

GT IOT // 15/12/2020



Emmanuel Landrison

- 1.Introduction à Node-RED
- 2.Exemple : API avec Labview
- 3.Exemple: LPWAN private
- 4.GT IOT

Introduction



My env. → Engineer // Research team // Chemical process

My job → Build test bench : 50 I/O, 1S/s.

Strategy : decoupled jobs

- Remote control → LabVIEW
- Security → autonomous Web relay card
- Monitoring and alerting → **Node-red** + passerelle



Benefices : Set it and forget it

- Quick to deploy and easy to maintain
- Autonomous end-user oriented

Node-RED

Low-code programming for event-driven applications

Latest version: v1.2.5 (npm)

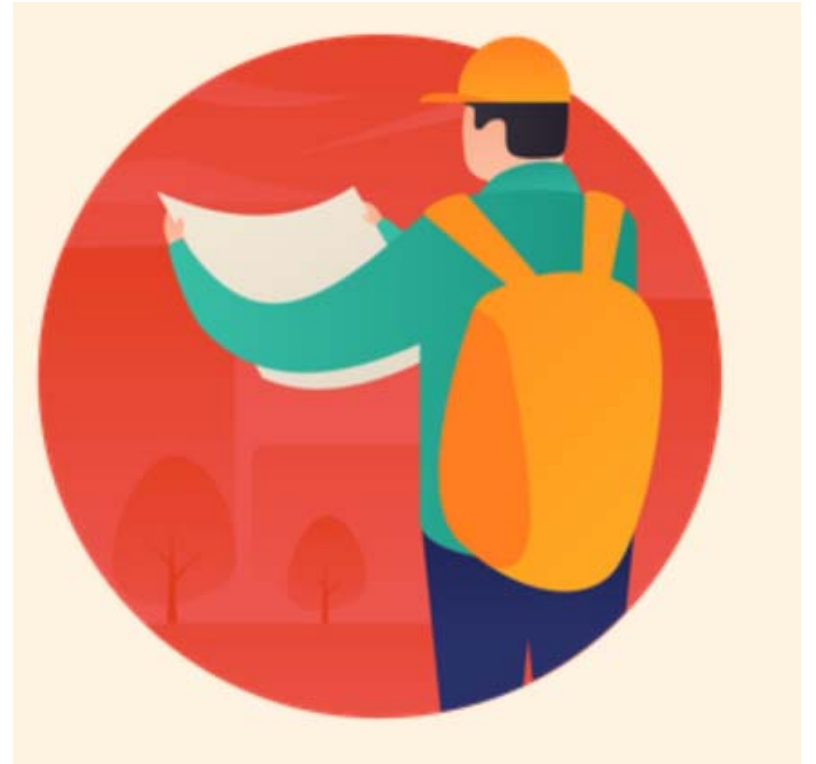
<https://nodered.org/>



Emmanuel Landrison

You will Discover...

- Node-red **environment**
- **Install** node-red
- Demo basics
- How to start learning



Browser-based flow editing, on a server

data = {Json object}

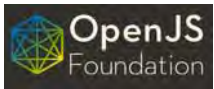
<http://localhost:1880>



Interface Environnement
Deploy
Debug

Graphical programming :

- Quick to deploy
- Increase code maintainability
- Easy to debug



Designed and built by IBM

Deploy button (to push before executing a flow)

Menu button

Palette of node types

Layout to draw flows

Panel showing contents:

- Info of selected node
- Debug
- Configuration Nodes

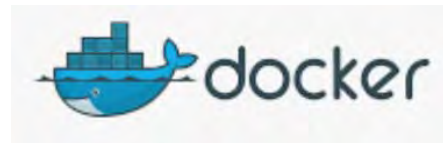
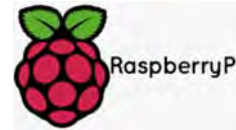
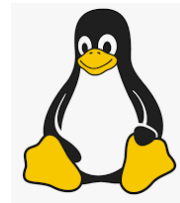
The screenshot shows the Node-RED web interface in a browser window. The address bar shows 'localhost:1880/#flow/7a1f7436.7aac3c'. The interface includes a left sidebar with a 'filter nodes' search bar and a 'palette of node types' containing categories like 'input', 'output', and 'http'. The main workspace is labeled 'Flow 1' and contains a dashed green box representing the 'Layout to draw flows'. On the right, there is a 'Panel showing contents' with sections for 'Info', 'Information', and 'Description'. A 'git' logo is overlaid on the right panel. A 'Deploy' button and a 'Menu' button are visible in the top right corner of the interface.

Runs everywhere with everything (almost)

