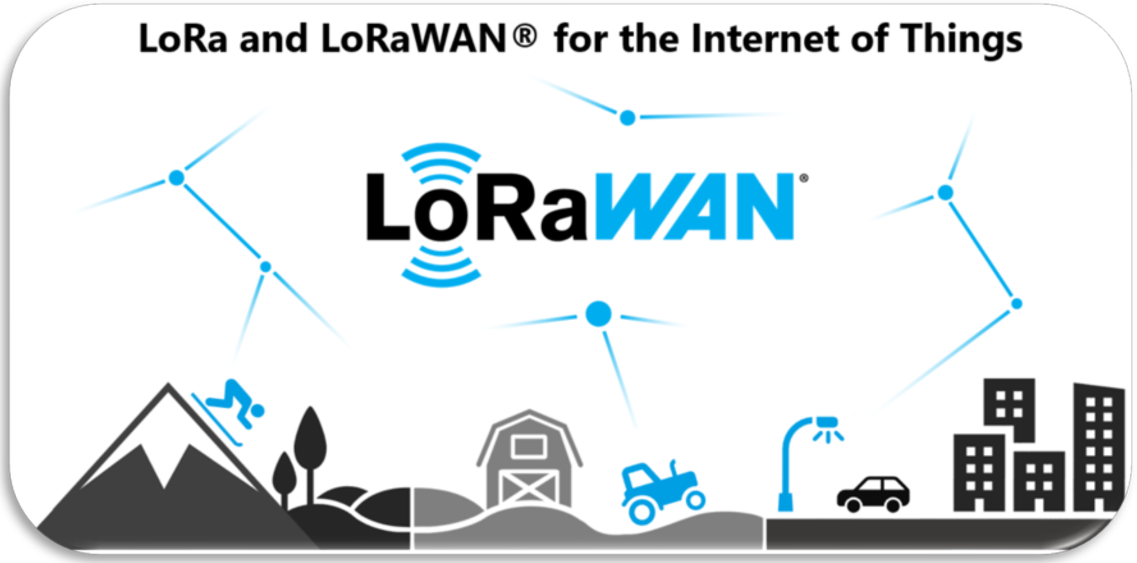
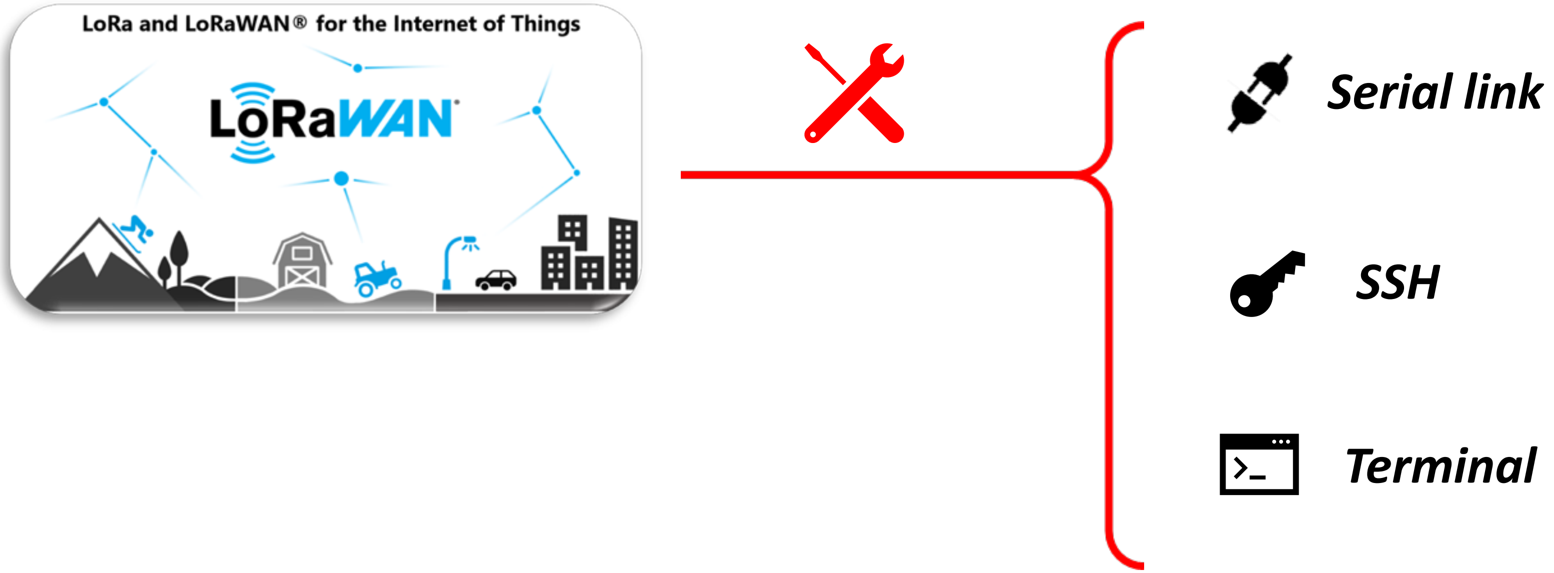
LoRa and LoRaWAN® for the Internet of Things

HOWTO - MobaXterm installation



HOWTO - MobaXterm installation



HOWTO - MobaXterm installation

Serial link

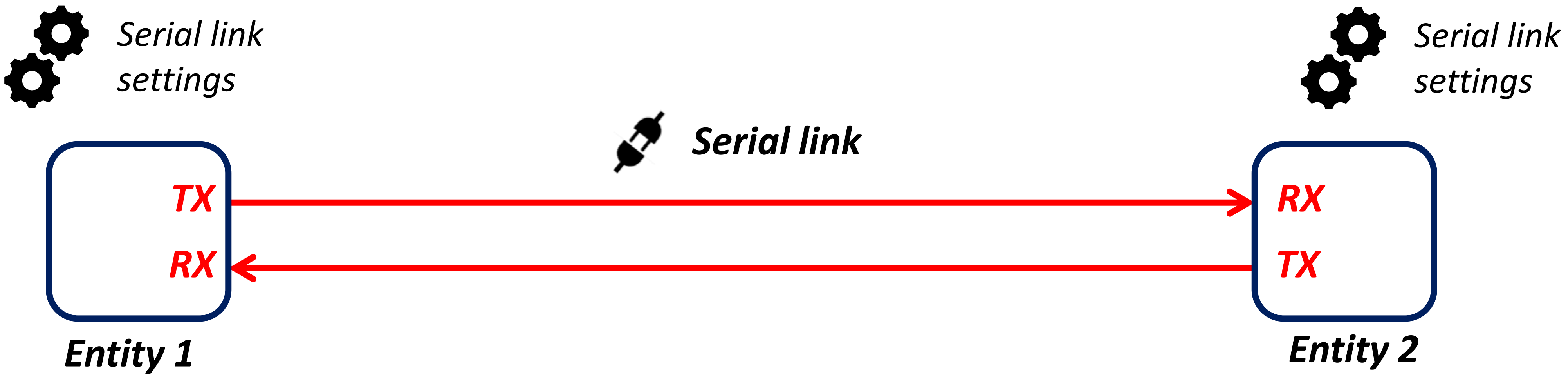
SSH

Terminal

- *Explanations and no tutorial style*
- *Demonstrations, that goes straight to the point*
- *Proposition of tools to achieve our goals*

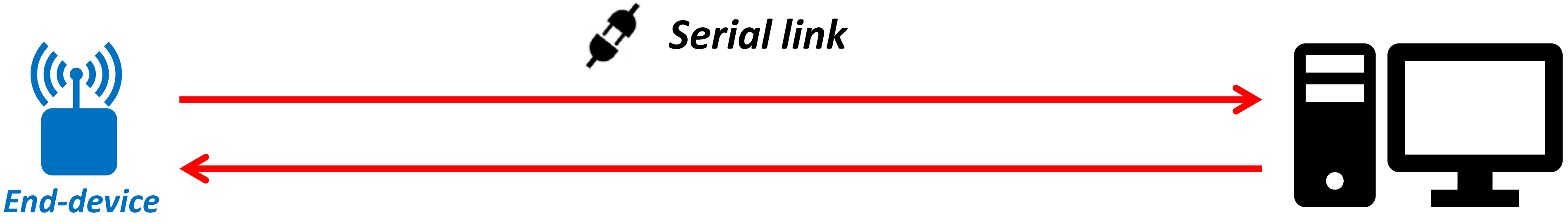
HOWTO - MobaXTerm - Open a serial link connection

HOWTO - MobaXTerm - Open a serial link connection



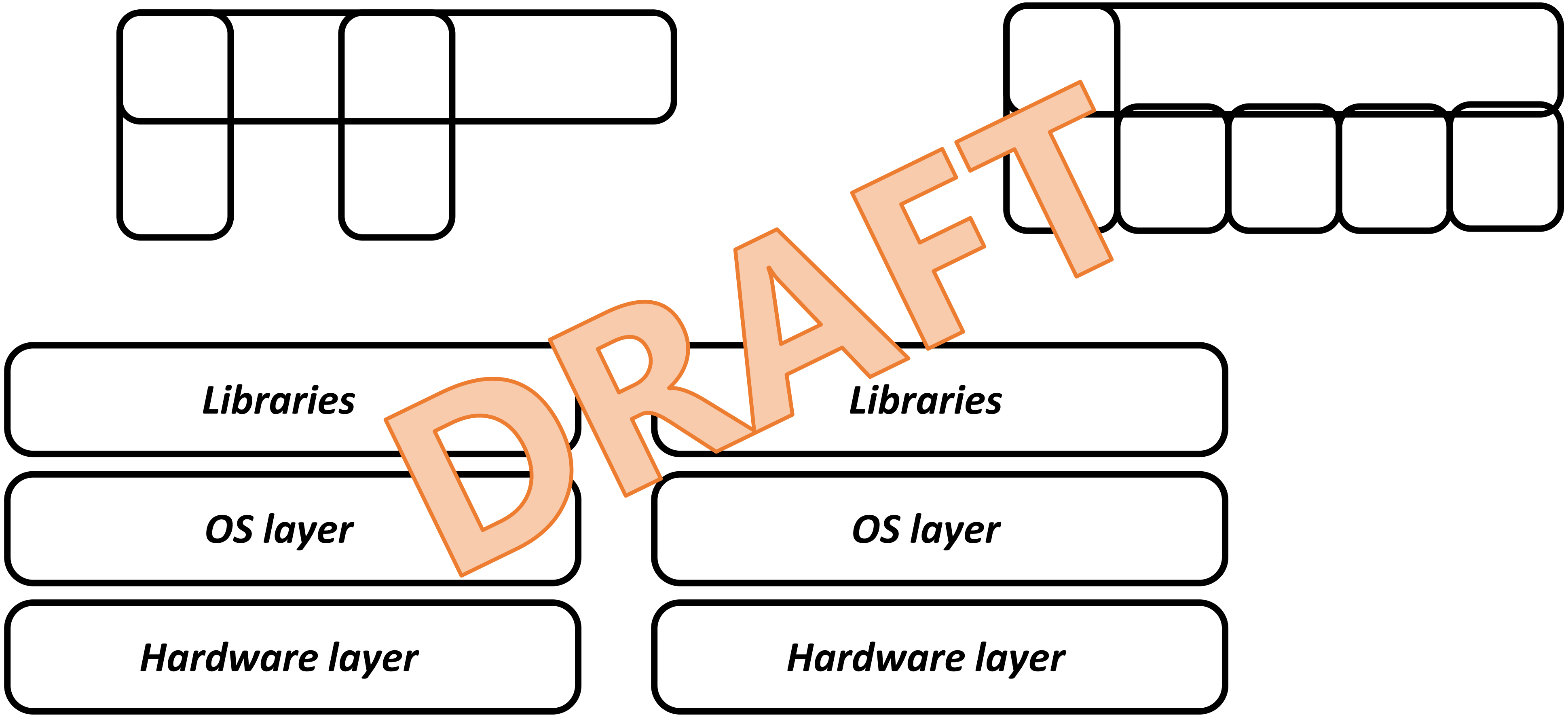
HOWTO - MobaXTerm - Open a serial link connection

Our example:

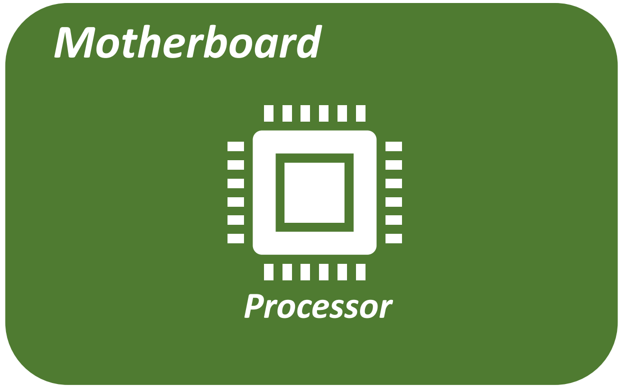


LOGs activated
(status, what it sends, etc.)

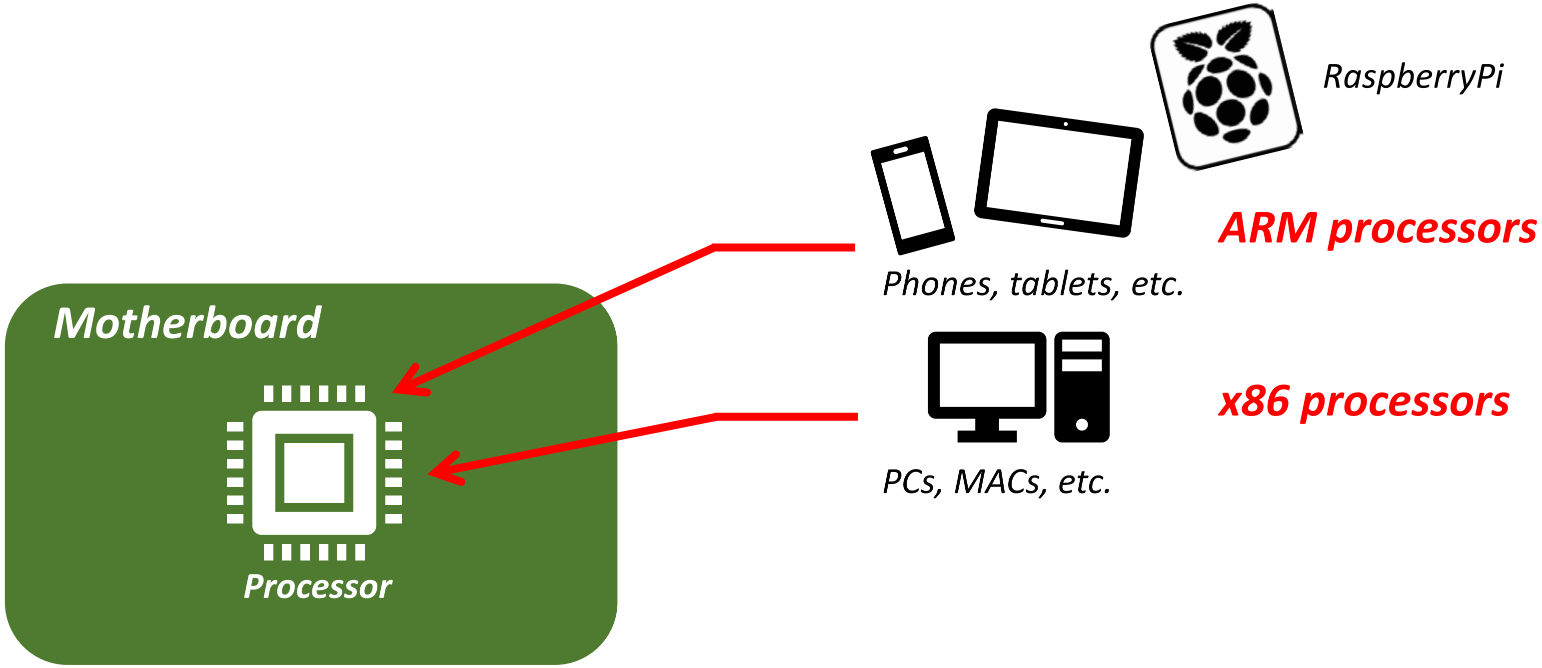
HOWTO - Setup docker and docker-compose



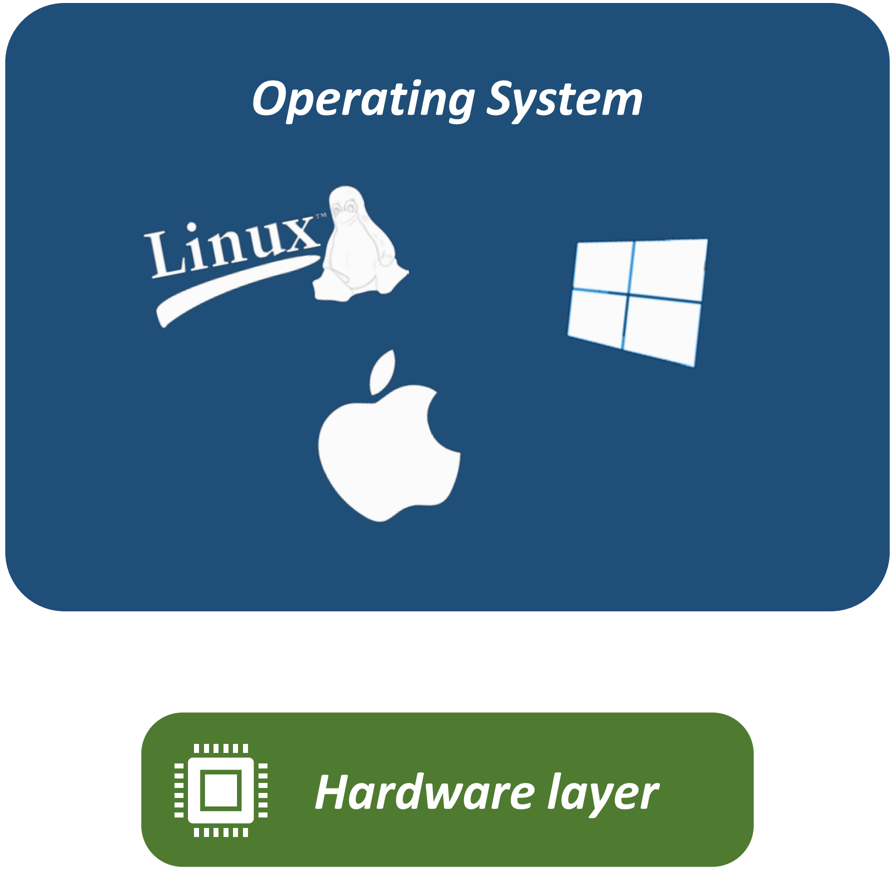
HOWTO - Setup docker and docker-compose



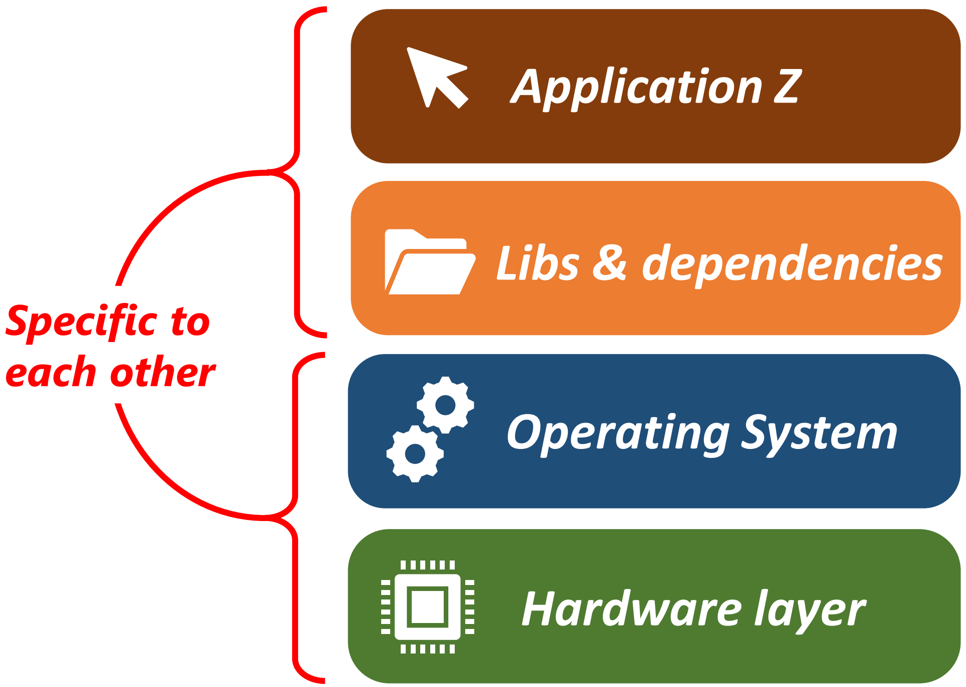
HOWTO - Setup docker and docker-compose



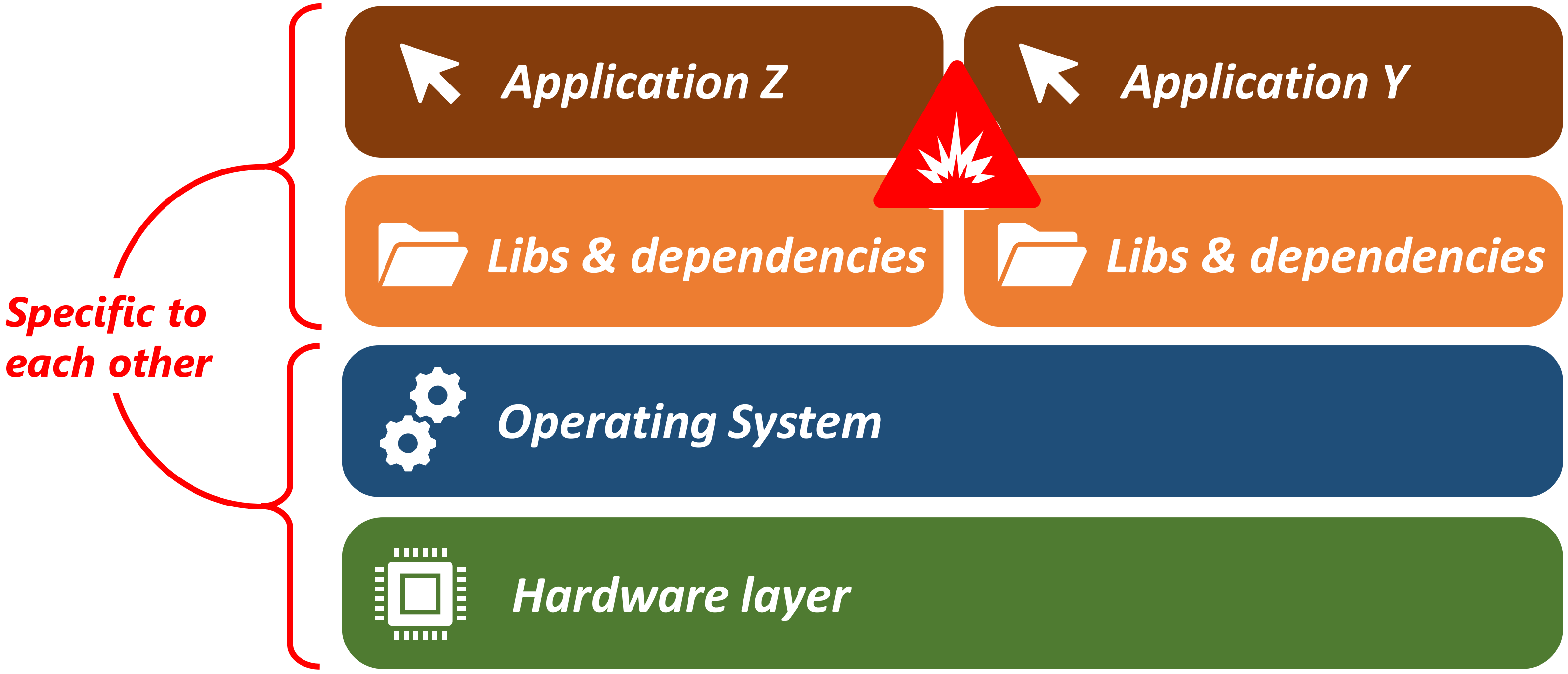
HOWTO - Setup docker and docker-compose



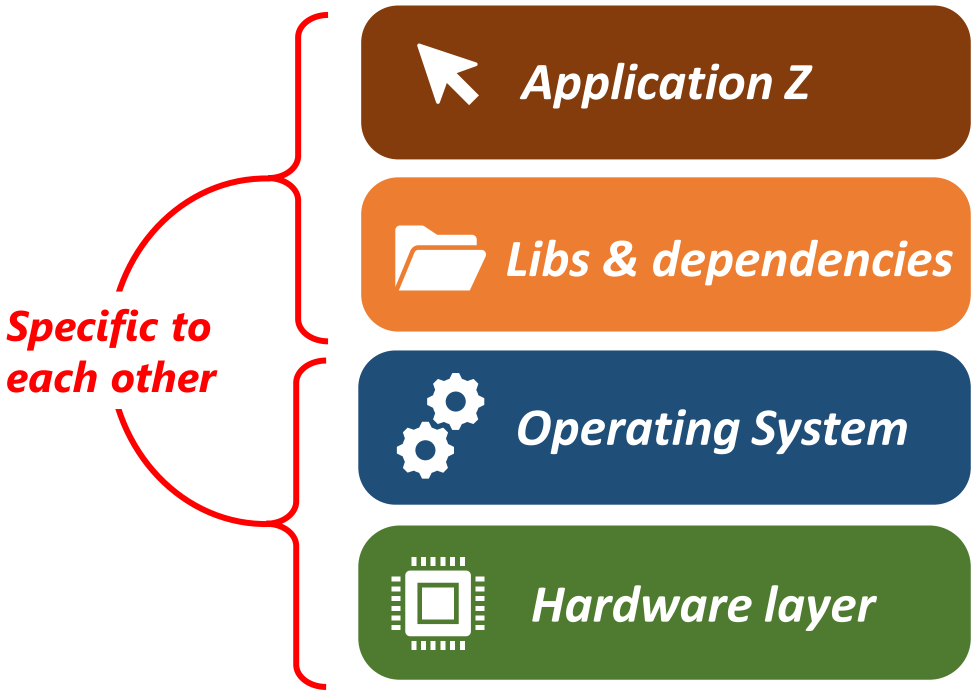
HOWTO - Setup docker and docker-compose



HOWTO - Setup docker and docker-compose



HOWTO - Setup docker and docker-compose



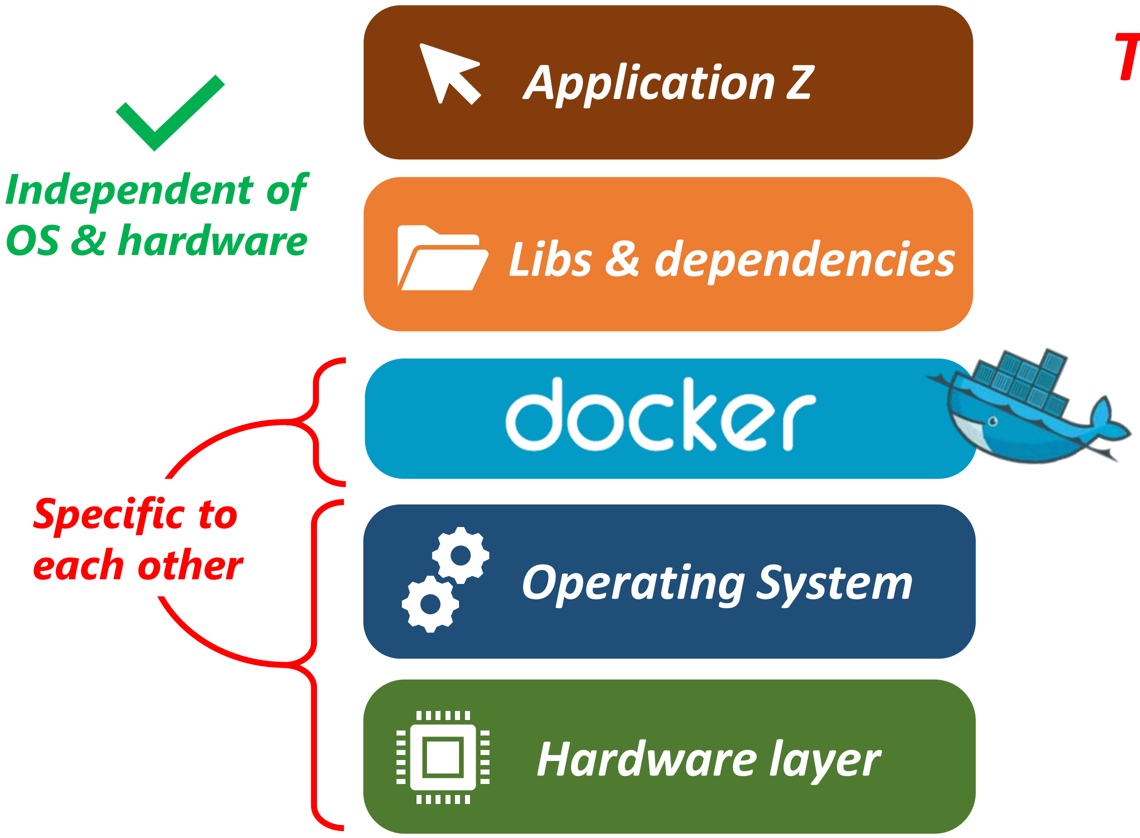
TO SIMPLIFY THE PROCESS:

→ *complete separated environment for applications*

→ *independent of the processor type*

→ *independent of the Operating system*

HOWTO - Setup docker and docker-compose

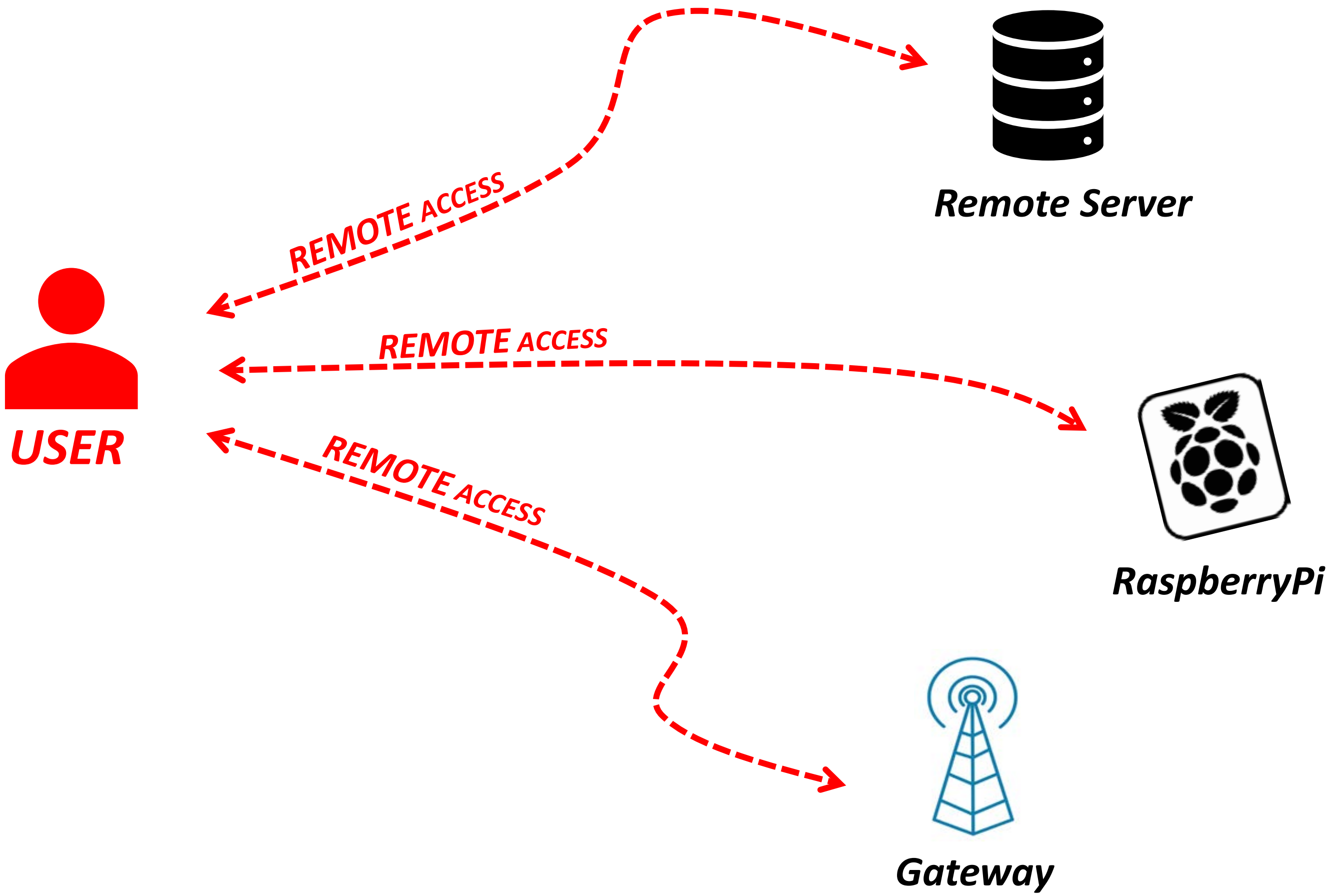


TO SIMPLIFY THE PROCESS:

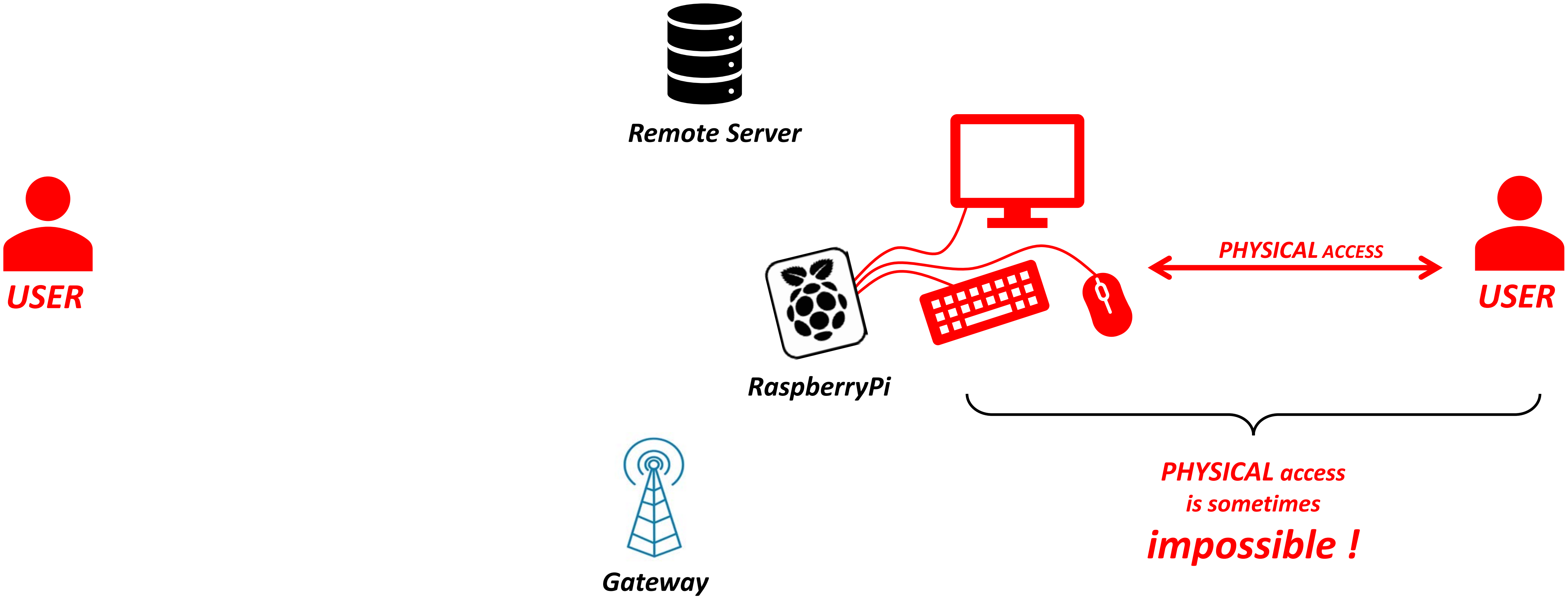
- → *complete separated environment for applications*
- → *independent of the processor type*
- → *independent of the Operating system*

docker

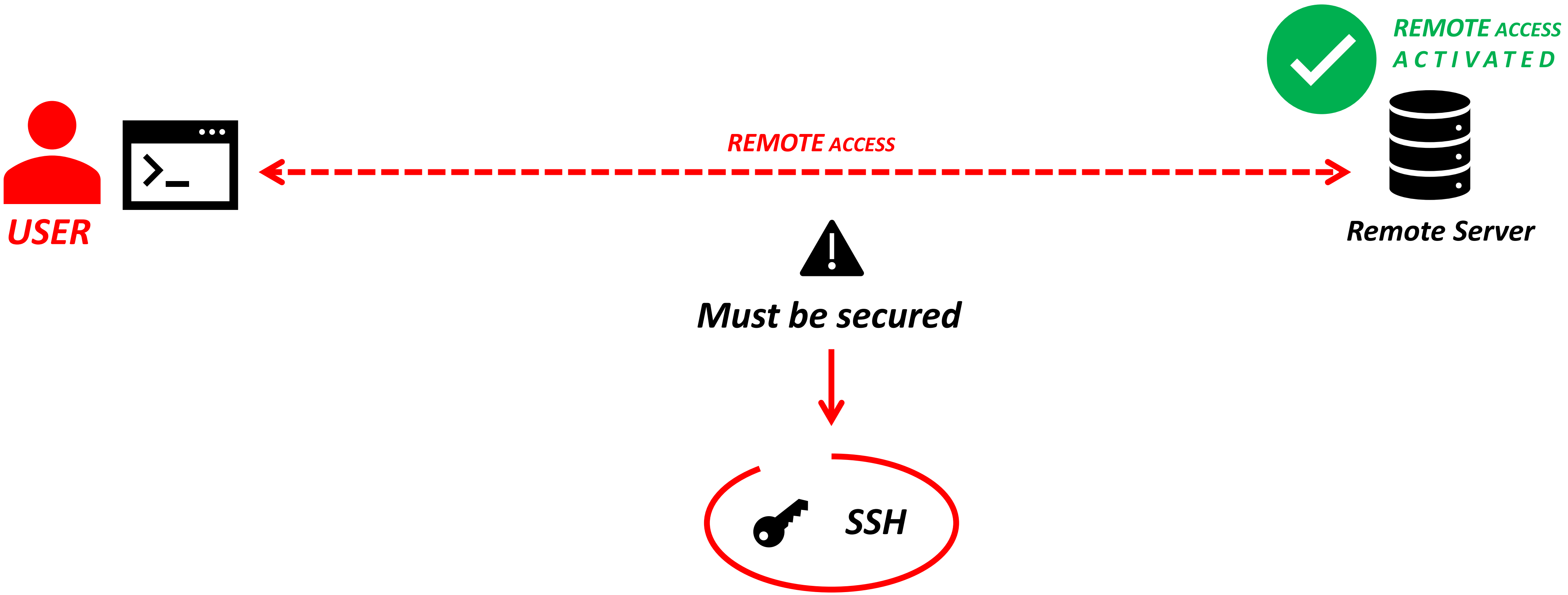
HOWTO - MobaXterm - Open a SSH connection



HOWTO - MobaXterm - Open a SSH connection



HOWTO - MobaXterm - Open a SSH connection



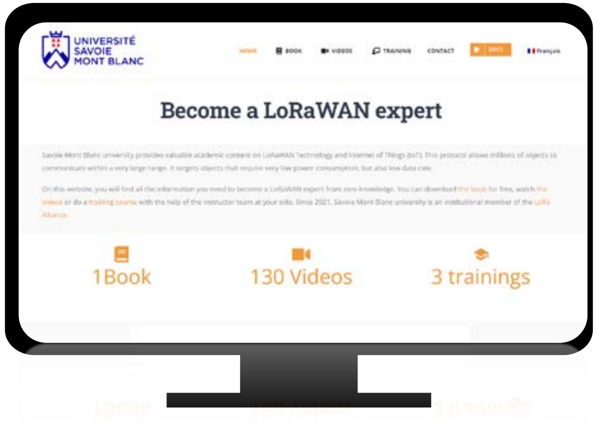
LoRa and LoRaWAN® for the Internet of Things

LoRa and LoRaWAN® for the Internet of Things

LoRaWAN

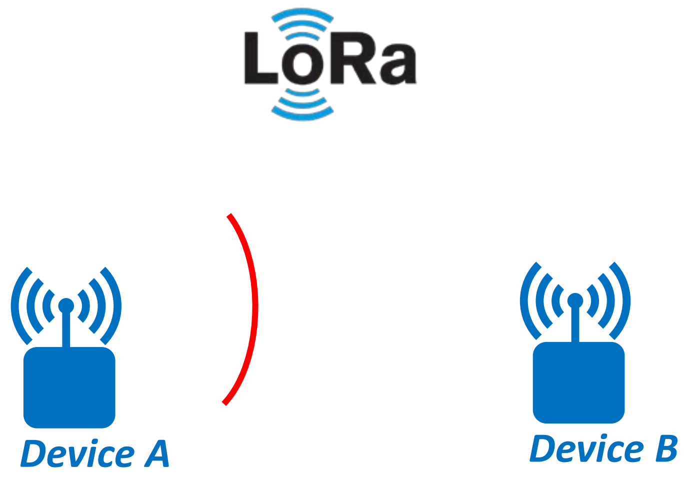

UNIVERSITÉ SAVOIE MONT BLANC
LoRa - LoRaWAN and Internet Of Things
LoRaWAN
Sylvain MONTAGNY
A LOW POWER, LONG RANGE, WIRELESS TECHNOLOGY

LoRaWAN®

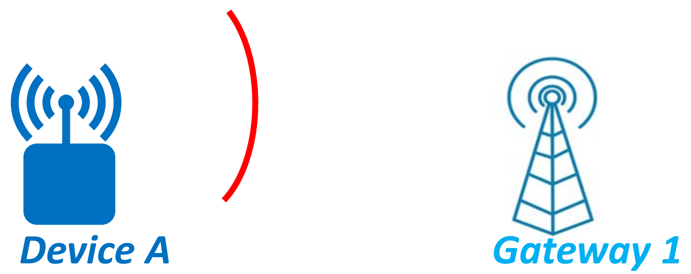


www.univ-smb.fr/lorawan/en/

www.univ-smb.fr/lorawan/en/

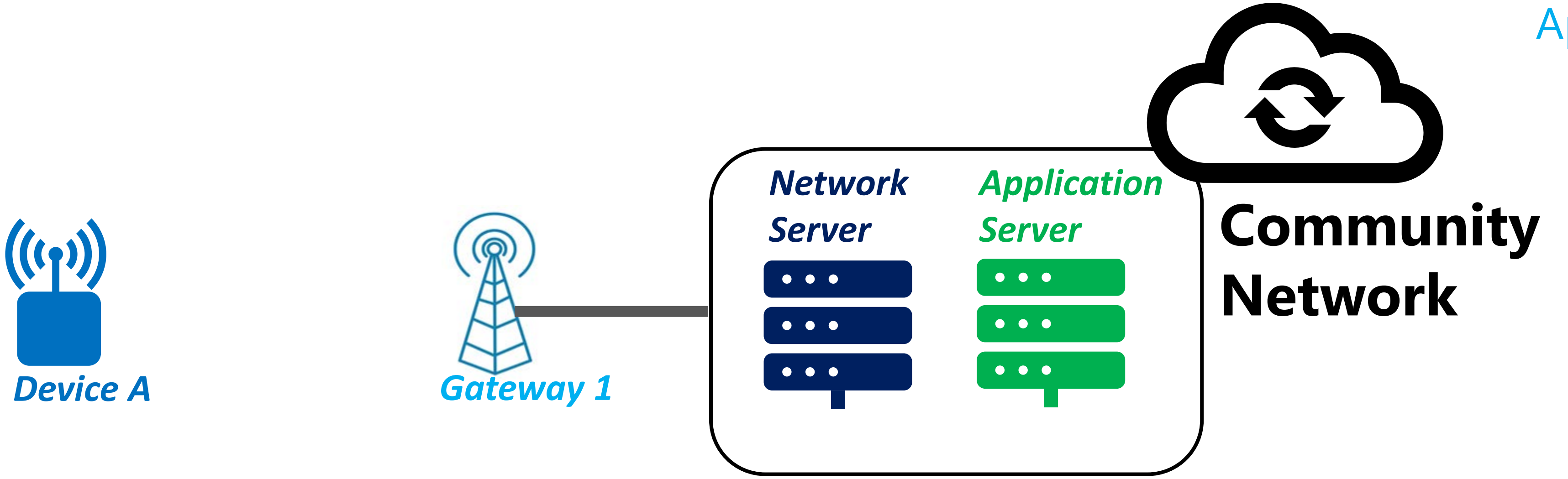

LoRa
Device A
Device B

LoRa



Bandwidth, channels, Spreading Factor

LoRaWAN®



Application Server

Network Server

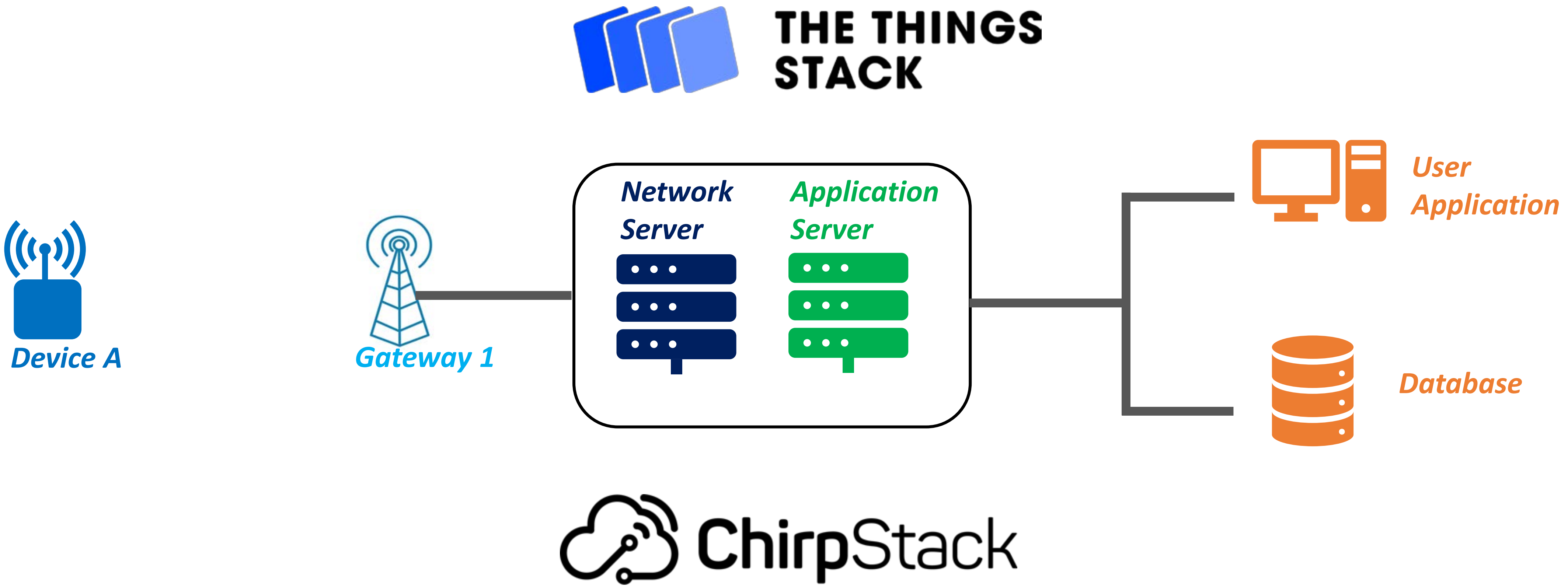
Low Power

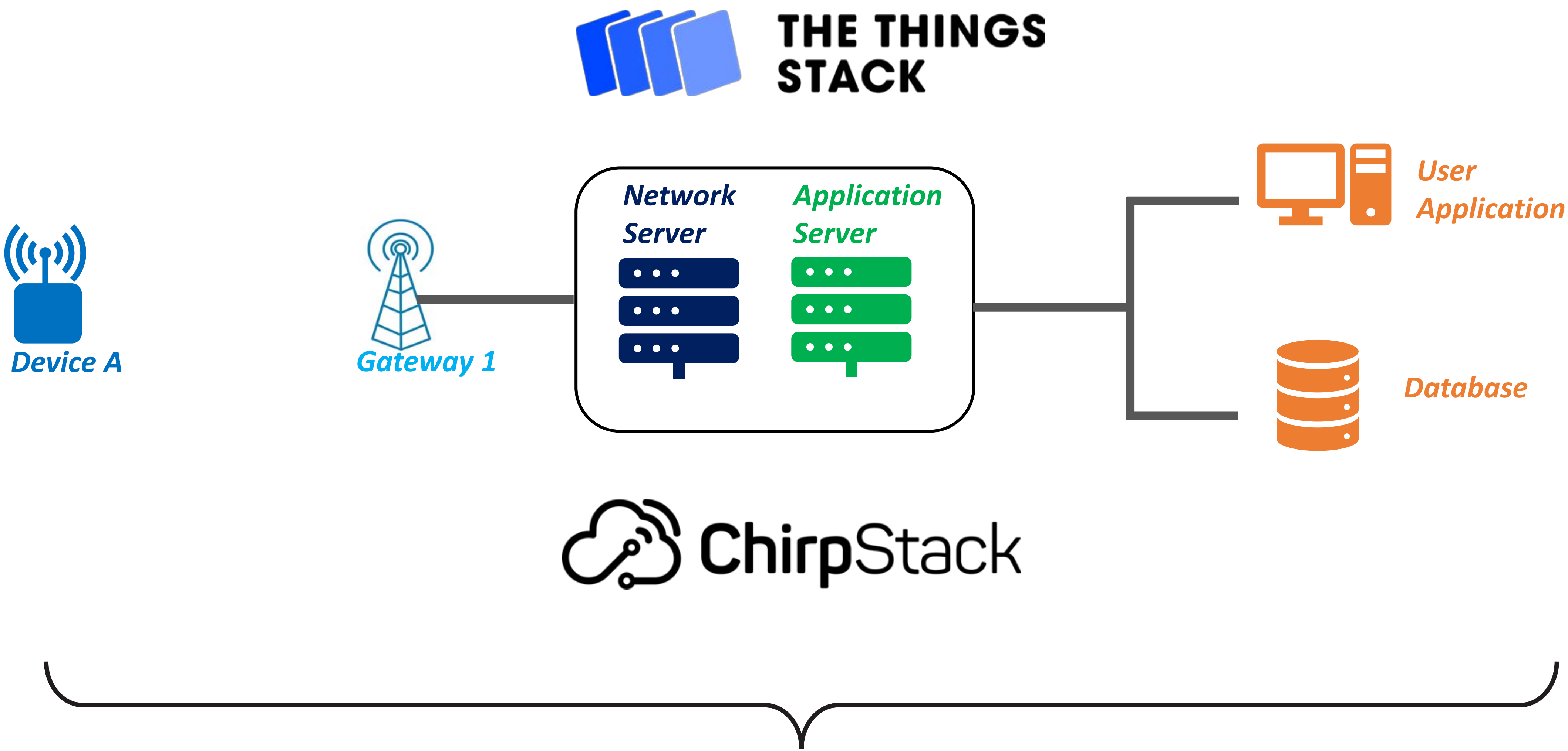
Gateway

Data Rate

Class A

End-device


THE THINGS STACK
Device A
Gateway 1
Network Server
Application Server
User Application
Database
ChirpStack



Complete IoT solution with LoRaWAN

LoRa and LoRaWAN® for the Internet of Things

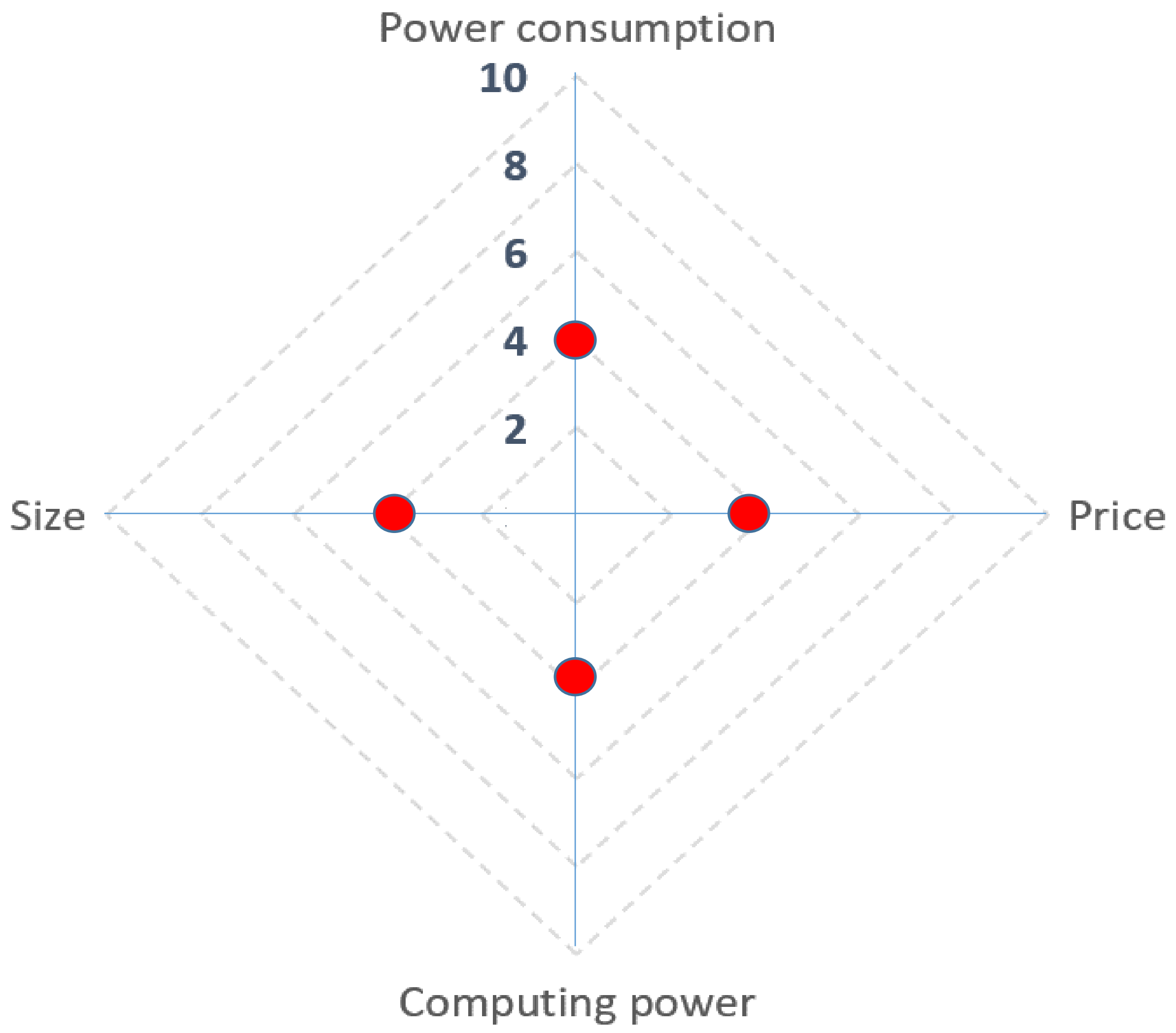
Thank you for choosing this course

Feel free to ask any question!

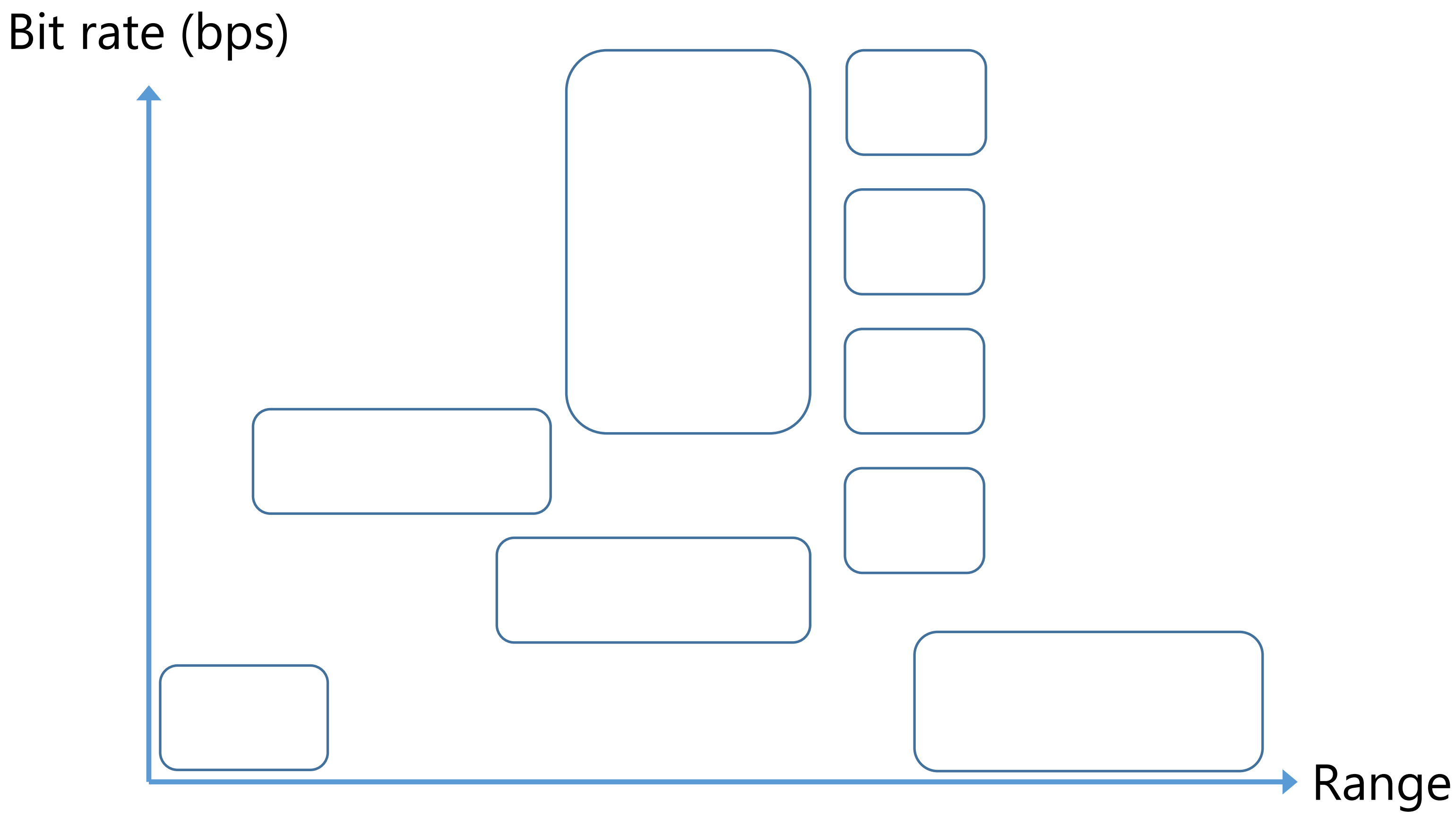
Embedded systems
&
Internet of Things

- ✓ Characteristics of embedded systems
- ✓ The protocols used in IoT
- ✓ How can thousands of devices can transmit at the same frequency

Embedded systems and IoT



Protocols used in IoT



The free frequency bands

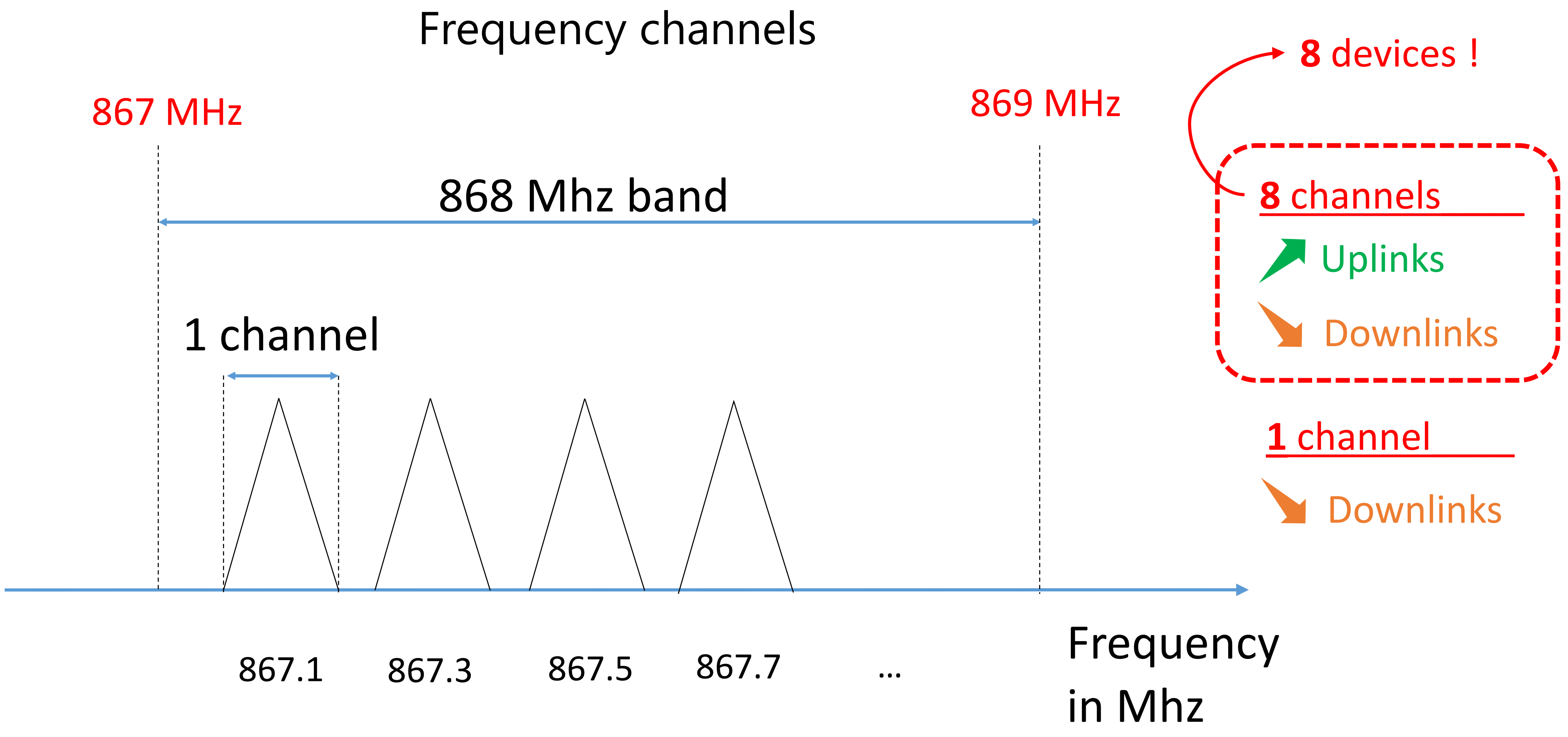
Free bands in Europe:

✓ No need for authorization

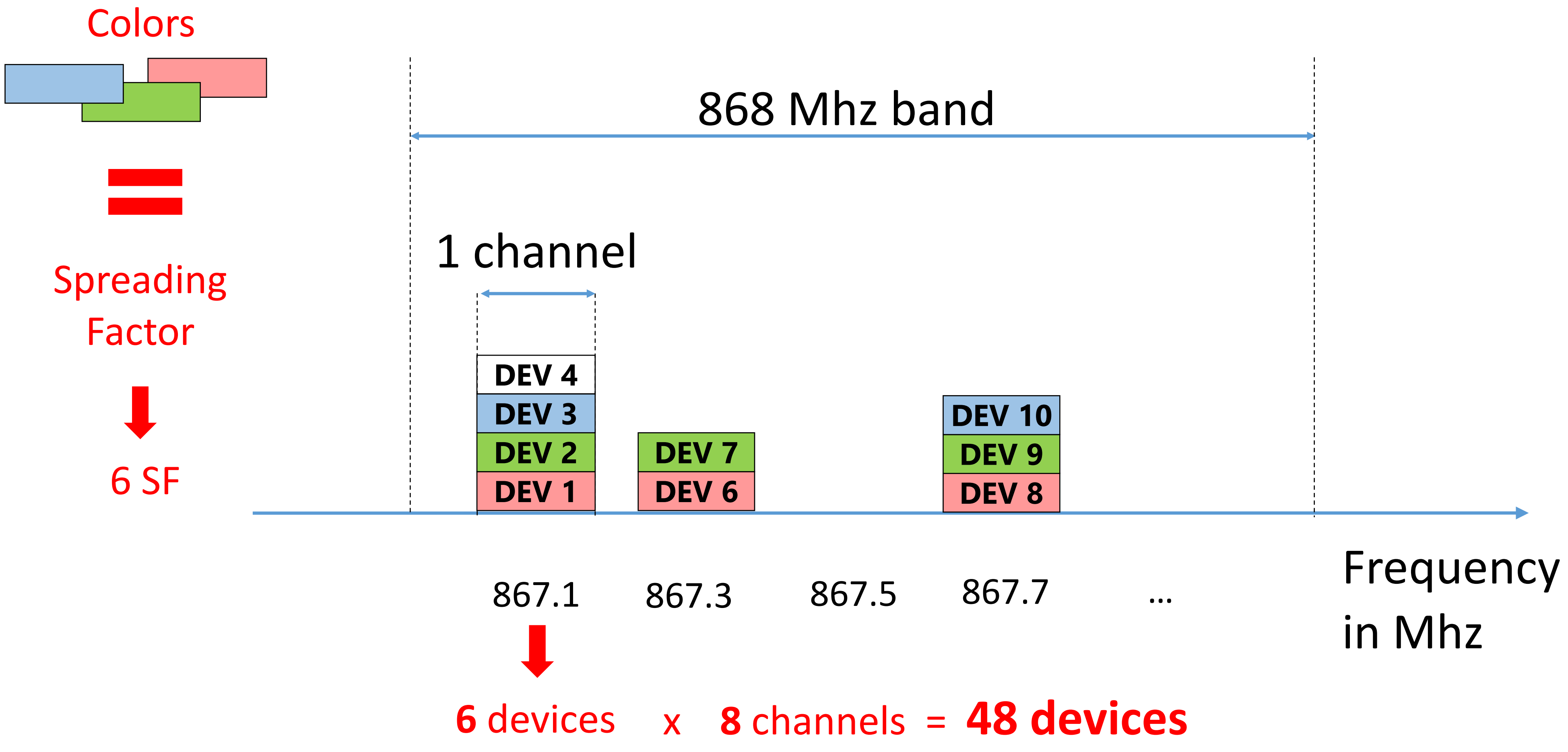
✓ Free of charge

Frequencies	Examples of protocols		
13.56 Mhz	NFC		
433 MHz	LoRa		
868 MHz	Sigfox, LoRa		
2.4 Ghz	WiFi	BLE	Zigbee
5 Ghz	WiFi		

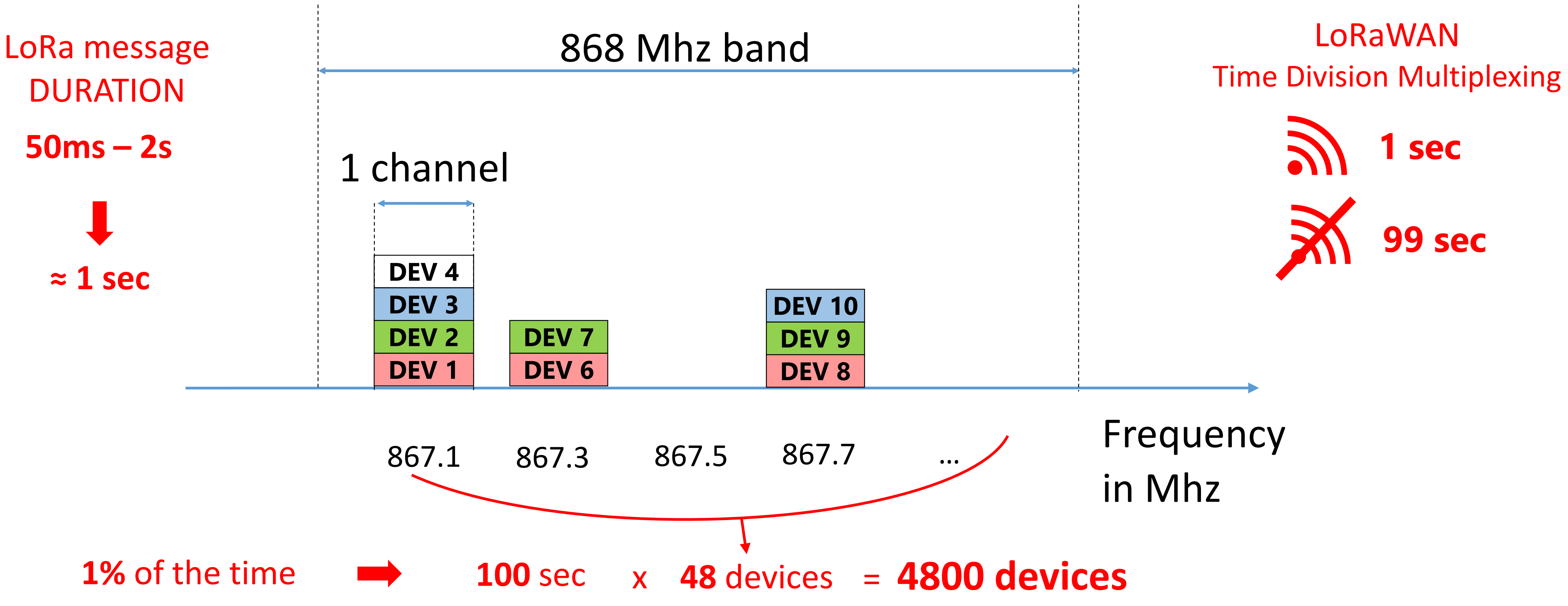
Frequency Division Multiplexing



Spread Spectrum



Time Division Multiplexing



LET'S RECAP

- ❑ LoRa is a "**long distance**" and "**low speed**" protocol
- ❑ It transmits on **channels** in the **868 MHz** band
- ❑ LoRa uses Spreading Factors :

LoRa ← SF5, SF6, SF7, SF8, SF9, SF10, SF11 and SF12 → LoRaWAN

If two LoRa devices are using different SF :

> They can transmit at the same time on the same channel

Radio Transmission and propagation

- ✓ Units and definitions: Decibel, dBm, RSSI, Sensitivity and SNR
- ✓ Examples and studies of technical documentation

Units and definitions : Decibel - 1

dB: **Ratio between two powers**

An <u>attenuation</u> → represented by a <u>negative number</u> (**-**).

A <u>gain</u> → represented by a <u>positive number</u> (**+**).

$$P_T \longrightarrow P_R$$

dB ?

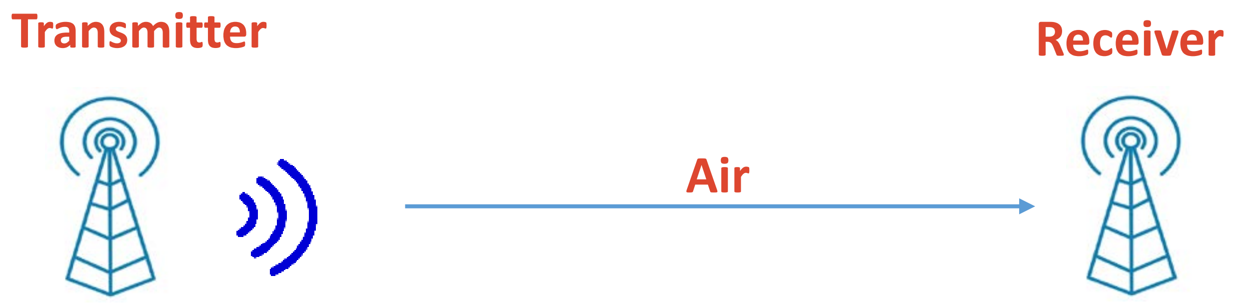
⬇

$$G_{dB} = 10\log(\frac{P_R}{P_T})$$

Power ratio in dB	Power ratio	
10 dB	Multiplication by 10	$P_R = 10\ P_T$
3 dB	Multiplication by 2	$P_R = 2\ P_T$
0 dB	Equality	$P_R = P_T$
-3 dB	Division by 2	$P_R = P_T/2$
-10 dB	Division by 10	$P_R = P_T/10$

Interest ? × ÷ ⟵ **dB** ⟶ − +

Units and definitions : Decibel - 2



P_T : 90 dB attenuation

P_T : Transmit power

P_R : Received power

P_R is 1 billion times smaller than **P_T**

$$P_T = 10^9 \times P_R = \underbrace{10 \times 10 \times 10 \ldots}_{9\ times} \times P_R$$

$$G_{dB} = \overbrace{10 + 10 + 10 \ldots}^{9\ times} = 90\ dB$$

Power ratio in dB	Power ratio
+ 10 dB	Multiplication by 10
+3 dB	Multiplication by 2
0 dB	Equality
-3 dB	Division by 2
-10 dB	Division by 10

The decibel for power : dBm

dBm : Ratio (in decibel) between a power and the power of 1mW

Power in dBm	Power ratio	
10 dBm	Multiplication by 10	**x10 → 10 mW**
3 dBm	Multiplication by 2	**x2 → 2 mW**
0 dBm	Equality	**1 mW**
-3 dBm	Division by 2	**0,5 mW**
-10 dBm	Division by 10	**0,1 mW**

The walkie-talkie has a transmission power of 2W. What is the transmission power in dBm?

$$P_T = 1mW$$

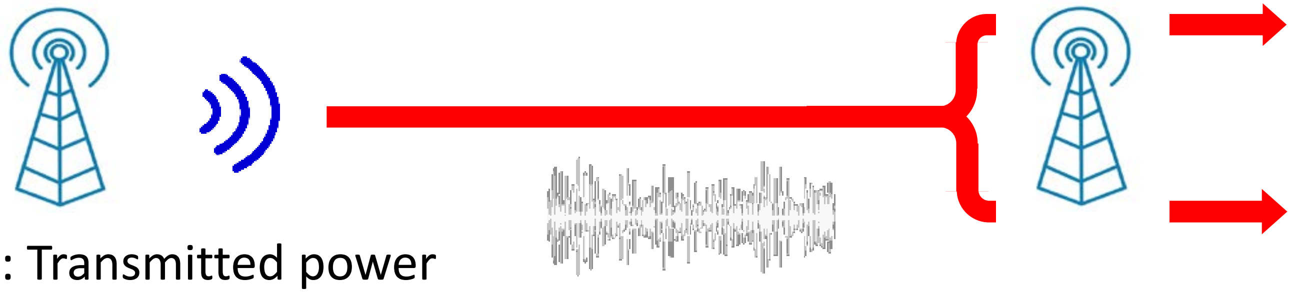
$$P_{T(dBm)} =$$

Definitions: RSSI, Sensitivity, SNR

We want:

Transmitter

Receiver



P_R : Received power $\quad P_R \approx P_T$

P_N : **Noise** power $\quad P_B \approx 0$

P_T : Transmitted power

Noise

dBm

- ❑ **RSSI** (Received Signal Strength Indication): power $\mathbf{P_R}$ of the received signal

dBm

- ❑ **Sensitivity**: minimum RSSI to receive a signal

RSSI > Sensitivity ➡

RSSI < Sensitivity ➡

dB

- ❑ **SNR** (Signal over Noise Ratio): Ratio of P_R and P_N $\quad \mathbf{P_R/P_N}$ ↗

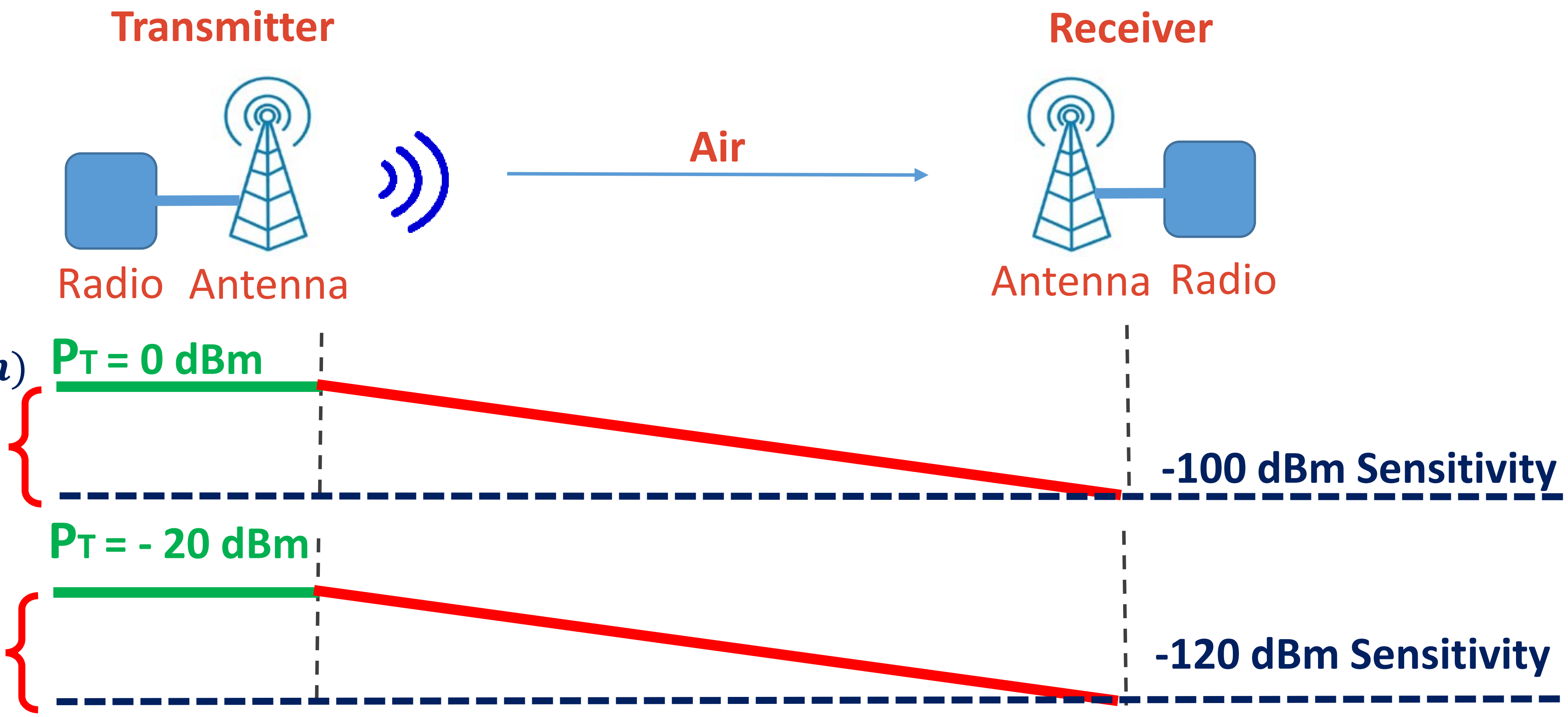
Definitions: Link Budget

LINK BUDGET

$$= 0\ dBm - (-100 dBm)$$

$$= 100\ dBm$$

$$= -20\ dBm - (-120 dBm)$$

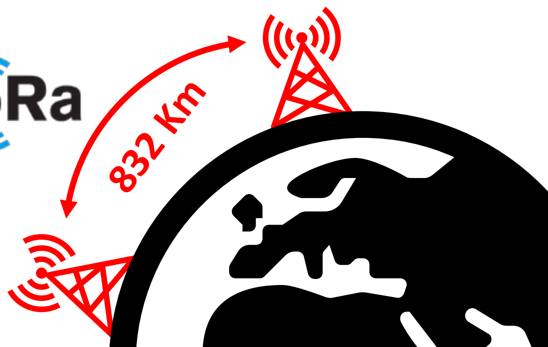


LINK BUDGET

- ❑ **Link Budget**: P_T - Sensitivity
  - ✓ **In LTE (4G):** 130 dB
  - ✓ **In LoRa:** 157 dB

BIG LINK BUDGET ➡ LONG DISTANCE

LoRa

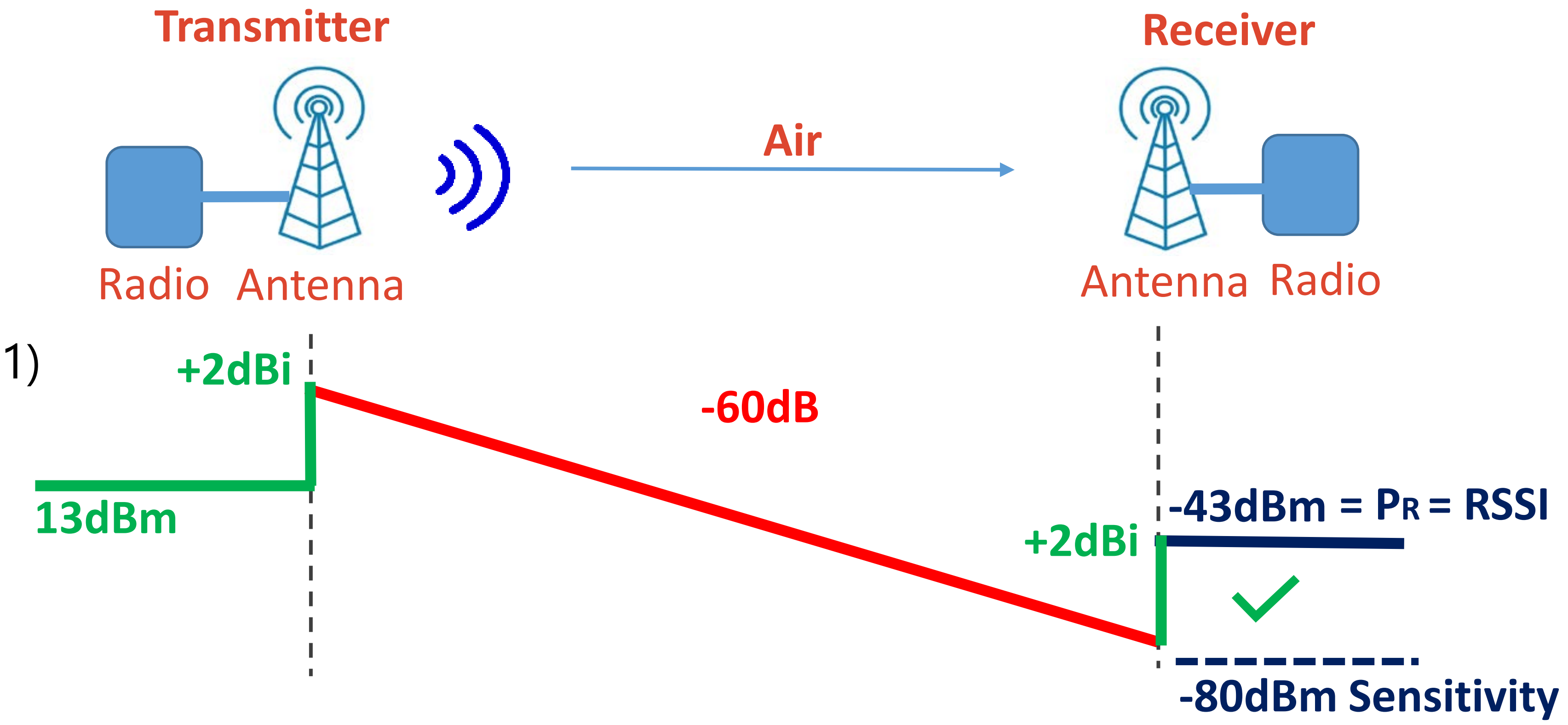


Example - 1

A transmitter uses 13dBm with a 2dBi gain antenna. The air loss is 60dB. Then a 2dBi gain antenna is connected to a receiver with a -80dBm sensitivity.

1) Will the signal be received? ✓ **YES**

2) The noise on the receiver is measured at -50dBm. What is the SNR? ✓ **SNR = 7 dB**

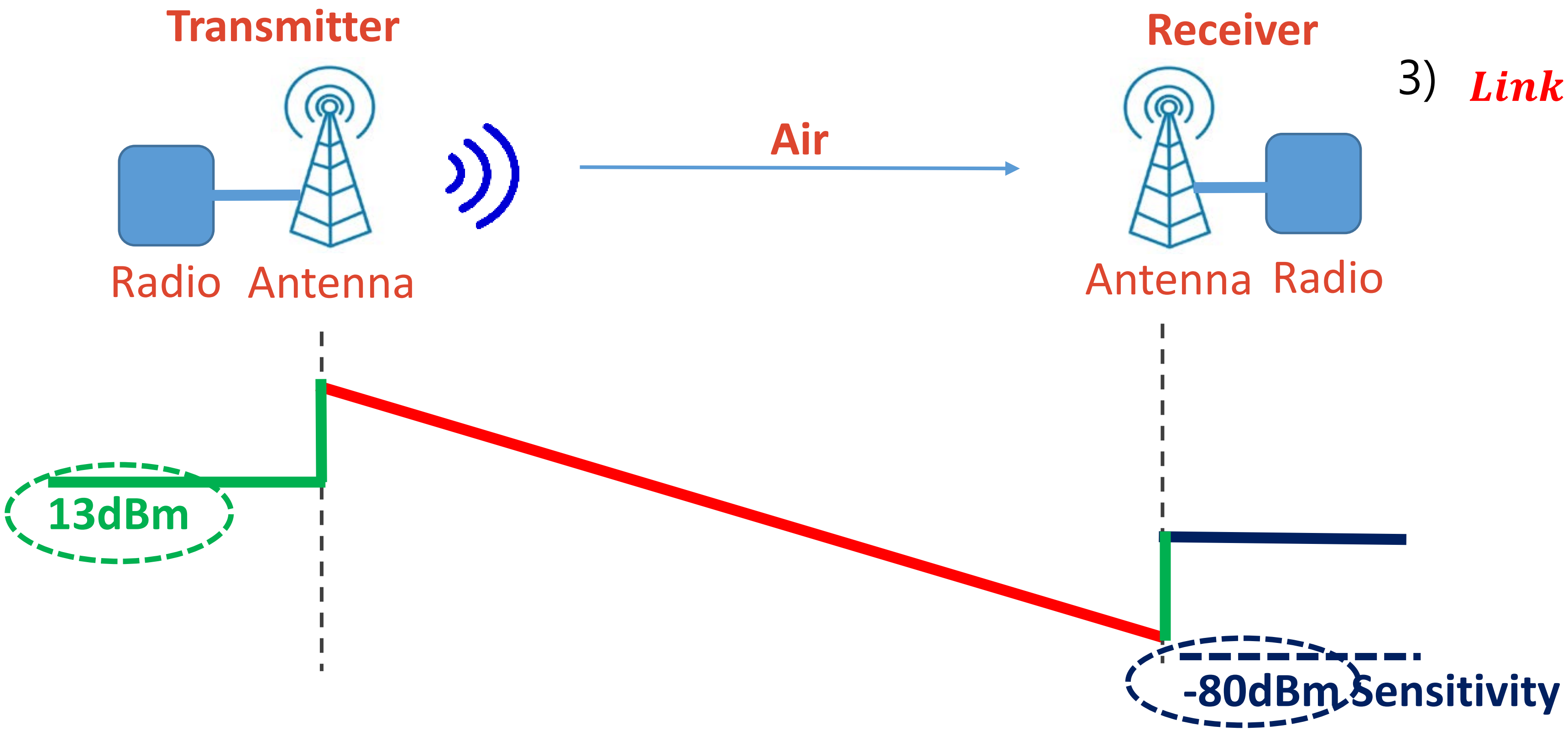


2)
$$SNR_{dB} = P_R - P_N$$
$$= RSSI - (-50)$$
$$= 7dB$$

Example - 2

A transmitter uses 13dBm with a 2dBi gain antenna. The air loss is 60 dB. Then a 2dBi gain antenna is connected to a receiver with a -80dBm sensitivity.

3) What is the link budget ?



3) $\boldsymbol{Link\ Budget = 13\ dBm\ - (-80\ dBm)}$

$\boldsymbol{= 93\ dBm}$

Reception on Gateway


```
"gateways":
    {
      "time": "2020-04-29T12:09:45.563621044Z",
      "channel": 0,
      "rssi": -13,
      "snr": 9.8
    }
```

Example of LoRa transceiver documentation

- ❑ Study of the characteristics of the SX1272
- ❑ Study of the characteristics of the SX1276
- ❑ Spreading Factor and SNR (page 25)

LET’S RECAP

- ❑ The decibel helps to represent ratio (dB) or power (dBm).
- ❑ RSSI is the power received.
- ❑ If the RSSI is above the receiver sensitivity, the transmission is successful.
- ❑ The Link budget evaluate the potential of the transmission.
- ❑ The SNR is the ratio between the signal and the noise.
- ❑ With a high SF, the reception capabilities are improved (sensitivity and SNR)

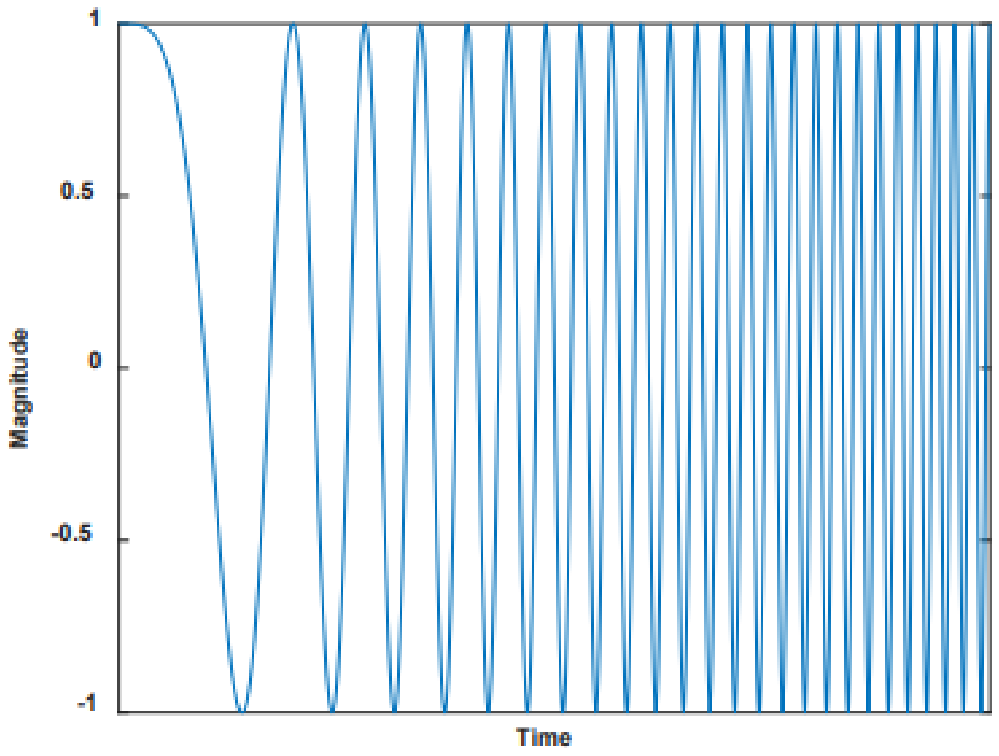
What about the bitrate ?

