

Meshlium Xtreme

Technical Guide



INDEX

1. General and safety information	6
2. Important: read me before using	7
3. Contents of the box	8
4. Specifications.....	10
5. Accessories	15
6. How to use Meshlium	17
6.1. Power supply	17
6.2. External SIM socket	22
6.3. How to install the antennas	23
6.4. Installation of the IP65 Ethernet cable	24
6.5. Installing Meshlium	27
6.6. Initialization, Restart and Shutdown.....	27
6.7. Setting the time.....	28
7. Understanding Meshlium	29
7.1. Concepts	29
7.2. Meshlium Models.....	30
7.3. Storage Options.....	32
7.4. Applications model by model	32
8. Meshlium Presets.....	43
8.1. Presets	43
8.2. Validation of Changes in Interfaces	45
9. Accessing Meshlium - Make it Easy!	46
10. Ethernet.....	48
10.1. Configuration.....	48
11. Wifi AP	51
11.1. Configuration.....	51
11.2. Clients connected.....	53

12. Wifi Mesh	54
12.1. Configuration.....	54
12.2. OLSR Options	55
12.3. The Gateway Node.....	57
13. 3G/GPRS	58
13.1. Configuration.....	58
14. Joined Networks	60
15. Wireless Sensor Networks.....	62
15.1. Configuration.....	62
15.2. Configuration of the ZigBee interface	63
15.3. Capturing and storing sensor data	66
15.3.1. Capturing and storing sensor data from XBee / LoRa	66
15.3.2. Capturing and storing data from 3G/GPRS, WiFi and Ethernet.....	68
15.3.3. Local Data Base.....	69
15.3.4. External Data Base.....	70
15.3.5. Show me now!	74
15.3.6. Advanced Database	75
15.4. Capturer logs.....	77
15.5. Sensors	78
15.6. OTA via FTP	79
15.7. Sending frames from Meshlium to Waspnote	81
15.8. Key Management on Meshlium	82
15.8.1. Link layer key Management (AES-128).....	82
15.8.2. Application Layer Key Management (AES-192/256)	83
15.9. AES on Meshlium.....	84
16. Meshlium Visualizer	85
16.1. Working with the Visualizer	85
17. Cloud Connector	89
17.1. IoT Platforms.....	90
17.1.1. Amazon IoT	90
17.1.2. Amplía's OpenGate.....	93
17.1.3. eagle.io	94
17.1.4. Esri.....	95
17.1.5. Extunda	103
17.1.6. IBM Bluemix	104
17.1.7. IoT-Ticket	105
17.1.8. Microsoft Azure Event Hubs	111
17.1.9. Microsoft Azure Service Bus	113
17.1.10. MQTT	116

17.1.11. Simfony	119
17.1.12. SmartCityPlatform	123
17.1.13. Telefónica IoT Platform	125
17.1.14. ThingWorx.....	126
17.2. IoT Solutions.....	130
17.2.1. B-Scada.....	130
17.2.2. DeviceLynk.....	131
17.2.3. Devicify.....	133
17.2.4. ElementBlue – RightSensor.....	134
17.2.5. IoTens.....	136
17.2.6. Sentilo	137
17.2.7. Sofia2.....	138
17.2.8. SolvView.....	145
17.2.9. ThingPlus	146
18. Smartphone Detection	149
18.1. Update! (Version 3.1.4)	152
18.2. WiFi Scan.....	157
18.2.1. Concepts	157
18.2.2. Capturing and Storing WiFi Scans	158
18.2.3. Local Data Base	159
18.2.4. External Data Base.....	160
18.3. Bluetooth Scans	163
18.3.1. Concepts	163
18.3.2. Local Data Base.....	165
18.3.3. External Data Base.....	166
18.3.4. Local File System.....	169
19. GPS.....	170
19.1. Concepts.....	170
19.2. Configuring GPS service.....	170
19.3. Capturing and Storing GPS Data.....	171
19.3.1. Local Data Base.....	171
19.3.2. External Data Base.....	173
19.3.3. Show me now!	176
20. Networking tools.....	177
20.1. Ping.....	177
20.2. Iperf	178
20.3. Traceroute.....	179
20.4. Netstat	180
20.5. Beep.....	180

21. Special options for Wifi networks.....	181
21.1. Long range links.....	181
21.2. Fresnel calculator.....	181
22. System	182
22.1. Hostname	182
22.2. Users Manager	182
22.3. Setting the time	182
22.4. Disk Usage.....	183
22.5. Internal temperature sensors.....	184
23. Data Base Management.....	185
23.1. MySQL Direct Access	185
23.2. PHPMyAdmin.....	186
24. FTP Management	187
24.1. Creating user	187
24.2. Setting permission	187
24.3. Accessing FTP server	188
25. Uploading my own applications to Meshlium	189
26. Installing new packages and libraries	191
27. Upgrading old versions of Meshlium.....	192
27.1. Checking the Libelium Repository.....	193
27.2. Local File	193
27.3. URL.....	194
28. Rescue System	195
28.1. Steps.....	195
28.2. How to Recover the file system	196
29. USB Device Connectivity.....	197
30. ManagerSystem Changelog	199
31. Documentation Changelog	204
32. Certifications.....	206
32.1. CE.....	206
32.2. FCC.....	208
33. Maintenance	211
34. Disposal and Recycling	212

1. General and safety information

- Read carefully the Limited Warranty and Terms and Conditions of Use before using “Meshlium”.
- Read carefully the “General Conditions of Sale and Use of Libelium”. This document can be found at: http://www.libelium.com/development/meshlium/technical_service
As specified in the Warranty document which you can find at: <http://www.libelium.com/development/meshlium/documentation>, the client has **7 days** from the day the order is received to detect any failure and report that to Libelium. Any other failure reported after these 7 days may not be considered under warranty.
- Do NOT open the casing. If you do so, you will lose the guarantee.
- Do not remove any of the components.
- Do not allow contact between metallic objects and the electronic part to avoid injury and burns.
- NEVER immerse the equipment in any liquid.
- Keep the equipment in a dry place away from any liquids that could spill.
- Check from the label that comes with the equipment the maximum permitted voltage and amperage range for powering it and use a power transformer within that range.
- Keep the equipment within the temperature range indicated in the specifications section.
- Do not connect or power the equipment using cables that have been damaged.
- Place the equipment in an area to which only maintenance personnel can have access (in a restricted access zone)
- In any case keep children away from the machine.
- If there is a power failure, immediately disconnect from the mains.
- If using the car lighter as a power source, make sure that you follow the voltage and current specifications indicated in the section “How to use Meshlium”.
- If using a battery whether or not in combination with a solar panel as a power source follow the voltage and current specifications indicated in the section “How to use Meshlium”.
- If a software failure occurs, consult the section Libelium web support.
- Do not place the equipment on trees or plants as they could be damaged by its weight.
- Be particularly careful if you are connected through an interface to the software for handling the machine; if the settings of that interface are incorrectly altered, it could become inaccessible.

2. Important: read me before using

The following list shows **just some** of the actions that produce **the most common failures and warranty-voiding**. Complete documentation about usage can be found at:

http://www.libelium.com/development/meshlium/technical_service

Failure to comply with the recommendations of use will entail the guarantee cancellation.

Software:

- Do not access Meshlium with the shell terminal unless you have advanced Linux skills. Use the Meshlium Manager web application in order to configure and setup Meshlium. A bad usage of the shell terminal commands may leave Meshlium without connection or within a unstable mode. All the failures derived from a wrong usage of the shell terminal are not covered under the warranty.
- In the case you configured Meshlium through the terminal, there is **not** a method to take Meshlium to the configuration by default: the only way to get connection again is sending Meshlium back to Libelium and hiring this service. However, if you **only** use the Manager System, you can always go to the default configuration by clicking the "Presets" button.
- Do not interrupt the power supply before shutting down Meshlium properly through the "Shutdown" or "Restart" buttons in the Manager System or through the '`restart-secure`' and '`shutdown-secure`' commands from the terminal. If you do not do so you may take the system corrupted.
- If you need to modify a file or directory, after executing the command '`remountrw`' and modifying it, execute immediately the command '`remountro`'.
- In the case of developing an application for Meshlium, store the files in '`/mnt/user`' and not in other directory.
- Regarding packages:
 - Do not ever use '`apt-get`'; if needed, use '`aptitude`'.
 - Do not ever make an '`upgrade`' of the entire system (not even using '`aptitude`').
 - Before installing new packages with '`aptitude`', execute the '`update`' option.
- The SSH password cannot contain special chars (e.g. \$, %, =). The SSH password can only be composed of letters and numbers. Be careful since Meshlium could be damaged with a not appropriate password.

Hardware:

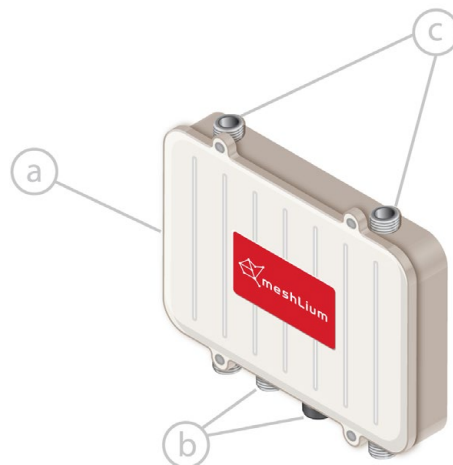
- Do not open the Meshlium enclosure in any case. This will automatically make the warranty void.
- Do not handle the numbered metallic seals in the screws of Meshlium: their integrity is the proof that the Meshlium enclosure has not been opened. If they have been handled, opened or broken, the warranty is void.
- Do not submerge Meshlium in liquids.
- Do not place Meshlium on places or equipment where the device could be exposed to shocks and/or vibrations.
- Do not expose Meshlium to temperatures below -20° C or above 50° C.
- Meshlium's microprocessor must not overpass 70 Celsius degrees. The user must ensure that this temperature never overpass. Especially when using Wifi Scan.
- Do not power Meshlium with other power sources than the original provided by Libelium.

For more information: <http://www.libelium.com/meshlium>

3. Contents of the box

1. Meshlium

- a. IP65 casing
- b. Ethernet connectors
- c. Antenna connectors

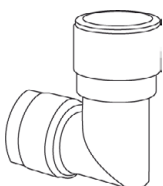


2. Antennas *

- a. Dipole 5dBi
(Bluetooth, ZigBee, Wifi 2.4GHz low)
- b. Dipole 5dBi (868MHz, 900MHz)
- c. Omnidirectional 5dBi
Dual (2.4GHz / 5GHz)
- d. 3G/GPRS
- e. 3G/GPRS
- f. GPS

(*) *ewwpende on the configuration you choose when you buy*

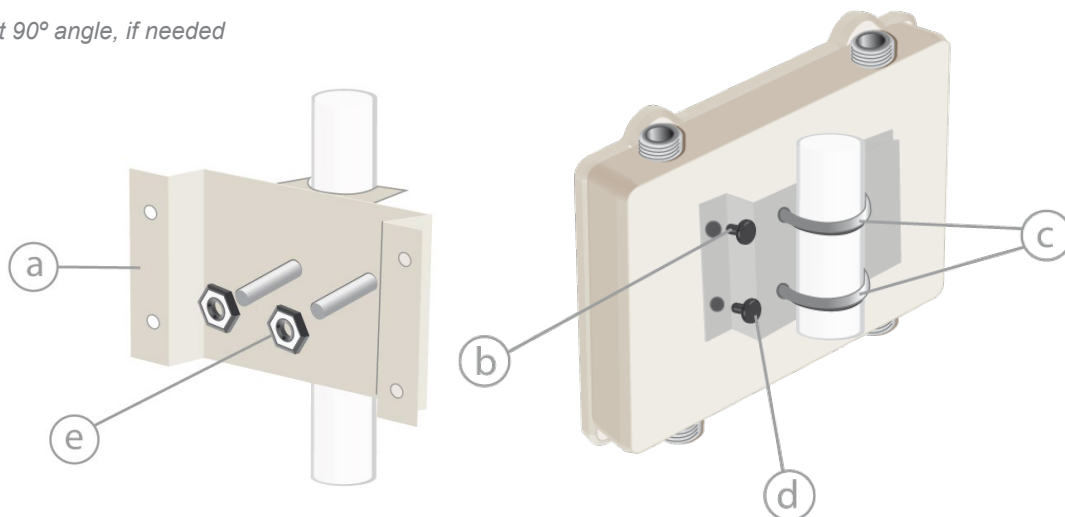
3. Antennas elbow connectors *



(*) *Allow to connect antennas at 90° angle, if needed*

4. Fixing

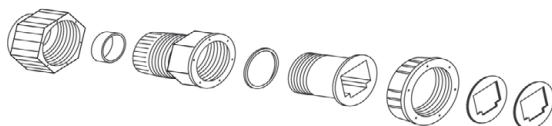
- a. Fixing plate
- b. 4 washers
- c. 2 U-shaped parts
- d. 4 screws
- e. 4 nuts



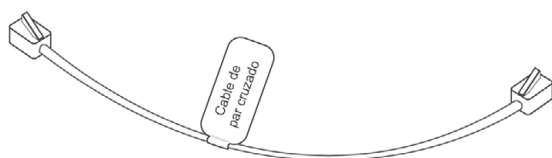
5. Ethernet cable



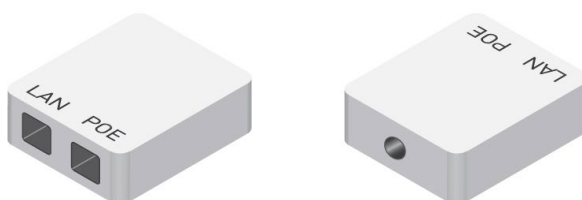
6. IP65 Ethernet cap



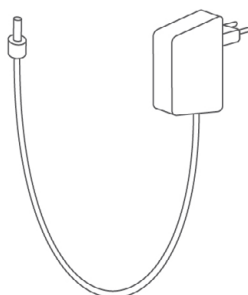
7. Crossover cable



8. POE



9. Charger



Note: you may receive an empty foam protector in the packaging, depending on the Meshlium configuration that you have purchased.

4. Specifications



Processor	500MHz (x86)	
RAM memory	256MB (DDR)	
Disk memory	8GB	
Power	5W (18V)	
Power Source	POE (Power Over Ethernet)	
Normal Current Consumption	270mA	
High Current Consumption	450mA	
Max Supply Current	1'5A	
Enclosure	Material	Aluminium
	Dimensions	210x175x50mm
	Weight	1,2Kg
	External protection	IP65
Temperature Range	-20°C / 50°C	
Response Time to ethernet ping	60s	
Time to have all the services running	90s	
Types of power supply * for POE	AC-220V	
	Battery – solar panel (DC-12V)	
	Car lighter (DC-12V)	
System	Linux, Debian. OLSR Mesh communication protocol.	
	Madwifi Drivers.	
Management software	Meshlium Manager System (open source)	
Security	Authentication WEP, WPA-PSK, HTTPS and SSH access.	

(*) Only with the accessories supplied by Libelium

Wifi AP - 2.4GHz Radio



WIFI RADIO	
Chipset	Atheros AR5213A - IEEE 802.11b/g
Tx-Power	100mW - 20 dBm
Distance	500m *

(*) Depending on antenna and line of sight

ANTENNA	
Type	Omni-directional
Gain	5dBi
Dimensions	224 x 22 mm

Wifi Mesh Dual Band 2.4 GHz / 5GHz Radio



WIFI RADIO	
Chipset	Atheros AR5213A - IEEE 802.11a/b/g
Tx-Power	20dB - 802.11b/g / 18dB - 802.11a
Distance	2-50km *

(*) Depending on antenna and line of sight

ANTENNA	
Type	Omni-directional
Gain	5dBi - 2.4GHz / 8dBi - 5GHz
Dimensions	224 x 22 mm

RF Module Radio (Xbee or LoRa)



Model	XBee - PRO - 802.15.4
Frequency	2.4GHz
Tx-Power	100mW
Rx Sensitivity	-100dBm
Antenna	5dBi Dipole
Distance	7km *

Model	XBee - PRO - ZigBee
Frequency	2,4GHz
Tx-Power	50mW
Rx Sensitivity	-102dBm
Antenna	5dBi Dipole
Distance	7km *

Model	XBee - PRO - 868
Frequency	868MHz
Tx-Power	315mW
Rx Sensitivity	-112dBm
Antenna	5dBi Dipole
Distance	12km *

Model	XBee - PRO - 900
Frequency	900MHz
Tx-Power	50mW
Rx Sensitivity	-100dBm
Antenna	5dBi Dipole
Distance	10km *

(*) Depending on antenna and line of sight

RF Module Radio (Xbee or LoRa) (continuation)



Model	XBee - PRO - Digimesh
Frequency	2.4GHz
Tx-Power	100mW
Rx Sensitivity	-100dBm
Antenna	5dBi Dipole
Distance	7km *

(*) Depending on antenna and line of sight



Model	LoRa (Semtech SX1272) 868 / 915MHz
Frequency	868 and 915 MHz
Tx-Power	14 dBm
Rx Sensitivity	-137 dBm
Antenna	4.5 dBi dipole
Distance	21+ km*

(*) Depending on antenna and line of sight

Wifi Scanner



Chipset	Atheros AR5213A - IEEE 802.11b/g
Distance	50-200m *
Antenna	5dBi Dipole

(*) Depending on antenna and line of sight

Bluetooth Scanner



Protocol	Bluetooth 2.1 + EDR. Class 2
Tx-Power	3dBm
Antenna	5dBi
Power	3dBm
Distance	20-30m *

(*) Depending on antenna and line of sight

3G/GPRS Module



Protocols	3G *, WCDMA, HSPA, UMTS, GPRS, GSM
Tri Band (UMTS)	900/1900/2100MHz
Quad Band (GSM/GPRS/EDGE)	850/900/1800/1900 MHz
Output power	UMTS 900/1900/2100: 0.25W GSM850/GSM900: 2W DCS1800/PCS1900: 1W
Rx Rate	7.2Mb/s
Tx Rate	5.5Mb/s
Antenna	3dBi
SIM card	Access via the External micro-SIM socket

(*) Note for US users: We tested the 3G shield with the AT&T network which supports natively the GSM and 3G protocols. With other carriers may also work although we haven't tried and thus we can not ensure it. For this reason we recommend to use AT&T SIM cards.

GPS Module



Modes	Assisted GPS (A-GPS), Standalone mode (NMEA frames)
Antenna	26dBi (+/-4.5dBi) - 3m cable. Magnetic

5. Accessories

Meshlium accessories are not included in the box.

1. 220AC Adapter - car lighter/battery (12V – 300W)

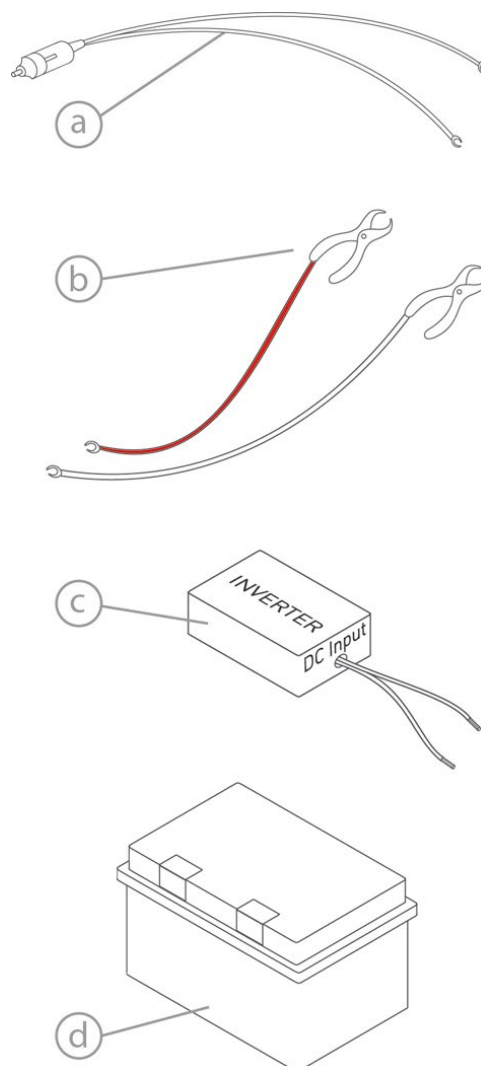
- a. Car lighter adapter (on demand)
- b. Battery clamps *
- c. 12 VDC - 220 VAC converter

Average power	300 W
Maximum power	600 W
Input voltage	12 VDC
Output voltage	220 VAC
Dimensions	205x165x67 mm
Weight	1 Kg

d. 12 VDC Battery *

(*) Libelium does not supply these components;

you can purchase it from battery sales points



2. Solar connection kit (12V – 20W):

a. Solar panel

Power	20 W
Maximum voltage	17 V
Maximum current	1.15 A
Dimensions	480x430x30 mm
Weight	2.50 Kg

b. 12 VDC - 220 VAC converter

Average power	300 W
Maximum power	600 W
Input voltage	12 VDC
Output voltage	220 VAC
Dimensions	205x165x67 mm
Weight	1 Kg

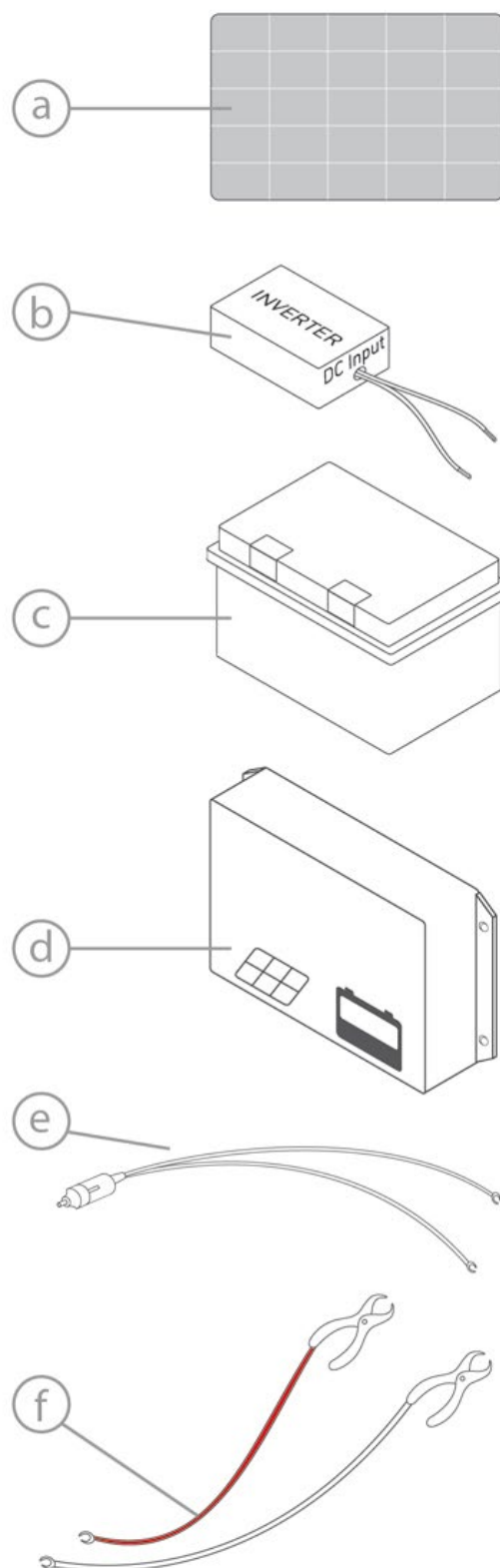
c. 12 VDC Battery *

d. Charge regulator

Charge voltage range	11.10V - 21.20V
Maximum input current	5A
Dimensions	130x70x25 mm
Weight	200 g

e. Car lighter adapter (on demand)

f. Battery clamps *



(*) Libelium does not supply these components;

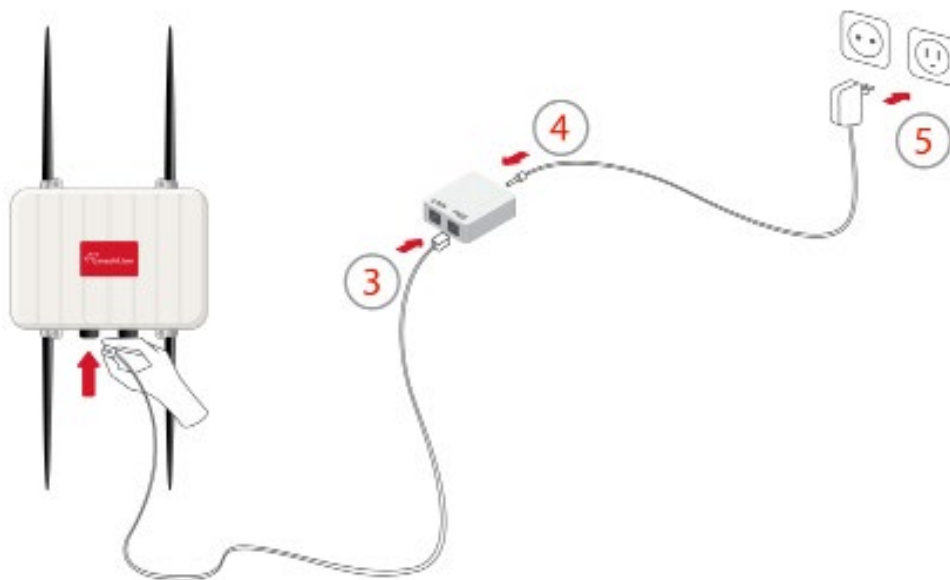
you can purchase it from battery sales points

6. How to use Meshlium

6.1. Power supply

How to connect Meshlium to 220V:

1. Take the Meshlium casing and unscrew the Ethernet connector cap shown in the drawing.
2. Join the end that has the IP65 protection of the IP65 Ethernet cable to said connector and screw the cap on to fix it.
3. Connect the other end of the cable to the POE input marked "POE". As explained in the section "Before using Meshlium" make sure that the POE is indoors.
4. Take the supplied POE power adapter and plug it into the corresponding POE connector.
5. Plug the other end of the adapter into the 220V socket and your Meshlium is now ready to operate.



In addition to this option, you can power Meshlium through a solar panel and battery or through a car lighter socket. The accessories for these options are NOT included in the box, and the necessary components must be purchased from Libelium.

Except for the battery it is not recommended to use third party components since they have not been tested by Libelium and could cause failures in the device.

At Libelium we offer an equipment for operating at 12V. It is very important to bear in mind that not all batteries supply the same voltage. You must use a 12V battery .

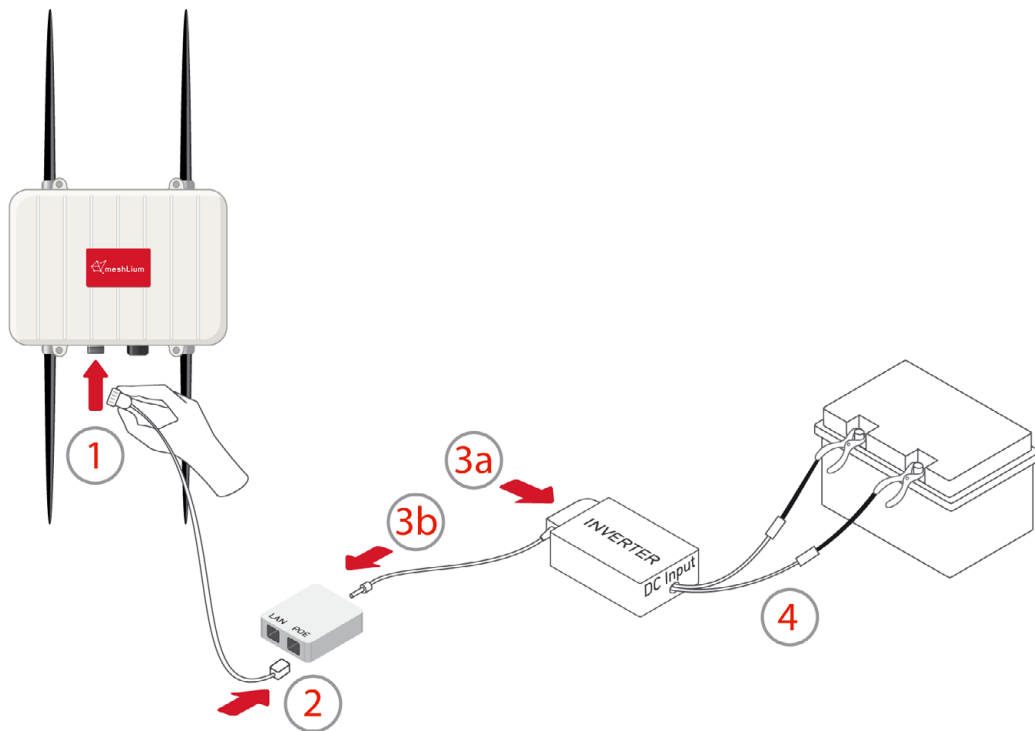
Note: Take special care to ensure that the battery you connect has the same power rating as the kit that you have purchased.

How to connect Meshlium for powering through the battery:

Additional required components: 12V battery, with exposed terminals for placing the clamps.

1. Connect the end with the IP65 protection of the Ethernet cable to the Meshlium connector that does not have the cap.
2. Connect the end of the Ethernet cable to the POE input marked "POE". As explained in the section "Before using Meshlium" make sure that the POE is indoors.
- 3a. Plug the POE charger into the Inverter output (AC output).
- 3b. Connect the POE charger cable into the POE.
4. Connect the ends of the cables which come from the Inverter to the battery. To do so, it is advised to use battery clamps. In this case the cables of the Inverter must be spliced to the cables of the clamps, and they are the clamps the part that is connected to the terminals of the 12 VDC battery.

Important: Check polarity at all times before making the connections. It is advised to avoid the possibility of contact with the positive and negative polo at any time. There is **lethal shock hazard** and the devices can also be damaged.



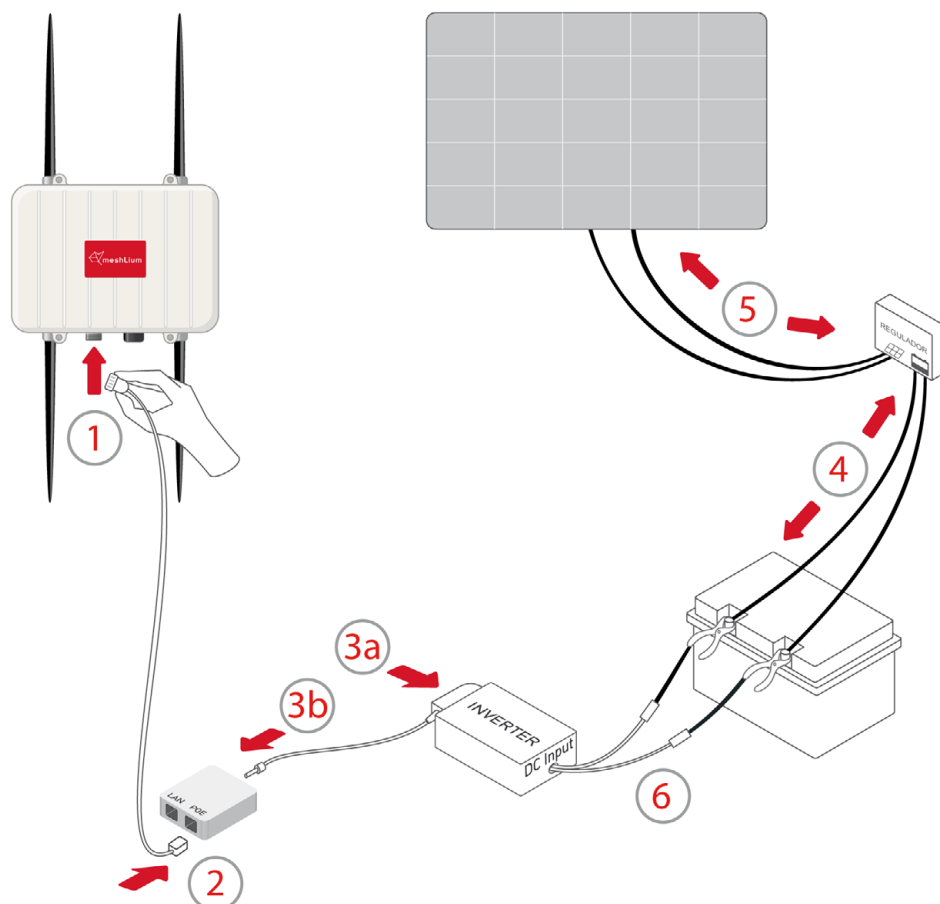
How to connect Meshlium for solar panel and battery powering:

Additional required components: solar panel / 12V-220V and 12V battery, with exposed terminals for placing the clamps.

Important: Check polarity to make sure that the positive (+) and negative (-) poles are connected to the right socket.

1. Connect the end with the IP65 protection of the Ethernet cable to the Meshlium connector that does not have the cap.
2. Connect the end of the Ethernet cable to the POE input marked "POE". As explained in the section "Before using Meshlium" make sure that the POE is indoors.
- 3a. Plug the POE charger into the Inverter output (AC output).
- 3b. Connect the POE charger cable into the POE.
4. Connect a cable to each battery terminal and screw the other end into the socket of the Charge Regulator marked as "battery". Take polarity (+) / (-) into consideration.
5. Connect the solar panel cables into the socket of the Charge Regulator marked as "solar". Take polarity (+) / (-) into consideration.
6. Connect the ends of the cables which come from the Inverter to the battery. To do so, it is advised to use battery clamps. In this case the cables of the Inverter must be spliced to the cables of the clamps, and they are the clamps the part that is connected to the terminals of the 12 VDC battery.

Important: Check polarity at all times before making the connections. It is advised to avoid the possibility of contact with the positive and negative pole at any time. There is **lethal shock hazard** and the devices can also be damaged.

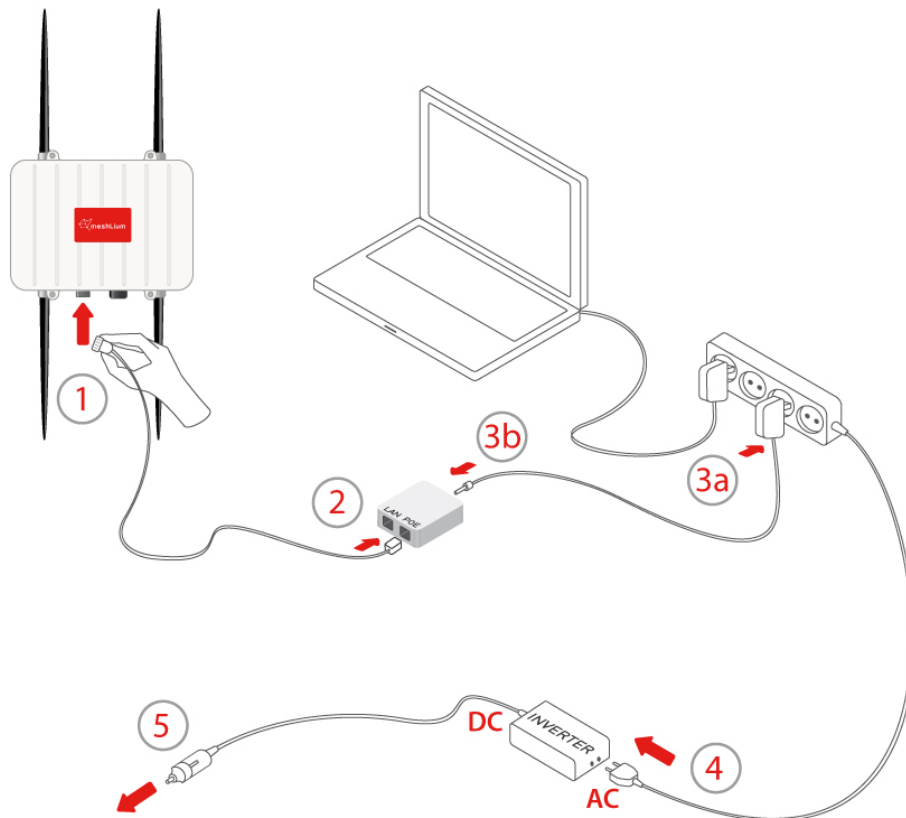


How to connect Meshlium for powering through a car lighter:

In this case, you must purchase the required components from Libelium. We do not advise using third party components as they have not been tested by Libelium and could cause failures in the device or the car.

Important: Bear in mind that not all car lighters supply the same voltage. The Inverter offered by Libelium works at 12 V DC input voltage. Although it must be checked in each case, cars normally have 12 V as output of the lighter.

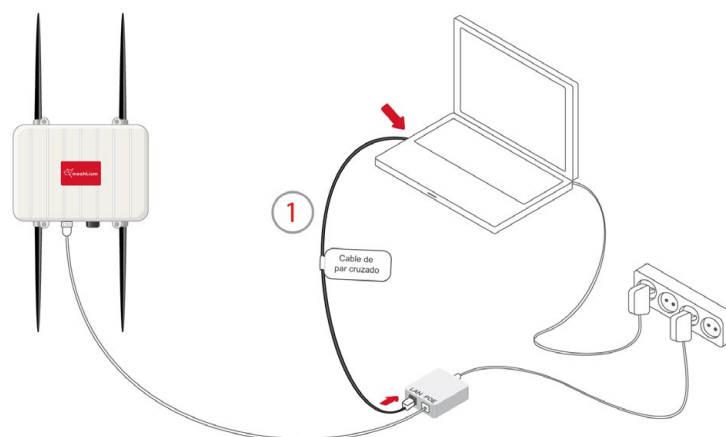
1. Connect the end with the IP65 protection of the IP65 Ethernet cable to the Meshlium connector without the cap.
2. Connect the other end of the cable to the POE input marked "POE". As explained in the section "Before using Meshlium" make sure that the POE is indoors.
- 3a. Plug the POE charger into the Inverter output (AC output). You can also connect it to a multi-socket adapter, as the image shows.
- 3b. Connect the POE charger cable into the POE.
4. In the case of using a multi-socket adapter, connect the end of its cable to the Inverter.
5. Plug the end of the cable which comes out from the Inverter to the lighter socket of a car.



How to connect Meshlium in order to get access by the Ethernet interface:

1. Connect the network crossover cable (it has an identifying label) included in the box to the POE input marked "LAN" and to the network socket of your PC as shown in the drawing. The procedure is the same for any of the selected power supply options.

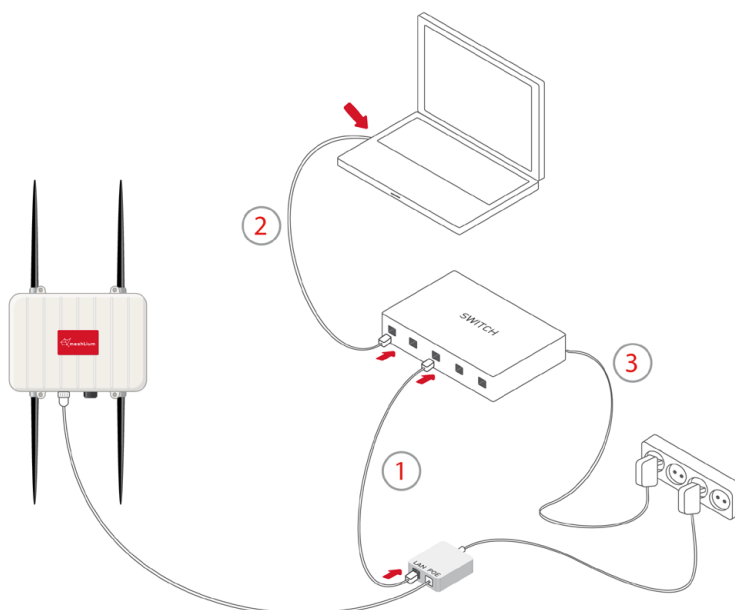
(*) See the "Accessing Meshlium" section in order to see how to get access wirelessly.



You can also carry out this connection through a switch (not supplied with Meshlium):

1. Connect the Ethernet cable (not the crossover) to the POE input marked "LAN" and to one of the switch inputs.
2. Connect another Ethernet cable to another one of the switch inputs and the opposite end to the network socket of your PC.
3. Plug the switch into the network.

(*) See the "Accessing Meshlium" section in order to see how to get access wirelessly.



6.2. External SIM socket

The External SIM socket replaces the USB socket in two devices:

- Meshlium devices with **3G/GPRS** module
- Waspote Plug & Sense! devices with **GPRS**, **GPRS+GPS** or **3G+GPS** module

The External SIM socket is composed of 2 connectors:

- micro-SIM card
- micro-USB (type B)



Figure : External SIM socket in a Meshlium with 3G/GPRS module

The micro-SIM card connector allows the user to connect the SIM card he likes from the outside. It is no longer necessary to send a SIM card to Libelium for proper installation. You can ask your telecommunication provider for a micro-SIM card. Alternatively you can take a normal SIM card and transform it into a micro-SIM card with a SIM card cutter.

Besides, the micro-SIM card connector has a push-pull mechanism, so it is really easy to remove the card with the aid of one nail.



Figure : Push-pull mechanism in the External SIM socket

It is highly important to turn off Meshlium device in a secure way before inserting a micro-SIM card, or removing an existing SIM-card. The user can damage the device if this operation is done "on-the-fly".

Make sure you closed the External SIM socket with its protection cap before outdoors deployment.

The operation with the micro-USB socket is just the same than with the normal USB socket (please read the Rescue System section). Just remember to use a micro-USB cable.

Take into account that the External SIM socket has a limited resistance so please be gentle and do not push too hard.



Figure : Inserting a SIM card with care in the External SIM socket

6.3. How to install the antennas

Check the labels next to the antennas to know which is the one that needs to be connected (refer to page 7 to know which technology corresponds to each antenna). If the antenna is of the GPS, 3G/GPRS, Wifi or Dipole 5dBi (868MHz, 900MHz) type, the corresponding adapter will have to be installed as indicated in figure "a". All the antennas are screwed into place. To install the Meshlium antennas place them in the corresponding connectors as shown in the drawing.



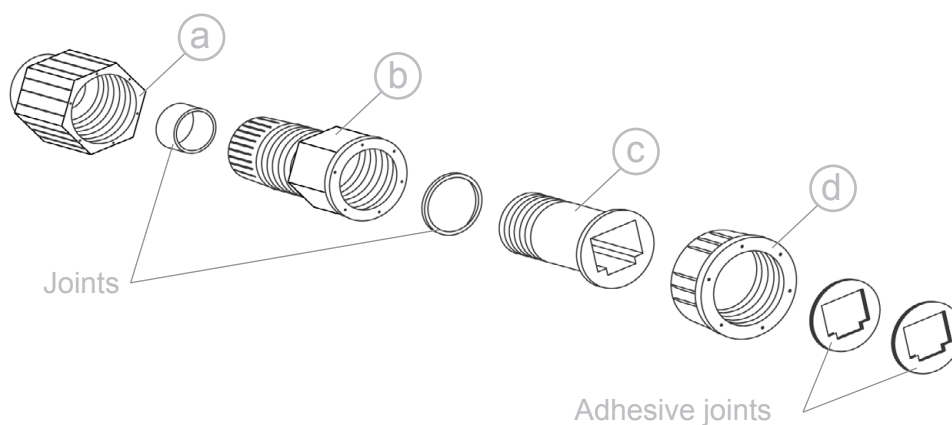
6.4. Installation of the IP65 Ethernet cable

Fitting of the IP65 cap:

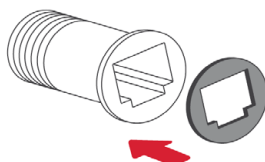
In order to fit the IP65 cap you will need a connector-free RJ45 cable. This cable is NOT included in the Meshlium box.

Important: Make sure that you buy a sufficiently long cable to connect Meshlium from its position to the POE situated indoors.

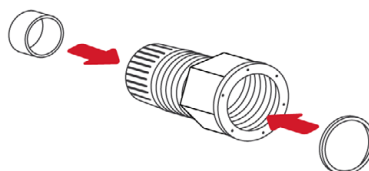
1. Take from the Meshlium box the bag containing the parts for fixing the IP cap. Check that you have all the parts that appear in the picture.



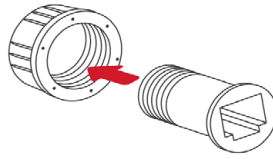
2. Stick one of the supplied adhesive joints to part C.



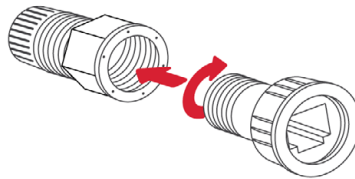
3. Slot the joints into part B as shown in the drawing.



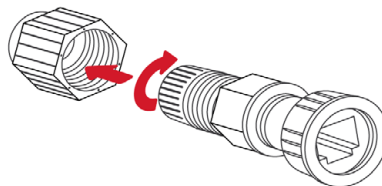
4. Insert part C into part D.



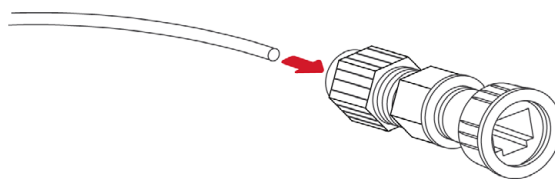
5. Screw both sets of parts in the direction shown in the diagram.



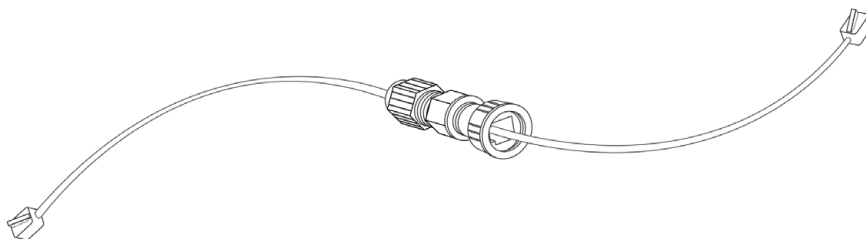
6. Screw part D to the end.



7. Pass the cable through the fitted cap.



8. Crimp the RJ45 connectors at the ends of the cable (the crimping tool is not supplied with Meshlium).



Your IP65 Ethernet cable is now ready for use.

How to connect the IP65 Ethernet cable to Meshlium:

1. Take the adhesive joint that has not been used for fitting the cap and stick it to the Meshlium bare Ethernet connector.



2. Connect the end of the Ethernet cable to the Meshlium Ethernet socket.



3. Screw part C onto the Meshlium connector. Your Meshlium is now ready for using outdoors.

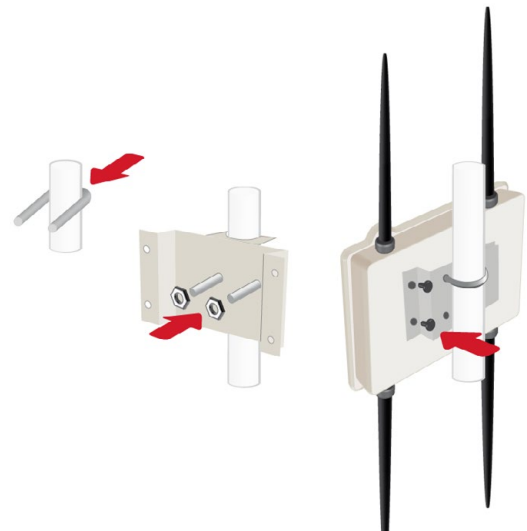


6.5. Installing Meshlium

Meshlium has been designed to operate in a vertical position. You will find the required fixtures for this in the box. The image shows the steps to follow to secure Meshlium to a vertical post.

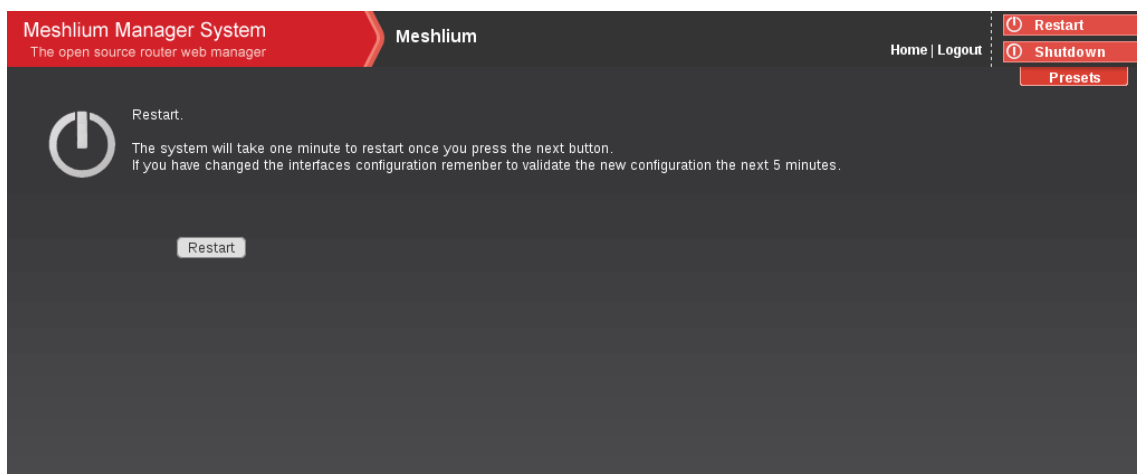
1. Place the U-shaped piece fitted against the post.
2. Slot the fixing plate into the U-shaped piece leaving the post between the two parts. Secure both parts by turning the nuts included in the U-piece. Repeat these steps with the second U-piece.
3. Place Meshlium so that the casing holes coincide with the fixing plate and fix Meshlium to the post by turning the screws into these holes.

As specified in the section "General and Safety Information" the device must be installed by trained personnel only in an area of restricted access.



6.6. Initialization, Restart and Shutdown

In order to allow Meshlium to close correctly all the daemons and applications it is important to use the buttons "Restart" and "Shutdown" placed in the upper right corner in the Manager System. This way you will keep maximum the performance and lifetime of the system.



If you are connected to Meshlium through a SSH connection you must use the following scripts:

```
/bin/restart-secure
/bin/shutdown-secure
```

They can be executed from the terminal directly:

```
$ restart-secure
$ shutdown-secure
```

Beep! System

Meshlium includes an internal speaker which will emit “beep!” sounds when initializing, rebooting and shutting down in order to inform about the state of the process.

Initialization beeps:

- 1 short beep when Meshlium is powered
- 1 long beep when Meshlium starts launching the operate system
- 2 long beeps when Meshlium has finished starting and it is ready to be used

Reboot beeps:

- long beep when the reboot order is executed.
- Initialization beeps when Meshlium starts again.
- Do not remove the power cable during this process is carried out.

Shutting down beeps:

- This process could take up to one minute.
- 2 long beeps when Meshlium is about to shutdown. A few seconds after the beeps, Meshlium can be unplugged.
- Do not remove the power cable until this process is totally completed.

Note: The “beep!” sound is not really loud so you will have to take attention and be close to the Meshlium box in order to hear them clearly.

Note 2: If Meshlium is unplugged before the acoustic signal of shutdown, internal memory could be damaged. Be sure to wait for several minutes if you are not sure the beeps sounded.

Note 3: The duration of the reboot or shut-down processes may vary. Make sure you heard the corresponing beeps and be patient.

Note 4: If the user does not follow these instructions, the risk is very high. Meshlium will become unresponsive and unaccessible. This problem is out of the warranty scope, because it is produced by bad use. The only possible solution will be a repair process in Libelium’s facilities, paid by the user.

6.7. Setting the time

In order to get all the data stored in the Meshlium Local Data Base with the right timestamp you must adjust the System time. To do so go to the “Setting the Time” section inside the “System” chapter in the current manual.

7. Understanding Meshlium

7.1. Concepts

Meshlium is a Linux router which can contain 6 different radio interfaces: Wifi 2.4GHz, Wifi 5GHz, 3G/GPRS, Bluetooth and **RF** communications. RF communications may be implemented by one **XBee** module or one **LoRa** module. As well as this Meshlium can also integrate a GPS module for mobile and vehicular applications and be solar and battery powered. These features along with an aluminium IP65 enclosure allows Meshlium to be placed anywhere outdoor.

Meshlium can work as:

- an RF (XBee/LoRa) to Ethernet router for Waspote nodes *
- an RF (XBee/LoRa) to 3G/GPRS router for Waspote nodes *
- a Wifi Access Point
- a Wifi Mesh node (dual band 2.4GHz-5GHz)
- a Wifi to 3G/GPRS router
- a Bluetooth scanner and analyzer
- a GPS-3G/GPRS real-time tracker
- a Smartphone scanner (detects iPhone and Android devices)

(*) More info about Waspote at: <http://www.libelium.com/waspote>

All the networking options can be controlled from two different sources:

1. **Manager System:** a web interface which comes with Meshlium. It allows you to control all the interfaces and system options in a secure, easy and quick way.
2. **SSH console:** for expert users direct access to the shell console is enabled.

Meshlium is also a complete Linux station which offers different services, programming environments and storage systems:

Services activated:

- HTTP / HTTPS
- SSH

Examples of programming environments to be installed:

- C (by default)
- C++
- Java
- PHP (by default)
- Python
- Perl
- Ruby

Regarding the information storage Meshlium counts with two different data base systems.

- MySQL (by default)
- Postgre

All the information coming from all the interfaces (RF module [XBee/LoRa], Bluetooth, 3G/GPRS, Wifi and from the GPS module) can be stored in the Local File System and/or the Local Data Base as explained in the "Storage Options" section or even exported to an external Data Base connected to the Internet.

Also, when Meshlium is the central node in a Wireless Sensor Network, it can perform the OTA feature (over the air programming). That is to say, Meshlium can send to some (or all) Wasmotes in the network a new program via wireless. This feature is available for Meshliums with all XBee radios (802.15.4, DigiMesh, 868 MHz and 900 MHz) with the **exception** of Meshliums with **ZigBee** radio. Also, OTA can be performed with Wasmotes with GPRS, 3G or WiFi. For more information, please read the OTA Programming Guide.

Note: SSH access to Meshlium devices provided as part of IoT Vertical Kits is not activated.

7.2. Meshlium Models

There are different Meshlium models ("Presets") depending on the radios integrated and on the fact if they act as a Gateway (GW) node or not.

Models	Wifi AP (2.4GHz)	Wifi Mesh (2.4GHz /5GHz)	3G/GPRS	RF Module (XBee/LoRa)	Ethernet	GW
Meshlium AP	✓				✓	✓
Meshlium 3G/GPRS-AP	✓		✓		✓	✓
Meshlium Mesh-AP	✓	✓			✓	
Meshlium Mesh-AP-GW	✓	✓			✓	✓
Meshlium Mesh-3G/GPRS-AP	✓	✓	✓		✓	✓
Meshlium RF-AP	✓			✓	✓	✓
Meshlium RF-3G/GPRS-AP	✓		✓	✓	✓	✓
Meshlium RF-Mesh-AP	✓	✓		✓	✓	
Meshlium RF-Mesh-AP-GW	✓	✓		✓	✓	✓
Meshlium RF-Mesh-3G/GPRS-AP	✓	✓	✓	✓	✓	✓
Meshlium Scanner AP	✓				✓	✓
Meshlium Scanner 3G-/GPRS-AP	✓		✓		✓	✓
Meshlium Scanner RF-AP	✓			✓	✓	✓

All the configurations will act as GW nodes except the nodes configured to act as simple mesh nodes (Mesh-AP and XBee-Mesh-AP) which need one GW node in the mesh network which shares its Internet connection. **However, all the nodes can be changed to act as GW nodes with just one click through the Presets section.** LoRa models do not have Mesh capabilities.

All the Meshlium models come with **Ethernet (IPv4 and IPv6)** interfaces.

On some of them two **extra modules** can be installed: **Bluetooth** and **GPS**.

Models	Wifi Scans	Bluetooth Scans	GPS
Meshlium AP			
Meshlium 3G/GPRS-AP			Optional
Meshlium Mesh-AP			
Meshlium Mesh-AP-GW			
Meshlium Mesh-3G/GPRS-AP			Optional
Meshlium RF-AP			
Meshlium RF-3G/GPRS-AP			Optional
Meshlium RF-Mesh-AP			
Meshlium RF-Mesh-AP-GW			
Meshlium RF-Mesh-3G/GPRS-AP			
Meshlium Scanner AP	✓	✓	
Meshlium Scanner 3G/GPRS-AP	✓	✓	
Meshlium Scanner RF-AP	✓	✓	

For more information about each radio interface and module see its specific section in the current manual.

