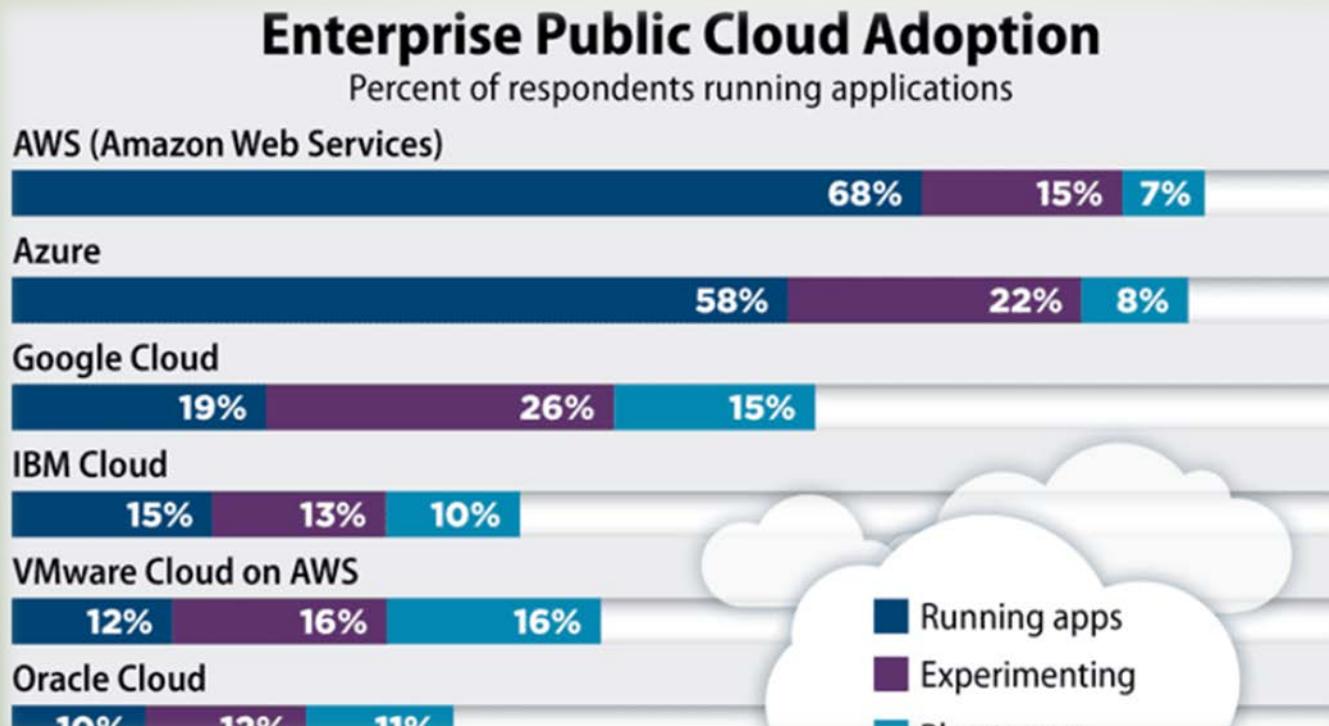


# Mise en œuvre de l'IOT

## Solutions matériels et logicielles



# Interopérabilité



# Quelles sont vos attentes ?

Pourquoi voulez vous mettre en place une solution IOT ?

Quelles sont vos principales difficultés ?

Qu'est ce que pour vous, un bon système IOT ?

