

Projet technologique Autopilote

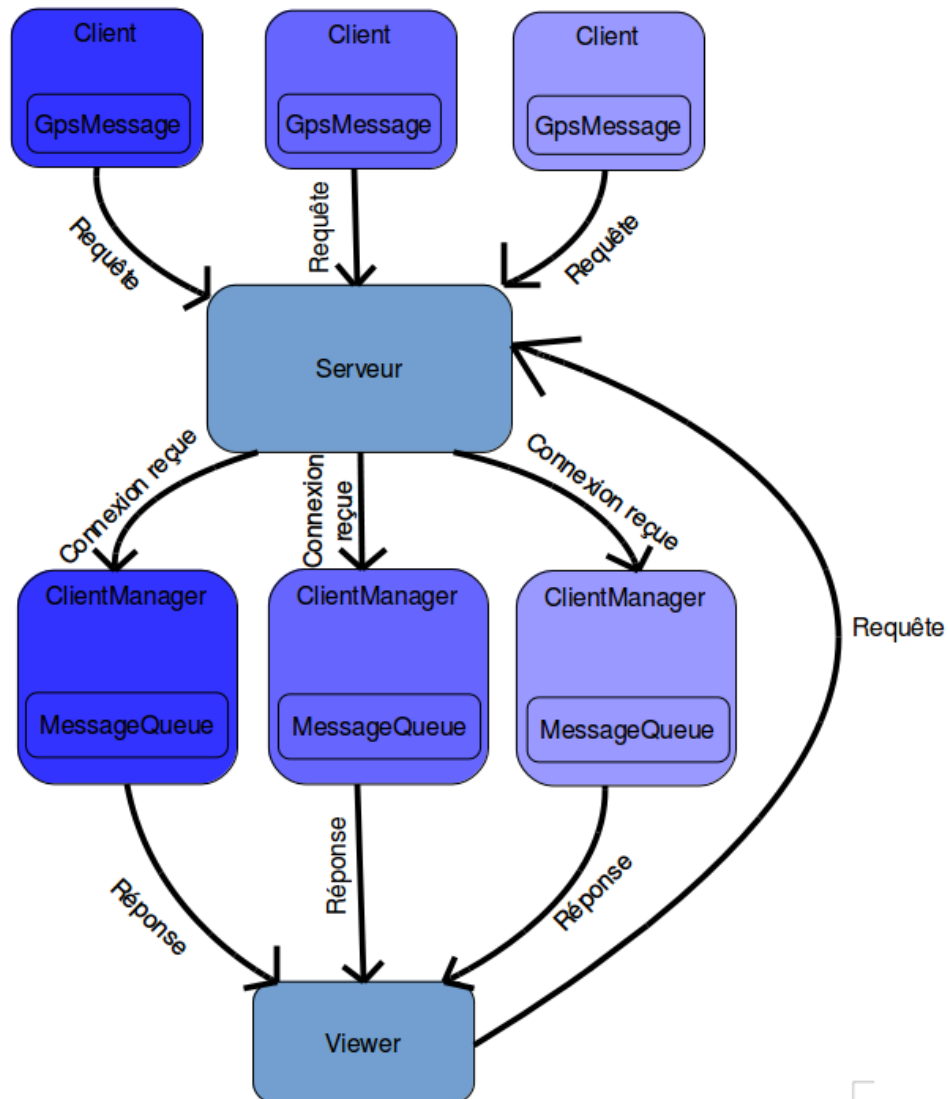
Robin Hardy
Paul José-Vedrenne

Introduction

- Conception d'un bus logiciel qui communique avec des capteurs
- Programmation en Java
- Programmation réseau
- Utilisation du Json

Développements effectués

Schéma de l'architecture globale du projet



Développements effectués

- Requêtes :
 - Côté capteur :
 - Register
 - Deregister
 - send
 - Côté observateur :
 - List
 - Get (par id)
 - get_last

Développements effectués

- Intégration d'un capteur Arduino
 - Fabrication du capteur
 - Fonctionnement
 - Intégration au bus
 - Bibliothèque RXTXComm.jar
 - Envoi des données sur le port USB
 - Classe Java qui lit les données du port USB et les place dans une instance de la classe Client

