

UNIVERSITE SAAD DAHLEB DE BLIDA

Faculté des sciences

Département d'informatique



MEMOIRE DE MASTER

En Informatique

Option : Ingénierie du Logiciel

THÈME :

Évaluation de la prise en compte du Règlement Général sur la Protection des Données dans le code source

Réalisé par

BELLACHE Soumia

ZAHAF Anfel Zahida

Soutenu devant le jury

Mme BENBLIDIA Nadjia, Promotrice

Mme LAMARI Selena, Co-Promotrice

Mme OUKID Saliha, Présidente

Mme GUESSOUM Dalila, Examinatrice

29 juin 2025

Remerciements

Nous souhaitons tout d'abord exprimer notre profonde gratitude envers Allah, qui nous a accordé la force, le courage et la patience nécessaires pour mener à bien cette tâche.

Nous adressons ensuite nos remerciements à notre promotrice, Mme Benblidia, pour son encadrement professionnel, ses précieux conseils et ses observations éclairantes tout au long de ce travail.

Nous tenons également à exprimer nos sincères remerciements à notre encadrante, Mme Lamari, pour ses idées innovantes et son accompagnement constant. Son expertise et ses encouragements ont été essentiels à la bonne réalisation de ce mémoire.

Nous remercions également l'ensemble des enseignants du département d'informatique pour la qualité de leur enseignement et leur accompagnement tout au long de notre parcours universitaire.

Enfin, nous exprimons notre reconnaissance aux membres du jury pour le temps qu'ils consacreront à la lecture de notre mémoire, ainsi que pour leurs remarques et suggestions qui contribueront à l'amélioration de notre travail.

Anfel et Soumia.

Dédicace

Bismillah

Tout d'abord, je tiens à remercier le Dieu. C'est par Sa volonté et Sa grâce que j'ai pu accomplir ce travail.

Je tiens ensuite à exprimer ma plus profonde gratitude à mes chers parents, qui ont été mes plus grands soutiens tout au long de mon parcours universitaire. Leur encouragement constant et leur force ont été le fondement de ma réussite.

À ma grand-mère, qui m'a toujours soutenue d'un regard rempli de fierté et d'amour, merci pour ta force silencieuse et ton affection infinie.

Je remercie également ma sœur bien-aimée Dounia et mon frère Adnane pour leur soutien inconditionnel. Une pensée particulière à mes tantes, qui n'ont jamais cessé de m'encourager avec amour et fierté.

Je suis profondément reconnaissante envers ma binôme Anfel, sans elle, ce travail n'aurait pas atteint ce niveau de perfection.

Enfin, je remercie chaque personne qui a cru en moi, m'a encouragée et m'a souhaité du succès.

Soumia.

Dédicace

Je dédie ce mémoire à mes parents, qui ont toujours cru en moi, pour leur soutien inconditionnel, leur patience et leurs encouragements tout au long de ce parcours académique.

À mes frères, Zinnedin, Youcef et Nassim qui ont toujours été là pour m'encourager et me rappeler que je ne suis jamais seule.

À mes chères amies, Lina, Sarah, Lydia et Imene, pour leur amour, leur écoute et leur présence rassurante, même dans les moments les plus stressants.

Et enfin, à ma précieuse binôme Soumia, avec qui j'ai partagé chaque moment de ce parcours. Ta collaboration a été un véritable soutien et a grandement contribué à la réussite de ce mémoire.

Anfel.

Résumé

La protection des données personnelles est aujourd’hui une exigence juridique, éthique et technique majeure. Notre travail s’intéresse à la manière dont la conformité aux règlements juridiques et plus précisément au Règlement Général sur la Protection des Données (RGPD) est concrètement mise en œuvre dans les projets informatiques.

Pour répondre à cette problématique, nous avons adopté une approche empirique basée sur la fouille de données logicielles et l’analyse automatisée du code. Cette approche s’appuie sur plusieurs étapes : nous avons constitué un ensemble de projets en lien avec la protection des données personnelles. Ensuite, nous avons exploré l’utilisation des grands modèles de langage (LLM) pour analyser la conformité de ces projets en adoptant deux stratégies. La première consistait à analyser les messages de commits, mais cette approche s’est révélée peu concluante. Face à ces limites, notre démarche s’est ensuite orientée vers l’analyse des fichiers de documentation (README, fichiers de description), ce qui a permis d’obtenir de meilleurs résultats et de mieux cerner les pratiques des projets. Par la suite, nous avons exploré la structure du code source à l’aide de représentations syntaxiques abstraites (AST). Tous les résultats sont disponibles sur GitHub ¹.

Enfin, nous avons conçu un pipeline d’analyse des ASTs, basé sur l’architecture RAG (Retrieval-Augmented Generation) capable d’identifier les pratiques liées à la protection des données personnelles et de fournir une évaluation automatique du niveau de conformité RGPD des projets. Le système peut également accompagner les développeurs en leur suggérant des bonnes pratiques, à partir d’exemples tirés de projets déjà conformes.

Les résultats obtenus montrent que notre approche permet une première évaluation pertinente et automatisée de la conformité au RGPD dans le cadre de projets open source. Ouvrant la voie à de futures améliorations dans le domaine de l’audit logiciel assisté par l’intelligence artificielle.

Mots clés :

Protection des données personnelles, RGPD, Mining, Code source, LLM, AST, RAG.

