Develop a “smart” wildlife asset tracking system

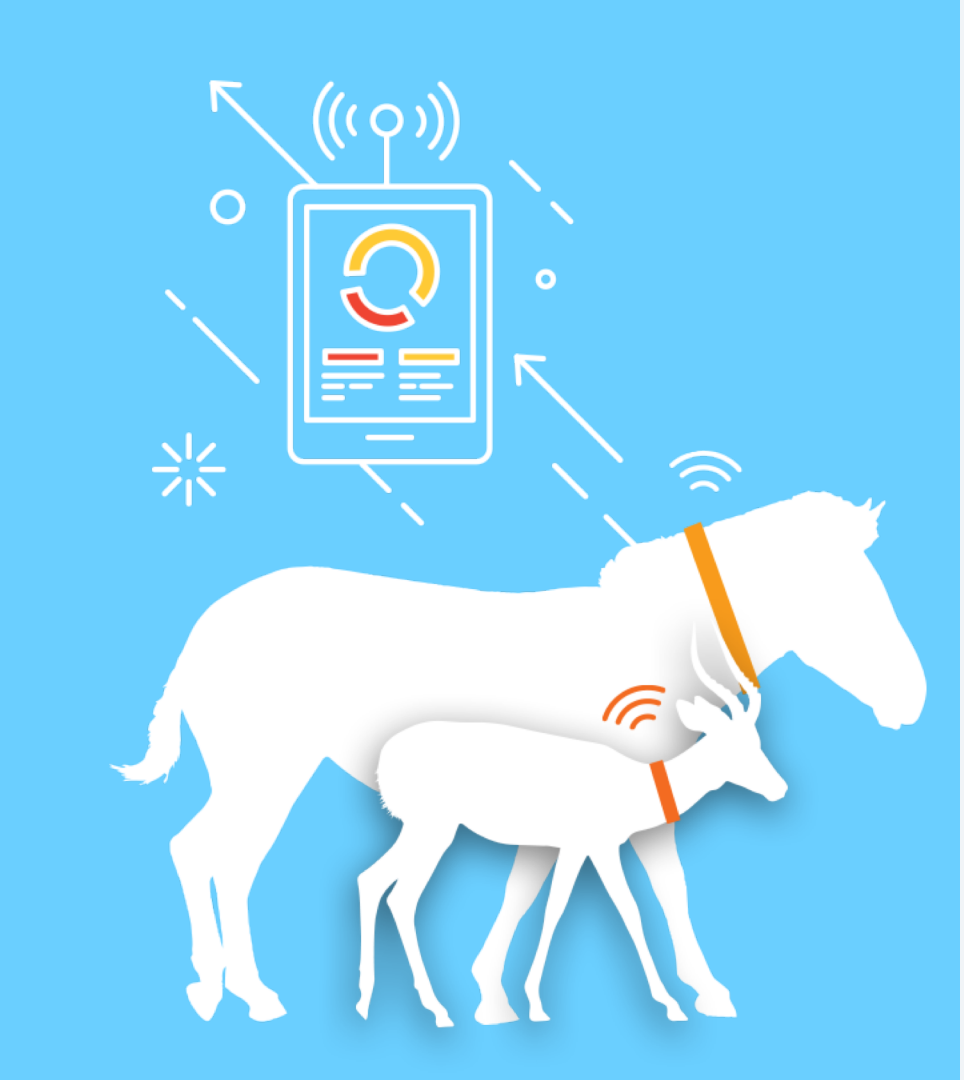
Kalonji Bankole
Developer Advocate

IBM **Developer**

IBM

Problem statement

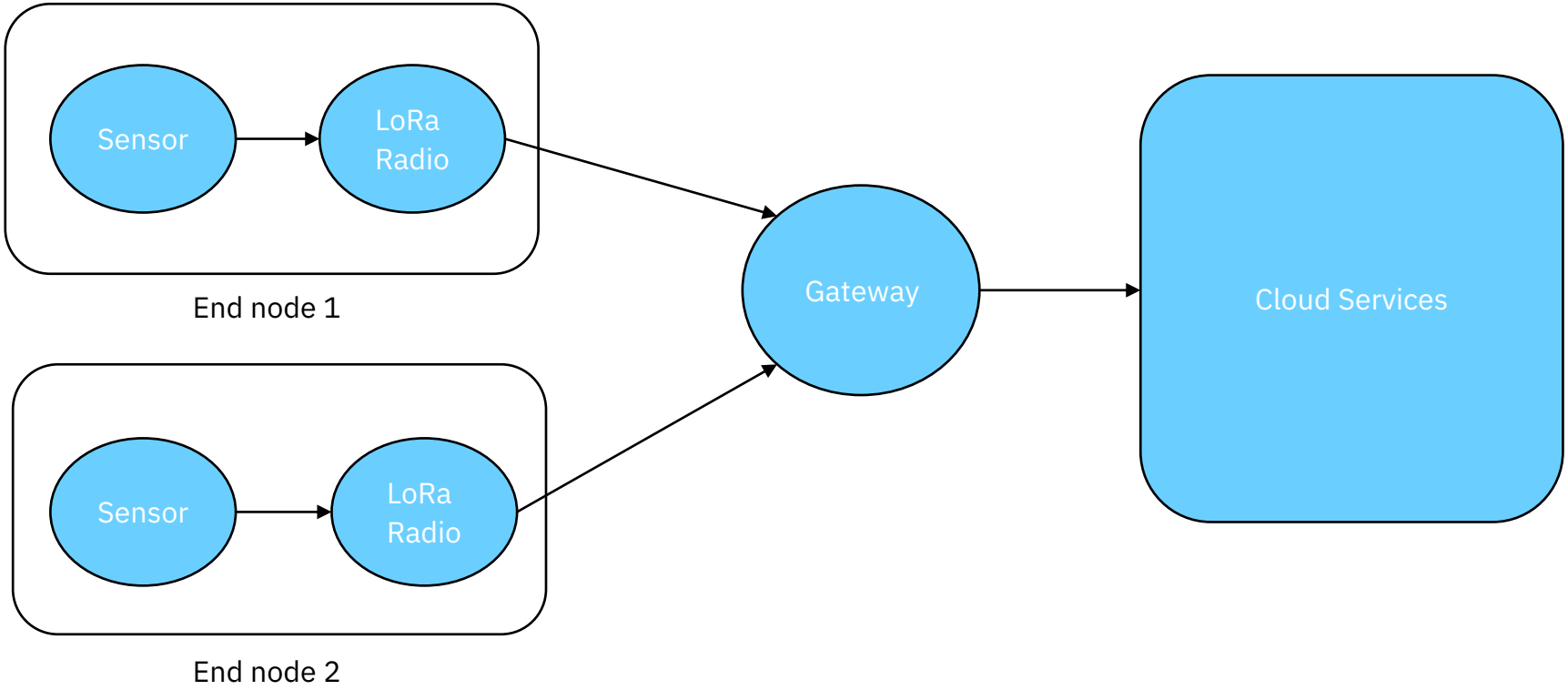
- Poachers attack thousands of endangered animals every year
- Researchers have found that animals move differently based on the type of perceived threat.
- Goal is to provide an early warning system to notify rangers when poachers are in the area



