Introducción a LoRaWAN

Conceptos básicos

Agenda

- Internet of Things (IoT)
- LoRaWAN™ Network Protocol
- LoRa® Technology Wireless Modules
- Getting Started with RN2903 Module

Primero lo primero

- Como se que todos quieren llevarse “algo”, vamos a hacer unos juegos con una app que se llama Kahoot
- Por favor, descarguen Kahoot en sus teléfonos o pueden usarla desde un navegador (www.kahoot.it)
- El que mas rápido responde (y bien) suma mas puntos
- Vamos a dar premios a los 3 mejores puntajes

¿Cómo funciona Kahoot!?

Kahoot!

Ejemplo sin puntaje

Agenda

- **Internet of Things (IoT)**
- LoRaWAN™ Network Protocol
- LoRa® Technology Wireless Modules
- Getting Started with RN2903 Module
- Hands-on Labs

Internet of Things (IoT)

- **Loosely defined paradigm**
  - Connected devices
  - Communication happens without human intervention
- **“Thing” can contribute to “big data”**
  - Cloud-handled data
  - Queries on the cloud data can offer a high-level view

Internet of Things (IoT)

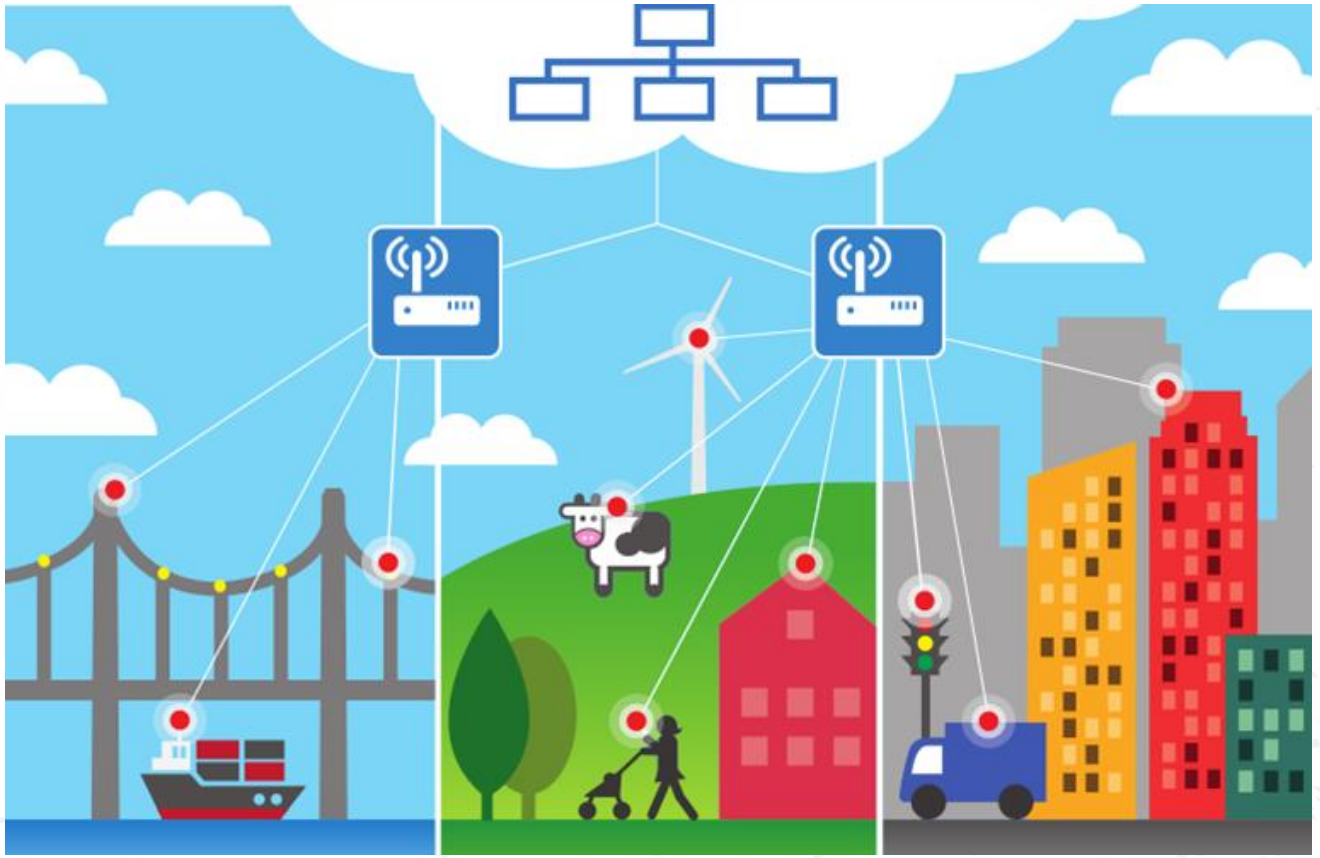
Types of Wireless Networks

Personal Area | **Local Area** | **Wide Area**

Bluetooth® | Wi-Fi® | Cellular (2G, 3G, 4G-LTE)

Internet of Things (IoT)

LoRaWAN™ Network



Monitoring / Control
Light Control

Smart Agriculture
Smart Energy

Smart City
Smart Home and Security

Internet of Things (IoT)

Who is the LoRa® Alliance?

- The LoRa® Alliance (http://lora-alliance.org/) is an open, non-profit association of members.
- Mission: to standardize Low Power Wide Area Networks (LPWAN)
- Alliance members will collaborate to drive the global success of the LoRaWAN™ protocol

Agenda

- Internet of Things (IoT)
- **LoRaWAN™ Network Protocol**
- LoRa® Technology Wireless Modules
- Getting Started with RN2903 Module
- Hands-on Labs

Sub-Agenda

- **LoRaWAN™ Network Protocol**
  - LoRa® Technology Modulation
  - How does LoRaWAN™ Technology Work?
  - End-Device Classes
  - End-Device Activation (Joining)
  - Security
  - End-Device Data Communication (Class A)
  - Adaptive Data Rate (ADR)

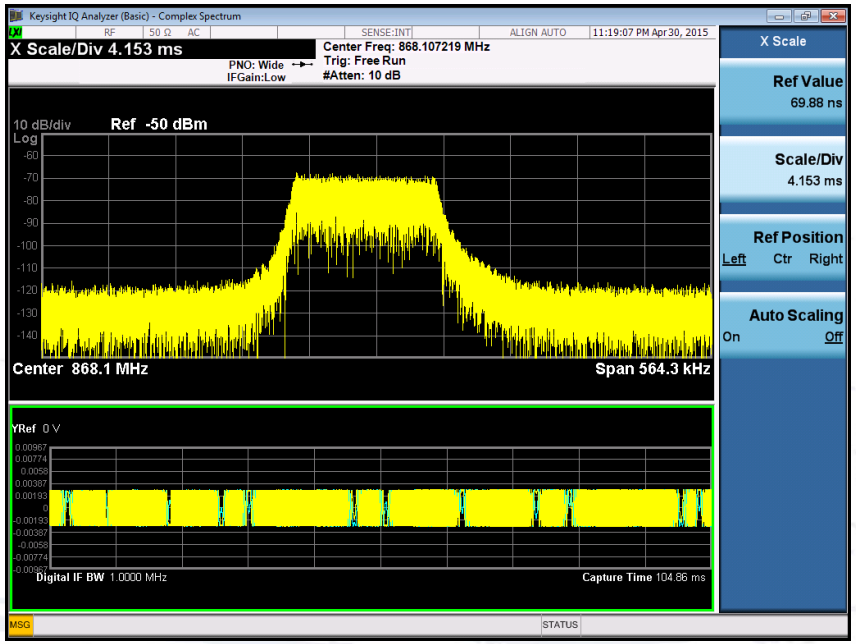
Sub-Agenda

- **LoRaWAN™ Network Protocol**
  - LoRa® Technology Modulation
  - How does LoRaWAN™ Technology Work?
  - End-Device Classes
  - End-Device Activation (Joining)
  - Security
  - End-Device Data Communication (Class A)
  - Adaptive Data Rate (ADR)

LoRaWAN™ Network Protocol

LoRa® Technology Modulation

- **Proprietary Spread Spectrum Technology**
  - Developed by Semtech Corporation (http://www.semtech.com/)
  - Chirped-FM
  - Processing gain = increased receive sensitivity
  - Enables longer range at expense of lower data rate



LoRaWAN™ Network Protocol

LoRa® Technology Modulation

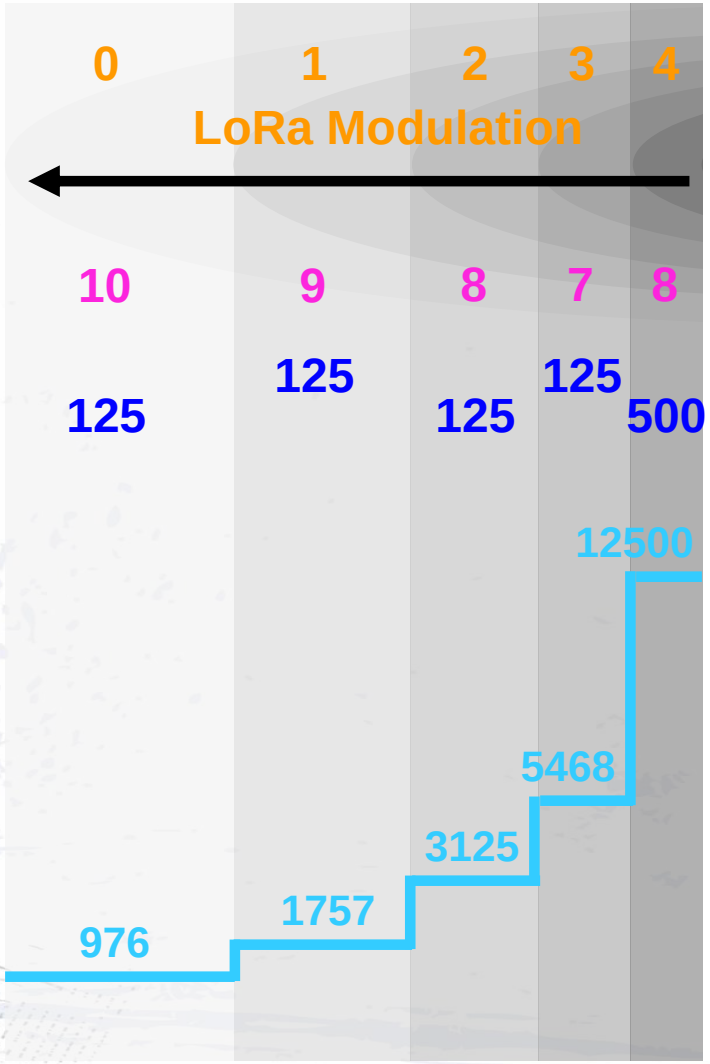
- **Spreading Factor (SF)**
  - Programmable SF:

    7, 8, 9, 10, 11, 12
  - The higher the SF the more information transmitted per bit; therefore higher processing gain
- **Bandwidth (BW)**
  - Programmable signal BW settings:

    125 kHz, 250 kHz, 500 kHz
  - For a given SF, a narrower BW = increased receive sensitivity; however, increased time on air
- **Forward Error Correction (FEC) Code Rate (CR)**
  - Additional coding rate provides more redundancy to detect errors and correct them

LoRaWAN™ Network Protocol

LoRaWAN™ Modulation Settings for North America



Data Rate (DR)

Range

Spreading Factor (SF)

Bandwidth (BW) (kHz)

Bitrate (BR) (bps)

LoRaWAN™ Network Protocol

LoRaWAN™ Modulation Settings for North America

Longest Distance on LoRa® Modulation

- **Data Rate (DR) = 0**
  - LoRa® modulation
  - Spreading Factor (SF) = SF10
  - Bandwidth (BW) = 125 kHz
  - Coding Rate (CR) = 4/5
- **Bit Rate = 976 bps**
- **Max Application Payload Size = 11 bytes**
  - Time On Air = 371 ms

LoRaWAN™ Network Protocol

LoRaWAN™ Modulation Settings for North America

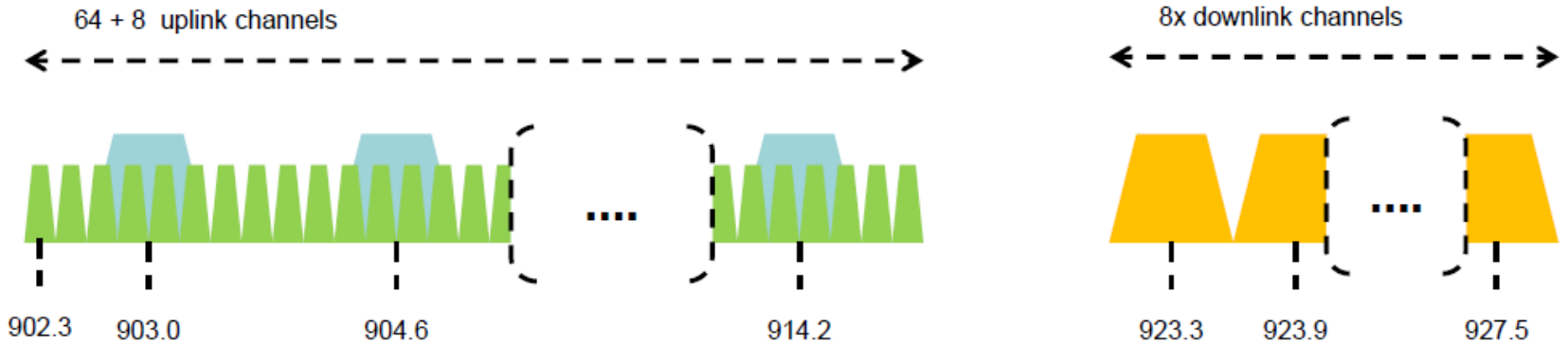
Highest Bit Rate on LoRa® Modulation

- **Data Rate (DR) = 4**
  - LoRa® modulation
  - Spreading Factor (SF) = SF8
  - Bandwidth (BW) = 500 kHz
  - Coding Rate (CR) = 4/5
- **Bit Rate = 12500 bps**
- **Max Application Payload Size = 242 bytes**
  - Time On Air = 175 ms

LoRaWAN™ Network Protocol

LoRaWAN™ Channels

- **License free Sub-GHz Frequencies**
  - North America: 915 MHz Band
  - Upstream: 64 channels numbered 0 to 63, DR0 to DR3
  - Upstream: 8 channels numbered 64 to 71, DR4
  - Downstream: 8 channels numbered 0 to 7, DR8 to DR13



LoRaWAN™ Network Protocol

LoRaWAN™ Channels

- **License free Sub-GHz Frequencies**
  - Argentina: 915-928 MHz Band
  - Upstream: 64 channels numbered 0 to 63, DR0 to DR3
  - Upstream: 8 channels numbered 64 to 71, DR4
  - Downstream: 8 channels numbered 0 to 7, DR8 to DR13

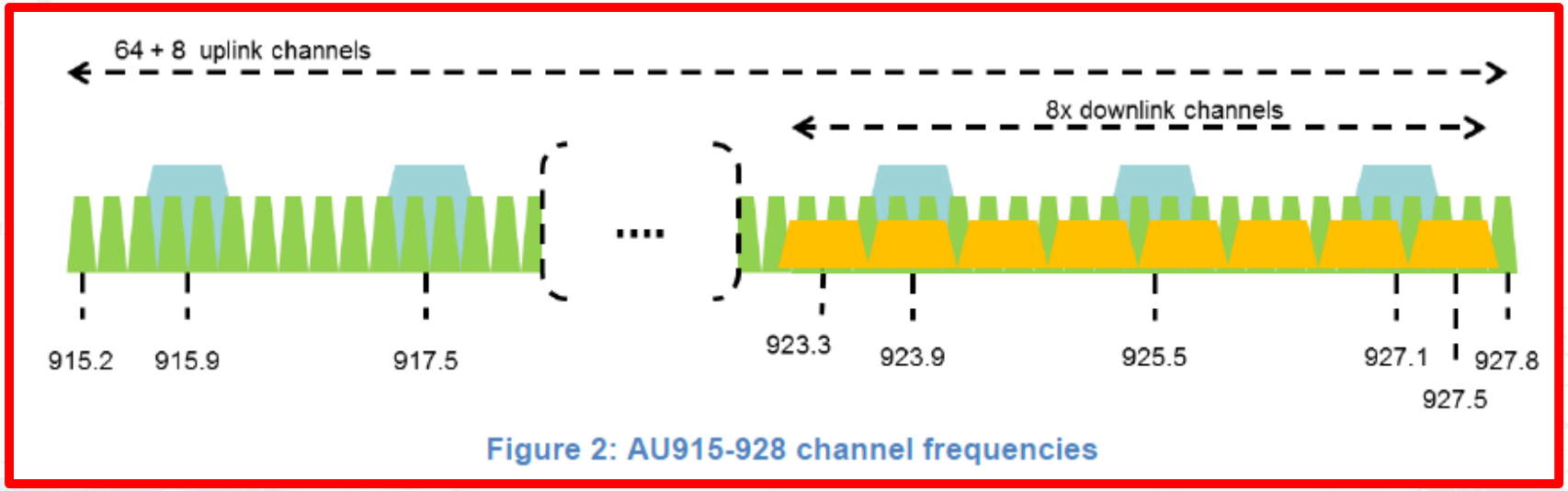


Figure 2: AU915-928 channel frequencies

Sub-Agenda

- **LoRaWAN™ Network Protocol**
  - LoRa® Technology Modulation
  - How does LoRaWAN™ Technology Work?
  - End-Device Classes
  - End-Device Activation (Joining)
  - Security
  - End-Device Data Communication (Class A)
  - Adaptive Data Rate (ADR)

LoRaWAN™ Network Protocol

What is LoRaWAN™ Network Protocol?

- **Low Power Wide Area Network (LPWAN)**
  - Bidirectional
  - Simple Star Network Topology
  - Low data rate
  - Low cost
  - Long battery life
- **Ideal for:**
  - Internet of Things (IoT)
  - Machine-to-Machine (M2M)
  - Industrial Automation
  - Low Power Applications
  - Battery Operated Sensors
  - Smart City
  - Smart Meter
  - Smart Agriculture

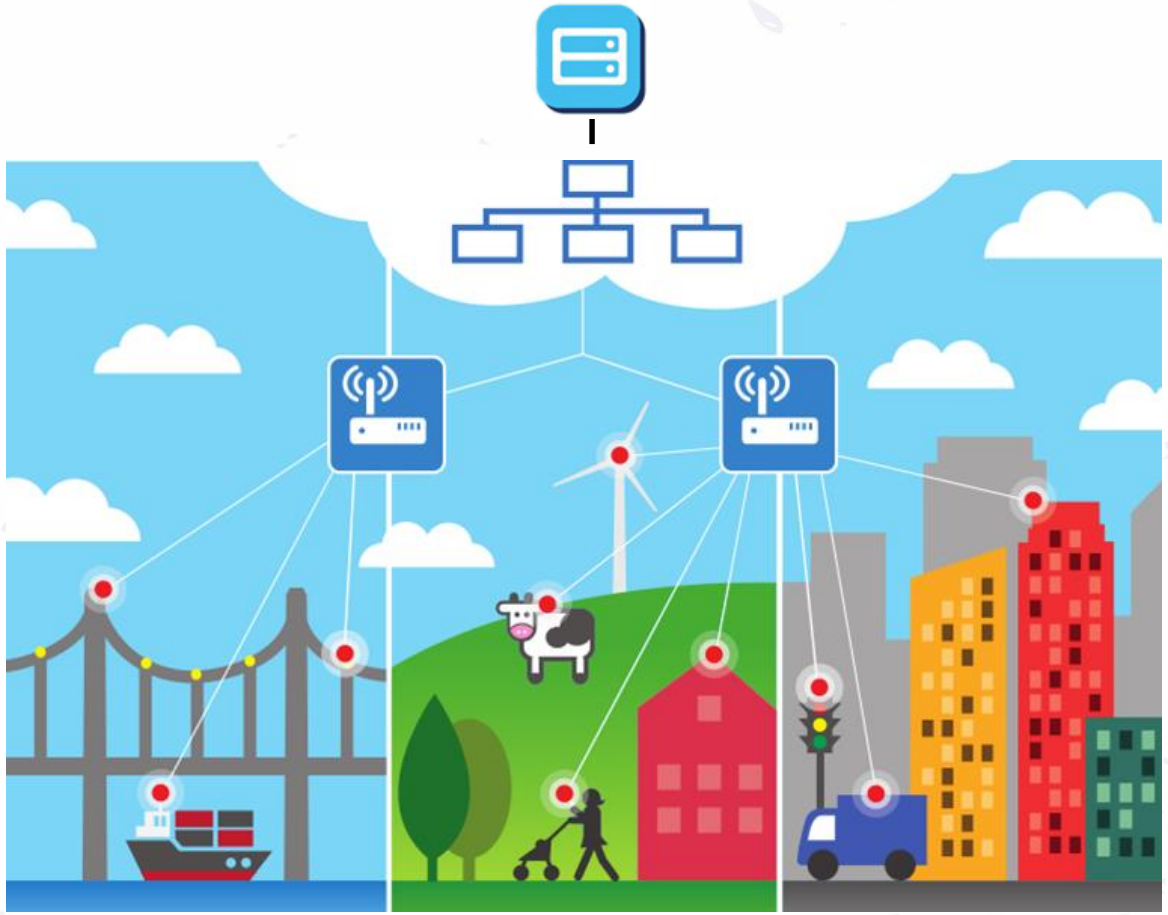
Enables simpler network architecture:
- ***No repeaters***
- ***No mesh routing complexity***

http://lora-alliance.org/What-Is-LoRa/Technology

Kahoot!

LoRaWAN™ Network Protocol

LoRaWAN™ Network



Application Server

Network Server

Gateways

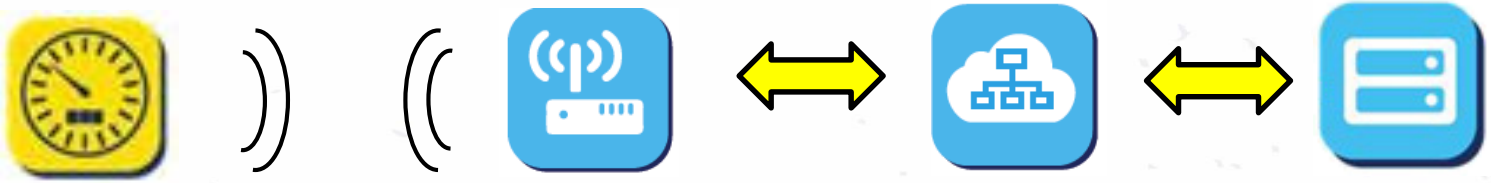
End-Devices

LoRaWAN™ Network Protocol

How does LoRaWAN™ Technology Work?

Physical Topology

End-Device | Gateway | Network Server | Application Server



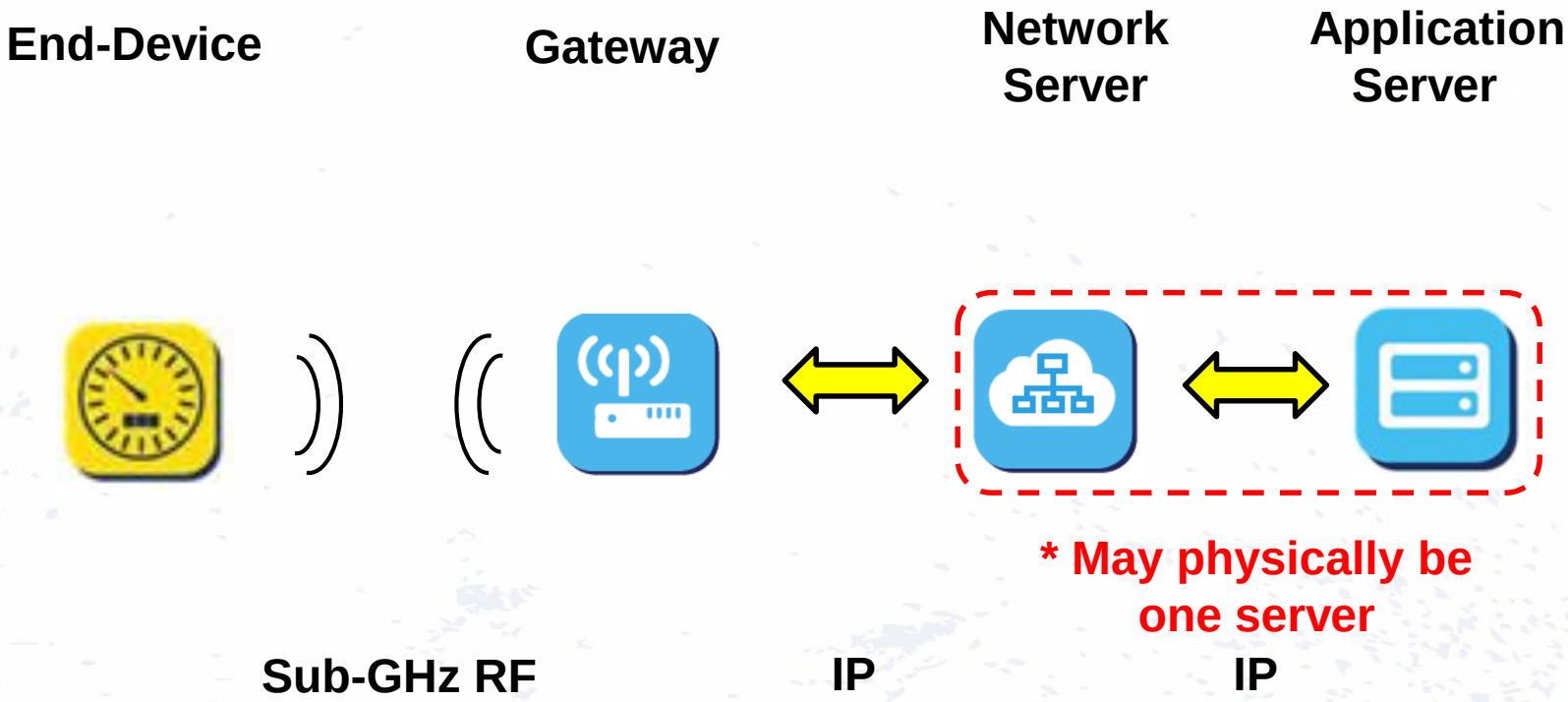
Sub-GHz RF | IP | IP

Advanced Network Topology

LoRaWAN™ Network Protocol

How does LoRaWAN™ Technology Work?