1. <https://github.com/AnfelZhif/gdpr-practices-mining->

Abstract

The protection of personal data is now a major legal, ethical, and technical requirement. It is in this context that our work focuses on how compliance with legal regulations specifically the General Data Protection Regulation (GDPR) is concretely implemented in IT projects.

To address this issue, we adopted an empirical approach based on software data mining and automated code analysis. This approach involves several steps : We first compiled a set of projects related to personal data protection. Then, we explored the use of large language models (LLMs) to analyze the compliance of these projects, using two strategies. The first involved analyzing commit messages, but this approach proved to be inconclusive. Given these limitations, our approach then shifted towards the analysis of documentation files (README, description files), which allowed us to obtain better results and to better understand the practices of the projects. Subsequently, we explored the structure of the source code using abstract syntax tree (AST) representations. All results are available on GitHub².

Finally, we designed an analysis pipeline based on the RAG (Retrieval-Augmented Generation) architecture, capable of identifying practices related to personal data protection and providing an automated assessment of a project's GDPR compliance level. The system can also support developers by suggesting best practices, based on examples drawn from already compliant projects.

The results show that our approach enables an initial relevant and automated evaluation of GDPR compliance in open-source contexts, paving the way for future improvements in AI-assisted software auditing.

Keywords :

Personal Data Protection, GDPR, Mining, Code source, LLM, AST, RAG.

2. <https://github.com/AnfelZhif/gdpr-practices-mining->

Table des matières

Table des figures	ix
Liste des tableaux	xi
Liste des acronymes	xiii
Introduction Générale	1
Contexte de travail	1
Problématique	1
Objectifs du travail	2
Organisation du mémoire	2
1 État de l’art	5
1.1 Introduction	5
1.2 Définitions et concepts clés	5
1.2.1 Données personnelles	5
1.2.2 La protection des données	6
1.2.3 La vie privée	6
1.3 Protection des données personnelles et cadre juridique	6
1.3.1 La loi Algérienne 18-07	6
1.3.2 RGPD (Règlement Général sur la Protection des Données)	9
1.3.3 CCPA (California Consumer Privacy Act)	11
1.3.4 Impact des réglementations sur le développement logiciel	13
1.3.5 Comparaison entre les lois 18-07, RGPD et CCPA	13
1.3.6 Enjeux de la conformité de RGPD dans le développement logiciel	15
1.4 Étude des projets open-source	15
1.4.1 GitHub	15
1.4.2 GitLab	16
1.4.3 Principaux éléments d’un projet open-source	16

1.4.4	Métadonnées associées à un projet open-source	16
1.5	Technologies avancées et étude de la conformité	17
1.5.1	Les grands modèles de langage (LLM)	17
1.5.2	Fouille de logiciels et LLMs	19
1.5.3	Représentation de code	19
1.5.4	La génération augmentée par récupération (RAG)	20
1.6	Travaux connexes	22
1.6.1	Degeling <i>et al.</i> (2019)	22
1.6.2	Abdel-Jaouad Aberkane (2021)	23
1.6.3	Tahaei <i>et al.</i> (2022)	23
1.6.4	Sangaroonsilp <i>et al.</i> (2023)	23
1.6.5	Lucas Franke (2023)	24
1.6.6	Hennig <i>et al.</i> (2023)	24
1.6.7	Lucas Franke <i>et al.</i> (2024) (1)	25
1.6.8	Lucas Franke <i>et al.</i> (2024) (2)	25
1.6.9	Kapitsaki et Papoutsoglou (2024)	25
1.6.10	Tableau de comparaison	27
1.6.11	Synthèse des travaux ultérieurs	29
1.7	Conclusion	30
2	Analyse de de la conformité RGPD dans le code source via une approche RAG	31
2.1	introduction	31
2.2	Processus de notre travail	31
2.3	Collecte des projets	32
2.4	Classification de la conformité RGPD des projets	34
2.4.1	Classification basée sur l'analyse des commits	35
2.4.2	Classification basée sur l'analyse des fichiers README & description	35
2.5	Épuration des projets	37
2.6	Analyse du code source	37
2.6.1	Collecte des fichiers de code source	37
2.6.2	Représentation des fichiers de code	37
2.6.3	Analyse des AST : LLM	38
2.6.4	Analyse des AST : Approche de RAG	39
2.7	Conclusion	45
3	Implémentation du système GDPR assistant	47
3.1	Introduction	47
3.2	Environnement de Développement	47
3.2.1	Environnement matériel	47
3.2.2	Environnement logiciel	47

3.3	Collecte des projets GitHub	49
3.4	Utilisation des LLM (Large Language Model) via une API	50
3.4.1	Choix du modèle	51
3.4.2	Construction des prompts	51
3.4.3	Paramètres de l'appel au modèle	51
3.4.4	Gestion des appels API	52
3.5	La classification automatique pour l'analyse de conformité	53
3.5.1	Analyse de conformité à travers les commits	53
3.5.2	Analyse de conformité à travers la documentation (README) et les descriptions de projets	55
3.6	Analyse du code source	57
3.6.1	Analyse des concepts couverts	57
3.6.2	Sélection des projets	58
3.6.3	Génération de l'AST	59
3.6.4	Unification des AST	60
3.6.5	Analyse des AST : Approche de RAG	62
3.7	Conclusion	67
4	Tests expérimentaux et validation	69
4.1	Introduction	69
4.2	Validation manuelle des résultats de classification	69
4.2.1	Méthodologie de validation	69
4.2.2	Validation des résultats pour l'approche basée sur les commits	70
4.2.3	Validation des résultats pour l'approche basée sur les README et descriptions	71
4.3	Validation des résultats du système RAG	73
4.3.1	Méthodologie d'évaluation du système RAG	73
4.3.2	Évaluation de la récupération (retrieval)	73
4.3.3	Amélioration via reranker	76
4.3.4	Conclusion des résultats	77
4.3.5	Évaluation de la génération des réponses	77
4.4	Test du système : Analyse des Pratiques RGPD dans des Projets Open Source	80
4.4.1	Partie 1 : Identification des pratiques de conformité	80
4.4.2	Partie 2 : Évaluation de la conformité par projet	89
4.4.3	Interface du système	91
4.5	Conclusion	92
	Conclusion et perspectives	93
	Bibliographie	i

