IoT Challenge 2018- Clustering and Twitter flows Intro

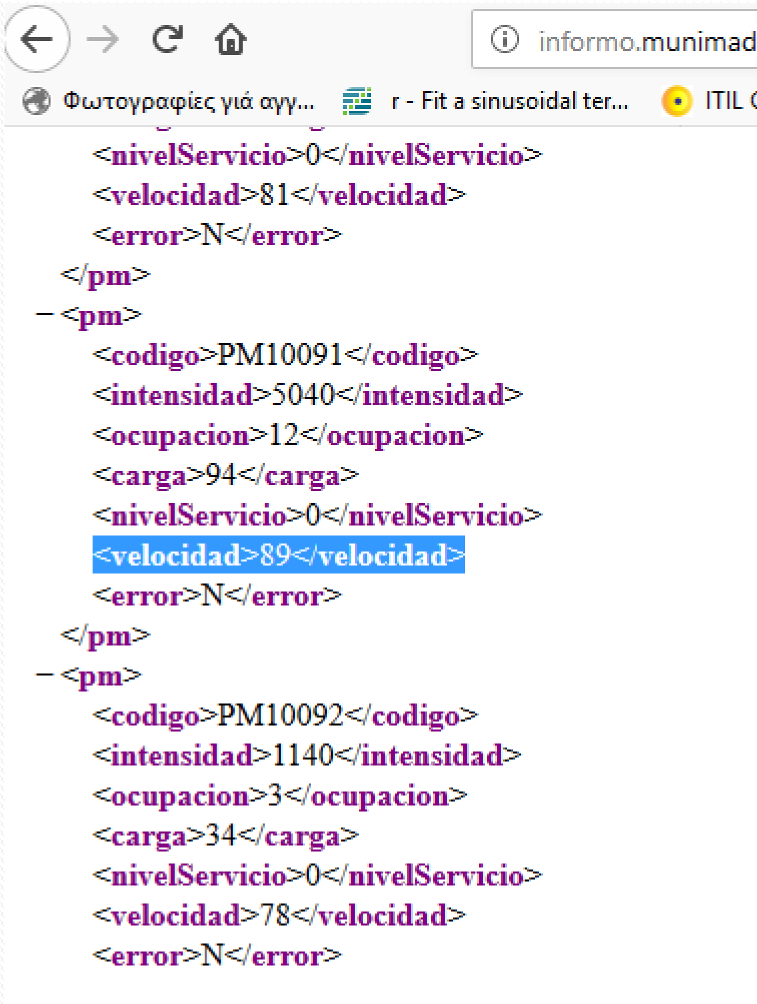
16-17 March 2018

Clustering flow

- Objective: implement a simple generic L architecture in Node-RED
- Specific example: Exploit street speed data from Madrid and group speeds in two states (good and bad state)
- Store and process, extract high and low speed clusters for a single point
- Can be generalized for all points, extended to use a database or used in a different data source

