Laboratoire d'Annecy-le-Vieux
de Physique des Particules

OPCUA : Nouvelle approche pour l'intégration du slow control dans une expérience

E. Chabanne, Th. Le Flour, J.L. Panazol

1ere journée dispositifs et installations instrumentation IN2P3
18 juin 2015

In2p3

Sommaire

- L'Instrumentaliste et le slow control
- Intégration HW/SW dans une architecture de slow control?
  - Hétérogénéité
  - HW/SW interface
  - IHM et outils
  - Exemple : l'expérience SuperNEMO
- Le standard OPCUA en quelques mots?
- Au-delà d'OPCUA : comment se simplifier la vie d'instrumentaliste?
- Un exemple sur plateforme ARM
- Conclusions

Quelques remarques générales

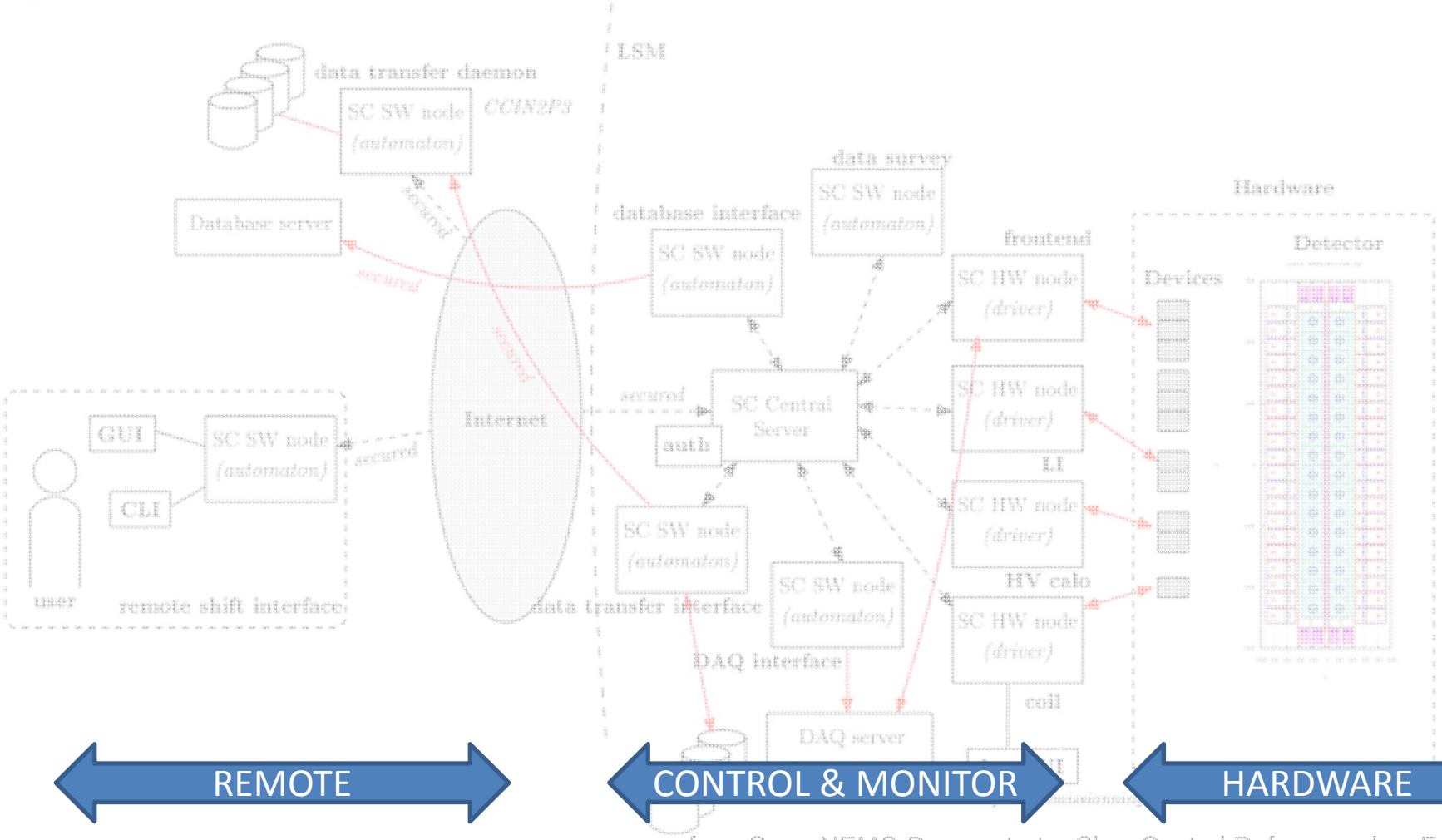
- En fonction du niveau d'expertise, un utilisateur doit connaitre l'état courant des matériels (localement et à distance)
  - –Hiérarchisation de l'information
  - –Agrégation multi-matériels d'un même type d'information
- L'accès à ce type d'information doit être également pris en compte
  - – Réseau local et réseau étendu
  - –Sécurité (firewall, réseau et données)
- Plusieurs type de données « Slow Control » coexistent :
  - – Celui lié à la qualité de la donnée physique.
  - – Celui lié à l'information/exploitation du HW/SW (statuts, taux d'utilisation, fréquence de panne)
    - •Utilisable directement par des outils de diagnostic.

Quelques remarques générales

Lorsque l'on conçoit une carte électronique:

- Il est rare qu'elle fonctionne seule (sans lien de communication)
- Dans la majorité des cas elle communique avec 1 ou des clients
- Pour la partie «slow control/monitoring»
  - Lire et écrire pour accéder à la carte (exemple registre interne pour la configuration)
  - Lire et écrire les capteurs/actionneurs gérés par la carte.
  - La carte peut gérer les capteurs/actionneurs directement ou elle même déporter la gestion via des bus (exemple : I2C, RS232, CAN ….)
- Et la partie application de supervision /monitoring ?
  - Une application par type de matériel, par protocole ?
  - Une application capable de gérer plusieurs matériels?
  - Un SCADA?