Physical Topology

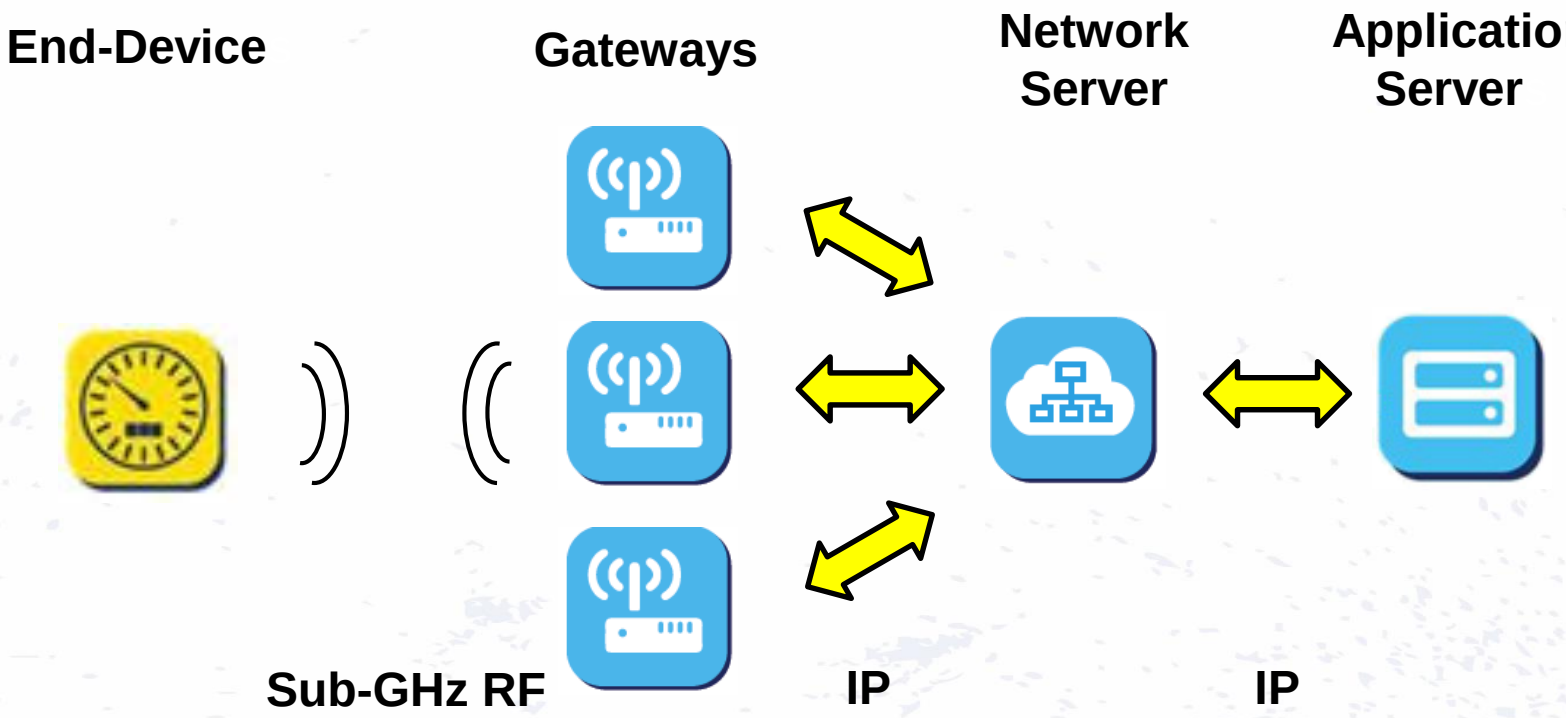


Advanced Network Topology

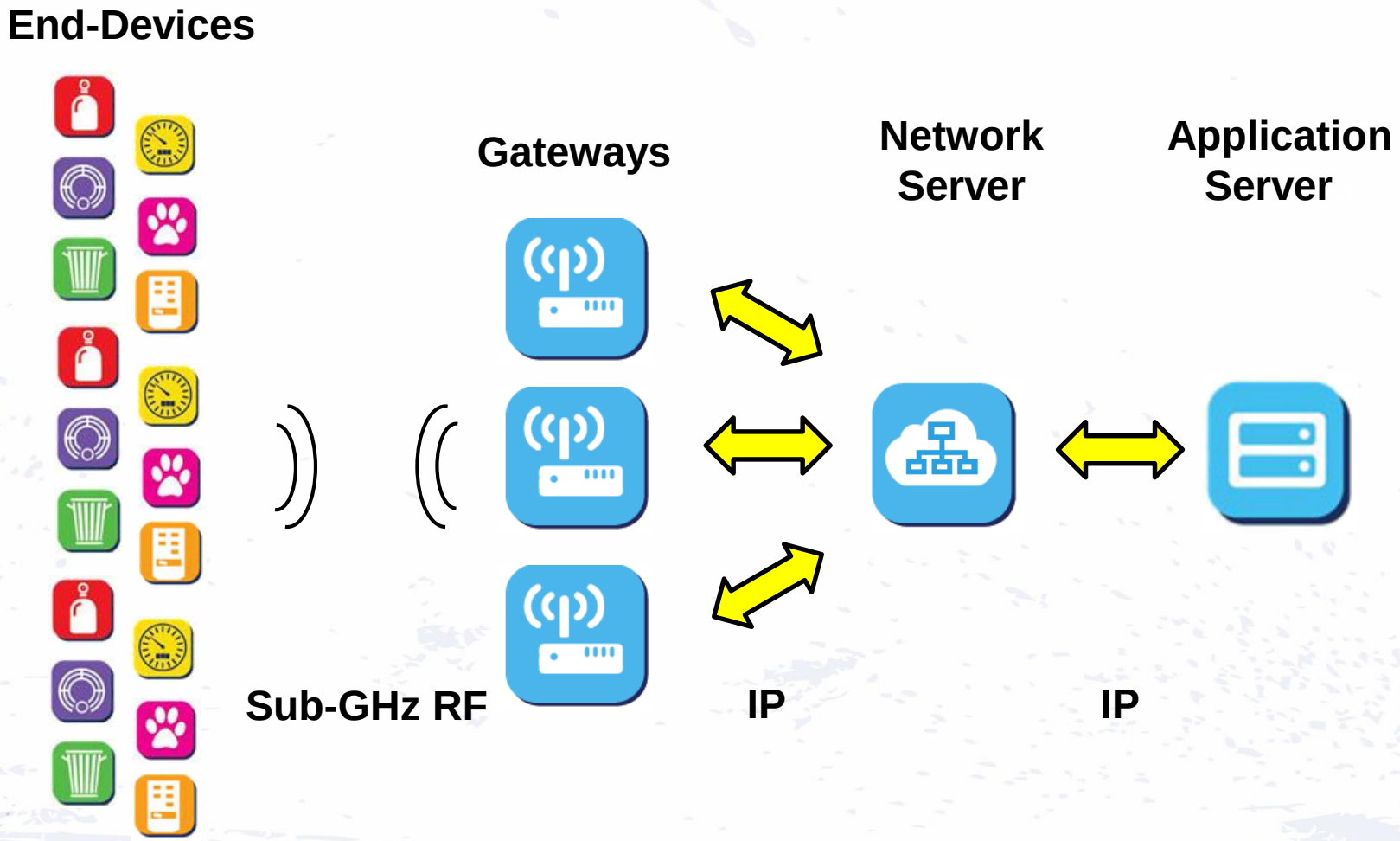
LoRaWAN™ Network Protocol

How does LoRaWAN™ Technology Work?

Physical Topology



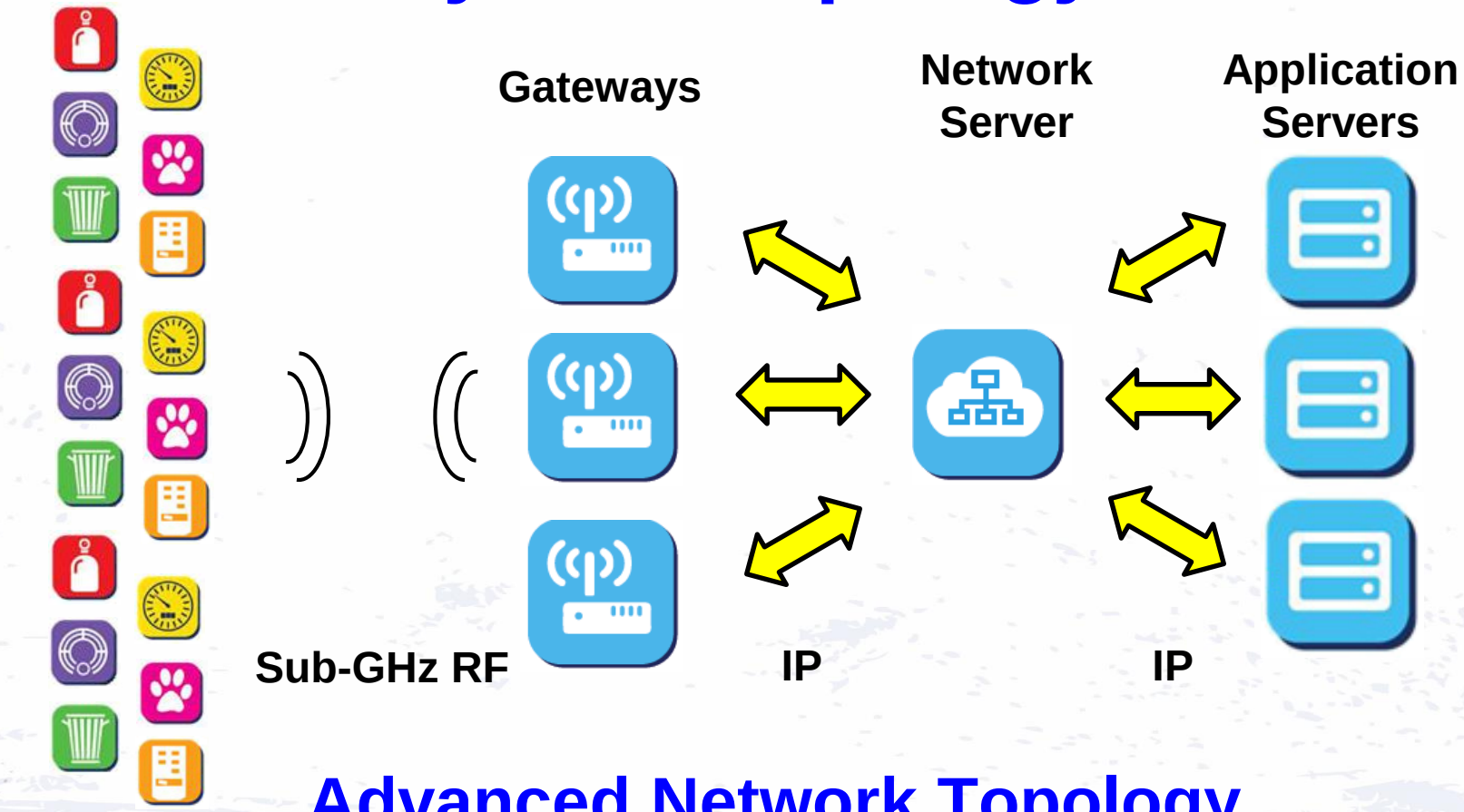
Advanced Network Topology



LoRaWAN™ Network Protocol

How does LoRaWAN™ Technology Work?

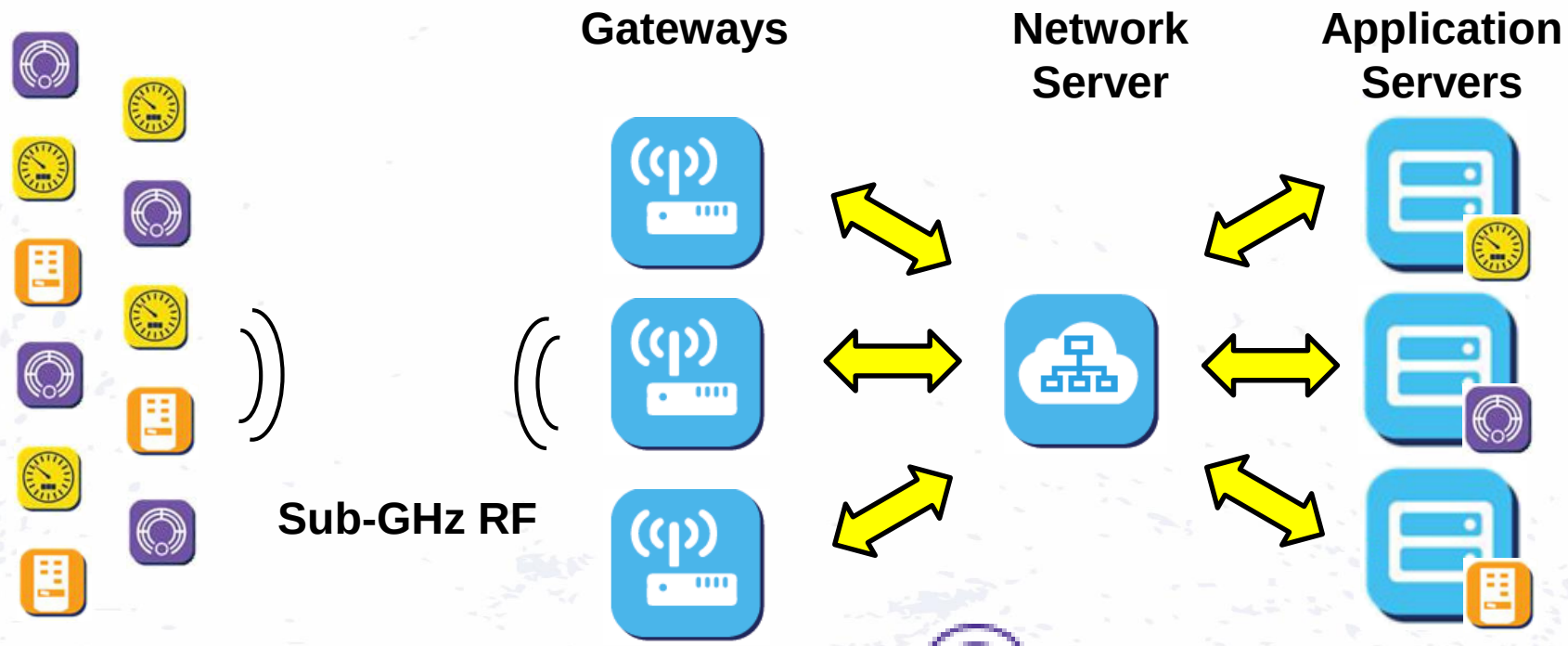
Physical Topology

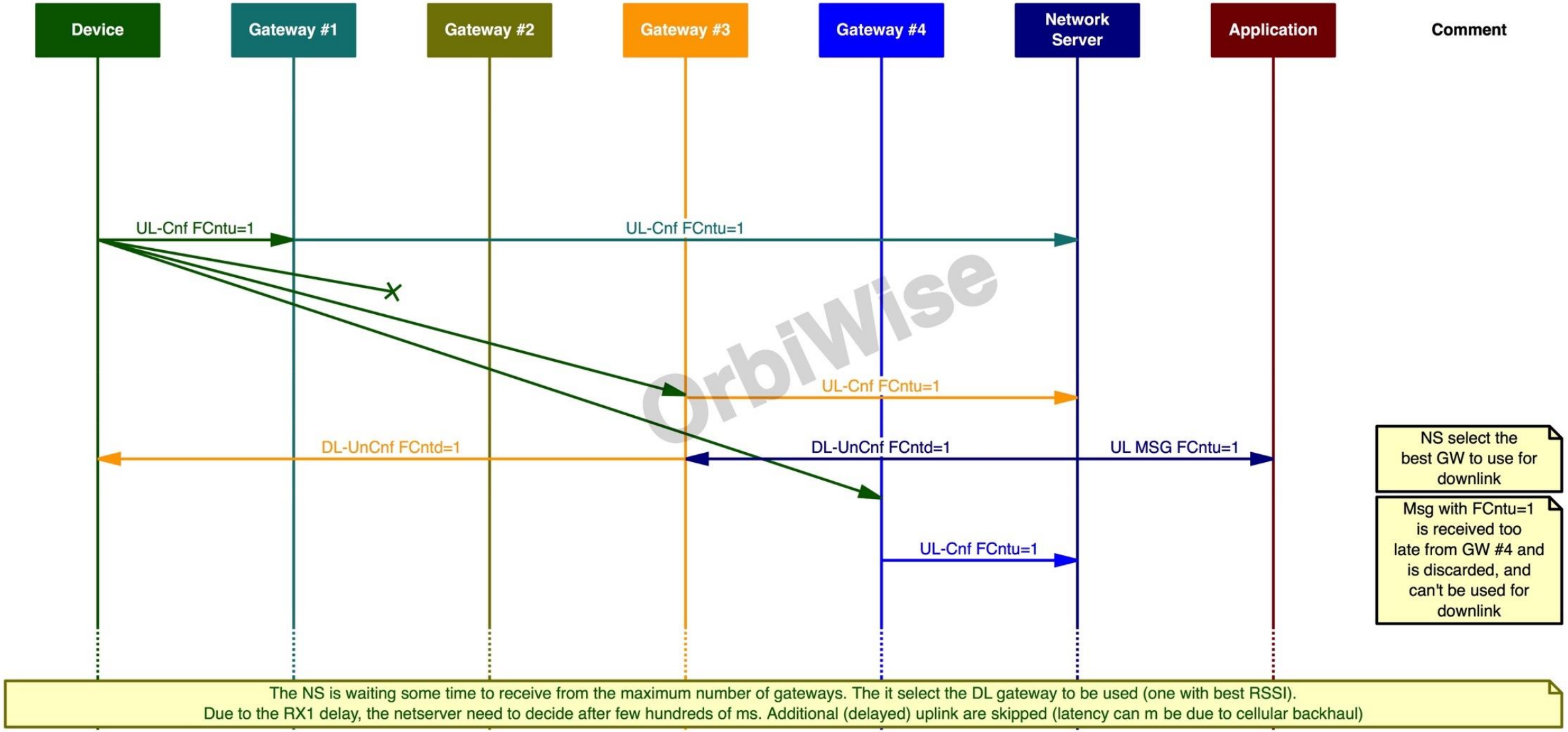


Advanced Network Topology

LoRaWAN™ Network Protocol

Multiple Application Servers Example

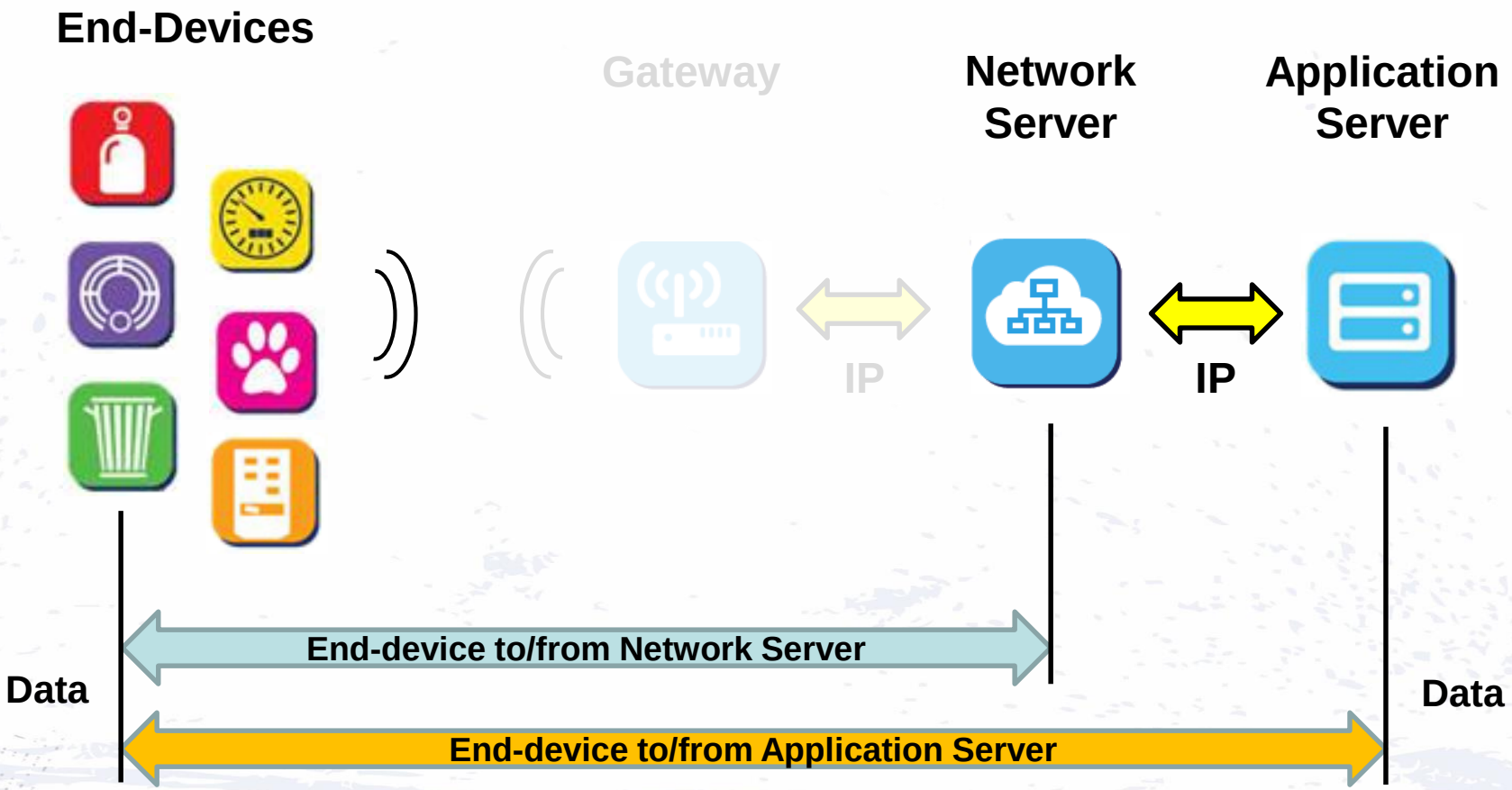




LoRaWAN™ Network Protocol

How does LoRaWAN™ Technology Work?

