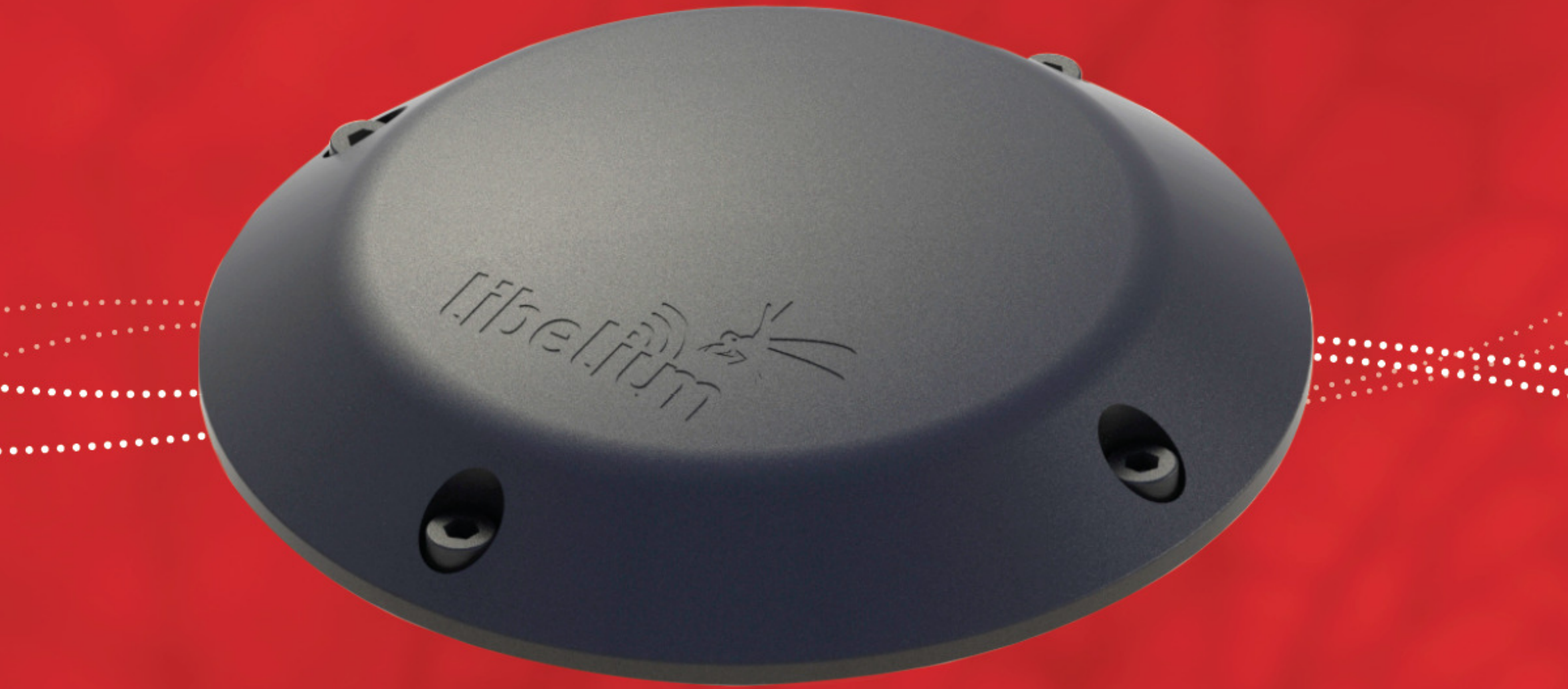
Smart Parking

Technical Guide



Document version: v7.0 - May 28, 2019

INDEX

1. Introduction

The Smart Parking v2 solution developed by Libelium allows citizens to detect available parking slots.

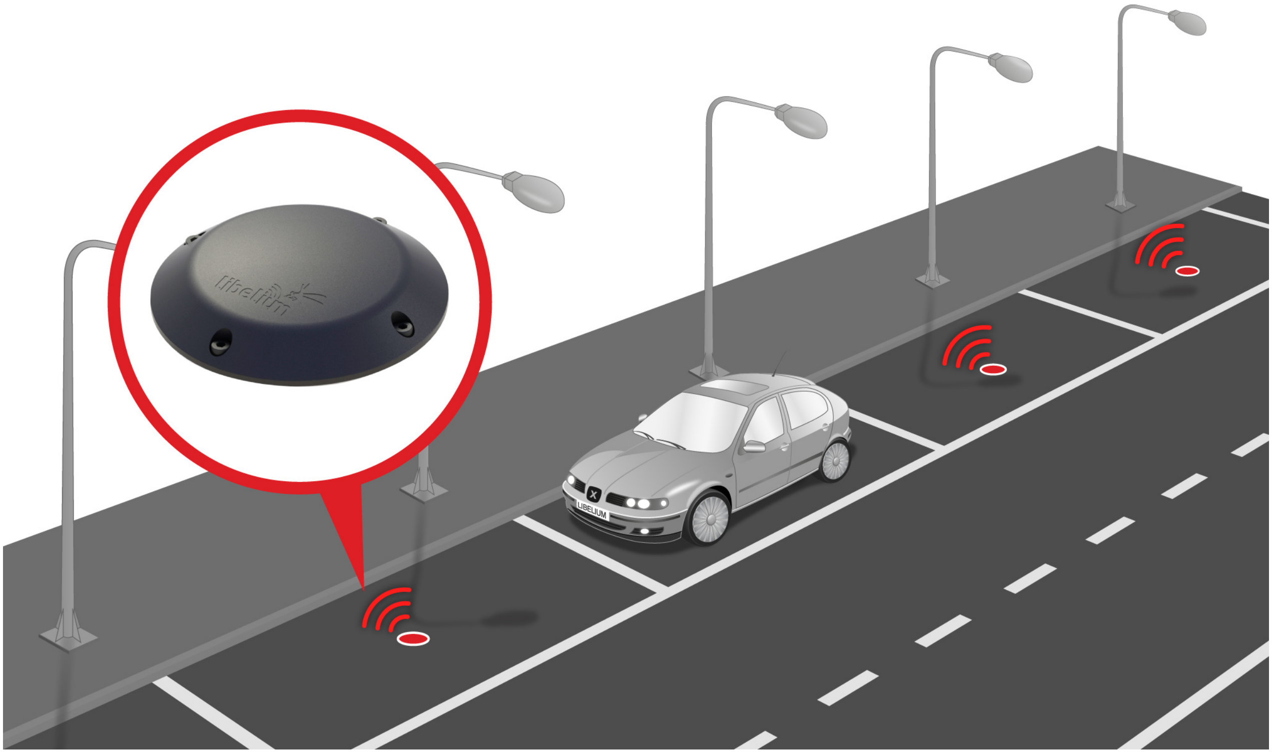


Figure : Smart Parking node developed by Libelium

The node applies intelligent algorithms to detect changes in the state of the parking slot. Then data is transmitted with the LoRaWAN radio to the final server.

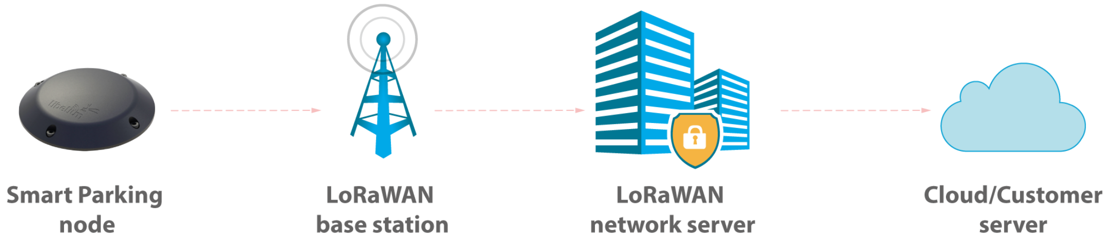


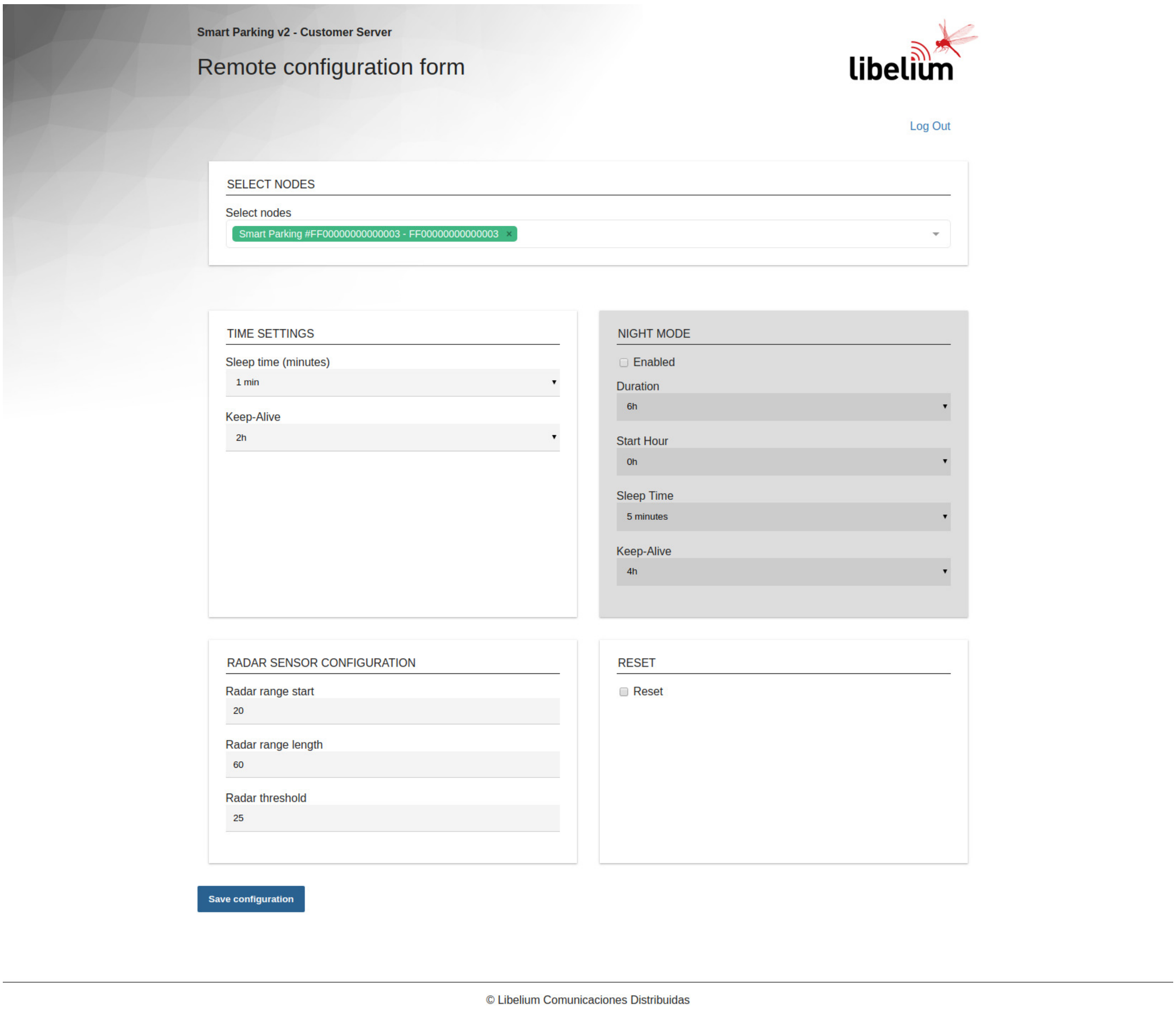
Figure : Simplified Smart Parking network architecture

The nodes provisioning has been enormously improved. The nodes are delivered with default time settings and also unique LoRaWAN identifiers and keys. So it is easy to use the default settings to register all nodes in the LoRaWAN network server at a time.

The Smart Parking node improves the detection and stability performance thanks to a radar sensor which permits to certainly know when objects are placed over the device. The next table shows a comparative analysis of the current sensor technologies in the Smart Parking market:

	Radar	Magnetometer	Infra-red
Immunity from vehicles adjacent to empty space	✓	✗	✓
Robust RF performance	✓	✗	✗
Ignores nearby double parked vehicles	✓	✗	✓
Accommodates dirt, dust or oil on sensor	✓	✓	✗
Not vulnerable to electromagnetic interference	✓	✗	✓
Consistent performance in all lighting conditions	✓	✓	✗
Reliable state during long-duration stays	✓	✗	✓
Unaffected by nearby slow moving traffic	✓	✗	✓

The node provides OTA-S (Over-The-Air Setup). This allows the user to remotely configure the node parameters (sleep time, keep-alive, night-mode, etc) via the Remote Configuration Form. That makes it possible to directly install the nodes with factory default settings and then update them from the server side.



Smart Parking v2 - Customer Server

Remote configuration form

SELECT NODES

Select nodes: Smart Parking #FF00000000000003 - FF00000000000003

TIME SETTINGS

Sleep time (minutes): 1 min

Keep-Alive: 2h

NIGHT MODE

□ Enabled

Duration: 6h

Start Hour: 0h

Sleep Time: 5 minutes

Keep-Alive: 4h

RADAR SENSOR CONFIGURATION

Radar range start: 20

Radar range length: 60

Radar threshold: 25

RESET

□ Reset

© Libelium Comunicaciones Distribuidas

Figure : Remote Configuration Form

2. Network architecture

The network architecture of Smart Parking is based on the next elements:

- Smart Parking node
- LoRaWAN base station
- LoRaWAN Network Server
- Libelium Cloud or Customer Server

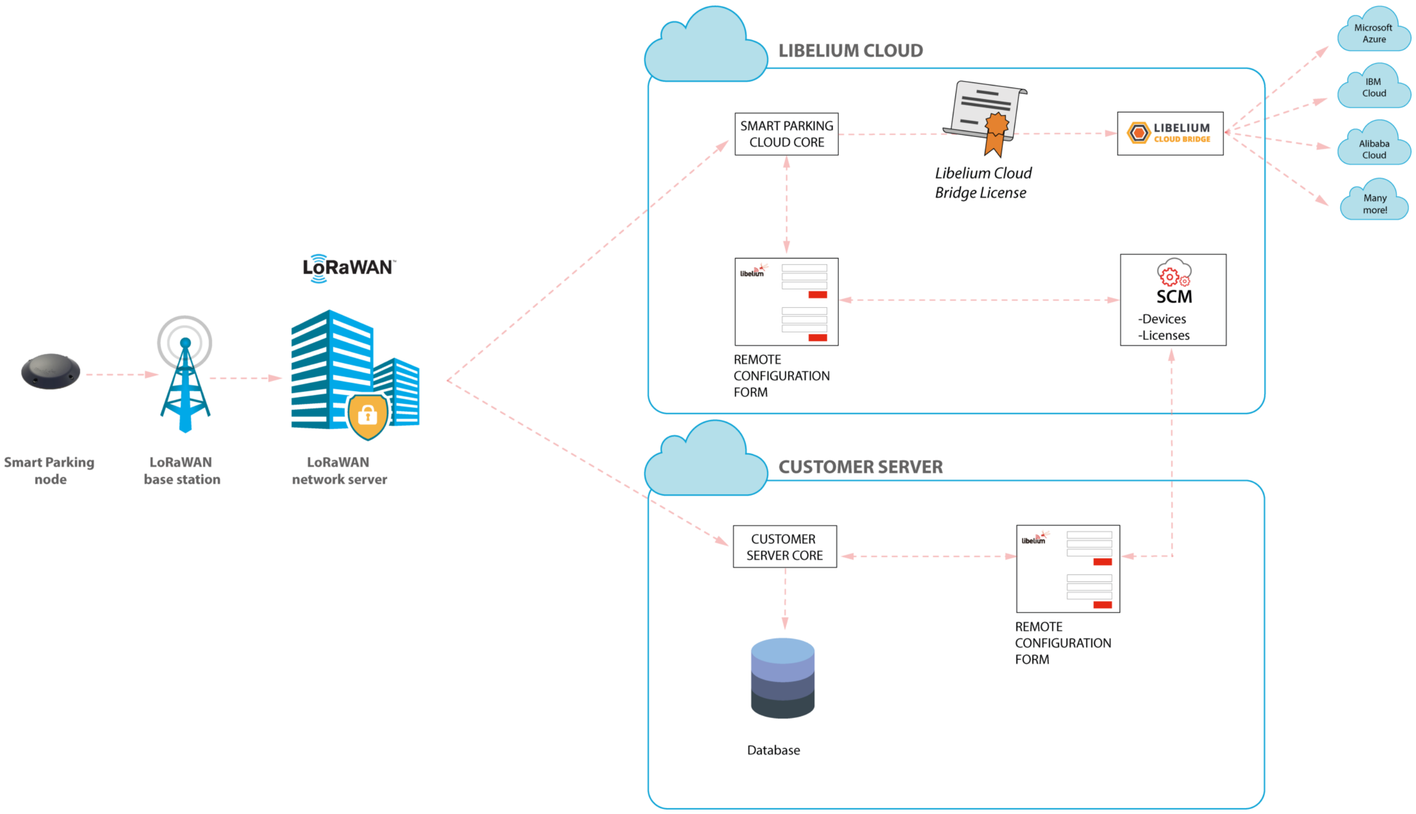


Figure : Smart Parking network architecture

2.1. Smart Parking node

The Smart Parking node is the device installed in each parking slot. When the device detects a change of the parking slot status (free/occupied), it sends a frame to the LoRaWAN base station.

Figure : Smart Parking node

2.2. LoRaWAN base station

The LoRaWAN base station (also known as gateway) must be installed in the surrounding area next to the parking nodes. It receives data and forwards it to the LoRaWAN Network Server.

Libelium distributes base stations for LoRaWAN networks. All of them have LoRaWAN connection; some feature Ethernet, WiFi or 4G connectivity too. Some base stations are ready to work outdoors (IP67 grade). Some of them come pre-configured for certain LoRaWAN network servers (see next section). Also, some of them integrate an embedded LoRaWAN Network Server.

Figure : LoRaWAN base station

2.3. LoRaWAN Network Server

The nodes registration must be done in the Network Server in order to receive LoRaWAN data from all nodes in the network. Each node must be registered with an identifier and some encryption keys so the Network Server can receive and decrypt the packets successfully.

The LoRaWAN Network Server purpose is to translate data from the LoRaWAN wireless network to an IP network. Therefore, when Smart Parking nodes packets are received, a callback is performed in order to send data to the

Libelium Cloud or to the Customer Server.

2.4. Libelium Cloud and Customer Server

The LoRaWAN Network Server connects to the final server, which can be the Libelium Cloud or the Customer Server.

The LoRaWAN network servers currently supported are:

- Loriot
- Actility

If the customer wants to use a new LoRaWAN Network Server, then the Data Parser block must be modified in order to receive data properly. Keep in mind that each Network Server implements their own HTTPS callback using a different format.

The Remote Configuration Form allows the user to update the settings of each node (sleep time, keep-alive time, night-mode, etc). The update is done remotely via LoRaWAN downlink radio packets.

The difference between the 2 types of server differ in the the possible client needs:

- The Customer Server is a software system provided by Libelium which permits to receive, decode and insert data into a standard MySQL database. It is mandatory that the user sets up her own server to host the Customer Server. See "Customer Server" chapter for further information.
- The Libelium Cloud is a service provided by Libelium which permits to receive, decode and redirect the data to the final 3rd party IoT cloud (Amazon, Azure, etc). This retransmission is done thanks to the cloud connectors running on the Libelium Cloud Bridge.

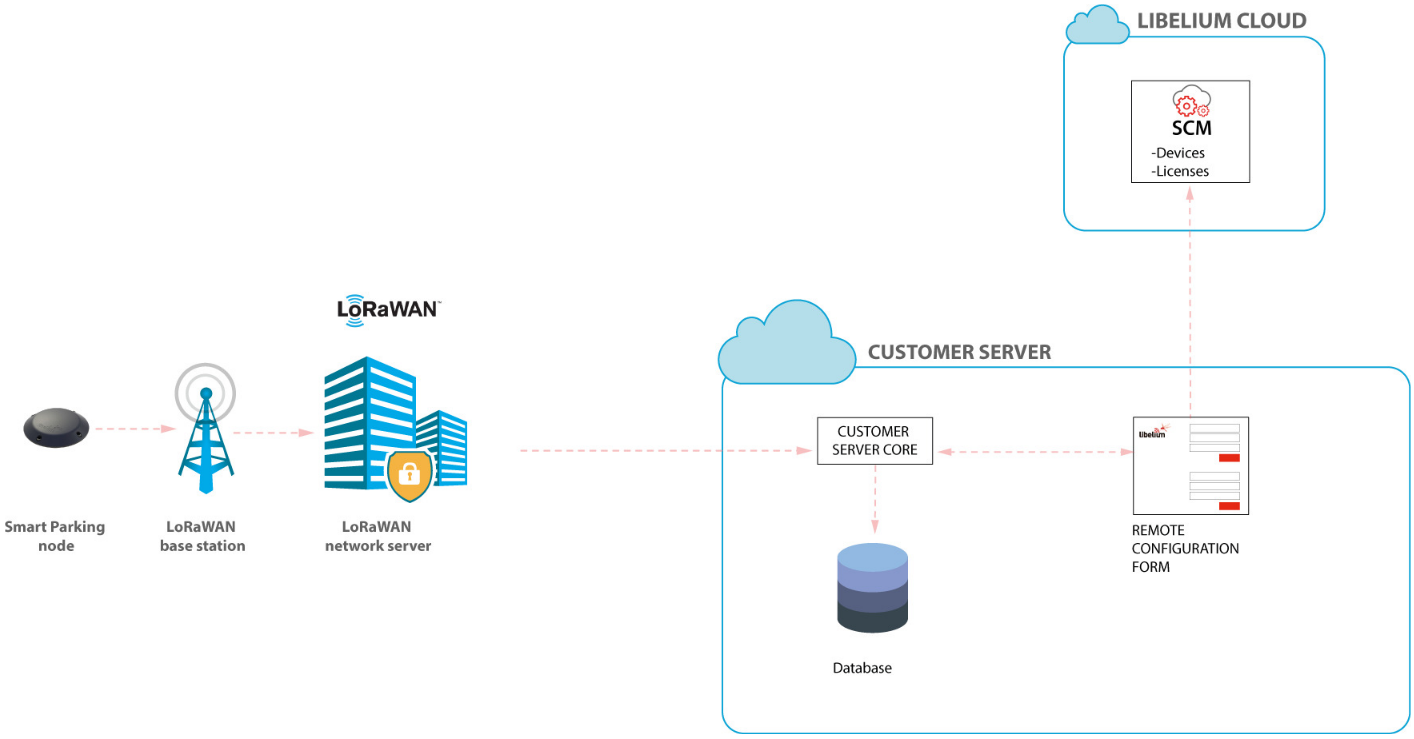


Figure : Customer Server scenario

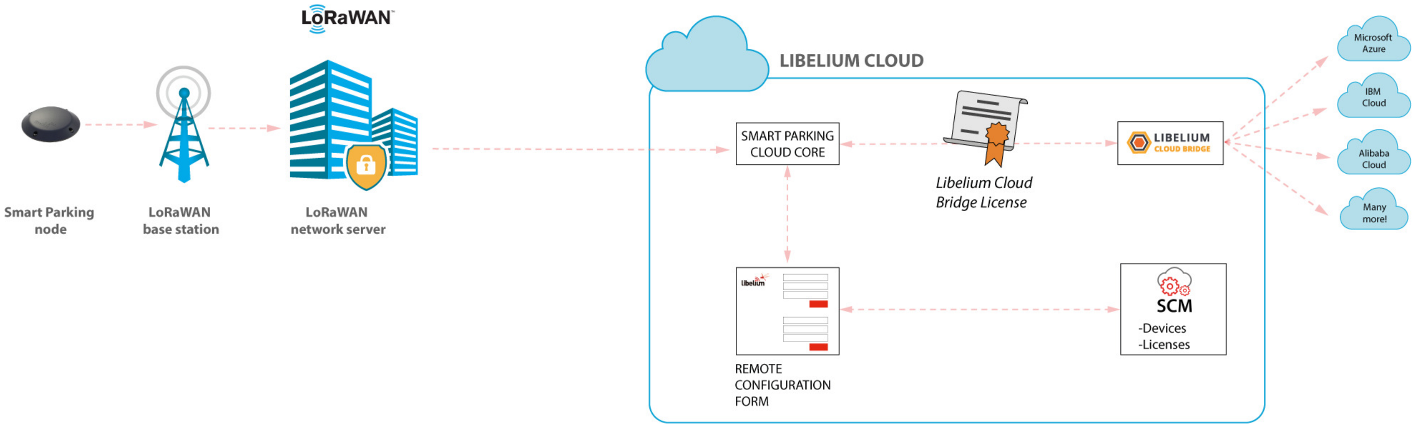


Figure : Libelium Cloud service scenario

3. Smart Parking node

3.1. Hardware description

The Smart Parking node is based on 2 different pieces: the base and the external enclosure. The base of the Smart Parking node includes the PCB, the battery, the antenna and the internal enclosure piece.

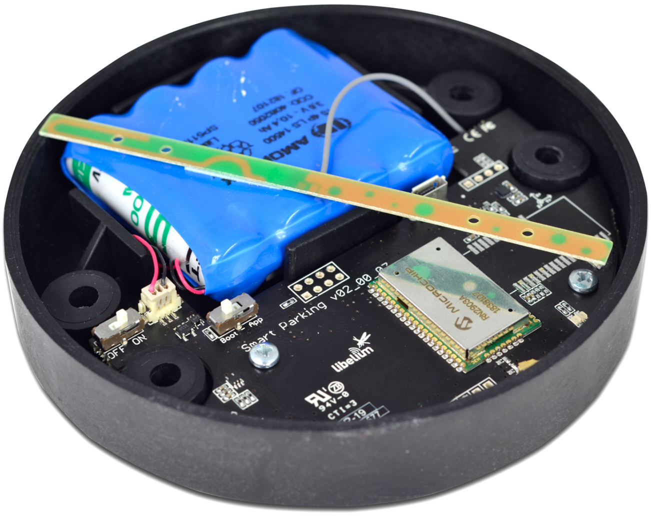


Figure : Base of a Smart Parking node

The base is screwed to the external enclosure piece:

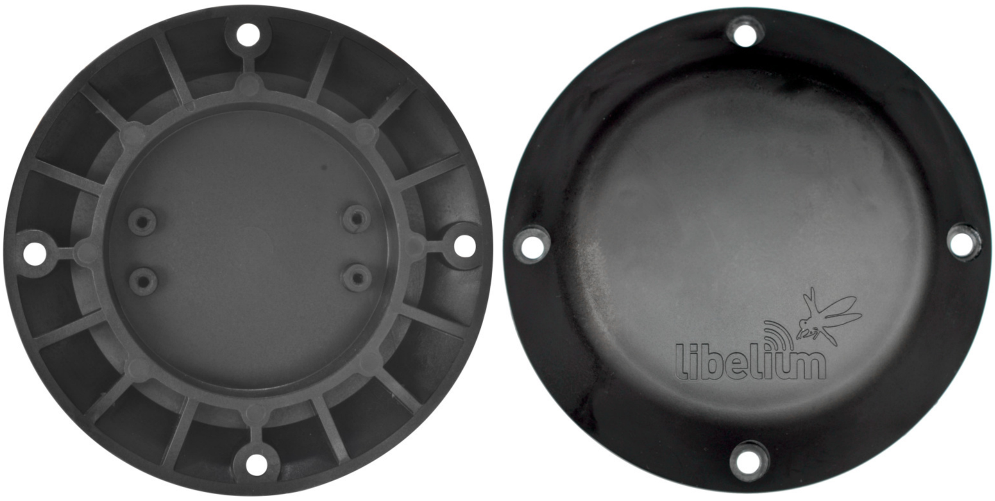


Figure : External enclosure

The next table shows the basic Smart Parking node characteristics.

Power supply	Built-in lithium-thionyl chloride (Li-SOCl2) batteries; expected lifetime of 4-10 years*
Configurable sleep time	Min: 20 s / max: 10 min
Radio protocol	LoRaWAN module
Dual detection	Radar (main) and magnetic (backup)
Provisioning	Ready to install (default LoRaWAN OTAA IDs and key are loaded to each node)
Node configuration	Via "Remote Configuration Form" (web form)
LoRaWAN configuration	Via "Smart Devices App" (Java desktop application)
Operating temperature	-20 to +65 °C

(*) Under normal circumstances and depending on settings

3.1.1. Node versions

Libelium provides the next versions of Smart Parking:

Reference	Version	Operating frequency
SP-EU	Smart Parking EU	863.0 to 870.0 MHz (LoRaWAN EU863-870)
SP-US	Smart Parking US	902.0 to 928.0 MHz (LoRaWAN US902-928)
SP-APLA-AU	Smart Parking APAC / LATAM / AU / AU915	915.2 to 927.8 MHz (LoRaWAN AU915-928)
SP-IN	Smart Parking IN	865.0 to 867.0 MHz (LoRaWAN IN865-867)
SP-APLA-AS	Smart Parking APAC / LATAM / AU / AS923	923 MHz (LoRaWAN AS923)

3.1.2. LoRaWAN regions

The Smart Parking node supports the next LoRaWAN regions:

LoRaWAN region	Supported by
EU 863-870 MHz ISM Band (Europe)	Smart Parking EU
US 902-928 MHz ISM Band (United States)	Smart Parking US
AU 915-928 MHz ISM Band (Australia)	Smart Parking APAC / LATAM / AU / AU915
IN 865-867 MHz ISM Band (India)	Smart Parking IN
AS 923 MHz ISM Band (Asia and ASEAN region)	Smart Parking APAC / LATAM / AU / AS923
CN 779-787 MHz ISM Band (China)	Not available
CN 470-510 MHz ISM Band (China)	Not available
KR 920-923 MHz ISM Band (South Korea)	Not available
433 MHz ISM Band (Worldwide)	Not available

If you are interested in further information about LoRaWAN country regulations, please refer to the LoRa Alliance regional parameters document.

3.1.3. LoRaWAN protocol and parameters

LoRaWAN is a Low Power Wide Area Network (LPWAN) protocol. It is a spread-spectrum modulation technique at extremely low data-rates which permits sending data achieving long ranges.
The most important LoRaWAN parameters are:

- LoRaWAN EUI: Read-only, 8-byte, unique identifier which defines each LoRaWAN module in the market.
- Device EUI: Read/write, 8-byte identifier configured into the LoRaWAN module to be used as operating identifier. By default, the "LoRaWAN EUI" of the module is factory-configured as "Device EUI" in the Smart Parking node.
- Join mode: ABP or OTAA. Defines how the module joins the network. Different keys are needed for each method.
- Device address: Needed for ABP. The 4-byte address of the the LoRaWAN module. Must be unique in its own sub-network.
- Network Session Key: Needed for ABP. The 16-byte AES key. Used to generate Message Integrity Check.
- Application Session Key: Needed for ABP. The 16-byte AES key. Used to encrypt data.
- Application EUI: Needed for OTAA. The 8-byte application identifier. Needed for opening an OTAA session and exchange encryption keys.
- Application Key: Needed for OTAA. The 16-byte key. Needed for opening an OTAA session and exchange encryption keys.
- Data-rate: Defines the transmission rate (bits per second). Each data-rate settings combines different Spreading Factor (SF) and bandwidth (BW). By default, all LoRaWAN regions use the same data-rate (DR 0). However, depending on the region, that means different SF and BW:
    - LoRaWAN EU863-870 version: SF12 / 125 kHz
    - LoRaWAN IN865-867 version: SF12 / 125 kHz
    - LoRaWAN AS923 version: SF12 / 125 kHz
    - LoRaWAN US902-928 version: SF10 / 125 kHz
    - LoRaWAN AU915-928 version: SF10 / 125 kHz
- ADR: Adaptive Data Rate setting which can be enabled or disabled. If ADR is enabled, the server will optimize the data-rate based on the information collected from the network: the RSSI / SNR of the last received packets.

If you are interested in further information about LoRaWAN specifications, please refer to the LoRa Alliance specifications document.

3.2. Power and time consumption

The Smart Parking node firmware executes different steps since the node is started. Firstly, the node's setup and then an infinite loop where every cycle is based on measuring, sending if needed and sleeping. The next tables show the power and time consumption of each step modelled as a pulse of a specific time duration and average power consumption.

3.2.1. Smart Parking EU

	Power consumption	Time consumption
Node setup	22.9 mA	59 s
Measure cycle	26 mA	340 ms
Measure and send cycle	17 mA	6 s
Sleep cycle	5.5 uA	Depends on sleep time settings

(*) LoRaWAN EU is set to the default SF12 settings (worst case). The send process may be lower power if the node is close to the base station.

3.2.2. Smart Parking US

	Power consumption	Time consumption
Node setup	21.8 mA	53 s
Measure cycle	26 mA	340 ms
Measure and send cycle	20 mA	3.6 s
Sleep cycle	5.5 uA	Depends on sleep time settings

(*) LoRaWAN US is set to the default SF10 settings (worst case). The send process may be lower power if the node is close to the base station.

3.3. User switches

The Smart Parking node has 2 switches to manage the working mode:

- On/Off switch: Determines whether the node is powered-on or powered-off
- App/Boot switch: When the node is powered-on, this switch determines the performance state of the device
  - App position must be used for a normal operation mode, so the device executes the firmware within it
  - Boot position must be used for configuring purposes only

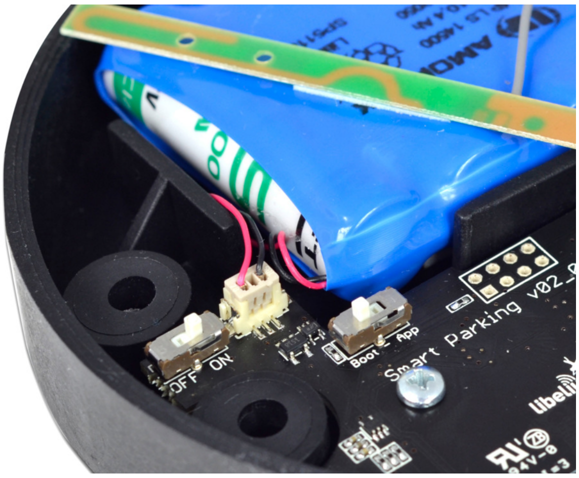


Figure : Smart Parking node "user switches"

When the node is powered-on (On switch), you can change from App to Boot or viceversa by changing the state of the App/Boot switch. However, you must press the reset button to apply the operation mode change. Another possibility to successfully change the operation mode step-by-step would be to: power down the device (Off switch), change the App/Boot switch, press the reset button and then power on the device.

Important:
Never leave the device set to On and Boot for more time than needed. The bootloader does not provide any sleep mode and it will waste the battery of the device. So when you finish reconfiguring the device, please set the node in off state.

3.4. Reset button

The reset button can be used to re-start the node in the corresponding operation mode (App or Boot). If the node is set up to "App" (normal operation mode), pressing the reset button will re-start the program execution. On the other hand, if the node is set up to Boot (configuration mode), pressing the reset button will re-start the MCU bootloader for reconfiguration or firmware update.

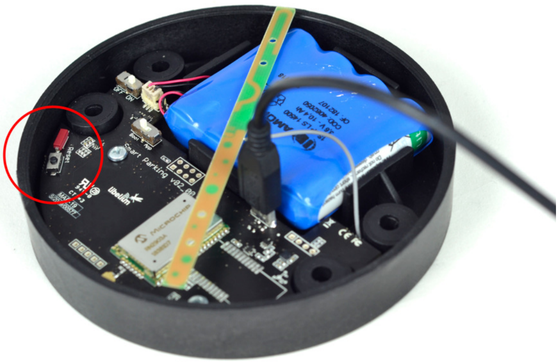


Figure : Reset button

3.5. Node setup

3.5.1. "Ready to install" state

Important:
Libelium provides the nodes "ready to install" so the user only needs to install the nodes and follow the ""Magnet start-up" process".

