IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

GEMELOS DIGITALES PARA LOS SISTEMAS CIBER-FÍSICOS

INTRODUCCION A CONTEXTO IOT Y PRESENTACION DEL LAB

Renzo E. NAVAS

Montevideo, 7-11 Noviembre 2022
Instituto de Computación - Facultad de Ingeniería

https://www.fing.edu.uy/inco/grupos/mina/wscf2022/

SUMMARY

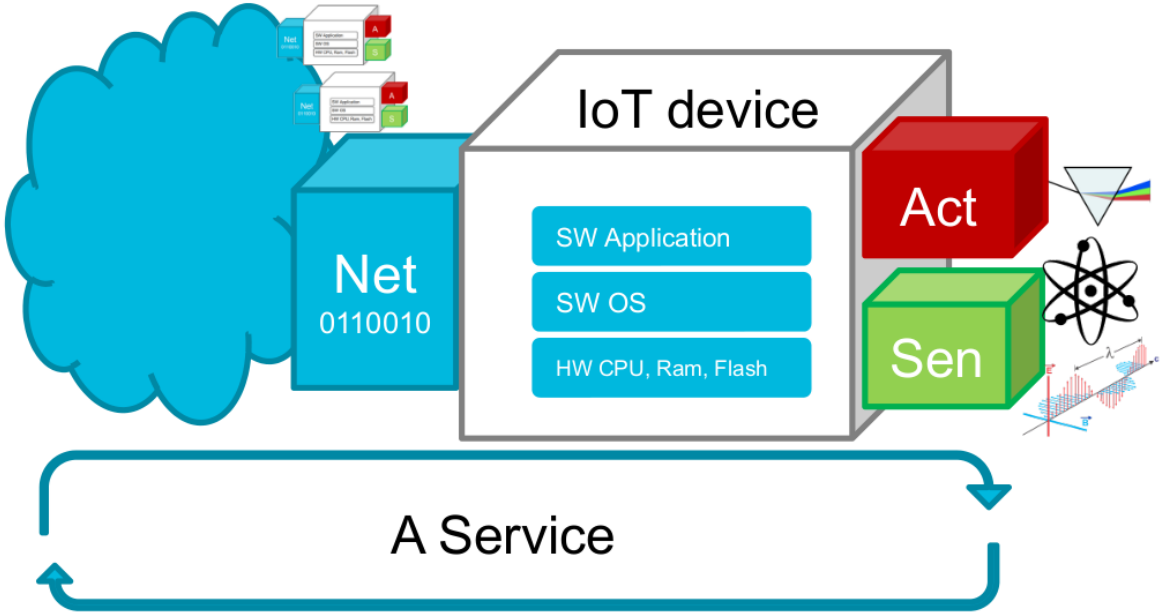
1. **THE CONSTRAINED IOT**
2. **LPWANS (LORA)**
3. **COMPACT DATA REPRESENTATION**
4. **LAB: USE CASE, HW/SW AND ARCHITECTURE**

THE CONSTRAINED IOT

Internet of Things
General Introduction

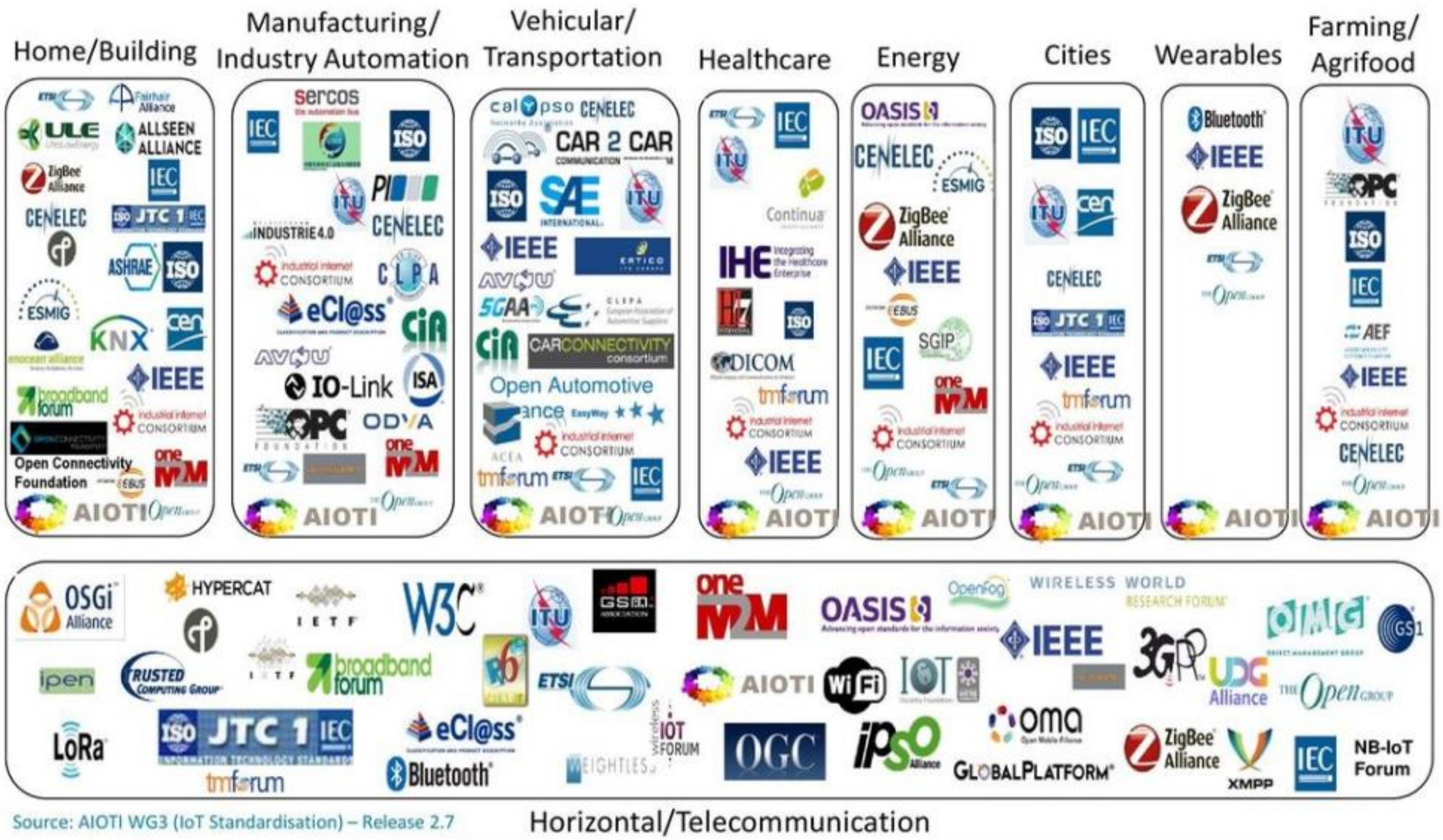
"Internet of Things" (IoT) Paradigm ~ 1999

- Adding **Network and Sensing** capabilities to unconnected things
- IoT devices are interfaces between **physical and digital** realms
- IoT system: many devices providing a "service"
  - App: Smart City, Agriculture, Industry 4.0, Transportation ...



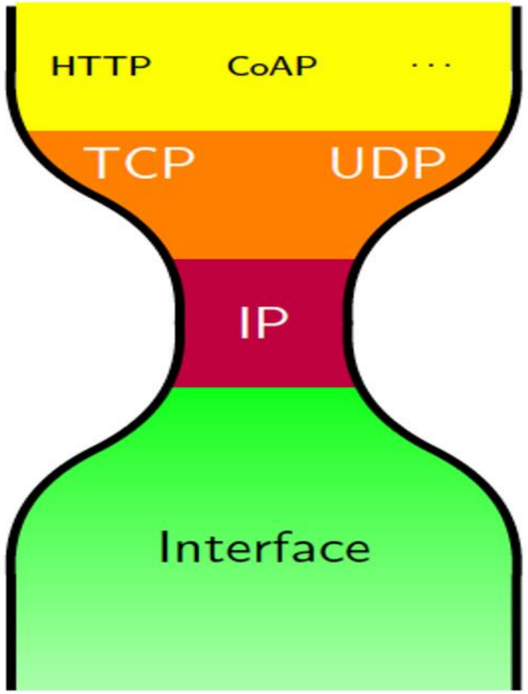
Internet of Things

Heterogeneous



- Internet Architecture
  - Goal: transport data of an application anywhere in the world.
  - OSI: 7 Layers, stacked on top; each layer relies on the services provided by the lower ones.