Inbuild
product



Local environnement



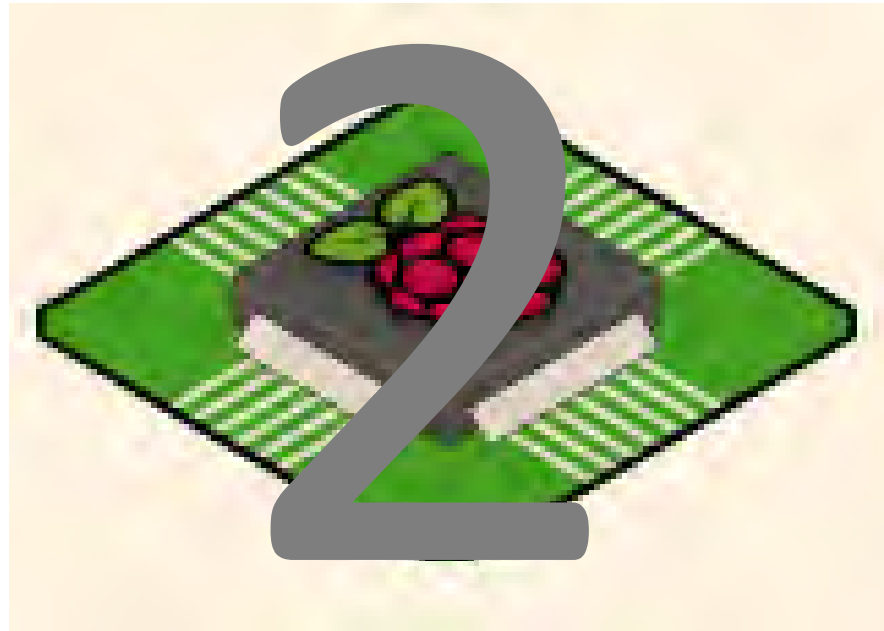
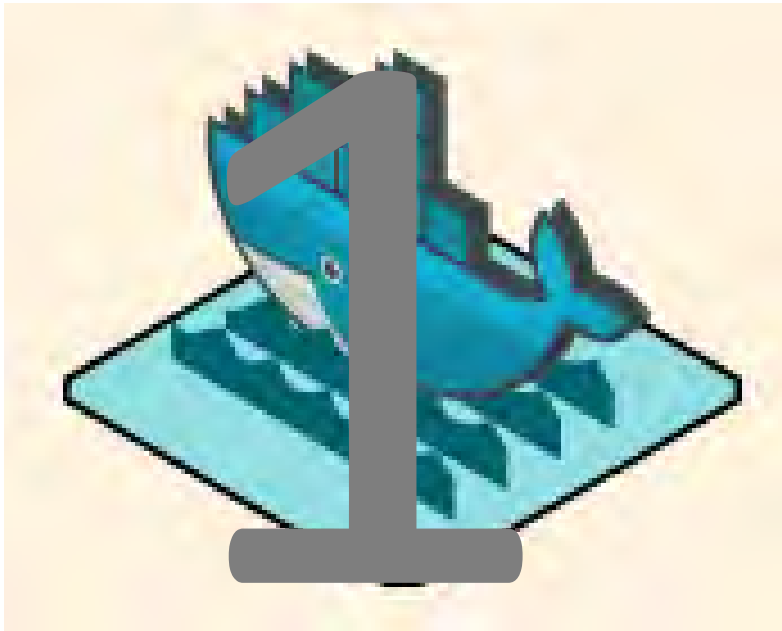
Synology



Cloud



Installation



install with  on  (2 min. with reboot)

- 1) Install the app “Docker desktop” (reboot...)
- 2) Open powershell (win+R , powershell)
- 3) Paste bellow command (change myfolder)

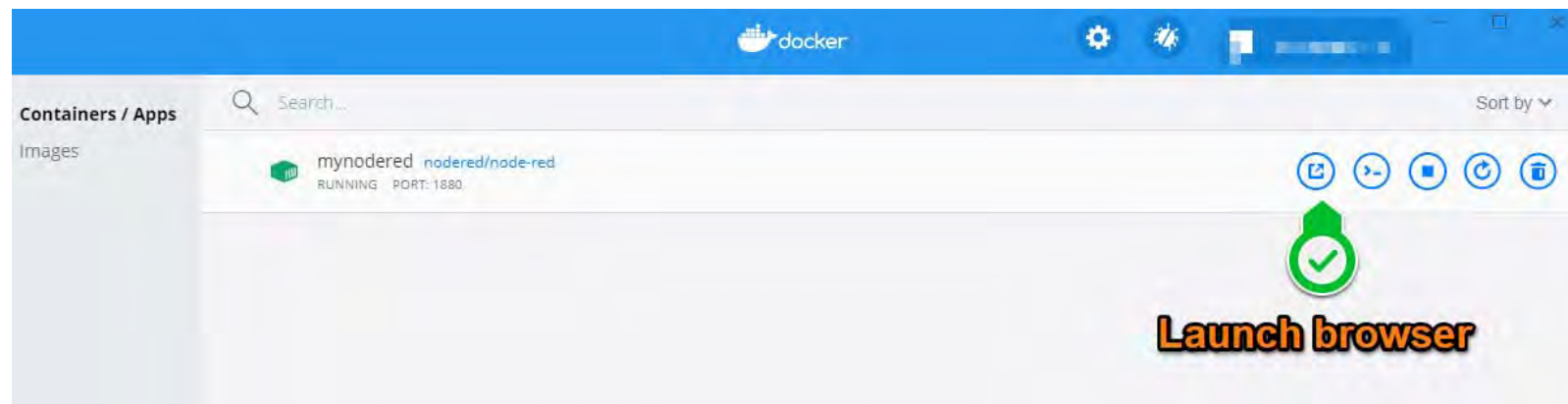
```
docker run -d -p 1882:1880 --name=mynodered --restart unless-stopped -v c:\myfolder:/data nodered/node-red
```

Run detach ports host:cont image-name run strategy mount data volume on host official Node-red image

- 4) Launch browser with Docker desktop



Copy / Paste PS
Docker desktop





Raspberry install (3 min)

- 1) Run the Bash Script : `bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)`
- 2) Run as a service : `sudo systemctl enable nodered.service`
- 3) Access with browser : `http://<hostname>:1880`