The Smart Parking node has a power-on process in order to put the device into a "ready-to-install" state:

- Step 1: The switches are set to "App" and "Off" (press the reset button to make sure you discharge capacitors)
- Step 2: You power the device on by sliding the switch from "Off" to "On"
- Step 3: Both LEDs (red and green) blink rapidly for 5 times
- Step 4: Red LED blinks once for 1 second to indicate that the device enters sleep mode for the 1st time. Now the node is in a "ready to install" state. The customer should install the node on the real scenario and perform the "Magnet start-up" process.

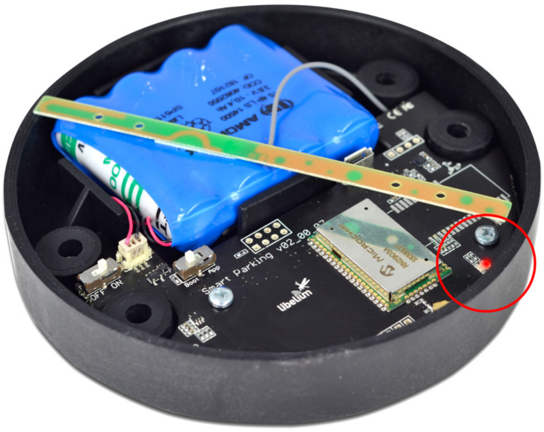


Figure : The red LED blinks once to indicate ready-to-install state

You can see how the previous steps are performed in this video: Ready to install process

3.5.2. How to close the Smart Parking node

After following the previous steps, the device can be closed. In order to close the node correctly and ensure correct sealing, the following steps must be strictly followed.

- Step 1: Make sure that the screws have the o-rings to prevent water ingress.

Figure : Screws with o-ring

- Step 2: Ensure that the top surface of the gasket is clean and contains no foreign objects.
- Step 3: Place the inner casing inside the outer casing and make sure that the 2 position marks match.

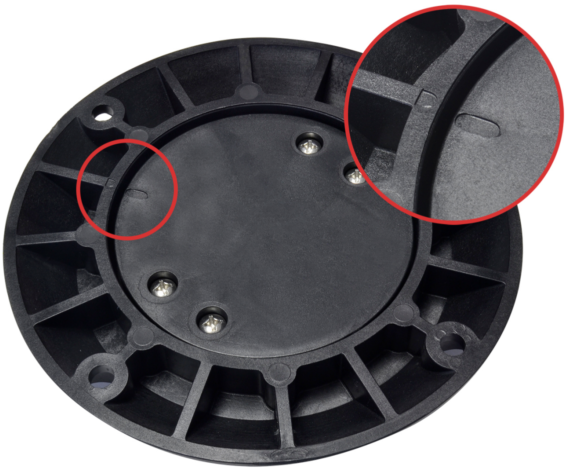


Figure : Enclosure position marks

- Step 4: Insert the screws and tighten them halfway.

Figure : Screws in their position

- Step 5: Finally, tighten the 4 screws firmly. Do not use the maximum pressure (do not go all the way with the screws), because the o-rings could be ejected from the screws, and then the waterproof feature would NOT be valid. Besides, do not screw too hard and keep on screwing, because the screws could carve the female sockets, expanding their inner diameter; this would cancel the waterproof quality too.

Libelium manufactures and provides all nodes configured after following all explained steps, so the node is "ready to install". By factory default, all nodes are configured with their unique LoRaWAN EUI and random private keys. On the other hand, if different LoRaWAN parameters are desired, "Smart Devices App " must be used to change the settings and repeat the previously explained steps.

3.5.3. "Magnet start-up" process

Once the node has been set to "ready to install" state and it has been closed and placed on the parking slot, the "magnet start-up" must be done. This process consists on resetting the device using the magnet for 3 consecutive times. Each magnet reset must be separated by at least one second period.

The best way to proceed with the magnet is to go over the enclosure from left to right in a one-motion movement. Then wait for at least one second (although you can wait more) and proceed again until you complete 3 magnet resets.

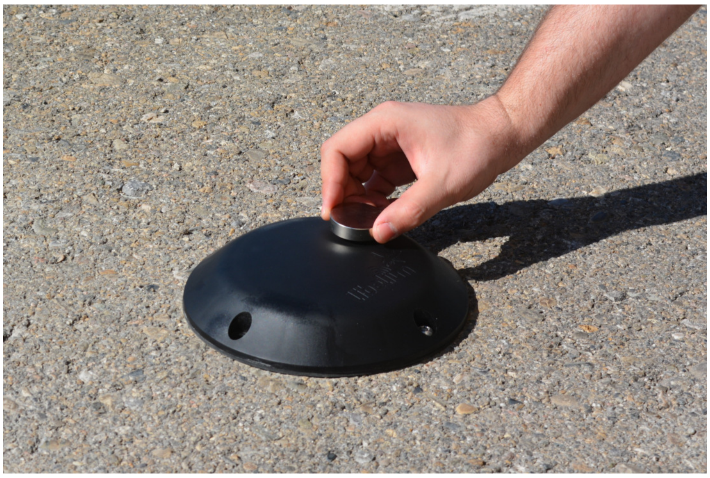


Figure : Magnet reset

In the next video-clip you can see how the "magnet start-up" is performed: Magnet start-up

After finishing the "magnet start-up", the node starts working normally for the rest of the time. No more three-time "magnet resets" are needed in order to reset the device properly. So if a 4th magnet reset or software reset is applied, the device will reset and continue working normally again.

Important:
The "magnet start-up" is only mandatory when the node is powered from a power-off state. In other words, when the device is set to a "ready to install" state.

3.6. How the node works

3.6.1. Frame types

The Smart Parking architecture manages different uplink and downlink frames.

The next table shows the Uplink frames:

Frame type	#num	Description
Start frame 1	4	First frame sent by the node when starting (with params settings)
Start frame 2	5	Second frame sent by the node when starting (with params settings)
Info frame	0	Used to inform a Parking Status change
Keep-alive frame	1	Used to inform the device keeps working since last reported status
Configuration uplink	2	Used to confirm a "Configuration downlink" was applied or not
RTC update request	7	Used to request for an RTC sync once every day

The next table shows the Downlink frames:

Frame type	#num	Description
Configuration downlink	3	Used to update the node parameters. After the customer sets up a new node configuration in the Remote Configuration Form a new "Configuration downlink" frame is enqueued into the LoRaWAN network server's downlink queue.
RTC sync frame	6	Used to sync the node's RTC to the server's timestamp. It is the mandatory response to "Start frame 1" and "RTC update request" uplink frames.

The uplink frames are 11-byte long to always comply with the LoRaWAN datarate worst case scenario. Their structure consists on 2 parts: header and payload. The "header" format is always the same for all uplink frame types. On the other hand, the "payload" format may be different for each frame type.

Header	Payload
2 bytes	9 bytes

Regarding the downlink frames, they have variable length and its format is private to the customer. The "RTC sync frame" is the mandatory response for both "Start Frame 1" and "RTC update request" frames. The "RTC sync frame" provides the server time to the nodes in order to keep the RTC updated. Also, the "Configuration downlink" is an asynchronous frame sent by the server when the Remote Configuration Form is managed by the customer.

You must keep in mind that when a downlink packet is requested there are usually some issues related to LoRaWAN network latency. This implies that the 1st request attempt usually fails. In that case, a 2nd attempt is sent in order to retrieve the lost downlink packet. For this reason, you might see that a couple of "Start Frame 1" or "RTC update request" frames are sent sequentially during the execution of the program.

3.6.2. Frame header

The "Header" included in each uplink frame contains 2 bytes:

Byte	Bit	Field
0	7	Parking lot status
0	6	Battery state
0	5	Configuration uplink acknowledgement
0	4	Sensor recalibration
0	3-0	Frame type
1	7-0	Sequence number

The meaning of each field is:

- Parking slot status:
  - 0: Free
  - 1: Occupied
- Battery status:
  - 0: OK
  - 1: Warning. The battery level measured is below the warning threshold (3340 mV)
- Configuration uplink frame acknowledgement status:
  - 0: ACK
  - 1: NACK
- Sensor recalibration:
  - 0: No calibration was done since the last uplink
  - 1: At least one calibration was done since the last uplink
- Frame type: Number related to frame type
  - 0: Info frame
  - 1: Keep-alive frame
  - 2: Configuration uplink frame
  - 3: Configuration downlink
  - 4: Start frame 1
  - 5: Start frame 2
  - 6: RTC sync frame
  - 7: RTC update request
- Sequence number: This is a 1-byte field so the sequence number goes from 0 to 255. When 255 is reached, the counter starts from zero again.

3.6.3. Frame payload

The "Payload" contents vary depending on each frame type.

The Start frame 1 frame contents are:

- Header ("Parking slot status" does not provide valid data)
- Firmware version:
  - From 1 to 8: Not released firmware versions
  - 9: v1.0.0
  - 10: v1.0.1
  - 11: v1.0.2 (last stable version)
- Battery level
- Radar settings (threshold and range)
- LoRaWAN settings (join mode and ADR)

The Start frame 2 frame contents are:

- Header ("Parking slot status" does not provide valid data)
- Sleep and Keep-alive time settings
- Night-mode settings (enabled/disabled, start hour, duration, sleep time, keep-alive time)

The Info frame contents are:

- Header
- Sensor error
- Temperature
- Timestamp (hour and minutes)
- Radar measurement (Distance, amplitude and number of reflections)

The Keep-alive frame contents are:

- Header
- Sensor error
- Temperature
- Timestamp (hour and minutes)
- Radar measurement (Distance, amplitude)
- Battery level

The RTC update request frame contents are the same as Keep-alive frame.

Important:
The Customer Server provides the needed source code to parse this data into a more comprehensive structure. The Libelium Cloud Bridge also provides the needed tools to transmit the parsed data to a 3rd party IoT cloud. For more information, please refer to the "Customer Server" section.

3.6.4. Node program flowchart

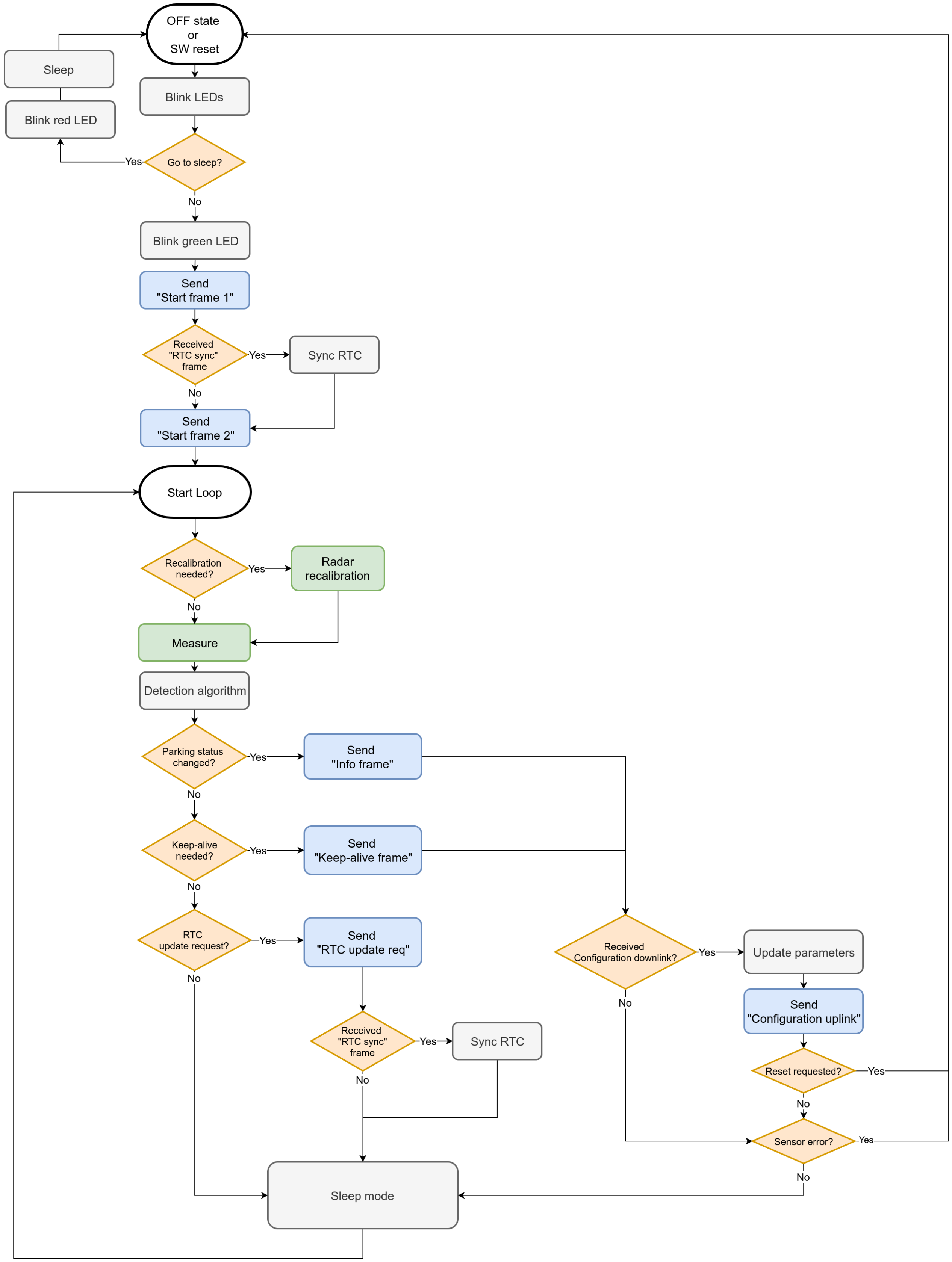


Figure : Smart Parking node program flowchart

3.7. Node parameters

3.7.1. Parameters description and ranges

The Smart Parking node has different parameters that change the timing and detection performance of the node. The next table shows the node parameters:

Parameter	Range	Description
Sleep time	1-10 min or 20-59 s	Minutes or seconds elapsed between each measurement cycle
Keep-alive time	0.5, 1, 2,..., 24 hour	Hours elapsed since last uplink message which triggers a new Keep-alive frame
Night-mode	0 or 1	Night-mode disabled/enabled
Night-mode start	0, 1,..., 23 hour	Night-mode starts when RTC reaches this parameter field
Night-mode duration	1, 2,..., 15 hour	Night-mode period is equal to this field
Night-mode sleep	1, 2,..., 10 min	Sleep time applied during night-mode
Night-mode keep-alive	1,..., 15 hour	Keep-alive time applied during night-mode
Radar range start	20 to 50 cm	Starting measurement distance (objects below this value are not detected)
Radar range length	50 to 100 cm	Range of measurement to be added to "range start" value
Radar threshold	5 to 100	Threshold used in detection algorithm, so higher threshold imply less sensitive detection
LoRaWAN join mode	0 (ABP) or 1 (OTAA)	Join mode used by the LoRaWAN radio module
LoRaWAN DevEUI	8-byte identifier	Defines the device EUI used by the LoRaWAN radio
LoRaWAN DevAddr	4-byte identifier	Defines the device address used by the LoRaWAN radio in ABP mode
LoRaWAN NwkSKey	16-byte key	Defines the LoRaWAN Network Session Key used by the LoRaWAN radio in ABP mode
LoRaWAN AppSKey	16-byte key	Defines the LoRaWAN Application Session Key used by the LoRaWAN radio in ABP mode
LoRaWAN AppKey	16-byte key	Defines the LoRaWAN Application Key used by the LoRaWAN radio in OTAA mode
LoRaWAN AppEUI	8-byte identifier	Defines the LoRaWAN Application EUI used by the LoRaWAN radio in OTAA mode
LoRaWAN port	1 to 223	Defines the port used for uplink sendings
LoRaWAN ADR	0 (off) or 1 (on)	Defines if Adaptive Data Rate is enabled or disabled
LoRaWAN RX1 Delay	0 to 65536	Defines the delay after first LoRaWAN rx window
LoRaWAN Subband	8-bit bitmap	Defines the sub-band used by the LoRaWAN radio (only applies to US and AU versions)

Important:
The LoRaWAN identifiers and keys must be registered in the LoRaWAN network server before starting the node in order to receive data. For OTAA mode: DevEUI, AppEUI and Appkey. For ABP mode: DevEUI, DevAddr, NwkSKey and AppSKey.

3.7.2. Understanding Info and Keep-alive frames

In the regular working mode (day-mode), "Sleep" and "Keep-alive" parameters are used. So the node normally sleeps for a specific "Sleep" time then wakes-up, measures and applies the algorithm detection in order to detect changes in the parking slot.

If a change is detected from 'free' to 'occupied' or viceversa, then an "Info" frame is sent. If no change occurred during the last "Keep-alive" time, then a Keep-alive frame is sent. Besides, if a sensor error is detected, a Keep-alive frame sending is forced in order to inform about this issue.

Example parameters used:

- Sleep: 7 minutes
- Keep-alive: 1 hour

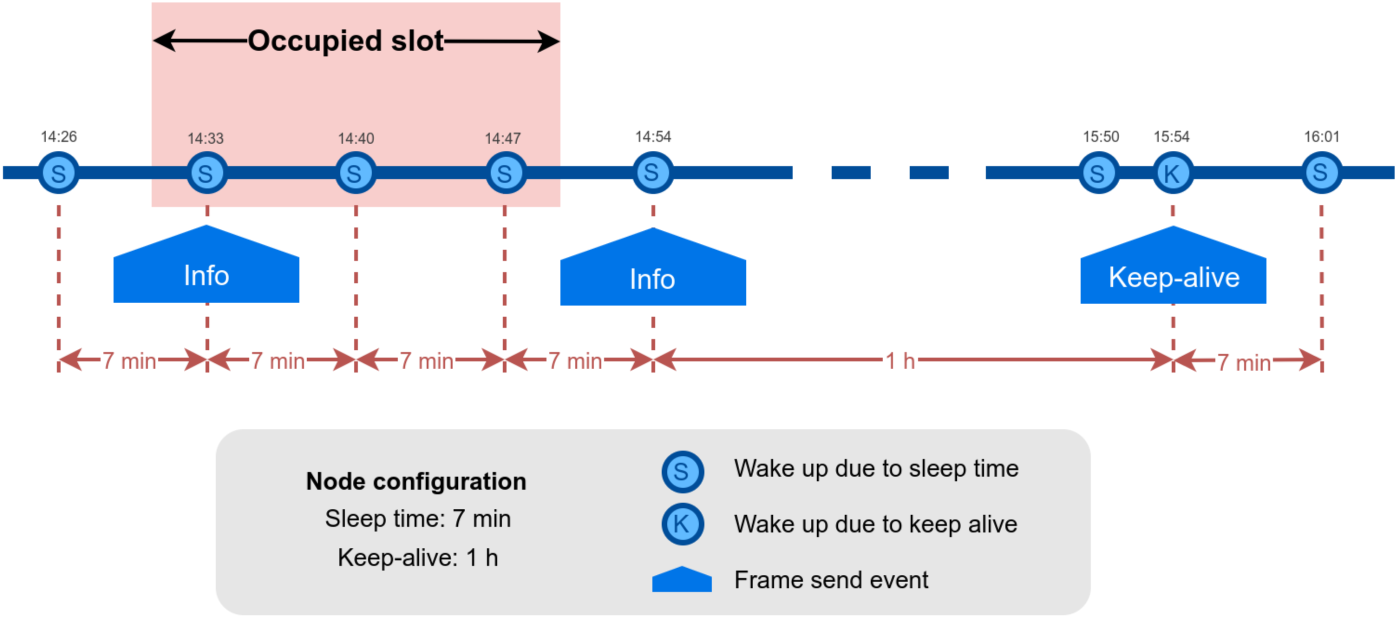


Figure : Example Info and Keep-alive frames

3.7.3. Understanding night-mode

As shown in the parameters table, there are some parameters that allow the user to configure the node to use 2 working modes depending on time settings: day-mode and night-mode.

The night-mode is a secondary and optional working mode that allows the user to configure a different time basis parameters in order to reduce the battery impact. So, it was developed to use it when the parking slot is expected to have fewer changes (i.e. at night). Therefore, different night-mode "Sleep" and "Keep-alive" settings are used.

It is not mandatory to use the night-mode during night. This mode is thought to be used when less vehicle movement is expected in the parking slots. Which could be during day time.

Example:

- Day-mode:
    - Sleep: 1 minute
    - Keep-alive: 1 hour
- Night-mode:

- Night-mode start hour: 21 hours (9 PM)
- Night-mode duration: 10 hours (Night-mode goes from 9 PM to 7 AM)
- Night-mode sleep time: 10 minutes
- Night-mode keep-alive: 10 hours

In the example, from 9 PM to 7 AM, the node will waste less battery because measurements are done every 10 minutes instead every minute. Something similar happens to the keep-alive events. If no parking slot change occurred during the night-mode, keep-alive events will happen every 10 hours instead every hour.

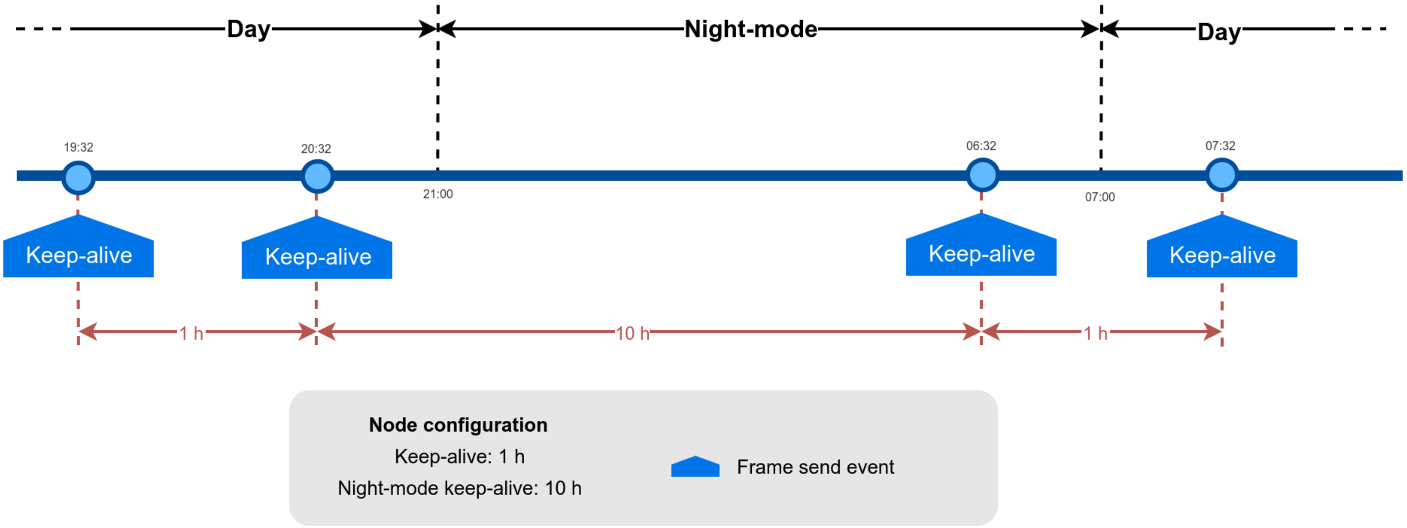


Figure : Example of day and night mode

Note: The conclusion is that the Night-mode is interesting for customers who certainly know the parking slot is expected to have fewer changes during large periods of time every day.

3.7.4. Understanding RTC synchronization

There are specific frame types that allow the node to synchronize the RTC to the server timestamp.

The "Start Frame 1" expects an answer from the server with the timestamp (hours and minutes). This frame is sent after starting the node or a software reset.

Besides, the node’s firmware provides a mechanism which an "RTC update request" frame is sent every 24 hours since the node was started or reset. This frame waits for a downlink frame which brings the current server timestamp (hour and minutes).

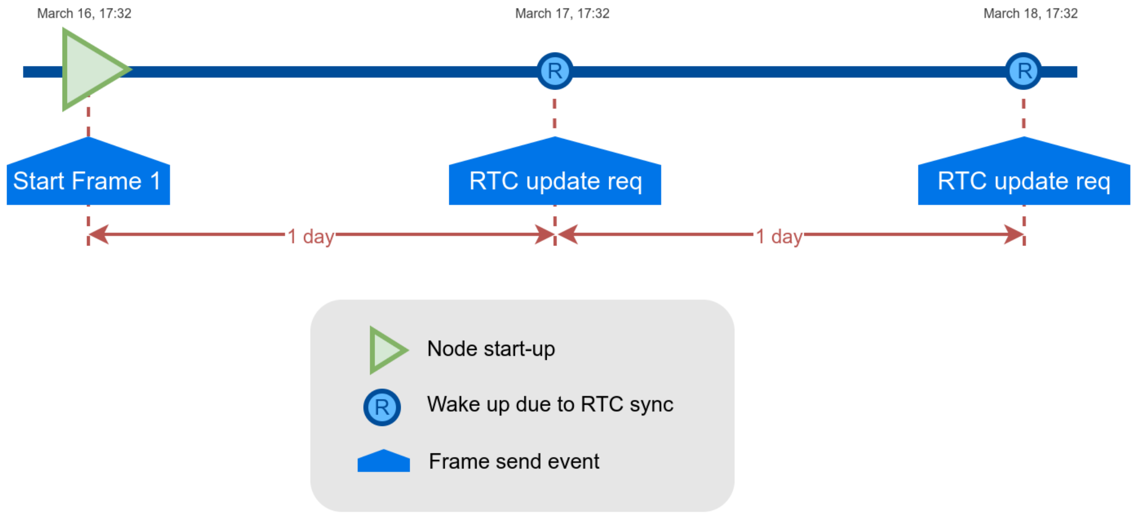


Figure : Example of RTC sync

Note: The RTC sync is important for Night-mode only where it mandatory to operate with a correct timestamp in order to enter and exit from night-mode to day-mode and viceversa.

3.7.5. Understanding uplink frames format (real example)

The next table shows all frames sent by a single node since it was started. The different columns display the parsed data from the received "uplink data".

Example:

- Day-mode:
  - Sleep: 1 minute
  - Keep-alive: 0.5 hour
- Night-mode:
  - Night-mode start hour: 22 hours (10 PM)
  - Night-mode duration: 8 hours (Night-mode goes from 10 PM to 6 AM)
  - Night-mode sleep time: 5 minutes
  - Night-mode keep-alive: 2 hours

It is possible to distinguish the starting frames at the beginning of the execution. Then the node informs with a new Keep-alive every 30 minutes. Any change of Parking slot status implies a new Info frame. And after 24 hours working, you can see the RTC request performed by the node.

Timestamp	Uplink data	F. Type	Parking lot	Battery	Recalibration	Sequence
04/15/19 15:59	040009c419143c01000000	4	NULL	0	0	0
04/15/19 15:59	0501010000011608050200	5	NULL	0	0	1
04/15/19 15:59	110200170f3b00000000b0	1	0	0	1	2
04/15/19 16:29	01030013101d00000000c8	1	0	0	0	3
04/15/19 16:59	01040013103b00000000c8	1	0	0	0	4
04/15/19 17:29	01050012111d00000000c8	1	0	0	0	5
04/15/19 17:59	01060010113b00000000c8	1	0	0	0	6

04/15/19 18:29	0107000f121d00000000c7	1	0	0	0	7
04/15/19 18:59	0108000e123b00000000d5	1	0	0	0	8
04/15/19 19:29	1109000d131d00000000d4	1	0	0	1	9
04/15/19 19:59	010a000c133b00000000d4	1	0	0	0	10
04/15/19 20:29	010b000b141d00000000d5	1	0	0	0	11
04/15/19 20:59	010c000b143b00000000d4	1	0	0	0	12
04/15/19 21:29	010d000a151d00000000d4	1	0	0	0	13
04/15/19 22:00	010e000a153b00000000d4	1	0	0	0	14
04/16/19 00:00	010f000a173b00000000d5	1	0	0	0	15
04/16/19 02:00	01100009013b00000000d5	1	0	0	0	16
04/16/19 04:00	0111000904000000000d4	1	0	0	0	17
04/16/19 06:00	0112000806000000000d5	1	0	0	0	18
04/16/19 06:04	8013000906040172410005	0	1	0	0	19
04/16/19 06:34	8114000906220175423 8ca	1	1	0	0	20
04/16/19 07:04	81150009070401704120d1	1	1	0	0	21
04/16/19 07:34	811600090722016f40b0d4	1	1	0	0	22
04/16/19 08:04	8117000a0804016d4268d4	1	1	0	0	23
04/16/19 08:34	8118000a0822016a3e50d4	1	1	0	0	24
04/16/19 09:04	8119000a090401743ec8d4	1	1	0	0	25
04/16/19 09:34	811a000b0922016d4160d4	1	1	0	0	26
04/16/19 10:04	811b000b0a04017b3e00d4	1	1	0	0	27
04/16/19 10:34	811c000b0a22016e4120d4	1	1	0	0	28
04/16/19 11:04	811d000c0b04017641d0d4	1	1	0	0	29
04/16/19 11:34	811e000c0b22016d41f8d4	1	1	0	0	30
04/16/19 12:04	811f000c0c04016d3e58d4	1	1	0	0	31
04/16/19 12:34	8120000d0c22016b3ea8d4	1	1	0	0	32
04/16/19 13:05	8121000e0d04016f3de8d4	1	1	0	0	33
04/16/19 13:35	8122000e0d23016f3e68c7	1	1	0	0	34
04/16/19 14:05	8123000e0e05016f3e80d4	1	1	0	0	35
04/16/19 14:35	8124000e0e2301af3de8c7	1	1	0	0	36
04/16/19 15:05	8125000e0f05016c3e50d5	1	1	0	0	37
04/16/19 15:35	8126000f0f2301703dd8c7	1	1	0	0	38
04/16/19 15:58	8727000f0f3b016b3dd0d4	7	1	0	0	39
04/16/19 15:59	8728000f0f3b016b3dd0d4	7	1	0	0	40
04/16/19 16:29	8129000f101c016e3c98c7	1	1	0	0	41
04/16/19 16:32	002a000f101f0000000000	0	0	0	0	42
04/16/19 17:02	012b001011010000000 0c7	1	0	0	0	43

04/16/19 17:32	012c0011111f00000000c8	1	0	0	0	44
04/16/19 18:02	012d0010120100000000c7	1	0	0	0	45

3.7.6. Factory default values

Libelium provides all Smart Parking nodes with factory default parameters.

Parameter	Default value
Sleep time	1 min
Keep-alive time	2 hour
Night-mode	0 (disabled)
Night-mode start	0 hour
Night-mode duration	6 hour
Night-mode sleep	5 min
Night-mode keep-alive	4 hour
Radar range start	20 cm (should not be changed)
Radar range length	60 cm (should not be changed)
Radar threshold	25 (should not be changed)
LoRaWAN join mode	1 (OTAA)
LoRaWAN DevEUI	unique factory default value
LoRaWAN DevAddr	unique factory default value
LoRaWAN NwkSKey	unique factory default value
LoRaWAN AppSKey	unique factory default value
LoRaWAN AppKey	unique factory default value
LoRaWAN AppEUI	unique factory default value
LoRaWAN port	3
LoRaWAN ADR	0 (off)
LoRaWAN RX1 Delay	1000 (should not be changed)
LoRaWAN Subband	8-bit bitmap

3.7.7. Configure new parameter values

The Smart Devices App and the Remote Configuration Form allow the user to configure new parameters to the node. The 1st one is a desktop Java application which implies opening the node enclosure and plug a micro-USB cable to the node. The 2nd one is a form allocated in the Libelium Cloud or Customer Server which permits to remotely change some of the node parameters.

Regarding the time and sensor parameters, the same values are set to all nodes manufactured by Libelium. The default values can be seen in the previous section. However, the customer can configure the time and sensor settings using both Smart Devices App and Remote Configuration Form.

Regarding the LoRaWAN parameters, all keys are randomly generated for each node and kept secret. The DevEUI set to the node is the LoRaWAN hardcoded EUI which is unique for each radio chipset. However, the client can configure/modify all LoRaWAN parameters using the Smart Devices App only (the Remote Configuration Form does not permit it).

Note: For further information about this matter please refer to the “Smart Devices App ” and “Remote Configuration Form” sections.

4. Libelium Cloud management

4.1. Introduction to the Libelium SCM

According to the Smart Parking network architecture, users can select between 2 ways of working with the Smart Parking nodes: one using the Customer Server, and the other using the Libelium Cloud. Regardless of the solution chosen, users will always need to operate with the Services Cloud Manager (SCM), which is part of the Libelium Cloud.

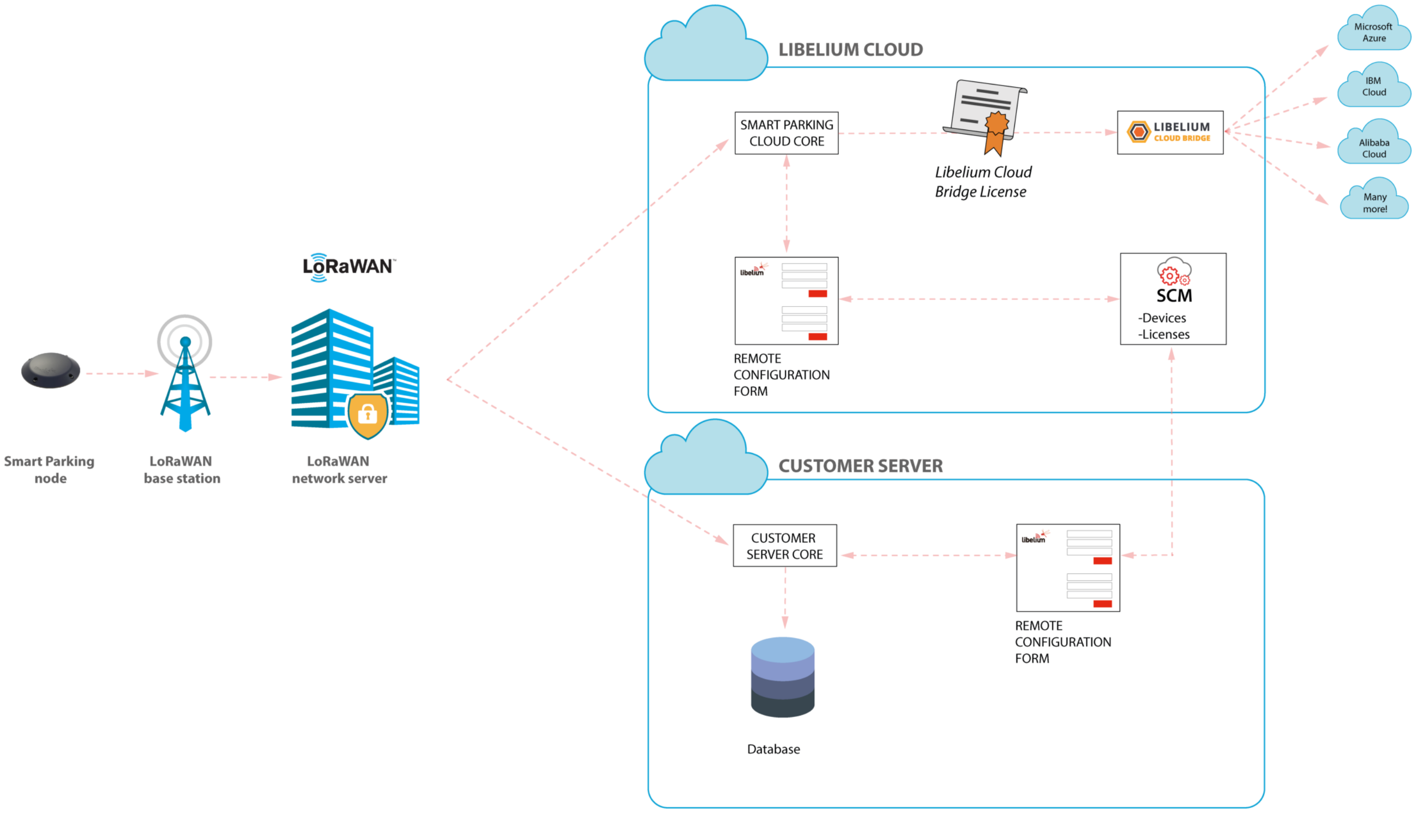


Figure : Smart Parking network architecture

4.2. SCM account

This is the 1st step users must follow. Anyone can create a new account in the SCM for free. This account will allow you to manage your Libelium devices and also all your licenses and services. You can access to the SCM at https://cloud.libelium.com.

