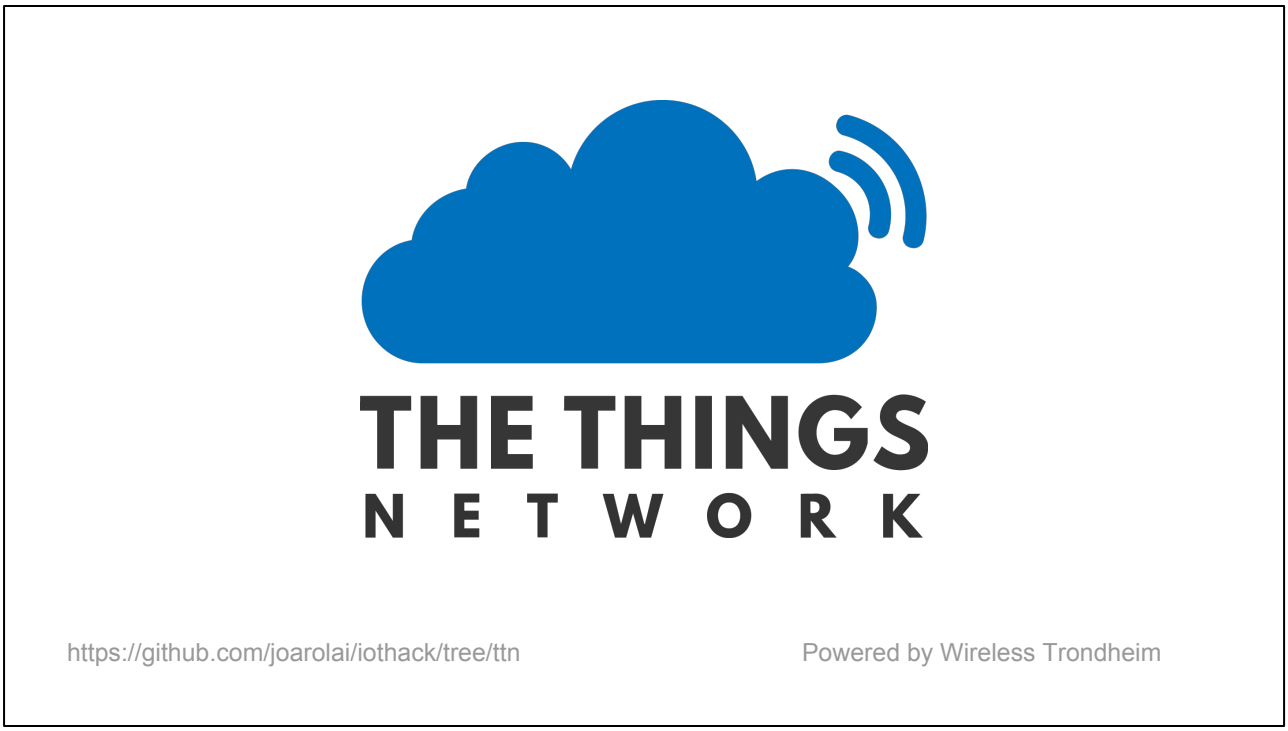
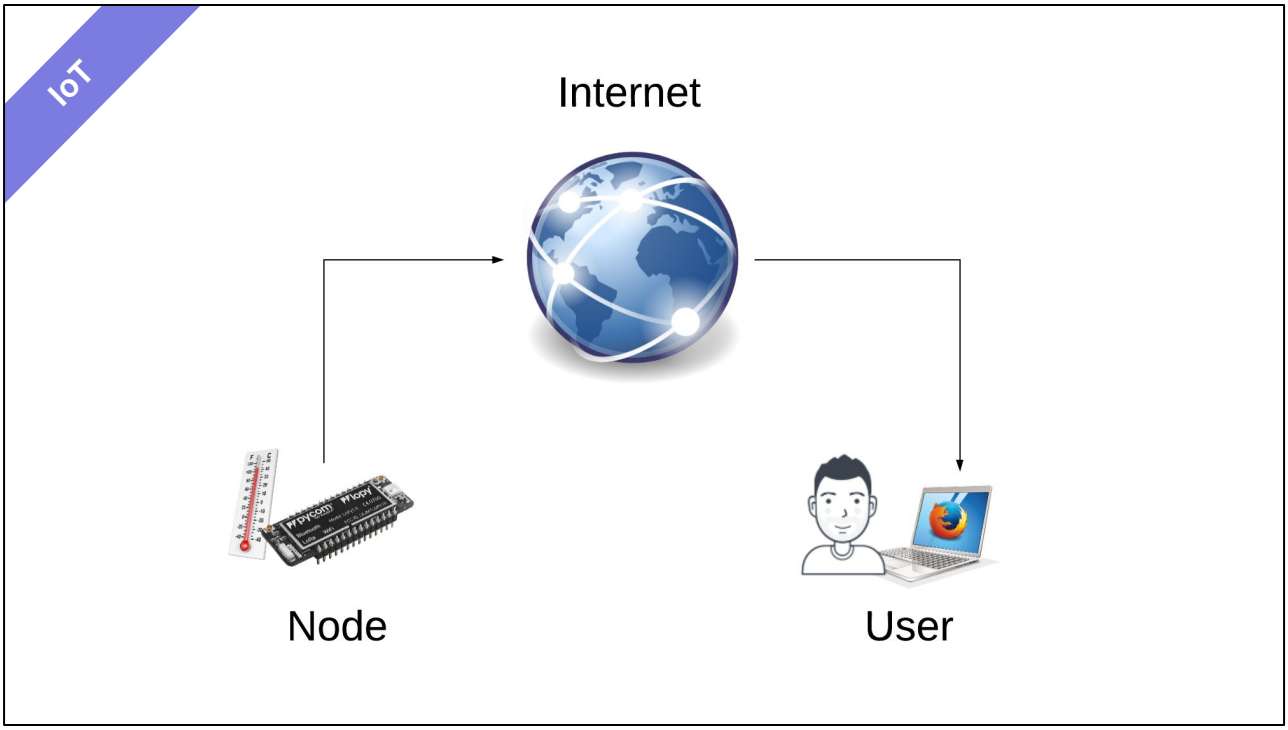
