# IoTRoam: design and implementation of a federated IoT roaming infrastructure using LoRaWAN

Sandoche Balakrichenan, Antoine Bernard, Michel Marot, Benoit Ampeau

# IoTRoam - Design and Implementation of a Federated IoT Roaming Infrastructure using LoRaWAN

Sandoche Balakrichenan[1], Antoine Bernard[1,2], Michel Marot[2], and Benoit Ampeau[1]

[1]AFNIC
*firstname.surname@afnic.fr*
[2]Samovar, Télécom SudParis, Institut Polytechnique de Paris
*firstname.surname@telecom-sudparis.eu*

***Abstract*—There are no standardised interconnection procedures for interconnecting different IoT networks for roaming as far as we know. Currently, roaming infrastructures are siloed to their respective IoT technologies or applications. Roaming mechanisms used in Cellular, Wi-Fi cannot be applied directly in IoT for various reasons. For example, a human is not behind the IoT End device for decision making, multi-stakeholders involving multi-layer interconnection agreements and heterogeneity in IoT identification, technologies and applications. The focus of IoTRoam is to have an operational roaming set up that can be easily deployed today, scale seamlessly, integrated to existing IoT infrastructures with minimum adaptions, requires minimum initial configuration and work on a global basis. Based on the technical community's feedback, we started focusing on a specific IoT connectivity technology called LoRaWAN and extend the concept to other IoT technologies. As a Proof-of-Concept, we have designed, implemented and tested a roaming platform based on LoRaWAN. For implementing the Proof-of-Concept, we used well-established infrastructures such as PKI and the DNS. The IoTRoam experience has helped us update the LoRaWAN backend specifications and propose solutions for operational issues. We also evaluate whether the proposed mechanisms satisfy the constrained IoT requirements. We are in touch with several academic institutions and the LoRa Alliance community, who have expressed interest in the federated platform's interoperability testing.**

***Index Terms*—IoT; LoRaWAN,; DNS; PKI; AAA**

## I. Introduction

***Roaming*** is the capability of an End-Device (ED) to transmit and receive data on a ***visited network***. A classic example is Cellular roaming, wherein a subscriber can use the visited network infrastructure (such as the radio spectrum, base station) - when the subscriber's ***home network*** doesn't have coverage.

Roaming between the home and the visited network needs to take into account three broad criteria: technical, economic and regulatory. In order to allow the ED to use the visited network, *technically*, there should be an interconnection between the home and the visited network either directly or via a third party. *Economically*, the interconnections between different network operators are governed by agreements, which define the terms of interconnection. There should be an external body (such as Governments) which *regulates* these agreements, so that the terms and conditions are beneficial both to subscribers and network operators. This article will focus only on ***interconnection*** from the technical perspective.

Interconnection between different Internet of Things (IoT) networks become possible either by establishing *One-to-One* interconnection or using a *hub* model. One-to-One interconnection is like the Internet peering model wherein two IoT networks interconnect with each other. Hub is similar to an Internet transit model, wherein by establishing an interconnection with a single hub it is possible to exchange traffic with the networks connected to that hub as well as with the networks connected to its peers. The One-to-One or hub interconnection deployments have been done following out-of-band mechanisms, and as to our knowledge, there are no ***standardised*** interconnection procedures for interconnecting different IoT networks for roaming.

Both the hub and the One-to-One interconnection models create a *Walled garden* [1], wherein the ED in the coverage area of a visited network can connect to its service only if there is a *prior* interconnection agreement between the home and the visited network. We have designed and tested a ***federated platform***, wherein it is possible for an IoT ED to move to a visited network and still access its required service without the need of having a prior interconnection agreement.

The IoT connectivity technology deployed in this federated platform is Long Range Wide Area Network (LoRaWAN) [2], which falls under the Low Power Wide Area Technologies (LPWAN) [3] category. Since economic factors are not taken into consideration, we intend to test the federated platform with academic Institutions as an open and free interconnected IoT network. We have started roaming testing using this platform with research and academic institutions.

The platform is labelled as ***IoTRoam*** and our objective is to achieve, with IoT connectivity technologies the same interconnection functionalities that eduroam [4] proposes with Wi-Fi connection. In eduroam, an end-user who has credentials

***Note:*** All identifiers (NetID, JoinEUI, IP address etc.) used in this article are intended to be fictional

to connect to a particular eduroam Wi-Fi network for Internet service can access the Internet from any other eduroam network seamlessly. *Primary* requirement is that an ED having credentials to connect to a particular IoTRoam network should be able to access its service seamlessly (with minimum prior configuration requirements) in the event of finding itself in the coverage area of visited network. *Second* requirement is that the proposed federated model should be operationally feasible. The vision is to start with LoRaWAN and ***extend*** the design to be applicable to other IoT networks.

IoT Roaming is different from Cellular and Wi-Fi roaming and thus posing new ***challenges***. For eample, in Cellular roaming, interconnection is usually geographically defined and shared between different public Mobile Network Operators (MNOs) (usually three to four MNOs in a Country). In the IoT scenario, there are public, private, community based network operators and there could be thousands of private network operators within a Country. Adding to the complexity, IoT roaming mostly needs to have multi-layer interconnection agreements. For example; the authentication and authorization of the ED to the roaming network maybe governed by a security solution provider or by the ED manufacturer rather than the network operator.

The IoTRoam experience brings the following ***contributions***:

- A model that seamlessly interconnect the multi-stakeholders, IoT connectivity technologies using standards and infrastructure currently employed in the Internet
- A model that is operationally feasible and can be deployed with minimum prior configuration requirements
- A Proof of Concept (PoC) [5] in place which is open, can be accessed freely and used by the community. In this process we have also contributed to software development [6]

Rest of this article is structured as follows: Based on the literature, Section II identifies the requirements for an interconnected federated IoT architecture and the possible existing solutions that could be used. In Section III, we argument our choice of selecting LoRaWAN followed by a brief background of LoRaWAN in interconnection scope in Section IV. Section V describes how DNS and PKI are deployed in the federated platform, in section VI we evaluate whether the proposed mechanisms satisfy LoRaWAN constraints and section VII briefly describe the contributions.

## II. Related Work

Since the focus of the article is on interconnecting IoT networks, our initial approach was to reuse existing standardised Interconnection models for roaming. The hurdle is that there is no single Standards Developing Organization (SDO) which has the sole responsibility of making IoT Standards. As to our knowledge, there is no standardisation work on interconnecting different IoT networks for roaming.

