



*Une architecture pour un système évolutif de
contrôle-commande d'expériences de
physique*

Olivier Taché

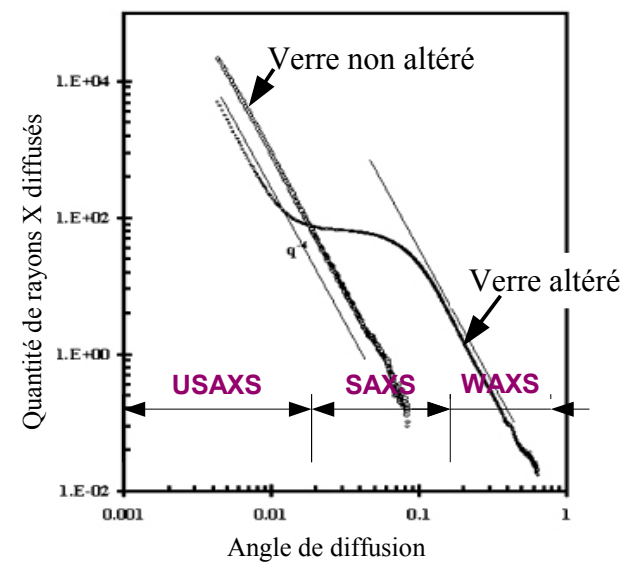
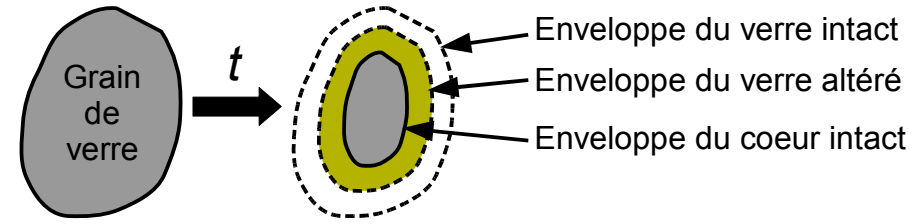
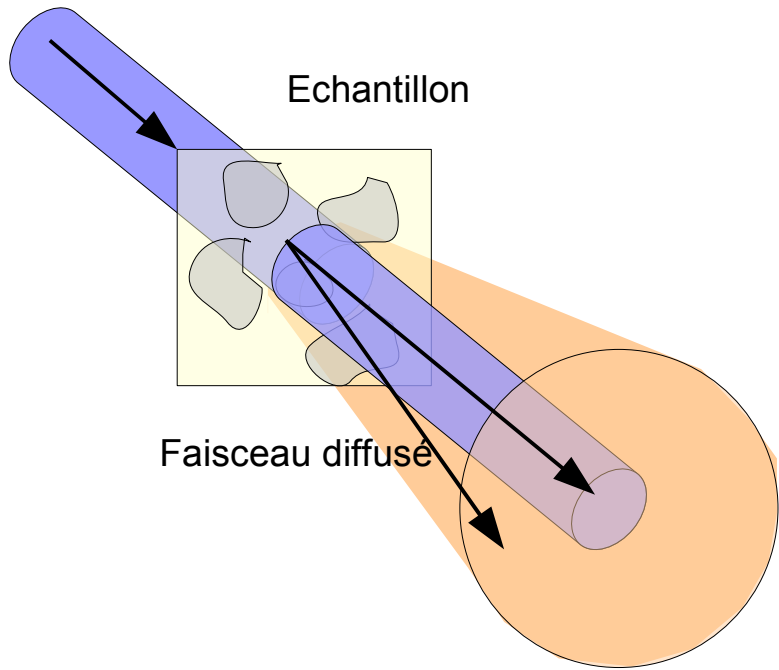
21 décembre 2006

- Contexte
- Problématique
- Architecture
- Couche abstraction Matériel-Logiciel
 - TANGO, principe
 - TANGO, fonctionnement
 - TANGO, outils associés
 - Architecture des *device servers*
- Interface Homme-Machine
 - Choix de la ligne de commande
 - Python et TANGO
 - Accès à TANGO
 - Développement de la logique d'expérience
- Conclusion et perspectives

Diffusion des rayons X aux petits angles

Principe et application :

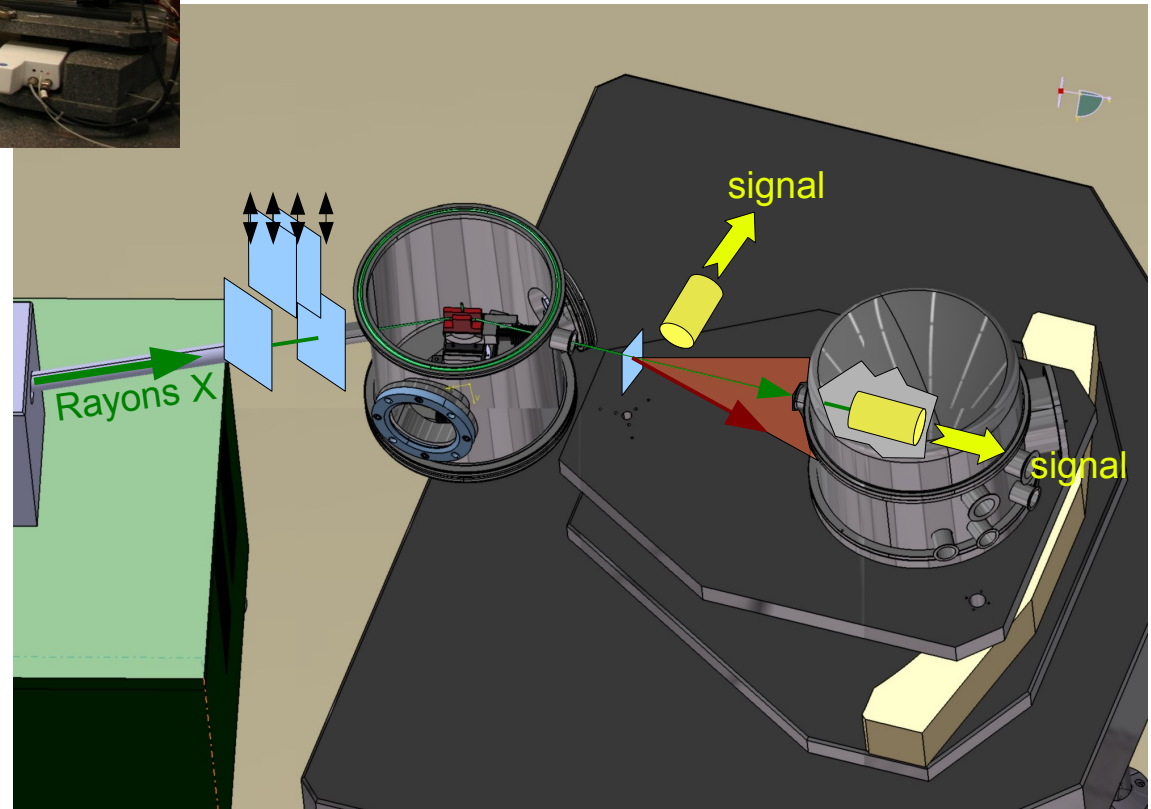
Rayons X



Diffusion des rayons X aux très petits angles



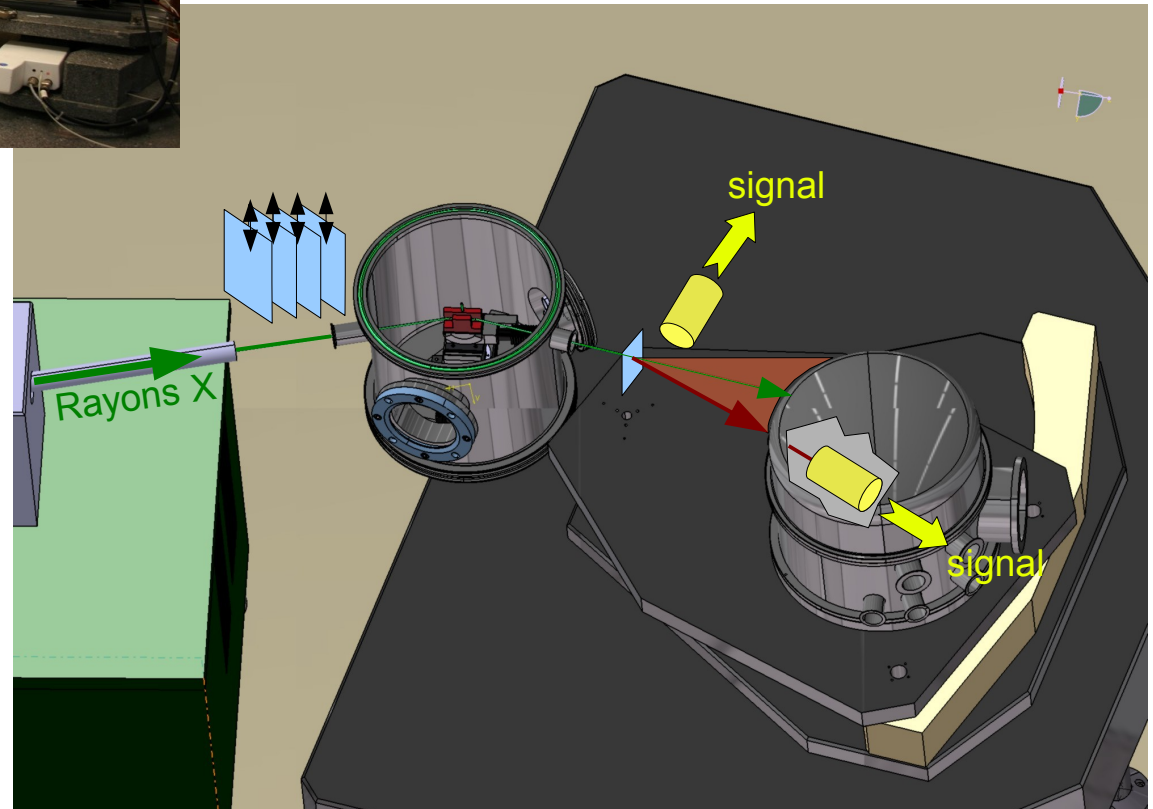
Le montage expérimental modèle :
USAXS
(Ultra Small Angle X-Ray Scattering)



