Projet RobAIR2013

RICM5

Elizabeth PAZ
Mickael NICOLACCINI
Arthur ALEXANDRE
Salem HARRACHE

Sommaire

- Introduction
- Gestion des équipes
- Architecture
- Conclusion
- Démo

Introduction

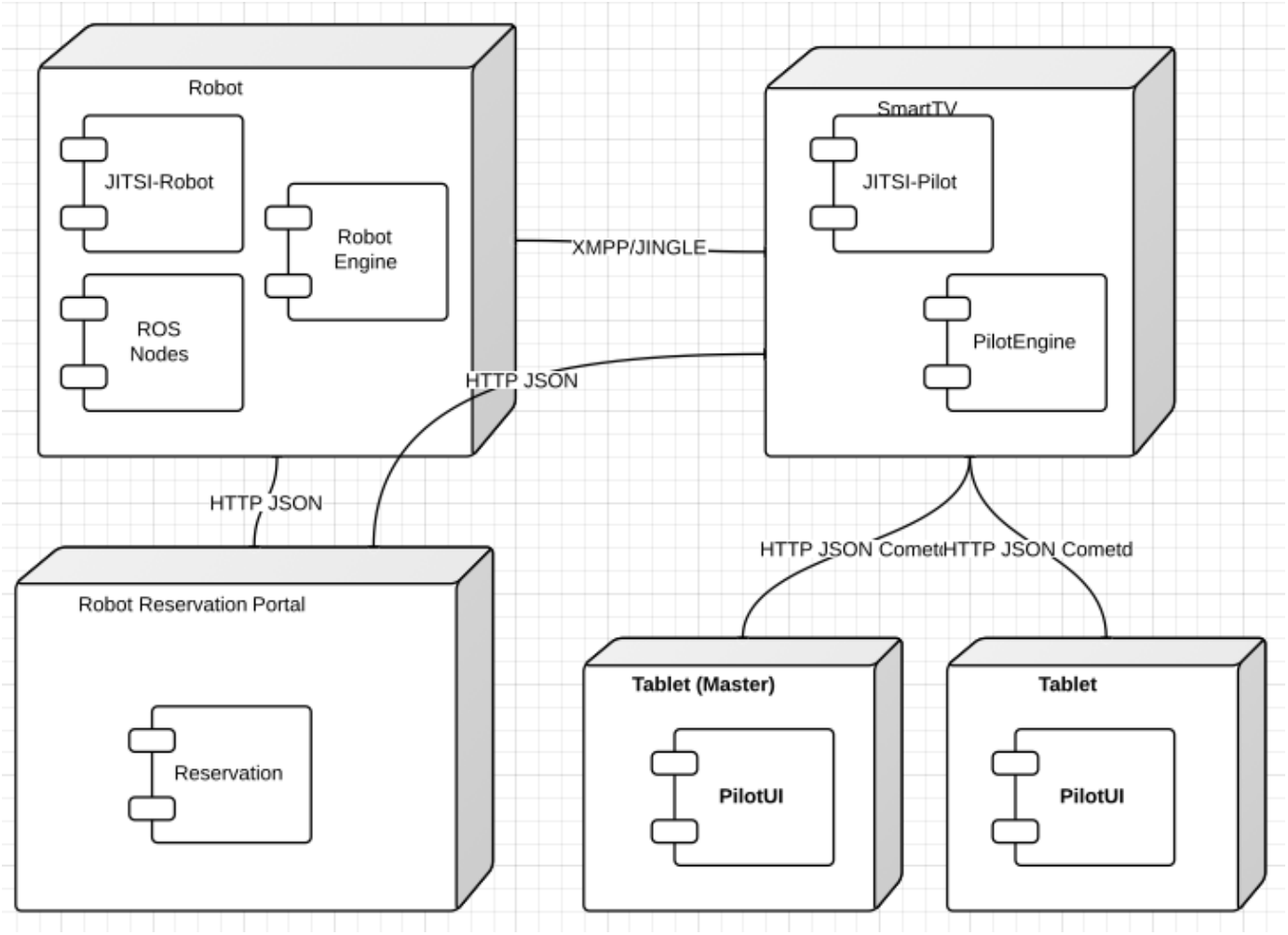
- Marché de la robotique de service en pleine expansion
- Potentiel fort avec des perspectives de croissance prometteuses
- RobAIR2013 pensé dans le contexte d'une visite au musée
- Projet intéressant avec outils open source

Gestion des équipes

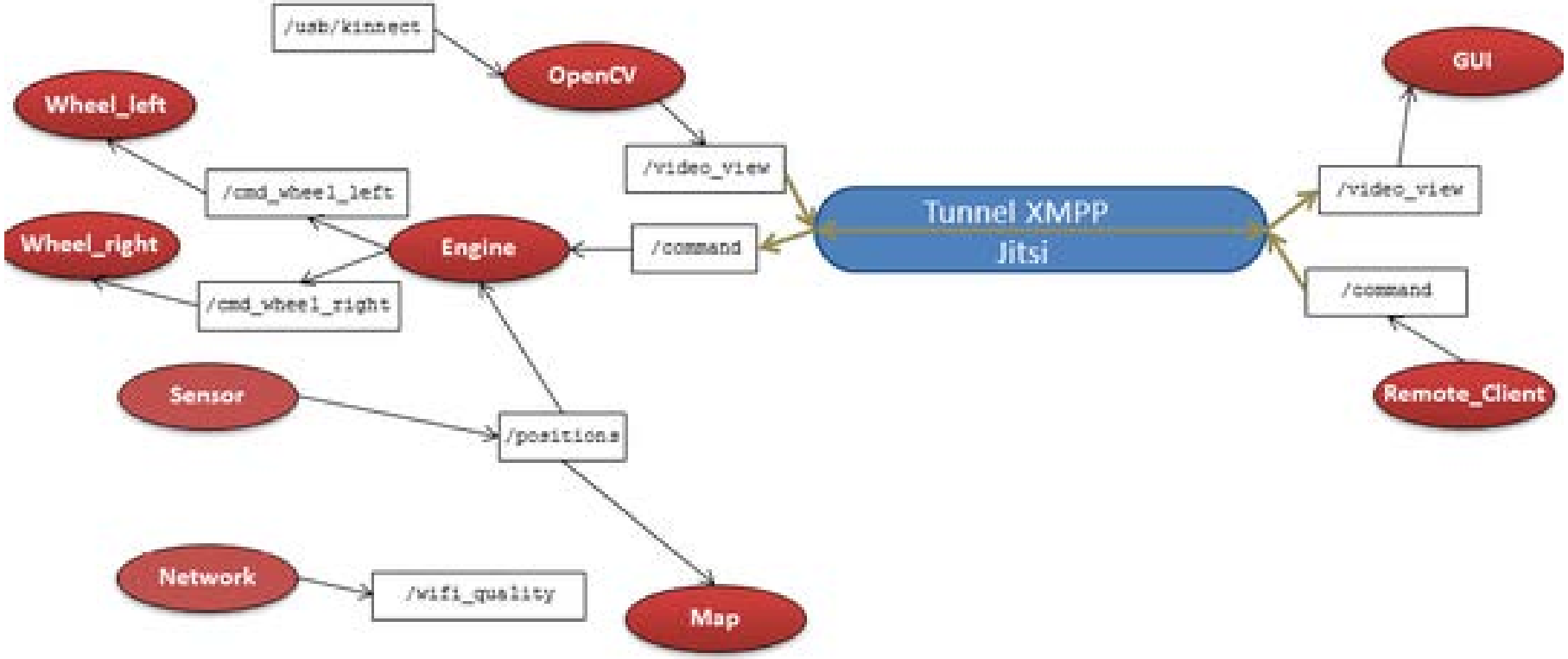
- 3i 4ème année
  - Conception et fabrication du robot
- RICM 4ème année
  - Roaming côté Robot
  - Architecture ROS côté Robot
  - Communication contrôleur/utilisateur et Robot
- Pôle Supérieur de Design de Villefontaine
  - Design et scénarios
- ENSIMAG
  - Pilotage automatique et semi-automatique
  - Géolocalisation et cartographie dynamique