- Convergence towards a reduced set of protocols and standardization of data representation



Constrained IoT
Definitions and context

- Among the heterogeneous IoT, we focus on **constrained** IoT
  - *C. Devices*: limited energy, processing power, or memory
  - C. *Networks*: bandwidth, reliability, or topology stability

- Challenges: standard protocols need to be adapted

- The **IETF** defines in RFC7228 * a terminology for constrained-node networks

Device Class	RAM (KiB)	Flash (KiB)
Class 0	$<< 10$	$<< 100$
Class 1	≈ 10	≈ 100
Class 2	100	250

* Being updated https://datatracker.ietf.org/doc/html/draft-ietf-lwig-7228bis-00#section-3

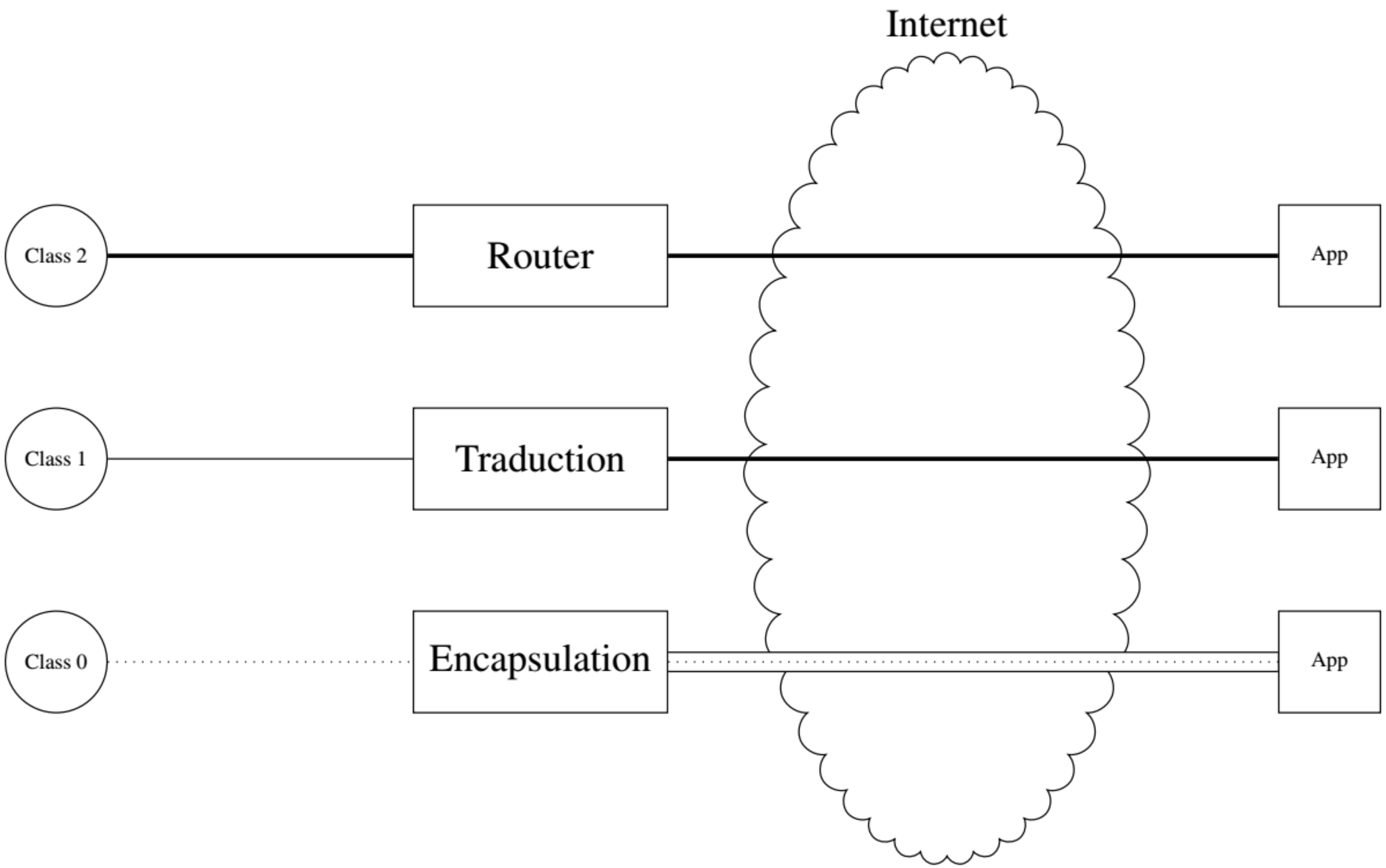


Fig. Interconnection possibilities, depending on device Class (Fig. From [LTN])

Constrained IoT

Protocols stack

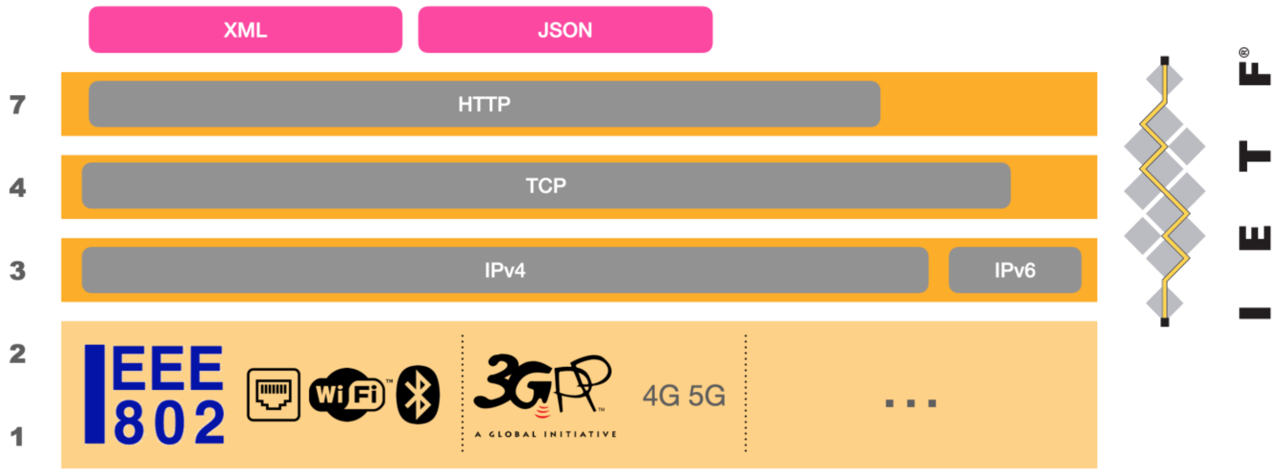


Fig. Main Internet protocols stack

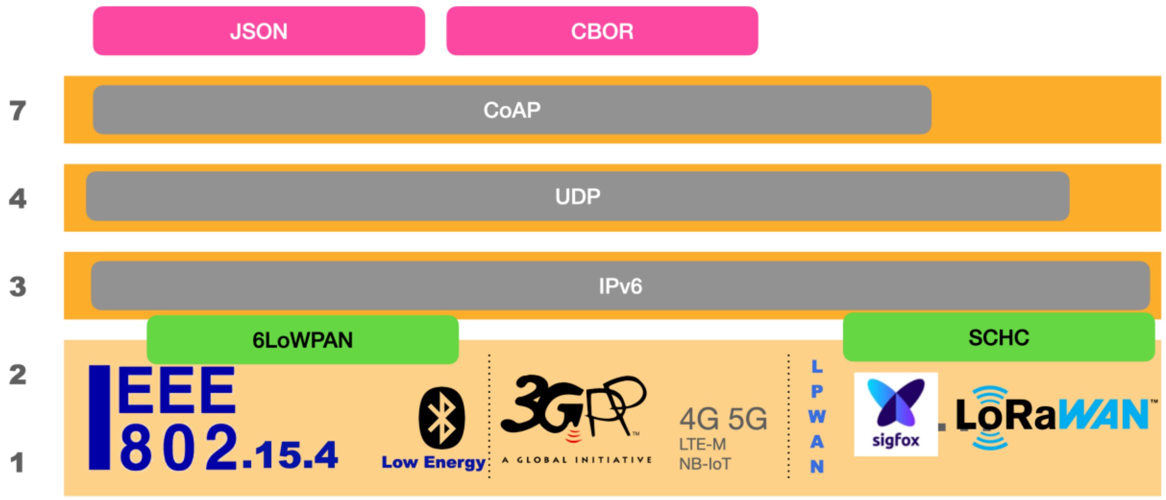


Fig. IoT protocols stack

- Different levels of interoperability (higher the layer, the higher level)

