

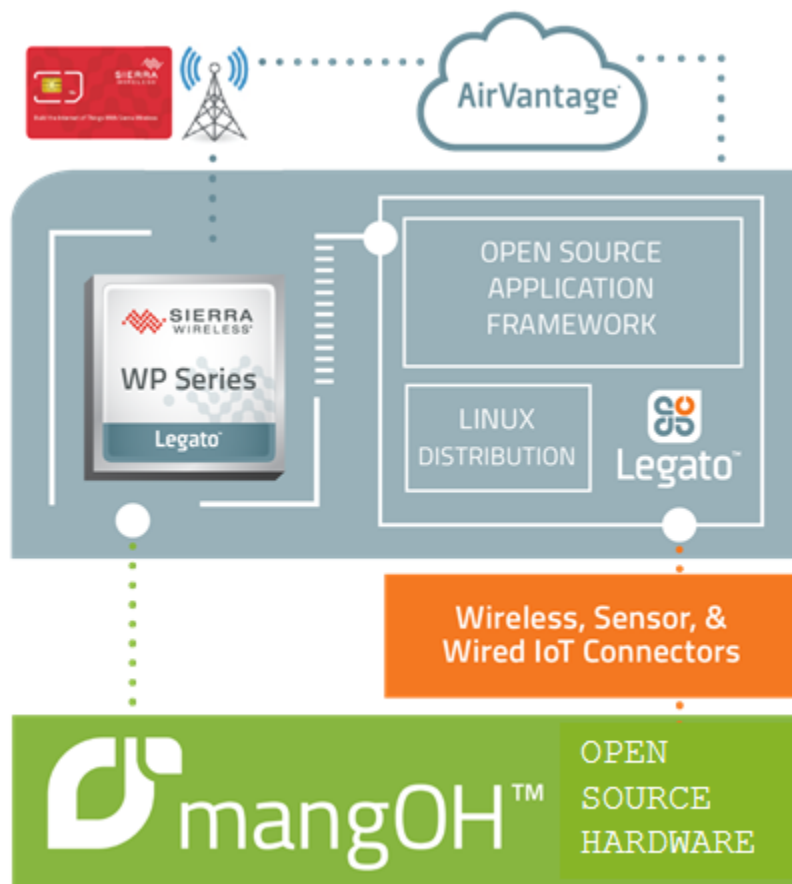
# Sierra Wireless (Legato/AirVantage APIs)

Sierra Wireless Device to Cloud platform will provide solution for end to end IoT applications.

The document below is in the following format:

