Déploiement d'un client IA Interne

Clément Le Guyadec - mars 2025

Déployer un client IA interne pour maîtriser les enjeux technologiques et réglementaires

Constats et enjeux pour la DSI

Shadow IT

Des utilisateurs exploitent déjà des IA externes sans contrôle (risques RGPD, fuite de données).

Formation des utilisateurs

Adoption de l'IA comme levier d'efficacité et d'évolution professionnelle.

Usages existants

Rédaction de documents, traductions, assistance contractuelle.

Optimisation des ressources

un LLM généraliste pour tous les usages n'est pas optimal.

Confidentialité

Certains traitements sensibles nécessitent une gestion interne des modèles IA.

Besoins identifiés par la DSI

- Avoir un système IA modulable, sécurisé et géré par la DSI
- Former les utilisateurs aux bonnes pratiques
- Maîtriser l'impact des modèles IA utilisés

Plan projet

Semaine	1	2	3	4	5	6	7	8	9	10
Enquête métier										
Sélection technologique										
Déploiement du MVP										
Formation des utilisateurs										
Évaluation et amélioration										

Besoins et cas d'usages métier

Cas d'usage	Confidentialité des données	Besoin IA
Rédaction de clauses contractuelles	Faible	LLM + RAG (sources internes)
Recherche de sources sur internet	Faible	LLM fiable + agent de recherche
Rédaction de procédures	Faible	LLM performant en rédaction
Traduction de texte	Haute	Modèle IA spécifique à la traduction
Validation informatique (URS, risques, tests)	Faible	RAG + LLM optimisé
Assistant de code IT	Faible	LLM adapté au développement
Analyse réglementaire	Faible	RAG avec référentiel GMP

Cas d'usage recensés

Solutions Retenues

Synthèse des URS retenues par la DSI

La DSI a défini un ensemble d'exigences utilisateur (URS) visant à garantir un déploiement sécurisé, efficace et conforme aux réglementations en vigueur. Les critères clés incluent :

- **Gestion centralisée des utilisateurs** : permettre un contrôle précis des accès et une administration fine des permissions.
- **Compatibilité hybride** : Intégration fluide avec des modèles IA locaux et des API distantes, assurant flexibilité et performance.
- **Sécurisation des données** : Stockage et traitement des données sous contrôle de l'entreprise pour éviter toute fuite et garantir la confidentialité.
- **Extensibilité et évolutivité** : Capacité d'ajouter des agents spécialisés (RAG, recherche web) et d'adapter l'outil aux futurs besoins métier.
- **Conformité réglementaire** : répondre aux exigences RGPD avec stockage des données en Europe et mise en place d'un DPA (data processing agreement).

Critère	URS	OpenWebUI
Gestion des utilisateurs	Multi-accès sécurisé	✅ Oui, gestion granularisée
IA locale et API	Compatibilité hybride	✅ Ollama + API externes
Sécurisation des données	Contrôle interne	✅ Données stockées localement
Extensibilité	Plugins et agents IA	✅ Agents intégrés (RAG, web search)
Conformité RGPD	Données en Europe	✅ Oui, stockage et utilisation contrôlés (selon modèle implémenté)