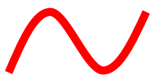
LoRa modulation
Physical Layer

- ✓ The LoRa frame
- ✓ LoRa bitrate: Spreading Factor, Coding Rate, Bandwidth, Time-On-Air and Duty-Cycle
- ✓ The simulation of a LoRa transmission
- ✓ The consumption and the autonomy of a LoRa device

The Chirp (symbol) - 1

Chirp **S**pread **S**pectrum **M**odulation





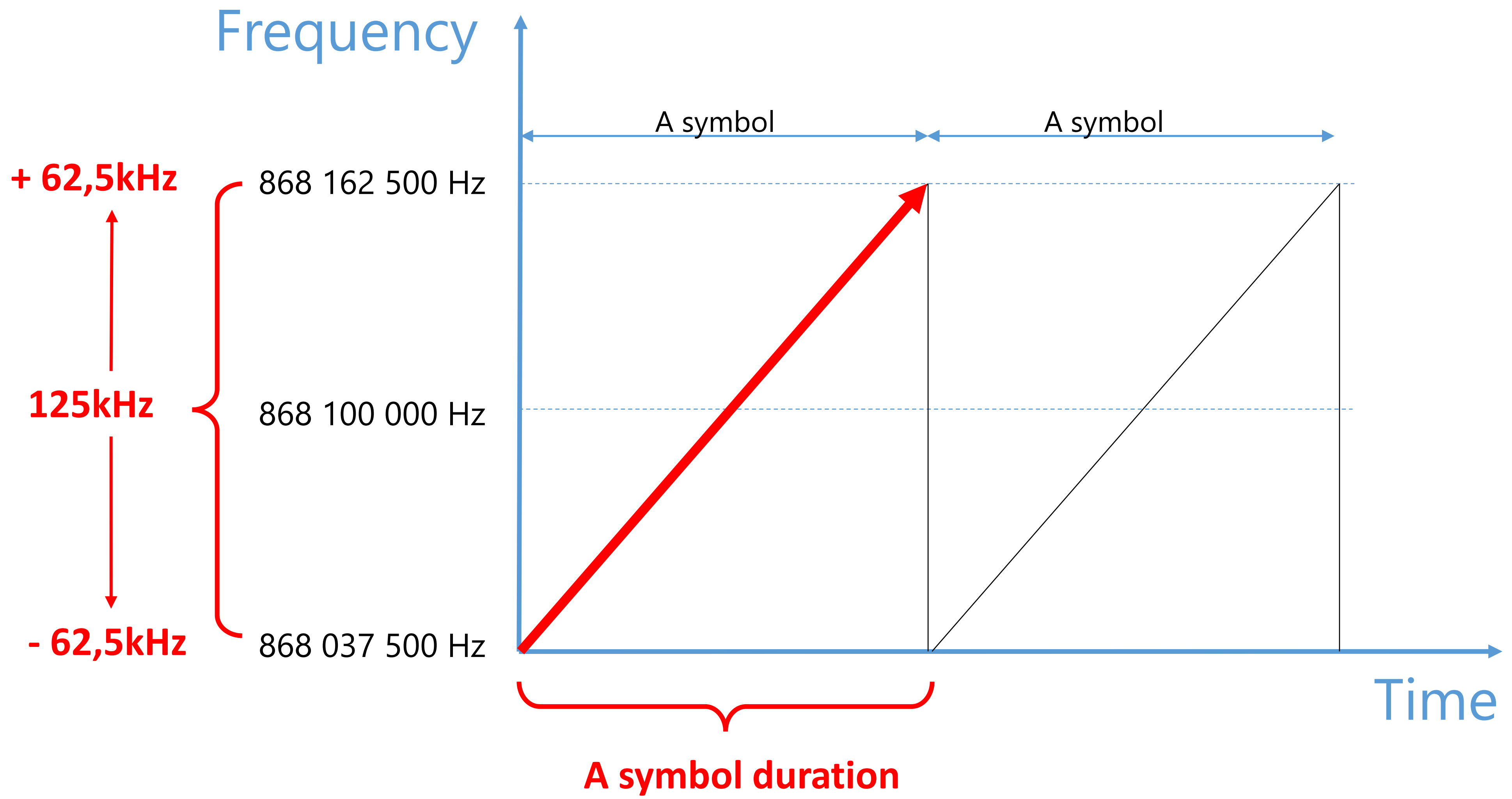
0 to **125 kHz**

or

0 to **500 kHz**

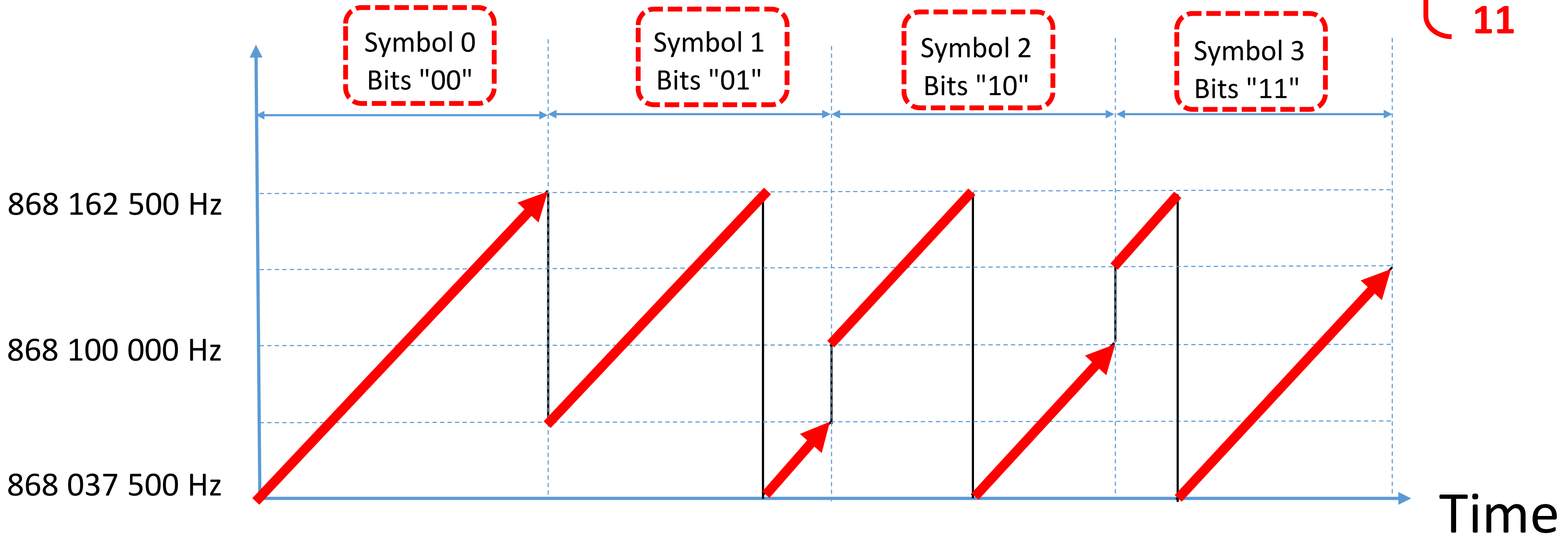
BANDWIDTH
(BW)

The Chirp (symbol) - 2



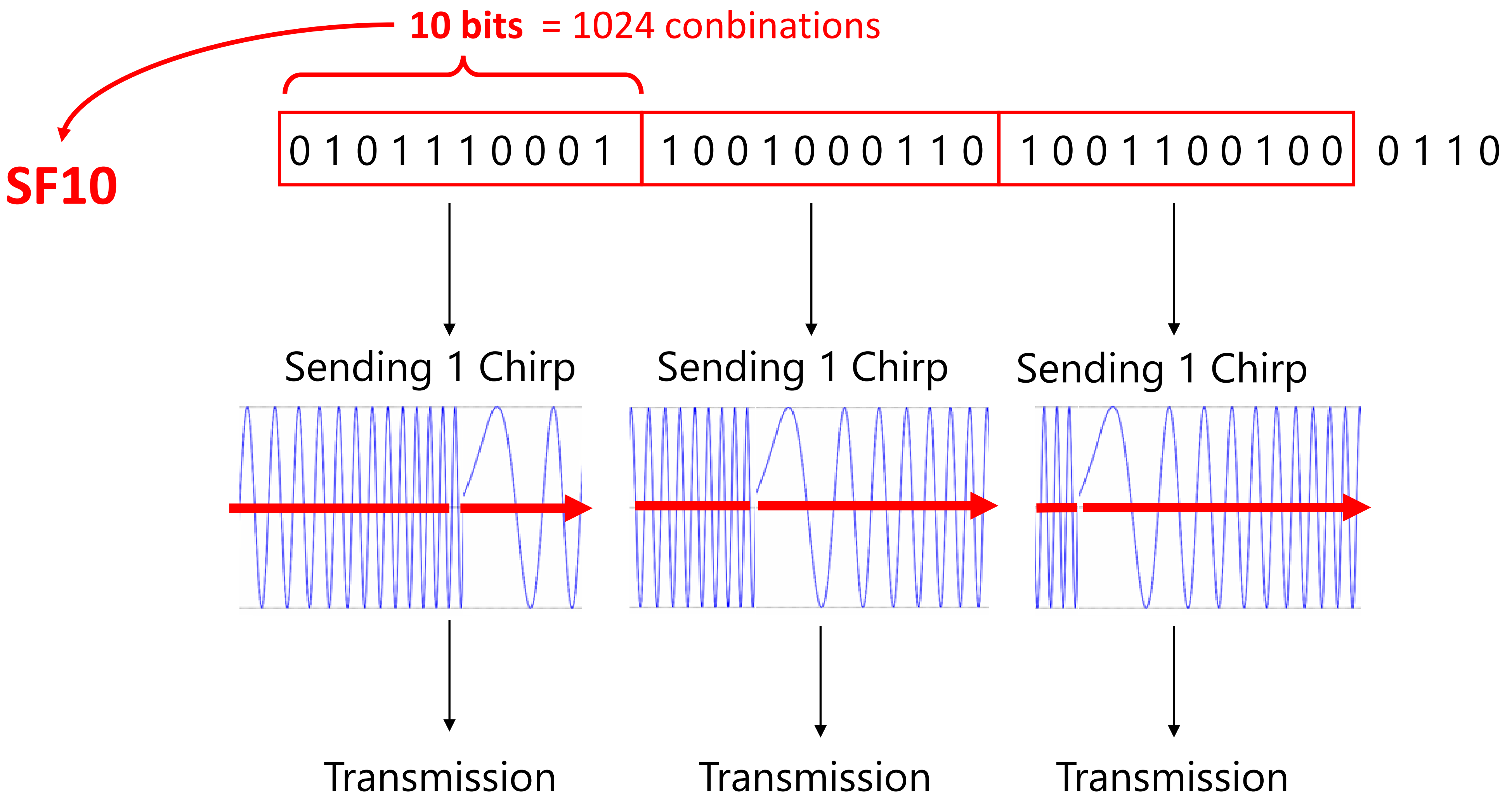
The Chirp (symbol) - 3

Example with 2 bits → 00 01 10 11 → **4 symbols**

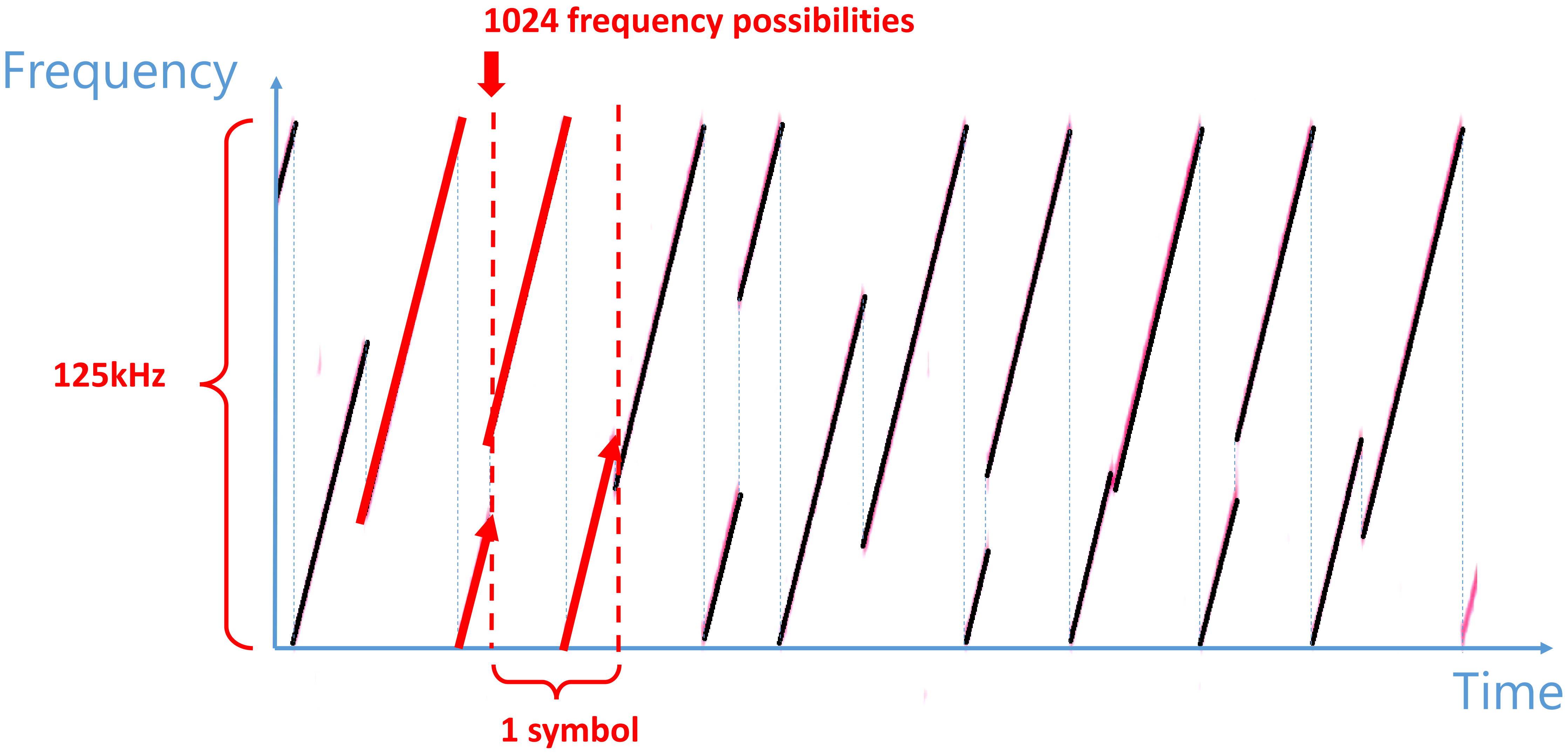


X-bits representation → SFX (5 to 12)

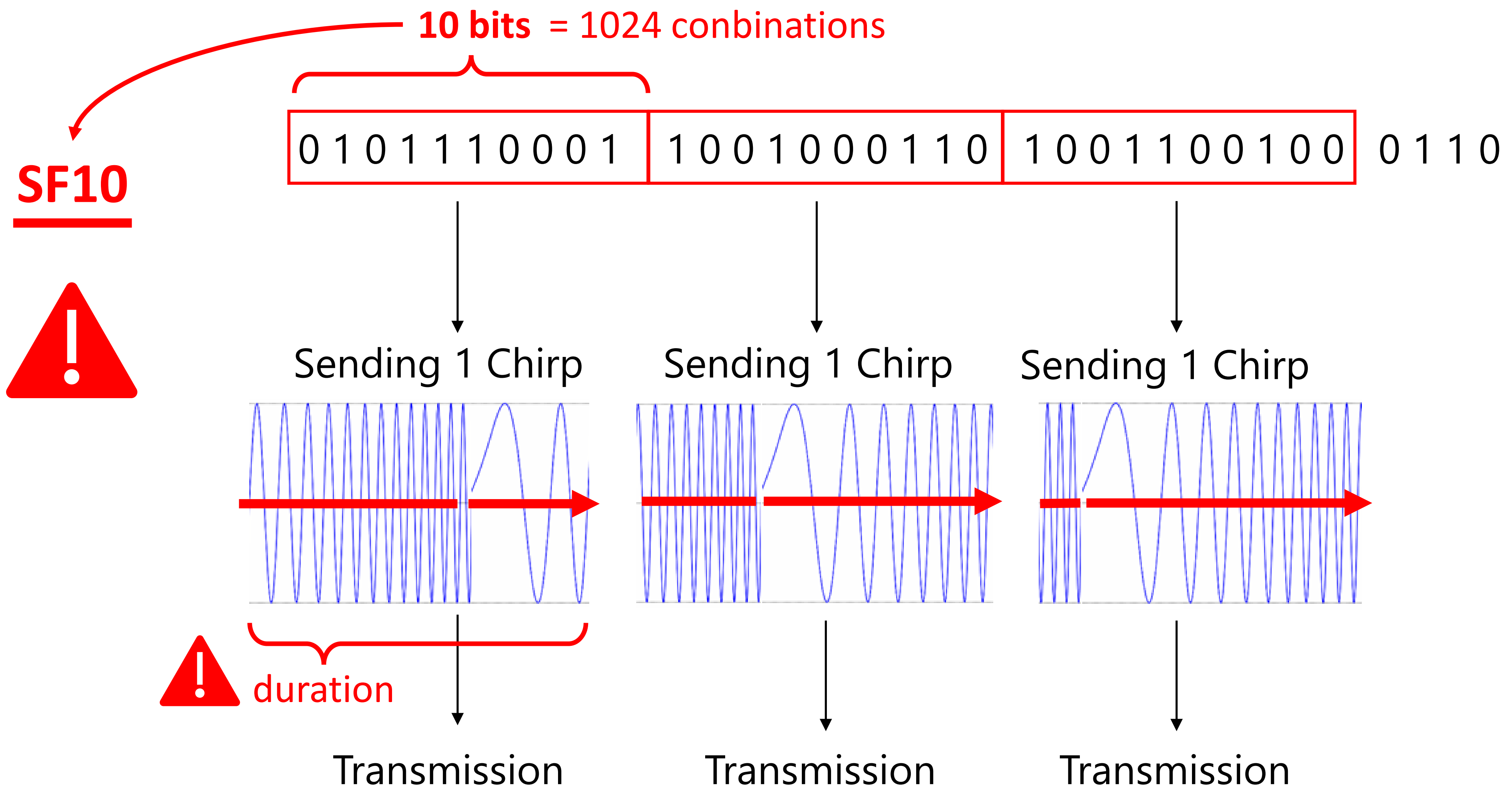
The LoRa frame - 1



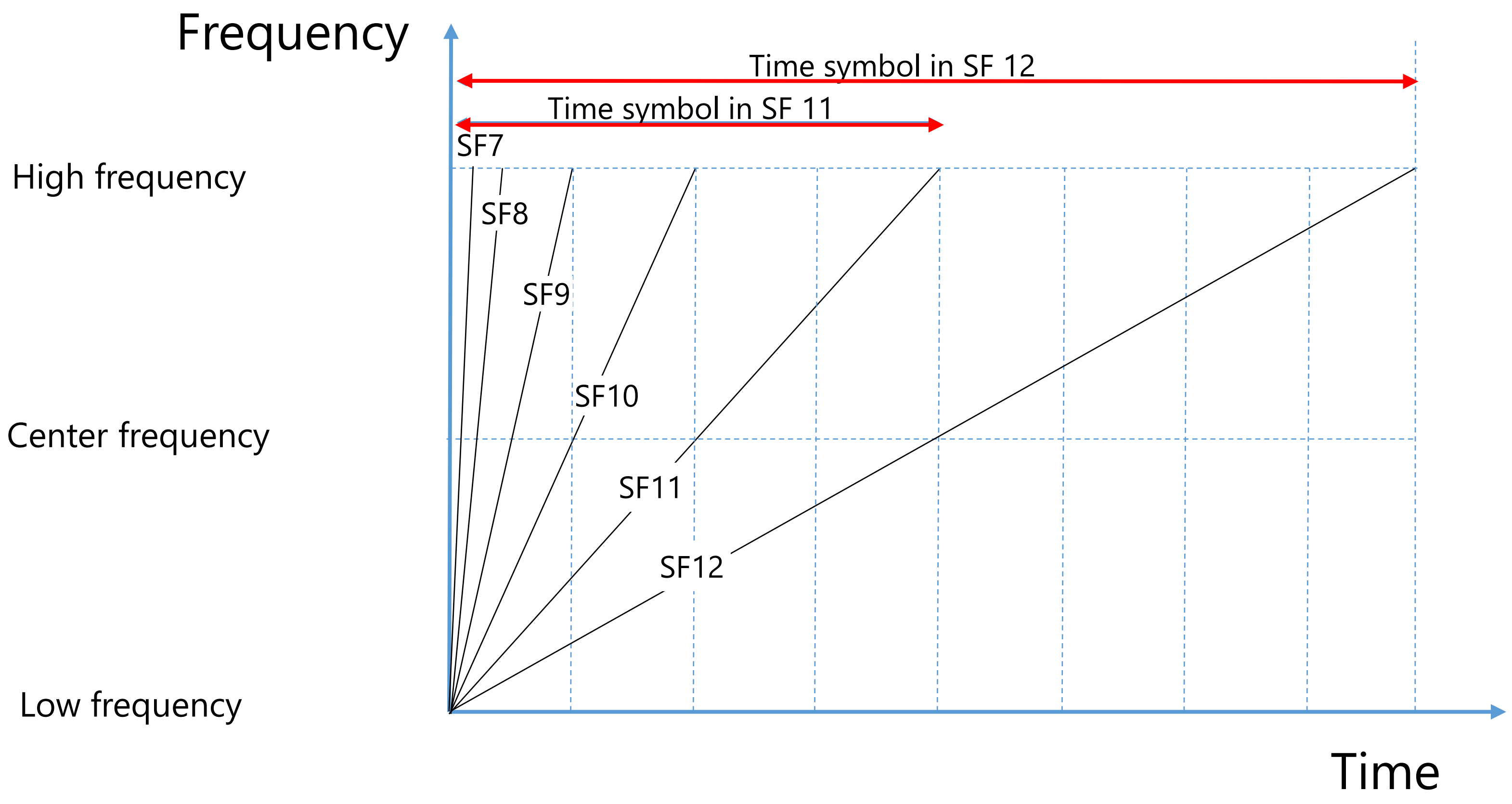
The LoRa frame - 2



The LoRa frame - 1



Symbol duration



$$T_{Symbol.SF12} = 2 \times T_{Symbol.SF11}$$

$$T_{Symbol} = \frac{2^{SF}}{Bandwidth}$$

$$\boxed{Bitrate = SF \times \frac{Bandwidth}{2^{SF}}}$$

LoRa bitrate

- **Case 1:** For SF7, 125 kHz > Bit Rate = $SF\frac{Bandwidth}{2^{SF}} = 7\frac{125.10^3}{2^7} = 6,836\ kbps$

- **Case 2:** For SF12, 125 kHz > Bit Rate = $SF\frac{Bandwidth}{2^{SF}} = 12\frac{125.10^3}{2^{12}} = 366\ bps$

$$SF \nearrow \quad \longleftrightarrow \quad BR \searrow$$

Bit rate

- **Case 1:** For SF7, 125 kHz > Bit Rate = $SF\frac{BW}{2^{SF}} = 7\frac{125.10^3}{2^7} = 6,836\ kbps$

- **Case 2:** For SF12, 125 kHz > Bit Rate = $SF\frac{BW}{2^{SF}} = 12\frac{125.10^3}{2^{12}} = 366\ bps$

$$BR = f(SF, BW, \ldots ?)$$

Coding Rate

$$\boldsymbol{Bit\ Rate = f(SF + BW + \mathbf{CodingRate})}$$

CodingRate (RegModemConfig1)	Cyclic Coding Rate	Overhead Ratio
1	4/5	1.25
2	4/6	1.5
3	4/7	1.75
4	4/8	2

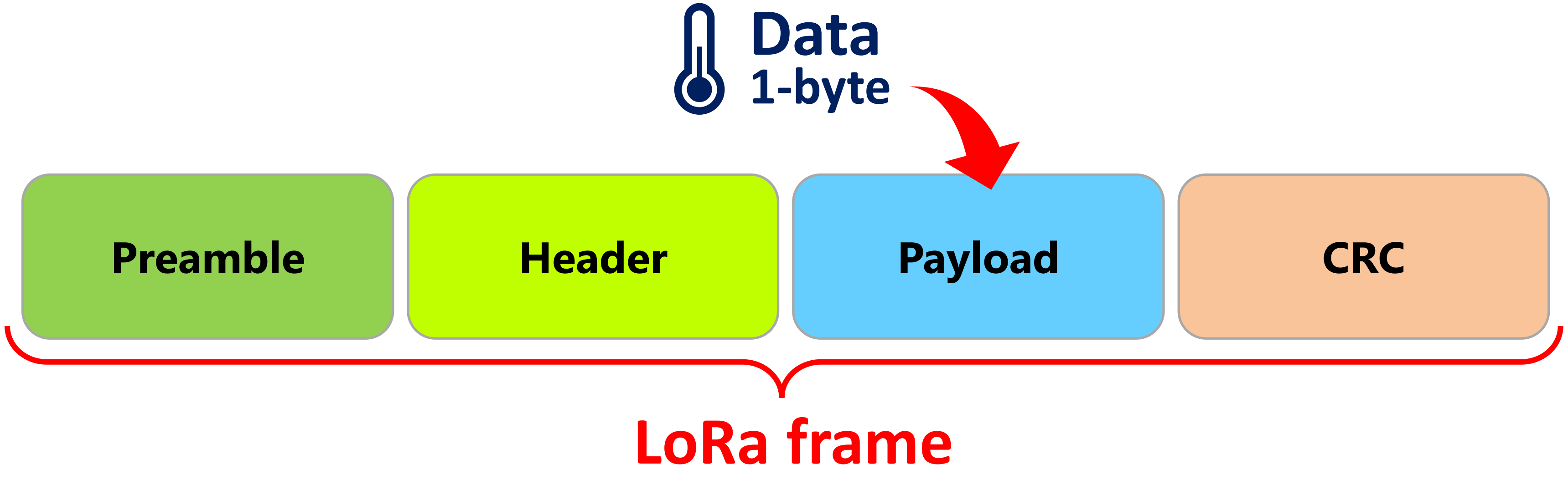
$4 \longrightarrow 5$

$4 \longrightarrow 8$

Additional bits ➡ 👍 Detect and correct errors 👎 Overhead Ratio

- ❑ **Case 1:** For SF7, 125 kHz and CR4/5 > Bit Rate = $\dfrac{\mathbf{6,836\ kbps}}{\mathbf{1,25}} = \mathbf{5469\ bps}$
- ❑ **Case 2:** For SF12, 125 kHz and CR4/5 > Bit Rate = $\dfrac{\mathbf{366\ bps}}{\mathbf{1,25}} = \mathbf{293\ bps}$

LoRa Frame



Previously, Bit Rate was:

$$Bit\ Rate = \frac{SIZE_{payload}}{TIME_{payload}}$$

Real Bit Rate is:

$$Bit\ Rate = \frac{SIZE_{payload}}{TIME_{frame}}$$

TIME ON AIR

LoRa Frame

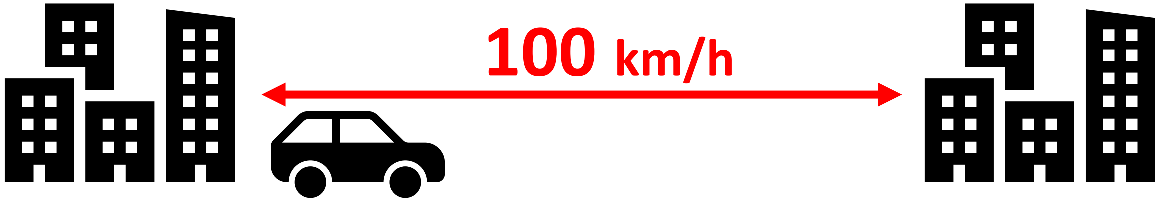
Previously, Bit Rate was:

$$Bit\ Rate = \frac{SIZE_{payload}}{TIME_{payload}}$$

Real Bit Rate is:

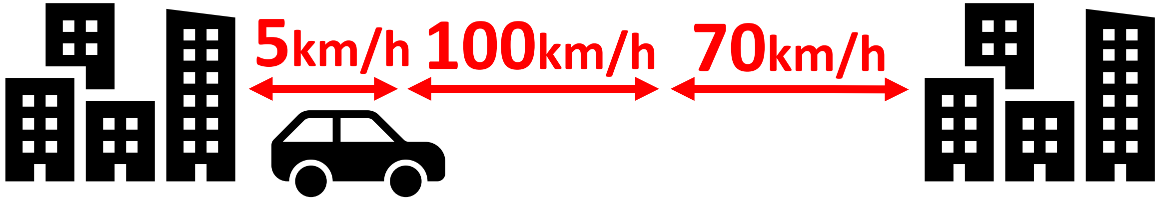
$$Bit\ Rate = \frac{SIZE_{payload}}{TIME_{frame}}$$

TIME ON AIR



Speed capacity

$\neq$



Average speed $= \dfrac{Journey\ distance}{Journey\ time}$

LoRa Frame

Previously, Bit Rate was:

$$Bit\ Rate = \frac{SIZE_{payload}}{TIME_{payload}}$$

Real Bit Rate is:

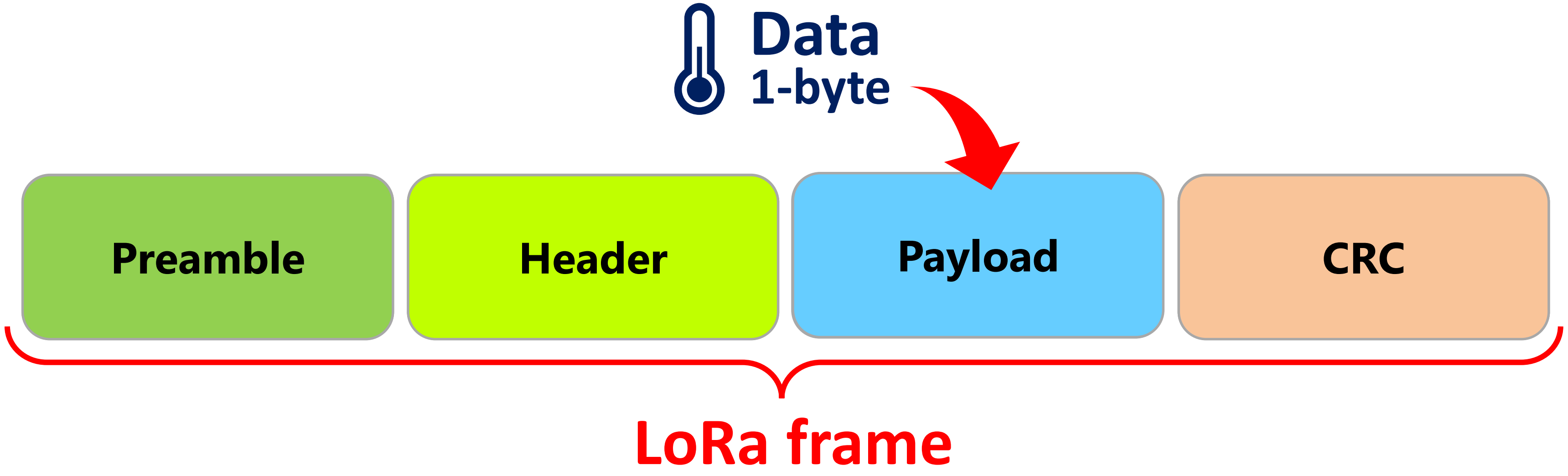
$$Bit\ Rate = \frac{SIZE_{payload}}{TIME_{frame}}$$

TIME ON AIR

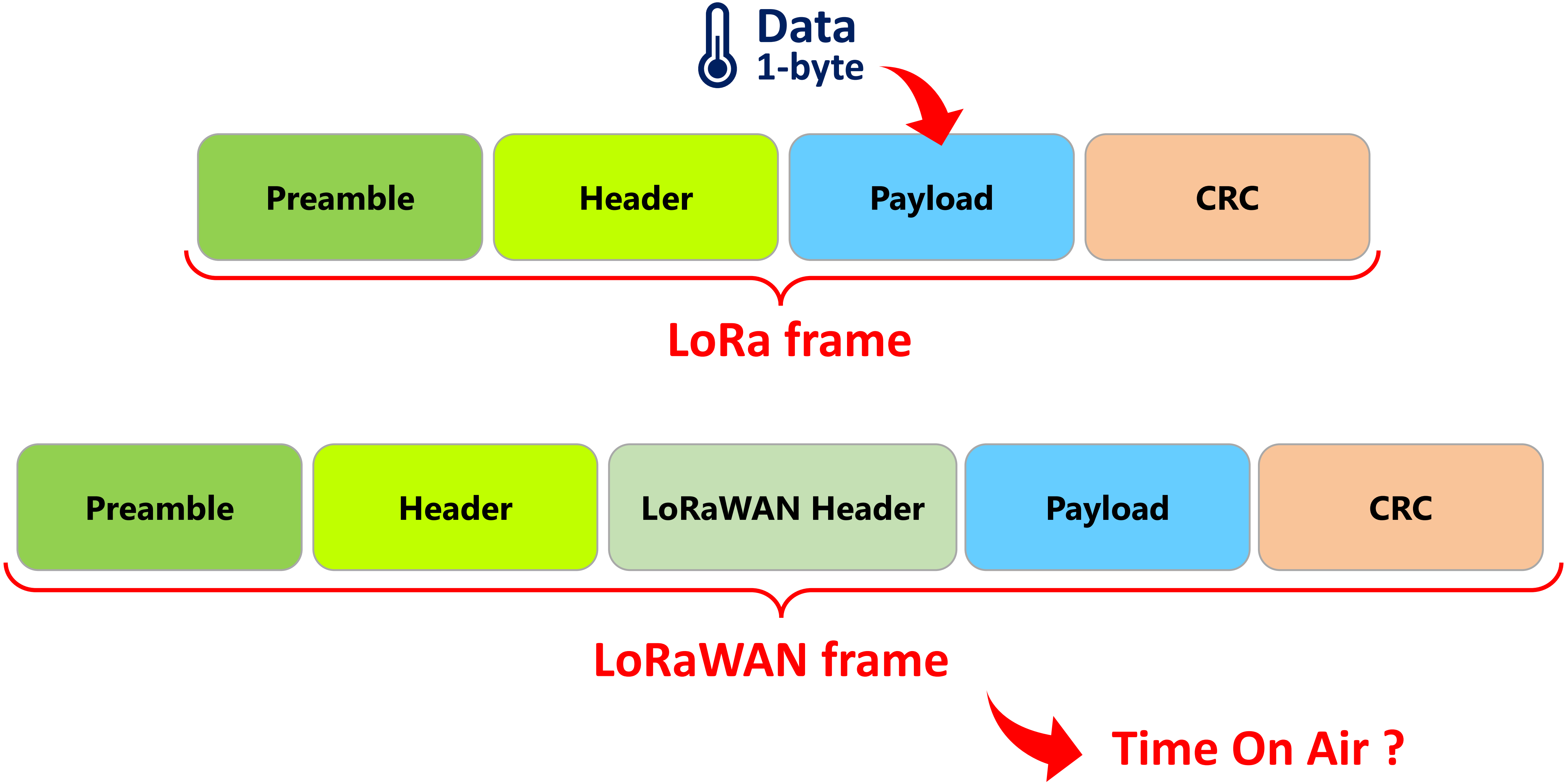
- ❑ **Case 1:** For SF7, 125 kHz and CR4/5 > Bit Rate $= \frac{8}{25,85.10^{-3}} = 309,3\ \mathrm{bps}$

- ❑ **Case 2:** For SF12, 125 kHz and CR4/5 > Bit Rate $= \frac{8}{827,39.10^{-3}} = 9,6\ \mathrm{bps}$

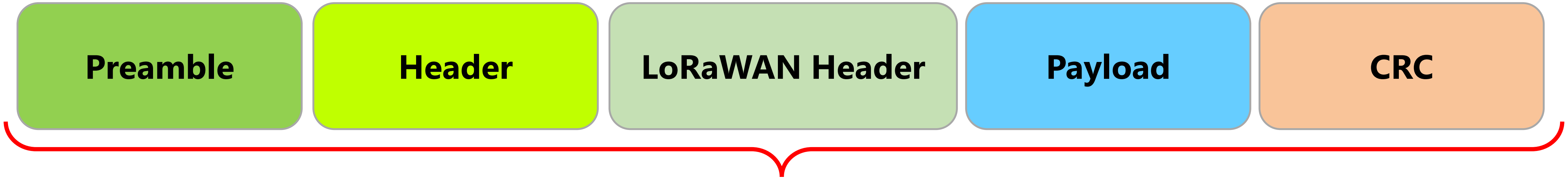
LoRaWAN Frame



LoRaWAN Frame



LoRaWAN Frame



LoRaWAN frame

Time On Air

46,3 ms (SF7)

1155,1 ms (SF12)

- ❑ **Case 1:** For SF7, 125 kHz and CR4/5 > Bit Rate = $\frac{\mathbf{8}}{\mathbf{46,3.10^{-3}}} = \mathbf{172,7\ bps}$

- ❑ **Case 2:** For SF12, 125 kHz and CR4/5 > Bit Rate = $\frac{\mathbf{8}}{\mathbf{1155,1.10^{-3}}} = \mathbf{6,9\ bps}$

Duty-cycle in LoRaWAN (EU868)

LoRa transmission

- **8 channels**
- **Spreading Factor**
- **Limit the transmission time** ➡ **1% of the time**

- ❑ **Case 1:** For SF7, 125 kHz and CR4/5 > Bit Rate $= \frac{\mathbf{172,7}}{\mathbf{100}} = \mathbf{1,73\ bps}$

- ❑ **Case 2:** For SF12, 125 kHz and CR4/5 > Bit Rate $= \frac{\mathbf{6,9}}{\mathbf{100}} = \mathbf{0,07\ bps}$

Energy consumption

- **Time On Air**
  (Spreading Factor, Data, BW, Coding Rate)
- **Collisions**
  (Confirmations, retransmissions)
- **Duty-Cycle**
- **Transmit power**
- **Low power mode**

The LoRaWAN protocol

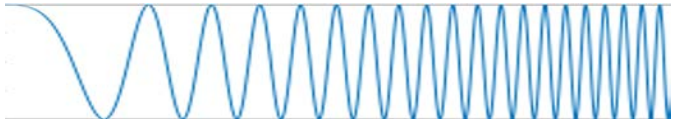
- ✓ LoRaWAN ecosystem
  - Specification
  - LoRa vs LoRaWAN
- ✓ LoRaWAN infrastructure
  - Device
  - Gateway
  - LoRaWAN Server
- ✓ LoRaWAN Security
- ✓ Device classes
- ✓ Activation methods: ABP or OTAA
- ✓ How to …
  - … prevent replay attack?
  - … change the communication parameters?
  - … change Network operator?
- ✓ Optimization of the communication: ADR

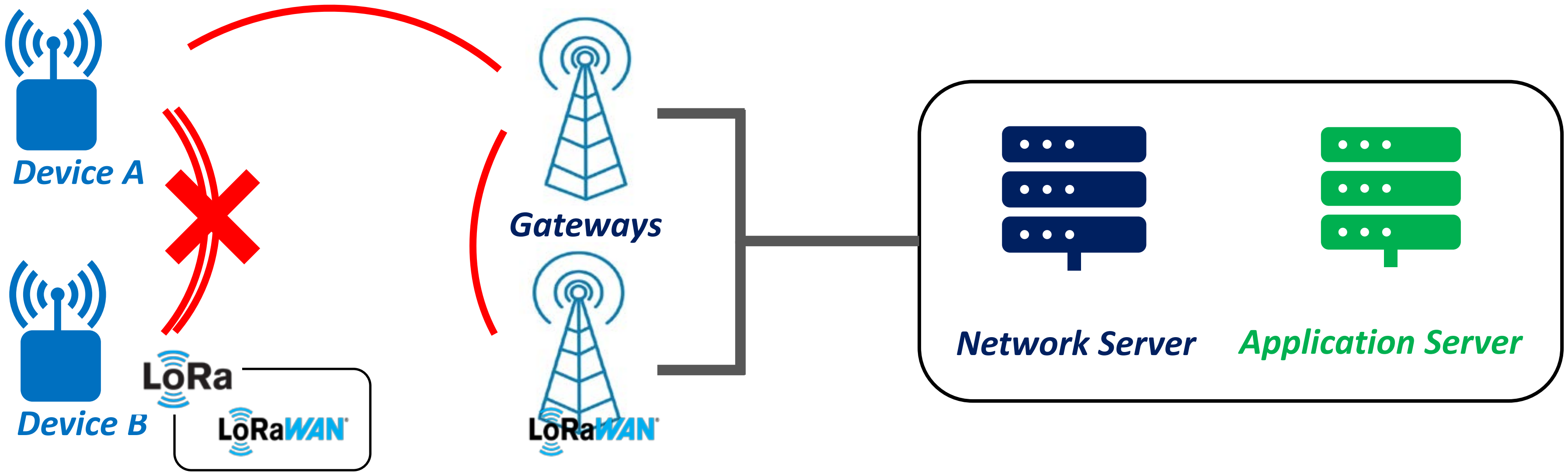
LoRa vs LoRaWAN

LoRa ≠ LoRaWAN®

LoRa vs LoRaWAN

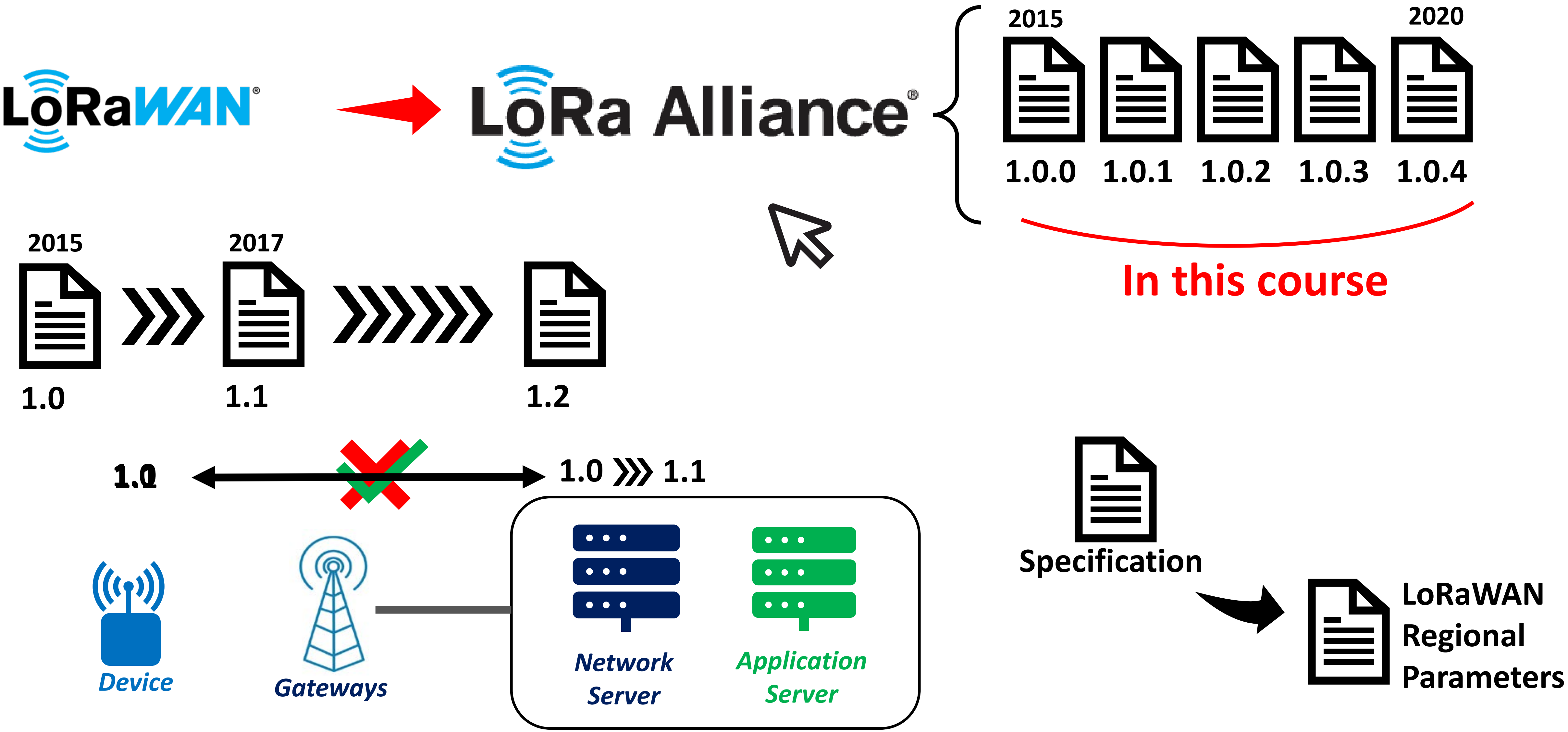
LoRa → **Type of modulation** (Chirp)





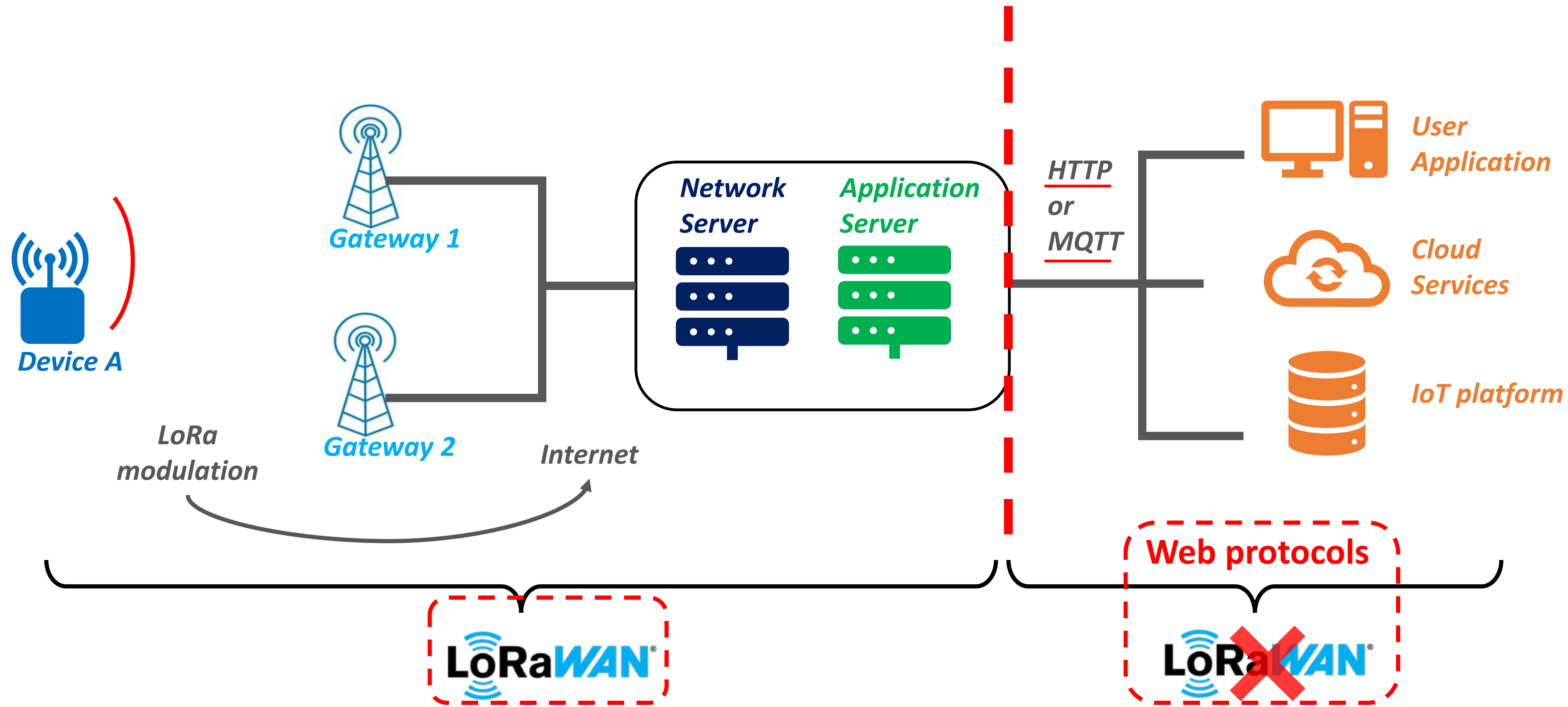
LoRaWAN → **Secured and standardized protocol**

LoRaWAN protocol versions

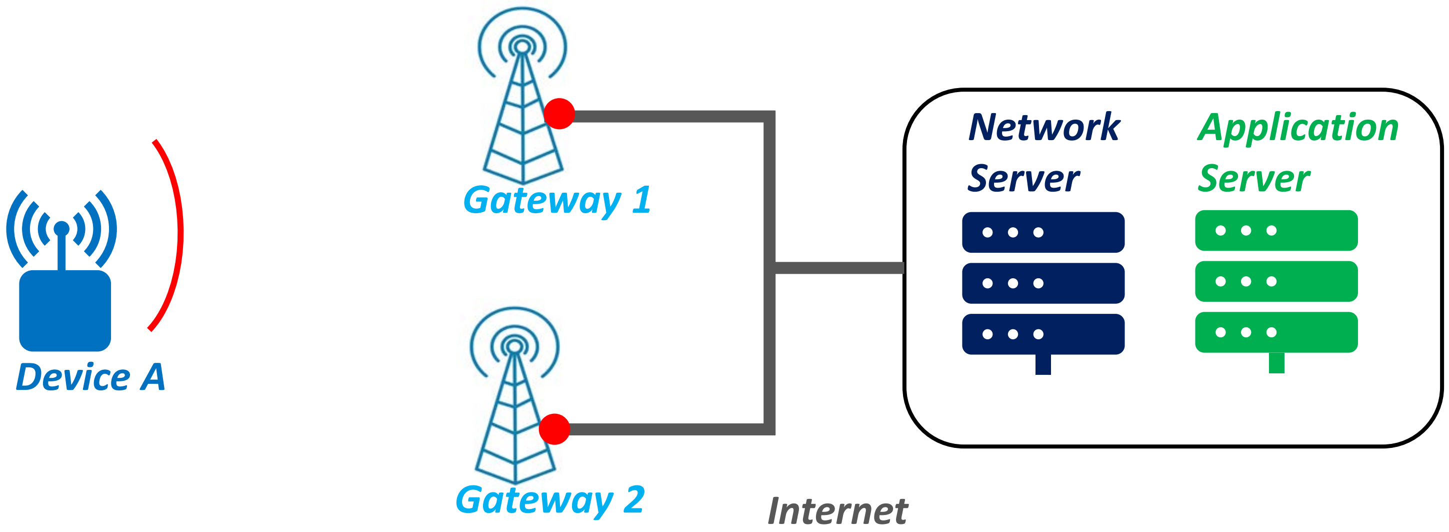


LoRaWAN network infrastructure

LoRaWAN frame = User Payload + Security information



LoRaWAN network infrastructure



LoRaWAN

LoRaWAN Network Server and Application Server

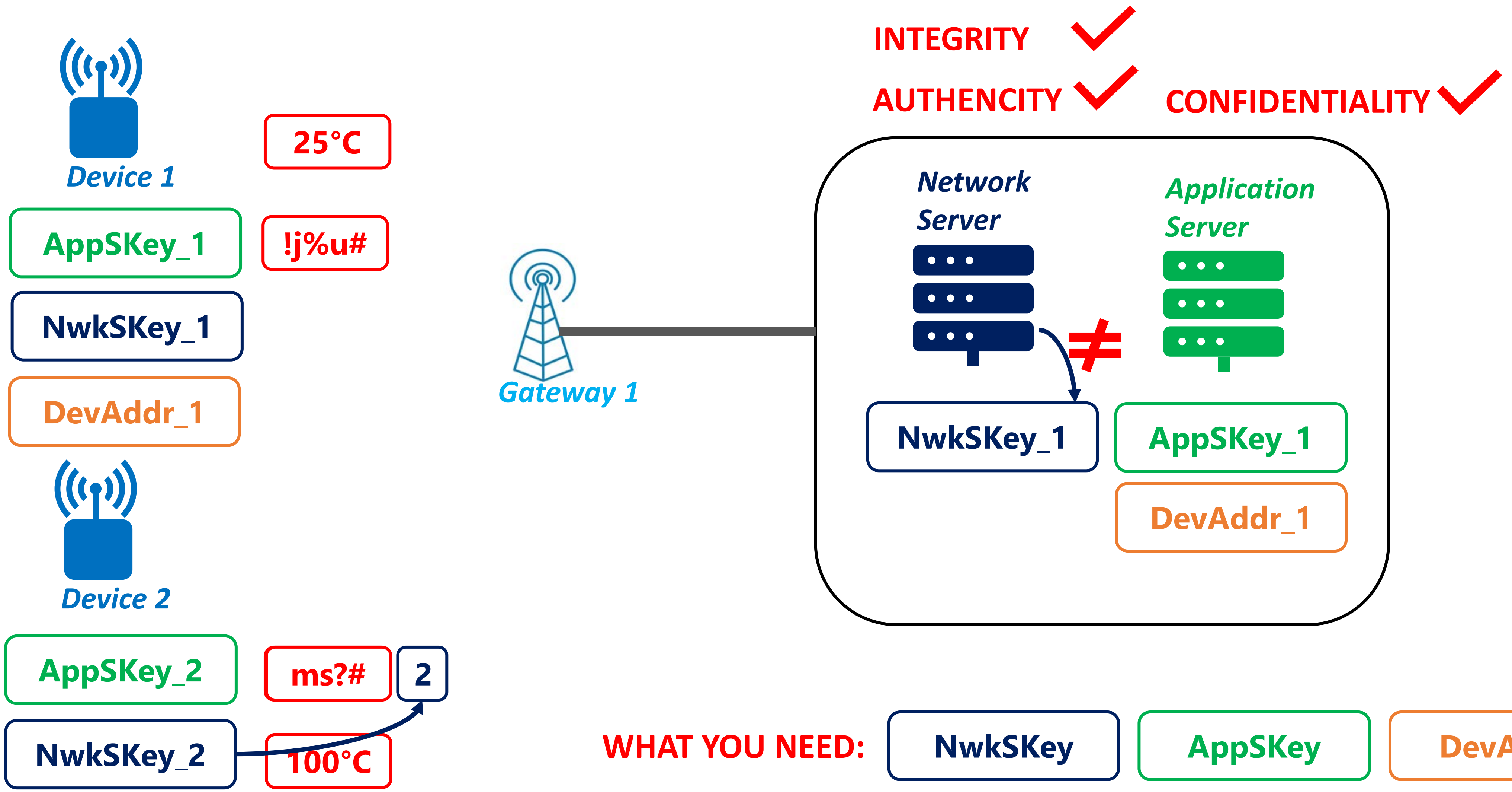
SECURITY

CONFIDENTIALITY
Nobody can understand what the end-device says

INTEGRITY
Nobody can change the transmitted frame

AUTHENCITY
Only allowed end-device can send on the Network Server

LoRaWAN Network Server and Application Server



WHAT YOU NEED: NwkSKey AppSKey DevAddr

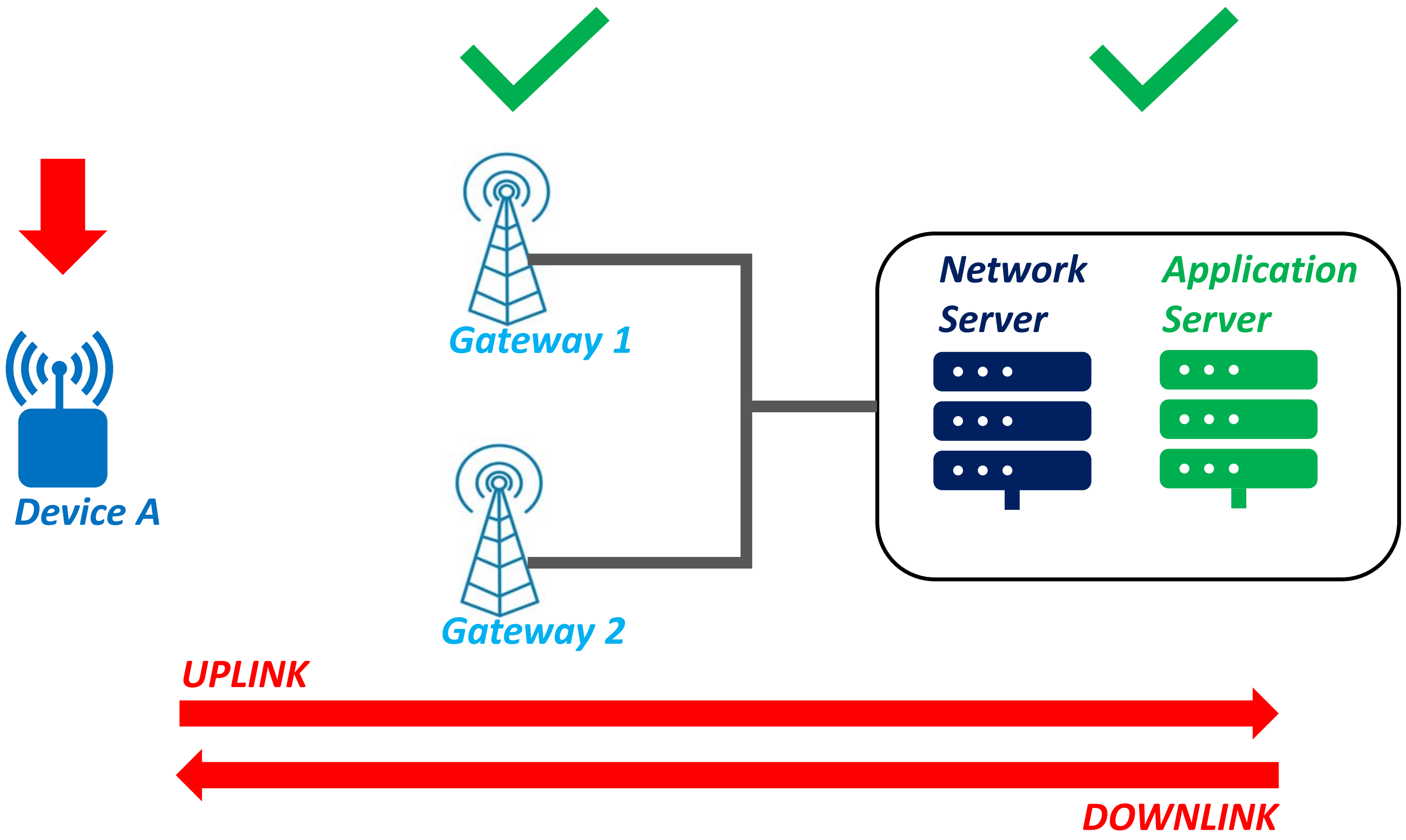
LoRaWAN end-device Classes

Packet transmission time

= TIME ON AIR

1% Time duration

= DUTY CYCLE



Downlink ?

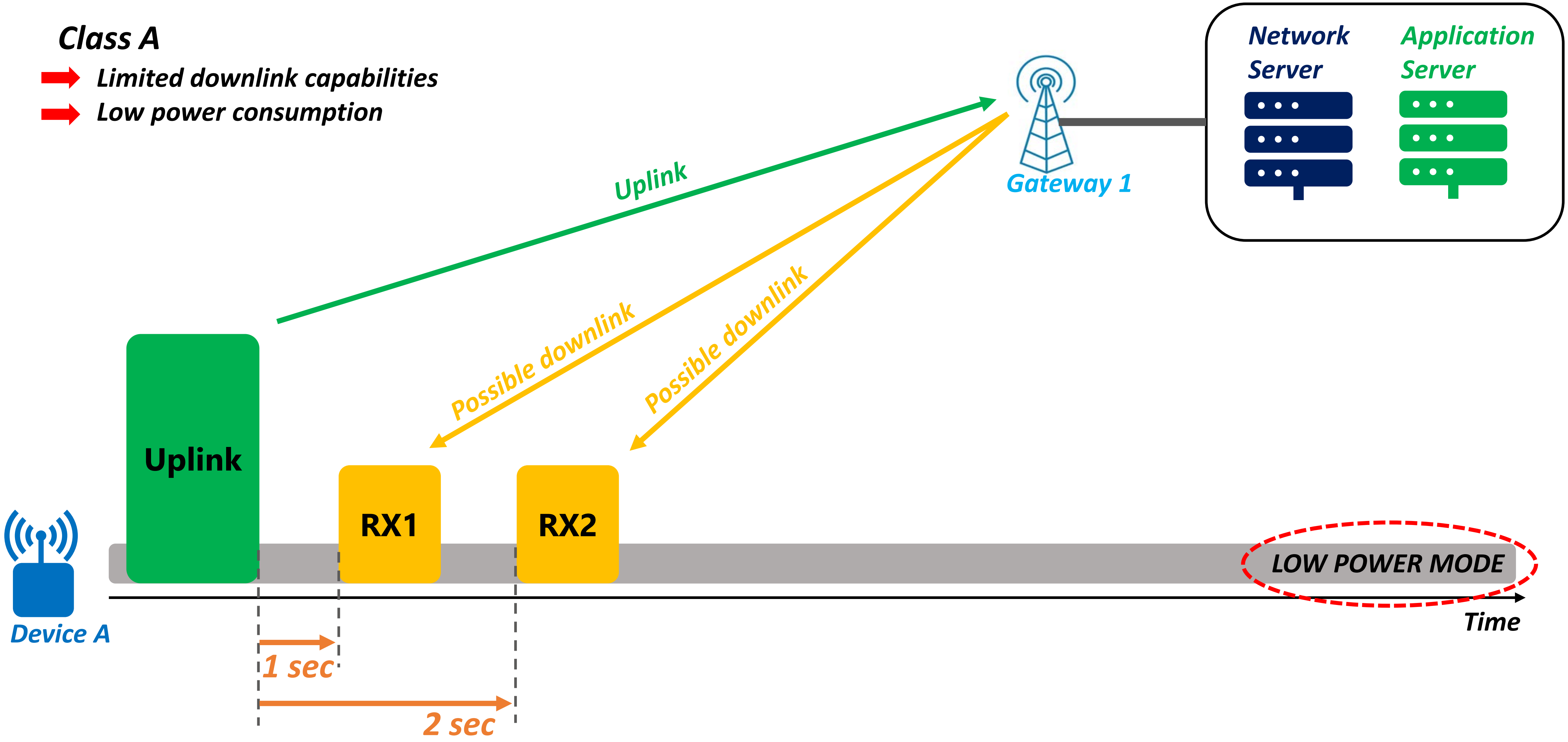
End-device classes

A or B or C

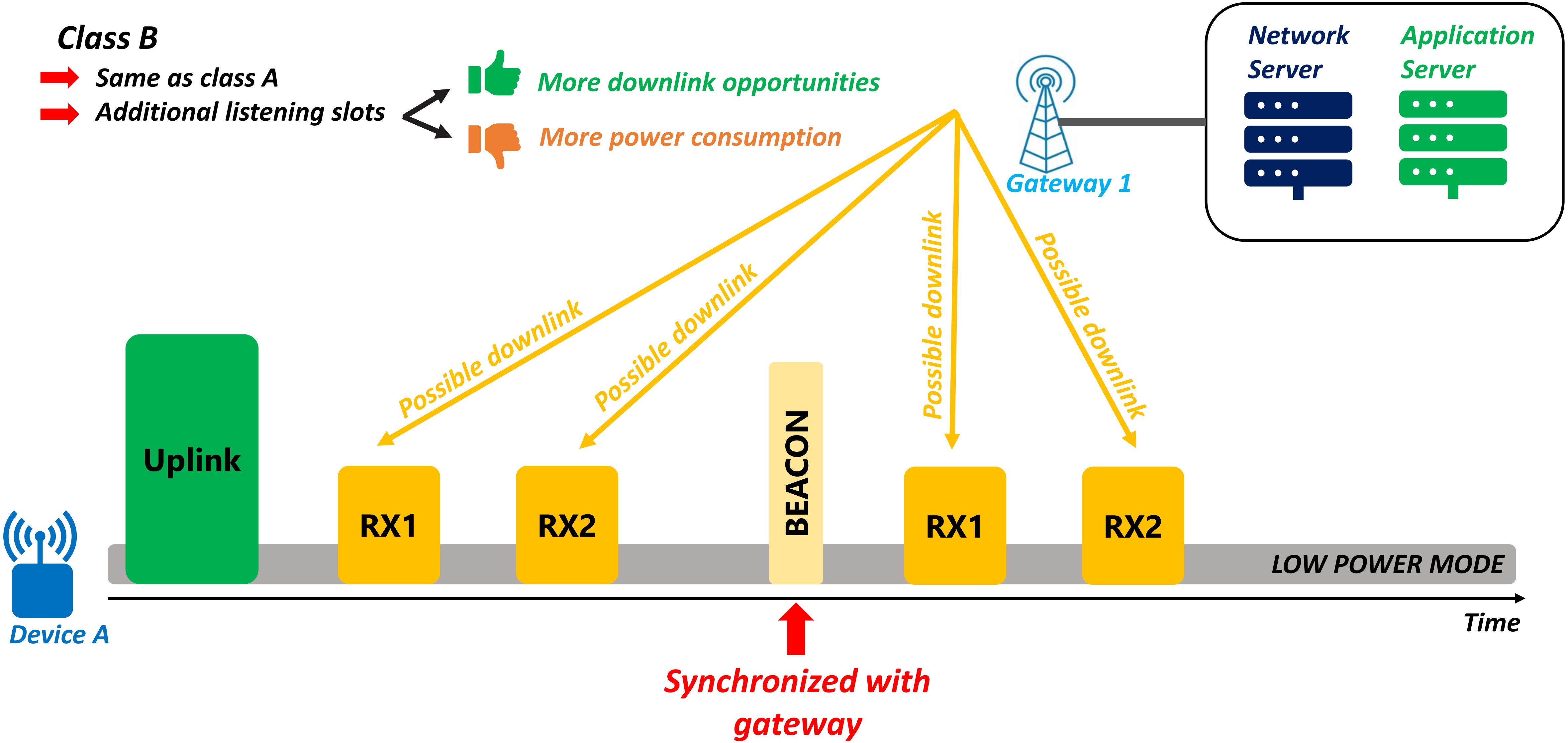
Class A end-devices

Class A

- ***Limited downlink capabilities***
- ***Low power consumption***



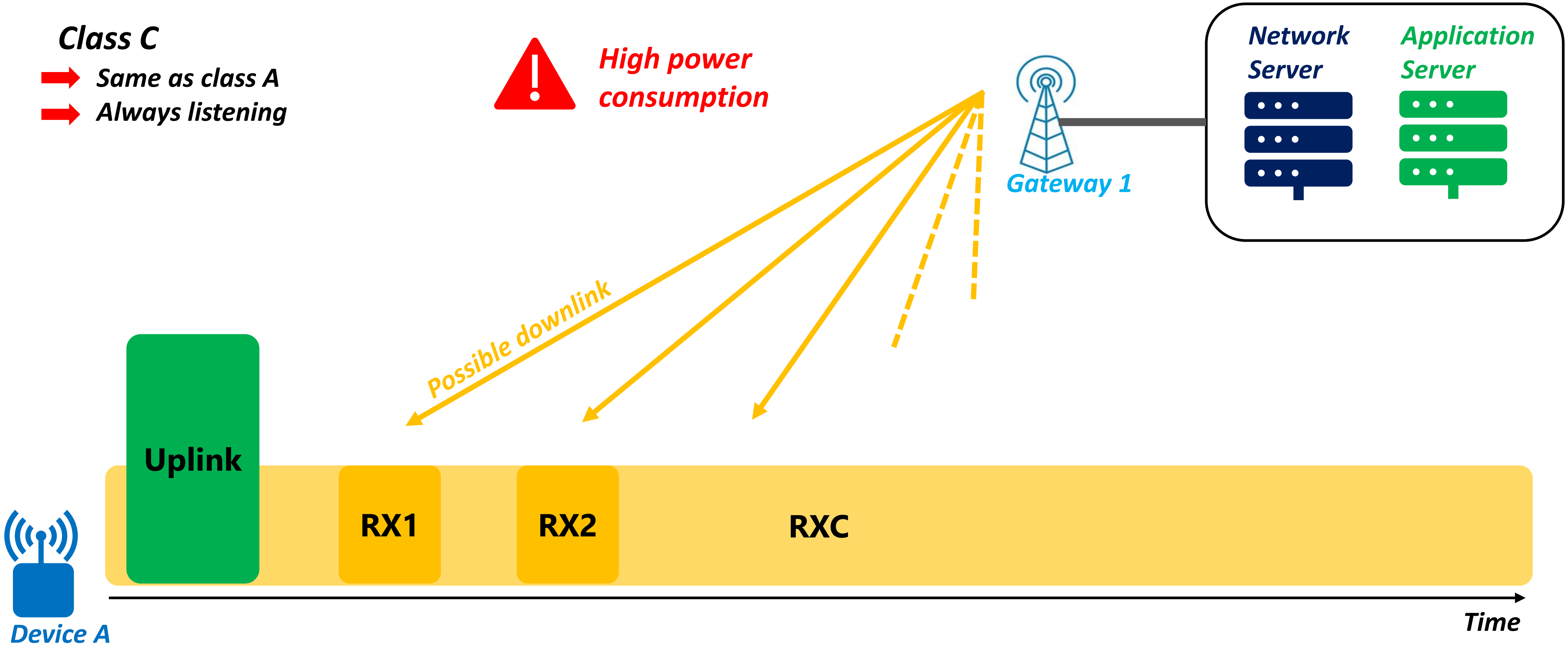
Class B end-devices



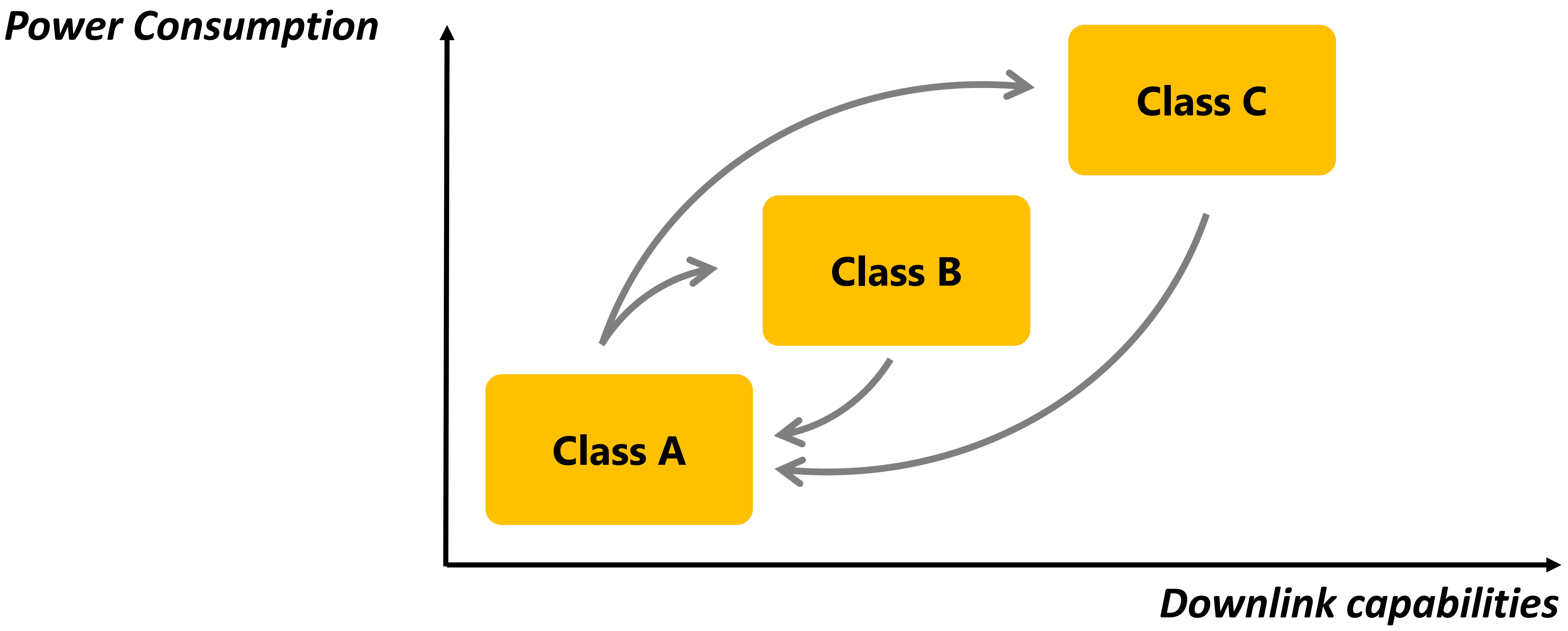
Class C end-devices

Class C

- ***Same as class A***
- ***Always listening***



LoRaWAN end-device Classes



LoRaWAN end-device Classes

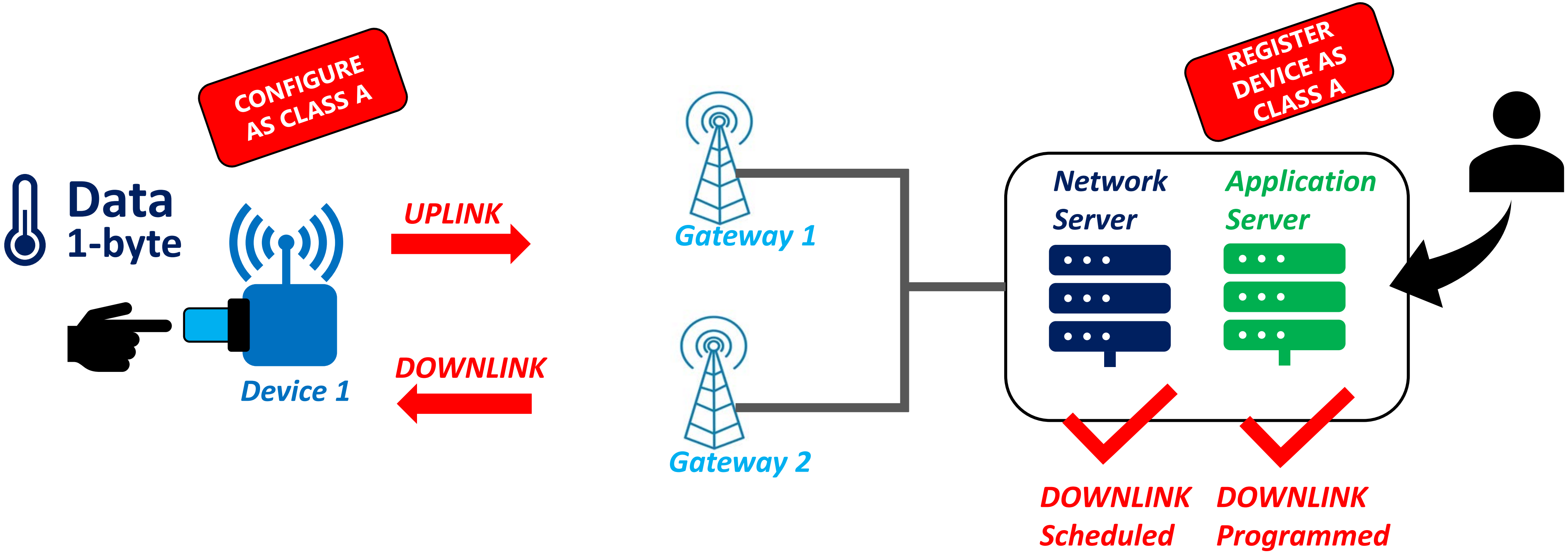
Class A → **All devices, Gateways and Network Server support class A**

Class B → **Device, Gateway, and Network Server must support class B**

⚠ **Specific gateway with absolute timestamp**

Class C → **Device, Gateway, and Network Server must support class C**

LoRaWAN classes – Demonstration



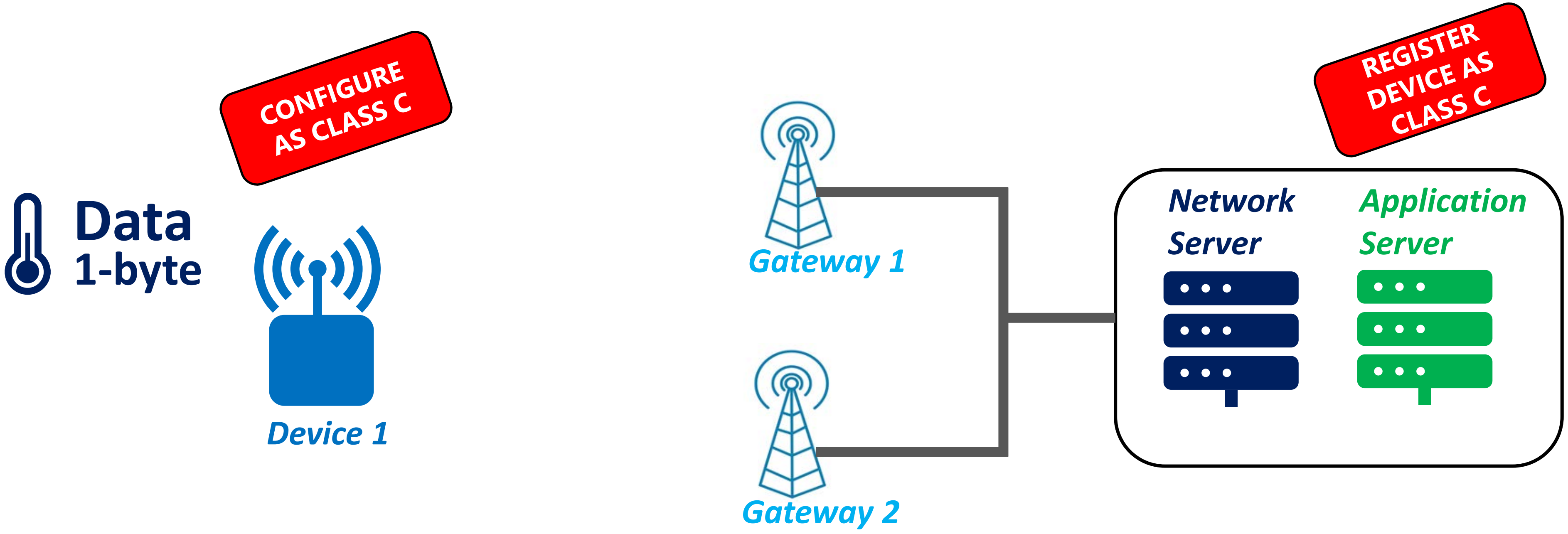
WHEN WILL THE DONWLINK FRAME BE SENT ?

CLASS A
RX1 or RX2

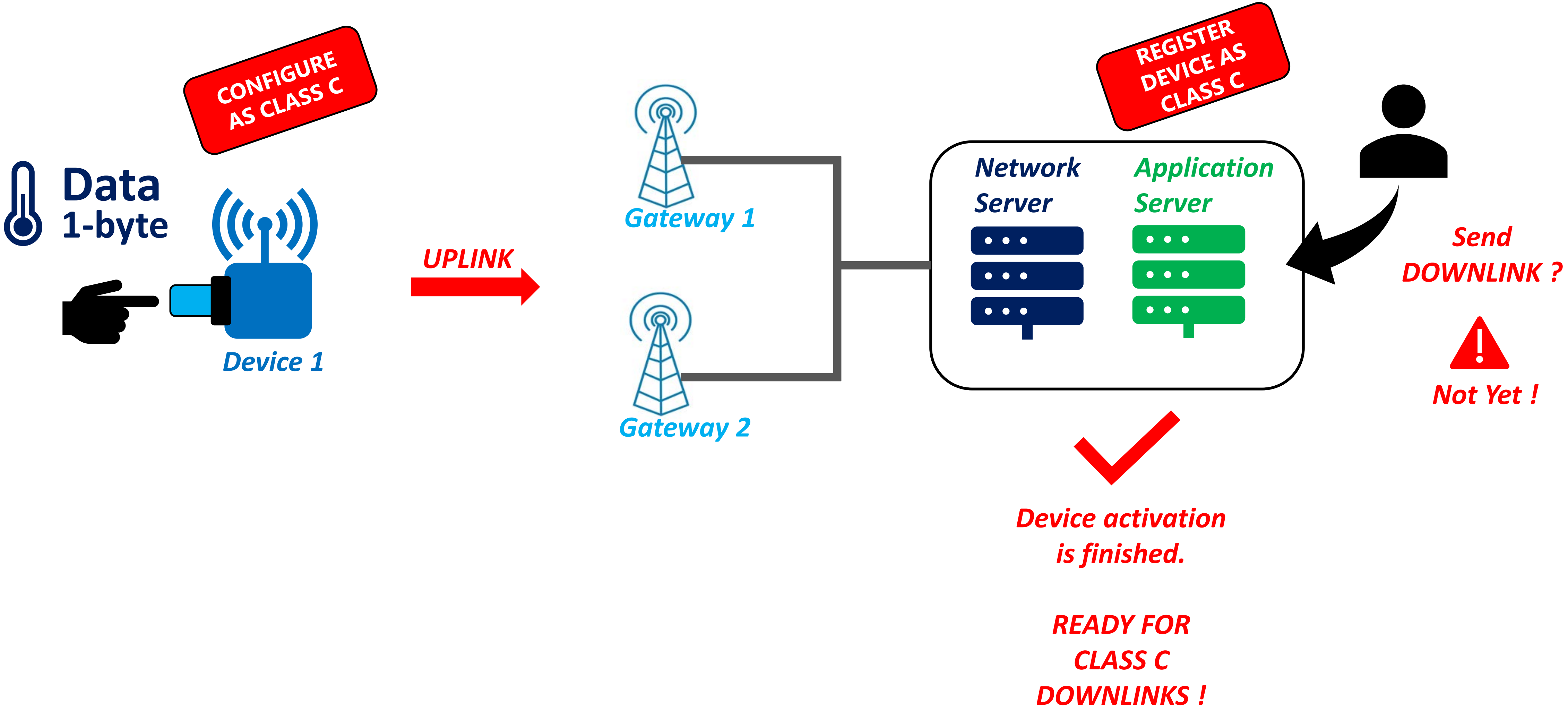
or

CLASS C
Whenever

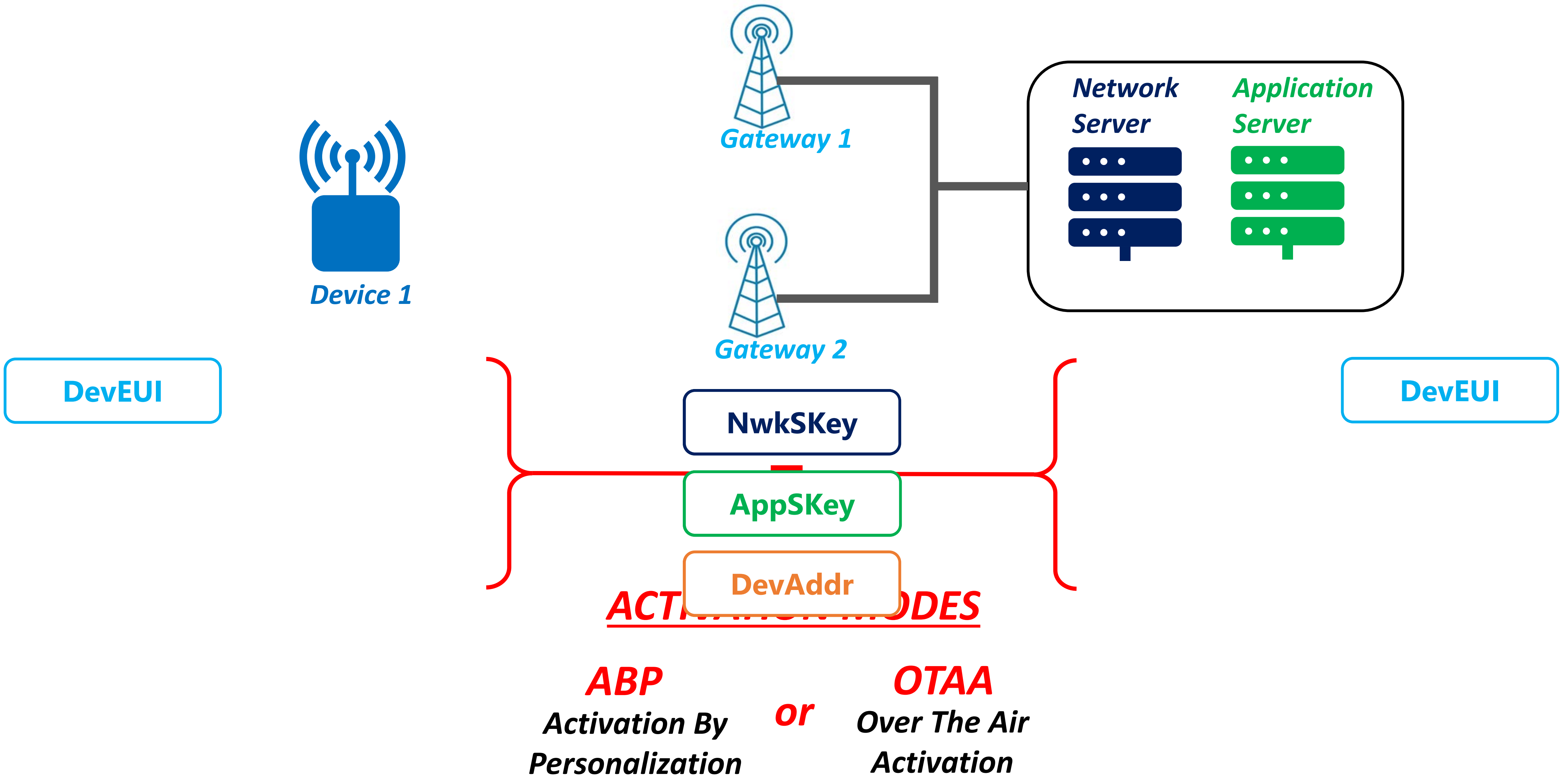
LoRaWAN classes – Demonstration



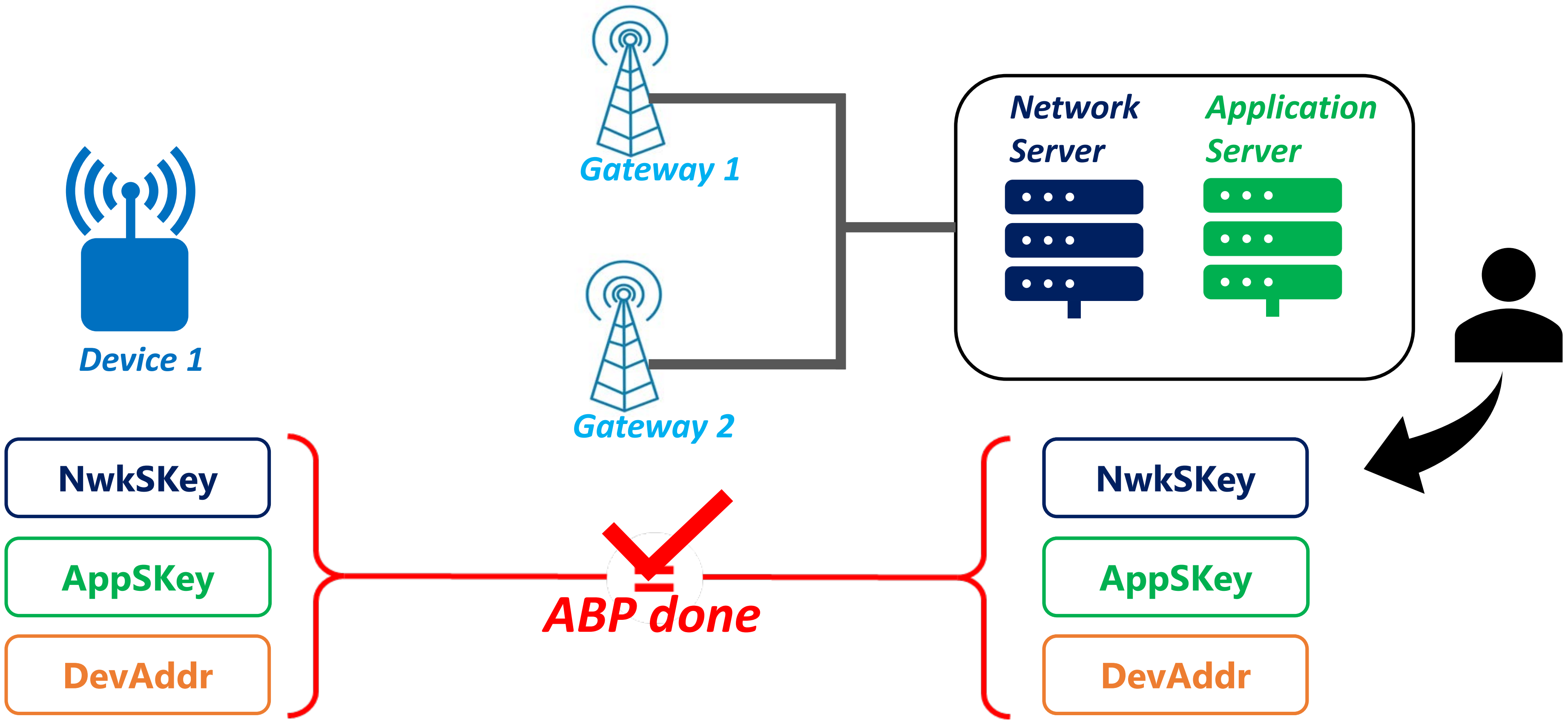
LoRaWAN classes – Demonstration



Activation modes – ABP or OTAA



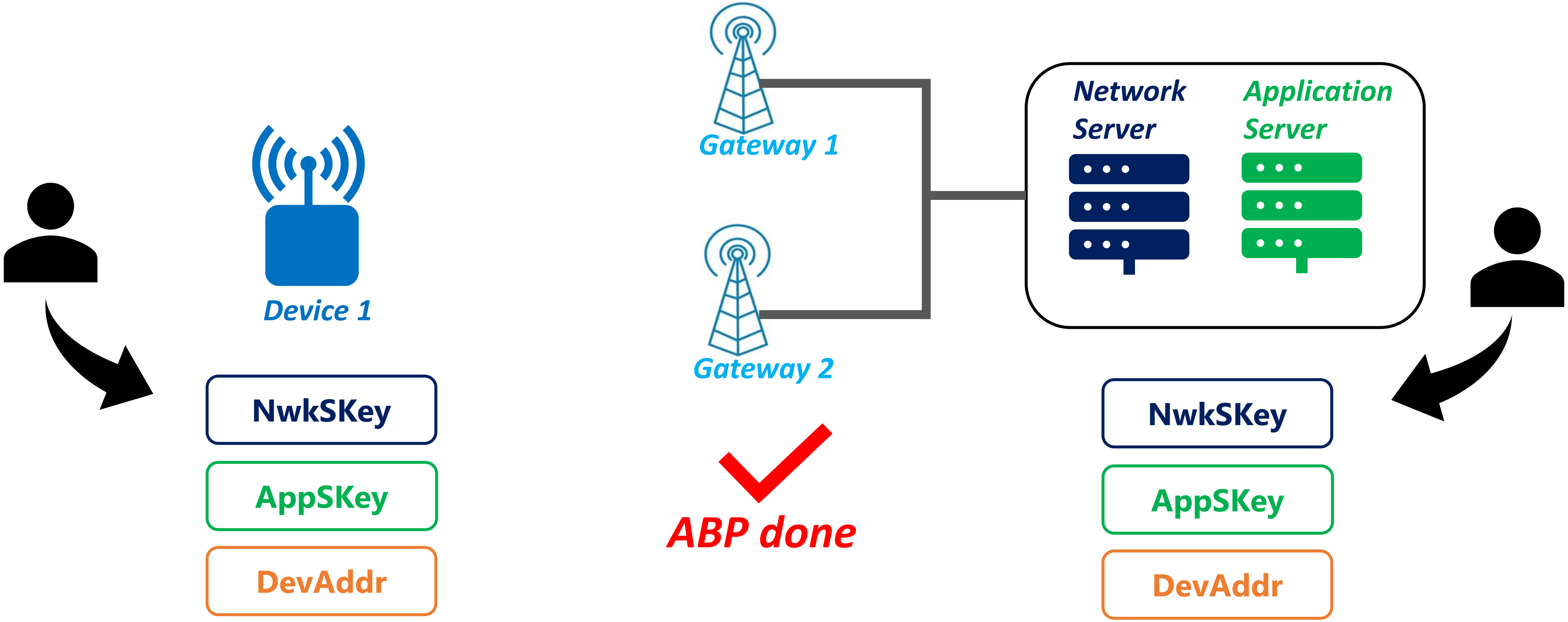
Activation By Personalization (ABP)



ABP – Way 1

The device is already provisioned

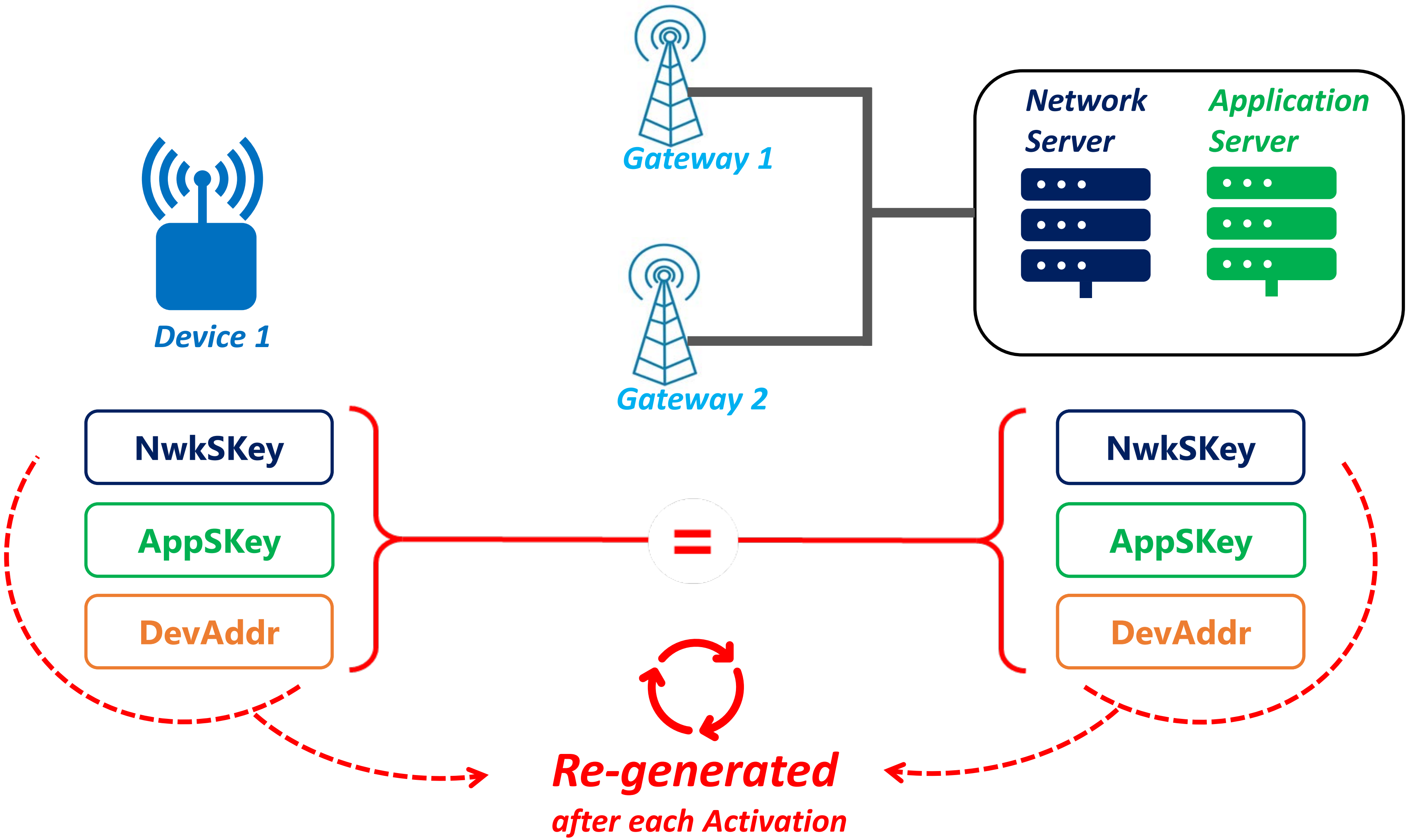
Activation By Personalization (ABP)