Diffusion des rayons X aux très petits angles



Le montage expérimental modèle :
USAXS
(Ultra Small Angle X-Ray Scattering)



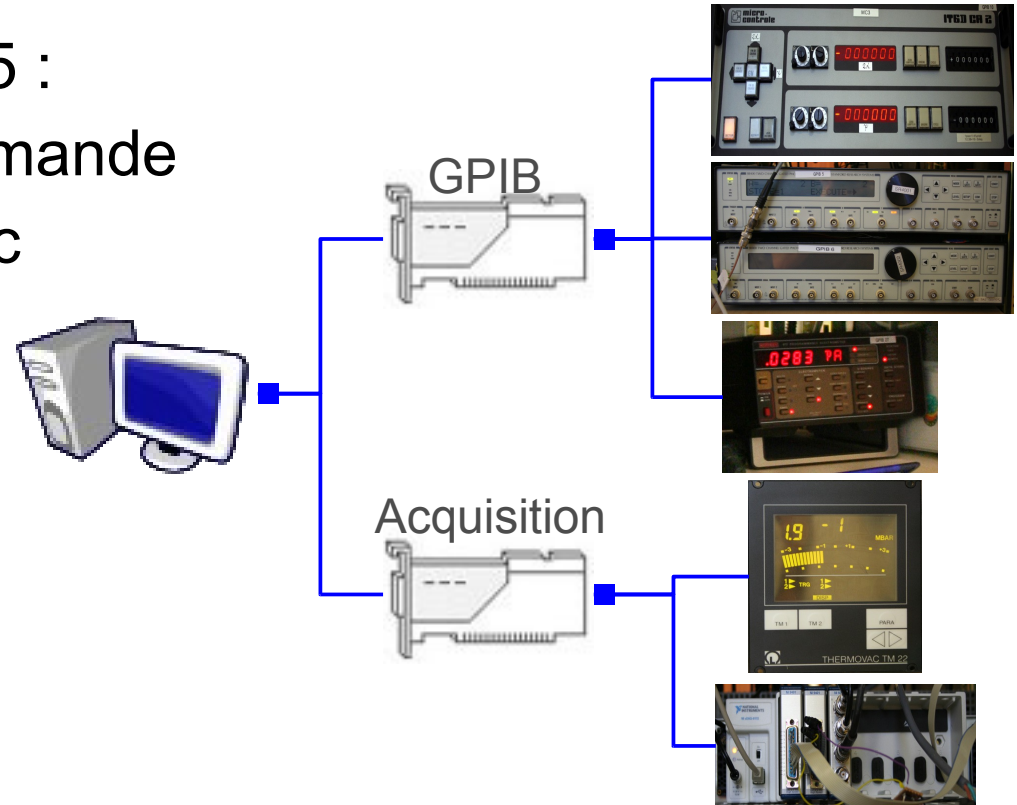
Système informatique 2005 :

- logiciel de contrôle-commande programmé en Visual Basic

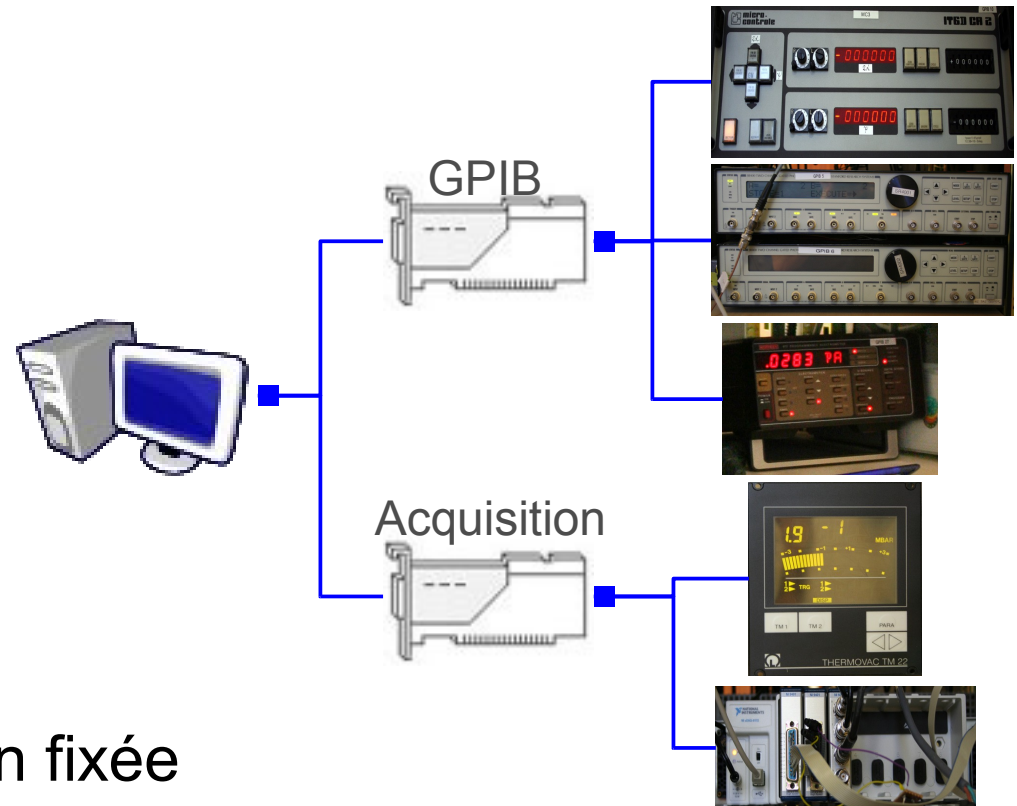
- Appels au matériels programmés en dur (commandes GPIB)

- Très peu de paramétrage du matériel

- Très peu de paramétrage des séquences d'acquisition (réglages et expérience de routine)



- Matériel hétérogène
- Modification du matériel
- Étendre les possibilités



- Logique d'expérience non fixée
- Plusieurs niveaux d'interaction utilisateurs
- Déploiement sur d'autres expériences