Table des figures

1.1	Le fonctionnement d'un système RAG [3]	20
2.1	Processus d'analyse de la conformité RGPD des projets open source	32
2.2	Processus de collecte des projets	33
2.3	Processus de classification des projets à l'aide d'un modèle LLM, basé sur le README et la description	36
2.4	Génération, unification et analyse des AST	38
2.5	Architecture du pipeline RAG	40
3.1	Initialisation de client Mistral	50
3.2	Appel au modèle Mistral via l'API	51
3.3	Exemple de gestion des appels api	52
3.4	Tableau des métadonnées des commits collectés	53
3.5	Tableau des commits classifiés	54
3.6	Exemple d'un prompt	56
3.7	Tableau des projet classifiés	57
3.8	Concepts du RGPD les plus couverts dans les projets	58
3.9	Exemple de la structure hiérarchique AST	60
3.10	Exemple du format d'un AST unifié	62
3.11	Ajout de séparateurs entre les documents	62
3.12	Exemple de représentation d'AST pour les imports	63
3.13	Exemple de représentation d'AST pour une fonction	63
3.14	Exemple d'une requêt utilisateur	65
3.15	Requête de recherche vectorielle avec index.search	66
3.16	Récupération des résultat	66
4.1	Calculateur de taille d'échantillon	70
4.2	Dictionnaire "ground_truth"	74
4.3	Résultats d'évaluation (FAISS) (1)	75
4.4	Résultats d'évaluation (FAISS) (2)	75

4.5	Résultats d'évaluation (reranker) (1)	76
4.6	Résultats d'évaluation (reranker) (2)	77
4.7	Réponse basée sur un projet java	78
4.8	Exemple d'un guide pratique	79
4.9	Réponse basée sur un projet js	79
4.10	Exemple de pratiques liées aux droits d'effacement	82
4.11	Exemple des pratiques liées à l'intégrité et Confidentialité	84
4.12	Exemple des pratiques identifiées	85
4.13	Principales pratiques de gestion du consentement identifiées à l'aide du Mistral	88
4.14	Exemple de détection des pratiques implémentées avec des détails techniques	90
4.15	Exemple de détection des pratiques manquantes	90
4.16	Détection de l'absence complète d'implémentation	91
4.17	Interface utilisateur	92

Liste des tableaux

1.1	Comparaison entre la loi 18-07, le RGPD et CCPA	14
1.2	Les éléments d'un projet open source	16
1.3	Les Métadonnées d'un projet open source	17
1.4	Comparaison entre les travaux	28
2.1	Un exemple des métadonnées enregistrées	34
3.1	Description des outils et des APIs utilisés	48
3.2	Processus de collecte des projets GitHub liés au RGPD	49
3.3	Tableau des projets sélectionnés pour l'analyse	59
3.4	Comparaison des représentations AST pour Java, Python, JavaScript en un format unifié	61
4.1	Validation de la classification des messages de commit	71
4.2	Validation manuelle de la classification des documentations	72
4.3	Pratiques d'implémentation de la gestion du consentement	81
4.4	Pratiques d'implémentation de l'effacement des données	83
4.5	Pratiques d'implémentation de l'intégrité et Confidentialité	85
4.6	Pratiques d'implémentation de responsabilité	86

Liste des acronymes

ANPDP	Autorité Nationale de Protection des Données Personnelles
AST	Abstract Syntax Tree
CCPA	California Consumer Privacy Act
LLM	Large Language Model
RGPD	Règlement Général sur la Protection des Données
RAG	Retrieval-Augmented Generation
UE	Union Européen

Introduction Générale

Contexte de travail

Avec l'émergence des technologies numériques et l'essor de l'économie numérique, une immense quantité de données personnelles est désormais collectée, traitée et stockée par diverses organisations à travers le monde. Ces données souvent sensibles, soulèvent des préoccupations croissantes en matière de protection de la vie privée. En réponse à ces enjeux, plusieurs lois et réglementations ont vu le jour, cherchant à encadrer l'usage de ces informations. Parmi elles, on retrouve la loi 18-07 [6] en Algérie, le Règlement Général sur la Protection des Données (RGPD) [16] en Europe, ou encore le California Consumer Privacy Act (CCPA) [7] aux États-Unis. Ces législations visent à garantir les droits des utilisateurs face à l'utilisation de leurs données. Dans ce travail, nous nous intéressons particulièrement au RGPD, étant la loi la plus robuste et large en matière de protection de la vie privée [16]. Dans ce contexte, les entreprises et les développeurs de logiciels doivent adapter leurs pratiques de développement pour se conformer à ces nouvelles exigences.

