



Livrables 2.1 & 2.2

Couche internet & logiciels audio/vidéos

Mise à jour du 13 /09 /2017

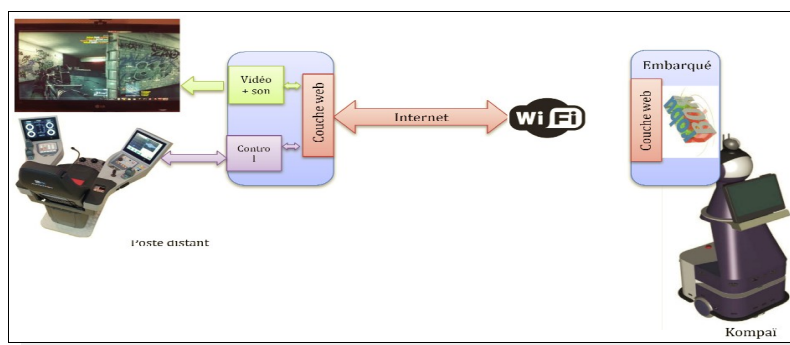
Thierry Bergeron, thbergeron@wanadoo.fr
Laboratoire I3S, Projet AZKAR, Sophia-Antipolis, France

Table des matières

A - Livrable 2.1 : Couche internet.....	2
B - Livrable 2.2 : Logiciels internet audio/vidéo.....	4
B.a - Déploiement & configuration.....	5
B.a.a - KomServer.....	5
B.a.b - Application 1to1.....	5
B.a.c - Proxy.....	6
B.a.d - Serveur Stun/Turn	6
B.a.e - VPN.....	6
B.b- Lancement & description de l'application.....	7
B.b.a - Connexion à l'application.....	7
B.b.b - Page d'accueil et interface d'administration (depuis la version 2.0.4).....	8
B.b.c - Page web du client "Pilote" (depuis la version 2.0).....	9
B.b.c.a - Module de commandes "Drive".....	10
B.b.c.b - Module "Auto Pilot".....	11
B.b.d - Interface WEB d'administration (Depuis la 2.0.4).....	12
B.b.d.a - GamePad.....	13
B.c - Code source.....	14
B.c.a - Organisation des fichiers.....	15
B.c.b - Algorithme principal.....	16
B.c.c - Limitations.....	19
B.c.d - Licence & versions.....	20

Ce workpackage constitue l'implémentation dans les équipements retenus pour les évaluations des solutions de contrôle-commande par internet. Ce sont essentiellement des développements logiciels, pour des équipements existants. Comme le montre la figure suivante, ces développements concernent :

- La couche web par laquelle transitent aussi bien les ordres de contrôle commande que les images issues du robot.
- L'interface entre la couche web embarquée et RobuBOX en charge de la gestion elle même du contrôle des mouvements du robot et de l'interaction avec l'utilisateur.



Synoptique des couches logicielles dans un schéma classique de contrôle distant de robots

A - Livrable 2.1 : Couche internet

Cette couche internet prend en charge:

- La gestion et l'optimisation des trames de commandes du robot,
- Les informations en provenance du robot (géolocalisation, niveau de batterie, etc),
- Les images et le son vers et en provenance du robot.,
- Les différents protocoles de connexion/déconnexion entre l'utilisateur et le robot.

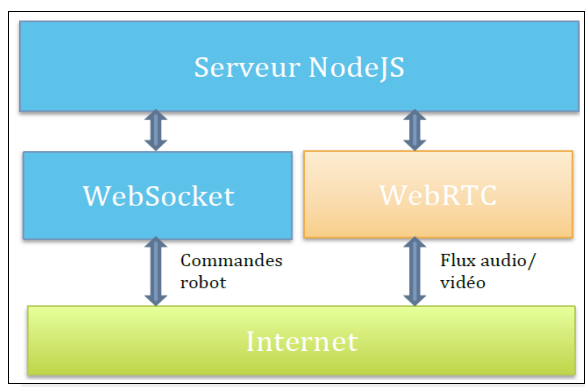


Schéma de principe de la couche internet

Optimisation des commandes : Pour réduire le nombre et la taille des requêtes de contrôle, il est fait appel à la technologie WebSocket. WebSocket est l'un des protocoles de transport les plus polyvalents et souples actuellement disponibles. L'API simple et minimale permet à la couche de livrer des protocoles arbitraires d'application entre client et serveur dans un mode de transmission en continu, où chaque côté peut envoyer des données à tout moment.

Informations techniques (Cartographie, géolocalisation, batterie..) : Là aussi, pour des raisons de fiabilité et de robustesse (reconnexions transparentes), il est fait appel à la technologie WebSocket pour faire transiter des informations techniques entre le robot et le poste client. Ainsi en cas déconnexion peer-to-peer, ces informations restent toujours disponibles.

Optimisation des flux audio/vidéo : le streaming de la vidéo et de l'audio nécessite des débits importants de données entre le robot et l'opérateur. La technologie WebRTC permet d'optimiser au maximum l'échange de ces données par une connexion de type pair-à-pair et permet l'utilisation de médias haute définition.

Protocoles de connexion/déconnexion: Compte-tenu de ses qualités (réactivité, transparence), la technologie WebSocket est encore mise à contribution comme canal de signaling pour gérer les protocoles de connexions et de reconnexion WebRTC. Cela permet d'ajouter de la "robustesse" aux communications audio/vidéo en peer-to-peer de WebRTC.

Ainsi, grâce à un serveur NodeJS installé sur un serveur distant, les commandes sont envoyées du poste de pilotage vers le robot via les Web Socket et les flux vidéo et audio bidirectionnels via WebRTC. De plus, la gestion asynchrone des requêtes et l'architecture basée sur les événements de NodeJS permet une meilleure gestion des commandes en boucle fermée.

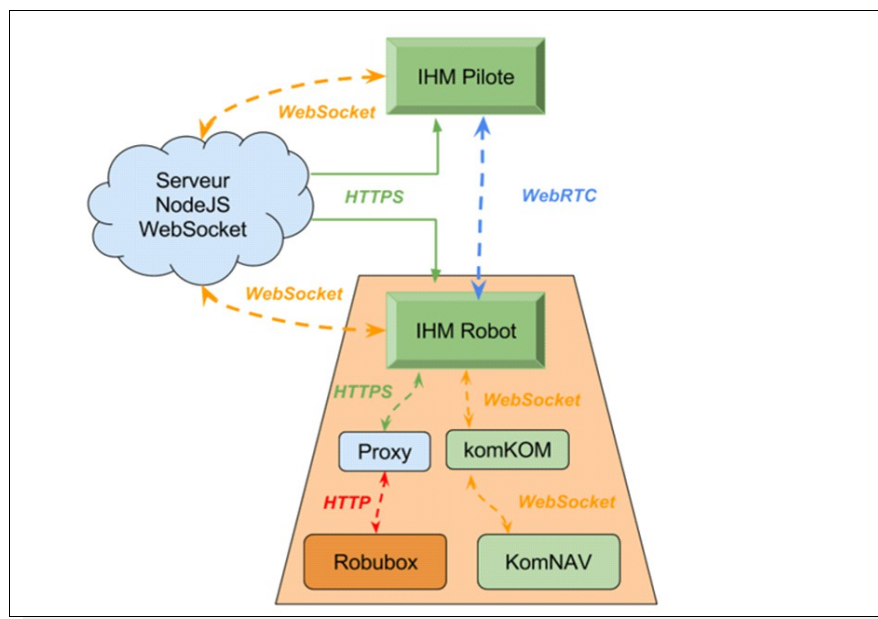


Schéma de déploiement de la couche internet

B - Livrable 2.2 : Logiciels internet audio/vidéo

Le livrable 2.2 est constitué d'un ensemble de logiciels nécessaires à l'écosystème de la solution de contrôle-commande par internet. Il comprend des applications développées au laboratoire I3S à l'aide de technologies open-sources et fournies dans le livrable, mais aussi des applications open-source tierces non fournies qu'il faudra télécharger à part.