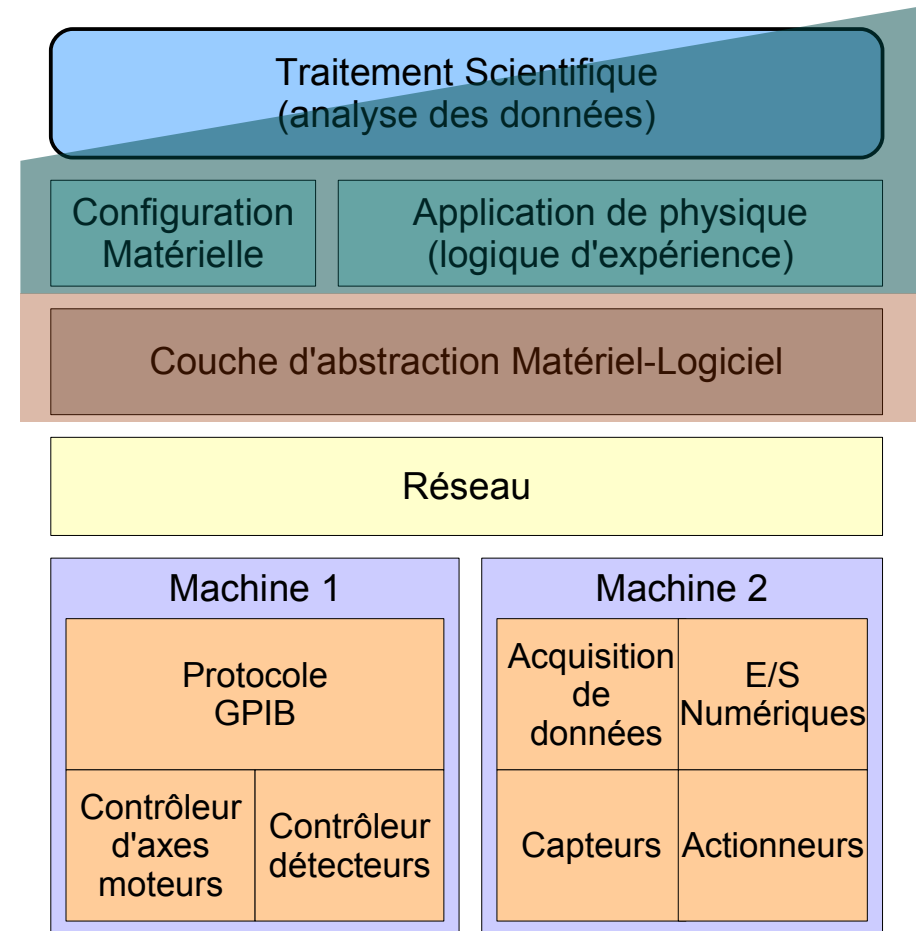
Python

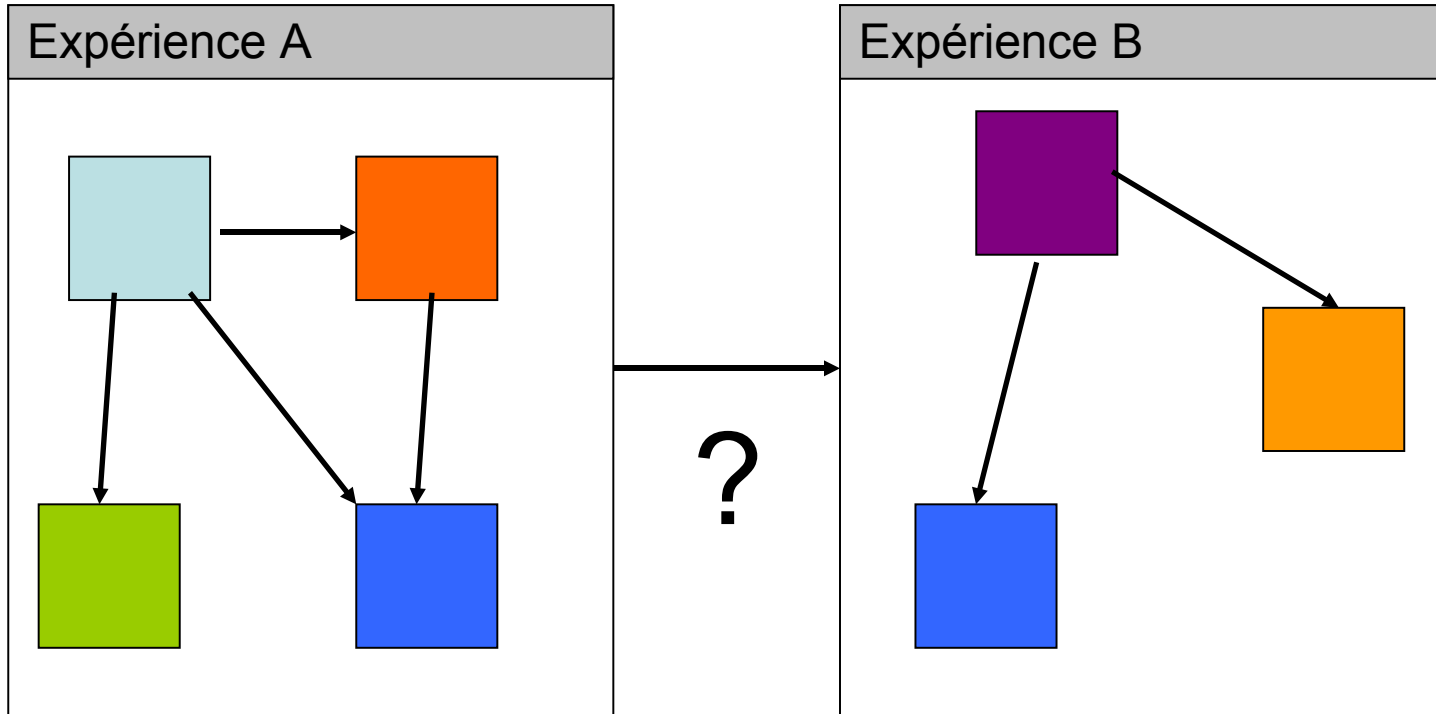
TANGO

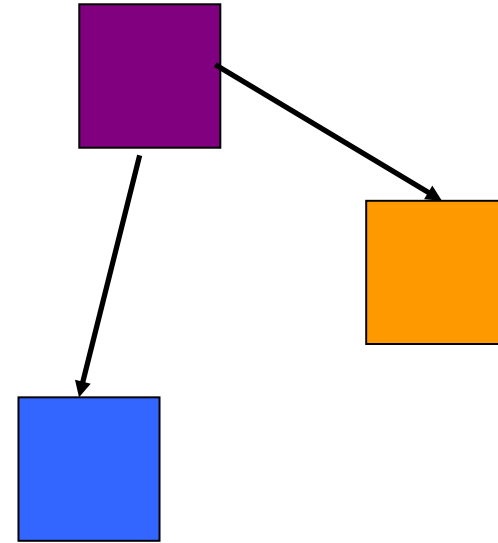
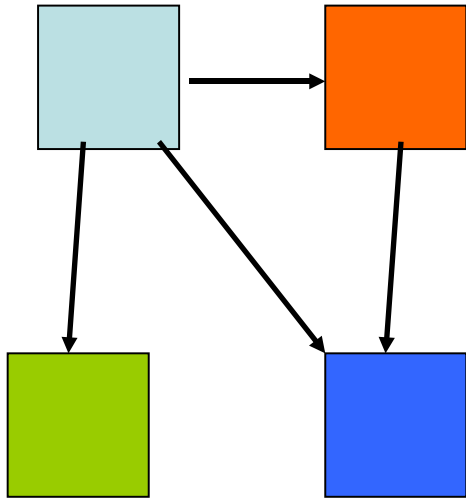
*Système de contrôle-commande
de l'ESRF, Soleil, ...*

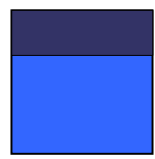
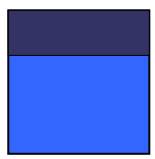
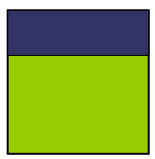
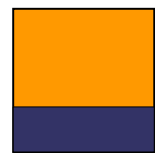
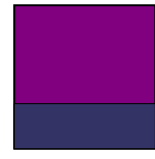
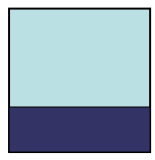
Séparer l'application finale du matériel :

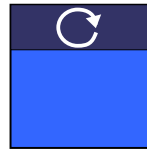
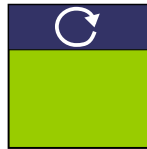
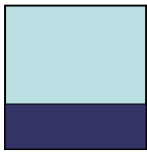
- une couche Abstraction Matériel Logiciel
- une interface Homme-Machine