Problématique

Dans un environnement où les lois sur la protection des données se multiplient, il est crucial de s'assurer que les pratiques de développement intègrent efficacement les exigences de conformité. Cependant, cette intégration reste souvent floue et insuffisamment structurée en termes d'exigences fonctionnelles, en particulier lorsque les développeurs doivent adapter ou faire évoluer des systèmes existants. Il devient donc essentiel d'identifier les pratiques concrètes, les méthodes et les outils qui peuvent accompagner et faciliter cette mise en conformité. En parallèle, une évaluation objective du niveau de conformité atteint par ces projets est indispensable. Cela permet non seulement de mesurer l'efficacité des pratiques mises en œuvre, mais aussi de détecter les lacunes potentielles, afin de guider les développeurs vers des améliorations concrètes et conformes au cadre réglementaire.

Objectifs du travail

Nous envisageons d'adopter une approche de génie logiciel empirique afin d'examiner comment les pratiques actuelles prennent en compte la protection des données personnelles dans le développement de logiciels. En analysant des projets open source disponibles sur des plateformes comme GitHub, nous cherchons à identifier ceux qui intègrent des préoccupations relatives à la gestion de la vie privée, voire si possible construire une métrique de mesure de la conformité à des standards comme le RGPD. Cette démarche devrait permettre d'offrir une évaluation objective de la conformité aux réglementations en vigueur et de contribuer à améliorer les pratiques de développement logiciel en matière de protection de la vie privée. Pour réaliser cet objectif, nous devons :

- Réaliser un état de l'art des pratiques actuelles en matière de protection des données personnelles dans le développement logiciel.
- Analyser des projets open source pour identifier ceux qui intègrent des préoccupations de gestion de la vie privée, notamment en lien avec des règlements comme le RGPD en se basant sur des techniques d'IA.
- Utiliser des techniques de fouille de texte et de classification pour examiner le code des projets.
- Concevoir une métrique permettant d'évaluer la conformité des applications.

Organisation du mémoire

Notre mémoire est structuré en quatre chapitres principaux :

- **Chapitre 1 : État de l'art**

Ce chapitre introduit les concepts fondamentaux liés à la protection des données personnelles, en mettant en lumière les principaux cadres réglementaires (loi 18-07, RGPD, CCPA). Il explore également les plateformes open source, les technologies avancées telles que les modèles de langage (LLMs), les arbres syntaxiques abstraits (AST) et les techniques de fouille de logiciels. Le chapitre se conclut par une revue des travaux connexes, afin de situer notre contribution dans les recherches scientifiques existantes.

- **Chapitre 2 : Analyse de la conformité RGPD dans le code source via une approche RAG**

Ce chapitre présente la méthodologie adoptée dans notre travail, en détaillant les étapes de collecte des dépôts open source, la classification des projets, l'analyse du code via AST et l'intégration des modèles LLM. L'ensemble du processus est structuré dans une architecture globale conçue pour l'évaluation de la conformité RGPD.

- **Chapitre 3 : Implémentation du système GDPR assistant**

Nous détaillons ici les outils, bibliothèques et environnements utilisés (Google Colab, Python, API des LLMs), ainsi que les différents choix de modèles et techniques adoptés. Ce chapitre décrit les différentes phases de mise en œuvre : collecte des données, classification automatique des projets et des commits, génération et unification des AST, ainsi que l'application de l'approche RAG pour l'analyse contextuelle.

— **Chapitre 4 : Tests expérimentaux et validation**

Ce dernier chapitre décrit les expérimentations menées pour tester notre démarche. Il présente la validation manuelle des classifications automatiques, l'évaluation de la performance du système RAG, ainsi que l'analyse de certaines pratiques RGPD spécifiques (consentement, droit à l'effacement, confidentialité). Il s'achève par une discussion sur les résultats obtenus, les limites identifiées et les pistes d'amélioration futures.

Ce plan structuré permet d'aborder progressivement les différentes dimensions de notre travail, de la contextualisation théorique à l'évaluation expérimentale, pour offrir une vision complète et rigoureuse de notre démarche.

Chapitre 1

État de l'art

1.1 Introduction

Ce chapitre établit les fondements théoriques et contextuels nécessaires à la compréhension de notre approche d'analyse de conformité dans le développement logiciel. Nous commençons par définir les notions clés telles que les données personnelles, la vie privée et la sécurité de l'information. Nous présentons ensuite les principaux cadres réglementaires, notamment le RGPD, la loi algérienne 18-07 et le CCPA, en analysant leur impact sur les projets open source. Le chapitre aborde également les technologies avancées qui constituent l'infrastructure technique de notre approche : les modèles de langage pour l'analyse sémantique, les arbres syntaxiques abstraits (AST) pour la représentation structurée du code source, ainsi que les techniques de fouille de logiciels (software mining) utilisées pour extraire des informations pertinentes à partir de vastes corpus de code. Enfin, une revue des travaux connexes met en lumière les recherches existantes portant sur l'intégration des exigences du RGPD dans les projets open source, soulignant les limites actuelles et les opportunités pour notre contribution.

1.2 Définitions et concepts clés

Avant d'aborder les lois et réglementations en vigueur, il est essentiel de bien comprendre les notions fondamentales qui servent de base à ces règles.

