Présentation du framework Flair

Guillaume Sanahuja
Ingénieur de recherche UTC
gsanahuj@hds.utc.fr

Sommaire

- Historique
- Principe de fonctionnement
- Station sol
- Simulateur
- Liaisons entre composants
- Environnement logiciel
- Démo
- Conclusions/perspectives

Historique

Historique

- **Jusque 2010:**
  - **Autant de solutions matérielles et logicielles que de plateformes**
  - **Pas de mutualisation**
  - **Pas de capitalisation (perte des démos)**

- **Septembre 2010:**
  - **Les drones deviennent une plateforme soutenue par le service plateformes du laboratoire**
    - **focalisation sur les structures types x4 et x8**

- **Septembre 2010 - septembre 2011:**
  - **Conception et réalisation d'une électronique de base (autopilote)**
  - **Support de Xenomai**
  - **Ecriture de drivers RTDM pour les périphériques utiles**
  - **Programme autopilote d'exemple monolithique, modifié au cas par cas**

Historique

- **Septembre 2011:**
  - **Début d'écriture d'un framework pour faciliter le développement d'applications**
  - **Abandon progressif des solutions non supportées par le service**
- **Avril 2013:**
  - **Mise en production du framework drone**
    - **13000 lignes de code (SLOC, software line of code)**
    - **99 fichiers .cpp**
    - **1 seule plateforme supportée (par conception)**
  - **Abandon progressif des solutions hors framework**
- **Avril 2016:**
  - **Le framework drone passe en Open Source (Cecill-C) (sauf Flair Hds)**
  - **Flair: Framework Libre Air**
- **Aout 2018:**
  - **73000 lignes de code (+ 16000 dans Flair Hds)**
  - **260 fichiers ,cpp (+60 dans FlairHds)**
  - **4 plateformes supportées, évolutif**

Drones supportés

hds_x8

Ardrone 2

Bebop 2

Minidrone (à finaliser)

Principe

- **Point de départ:**
  - **faciliter développement logiciels drones**
  - **avoir une station sol souple et générique**
- **Librairies C++**
  - **framework (cœur)**
  - **filtres**
  - **capteurs et actionneurs**
  - **simulateur**
- **Intégration temps réel (Xenomai), ou non**
- **Le programme distant construit la station sol (réglages, graphes, etc)**
- **Automatisation:**
  - **envoi des données aux graphes**
  - **enregistrement des données**
  - **mise à jour des filtres**

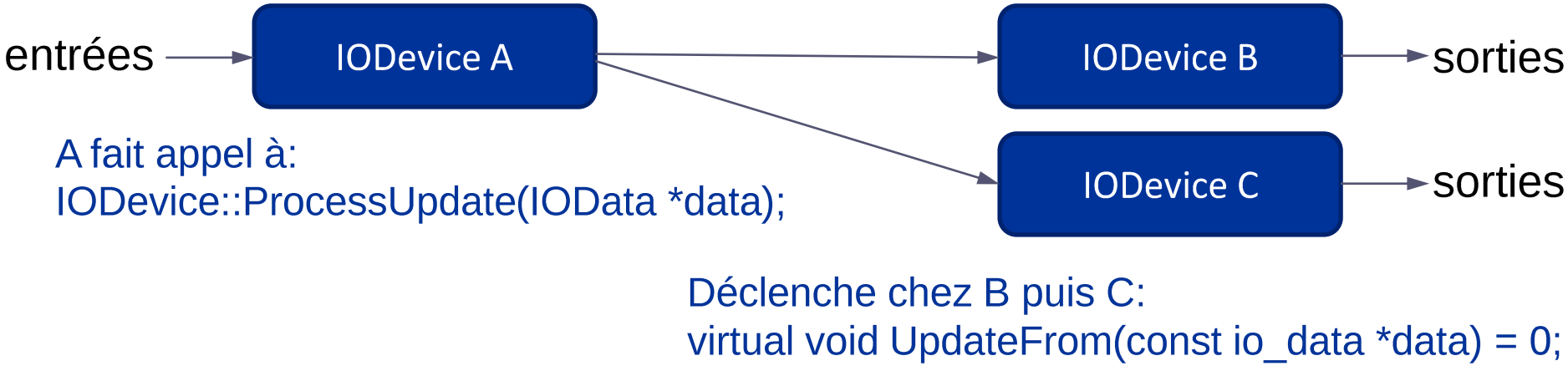
Principe: datas et IOs

- **Données:**
  - **Ex: matrices, images, coordonnées GPS, données inertielles, etc**
  - **Classes abstraites: *IOData*, *IODataElement***
  - **Contient données+timestamp**
  - ***IODataElement* permet d'accéder à un élément particulier des données**
  - **Un *IODataElement* peut être associé à un graphe de la station sol**
    - **transmission régulière**
    - **automatique et transparent**