Logical Data Flow (Programmer's Model)



Sub-Agenda

- **LoRaWAN™ Network Protocol**
  - Lora® Technology Modulation
  - How does LoRaWAN™ Technology Work?
  - End-Device Classes
  - End-Device Activation (Joining)
  - Security
  - End-Device Data Communication (Class A)
  - Adaptive Data Rate (ADR)

LoRaWAN™ Network Protocol

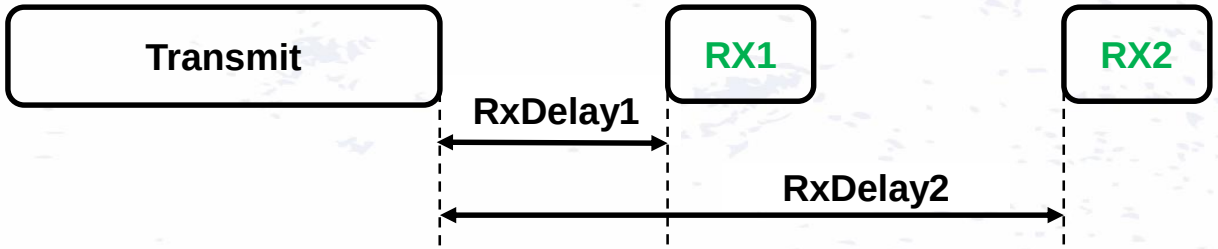
End-Device Classes

- **Each end-device class has different behavior depending on the choice of optimization:**
  - Battery Powered – Class A
  - Low Latency – Class B
  - No Latency – Class C

LoRaWAN™ Network Protocol

End-Device Classes

- **Battery Powered – Class A**
  - Bidirectional communications
  - Unicast messages
  - Small payloads
  - Long intervals
  - End-device initiates communication (uplink)
  - Server communicates with end-device (downlink) during predetermined response windows:



LoRaWAN™ Network Protocol

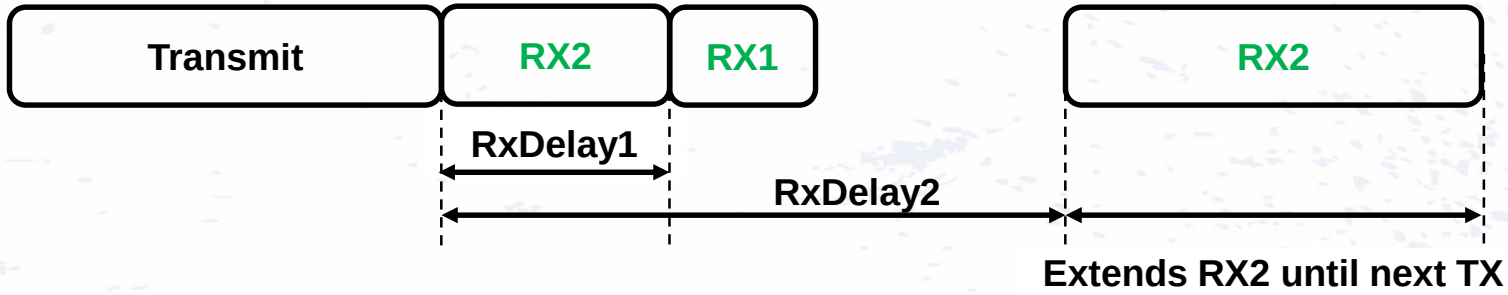
End-Device Classes

- **Battery Powered – Class A**
  - **Pros**
    - Lowest power consumption = longest battery life
  - **Cons**
    - Long latency
- **Examples**
  - Battery powered sensors

LoRaWAN™ Overview

End-Device Classes

- **No Latency – Class C**
  - Bidirectional communications
  - Unicast and Multicast messages
  - Small payloads
  - Server can initiate transmission at any time
  - End-device is constantly receiving



LoRaWAN™ Network Protocol

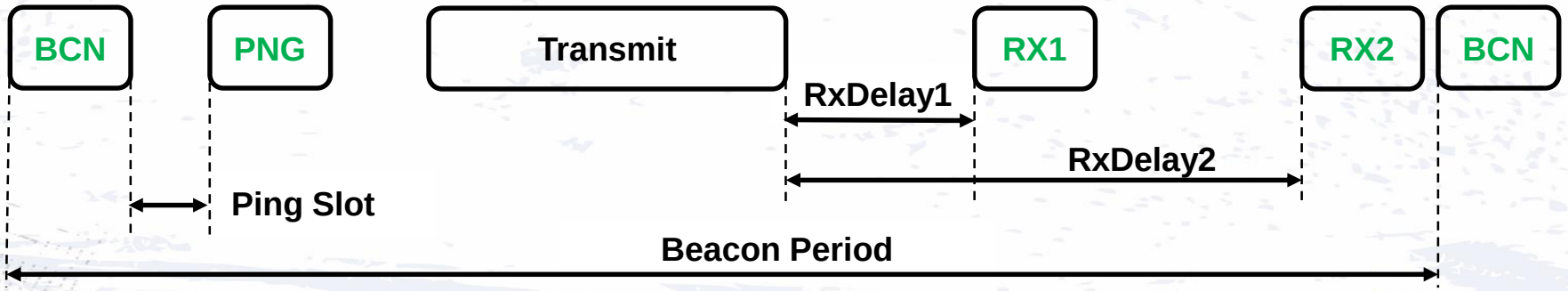
End-Device Classes

- **No Latency – Class C**
  - **Pros**
    - Lowest receive latency
    - End-device has continuous receive window
  - **Cons**
    - Highest power consumption (expect end-device to be mains powered)
- **Examples**
  - Mains power low-latency actuator end-device

LoRaWAN™ Network Protocol

End-Device Classes

- **Low Latency – Class B**
  - Bidirectional with scheduled receive slots
  - Unicast and Multicast messages
  - Small payloads
  - Long intervals
  - Periodic beacon from gateway
  - Extra receive window (ping slot)
  - Server can initiate transmission at fixed intervals



LoRaWAN™ Network Protocol

End-Device Classes

- **Low Latency – Class B**
  - **Pros**
    - Deterministic latency
  - **Cons**
    - Higher power consumption
- **Examples**
  - Battery powered actuator end-device

Sub-Agenda

- **LoRaWAN™ Network Protocol**
  - LoRa® Technology Modulation
  - How does LoRaWAN™ Technology Work?
  - End-Device Classes
  - End-Device Activation (Joining)
  - Security
  - End-Device Data Communication (Class A)
  - Adaptive Data Rate (ADR)

Kahoot!

LoRaWAN™ Network Protocol

End-Device Activation (Joining)

- **Before an end-device can communicate on the LoRaWAN network, it must be activated**
- **The following information is required:**
  - Device Address (DevAddr)
  - Network Session Key (NwkSKey)
  - Application Session Key (AppSKey)

Let's look at each of these in detail…

LoRaWAN™ Network Protocol

End-Device Activation (Joining)

- **Device Address (DevAddr)**
  - 32-bit identifier
  - Unique within the network
  - Present in each data frame
  - Shared between End-device, Network Server, and Application Server
- **Differentiates nodes within the network, allowing the network to use the correct encryption keys and properly interpret the data**

LoRaWAN™ Network Protocol

End-Device Activation (Joining)

- **Network Session Key (NwkSKey)**
  - 128-bit AES encryption key
  - Unique per end-device
  - Shared between end-device and Network Server
- **Provides message integrity for the communication**
- **Provides security for end-device to Network Server communication**

LoRaWAN™ Network Protocol

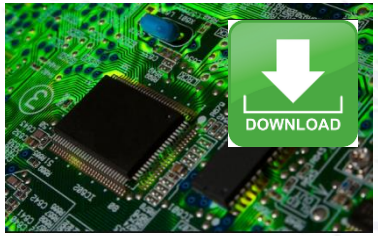
End-Device Activation (Joining)

- **Application Session Key (AppSKey)**
  - 128-bit AES encryption key
  - Unique per end-device
  - Shared between end-device and Application Server
  - Used to encrypt / decrypt application data messages
- **Provides security for application payload**

LoRaWAN™ Network Protocol

End-Device Activation (Joining)

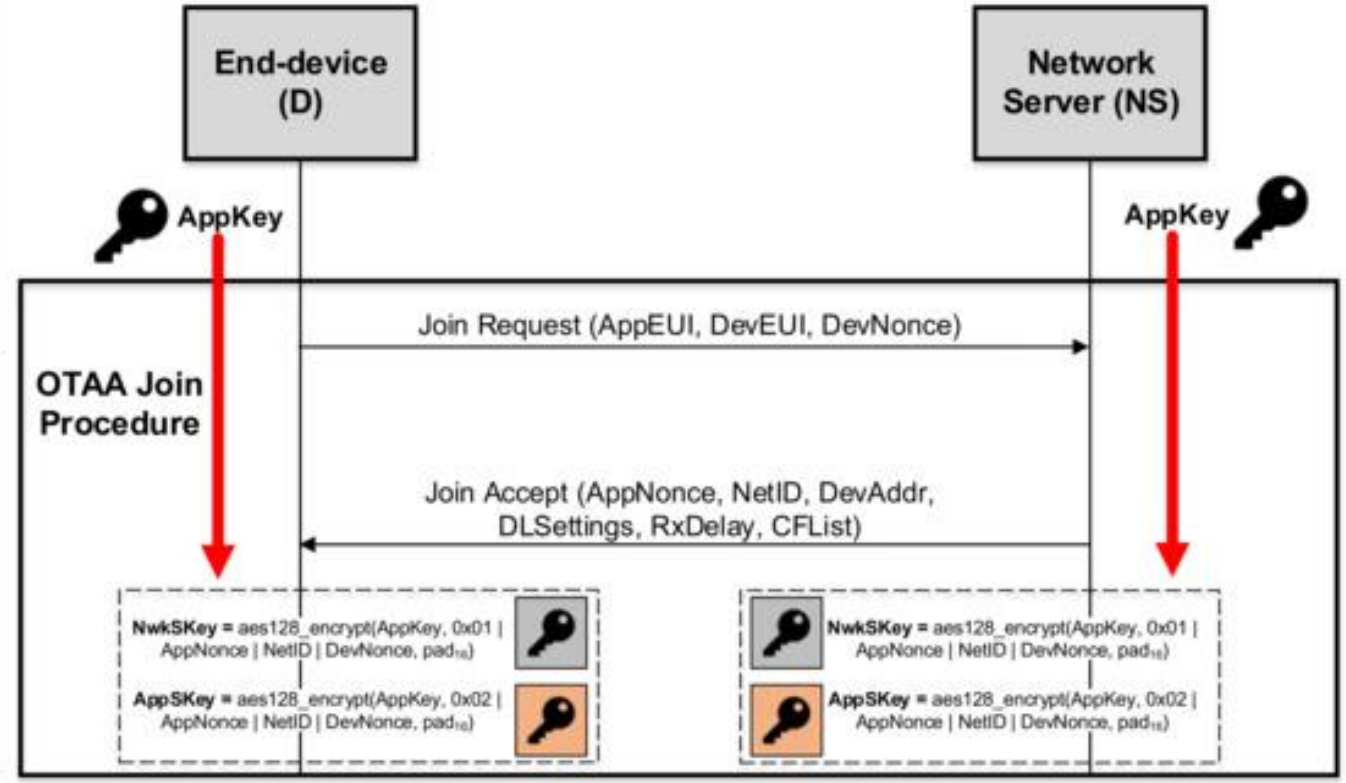
- **To exchange this information, two activation methods are available:**

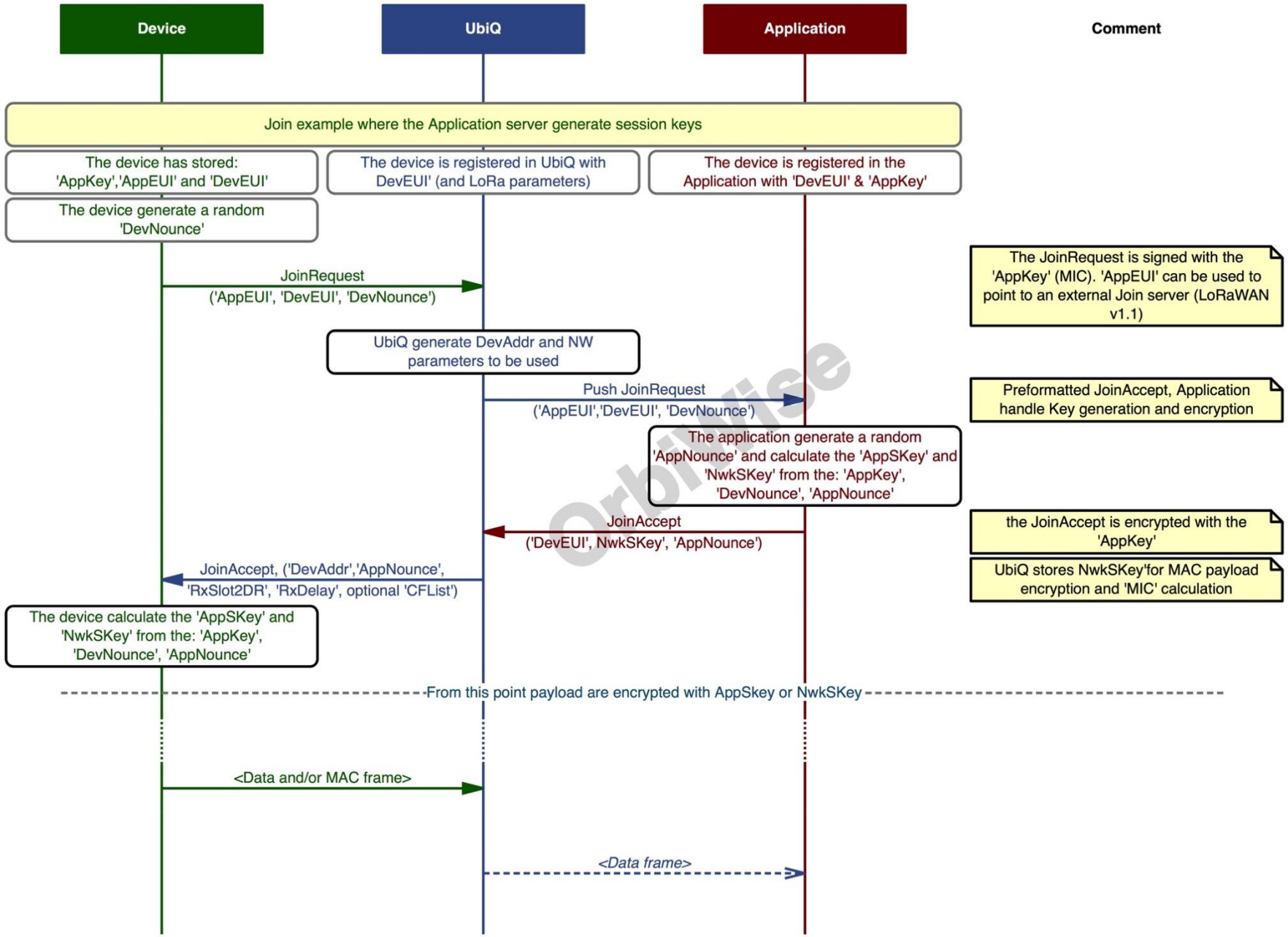
Over-the-Air Activation (OTAA)	Activation By Personalization (ABP)
Based on Globally Unique Identifier Over the air message handshaking 	Shared keys stored at production time Locked to a specific network 

LoRaWAN™ Network Protocol

End-Device Activation (Joining)

Over-the-Air-Activation (OTAA)





LoRaWAN™ Network Protocol

End-Device Activation (Joining)

Over-the-Air-Activation (OTAA)

- End-device transmits Join Request to application server containing:
  - Globally unique end-device identifier (DevEUI)
  - Application identifier (AppEUI)
  - Authentication with Application key (AppKey)
- End-device receives Join Accept from application server

(continued…)

LoRaWAN™ Network Protocol

End-Device Activation (Joining)

Over-the-Air-Activation (OTAA)

- End-device authenticates Join Accept
- End-device decrypts Join Accept
- End-device extracts and stores Device Address (DevAddr)
- End-device derives:
  - Network Session Key (NwkSKey)
  - Application Session Key (AppSKey)

Security Keys

LoRaWAN™ Network Protocol

End-Device Activation (Joining)

Activation By Personalization (ABP)

- The following information is configured at production time:
  - Device Address (DevAddr)
  - Network Session Key (NwkSKey)
  - Application Session Key (AppSKey)
- No over the air handshaking
- Device is ready to communicate on the network without any additional procedure.
- Note that the end result is the same, the DevAddr and security keys are now known to the end-device

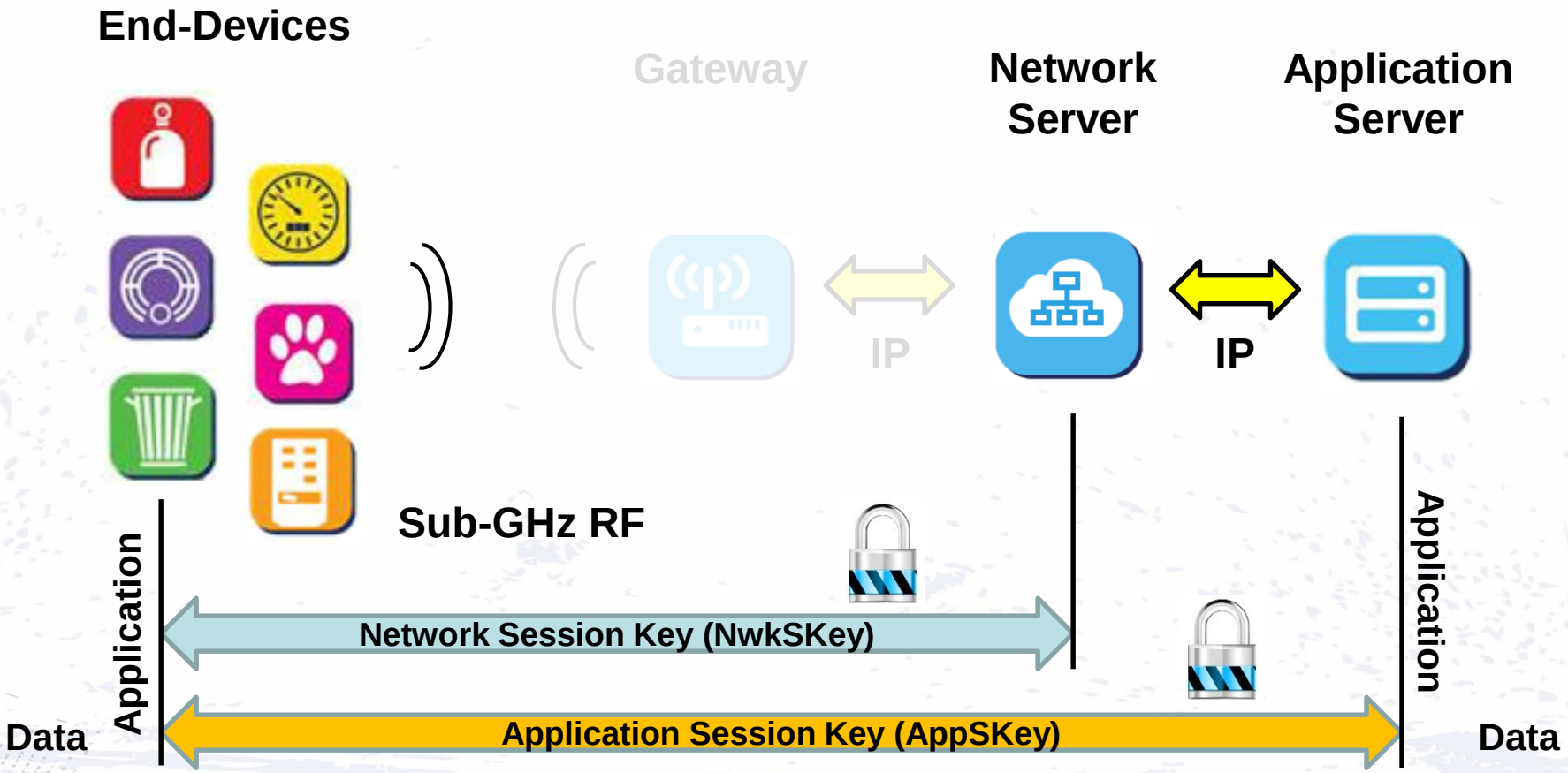
Sub-Agenda

- **LoRaWAN™ Network Protocol**
  - LoRa® Technology Modulation
  - How does LoRaWAN™ Technology Work?
  - End-Device Classes
  - End-Device Activation (Joining)
  - Security
  - End-Device Data Communication (Class A)
  - Adaptive Data Rate (ADR)

LoRaWAN™ Network Protocol

Security

Logical Data Flow (Programmer's Model)



LoRaWAN™ Network Protocol

Security

- **Based on 802.15.4 Security**
  - AES-128
- **Enhancement**
  - Network Session Key (NwkSKey)
  - Application Session Key (AppSKey)
  - Network Server authenticates Application Data
  - Network Server cannot decrypt Application Data

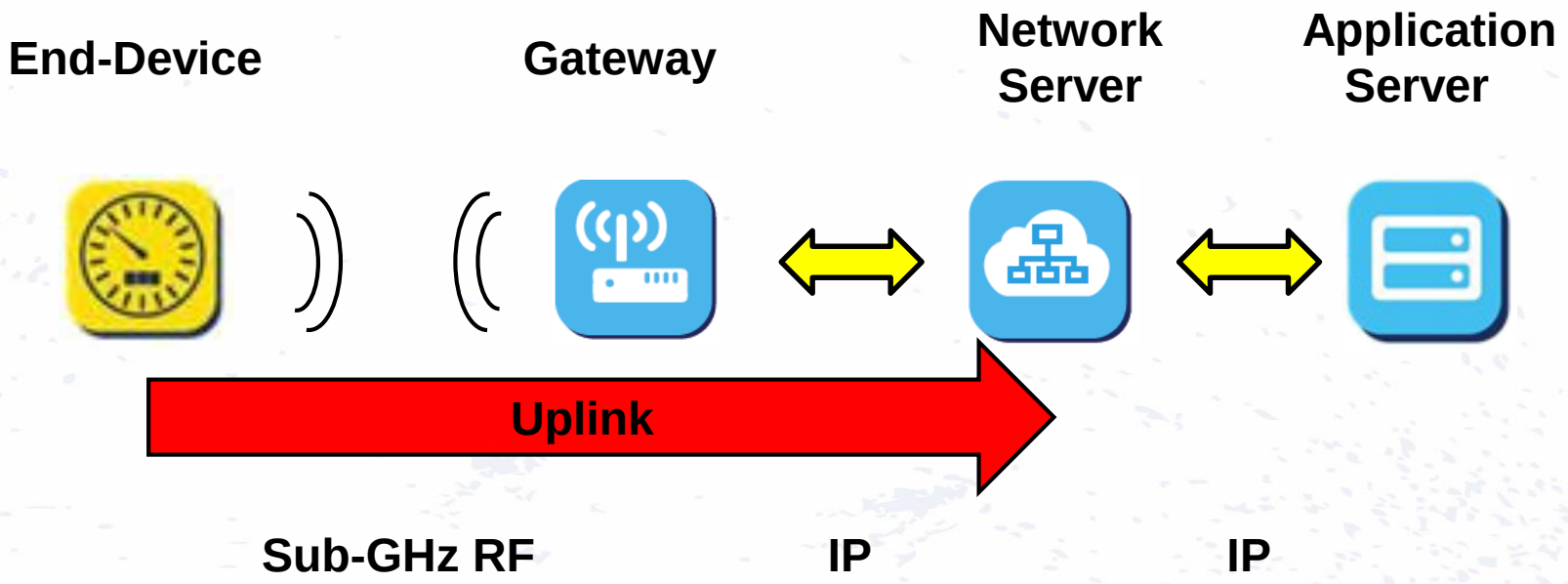
Sub-Agenda

- **LoRaWAN™ Network Protocol**
  - LoRa® Technology Modulation
  - How does LoRaWAN™ Technology Work?
  - End-Device Classes
  - End-Device Activation (Joining)
  - Security
  - End-Device Data Communication (Class A)
  - Adaptive Data Rate (ADR)

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

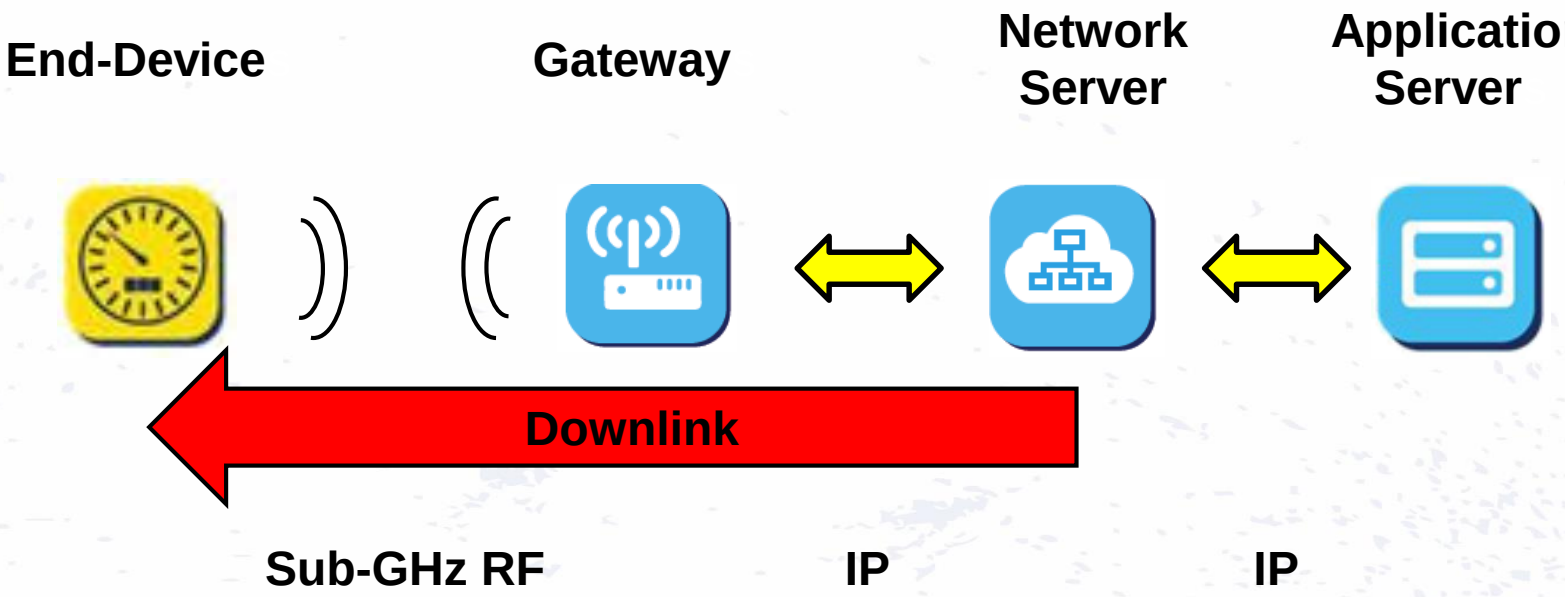
- **Uplink Message**
  - End-Device to Network Server relayed by one or many Gateways



LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

- **Downlink Message**
  - Sent by the Network Server to only one End-Device and is relayed by a single Gateway



LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Unconfirmed-Data Message

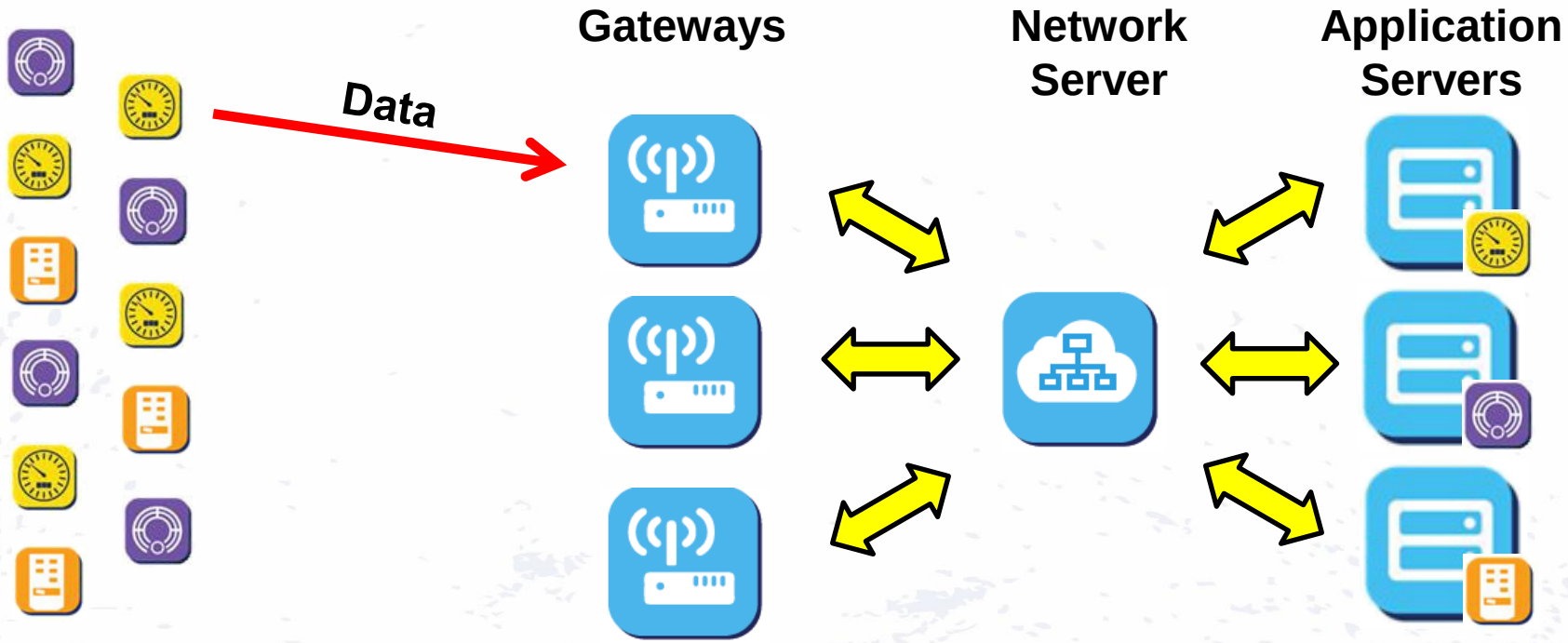
End-Device Data Message does not require an acknowledgement

Let's look at an example…

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Unconfirmed-Data Message

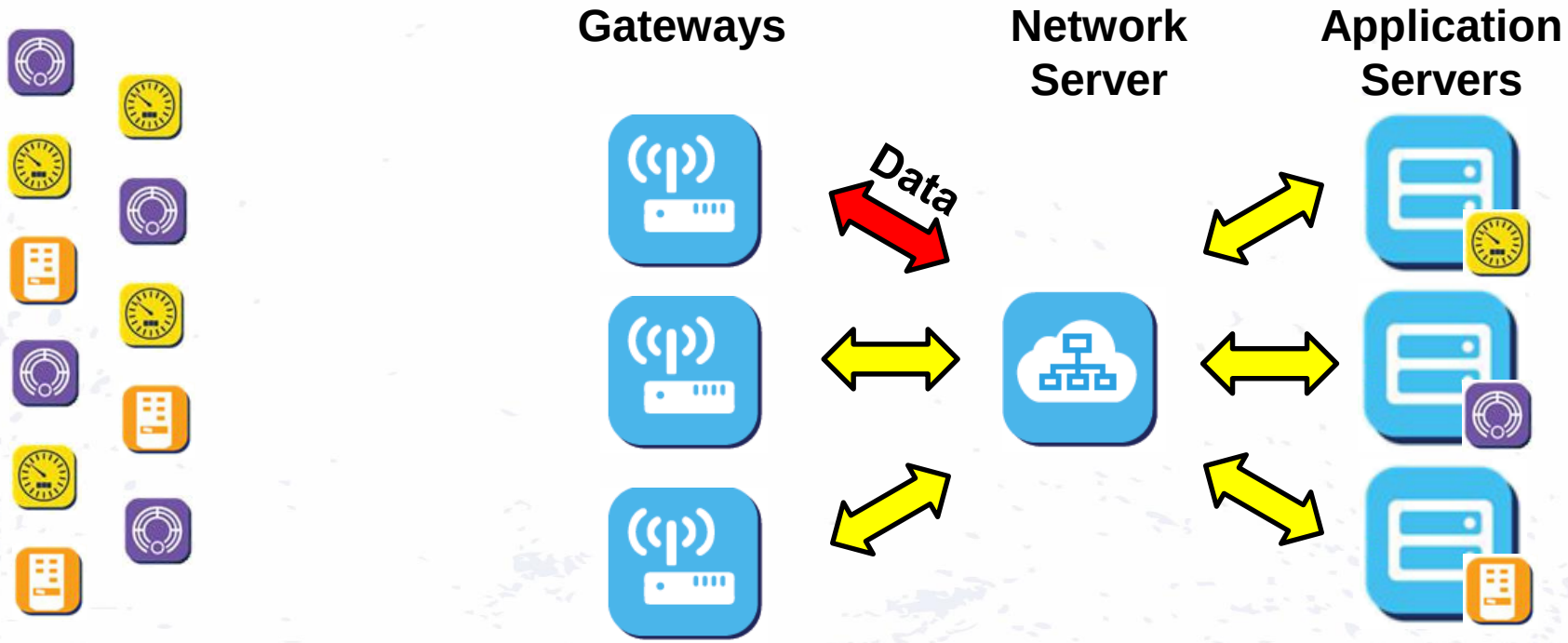


1. Electric meter transmits data

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Unconfirmed-Data Message

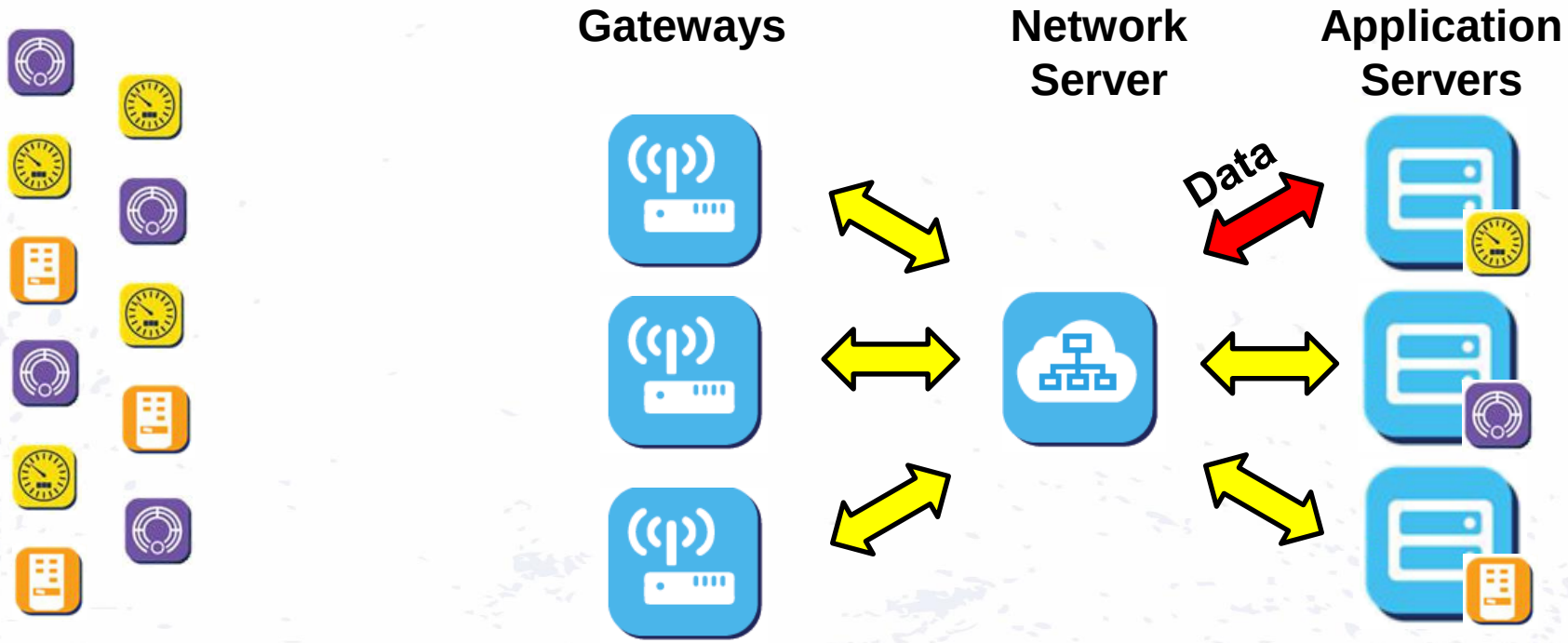


2. Gateway receives data and passes to Network Server

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Unconfirmed-Data Message

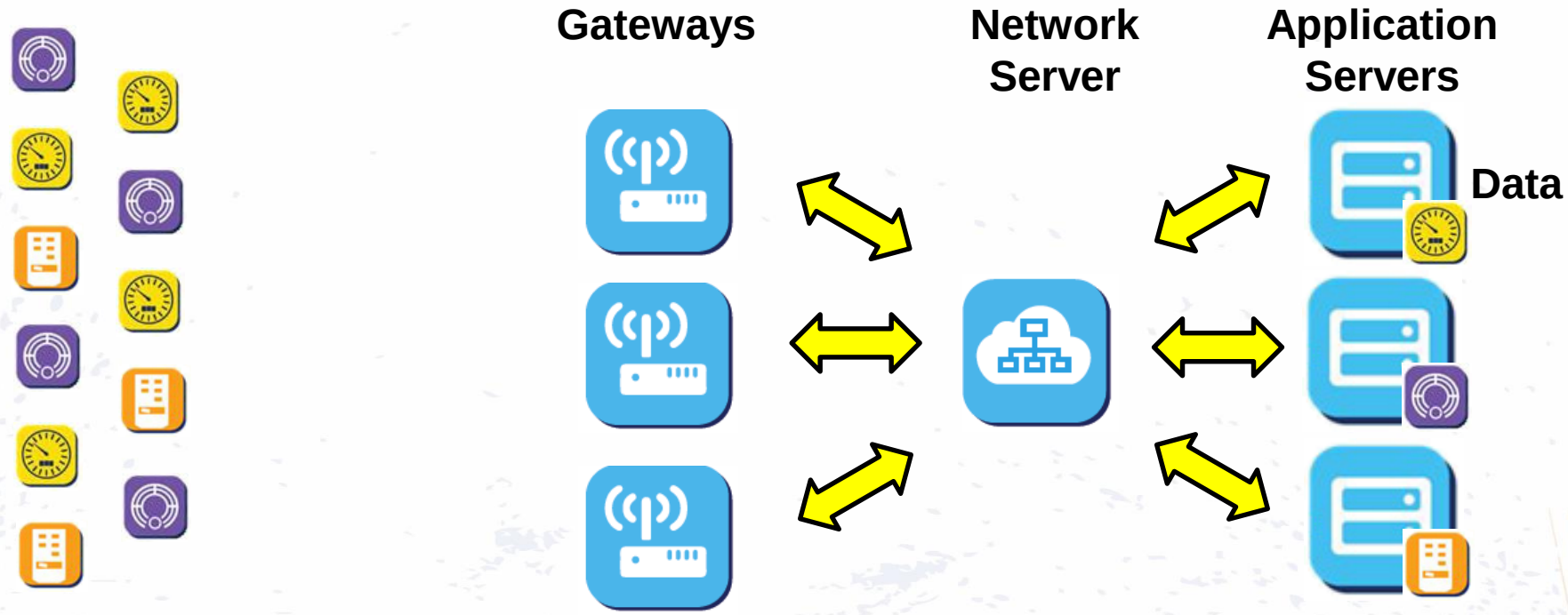


3. The Network Server authenticates data and passes it to Electric Meter Application Server

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Unconfirmed-Data Message



4. Electric Meter Application Server decrypts data

LoRaWAN™ Network Protocol

End-Device Data Communications

Confirmed-Data Message

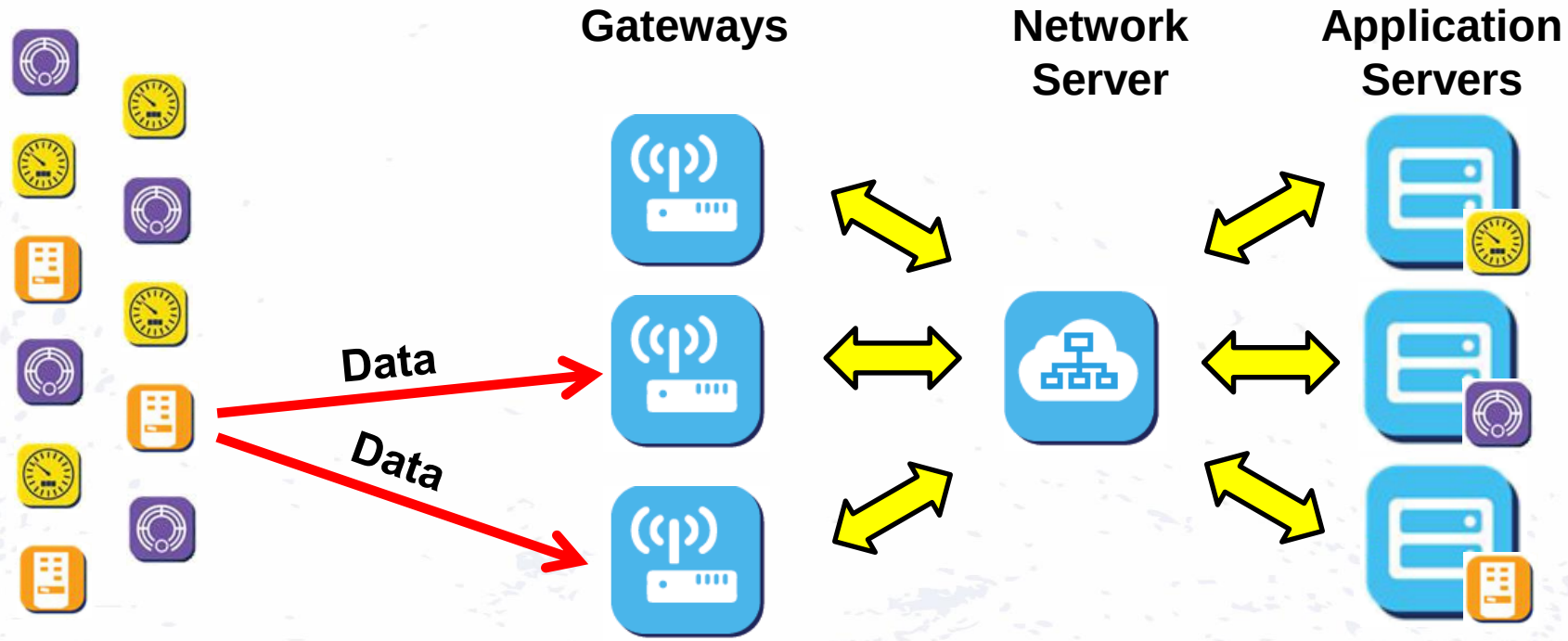
End-Device Data Message has to be acknowledged by the receiver