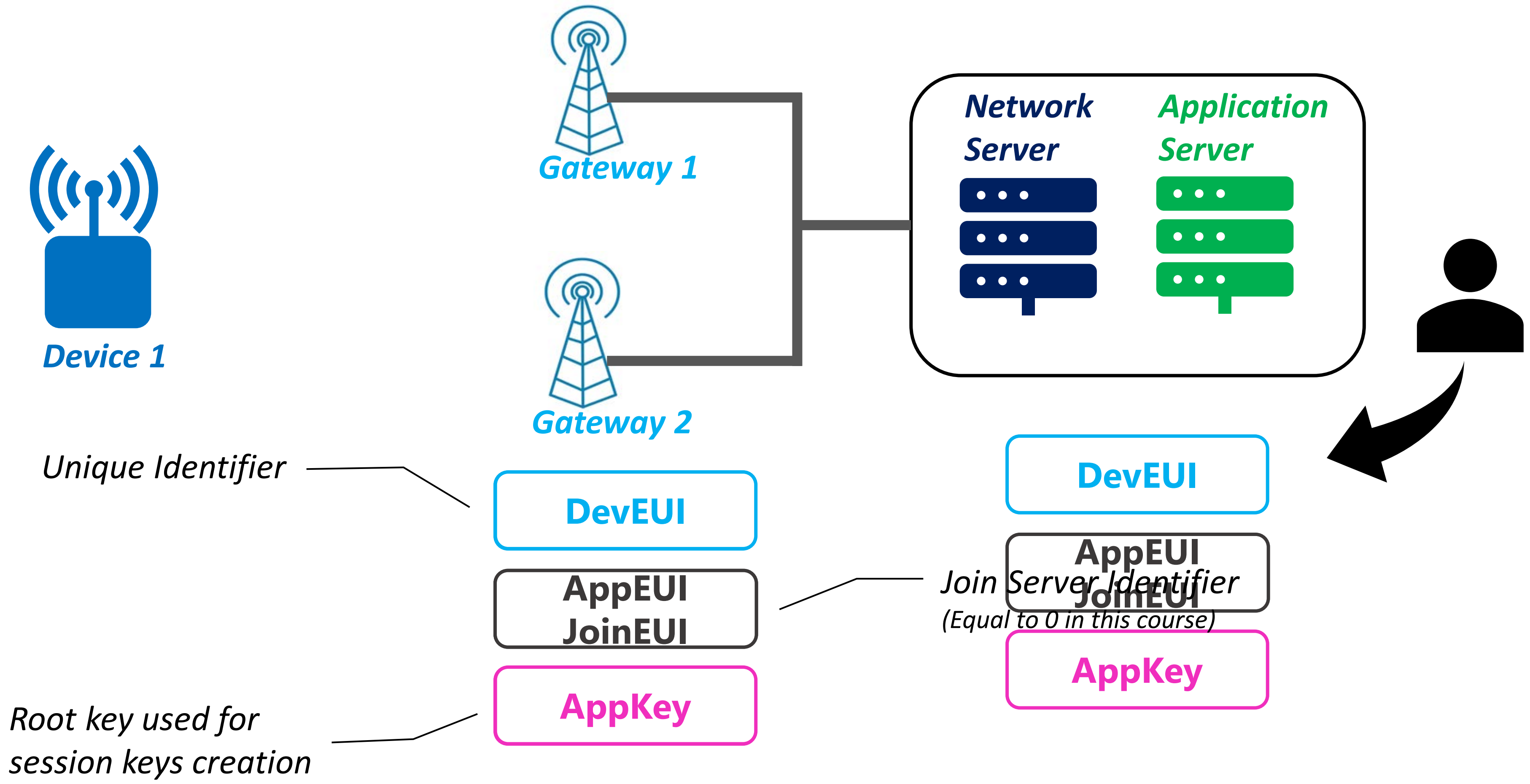
ABP – Way 2

Register the device in the server first

Over The Air Activation (OTAA)



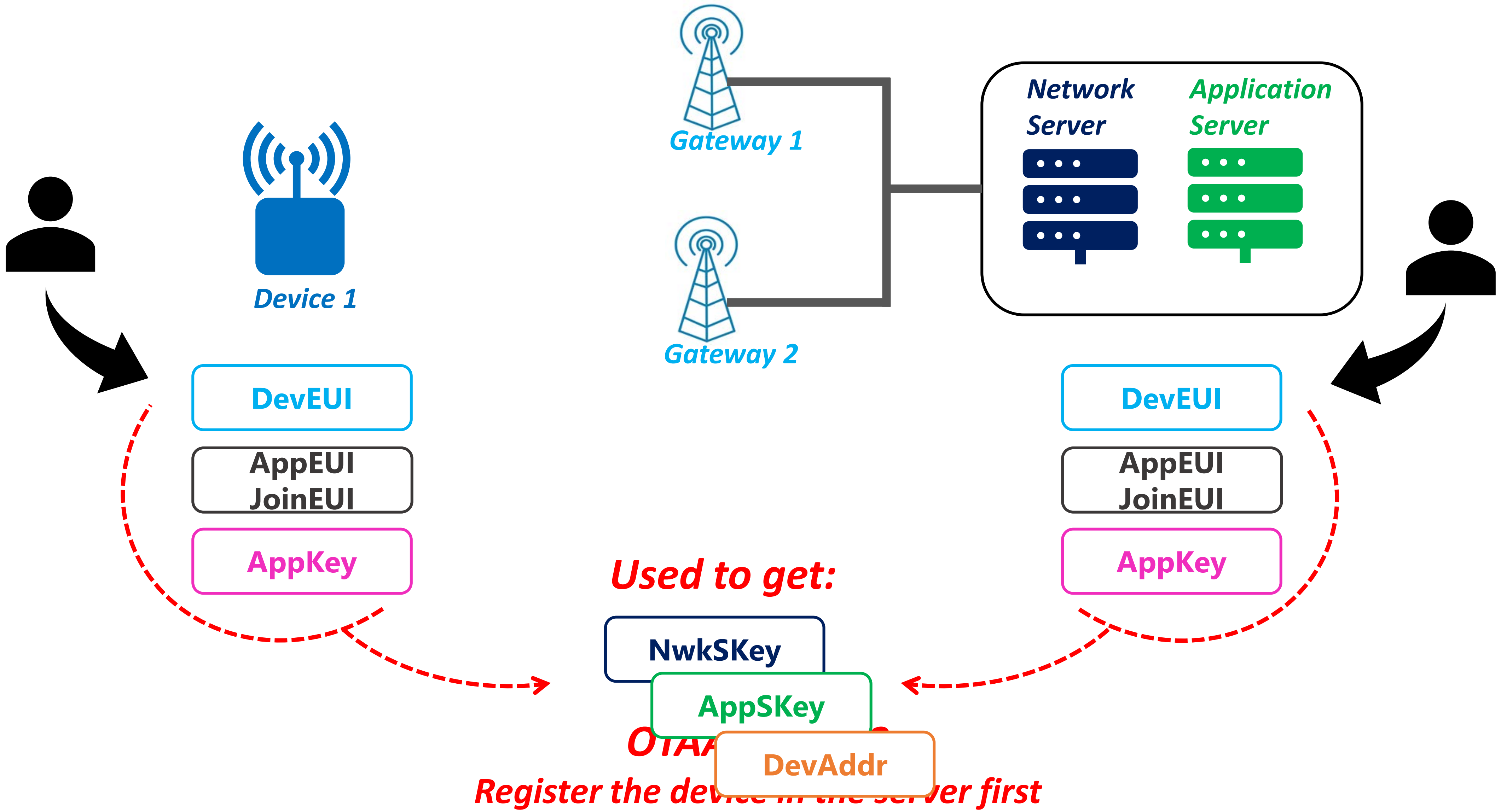
Over The Air Activation (OTAA)



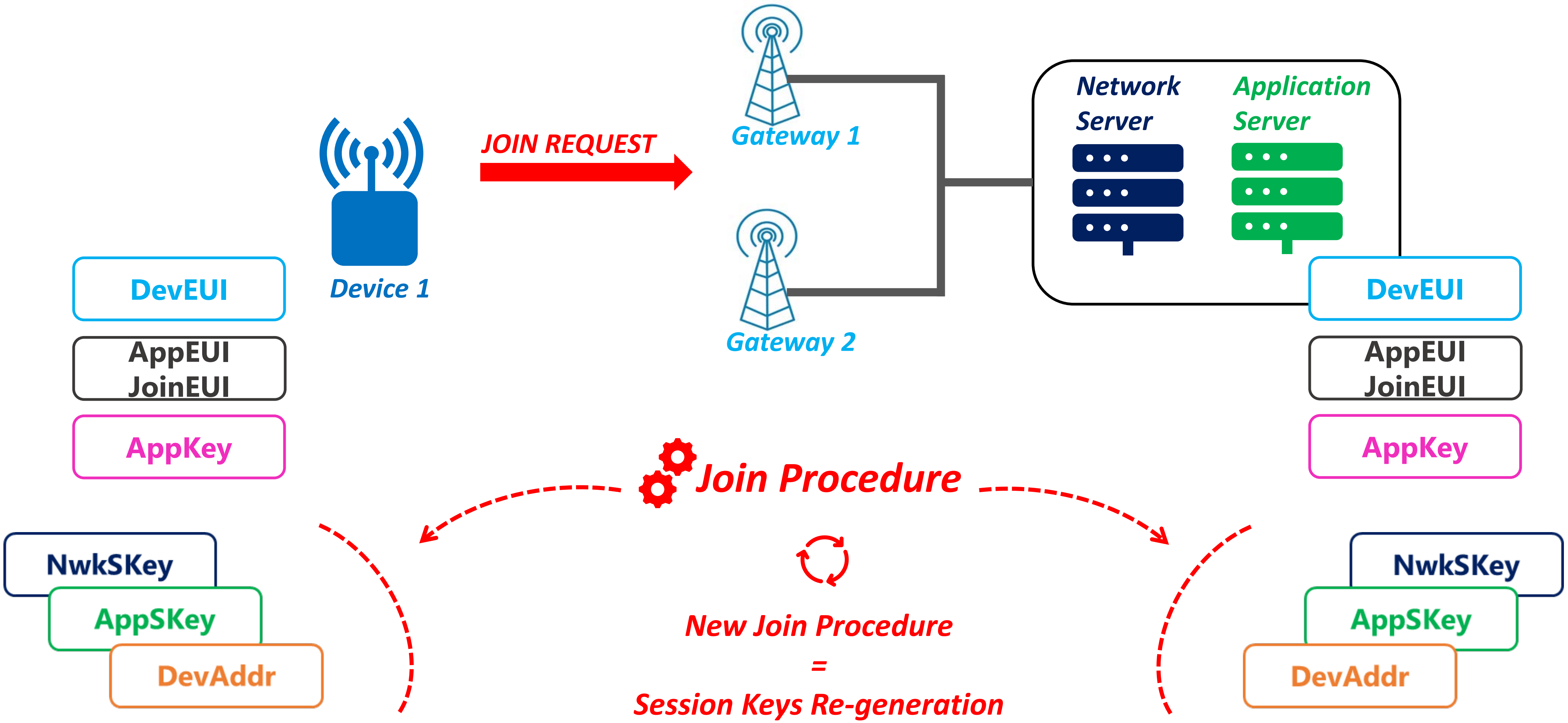
OTAA – Way 1

The device is already provisioned

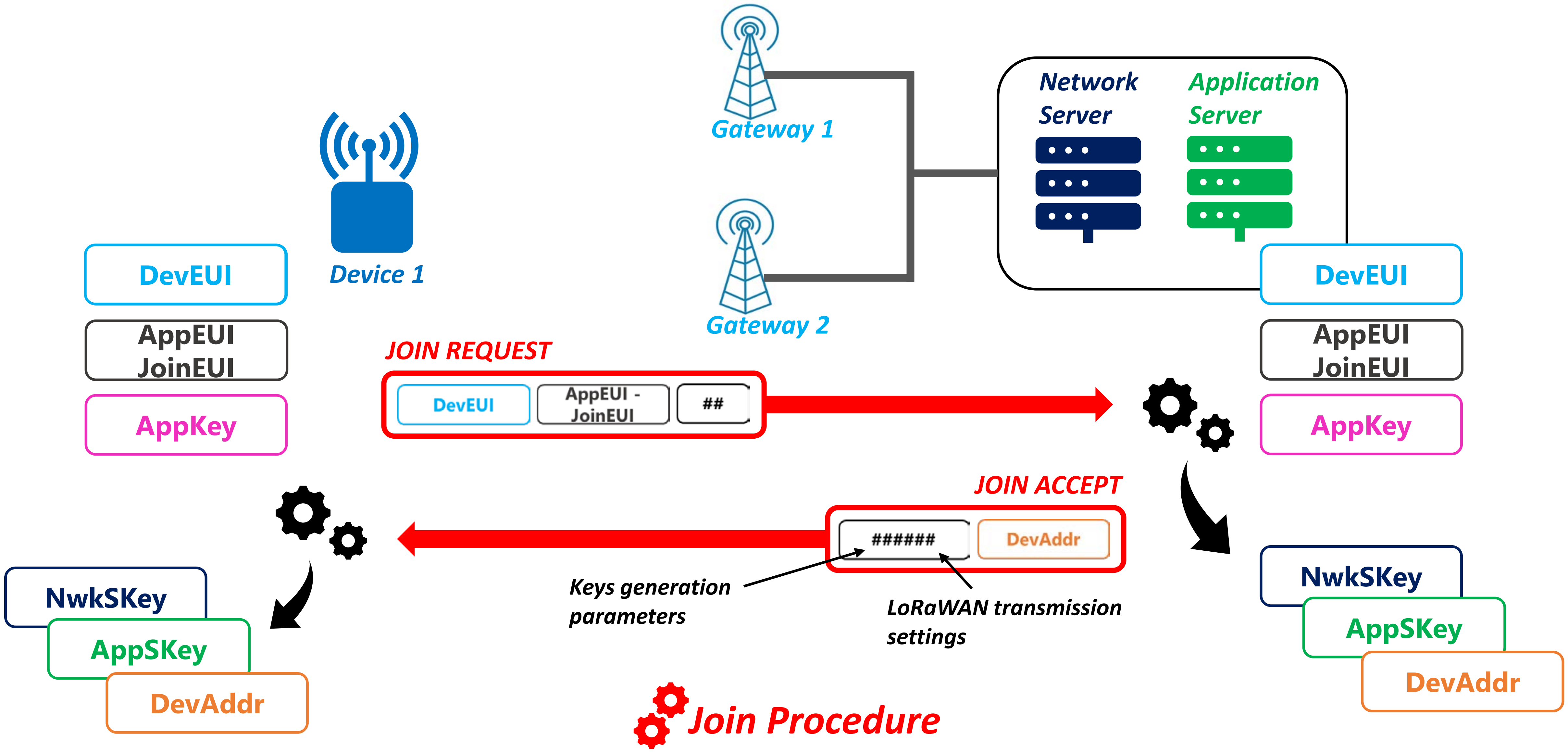
Over The Air Activation (OTAA)



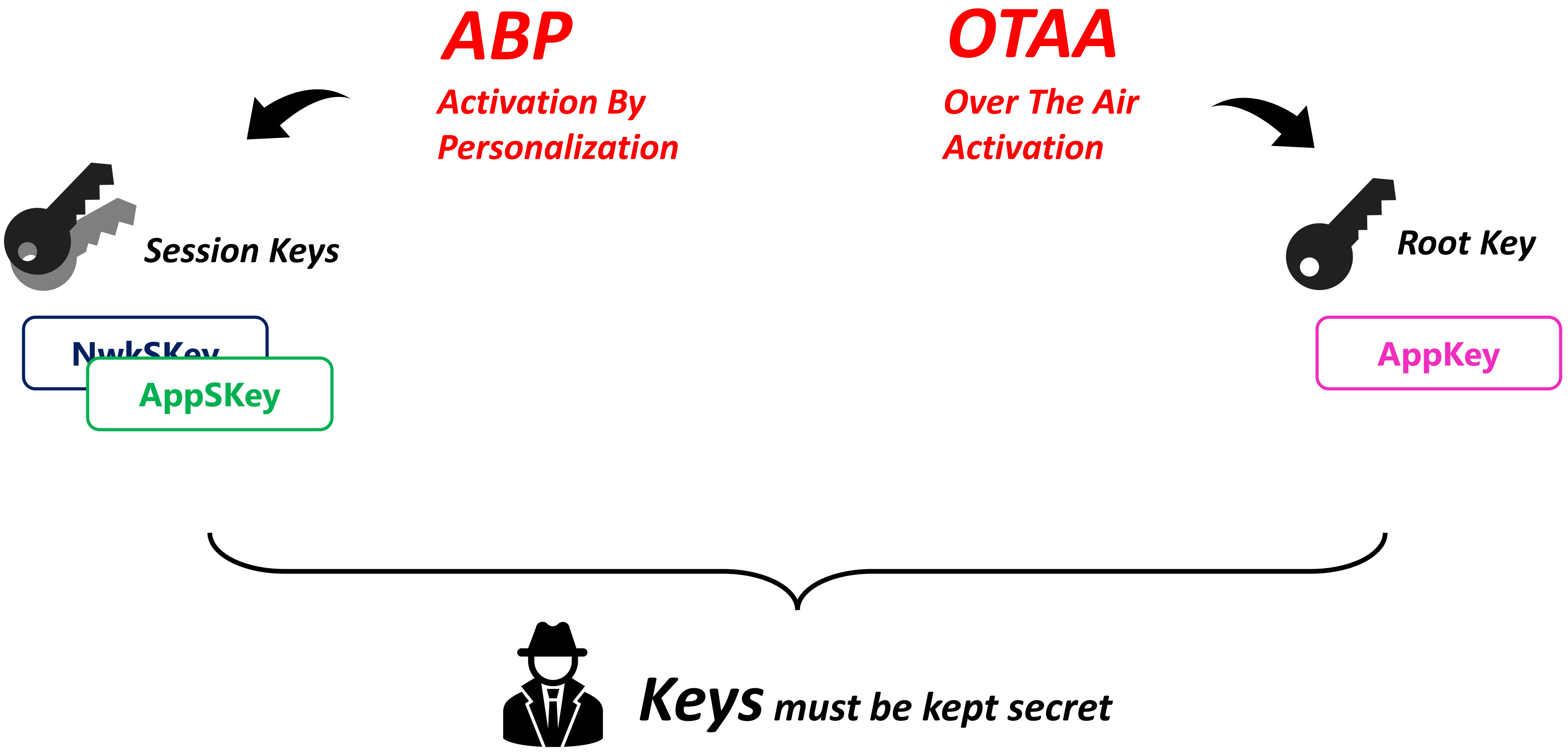
Over The Air Activation (OTAA)



Over The Air Activation (OTAA)



Choice between ABP or OTAA - 1



Choice between ABP or OTAA - 2

Example

Device Maker

Customer

Network Server

ABP

- ✓ Secure Keys Storage
- ✗ Session Key re-generation

BRUTFORCE ATTACKS

OTAA

- ✓ Secure Keys Storage
- ✓ Session Key re-generation

Generated Keys

Generated Keys

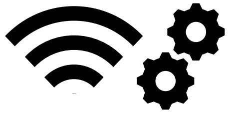
SE
Secure Element

KMS
Key Management System

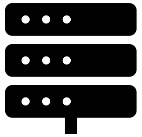
Choice between ABP or OTAA - 3

Questions !

LoRaWAN & Replay-Attack



LoRaWAN Communication parameters

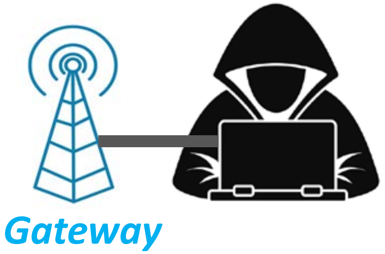


Change of Network Server

The REPLAY attack

LoRaWAN is Encrypted & Secured

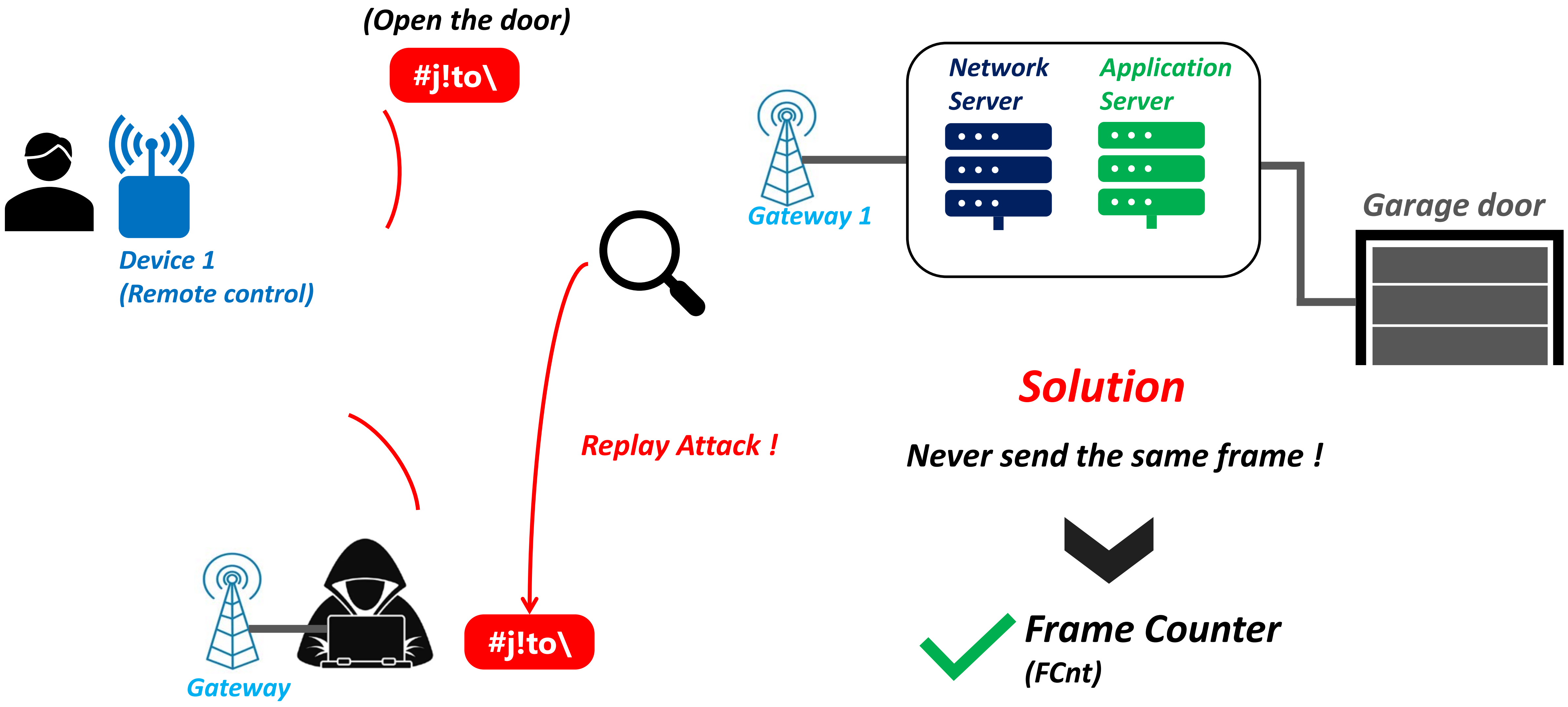
BUT



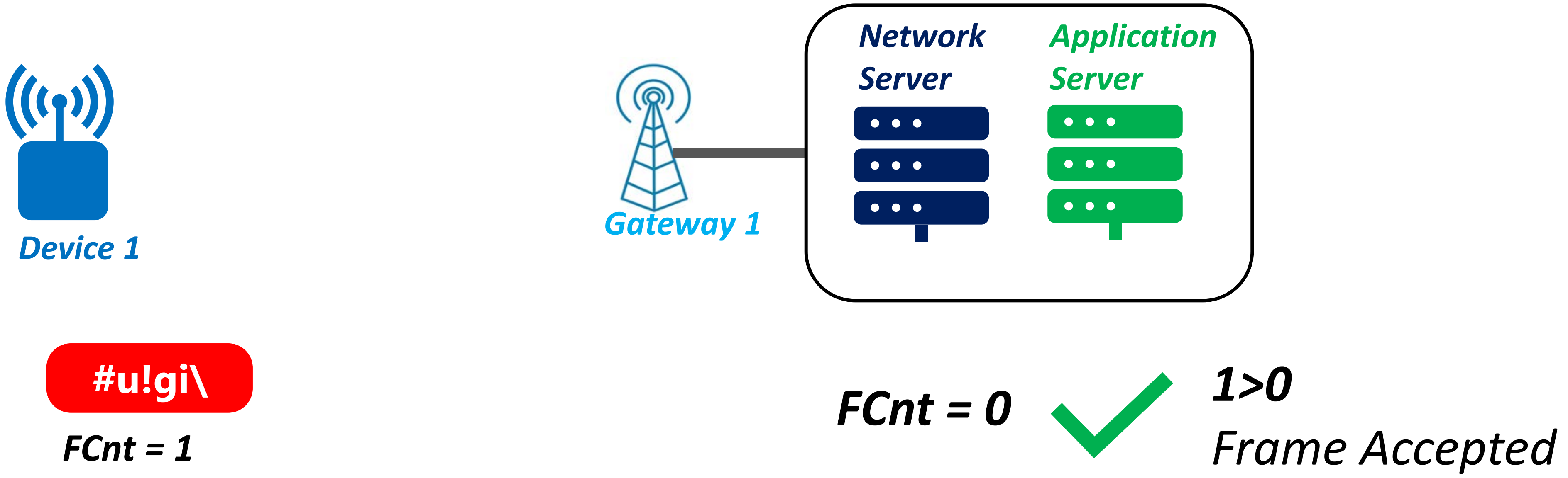
Everyone can listen LoRaWAN communications

Encryption → ***Confidentiality*** ***Authenticity***

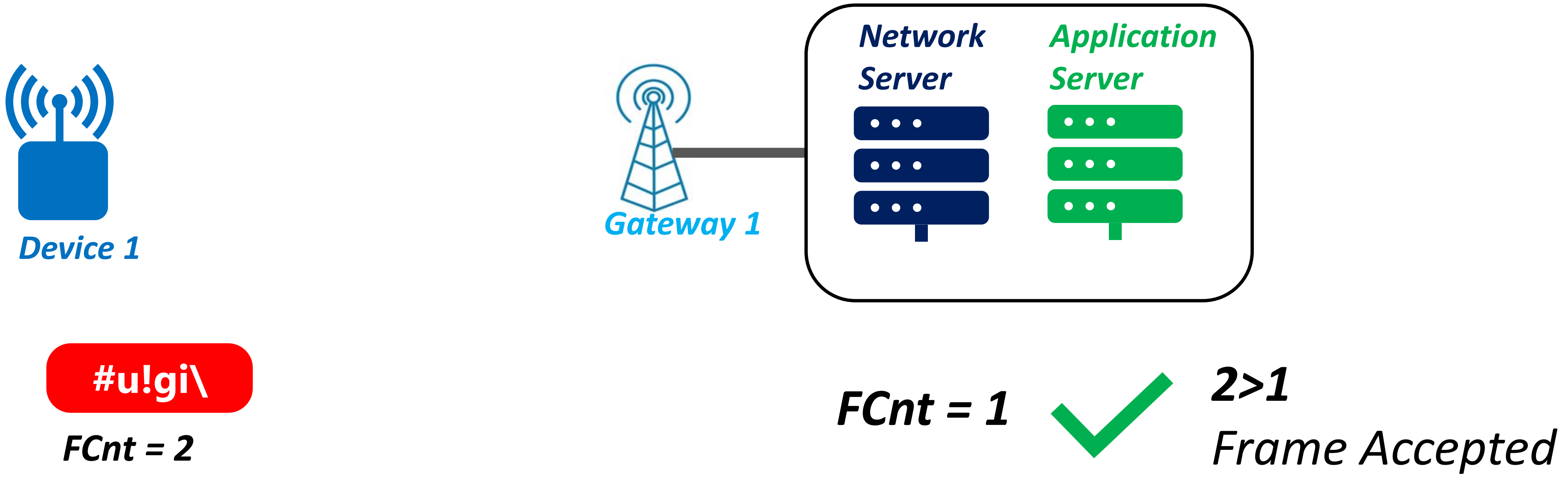
The REPLAY attack



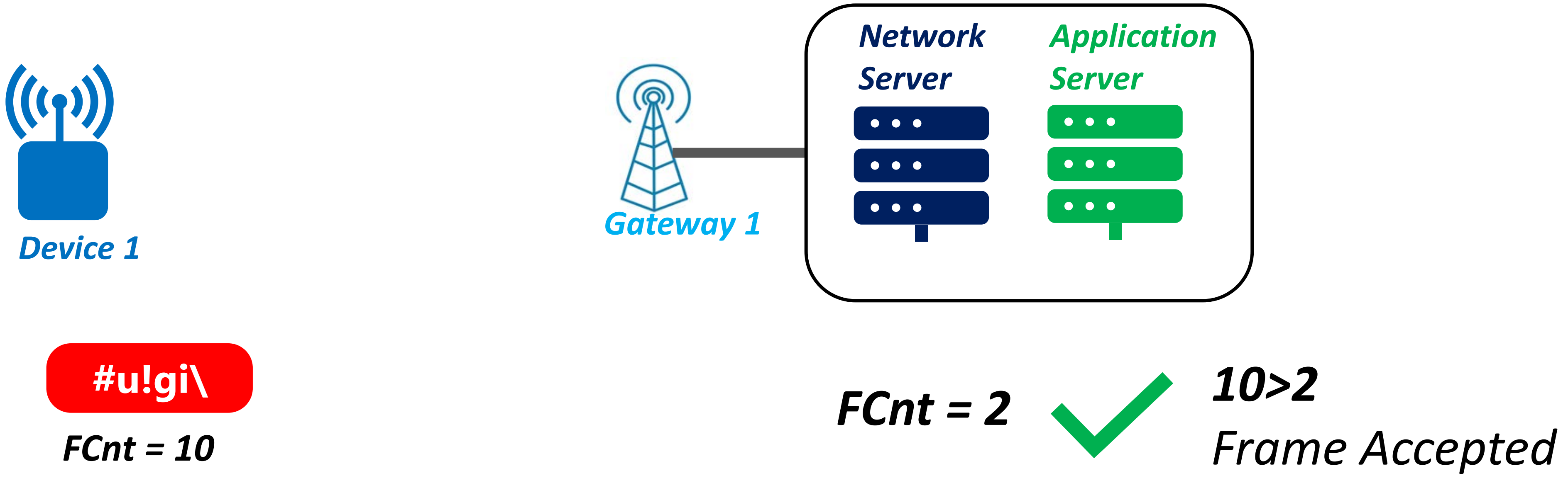
The REPLAY attack



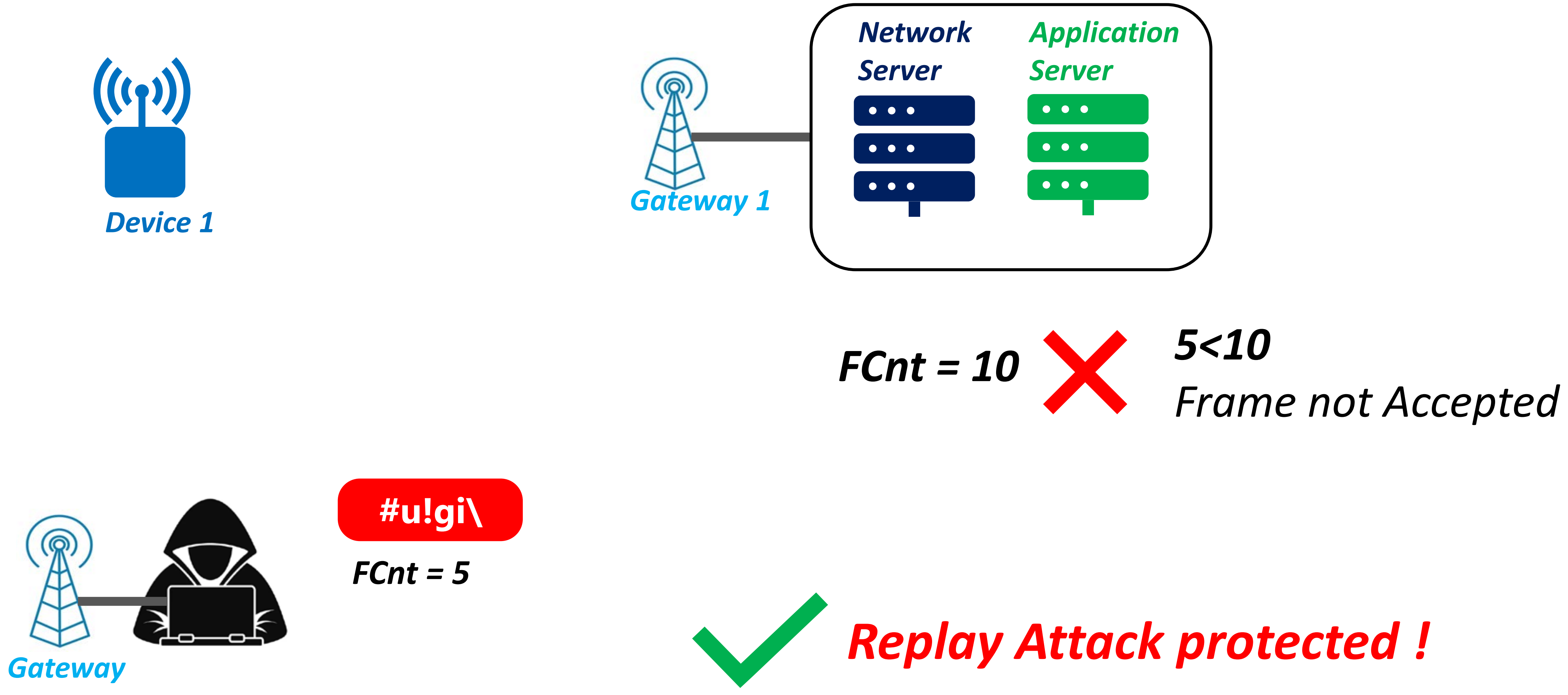
The REPLAY attack



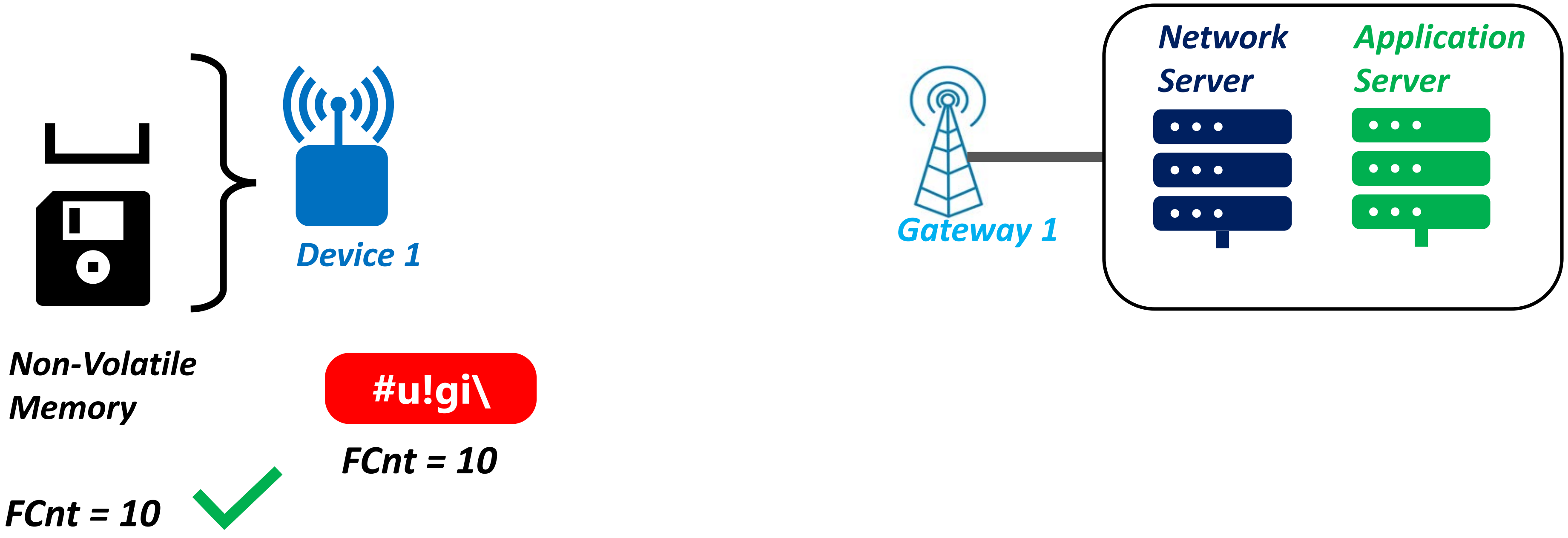
The REPLAY attack



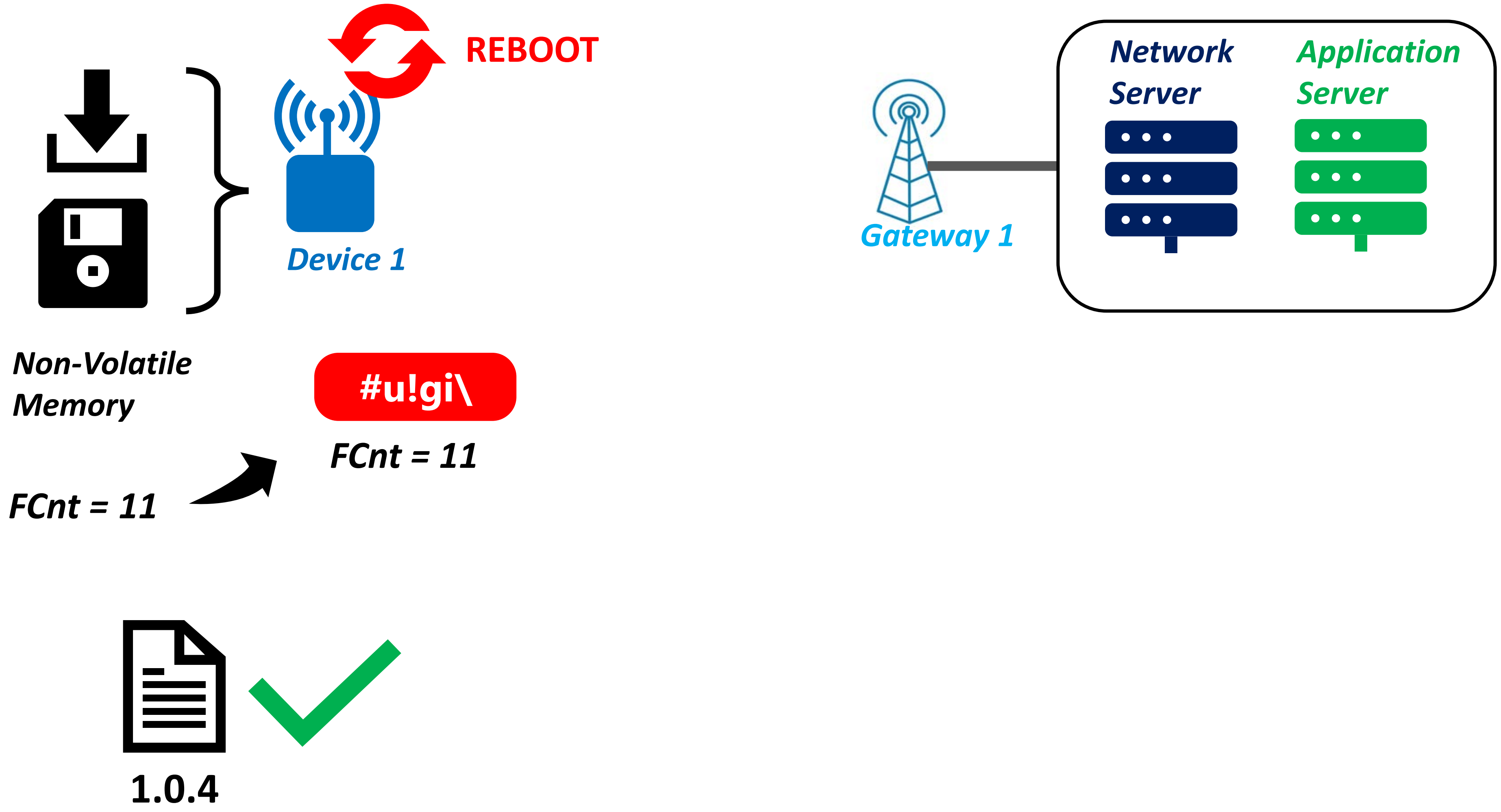
The REPLAY attack



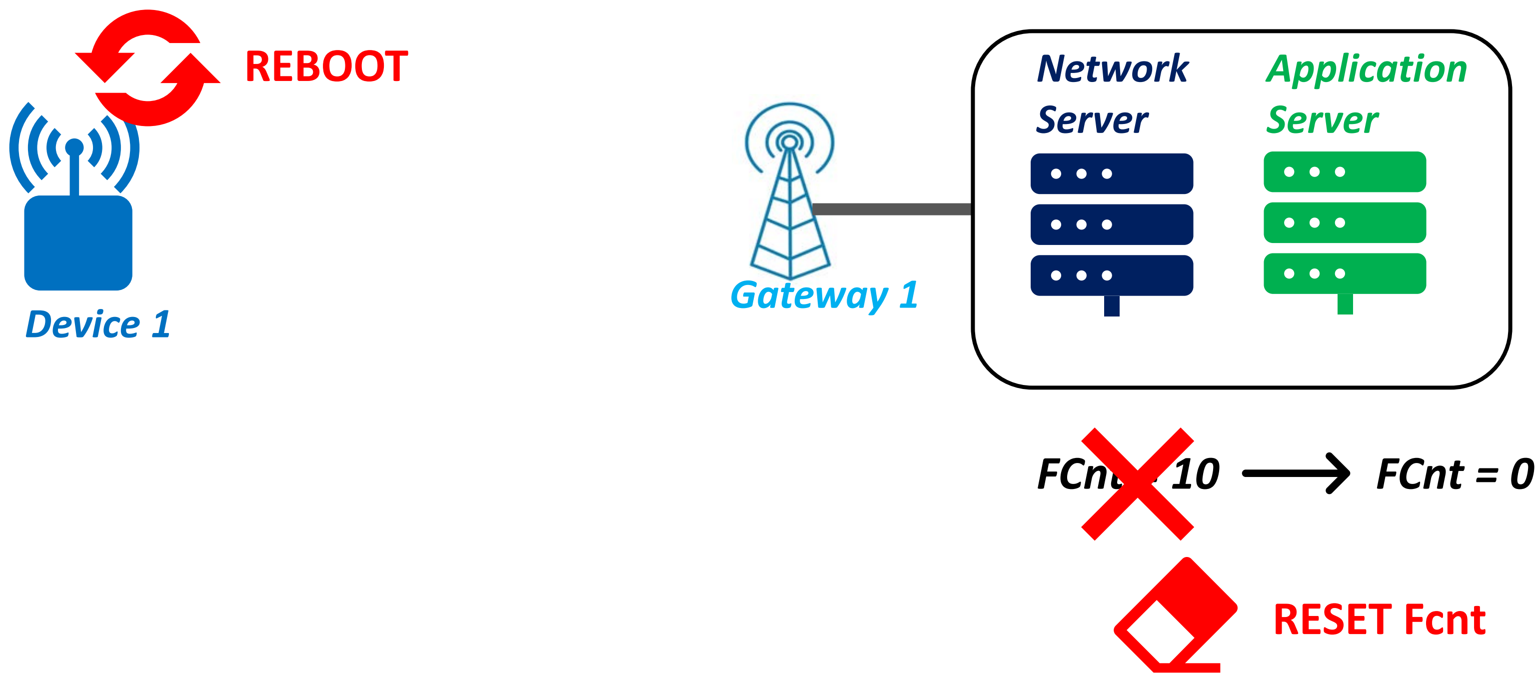
The REPLAY attack



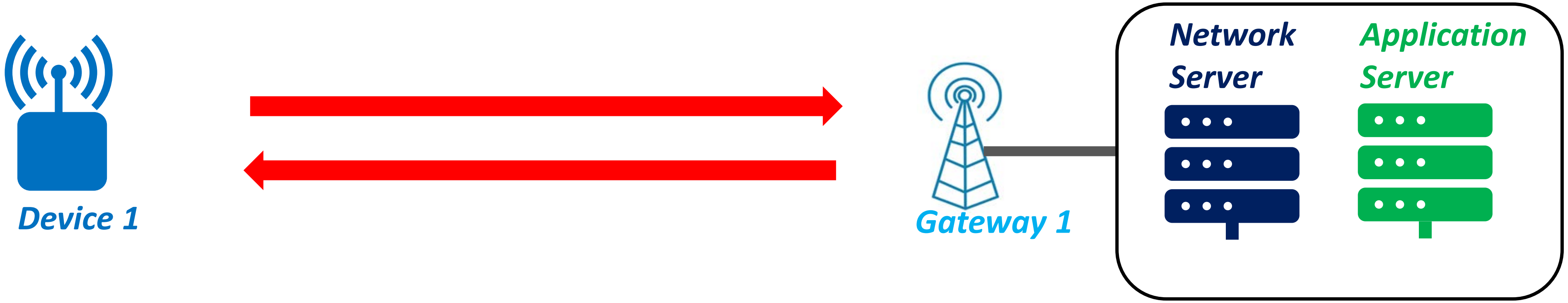
The REPLAY attack



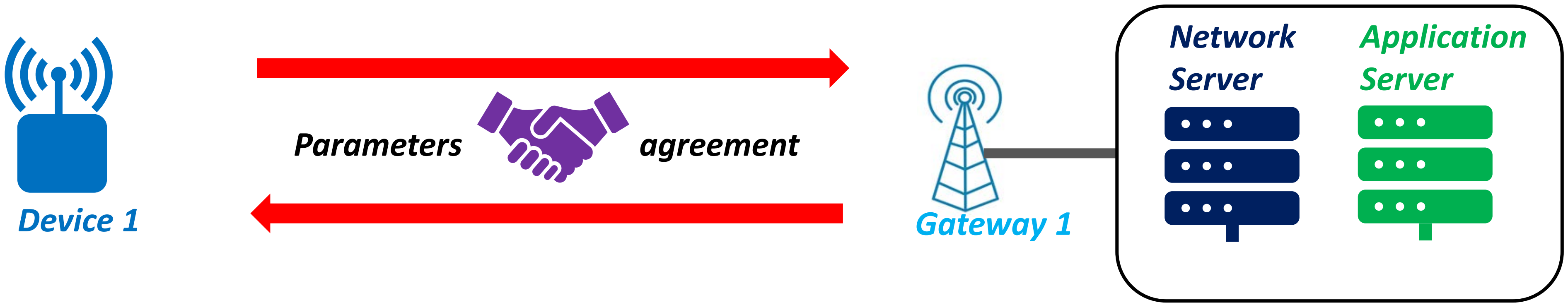
The REPLAY attack



The LoRaWAN parameters



The LoRaWAN parameters



CHANNELS

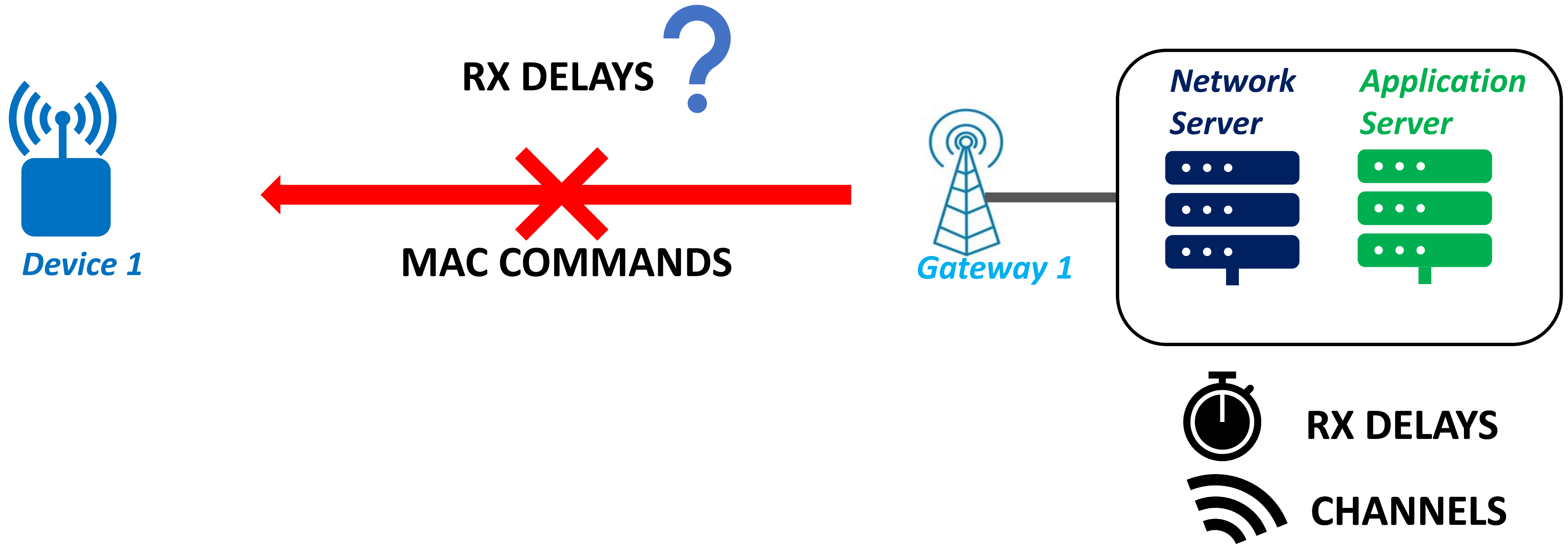
- **Mandatory** *(End-devices know them)* — 868.1MHz – 868.3MHz – 868.5MHz
- **Extra channels** *(Shared between the NS and the end-device)* — **CF List** *(gives the Frequency Plan)*

RX DELAYS

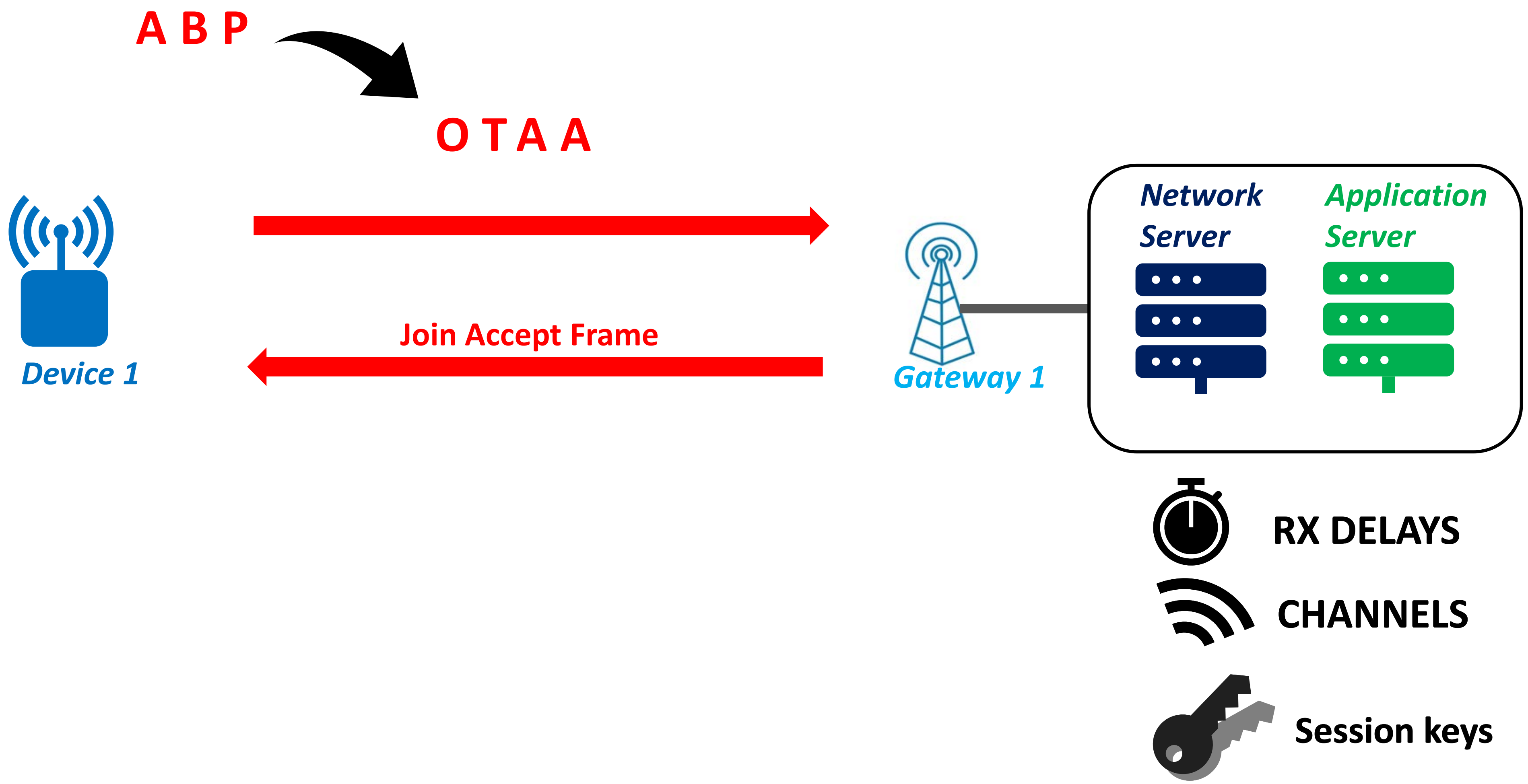
- **Default time slots** *(Configured in the end-device)* — RX1, 1sec / RX2, 2sec
- **Other time slots** *(Configured in the NS)* — **RX Delays** *(gives the actual time slots)*

The LoRaWAN parameters

The LoRaWAN parameters



The LoRaWAN parameters



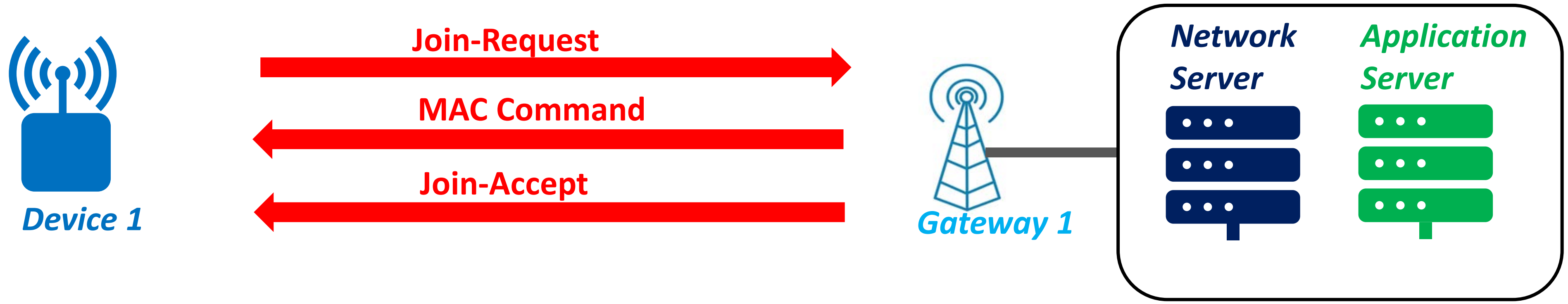
The LoRaWAN parameters - demonstration

Default channels

868.1 MHz
868.3 MHz
868.5 MHz

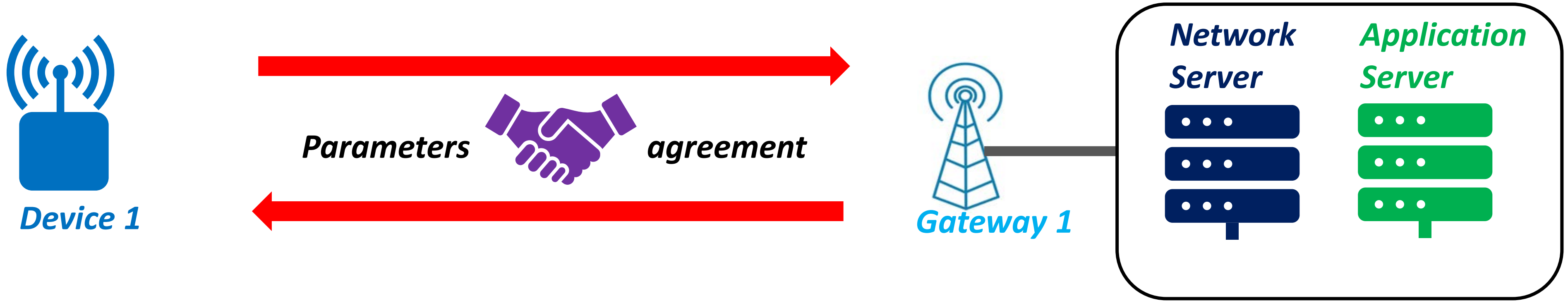
Network Server Frequency plan

868.1 MHz	867.1 MHz	867.7 MHz
868.3 MHz	867.3 MHz	867.9 MHz
868.5 MHz	867.5 MHz	



1. ABP end-device with default channels
2. ABP end-device with default channels + MAC Command
3. OTAA end-device with default channels

The LoRaWAN parameters



ABP

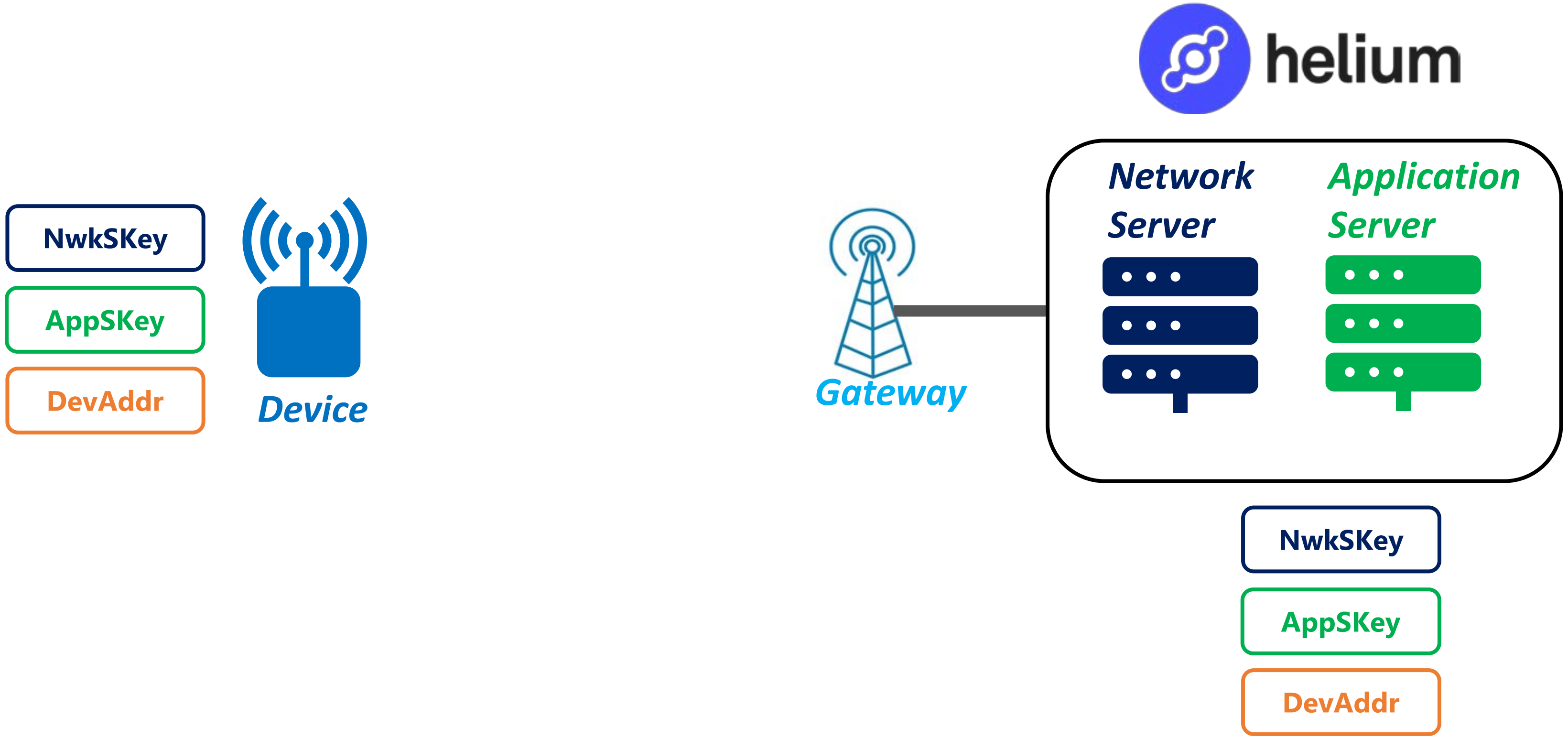
Configuration during development

Configuration with MAC Commands

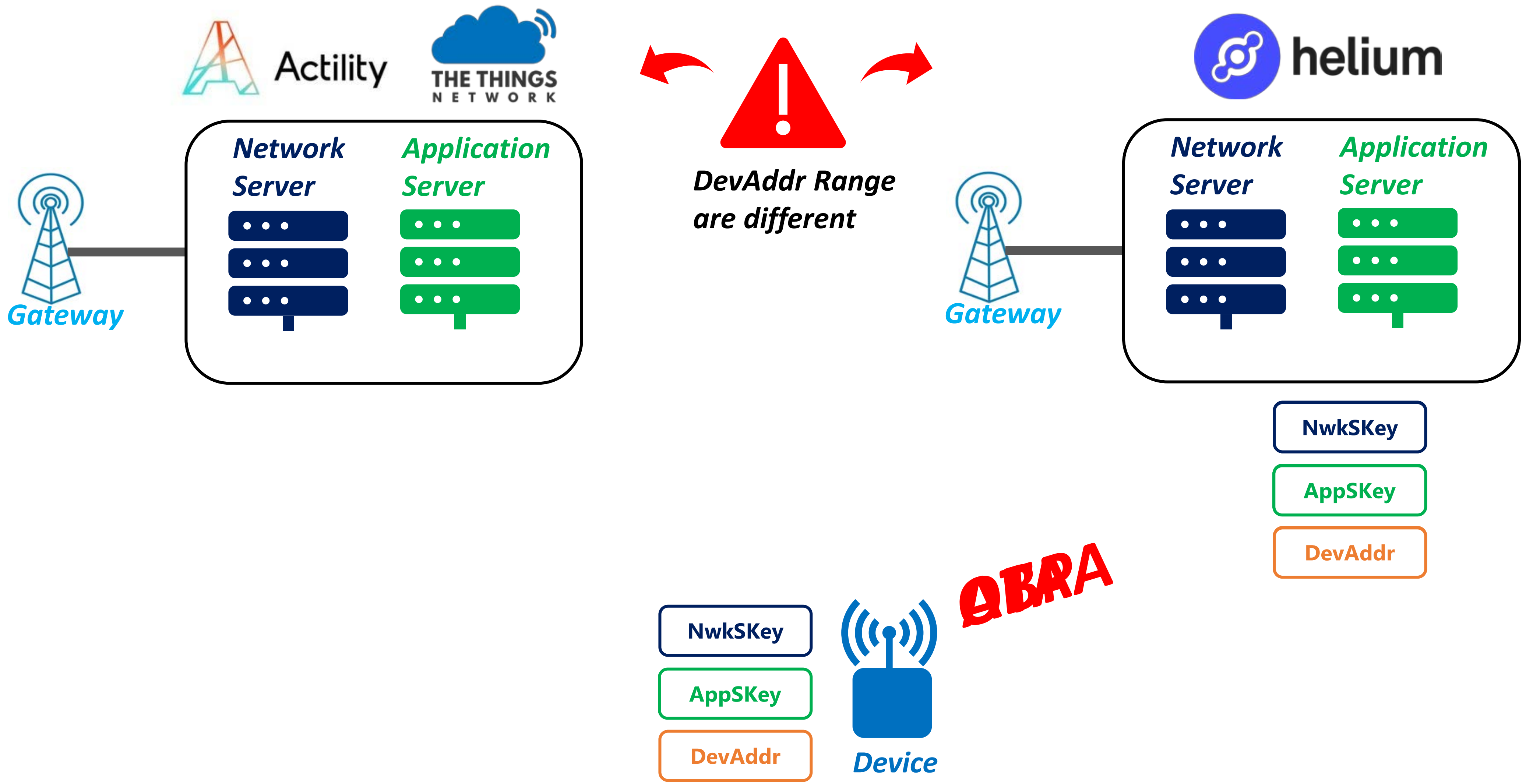
OTAA

Seamless configuration during JOIN

How to change network operator?



How to change network operator?



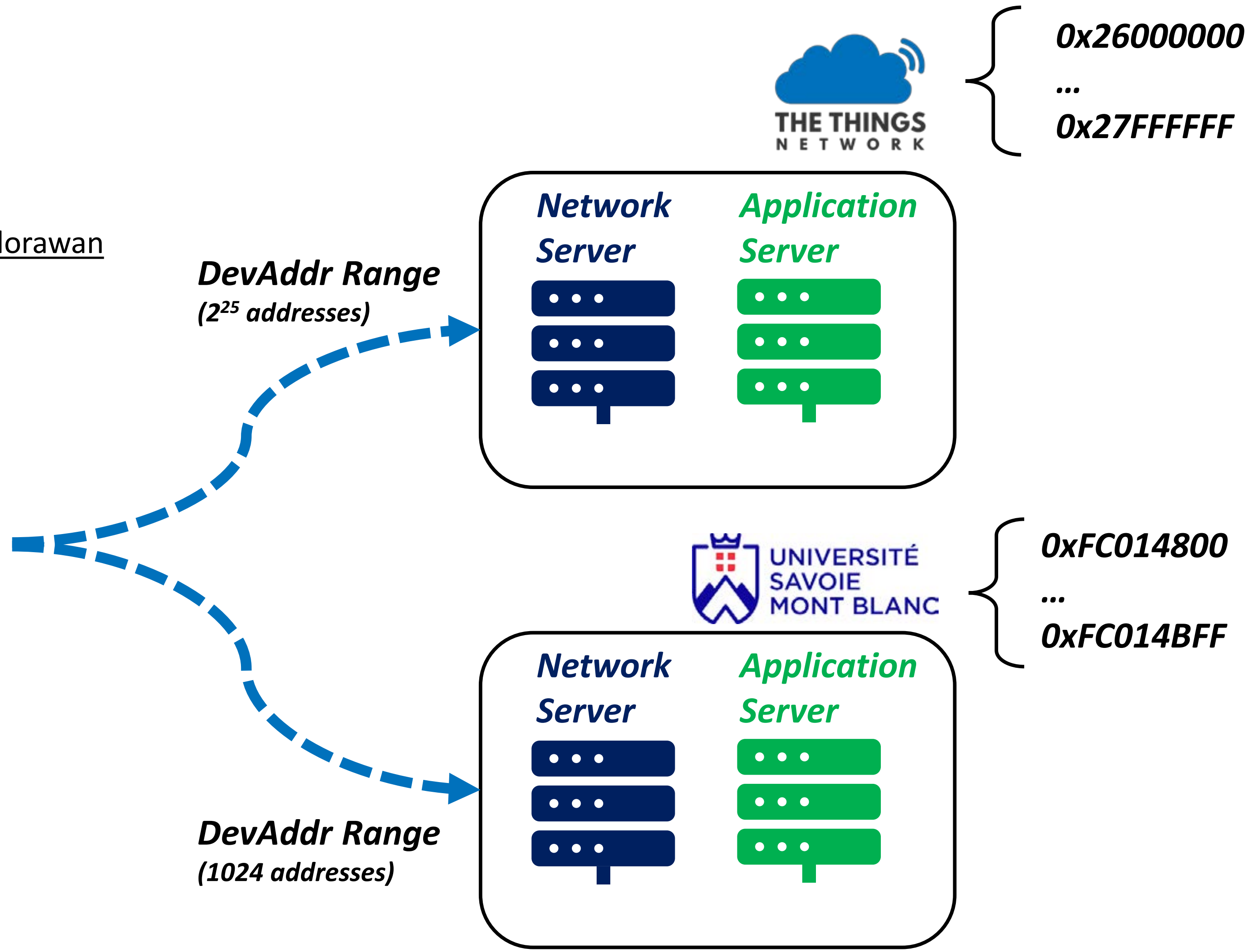
How to change network operator?

ℹ️ ***ROAMING***

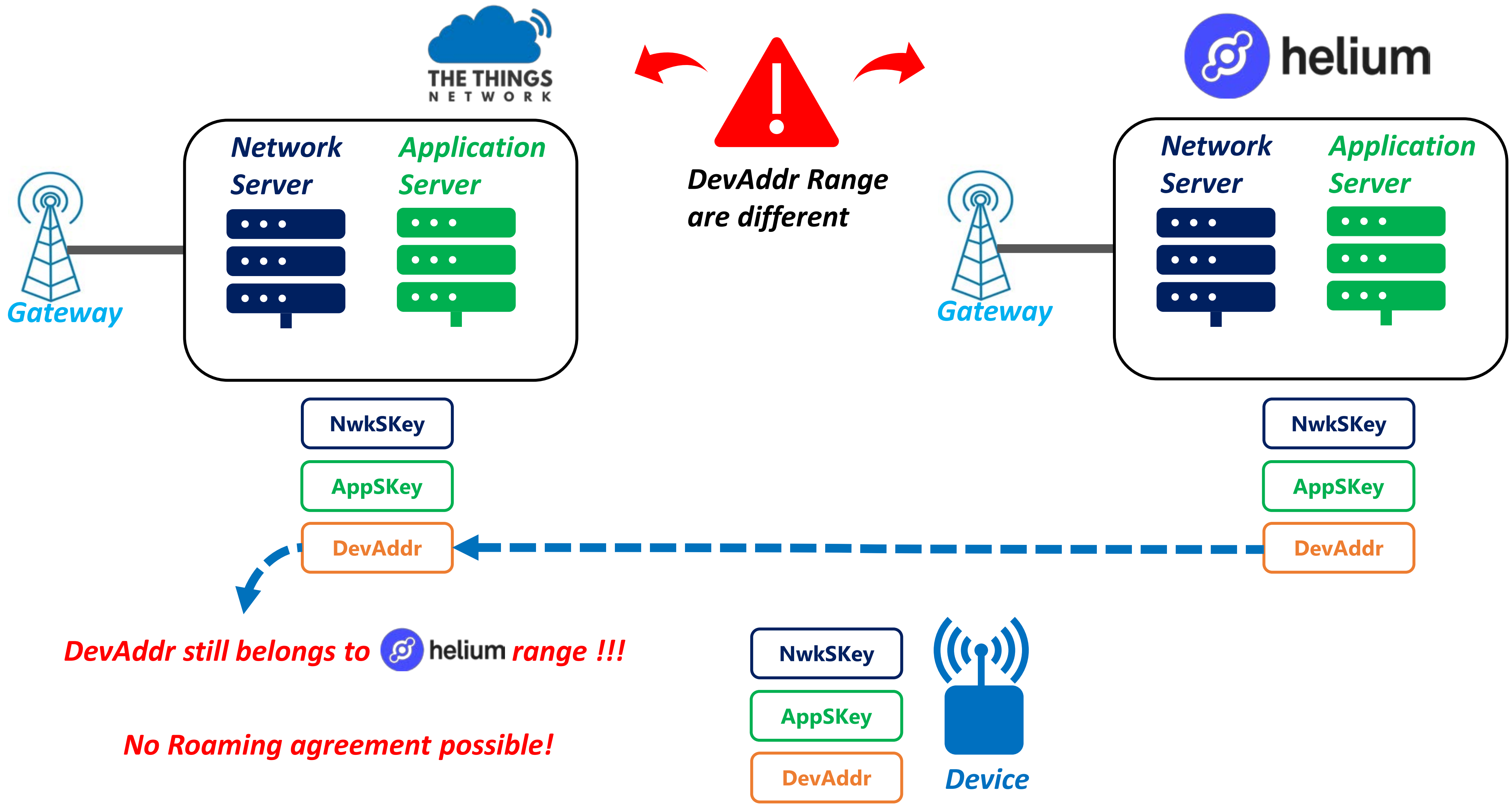
- *coverage extension*
- *gateway densification*

Free e-book
www.univ-smb.fr/lorawan

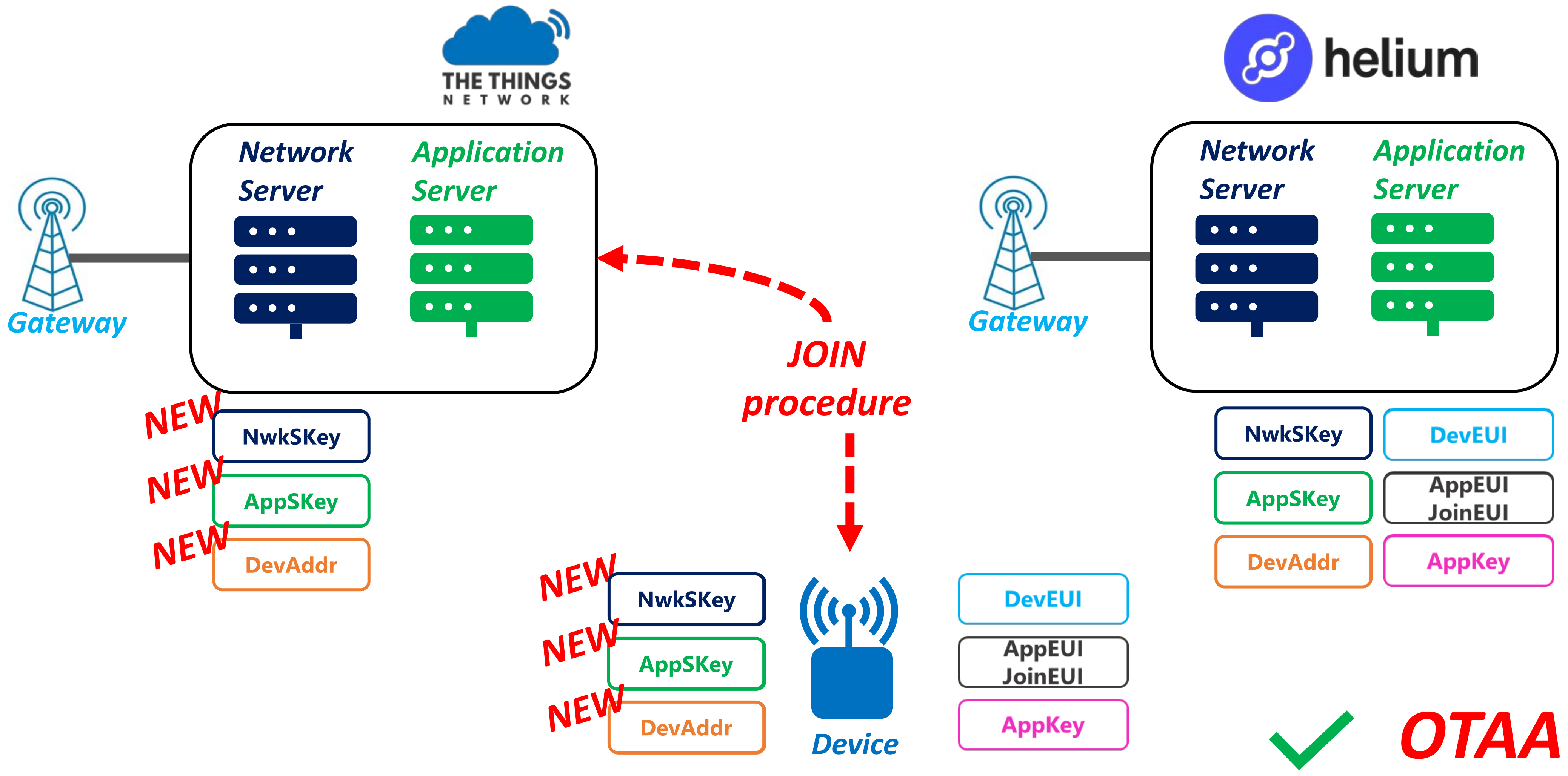
LoRa Alliance®



How to change network operator?



How to change network operator?

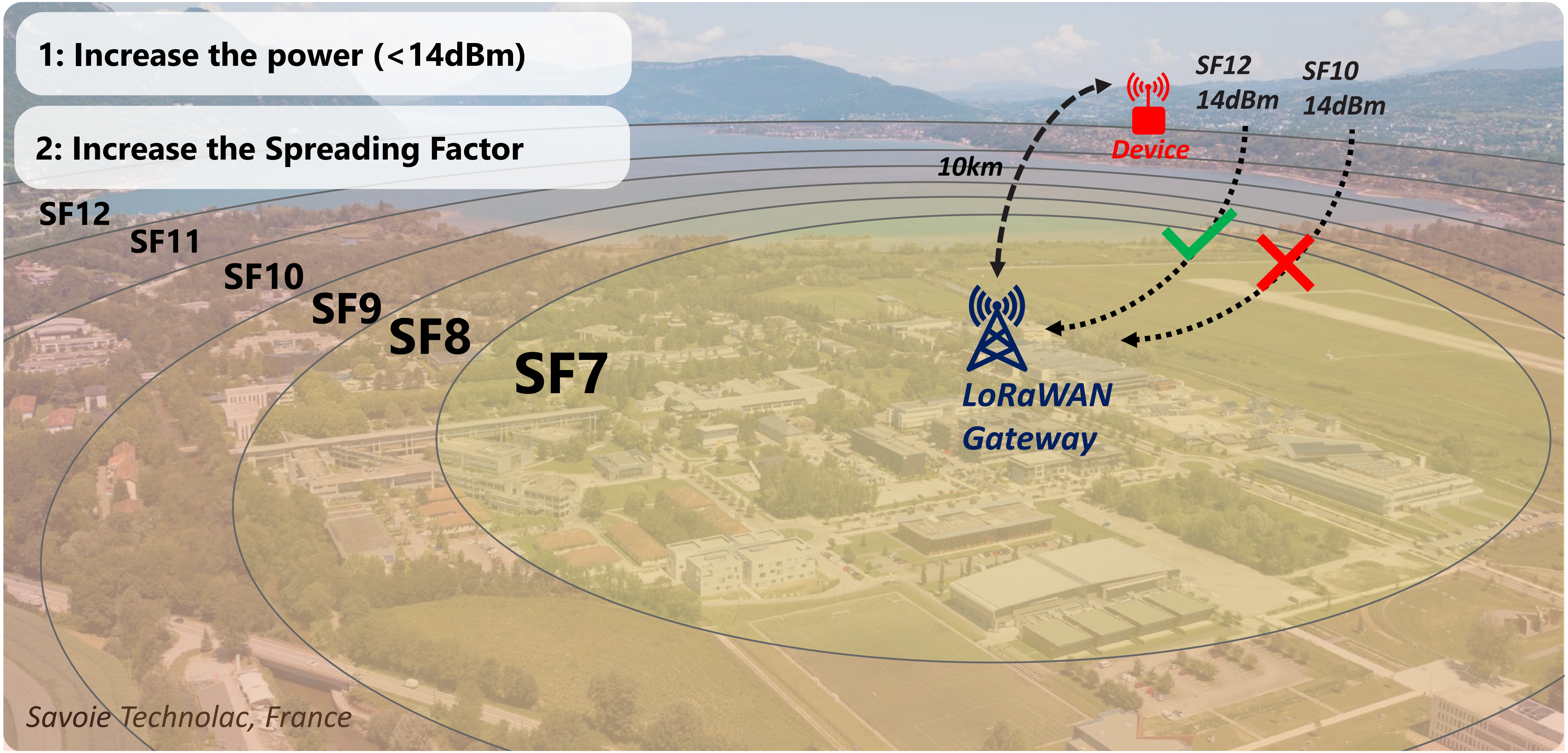


ABP and OTAA: let's recap!

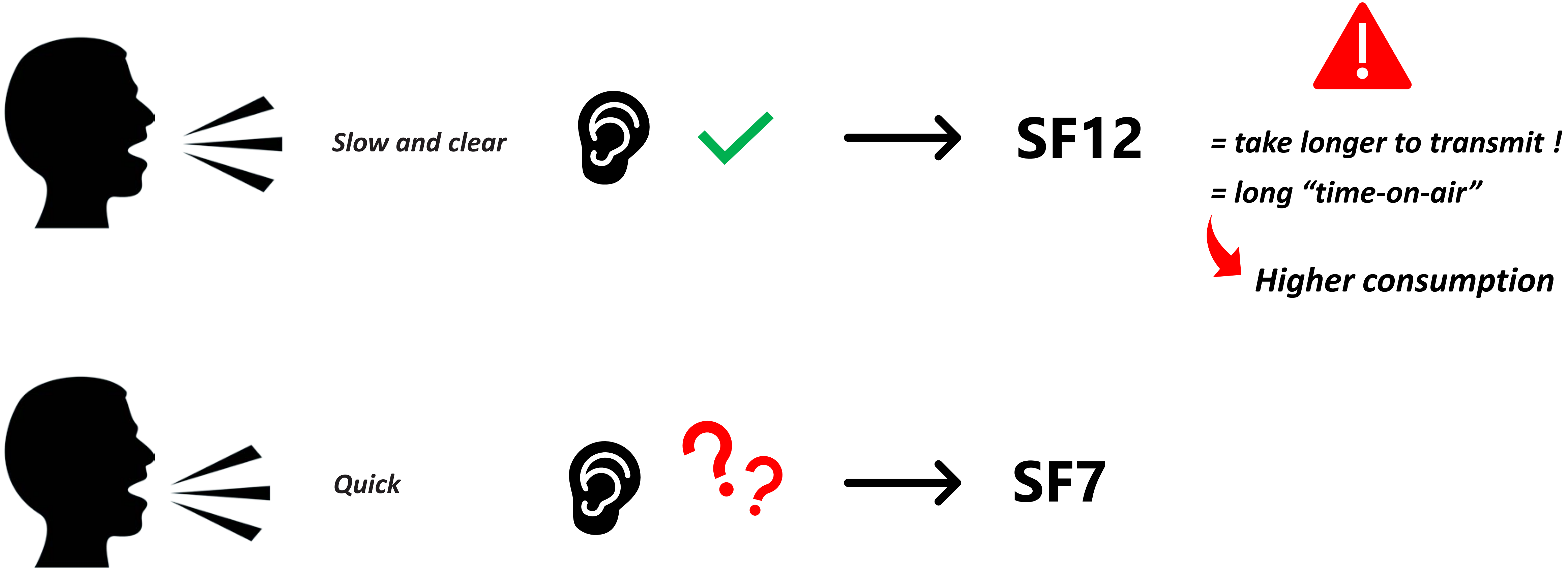
	ABP	OTAA
Security	✓ *Secured*	✓ *Secured*
	✗ *Keys re-generation*	✓ *Keys re-generation*
Frame Counter	✗ *Difficult (non volatile memory)*	✓ *Easy (automatically reset)*
LoRaWAN parameters	✗ *Difficult (to setup)*	✓ *Easy (nothing to do)*
Changing Network Operator	✓ *Feasible*	✓ *Feasible*
	✗ *No Roaming if changing operator*	✓ *Roaming*

How to increase the network coverage?

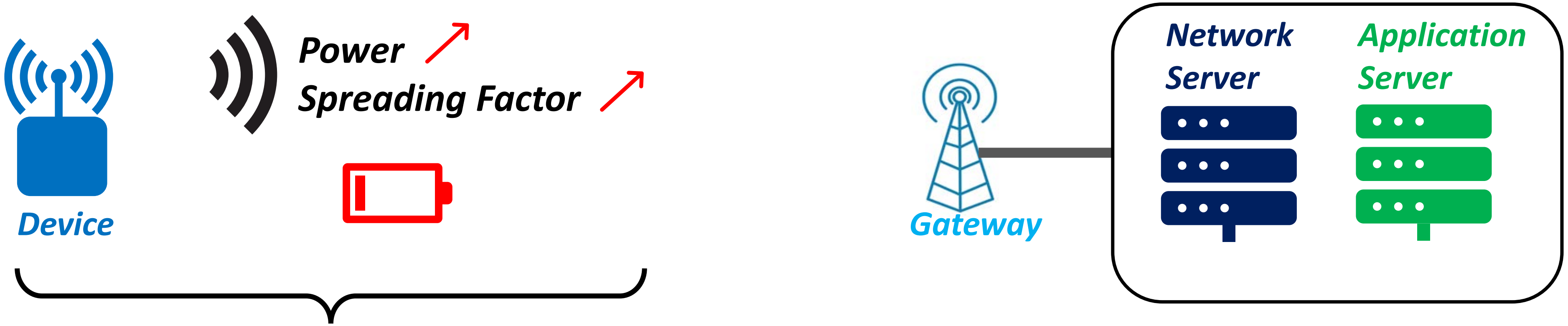
How to increase the network coverage?



How to increase the network coverage?

