Evaluation Certifiante

Scénario de crise : interdiction soudaine d'accès aux API LLM dans un projet d'automatisation du SAV

Bloc 3 - C3.3

Réalisé par :

SGHIOURI Mohammed
SOBGUI Ivan Joel
BOTI Armel Cyrille
BEN LOL Oumar
DIVENGI KIBANGUDI BUNKEMBO Nagui
ELOUMOU MBOUDOU Pascal Aurele

Date : 19 Novembre 2025

Table des matières

Liste des tableaux

Liste des figures

Introduction

Le projet Dallosh pour Free Mobile, développé dans le cadre des blocs C3.1 et C3.2, repose sur l'utilisation d'API LLM tierces (Google Gemini, Mistral AI, OpenAI) pour l'analyse automatisée de 5 000 tweets SAV. Dans un contexte où un décret gouvernemental fictif interdit soudainement l'usage de ces modèles pour le traitement de données personnelles, l'accès aux API est suspendu immédiatement, bloquant ainsi la solution.

Face à cette situation de crise, une stratégie de sortie de crise complète est nécessaire. Elle doit couvrir l'analyse de la situation, l'identification des impacts, la conception d'une réponse technique et organisationnelle, la préparation d'une solution de repli viable, et l'anticipation de la suite. L'objectif est de démontrer la capacité à gérer une crise réglementaire majeure tout en préservant la relation client et la crédibilité technique.

1. Présentation du problème

1.1. Contexte général

Le projet Dallosh vise à automatiser le SAV de Free Mobile sur Twitter à partir d'un jeu de 5 000 tweets clients. La solution, conçue et testée dans les blocs précédents, repose sur des modèles LLM via API tierces pour :

- Analyse de sentiment (négatif, neutre, positif)
- Classification des requêtes (plainte, demande, suggestion)
- Priorisation (0 = basse, 1 = moyenne, 2 = haute)
- Extraction du sujet principal (main_topic)

Figure 1 : Diagramme global du projet Dallosh

1.2. Crise : interdiction d'accès aux API LLM

Un décret gouvernemental fictif, publié soudainement, interdit l'usage des modèles LLM hébergés à l'étranger pour tout traitement de données personnelles de citoyens français. Conséquence immédiate : suspension des accès API, solution inopérante.

1.3. Raisons réglementaires possibles

1.3.1. RGPD et transfert de données

Le risque de non-conformité avec le RGPD concernant le transfert transfrontalier de données par Transfert de données personnelles (tweets clients) vers serveurs non conformes (États-Unis notamment).

1.3.2. Souveraineté numérique

La dépendance technologique aux modèles étrangers compromettant la souveraineté numérique française. Il faut donc la volonté de réduire cette dépendance stratégique.

1.3.3. Cloud Act américain

Le risque d'accès aux données par juridictions extérieures via le Cloud Act, permettant aux autorités américaines d'accéder aux données stockées par des entreprises américaines, même hors territoire.

1.4. Conséquences immédiates

- **Blocage complet** : fonctions d'analyse automatique inopérantes
- **Interruption du traitement** : pipeline arrêté, pas de génération de rapports
- **Retard sur livraison** : impact sur les délais contractuels avec Free Mobile
- **Atteinte à la confiance client** : Free Mobile inquiet, direction en attente de réponse
- **Risques juridiques** : non-conformité potentielle, responsabilité contractuelle
- **Perte de crédibilité** : impact sur l'image de Dallosh

2. Analyse des impacts

2.1. Parties prenantes affectées

Tableau 1 : Évaluation des impacts du décret sur les parties prenantes

Partie prenante	Impact principal	Risques associés
Free Mobile	Suspension du projet, image ternie	Perte de confiance clients, insatisfaction
Équipe technique Dallosh	Outils et pipelines bloqués	Retard, surcharge de travail, stress
Direction Dallosh	Crise de crédibilité	Rupture contractuelle, perte de revenus
Utilisateurs finaux (agents SAV)	Réponses automatiques indisponibles	Hausse du taux d'insatisfaction, surcharge manuelle

2.2. Processus rendus inopérants

Tous les processus automatisés dépendant des API LLM sont bloqués :

- **Analyse automatique des tweets** : sentiment, intention, priorisation
- **Génération de tickets SAV automatisés** : classification et routage
- **Tableaux de bord en temps réel** : visualisations Recharts non mises à jour
- **Alertes automatiques** : notifications vers agents humains interrompues
- **Traitement des nouveaux tweets** : pipeline complet à l'arrêt

2.3. Coûts estimés

Type de coût	Description	Estimation
Financier	Suspension du contrat Free, pénalités potentielles	~15 000 € de perte
Opérationnel	Temps de réécriture des modules, développement solution de repli	2-3 semaines
Humain	Charge psychologique équipe, perte de confiance, stress	Élevé
Réputationnel	Impact image de marque Dallosh, crédibilité technique	Difficile à quantifier

3. Solutions alternatives envisagées

Trois solutions alternatives sont envisagées, avec des compromis différents entre délai, coût et conformité.