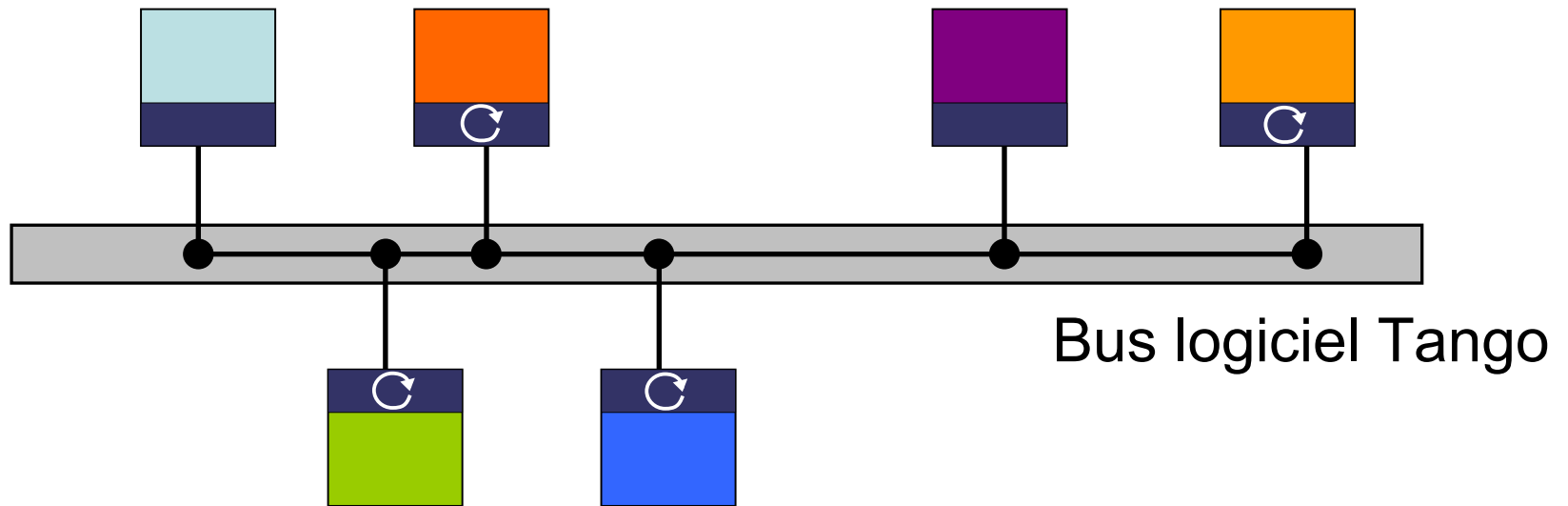




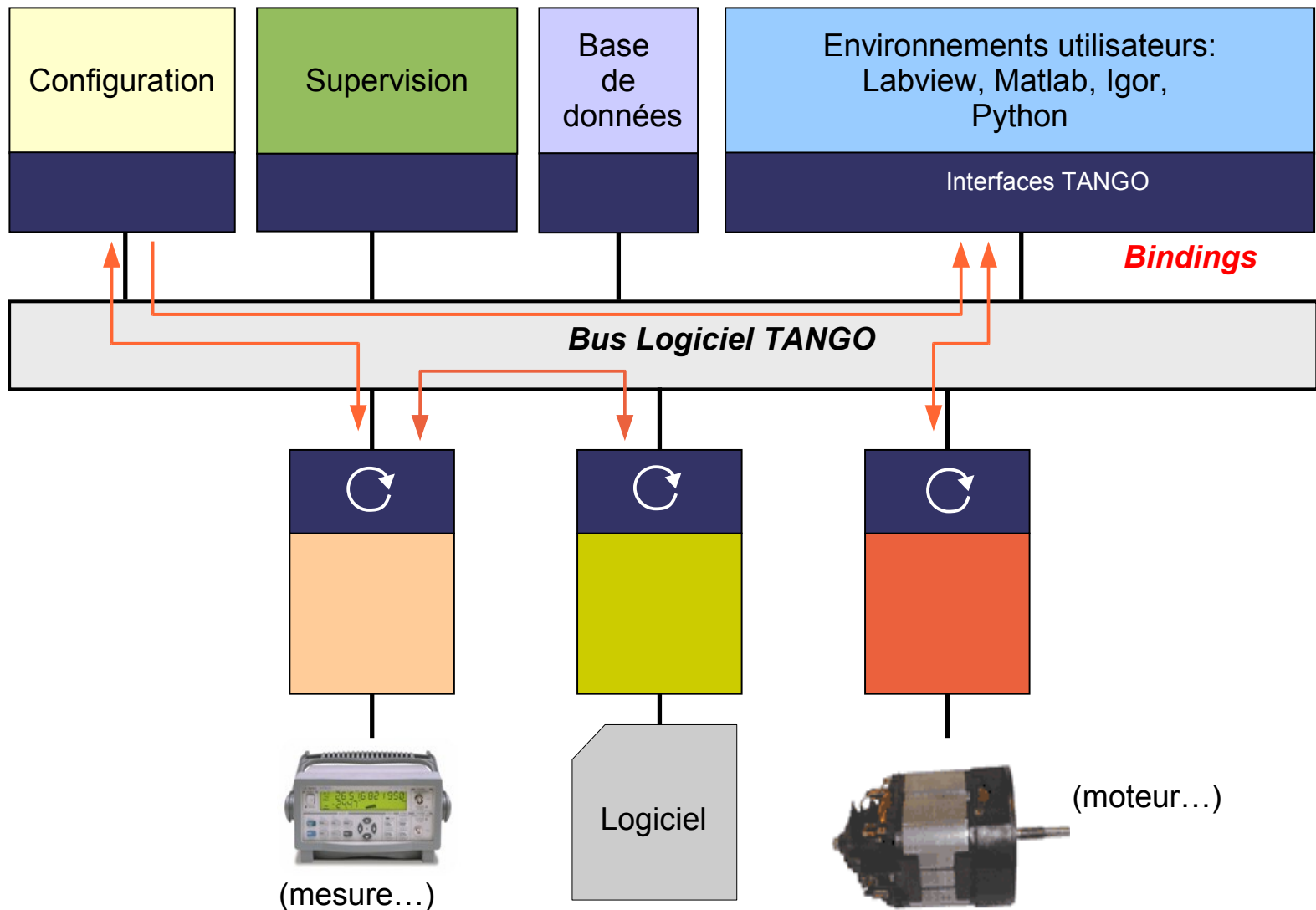






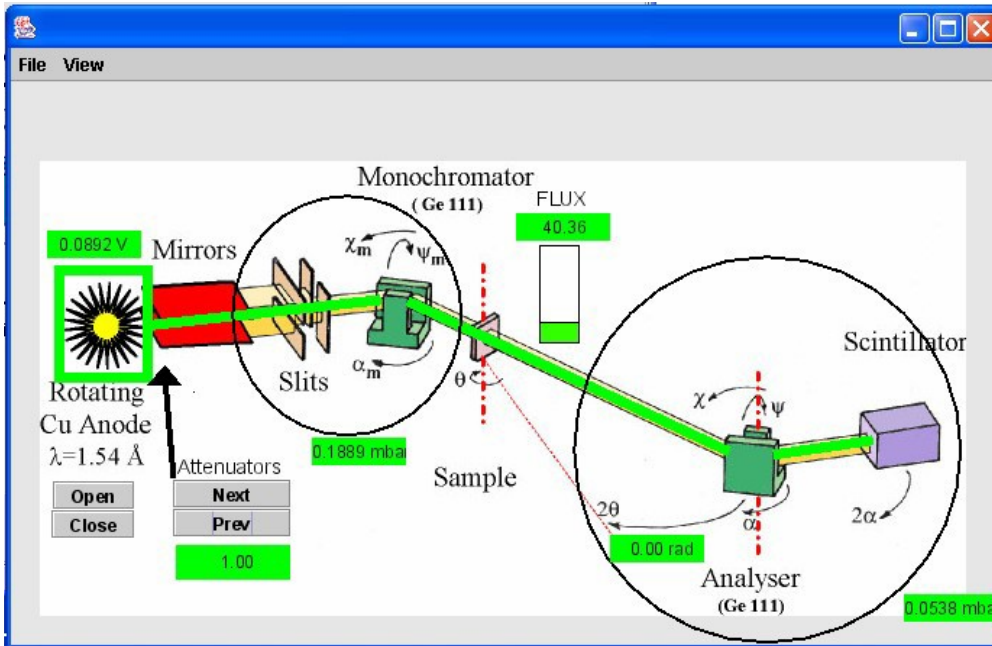


Bus logiciel : système permettant aux applications de communiquer

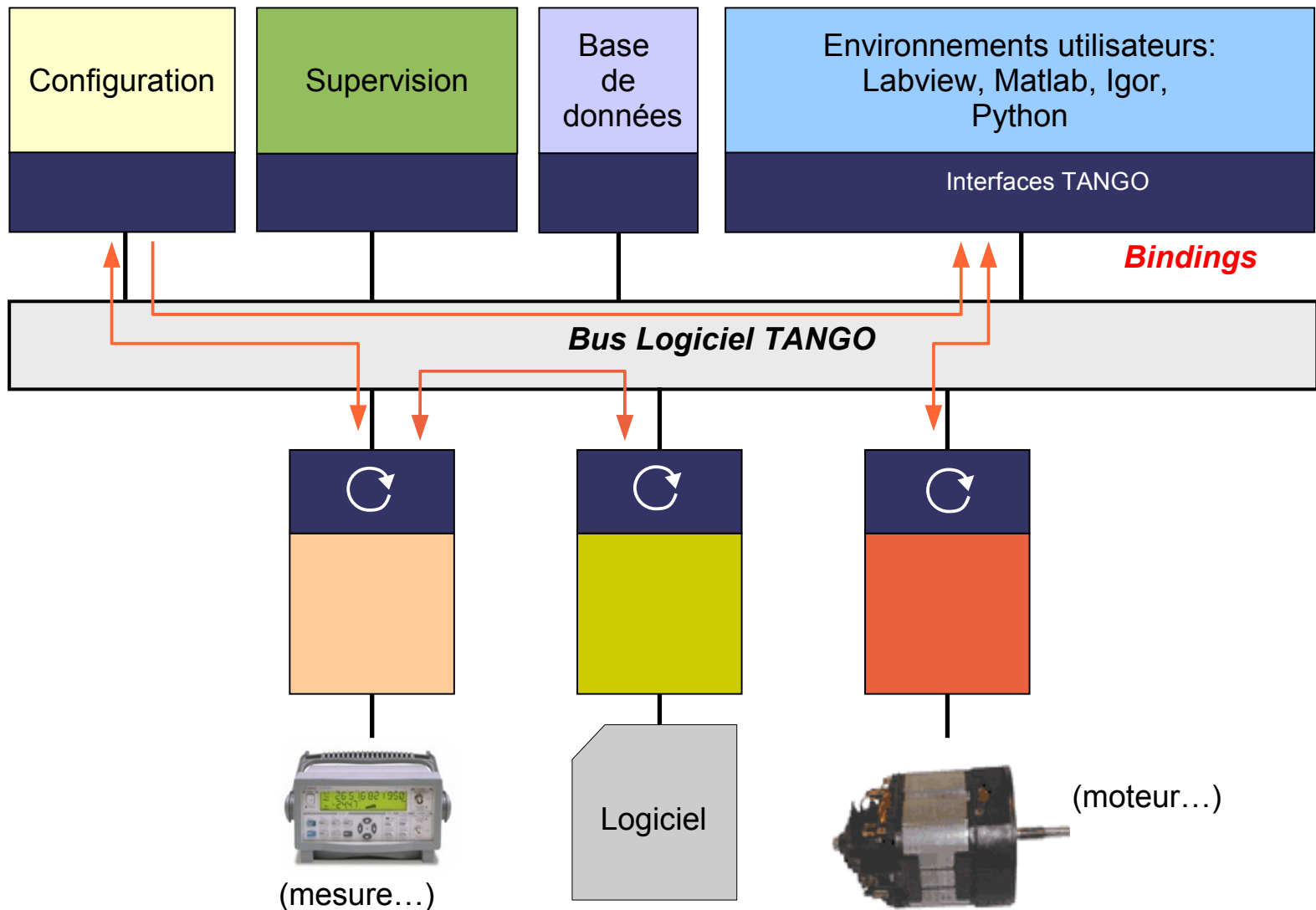


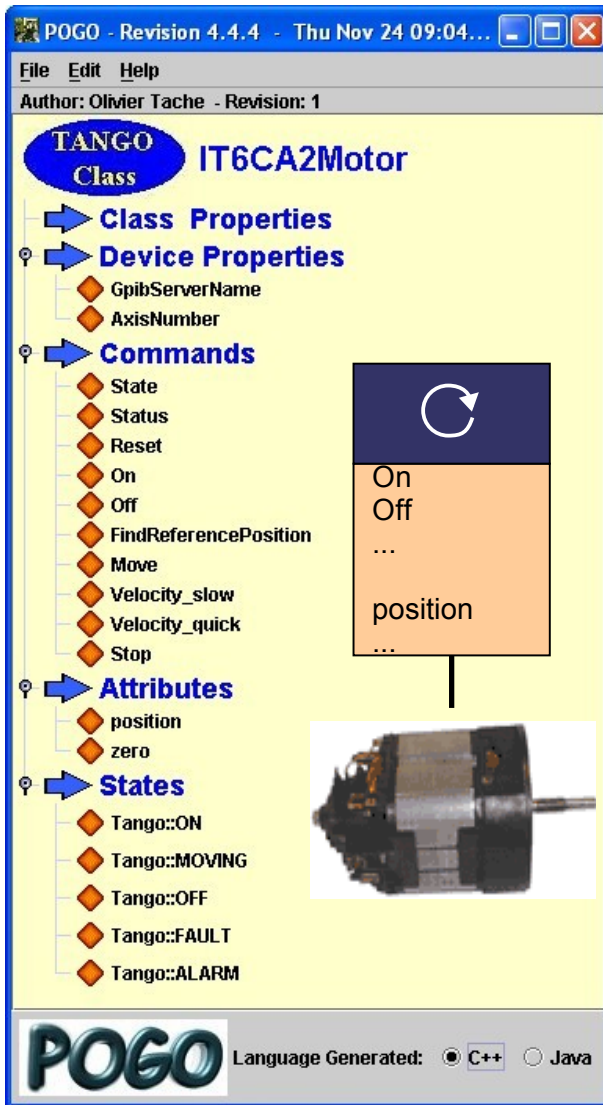
Outils graphiques pour :