Clustering flow

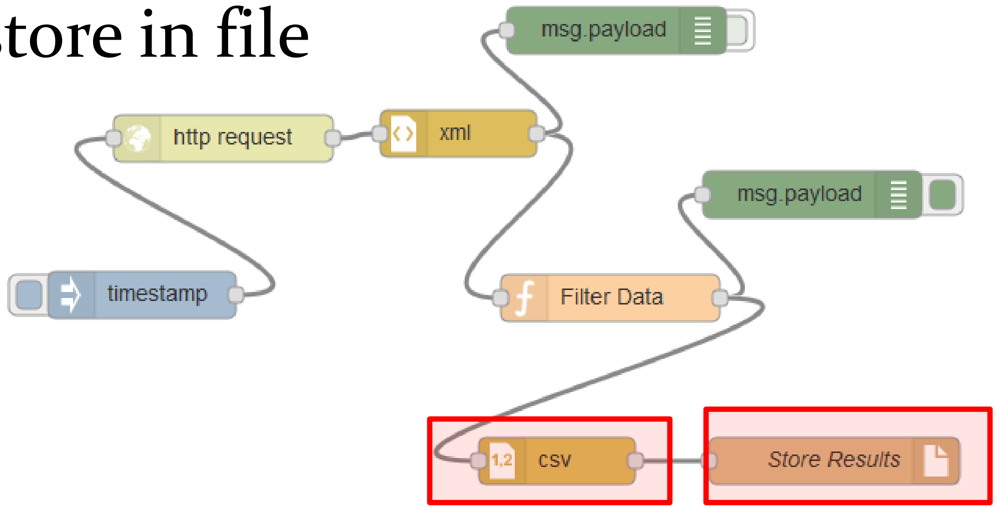
- Exploit open street data from madrid
  - http://informo.munimadrid.es/informo/tmadrid/pm.xml
  - Be careful, not all PMs have speed (velocidad field) in them



```
    <nivelServicio>0</nivelServicio>
    <velocidad>81</velocidad>
    <error>N</error>
  </pm>
- <pm>
    <codigo>PM10091</codigo>
    <intensidad>5040</intensidad>
    <ocupacion>12</ocupacion>
    <carga>94</carga>
    <nivelServicio>0</nivelServicio>
    <velocidad>89</velocidad>
    <error>N</error>
  </pm>
- <pm>
    <codigo>PM10092</codigo>
    <intensidad>1140</intensidad>
    <ocupacion>3</ocupacion>
    <carga>34</carga>
    <nivelServicio>0</nivelServicio>
    <velocidad>78</velocidad>
    <error>N</error>
```

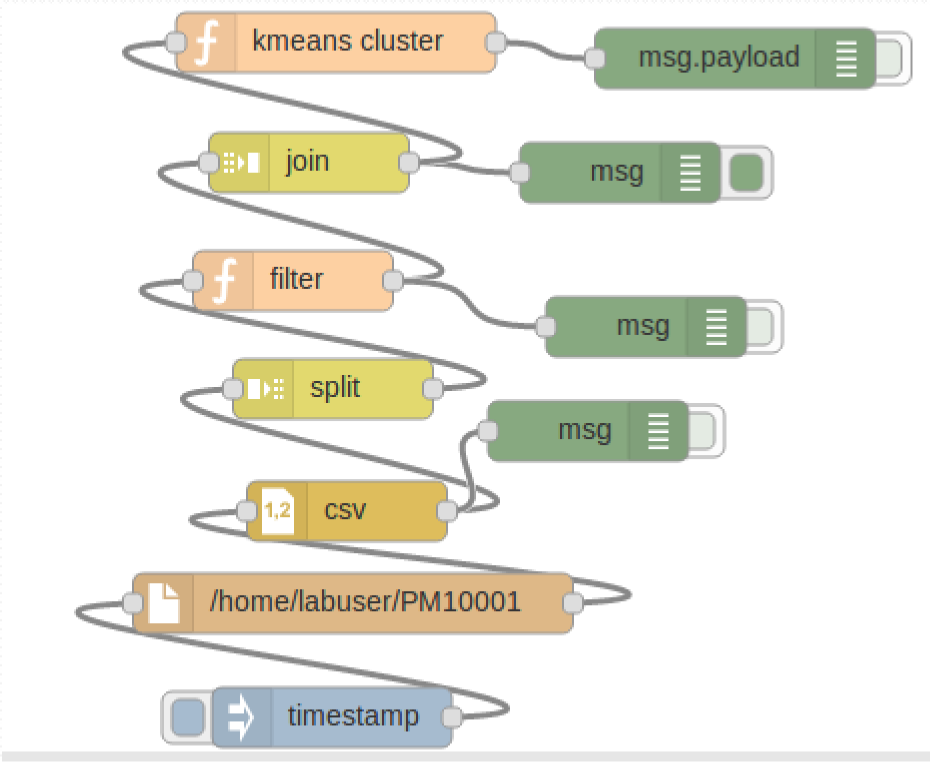
Clustering flow horizontal L architecture

