



Politecnico di Milano

Advanced **N**etwork **T**echnologies **Lab**oratory



Challenge

Home challenge #2: Node-RED

Home Challenge #2



- The goal of this homework is to replicate real MQTT messages and send them to a thingspeak channel.
- Input data is in the file «traffic.csv». It contains a subset of the messages used for the challenge #1.
- Thingspeak channel receives data and visualize them using:
 - 2 charts for field1 and field2
 - 2 lamps indicator (turn it ON w/ data ≥ 2000) for field1 and field2

Home Challenge #2



- Node-red application must
 - Read the traffic.csv file
 - Keep only publish messages coming from the following topics:
 - factory/department1/section1/plc
 - factory/department3/section3/plc
 - factory/department1/section1/hydraulic_valve
 - factory/department3/section3/hydraulic_valve
- Send the «value» field of the original message as MQTT messages to the thingspeak channel in order to fill charts and activate indicators
- Wait 1 minute between consecutive messages

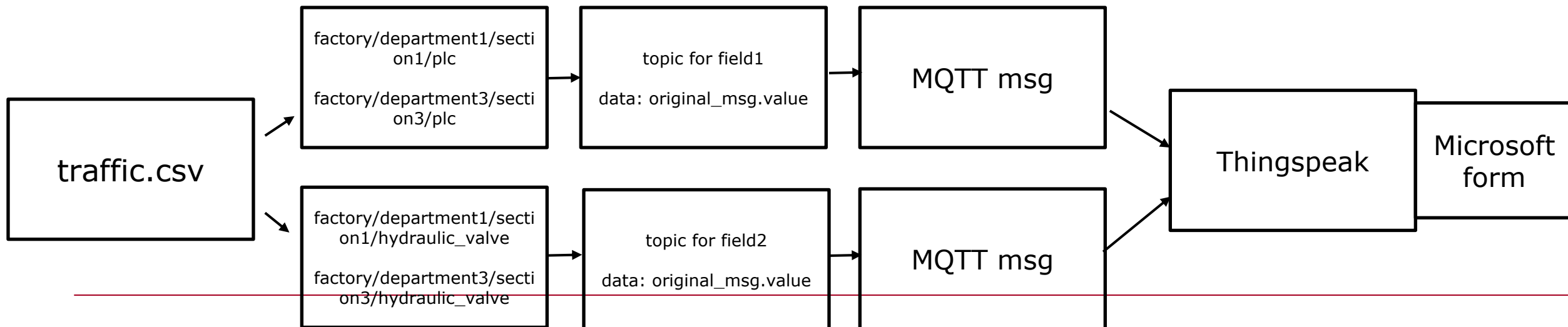
```
{"id": "Actuator 929",  
  "value": 24,  
  "lat": 1300,  
  "lng": 1430,  
  "unit": ".",  
  "type": "Motion"}
```

Home Challenge #2



- Field1 must receive data from: (merge the values)
 - factory/department1/section1/plc
 - factory/department3/section3/plc } factory/all_departments/plc

- Field2 must receive data from: (merge the values)
 - factory/department1/section1/hydraulic_valve
 - factory/department3/section3/hydraulic_valve } factory/all_departments/hydraulic_valve



Home Challenge #2 (hints)



- ❑ If you're not familiar with javascript (it's super easy btw) use the function node only when necessary
- ❑ The payload of the original message is in hex, convert it to string
 - you could use: https://nodejs.org/dist/latest-v10.x/docs/api/buffer.html#buffer_class_method_buffer_from_string_encoding
- ❑ Use the delay node (rate limiter) in order to wait 1 minute between messages
- ❑ Debug node is your friend!
- ❑ Google is your friend 😊