1.2.1 Données personnelles

Selon le Règlement général sur la protection des données (RGPD), les données personnelles désignent toute information se rapportant à une personne physique identifiée ou identifiable. Une personne physique identifiable est une personne qui peut être identifiée, directement ou indirectement, notamment par référence à un identifiant tel qu'un nom, un numéro d'identification, des données de localisation, un identifiant en ligne ou à un ou plusieurs facteurs spécifiques à

l'identité physique, physiologique, génétique, mentale, économique, culturelle ou sociale de cette personne physique [16].

La CCPA (California Consumer Privacy Act) définit les informations personnelles comme toute information qui identifie, concerne ou peut être liée à un consommateur ou un ménage spécifique, y compris les identifiants (par exemple, nom, e-mail, adresse IP), les informations commerciales, l'activité sur internet, la géolocalisation, les données biométriques et les inférences sur les préférences ou comportements (Code civil de Californie, 1798.140(c)) [7].

1.2.2 La protection des données

La protection des données désigne un ensemble de principes et de pratiques régissant la collecte, l'utilisation, le stockage et la sécurité des données personnelles. Elle garantit que les données sont traitées à des fins spécifiques, légales et équitables. Seules les informations pertinentes et nécessaires doivent être collectées. Par ailleurs, les données personnelles doivent être conservées pendant une durée limitée et appropriée. Enfin, elles doivent être protégées par des mesures de sécurité et de confidentialité. L'accès aux données est limité aux utilisateurs autorisés[9].

1.2.3 La vie privée

Selon Ross Anderson, la vie privée est la capacité et/ou le droit de protéger vos informations personnelles et s'étend à la capacité et/ou au droit d'empêcher toute atteinte à votre espace personnel (dont la définition exacte varie d'un pays à l'autre). La vie privée peut s'étendre aux familles, mais pas aux personnes morales telles que les sociétés[4].

1.3 Protection des données personnelles et cadre juridique

De nombreux gouvernements à travers le monde mettent en place des lois et des réglementations visant à encadrer la gestion des données personnelles, dans le but de mieux protéger la vie privée des citoyens.

1.3.1 La loi Algérienne 18-07

La loi 18-07, promulguée en 2018 et entrée en vigueur le 10 août 2023, conformément à l'article 75, vise à assurer la confidentialité et la sécurité des informations individuelles en Algérie. Elle s'applique à tous les organismes publics et privés établis en Algérie ou ciblant directement des résidents algériens. Son objectif est de fixer les règles juridiques de protection de la personne physique dont ses données personnelles font l'objet d'un traitement [6].

1.3.1.1 Principes fondamentaux

La loi 18-07 responsabilise les organisations traitant des données personnelles. Elles doivent se conformer à ces principes fondamentaux et mettre en place des mesures techniques et organisationnelles adéquates [6] :

1. **Répertoriation et déclaration** : Toutes les opérations de traitement de données doivent être enregistrées et déclarées.
2. **Déclaration des traitements à l'ANPDP** : Déclaration des traitements de données personnelles à l'ANPDP (Autorité Nationale de Protection des Données Personnelles).
3. **Garanties des droits des personnes concernées** : Les individus doivent pouvoir exercer leurs droits sur leurs données (accès, rectification, opposition).
4. **Le consentement** : Toute manifestation de volonté, en connaissance de cause, par laquelle la personne concernée ou son représentant légal, accepte que ses données personnelles fassent l'objet d'un traitement manuel ou électronique. Ce traitement ne peut être effectué qu'avec son consentement exprès. Concernant les mineurs, le consentement doit être obtenu auprès du représentant légal ou à défaut, sur autorisation du juge compétent.
5. **Conservation des données** : Les données doivent être conservées uniquement pour la durée nécessaire aux finalités du traitement. L'ANPDP peut autoriser la conservation de données personnelles à des fins historiques, statistiques ou scientifiques si un intérêt légitime est démontré.
6. **Traitement des données sensibles** : Les données sensibles (santé, opinions politiques, origines ethniques, etc.) sont soumises à des règles strictes pour éviter toute exploitation abusive. Leur traitement est interdit, sauf si :
 - La personne concernée a donné son consentement préalable.
 - Le traitement est justifié par un intérêt public essentiel.
 - La loi l'autorise.
 - Une autorisation est délivrée par l'ANPDP.
7. **Interconnexion de fichiers**
 - L'interconnexion de fichiers entre organismes publics doit être autorisée par l'ANPDP et répondre à un objectif d'intérêt général.
 - Cette interconnexion ne doit pas entraîner de discrimination ni réduire les droits et libertés des personnes concernées.
8. **Sécurité des données personnelles** : L'ANPDP peut imposer des mesures de sécurisation, comme le cryptage, lorsque la transmission des données présente des risques. Le responsable du traitement doit mettre en place des mesures pour protéger les données contre :
 - L'accès non autorisé, la perte, l'altération ou la destruction.

- Les risques liés à la transmission sur les réseaux.
- Toute utilisation abusive, notamment en cas de sous-traitance, qui doit être encadrée par un contrat de sécurité.

9. **Secret professionnel et confidentialité**

- Le traitement des données personnelles est confidentiel.
- Toute personne accédant à ces données dans le cadre de ses fonctions est tenue au secret professionnel, même après cessation de son activité.
- Le responsable du traitement doit établir une charte de confidentialité à signer par les personnes ayant accès aux données.

10. **Transfert des données vers l'étranger** : Conformément aux articles 25 et 44 de la loi 18-07, tout transfert de données personnelles à l'étranger nécessite une demande d'autorisation auprès de l'ANPDP.