Exemple d'architecture slow control pour l'expérience SuperNEMO (neutrinos)

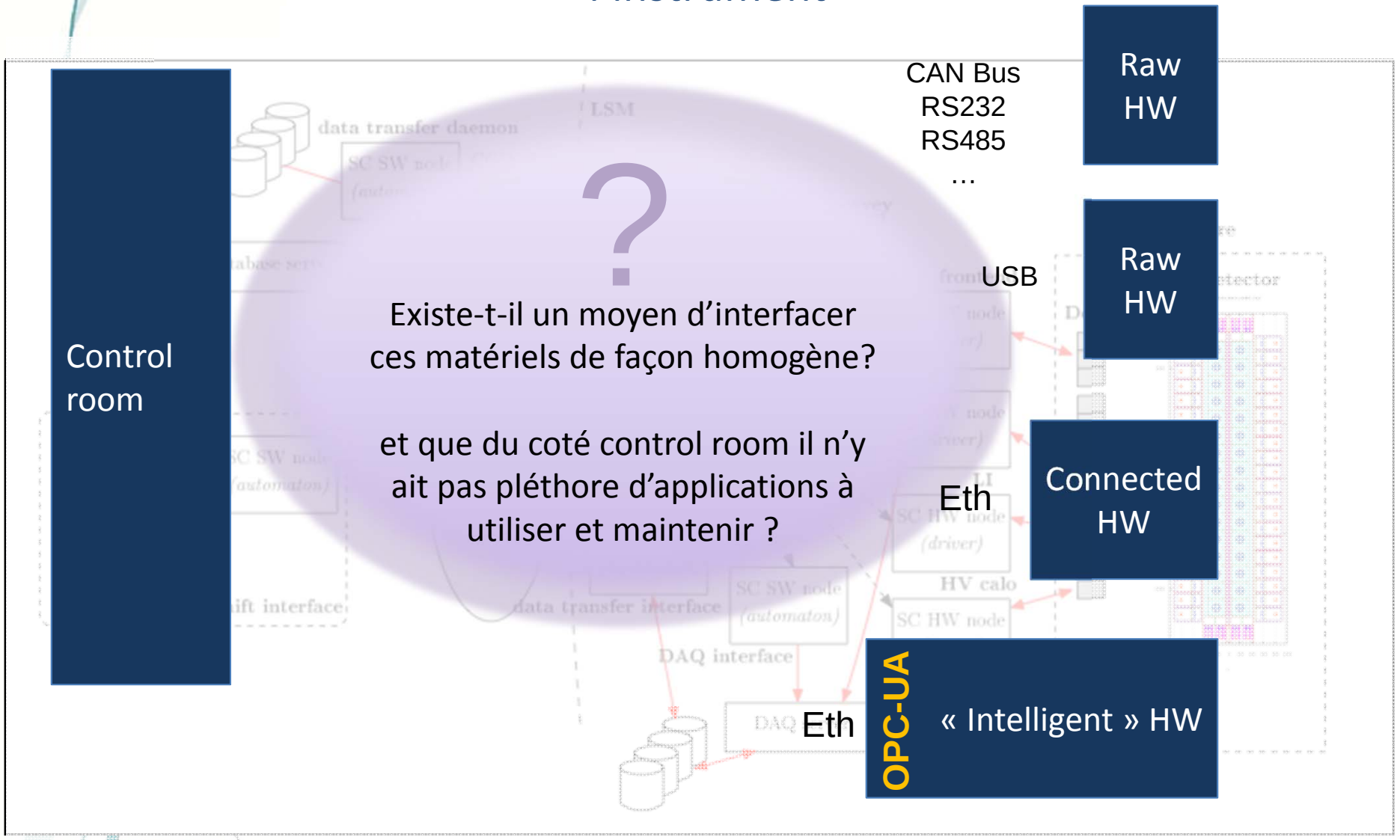


from SuperNEMO Demonstrator Slow Control Reference doc. F. Mauger

Plusieurs sous-systèmes identifiés dans l'instrument à contrôler

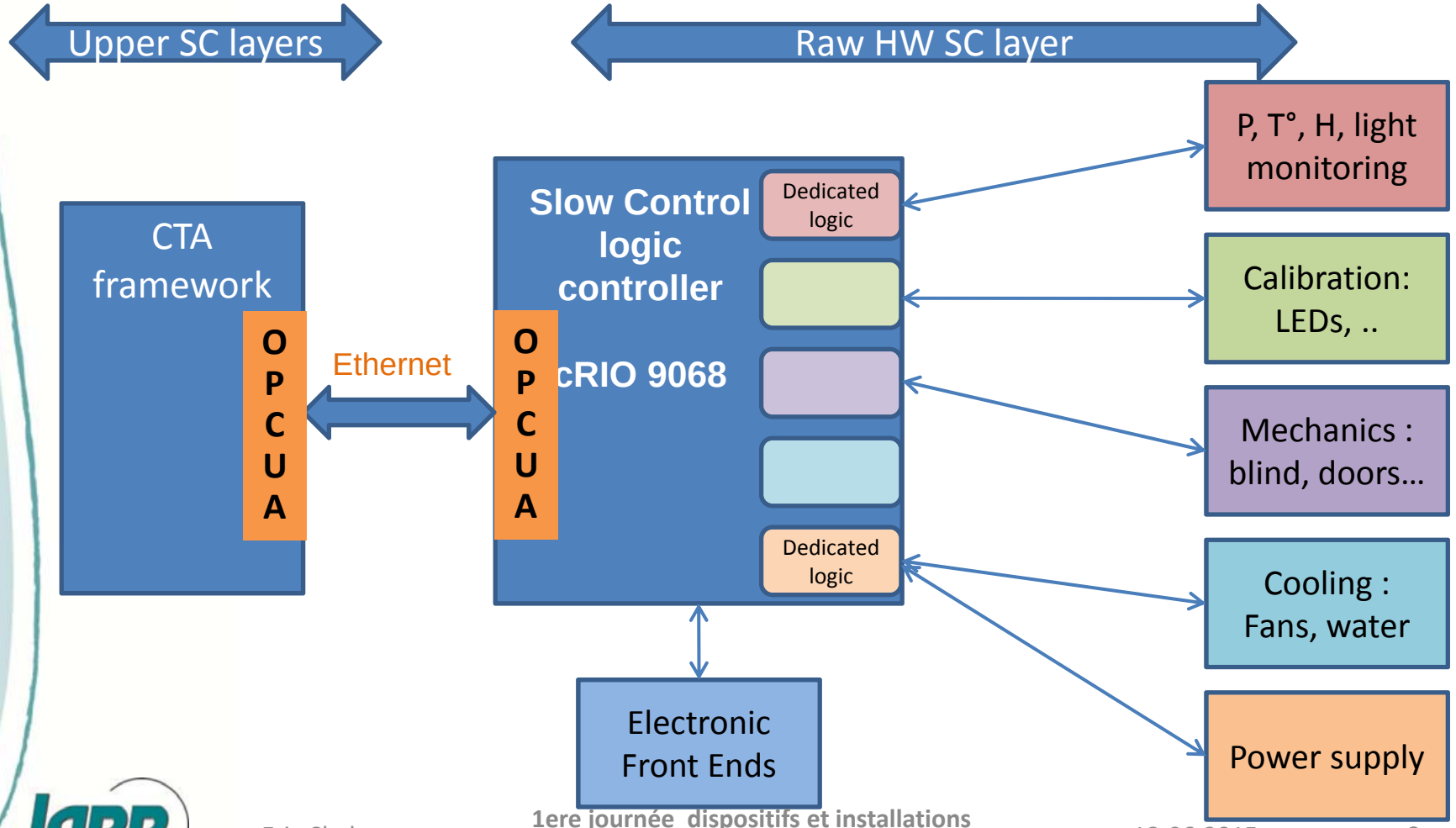
- Front End Electronics (Tracker, Calo, integration)
- Calorimeter's High Voltage
- Tracker's High Voltage
- Gas system factory
- Coil
- Light injection system
- Source calibration system
- DAQ
- Database storage
- Data management and Storage
- Environmental parameters
- Radon detectors
- Deradonized air factory
- Network
- Servers
- …

Différentes façons d'interfacer du matériel pour réaliser l'instrument



Projet CTA : exemple d'architecture du SC des caméras

Un système de control & command assure la sécurité des différentes parties de la camera.