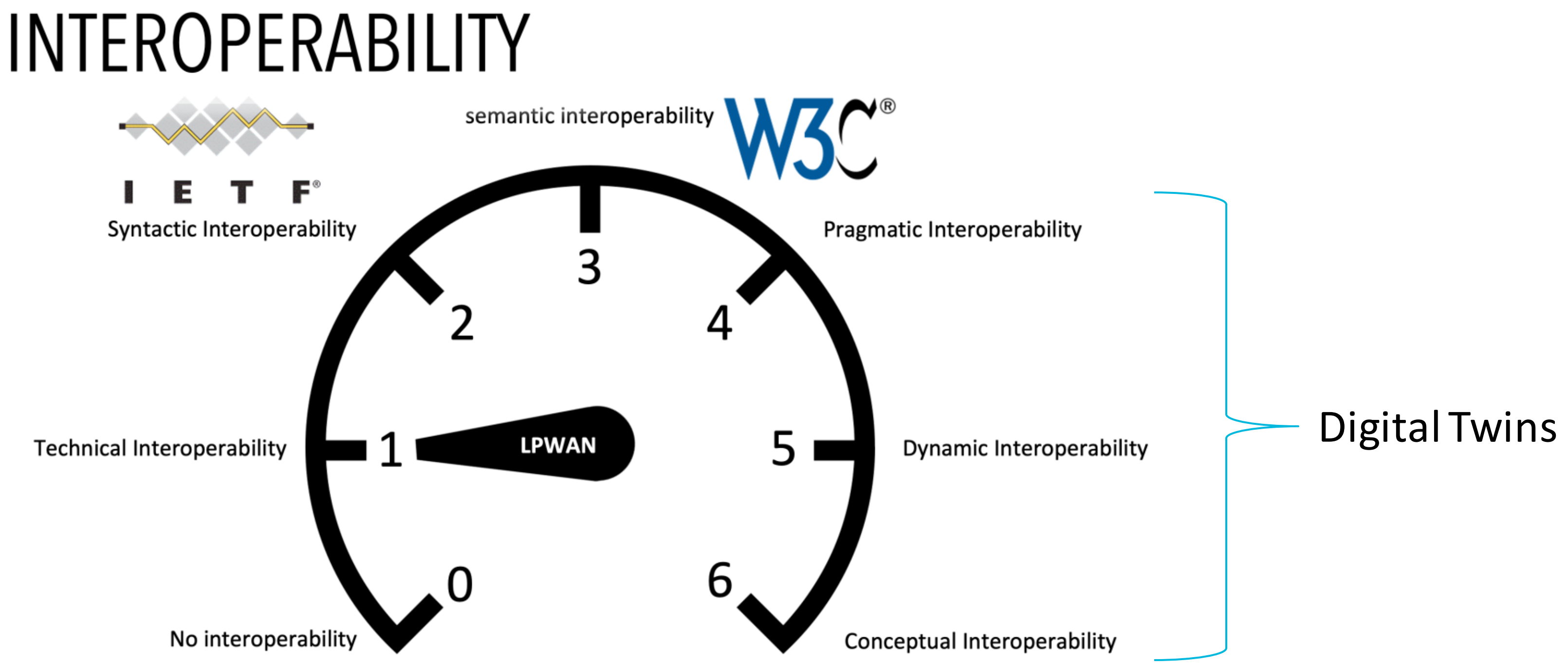


Fig. Levels of Conceptual Interoperability Model (LCIM)* (Fig. From [LTN])

* Andreas TOLK et James AMUGUIRA . "The levels of conceptual interoperability model". In : Proceedings of the 2003 fall simulation interoperability workshop. Tome 7. Citeseer. 2003, pages 1-11 (cf. page 22).

LPWANS:
FOCUS ON LORA AND LORAWAN

Introduction
Wireless Communication Technologies

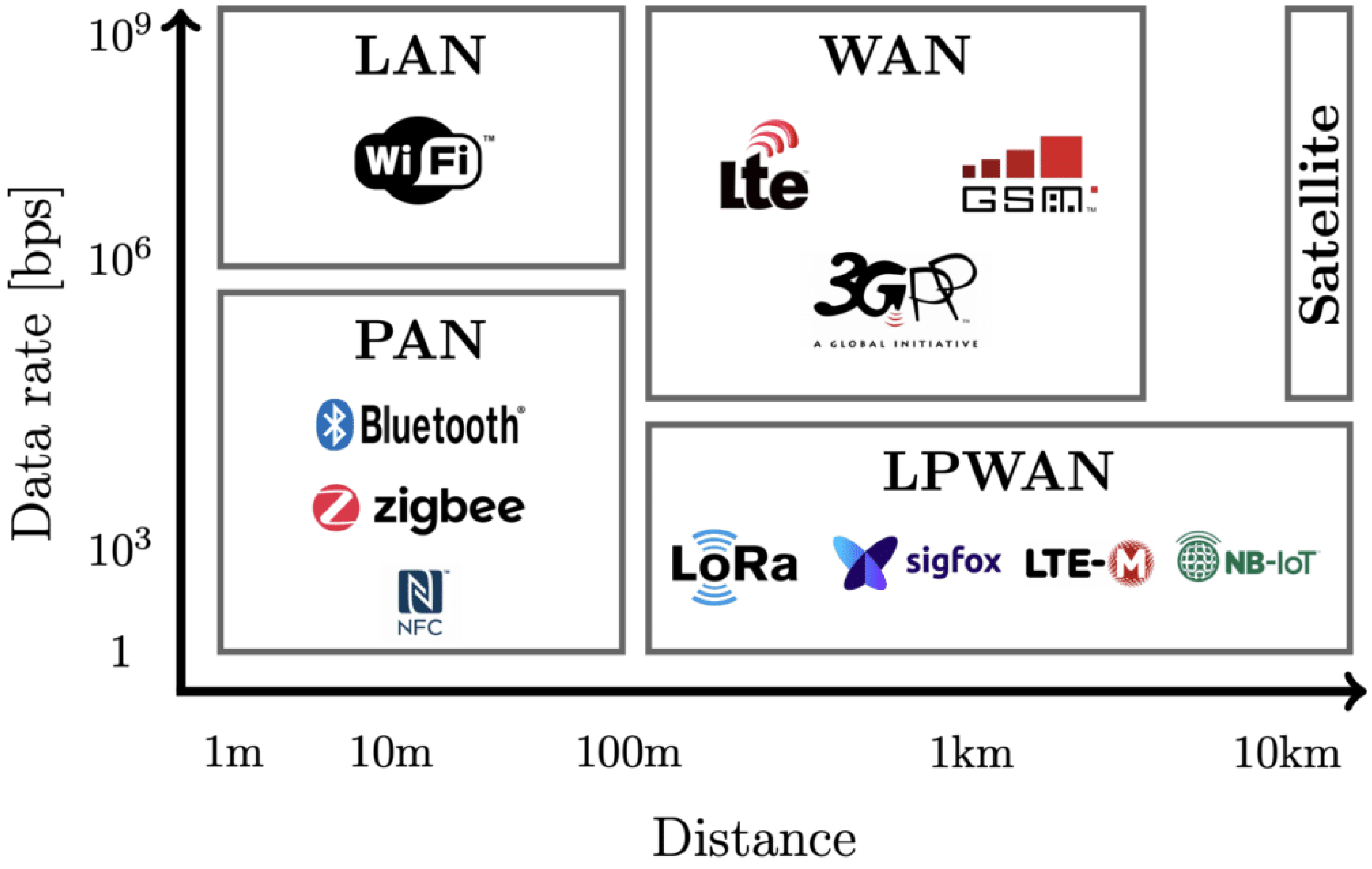


Fig. Main Wireless Communication Technologies qualitative Distance vs Data Rate categorization

Source: https://emanuelepagliari.it/2020/10/13/internet-of-things-wireless-communication-protocols/

LPWANs

Low-Power Wide Area Networks (LPWANs)

- ## Characteristics
  - large coverage areas, low bandwidth,
  - very small packet and application-layer data sizes,
  - long battery life operation

- ## Star Topologies
  - No need/possibility for routing (Mesh)

- ## LPWAN Overview RFC8376
  - https://www.rfc-editor.org/rfc/rfc8376

- ## Ex: LoRaWAN, NB-IoT, Sigfox, Wi-SUN

```
                                                    +------+
 ()    ()   ()         |                            |LPWAN-|
   ()  () () ()       / \           +---------+     | AAA  |
() () () () () ()    /   \========|    /\    |====|Server|  +-----------+
 ()  ()    ()       |             | <--|--> |    +------+  |APPLICATION|
()  ()  ()  ()     / \============|    v    |==============|   (App)   |
  ()  ()  ()      /   \           +---------+              +-----------+
 DEV           Radio Gateways         NGW
```

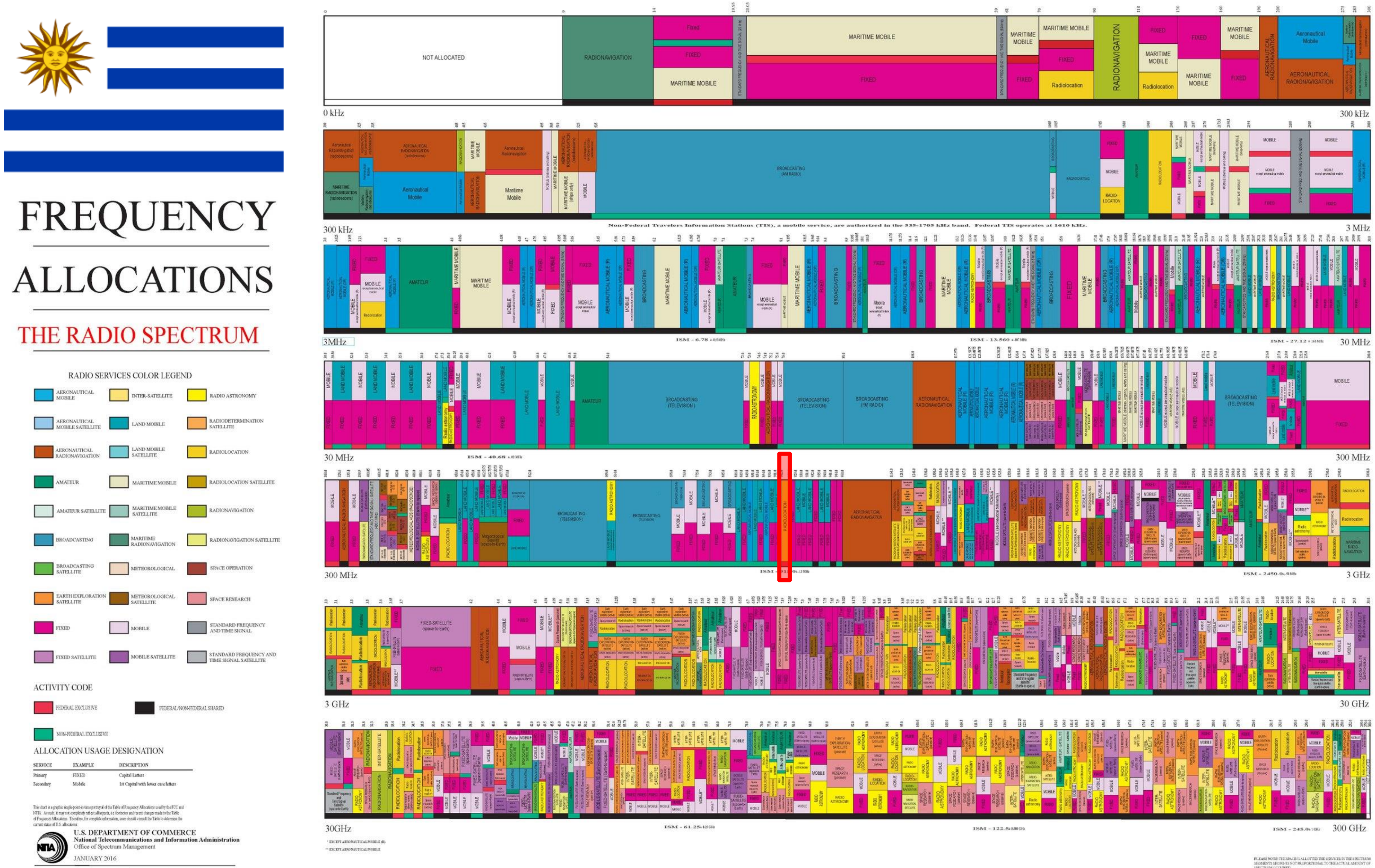
Fig. LPWAN Architecture

```
+----------+
|End Device| * * *
+----------+       *   +---------+
                     * | Gateway +---+
+----------+       *   +---------+   |   +---------+
|End Device| * * *                   +---+ Network +--- Application
+----------+       *                 |   | Server  |
                     * +---------+   |   +---------+
+----------+       *   | Gateway +---+
|End Device| * * *   * +---------+
+----------+
    Key: *      LoRaWAN Radio
       +---+    IP connectivity
```