11. **Interdiction de la prospection directe** : La loi 18-07 interdit l'utilisation des données personnelles à des fins de prospection directe (publicité, démarchage, marketing) sans le consentement préalable de la personne concernée. Toute personne prospectée sans son accord peut s'y opposer par tout moyen auprès du prestataire de services ou de l'opérateur concerné, qui doit y mettre fin immédiatement.

1.3.1.2 **Les droits des individus selon la loi 18-07**

La loi algérienne 18-07 accorde aux personnes concernées plusieurs droits fondamentaux en matière de protection de leurs données personnelles. Parmi eux figurent :

- a) **Droit à l'information** : Selon l'article 32 de la loi 18-07, sauf si elle en a déjà eu connaissance, toute personne sollicitée pour une collecte de données personnelles doit être informée de manière explicite et non équivoque par le responsable du traitement ou son représentant sur :
 - L'identité du responsable du traitement et, le cas échéant de son représentant.
 - Les finalités du traitement.
 - Toute information supplémentaire utile (destinataire, obligation de réponse et ses conséquences, droits de la personne concernée, transfert des données à l'étranger).
- b) **Droit d'accès** : Selon l'article 34, la personne concernée a le droit d'obtenir du responsable du traitement :
 - La confirmation que ses données personnelles sont ou non traitées, ainsi que les finalités du traitement, les catégories de données concernées et les destinataires.
 - La communication, sous une forme intelligible, de ses données traitées et de toute information sur leur origine.

c) **Droit de rectification** : Selon l'article 35, la personne concernée a le droit d'obtenir gratuitement :

- L'actualisation, la rectification, l'effacement ou le verrouillage des données personnelles dont le traitement n'est pas conforme à la loi (données incomplètes, inexactes ou interdites par la loi).

d) **Droit d'opposition** : Selon l'article 36, la personne concernée peut s'opposer, pour des motifs légitimes, au traitement de ses données personnelles. Elle peut également refuser que ses données soient utilisées à des fins de prospection, notamment commerciale.

1.3.1.3 Sanctions en cas de non-conformité à la loi 18-07

Sanctions administratives : Avertissements, mises en demeure, amendes et retrait d'autorisation.

Sanctions pénales : Peines de prison (jusqu'à 5 ans) et amendes pouvant atteindre 1 000 000 DA pour des infractions graves (ex. transfert illégal de données, traitement sans consentement etc.).

1.3.2 RGPD (Règlement Général sur la Protection des Données)

Règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016, relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/CE .Le Règlement général sur la protection des données (RGPD) est un texte réglementaire européen qui encadre le traitement des données de manière égalitaire sur tout le territoire de l'Union européenne (UE). Il est entré en application le 25 mai 2018. Le RGPD s'applique à tout traitement de données personnelles effectué par une organisation, publique ou privée, établie dans l'Union européenne, ou visant des résidents de l'UE, même si le responsable du traitement est situé hors de l'UE [16].

1.3.2.1 Objectifs du RGPD

Le Règlement Général sur la Protection des Données (RGPD) poursuit plusieurs objectifs fondamentaux. Il vise tout d'abord à établir un cadre clair pour la protection des personnes physiques en ce qui concerne le traitement des données à caractère personnel. Il cherche également à garantir les libertés et droits fondamentaux des individus, en particulier leur droit à la protection de leurs données personnelles. Enfin, le RGPD a pour ambition de favoriser la libre circulation des données au sein de l'Union européenne, en évitant les restrictions injustifiées qui pourraient être imposées au nom de la protection des personnes concernées [16].

1.3.2.2 Les principes fondamentaux du RGPD

Les principes du RGPD sont au cœur de la législation sur la protection des données. Ils guident les pratiques des organisations afin de garantir un traitement éthique et respectueux des données personnelles [16].

1. Minimisation des données

- Ne collecter que les données strictement nécessaires pour atteindre un objectif déterminé et légitime.
- Ne pas traiter les données ultérieurement de façon incompatible avec cet objectif initial.

2. Le recueil du consentement

Le consentement était déjà inscrit dans la loi Informatique et Libertés, mais le RGPD renforce les exigences :

- Droit au retrait : Une personne doit pouvoir retirer son consentement à tout moment via un moyen simple et équivalent à celui utilisé pour le donner.
- Preuve du consentement : Le responsable du traitement doit pouvoir prouver que la personne a donné son consentement dans des conditions valides.

3. Transparence

- Informer clairement les individus sur l'utilisation de leurs données dès leur collecte.
- Leur fournir des informations sur leurs droits et les modalités d'exercice de ces droits.

4. Facilitation de l'exercice des droits des personnes

- Mettre en place des modalités permettant aux personnes d'accéder, rectifier ou supprimer leurs données.
- Répondre rapidement aux demandes d'opposition ou de portabilité des données.

5. Limitation des durées de conservation

- Conserver les données uniquement le temps strictement nécessaire à la finalité poursuivie.
- Les anonymiser, archiver ou détruire lorsque leur conservation n'est plus justifiée.

6. Sécurisation des données

- Mettre en place des mesures techniques et organisationnelles pour garantir la sécurité des données.
- Restreindre les accès aux seules personnes autorisées.

7. Mise en conformité continue

- Vérifier régulièrement que les traitements restent conformes aux exigences du RGPD.
- Mettre à jour les mesures de sécurité et les procédures si nécessaire.