7.3. Storage Options

The size of the Meshlium hard disk is 8 GB . The Operating System and the Manager System take ~2.5GB. This means the space which can be used to store the data captured and to be used by the applications loaded by the user is:

- $8\text{GB} - 2.5\text{GB} = 5.5\text{GB}$

This space is assigned to the user partition: `"/mnt/user"`

There are three folders where the captured data is stored when the "Store to a File or Data Base" option is activated:

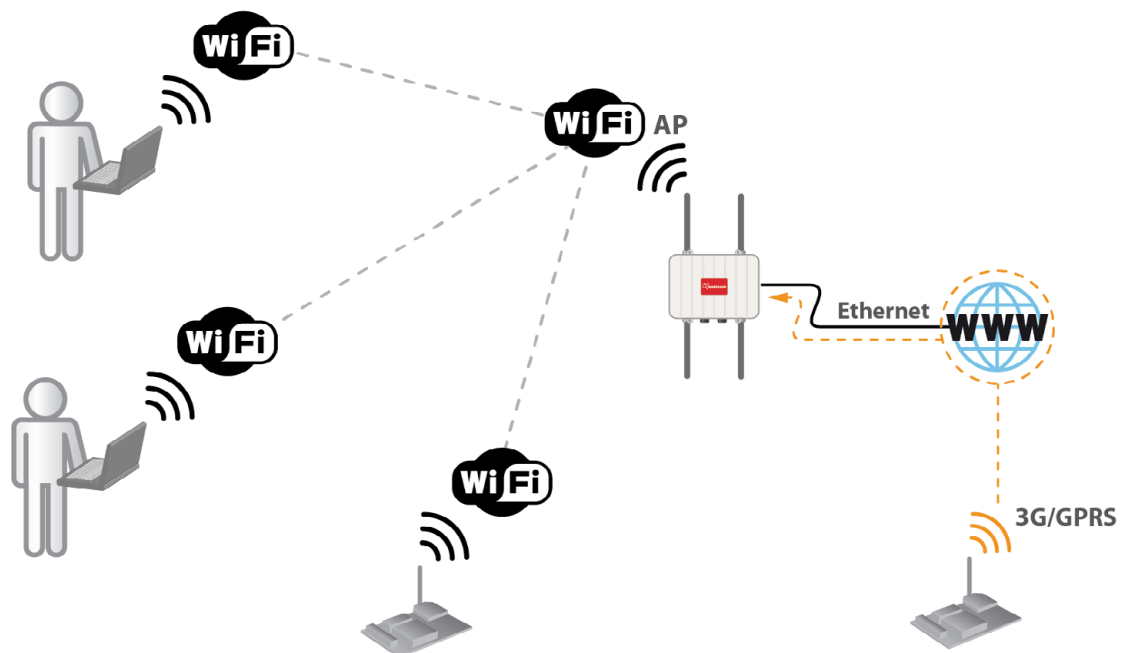
- `/mnt/user/zigbee_data`
- `/mnt/user/bluetooth_data`
- `/mnt/user/gps_data`

The Local Data Base files can be found in: `"/mnt/user/mysql/MeshliumDB"`

7.4. Applications model by model

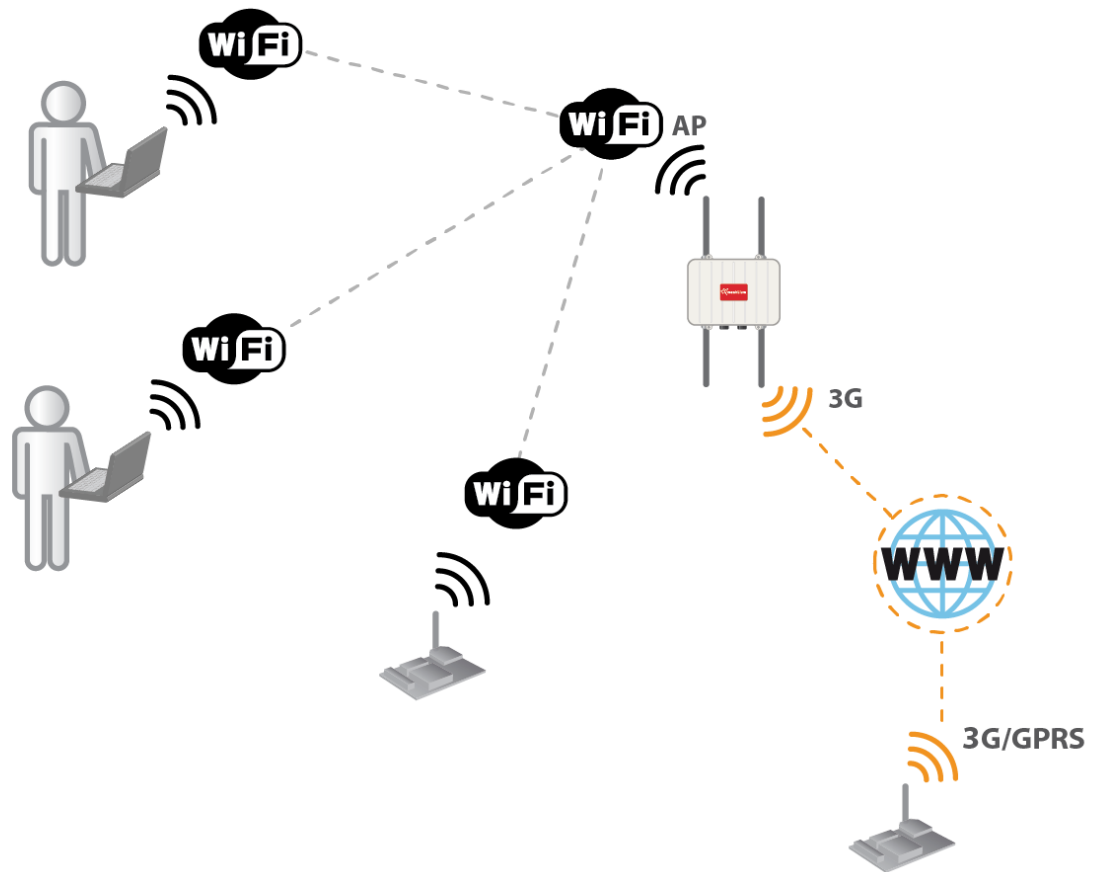
Meshlium AP

Clients can connect to Meshlium via Wifi with laptops and smart phones and get access to the Internet. In order to give Internet access Meshlium uses the Ethernet connection. Just connect it to your hub or switch and it will get automatically an IP from your network using DHCP. Wasmotes with GPRS, GPRS+GPS, 3G+GPS or WiFi can send sensor data through the access point or through the Internet via HTTP protocol.



Meshlium 3G/GPRS AP

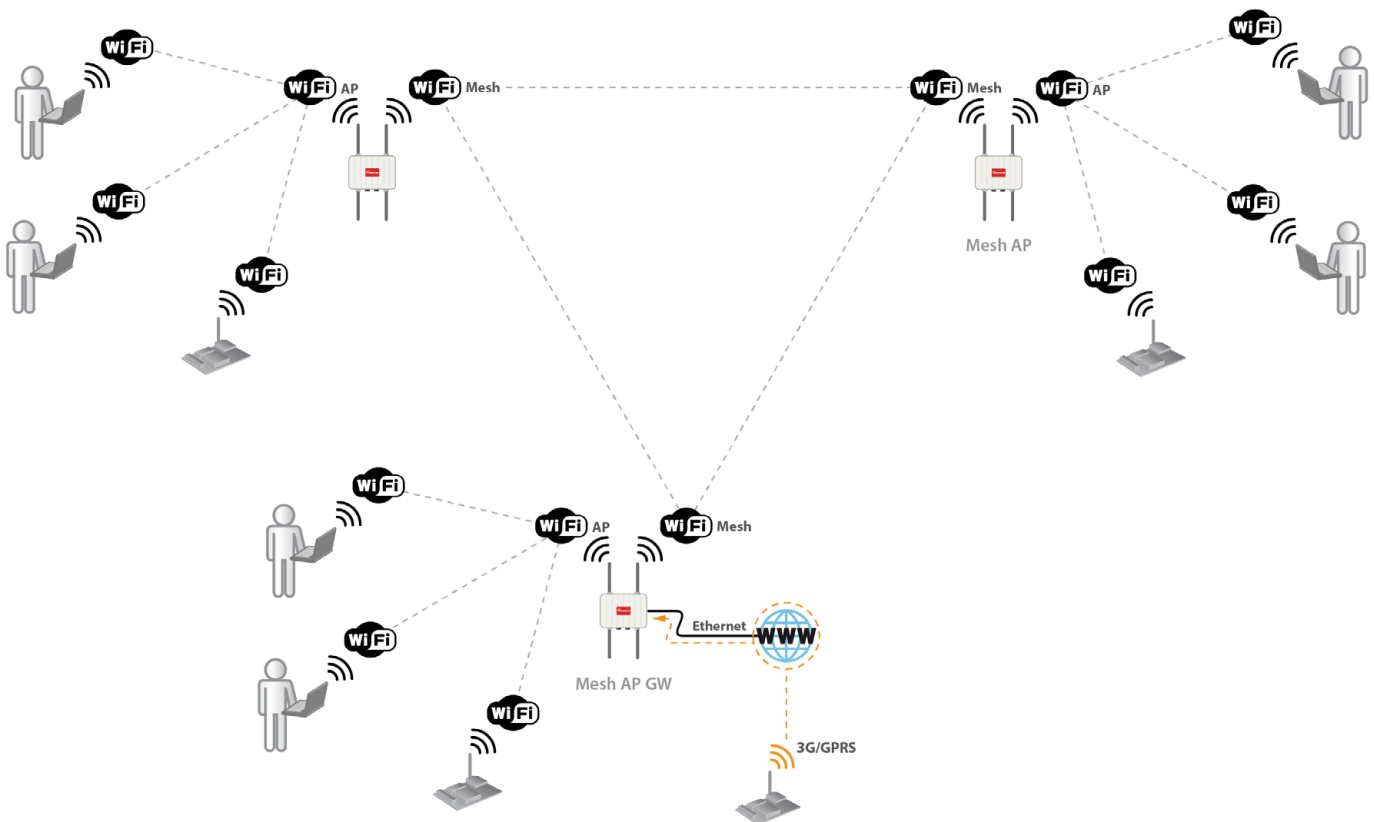
Clients can connect to Meshlium via Wifi with laptops and smart phones and get access to the Internet. In order to give Internet access Meshlium uses the 3G/GPRS connection. Waspnotes with GPRS, GPRS+GPS, 3G or WiFi can send sensor data through the access point or through the Internet via HTTP protocol.



Meshlium Mesh AP - Meshlium Mesh AP GW

Meshlium can work as a Mesh node. This means we can interconnect several Meshliums in order to share a common resource as the Internet connection. This way, the clients connected to a certain node can access to the Internet connection of a third node which is some hops far away the actual connecting point. Wasmotes with GPRS, GPRS+GPS, 3G or WiFi can send sensor data through the access point or through the Internet via HTTP protocol. In a mesh network there are two kinds of nodes:

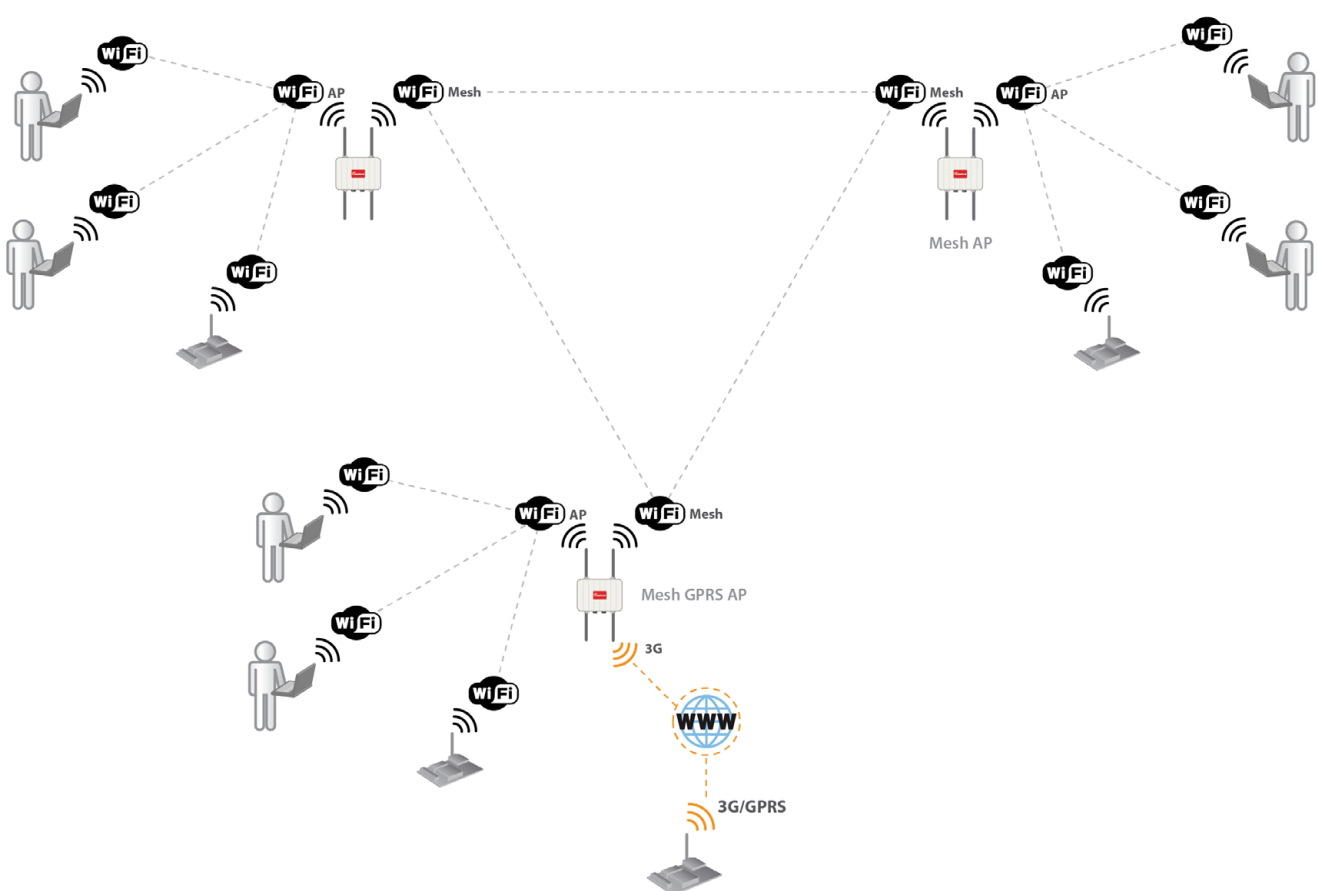
- **Meshlium Mesh AP** allows clients to connect via Wifi and creates links with other nodes using a second Wifi radio which operates in the 5GHz band.
- **Meshlium Mesh AP GW** (gateway) is the node which shares its Internet connection with the rest of the network. It takes the Internet connection from the Ethernet interface. There is only one GW in the mesh network.



Meshlium Mesh AP - Meshlium Mesh 3G/GPRS AP:

Meshlium can work as a Mesh node. This means we can interconnect several Meshliums in order to share a common resource as an Internet connection. This way, the clients connected to a certain node can access to the Internet connection of a third node which is some hops far away the actual connecting point. Wasmotes with GPRS, GPRS+GPS, 3G or WiFi can send sensor data through the access point or through the Internet via HTTP protocol. In a mesh network there are two kinds of nodes:

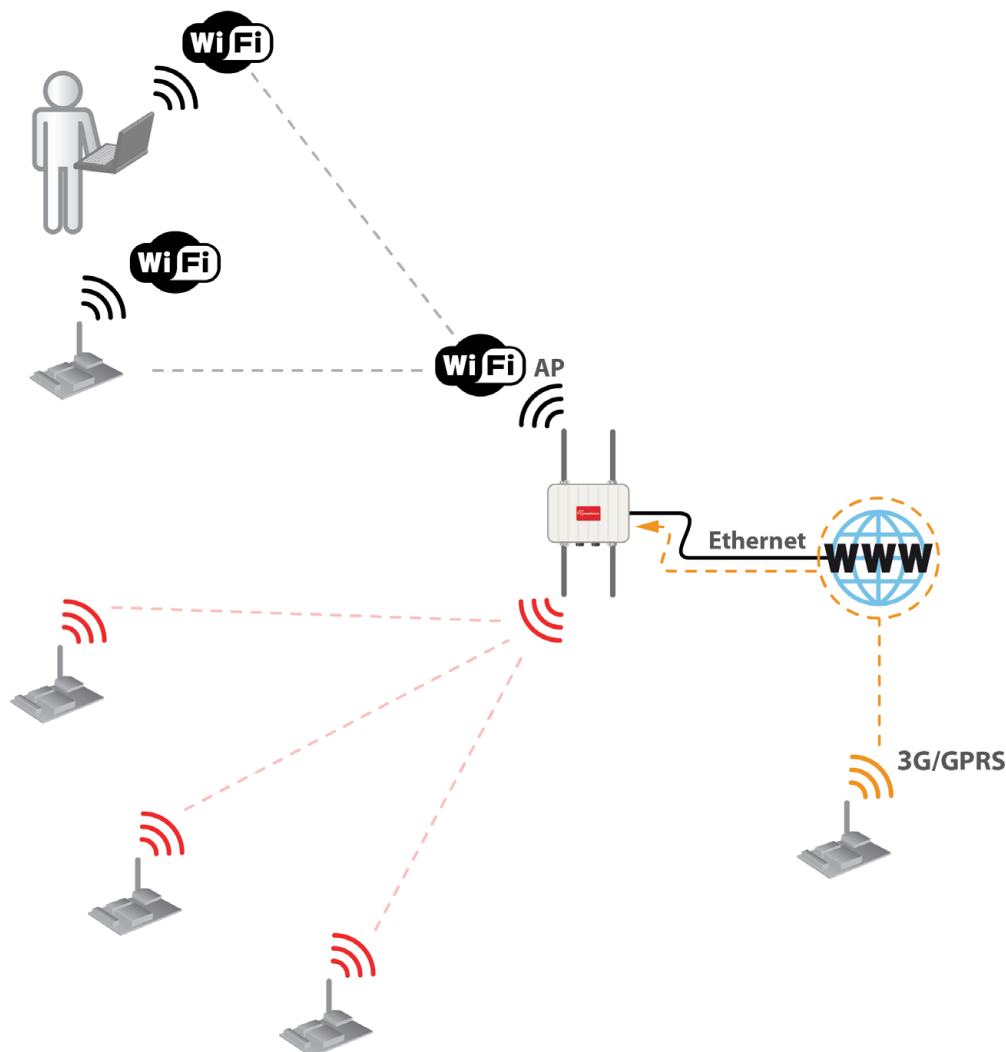
- **Meshlium Mesh AP** allows clients to connect via Wifi and creates links with other nodes using a second Wifi radio which operates in the 5GHz band.
- **Meshlium Mesh 3G/GPRS AP** (3G/GPRS gateway) is the node which shares its Internet connection with the rest of the network. It takes the Internet connection from the 3G/GPRS interface. There is only one GW in the mesh network.



Meshlium RF AP

Meshlium can take the sensor data which comes from a Wireless Sensor Network (WSN) made with Wasp mote sensor devices * and send it to the Internet using the Ethernet interface. Wasp motes with GPRS, GPRS+GPS, 3G or WiFi can send sensor info through the access point or through the Internet via HTTP protocol. Users can also connect directly to Meshlium using the Wifi interface to control it and access to the sensor data.

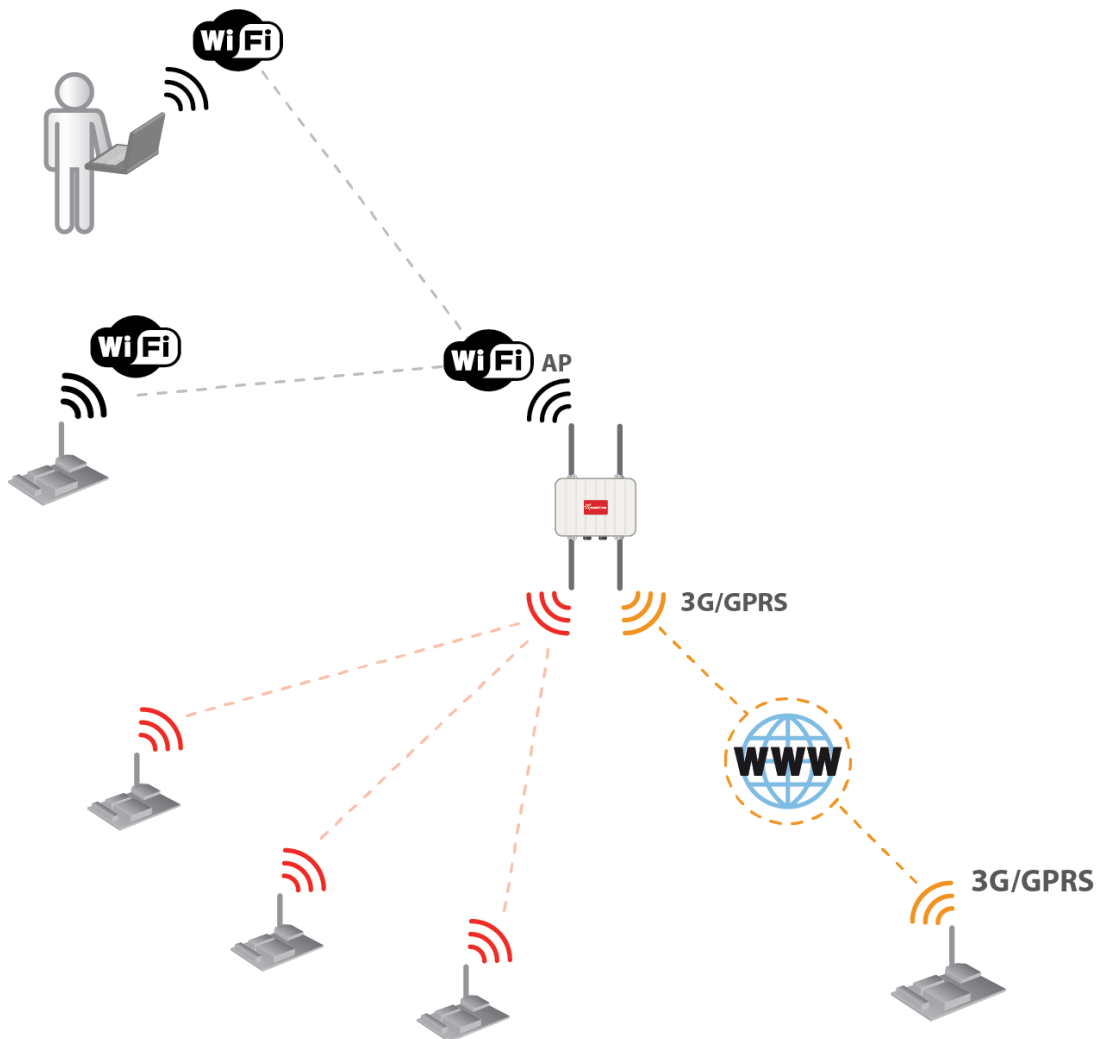
(*) <http://www.libelium.com/waspote>



Meshlium ZigBee 3G/GPRS AP

Meshlium can take the sensor data which comes from a Wireless Sensor Network (WSN) made with Waspote sensor devices *and send it to the Internet using the 3G/GPRS interface. Waspotes with GPRS, GPRS+GPS, 3G or WiFi can send sensor info through the access point or through the Internet via HTTP protocol. Users can also connect directly to Meshlium using the Wifi interface to control it and access to the sensor data.

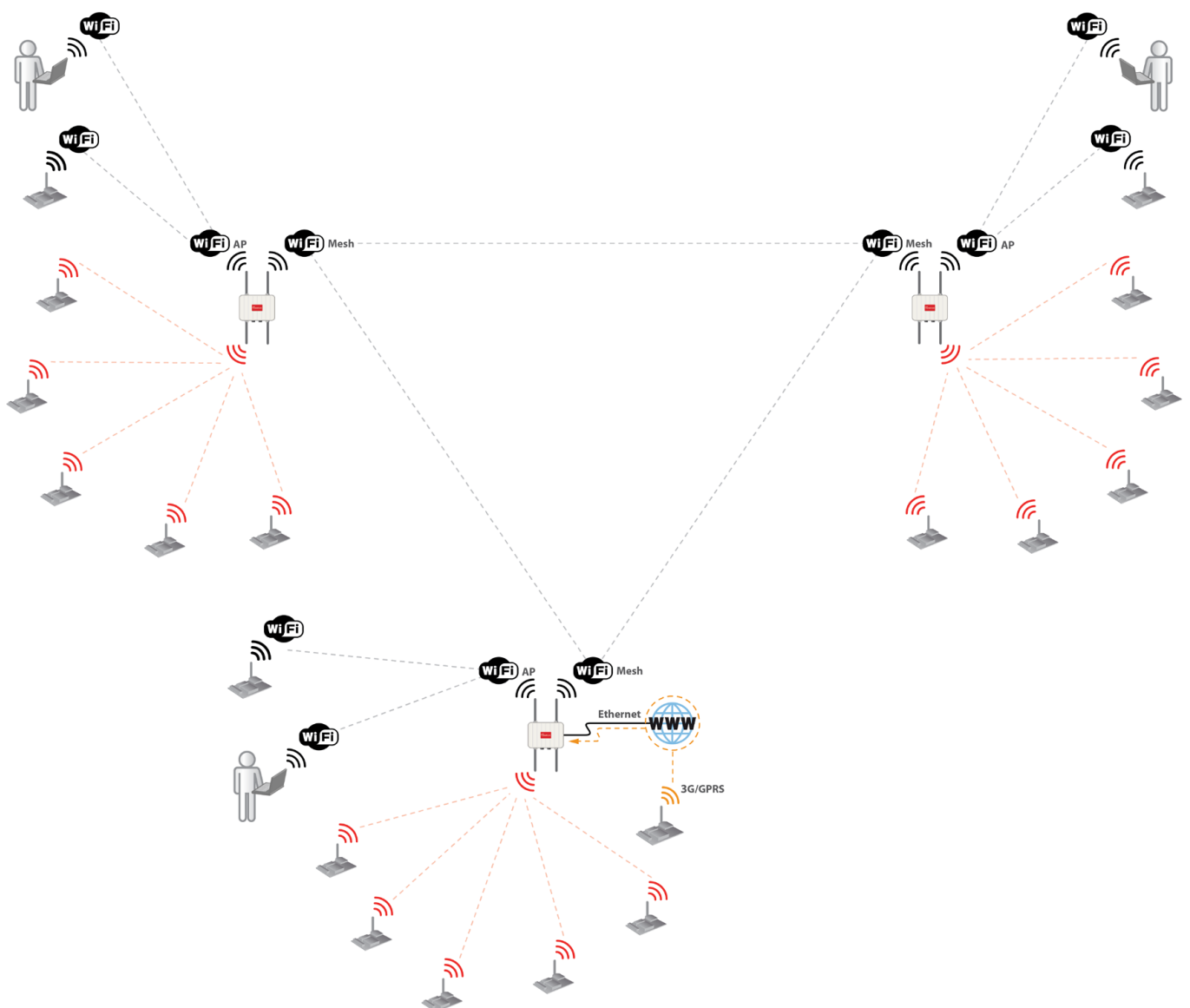
(*) <http://www.libelium.com/waspote>



Meshlium XBee Mesh AP - Meshlium XBee Mesh AP GW:

Meshlium can work as a XBee Mesh node. This means we can interconnect several nodes in order to share a common resource as an Internet connection. This way, the sensor nodes connected to a node via XBee can send the information to the Internet link set on a third node which is some hops far away the actual point. In this hybrid XBee - Wifi mesh network there are two kinds of nodes:

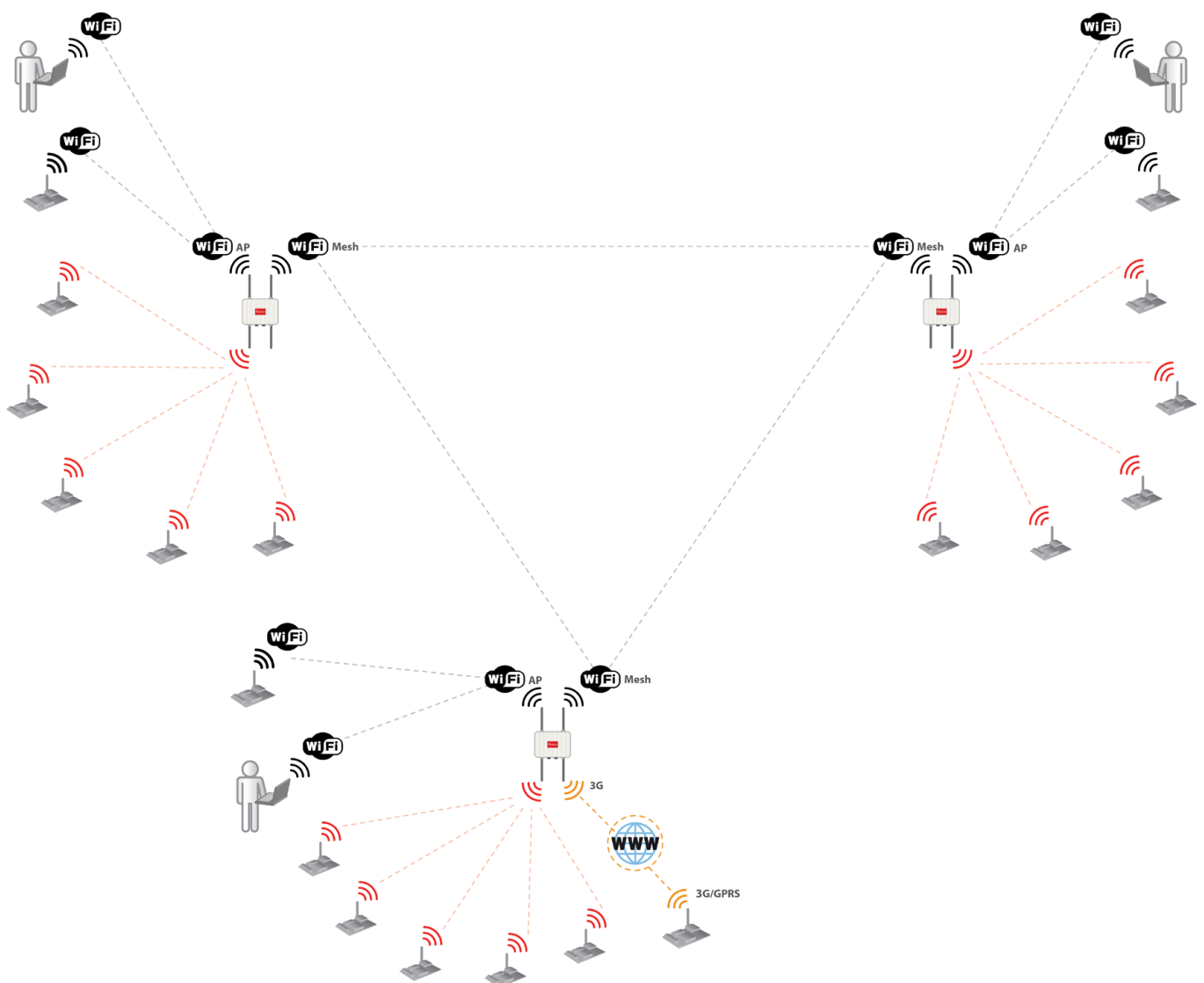
- **Meshlium XBee Mesh AP** allows the sensor devices to connect via XBee, WiFi or Internet (Ethernet) and creates links with other nodes using a Wifi radio which operates in the 5GHz band. Users can also connect directly to Meshlium using the 2.4GHz Wifi interface to control it and access to the sensor data.
- **Meshlium XBee Mesh AP GW** (gateway) is the node which shares its Internet connection with the rest of the network. It takes the Internet connection from the Ethernet interface. There is only one GW in the mesh network.



Meshlium XBee Mesh AP - Meshlium XBee Mesh 3G/GPRS AP

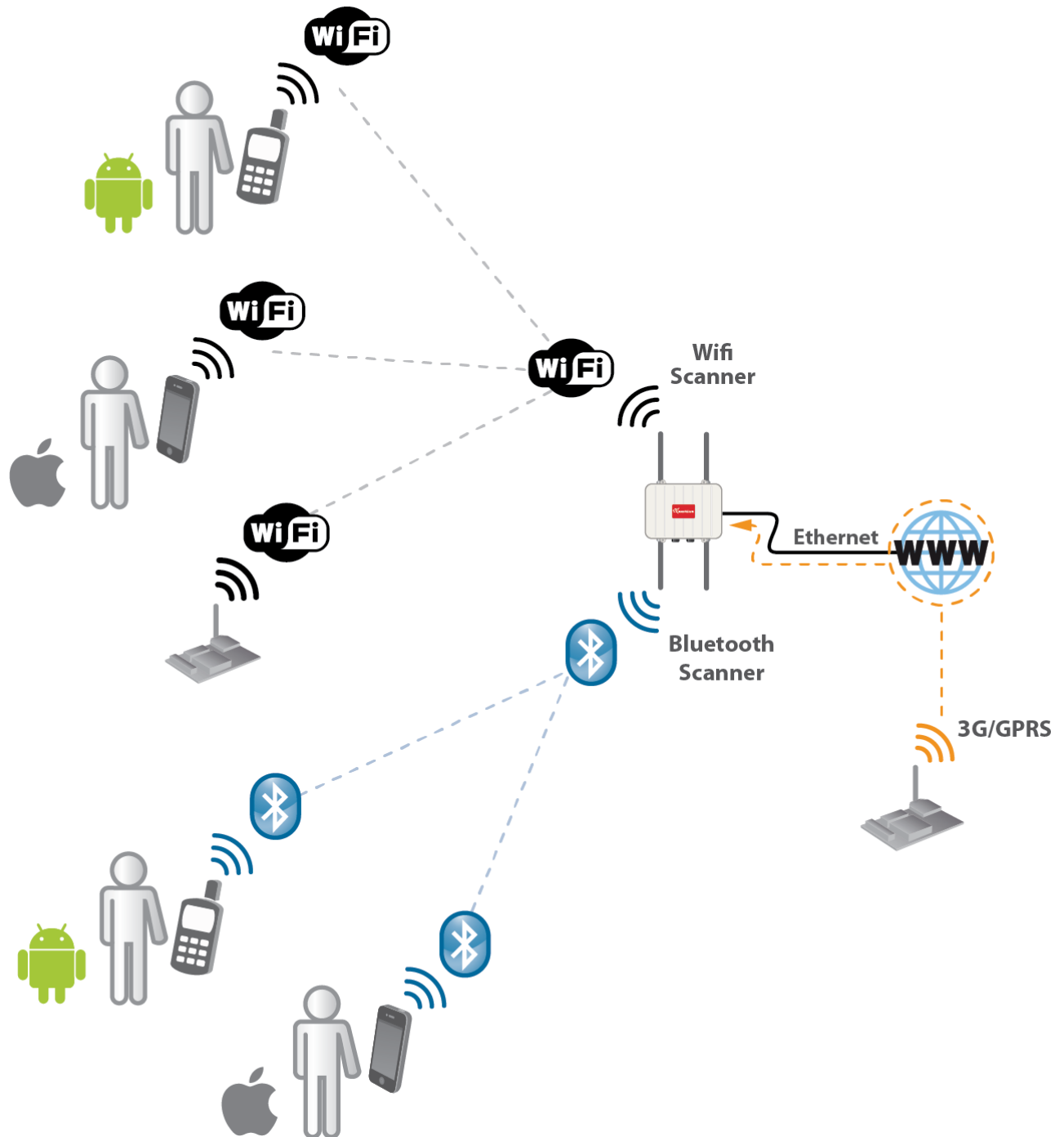
Meshlium can work as a ZigBee Mesh node. This means we can interconnect several nodes in order to share a common resource as an Internet connection. This way, the sensor nodes connected to a node via XBee can send the information to the Internet link set on a third node which is some hops far away the actual point. In this hybrid XBee - Wifi mesh network there are two kinds of nodes:

- **Meshlium XBee Mesh AP** allows the sensor devices to connect via XBee, WiFi or Internet (Ethernet) and creates links with other nodes using a Wifi radio which operates in the 5GHz band. Users can also connect directly to Meshlium using the 2.4GHz Wifi interface to control it and access to the sensor data.
- **Meshlium XBee Mesh 3G/GPRS AP** (3G/GPRS gateway) is the node which shares its Internet connection with the rest of the network. It takes the Internet connection from the 3G/GPRS interface. There is only one GW in the mesh network.



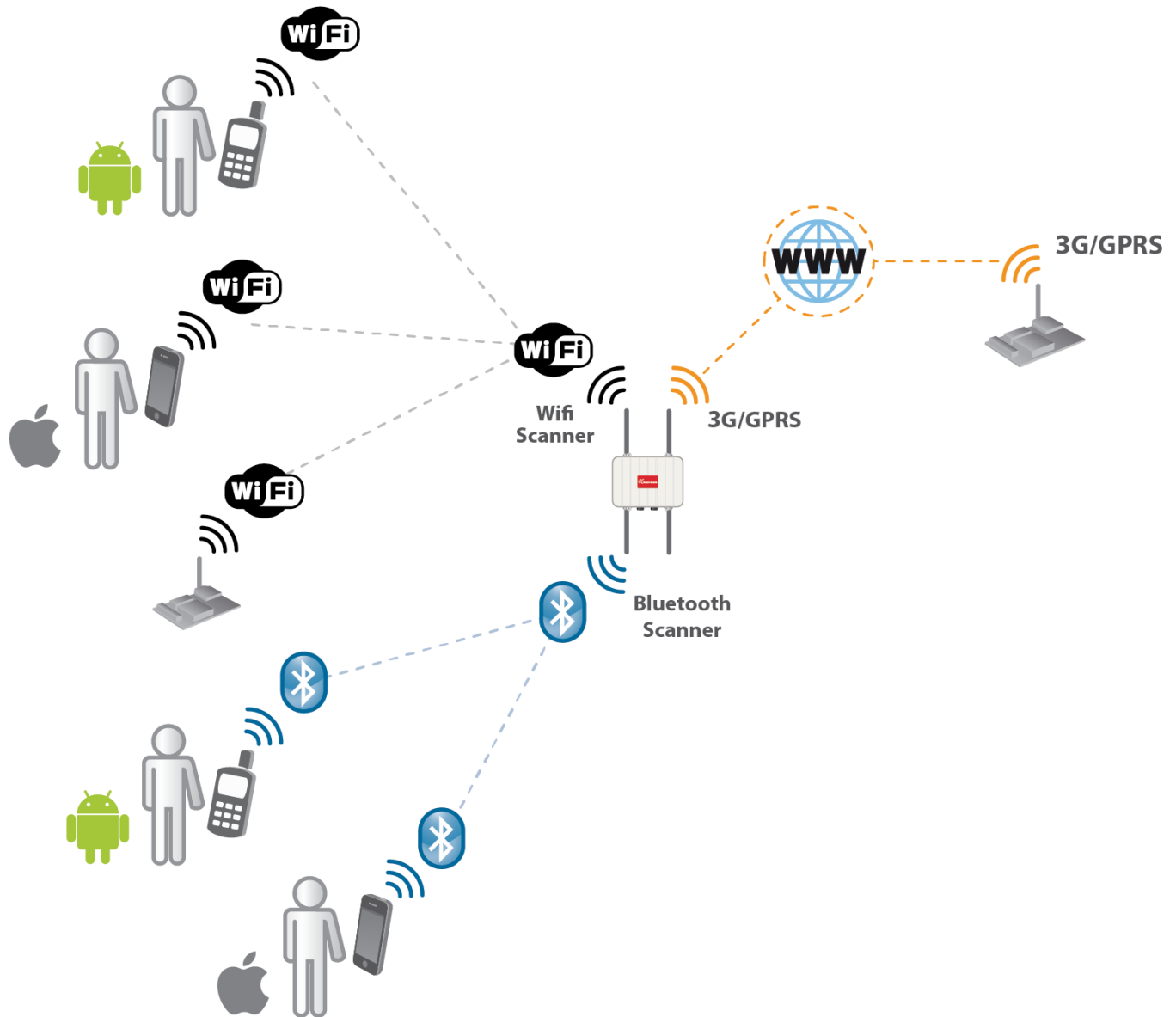
Meshlium Scanner AP

It allows to detect Smartphones (iPhone, Android) and in general any device which works with **Wifi** or **Bluetooth** interfaces. The collected data can be send to the Internet by using the Ethernet. Clients can also connect to Meshlium via Wifi with laptops and smart phones and get access to the Internet (as a common Access Point). Wasmotes with GPRS, GPRS+GPS, 3G or WiFi can send sensor info through the access point or through the Internet via HTTP protocol.



Meshlium Scanner 3G/GPRS-AP

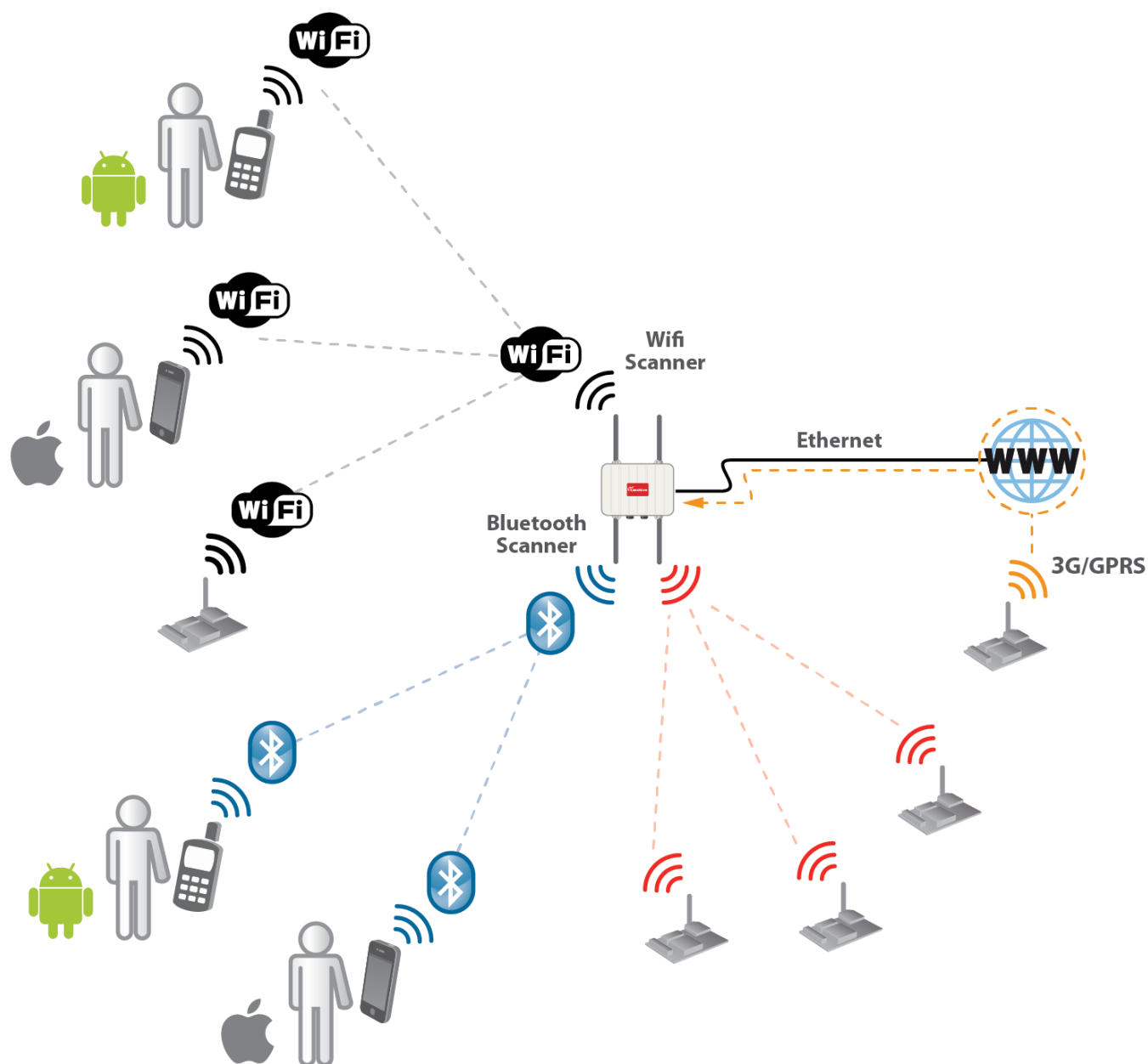
It allows to detect Smartphones (iPhone, Android) and in general any device which works with **Wifi** or **Bluetooth** interfaces. The collected data can be send to the Internet by using the Ethernet, and 3G/GPRS connectivity. Clients can also connect to Meshlium via Wifi with laptops and smart phones and get access to the Internet (as a common Access Point). Wasmotes with GPRS, GPRS+GPS, 3G or WiFi can send sensor info through the access point or through the Internet via HTTP protocol.



Meshlium Scanner RF-AP

It allows to detect Smartphones (iPhone, Android) and in general any device which works with **Wifi** or **Bluetooth** interfaces. It can also capture the sensor data which comes from the Wireless Sensor Network (WSN) made with Waspmote sensor devices.

The collected data can be send to the Internet by using the Ethernet and Wifi connectivity. Clients can also connect to Meshlium via Wifi with laptops and smart phones and get access to the Internet (as a common Access Point).

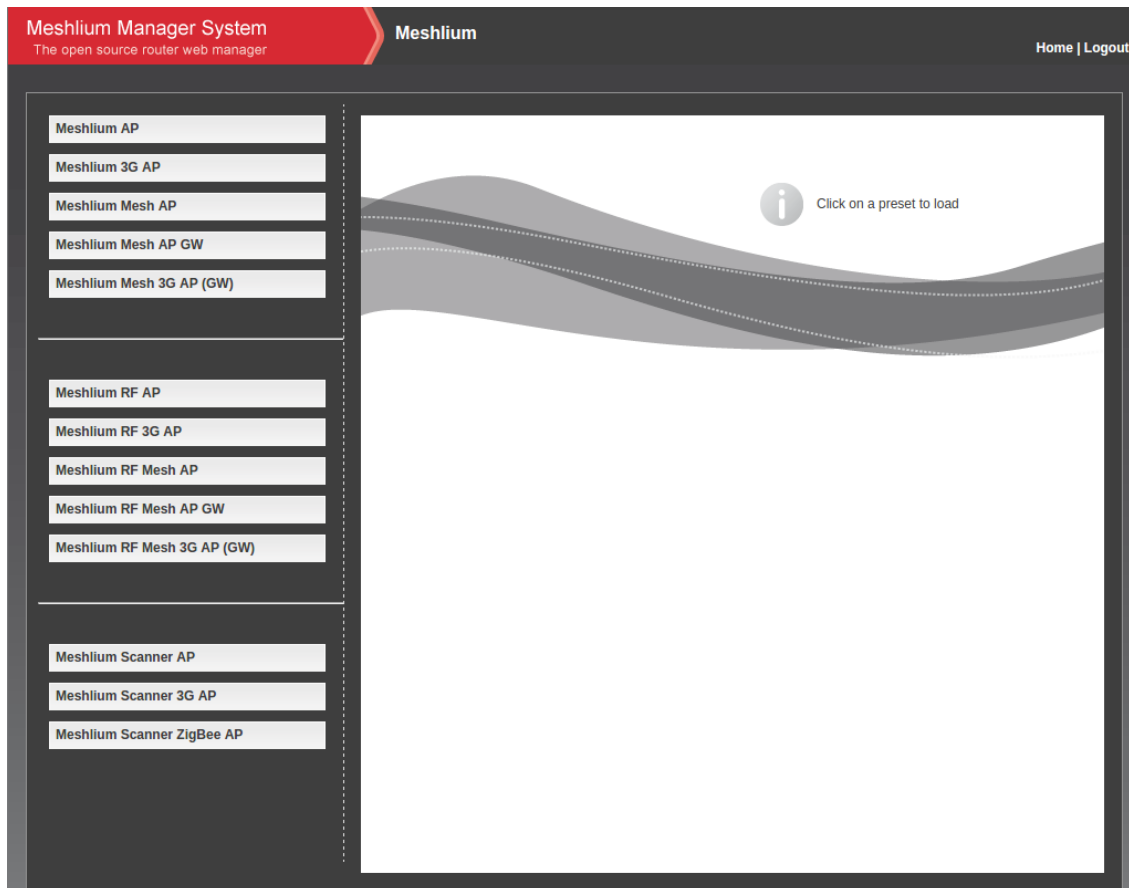


8. Meshlium Presets

8.1. Presets

The Presets section allows us to set a certain node type as presented in the Meshlium Models section. **This means you can change quickly from one setting to another just pressing one button.**

The different Meshlium configurations you can set depends on the hardware and if they have a GW roll or not.



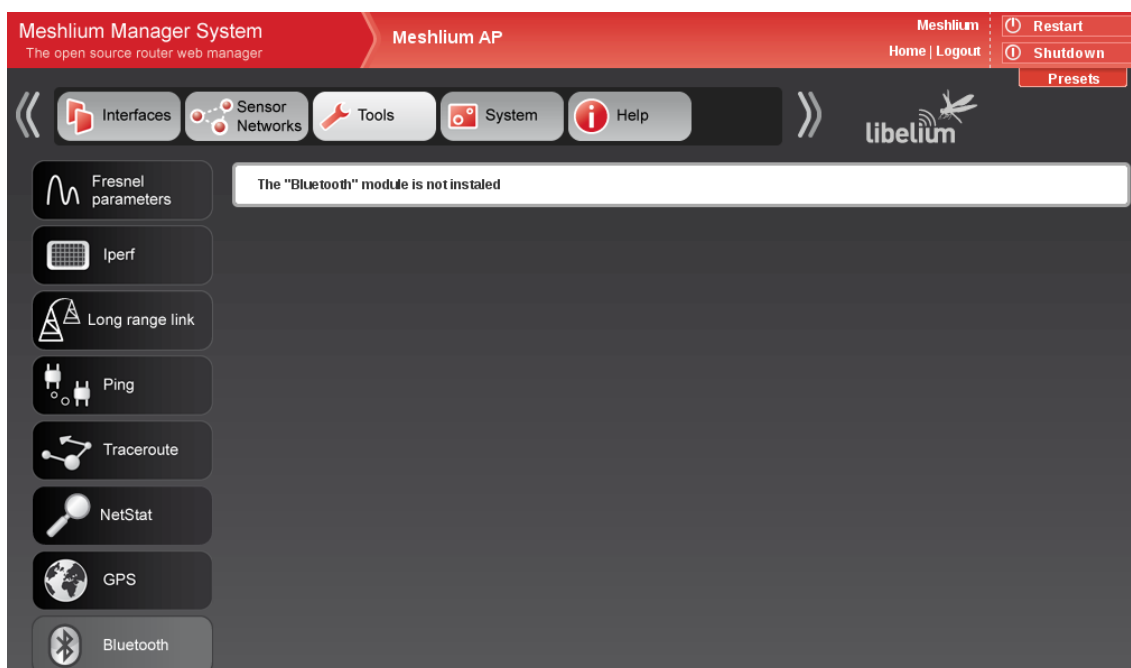
All the configurations can act as GW nodes except the nodes configured to act a simple mesh nodes (Mesh-AP and XBee-Mesh-AP) which need one GW node in the mesh to get the information out to the Internet. Anyway all the nodes can be changed to act as GW nodes in the Preset section.

Once we know the Preset you want to use just press the "Set Preset" button and restart the machine with the "Restart" button (upper right side). The system will start now with the factory default options for this Preset. Go to the "Accessing Meshlium" chapter in order to see how to connect to Meshlium.

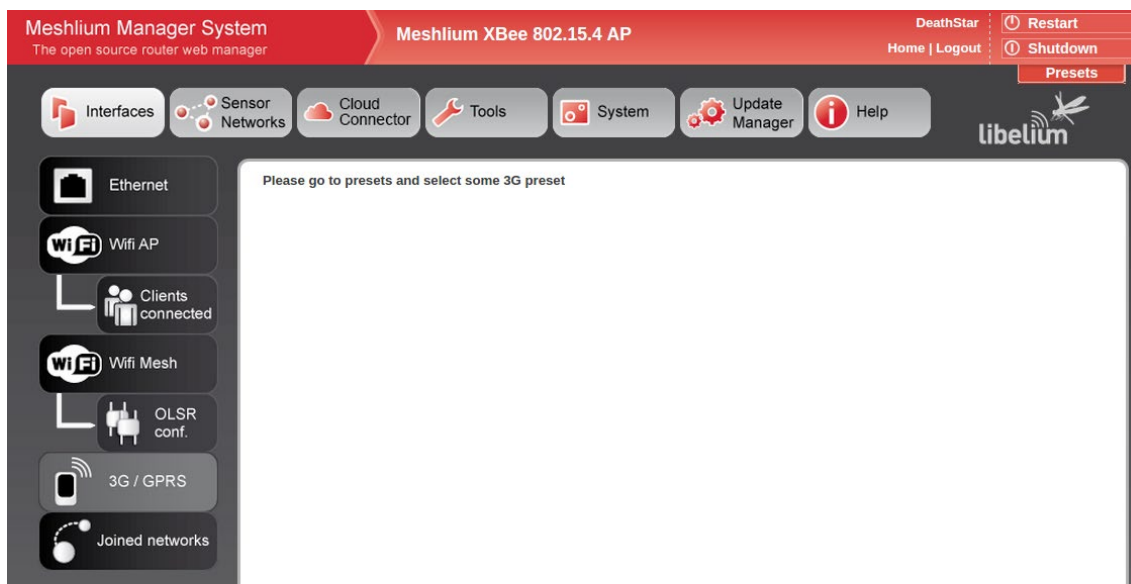
If we have selected a 3G/GPRS Preset the Manager System will take you to the 3G/GPRS Interface configuration screen so that you can configure the parameters which will be used in order to connect to the carrier each time Meshlium starts.

Before restarting Meshlium you can access to the Interface sections in order to change essential parameters such as the Ethernet IP (if we want to use the static mode and not DHCP).

On the one hand if we try to access to an Interface such as RF module (XBee / LoRa) or 3G/GPRS or to a module which is not physically installed, we get the message "The XXX module is not installed".



On the other hand if we try to access to an Interface which is physically installed but it is not internally activated in the current preset the next message will be shown: "Please go to presets and select the right one".



8.2. Validation of Changes in Interfaces

After changing a parameter in any Interface (Ethernet, Wifi AP, Wifi Mesh, 3G/GPRS) and once Meshlium is restarted you have to **validate** the changes done in the next **5 minutes**.

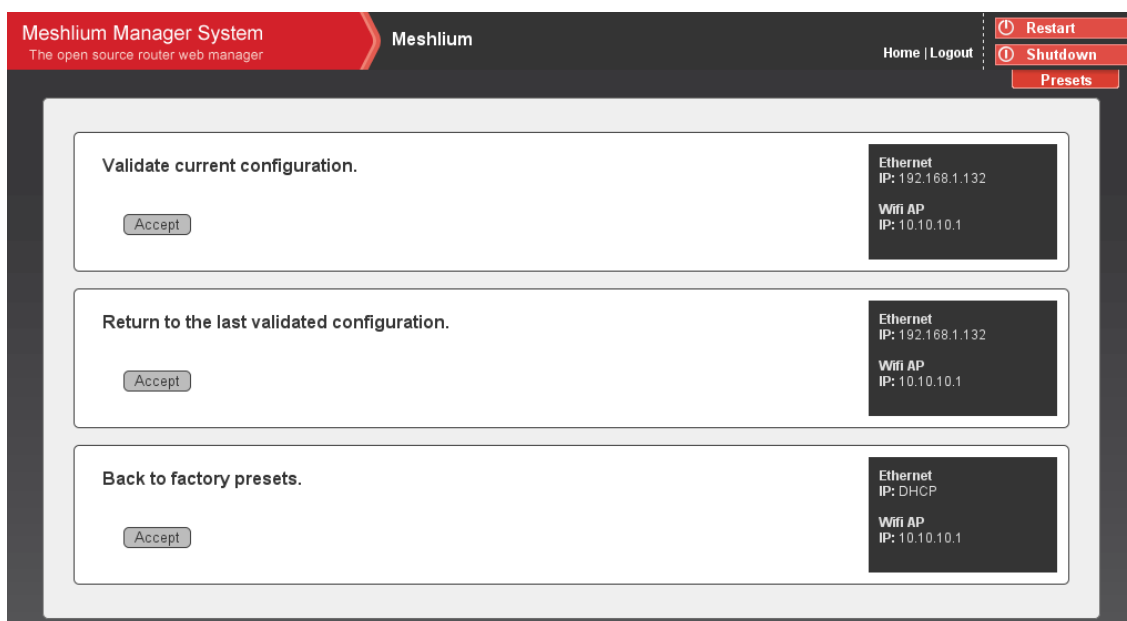
This is a security mechanism in order to make sure that the user is still able to communicate with Meshlium after changes are performed.

If the changes are not validated, Meshlium will set the factory defaults and the user will be able to connect again as described in the “Accessing Meshlium” chapter. If the connection fails just keep Meshlium ON for five minutes and try to access as pointed in the “Accessing Meshlium” chapter.

If your settings were successful, you will get into the Validation Screen where you have three different options:

- Validate current configuration
- Return to the last validated configuration
- Back to factory presets

Choose the option desired to save the changes permanently.



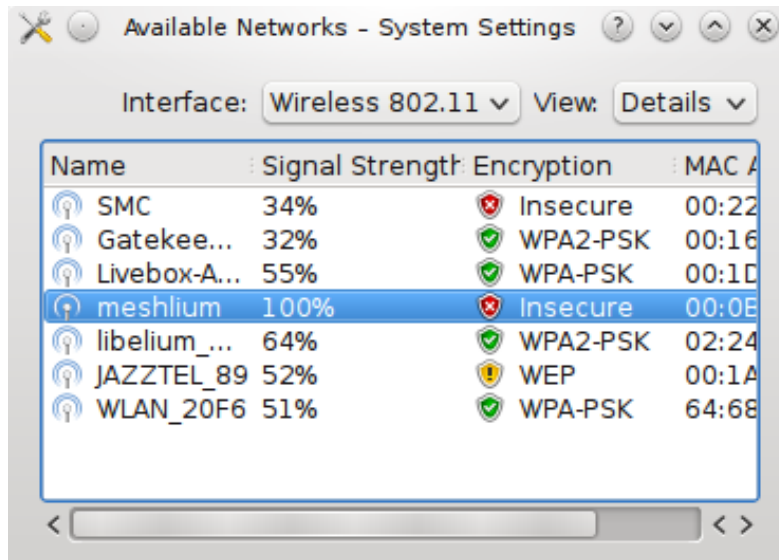
If you can not access to the Manager System due to a bad Interface configuration just keep Meshlium 5 minutes running and then restart. The system will start with the factory default settings so you will be able to access as specified in the “Accessing Meshlium” chapter.

9. Accessing Meshlium - Make it Easy!

Meshlium comes with all the radios ready to be used. Just “plug & mesh!”. All the Meshlium nodes come with the Wifi AP ready so that users can connect using their Wifi devices. Connect the Ethernet cable to your network hub, restart Meshlium and it will automatically get an IP from your network using DHCP*.

(*) For the Meshlium Mesh AP and for the Meshlium XBee Mesh AP the Internet connection depends on the GW of the network.

Then access Meshlium through the Wifi connection. First of all, search the available access points and connect to “Meshlium”.

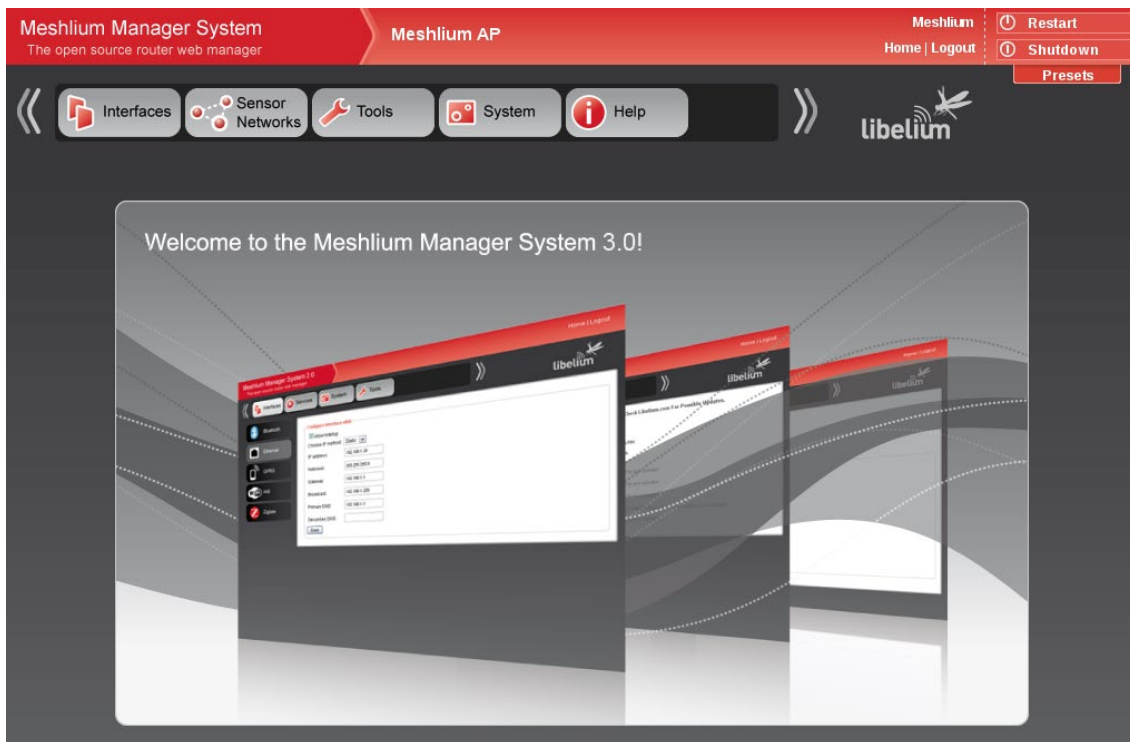


No password is needed as the network is public (you can change it later in the Wifi AP Interface options). When you select it, Meshlium will give an IP from the range 10.10.10.10 - 10.10.10.250.