Fig. LoRaWAN Architecture

LoRA PHY

Shared Spectrum@915MHz

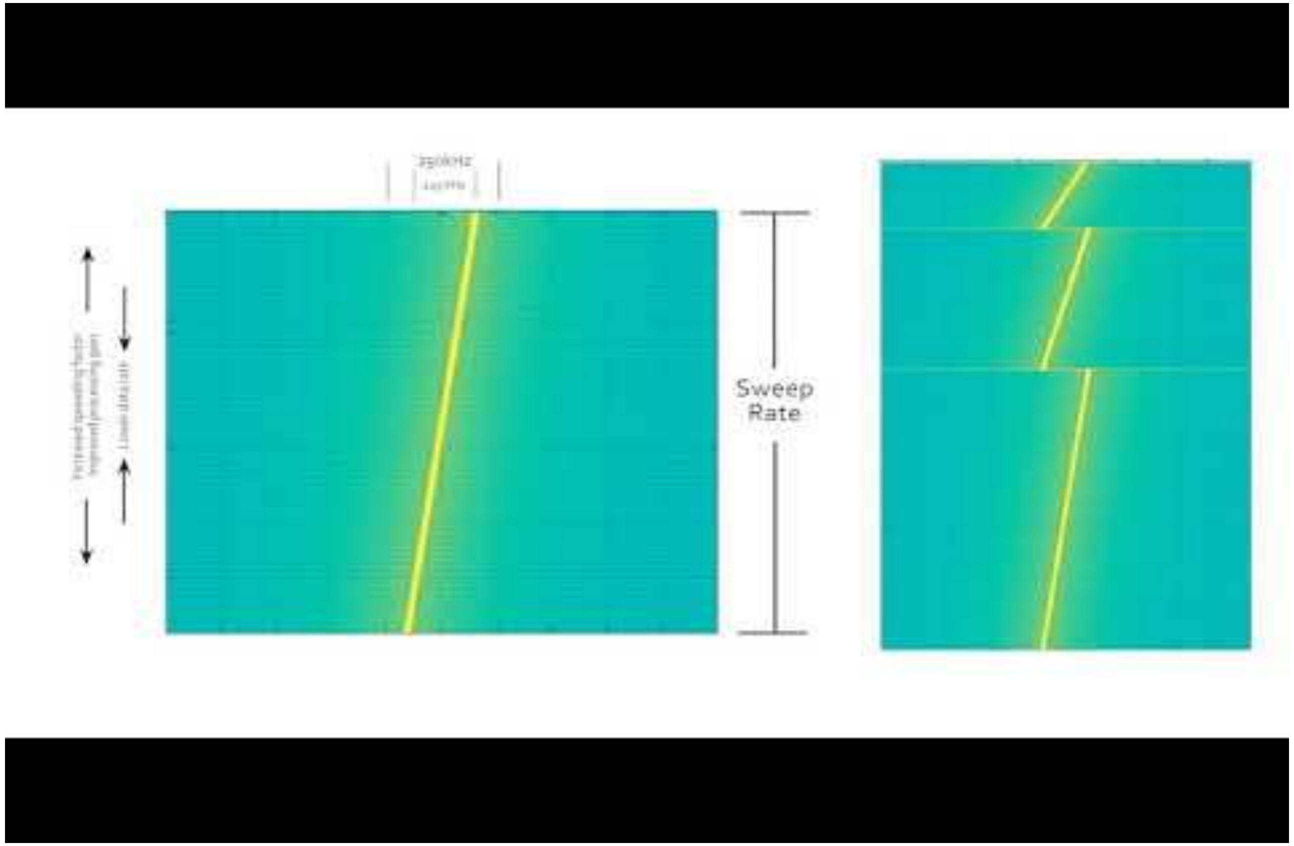


LoRA PHY

Chirp Spread Spectrum

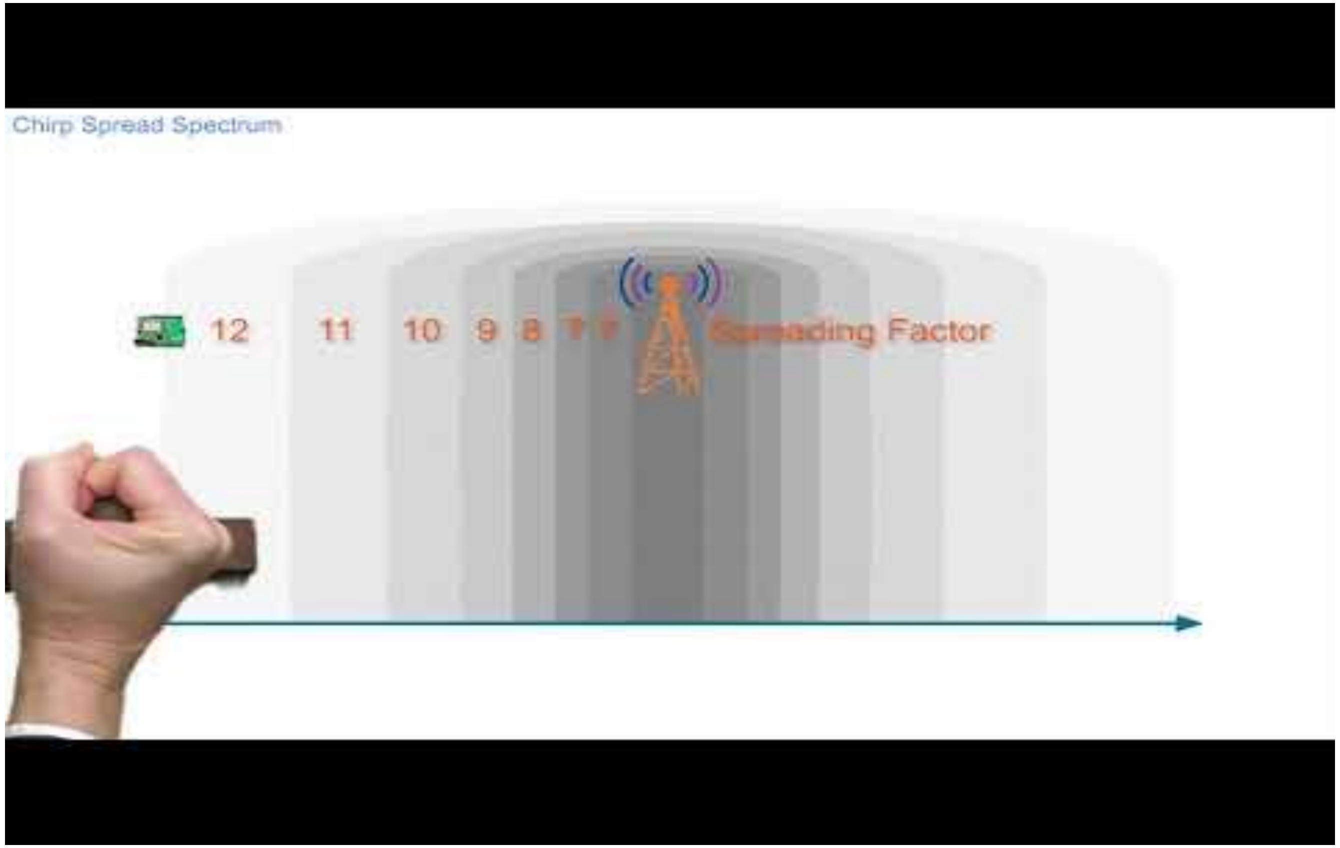
- Semtech's proprietary modulation
  - Transceivers: SX127X (Device) , SX1302 (GWs)
  - Has been reverse engineered (SDR)*
  - Code: https://github.com/Lora-net/sx1302_hal