8. Transfert des données vers l'étranger

- Les transferts internationaux de données sont autorisés sous conditions strictes (articles 44 à 50 du RGPD), notamment vers des pays offrant un niveau de protection adéquat.

1.3.2.3 Les droits des individus selon le RGPD

Le RGPD garantit aux citoyens européens plusieurs droits fondamentaux.

a) **Droit d'accès (Article 15)**

- Les individus peuvent demander l'accès à leurs données et obtenir des informations sur leur utilisation.

b) **Droit à l'information (Article 13)**

- C'est le droit de toute personne à être informée de façon claire, transparente et complète lorsque ses données personnelles sont collectées ou traitées.

c) **Droit de rectification (Article 16)**

- Correction des données inexactes ou incomplètes.

d) **Droit à la limitation du traitement (Article 18)**

- Ce droit permet à une personne concernée de demander que ses données personnelles ne soient plus utilisées temporairement, mais qu'elles soient conservées.

e) **Droit à l'effacement « Droit à l'oubli » (Article 17)**

- Suppression des données si elles ne sont plus nécessaires, si le consentement est retiré, ou si elles ont été collectées illégalement.

f) **Droit à la portabilité des données (Article 20)**

- Recevoir ses données dans un format structuré et les transférer à un autre organisme.

g) **Droit d'opposition (Article 21)**

- S'opposer au traitement des données, notamment en cas de marketing direct.

1.3.2.4 Sanctions en cas de non-conformité au RGPD

Les entreprises qui ne respectent pas le RGPD s'exposent à des sanctions financières sévères : Jusqu'à 20 millions d'euros ou 4 % du chiffre d'affaires annuel mondial (selon le montant le plus élevé) (Article 83) [16].

1.3.3 CCPA (California Consumer Privacy Act)

C'est une loi de l'État de Californie promulguée en 2020 qui protège et applique les droits des Californiens concernant la confidentialité des informations personnelles des consommateurs (PI) [28].

En Californie, les objectifs de ce mouvement ont été transformés en loi, grâce à l'adoption du CCPA. Il porte un coup dur aux droits des consommateurs et à la cybersécurité en dotant l'État de Californie d'un cadre efficace pour faire respecter les lois et réglementations sur la confidentialité des données. Il offre aux résidents californiens un moyen d'accéder au droit d'action privé, afin de demander un recours juridique en cas de violation de données [28].

1.3.3.1 Les principes fondamentaux du CCPA

La CCPA énonce plusieurs principes clés visant à encadrer la gestion des données personnelles, parmi lesquels [28] :

1. **Localisation des données** : Identifier et répertorier les données collectées, qu'elles soient internes (employés, candidats) ou externes (clients).
2. **Protection des données** : Sécuriser toutes les données personnelles collectées, notamment celles des mineurs.
3. **Notification des consommateurs** : Informer les personnes concernées via une déclaration de collecte, avant ou au moment de la collecte.
4. **Politique de confidentialité** : Établir et publier une politique de confidentialité détaillée.
5. **Gestion des demandes de consommateurs** : Mettre en place un système efficace pour traiter les demandes d'accès, de suppression ou de modification des données.
6. **Minimisation des données** : Ne collecter que les informations strictement nécessaires et prévoir des mesures pour limiter les risques en cas de violation.
7. **Formation et sensibilisation** : Assurer que dirigeants et employés comprennent et respectent les exigences du CCPA.

8. Veille réglementaire

Suivre les évolutions du CCPA et adapter les pratiques en conséquence .

1.3.3.2 Droits et protections du CCPA

Les directives de la CCPA ont été conçues pour donner aux consommateurs californiens un ensemble de droits qui traitent expressément de la confidentialité des données personnelles et leur offrent des garanties de sécurité raisonnables. Ces droits incluent la capacité des Californiens à formuler des demandes de consommateurs concernant leurs données clients. Ces demandes peuvent inclure comment [28] :

- Refuser la vente de leurs informations personnelles à des tiers, c'est-à-dire exercer leur droit de retrait (*opt-out*) en utilisant la directive dite « Ne pas vendre mes informations personnelles » (*Do Not Sell My Personal Information*).
- Consentement requis uniquement pour la vente de données de mineurs.
- Demander des données sur toute information personnelle qui a été collectée (droit d'accès)
- Demander que toutes les données collectées sur ce consommateur soient supprimées (Le droit à l'oubli).

1.3.3.3 Sanctions en cas de non-conformité à la CCPA

Le courtage de données est une industrie en pleine croissance, évaluée à 240 milliards de dollars en 2021 et pouvant atteindre 450 milliards d'ici 2030. Face à cet enjeu, la California Privacy

Protection Agency (CPPA) veille à l'application du CCPA et sanctionne les entreprises non conformes.

Sanctions prévues :

- 2 500 \$ par violation involontaire.
- 7 500 \$ par violation intentionnelle.

1.3.4 Impact des réglementations sur le développement logiciel

Les produits logiciels collectent des données utilisateur afin d'améliorer l'expérience utilisateur, notamment grâce à des comportements applicatifs personnalisés rendus possibles par l'apprentissage automatique. Si cela peut être bénéfique pour les utilisateurs, cela peut également représenter une menace pour leur bien-être. Pour protéger leurs citoyens, plus de 100 gouvernements à travers le monde élaborent des lois et réglementations sur la protection des données, afin de limiter la manière dont les données personnelles sont collectées, traitées, stockées et conservées [18]. Les avancées récentes en matière de législation sur la vie privée, comme le Règlement Général sur la Protection des Données (RGPD) de l'Union européenne, ont contraint la communauté logicielle à accorder une attention particulière à la confidentialité des données des utilisateurs. Toutefois, peu d'études ont été consacrées à la manière dont ces lois sont mises en œuvre dans le processus de développement logiciel [18].

