

# Architectures Distribué - Java RMI

GARCIA PENA Loris

8 octobre 2023

## Résumé

Ce rapport présente notre travail réalisé dans le cadre du TP1 portant sur Java RMI (Remote Method Invocation). Le TP1 avait pour principaux objectifs :

- Comprendre la notion d'objet distribué, qui permet à des objets Java de communiquer à travers un réseau.
- Utiliser le passage de stub ou d'objet sérialisé, des mécanismes essentiels pour la communication entre objets distants.
- Explorer le téléchargement de code via l'utilisation de la codebase, une fonctionnalité importante dans le déploiement d'applications RMI.

## Table des matières

<b>1</b>	<b>Une première version simple</b>	<b>2</b>
1.1	Introduction . . . . .	2
<b>2</b>	<b>Classe Cabinet Vétérinaire</b>	<b>3</b>
<b>3</b>	<b>Création de Patient</b>	<b>4</b>
<b>4</b>	<b>Téléchargement de Code</b>	<b>5</b>
<b>5</b>	<b>Alerte</b>	<b>6</b>

# 1 Une première version simple

## 1.1 Introduction

La technologie Java Remote Method Invocation (RMI) est un système puissant qui permet à un objet s'exécutant dans une machine virtuelle Java d'invoquer des méthodes sur un objet s'exécutant dans une autre machine virtuelle Java distante. Cette capacité de communication à distance entre des programmes repose sur l'invocation de méthodes sur des objets distribués appelés stub. En d'autres termes, elle offre la possibilité d'interagir avec un objet distant comme s'il était local. Cela favorise la construction d'applications réparties en utilisant des appels de méthode au lieu d'appels de procédure, simplifiant ainsi le développement d'applications distribuées.

Dans le cadre de ce TP, nous nous plaçons dans le contexte d'un cabinet vétérinaire. Chaque patient du cabinet, c'est-à-dire chaque animal, possède une fiche individuelle avec un dossier de suivi médical. L'objectif est de créer un système où chaque vétérinaire du cabinet peut accéder aux fiches des patients à distance. Nous mettrons en place un serveur et développerons un client pour les vétérinaires.

Au cours de cette première étape de notre projet, nous allons établir les bases en créant une version simple de notre application, en développant une classe "Animal" distribuée, et en configurant un serveur et un client pour la communication. Nous débuterons en utilisant un environnement local sans gestion de la sécurité, afin de nous familiariser avec les principes fondamentaux de Java RMI. Par la suite, nous enrichirons notre application en ajoutant des fonctionnalités avancées pour répondre aux besoins d'un cabinet vétérinaire moderne.

## **2 Classe Cabinet Vétérinaire**

### 3 Création de Patient

## 4 Téléchargement de Code

## 5 Alerte