with the stored one the **IoT_Client** is considered authenticated and directed back to the **IoT_Server**, which grants access to **IoT_Client.**

The authentication process is hence completed without active participation of the **IoT_Server** which does not have to administrate passwords of its **IoT** clients while strong security is still ensured.

## 5 The Cellular Identity Federation Proof-of-Concept

To validate the proposed Cellular Identity Federation solution a proof-of-concept is built at the Secure 5G4IoT Lab at Oslo Metropolitan University consisting of a 4G/5G mobile network extended with 3 Identity Management entities and 1 IoT entity as shown in
**Fig. *5***.

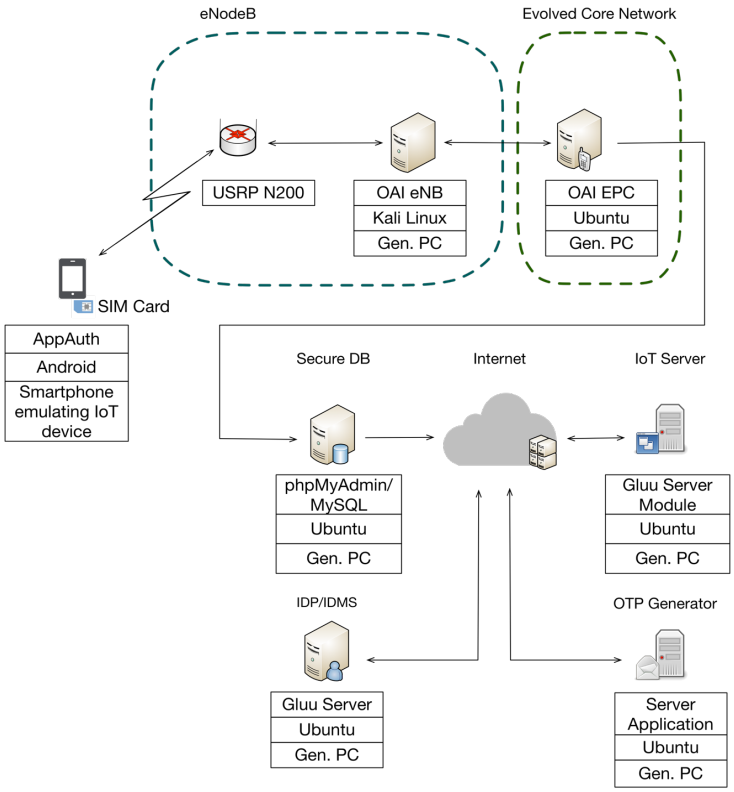


**Fig. 5** The Cellular Identity Federation PoC at the Secure 5G4IoT Lab

### 5.1 4G/5G Mobile Network

To realize an early implementation of a **5G** mobile network OpenAirInterface [14], an open source communication software elaborated by EURECOM are first installed in generic computers and then later virtualized on the OsloMet to achieve a softwarized **5G** mobile network.

The 4G LTE base station **eNodeB** is realized by:

- A generic PC running Kali Linux and OpenAirInterface **eNB** connected to
  A USRP (Universal Software Radio Peripheral) N200, which is a software-defined radio designed and commercialized by Ettus Research [15].

The whole **Evolved Packet Core** including **HSS** is realized by:

- A generic PC running Ubuntu and OpenAirInterface, which includes an HSS.

### 5.2 IoT Server

The **IoT Server** is realized by:

- A generic PC running Ubuntu and Gluu Server 3.1.5 and also a lightweight M2M server open source using Eclipse Leshan [16].

### 5.3 Identity Management entities

The **IDP** is realized by:

- A generic PC running Ubuntu and Gluu Server 3.1.5 [17], which is an open source identity provider server software compliant with OpenID Connect.
- Gluu Server is adopted because of its fast deployment and flexibility and also because it can act not only as an IDP but also as an IDMS which permits us to review and control all the identities to be issued and to handle basic profiling options, such as user grouping and role assignment.

The **Secure Database** is implemented by:

- A generic PC running Ubuntu and MySQL, an open source relational database management system (RDBMS).
- An extension is implemented allowing the database to establish a read-only SQL connection to the ***HSS*** database to extract the parameters of successfully authenticated devices which are necessary for identification and authentication of IoT clients.
- Another functionality is added to enable the Secure Database to set up a RESTful request by creating a JSON object for each device identity and then to send it the ***IDP*** for the establishment or updating of device identities.

The **OTP Generator** is implemented by:

- A generic PC running Ubuntu and an application which is able to generate a one-time password for each successfully authenticated device upon request from the ***IDP***. This one-time password is then sent both the ***IDP*** and the **IoT** Device using its PDP address. This one-time password will be presented by the **IoT** Device to the **IoT** Server at login.

### 5.4 IoT Devices

The **IoT Devices** are emulated by:

- Smartphone devices with Android OS equipped with the *AppAuth* [18] Software Development Kit (**SDK**) in order to interact with the ***IDP*** and the network.