Les applications développées au laboratoire I3S, en collaboration avec Robosoft et Anotherworld:

- Le logiciel principal, baptisé "1to1", exposant deux clients web, l'un destiné au robot et fonctionnant en mode passif (aucune intervention de l'utilisateur n'est requise sur cette IHM) , l'autre faisant office de poste de pilotage distant. Ce logiciel assure en arrière-plan les fonctions de Signaling, de configuration des caméras distantes, de "robustesse" du signal, et la commande à distance à l'aide d'un Gamepad ou d'un joystick virtuel, mais aussi la cartographie de géolocalisation et la liaison audio/vidéo. Exposée à l'aide de NodeJS en HTTPS, fonctionnant exclusivement sous Chrome et reposant sur un serveur webSocket, cette application est écrite en javascript et HTML5. Cet ensemble de scripts fait appel à plusieurs bibliothèques open-source dont la plus connue est jQuery.
- Un proxy à installer sur le robot afin d'assurer le dialogue entre le navigateur (client web) du robot et le système embarqué "KomServer".

Les applications Open-Source Tierces:

- Serveur nodeJS & WebSocket installé sur un serveur distant.
- Serveur STUN/TURN (RFC-5766) installé sur un serveur distant.
- VPN (solution commerciale ou serveur de type OpenVPN) pour les rares cas où les filtrages réseaux rendraient les relais TURN inopérants.

Concernant les serveur STUN/TURN, les solutions commerciales semblent encore peu nombreuses et seules deux sont proposées sous forme de service (xirsys.com & anyfirewall.com). Toutefois, aucune de ces deux solutions n'as donné satisfaction lors des essais:

- Soit l'API était écrite en php et son portage en javascript (non documenté) s'est révélé inopérant.
- Soit les credentials éphémères fournis par l'API avaient une durée de vie trop courte pour s'accorder à l'algorithme de signaling WebRTC de l'application 1to1.

La solution OpenSource RFC-5766 qui permet de relayer les connexions peer-to-peer en TURN si le STUN ne passe pas fonctionne dans la majorité des cas. Toutefois, à notre grande surprise, il s'est révélé des situations où les filtrages des réseaux Wifi étaient tels que mêmes ces relais TURN ne passent plus (c'est le cas des réseau Wifi universitaire...). Ce cas de figure n'est malheureusement pas documenté ni même évoqué dans les forums spécialisés. La seule solution de contournement dans ce cas est d'utiliser un VPN afin de "tunelliser" la connexion des clients filtrés.

Des solution VPN open-source existent (la plus recommandée étant OpenVPN), mais sont complexes et laborieuses à mettre en œuvre. Le plus simple et le moins coûteux en terme de ressources temps est de souscrire à une offre commerciale illimitée en débit et de connecter au web le client "filtré" par le biais de ce VPN.



B.a - Déploiement & configuration

B.a.a - KomServer

Au préalable, récupérer manuellement la cartographie du lieu où évoluera le robot. Il faudra ensuite utiliser l'interface d'administration de l'application 1to1 pour uploader et activer la carte.

B.a.b - Application 1to1

Le code source de l'application est disponible sur GitHub à l'adresse suivante:

<https://github.com/AzkarProject/websocket-azkar/tree/1to1-cnrs>

1. La solution la plus simple consiste à décompresser le contenu du zip dans un répertoire dédié Sur le serveur choisi (local, distant ou virtuel, avec nodeJS et npm préalablement installé). Toutefois, pour des raisons d'évolutivité et de maintenance, il est préférable d'utiliser un environnement de type GitHub pour déployer le projet.
2. Renseigner les paramètres de configuration `js/common_app_settings.js`.
 1. Dans la fonction `setServers()`:
 1. Générer des certificats auto-signés en suivant le tutoriel présent dans le répertoire `ssl` de l'application, puis renseigner les variables aux certificats générés. Il est toutefois possible de laisser les certificats et clefs proposés par défaut (`hacksparow-key` et `acksparow-cert`), mais ceux-ci ne sont pas mémorisables par le navigateur et il faudra à chaque nouvelle instance de Chrome autoriser manuellement la connexion `https` sur les deux clients.
 2. Dans la fonction `setIceServers()` Renseigner la liste des serveurs STUN/TURN (). Il est conseillé de ne mettre qu'un seul serveur TURN fonctionnel pour diminuer les échanges de "candidates" superflus et raccourcir les délais de connexion/reconnexion.
 3. Pour lancer l'application, se placer dans son répertoire racine et lancer la commande appropriée (sous Windows: "node server" et sous Linux: "sudo nodejs server.js")

B.a.c - Proxy

Récupérer sur dans le Github du projet ou dans son archive zip le répertoire "PROXY" et le copier quelque part sur le pc qui embarque la robuBox. L'utilisation de ce proxy est inutile si le mode "fakeRobubox" est activé. Pour lancer le proxy, se placer dans son répertoire et faire (sous Windows: "node proxy.js"). Remarque: Ce proxy est obligatoire pour communiquer avec les caméras PTZ (Pan Tilt Zoom).

B.a.d - Serveur Stun/Turn

Téléchargement, installation & configuration basique sur un serveur Ubuntu 14.04 desktop:

1. Dans la logithèque ubuntu, entrer dans l'onglet de recherche "rfc 5766" et sélectionner l'archive "TURN and STUN server for Voip" et précisant "rfc5766-turn-server" et procéder à l'installation. Les fichiers seront installés dans le répertoire `usr/share/rfc5766-turn-server`.
2. Copier les fichiers de configuration `turnserver.conf` et `turnuserdb.conf` présent dans `usr/share/rfc5766-turn-server/examples/etc`
3. Renseigner ces fichiers en suivant les instructions en commentaires et les placer dans etc/
4. Dans un terminal, lancer manuellement le serveur en tapant par exemple la ligne de commande suivante: "`sudo turnserver -v -r xx.xxx.xxx.xx -a -b turnuserdb.conf -c turnserver.conf`". Les détails sont les suivants:
 - `-v` = Petite verbosité (ou `-V` = grande verbosité)
 - `-r` = Realm (obligatoire avec le mecanisme "Longs terms credentials")
 - `xx.xxx.xxx.xx` correspondant a l'adresse IP ou à l'url du serveur
 - `-a` = Mécanisme "Longs terms credentials" (obligatoire pour WebRTC)
 - `-b` = Nom du fichier stockant les credentials utilisateurs
 - `-c` = Nom du fichier de configuration
5. Pour vérifier que le serveur est bien lancé, dans un teminal taper la commande "`sudo pgrep turnserver`" qui retournera l'ID du processus si celui-ci est actif.
6. Si le processus est lancé, pour l'arrêter taper "`sudo pkill turnserver`".

B.a.e - VPN

De nombreuses solutions commerciales existent et sont extrêmement simples à mettre en œuvre. Il suffit de souscrire à l'abonnement désiré auprès du fournisseur de service et de charger le client proposé sur la machine à connecter. Nous utilisons au laboratoire I3S une solution commerciale basée sur OpenVPN, privatetunnel.com, qui nous permet de passer les filtrages des réseaux wifi universitaires. Cette solution nous a donné entière satisfaction en terme de rapidité, de stabilité et de facilité de mise en œuvre.


A titre d'exemple, un abonnement permet de connecter jusqu'à 10 appareils simultanément pour un coût de 25/30 € par ans sans limitation de bande passante et avec un débit très suffisant pour faire transiter un flux audio/video bidirectionnel en haute définition.