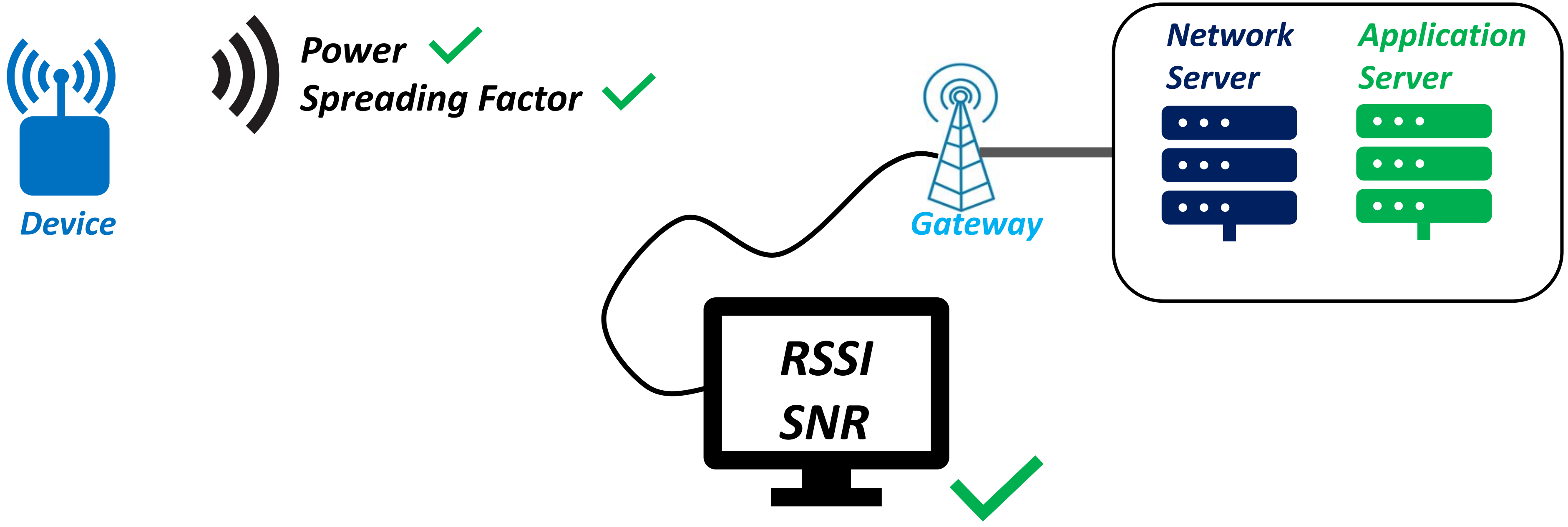


Adaptive Data Rate

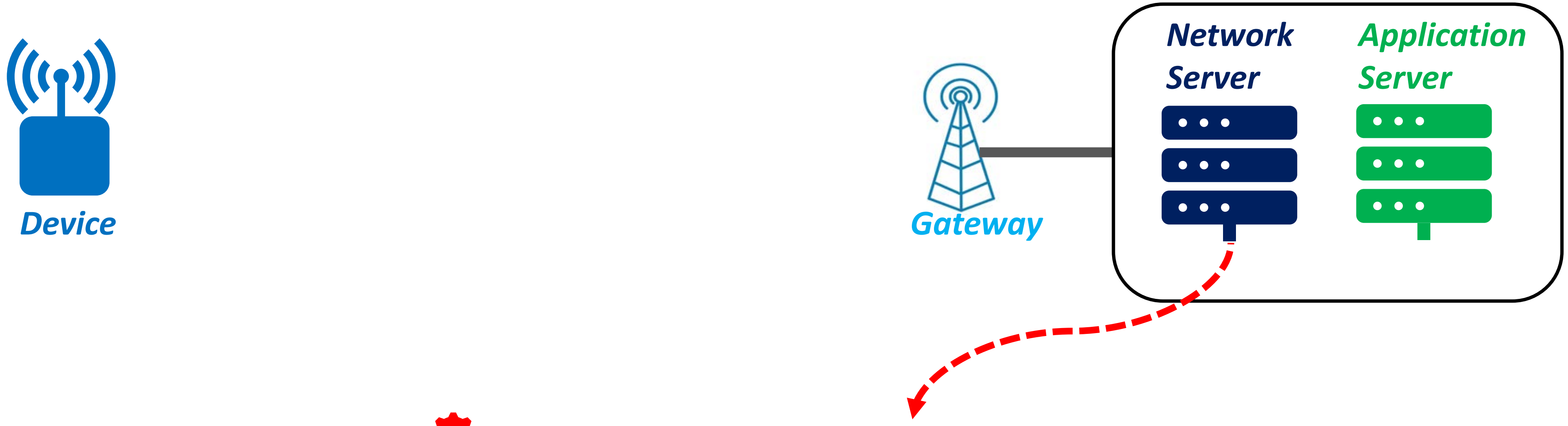


Power + SF: find the **best deal**

Adaptive Data Rate



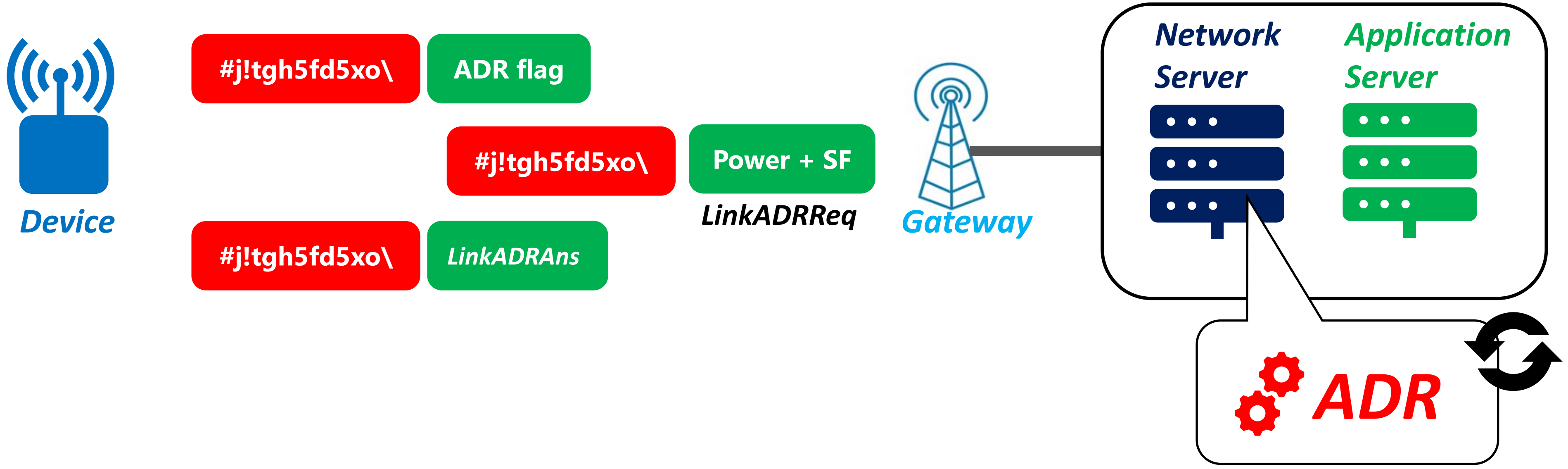
Adaptive Data Rate



ADR : Adaptive Data Rate

Power *to use*
Spreading Factor *to use*

Adaptive Data Rate

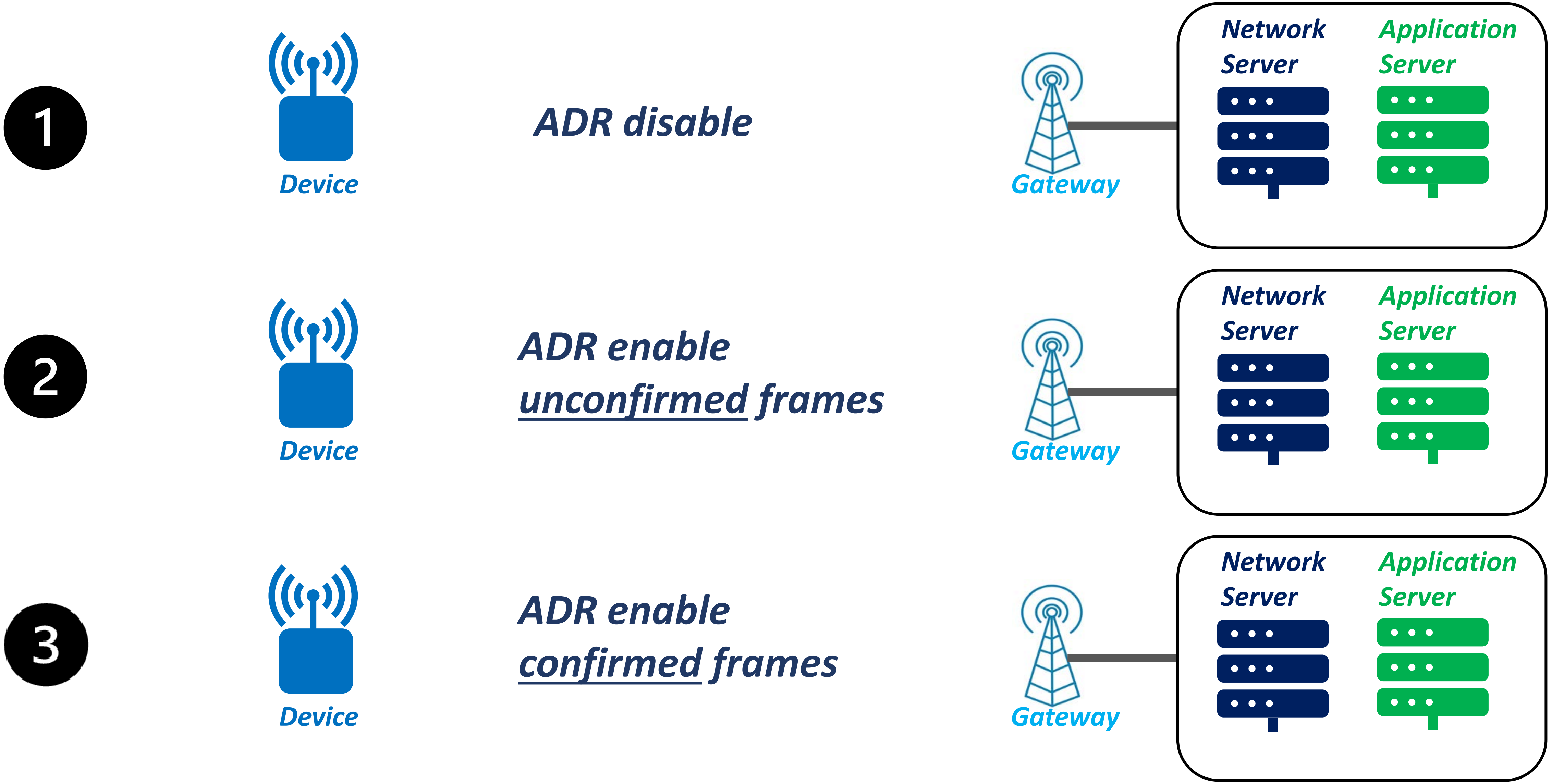


? *When are ADR commands sent ?*

Adaptive Data Rate - Demonstrations

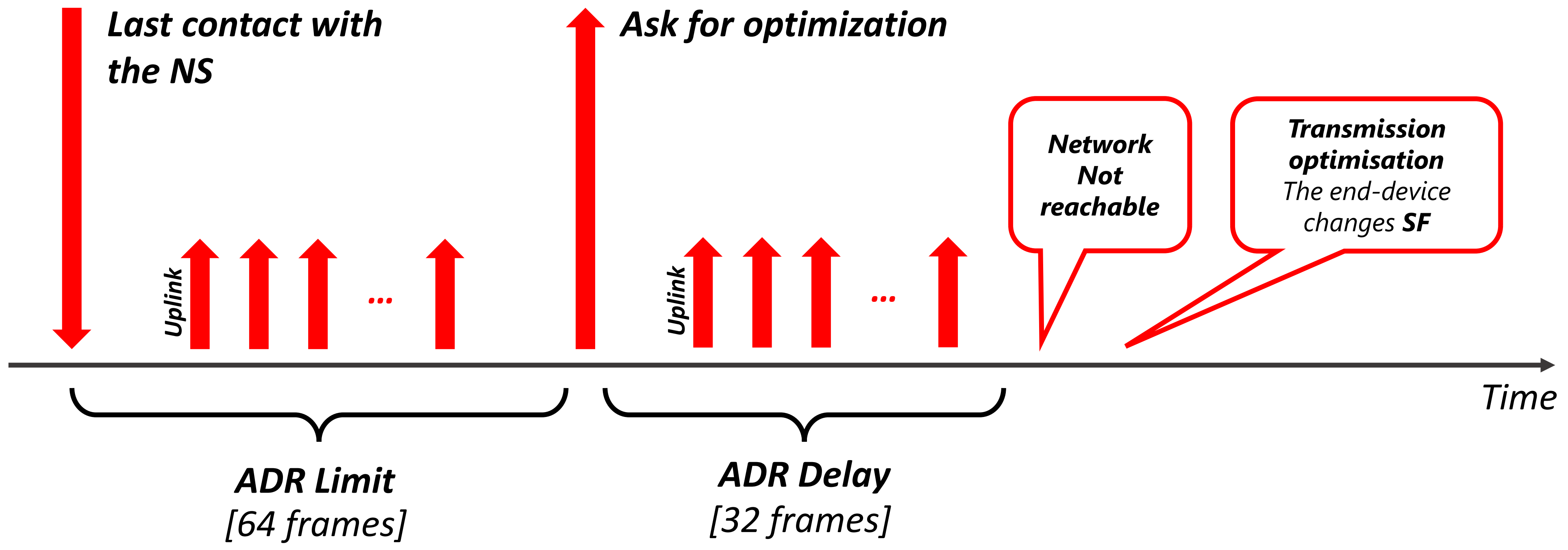
- ✓ ***OTAA***
- ✓ ***Send a frame every 5 sec***
- ✓ ***SF9***

Adaptive Data Rate - Demonstrations



Adaptive Data Rate - Demonstrations

End-device side

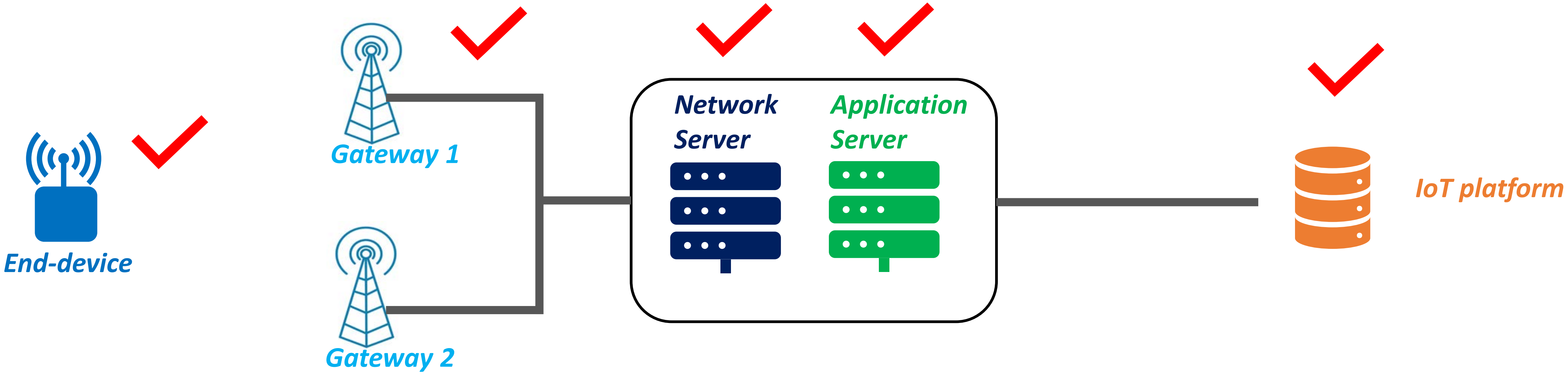


For the demo → *[8 frames]* *[4 frames]* → ***Optimization*** *after 8 + 4 =* ***12 frames***

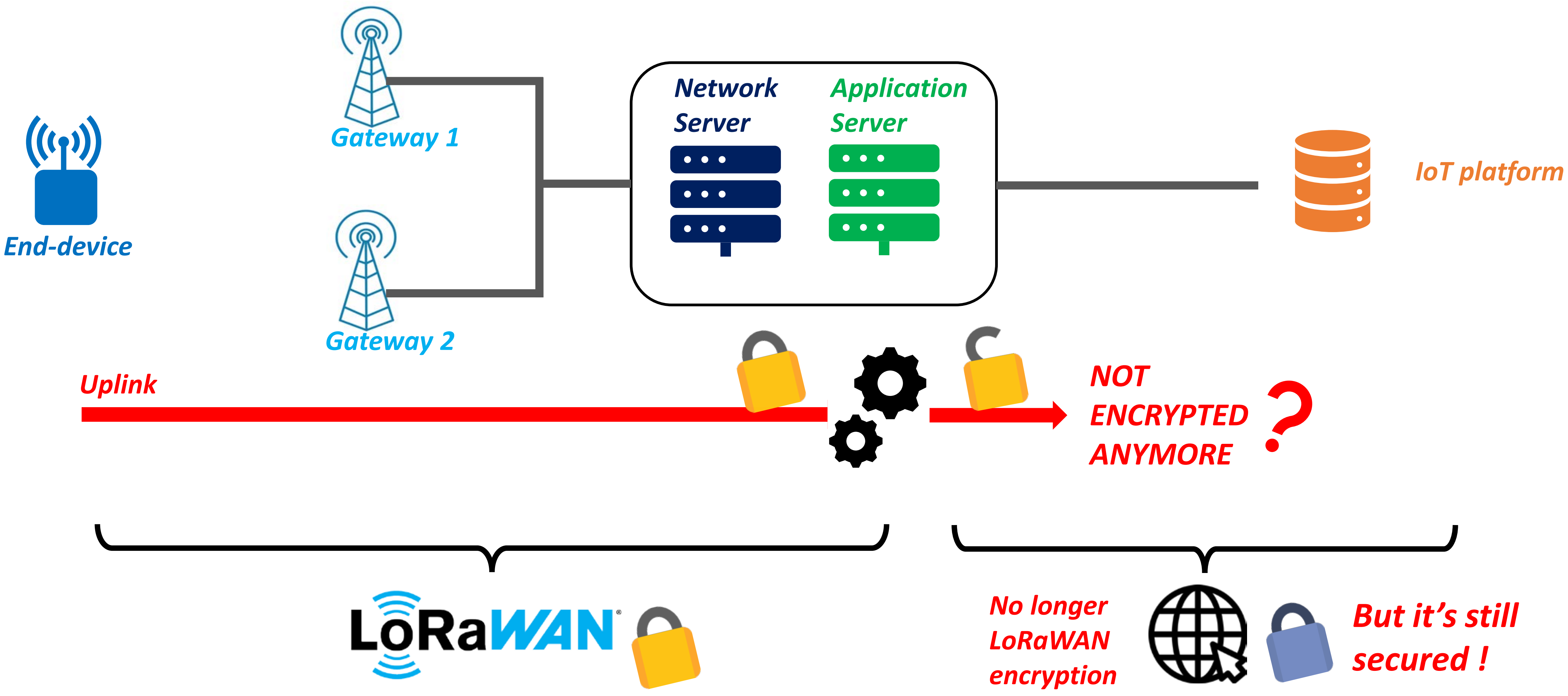
LoRaWAN Networks

- ✓ LoRaWAN Networks: different architectures
- ✓ Presentation of each architecture
- ✓ Configuration of a LoRaWAN Network

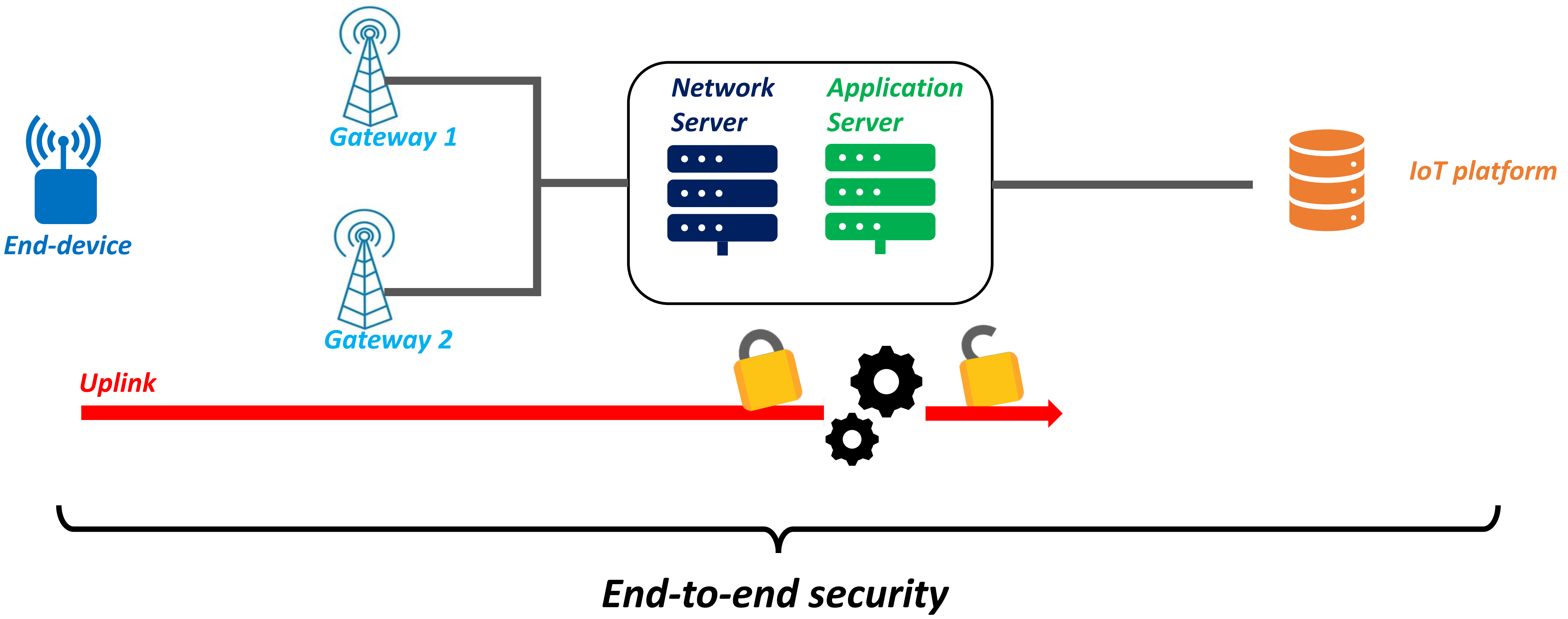
LoRaWAN Networks – End to end security



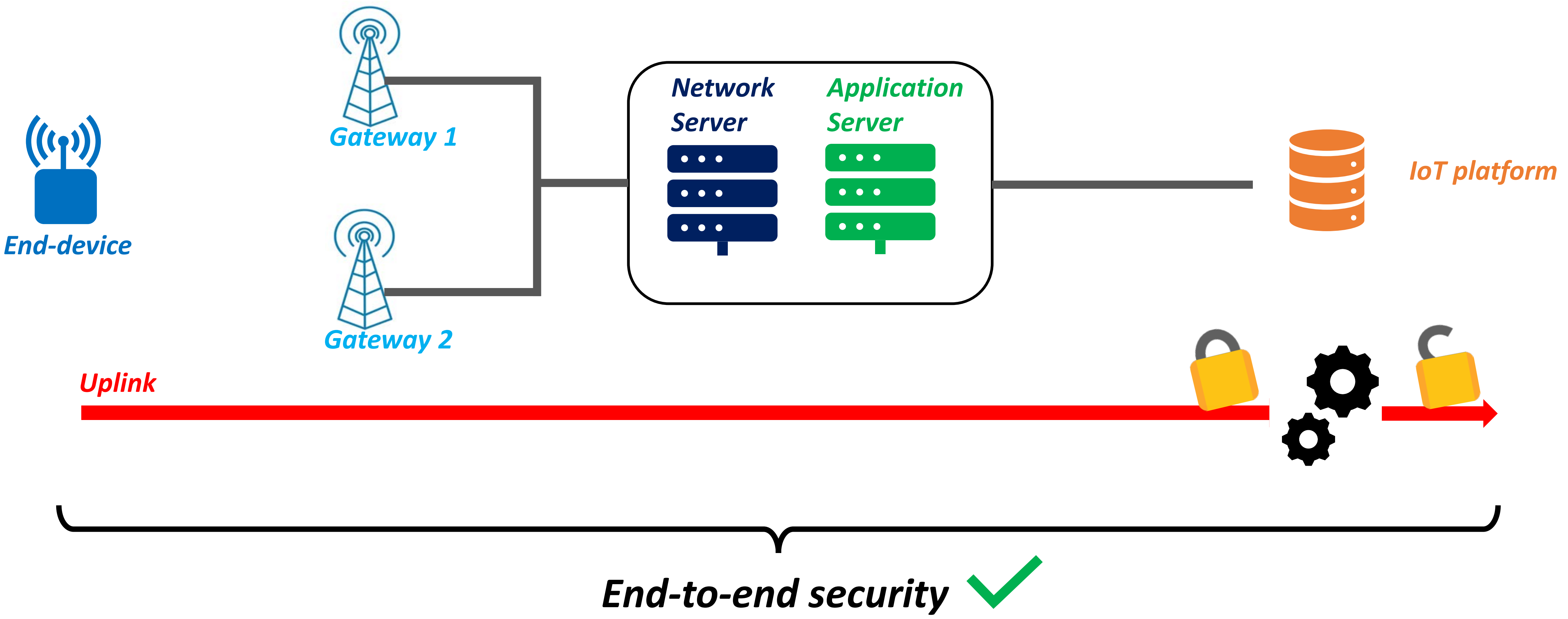
LoRaWAN Networks – End to end security



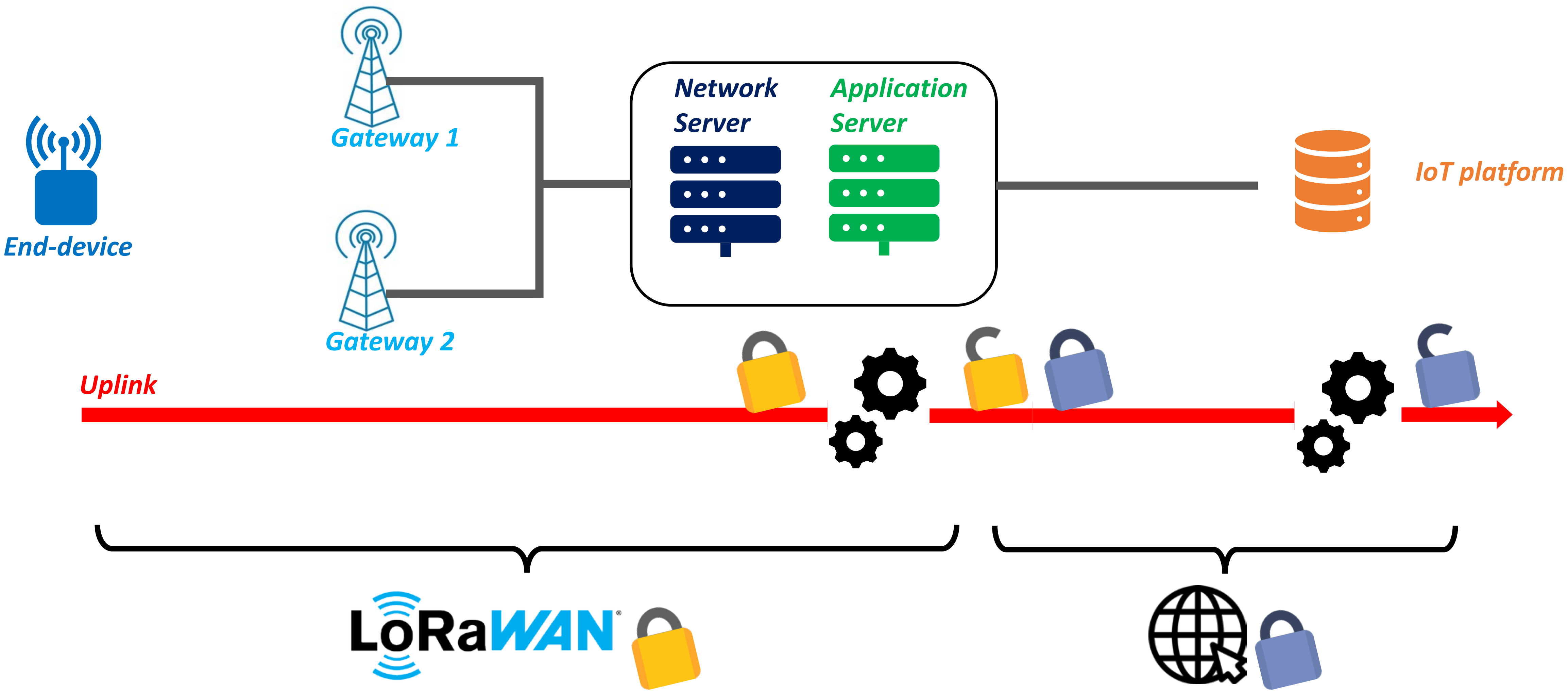
LoRaWAN Networks – End to end security



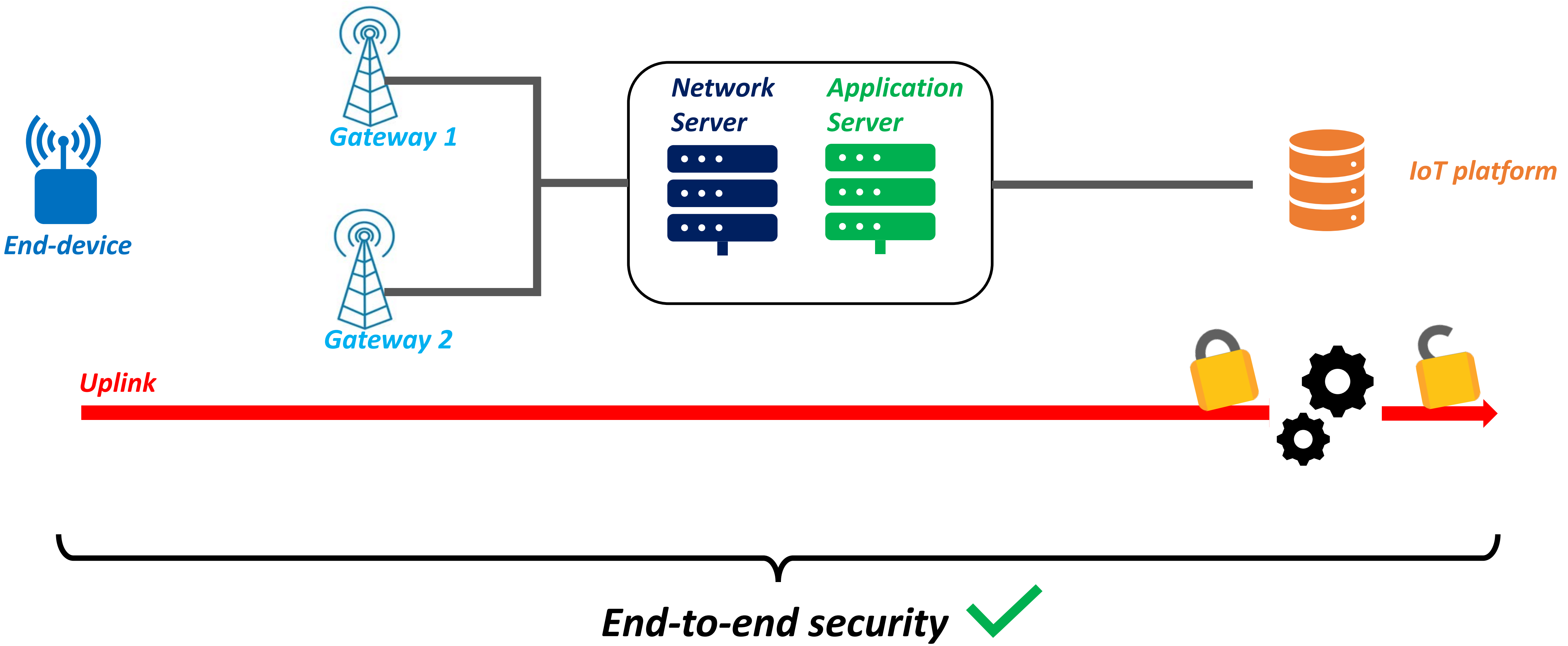
LoRaWAN Networks – End to end security



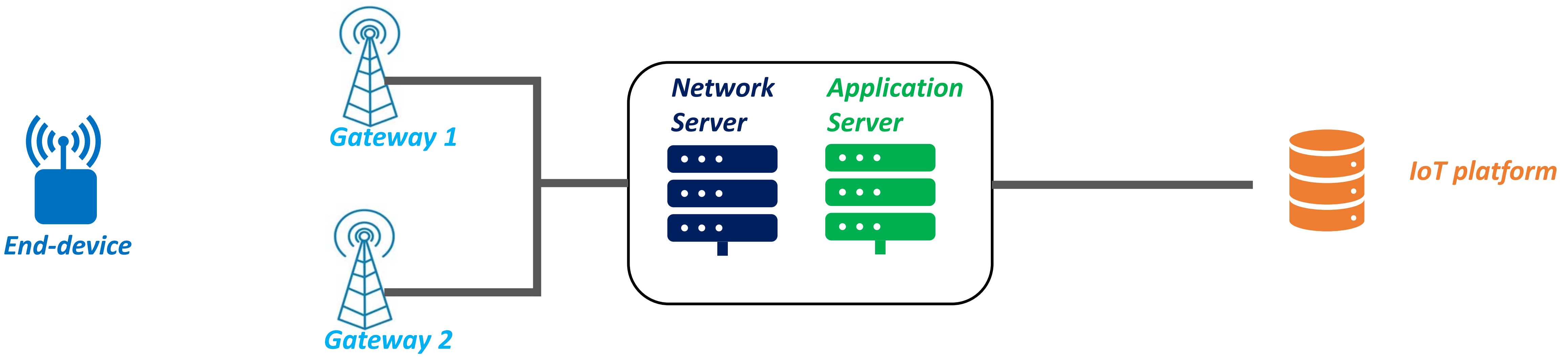
LoRaWAN Networks – End to end security



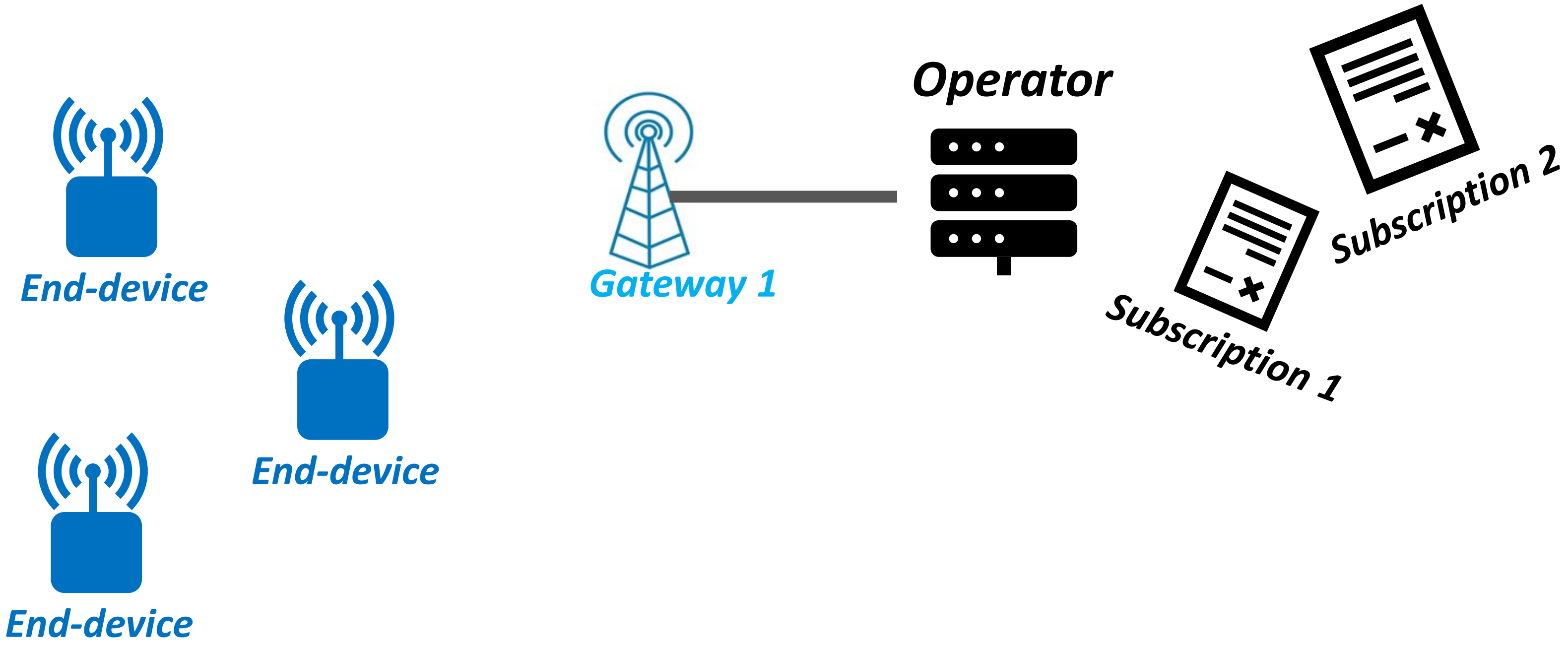
LoRaWAN Networks – End to end security



LoRaWAN Networks – End to end security



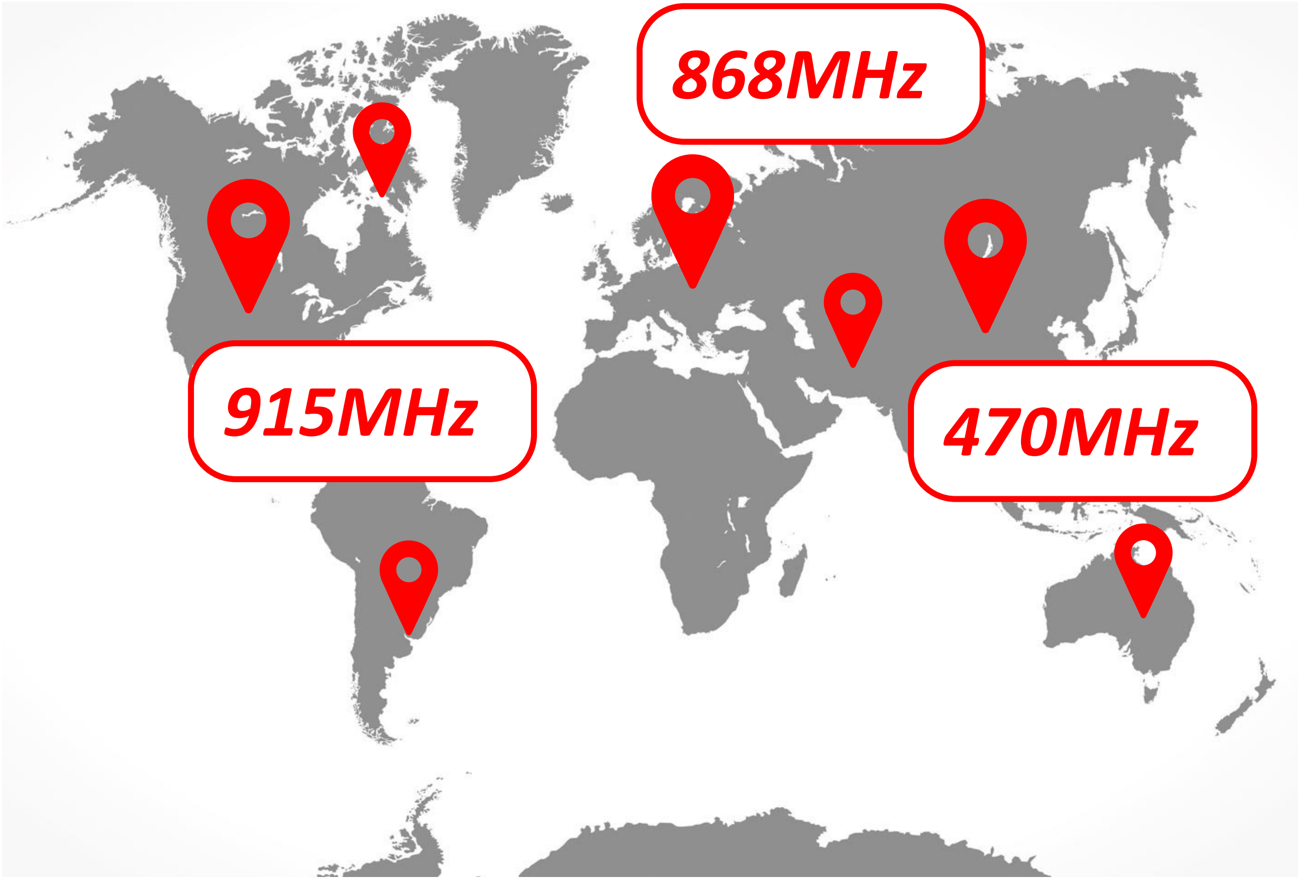
LoRaWAN Networks



LoRaWAN = *Free bands !*

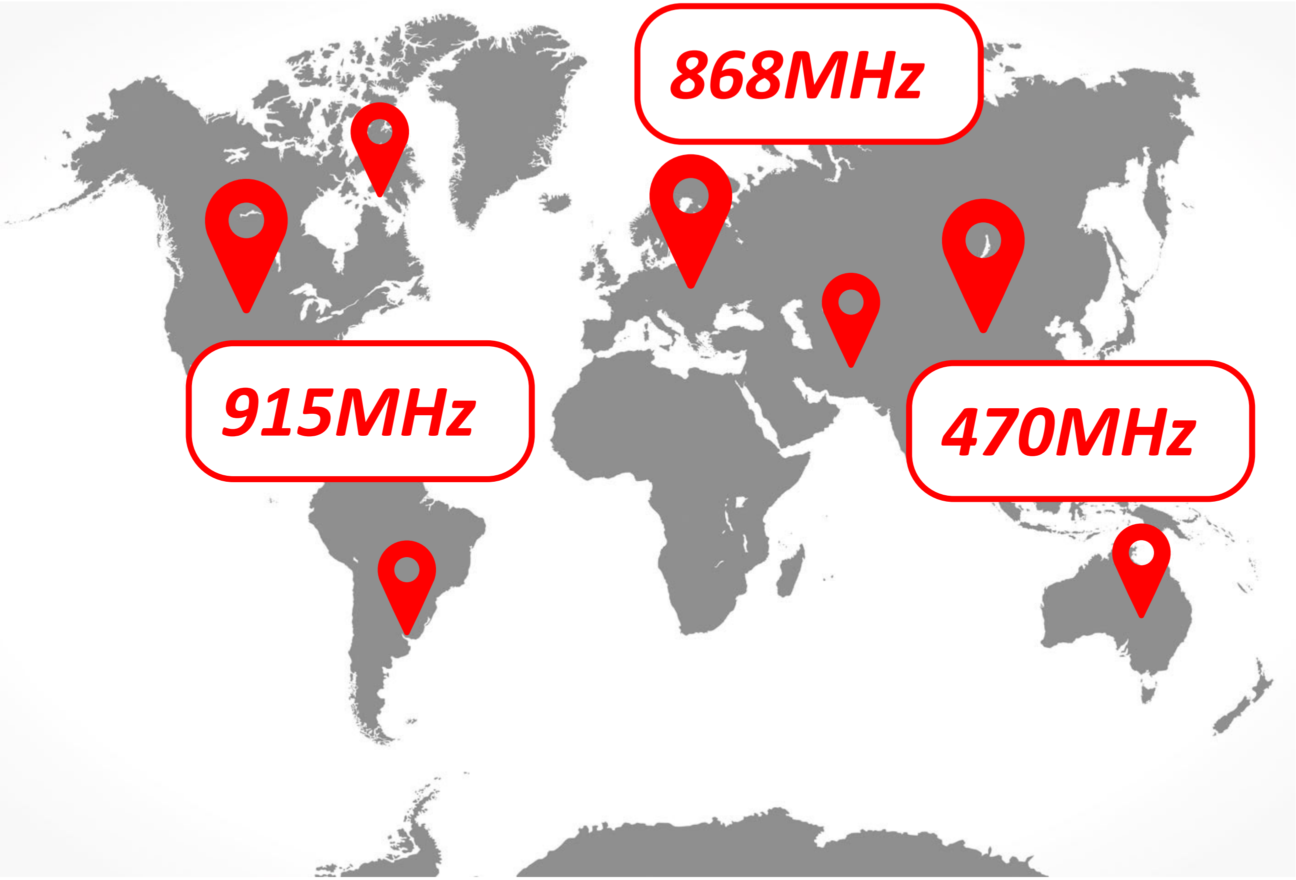
LoRaWAN Networks

LoRaWAN = *Free bands !*

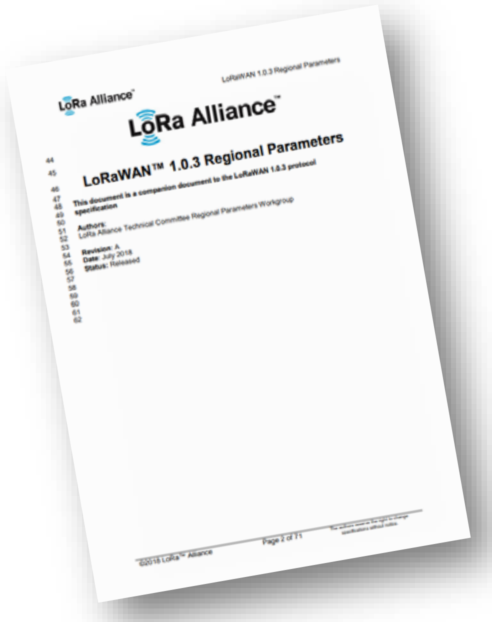


LoRaWAN Networks

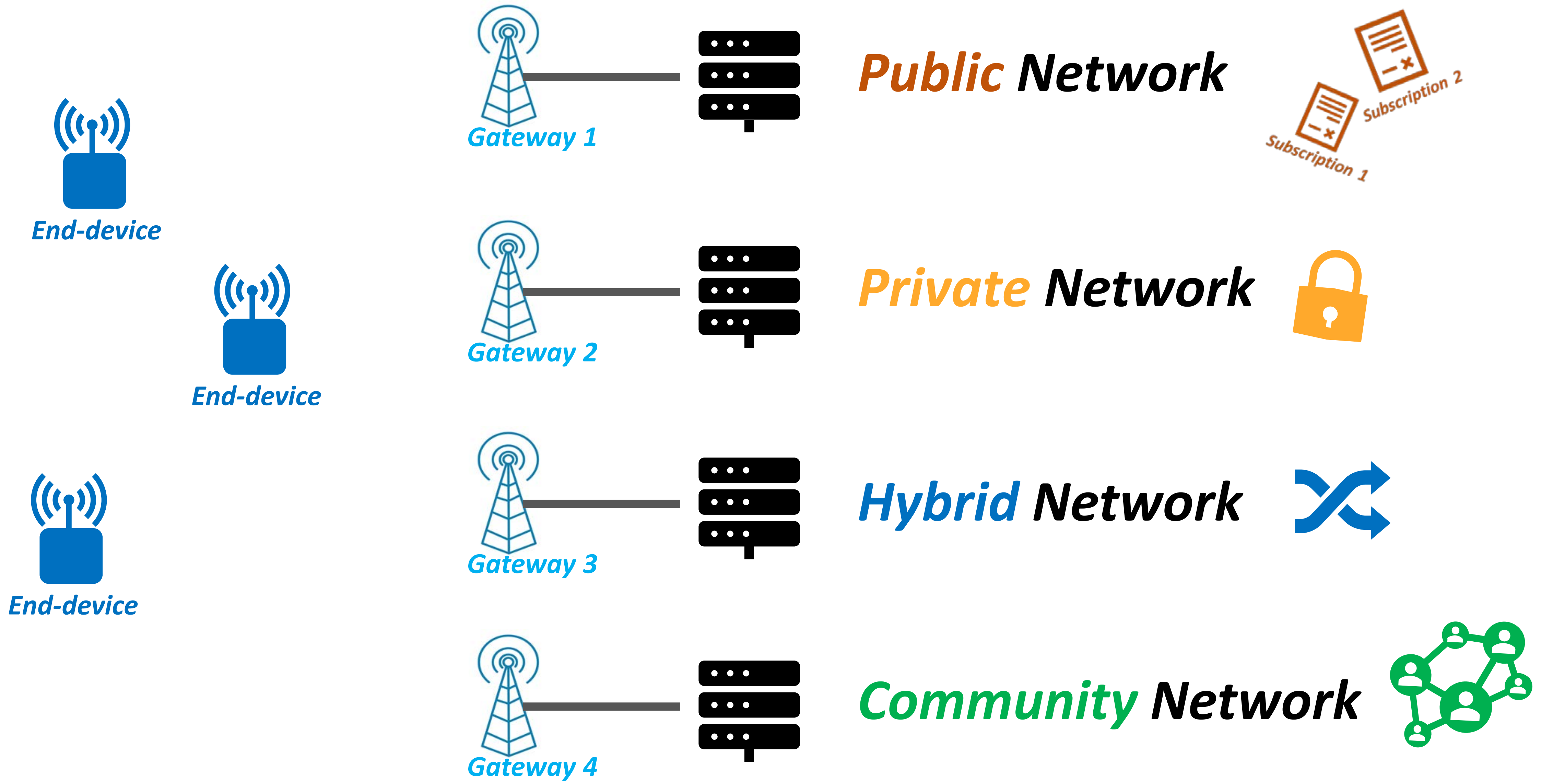
LoRaWAN = ***Free bands !***



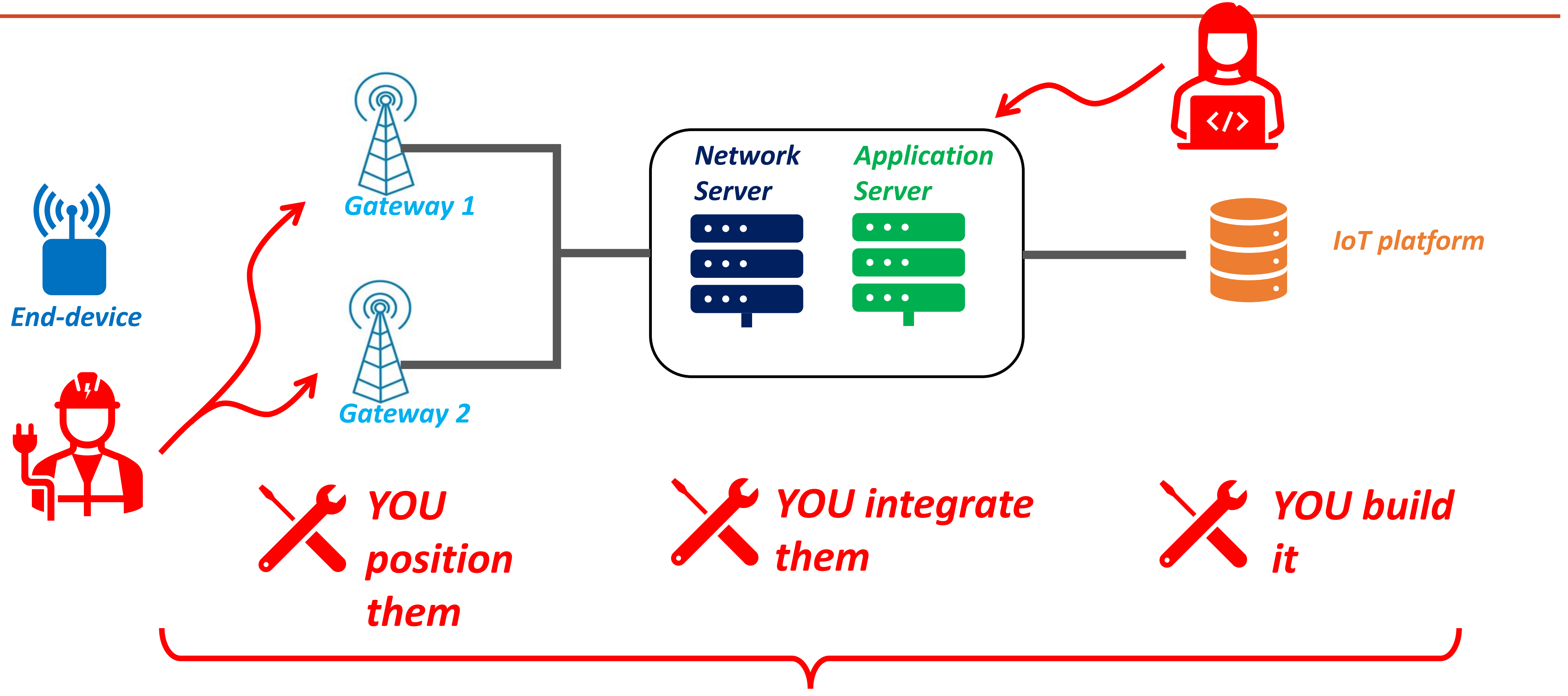
Regional Parameters



LoRaWAN Networks

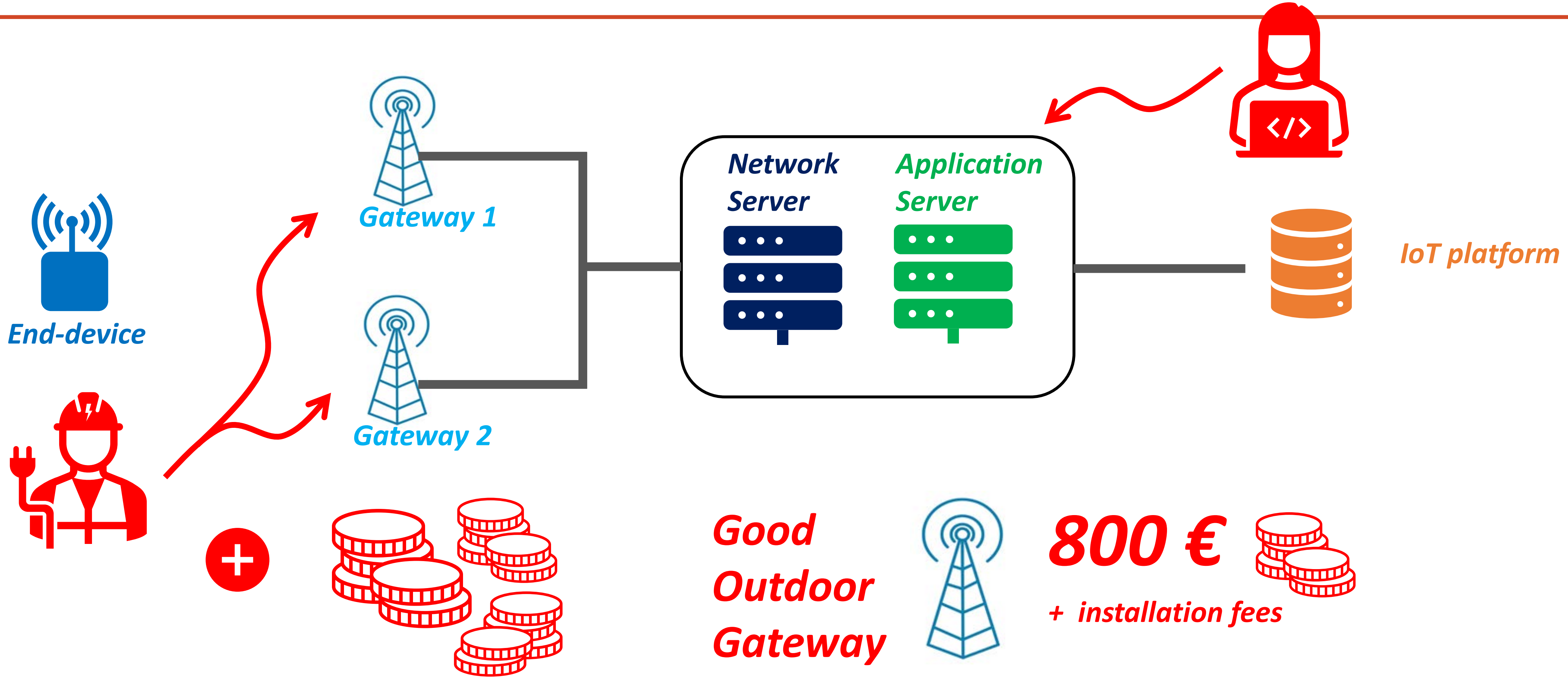


LoRaWAN Networks – Private Network



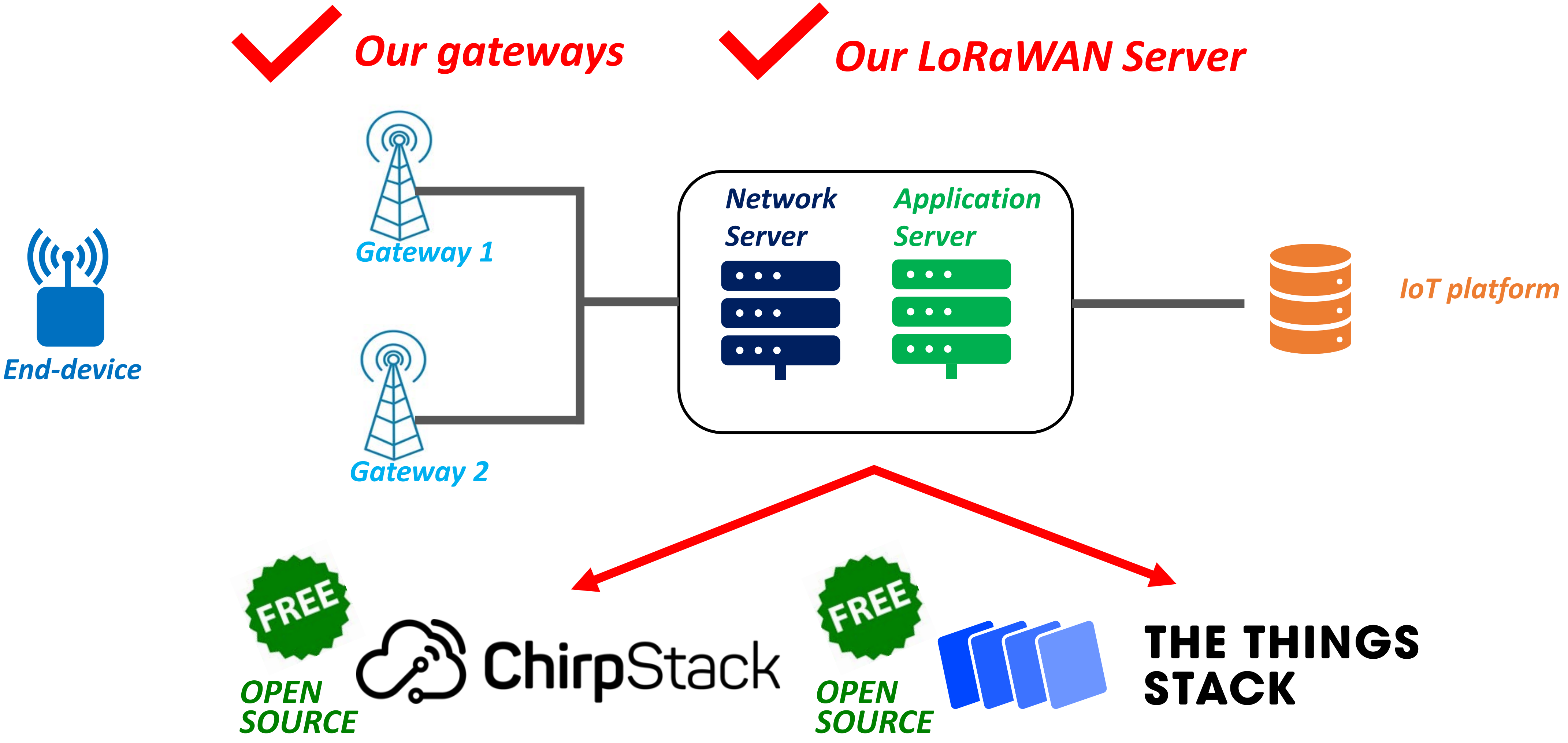
Private Network

LoRaWAN Networks – Private Network



BUT, No subscriptions + No restrictions

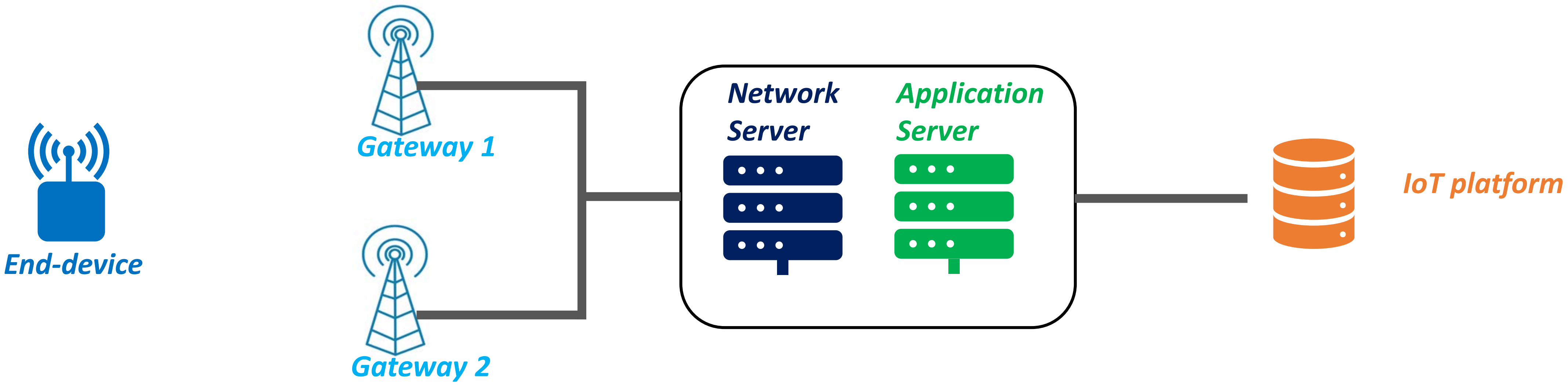
LoRaWAN Networks – Private Network



LoRaWAN Networks – Private Network

Our gateways

Our LoRaWAN Server



FREE

OPEN SOURCE

ChirpStack

FREE

OPEN SOURCE

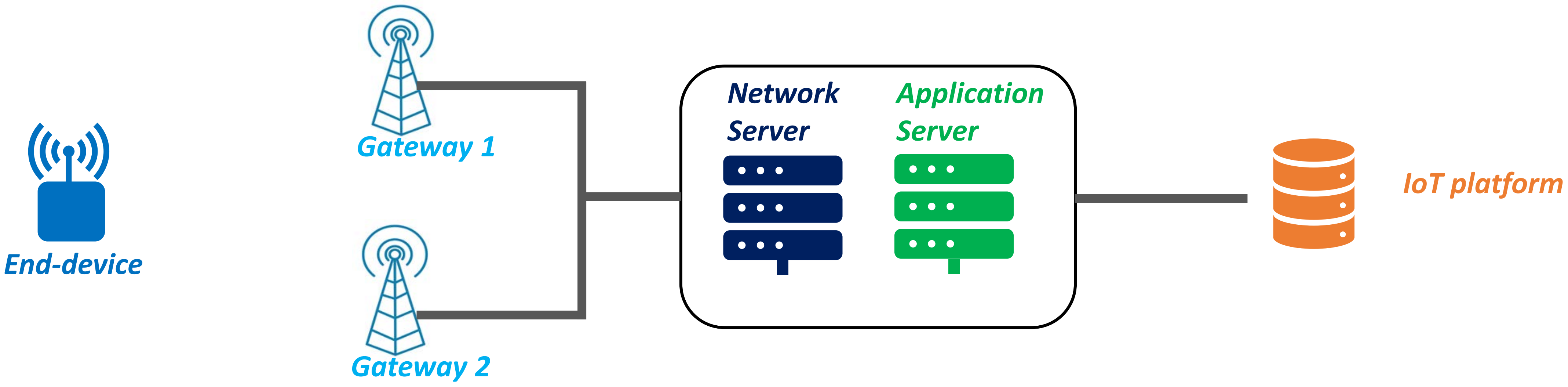
THE THINGS STACK

PROPRIETARY

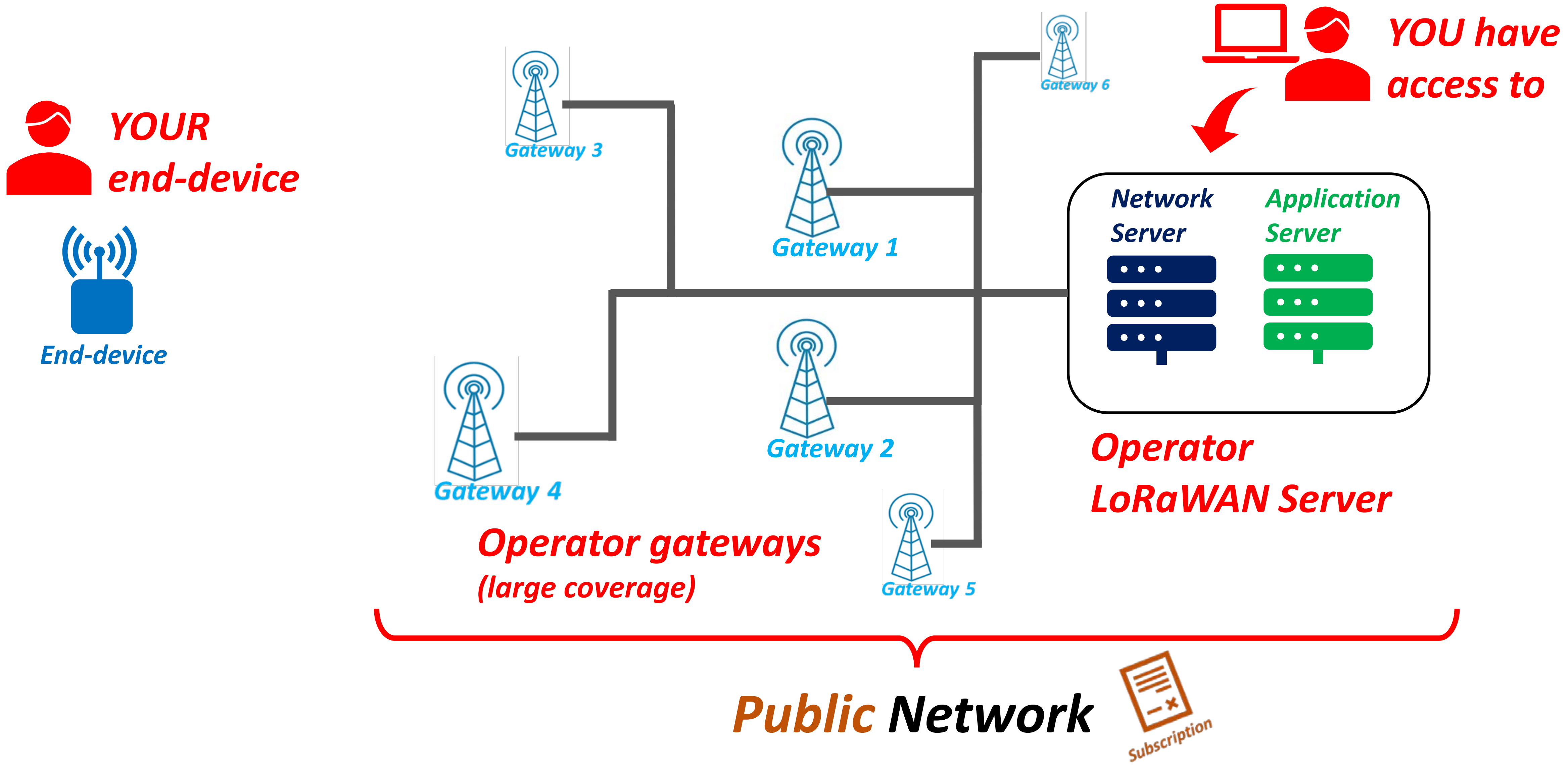
Actility

Connecting with intelligence

LoRaWAN Networks – Public Network

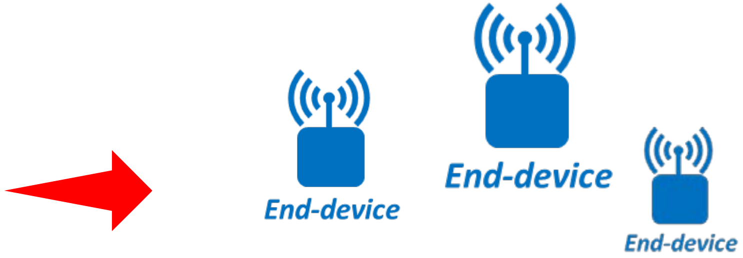


LoRaWAN Networks – Public Network

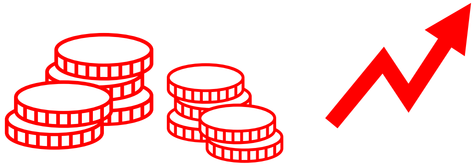


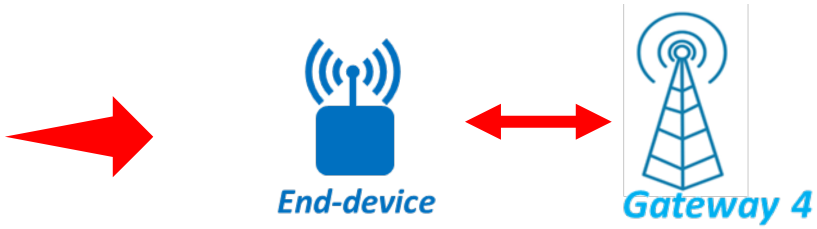
LoRaWAN Networks – Public Network

Public Network

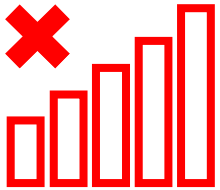


Subscriptions





Lack of coverage



LoRaWAN Networks – Public Network

***Public* Network**

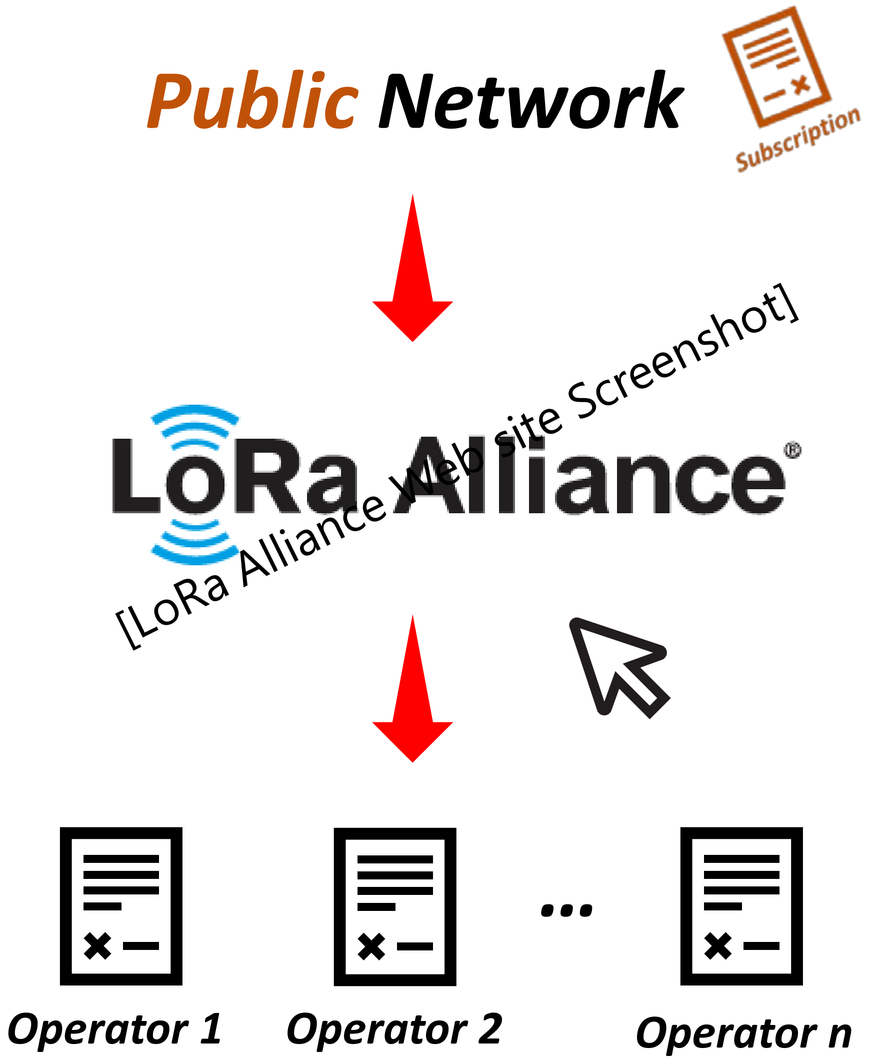
Subscriptions
Cost

Depends on number of <u>uplinks</u>

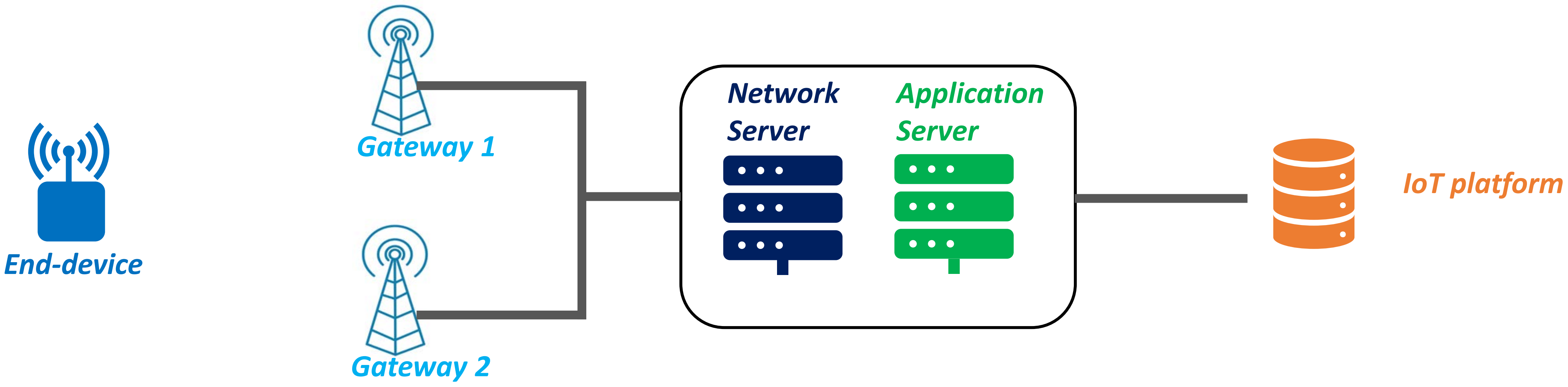
Depends on number of <u>downlinks</u>

1,50€ per month and per device

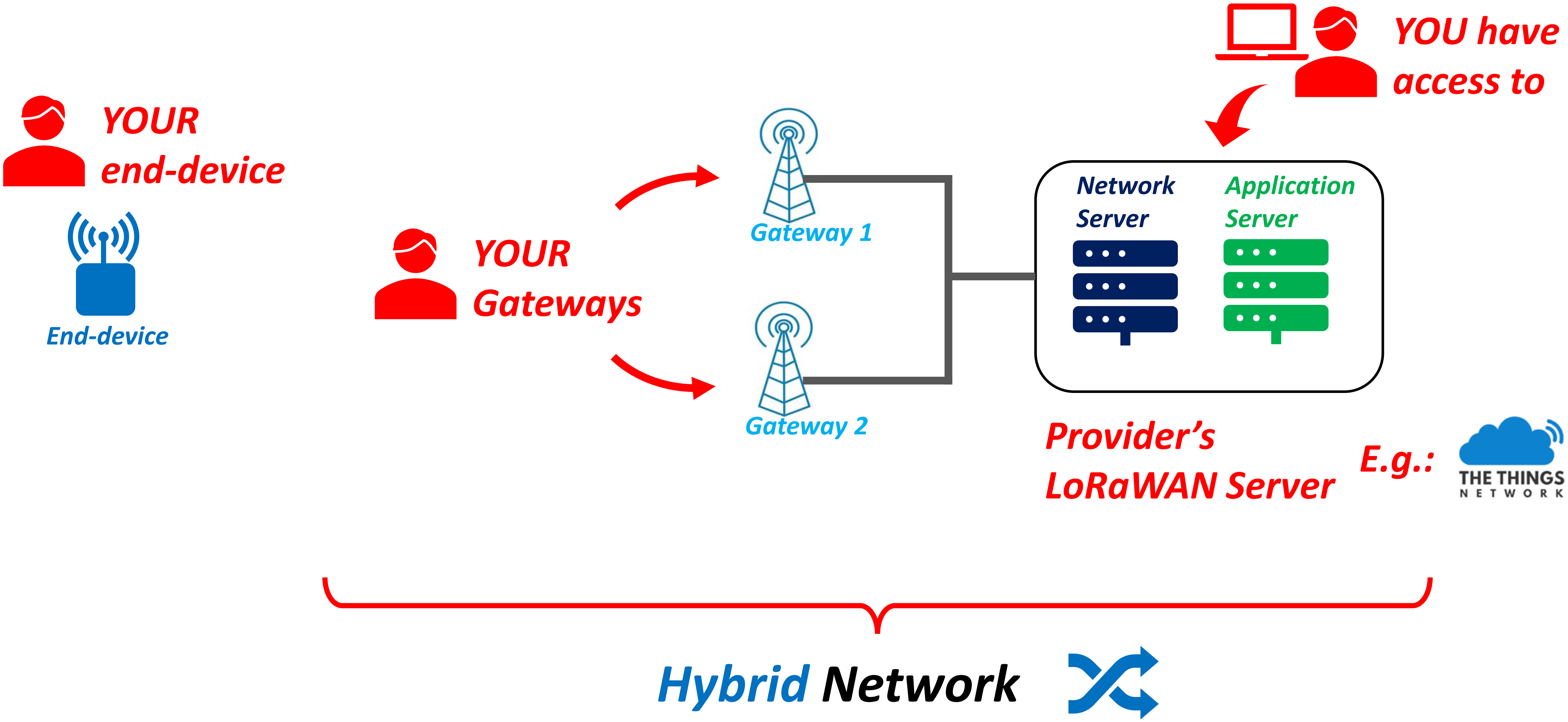
LoRaWAN Networks – Public Network



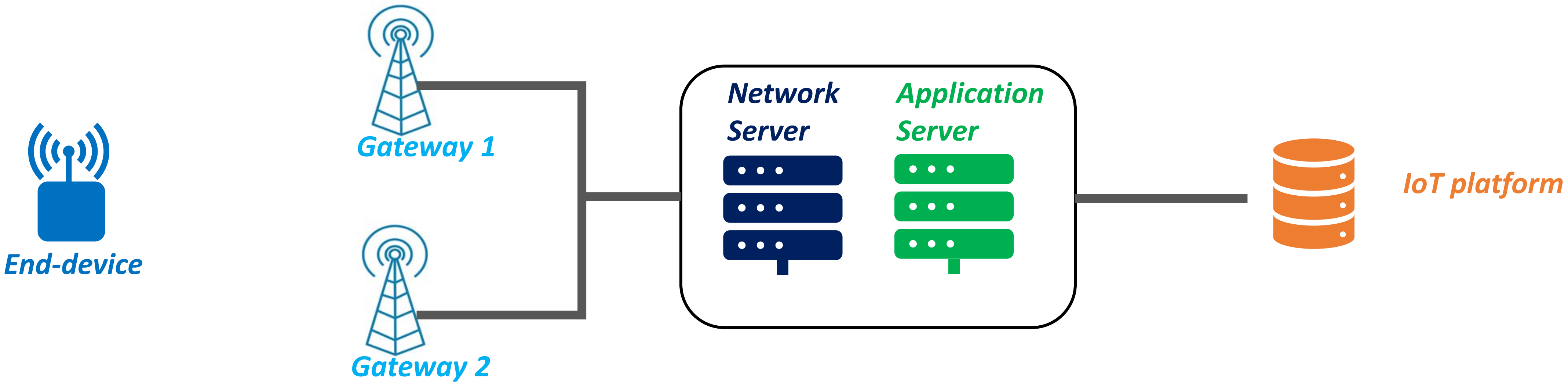
LoRaWAN Networks – Hybrid Network



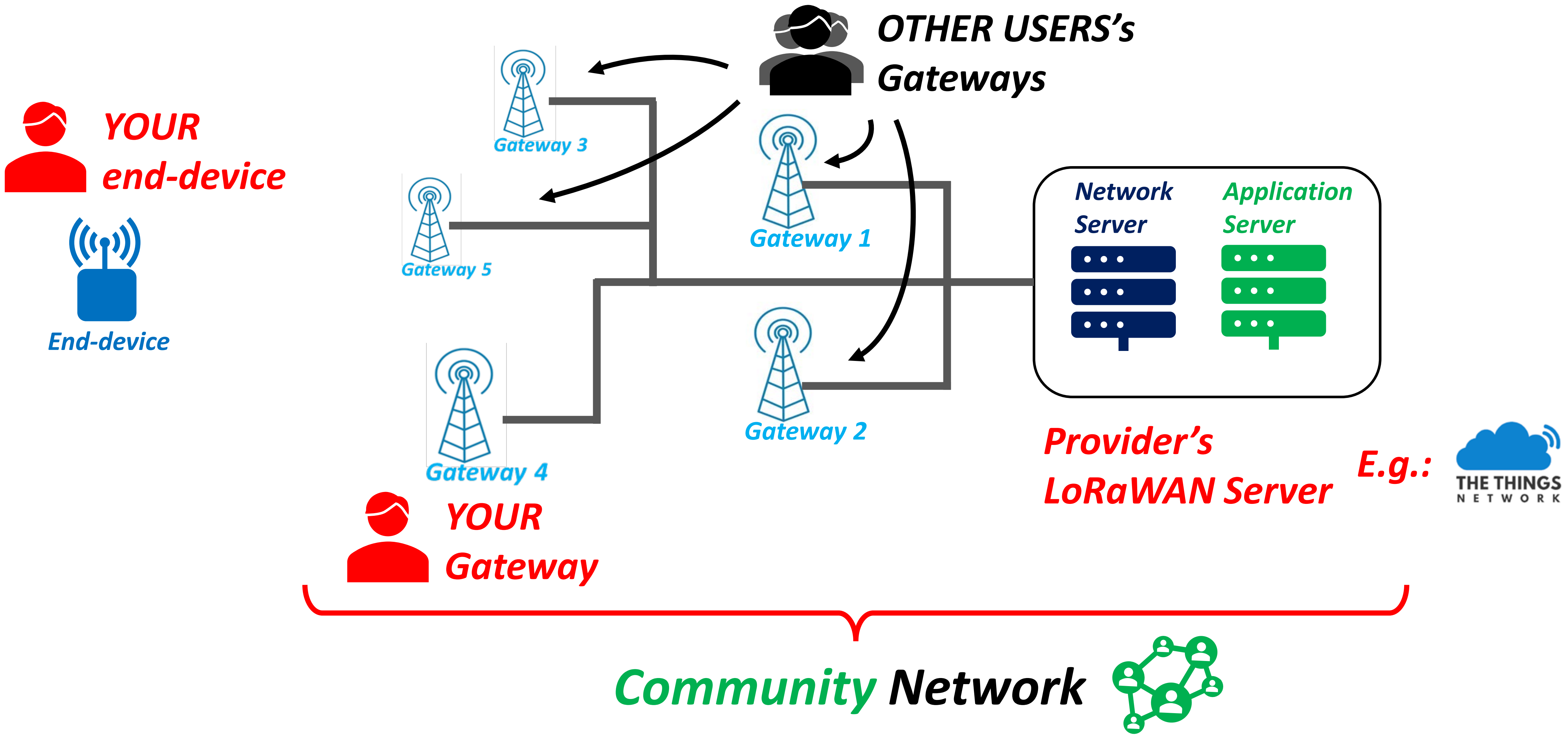
LoRaWAN Networks – Hybrid Network



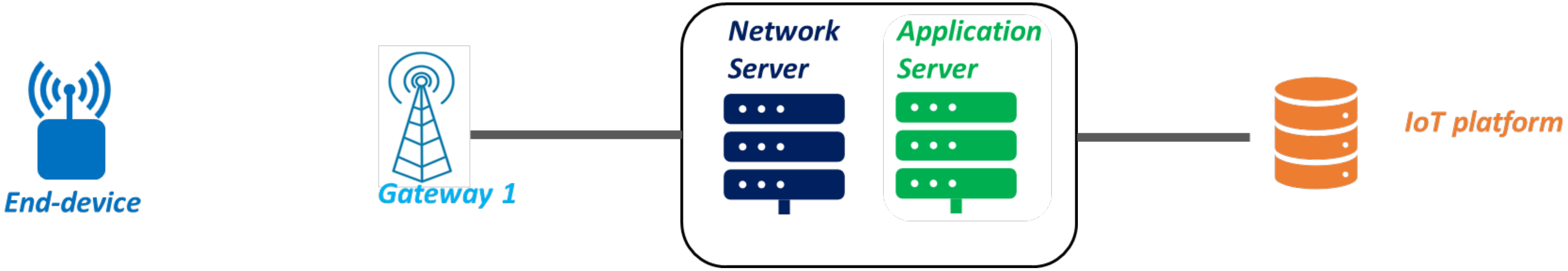
LoRaWAN Networks – Community Network



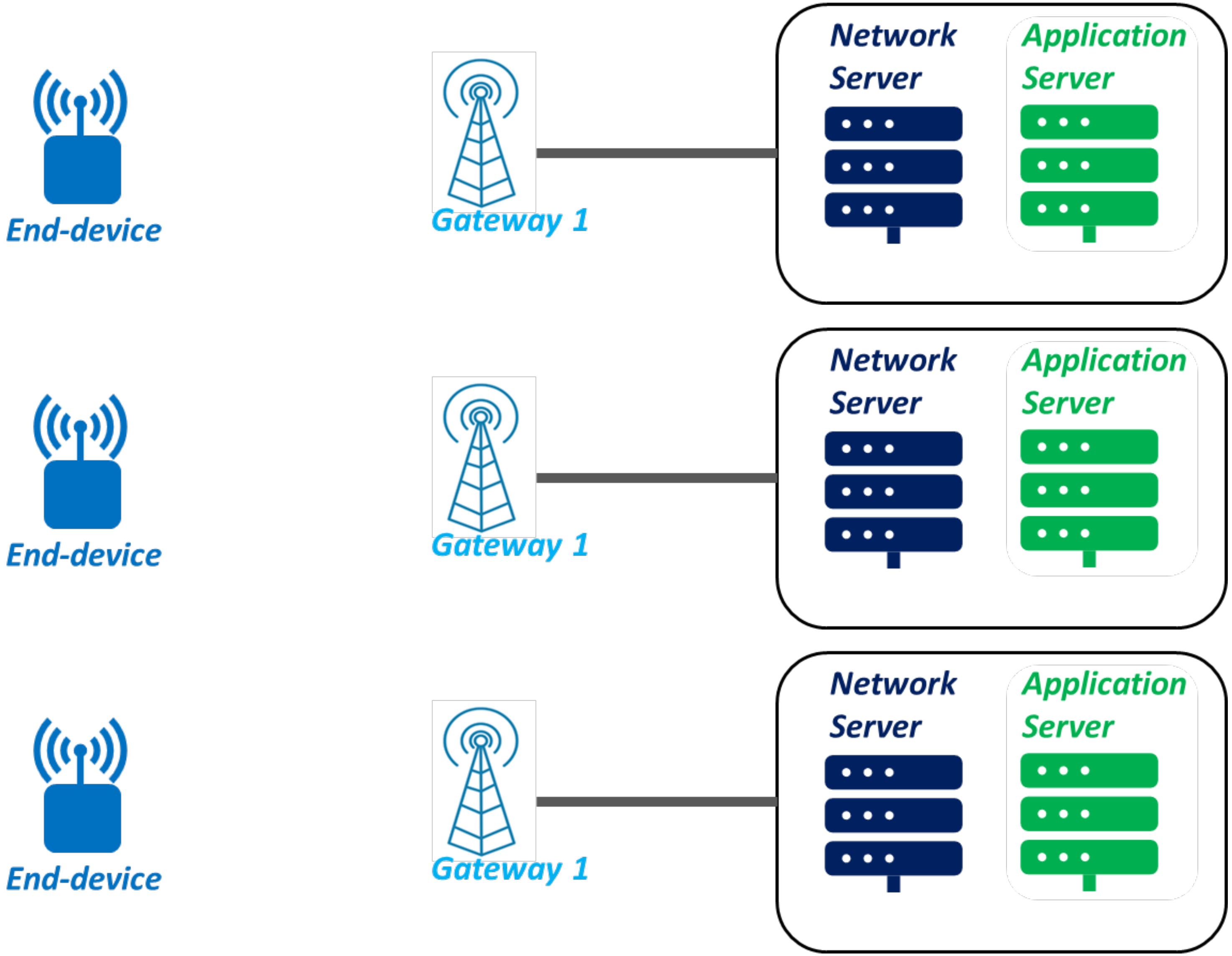
LoRaWAN Networks – Community Network



Network Configuration



Network Configuration



Private Network

- *Gateway configured*
- *LoRaWAN Server installed*

Hybrid Network

- *Gateway configured*
- *Subscribe to a Network Provider*

Can be a ***Community Network***

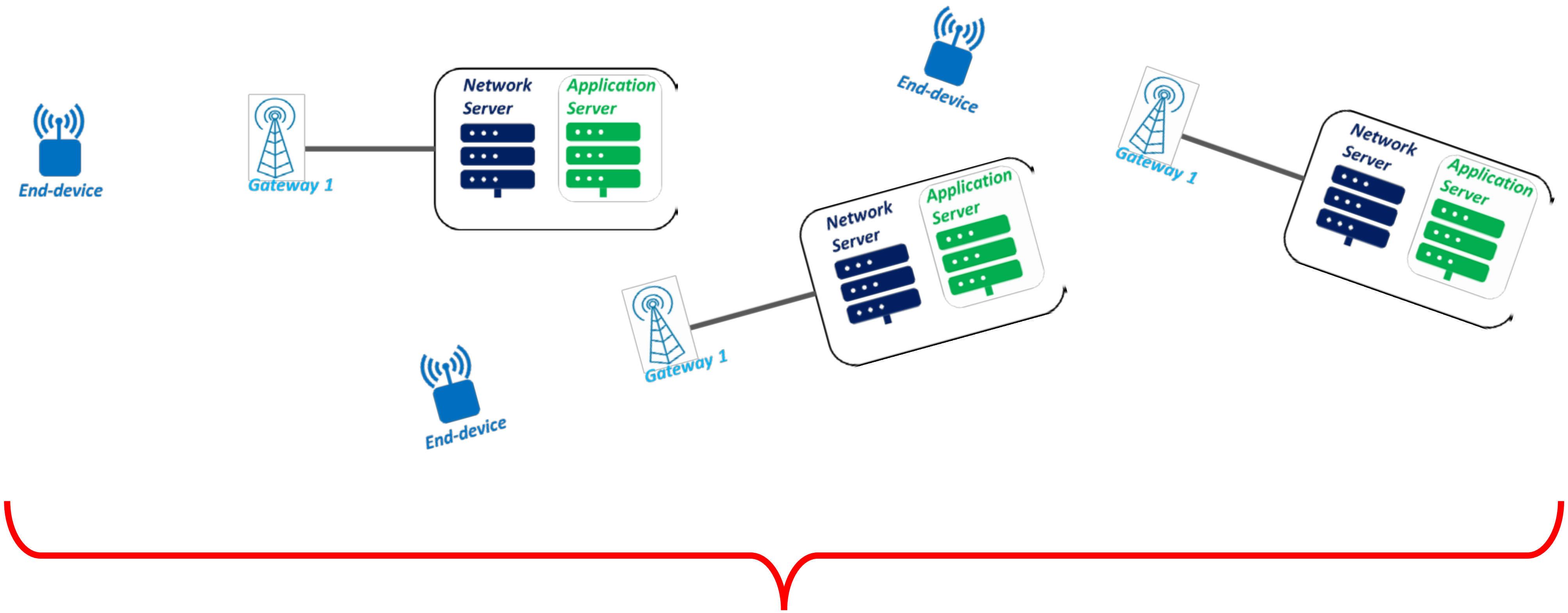
(Check the coverage if you don't use your own Gateway)

Public Network

- *Subscribe to a LoRaWAN operator*

(Check the coverage)

Network Configuration



4 steps configuration

*If you use someone else Gateways (**Community** or **Public** Network): only **2 steps***

Network Configuration

Step 1

Gateway configuration

Packet Forwarders

Semtech Packet Forwarder

Packet Forwarder software

LoRa Basic Station

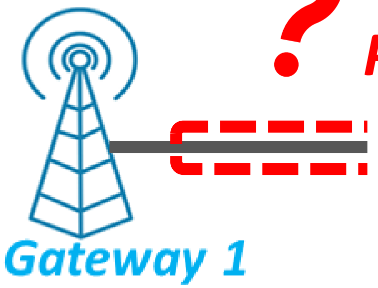
LRR

Loriot Packet Forwarder

…

Compatibility with the Network Server

The Protocol



LoRaWAN Server Target

Network Configuration

Step 1

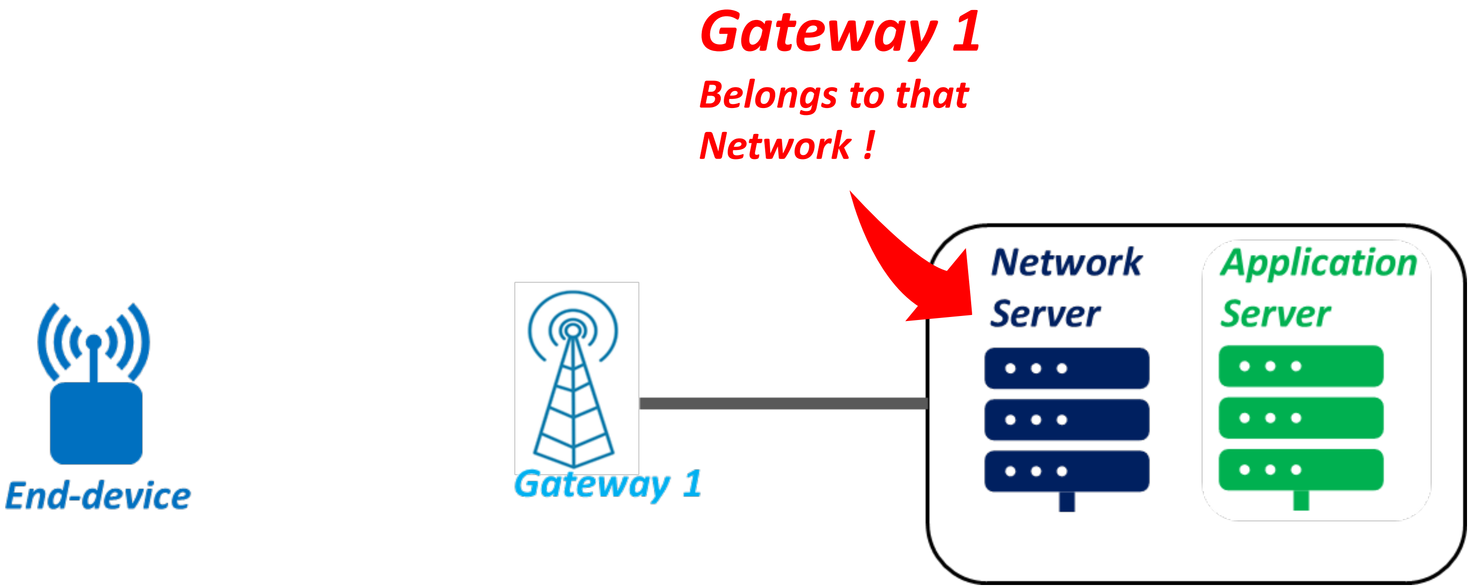
Gateway configuration

- ***Network Server IP Address***
- ***UDP/TCP port***
- ***Keys & Certificates (for secure transmission)***

Network Configuration

Step 1
Gateway configuration

Step 2
Gateway registration

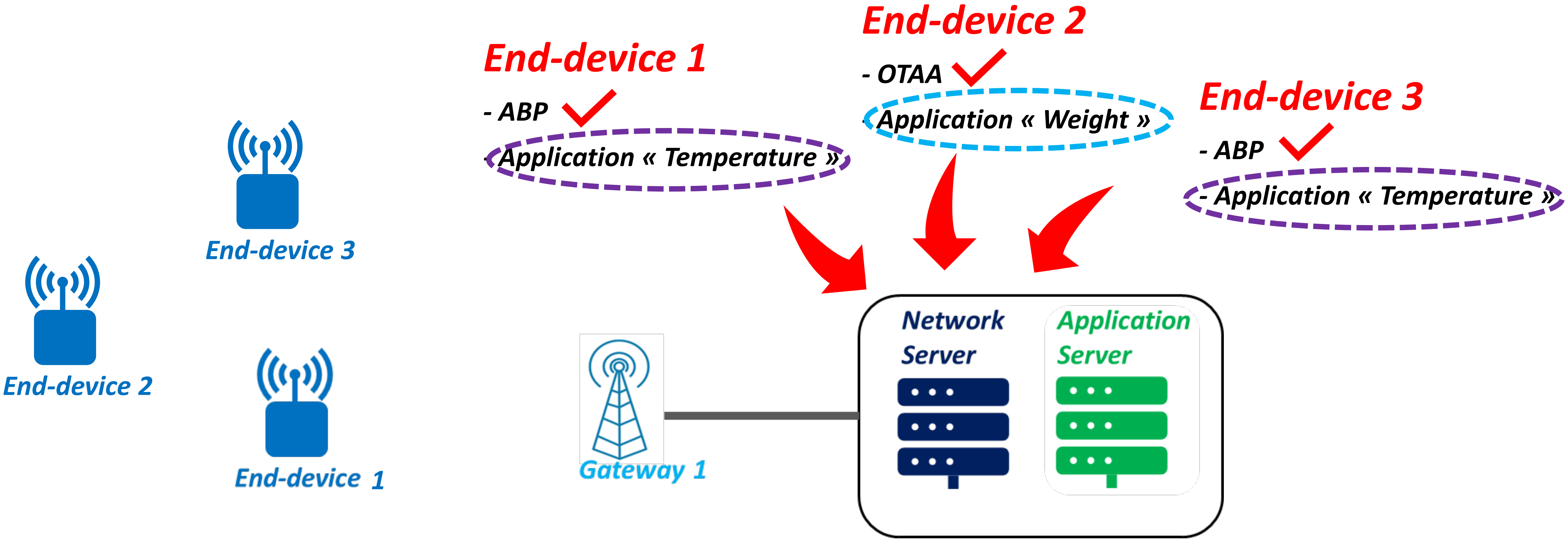


Network Configuration

Step 1
Gateway configuration

Step 2
Gateway registration

Step 3
Device registration



Network Configuration

Step 1 *Gateway configuration*

Step 2 *Gateway registration*

Step 3 *Device registration*

Step 4 *Device configuration*

Step 1: Gateway configuration

1.a. Choose the Packet Forwarder and install it on the Gateway

Packet Forwarder Compatibility

Step 1: Gateway configuration

1.a. Choose the Packet Forwarder and install it on the Gateway

→ *Semtech UDP Packet Forwarder*

1.b. Set the Network Server IP address

1.c. Set the UDP Ports

Packet Forwarder

Compatibility

Step 2: Gateway registration

Step 1: Gateway configuration ✓

Semtech UDP Packet forwarder is used

Step 2: Gateway registration

2.a. Gateway EUI

2.b. Name or ID (or both)

2.c. Region (frequency plan)

Unique Identifier

(This number comes with the Gateway)

- ***Most of the time written on the Gateway***
- ***Can be found on the Gateway interface***