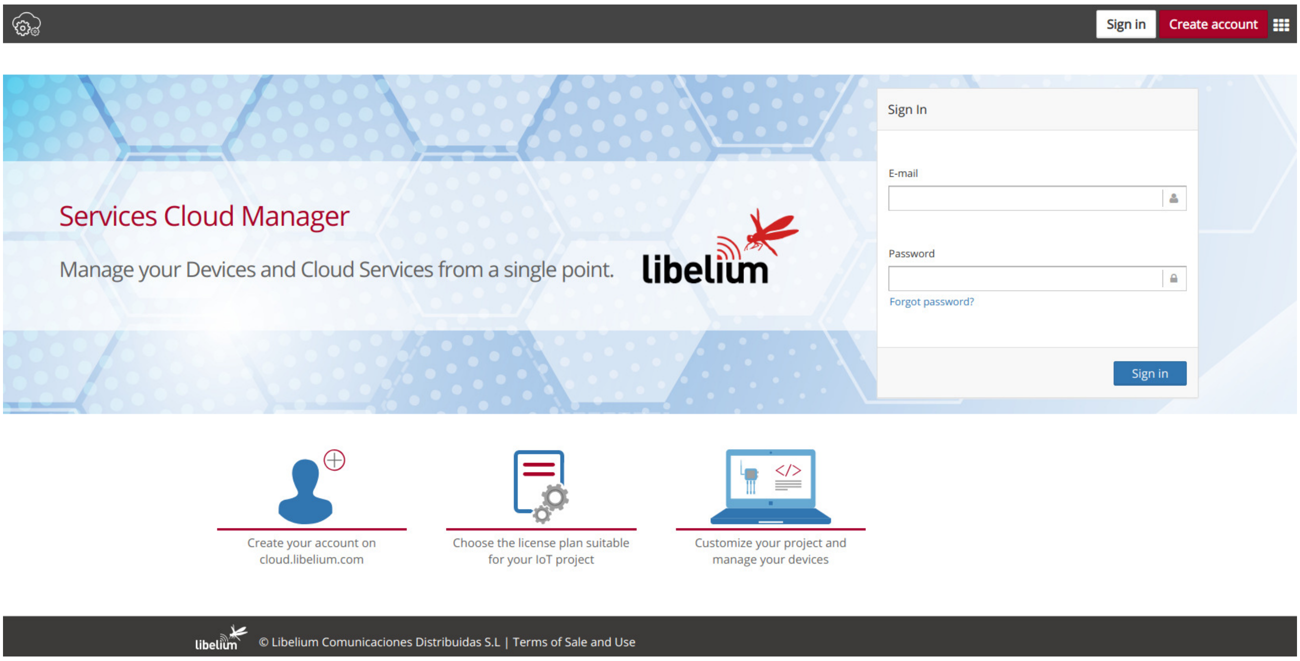

Sign in
Create account
Services Cloud Manager
Manage your Devices and Cloud Services from a single point.
libelium
Sign In
E-mail
Password
Forgot password?
Sign in
Create your account on cloud.libelium.com
Choose the license plan suitable for your IoT project
Customize your project and manage your devices
libelium
© Libelium Comunicaciones Distribuidas S.L | Terms of Sale and Use

Figure : SCM Home page

4.2.1. Creating an account

To register and set up a password, go to https://cloud.libelium.com/register and click on “Create account”.

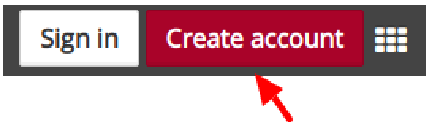
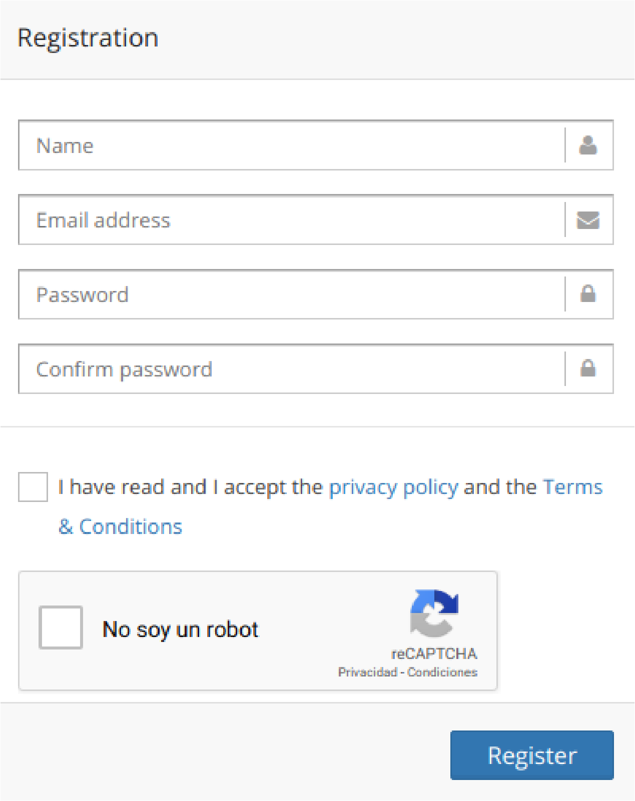


Figure : "Create account" button

Complete the information in the form (all the fields are mandatory):

- Name: Your name
- E-mail address: A valid e-mail address, it will be used for verification purposes
- Password and Confirm password: Set your password

Accepting Terms & Conditions is also mandatory to create a new account in the SCM.



□ I have read and I accept the privacy policy and the Terms & Conditions

Figure : Create account form

4.2.2. Signing in

To sign in the SCM, go to the Libelium Cloud and click on the “Sign In” button.

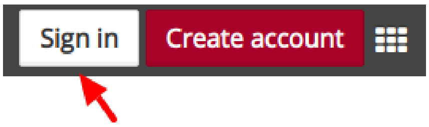


Figure : "Sign-in" button

Complete the information in the form. The e-mail and password used when creating the account will be required:

- E-mail address: E-mail address used for registration
- Password: Password used for registration

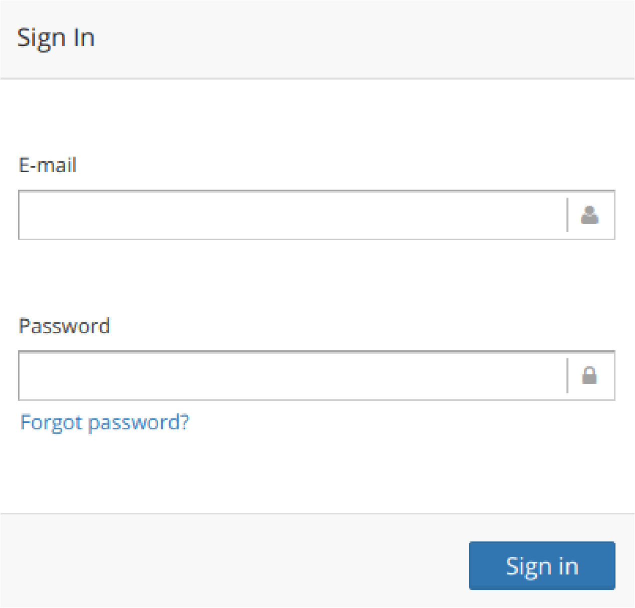
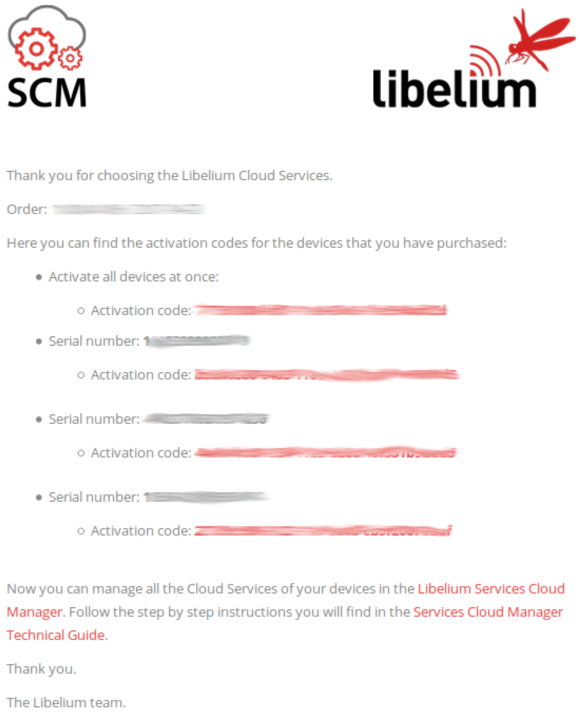


Figure : Sign-in form

The Terms & Conditions accepted when creating the account apply for any time you sign in the SCM. If you do not have an account yet, follow the steps described on the section "Create an account" to obtain valid credentials. If you already created an account and need a password reminder, you can click on "Forgot password?".

4.3. Smart Parking nodes registration

After placing an order including Smart Parking nodes you should receive an e-mail with the activation codes for all your devices. In this e-mail you will directly find links to register the devices in the Libelium Cloud.



Thank you for choosing the Libelium Cloud Services.

Order: [redacted]

Here you can find the activation codes for the devices that you have purchased:

- Activate all devices at once:
  - Activation code: [redacted]
- Serial number: [redacted]
  - Activation code: [redacted]
- Serial number: [redacted]
  - Activation code: [redacted]
- Serial number: [redacted]
  - Activation code: [redacted]

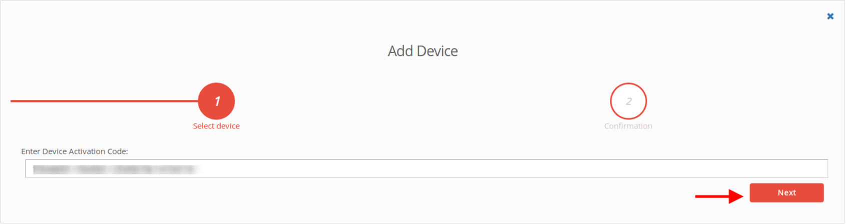
Now you can manage all the Cloud Services of your devices in the Libelium Services Cloud Manager. Follow the step by step instructions you will find in the Services Cloud Manager Technical Guide.

Thank you.

The Libelium team.

Figure : Activation codes e-mail

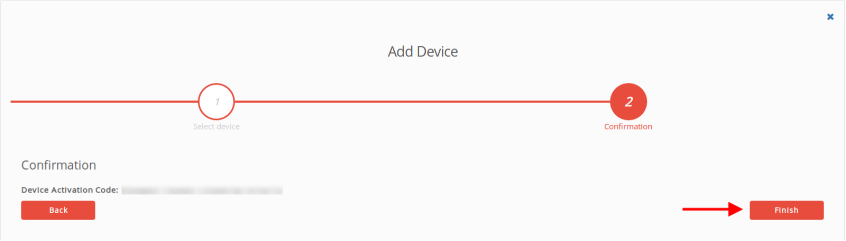
By clicking on these links you will be able to complete the registration process by following the wizard:



Add Device

Enter Device Activation Code:

Figure : Step1. Enter device activation code



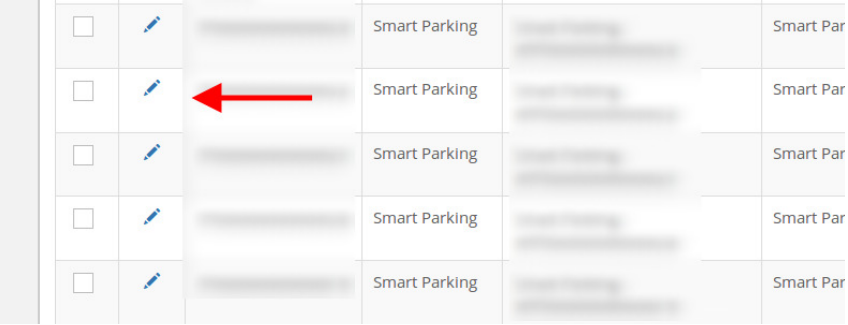
Add Device

Confirmation

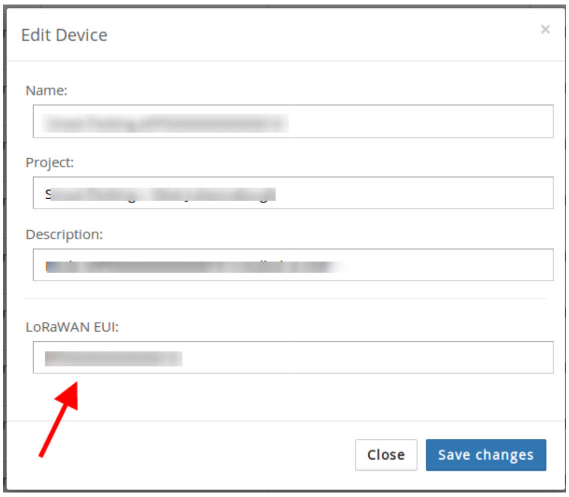
Device Activation Code:

4.4. Editing Smart Parking nodes

You can change the "name", "project", "description" and (if the device is a Smart Parking node) the "LoRaWAN EUI" of the device by clicking the pencil icon located next to the each device. Finally you must save the changes.



□	✎		Smart Parking		Smart Par
□	✎		Smart Parking		Smart Par
□	✎		Smart Parking		Smart Par
□	✎		Smart Parking		Smart Par
□	✎		Smart Parking		Smart Par



Edit Device

Name:

Project:

Description:

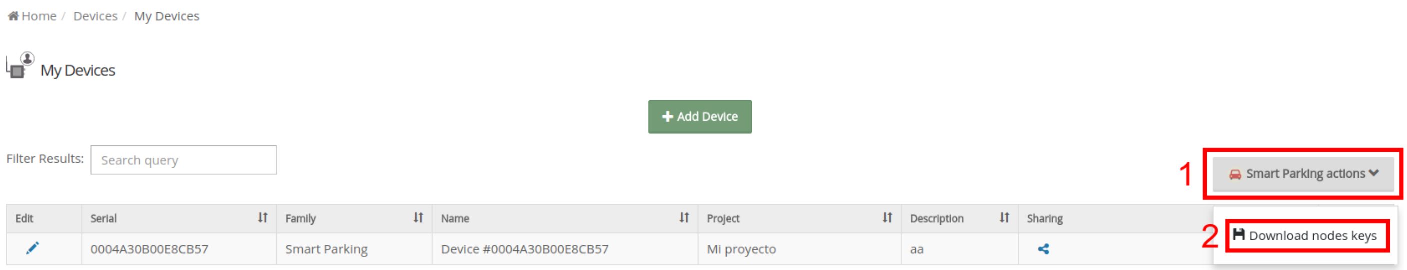
LoRaWAN EUI:

Figure : Edit a device

4.5. Export CSV file with nodes credentials

The Libelium Cloud allows the user to export all nodes credentials via HTTPS secure connection. The exported file is a CSV file with all credentials needed for OTAA join mode. The exported LoRaWAN credentials are the same as they were configured by factory default. This means that if the user changed the LoRaWAN credentials via "Smart Devices App", then the exported credentials will not be same.

In order to export the file, you must go to https://cloud.libelium.com, log in and then access "My Devices". On the website you will identify the "Smart Parking actions" button which permits to download the nodes credentials by clicking on the "Download nodes keys" button.



Edit	Serial	Family	Name	Project	Description	Sharing
✎	0004A30B00E8CB57	Smart Parking	Device #0004A30B00E8CB57	Mi proyecto	aa	⋖

Figure : Export CSV file with all nodes credentials

The CSV file for 'N' devices exported from Libelium Cloud follows the next format:

```
<DevEUI_1>,<AppEUI_1>,<AppKey_1>
<DevEUI_2>,<AppEUI_2>,<AppKey_2>
<DevEUI_3>,<AppEUI_3>,<AppKey_3>
...
<DevEUI_N>,<AppEUI_N>,<AppKey_N>
```

5. Smart Devices App

Libelium Smart Devices App is an software tool developed by Libelium that allows users install new firmware versions and program the configuration of the new Libelium devices in a few clicks. At the moment it is only available for Smart Parking and MySignals products, but the list will be incremented shortly.

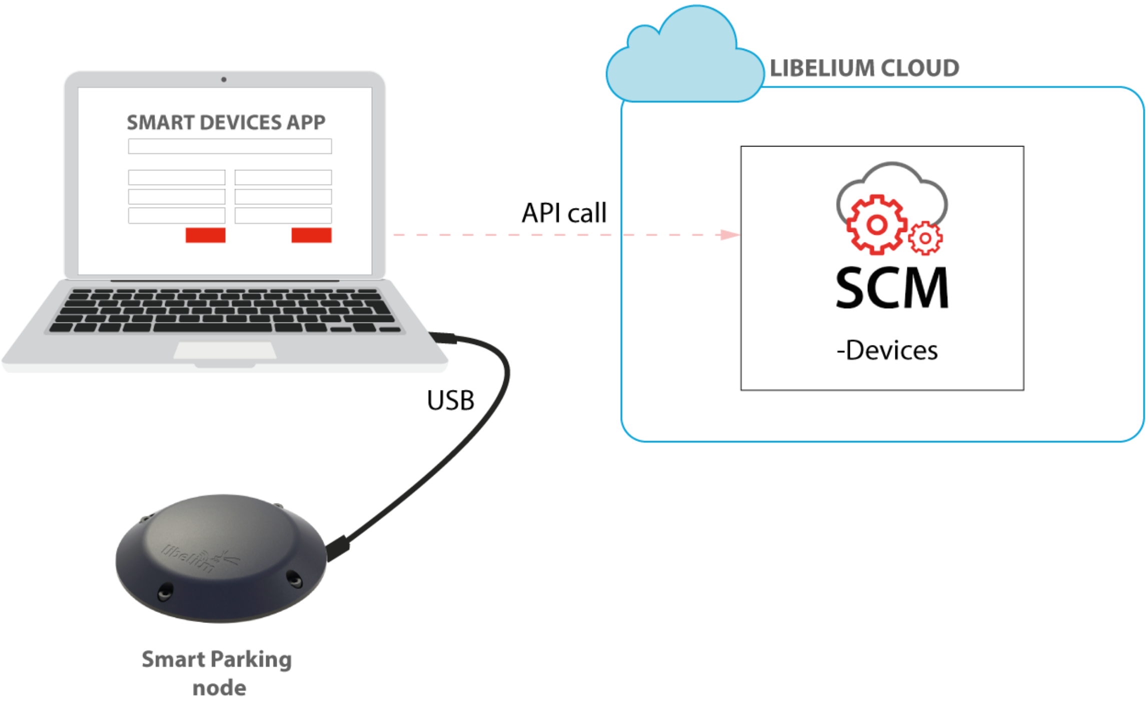


Figure : Smart Devices App is connected to the node via USB

5.1. How to install the Smart Devices App

First of all and before installing anything, users have to take into account the platform where the application is going to be installed. To install the Smart Devices App , it is compulsory to have installed the JDK 1.8. If it is not installed in the computer, you can download it from the website below and follow the steps described:

https://docs.oracle.com/javase/8/docs/technotes/guides/install/install_overview.html .

Once installed JDK, users can download the application in the link below, selecting the appropriate operative system and architecture:

http://www.libelium.com/development/smart-parking/software-and-applications

Then customers only have to extract the content of the SmartDeviceApp zip file downloaded in a place with the right permissions, and finally execute the file called "SmartDeviceApp" that will initialize the application. Please, note that the extension of this file will depend on the operating system the user is using at the moment (.sh for Linux and OSX, and .bat for Windows).

This new Smart Devices App uses a new programmer under the hood called STM32CubeProgrammer, so it is a good practice to before using the Smart Devices App :

https://www.st.com/en/development-tools/stm32cubeprog.html

Windows systems

It is important to install STM32CubeProgrammer as the software contains the drivers needed to set the USB ports in DFU mode, if you want to know more about how to install this software and know about how to set the USB in DFU mode, please see the official guide from STM32CubeProgrammer:

Link to ST documentation.

Make sure that your USB can switch to DFU mode, you can check it with the STM32CubeProgrammer.

MacOS systems

It is not necessary to install additional software, but you can install STM32CubeProgrammer:

https://www.st.com/en/development-tools/stm32cubeprog.html

GNU/Linux systems

It is necessary to change permissions to USB ports. If you have problems with your USB device on your GNU/Linux (or the Smart Devices App does not recognize the SmartParking v2 device) you can compile the *stlink driver discover*:

```
git clone https://github.com/texane/stlink.git stlink
cd stlink
make
# install binaries:
sudo cp build/Release/st* /usr/local/bin/
# install udev rules
sudo cp etc/udev/rules.d/49-stlinkv* /etc/udev/rules.d/
# and restart udev
sudo udevadm control --reload
```

If you have problems with stlink driver installation please refer this site: https://github.com/texane/stlink.

Review the UDEV rules and they should be as follows:

```
nano /etc/udev/rules.d/49-stlinkv2.rules
SUBSYSTEMS=="usb", ATTRS{idVendor}=="0483", ATTRS{idProduct}=="3748", \
    MODE:="0666", \
    SYMLINK+="stlinkv2_%n"
```

Add your user to **dialout**:

```
sudo addgroup <username> dialout
```

Then you will be able to use the USB ports on Linux systems. It is also recommended to install *STM32CubeProgrammer* for the current OS the user is running:

https://www.st.com/en/development-tools/stm32cubeprog.html

5.1.1. Start Smart Devices App on Windows

Go to the Smart Devices App folder and click on the file **SmartDevicesApp.bat**.

5.1.2. Start Smart Devices App on GNU/Linux

Open a new terminal window and navigate to the folder containing the Smart Devices App application, then type:

```
        sudo ./SmartDevicesApp.sh
```

5.1.3. Start Smart Devices App on MacOSX

Open a new terminal window and navigate to the folder containing the Smart Devices App application, then type:

```
        sudo ./SmartDevicesApp.sh
```

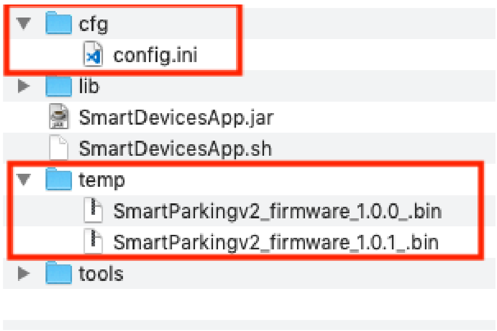
5.2. Upgrading the Smart Devices App

Occasionally, some improvements and new services will be released; if this is the case, the Smart Devices App will show a notification encouraging the user to go to the “Installation” section and download the new version.

Important:
It is mandatory to have an active internet connection in order to use the Smart Devices App .

To upgrade the Smart Devices App , you must follow the installation steps. This will overwrite the files from previous versions. As a 1st step, we recommend to backup the following files:

- cfg/config.ini: This file stores the Smart Devices App configuration parameters. To restore those settings, every single parameter value (username, password, etc) should be copied into the new version of the “config.ini” file. The file should not be overwritten, new parameters must be added and merging files is mandatory.
- temp: This folder contains the firmware files previously used. To restore these files, copy the content of the saved “temp” directory into the new version “temp” directory.



```
cfg
    config.ini
lib
SmartDevicesApp.jar
SmartDevicesApp.sh
temp
    SmartParkingv2_firmware_1.0.0_.bin
    SmartParkingv2_firmware_1.0.1_.bin
tools
```

Figure : Smart Devices App files

5.3. Smart Parking v2

There is a new tab on the application for the new Parking version section called “Smart Parking v2 ”. Inside the tab there are a lot of new characteristics and options to configure the Smart Parking v2 device.

The main sections are: Programmer, Firmware Update and Configuration.

5.3.1. How to plug the Smart Parking node

Before using the Smart Devices App, you must keep in mind how to plug and initialize it properly.

As explained in the "Smart Parking node" section, there are 2 switches to manage the node working mode. In order to use the node with the Smart Devices App, it is mandatory to set the node in "Boot" mode and then power it on. If you changed from "App" to "Boot" mode, then you can use the reset button to apply the change and restart the node in the bootloader section.

Note: Go to the "User switches" section for further information

> **Important**:
> *Never leave the device set to "On" and "Boot" for more time than needed: the bootloader does not provide any sleep mode and it will waste the battery of the device. So when you finish reconfiguring the device, please set the node to "Off" state.*

5.3.2. Configuration

In the last tab, called **Configuration**, 2 parameters are available: "Username" and "Password". These are the same credentials needed for the Libelium Cloud (SCM, https://cloud.libelium.com/login).

Filling these fields is mandatory before start using the Smart Devices App with the Smart Parking node. Please remember to click the "Save" button at the bottom of the application to store the correct credentials.

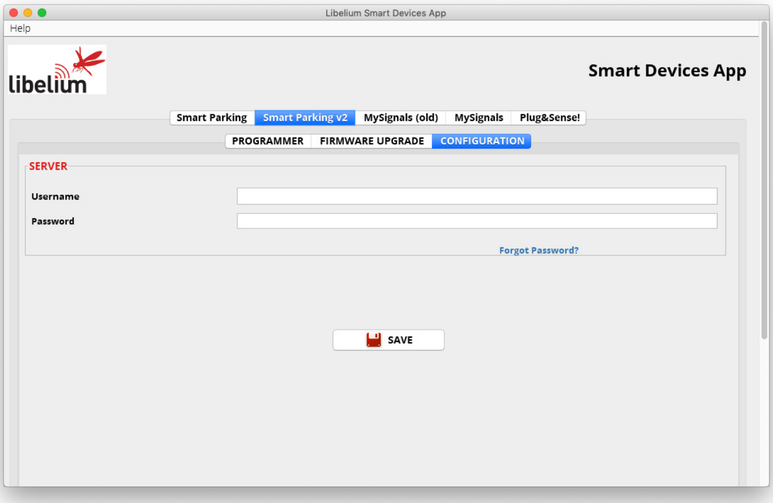


Figure : Configuration screen on Smart Parking v2

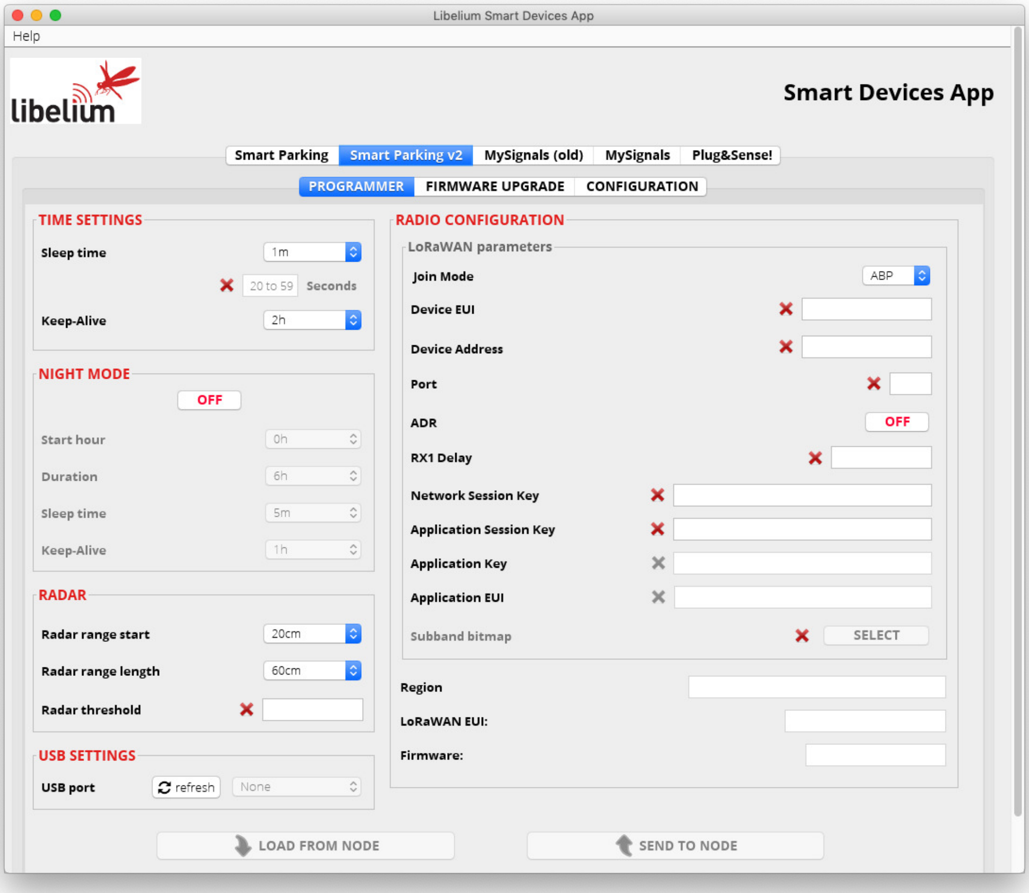
There are some differences between the 1st version and the 2nd one. In Smart Parking v2 you must type your credentials to authenticate your nodes against the Libelium Cloud.

Note: You should have an account on Libelium Cloud to get your username and password.

If you forgot your password you can always recover it by clicking on the "Forgot password?" link on the application:

https://cloud.libelium.com/password/reset

5.3.3. Programmer (change node parameters)



TIME SETTINGS

Sleep time: 1m

✗ 20 to 59 Seconds

Keep-Alive: 2h

NIGHT MODE

OFF

Start hour: 0h

Duration: 6h

Sleep time: 5m

Keep-Alive: 1h

RADAR

Radar range start: 20cm

Radar range length: 60cm

Radar threshold ✗

USB SETTINGS

USB port: refresh None

RADIO CONFIGURATION

LoRaWAN parameters

Join Mode: ABP

Device EUI ✗

Device Address ✗

Port ✗

ADR: OFF

RX1 Delay ✗

Network Session Key ✗

Application Session Key ✗

Application Key ✗

Application EUI ✗

Subband bitmap ✗ SELECT

Region:

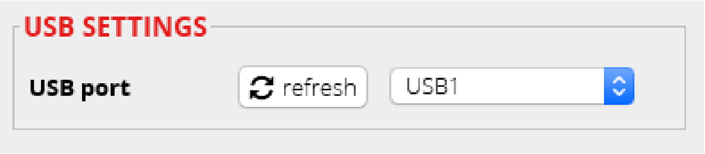
LoRaWAN EUI:

Firmware:

Figure : Smart Parking v2 Programmer section

Users can read and write all node parameters in this section. The process is quite simple: just connect the device to the computer where the Smart Devices App is installed using the USB cable provided and switch on the node in "Boot" mode.

Next, refresh the "USB settings" section (bottom-left corner), clicking the "Refresh" button. Then select the port where the device has been connected. After that it is a good practice to hit the "Load from node" button to get the current configuration from the connected Smart Parking v2 device.



USB SETTINGS

USB port: refresh USB1

Figure : Smart Parking v2 USB settings

The "Load from node" button will read all parameters from the node and will display the information on the app. On the other hand, the "Send to node" button will overwrite the configuration on the node. All available fields have to be filled with the proper format. If any parameter does not have an acceptable format, a red cross like this ✗ is displayed close to it, and you will not be able to write the information on the node. If the information introduced is valid, a green tick ✓ is shown. When loading/sending information to the node, the application will warn the user through a pop-up window about the status of the operation.

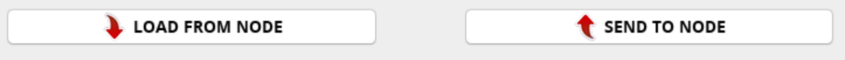


Figure : "Load configuration from node" & "Send configuration to node" buttons

Smart Parking v2 shares most of the configuration fields with Smart Parking v1 . However, it is important to know the purpose for each field on the Smart Parking v2 configuration. To know more about each field and how they work, please refer to the chapter "Smart Parking node".

5.3.4. Firmware upgrade

Inside this tab, users can select the firmware version to install in their devices.

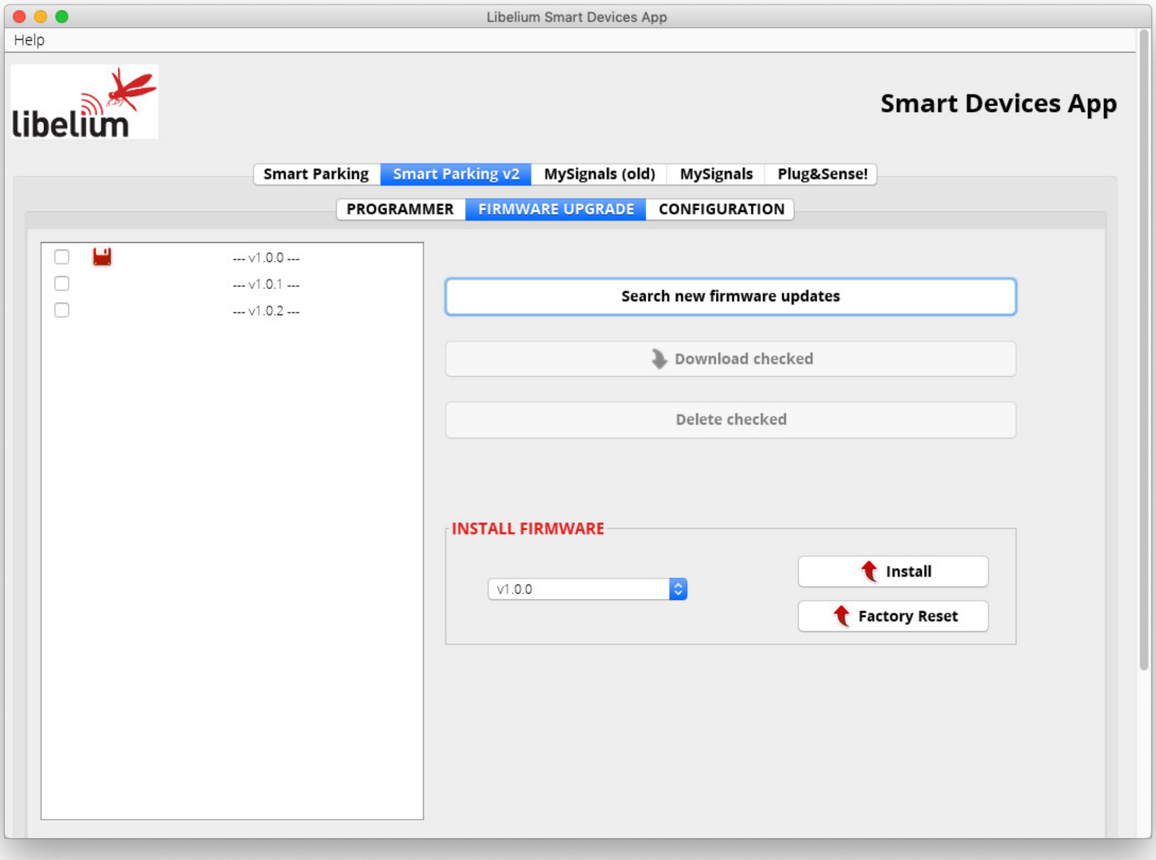


Figure : Firmware update and factory reset section for Smart Parking v2

The list with all available firmwares is loaded when the program starts, but users can manually update it by clicking on the "Search new firmware updates" button. Before installing the firmware, it is necessary to download it. This process is very simple, just mark the check of the version you want to install from the list ☑ and click on **"Download checked"**.

To summarize, the steps are:

1. Click the "Search new firmware updates" button to load new firmware updates
2. Check one or more firmwares to download from the firmware list on the left side
3. Click on "Download checked" to download the selected firmwares

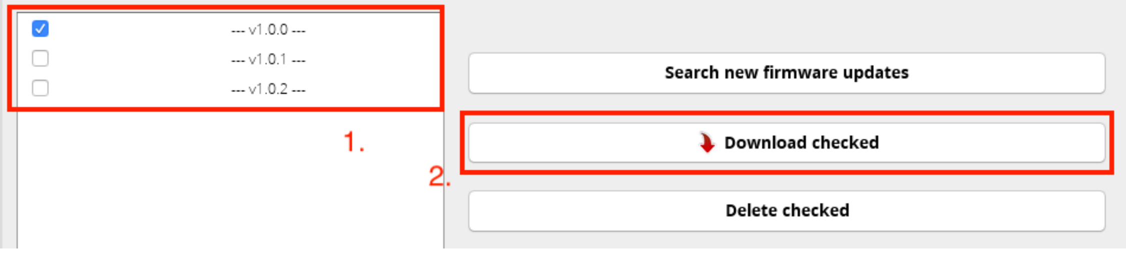


Figure : Download new firmware versions

Now the firmware files are downloaded. The drop-down menu will show the available firmware files to be installed on the node.