Implementation

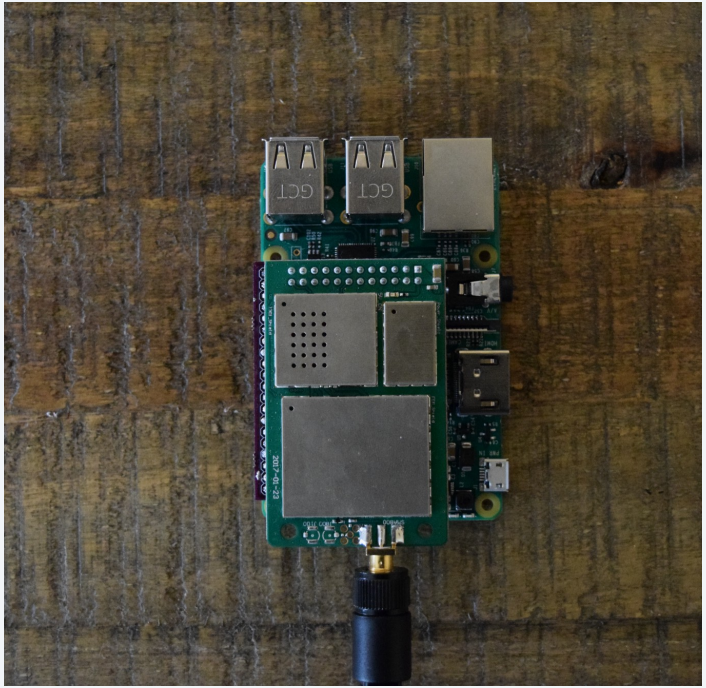
- Wireless network
- Wearable sensors
- Cloud Analytics services