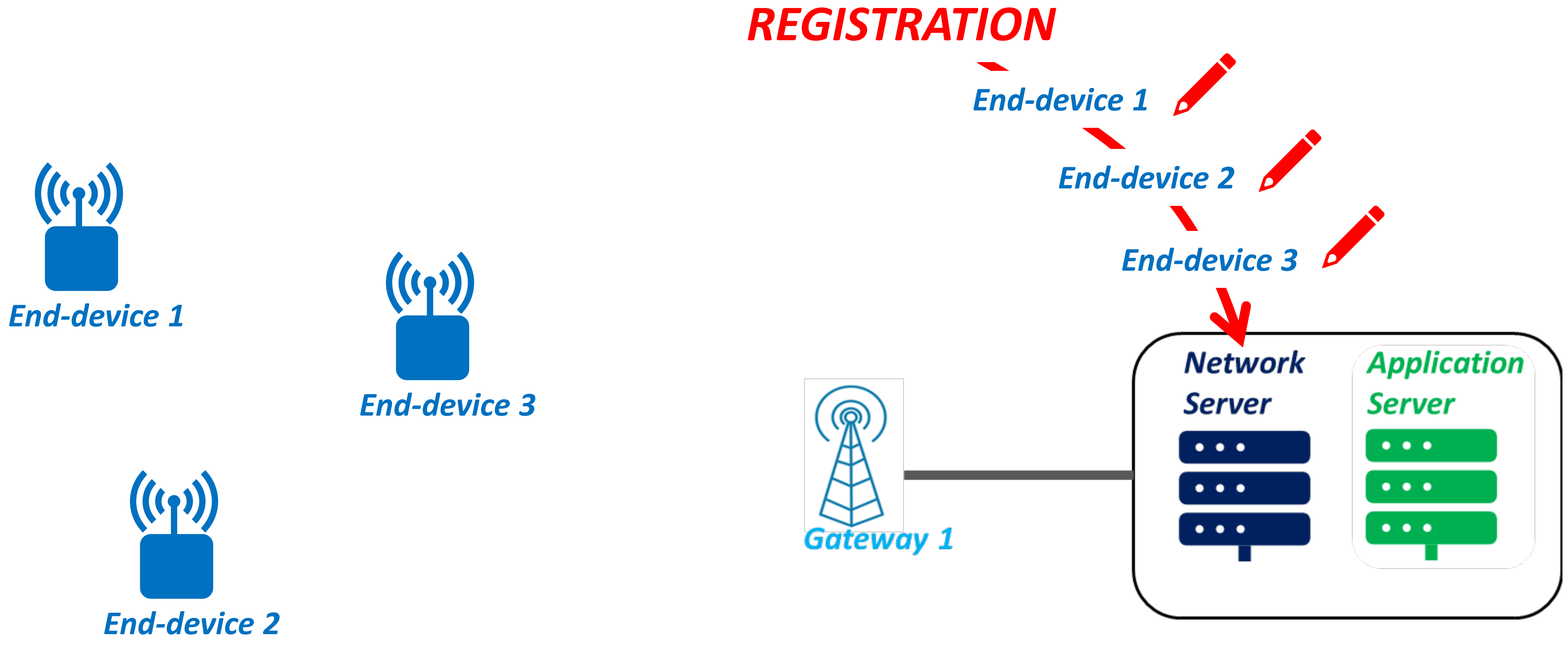
Step 3: Device registration

Step 1: Gateway configuration

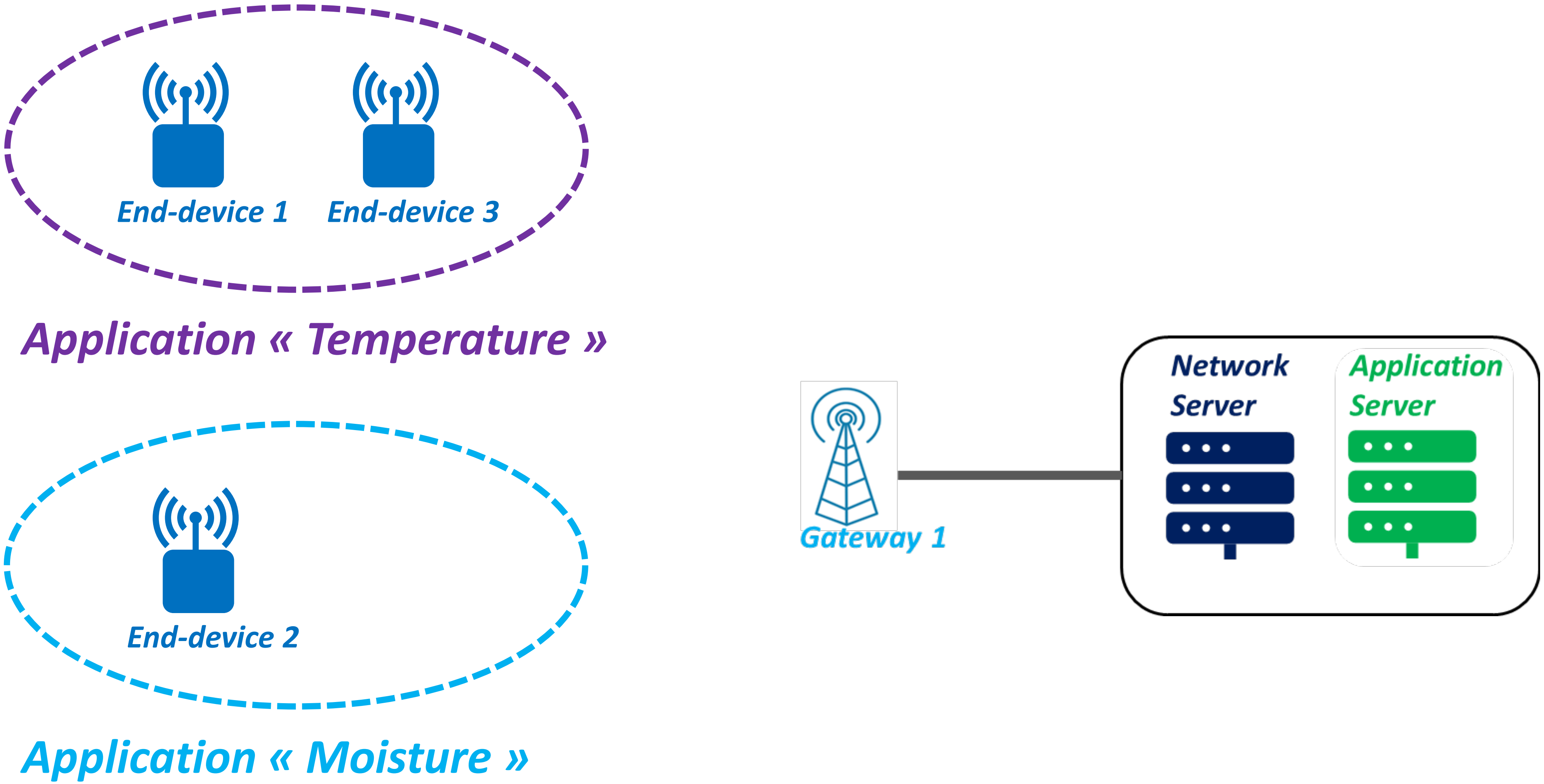
Semtech UDP Packet forwarder is used

Step 2: Gateway registration

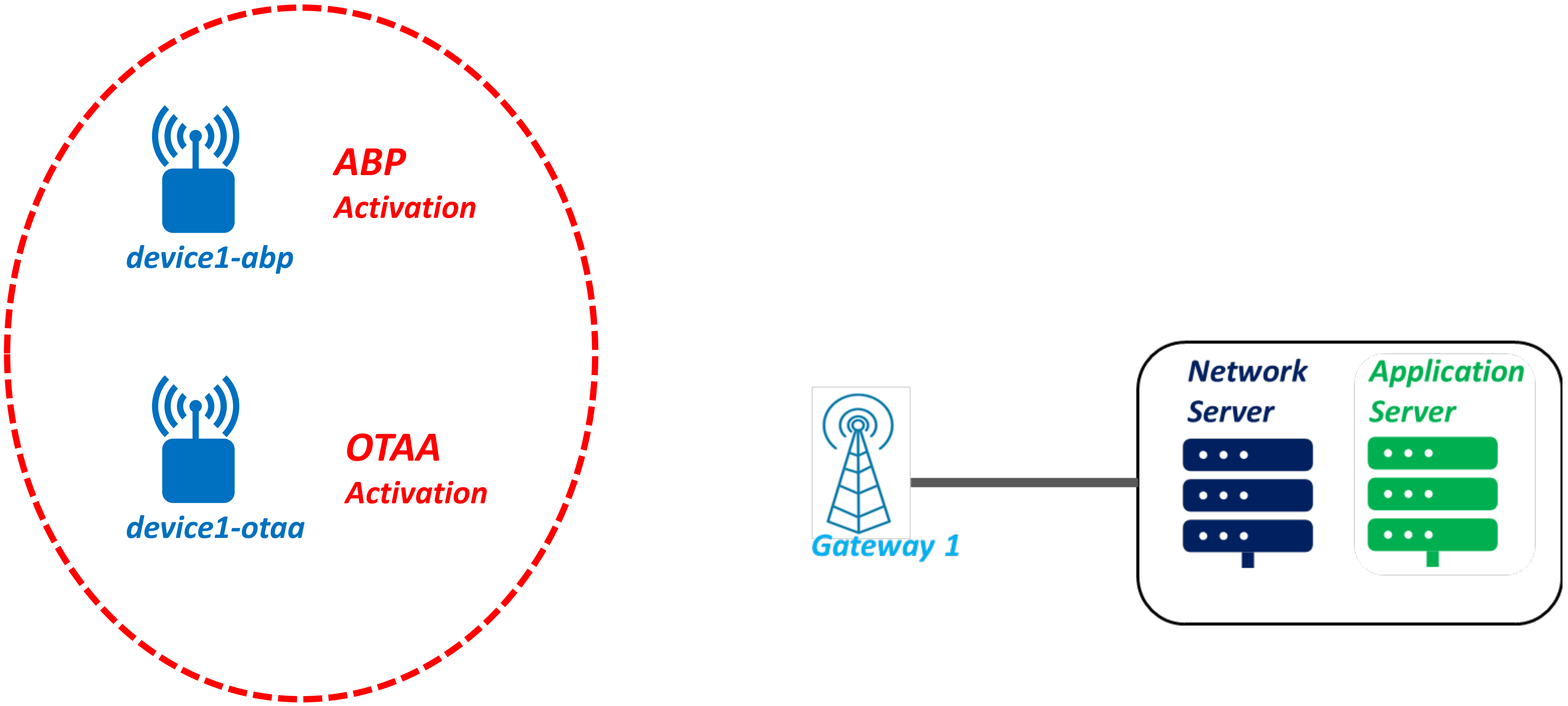
Step 3: Device registration



Step 3: Device registration



Step 3: Device registration



Application « training-usmb »

Step 4: Device configuration

Step 1: Gateway configuration

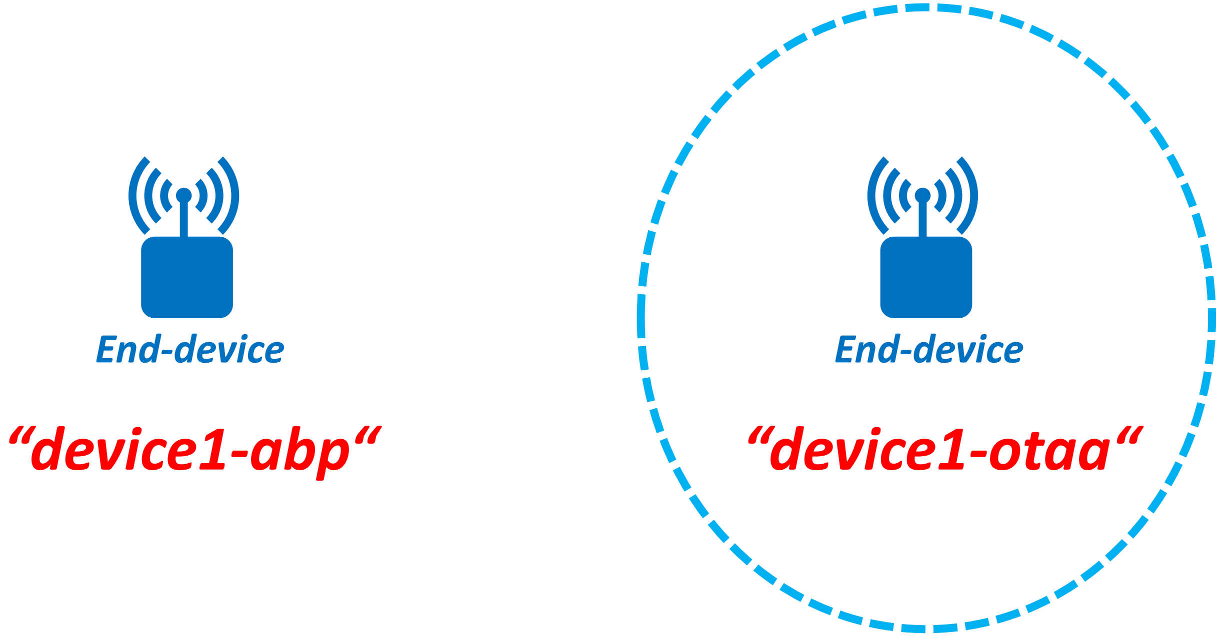
Semtech UDP Packet forwarder is used

Step 2: Gateway registration

Step 3: Device registration

Step 4: Device configuration

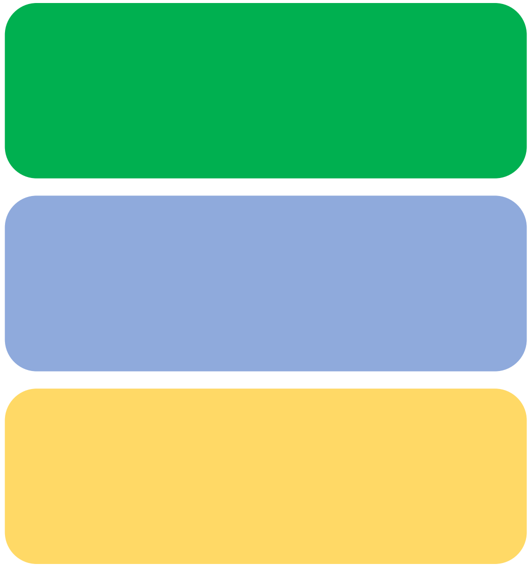
CONFIGURATION



The LoRaWAN frame

- ✓ The Physical (PHY), MAC and Application layer
- ✓ The LoRa modulation, the PHY payload, the Frame payload

LoRaWAN OSI Model



LoRaWAN OSI Model

PHY LAYER
(Physical layer)

Modulation → Chirp Spread Spectrum

↳ LoRa modulation

To add security to the LoRa modulation:

Confidentiality **Authenticity** **Integrity**

Identity **Acknowledgment** **…**

LoRaWAN OSI Model

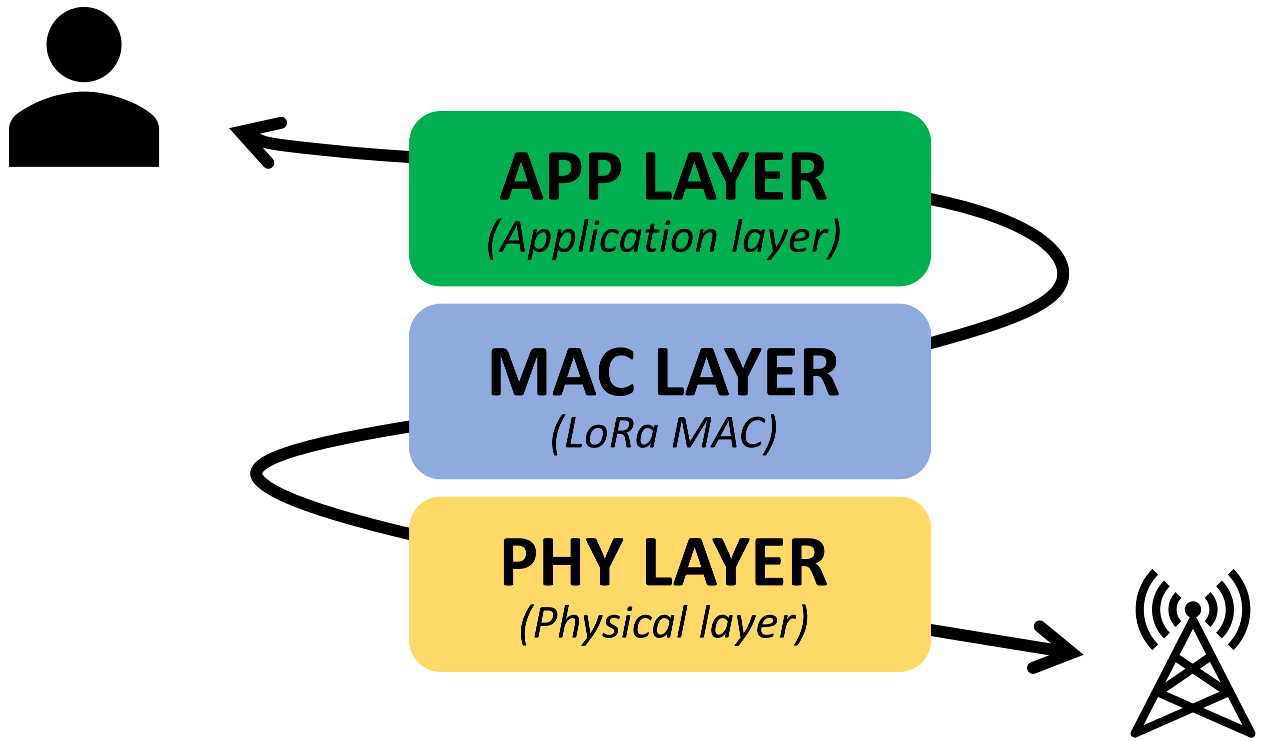
APP LAYER
(Application layer)

MAC LAYER
(LoRa MAC)

PHY LAYER
(Physical layer)

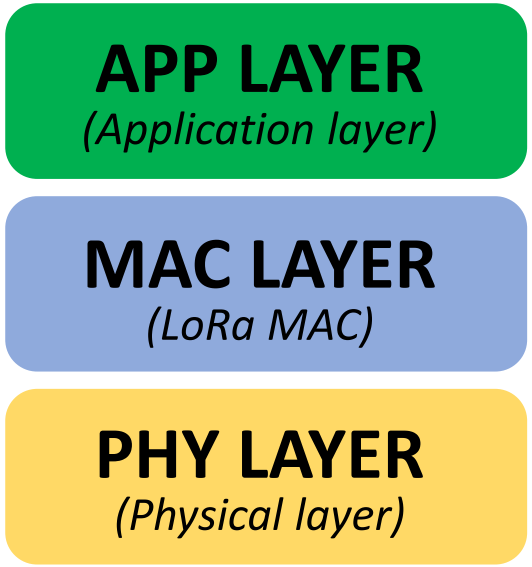
The LoRaWAN® protocol

LoRaWAN OSI Model



LoRaWAN OSI Model

<u>*The LoRa frame over the air:*</u>



LoRaWAN OSI Model

The LoRa frame over the air:

Preamble + Header

CRC

APP LAYER
(Application layer)

MAC LAYER
(LoRa MAC)

PHY LAYER
(Physical layer)

LoRaWAN OSI Model

The LoRa frame over the air:

Preamble + Header	MAC Layer	MAC Layer	CRC

APP LAYER
(Application layer)

MAC LAYER
(LoRa MAC)

PHY LAYER
(Physical layer)

LoRaWAN OSI Model

The LoRa frame over the air:

Preamble + Header	*MAC Layer*	*User data*	*MAC Layer*	*CRC*

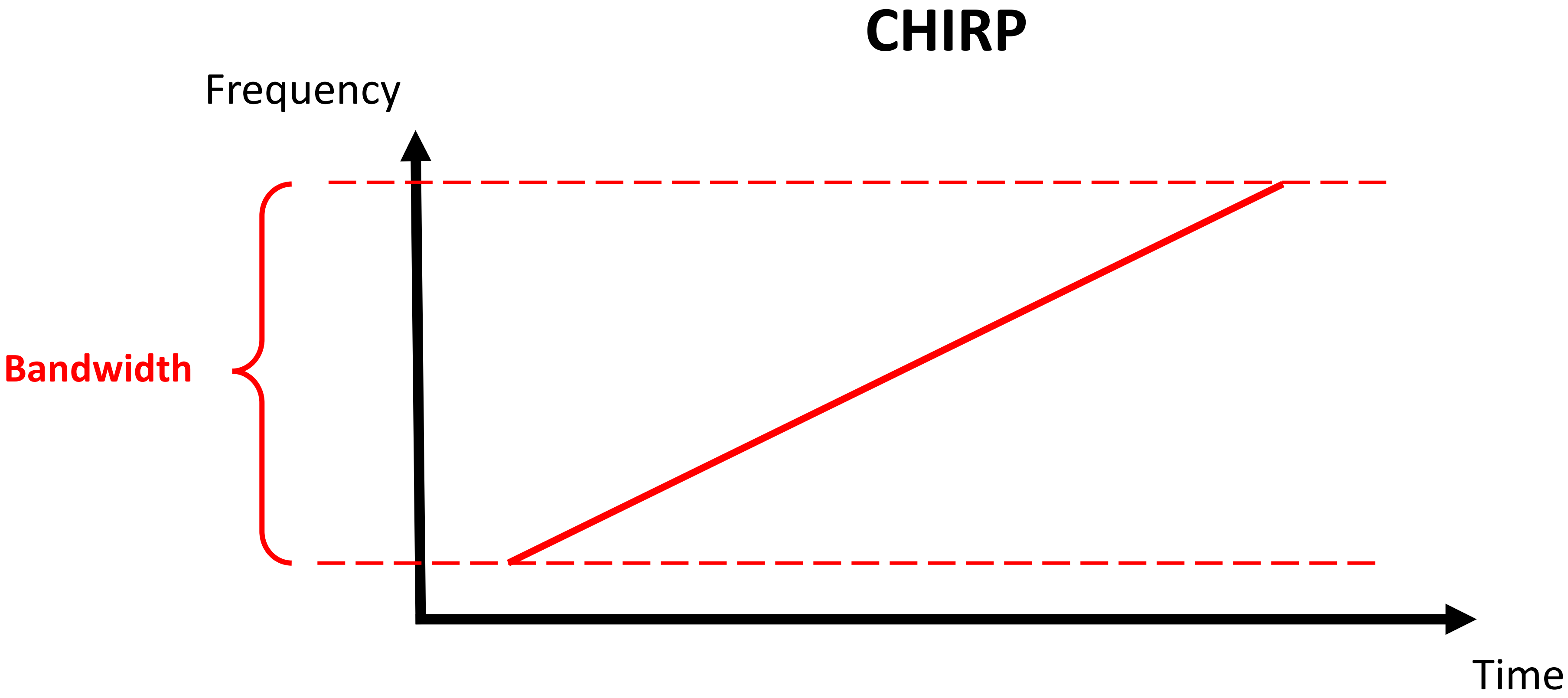
APP LAYER
(Application layer)

MAC LAYER
(LoRa MAC)

PHY LAYER
(Physical layer)

Physical layer

<u>As a reminder:</u>



Physical layer

<u>The LoRa Frame:</u>

Preamble + Optional Header	***PHY** Payload*	*CRC*

Preamble, header, CRC: can be configured.

<u>1st demonstration:</u>

Our message:	**“HELLO”**
Spreading Factor:	**12**
Bandwidth:	**125 kHz**
Frequency channel:	**868,1 MHz**
Preamble:	**8 symbols**
Header:	**YES**
CRC:	**YES**

4,25 symbols in addition to the preamble

→ 8 + 4,25 = **12,25 symbols**

→ 4,25 symbols = **2 Upchirps + 2,25 Downchirps**

Physical layer

<u>The LoRa Frame:</u>

Preamble + Optional Header	***PHY** Payload*	*CRC*

Preamble, header, CRC: can be configured.

<u>2nd demonstration:</u>

Our message:	**"HELLO"**
Spreading Factor:	**12**
Bandwidth:	**125 kHz**
Frequency channel:	**868,1 MHz**
Preamble:	**8 symbols**
Header:	**YES**
CRC:	**YES**

LoRa MAC Layer

The LoRa Frame:

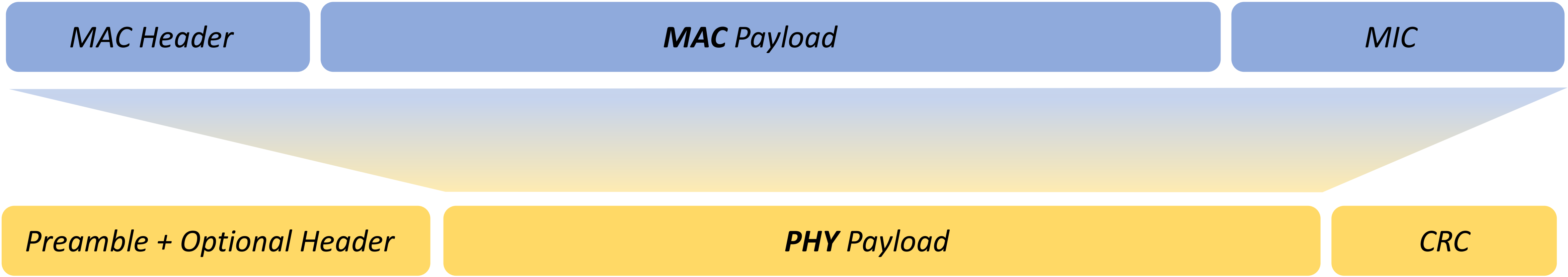
Preamble + Optional Header	***PHY** Payload*	*CRC*

Carries the upper layer protocol

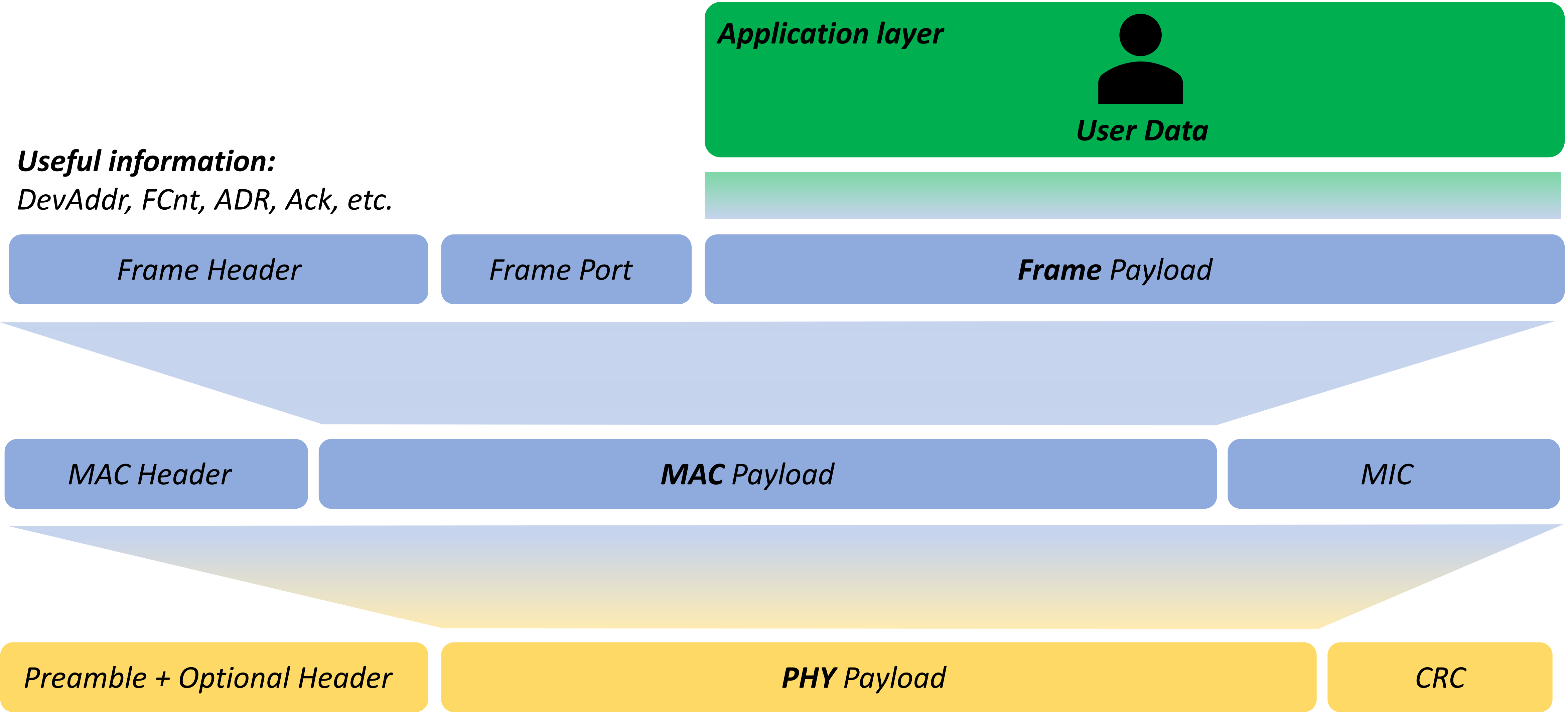
→ ***LoRa MAC (LoRaWAN)***

LoRa MAC Layer

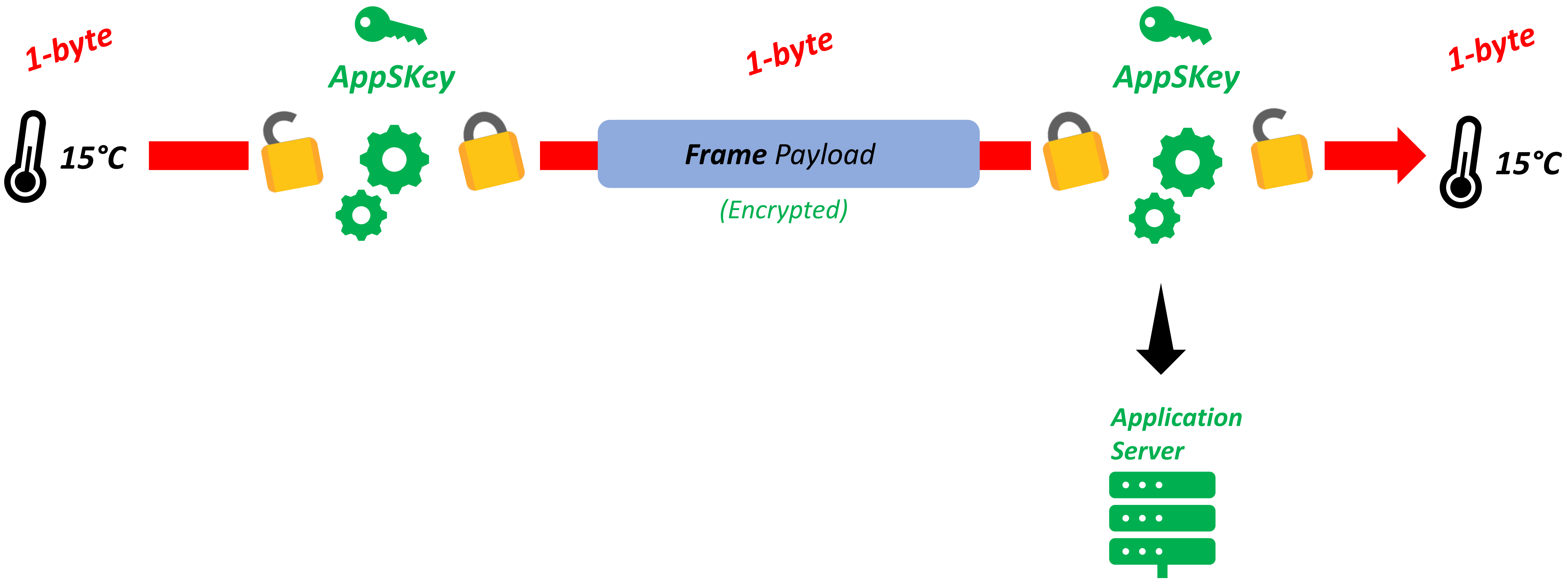
Type of frame:
Join, Data up, etc.



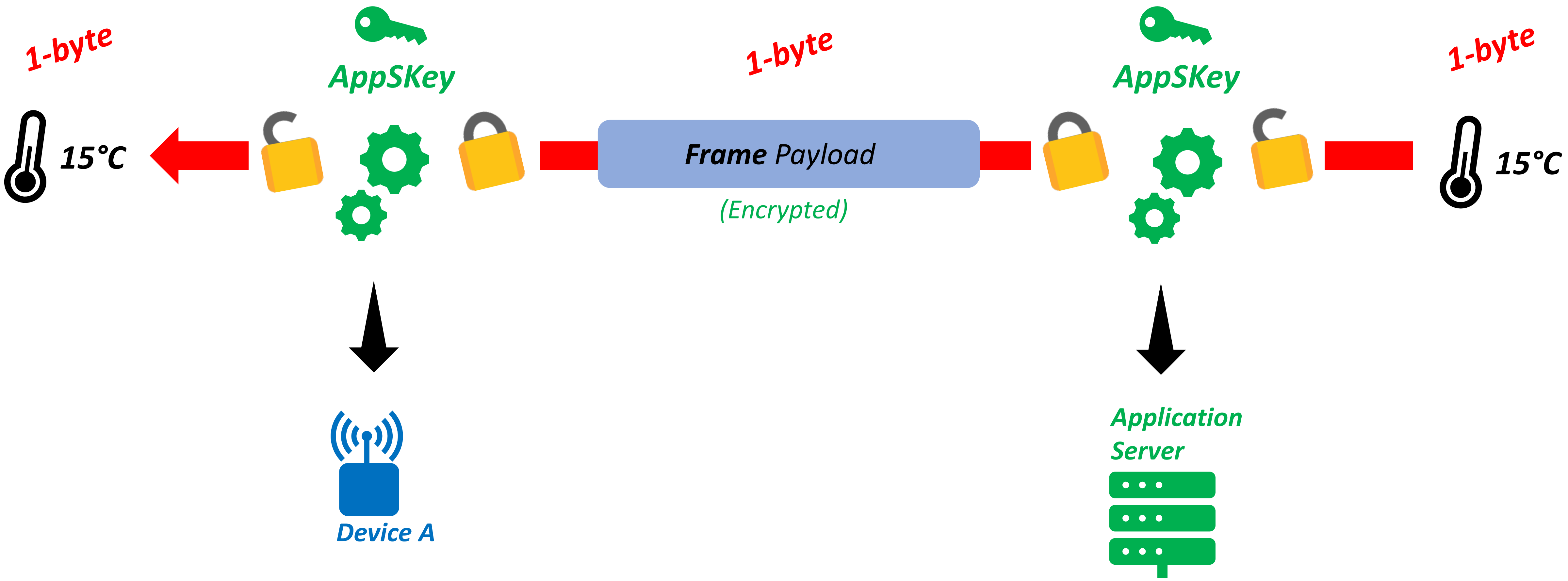
LoRa MAC Layer



Application



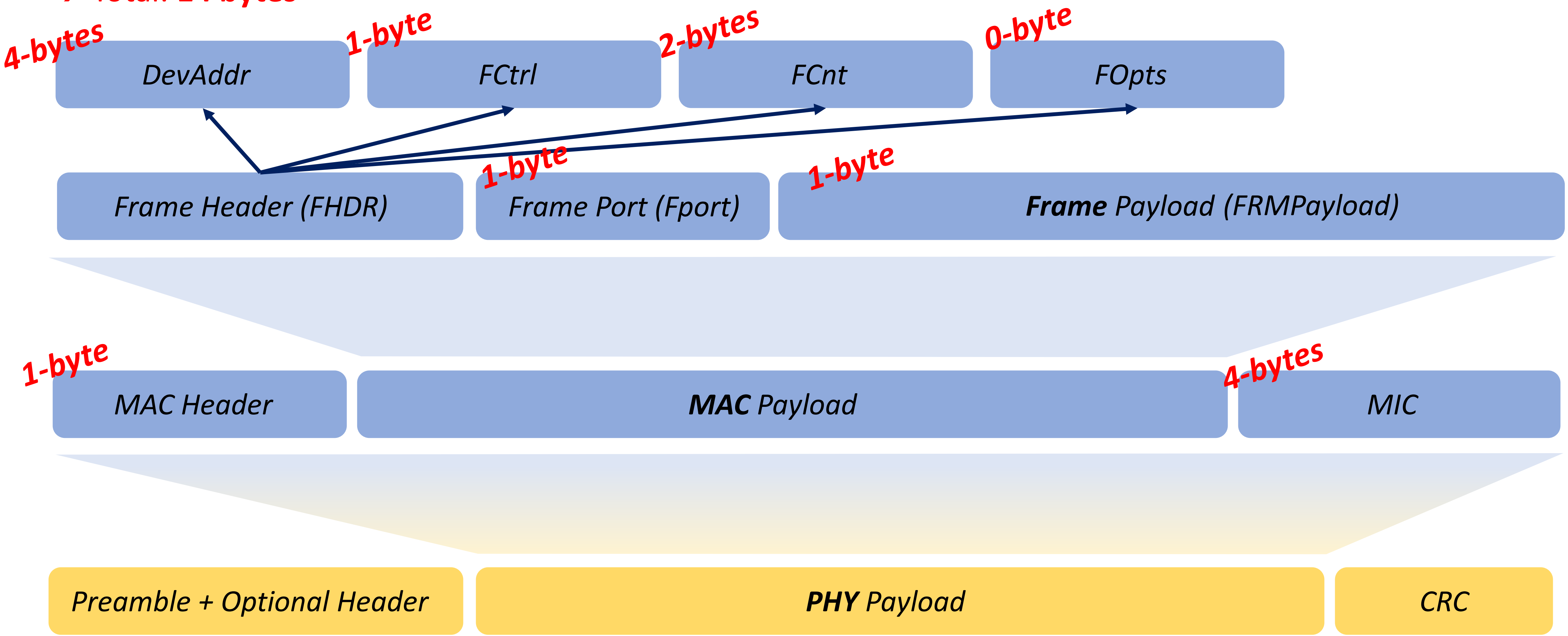
Application



Application

MAC Header (1-byte) | **DevAddr** (4-bytes) | **FCtrl** (1-byte) | **FCnt** (2-bytes) | **FPort** (1-byte) |**FRMPayload** (1-byte) | **MIC** (4-bytes)

→ Total: **14 bytes**



IP Frame and JSON Object

Network Configuration, Step 1

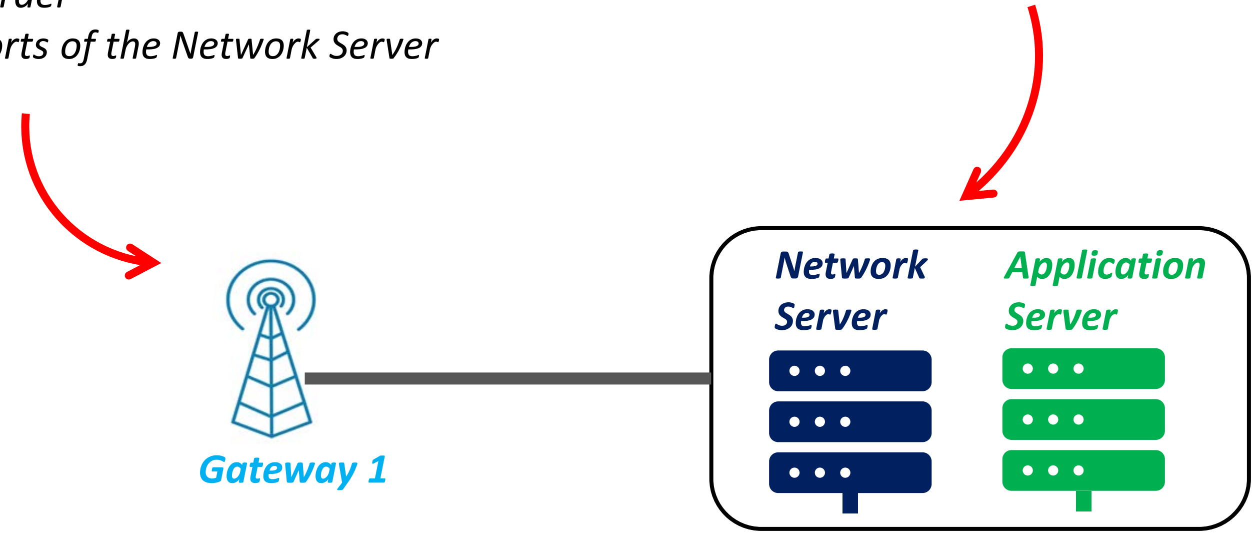
→ Gateway configuration

Packet Forwarder

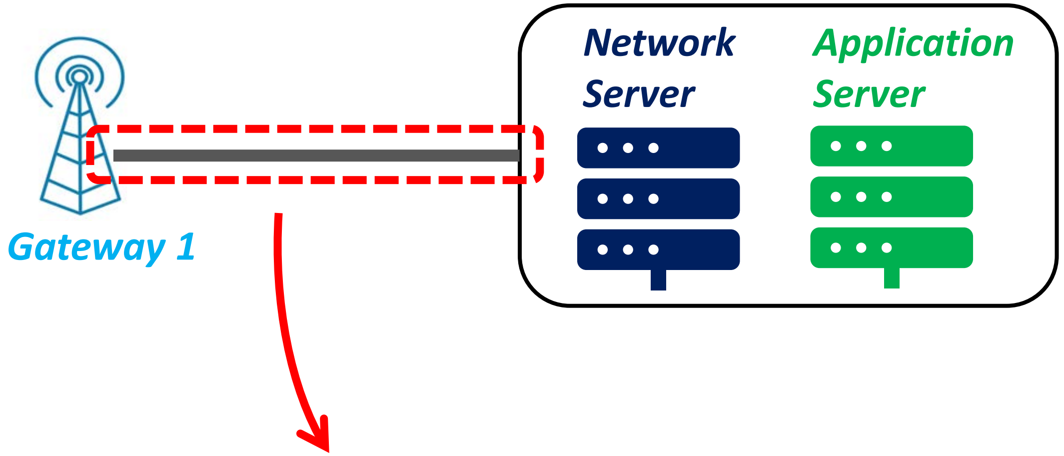
IP address/Ports of the Network Server

Network Configuration, Step 2

→ Gateway registration



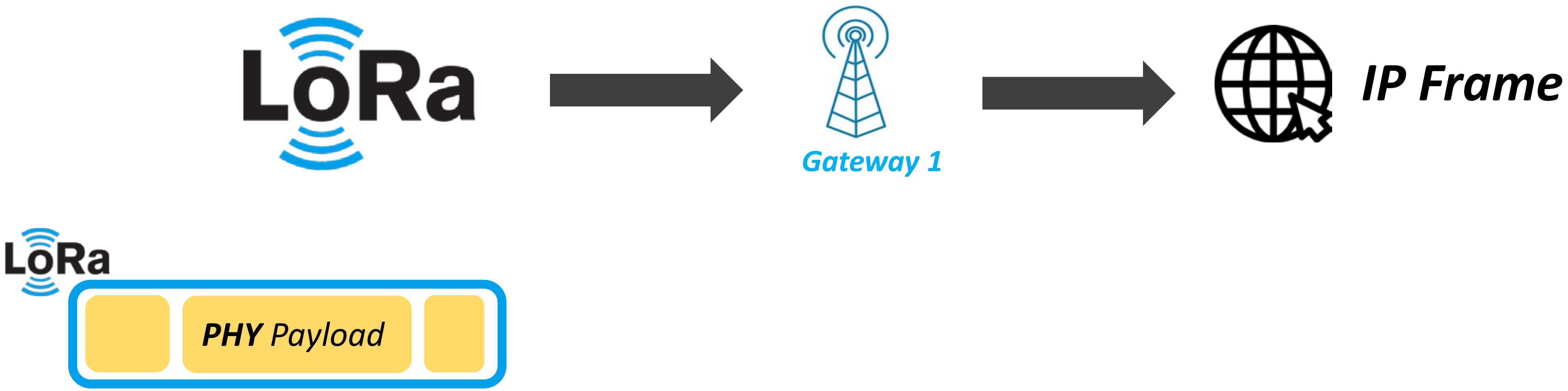
IP Frame and JSON Object



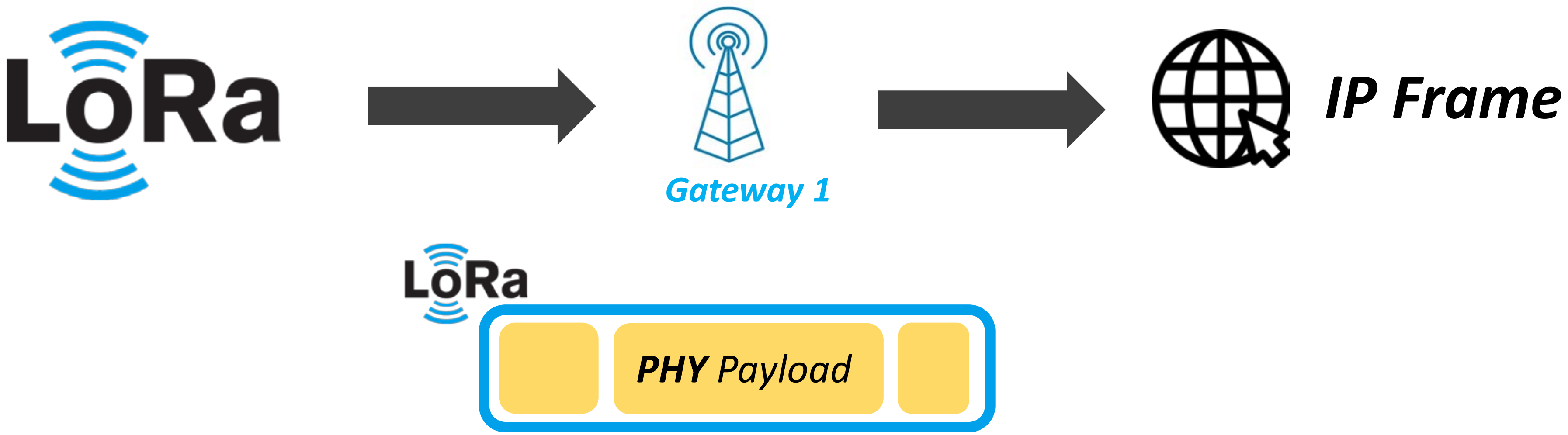
Backhaul connectivity

→ ***Ethernet***
→ ***Wi-Fi***
→ ***4G***
→ ***LTE-M***
→ ***…***

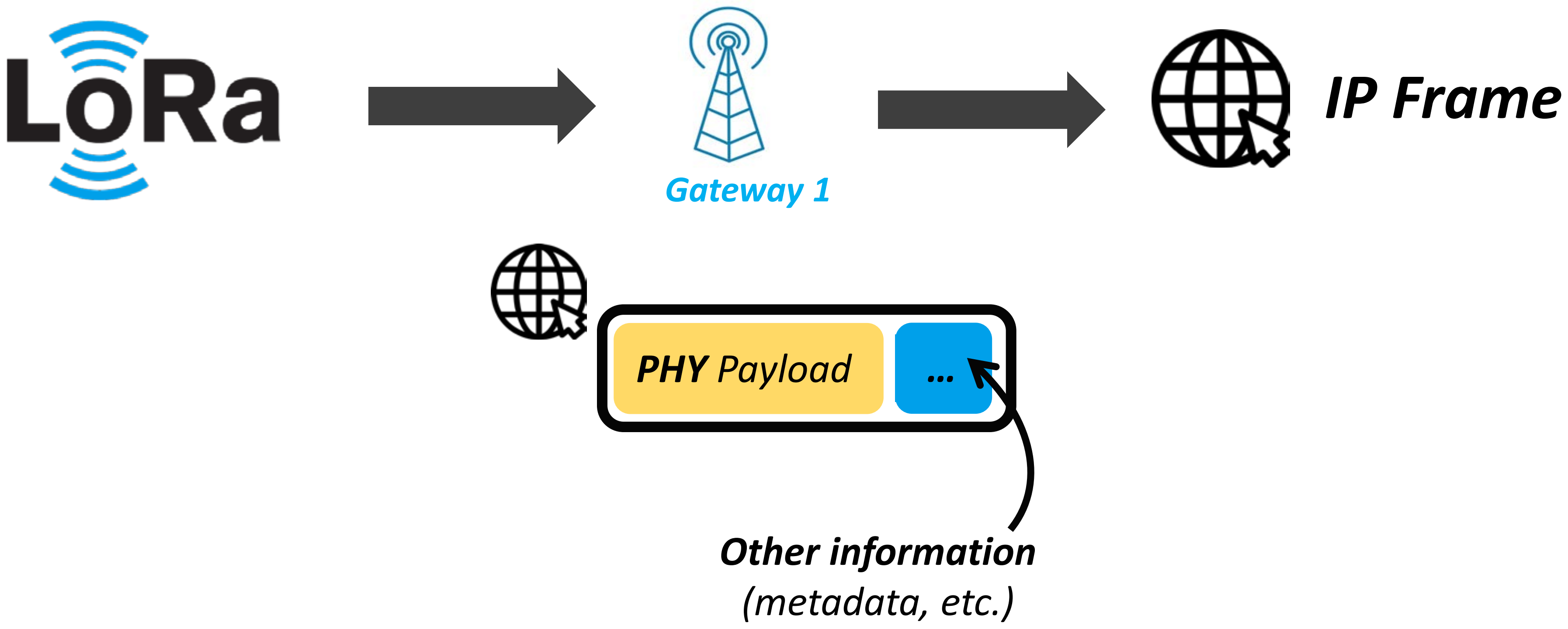
IP Frame and JSON Object



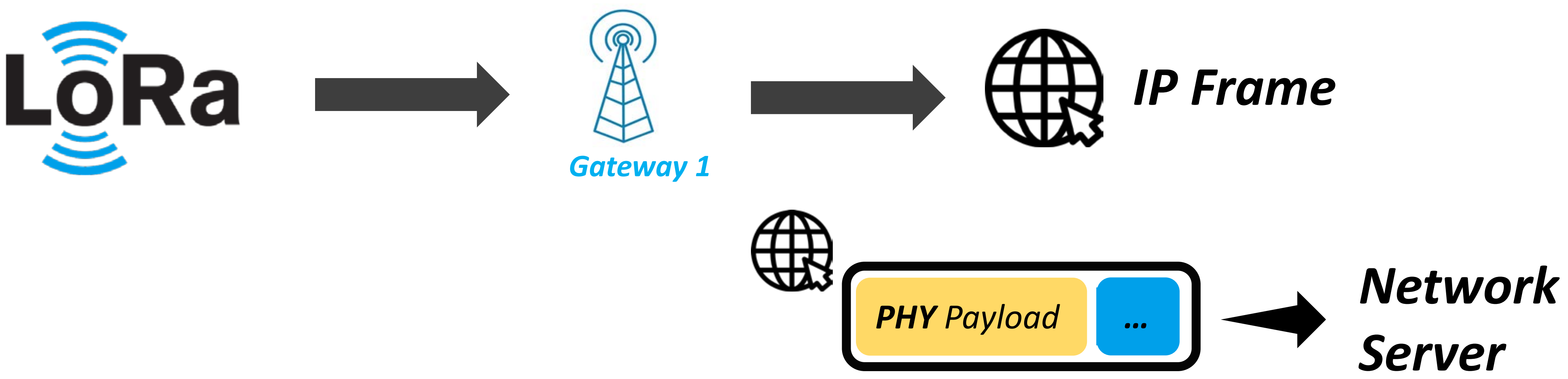
IP Frame and JSON Object



IP Frame and JSON Object



IP Frame and JSON Object



IP Frame and JSON Object

***JSON object** (text format)*

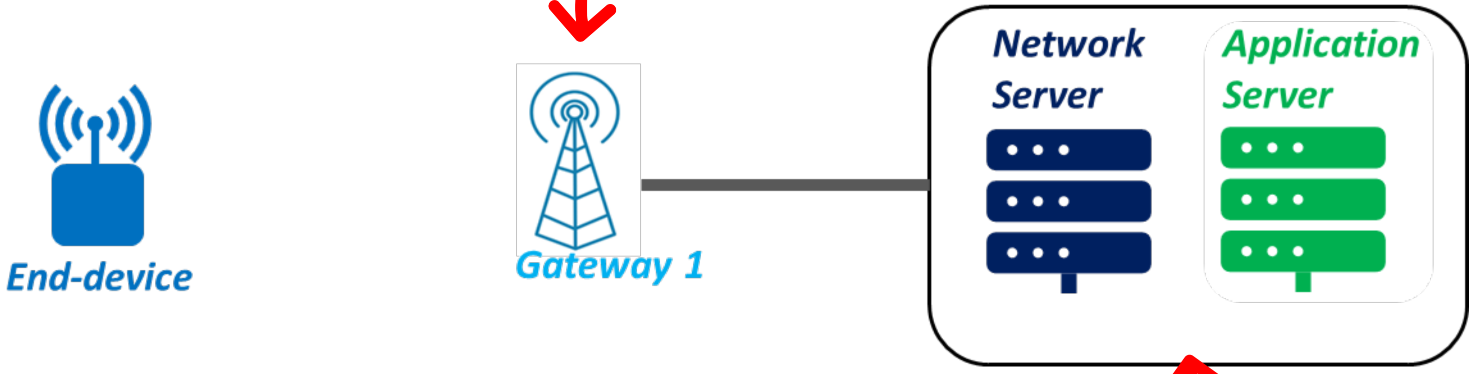
```
{
        "PHY Payload": "QNAPvA6Ax3ECqHk4C8hfzuA="
        "Radio":{
                        "RSSI": -80,
                        "SNR": 7,
                        "SF": 10,
                        "Channel": 867100000,
                        …
               }
        …
}
```

Will be used by
- the USER
- the LoRaWAN stack

Decoding a LoRaWAN frame

<u>LoRaWAN frame demonstration:</u>

Coverage for our demonstration



The ABP device is registered

Activation mode: ABP

DevAddr: 2601168E ✓

NwkSKey: FBEC5289D32738C718402B7984856702 ✓

AppSKey: 63A9BD8DC745F58CE0FAECC4BB88A5BE ✓

Decoding a LoRaWAN frame

The JSON text message transmistted by the Gateway:

Radio transmission information

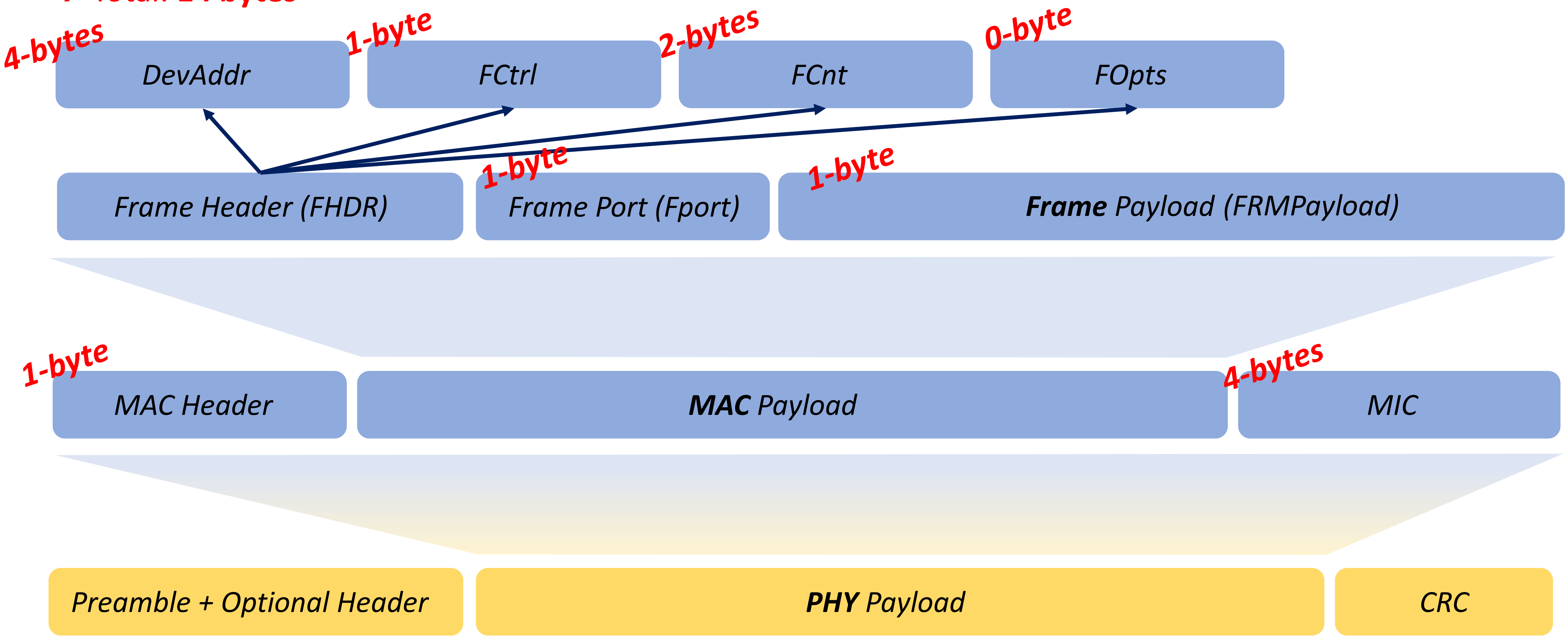
PHY Payload
(or RAW payload)

{ **"gw_id":** "eui-0080e1ffff01510e",
"payload": "QI4WASYAAAABLbLr5Vw=",
"lora": {
"spreading_factor": 7,
"bandwidth": 125,
"air_time": 46336000 },
"coding_rate": "4/5",
"timestamp": "2020-06-16T08:39:01.340Z",
"rssi": -76, **"snr":** 8.8,
"dev_addr": "2601168E",
"frequency": 868500000}

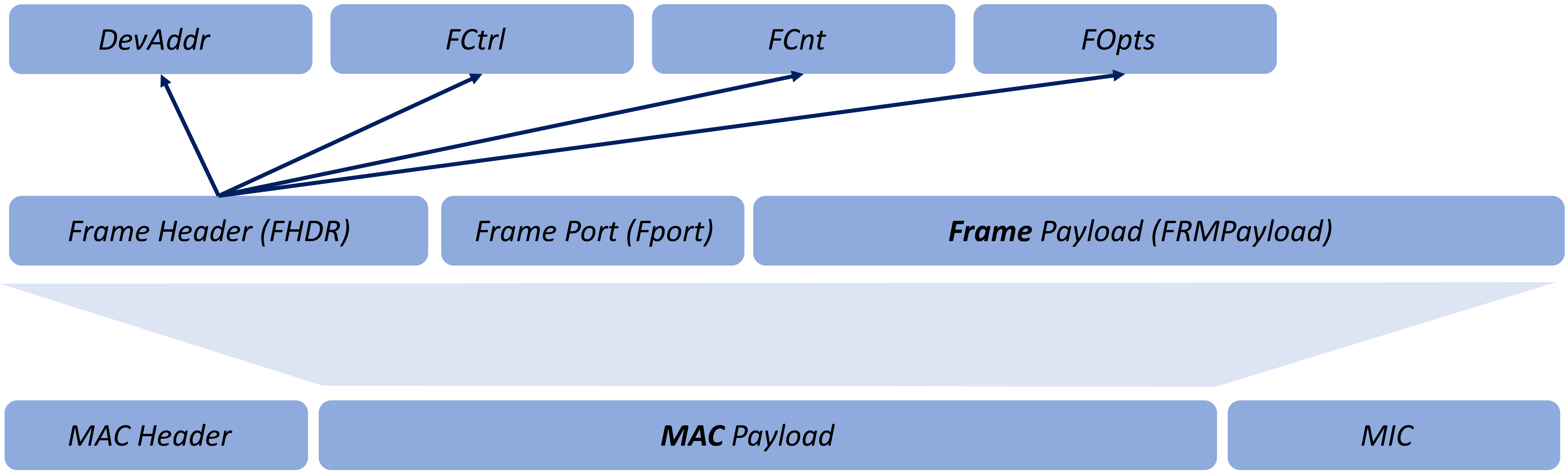
Application

MAC Header (1-byte) | **DevAddr** (4-bytes) | **FCtrl** (1-byte) | **FCnt** (2-bytes) | **FPort** (1-byte) |**FRMPayload** (1-byte) | **MIC** (4-bytes)

→ Total: **14 bytes**



Application



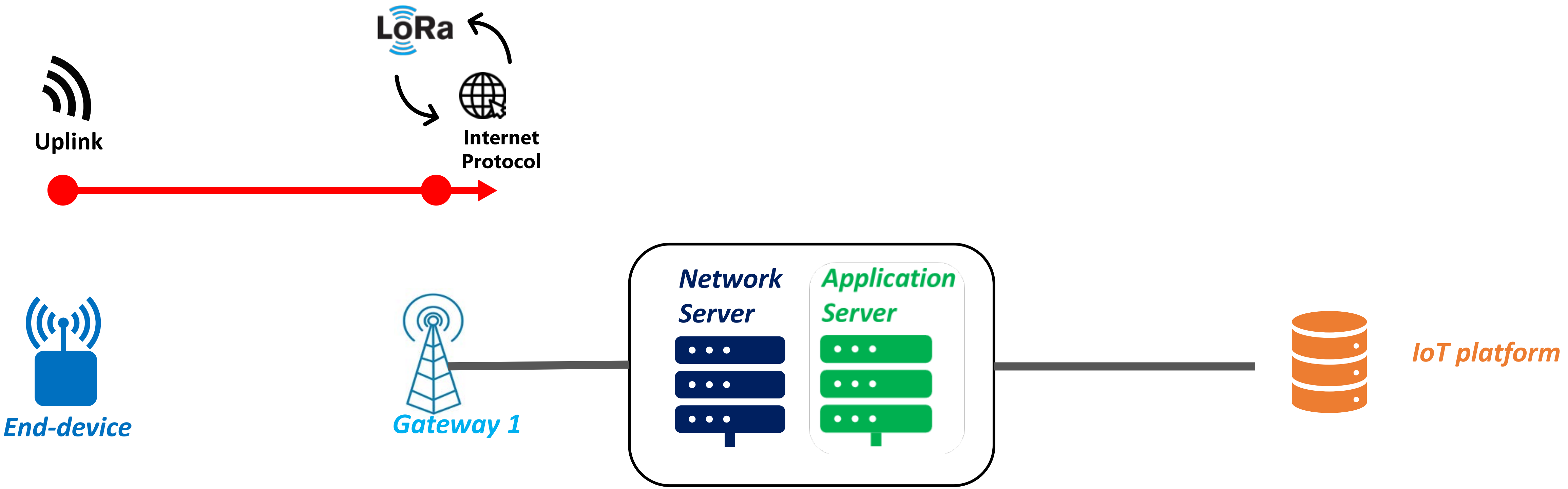
MAC Header (1-byte) | **DevAddr** (4-bytes) | **FCtrl** (1-byte) | **FCnt** (2-bytes) | **FPort** (1-byte) |**FRMPayload** (1-byte) | **MIC** (4-bytes)

→ **14 bytes**

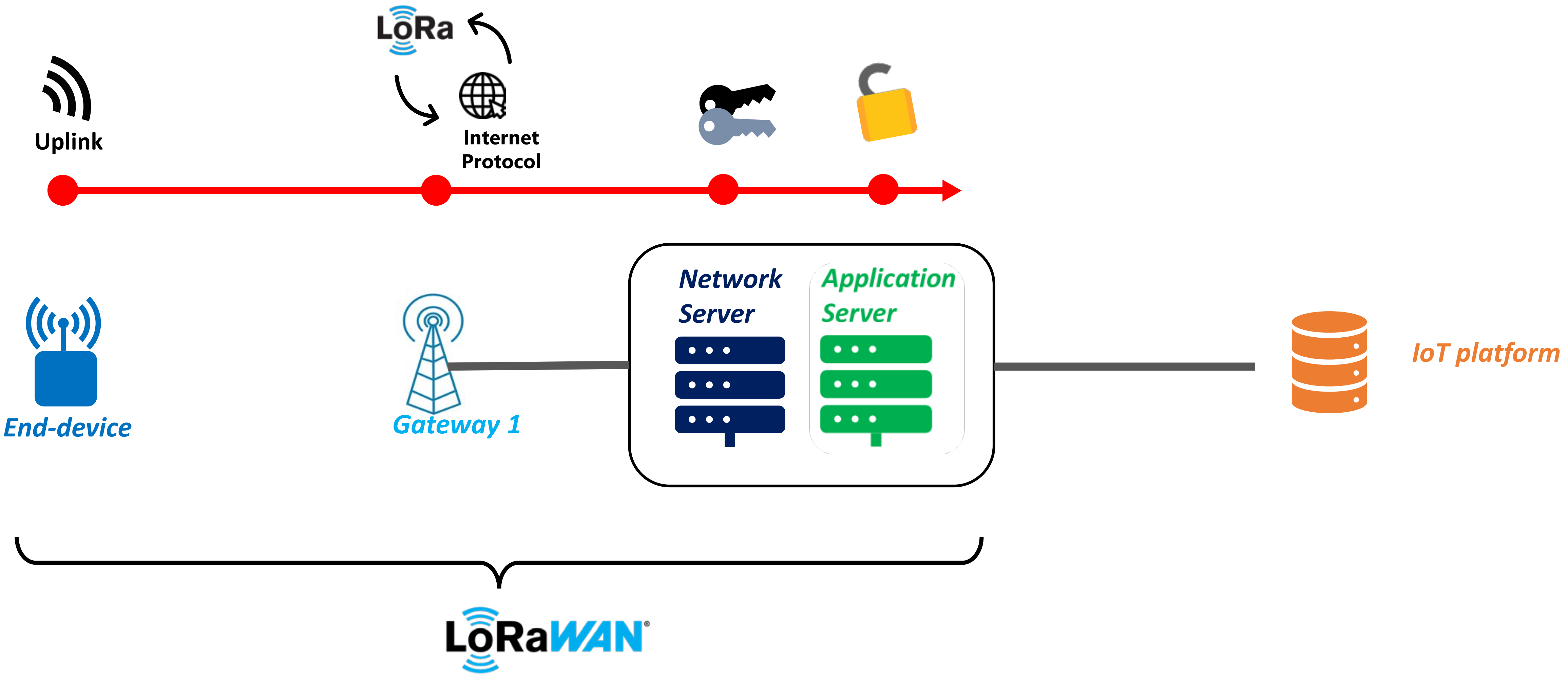
Exporting data from the LoRaWAN Server

- ✓ Retrieving the data and storing it in an IoT Platform
- ✓ Dialog between the LoRaWAN Server and the IoT Platform : HTTP & MQTT
- ✓ Demonstrations: uplink stream and downlink stream

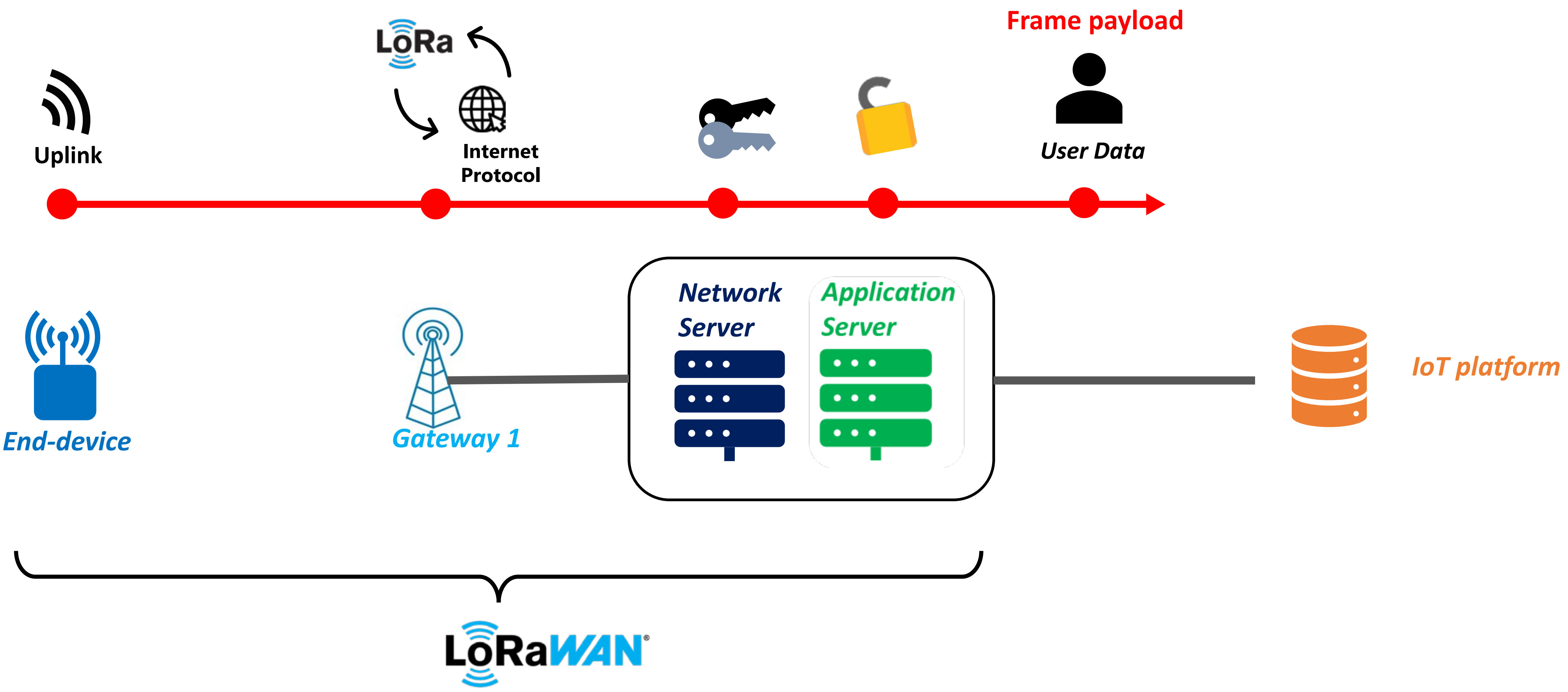
The IoT platform



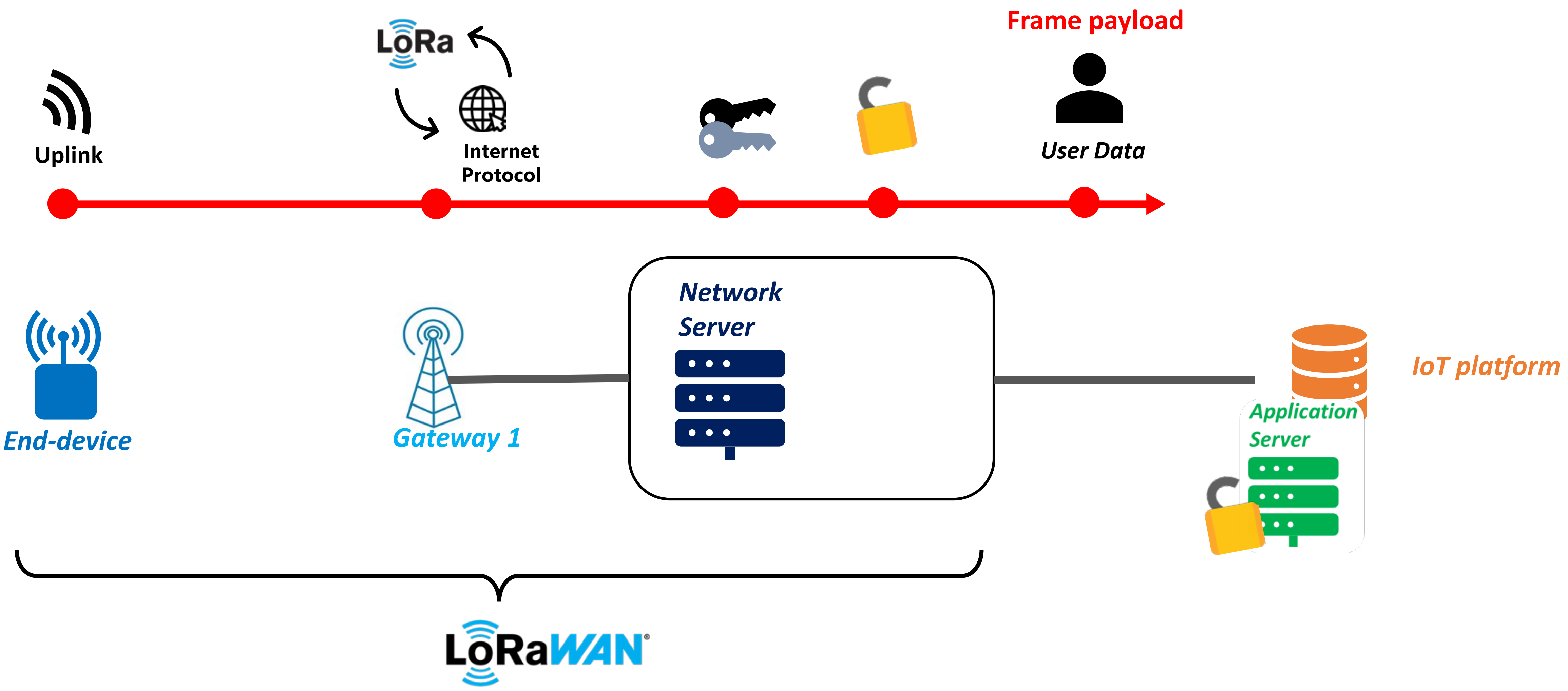
The IoT platform



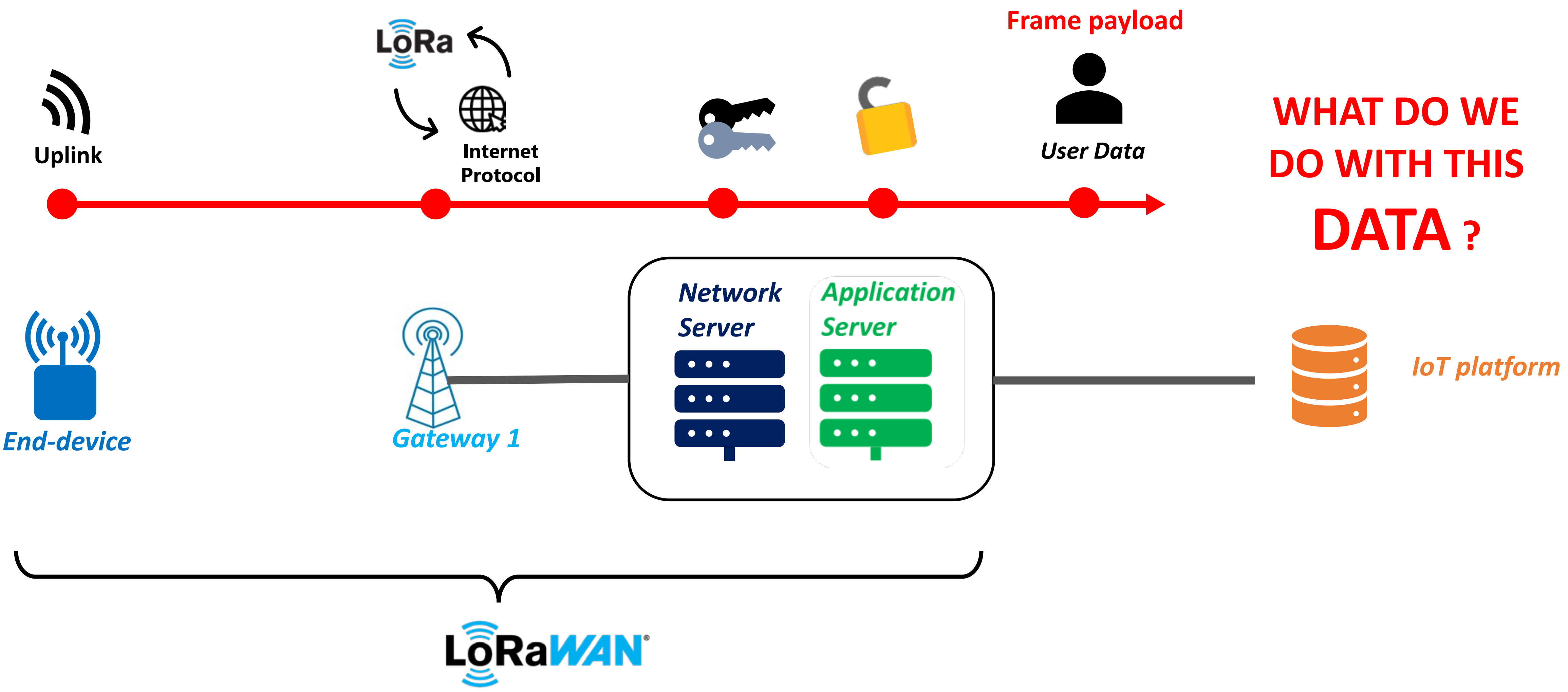
The IoT platform



The IoT platform

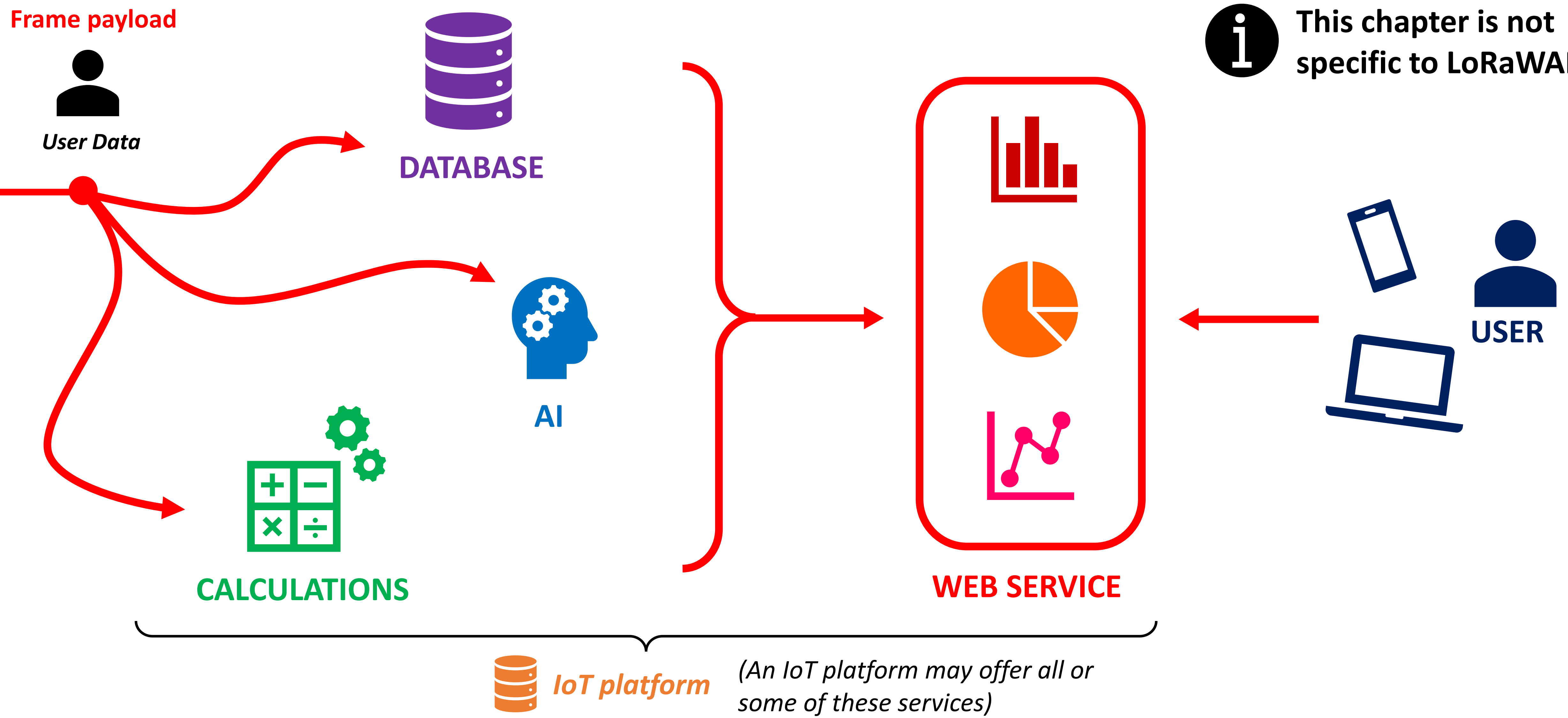


The IoT platform



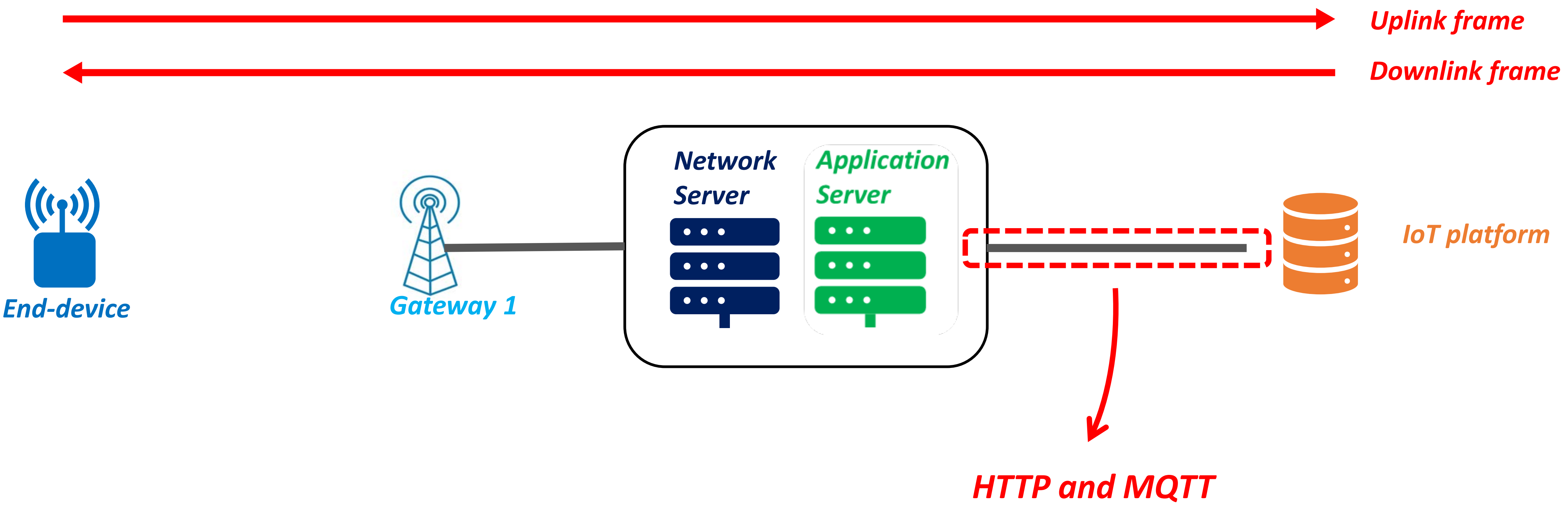
The IoT platform

This chapter is not specific to LoRaWAN

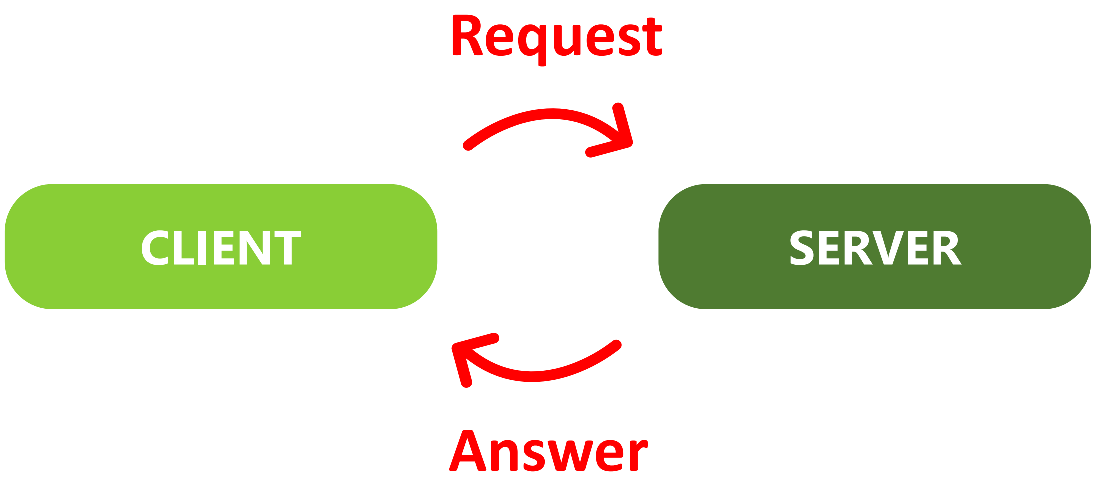


The IoT platform

Vocabulary



Client and Server principle



Examples:

Email
(POP/SMTP)

File
(FTP)

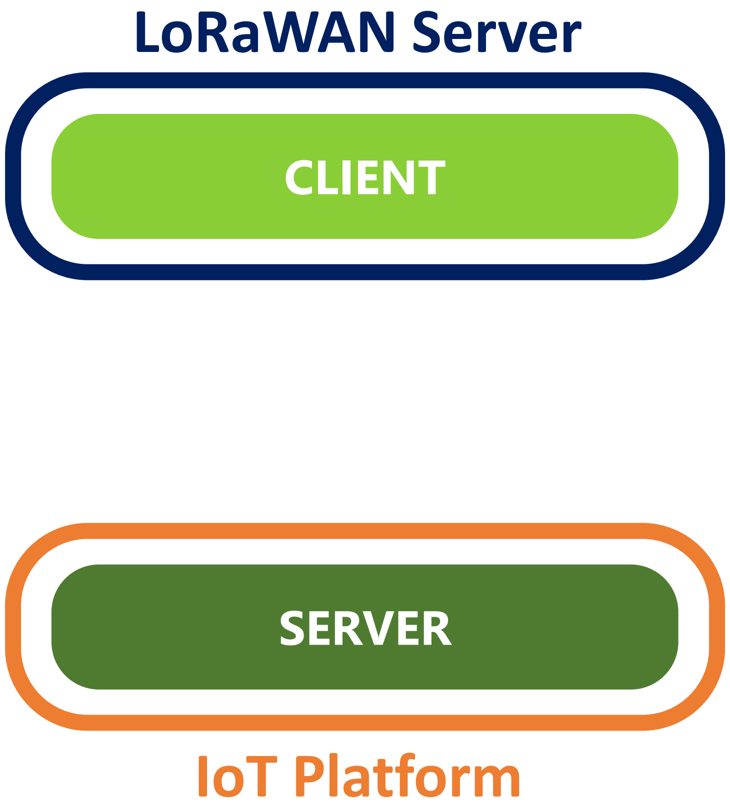
WEB
(HTTP)

Client and Server principle

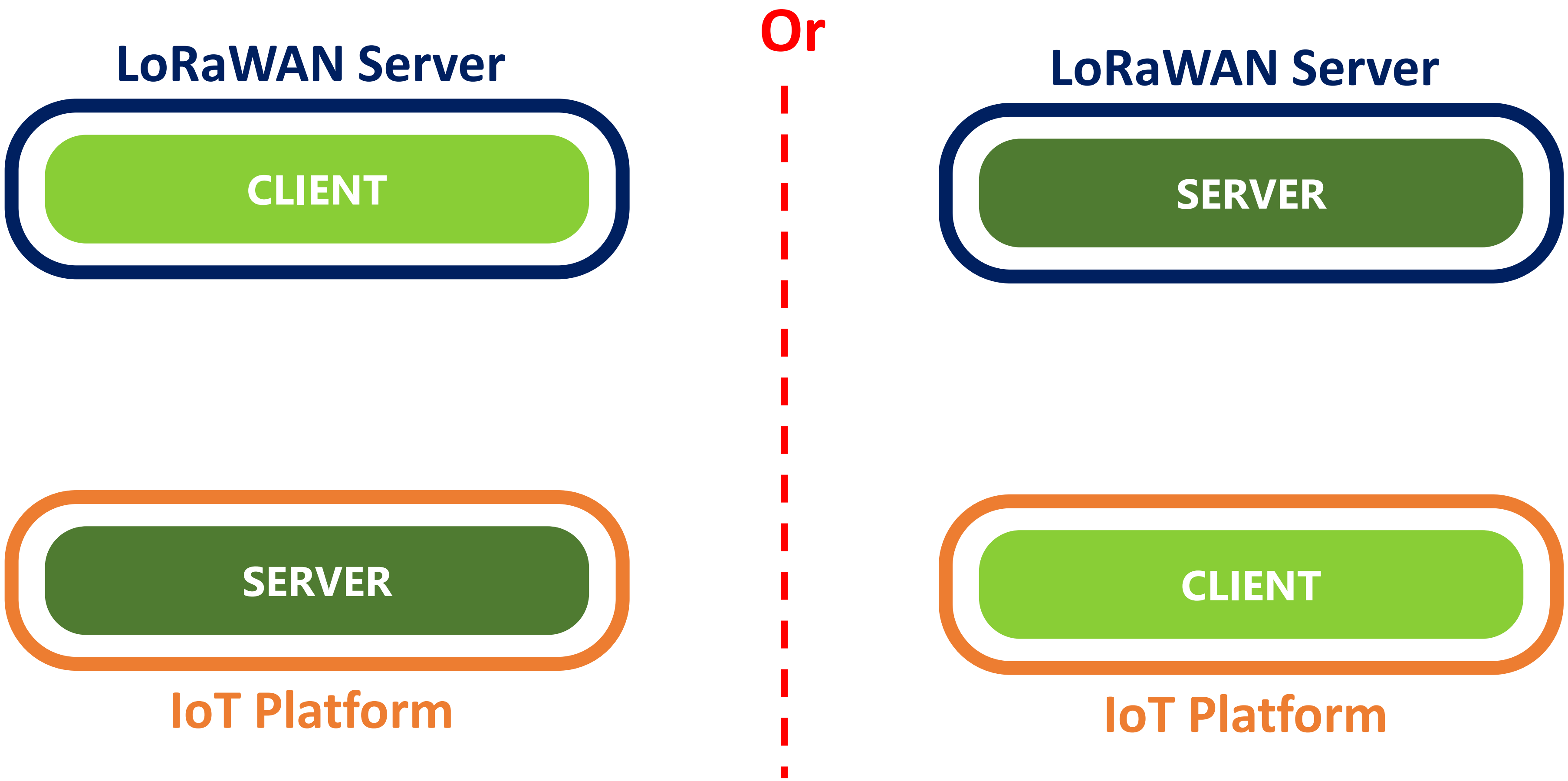
LoRaWAN Server

IoT Platform

Client and Server principle



Client and Server principle



Client and Server principle

LoRaWAN Server

THE THINGS NETWORK

LORIOT

Actility

ChirpStack

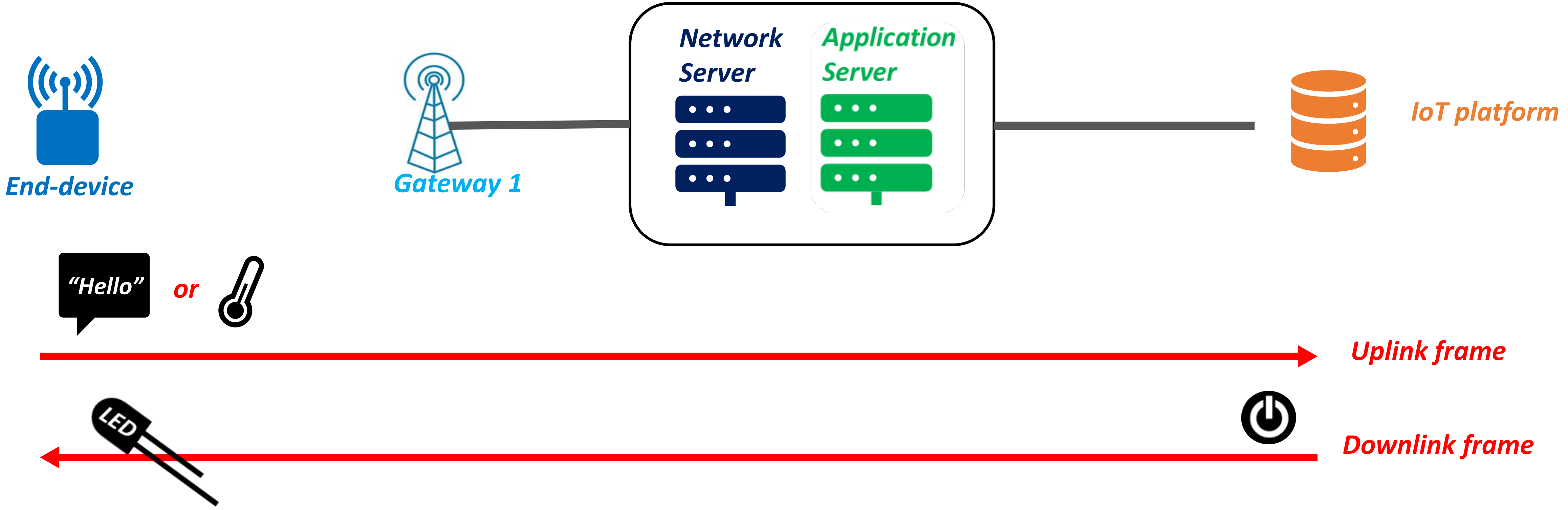
Our platform

Open-Source platform

Professional platform

IoT Platform

Client and Server principle



<u>Our situations:</u>

- ***Uplink*** *(hello string or temperature)*
- ***Downlink*** *(switch command)*

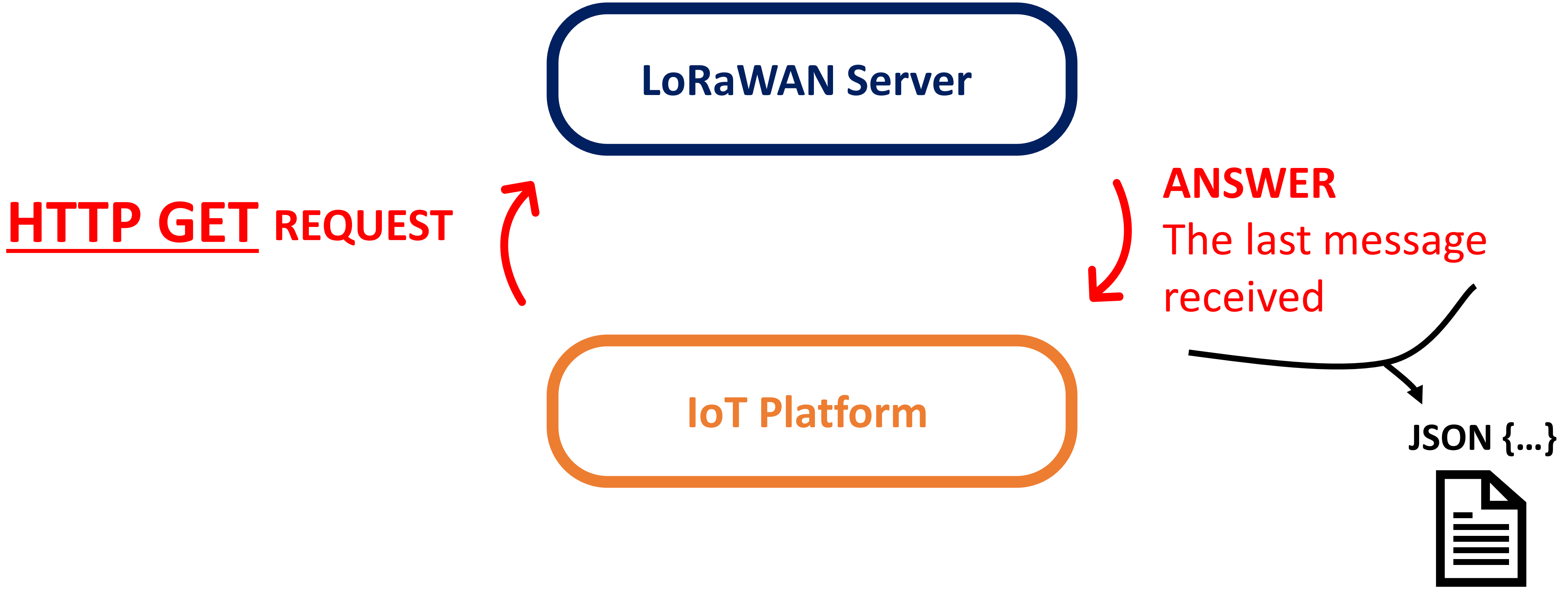
<u>HTTP Protocol methods to be studied:</u>

- ***HTTP GET***
- ***HTTP POST***

HTTP GET

1. Uplink Stream:

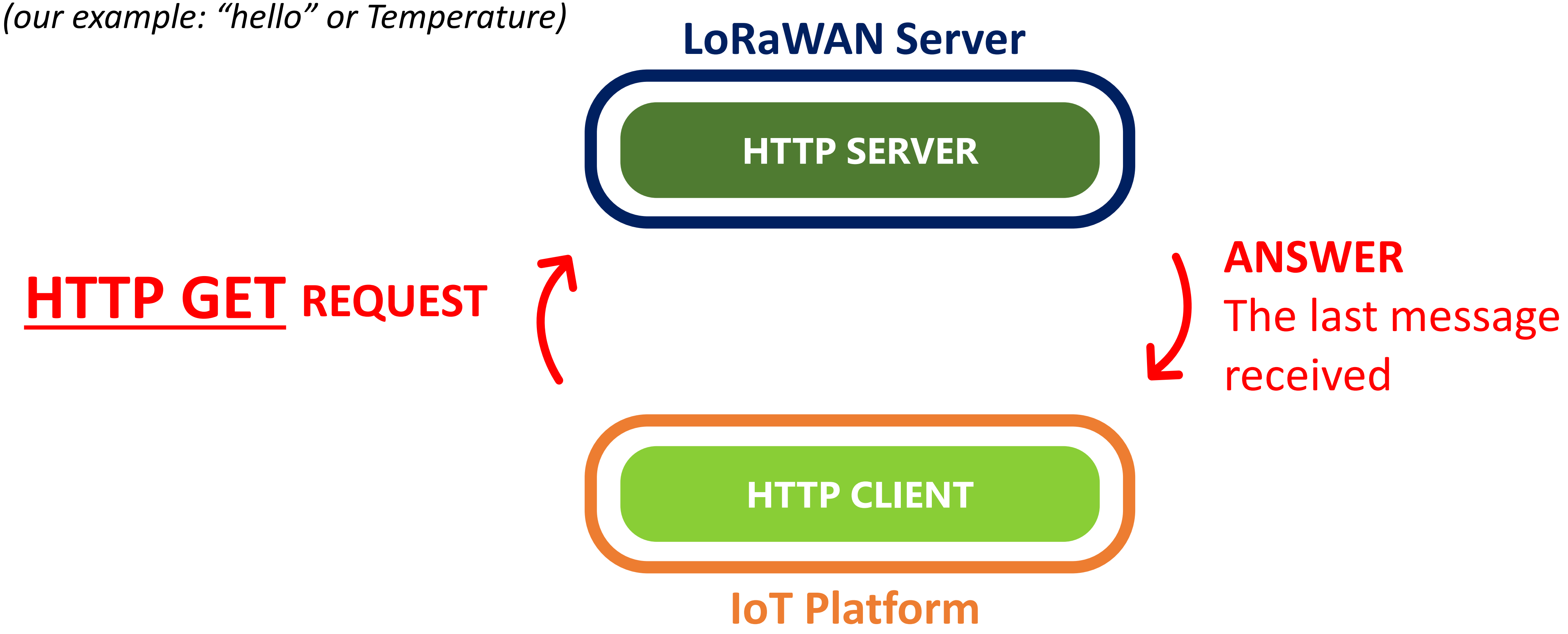
(our example: “hello” or Temperature)



HTTP GET

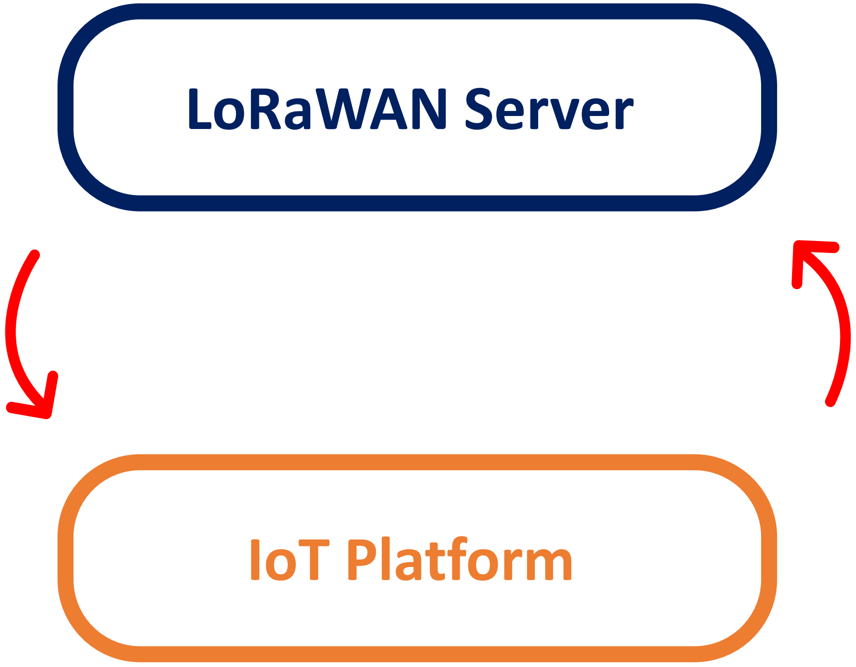
1. Uplink Stream:

(our example: “hello” or Temperature)



HTTP GET

2. Downlink Stream:
(our example: Turn ON/OFF a LED)



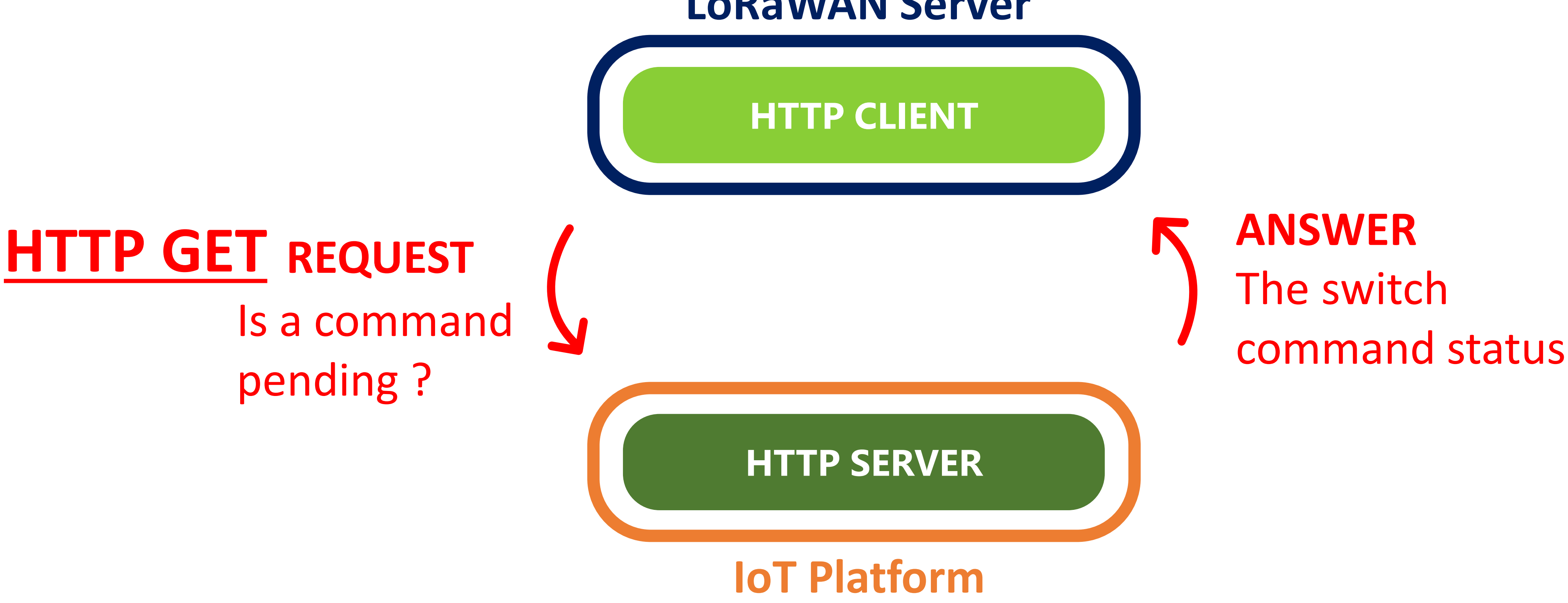
HTTP GET REQUEST
Is a command pending ?