When the file is downloaded, a disk icon 💾 is displayed close to it, indicating it is downloaded. On the "Install Firmware" section you can select the firmware to install and then hit on "Install" to execute the installation. The app will display a pop-up message for the status of the firmware installation.

Note: Remember that the USB port **must be selected** in the programmer tab

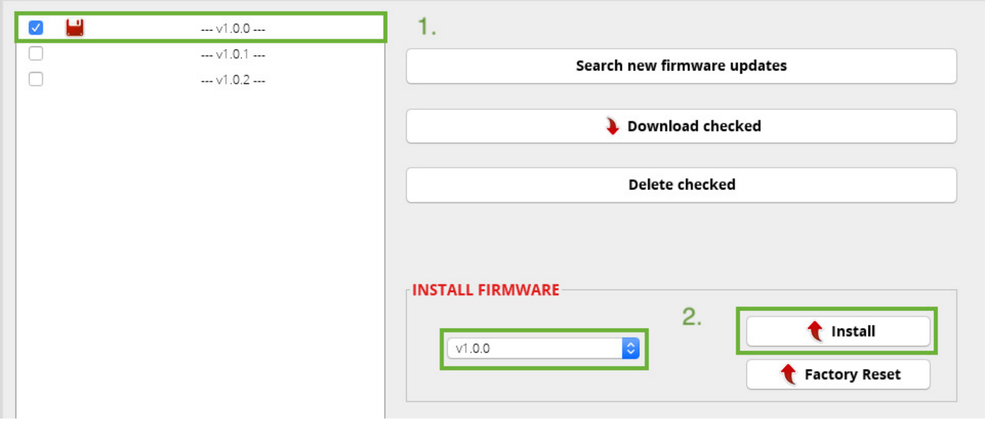


Figure : Firmware installation

You can also delete the downloaded firmware files by selecting the check box from the list and then clicking on the "Delete checked" button.

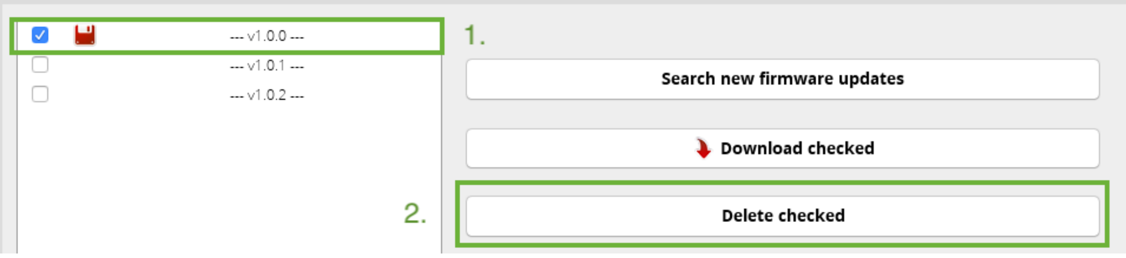


Figure : Firmware deletion process for Smart Parking v2

5.3.5. Factory Reset

The Factory Reset is the last option to recover a non-working SmartParking v2 device. It performs a full reset followed by a firmware and configuration reset. To get the recovery configuration file, send an e-mail to tech@libelium.com pointing your Smart Parking v2 order number and the Technical Service Team will send the configuration file needed.

For this process it is mandatory to download a firmware, as explained on the previous section "Firmware upgrade". Then select a firmware from the drop-down menu and press the "Factory Reset" button.

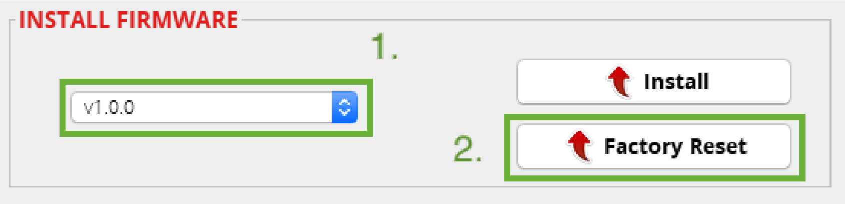
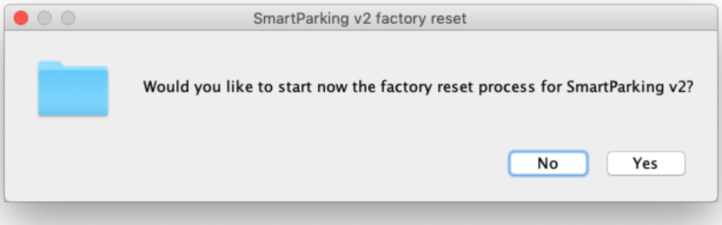


Figure : Factory reset process for Smart Parking v2 - Step 1

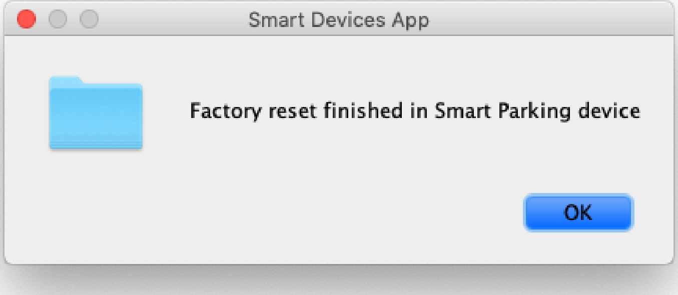
Pressing the "Factory Reset" button will display a file explorer window to select the configuration file provided by Libelium. Once you selected this file, a message will show up asking if you want to start the process. If the "Yes" option is selected, the Factory Reset process starts.


SmartParking v2 factory reset

Would you like to start now the factory reset process for SmartParking v2?

Figure : Factory reset process for Smart Parking v2 - Step 2

When the Factory Reset finishes, another message informs about the final result:


Smart Devices App

Factory reset finished in Smart Parking device

Figure : Factory reset finished for Smart Parking v2

After this process, you can configure your Smart Parking v2 node again on the "Programmer" tab.

6. LoRaWAN Network Server setup

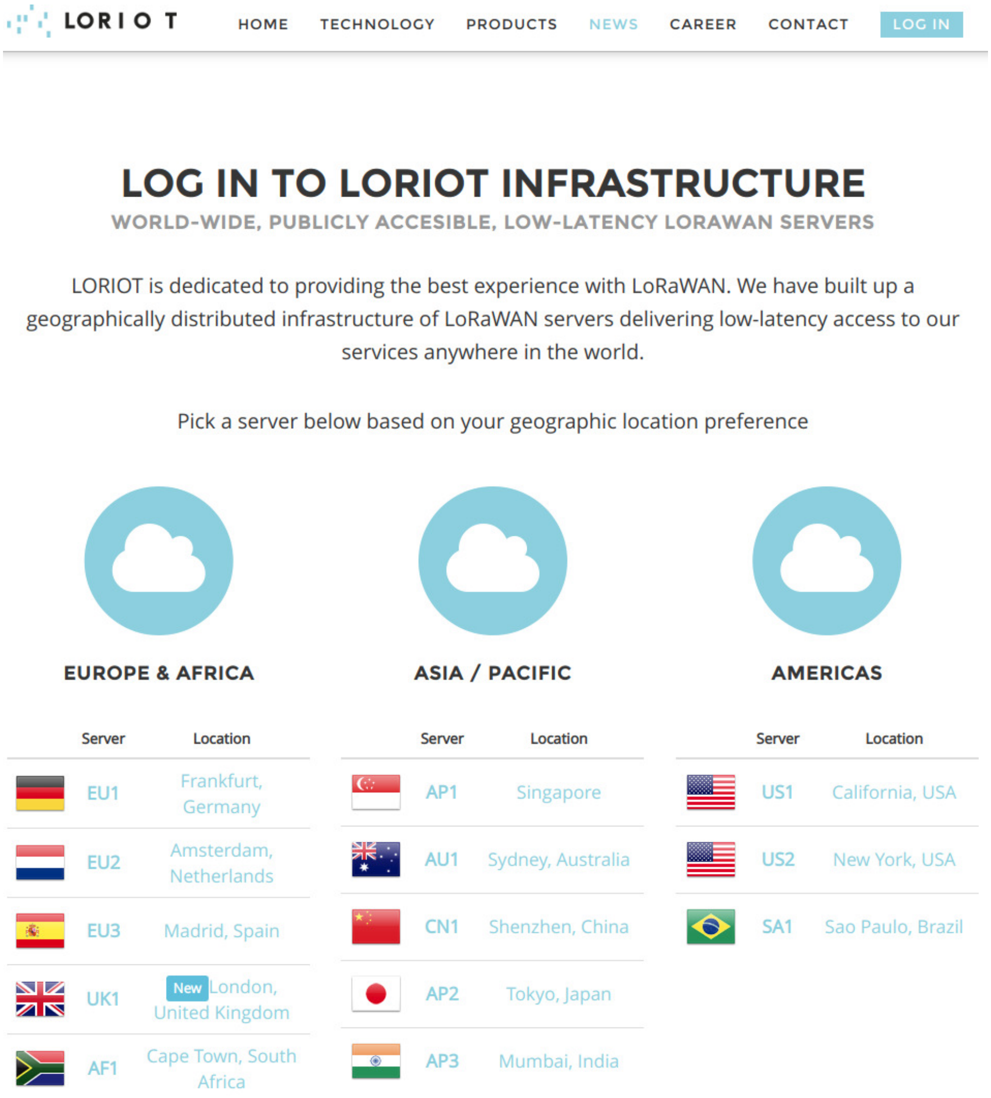
Libelium currently supports two services in the Customer server application: **Loriot** and **Actility**. In this section, it is explained how to set them up with a simple configuration.

6.1. Loriot

This section explains how to route the information via Loriot network server to the "Customer server" using the HTTPS protocol.

6.1.1. Log in

Log in in the Loriot platform, selecting your server location.



LOG IN TO LORIOT INFRASTRUCTURE

WORLD-WIDE, PUBLICLY ACCESIBLE, LOW-LATENCY LORAWAN SERVERS

LORIOT is dedicated to providing the best experience with LoRaWAN. We have built up a geographically distributed infrastructure of LoRaWAN servers delivering low-latency access to our services anywhere in the world.

Pick a server below based on your geographic location preference

EUROPE & AFRICA

Server	Location
EU1	Frankfurt, Germany
EU2	Amsterdam, Netherlands
EU3	Madrid, Spain
UK1	New London, United Kingdom
AF1	Cape Town, South Africa

ASIA / PACIFIC

Server	Location
AP1	Singapore
AU1	Sydney, Australia
CN1	Shenzhen, China
AP2	Tokyo, Japan
AP3	Mumbai, India

AMERICAS

Server	Location
US1	California, USA
US2	New York, USA
SA1	Sao Paulo, Brazil

Figure : Loriot login web page

6.1.2. Create a new Loriot application

Once you selected the server location, a new prompt will be displayed asking your credentials. After logging in, users have to create a new application in the Loriot dashboard. To do this it is necessary to click on **"Applications"** in the left side bar and then in **"New Application"** option located in the same bar:

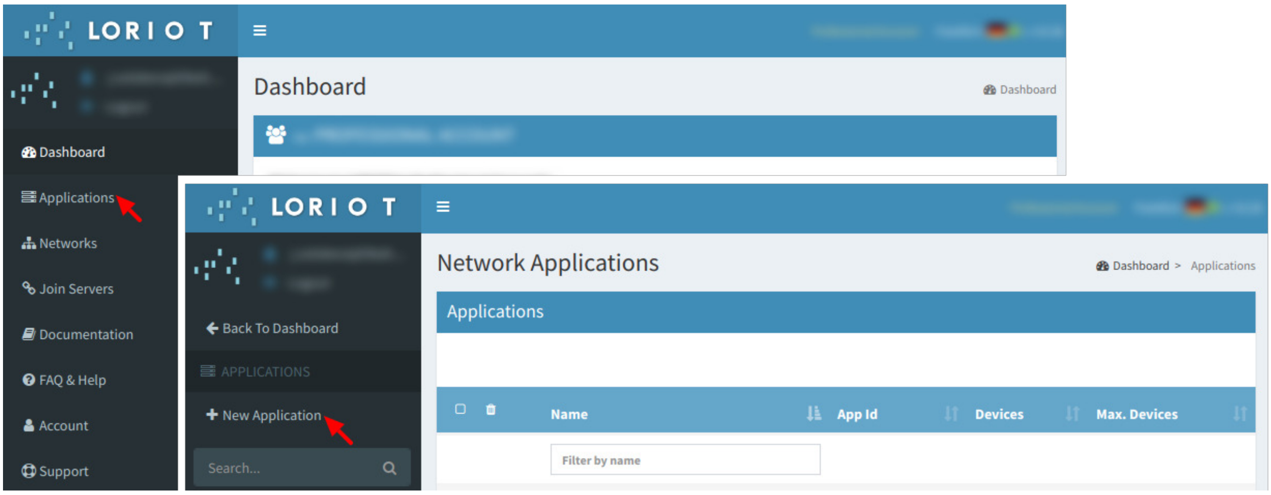
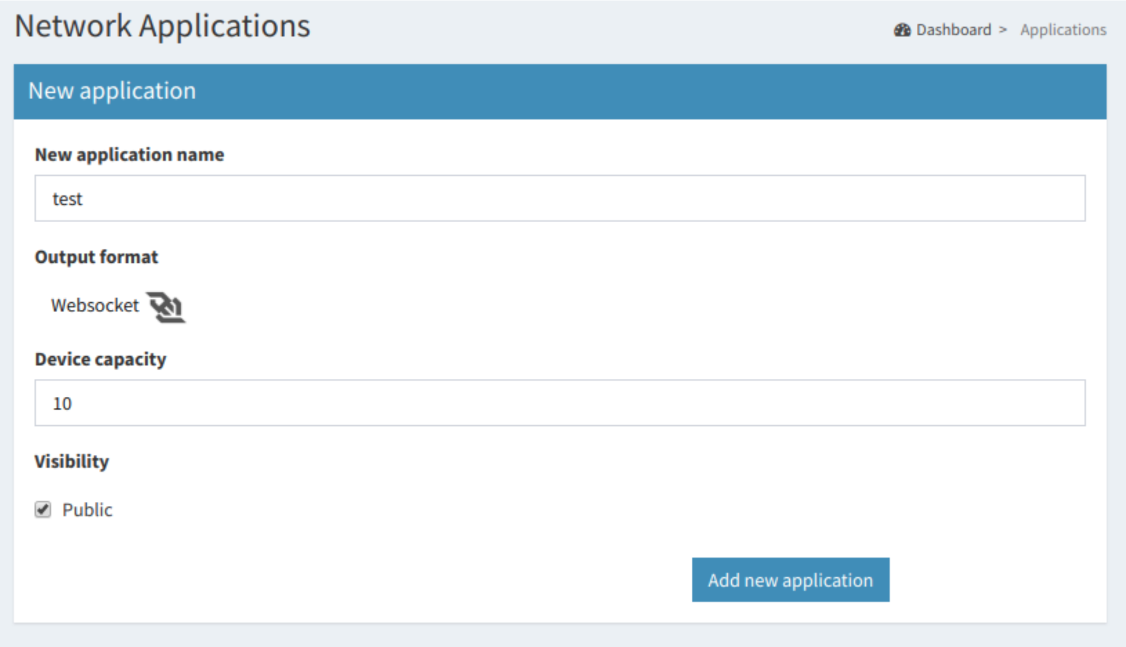


Figure : Creating an application

Fill the form in order to create the desired application in Loriot:

- **New application name**: Name to identify the application.
- **Output format**: "Websocket" is selected by default and it is not possible to change it in this form. How to change it will be explained in the following step.
- **Device capacity**: Maximum number of devices to be registered in this application. The maximum value will depend on the account type you have.
- **Visibility**: This checkbox permits to set the application as "public" or "private".



New application name
test

Output format
Websocket

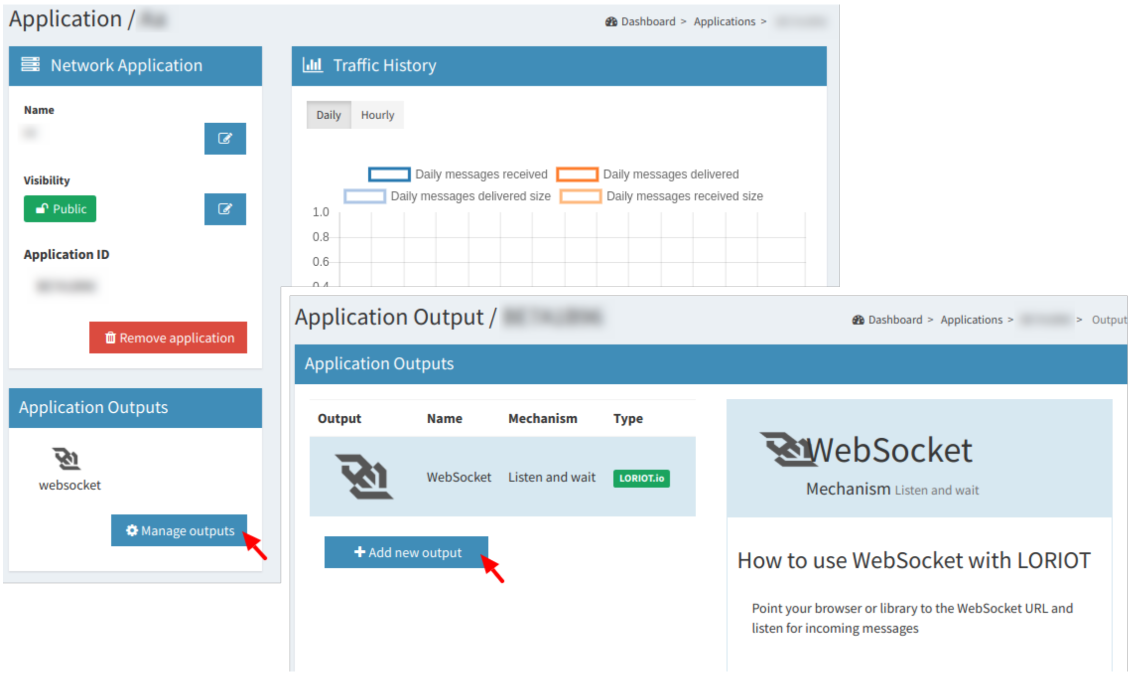
Device capacity
10

Visibility
☑ Public

Figure : New application form

6.1.3. Manage Loriot output data

Once the application is created, the dashboard will be shown. Go to **"Application Outputs"** → click on **"Manage outputs"** → click on "Add new output":



Name: [redacted]

Visibility: Public

Application ID: [redacted]

Output	Name	Mechanism	Type
	WebSocket	Listen and wait	LORIOT.io

WebSocket

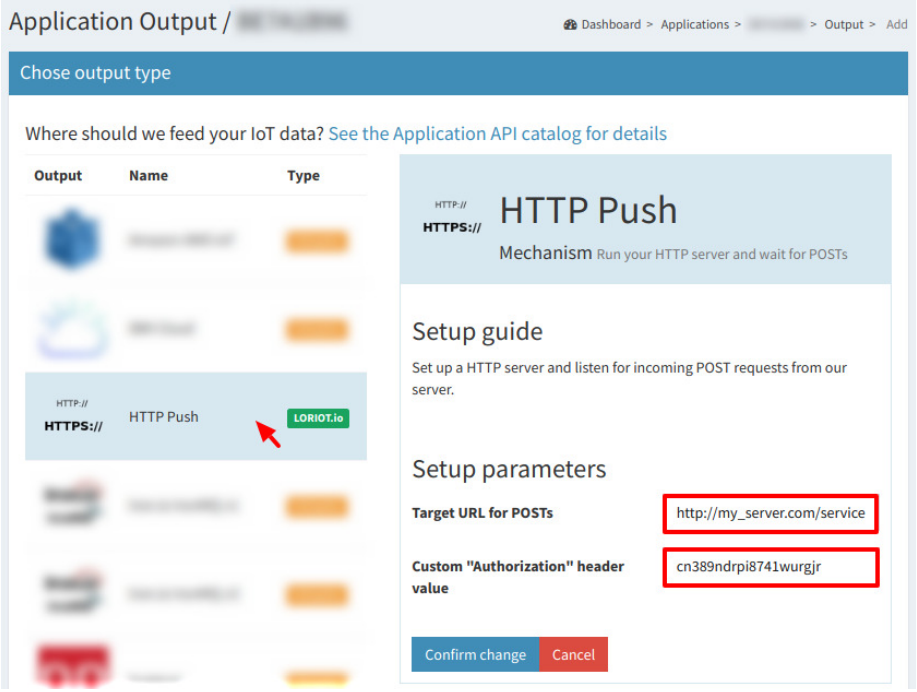
Mechanism Listen and wait

How to use WebSocket with LORIOT

Point your browser or library to the WebSocket URL and listen for incoming messages

Figure : Create a new application output

Select **"HTTP Push"** from the list and fill the form displayed on the right side. In the **"Target URL for POSTs"** field, write the URL where Loriot must send data to (i.e. http://my_server.com/services/actility/). Besides, users can add an authorization header in **"Custom Authorization header value"** in order to increase the security in the communications, so the final server only accepts the information from an authorized account. Finally, click on the **"Confirm change"** button to save the information.



Application Output / [redacted]
Dashboard > Applications > [redacted] > Output > Add
Chose output type
HTTP://
HTTPS://
Confirm change
Cancel

Where should we feed your IoT data? See the Application API catalog for details

Output	Name	Type
[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]
HTTP:// HTTPS://	HTTP Push	LORIOT.io
[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]

HTTP Push

Mechanism Run your HTTP server and wait for POSTs

Setup guide

Set up a HTTP server and listen for incoming POST requests from our server.

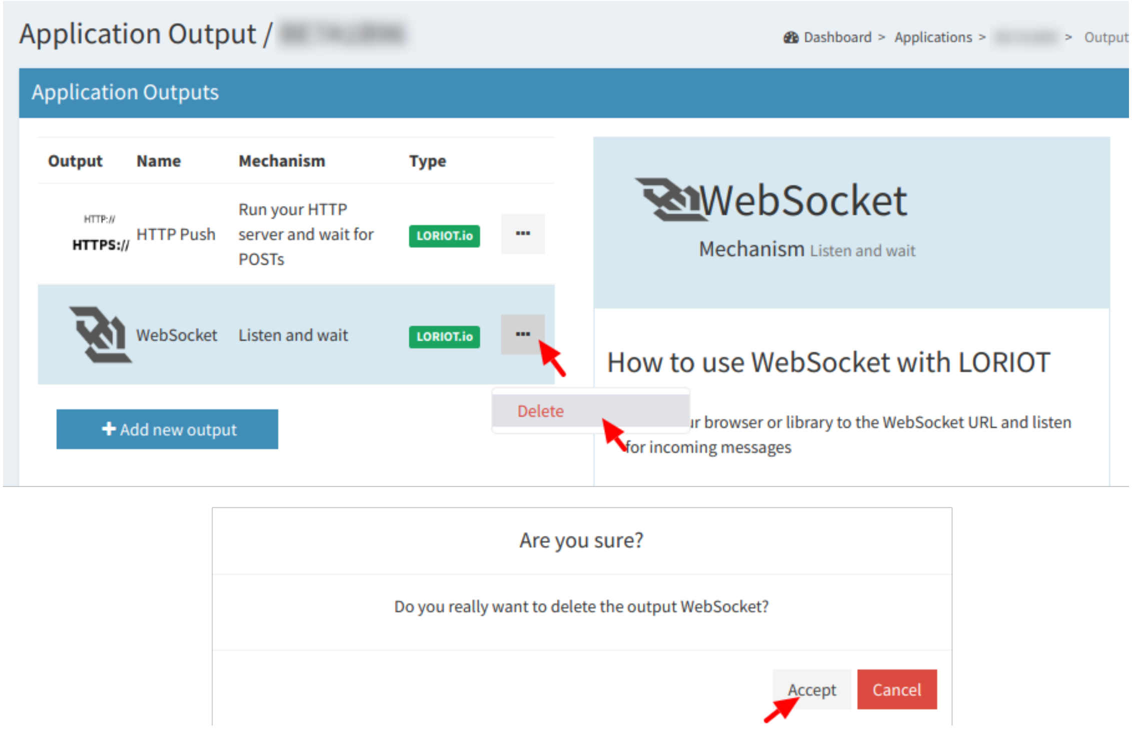
Setup parameters

Target URL for POSTs	http://my_server.com/service
Custom "Authorization" header value	cn389ndrpi8741wurgjr

Figure : Application output configuration

6.1.4. How to delete unused Loriot applications

A list with all application outputs will be displayed in the "Application Outputs" window. It is recommended to delete unused outputs and keep only "HTTP Push" in the list. The procedure to delete all other applications is to click on the desired ellipsis button of the application **"..."** and select **"Delete"** option. A pop-up window will ask you to confirm the operation.



Application Output / [redacted]

Output	Name	Mechanism	Type
HTTP:// HTTPS://	HTTP Push	Run your HTTP server and wait for POSTs	LORIOT.io
	WebSocket	Listen and wait	LORIOT.io

WebSocket

Mechanism Listen and wait

How to use WebSocket with LORIOT

...r browser or library to the WebSocket URL and listen for incoming messages

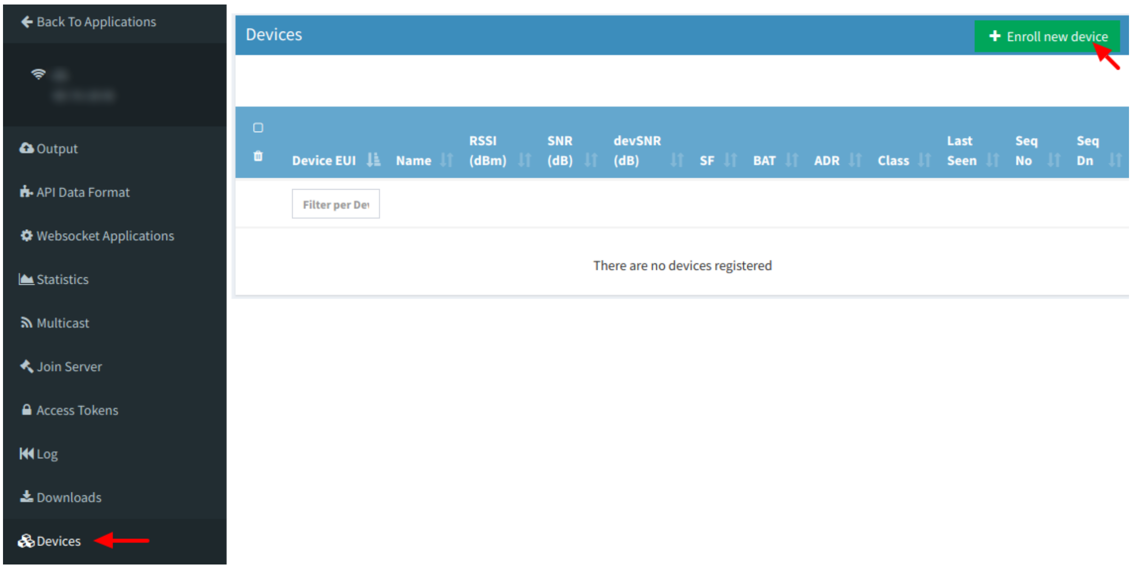
Are you sure?

Do you really want to delete the output WebSocket?

Figure : Delete unused application outputs

6.1.5. How to create a single device manually

Click on **"Devices"** left menu to show the device list. Click on **"Enroll new device"** button to add a new device:

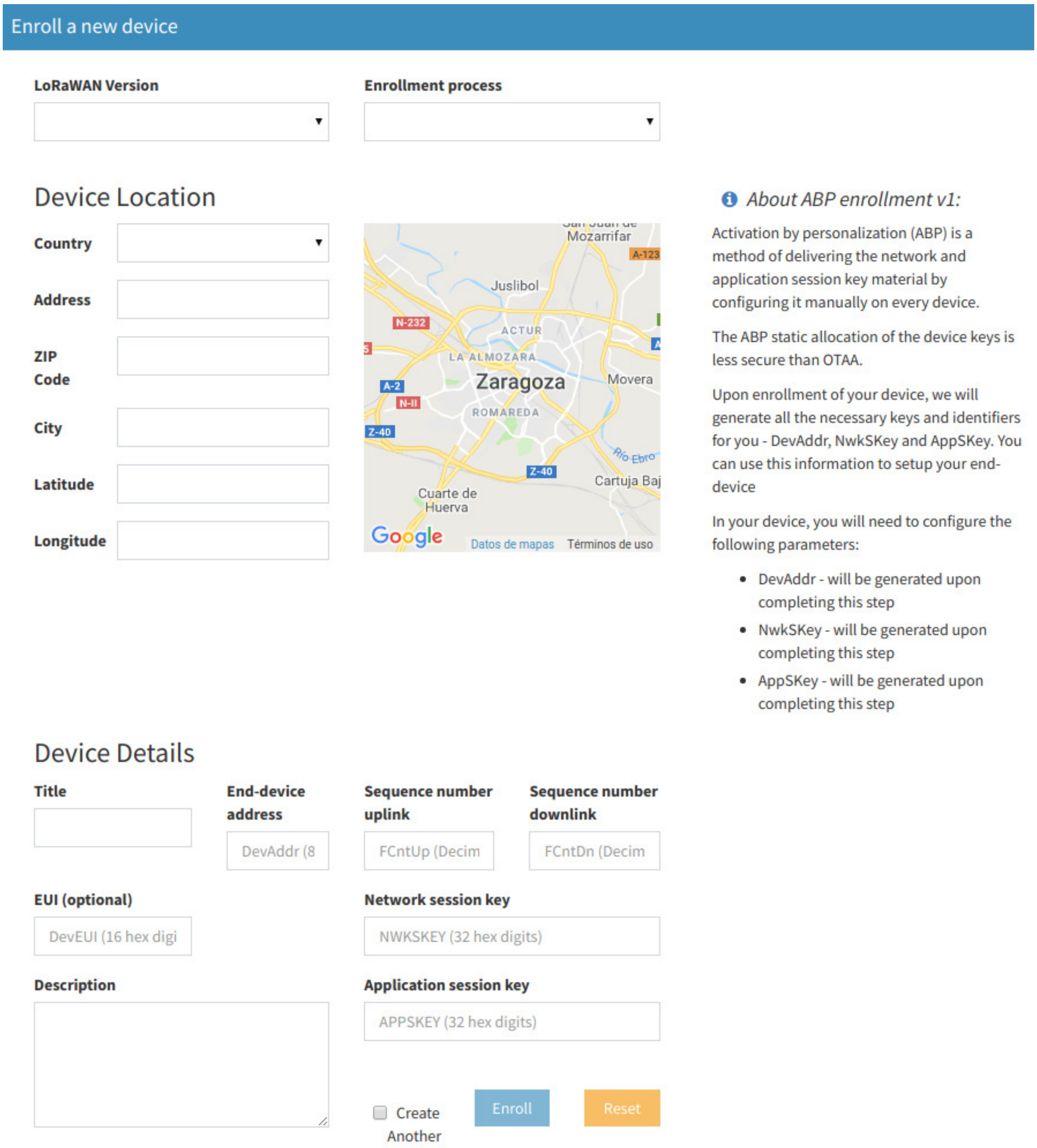


□	Device EUI	Name	RSSI (dBm)	SNR (dB)	devSNR (dB)	SF	BAT	ADR	Class	Last Seen	Seq No	Seq Dn

There are no devices registered

Figure : Enroll new device

A new form will be displayed in order to create the new device associated to the application. In this form it is possible to choose different enrolling processes based on OTAA or ABP. So, depending on the option chosen, it will be mandatory to insert different EUIs or keys. Please refer to "LoRaWAN protocol and parameters" to understand the all parameters involved in this form.



Enroll a new device

LoRaWAN Version ▾

Enrollment process ▾

Device Location

Country	▾
Address	
ZIP Code	
City	
Latitude	
Longitude	

About ABP enrollment v1:

Activation by personalization (ABP) is a method of delivering the network and application session key material by configuring it manually on every device.

The ABP static allocation of the device keys is less secure than OTAA.

Upon enrollment of your device, we will generate all the necessary keys and identifiers for you - DevAddr, NwkSKey and AppSKey. You can use this information to setup your end-device

In your device, you will need to configure the following parameters:

- DevAddr - will be generated upon completing this step
- NwkSKey - will be generated upon completing this step
- AppSKey - will be generated upon completing this step

Device Details

Title	End-device address	Sequence number uplink	Sequence number downlink
	DevAddr (8	FCntUp (Decim	FCntDn (Decim

EUI (optional)	Network session key
DevEUI (16 hex digi	NWKSKEY (32 hex digits)

Description	Application session key
	APPSKEY (32 hex digits)

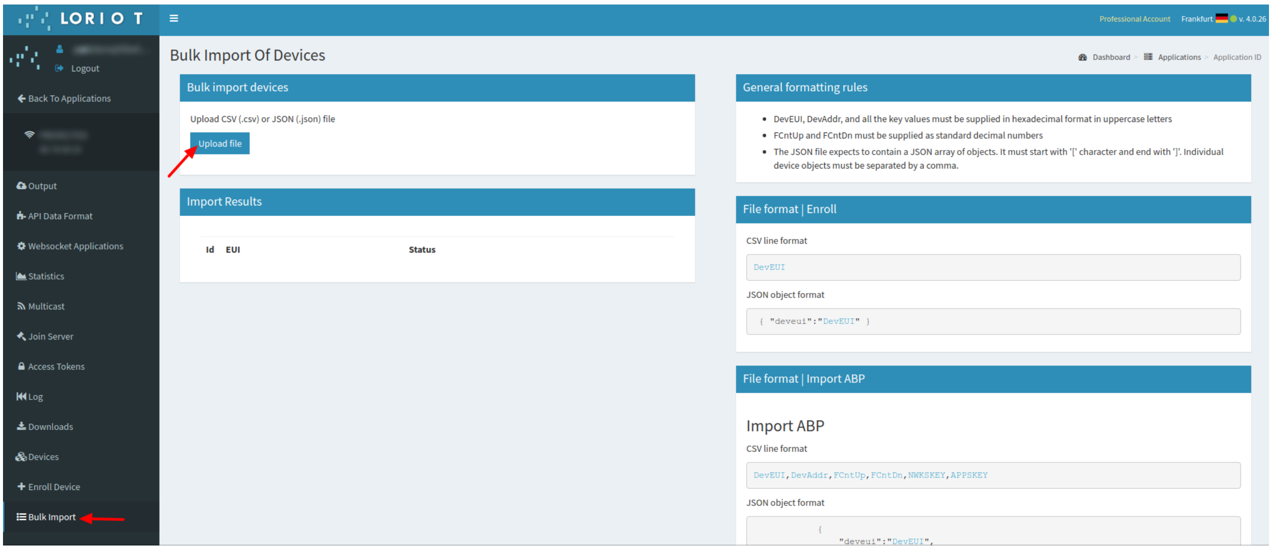
□ Create Another Enroll Reset

Figure : New device form

6.1.6. Nodes batch provisioning in Loriot

Inside the Loriot application click on **"Bulk import"** and then click on **"Upload file"** to register several devices at a time. It asks for a CSV file which must be exported from Libelium Cloud (SCM).

Please refer to "Libelium Cloud management" section for further information on how to export this CSV file which includes all nodes OTAA credentials needed for direct batch import.



Bulk Import Of Devices

Bulk import devices

Upload CSV (.csv) or JSON (.json) file

Import Results

Id	EUI	Status

General formatting rules

- DevEUI, DevAddr, and all the key values must be supplied in hexadecimal format in uppercase letters
- FCntUp and FCntDn must be supplied as standard decimal numbers
- The JSON file expects to contain a JSON array of objects. It must start with '[' character and end with ']'. Individual device objects must be separated by a comma.