The proof-of-concept has been tested with the focus on flexibility and usability for **IoT** applications. The ability of registering and removing new **IoT** owners and their devices at the ***IDP*** has been verified.

## 6 Conclusions

In this paper we have introduced a Cellular Identity Federation solution which aims at simplifying the authentication of **IoT** devices in **IoT** applications which could be both technically and economically challenging for the users. By removing the need for **IoT** devices authentication, the proposed solution will contribute to the success of the coming **5G** mobile network.

Although the feasibility of the solution has been verified, the performed tests are still limited, and more diversified tests are needed in order to cover all the relevant scenarios. Most straightforward, the tests with real **IoT** devices such as security sensors, smart locks, e-health, etc. will have to replace the ones with emulated smartphones. This is a more demanding work because a client will have to be implemented for these **IoT** devices. A reasonable approach is to design and implement an open and generic client which can be customised and installed in multiple heterogeneous devices.

In the current solution the **IoT_Client** uses a one-time password to identify and authenticate itself towards the **IoT_Server**. Although it is functional, the solution is not optimal since it requires an **OTP** generator. A better solution would be to establish communications between the **IoT_Client** and its hosted **SIM** such that the **IoT_Client** can query the **TMSI** (Temporary Mobile Subscriber Identity) and use it instead of the one-time password. **TMSI** is temporarily assigned to the device during location registration and can be reallocated at certain intervals determined by the mobile operator. The usage of **TMSI** will prevent sniffing and replay.

Last but not least is to carry out a trial in real **5G** mobile network environment such as the **5G** VINNI Norway **5G** Facility Site at Kongsberg in Norway which will be available by late 2019. Such a trial with real users will make it possible to collect feedbacks that again can be used to improve the proposed solution.

## References


1. GSMA: 3GPP Low Power Wide Area Technologies White Paper, 1 Sept 2016, Svetlana Grant
2. Dzogovic. B., Do, V. Thuan, Feng, B. & Do, Thanh van: Building virtualized 5G networks using open source software, DOI: 10.1109/ISCAIE.2018.8405499, Proceedings of 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE 2018),
3. Grady Booch, James Rumbaugh, Ivar Jacobson: Unified Modeling Language User Guide, The, 2nd Edition, May 19, 2005, Addison-Wesley Professional, ISBN-13: 978-0-321-26797-9
4. Santos,B. , Do, Van Thuan, Feng, Boning & Do, Thanh van: Identity Federation for Cellular Internet of Things, Proceedings of 2018 7th International Conference on Software and Computer Applications (ICSCA 2018), ACM ISBN: 978-1-4503-5414-1
5. 3rd Generation Partnership Project: 3GPP TS 33.220 V8.2.0 (2007-12) Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA) Generic bootstrapping architecture (Release 8)
6. Timo Olkkonen: Generic Authentication Architecture, Helsinki University of Technology -http://www.tml.tkk.fi/Publications/C/22/papers/Olkkonen_final.pdf
7. Do Van Thanh, Tore Jønvik, Do Van Thuan & Ivar Jørstad: Enhancing Internet service security using GSM SIM authentication, Proceedings of the IEEE Globecom2006 conference – ISBN 1-4244-0357-X – San Francisco, USA, Nov 27 - Dec 1, 2006
8. Do van Thanh, Tore Jønvik, Boning Feng, Do van Thuan & Ivar Jørstad: Simple Strong Authentication for Internet Applications using mobile phones, Proceedings of IEEE Global Communications Conference (IEEE GLOBECOM 2008), ISBN 978-1-4244-2324-8, New Orleans, LA, USA, Nov 30 – Dec 4, 2008
9. Liberty Alliance: ID-FF Architecture Overview – vers. 1.2-errata-v1.0
10. IETF Request for Comments: 6749: The OAuth 2.0 Authorization Framework, October 2012
11. OpenID Connect: http://openid.net/connect/, last accessed October 2018
12. IEEE: 802.11 Wireless Local Area Networks http://www.ieee802.org/11/, last accessed October 2018
13. Zigbee Alliance: https://www.zigbee.org, last accessed October 2018
14. The OpenAirInterfaceTM Software Alliance (OSA) http://www.openairinterface.org/, last accessed October 2018
15. Ettus Research, Inc., USRP N200 (Online] Available at: https://www.ettus.com/product/details/UN200-KIT (Accessed November 2017]
16. Leshan: https://eclipse.org/leshan/, last accessed October 2018
17. Gluu: https://www.gluu.org/, last accessed April 2019
18. AppAuth: https://appauth.io/, last accessed April 2019


## Acknowledgement

This paper is a result of the H2020 Concordia project (https://www.concordia-h2020.eu) which has received funding from the EU H2020 programme under grant agreement No 830927. The CONCORDIA consortium includes 23 partners from industry and other organizations such as Telenor, Telefonica, Telecom Italia, Ericsson, Siemens, BMW, Airbus, etc. and 23 partners from academia such as CODE, university of Twente, OsloMet, etc.