B.b - Lancement & description de l'application

B.b.a - Connexion à l'application

1. Côté Robot:
 1. Activer la solution embarquée KomServer. (Inutile si fakeRobubox est activé).
 2. Activer le proxy. (Inutile si fakeRobubox est activé).
 3. Dans Chrome, ouvrir le proxy et garder l'onglet actif (Inutile si fakeRobubox est activé).
 4. Dans un autre onglet, entrer l'url de l'application puis aller sur la page "Robot Interface."
 5. Entrer un nom d'utilisateur si demandé ou laisser par défaut.
(Ce pseudo est enregistré dans un cookie)
 6. Attendre qu'un pilote se connecte et laisser faire...
2. Côté pilote:
 1. Dans Chrome, entrer l'url de l'application puis aller sur la page "Pilote Interface."
 2. Entrer un nom d'utilisateur si demandé ou laisser par défaut.
(Ce pseudo est enregistré dans un cookie)
 3. Sélectionner la résolution voulue dans le module « Video quality », la connexion se lance automatiquement.
 4. Lancer la connexion et attendre que la caméra distante apparaisse.
 5. Si le message d'erreur suivant "Connection Failed" apparaît, alors l'un des deux clients (ou les deux) sont probablement filtrés par le réseau. Il faudra alors ouvrir une connexion « tunellisée » VPN sur le client concerné et reprendre l'opération.
 6. S'il n'y a aucun message d'erreur mais que l'image distante n'apparaît pas, on peut dans un premier temps "fermer la connexion" et relancer celle-ci en appuyant sur le bouton « close » du module « video quality ». Si ça échoue, alors il faut recharger la page, reprendre tous les réglages (caméras, résolutions, etc..) avant de lancer une nouvelle connexion.
3. Interface d'administration (depuis la 2.04)
 1. L'interface d'administration est accessible directement depuis la page d'accueil. Onglet 'Robot Settings'
 2. Les onglets Map, Scene, Medias, Trails et Object sont dédiés à l'administration des outils d'enrichissement de visite en web sémantique

B.b.b - Page d'accueil et interface d'administration (depuis la version 2.0.4)



AZKAR Project Version 2.0.5 (Branche Ito1-2017 / Serveur: AZKAR-1)
© 2015-2017 - CNRS (Laboratoire I3S) / Université de Nice
Bienvenue visiteur

Home | Robot interface | Pilot interface | Robot Settings | Map | Scenes | Medias | Trails | Objects

First connection

Controls

There are different ways to pilot the robot: with the mouse only, with the keyboard or with an Xbox 360 controller.

- With the mouse :

To move the robot, hold the virtual joystick in the required direction. When the joystick is realised, it come back to the neutral position.
The robot can also be controlled by the virtual arrow keys.

In order to control the camera, there is the camera control pad which allows you to look around and zoom.

- Using the Keyboard :

In order to control the robot via the keyboard, you need to hold ctrl and use the arrow keys.

Controlling the camera via the keyboard is not possible at the moment. You have to use the camera pad.

- Controller :

When using an Xbox 360 Controller there are two ways to move the robot:
Hold A for normal speed or X for slow speed and:

- Hold RT to move forward
- Hold LT to move backward
- Use the left joystick to turn

OR

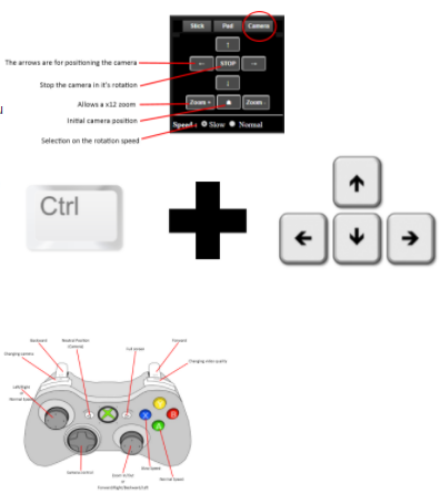
- Press the left joystick and use to move the robot at normal speed


Camera:

Use the arrows to control the camera

Other Controls

- Press RB to change the video quality (increasing order)
- Press LB to switch between the cameras
- Press Start for full screen mode
- Press Back to get the camera to its home position





AZKAR Project Version 2.0.5 (Branche Ito1-2017 / Serveur: AZKAR-1)
© 2015-2017 - CNRS (Laboratoire I3S) / Université de Nice
Bienvenue 'Administrateur'

Home | Robot interface | Pilot interface | Robot Settings | Map | Scenes | Medias | Trails | Objects

Connected users

Simulation mode "FakeRobubox" = disabled

Map setting = valerian3.png

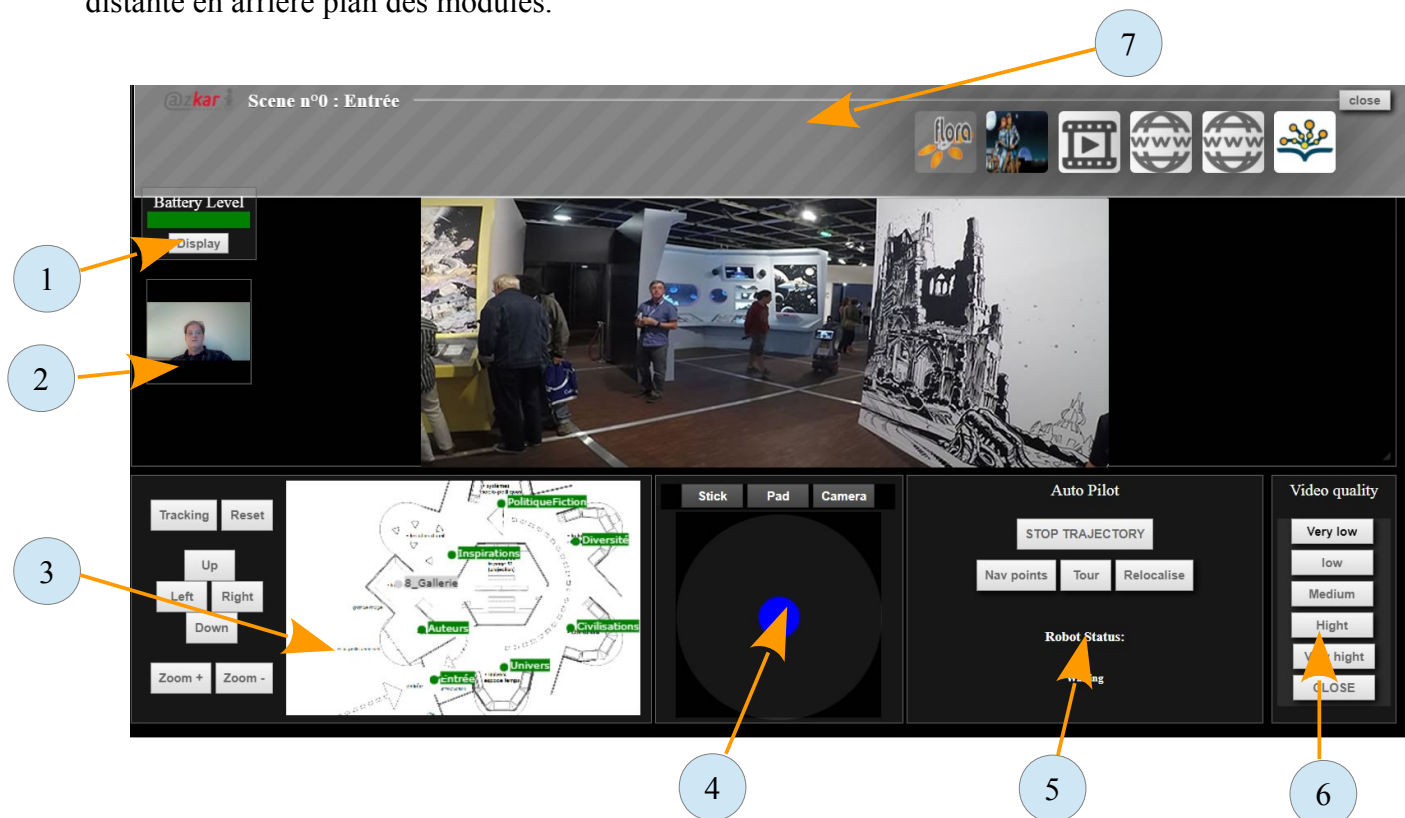
Xbox 360 gamepad = enabled

Pan/Tilt camera setting = IP Camera : 192.168.1.88 (IP ptzOptic MGG)