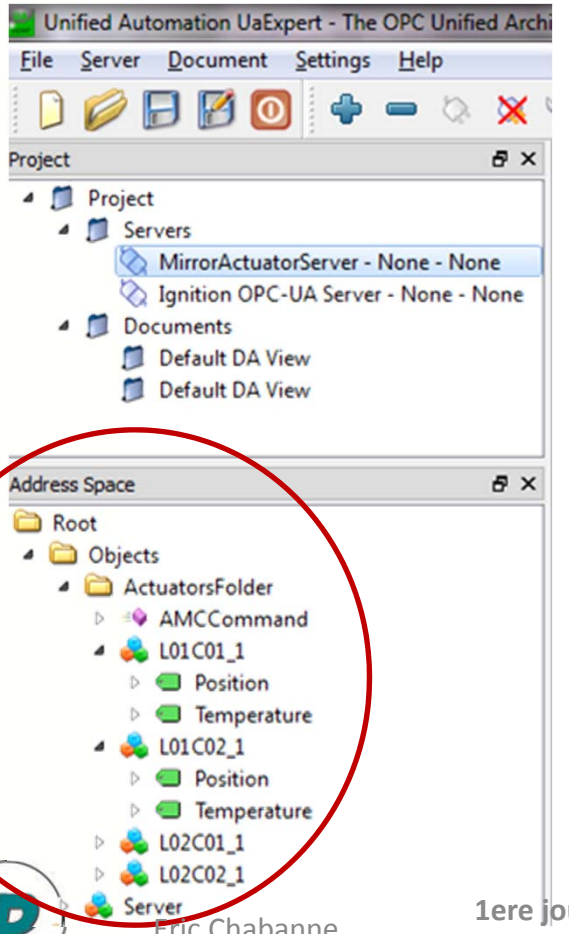


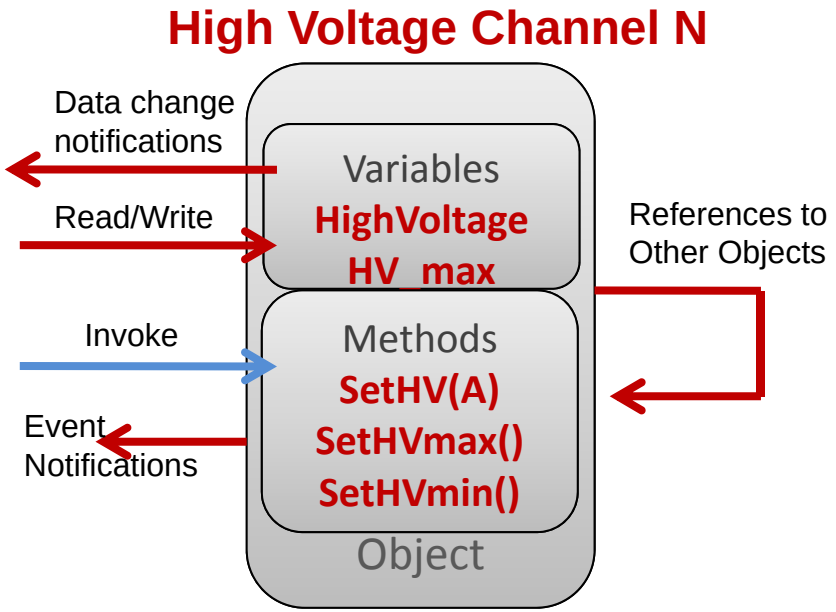
OPCUA kesako?

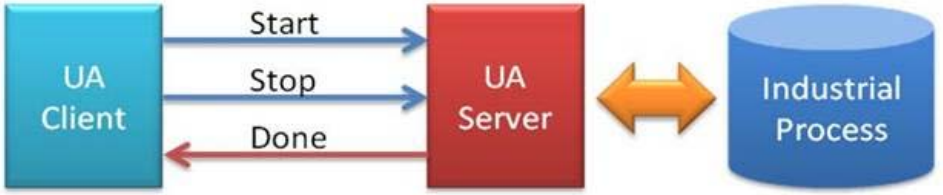
- **Un standard** défini par des fabriquants et vendeurs de matériels/logiciels de control industriel
- L' OPC foundation a récemment (mars 2015) ouvert les spécifications du standard.
- **Un framework** (du soft!) **cohérent, sécurisé et fiable orienté objet et multi-plateformes** permettant **d'accéder aux données et événements en temps réel et de façon historisée**, à différents niveaux (aggrégation)
  - Des capteurs/actuateurs aux application de control centralisé,
  - Des automates programmables industriels aux cartes éléctroniques embarquées.
- **Indépendent des platformes**
  - Différents OS supportés pour les parties clientes et serveurs d'OPCUA => interopérabilité
  - Implémentation multi-languages (C, C++ , JAVA)
- **Testé sur des plateformes embarquées**
  - java et C++ sur des CPU ARM et X86
- **Canal de communication fiable et simplifié**
  - Les clients & serveurs font directement partie du système de command & control
  - Support natif de l'encryptage sécurisé(RSA 128 and 512bits)
  - Possibilité de redondance des serveurs
- **Architecture Orientée Service (SOA)**
  - Services génériques pour parcourir et interroger le NameSpace du serveur, données en lecture/écriture, et publication/souscription aux événnements et changements sur les données.
  - **Le NameSpace OPCUA est un moyen standard pour les serveurs et clients d'accèder aux objets de ceux-ci.**

OPCUA pour faire court

- Framework Orienté Object
  - Chaque nœud OPCUA est un objet qui peut exposer ses variables et méthodes aux autres nœuds
- Tous les nœuds sont organisés à l'intérieur d'un 'Address Space'
  - Moyen standard de servir les objets entre serveurs et clients



