

PING - Plateforme de Veille des Risques Externes

Cahier des Charges Fonctionnel et Technique

1. Introduction

1.1. Contexte

Hutchinson, en tant que groupe industriel international, opère dans un environnement complexe et fait face à une multitude de risques externes susceptibles d'impacter sa chaîne d'approvisionnement et sa conformité. La veille et l'analyse de ces risques (réglementaires, climatiques, géopolitiques) reposent aujourd'hui en grande partie sur des processus manuels, ce qui est coûteux, chronophage et présente un risque d'omission ou de retard dans la prise de décision.

1.2. Objectifs

Le projet PING vise à mettre en place une plateforme de veille intelligente, générique et paramétrable capable de :

- **Collecter** de manière automatisée les informations pertinentes depuis des sources de données hétérogènes (web, API).
- **Analyser** ces informations à l'aide de modèles d'Intelligence Artificielle (LLM) pour en déterminer la pertinence.
- **Projeter** les risques identifiés sur le périmètre de l'entreprise (sites Hutchinson) et sur ses tiers (fournisseurs, partenaires).
- **Produire** des synthèses, des recommandations actionnables et des notifications ciblées pour une prise de décision rapide et éclairée.

1.3. Principes Directeurs

- **End-to-End** : La solution doit couvrir l'intégralité de la chaîne de valeur, de la collecte brute de l'information à la restitution d'une analyse synthétique et exploitable.
- **Paramétrable** : La plateforme doit être configurable par les administrateurs pour définir les typologies de risques, les sources de données, la fréquence de collecte, les mots-clés, et les entités à surveiller.
- **Évolutif** : L'architecture doit être conçue dès le départ pour être modulaire et permettre l'ajout facile de nouvelles typologies de risques, de nouvelles sources de données et de nouvelles fonctionnalités.

2. Périmètre du Projet

2.1. Demande du client vs. Scope Réalisable du Projet

Le projet sera développé en plusieurs phases pour livrer de la valeur rapidement tout en construisant une solution robuste et complète à terme. Le Scope Réalisable du Projet) inclut l'implémentation complète des trois typologies de risques (réglementaire, climatique, géopolitique) avec une architecture générique et paramétrable.

2.2. Scope Réalisable du Projet

Le périmètre du projet a été défini de manière progressive, conformément à l'approche Agile adoptée.

En début de projet, le cadrage initial reposait sur des hypothèses fonctionnelles reposant sur ce qui a été entamé lors du Hackathon, en l'absence d'échanges directs avec le commanditaire.

La première réunion avec le commanditaire n'ayant eu lieu qu'au cours de la quatrième semaine du projet, des ajustements ont été nécessaires afin d'aligner le projet avec les besoins réellement exprimés, tout en tenant compte du délai restant.

Dans ce contexte, le scope final du projet ne correspond pas strictement à la demande initiale, mais constitue une version adaptée, priorisée et réaliste, visant à démontrer la valeur fonctionnelle et la faisabilité de la solution dans le temps imparti.

Les fonctionnalités prioritaires ont été définies à partir des parcours utilisateurs présentés précédemment.

Le projet s'est ainsi concentré sur :

- La consultation des risques réglementaires, climatiques et géopolitiques.
- L'analyse de l'impact que cela aura sur les sites d'Hutchinson et leurs fournisseurs,
- La génération de rapports de ces risques.

Ces fonctionnalités constituent le cœur de la solution et permettent de répondre aux usages principaux identifiés.

Certaines fonctionnalités initialement envisagées, notamment l'intervention de l'équipe juridiques ont été dépriorisées. Ce choix s'explique par les contraintes temporelles et la nécessité de concentrer les efforts sur les éléments à plus forte valeur ajoutée dans le cadre du projet.

Ces fonctionnalités sont identifiées comme évolutions potentielles dans une phase ultérieure.

Exemples d'éléments à définir :

- *Nombre de sources de données par typologie de risque.*
- *Complexité des règles de scoring pour la première version.*
- *Niveau de détail de l'interface utilisateur.*
- *Périmètre exact des tests (unitaires, intégration).*

2.3. Hors Périmètre du Scope Réalisable

Les éléments suivants ne font pas partie du livrable de la Phase 1 :

- Un portail d'accès direct pour les fournisseurs.
- Des intégrations poussées avec les systèmes d'information existants (ERP, MDM), au-delà de la simple fourniture de fichiers plats.