- Réunion hebdomadaire avec les RICM4

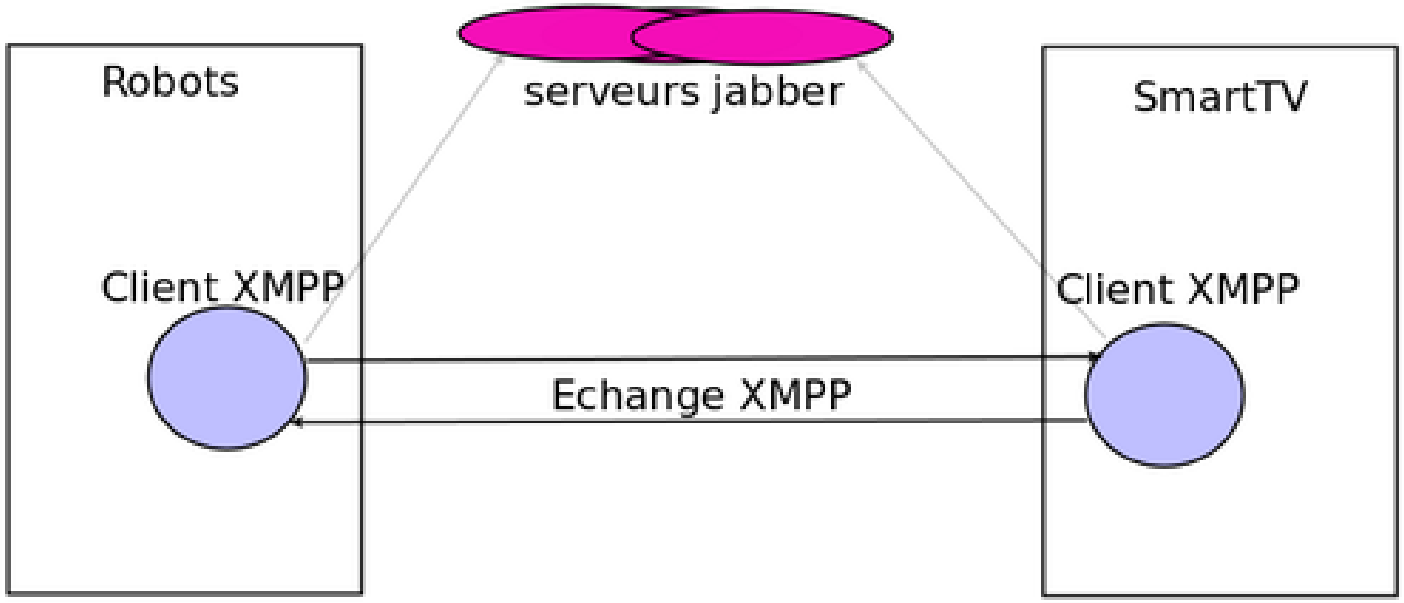
Architecture : point de départ



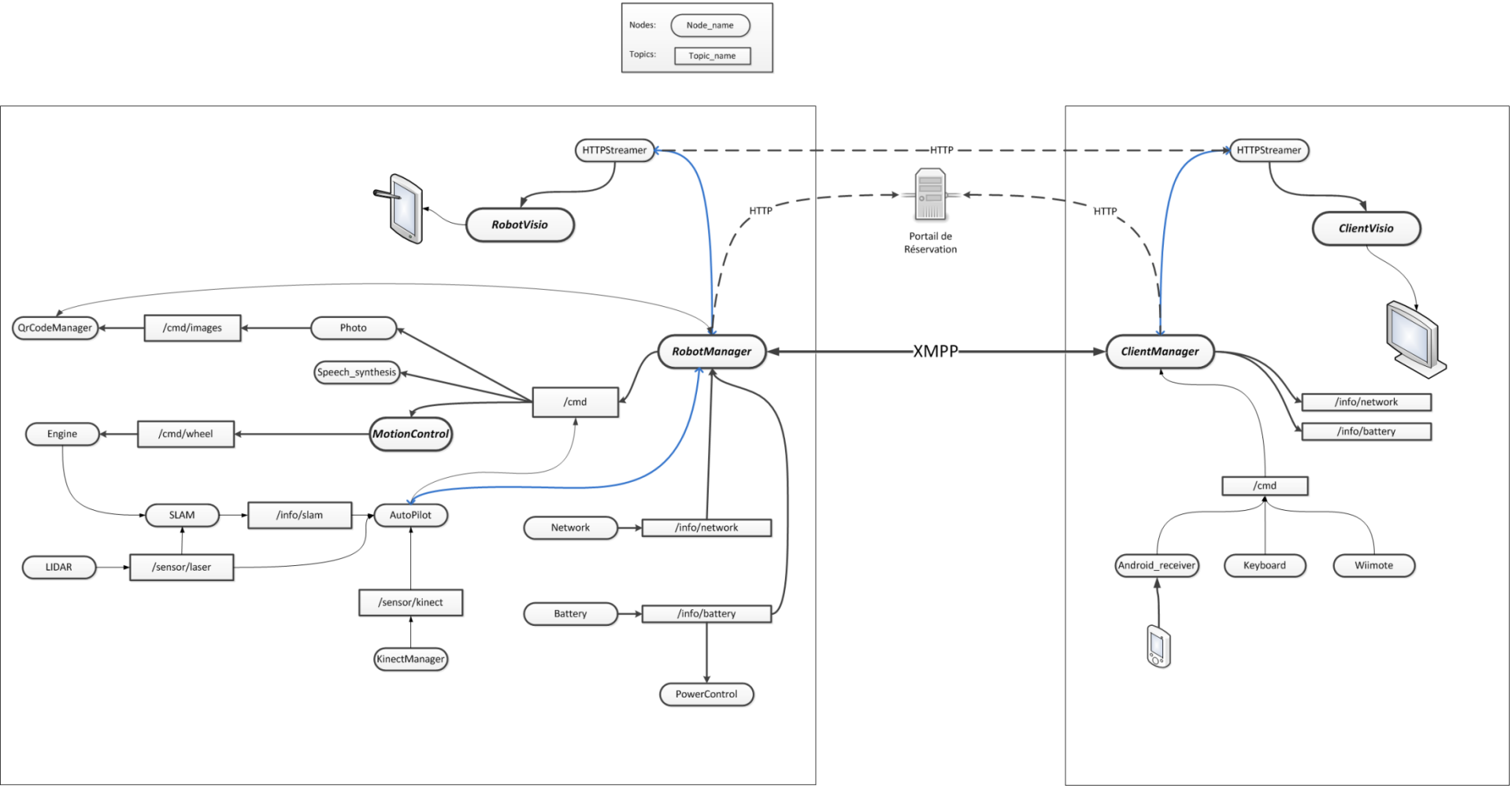
Architecture : Modèle ROS



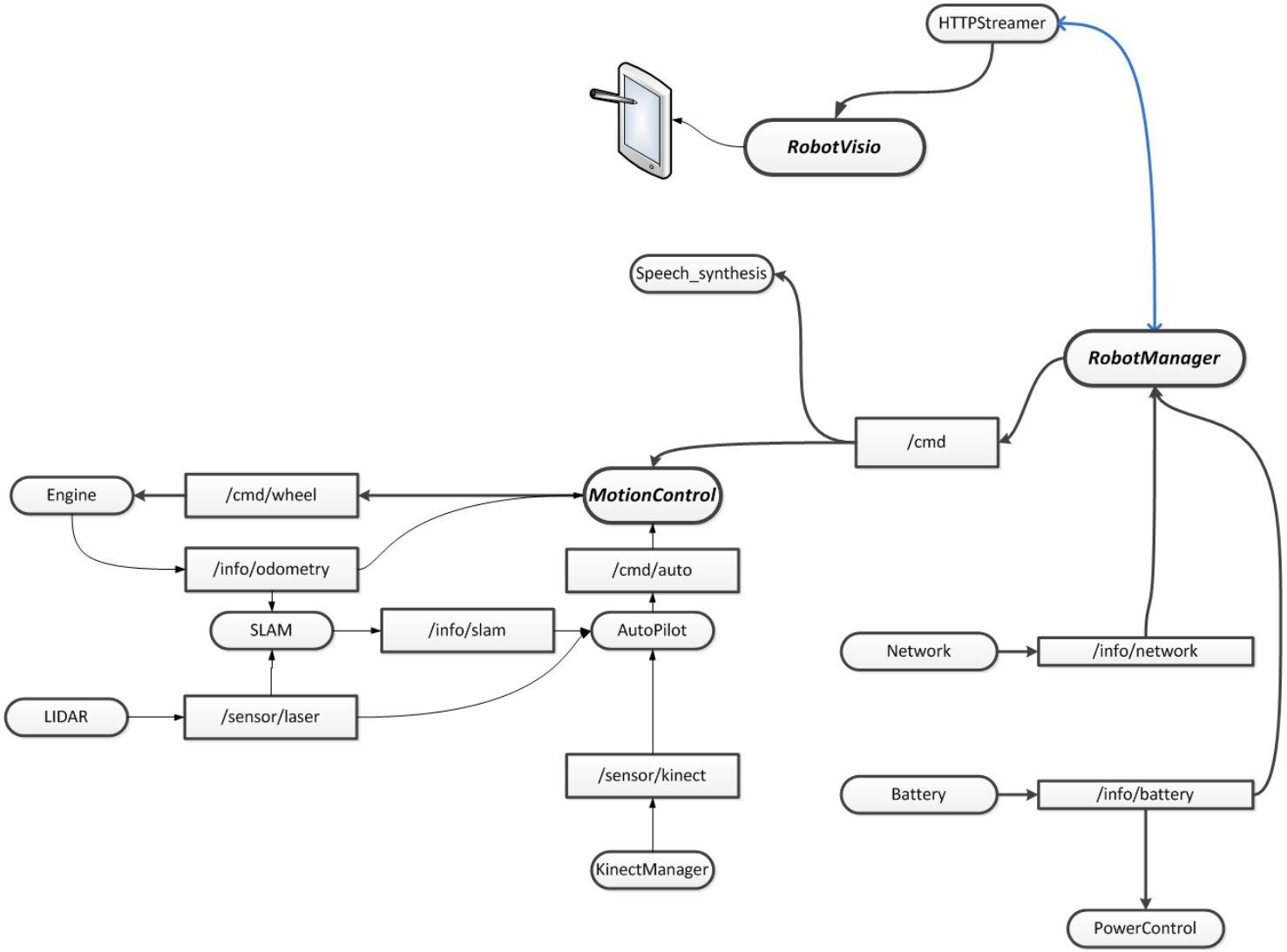
Architecture : XMPP+ vidéoconférence



