

Documentation Technique

Informations

| | |
|-------------------------|--|
| Nom du projet | Protocole POBEC |
| Type de document | Documentation technique |
| Date | 21/01/2022 |
| Version | 1.0 |
| Auteurs | PORTIER Loane DEMAN Y Theo NICOLAS Nathan ROBE Titouan VAUGRENTE Lucas MABILEAU Clément MAINCENT Oscar |

Table des matières

| | |
|---|---|
| Table des matières | 1 |
| 1 – Résumé du document | 2 |
| 2 – Rappel sur le fonctionnement de l'application | 2 |
| 2.1 – Description du simulateur | 2 |
| 2.2 – Décomposition du projet | 2 |
| 2.3 – Architecture globale | 3 |
| 3 – Simulateur | 3 |
| 3.1 – Architecture | 4 |
| 3.2 – Technologies utilisées | 5 |
| 4 – Bugs connus | 6 |
| 4.1 – Fonctionnalité | 7 |

I - Résumé du document

Ce document est la documentation technique officielle du simulateur de livraison du site [Alizon.bzh](https://alizon.bzh), entreprise COBREC.

Il est divisé en 2 parties :

- Description général du Simulateur
- Description détaillé du protocole POBEC

2 - Rappel sur le fonctionnement de l'application

2.1 - Description du simulateur

Le protocole POBEC (Protocole Open source Breton Express Commande) à pour but de simuler des livraisons de commandes en temps réel effectuées sur [Alizon.bzh](https://alizon.bzh).

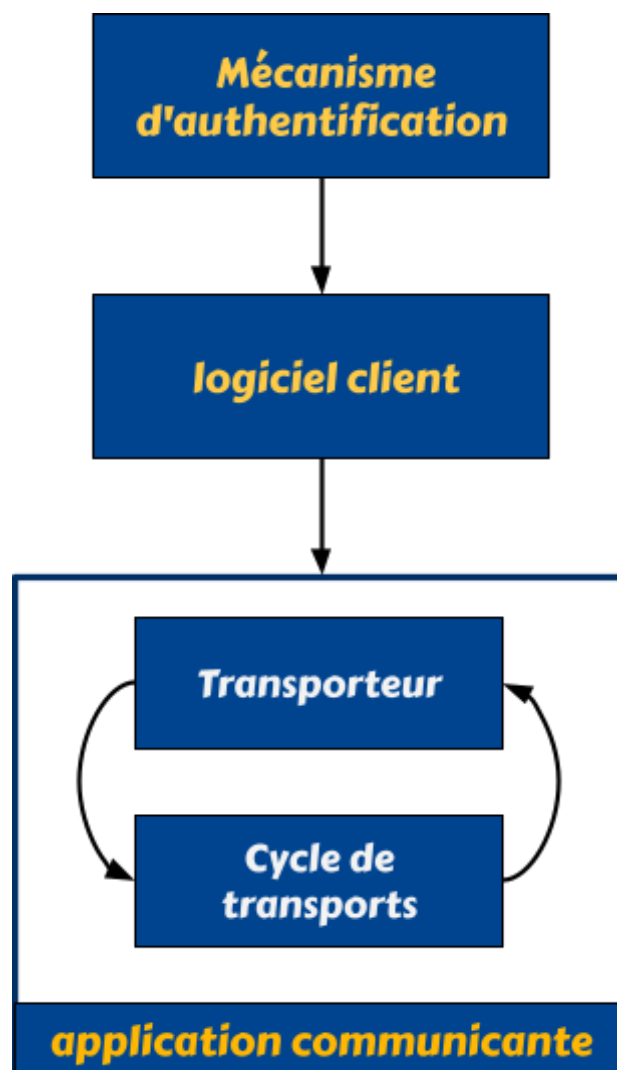
Il permet, au client connecté, de récupérer d'ajouter de nouvelles commandes, de suivre leur avancement. Il est également possible de récupérer les commandes terminées et archivées.