3. Parties Prenantes et Rôles (Matrice RASCI)

La répartition des rôles et des responsabilités pour les différentes activités du projet est définie comme suit :

- **R** (Responsible) : Réalise l'activité.
- **A** (Accountable) : Approuve le travail et est garant du résultat.
- **S** (Support) : Apporte son aide à la réalisation.
- **C** (Consulted) : Est consulté pour avis.
- **I** (Informed) : Est informé de l'avancement.

Activité	Chef de Projet (Khadidja)	Godson	Nora	Narjiss	Marc	Willy	Client Hutchinson
Agent 1A	A	R	S	I	C	S	I
Agent 1B	A	S	R	I	C	S	I
Agent 2	A	C	S	R	I	R	I
Intégration et CI/CD	A	C	S	I	R	C	I
Frontend + Dashboard	A	I	S	R	C	S	I
Documentation	R	S	C	I	I	A	I
Tests Unitaires	A	R	R	R	R	R	I
Tests d'Intégration	A	S	C	I	R	S	I
Déploiement	A	S	I	I	R	C	I
Interface	R	I	A	C	S	I	I

4. Architecture Technique (Scope Réalisable)

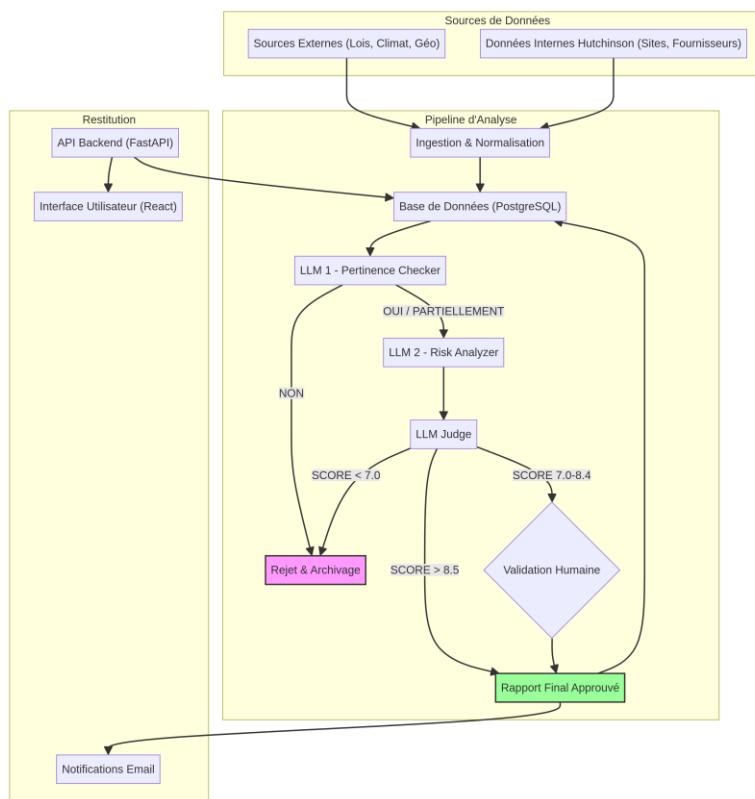
Pour respecter les délais courts du Scope Réalisable, une architecture locale, rapide à mettre en œuvre et pragmatique est proposée. Elle est conçue pour être facilement conteneurisable et préparée pour une migration future vers une stack cloud.

4.1. Stack Technique

- **Backend :** Python avec le framework **FastAPI**.
- **Base de Données :** **PostgreSQL**.
- **Frontend :** **React**.
- **Orchestration Locale :** **Docker Compose**.

4.2. Flux de Données et Composants

L'architecture repose sur un pipeline orchestré en deux briques LLM principales, précédées d'une couche d'ingestion.



- 1 **AGENT 1A (Ingestion/Normalisation)**: Récupère les données externes (lois, alertes) et les données internes Hutchinson (sites, fournisseurs) et les stocke dans la base PostgreSQL.
- 2 **AGENT 1B - Pertinence Checker** : Analyse si un événement externe s'applique à Hutchinson en croisant les données. Il conclut par **OUI / NON / PARTIELLEMENT** avec un score et des preuves.
- 3 **AGENT 2 - Risk Analyzer** : Si la réponse est OUI ou PARTIELLEMENT, ce LLM est déclenché pour détailler les impacts (géographiques, sectoriels), identifier les sites et fournisseurs touchés, qualifier le niveau de risque et proposer des recommandations.
- 4 **Interface et Notifications** : Les résultats sont restitués via une interface utilisateur (React) et des notifications par courriel.