- **Entrées-sorties:**
  - **Ex: capteur, actionneur, filtre, loi de commande**
  - **Classe abstraite: *IODevice***
  - **Entrées sorties de type *IOData***
  - ***IODevices* chainables à la construction!**
  - **Possibilité de définir des *IODatas* à sauvegarder (logs)**

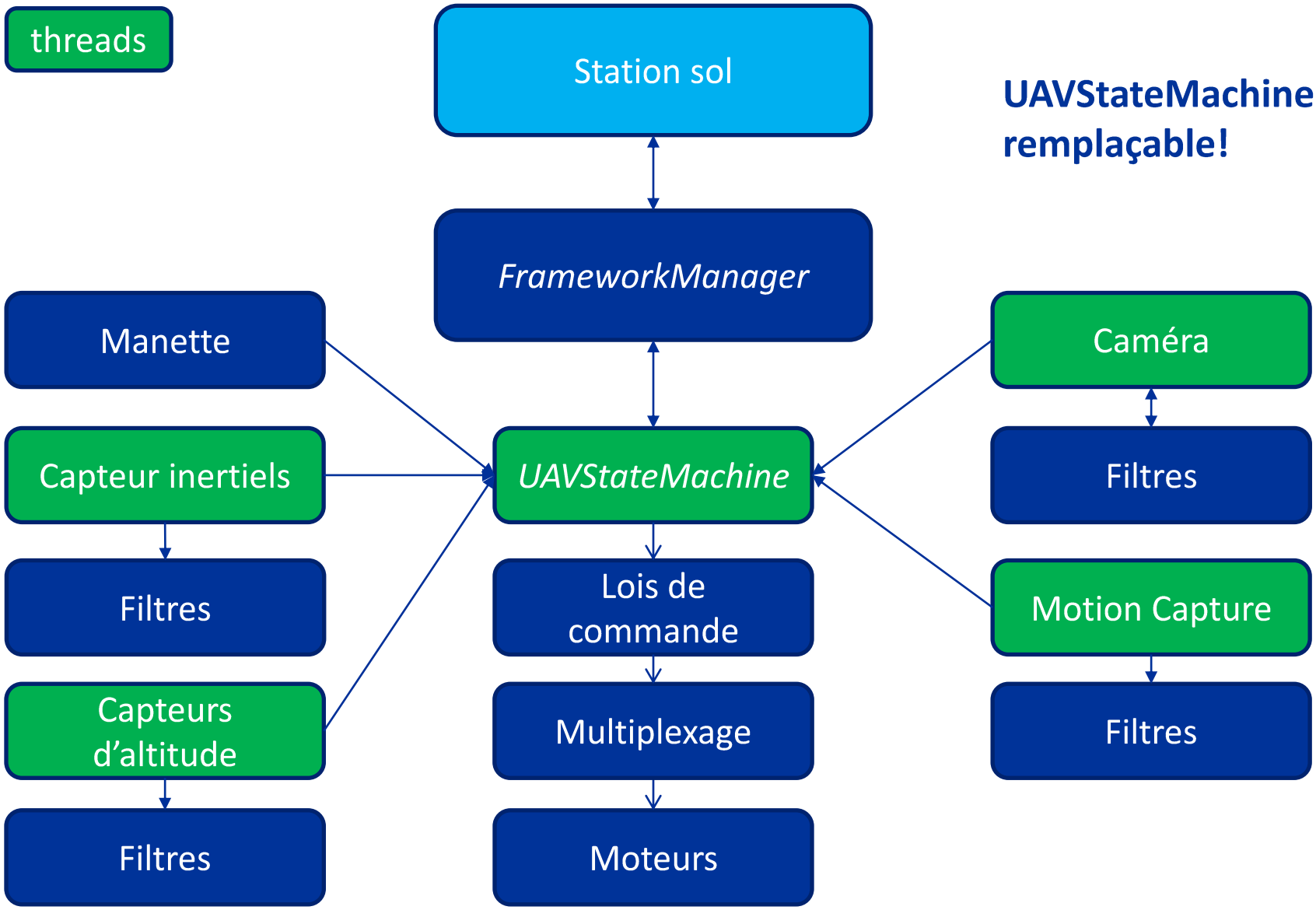
Principe: datas et IOs

- **Construction de l'*IODevice* A sans parent**
- **Les *IODevices* B et C passent A comme parent à la construction**
- **A,B et C définissent des *IODatas* à sauvegarder dans leurs constructeurs**