1. Introduction to Device to Cloud platform
2. APIs Hackathon
3. Architecture Explanation
4. Using the platform

## Introduction to Device to Cloud platform->

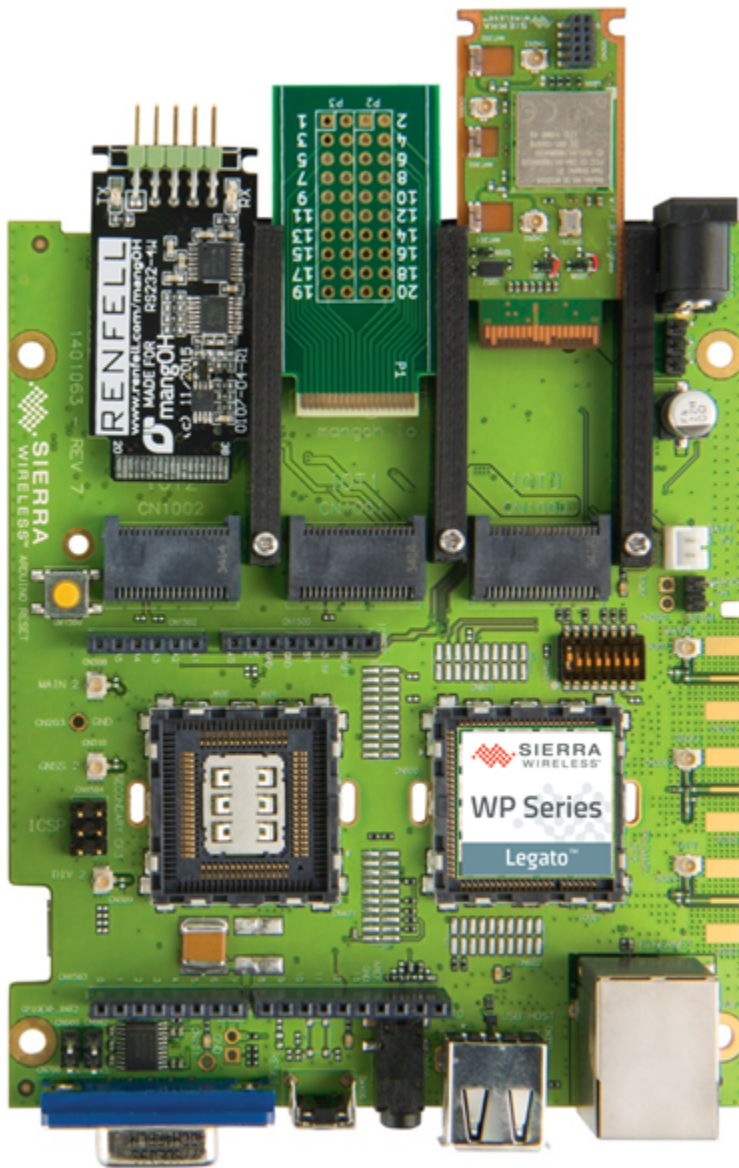


The platform includes the following components:

- a. **mangOH Green open hardware platform:** mangOH Green is an open hardware reference design purpose built for cellular connectivity.

Smart WP module forms the core of the mangOH platform . It incorporates a telecom core that will provide 2G to 4G connectivity as well as a GNSS connection. It also has an application core with multiple interfaces like SPI,I2C,SDIO, HSIC, UART, USB. These interfaces allows developers to connect Wireless, Wired and sensor technologies. The present gen WP application core has a cortex A5 at 550MHz and 2Gb NAND and 1Gb RAM reserved for application development

mangOH allows you to use existing open hardware Arduino compatible shields. There is an Arduino compatible connector that allows any Arduino shields to be plugged on top of the mangoh. There is an onboard ATmega32u4 chipset directly talking to the WP module. We have a prebuilt legato bridge application allowing seamless connection between the Arduino shield and WP module. We have a two way data exchange between the Arduino shield and anyother application connected to the mangoh. We also provide a couple of additional APIs that allow you to push the data to the cloud. For the Hackathon, the Arduino Legato app (called Arduino Bridge) talks to the ATmega chipset and takes care of the bringing data from arduino sensors into the WP module





b. **Legato Open Source Embedded Linux Framework**: An open source Linux-based embedded platform designed to simplify connected IoT application development. Using Legato Framework developers with limited hardware, wireless, embedded or cloud expertise can build applications easily.

There are essentially 3 parts to Legato.

1. The first part is a **Linux Distribution**. Specifically, it's an embedded Linux distribution tailored to run on Sierra Wireless's Linux-capable modules to start with (since Sierra has funded the whole project so far), with select security features enabled. Legato Linux is based on the Linux Foundation's Yocto Project.
2. The second part is the **Legato Application Framework**. This builds on top of standard POSIX and Linux APIs, complementing them. It includes Connectivity APIs and tools to make using cellular easy. It provides helpful tools for building and deploying embedded applications into the field. Includes support for multiple programming languages. And it also provides application sandboxing. That is all applications are isolated from each other to help prevent any problems in one application taking down another. The framework update process also enables automatic rollback so that a bad software update will not brick your device.
3. The third part is the **development environment**.

We start with command-line tools that help you:

- build your apps.
- configure your devices.
- install, remove, start, and stop apps on your devices.
- debug your running code.

We then add an Eclipse-based IDE, which supports most of the features offered by the command-line tools, and adds graphical tools for

- discovering target devices on your network.
- managing and editing files directly on your device via sftp.
- doing symbolic debugging using a graphical frontend to gdb.
- editing source code, with syntax highlighting and context-sensitive help.

c. **Air Vantage cloud platform**: AirVantage lets you capture and exploit data from any device deployed in the field, opening many new possibilities for your business processes, service applications, and revenue streams.

d. **Sierra Wireless Smart SIM**: The Sierra Wireless Smart SIM is specifically designed for mission critical mobile and fixed IoT use cases.