ANSWER
The switch command status

HTTP GET

2. Downlink Stream:

(our example: Turn ON/OFF a LED)



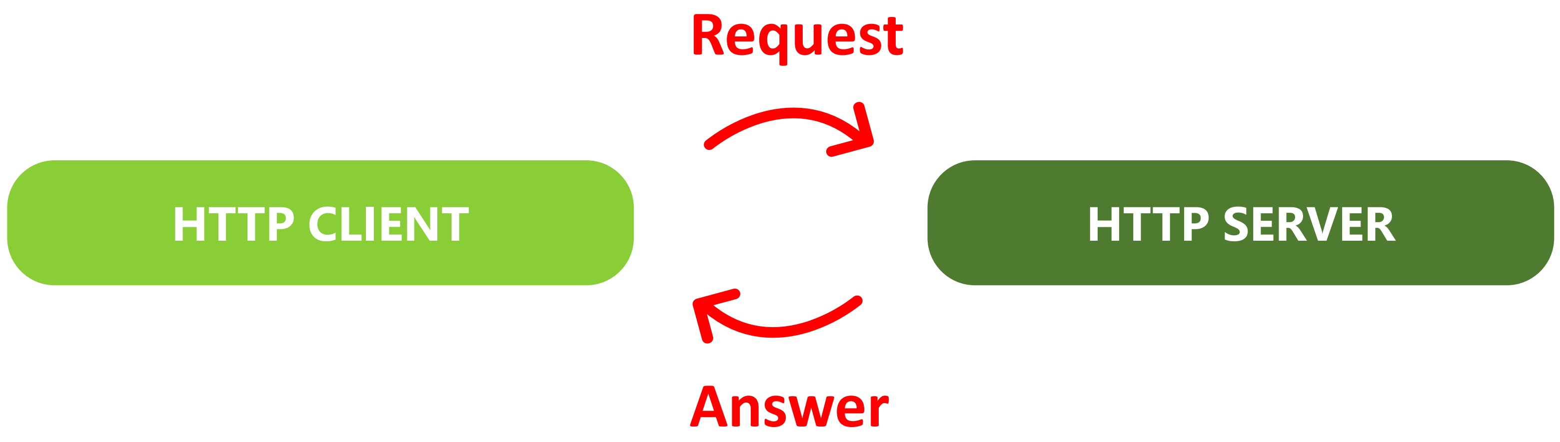
HTTP GET

GET format needs: the request URL

1 request ⇔ 1 URL

(to be found in the documentation)

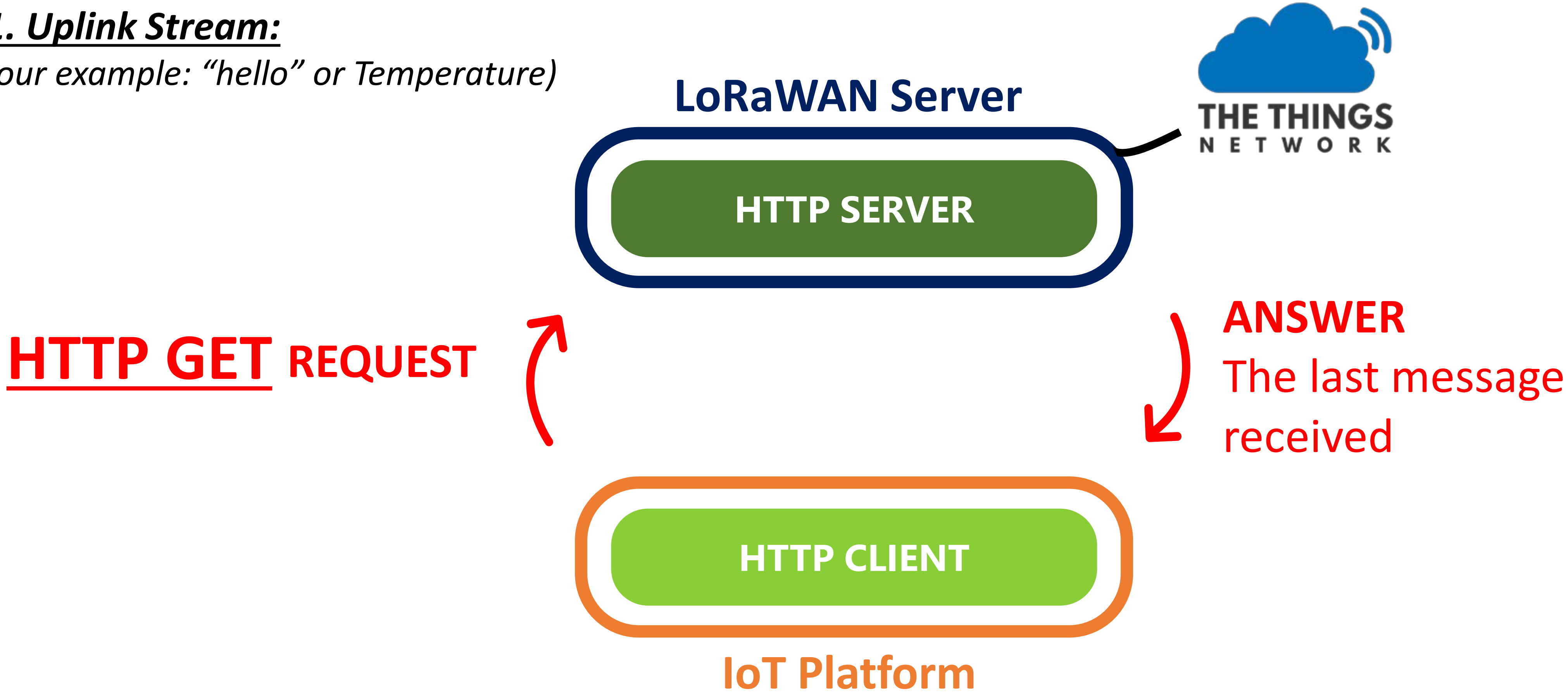
+ a way of **authentication**



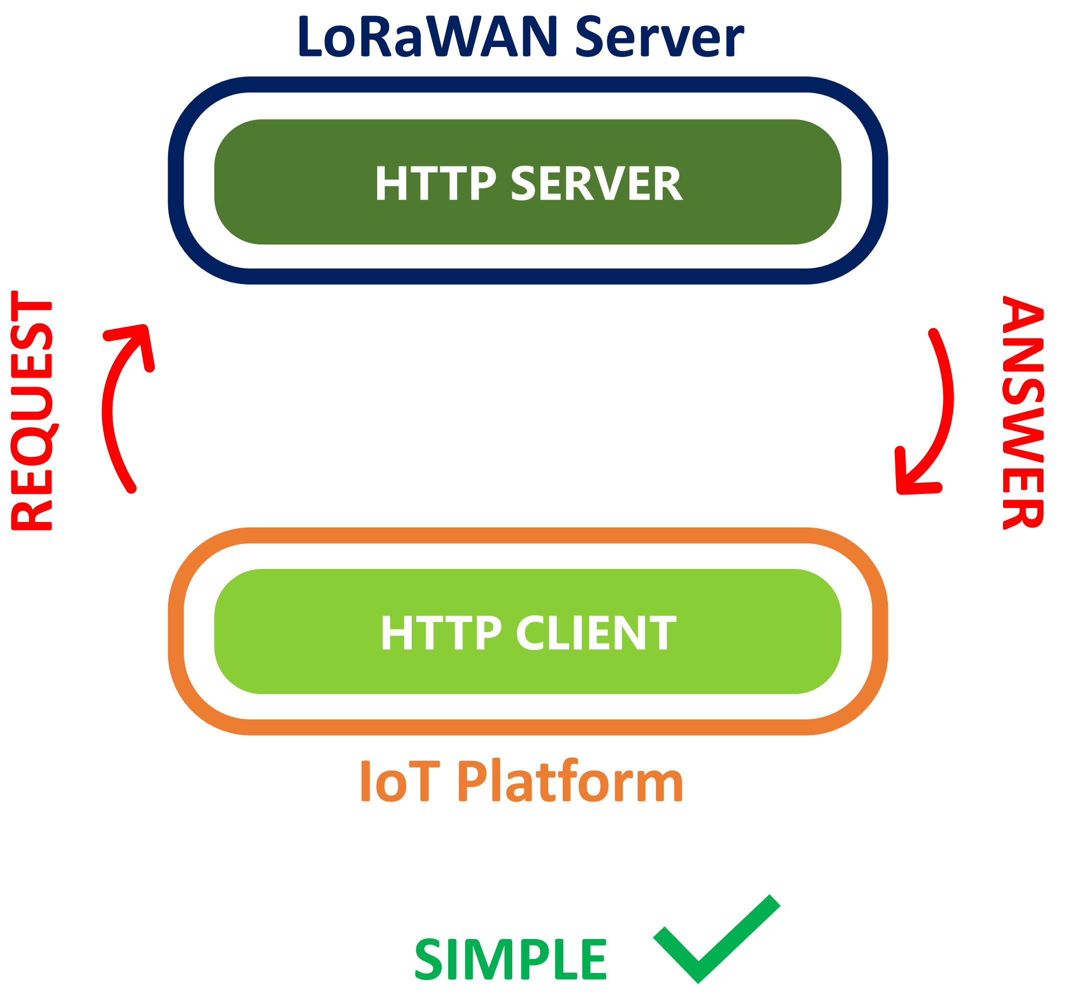
HTTP GET – Demonstration

1. Uplink Stream:

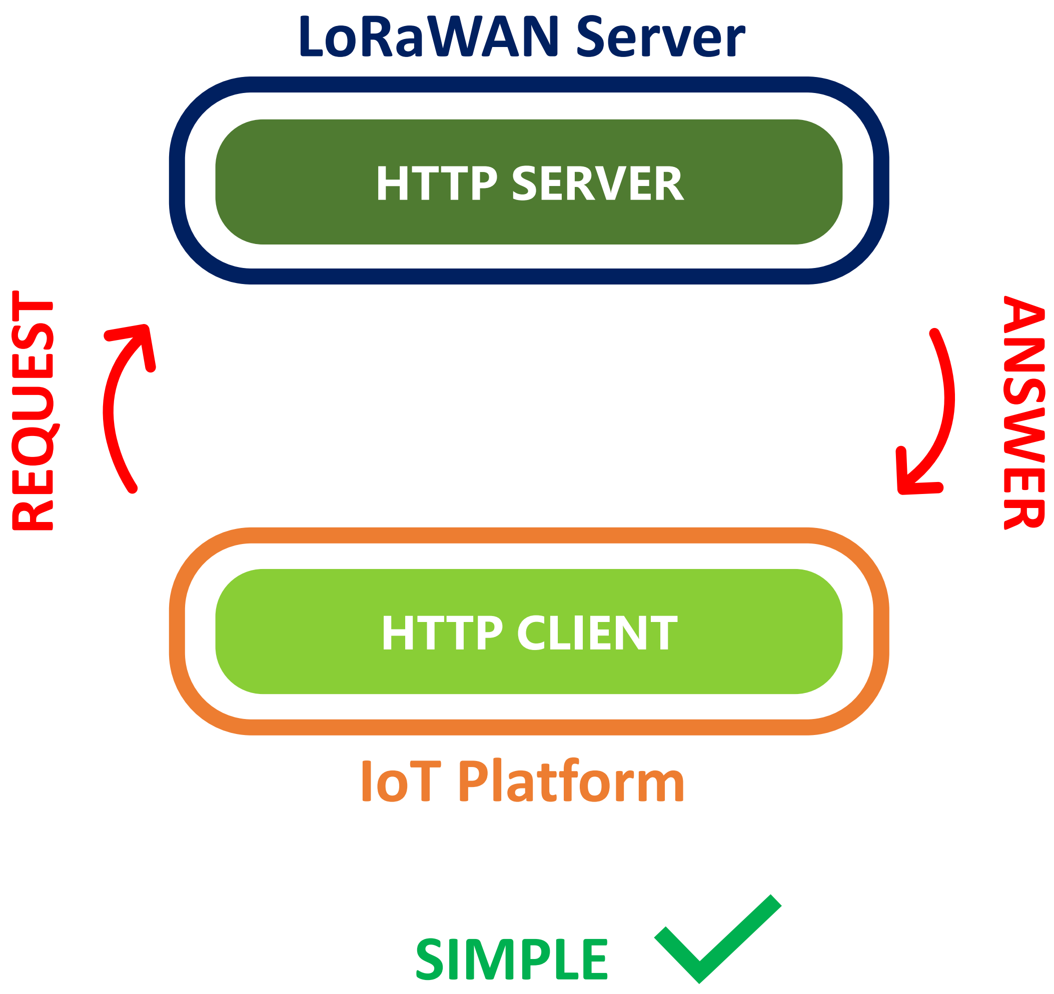
(our example: "hello" or Temperature)



HTTP GET Discussion



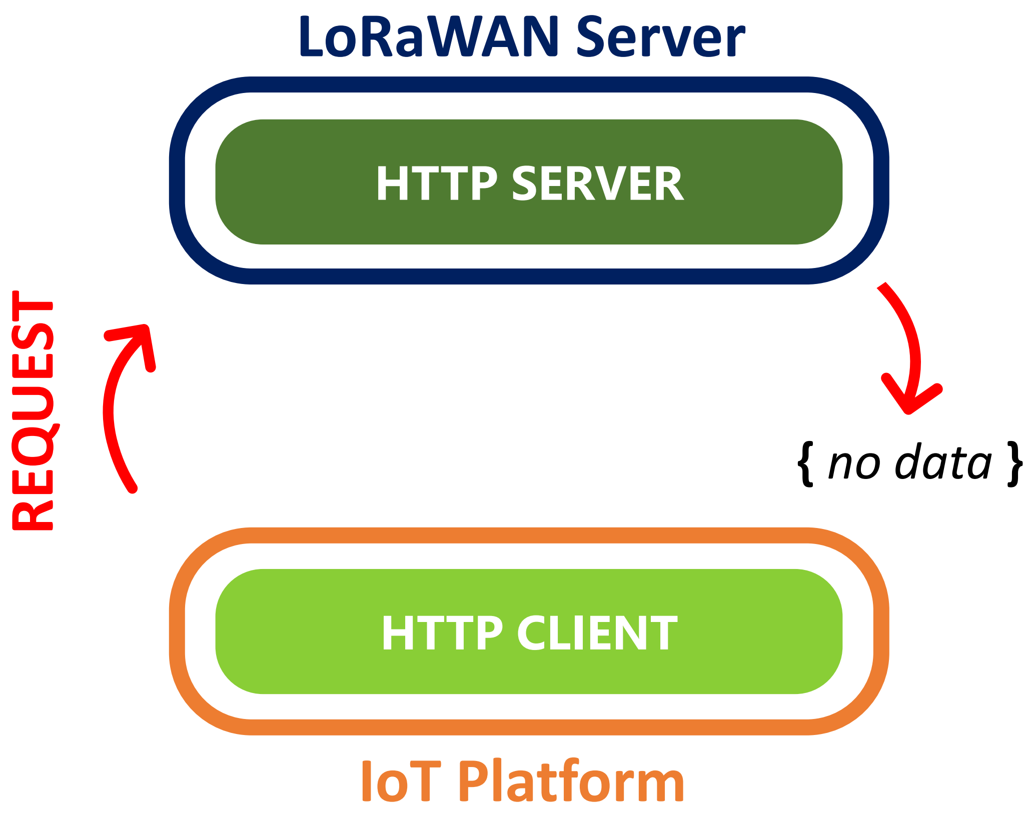
HTTP GET Discussion



1 Get the Uplink stream ONLY !

Cannot send Downlinks

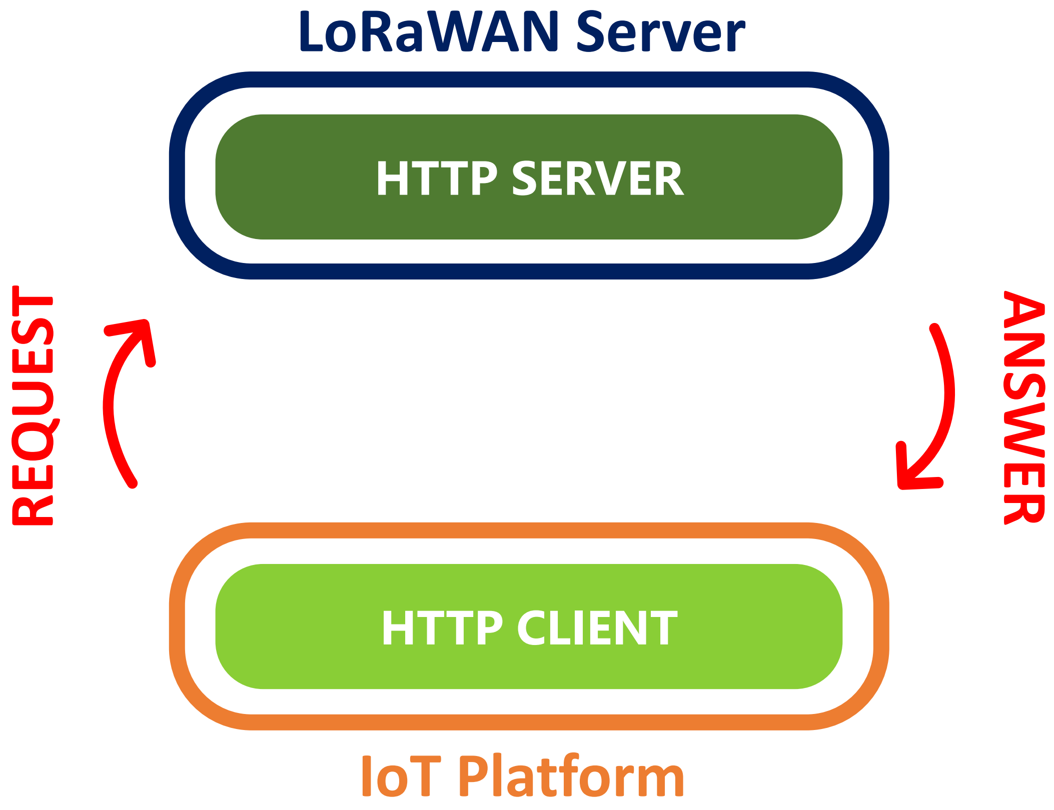
HTTP GET Discussion



1. Get the Uplink stream ONLY !
   Cannot send Downlinks ❌

2. Ask for data that doesn't exist !
   → Not efficient

HTTP GET Discussion

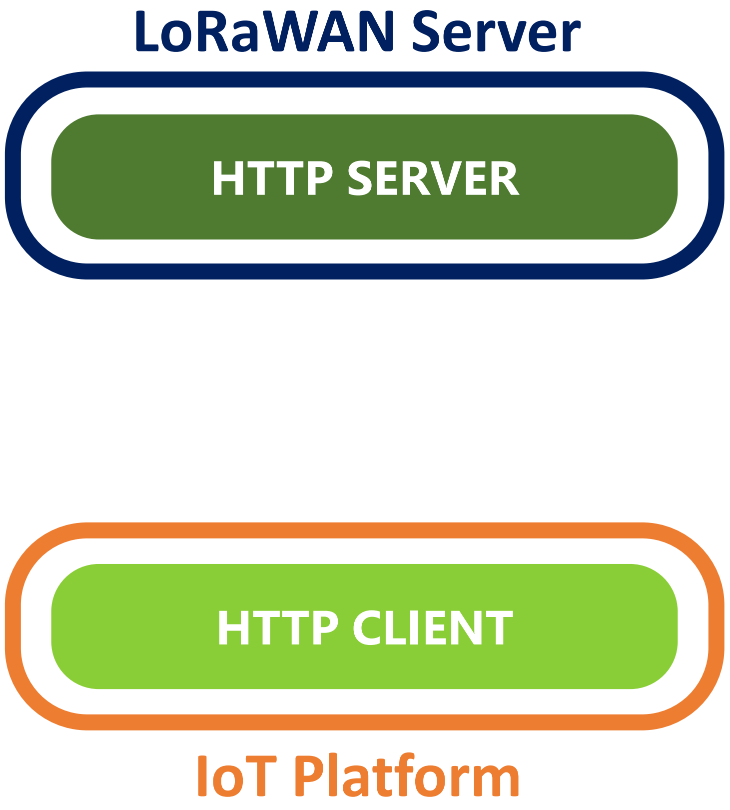


1. Get the Uplink stream ONLY !
   Cannot send Downlinks ❌

2. Ask for data that doesn't exist !
   → Not efficient

Then… Why "GET request" ?
→ Still useful in some cases

HTTP GET Discussion



1. **Get the <u>Uplink stream ONLY !</u>**
   Cannot send Downlinks ✗

2. **Ask for data that doesn't exist !**
   → Not efficient

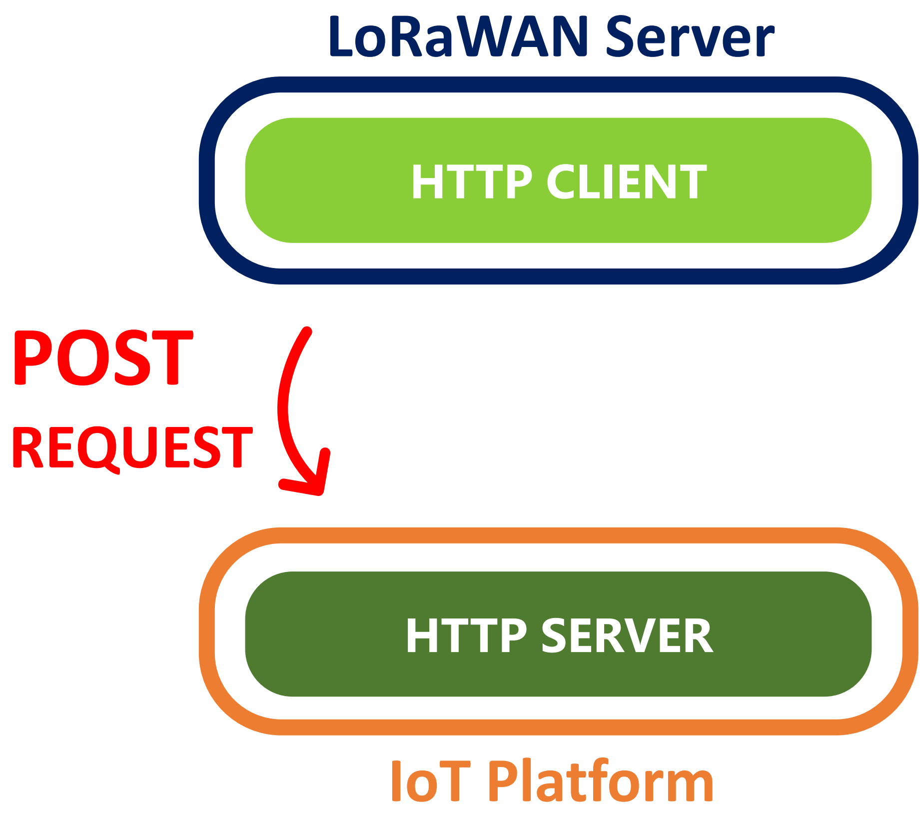
Then… Why "GET request" ?
→ Still useful in some cases

How to improve transmissions ?

IoT platform:

- ✗ **Ask for data**
- ✓ **Receive data**

HTTP GET Discussion



1 Get the Uplink stream ONLY !
Cannot send Downlinks ✗

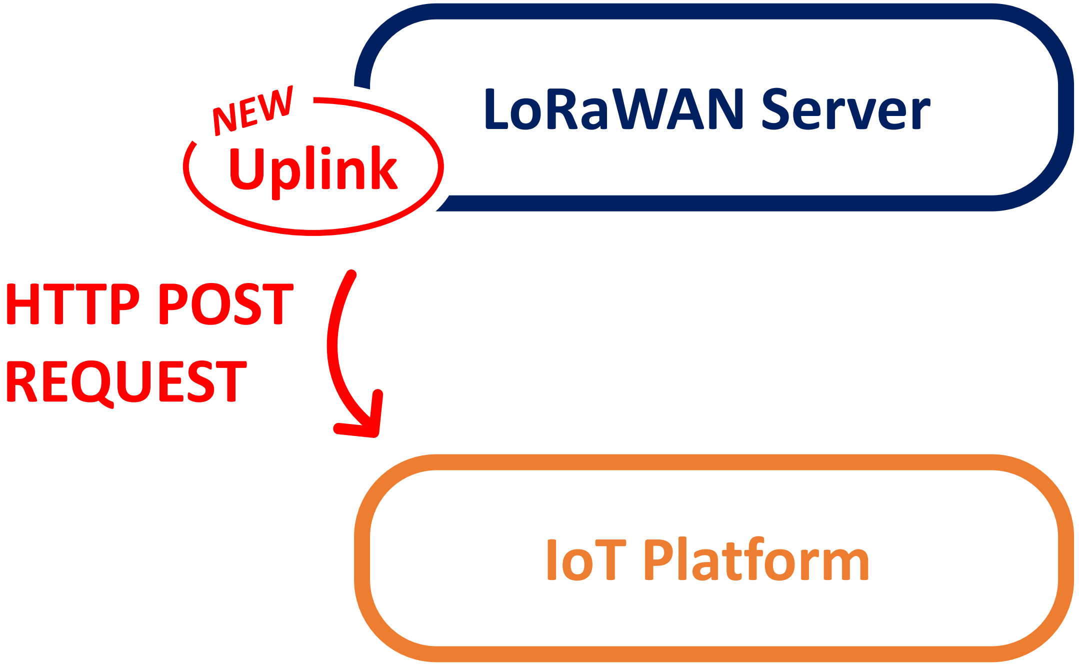
2 Ask for data that doesn’t exist !
→ Not efficient

Then… Why “GET request” ?
→ Still useful in some cases

How to improve transmissions ?
IoT platform:
✗ Ask for data
✓ Receive data

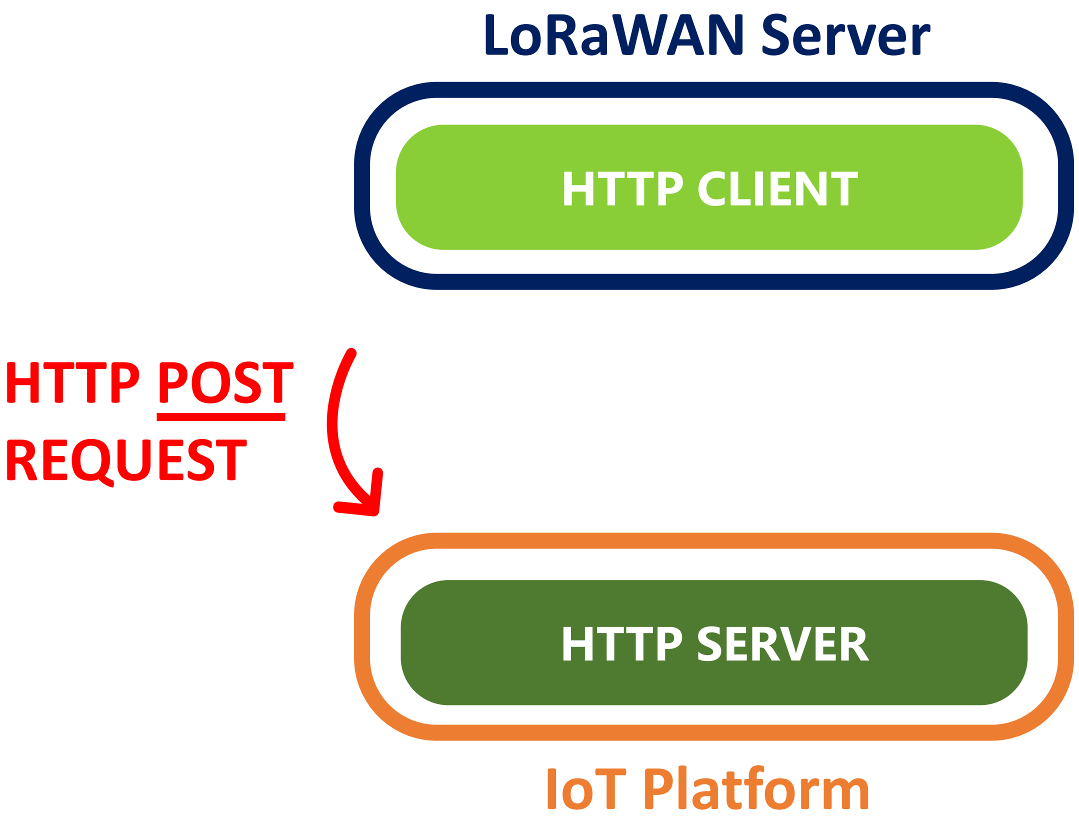
HTTP POST

1. Uplink Stream:



HTTP POST

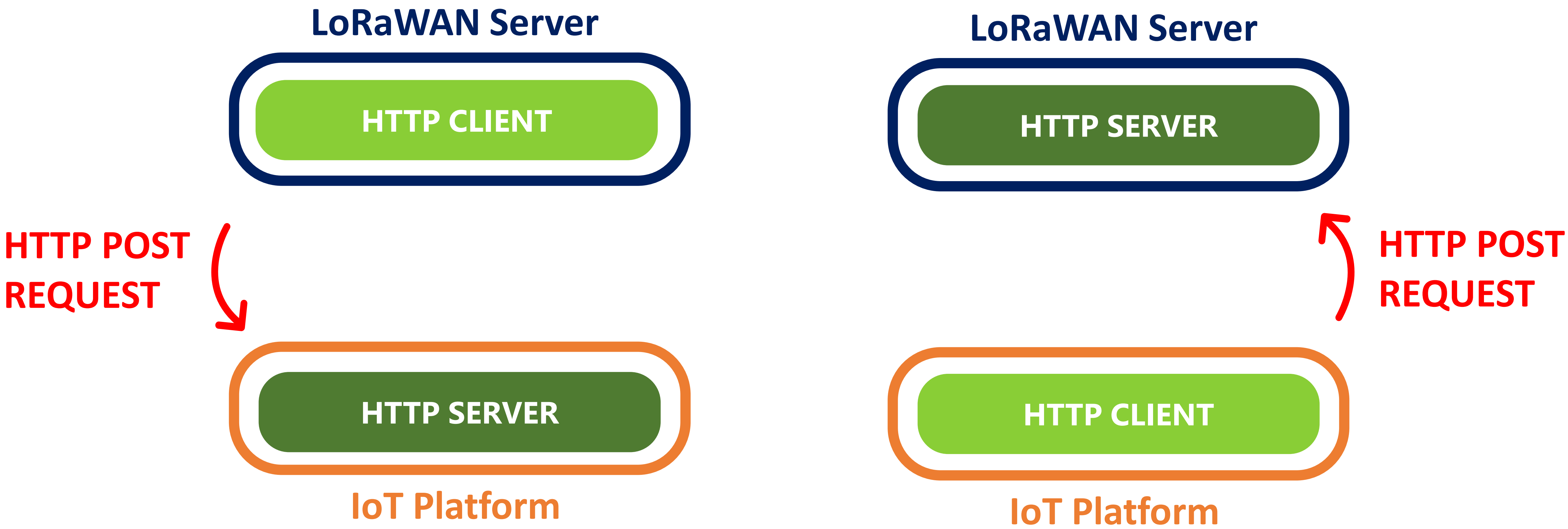
1. Uplink Stream:



HTTP POST

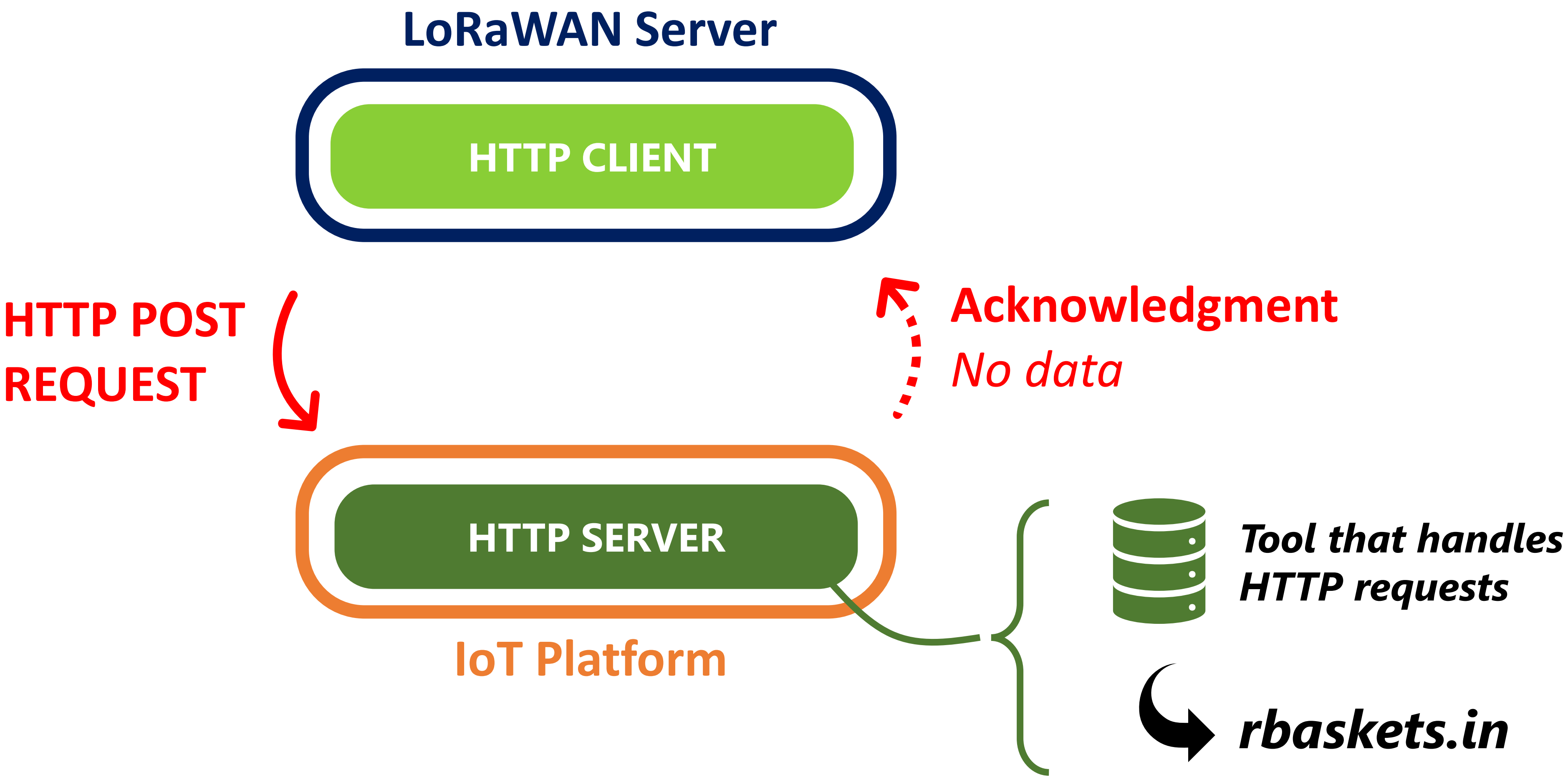
1. Uplink Stream:

2. Downlink Stream:



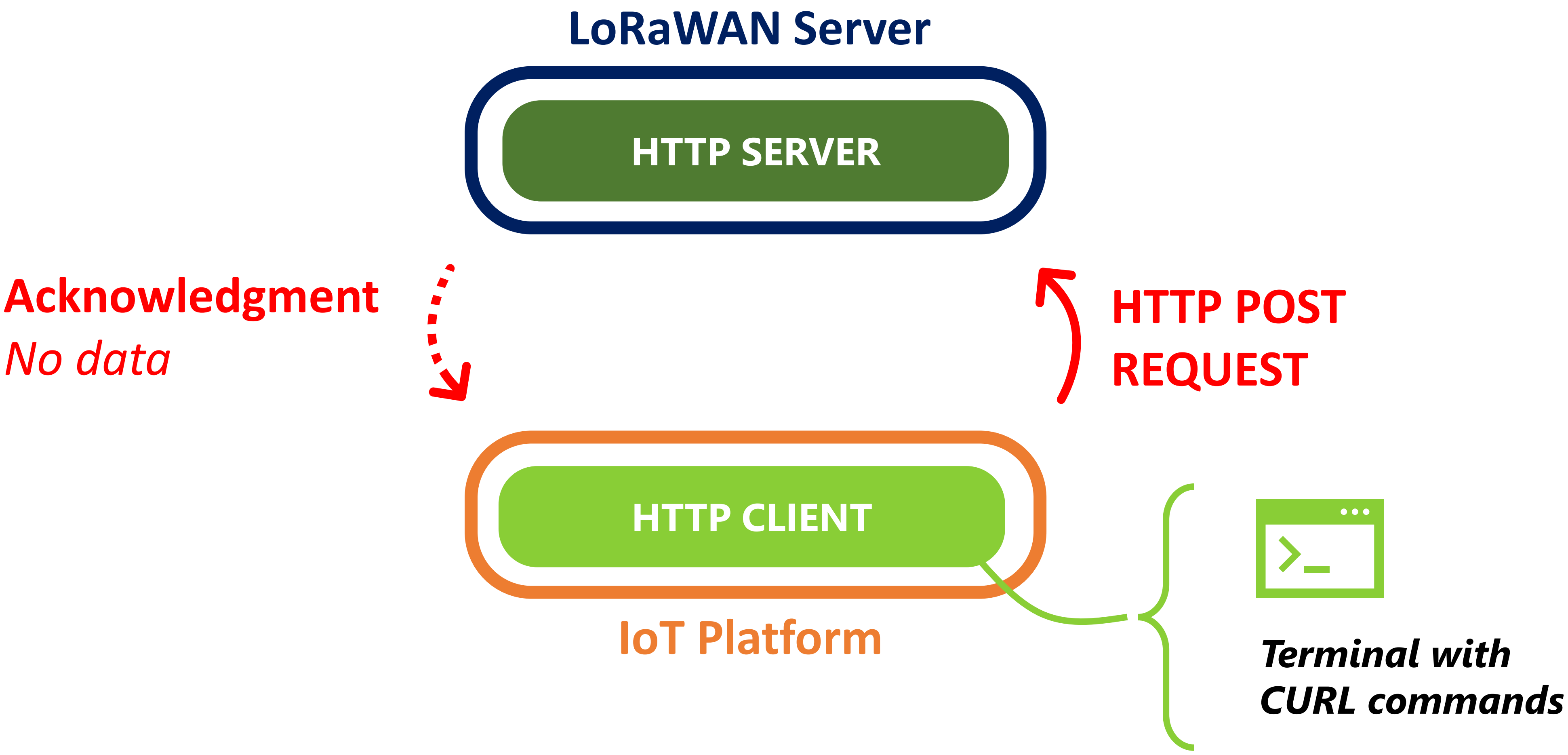
HTTP POST uplink - Demonstration with TTN

1. Uplink Stream:



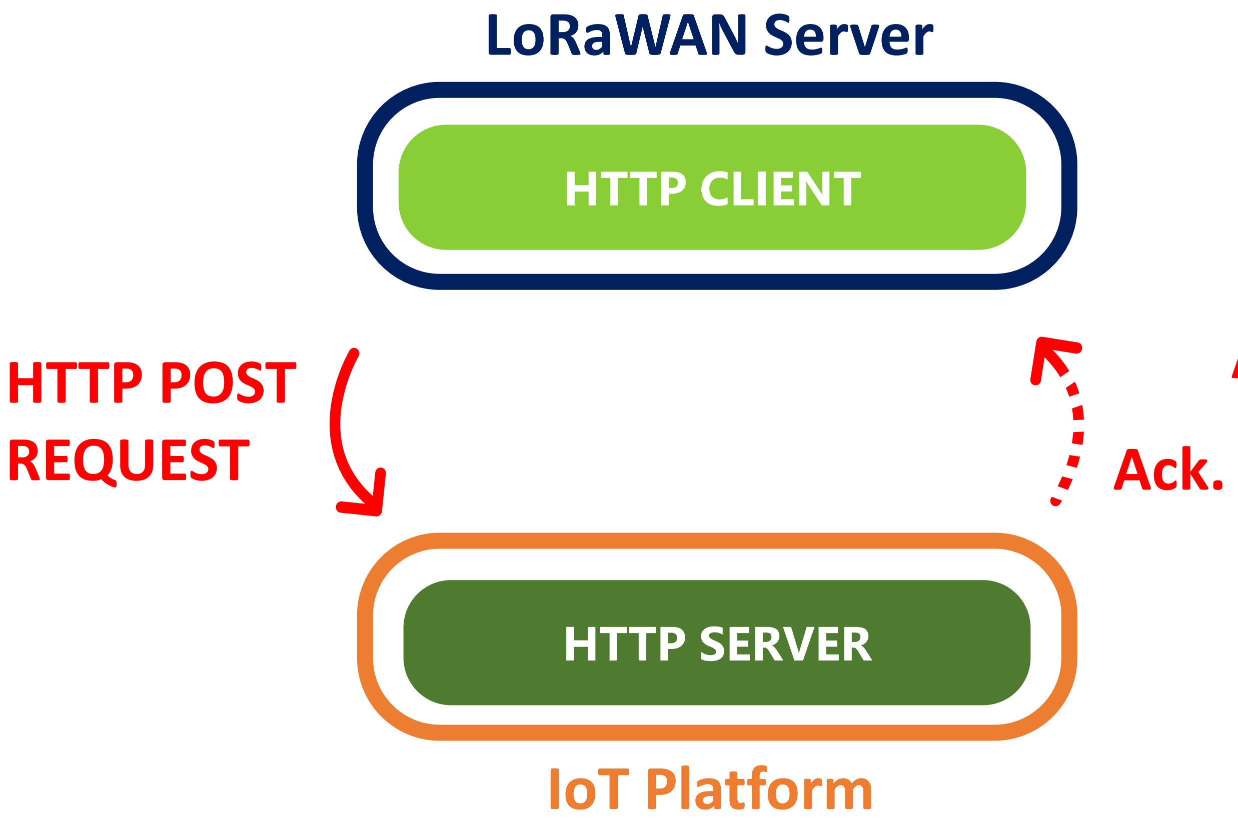
HTTP POST downlink - Demonstration with TTN

2. Downlink Stream:

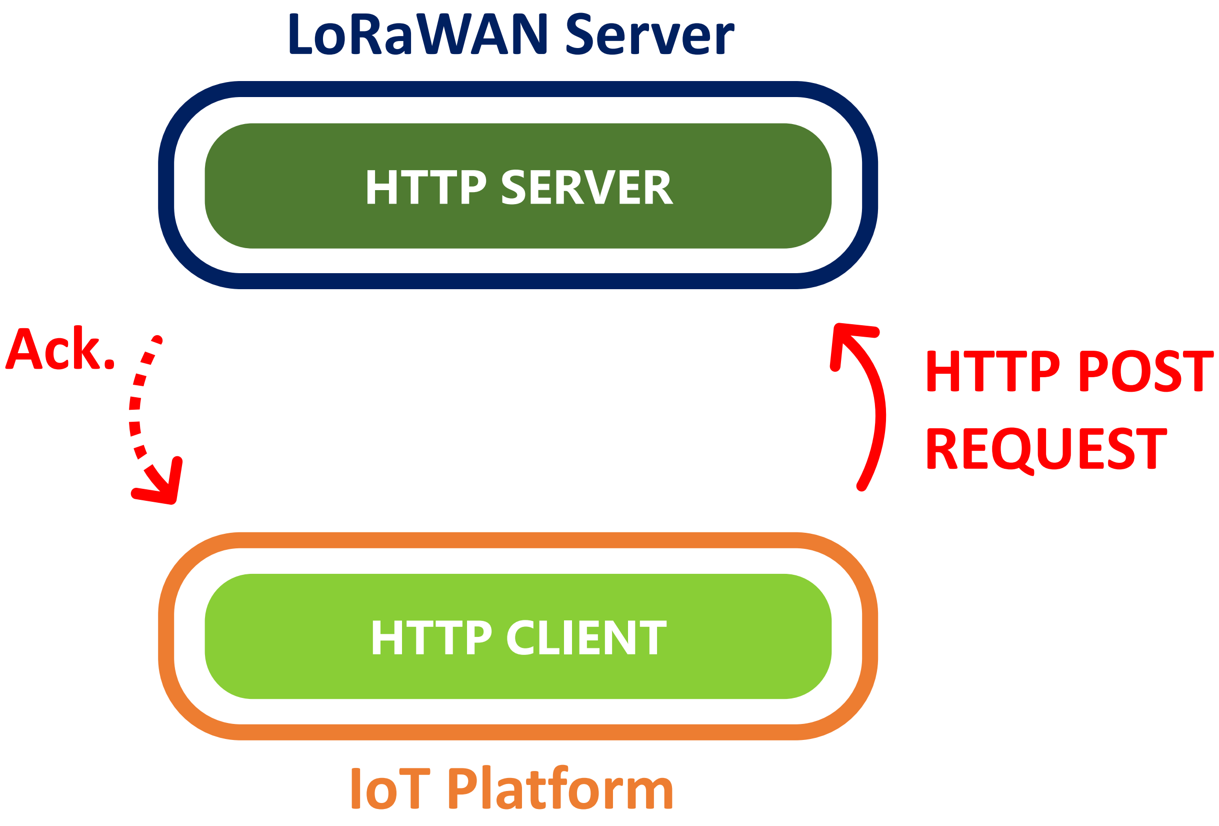


HTTP POST analysis

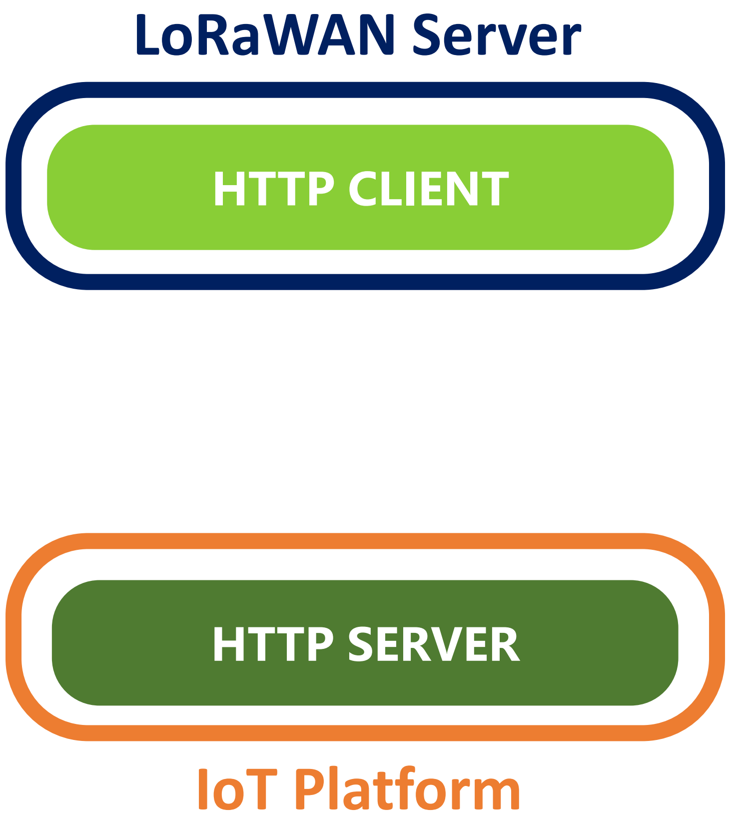
1. Uplink Stream:



2. Downlink Stream:

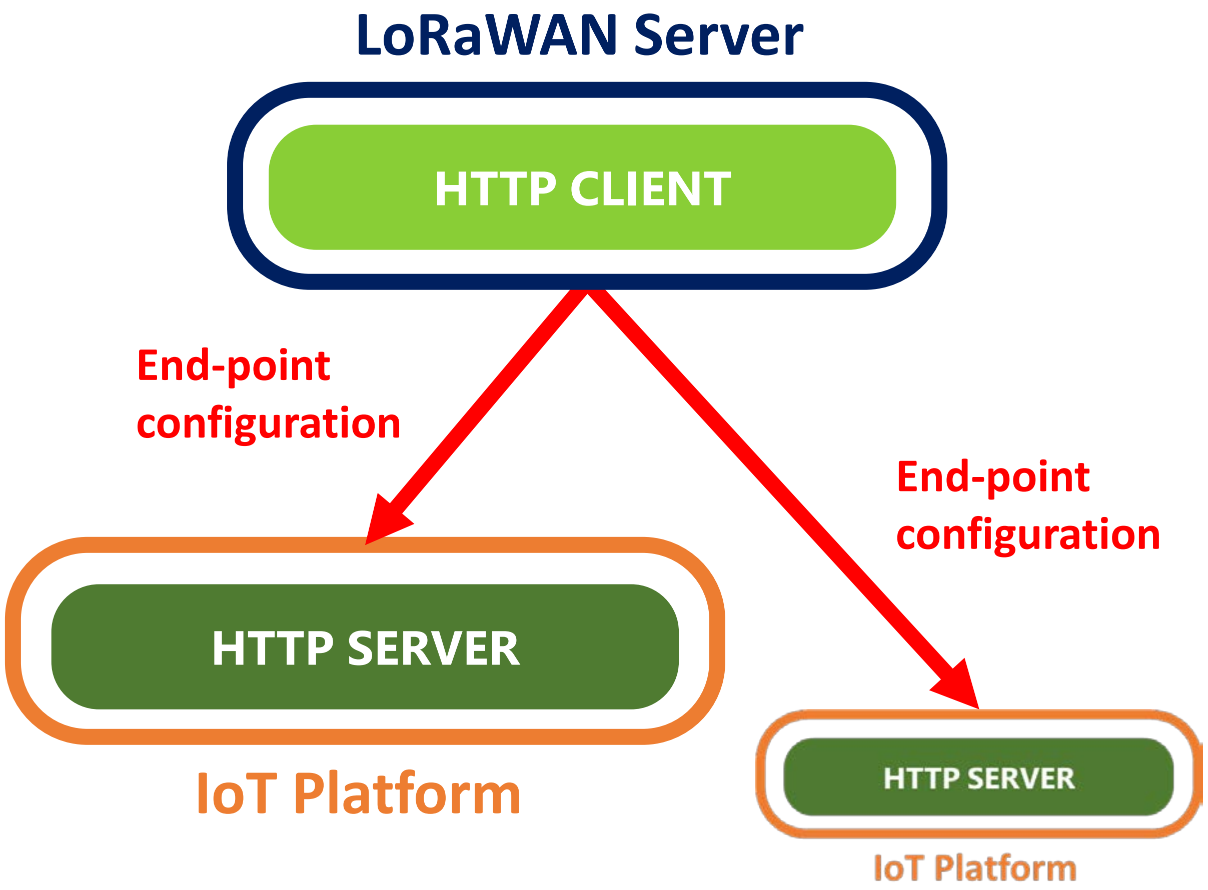


HTTP POST analysis



1. **What if… we connect another IoT platform**

HTTP POST analysis

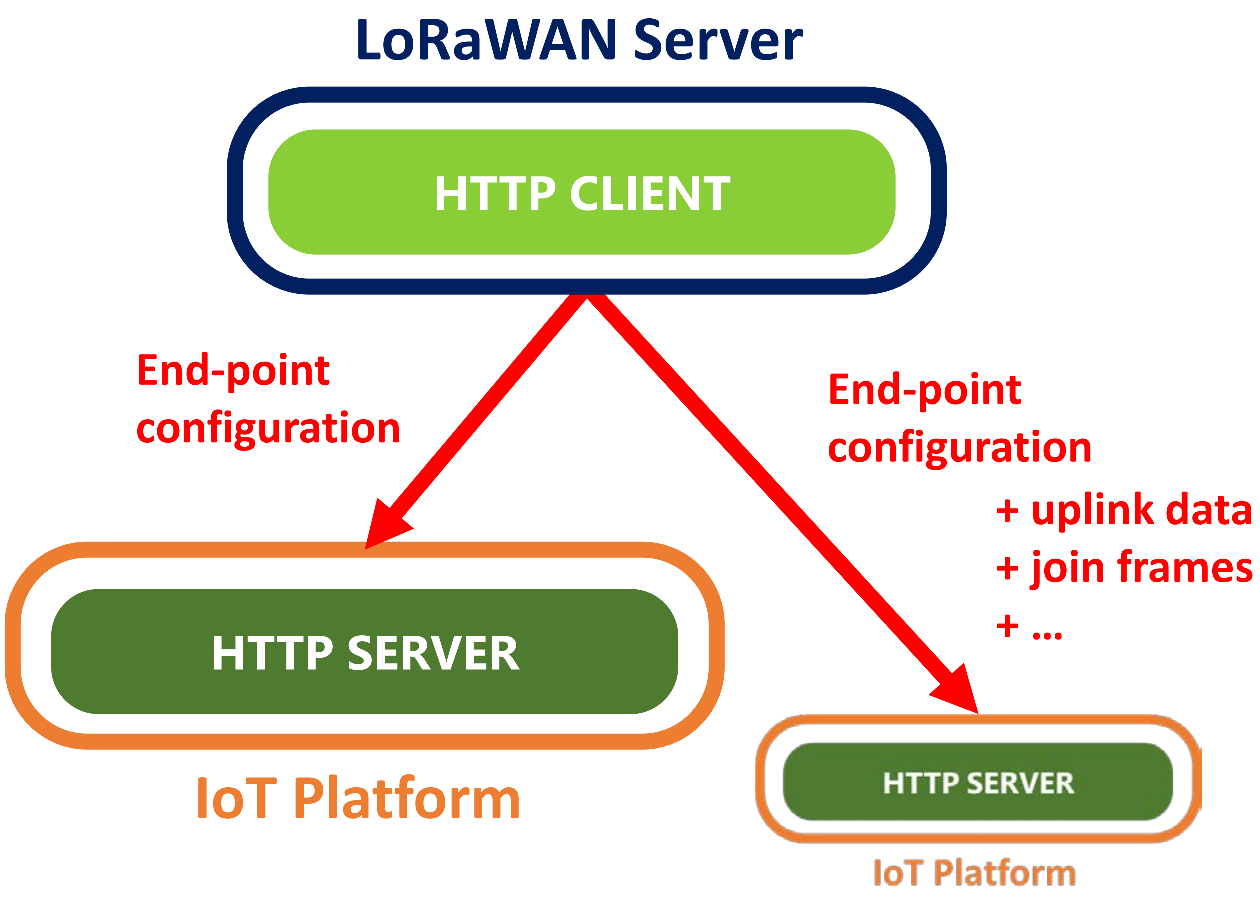


1. **What if… we connect another IoT platform**

   → Simple, but it still has to be done

2. **Definition of the events to be sent**

HTTP POST analysis



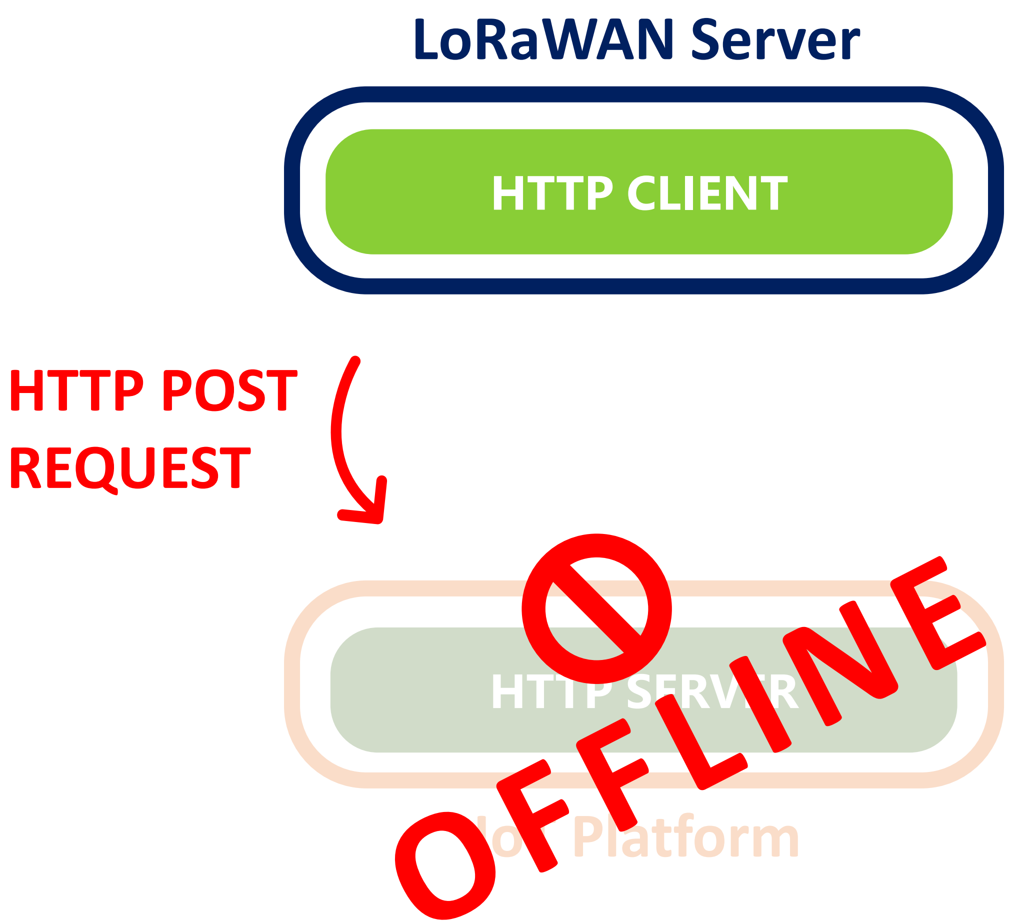
1. **What if… we connect another IoT platform**

   → Simple, but it still has to be done

2. **Definition of the events to be sent**

   → Simple, but it still has to be done

HTTP POST analysis



1. **What if… we connect another IoT platform**

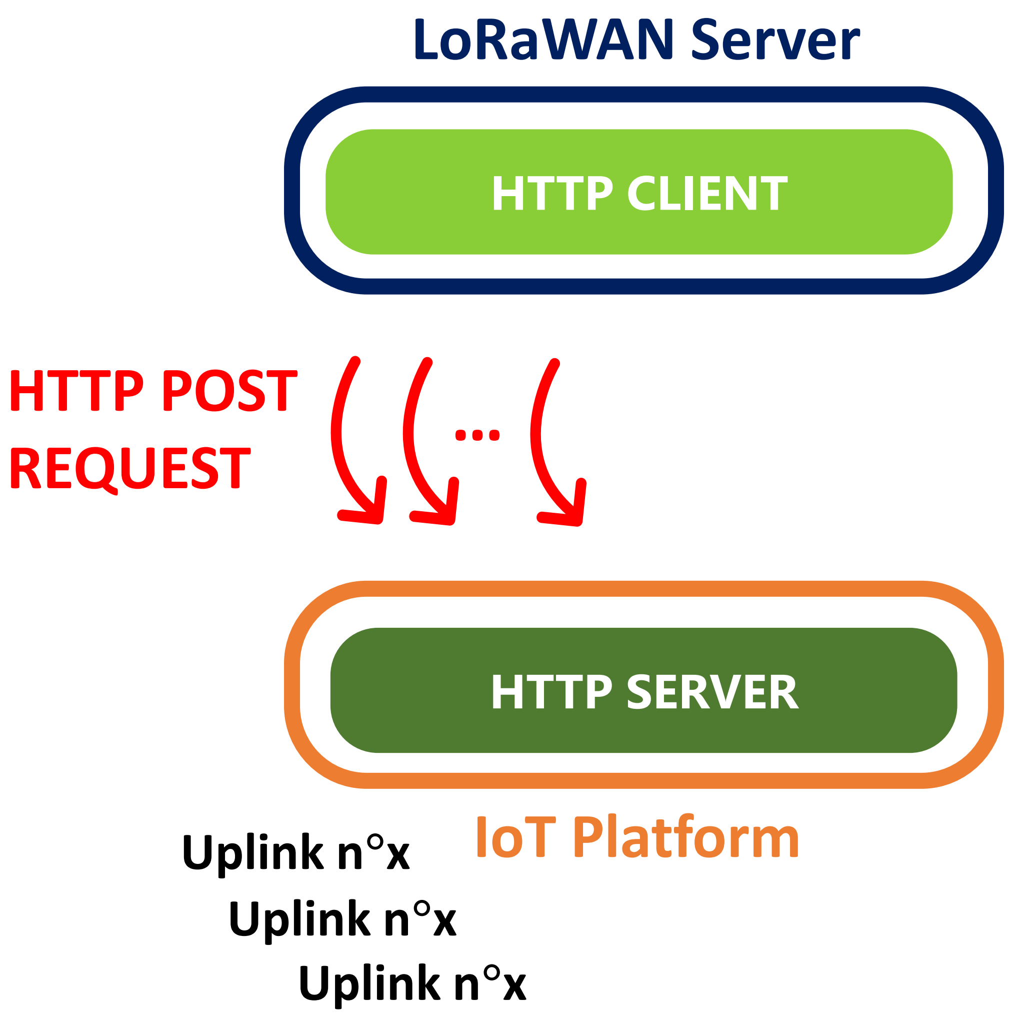
   → Simple, but it still has to be done

2. **Definition of the events to be sent**

   → Simple, but it still has to be done

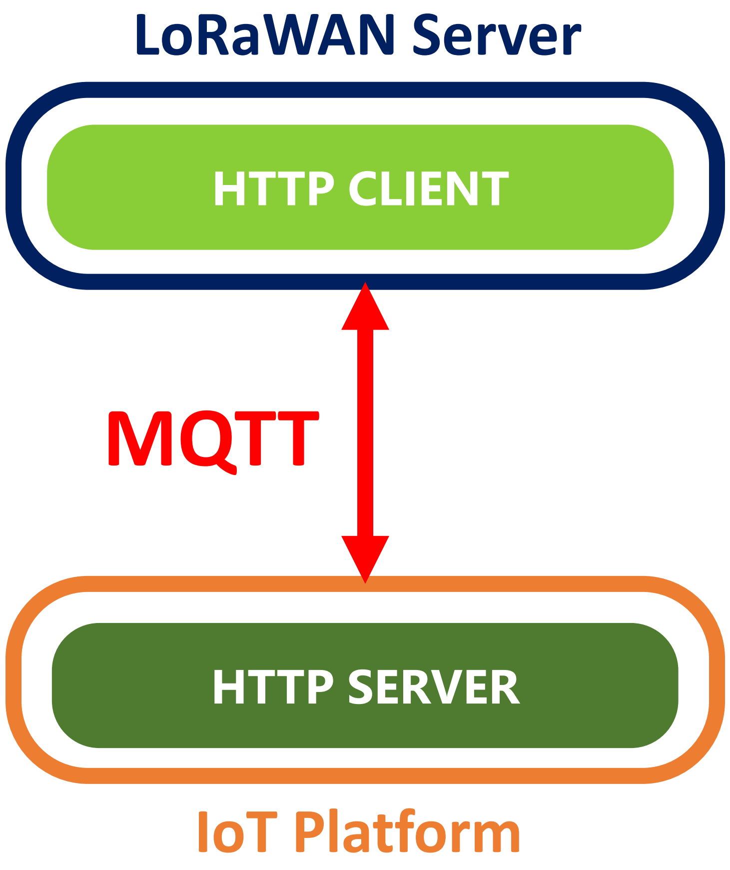
3. **The risk of losing data**

HTTP POST analysis



1. **What if… we connect another IoT platform**
   → Simple, but it still has to be done
2. **Definition of the events to be sent**
   → Simple, but it still has to be done
3. **The risk of losing data**
4. **The risk of having duplicate data**

HTTP POST analysis

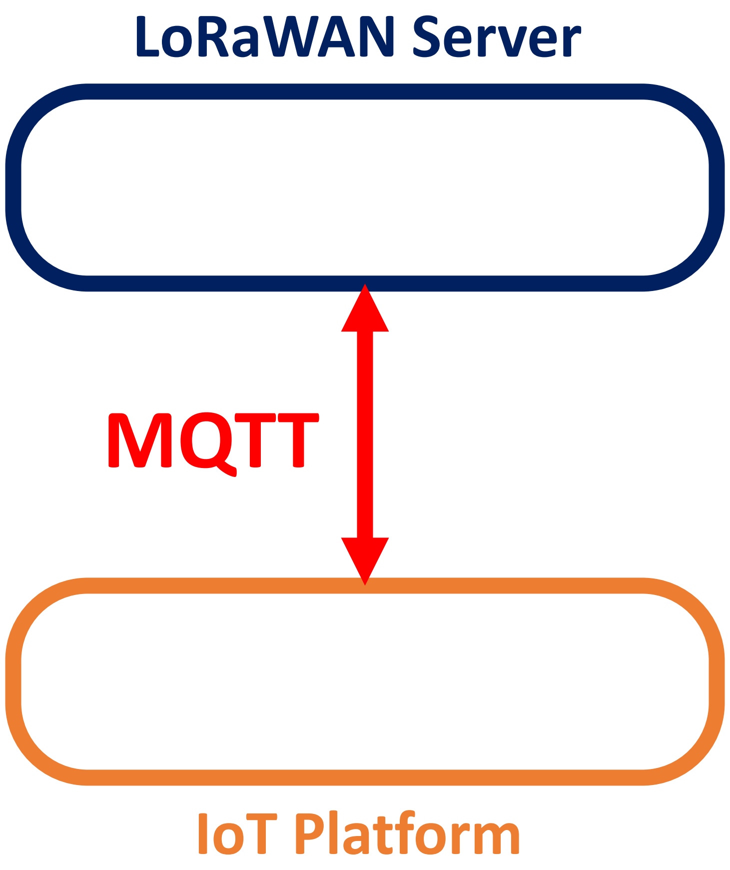


1. **What if… we connect another IoT platform**
   → Simple, but it still has to be done
2. **Definition of the events to be sent**
   → Simple, but it still has to be done
3. **The risk of losing data**
4. **The risk of having duplicate data**

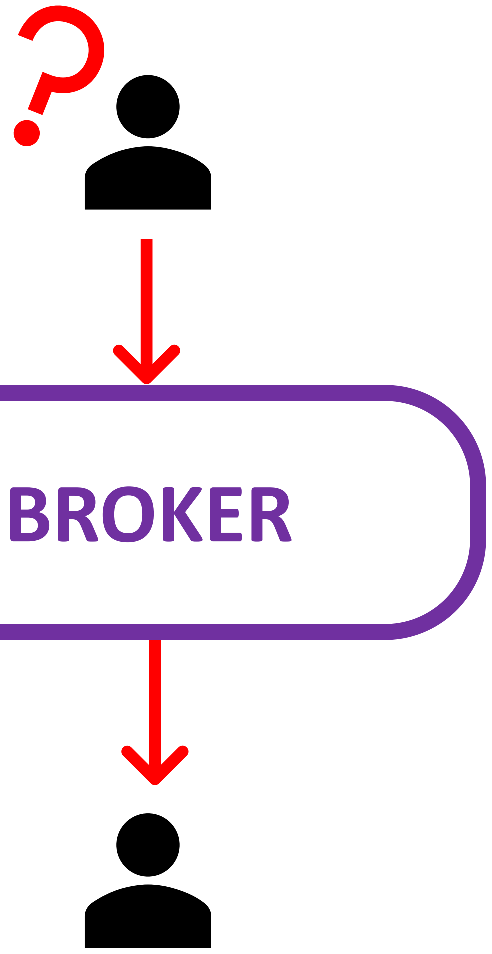
HTTP POST is great ! However, let's see another protocol : MQTT

MQTT - Publisher - Broker - Subscriber

MQTT = **M**essage **Q**ueuing **T**elemetry **T**ransport

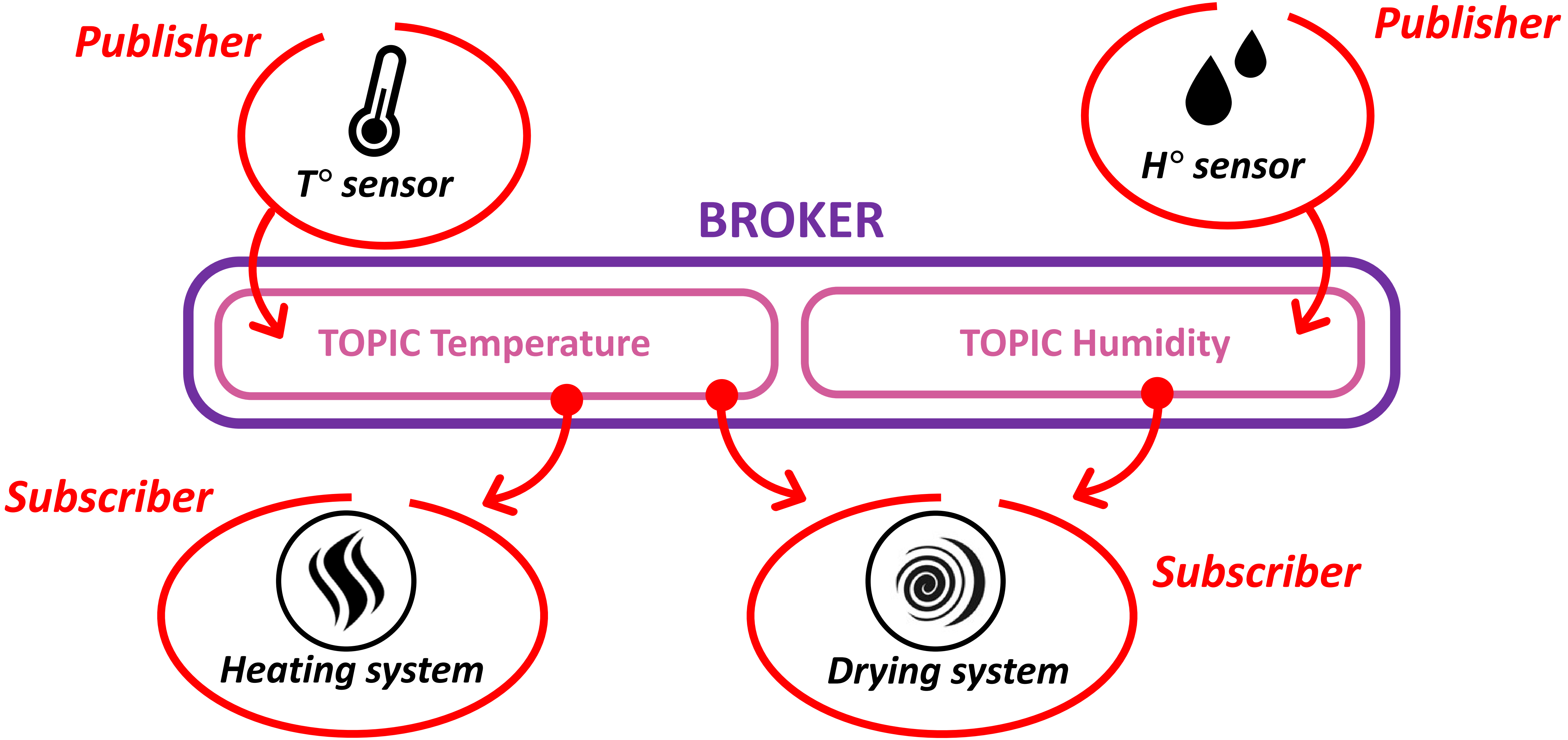


MQTT - Publisher - Broker - Subscriber

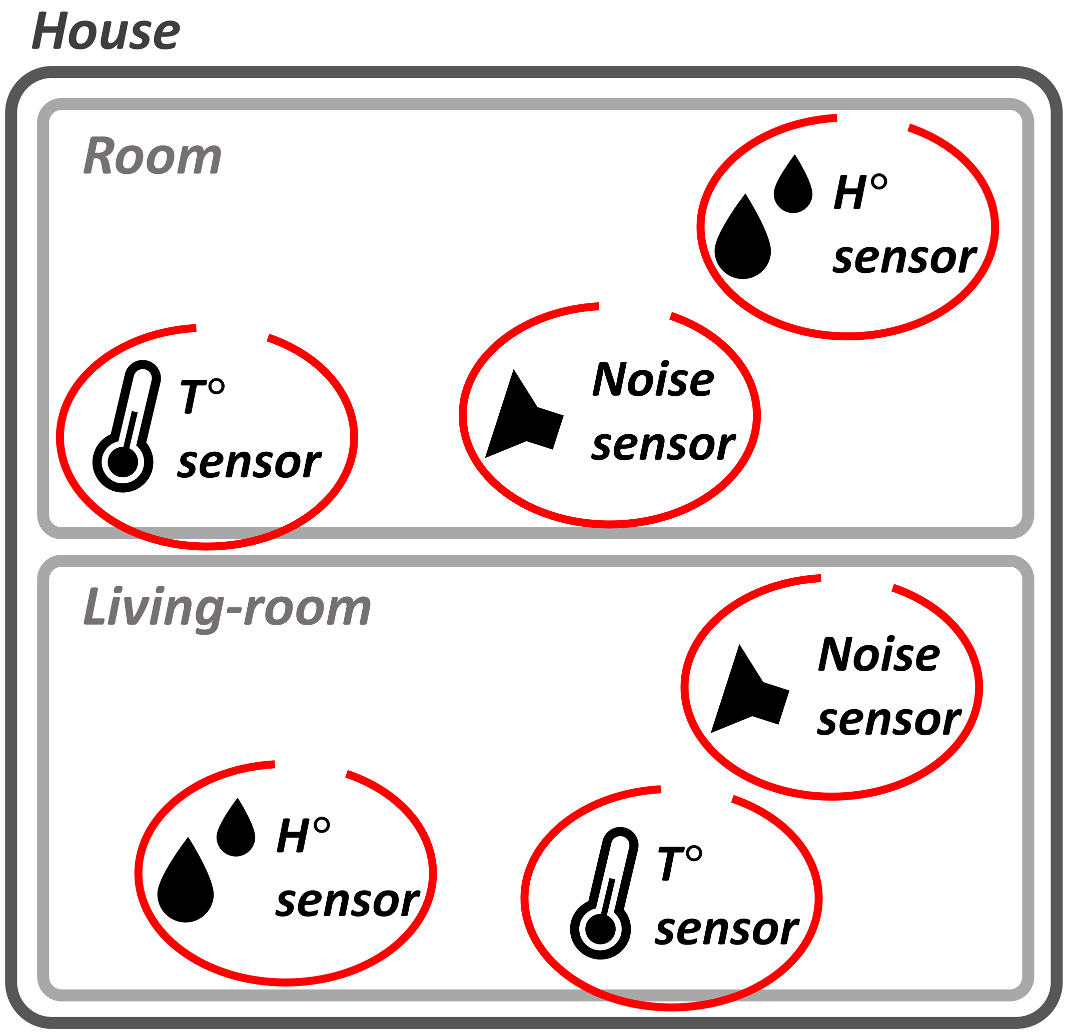


MQTT - Publisher - Broker - Subscriber

MQTT - Publisher - Broker - Subscriber



MQTT - Publisher - Broker - Subscriber



I'm looking for ***the temperature in the bedroom of the house:***

TOPIC → House/Room/T°sensor

I'm looking for ***the temperature in the living-room of the house:***

TOPIC → House/Living-room/T°sensor

I'm looking for ***all the sensors in the house:***

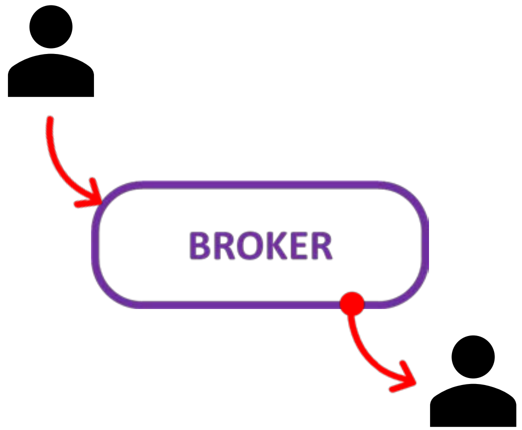
TOPIC → House/# or #

I'm looking for ***all temperatures in the house:***

TOPIC → House/+/T°sensor

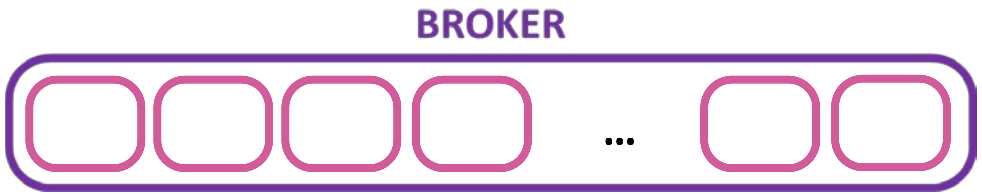
MQTT - Publisher - Broker - Subscriber

MQTT characteristics

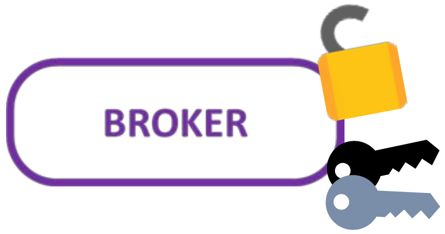


Publisher and subscriber don't know each other.

Infinite number of topics.
Topics are created by the <u>publishers.</u>



Authentication procedure to publish or subscribe to the broker.