2.2 - Décomposition du projet

Notre projet se décompose en différentes parties :

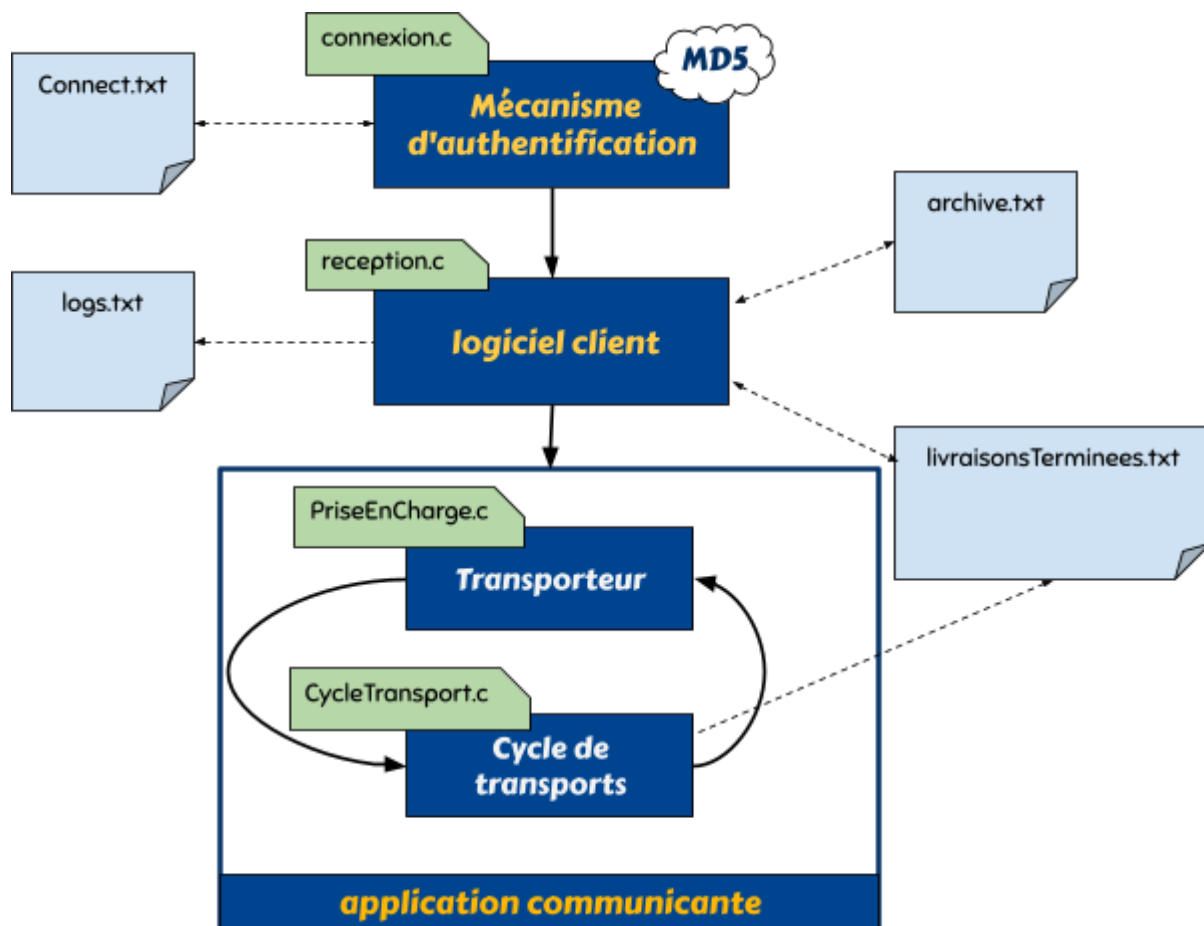
- Le Mécanisme d'authentification, permettant de se connecter en toute sécurité sur le simulateur.
- Le logiciel client, sous la forme d'une invite de commande permettant la gestion et le suivi direct des livraisons;
- L'application communicante détaillant le transporteur et le cycle de transport de chaque livraisons

2.3 - Architecture globale



3 - Simulateur

3.1 - Architecture



L'application démarre en commençant par le mécanisme d'authentification. Les logins et mots de passe sont stockés dans un fichier texte 'Connect.txt'. L'utilisateur peut alors se connecter en rentrant ses identifiants, vérifiés par la suite par le système. Par sécurité, les mots de passe enregistrés sont hachés en MD5 (Message Digest 5).

Une fois la connexion réussie, le logiciel client se lance, il est alors possible d'effectuer les requêtes suivantes :

- Ajouter une nouvelle livraison
- Consulter les commandes terminées
- Consulter les commandes archivées

(Pour savoir comment utiliser le Simulateur, [veuillez consulter la documentation Client](#))

Les nouvelles livraisons passent tout d'abord par le transporteur afin de vérifier sa capacité actuelle (le transporteur possède une capacité maximum de livraisons en cours défini par le client).