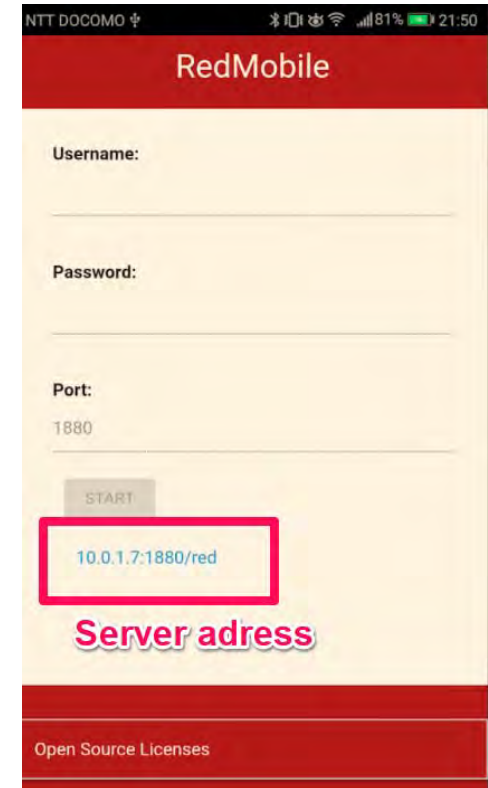
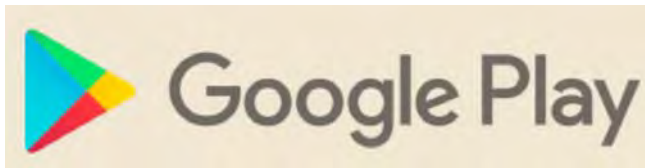


Already installed on the Raspbian desktop image version → just make it run as service, reboot and check

Install on



(play at home)



- Do not use in production
- 5 €

Social Development

→ Easy sharing, flow community, clone Git repo...



<https://flows.nodered.org/>

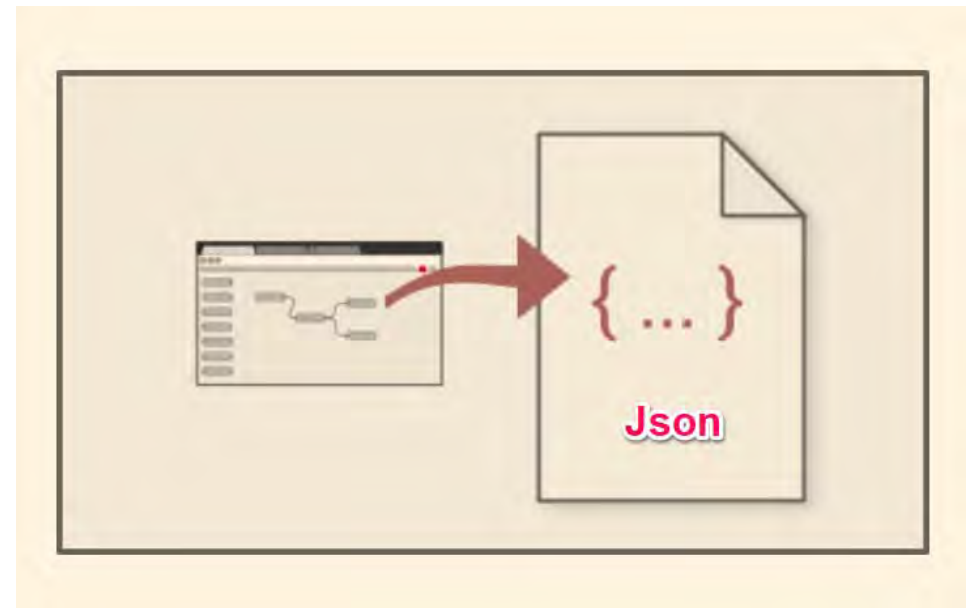
Find the nodes, share your flows and see what other people have done with Node-RED.

Recent nodes see more (2900) ▶

- node-red-contrib-eesmart-d2l
Converts the data sent by the D2L into readable data.
v0.1.0 📄 84 ⭐ 5.0
- node-red-contrib-c... piagara
Nodes
- node-red-node-geofence
A simple node to filter based on location.
v0.1.4 📄 1251 ⭐ 5.0

Recent flows see more (1759) ▶

- Node-Red MotionLight3
This flow allows you to enable your lights based on motion. It contains a subflow for a delay which is needed because the normal...
holomeke flow
- ...a for Halloween
Flows
notenoughtech flow
- Node-Red NowDB Demo
Node-Red NowDB DEMO
taufiksu flow



Check code compliance (coder reputation, git statistics, user comments) of the nodes.

Some Flows (1/2)

Inject Node



Event
Json manipulation

- Insert various messages
 - time stamp
 - object
 - String
- Trigger options
 - on startup
 - on schedule
 - on click

Debug Node



- Output option
 - Log
 - Debug Window
- Output scope
 - msg
 - payload
 - custom