MQTT – Demonstration with a public broker

Our demonstration

MQTT – Demonstration with a public broker

Our demonstration

BROKER

We could set up our own

or

We can use a public broker

A famous one:

mosquitto

Another example:

HIVEMQ

Publisher

Subscriber

MQTT client

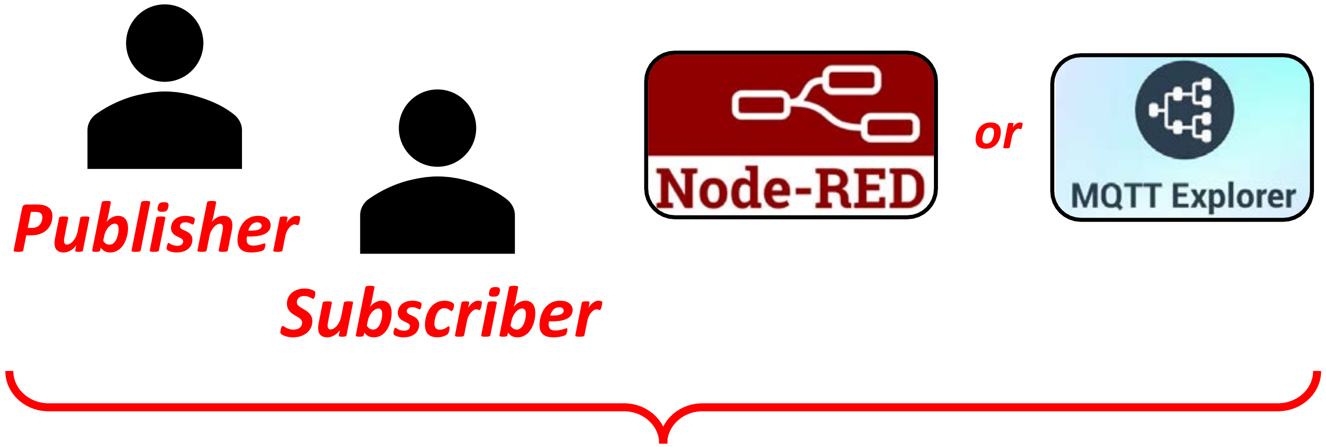
- Client controlled by a Command Line Interface (CLI)
- Client using Node-RED
  Node-RED
- Client with a standalone user interface
  E.g.: MQTT Explorer

MQTT – Demonstration with a public broker

Our demonstration

BROKER

mosquitto



Client configuration:

- **Server we want to connect to**
- **The Login and password for the connection**
- **The topic we want to target (for publisher). The topic we want to read (for subscriber)**
- **The message to send to the specified topic (for publisher)**

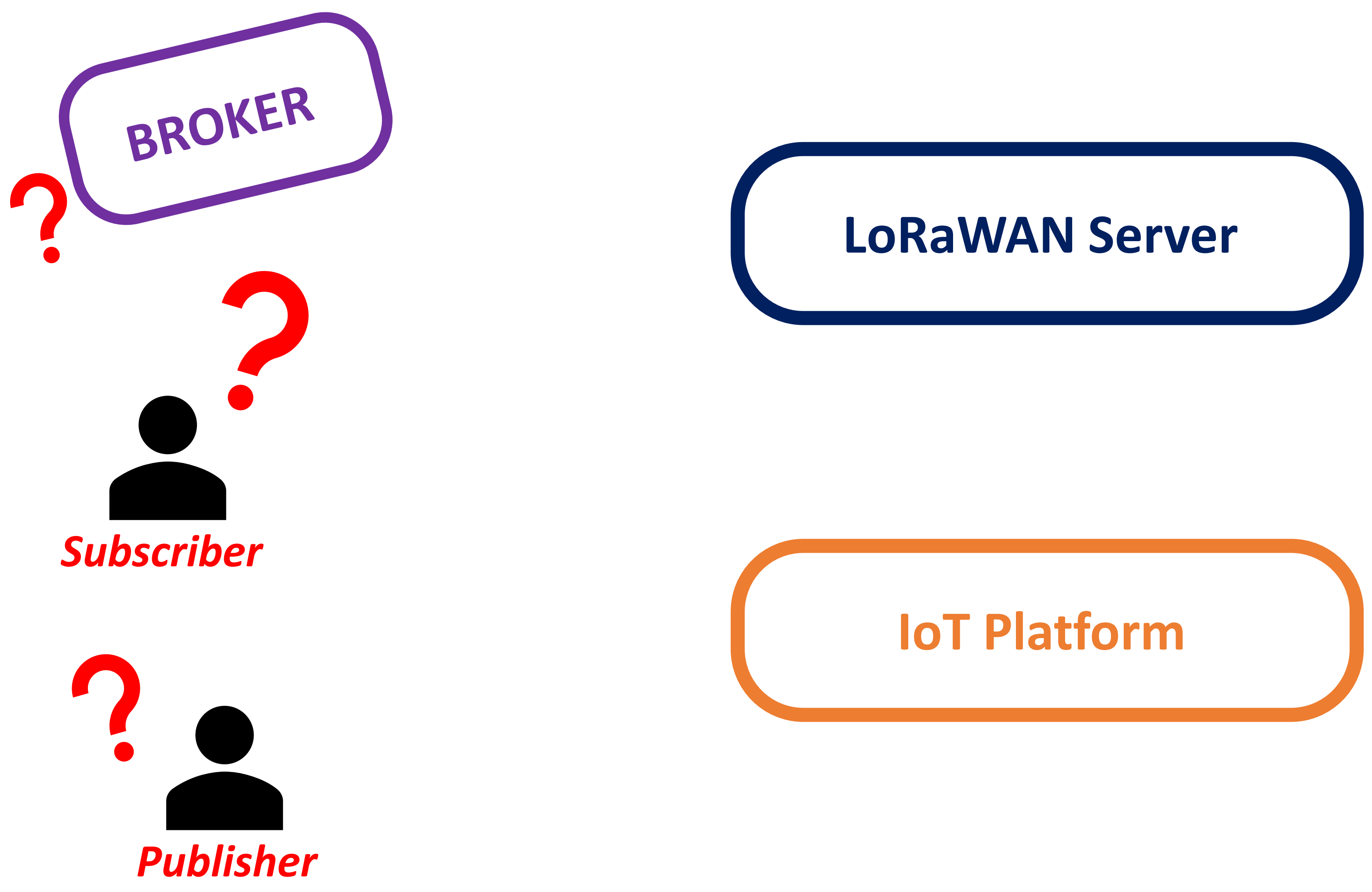
MQTT – Demonstration with a public broker

Our demonstration

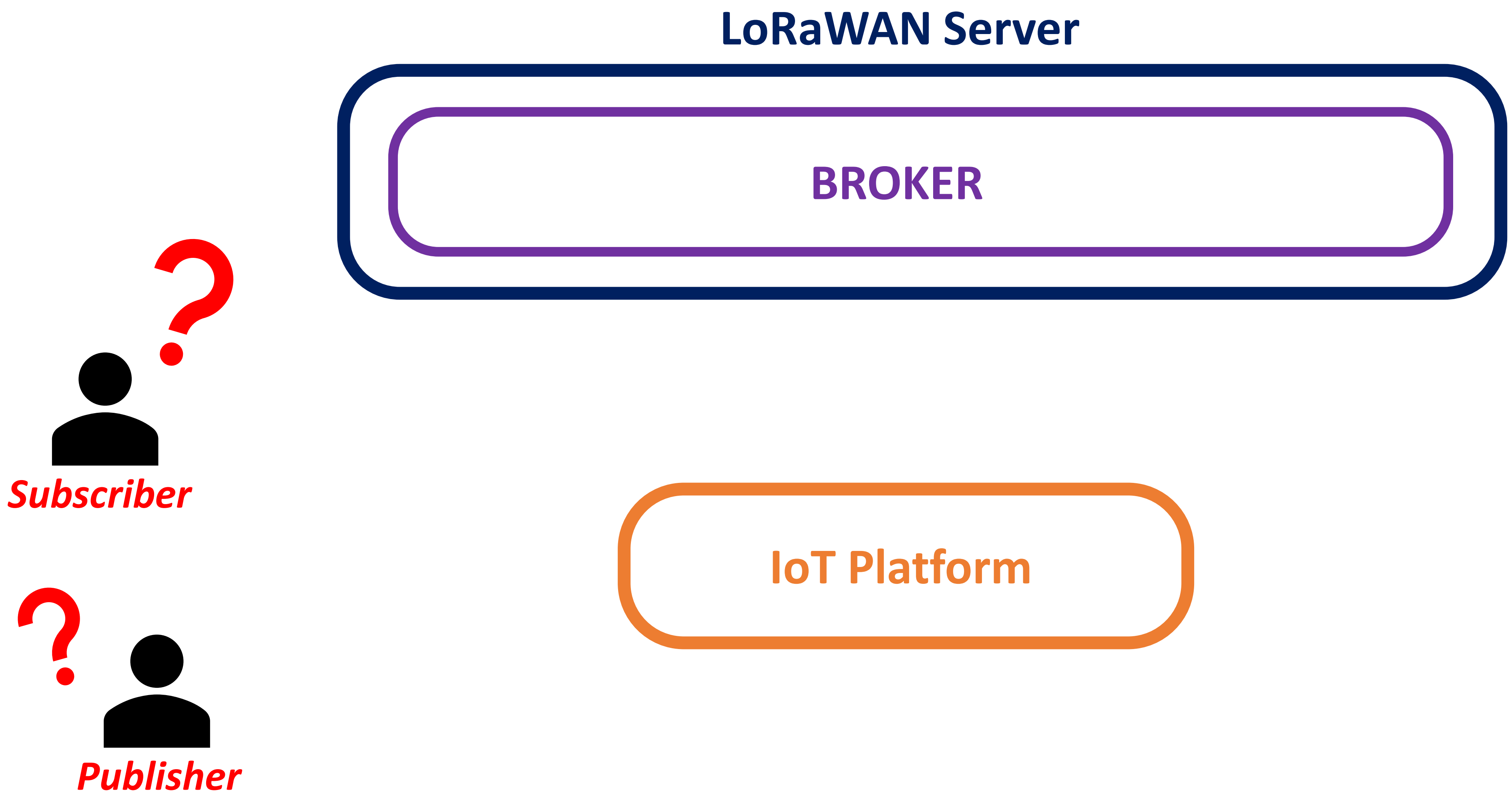
First part of the demonstration

Second part of the demonstration

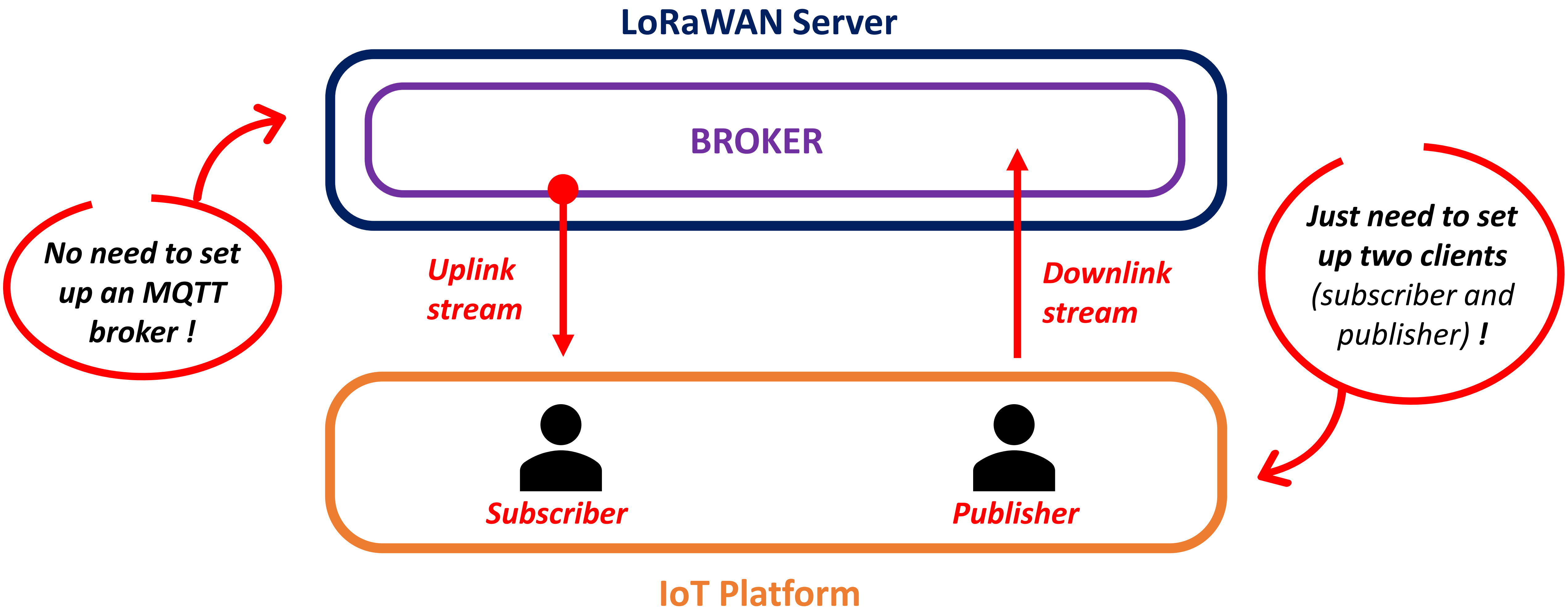
MQTT - LoRaWAN server as a broker



MQTT - LoRaWAN server as a broker



MQTT - LoRaWAN server as a broker

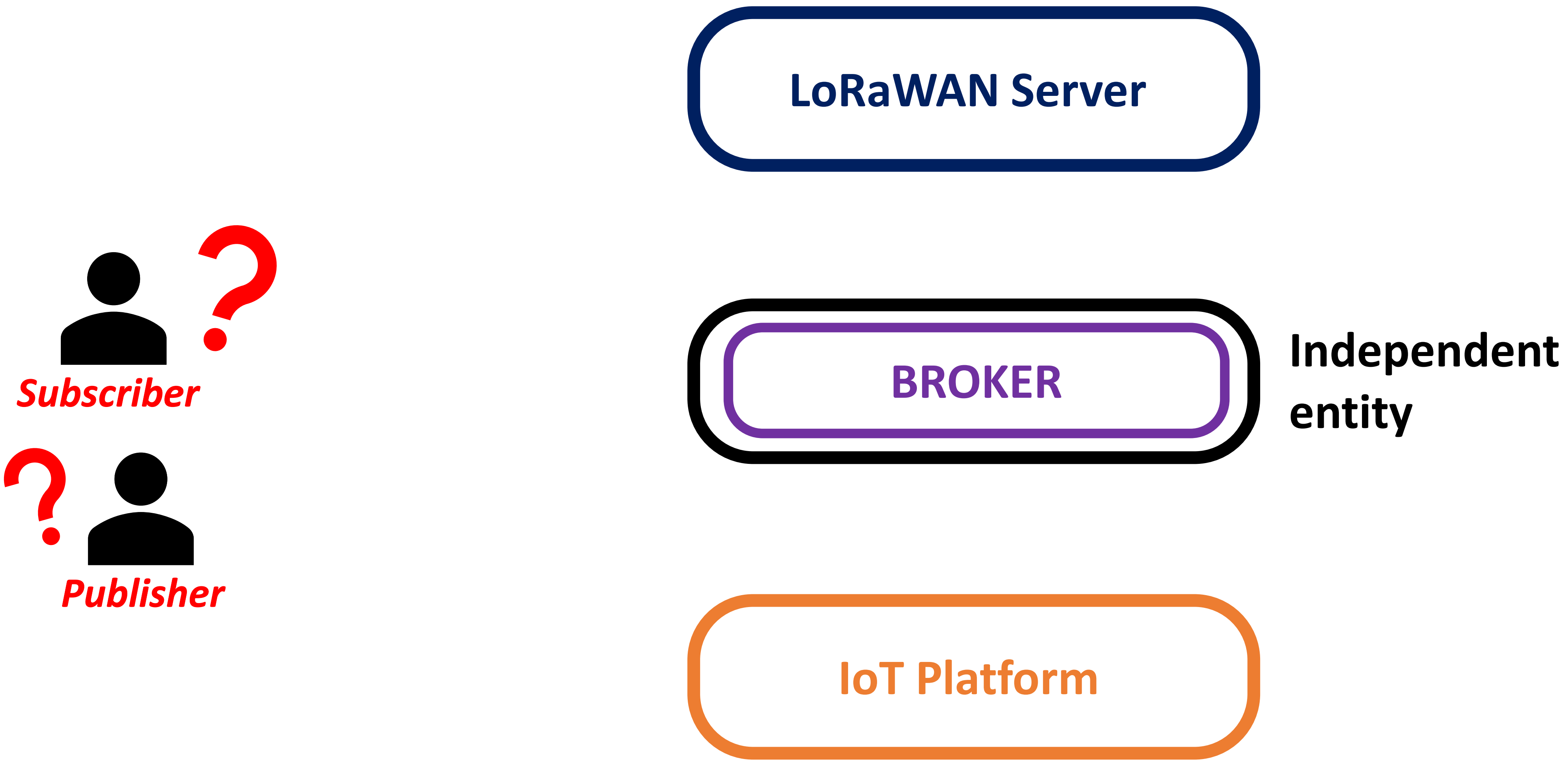


MQTT - LoRaWAN server as a MQTT client

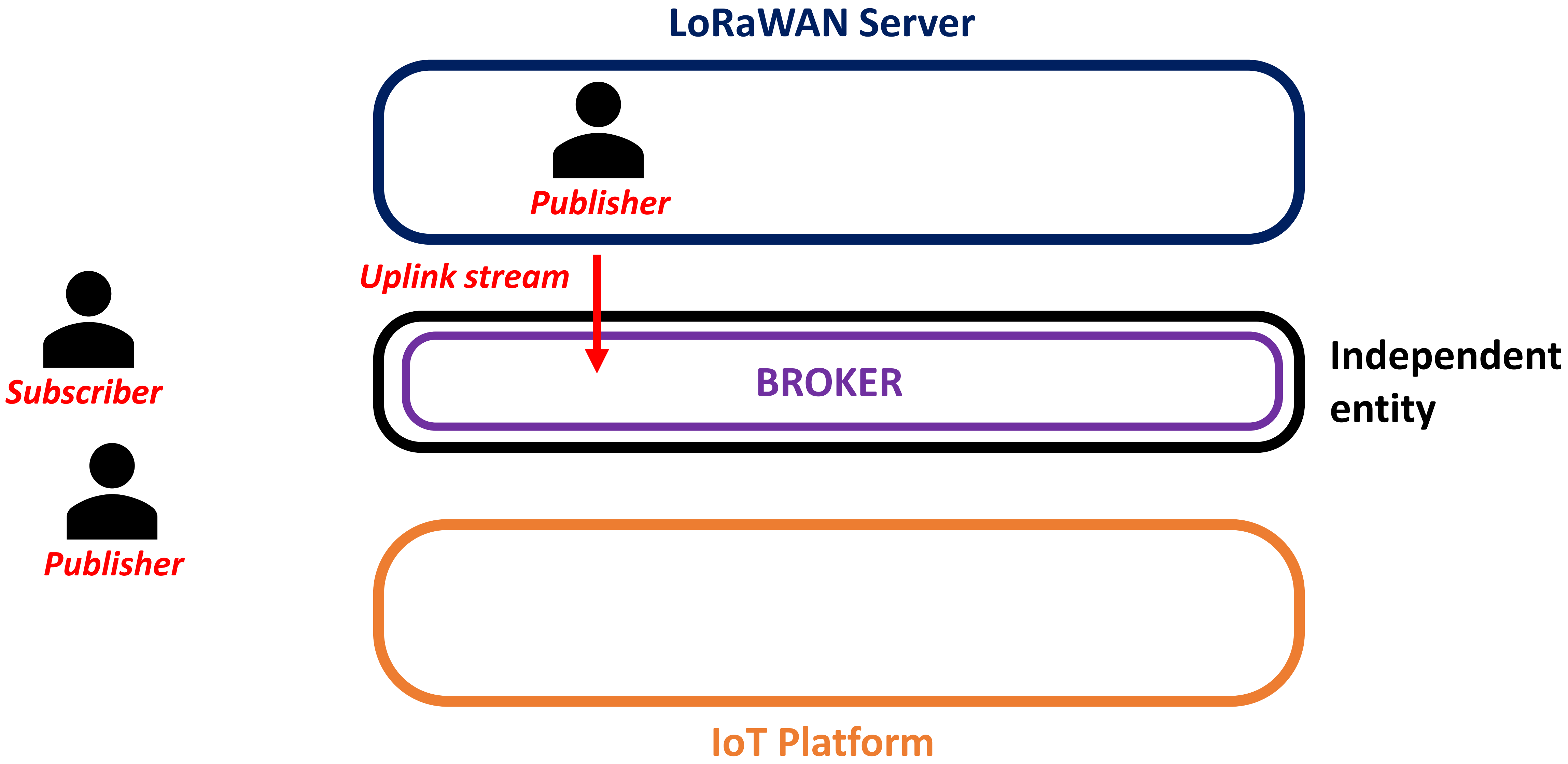
LoRaWAN Server

IoT Platform

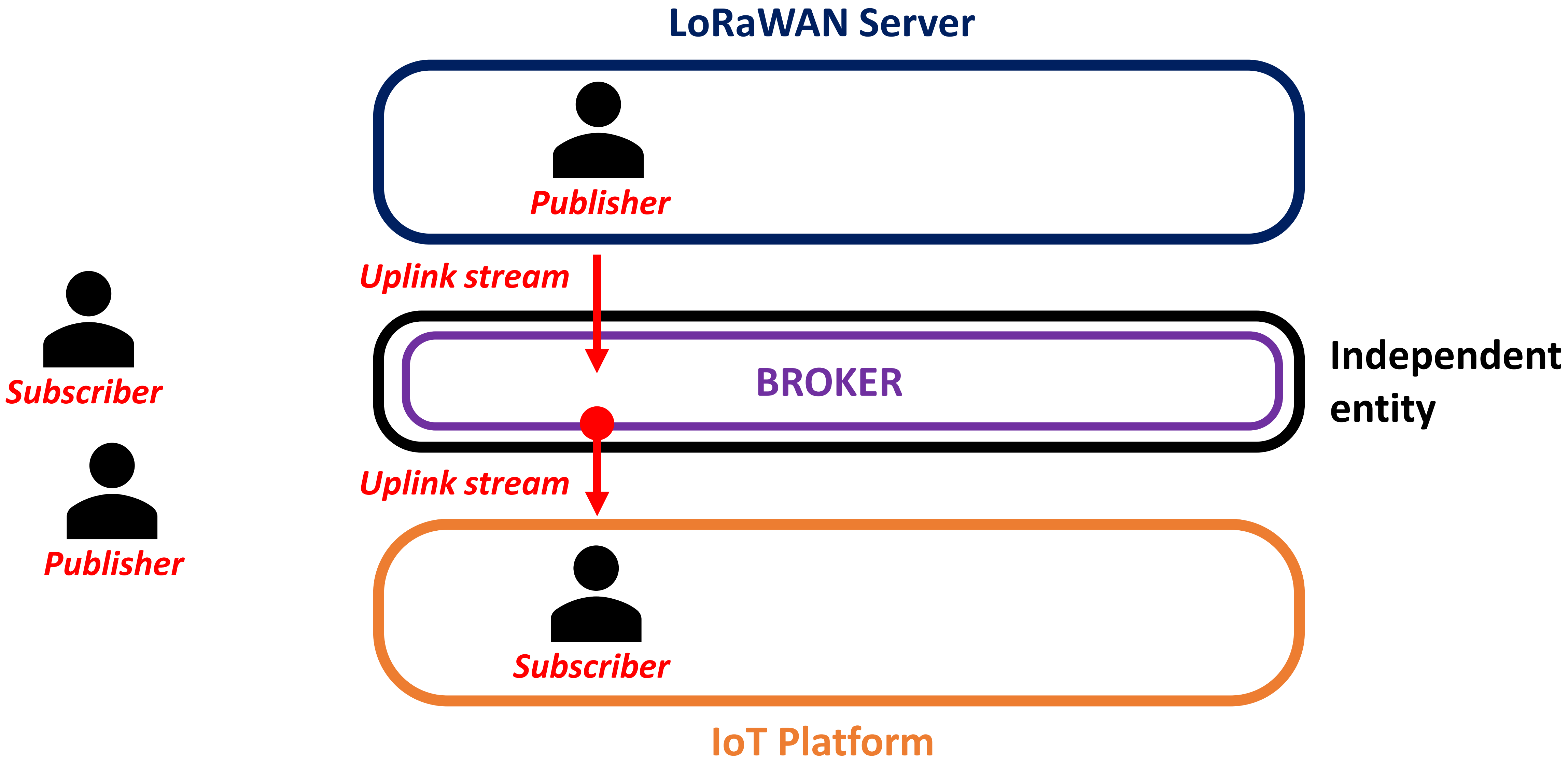
MQTT - LoRaWAN server as a MQTT client



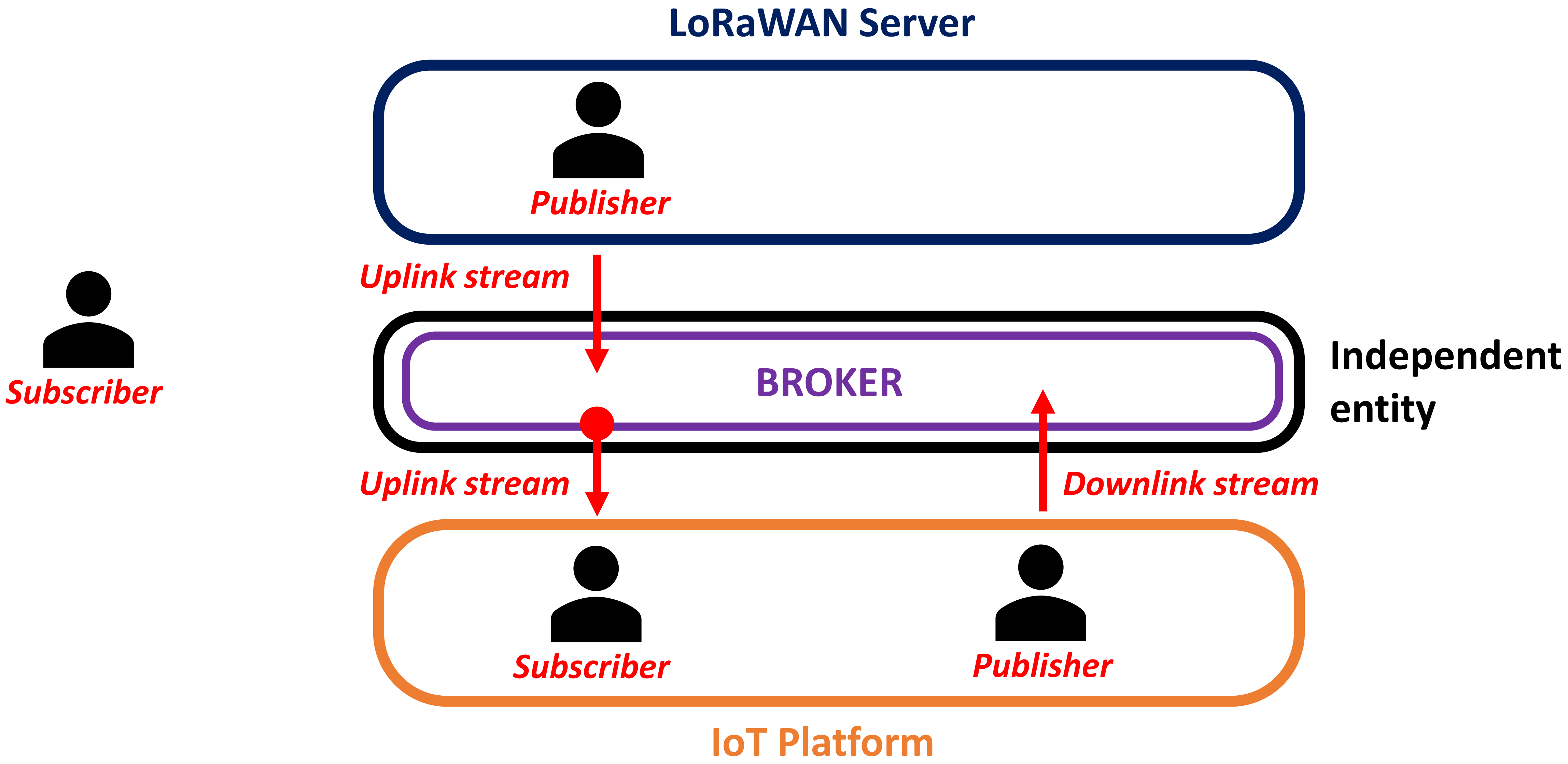
MQTT - LoRaWAN server as a MQTT client



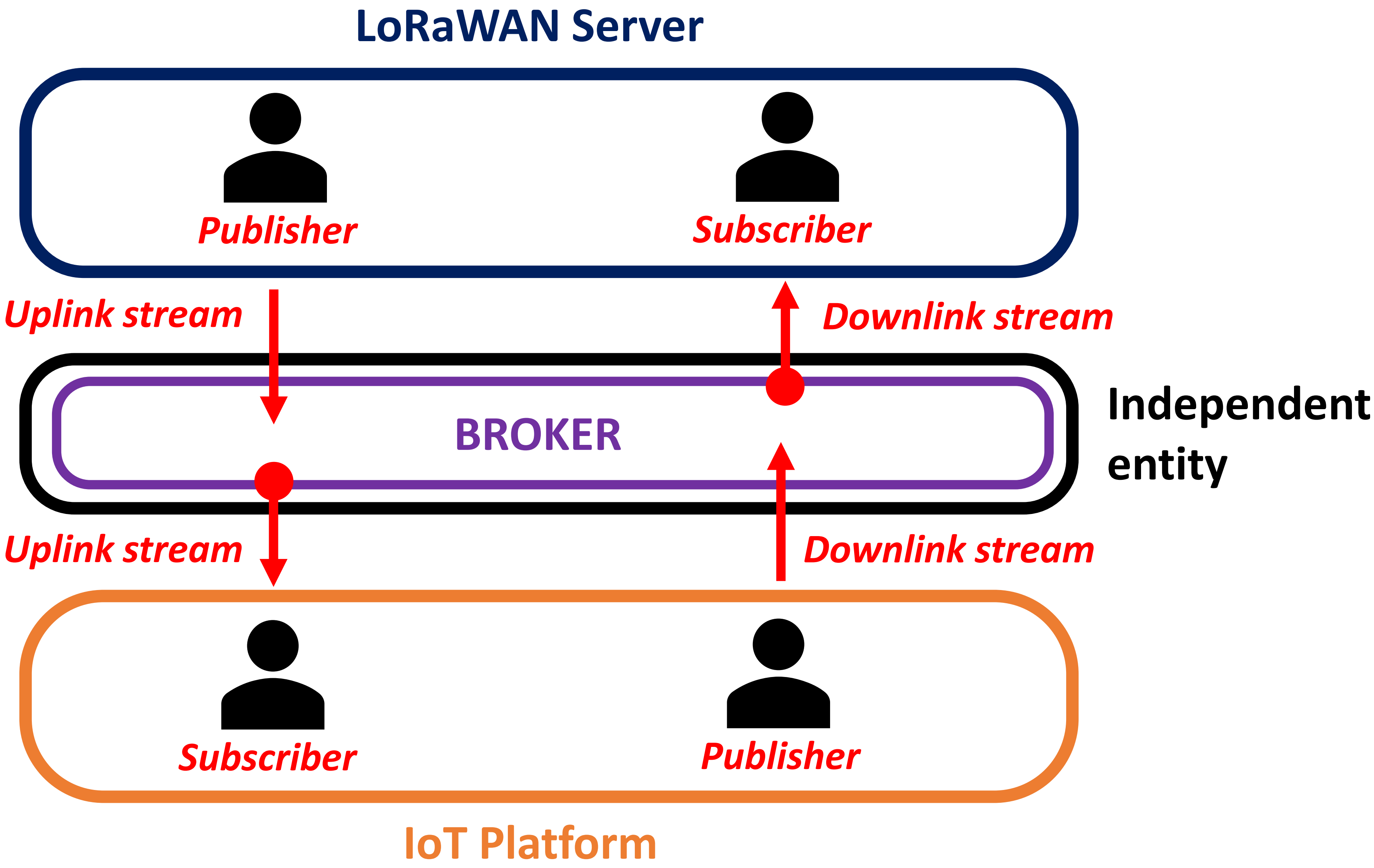
MQTT - LoRaWAN server as a MQTT client



MQTT - LoRaWAN server as a MQTT client



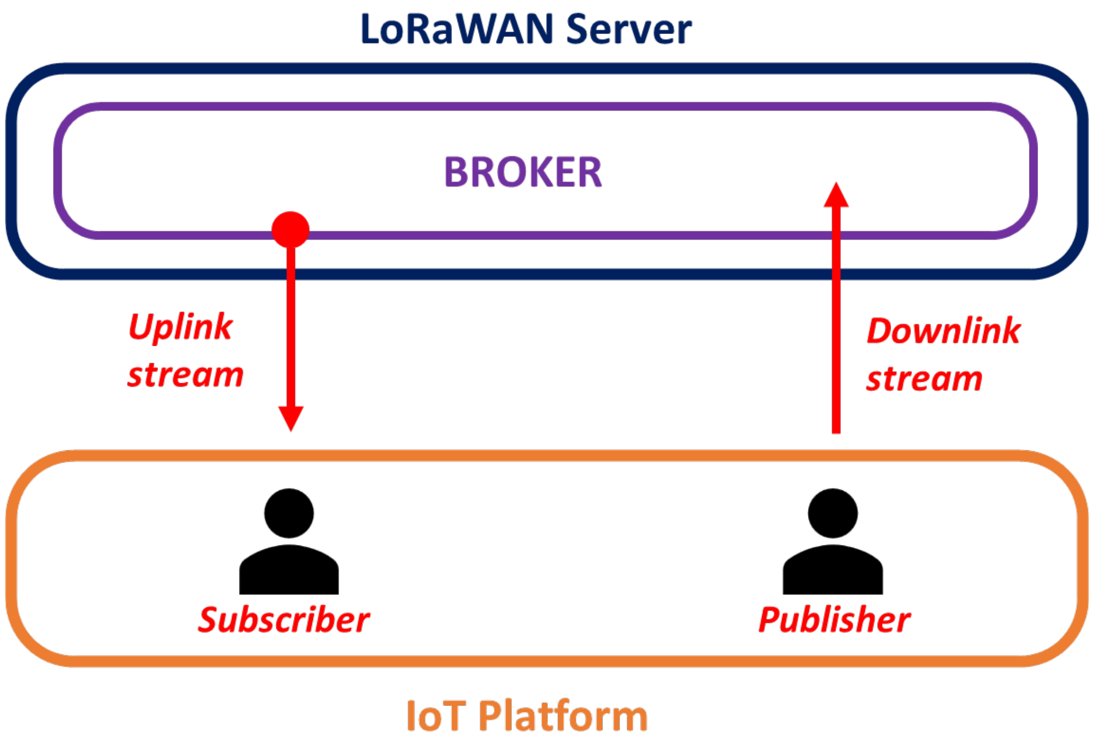
MQTT - LoRaWAN server as a MQTT client



MQTT - Demonstration with TTN

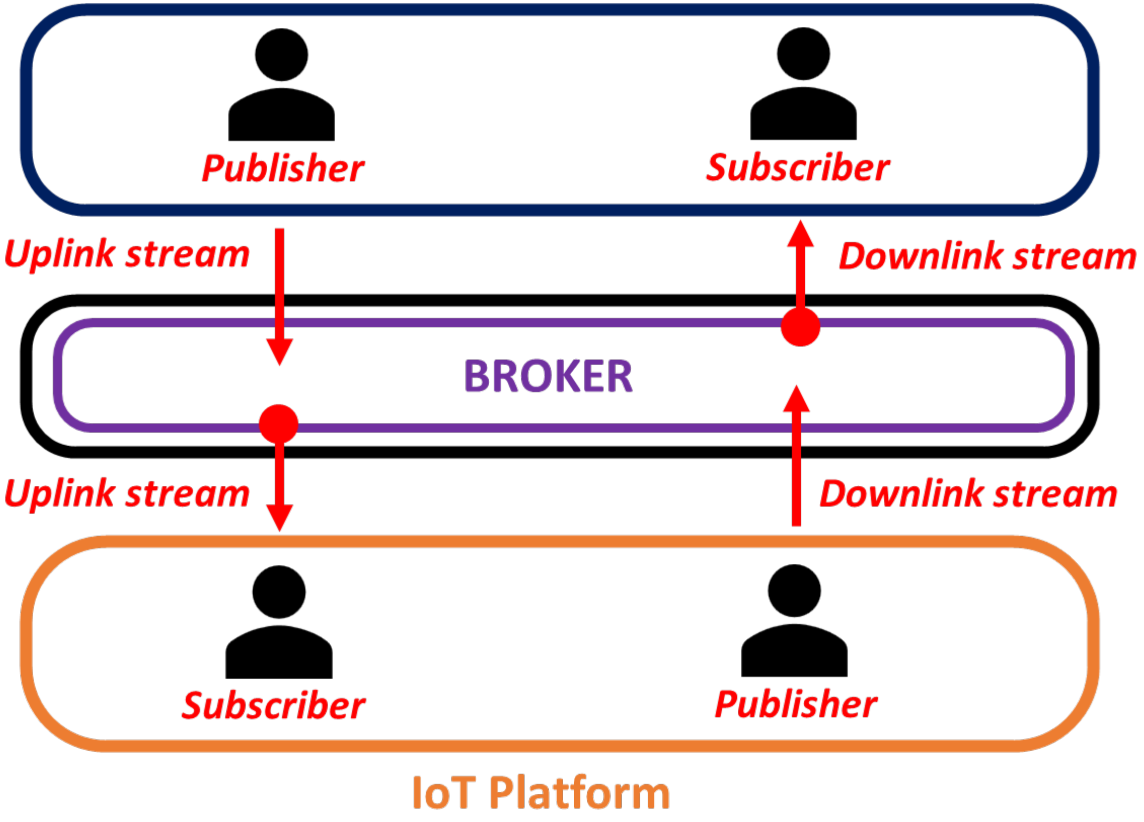
THE THINGS STACK Community Edition (TTN)

Architecture available on TTN



THE THINGS STACK

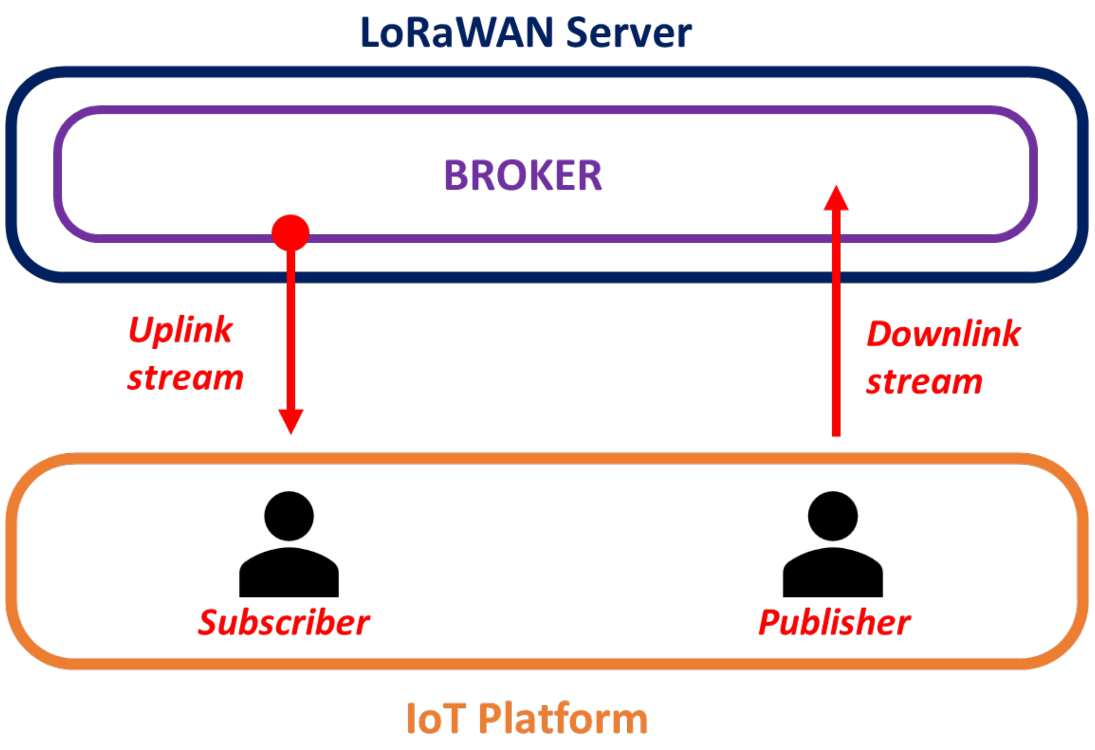
Architecture available on TTS server



MQTT - Demonstration with TTN

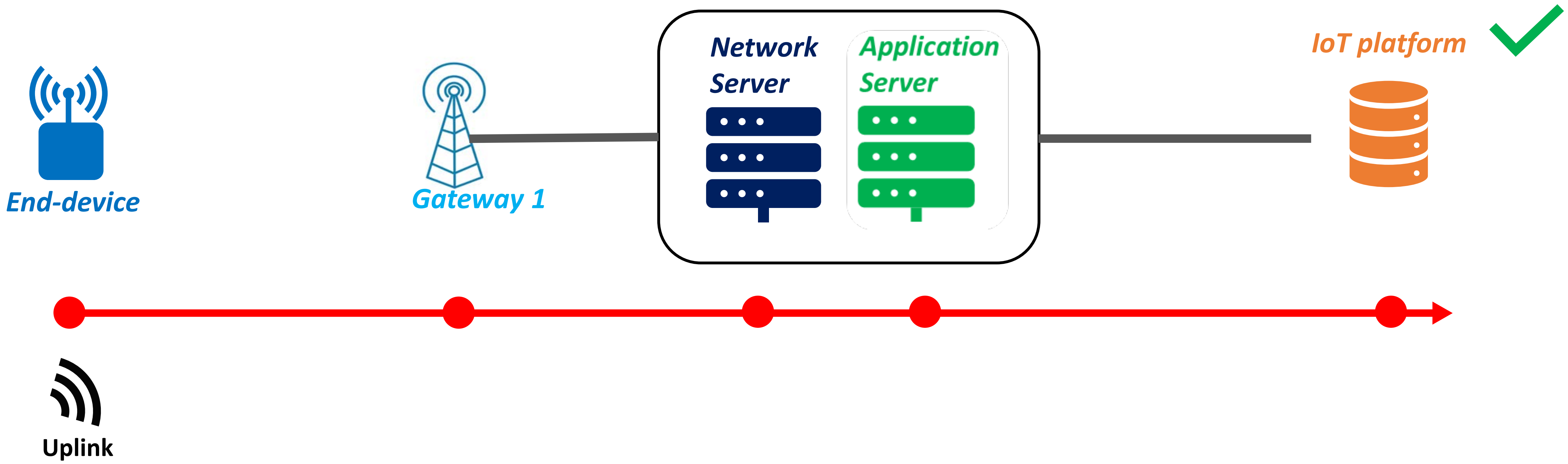
Our demonstration

THE THINGS STACK
Community Edition (TTN)



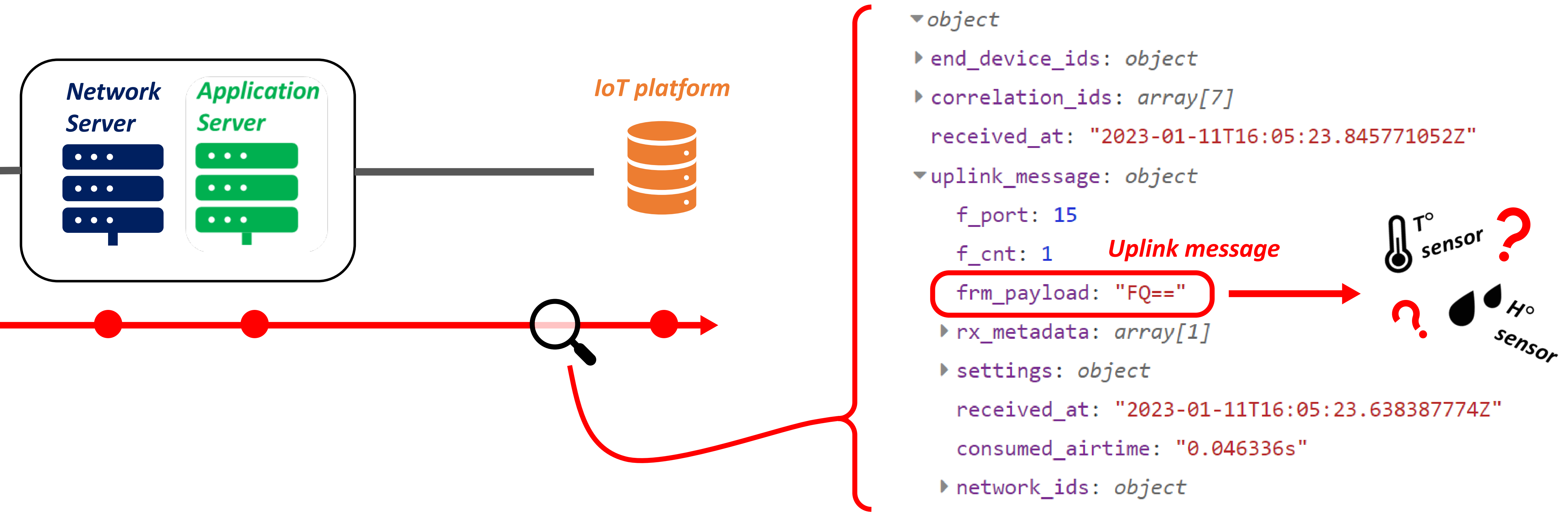
Payload decoder - Encoder

Uplink way



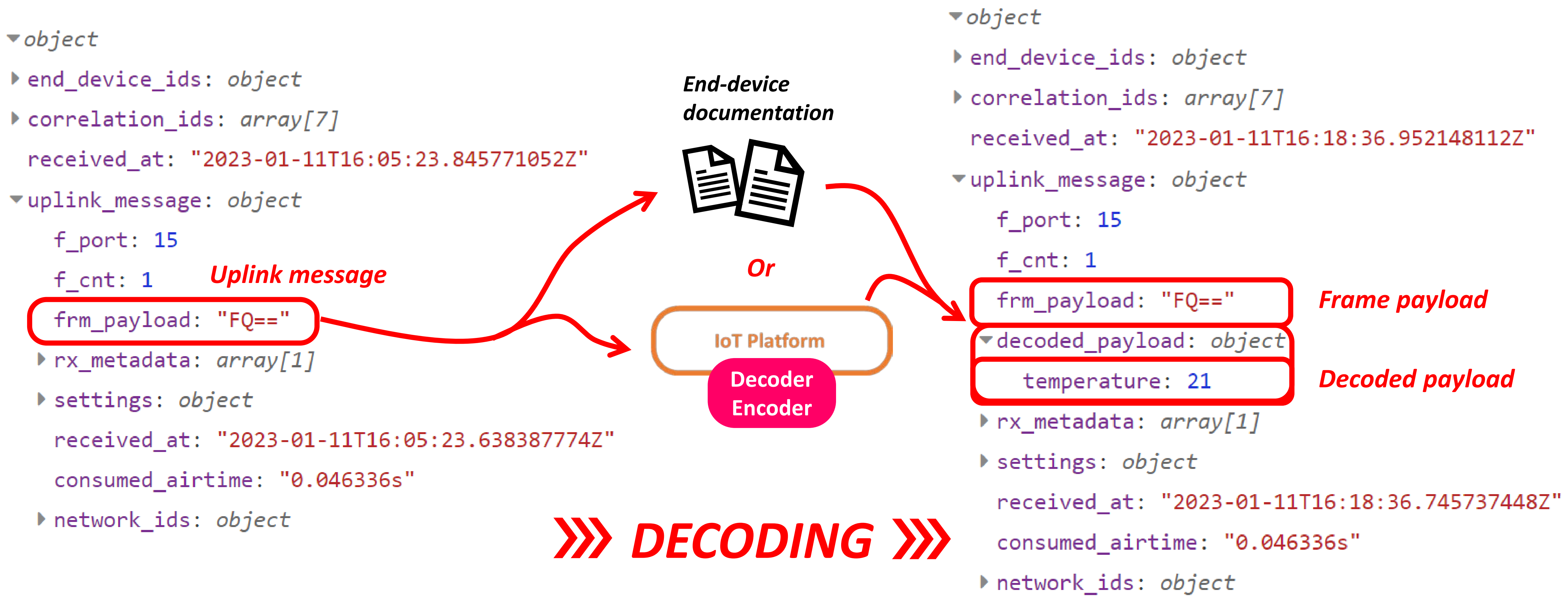
Payload decoder - Encoder

Uplink way



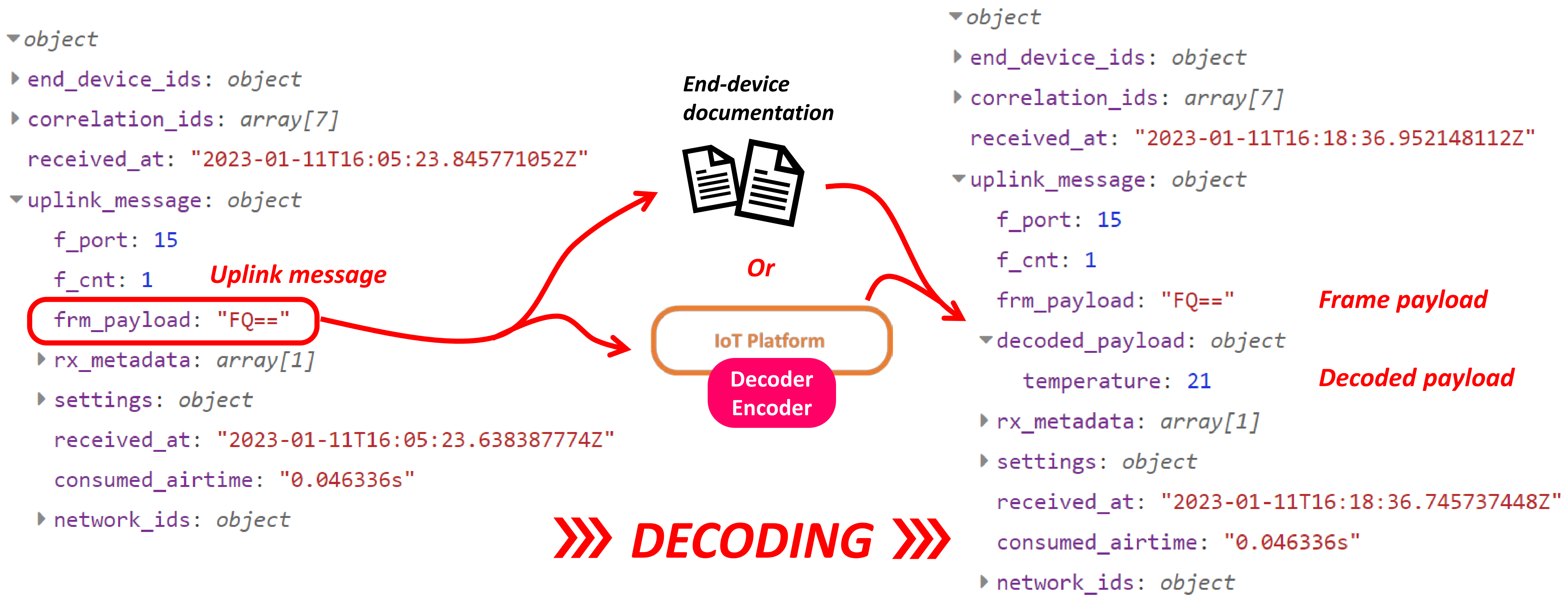
Payload decoder - Encoder

Uplink way



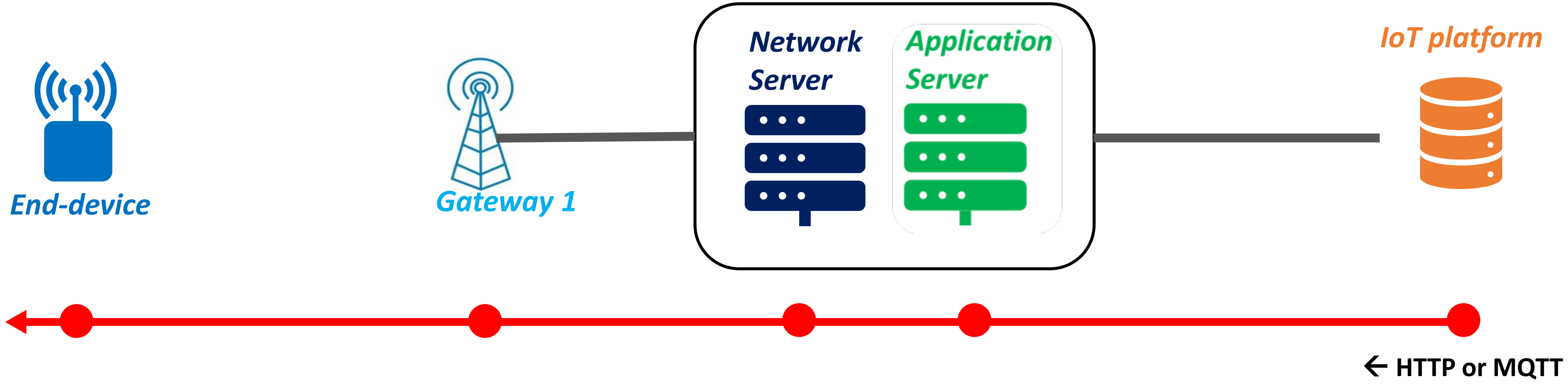
Payload decoder - Encoder

Uplink way



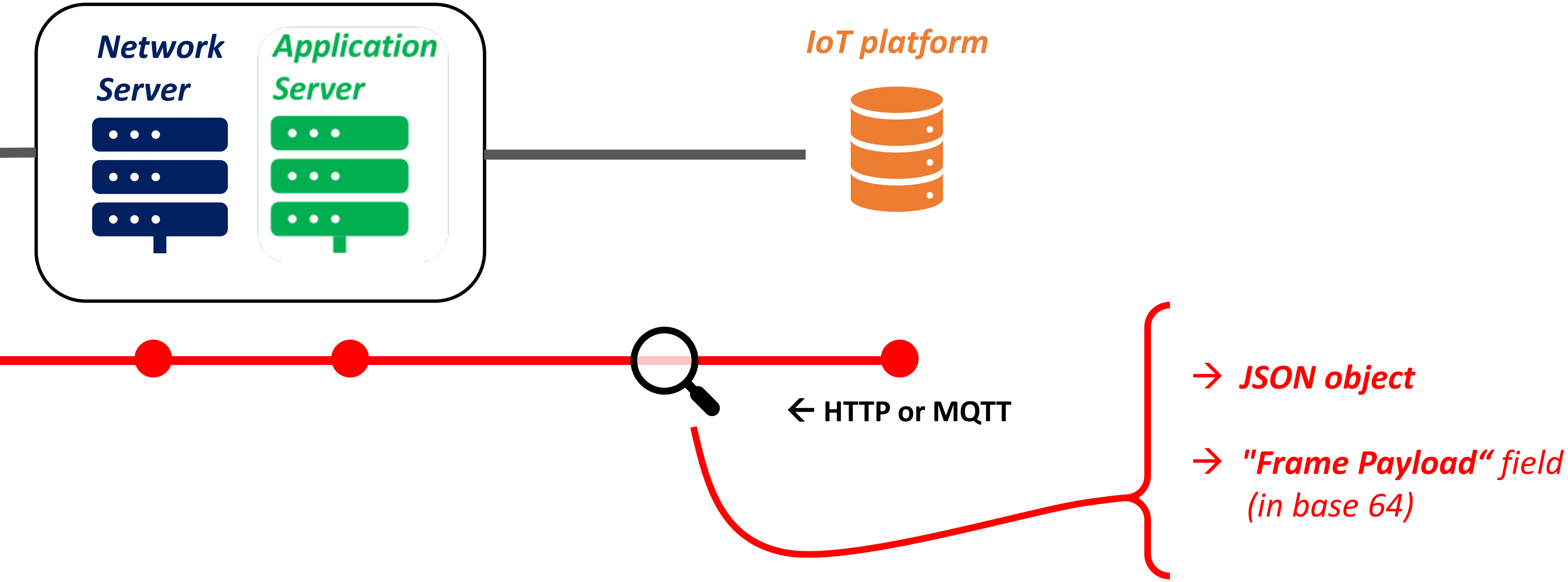
Payload decoder - Encoder

Downlink way



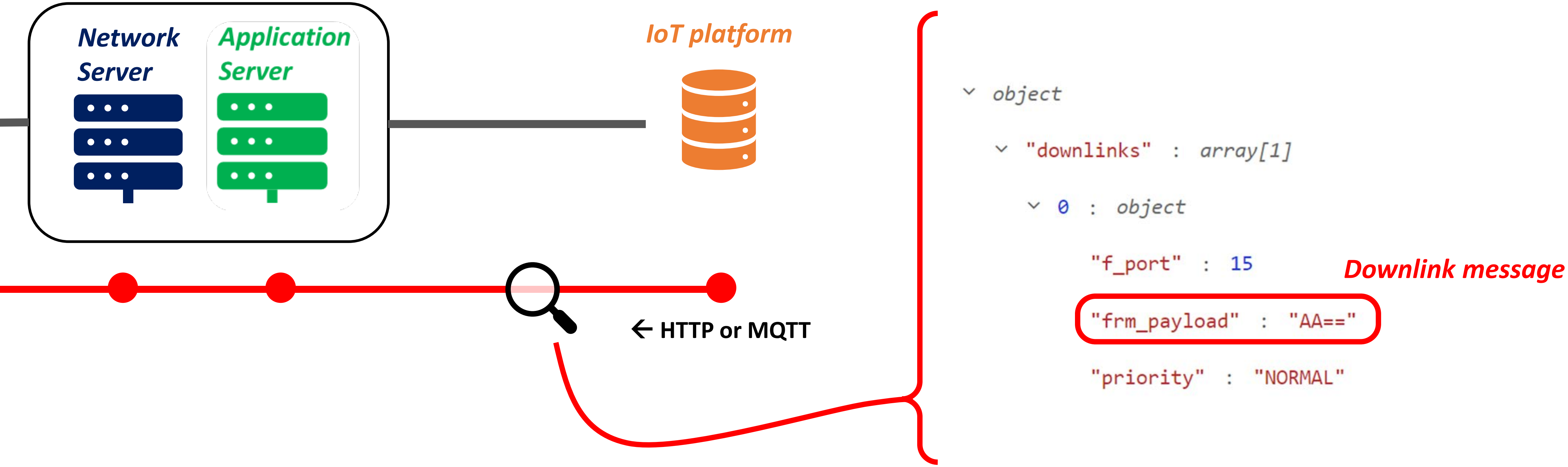
Payload decoder - Encoder

Downlink way



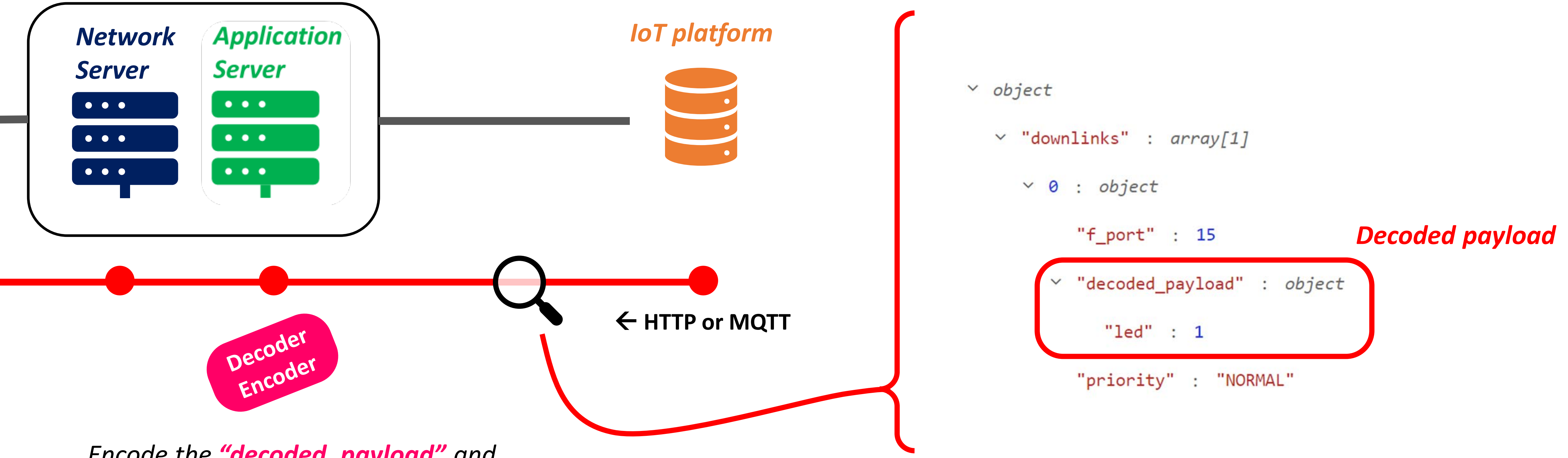
Payload decoder - Encoder

Downlink way



Payload decoder - Encoder

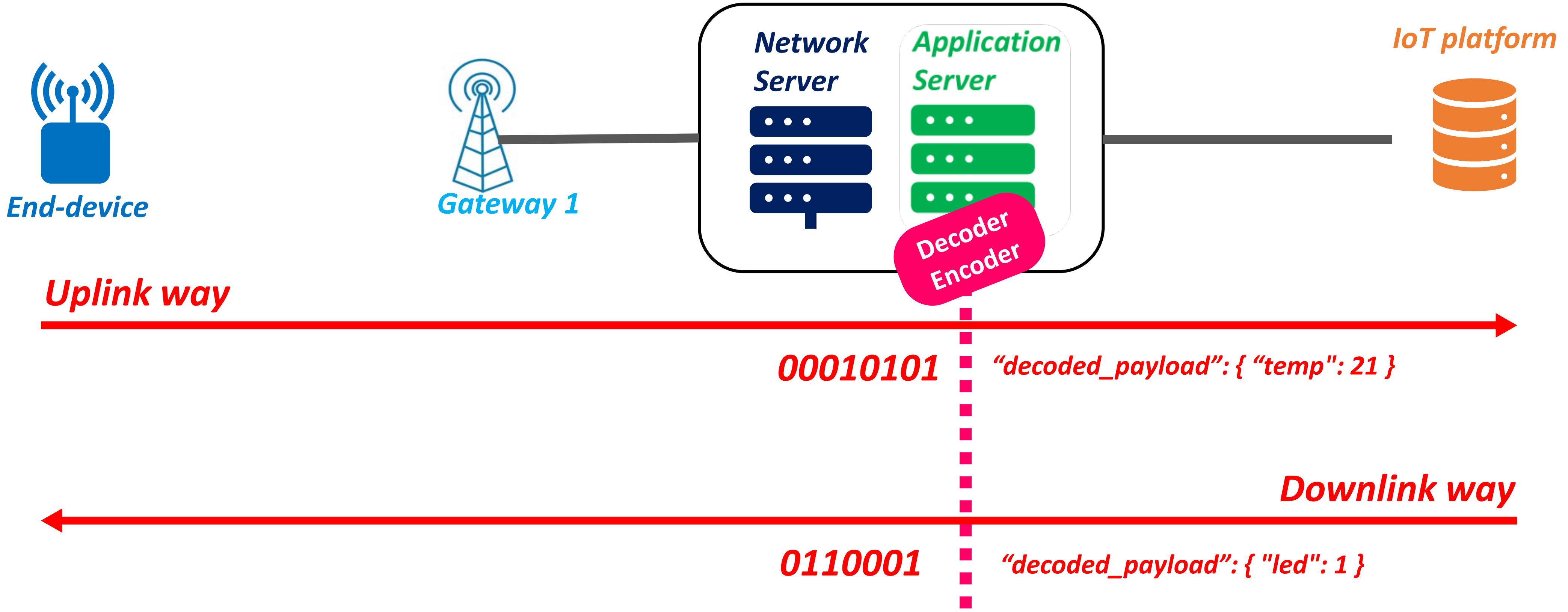
Downlink way



Encode the ***"decoded_payload"*** *and add a* ***"frm_payload"*** *field*

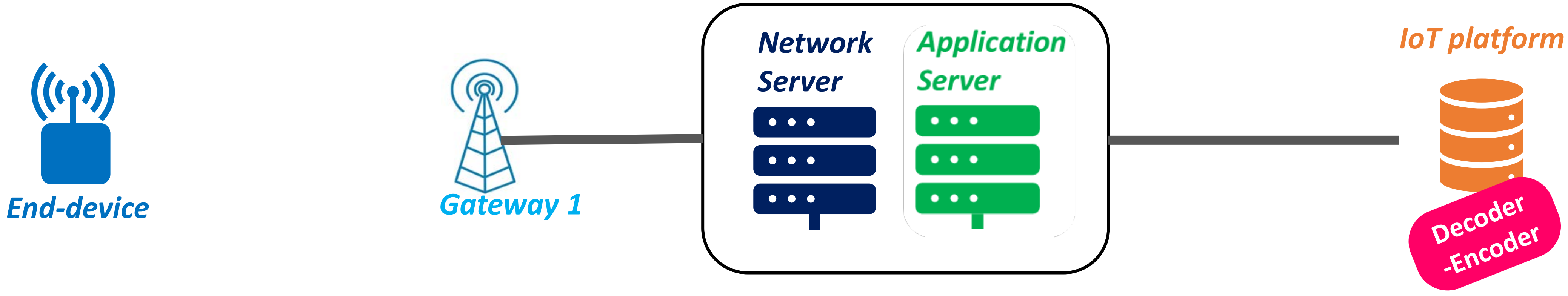
Payload decoder - Encoder

RECAP !



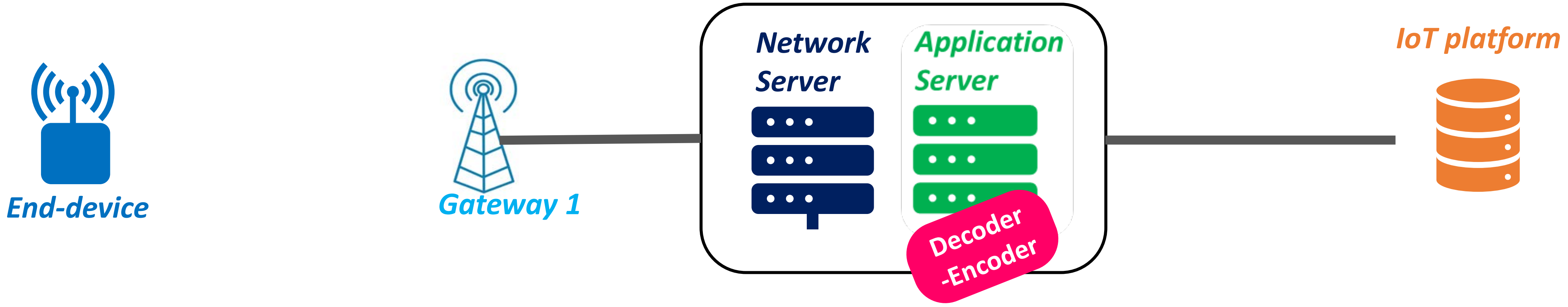
Payload decoder - Encoder

RECAP !



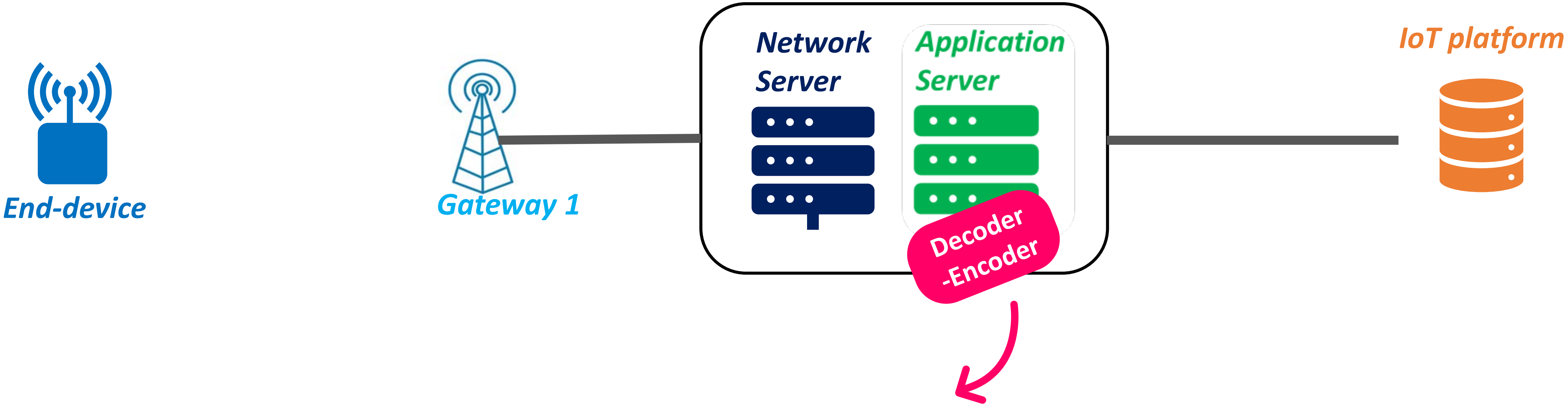
Payload decoder - Encoder

RECAP !



Option that will be presented !

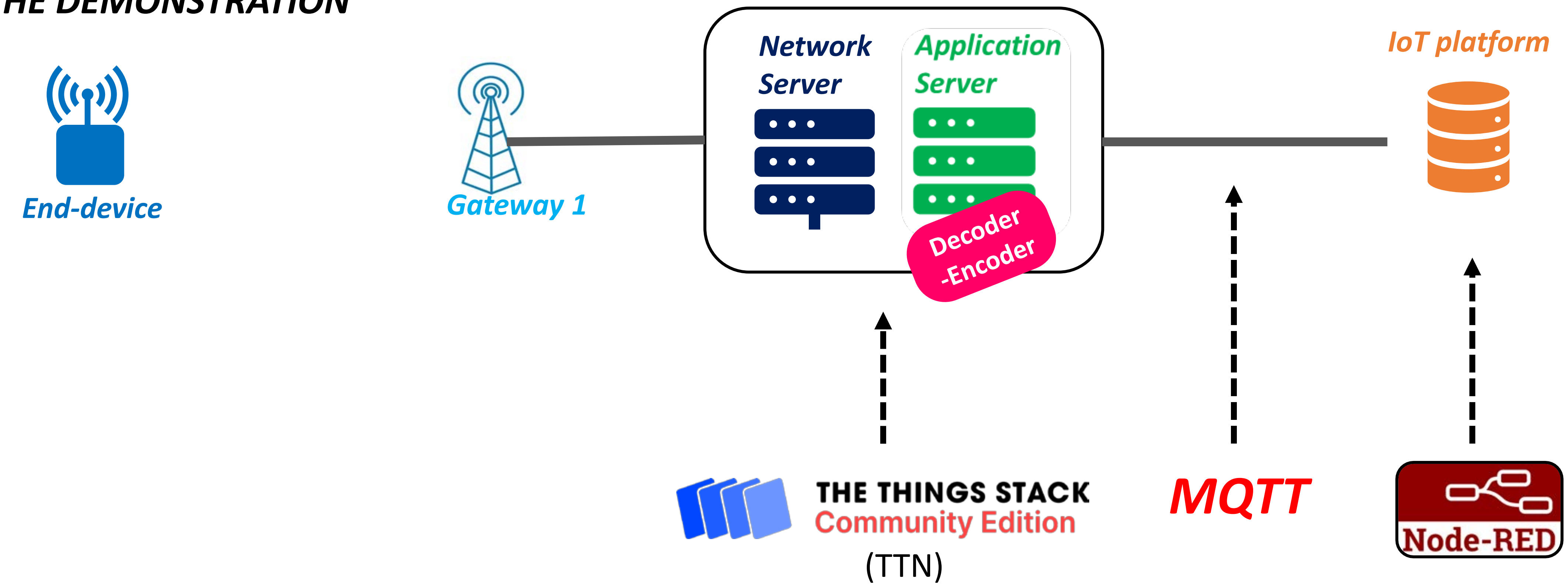
Payload decoder - Encoder



- *1 device = 1 decoder-encoder ?* *Yes*
- ***It exists predefined payload structure*** *(e.g.: Cayenne Low Power Payload)*
- LoRa Alliance® ***has standardized the decode - encode functions*** *(but not the payload structure)*

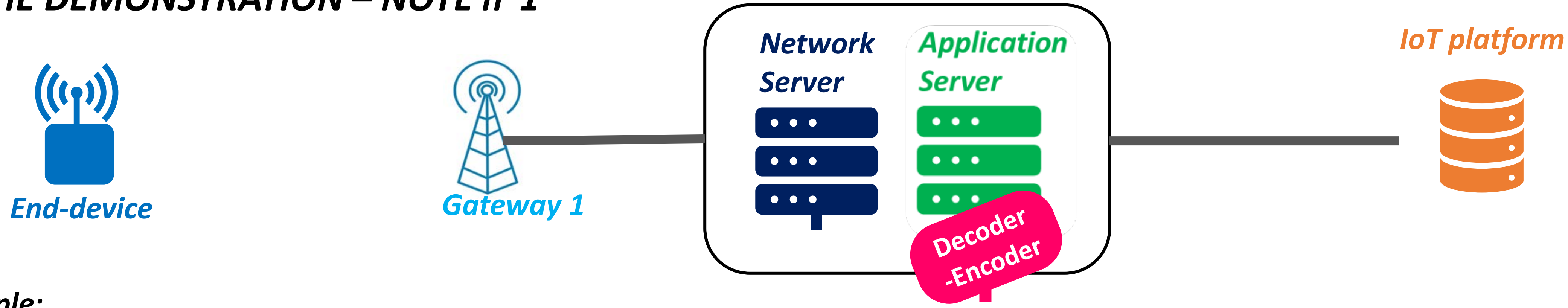
Payload decoder - Encoder

THE DEMONSTRATION



Payload decoder - Encoder

THE DEMONSTRATION – NOTE n°1



Example:
My device sends a temperature of ***21°C*** *port* ***15***

```
input = {
        "fPort" :  15,
        "bytes" :  [0x15],
}
```

Decode function

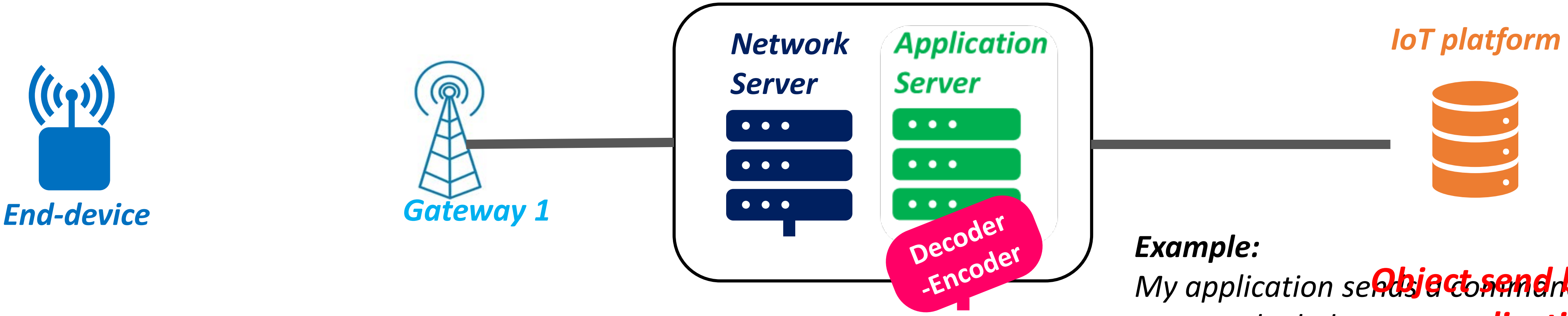
All the bytes received

Note 1.a : $[0x15]_{hex} = [21]_{dec}$

Note 1.b : Temperature = ***input.bytes[0]***

Payload decoder - Encoder

THE DEMONSTRATION – NOTE n°2



Example:

*My application sends a command on port **15** to* ***turn on** the led*

Object send by our application

Encode function

```
input = {
        "data" :  { "led" : 1},
        "fport" :  15,
}
```

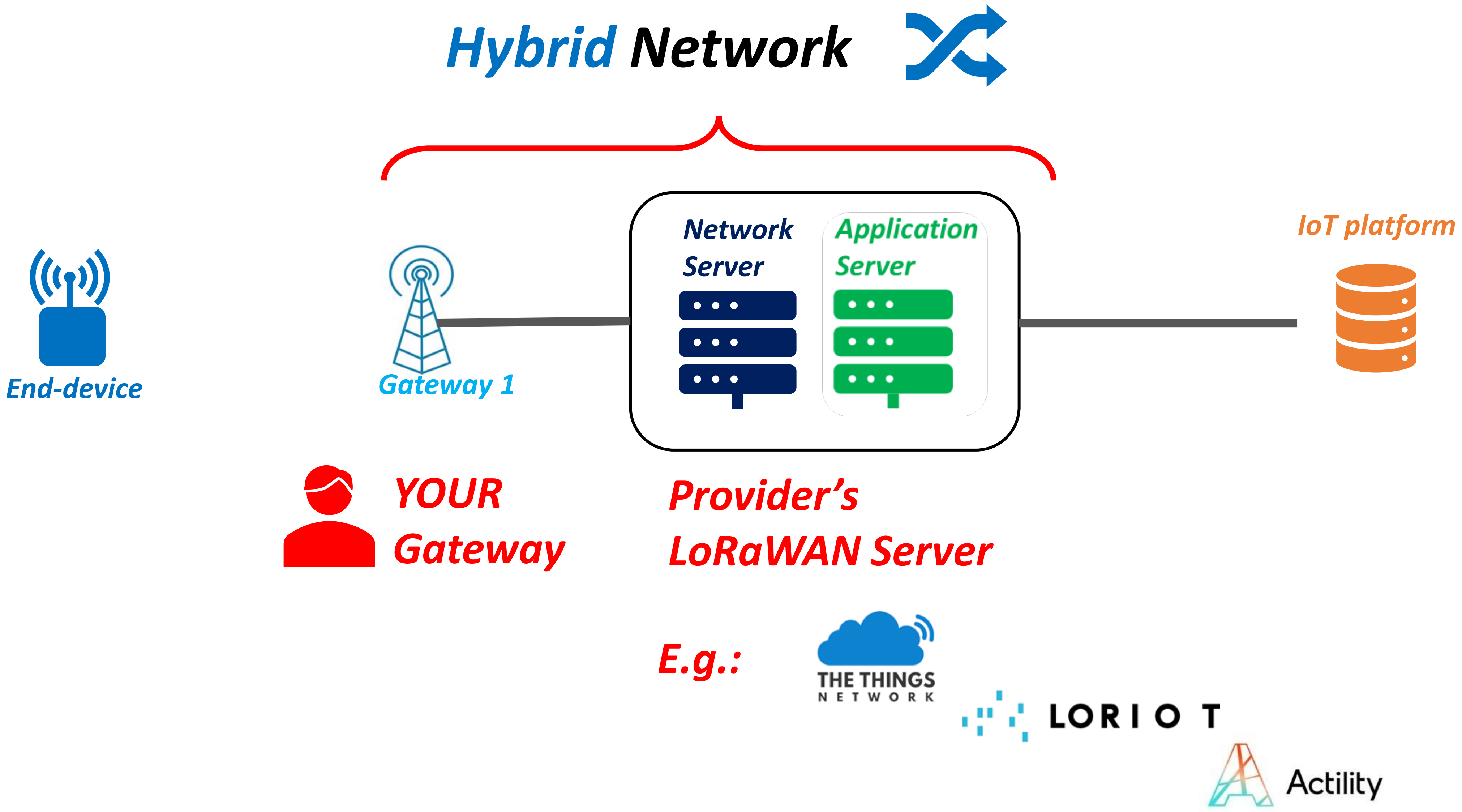
LoRaWAN fPort

*Note 2.a : bytes = **input.data.led***

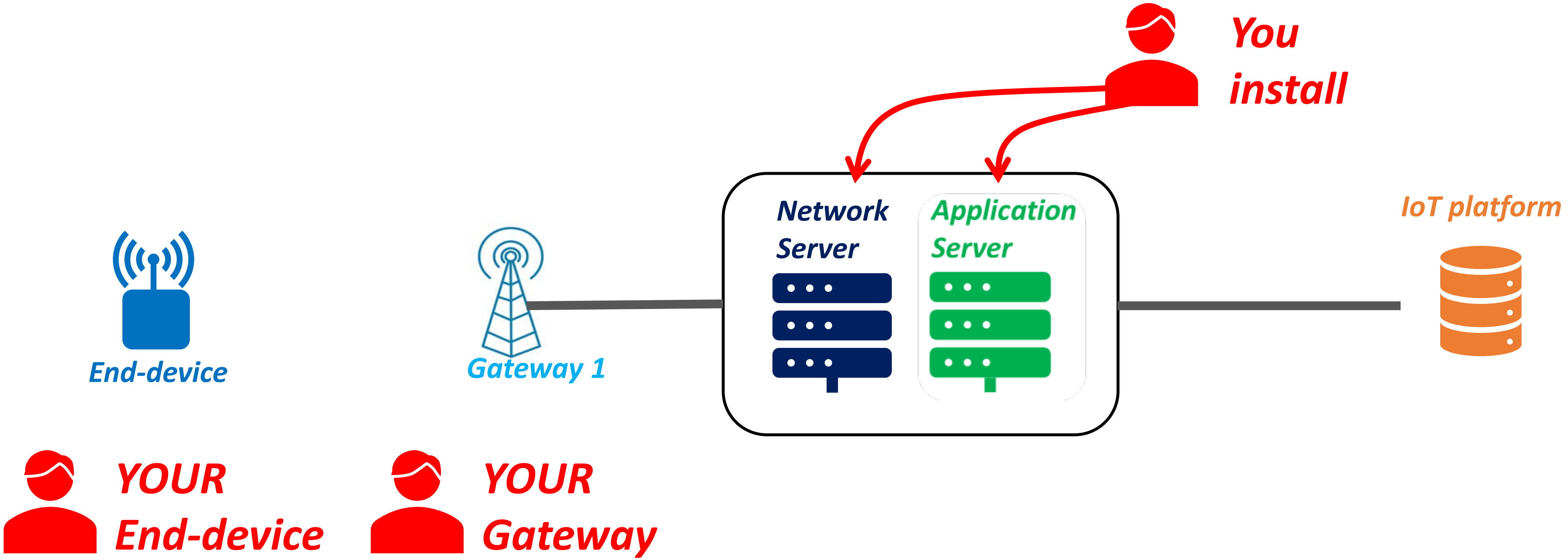
Setting up our own LoRaWAN Server

- ✓ …
- ✓ …
- ✓ …

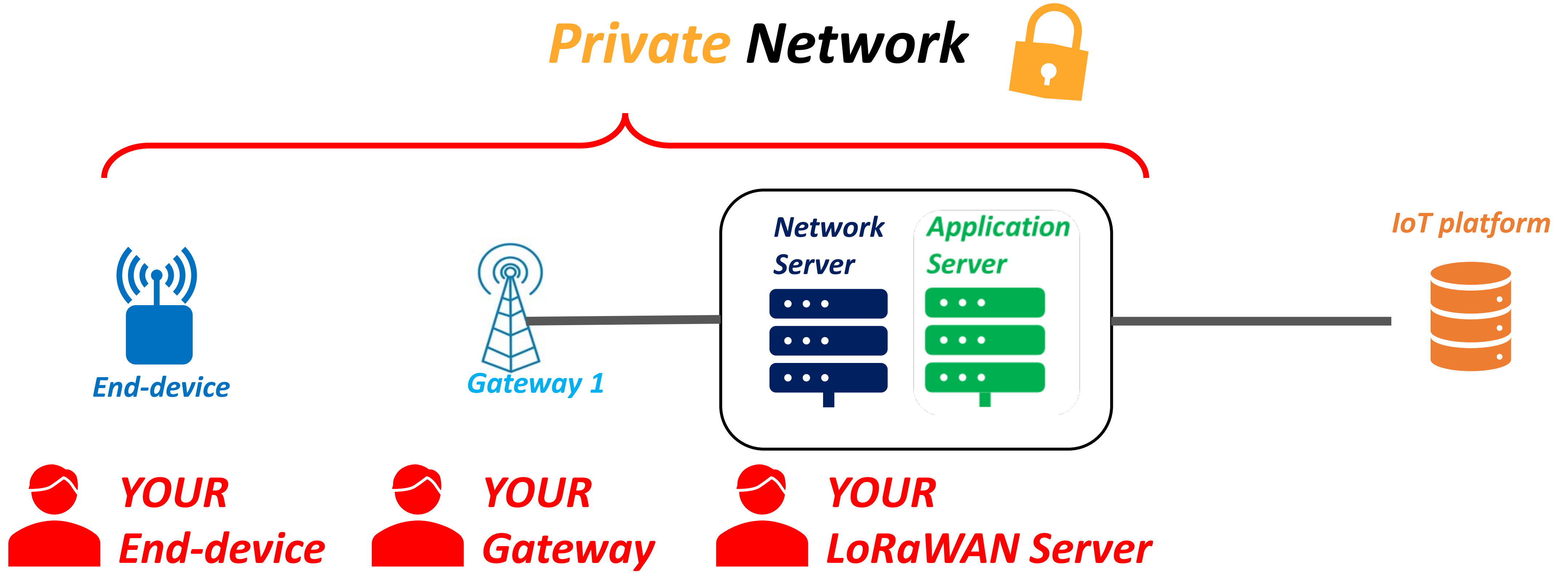
Getting Started with private network



Getting Started with private network



Getting Started with private network



Getting Started with private network

Private Network

A good choice

A confidential network with no data transferred on public infrastructure

YOU administrate YOUR servers

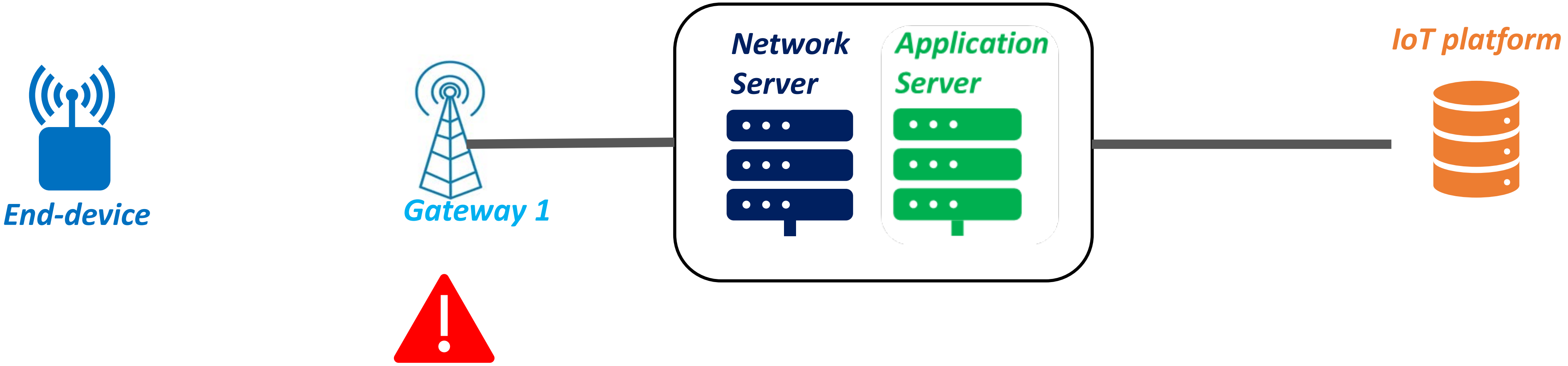
- ***You can scale its capacity***
- ***You can handle its availability***

No subscription fees to a provider

Getting Started with private network

Private ***Network***

Configuration to take care about…



Gateway parameters change !

Getting Started with private network

Private ***Network***

LoRaWAN Servers…

ChirpStack

Orne Brocaar

THE THINGS STACK

THE THINGS INDUSTRIES

Used by

THE THINGS NETWORK

Free and open source

Strong community

Commercial support

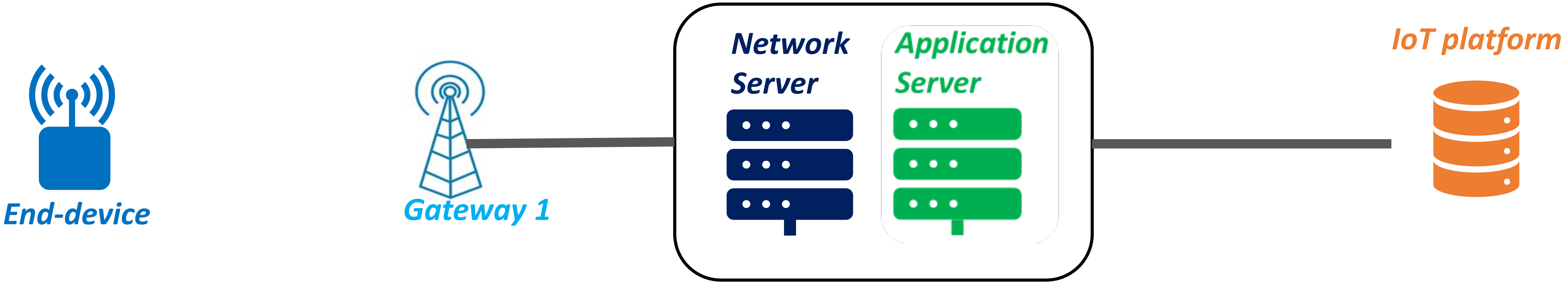
Getting Started with private network

Private ***Network***

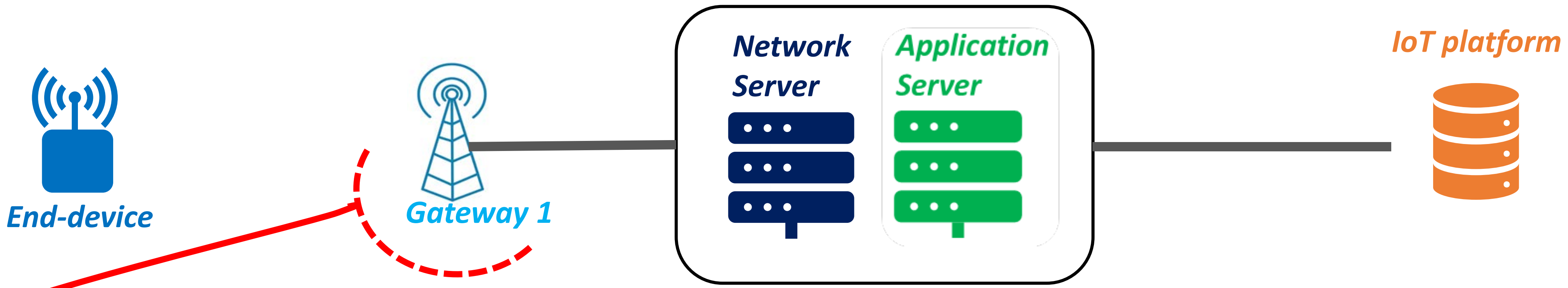
Before starting the installation…

- ***Check the gateway configuration*** *(LoRaWAN Server compatibility, etc.)*
- ***Choose the right solution to install the LoRaWAN Server***

The packet forwarder



The packet forwarder



Gateway configuration

- *Set the targeted Network Server*
- *The protocol used*

Packet forwarder

- UDP SEMTECH packet forwarder
- LoRa Basic Station
- LRR (Long Range Relay)
- LORIOT packet forwarder
- …

LoRaWAN *doesn't specify one protocol*

The packet forwarder

Packet forwarder

- UDP SEMTECH packet forwarder

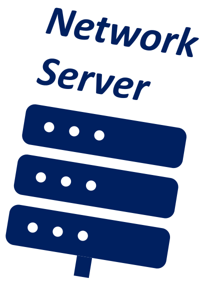
- LoRa Basic Station

- LRR (Long Range Relay)

- LORIOT packet forwarder

- …

Compatibility
with the Network Server



Most Gateways support several packet forwarders

The packet forwarder

ChirpStack

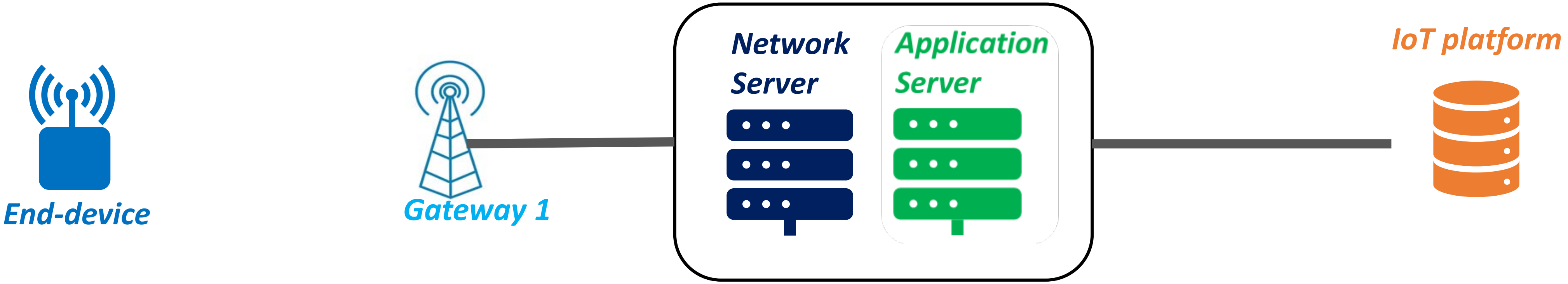
- UDP SEMTECH packet forwarder
- LoRa Basic Station

THE THINGS STACK

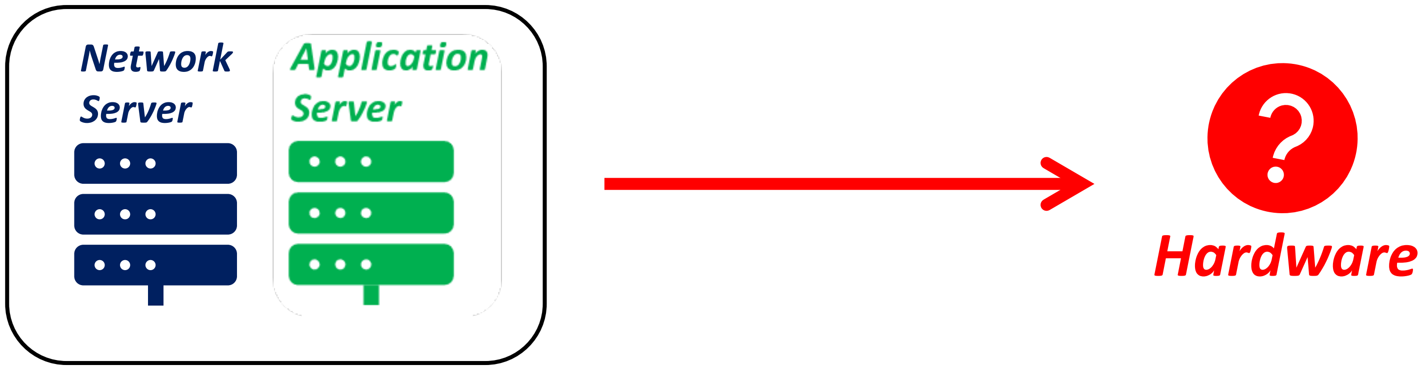
- UDP SEMTECH packet forwarder
- LoRa Basic Station

We will use the UDP SEMTECH packet forwarder

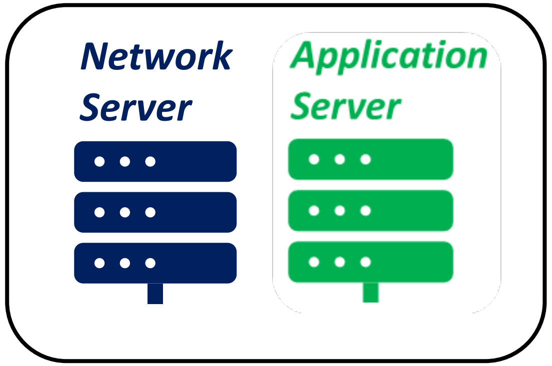
Installation options



Installation options



Installation options

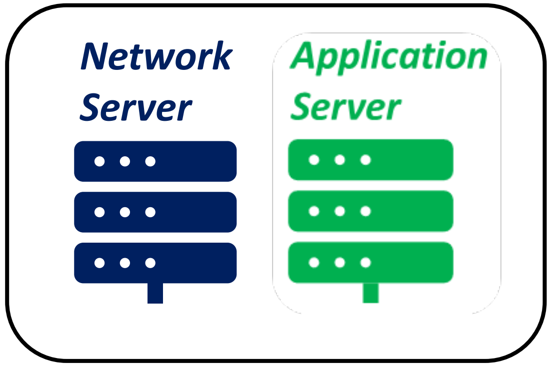


Hardware

It depends on the type of <u>project</u> !

- ***Proof of Concept***
- ***Infrastructure with a guarantee of availability***

Installation options



Hardware

It depends on the type of __project__ !

