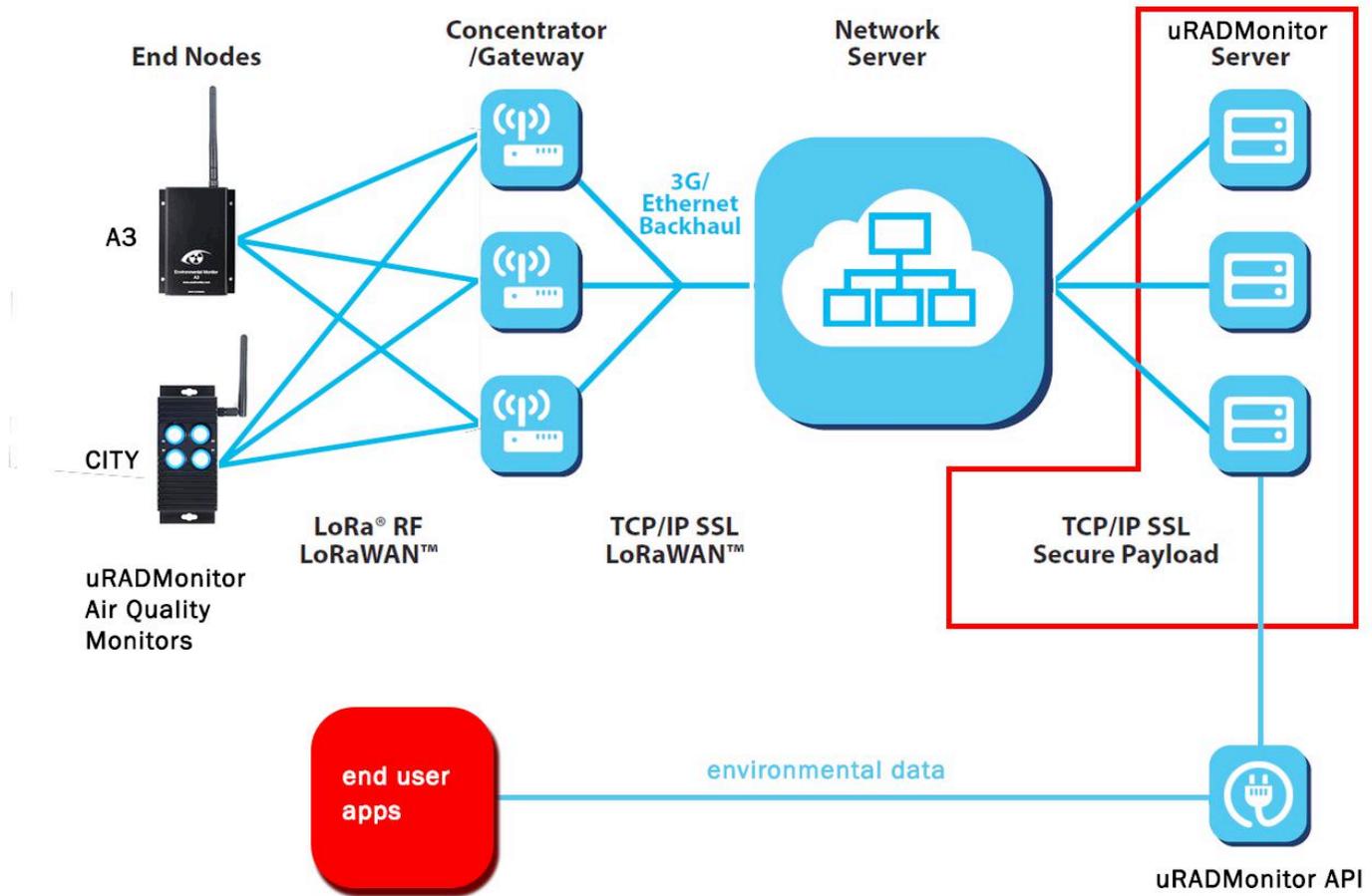


Network Server Data integration



Introduction

The innovative core of the uRADMonitor project was backed up by the newest technologies at its every design step. With a plenitude of connectivity options available for data export, considerable effort was invested to make sure the IOT uRADMonitor hardware will fit as many real life scenarios as possible. One is related to covering remote areas via long range radio links, while using only very little power. Here we designed uRADMonitor devices equipped with LoRaWAN wireless connectivity, so we can do air quality monitoring even in remote areas that have no Internet infrastructure.



Summary

1	What is LoRaWAN?	3
2	System architecture	3
3	Data export via callback mechanisms	3
4	Actility Network server	5
5	TTN Network server	6
6	Everynet platform	7
7	Loriot platform	9
8	Multitech Node-RED	9
9	Stream IoT-X	10
10	LoRaServer.io	11
11	Technical support	12

1. What is LoRaWAN?

LoRaWAN™ is a Low Power Wide Area Network (LPWAN) specification intended for wireless battery operated Things in a regional, national or global network. LoRaWAN targets key requirements of Internet of Things such as secure bi-directional communication, mobility and localization services. The LoRaWAN specification provides seamless interoperability among smart Things without the need of complex local installations and gives back the freedom to the user, developer, businesses enabling the roll out of Internet of Things.

2. System Architecture

LoRaWAN network architecture is typically laid out in a star-of-stars topology in which **gateways** is a transparent bridge relaying messages between **end-devices** and a central **network server** in the backend. Gateways are connected to the network server via standard IP connections while end-devices use single-hop wireless communication to one or many gateways.