Recent study [7] proposed a mechanism to enable roaming between LoRaWAN and 5G network. This proposal includes a handover roaming mechanism for LoRaWAN that relies on 5G for the authenticating of the ED to the network. As to our knowledge by being part of the LoRa community, tests have been done for passive roaming, but handover roaming scenarios are still in the pipeline. Also, in the article, the ED is intended to be equipped with both 5G and LoRa interface. Adding 5G interface to the ED will considerably reduce the battery life, which is a disadvantage when one considers operational feasibility.

Keeping our focus on building an operationally feasible interconnected IoT platform, we turned to the WBA Open Roaming [8] initiative for guidelines. Concerning the basic requirements for designing an architecture for an Open, seamless IoT interconnection, a WBA study [9] outlined a minimum set of requirements to consider. The considered solutions should be:

- Able to scale to support a potentially massive number of IoT EDs roaming on visited networks
- Able to overcome interoperability challenges that can occur between IoT technologies
- Able to ensure a secure and scalable Authentication, Authorization and Accounting (AAA) framework that is compatible with different IoT technologies/applications

The State of the art presented in this section focuses on the three points enumerated above.

### A. A Scalable identity provisioning and resolution infrastructure for interconnection

***Identification*** plays a vital role in interconnecting heterogeneous IoT networks. When an IoT ED is roaming, the visited network should retrieve its identifier and bootstrap the interconnection process to access the service related to the identifier in the Internet. If the obtained identifier is not unique, then there is a possibility of collision. Hence the identifier for an IoT ED should be *unique*.

Identifiers used in IoT includes heterogeneous identifiers encoded in different standardised naming formats such as IPv6, EUI-48, EUI-64, EPC, DOI, RF-ID, non-standardised identifiers for a specific industry such as Apple Unique Device Identifier (UDID) and user-generated identifiers.

One possible way to solve the issue of heterogeneity in identifiers is for all IoT stakeholders to move to a globally unique identifier, such as the IPv6 address (since it has a large addressing space and capable of allocating a unique identifier for every IoT ED on the planet). In reality, this solution of migrating to one globally unique identifier for the whole IoT industry is impossible. The reason being cost and the technical complexities in migrating the IoT infrastructure with their existing identifiers to IPv6. A feasible alternative would be to let the different sectors in the IoT use their existing identifiers but to use a ***mapping*** service which can map the identifier of an IoT ED to its network.

The well-known mapping service on the Internet is the Domain Name System (DNS) , basically conceived to translate human-friendly computer host-names on a TCP/IP network into their corresponding "machine-friendly" IP addresses. Our

previous work [10] provided arguments based on existing standards and deployments to illustrate why DNS should be the naming (i.e. mapping) service for IoT.

Examples of leveraging DNS for mapping identifiers other than domain names include: Electronic NUmber Mapping (ENUM [11]) for telephone numbers, and for IoT; there exists already standards such as Object Naming Service (ONS) [12] for the consumer industry, Object Resolution System (ORS) standardised jointly by the ITU-T and ISO/IEC and the Handle system standardised by the ISO which uses the DNS infrastructure to resolve the IoT identifiers to its related service in the Internet.

IoT identifiers are structured into two different categories: *Hierarchical*, and *Flat*. The hierarchical identifiers are allocated hierarchically, control is decentralized, and the nature of allocation makes sure that there is no duplicity. These features are similar to the domain name allocation and management, and thus these type of identifiers could naturally leverage the DNS infrastructure for allocation and resolution.

An example of a hierarchical identifier used in the supply chain industry is the Electronic Product Code (EPC). The barcodes attached to consumer prodcuts follow the EPC naming convention, which can be hierarchically partitioned into Country, Organisation and product level.

An example of a flat identifier is Unique Device Identifier (UDID), a unique serial number assigned to each Apple manufactured device. Apple uses UDID to track and record Apple manufactured devices, and it does not have an hierarchical allocation as that of EPC. The UDID is unique within the Apple UDID namespace. It is a 40-character alphanumeric string of code as follows:

2b6f0cc904d137be2e1730235f5664094b831186

Provisioning both identifiers types; EPC and UDID could be included into the Internet via the DNS namespace as shown in the Figure 1. Then it is up to the client libraries to make the conversion and add the specific sub-domain suffix (apple for UDID and gs1 for EPC) to the identifiers. Once the identifier is converted to a FQDN as follows :

2b6f0cc904d137be2e1730235f5664094b831186.udid.apple.
3.1.3.1.6.2.3.3.9.3.4.0.3.gs1. (supposing that there is a TLD called 'gs1')

they will follow the normal DNS resolution to resolve their associated resource/ED/metadata.

When an IoT ED is in the coverage area of a visited network, the visited network could use the DNS infrastructure to identify the home network of the ED by converting the identifier of the ED to a domain name. Based on the different standards enumerated earlier, our hypothesis is that DNS is the only infrastructure that could scale to billions of EDs in the context of IoT interconnection, similar to how it has withstood the meteoric rise from hundreds of domains at the beginning of the Internet to billions currently [13]. Thus, we propose using DNS infrastructure as a scalable solution to satisfy the *first* requirement outlined by WBA, described in section II.

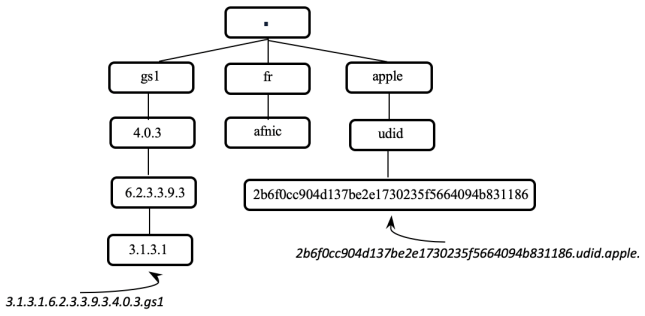



Figure 1: Provisioning IoT identifiers in the Internet domain namespace

### B. Overcome interoperability challenges

Interoperability is the ability of a system to work with or use the components of another system. As mentioned in this article [14], there are four levels of interoperability; we will narrow our focus on an ***Organisational Interoperability*** approach to overcome interoperability challenges. Organisational interoperability is the ability of organisations to effectively communicate and transfer information across different information systems, infrastructures spanned over different geographic regions and cultures [15]. The EU project - symbIoTe [16], proposes a finer granularity of organisational interoperability, enabling IoT platforms to collaborate by forming *federations*. Thus supporting roaming, where the EDs' could find their core services while in the coverage area of the visited network with the help of their unique identity.