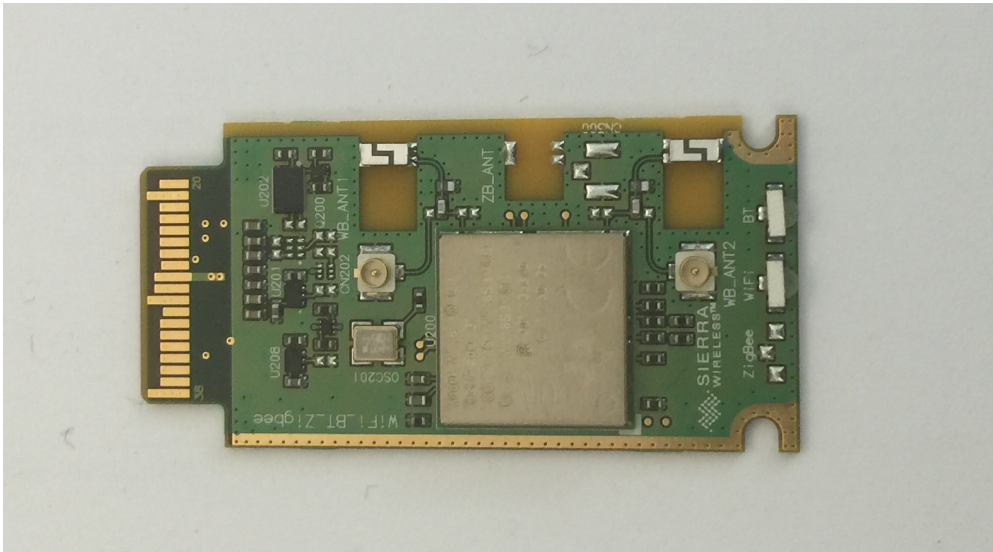
## **mangOH Green open hardware platform**

More information on mangOH/Legato/AV can be found at:

- [mangOH/Legato webinar](#):
- [mangOH Getting Started](#):
- [How to use API with AirVantage.docx](#)

**Connectivity for the hackathon** is provided by the following :

- [2G/3G/4G CF3 WP modem](#)
- WiFi /BT IoT Expander card for Client or AP mode and to attach to BLE sensors
- Location sensing is provided by an on board GPS.
- Sierra Wireless Smart SIM



WiFi/BT IoT Expander Card

## APIs for Hackathon

The APIs we will provide for building various IoT use cases include (but not limited to):

### Legato API

- Communication Manager/Modem services: (<http://legato.io/legato-docs/latest/legatoServicesModem.html>)
- Location Sensing: (<http://legato.io/legato-docs/latest/legatoServicesPositioning.html>)
- WiFi: (<http://legato.io/legato-docs/latest/legatoServicesWiFi.html>)
- Data Router: (<https://github.com/mangOH/DataRouter/blob/master/dataRouter.api>)
- MQTT: ( <https://github.com/mangOH/MqttClient/blob/master/mqtt.api>)
- Misc API found at [legato.io](http://legato.io)

### AirVantage API:

API to connect AirVantage to enterprise software (<https://doc.airvantage.net/av/howto/gettingstarted/>)

### API Methods

#### SYSTEM

Find  
Create  
Edit  
Delete

#### COMMUNICATION

Apply Settings  
Retrieve Data  
Applicative command  
Text SMS

#### DATA & MESSAGES

Last datapoints  
Historical aggregated datapoints  
Historical raw datapoints  
Export datapoints  
List of messages  
Details of a message

#### APPLICATIONS

Install Application  
Uninstall Application  
Start Application  
Stop Application

#### DEVICE MANAGEMENT

Synchronize  
Configure communication  
AT command  
Wake-up  
Reboot  
Factory Reset

### API Methods

#### APPLICATION

Create  
Find  
Find public  
Edit  
Edit communication  
Edit data  
Delete  
Validate  
Get published partners  
Get application model  
Get protocols  
Get data configuration  
Get actions