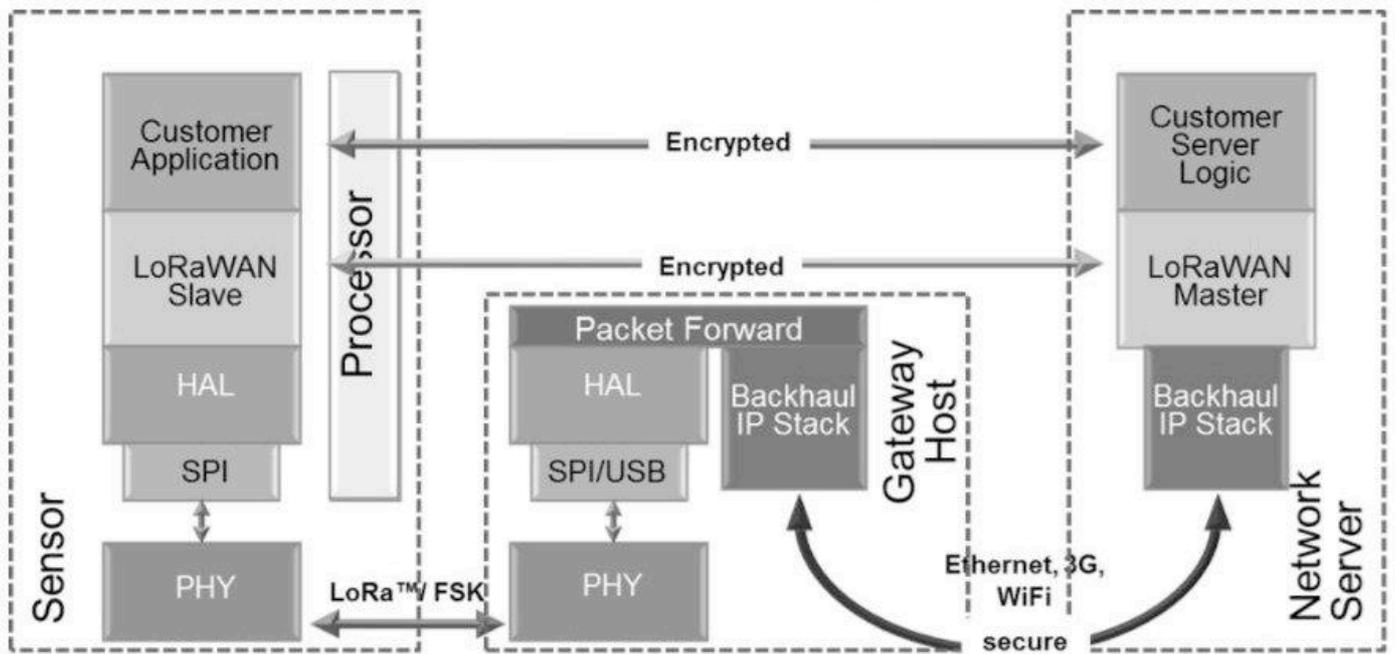


Figure 1. LoRaWAN system architecture, depicting the end-devices (sensor), the gateways and the network server

The uRADMonitor hardware (sensor) sends out encrypted environmental data, collected from the air quality sensors. LoRaWAN adds an additional encryption layer and the data is transparently handled over to the Network Server, by using one or more gateways in range of the uRADMonitor hardware.

3. Data export via callback mechanisms

The Network Server is a remote machine that receives the uRADMonitor encrypted data. The Network Server cannot access the uRADMonitor readings directly, its purpose is solely related to LoRaWAN retrieval of data and its functionality is limited to the data transmission protocol itself.

Network Server Data integration

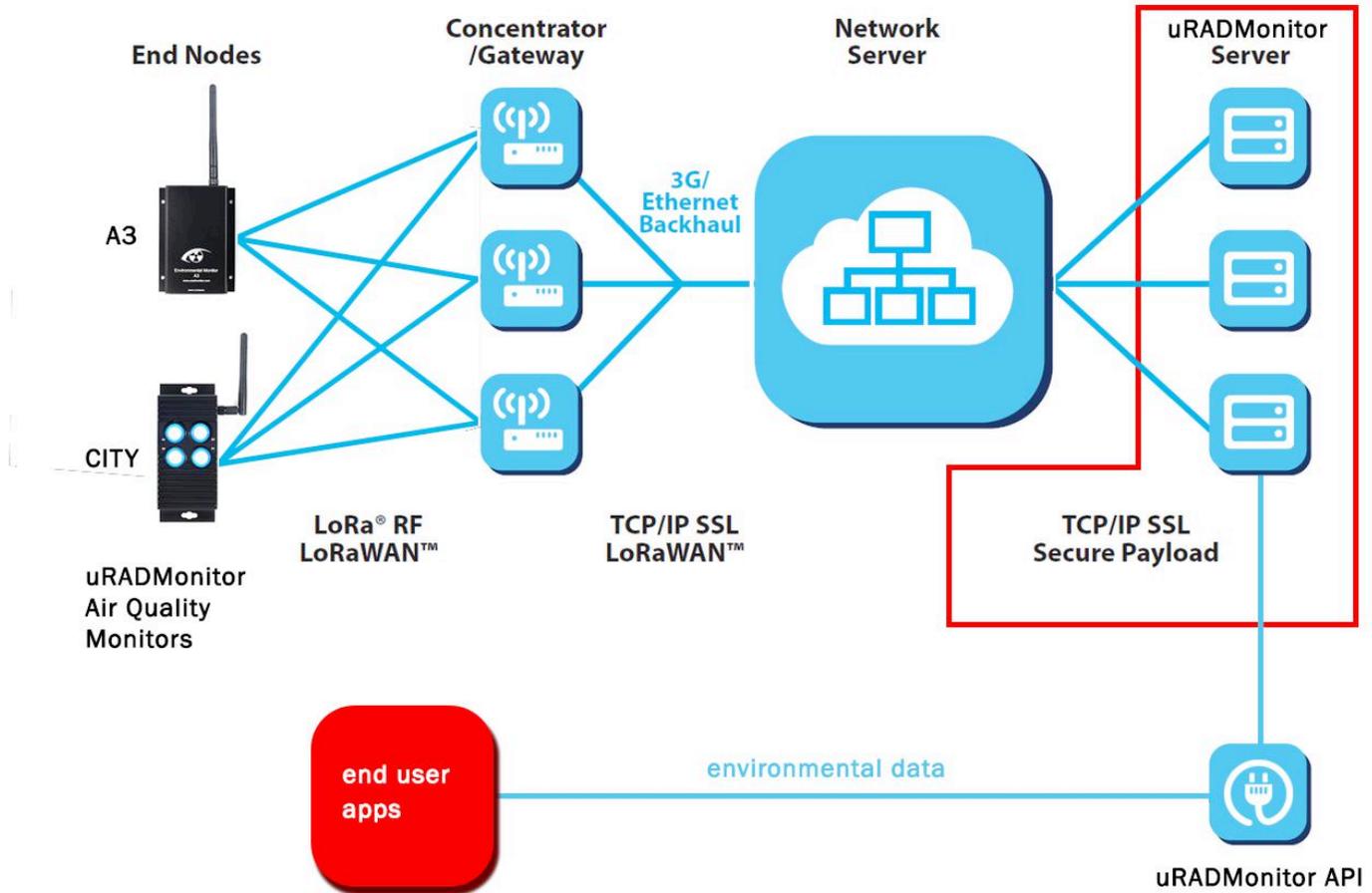


Figure 2. The uRADMonitor Server works as an Application Server

The Network Server on its own is not responsible for processing the data outside the LoRaWAN transmission specifications. The data is passed to the Application Server for further use. There are several mechanisms that can be involved, and one of them is the callback, where the Network Server calls a server resource on the Application Server via HTTP Post, to pass the data payload received over the LoRaWAN network from the uRADMonitor hardware devices.

In just a few steps the mechanism is as follows:

1. the uRADMonitor hardware reads the sensors, packs the data and encrypts it before sending it over the air
2. the LoRaWAN gateway picks up the radio transmission and handles it to the Network Server via its backhaul connection
3. the Network Server decodes the uRADMonitor data, to the same form it was at step 1, and passes it to the Application Server via the callback mechanism
4. the uRADMonitor Application Server receives the payload, decrypts the data to obtain the original uRADMonitor readings and adds them to the database.
5. the Air Quality data is available via the uRADMonitor API

4. Configuration example for Actility Network Server

Complete instructions are available in your "ThingPark Wireless Advanced Developer Guide" document, provided under NDA by Actility SA, chapter "Application Server". This document offers a basic tutorial for the minimum settings required to send the uRADMonitor LoRaWAN data to the uRADMonitor Application Server.