# Contexte / environnement travail (1/2)



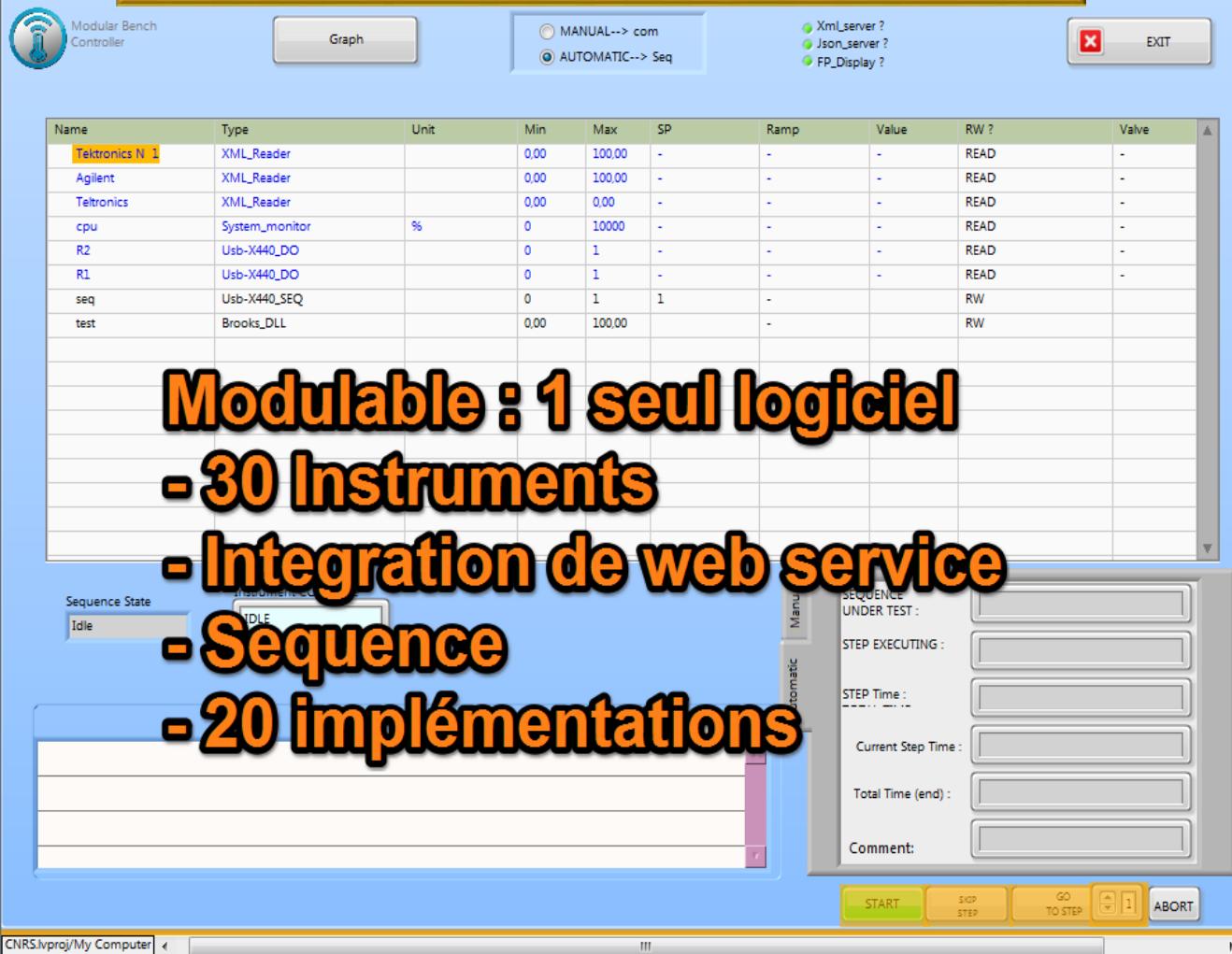
- Laboratoire chimie #200 personnes / catalyse
- **Unité fonctionnelle : Equipe Ingénierie**
- Responsable instrumentation de l'équipe
  - # 30 personnes / 20 étudiants ou CDD
- Changement étudiants / Projets tous les 3 ans

Conception modulaire pour le matériel et les logiciels  
→ **Simplification + optimisation**

IOT : qualité , sécurité , Cout.