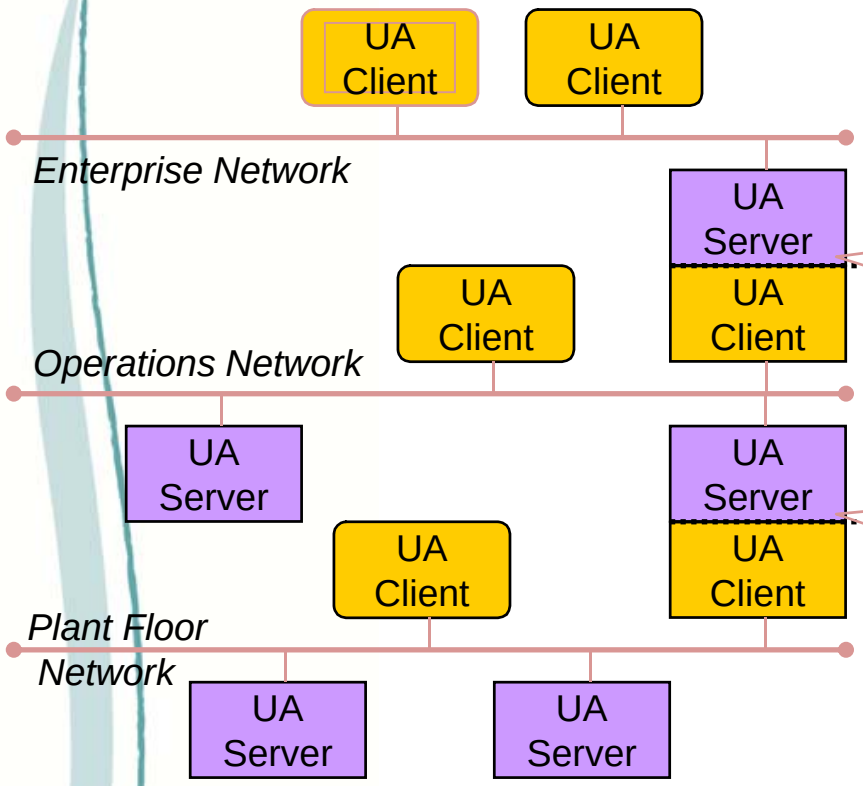




Protocoles OPC UA

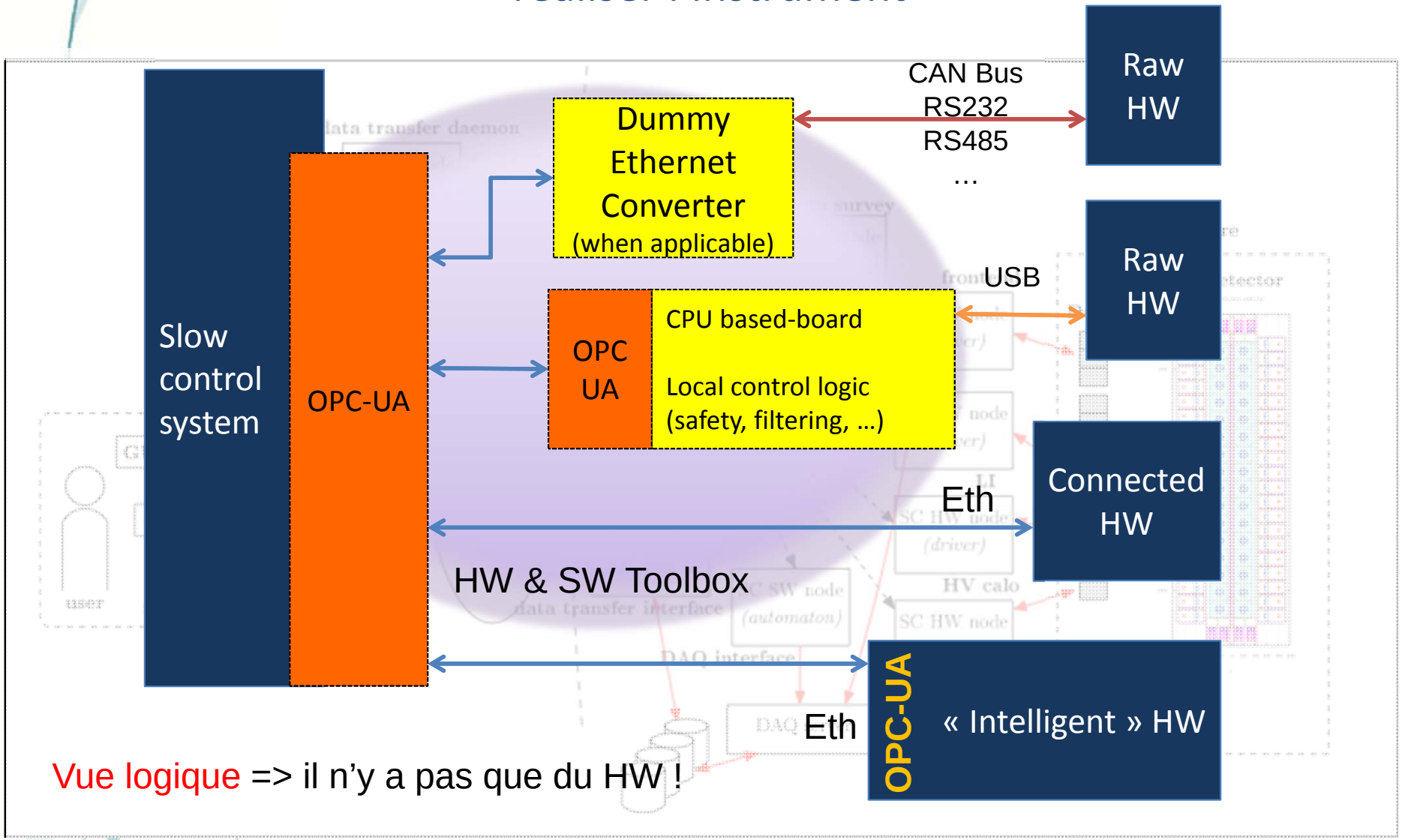
- 2 protocoles sont supportés :
  - Le protocole « binaire » : opc.tcp://Server
    - Offre la meilleure performance
    - Utilise le minimum de ressources (Pas de XML,HTTP et SOAP)
    - Offre une meilleure interopérabilité
    - Utilise un seul port TCP pour la communication (facilitant le tunneling mais nécessitant d'ouvrir spécifiquement un port dans les pare-feu).
  - Le protocole HTTP http://Server pour les Web Service.
    - SOAP pour l'interopérabilité avec les applications type « ERP » existants.
    - Passe à travers les ports standards des pare-feu (Port 80, 8080, 443)