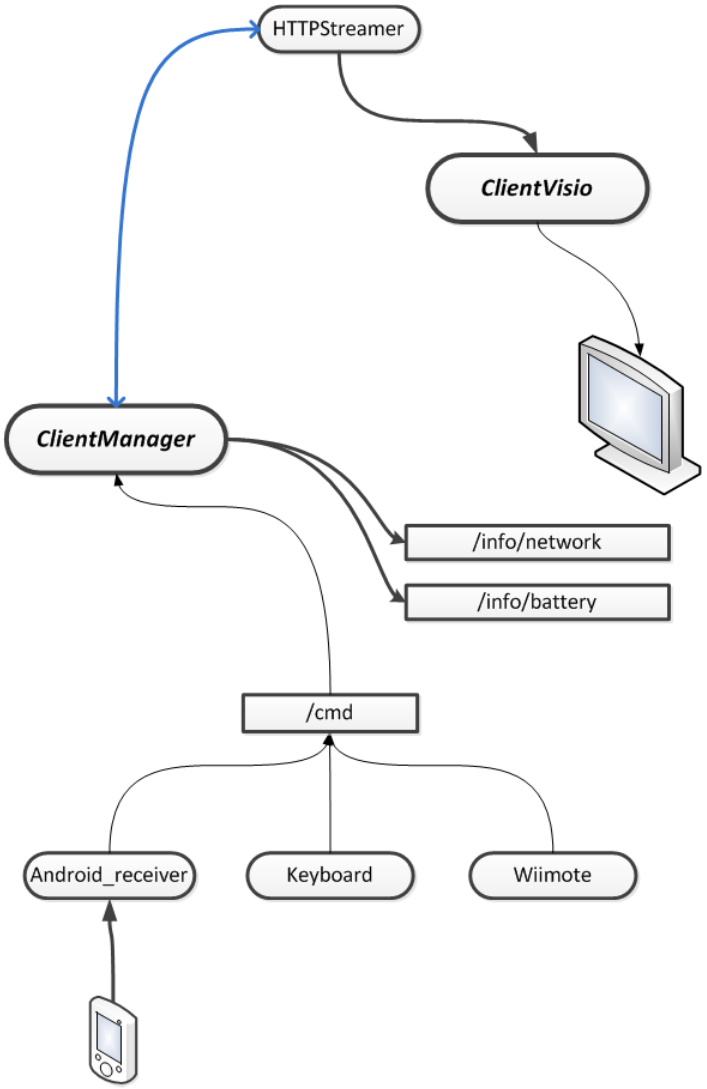
Architecture finale



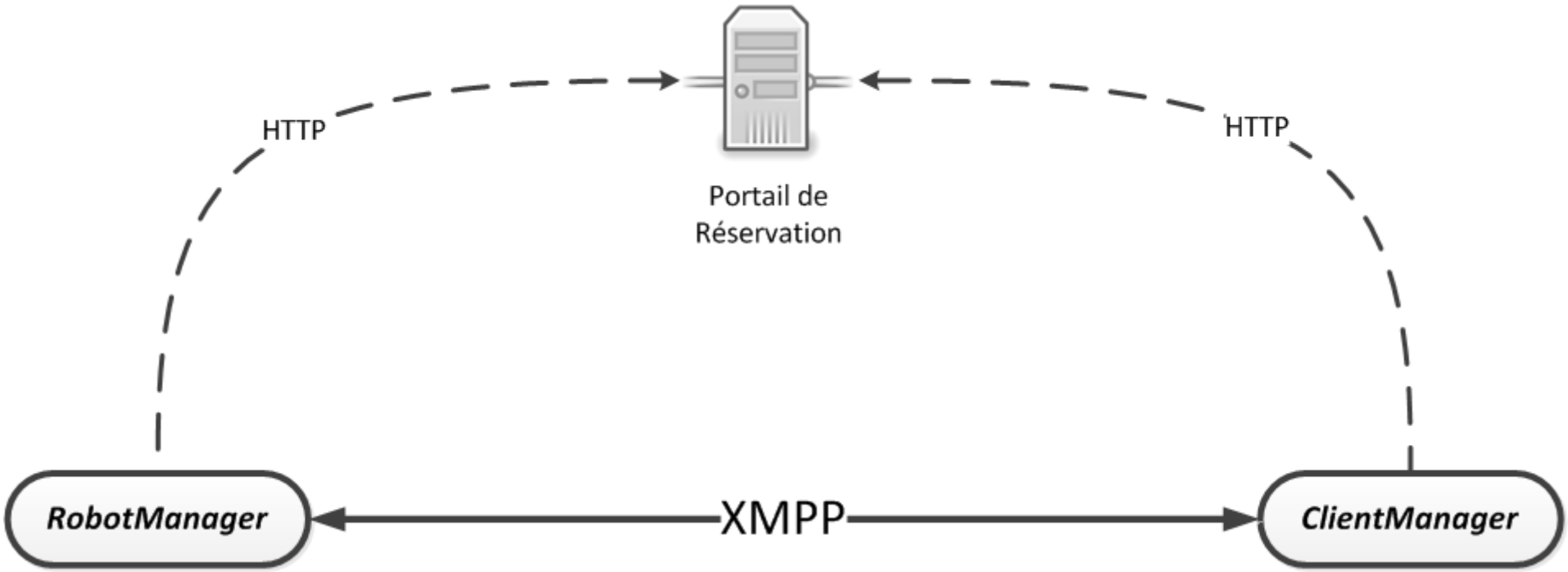
Architecture finale – côté Robot



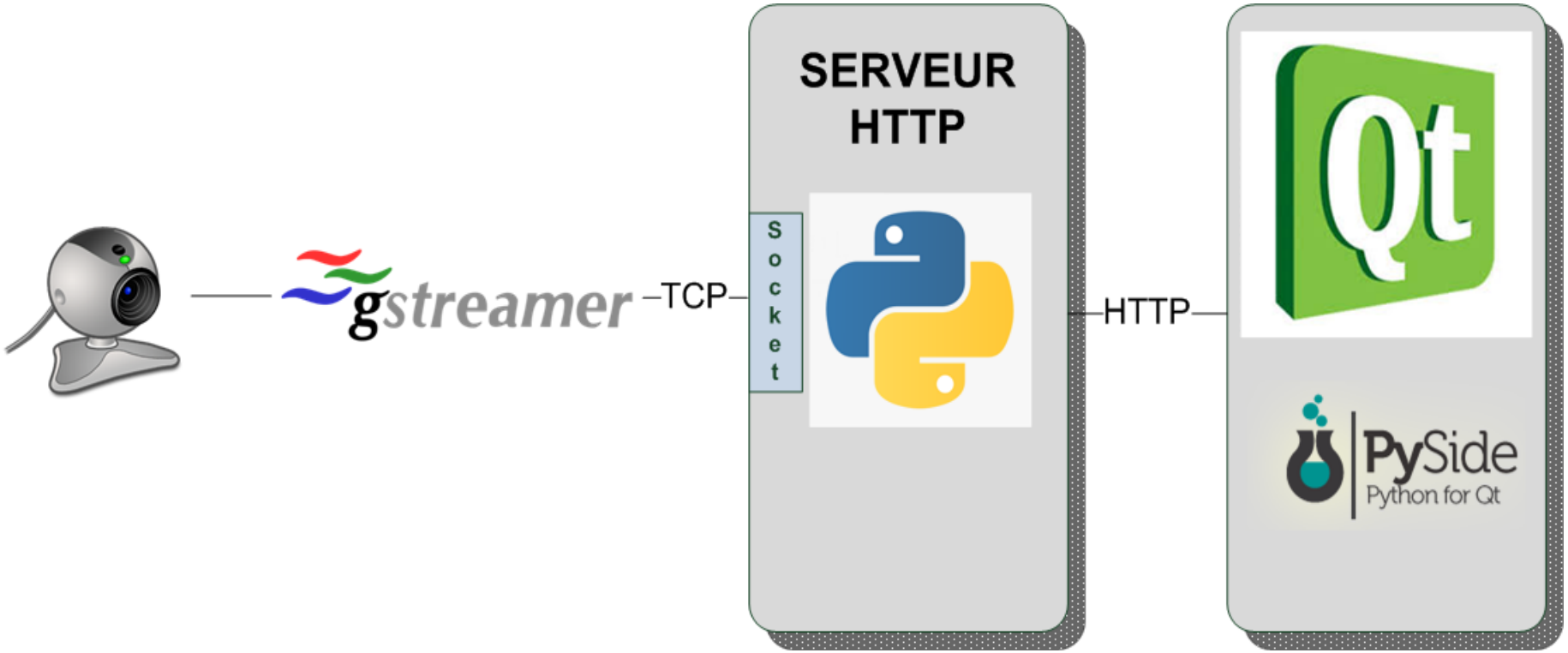
Architecture finale – côté Client



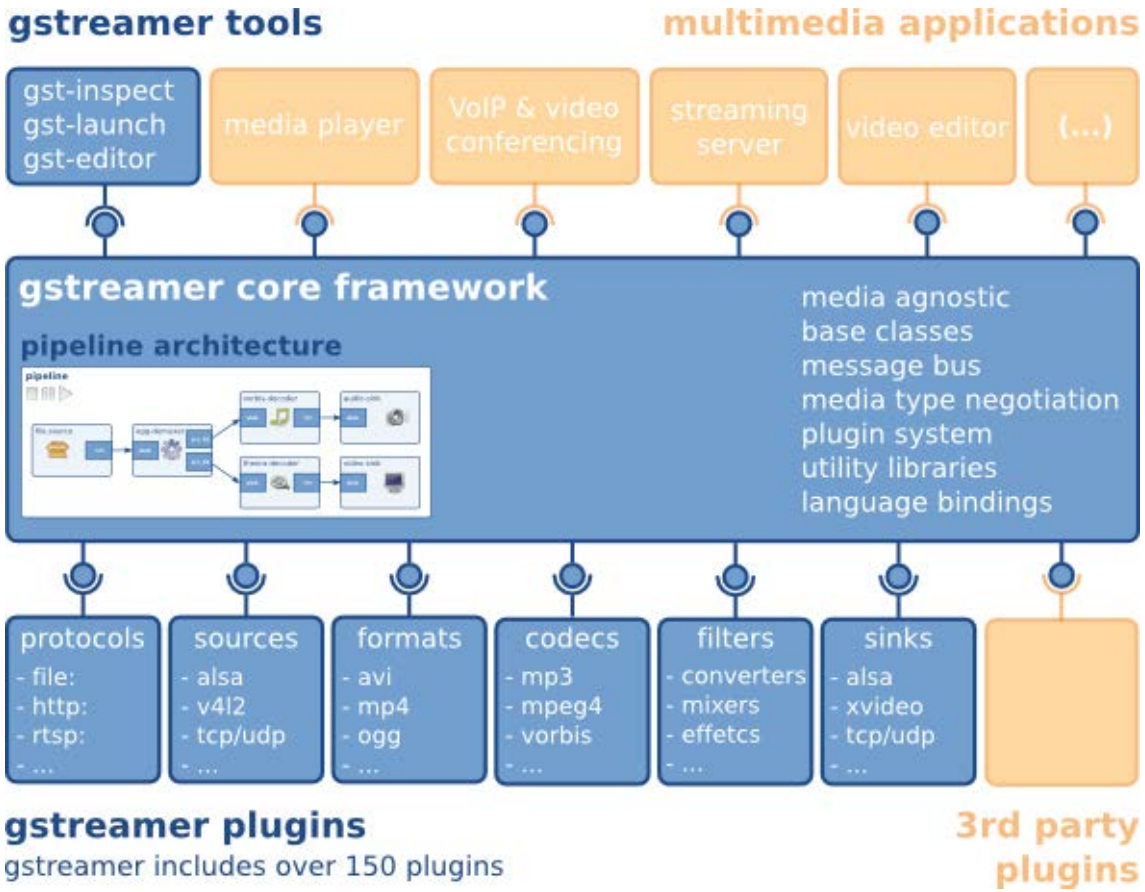
Architecture finale - portail