Let’s look at an example…

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Confirmed-Data Message

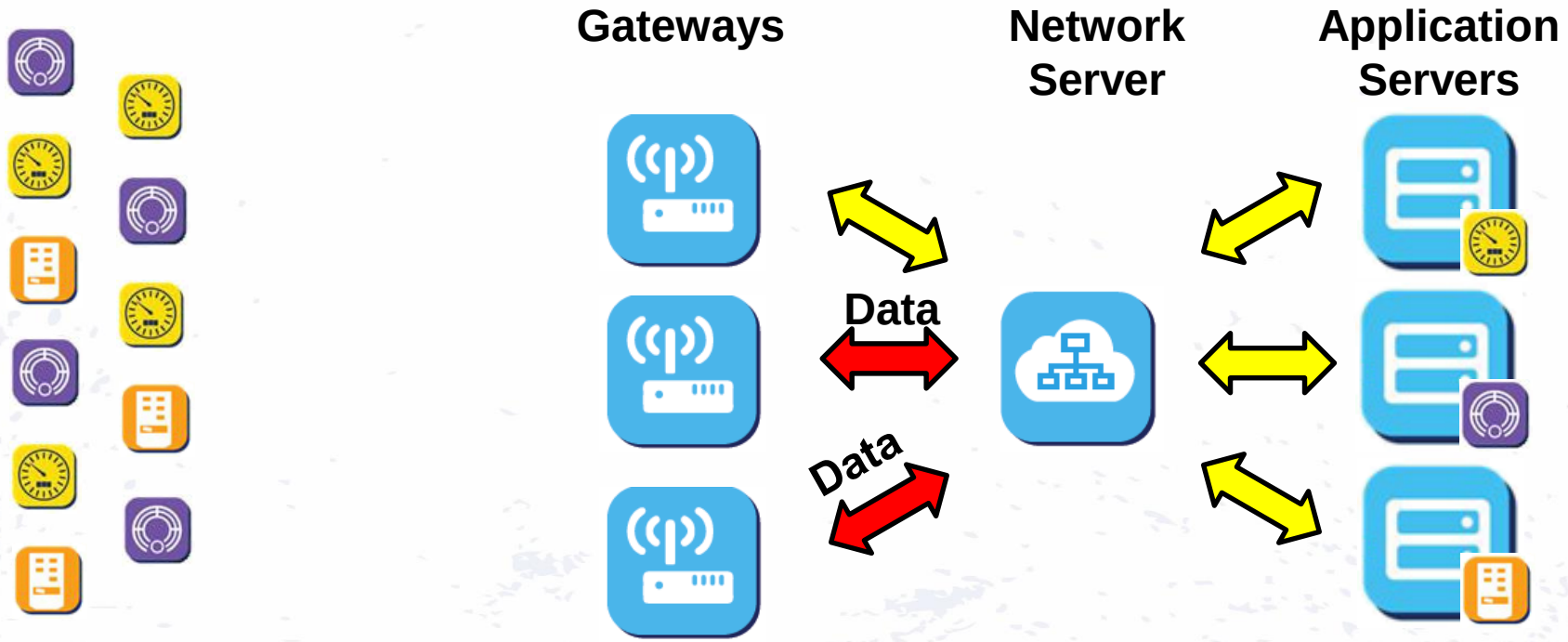


1. Vending Machine transmits data.
It is received by two Gateways.

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Confirmed-Data Message

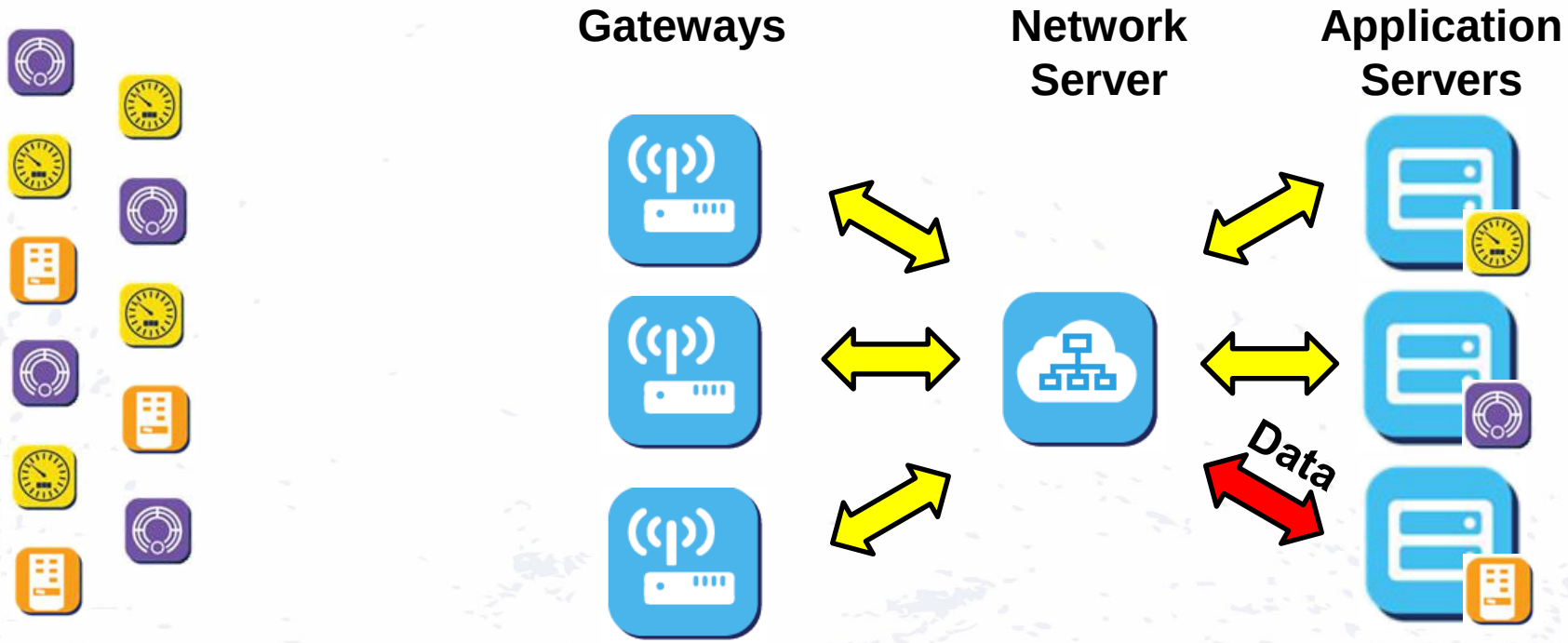


2. Both gateways "pass through" the data to the Network Server.

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Confirmed-Data Message

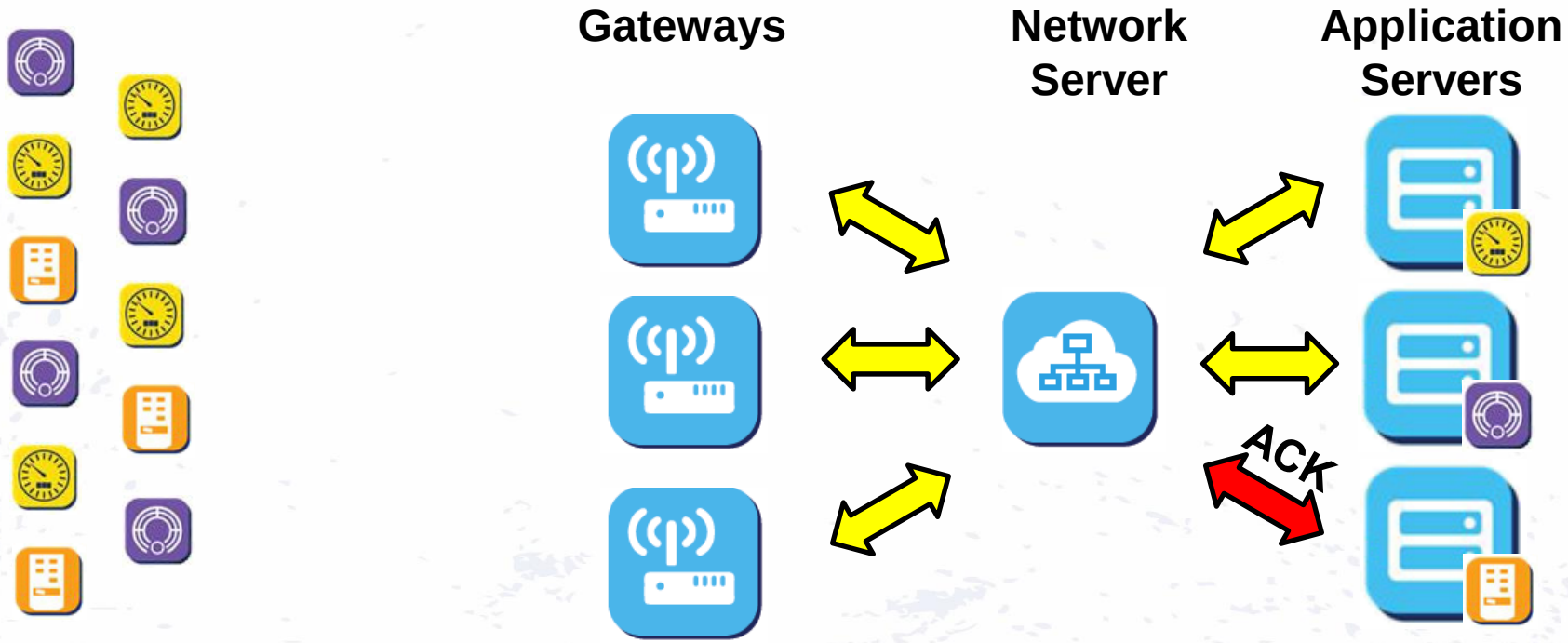


3. The Network Server forwards the data to the Vending Machine Applications Server

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Confirmed-Data Message

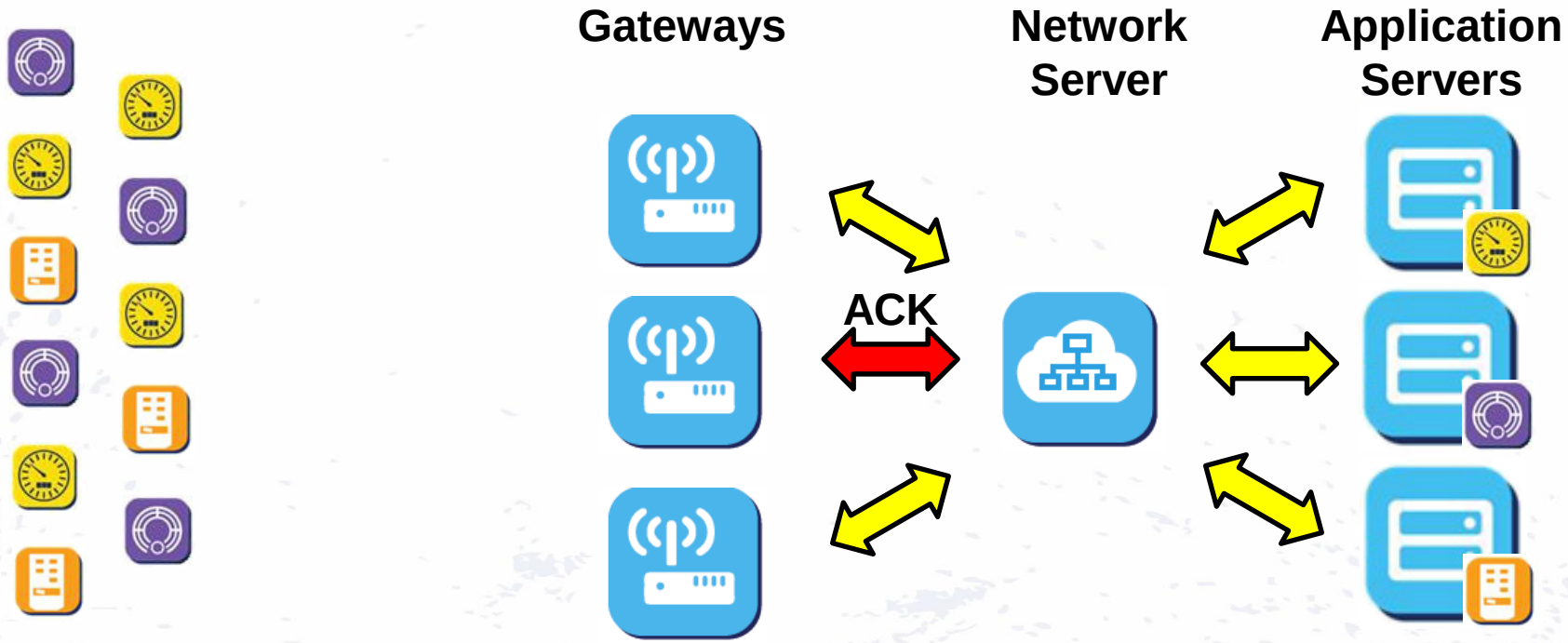


4. The Vending Machine Applications Server sends an acknowledgement

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Confirmed-Data Message

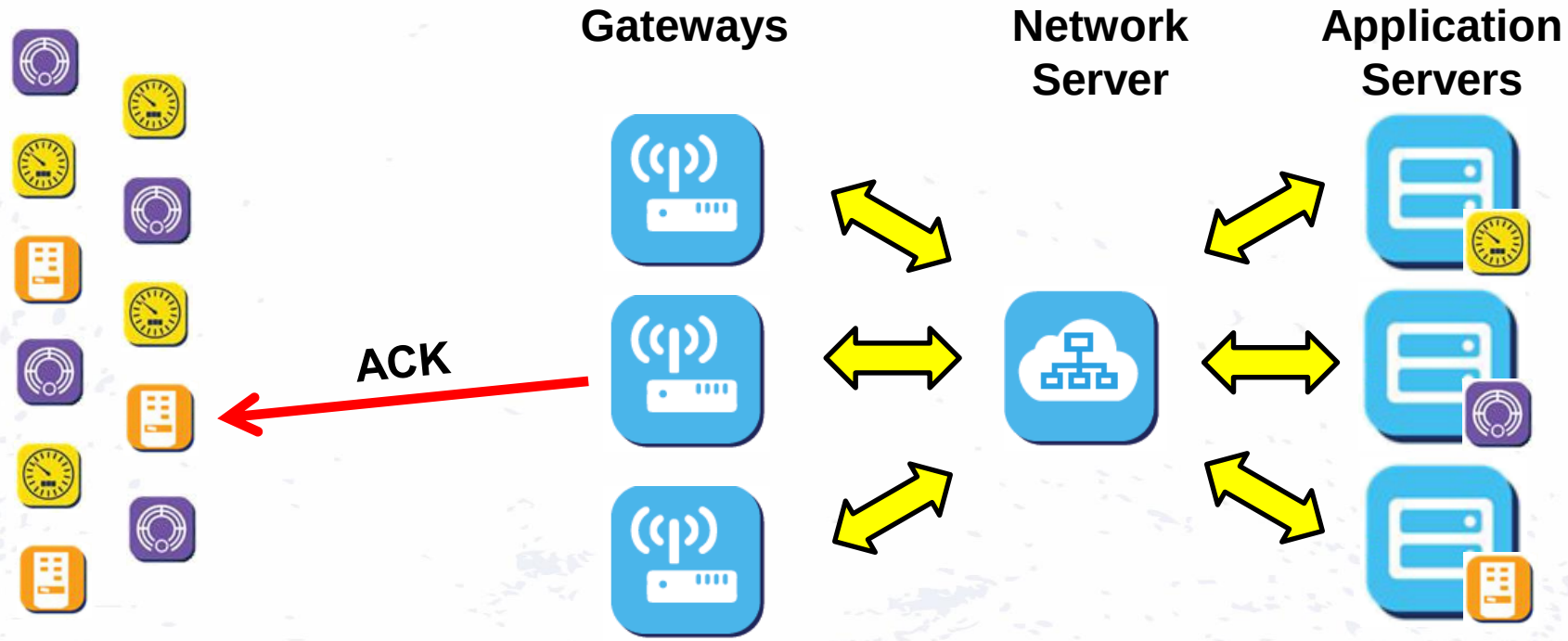


5. The Network Server selects the best path (gateway) to transmit the acknowledgement to the end-device.

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Confirmed-Data Message



6. The Gateway transmits the acknowledgement to the end-device

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Application Server Data Message

If the Application Server has a Data Message for the End-Device…

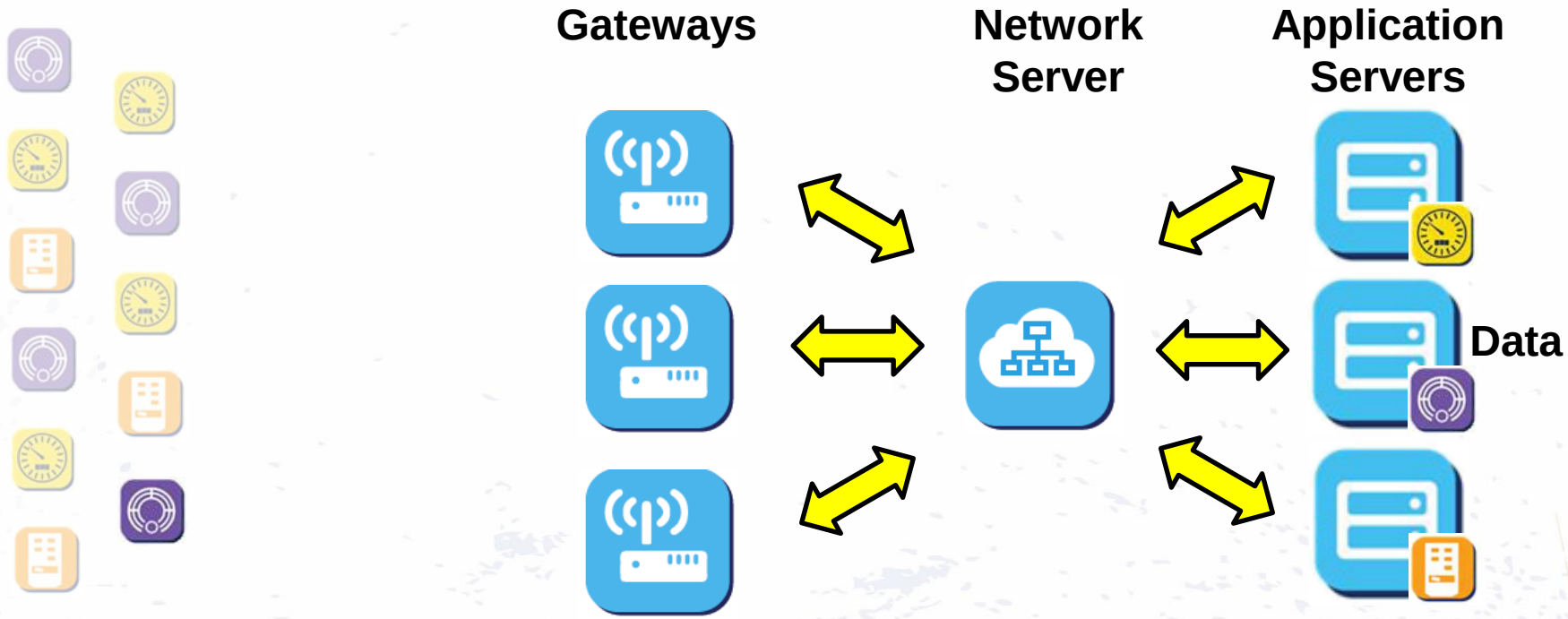
… the Application Server has to wait until the End-Device initiates a transmission.

Let's look at an example…

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Application Server Data Message

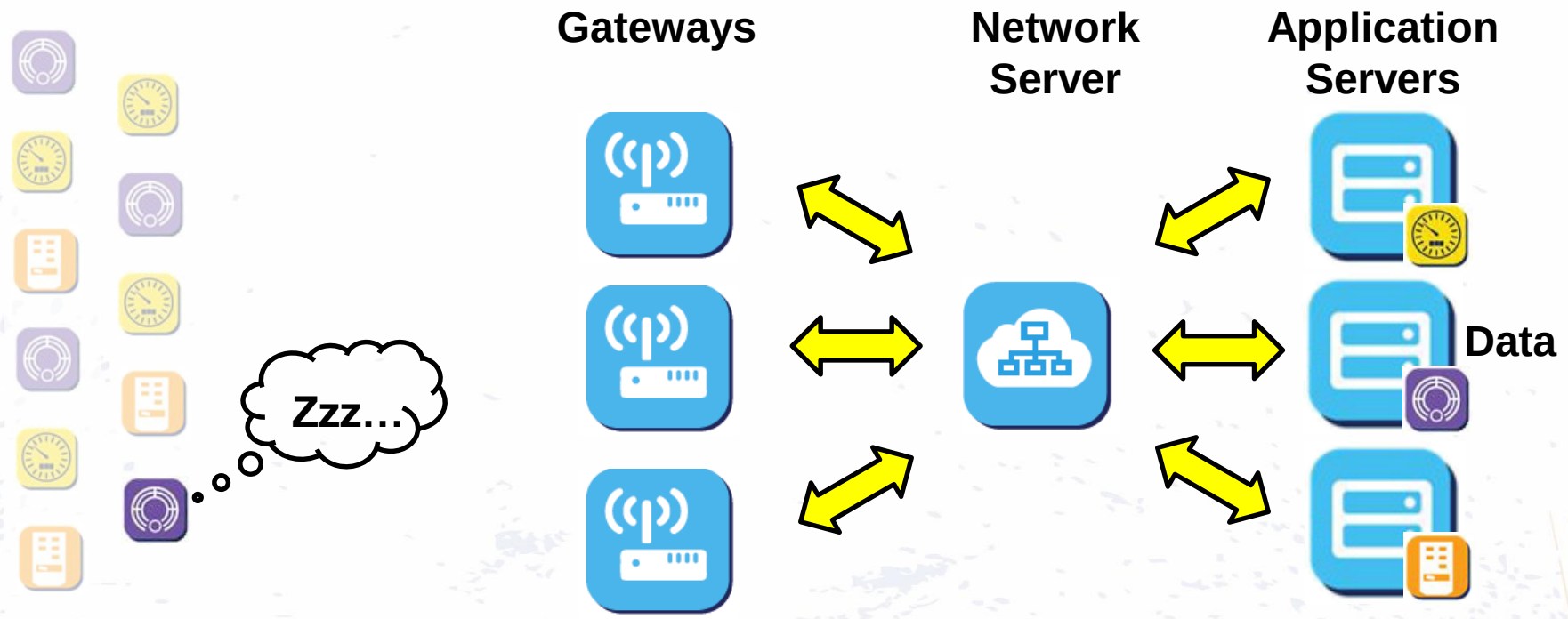


1. The Smoke Detector Application Server has Data for the highlighted Smoke Detector

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Application Server Data Message

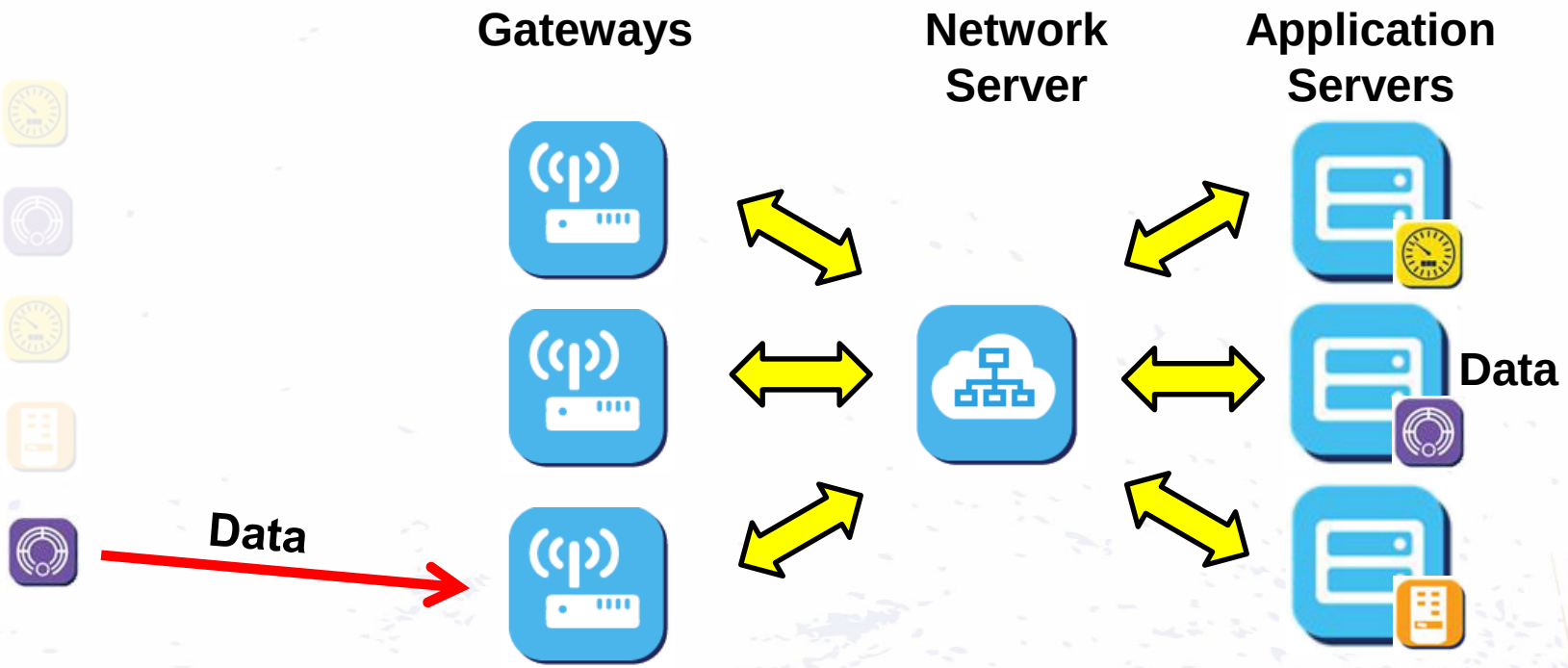


2. However, it has to wait until the Smoke Detector wakes up and transmits a Data Message

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Application Server Data Message

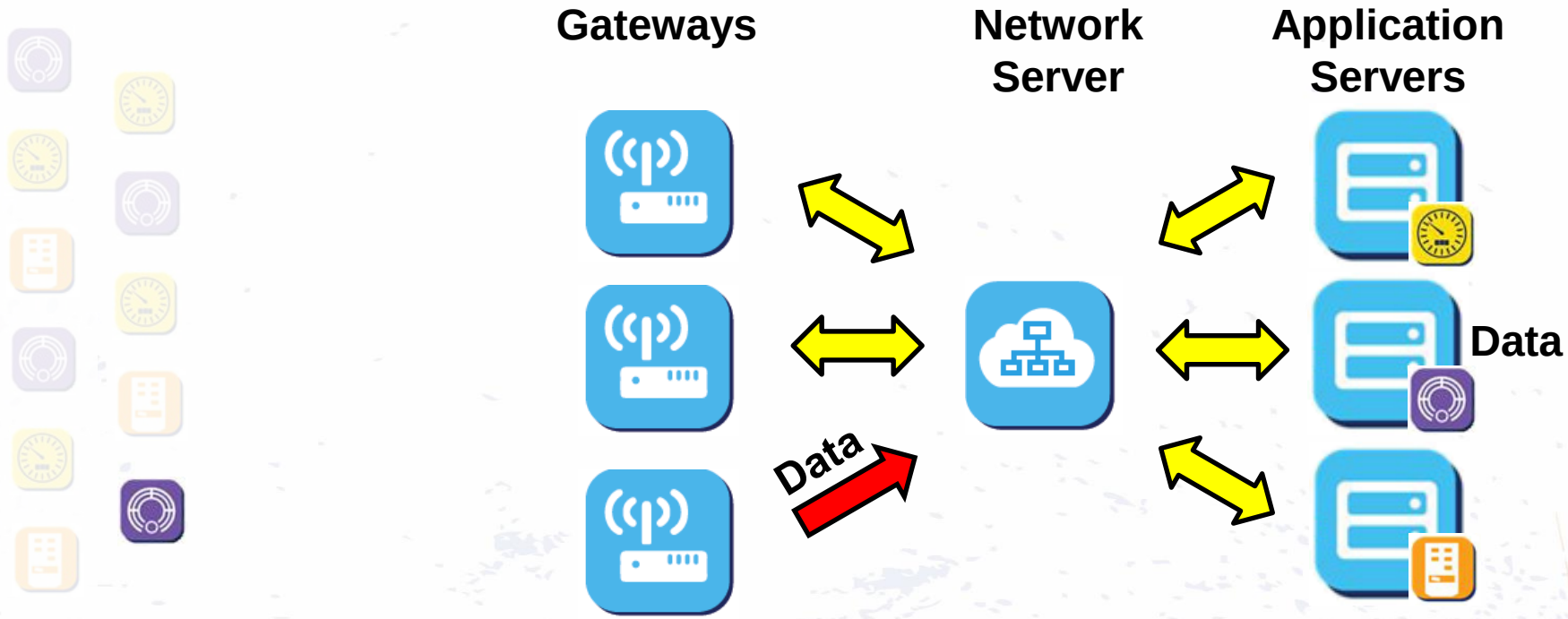


3. When the Smoke Detect transmits, the Data Message moves Upstream

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Application Server Data Message

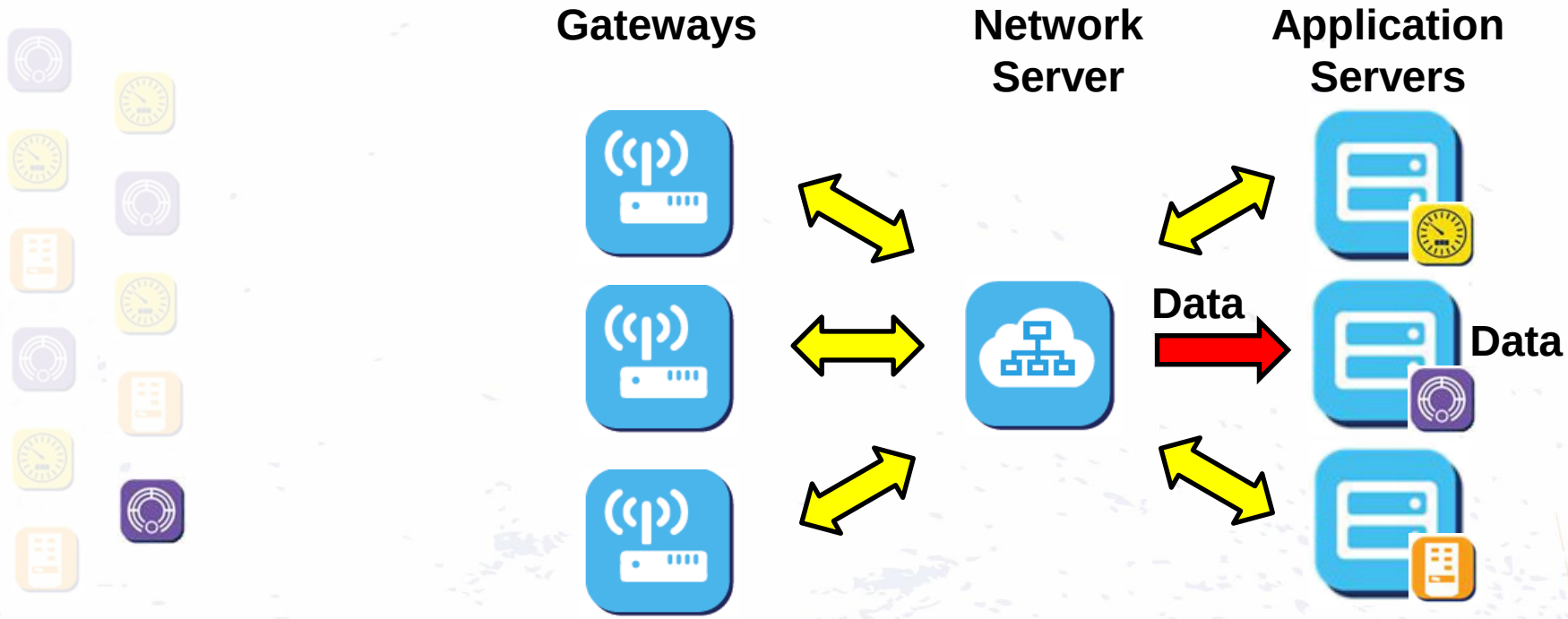


4. Passed through the Gateway…

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Application Server Data Message

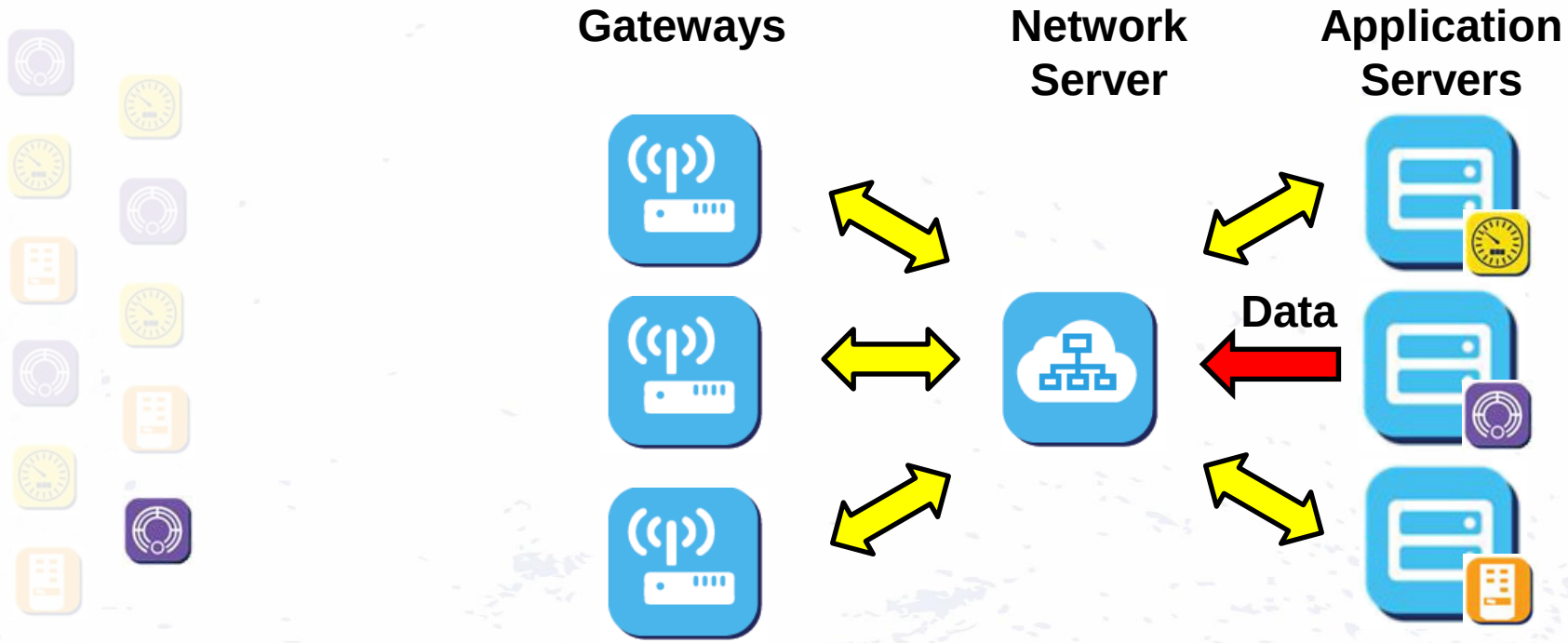


5. … and the Network Server sends to the Smoke Detector Application Server.