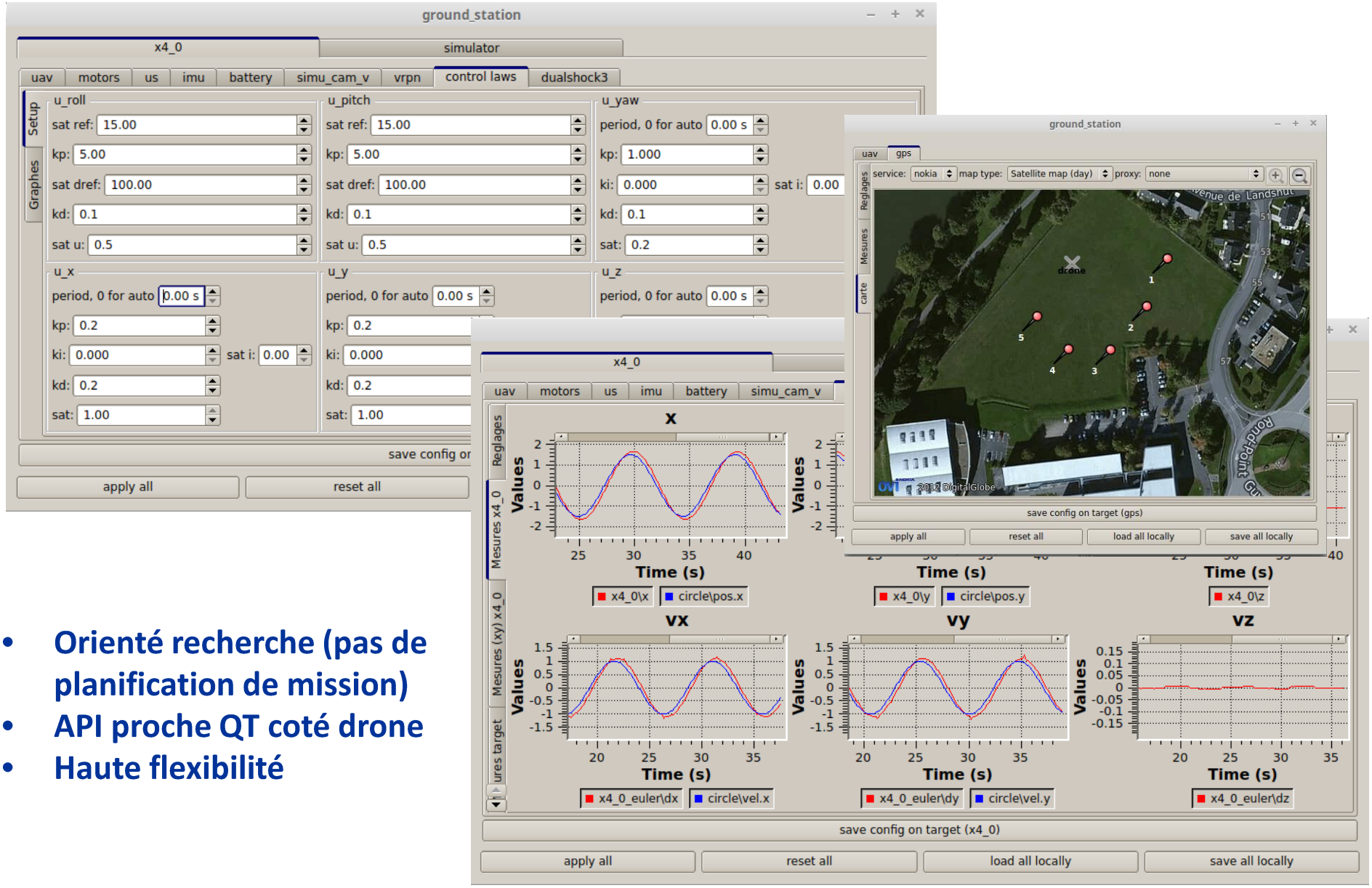


- ***ProcessUpdate* sauvegarde les différents logs**
- **Nécessité d'apeller *ProcessUpdate* dans *UpdateFrom***