3.1. Hébergement local de LLM open source

Option long terme : déploiement local de modèles open source (Mistral 7B, LLaMA 3, Qwen3) sur serveurs souverains (OVH, Scaleway).

Avantages : - Souveraineté numérique garantie - Conformité RGPD totale - Pas de dépendance externe

Inconvénients : - Coûts élevés (GPU, stockage, maintenance) - Délai long (3-6 mois pour stabilisation) - Performance potentiellement inférieure aux modèles cloud

Délai estimé : 3 à 6 mois pour mise en production stable.

3.2. Refonte partielle avec NLP traditionnel

Option court terme : remplacer temporairement les LLM par une chaîne NLP classique.

Composants : - **Préprocessing** : nettoyage tweets (regex, stopwords, normalisation) - **Analyse de sentiment** : VADER, TextBlob (lexicons et règles) - **Classification** : TF-IDF + SVM / Naïve Bayes - **Priorisation** : règles heuristiques (détection mots-clés “urgent”, “problème”, “panne”)

Avantages : - Déploiement immédiat (1-2 semaines) - Coût faible (pas d’API externe) - Conformité RGPD (traitement local)

Inconvénients : - Performance inférieure aux LLM (moins de nuances, difficulté avec ironie/sarcasme) - Maintenance des règles heuristiques nécessaire

3.3. Traitement hybride

Option intermédiaire : combiner pré-filtrage automatique par NLP classique + relecture manuelle pour tweets prioritaires.

Fonctionnement : 1. Pré-filtrage automatique : NLP classique identifie tweets prioritaires (score > seuil) 2. Validation manuelle : équipe SAV valide les tweets prioritaires 3. Traitement standard : tweets non prioritaires traités automatiquement

Avantages : - Service minimal maintenu rapidement - Qualité préservée sur tweets critiques - Délai acceptable (2-3 semaines)

Inconvénients : - Charge manuelle accrue - Scalabilité limitée

Tableau 2 : Comparatif des solutions alternatives de traitement NLP

Solution	Délai mise en œuvre	Coût	Conformité RGPD	Performance	Pertinence court terme
LLM local (Mistral 7B)	3-6 mois	€€€€	Oui	Élevée	Non
NLP classique (spaCy, TF-IDF)	1-2 semaines	€	Oui	Moyenne	Oui
Hybride NLP + manuel	2-3 semaines	€€	Oui	Moyenne-Élevée	Oui

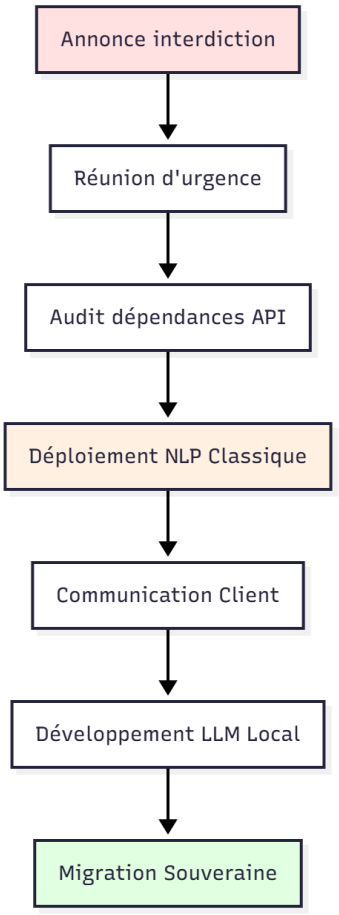
4. Plan d’action de sortie de crise

4.1. Étapes de transition

Plan en 4 étapes pour gérer la transition :

1. **Gel temporaire du pipeline LLM** : sécurisation des accès API, arrêt propre des services
2. **Audit interne** : identification des dépendances aux services tiers, évaluation des impacts
3. **Déploiement solution de repli** : mise en place du module NLP classique pour continuité minimale
4. **Communication transparente avec Free Mobile** : explication de l’adaptation réglementaire, engagement sur délais de reprise

Figure 2 : Schéma de la gestion de crise proposée



4.2. Gestion du projet pendant la crise

Organisation de la gestion de crise selon les rôles définis dans le Bloc C3.1 :

- **Chef de projet / Scrum Master** : coordination entre équipes NLP, DevOps et communication, daily stand-up renforcés
- **Équipe technique** : mise à jour du pipeline en mode “dégradé”, développement solution de repli
- **Équipe communication** : relation client, justification réglementaire, transparence sur les délais
- **Direction** : validation des priorités, allocation de ressources supplémentaires si nécessaire

Outils utilisés : Jira pour suivi des tâches critiques et documentation de crise, Slack pour communication urgente.