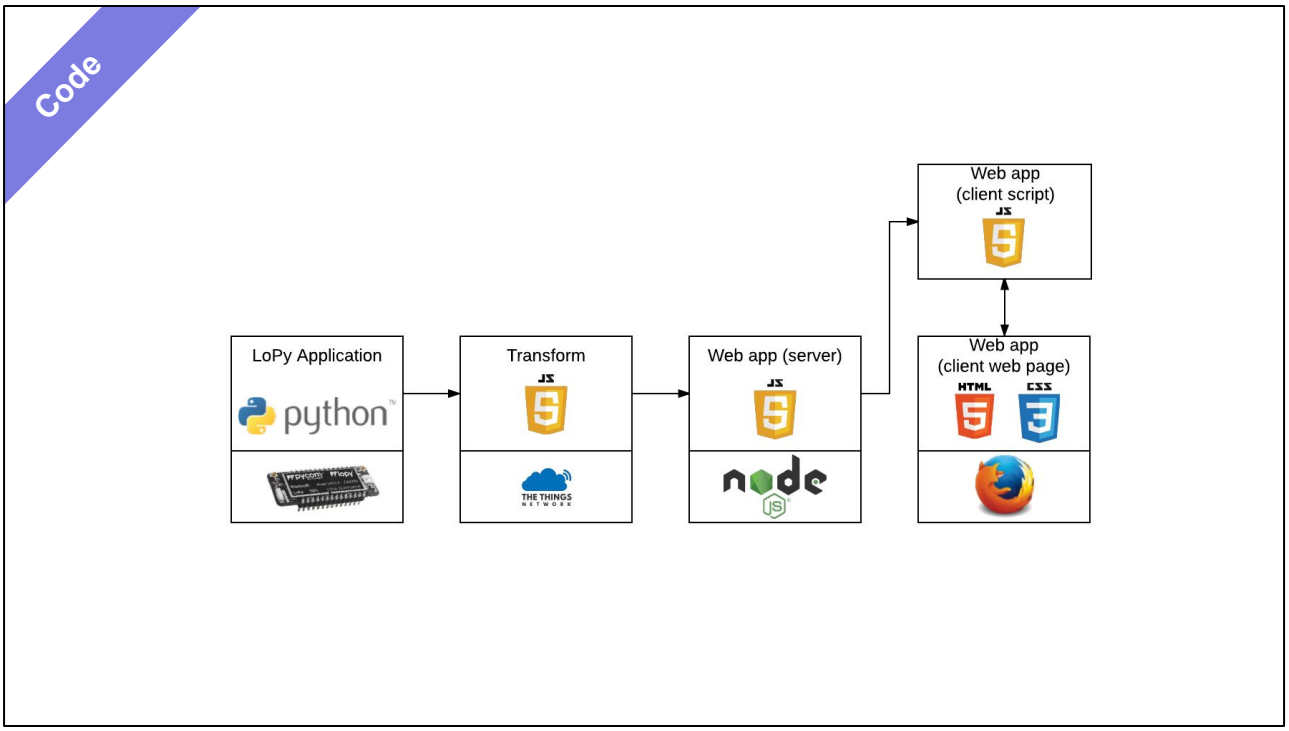