High Level Architecture

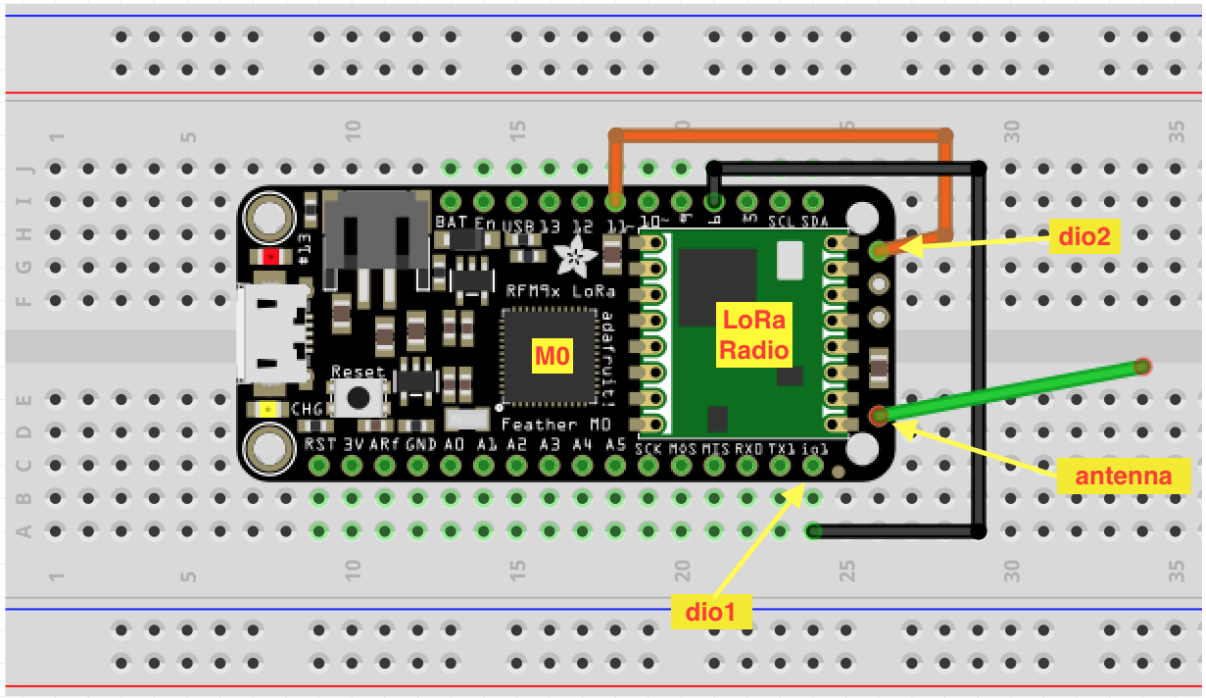


LoRaWAN Gateway

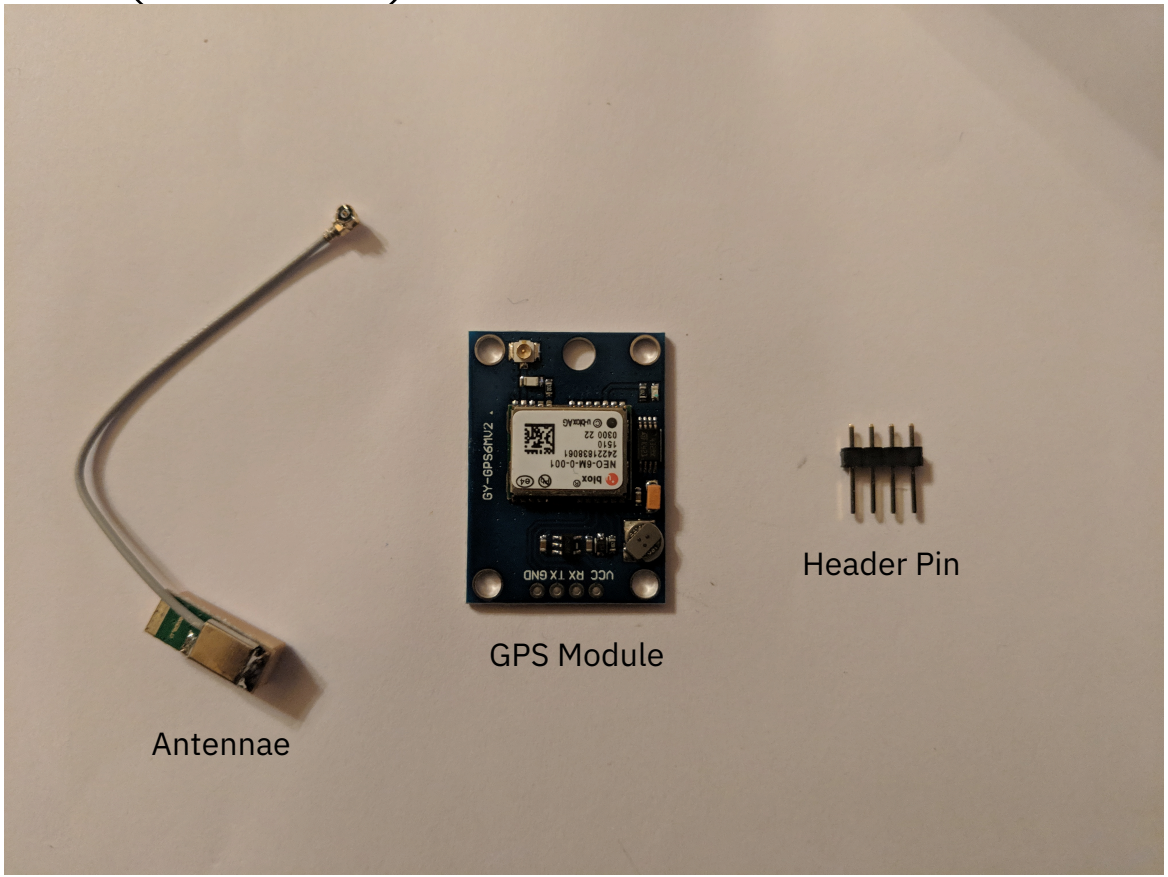
- Raspberry Pi
- LoRaWAN Concentrator Board
- 915 MHz antennae
- PCB Shield



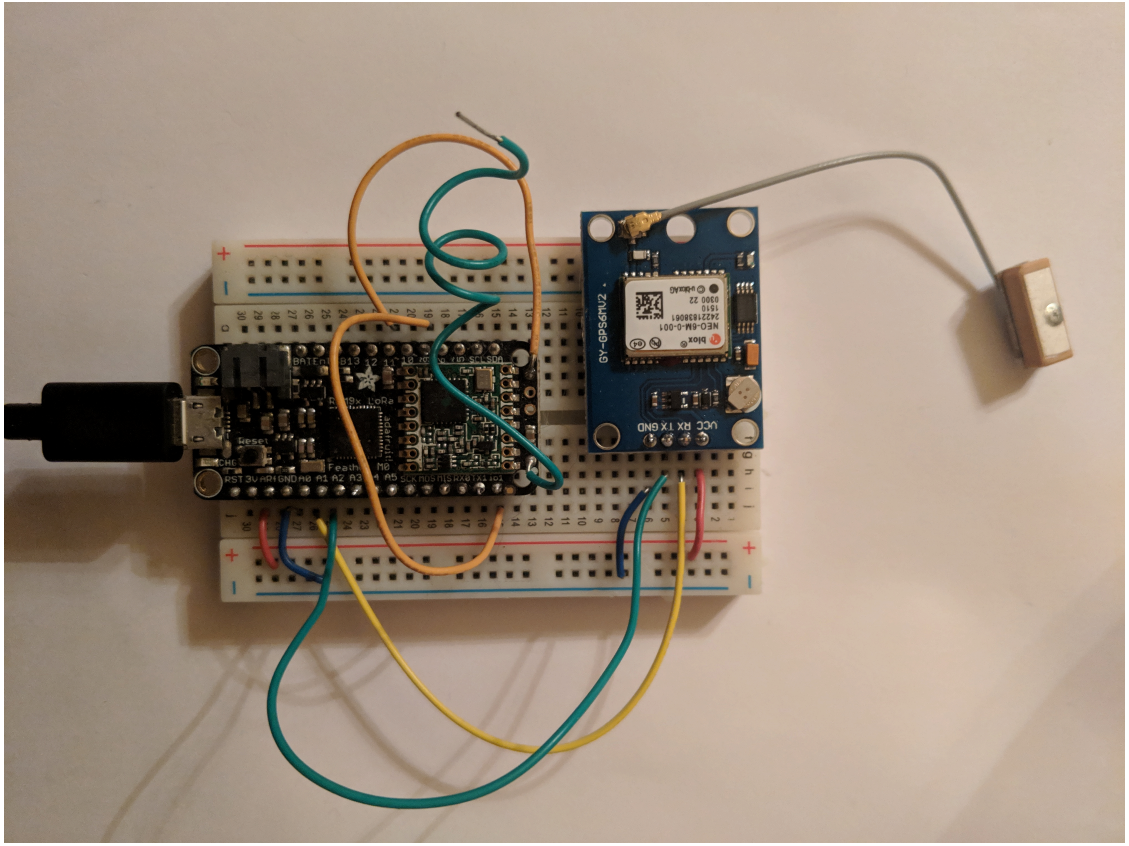
LoRaWAN End Node (Base)



