

Order Process

Maxime VILLAREJO

Travail de bachelor 2025

Informatique et systèmes de communication – Informatique logicielle

Professeur : Marcelo PASIN

Expert : Maël PEDRETTI

Description

Le projet consiste à moderniser le processus de commande de ressources cloud chez Swissmedic. Jusqu'alors, les demandes passaient par des fichiers YAML et des pull requests GitHub, un mode de travail manuel, lent et peu transparent. La solution mise en place introduit une plateforme de self-service permettant de saisir les demandes via un portail web. Celles-ci sont stockées en base de données, transformées automatiquement en fichiers YAML et intégrées aux dépôts d'infrastructure. Les déploiements sont ensuite orchestrés par GitHub Actions et Terraform. L'architecture repose sur un frontend React, deux backends Spring Boot, une base Azure SQL et une Azure Function, offrant modularité et traçabilité.

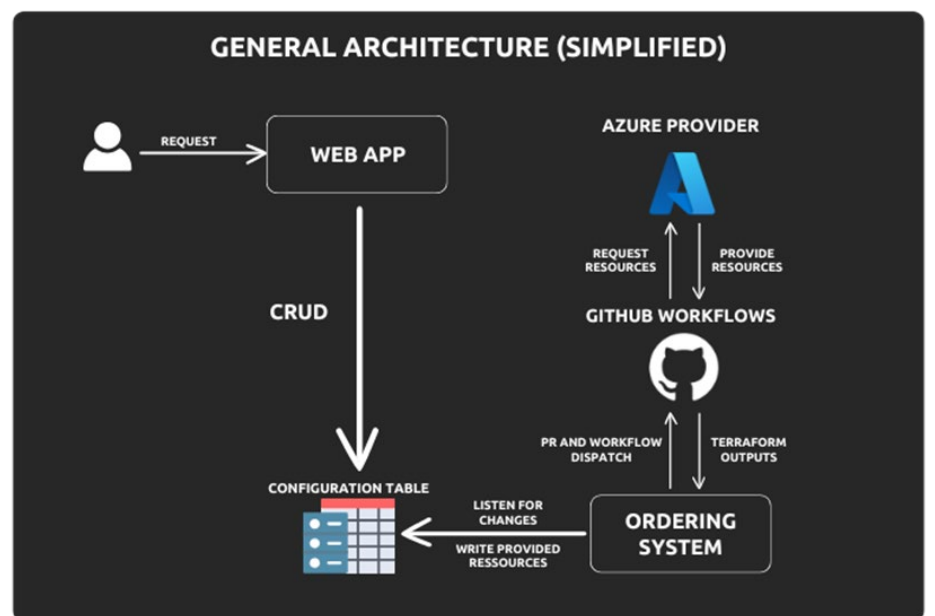
Résultats

Le résultat est un système opérationnel, privé au réseau Swissmedic : saisie guidée d'une demande, conversion automatique en YAML, création d'une branche et d'une PR, exécution des workflows Terraform, et retour d'état visible dans le portail. Le cas «Service Principal» est en production interne ; l'ajout d'autres ressources est prévu sans refonte grâce à l'architecture configurable et modulaire.

Le schéma ci-dessous illustre le fonctionnement global de la plateforme : les demandes passent par l'application web, sont stockées et transmises au système de commande, puis déployées automatiquement grâce aux workflows GitHub et Terraform, avec interaction directe avec le fournisseur cloud, Azure.

Déroulement

Le projet a démarré par l'analyse des besoins (autonomie, traçabilité, sécurité) et l'étude du flux existant. Des «proofs of concept» ont ensuite validé les choix clés (stockage, déclenchement d'événements, génération de formulaires). La conception a visé une architecture adaptée au contexte Swissmedic : composants simples, séparés et intégrables. L'implémentation a suivi (front React, backends Spring Boot, base SQL, automatisation GitHub/Terraform), puis le déploiement via Terraform et Helm, et enfin la rédaction du rapport et des manuels.



High level project architecture

Discussion : conclusions et perspectives

La plateforme apporte plus d'autonomie et de transparence en automatisant les demandes d'infrastructure, tout en s'intégrant aux outils existants pour garantir des déploiements reproductibles. Certaines limites subsistent, comme l'absence d'authentification par utilisateur, des retours Terraform encore peu exploitables et une couverture de tests incomplète. Les prochaines évolutions visent l'intégration Entra ID avec gestion des rôles, l'extension à de nouvelles ressources, une meilleure restitution des workflows et la génération automatique de formulaires, afin de renforcer la fiabilité et l'évolutivité de la solution.