Principe: UAV

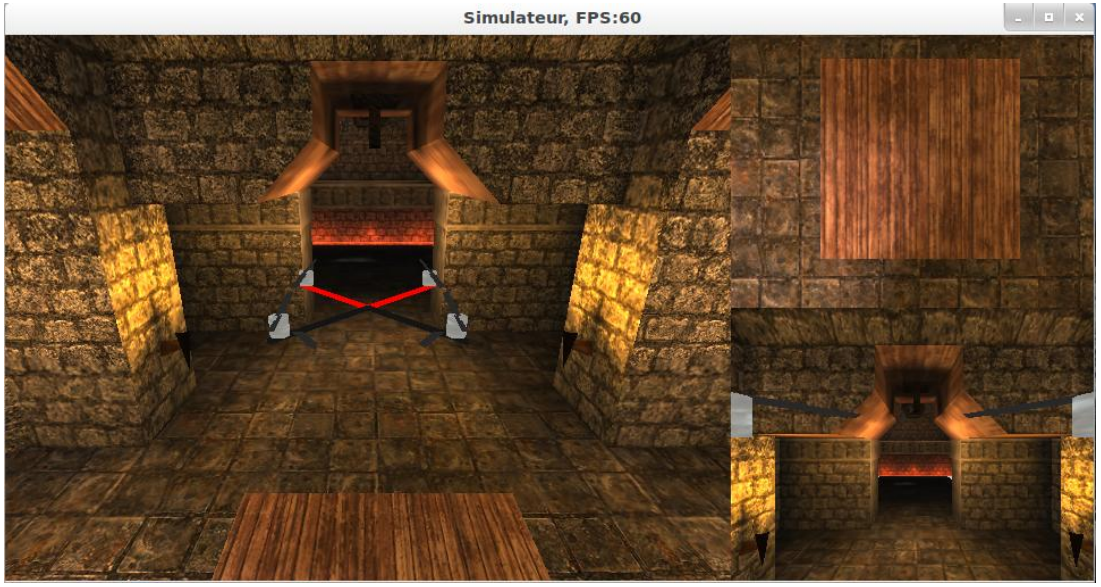


Flair GCS



- **Orienté recherche (pas de planification de mission)**
- **API proche QT coté drone**
- **Haute flexibilité**

