CAPCOD Case Study : IA appliquée au support client

AO Conseil de l'Europe - Revue stratégique de la présence web

1. Identification et contexte

Nom du projet / de l'outil : Canival Medical RAG-based Knowlege System

Équipe utilisatrice : Canival Medical Scientific and Engineering teams
https://canivalmedical.com/

Date de démarrage du projet : Mai 2026

Statut actuel (pilote interne / test / production) : Pilote interne, phase 2 en cours de design

Date de mise en production (réelle ou prévue) : Septembre 2026

2. L'enjeu : le problème résolu

Description du problème initial (ex. temps de triage long, contexte dispersé entre historique et échanges)

L'équipe CANIVAL MEDICAL conçoit une valve cardiaque artificielle permettant de soigner les chiens de façon non invasive. En phase de R&D, les challenges sont nombreux et les équipes scientifiques collectent énormément de données d'ingénierie mais aussi liées aux procédures médicales. Toutes les interventions sont documentées, toutes les procédures donnent lieu à un protocole qui produit une documentation importante. On y retrouve : des vidéos d'opérations chirurgicales, des rapports d'intervention, des documents d'ingénierie produit, comptes rendus de réunions, des transcriptions de vidéos, des documents marketing, des fiches produit, des descriptions de protocoles, des retours d'expérience, etc.

Les contributeurs de cette base de connaissance sont répartis partout sur la planète. Cela signifie que l'information est centralisée mais que personne n'a une vue globale de ce qui s'y trouve.

Le volume d'information est considérable, un téraoctet pour la partie produit et plusieurs Téraoctets pour la partie scientifique. Cette quantité d'information est une mine d'or pour l'équipe mais ce volume la rend difficilement exploitable, surtout manuellement avec des recherches par mots clés.

Le client souhaite pouvoir poser des questions sur ces documents comme avec un système IA grand public. Mais en complément, il souhaite aussi pouvoir interroger la base depuis le logiciel Asana qui sert à la gestion de projet. Cela permet de compléter automatiquement les tâches avec des informations techniques du système sans avoir à quitter son environnement de travail.

Dans la conception de la solution il a été impératif de tenir compte des droits d'accès des différents utilisateurs sur les fichiers de la base de connaissance. En effet, il ne faut pas que le système de RAG puisse être utilisé pour consulter ou exfiltrer des données non autorisées. Cela a été implémenté dans la couche de sécurité de l'API.