File format | Enroll

CSV line format

```
DevEUI
```

JSON object format

```
{ "deveui":"DevEUI" }
```

File format | Import ABP

Import ABP

CSV line format

```
DevEUI,DevAddr,FCntUp,FCntDn,NWKSKEY,APPSKEY
```

JSON object format

```
        {
            "deveui":"DevEUI",
```

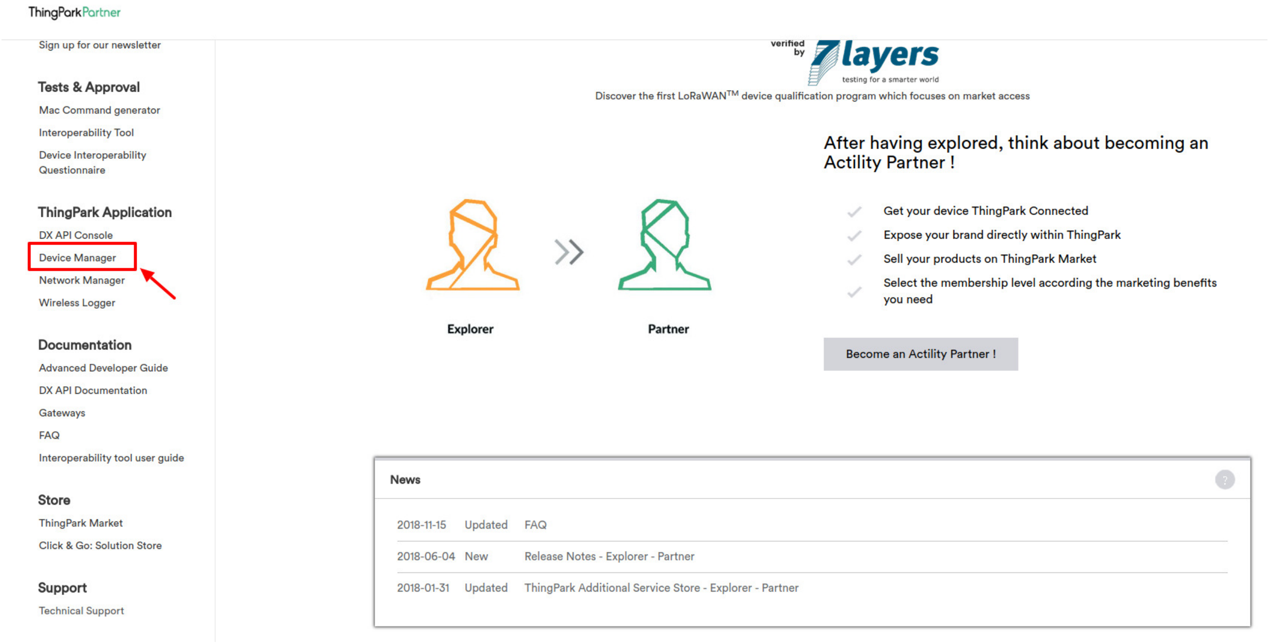
Figure : Loriot bulk import

The CSV file for 'N' devices must respect the next format:

```
<DevEUI_1>,<AppEUI_1>,<AppKey_1>
<DevEUI_2>,<AppEUI_2>,<AppKey_2>
<DevEUI_3>,<AppEUI_3>,<AppKey_3>
...
<DevEUI_N>,<AppEUI_N>,<AppKey_N>
```

6.2. Actility

This section explains how to route the information via Actility network server to the "Customer server" using the HTTPS protocol. Firstly, you must access into the Actility's ThingPark portal and log in. IN the main portal, you must go to **"ThingPark Application"** section and access to **"Device Manager"**:



ThingPark Partner

Sign up for our newsletter

Tests & Approval

Mac Command generator

Interoperability Tool

Device Interoperability Questionnaire

ThingPark Application

DX API Console

Device Manager

Network Manager

Wireless Logger

Documentation

Advanced Developer Guide

DX API Documentation

Gateways

FAQ

Interoperability tool user guide

Store

ThingPark Market

Click & Go: Solution Store

Support

Technical Support

verified by 7layers testing for a smarter world

Discover the first LoRaWAN^{TM} device qualification program which focuses on market access

Explorer

Partner

After having explored, think about becoming an Actility Partner !

- ✓ Get your device ThingPark Connected
- ✓ Expose your brand directly within ThingPark
- ✓ Sell your products on ThingPark Market
- ✓ Select the membership level according the marketing benefits you need

Become an Actility Partner !

News

2018-11-15	Updated	FAQ
2018-06-04	New	Release Notes - Explorer - Partner
2018-01-31	Updated	ThingPark Additional Service Store - Explorer - Partner

Figure : Access to "Device Manager" menu

Actility requires an "Application Server" and an "AS routing profile" in order to establish the communication with the Customer server or Libelium Cloud. These two elements are required before starting the service the first time, and will be the base to create new devices.

6.2.1. How to create a new Application Server

Starting from the main ThingPark Device Manager window, you shall create a new "Application Server" by clicking on **"Application Servers"**:

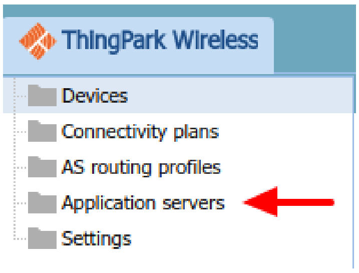
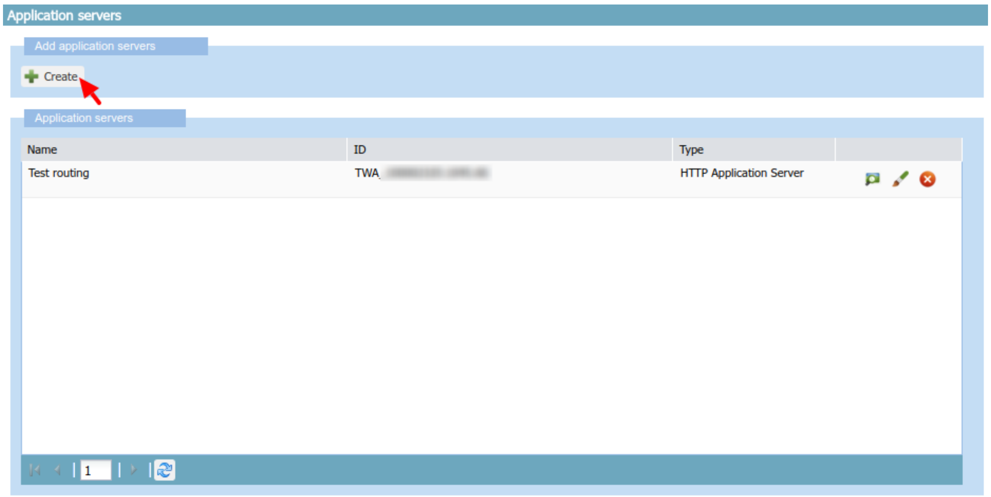


Figure : Device Manager menu

The first section allows you to create a new "Application Server". Then the second section shows a list of all "Application servers" created. Click on the **"Create"** button in the first section to create a new "Application Server":



Application servers

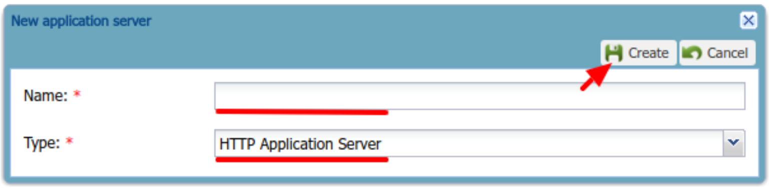
Add application servers

Application servers

Name	ID	Type
Test routing	TWA_[redacted]	HTTP Application Server

Figure : Application servers menu

In "Name" field define your Application server name. In "Type" field, you must select "HTTP Application Server". Finally, click on the **"Create"** button to continue:



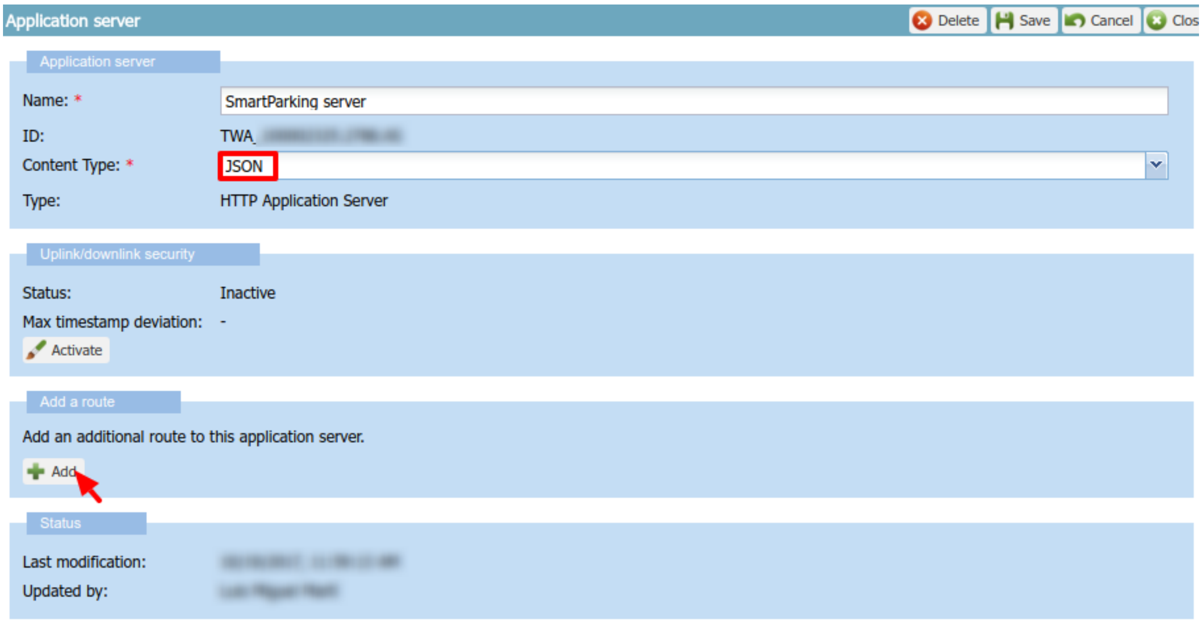
New application server

Name: *

Type: * HTTP Application Server

Figure : Application server name

A new window appears, where the name is copied from the previous pop-up. In the "Content Type" field you must select **"JSON"**. Then click on the **"Add"** button in "Add a route" section to create the link to the callback server installed:



Application server

Application server

Name: * SmartParking server

ID: TWA_[redacted]

Content Type: * JSON

Type: HTTP Application Server

Uplink/downlink security

Status: Inactive

Max timestamp deviation: -

Add a route

Add an additional route to this application server.

Status

Last modification: [redacted]

Updated by: [redacted]

Figure : Application server form

A "Route" section will be generated with a new form to be completed. Just click the **"Add"** button to proceed:

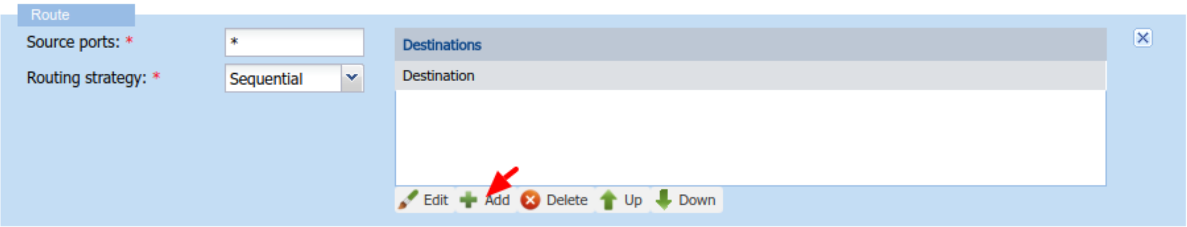
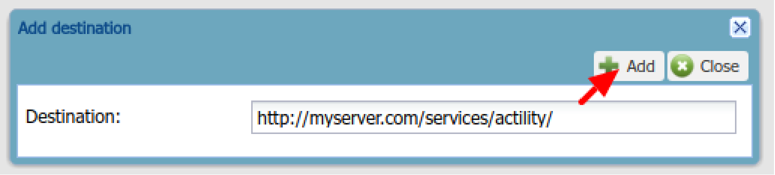


Figure : Route form

A new pop-up appears. In the "Destination" field, you must enter the URL of the server where the data is going to be sent. Click on the **"Add"** button to save the information:



Destination: http://myserver.com/services/actility/

Figure : Destination form

Navigate back to the Application server form, click on **"Save"** in order to create the "Application server".

6.2.2. How to create a new AS routing profile

The Application Server (AS) routing profile defines how a previously created "Application server" will behave. In order to create a new "AS routing profile" in the Device Manager, click on **"AS routing profiles"** button on the left sidebar menu:

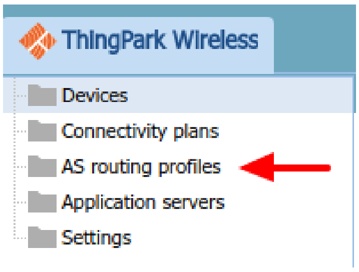
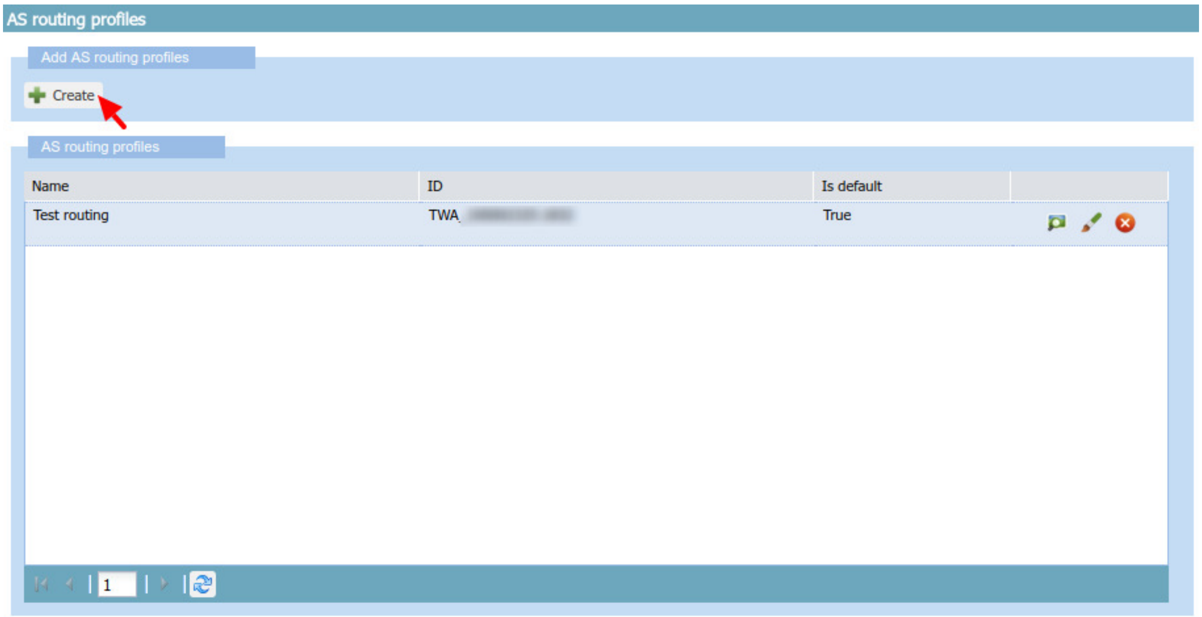


Figure : Destination form

Then you must click on the **"Add"** button in the first section. Below, in the second section, a list with all existing "AS routing profiles" will be displayed:



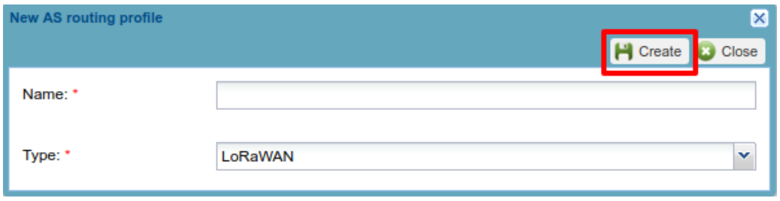
AS routing profiles

AS routing profiles

Name	ID	Is default
Test routing	TWA_[redacted]	True

Figure : AS Routing profiles

In the "Name" field write the name of the AS routing profile. In the "Type" field, select LoRaWAN. Then click on the **"Create"** button to continue the process:



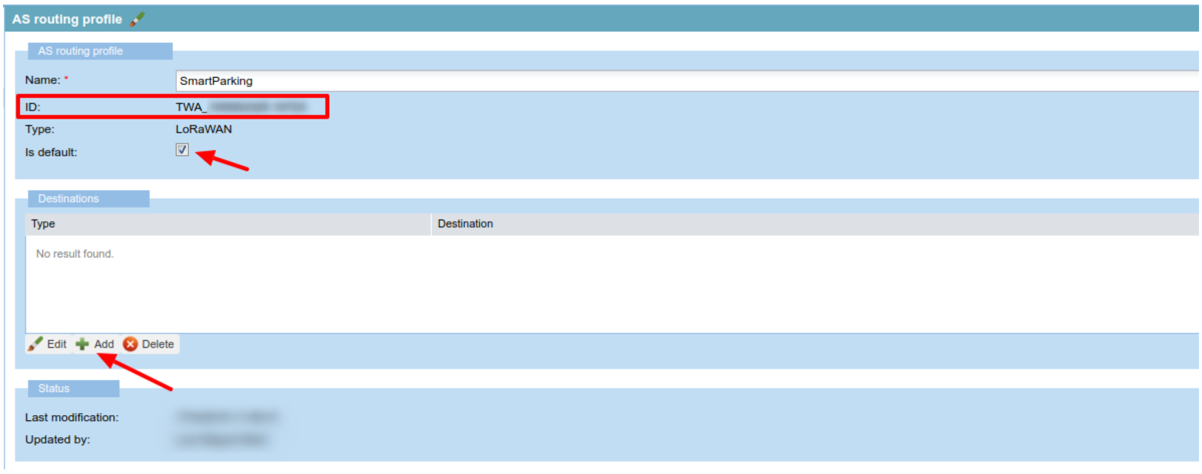
New AS routing profile

Name: *

Type: * LoRaWAN

Figure : AS Routing profile name

In the new window, you can read the "AS routing profile ID" which will be needed for batch import. Besides, mark the check "Is default" and click the **"Add button"**:



AS routing profile

AS routing profile

Name: * SmartParking

ID: TWA_[redacted]

Type: LoRaWAN

Is default: ☑

Destinations

Type	Destination
No result found.	

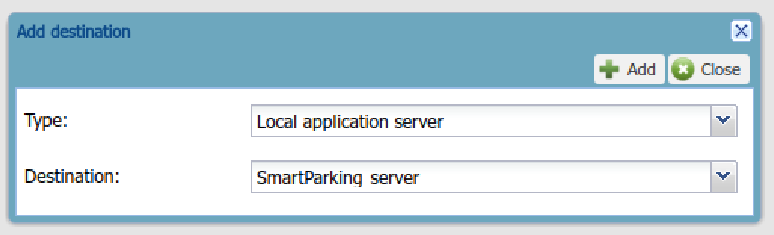
Status

Last modification: [redacted]

Updated by: [redacted]

Figure : AS Routing profile route

A new pop-up will appear with two parameters to be completed. "Type" is the kind of the application for our destination, **"Local application server"** must be selected. In "Destination" drop-down field, all available Application servers will be displayed. So, the correct "Application server" previously created must be selected:



Type: Local application server

Destination: SmartParking server

Figure : Add the created Application server as "destination" for the routing profile

Navigate back to the AS routing profile form, click on **"Save"** in order to create it. This process should be completed just once. From this moment, customers can add devices to the service.

6.2.3. How to create new devices manually

Once created the "Application server" and "AS routing profile", you can create new devices.

In the main ThingPark menu, click on **"Devices"** option . Then click on the **"Create"** button:

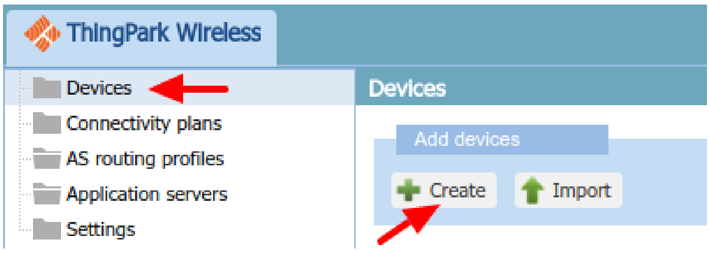


Figure : Create new devices

A new form will be displayed in order to define the new device’s parameters:

Administrative data

- Device name: Name to identify the device
- Marker: Icon to identify the device on the map list
- Administrative info: Relevant information of the device
- Administrative location: Location of the device
- Motion indicator: List with several options

Device identification

- Manufacturer: Select "Generic".
- Model: Select the corresponding region:
  - LoRaWAN EU863-870: LORA/GenericA.1.0.2a_ETSI_Rx2-SF12
  - LoRaWAN US902-928: LORA/GenericA.1.0.2a_FCC_Rx2-SF12
  - LoRaWAN IN865-867: LORA/GenericA.1revB_IN865_Rx2-SF12
  - LoRaWAN AU915-928: LORA/GenericA.1revB_AU915_Rx2-SF12
  - LoRaWAN AS923: LORA/GenericA.1_AS923_Rx2-SF10
- Device activation: Select "Over The Air Activation (OTAA)" if default parameters are used. ABP is the other option.
- DevEUI: Device EUI, globally unique IEEE EUI-64 address.
- AppEUI: Device Application identifier (mandatory for OTAA).
- Key format: Should be "Clear text" for non-encrypted key format.

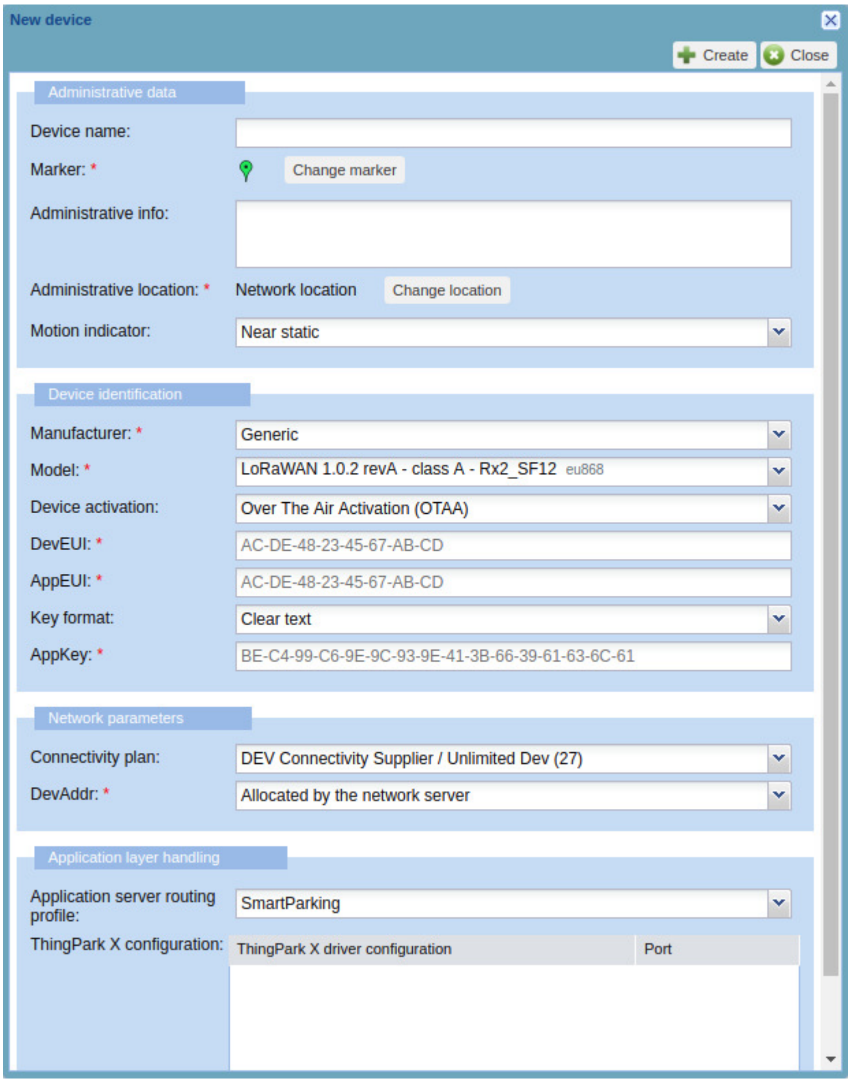
- AppKey: 16-byte Application Key (mandatory for OTAA).

Network parameters

- Connectivity plan: Select a Connectivity plan on the drop-down menu.
- DevAddr: Select "Allocated by the network server".

Application layer handling

- Application server routing profile: Select the "Application server" created before.



Administrative data

Device name:

Marker: *

Administrative info:

Administrative location: * Network location

Motion indicator: Near static

Device identification

Manufacturer: * Generic

Model: * LoRaWAN 1.0.2 revA - class A - Rx2_SF12 eu868

Device activation: Over The Air Activation (OTAA)

DevEUI: * AC-DE-48-23-45-67-AB-CD

AppEUI: * AC-DE-48-23-45-67-AB-CD

Key format: Clear text

AppKey: * BE-C4-99-C6-9E-9C-93-9E-41-3B-66-39-61-63-6C-61

Network parameters

Connectivity plan: DEV Connectivity Supplier / Unlimited Dev (27)

DevAddr: * Allocated by the network server

Application layer handling

Application server routing profile: SmartParking

ThingPark X configuration:

ThingPark X driver configuration	Port

Figure : Enter all device's parameters

Finally click on the **"Create"** button in order to save the information of the device.

6.2.4. Nodes batch provisioning in Actility

It is possible to register several nodes at a time using the **"Import"** button:

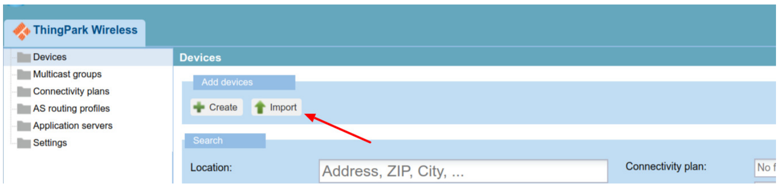
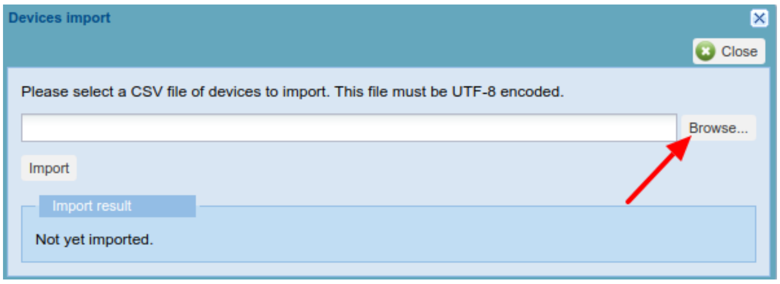


Figure : Import devices

A new window is displayed in order to import a CSV file with all devices to be imported:



Please select a CSV file of devices to import. This file must be UTF-8 encoded.

Not yet imported.

Figure : Import CSV file

Regarding the CSV file format, you must start from the exported CSV file from Libelium Cloud (SCM). Please refer to “Libelium Cloud management” section for further information on how to export this CSV file which includes all nodes OTAA credentials needed for direct batch import. Then, you must keep in mind that Actility needs more information to be included in that CSV file. The CSV to be imported must follow the next format (15 columns).
The CSV file for ’N’ devices must respect the next format:

```
CREATE_OTAA,<DevEUI_1>,,<DevProfile>,<AppEUI_1>,<AppKey_1>,,,<AS_profile_ID>,<ConnectivityPlan>,<name>,,,,,
CREATE_OTAA,<DevEUI_2>,,<DevProfile>,<AppEUI_2>,<AppKey_2>,,,<AS_profile_ID>,<ConnectivityPlan>,<name>,,,,,
CREATE_OTAA,<DevEUI_3>,,<DevProfile>,<AppEUI_3>,<AppKey_3>,,,<AS_profile_ID>,<ConnectivityPlan>,<name>,,,,,
...
CREATE_OTAA,<DevEUI_N>,,<DevProfile>,<AppEUI_N>,<AppKey_N>,,,<AS_profile_ID>,<ConnectivityPlan>,<name>,,,,,
```

Where:

- **CREATE_OTAA**: Should always be the same in order to import the devices using OTAA join mode.
- **<DevEUI>**: This is the Device EUI exported from Libelium SCM CSV file. Should not be changed.
- **<DevProfile>**: This field is related to the LoRaWAN region of the node. Possibilities depending on the version:
  - LoRaWAN EU863-870: LORA/GenericA.1.0.2a_ETSI_Rx2-SF12
  - LoRaWAN US902-928: LORA/GenericA.1.0.2a_FCC_Rx2-SF12
  - LoRaWAN IN865-867: LORA/GenericA.1revB_IN865_Rx2-SF12
  - LoRaWAN AU915-928: LORA/GenericA.1revB_AU915_Rx2-SF12
  - LoRaWAN AS923: LORA/GenericA.1_AS923_Rx2-SF10
- **<AppEUI>**: This is the Application EUI exported from Libelium SCM CSV file. Should not be changed.
- **<AppKey>**: This is the Application Key exported from Libelium SCM CSV file. Should not be changed.
- **<AS_profile_ID>**: This is Actility’s "AS routing profile ID" which belongs to each customer’s account. It can be read from the profile when it is first created.
- **<ConnectivityPlan>**: This must be set as "dev-cs/testing".
- **<name>**: Name of each device (optional).

Example for some Smart Parking EU nodes (LoRaWAN EU863-870). Let’s assume TWA_123 was the "AS routing profile ID" given by Actility for our application:

```
CREATE_OTAA,<EUI1>,,LORA/GenericA.1.0.2a_ETSI_Rx2-SF12,<AppEUI1>,<AppKey1>,,dev-cs/testing,TWA_123,,Node1,,,,,
CREATE_OTAA,<EUI2>,,LORA/GenericA.1.0.2a_ETSI_Rx2-SF12,<AppEUI2>,<AppKey2>,,dev-cs/testing,TWA_123,,Node2,,,,,
CREATE_OTAA,<EUI3>,,LORA/GenericA.1.0.2a_ETSI_Rx2-SF12,<AppEUI3>,<AppKey3>,,dev-cs/testing,TWA_123,,Node3,,,,,
CREATE_OTAA,<EUI4>,,LORA/GenericA.1.0.2a_ETSI_Rx2-SF12,<AppEUI4>,<AppKey4>,,dev-cs/testing,TWA_123,,Node4,,,,,
CREATE_OTAA,<EUI5>,,LORA/GenericA.1.0.2a_ETSI_Rx2-SF12,<AppEUI5>,<AppKey5>,,dev-cs/testing,TWA_123,,Node5,,,,,
```

7. Customer Server

In order to communicate with any LoRaWAN Network Server you need a web application up and running. The web application will receive LoRaWAN requests via POST/GET requests depending on the server. Libelium provides the source files of a simple web application called Customer Server, created to be deployed on your server.

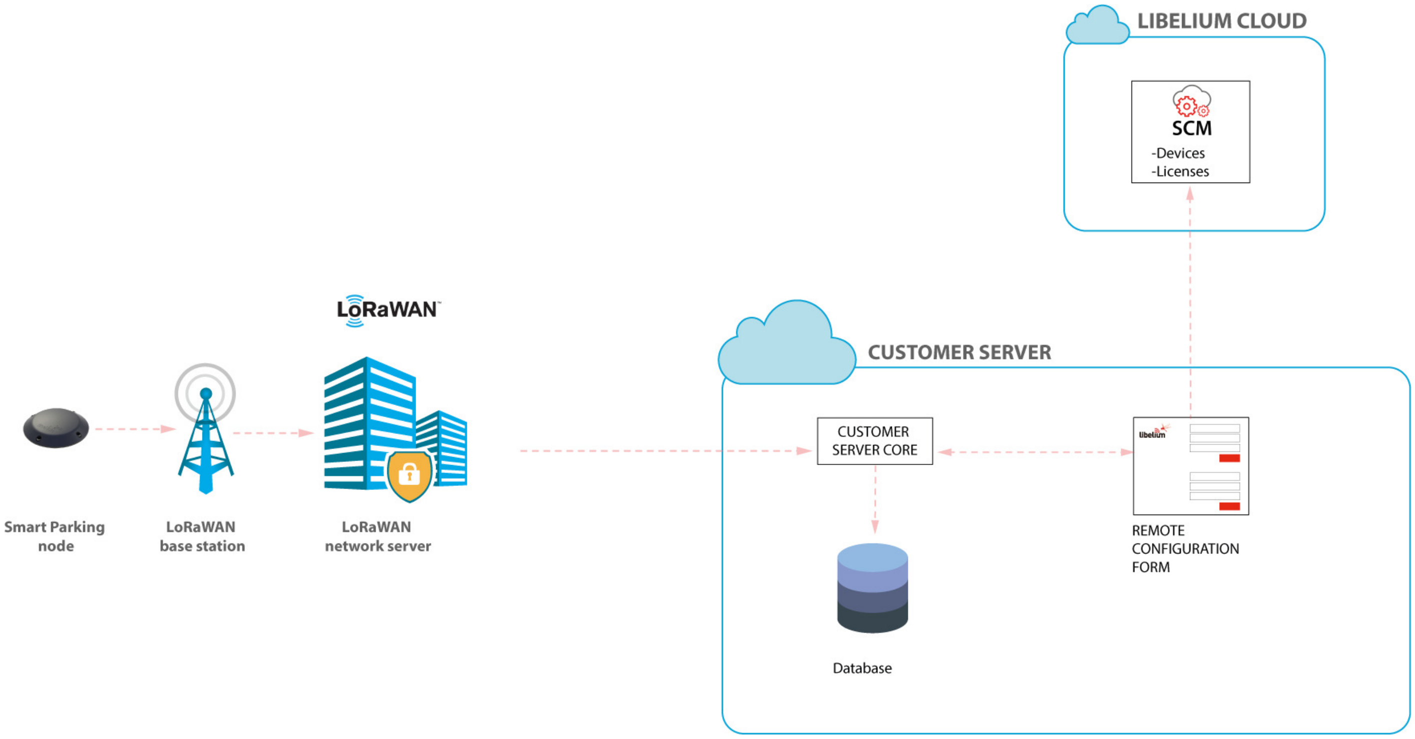


Figure : Customer Server

The Customer Server is composed of 2 parts:

- the Remote Configuration Form: it manages the configuration values of the Smart Parking nodes)
- the Customer Server Core: it deals with the LoRaWAN Network Server requests, sending back a response when needed.

If the user wants to use the Remote Configuration Form, all Smart Parking nodes must be registered on the Libelium Cloud. There is no need of additional licenses.

The Customer Server must be deployed on your own server. Besides, the LoRaWAN Network Server must be configured with the complete URL containing your server domain name or IP. The context should point to the web application deployment path.

Note: Customers can ask for this source code to our Sales Team (sales@libelium.com) after buying the nodes.

7.1. Installation

There are 2 ways to install the Customer Server: installing it on your own server with Docker or installing it manually on your own server.

Assigning a public IP and a registered domain name to the server is recommended to ease the task of configuring LoRaWAN Network Server to send HTTP/HTTPS requests to this server. It is also a good practice to implement existing security policies (user credentials, SSL, firewall, and tools to avoid DoS attacks) for servers with open ports to Internet.

7.1.1. Docker

If you want to use Docker, you must install Docker and Docker Compose on your server. In order to install these tools, you should follow the instructions from the official documentation by Docker:

- Docker installation instructions: https://docs.docker.com/install/
- Docker Compose installation instructions: https://docs.docker.com/compose/install/

You can also hire online services which offer a virtual server with Docker already installed on it.

7.1.2. Server

The minimum requirements for the server are:

- Apache web server >= 2.4
- PHP >= 7.1.3
- Database: MySQL or PostgreSQL
- ionCube Loader

An Apache webserver with PHP support and a database must be configured on your server. Those are the minimum requirements to deploy the Customer Server web application.