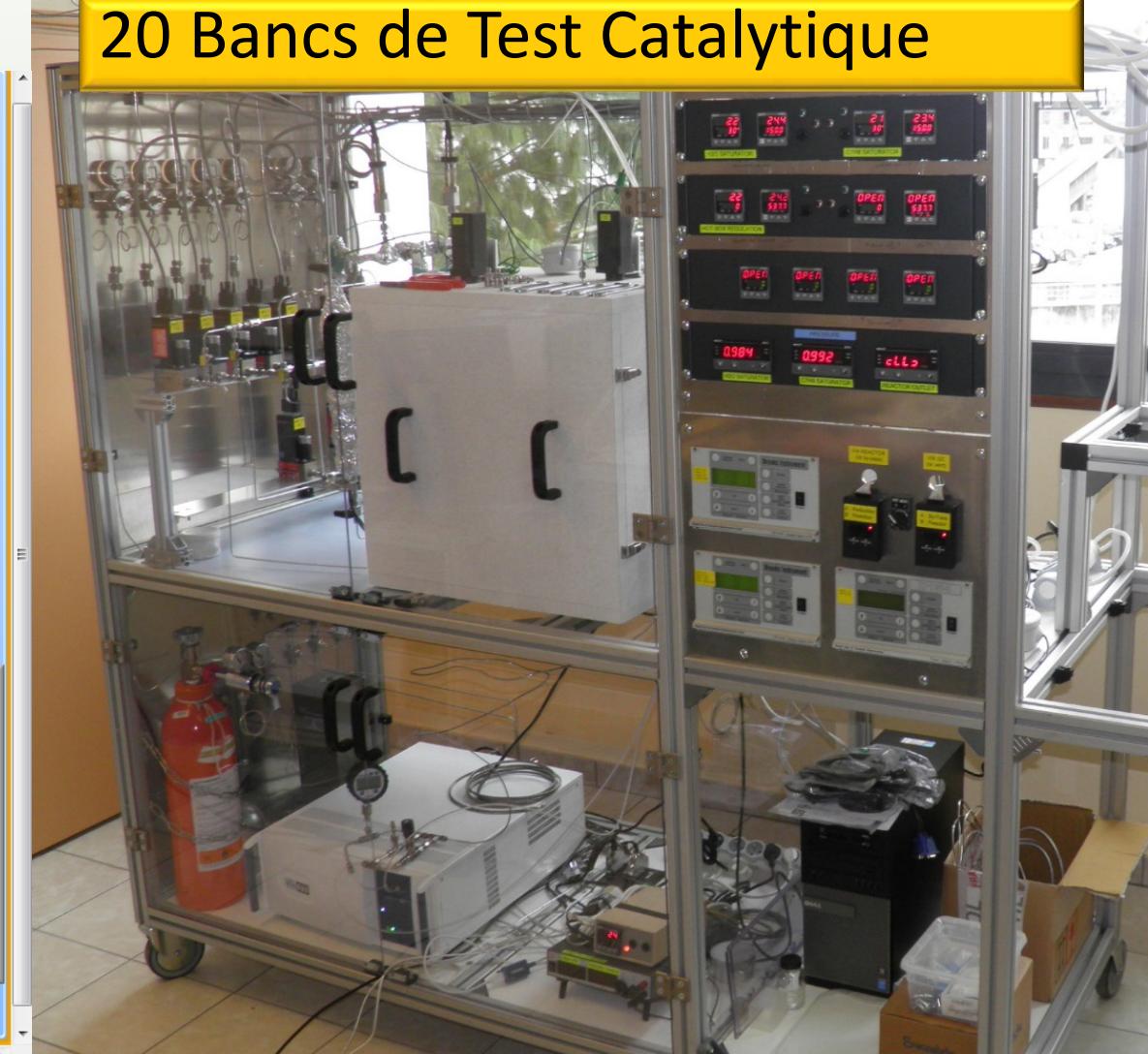
# Contexte / environnement travail (2/2)

## 1 logiciel



**Modulable : 1 seul logiciel**  
**-30 Instruments**  
**-Integration de web service**  
**-Sequence**  
**-20 implémentations**