3. La démarche et les technologies

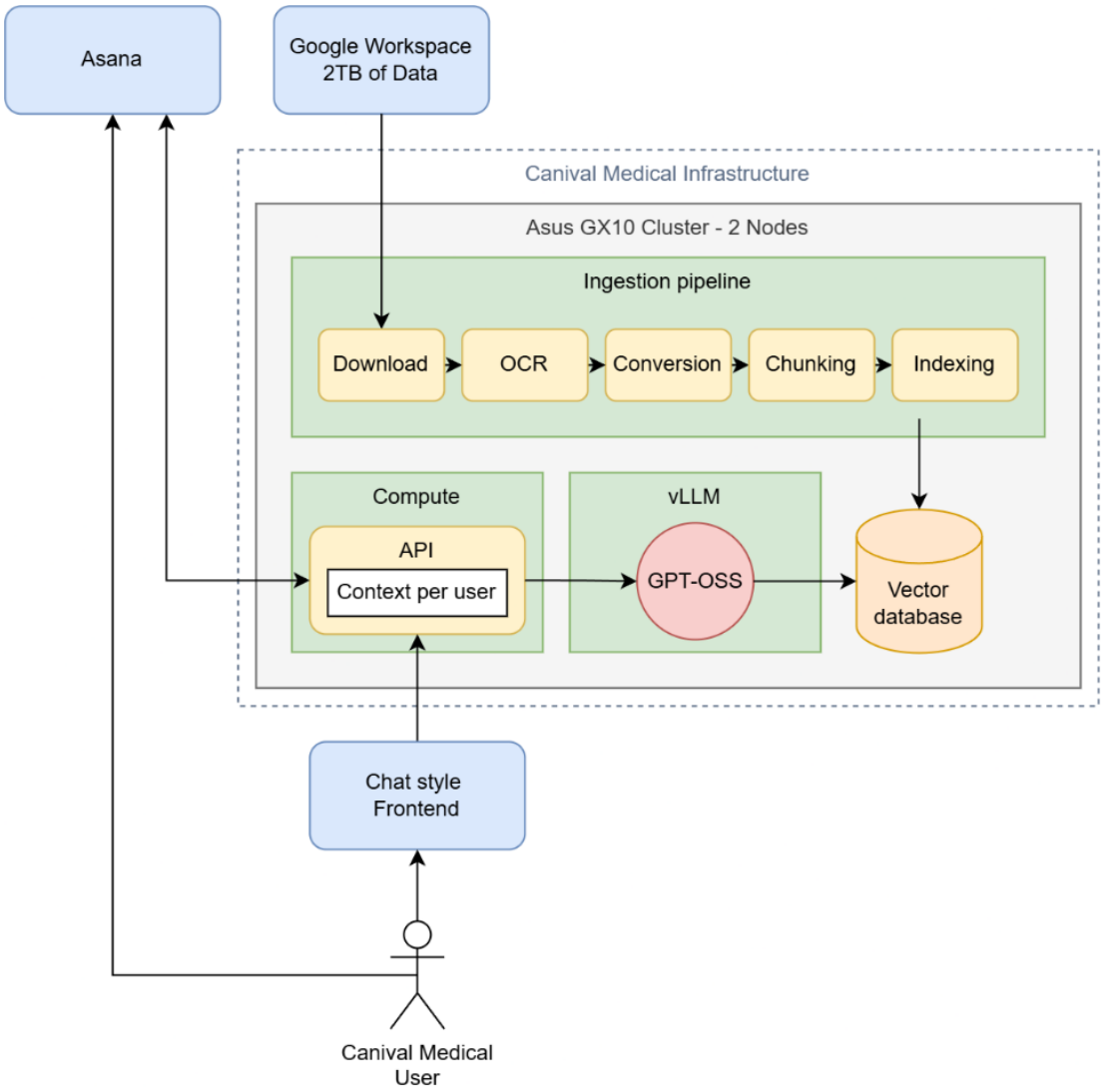
Technologie IA utilisée :

LLM Open Source (GPT-OSS) hébergé localement sur cluster de deux nœuds Asus GX10.

La première partie du projet (la phase 1) consiste à mettre en place un pipeline d'indexation pour stocker les documents dans une Vector database utilisable par un LLM. Par-dessus cette base, une API a été implémentée pour piloter les échanges entre l'utilisateur et le LLM ainsi qu'entre Asana et le LLM. Cette API est également en charge de la sécurité.

Plusieurs modèles ont été testés, tous Open Source et hébergées en local pour garantir que les données, très confidentielles, ne sortent pas du réseau du client. Ont été testées : Gemma4 (différentes variantes), GPT-OSS, Llama3, Mistral.