Simulateur

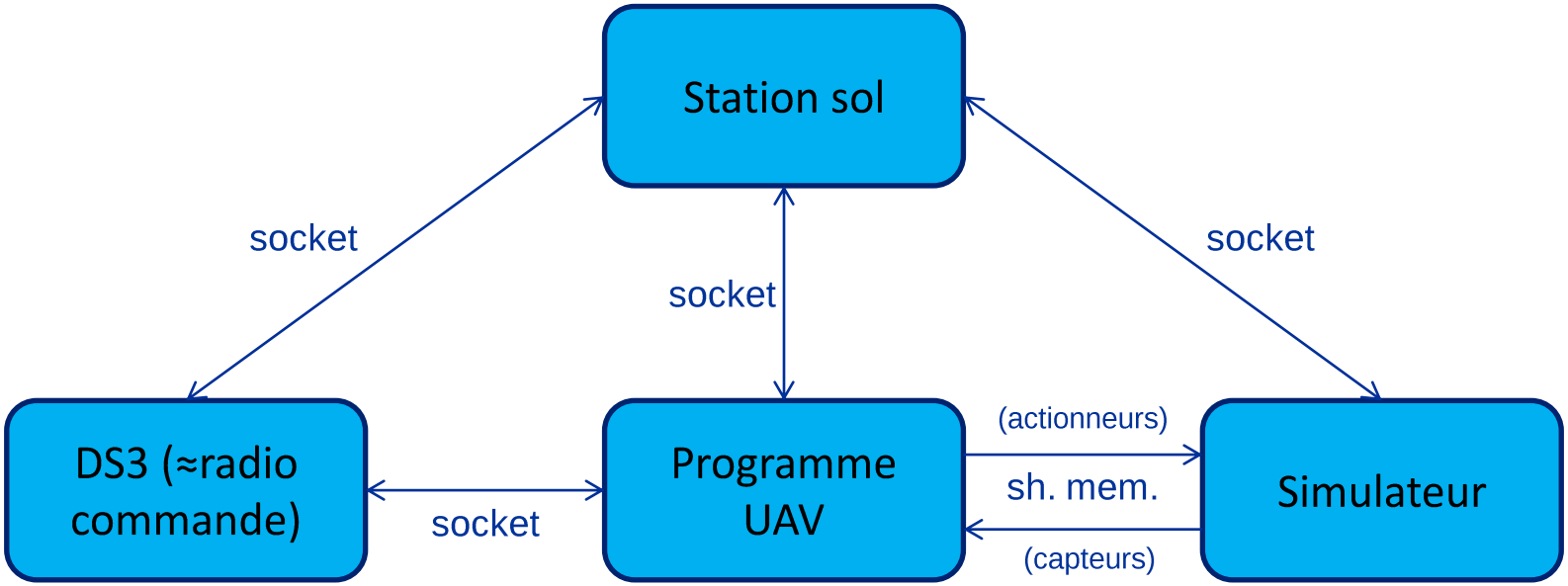


- **Exécution du code drone sur un PC**
  - **PC temps réel ou non**
  - **émulation capteurs**
  - **caméras embarquées**
- **Possibilité de simuler *n* drones**
- **Environnements 3D préexistants, ou personnalisés**
- **Permet de tester tous les programmes en sécurité**