#### OPERATIONS

Release  
Publish  
Share  
Add  
Deprecate  
Change labels

#### LIFECYCLE

Activate  
Switch to maintenance

#### OTHER OPERATIONS

Import  
Multiple Edit  
Multiple Delete  
Export  
Change labels

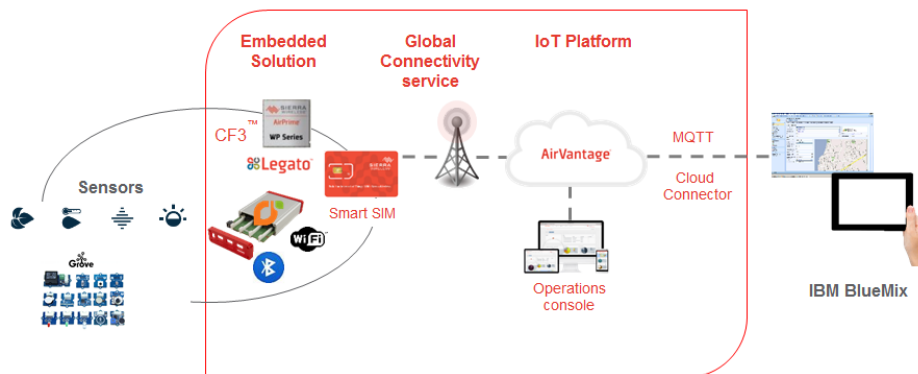
## Hackathon Architecture

The IoT application will take input from various Bluetooth or Grove Studio sensors and provide actuator functions as needed. Bluetooth data collection is done by on board Bluetooth IoT expander card.

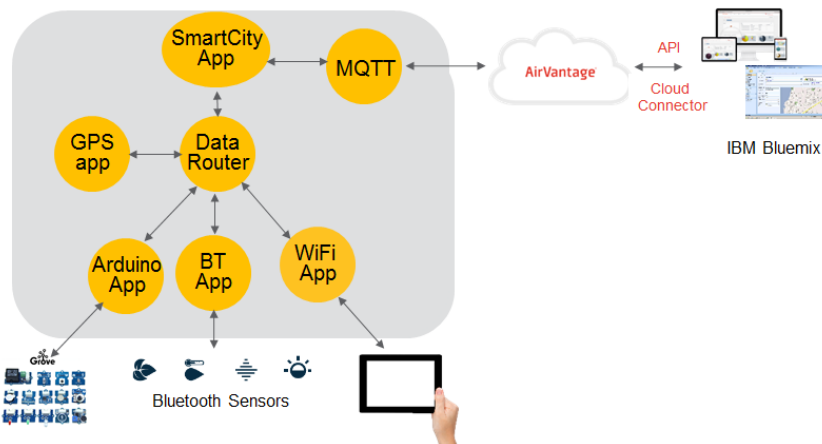
The sensors kit for the hackathon includes the following sensors:

- Grove Studio sensors based : Temperature, Humidity, Dust, Oxygen, Noise, Light, Sound. Tutorial : [4119370\\_mangOH Green Tutorial - Arduino to AirVantage\\_r1.pdf](#)
- Bluetooth based: digital microphone, magnetic sensor, humidity, pressure, accelerometer, gyroscope, magnetometer, object temperature, and ambient temperature <http://www.ti.com/lit/wp/swry023/swry023.pdf>.

## Sierra Wireless IoT Solution for TMForum



## System Architecture



The platform architecture for the hackathon will involve the following applications :