Even if you are not using Docker, you can use 2 files as a reference to install all the dependencies of the project. These files are Dockerfile and docker-compose.yml, and can be found on the source code that the Sales Team provides (bin/ci/smart_parking).

Info and tutorials about installation of minimum requirements of your server:

- Ubuntu: https://help.ubuntu.com/community/ApacheMySQLPHP
- Windows: http://www.ampsoft.net/webdesign-l/how-to-install-apache-php-mysql.html
- Mac: http://jason.pureconcepts.net/2012/10/install-apache-php-mysql-mac-os-x/
- ionCube Loader: http://www.ioncube.com/loaders.php

7.2. Deploying

7.2.1. Docker

- Step 1: Extract on your server the zip file provided by Libelium containing the Customer Server application source files.
- Step 2: Go to the folder bin/ci/smart_parking.
- Step 3: You should modify the files Dockerfile and docker-compose.yml that we provide in order to make sure that the security required for your project is matched.

- **Database**: We provide a file with the structure for the database, you can change the name of the main table and also use it to add any additional tables you may need, the file is located on:

  ```
  bin/ci/smart_parking/docker-entrypoint-initdb.d/init.sql
  ```

  This file will be executed one time when the database container is created and it should be added to the docker group.

- File structure: The application files should be placed in the "www-data" group from the web container. You can map a host user to www-data by UID as it is commented on the Dockerfile or you can assign the files to the correct UID (33 by default).

- Step 4: Execute "docker-compose up" to launch the project (the 1st time, it may take some time to execute).
- Step 5: You can check if the installation is completed navigating to "http://MACHINE_IP:7000": you should see the Remote Configuration Form login screen.

7.2.2. Server

- Step 1: Extract on your server the zip file provided by Libelium containing the Customer Server application source files.
- Step 2: Check the right owner/group and permissions of all the files extracted, usually using www-data group is default in Ubuntu environment.
- Step 3: Check the permissions of all folders and files, usually using 0770 for directories and 0660 for files is default in Ubuntu environment.
- Step 4: Configure in your server the context of the callback server application with the path where the source files were extracted.

  Copy and paste the following text as content of the file "my_server.conf" file located on "/etc/apache2/sites-available" (replace paths to match your server deployment locations):

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        ServerName my_server.com
        ServerAlias my_server.com

        DirectoryIndex index.html index.php
        DocumentRoot /path/zip/extracted

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        <Directory /path/zip/extracted>
                Options Indexes FollowSymLinks
                AllowOverride All
                Require all granted
        </Directory>
</VirtualHost>
```

7.3. Configuring the application

Initialize the nodes storage:

```
cd src/storage/app/parking
cp nodes.example.json nodes.json
```

Copy “src/.env.example” to “src/.env” and edit the file:

```
APP_NAME='Smart Parking v2 - Customer Server'
APP_ENV=local
APP_DEBUG=false
APP_URL=http://localhost
APP_TIMEZONE=UTC

# Set to true to use the database:
USE_DB=false

# This data should match the data on the file docker-compose.yml:
DB_CONNECTION=mysql
DB_HOST=db
DB_PORT=7001
DB_DATABASE=LoriotDB
DB_USERNAME=user
DB_PASSWORD=password

CACHE_DRIVER=file
```

```
# It is mandatory to set a service (Loriot|Actility):
SERVICE=''

# If you want to use Loriot you must fill these fields:
LORIOT_URI=''
LORIOT_APP_ID=''
LORIOT_TOKEN=''

LORIOT_API_KEY=''

# If you want to use Loriot you must fill these fields:
ACTILITY_URI=''
ACTILITY_TOKEN=''
```

If you want to use either Actility or Loriot you must specify which one you want to use; you must fill all the configuration fields.

7.3.1. How to configure Loriot

Write “Loriot” in the “SERVICE” field:

```
SERVICE='Loriot'
```

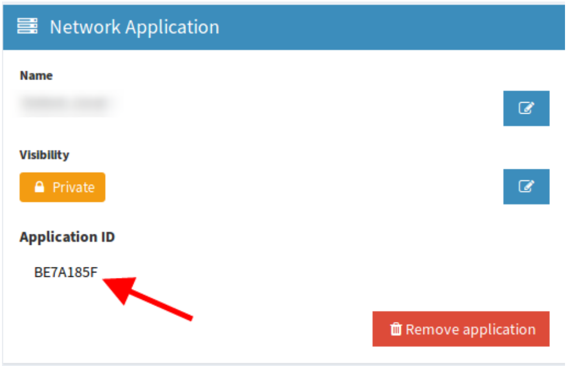
For the remaining fields:

```
LORIOT_URI='https://serverID.loriot.io/1/'
```

“serverId” is the Loriot server from which your Loriot application is served, you can find it on the URL of your Loriot account.

```
LORIOT_APP_ID='appId'
```

“appId” is the ID of your Loriot application. You can find it on the “Network application” section of your Loriot account:



Network Application

Name

Visibility

Private

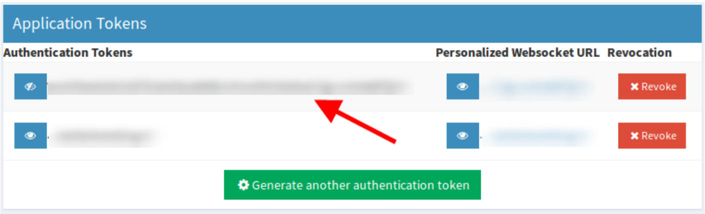
Application ID

BE7A185F

Figure : Loriot application ID

```
LORIOT_TOKEN='loriotToken'
```

“loriotToken” is the ID of your Loriot application. You can find it on the “Application tokens” section of your Loriot account:



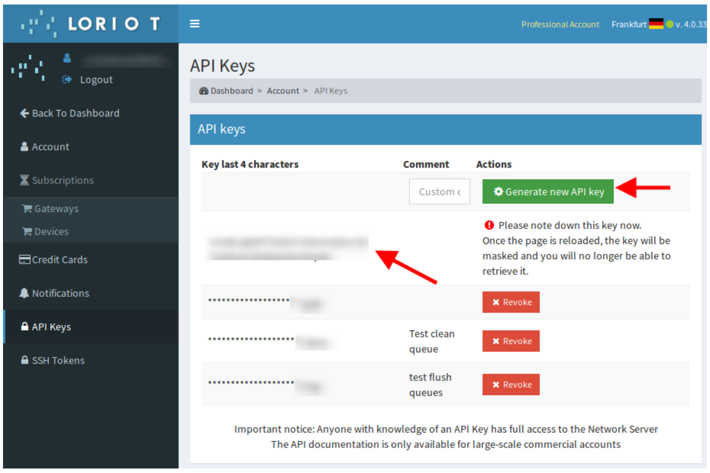
Application Tokens

Authentication Tokens	Personalized Websocket URL	Revocation
		✖ Revoke
		✖ Revoke

Figure : Loriot token

```
LORIOT_API_KEY='loriotApiKey'
```

“loriotApiKey” is a key you can obtain on the “Account” section of the Loriot panel:



API Keys

API keys

Key last 4 characters	Comment	Actions
	Custom c	Generate new API key
[redacted]		Please note down this key now. Once the page is reloaded, the key will be masked and you will no longer be able to retrieve it.
******************[redacted]		✖ Revoke
******************[redacted]	Test clean queue	✖ Revoke
******************[redacted]	test flush queues	✖ Revoke

Important notice: Anyone with knowledge of an API Key has full access to the Network Server
The API documentation is only available for large-scale commercial accounts

Figure : Loriot API key

7.3.2. How to configure Actility

Write “Actility” in the “SERVICE” field:

```
SERVICE='Actility'
```

And the remaining field:

```
ACTILITY_URI='actilityURI'
```

“actilityURI” is the downlink URL that Actility provides to send information from the server to the node. This URL is the address of the primary Actility LRC cluster.

7.4. Making the server accessible from anywhere

7.4.1. Mandatory for Loriot and Actility platforms

Optimal deployment includes a server name pointing to a public and static IP. Using dynamic DNS could be done with services like no-ip which has a free package (see http://www.noip.com/). No-ip has also a client application responsible for updating any IP address changes in the background (for more information go to http://www.noip.com/download).

Finally, the URL to be configured in the LoRaWAN Network Server must be created concatenating the server name and the paths:

Loriot: https://my_server.com/api/parking/v2/services/loriot

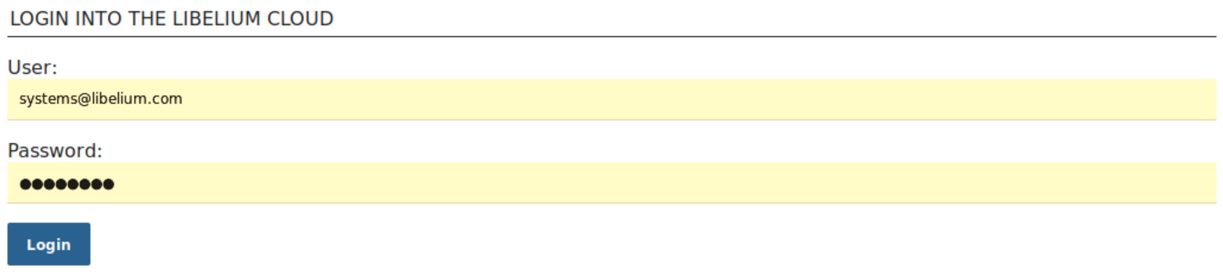
Actility: https://my_server.com/api/parking/v2/services/actility

7.5. Remote Configuration Form

The Remote Configuration Form is a web application which permits to update some node parameters (sensor and timing parameters). This is done thanks to the OTA-S (Over-The-Air-Setup) functionality. The update is done remotely via LoRaWAN downlink packets. LoRaWAN parameters cannot be updated with the Remote Configuration

Form.

For accessing the Remote Configuration Form, log in with your user and password from your Libelium Cloud account. Enter your credentials and press “Login”.



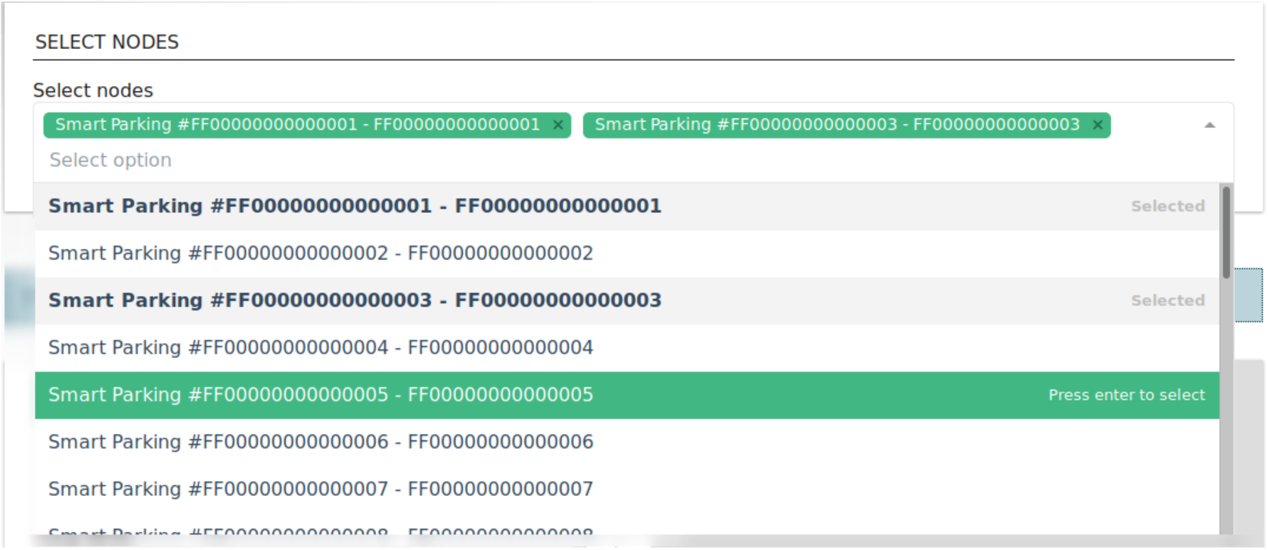
LOGIN INTO THE LIBELIUM CLOUD

User:
systems@libelium.com

Password:
●●●●●●●●

Figure : User login

Select one or several devices on the drop-down menu. Each device is identified by their unique LoRaWAN EUI. The nodes displayed in the drop-down menu correspond to the nodes you have claimed for on your Libelium Cloud account. You can also deselect previously selected nodes using the cross icon.



SELECT NODES

Select nodes

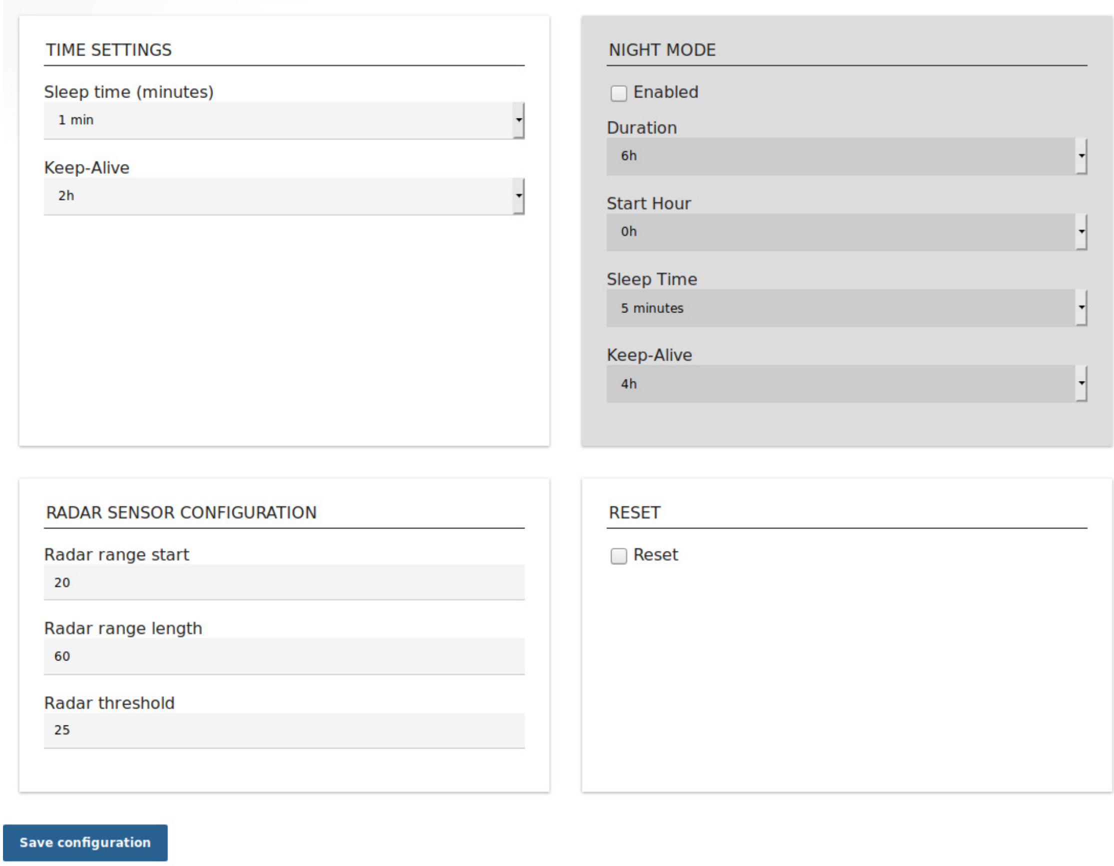
Smart Parking #FF00000000000001 - FF00000000000001 ×
Smart Parking #FF00000000000003 - FF00000000000003 ×

Select option

Smart Parking #FF00000000000001 - FF00000000000001 Selected
Smart Parking #FF00000000000002 - FF00000000000002
Smart Parking #FF00000000000003 - FF00000000000003 Selected
Smart Parking #FF00000000000004 - FF00000000000004
Smart Parking #FF00000000000005 - FF00000000000005 Press enter to select
Smart Parking #FF00000000000006 - FF00000000000006
Smart Parking #FF00000000000007 - FF00000000000007
Smart Parking #FF00000000000008 - FF00000000000008

Figure : Smart Parking node selection

Once a node is selected, its current configuration is displayed and you can modify it. For further information of the node parameters, please refer to the chapter “Smart Parking node”.



TIME SETTINGS

Sleep time (minutes): 1 min

Keep-Alive: 2h

NIGHT MODE

□ Enabled

Duration: 6h

Start Hour: 0h

Sleep Time: 5 minutes

Keep-Alive: 4h

RADAR SENSOR CONFIGURATION

Radar range start: 20

Radar range length: 60

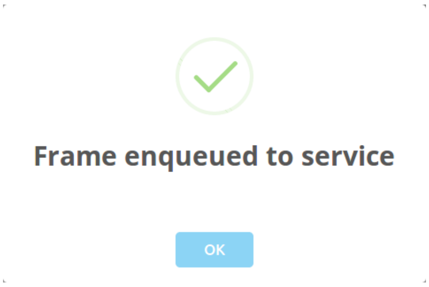
Radar threshold: 25

RESET

□ Reset

Figure : Smart Parking node configuration

Once you have configured the node or nodes, press the “Save configuration” button. A message is shown indicating that a downlink frame with the node’s configuration has been enqueued on the downlink queue of the LoRaWAN Network Server. Therefore, the node will receive a downlink frame with the new configuration in response to the next Info frame or Keep Alive frame.



Frame enqueued to service

Figure : Frame enqueued for processing

So the Remote Configuration Form application allows the user to update some node parameters, just as the Smart Devices App does. However, using the Smart Devices App requires physical access to the Smart Parking node: the node must be plugged to a computer with a USB cable.

Note: The Remote Configuration Form does not permit to update any LoRaWAN parameter. The Smart Devices App should be used instead.

7.6. Customer Server Core

The Customer Server Core deals with the LoRaWAN Network Server requests delivering the proper replies if needed. When it receives a new uplink frame, it parses its data to identify all fields and perform adequate operations. By default, the source code is prepared to insert the parsed data into a MySQL database.

Regarding the frame type, if a "Start Frame 1" or "RTC Update Request" frames are received, then a new downlink frame is immediately enqueued into the LoRaWAN Network Server with the proper answer.

On the other hand, when the Remote Configuration Form is managed in order to update node parameters, a downlink frame is sent to the LoRaWAN Network Server with the new values.

For further information about the different frame types please refer to the "Frame types" section.

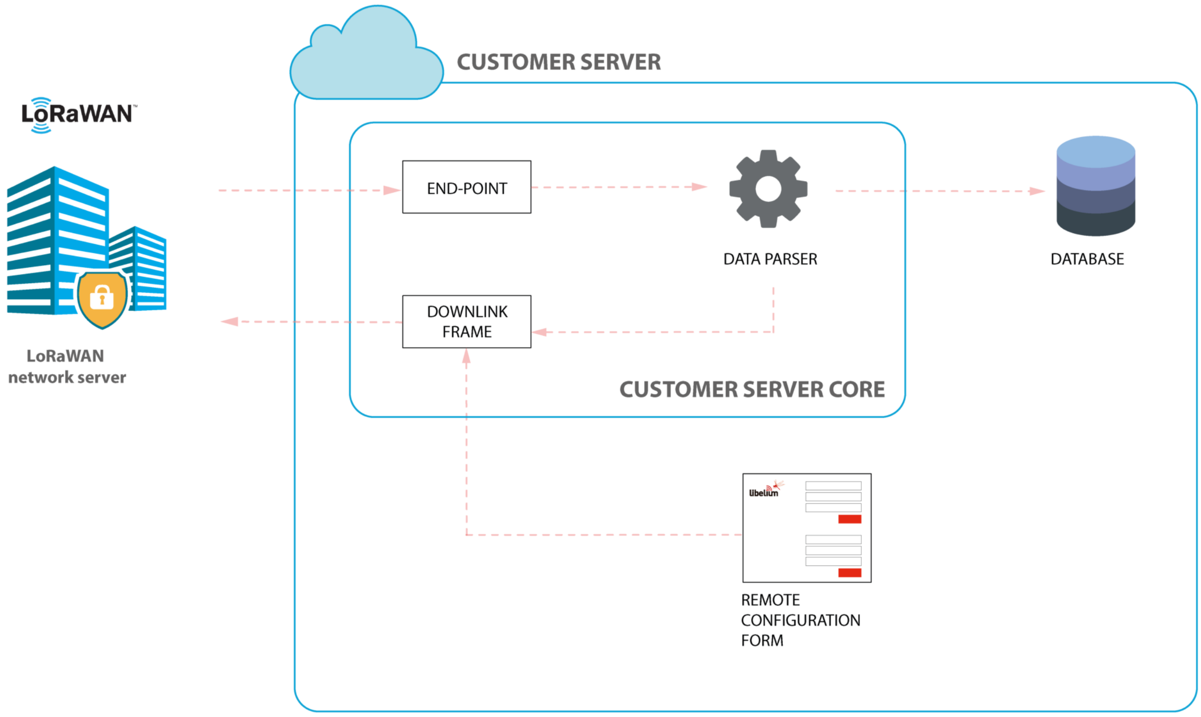


Figure : Customer Server Core

7.6.1. End-point

Once you have installed and configured the Customer Server, the system will be prepared to listen to LoRaWAN requests. The end-point will vary depending on the configured LoRaWAN service. By default, the source code is prepared for both Loriot and Actility network servers.

7.6.2. Data Parser

The Data Parser will unpack the payload data sent by the node before the insertion on the database. It will also prepare the proper response to the LoRaWAN Network Server if needed.

7.6.3. Database

The Customer Server Core inserts data into a MySQL database upon uplink frames arrival. The uplink frames are parsed by the Data Parser and the source code is capable of performing smart database insertions by accessing to previously stored information. For example, a loss in the sequence number is calculated by reading the previous sequence number insertion. Besides, the node time settings (Sleep time, Keep Alive time, etc) are stored since the "Start frame 1" and "Start frame 2" are received so it is possible to access to the current node configuration any time.

The structure of the database provided is:

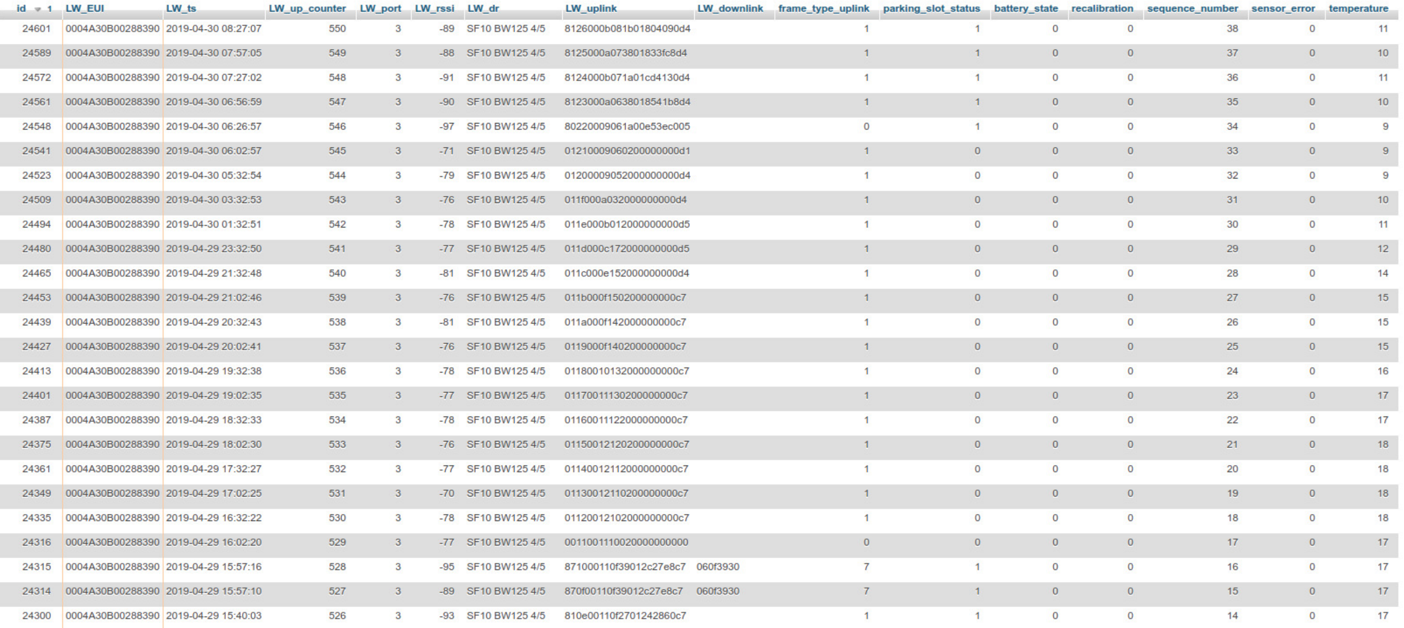
Database Field	Field updated by	Description
LW_EUI	All uplink frames	The node's unique LoRaWAN EUI.
LW_ts	All uplink frames	The uplink frame timestamp.
LW_up_counter	All uplink frames	The LoRaWAN protocol uplink counter. This field belongs to the link level. Range: from 0 to 4294967295.
LW_port	All uplink frames	The LoRaWAN port used to send the data. Range: from 1 to 223.
LW_rssi	All uplink frames	The uplink "Received Signal Strength Indication" (dBm).
LW_dr	All uplink frames	The uplink frame datarate.
LW_uplink	All uplink frames	The uplink frame contents in HEX digits.
LW_downlink	All uplink frames	The enqueued downlink frame in response to the uplink received. The contents in HEX digits. Normally this field is empty.
frame_type_uplink	All uplink frames	The frame type of the received uplink frame.
parking_slot_status	Info, Keep-alive, RTC update	The status of the parking lot (0: free; 1: occupied).
battery_state	All uplink frames	The battery state of the node (0: OK; 1: warning).
recalibration	All uplink frames	Flag to indicate sensor recalibration since last uplink (0: no; 1: yes).
sequence_number	All uplink frames	The application level sequence number. This field belongs to the frame contents. Range: from 0 to 255.
lost_up_counter (*)	All uplink frames	Field to indicate the loss of a LoRaWAN protocol sequence number.
lost_seq (*)	All uplink frames	Field to indicate the loss of an application sequence number.
sensor_error	Info, Keep-alive, RTC update	Field to indicate an error in sensor management (0: no error).
temperature	Info, Keep-alive, RTC update	The temperature value (Celsius degrees).
time_hh	Info, Keep-alive, RTC update	The node's RTC timestamp (hours field).
time_mm	Info, Keep-alive, RTC update	The node's RTC timestamp (minutes field).
distance	Info, Keep-alive, RTC update	The distance to an object provided by the sensor reading (mm units).
amplitude	Info, Keep-alive, RTC update	The amplitude of the object reflection provided by the sensor reading (no units).
n_reflections	Info	Number of reflections in sensor reading. The first reflection (most important one) is related to amplitude and distance values.

battery	Keep-alive, RTC update	The battery level read by the node (mV units).
radar_threshold	Start frame 1	The threshold configured to the radar sensor.
radar_range_start	Start frame 1	The starting range configured to the radar sensor (mm units).
radar_range_length	Start frame 1	The length of range configured to the radar sensor (mm units).
firmware_version	Start frame 1	The number related to firmware version.
sleep_time_minutes	Start frame 2	Time settings related to sleep time minutes field.
sleep_time_seconds	Start frame 2	Time settings related to sleep time seconds field.
keep_alive	Start frame 2	Time settings related to Keep-alive time field (hours field except for 0.5h which is equal to zero).
nm	Start frame 2	Field to indicate if night-mode is active (0: disabled; 1: enabled).
nm_start_hour	Start frame 2	Field to indicate when night-mode starts (hours field).
nm_duration	Start frame 2	Field to indicate night-mode duration (hours field).
nm_sleep_time	Start frame 2	Field to indicate night-mode sleep time (minutes field).
nm_keep_alive	Start frame 2	Field to indicate night-mode Keep-alive time (hours field).

(*) Fields calculated by the Data Parser by accessing to previous database insertions.

For further information about the fields please refer to the "Frame types" chapter.

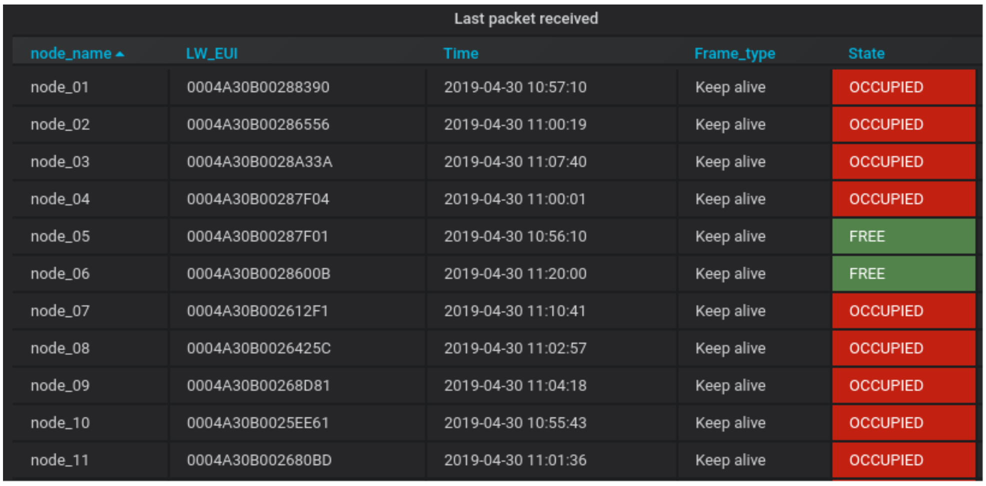
The next figure shows a fragment of the table on the database where data was inserted for a specific LoRaWAN EUI:



id ▼ 1	LW_EUI	LW_ts	LW_up_counter	LW_port	LW_rssi	LW_dr	LW_uplink	LW_downlink	frame_type_uplink	parking_slot_status	battery_state	recalibration	sequence_number	sensor_error	temperature
24601	0004A30B00288390	2019-04-30 08:27:07	550	3	-89	SF10 BW125 4/5	8126000b081b01804090d4		1	1	0	0	38	0	11
24589	0004A30B00288390	2019-04-30 07:57:05	549	3	-88	SF10 BW125 4/5	8125000a073801833fc8d4		1	1	0	0	37	0	10
24572	0004A30B00288390	2019-04-30 07:27:02	548	3	-91	SF10 BW125 4/5	8124000b071a01cd4130d4		1	1	0	0	36	0	11
24561	0004A30B00288390	2019-04-30 06:56:59	547	3	-90	SF10 BW125 4/5	8123000a0638018541b8d4		1	1	0	0	35	0	10
24548	0004A30B00288390	2019-04-30 06:26:57	546	3	-97	SF10 BW125 4/5	80220009061a00e53ec005		0	1	0	0	34	0	9
24541	0004A30B00288390	2019-04-30 06:02:57	545	3	-71	SF10 BW125 4/5	0121000906020000000d1		1	0	0	0	33	0	9
24523	0004A30B00288390	2019-04-30 05:32:54	544	3	-79	SF10 BW125 4/5	0120000905200000000d4		1	0	0	0	32	0	9
24509	0004A30B00288390	2019-04-30 03:32:53	543	3	-76	SF10 BW125 4/5	011f000a032000000000d4		1	0	0	0	31	0	10
24494	0004A30B00288390	2019-04-30 01:32:51	542	3	-78	SF10 BW125 4/5	011e000b012000000000d5		1	0	0	0	30	0	11
24480	0004A30B00288390	2019-04-29 23:32:50	541	3	-77	SF10 BW125 4/5	011d000c172000000000d5		1	0	0	0	29	0	12
24465	0004A30B00288390	2019-04-29 21:32:48	540	3	-81	SF10 BW125 4/5	011c000e152000000000d4		1	0	0	0	28	0	14
24453	0004A30B00288390	2019-04-29 21:02:46	539	3	-76	SF10 BW125 4/5	011b000f150200000000c7		1	0	0	0	27	0	15
24439	0004A30B00288390	2019-04-29 20:32:43	538	3	-81	SF10 BW125 4/5	011a000f142000000000c7		1	0	0	0	26	0	15
24427	0004A30B00288390	2019-04-29 20:02:41	537	3	-76	SF10 BW125 4/5	0119000f140200000000c7		1	0	0	0	25	0	15
24413	0004A30B00288390	2019-04-29 19:32:38	536	3	-78	SF10 BW125 4/5	01180010132000000000c7		1	0	0	0	24	0	16
24401	0004A30B00288390	2019-04-29 19:02:35	535	3	-77	SF10 BW125 4/5	01170011130200000000c7		1	0	0	0	23	0	17
24387	0004A30B00288390	2019-04-29 18:32:33	534	3	-78	SF10 BW125 4/5	01160011122000000000c7		1	0	0	0	22	0	17
24375	0004A30B00288390	2019-04-29 18:02:30	533	3	-76	SF10 BW125 4/5	01150012120200000000c7		1	0	0	0	21	0	18
24361	0004A30B00288390	2019-04-29 17:32:27	532	3	-77	SF10 BW125 4/5	01140012112000000000c7		1	0	0	0	20	0	18
24349	0004A30B00288390	2019-04-29 17:02:25	531	3	-70	SF10 BW125 4/5	01130012110200000000c7		1	0	0	0	19	0	18
24335	0004A30B00288390	2019-04-29 16:32:22	530	3	-78	SF10 BW125 4/5	01120012102000000000c7		1	0	0	0	18	0	18
24316	0004A30B00288390	2019-04-29 16:02:20	529	3	-77	SF10 BW125 4/5	00110011100200000000000		0	0	0	0	17	0	17
24315	0004A30B00288390	2019-04-29 15:57:16	528	3	-95	SF10 BW125 4/5	871000110f39012c27e8c7	060f3930	7	1	0	0	16	0	17
24314	0004A30B00288390	2019-04-29 15:57:10	527	3	-89	SF10 BW125 4/5	870f00110f39012c27e8c7	060f3930	7	1	0	0	15	0	17
24300	0004A30B00288390	2019-04-29 15:40:03	526	3	-93	SF10 BW125 4/5	810e00110f2701242860c7		1	1	0	0	14	0	17

Figure : Database table example

Once data is stored on the database, you can use any compatible tool to query the database and develop your own data analysis and visualization. This task belongs entirely to the customer or system integrator.



Last packet received

node_name ▲	LW_EUI	Time	Frame_type	State
node_01	0004A30B00288390	2019-04-30 10:57:10	Keep alive	OCCUPIED
node_02	0004A30B00286556	2019-04-30 11:00:19	Keep alive	OCCUPIED
node_03	0004A30B0028A33A	2019-04-30 11:07:40	Keep alive	OCCUPIED
node_04	0004A30B00287F04	2019-04-30 11:00:01	Keep alive	OCCUPIED
node_05	0004A30B00287F01	2019-04-30 10:56:10	Keep alive	FREE
node_06	0004A30B0028600B	2019-04-30 11:20:00	Keep alive	FREE
node_07	0004A30B002612F1	2019-04-30 11:10:41	Keep alive	OCCUPIED
node_08	0004A30B0026425C	2019-04-30 11:02:57	Keep alive	OCCUPIED
node_09	0004A30B00268D81	2019-04-30 11:04:18	Keep alive	OCCUPIED
node_10	0004A30B0025EE61	2019-04-30 10:55:43	Keep alive	OCCUPIED
node_11	0004A30B002680BD	2019-04-30 11:01:36	Keep alive	OCCUPIED

Figure : Example of database visualization using Grafana (basic open-source application)

7.7. Modifying the Customer Server

Maybe you want to modify the Customer Server to match your requirements, this section describes the best way to do it.

7.7.1. Modifying the Database

If you want to use a database which is different to the provided one, or if you want to change the way data is saved on the database you must modify the file src/app/Helpers/DataFlow.php.

This class has 3 methods:

- **get_null_record**: This method should return an array of **NULL** values foreach field defined on the database.
- **get_last_record**: This method should return the previous record for the device. It should call the previous method if there isn't any previous value for the node.
- **insert_record**: This method receives the data prepared to be inserted on the database as input and therefore it makes the database insertion on a new row.

7.7.2. Adding a new unsupported LoRaWAN Network Server

To add a new LoRaWAN Network Server (different from Loriot and Actility) you will have to create a new PHP class that must implement the interface located on **src/app/Helpers/Services/LwServiceInterface.php**. There are 4 methods to be implemented:

- **enqueue_message**: It allows to send data to the new LoRaWAN Network Server.
- **get_lw_info_from_request**: It returns all the LoRaWAN information from the request of the new LoRaWAN Network Server.
- **get_eui_from_request**: It returns the node EUI from the request of the new LoRaWAN Network Server.
- **get_payload_from_request**: It returns the payload from the request of the new LoRaWAN Network Server.

The next step would be to create an endpoint to be able to receive data from the LoRaWAN Network Server. Therefore, you have to create a new method on the services app's controller:

```
src/app/Http/Controllers/api/v2/ServiceController.php
```

This method will receive a request object with all the data provided from the service, you can use the helper methods from the service class you have created previously. The implementation details can vary from service to service, so this cannot be explained in detail (you can use the Loriot and Actility ones as a reference).

The functionality that this method must implement is:

- Get data from the request
- Parse the payload using the same helper class used by the provided services:

```
$a_result = $this->frames->main($EUI, $payload);
```

- Enqueue the data if the parser returns something on the **output** key returned form the parser:

```
if($a_result['output']){
  YourService::enqueue_message($EUI, $a_result['output']['data']);
}
```

- Optionally, trigger an event to save the transaction to the database:

```
event(new \App\Events\DecoderEvent($a_result));
```

Once all steps have been completed you have to register the service on the file:

```
src/app/Providers/AppServiceProvider.php
```

The register method should pair LwService with the service you have just created.

8. Libelium Smart Parking Cloud Service

You can use the Libelium Cloud service to configure your nodes and to receive data from them. Using the Libelium Cloud Service you do not need any infrastructure on your side, you just need to purchase a Smart Parking Cloud Service license.

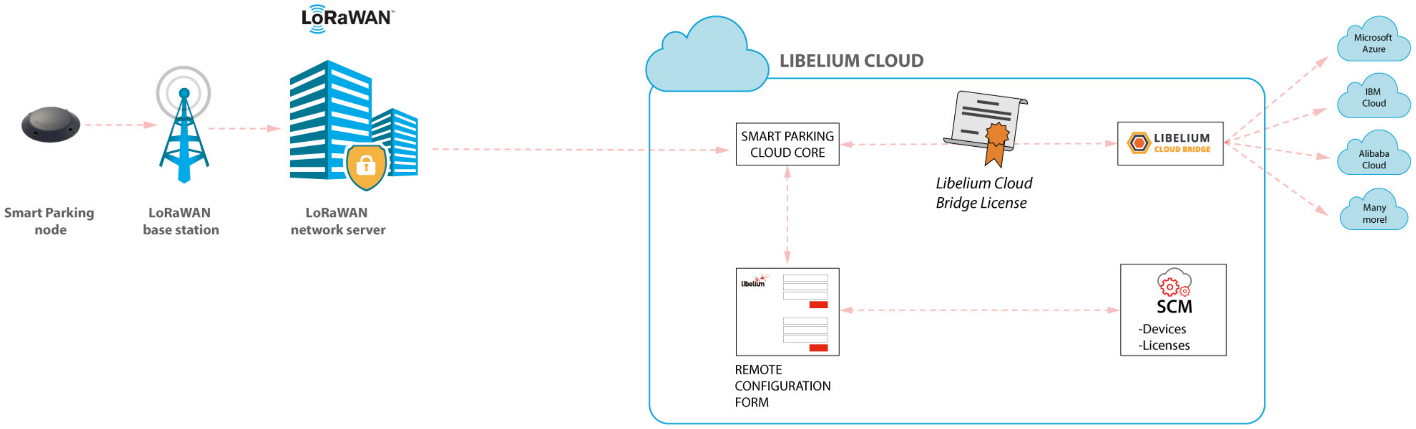


Figure : Smart Parking Cloud Service

The Smart Parking Cloud Service includes the Remote Configuration Form to manage the configuration values of the Smart Parking nodes, and the Smart Parking Cloud Core, that deals with the LoRaWAN network server requests, sending back a response if needed. The data sent from the Smart Parking nodes will be sent to any compatible final IoT cloud service using the Libelium Cloud Bridge (you will need a license to use the Libelium Cloud Bridge).

8.1. Smart Parking Cloud Service

Users can access this service clicking on the “Smart Parking” option through the left-side menu under the “Services” section. A valid licence is compulsory in order to work with this service.

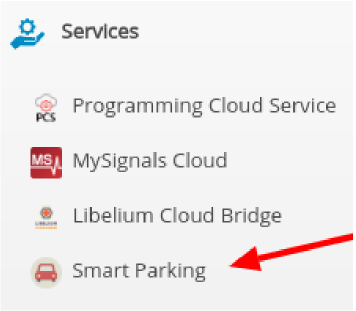


Figure : Accessing Smart Parking Cloud Service (1st way)

Another way to access the Smart Parking Cloud Service is possible through the collapsed menu at the top right of the page. Clicking on it and then on the “Smart Parking” option, users will arrive to the same place. As in the previous case, users must possess a valid license. Otherwise, this option will not be displayed.

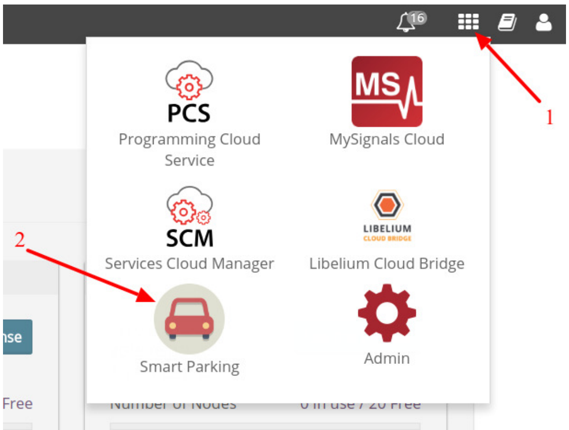
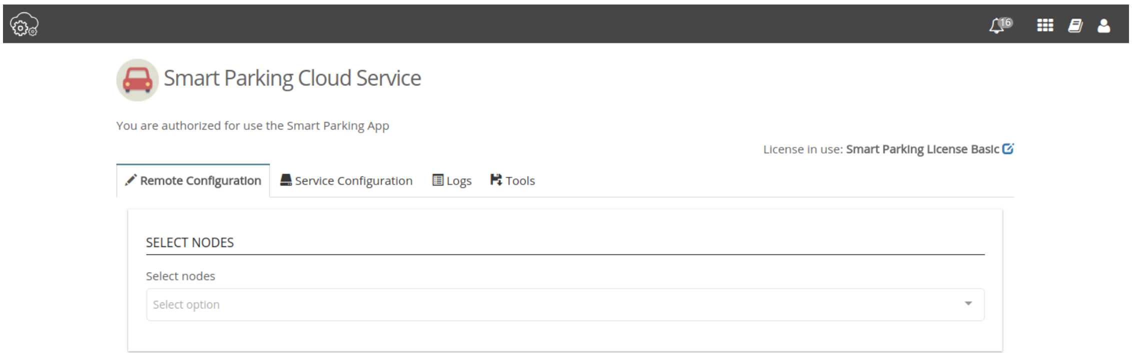


Figure : Accessing the Smart Parking Cloud Service (2nd way)

Once the user has a valid license, the following features will be available accessing the Smart Parking Cloud Service: configure nodes remotely, configure the LoRaWAN service, check the logs for the configured service and generate the tokens and keys needed for the LoRaWAN network server.


Smart Parking Cloud Service

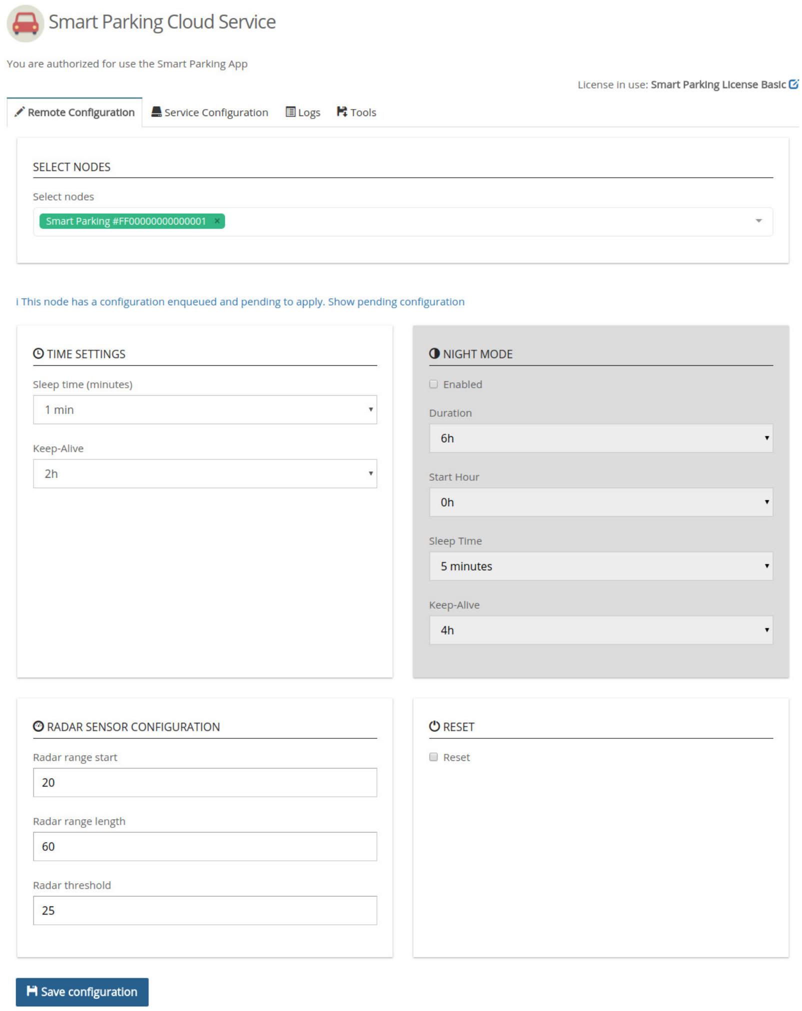
You are authorized for use the Smart Parking App

License in use: Smart Parking License Basic

Figure : Smart Parking Cloud Service landing page

8.1.1. Remote Configuration

This tab shows a drop-down menu to select one or several devices. Each one is identified by their unique LoRaWAN EUI. The nodes displayed in the drop-down menu correspond to the nodes you have claimed for on your Services Cloud Manager account. You can also deselect previously selected nodes using the cross in the node's identification. Once a node is selected, the current configuration of this node is displayed and you can modify it. For further information of the displayed node parameters, please refer to the chapter "Smart Parking node".



Smart Parking Cloud Service

You are authorized for use the Smart Parking App

License in use: **Smart Parking License Basic**

Remote Configuration | Service Configuration | Logs | Tools

SELECT NODES

Select nodes: Smart Parking #FF00000000000001 ×

i This node has a configuration enqueued and pending to apply. Show pending configuration

TIME SETTINGS

Sleep time (minutes): 1 min

Keep-Alive: 2h

NIGHT MODE

□ Enabled

Duration: 6h

Start Hour: 0h

Sleep Time: 5 minutes

Keep-Alive: 4h

RADAR SENSOR CONFIGURATION

Radar range start: 20

Radar range length: 60

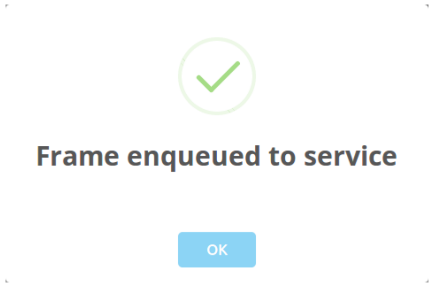
Radar threshold: 25

RESET

□ Reset

Figure : Remote Configuration Form

Once you have configured the node(s) and the service as explained in the section “Service Configuration”, press the “Save configuration” button. A message indicates that a downlink frame with the node’s configuration has been enqueued to the LoRaWAN Network Server downlink queue. The node will receive a downlink frame with the new configuration in response to the next Info or Keep-Alive uplink frame.



Frame enqueued to service

Figure : Frame enqueued for processing

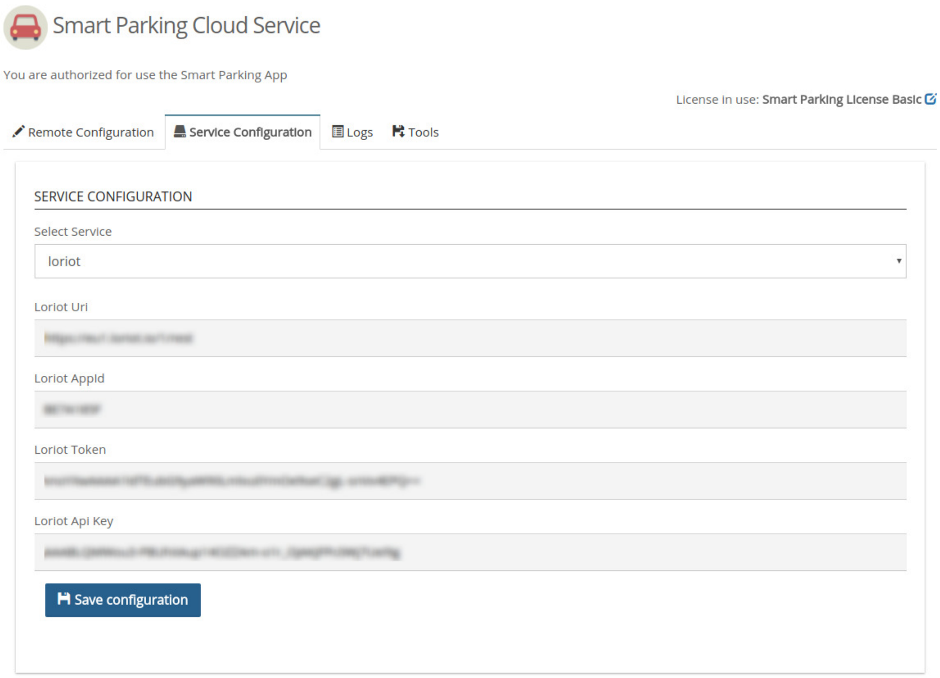
The Remote Configuration Form application allows the user to update some node parameters just as the Smart Devices App does. However, using the Smart Devices App requires physical access to the node: it must be plugged to your computer via a USB cable.

Note: The Remote Configuration Form does not permit to update any LoRaWAN parameter. The Smart Devices App should be used instead.

8.1.2. Service Configuration

The Smart Parking Cloud Service is compatible with Loriot and Actility, but only one of these can be active at the same time.

If you want to use Loriot as your LoRaWAN Network Server, you have to select it on the drop-down menu and complete the required fields. To find the information on the Loriot panel, you can follow the instructions described on the "How to configure Loriot" section.



Smart Parking Cloud Service

You are authorized for use the Smart Parking App

License in use: **Smart Parking License Basic**

SERVICE CONFIGURATION

Select Service: loriot

Loriot Url: [redacted]

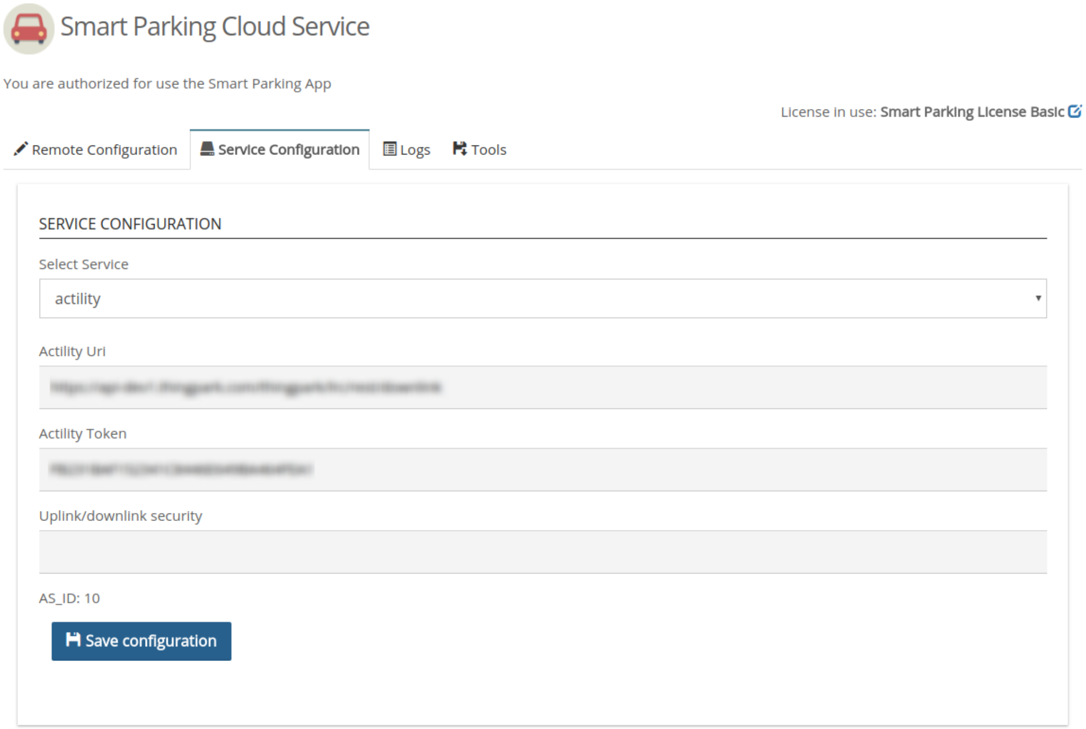
Loriot AppId: [redacted]

Loriot Token: [redacted]

Loriot Api Key: [redacted]

Figure : Loriot configuration

If you want to use Actility as your LoRaWAN Network Server, you have to select it on the drop-down menu and complete the required fields. To find the information on the Actility panel, you can follow the instructions described on the "How to configure Actility" section.



Smart Parking Cloud Service

You are authorized for use the Smart Parking App

License in use: **Smart Parking License Basic**

SERVICE CONFIGURATION

Select Service

actility

Actility Uri

[redacted]

Actility Token

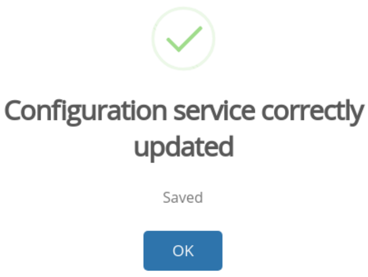
[redacted]

Uplink/downlink security

AS_ID: 10

Figure : Actility configuration

Regardless of the service you have configured, click on the button “Save Configuration” to save your LoRaWAN service configuration. Then, a message will be displayed indicating that the service has been updated or saved correctly.



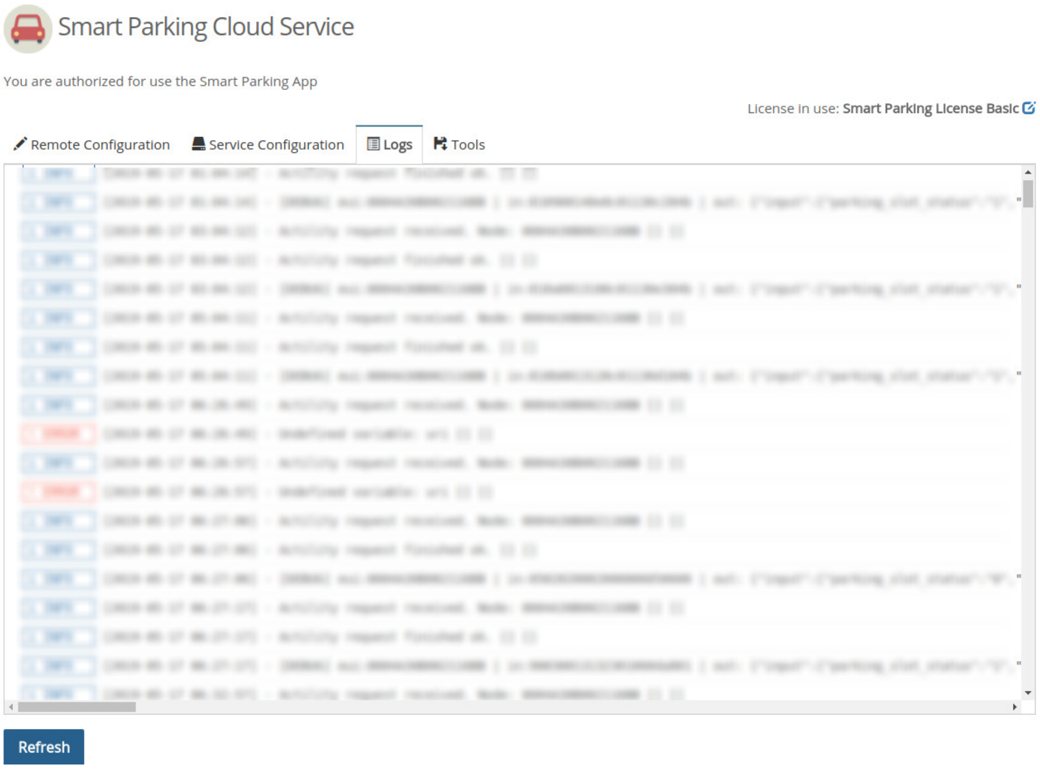
Configuration service correctly updated

Saved

Figure : Service configuration saved

8.1.3. Log

The “Logs” tab shows the information and error messages registering the result of the communication requests of the LoRaWAN services. Different communication protocols are used to configure the service. All messages related to the transport layer of the service are identified as INFO or ERROR. User intervention may be required to analyse the meaning of the messages generated in the application layer and decide if some actions are required.



Smart Parking Cloud Service

You are authorized for use the Smart Parking App

License in use: **Smart Parking License Basic**

Figure : Log viewer

Click the “Refresh” button to update the log view and load new messages available. The latest messages appear at the bottom of the viewer.

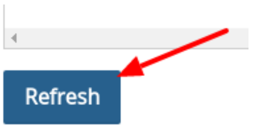
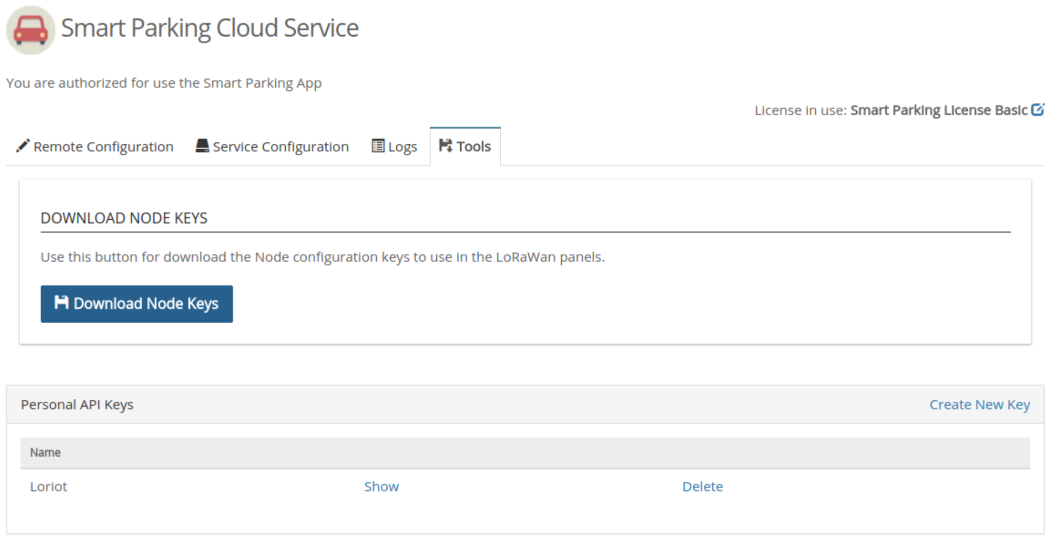


Figure : “Refresh” button

8.1.4. Tools

The “Tools” tab allows users to download the Device Node Keys associated with their account, as well as generate the necessary token to configure your Loriot network server correctly.



You are authorized for use the Smart Parking App

License in use: Smart Parking License Basic

DOWNLOAD NODE KEYS

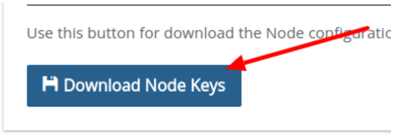
Use this button for download the Node configuration keys to use in the LoRaWan panels.

Personal API Keys

Name		
Loriot	Show	Delete

Figure : Tools

The Node Keys is a CSV-format file. It is ready to use in the LoRaWAN Network Server to configure all the Smart Parking nodes associated with their account in the Libelium Cloud. It can be downloaded by clicking on the "Download Nodes Keys" button.



Use this button for download the Node configuratic

Figure : "Download Nodes Keys" button

Finally, a token must be generated clicking on the "Create New Key" link in order to finish the Loriot service configuration.

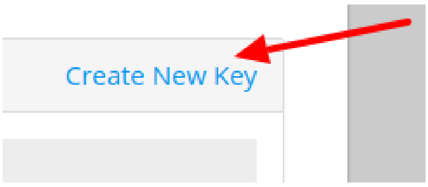
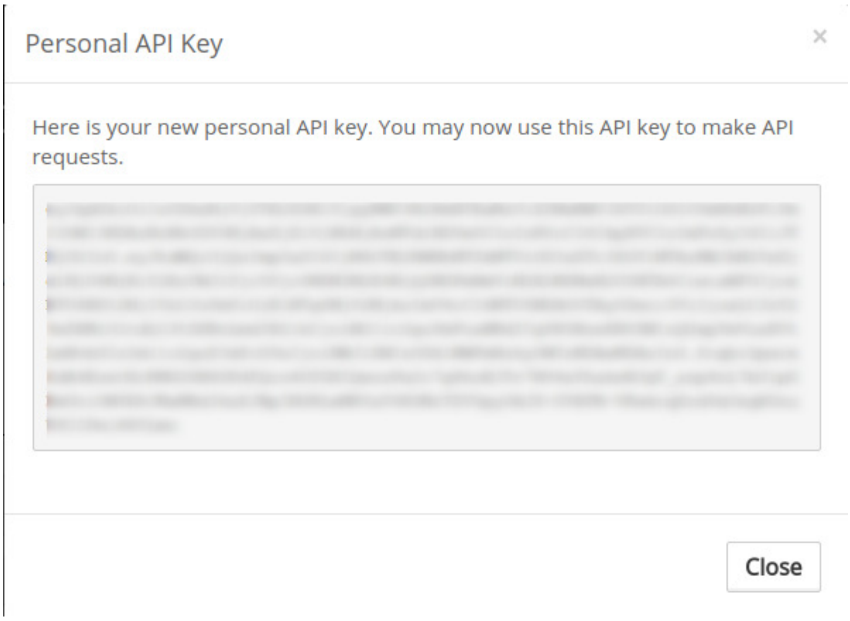


Figure : "Create New Key" link

A pop-up window will be displayed, in which users must introduce a name to identify the new token. Then, clicking the "Create" button shows a password in a new window, ready to be used in the Loriot service.



Personal API Key

Here is your new personal API key. You may now use this API key to make API requests.

[redacted]

Figure : Generated key

Once the token has been generated, you can use it on the Loriot backend as described in the section "Manage Loriot output section.

9. Deployment and installation

9.1. Step-by-step guideline

Once you receive the Smart Parking batch, the steps involved in the deployment of the network are:

1. **Nodes provisioning**: Once the client owns a LoRaWAN Network Server license, it is mandatory to register the nodes with the configured identifier and keys for each node. The client must decide whether they use the factory default LoRaWAN unique identifier and random private keys provided by Libelium or they prefer to configure their own identifier and keys. You must keep in mind that every node is provided "ready to install" with factory default identifier and keys. If client-configured identifier and keys are desired, then each node will have to be configured separately using the Smart Devices App. This 2nd option increases the provisioning time. So Libelium recommends to use the default IDs and keys. If factory default keys are used, then a CSV file exported from Libelium Cloud Services Manager (SCM) can be used to ease the bulk import into the LoRaWAN Network Server.
   Please refer to the “Smart Devices App ” chapter for further information about the Smart Devices App.
   Please refer to the “Libelium Cloud management” chapter for further information about how to extract the factory default IDs/keys.