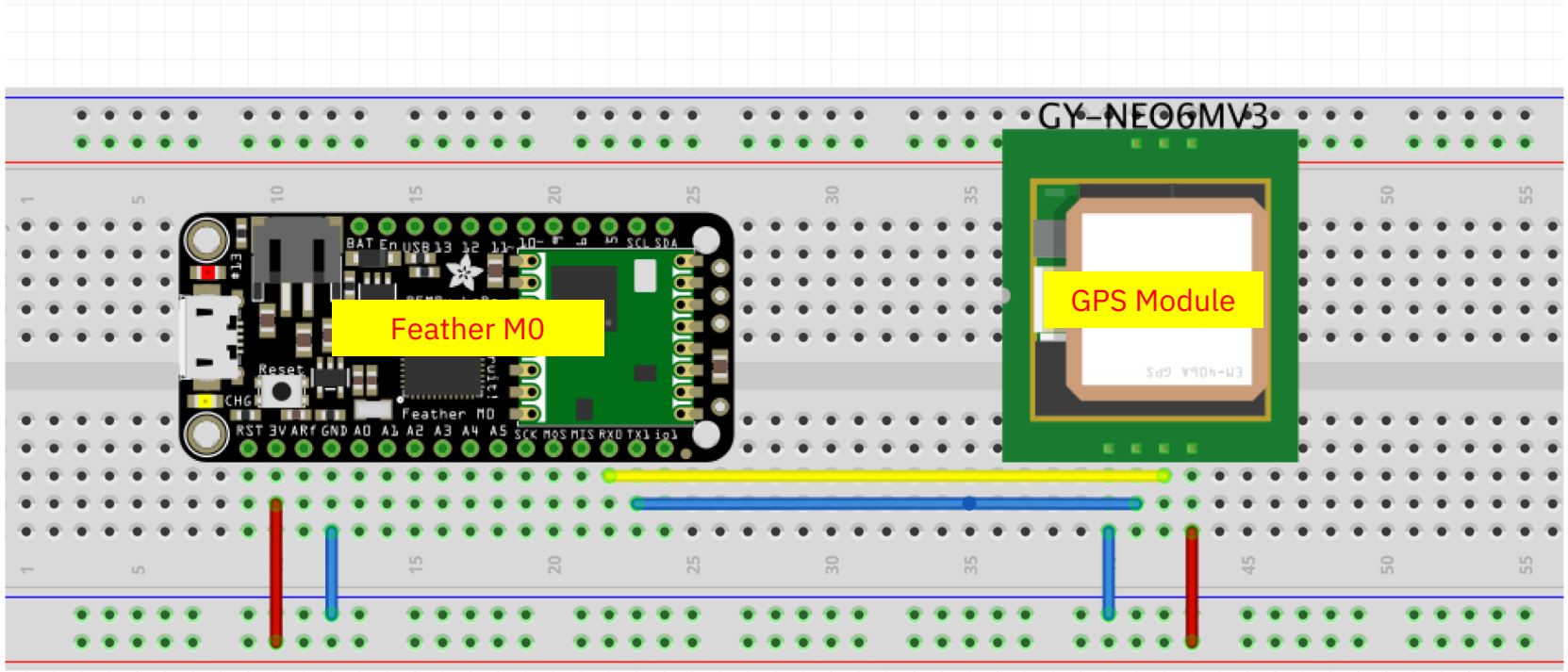
GPS Module (NEO-6M)



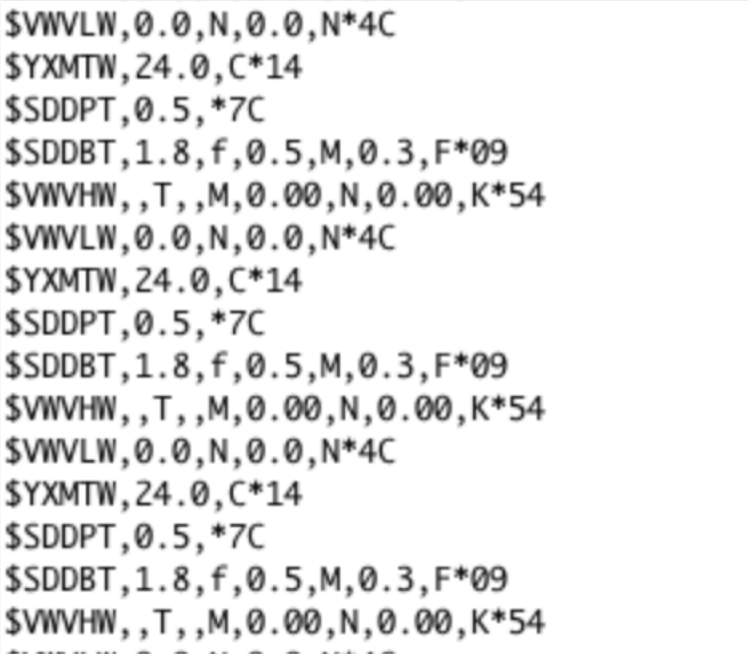
LoRaWAN End Node (GPS)



LoRaWAN End Node (GPS)



GPS Raw Output



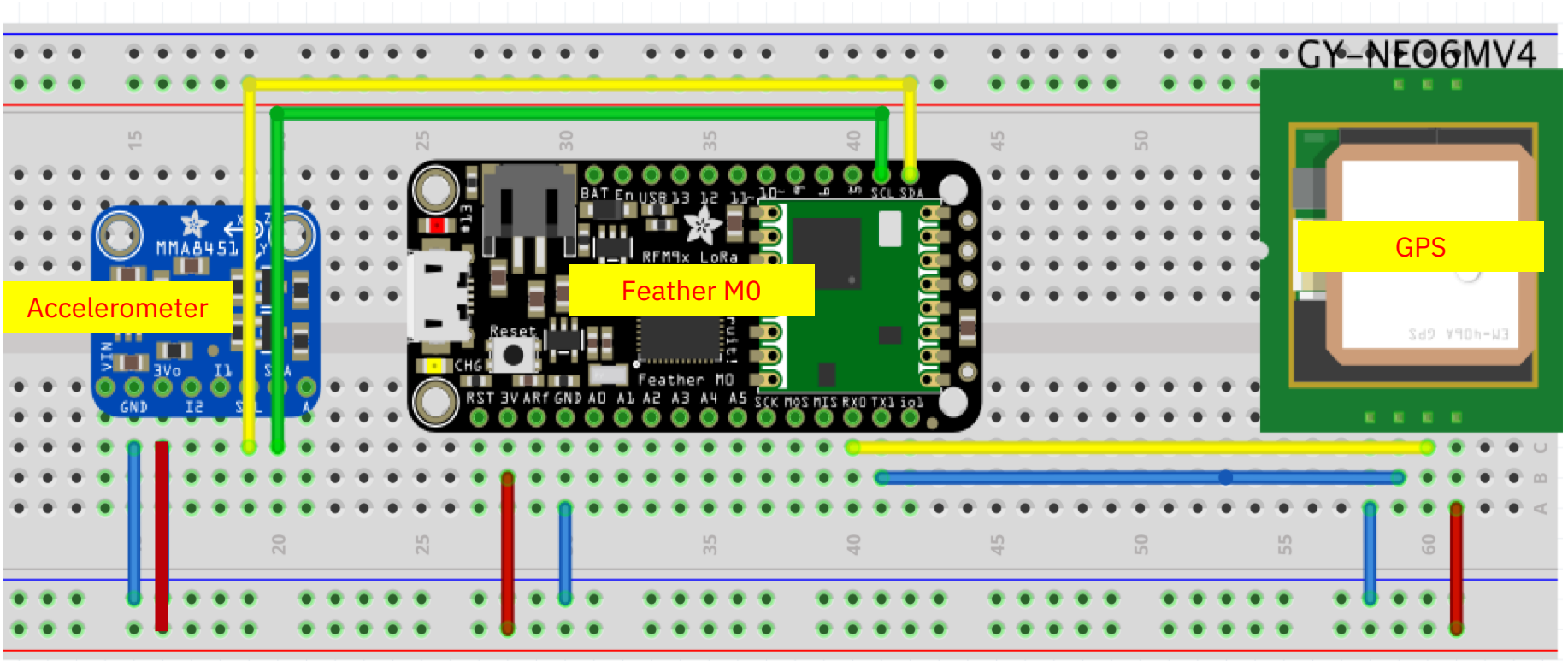
```
$VWVLW,0.0,N,0.0,N*4C
$YXMTW,24.0,C*14
$SDDPT,0.5,*7C
$SDDBT,1.8,f,0.5,M,0.3,F*09
$VWVHW,,T,,M,0.00,N,0.00,K*54
$VWVLW,0.0,N,0.0,N*4C
$YXMTW,24.0,C*14
$SDDPT,0.5,*7C
$SDDBT,1.8,f,0.5,M,0.3,F*09
$VWVHW,,T,,M,0.00,N,0.00,K*54
$VWVLW,0.0,N,0.0,N*4C
$YXMTW,24.0,C*14
$SDDPT,0.5,*7C
$SDDBT,1.8,f,0.5,M,0.3,F*09
$VWVHW,,T,,M,0.00,N,0.00,K*54
```