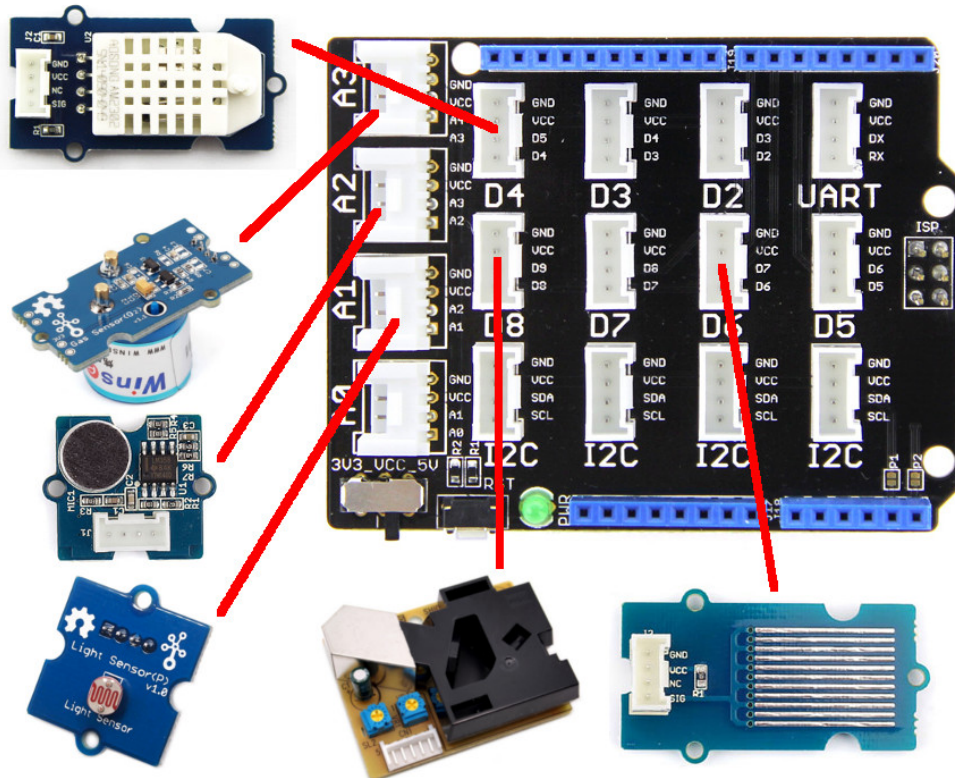
- Arduino app: Gathering & publishing data provided by the Grove studio sensors
- BT app: Gathering & publishing data using the BLE based Sensor Tag sensors
- WiFi app : Used to provide connectivity either in client or AP mode
- Twitter app: To push data to twitter
- Data Router: Provides a way for the apps to publish and subscribe to data as well as pushes it to the AirVantage cloud using MQTT or LWM2M
- MQTT: Provides mechanism to push data to the cloud



- Smart City app: Subscribes to sensor data from BT and Arduino app. Publishes set of chosen data to Cloud

### Sensors used in Hackathon and potential use cases

Sensor	Attached to	Detection	Some Use cases
Dust	mangOH direct only	Particle detection. Reading is ppm	smart city, health for air quality
Oxygen sensor	mangOH direct only	Exact value of Oxygen in the air. Reading is percentage	smart city, health for air quality
Noise	mangOH direct only	Monitor noise pollution from traffic, industry, construction. Reading is analog value	smart city, health for sound pollution
Light	mangOH direct & via Sensor Tag	Measure light as analog reading (direct attach to mangoh) or Lux (BT sensor)	Smart City (Dimm or turn on lights when no is there, save cost)
Accelerometer	Sensor Tag only	3 axis acceleration detection in g	Smart vehicle (accident, speed, container shipment) for smart city, Smart health for accident
Gyroscope	Sensor Tag only	Used for angular measurements. deg/s	Smart vehicle (accident, speed) for smart city, Smart health for accident
Barometer	Sensor Tag only	For height detection. Value is in hectopascal	Smart vehicle (dead reckoning applications etc.), weather Smart health for wearables/ weather etc.
Humidity	mangOH direct & via Sensor Tag	Humidity measurement in %	Smart city & smart health environmental
Object temperature	Sensor Tag only	Infra Red Temperature measurement in C	Smart city infra red temperature monitoring
Water	mangOH direct	Water presence. Reading is boolean	Smart city water management
Ambient temperature	mangOH direct & via Sensor Tag	Temperature measurement in C	Smart city & smart health environmental
Digital Microphone -> actuated	Sensor Tag only	Actuation of alert via sound	Smart city, smart health alarm
LEDs -> actuated	Sensor Tag only	Actuation of alert via light	Smart city, smart health alarm



## Using the Hackathon Platform

There is an incompatibility between NetworkManager in Ubuntu 16.04 and the ECM interface on the mangOH Green. In order to bring up the interface, run the command `sudo ifconfig enx1e1e131e3b4a up 192.168.2.3`

Sierra will provide 16 completely assembled kits with the following:

- 16 mangOH Open Hardware platforms
- 16 Bluetooth/WiFi IoT Expander cards
- 16 TI Sensor Tag 2.0 will be provide
- 10 Arduino Sensor Grove Shields