- Démarrage
- Configuration
- Supervision
- Synoptique



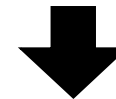
➔ Plusieurs représentations d'un Device Server



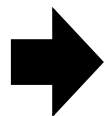
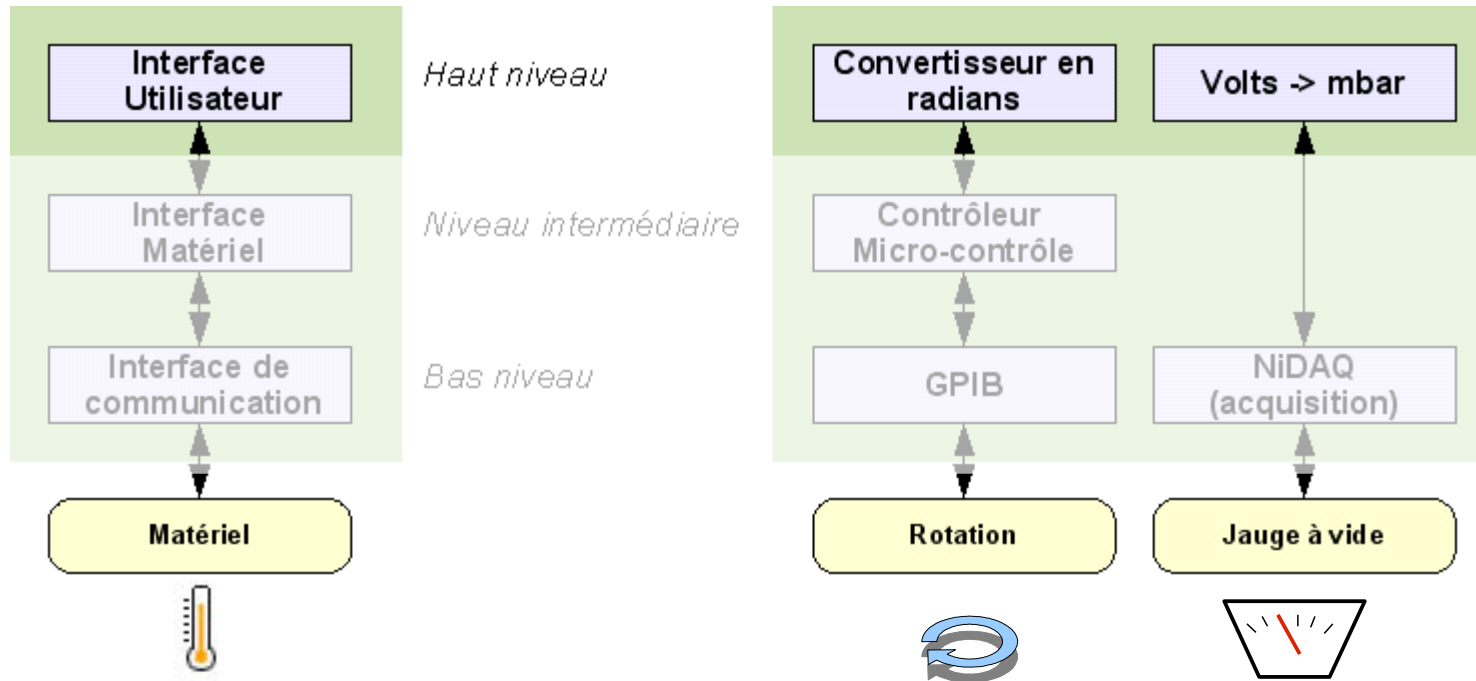


DEVICE SERVER :

- Programmé en C++, Java, maintenant Python
- A la charge du développeur
- Mais générateur de code
- Existe des bibliothèques de Device Servers



Pour pouvoir les réutiliser,
il faut développer des DS simples et modulaires



Véritable abstraction du matériel : l'utilisateur final accède à des variables en unités scientifiques

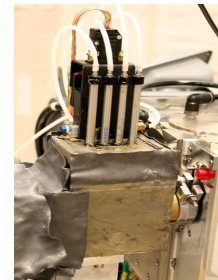
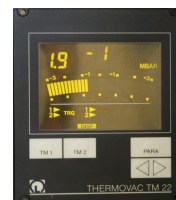
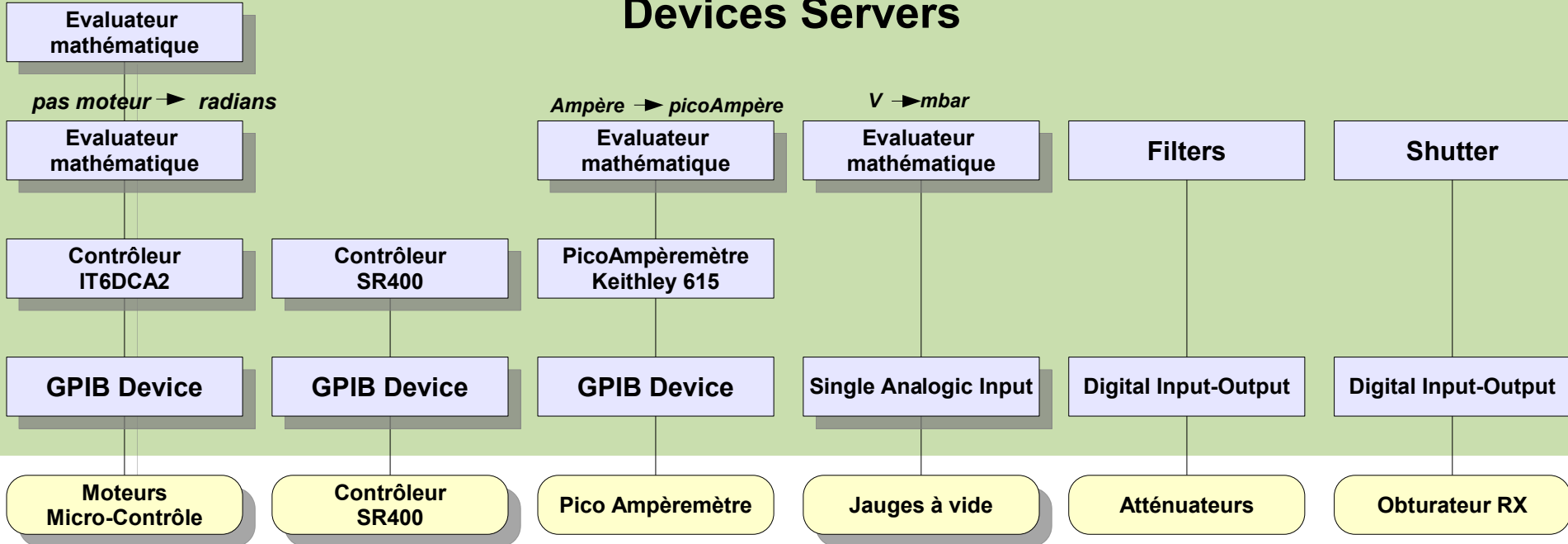
radians → *vecteur d'onde*