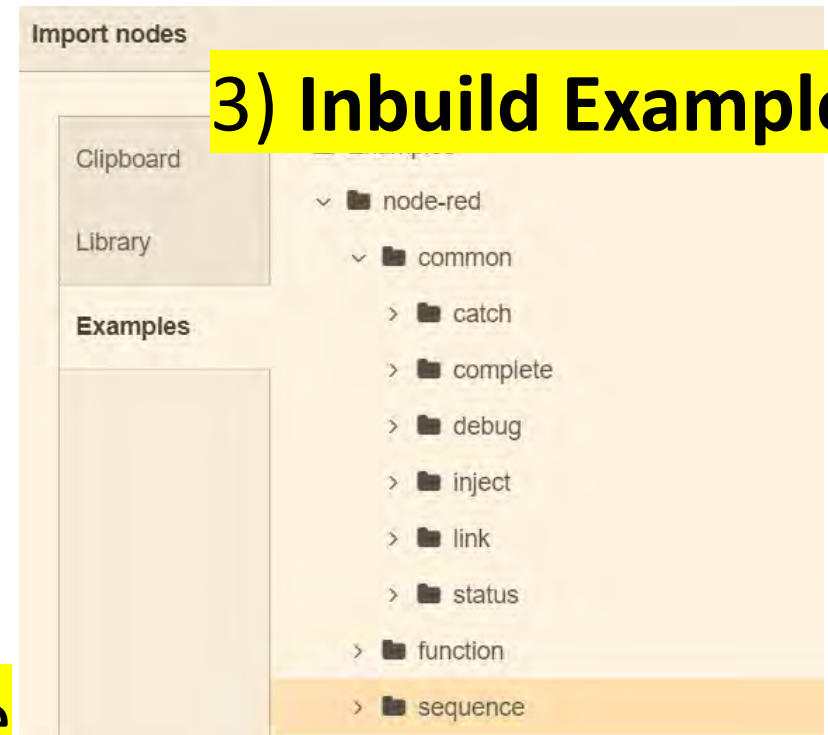
How to start → <https://nodered.org/docs/>



1) Creating flows Step by step



2) Cookbook watch, copy and paste



3) Inbuild Examples

Exemples Nœuds / implémentations

- RDE
- ASR
- Devlog

Devlog

Nœud Uibuilder → Page web
Back + Front : géré par node-red



Pages web simple
(html, css, js)

ou

Implémentation
de Frameworks



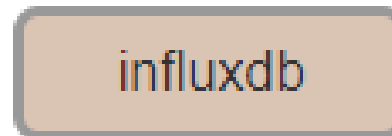
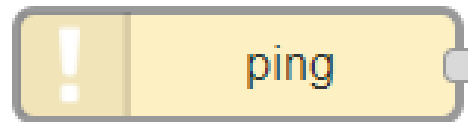
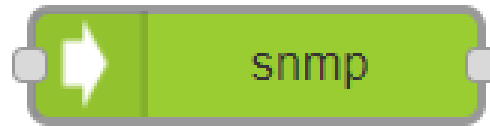
Gestion des bases de données
API d'un programme



Python



ASR



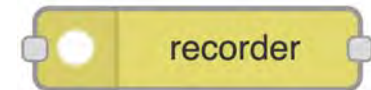
Météo de services
Watchdog, alertes

Génération de documentation d'API

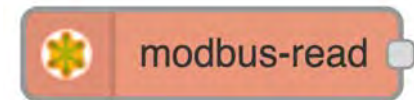
Dashboard

RDE

Petite application séquentielle pour le contrôle d'instruments



Surveillance consommation de gaz du bâtiment



Lien avec une API RESTFUL LabVIEW

{REST:API}



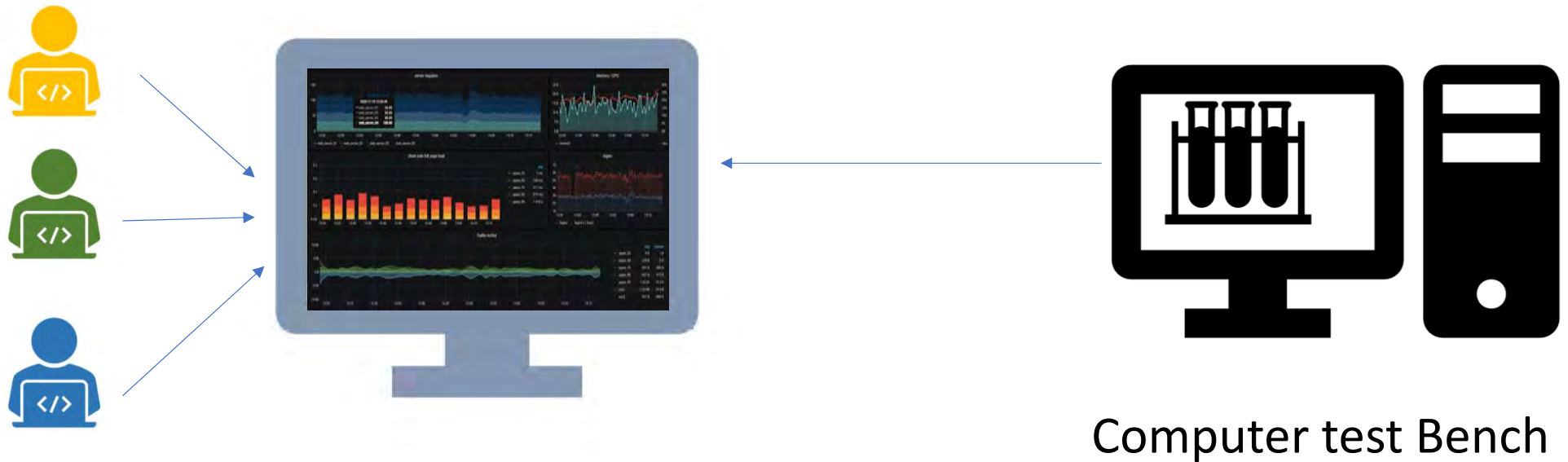


LabVIEW

{REST:API}

Node-RED

Why web interface ?