Each individual mangOH platform will be preassembled for you:

- 10 units will have all the Arduino sensors plugged in. The Arduino app will automatically start. Tutorial [4119370\\_mangOH Green Tutorial - Arduino to AirVantage\\_r1.pdf](#)
- One TI Sensor Tag Bluetooth sensor will be in every box. Each unit will be automatically pre-associated to one sensor tag. In case you run into issue, please the sensor tag into scan mode by pressing the power button. Then to connect mangOH to sensor tag run the following commands on console:
  - `root@swi-mdm9x15:~# hcitool lescan` ( this will show you MAC address of sensor tag)
  - `root@swi-mdm9x15:~# config set bleSensorInterface/sensorMac < MAC address>`
  - `root@swi-mdm9x15:~# config get bleSensorInterface/sensorMac` (confirms the attachment)

- Set the AirVantage on configuration database for the unit

```
root@swi-mdm9x15:~# config set sensorToAirVantage/mqttBrokerHost eu.airvantage.net
```

```
root@swi-mdm9x15:~# config set sensorToAirVantage/mqttBrokerPort 1883 int
```

```
root@swi-mdm9x15:~# config set sensorToAirVantage/mqttBrokerPassword SWI
```

- For next steps, you need to get IMEI of board: `root@swi-mdm9x15:~# cm info`
- Your unit is pre-registered to AirVantage
- Login to your AirVantage account :

```
url: eu.airvantage.net
```

```
user name: mangoh_handson@sierrawireless.com
```

```
password: swis2016!
```

Look for your unit name and if you want to look at data follow section 5 of [4119370\\_mangOH Green Tutorial - Arduino to AirVantage\\_r1.pdf](#)

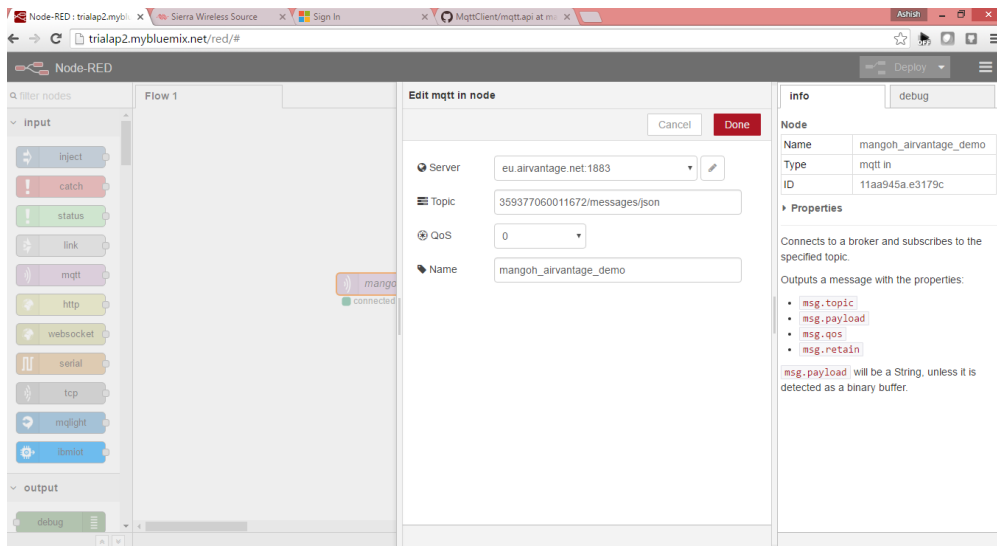
- Creating a Cellular/WiFi Gateway : follow the tutorial section 5 [4119378\\_mangOH Green Tutorial - Wi-Fi\\_r1.pdf](#)
- Connecting to BlueMix: <http://www.ibm.com/cloud-computing/bluemix/>

Get an account (follow standard process at bluemix site)