Devices Servers

pas moteur → *radians*

Ampère → *picoAmpère*

V → *mbar*



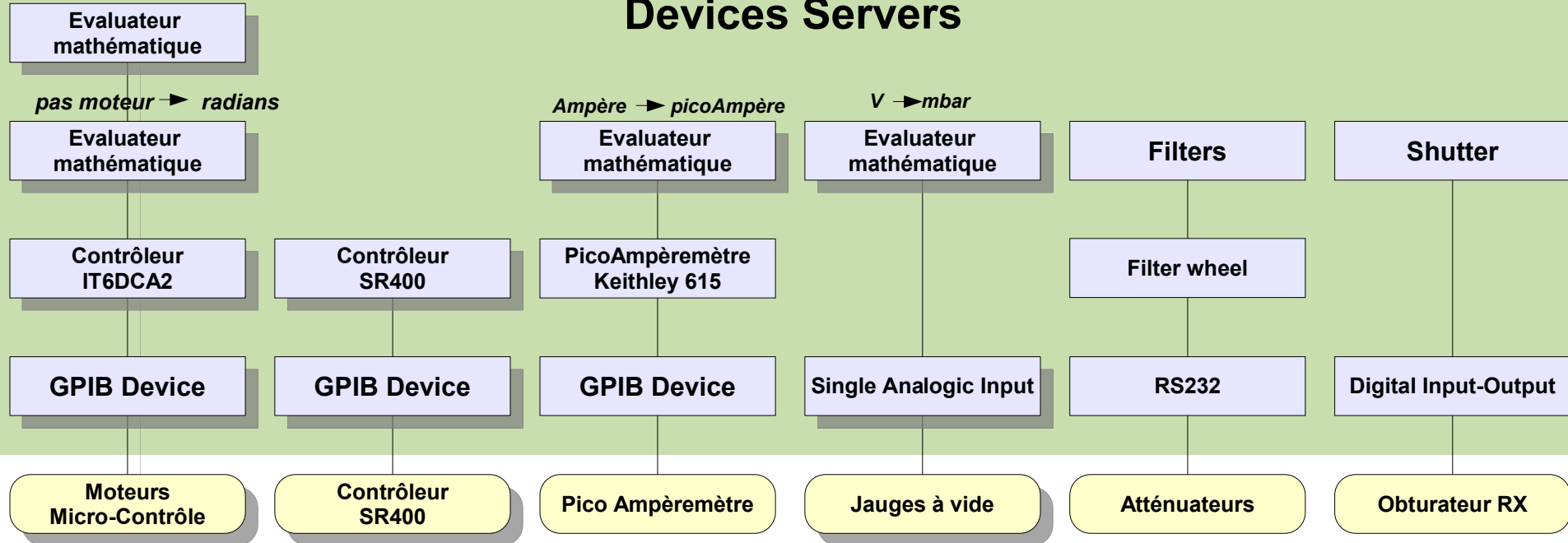
radians → *vecteur d'onde*

Devices Servers

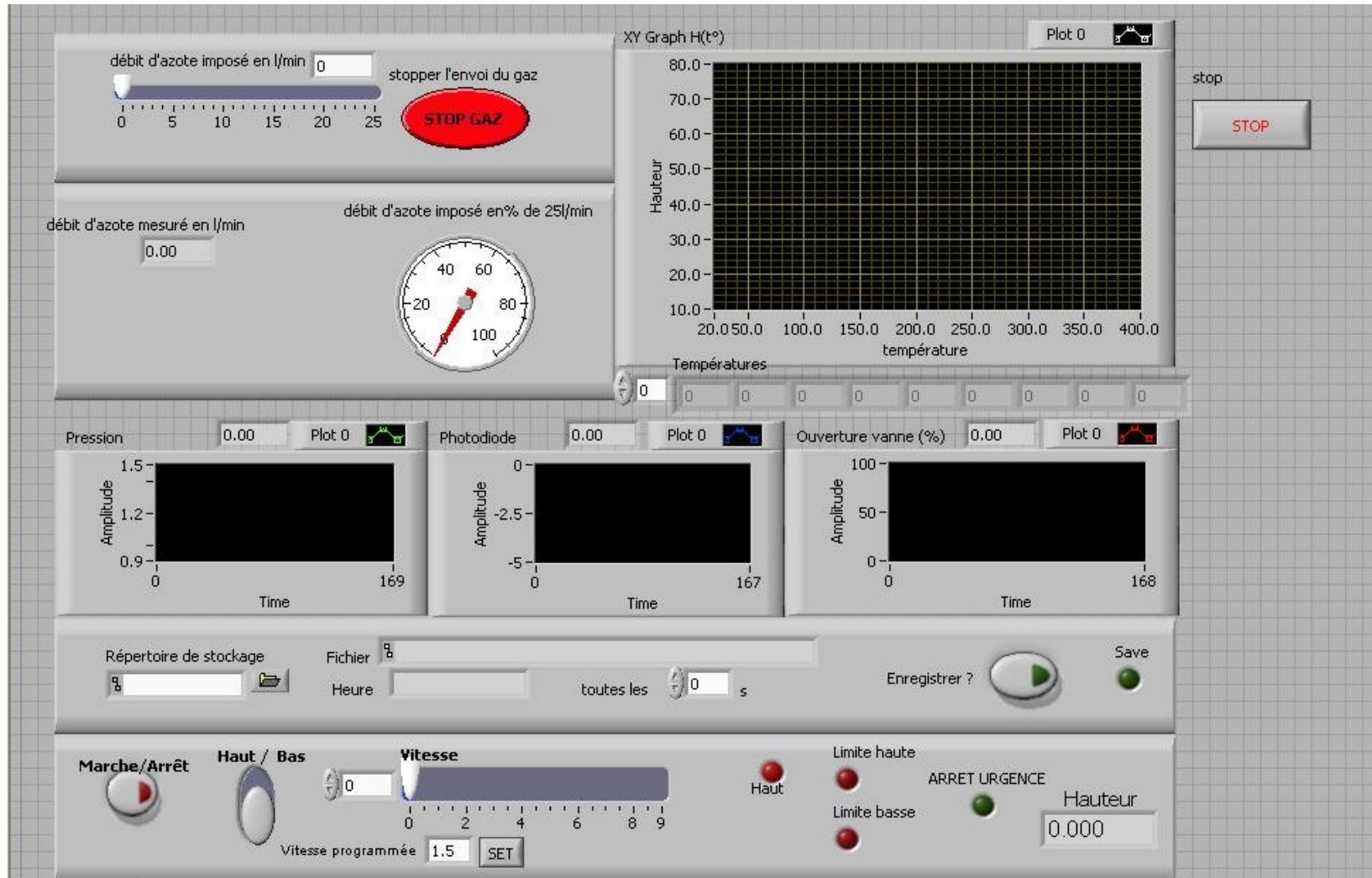
pas moteur → *radians*

Ampère → *picoAmpère*

V → *mbar*



Quelle interface Homme-Machine pour un tel système ?



Choix de la ligne de commande :

```
Python Shell
File Edit Shell Debug Options Windows Help
14:38:36      23.828125
14:38:37      23.9013671875
14:38:37      23.876953125
14:38:37      23.876953125
14:38:37      23.8525390625
14:38:37      23.828125
14:38:37      23.8525390625
14:38:37      23.8525390625
14:38:37      23.8525390625
14:38:38      23.828125
14:38:38      23.8525390625
>>> demo
found : demo/daqpad/input      ->input  : ON
found : demo/daqpad/led ->led   : ON
found : demo/daqpad/output     ->output  : ON
found : demo/physic/motor      ->motor   : ON
found : demo/physic/therm      ->therm   : ON
found : demo/simul/simul       ->simul   : ON

>>> demo.led.Value=5
>>> demo.led.Value=0
>>>
```

Ergonomie sommaire...

MAIS

- Logique d'expérience évolutive
- Traçabilité des commandes

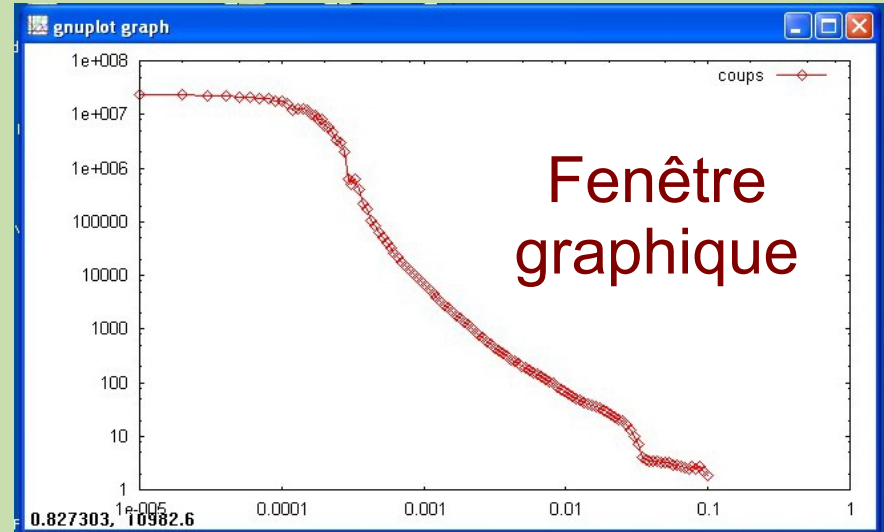


```
>>>usaxs.shutter.Open()
>>>usaxs.theta.Value
0.001
```

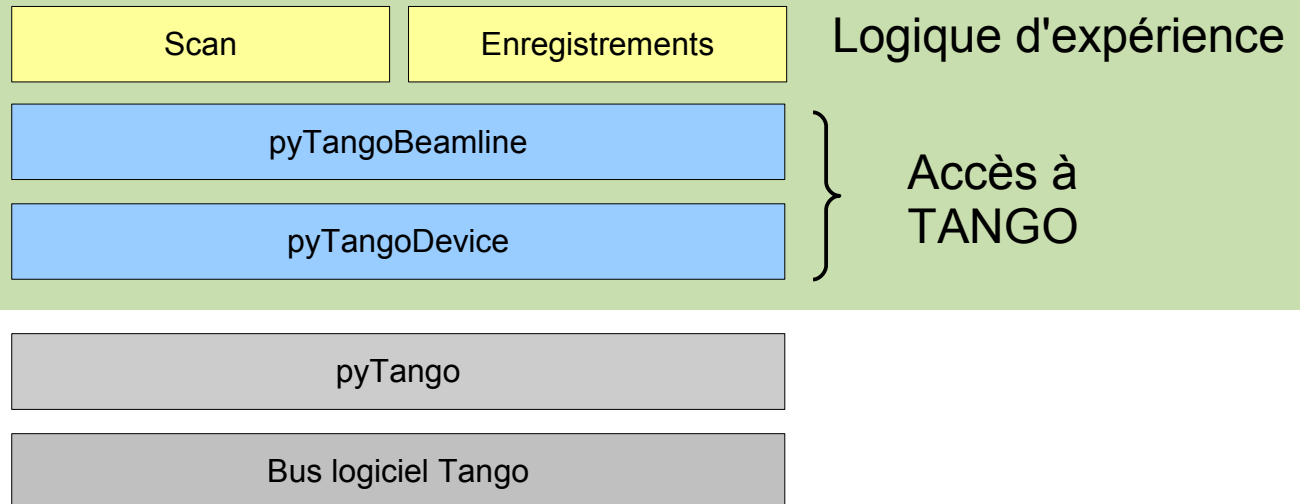
- Langage Python :
 - Open Source et gratuit
 - Portable
 - Syntaxe simple et facile mais types de données évolués
 - Orienté objet
 - Bibliothèques scientifiques et graphiques
 - **Interpréteur avec ligne de commande**
- PyTANGO
 - Librairie TANGO pour Python
 - Applications clientes et serveur
- Gnuplot
 - Pour le tracé des courbes