Serveur OPCUA : l'aggrégation

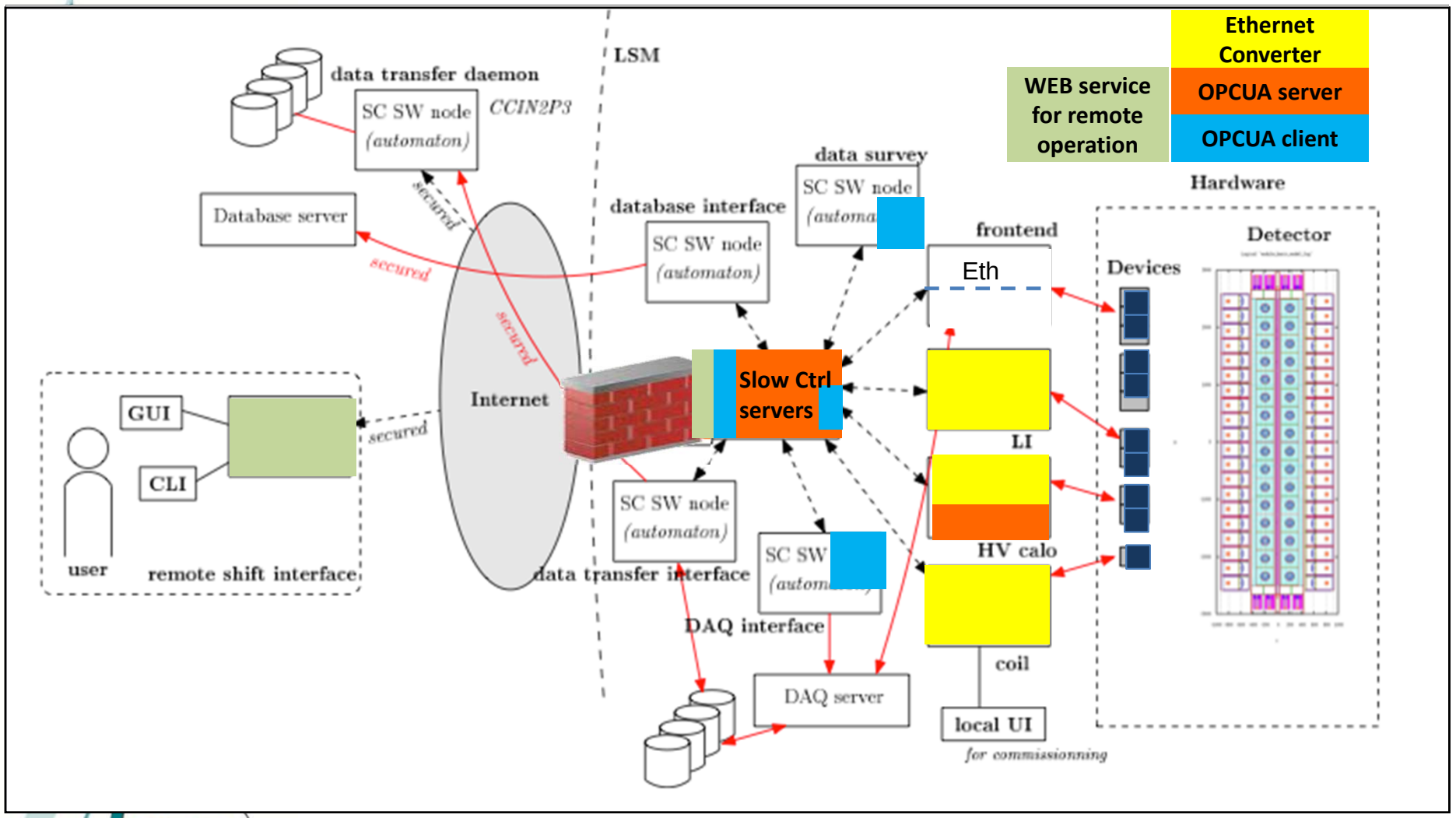


- L' agrégation de serveurs OPCUA permet l'extraction et le traitement de données depuis les couches basses(matériel) vers les données « Entreprise »
- Les données sont « typées » en utilisant un modèle d'information différent approprié pour les clients du niveau supérieur

Différentes façons d'interfacer du matériel via OPCUA pour réaliser l'instrument



Implémentation possible d'OPCUA pour l'expérience SuperNEMO



Un serveur SCADA générique…

… pour se simplifier la vie d'instrumentaliste

- OPCUA : mise en place.
  - OPCUA complet mais complexe.
  - On ne veut pas s'intéresser au contenant mais au contenu.
- Serveur OPCUA générique
- Une seule application commune a tout type de device.
- **Un** fichier de description XML avec une syntaxe (dictionnaire XSD) qui permet de construire **des** serveurs « spécifiques » adaptés à la configuration HW

