

# Rapport de projet de Systèmes Concurrents

Damien Hostettler et Vicky Dincher

25 Janvier 2016

# Table des matières

1    Architecture de l'application . . . . . 2

# Introduction

Ce projet consiste à gérer la concurrence entre plusieurs processus dans le cadre d'objets partagés sur un serveur. Il s'agit de synchroniser la lecture et l'écriture de ces objets afin que chaque client ayant effectué une action ait la dernière version de l'objet.

## 1 Architecture de l'application

L'application implémentée comporte quatre classes :

- La classe **ServerObject**, qui est la dernière version d'un objet partagé. Il peut être en *lock\_read*, si plusieurs personnes lisent dans cet objet, en *lock\_write*, si une personne est en train d'écrire, ou en *no\_lock*, si personne ne le possède. Les deux positions *lr* et *lw* sont incompatibles. Lors de chaque changement (lorsque l'objet passe en lecture ou en écriture), l'objet doit automatiquement repasser par le serveur.
- La classe **Server** : Cette classe permet d'établir le serveur qui contient tous les objets. Il contient un tableau de **ServerObject**, qui possèdent chacun un statut propre à eux-même.
- La classe **SharedObject**, qui est la copie de l'objet, présente chez le client. Elle possède plusieurs status, qui sont ceux décrits dans le sujet. Chaque *SharedObject* est propre à un *Client*. Le statut de l'objet, indique donc l'état du client.
- La classe **Client**, qui représentera chaque client pouvant lire ou écrire.