IP Robot setting = IP Robot : 192.168.1.74:7007 (IP Komserver MGG/CSI)

B.b.c - Page web du client "Pilote" (depuis la version 2.0)

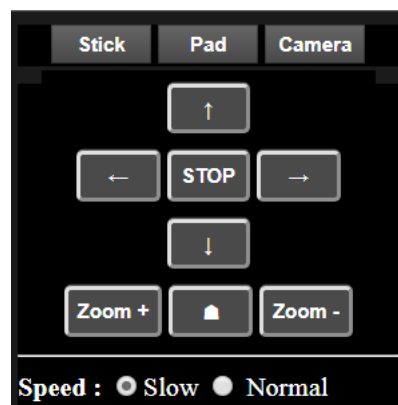
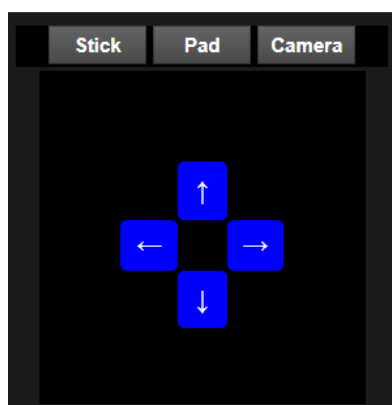
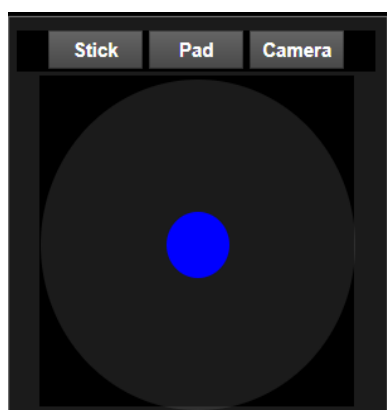
Le système d'interfaces Robot et pilote est composé de modules repositionnables par l'utilisateur lui-même à l'aide de glisser/déposer. Ces modules sont escamotables à volonté via un menu de configuration. Le mode plein écran est remplacé par la possibilité de redimensionner la vidéo distante en arrière plan des modules.



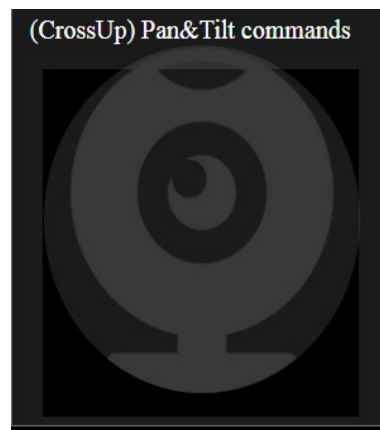
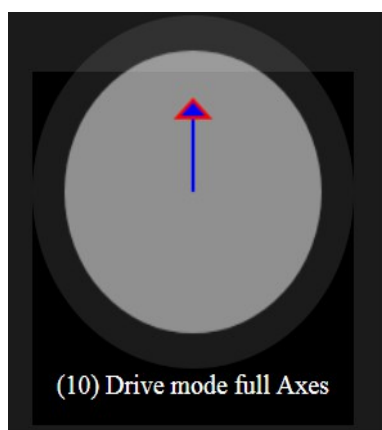
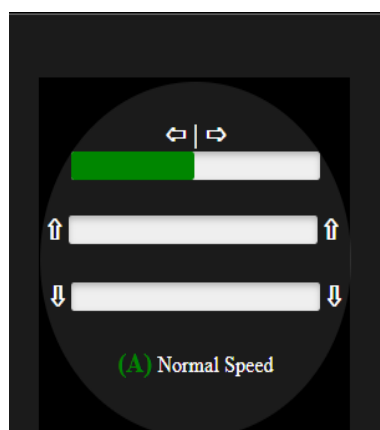
1. Ouverture/fermeture des menus de configuration
2. Affichage de la caméra locale (pilote)
3. Module de cartographie « Map »
4. Module de commande « Drive » (pilotage robot & Caméra)
5. Module de navigation « Auto Pilot » :
 - Menu de navigation par points d'intérêts,
 - menu des parcours recommandés (Web sémantique),
 - menu de relocalisation.
6. Module « Video Quality » d'ouverture/fermeture caméra et sélection de la qualité vidéo
7. Bandeau escamotable de ressources recommandées (web sémantiques)

B.b.c.a - Détails du module de commandes "Drive"

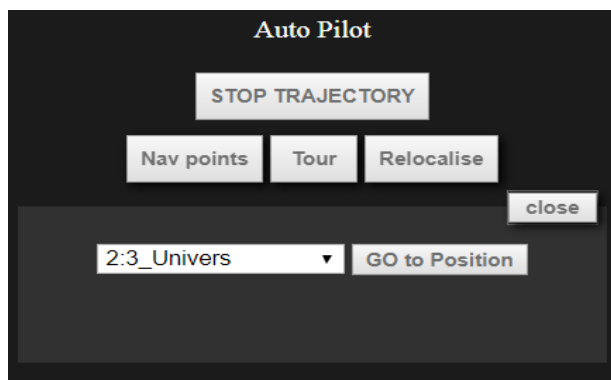
Commandes virtuelles



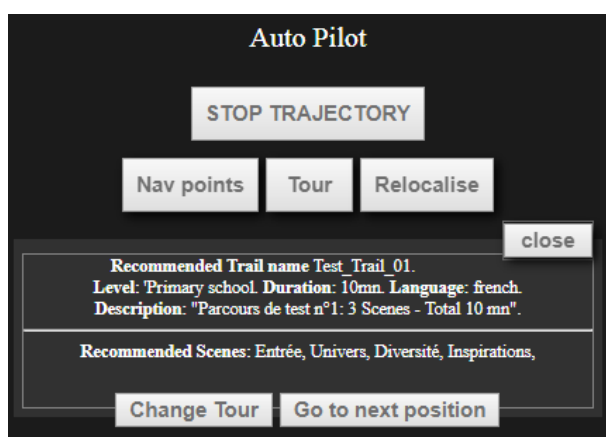
Commandes Gamepad



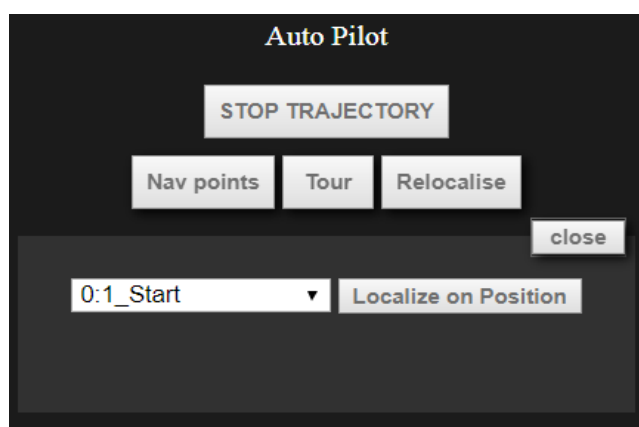
B.b.c.b - Détails du module "Auto Pilot"



Menu "Nav Points" navigation par POI (Poits of Interest)