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Application Server Data Message

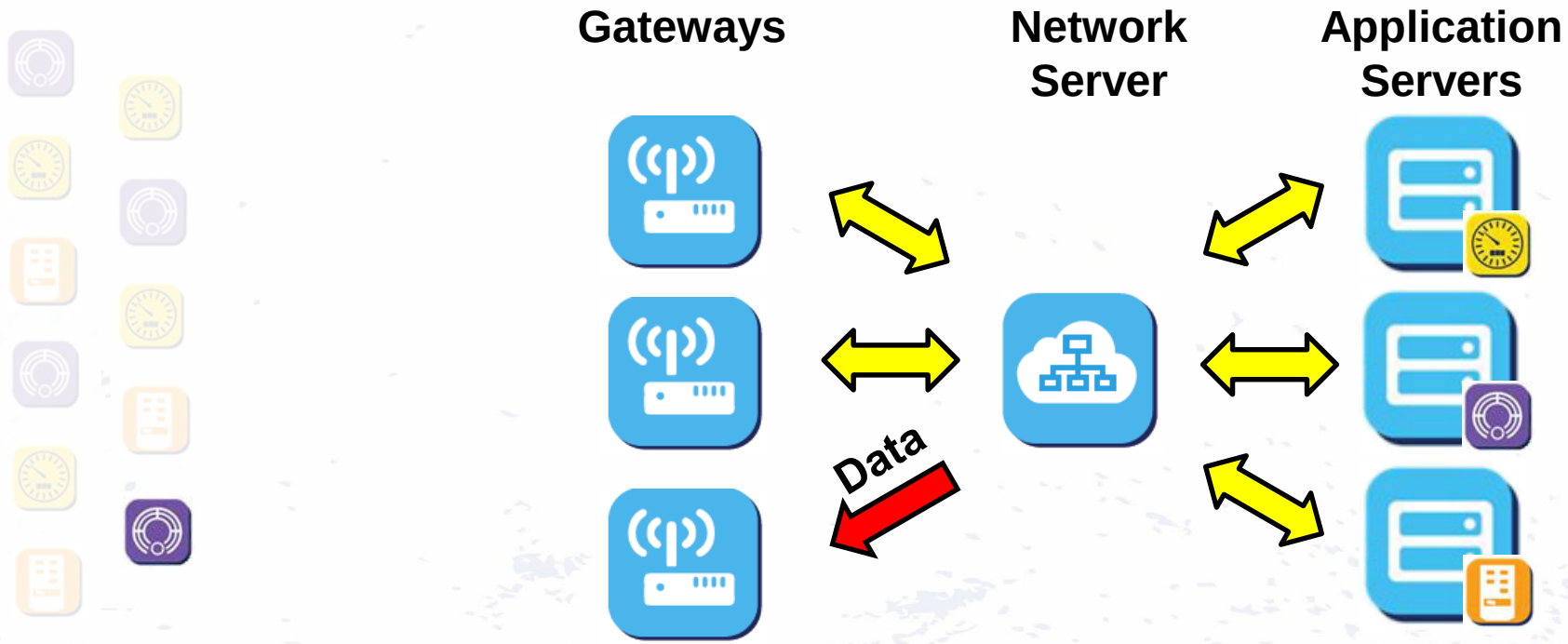


6. The Smoke Detector Application Server can now send the data message to the Smoke Detector.

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Application Server Data Message

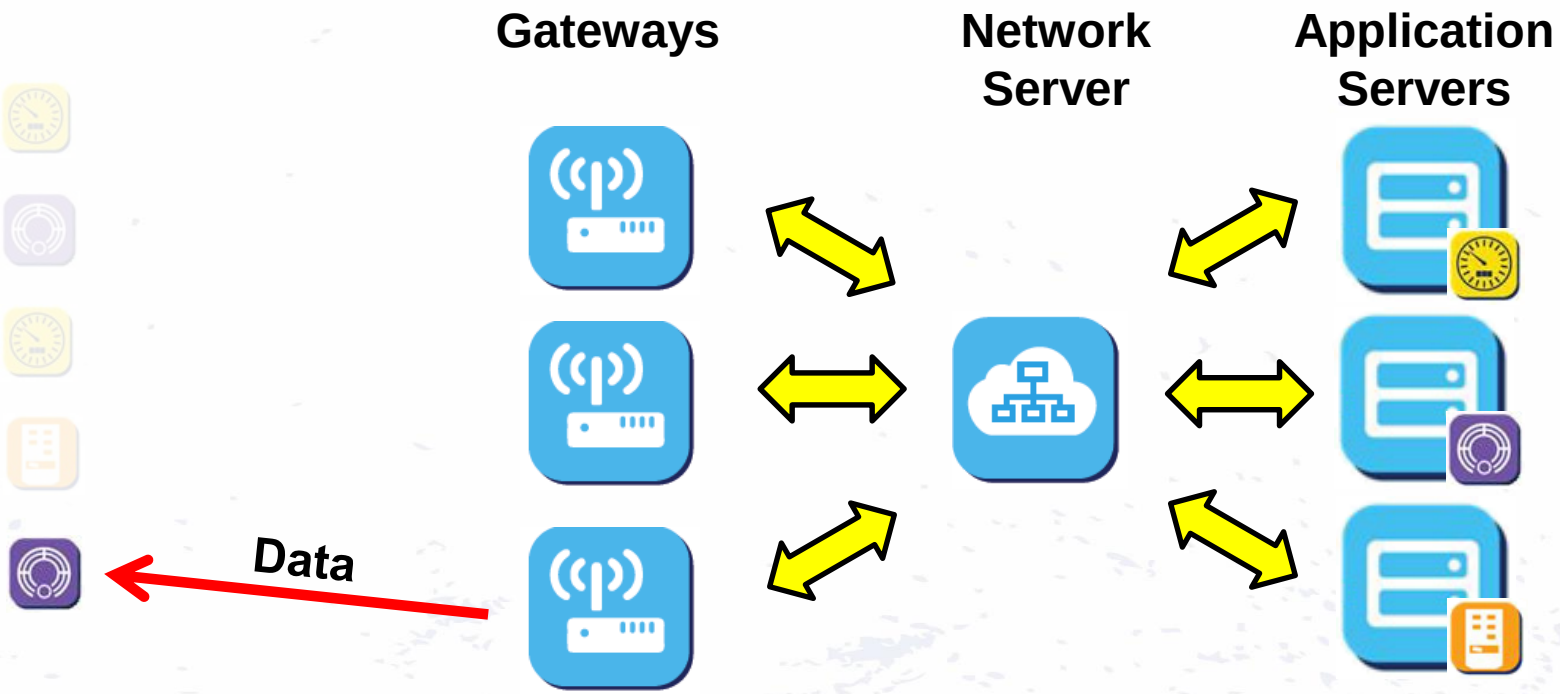


7. The Network Server sends the Data Message to the appropriate Gateway.

LoRaWAN™ Network Protocol

End-Device Data Communications (Class A)

Application Server Data Message



8. The Data Message is transmitted to the Smoke Detector during one of the two Receive Windows.

Sub-Agenda

- **LoRaWAN™ Network Protocol**
  - LoRa® Technology Modulation
  - How does LoRaWAN™ Technology Work?
  - End-Device Classes
  - End-Device Activation (Joining)
  - Security
  - End-Device Data Communication
  - Adaptive Data Rate (ADR)

LoRaWAN™ Network Protocol

Adaptive Data Rate (ADR)

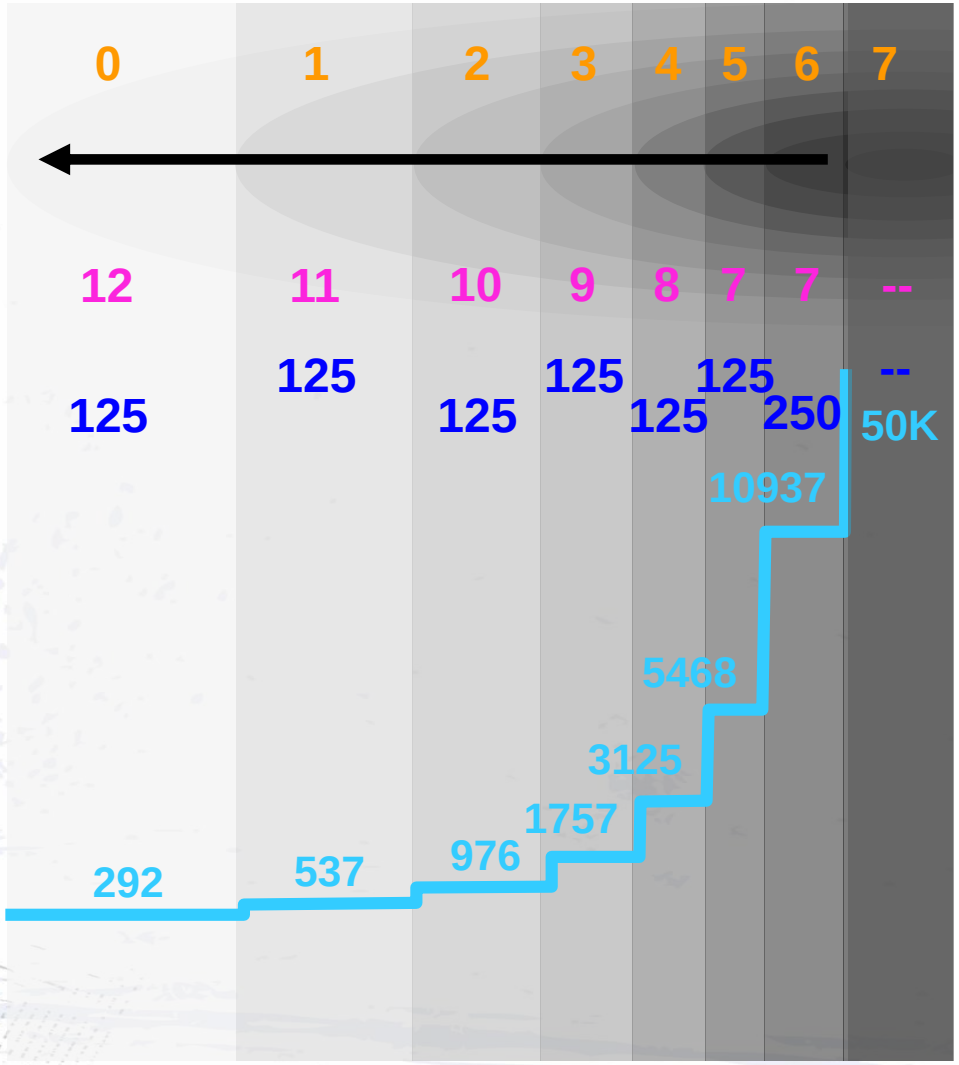
- **LoRaWAN can manage**
  - data rate and
  - RF power output

  for each end-device to
  - Optimize for fastest data rate,
  - Maximize battery life, and
  - Maximize network capacity

  based on range from gateway

LoRaWAN™ Network Protocol

Adaptive Data Rate (ADR)



Data Rate (DR)

Range

Spreading Factor (SF)

Bandwidth (BW) (kHz)

Bitrate (BR) (bps)

Note: European data rates shown

Summary

- **LoRaWAN™ Network Protocol**
  - LoRa® Technology Modulation
  - How does LoRaWAN™ Technology Work?
  - End-Device Classes
  - End-Device Activation (Joining)
  - Security
  - End-Device Data Communication
  - Adaptive Data Rate (ADR)

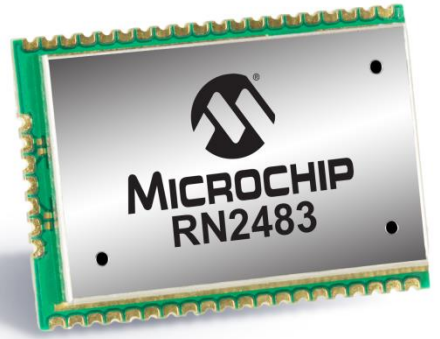
Agenda

- Internet of Things (IoT)
- LoRaWAN™ Network Protocol
- **LoRa® Technology Wireless Modules**
- Getting Started with RN2903 Module

Kahoot!

LoRa® Technology Wireless Modules

- **RN2483 LoRa® Technology Transceiver Module**
  - European (EU) 868/433 MHz
  - R&TTE Directive Assessed Radio Module
  - TX Power: up to +14 dBm
  - Power Consumption: 1.6 uA in Sleep



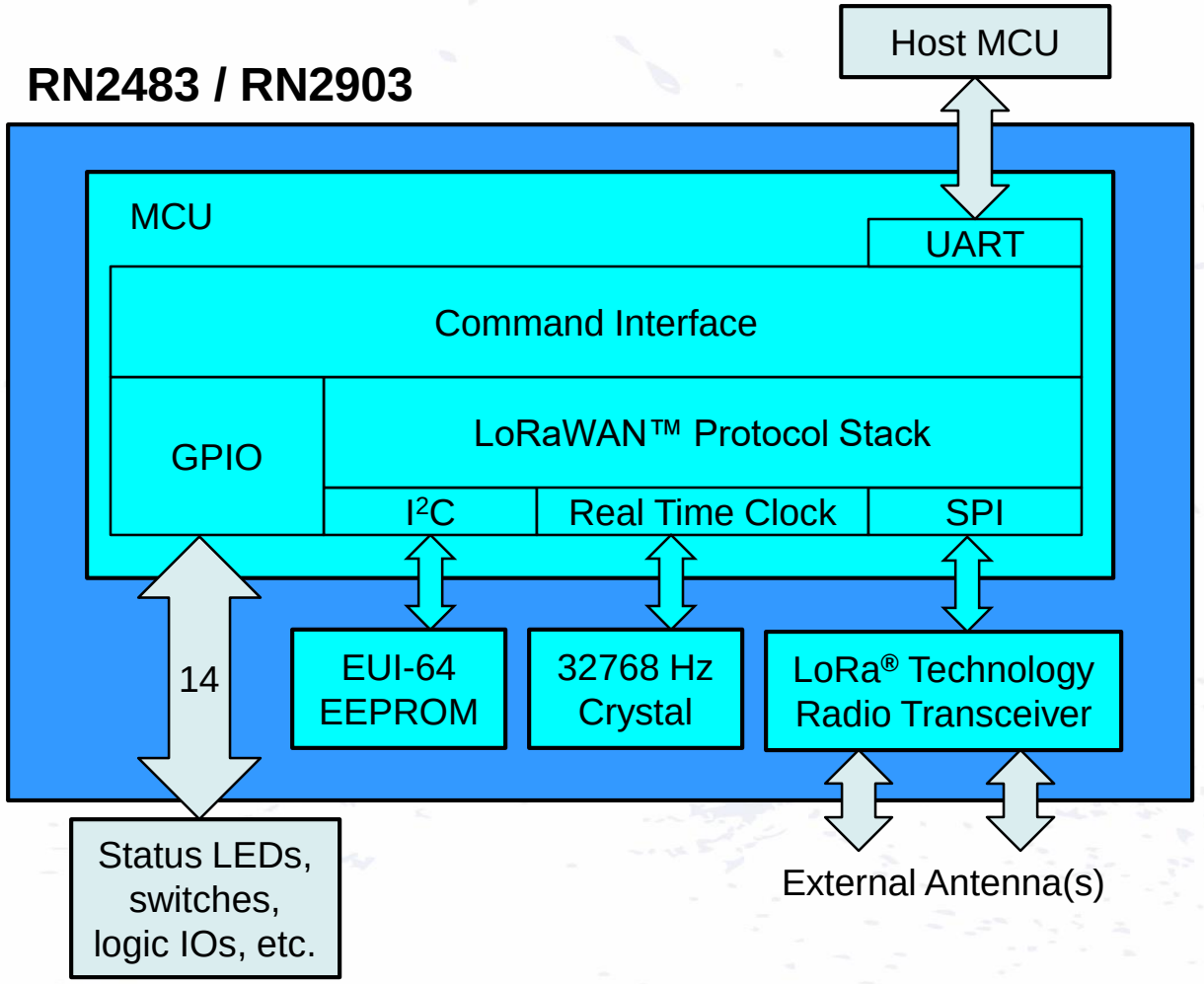
- **RN2903 LoRa® Technology Transceiver Module**
  - North American (NA) 915 MHz
  - FCC and IC modular certification
  - TX Power: up to +20 dBm
  - Power Consumption: 2.2 uA in Sleep

LoRa® Technology Wireless Modules

- **General Features**
  - Fully integrated module
  - On-board LoRaWAN™ Class A protocol stack
  - ASCII Command Interface over UART
  - UART Device Firmware Upgrade (DFU)
  - Integrated MCU and Crystal
  - EUI-64 Node Identity Serial EEPROM
  - 14 GPIOs
  - Compact form factor: 17.8 x 26.7 x 3 mm

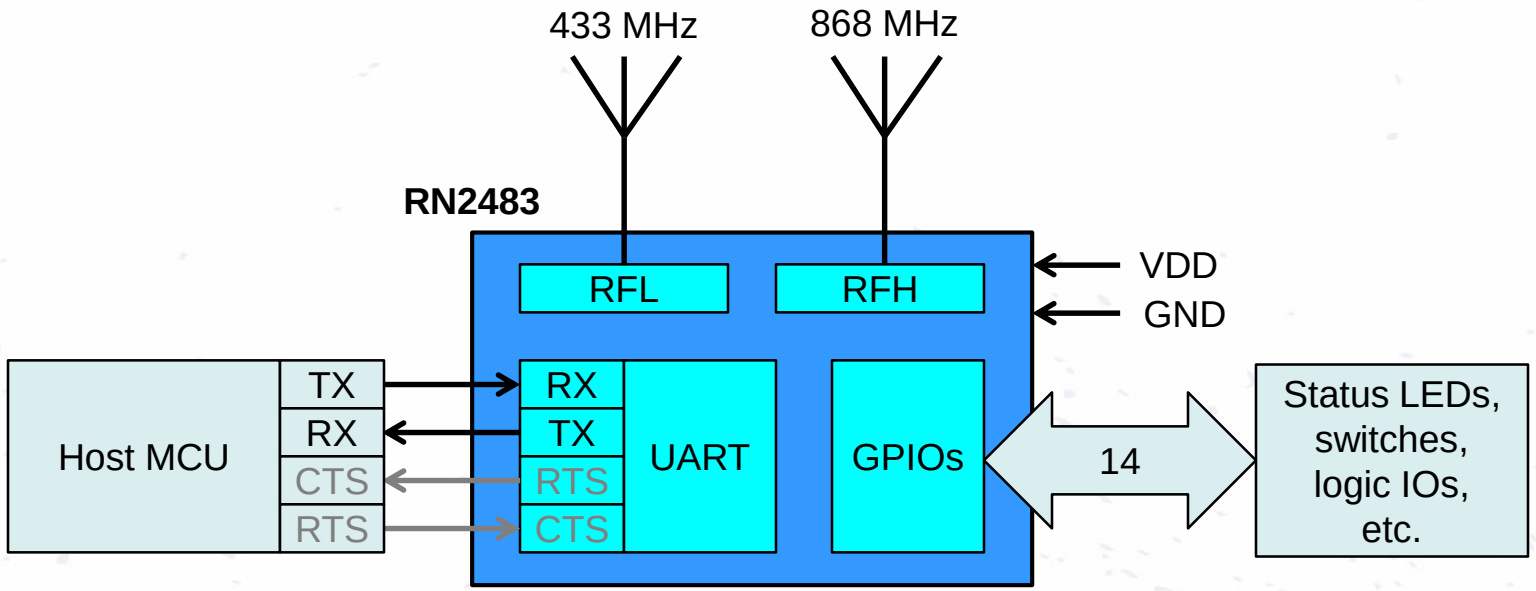
LoRa® Technology Wireless Modules

Block Diagram



LoRa® Technology Wireless Modules

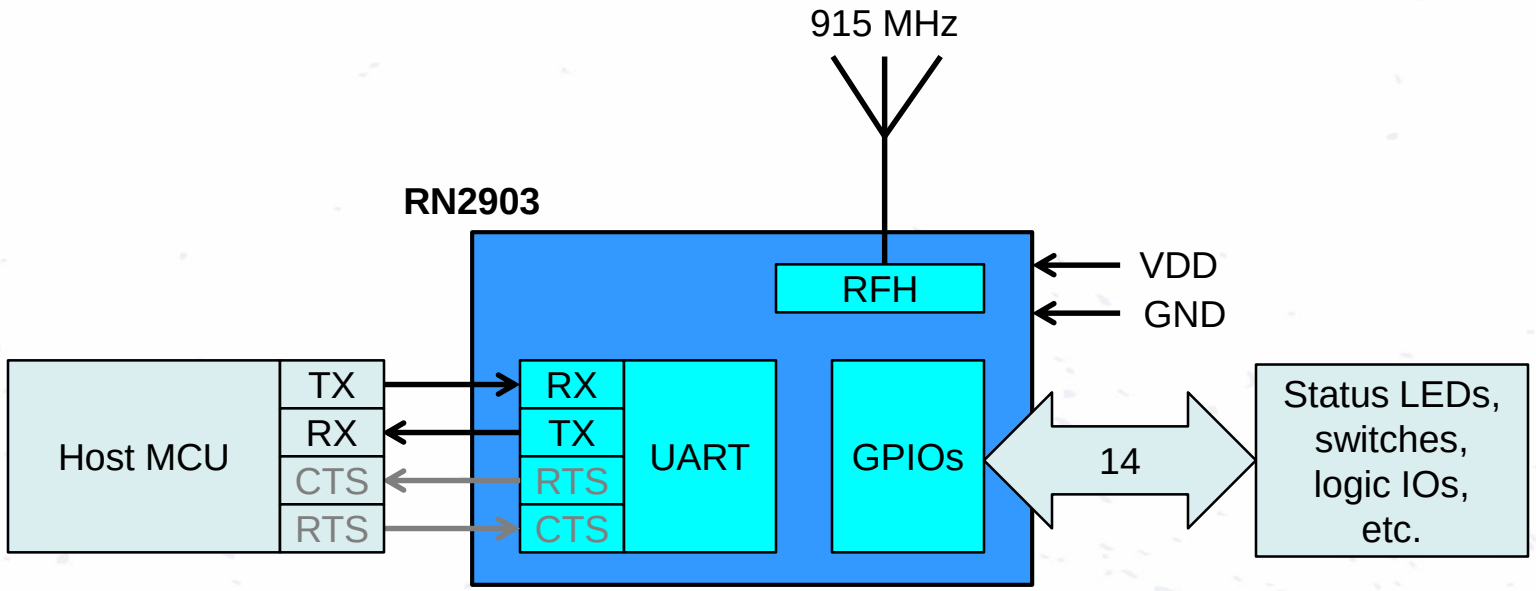
RN2483 LoRa® Technology Transceiver Module



Note: Optional RTS and CTS control lines will be supported in future firmware releases.