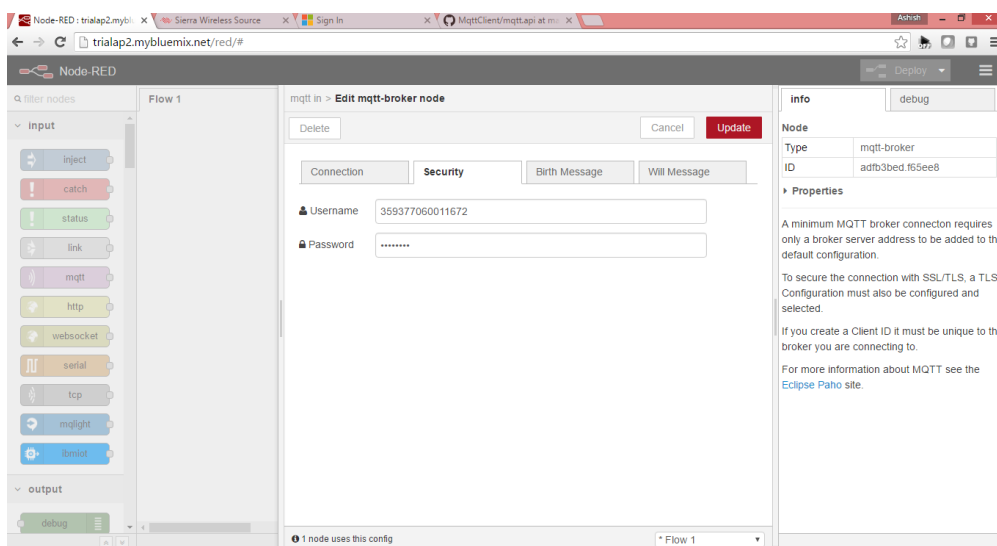
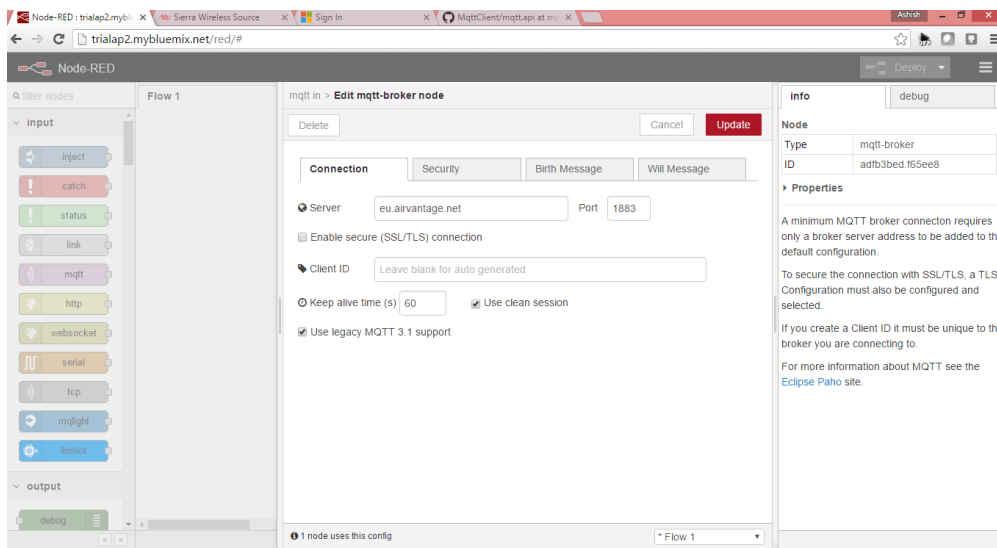
Follow steps (TBC) to setup a nodeRed system.

In the system, in order to get data from mangOH you need to enter the following:

- Server: [eu.airvantage.net](http://eu.airvantage.net) port 1883
- Topic : IMEI/messages/json
- QoS: 0 (Default)
- Name: {Any desired text string} This is a text identifier for the MQTT Input Node



Match your input values for the following screen



- User name : IMEI# of your module (you got that from CM info command)



- password: SWI

Some users have reported problems consuming the payload once it is in Node-RED. The javascript below should help to avoid a couple of pitfalls.

```
var msgStr = '{"payload": "{\\"arduino.water\\": true}"}';  
var msgObj = JSON.parse(msgStr);  
var payloadStr = msgObj["payload"];  
var payloadObj = JSON.parse(payloadStr);  
alert(payloadObj["arduino.water"]);
```

For direct response to questions, please register on the forum at mangoh.io ( <http://forum.mangoh.io/>)