Développement – HTTPStreamer

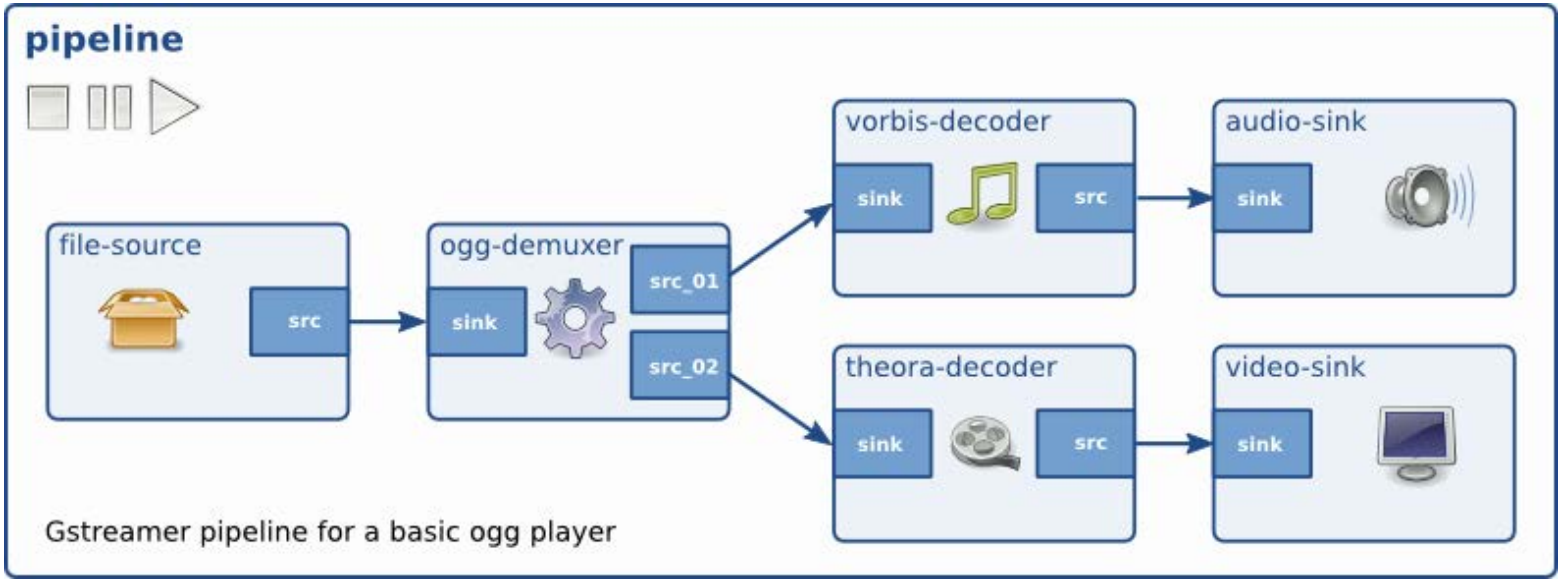


Développement: Gstreamer 1/2



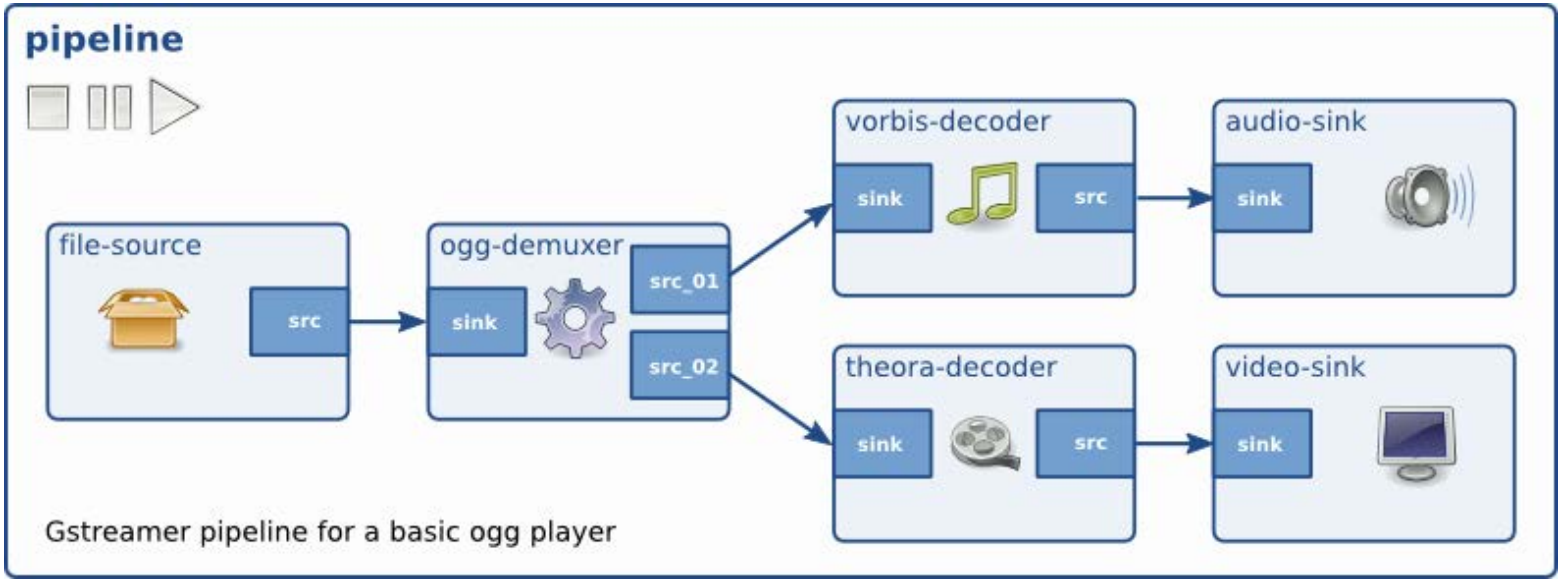
Développement: Gstreamer 2/2

‘gst-launch v4l2src device=%s !
video/x-raw-yuv,width=640,height=480 !