- Open your LoRaWAN console page:

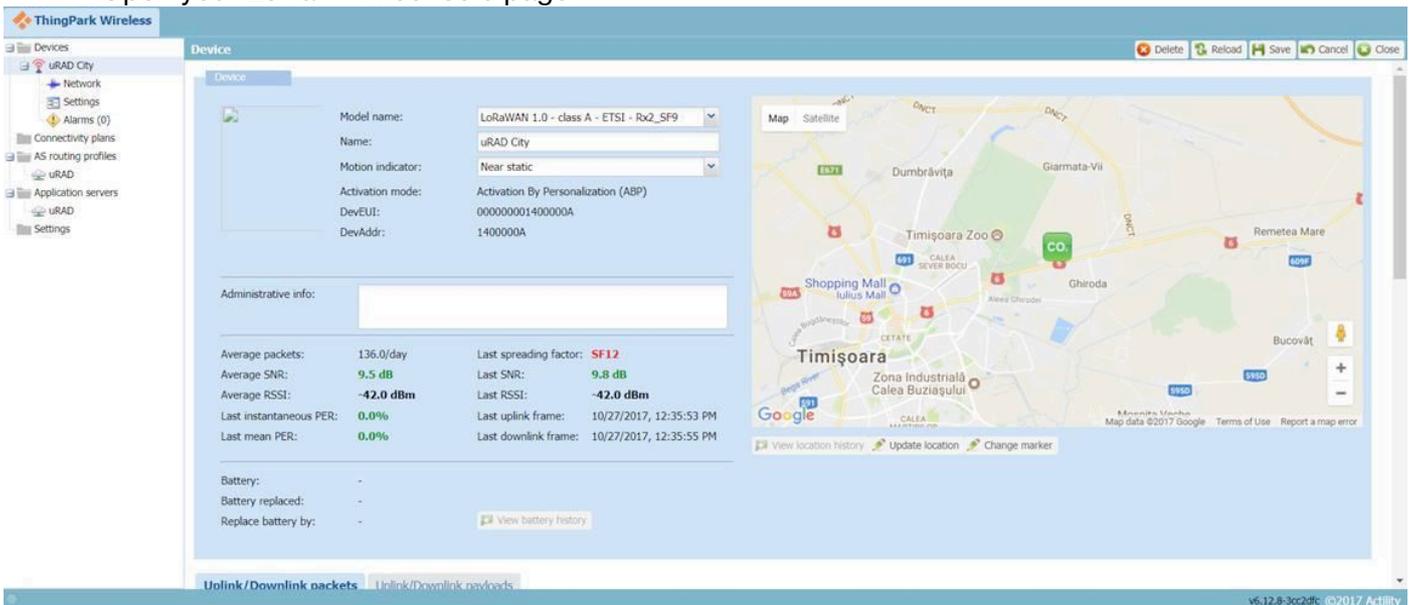


Figure 3. The Actility Dashboard

- Navigate to "Application servers", and create a new entry. You can set any name, preferable "uRADMonitor Application Server Callback". Set the Content Type to JSON. Source ports must be star "*" and Routing strategy must be set to Sequential:

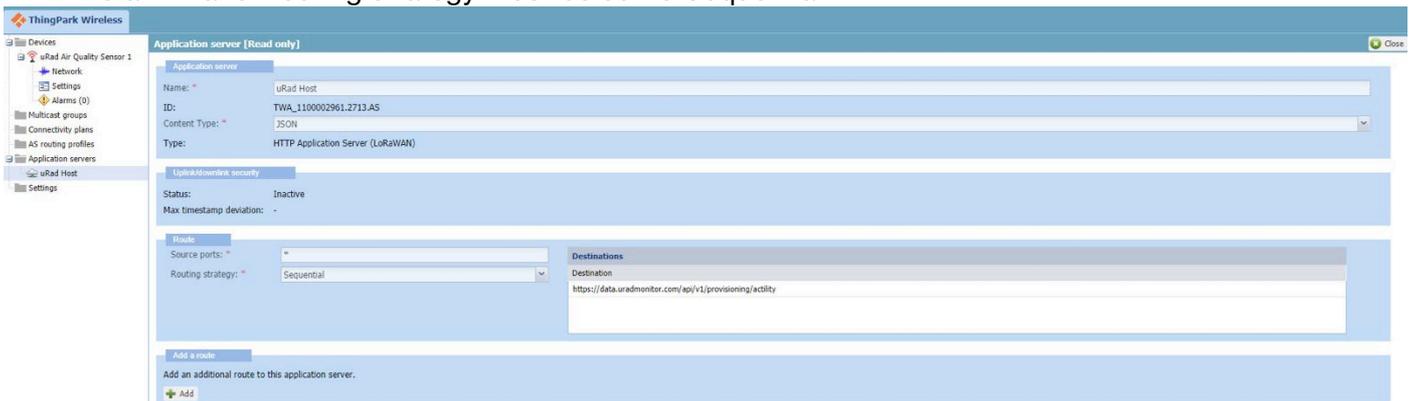


Figure 4. Configuring the Application Server under Actility's ThingPark console

- Edit destination to <https://data.uradmonitor.com/api/v1/provisioning/activity>

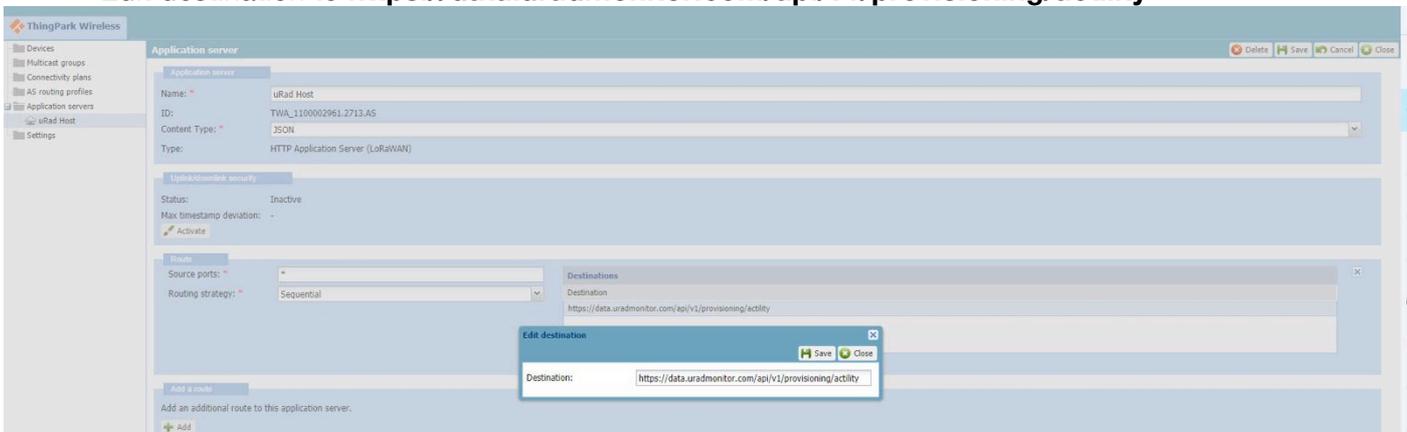


Figure 5. Setting the uRADMonitor callback URL

You will now be able to access the uRADMonitor data via the uRADMonitor API. The unit should also become visible on the uRADMonitor Global map on www.uradmonitor.com and you can jump directly to it by opening <https://www.uradmonitor.com/?open=YourUnitID> (eg.82000012)

5. Configuration example for TTN Network Server

TTN or "The Things Network" provides excellent documentation and configuration tutorials on <https://www.thethingsnetwork.org>. Additionally, you can receive community professional assistance related both to hardware options and software configuration on the TTN Forums: <https://www.thethingsnetwork.org/forum/>. This tutorial will only offer the basic steps to configure the uRADMonitor Application Server Callback with the TTN Network Server.