Clients needs :

- Watch test bench indicators
- Retreive the data
- Alerts



LabVIEW™



nodered [nodered/node-red](#)

RUNNING PORT: 1880



mosquitto [eclipse-mosquitto](#)

RUNNING PORT: 1883



influxdb [influxdb:latest](#)

RUNNING PORT: 8086



grafana [grafana/grafana](#)

RUNNING PORT: 3333

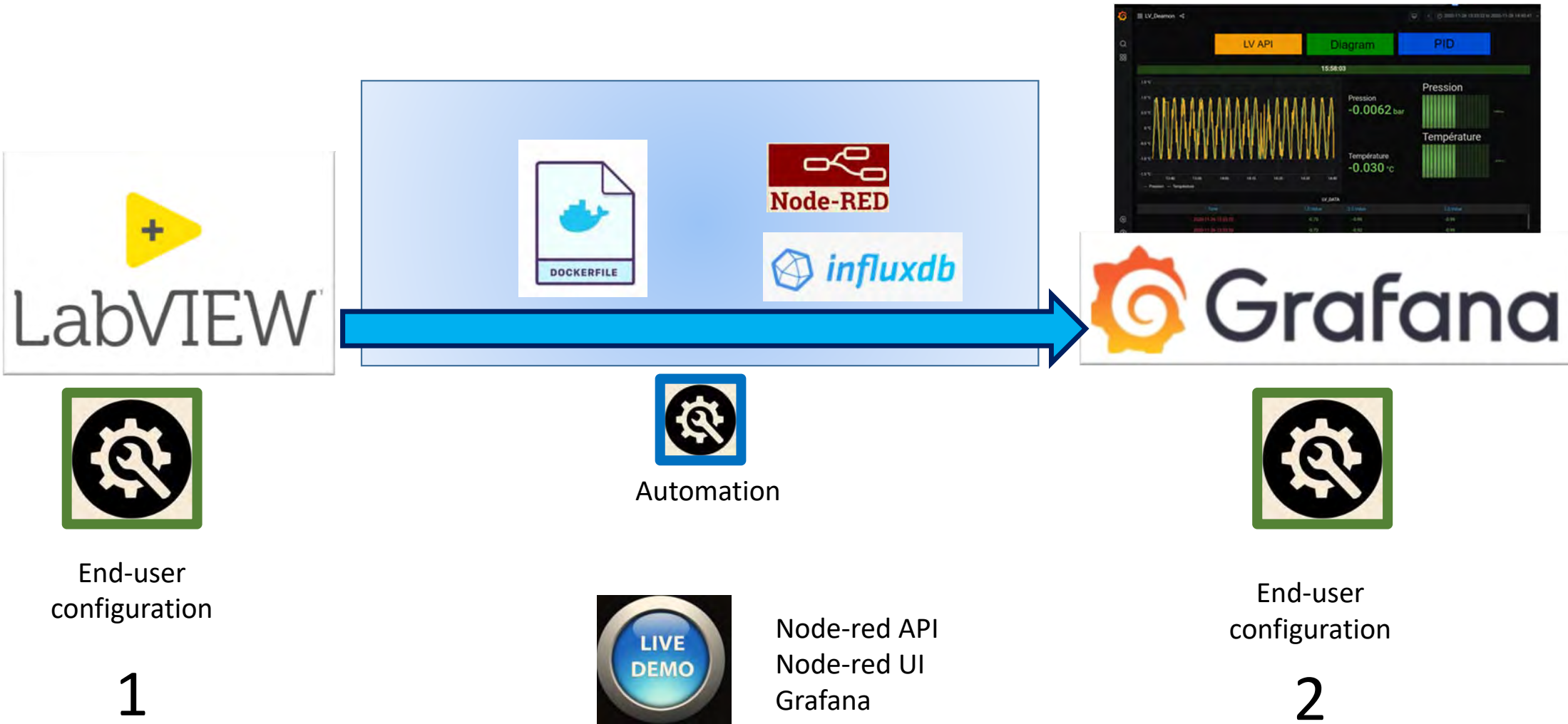
Architecture de mon application

Architecture Plugins → Modularité

Démons instruments → autonomie et instanciation

API RESTFUL → Interopérabilité

Integration : Only 2 end-user « **super easy** » actions

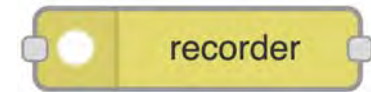
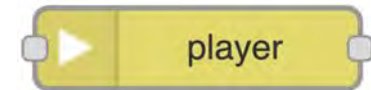


Exemples Nœuds / implémentations

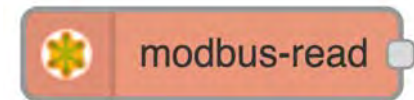
- RDE
- ASR
- Devlog

RDE

Petite application séquentielle pour le contrôle d'instruments



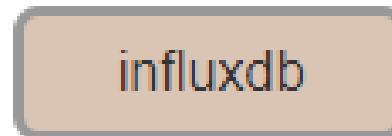
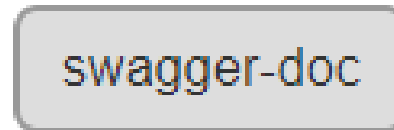
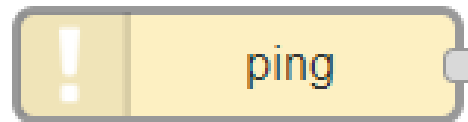
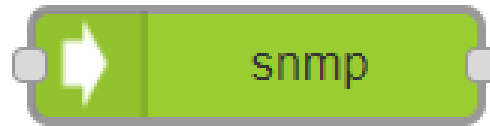
Surveillance consommation de gaz du bâtiment