THE THINGS
NETWORK
https://github.com/joarolai/iothack/tree/ttn
Powered by Wireless Trondheim

Version 1.1:



In this presentation we are looking at a simple application that is sending sensor data (temperature) via the internet to a web application where the users can observe the temperature change in real time (more or less) via their browsers.

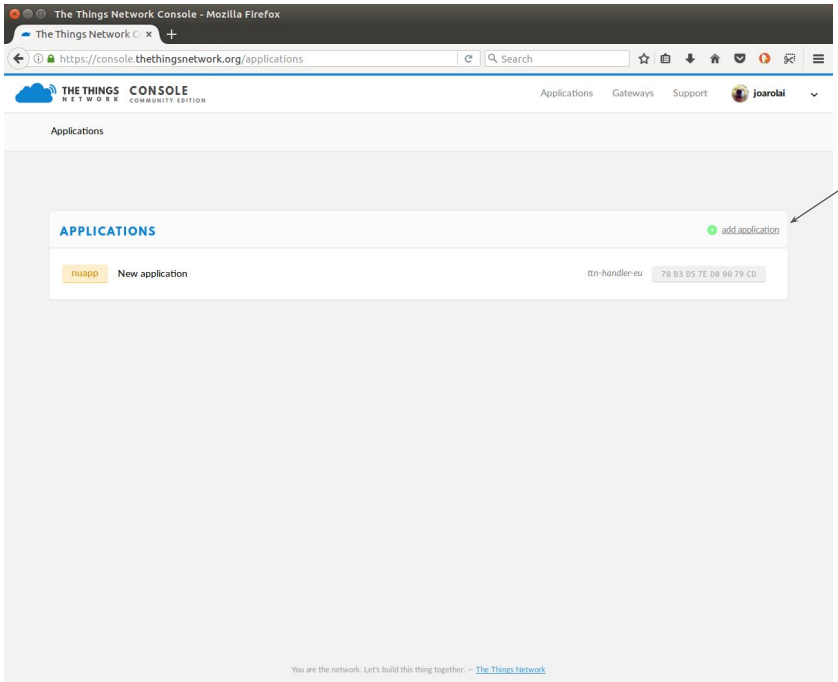
Node = Sensor node operating on a network, also called a mote
TTN = The Things Network



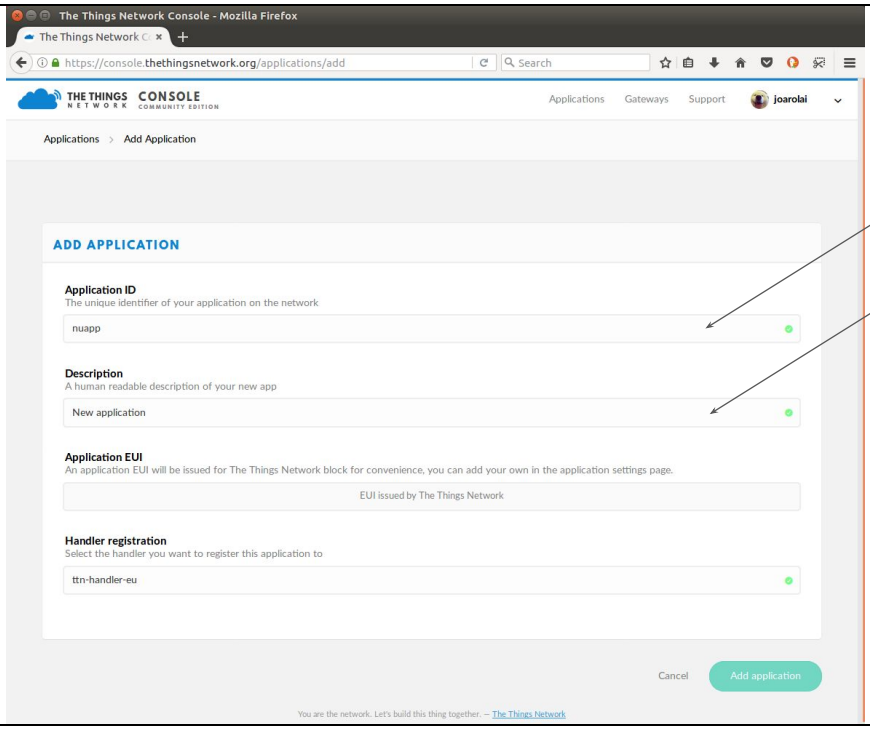
A typical IoT application may include several components that each need custom code developed for it. In our example we have the following:

- Code for the LoPy written in Python
- A transform script written in Javascript that shall run in TTN
- A node.js application written in Javascript that shall run on your web server
- A user interface in the form of a webpage written in HTML, CSS and Javascript