IoTRoam approach is to build on a federated open interconnection model. IoTRoam will be a federation of different IoT platforms, and each organisation that is part of the federation needs to share their IoT resources freely. An ED that is part of the federation and is roaming, should be able to identify its home network based on DNS resolution and use its core services via the freely shared resource of the visited network. With a combination of federation model, DNS provisioning and resolution, it is possible to interconnect IoT infrastructures using different IoT technologies, thus satisfying the *second requirement* outlined by WBA described in section II.

### C. Scalable Authentication and Authorization (AA) framework compatible with different IoT technologies

The proposed IoTRoam federated platform should ***control the terms*** under which a roaming ED is allowed to securely use the resources in the environments operated by the visited network. AAA model is the one that is used to control the terms of mediating in traditional network access and is also an optimal model for IoT [17].

AAA functionalities are usually consolidated in a single centralized database [18]. The centralized AAA framework has its advantages but also has significant disadvantages such as creating a single point of failure, and the issue of consolidating all user information into one database, goes against GDPR

regulation. Blockchain using distributed ledger has been experimented and deployed[19][20] to accomplish a scalable decentralized AAA framework. But the blockchain model has several drawbacks as a feasible operational model [21] in an open/global scenario.

eduroam uses the ***distributed*** Public Key Infrastructure (PKI) based on X.509 digital certificates for AA. The PKI model has been tested both on the Internet and in IoT for dynamicity and scalability. The primary issue with the X.509 digital certificates is its size and is not compatible with resource-constrained IoT networks. Since our focus is on organisational interoperability between the networks in the federation operating in the IP space, we plan to employ PKI for AA.

eduroam uses a combination of IEEE 802.1X, the Extensible Authentication Protocol (EAP) and RADIUS to provide [22] AA of the Wi-Fi ED to the network. The trust fabric in eduroam is a Pre-Shared Secret Key (PSK) between the RADIUS servers (organisational, national, global) based on the DNS hierarchy. The organisational RADIUS servers agree on a shared secret with a national server, which in turn agrees on a shared secret with the root (i.e. global) server. The RADIUS hierarchy forwards user credentials securely to the users' home institutions, where they are verified and validated.

Such a trust fabric, wherein there is a PSK to be shared hierarchically is a ***hindrance*** for the federated model that we envision for IoTRoam. The reason being; different IoT networks uses different mechanisms to share the PSK between the ED and the AA servers in the Internet to securely onboard the ED in the home network. Forcing them to transition to a newly proposed PSK mechanism is not operationally possible, since there are multiple stakeholders involved. For securely onboarding the ED, we will use the existing mechanisms used by the respective IoT technologies and propose a global PKI mechanism that has been tested in the Internet to mutually authenticate different servers (in IP space) in the federation involved in ED onboarding.

Using a combination of applying existing PSK mechanism used by the respective IoT technologies/industries/applications and a global PKI mechanism proposes a design that satisfies the *third* requirement outlined by WBA described in section II, for a scalable (even when there are millions of networks interconnected in the federation) AA framework applicable for all IoT technologies/applications.

## III. Choosing the IoT connectivity technology

We started with an idea of setting up an open roaming federated platform integrating all IoT connectivity technologies. We debated this idea by querying the IETF IoT onboarding mailing list [23]. This discussion made us realize that we should focus on a specific IoT connectivity technology and if possible, ***extend*** the concept to other IoT technologies.

IoT connectivity technologies could be classified broadly into three categories [24] : Short Range (Bluetooth, Zigbee, Zwave), Medium range (Wi-Fi) and Long range (LoRa, NB-IoT, Wi-Sun, Sigfox). We eliminated from our focus, technologies that are not in a position to support roaming such as Short Range technologies and also closed networks such as Sigfox, which does not require the roaming feature due to its vertical ecosystem.

Narrowing our focus based on requirements, we short-listed ***LoRaWAN***. Compared to other IoT connectivity technologies, LoRaWAN ecosystem provides freedom to its stakeholders to make their own choices in terms of choosing the ED manufacturers, Network Service Providers and Application Service Providers. Since the radio connectivity uses a license-free spectrum, the freedom of choice in LoRaWAN extends to deployment options also. There are ***public*** LoRaWAN having nationwide coverage; ***private*** LoRaWAN focusing on specific use-cases and ***community*** networks that can be used for free by end-users.

Concerning the requirements in section II-A, LoRaWAN has already included in its backend specification [2], the possibility of using DNS for ED onboarding to the network and roaming. It is possible to have organisational interoperability set up with LoRaWAN, as discussed in section II-B. Mutual AA between the different entities in the IP end of the LoRaWAN star topology is left to the implementor's choice. Hence, we decided to implement in LoRaWAN a AA model as discussed in section II-C.

## IV. LoRaWAN background in interconnection scope

LoRaWAN is an asymmetric protocol, with a star topology as shown in (Figure 2). Data transmitted by the ED is received by a Radio Gateway (RG), which relays it to a Network Server (NS). The NS decides on further processing the incoming data based on the ED's *unique* identifier (DevEUI). The NS has multiple responsibilities like forwarding the uplink from the ED to the Application Server (AS), queuing the downlink from the AS to the ED, forwarding the ED onboarding request to the appropriate AA servers, named as Join Server (JS) in LoRaWAN terminology. The AS handles all the application-layer payloads of the associated EDs' and provides application-level service to the end-user. While the ED is connected to the RG via ***LoRa modulated RF messages***, the connection between the RG, the NS and the AS is done through ***IP traffic*** and can be backhauled via Wi-Fi, hardwired Ethernet or Cellular connection.

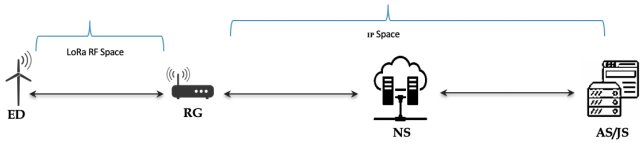



Figure 2: Basic LoRaWAN set up

The JS acting as the AA server *control the terms* on how the ED gets ***activated*** (i.e. onboarded) to a selected LoRaWAN. There are two types of ED activation: Over the Air Activation (***OTAA***) and Activation by Personalization (ABP). With ABP, the ED is directly connected to a LoRaWAN by hardcoding the

cryptographic keys and other parameters required for secured communication. With OTAA, the parameters necessary to create a secured session between the ED and the servers in the Internet are dynamically created for a session. This is similar to TLS handshake used in HTTPS connection. OTAA is preferred over ABP since it is dynamic, decouples the ED and the backend infrastructure and doesn't need session keys to be harcoded. This article will focus only on the OTAA process.