Lien avec une API RESTFUL LabVIEW **{REST:API}**



ASR



Météo de services

Génération de documentation d'API

Dashboard

Devlog

Uibuilder → Page web

Back + Front : géré par node-red



Simple pages web
(html, css, js)

ou

Implémentation
de Frameworks



Gestion des bases de données (API)



Python

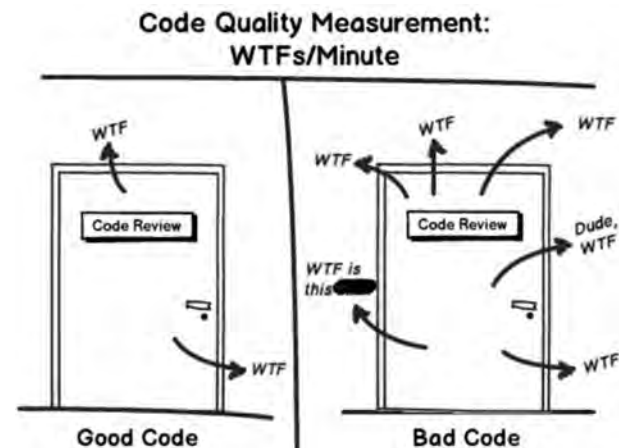


Node-red : langage commun autour de l'IOT ?

- Simple et graphique
- Partage facile
- Léger
- Intégration d'un client git
- Gestion des dépendances
- Nœuds adaptés aux différents besoins métiers

Propositions et perspectives pour Node-red

- Création d'un **document** sur les bonnes pratiques (clean coding)
 - Maintenance du code
 - Découplage des processus
 - Sécurité
 - Intégration continue (Gitlab, Docker...)
- Création d'un **Template** sur « Gitlab public » : projet de départ
 - Gestion des variables / config
 - Couche d'abstraction matériel
 - CVT (current value table)
 - Manipulation des données
 - UI : user interfaces
 - Datalog
 - Bibliothèques, toolbox
- **Revue de code** (via Gitlab)
 - Clone et Fork du projet



~~1.Introduction à Node-RED~~

~~2.Exemple : API avec Labview~~

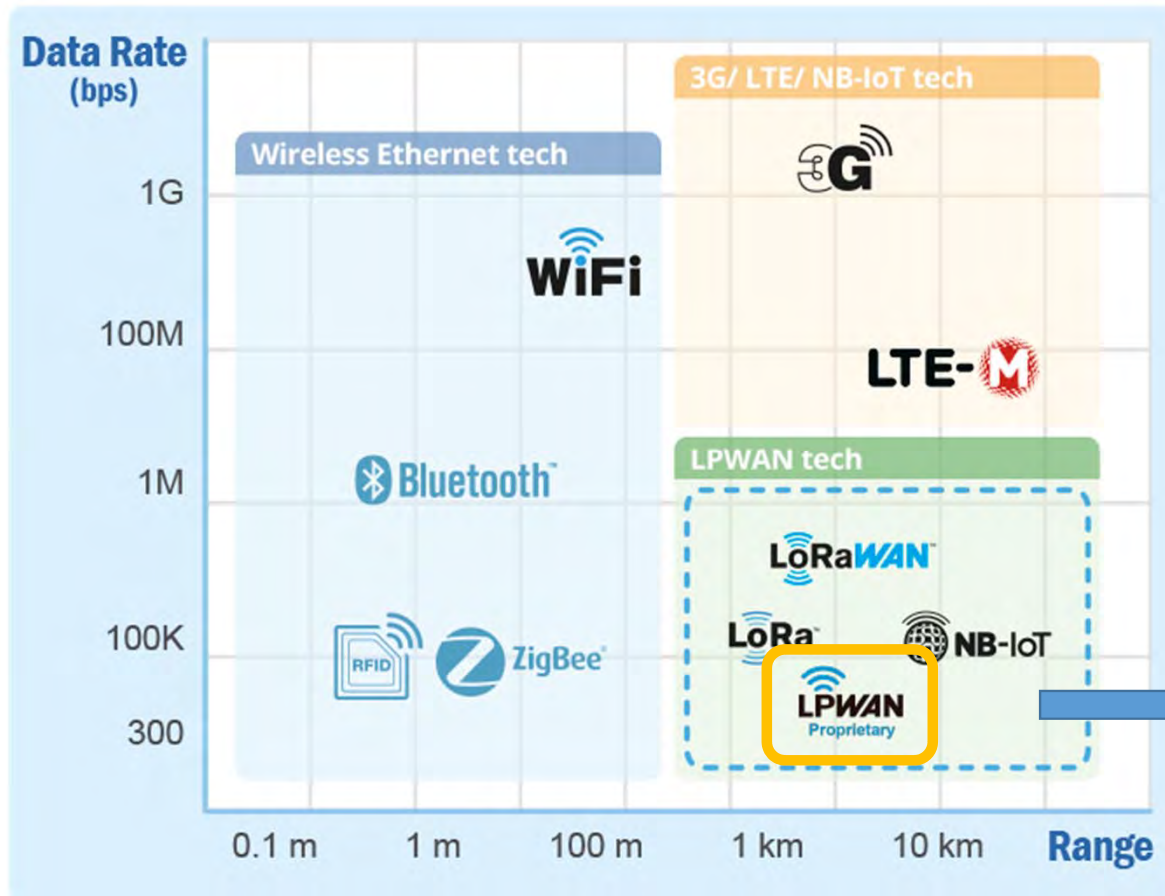
3.Exemple: LPWAN private

Advantech : Wise 4210 series

Industrial Proprietary LPWAN
(SUB-G) Wireless I/O Module



Tawain : IOT company solutions



Low Power Wide Area Network
#1GHz

Sensor node



LPWAN Sensor Node
WISE-4210-S200 Series

WISE-S214 (4AI/4DI)