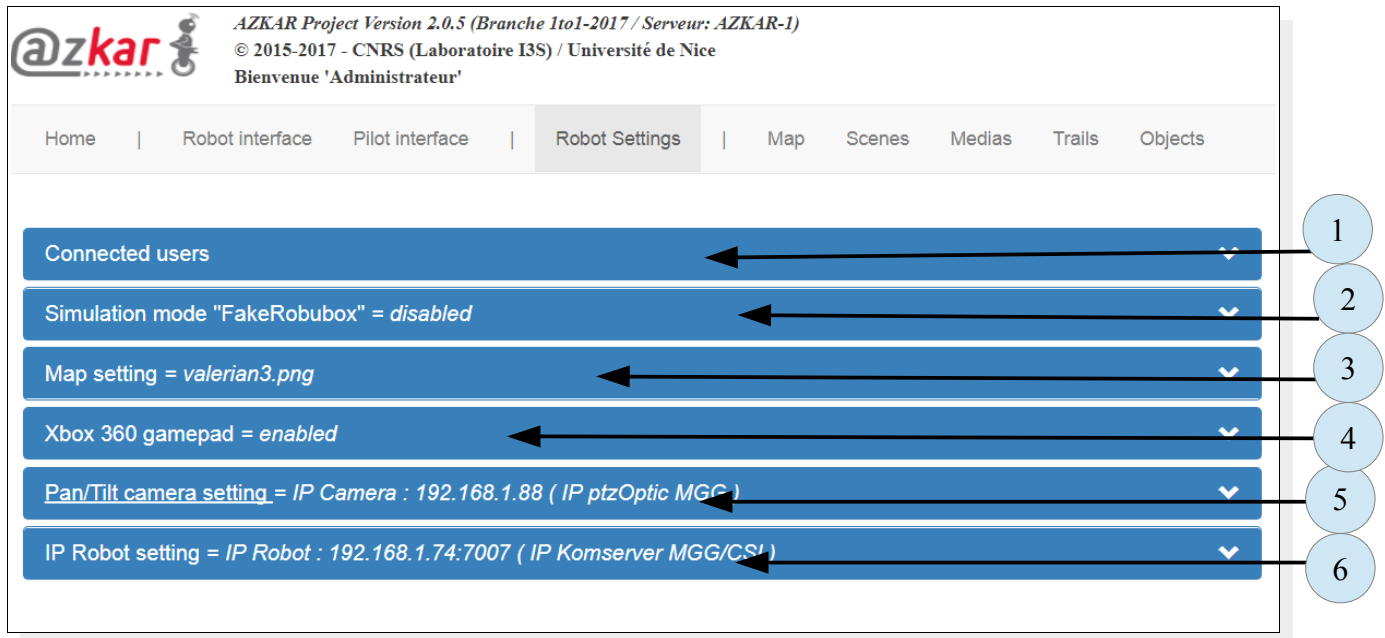


Menu "Tour" (recommandations de parcours en web sémantique)



Menu "Relocalisation"

B.b.d - Interface WEB d'administration (Depuis la 2.0.4)

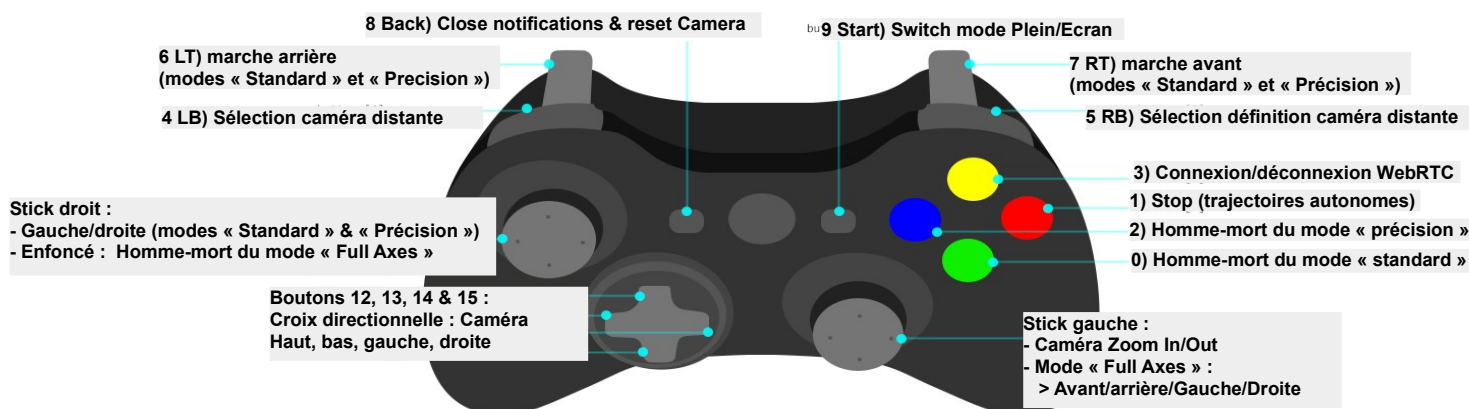


1. Onglet des commandes d'éjection et de reload forcé des pilotes et robots connectés
 - Le bouton « Reload » permet de forcer le rafraîchissement de la page du "pilote" ou du "robot". Attention, si exécuté dans une même instance du navigateur (dans un autre onglet par exemple), cela forcera aussi le retour du client connecté dans l'autre onglet à la page d'accueil...
 - Le bouton « Ejection » permet de forcer la déconnexion du pilote ou du robot et le redirige sur la page d'accueil afin de vider la liste des utilisateurs connectés. L'inconvénient est qu'il faut manuellement reconnecter le robot à sa page (via teamviewer par exemple)
 - En cas d'utilisateur fantôme (un utilisateur qui n'a pas pu se déconnecter proprement de sa page), le bouton « Ejection de tous les clients » en plus d'envoyer un ordre d'éjection à tous les clients nettoie la liste des utilisateurs connectés, y compris les utilisateurs fantômes.
2. Onglet de bascule en mode de simulation robot "FakeRobubox". Emulation logicielle d'un système embarqué pour la cartographie, la jauge de batterie et les commandes drive. Si activé, ne tient pas compte de la carte chargée. Simulation basée sur la carte du laboratoire I3S (Labo261.png). La navigation automatique par POI et les commandes de relocalisations sont désactivées dans ce mode. Vous pouvez l'utiliser pour simuler un client robot dans un onglet et un client pilote dans un autre onglet
3. Onglet d'upload, de sélection et de suppression à la volée de maps pour la cartographie
4. Onglet de Gestion du gamepad
5. Onglet de gestion IP des caméras Pan/Tilt.
6. Onglet de gestion IP du PC embarqué du robot affecté à la navigation.

B.b.d.a - GamePad

Le Gamepad est reconnu automatiquement par l'application dès qu'il est branché ou allumé.. Celui-ci doit obligatoirement être de type Xbox 360. Tout autre modèle ne sera pas pris en compte.

Au premier envoi d'une commande valide, l'affichage du bloc de commande bascule et affiche en temps réel les vitesses avant/arrière et la direction sous forme de jauges. Pour revenir au Joystick/pad virtuel, il suffit de débrancher ou d'éteindre le Gamepad et l'affichage bascule automatiquement. Depuis la Version 1.6.1, il est aussi possible de désactiver la détection du Gamepad depuis l'interface d'administration.