- Http GET call to Madrid for data
- Transformation from XML to JSON
- Filter to get only the point we need
  - Check returned JSON from Madrid for the structure of the data!!
- Transform to csv and store in file



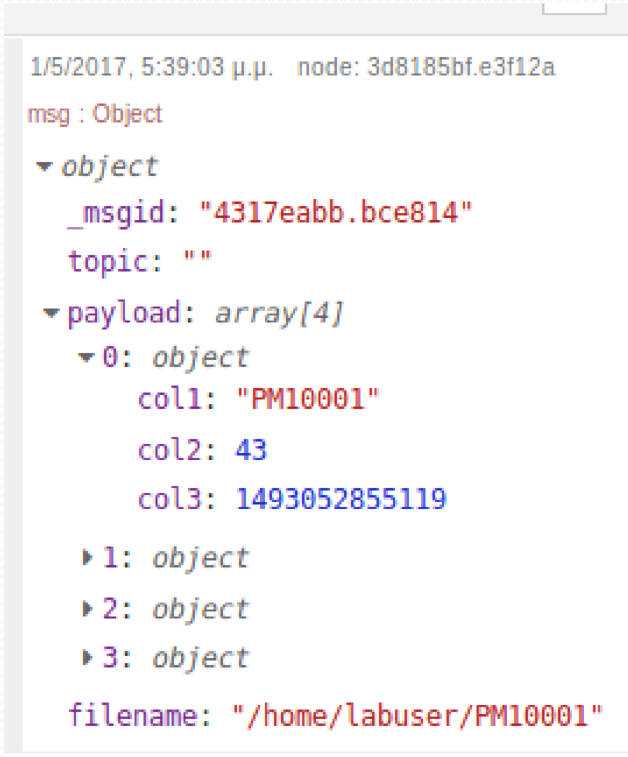
Clustering flow- Vertical part

- Usage of external node.js function 'clusters' in kmeans cluster function node
  - Be sure to include it in settings.js file
- This needs data inserted in a specific form
  - Array of arrays of data points
  - [[2,4],[3,5]...]
- Data points come as array of objects from csv file
  - Transformed in array of arrays in the split-filter-join part of the flow
- Read file produced from the previous flow