4 AI, 16 bit (diff)
4 DI



Datas :

- Polling : toutes les x min
- Push : Sur évènement (Ai % / Di)



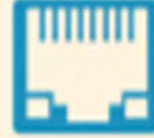
LPWAN



Point d'accès Passerelle



Ethernet
Alimenté DC (10-50 V)

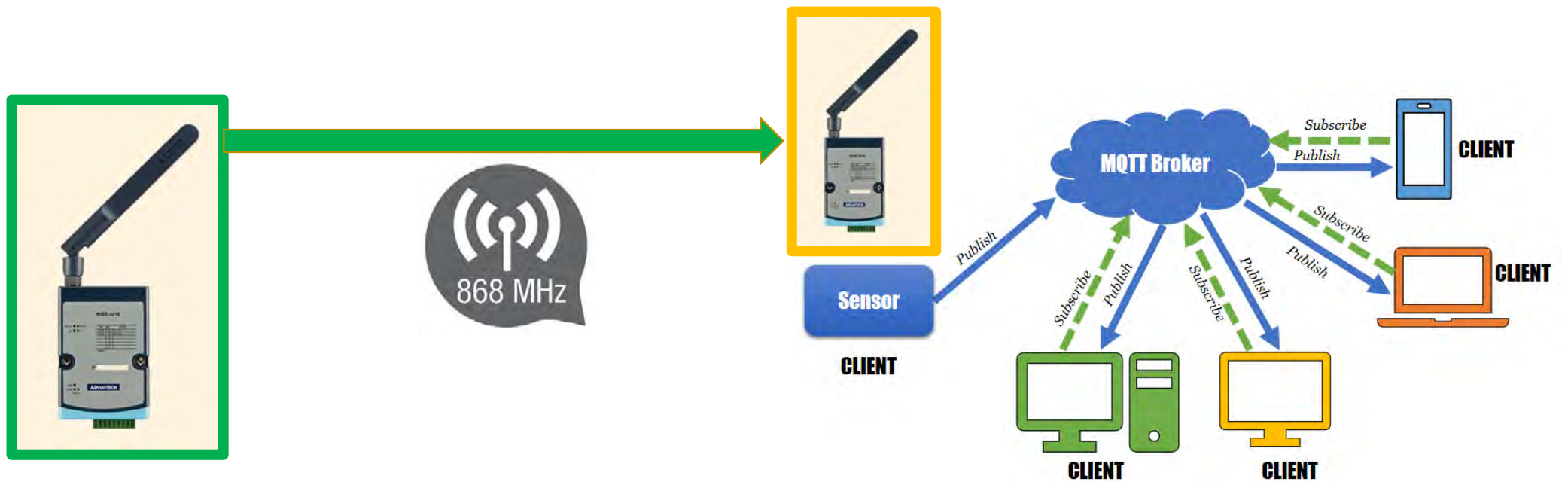


Modbus MQTT RESTful



LPWAN AP
WISE-4210-AP

Communication MQTT



Sensor Node

Push (sur évènement)
IOT Standard.
Information traitée.

Besoin d'un serveur MQTT

Node-red Integration



1) Configuration du point d'accès : envoie des données vers le **serveur MQTT**



Publish data

Configuration

Information Wireless Network App Time & Date Time Sync Modbus Control General **Cloud** Firmware Account

Cloud Configuration

Select Service iSensing MQTT

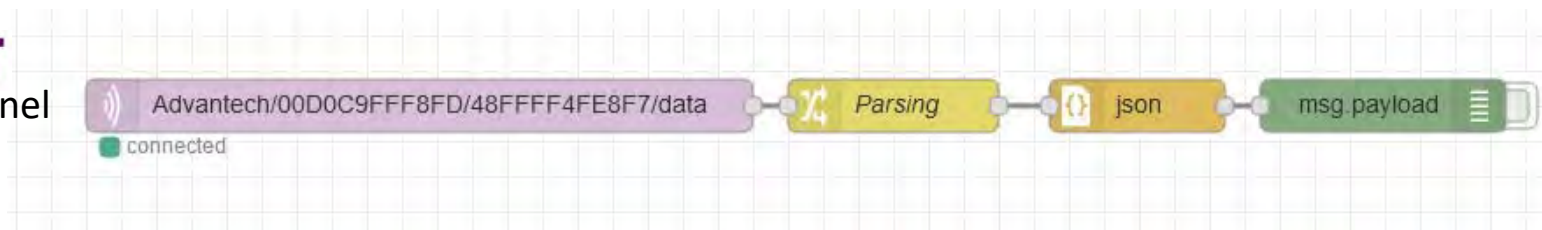
Setting

MQTT Host Name	192.168.0.101 <i>IP_addr_of_PC</i>	Port Number	8443
SSL secure	<input type="radio"/> Disable <input checked="" type="radio"/> Enable	User Name	
Heartbeat Frequency	60	Password	
Publish QoS	1	Subscribe QoS	1