x264enc pass=cbr quantizer=20 tune=zerolatency !
avimux !tcpserversink port=9999' % device)



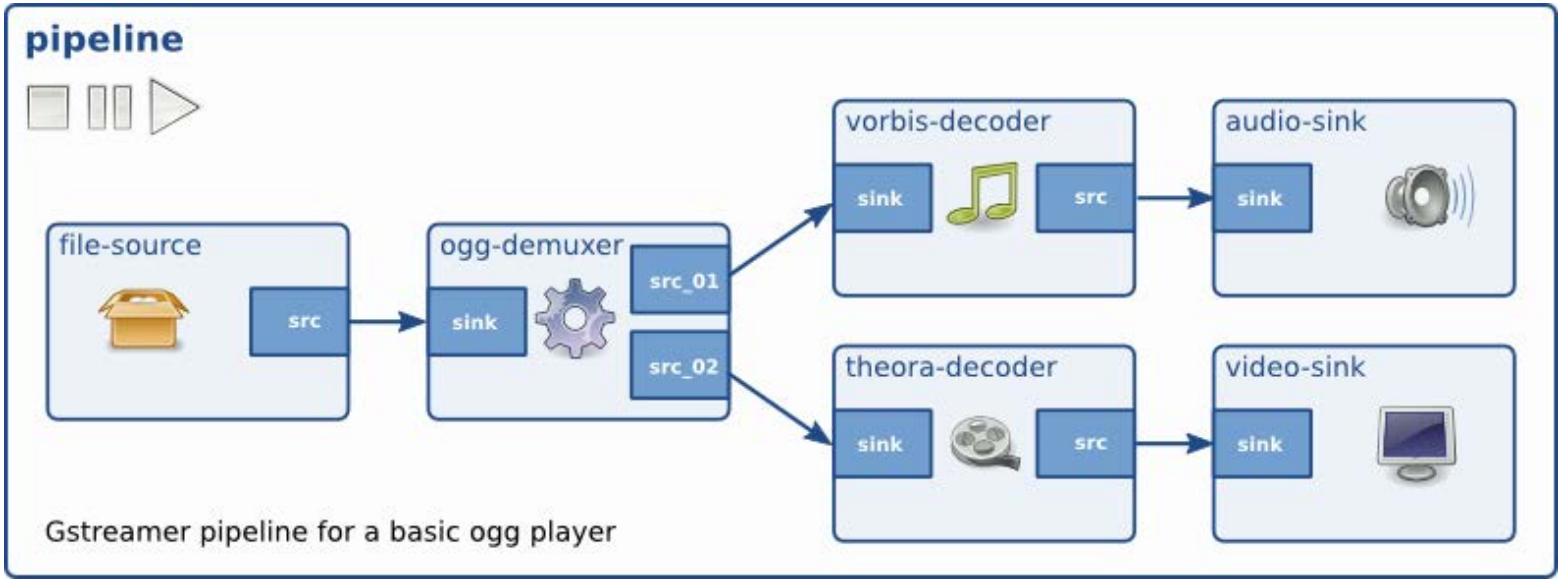
Développement: Gstreamer 2/2

‘gst-launch v4l2src device=%s !
video/x-raw-yuv,width=640,height=480 !
x264enc pass=cbr quantizer=20 tune=zerolatency !
