



**RÉPUBLIQUE TUNISIENNE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ TUNIS EL MANAR
FACULTÉ DES SCIENCES DE TUNIS
DÉPARTEMENT DES SCIENCES DE L'INFORMATIQUE**

MINI PROJET SUJET

**Implémentation de serveurs et clients en utilisant Java
RMI, gRPC et les sockets**

Elaboré par :

Attia Ranim

Bel Hadj Said Mahdi

Année Universitaire 2023 - 2024

I. Introduction générale

Commençons par une introduction générale à chacune de ces technologies, puis ensuite nous pourrions procéder à la comparaison des trois technologies.

A. Java RMI (Remote Method Invocation) :

- ❖ Java RMI permet à un objet Java d'appeler des méthodes sur un objet distant, ce qui permet une communication entre processus sur différentes machines virtuelles Java.
- ❖ Les objets distants doivent être des instances de classes qui implémentent l'interface Remote.
- ❖ Java RMI fournit des mécanismes pour le registre des objets distants, la résolution des noms et la sérialisation des objets.

B. gRPC :

- ❖ gRPC est un framework RPC (Remote Procedure Call) open source développé par Google.
- ❖ Il utilise le protocole HTTP/2 pour la communication, ce qui permet des appels de procédure à distance plus efficaces.
- ❖ Il prend en charge plusieurs langages de programmation et offre une facilité d'utilisation grâce à la génération automatique de code.

C. Sockets :

- ❖ Les sockets sont une API de communication utilisée pour établir des connexions entre des applications sur un réseau.
- ❖ Ils fournissent des interfaces de programmation pour créer des connexions TCP ou UDP.
- ❖ Avec les sockets, tu as un contrôle total sur le flux de données, ce qui permet une grande flexibilité mais demande également plus de travail de programmation.

II. Rapport Comparatif

A. Java RMI

Facilité de mise en œuvre : Relativement simple à mettre en œuvre pour les développeurs Java, mais peut nécessiter une configuration complexe du registre RMI et des politiques de sécurité.

Performances : Bonnes performances pour les charges de travail légères à modérées, mais peut être limité pour les applications à forte demande en raison de l'overhead de la sérialisation des objets.

Flexibilité : Bien intégré à Java, mais limité aux applications Java et peut être moins flexible pour les environnements hétérogènes ou nécessitant une interopérabilité avec d'autres langages de programmation.

B. gRPC

Facilité de mise en œuvre : Facile à mettre en œuvre grâce à la génération automatique de code à partir de définitions de service protobuf, mais peut nécessiter une configuration initiale plus complexe que Java RMI.

Performances : Offre généralement de bonnes performances grâce à l'utilisation du protocole HTTP/2, adapté aux applications nécessitant une communication rapide et efficace.

Flexibilité : Prend en charge plusieurs langages de programmation et offre une flexibilité accrue grâce à la définition des services dans un fichier protobuf, mais optimisé pour les environnements conteneurisés.

C. Sockets

Facilité de mise en œuvre : Offre une flexibilité maximale mais nécessite un travail de bas niveau pour la mise en œuvre et la gestion manuelle des détails du protocole.

Performances : Les performances dépendent largement de la façon dont elles sont mises en œuvre et configurées, offrent une flexibilité maximale pour les applications nécessitant une personnalisation poussée du protocole

Flexibilité : Offre une flexibilité maximale en permettant un contrôle total sur la communication réseau et peut être utilisé avec n'importe quel langage de programmation prenant en charge les opérations de lecture/écriture de fichiers.

III. Conclusion

En conclusion, le choix de la technologie dépendra des besoins spécifiques du projet en termes d'interopérabilité, de performance et de facilité de mise en œuvre.

En effet, Java RMI est bien adapté aux applications Java natives, gRPC offre une meilleure flexibilité multi-langage et des performances élevées, tandis que les sockets offrent une flexibilité maximale pour les applications nécessitant un contrôle total sur la communication réseau.