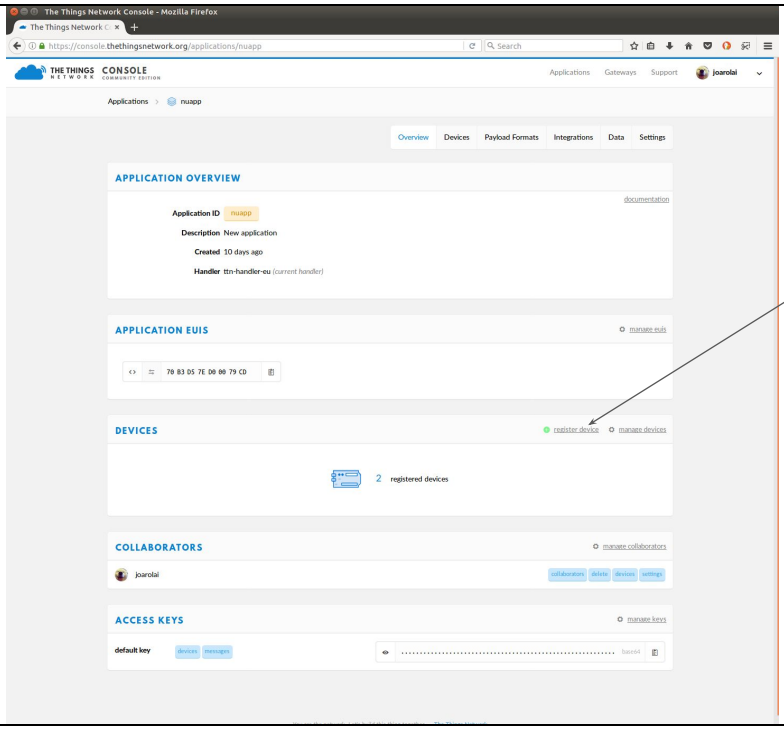
Console

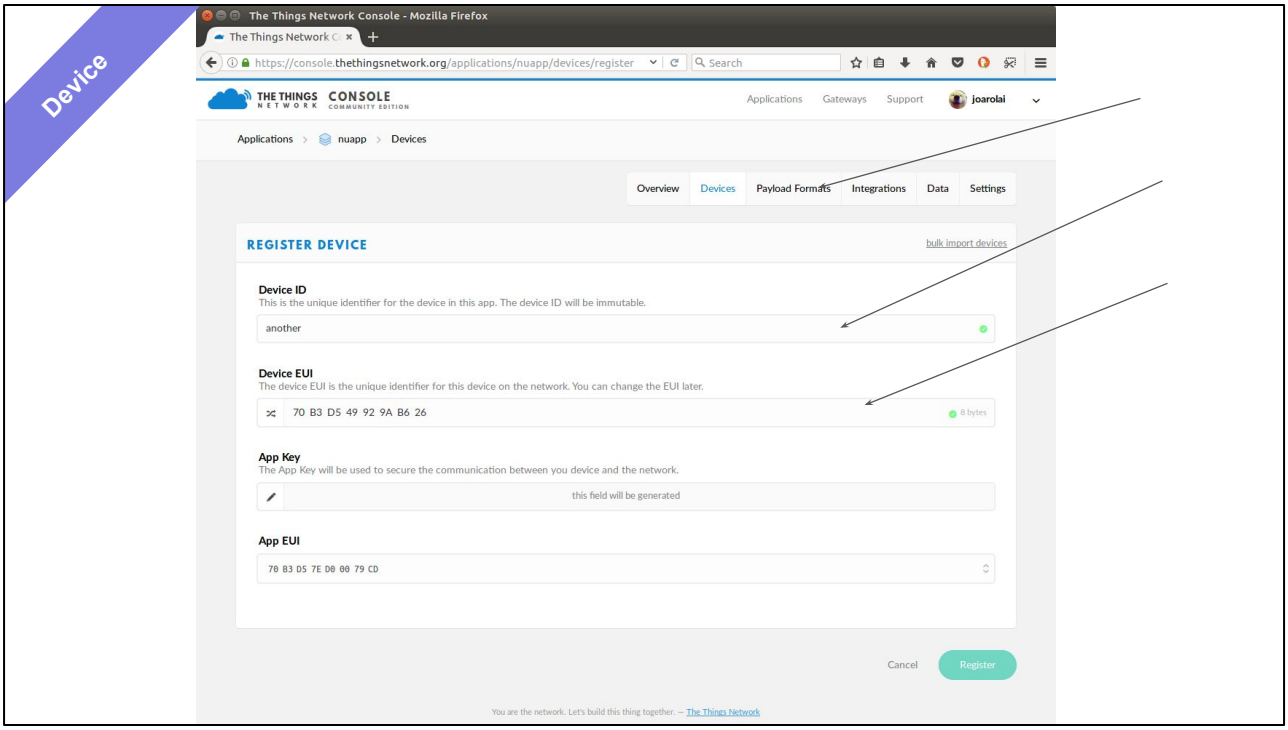

The Things Network Console - Mozilla Firefox
The Things Network
https://console.thethingsnetwork.org/applications
Search
THE THINGS NETWORK CONSOLE COMMUNITY EDITION
Applications
Gateways
Support
joarolai
Applications
APPLICATIONS
add application
nuapp
New application
ttn-handler-eu
70 B3 D5 7E D0 00 79 CD
You are the network. Let's build this thing together. – The Things Network

Application



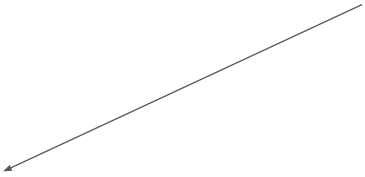
Register device




Device
The Things Network Console - Mozilla Firefox
The Things Network
https://console.thethingsnetwork.org/applications/nuapp/devices/register
Search
THE THINGS NETWORK CONSOLE COMMUNITY EDITION
Applications
Gateways
Support
joarolai
Applications
nuapp
Devices
Overview
Devices
Payload Formats
Integrations
Data
Settings
REGISTER DEVICE
bulk import devices
Device ID
This is the unique identifier for the device in this app. The device ID will be immutable.
another
Device EUI
The device EUI is the unique identifier for this device on the network. You can change the EUI later.
70 B3 D5 49 92 9A B6 26
8 bytes
App Key
The App Key will be used to secure the communication between you device and the network.
this field will be generated
App EUI
70 B3 D5 7E D0 00 79 CD
Cancel
Register
You are the network. Let's build this thing together. – The Things Network