avimux !tcpserversink port=9999' % device)



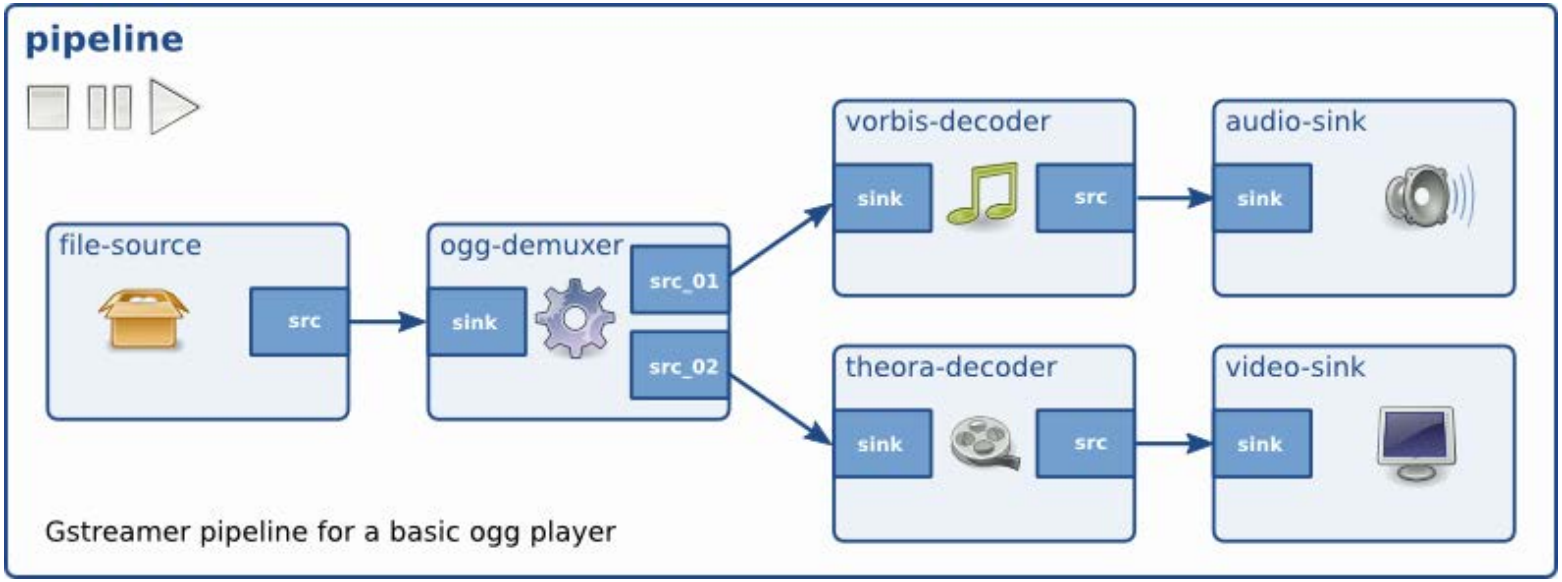
Développement: Gstreamer 2/2

‘gst-launch v4l2src device=%s !
video/x-raw-yuv,width=640,height=480 !
x264enc pass=cbr quantizer=20 tune=zerolatency !
avimux !tcpserversink port=9999' % device)