Nos constats

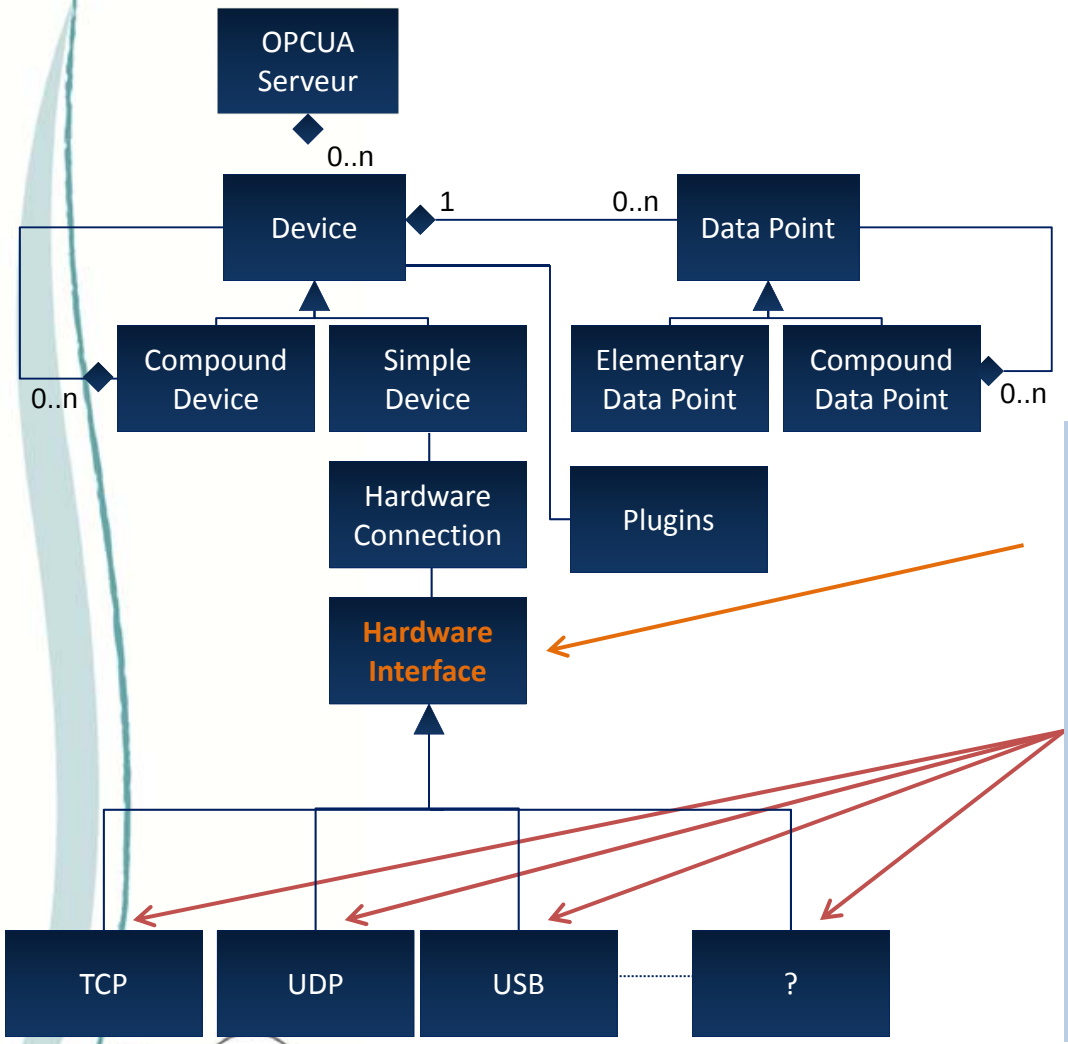
- Pour un «non-expert» :
  - L'écriture d'un serveur OPCUA peut être fastidieuse.
  - Le «développeur» n'est pas nécessairement l'intégrateur du matériel
- Il faut rendre l'intégration de matériel plus simple :
  - En cachant le plus possible la "machinerie" OPCUA
  - En limitant le code à écrire
  - En fournissant des moyens de décrire le matériel à intégrer
    - Les points de contrôle(set, get, …)
    - Le type de connexion du matériel(USB , RS232, TCPIP, …)
    - Le type de transport (UDP, TCP, …)
    - Le type de transfert (PUSH, PULL)
    - Les conditions de démarrage et d'arrêt du matériel :
      - Aspects de sécurités du matériel s , de l'instrument, des opérateurs.
    - Les dépendances et hiérarchies entre différents matériels à intégrer
- La description peut être partagée par le serveur et les clients

OPCUA : La description du matériel

- Basée sur XML et XSD
  - Le fichier « xsd » servant de dictionnaire et permettant la validation des fichiers XML.

- Les fichiers XML sont utilisés comme « entrée » du serveur OPCUA pour:
  - Décrire la connexion au matériel
  - Alimenter les « points de données » dans le serveur OPCUA
  - Décrire le flot de données issu du matériel
    - Afin d'analyser le contenu du flot
    - Afin de localiser dans le flot le nom et les valeurs des points de données.

OPCUA Server : Vers un modèle d'implémentation



- **La partie "abstraite"(Hardware Interface) sera constituée d'un jeu de méthodes « standards »**
  - **connect(…), read(…), write(…), close(…)**
  - **Le flux d' entrée/sortie sera également décrit pour permettre son analyse.**
- **L'intégration d'un matériel sera réalisée par héritage de cette partie abstraite.**
  - **Les méthodes connect(…), read(…), write(…), close(…),… seront implémentées a ce niveau**

Un exemple sur plateforme ARM RPi

Control & Command d'un dispositif occultant d'une caméra de télescope

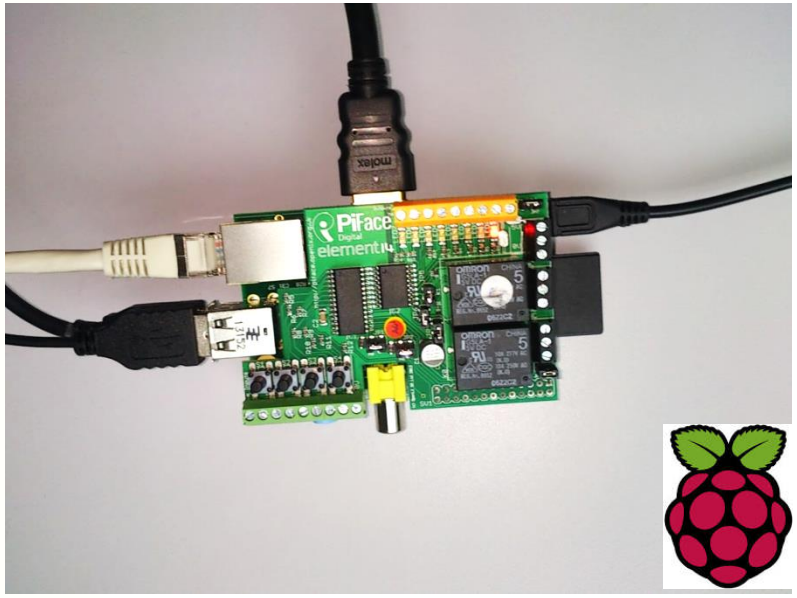
Client
Ua expert (commande du volet)

Client
Ua expert visualisation des alarmes)

Client
Ua expert (monitoring et historisation d'un node)

Client
Labview (IHM monitoring du volet

Client
CLI C++ (monitoring d'un node)