- *Proof of Concept*
- *Infrastructure with a guarantee of availability*

__Our case :__

- ***A small infrastructure***
- ***One condition: the system must support Docker containers***

Installation options

1 *Our own PC*

Good solution for testing

Operating System can be Windows, Linux or Mac OS with Docker

2 *Our own PC*
With a virtual machine

With a software such as VirtualBox or VMware

Same hardware BUT the system is independent of the environment

3 *Standalone machine*

The services run permanently and autonomously

Several hardware available :
- PC, …
- Rapspberry Pi

4 *Server on a cloud provider*

For this course, the cloud service only costs 4€/month

Increasing the server capabilities is easy

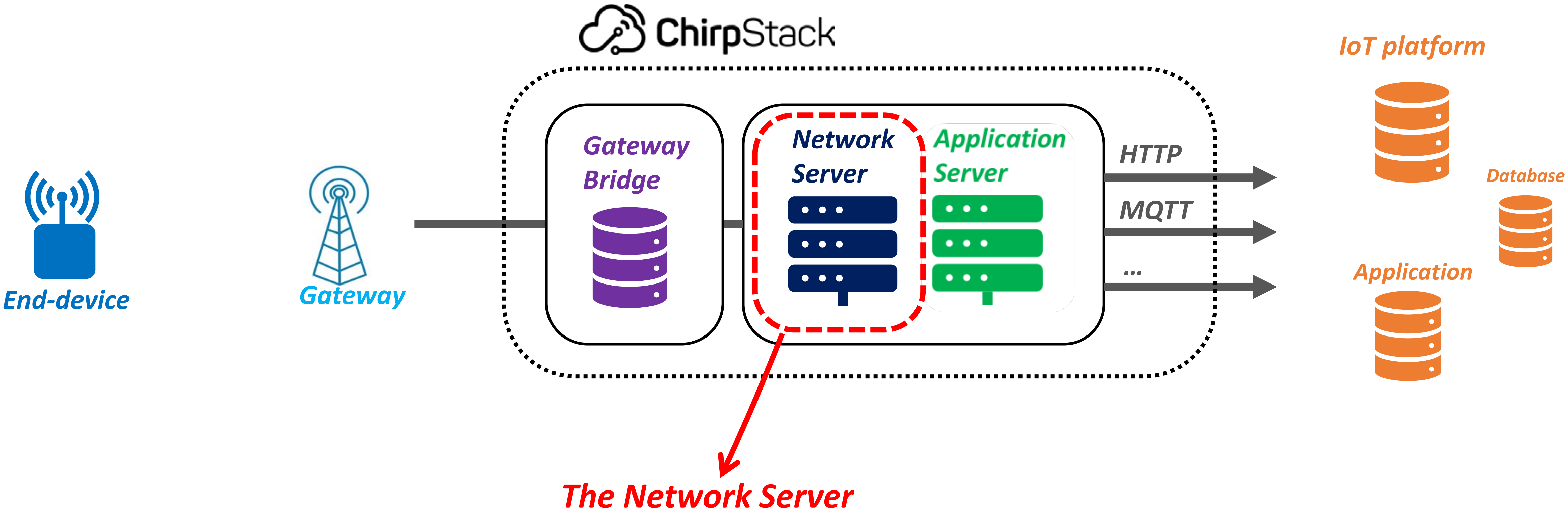
Best choice for scalability and availability

Chirpstack - Presentation

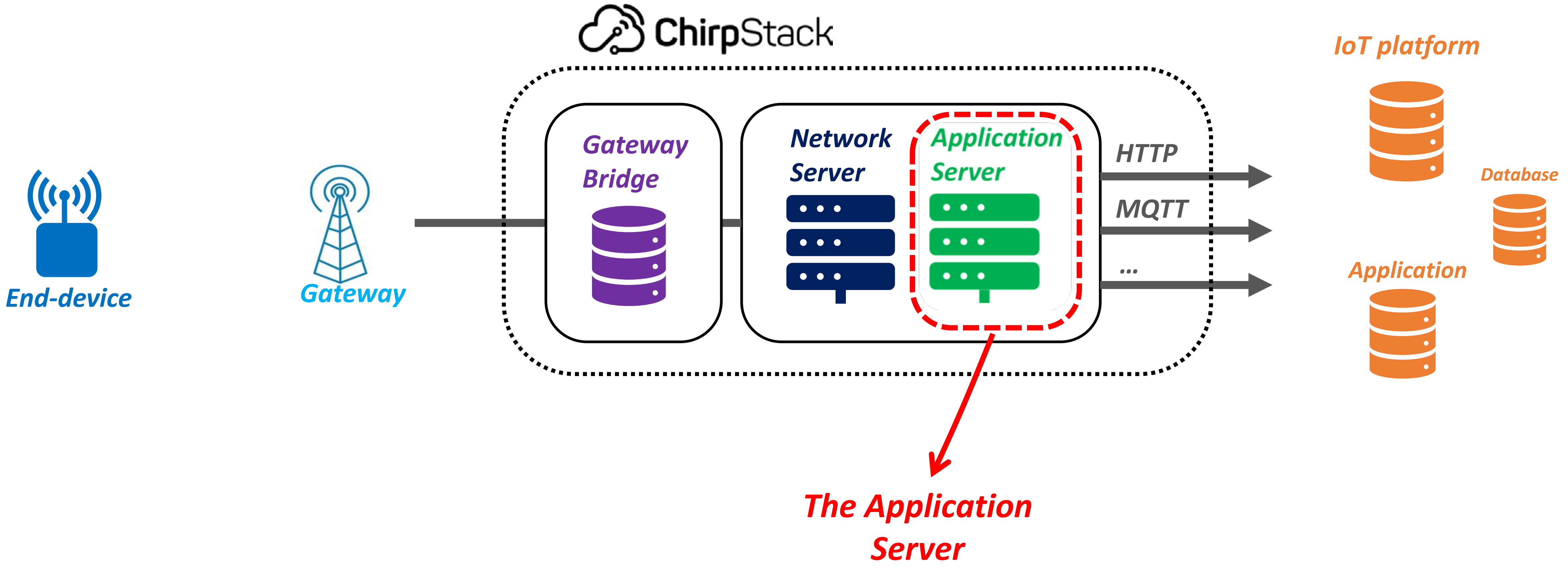
ChirpStack

- Free and open source
- Supports all regions
- Supports all end-devices classes
- Supports latest LoRaWAN specification

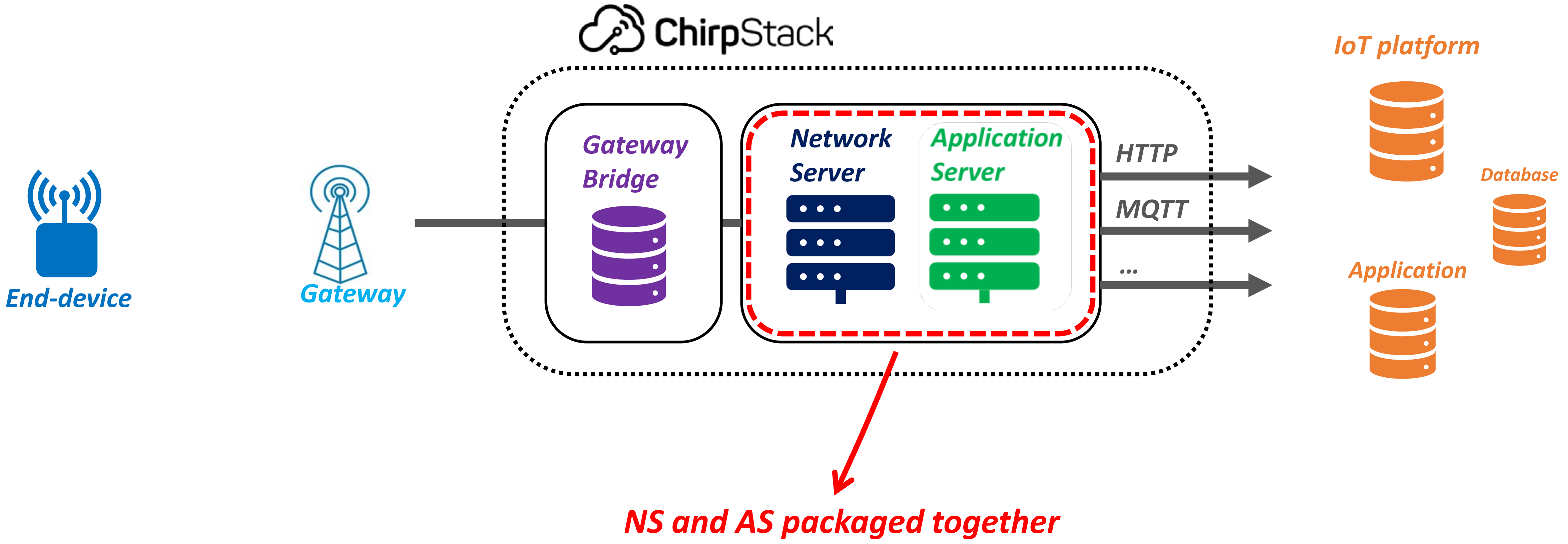
Chirpstack - Presentation



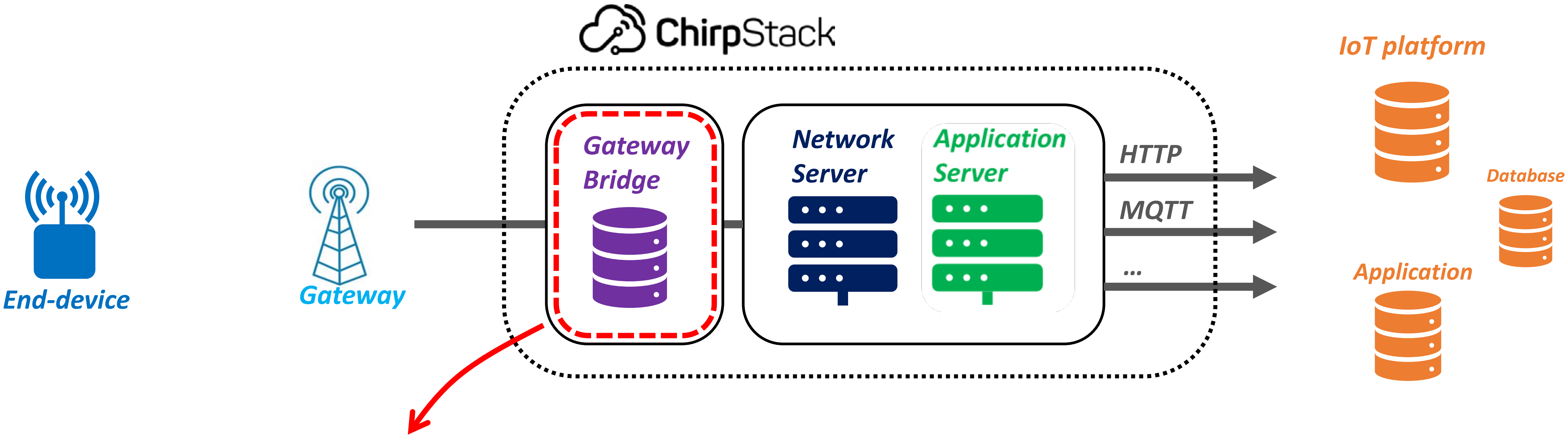
Chirpstack - Presentation



Chirpstack - Presentation



Chirpstack - Presentation



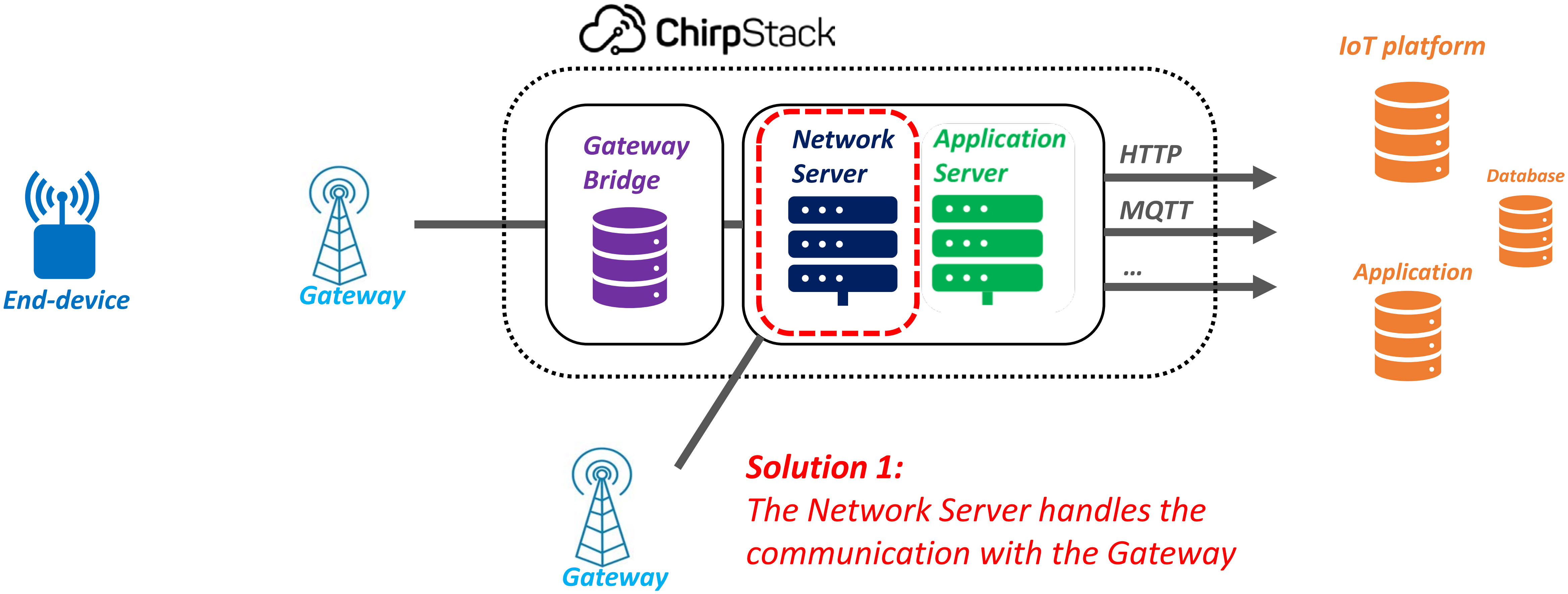
<u>The Gateway Bridge</u>

→ *It is not a LoRaWAN service*

→ *To simplify the message processing*

→ *Must know all the protocols* *(UDP Semtech Packet Forwarder, Basic Station, etc.)*

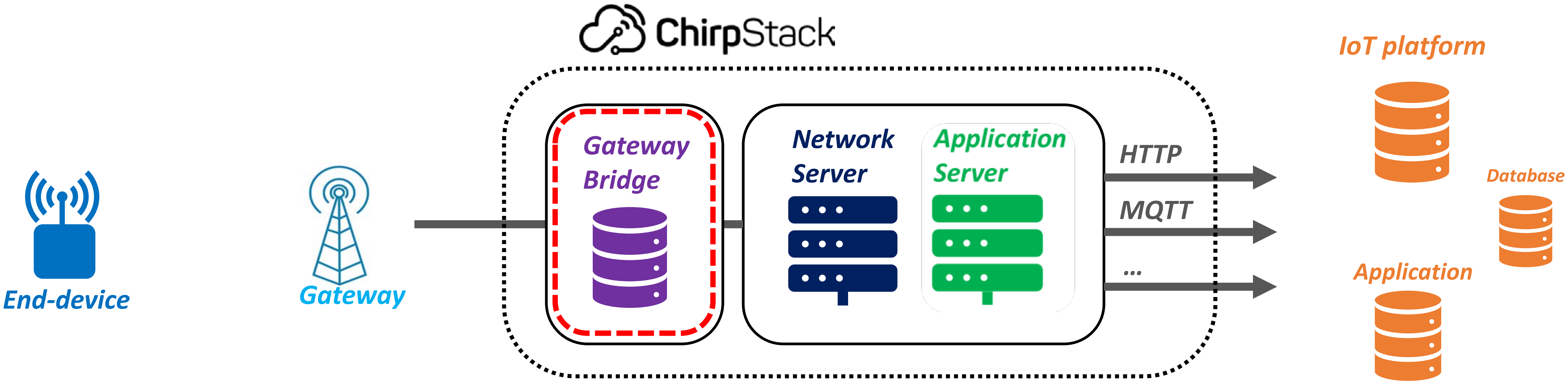
Chirpstack - Presentation



Solution 1:

The Network Server handles the communication with the Gateway

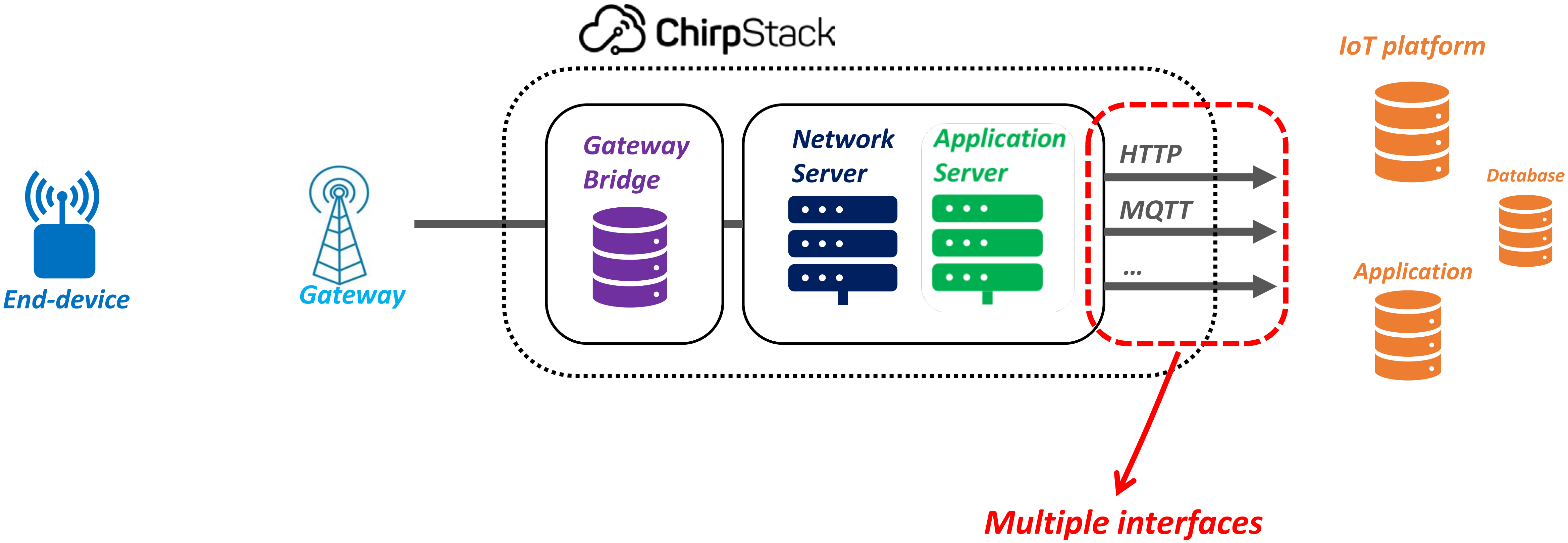
Chirpstack - Presentation



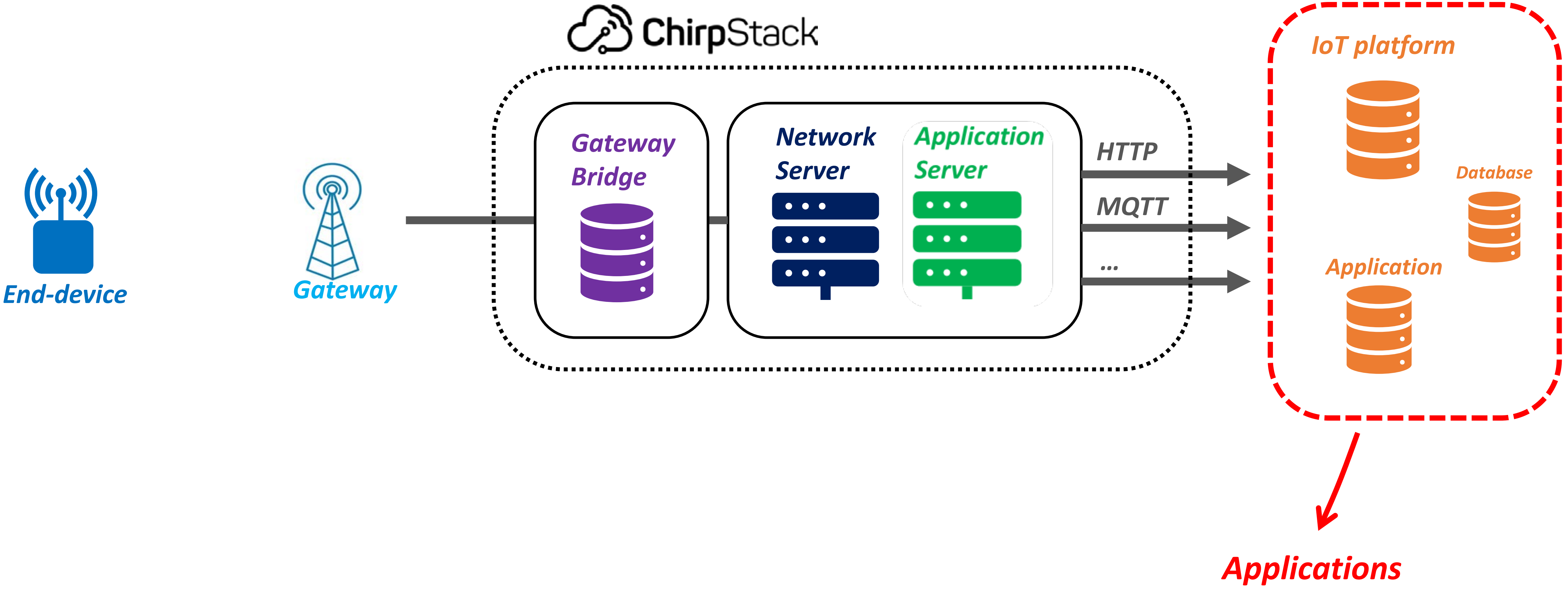
Solution 2:
The Gateway Bridge handles the communication with the Gateway

 ***Interesting for the implementation of a new packet forwarder***

Chirpstack - Presentation



Chirpstack - Presentation



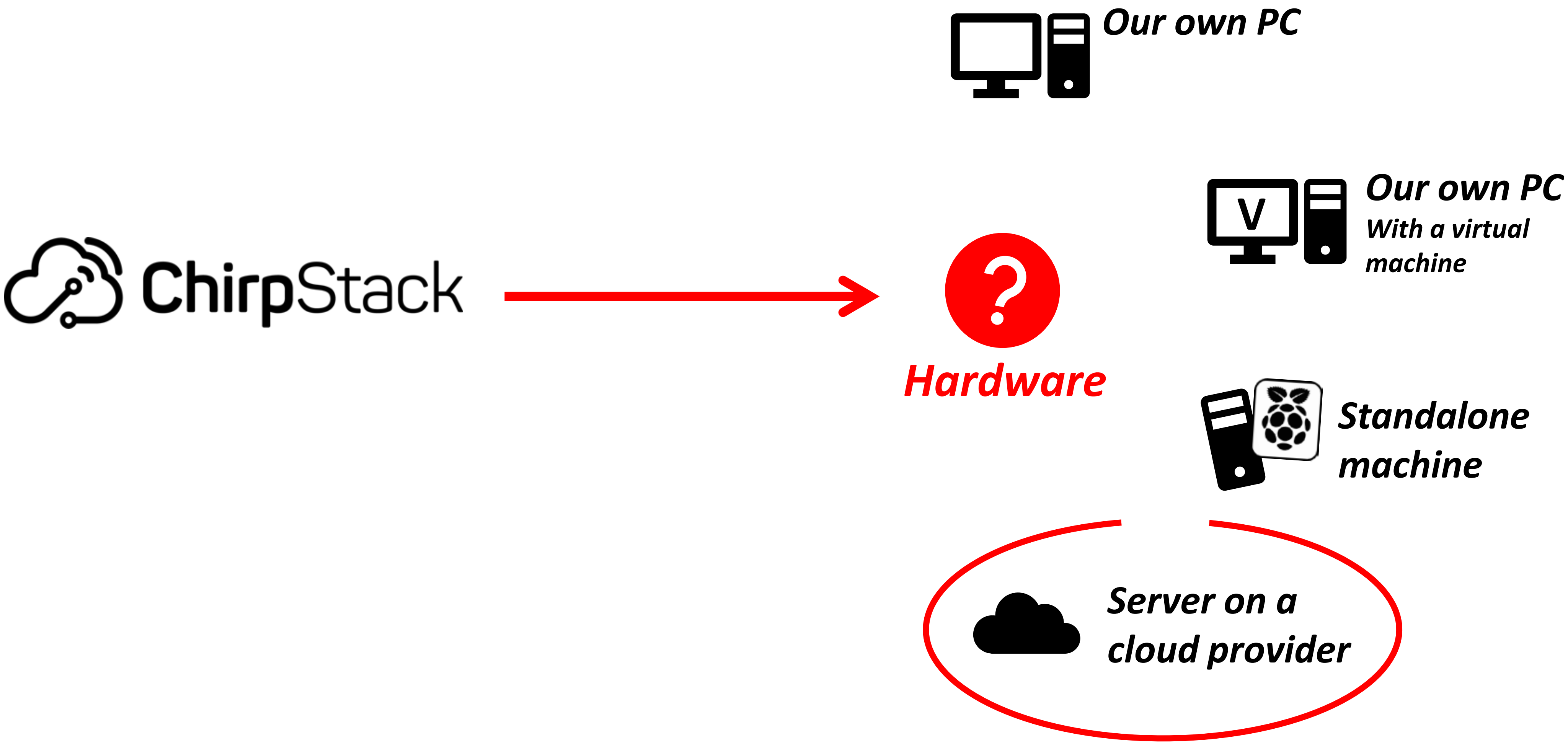
Chirpstack - Installation

ChirpStack

?

Hardware

Chirpstack - Installation



Chirpstack - Installation

Server on a cloud provider

- ***2 Giga Bytes of RAM***
- ***1 virtual Core***
- ***Basic Linux Debian Operating System***
- ***Installed: Docker and Docker-compose***

Domain name

- ***Instead of the server IP address:*** *chirpstack.univ-lorawan.fr*
- ***<u>It is not compulsory</u>, just more convenient.***

Chirpstack - Demonstration

Our Private Network configuration:

Step 1
Gateway configuration → ***It tells the Gateway how to reach the NS***

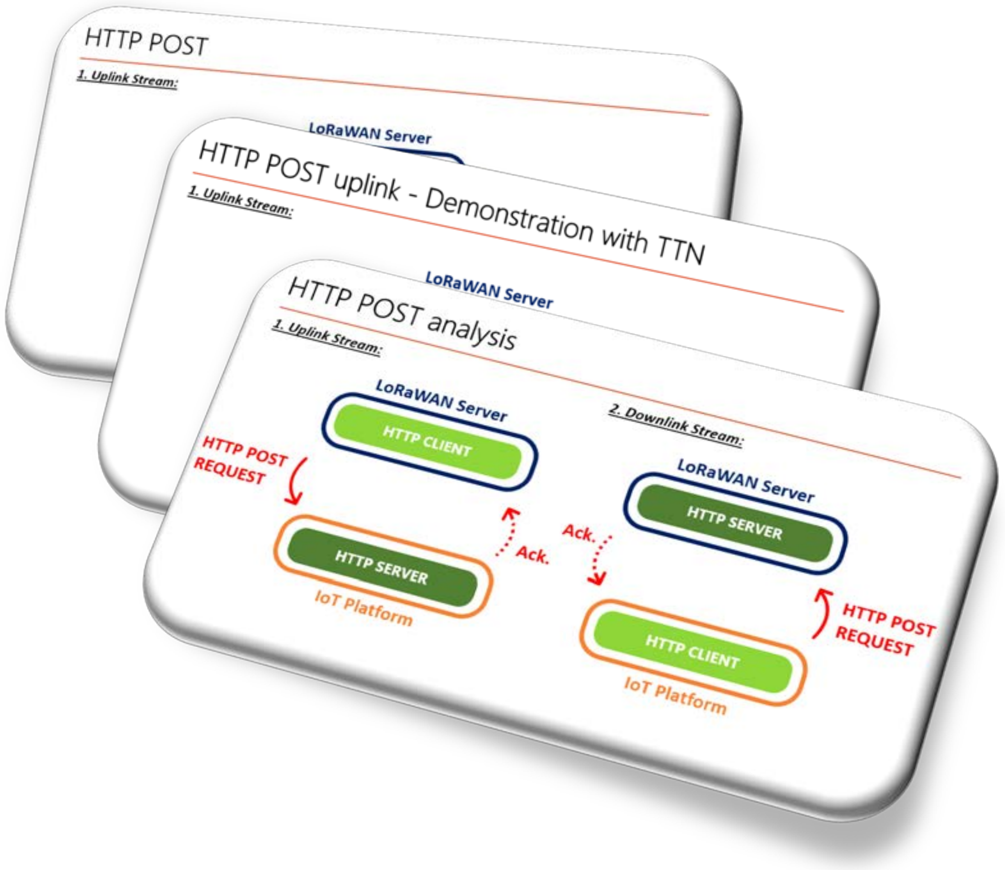
Step 2
Gateway registration → ***Only a registered Gateway is allowed to talk to the NS***

Step 3
Device registration → ***One with an OTAA activation***
One with an ABP activation

Step 4
Device configuration → ***Dedicated firmware with the right LoRaWAN keys***

Chirpstack - HTTP integration

Previous chapter about HTTP

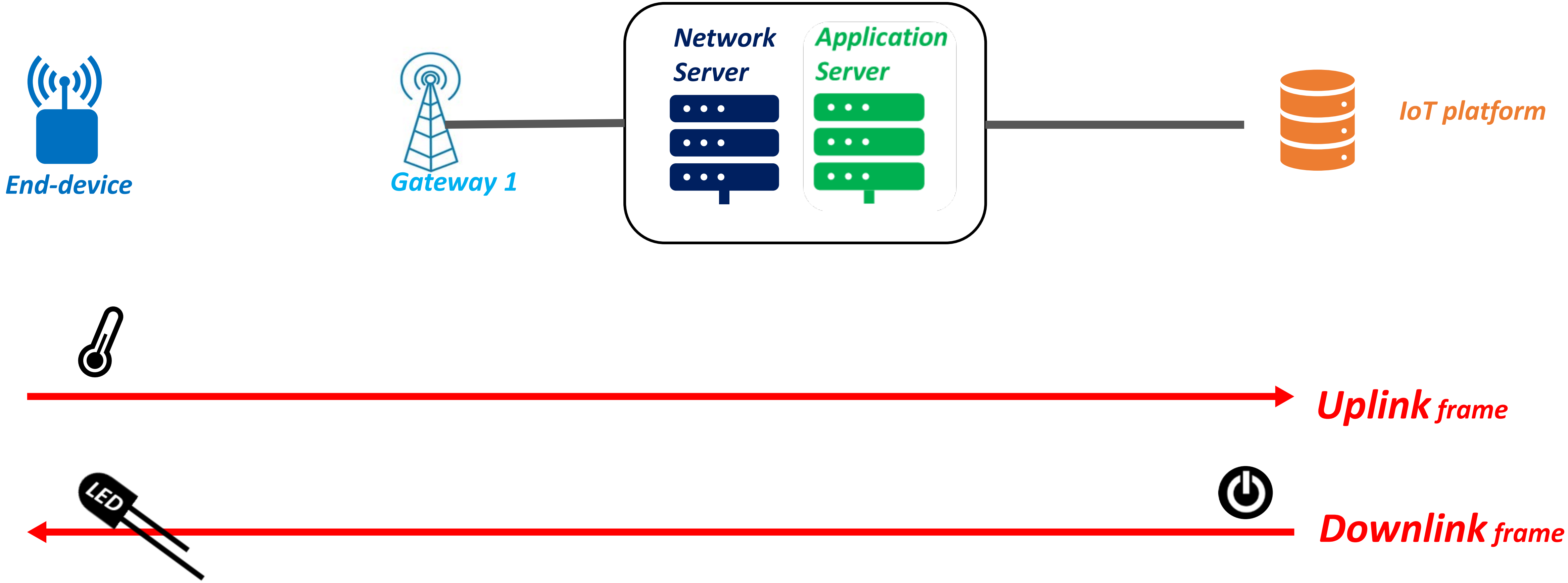


HTTP with ChirpStack

→ We will use the same diagram as the one in the previous chapter

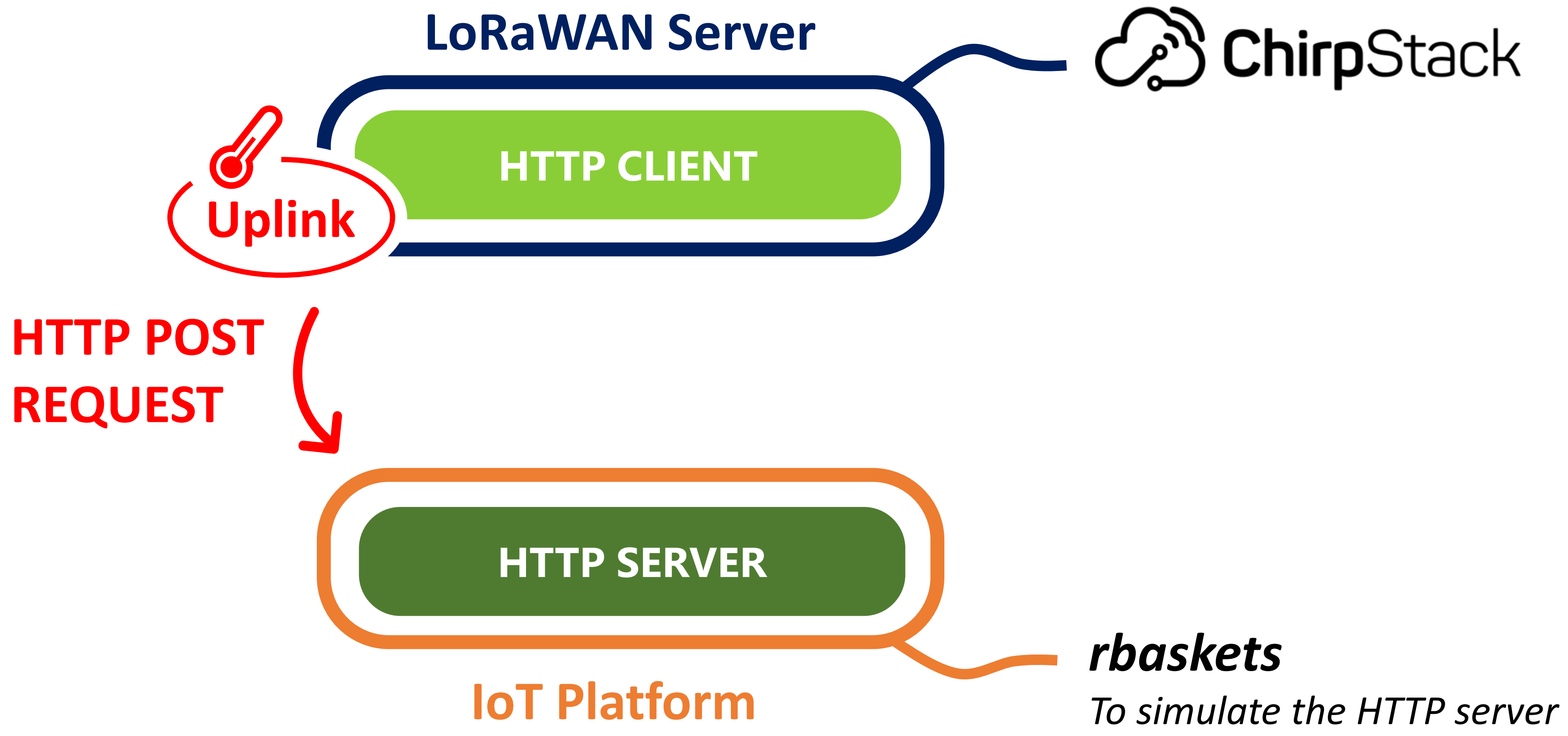
→ HTTP integration always looks similar

Chirpstack - HTTP integration



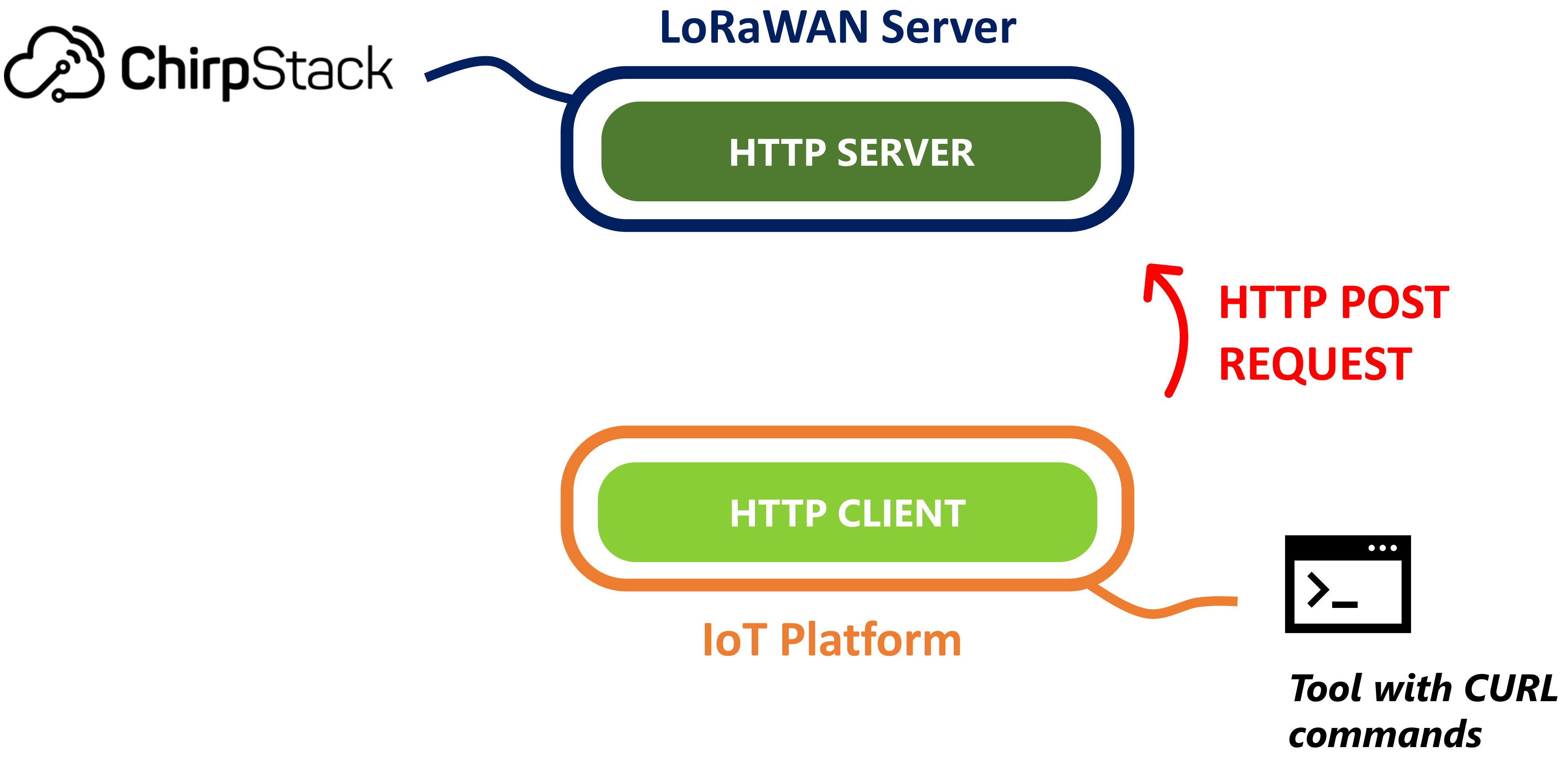
Chirpstack - HTTP integration

1. Uplink Stream:



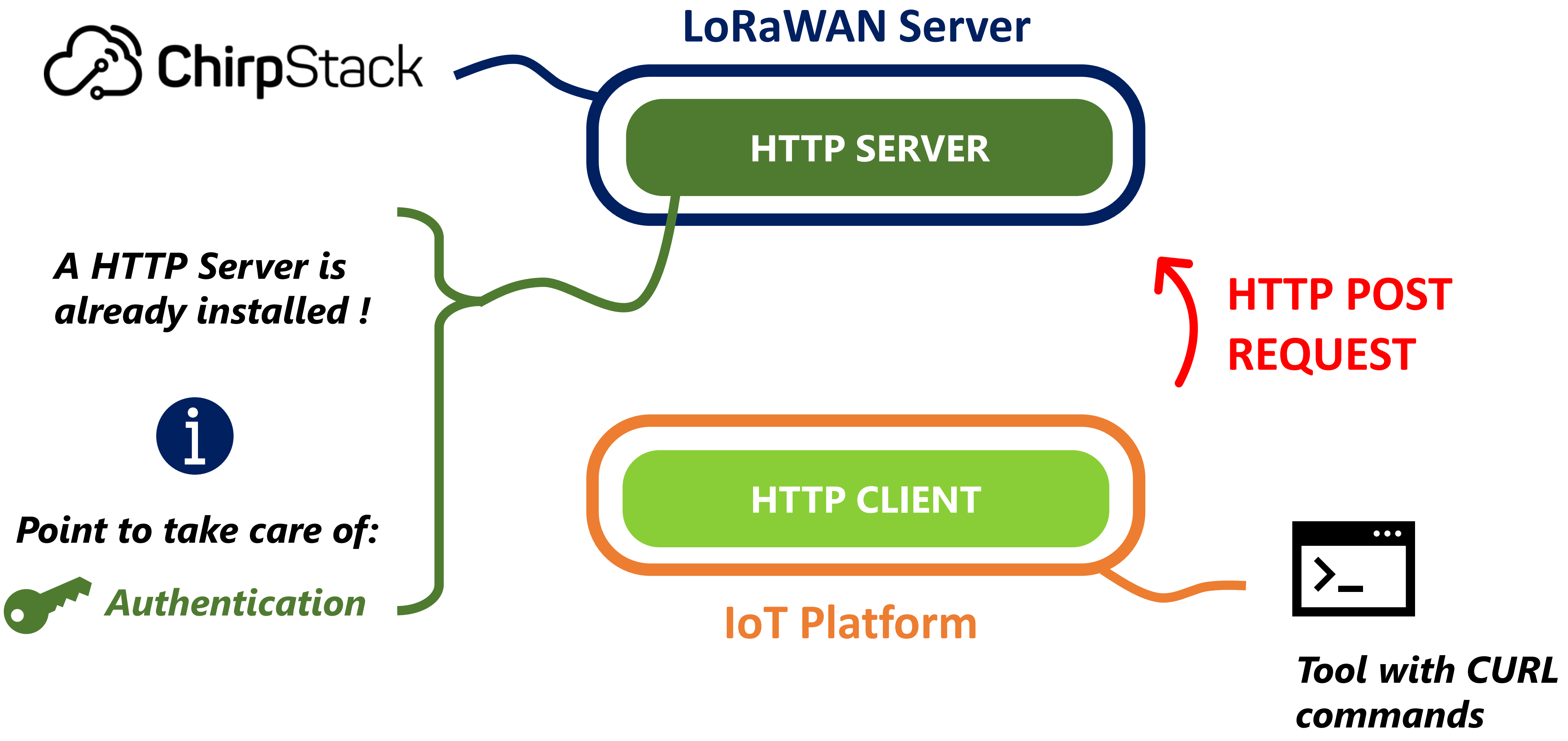
Chirpstack - HTTP integration

2. Downlink Stream:

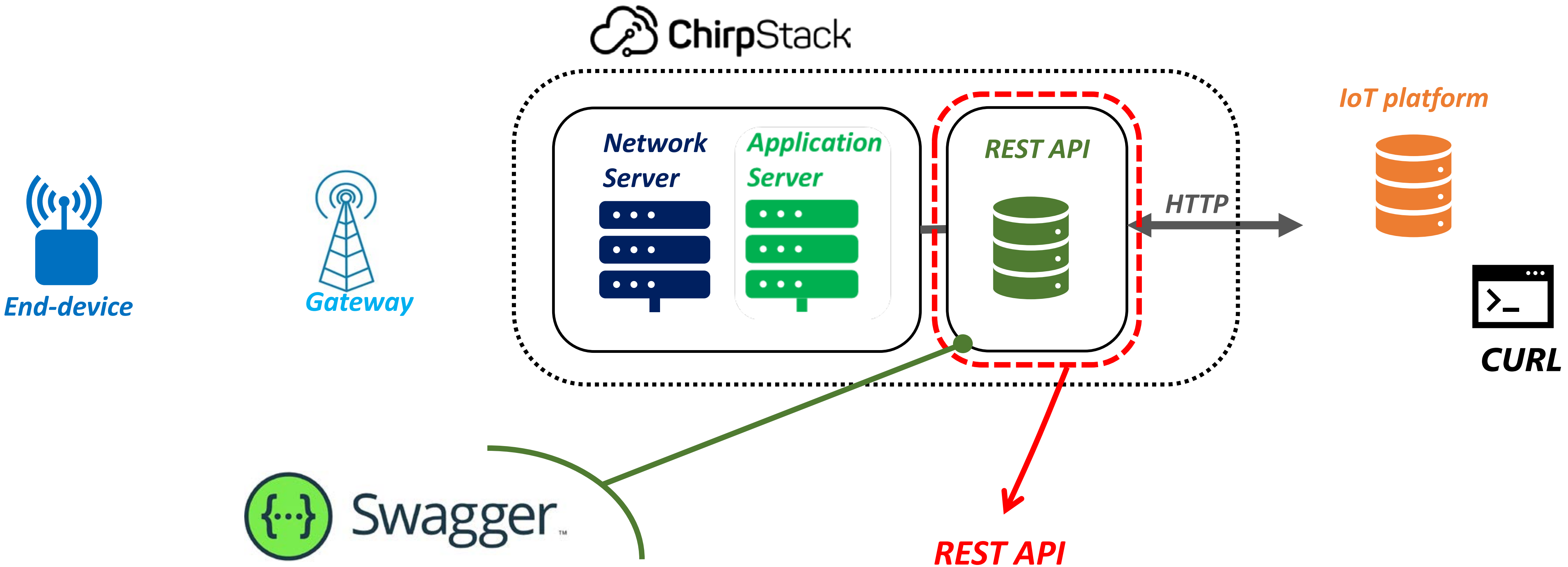


Chirpstack - HTTP integration

2. Downlink Stream:



Chirpstack - HTTP integration

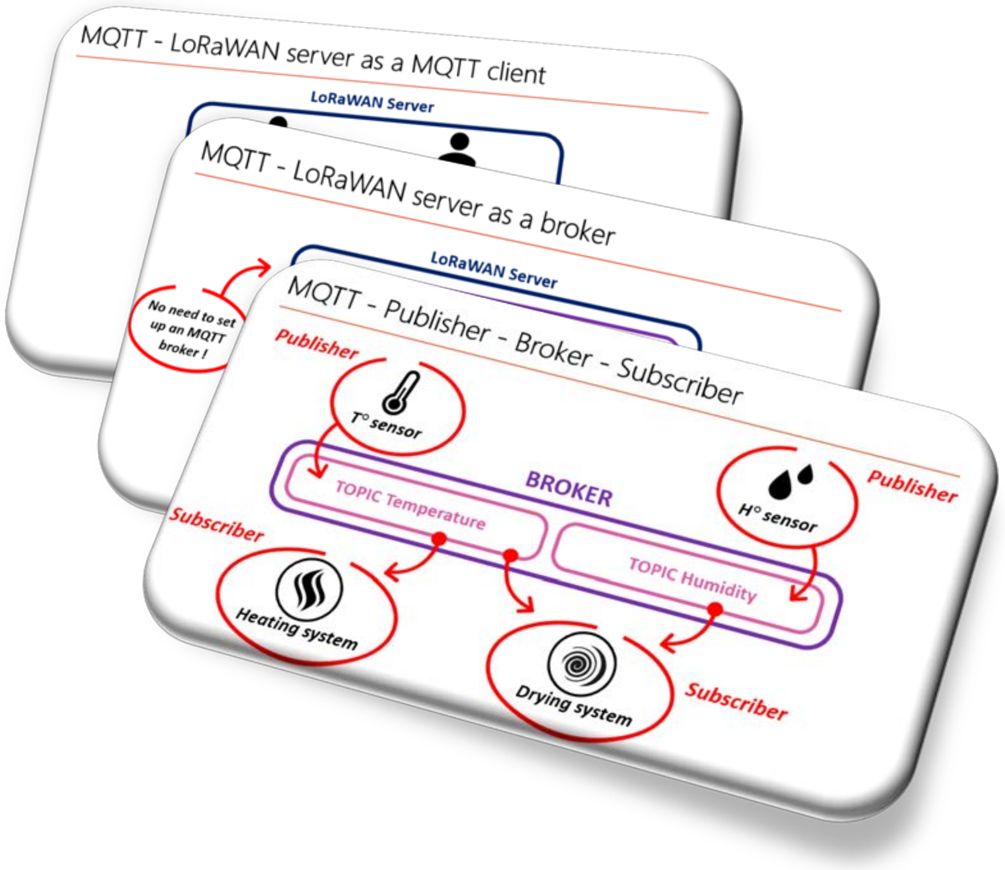


A tool with a user interface

REST API
handles HTTP Request used to access data on the LoRaWAN server

Chirpstack - MQTT integration

Previous chapter about MQTT

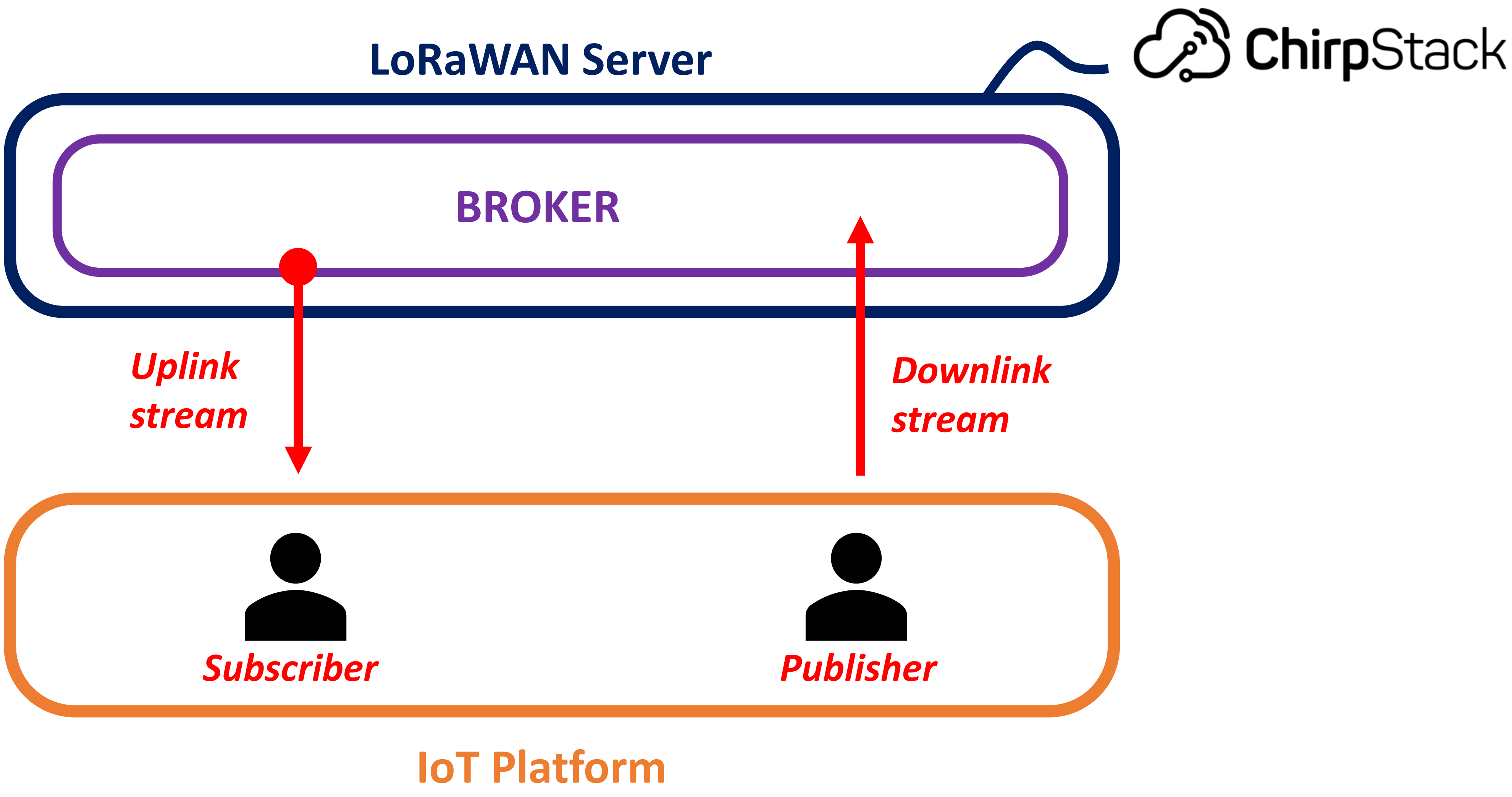


MQTT with ChirpStack

→ We will use the same diagram as the one in the previous chapter

→ MQTT works exactly in the same way as presented in the previous chapter

Chirpstack - MQTT integration



Chirpstack - MQTT integration

Step 1
Broker address

LoRaWAN_IP_Address:1883

Our case: chirpstack.univ-lorawan.fr:1883

Step 2
Credentials

Credentials

See the Broker configuration file

Step 3
Subscriber

Subscriber

Set the topic to subscribe to

Our topic: # (all events)

Step 4
Publisher

Publisher

Set the topic to publish on AND payload format

Our topic: see documentation
Our payload format: JSON (see documentation)

The Things Stack - Presentation

Previously:

ChirpStack

The Things Stack - Presentation

Now:

THE THINGS STACK

The Things Stack - Presentation

The Things Stack - Presentation

THE THINGS NETWORK

THE THINGS STACK

GitHub

THE THINGS STACK Community Edition

→ *Community edition (also called "The Things Network")*

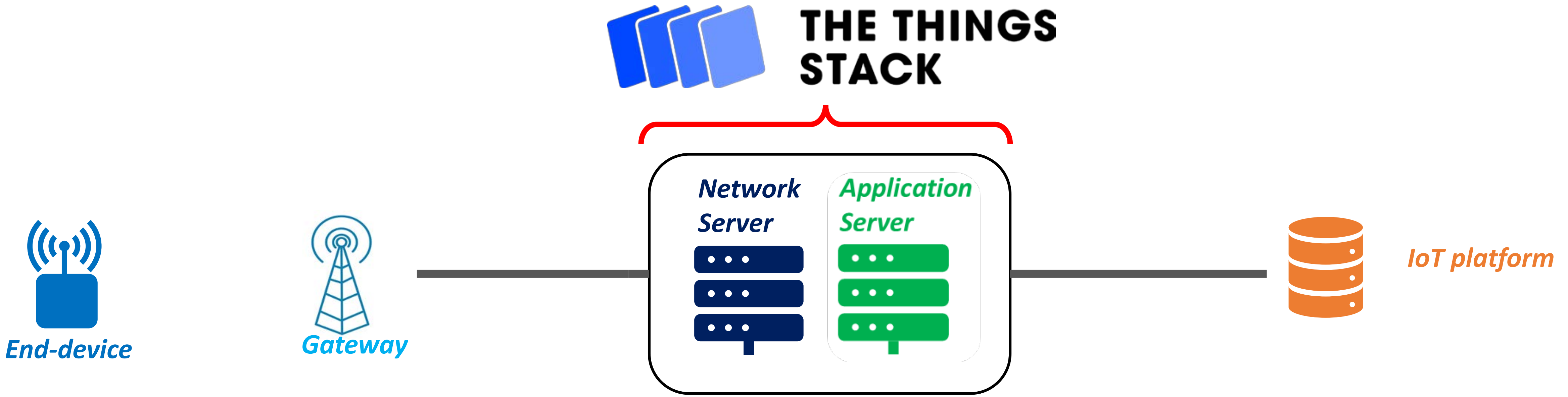
THE THINGS STACK Cloud

→ *Commercial edition*

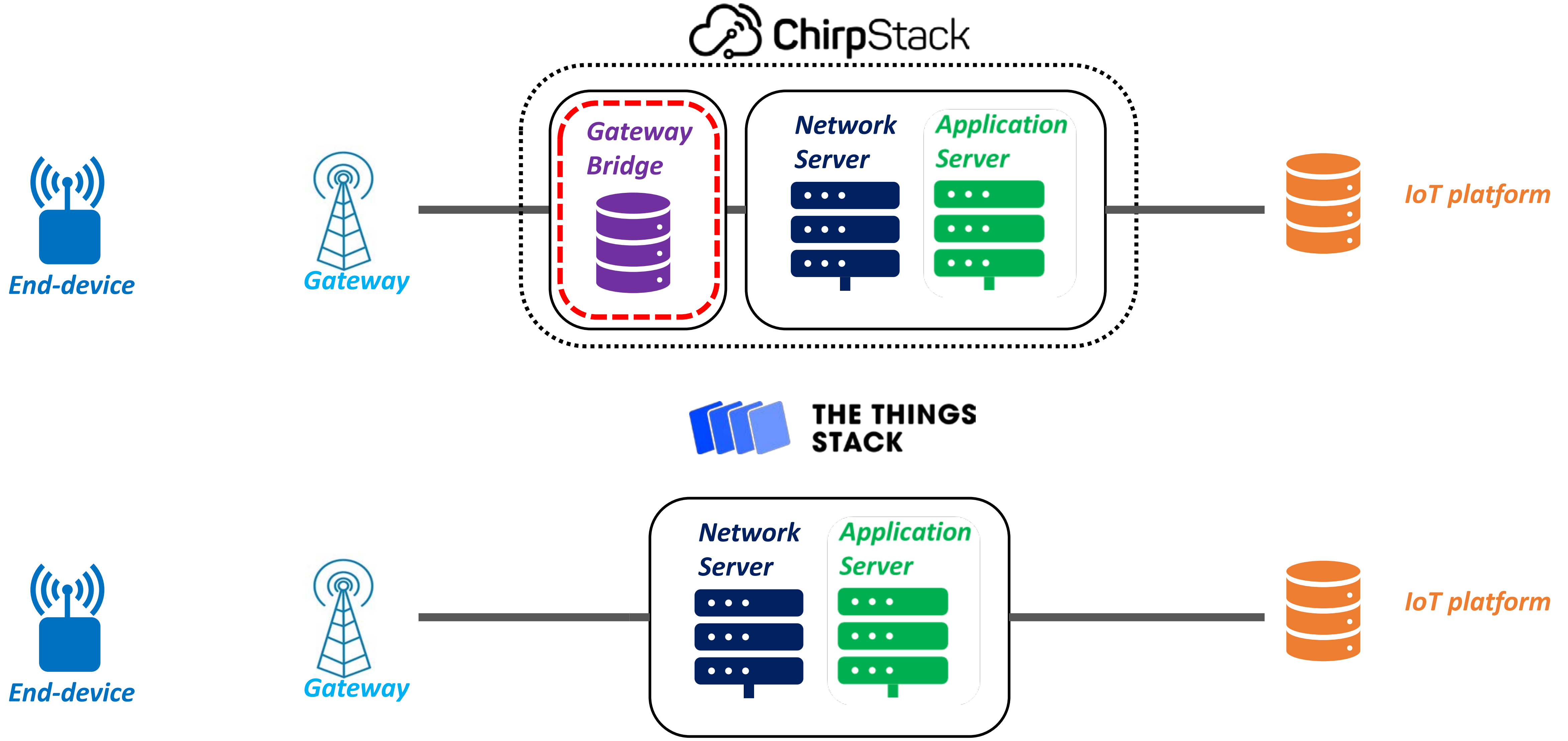
THE THINGS STACK Open Source

→ *Downloadable edition (for free)*

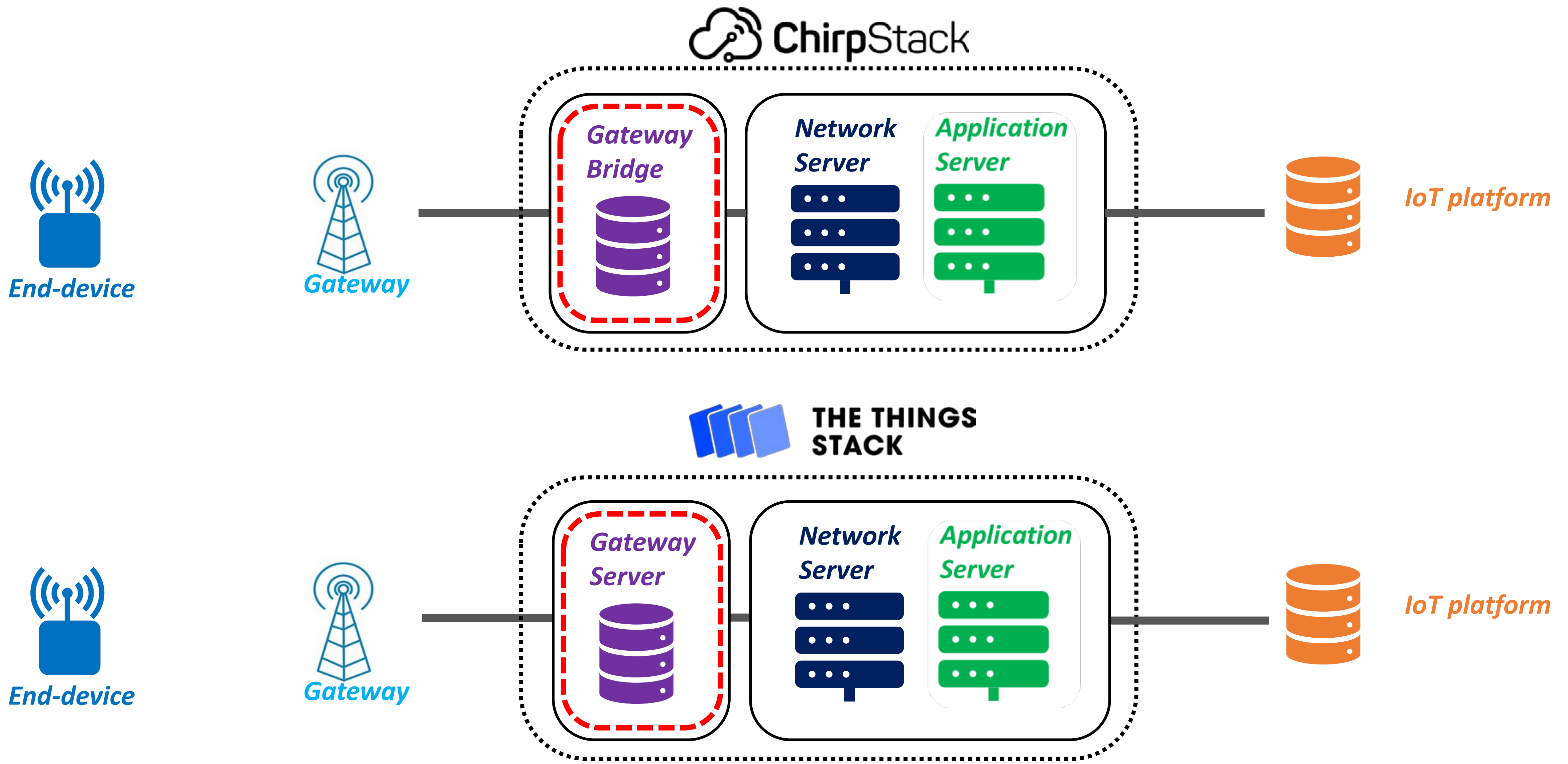
The Things Stack - Presentation



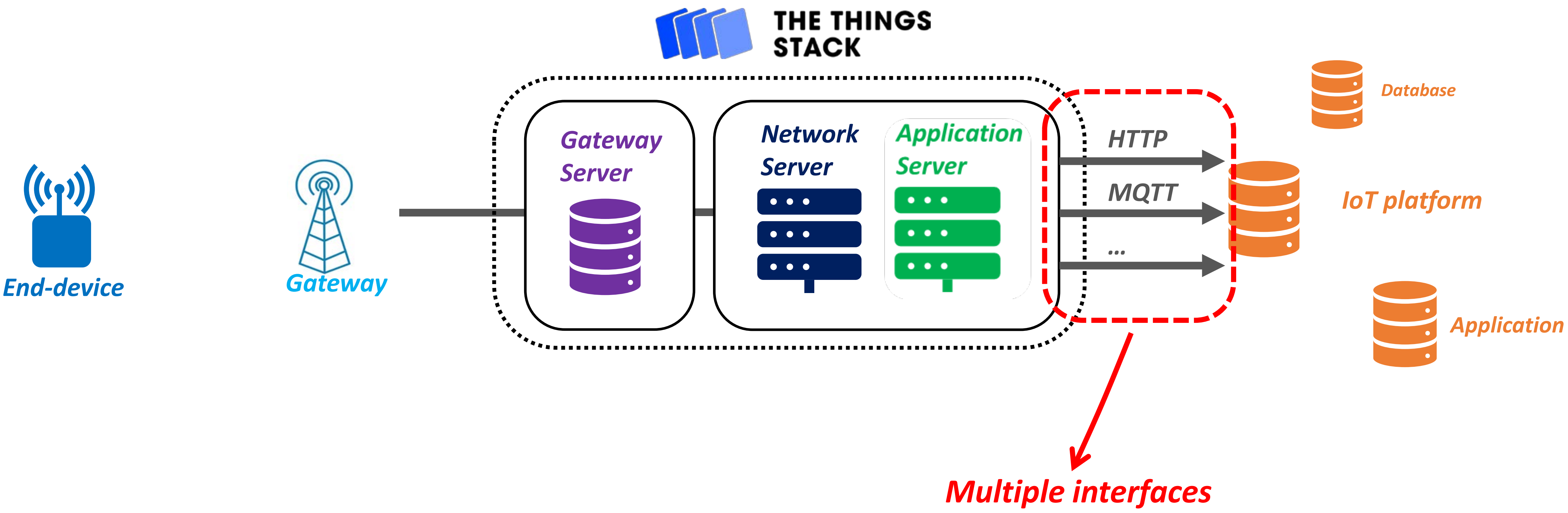
The Things Stack - Presentation



The Things Stack - Presentation



The Things Stack - Presentation



The Things Stack - Installation

The Things Stack - Installation

THE THINGS STACK

SO FAR,

We used Docker

With a *docker-compose.yml* ***file***

And a *docker-compose up* ***command***

WITH THE THINGS STACK **,**

TTS comes up with a secure connection

HTTPS → "S" is for secure

We need to setup HTTPS

- ***A domain name***
- ***Set up the configuration file***

Go to the "HOW TO" section to know more about secure connections

The Things Stack - Installation

Server on a cloud provider

- ***2 Giga Bytes of RAM***
- ***1 virtual Core***
- ***Basic Linux Debian Operating System***
- ***Installed: Docker and Docker-compose***

Domain name

- ***Instead of the server IP address:*** *tts.univ-lorawan.fr*
- ***<u>It is not compulsory</u>, just more convenient.***

The Things Stack - Demonstration

Our Private Network configuration:

Step 1
Gateway configuration → ***It tells the Gateway how to reach the NS***

Step 2
Gateway registration → ***Only a registered Gateway is allowed to talk to the NS***

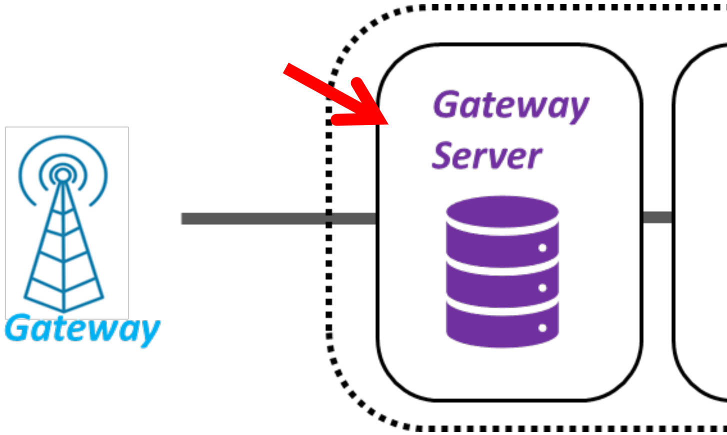
Step 3
Device registration → ***Register our end-device on the NS***

Step 4
Device configuration → ***Dedicated firmware with the right LoRaWAN keys***

The Things Stack - Demonstration

Step 1
Gateway configuration

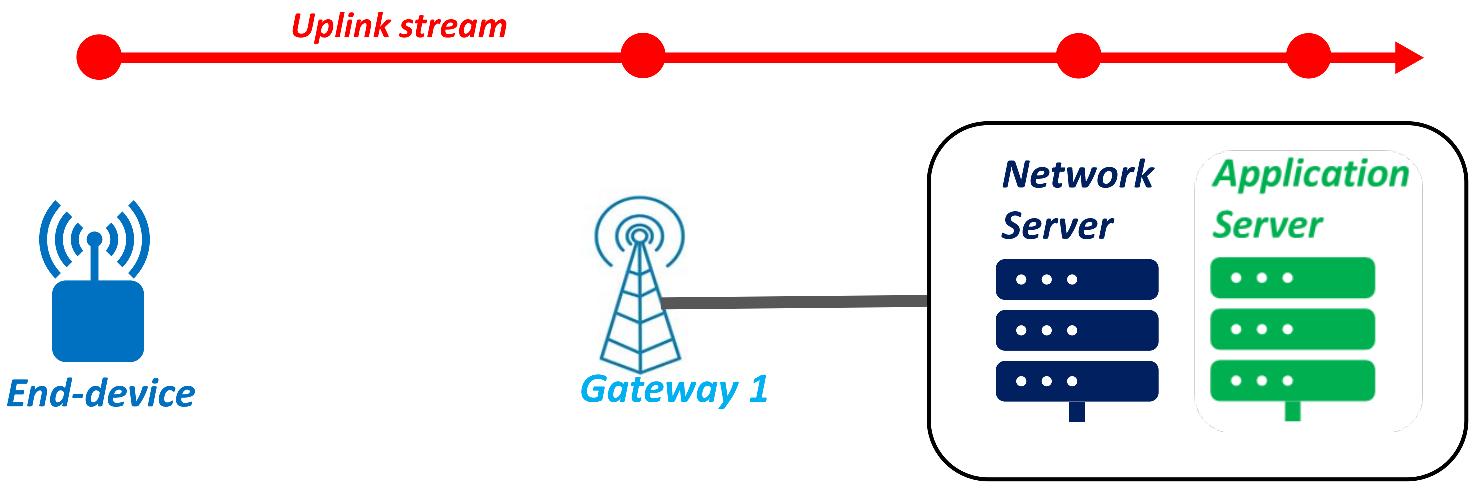
- **_TTS listens on port 1700_**
- **_We will use UDP Semtech packet forwarder_**
- **_We need to configure,_**
  - **_Address:_** *tts.univ-lorawan.fr*
  - **_Port:_** *1700*



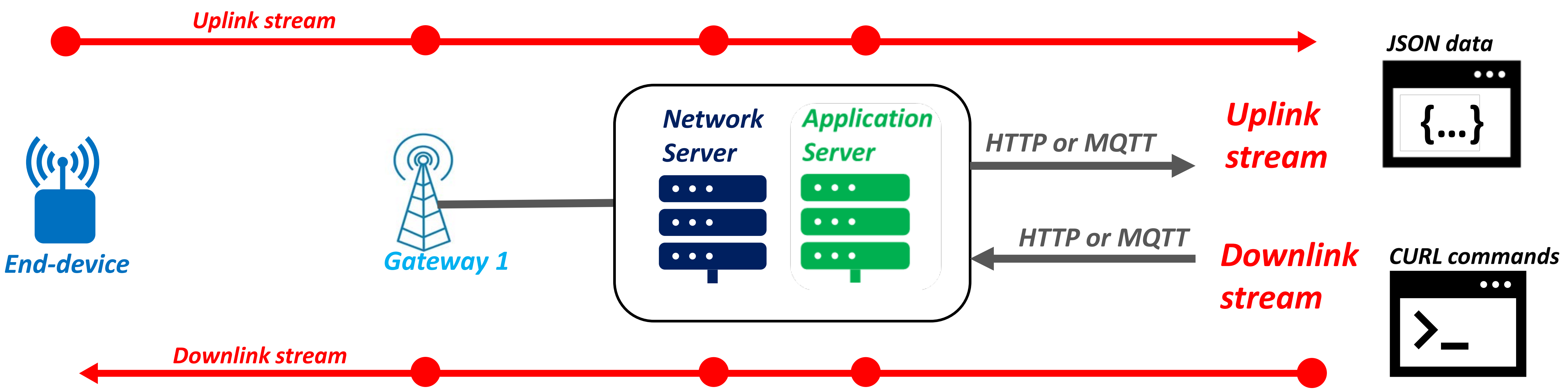
Setting up our own IoT Platform

- ✓ …
- ✓ …
- ✓ …

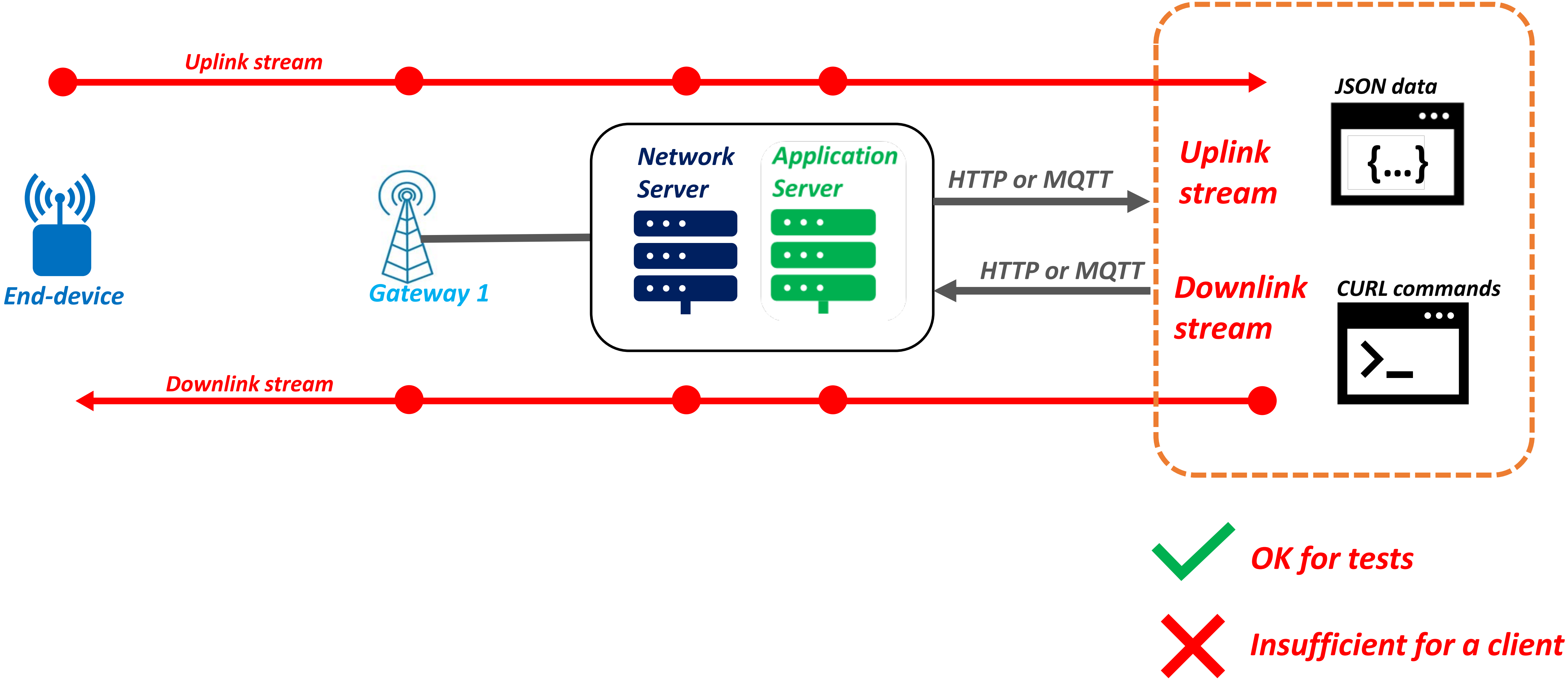
IoT platform purposes



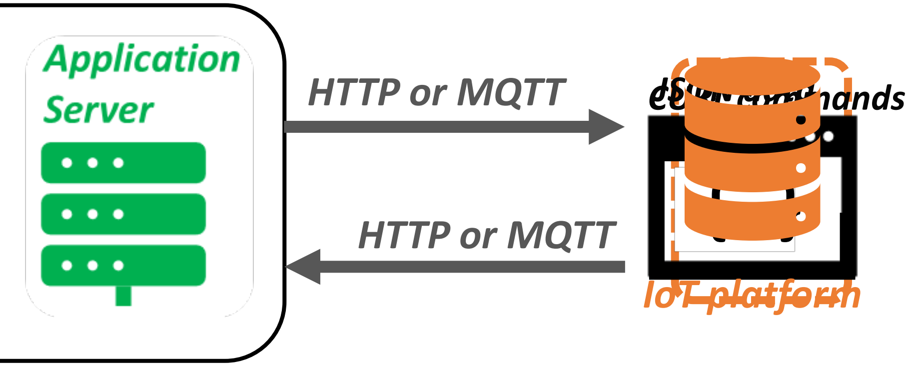
IoT platform purposes



IoT platform purposes



IoT platform purposes



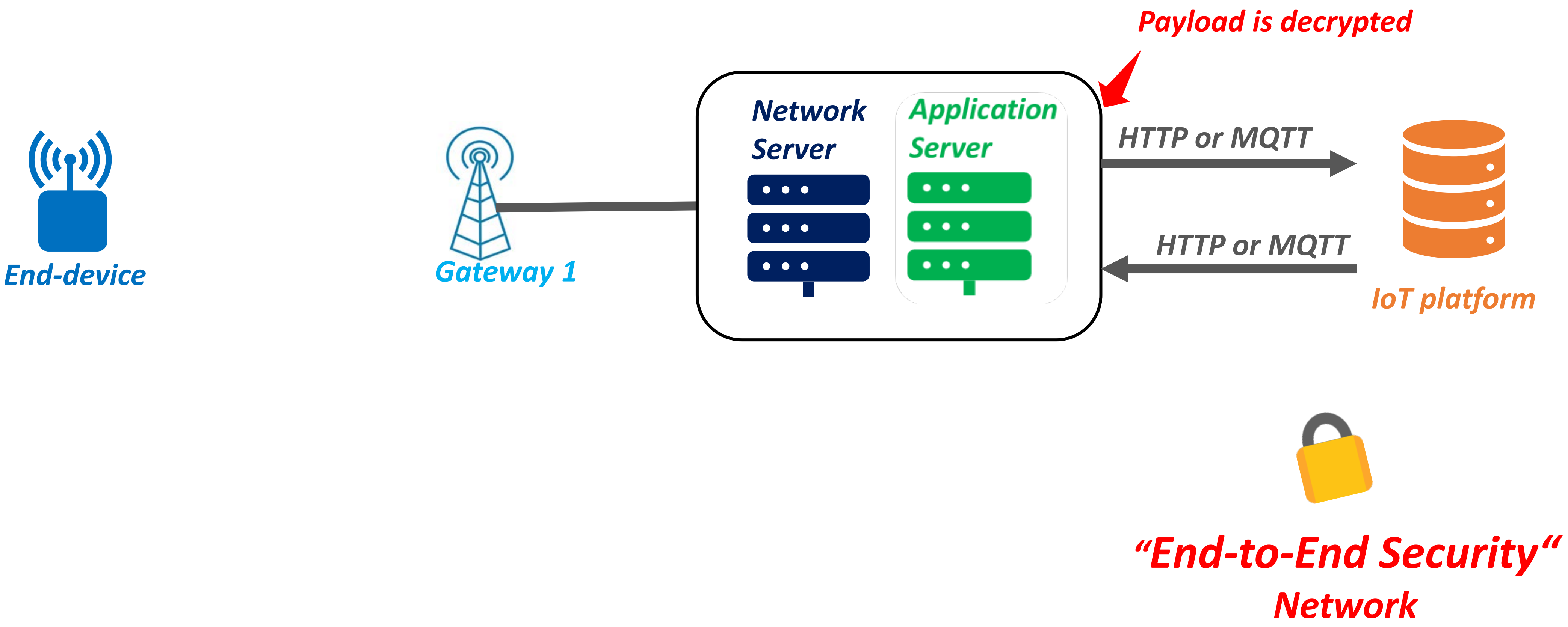
UPLINK STREAM

- HTTP MQTT — *Need to support MQTT/HTTP protocol*
- *Need to store data*
- *Need to process or decode the data*
- *Need to represent the data*

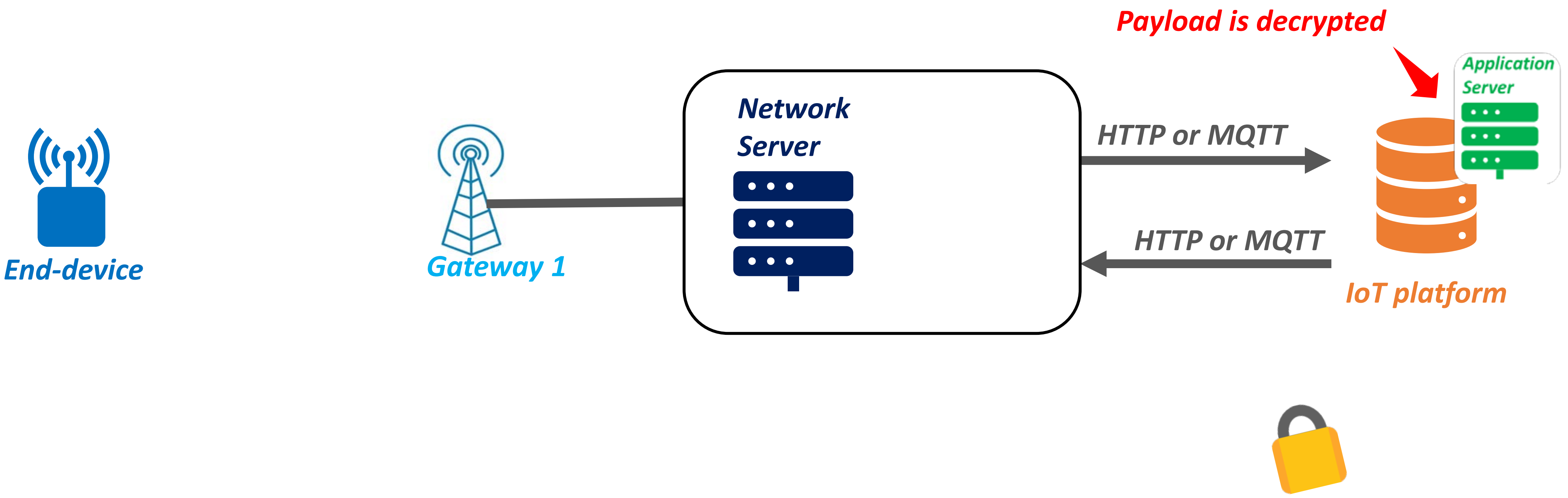
DOWNLINK STREAM

- *Need to send commands from the User Interface*
- *Need to store the commands sent*
- HTTP MQTT — *Need to support MQTT/HTTP protocol*

IoT platform purposes



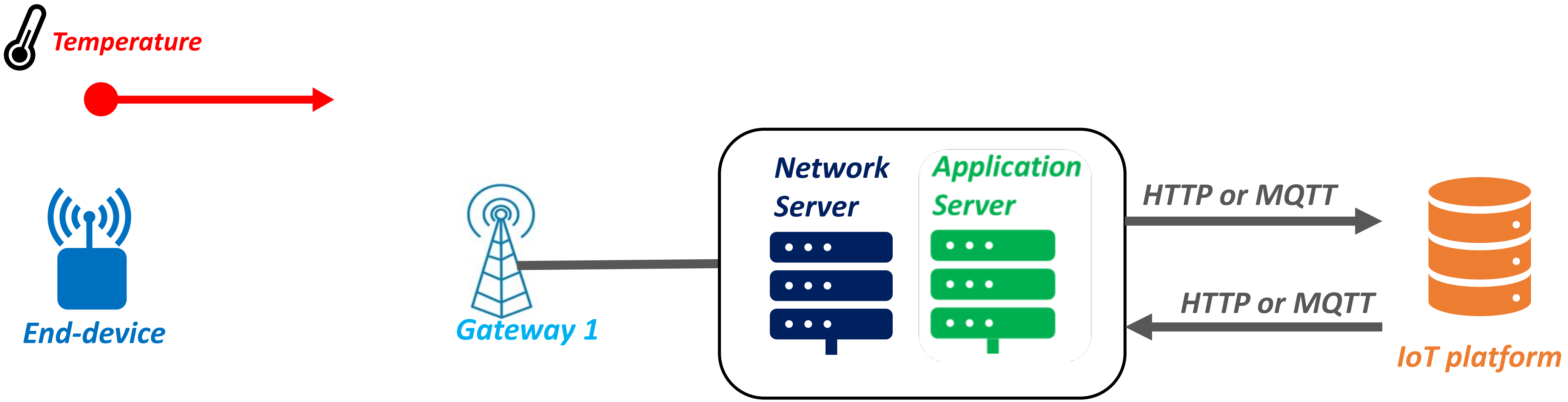
IoT platform purposes



“End-to-End Security“
Network

IoT platform purposes

Our demonstration :



IoT Platform architectures

IoT platform

IoT Platform architectures

IoT Platform solutions … There are plenty of them !

Our solutions:

- ***Most of them are free & open-source***
- ***Can work both with <u>uplink</u> and downlink streams***

IoT Platform architectures

IoT platform

HTTP
MQTT

Data collection and transfer

Data storage

Data visualization

Web Server and UI

Example:

Node-RED

HTTP and MQTT connections, Graphical UI, etc.

IoT Platform architectures

IoT platform

HTTP MQTT — **_Data collection and transfer_**

Data storage

Data visualization

Web Server and UI

Our solution :

Node-RED — **_To connect our IoT Platform to our LoRaWAN Server_**

influxdb — **_To store data_**

Grafana — **_To create dashboards and customize our User Interface_**

IoT Platform architectures

Our solution :

IoT platform

Node-RED — *To connect our IoT Platform to our LoRaWAN Server*

influxdb — *To store data*

Grafana — *To create dashboards and customize our User Interface*

\+ mosquitto — *MQTT broker*

MING Stack

***M**osquitto*
***I**nfluxdB*
***N**odeRED*
***G**rafana*

IoT Platform architectures

Another solution:

IoT platform

To connect our IoT Platform to our LoRaWAN Server

To store data

To create dashboards and customize our User Interface

ELK Stack

Elasticsearch

Logstash

Kibana

IoT Platform architectures

Another way:

IoT platform

Use an IoT platform without creating one.

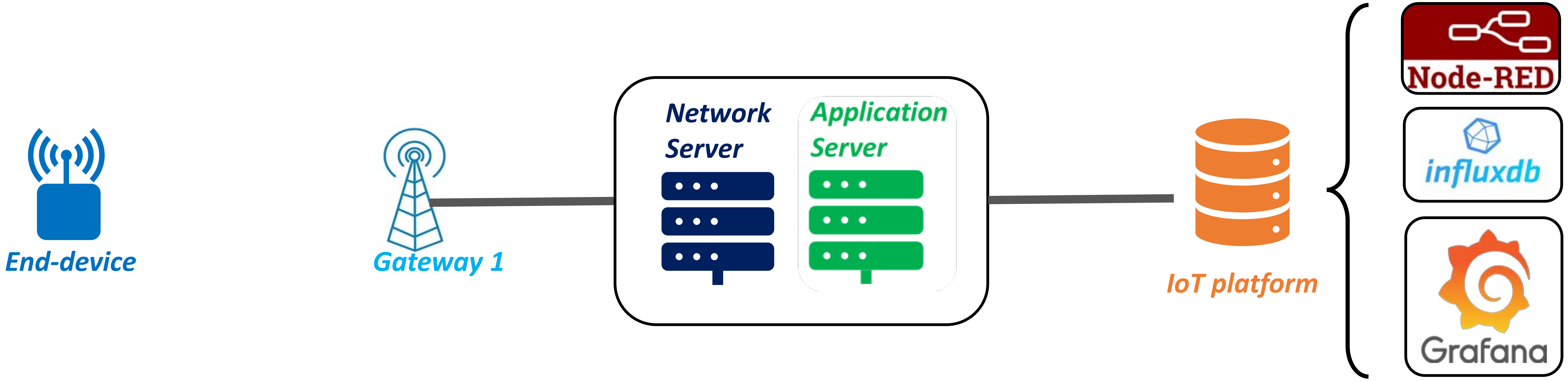
A professional service !

Not free

Good choice for a big project

Our IoT platform solution

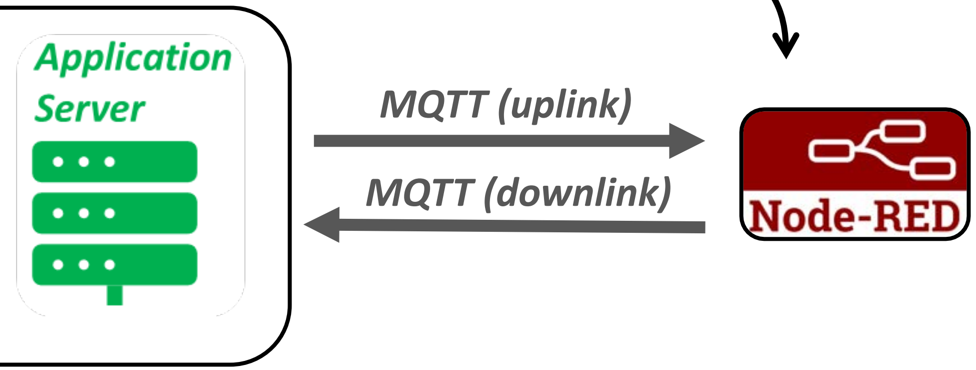
<u>Our solution :</u>



Our IoT platform solution

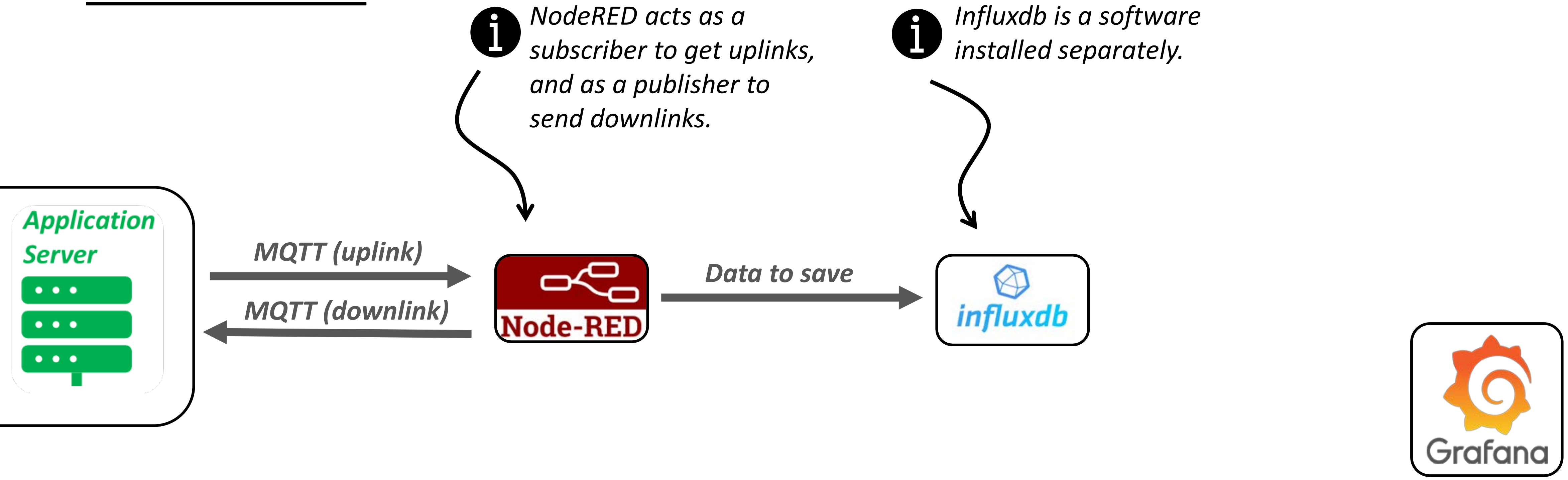
Our solution :

NodeRED acts as a subscriber to get uplinks, and as a publisher to send downlinks.



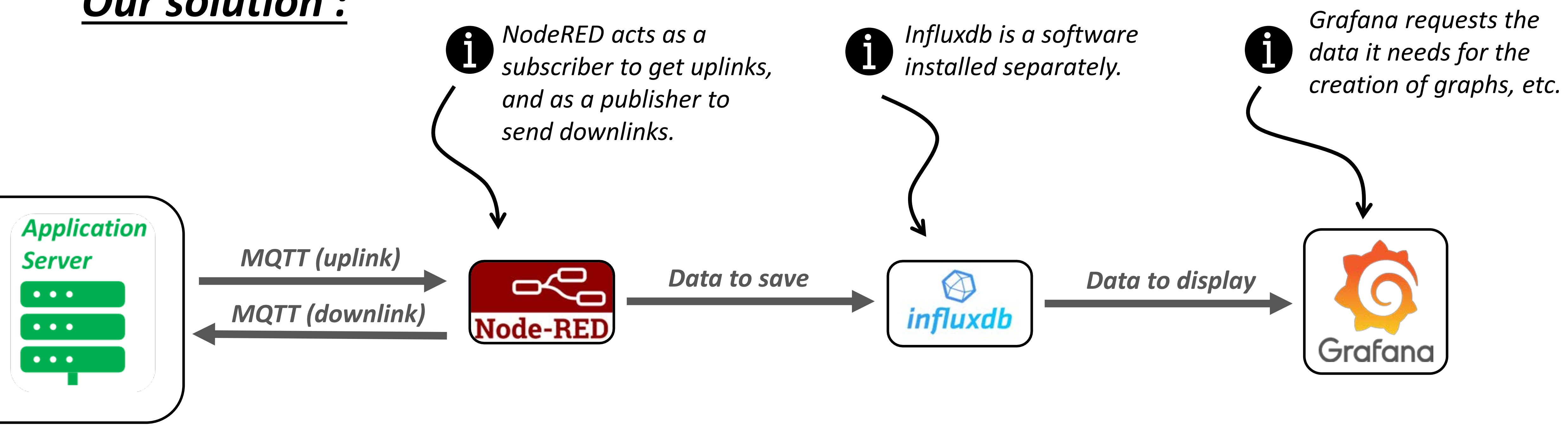
Our IoT platform solution

Our solution :



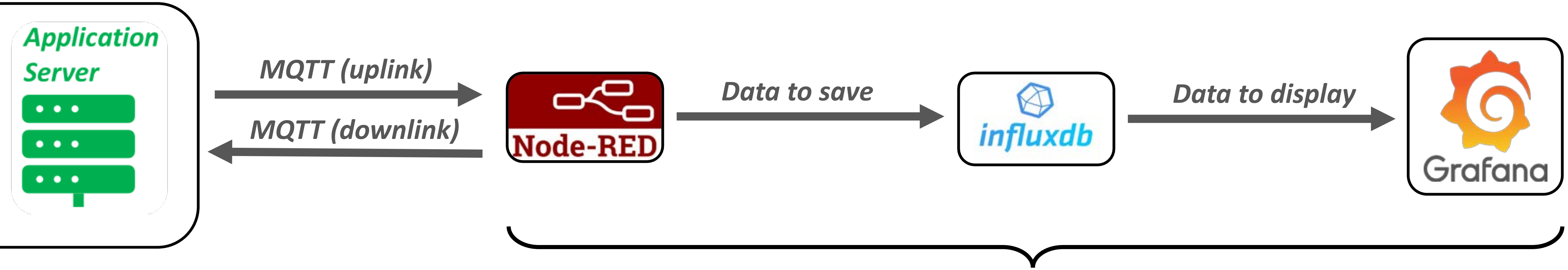
Our IoT platform solution

Our solution :



Our IoT platform solution

Our solution :



How to install these services ?

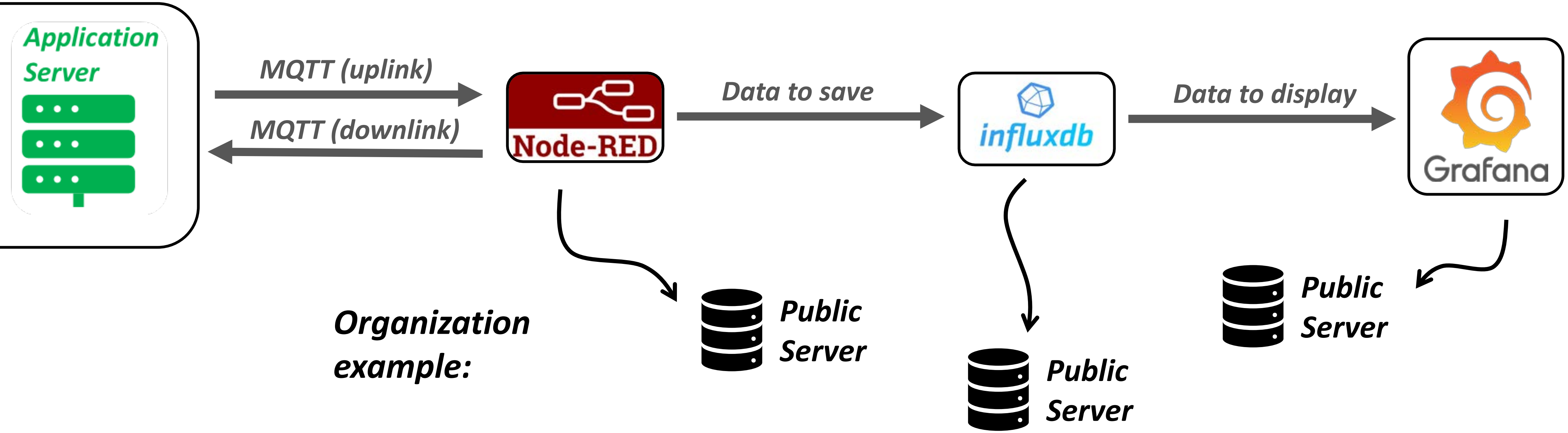
With Docker !

Where can we set up this IoT platform ?

Anywhere !
In our case: Public Server

Our IoT platform solution

Our solution :

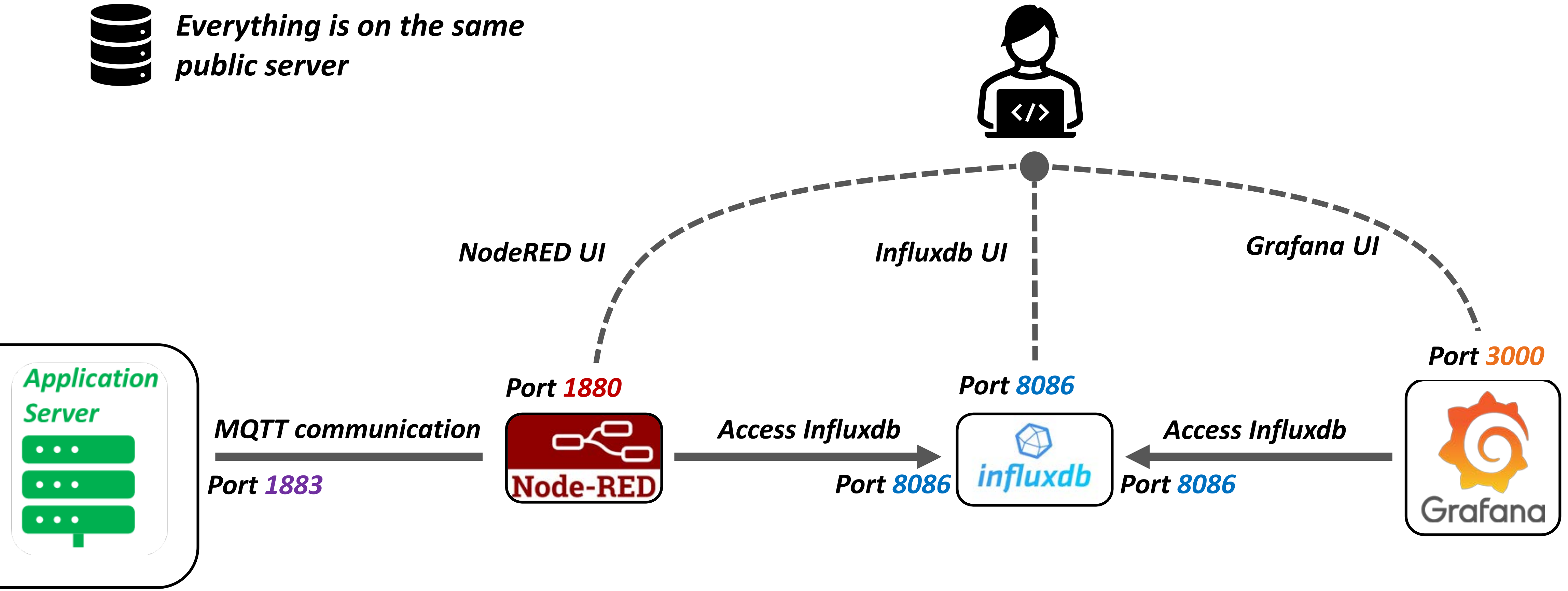


Container talk to each other thanks to an IP address on a specific port.

Our IoT platform solution

Our solution:

Everything is on the same public server



Our IoT platform specifications

<u>Let’s build our IoT Platform !</u>

Our requirements: ***1 dashboard with Temperature and RSSI***

Steps to achieve our goal:

Step 1 ***Build an MQTT subscriber on Node-RED to get the uplink data.***

Step 2 ***Extract the exact value we want: the temperature & the RSSI.***
Send the extracted values to Influxdb.
Check our data are available on Influxdb.

Step 3 ***Tell Grafana how it can access the database.***
Create a request to get the values we need.