# Proposition de principes

- Une interface, plusieurs usages
- Approche unique pour l'interaction en local ou au travers d'un réseau
- Découplage entre interface graphique et application de mesure

## Proposition d'architecture 1

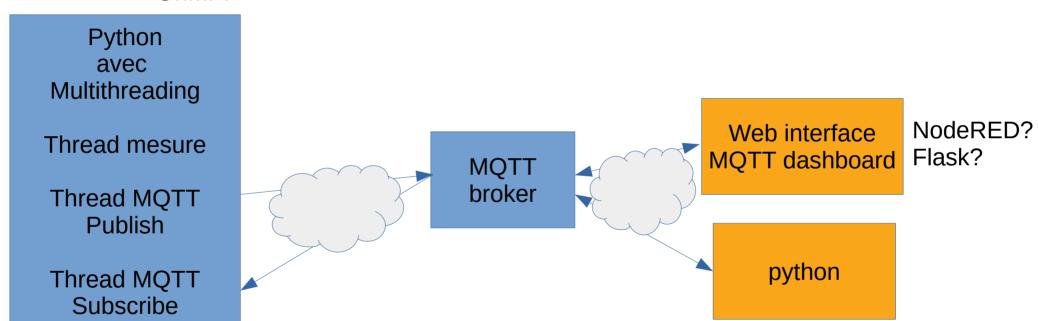
**OhmPi** Client **Python** Set d'instruction et avec de messages simples Flask multithreading Web interface Thread mesure **Thread** Telnet client telnet lecture TCP Port TCP **Thread** écriture TCP Code python dédicacé Exemple: implémentation dans Ardas

## Discussion

- Possibilité de placer l'interface Web sur l'OhmPi, sur un server Web local sur le client ou sur un serveur Web
- Jeu de commandes simples accessibles en terminal
- Large expérience précédente
- Protocole très (trop?) simple
- Sécurité → gestion des accès réseau

# Proposition d'architecture 2

**OhmPi** 



## Discussion

- Possibilité de placer le broker sur l'OhmPi, sur le client local ou sur un serveur
- Standard de l'IOT → potentiel d'interfaçage
- Expérience personnelle plus limitée (mais intérêt)
- Sécurité ?
- Interface Web (développement propre Flask, composant d'un dashboard à la nodeRED (js mais certaines possibilités de développer des fonctions python)?

### Interface architecture concept



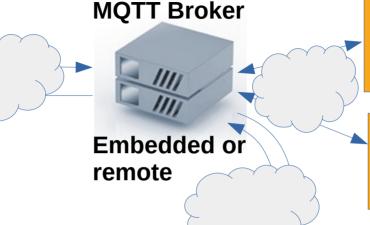
**MQTT Client** 

OhmPi (multithreading) Publishes on data/log/SOH Subscribes to control

#### MQTT Channels:

- → control
- → data
- → log
- → SOH

...



**MQTT Client(s)** 

System controler(s)
Publishes on control &
subscribes to data/log/SOH
example tasks: send a predefined
or adaptative sequence of
measurements, start or stop the
system on alarms (e.g. other
sensor...)

Web interface MQTT dashboard

Python,...

#### **MQTT Clients**

Data consumers/monitors
Subscribe to data/log/SOH
[Publish on additional channels]
example tasks: display the
progress of aquisitions, process
data on messages or conditions,
generate alarms to trigger other
processes, store data/log/SOH
in DB...

### Interface architecture concept