2. **Libelium Cloud or Customer Server setup**: The client must have chosen the type of cloud service to be used: The Libelium Cloud or her own Customer Server. The Libelium Cloud is used to receive and forward data to a 3rd party cloud system and the Customer Server is used to receive and store data. So depending on the end-point used, the LoRaWAN Network Server must be configured to properly forward all packets to the desired server.

3. **Nodes installation**: Each node must be placed on the proper parking slot and then follow the "magnet start-up" protocol explained in the “"Magnet start-up" process” section.

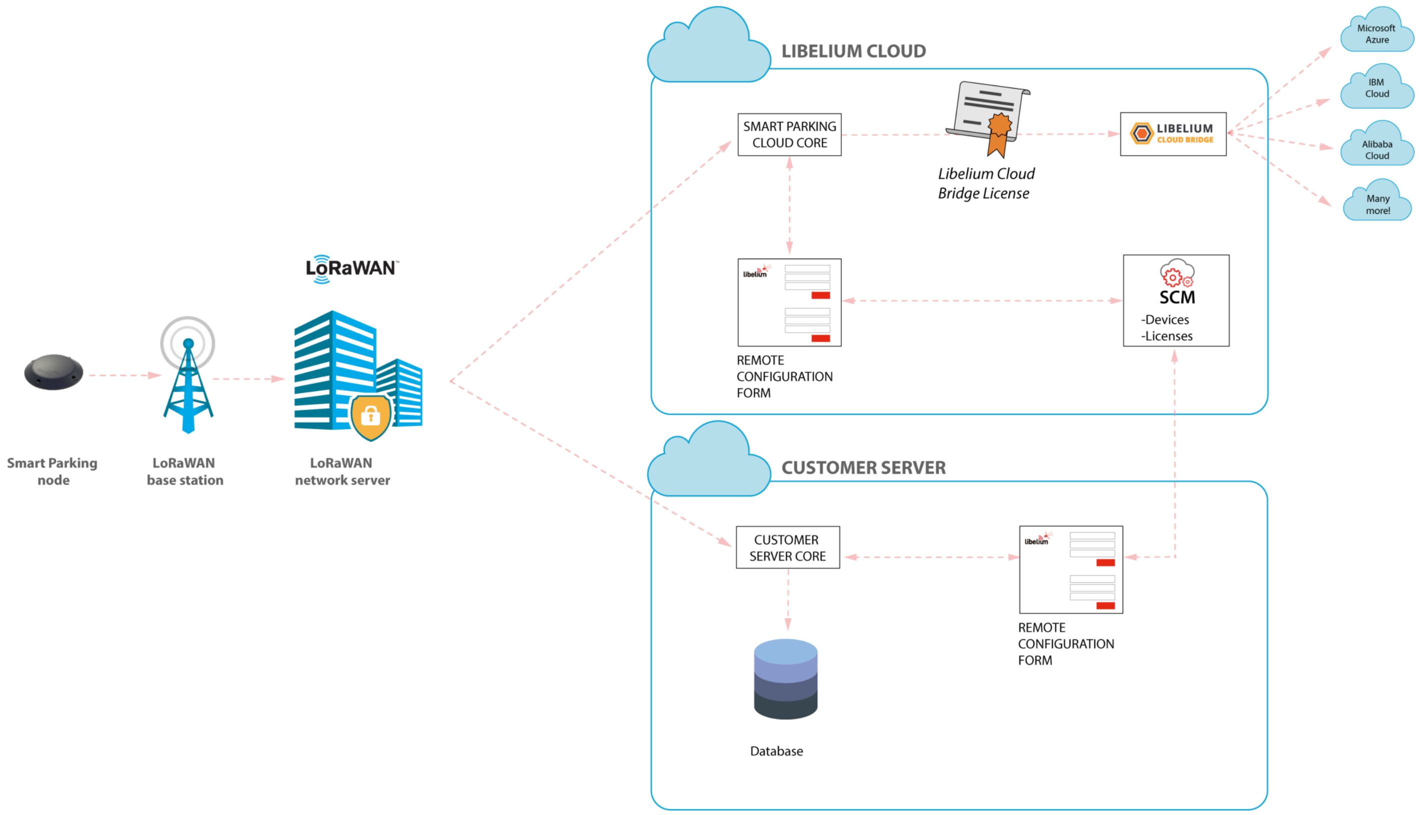


Figure : Smart Parking network architecture

9.2. How to place the nodes

The correct installation spot is the center of the parking slot to ensure the best detection. It is possible to install the nodes on different parking slot configurations:

- Parallel
- Perpendicular
- Angled

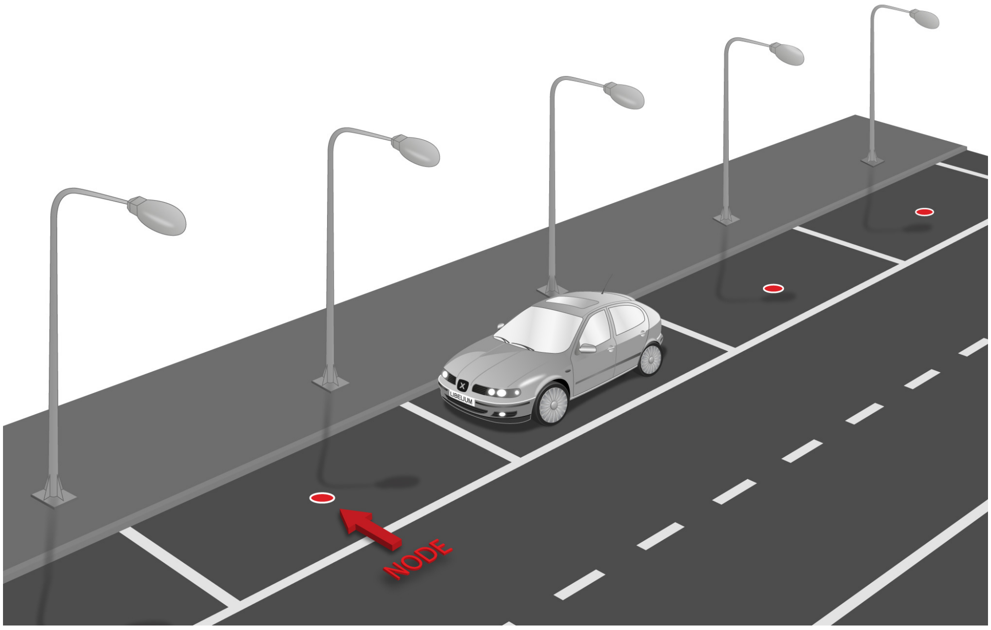


Figure : Parallel parking slots

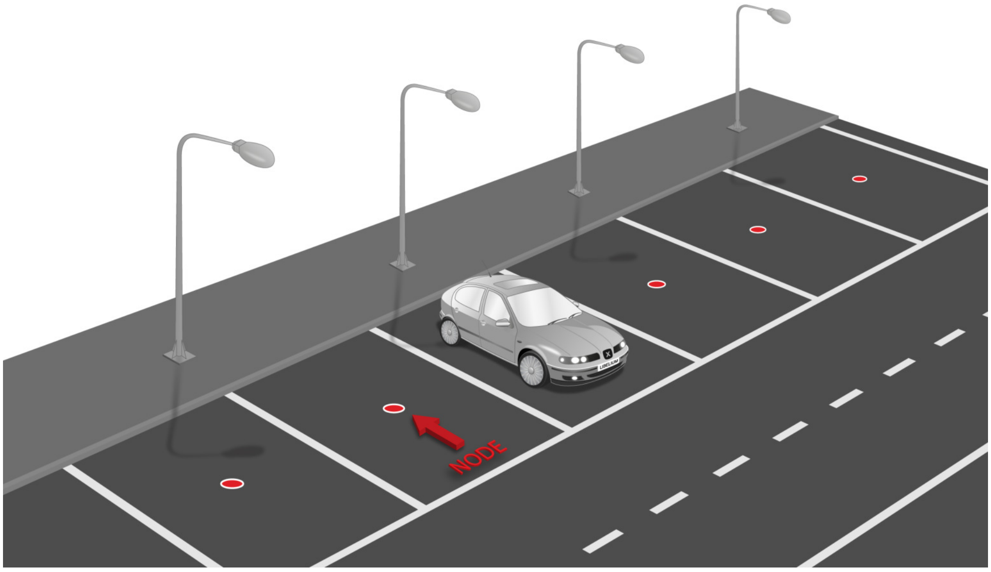


Figure : Perpendicular parking slots

Important:
It is important to keep in mind that parking slots must be delimited by lines so the node is placed in the center of the parking area. Placing the nodes with no line definitions may lead into bad behaviour.

9.3. Node installation

9.3.1. Triple installation option

The user can choose among the 3 types of installation for the node. This table summarizes pros and cons:

	On-surface installation	**Semi-underground installation**	**Underground installation**
Type of enclosure	Vaulted	Vaulted	Flat
Presence of node over the ground	Full node (40.25 mm)	Part of node (6 mm)	Node completely buried (0 cm)
Rain immunity	Yes	Yes	Partial
Puddle immunity	Yes	Yes	Detection may fail if puddle appears
Vandalism/robbery immunity	Medium	High	High
Snowplow immunity	No	Possible	Yes
Installation process	Fast and easy	Special tools needed	Special tools needed
Tools needed	Normal drill	Industrial big crown drill. Epoxy.	Industrial big crown drill. Epoxy.
Can be dismantled for maintenance	Yes	No	No

9.3.2. On-surface node installation

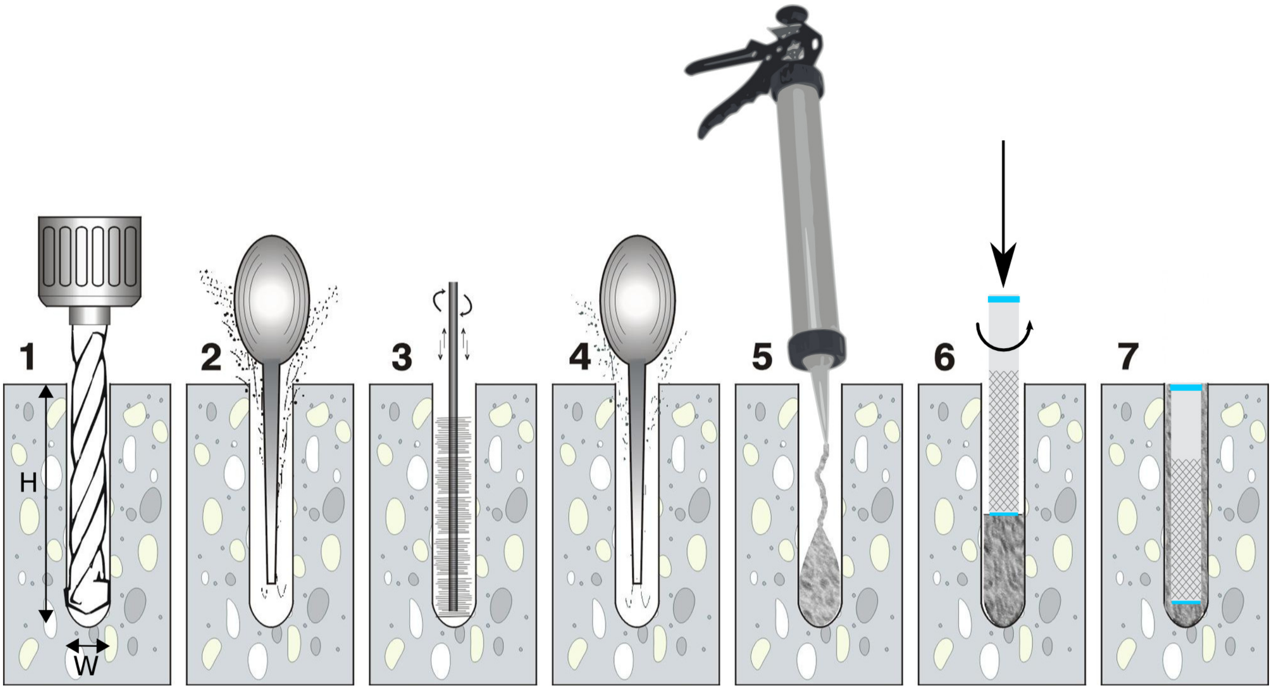


Figure : Step by step chemical fixing and anchor installation

Firstly, mark 4 holes in the asphalt. Select an area as flat and regular as possible, avoid irregularities on the terrain. You should place the enclosure in the final location and use the 4 holes as a reference to drill 4 little marks on the ground. You can also draw 4 dots with ink.

Figure : Marking the holes for the node

Then continue with the chemical fixing installation step by step as described above. Also, in the next video you can see how the installation is done chemical fixing installation.

Step 1: Drill the holes.
The drill diameter must be W=14 mm minimum. Try to drill as vertical as possible to enable the best enclosure installation. The length of the hole must be about H=85 mm.

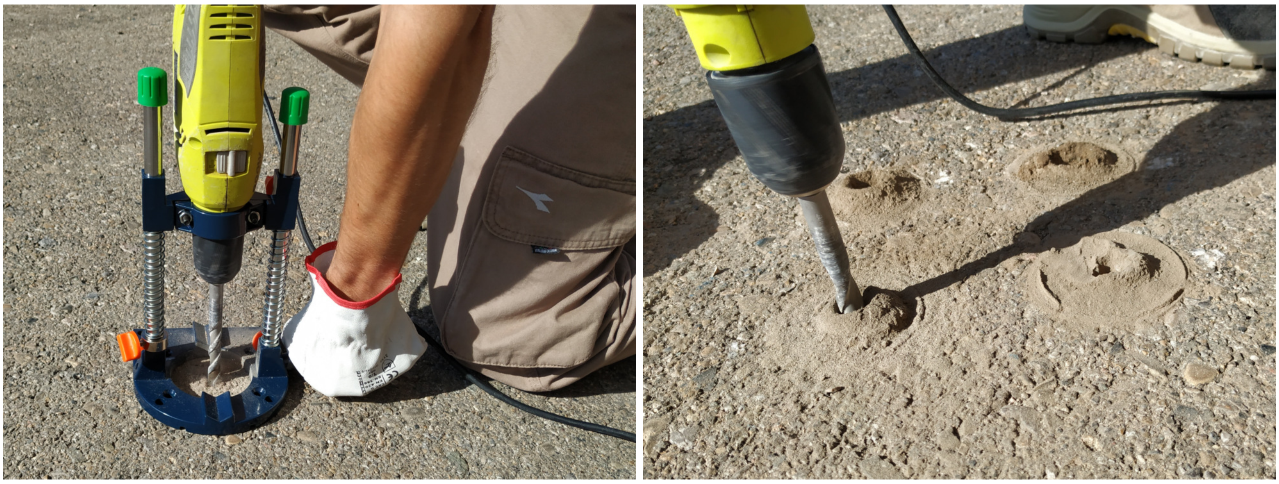


Figure : Drill the holes (step 1)

Step 2, 3 and 4: Blow out dust and loose materials.
Remove all the dust and little rocks inside the 4 holes, created during the drilling operation. This is important to allow the anchors go all the way down. You can clean each hole mechanically and then use a bottle of spray duster (high-pressure air bottle). Make sure that the real depth of the hole is enough.

Figure : Brush the hole (step 3)

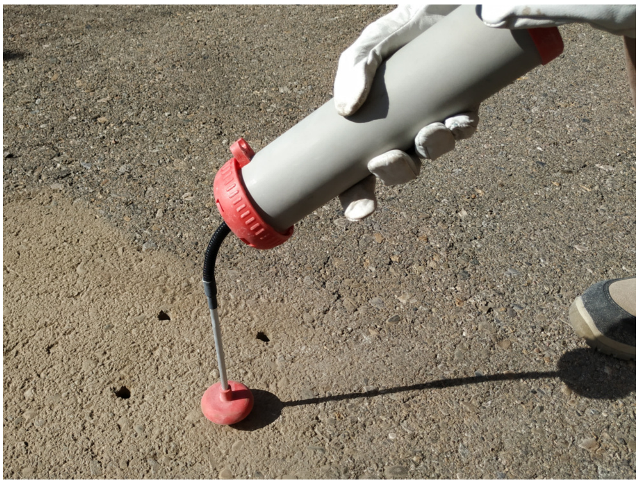


Figure : Blow out dust and loose materials (steps 2 and 4)

Step 5: Inject the chemical fixing.
The chemical fixing shall be injected into the 4 holes carefully. You must keep in mind that this chemical adhesive must fill the complementary space the anchor leaves in the hole. So the whole hole does not have to be fully filled with adhesive because there is no space left for the anchor. The chemical adhesive quantity to be injected will depend on the dimensions of the holes.

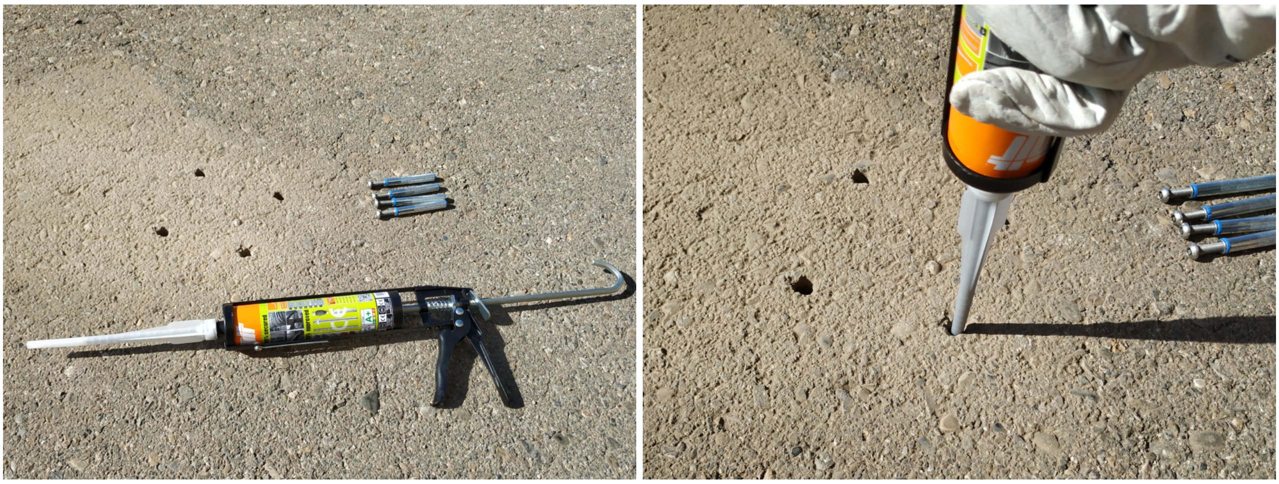


Figure : Inject chemical fixing adhesive (step 5)

Step 6: Insert the anchors.
Once the adhesive is injected, the anchor must be carefully inserted into the hole. Rotating the anchor by hand while inserting it is a good practice. Therefore, the adhesive fills and completes all the empty space left by the anchor in the hole.

Once the anchor is correctly inserted, some adhesive might exceed from the hole. This residuary adhesive must be removed from the installation so the are no extra adhesive and the anchor installation is clean and smooth.

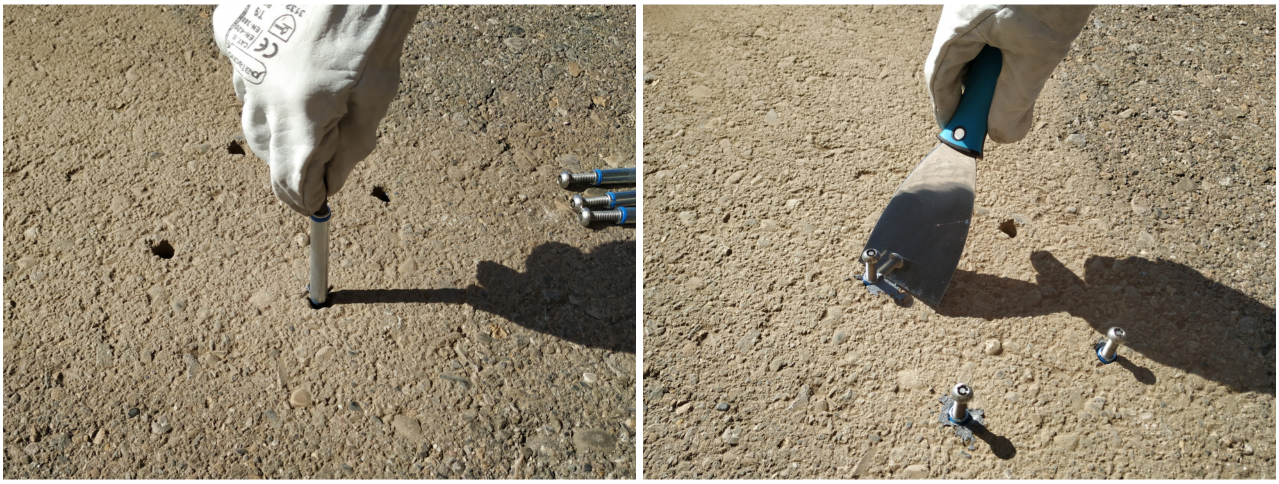


Figure : Insert the anchors (step 6)

Step 7: Finish anchors installation.
The anchors have an internal thread to screw the screws. Inserting the anchors with the screws partially inserted might be a good practice in order to place the anchor correctly aligned within the hole (that means you can use the screw as a tool if it is partially inserted inside the anchor). Also, if the screw is inside the anchor, avoids that the adhesive accidentally gets inside the anchor. After cleaning the remaining adhesive from the hole, the user must wait some time until the chemical resin is solid and stable. After this period of time, the screws can be removed from the anchors and the installation is finished.

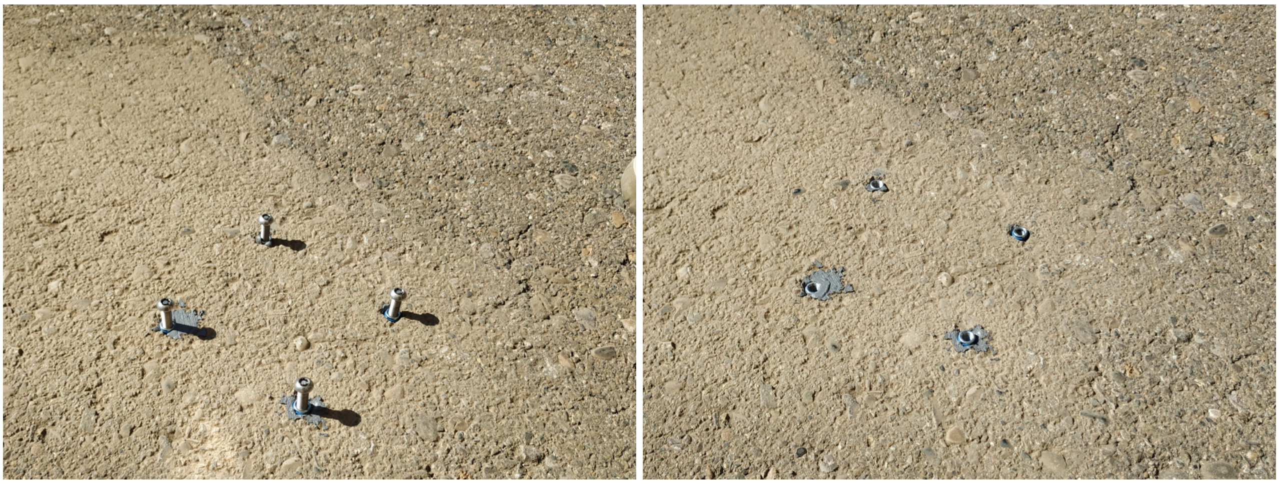


Figure : Final anchor installation (step 7)

Step 8: Place the node.

Place the black rubber circle on the ground. To enable the correct screwing of the screws inside the anchors, make sure that there is no dust in the internal thread of the anchor. Install the node in its final position and screw the 4 special (anti-vandalism) screws provided. Anti-vandalism screws are recommended to avoid problems: anyone could unscrew a node if fixed with standard screws.

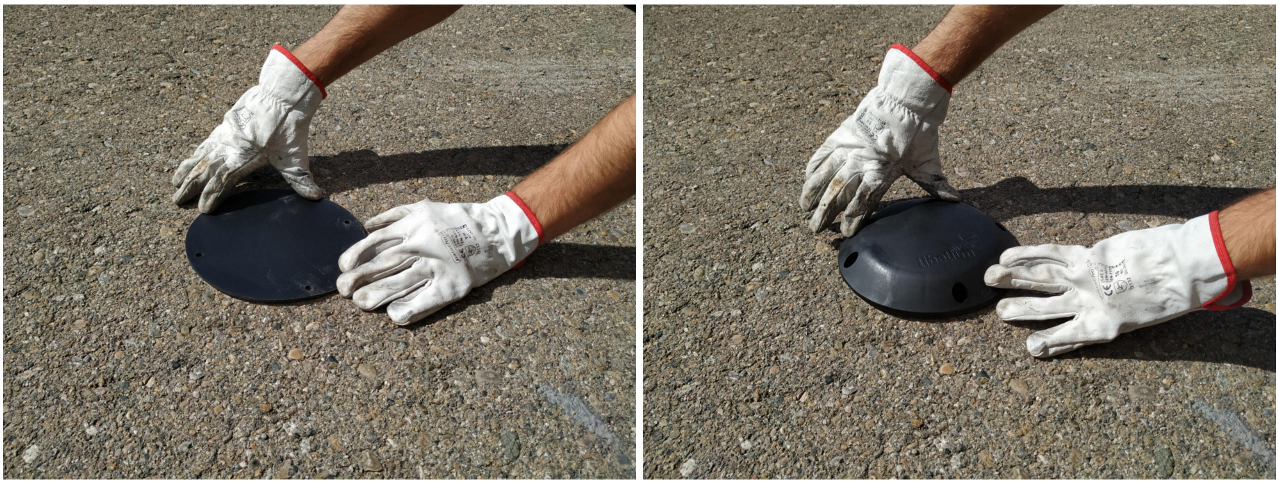


Figure : Place the circle rubber and the node above it

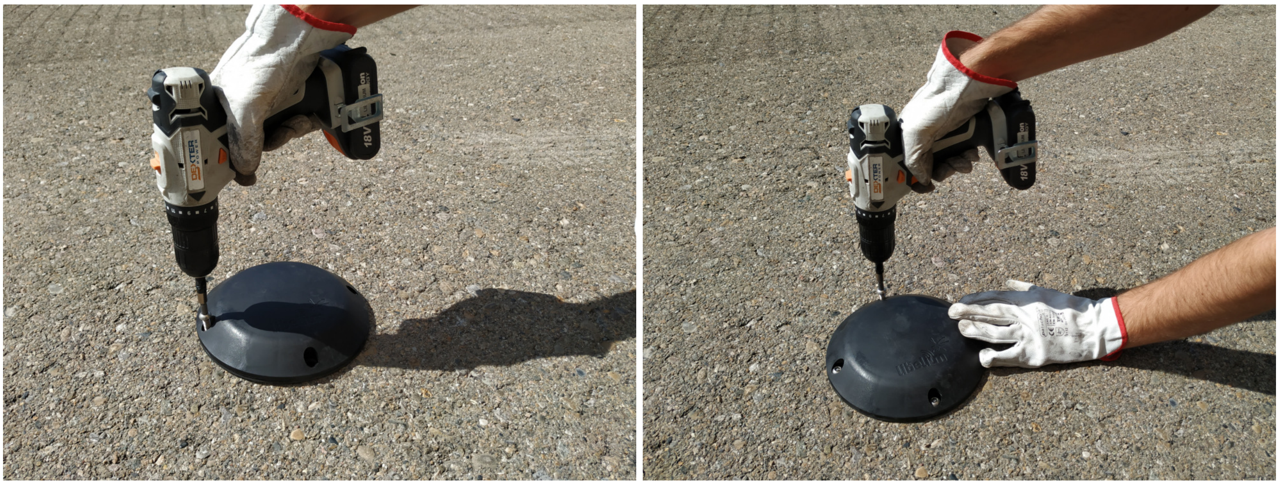


Figure : Screw the node

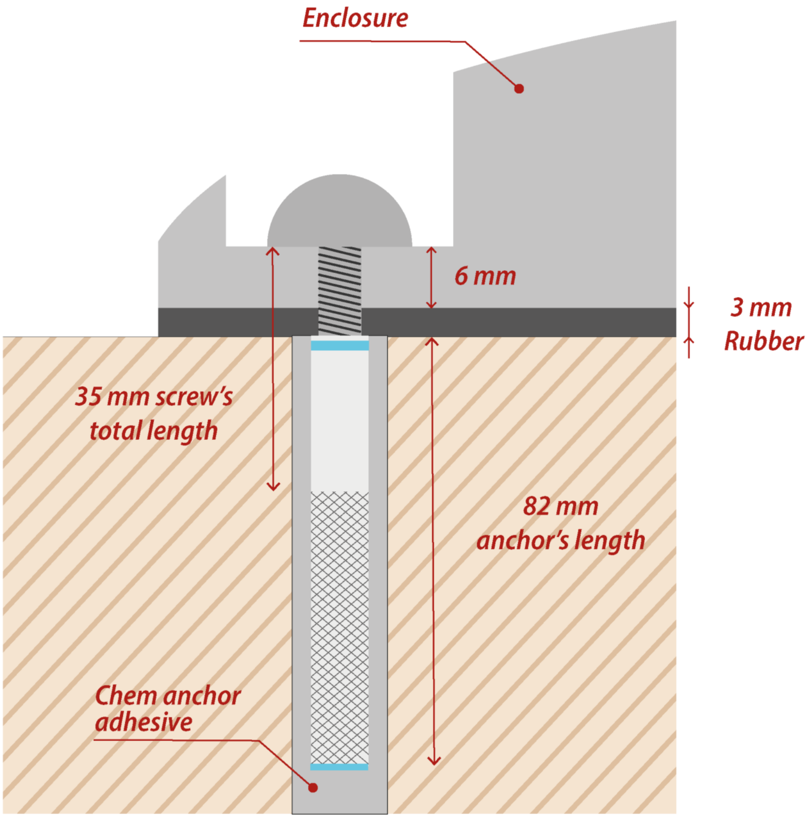


Figure : Final section of the anchor and screw

9.3.3. Underground node installation

Depending on the scenario, placing the node under the pavement can be a must. In cold places where snow is usual, installing the nodes underground permits the use of snowplow equipment without affecting the Smart Parking node. Also, this kind of installation allows the client to deploy much safer networks against vandalism or theft as the device is not accessible directly from the outside.

Figure : Snowplow vehicle

For this special installation, Libelium provides flat enclosures. Contact your Sales agent to order flat enclosures.

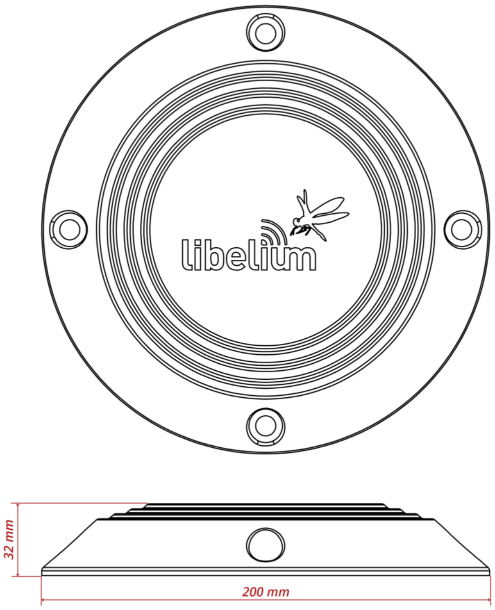


Figure : Flat Smart Parking enclosure dimensions

Regarding the installation process, the next steps explain how to correctly install the device under the ground:

Step 1: Drill the hole.
Use a hole saw or other dedicated tool to drill a hole into the pavement bigger enough for the enclosure (minimum diameter 202 mm). Try to perforate it deep enough to insert the Smart Parking node completely without rising over the surface, be sure that the enclosure remains under the pavement line approximately 10 mm.

Figure : Making a hole in the pavement

Step 2: Place the node into the pavement hole.
Below the device, a layer of cement must be spread to elevate the device to the correct horizontal position independently of the depth of the hole. The diameter of the hole must be at least 202 mm.

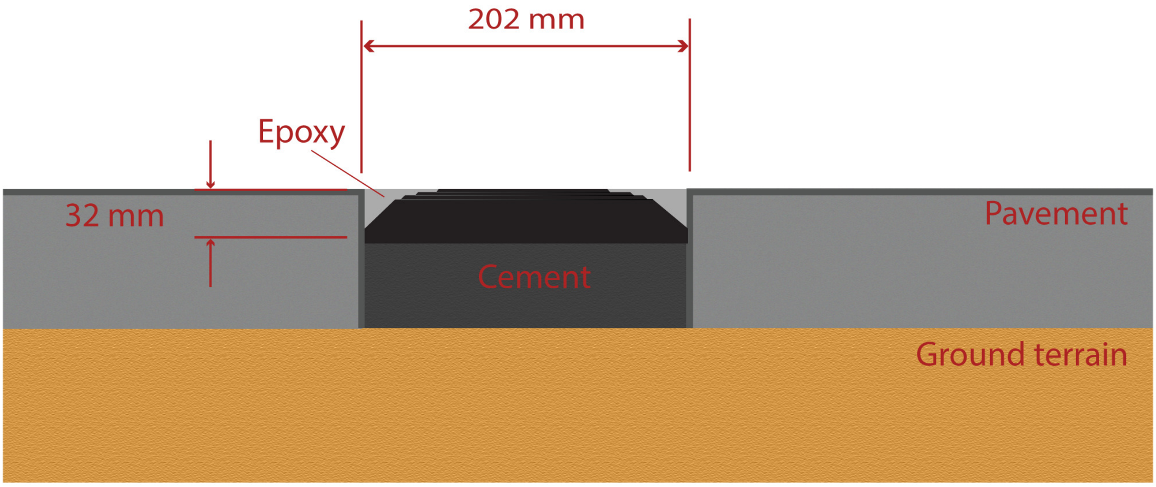


Figure : Placing the node in the hole

Step 3: Fill the hole.
Finally, the node needs to be fixed to the pavement using Epoxy resin (commonly used for filling road fissures). This resin should fill all the space between enclosure and pavement. Also, it should completely cover the top of the enclosure ensuring protection against water, bad environmental conditions and vandalism. The Epoxy resin tested by Libelium provides a curing time around 6-12 hours at 20 ºC and RH 50%. So the parking installation must not be used prior this curing time.

Figure : Pour Epoxy curing agent into the granular resinous material

Figure : Preparing the Epoxy resin mixture

In this installation, it is crucial that the upper side of the enclosure is not covered with Epoxy or other material. Only the side slides can be covered, the center of the node must be clean to allow the radar signal work properly.

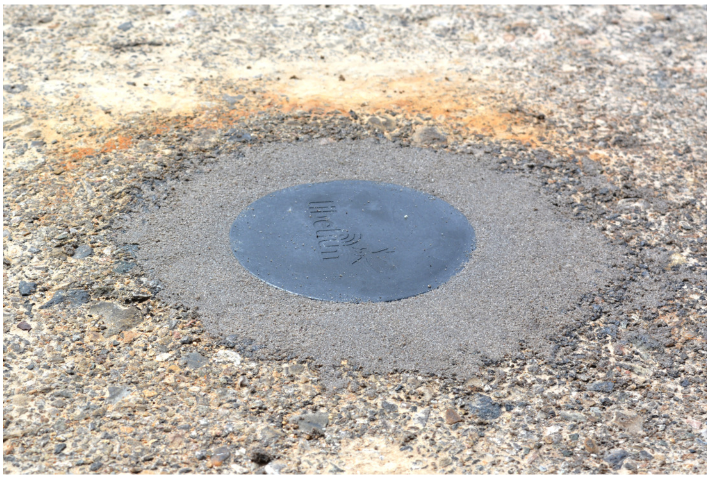


Figure : Final Epoxy application

Step 4: Start the node using the magnet.
Proceed with the "Magnet start-up" process in order to initialize the node.

9.3.4. Semi-underground node installation

This is a possible installation in order to provide vandalism immunity and also rain/puddle immunity. So the node partially inserted in the ground and the vaulted enclosure minimizes the water issues.

The same underground installation steps must be followed as seen in the previous section. In the node placement step, the vaulted enclosure must exceed the horizontal pavement layer by 6-7 mm approximately. The next figure shows how the node is placed:

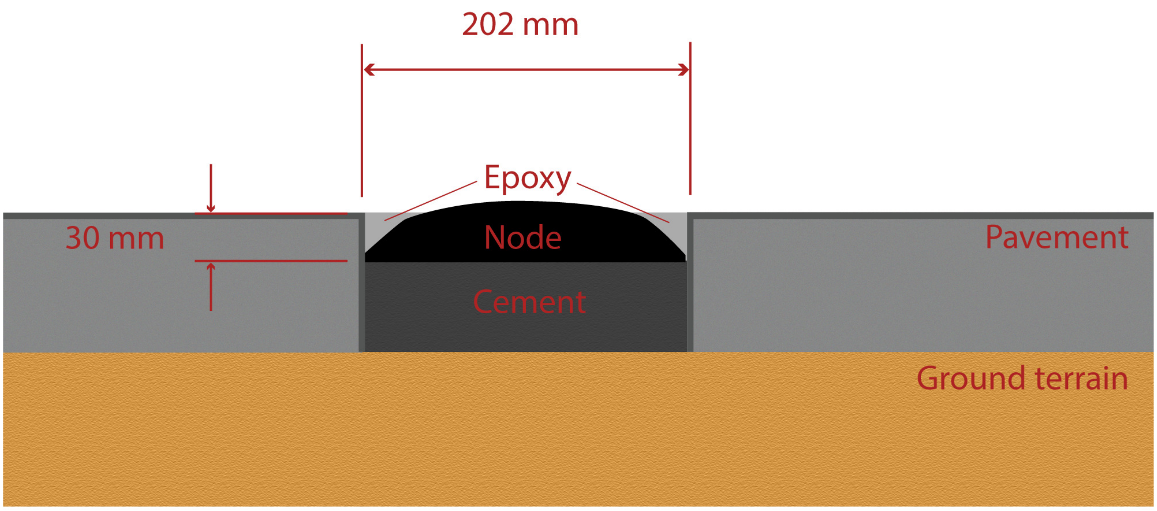


Figure : Semi-underground node installation

9.4. Node start-up

After the node is installed, you must follow the "Magnet start-up" steps explained in the "Magnet start-up process" section. This process is based on 3 consecutive magnet resets waiting at least one second between each reset. In the future need of resetting the node manually, just a single magnet reset will be needed.

Once the node has been started, it will begin to run the firmware step-by-step as explained in the “How the node works” section.

In the next video you can see how the "magnet start-up" is performed: magnet start-up.

Figure : Using the magnet to start the node

9.5. Working example

The next diagram explains how the node program works when the node is installed and started when the parking slot is free. On the left column you can appreciate the parking lot status during the execution. On the right side you can see the frame types that are sent to the Libelium Cloud or Customer Server depending on: parking slot status changes, elapsed time and user configuration changes.

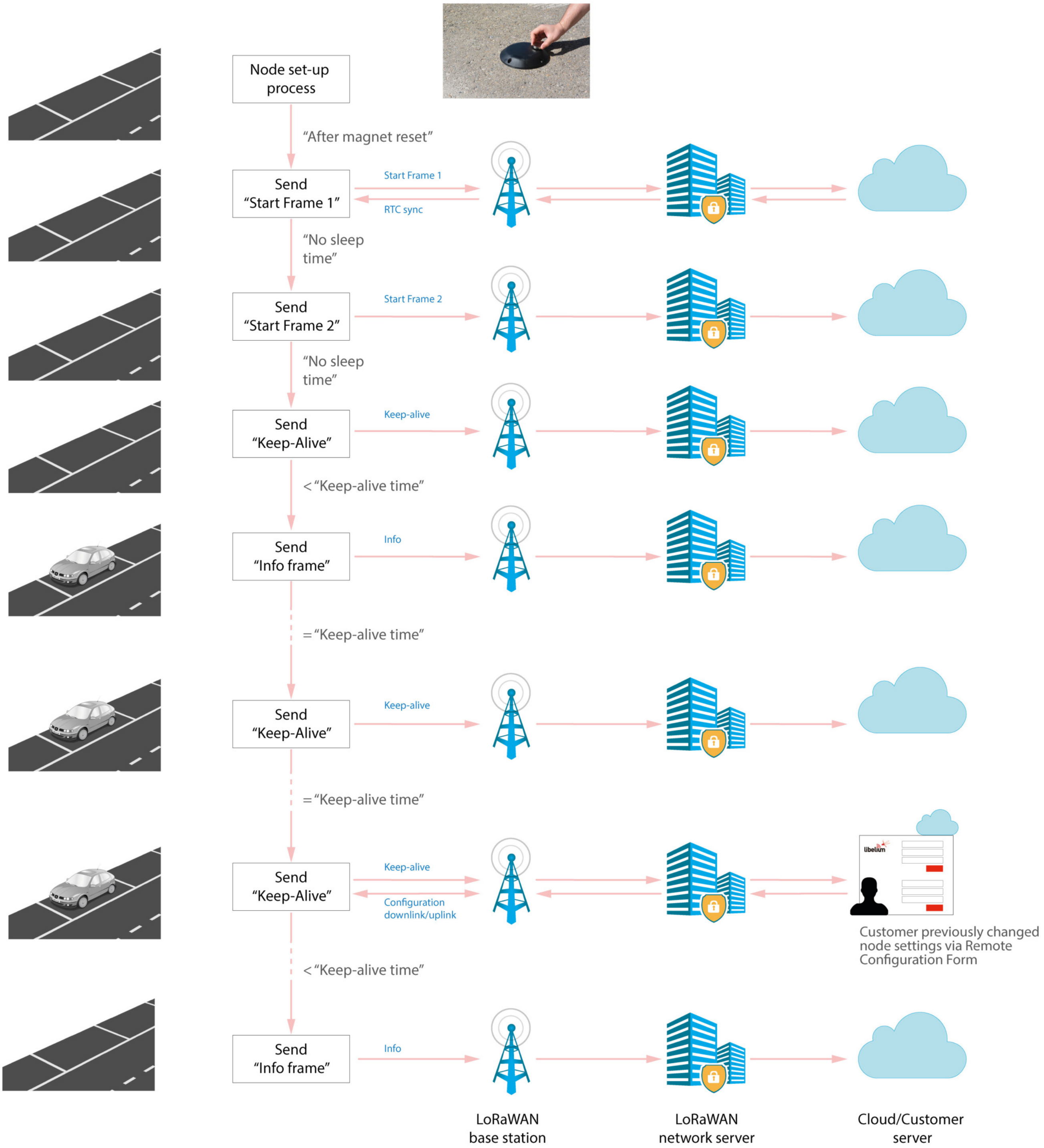


Figure : Smart Parking node working example

10. Certifications

10.1. CE (European Union)

In accordance with the RED Directive, Libelium Comunicaciones Distribuidas S.L. declares that the “Smart Parking v2 EU” device is in conformity with the European Union legislation and with the following harmonized standards and technical specifications:

- EN 50581:2012
- EN 301 489-1 Ver. 2.1.1 (2017-02)
- EN 301 489-3 Ver. 2.1.1 (2017-03)
- EN 305 550-2 V1.2.1 (2014-10)
- Draft EN 305 550 V2.1.0 (2017-10)
- EN 300 220-1 V3.1.1 (2016-11)
- EN 300 220-2 V3.1.1 (2016-11)
- EN 62479:2011
- UNE-EN 62368-1:2014 + AC:2015 + AC1:2015 + AC2:2015 + AC:2017-03 + A11:2017

10.2. FCC (United States)

“Smart Parking v2 US” complies with Title 47 Part 15 subpart B of the FCC rules and regulations for Radio Frequency Devices – Unintentional Radiators (2019). This device contains the following transmitter modules:

- FCC ID: T9JRN2903
- FCC ID: 2AQ6KA1

If you want to know more, contact the Libelium Sales Department.