LoRa® Technology Wireless Modules

RN2903 LoRa® Technology Transceiver Module



Note: Optional RTS and CTS control lines will be supported in future firmware releases.

LoRa® Technology Wireless Modules

Development Tools

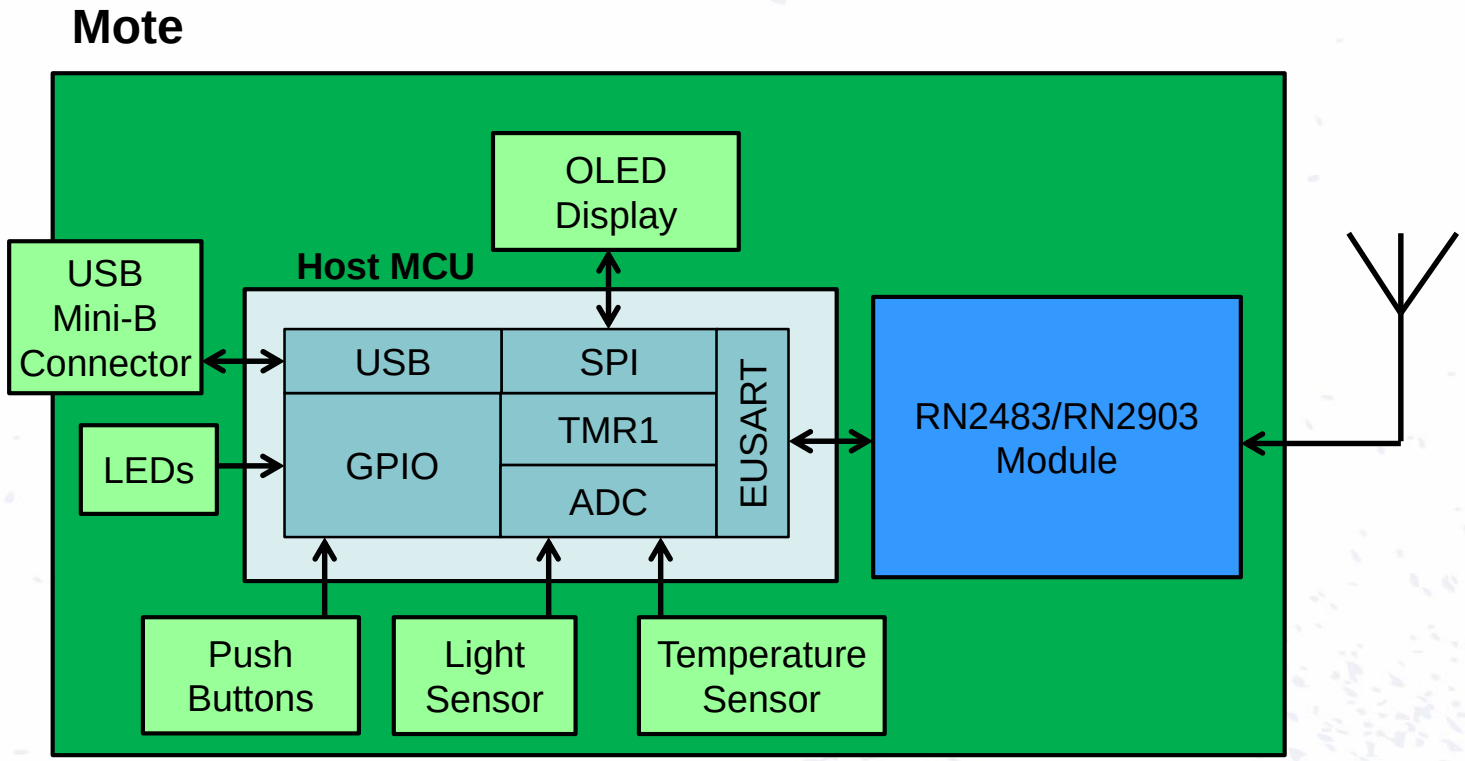
RN-2483-MOTE
RN-2903-MOTE

RN-2483-PICTAIL™
RN-2903-PICTAIL

EESA-IOT 5.0

LoRa® Technology Wireless Modules

LoRa® Technology Mote Block Diagram



LoRa® Technology Wireless Modules

- **Control Interface**
  - UART (TX/RX) communication
  - Default Baud Rate: 57600, 8N1, no flow control
  - Supports Auto Baud Detection

- **Command Interface**
  - Human Readable Text
  - Command Request => Command Reply / Replies
  - Command Request initiated by Host MCU
  - Command Reply initiated by the LoRa Technology Wireless Module

LoRa® Technology Wireless Modules

- **Command Syntax**
  - Key word(s) issued, followed by optional parameter(s)
  - Separated by **space** Character
  - Key Word(s) Case Sensitive
  - Parameter(s) Case Insensitive
  - **CR+LF** Command Delimiter

- **Command Request example:**

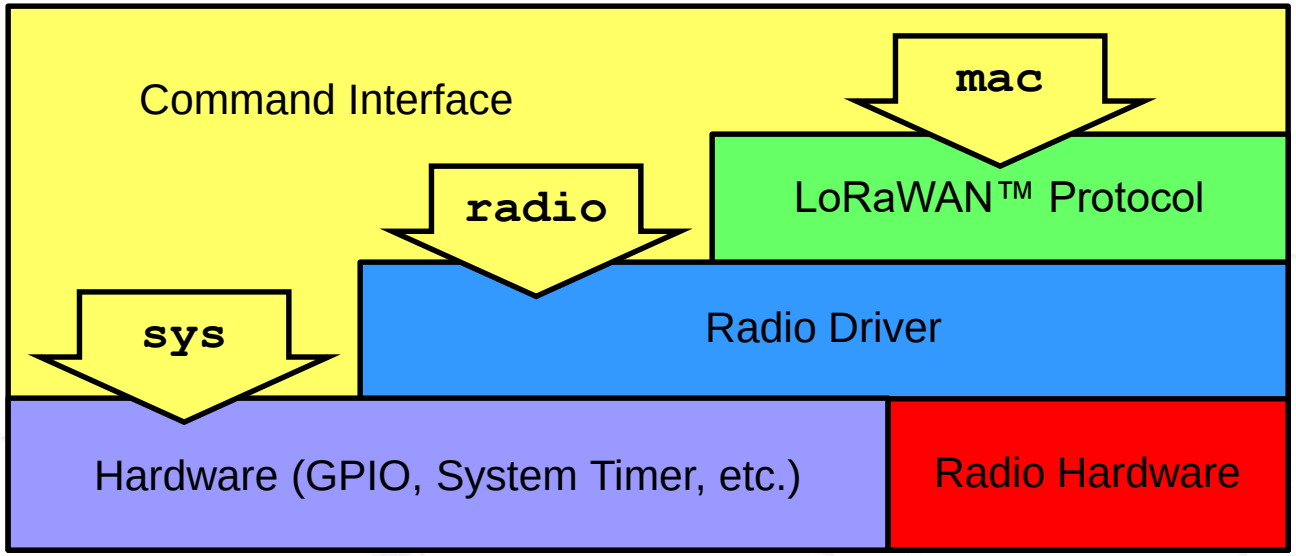
```
< mac set devaddr 048E436e\r\n
```

- **Command Reply example:**

```
> ok\r\n
```

LoRa® Technology Wireless Modules

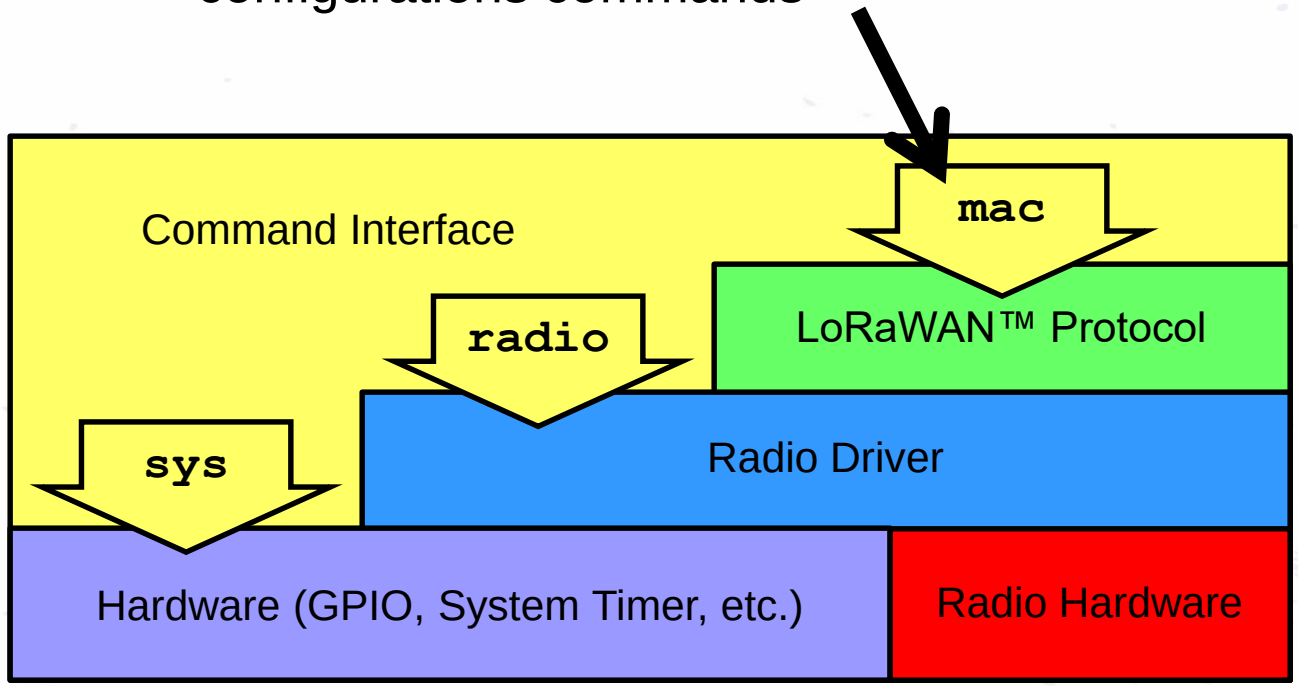
Command Interface



LoRa® Technology Wireless Modules

Command Interface

`mac` : Issues LoRaWAN™ Class A protocol network communication behaviors, actions and configurations commands



LoRa® Technology Wireless Modules

`mac` : Issues LoRaWAN™ Class A protocol network communication behaviors, actions and configurations commands

Parameter	Description
`reset`	Resets the RN2483 module to a specific frequency band.
`tx`	Sends the data string on a specified port number and sets default values for most of the LoRaWAN parameters.
`join`	Informs the RN2483 module to join the configured network.
`save`	Saves LoRaWAN Class A configuration parameters to the user EEPROM.
`forceENABLE`	Enables the RN2483 module after the LoRaWAN network server commanded the end device to become silent immediately.
`pause`	Pauses LoRaWAN stack functionality to allow transceiver (radio) configuration.
`resume`	Restores the LoRaWAN stack functionality.
`set`	Accesses and modifies specific MAC related parameters.
`get`	Reads back current MAC related parameters from the module.

LoRa® Technology Modules

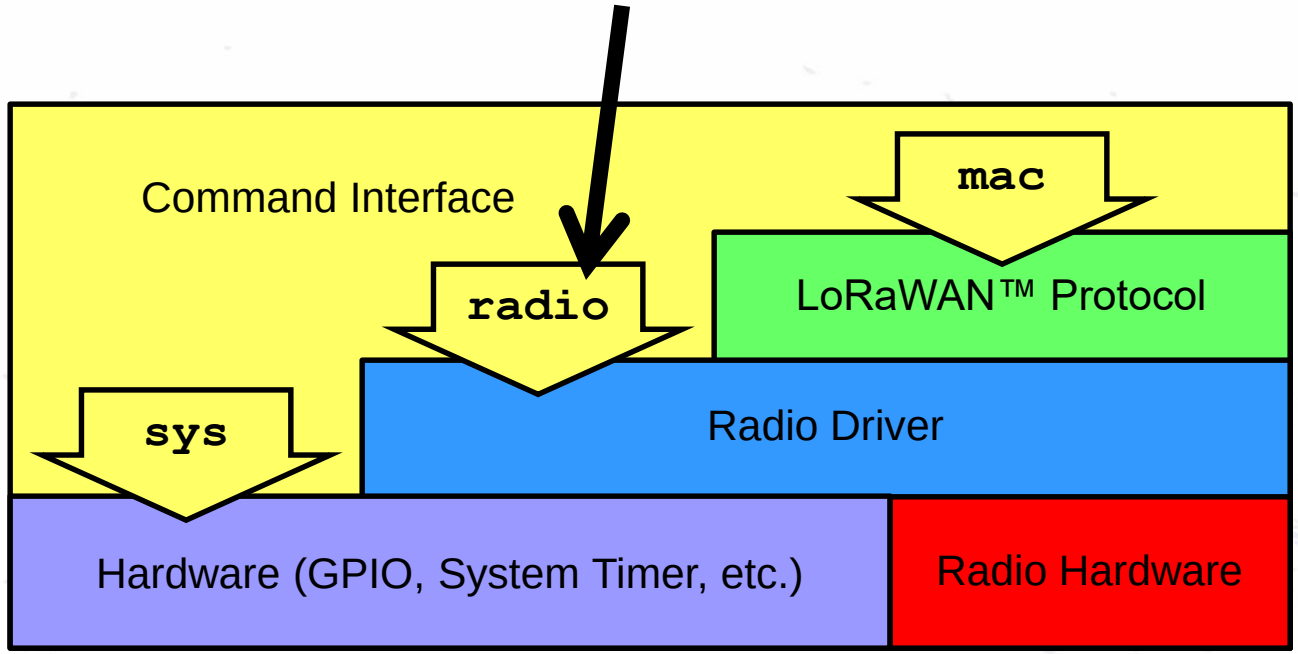
```
< mac set devaddr 048E436E
> ok


< mac join abp
> ok
> accepted
```

LoRa® Technology Wireless Modules

Command Interface

`radio` : Issues radio specific configurations, directly accessing and updating the transceiver setup



LoRa® Technology Wireless Modules

`radio`: Issues radio specific configurations, directly accessing and updating the transceiver setup

Parameter	Description
`rx`	This command configures the radio to receive simple radio packets according to prior configuration settings.
`tx`	This command configures a simple radio packet transmission according to prior configuration settings.
`cw`	This command will put the module into a Continuous Wave (cw) Transmission for system tuning or certification use.
`set`	This command allows modification to the radio setting directly. This command allows for the user to change the method of radio operation within module type band limits.
`get`	This command grants the ability to read out radio settings as they are currently configured.

Note 1: The `mac pause` command must be called before any radio transmission or reception, even if no MAC operations have been initiated before.

LoRa® Technology Wireless Modules

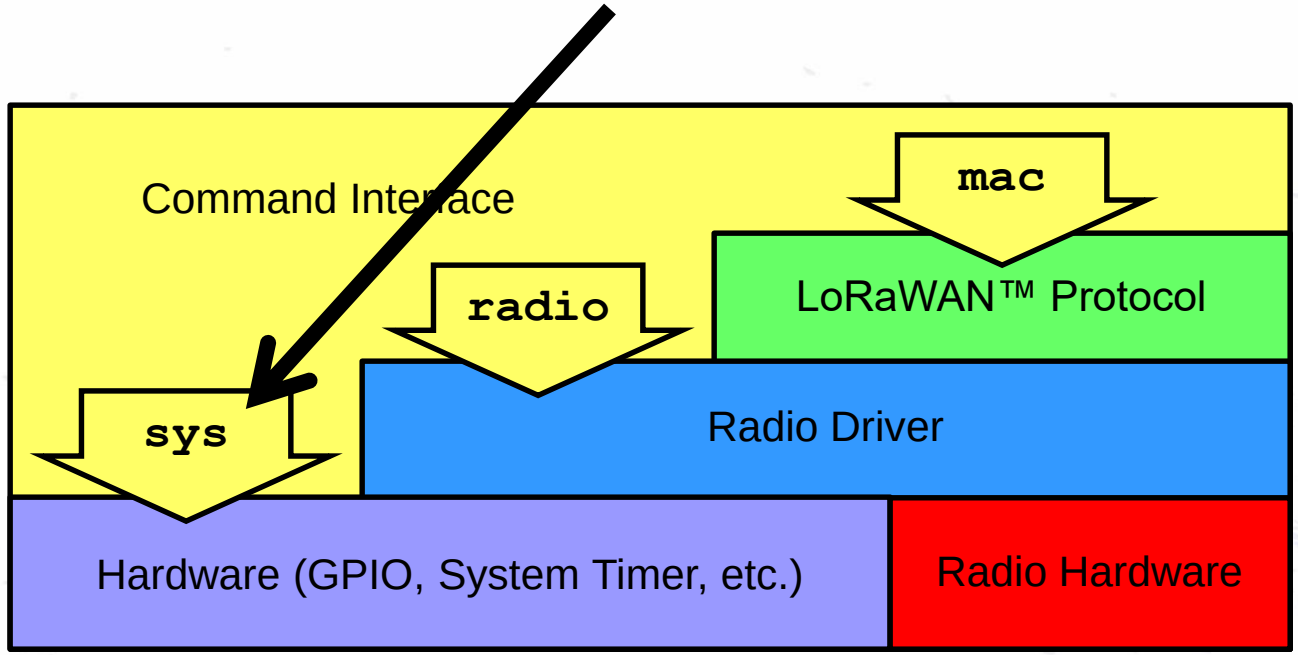
```
< radio cw on
> ok

< radio get mod
> lora
```

LoRa® Technology Wireless Modules

Command Interface

`sys` : Issues system level behavior actions, gathers status information on the firmware and hardware version, or accesses the module user EEPROM memory



LoRa® Technology Wireless Modules

sys : Issues system level behavior actions, gathers status information on the firmware and hardware version, or accesses the module user EEPROM memory

Parameter	Description
sleep	Puts the system in Sleep for a finite number of milliseconds.
reset	Resets and restarts the RN2483 module.
eraseFW	Deletes the current RN2483 module application firmware and prepares it for firmware upgrade. The RN2483 module bootloader is ready to receive new firmware.
factoryRESET	Resets the RN2483 module's configuration data and user EEPPROM to factory default values and restarts the RN2483 module.
set(1)	Sets specified system parameter values.
get(1)	Gets specified system parameter values.

LoRa® Technology Wireless Modules

```
< sys sleep 5000
> ok

< sys reset
> RN2483 0.9.5 Mar 24 2015 14:17:03
```

Agenda

- Internet of Things (IoT)
- LoRaWAN™ Network Protocol
- LoRa® Technology Wireless Modules
- **Getting Started with RN2903 Module**

Getting Started with RN2903 Module

End-Device

Gateway

Network Server

Application Server

Getting Started with RN2903 Module

End-Device with Over-the-Air Activation (OTAA) and uplink data transmission

- **Configuration**
  - `mac set deveui C3D1000030000001`
  - `mac set appeui DEDEAAAA00000030`
  - `mac set appkey ABAAAA9AAAAA7B69545555555555496`
- **Activation**
  - `mac join otaa`
- **Communication**
  - `mac tx uncnf 16 48454c4c4f`

Getting Started with RN2903 Module

End-Device with Activation By Personalization (ABP) and uplink data transmission

- **Configuration**
  - `mac set devaddr 0482FF05`
  - `mac set nwkskey D95AC917E01FF24B69F4D9F9A0C4EC8D`
  - `mac set appskey 70169735FDC5CD64F3C3ECE938DFCFE2`
- **Activation**
  - `mac join abp`
- **Communication**
  - `mac tx uncnf 16 48454c4c4f`

Q & A

Gracias!!

LEGAL NOTICE

SOFTWARE:

You may use Microchip software exclusively with Microchip products. Further, use of Microchip software is subject to the copyright notices, disclaimers, and any license terms accompanying such software, whether set forth at the install of each program or posted in a header or text file.

Notwithstanding the above, certain components of software offered by Microchip and 3rd parties may be covered by "open source" software licenses – which include licenses that require that the distributor make the software available in source code format. To the extent required by such open source software licenses, the terms of such license will govern.

NOTICE & DISCLAIMER:

These materials and accompanying information (including, for example, any software, and references to 3rd party companies and 3rd party websites) are for informational purposes only and provided "AS IS." Microchip assumes no responsibility for statements made by 3rd party companies, or materials or information that such 3rd parties may provide.

MICROCHIP DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING ANY IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY DIRECT OR INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND RELATED TO THESE MATERIALS OR ACCOMPANYING INFORMATION PROVIDED TO YOU BY MICROCHIP OR OTHER THIRD PARTIES, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBLITY OF SUCH DAMAGES OR THE DAMAGES ARE FORESEEABLE. PLEASE BE AWARE THAT IMPLEMENTATION OF INTELLECTUAL PROPERTY PRESENTED HERE MAY REQUIRE A LICENSE FROM THIRD PARTIES.

TRADEMARKS:

The Microchip name and logo, the Microchip logo, AnyRate, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, LINK MD, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, ETHERSYNCH, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and QUIET-WIRE are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, RightTouch logo, REAL ICE, Ripple Blocker, Serial Quad I/O, SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

Desarrollando aplicaciones con LoRaWAN

- Del laboratorio al campo
- Preguntas y respuestas
- Problemas y soluciones
- TIPs
- Demos
- Y mucho mas…

Primero lo primero

- Como se que todos quieren llevarse “algo”, vamos a hacer unos juegos con una app que se llama Kahoot
- Por favor, descarguen Kahoot en sus teléfonos o accedan desde un navegador (www.kahoot.it)
- El que mas rápido responde (y bien) suma mas puntos
- Vamos a dar premios a los 3 mejores puntajes

¿Cómo funciona Kahoot!?

Kahoot!

Ejemplo sin puntaje

Objetivo

- **Al final de esta clase deberían poder:**
  - Empezar un proyecto LoRaWAN con confianza
  - Armar una estrategia de funcionamiento de sus equipos, teniendo en cuenta las particularidades de LoRaWAN
  - Poder resolver los problemas mas comunes
  - Usar la placa EESA-IOT para conectarse a una Red LoRaWAN
  - Saber a quién llamar cuando todo lo de arriba falle ☺

Suposiciones…

- **Se suponen las siguientes cosas:**
  - Conocen lo básico de LoRaWAN
  - Saben programar en C y/o Arduino
  - Vamos a tener muy pocos “slides”
  - Necesitamos que la clase sea interactiva, con lo cual espero muchas preguntas

Agenda

- Preguntas y respuestas
- Problemas mas comunes (y sus soluciones)
- TIPs para usar el RN2903A y LoRaWAN en general
- Demostraciones prácticas de uso
- Demostración de la placa EESA-IOT 5.0

Pregunta #1

¿Qué plan de frecuencias me conviene usar?

¿Qué plan de frecuencias me conviene usar?

- **En Argentina y en Latinoamérica en general usamos AU915-928**
- **La banda 902-915 está reservada para banda 8 de celulares**
- **Microchip va a comenzar a producir un módulo que va a tener esta frecuencia de fábrica**
  - RN2903A-I/RMSA103

¿Qué plan de frecuencias me conviene usar?