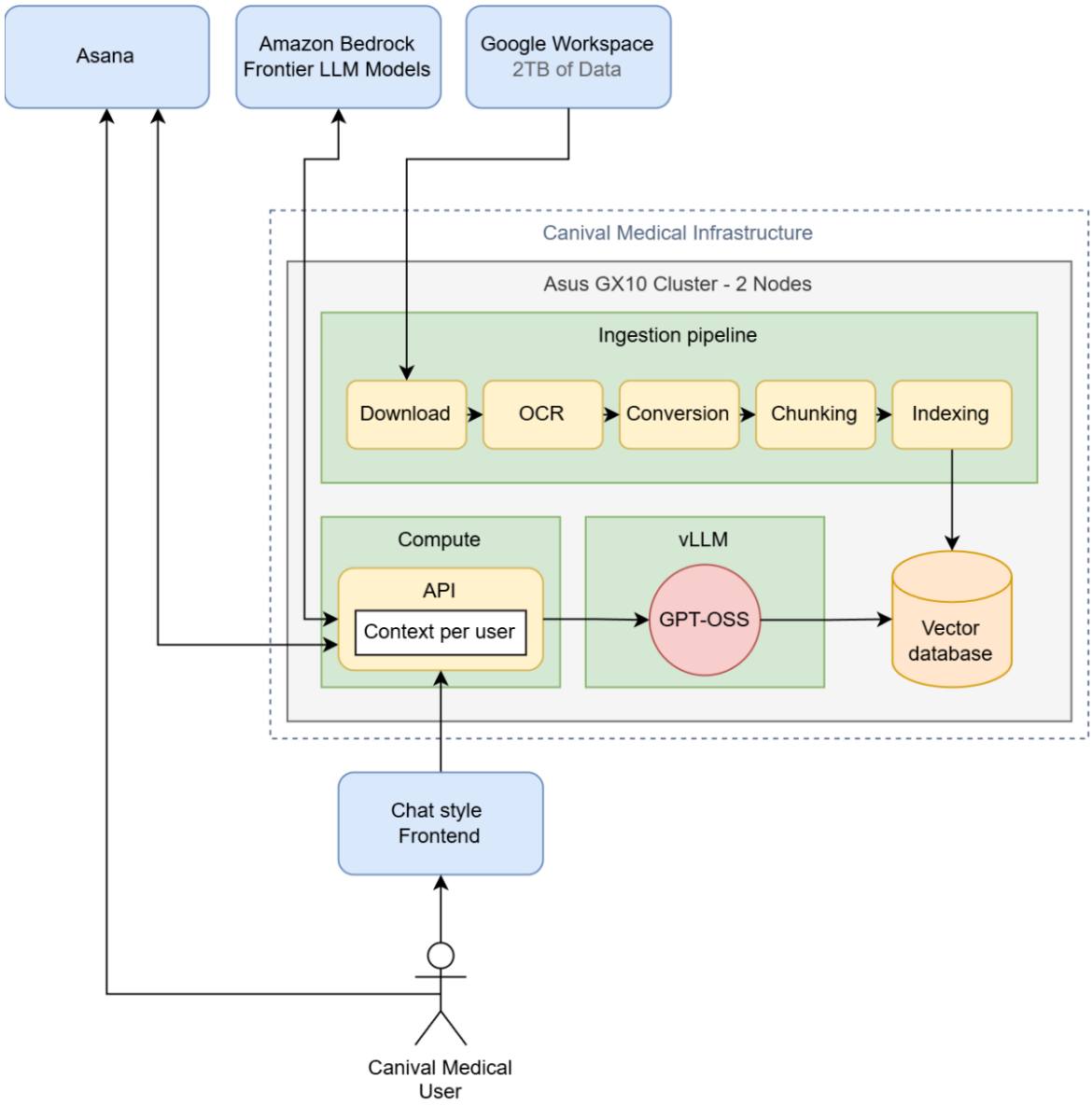
Architecture phase 1 :



Ce fonctionnement a fait ses preuves et le client souhaite aller plus loin en étendant le fonctionnement du système aux LLM « frontière » proposés par les fournisseurs OpenAI et Anthropic. Pour que les données restent privées, ces modèles sont utilisés au travers de la plateforme Bedrock d'AWS qui met à disposition des modèles frontière en garantissant par contrat que les données ne sortent pas du compte AWS client pour remonter chez les fournisseurs de LLM.

Cette combinaison Local + Bedrock permet un workflow optimisé pour cadrer un contexte à partir des données métier puis l'envoyer à un LLM frontière (de classe Fable 5) pour un raisonnement plus poussé. Dans ce fonctionnement, l'utilisateur peut travailler soit en local soit chez un fournisseur et faire des itérations entre les deux. En complément, il garde la maîtrise de ce qu'il envoie chez le fournisseur en affinant son contexte.

Architecture phase 2 :



4. Réalisation

Fonctionnalités clés livrées (ex. synthèse contextuelle automatique, état des lieux des dernières modifications, etc.)

Téléchargement, OCR, chunking, indexation, RAG, intégration ASANA, frontend d'usage chat, sécurité.

Porteur du projet :

Jean-Noël CHARON : Architecte cloud et IA - Conception de la solution

Axel Biehler : Ingénieur cloud et IA - Mise en œuvre de la solution

Durée de développement : 3 mois

5. Résultats et métriques

Volume de données ingérées : 1 To, en augmentation constante

Taux d'adoption par l'équipe support (ou nombre d'utilisateurs actifs) : 40, en cours de déploiement

Retour qualitatif

Résultats très qualitatifs sur les réponses aux questions posées, d'où le lancement de la phase 2 du projet.

6. Preuves visuelles et lien vers le résultat

Captures d'écran disponibles :