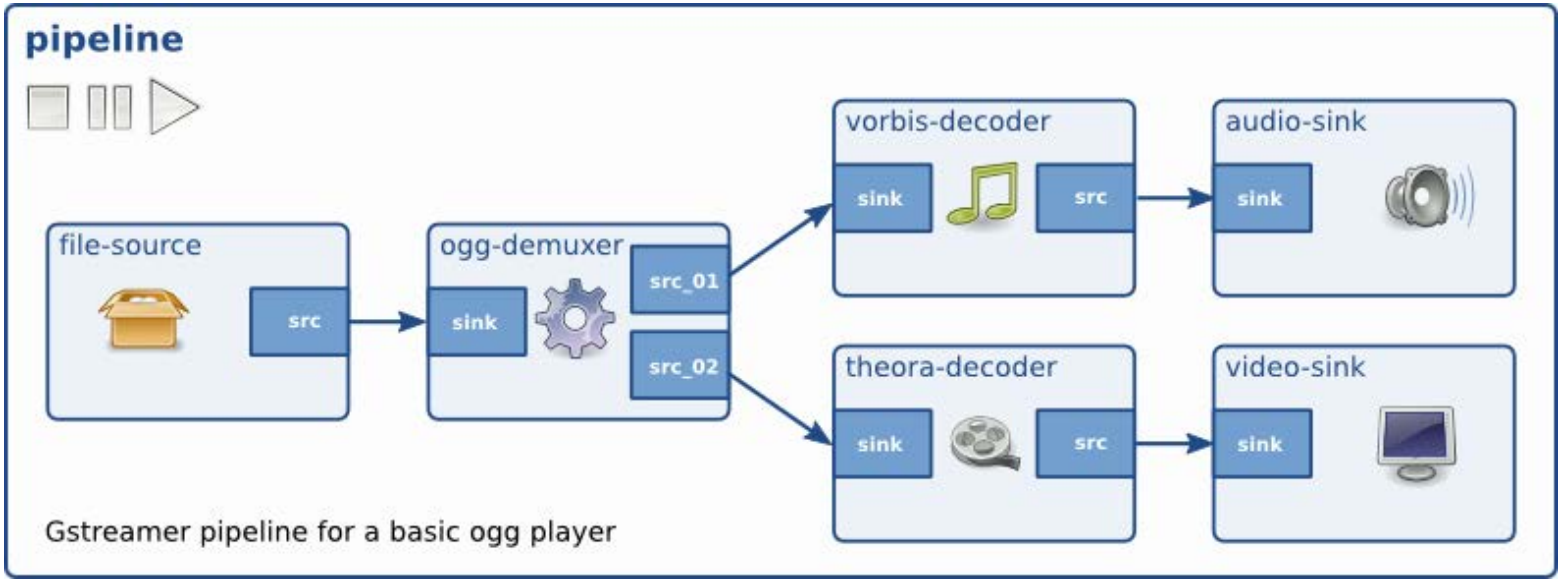
Développement: Gstreamer 2/2

‘gst-launch v4l2src device=%s !
video/x-raw-yuv,width=640,height=480 !
x264enc pass=cbr quantizer=20 tune=zerolatency !
avimux !tcpserversink port=9999' % device)



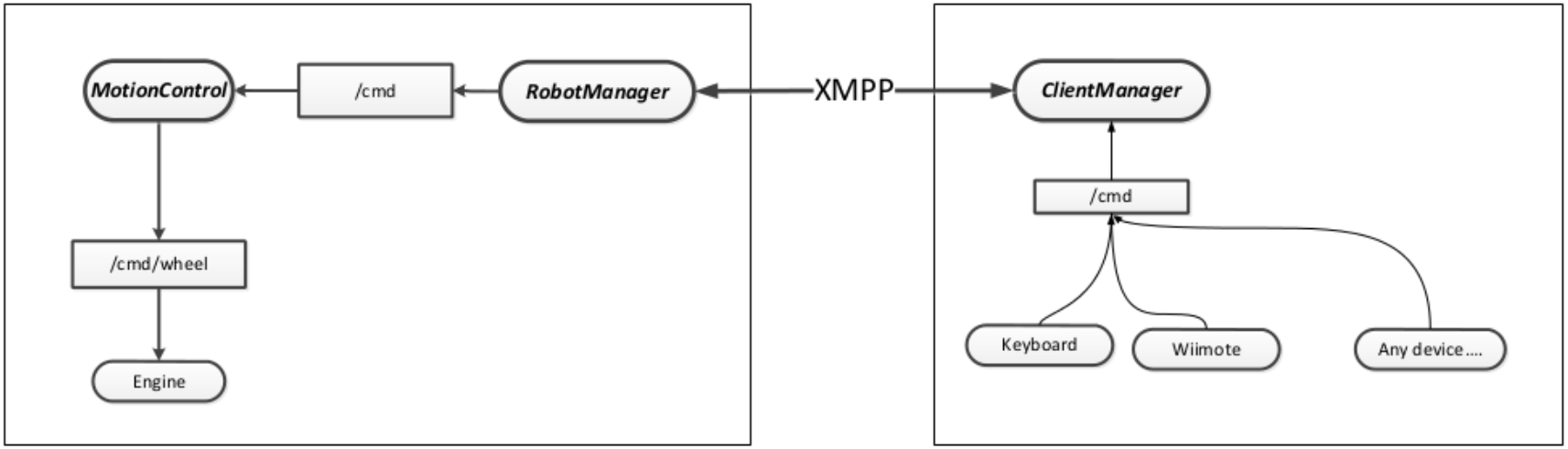
Développement: Gstreamer 2/2

‘gst-launch v4l2src device=%s !
video/x-raw-yuv,width=640,height=480 !
x264enc pass=cbr quantizer=20 tune=zerolatency !
avimux !tcpserversink port=9999' % device)