Il prévient le client de la prise en charge de sa commande et l'envoie par la suite dans 'Cycle de transport' : Un programme autonome effectuant les différentes étapes de livraisons.

Il commence par envoyer un signal de confirmation au client ("Commande envoyé"), puis informe le client de chaque étape de livraisons :

1. Prise en charge de la commande
2. Transport vers la plateforme régionale
3. Transport entre la plateforme et le site local de livraison
4. En cours de livraison chez le destinataire
5. Accusé de réception de clôture de cycle

Une fois la livraison clôturée, le programme entre le numéro de commande, le destinataire et la date de livraison dans le fichier texte 'livraisonsTerminees.txt'.

La livraisons étant autonome, le client peut pendant ce temps effectuer d'autres requêtes, comme consulter les commandes terminées listé dans le fichier texte précédant. Il à la possibilité d'archiver toutes ces commandes dans le fichier 'archive.txt'.

Toutes les requêtes effectuées sont enregistrées en temps réel dans le fichier 'log.txt'.

3.2 - Technologies utilisées

Le logiciel client est codé en C sous Linux et utilise beaucoup de fonctionnalité système notamment les suivantes :

- **Les Sémaphores :**
Un sémaphore est une variable partagée par différents « acteurs », qui garantit que ceux-ci ne peuvent y accéder que de façon séquentielle (à la suite), et constitue la méthode utilisée couramment pour restreindre l'accès à des ressources partagées (par exemple un espace de stockage) et synchroniser les processus entre eux dans un même environnement.

Pour notre simulateur, les sémaphores sont utilisés à chaque lecture ou écriture sur les fichiers textes afin d'éviter tout problème de suppression de données.

- **Les tubes :**
Un tube (ou pipeline) est un mécanisme de communication inter-processus.

Ils nous permettent de partager des données entre les programmes sans les stocker inutilement dans un fichier. Après avoir reçu le signal, le programme "se met en

écoute” sur un tube, il va simplement lire les données passées à l'intérieur pendant qu'un autre programme écrit dedans.

- **Les signaux :**

Les signaux sont des messages envoyés par le système d'exploitation à notre simulateur en cours d'exécution pour lui indiquer qu'un événement spécifique s'est produit. Ces événements peuvent inclure des choses comme une erreur de segmentation, un Ctrl-C tapé par l'utilisateur pour interrompre le programme, ou encore un signal de fermeture de programme demandé par le système d'exploitation..

Important tout le long du processus, les signaux permettent de partager l'avancement d'une livraison notamment sa prise en charge et sa phase de clôture. Dès que le transporteur reçoit un signal, il est immédiatement reconnu puis traité, par exemple pour libérer une place si une livraison s'est terminée. Les signaux utilisés sont les suivants :

- **SIGUSR1**
Envoyé depuis 'cycle de transport', informe le transporteur du début de la livraison
- **SIGUSR2**
Envoyé depuis 'cycle de transport', informe le transporteur la fin de la livraison
Envoyé depuis le transporteur, informe le client de son bon lancement
- **SIGALRM**
Envoyé depuis le client, informe le transporteur de se mettre en écoute sur un tube pour une nouvelle livraisons

- **Telnet :**

Telnet (terminal network ou télécommunication network, ou encore teletype network) est un protocole utilisé sur tout réseau TCP/IP, permettant de communiquer avec un serveur distant en échangeant des lignes de texte et en recevant des réponses également sous forme de texte.

Primordiale pour le simulateur, il garantit l'interaction du client avec le simulateur. Les requêtes du client passent par le port 8080, ce qui permettrait plus tard de relier le site web au simulateur.

- **Le multitâche :**

Le multitâche permet à notre programme d'effectuer plusieurs tâches en même temps. Cela signifie que plusieurs programmes peuvent être mis en cours d'exécution simultanément. Avec le multitâche on va créer des threads, et ces threads vont exécuter des tâches simultanément comme lire des données à partir d'un fichier pendant que le programme principale traite d'autres tâches .

Le multitâche pour le simulateur est essentiel . C'est avec la fonction système 'fork()' en langage C, que le transporteur est capable de lancer plusieurs livraisons en même temps. Celles-ci sont alors toutes indépendantes, relier néanmoins avec le simulateur grâce aux signaux vu précédemment.

4 - Bugs connus

4.1 - Fonctionnalité

1. L'ajout de commande est pour l'instant, sans effet, entraînant le suivi de commande et l'ajout dans la liste des livraisons terminée dysfonctionnel.