

Architecture générique d'un serveur gRPC/REST en Golang

Étudiant : David Crittin
Professeur : Jean-Luc Beuchat

Résumé

1. Prise en main du langage Go et de gRPC/REST
2. Réalisation de middlewares
3. Conception d'un service de chiffrement & intégration dans une image Docker

Introduction

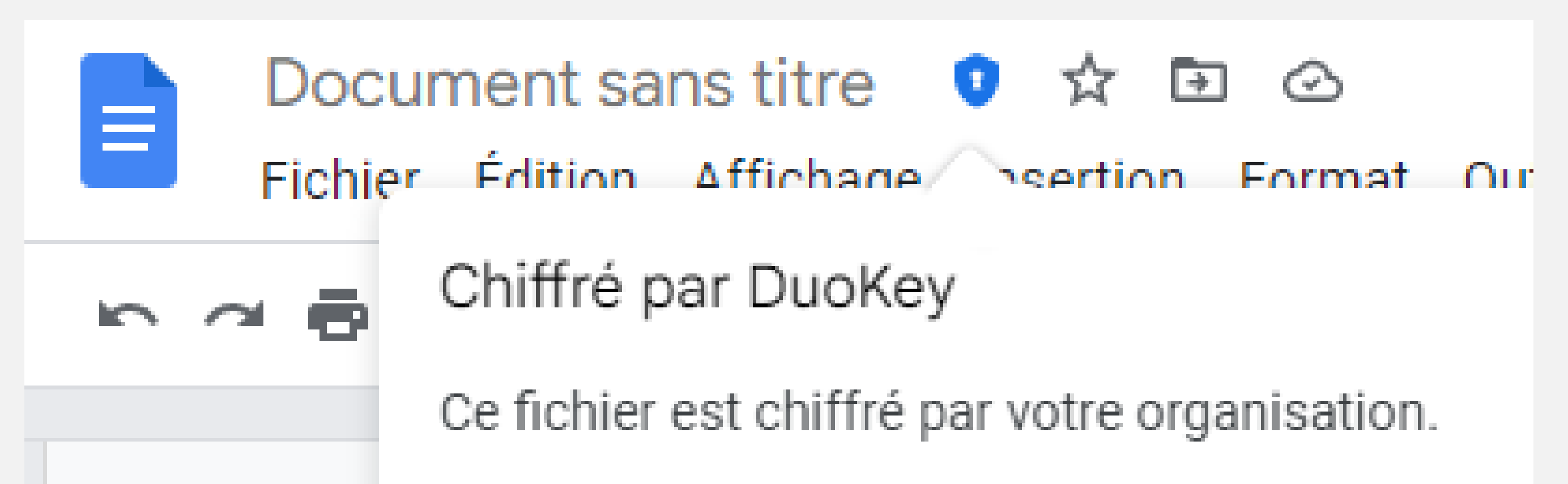
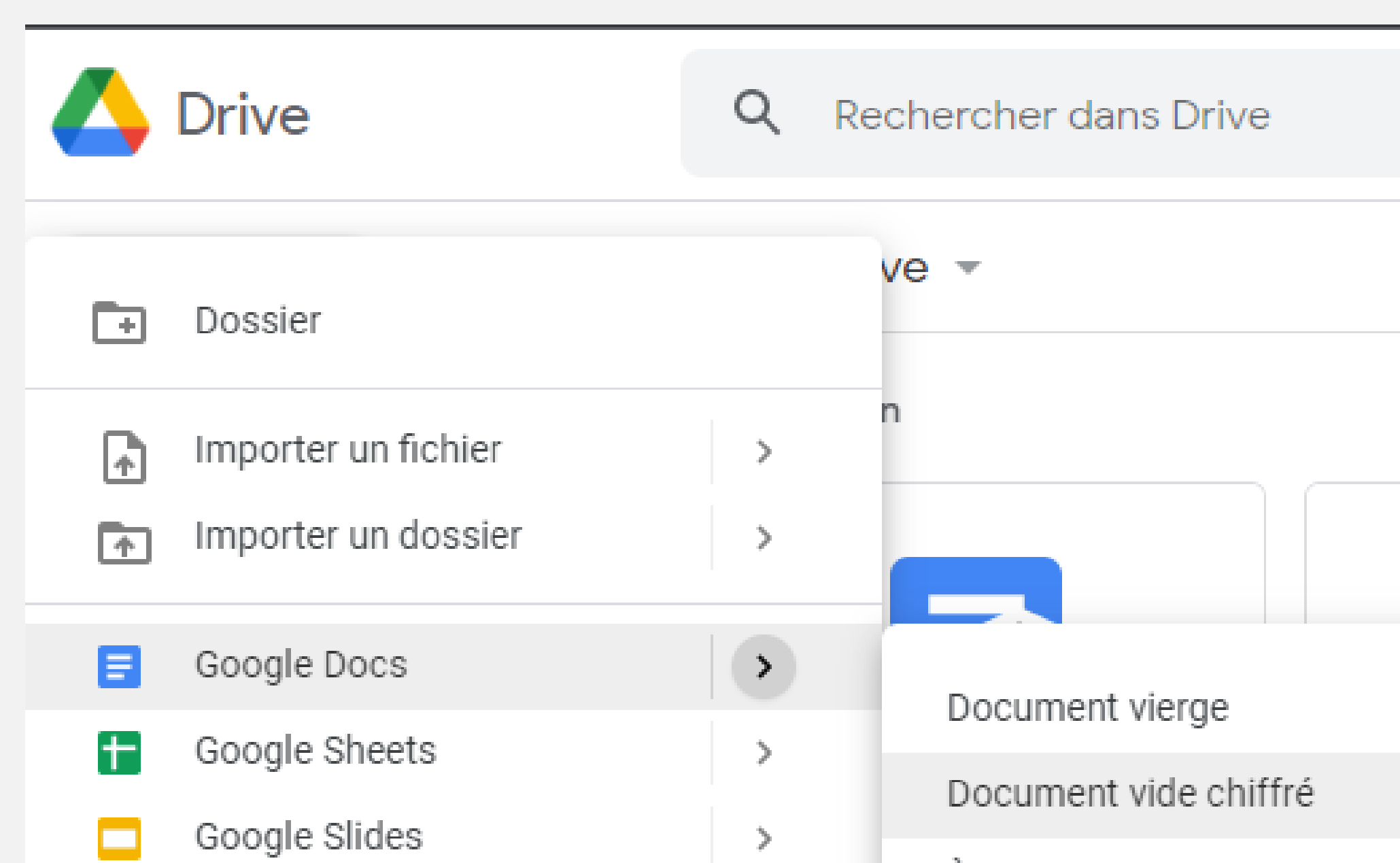
- L'**Applied Cryptography Laboratory** a exprimé le besoin d'avoir un serveur générique **gRPC/REST** développé en **Go**. Ce serveur devait mettre à disposition une bibliothèque de *middlewares* tels que le chiffrement de données, l'authentification, la gestion des sessions ou le *throttling*.
- Une fois ce serveur développé, celui-ci a été utilisé pour intégrer une API de chiffrement externe appelée **Google Client-side Encryption**. Les clés utilisées pour le chiffrement sont contrôlées par l'utilisateur et ne sortent pas de Suisse. La solution a été testée dans un *proof of concept* avec la société DuoKey.
- L'ensemble a ensuite été intégré dans une image **Docker**.

Technologies



Résultats

- Serveur gRPC/REST avec plusieurs *middlewares*
- **Google Client-side Encryption** fonctionnel



Conclusions

- Le serveur gRPC/REST dispose de plusieurs *middlewares* fonctionnels et peut être réutilisé comme base dans différents projets.
- La solution proposée est appliquée dans le cadre de l'utilisation de **Google Client-side Encryption**.
- Le serveur est disponible dans une image **Docker** minimale.