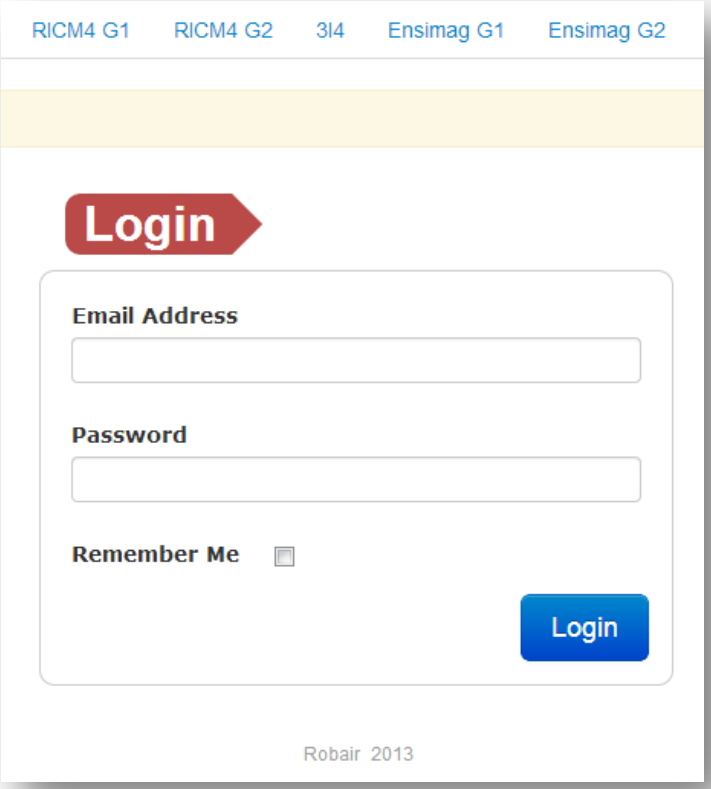


Développement – Périphériques de pilotage: la wiimote

- Noeud keyboard : Utilisation d'un keylogger
- Noeud wiimote : Utilisation de libcwiid
- Topic /cmd : commandes reçues de tout périphérique dans un format simple et unique : ajout de périphérique facile.



Développement – Portail de réservation




```
$ curl http://127.0.0.1:9090/api/new?jid=test2@test.com
{
  "jid": "test2@test.com",
  "end": [
    "2013-04-04",
    "03:31:56"
  ],
  "started": true,
  "start": [
    "2013-03-23",
    "13:45:17"
  ],
  "key": "2b86b2ebe-e7f0-4617-84de-7ef7ab79323a",
  "error": false,
  "expired": false,
  "id": 2
}
```

Développement – Portail de réservation

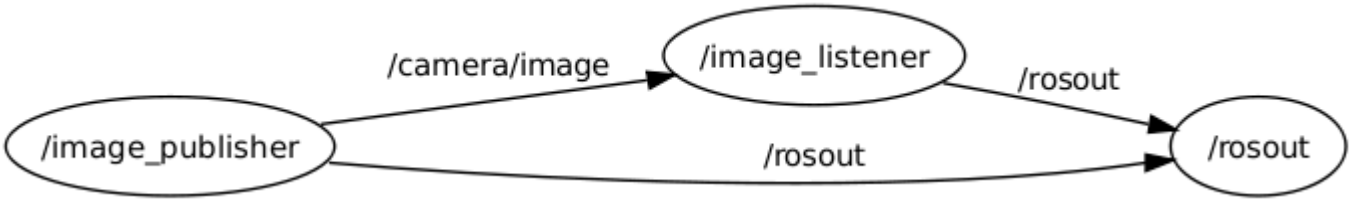
```
$ curl http://127.0.0.1:9090/api/check?key=1bde8bac8...558b790d7cc3
{
  "jid": "test@test.com",
  "end": [
    "2013-03-23",
    "13:44:17"
  ],
  "started": false,
  "start": [
    "2013-03-11",
    "23:57:38"
  ],
  "valid": true,
  "key": "1bde8bac8-f654-42d5-aa6f-558b790d7cc3",
  "expired": false,
  "id": 1
}
```

```
$ curl http://127.0.0.1:9090/api/check?key=bad_key
{
  "valid": false
}
```

Développement – ROS/XMPP

- ROS : mécanisme de Publication/Souscription
  => Fonctions callback

- Etendre le modèle ROS à travers XMPP
- Solution : callback distantes ! (RPC)




```
rospy.Publisher('/camera/image', Image)
rospy.Subscriber ('/camera/image', image_callback)
```

Développement – ROS/XMPP
RPC et Proxy

```
class Robot(sleekxmpp.ClientXMPP):

    @remote
    def add(self, *args):
        return sum(args)
```

```
client = ClientXMPP(jid, password)
robair_proxy = client.get_proxy("robair@example.com")

a = robair.add(2, 4, 5)
```

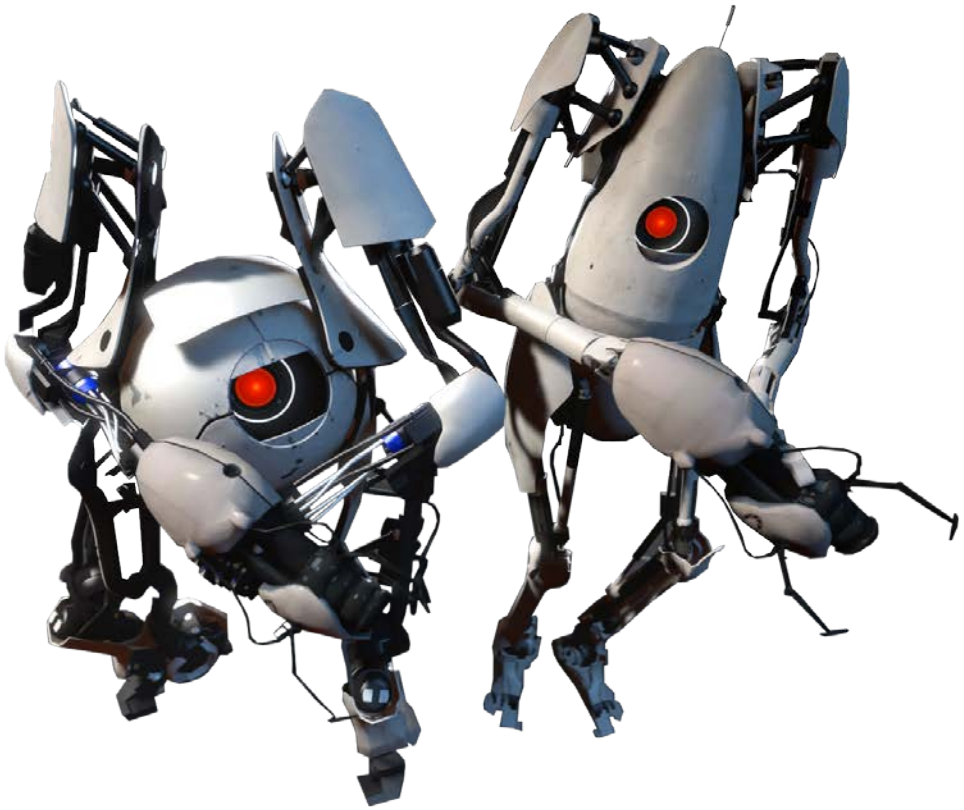
Développement – ROS/XMPP
Proxy et magie

```
class RemoteXMPPProxy(object):

    def __getattr__(self, name):
        return lambda *args, **kwargs: self.__rpc_send(name, *args, **kwargs)

    def __rpc_send(self, name, *args, **kwargs):
        [...]
```

Au final…

```
rospy.Subscriber('/cmd', Command, self.proxy_robot.publish_cmd)
```

Conclusion

- Projet difficile
  - Pas de robot pour tester
  - Gestion et communication
    - RICM4 : emploi du temps
    - Ensimag : projet très complexe.
    - 3I ??
- Côtoyer les nouvelles générations
- Choix de ROS : Pérennité et modularité
- Formateur :
  - Abandonner le rôle de développeur
  - Assumer celui d'ingénieur

Démo