- Chirp Spread Spectrum
  - Let's see a video... (3 min)
  - https://www.youtube.com/watch?v=dxYY097QNs0



* A. Marquet et al. "Towards an SDR implementation of LoRa: Reverse-engineering, demodulation strategies and assessment over Rayleigh channel" https://hal.archives-ouvertes.fr/hal-02485052

- Spreading Factors and Data Rate
  - Yep... another one (2min) -- https://www.youtube.com/watch?v=B580NvdXtjs



Data Rate	Configuration	Indicative Physical Bit Rate [bit/sec]
0	LoRa: SF12 / 125 kHz	250
1	LoRa: SF11 / 125 kHz	440
2	LoRa: SF10 / 125 kHz	980
3	LoRa: SF9 / 125 kHz	1760
4	LoRa: SF8 / 125 kHz	3125
5	LoRa: SF7 / 125 kHz	5470
6	LoRa: SF8 / 500 kHz	12500
7	LR-FHSS CR1/3: 1.523 MHz BW	162
8	LoRa: SF12 / 500 kHz	980
9	LoRa: SF11 / 500 kHz	1760
10	LoRa: SF10 / 500 kHz	3900
11	LoRa: SF9 / 500 kHz	7000
12	LoRa: SF8 / 500 kHz	12500
13	LoRa: SF7 / 500 kHz	21900
14	RFU	
15	Defined in [TS001][27]	

Table 31: AU915-928 data rate

LoRaWAN

Introduction

- LoRaWAN is a Media Access Control (MAC) layer protocol built on top of LoRa modulation
- Defined by "LoRa Alliance" -- https://resources.lora-alliance.org/technical-specifications
- Nice Intro: https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/

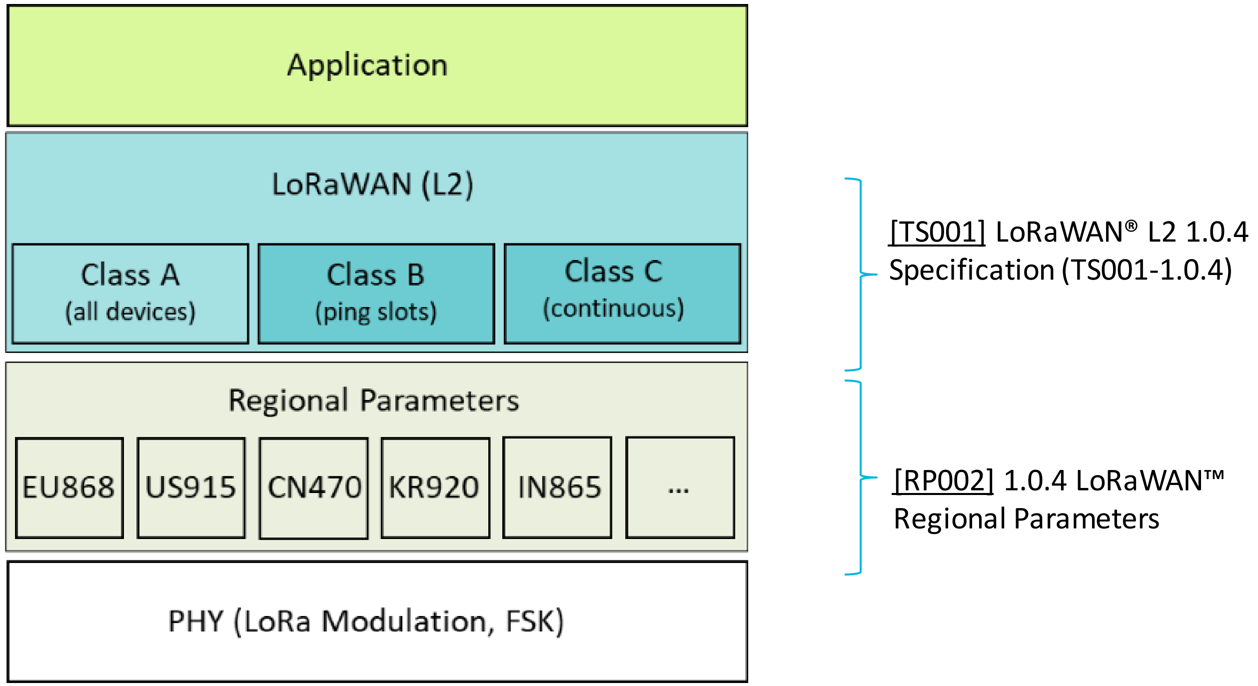
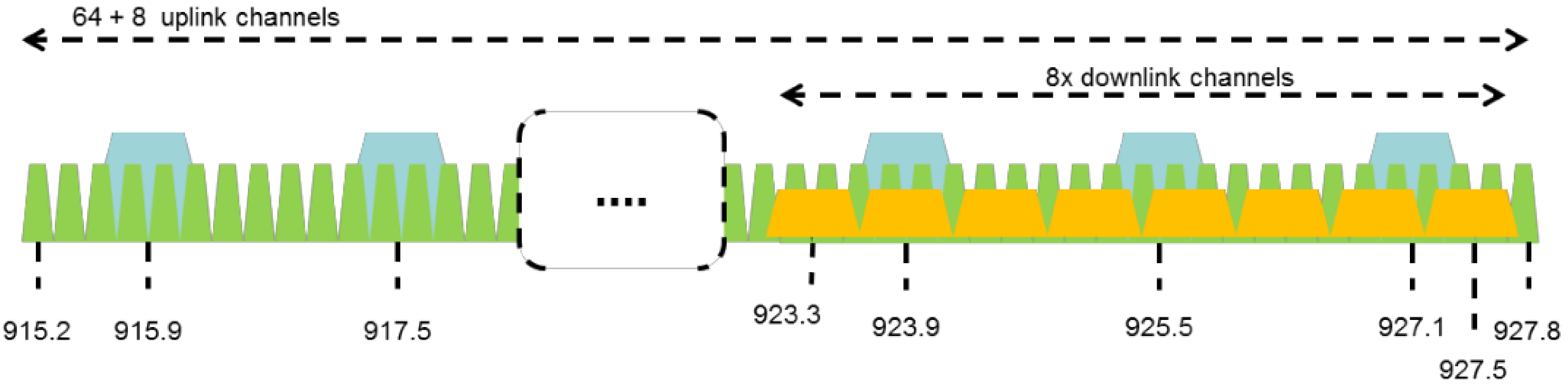


Fig. LoRaWAN Stack

Uruguay: AU915-928 MHz Band (Also AR, BR, CL, PY; not BO)



AU915-928 channel frequencies

The AU915-928 band SHALL be divided into the following channel plans:

- Upstream – 64 channels, numbered 0 to 63, utilizing LoRa 125 kHz BW, varying from DR0 to DR5, using coding rate 4/5, starting at 915.2 MHz and incrementing linearly by 200 kHz to 927.8 MHz
- Upstream – 8 channels, numbered 64 to 71, utilizing LoRa 500 kHz BW at DR6 or LR-FHSS 1.523 MHz BW at DR7, starting at 915.9 MHz and incrementing linearly by 1.6 MHz to 927.1 MHz
- Downstream – 8 channels, numbered 0 to 7, utilizing LoRa 500 kHz BW at DR8 to DR13), starting at 923.3 MHz and incrementing linearly by 600 kHz to 927.5 MHz