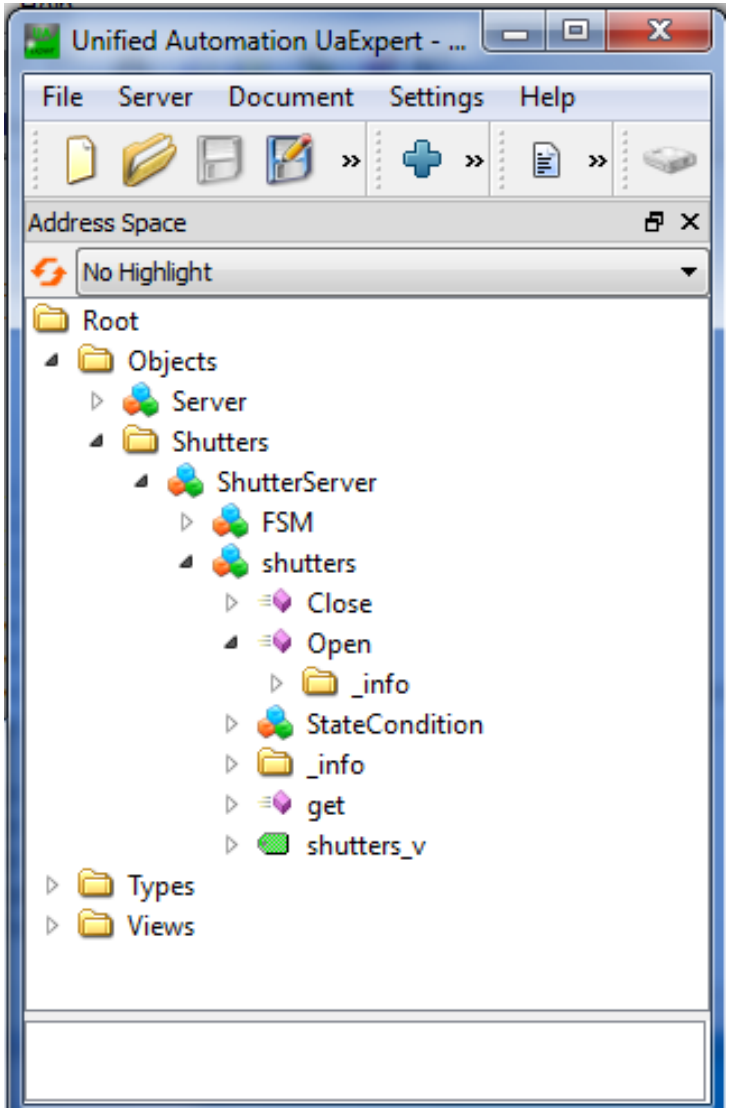
Serveur OPCUA générique

Raspberry PI

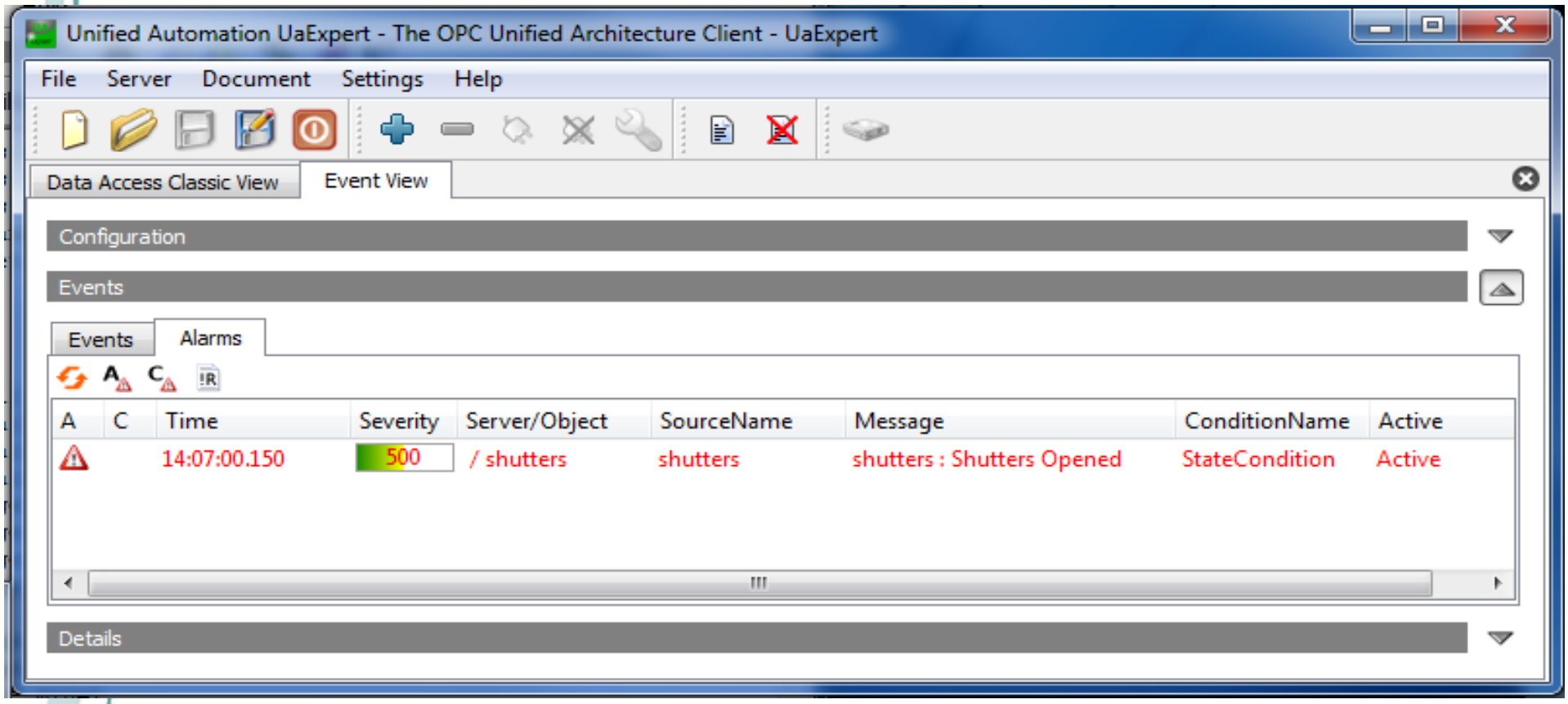
PIFace

Capteurs / Ationneurs

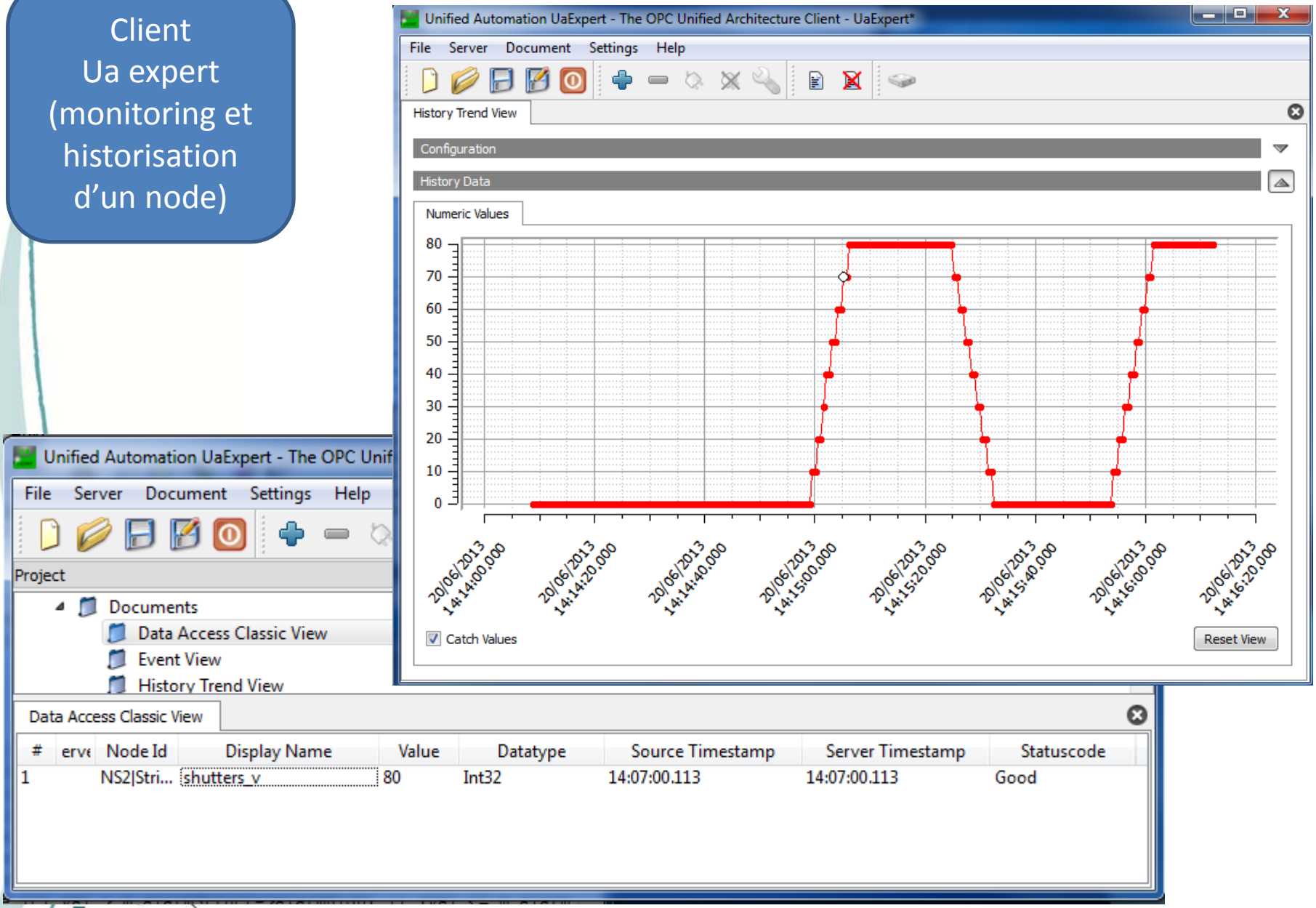
Client
Ua expert
(commande
du volet)



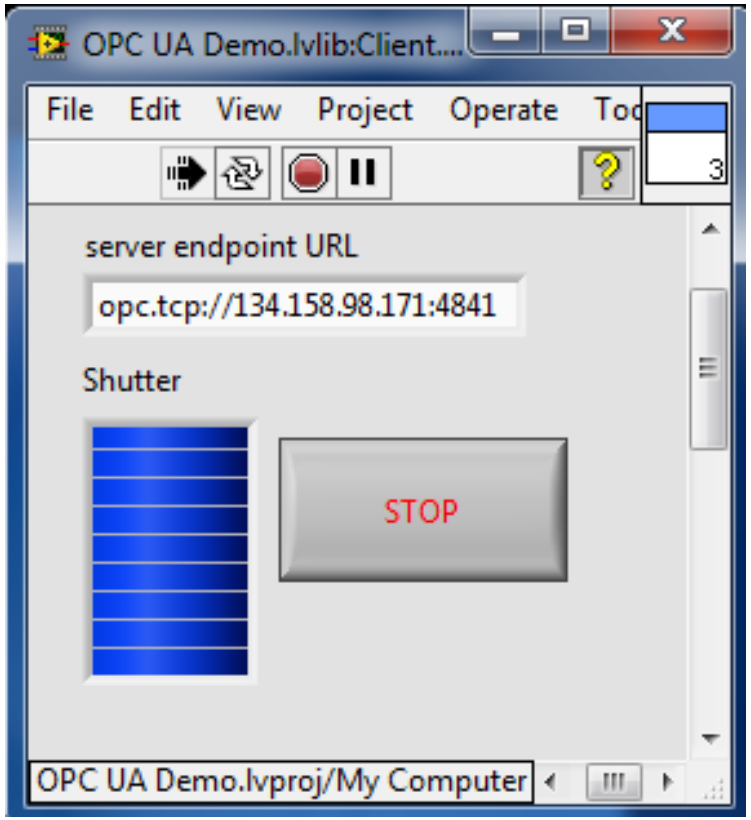
Client
Ua expert
visualisation
des
alarmes)



Client
Ua expert
(monitoring et historisation d'un node)



Client Labview (IHM monitoring du volet



OPCUA : quelques conclusions

- **Évolution naturelle de la connexion de matériel aux systèmes de supervision ou autres applicatifs.**
- **Représentation /accès unifié aux points de données via un protocole standardisé (comme CORBA) proche du HW**
- **Compatible avec les frameworks «grands instruments » Tango, EPICS, ACS ou autres par développement de connecteurs / bridges**
- **L'approche « générique » doit pouvoir s'appliquer à des environnements variés et donc indépendante d'un contexte**
- **Pérennité industrielle et interopérabilité assurées pour les 20 prochaines années (au moins?)**
- **Limite(s) actuelle(s) de OPCUA :**
  - **licences et prix des SDK « commerciaux »**
  - **Quid d'une gestion de licence centralisée IN2P3 (comme Labview, Cadence, Altium…)**
  - **Peu d'alternatives Open Source sérieuses actuellement :**