4.3. Plan de redéploiement

Plan en 4 phases pour le redéploiement progressif :

Phase	Objectif	Délai	Livrables
Phase 1	Remplacement par NLP traditionnel	2 semaines	Pipeline fonctionnel avec NLP classique
Phase 2	Stabilisation et reporting client	1 semaine	Dashboard opérationnel, métriques validées
Phase 3	Étude hébergement local LLM	3 mois	Architecture technique validée, POC
Phase 4	Migration complète souveraine	6 mois	Solution 100% souveraine en production

Critères de succès Phase 1 : - Pipeline NLP classique opérationnel - Traitement de 100% des tweets (performance dégradée acceptable) - Dashboard Next.js/React fonctionnel avec nouvelles données - Communication client validée

5. Prévention à moyen terme

5.1. Monitoring réglementaire

Mise en place d’une veille juridique automatisée pour anticiper les futures restrictions :

- **Sources** : CNIL, AI Act européen, décrets gouvernementaux français
- **Outils** : flux RSS, alertes Google, veille spécialisée

- **Fréquence** : revue hebdomadaire, alerte immédiate sur changements majeurs
- **Responsable** : Chef de projet avec support juridique

5.2. Préparation de versions “LLM-free”

Architecture modulaire systématique avec deux pipelines :

- **Pipeline principal** : LLM (performance optimale)
- **Pipeline de secours** : NLP classique (conformité, résilience)

Implémentation : - Abstraction de l’interface de traitement (interface commune) - Switch automatique en cas d’indisponibilité API - Tests réguliers du pipeline de secours

Cette approche s’inscrit dans les améliorations proposées dans le Bloc C3.2 concernant la modularité du code (SOLID).

5.3. Documentation des dépendances critiques

Registre interne documentant pour chaque microservice :

- **API externes utilisées** : fournisseur, modèle, endpoint
- **Clés d’accès** : localisation, rotation, gestion
- **Coûts** : estimation par transaction, budget mensuel
- **Risques** : dépendance, conformité, disponibilité
- **Plan de repli** : solution alternative documentée

Format : document Jira partagé, mis à jour à chaque modification.

5.4. Pilotage par scénarios

Documentation de scénarios de crise pour anticipation :

1. **Panne d’API ou indisponibilité cloud** : fallback automatique, monitoring proactif
2. **Crise éthique (biais IA)** : audit des résultats, ajustement des prompts, communication transparente
3. **Cyberattaque** : isolation des systèmes, sauvegarde des données, plan de reprise
4. **Changement réglementaire soudain** : procédure d’urgence documentée, contacts clés identifiés

Chaque scénario inclut : déclencheurs, actions immédiates, responsables, délais, communication.

6. Conclusion

Cette crise révèle la dépendance excessive de Dallosh vis-à-vis des LLM externes, identifiée comme risque dans le Bloc C3.1 mais non anticipée avec un plan de repli opérationnel. Elle offre cependant une opportunité stratégique : développer une offre “SAV IA souveraine” 100% conforme RGPD et opérable sur infrastructure française.

La stratégie adoptée — réponse rapide (NLP classique en 2 semaines), solution de repli viable (hybride), et anticipation réglementaire (monitoring, architecture modulaire) — renforce la crédibilité technique de Dallosh et préserve la confiance de Free Mobile.

Leçons principales : - Anticiper la résilience technologique dès la conception - Maintenir systématiquement un pipeline de secours - Documenter les dépendances critiques et les scénarios de crise - Communiquer de manière transparente avec le client en situation de crise

Cette expérience permet d’améliorer la robustesse de la solution et de transformer une contrainte réglementaire en avantage concurrentiel (souveraineté numérique).

7. Références et sources

- CNIL – RGPD : https://www.cnil.fr/fr/reglement-europeen-protection-donnees
- Rapport de planification (Bloc C3.1) : stratégie de pilotage et contraintes identifiées

8. Annexes

Annexe 1 : Procédure d'urgence documentée

Déclencheur : Interdiction soudaine d'accès aux API LLM

Actions immédiates (H+0 à H+4) : 1. H+0 : Notification équipe technique, arrêt pipeline 2. H+1 : Réunion d'urgence (direction, chef de projet, tech lead) 3. H+2 : Audit des dépendances, évaluation impacts 4. H+4 : Communication client (transparence, plan d'action)

Annexe 2 : Architecture modulaire proposée

Schéma technique de l'architecture avec abstraction permettant le switch entre LLM et NLP classique :

```
# Interface abstraite
class TextAnalyzer:
        def analyze_sentiment(self, text: str) -> dict
        def classify(self, text: str) -> dict
        def prioritize(self, text: str) -> int

# Implémentation LLM
class LLMAnalyzer(TextAnalyzer):
        def __init__(self, api_client):
        self.api = api_client

# Implémentation NLP classique
class ClassicNLP(TextAnalyzer):
        def __init__(self):
        self.sentiment_analyzer = VADER()
        self.classifier = load_model('svm_classifier.pkl')
```

Cette architecture permet un switch transparent entre les deux implémentations.