- **Bouton 0 (A)** Homme mort pour la vitesse normale (conduite mode «standard»)
- **Bouton 2 (X)** Homme mort pour la vitesse réduite (conduite mode «précision»)
- **Bouton 3 (Y)** Connexion/déconnexion WebRTC
- **Bouton 1 (B)** Commande STOP (trajectoires autonomes par points d'intérêts)
- Bouton 4 (LB): Cycle de sélection caméra distante
- Bouton 5 (RB): Cycle de sélection définition caméra distante
- Bouton 6 (LT): Gâchette marche arrière
- Bouton 7 (RT): Gâchette marche avant
- Bouton 8 (Back): Fermeture des notifications & reset orientation/zoom de la caméra Pan/Tilt
- Bouton 9 (Start): Activation/désactivation du mode plein écran
- Bouton 10 (Joystick gauche enfoncé): Homme mort pour la conduite mode «Full Axes» du joystick gauche
- Bouton 11 (Joystick droit enfoncé): Non attribué
- Joystick droit: Gauche/droite conduite mode «standard» et mode «précision»
- Joystick gauche: Si Joystick droit enfoncé: Conduit en mode «Full Axes» (Avant/arrière/gauche/droite)
- Joystick gauche: Si Joystick droit relâché: Zoom caméra avant/arrière

Pour que les déplacements soient actifs, il est impératif de maintenir enfoncé l'un des boutons homme-mort: A ou X pour la conduite traditionnelle (joystick gauche droite/gauche + gâchettes) ou le bouton 10 du (Joystick gauche enfoncé) pour activer le mode «Full Axes» du Joystick droit.



B.c - Code source.

Le code source de l'application est disponible sur GitHub à l'adresse suivante:

<https://github.com/AzkarProject/websocket-azkar/tree/1to1-cnrs>

A chaque déploiement d'une nouvelle version sur un serveur de test ou de production, **il faut impérativement supprimer le fichier parametre.json avant de relancer l'application**. Ce fichier de paramètre sera recréé automatiquement avec ses nouvelles valeurs par défaut au premier lancement.

B.c.a - Organisation des fichiers

Pour adapter l'application aux besoins spécifiques de chaque évaluation et à l'évolution des systèmes embarqués par le robot (passage progressif de la Robubox au nouveau système KomServer), le code de l'application a été découpé en différents modules fonctionnels. La convention de nommage adoptée est la suivante:

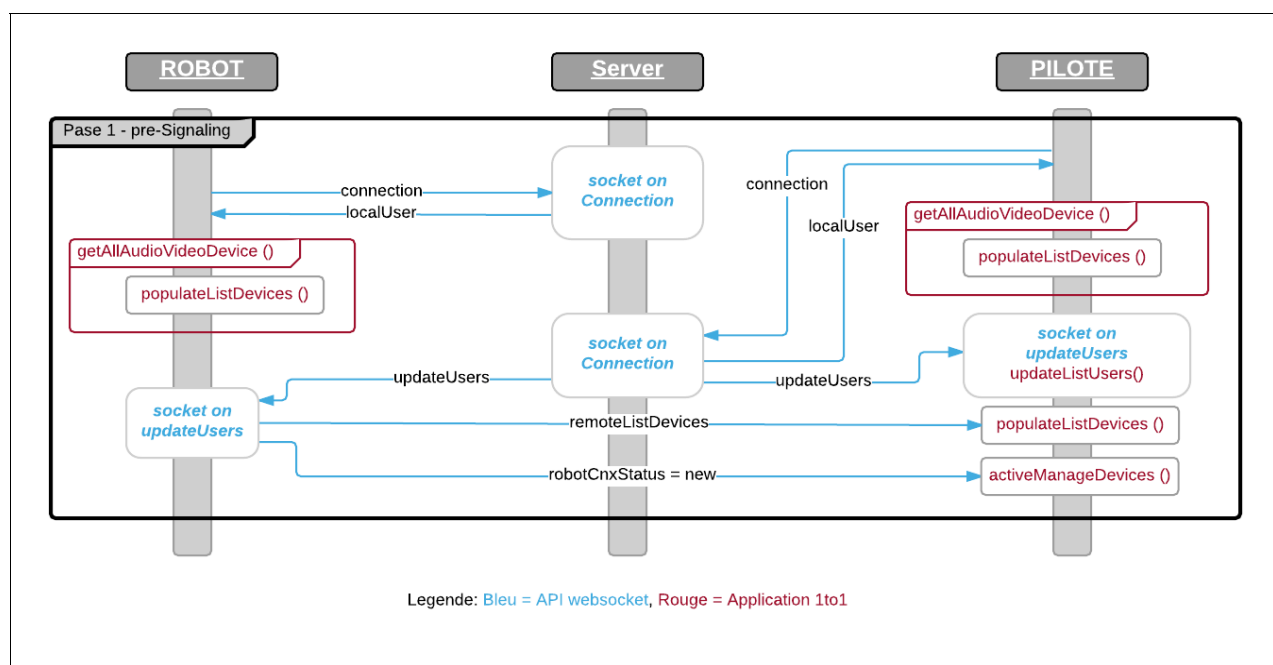
- **common_** pour les fonctions génériques, objets et paramètres communs à la partie cliente et serveur. Ces éléments doivent être chargés en tout premier lieu.
 - common_tools.js
 - common_appsettings.js
 - common_models.js
- **1to1_** pour les scripts procéduraux et l'algorithme principal de l'application. Ces scripts doivent être chargés et exécutés dans un ordre précis. Il ne devrait pas avoir lieu de les modifier sauf pour des raisons de débogage ou de factorisation du code.
 - 1to1_01_webSocket.js
 - 1to1_02_pre_signaling.js
 - 1to1_03_signaling
 - 1to1_04_post_signaling.js
- **module_** pour les sous-ensembles fonctionnels regroupés par usage.
 - module_carto.js
 - module_fake_robubox.js
 - module_forms.js
 - module_gamepad.js
 - module_foscam.js
 - module_ihm.js
 - module_navigation.js
 - module_navigation_interface.js
 - module_notifications.js
 - module_userconnections.js
 - module_logs.js
 - module_komremote.js*
 - module_komcom.js**

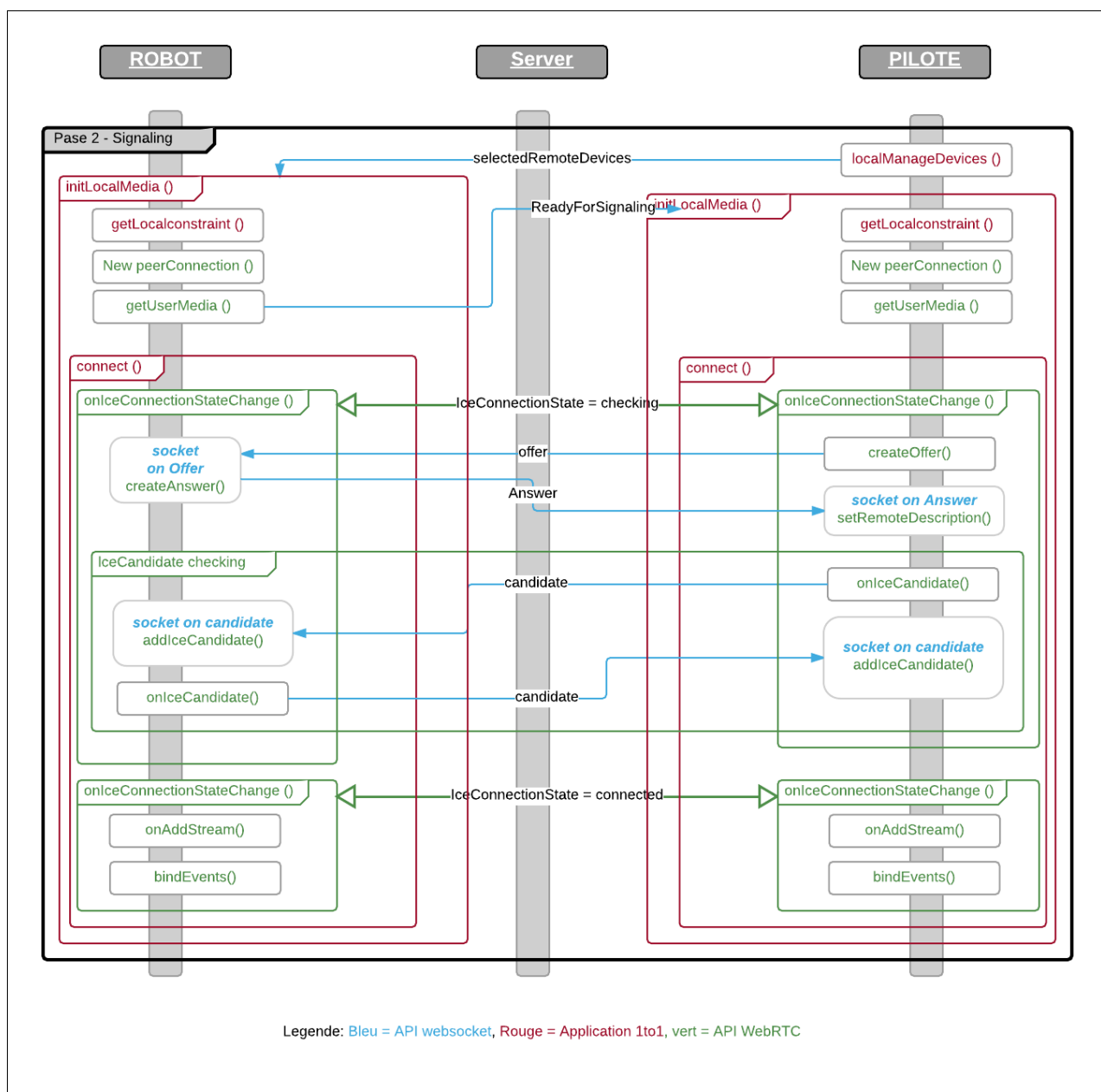
* KomRemote est le module permettant de gérer le joystick virtuel et la croix directionnelle virtuelle