- Open the TTN console and go to Applications. Create a new application, using the "Add application" button:

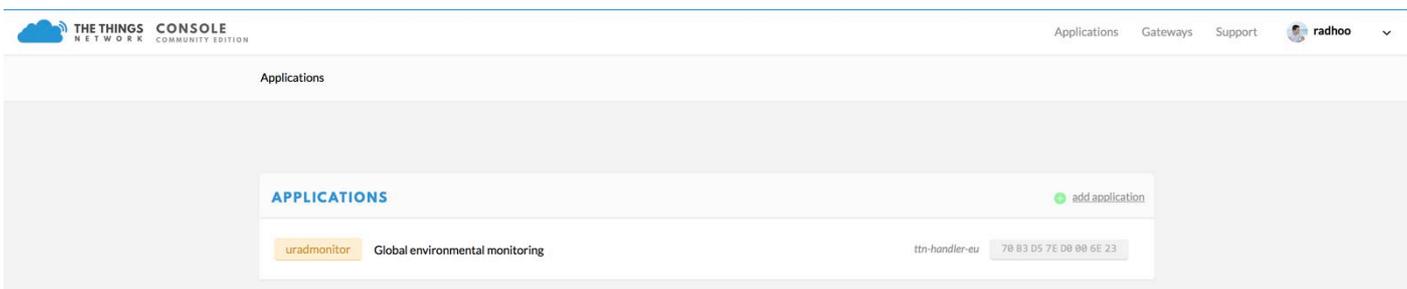


Figure 6. Creating a new application under TTN

- Open your newly created application, and go to the "Integrations" tab. Select "HTTP Integration"

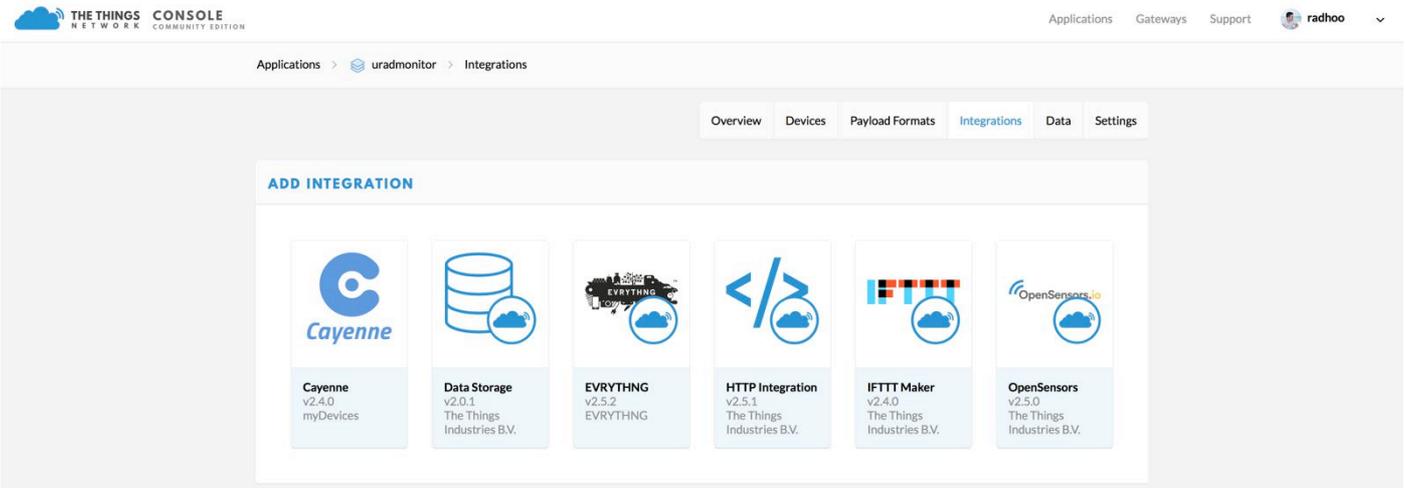


Figure 7. The TTN Callback is configured via the HTTP Integration option

You will only need to configure a unique Process ID, the Method as POST, and important, the URL must be set to the uRADMonitor server callback, <https://data.uradmonitor.com/api/v1/provisioning/ttn> :