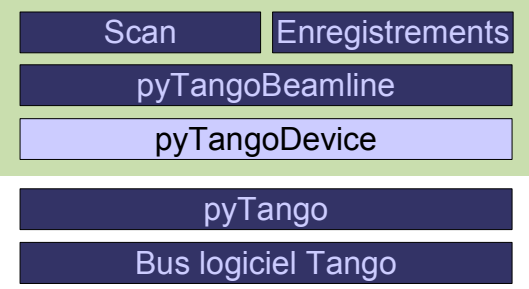
```
Python Shell
File Edit Shell Debug Options Windows Help
0.0501319322368 - 3.35215946844 1.0
measured : 1
0.0533797784869 - 3.17940199336 1.0
measured : 3
0.0568380396322 - 2.87707641196 1.0
measured : 1
0.0605203476073 - 2.8903654485 1.0
measured : 9
0.0644412175053 - 2.68438538206 1.0
measured : 8
0.0686161047935 - 2.6511627907 1.0
0.0730614662369 - 2.4584717608 1.0
measured : 1
0.0777948247683 - 2.71096345515 1.0
measured : 5
0.0828348385605 - 2.51827242525 1.0
measured : 7
0.0882013745745 - 2.82724252492 1.0
measured : 2
0.0939155868717 - 2.1561461794 1.0
measured : 1
0.1 - 1.88704318937 1.0
Finished
End
>>>
```

Fenêtre de saisie



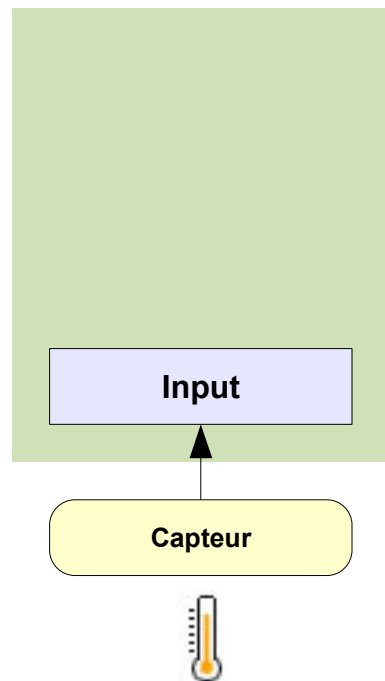
Fenêtre graphique

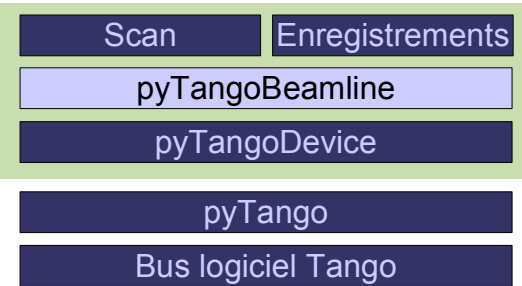




Module pyTangoDevice

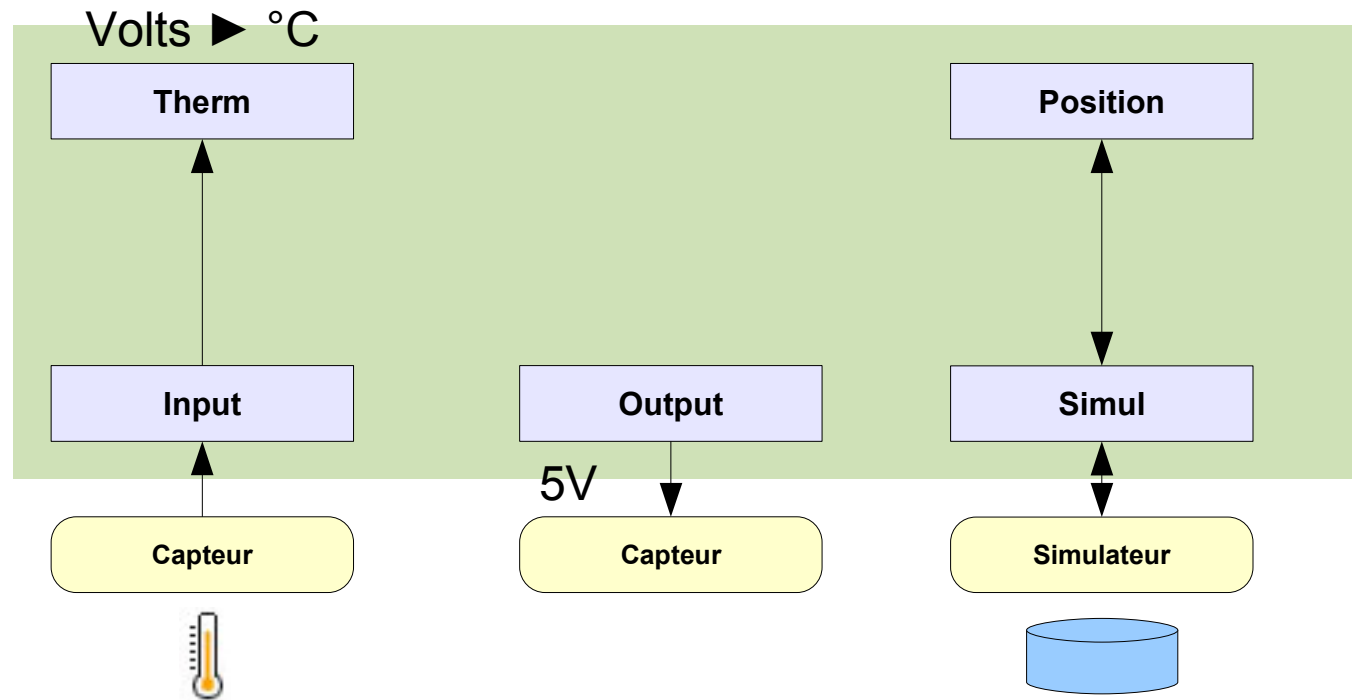
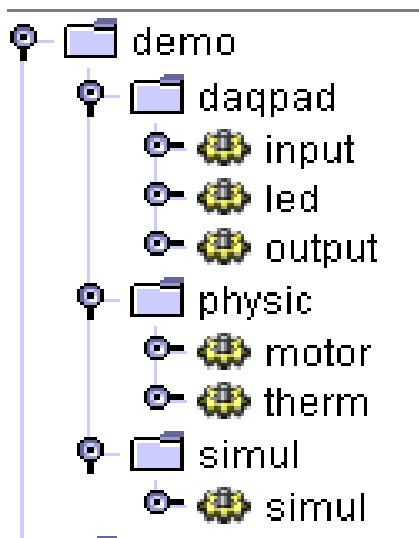
Pour accéder à des devices TANGO à l'aide de variables et commandes Python

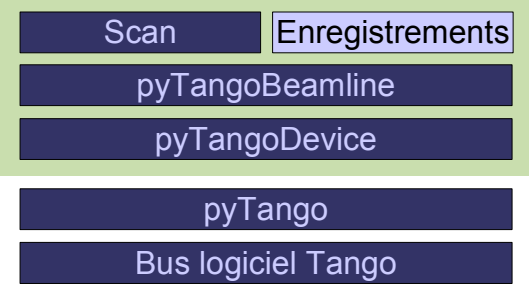




Module pyTangoBeamline

Pour accéder à un ensemble de devices TANGO à l'aide de variables et commandes Python