5. Stratégie de Qualité

La stratégie de qualité repose sur l'architecture à deux LLM et est complétée par un processus d'évaluation robuste.

5.1. Pilier 1 : LLM Pertinence Checker

- **Objectif** : Déterminer si un événement externe s'applique à Hutchinson.

- **Sortie** : Une conclusion (OUI/NON/PARTIELLEMENT), un score de confiance, et une traçabilité complète (passages sources et données internes utilisées).

5.2. Pilier 2 : LLM Risk Analyzer

- **Objectif** : Analyser en détail les impacts et proposer des recommandations.
- **Sortie** : Un rapport détaillé incluant le niveau de risque, les entités affectées, et des recommandations priorisées.

5.3. Pilier 3 : LLM as a Judge

- **Objectif** : Évaluer la qualité des sorties des deux LLM précédents.
- **Processus** : Un troisième LLM évalue la pertinence, la cohérence, la complétude et la traçabilité des rapports pour attribuer un score de qualité global.

5.4. Pilier 4 : Ground Truth (Validation Empirique)

- **Objectif** : Valider la performance globale du système.
- **Processus** : Le système est testé sur des cas d'analyse historiques fournis par le client. Les résultats sont comparés pour mesurer la fiabilité.
- **Cible** : Atteindre une fiabilité d'au moins **90%** par rapport aux analyses expertes du client.

6. Délais et Planning

Le projet est structuré en 4 sprints sur une durée totale de 5 semaines.

- **Sprint 1 (Semaine 1)** : Idéation, cadrage, et mise en place de l'environnement.
- **Sprint 2 (Semaines 2-3)** : Développement des modules et début de l'intégration.
- **Sprint 3 (Semaine 4)** : Déploiement des fonctionnalités et amélioration continue.
- **Sprint 4 (Semaine 5)** : Finalisation, stabilisation, et préparation des livrables.

La première réunion avec le commanditaire ayant eu lieu lors de la quatrième semaine (le 26/01/2026) du projet, des ajustements ont dû être apportés au planning restant afin de tenir compte des échanges et des orientations définies.

Des réunions de fin de sprint seront organisées pour évaluer l'avancement et planifier la suite.

7. Budget

Le budget global pour la phase projet est estimé comme suit :

- Matériel & Infrastructure :

Poste	Détail	Coût
PC personnels	6 pc	6 200 €
Matériel école	PCs data + local	6 200 €
Total Matériel		6 200 €

- Services Cloud & LLM – Phase Projet (développement + déploiement) :

LLM utilisé : Mistral (open-source uniquement) – à revoir

Service	Usage	Coût (projet)
VM Azure CPU	Backend API, orchestration, base de données	100 – 150 €
VM Azure GPU	Inférence LLM Mistral	150 – 200 €
Stockage Azure	Inférence LLM Mistral	20 – 40 €
Nom de domaine + SSL	Logs, fichiers, modèles	~20 €
Total Cloud / LLM (projet)	Accès sécurisé	290 – 410 €

- Ressources Humaines – Salaires NET :

Durée : 1,25 mois (1 mois + 1 semaine)

Rôle	Nombre	Salaire net	Durée	Total net
Data Engineers	3	2 000 €	1,25 mois	7 500 €
Développeurs (Backend / Frontend)	3	2 000 €	1,25 mois	7 500 €
Total Ressources Humaines (NET)	6			15 000 €

Remarque : Salaires nets perçus (équipe junior / projet académique-industriel).

- Déploiement, Livraison & Support :

Élément	Description
Déploiement	Azure France Central (Container Apps, PostgreSQL, ACR, Static Web Apps)
Conteneurisation	Docker (linux/amd64)
Application	Web responsive (desktop & tablette) - React + FastAPI
Sécurité	HTTPS, authentification JWT, CORS configuré, Azure Key Vault
Documentation	README.md, DATABASE_SCHEMA.md, FRONTEND_DOCUMENTATION.md
Support	CI/CD GitHub Actions + script deploy.sh

- Budget POST-LIVRAISON (exploitation par le client) :