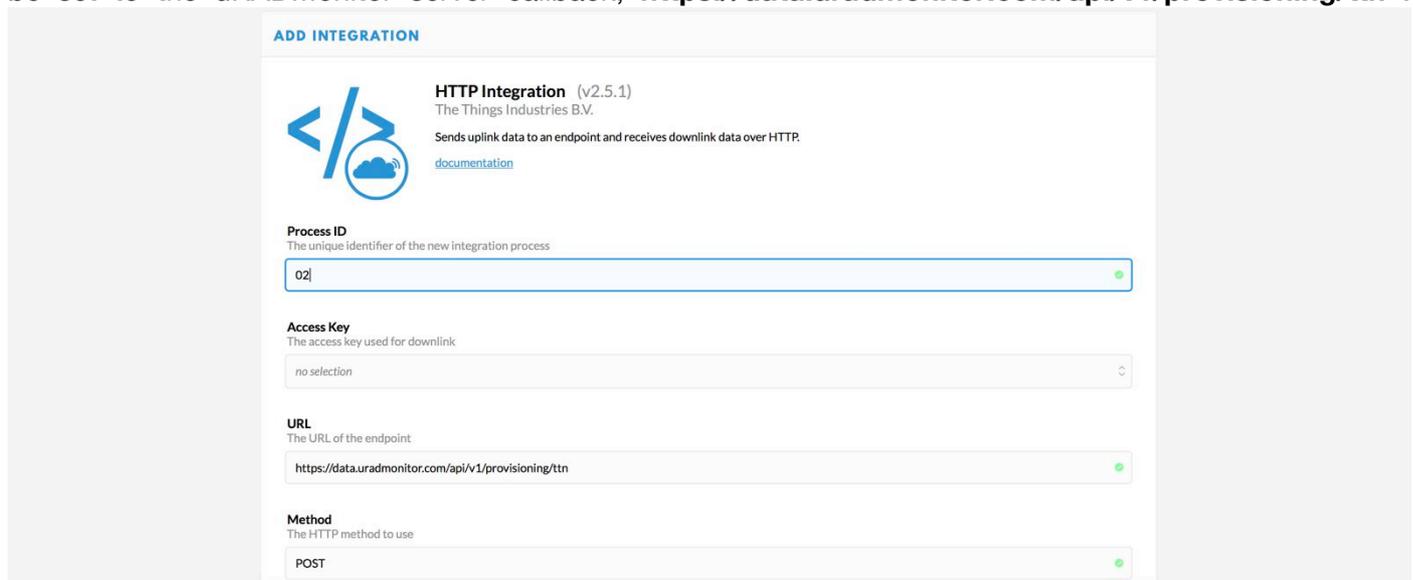


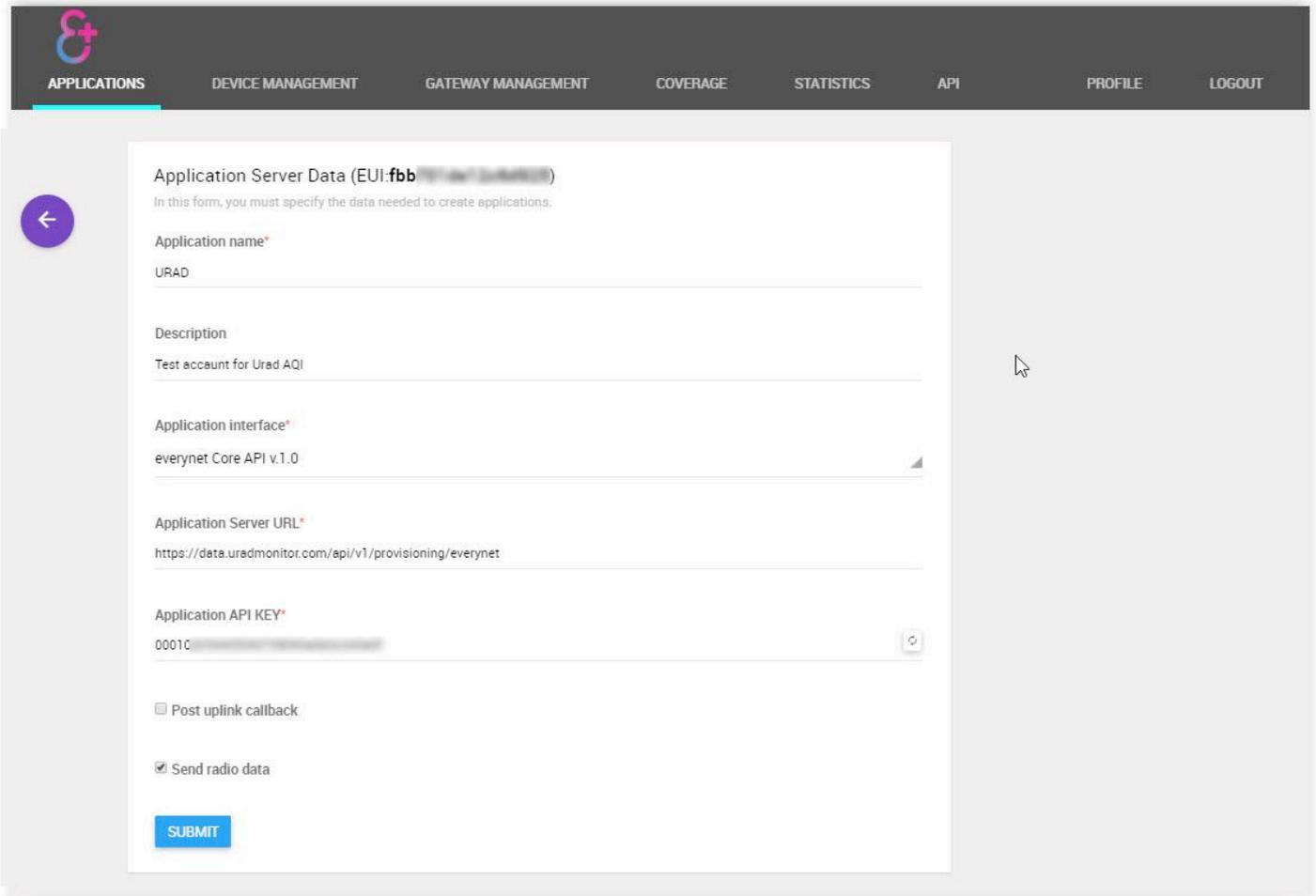
Figure 8. Configuring the URL

With the callback correctly configured, the data will be redirected to the uRADMonitor server, and the sensors will be addressable directly on www.uradmonitor.com .

6. Configuration example for Everynet Network Server

Everynet is an IoT enabler focused on providing innovative LPWA solutions like the LoRaWAN. Here is a short tutorial offering the basic steps to configure the uRADMonitor Application Server Callback with the Everynet platform:

- Open the settings page and go to "Applications" to setup a first application:



Application Server Data (EUI: fbb...)

In this form, you must specify the data needed to create applications.

Application name*
URAD

Description
Test account for Urad Aqi

Application interface*
everynet Core API v.1.0

Application Server URL*
https://data.uradmonitor.com/api/v1/provisioning/everynet

Application API KEY*
00010

Post uplink callback

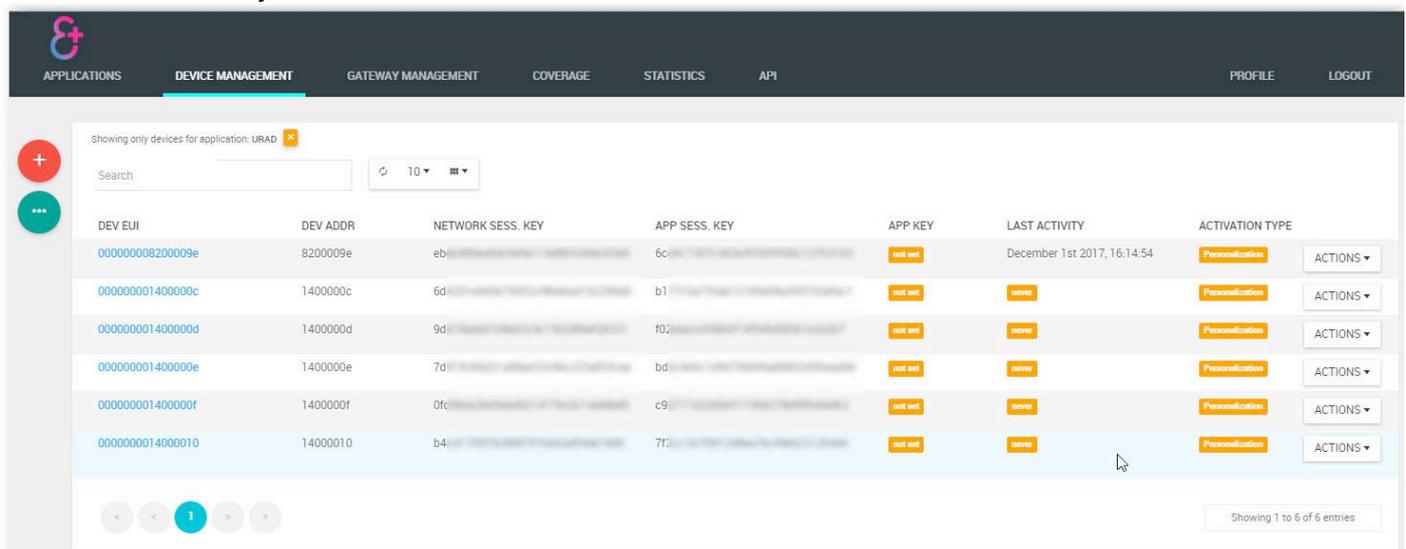
Send radio data

SUBMIT

Figure 9. Configuring the URL callback for Everynet platform

The "Application server URL" is: **https://data.uradmonitor.com/api/v1/provisioning/everynet**

- For the application you just created, you can add one or more uRADMonitor devices with the corresponding LoRaWAN settings, either ABP or OTAA. Data generated by these devices will automatically reach the uRADMonitor server