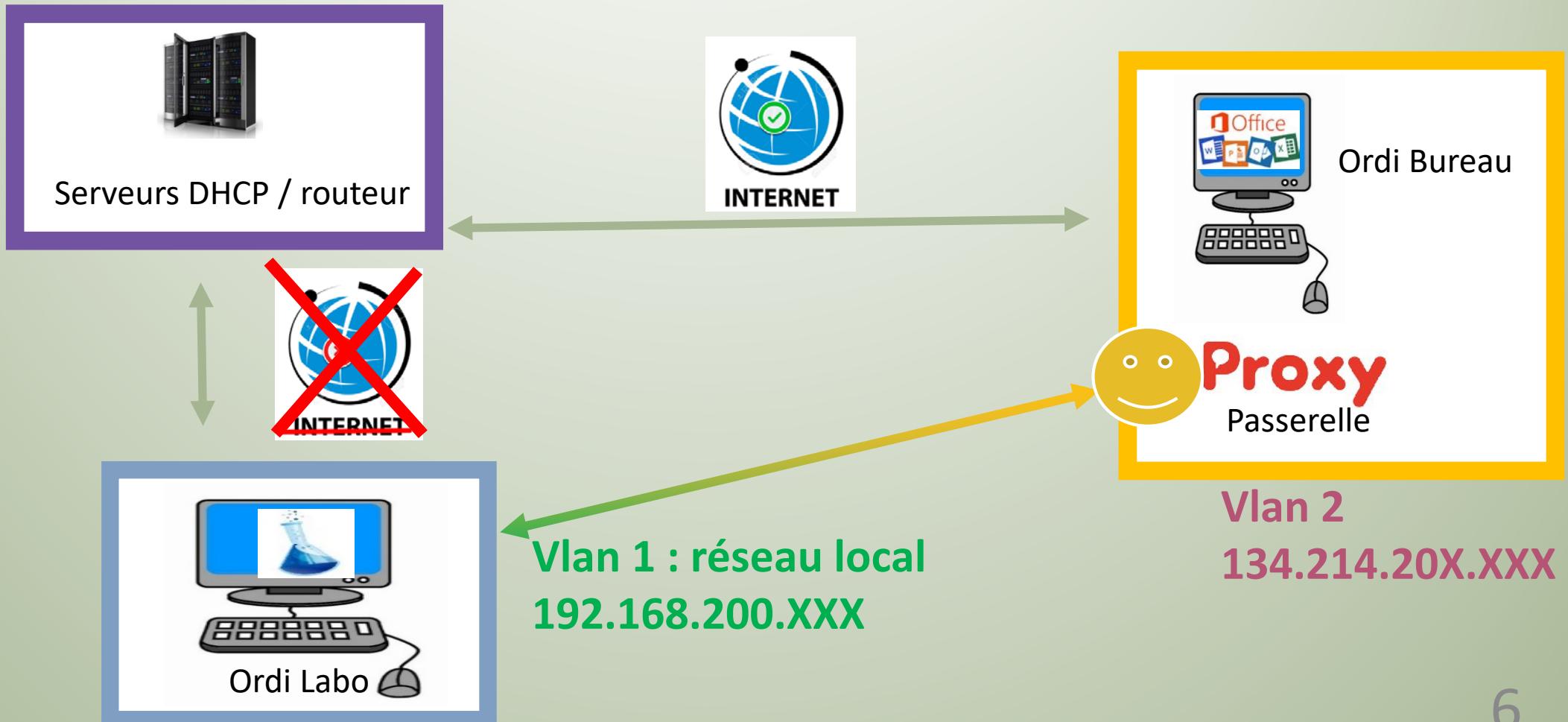
## 20 Bancs de Test Catalytique



# Intro : Idée reçue N°1

→ Mon Ordinateur de manip n'est pas connecté a internet  
donc je ne peux pas accéder au données en dehors du laboratoire !

**FAUX**



Intro : Idée reçue N°2

→ Ce n'est pas assez sécurisé



**FAUX**

→ On peut lire uniquement les données → aucunes actions



→ On choisit les données que l'on veux envoyer

→ On accède pas aux données directement : méthode **encapsulation**

→ On peut **limiter** l'accès à certains utilisateurs (root/admin/user)

# Intro : Idée reçue N°3

→ Cela ne sert à pas grand-chose d'envoyer sur internet...

**FAUX**

→ Alerte des utilisateurs (mail/sms...) 



→ Watchdog avancé.



→ Historique évènements / log



→ Visualisation des données.



Objectif → Satisfaction utilisateurs

# Interface logiciel : Intégration (1/2)

Objectif : Processus découplé

## Pourquoi



RT  $\neq$  IOT  $\rightarrow$  Choix et traitement des données à envoyées

## Comment

- Logiciel  $\rightarrow$  serveur de données (xml/json)

## Gain

- Faciliter le debug.
- Sécurité
- Robustesse



# Interface Matériel : choix (2/2)

*Device with I/O (Server)*



*Client Machine*



*HTTP Request*

{ REST }

*XML Response*

```
- <Response>
- <Terminal>
  <Name>Sum</Name>
  <Value>9.000000</Value>
</Terminal>
</Response>
```

# Interface Matériel : intégration (1/2)

- Quels sont les valeurs les plus **importantes** ?
- Quels sont les **objectifs** :

sécurité / dérive / validation / sauvegarde de données / proof of concept

**Découplage du système existant :**

Matériel → utiliser des I/O qui ne perturberons pas le système en place



Ne Jamais modifier un système sûr en fonctionnement

# Interface Matériel : choix (2/2)

- **Intelligence** aux niveau du **matériel** → simplification et robustesse
  - Régulateur de température (autonome, rampe, paliers)
  - DéTECTEUR de gaz (avec afficheur et relais)

- Haute interopérabilité → facilité et souplesse d'intégration
  - **API RESTFUL** → commandes http
  - **Serveur web** → XML ou Json
  - **I/O** : Pilotage sans com possible / découplage du système
  - Port communication Industriel : modbus, OPC

  Systèmes simple, configurable et auto-documenté

 Eviter les solutions propriétaires (pérennité et interopérabilité réduite)