In the *home network scenario*, the ED performs a Join procedure with the JS during OTAA by sending the *Join Request (JR)*. The JR payload contains the ED's unique identifier (*DevEUI*), the cryptographic AES-128 root keys: *NwkKey, AppKey* and *JoinEUI* (unique identifier pointing to the JS).

The JS associated to the ED also has prior information such as the ED's DevEUI, the cryptographic keys: NwkKey and AppKey required for generating ***session*** keys to secure the communication between the ED and the NS and AS. These are the pre-shared information between the ED and JS (the AA server) in the Internet, which we proposed not to modify (as described in section II-C)

Once the JS authenticates the ED, it responds with a JoinAns message to the NS. The JoinAns message contains different session keys derived from the root keys: one set of cryptographic keys for securing the *ED <-> NS* interface and another for securing the communication between the *ED <-> AS* interface, for a particular session.

In the *visite network scenario*, a *non-activated* ED should first activate itself and then transmit/receive the payload. Roaming scenarios in LoRaWAN are classified into ***passive*** and ***handover*** roaming. We will limit to passive roaming, since handover roaming is still in testing stage and an open LoRaWAN software stack for handover roaming is not yet available.

In ***passive roaming***, the MAC layer control of the ED is maintained by the home network NS, which becomes the serving NS (***sNS***), as shown in Figure 3. The roaming ED uses the NS of the visited network named as forwarding NS (***fNS***), to send messages to its sNS. The fNS forwards messages between the sNS and the ED.

If the ED is not yet activated, then it has to get activated using ***passive roaming activation*** process as shown in Figure 3. When the fNS doesn't have prior information about the sNS, the fNS SHALL use the DNS to find the roaming ED's JS IP address.

As per the LoRaWAN backend specifications, the LoRa Alliance has allocated a DNS Zone file (*joineuis.lorawan.net*) for provisioning the information mapping the JoinEUI to its corresponding JS operator. Each nibble of the JoinEUI represented in the hexadecimal format *0x00005E100000002F* is first reversed. Then, periods are inserted between each nibble and the domain name *joineuis.lorawan.net* is concatenated as the suffix. The final result is a domain name that can be provisioned in the DNS zone file *joineuis.lorawan.net* pointing to their respective JS as follows:

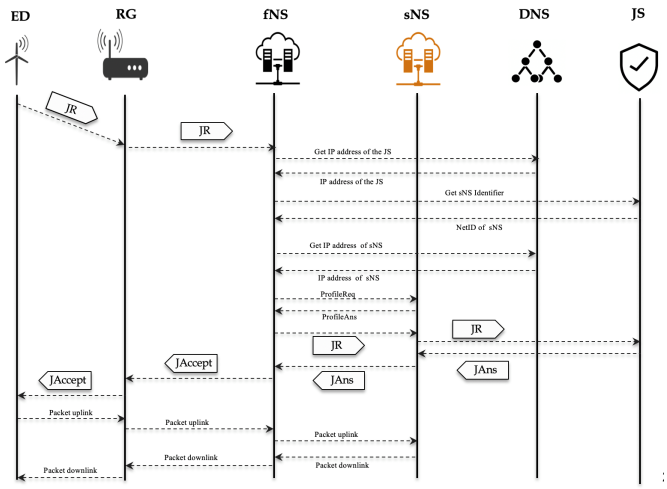



Figure 3: Passive Roaming Activation message flow

f.2.0.0.0.0.0.0.1.e.5.0.0.0.0.joineuis.lorawan.net. IN A 192.168.1.1

Thus, the fNS by running a standard DNS resolution can retrieve the IP address of the JS corresponding the JoinEUI and kickstart the *Join procedure*.

The fNS then queries and obtains the ***NetID*** (i.e. the 24 bit Unique Network Identifier of the sNS represented in the hexadecimal format: 0xC0002F) from the JS. Similar to JoinEUI, the LoRaWAN backend specifications has allocated a specific DNS Zone file (*netids.lorawan.net*) for mapping the NetID's to their corresponding NS. Any LoRaWAN (either private, public or community) needs to obtain from the LoRa Alliance a unique NetID, and provision it in the DNS zone file *netids.lorawan.net*, a standardised DNS resource record pointing the allocated NetID to its sNS as follows:

c0002f.netids.lorawan.net. IN A 192.168.1.2

Thus for a fNS, it is possible to resolve the sNS of a ED by querying the NetIDs DNS zone file, even if there is no prior roaming agreement. The sNS and the fNS exchange data and finally the ED is activated once the JoinAccept (JAccept) is received by the ED with the session keys for transmitting up/down link.

The DNS provisioning mechanism have ensured that both JoinEUI and NetID could be provisioned or updated by different entities in their respective DNS Zones (Servers), they are unique in the global scope and cannot be duplicated. The DNS resolution mechanism ensures that both the JS and the NS can be accessed from anywhere in the Internet with a simple DNS resolution. Figure 3 demonstrates how multi-stakeholders interconnection complexities are solved due to DNS provisioning and resolution, since for a single ED onboarding, the JS could be operated by different entity than the NS operator. Thus, the LoRaWAN architecture design by itself provides a ***partial*** solution to the WBA requirements described in section II.

## V. Federated Experimental Setup

Roaming deployments currently in LoRaWAN are based on One-to-One interconnection agreements or hub model. Both these models need to know ***in advance*** number of configuration parameters (such as *JS URI or IP address, NetID URI or IP address, Security mechanism for mutual authentication between the backend network elements in the IP space etc.*) before letting the roaming ED to one's network.

This section describes how we ***extended*** the existing LoRaWAN either by design or by implementation, to make it possible for an IoT ED to roam and access its required service without the need for having to know in advance the required configuration parameters.

Each of the LoRaWAN in our federated experimental setup has a basic set up as shown in Figure 2, wherein the AS and the JS runs on the same physical server. The NS and the AS server is installed with the open-source Chirpstack [25] as well as the pre-requisite software [26]. The IP interfaces between the different ***backend network elements*** (i.e. the JS, NS and the AS) in the IP space needs to take into account for the firewall rules, so that the required ports are open between the different interfaces.