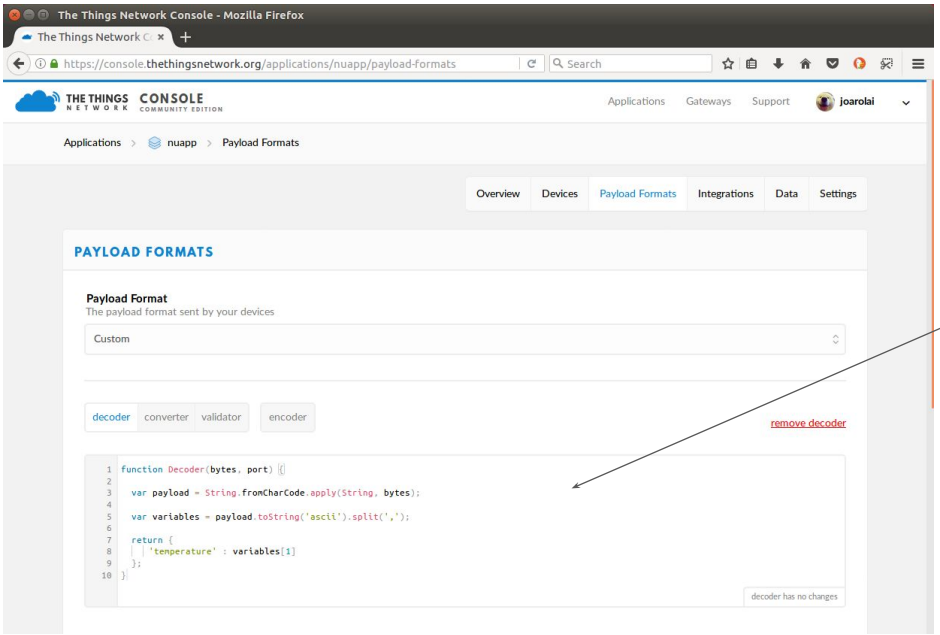
Device EUI

Use REPL for this



```
import network,binascii
binascii.hexlify(network.LoRa().mac())
```

Payload formats



Transform

```
function Decoder(bytes, port){

      var payload = String.fromCharCode.apply(String, bytes);

      var variables = payload.toString('ascii').split(',');

      return {
            'temperature' : variables[1]
      };
}
```

Data structures

Raw data (string)

TEMP,49

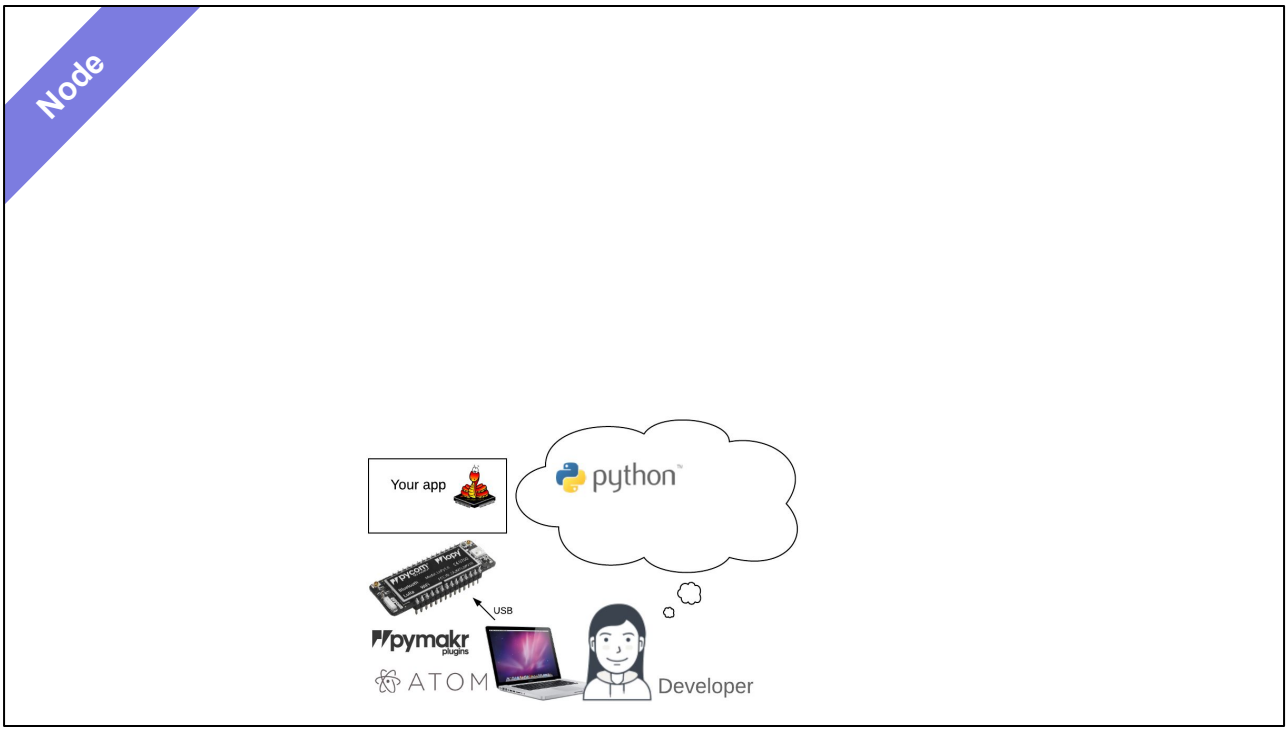
Uplink data (data structure)