LoRaWAN
Architecture

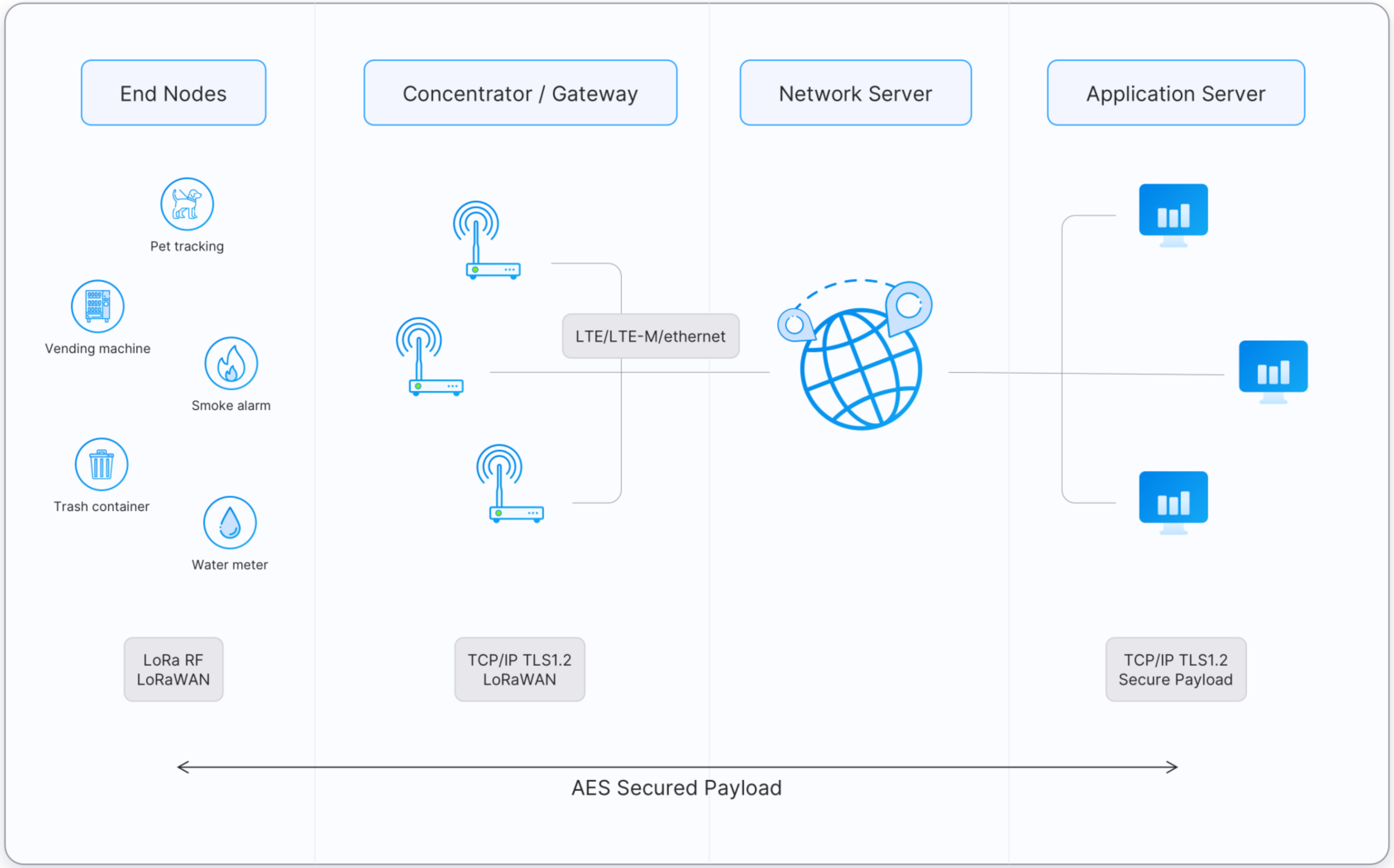


Fig. A tipical LoRAWAN Architecture
(https://www.thethingsnetwork.org/docs/lorawan/architecture/)

- ED must be registered with a network before sending and receiving messages
  - Over-The-Air-Activation (OTAA) : Dynamic Dev. Address and security keys

  - **Activation By Personalization (ABP)**: hardcoding device address and the security keys (ties an end-device to a pre-selected network, bypassing the OTAA procedure)

Fig. Activation By Personalisation in LoRaWAN 1.0.x
(https://www.thethingsnetwork.org/docs/lorawan/end-device-activation/)

(COMPACT) DATA REPRESENTATION

- Data Serialization over the Network
  - Every bit counts!

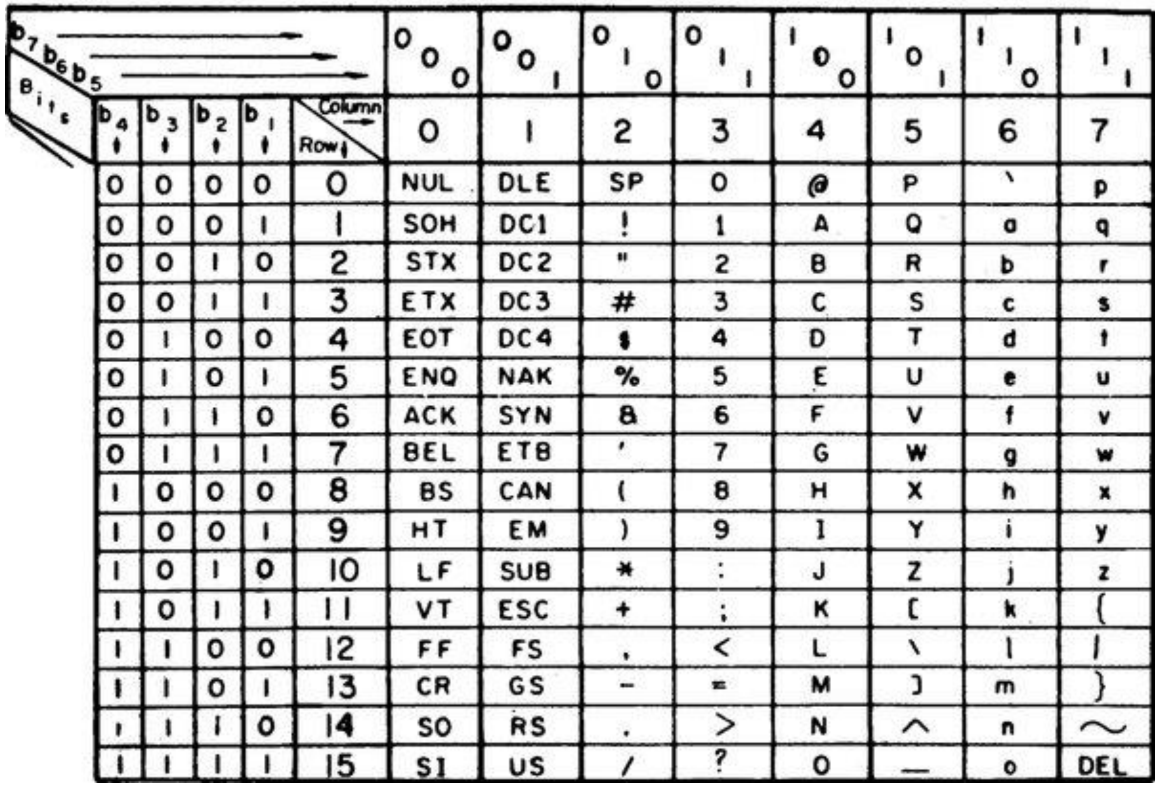
- What formats you know?
  - ASCII, Base64

- Ex. Integer $(10)_{10}$
  - ≠ internal Rep 32-bits, 64-bits
  - And if you send two integers? Delimiters

- TLV (type-length-value)

- Tradeoff: compactness vs genericity

USASCII code chart



b7 b6 b5					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
Bits b4	b3	b2	b1	Column / Row	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	—	o	DEL

Data Serialization Formats
XML and JSON

- World Wide Web consortium (W3C) very popular encodings:
  - **HTML** (textual data to screen) and **XML** (application to application)
- Others: ASN.1, YAML, **JSON, CBOR**
  - The JavaScript Object Notation (JSON) Data Interchange Format -- RFC8259
  - Concise Binary Object Representation (CBOR) -- RFC8949

XML	JSON

<consommations>
 	
<machine name="Publique">
<expresso>4</expresso>
<long>11</long>
<capuccino>6</capuccino>
</machine>
 	
<machine name="Recherche">
<gratuite/>
<expresso>25</expresso>
<long>18</long>
<capuccino>19</capuccino>
</machine>
 	
</consommations>
	[
	{"name": "Publique",
	"gratuite": false,
	{
	"expresso": 4,
	"long": 11,
	"capuccino": 6
	}
	}
	{"name": "Recherche",
	"gratuite": true,
	{
	"expresso": 25,
	"long": 18,
	"capuccino": 19
	}
	}
	]

- JSON, 4 data types:
  - Number, Text, Array, and
  - Object (list of key-value pairs)
- JSON Examples
  - [1, -2, 0.3, 4e1] is an array that contains 4 numbers
  - [1, "2", "34"] is an array containing a number and two strings
  - [1, [2, 3 , "4"]] is an array of two elements whose second element is also an array of 3 elements
  - { "color": [34, 16, 3]} is an object that contains an element and the value is an array

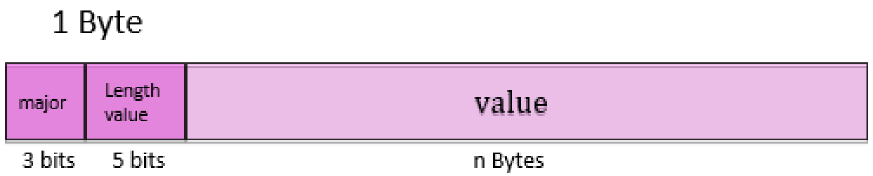
XML-JSON example taken from : https://www.bortzmeyer.org/data-formats.html
JSON examples from [LTN]

Data Serialization Formats
JSON and CBOR

- Compared to XML, JSON is much more permissive and lacks a formalism to describe the structure

- JSON for Linked Data (JSON-LD) defined by the W3C reinforces the interoperability of JSON by introducing specific keys describing the data structure, a reference to units, etc.