Now you can open your browser and access to the Meshlium Manager System:

- **URL:** http://10.10.10.1/ManagerSystem
- **user:** root
- **password:** libelium





If your network does not offer DHCP service Meshlium starts with a default IP (192.168.1.100). In this case you can connect Meshlium through the Wifi connection (which is always available) or with the crossover cable provided with Meshlium.

If you want to access to the Manager System using the crossover ethernet cable go to:

- **URL:** <http://192.168.1.100/ManagerSystem>
- **user:** root
- **password:** libelium

10. Ethernet

10.1. Configuration

By default Meshlium comes with the Ethernet interface activated to get dynamically the IP using the DHCP service. If the case a static configuration is required the next parameters can be configured:

Ethernet Network

Choose IP method: Static

IP address: 192.168.1.210

Netmask address: 255.255.255.0

Gateway: 192.168.1.1

Broadcast: 192.168.1.255

Primary DNS: 8.8.8.8

Secondary DNS: 8.8.4.4

Use IPv6: ☒ Generate IPv6 Address

IPv6 address: 2001::20d:b9ff:fe26:b620

Netmask number: 64

Gateway: 2001::1

Save

© Libelium Comunicaciones Distribuidas S.L. | [Terms of use](#)

Ethernet Network

Choose IP method: Static

IP address: 192.168.1.210

Netmask address: 255.255.255.0

Gateway: 192.168.1.1

Broadcast: 192.168.1.255

Primary DNS: 8.8.8.8

Secondary DNS: 8.8.4.4

Use IPv6: ☐

Save

You can also use IPv6 (Internet Protocol version 6) by setting the check box "Use IPv6". IPv6 is a version of the Internet Protocol (IP) intended to succeed IPv4. The next parameters can be configured:

Ethernet Network

Choose IP method	Static ▼
IP address	192.168.1.210
Netmask address	255.255.255.0
Gateway	192.168.1.1
Broadcast	192.168.1.255
Primary DNS	8.8.8.8
Secondary DNS	8.8.4.4
Use IPv6	<input checked="" type="checkbox"/> Generate IPv6 Address
IPv6 address	2001::20d:b9ff:fe26:b620
Netmask number	64
Gateway	2001::1

Save

In many cases, IPv6 addresses are composed of two logical parts: a prefix of 64-bit (2001::) and a 64 bit part that is generated automatically from the MAC address of the interface.

The button "Generate IPv6 address" performs this task.

After saving the new options and once you have restarted Meshlium you have to validate the new configuration before the next 5 minutes, if not, the factory default configuration will be restored to avoid leaving Meshlium without connectivity. More info see the "Default Interfaces" section.

To check IPv6 configuration, after save and restart Meshlium, go to Tools -> Ping. Select Ethernet (IPv6), by default ipv6.google.com appears as destination host.

Ping

Select interface	Ethernet (IPv6) ▼
Destination Host	ipv6.google.com Do Test

If your ISP doesn't support external IPv6 addresses yet, you can change it to a local address.

Ping

Select interface

Destination Host

Then press "Do Test". If something like next image appears, you have IPv6 correctly configured.

Ping

Select interface

Destination Host

```
Launching: ping6 '2001::20d:b9ff:fe26:b620' -c 10 -I eth0

PING 2001::20d:b9ff:fe26:b620(2001::20d:b9ff:fe26:b620) from 2001::20d:b9ff:fe26:b620 :
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=1 ttl=64 time=0.096 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=2 ttl=64 time=0.099 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=3 ttl=64 time=0.098 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=4 ttl=64 time=0.107 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=5 ttl=64 time=0.107 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=6 ttl=64 time=0.107 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=7 ttl=64 time=0.107 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=8 ttl=64 time=0.108 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=9 ttl=64 time=0.106 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=10 ttl=64 time=0.109 ms

--- 2001::20d:b9ff:fe26:b620 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9002ms
rtt min/avg/max/mdev = 0.096/0.104/0.109/0.010 ms

ping finished.
```

11. Wifi AP

By default Meshlium comes with a 2.4GHz Wifi interface ready to work as an Access Point (AP). It also comes with the DHCP Service activated so when users connect through the Wifi connection will get automatically an IP from the range 10.10.10.10 - 10.10.10.250.

Meshlium Manager System
The open source router web manager

Meshlium Zigbee Mesh 3G AP

meshlium Home | Logout

Restart Shutdown Presets

Interfaces Sensor Networks Tools System Update Manager Help

libelium

Wifi AP Network

Address: 10.1.20.1 DHCP start ip address: 10.10.10.10
 Netmask: 255.255.255.0 DHCP end ip address: 10.10.10.250
 Broadcast: 10.10.10.255 DHCP expire time: 24 hours
 Primary DNS: 8.8.8.8
 Secondary DNS: 8.8.4.4

Radio

ESSID: meshliumMisco Hide? ☐
 Channel: 6
 Protocol: g
 Tx power: auto
 Rate: auto

Security

Protocol: WPA
 Password:
 Confirm password:
 *8 to 63 characters

Save

© Libelium Comunicaciones Distribuidas S.L. | Terms of use

11.1. Configuration

There are three sections in the configuration screen: Network, Radio, Security.

Network:

Here you can change the common Network options and the DHCP IP range which is offered to the clients. Remember that if you change the IP of the AP you should also change the DHCP options in order to make all them work in the same network.

Wifi AP Network

Address	<input type="text" value="10.10.10.1"/>	DHCP start ip address	<input type="text" value="10.10.10.2"/>
Netmask	<input type="text" value="255.255.255.0"/>	DHCP end ip address	<input type="text" value="10.10.10.254"/>
Broadcast	<input type="text" value="10.10.10.255"/>	DHCP expire time	<input type="text" value="24"/> hours
Primary DNS	<input type="text" value="8.8.8.8"/>		
Secondary DNS	<input type="text" value="8.8.8.4"/>		

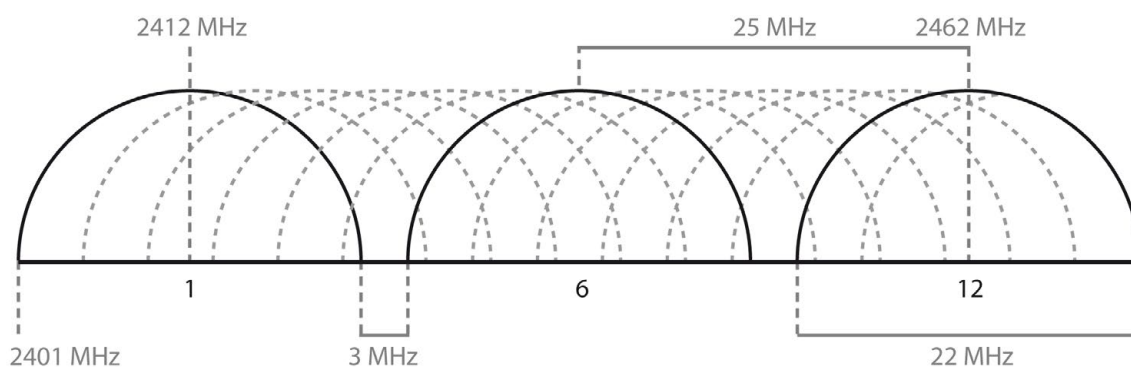
Radio:

These are specific Wifi parameters. You can change the public name of the AP which is offered to the clients -ESSID-, by default is set "Meshlium". You can also make it invisible so that only users who know it can connect.

Radio

ESSID	Meshlium	Hide? <input type="checkbox"/>
Channel	6	
Protocol	g	
Tx power	auto	
Rate	auto	
Fragmentation		[256-2346] or off

You can also change the radio channel which is being used for the transmissions, according to the next diagram:



The Protocols that can be used are 802.11b for long links and 802.11g for high bandwidth links. Transmission Power, Rate and Fragmentation can also be controlled for experts users. Do not change them if you don't know exactly what you are doing.

Security:

You can activate encryption in the communication using the WEP and WPA protocols.

WEP is enabled in the 5 and 13 characters configurations while WPA-PSK can be used with a password from 8 to 63 characters. We recommend use WPA in order to get the a major security in the network.

Security

Protocol:

Key size:

Password:

Confirm password:

*13 characters

Protocol:

Password:

Confirm password:

*8 to 63 characters

After saving the new options and once you have restarted Meshlium you have to validate the new configuration before the next 5 minutes, if not, the factory default configuration will be restored to avoid leaving Meshlium without connectivity. More info see the "Default Interfaces" section.

11.2. Clients connected

In this section you can see the MAC address of clients along with the IP assigned by the AP. It is a quick way to know how many clients are connected and who they are.

Time	MAC address	IP address	Hostname
Thu Nov 11 17:55:54 GMT 2010	1c:4b:d6:ba:be:15	10.10.10.10	*

12. Wifi Mesh

The Wifi Mesh Interface enables dual band networks as it can be configured to work in the 2.4GHz or in the 5GHz in order to avoid interferences with common Wifi networks. The mesh protocol used is Optimized Link State Routing (OLSR), an open source protocol which allows us to control if a certain node has to work as a fixed node or as a mobile one.

The antennas used are also dual band so you can change the frequency dynamically without changing the antennas attached to this radio.

The screenshot shows the Meshlium Manager System web interface. The top navigation bar includes 'Meshlium Manager System', 'Meshlium XBee 802.15.4 AP', and user controls like 'DeathStar', 'Home | Logout', 'Restart', 'Shutdown', and 'Presets'. A secondary navigation bar contains icons for 'Interfaces', 'Sensor Networks', 'Cloud Connector', 'Tools', 'System', 'Update Manager', and 'Help'. A left sidebar lists 'Ethernet', 'Wifi AP', 'Clients connected', 'Wifi Mesh' (selected), 'OLSR conf.', '3G / GPRS', and 'Joined networks'. The main content area is titled 'Wifi Mesh Network' and contains two sections: 'Wifi Mesh Network' with fields for Address (10.1.20.1), Netmask (255.255.255.0), Broadcast (10.10.11.255), Primary DNS (8.8.8.8), and Secondary DNS (8.8.4.4); and 'Radio' with fields for ESSID (meshlium-X), CELL ID (AA:AA:AA:AA:AA), Frequency (5GHz), Channel (64), Tx power (auto), and Rate (auto). A 'Save' button is at the bottom right of the form. The footer indicates '© Libelium Comunicaciones Distribuidas S.L. | Terms of use'.

12.1. Configuration

By default, when you buy Meshlium with the Wifi Mesh radio included it comes already configured to create the mesh network. This means that when you buy several Meshlium, one comes preconfigured with the "Mesh Gateway" preset and the rest with the normal "Mesh" preset.

Just connect the router labeled as GW to your Ethernet connection and it will share the Internet access to the rest of the neighbour nodes.

In order to distinguish between them they come with different ESSID's for the Wifi AP Interface. The GW has "Meshlium AP1" and the rest come with the form "Meshlium AP2", "Meshlium AP3", etc.

If you want to do it manually:

To set a Mesh network you need several nodes with a Meshlium Mesh AP or Meshlium XBee Mesh AP preset and one with a Meshlium Mesh AP GW or Meshlium Mesh AP 3G/GPRS preset *.

* More info about the different roles in the "Presets" section.

Once you have chosen the preset you only need to give a valid IP in order to get this node communicating with the rest of the neighbours of the mesh network. The IP can be changed in the main section. Just set one IP which is not being used by any node and the current node will be able to join automatically to the mesh network. **Just “plug & mesh!”**

By default all the nodes with a Mesh preset come with the same ESSID and CELL ID. It is important all the nodes have this two fields with the same data. On the one hand the ESSID will allow the nodes to share a common ID in order to be detected as the same mesh network. On the other hand the CELL ID lets the nodes to act with a unique virtual MAC address, what facilitates the connections between the nodes in the network.

In the frequency field you can choose the band in which you want the mesh network to be created: 2.4GHz or 5GHz. As the antennas included for the Wifi Mesh radio are dual you don't need to change anything in the hardware configuration.

After saving the new options and once you have restarted Meshlium you have to validate the new configuration before the next 5 minutes, if not, the factory default configuration will be restored to avoid leaving Meshlium without connectivity. More info see the “Default Interfaces” section.

Wifi Mesh Network

Address	<input type="text" value="10.1.11.1"/>
Netmask	<input type="text" value="255.255.255.0"/>
Broadcast	<input type="text" value="10.1.11.255"/>
Primary DNS	<input type="text" value="8.8.8.8"/>
Secondary DNS	<input type="text" value="8.8.8.4"/>

Radio

ESSID	<input type="text" value="Meshlium-X"/>
CELL ID	<input type="text" value="AA:AA:AA:AA:AA:AA"/>
Frequency	<input type="text" value="5GHz"/>
Channel	<input type="text" value="64"/>
Tx power	<input type="text" value="auto"/>
Rate	<input type="text" value="auto"/>

12.2. OLSR Options

In this section you can set if the current node is the **GW** of the mesh network. If you check the “**Share Internet Connection**” box it will offer its Internet connection to the rest of the brothers in the network. **Remember that only one node in the mesh network can be working as GW.**

The rest of the parameters allow us to change the time intervals used by the mesh daemon in order to send topology information to the neighbours.

The “Hello Interval” and “Validity Time” set how often this node will be broadcasting information about its presence to the rest of the nodes “hello!”.

The “TC Interval” and “Validity Time” set how often this node will be broadcasting information about its connections and topology to the rest of the nodes.

The “Hna Interval” and “Validity Time” set how often this node will be broadcasting that is sharing its Internet connection. This parameter is only used by the GW node.

In order to make this process easier we have created two different roles: **fixed node** and **mobile node** which can be set just pressing each button.

OLSR configuration

[Access OLSR summary](#)

OLSR parameters

☒ Share the internet connection (make this node as mesh gateway)

			Fixed	Mobile
HelloInterval	<input type="text" value="1.0"/>	[0.0-]	5.0	1.0
HelloValidityTime	<input type="text" value="10.0"/>	[0.0-]	100.0	20.0
TcInterval	<input type="text" value="1.0"/>	[0.0-]	5.0	1.0
TcValidityTime	<input type="text" value="10.0"/>	[0.0-]	100.0	20.0
HnaInterval	<input type="text" value="2.0"/>	[0.0-]	5.0	2.0
HnaValidityTime	<input type="text" value="50.0"/>	[0.0-]	200.0	50.0

In the “OLSR Summary” section you can see direct information of the mesh daemon which is running inside Meshlium.

The Links/Topology section shows the direct brothers and the two hop neighbours along with the “next hop” node which will be accessed in order to reach these nodes.

Configuration	Routes	Links/Topology	All	About
Links				
Local IP	Remote IP		Hysteresis	
10.10.11.2	10.10.11.1		0.00	
Neighbors				
IP Address	SYM	MPR	MPRS	Willingness
10.10.11.1	YES	YES	YES	7
2 Hop Neighbors				IP ADDRESS (0)
Topology Entries				
Destination IP				Last Hop IP
10.10.11.2				10.10.11.1
10.10.11.1				10.10.11.2
MID Entries				
Main Address	Aliases			

The “Routes” shows what networks can be accessed regarding the information passed by the rest of the nodes. In the image below we can see as the node 10.10.11.1 is offering access to the Internet (0.0.0.0/0). This means this node (10.10.11.1) is acting as GW of the Mesh network.

Configuration	Routes	Links/Topology	All	About
OLSR Routes in Kernel				
Destination	Gateway	Metric	ETX	Interface
0.0.0.0/0	10.10.11.1	1	0.000	ath1
10.10.11.1	10.10.11.1	1	0.000	ath1

12.3. The Gateway Node

In a mesh network there is a node which shares its Internet connection to the rest of the neighbours. It is called the Gateway (GW).

There is only one GW in the mesh network and it has to be specified in the OLSR configuration (as pointed before) checking the "Share Internet Connection" option. You can also choose the Mesh Gateway Preset and change the Wifi Mesh IP to one not used in the network.

The Internet connection can be shared from the Ethernet cable or from the 3G/GPRS link. For this reason there are two kinds of Mesh GW:

- Meshlium Mesh AP GW -> from Ethernet
- Meshlium Mesh 3G/GPRS AP -> from 3G/GPRS

For more information about how to change the roll of a Mesh node into a Mesh Gateway node go to the "Presets section".

13. 3G/GPRS

13.1. Configuration

Here you can set the parameters of the mobile operator you have chosen. There is a list with some initial configurations depending on the country and the operator *. However, this list may not be updated with the last valid configuration of your mobile provider. Ask your mobile company for the information required to connect (APN, Username, Password) and add the PIN code of the SIM card used (leave empty if there is no PIN). We recommend to disable the PIN in the SIM card as this will make easier the test and validation process and will avoid to block the SIM card.

* You can find more information about the 3G/GPRS settings listed by country at: http://www.unlocks.co.uk/GPRS_settings.php

3G Network

Connectivity information from operators provided without warranty.

Select country

Choose operator

[Click here](#) to edit

Card PIN Leave it empty for no PIN

APN If more than one init is required you should edit wvdial.conf manually .

Username Should be provided by your operator.

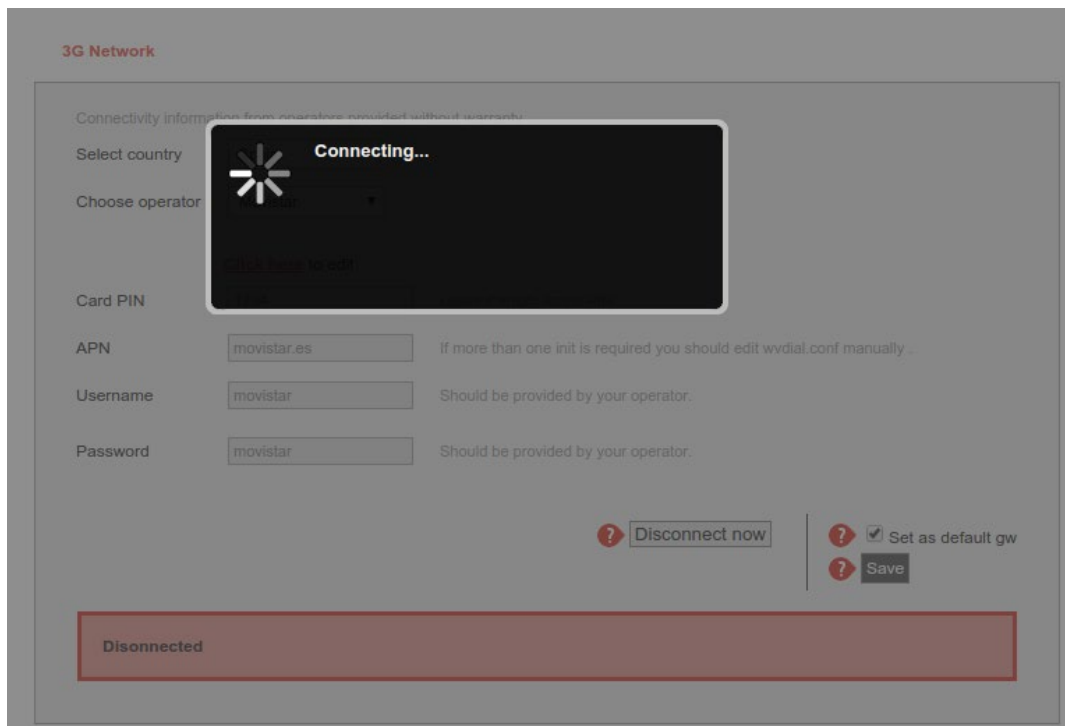
Password Should be provided by your operator.

☒ Set as default gw

Disconnected

After setting the 3G/GPRS parameters and before save them you can **test your connection** through the “**Connect now**” button. It will try to connect to your carrier and get a valid IP. Once the connection has been made the default gateway of the machine is changed so all the clients connected through Wifi will reach the Internet via 3G/GPRS.

Important: once you get a valid 3G/GPRS IP through the “Connect now” button, you will not be able to access Meshlium via Ethernet unless you are connected through the same Local Area Network. For this reason we recommend to make all the tests using the WiFi connection.



3G Network

Connectivity information from operators provided without warranty.

Select country: [Dropdown menu]

Choose operator: [Dropdown menu]

Card PIN: [Input field]

APN: [Input field: movistar.es] If more than one init is required you should edit wvdial.conf manually.

Username: [Input field: movistar] Should be provided by your operator.

Password: [Input field: movistar] Should be provided by your operator.

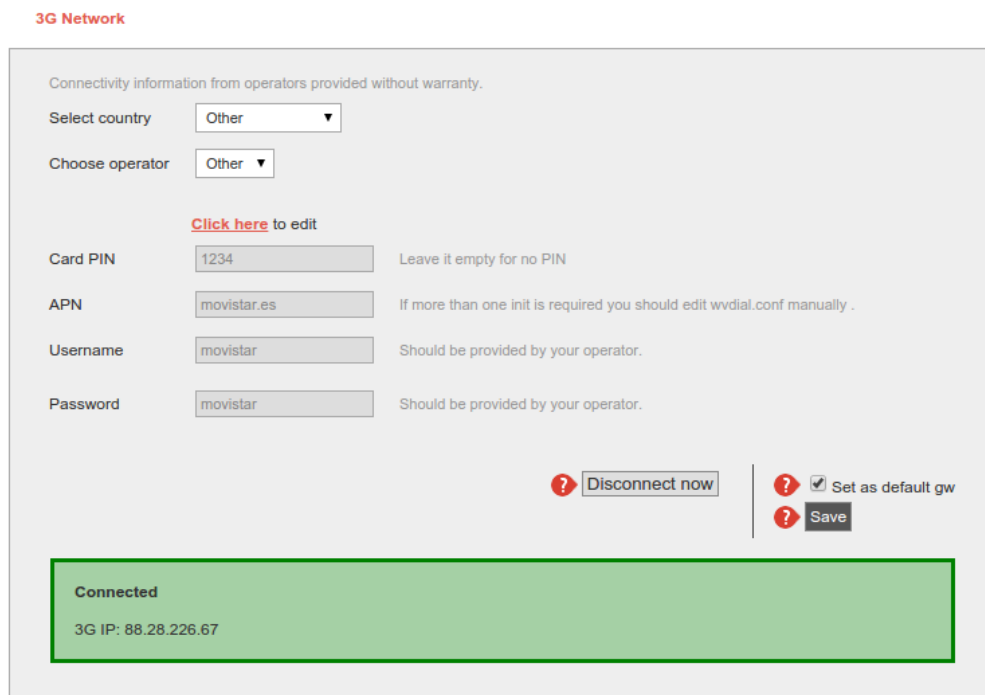
[?] Disconnect now

[?] ☒ Set as default gw

[?] Save

Disconnected

Once validated your settings press the Disconnect button and save your configuration. If you want the 3G/GPRS to be the Default Gateway of Meshlium each time it starts just activate the service in the **“Set as Default Gateway”** check box before saving. Setting this each time Meshlium restarts will connect to the Internet using the 3G/GPRS radio.



3G Network

Connectivity information from operators provided without warranty.

Select country: [Dropdown menu: Other]

Choose operator: [Dropdown menu: Other]

[Click here to edit](#)

Card PIN: [Input field: 1234] Leave it empty for no PIN

APN: [Input field: movistar.es] If more than one init is required you should edit wvdial.conf manually.

Username: [Input field: movistar] Should be provided by your operator.

Password: [Input field: movistar] Should be provided by your operator.

[?] Disconnect now

[?] ☒ Set as default gw

[?] Save

Connected

3G IP: 88.28.226.67

Note for US users: We tested the 3G shield with the AT&T network which supports natively the GSM and 3G protocols. With other carriers may also work although we haven't tried and thus we can not ensure it. For this reason we recommend to use AT&T SIM cards.

14. Joined Networks

This is an information section where you will find how the networks have been joined in order to give access from one to other interface. Possible networks joined are:

Wireless AP -> Ethernet

This bridge allows the clients connected to the Wifi AP access to the Internet through the Meshlium Ethernet connetion.



Wireless AP -> 3G/GPRS

This bridge allows the clients connected to the Wifi AP access to the Internet through the Meshlium 3G/GPRS connetion.



Wireless AP -> Wireless Mesh

This bridge allows the clients connected to the Wifi AP access to other nodes of the Wifi Mesh network. It also let access to the Internet if it is shared for the Gateway node in the Mesh network.



Wireless Mesh -> Ethernet

This bridge allows a Gateway node in a Wifi Mesh Network to share its Internet connection which comes from Ethernet with the rest of the nodes in the Mesh Network and the clients connected to their Wifi AP interface.



Wireless Mesh -> 3G/GPRS

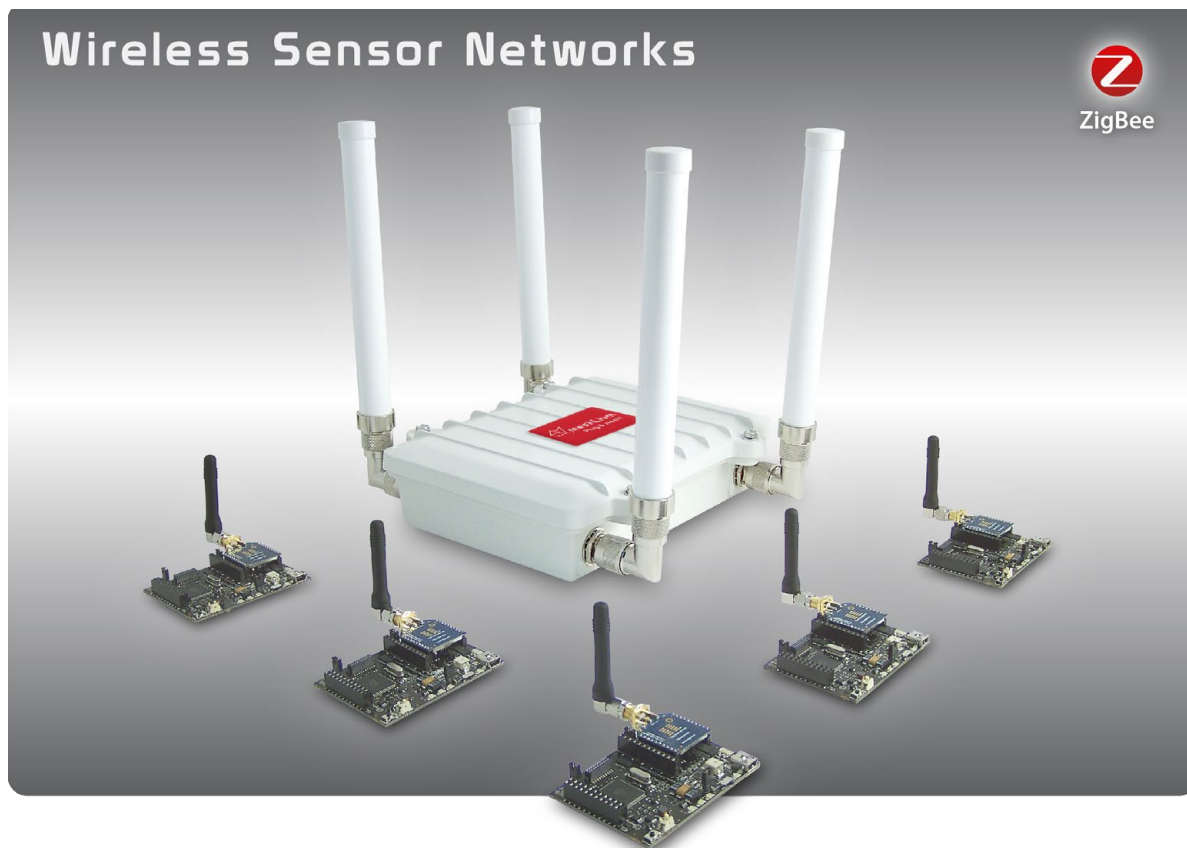
This bridge allows a Gateway node in a Wifi Mesh Network to share its Internet connection established via 3G/GPRS with the rest of the nodes in the Mesh Network and the clients connected to their Wifi AP interface.



The real union of the networks is made automatically when setting the node type in the Presets section.

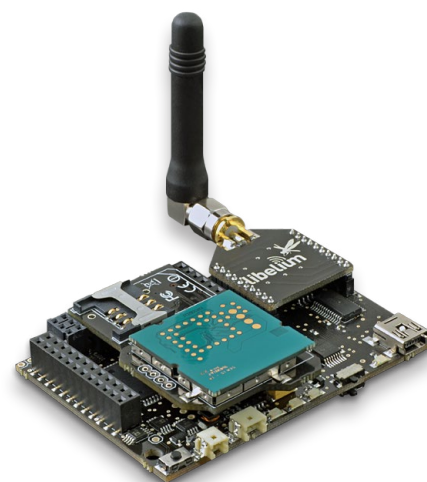
15. Wireless Sensor Networks

15.1. Configuration



One of the applications of Meshlium is to act as a Gateway of the **Wasmote Sensor Networks**. Wasmote is a sensor device specially oriented to developers. It works with different protocols (ZigBee, XBee, LoRa, Bluetooth, WiFi, 3G/GPRS) and frequencies (2.4GHz, 868MHz, 900MHz) being capable of getting links up to 22km. It counts with a hibernate mode of **0.7uA** which allows to save battery when it is not transmitting. More than **50 sensors** already available and a complete open source IDE (API libraries + compiler) made really easy to start working with the platform.

More info at: <http://www.libelium.com/wasmote>





The screenshot shows the Meshlium Manager System web interface. The top navigation bar includes 'Meshlium Manager System', 'Meshlium Zigbee 3G AP', and user controls for 'meshlium' (Home | Logout), 'Restart', 'Shutdown', and 'Presets'. The main menu on the left includes 'Interfaces', 'Sensor Networks', 'Cloud Connector', 'Tools', 'System', 'Update Manager', and 'Help'. The 'Sensor Networks' section is active, showing a sidebar with 'Zigbee', 'Capturer', 'Logs', 'Sensor list', 'Photo / Video', and 'OTA - FTP'. The main content area displays 'Captured Data' with tabs for 'Local DataBase', 'External Database', 'Show me NOW', and 'Advanced'. The 'Local DataBase' tab is selected, showing a 'Connection data' form with fields for Database (MeshliumDB), Table (sensorParser), IP (localhost), Port (3306), User (root), and Password (libelium2007). Below the form is a table of captured data with columns: ID, Date, Sync, ID Wasp, ID Secret, Frame Type, Frame Number, and Ser. The table contains 10 rows of data, all with a Sync value of 0 and a Frame Type of 128.

ID	Date	Sync	ID Wasp	ID Secret	Frame Type	Frame Number	Ser
107522	2014-10-30 13:24:28	0	node_001	387227154	128	30	ACC
107521	2014-10-30 13:24:28	0	node_001	387227154	128	30	BAT
107520	2014-10-30 13:24:20	0	node_001	387227154	128	29	ACC
107519	2014-10-30 13:24:20	0	node_001	387227154	128	29	BAT
107518	2014-10-30 13:24:12	0	node_001	387227154	128	28	ACC
107517	2014-10-30 13:24:12	0	node_001	387227154	128	28	BAT
107516	2014-10-30 13:24:04	0	node_001	387227154	128	27	ACC
107515	2014-10-30 13:24:04	0	node_001	387227154	128	27	BAT
107514	2014-10-30 13:23:55	0	node_001	387227154	128	26	ACC
107513	2014-10-30 13:23:55	0	node_001	387227154	128	26	BAT
107512	2014-10-30 13:23:47	0	node_001	387227154	128	25	ACC
107511	2014-10-30 13:23:47	0	node_001	387227154	128	25	BAT
107510	2014-10-30 13:23:39	0	node_001	387227154	128	24	ACC

15.2. Configuration of the ZigBee interface

There are 5 different XBee models can can be configured:

 802.15.4	➡	XBee 802.15.4
 Zigbee	➡	XBee ZigBee (ZB)
 868 MHz	➡	XBee 868MHz
 900 MHz	➡	XBee 900MHz
 DigiMesh	➡	XBee DigiMesh
 LoRa	➡	LoRa

Depending the kind of XBee model the parameters to be configured may vary.

Complete list:

- **Network ID:** Also known as PAN ID (Personal Area Network ID)
- **Channel:** frequency channel used
- **Network Address:** 16b address (hex field) - MY
- **Node ID:** maximum 20 characters (by default "Meshlium")
- **Power level:** [0..4] (by default 4)
- **Encrypted mode:** true/false (by default false)
- **Encryption Key:** 16 characters maximum
- **MAC:** 64b hardware address. It is a read only value divided in two parts:
 - MAC-high: 32b (hex field)
 - MAC-low: 32b (hex field)

These parameters must be also configured in the Wasp mote sensor nodes. Access to all the information related to Wasp mote at:

<http://www.libelium.com/waspote>

DigiMesh

Network ID:	<input type="text" value="3332"/>
Channel:	<input type="text" value="0x0E"/>
Node ID:	<input type="text" value="Meshlium"/>
Power Level:	<input type="text" value="2"/>
Encrypted mode:	<input type="text" value="Off"/>
Encryption key:	<input type="text"/>
MAC high:	<input type="text" value="13a200"/>
MAC low:	<input type="text" value="407791fc"/>

To discover the MAC address of the XBee module just press the "Load MAC" button.

The "Check status" option allows to see if the RF module is working properly and if the configuration stored on it matches the values set in the Manager System.

Both process ("Load MAC" and "Check status") require the sensorParser daemon to be stopped. This means no frames will be received while executing this actions. Be patient this can take up to 1 minute to finish.

DigiMesh

Network ID:

Channel:

Node ID:

Power Level:

Encrypted mode:

Encryption key:

MAC high:

MAC low:

Connecting to serial port ...
Connected.

Network ID: OK
Node ID: OK
Power Level: OK
Encrypted Mode: OK

Meshlium Manager System

The open source router web manager

Meshlium LoRa SX1272 AP

DeathStar

Home | Logout

LoRa

Capturer

Logs

Sensor list

Photo / Video

OTA - FTP

LoRa setup

Band:

Channel:

LoRa Address: (1-254)

Bandwidth:

Coding Rate:

Spreading Factor:

Connecting to serial port ...
Connected.

Band: OK → 868 MHz
Channel: OK → 12
LoRa Address: OK → 1
Bandwidth: OK → 125 kHz
CR: OK → 4/5
SF: OK → 12
SNR: 4 dB
General RSSI: -106 dBm
Last packet RSSI: -97 dBm

Send data

Destiny Address: (1-254) (Comma separated for multiple addresses)

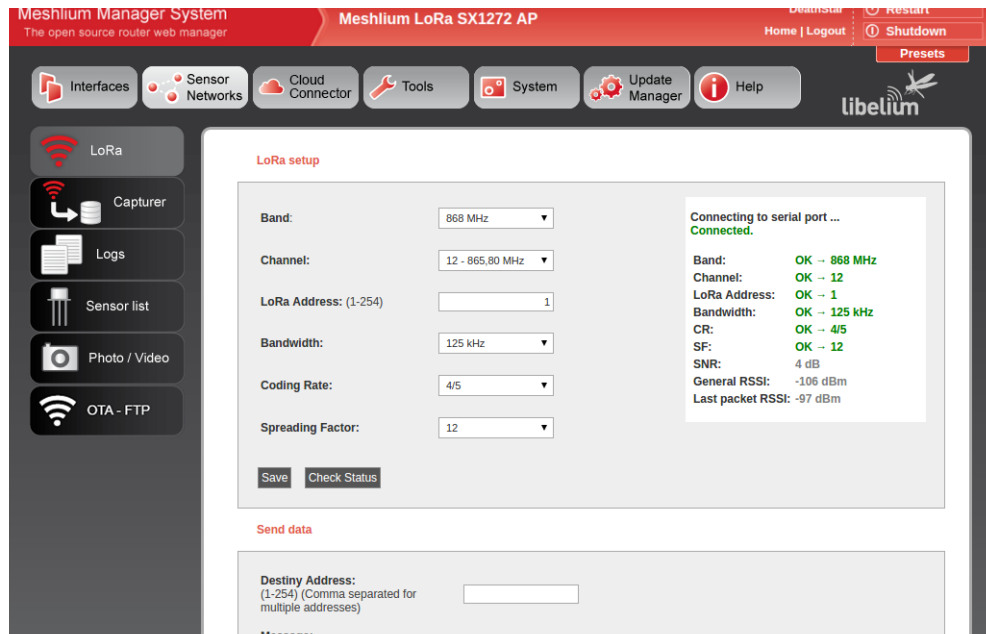
Message: (max. 240 characters, only alphanumeric characters allowed)

Note: When you buy a Waspnote Developer kit with Meshlium and with the XBee ZB as ZigBee radio both the Waspnote GW and Meshlium come configured as Coordinator of the network. Take into account that only one of them can be working at the same time.

Note: If the encryption check fails but the rest of parameters are OK, it means the ZigBee radio has an old version of the firmware but it is working perfectly.

In the model with LoRa Semtech Sx1272 module, the parameters to setup are:

- **Frequency Band:** Options are 868 MHz and 900 MHz bands.
- **Channel:** Frequency channel used.
- **LoRa Address:** Address for the Meshlium LoRa module. Waspnotes will have to send information to this address.
- **Bandwidth:** Choose between 125 kHz, 250 kHz and 500 kHz.
- **Coding rate:** Possible values are 4/5, 4/6, 4/7 and 4/8.
- **Spreading Factor:** Values from 6 to 12.



These parameters must be also configured in the Wasp mote sensor nodes. Access to all the information related to Wasp mote at: <http://www.libelium.com/waspote>

The “Check status” option allows to see if the RF module is working properly and if the configuration stored on it matches the values set in the Manager System.

“Check status” require the sensorParser daemon to be stopped. This means no frames will be received while executing this actions. Be patient this can take up to 1 minute to finish.

15.3. Capturing and storing sensor data

15.3.1. Capturing and storing sensor data from XBee / LoRa

When you buy a kit containing Meshlium and Wasp mote, they already come configured to send frames to the Gateway. Later, once the user has developed the code for transmitting to Gateway, he can switch to Meshlium.

Meshlium will receive the sensor data sent by Wasp mote using the RF radio and it will store the frames in the Local Data Base. That can be done in an automatic way thanks to the **Sensor Parser**.

The **Sensor Parser** is a software system which is able to do the following tasks in an easy and transparent way:

- receive frames from XBee / LoRa (with the Data Frame format)
- receive frames from 3G/GPRS, WiFi and Ethernet via HTTP protocol (Manager System version 3.1.4 and above)
- parse these frames
- store the data in local Database
- synchronize the local Database with an external Database

Besides, the user can add his own sensors.

The initial frames sent by Wasp mote contain the next sequence:

```
~\0x00I\0x90\0x00}3\0xa2\0x00@z\0xcb\0x92\0xd8\0xd3\0x02<=>\0x80\0x03#35689722##7#ACC:80;10;987#IN_TEMP:22.50#BAT:93#\0xb4
```

Initially there are some hexadecimal characters, which belong to the API frame, followed by the message. In the above example the message is:

```
<=>\0x80\0x03#35689722##7#ACC:80;10;987#IN_TEMP:22.50#BAT:93#
```

They are formed by the accelerometer values, RTC internal temperature value, and battery level. The MAC address is added and other helpful information.

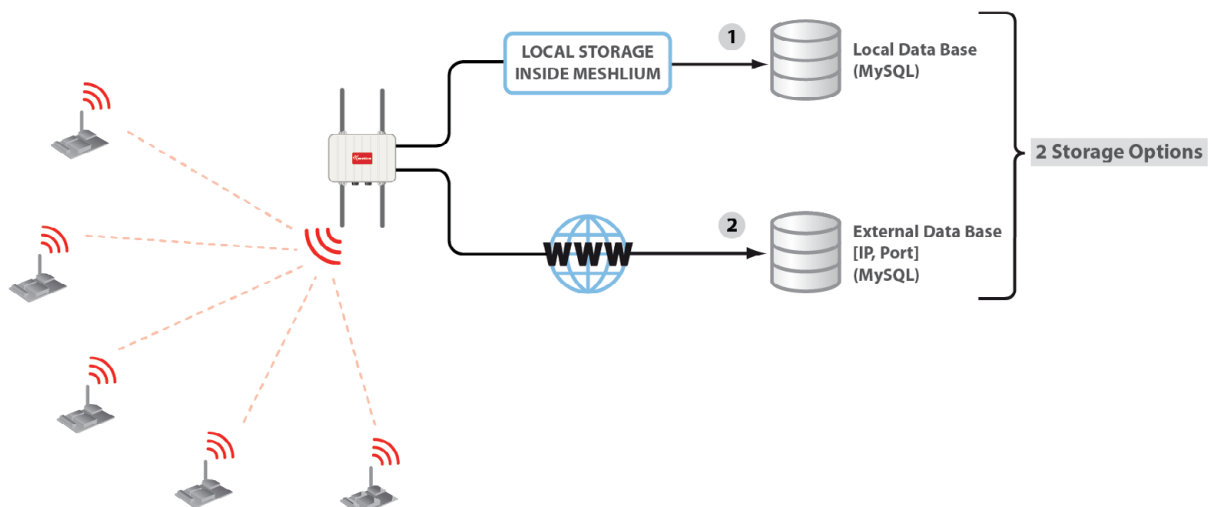
In order to add your own sensor frames properly go to the section "Capturing and storing your own ZigBee frames". All frames captured will be able to stored on Local Database, however the frame has not been defined is stored in the database. See the picture below in order to see different frames types and how they are saved in the database.

ID	Date	Sync	ID Wasp	ID Secret	Frame Type	Frame Number	Se
87493	2013-01-31 08:33:38	0	N1	35690399	253	57	IN
87492	2013-01-31 08:33:38	0	N1	35690399	253	57	BA
87491	2013-01-31 08:33:38	0	N1	35690399	253	57	ST
87489	2013-01-31 08:33:27	0	<=>\0x80\0x03#35690399#N1#56#STR:XBee frame#BAT:90#IN_TE				
87488	2013-01-31 08:33:17	1	N1	35690399	253	55	IN
87487	2013-01-31 08:33:17	1	N1	35690399	253	55	BA
87486	2013-01-31 08:33:17	1	N1	35690399	253	55	ST
87485	2013-01-31 08:33:06	1	N1	35690399	253	54	IN
87484	2013-01-31 08:33:06	1	N1	35690399	253	54	BA
87483	2013-01-31 08:33:06	1	N1	35690399	253	54	ST
87482	2013-01-31 08:32:56	1	N1	35690399	253	53	IN
87481	2013-01-31 08:32:56	1	N1	35690399	253	53	BA
87480	2013-01-31 08:32:56	1	N1	35690399	253	53	ST

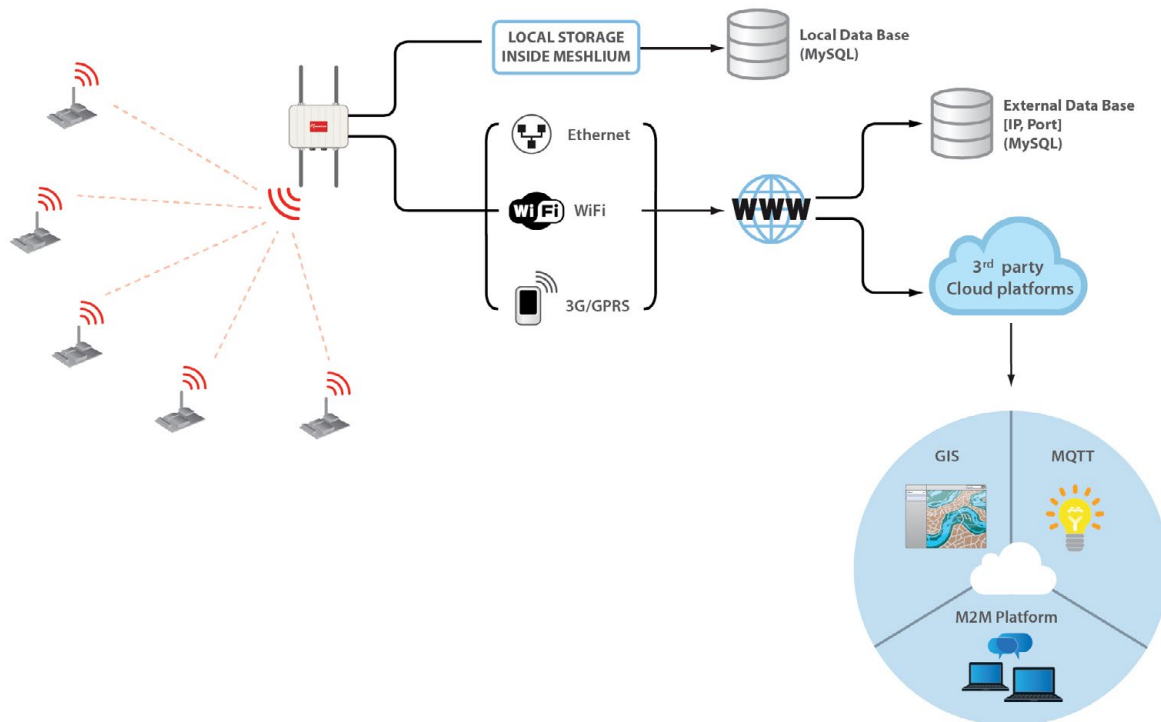
If you change any of the parameters in Waspnote or Meshlium you will have to do it in both platforms so that they still can communicate.

We can perform two different storage options with the frames captured:

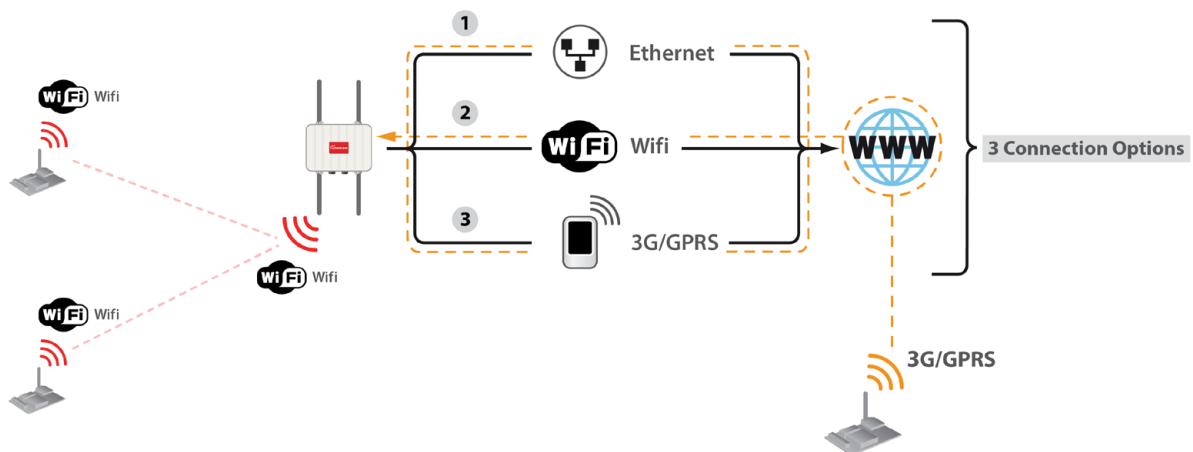
- Local Data Base
- External Data Base



You can also send the information received to the Internet using the Ethernet, Wifi and 3G/GPRS interfaces.



15.3.2. Capturing and storing data from 3G/GPRS, WiFi and Ethernet



From version 3.1.4, Meshlium accepts POST and GET requests in any of its interfaces so Waspmodules are capable of sending frames, through GPRS, 3G or WiFi modules, via HTTP requests. Meshlium, through HTTP requests is capable of:

- receive frames from 3G/GPRS, WiFi or Ethernet via HTTP (version 3.1.4 or older)
- parse these frames
- store the data in local Database
- synchronize the local Database with an external Database

Frames received by this method are stored the same way that ZigBee frames, and are identically processed at synchronization stage. You can view received frames in the same way that radio frames.

Like the case of ZigBee, the user can add his own sensors.

15.3.3. Local Data Base

Meshlium has a MySQL data base up and running which is used to store locally the information captured. In the “Local Data Base” tab you can see the connection parameters.

- **Database:** MeshliumDB
- **Table:** sensorParser
- **IP:** localhost
- **Port:** 3306
- **User:** root
- **Password:** libelium2007

Captured Data

Local DataBase External Database Show me NOW Advanced

Connection data

Database: MeshliumDB

Table: sensorParser

IP: localhost

Port: 3306

User: root

Password: libelium2007

Show data Last 100 insertions.

ID	Date	Sync	ID Wasp	ID Secret	Frame Type	Frame Number	Sensor
791	2014-05-19 08:05:37	128	wasp5	1111	132	22224	BAT
790	2014-05-19 08:05:37	128	wasp5	1111	132	22224	HUMA
789	2014-05-19 08:05:37	128	wasp5	356895649	128	0	BAT
788	2014-05-19 08:05:37	128	wasp5	356895649	128	0	TIME
787	2014-05-16 12:44:45	128	wasp5	1111	132	22224	BAT
786	2014-05-16 12:44:45	128	wasp5	1111	132	22224	HUMA
785	2014-05-16 11:57:27	128	wasp5	356895649	128	0	BAT
784	2014-05-16 11:57:27	128	wasp5	356895649	128	0	TIME
783	2014-05-16 11:41:28	128	wasp5	382544983	128	18	GPS
782	2014-05-16 11:41:28	128	wasp5	382544983	128	18	ACC
781	2014-05-16 11:41:28	128	wasp5	382544983	128	18	IN_TEMP
780	2014-05-16 11:41:28	128	wasp5	382544983	128	18	TIME
779	2014-05-16 11:40:53	128	wasp5	382544983	128	17	GPS

Steps:

At any time you can see the last “x” records stored. Just set how many insertions you want to see and press the “Show data” button.

15.3.4. External Data Base

Meshlium can also store the information captured in an External Data Base.

Steps:

1. Pressing the "Show sql script" you will get the code needed to create the data base along with the table and the right privileges.