Développements effectués

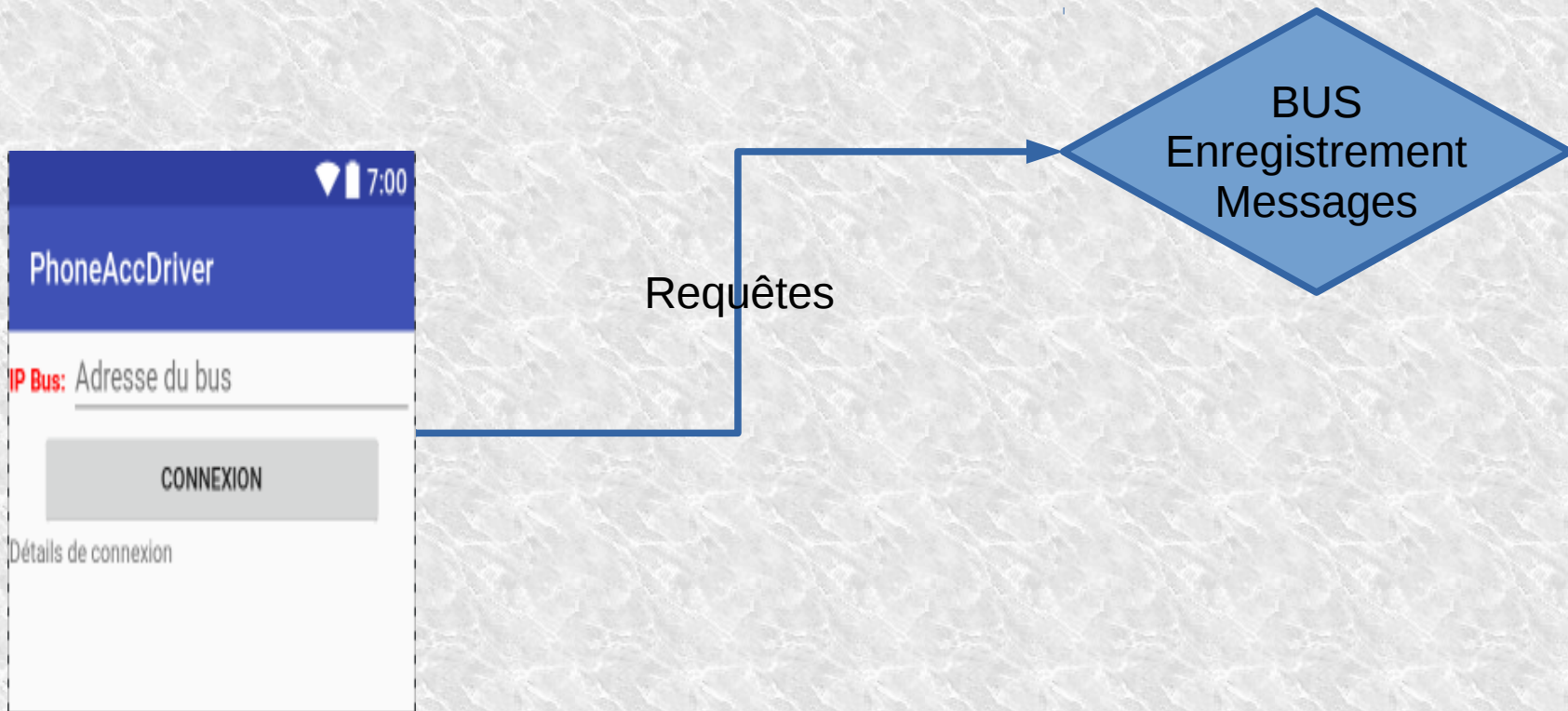
- Plusieurs architectures utilisés :
 - Capteur simulé (Logiciel)
 - Capteur Android (Smart Phone)
 - Capteur Arduino (« Interfacage » USB)
- Trois capteurs :
 - Gyroscope (Arduino, non-implementé)
 - GPS (Simulé, données aléatoire)
 - Accéléromètre (Android, récupération des « sensors »)

Développements effectués

Driver Accéléromètre (Smart Phone):

- Création d'une application Android
 - Activity + GUI
- Accès au données de l'accéléromètre
- Envoie des données sur le bus :
 - Connexion au bus
 - Utilisation de l'Api (classe Client)

Développements effectués



Organisation du travail

- Diagramme de Gantt

Février		Mars		Avril	
Conception des spécifications		Tests et débuggages			
	Json	Implémentation requêtes			
			SimGps	IHM	Capteur mobile

Organisation du travail

- Difficultés rencontrées
 - Programmer en réseau
 - Conception des spécifications
 - Apprendre le Json

Conclusion

- Gestion de projet :
 - Conception → nouveau
 - Développement → nouvelles techniques