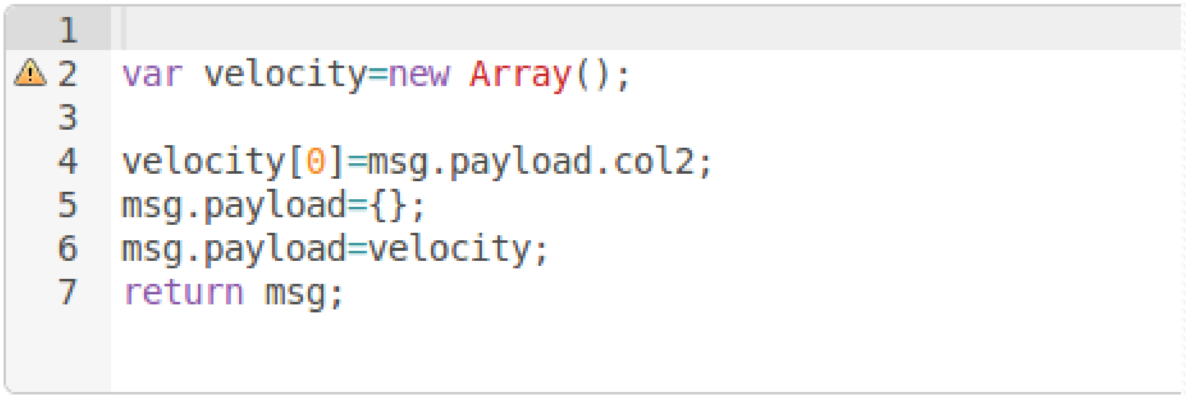


Object to array code in filter function node

- If more dimensions are needed in the clustering case, one can add more points in the array



```
1/5/2017, 5:39:03 μ.μ.   node: 3d8185bf.e3f12a
msg : Object
▾ object
   _msgid: "4317eabb.bce814"
   topic: ""
  ▾ payload: array[4]
    ▾ 0: object
        col1: "PM10001"
        col2: 43
        col3: 1493052855119
    ▸ 1: object
    ▸ 2: object
    ▸ 3: object
   filename: "/home/labuser/PM10001"
```



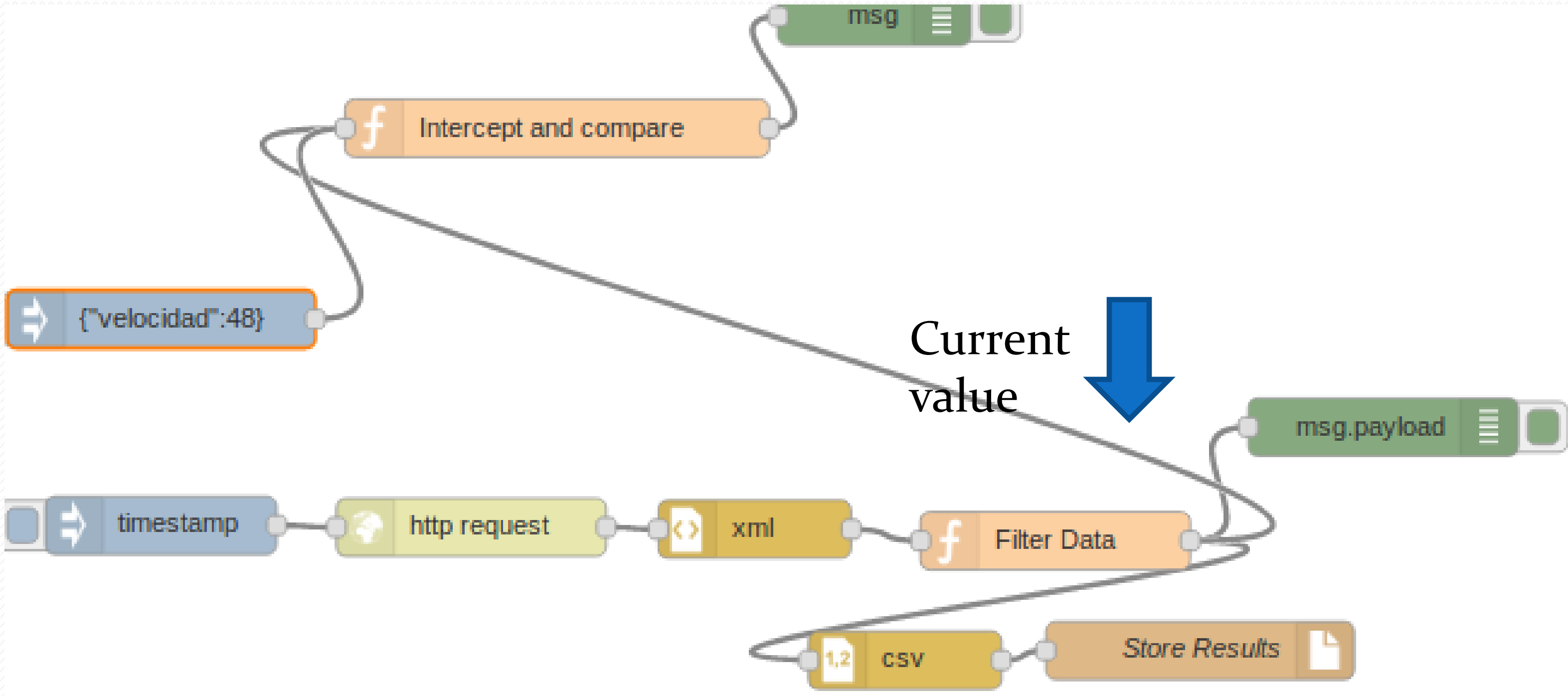
```

var velocity=new Array();

velocity[0]=msg.payload.col2;
msg.payload={};
msg.payload=velocity;
return msg;
```