TTN data structure

```
{
   app_id: 'nuapp',
   dev_id: 'another',
   hardware_serial: '70B3D549929AB626',
   port: 2,
   counter: 5892,
   payload_raw: <Buffer 54 45 4d 50 2c 35 38 39 31>,
   payload_fields: {
      temperature: '49'
   },
   metadata: {
      time: '2017-10-10T11:34:50.160318598Z',
      frequency: 867.9,
      modulation: 'LORA',
      data_rate: 'SF7BW125',
      coding_rate: '4/5',
      gateways: [ [Object] ]
   }
}
```

TTN gateways

```
[ {
    gtw_id: 'trt-samf-loragw01',
    gtw_trusted: true,
    timestamp: 3869682171,
    time: '2017-10-10T11:37:47Z',
    channel: 4,
    rssi: -118,
    snr: -9.25,
    rf_chain: 0,
    latitude: 63.422485,
    longitude: 10.395755,
    altitude: 20
 } , {
    gtw_id: 'eui-008000000000bc6c',
    timestamp: 4068030371,
    time: '2017-10-10T11:33:13.36681Z',
    channel: 4,
    rssi: -115,
    snr: -4.2,
    rf_chain: 0,
    latitude: 63.42883,
    longitude: 10.3857,
    altitude: 21
 } ]
```



A sensor node usually contain:

- A sensor
- A microcontroller with networking capability
- A microcontroller application

For our scenario the LoPy is our microcontroller, and our application is written in a variant of Python called Micropython.

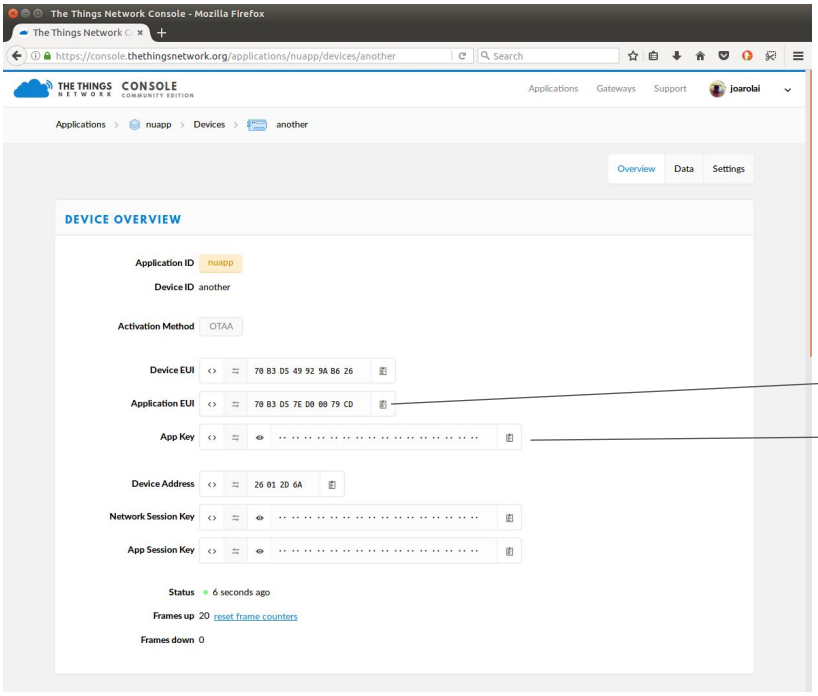
Our application can be created using the Atom editor, and the LoPy can be programmed directly from Atom, via USB, after installing the Pymakr plugin.

https://atom.io/
https://docs.pycom.io/chapter/gettingstarted/installation/pymakr.html