Example NMEA Output

NeoGPS (Code Snippet)

```
#include <NMEAGPS.h> // Load NeoGPS lib
static NMEAGPS gps; // This parses the GPS characters
gps_fix fix; // This holds on to the latest values
void loop()
{
    while (gps.available( gps_port )) {
        fix = gps.read();
        Serial.print( fix.latitude() );
        Serial.println( fix.longitude() );
    }
}
```

LoRaWAN End Node (GPS + Accelerometer)



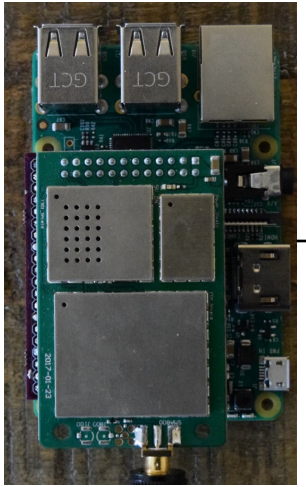
End Node Collar

Photos from @markusvankempen – MVN, IBM, and Wageningen University

Field Deployment

Animal with tagged collar

LoRaWAN™

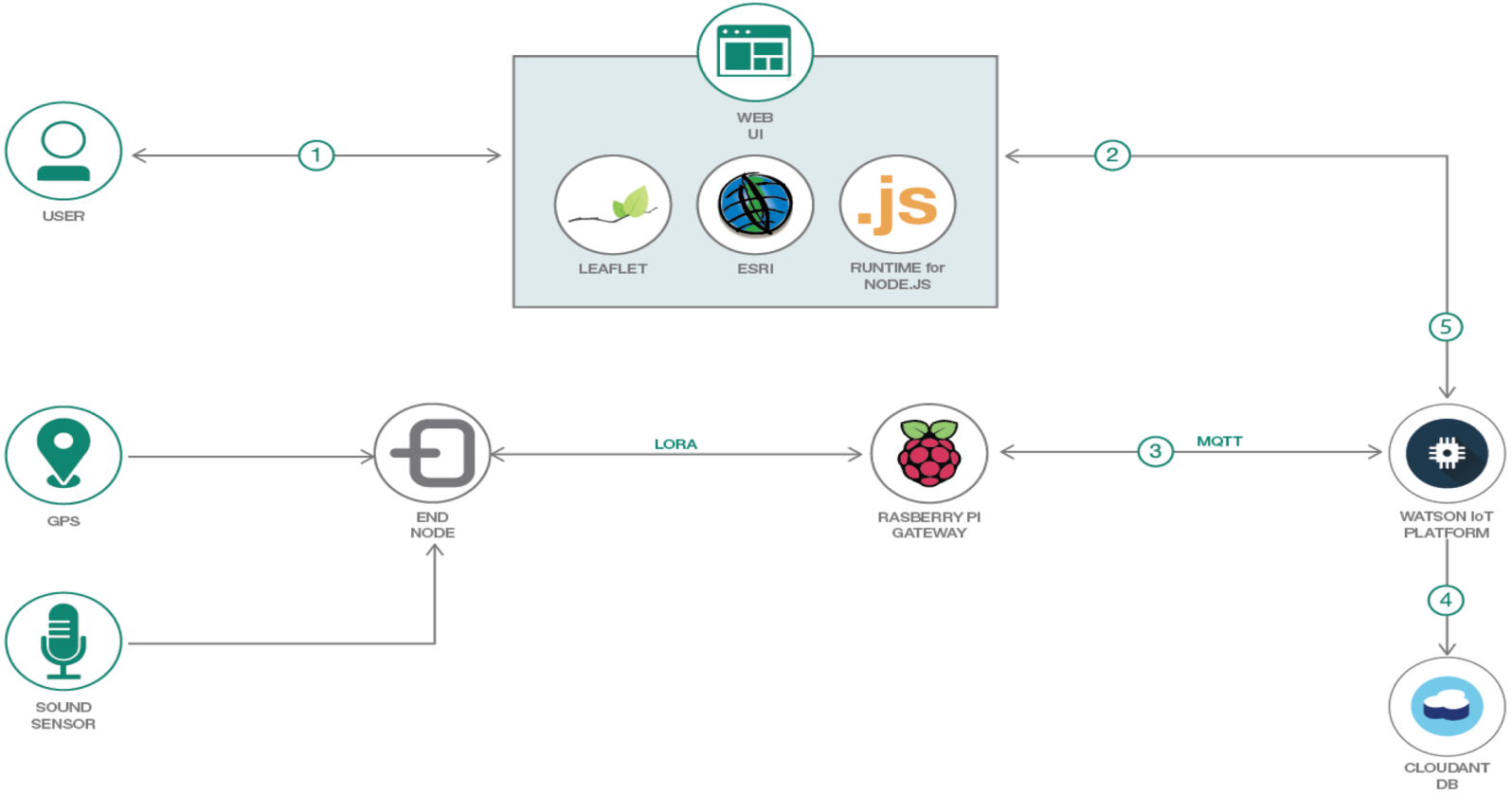


Raspberry Pi Gateway

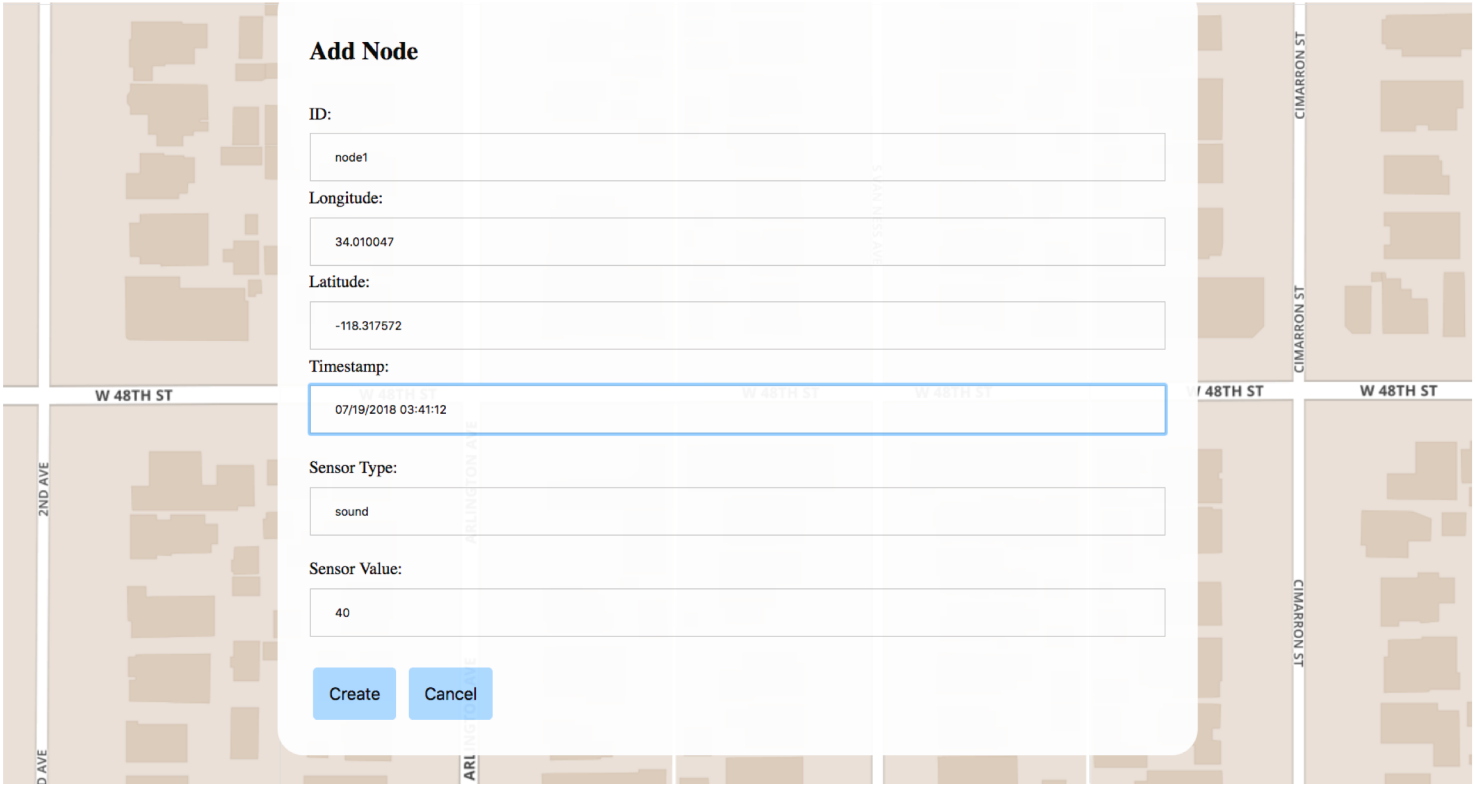
MQTT

IBM Cloud

Sample App Architecture



Sample App


Add Node
ID:
node1
Longitude:
34.010047
Latitude:
-118.317572
Timestamp:
07/19/2018 03:41:12
Sensor Type:
sound
Sensor Value:
40
Create
Cancel
W 48TH ST
W 48TH ST
CIMARRON ST