# IOT MOXA : Modbus Gateway

**MB** : Armv7 / 512 Mo / 1Ghz / 8Gb eMMC

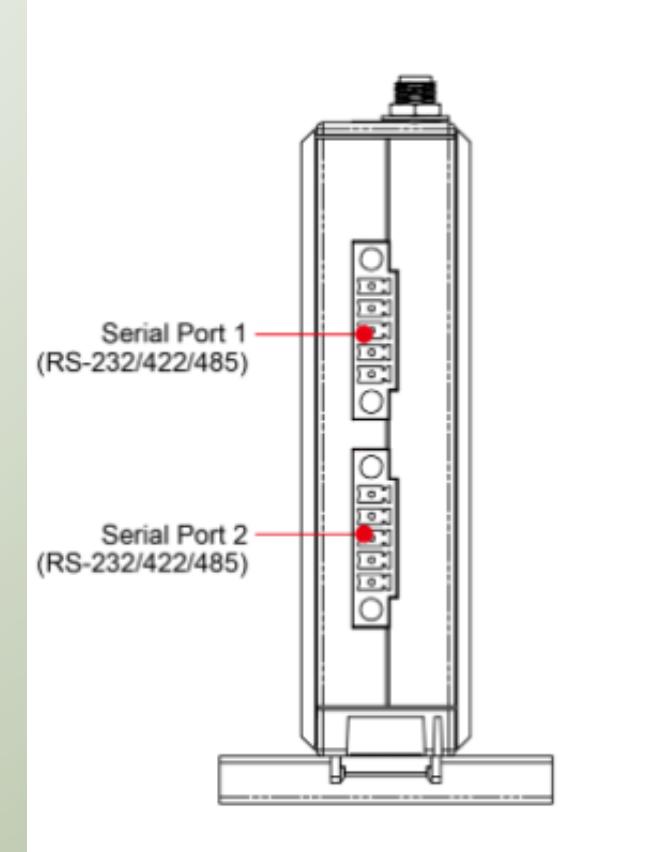
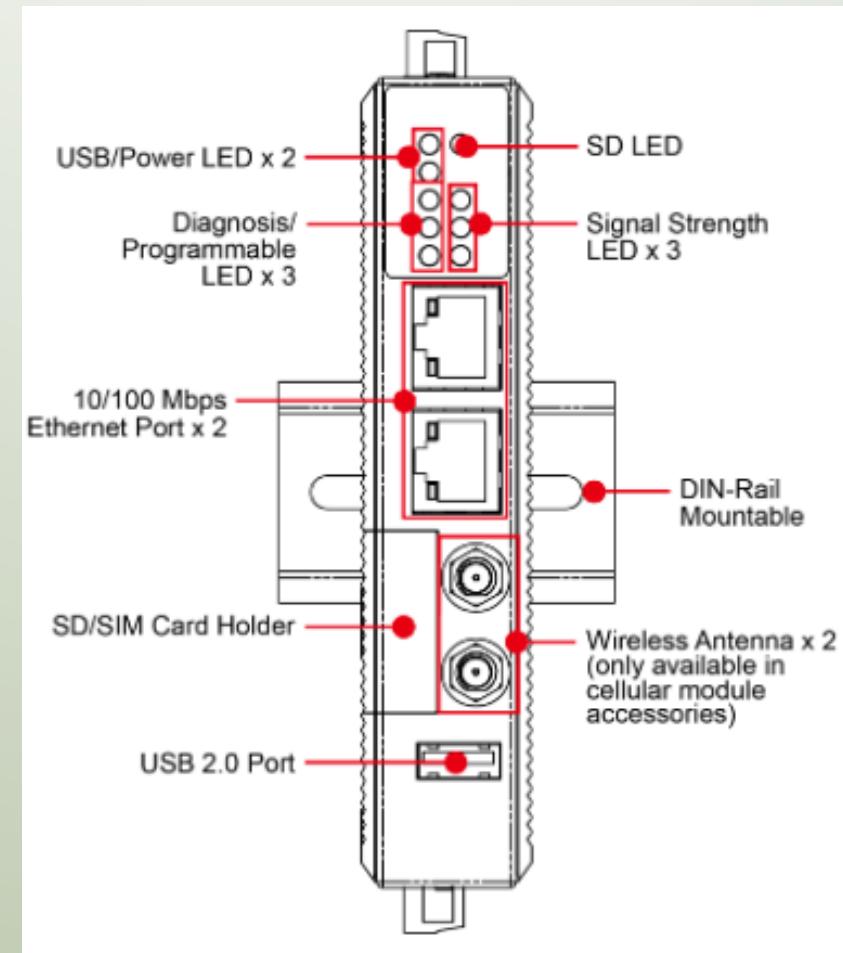
**Pre-installed OS** : Linux Debian 9 (Linux kernel v4.4)