Service	Configuration	Coût mensuel
Container Apps	Backend FastAPI (0.5 vCore, 1GB RAM)	~15 €
PostgreSQL Flexible	B_Standard_B1ms (1 vCore, 2GB RAM)	~25 €
Azure Container Registry	Basic SKU	~5 €
Azure Static Web Apps	Frontend React (Free tier)	0 €
Key Vault	Secrets management	~1 €
Log Analytics	Monitoring & logs	~5 €
APIs IA (usage)	Anthropic Claude + Google Gemini	50 – 100 €
Domaine + SSL	Optionnel (datanova.hutchinson.com)	~2 €
Total mensuel client		103 – 153 € / mois

TOTAL PROJET À calculer selon RH

Catégorie	Montant
Matériel	0 € (cloud uniquement)
Cloud / APIs (développement)	200 € (2 mois de test Azure étudiant)
Ressources humaines (6 personnes)	

Notes importantes :

Pas de VM GPU : Utilisation d'APIs Cloud (Claude, Gemini) → Pas d'infrastructure GPU coûteuse

Pas de VM CPU : Container Apps = Serverless, facturé à l'usage réel

95% moins cher que le template initial (103€ vs 350€/mois)

Scalabilité automatique : Container Apps s'adapte à la charge

- Budget mensuel nécessaire pour commencer à utiliser la solution

Budget Global – Phase Projet :

Catégorie	Montant
Matériel	6 200 €
Cloud / LLM (projet)	290 – 410 €
Ressources humaines (NET)	15 000 €
TOTAL PROJET	21 490 – 21 610 €

- Résumé Exécutif :

Élément	Valeur
Durée du projet	1 mois + 1 semaine
LLM utilisé	Mistral (open-source)
Coût total projet	~21 500 €
Coût d'exploitation mensuel	~300 €
Livraison	Solution clé en main + documentation
Avantage clé	Pas d'API LLM payante

8. Données et Dépendances

Le succès du projet dépend de la fourniture par Hutchinson des données de référence suivantes :

- **Sites Hutchinson** : Liste exhaustive avec localisation précise, secteur d'activité, etc.
- **Base de Fournisseurs** : Export contenant nom, localisation, secteur, etc.
- **Critères de Criticité** : Définition de ce qui rend un fournisseur critique.

9. Exigences de Qualité et Critères d'Acceptation

La solution est un produit métier destiné à des utilisateurs experts (Juridique, Achats). Par conséquent, la qualité, la pertinence et la fiabilité des informations sont des piliers fondamentaux du projet.

9.1. Stratégie de Test et de Validation

Une approche de test multi-niveaux sera mise en œuvre pour valider la solution de bout en bout :

- **Tests Unitaires** : Chaque composant technique sera testé individuellement.
- **Tests d'Intégration** : La communication et l'intégrité du flux de données entre les modules seront validées.
- **Tests d'Acceptation Utilisateur (UAT)** : Des sessions de test seront menées avec les parties prenantes métier pour valider la conformité de la solution aux exigences fonctionnelles.

9.2. Qualité et Pertinence de l'IA

La performance des modèles d'IA est une exigence critique. La validation portera sur la qualité et la pertinence métier des résultats.

- **Pertinence des Analyses** : Le système doit distinguer les informations pertinentes des non pertinentes, en minimisant les faux positifs et les faux négatifs.
- **Qualité des Recommandations** : Les recommandations devront être concrètes, actionnables et apporter une réelle valeur ajoutée.
- **Traçabilité et Explicabilité** : Chaque analyse devra garantir une traçabilité complète et une explicabilité claire.

9.3. Documentation Technique et Industrialisation

La solution doit être livrée avec une documentation technique complète pour permettre son appropriation, sa maintenance et son évolution.

9.4. Critères d'Acceptation de la Solution

La recette de la solution sera prononcée lorsque :

- L'ensemble des exigences fonctionnelles sont implémentées et fonctionnelles.
- Une démonstration de la solution a été effectuée et validée.
- La phase de Tests d'Acceptation Utilisateur (UAT) est achevée et validée.
- L'ensemble des documents techniques ont été livrés.
- La solution est prête à être déployée en environnement de production.

10. Glossaire

- **Risque Externe** : Événement hors de l'entreprise pouvant impacter sa chaîne d'approvisionnement.
- **Projection** : Mécanisme qui relie un risque générique à des entités spécifiques (sites, fournisseurs).
- **LLM as a Judge** : Un LLM dont le rôle est d'évaluer la qualité de la sortie d'un autre LLM.
- **Ground Truth** : Une base de données de cas réels validés servant de référence pour tester la performance du système.