CIMARRON ST
2ND AVE

Sample App

```
var initTrackableAsset = function(id, long, lat, time, type = undefined, val = undefined) {
  console.log("initializing asset: " + id)
  var node = {
    marker: L.marker([long, lat]).addTo(mymap).bindPopup("<b>" + id +"</b>").openPopup(),
    points: [ new L.LatLng(long, lat) ],
    timestamps: [ time ]
  }
  if (type) {
    var node1 = Object.assign ({
      circle: L.circle([long, lat], val, {
        color: 'red',
        fillColor: '#f03',
        fillOpacity: 0.1
      }).addTo(mymap).bindPopup("LoRA Node: " + id)
    }, node)
    assets[id] = node1
```

Sample App

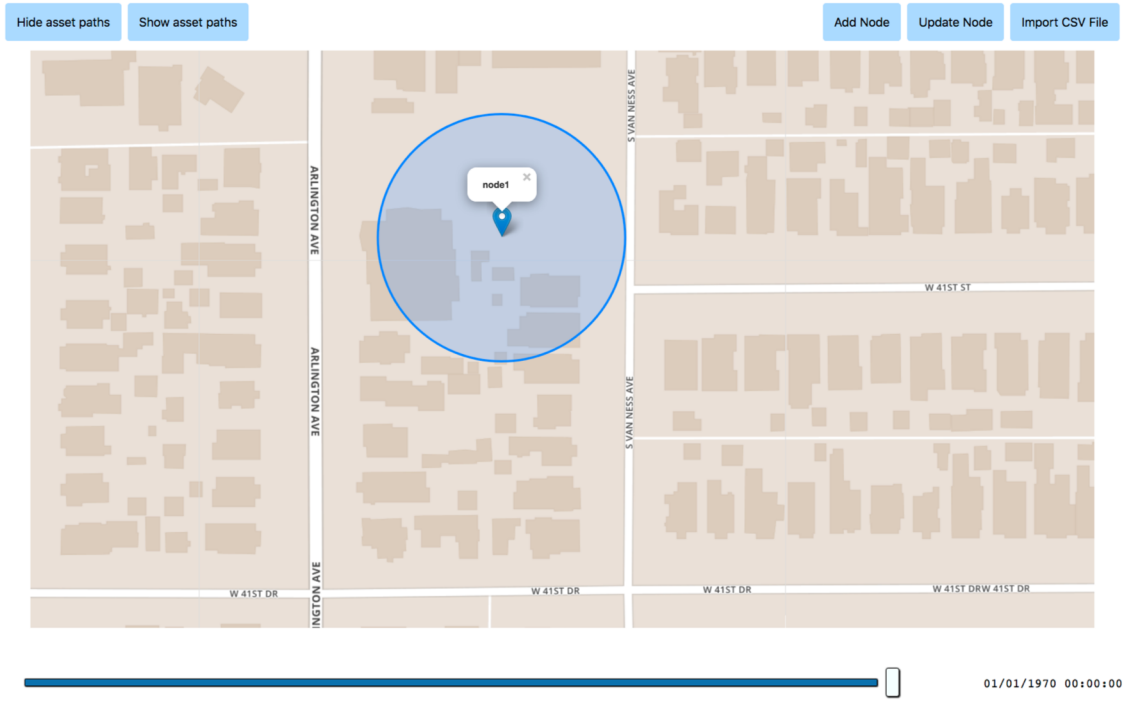
```
mqtt_pub -v -i "a:${IOT_ORG_ID}:client_pub1" -u "${IOT_API_KEY}" -P "${IOT_AUTH_TOKEN}" -h
    "d" : {
    "node_id": "node2",
    "lat": "-118.317392",
    "long": "34.100057",
    "timestamp": "2018-06-30T07:10:55.174Z",
    "sensor": {
      "sound": "72"
    }
  }
}'
```