The Chirpstack software stack did not have the functionalities of DNS resolution for OTAA or passive roaming by default. We ***collaborated*** with the Chirpstack developer to update the software to integrate both functionalities. The updated version of the software is available for beta testing [27] [28].

As explained in section IV, the NetIDs, JoinEUIs provisioning and resolution using the DNS are normalized in the LoRaWAN backend specification. But, there is no LoRaWAN DNS service in operation. As far as we know, from the literature and from being a member of the LoRa Alliance, we are the first to operate a PoC to test and validate the DNS functionality for OTAA and passive roaming in LoRaWAN. We set up a DNS infrastructure under the domain *iotreg.net* (instead of *lorawan.net* as defined in the LoRaWAN backend specifications) and tested by provisioning the JoinEUIs and NetIDs as shown in Figure 4. The DNS provisioning infrastructure and the resolution (thanks to the capability added in Chirpstack) enables to resolve the JoinEUI to its JS and NetID to its NS without having to share them in advance.

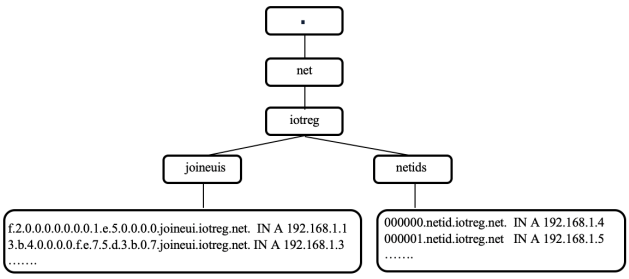



Figure 4: LoRaWAN DNS hierarchy set up for the PoC

For secure communication, the interface between the backend network elements in the IP space should be mutually authenticated (i.e. both the client and the server authenticates each other), as per the LoRaWAN backend specifications. But, the manner how the mutual authentication should be done is left to the implementer's choice and is not normative. In section II-C, we have reasoned using the X.509 digital certificates widely used to secure web traffic for IoTRoam. However, the Certificate Authority (CA) trust model for issuing the X.509 digital certificates is not operationally feasible for IoTRoam:

- In the web, the browser client (such as Chrome, Firefox) has a certificate store containing thousands of CA certificates. The browser authenticates any server that delivers a X.509 certificate digitally signed by anyone of the CA in its certificate store. Such certificate store infrastructure is not available in the LoRaWAN backend network elements or any IoT Backend infrastructures.
- Even if we assume, the infrastructure exists, the digital certificated comes at a cost, which is more than the cost of a normal IoT service. Hence, not operationally viable.
- A solution to resolve the cost issue is to use Self-Signed certificates. However, the downside is that they are not scalable or dynamic.

We propose to follow the eduPKI [29] model used in eduroam. Each IoT organisation (e.g. LoRa Alliance for LoRaWAN, GS1global for supply chain Industry) acts as the Root CA and generates Intermediate CA to different networks in their respective communities. The intermediate CAs will, in turn, generate the leaf certificates for individual servers (Figure 5 shows a scenario wherein Afnic plays the role of root CA and generates intermediate certificates for two LoRaWANs - TSP & Afnic Labs). Typically, the organisations should provide the intermediate CAs free of cost, and if each node has to have a certificate store of root CA certificates for each IoT standard (the number of CAs will be less than 100 if we envision a root CA for each IoT standard or technology), then implementation becomes operationally feasible.

A CA provisioning infrastructure (Figure 5) was set up, and details on how to obtain an Intermediate Certificate and generate the leaf certificates documented [30], to benefit Institutions willing to test the federated IoTRoam platform. We further simplified the process, wherein the interested organisation can generate the leaf certificates by just running a makefile [31] after customising the configuration files [32]. Each of the NS [33] and the AS [34] in the federated platform add the certificates generated into their configuration file.

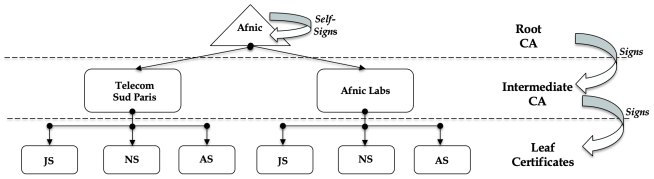



Figure 5: IoTRoam Certificate provisioning infrastructure

### A. Testing the federated IoTRoam platform

For testing the federated platform set up, two LoRaWAN locations, separated by a distance of 34 kilometres as per Google maps were used. The two locations are: Afnic (in the Yvelines department in France) and Telecom Sud Paris (TSP) (in the Essonne department in France). A ED configured with Afnic backend network elements was ***moved*** to TSP and a TSP associated ED was moved to Afnic. Both the EDs were not ***activated***.

Figure 6 shows that the ED configured with the TSP backend elements uses the RG in Afnic's coverage area for activation (Step 1). The RG forwards (Step 2) the incoming JR to the Afnic NS which plays the role of the fNS, which in turn uses the DNS infrastructure (Step 3 & 4) for JS resolution, since the ED is not known to it. After retrieving the IP address of the TSP-JS (Step 4), the fNS and the TSP-JS runs a TLS handshake for mutual authentication (Step 5). During the course of mutual authentication testing, we identified that ***combining the intermediate and the server leaf certificate*** (combined trust chain) during a TLS handshake could bypass the need for having a certificate store with all intermediate certificates and store only the root CA certificate. The certificate validation process is done by sending the combined trust chain to the IP address of the server resolved. On receiving the combined trust chain, the server first verifies the leaf certificate in the combined trust chain. When the leaf certificate is unknown , it checks the next certificate in the chain, which is the intermediate certificate. Since the intermediate certificate is signed by the root CA and is known to all the servers in the federation, the chain becomes trusted. Thus, the backend network elements (NS, AS and the JS) could be mutually authenticated even if they are in different networks, since they have a common root CA at the top of the chain of trust.

Once the mutual authentication between the fNS and the TSP-JS is successful, the fNS retrieves the NetID of the ED from the TSP-JS (Step 6). Using the retrieved NetID, the IP address of the ED's NS (i.e. the sNS) is obtained (Step 7 & 8) via DNS resolution. Then the fNS and the sNS does the passive roaming message flow thanks to the mutual authentication (Step 9)

## VI. Performance evaluation

When the usage of DNS for OTAA and roaming scenarios were presented before the LoRa Technical Committee, there were concerns about the delay caused by DNS resolution time and the impact on Class A ED's downlink reception. There are three Class of LoRaWAN EDs' : Class A, B and C. Class A is the most energy efficient and results in longest battery life, since most of the time the ED is in sleep mode. Following an uplink, the Class A ED opens a receive window for 1 second (default value) and if no downlink is received during the period, it opens a second receive window for another second (default value). If no downlink communications is received from the server between the two receiver window, then it must wait until the ED triggers next uplink and opens a receiver window.