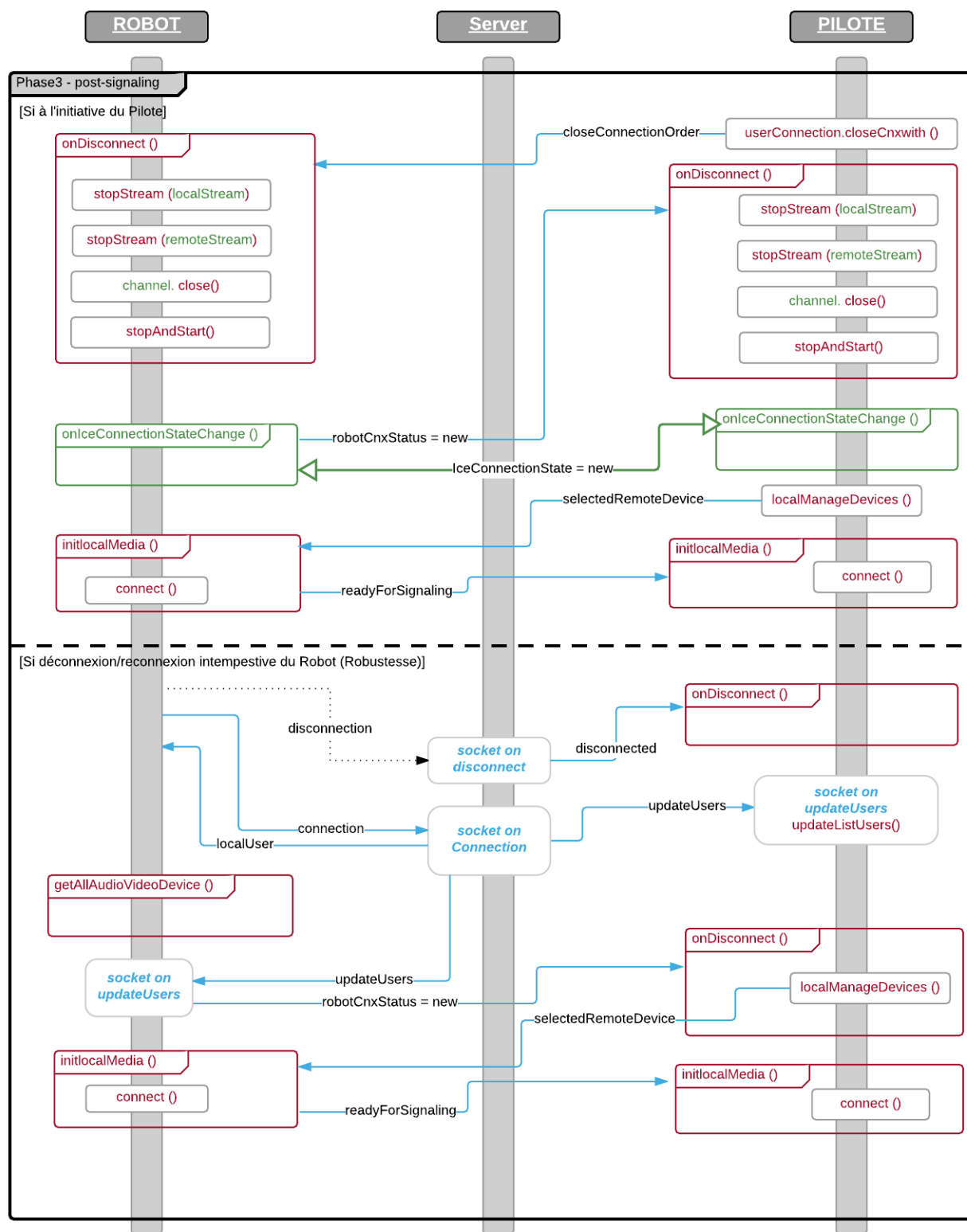
** Komcom est le module faisant interface entre l'application et les versions actuelles de la Robubox, Mobiserv et KomNav. C'est ici qu'il faudra intervenir pour interfacer l'existant avec les futures versions de KomNav.

L'algorithme procédural de l'application est divisé en 3 grandes phases.

1. **pre-signaling**: Initialisation et gestion de la connexion Websocket, sélection des paramètres d'exécution (caméras, définitions et affichages locaux et distants)
2. **signaling**: Initialisation de la connexion WebRTC. Ouverture des flux audio/vidéos et du canal de donnée (WebRTC data channel).
3. **post-Signaling**: Procédures de déconnexion & reconnexion WebRTC, manuelles et automatiques (robustesse).







Legende: Bleu = API websocket, Rouge = Application 1to1, vert = API WebRTC

B.c.c - Limitations

- Le poste pilote doit **OBLIGATOIREMENT** disposer d'une webcam. Cette WebCam ne doit pas être utilisée par une autre application.
- Le gamepad doit **IMPERATIVEMENT** être de type XBOX 360 (en USB filaire de préférence)
- Mêmes navigateurs (et mêmes versions) coté pilote et robot: Chrome V 49 minimum
- Pas de mise à jour en attente coté navigateurs
- Bug aléatoires non bloquants:
 - **L'image du pilote se fige coté robot alors que les connexions WebRTC et webSocket ne sont pas interrompue.** Il semblerait que le problème soit dû à une chute du framerate de l'image reçue. L'implémentation actuelle de WebRTC ne permet pas de détecter et d'intercepter ce type d'événement, pour, par exemple, enclencher un processus de reconnexion automatique (robustesse).
 - **les caméras distantes ne sont pas identifiées par un label** (mais restent sélectionnables). Le mieux est d'abord un reload de la page coté robot puis un reload de la page coté pilote.
 - **La connexion WebRTC s'établit mais l'une des caméras affiche un rectangle noir en local ainsi que sur le client distant distant.** Soit la caméra est déjà utilisée par une autre application, soit il s'agit d'un problème d'alimentation du port USB sur lequel est branché la caméra. Dans ce dernier cas de figure:
 1. Débrancher/rebrancher la caméra et relancer le navigateur pour réinitialiser sa détection.
 2. Si le problème persiste, s'assurer que le port USB sur lequel est montée la caméra est bien alimenté
- Bug aléatoires bloquants:
 - **L'image du robot reste vide et la connexion distante en WebRTC ne se fait pas.** C'est souvent un échec pendant la phase d'échange de candidates du signaling (message en console: "Error processing ICE candidate"). La meilleure des solutions reste le reload de la page pilote et la relance du processus. Cela peut être dû aussi à un filtrage fort du réseau qui ne laisse passer ni STUN ni TURN. Dans ce cas, le mieux est de passer par un VPN (coté robot et/ou pilote).
 - **Impossible de se connecter aux pages pilote ou robot:** Par mesure de sécurité, le contrôle d'accès redirige vers la page d'accueil tout utilisateur qui tente de se connecter dès lors qu'il y a déjà un utilisateur connecté à cette interface. Toutefois, suite à un problème de chargement ou d'exécution des scripts lors de la connexion ou de la déconnexion, il peut arriver qu'il reste un utilisateur « fantôme » dans la liste des utilisateurs connectés. Dans ce cas, est présent dans l'interface d'administration un bouton «Ejection de tous les clients» qui en plus d'envoyer un ordre d'éjection à tous les clients réellement connectés nettoie la liste des utilisateurs connectés, y compris les utilisateurs fantômes. Enfin, si rien ne fonctionne, on peut toujours relancer le serveur nodejs de l'application pour réinitialiser la session websocket.