Sample App

```
var initMQTTClient = function(mqttCreds) {
  var watson_channel = 'iot-2/type/' + mqttCreds.IOT_DEVICE_TYPE + '/id/' + mqttCr
  var cleanSession = true;

  var subscribeOptions = {
    onSuccess: function() {
      console.log("subscription set")
      mqttClient.onMessageArrived = function (messageObj) {
        var message = JSON.parse(messageObj.payloadString).d
        console.log( message)
        if ( ! assets[message['node_id']] ) {
          initTrackableAsset(message['node_id'], message['long'], message['lat'] )
        }
        if (message['sensor']) {
          updateTrackableAsset(message['node_id'], message['long'], message['lat']
```

Sample App

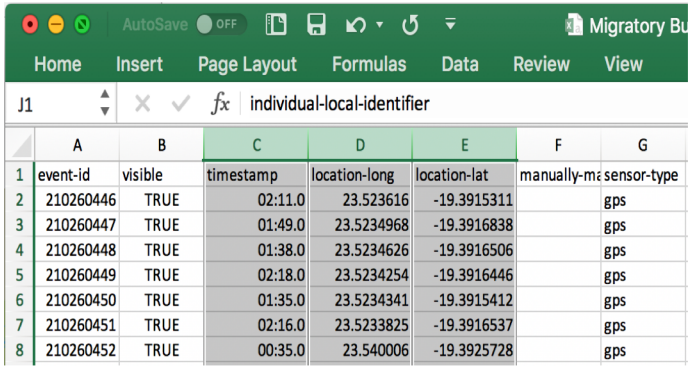


node1

Location: LatLng(34.010047, -118.317572)