In order to evaluate the performances of using DNS and PKI to support roaming in a federation, we flashed a short algorithm on a Multitech MDOT ED aiming to measure the activation time as well as the time the ED needs in order to send its first packet. To ensure that the measurement is precise and get rid of any synchronisation error between the ED and the backend network elements, all measurements were realised directly on the ED. For OTAA measurements, we registered the ED in the TSP (Figure 6) backend infrastructure and moved it in the Afnic radio coverage in order to test ED transmission in a visiting scope.

Figure 7a shows the distribution of the activation time for the ED measured across multiple OTAA. The ED is able to consistently activate in ***32.6 ms*** while roaming in the visited infrastructure which proves that our proposed mechanisms does not considerably increase the latency of the activation time, thus satisfying the LoRaWAN Class A constraints.

Fig 7b shows the distribution of the time the device needs in order to send its first packet as a visiting device, after the aforementioned activation. ***95 %*** of the first packet are sent successfully as soon as possible for our device respecting duty cycle regulation, while ***5 %*** encounter transmission error which force a ***6 second*** backoff in data transmission. The times and error rate measured from the visiting device are similar to the ones we can observe from a device in their own home network, hence we can conclude that the roaming infrastructure does not considerably augment the latence which hinder packet transmission. In our set up, we did have the positive effects of DNS caching, but the performance evaluation demonstrates that DNS will not considerably impact the Class A downlink transmission within the receiver window.

## VII. Contributions

The IoTRoam experience enabled us to set up a federated platform which has been documented and the software provided as open source to the community. It also helped us to identify operational issues which has not been encountered earlier, since there is no LoRaWAN operational infrastructure using DNS for OTAA and roaming. The PoC tests has proposed solutions to some of the operational issues and also led to ***four change requests*** (Change request is the procedure to provide modifications to the LoRaWAN specifications) of which ***three*** has been adopted by the LoRaWAN backend specifications. This section will detail the contributions.

*a) Contribution 1:* The networks based on One-to-One interconnection or hub drops the incoming packet from an ED if it is not part of its network or its partners. With IoTRoam federated model, these networks could make a DNS resolution to identify the home network of the ED. Thus, the IoTRoam federated model caters to the whole ecosystem wherein networks based on the hub or One-to-One interconnection could co-exist.

*b) Contribution 2:* In the Cellular model, portability between operators becomes possible since there is a human

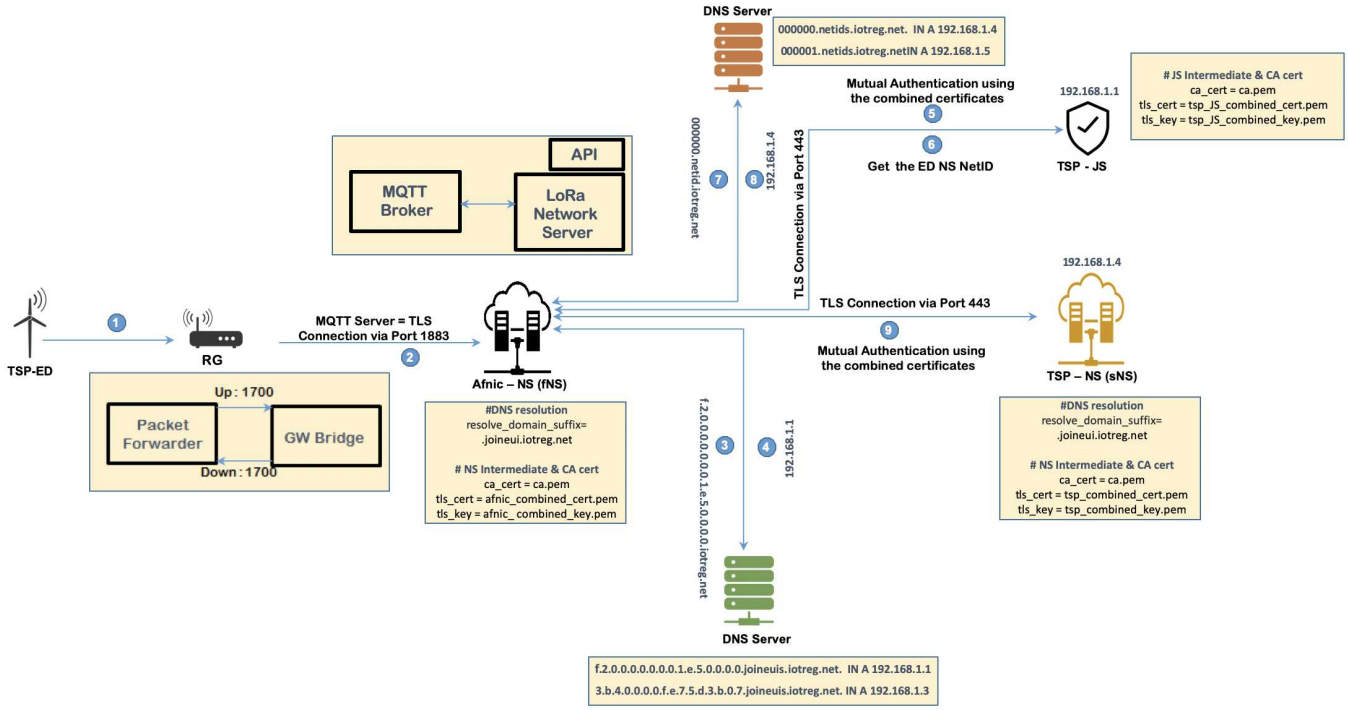
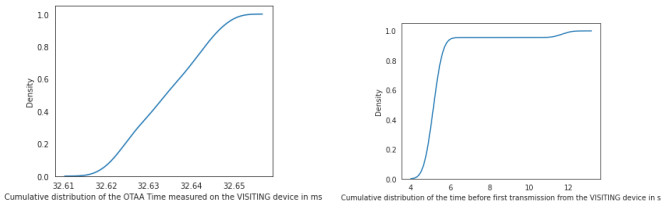