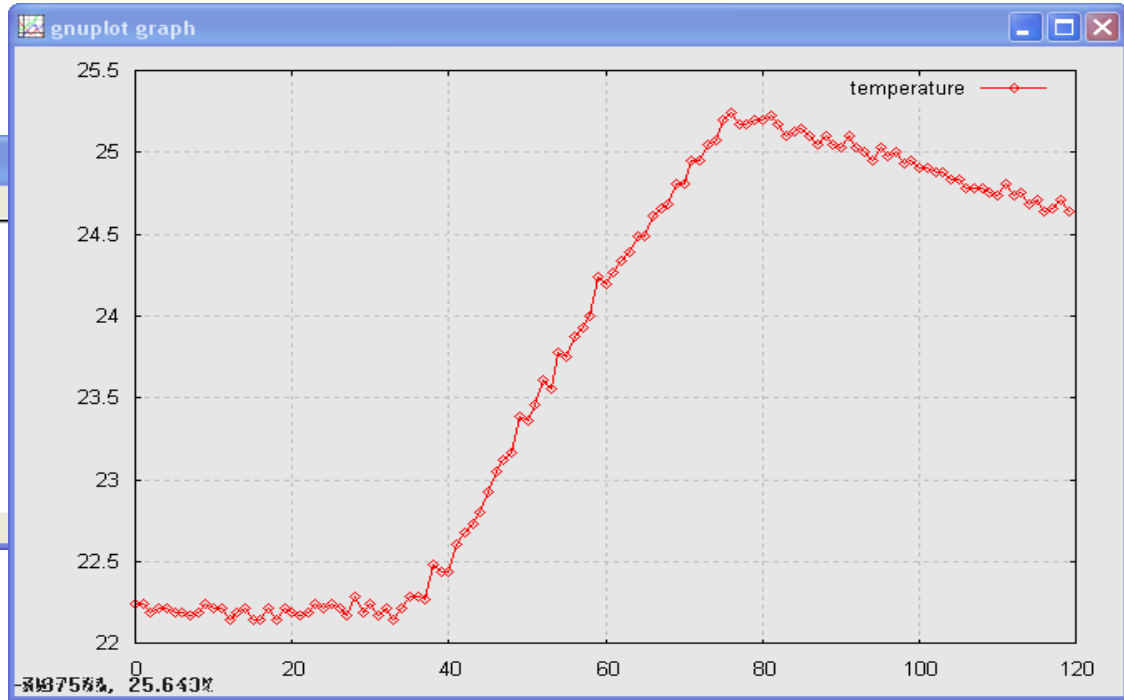


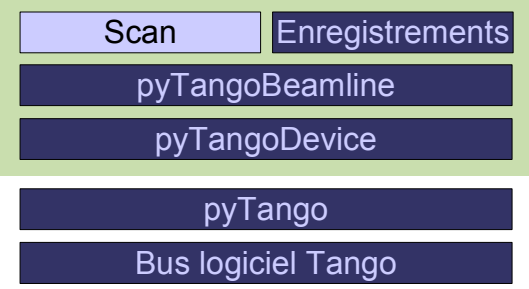


Module d'enregistrement

Pour réaliser des enregistrements

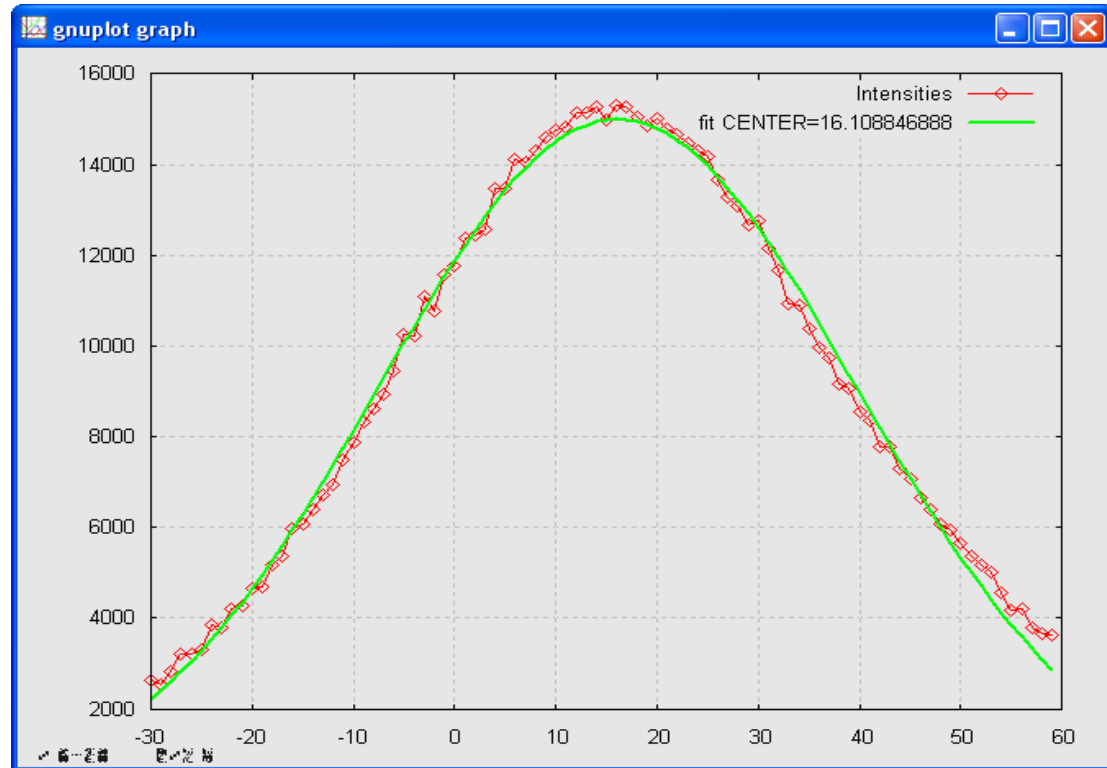
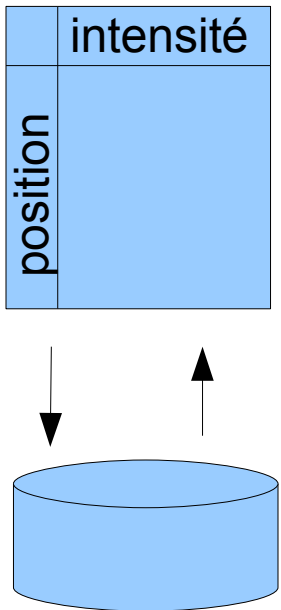
```
demoR.py - C:\Python24\demoR.py
File Edit Format Run Options Windows Help
from pyTangoBeamline import *
demo=Beamline("demo")
from pyRecorder import *
r=pyRecorder()
r.addMeasure("temperature", demo.therm.get_Value)
r.start(many=120, scheduler=0)
```

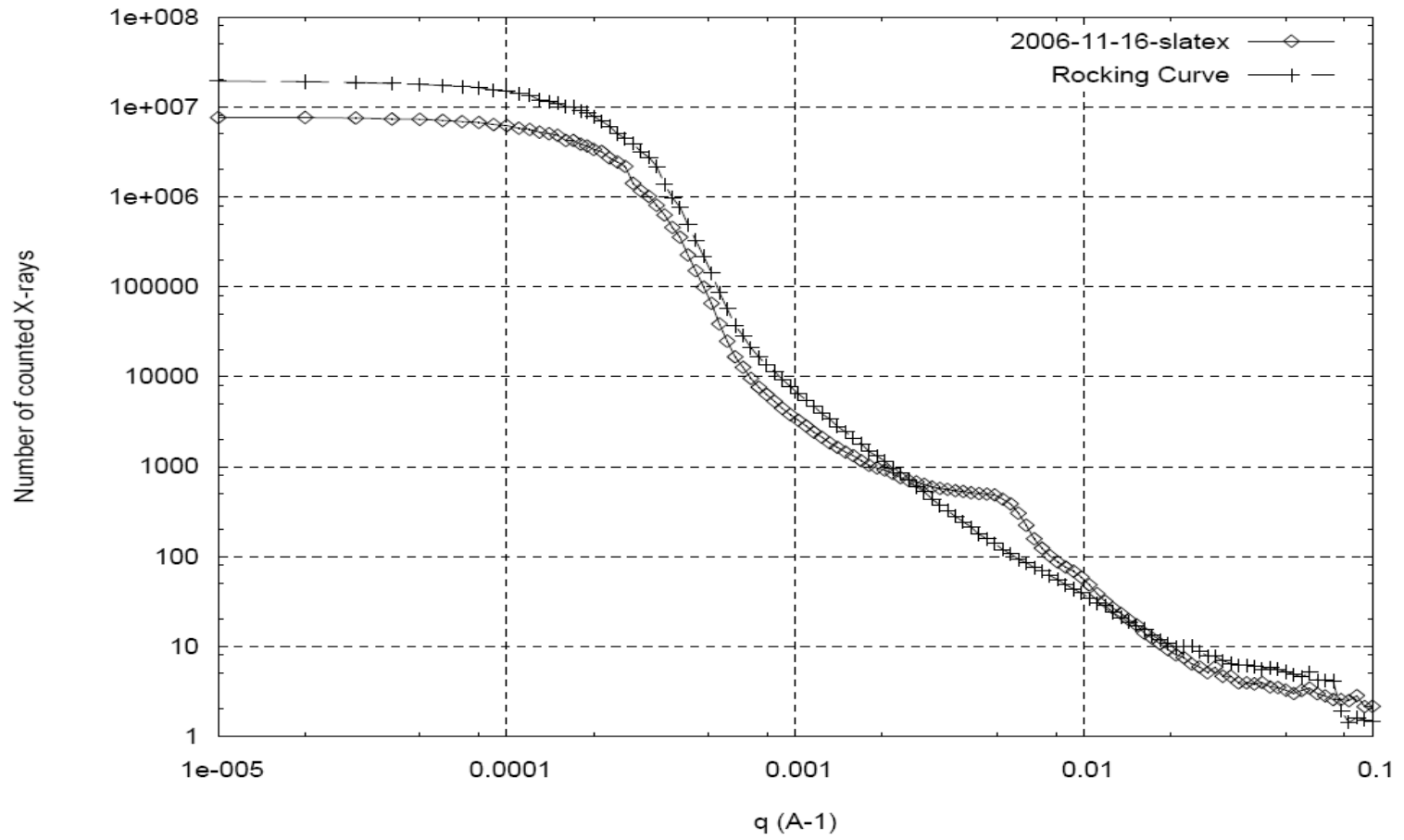




Module de scan

Pour réaliser des « scans »





- TANGO
 - Gestion de tous les types de matériel
 - Modulaire et robuste
 - Développement rapide des DS
 - Maintenance des serveurs en Python
- Interface Homme Machine
 - Logique d'expérience par les physiciens
 - Outils scientifiques
 - ▶ développer une interface graphique pour les expériences de routine
 - ▶ Travailler sur l'ergonomie

Originalité ?

- Ligne de commande pour TANGO
- Basé sur des composants Open Source
- Diffusé librement (www-drecam.cea.fr/scm/lions)
- Pas de barrières entre contrôle-commande et applications scientifiques (bibliothèques de traitement et analyse des données)

Perspectives :

- Évolutions sur le montage USAXS
(nouveaux appareillages)
- déploiement sur d'autres expériences