Showing only devices for application: URAD

DEV EUI	DEV ADDR	NETWORK SESS. KEY	APP SESS. KEY	APP KEY	LAST ACTIVITY	ACTIVATION TYPE	ACTIONS
000000008200009e	8200009e	eb...	6c...	not set	December 1st 2017, 16:14:54	Personalization	ACTIONS
000000001400000c	1400000c	6d...	b1...	not set	never	Personalization	ACTIONS
000000001400000d	1400000d	9d...	f0...	not set	never	Personalization	ACTIONS
000000001400000e	1400000e	7d...	bd...	not set	never	Personalization	ACTIONS
000000001400000f	1400000f	0f...	c9...	not set	never	Personalization	ACTIONS
0000000014000010	14000010	b4...	7f...	not set	never	Personalization	ACTIONS

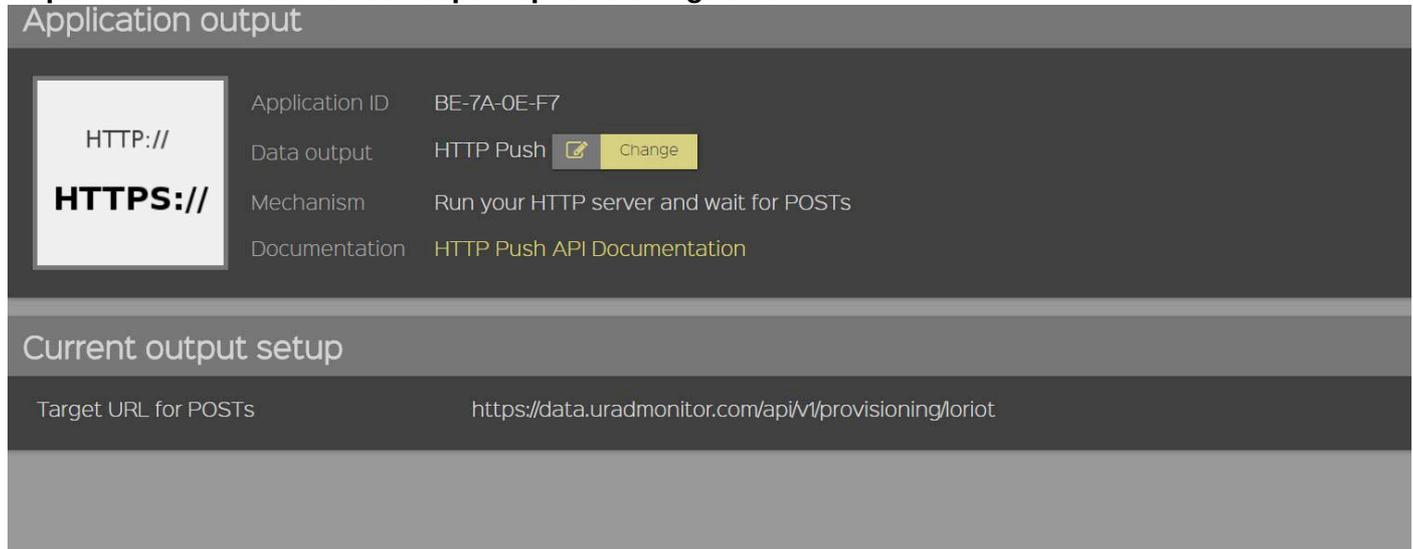
Showing 1 to 6 of 6 entries

Figure 10. Configuring the Everynet LoRaWAN devices

7. Configuration example for Loriot

This is probably one of easiest to configure, just open the Loriot configuration panel and under Application output add the uRADMonitor data server callback URL:

<https://data.uradmonitor.com/api/v1/provisioning/loriot>



The screenshot shows the Loriot configuration interface. On the left, there are two buttons: 'HTTP://' and 'HTTPS://'. The 'HTTPS://' button is selected. The main area is titled 'Application output' and contains the following information:

Application ID	BE-7A-0E-F7
Data output	HTTP Push <input type="button" value="Change"/>
Mechanism	Run your HTTP server and wait for POSTs
Documentation	HTTP Push API Documentation

Below this is the 'Current output setup' section, which shows the 'Target URL for POSTs' set to `https://data.uradmonitor.com/api/v1/provisioning/loriot`.

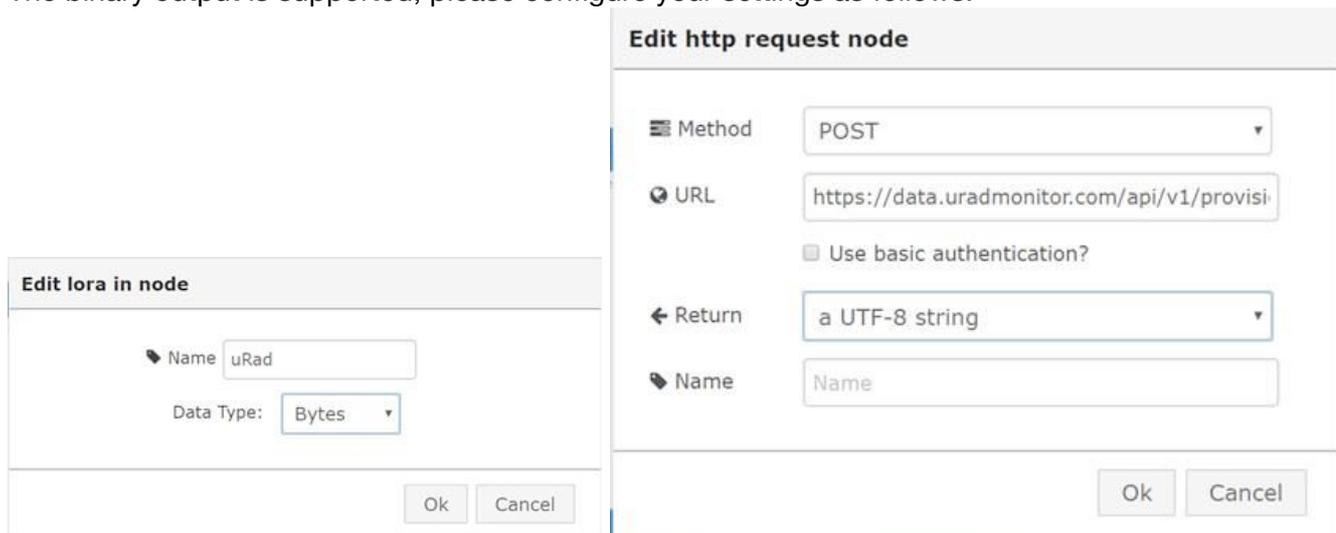
Figure 11. Configuring the Loriot LoraWAN callback for uRADMonitor data server

8. Multitech Node Red

The callback URL for Multitech network servers is

<https://data.uradmonitor.com/api/v1/provisioning/multitech>

The binary output is supported, please configure your settings as follows:



The screenshot shows two overlapping configuration windows. The 'Edit lora in node' window has the following settings:

- Name: `uRad`
- Data Type: `Bytes`

The 'Edit http request node' window has the following settings:

- Method: `POST`
- URL: `https://data.uradmonitor.com/api/v1/provisi`
- Use basic authentication?:
- Return: `a UTF-8 string`
- Name: `Name`