Figure 6: Testing Passive roaming OTAA using the proposed federated model




(a) Activation Time (b) First transmission time

Figure 7: Performance evaluation measured on the ED in a visiting scope

Table I: Sample representation of DevEUI and JoinEUI 64 bits partitioning

| | DevEUI | JoinEUI |
|---|---|---|
| ED 1 | OUI-ABBB-0001 | OUI-ABBB-JJJ1 |
| ED 2 | OUI-ABBB-0002 | OUI-ABBB-JJJ1 |
| ED 3 | OUI-ABBB-0003 | OUI-ABBB-JJJ1 |
| .... | .... | ... |
| ED 501 | OUI-BBBB-0001 | OUI-BBBB-JJJ2 |
| ED 502 | OUI-BBBB-0002 | OUI-BBBB-JJJ2 |
| ED 503 | OUI-BBBB-0003 | OUI-BBBB-JJJ2 |
| .... | .... | ... |

subscriber involved, which is not the case in LoRaWAN. In LoRaWAN, the EDs' with a battery life spanning for a decade are supposed to be set up in remote places and not readily accessible in the necessity of a network operator change. IoTRoam enables ***portability*** between different operators; thanks to the DNS database, the JS pointing to a JoinEUI can be modified without making any modification at the ED level.

To understand the importance of operator portability, a brief background of how the ED is provisioned with the JoinEUI and DevEUI are needed. The JoinEUI 64 bit address could be divided into three broad ranges :*OUI* of the manufacturer, the *Batch ID* of the manufacturer and the *JoinEUI* value assigned to the batch. The DevEUI is also a unique IEEE EUI-64 bit address divided in the same categorization as the JoinEUI. The difference is - for every ED there is a unique DevEUI, but thousands of EDs' could be assigned a single JoinEUI as shown in the table I :

During the JoinEUI assigning process, the ED manufacturer is not yet aware who will be the buyer. If a client is buying only 500 EDs' from a batch of 1000, the remaining 500 EDs' JoinEUI need to be re-provisioned with a new JoinEUI, if a new buyer wants the remaining 500 EDs' to point to a different JS. Similarly, if the buyer who has bought 500 EDs' from a batch needs to assign a different JS for a set of 100 EDs', the JoinEUI needs to be modified in each ED. This modification is done by re-flashing the EDs' with the new JoinEUI and thus is operationally time consuming and costly.

The PoC experience enabled us to suggest a ***change request*** to provide an operationally feasible solution, which has been accepted and included in the LoRaWAN backend specifications. The solution proposed by the change request is the creation of a combination of the DevEUI (which is unique for each device) and JoinEUI, and provision them in the DNS. In order to adapt to this requirement, the NS should first make a

DNS query using the concatenation of DevEUI and JoinEUI, and if the resolution fails, it falls back in making a DNS query only using the JoinEUI.

Taking an example where two ED's (*0xACDE480001020234*, *0xACDE480001020ABC*) should point to two different JS's, but has a single JoinEUI represented in the hexadecimal format as *0x00005E100000002F*. The DevEUI JoinEUI combination could be provisioned in the DNS pointing to two different JS's as follows:

4.3.2.0.2.0.1.0.0.0.8.4.e.d.c.a.f.2.0.0.0.0.0.0.0.1.e.5.0.0.0.0.
joineuis.lorawan.net IN 192.168.2.6
a.b.c.0.2.0.1.0.0.0.8.4.e.d.c.a.f.2.0.0.0.0.0.0.0.1.e.5.0.0.0.0.
joineuis.lorawan.net IN 192.168.2.7

Based on the longest match algorithm, the DNS resolution will resolve to two different JS's for the two ED's even though the JoinEUI are same for both the EDs'

*c) Contribution 3:* The second ***change request*** which has been adopted into the LoRaWAN backend specification include modifying the sub domains for join and roaming from *lora-alliance.org* to *lorawan.net*, thus separating the LoRa Web and DNS service.

*d) Contribution 4:* A third ***change request*** which has been adopted into the LoRaWAN backend specification include updating the DNS provisioning and resolution section to enable the usage of any DNS resource record for OTAA and roaming functionalities. Before the change request, the LoRaWAN backend specifications was normalized using NAPTR DNS resource record which is considered quite complex (Explained in RFC 3401, 3402 & 3403) for operational purposes.

*e) Contribution 5:* We developed and provided a secure, automatized DNS provisioning platform that could be used by the community. Any authorized user can access the User Interface (UI) (via web or API) with a secured API key. The UI enables authorized users to do multiple operations (creation, modification, deletion) of only their data in the DNS database. To make it easy for the community to understand and use the interface, a Video Tutorial [35] is provided.

While testing the UI with some LoRa Alliance community members, we encountered operational issues such as: how to validate the data provisioned in the DNS is manipulated by the rightful owner. The need for validating the JoinEUI (which is a IEEE EUI-64 identifier provisioned by the IEEE and has Organisational Unique Identifier (OUI) in the IEEE EUI-64) with the IEEE OUI database, were identified and implemented, thanks to the PoC. The implemented solution has been provided as feedback to the LoRa Alliance, which could be integrated when the DNS service operated by the LoRa Alliance is deployed.

## VIII. Conclusion

Our vision with IoTRoam was to achieve the same service as that of Cellular or Wi-Fi Roaming, but with an increased level of heterogeneity (different IoT identifiers and connectivity technologies) and complexities (multi-stakeholders involved in AA the ED in a roaming scenario). We added a hard requirement that the infrastructure or technologies used to achieve this vision should be viable, operationally feasible and could be integrated to existing IoT infrastructures with minimum changes. We followed the WBA guidelines for open roaming and satisfied the requirements outlined, by employing open standards extensively used in the Internet such as DNS and PKI, to achieve our vision. We chose LoRaWAN (a standard that is evolving) and demonstrated that seamless IoT roaming with minimum prior configuration is possible with the federated IoTRoam model. In this process we have deployed a PoC and provided all necessary building blocks (documentation, software, UI, video tutorial) so that the community could make use of them to implement their own network or federation. This experience has also helped us to provide feedback as change requests, that has been adopted into the LoRaWAN backend specifications. In addition to TSP, we are in communication with several institutions in France and one in Denmark and Italy to run interoperable testing using the federated platform. Running additional tests with these Institutions would help us study the impact of heterogenous backend infrastructures and their effect on the quality of the communication channel. It would also allow us to gather additional data on the impact of DNS complete resolution on the LoRaWAN/IoT traffic.