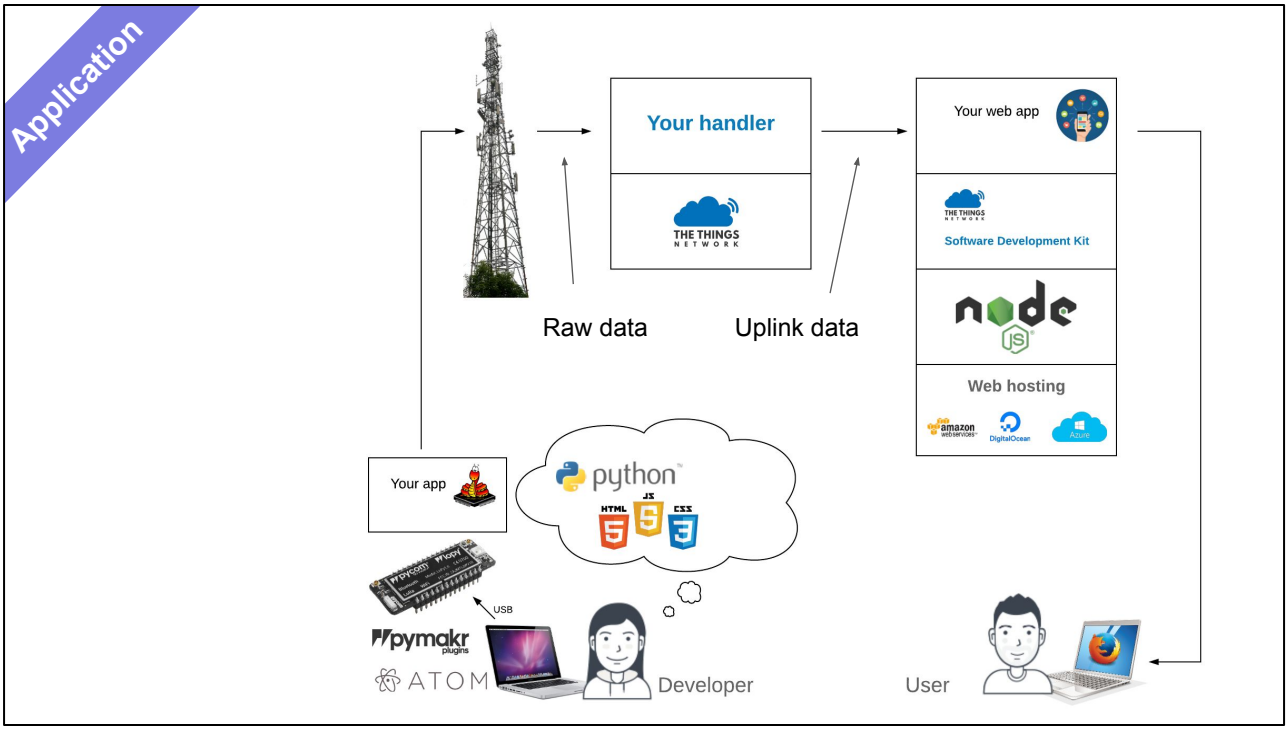
You should also consider going through the complete getting started guide from Pycom: https://docs.pycom.io/chapter/gettingstarted/

You are not required to install python on your computer since micropython runs inside the LoPy.