    - **Cependant, quelques initiative à suivre : OpenOPCUA , FreeOPCUA,**
    - **Passerelle Eolane (basée sur OpenOPCUA)**

Vers un Control/Command “générique” pour nos instruments ?

- Promouvoir un interfaçage hardware orienté instrumentalistes, indépendant du contexte d’utilisation, du type de matériel.
- intégrable dans plusieurs environnements SCADA (ou non)
- Faciliter la vie de l'intégrateur dans différents domaines (mise en service, opérations, maintenance…)
- intelligence au plus près du device -> intégration du MOS dans un cRIO 9068 National Instruments (base SoC ARM Zynq 7000 cohabitation avec le runTime LW – communication ETH entre les 2)
- Data format (= interface) standardisé et uniformisé (ex: station météo, alimentation, …
- Éviter de changer la partie cliente lorsque le device change (via la couche d'abstraction HW/SW)

Thème de réflexion proposé par le réseau Control/Commande – IN3P3 . s’il y a des intéressés, nous partageons volontiers ;-)

Merci de votre attention

La Mailling list du réseau Control/command IN2P3 :

instrum-control-command-l@in2p3.fr

Le wiki du réseau :

forge.in2p3.fr/projects/slowcontrol-4-instru-le-reseau-controle-commande-in2p3/wiki

chabanne@lapp.in2p3.fr