2) Recuperation des données sous Node-red : MAC_Eth/MAC_End-node/data



MQTT
Subscribe channel



Communication et Informations

Modbus TCP serveur (protocole industriel)



Point d'accès

2. Modbus Address Table for WISE-4210-S250

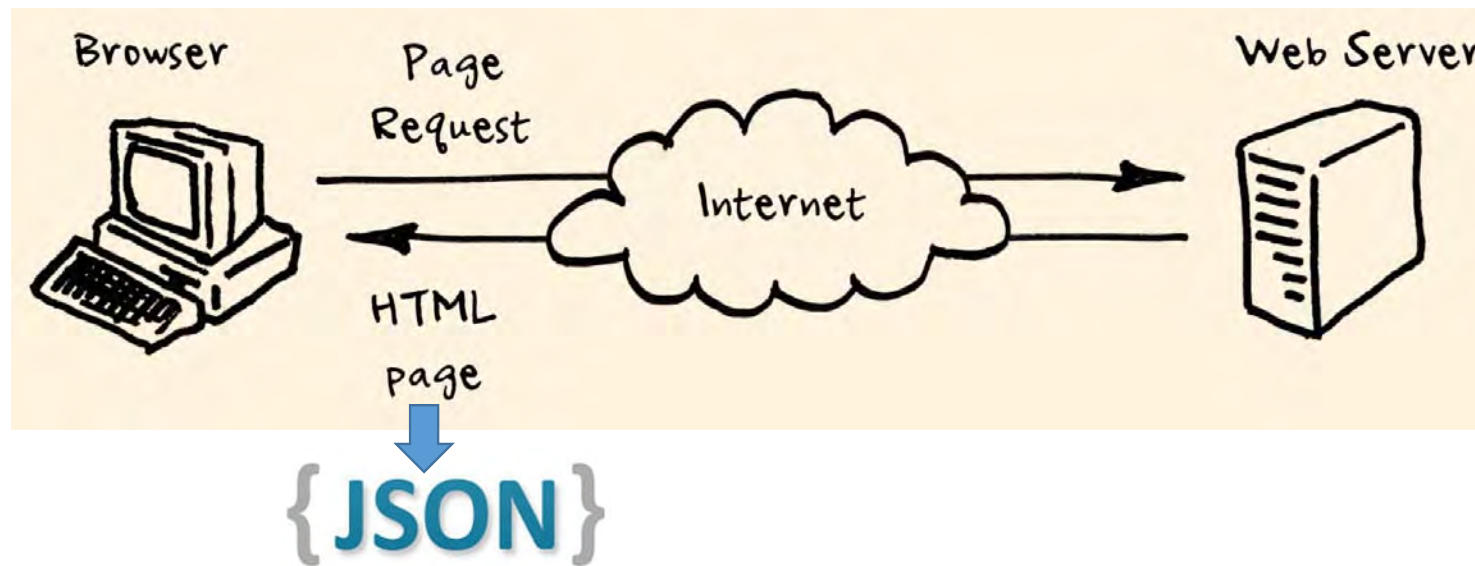
Modbus Address	I/O
00001	D10
00002	D11
00003	D12
00004	D13
00005	D14
00006	D15
01001~01032	RS-458 Coils
41001~41032	RS-458 Registers
45302	RSSI (-dBm)

Implémentation automate



Traitement de l'information

Communication et Informations { REST }



Info sur la puissance du signal

Polling (requête : X / min)
Traitement de l'information

Node-red Integration

{ REST }

HTTP GET : IP/lpwan_message/slot_MacID

```
Content-type: application/json
Response: 200 OK
{
  "TIM": "2014-11-11T15:48:32+08:00",
  "UID": "WISE-4210-AP_00D0C90E8738",
  "MAC": "00-D0-C9-FE-16-01",
  "Rssi": -33,
  "Record" :
  [
    [0,0,1,0],
    [0,0,2,50],
    [0,0,3,10],
    [0,1,4,0],
    [0,1,5,456],
    [0,1,6,0],
    [0,0,30,0],
    [0,0,31,16],
    [0,0,32,32767]
  ]
}
```



Conclusion : facilité d'intégration

MQTT

{ REST }

Modbus

2

1

3

Cout # 500 euros

Rapidité d'integration
→ Pas de décodage trames

Multi-protocole

Support

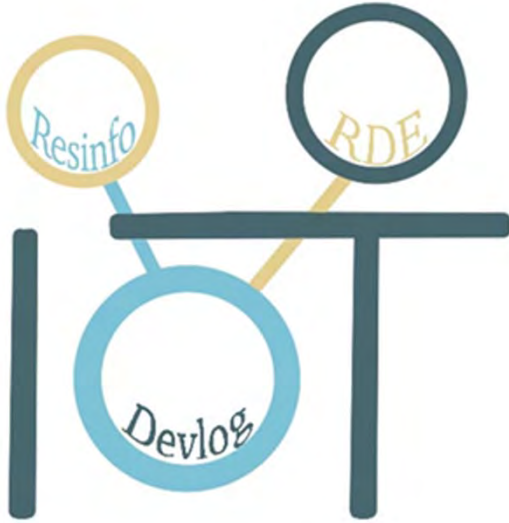


Access point	218
Sensor module	150
Module 4AI	117

Modulaire

64 end-nodes

GT





- Base de donnée du groupe
 - → se connaitre, compétences
- Base de données : Ref, tutos...

- Consultation de solutions techniques
 - réflexion sur des architectures, produits.

- Node-red
 - Template / Framework
 - POC Cloud
 - Formation