App Key



lopy/main.py

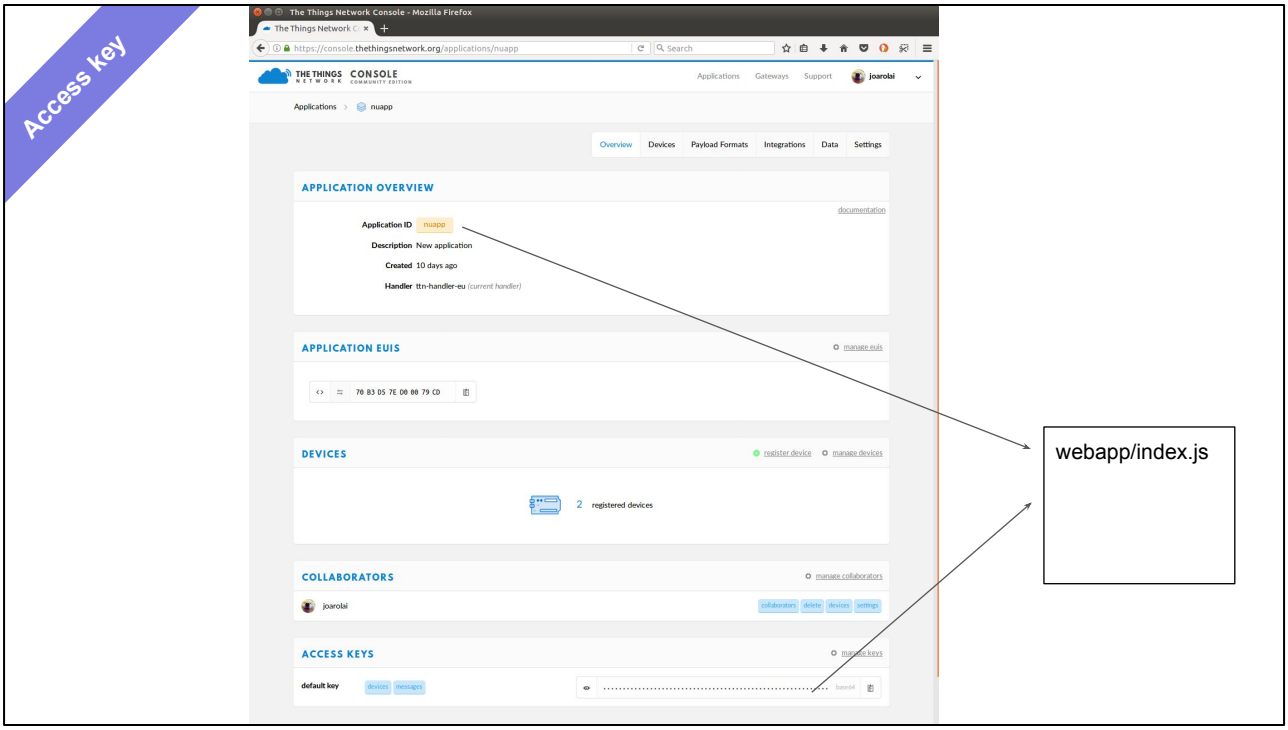


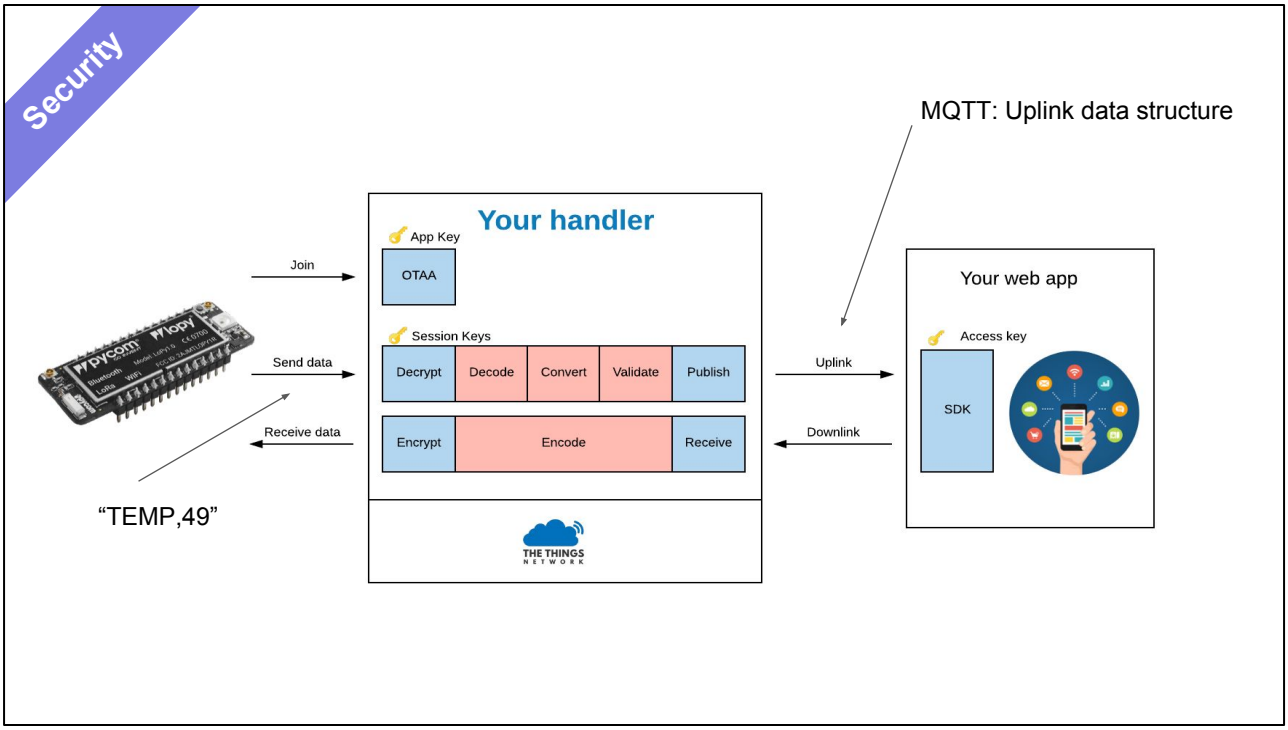
When you have been able to send data to TTN from your LoPy, the next step is to present the data in a web application that your users can access.

The code shown in the next slide, and the transform shown in the slide after that is needed to get the application in the tutorial to work.

If you are hosting the node.js server on your own computer then you do not need to change anything in the **index.html** file

If you want to install the node.js server on a hosting service like Digital Ocean, you must change line 34 in **index.html** to match the IP/domain of your server instead of **localhost**


Access key
The Things Network Console - Mozilla Firefox
https://console.thethingsnetwork.org/applications/nuapp
THE THINGS NETWORK CONSOLE
Applications
Gateways
Support
joarolai
Applications
nuapp
Overview
Devices
Payload Formats
Integrations
Data
Settings
APPLICATION OVERVIEW
documentation
Application ID nuapp
Description New application
Created 10 days ago
Handler ttn-handler-eu (current handler)
APPLICATION EUIS
manage euis
70 B3 D5 7E D0 00 79 CD
DEVICES
register device
manage devices
2 registered devices
COLLABORATORS
manage collaborators
joarolai
collaborators
delete
devices
settings
ACCESS KEYS
manage keys
default key
devices
messages
base64
webapp/index.js


Security
MQTT: Uplink data structure
Your handler
App Key
OTAA
Session Keys
Decrypt
Decode
Convert
Validate
Publish
Encrypt
Encode
Receive
Join
Send data
Receive data
Uplink
Downlink
Your web app
Access key
SDK
THE THINGS NETWORK
"TEMP,49"