- CBOR (RFC8949) is based on JSON, and changes ASCII for binary representation
  - Ex. 123 in ASCII is 3 bytes (one byte per character), but in binary we could use only one byte $(0111\ 1011)_2$
  - CBOR defines 8 major types which are represented by the first 3 bits of a CBOR structure
  - http://cbor.io/



0 : Positive Integer
1 : Negative Integer
2 : Byte string
3 : Text string
4 : Array
5 : Object list
6 : Optional semantic tagging
7 : Simple value and float

Fig. CBOR Major Definitions

0	1	01
1	10	0a
2	100	1864
3	1000	1903e8
4	10000	192710
5	100000	1a000186a0
6	1000000	1a000f4240
7	10000000	1a00989680
8	100000000	1a05f5e100
9	1000000000	1a3b9aca00
10	10000000000	1b00000002540be400
11	100000000000	1b000000174876e800
12	1000000000000	1b000000e8d4a51000

Fig. CBOR encoding of some Integer values

Data Serialization Formats

JSON vs CBOR – https://cbor.me

CBOR playground. See RFC 8949 for the CBOR specification, and cbor.io for more background informatio

CBOR

Diagnostic ☐plain hex

26 Bytes ☐as text ☐utf8 ☐emb cbor ☐cborseq

enter hex below or Browse... No file selected.

```
{
  "on": true ,
    0 : [1,1,1],
"rssi": -75,
  "t" : 31.5 ,
  "h" : 70
 }
```

bytes 74 Bytes


```
7b 0a 20 20 22 6f 6e 22 3a 20 74 72 75 65 20 2c 0a 20 20
20 20 30 20 3a 20 5b 31 2c 31 2c 31 5d 2c 0a 22 72 73 73
69 22 3a 20 2d 37 35 2c 0a 20 20 22 74 22 20 3a 20 33 31
2e 35 20 2c 0a 20 20 22 68 22 20 3a 20 37 30 20 0a 7d
```


```
A5                # map(5)
   62             # text(2)
      6F6E        # "on"
   F5             # primitive(21)
   00             # unsigned(0)
   83             # array(3)
      01          # unsigned(1)
      01          # unsigned(1)
      01          # unsigned(1)
   64             # text(4)
      72737369    # "rssi"
   38 4A          # negative(74)
   61             # text(1)
      74          # "t"
   F9 4FE0        # primitive(20448)
   61             # text(1)
      68          # "h"
   18 46          # unsigned(70)
```

Cayenne Low Power Payload (LPP)

- https://docs.mydevices.com/docs/lorawan/cayenne-lpp
- Not from a standard. body, but heavily used in Chirpstack/TTN/LoRaWAN community

Each sensor data must be prefixed with two bytes:

- **Data Channel**: Uniquely identifies each sensor in the device across frames (can be used for multiplexing)
- **Data Type**: Identifies the data type in the frame; e.g., Temperature Sensor (0x67)

Payload structure

1 Byte	1 Byte	N Bytes	1 Byte	1 Byte	M Bytes	...
Data1 Ch.	Data1 Type	Data1	Data2 Ch.	Data2 Type	Data 2	...

Cayenne LPP

Data Types

Type	IPSO	LPP	Hex	Data Size	Data Resolution per bit
Digital Input	3200	0	0	1	1
Digital Output	3201	1	1	1	1
Analog Input	3202	2	2	2	0.01
Analog Output	3203	3	3	2	0.01
Illuminance Sensor	3301	101	65	2	1
Presence Sensor	3302	102	66	1	1
Temperature Sensor	3303	103	67	2	0.1°C Signed MSB
Humidity Sensor	3304	104	68	1	0.5 % Unsigned
Accelerometer	3313	113	71	6	0.001 G Signed MSB per axis
Barometer	3315	115	73	2	0.1 hPa Unsigned MSB
Gyrometer	3334	134	86	6	0.01 °/s Signed MSB per axis
GPS Location	3336	136	88	9	Latitude : 0.0001 ° Signed MSB
GPS Location	3336	136	88	9	Longitude : 0.0001 ° Signed MSB
GPS Location	3336	136	88	9	Altitude : 0.01 meter Signed MSB

Cayenne LPP

Examples

Device with 2 temperature sensors

Payload (Hex)	**03 67 01 10 05 67 00 FF**	
Data Channel	Type	Value
03 ⇒ 3	67 ⇒ Temperature	0110 = 272 ⇒ 27.2°C
05 ⇒ 5	67 ⇒ Temperature	00FF = 255 ⇒ 25.5°C

Device with temperature and acceleration sensors

Frame N

Payload (Hex)	**01 67 FF D7**	
Data Channel	Type	Value
01 ⇒ 1	67 ⇒ Temperature	FFD7 = -41 ⇒ -4.1°C

Frame N+1

Payload (Hex)	**06 71 04 D2 FB 2E 00 00**	
Data Channel	Type	Value
06 ⇒ 6	71 ⇒ Accelerometer	X: 04D2 = +1234 => + 1.234G
06 ⇒ 6	71 ⇒ Accelerometer	Y: FB2E = -1234 => - 1.234G
06 ⇒ 6	71 ⇒ Accelerometer	Z: 0000 = 0 => 0G

LAB: USE CASE

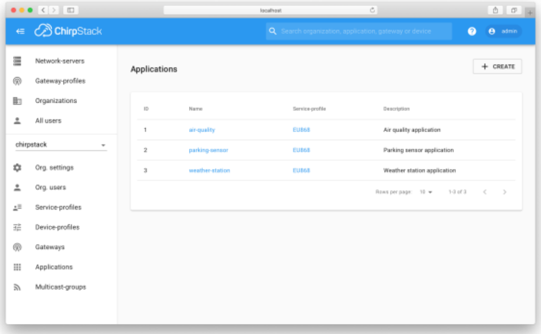
Use Case

Objective: "digital twin" of the IEEE 802.11 wireless spectrum of an area

- **Measure a WiFi 802.11 AP signal in several spots**
  - LoRa is used to transport the measured data
- Import this empirical data in the ns-3 simulator
  - (Work/tune the model with empirical data. +Partial validation?)
- Use the model/twin to predict unobserved phenomena (TBD)

LAB: HW/SW AND ARCHITECTURE

UDP+JSON+
[MAC+**CLPP**]



HTTP+JSON

GW
(Dragino)

ChirpStack

ThingsBoard

ED – LoPy4
(IoT node)

LNS+AS

APP

LoRaWAN®

LoPy4

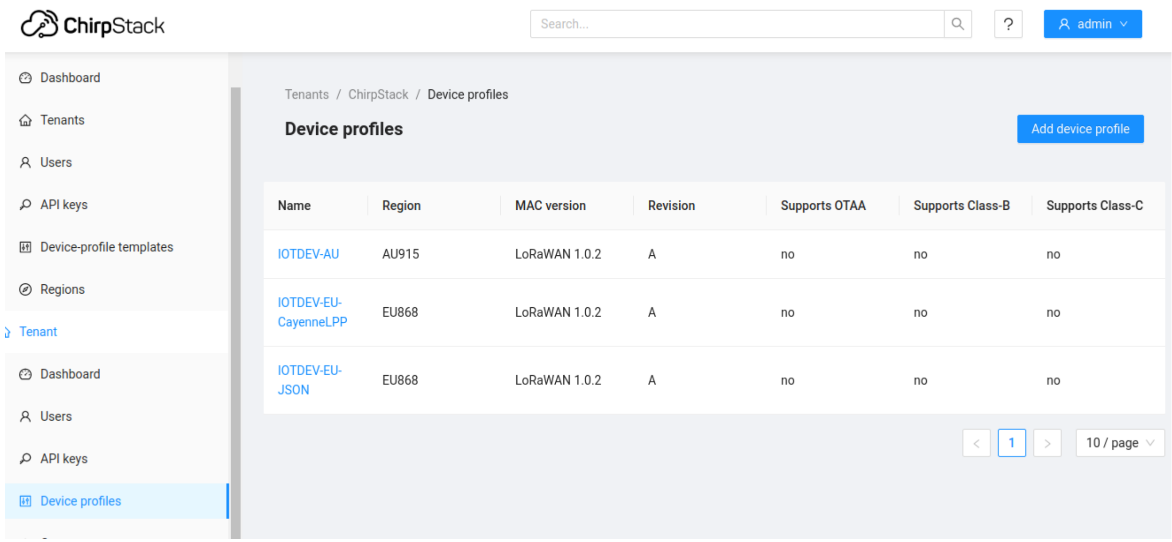
ED: LoPy4

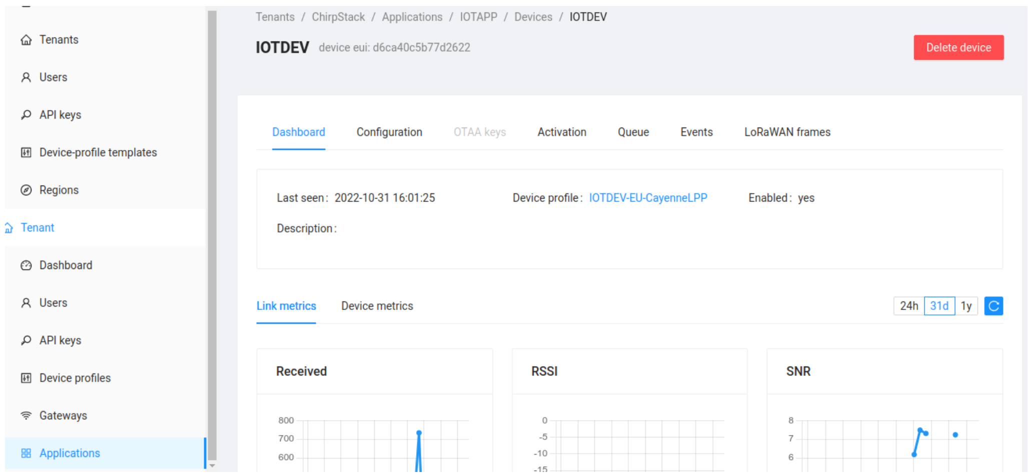
- Quite powerful (>= Class 2) ESP32
- RF: WiFi, LoRA, BT, Sigfox
- Runs micropython -- https://micropython.org/
- Tutorials:
  - https://docs.pycom.io/gettingstarted/
  - https://alepycom.gitbooks.io/pycom-documentation/

- SW Install Visual Studio Code + Node.js
  - Examples code : https://github.com/renzoe/2022-10-WSCF2022PUB
  - We have to configure the LoRaWAN's Data Rate/SF, Join Keys, etc.