Captured Data

Local DataBase External Database Show me NOW Advanced

Connection data

Database: ParserExternal


Table: zigbeeParser

IP: 192.168.1.6

Port: 3306

User: root

Password: root

 ☒ **Store frames in the external data base**

Synchronize each 30 seconds Save

Show data Last 100 insertions. Show sql script (to create database and table)

Save Check Connection Synchronize Now

Just copy paste:

```
CREATE database MeshliumDB;
```

Just copy paste:

```
CREATE TABLE IF NOT EXISTS `sensorParser` (
  `id` int(11) NOT NULL auto_increment,
  `id_wasp` text character set utf8 collate utf8_unicode_ci,
  `id_secret` text character set utf8 collate utf8_unicode_ci,
  `frame_type` int(11) default NULL,
  `frame_number` int(11) default NULL,
  `sensor` text character set utf8 collate utf8_unicode_ci,
  `value` text character set utf8 collate utf8_unicode_ci,
  `timestamp` timestamp NOT NULL default CURRENT_TIMESTAMP,
  `raw` text character set utf8 collate utf8_unicode_ci,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

2. Insert this code in your MySQL management application.

3. Fill the Connection Data fields with the information about where the data base is located (IP, Port) and with the authentication options (Database, Table, User, Password).

This data are stored in `/mnt/lib/cfg/sensorExternalDB` file.

4. Now press the "Check Connection" button to see if the configuration is correct.

Captured Data

Local DataBase External Database Show me **NOW** Advanced

Connection data

Database: ParserExternal


Table: zigbeeParser

IP: 192.168.1.6

Port: 3306

User: root

Password: root

 ☒ **Store frames in the external data base**

Synchronize each seconds **Save**

Show data Last insertions. **Show sql script** (to create database and table)

Save **Check Connection** **Synchronize Now**

Connecting to the database server ...

Selecting database ...

OK

5. Set the check box "Store frames in external database", you can defined the interval how often to synchronize the local database with external database and press the "Save" button.

From this time Meshlium will automatically perform Scans and will store the results in the External Data Base each . This process will also continue after restarting Meshlium.

You can also choose to sync when you want. Just press the “Synchronize Now” button.

Captured Data

Local DataBase External Database

Connection data

Database:

Table:

IP:

Port:

User:

Password:

Show me NOW

Synchronizing...

☒ Store frames in the external data base

Synchronize each seconds

Last insertions (to create database and table)

ID	Date	ID Wasp	ID Secret	Frame Type	Frame Number	Ser
73848	2013-01-30 19:03:06	N1	35690399	253	62	IN_
73847	2013-01-30 19:03:06	N1	35690399	253	62	BAT
73846	2013-01-30 19:03:06	N1	35690399	253	62	STR
73845	2013-01-30 19:02:56	N1	35690399	253	61	IN_
73844	2013-01-30 19:02:56	N1	35690399	253	61	BAT
73843	2013-01-30 19:02:56	N1	35690399	253	61	STR
73842	2013-01-30 19:02:45	N1	35690399	253	60	IN_
73841	2013-01-30 19:02:45	N1	35690399	253	60	BAT
73840	2013-01-30 19:02:45	N1	35690399	253	60	STR
73839	2013-01-30 19:02:35	N1	35690399	253	59	IN_
73838	2013-01-30 19:02:35	N1	35690399	253	59	BAT
73837	2013-01-30 19:02:35	N1	35690399	253	59	STR
73836	2013-01-30 19:02:24	N1	35690399	253	58	IN_

At any time you can see the last “x” records stored. Just set how many insertions you want to see and press the “Show data” button.

Captured Data

Local DataBase


External Database

Show me NOW

Advanced

Connection data

Database:
Table:
IP:
Port:
User:
Password:


☒ Store frames in the external data base

Synchronize each seconds

Save

Show data

Last insertions.

Show sql script

(to create database and table)

Save

Check Connection

Synchronize Now

ID	Date	ID Wasp	ID Secret	Frame Type	Frame Number	Ser
73593	2013-01-30 18:48:08	N1	35690399	253	233	IN_
73592	2013-01-30 18:48:08	N1	35690399	253	233	BAT
73591	2013-01-30 18:48:08	N1	35690399	253	233	STR
73590	2013-01-30 18:47:57	N1	35690399	253	232	IN_
73589	2013-01-30 18:47:57	N1	35690399	253	232	BAT
73588	2013-01-30 18:47:57	N1	35690399	253	232	STR
73587	2013-01-30 18:47:47	N1	35690399	253	231	IN_
73586	2013-01-30 18:47:47	N1	35690399	253	231	BAT
73585	2013-01-30 18:47:47	N1	35690399	253	231	STR
73584	2013-01-30 18:47:36	N1	35690399	253	230	IN_
73583	2013-01-30 18:47:36	N1	35690399	253	230	BAT
73582	2013-01-30 18:47:36	N1	35690399	253	230	STR
73581	2013-01-30 18:47:26	N1	35690399	253	229	IN_

15.3.5. Show me now!

In the “Show me now!” tab you can see in real time the Scans captured.

You can specify if you want the information to be updated periodically with the defined interval just checking the “Use the Defined Interval” button.

Captured Data

Local DataBase	External Database	Show me NOW	Advanced Database
----------------	-------------------	--------------------	-------------------

Start Scan

☒ Use the defined Scan interval

10

Seconds

Clean

ASCII frame
Internal ID:35690399 **Wasmote ID:**N1 **Frame Type:**253 **Frame Number:**64
STR:XBee frame
BAT:93
IN_TEMP:29.75

ASCII frame
Internal ID:35690399 **Wasmote ID:**N1 **Frame Type:**253 **Frame Number:**62
STR:XBee frame
BAT:93
IN_TEMP:29.75

ASCII frame
Internal ID:35690399 **Wasmote ID:**N1 **Frame Type:**253 **Frame Number:**60
STR:XBee frame
BAT:93
IN_TEMP:29.75

ASCII frame
Internal ID:35690399 **Wasmote ID:**N1 **Frame Type:**253 **Frame Number:**58
STR:XBee frame
BAT:93
IN_TEMP:29.75

15.3.6. Advanced Database

In the “Advanced” tab you can see information about the state in which they are databases.

It displays information about the Local and External database, showing the following information:

- Local and External Database names
- Local and External Database sizes
- Local and External Tables
- Total Local and External Entries
- Synchronized Local Frames
- Unsynchronized Local Frames

Captured Data

Local DataBase External Database Show me NOW Advanced

Local Database

Database:	MeshliumDB
Database Size:	12.35 Mb
Table:	sensorParser
Entries:	900
Synchronized Frames:	0
Unsynchronized Frames:	900

Remove synchronized Data Remove ALL Content

External Database

Database:	ParserExternal
Database Size:	5.05 Mb
Table:	zigbeeParser
Entries:	72852

Logs Sync

2013-01-30 17:48:50.257 - Synchronization OK
 2013-01-30 17:49:20.157 - Synchronization OK
 2013-01-30 17:49:50.218 - Synchronization OK
 2013-01-30 17:50:20.077 - Synchronization OK
 2013-01-30 17:50:50.327 - Synchronization OK
 2013-01-30 17:51:20.088 - Synchronization OK
 2013-01-30 17:51:50.187 - Synchronization OK
 2013-01-30 17:52:24.039 - Synchronization OK
 2013-01-30 17:52:53.808 - Synchronization OK

From this tab, **you can delete all the information contained in the Local database or Remove synchronized data**. Before performing these actions, a confirmation message will be displayed.

Note: Before running these options, it is recommended to have a backup or having synchronized your local database with external database.

Captured Data


Local DataBase | External Database | Show me **NOW** | Advanced

Local Database

Database:	MeshliumDB
Database Size:	13.50 Mb
Table:	sensorParser
Entries:	15301
Synchronized Frames:	15295
Unsynchronized Frames:	6

External Database

Mensaje de la página 192.168.1.103:

 Synchronized data of sensorParser table will be deleted. Do you want to continue?

2013-01-31 08:33:49.315 - Synchronization OK
 2013-01-31 08:34:19.401 - Synchronization OK
 2013-01-31 08:34:49.138 - Synchronization OK

In addition can display a log of the date of the last synchronization between the local database and external database was successful.

Logs Sync

2013-01-30 17:48:50.257 - Synchronization OK
 2013-01-30 17:49:20.157 - Synchronization OK
 2013-01-30 17:49:50.218 - Synchronization OK
 2013-01-30 17:50:20.077 - Synchronization OK
 2013-01-30 17:50:50.327 - Synchronization OK

15.4. Capturer logs

Inside “Sensor Networks” exist the section **Logs**, in this section you can see the last frames received on Meshlium.

Sensor Log

```
ASCII-35690399-N1-253-43-,STR:XBee frame,BAT:93,IN TEMP:25.50
ASCII-35690399-N1-253-44-,STR:XBee frame,BAT:93,IN TEMP:25.50
ASCII-35690399-N1-253-45-,STR:XBee frame,BAT:93,IN TEMP:25.50
ASCII-35690399-N1-253-46-,STR:XBee frame,BAT:93,IN TEMP:25.50
ASCII-35690399-N1-253-47-,STR:XBee frame,BAT:93,IN TEMP:25.50
ASCII-35690399-N1-253-48-,STR:XBee frame,BAT:93,IN TEMP:25.50
ASCII-35690399-N1-253-49-,STR:XBee frame,BAT:93,IN TEMP:25.50
ASCII-35690399-N1-253-50-,STR:XBee frame,BAT:93,IN TEMP:25.50
ASCII-35690399-N1-253-51-,STR:XBee frame,BAT:93,IN TEMP:25.50
ASCII-35690399-N1-253-52-,STR:XBee frame,BAT:93,IN TEMP:25.50
ASCII-35690399-N1-253-53-,STR:XBee frame,BAT:93,IN TEMP:25.50
ASCII-35690399-N1-253-54-,STR:XBee frame,BAT:93,IN TEMP:25.50
ASCII-35690399-N1-253-55-,STR:XBee frame,BAT:93,IN TEMP:25.75
ASCII-35690399-N1-253-56-,STR:XBee frame,BAT:93,IN TEMP:25.75
ASCII-35690399-N1-253-57-,STR:XBee frame,BAT:93,IN TEMP:25.75
ASCII-35690399-N1-253-58-,STR:XBee frame,BAT:93,IN TEMP:25.75
ASCII-35690399-N1-253-59-,STR:XBee frame,BAT:93,IN TEMP:25.75
```

Frame Log

```
<=>?#35690399#N1#17#STR:XBee frame#BAT:93#IN TEMP:23.50#
<=>?#35690399#N1#18#STR:XBee frame#BAT:93#IN TEMP:23.50#
<=>?#35690399#N1#19#STR:XBee frame#BAT:93#IN TEMP:23.50#
<=>?#35690399#N1#20#STR:XBee frame#BAT:93#IN TEMP:24.25#
<=>?#35690399#N1#21#STR:XBee frame#BAT:93#IN TEMP:24.25#
<=>?#35690399#N1#22#STR:XBee frame#BAT:93#IN TEMP:24.25#
<=>?#35690399#N1#23#STR:XBee frame#BAT:93#IN TEMP:24.25#
<=>?#35690399#N1#24#STR:XBee frame#BAT:93#IN TEMP:24.25#
<=>?#35690399#N1#25#STR:XBee frame#BAT:93#IN TEMP:24.25#
<=>?#35690399#N1#26#STR:XBee frame#BAT:93#IN TEMP:24.25#
<=>?#35690399#N1#27#STR:XBee frame#BAT:93#IN TEMP:24.25#
<=>?#35690399#N1#28#STR:XBee frame#BAT:93#IN TEMP:24.25#
<=>?#35690399#N1#29#STR:XBee frame#BAT:93#IN TEMP:24.25#
<=>?#35690399#N1#30#STR:XBee frame#BAT:93#IN TEMP:24.25#
<=>?#35690399#N1#31#STR:XBee frame#BAT:93#IN TEMP:24.25#
<=>?#35690399#N1#32#STR:XBee frame#BAT:93#IN TEMP:25.00#
<=>?#35690399#N1#33#STR:XBee frame#BAT:93#IN TEMP:25.00#
```

First show the “sensor log”, in this logs shows the frames are stored after being processed.

ASCII-35690399-N1-253-198-,STR:XBee frame,BAT:93,IN_TEMP:31.50

Secondly shown “Frame Log”, in this logs shows the frames stored as the arrive to Meshlium.

<=>?#35690399#N1#198#STR:XBee frame#BAT:93#IN_TEMP:31.50#

15.5. Sensors

In section “Sensor List”, the user can **add new sensors or delete sensors**.

By default, Meshlium recognize all Libelium official sensors frames. All sensor frames that Meshlium can capture and store must be specified in an XML file.

The file with official sensors of Libelium is located in [/mnt/lib/cfg/parser/sensors.xml](#)

The button “update sensors” update the Libelium official sensor. User sensors remaining unchanged.

Users can add and remove sensors in an easy and simple from the ManagerSystem.

To add a new sensor the user must complete the fields:

- ASCII ID: sensor id for ASCII frame.
- Fields: This field specifies the number of sensor fields sent in the frame. This helps to calculate the frame length.
- Type: type of fields
 - uint8_t
 - int
 - float
 - string
 - ulong
 - array(ulong)

Once all fields are filled in, click on the button “Add sensor”

Aviable Sensors

Update sensors

ASCII ID:

Fields:

Type: uint_8 ▼

Add sensor

Standard sensors

ID	ASCII ID	Fields	Type
0	CO	1	float
1	CO2	1	float
2	O2	1	float
3	CH4	1	float
4	LPG	1	float
5	NH3	1	float
6	AP1	1	float
7	AP2	1	float
8	SV	1	float
9	NO2	1	float
10	O3	1	float
11	VOC	1	float
12	TCA	1	float
13	TFA	1	float
14	HUMA	1	float
15	PA	1	float
16	PW	1	float
17	BEND	1	float
18	VBR	1	uint_8
19	HALL	1	uint_8
20	LP	1	uint_8
21	LL	1	uint_8
22	LUM	1	float
23	PIR	1	uint_8
24	ST	1	float
25	MCP	1	uint_8
26	CDG	1	uint_8

User sensors


	ID	ASCII ID	Fields	Type
	200	AGM	9	uint_8

The new user sensors will be added to the new XML file, the file with user sensors is located in `/mnt/lib/cfg/parser/user_sensors.xml`

Note: In "Waspnote data frame guide" document is located more extensive information about how to build the frame.

To delete sensor the user must press the garbage can that appears to the left of the description of the sensor. To complete the action should accept a confirmation message.

User sensors

	ID	ASCII ID	Fields	Type
	200	AGM	9	uint_8

15.6. OTA via FTP

From version 3.0.7, Meshlium can also be used like FTP server to prepare the binary files to be downloaded by Waspnote.

For more info about Over the Air Programming go to:

<http://www.libelium.com/development/waspnote/documentation/over-the-air-programming-guide-otap/>

This feature allows reprogramming Waspnote using an FTP server and FTP client which is Waspnote itself.

There are two basic steps involved in OTA procedure:

- **Step 1:** Waspnote requests a special text file which gives information about the program to update: program name, version, size, etc.
- **Step 2:** If the information given is correct, Waspnote queries the FTP server for a new program binary file and it updates its flash memory in order to run the new program.

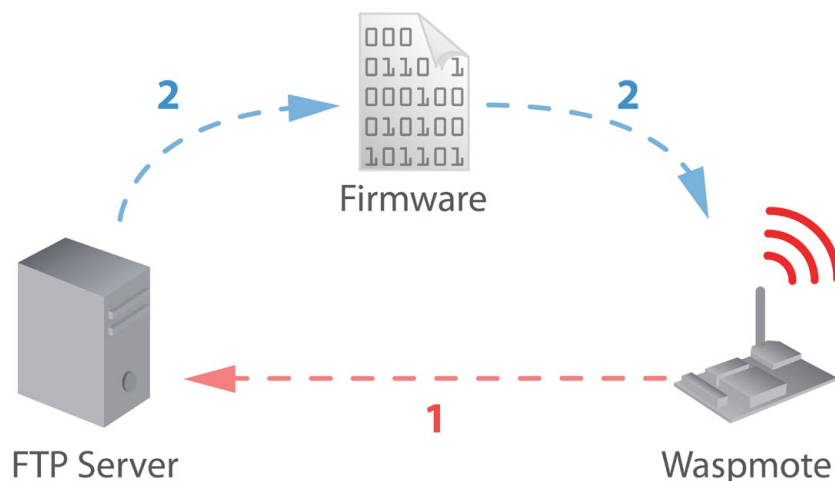


Figure : OTA via FTP protocol

Besides, a default user is configured in Meshlium FTP Server with the following settings:

user: ota

password: libelium

This user directly connects to the following path in Meshlium's system directory where the application creates all the binary and UPGRADE.TXT files:

/mnt/user/ota

Inside "Sensor Network" there is the section **OTA - FTP**. Users can prepare the binary files to be downloaded by Wasp mote. So, you can generate **UPGRADE.TXT** text file necessary to do OTA with 3G/GPRS/WiFi via FTP.

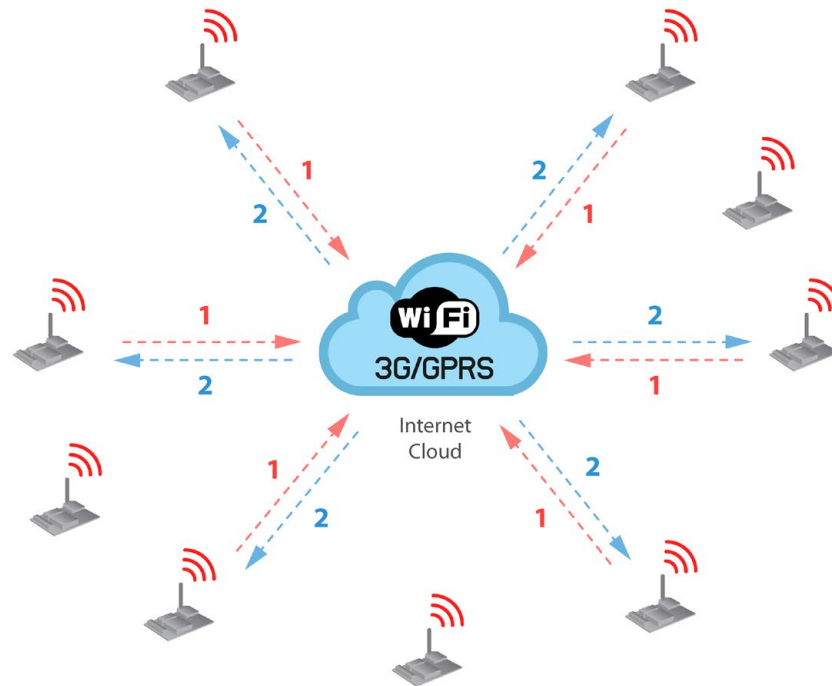


Figure : OTA-FTP Meshlium plug-in

Firstly, there are three possibilities to be chosen:

- Select NO_FILE to inform Wasp mote that no OTA is necessary
- Select a new file generated by Wasp mote-IDE so as to update the Wasp mote's program.
- Select a existing binary if the user needs to update to an older program. The files are stored in the following path: /mnt/user/ota

Secondly, the program version is always set by the user before generating the new UPGRADE.TXT file. There is a specific input to indicate the program version. It must be defined as a 1-unsigned-byte number (range: from 0 to 255).

Finally, there is a button to generate the new UPGRADE.TXT file.

Once these steps have been completed, the binary file and the proper UPGRADE.TXT file are ready for the Wasp mote devices deployed which try to perform OTA via FTP transmission. This file is shown in the last window of the application representing the actual binary prepared for OTA.

15.7. Sending frames from Meshlium to Wasmote

Meshlium can also send frames to the Wasmote nodes. In order to use this feature you have to stop the “capturing and storing” daemon which is running in the system.

To do so access by SSH to Meshlium and stop the default ZigBee daemon::

```
$ sensorParserD.sh stop
```

Now you can execute the ZigBeeSend command. There are several ways to send information to a node:

- Using its 802.15.4 MAC address (64b)
- Using its Network address (MY) (16b)
- Performing a broadcast transmission

Sending to Wasmote using its MAC address (64b):

```
$ ./ZigBeeSend -mac 0013a2004069165d "Hello Wasmote!"
```

Sending to Wasmote using its Net address (MY - 16b):

```
$ ./ZigBeeSend -net 1234 "hello Wasmote!"
```

Send to all the Wasmote devices at the same time - Broadcast mode:

```
$ ./ZigBeeSend -b "hello everybody!"
```

The source code “ZigbeeSend.c” in the path /var/ManagerSystem/plugins/b_SensorData/b0_capturer/bin/

You can download these files and change them in order to get new features and sending options.

Compilation:

The compilation can be done in the same Meshlium. Just copy these files in a folder accessing by SSH and execute:

```
$ gcc -o ZigBeeSend ZigBeeSend.c -lpthread
```

Important: If you want to create a “ZigBee sending” daemon that is executed each time Meshlium starts you have to deactivate the “ZigBee Capturer” daemon (/etc/init.d/ZigbeeScanD.sh) as the ZigBee radio has to be used by one process at a time.

You will find support in the Libelium Forum at: <http://www.libelium.com/forum>

Note: Please contact your sales agent for custom developments concerning Meshlium sending to Wasmote nodes.

15.8. Key Management on Meshlium

15.8.1. Link layer key Management (AES-128)

This feature is provided by XBee and ZigBee modules, and it's not available in LoRa module.

Encryption is this layer provided through the AES 128b algorithm. Specifically through the type AES-CTR. In this case the Frame Counter field has a unique ID and encrypts all the information contained in the Payload field which is the place in the link layer frame where the data to be sent is stored. The way in which the libraries have been developed for module programming means that encryption activation is as simple as running the initialization function and giving it a key to use in the encryption.

```
{  
  xbee.encryptionMode(1);  
  xbee.setLinkKey(key);  
}
```

In Manager System, on Sensor Network section, users can encrypt messages on link layer.

Depending the kind of module model the parameters to be configured may vary.

Complete list:

Network ID: Also known as PAN ID (Personal Area Network ID)

Channel: frequency channel used

Network Address: 16b address (hex field) - MY

Node ID: maximum 20 characters (by default "Meshlium")

Power level: [0..4] (by default 4)

Encrypted mode: true/false (by default false)

Encryption Key: Must be 16 characters

MAC: 64b hardware address. It is a read only value divided in two parts:

MAC-high: 32b (hex field)

MAC-low: 32b (hex field)

The parameter to providing AES-128 to link layer are: encrypted mode and encrypted key.

DigiMesh

Network ID:

Channel:

Node ID:

Power Level:

Encrypted mode:

Encryption key:

MAC high:

MAC low:

Connecting to serial port ...
Connected.

Network ID: **OK**
Node ID: **OK**
Power Level: **OK**
Encrypted Mode: **OK**

15.8.2. Application Layer Key Management (AES-192/256)

Meshlium is capable to properly receive encrypted data from Waspnote. The coding process is made in the application layer, so it's Waspnote and Meshlium processor and not XBee/ZigBee module who encrypts and decrypts the messages.

The user have to set a key for the encryption in Waspnote and Meshlium.

In Manager System, inside section Tools, go to encryption section:

Meshlium Manager System
The open source router web manager

Meshlium LoRa SX1272 AP

DeathStar

Home | Logout

Interfaces Sensor Networks Cloud Connector Tools System Update Manager Help

libelium

AES Management

Encryption Waspnote List

Waspnote wasp1

Node ID:

AES Secret Key: (16,24 or 32 characters)

Encryption

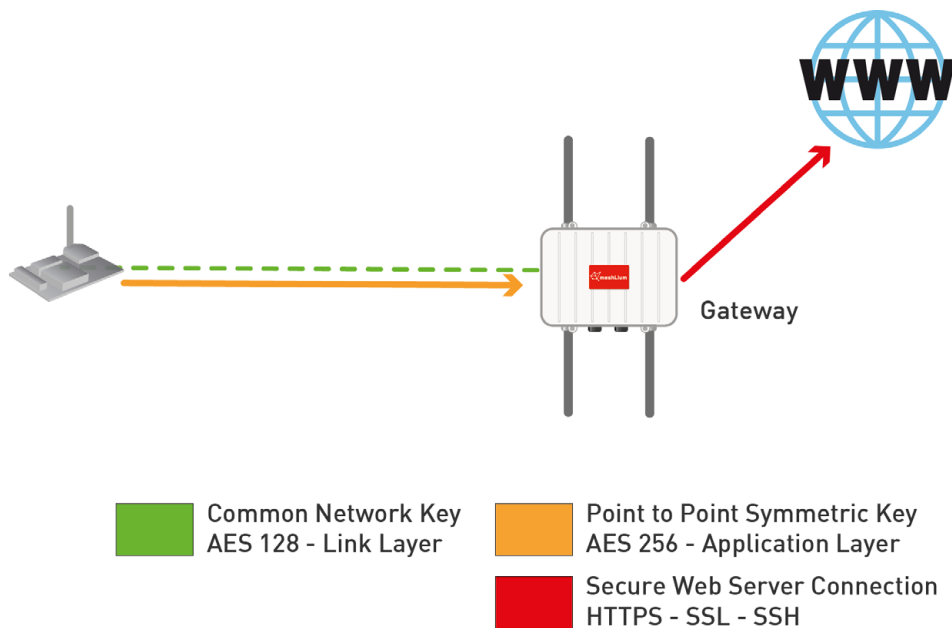
© Libelium Comunicaciones Distribuidas S.L. | [Terms of use](#)

For each Wasmote can send frames to Meshlium, Wasmotes keys can be added to a *encryption Key* file. In this interface the user must specify the node ID and the Wasmote AES secret key (128, 192 or 256 bits).

After defining the above fields to press the button "Add Wasmote". A new entry is generated in the left list.

To delete Wasmote of list, select the Wasmote and press "Delete Wasmote". **The encrypted Wasmote frames received can not be decrypted anymore.**

The AES secret key is necessary to recognize the frames sent each Wasmote to Meshlium.



When a frame arrives at Meshlium sensorParser will consult the *encryptionKey* file for the AES secret key, and use the AES algorithm to decrypt the message.

15.9. AES on Meshlium

Once the user has properly set the AES keys associated to every waspmote, receiving AES encrypted frames in Meshlium is a straightforward process.

As an encrypted frame arrives to Meshlium, sensorParser program takes the appropriate key for the Wasmote ID. The frame is decoded with the key and the information is extracted to the sensor database.

Bear in mind that to use this feature, the frame have to be created with the Wasmote libraries for AES frames. You can see futher information about this in the Wasmote guides.

<http://www.libelium.com/wasmote>

16. Meshlium Visualizer

Meshlium Visualizer is a plugin which plots graphs and maps with the data stored in the database. It can also export data in common formats. Meshlium Visualizer is a special software feature **only available** in the Meshlium units included in the IoT Vertical Kits (Smart Cities IoT Vertical Kit, Smart Water IoT Vertical Kit, etc).

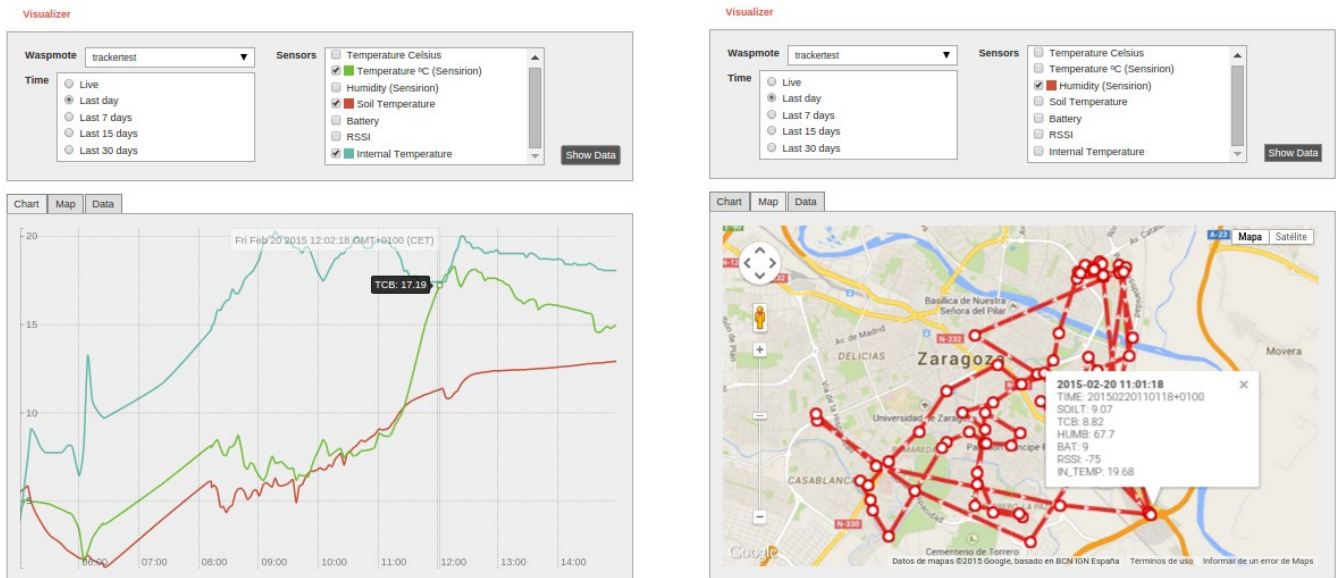


Figure : Meshlium Visualizer can plot graphs and geo-locate data on maps

Please note that this is a paid service. In every IoT Vertical Kit, each Meshlium comes with **100** visualizations. After 100 visualizations, users can contact Libelium Sales Department (sales@libelium.com) if they want to continue using the service.

Note: SSH access to Meshlium devices provided as part of IoT Vertical Kits is not activated.

16.1. Working with the Visualizer

On the top of the page you can use a simple form to make all your queries. To do so, just follow these steps:

Figure : Filling Meshlium Visualizer's form

1. Select one Plug & Sense! from the list. All Plug & Sense! units with frames in the database will be shown.
2. Once a Plug & Sense! Is selected, all its sensors will be loaded. This process is repeated each time you change the selected Plug & Sense!.
3. Select the period of time you want to see in the chart. The "Live" option reads directly from the database, while the rest options read from a file generated everyday by the service cron. For each Plug & Sense!, cron generates 4 files each day, one for the last day, other for last 7 days, other for last 15 days and other for the last 30 days.
4. Hit on the "Show Data" button and, if your query has results to show, Meshlium Visualizer will show them. The remaining visualization number will decrease in one unit. If the query does not have any results, a message will appear notifying the situation; the available visualizations remain without changes.

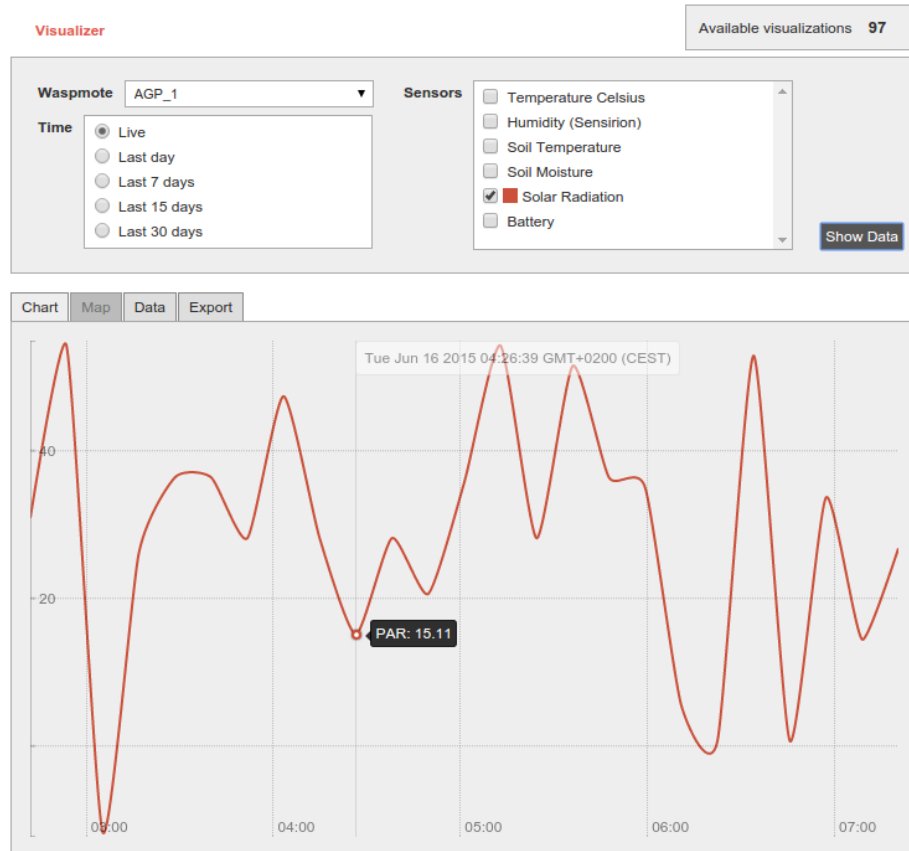


Figure : Meshlium Visualizer showing one graph

If your query has GPS results (data frames with GPS information), the “Map” tab will be shown. If it is not the case, like in the previous picture, this tab remains disabled.

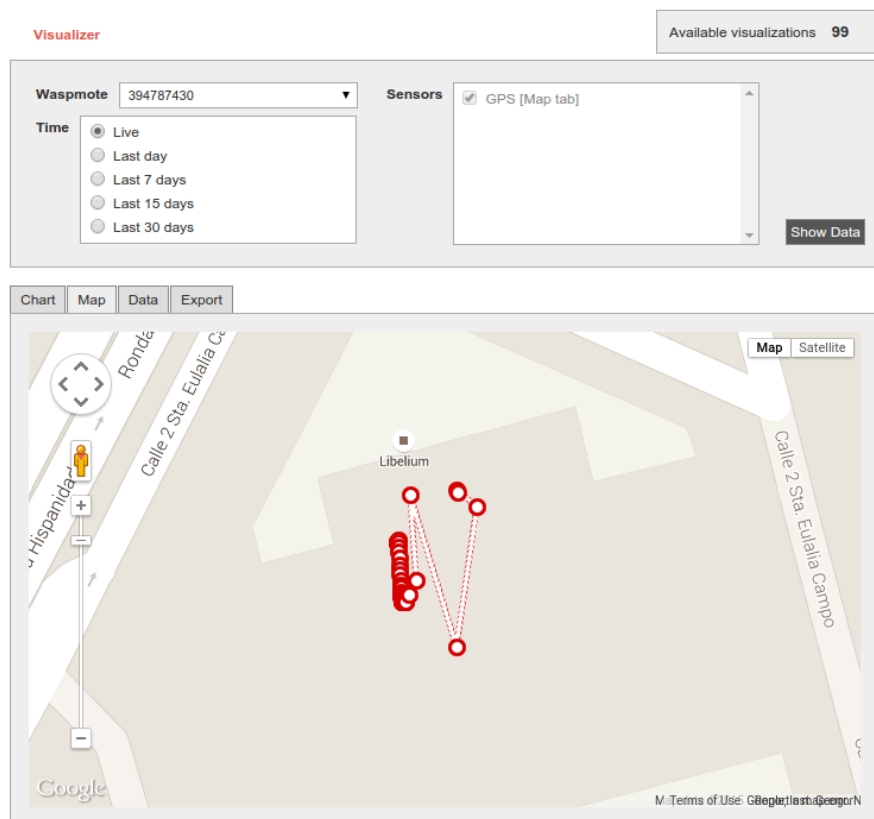


Figure : Locating nodes on a map thanks to Meshlium Visualizer

The “Data” tab shows a list of sensor values, ordered by time.

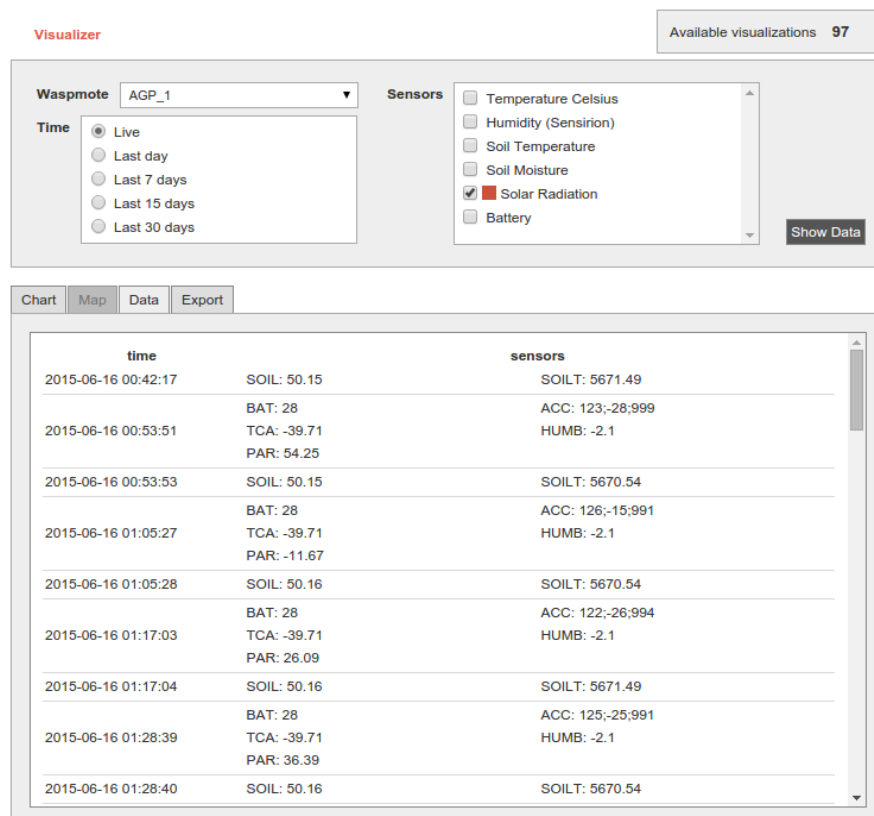


Figure : Meshlium Visualizer showing the Data tab

The "Export" tab shows two calendars to select the initial and final date. This feature does not take into account the block on the top of the page, it will export all data from all Plug & Sense! units between these dates. Data can be exported in 5 formats (CSV, SQL, XML, TXT & HTML) and compressed in ZIP.

Visualizer

Available visualizations 97

Waspnote AGP_1

Time

- Live
- Last day
- Last 7 days
- Last 15 days
- Last 30 days

Sensors

- ☐ Temperature Celsius
- ☐ Humidity (Sensirion)
- ☐ Soil Temperature
- ☐ Soil Moisture
- ☒ Solar Radiation
- ☐ Battery

Show Data

Chart

Map

Data

Export

Start date

June - 2015 -

Sun	Mon	Tue	Wed	Thu	Fri	Sat	06:00
31	1	2	3	4	5	6	07:00
7	8	9	10	11	12	13	08:00
14	15	16	17	18	19	20	09:00
21	22	23	24	25	26	27	10:00
28	29	30	1	2	3	4	11:00

End date

June - 2015 -

Sun	Mon	Tue	Wed	Thu	Fri	Sat	06:00
31	1	2	3	4	5	6	07:00
7	8	9	10	11	12	13	08:00
14	15	16	17	18	19	20	09:00
21	22	23	24	25	26	27	10:00
28	29	30	1	2	3	4	11:00

Output file

- ☒ CSV
- ☐ SQL
- ☐ XML
- ☐ TXT
- ☐ HTML

- ☒ Compress in ZIP file

Export

Figure : Configuring Meshlium Visualizer to export data

17. Cloud Connector

The aim of this chapter is to introduce the user to the Meshlium Cloud Connector functionality. This section will help you to connect your Meshlium to a third party cloud platform.

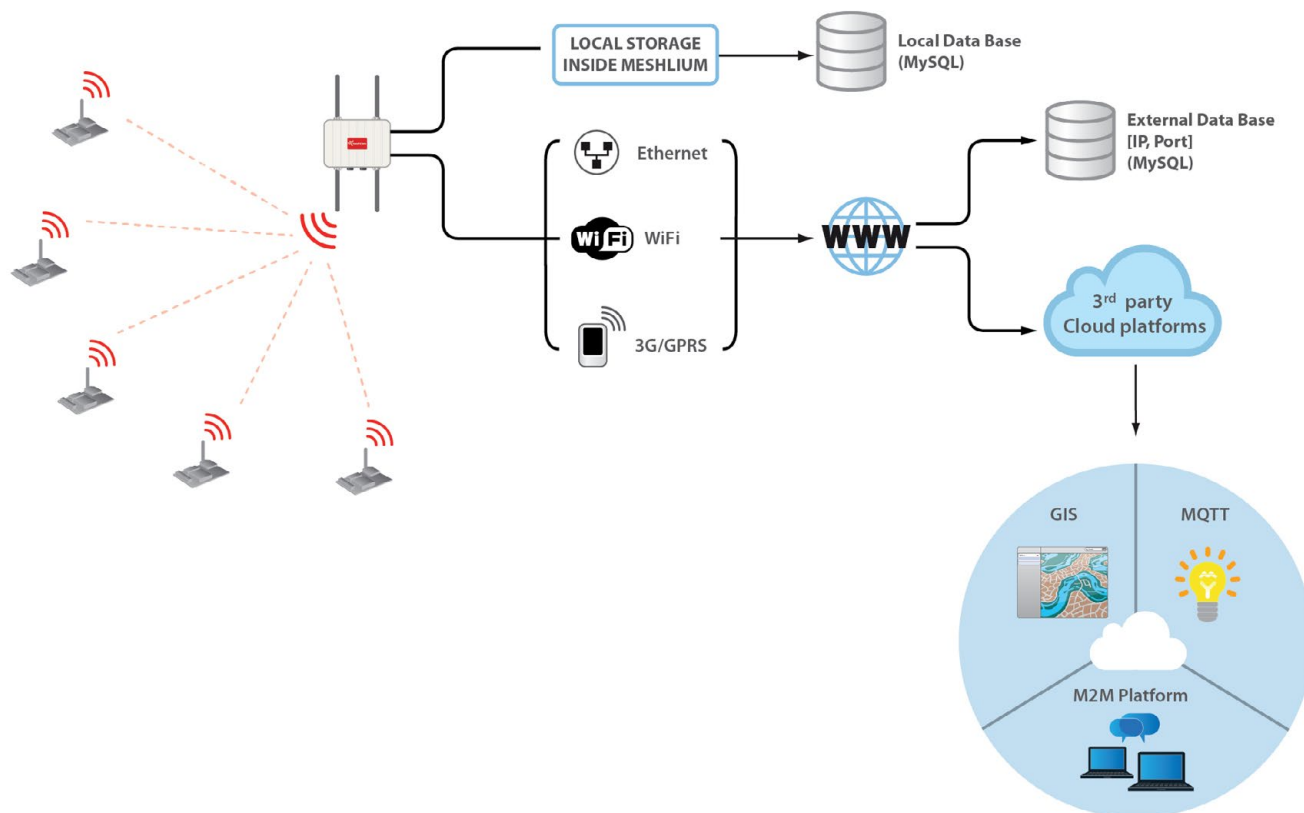


Figure : Cloud connector diagram

Interfacing Meshlium with 3rd party cloud services should be the last step the user develops in any project. The user should analyze if the use of clouds is needed, and in positive case, that will be the last step in the project. Before trying clouds, make sure all the Wasp mote units are sending frames to Meshlium, and Meshlium is receiving and inserting them on the local database properly.

This chapter is only intended for Meshlium Xtreme versions higher or equal to v3.1.0 If you do not have that version, see chapter “Upgrading old versions of Meshlium” on this guide.

What is a cloud platform?

Cloud computing is a major change in our industry. One of the most important parts of that paradigm are cloud platforms. This kind of platforms lets developers write applications that run in the cloud, or use services provided from the cloud, or both.

Meshlium Cloud Connector

Meshlium runs a set of scripts for implementing the data synchronization from its internal database “to the clouds”. In other words, those scripts send data to webservers where the cloud service providers host their clouds. Those scripts are called Cloud Connector.

We have divided the Cloud Connector into 2 groups: “IoT Solutions” and “IoT Platforms”.

IoT Platforms are professional development frameworks for developing data management applications, including Esri, ThingWorx, IBM Bluemix, Telefónica and Microsoft Azure.

IoT Solutions are specific applications focused in different verticals. Libelium promotes the Cloud Partnership Program for any cloud service provider that would like to foster their very own solution using our products.

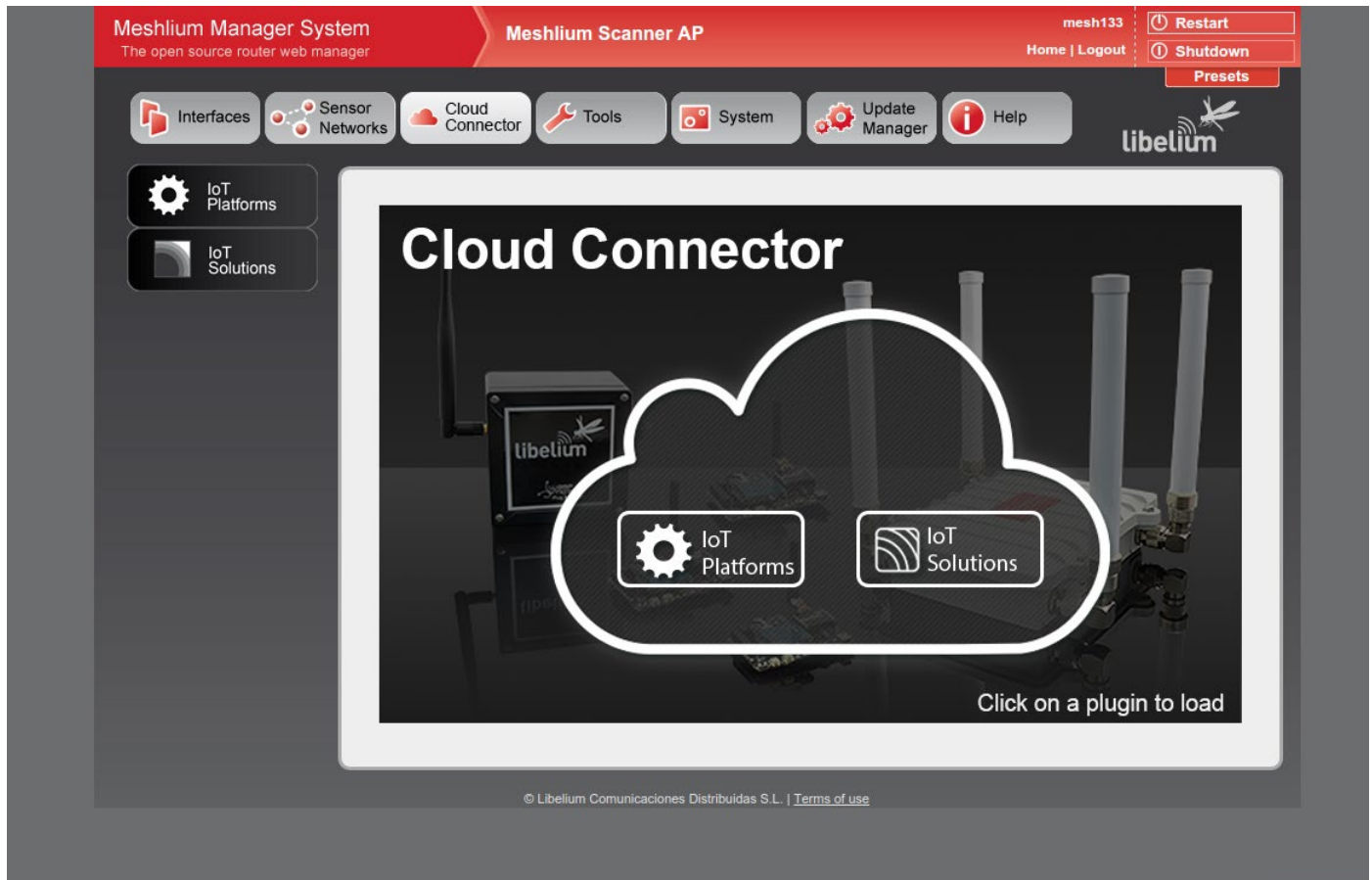


Figure : Cloud Connector main menu on the Manager System

17.1. IoT Platforms

17.1.1. Amazon IoT

Amazon Web Services IoT enables secure, bi-directional communication between Internet-connected things (such as sensors, actuators, embedded devices, or smart appliances) and the AWS cloud over MQTT and HTTP.

More information: <http://aws.amazon.com/iot/>.

With this plugin, Waspote sensor data can be directly integrated with Amazon AWS IoT broker.



Figure : Amazon IoT plugin

Amazon IoT plugin is located in:

Manager System > Cloud Connector > IoT Platforms > Amazon Web Services.

17.1.1.1. Register Meshlium in Amazon IoT

To register Meshlium in Amazon IoT, you have to follow the guide [Getting Started with AWS IoT](#). The guide explains how to create a thing and connect the device. It is important to annotate the configuration displayed and save the credential files when connecting the device. You will need these files and parameters later for the Meshlium configuration.

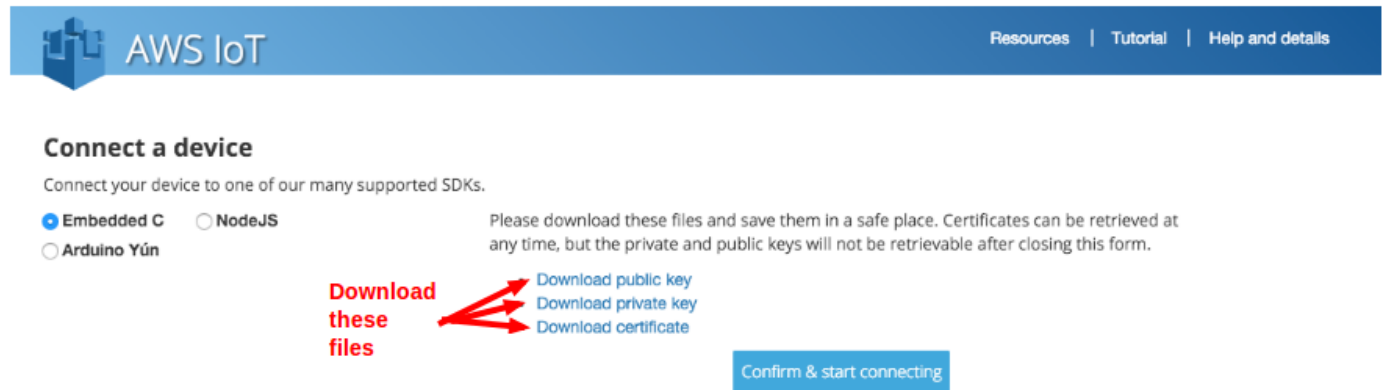


Figure : Save the credential files when connecting device

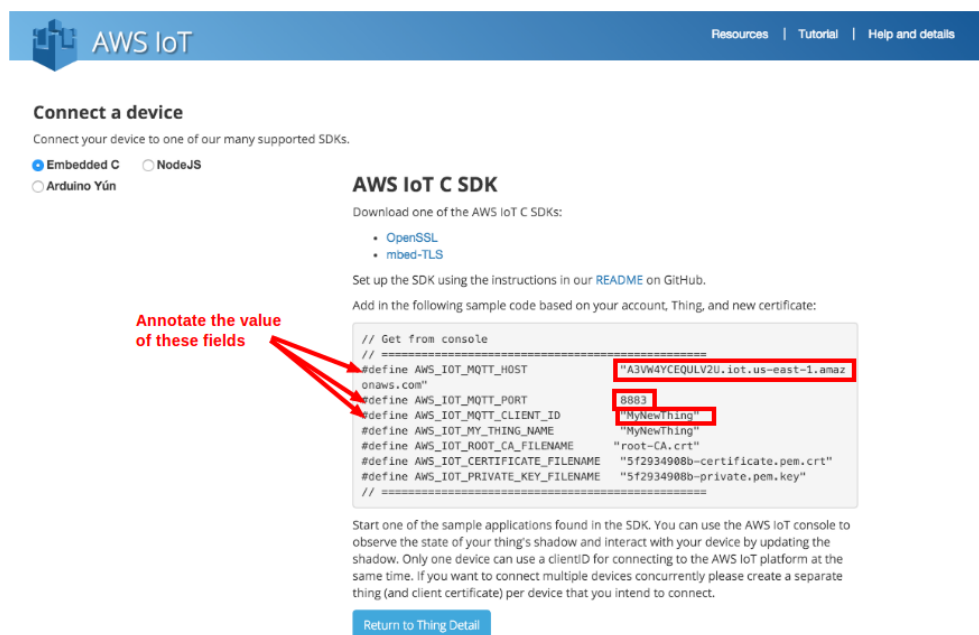


Figure : Annotate the value of the fields

17.1.1.2. Configuration

You will use the previously obtained configuration from the AWS IoT platform to certificate your Meshlium as a valid sender of MQTT messages.

In the Configuration panel, the user can set:

- Public key: User public key file previously downloaded.
- Private key: User private key file previously downloaded.
- Certificate: Certificate file previously downloaded.
- Host: AWS IoT MQTT host.
- Port: AWS IoT MQTT port.
- ClientID: AWS IoT Client identification.

- Message template: Data structure of your message. The user can use these wild-cards creating a customized content:
 - #ID#: Unique identifier for data
 - #MESHLIUM#: host name of the Meshlium
 - #ID_WASP#: Identifies the Wasp mote unit
 - #ID_SECRET#: Secret identifier
 - #SENSOR#: Identifies the sensor
 - #VALUE#: Value obtained from the sensor
 - #TIMESTAMP#: MySQL [TIMESTAMP](#) type ('YYYY-MM-DD HH:MM:SS' UTC)

IoT Platforms > Amazon Cloud

Configuration

Public key: No file selected.

Private key: No file selected.

Certificate: No file selected.

Host:

Port:

ClientID:

Message template:

powered by
amazon
web services

● **Amazon IoT synchronization status**

Figure : Amazon IoT configuration panel

17.1.1.3. Controlling synchronization

Once configured the server/broker, the user can launch the Meshlium Amazon IoT script (Start button). The program will search for the received frames on the local database, and will send them to the Amazon IoT platform via MQTT protocol. The status indicator displays the current state, saying "Running" or "Stopped".



Figure : Amazon IoT sender is running

You can stop the Amazon IoT program anytime clicking on the "Stop" button.



Figure : Amazon IoT sender is stopped

17.1.2. Amplía's OpenGate

OpenGate platform IoT, developed by Amplía Solutions, allows you to cover the entire circle related to data management IoT, collection, query, analysis ... while also managing the inventory of the devices of your solution.

For more information about Opengate, please see the link www.amplia-iiot.com.

17.1.2.1. Configuration

Inside the Amplía's plugin you can find the different fields that you must configure for using your Meshlium against OpenGate.

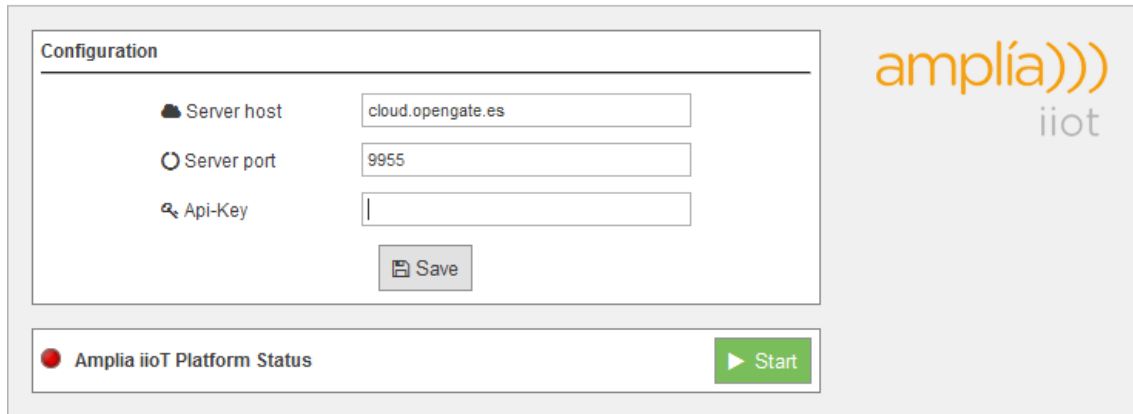


Figure : Amplía Cloud Connector configuration panel

- **Server Host:** You must enter the host name that you are going to use for collecting the Meshlium events.
- **Server Port:** The port where the host is accessible.
- **API-Key:** Security key used for validating the access to the Host.

Click on the **"Save"** button for storing the configuration fields.

After that, press the **"Start"** button, and you will start to receive data from the configured Meshlium.

If you want to stop the event sending, just press the **"Stop"** button.

In the OpenGate OSS web portal you could check the different values collected by Meshlium and by the Waspnote units which have sent messages using the configured Meshlium as gateway.

17.1.2.2. How to get your own API-key

For getting your own API-key you have to send a mail to <mailto:info@amplia.es> and Amplía Solutions will provide one for you. In the same mail send the serial number of your device for creating it in the OpenGate platform.

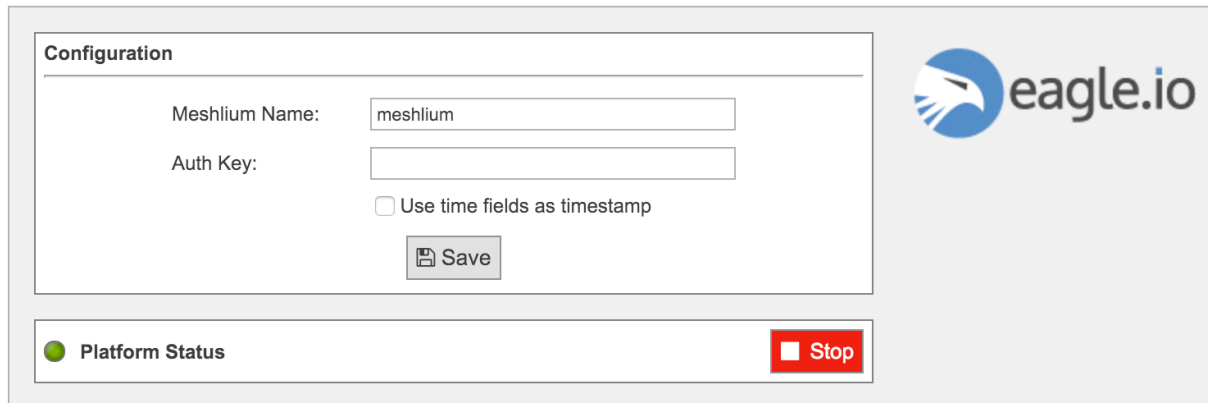
17.1.3. eagle.io

eagle.io is a hub connecting monitoring assets, engineers and decision makers. Acquire data in real-time from Meshlium and Waspnote devices, receive alerts for critical events, and share access with stakeholders. Transform your time-series data into beautifully presented, actionable information.

More information can be found at www.eagle.io.

17.1.3.1. Configuration

IoT Solutions > eagle.io



The eagle.io plugin is configured with the following three parameters:

- **Meshlium Name:** a name to help identify this device (required parameter).
- **Auth Key:** optional secret key; if this is defined then the same key will be required when configuring the device within eagle.io as a data source. If this key is not defined, then the Meshlium device ID is sufficient to identify the device within eagle.io configuration.
- **Use time fields as timestamp:** if this box is checked, then any time field contained in a Meshlium database record will be used as the eagle.io timestamp for the record. If this box is not checked, the data reception time of the record will be used as the eagle.io timestamp.

After changing any of these parameters, save the configuration by clicking the “Save” button, then restart the eagle.io synchronization (if it is currently running) by clicking the “Stop” button followed by the “Start” button.

17.1.3.2. Controlling synchronization

The synchronization will be done in batches of 200 records at a time, so the system is not overloaded. The time between batches is 60 seconds. This means that when synchronization is first started on a device with many existing records in the database, it may take some time for all the records on the device to be synchronized with eagle.io.

When the synchronization is not running (red status indicator), it can be started by clicking the green “Start” button.



Figure : Eagle.io start button

When the synchronization is running (green status indicator), it can be stopped by clicking the red “Stop” button.



Figure : Eagle.io stop button

17.1.4. Esri

ArcGIS is a complete spatial information platform provided by **Esri**, that allows to create, analyze, store and spread data, models, maps and 3D globes. It can be accessed via desktop application, browser or handsets. ArcGIS is targeted at GIS professionals, location analysts and developers that want to create their own applications based on geographical data.

More information: <http://www.esri.com/products>

Waspnote sensor data could be integrated into your existing maps and ArcGIS applications following the configuration steps described for both mechanisms, the simple Feature Server and the advanced ArcGIS Online service.

17.1.4.1. Feature Server

17.1.4.1.1. Security

The first section in the first tab of the plugin contains the security configuration.

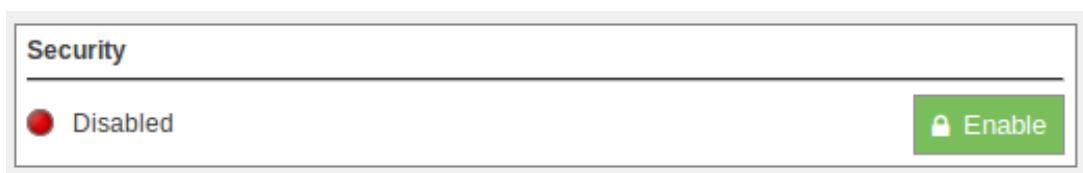


Figure : Security layer in ArcGIS

By default, it is disabled; click on the button to enable it.

Users List

In “Users list”, the user can manage the users that can request tokens to make connections with ArcGIS in a secure way.

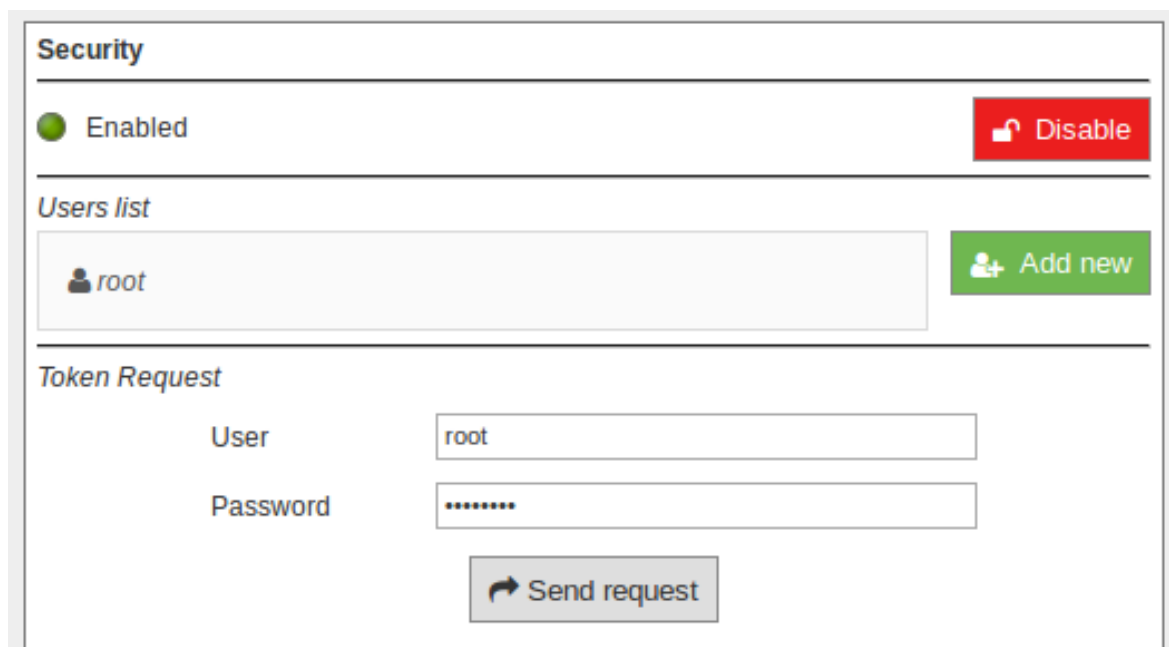


Figure : Security users list in ArcGIS

To add a new user, click on the “Add New” button.

Then fill up the information (user and password), and click “Add”.

To modify or delete a user, click on the user name and a user view will be deployed.

Users list

Users list

root

User (*)

Old Password (*)

New Password

Delete Save

Add new

Figure : Modifying security configuration in ArcGIS

Then the user can change the password or delete the user. For both operations, the user has to enter the old password.

Token request

To request a token, enter the user and password and click on “Send Request” button.

Token Request

User

Password

Send request

Received Token:

```
{"token":"0b8ec3cfcda1c9b8042ae6a1afc61936","expires":1461144103}
```

Figure : Security token request in ArcGIS

Then, the “Received Token” box will show the token and its expiration time.

17.1.4.1.2. Service Info

REST Service Info shows the information needed to operate with any ArcGIS software.

Service info

Explore REST Services Directory View everything in ArcGIS.com

Figure : REST Service Info section in ArcGIS

- A link to **REST Services Directory**: a view of the REST API in HTML format. Here the user can browse the contents of a Meshlium-ArcGIS Server and obtain information such as service metadata and supported operations that can be useful in developing applications.
- A link to www.ArcGIS.com: view Meshlium, Waspnote and sensors data directly on the free ArcGIS Online map viewer.

With the ArcGIS Online map viewer, you can easily explore, visualize, and share GIS information. Use it to directly access basemaps and other content.

To view Meshlium, Waspnote and sensors data from any ArcGIS software, open your ArcGIS software, click on “Add Layer from Web” and enter:

`[MeshliumIP]/meshlium/rest/services/Libelium/FeatureServer`

Note: [MeshliumIP] must be a static IP address, hired by the user. It cannot be 192.168.x.x.

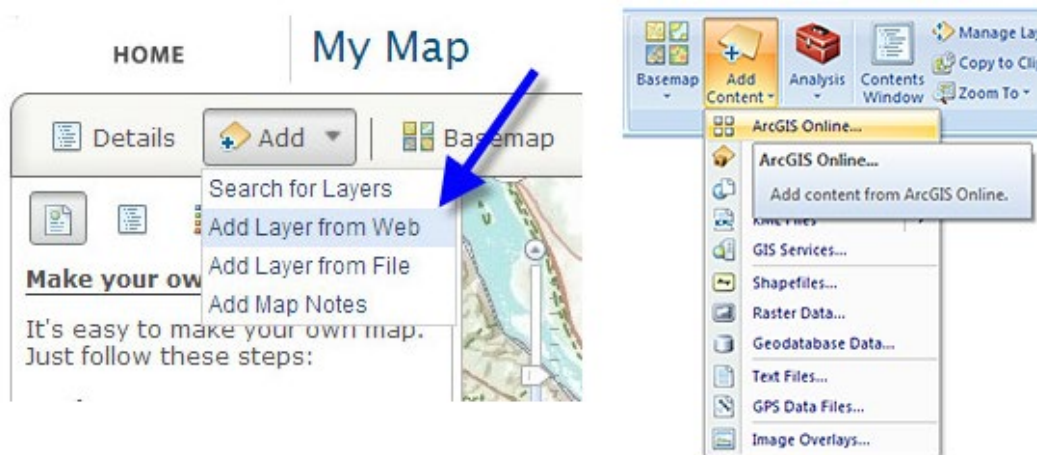


Figure : ArcGIS software, add Layer

Once the layer is added you will see the information parsed and stored in the Meshlium database on the ArcGIS software.

See chapter “Capturing and storing sensor data” for more information.

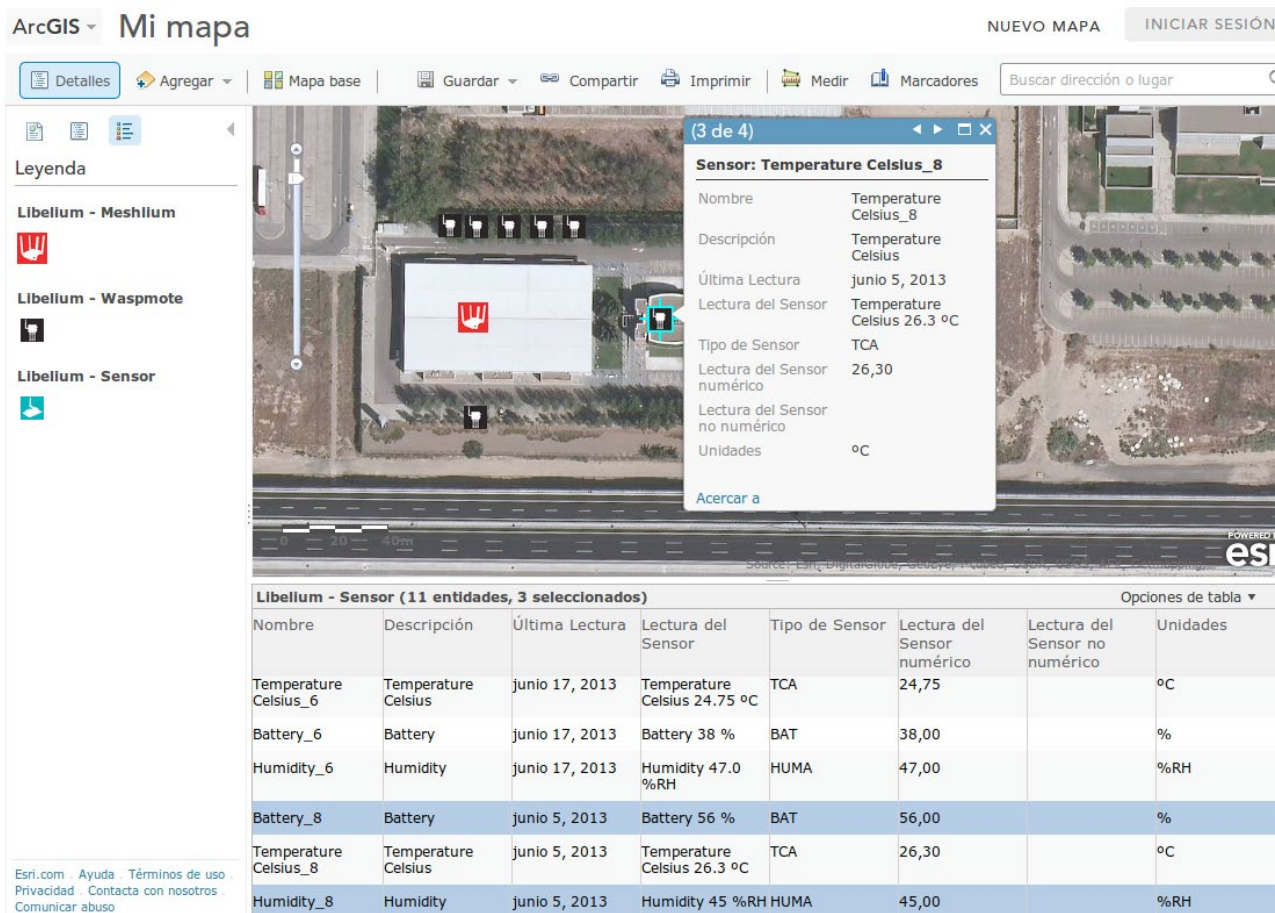


Figure : www.ArcGIS.com viewer with Meshlium layer

17.1.4.2. ArcGIS Online

We can configure in this form all the parameters needed to connect and send data to the ArcGIS Online platform.

IoT Platforms > Esri

Figure : ArcGIS Online configuration

The parameters to setup are:

- **esri_user:** User for the Esri ArcGIS online platform.
- **esri_password:** Password for this Esri user.
- **esri_service_name:** Name of the service which will receive the data.

Clicking on the “Save” button, this setup is sent to the ArcGIS online service.

Clicking on the “Start” button enables the Esri Cloud Connector to send data periodically to the ArcGIS Online service previously configured. A “running” status is displayed on screen showing that the Cloud Connector is sending data.



Figure : ArcGIS Online “Start” button

Clicking on the “Stop” button will disable the Esri Cloud connector so Meshlium device stops feeding the ArcGIS Online service with data.



Figure : ArcGIS Online “Stop” button

17.1.4.2.1. Check the Feature Server in ArcGIS Online

In order to check that data is arriving to ArcGIS Online, you should login in the platform:
<https://www.arcgis.com/home/signin.html>

Click on the option named "Gallery" and you should see a new Feature Server with the name that you provided in the Meshlium configuration plugin:

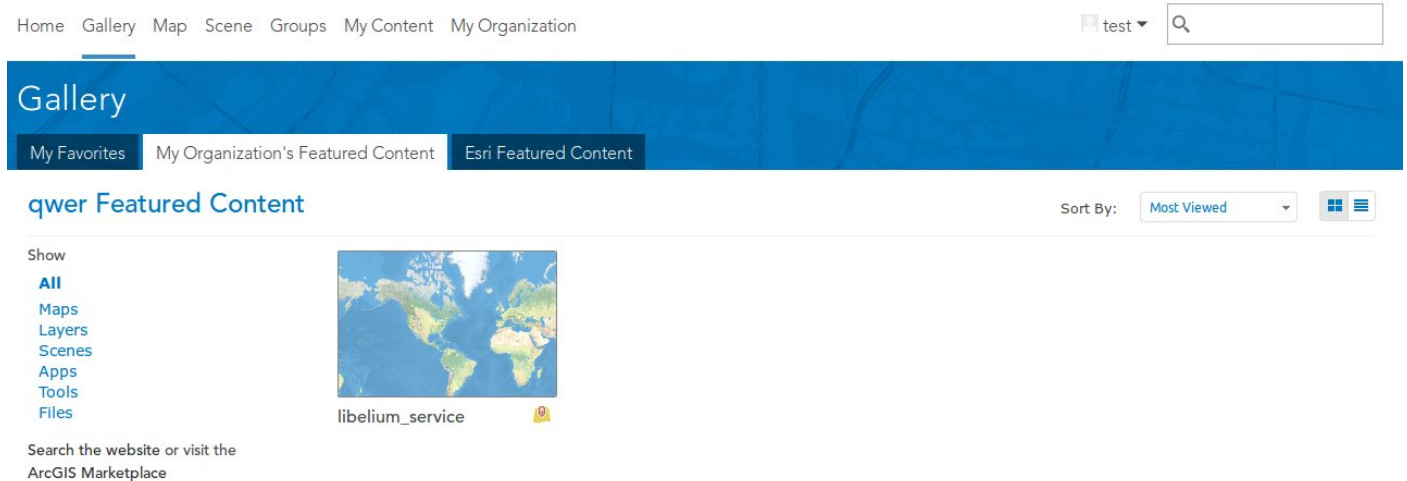


Figure : ArcGIS Gallery

Opening the new content, you should see a map where each layer is one sensor type available in your project. Clicking on the table icon, all the data collected for this type of sensor will be displayed.

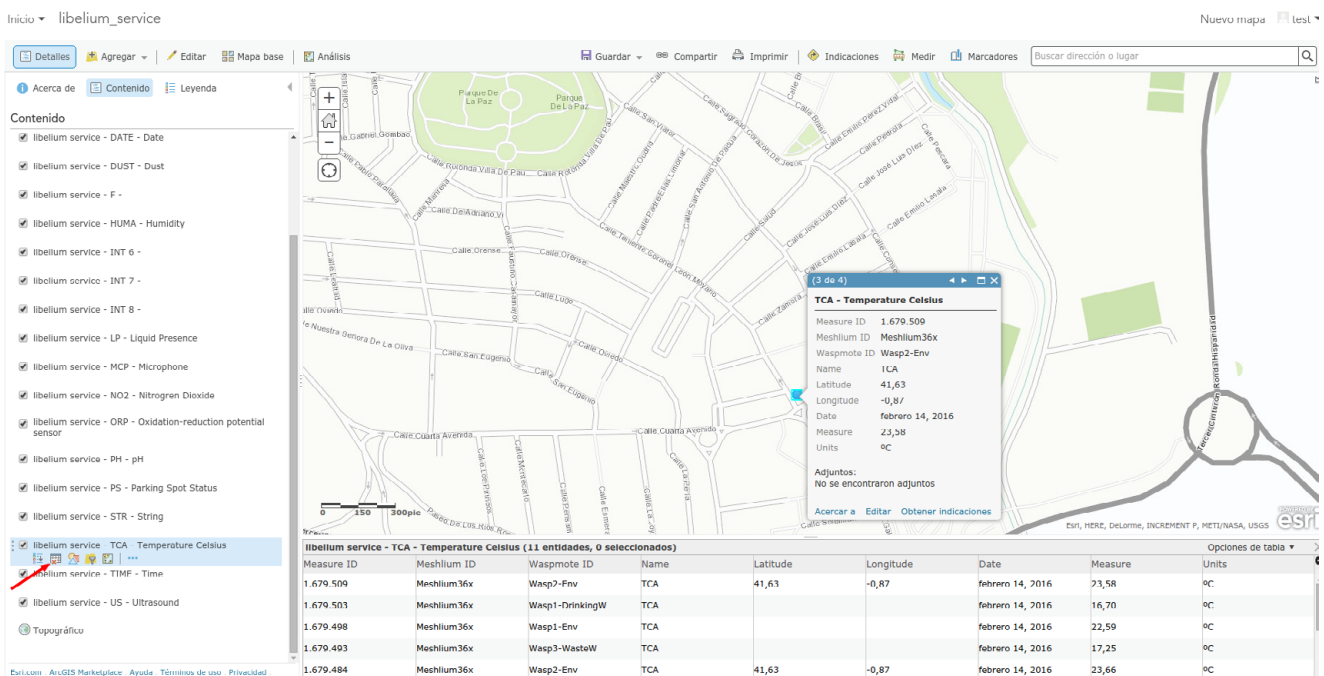


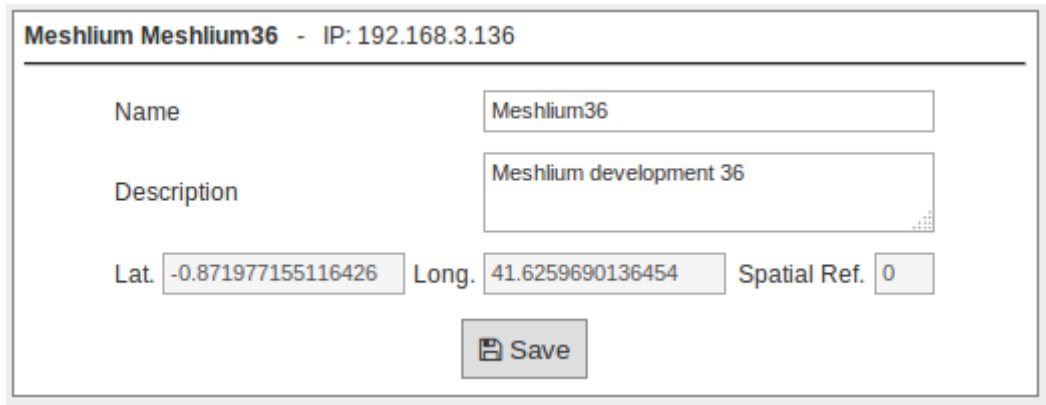
Figure : ArcGIS sensor map view

At this point, it is possible to use this data to create new maps, collaborative apps or analytics making use of the complete array of services provided by ArcGIS Online: <https://developers.arcgis.com/en/>

17.1.4.3. Devices

17.1.4.3.1. Meshlium

In the Meshlium section, the user can set and modify the name and description of the Meshlium.



Meshlium Meshlium36 - IP: 192.168.3.136

Name: Meshlium36

Description: Meshlium development 36

Lat. -0.871977155116426 Long. 41.6259690136454 Spatial Ref. 0

Save

Figure : Meshlium info in ArcGIS

17.1.4.3.2. Waspnotes

In the Waspnotes section, the user can manage the Waspnote units which are sending information to Meshlium.



Waspnotes

Wasp2-Env

Wasp1-Cities

Add new

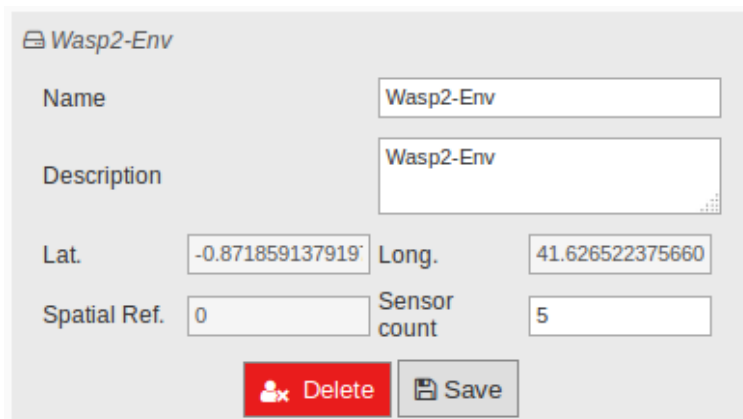
Figure : Waspnotes list in ArcGIS

To Add a new Waspnote, click on "Add new". Then fill up this information:

- Name: The Waspnote name. **Must match with the Waspnote identifier used with the frame.** See chapter "Capturing and storing sensor data" for more information.
- Description: A description of that Waspnote unit.
- Sensor count: Number of sensors on that Waspnote. **Must match with the number of fields of the frame.** See chapter "Capturing and storing sensor data" for more information.

And click on the "Add" button.

To modify a Waspnote, click on the Waspnote name for showing the attributes view.



Wasp2-Env

Name: Wasp2-Env

Description: Wasp2-Env

Lat: -0.871859137919 Long: 41.626522375660

Spatial Ref: 0 Sensor count: 5

Delete Save

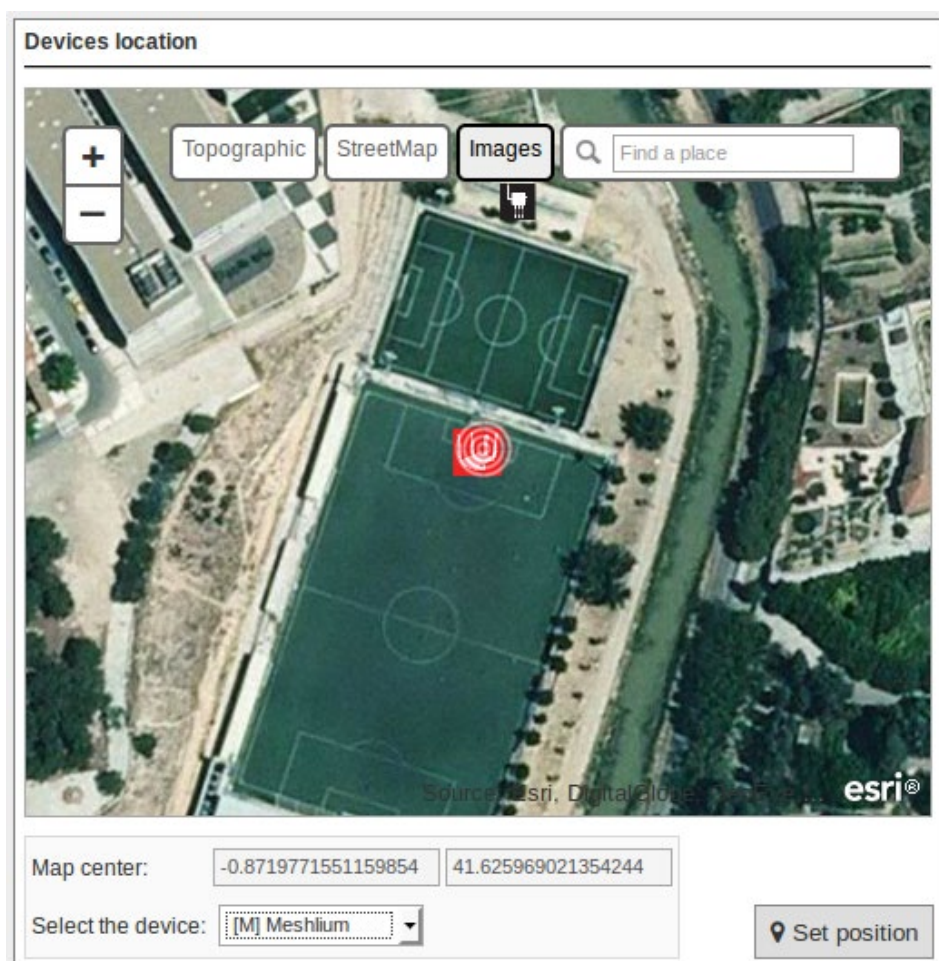
Figure : Modify Waspnote in ArcGIS

Then the user can modify the name, description, and sensor count information. To save the properties, click on “Save”.

To delete this Waspnote unit, click on “Delete”.

17.1.4.3.3. Devices location

In the section Devices location, there is a viewer where the user can see Meshlium and Waspnote located on a map.



Devices location

Topographic StreetMap Images Find a place

Map center: -0.8719771551159854 41.625969021354244

Select the device: [M] Meshlium

Set position

Figure : Devices location in ArcGIS

To change the location of the devices, center the map on the desired location, select the device, and click on “Set Position”.

17.1.5. Extunda

Extunda IoT platform (<http://www.extunda.com/>) is a horizontal platform which also enables vertical applications. Libelium Smart Cities, agriculture and various devices are ready to be launched for service over Extunda IoT platform. The sensor data can be gathered, analyzed, stored and reported so the users can interpret and develop actions based on online data.

Extunda uses MQTT structure for the integration of your Meshlium devices to its platform easily. Therefore, the connector will easily send the sensor data to Extunda platform.

17.1.5.1. Configuration

IoT Solutions > Extunda Cloud

Configuration

IP Address:

Port number:


User:

Password:


Topic template:

Message template:

Save



extunda
Connecting the dots

 MQTT Sender

Start

Figure : Extunda cloud connector configuration panel

- **IP Address / URL:** The IP address or the URL will be provided to you by Extunda.
- **Port Number:** This is the port which Extunda server is listening for connections.
- **User Name & Password:** This is the Extunda Server username & password to be used for connecting to Extunda servers. This information will be provided by Extunda.
- **Waspnote ID:** When you login to your Extunda IoT platform account with your username and password, you are authorized to define and match your Waspnote with a specific definition (ie. Istanbul_gases_1). Your Waspnote data will be transferred to the server as in the above message template.

17.1.5.2. Controlling synchronization

The synchronization will be done for all data that has not been synchronized in the Sensor Parser table each time. You can start and stop the data synchronization to the Extunda service. In the interface you can see an indicator of whether the status service is running or not. If you click on "Start", the synchronization will begin.



Figure : Extunda start button

You can stop the synchronization at any moment clicking on the "Stop" button.



Figure : Extunda stop button

17.1.6. IBM Bluemix

IBM Bluemix is a cloud [platform as a service \(PaaS\)](#) developed by IBM that gives a wide scope of services to use the cloud, one of them is based on MQTT communications. This is a great alternative if the user do not want to build his own MQTT server.


17.1.6.1. Configuration

Configuration options are shown in the M2M Platform menu, enlarging the IBM Bluemix MQTT section. You will notice that the configuration for this plugin is very straight-forward, you have most of the needed parameters on the IBM Bluemix web panel:

IoT Platforms > IBM Bluemix

Configuration

Organization:
API user:
API password:
Event ID:
Interval:




 **Cloud Synchronization Status**

Figure : Configuring IBM Bluemix in Meshlium

- **Organization ID:** Identifier of your organization; you can get it from the platform credentials.
- **API user:** User generated in the API platform section.
- **API password:** Password generated in the API platform section.
- **Event ID:** Used to configure the event where you want to send the information. If you do not know what to type in this field, you can use 'eid'.
- **Interval:** Used to delay the communication after a bunch of messages were sent.

17.1.6.2. Controlling synchronization

You can stop or start the IBM Bluemix synchronization process anytime, hitting on the buttons "Start" and "Stop". Then, the status indicator displays the current state, saying "Running" or "Stopped".

 **Cloud Synchronization Status**

Figure : IBM Bluemix synchronization service is running

You can stop the synchronization anytime clicking on the "Stop" button.

 **Cloud Synchronization Status**

Figure : BM Bluemix synchronization service is stopped

More information can be found on this Recipe we created for IBM:

<https://developer.ibm.com/recipes/tutorials/bluemix-configuration-guide-for-meshlium/>

17.1.7. IoT-Ticket

IoT-Ticket is one of the world's most complete, advanced and easy to use Industrial Internet of Things platforms with over 1.6 million users mainly in the energy and mobile machinery industry. Using IoT-Ticket you can build IoT applications in your web-browser in minutes, no plug-ins required. You can create dashboards, reports, analytics or augmented reality based on big-data collected from your things.



Figure : IoT-Ticket panel examples

Some benefits offered by IoT-Ticket:

- **Complete, up-to-date solution.** IoT-Ticket is a complete remote management system which includes the electronics, software and server. The platform is continuously developed further with new features and options.
- **Easy to get started and integrated.** You can use the platform as a service (SaaS or PaaS) or deploy to your own servers. We can integrate IoT-Ticket with any of your other information systems.
- **Flexibility and choice.** Use the whole IoT-Ticket platform or part of it. Use IoT-Ticket specific electronics or use your own, already deployed, electronics. Easy to use API in many programming languages allows a huge selection of devices to be easily connected.
- **Easy to use and Customizable.** The IoT-Ticket web dashboards allows you to be up and running in minutes using only your web browser. IoT-Ticket can be customized to meet your unique needs, even the look and feel can be made to match your corporate brand identity.

More information can be found at www.iot-ticket.com

17.1.7.1. IoT-Ticket Meshlium integration

Once Libelium's Cloud Connector has been configured, all your available data will show up automatically in your IoT-Ticket web-based dashboard / report designer from where you can easily design Internet of Things applications.

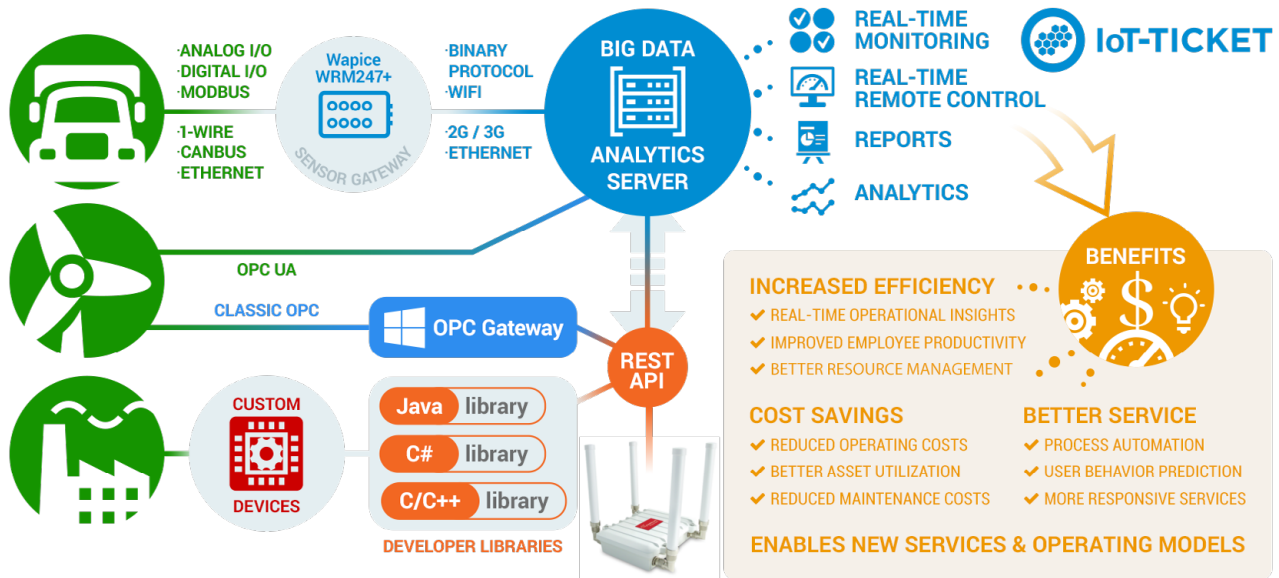


Figure : IoT-Ticket Meshlium integration

The IoT-Ticket cloud connector settings can be found under the IoT section of the browser-based Meshlium Manager System. The configuration is split into three parts Login Configuration, Connector Settings and Waspnote Filtering, as well as a section for information about the current status of the connector with controls to start and stop the program.

The screenshot shows the IoT-Ticket plugin panel with the following sections:

- Login Configuration:**
 - Service Owner:
 - Service Key:
 - Server:
 - IoT Device Name:
- Connector Settings:**
 - Launched at startup: ☐
 - Time between transmits (s):
 - Max. size of MySQL query:
 - Sleep between queries (s):
 - Log file max. size (Kbytes):
 - Max num. of log files:
 - Buttons: ☒ Validate
- Waspnote Filtering:**
 - Waspnote filtering active: ☐
- IoT-Ticket Synchronization Status:**
 - Status: ● IoT-Ticket Synchronization Status
 - Buttons: [View log files](#)

Figure : IoT-Ticket plugin panel

The Login Configuration section sets up the information for your IoT-Ticket account, and consists of four parts:

- **Service Owner/Key** gives the username and password of the my.iot-ticket.com service account to which you wish to connect your Meshlium device.
- **Server** specifies the IoT-Ticket server to use, by default my.iot-ticket.com.
- **IoT Device Name** is a read-only field showing the IoT-Ticket device name used for that Meshlium unit. It is set when the connector is started and is empty if no name has yet been set (in such a case, use browser “Refresh” after connector has started to see the name).

The Connector Settings section has parameters for the operation of the connector itself. These values affect time between updates to IoT-Ticket as well as size of transmitted batches of data. More frequent data updates may come at the cost of increased system resource usage.

- **Launch at start.** This checkbox indicates whether the connector is set to start automatically when the Meshlium is powered on.
- **Time between transmits** gives the minimum elapsed time between transmissions to IoT-Ticket. Values less than 60 seconds may consume high system resources.
- **Sleep between queries** is the time the program sleeps between SQL queries, in order to conserve resources.
- **Max. size of SQL query** is the maximum number of results for a single SQL query to the Meshlium database. Values greater than 200 may lead to high system load.

The section also allows for configuring connector logging:

- **Log file max. size (Kbytes).** The maximum size of a single log file in kilobytes.
- **Max. num. of log files** is the number of log files that can be written before the logging handler begins overwriting the first.

17.1.7.2. Save, Load and Verify

These buttons allow saving, loading and validating entered configuration data to a local file on the Meshlium disk which is read by the connector. The saved data includes both the Login and Connector settings as well as any entered Waspmote filtering rule (see “Waspmotes” section below).

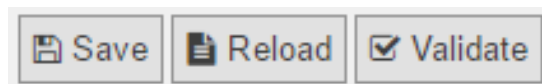


Figure : Save, Load and Verify buttons

- **Save** validates the data entered into the form and saves to it to disk.
- **Reload** reads data back from disk, erasing any fields that have been changed since last save.
- **Validate** runs a check that entered fields are of the correct type and connects to IoT-Ticket to check the entered username and password. If verification fails for a field, it will be marked in red and an error message appears.

17.1.7.3. Validation of settings

Configuration settings are validated to make sure the entered data fields make sense (e.g. numeric fields such as sleep and query size must be numbers). Additionally, the validation will issue warnings if any parameters might cause high system load on Meshlium.

Additionally the “Validate” button checks entered login information with the IoT-Ticket server and shows a warning if settings are incorrect and a green confirmation message if they are correct.

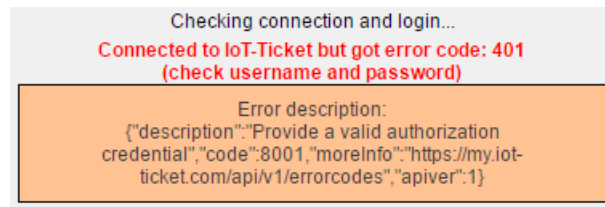


Figure : Validation error

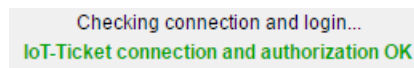


Figure : Validation success

17.1.7.4. Waspnote filtering

This section allows filtering of which Waspnote data is synchronized to IoT-Ticket. The section is enabled by ticking the Waspnote filtering active checkbox.

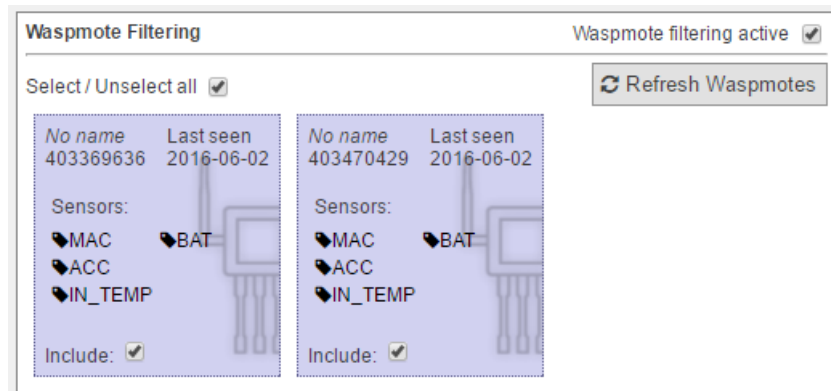


Figure : Waspnotes list

- Refresh Waspnotes reloads the list of Waspnotes from the Meshlium database
- Select / Unselect All allows for quick selection or deselection of all present Waspnotes

The Waspnote infobox contains the following values:

- Name in the top-left shows the name a Waspnote broadcasts to the Meshlium with its readings (this is set in Waspnote code). If not set, No name is displayed instead.
- Last Seen is the last date at which a sensor entry was sent from the Waspnote to the Meshlium.
- Sensors is a list of sensors present on that Waspnote device. Only the latest detected set is displayed here, in case sensors are changed. Full names may be seen by hovering the mouse over the abbreviated names in the list.
- Include specifies whether the Waspnote should be included in the data transmitted by the connector. Deselected Waspnotes have their info box greyed out.

17.1.7.5. Synchronization status

This section allows the user to start and stop the connector and displays information about its current status. When the connector is off, the indicator marker is red.

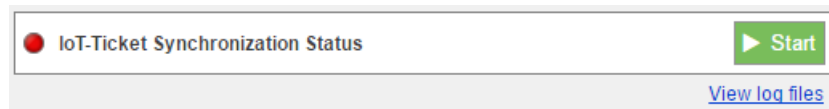


Figure : Start button

After clicking "Start", the connector shows a startup sequence, and when finished the running status will be indicated by the status icon turning green. The start button becomes a red "Stop" button.

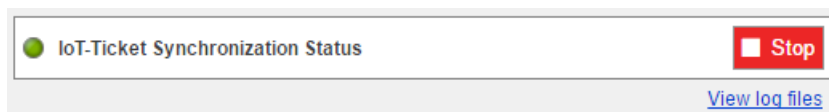


Figure : Stop button

The link "View log files" will allow you to see the status of the running connector via its log files. A filtered set of this logging data will also be available as a data node in your IoT-Ticket enterprise.

17.1.7.6. IoT-Ticket view

Once the connector is running you can use your web browser to see the Meshlium data coming into your IoT-Ticket Dashboards and Enterprise Manager on my.iot-ticket.com.

In your IoT-Ticket enterprise the Meshlium device will be viewable as an IoT-Ticket device under your enterprise and can now easily be used in Enterprise Dashboards to create views of your incoming data, even mixing it with data coming from other IoT-sources.

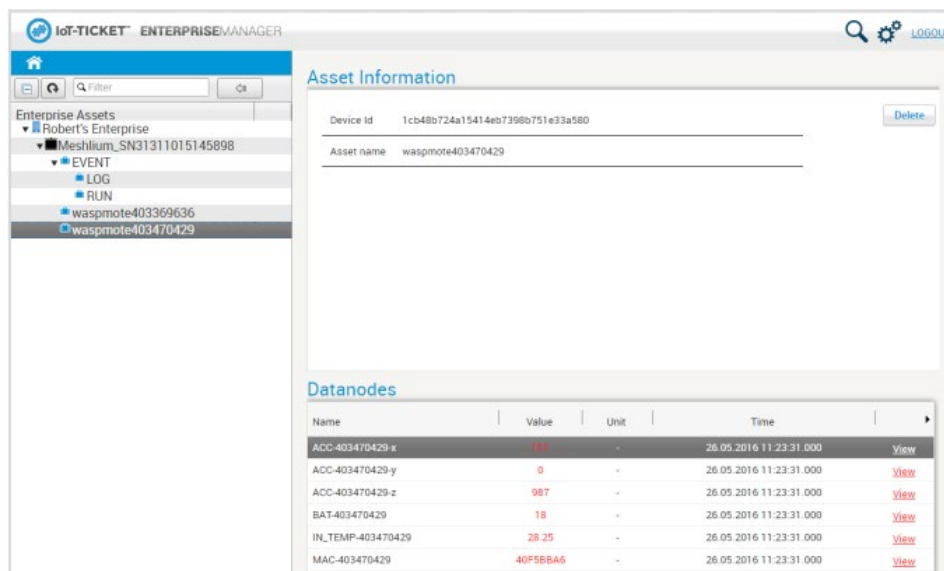


Figure : IoT-Ticket panel

You can now configure your own dashboard with sensor data and have it up and running in a matter of minutes.

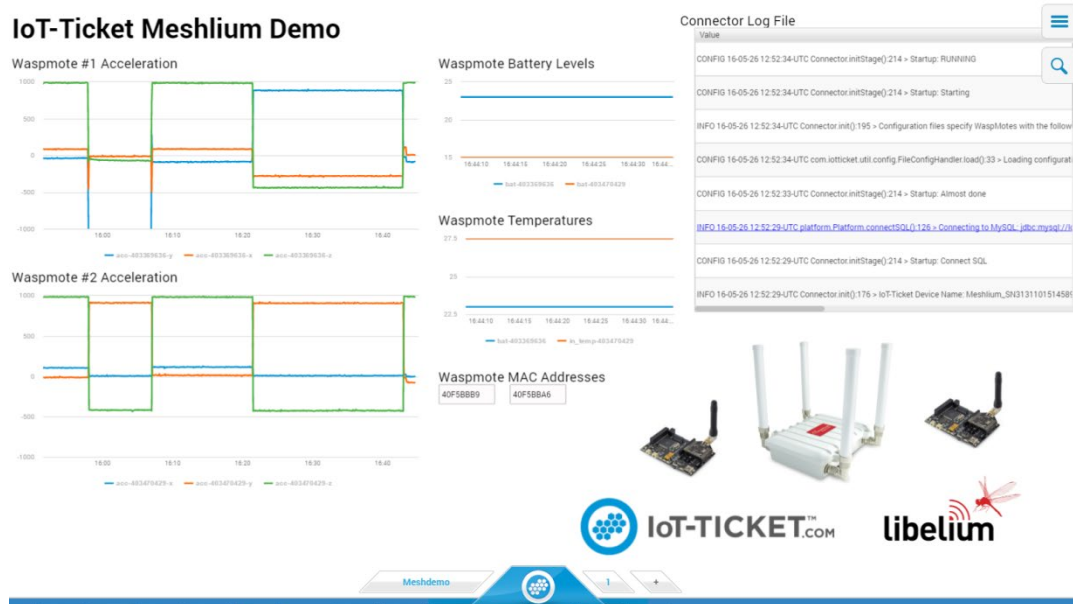


Figure : IoT-Ticket panel

17.1.8. Microsoft Azure Event Hubs

Azure is a cloud platform provided by Microsoft. This platform has a lot of services for communications between machines and devices. Service Bus and Event Hubs are 2 different tools to insert data into Azure, but Events Hubs is specially designed to offer enhanced services for IoT applications. This section focuses on Event Hubs, we can understand this technology as another way to send short messages via HTTP REST request. Event Hubs is part of Service Bus, it is based on simple M2M messages.

For more information about Event Hubs, see the following link: <https://azure.microsoft.com/en-us/services/event-hubs/>

17.1.8.1. Configuration

In the Azure Event Hubs configuration plugin, you can setup all the information needed to connect to this platform from Meshlium.

IoT Platforms > Microsoft Azure Event Hubs

Configuration

Namespace:

Directive name:

Directive key:

Name:

Template file:

```
"id": "#ID#"
"id_wasp": "#ID_WASP#",
"id_secret": "#ID_SECRET#",
"sensor": "#SENSOR#",
"value": "#VALUE#",
"datetime": "#TS("c")#"
"datetime2": "#TS("r")#"
"datetime3": "#TS("Y:m:d","1231234"-H:i:s)#"

```

Save

Cloud Synchronization Status

Figure : Configuring Azure Event Hubs in Meshlium

- **Namespace:** Name of the space created in the Azure service cloud.
- **Directive name:** Name of the directive created in Azure.
- **Directive key:** Key of the directive associated to the previous name.
- **Name:** Name of the Event Hub established in Azure.
- **Template file:** Users can define their own data structure using these wild-cards (as you can see in the previous figure):
 - #ID# : Unique identifier for data
 - #ID_WASP# : Identifies the Wasp mote unit
 - #ID_SECRET# : Secret identifier
 - #SENSOR# : Identifies the sensor
 - #VALUE# : Value obtained from the sensor
 - #TS("c")# : Date with custom format. The parameter passed in this wild-card corresponds to the same ones you can use in PHP date function (see format parameters in <http://php.net/manual/es/function.date.php#refsect1-function.date-parameters>)

17.1.8.2. Controlling synchronization

Once you have saved the configuration, you can start sending your data via Event Hub to your Azure Cloud by pressing the “Start” button. You will notice about it because the screen shows a spinning wheel when the process starts and displays a “running” status.



Figure : Azure Event Hubs synchronization service is running

If you want to stop this process just press the “Stop” button. You can start/stop this process whenever you want.



Figure : Azure Event Hubs synchronization service is stopped

17.1.9. Microsoft Azure Service Bus

Azure is a cloud computing platform by Microsoft. It offers services of computing, virtual machines, databases, storage, messaging and many others.

Discover Microsoft Azure here:

<http://azure.microsoft.com/en-gb/overview/what-is-azure/>

Azure Service Bus is a message system that allows to exchange information between different systems, inside and outside Azure platform, independent of the architecture, operating system or complexity level.

All the information about Service Bus can be found here:

<http://azure.microsoft.com/en-us/documentation/services/service-bus/>

In particular, Meshlium implements the use of queues, a FIFO system where one or more senders can push messages to the queue and an application or service can receive this data.

More information about queues:

<http://azure.microsoft.com/en-us/documentation/articles/service-bus-dotnet-how-to-use-queues/>

17.1.9.1. Setup in Azure - Creating NameSpace

A service namespace is a scoping container for a set of Service Bus messaging entities. The boundary formed by service namespace is expressed in the form of URI subdomains, to distinguish between different domains of control in the naming system. The URL for a namespace will have the form:

```
namespace.servicebus.windows.net
```

In order to use Service Bus with Meshlium, we are using the Java API distributed by Microsoft. This API is still not compatible with the latest authentication method used in Azure (SAS). You need to use ACS authentication. If you create the namespace in the web administration panel, only SAS authentication method is available, so you have to create your namespace with a command line tool to enable ACS authentication.

In order to do this process you have first to install the Client Line Interface tool. Updated instructions and files can be found here:

<http://azure.microsoft.com/en-us/documentation/articles/xplat-cli/>

Once installed the CLI tool, you have to download your account. Do this with the commands: (Linux commands used in this guide, check documentation for this process in other platforms.)

```
azure account download
```

This will return a url. Put this url in your browser and login in your Azure account. Then you will be able to download the credential file.

Then you have to import the file with the command:

```
azure account import <path_to_file>
```

You will be then enabled to manage your Azure settings from the CLI.

In order to create the namespace for Service Bus, use the command:

```
azure sb namespace create <your_namespace>
```

Select your region from the list and the namespace will be created.

You can then access the web manager and your namespace will be there. Go to **Connection Information** at the bottom of the page.

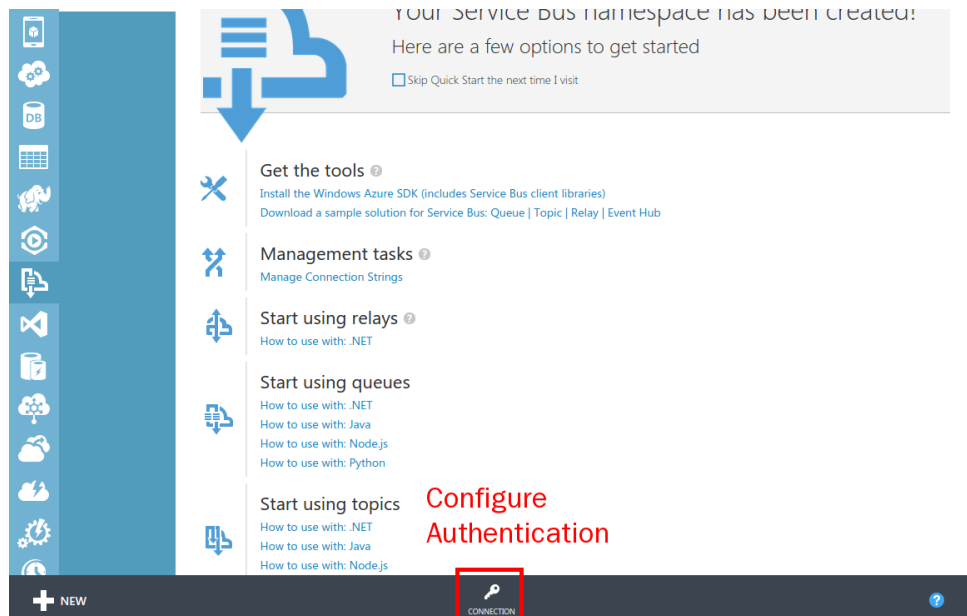


Figure : Getting Azure credentials, 1

There you will be able to see the credentials and to access to the **ACS Management Portal**.

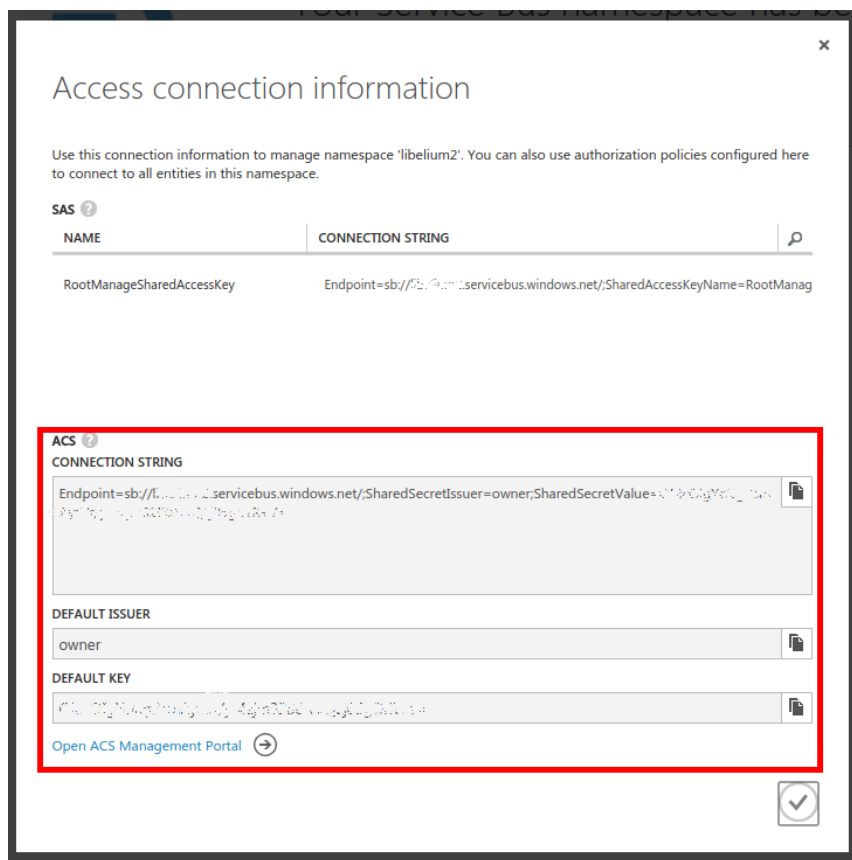


Figure : Getting Azure credentials, 2

These credentials have to be set in Meshlium Manager System in order to use Azure Service Bus synchronization.

17.1.9.2. SETUP in Meshlium

Meshlium can send sensor data to an Azure queue located in a specific namespace. This process needs a basic configuration in Manager System. You need to setup the namespace and the credentials of your Azure account.

With the parameters obtained in the previous section “Setup in Azure - Creating NameSpace” we can fill the form in the plugin of Meshlium Manager System located in:

ManagerSystem → Cloud Connectors → M2M platforms → Azure Service Bus

IoT Platforms > Microsoft Azure Service Bus

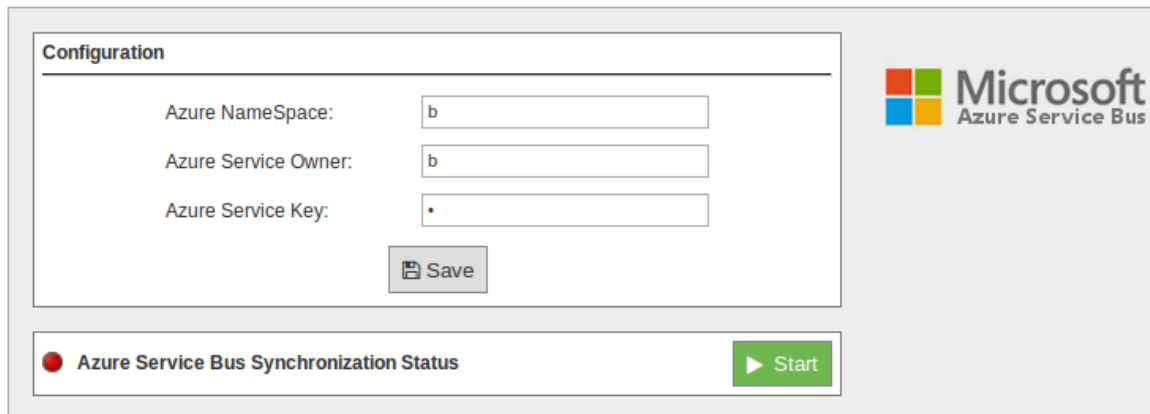


Figure : Azure setup example on Manager System

17.1.9.3. Synchornization process

Once configured, you can start the synchronization process pushing **start** button.

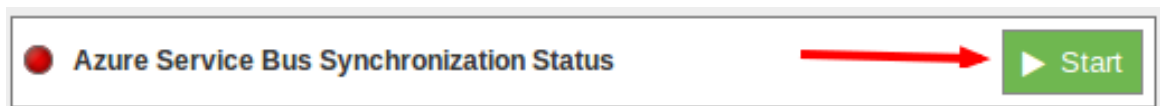


Figure : Azure Start button

The synchronization process will create a queue for each waspmote present in the database, with the name **waspmote_id**, and will send every data in a message with the json format:

```
{
  "id_wasp": "SmartWater",
  "id_secret": "366360762",
  "sensor": "ORP",
  "value": "0.380",
  "datetime": "02/12/2014T12:18:57+0000"
}
```

Any application or system can now subscribe to the queue and receive the data.

The synchronization is done with an interval of 60 seconds and it is limited to 100 data per iteration.

Each data is sent in a single message. This is done with the purpose of fragmenting the messages as much as possible. This enables the use of advanced features of Service Bus queues, such as filtering, allowing reading each sensor separately for an easier representation. This way, in the same queue, you can set a process reading from a queue only the data that matches one specific sensor, letting other sensor data for other processes specifically designed for these other sensors.

The json message has been created for the purpose by Libelium development team, so there is an open chance of designing other schema or representation if any customer wants a personal development with this system.

The synchronization process can be stopped pushing the “stop” button of the web interface:

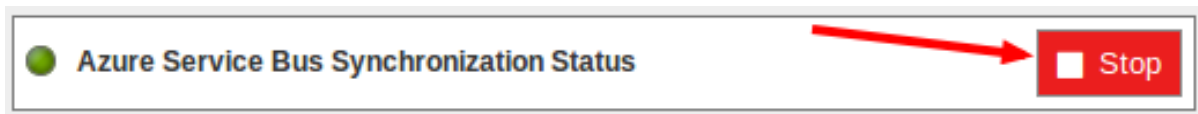


Figure : Azure Stop button

17.1.10. MQTT

MQTT is a publish/subscribe, extremely simple and lightweight messaging protocol, designed by IBM for constrained devices and low-bandwidth, high-latency or unreliable networks, where battery power is critical. Due to its features of delivery assurance and bandwidth reduction, MQTT is being used by some Cloud platforms such as IBM or Carriots, which means that Waspote data can be stored inside them or in any other one based on this protocol.

More information: <http://mqtt.org/faq>

With this plugin, Waspote sensor data can be directly integrated with a MQTT broker.



Figure : MQTT plugin

MQTT plugin is located in:

Manager System > Cloud Connector > IoT Platforms > MQTT Solutions.

17.1.10.1. Server/Broker Configuration

The broker is a key agent in MQTT protocol. The broker is a server which receives all the frames and distributes each one of them to the subscribers clients.

In Server/Broker Configuration, the user can set:

- **IP Address:** Server IP address.
- **Port number:** Server port number.
- **User:** Server user name to log in the MQTT system.
- **Password:** Server password to log in the MQTT server.
- **Topic template:** Topic of your message. The user can use these wild-cards creating a personalized structure:
 - #MESHLIUM# : Identifier for Meshlium
 - #ID# : Unique identifier for data
 - #ID_WASP# : Identifier for Waspote
 - #SENSOR# : Sensor identification
- **Message template:** Data structure of your message. The user can use these wild-cards creating a customized content:
 - #ID# : Unique identifier for data
 - #ID_WASP# : Identifies the Waspote unit
 - #ID_SECRET# : Secret identifier
 - #SENSOR# : Identifies the sensor
 - #VALUE# : Value obtained from the sensor
 - #TS("c")# : Date with custom format. The parameter passed in this wild-card corresponds to the same ones you can use in PHP date function (see format parameters in <http://php.net/manual/es/function.date.php#refsect1-function.date-parameters>)

IoT Platforms > MQTT

Configuration

IP Address:

Port number:

User:

Password:


Topic template:


#MESHLIUM##ID_WASP##SENSOR#

Message template:

```
"id": "#ID#"
"id_wasp": "#ID_WASP#",
"id_secret": "#ID_SECRET#",
"sensor": "#SENSOR#",
"value": "#VALUE#",
"datetime": "#TS("c")#"
"datetime2": "#TS("r")#"
"datetime3": "#TS("Y:m:d","1231234"-H:i:s)"#
```

Save




MQTT Sender

Start

Figure : Server/Broker Configuration

Examples about MQTT Servers/Brokers:

- <http://mqtt.org/wiki/doku.php/brokers>
- <http://mosquitto.org/>
- <http://mqtt.io/>

Note: in this example, the broker was running on a computer inside our local network for test purposes only. For professional use, it is recommended to work with a 24/7 server with static IP address.

17.1.10.2. Controlling status

Once configured the server/broker, the user can launch the Meshlium MQTT program (Start button). The program will search for the received frames on the local database, and will send them to the broker via MQTT protocol. The status indicator displays the current state, saying "Running" or "Stopped".



Figure : MQTT sender is running

You can stop the MQTT sender anytime clicking on the "Stop" button.



Figure : MQTT sender is stopped

17.1.10.3. Platforms using MQTT

MQTT has been widely implemented across a variety of industries. As of March 2013, MQTT is in the process of undergoing standardization at OASIS protocol stack. The protocol specification has been openly published with a royalty-free license for many years, and companies such as Eurotech (formerly known as Arcom) have implemented the protocol in their products.

here are a number of notable projects that have made use of MQTT and related technologies. Companies like Cisco, Eclipse Foundation, Eurotech, IBM, Kaazing, M2Mi, Red Hat, Software AG, TIBCO and Carriots, among other companies, are working with this protocol.

More information about examples and uses: http://mqtt.org/wiki/doku.php/example_uses

17.1.11. Simfony

Simfony's IoT Platform is focused on providing the core set of tools that enables the rapid roll-out of any IoT project or service. Companies can use the service to easily and rapidly design, prototype and deploy IoT projects that match their exact needs and requirements, rather than looking for an off the shelf product that fits best. The platform provides the following services: global mobile data connectivity, SIM management and control, device authentication and authorization, a visual service designer, data storage, reporting and visualization, IoT VPN and an extensive API exposing all of these capabilities for enterprise integration. All these services are available on a "pick and choose" basis enabling maximum flexibility and optimizing costs.

More information can be found at www.simfonymobile.com.

17.1.11.1. Configuration

The Simfony Cloud Connector is capable of self-configuration using data already provisioned from the Simfony Cloud Platform. This function requires that the user is authenticated and authorized into the Cloud Service with a specific set of credentials provided through the Self-Care portal. These credentials are not stored on the Meshlium or by the Connector and will have to be entered manually each time an Auto-Configuration action is requested through the web GUI. This functionality runs on the user's browser that is connected to the Meshlium device and requires Internet access, i.e. from the browser to the Cloud Platform API.

Once the user is authenticated, a list of all the Cloud provisioned devices is available for selection. If one of the devices is selected in the drop-down list, the web GUI will automatically fill in or overwrite the following parameter: Client ID, Device ID, Device Name, and Device Password.

IoT Platforms > Simfony

Configuration Advanced

Simfony Cloud Auto-configuration (optional)

Username

Password


 Login

Figure : Simfony cloud connector configuration panel

If the Cloud Connector is already configured with a valid Device ID the Auto-Configuration feature will automatically retrieve the Cloud provisioned data corresponding to that Device ID and fill in the parameters mentioned above. Any previous configuration is overwritten. This functionality can be used to resync the data provisioned in the Cloud with the configuration data of the Connector.

IoT Platforms > **Simfony**

Configuration

Advanced

Simfony Cloud Auto-configuration (optional)

Meshlium Connector (tR8tWgDIhA)

Configuration

Client ID ⁱ

067171b57687f5e547c7cccc8d9ab2

Device ID ⁱ

tR8tWgDIhA

Device name ⁱ

Meshlium Connector

Device username ⁱ

BQ6ef

Device password ⁱ

.....

Connection type ⁱ

MQTT

Path ⁱ

/libelium/2

Save

● Cloud Connector Status

Start




Figure : Symfony cloud connector configuration synchronization panel

All the data retrieved automatically from the Cloud can also be entered manually.

The user must use the Save button to save any newly configured data or apply any changes to it.

17.1.11.2. Advanced configuration

The advanced configuration window of the Connector allows the setting of the following parameters:

- **Connection retries:** Controls the number of connection testing retries before suspending operations and going to the sleeping phase (see the Functional description chapter, Test connectivity phase in "[Simfony Meshlium Connector- User Guide](#)"). **Default: 3.**
- **MQTT QoS:** Controls the QoS of the MQTT PUBLISH messages. **Default: 1.**
- **Permanent MQTT connection:** In case of MQTT connections, controls if the Connector will close the MQTT connection or not during the sleep phase. **Default: false.**
- **Refresh Interval:** The number of seconds the Connector will suspend its operations (sleep time) before starting a new extract and transmit cycle (see the Functional description chapter in "[Simfony Meshlium Connector- User Guide](#)"). **Default: 300.**
- **Maximum transmit interval:** The number of Refresh Intervals after which the Connector will transmit the data independently of the number of new database records found and the "Minimum number of DB records" parameter value. **Default: 5.**
- **Minimum number of DB records:** The minimum number of new database records that will trigger a sending procedure of the Connector. If the found new number of records is lower (strictly) than the value of this parameter, the transmit phase will be suspended until the number of records reaches the threshold or the condition expires (see Maximum transmit interval in "[Simfony Meshlium Connector- User Guide](#)"). **Default: 1.**
- **Aggregate sensor data:** Controls the way the Connector aggregates the sensor data found in the DB. If "true", the connector will aggregate sensor data from the same Waspmote frames into a single message. If "false" the Connector will transmit the data individually as extracted from the database. **Default: true.**

IoT Platforms > Simfony

Configuration Advanced

Cloud Connector Advanced Options

Connection retries: 5

MQTT QoS: 2

MQTT permanent connection: True

Refresh interval: 10

Max transmit interval: 6 x 10 = 60

Min number of records: 1

Agregate sensor data: False

Extended data: True

Sensor include list:

Sensor exclude list:

Save




Figure : Simfony cloud connector advanced configuration panel

- **Extended sensor data:** Controls the number of parameters the Connector will transmit to the cloud. If “true”, the whole data extracted from the database will be sent. If “false”, only a subset of the data stored in the database is sent. **Default: false.**
- **Include sensor list:** The sensor ID list ([Meshlium Technical Guide](#)) that the Connector will look for when extracting data from the Meshlium database. The “Include” and “Exclude” lists are exclusive with the Exclude list having higher precedence. **Default: empty.**
- **Exclude sensor list:** The sensor ID list ([Meshlium Technical Guide](#)) that the Connector will exclude when extracting data from the Meshlium database. The “Include” and “Exclude” list are exclusive with the Exclude list having higher precedence. **Default: empty.**

17.1.11.3. Running the Connector

After the entire configuration is complete, the user can start the connector using the “Start” button of the GUI. The Connector will be started and run seamlessly in background.



Figure : Simfony start button

The Status box will show the Connector’s state whenever the page is viewed by the user.

To stop the Connector, the user can press the “Kill” button that will stop the connector from running.

Warning: The “Kill” operation will terminate the Connector process and all its procedures abruptly independently of the stage they are in, i.e. extracting, transmitting, etc.



Figure : Simfony stop/kill button

To stop gracefully the connector the “Stop” button can be used. This will not interrupt any ongoing operations but rather wait for the connector to finish any ongoing activities. The Connector will look for this graceful stop signal each time it is starting or finishing the sleep cycle.

17.1.11.4. Integration with Simfony's IoT platform

The Simfony Cloud Connector is intended to work with any type of connectivity provided by the Meshlium device it is deployed on. The Connector has two standard protocols available for communicating with the Simfony Cloud: MQTT and HTTP. Both of them are available in the encrypted version also, i.e. MQTT+SSL and HTTPS. The customer it is able to choose the most appropriate protocol for his application.

Each Cloud Connector/Meshlium device must be individually authenticated and authorized before it can send data to the Simfony Cloud Service. The IoT platform will perform protocol specific authentication and authorization procedures and will allow the connectors to send data only if these are successful. The Connector configuration data must contain these credentials before the Connector can run properly.

Before data can be sent from the Connector to the Cloud Platform, a Cloud IoT Application must be deployed in order to listen for data. Customers can easily create, test and deploy their own applications via the Application Designer GUI. Each application can have a specific entry point for the data coming from the sensor and connectors. This entry point is defined by the used Protocol (MQTT or HTTP) and a custom target (MQTT-topic; HTTP-path). This entry point must be also configured in the Connector via the "Connection Type" and "Connection Path" parameters. Once the application is deployed, the connectors can start sending data into it and the custom business logic will be triggered.

Find out more about running the connector in the "[Simfony Meshlium Connector- User Guide](#)".

17.1.12. SmartCityPlatform

SmartCityPlatform is an innovative platform for digital transformation of cities. Our solution connects key city stakeholders such as: city leaders, citizens and businesses. The platform serves as a smart governance tool and big data market place that create value for future city development. SmartCityPlatform enables management of urban, socio-economic and technological development of the city. It measures city data, reports progress and provides business intelligence in real time.

Website URL: www.smartiscity.eu

By using our cloud connector, you connect to the Sense module, which is a part of SmartCityPlatform. It allows an overview of the city's pulse by gathering, measuring & monitoring happenings in the city.

17.1.12.1. Configuration

To use the Cloud Connector, you need to register your Meshlium in our authentication server. The server provides you with credentials, which you enter in the form below. When you finish configuring, click the "Save" button to save the configuration.

IoT Solutions > SmartCityPlatform

Configuration

Hostname:	<input type="text" value="broker.smartis.si"/>
Port:	<input type="text" value="1883"/>
Account server:	<input type="text" value="accunt.smartis.si"/>
Client ID:	<input type="text" value="0e8e048c-90ed-4d0d-826a-53ffdd51fa84"/>
Client secret:	<input type="password" value="....."/>
Device ID:	<input type="text" value="57022239a35518524bf8fdb9"/>
Device secret:	<input type="password" value="....."/>
Device group:	<input type="text" value="libelium"/>
Interval:	<input type="text" value="60"/>

● Cloud Synchronization Status

Figure : SmartCityPlatform cloud connector configuration panel

- **Hostname:** the IP or hostname of the sensor broker.
- **Port:** the port where the sensor broker is listening for connections.
- **Account server:** the IP or hostname of the account server.
- **Client ID:** client identification provided by the authentication server.
- **Client secret:** client secret provided by the authentication server.
- **Device ID:** device identification provided by the authentication server.
- **Device secret:** device secret provided by the authentication server.
- **Device group:** the name you set for your device group.
- **Interval:** time duration in seconds between synchronizing data batches.

17.1.12.2. Controlling Synchronization

With the configuration saved, you can start using the cloud connector. To start the synchronization, press the green “Start” button on the right.



Figure : SmartCityPlatform start button

You get a “loading” status inside the synchronization control section, indicating the synchronization is starting.



Figure : SmartCityPlatform loading button

When the cloud connector starts, a green dot on the left appears, indicating the synchronization is running. To stop the synchronization, simply click on the red “Stop” button on the right.



Figure : SmartCityPlatform stop button

17.1.13. Telefónica IoT Platform

Telefónica provides an M2M cloud to collect and analyze data. This platform is based on assets and models and you can optimize your business processes implementing rules and notifications, and subscribing to data from different hosts.

17.1.13.1. Configuration

A new option is shown in M2M Platform menu, in the Cloud Connector main option. If you expand it, you can see this form with 3 fields in it:

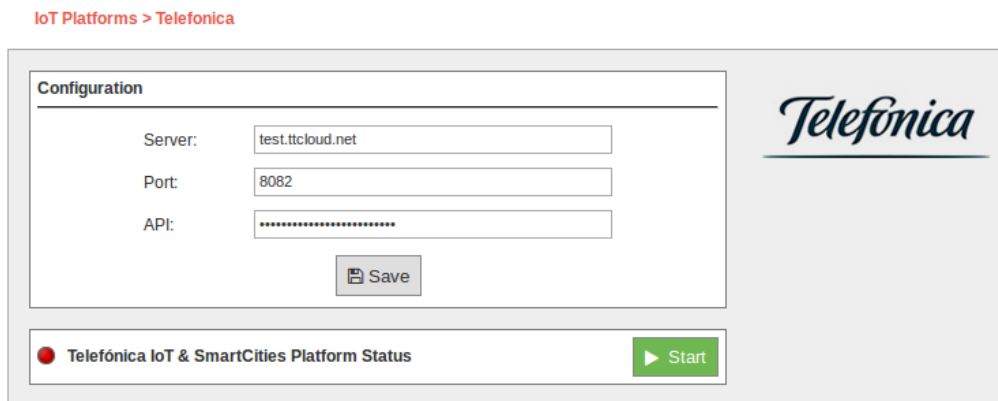


Figure : Telefónica IoT setup example on Manager System

- **URL:** Address of the API service of Telefónica IoT. This address should be provided without the “http://”, usually int.dca.tid.es.
- **Port:** The port in which the API listens to connections.
- **API:** The security key to send data to Telefónica IoT.

All this data are provided by Telefónica service administrators.

17.1.13.2. Controlling synchronization

The synchronization will be done in packs of 100 data at a time, so the system is not overloaded. You can start and stop the synchronization of the data to the Telefonica service. In the interface, you can see an indicator of whether the Telefonica service is running or not. If you click on “Start”, the synchronization will begin.

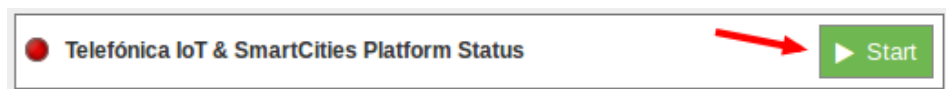


Figure : Telefónica IoT Start button

You can stop at any moment clicking on “Stop” button.

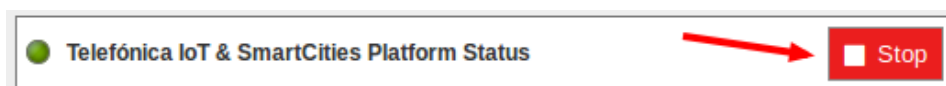


Figure : Telefónica IoT Stop button

17.1.14. ThingWorx

ThingWorx is the first software platform designed to build and run the applications of the connected world. ThingWorx reduces the time, cost, and risk required to build innovative Machine-to-Machine (M2M) and Internet of Things applications by providing a complete application design, runtime, and intelligence environment. The ThingWorx platform includes flexible device connectivity options, rapid application development tools, scalable storage, and supports various deployment models.

More information: <http://www.thingworx.com>

ThingWorx includes the following features:

- ThingWorx Composer™ - an end-to-end application-modeling environment designed to help you easily build the unique applications of today's connected world. Composer makes it easy to model the Things, Business Logic, Visualization, Data Storage, Collaboration, and Security required for a connected application.
- Codeless Mashup Builder – a “drag and drop” Mashup Builder empowers developers and business users to rapidly create rich, interactive applications, real-time dashboards, collaborative workspaces, and mobile interfaces without the need for coding.
- Execution and Storage Engine - ThingWorx's event-driven execution engine and 3-Dimensional storage allows companies to make business sense of the massive amounts of data from their people, systems, and connected “Things” – making the data useful and actionable. It also features a data collection engine that provides unified, semantic storage for time-series, structured, and social data at rates 10X faster than traditional RDBs.
- Search-based Intelligence - ThingWorx SQUEAL™ (Search, Query, and Analysis) brings Search to the world of connected devices and distributed data. With SQUEAL's interactive search capabilities, users can correlate data that delivers answers to key business questions.

Note: If you need more information about these components, go to <http://www.thingworx.com/platform/>

17.1.14.1. Configuration

Inside the “ThingWorx” plugin you can setup which Waspmites in the system will be published in ThingWorx server.

IoT Platforms > Thingworx

Configuration

Server address:

Server Port:

Meshlium bind name:

Thingworx App Key:

Waspmites in Thingworx

waspote_test Sensors TCB <input type="button" value="Delete"/>	Wasp3-WasteW Sensors TCA TCB ORP <input type="button" value="Delete"/>	Wasp3-Park Sensors PS <input type="button" value="Delete"/>
Wasp2-Park Sensors PS	Wasp2-InfrastW Sensors LP US	Wasp2-Env Sensors CO CO2 NO2 TCA HUMA

☒ Edge MicroServer (EMS) Status

☒ LUA Script resource Status

Figure : ThingWorx configuration

The parameters to setup are:

- **Server address:** The address of your ThingWorx server.
- **Server Port:** The port where your ThingWorx server is accessible.
- **Meshlium bind name:** The name of the Meshlium “thing” in ThingWorx. Meshlium thing is detected in ThingWorx but won’t send any data.
- **ThingWorx App Key:** Security key to send data to your ThingWorx server.

Click on the “Save” button to write this setup to the ThingWorx service.

The steps to setup Waspmites to send to ThingWorx are:

- Click on the button “Load local WM”. This will read Waspmites that have data in the sensor database.

Waspmites in Thingworx

Figure : Getting Waspmites from the system

- A panel with the devices is displayed, with a list of the sensors received from each Wasmote.

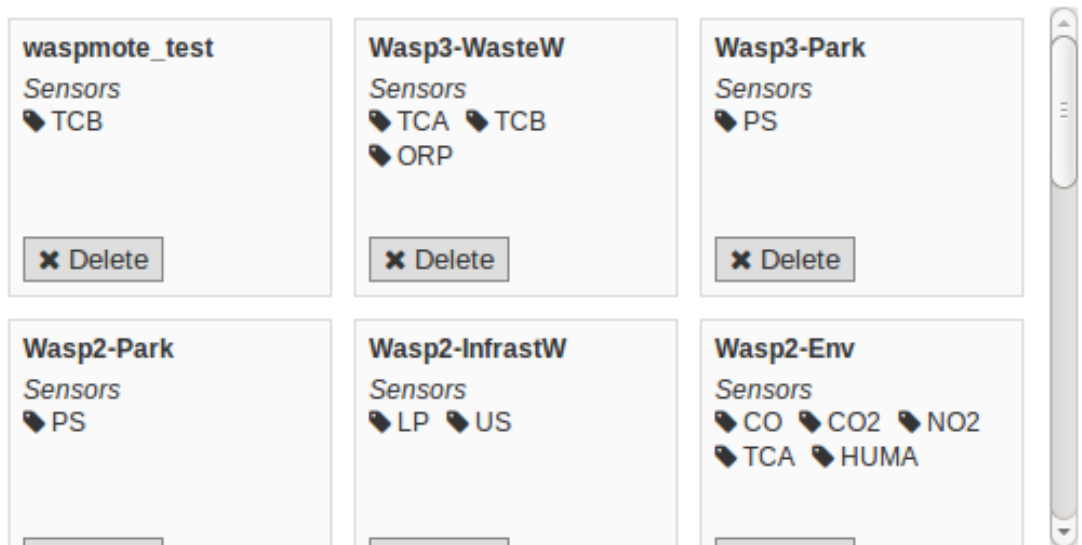


Figure : Wasmotes to be send to ThingWorx

- It is possible to delete a Wasmote from the list clicking on its “Delete” button. This device won’t be published to the ThingWorx platform.

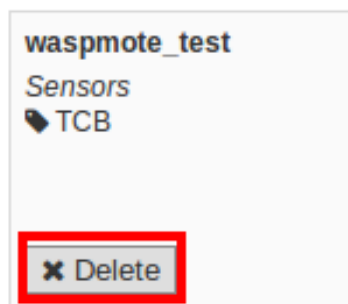


Figure : Delete a Wasmote from the list

- Once the list is correct, clicking on the button “Write ThingWorx setup” will push this setup to the ThingWorx EMS service.

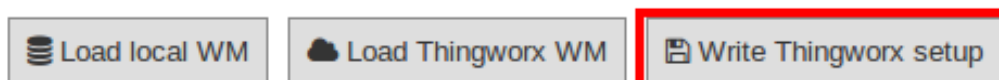


Figure : Write the setup to ThingWorx service

After restarting the EMS and LUA Script services, the setup will be applied and you will see your devices in your ThingWorx server.

If you click again on the “Get Wasmotes from DB” button, the plugin will read again the DB and display all the Wasmotes. If you don’t write this changes to ThingWorx setup, this won’t propagate to the EMS service.

You can recover the current ThingWorx EMS service setup by clicking on the “Load WM from ThingWorx”.

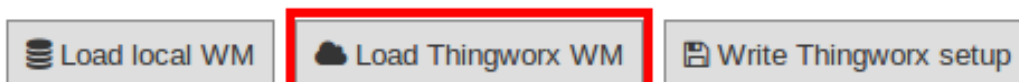


Figure : Write the setup to ThingWorx service

17.1.14.2. Using The Edge MicroServer (EMS)

ThingWorx has the ability to connect very quickly and easily to the physical world and structured data systems. ThingWorx supports a variety of communication protocols and system interfaces. Many are implemented directly on the ThingWorx Platform. However, for edge devices or data stores that need to connect to the platform using the Internet or through firewalls on an Intranet, ThingWorx provides an Edge MicroServer (EMS) solution that can be deployed where the data is, and allows secure, efficient communication back to the ThingWorx Platform. This section will concentrate on the EMS and the corresponding Edge Thing software components.

In order to send data to the ThingWorx environment, another component is needed: the LUA Script Resource service. This service needs to be running at the same time with Edge Micro Server to allow data acquisition.

To launch the Edge MicroServer (EMS) press Start button, and to stop it, press Stop button.



Figure : ThingWorx Edge MicroServer running



Figure : ThingWorx Edge MicroServer stopped

To launch the LUA Script resource press the Start button, and to stop it, press the Stop button.



Figure : ThingWorx Edge MicroServer running



Figure : ThingWorx Edge MicroServer stopped

17.2. IoT Solutions

17.2.1. B-Scada

B-Scada® VoT platform allows you to create rich, sophisticated IoT and M2M applications that consolidate and organize data from anywhere, and visualize it in real-time on any device. Connect to thousands of potential data sources. Visualize your data using modern, high-performance customized graphics. Leverage powerful analytic tools and automation. Connect your devices, processes and people in a continuous real-time information system.

More information about VoT Platform: <http://www.votplatform.com/>

17.2.1.1. Configuration

A new option is shown in the M2M Platforms menu: the **B-Scada Cloud Connector**. If you expand it, you can see this form with 6 fields in it:

IoT Solutions > B-Scada

Configuration

Url : 80.32.200.207

Port: 1883

Client ID : Client123

Secret Key :

Interval(s) : 60

Enable Log : ☒

Save

Cloud Synchronization Status Start

Figure : Configuring B-Scada in Meshlium

- **URL:** IP address of the VoT platform service by B-Scada. This address should be provided by B-Scada.
- **Port:** The port in which the VoT Server is listening to connections.
- **Client ID:** Customer's identifier or company name.
- **Secret Key:** The security key to send encrypted data to VoT.
- **Interval(s):** Time duration in seconds between operations of updating data.
- **Enable Log:** This option enables the creation of log files to save all communications processed to the VoT server.

17.2.1.2. Controlling synchronization

The synchronization will be done for all data that has not been synchronized in the Sensor Parser table each time. You can start and stop the data synchronization to the VoT service. In the interface you can see an indicator of whether the status service is running or not. If you click on "Start", the synchronization will begin.

Cloud Synchronization Status Stop

Figure : B-Scada synchronization service is running

You can stop the synchronization at any moment clicking on the "Stop" button.


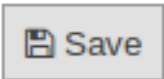
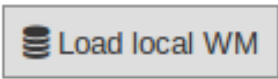
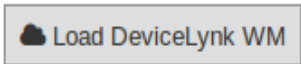
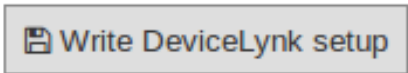






Figure : B-Scada synchronization service is stopped

17.2.2. DeviceLynk

DeviceLynk is a high-level cloud service based on the ThingWorx cloud.

Interface walk-through:

Interface Feature	Description
	<u>DeviceLynk logo</u> Click to jump to the DeviceLynk website.
Server Address: <input type="text" value="666"/>	<u>Server Address</u> The DeviceLynk Server address you wish to connect to.
Meshlium Bind Name: <input type="text" value="777"/>	<u>Meshlium Bind Name</u> The name that the DeviceLynk Server uses to identify the Meshlium unit.
Enable Logging: <input type="checkbox"/>	<u>Logging Check-box</u> Check-box to enable/disable internal logging.
	<u>Save Button</u> Click to save the Server Address, Meshlium Bind Name, and Logging configuration.
	<u>Load Local Waspnotes button</u> Click this button to load the list of Waspnotes that the Meshlium is connected to.
	<u>Load WM Config. from DeviceLynk button</u> Click to load the list of Waspnotes which are connected to the DeviceLynk Server.
	<u>Write DeviceLynk Setup button</u> Click to write the current list of Waspnotes to the DeviceLynk Server (Waspnotes details are sent to the Server).

Interface Feature	Description
	<p><u>Waspnote</u></p> <p>Click the “Delete” button to remove that Waspnote unit from the current Waspnote clicking the “Load Local Waspnotes” button, or by clicking the “Load WM Config. from DeviceLynk” button.</p>
	<p><u>DeviceLynk Agent Status</u></p> <p>The status of the DeviceLynk Agent will be indicated, displaying “Running” or “Stopped”.</p>
	<p><u>LUA Script Resource Status</u></p> <p>The status of the LUA Script Resource will be indicated, displaying “Running” or “Stopped”.</p>
	<p><u>Start/Stop Service Toggle button</u></p> <p>Click to start/stop the service that this button is next to.</p>

Steps to start the DeviceLynk plugin:

1. Type the DeviceLynk Server address that you wish to connect to in the Server Address field.
2. Type the Meshlium Bind Name that the DeviceLynk Server will use to identify the Meshlium device.
3. Click the “Save” button.
4. Click the “Load Local Waspnotes” button. All Waspnotes connected to this Meshlium unit will show up.
5. Delete the Waspnotes that you do not want to be connected to the DeviceLynk Server by clicking on their respective “Delete” button.
6. Click the “Write DeviceLynk Setup” button to make the DeviceLynk Server listen to those Waspnotes.
7. To show the Waspnotes units that the DeviceLynk Server is currently listening to, click the “Load WM Config. from DeviceLynk” button.
8. Click the “Start” button under “DeviceLynk Agent Status”. “Running” will be displayed.
9. Click the “Start” button under “LUA Script Resource Status”. “Running” will be displayed.

17.2.3. Devicify

Devicify is a fully hosted solution that spans two environments, one optimized for business functions and one optimized for device functions. As such it is delivered as a Salesforce.com managed package as well as an extension package utilizing a managed ThingWorx server. These two sides of the solution are seamlessly unified as a cohesive application leveraging a common Account and Device model.

Devicify's integration with Libelium is meant to help Libelium users easily sync their sensor data into their Devicify CPM platform, allowing for seamless integration between their Libelium sensor network, ThingWorx, and Salesforce.


17.2.3.1. Configuration


Navigating to the Devicify plugin is done via the Cloud Connector tab and then M2M Platform option. Under the Devicify dropdown, you will find the configuration setup and start/stop options for the Devicify Cloud Connector.


IoT Solutions > Devicify


Configuration

URL:	<input type="text" value="yourserver.dvsvr.comxxx"/>
Namespace:	<input type="text" value="NS"/>
Meshlium Serial Number:	<input type="text" value="15068091009957"/>
Device Serial Number:	<input type="text" value="15068091009957"/>
Device Access Key:	<input type="text" value="XXXXXXXX-XXXX-XXXX-XXXX-X"/>
Reporting Frequency (seconds):	<input type="text" value="60"/>

 Copy Meshlium Serial Number

 Save



 Cloud Synchronization Status




Figure : Configuring Devicify in Meshlium

The configuration settings for the Devicify Cloud Connector plugin come with defaults, but must be changed by you to establish the connection between your Meshlium and Devicify CPM.

- **URL:** This is the URL of your ThingWorx server. For example, if you access your ThingWorx composer via the URL <http://NS.dvsvr.com/ThingWorx/Composer> you will enter NS.dvsvr.com in this field. If you are unsure of this or do not work in ThingWorx platform, contact Devicify
- **Namespace:** This is your Devicify Namespace. You will find this used across your Devicify CPM platform, often as a prefix to your device or account names. For example, if your Meshlium entity in ThingWorx is named NS.Device.SerialNumber you will enter NS in this field.
- **Meshlium Serial Number:** This is the serial number assigned to the Meshlium by Libelium at the factory.
- **Device Serial Number:** This is the serial number you assigned to your Meshlium device when it was created in Devicify CPM. If you used the Meshlium's Serial Number, simply use the provided Copy button in the UI to copy that serial number into this field. If you created your own, be sure to use that serial number here. This field is critical because it is used by the Devicify Cloud Connector to identify the ThingWorx Meshlium entity to sync with. Unless you have a good reason otherwise, we recommend re-using the serial number already provided by Libelium.
- **Device Access Key:** This is the unique access key assigned to your Meshlium entity in ThingWorx. It is found in the Properties section, with the inherited DV.DeviceShape properties.
- **Reporting Frequency (seconds):** This is where you are able to set how often the Meshlium syncs its database entries with the Meshlium entity in ThingWorx. See the Meshlium Properties section in the ThingWorx section of this document for restrictions. It is also advisable to read the Sync Process section of the Developer Guide to better formulate an idea of a desirable value here.

17.2.3.2. Controlling synchronization

Clicking the start button will begin the sync process. Sensor data entries will be synced at the interval designated in the configuration section. If there are more than 100 non-synced entries at the start of any sync process, the Devicify plugin will continuously sync in “batches” of 100 entries so as to not overload Meshlium. However, all these “batches” will sync with a common sync date.



Figure : Devicify synchronization service is running

You can stop at any moment clicking on the “Stop” button.



Figure : Devicify synchronization service is stopped

17.2.4. ElementBlue – RightSensor

RightSensor is a solution company designed to provide sensors, services and support for Industrial Internet of Things projects.

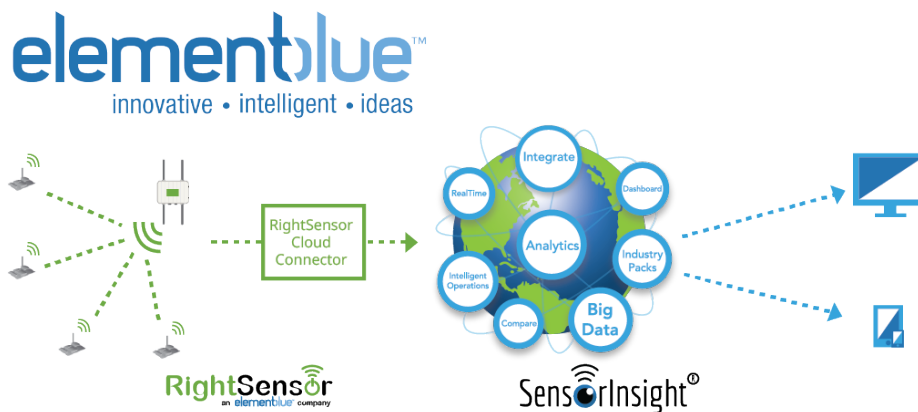


Figure : ElementBlue RightSensor Cloud

The RightSensor cloud connects interface works directly with Element Blue’s SensorInsight® Industrial Internet of Things Platform.

SensorInsight® is used by companies for the integration, display and detailed analysis of data from multiple sources providing an environment where users can view and compare real-time and historic data patterns, receive alerts and notifications, and trigger workflows and actions based on the data for use in industrial environments.

To use the service you must have an account with Element Blue’s SensorInsight Industrial Internet of Things Platform. For more information visit: www.sensorinsight.io and www.rightsensor.com

17.2.4.1. Configuration

By expanding the RightSensor menu item on the list you can see the form in which to set your connection parameters. The form accepts the following 4 parameters:

- **Client ID:** This is a unique ID provided to you from the SensorInsight service.
- **Gateway ID:** This is an ID you provide to uniquely identify this Meshlium device.
- **User Name:** This is the Username required to send your data to the SensorInsight cloud.
- **Password:** This is the Password required to send your data to the SensorInsight cloud.

IoT Solutions > Elementblue


Configuration


Client ID:

Gateway ID:

User Name:

Password:

 Save



innovative • intelligent • ideas



 **SensorInsight Synchronization Status**


Figure : Configuring ElementBlue RightSensor in Meshlium

The form accepts the following 4 parameters:

- **Client ID:** This is a unique ID provided to you from the SensorInsight service.
- **Gateway ID:** This is an ID you provide to uniquely identify this Meshlium device.
- **User Name:** This is the Username required to send your data to the SensorInsight cloud.
- **Password:** This is the Password required to send your data to the SensorInsight cloud.

These parameters can be obtained from your SensorInsight account page. Learn more at www.sensorinsight.io.

17.2.4.2. Controlling synchronization

To launch the cloud connector service and start sending your data to RightSensor press the “Start” button.



 **SensorInsight Synchronization Status**


Figure : ElementBlue RightSensor synchronization service is running

You can stop at any moment clicking on the “Stop” button.



 **SensorInsight Synchronization Status**


Figure : ElementBlue RightSensor synchronization service is stopped

17.2.4.3. Problems

The RightSensor Cloud Connector service has built-in logging and debugging capability. Please contact RightSensor at www.rightsensor.com for troubleshooting information.

17.2.5. IoTSENS

IoTSENS (<http://www.iotsens.com/>) is a horizontal platform for the development of smart cities which provides functionalities for gathering, integrating, storing and analyzing data from the city from a global point of view, so managers and citizens know what is happening and can immediately act.


IoTSENS seamlessly integrates with Meshlium devices by means of MQTT queues so the connector will send all the sensors data to your IoTSENS platform in order to be processed.

17.2.5.1. Configuration

IoT Solutions > IOTSENS

Configuration

MQTT Server IP:	<input type="text" value="172.20.16.201"/>
MQTT Server Port:	<input type="text" value="1883"/>
MQTT Server User:	<input type="text" value="libeliumhhuuu"/>
MQTT Server Password:	<input type="password" value="*****"/>
Gateway Identifier:	<input type="text" value="60xxxx"/>
Sleeping time (s):	<input type="text" value="200"/>
Max. measures to sync:	<input type="text"/>



IOTSENS
GRUPO GIMENO IoT Division

● IoTSENS Synchronization Status

Figure : Configuring IoTSENS in Meshlium

The IoTSENS provider will supply you with the MQTT connection configuration attending your particular deployment:

- **MQTT Server IP:** IP address where the MQTT Server is deployed.
- **MQTT Server Port:** Port number where the MQTT Server is listening for connections.
- **MQTT Server User:** User name for connecting to the MQTT Server. This field can be empty if no user is required.
- **MQTT Server Password:** Password for connecting to the MQTT Server. This field can be empty if no user is required.

Additionally, the IoTSENS plugin supports the configuration of some parameters regarding how the synchronization process works:

- **Sleeping time:** The synchronization process sleeps some time between executions. This parameter configures how many seconds it will sleep before starting the synchronization process again once it has finished. The sleeping time must be long enough to give time to other device processes to do their work.
- **Max. measures to sync:** This parameter configures how many sensor measures are synchronized at most in every synchronization process. The number of measures to synchronize must be limited in order to avoid the synchronization process to overload the system for a long time.

17.2.5.2. Controlling synchronization

You can start and stop the synchronization of the data to the IoTSENS service. In the interface, you can see an indicator of whether the IoTSENS service is running or not. If you click on "Start", the synchronization will begin.



Figure : IoTSENS synchronization service is running

You can stop at any moment clicking on “Stop” button.



Figure : IoTens synchronization service is stopped

17.2.6. Sentilo

Sentilo is an open source sensor and actuator platform designed to fit in the Smart City architecture of any city who looks for openness and easy interoperability. It's built, used, and supported by an active and diverse community of cities and companies that believe that using open standards and free software is the first smart decision a Smart City should take.

17.2.6.1. Configuration

IoT Solutions > Sentilo


Configuration


Sentilo URL:

Connection Port:


Sentilo Provider:

Sentilo Authentication Key:

 Save



Sentilo Synchronization Status



Inside the “Sentilo” plugin, you have a form to introduce your credentials to access your Sentilo system. You have to enter here these parameters:

- **Sentilo URL:** Address of the API service of Sentilo. This address should be provided without the “http://”.
- **Connection Port:** The port in which the API listens to connections.
- **Sentilo Provider:** The provider is the identity of who is sending data to Sentilo.
- **Sentilo Key:** The security key to send data to Sentilo.

This data will be provided by the administrators of the Sentilo system you are using.

17.2.6.2. Controlling synchronization

The synchronization will be done in packs of 100 data at a time, so the system is not overloaded. You can start and stop the synchronization of the data to the Sentilo service. In the interface, you can see an indicator of whether the Sentilo service is running or not. If you click on “Start”, the synchronization will begin.



You can stop at any moment clicking on “Stop” button.



17.2.7. Sofia2

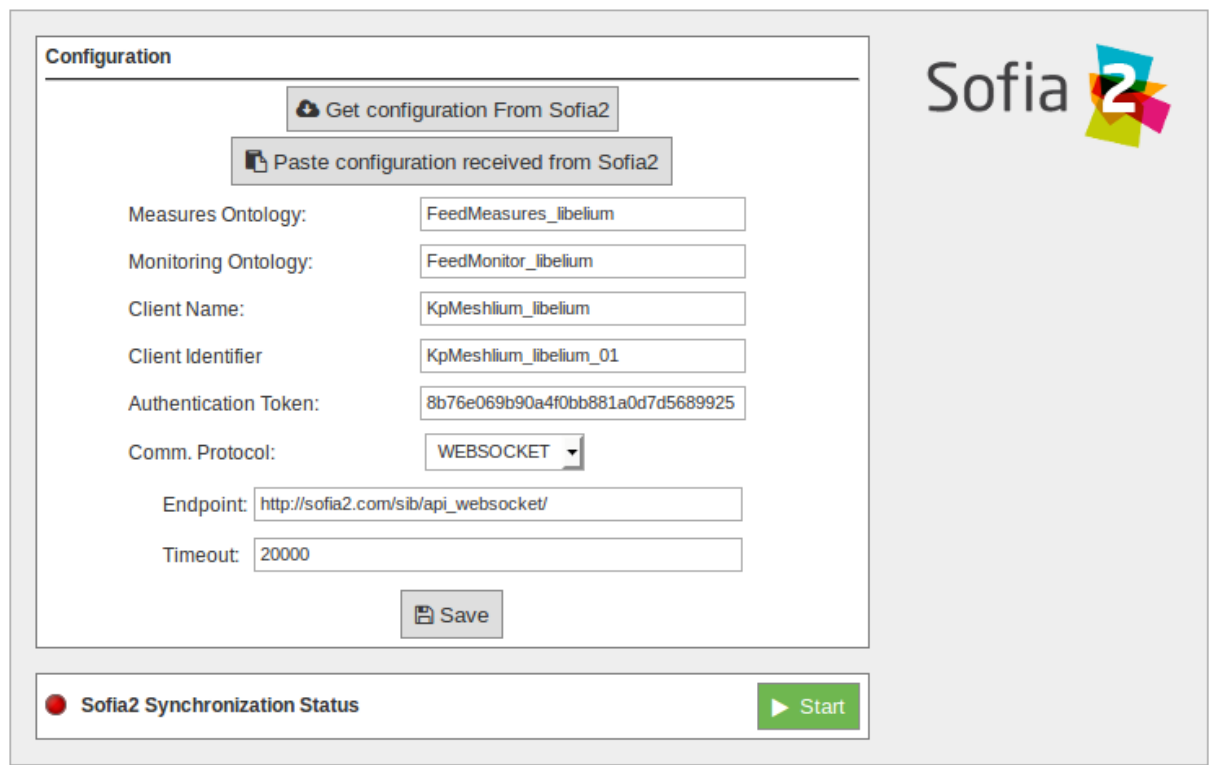
Sofia2 is a middleware developed by Indra that allows the interoperability of multiple systems and devices, offering a semantic platform to make real world information available to smart applications (Internet of Things).

It is multi-language and multi-protocol, enabling the interconnection of heterogeneous devices. It provides publishing and subscription mechanisms, facilitating the orchestration of sensors and actuators in order to monitor and act on the environment.

17.2.7.1. Configuration

The plugin to connect Meshlium to Sofia2 platform is in the Manager System section **Cloud Connector > IoT Solutions > Sofia2**.

IoT Solutions > Sofia2



Configuration

Get configuration From Sofia2

Paste configuration received from Sofia2

Measures Ontology: FeedMeasures_libelium

Monitoring Ontology: FeedMonitor_libelium

Client Name: KpMeshlium_libelium

Client Identifier: KpMeshlium_libelium_01

Authentication Token: 8b76e069b90a4f0bb881a0d7d5689925

Comm. Protocol: WEBSOCKET

Endpoint: http://sofia2.com/sib/api_websocket/

Timeout: 20000

Save

Sofia2 Synchronization Status Start

Figure : Sofia2 configuration

Registering the Meshlium device in Sofia2 is a previous step required to connect the Meshlium with Sofia2. At the end of the registration process you will obtain the configuration parameters needed to set up the Meshlium plugin properly.

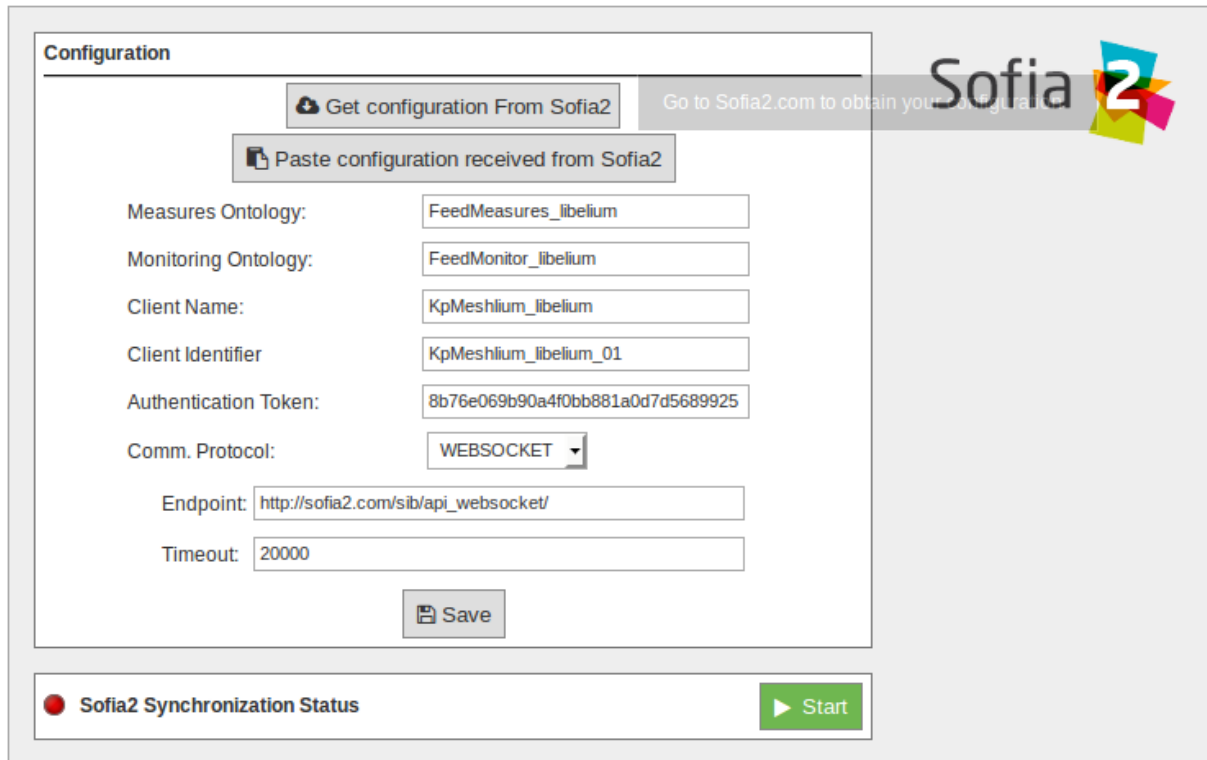
This configuration includes the following information:

- **Measures Ontology:** Collection (table or storage) where the measures from Wasp mote sensors sent by Meshlium (the gateway) will be stored into the platform.
- **Monitoring Ontology:** Collection (table or storage) where monitoring values (internal temperature, battery level...) of the Waspmotes connected to Meshlium, will be stored into the platform.
- **Client Name:** Name of the Meshlium unit to be identified by Sofia2 platform, checking if it has permission to write on the ontologies.
- **Client Identifier:** Identifier of the Meshlium unit to differentiate between several Meshliums using the same Client Name.
- **Authentication Token:** Token to authenticate the Meshlium device during the establishment of a session with Sofia2 platform.

17.2.7.2. Register Meshlium in Sofia2

To register Meshlium in Sofia2, click on the link Get Configuration From Sofia2.

IoT Solutions > Sofia2



Configuration

Get configuration From Sofia2

Paste configuration received from Sofia2

Measures Ontology: FeedMeasures_libelium

Monitoring Ontology: FeedMonitor_libelium

Client Name: KpMeshlium_libelium

Client Identifier: KpMeshlium_libelium_01

Authentication Token: 8b76e069b90a4f0bb881a0d7d5689925

Comm. Protocol: WEBSOCKET

Endpoint: http://sofia2.com/sib/api_websocket/

Timeout: 20000

Save

Sofia2 Synchronization Status

Start

Figure : Sofia2 configuration link

You will be redirected to the following page:

Web access portal

To start working with your meshlium device you need to be logged in the platform

[Login or Register for this site](#)

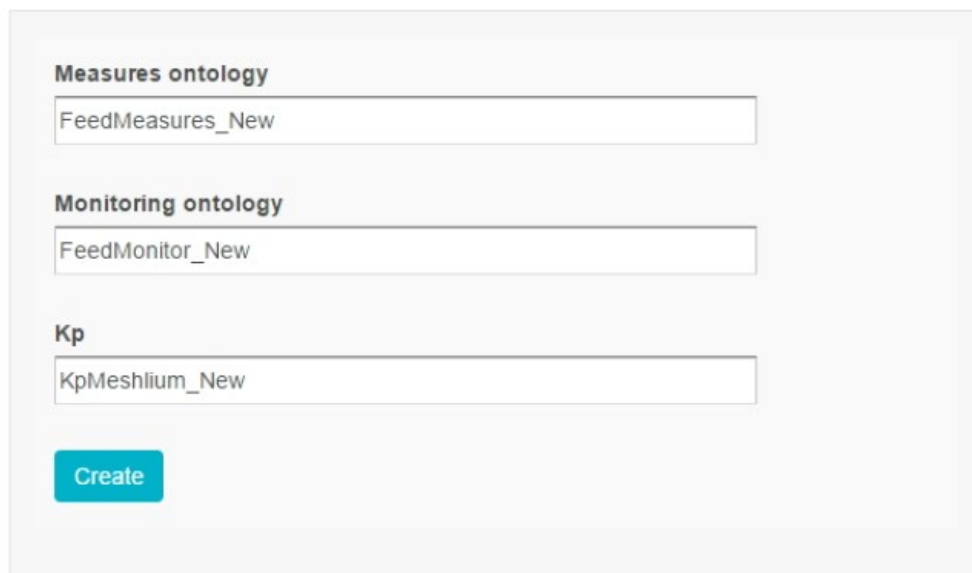
Figure : Sofia2 configuration portal

Where:

- Login using your Sofia2 account.
- Create a new account if you do not have a valid login user.

Logging in Sofia2 platform will redirect to the following page that suggests a name for your collections of measures and monitoring data, and for your gateway identifier (KP in Sofia2 terms):

Meshlium Cloud Conector



Measures ontology
FeedMeasures_New

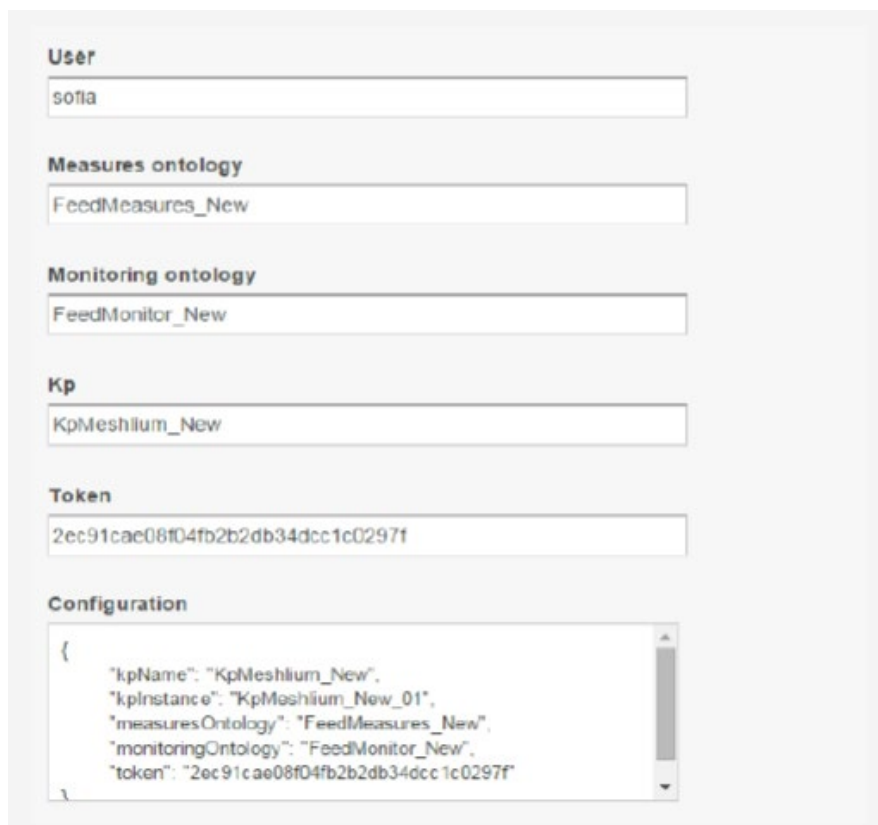
Monitoring ontology
FeedMonitor_New

Kp
KpMeshlium_New

Create

Figure : Sofia2 configuration information

Finally, after creating the configuration, you will obtain a JSON file containing all configuration values ready to be pasted on the Manager System, in order to setup the Sofia2 Cloud Connector:



User
sofia

Measures ontology
FeedMeasures_New

Monitoring ontology
FeedMonitor_New

Kp
KpMeshlium_New

Token
2ec91cae08f04fb2b2db34dcc1c0297f

Configuration