Clustering flow- diagonal part

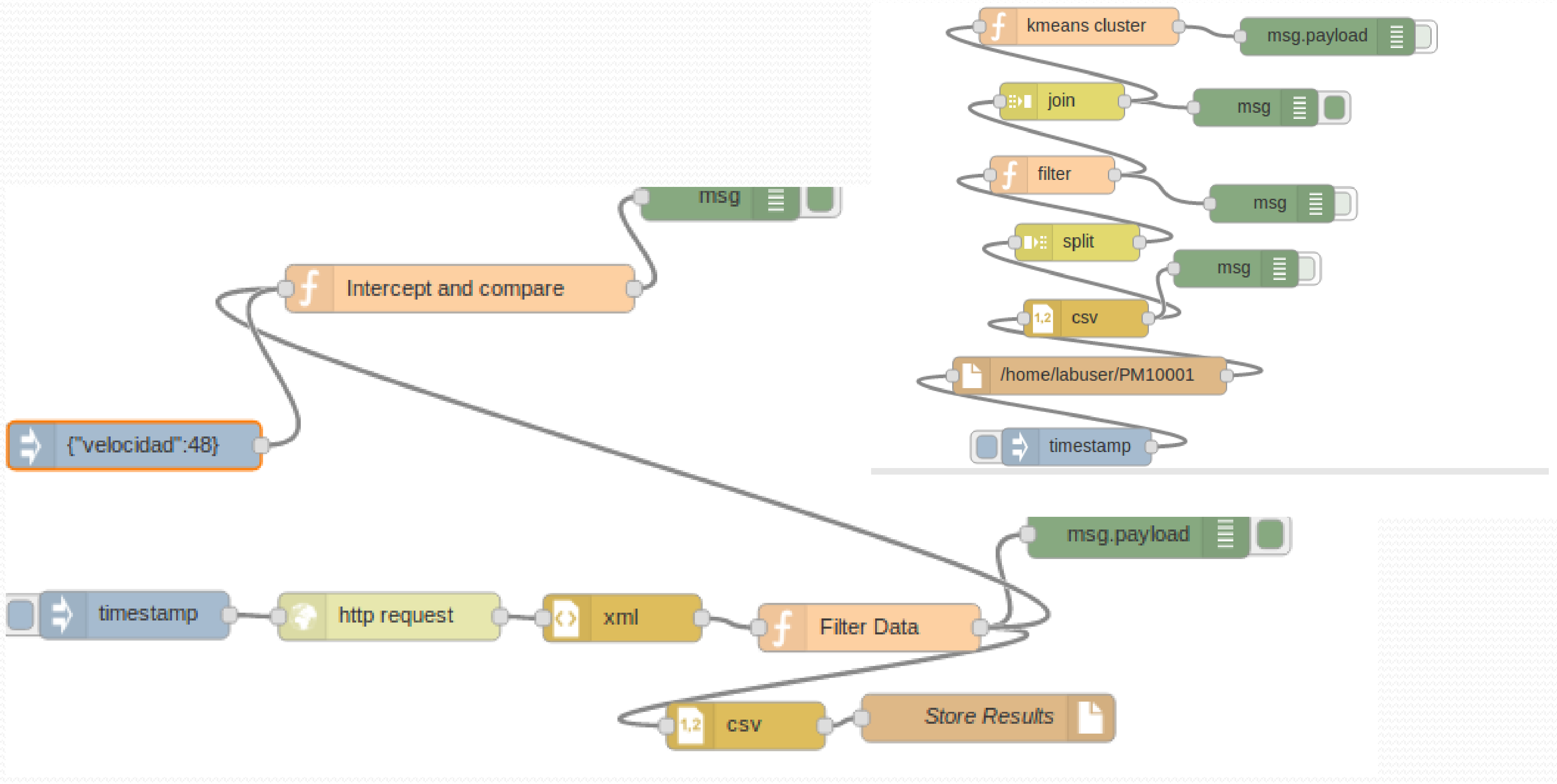
- Retrieve real time value and compare with cluster centroids
  - Current value is already extracted from the horizontal flow