**Storage Slot** : SD / USB

**Serial Ports** : 2 x RS-232/422/485

**Wireless (option)** : Wifi or 4g

**Ethernet** : Wan/Lan



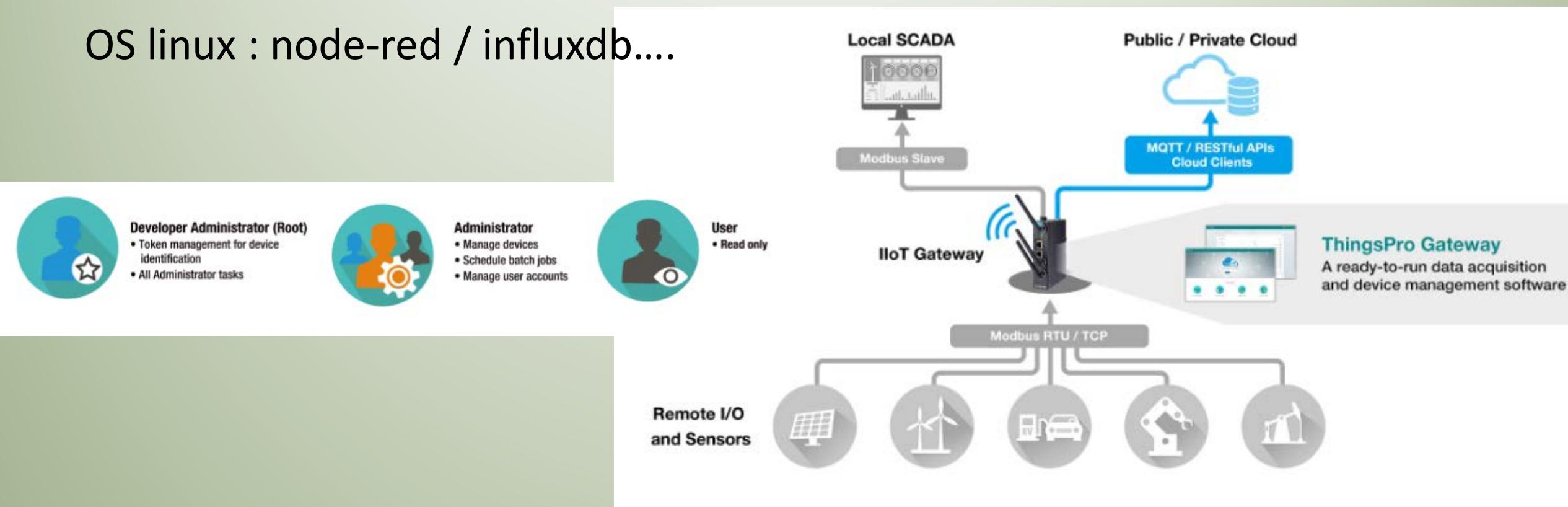
# Things Pro : surcouche logiciel

Simplification :  → IOT

Protocoles IOT : Mqtt / Azure / Aws / API { REST } ❤

Protocoles routeur : DHCP / Firewall / route

OS linux : node-red / influxdb....



# Calamp : Modem 4G

Objectif : accès au réseau 4G internet + I/O (4ai, 8di,8do)

OS : OpenWrt

Protocoles : SNMP / DHCP /firewall / VPN

Wireless : Wifi / GPS

Com port : X2



Command SMS-CLI (sans 4g)  
`/output pw=[password] rn=v...`  
`/iostatus pw=[password]`  
`/status pw=[password]`



# Eedomus : Box domotique



Double API local/distante { REST } ❤

Sauvegarde illimité

Pas de maintenance et Maj régulière ❤

Market, communauté



# Jeedom : Box domotique

Base de donnée + dashboard ❤

Opensource ❤

Local (+sauvegarde config Cloud)

API REST, Market



A screenshot of the Jeedom dashboard, which is a comprehensive control center for a home. The dashboard is organized into several panels: 1. CONFORT: Shows indoor (18.43°C) and outdoor (7.43°C) temperatures, humidity (58.6% and 82.83%), and a weather forecast for the next few days. 2. PRÉVISIONS: Provides a detailed weather forecast for the week, including temperature, precipitation, and UV index. 3. PRÉSENCE: Monitors the presence of family members (Benjamin and Emilie) and tracks their movements. 4. AUJOURD'HUI: Displays the current date (Ven, 7. Décembre 2018) and time (09:12:27). 5. RACCOURCIS: Offers quick links to various services like prime video, onkyo, and tv6play. 6. COURRIER: Shows a message from a delivery service. 7. TRAJETS: Lists travel plans for family members. 8. SÉCURITÉ: Manages security systems, including an alarm system (deactivated since Dec 7) and intrusion detection. 9. CHAUFFAGE: Manages heating systems, including a weekend mode (activated since Oct 31) and various heating zones. 10. RÉSUMÉ: Provides a summary of energy consumption (47.2°C, 600W, 450L, 10250L, 0.18€) and water usage (36L, 0.18€). 11. SUPERVISION: Monitors the status of various components like batteries, aggregators, cameras, and IHMs. The dashboard is highly modular and customizable, reflecting the open-source nature of the platform.