B.c.d - Licence & versions

Copyright © CNRS (Laboratoire I3S) / université de Nice

Contributeurs: Thierry Bergeron & Michel Buffa, 2015-2016

Prototype d'application Web pour le Contrôle/Commande à distance d'un Robot par internet.

Cette application est régie par la licence CeCILL-C soumise au droit français et respectant les principes de diffusion des logiciels libres. Vous pouvez utiliser, modifier et/ou redistribuer ce programme sous les conditions de la licence CeCILL-C telle que diffusée par le CEA, le CNRS et l'INRIA sur le site "<http://www.cecill.info>".

- **Révision 1.5.0**
 - Implémentation d'un mode conférencier (Commandes au Gamepad + Plein écran)
 - Implémentation d'un mode tracking pour la géolocalisation du robot
- **Révision 1.5.1:**
 - Modification de l'architecture des fichiers de configuration
 - Implémentation d'un nouvel affichage de la vitesse et de la direction
 - Ajout d'un mode de conduite "Full Axe" sur un seul stick analogique
 - Simplification des procédures de configurations caméra en mode conférencier
 - Simplification des procédures de connexion/déconnexion en mode conférencier
 - Fix d'un bug aléatoire (liste des caméras distantes vide)
 - Modification de la gestion des résolutions de caméras.
 - Optimisation générale de l'ergonomie de l'IHM de commande
- **Révision 1.5.2:**
 - Abandon de la rétrocompatibilité avec la Robubox au profit de l'actuelle version de KomNav/Mobiserv (par protocole HTTP en attendant la version WebSocket...)
 - La compatibilité cartographique avec la dernière version de Pure.
 - L'implémentation de la navigation par points d'intérêts.
 - L'implémentation de la commande 'Stop' pour les trajectoires autonomes.
 - La détection de proximité des points d'intérêts pour le déclenchement d'évènements.

- **Révision 1.6.0**

- Implémentation d'une couche de persistance sous MongoDB (non exploitée pour l'instant):
- Implémentation d'une couche de web sémantique permettant:
 - Selon le POI le plus proche, le téléchargement de ressources complémentaires
 - L'affichage de ces ressources (Photos, vidéos, articles, données fournies par le musée)
- Implémentation des commandes de caméra IP de type Foscam
- Implémentation d'une interface d'administration permettant:
 - l'éjection ou le reload forcé du pilote et du robot
 - l'ajout, la sélection et la suppression à la volée de maps pour la cartographie
 - la bascule à la volée en mode de simulation robot "FakeRobubox"
 - l'ajout, la sélection et la suppression à la volée des IPs pour les caméras Foscam et les robots
 - La persistance de ces configurations
- Modification du Header des IHMs (Ajout d'un logo cliquable, affichage du login)
- Réorganisation de l'IHM Robot.

- **Révision 1.6.1**

- Cartographie : Refonte algorithme du mode tracking
- Interface d'administration : Activation/désactivation du Gamepad Physique
- Interface d'administration : Optimisation de l'éjection forcée des utilisateurs
- Système de recommandations (web sémantique) : Implémentation Menu de sélection type coverflow
- Système de recommandations (web sémantique) : Implémentation d'une lightBox pour l'affichage des ressources

- **Révision 1.6.2**

- Caméra Foscam: Implémentation commandes de Zoom & reset
- Caméra Foscam: Simplification du code source côté Gamepad
- Caméra Foscam: Factorisation du code dans une classe dédiée
- Caméra Foscam: Modification mapping Gamepad des commandes caméra
- Caméra Foscam: Implémentation d'une interface Web avec commandes en mode "pas à pas" et "continu"
- Caméra Foscam: Ajouts d'icônes et corrections de Bugs dans le bloc de commandes

- **Revision 2.0.0**
 - IHM Robot V2
 - Refonte & Simplification en blocs IHM Robot
 - Création modules (Map, localCam, remoteCam, Tchat)
 - Ajout Système Drag&Drop (souris)
 - IHM Pilote V2
 - Refonte & Simplification en blocs IHM Pilote
 - Création modules (Map, Drive, Navigation, Video Quality, localCam, remoteCam, Setings, Tchat)
 - Ajout Système Drag&Drop (souris)
 - Fiabilisation solution embarquée Kompaï Robotics
 - Abandon Mobiserv pour KomServer (V1 - version http)
 - Refonte Services & données KomServer > 1to1
 - Modification algorithmes d'affichages trajectoires
 - Modification des messages d'erreur
 - Modification du système FakeRobubox
 - Caméra au sol: Intégration ds module Drive IHM
- **Revision 2.0.1**
 - Caméra Pan / Tilt: Abandon système FOSCAM
 - IHM Pilote: Réorganisation modules
 - IHM Pilote: Fonction reset Display
 - Commandes Drive : Ajout commandes directionnelles clavier
- **Revision 2.0.2**
 - Interface d'administration: Changement carte par défaut du mode FakeRobubox
 - Interface d'administration: Réorganisation modules gestion IP robot et IP caméra
 - IHMs: Epuration CSS coté pilote
 - IHMs: Reprise ergonomique module cartographie
- **Revision 2.0.3**
 - Caméra Pan / Tilt: Intégration module commande caméra PTZ

- **Revision 2.0.4**
 - Refonte charte graphique interface d'administration
 - Refonte page d'accueil

- **Revision 2.0.5**
 - Interface d'administration: Refonte charte graphique
 - Interface d'administration: Gestion web sémantique
 - IHMs: Refonte page d'accueil
 - Web sémantique: C.R.U.D des scenes
 - Web sémantique: C.R.U.D des medias externes
 - Web sémantique: C.R.U.D des parcours de visite
 - Web sémantique: C.R.U.D des maps et des POI

- **Todo:**
 - Caméra au sol: Switch auto caméra (mode conduite/précision)
 - Caméra Pan/Tilt: Recentrage automatique en mode Drive
 - IHM robot: module navigation
 - IHM robot: module tchat
 - IHM robot: contrôle d'accès (si mode fakeRobubox désactivé)
 - IHM pilote: Centrage IHM
 - IHMs: Système touch Screen
 - Cartographie: Clic & Go POI
 - Cartographie: Mécanisme de superposition map originale/retravaillée
 - Cartographie: Upload automatique de la map
 - Web sémantique: Filtres (level, language, theme, keywords, scenes, poi, duration)
 - Web sémantique: C.R.U.D des objets exposés
 - Web sémantique: Mécanisme suggestion ressources via mots clefs
 - Dailymotion, Youtube
 - DiscoveryHub, DbPedia, Quackis
 - Navigation: Implémentation mode parcours automatique chronométré