Intercept and compare function

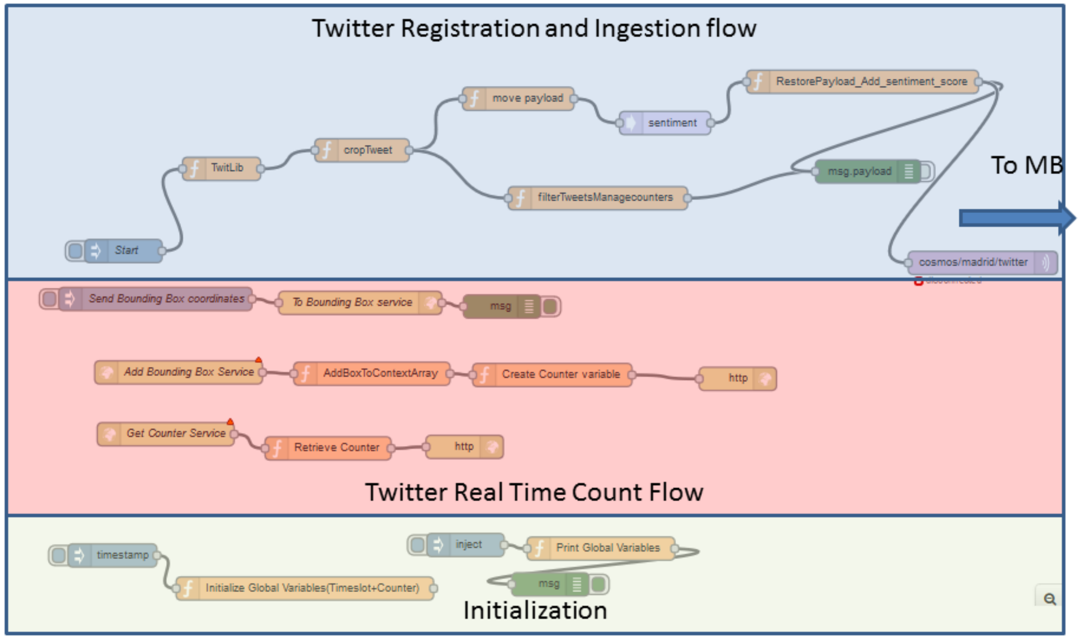
```
var velocity=msg.payload.velocidad;
localclusters=global.get("foundclusters");
var overall_cluster=-1;
var diff;
var min=10000000;
for (var k=0;k<localclusters.length;k++){
    diff=Math.abs(localclusters[k].centroid[0]-velocity);
    if (diff<min){
        overall_cluster=k;
        min=diff;
    }
    console.log(diff));
}
if (overall_cluster===0){
    msg.payload="bad state";
}
if (overall_cluster===1){
    msg.payload="good state";
}
msg.payload.cluster_number=overall_cluster;
return msg;
```

Overall flow



Twitter count per location flow

- Dependencies
  - https://github.com/ttezel/twit
  - To be added in settings.js



Twitter count per location flow

- Functionalities
  - Register in Twitter Streaming API (needs credentials from a Twitter account inside the Twitlib function node)
  - REST Service to register a location bounding box for monitoring
    - http://localhost:1880/PostBox
  - REST Service to get current counters for all bounding boxes
    - http://localhost:1880/counters
  - REST Service to delete a bounding box from monitoring
    - http://localhost:1880/deleteBox