# Web relay card : I/O interface

**8Di/8Do/ 4Ai (16 bit)**  
Moteurs de scénarios  
API { REST } ❤  
M2M  
Push  
Watchdog...



# Moxa ioLogik E1200 Series : I/O



Modbus TCP → Scada / contrôle commande  
API { REST } → IOT ❤  
MX-AOPC UA Server (push)

# Passerelle avec I/O

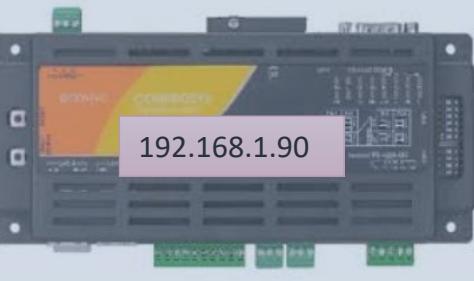


- Built-in **OPC UA Server**
- Promoting a multi-vendor system supporting a varied lineup of PLCs and **Modbus** devices
- Connect up to 10 links and data collection from 100 group registers
- Measurement and Upload
- **Web Monitoring build-in** ❤
- **Web Task Script**



Routeur 4G

API  
{ REST }



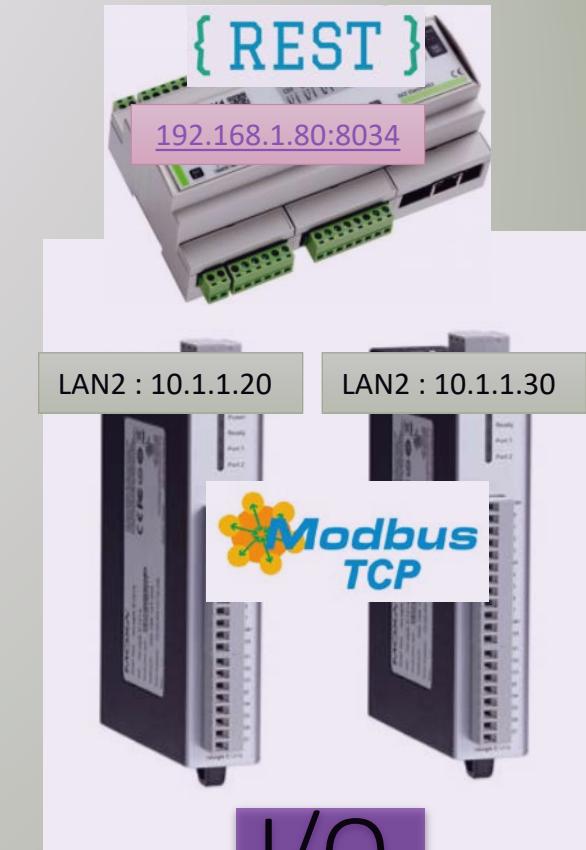
Passerelles



API  
{ REST }



API  
{ REST }



I/O

Modbus  
RTU



# Analog output



0-10 V // 12 bit  
Modbus TCP  
API REST : JSON



0-10 V // 14 bit  
Modbus TCP  
API REST : JSON



0-3,3 V // 16 bit  
API REST : XML



0-10 V // 10 bit  
Z-WAVE  
Polling



0-80 mv // 12 bit  
Modbus RTU  
API REST

# API RESTFUL

Low level : I/O moxa / IPX

Middleware : Moxa Things pro

Admin local : Eedomus php script

End-user Cloud: API eedomus

## ThingsPro API Reference

**modbus** : Handles Modbus framework configuration

DELETE	/mxc/modbus/equipment-templates
GET	/mxc/modbus/equipment-templates
POST	/mxc/modbus/equipment-templates
DELETE	/mxc/modbus/equipment-templates/{id}
GET	/mxc/modbus/equipment-templates/{id}
PUT	/mxc/modbus/equipment-templates/{id}
GET	/mxc/modbus/interfaces
POST	/mxc/modbus/interfaces
DELETE	/mxc/modbus/interfaces/{id}
GET	/mxc/modbus/interfaces/{id}
PUT	/mxc/modbus/interfaces/{id}
GET	/mxc/modbus/equipments
GET	/mxc/modbus/access/{equipment}/{tag}
PUT	/mxc/modbus/access/{equipment}/{tag}