```
{
  "kpName": "KpMeshlium_New",
  "kpInstance": "KpMeshlium_New_01",
  "measuresOntology": "FeedMeasures_New",
  "monitoringOntology": "FeedMonitor_New",
  "token": "2ec91cae08f04fb2b2db34dcc1c0297f"
}
```

Figure : Sofia2 JSON configuration

17.2.7.3. Configure the Cloud Connector

The configuration from Sofia2 can be setup in Meshlium just by clicking Paste configuration received from Sofia2 and pasting the JSON generated in the previous step.

IoT Solutions > Sofia2

Configuration

Get configuration From Sofia2

Paste configuration received from Sofia2

Click here to paste the configuration generated in Sofia2 console

Measures Ontology: FeedMeasures_libelium

Monitoring Ontology: FeedMonitor_libelium

Client Name: KpMeshlium_libelium

Client Identifier: KpMeshlium_libelium_01

Authentication Token: 8b76e069b90a4f0bb881a0d7d5689925

Comm. Protocol: WEBSOCKET

Endpoint: http://sofia2.com/sib/api_websocket/

Timeout: 20000

Save

Sofia2 Synchronization Status

Start

Figure : Sofia2 pasting JSON configuration

Configuring the plugin this way, the fields for Measures Ontology, Monitoring Ontology, Client Name, Client Identifier and Authentication Token will be completed.

Alternatively, these fields can be filled in manually, with the information received from the configuration page of Sofia2 showed in the first step.

17.2.7.4. Select communication protocol

The Cloud Connector for Sofia2 platform provides three different communication protocols, REST, MQTT and Websocket. This way, the user can select the most appropriate.

Comm. Protocol:

MQTT

Connection protocol Sofia2

Server:

sofia2.com

Port:

1883

KeepAlive:

5

Connection Timeout(ms):

5000

Response Timeout(ms):

6000

Auth user:

Auth password:

Figure : Sofia2 communication protocols

Each protocol has different features and configuration parameters. By default, the plugin is configured to connect with the public instance of Sofia2 under www.sofia2.com, but it can be changed to connect with any other Sofia2 installations.

17.2.7.4.1. REST

REST is a stateless communication protocol over HTTP. Using this protocol, the Cloud Connector is a client of the Sofia2 platform, that periodically opens a connection with the platform to send an HTTP POST operation containing the sensor measurements.

The parameter of the REST protocol are:

- **Endpoint:** URL of the REST Gateway of Sofia2 platform. It is the REST server that will receive requests from clients.

Comm. Protocol:

17.2.7.4.2. MQTT

MQTT is a stateful communication protocol over TCP. Using this protocol, the Cloud Connector is a client of the Sofia2 platform, that initially opens a connection with the platform, maintains it alive during that time, and periodically sends an MQTT packet containing the sensor measurements. In case of disconnection, the connector periodically tries to reconnect.

The parameters of the MQTT protocol are:

- **Server:** IP or machine name of the MQTT gateway in the Sofia2 server.
- **Port:** Port of the MQTT gateway in the Sofia2 server.
- **KeepAlive:** Interval in seconds that the connector will use to check the status of the connection.
- **Connection Timeout:** Timeout to establish a connection.
- **Response Timeout:** Timeout to wait response from the Sofia2 server.
- **Auth user:** Optional. MQTT protocol authentication user.
- **Auth password:** Optional. MQTT protocol authentication password.

Comm. Protocol:

Connection protocol
Sofia2

Server:

Port:

KeepAlive:

Connection Timeout(ms):

Response Timeout(ms):

Auth user:

Auth password:

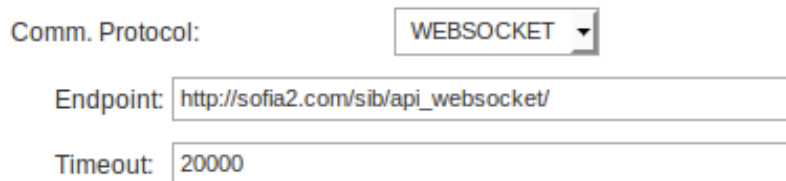
Figure : Sofia2 MQTT protocol

17.2.7.5. Websocket

It is a stateful communication protocol over HTTP. Using this protocol, the Cloud Connector is a client of Sofia2 platform, that initially open a connection with the platform, maintains it alive during a defined time, and periodically sends a HTTP packet containing the sensor measurements. In case of disconnection, the connector periodically tries to reconnect.

The parameters of the Websocket protocol are:

- **Endpoint:** URL of the Websocket gateway of Sofia2 platform. It is the server that will receive requests from clients.
- **Timeout:** Timeout for any operation with the server.



Comm. Protocol: WEBSOCKET

Endpoint:

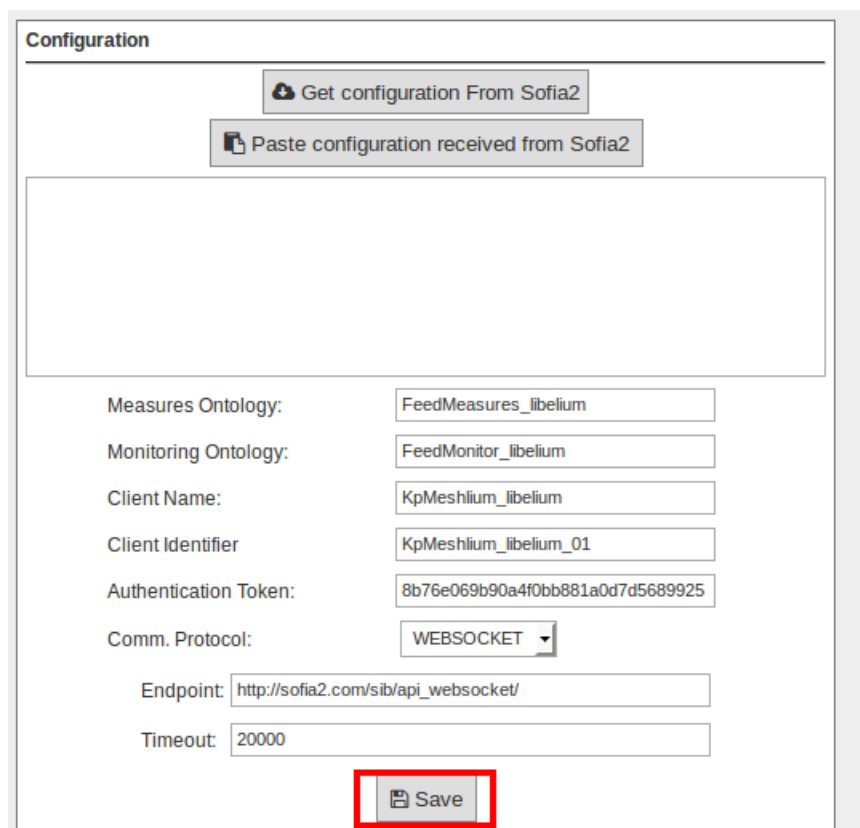
Timeout:

Figure : Sofia2 Websocket protocol

17.2.7.6. Save the configuration and start the connector

Once all configuration and connection parameters are setup, they can be stored and the connector can be started to send information to Sofia2.

To save the configuration, click on the “**Save**” button:



Configuration

Measures Ontology:

Monitoring Ontology:

Client Name:

Client Identifier:

Authentication Token:

Comm. Protocol: WEBSOCKET

Endpoint:

Timeout:

Figure : Sofia2 save configuration button

After saving the configuration, the Cloud Connector can be started by clicking on the “Start” button:



Figure : Sofia2 synchronization service running

You can stop the process at any moment by clicking on the “Stop” button.



Figure : Sofia2 synchronization service stopped

17.2.8. SolvView

SolvView is a real-time monitoring Cloud platform developed by Solvver that allows users to benefit from the power of the Cloud with little risk and cost. SolvView can aggregate, visualize and analyze real-time or historical data securely, easily, and in any device, and is capable of processing a huge number of parallel data streams or single-channels with very high datarates and executing complex algorithms natively in its in-house built mathematical engine. SolvView works without hassle in hostile network environments, assuring data integrity and can scale seamlessly and efficiently as project scope requires.

17.2.8.1. Configuration

SolvView's plugin is simple and intuitive to configure just by filling two parameters:


- **SolvView UUID:** UUID provided by Solvver
- **SolvView Token:** Token provided by Solvver


IoT Solutions > Solvver


Configuration

Solvview UUID:

Solvview Token:

 Save

 Solvview Synchronization Status






Figure : Configuring Solvver in Meshlium

17.2.8.2. Controlling synchronization

You can start and stop the synchronization of data to SolvView platform clicking on the buttons "Start" and "Stop" in the plugin. The status indicator says if the service is "Running" or "Stopped".

 Solvview Synchronization Status



Figure : Solvver synchronization service is running

You can stop at any moment clicking on the "Stop" button.

 Solvview Synchronization Status



Figure : Solvver synchronization service is stopped

17.2.9. ThingPlus

Thing+ allows customers to build their own IoT services with high speed, reliability, scalability, and cost competitiveness, connected by a SaaS or PaaS IoT platform. When Thing+ Embedded devices connect to the Thing+ Cloud (public or private), customers can visualize various data graphs and charts from sensors directly on the Thing+ Portal. Device registration is easy as the Thing+ Portal provides dashboard widgets, a trigger-condition-action-based rule engine for alert notifications or to control actuators, and results in the form of an event timeline.

17.2.9.1. Get API Key

- Get the "Gateway ID" to register:
 - Open the Meshlium Manager System
 - Click Cloud Connector
 - Open the ThingPlus plugin
 - You can see ThingPlus configuration and "Gateway ID"
 - Copy the "Gateway ID"
- Get the API Key:
 - Go to your ThingPlus service (if you have no registered service, register your service) <https://yourservice.thingplus.net>
 - Go to Gateway Management page (via upper right menu)
 - Click the + button in the upper right corner
 - Click "Request for Gateway certificates or API Key" button

The screenshot displays the 'Gateway Management' interface. At the top right, there is a button labeled 'Request for Gateway Certificates or API Key' with a key icon, which is highlighted with a red rectangular box. Below this, the 'Register Gateway' form is visible. It contains four fields: 'Gateway Model' with a dropdown menu showing '-- Select Gateway Model --'; 'Gateway ID' with a text input field containing 'Mac address' and a green 'Required' label below it; 'Gateway Name' with a text input field containing 'Gateway Name' and a green 'Required' label below it; and 'Site Name' with a dropdown menu showing '-- Select Site --' and a green 'Required' label below it. At the bottom right of the form, there is a yellow button labeled 'Register a Gateway, Devices and Sensors'.

Figure : Request API Key in the ThingPlus panel

- Fill form
 - Gateway ID
 - Select "API Key" (Authentication Type)
- Click "Get API Key"
- Copy the "API Key"

Gateway Management

Register Gateway

Request for Gateway Certificates or API Key

Gateway ID: 00:0d: [redacted]

Authentication Type: API Key

Get API Key

API Key: oitQ8: [redacted]

After installing the downloaded certification or API Key to your gateway, register your gateway.

Figure : Get the API Key in the ThingPlus panel

- Set "API Key" in the Meshlium Manager System
 - Go to ThingPlus configuration again
 - Paste the "API Key"

IoT Solutions > Thing+

Configuration

Gateway ID: [redacted]

API Key: [redacted]

Save

Cloud Synchronization Status

Start

Figure : Enter API Key in the ThingPlus plugin

- Click the "Save" button (ThingPlus gateway app will restart)
- If the status is "STOPPED", then click on the "Start" button after saving API Key

17.2.9.2. Register Gateway and Sensors

- Go to your ThingPlus service
- Go to Gateway Management page (via upper right menu)
- Click the + button in the upper right corner
- Fill the form:
 - Select Gateway Model as "Libelium Meshlium"
 - Input Gateway ID (the same MAC address when registering gateway)
 - Select Device Model as "Waspote Basic"
 - Input Device Address (Waspote address is the id_wasp field)
 - Input Device Name
 - Select Sensors to register (all sensors are selected as default)
 - Select Site Name (default)
- Click "Register a Gateway, Devices and Sensors" button:
 - ThingPlus gateway app will restart and send the sensor data in a few minutes
 - You can see the sensor data at Dashboard or Sensor page

Gateway Management

Request for Gateway Certificates or API Key

Register Gateway

Gateway Model: Libelium Meshlium

Gateway ID: 00:0d:00:00:00:00

Gateway Name: Mesh GW Office 1

Device Model: Waspote Basic

Device ID: 000d:00:00:00:00:00-1

Device Address: 1

Device Name: Mesh Device Office 1

Sensors to create

Type	Name	ID	Bus	Address	Model
<input checked="" type="checkbox"/>	accelerometer_ACC <small>Required</small>	000d:00:00:00:00:00-1- accelerometer- ACC	1	1	libeliumAccelerometer
<input checked="" type="checkbox"/>	batteryGauge_BAT <small>Required</small>	000d:00:00:00:00:00-1- batteryGauge- BAT	1	1	libeliumBatteryGauge
<input checked="" type="checkbox"/>	string_MAC <small>Required</small>	000d:00:00:00:00:00-1-string-MAC	1	1	libeliumString
<input checked="" type="checkbox"/>	temperature_IN_TEMP <small>Required</small>	000d:00:00:00:00:00-1-temperature- IN_TEMP	1	1	libeliumTemp

Site Name: default

Add Device

Register a Gateway, Devices and Sensors

Figure : Registering a Gateway in ThingPlus service.

18. Smartphone Detection

The Meshlium Xtreme allows to detect **iPhone** and **Android** devices and in general any device which works with **Wifi** or **Bluetooth** interfaces.

These devices can be detected without the need of being connected to a specific Access Point, enabling the detection of any smartphone, laptop or hands-free car kit device which comes into the coverage area of Meshlium.

The idea is to be able to measure the amount of people and cars which are present in a certain point at a specific time, allowing the study of the evolution of the traffic congestion of pedestrians and vehicles.



Figure : Smartphone Detection

Users have to do nothing to be detected as the Wifi and Bluetooth radios integrated in their smartphones periodically send a "hello!" message telling about their presence. The information read from each user contains:

- The MAC address of the wireless interface, which allows to identify it uniquely
- The strength of the signal (RSSI), which gives us the average distance of the device from the scanning point
- The vendor of the smartphone (Apple, Nokia, etc)
- The Wifi Access Point where the user is connected (if any) and the Bluetooth friendly name. Users not connected to an AP will be shown as "free users".
- The Class of Device (CoD) in case of Bluetooth which allows us to differentiate the type of device (smartphone, hands-free, laptop, LAN/Network AP). With this parameter we can differentiate among pedestrians and vehicles

The coverage areas may be modified by changing the power transmission of the radio interfaces allowing the creation of different scanning zones from a few meters (in order to study a specific point) to dozens of meters (to study the whole street or even the entire floor of a shopping mall).

Applications related to Shopping and Street activities:

- Number of people passing daily in a street
- Average time of the stance of the people in a street
- Differentiate between residents (daily matches) and visitants (sporadic matches)
- Walking routes of people in shopping malls and average time in each area

The Vehicle Traffic Monitoring is also another important application as understanding the flow and congestion of vehicular traffic is essential for efficient road systems in cities. Smooth vehicle flows reduce journey times, reduce emissions and save energy. Similarly the efficient flow of pedestrians in an airport, stadium or shopping centre saves time and can make the difference between a good and a bad visit. Monitoring traffic - whether road vehicles or people - is useful for operators of roads, attractions and transport hubs.

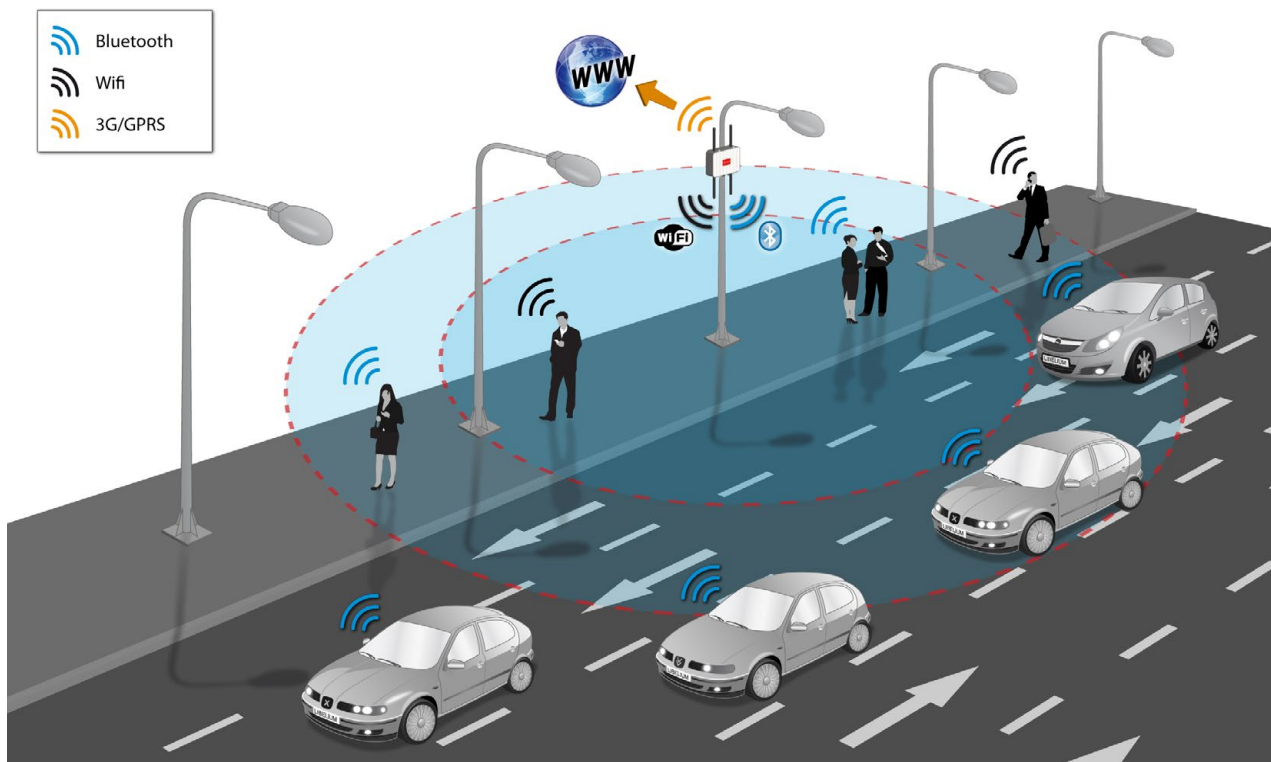


Figure : Vehicle Traffic Detection

Applications for Vehicle Traffic Detection:

- Monitor in real time the number of vehicles passing for a certain point in highways and roads
- Detect average time of vehicle stance for traffic congestion prevention
- Monitor average speed of vehicles in highways and roads
- Provide travel times on alternate routes when congestion is detected

The monitoring system can also be used to calculate the average speed of the vehicles which transit over a roadway by taking the time mark at two different points.

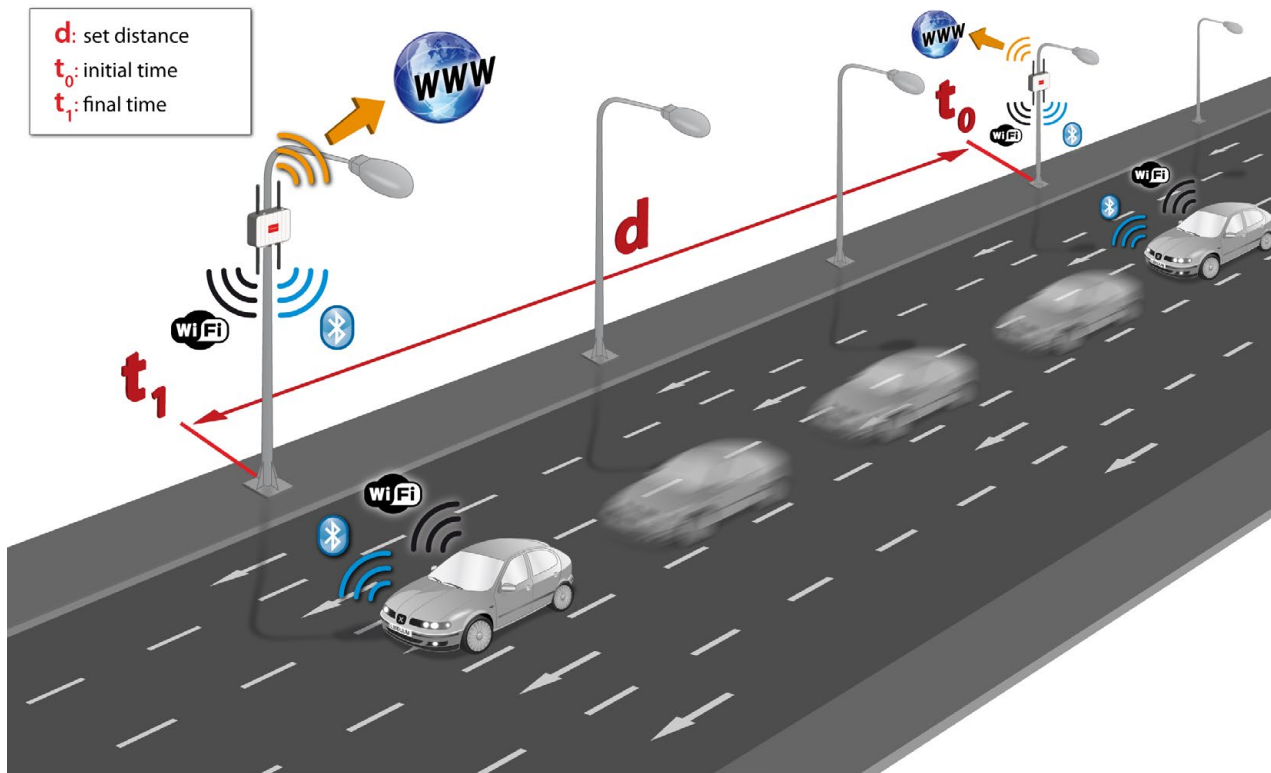


Figure : Calculate the average speed

18.1. Update! (Version 3.1.4)

The new version of the software included in **Meshlium Scanner (v3.1.4)** increments **from 70% to the 95%** the amount of smartphones, tablets, hands-free, and laptops detected by its MAC address by scanning WiFi and Bluetooth signals.

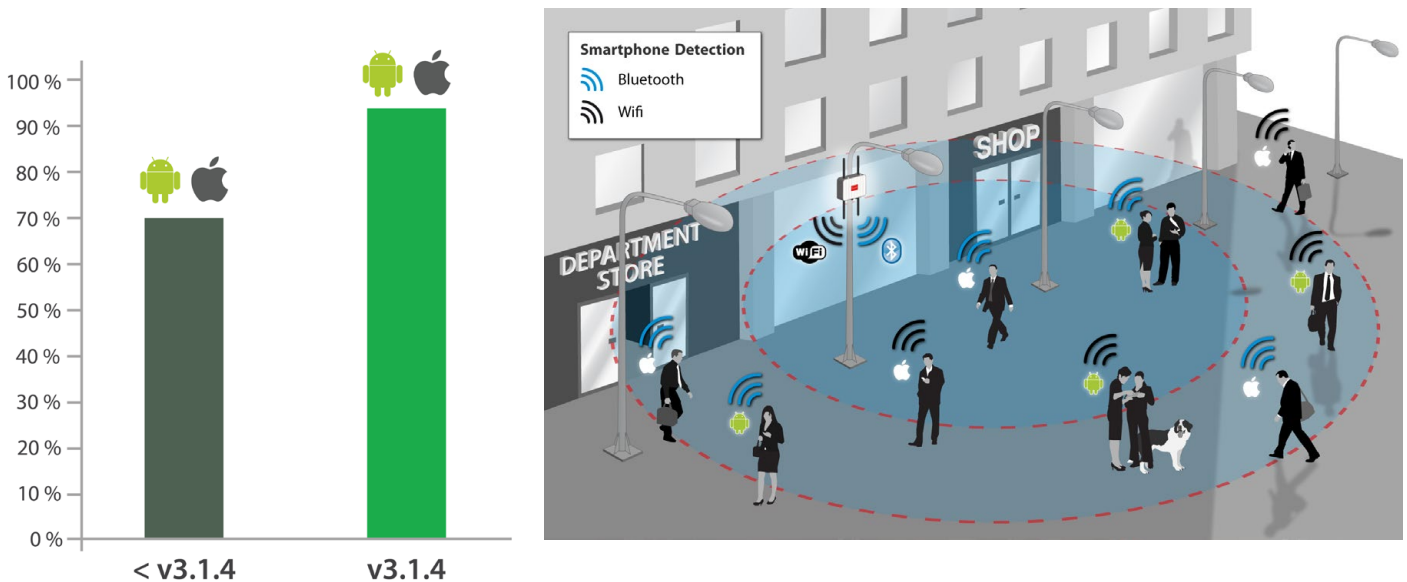


Figure : Now more devices are detected

Detection of the current version includes any of the last models even those that implement **low consumption techniques** when using the radio interfaces:

- iPhone (*all models*): 4, 4S, 5, 5S, 5C, iPad (2, 3, 4, Air, Mini, Retina)
- Android (*all models*): Nexus, Samsung Galaxy, LG, Sony Xperia, HTC, Motorola, Huawei, Asus,...



Figure : Some of the supported smartphones

Minimum time between scanning intervals

The improvement of the number of devices detected in the new version is due to the reduction between scanning intervals. Now it is just **1 second** against the 20 seconds of the previous versions. So even those models that have long cycles between advertising WiFi frames (beacons) will be captured.

Vehicle Traffic Monitoring

Due to the reduction of the time between scanning intervals, now vehicle traffic detection rate has increased **from 50% to 80%** even at a speed of 100 km/h (62 miles/h)

- Monitor in real time the number of vehicles passing for a certain point in highways and roads
- Detect average time of vehicle stance for traffic congestion prevention
- Monitor average speed of vehicles in highways and roads
- Provide travel times on alternate routes when congestion is detected

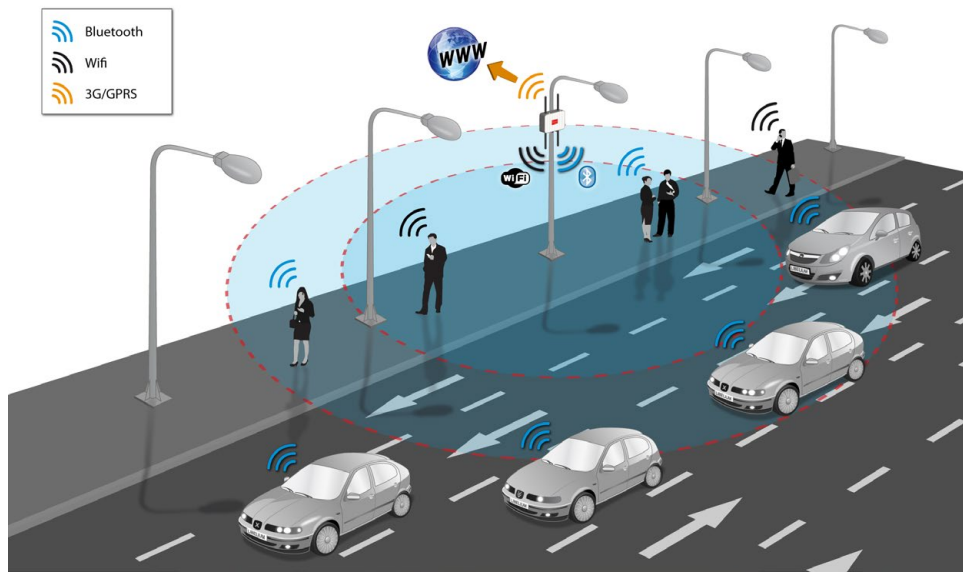


Figure : WiFi and Bluetooth scanning in the street

For Vehicle Traffic Monitoring applications, it is recommended to purchase the special pack of 2 directional antennas which will extend the range of WiFi and Bluetooth scanning in the required direction. The size of one antenna is 40 x 36 x 4 cm. Each one weights about 2 kg. The antennas are 18 dB and come with the needed mounting system, 2 m cables and screw adapters.



Figure : Directional antenna for Meshlium Scanner

Do the users need to have a specific app installed or interact somehow to be detected?

No, the scan is performed silently, Meshlium just detects the “beacon frames” originated by the Wifi and Bluetooth radios integrated in the smartphones. Users just need to have at least one of the two wireless interfaces turned on.

How do we differentiate if the Bluetooth device detected is a car’s hands-free or a smartphone?

In the scanning process each Bluetooth device gives its “Class of Device” (CoD) attribute which allows to identify the type of service it gives. We can differentiate easily the CoD’s generated by the car’s hands-free from the people’s phone ones.

How do I control the inquiry area?

In the Bluetooth inquiry there are seven different power levels which go from -27dBm to +3dBm in order to set different coverage zones from 10 to 50m. In both Wifi and Bluetooth radios these zones can also be increased or decreased by using a different antenna for the module as it counts with a standard N-Male connector. The default antenna which comes with the scanning modules is an omnidirectional antenna with a gain of 5dBi.

How do I calculate the distance of any of the devices detected?

In the inquiry process we receive the MAC address of the Bluetooth device along with the Received Signal Strength Indicator (RSSI) which gives us the quality of the transmission with each device. RSSI values usually go from -40dBm (nearest nodes) to -90dBm (farthest ones). In the tests performed Bluetooth devices at a distance of 10m reported -50dBm as average, while the ones placed at 50m gave us an average of -75dBm.

What about privacy?

The anonymous nature of this technique is due to the use of MAC addresses as identifiers. MAC addresses are not associated with any specific user account or mobile phone number not even to any specific vehicle. Additionally, the “inquiry mode” (visibility) can be turned off so people have always chosen if their device will or won’t be detectable.

How do the Bluetooth, Wifi and ZigBee radios coexist without causing interferences with each other?

Wifi, ZigBee and Bluetooth work in the 2.4GHz frequency band (2.400 - 2.480MHz), however, the Bluetooth radio integrated in Meshlium uses an algorithm called Adaptive Frequency Hopping (AFH) which improves the common algorithm used by Bluetooth (FHSS) and enables the Bluetooth radio to dynamically identify channels already in use by ZigBee and Wifi devices and to avoid them.

Anyway, in the case of sending 802.15.4 frames from Wasmote or Plug & Sense! to a Meshlium Scanner equipped with XBee 802.15.4, it is recommended to perform re-tries in the sender application, just to minimize possible interferences and ensure a good received frames percentage in Meshlium.

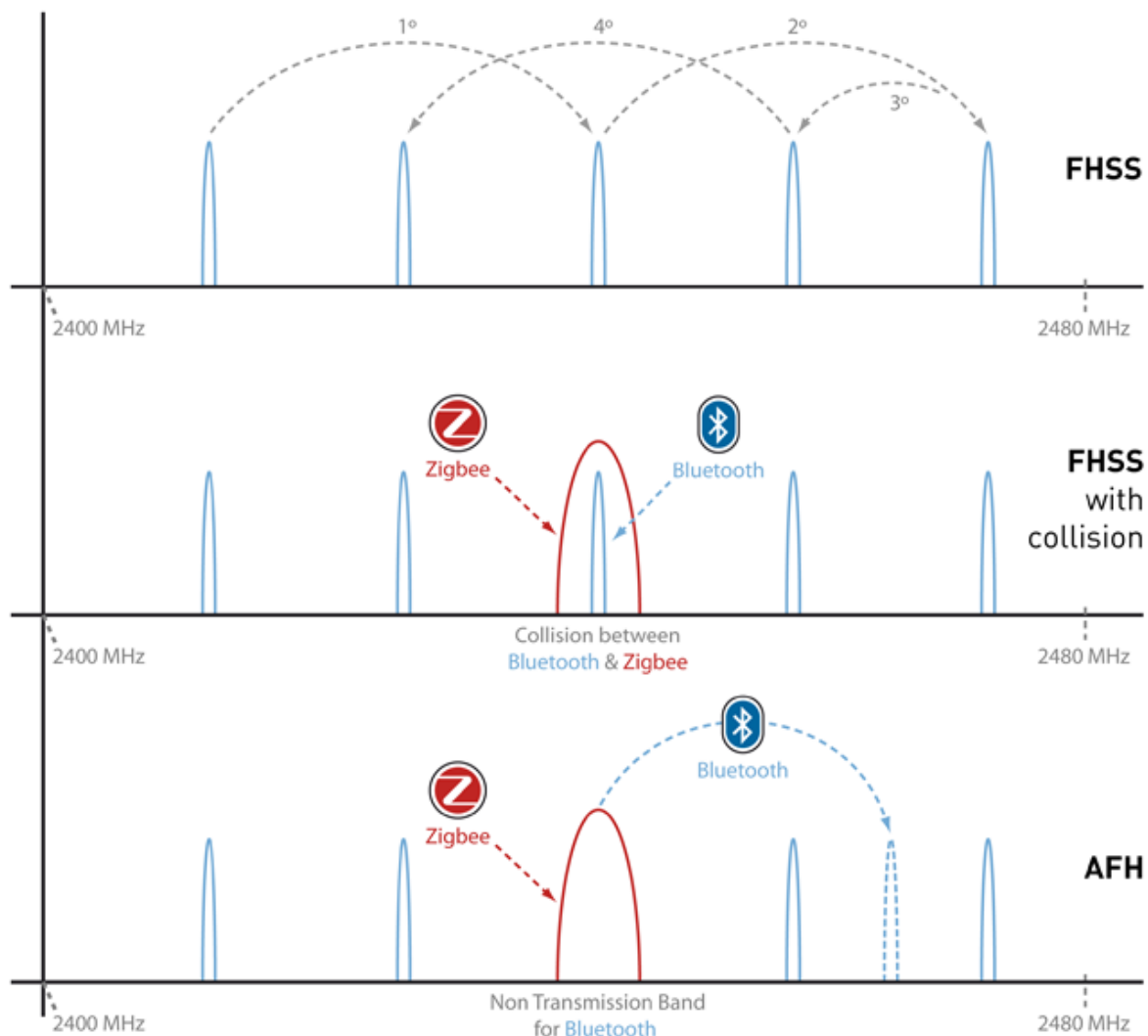


Figure : Bluetooth, WiFi and ZigBee radios coexist

Under which conditions do you get a 95% detection rate of devices?

A set of conditions must be respected to keep the detection rate high.

The devices to be detected must be some meters away from the Scanner and must remain some seconds inside the coverage area to give time to the system to detect them.

The setup of the WifiScan feature in Meshlium is 40 seconds of scan span. This means Meshlium Scanner listens for 40 seconds and then stores the results of the scan.

Android and iOS devices have a special option to disable WiFi connection when the user locks the screen in order to save battery. All cases are studied here. This option changes with the iOS version, but will be present in the majority of iOS devices. It makes iOS devices difficult to detect: the device is not undetectable, but it would need several scan cycles to get one beacon.

When a device is connected to a WiFi Access Point it is easier to detect, as it needs to send radio packets to allow communication.

The results not Android or iOS devices may vary depending on the type of system. Usually, APs are detected easily, as they broadcast the SSID. Hidden SSID are detected too. The only APs that can be hard to detect are the APs that do not broadcast their presence. This APs can only be detected when there is traffic from connected devices.

Regarding other WiFi devices, the individual behaviour will define if they are detectable. As a general rule, every device that broadcasts beams or is connected generating traffic will be detected.

Our tests results are shown in these tables:

Android:

	Screen ON	Screen OFF (power saving off)	Screen OFF (power saving on)
WiFi radio OFF	NO	NO	NO
WiFi radio ON (not connected to an AP)	YES (almost every scan cycle)	YES (most of scan cycle)	NO
WiFi radio ON (connected to an AP)	YES	YES	NO

iOS:

	Screen ON	Screen OFF (power saving off)
WiFi radio OFF	NO	NO
WiFi radio ON (not connected to an AP)	YES (after several scans)	SOMETIMES (after a random number of scans)
WiFi radio ON (connected to an AP)	YES	YES (after several scans)

Bluetooth scanning, unlike WiFi scanning, is based on polling, and not in passive listening. This makes Bluetooth detection slower and left the device the chance of avoiding detection (it just needs to ignore the polling request).

Nevertheless, Bluetooth scan is still useful in some applications, like car detection, as most of modern cars have a Bluetooth hands-free device, and these devices are most of the time listening for connections.

Any smartphone can be configured to be visible (or not) by other Bluetooth devices. Putting this option as "NOT VISIBLE" will make the smartphone undetectable by any other Bluetooth device, which includes Meshlium Scanner. Note that some latest-technology hands-free devices implement the "not visible" mode too.

When the Bluetooth interface is set as visible, the phone will be listening for incoming queries. This way the device can be scanned. The visibility setup may be different in different devices. Some of them activate visibility for a limited time (usually 30 seconds), some others have a manual control to enable/disable the visibility.

Different Android and iOS versions have different behaviours about Bluetooth visibility. In most of the modern versions, Bluetooth visibility is disabled when the screen is locked (or even when the user exits the Bluetooth configuration menu). There is no way to detect an Android or iOS phone with the screen off, which makes it very difficult to scan Android or iOS devices in a real environment.

There are a lot of types of Bluetooth devices. Most of slave Bluetooth devices are designed to wait for incoming connections. This makes highly possible to detect devices like hands-free car kits, headsets, HID, etc.

The scanning time is more important in Bluetooth as the devices need some time to reply to the queries.

Device name is not always obtained, as some devices take some time to reply to the name queries. Nevertheless, the device can be easily identified by its MAC address.

How can I calculate the total number of people from the number of detected devices?

It depends. Not all the people have a smartphone. Also, not all the people switch WiFi and/or Bluetooth radios on their smartphones. It all depends on so many economic, social and cultural factors. The percentage of people with WiFi or Bluetooth on depends on the scenario where they are too. For example, if a Meshlium Scanner is installed in a college campus which provides free WiFi service, many students will be detected because they will probably keep their smartphones, tablets or notebooks with WiFi on. The same would happen in a mall, airport or hotel with free WiFi.

Besides, consider that not all the people who could be detected will remain enough time inside of the coverage area of Meshlium Scanner.

Also, keep in mind that some people can carry several WiFi or Bluetooth devices. For example, a driver with smartphone in his pocket and a Bluetooth device in his car can be detected as 2 different users by Meshlium.

To sum up, in Libelium we consider that the total number of people can be approximated multiplying the number of detected devices by a factor, from 3 or 5:




$$3 * \text{Detected devices} < \text{Total people} < 5 * \text{Detected devices}$$

It all depends on a number of variables. The administrator of Meshlium Scanner can perform real tests in order to find the exact value of this factor in the specific scenario under study.




18.2. WiFi Scan

18.2.1. Concepts

The WiFi radio integrated in the new Meshlium Xtreme allows to scan WiFi devices in a range of action up to 200m depending on the line of sight conditions.

The idea is to search for WiFi devices in a defined interval which can be configured. Meshlium will get the **MAC address**, information about the detected **Device**. Regarding these devices, we can distinguish Access Points  and Clients  . In the case of each client, Meshlium gets which Access Point the device is connected to (if any). Also, the signal strength (**RSSI**) of the devices along with a **timestamp** which identifies when the scan detects it. For this reason, it is important to set the correct time in the System before starting with the storage of the data. See the Time Synchronization in the System section.

As extra information, the System also identifies the **Vendor** of the WiFi devices using its MAC address and if the information is synchronized to the external database (**Sync**). Example of information scanned:


DB ID	Sync	Timestamp	MAC	Device	RSSI	Vendor
53483	0	2012-04-24 07:56:25	C4:2C:03:96:0E:4A	  (not detected)	69	Apple
53482	1	2012-04-24 09:11:26	D8:2A:7E:10:1E:63	 libelium_wsn1	60	Nokia Corporation

We can select the Scanning Time from a drop-down list. This time specifies how many seconds the scanner will spend searching. After each scanning process, the system performs a pause of one second before starting again.

The Scanning Time must be trimmed in order to avoid that a temperature of 70 °C is reached in the Meshlium's microprocessor. See chapter "Internal temperature sensors" to know how to monitor the microprocessor's temperature.

Wifi scan

Scanning Time Seconds

 **Wifi Scan Running**

☐ Anonymize MAC

Please, note that if you choose a lower interval, you will get less results. By default 40 seconds.

Figure : Configuring WiFi Scanner

We can also activate the anonymization of the MAC addresses. This option will store the MAC address cyphered with an MD5 hash. The hash will be consistent in the same day, but will change from one day to another. This system allows to follow a particular user in the same day, but keeps the privacy of the user, not storing the real MAC of the device and not allowing to track a user more than one day.

From this section the user can start and stop the service from the button next to the status indicator.

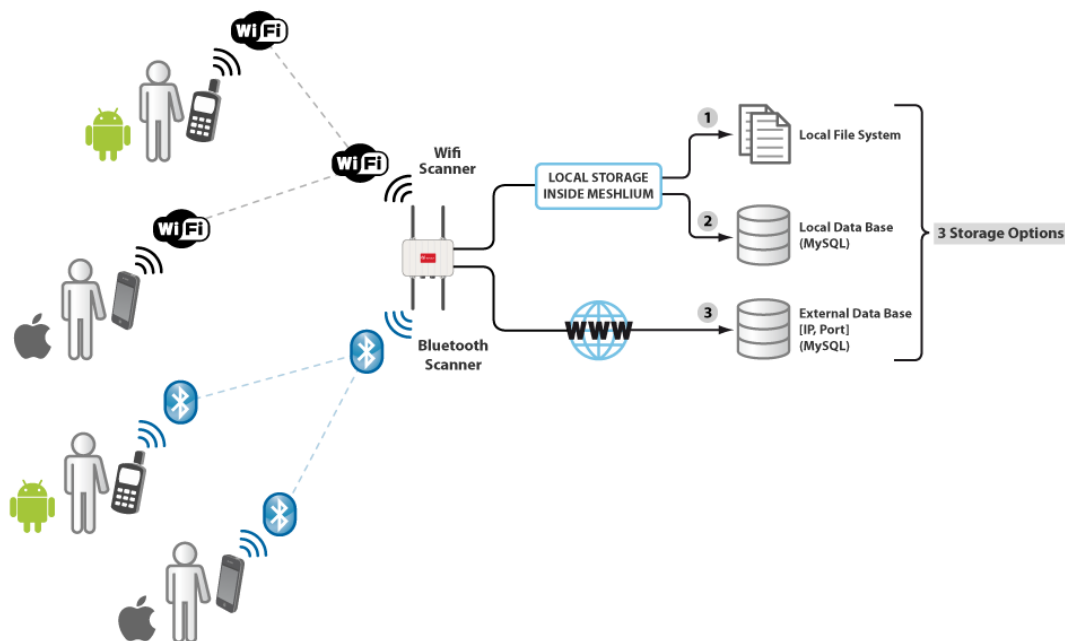


Figure : WiFi Scanner was stopped

18.2.2. Capturing and Storing WiFi Scans

We can perform two different storage options with the data captured.

- Local Data Base
- External Data Base



18.2.3. Local Data Base

Meshlium has a MySQL data base up and running which is used to store locally the information captured. In the “Local Data Base” tab you can see the connection parameters.

- **Database:** MeshliumDB
- **Table:** wifiData
- **IP:** localhost / 10.10.10.1 *
- **Port:** 3306
- **User:** root
- **Password:** libelium2007

Note: Password of the local MySQL Database can not be changed. An external Data Base may be used to store the frames with a new password. Go to the External Database tab for more information.”

(*) Depending on the parameters set in the Interfaces section.

Steps:

1. Set the check box “Store frames in the selected file” and press the “Save” button.

From this time Meshlium will automatically perform Scans and will store the results in the Local Data Base. This process will also continue after restarting Meshlium.

At any time you can see the last “x” records stored, filtered by access points or clients if you want. Just set how many and what kind of insertions you want to see and press the “Show data” button.

The data from the database can be deleted pressing the button “Delete Data”. Be careful, as this option deletes all the information of WiFi scans in local database.

There is an option to program an automatic purge in the database every day, keeping the information in the database the days you want to specify. Furthermore, if you intend to configure the external database, you can choose if you want to delete only synchronized data or everything, taking care of the days established before.

Captured Data

Local DataBase

External DataBase

Connection data
Database: MeshliumDB
Table: wifiScan
Host: localhost
Port: 3306
User: root
Password: libelium2007

☒ **Store frames in the local data base** Save
☒ **Auto-purge**
Keep the last days in the database
☐ deleting only synchronized data
☒ deleting all data Save
☒ Access points
☒ Clients
Last insertions.
Show data Delete all data

TimeStamp	Sync	MAC	Device	RSSI	Vendor
2015-11-18 11:36:09	1	1ea3fdb90b59049be	60e09eb5bb9a8351a	28	Liteon Technology Corporati
2015-11-18 11:36:07	1	5ac401c56b456233b	74e38b1c424a02d6a	23	Liteon Technology Corporati
2015-11-18 11:36:00	1	ee7fdb707182ed82c	a100f5623c9af7e5a	40	LG Electronics
2015-11-18 11:35:46	1	ce34216c02aa2560e	01ae40418d84058f9	34	LG Electronics
2015-11-18 11:35:43	1	399eae15fd186c2c3	570bfb738c3c05bb2	36	LG Electronics
2015-11-18 11:35:43	1	9077a1c77b19fb902	01ae40418d84058f9	35	Mundo Reader (bq)
2015-11-18 11:35:43	1	59d75f7c53afea46d	74e38b1c424a02d6a	33	Hon Hai Precision Ind. Co.,L
2015-11-18 11:35:42	1	28484067168ab8e40	ba7ada682d8028e76	30	Mundo Reader (bq)
2015-11-18 11:35:37	1	83aa0f610321fca7e	01ae40418d84058f9	38	Lenovo Mobile Communicat
2015-11-18 11:35:32	1	e1ed5ce8c17c4a80f	74e38b1c424a02d6a	28	Liteon Technology Corporati
2015-11-18 11:35:32	1	df4c9e5bfc0b4908		34	Unknown
2015-11-18 11:35:32	1	74e38b1c424a02d6a		32	Unknown
2015-11-18 11:35:32	1	d0bb05bc5c68c341b		32	Unknown
2015-11-18 11:35:32	1	078ca52e83929bcd0		32	Unknown
2015-11-18 11:35:32	1	a100f5623c9af7e5a		31	Unknown

Figure : Local database for WiFi Scanner

18.2.4. External Data Base

Meshlium can also store the information captured in an External Data Base, keeping all the records from the local database synchronized.

Note: From 3.2.3 version the system has been improved marking all data synchronized from local database. If you have configured a WiFi External DataBase, you have to delete all your table in this external server and start this process again in order to maintain the data integrity and a proper functionality.

Steps:

1. Pressing the "Show sql script" you will get the code needed to create the data base along with the table and the right privileges.

Captured Data

Local DataBase
External DataBase

Connection data

Database: MeshliumDB

Table: wifiScan

Host: mysql.mydomain.com

Port: 3306

User: root

Password: password

Save
Check Connection

☒ Store frames in the external data base
Save

☒ Access points
☒ Clients

Last 100 insertions.

Show data
Show sql script

Just copy and paste:

```

CREATE database MeshliumDB;

CREATE TABLE `wifiScan` (
  `ID_frame` int(11) NOT NULL AUTO INCREMENT,
  `TimeStamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `MAC` varchar(17) COLLATE utf8_unicode_ci NOT NULL,
  `SSID` varchar(32) COLLATE utf8_unicode_ci NOT NULL,
  `RSSI` varchar(3) COLLATE utf8_unicode_ci NOT NULL,
  `Vendor` varchar(150) COLLATE utf8_unicode_ci NOT NULL,
  `Type` varchar(45) COLLATE utf8_unicode_ci NOT NULL,
  `AP` varchar(17) COLLATE utf8_unicode_ci NOT NULL,
  `MeshliumID` varchar(150) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`ID_frame`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;

GRANT ALL PRIVILEGES ON *.* TO root@'%' IDENTIFIED BY 'passw';

```

2. Insert this code in your MySQL management application.
3. Fill the Connection Data fields with the information about where the data base is located (IP, Port) and with the authentication options (Database, Table, User, Password).

- Now press the "Check Connection" button to see if the configuration is correct.

Captured Data

Local DataBase External DataBase

Connection data

Database: MeshliumDB

Table: wifiScan


Host: mysql.mydomain.com

Port: 3306

User: root

Password: password

Save Check Connection

 ☒ Store frames in the external data base Save

☒ Access points
☒ Clients

Last 100 insertions. (to create database and table)

Show data Show sql script

Connecting to the database server ...
 Sending Inquiry ...
 Query generated with id: 1
OK

5. Set the check box “Store frames in the selected file” and press the “Save” button.

From this time Meshlium will automatically perform Scans and will store the results in the Local Data Base. This process will also continue after restarting Meshlium.

At any time you can see the last “x” records stored, filtered by access points or clients if you want. Just set how many and what kind of insertions you want to see and press the “Show data” button.

Captured Data

Local DataBase External DataBase

Connection data

Database: MeshliumDB

Table: wifiScan


Host: mysql.mydomain.com

Port: 3306

User: root

Password: password
















Save **Check Connection**

 ☒ **Store frames in the external data base** **Save**

☒ Access points
☒ Clients

Last insertions. (to create database and table)

Show data **Show sql script**

TimeStamp	MeshliumID	MAC	Device	RSSI	Vendor
2015-11-18 11:36:09	meshlium	1ea3fdb90b59049be	 60e09eb5bb9a8351a	28	Liteon Technology C
2015-11-18 11:36:07	meshlium	5ac401c56b456233b	 74e38b1c424a02d6a	23	Liteon Technology C
2015-11-18 11:36:00	meshlium	ee7fdb707182ed82c	 a100f5623c9af7e5a	40	LG Electronics
2015-11-18 11:35:46	meshlium	ce34216c02aa2560e	 01ae40418d84058f9	34	LG Electronics
2015-11-18 11:35:43	meshlium	399eae15fd186c2c3	 570bfb738c3c05bb2	36	LG Electronics
2015-11-18 11:35:43	meshlium	9077a1c77b19fb902	 01ae40418d84058f9	35	Mundo Reader (bq)
2015-11-18 11:35:43	meshlium	59d75f7c53afea46d	 74e38b1c424a02d6a	33	Hon Hai Precision Ir
2015-11-18 11:35:42	meshlium	28484067168ab8e40	 ba7ada682d8028e76	30	Mundo Reader (bq)
2015-11-18 11:35:37	meshlium	83aa0f610321fca7e	 01ae40418d84058f9	38	Lenovo Mobile Com
2015-11-18 11:35:32	meshlium	e1ed5ce8c17c4a80f	 74e38b1c424a02d6a	28	Liteon Technology C
2015-11-18 11:35:32	meshlium	df4c9e5bfc0b4908	 74e38b1c424a02d6a	34	Unknown
2015-11-18 11:35:32	meshlium	74e38b1c424a02d6a	 74e38b1c424a02d6a	32	Unknown
2015-11-18 11:35:32	meshlium	d0bb05bc5c68c341b	 078ca52e83929bcd0	32	Unknown
2015-11-18 11:35:32	meshlium	078ca52e83929bcd0	 078ca52e83929bcd0	32	Unknown
2015-11-18 11:35:32	meshlium	a100f5623c9af7e5a	 a100f5623c9af7e5a	31	Unknown

18.3. Bluetooth Scans

18.3.1. Concepts

This radio integrated in Meshlium allows to scan Bluetooth devices in a range of action up to 200m depending on the line of sight conditions.

The idea is to search for Bluetooth devices in a defined interval which can be configured. Meshlium will get the **MAC address**, the **Bluetooth ID** and the **RSSI** of the devices along with a **timestamp** which identifies when the scan was performed. For this reason it is important to set the correct time in the System before starting with the storage of the data. See the Time Synchronization in the System section.

Other interesting parameters the system also detects are the **Class of Device (CoD)** which allows us to differentiate the type of device (smartphone, hands-free, laptop, LAN/Network AP) and the **Vendor** of the Bluetooth devices using its MAC address.

With these parameters we can differentiate among pedestrians and vehicles.

DB ID	Timestamp	MAC	ID	RSSI	CoD	Vendor
45400	2012-05-16 16:18:12	00:26:7e:5f:3c:18	myCar	-72	Handsfree	PARROT SA
78005	2012-04-20 12:59:27	D8:2A:7E:0E:C3:10	Tropic	-85	Smartphone	Nokia Corporation

We can configure the Scanning Type which specifies the use of our Bluetooth Scanner:

- Indoor type is recommended to scan static devices or devices with slow movement (offices, malls, etc). This option retrieves device names after 10 seconds scanning.
- Outdoor type focus on devices which stay a brief period of time in our Bluetooth action range (roads, highways,...). This option does not ask the device name and the scanning period is 40 seconds.

In both types, there is a second between two consecutive scans.

Bluetooth scan


Name	<input type="text" value="MeshliumBT"/>	 Bt Scan Running <input type="button" value="Stop Service"/>
Visible	<input checked="" type="checkbox"/> Activate	
Scanning Type	<input type="text" value="OUTDOOR"/>	
Anonymize MAC	<input type="checkbox"/> Activate	

Figure : Configuring Bluetooth Scanner

We can also activate the anonymization of the MAC addresses. This option will store the MAC address cyphered with an MD5 hash. The hash will be consistent in the same day, but will change from one day to another. This system allows to follow a particular user in the same day, but keeps the privacy of the user, not storing the real MAC of the device and not allowing to track a user more than one day.

From this section the user can start and stop the service from the button next to the status indicator.

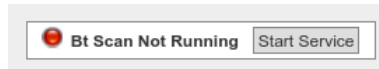


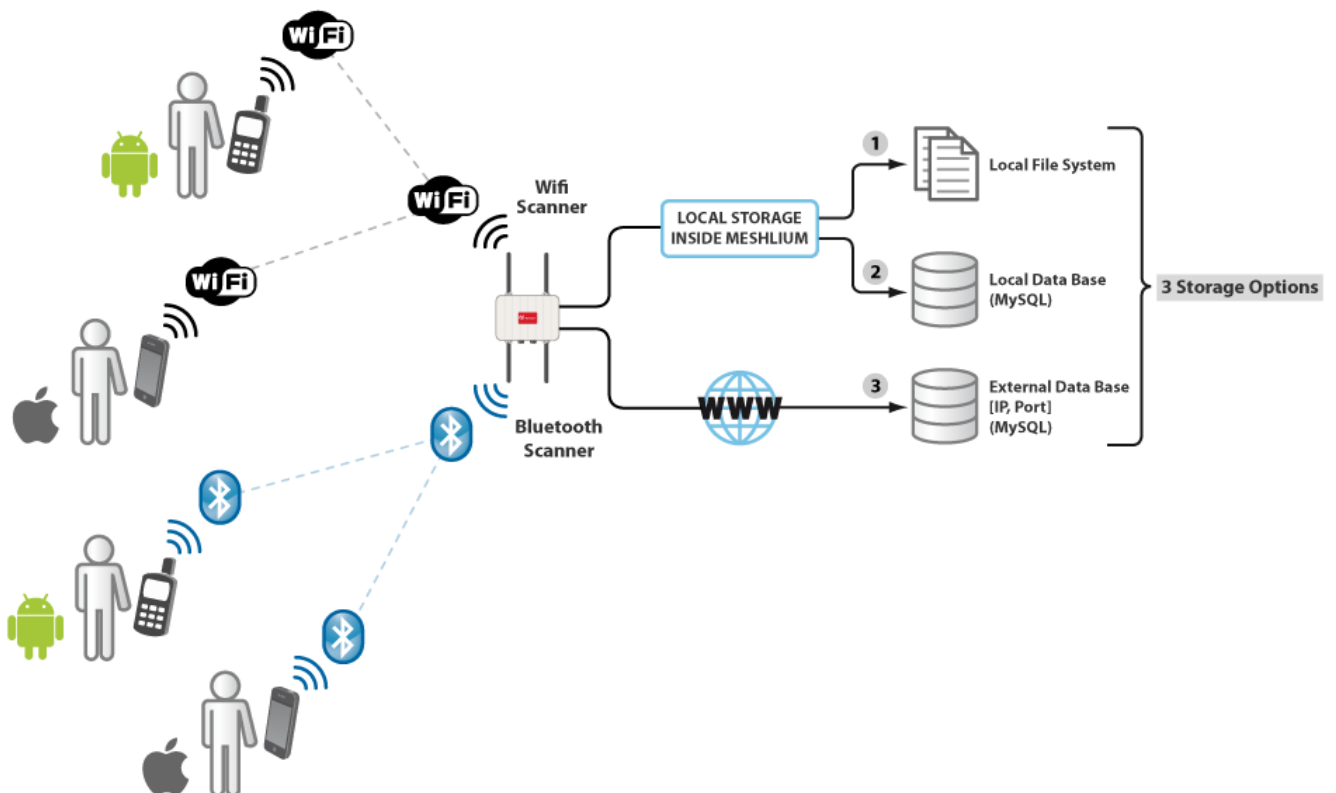
Figure : Bluetooth Scanner was stopped

Note: Last versions of Android and iOS devices may need the Bluetooth Setup Screen be activated to be detected.

Capturing and Storing Bluetooth Scans

We can perform three different storage options with the data captured.

- Local File System
- Local Data Base
- External Data Base



18.3.2. Local Data Base

Meshlium has a MySQL data base up and running which is used to store locally the information captured. In the “Local Data Base” tab you can see the connection parameters.

- **Database:** MeshliumDB
- **Table:** bluetoothData
- **IP:** localhost / 10.10.10.1 *
- **Port:** 3306
- **User:** root
- **Password:** libelium2007

Note: Password of the local MySQL Database can not be changed. An external Data Base may be used to store the frames with a new password. Go to the External Database tab for more information.”

(*) Depending on the parameters set in the Interfaces section.

Steps:

1.Set the check box “Store frames in the selected file” and press the “Save” button.

From this time Meshlium will automatically perform Scans and will store the results in the Local Data Base. This process will also continue after restarting Meshlium.

At any time you can see the last “x” records stored. Just set how many insertions you want to see and press the “Show data” button.

The data from the database can be deleted pressing the button “Delete Data”. Be careful, as this option deletes all the information of WiFi scans in local database.

Captured Data

Local Database External DataBase Local file

Connection data

Database: MeshliumDB

Table: bluetoothData

IP: localhost

Port: 3306

User: root

Password: libelium2007

☒ Store frames in the local data base

Last 100 insertions.

TimeStamp	MAC	ID	RSSI	CoD	Vendor
2015-06-29 09:19:12	00:0C:29:00:00:00	00:0C:29:00:00:00	-83	Computer	Intel Corporate
2015-06-29 09:19:12	00:0C:29:00:00:00	00:0C:29:00:00:00	-70	Smartphone	LG Electronics
2015-06-29 09:19:12	00:0C:29:00:00:00	00:0C:29:00:00:00	-44	Computer	Cisco-Linksys, LLC
2015-06-29 09:17:56	00:0C:29:00:00:00	00:0C:29:00:00:00	-83	Computer	Intel Corporate
2015-06-29 09:17:56	00:0C:29:00:00:00	00:0C:29:00:00:00	-70	Smartphone	LG Electronics
2015-06-29 09:17:56	00:0C:29:00:00:00	00:0C:29:00:00:00	-48	Computer	Cisco-Linksys, LLC
2015-06-29 09:17:29	00:0C:29:00:00:00	00:0C:29:00:00:00	-70	Smartphone	LG Electronics
2015-06-29 09:17:29	00:0C:29:00:00:00	00:0C:29:00:00:00	-43	Computer	Cisco-Linksys, LLC
2015-06-29 09:17:02	00:0C:29:00:00:00	00:0C:29:00:00:00	-71	Smartphone	LG Electronics
2015-06-29 09:17:02	00:0C:29:00:00:00	00:0C:29:00:00:00	-43	Computer	Cisco-Linksys, LLC
2015-06-29 09:16:35	00:0C:29:00:00:00	00:0C:29:00:00:00	-85	Computer	Intel Corporate
2015-06-29 09:16:35	00:0C:29:00:00:00	00:0C:29:00:00:00	-77	Smartphone	LG Electronics
2015-06-29 09:16:35	00:0C:29:00:00:00	00:0C:29:00:00:00	-52	Computer	Cisco-Linksys, LLC
2015-06-29 09:16:08	00:0C:29:00:00:00	00:0C:29:00:00:00	-80	Computer	Intel Corporate
2015-06-29 09:16:08	00:0C:29:00:00:00	00:0C:29:00:00:00	-66	Smartphone	LG Electronics

Figure : Local database for Bluetooth Scanner

18.3.3. External Data Base

Meshlium can also store the information captured in an External Data Base.

Steps:

1. Pressing the "Show sql script" you will get the code needed to create the data base along with the table and the right privileges.

Captured Data

Local Database External DataBase Local file

Connection data

Database: MeshliumDB


Table: bluetoothData

IP: 192.168.1.37

Port: 3306

User: root

Password: libelium2007

 ☒ Store frames in the external data base

Last insertions. (to create database and table)

Just copy paste:

```
CREATE database MeshliumDB;
```

Just copy paste:

```
CREATE TABLE IF NOT EXISTS `bluetoothData` (
  `ID_frame` int(11) NOT NULL auto_increment,
  `TimeStamp` timestamp NOT NULL default CURRENT_TIMESTAMP,
  `MAC` varchar(17) collate utf8_unicode_ci NOT NULL,
  `ID` text collate utf8_unicode_ci NOT NULL,
  `RSSI` varchar(3) collate utf8_unicode_ci NOT NULL,
  `Vendor` text collate utf8_unicode_ci NOT NULL,
  `cod` varchar(20) collate utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`ID_frame`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
```

Just copy paste:

```
GRANT ALL PRIVILEGES ON *.* TO root@'%' IDENTIFIED BY 'passw';
```

2. Insert this code in your MySQL management application.
3. Fill the Connection Data fields with the information about where the data base is located (IP, Port) and with the authentication options (Database, Table, User, Password).
4. Now press the "Check Connection" button to see if the configuration is correct.

Captured Data

Local Database | External DataBase | Local file

Connection data

Database: MeshliumDB


Table: bluetoothData

IP: 192.168.1.37

Port: 3306

User: root

Password: libelium2007

 ☒ Store frames in the external data base Save

Show data Last insertions. Show sql script (to create database and table)

Save Check Connection

Connecting to the database server ...
 Sending Inquiry ...
 Query generated with id: 2033
OK

5. Set the check box “Store frames in the selected file” and press the “Save” button.

From this time Meshlium will automatically perform Scans and will store the results in the Local Data Base. This process will also continue after restarting Meshlium.

At any time you can see the last “x” records stored. Just set how many insertions you want to see and press the “Show data” button.

Captured Data

Local Database | External DataBase | Local file

Connection data

Database: MeshliumDB


Table: bluetoothData

IP: 192.168.1.37

Port: 3306

User: root

Password: libelium2007

 ☒ Store frames in the external data base

Last insertions. (to create database and table)

TimeStamp	MAC	ID	RSSI	CoD	Vendor
2015-06-29 07:29:26	00:0C:29:00:00:00	meshlium_db	-79	Computer	Intel Corporate
2015-06-29 07:29:26	00:0C:29:00:00:00	meshlium_db	-66	Smartphone	LG Electronics
2015-06-29 07:29:26	00:0C:29:00:00:00	meshlium_db	-42	Computer	Cisco-Linksys, LLC
2015-06-29 07:29:00	00:0C:29:00:00:00	meshlium_db	-89	Computer	Intel Corporate
2015-06-29 07:29:00	00:0C:29:00:00:00	meshlium_db	-70	Smartphone	LG Electronics
2015-06-29 07:29:00	00:0C:29:00:00:00	meshlium_db	-44	Computer	Cisco-Linksys, LLC
2015-06-29 07:28:33	00:0C:29:00:00:00	meshlium_db	-82	Computer	Intel Corporate
2015-06-29 07:28:33	00:0C:29:00:00:00	meshlium_db	-68	Smartphone	LG Electronics
2015-06-29 07:28:33	00:0C:29:00:00:00	meshlium_db	-52	Computer	Cisco-Linksys, LLC
2015-06-29 07:28:06	00:0C:29:00:00:00	meshlium_db	-88	Computer	Intel Corporate
2015-06-29 07:28:06	00:0C:29:00:00:00	meshlium_db	-71	Smartphone	LG Electronics
2015-06-29 07:28:06	00:0C:29:00:00:00	meshlium_db	-43	Computer	Cisco-Linksys, LLC
2015-06-29 07:27:39	00:0C:29:00:00:00	meshlium_db	-83	Computer	Intel Corporate
2015-06-29 07:27:39	00:0C:29:00:00:00	meshlium_db	-83	Handsfree	PARROT
2015-06-29 07:27:39	00:0C:29:00:00:00	meshlium_db	-78	Smartphone	LG Electronics

18.3.4. Local File System

Steps:

1. Give a name to create a new file where the Bluetooth scans will be saved.
2. Select this file and press the "Select file" button.
3. Set the check box "Store frames in the selected file" and press the "Save" button.

From now on, Meshlium will automatically perform scans and will store the results in this file. This process will also continue after restarting Meshlium.

The file will be created in the folder `/mnt/user/bluetooth_data` and can be downloaded just selecting it and pressing the "Download" button.

At any time you can see the last "x" lines added to the file. Just set how many lines you want to see and press the "Show data" button.

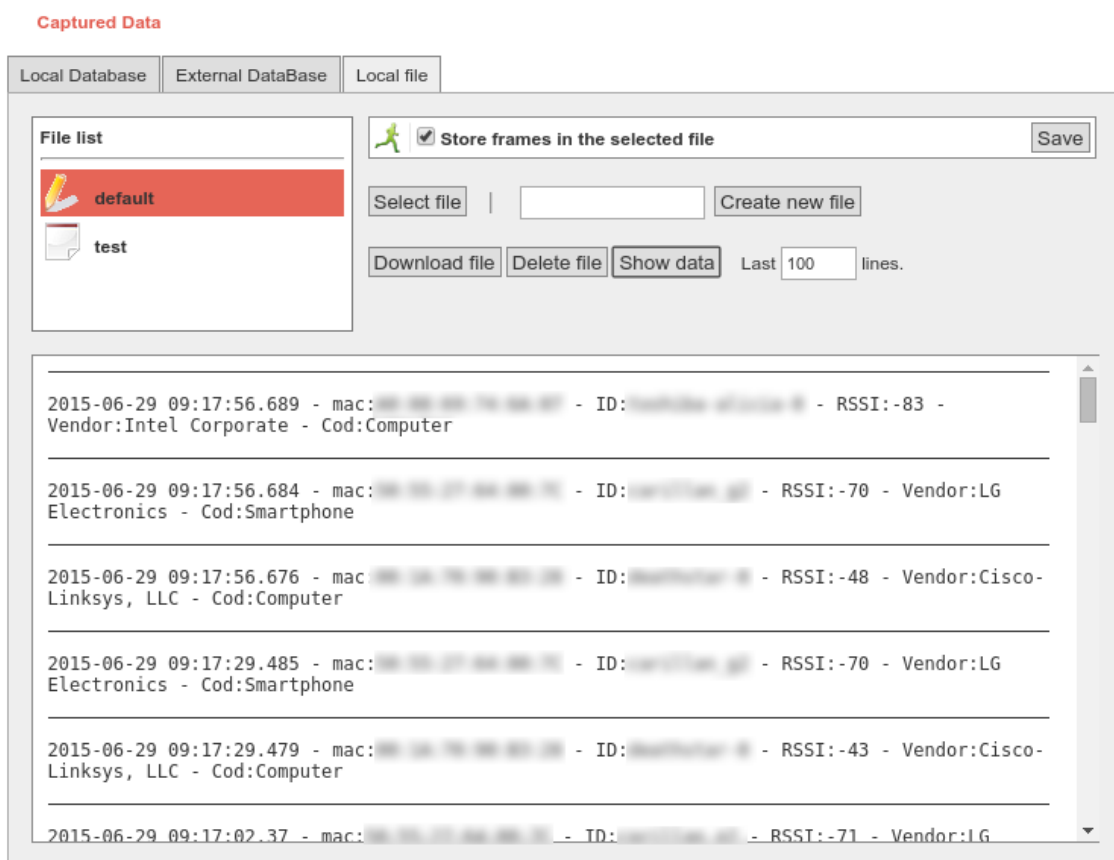


Figure : Operating with Bluetooth Scanner local file

19. GPS

19.1. Concepts

Meshlium can integrate a GPS receiver which allows to know the **exact location** of the router any time. It is specially interesting for mobile and vehicular applications and when setting long range links as the **GPS position** also gives information about the height of each point so the Fresnel Zone can be accurately known.

The GPS module gives us information about:

- latitude
- longitude
- height (meters)
- speed (km/h)
- date/time

- Standalone mode (NMEA frames):

This is the simplest mode: the GPS module calculates the NMEA frames which contain all the important parameters (latitude, longitude, speed, height, date and time). This mode only works outdoors with clear view of the sky.

The GPS receiver works with the NMEA mode (National Marine Electronic Association).

The different types of NMEA statements supported by Meshlium are:


- NMEA GGA: provides location data and an indicator of data accuracy.
- NMEA GSA: provides the status of the satellites the GPS receiver has been connected to.
- NMEA GSV: provides information about the satellites the GPS receiver has been connected to.
- NMEA RMC: provides information about the date, time, location and speed.
- NMEA VTG: provides information about the speed and course of the GPS receiver.


The most important NMEA statements are the GGA statements, which provide a validity indicator of the measurement carried out, the RMC statement which provides location, speed and date/time and the GSA statement which provides information about the status of the satellites the GPS receiver has been connected to.


19.2. Configuring GPS service

Meshlium has a service to store GPS data. The user can set the time interval between data acquisition and manually start and stop the service.

GPS

Scan interval	<input type="text" value="30"/>	Seconds	 GPS Service Running <input type="button" value="Stop Service"/>
			<input type="button" value="Save"/>

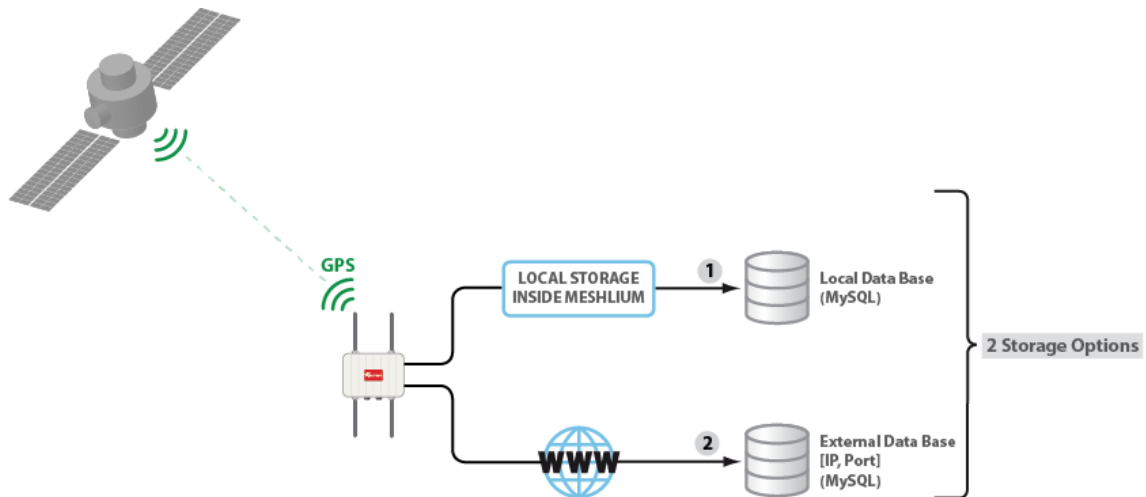
 **GPS Service Running**

 **GPS Service Not Running**

19.3. Capturing and Storing GPS Data

We can perform two different storage options with the data captured.

- Local Data Base
- External Data Base



19.3.1. Local Data Base

Meshlium has a MySQL data base up and running which is used to store locally the information captured. In the “Local Data Base” tab you can see the connection parameters.

- **Database:** MeshliumDB
- **Table:** gpsData
- **IP:** localhost / 10.10.10.1 *
- **Port:** 3306
- **User:** root
- **Password:** libelium2007

Note: Password of the local MySQL Database can not be changed. An external Data Base may be used to store the frames with a new password. Go to the External Database tab for more information.”

(*) Depending on the parameters set in the Interfaces section.

Meshlium will automatically perform Scans and will store the results in the Local Data Base. This process will also continue after restarting Meshlium.

At any time you can see the last “x” records stored. Just set how many insertions you want to see and press the “Show data” button.

Captured Data

Local DataBase External Database Show me NOW

Connection data

Database: MeshliumDB

Table: gpsData

IP: localhost

Port: 3306

User: root

Password: *****

Show data Last 100 insertions. (500 Max.)

ID	Date	Longitude	Latitude	Altitude	Speed
7476	2015-03-30 06:53:21 GMT	00051.417077,W	4140.132024,N	256.0 m	0 kn
7475	2015-03-30 06:52:51 GMT	00051.417067,W	4140.132094,N	256.0 m	0 kn
7474	2015-03-30 06:52:21 GMT	00051.417077,W	4140.132227,N	256.0 m	0 kn
7473	2015-03-30 06:51:51 GMT	00051.417057,W	4140.132366,N	255.5 m	0 kn
7472	2015-03-30 06:51:21 GMT	00051.417026,W	4140.132469,N	255.0 m	0 kn
7471	2015-03-30 06:50:51 GMT	00051.416911,W	4140.132519,N	255.0 m	0 kn
7470	2015-03-30 06:50:21 GMT	00051.416896,W	4140.132725,N	254.0 m	0 kn
7469	2015-03-30 06:49:51 GMT	00051.416871,W	4140.132959,N	253.5 m	0 kn

19.3.2. External Data Base

Meshlium can also store the information captured in an External Data Base.


Steps:

1. Pressing the "Show sql script" you will get the code needed to create the data base along with the table and the right privileges.

Captured Data

Local DataBase External Database Show me NOW

Connection data
Database: MeshliumDB
Table: gpsData
Host: 192.168.1.68
Port: 3306
User: root
Password:

 ☒ Store frames in the external data base
 Last insertions (max. 500).

Just copy paste:

```
CREATE database MeshliumDB;
```

Just copy paste:

```
CREATE TABLE IF NOT EXISTS `gpsData` (
  `ID_frame` int(11) NOT NULL auto increment,
  `TimeStamp` timestamp NOT NULL default CURRENT_TIMESTAMP,
  `date` DATETIME NOT NULL,
  `longitude` text collate utf8_unicode_ci NOT NULL,
  `latitude` text collate utf8_unicode_ci NOT NULL,
  `altitude` text collate utf8_unicode_ci NOT NULL,
  `satellites` int(11) NOT NULL,
  `speed` text collate utf8_unicode_ci NOT NULL,
  `sync` int(1) NOT NULL default '0',
  PRIMARY KEY (`ID_frame`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;
```

Just copy paste:

2. Insert this code in your MySQL management application.
3. Fill the Connection Data fields with the information about where the data base is located (IP, Port) and with the authentication options (Database, Table, User, Password).

- Now press the "Check Connection" button to see if the configuration is correct.

Captured Data

Local DataBase External Database Show me NOW

Connection data

Database: MeshliumDB

Table: gpsData

Host: 192.168.1.68

Port: 3306

User: root

Password:

☒ Store frames in the external data base

Last 100 insertions (max. 500).

Connecting to the database server ...
 Sending Inquiry ...
 Query generated with id: 3879
OK

- Set the check box "Store frames in the selected file" and press the "Save" button.

☒ Store frames in the external data base

From this time Meshlium will automatically perform Scans and will store the results in the Local Data Base. This process will also continue after restarting Meshlium.

At any time you can see the last “x” records stored. Just set how many insertions you want to see and press the “Show data” button.

Captured Data

Local DataBase External Database Show me NOW

Connection data

Database: MeshliumDB


Table: gpsData

Host: 192.168.1.68

Port: 3306

User: root

Password:

 ☒ Store frames in the external data base Save

Show data Last insertions (max. 500). Show sql script

Save Check Connection

ID	Date	Longitude	Latitude	Altitude	Speed
3876	2015-03-20 14:01:42 GMT	00051.419144,W	4140.134510,N	248.0 m	0 kn
3875	2015-03-20 14:02:13 GMT	00051.419149,W	4140.134493,N	248.0 m	0 kn
3874	2015-03-20 14:02:43 GMT	00051.419129,W	4140.134448,N	248.0 m	0 kn
3873	2015-03-20 14:00:41 GMT	00051.417102,W	4140.129499,N	262.5 m	0 kn
3872	2015-03-20 14:01:11 GMT	00051.418666,W	4140.134495,N	248.5 m	0 kn
3871	2015-03-20 13:59:41 GMT	00051.416981,W	4140.129381,N	263.0 m	0 kn
3870	2015-03-20 14:00:12 GMT	00051.417062,W	4140.129466,N	262.5 m	0 kn
3869	2015-03-20 13:58:42 GMT	00051.417026,W	4140.129406,N	263.0 m	0 kn

19.3.3. Show me now!

In the “Show me now!” tab you can see in real time the Scans captured.

You can specify if you want the information to be updated periodically with the defined interval just checking the “Use the Defined Interval” button.

Captured Data

Local DataBase	External Database	Show me NOW								
<div><div>Start Scan</div><div><input type="checkbox"/> Use the defined Scan interval</div></div> <div><p>2015-03-30 06:53:53 GMT</p><table><tbody><tr><td>Longitude:</td><td>00° 051.417117,W</td></tr><tr><td>Latitude:</td><td>41° 40.131956,N</td></tr><tr><td>Altitude:</td><td>256.0 m</td></tr><tr><td>Speed:</td><td>0 knots</td></tr></tbody></table></div>			Longitude:	00° 051.417117,W	Latitude:	41° 40.131956,N	Altitude:	256.0 m	Speed:	0 knots
Longitude:	00° 051.417117,W									
Latitude:	41° 40.131956,N									
Altitude:	256.0 m									
Speed:	0 knots									

20. Networking tools

20.1. Ping

It let you test if you can reach a certain IP or Hostname through a specific Interface: Ethernet (IPv4), Ethernet (IPv6), Wifi AP, Wifi Mesh and 3G/GPRS.

Ping

Select interface:

Destination Host:

```
Launching: ping 'www.google.com' -c 10 -I 'eth0'

PING www.l.google.com (173.194.34.240) from 192.168.1.210 eth0: 56(84) bytes of data.
64 bytes from mad01s09-in-f16.1e100.net (173.194.34.240): icmp_seq=1 ttl=54 time=50.2 ms
64 bytes from mad01s09-in-f16.1e100.net (173.194.34.240): icmp_seq=2 ttl=54 time=47.1 ms
64 bytes from mad01s09-in-f16.1e100.net (173.194.34.240): icmp_seq=3 ttl=54 time=44.7 ms
64 bytes from mad01s09-in-f16.1e100.net (173.194.34.240): icmp_seq=4 ttl=54 time=63.1 ms
64 bytes from mad01s09-in-f16.1e100.net (173.194.34.240): icmp_seq=5 ttl=54 time=45.9 ms
64 bytes from mad01s09-in-f16.1e100.net (173.194.34.240): icmp_seq=6 ttl=54 time=44.0 ms
64 bytes from mad01s09-in-f16.1e100.net (173.194.34.240): icmp_seq=7 ttl=54 time=47.0 ms
64 bytes from mad01s09-in-f16.1e100.net (173.194.34.240): icmp_seq=8 ttl=54 time=44.1 ms
64 bytes from mad01s09-in-f16.1e100.net (173.194.34.240): icmp_seq=9 ttl=54 time=184 ms
64 bytes from mad01s09-in-f16.1e100.net (173.194.34.240): icmp_seq=10 ttl=54 time=513 ms

--- www.l.google.com ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9020ms
rtt min/avg/max/mdev = 44.037/108.448/513.263/141.028 ms

ping finished.
```

Meshlium Xtreme can also perform this test over the Ethernet (IPv6) interface.

Ping

Select interface:

Destination Host:

```
Launching: ping6 '2001::20d:b9ff:fe26:b620' -c 10 -I eth0

PING 2001::20d:b9ff:fe26:b620(2001::20d:b9ff:fe26:b620) from 2001::20d:b9ff:fe26:b620 :
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=1 ttl=64 time=0.096 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=2 ttl=64 time=0.099 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=3 ttl=64 time=0.098 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=4 ttl=64 time=0.107 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=5 ttl=64 time=0.107 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=6 ttl=64 time=0.107 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=7 ttl=64 time=0.107 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=8 ttl=64 time=0.108 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=9 ttl=64 time=0.106 ms
64 bytes from 2001::20d:b9ff:fe26:b620: icmp_seq=10 ttl=64 time=0.109 ms

--- 2001::20d:b9ff:fe26:b620 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9002ms
rtt min/avg/max/mdev = 0.096/0.104/0.109/0.010 ms

ping finished.
```

20.2. Iperf

This tool let you know the real bandwidth between two different nodes through a specific Interface. Meshlium comes with the Iperf service activated by default so you can test the bandwidth quality of a link between two Meshlium just setting the IP of the node in the input box.

Iperf test

Select interface

Ethernet

Destination Host

192.168.1.131

Do Test

Launching: iperf -c '192.168.1.131' -r

Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)

Client connecting to 192.168.1.131, TCP port 5001
TCP window size: 26.9 KByte (default)

[17] local 192.168.1.7 port 55189 connected with 192.168.1.131 port 5001

[ID] Interval Transfer Bandwidth

[17] 0.0-10.0 sec 112 MBytes 94.1 Mbits/sec

iperf finished.

20.3. Traceroute

Another interesting tool to discover which is the route performed by the data which is being sent through a certain Interface while trying to reach a host.

Traceroute

Select interface

Destination Host

```
Launching: traceroute 'www.libelium.com' -i 'eth0'

traceroute to www.libelium.com (46.105.20.38), 30 hops max, 40 byte packets
 1 192.168.1.1 (192.168.1.1) 0.559 ms 0.324 ms 0.360 ms
 2 * * *
 3 90.165.99.1 (90.165.99.1) 83.327 ms 83.317 ms 83.249 ms
 4 10.255.44.29 (10.255.44.29) 83.208 ms 95.037 ms 98.598 ms
 5 62.36.202.93 (62.36.202.93) 102.872 ms 102.812 ms 102.745 ms
 6 (62.36.187.82) 102.678 ms 96.611 ms 99.920 ms
 7 81.52.186.193 (81.52.186.193) 103.700 ms 157.763 ms 115.890 ms
 8 mad-1-6k.es.eu (91.121.131.105) 115.601 ms * *
 9 mad-5-6k.es.eu (94.23.122.245) 115.402 ms * *
10 gsw-g1-a9.fr.eu (91.121.128.33) 122.474 ms 122.476 ms 113.616 ms
11 rbx-g1-a9.fr.eu (91.121.215.151) 124.810 ms rbx-g2-a9.fr.eu (91.121.131.213) 124
12 rbx-s10-6k.fr.eu (178.33.100.38) 122.667 ms * rbx-s10-6k.fr.eu (178.33.100.126)
13 vps16563.ovh.net (46.105.20.38) 139.130 ms 111.888 ms 114.135 ms

Traceroute finished.
```

Meshlium Xtreme can also perform this test over the Ethernet (IPv6) interface.

Traceroute

Select interface

Destination Host

20.4. Netstat

Discover which connections IPv4-Port (tcp), and IPv6-Port (tcp6) are active.

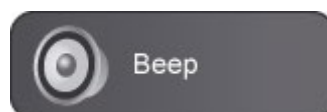
Netstat

Active Internet connections (servers and established) at: Mon, 07 May 12 12:53:31 +0000

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:5001	0.0.0.0:*	LISTEN	-
tcp	0	0	0.0.0.0:3306	0.0.0.0:*	LISTEN	-
tcp	0	0	0.0.0.0:111	0.0.0.0:*	LISTEN	-
tcp	0	0	192.168.1.210:80	192.168.1.150:43059	SYN_RECV	-
tcp	0	0	0.0.0.0:8080	0.0.0.0:*	LISTEN	-
tcp	0	0	0.0.0.0:53	0.0.0.0:*	LISTEN	-
tcp	0	0	0.0.0.0:2006	0.0.0.0:*	LISTEN	-
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	-
tcp	0	0	0.0.0.0:1723	0.0.0.0:*	LISTEN	-
tcp	0	0	192.168.1.210:22	192.168.1.150:48102	ESTABLISHED	-
tcp	0	0	192.168.1.210:22	192.168.1.150:48095	ESTABLISHED	-
tcp6	0	0	:::80	:::*	LISTEN	31743/netstat
tcp6	0	0	:::53	:::*	LISTEN	-
tcp6	0	0	:::22	:::*	LISTEN	-
tcp6	0	0	:::443	:::*	LISTEN	31743/netstat
tcp6	0	0	192.168.1.210:80	192.168.1.150:43058	ESTABLISHED	31743/netstat

20.5. Beep

When setting up and configuring several Meshlium at a time in the laboratory, can be difficult to distinguish between them (overall when the IP addresses are given by a external DHCP router). For this reason we have added a "Beep" button in the "Tools" section which will make the current Meshlium to make a short sound ("beep!").



When rebooting and initializing Meshlium it will also perform different "beeps" in order to give information about its state in each moment. Go to the section "Initialization, Restart and Shutdown" to know more.

21. Special options for Wifi networks

21.1. Long range links

If you want to make long range Wifi links (km) between several Meshlium you will need to adjust some parameters (Acktimeout, Ctstimout, Slottime) in the Wifi interfaces. We have enabled in the tools section the possibility of doing it automatically. Just set the number of Km of the link desired and Save. *

(*) Depending on the range, directional antennas must be used.

You can restore the factory defaults just pressing the "Restore Defaults" button.

Long range link

Select interface

Long range link configuration

Select input method

Distance (Km)

ACKTIMEOUT

CTSTIMEOUT

SLOTTIME

Restore Defaults

Save

21.2. Fresnel calculator

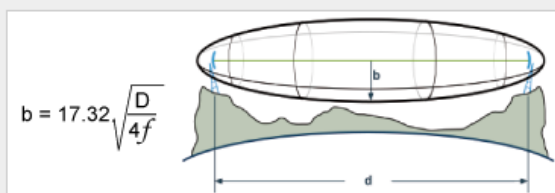
The Fresnel Zone is the space which should be empty of objects in a wireless transmission between two points to get the maximum throughput and transmission quality. Here you can find a tool in order to calculate when choosing the right points for your nodes.

Fresnel parameters

Distance (km)

Calculate

2.4 Ghz	5 Ghz
24.999 m	17.32 m



22. System

22.1. Hostname

Set the name of the node. By default is "Meshlium".

HOSTNAME

Meshlium's hostname

22.2. Users Manager

Here you can change two different passwords.

The "Users Manager" section contains the password needed to access to the Manager System. It is the same password used in the system to access by SSH. This means that when you change it you are also changing the system password for 'root'.

Users Manager

Name

root

Password

Confirm password

Note: the SSH password cannot contain special chars (e.g. \$, %, =). The SSH password can only be composed of **letters and numbers**. Be careful since Meshlium could be damaged with a not appropriate password (warranty would be void).

22.3. Setting the time

In order to store correctly in the File System and in the Local Data Base the data captured from the ZigBee and Bluetooth radios and from the GPS module is important to set the system time previously. The idea is each sensor data frame to be stored along with a time stamp so that they can be chronologically ordered.

Time synchronization

Mon Jun 1 12:57:36 CEST 2015

Date and hour for meshlium

Year

Month

Day

Hour

Minute

Time Zone

The system does not perform any time synchronization with a time server on the Internet (NTP) by default. This service would need Internet connection and could potentially spend unwanted data traffic, affecting a limited data plan.

A time synchronization can be manually done accessing Meshlium SSH console and running the command (using the ntp.org or any other time server):

```
ntpdate pool.ntp.org
```

The NTP service is installed in Meshlium too, so you can perform a continuous synchronization. Be careful as this option could spend data in the Internet connection and potentially waste a limited data plan. Activate it with the command:

```
ntpd
```

Meshlium also has a hardware Real Time Clock (RTC) that can be used to set the time. To set the time to the RTC in the SSH console, execute this (it is recommended perform a NTP synchronization before setting the time to RTC):

```
hwclock --systohc
```

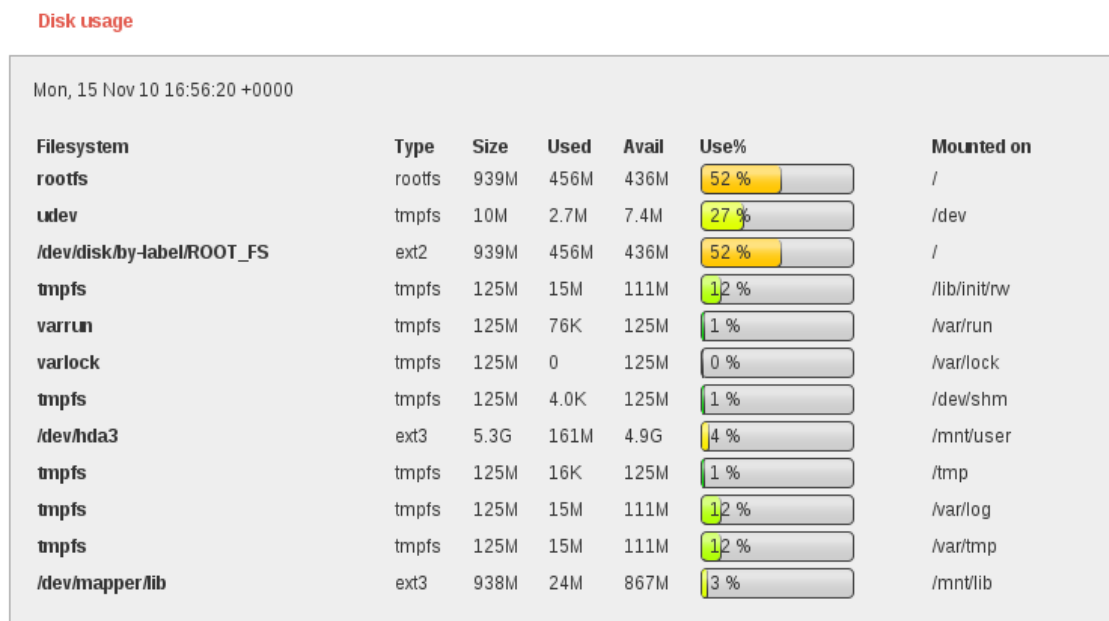
Then, you can set the system time using the RTC anytime with the command:

```
hwclock --hctosys
```

22.4. Disk Usage

Graphical board to see the amount of disk being used in each partition.

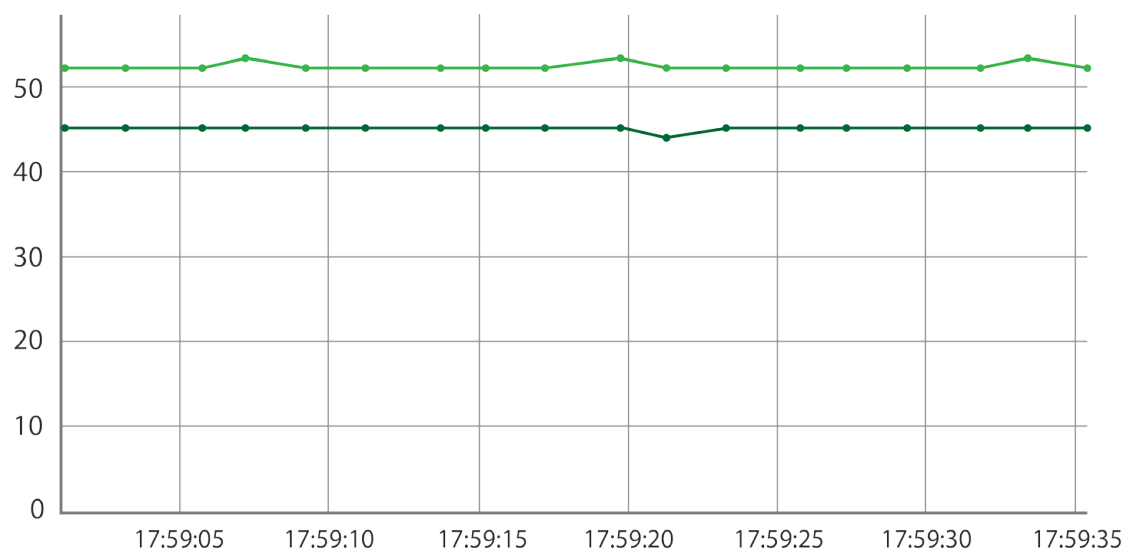
The most important partition is /dev/hda3 which is mounted in the "/mnt/user folder". All the data which is captured (ZigBee, Bluetooth, GPS) is stored in this partition.



22.5. Internal temperature sensors

You can see in real time the microprocessor (upper) and mother board temperature in Celsius degrees.

Meshlium temperature sensors graph



23. Data Base Management

23.1. MySQL Direct Access

In order to access to the Meshlium Data Base from an external application you have to use the next parameters:

- **IP Wifi:** 10.10.10.1
- **IP Ethernet:** Depending on your DHCP server (You can specify a static IP in the Interfaces section). 192.168.1.100 (if there is no DHCP server)
- **Database:** MeshliumDB
- **Table:** Depending on the data to be extracted. options are: zigbeeData, bluetoothData, gpsData.
- **Port:** 3306
- **User:** root
- **Password:** libelium2007

Note: Password of the local MySQL Database can not be changed. An external Data Base may be used to store the frames with a new password. Go to the External Database tab for more information."

You can also access directly the Meshlium Data Base using the SSH connection.

Via Wifi:

```
$ ssh root@10.10.10.1
```

Via Ethernet:

```
$ ssh root@ip_given_by_dhcp
```

Via Ethernet (if there is no DHCP server):

```
$ ssh root@192.168.1.100
```

- **user:** root
- **password:** libelium

Now execute you can enter the MySQL shell (password: libelium2007)

```
meshlium:$ mysql -u root -p
>libelium2007

mysql> use MeshliumDB;
Database changed
mysql> select * from XXX order by YYY;
```

or, extract the results of a query directly to an external file without entering to this shell:

```
meshlium:$ echo "select * from XXX order by YYY;" | mysql -u
root --password="libelium2007" MeshliumDB > /PATH/FILE
```

The Data Base files can be found at "/mnt/lib/mysql/MeshliumDB/".

23.2. PHPMYAdmin

Also you can manage all mysql databases in Meshlium using “phpmyadmin”. Go to the phpMyAdmin plugin in the Tools section and work directly there or open it in a new browser window.

You can also access with at: <http://10.10.10.1/phpmyadmin>. If you are not accessing via Wifi or you changed this interface the IP may change.

Using **root** as user, **libelium2007** as password.

Note: Password of the local MySQL Database can not be changed. An external Data Base may be used to store the frames with a new password. Go to the External Database tab for more information.”



24. FTP Management

As of version 3.0.6 or later an FTP sever has been added to Meshlium.

FTP is built on a client-server architecture and uses separate control and data connections between the client and the server. FTP users may authenticate themselves using a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that hides (encrypts) the username and password, and encrypts the content, FTP is often secured with SSL/TLS ("FTPS").

Meshlium has **ProFTPD** installed as FTP server software.

The following sections explain how to configure the FTP server to receive data from Waspnote 3G/GPRS board. If you wish that all data received from Waspnote 3G/GPRS board can be displayed on ManagerSystem, use this path:

```
/mnt/user/camera
```

24.1. Creating user

Before creating the user, it is necessary create a directory to stored the received data:

```
sudo mkdir /mnt/user/ftp
```

After creating a new folder to store data, you have to create a user with false shell, because this type of user don't need a valid shell (more secure), therefore select /bin/false shell for user and /mnt/user/ftp as directory:

```
sudo useradd -d /mnt/user/ftp -s /bin/false user
```

Modify folder permissions:

```
sudo chown -R user /mnt/user/ftp
sudo chmod -R 755 /mnt/user/ftp
```

Set the user password:

```
sudo passwd user
```

24.2. Setting permission

To configure **ProFTPD** to only allow access to the user created, edit the file /etc/proftpd/proftpd.conf, for example with **vim**:

```
remountrw
vim /etc/proftpd/proftpd.conf
```

And add the following to the end file:

```
<Limit LOGIN>
  AllowUser user
  DenyAll
</Limit>
RequireValidShell off
```

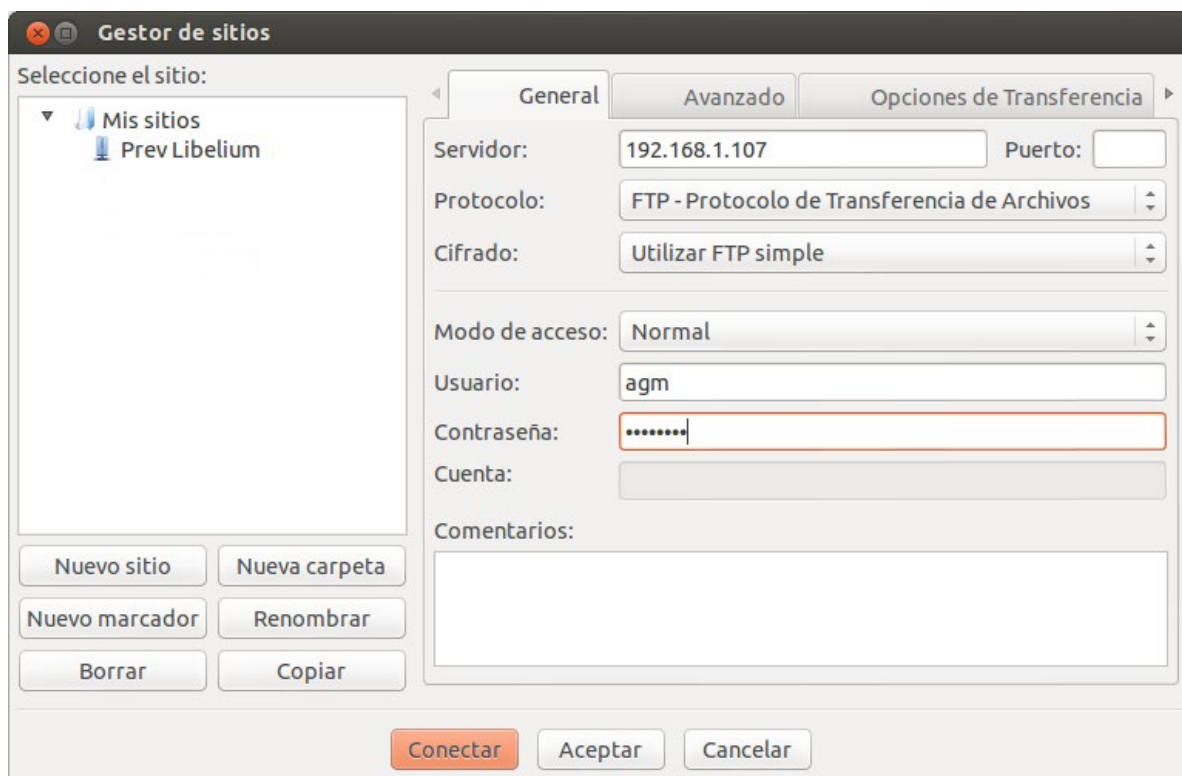
The last line that we have introduced is to be able to connect to FTP users with false shell.

24.3. Accessing FTP server

To access Meshlium FTP server from any computer, you can use FTP client like **FileZilla**.

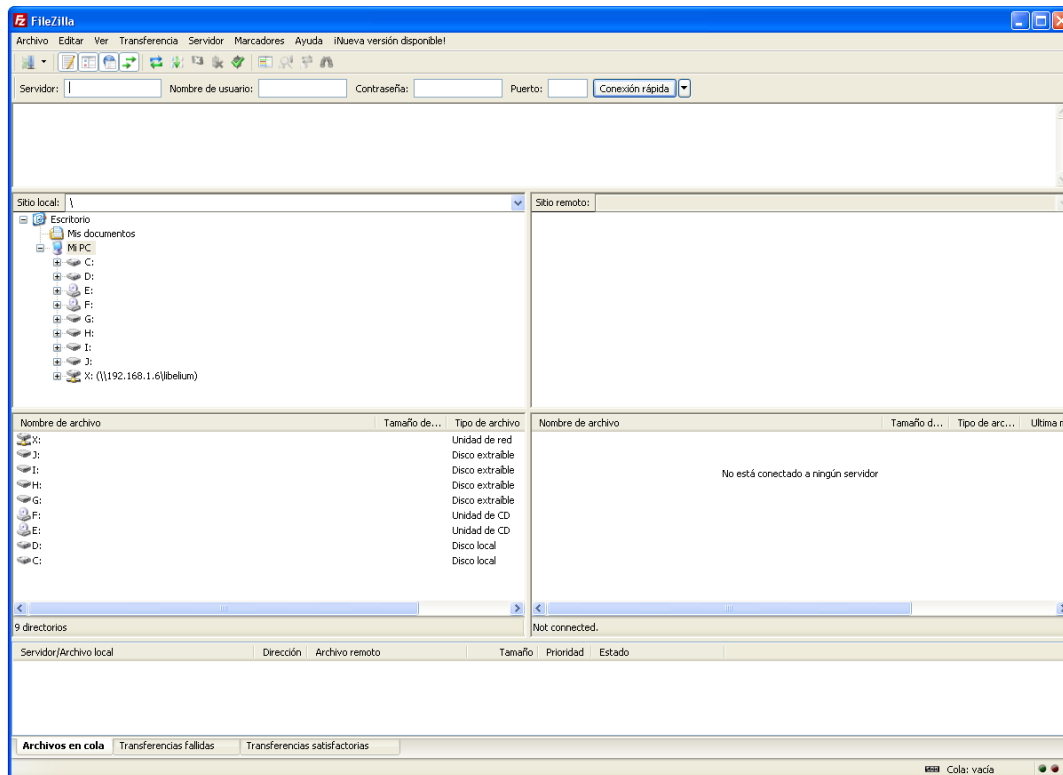
To connect to FTP server you must enter:

- Meshlium IP
- Username
- Password



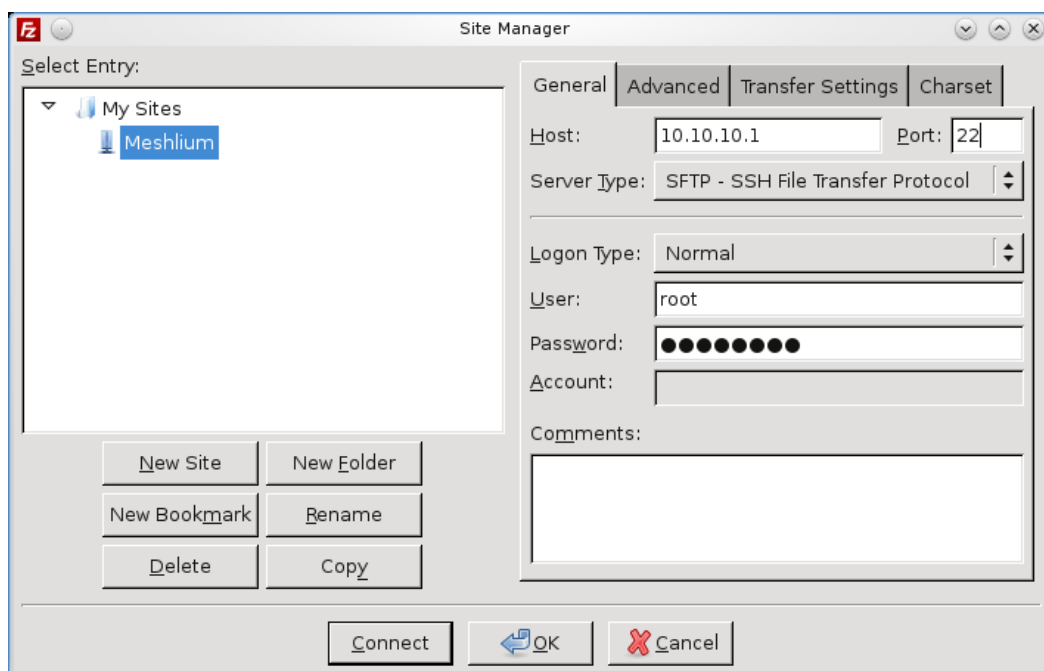
Once connected to the FTP server the user can perform operations like download, delete, or modify storage data.

Alternatively you can use a sftp client program, like Filezilla, and use it from you pc to upload files to Meshlium:



Using the following data:

- **Host:** 10.10.10.1
- **Server Type:** SFTP
- **Port:** 22
- **User:** root
- **Password:** libelium



26. Installing new packages and libraries

In order to install new applications you must use **aptitude**. Any other tool including apt-get could set the system unstable as the packet dependencies are not always perfectly solved using other tools.

Important:

- *Never use "apt-get", use always "aptitude".*
- *Never perform an "upgrade" (not even using "aptitude").*
- *Before installing new packages make an "update".*
- *Installing new software is a delicate task which has to be done for experts. If you are not sure about using "aptitude" do not perform any action as the system could get damaged. You can ask to our Developing team at: <http://www.libelium.com/forum>*

First of all you must remount the system in read and write mode. To do so execute as root: "remount rw".

```
$ remount rw
```

Now you can execute "aptitude":

```
$ aptitude
```

1. Perform an "update" pressing the "u" key
2. Search for your package in the input form that appears when pressing the "/" key
3. Select the desired package and press the "+" key
4. Check that there are no dependencies to solve
5. If there are dependencies you should fix them first (read the aptitude manual and tutorials on the web)
6. Once dependencies are solved press the "g" key and the new package will be download and installed.

If you are not sure about using "aptitude" do not perform any action as the system could get damaged. You can ask to our Developing team at: <http://www.libelium.com/forum>

27. Upgrading old versions of Meshlium

Note: The Update functionality is only for versions up from 3.0.0 if you have a older version contact the Commercial Department.

To see your current version go to the “Help” section.

ABOUT

About

Meshlium Manager System is the Open Source router web manager designed by [Libelium](#).
This web application is optimized to get the most of the [Meshlium](#) router.
The code is released under the terms of the [GPL2 License](#).

Version

Meshlium Manager System v.3.0.0
Serial number: 12345678901234

Documentation and resources

[Documentation](#)
[Resources for developers](#)

In the “Update Manager” section you can upgrade the entire Manager System or a certain plugin individually. There are three ways of getting the new version installed.

- Checking the Libelium Repository
- Local File
- URL

The screenshot displays the Meshlium Manager System web interface. The top navigation bar includes the title 'Meshlium Manager System' and 'Meshlium Zigbee Mesh 3G AP'. The main menu contains icons for Interfaces, Sensor Networks, Tools, System, Update Manager, and Help. The 'Update Manager' section is active, showing the 'Upload Update' form. The current version is 3.0.0. The form has two radio buttons: 'Local file' and 'Url'. The 'Local file' option is selected, and there is a 'Select' button next to it. The 'Url' option is also available with a 'Download' button. Below the form, there is a 'Check for Updates' section with a 'Check for updates' button. The footer of the interface shows the copyright notice: '© Libelium Comunicaciones Distribuidas S.L. | Terms of use'.

27.1. Checking the Libelium Repository

In order to use this version you need to have **Meshlium connection to the Internet**. If so just press the “Check for updates” button and the system will connect with the Libelium server and will download the latest updates.

Upload Update

Current version: 3.0.0

☐ Local file:

☐ Uri:

Check for Updates

Current version: 3.0.0

Up 3.0.0 to 3.0.1

Security improved.

All the available updates are shown. It is possible that several “middle step” updates have to be done before getting the latest version installed.

After installing all the pending updates reboot the system pressing the “Restart” button.

Remember that after upgrading the system you will have to access to the system as specified in the “Accessing to Meshlium” chapter.

27.2. Local File

You can also make installations in “offline” mode. To do so download first the desired package from the next URL:
<http://www.libelium.com/downloads/managersystem/updates/updates.xml>

Then select it from your hard disk using the “Select” button.

The Libelium Packages will have the extension:

- libupd - Manager System updates

Once uploaded the update package just press the “Install” button.

Considerations: You have to finish the installation of a previous package in order to install the next one. If you upload a new package without installing the previous one it will be removed.

27.3. URL

You can also make installations using a specific URL (www.libelium.com/downloads/managersystem/updates/updates.xml). To do so just write this URL into the right input box and press the "Download" button. Remember that in the same way as in the "Checking for updates" case, Meshlium need to have access to the Internet.

The Libelium Packages will have the extension:

- libupd - Manager System updates

Once uploaded the update package just press the "Install" button.

Important: Sometimes you need to upgrade first the Manager System in order to install.

Considerations:

- You have to finish the installation of a previous package in order to install the next one. If you download a new package without installing the previous one it will be removed.
- The Update functionality is only for versions up from 3.0.0
- In the Meshlium Development web section you can find the file "Changelog.txt" with all the changes and new features added version by version.

28. Rescue System

The Rescue System must be used just when Meshlium is completely inaccessible even after trying to access through a crossover Ethernet cable as specified in the section "How to use Meshlium".

Possible cases where the Rescue System is useful are:

- When the operate system has been corrupted when installing a new packet or upgrading the system
- The user has deleted or modified wrongly an important system file

Obviously this cases are not covered by the warranty as described in the section "Important: read me before using". For this reason we give the option to users of rescuing the system by themselves without having to send Meshlium to the technical service.

Important:

- *Executing the Rescue System will delete all the user information stored in Meshlium including sensor information stored in the internal database.*
- *Be careful when using "dd" in Linux or Windows. A mistake in the path of the USB disk could destroy the information contained in your own hard disk.*
- *The recover system may not work if the file system is severely damaged.*

28.1. Steps

You will need a USB pen drive of at least 8 GB.

The first thing to do is download from the website of Libelium the image file "meshliumrescue.img" needed to restore Meshlium to factory defaults.

Note: You need to contact first our Technical Service Department in order to get the user and password and URL to download the image.

Go to: http://www.libelium.com/development/meshlium/technical_service

Once the file has downloaded, you must burn it in a USB pen drive.

- Linux

Important: Performs all operations as root.

1.- Uncompress rescue.tar.gz in your PC:

```
tar xvf rescue.tr.gz
```

2.- Format a pen drive as **ext2** in your PC, for example with Gparted.

3.- Copy all files of the rescue folder to the pen drive (directory changes depending on the user).

```
sudo cp rescue/* /media/"user"/"pendrive"
```

4.- Shutdown Meshlium (in the secure way).

5.- Plug rescue pendrive.

6.- Turn on Meshlium and wait about 15 minutes when the restoration is finished Meshlium will emit several beeps.

7.- Unplug rescue pendrive.

8.- Unplug and plug Ethernet wire to turn on Meshlium (this is the only case when the user must restart Meshlium like this).

28.2. How to Recover the file system

1. Unplug the power Ethernet cable in Meshlium.
2. Unscrew with the hand the right stopper (next to the Ethernet connector).
3. Plug the recovery USB pen drive to Meshlium. If your Meshlium has an external SIM socket, you will have to use an adapter between micro-USB and female USB.
4. Connect the Ethernet cable to Meshlium. After the initial short beep, the process will start and the system will try to recover the system from the USB. This can take 15 minutes or so. Wait enough time to ensure the process has finished.



5. After waiting enough time, reboot Meshlium by unplugging and plugging again the Ethernet cable (this method is only recommended in the recovery process).
6. If the process has been successful, Meshlium will start and get an IP by DHCP. If it's not able to get an IP, it will set the address 192.168.1.100.
7. Access the Manager System and go to presets page (in the top right corner, below shutdown). Apply again your preset and reset Meshlium again.
8. Once Meshlium has restarted, access again the manager System, and go to the Update Manager page. You will have to apply updates from 3.1.0 up to the latest update.
9. Once the updates are applied, you can now setup again your own configurations for all the interfaces and use again Meshlium normally.

29. USB Device Connectivity

The external USB connector lets you connect any USB device to Meshlium. The only limitation is that your device must be supported by a Linux system (obviously you can install its drivers through a repository or uploading the files directly).

In the next example we will connect a webcam and will capture several images which will be accessed from a web page. Obviously the process will vary depending on the camera or USB device we want to integrate.

Important: if you want to place outdoor the Meshlium with the external USB device you have to protect the USB cable in order to make it waterproof. See page 8 in the current manual to see how the Ethernet cable is protected.



Steps:

1. Plug the webcam to the USB port.
2. Wait 10 or 15 seconds.
3. Open prompt and connect Meshlium using `ssh` command.
4. Mount file system in read/write mode using `remount rw` command.
5. Execute `lsusb` command. Thus we will be able identify the device and check that it is well connected. In this example, it is the output:

```
Bus 001 Device 003: ID 0ac8:301b Z-Star Microelectronics Corp. ZC0301 Webcam
```

6. Update the packets list from the repository:

```
aptitude update
```

7. Install the module necessary for the webcam or USB device:

```
aptitude install gspca-modules
```

Considerations: In exceptional cases, can be necessary recompiling the module.

8. Install the "camserv" package:

```
aptitude install camserv
```

9. Create "webcam.html" in the directory "/var/www/" with the following content:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
  <meta name="author" content="libelium@libelium.com" />
  <title>Webcam - Test</title>
</head>
<body>
  
</body>
</html>
```

10. Mount file system on read only mode using "remountro" command. The pictures taken with the webcam can be found in *http://Meshlium_IP/webcam.html*

30. ManagerSystem Changelog

Update from 3.0 to 3.0.1:

- Security updates

Update from 3.0.1 to 3.0.2:

- Meshlium Scanner updates
- Manager System internal updates
- Security updates

Update from 3.0.2 to 3.0.3:

- WiFi Scanner restrictions deleted
- Bluetooth Scanner RSSI problem fixed

Update from 3.0.3 to 3.0.4:

- Encryption plugging added
- Encryption database created

Update from 3.0.4 to 3.0.5:

- New Sensor Parser daemon
- New syncDB daemon
- Deleted Local File stored on Sensor Networks Capturer section
- New section Logs on Sensor Networks plugin
- New section Sensor List on Sensor Networks plugin

Update from 3.0.5 to 3.0.6:

- FTP server installed
- Update SensorCapturer, now captured and stored photos / videos
- New plugin Photo/Video created on Sensor Network section

Update from 3.0.6 to 3.0.7:

- OTA via FTP plugin added for 3G, GPRS and WiFi

Update from 3.0.7 to 3.0.8:

- Now iPhone 5 is detected with WiFi Scan plugin

Update from 3.0.8 to 3.0.9:

- SensorParser: fixed bug and improved performance

Update from 3.0.9 to 3.0.10:

- New cloud connector section
- After the update, the system will reboot. If not start, unplug and plug Meshlium.
- Note: Cloud Platform need sensor data in sensorParser database

Update from 3.1.0 to 3.1.1:

- Wifiscan bug fixes

Update from 3.1.1 to 3.1.2:

- Internal changes for Bluetooth

Update from 3.1.1 to 3.1.3:

- Fixed RSSI parameter missing and improved frames parser in Bluetooth Scanner for Meshliums Scanner v3.1.2.
- In Manager System "Show me now" tabs removed from WiFi Scanner and Bluetooth Scanner sections. Instead of this tab, the "Show data" button in Local DataBase tab should be used.
- NOTE: Password of the local MySQL database cannot be changed. An external database may be used to store the frames with a new password. Go to the External Database tab for more information.

Update from 3.1.3 to 3.1.4:

- WiFi scanner improvement for Meshlium Scanner.
- In WiFi Scanner section inside Manager System, you can choose the scanning time from a drop-down box. We recommend 40 seconds, value by default from now on.
- Bluetooth Scanner improvement for Meshlium Scanner.
- In Bluetooth Scanner section inside Manager System, you can choose the scanning type from a drop-down box. Indoor scans for 10 seconds and get devices names and Outdoor scans for 40 seconds without devices names. Indoor value by default from now on.
- New HTTP parser to receive frames through GPRS, 3G, WiFi or Ethernet interface. Include capturer logs and sensor list shown to users in Manager System.
- Rescue System was improved.
- NOTE: The password of the local MySQL Database can not be changed. An external Database may be used to store frames with a new password. Go to the External Database tab for more information.

Update from 3.1.4 to 3.1.5:

- NEW: Sentilo synchronization: You can now send the sensor data received from Waspote to your Sentilo platform. You can set the parameters in the "cloud connector" page in Manager System.
- Sensor list updated: Smart Water sensors added.
- Removed option to disable local database storing. Sensor parser always stores data in the local database.
- Memory resources optimization for Scanner services.
- NOTE: Password of the local MySQL Database can not be changed. An external Data Base may be used to store the frames with a new password. If external database synchronization was enabled before upgrading to 3.1.4, this command has to be run in the destination database (changing the table name if needed): "alter table sensorParser add parser_type tinyint(3) NOT NULL default '0';". Go to the External Database tab for more information.

Update from 3.1.5 to 3.1.6:

- Sensor list updated.
- Improved sensorParser stability.
- Memory optimization in external DB synchronization.
- MQTT client improved. Added Waspote ID in the MQTT message.
- Improved configuration of radio modules from Manager System.
- Minor bug corrections in Manager System.
- NOTE: Password of the local MySQL Database can not be changed. An external Data Base may be used to store the frames with a new password.
- NOTE 2: This update will revert to default time settings in Meshlium Scanner.

Update from 3.1.6 to 3.1.7:

- New LoRa module included.
- Improvements in sensorParser. Binary and cyphered frames has size information in bytes.
- Encryption options simplified and decode process integrated in sensorParser.
- MQTT synchronization time can be set in a file.
- Logs now moved to /mnt/user/logs.
- NOTE: Password of the local MySQL Database can not be changed. An external Data Base may be used to store the frames with a new password.
- NOTE 2: Due to changes in binary frame (size in bytes) in order to properly receive these frames with this version is required to use the latest version of Wasmote API (v12 or above).

Update from 3.1.7 to 3.1.8:

- Telefonica DCA cloud synchornization.
- Microsoft Azure Service Bus synchronization.
- Minor bug corrections.
- NOTE: Password of the local MySQL Database can not be changed. An external Data Base may be used to store the frames with a new password.

Update from 3.1.8 to 3.1.9:

- 3G connection setup simplified and connection process improved.
- GPS data reading simplified and performance improved.
- Sensor list updated with the new Gases PRO line sensors.
- Network configuration process revised to avoid setup file corruption.
- NOTE: Password of the local MySQL Database cannot be changed. An external Data Base may be used to store the frames with a new password.

Update from 3.1.9 to 3.2.0:

- Sensor list updated with the new Smart Water Ions line sensors.
- Added the Time Zone field in Time Synchronization.
- MQTT server was corrected and updated.
- NOTE: Password of the local MySQL Database cannot be changed. An external Data Base may be used to store the frames with a new password.

Update from 3.2.0 to 3.2.1:

- Added IoTSENS cloud connector.
- Scanner vendor list updated.
- Added option for MAC randomization in scanner.
- NOTE: Password of the local MySQL Database cannot be changed. An external Data Base may be used to store the frames with a new password.
- NOTE2: TX power of AP is fixed to "auto" due to a bug detected in the drivers. We will enable a way to set the power in future releases.

Update from 3.2.1 to 3.2.2

- ThingWorx cloud connector renewed
- Added Devicify cloud connector
- Added SolvView cloud connector
- Added ThingPlus cloud connector
- Added ElementBlue cloud connector
- Bug correction in Scanner
- NOTE: This update needs to download some components at install time. Internet connection is required through the install process.
- NOTE: Password of the local MySQL Database can not be changed. An external Data Base may be used to store the frames with a new password.
- NOTE: TX power of AP is fixed to "auto" due to a bug detected in the drivers.

Update from 3.2.2 to 3.2.3

- Added IBM Bluemix cloud connector
- Added Microsoft Azure Event Hubs cloud connector
- Added DeviceLynk cloud connector
- Added B-Scada cloud connector
- WiFi Scan service renewed
- MQTT service renewed

Update from 3.2.3 to 3.2.4

- Added Sofia2 cloud connector
- New updates in IBM Bluemix cloud connector
- New updates in Telefónica IoT cloud connector
- New updates in Esri cloud connector
- New updates in ThingPlus cloud connector
- New updates in MQTT connector
- Cloud connector section renewed
- Sensor list updated
- NOTE: Password of the local MySQL Database can not be changed. An external Data Base may be used to store the frames with a new password.
- NOTE: TX power of AP is fixed to "auto" due to a bug detected in the drivers.

Update from 3.2.4 to 3.2.5

- New updates in IBM Bluemix cloud connector
- New updates in Esri cloud connector
- Cloud connector configuration section renewed
- NOTE: Password of the local MySQL Database can not be changed. An external Data Base may be used to store the frames with a new password.
- NOTE: TX power of AP is fixed to "auto" due to a bug detected in the drivers.
- NOTE: Meshlium Visualizer map section improved, only for Meshliums of IoT Vertical Kits.

Update from 3.2.5 to 3.2.6

- Add Cloud Connectors: Amazon IoT, Amplía's OpenGate, eagle.io, Extunda, IoT-Ticket, Simfony, SmartCityPlatform
- Fix MQTT sync bug and several minor bug fixes
- NOTE: Password of the local MySQL Database can not be changed. An external Data Base may be used to store the frames with a new password.
- NOTE: TX power of AP is fixed to "auto" due to a bug detected in the drivers

31. Documentation Changelog

From v5.9 to v6.0:

- Added documentation for 7 new cloud services: Amazon IoT, Amplía's OpenGate, eagle.io, Extunda, IoT-Ticket, Simfony, SmartCityPlatform.

From v5.8 to v5.9:

- Added documentation for new cloud services: Sofia2 and Esri.
- Updated information for IBM Bluemix and Telefónica IoT Platform cloud services.
- New distribution for Cloud Connector section and new images updated according to the renewed section.
- Note added in section "Sending frames from Meshlium to Waspnote".
- Note added in section "Meshlium Visualizer" and "Understanding Meshlium".
- Meshlium Visualizer map section improved, only for Meshliums of IoT Vertical Kits.

From v5.6 to v5.7:

- Update done in a table about the LoRa radio module.

From v5.5 to v5.6:

- Added documentation for new cloud services: IBM Bluemix, Microsoft Azure Event Hubs, DeviceLynk and B-Scada.
- Updated sections: WiFi Scan and MQTT.

From v5.4 to v5.5:

- Added clarifications about detecting devices with Meshlium Scanner.

From v5.3 to v5.4:

- ThingWorx documentation updated.
- Added documentation for new cloud services: Devicify, ElementBlue, SolvView and ThingPlus.

From v5.2 to v5.3:

- Axeda is removed from Cloud Connector. No longer compatibility because of changes on Axeda service.

From v5.1 to v5.2:

- Time synchronization with RTC and NTP explained.
- Added IoTSENS cloud documentation.
- Scanner documentation updated to add new feature, MAC anonymization.
- Added chapter for Meshlium Visualizer.

From v5.0 to v5.1:

- Time Synchronization now accepts a field for Time Zone.
- Small changes were added in the Manager System Changelog chapter.

From v4.9 to v5.0:

- Changed 3G connection section.
- Changed GPS section.
- Specific instructions for the reboot and shut-down processes.
- Added the directional antennas size and weight.

From v4.8 to v4.9:

- Added Telefonica DCA section.
- Added Microsoft Azure Service Bus section.

From v4.7 to v4.8:

- Added new LoRa module.
- Encryption section simplified.
- Changed presets section due to new naming.
- Camera section deleted.
- 16/32 GB storage options were deleted.

From v4.6 to v4.7:

- Improvements in section "Smartphone detection".
- Added new directional antenna pack for Meshlium Scanner.
- Added section "HTTP parser" (GPRS / 3G / WiFi).
- Improved Meshlium diagram to include receiving via HTTP.

From v4.5 to v4.6:

- Deleted references to Show me now! feature in WiFi and Bluetooth scans.
- Added advice about the password of the local MySQL Database.

From v4.4 to v4.5:

- Added section for the Cloud Connector.
- Added section for the new External SIM socket.

From v4.3 to v4.4:

- iPhone 5 bug fixed.

From v4.2 to v4.3:

- Details about special chars in the SSH password.

From v4.1 to v4.2:

- Added section "OTA via FTP".
- iPhone 5 WiFi Scanner warning.
- Bluetooth Scanner distance changed.
- New explanation for Rescue System.

From v4.0 to v4.1:

- Added section "Capturing and Storing photos and videos".
- Added section "FTP Management".

32. Certifications

32.1. CE



In compliance with the 1999/05/EC directive, Libelium Comunicaciones Distribuidas declares that Meshlium complies with the following norms:

- EN 55022:1998
- EN 55022:1998/A1:2000
- EN 55022:1998/A2:2003
- EN 61000-4-2:1995
- EN 61000-4-2/A1:1998
- EN 61000-4-2/A2:2001
- EN 61000-4-3:2006
- EN 61000-4-4:2004
- EN 61000-4-6:1996
- EN 61000-4-6/A1:2001
- UNE-EN 60950-1:2007
- ETSI EN 301 489-1 V1.6.1 , EN 300 328
- M528785W-EO

Date: March 30th, 2009

You can ask for the product's Declaration of Conformity through the contact section at: <http://www.libelium.com>

Meshlium is equipment defined as a wireless communication device that offers:

- Short and long-distance data, voice and image communication.
- Wireless access to electronic communication networks and wireless connection on local networks between computers and/or terminals and peripheral devices.
- Geolocation .
- Joining cable networks to wireless networks at different frequencies.
- Joining wireless networks at different frequencies to each other.
- Output of information obtained from wireless sensor networks.
- To act as a data storage station .
- To capture information from the surrounding environment by means of connected interfaces, peripherals and sensors.
- To interact with the surrounding environment by activating/deactivating electronic mechanisms (both analogue and digital).

Characteristics of use of the equipment:

- Equipment declared a non-mobile device.
- Equipment prepared for handling by qualified personnel only.
- Equipment for installation in zones of restricted access for qualified personnel only.
- The configuration of additional modules, antenna and other accessories, must be carried out by qualified personnel only.

Limitations of Use:

- The Bluetooth modules have a variable transmission power of 13-17dBm. They fulfil standard IEEE 802.15.1 - Bluetooth 1.2.
- The low power Wifi module has a transmission power of 18dBm for 2.4GHz and 16dBm for 5GHz, which can be regulated using the configuration software. It fulfils standards IEEE 802.11a/b/g.
- The high power 5GHz Wifi module has a transmission power of 28dBm, which can be regulated using the configuration software. It fulfils standard IEEE 802.11a.
- the ZigBee/IEEE 802.15.4 module has a maximum transmission power of 20dBm. It is regulated by EN 301 489-1 v 1.4.1 (2002-04) and EN 301 489-17 V1.2.1 (2002 - 08). It must be limited using the configuration software to a maximum power of 12'11dBm (PL=0).
- The XBee 868MHz module has a maximum transmission power of 27dBm. This module is regulated for its use in Europe only.
- The XBee 900MHz module has a maximum transmission power of 20dBm. This module is regulated for its use in the US only.
- The GSM/3G/GPRS module has a powerh of 2W(Class 4) for the 850MHz/900MHz band and 1W(Class 1) for the 1800MHz and 1900MHz frequency band.
- **Note:** *The 850MHz band is not allowed in Spain. For further information about frequency and power restrictions contact the official regulation office in your country.*
- The pigtail used to connect the radio module to the antenna's connector introduces a loss of approximately 1dB for the 5GHz module and 0.25dB for 2.4GHz, 868MHz, 900MHz and GSM/3G/GPRS.
- The broadcasting power at which the Wifi, XBee 2.4GHz, XBee 868MHz, XBee 900MHz modules function can be limited using the configuration software. It is the installer's responsibility to choose the correct power in each case, taking into account the following limitations:
 - The broadcasting power of any of the modules added to that of the antenna minus the loss introduced by the pigtail and cable joining the connector to the antenna (in the event of having an extra connection cable) must never exceed 20dBm (100mW) in the 2.4GHz frequency band, 30dBm in the 5470MHz-5725MHz band, 23dBm in the 5150MHz-5350MHz band, and 27dBm in the 868MHz band following ETSI/EU regulations.
 - It is the installer's responsibility to correctly configure the equipment's various parameters, whether hardware or software, in order to ensure compliance with the regulations of each country where it is used.

Specific limitations for the 2.4GHz band:

- In Belgium it is authorised for use outdoors only on channels 11(2462MHz), 12(2467MHz) and 13(2472MHz). It can be used without a licence if for private use over a distance of less than 300m. For a greater distance or public use, an IBPT license is required.
- In France, its use is restricted to channels 10 (2457MHz), 11(2462MHz), 12(2467MHz), and 13(2472MHz). A license is required for both indoor and outdoor use. Please contact ARCEP (<http://www.arcep.fr>) for further information.
- In Germany, a license is necessary for its use outdoors.
- In Italy, a license is necessary for its use indoors. Outdoor use is not permitted.
- In Holland, a license is necessary for its use outdoors.
- In Norway, use is prohibited near Ny-Alesund in Svalbard. For further information visit Norway Posts and Telecommunications (<http://www.npt.no>).

Specific limitations for the 868MHz band:

- In Italy the maximum transmission power is 14dBm.
- In the Slovak Republic the maximum transmission power is 10dBm.

Important:

Libelium Comunicaciones Distribuidas S.L does not list the entire set of standards that must be met for each country. Libelium customers assume full responsibility for learning and meeting the required guidelines for each country in their distribution market. For more information relating to European compliance refer to the following web sites:

- CEPT ERC 70-03E - Technical Requirements, European restrictions and general requirements: <http://www.erodocdb.dk/>
- R&TTE Directive - Equipment requirements, placement on market: <http://www.erodocdb.dk/>

32.2. FCC



Meshlium models:

Meshlium AP Contains:

- FCC ID: NKRCM9
- FCC ID: Q87-USBBT100V2

Meshlium 3G/GPRS-AP Contains:

- FCC ID: NKRCM9
- FCC ID: Q87-USBBT100V2
- FCC ID: UDV-0200901181058

Meshlium Mesh-AP Contains:

- FCC ID: NKRCM9
- FCC ID: Q87-USBBT100V2

Meshlium Mesh-AP-GW Contains:

- FCC ID: NKRCM9
- FCC ID: Q87-USBBT100V2

Meshlium Mesh-3G/GPRS-AP Contains:

- FCC ID: NKRCM9
- FCC ID: Q87-USBBT100V2
- FCC ID: UDV-0200901181058

Meshlium RF-AP Contains:

- FCC ID: NKRCM9
- FCC ID: Q87-USBBT100V2
- FCC ID: OUR-XBEEPRO
- FCC ID: MCQ-XBEEPRO2
- FCC ID: MCQ-XBEE09P

Meshlium RF-3G/GPRS-AP Contains:

- FCC ID: NKRCM9
- FCC ID: Q87-USBBT100V2
- FCC ID: UDV-0200901181058
- FCC ID: OUR-XBEEPRO
- FCC ID: MCQ-XBEEPRO2
- FCC ID: MCQ-XBEE09P

Meshlium RF-Mesh-AP Contains:

- FCC ID: NKRCM9
- FCC ID: Q87-USBBT100V2
- FCC ID: OUR-XBEEPRO
- FCC ID: MCQ-XBEEPRO2
- FCC ID: MCQ-XBEE09P

Meshlium RF-Mesh-AP-GW Contains:

- FCC ID: NKRCM9
- FCC ID: Q87-USBBT100V2
- FCC ID: OUR-XBEEPRO
- FCC ID: MCQ-XBEEPRO2
- FCC ID: MCQ-XBEE09P

Meshlium RF-Mesh-3G/GPRS-AP Contains:

- FCC ID: NKRCM9
- FCC ID: Q87-USBBT100V2
- FCC ID: UDV-0200901181058
- FCC ID: OUR-XBEEPRO
- FCC ID: MCQ-XBEEPRO2
- FCC ID: MCQ-XBEE09P

Meshlium Scanner AP

- FCC ID: NKRCM9
- FCC ID: Q87-USBBT100V2
- FCC ID: QOQWT12

Meshlium Scanner 3G-AP

- FCC ID: NKRCM9
- FCC ID: Q87-USBBT100V2
- FCC ID: UDV-0200901181058
- FCC ID: OUR-XBEEPRO
- FCC ID: MCQ-XBEEPRO2
- FCC ID: MCQ-XBEE09P
- FCC ID: QOQWT12

Meshlium Scanner RF-AP

- FCC ID: NKRCM9
- FCC ID: Q87-USBBT100V2
- FCC ID: OUR-XBEEPRO
- FCC ID: MCQ-XBEEPRO2
- FCC ID: MCQ-XBEE09P
- FCC ID: QOQWT12

Installation and operation of any Meshlium model must assure a separation distance of 20 cm from all persons, to comply with RF exposure restrictions.

Module Grant Restrictions:**FCC ID OUR-XBEEPRO:**

The antenna(s) used for this transmitter must be installed to provide the separation distances, as described in this filing, and must not be co-located or operating in conjunction with any other antenna or transmitter. Grantee must coordinate with OEM integrators to ensure the end-users of products operating with this module are provided with operating instructions and installation requirements to satisfy RF exposure compliance. Separate approval is required for all other operating configurations, including portable configurations with respect to 2.1093 and different antenna configurations. Power listed is continuously variable from the value listed in this entry to 0.0095W

FCC ID MCQ-XBEEPRO2:

OEM integrators and End-Users must be provided with transmitter operation conditions for satisfying RF exposure compliance. The instruction manual furnished with the intentional radiator shall contain language in the installation instructions informing the operator and the installer of this responsibility. This grant is valid only when the device is sold to OEM integrators and the OEM integrators are instructed to ensure that the end user has no manual instructions to remove or install the device.

FCC ID UDV-0200901181058:

The antenna(s) used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be co-located or operating in conjunction with any other antenna or transmitter. OEM integrators must be provided with antenna installation instructions. OEM integrators and end-Users must be provided with transmitter operation conditions for satisfying RF exposure compliance.

33. Maintenance

- Although Meshlium is a highly resistant product, please handle with care in order to enjoy a longer useful life.
- Handle Meshlium with care, do not allow it to drop or move roughly.
- Avoid placing the devices in areas reaching high temperatures that could damage the electronic components.
- The antennas screw on gently to the connector, do not force upon installing or you could damage the connectors.
- Do not use any type of paint on the device, it could affect the operation of connections and closing mechanisms.
- Chargers must only be used indoors.
- Do not store Meshlium Lite in places exposed to dirt and dust in order to avoid damage to electronic components.
- Never open the casing, the guarantee will not cover products that have been opened.
- For cleaning, use a damp cloth, no aggressive chemical products.

34. Disposal and Recycling

- When Meshlium reaches the end of its useful life it must be taken to a recycling point for electronic equipment.
- The equipment should be disposed of separately from solid urban waste, please dispose of correctly.
- Your distributor will advise you on the most appropriate and environmentally-friendly way of disposing of the product and its packing.