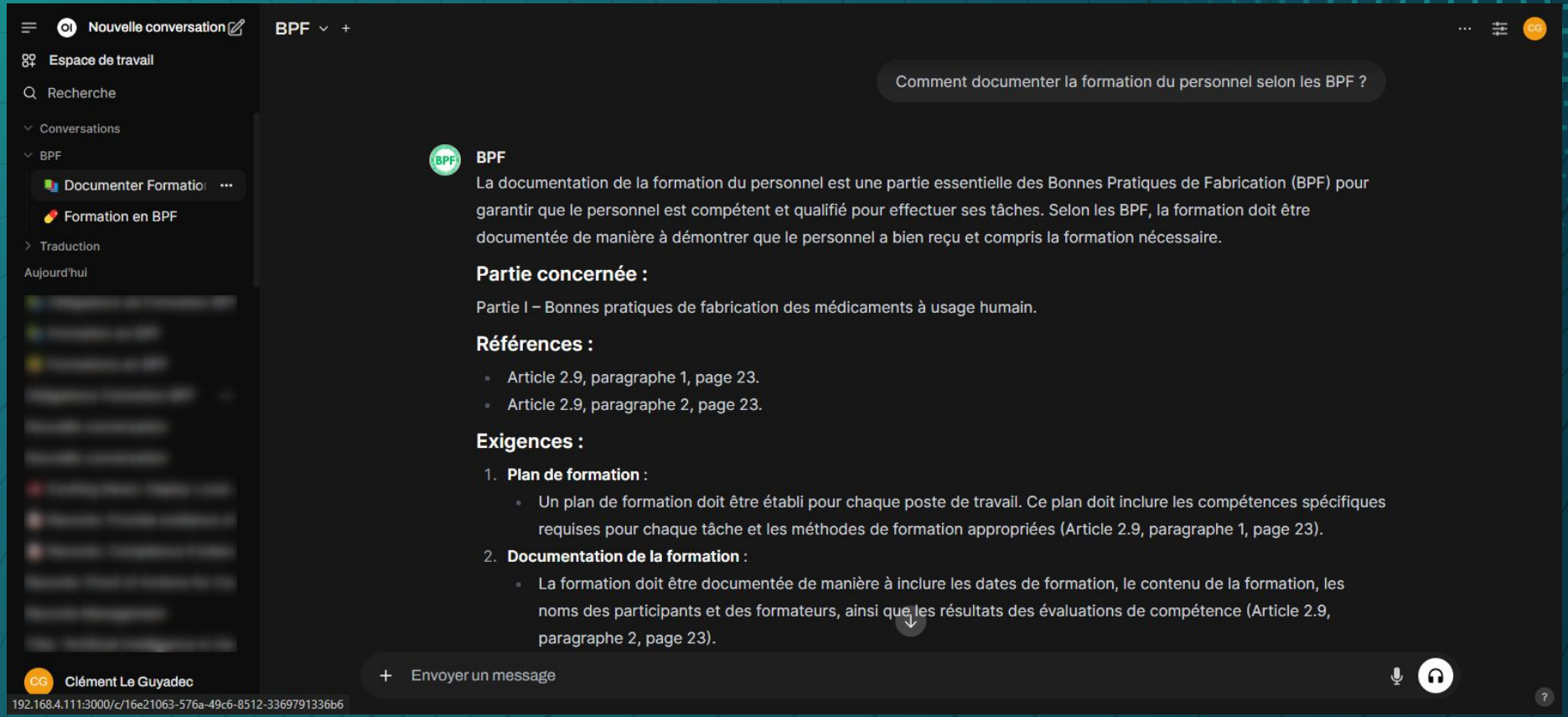
- OpenWebUI

Utilisation de Mistral

- **Engagement contractuel** : non-utilisation des données pour l'entraînement des modèles.
- Hébergement en Europe pour conformité avec le RGPD.
- Signature d’un Data Processing Agreement (DPA) avec clauses spécifiques sur la gestion des données sensibles.
- Possibilité de fine-tuning sans transfert de données externes.

Focus sur un cas d'usage

Exploitation d'un Référentiel Qualité (RAG)



Exploitation d'un Référentiel Qualité (RAG)

Configuration du client AI et test du modèle

- Utilisation de chromaDB et de d'un model multilangues (intfloat/multilingual-e5-large) pour l'embedding et le stockage local
- Ajustement de la taille des chunks et du chevauchement des chunks pour affiner le référencement
- Utilisation d'un modèle de re-ranking (bge-reranker-v2-m3) couplé à un content analysis tookit (Tika) pour améliorer la qualité des résultats
- Création d'une base de connaissance "BPF" contenant le référentiel au format PDF des fichiers structurés par partie/chapitre avec les mots-clés (fichier json)
- Utilisation des API Mistral et du modèle Mistral-large pour l'analyse du texte et les réponses utilisateur
- Configuration d'un prompt système dédié pour encadrer la démarche du modèle et répondre de manière généraliste et d'un prompt pré-enregistré (/bpf) en cas de besoin d'une réponse structurée comprenant le détail des chapitres

Validation du modèle

- Définition d'une grille d'évaluation
  - Exactitude (50%)
  - Complétude (30%)
  - Format & clarté (20%)
- Réalisation d'une série de tests avec le service Affaires réglementaires pour valider la pertinence et la qualité des réponses

Challenges rencontrés

Capacité du serveur pour l'utilisation de modèles locaux (serveur virtuel avec 16 coeurs et 64 Go de ram, pas de GPU) : capacité trop faible pour une utilisation de modèles > 3b pour du multi-user

=> Utilisation de modèles via les API

Capacité du modèle à traiter les données du référentiel (PDF de plus de 400 pages)

=> Création de fichier json en complément pour segmenter les données chapitres et mots-clés

=> Utilisation d'un prompt utilisateur pré-configuré en complément du prompt système pour structurer la réponse

Merci pour votre attention

- https://github.com/cleguyadec/jedha-Generative-AI
  - Prompt système et user prompt pré-configuré dans OpenWebUI
  - Fichiers json alimentant la base de connaissance pour améliorer la qualité des réponses
  - Grille d'évaluation du modèle