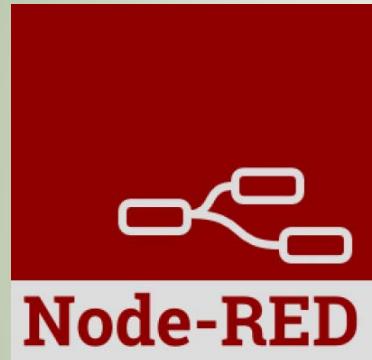
DE A VOUS  
JOUER

# Mise en œuvre

## **Changement Sortie analogique**

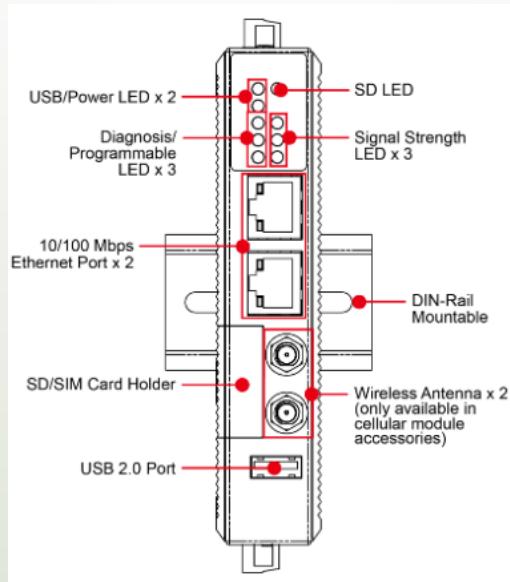
- Appui bouton → 3V
- Alerte si  $> 2$  Volt
  - Appel vocal
  - Sms
  - Push

## **Evènement Capteur de niveau**

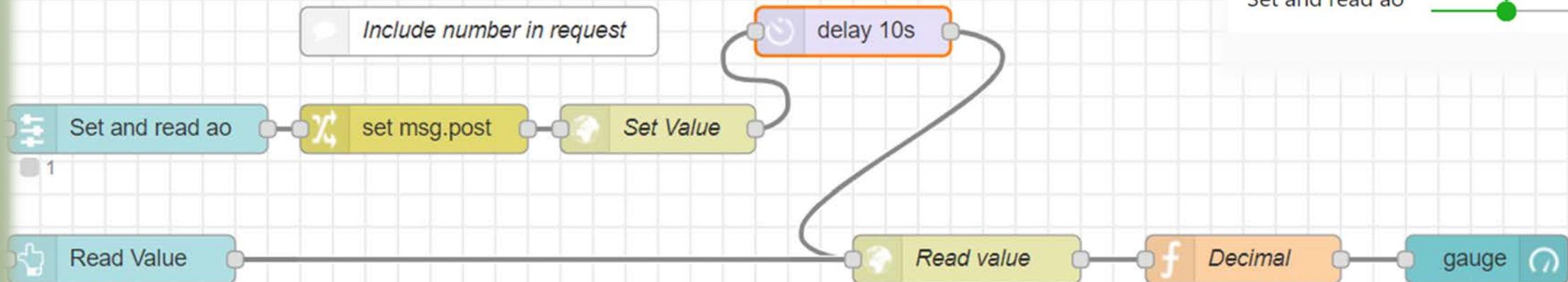
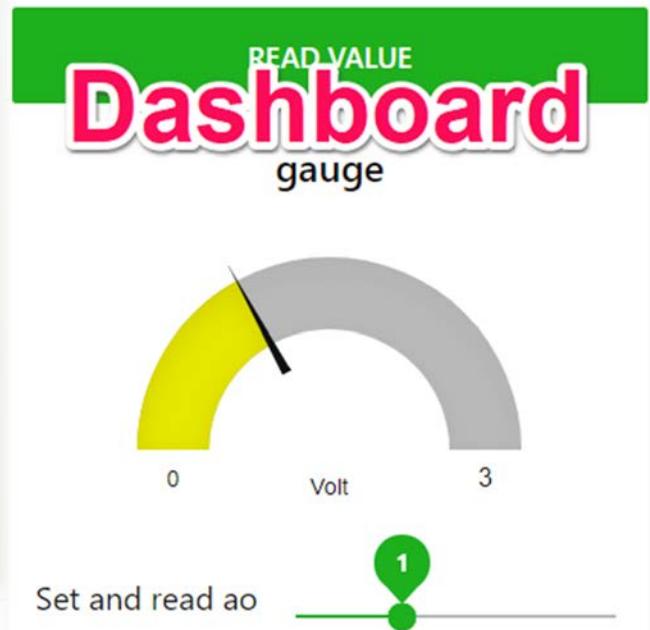


Node-RED

<http://192.168.1.60:1880/>



eedomus\_local\_API



Local API eedomus

Read Ao1 Value