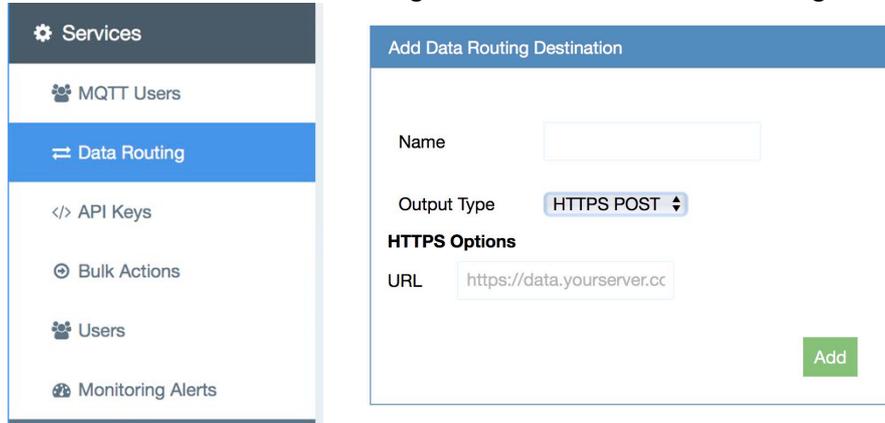
Figure 12. Example for Multitech Node Red config

9. Stream IoT-X

The callback URL for the IoT-X network servers is

<https://data.uradmonitor.com/api/v1/provisioning/streamiotx>

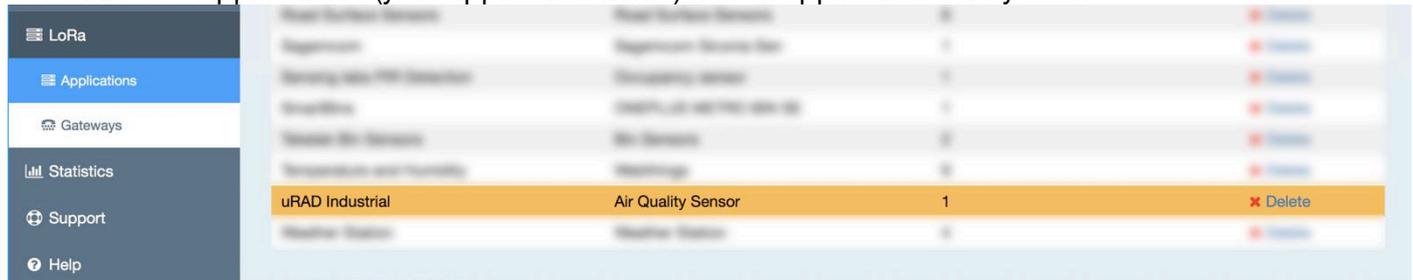
Select "Services->Data Routing". Create new data route using the URL above:



The screenshot shows the 'Add Data Routing Destination' form. On the left is a sidebar menu with 'Data Routing' selected. The form contains the following fields:

- Name:** An empty text input field.
- Output Type:** A dropdown menu set to 'HTTPS POST'.
- HTTPS Options:** A section containing a 'URL' field with the placeholder 'https://data.yourserver.cc' and a green 'Add' button.

Select LoRa->Application->(your application name) .. click application set by user.



The screenshot shows the 'Applications' list. The 'uRAD Industrial' application is highlighted in orange. The table below shows the details of the applications:

Application Name	Type	Count	Actions
uRAD Industrial	Air Quality Sensor	1	Delete

Click on the device id:

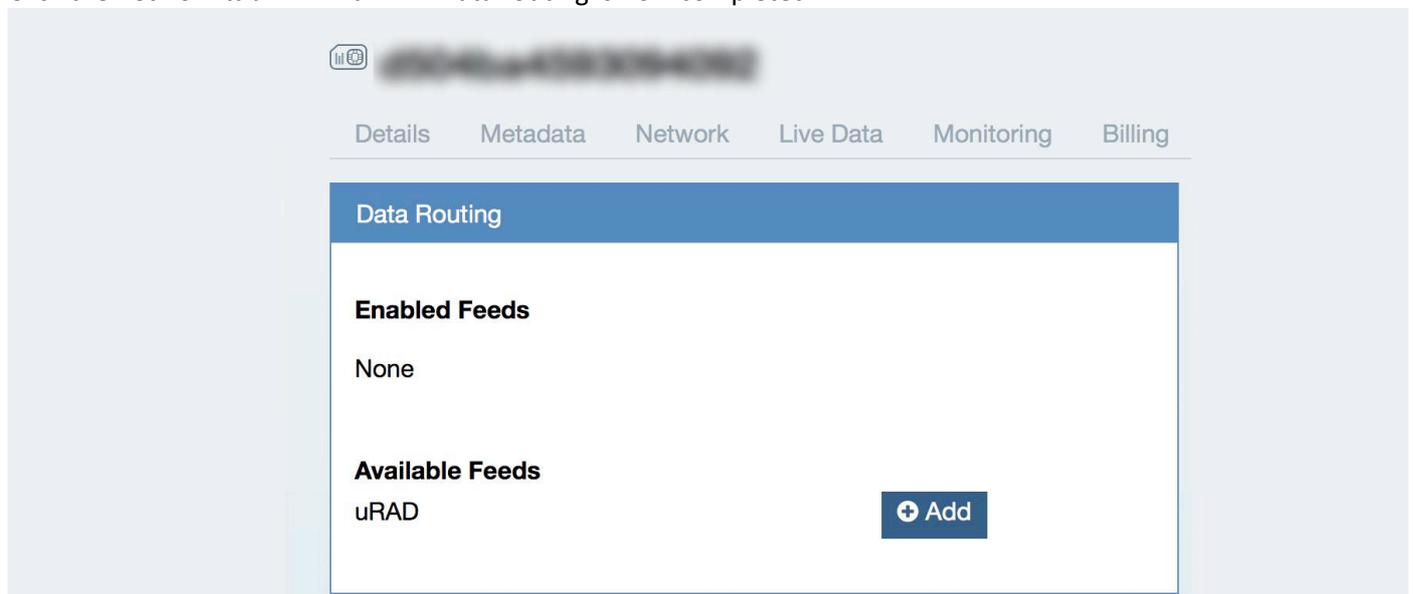


The screenshot shows the 'Application Devices' view for a specific device. The table below shows the device details:

Device ID	Type	Activated
[Device ID]	OTA	✓

Showing 1 to 1 of 1 entries

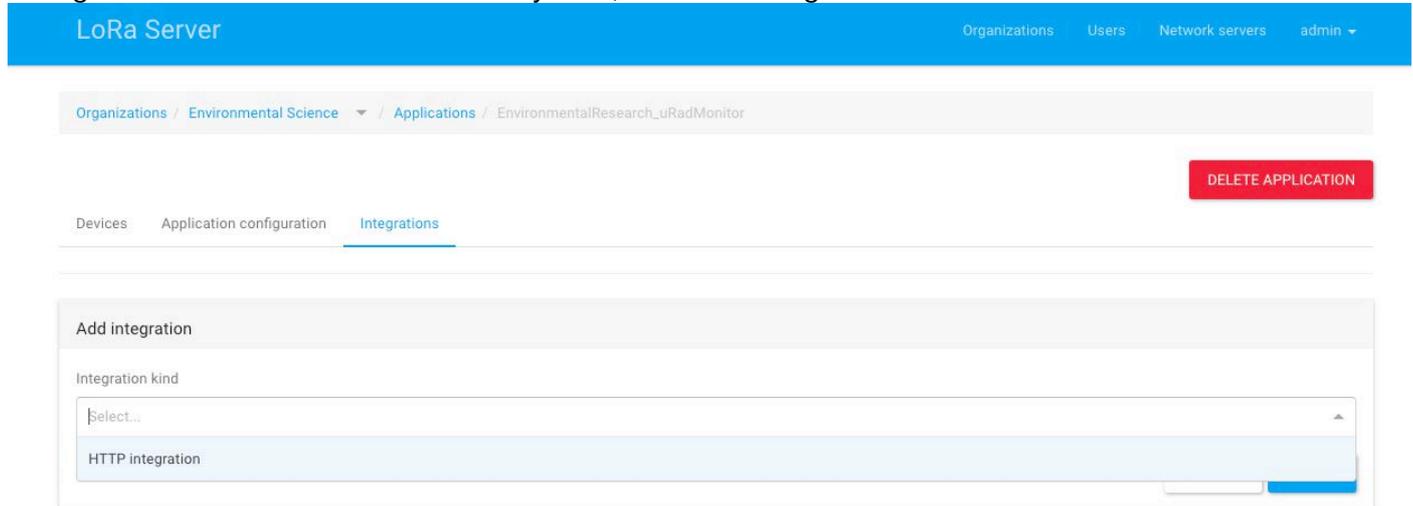
Click the network tab > ADD uRAD . Data routing is now completed.



The screenshot shows the 'Data Routing' configuration page. The 'Enabled Feeds' section is empty, and the 'Available Feeds' section shows 'uRAD' with an 'Add' button.

10. LoRaServer.io

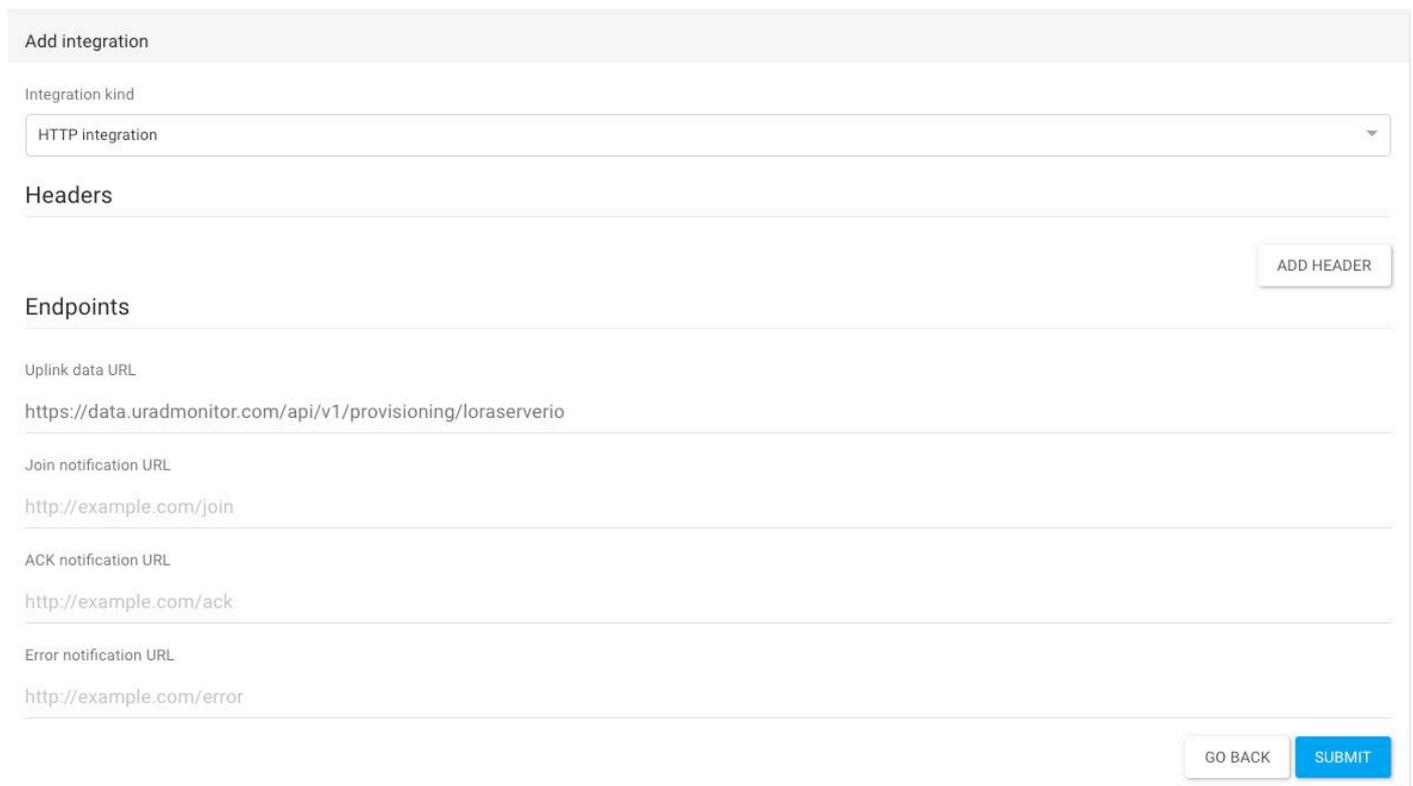
The LoRa Server project provides open-source components for building LoRaWAN networks. To configure the callback on this network system, first click Integrations:



The screenshot shows the LoRa Server web interface. At the top, there is a blue navigation bar with "LoRa Server" on the left and "Organizations", "Users", "Network servers", and "admin" on the right. Below the navigation bar, there is a breadcrumb trail: "Organizations / Environmental Science / Applications / EnvironmentalResearch_uRadMonitor". On the right side, there is a red button labeled "DELETE APPLICATION". Below the breadcrumb trail, there are three tabs: "Devices", "Application configuration", and "Integrations", with "Integrations" being the active tab. The main content area is titled "Add integration" and contains a form with the following fields:

- Integration kind: A dropdown menu with "select..." and "HTTP integration" as options.

Select Integration kind as "HTTP Integration" and add the following URL:
<https://data.uradmonitor.com/api/v1/provisioning/loraserverio>



The screenshot shows the configuration page for an HTTP integration. The form is titled "Add integration" and contains the following fields:

- Integration kind: A dropdown menu with "HTTP integration" selected.
- Headers: A section with an "ADD HEADER" button.
- Endpoints: A section with the following fields:
 - Uplink data URL: <https://data.uradmonitor.com/api/v1/provisioning/loraserverio>
 - Join notification URL: <http://example.com/join>
 - ACK notification URL: <http://example.com/ack>
 - Error notification URL: <http://example.com/error>

At the bottom right of the form, there are two buttons: "GO BACK" and "SUBMIT".



11. Support

For additional questions or for technical support please contact support@uradmonitor.com or visit www.uradmonitor.com.