There are three identified lacunaes in IoTRoam: 1. Since Accounting (the third 'A' in AAA) is not considered, the proposed model is not acceptable for commercial deployments, 2. There is no end-to-end AA, since X.509 digital certificates as it is cannot be transferred over bandwidth-constrained IoT networks and 3. How to convince well established IoT infrastructures to migrate from their respective identifier resolution and AA mechanism to using DNS and PKI?. For (2), as part of the French funded ANR project *DiNS*, we are working on designing a compressed X.509 certificate. We are also working on using DNS Authentication of Named Entities (DANE) since the certificate data itself can be stored in the DNS, possibly obsoleting the PKI. Achieving (1) and (3) seems to be a distant possibility.

## IX. Acknowledgements

The French Government ANR project DiNS has partially funded this work. We want to acknowledge Orne Brocaar, the author of Chirpstack LoRaWAN open source stack who has helped us to update the Chirpstack software for OTAA and passive roaming using DNS; Alper Yegin, the vice-chair of the LoRa Alliance for supporting this discussion at the LoRa Alliance and also the LoRaWAN community where we were able to gain insights on real operational issues.

## References

[1] *Walled Garden*. https://www.insidetheiot.com/freedom-walled-garden/.

[2] *LoRaWAN Backend specs*. https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf.

[3] Farrell, S., Ed. “"Low-Power Wide Area Network (LPWAN) Overview", RFC 8376”. In: (May 2018).

[4] *Eduroam Website*. https://www.eduroam.org/.

[5] *IoTRoam PoC*. https://github.com/afnic/IoTRoam-Tutorial/. 2020.

[6] *IoTRoam-QuickStart*. https://github.com/AFNIC/IoTRoam-Tutorial/blob/master/QuickStart.md.

[7] E. M. Torroglosa-Garcia et al. “Enabling Roaming Across Heterogeneous IoT Wireless Networks: LoRaWAN MEETS 5G”. In: *IEEE Access* 8 (2020), pp. 103164–103180.

[8] *OpenRoaming, Wireless Broadband Alliance*. https://wballiance.com/openroaming/. 2020.

[9] *"IoT New Vertical Value Chains and Interoperability", Wireless Broadband Alliance (WBA)*. https://www.wballiance.com/wp-content/uploads/2017/03/IoT-New-Vertical-Value-Chains-and-Interoperability-v1.00.pdf. 2020.

[10] *DINR - (DNS and Internet Naming Research Directions) Workshop*. https://ant.isi.edu/events/dinr2016/P/p72.pdf. Nov. 2016.

[11] J. Peterson. “Telephone Number Mapping (ENUM) Service Registration for Presence Services, RFC 3953”. In: (January 2005).

[12] *Object Naming Service*. https://www.gs1.org/sites/default/files/docs/epc/ons_2_0_1-standard-20130131.pdf. 2013.

[13] *Approximative number of websites*. https://www.internetlivestats.com/total-number-of-websites/. 2018.

[14] G. Hatzivasilis et al. “The Interoperability of Things: Interoperable solutions as an enabler for IoT and Web 3.0”. In: IEEE 23rd International Workshop on Computer Aided Modeling, Design of Communication Links, and Networks (CAMAD). IEEE, 2018.

[15] “Advancing IoT Platforms Interoperability”. In: ed. by Norway Ovidiu Vermesan SINTEF. River Publishers Series in Information Science and Technology, June 2018.

[16] Ivan Gojmerac et al. “Bridging IoT islands: the symbIoTe project”. In: *2nd IEEE International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*. Springer. September 2016, pp. 315–318.

[17] A. Blázquez Rodríguez. “Security and AAA Architectures in an IoT Marketplace”. http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-234660. MA thesis. Uppsala University, 2014.

[18] B. Stackpole. “Centralized authentication services (RADIUS, TACACS, DIAMETER)”. In: Sixth. Information Security Management Handbook, Jan. 2007, p. 909.

[19] Ali Dehghantanha Kim-Kwang Raymond Choo, ed. *Blockchain Cybersecurity, Trust and Privacy*. Vol. 79. Springer, 2020.

[20] "Na Shi et al. “A Blockchain-Empowered AAA Scheme in the Large-Scale HetNet”. In: *"Digital Communications and Networks"* (2020).

[21] *Distributed PKI vs Traditional PKI*. https://dzone.com/articles/distributed-pki-vs-traditional-pki. June 2020.

[22] K. Wierenga, S. Winter, and T. Wolniewicz. “"The eduroam Architecture for Network Roaming", RFC 7593”. In: (September 2015). https://www.rfc-editor.org/info/rfc7593.

[23] *IoT Onboarding Mail Archive discussion*. https://mailarchive.ietf.org/arch/browse/iot-onboarding/?gbt=1&index=XHSahBLvpUU8Sasvt77QLX43OhQ. December 2019.

[24] *WBA-IoT-Dynamic-Roaming*. WBA White paper. https://wballiance.com/iot-interoperability-and-roaming-iot-dynamic-roaming/. December 2019.

[25] *ChirpStack Website*. https://chirpstack.io/. 2020.

[26] *Chirpstack Pre-requisite Softwares*. https://www.chirpstack.io/network-server/install/requirements/.

[27] *ChirpStack v3.11 release note*. https://forum.chirpstack.io/t/release-chirpstack-network-server-v3-11-test-releases/9502/. 2020.

[28] *ChirpStack v3.13 release note*. https://forum.chirpstack.io/t/release-chirpstack-application-server-v3-13-test-releases/9501/. 2020.

[29] *eduPKI*. https://www.edupki.org/home/. 2020.

[30] *IoTRoam Certifices Tutorial*. https://github.com/AFNIC/IoTRoam-Tutorial/blob/master/Certificates-Tutorial.md.

[31] *IoTRoam Certificates Generation*. https://github.com/AFNIC/IoTRoam-Tutorial/blob/master/certificates/Makefile. 2020.

[32] *Certificates Config files*. https://github.com/AFNIC/IoTRoam-Tutorial/tree/master/certificates/config.

[33] *NS Config*. https://github.com/AFNIC/IoTRoam-Tutorial/blob/master/Server-Config-Files/chirpstack-network-server.toml.

[34] *AS Config*. https://github.com/AFNIC/IoTRoam-Tutorial/blob/master/Server-Config-Files/chirpstack-application-server.toml.

[35] *Video Afnic*. https://iot.rd.nic.fr/Video/version3.mp4. 2020.