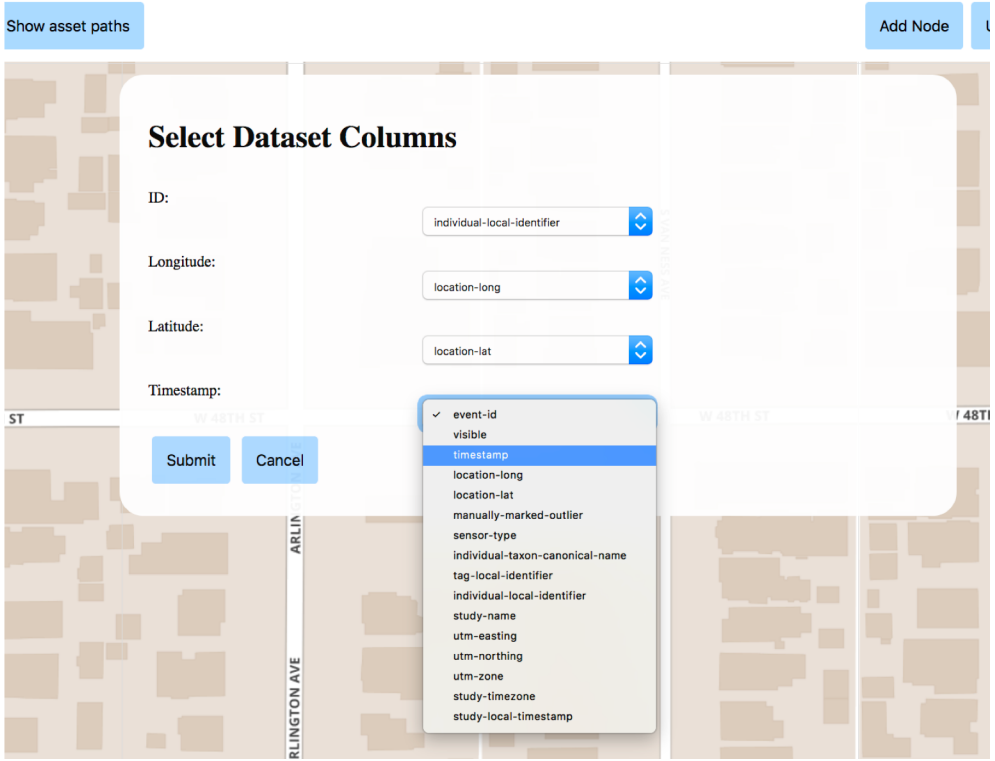
Last Update: 1534274056002

Sample App

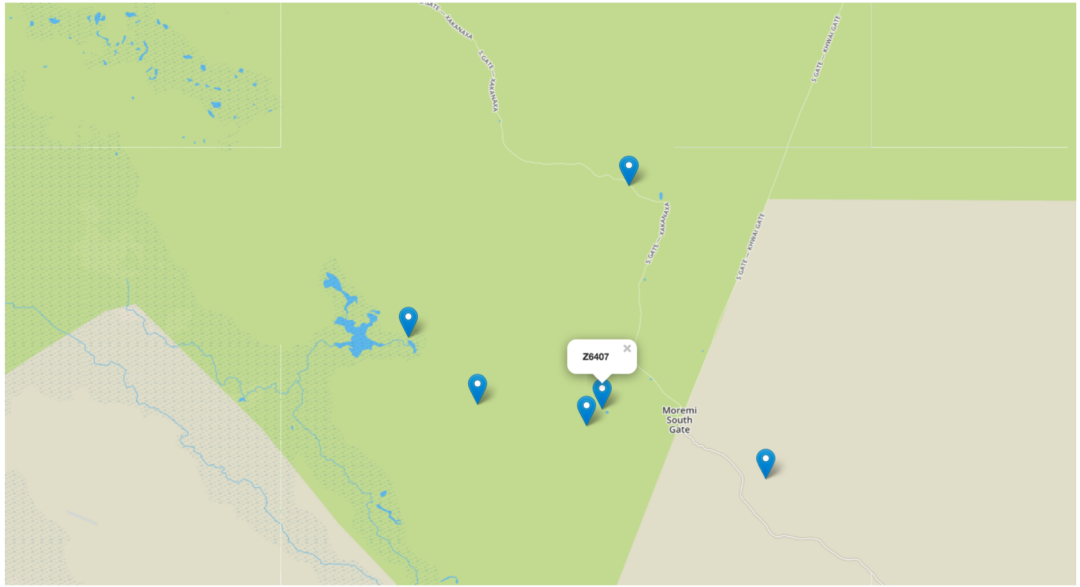


J1 | individual-local-identifier

	A	B	C	D	E	F	G
1	event-id	visible	timestamp	location-long	location-lat	manually-ma	sensor-type
2	210260446	TRUE	02:11.0	23.523616	-19.3915311		gps
3	210260447	TRUE	01:49.0	23.5234968	-19.3916838		gps
4	210260448	TRUE	01:38.0	23.5234626	-19.3916506		gps
5	210260449	TRUE	02:18.0	23.5234254	-19.3916446		gps
6	210260450	TRUE	01:35.0	23.5234341	-19.3915412		gps
7	210260451	TRUE	02:16.0	23.5233825	-19.3916537		gps
8	210260452	TRUE	00:35.0	23.540006	-19.3925728		gps



Sample App



01/01/1970 00:00:00

Z3864

Location:
-20.6066866,24.995666

Last Update: 210263166

Z3743

Location:
-19.3899775,23.5299457

Last Update: 209299877

Z3866

Location:
-19.4310398,23.5962208

Last Update: 209315664

Z6402

Location:
-20.3551241,24.508933

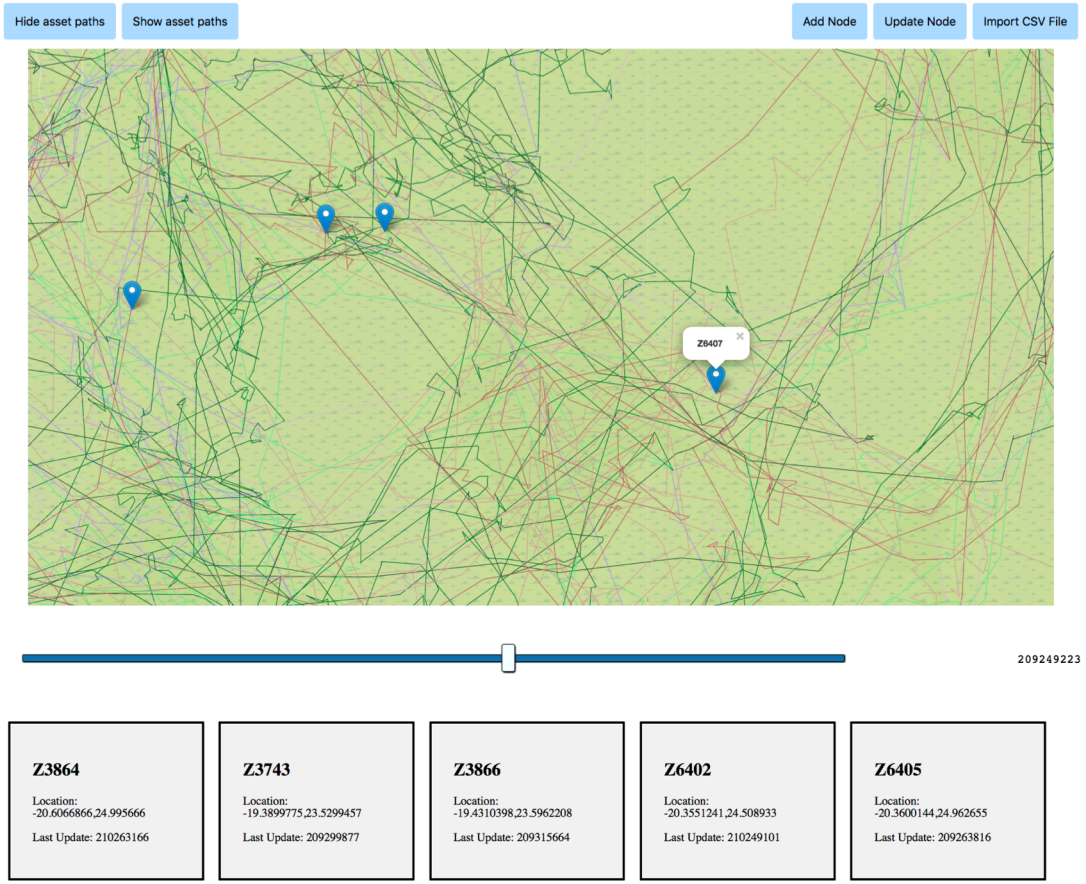
Last Update: 210249101

Z6405

Location:
-20.3600144,24.962655

Last Update: 209263816

Sample App


Hide asset paths
Show asset paths
Add Node
Update Node
Import CSV File
Z6407
209249223
Z3864
Location:
-20.6066866,24.995666
Last Update: 210263166
Z3743
Location:
-19.3899775,23.5299457
Last Update: 209299877
Z3866
Location:
-19.4310398,23.5962208
Last Update: 209315664
Z6402
Location:
-20.3551241,24.508933
Last Update: 210249101
Z6405
Location:
-20.3600144,24.962655
Last Update: 209263816

Sample App

```
var drawAssetPath = function( id ) {
    assets[id]['path'] = new L.Polyline( assets[id]['points'], {
        // color: 'red',
        color: "rgb(" + genRandomNums() + ")",
        weight: 1,
        opacity: 0.8,
        smoothFactor: 0
    })
    assets[id]['path'].addTo(mymap);
}
```

References

https://github.com/ibm/iot-mapping

https://www.slideshare.net/MarkusVanKempen/animal-tracking-using-watson-iot-and-lorawan

https://www.wur.nl/en/project/SmartParks-using-technology-for-anti-poaching.htm