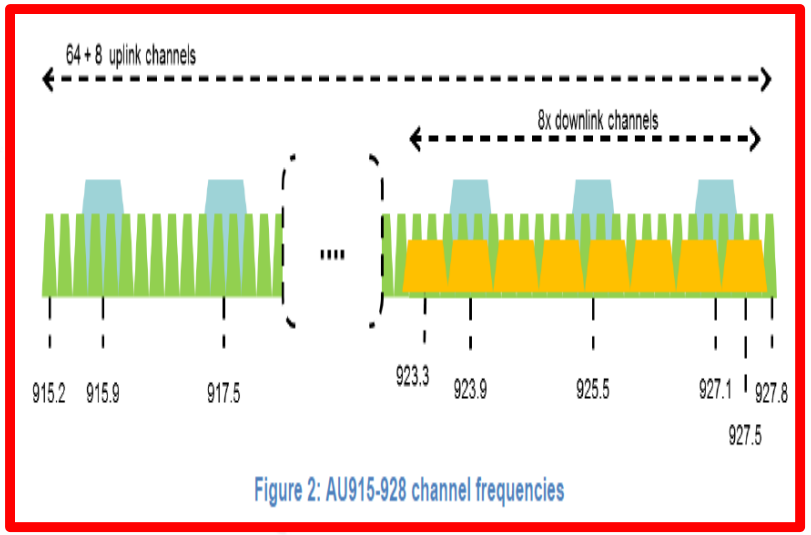


Figure 2: AU915-928 channel frequencies

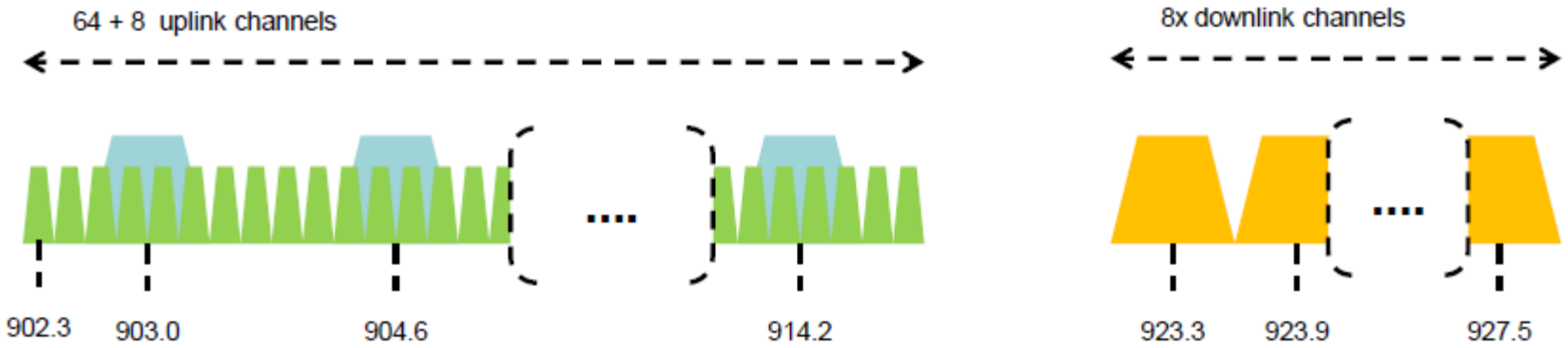


Figure 1: US902-928 channel frequencies

¿Qué plan de frecuencias me conviene usar?

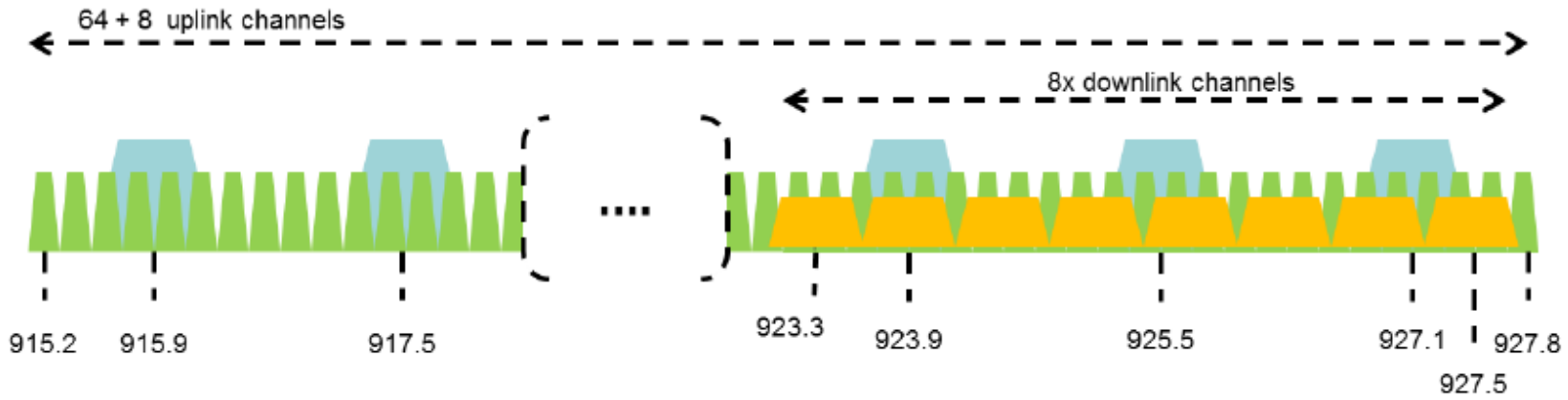


Figure 2: AU915-928 channel frequencies

Kahoot!

Pregunta #2

¿Puedo actualizar el firmware?

FOTA, FUOTA, OTA…

¿Puedo actualizar el firmware?

- **Si!**
- **El ancho de banda es pequeño, pero es totalmente viable**
- **Hay que usar esquemas de actualización de firmware modificado, es decir, que solo se actualiza el firmware que se modificó**
- **Hay que trabajar a bajo nivel con el bootloader y entender como funciona**
- **Usar multicast preferentemente**

¿Puedo actualizar el firmware?

- **Existe en github una demo para actualización de firmware usando un procesador ARM (como el ATSAMD20 o ATSAML21)**
  - Incluyen los algoritmos de compresión para enviar el firmware
  - Ver en www.github.com/TrackNet

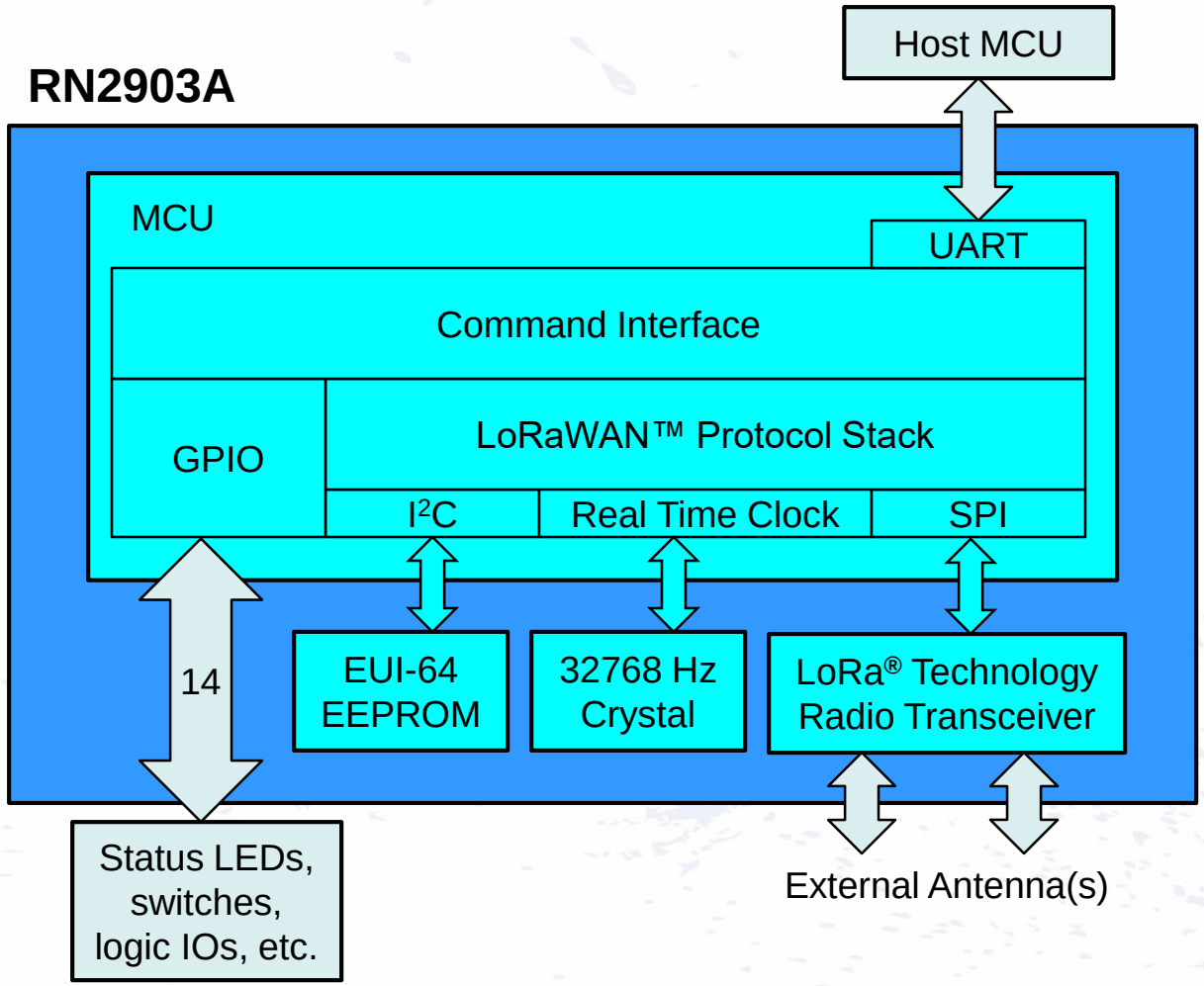
Pregunta #3

¿Puedo poner mi aplicación en el módulo?

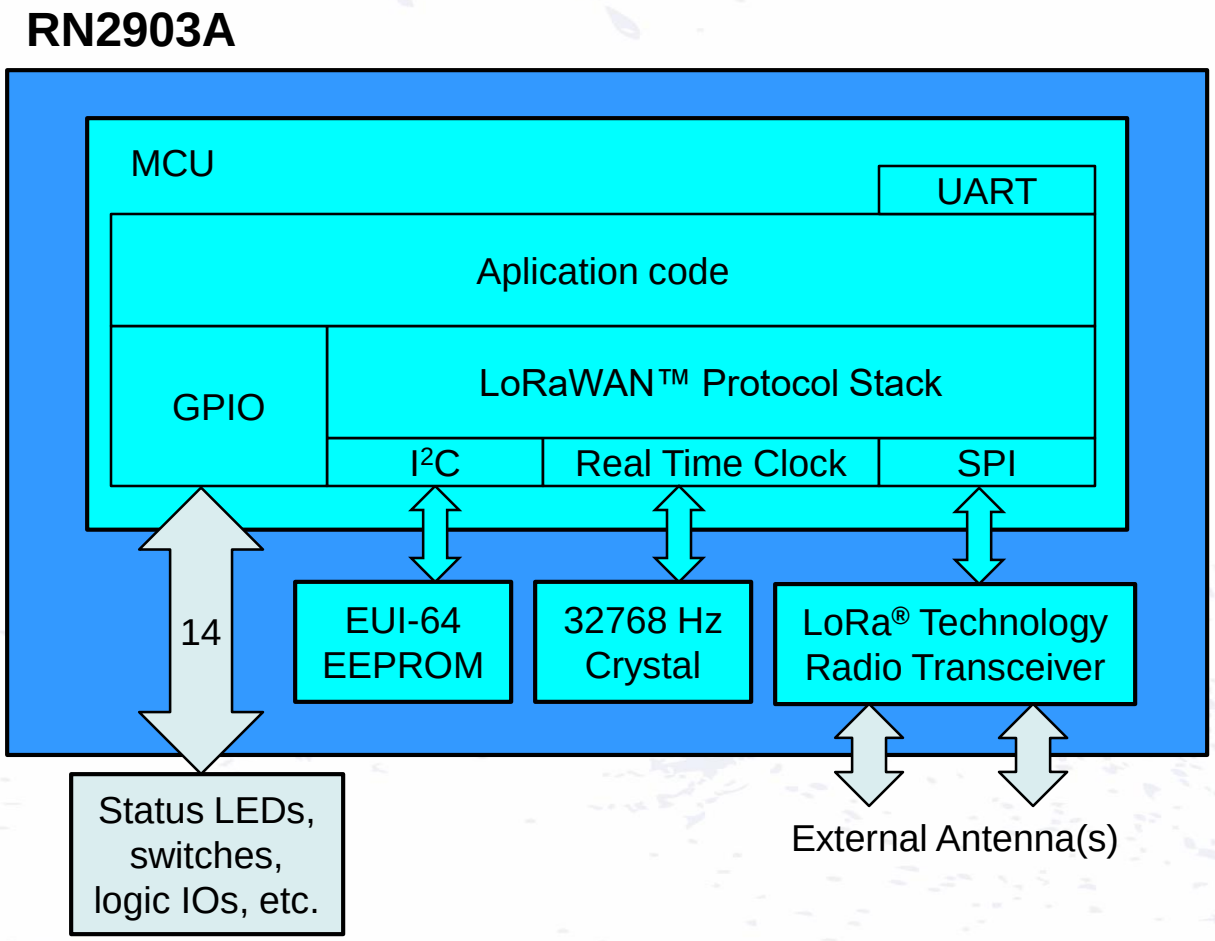
¿Puedo poner mi aplicación en el módulo?

- **Si!**
- **Se usa la librería LoRaWAN que es parte del MCC (MPLAB Code Configurator)**
- **El tiempo de desarrollo es un poco mayor, pero totalmente viable**
- **Al usar esta forma, el módulo ya no está mas certificado por la Alianza LoRa**

¿Puedo poner mi aplicación en el módulo?



¿Puedo poner mi aplicación en el módulo?



Pregunta #4

¿Qué versión de LoRaWAN uso?

¿Qué versión de LoRaWAN uso?

- **La versión que actualmente recomendamos es la 1.0.2 Rev. A**
- **Es importante que ser Rev. A**
  - La Rev. B tiene un dweling time mayor a 400ms, que no está aprobado con los entes reguladores (en Argentina y Brasil)
- **Pueden bajar la especificacion de www.lora-alliance.org o pedírmela a mi (mauricioj@artimar.com.br)**

Pregunta #5

¿Cuánt[illegible]es puedo transmitir?

Revisá
Tu
Fantástico
Manual

¿Cuántos bytes puedo transmitir?

- **Ver en la spec:**

2.5.6 AU915-928 Maximum payload size

The maximum **MACPayload** size length (*M*) is given by the following table. It is derived from the maximum allowed transmission time at the PHY layer taking into account a possible repeater encapsulation. The maximum application payload length in the absence of the optional **FOpt** MAC control field (*N*) is also given for information only. The value of *N* might be smaller if the **FOpt** field is not empty:

DataRate	*M*	*N*
0	19	11
1	61	53
2	134	126
3	250	242
4	250	242
5:7	Not defined	
8	41	33
9	117	109
10	230	222
11	230	222
12	230	222
13	230	222
14:15	Not defined	

Table 31: AU915-928 maximum payload size

¿Cuántos bytes puedo transmitir?

- **Ver en la spec:**

4.3.1.6 Frame options (FOptsLen in FCtrl, FOpts)

The frame-options length field (**FOptsLen**) in **FCtrl** byte denotes the actual length of the frame options field (**FOpts**) included in the frame.

FOpts transport MAC commands of a maximum length of 15 octets that are piggybacked onto data frames; see Chapter 5 for a list of valid MAC commands.

If **FOptsLen** is 0, the **FOpts** field is absent. If **FOptsLen** is different from 0, i.e. if MAC commands are present in the **FOpts** field, the port 0 cannot be used (**FPort** must be either not present or different from 0).

MAC commands cannot be simultaneously present in the payload field and the frame options field. Should this occur, the device shall ignore the frame.

Pregunta #6

¿Cuántos nodos se pueden conectar a un gateway?

¿Es realmente escalable la red?

Pero mirá que necesito mandar datos cada 10 segundos

Y son como 1000 sensores!!

¿Cuántos nodos se pueden conectar?

- **Esta pregunta normalmente está acompañada de otras dudas como**
  - Capacidad de mensajes de la red
  - Latencia
  - Cantidad de bytes (que ya vimos)

- **Veamos como responder todo…**

¿Cuántos nodos se pueden conectar?

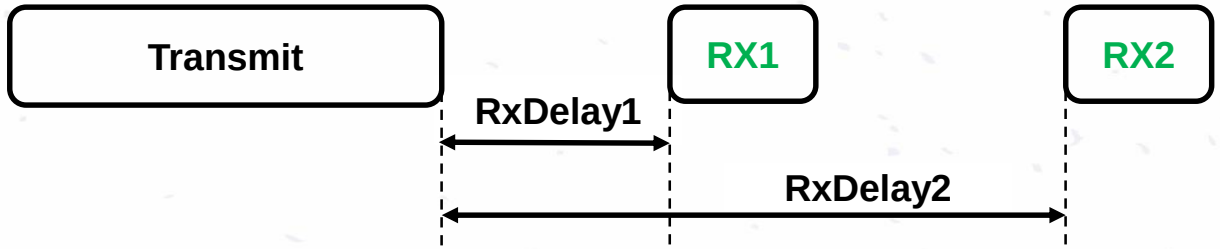
- **Asumiendo**
  - Gateway de 8 canales
  - Trama de 10 bytes
  - Data Rates diferentes y balanceados

  - Un GW tiene una capacidad de 700K paquetes por día

  - La red YEAP! tiene instalados GWs de 16 canales, con lo cual soportan 1.4 Mi de paquetes, por cada GW.

¿Cuántos nodos se pueden conectar?

- **Respecto de la latencia**
  - Depende de la aplicación, la configuración de la red y del backhaul usado en los gateways



  - Típicamente es de menos de 1 segundo

Pregunta #7

¿Qué canales uso?

¿Qué canales uso?

If using the over-the-air activation procedure, the end-device should broadcast the JoinReq message alternatively on a random 125 kHz channel amongst the 64 channels defined using **DR0** and a random 500 kHz channel amongst the 4 channels defined using **DR4**. The end-device should change channel for every transmission.

Personalized devices shall have all 72 channels enabled following a reset.

- **Está (obviamente) en la especificación de LoRaWAN**
- **Hay que usar los 64 canales**
- **La “red” luego define qué canales se van a usar (qué grupo de canales específicos)**

Pregunta #8

¿Existe un firmware clase C para el RN2903A?

¿Existe el firmware Clase C?

SI!

- **Es la versión 1.0.5**
  - Estuvimos entregando un reléase candidate, pero ya tenemos la versión final
  - Hay nuevos comandos, como por ejemplo:
    - `mac set class`
    - `mac set mcastdevaddr`
    - `mac set mcastnwkskey`
    - `mac set mcastappskey`

Problemas mas habituales y sus soluciones

Estos son los problemas que van a encontrar:

- **El módulo no se conecta a la red**
  - Dice ‘join denied’ constantemente
- **Recibo el mensaje ‘invalid_data_len’**
- **No puedo despertar el modulo del modo bajo consumo**
- **El módulo responde ‘busy’ cuando intento transmitir**

Join denied

- **Los motivos mas comunes de este error son:**
- **Que el módulo no está dado de alta en la red**
  - Darle de alta mediante el DASS
- **Que no hay cobertura o una red LoRaWAN cercana**
  - Ver pregunta específica
- **No se seleccionó la version correcta de LoRaWAN Regional Spec al darlo de alta**
- **Hay que mandar el “mac join accept” varias veces… ¿porqué?**

¿Dónde hay cobertura?

Hoy existe cobertura de YEAP!, una red publica, en las principales ciudades de Argentina y otros países.

Los invitamos a contactarnos para conversar sobre su proyecto y ver como podemos atender sus necesidades de red.

Invalid_data_len

- **Significa que los bytes que quiero transmitir, no caben en el buffer de transmisión**
- **Según lo que hablamos, ¿Porqué podría pasar esto?**
  - Fops, MAC Commands

Modo bajo consumo

- **Actualmente el modulo se despierta de SLEEP enviando un ‘break’ por la UART.**
  - Si este ‘break’ no se envía correctamente la UART calcula mal el baudrate y se pierde comunicación con el modulo
- **Puedo ofrecerles una versión ‘custom’ que se despierta de SLEEP y re-configura la UART al valor de fábrica (57600 8N1)**

TIP #1

- **Usar el commando link check**

```
mac set linkchk 60
```

- **Pide al network server el “nivel de señal recibido” y la cantidad de GWs que “vieron” el mensaje**
- **La respuesta se puede obtener con los comandos:**

```
mac get gwnb
mac get mrgn
```

TIP #2

- Ante un error invalid_data_len la solucion sugerida para resolver este problema es transmitir un mensaje de 1 byte “dummy” que servirá solo para “limpiar” el buffer de cualquier commando MAC pendiente

TIP #3

- **Usar Automatic Reply**

```
mac set ar on
```

- **Este commando responde automaticamente cuando el network server pide algo, por ejemplo, un ACK**
- **Vamos a tener menos (o ninguna) incidencia de invalid_data_len**

TIP #4

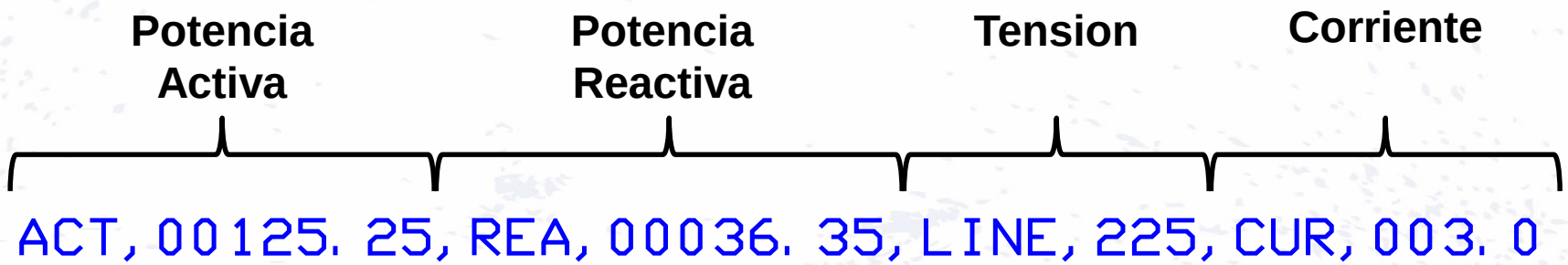
- Activar todos los canales
- NO usen “solo los 8 primeros canals” o esquemas similares
- Si la red cambia de canals su equipo queda imposibilitado de conectarse nuevamente

TIP #5

- **Usar diferentes appkeys en cada equipo**
- **Esto es crítico para la seguridad**
  - Los DEVEUI son públicos
  - Los APPKEY también, pero tienen que ser únicos para evitar que sea posible cloran los devices en caso de que se divulgue un APPKEY

TIP #6

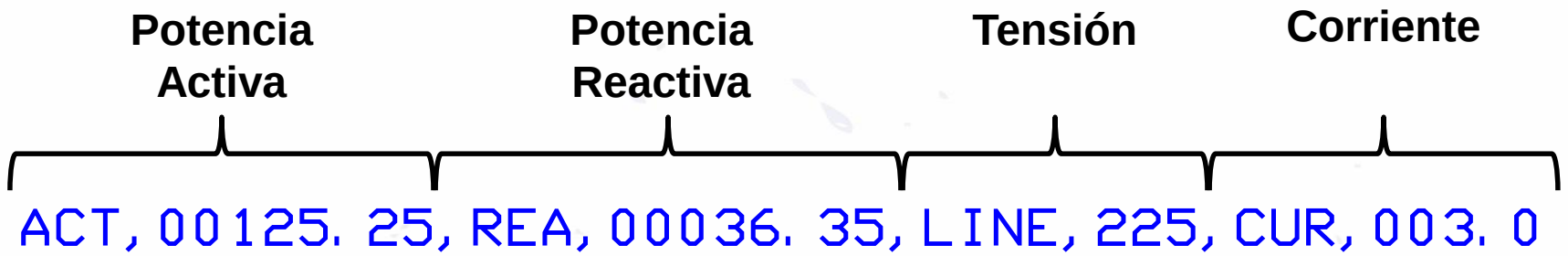
- Optimizar cantidad de bytes transmitidos
- Por ejemplo, quiero transmitir la potencia que me entrega un medidor de energia.
- El dato que viene del medidor tiene este formato (así lo lee mi UART):



Ejemplo

- **Antes de meternos de lleno en la placa, usando Arduino, hagamos algo mas “real” usando los comandos del modulo RN2903A**

TIP #6



- **Una buena forma de resolver esto es usando los PUERTOS de LoRaWAN**
- **Cada dato los transmito únicamente por un puerto específico**

Q & A

Gracias!!

LEGAL NOTICE

SOFTWARE:

You may use Microchip software exclusively with Microchip products. Further, use of Microchip software is subject to the copyright notices, disclaimers, and any license terms accompanying such software, whether set forth at the install of each program or posted in a header or text file.

Notwithstanding the above, certain components of software offered by Microchip and 3rd parties may be covered by "open source" software licenses – which include licenses that require that the distributor make the software available in source code format. To the extent required by such open source software licenses, the terms of such license will govern.

NOTICE & DISCLAIMER:

These materials and accompanying information (including, for example, any software, and references to 3rd party companies and 3rd party websites) are for informational purposes only and provided "AS IS." Microchip assumes no responsibility for statements made by 3rd party companies, or materials or information that such 3rd parties may provide.

MICROCHIP DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING ANY IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY DIRECT OR INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND RELATED TO THESE MATERIALS OR ACCOMPANYING INFORMATION PROVIDED TO YOU BY MICROCHIP OR OTHER THIRD PARTIES, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBLITY OF SUCH DAMAGES OR THE DAMAGES ARE FORESEEABLE. PLEASE BE AWARE THAT IMPLEMENTATION OF INTELLECTUAL PROPERTY PRESENTED HERE MAY REQUIRE A LICENSE FROM THIRD PARTIES.

TRADEMARKS:

The Microchip name and logo, the Microchip logo, AnyRate, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, LINK MD, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, ETHERSYNCH, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and QUIET-WIRE are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, RightTouch logo, REAL ICE, Ripple Blocker, Serial Quad I/O, SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.