gain de temps, mise au point de programmes plus complexes

Liaisons composants

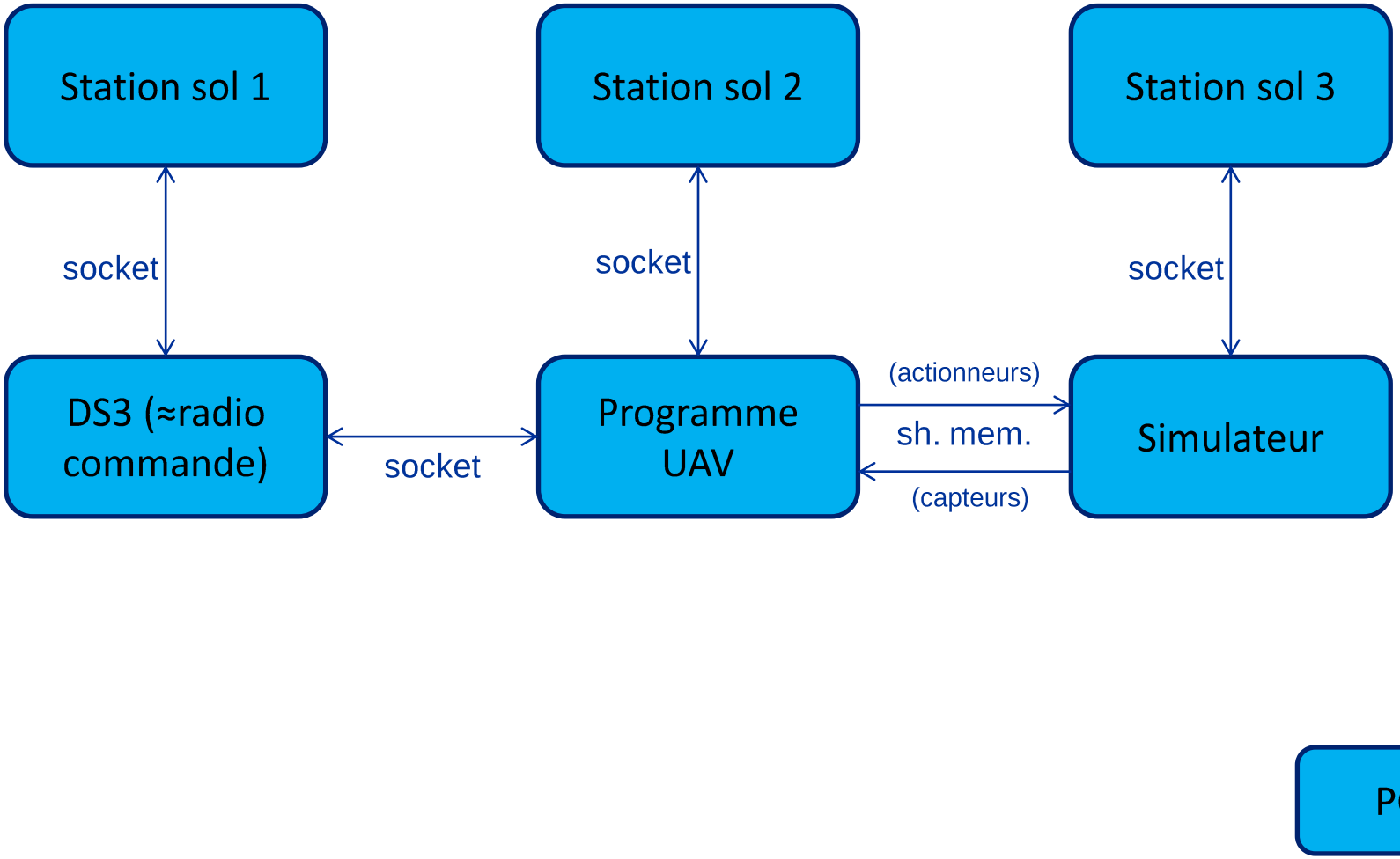
1 drone simulé sur 1 PC



PC

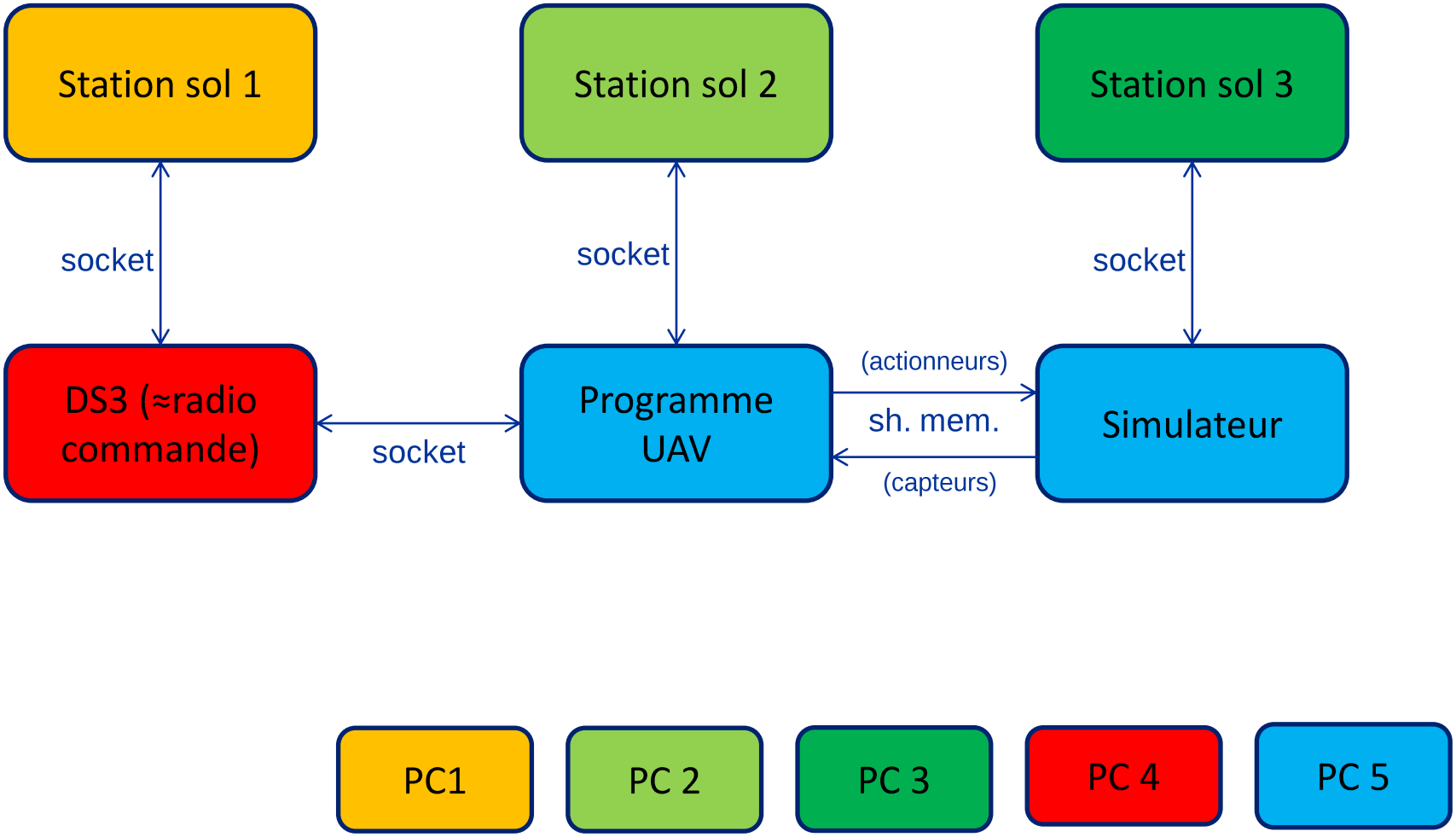
Liaisons composants

1 drone simulé sur 1 PC



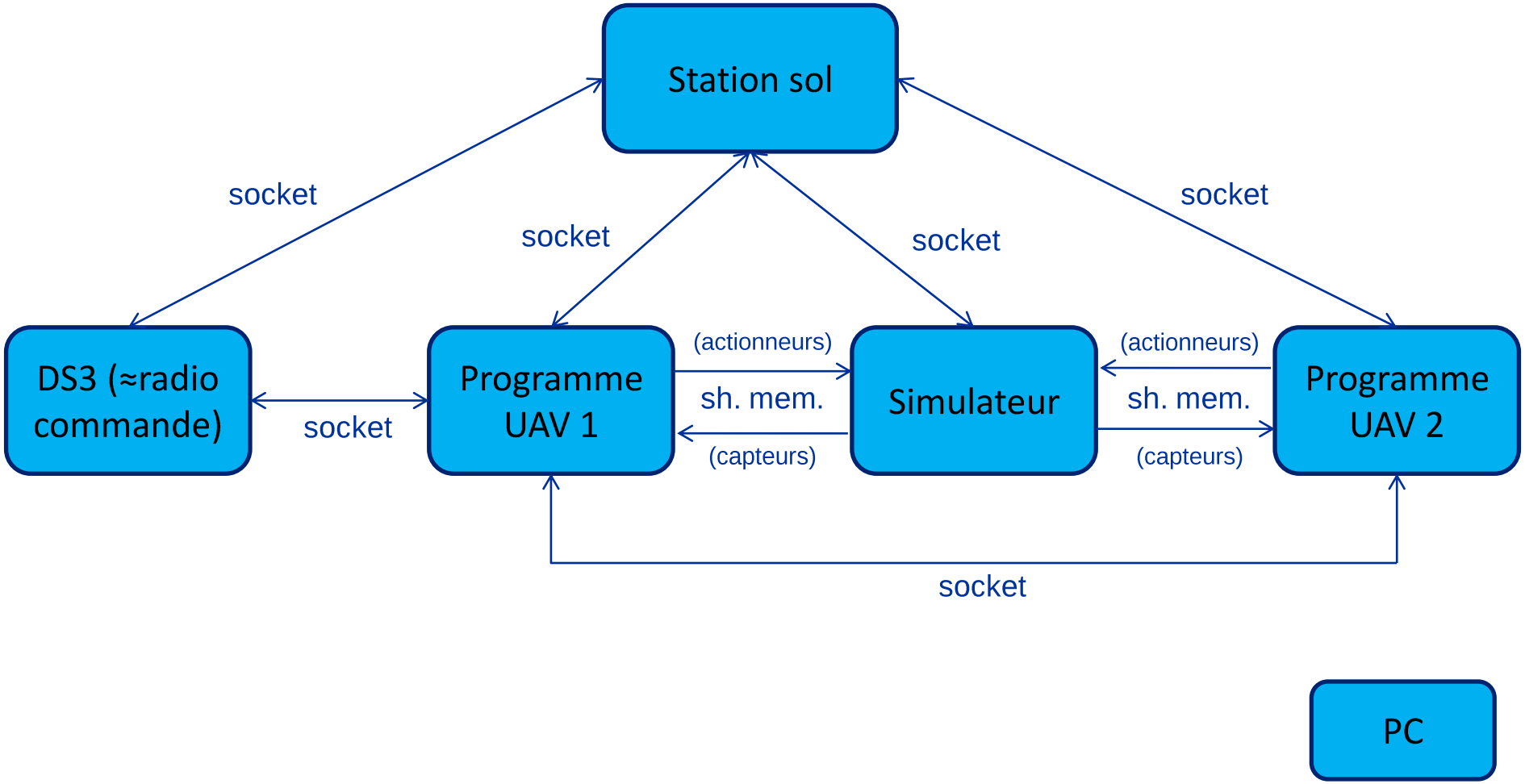
Liaisons composants

1 drone simulé sur 5 PCs



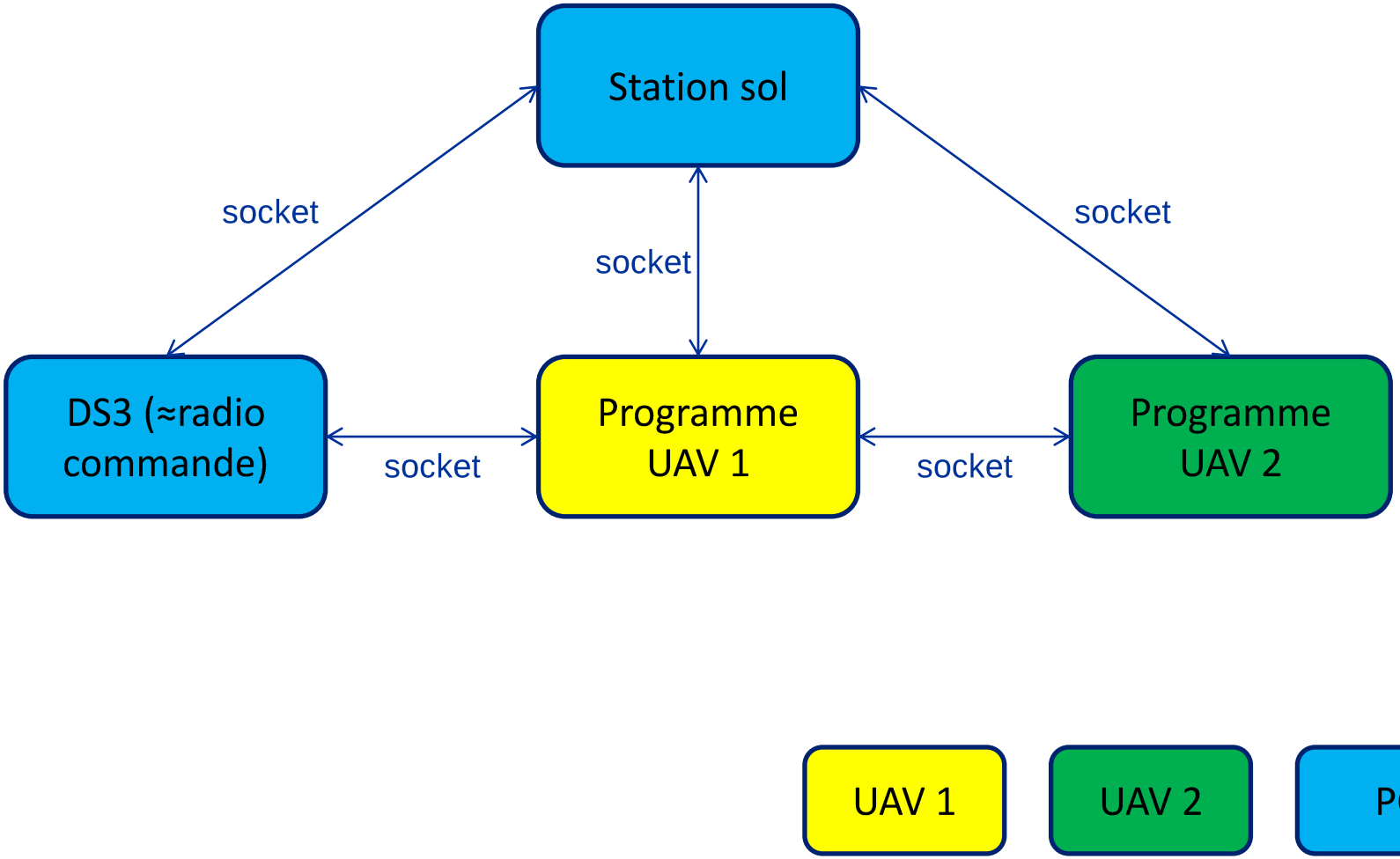
Liaisons composants

2 drones simulés sur 1 PC



Liaisons composants

2 drones réels avec 1 PC



Environnement logiciel

RobOMAP3-RT

yocto PROJECT

XENOMAI

IRRLICHT lightning fast realtime 3D engine >>>

Qt

CMake

Code::Blocks
The open source, cross-platform IDE
13.12

trac
Integrated SCM & Project Management

Doxygen

SUBVERSION

Démo

Conclusions

- **Flair utilisé dans de nombreuses thèses à HDS (voir démos) et par les visiteurs**
- **Orienté drone mais utilisable en robotique**
- **Nécessité d'ajouter du bruit aux capteurs**
- **Flair Hds à réécrire**

- **Manque de main d'œuvre/ communauté**
- **Impossible de concurrencer les autopilotes communautaires (Pixhawk, APM, Paparazzi)**

Perspectives

- **CPER: achat de nouveau drones (court terme)**
- **Évaluation des Intel Aero**
  - **configuration AP+ROS**
- **Volonté initiale de tout faire/maitriser**
  - **évolution vers des applis plus haut niveau**
- **Vers un abandon progressif de Flair**

Questions?

https://devel.hds.utc.fr/software/flair