Chirpstack

APP and Device Configuration



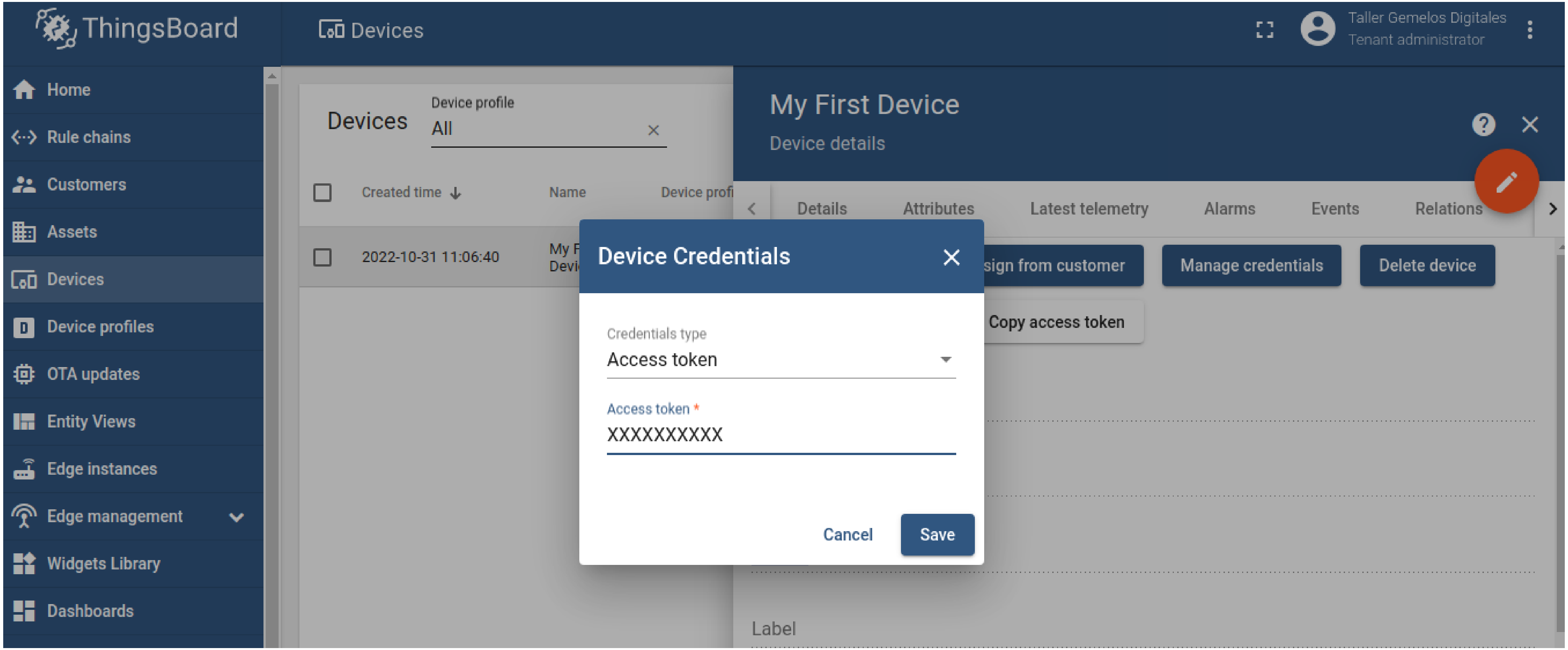


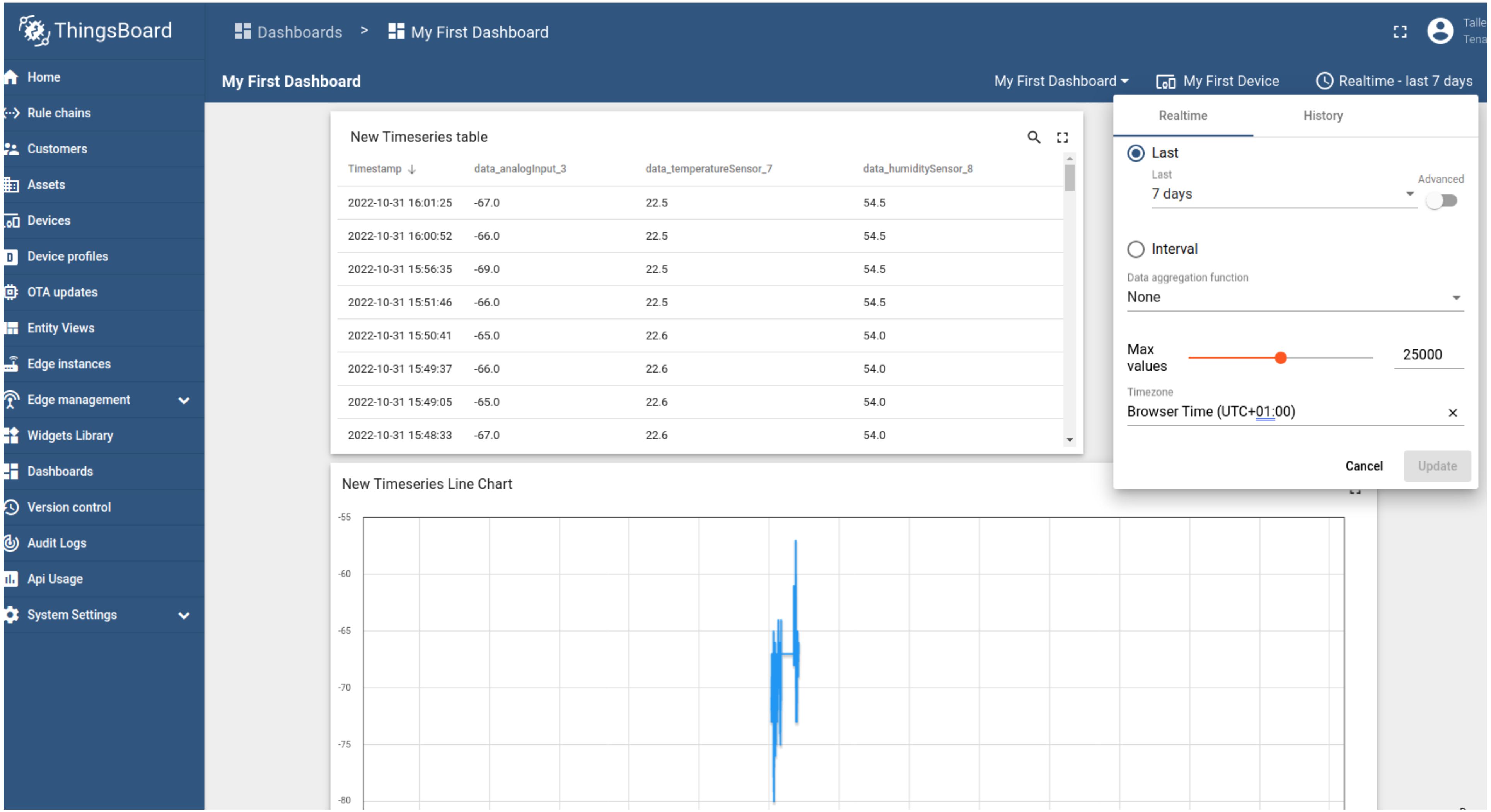
Checklist

- Add Device Profile
  - Region, Codec (Cayenne LPP)
- Add Application
  - Integration TTB
- Add Device
  - Activation (Dev Addr, NSKey, AppSKey)
  - Config : Disable Frame Cnt Validation
  - Integration TTB : ThingsBoardAccessToken (Conf->Variables)

The Things Board

Device Credentials




ThingsBoard
Dashboards > My First Dashboard
Home
Rule chains
Customers
Assets
Devices
Device profiles
OTA updates
Entity Views
Edge instances
Edge management
Widgets Library
Dashboards
Version control
Audit Logs
Api Usage
System Settings
My First Dashboard
My First Dashboard
My First Device
Realtime - last 7 days
New Timeseries table
Timestamp
data_analogInput_3
data_temperatureSensor_7
data_humiditySensor_8
2022-10-31 16:01:25 -67.0 22.5 54.5
2022-10-31 16:00:52 -66.0 22.5 54.5
2022-10-31 15:56:35 -69.0 22.5 54.5
2022-10-31 15:51:46 -66.0 22.5 54.5
2022-10-31 15:50:41 -65.0 22.6 54.0
2022-10-31 15:49:37 -66.0 22.6 54.0
2022-10-31 15:49:05 -65.0 22.6 54.0
2022-10-31 15:48:33 -67.0 22.6 54.0
Realtime
History
Last
Last
7 days
Advanced
Interval
Data aggregation function
None
Max values
25000
Timezone
Browser Time (UTC+01:00)
Cancel
Update
New Timeseries Line Chart
-55
-60
-65
-70
-75
-80

QUESTIONS?
LET'S GO WORK!

Bibliography

- [LTN] Laurent TOUTAIN "Programming the Internet of Things" https://github.com/ltn22/PLIDO_BOOK (ver 18 OCT 2022)
- [REN] Renzo E. NAVAS "Improving the resilience of the constrained Internet of Things: a moving target defense approach." PhD diss., Ecole nationale supérieure Mines-Télécom Atlantique, 2020. https://tel.archives-ouvertes.fr/tel-03123143/