1.3.5 Comparaison entre les lois 18-07, RGPD et CCPA

Afin de mieux comprendre les spécificités et les différences entre les principaux cadres juridiques de protection des données, nous proposons une comparaison entre trois réglementations. Le tableau 1.1 présente une comparaison entre les lois 18-07, RGPD et CCPA.

Critères	RGPD (UE)	Loi 18-07 (Algérie)	CCPA (Californie, USA)
Objectif	Protection des données tout en assurant leur libre circulation dans l'UE et le respect des droits fondamentaux.	Sécuriser les données personnelles des citoyens	Donner aux consommateurs californiens plus de contrôle sur leurs données
Portée géographique	UE et les organisations qui traite des données de personnes de l'UE.	Nationale (Algérie)	Régionale (Californie)
Champ d'application	Toute organisation traitant des données de résidents européens	Entreprises établies en Algérie ou ciblant ses résidents	Entreprises traitant les données des résidents californiens
Consentement	Doit être libre, éclairé, explicite et révocable	Le droit au retrait et preuve du consentement	Droit de refus (opt-out), consentement requis uniquement pour les mineurs
Droits des individus	Accès, information, rectification, suppression, portabilité, opposition, limitation	Droit à l'information, accès, rectification, opposition	Accès, suppression, refus de la vente des données
Transfert des données	Hors l'UE : Mécanismes détaillés (décisions d'adéquation)	Hors l'Algérie : Nécessite une autorisation de l'ANPDP	Pas de restrictions particulières
Portée d'application	Référence européenne avec influence mondiale	Adoption progressive	Influence régionale croissante
Sanctions	Jusqu'à 20 M€ ou 4% du chiffre d'affaires annuel global de l'entreprise	Amendes et peines de prison (jusqu'à 5 ans)	2 500 \$ par violation involontaire et 7 500 \$ par violation intentionnelle

TABLE 1.1 – Comparaison entre la loi 18-07, le RGPD et CCPA

Ce tableau comparatif met en évidence les convergences et les divergences entre les trois législations en matière de protection des données personnelles. On constate que le RGPD est le cadre le plus strict et le plus complet, avec une portée extraterritoriale, une exigence forte en matière de consentement, ainsi que des sanctions élevées. La loi 18-07 algérienne reste encore en phase d'adoption progressive et nécessite des autorisations spécifiques, notamment pour le transfert des données. Quant au CCPA, bien qu'il soit plus limité géographiquement, il renforce les droits des consommateurs californiens et s'inscrit dans une tendance croissante aux États-Unis vers une meilleure protection des données. Cette comparaison met en lumière l'importance croissante de la conformité réglementaire à l'échelle mondiale.

C'est pour cela que nous avons choisi le RGPD, en raison de sa portée, de son exigence en matière

de conformité et de son influence internationale.

1.3.6 Enjeux de la conformité de RGPD dans le développement logiciel

Les exigences du RGPD, la complexité du jargon juridique et la longueur des textes réglementaires ont entraîné des efforts de développement considérables. Les développeurs ont également rencontré des difficultés pour gérer les données des utilisateurs, garantir l'anonymisation des informations personnelles et respecter les droits des individus. Ces nouvelles tâches ont ajouté de la complexité aux processus de développement[20]. De plus, de nombreux développeurs rencontrent des difficultés à se conformer au RGPD, principalement en raison de l'absence d'exigences fonctionnelles explicites permettant de guider efficacement cette conformité[19].

Face à ces défis, la nécessité d'outils et de ressources automatisés pour aider à la conformité au RGPD s'est clairement manifestée comme une véritable opportunité d'innovation. Ces outils permettent de réduire la charge de travail des développeurs et d'améliorer l'efficacité des processus [19].

Dans ce contexte, il faut d'abord examiner comment les pratiques de conformité sont adoptées au sein des projets open-source.

1.4 Étude des projets open-source

Les projets open-source constituent des ressources précieuses pour les développeurs, en garantissant la transparence du code et en favorisant la collaboration au sein de la communauté de programmation. Ces projets offrent une opportunité pour mener des analyses approfondies, notamment en matière de conformité réglementaire. Pour explorer ces aspects, il est essentiel de s'appuyer sur des plateformes de développement collaboratif largement utilisées :

1.4.1 GitHub

GitHub est une plateforme cloud qui permet aux développeurs de stocker, gérer, suivre et collaborer sur du code source. Selon des recherches menées par GitHub, plus de 100 millions de développeurs utilisent actuellement GitHub pour créer, maintenir et contribuer à des projets logiciels. La communauté du développement logiciel a adopté GitHub comme une plateforme essentielle pour gérer leurs projets [23].

1.4.2 GitLab

GitLab est une plateforme complète de développement logiciel et de DevOps¹ qui aide les équipes à collaborer sur le codage, les tests et le déploiement d'applications. Elle intègre le contrôle de version, l'intégration continue et la livraison continue (CI/CD), la gestion de projet, et bien plus encore dans une seule interface [24].

1.4.3 Principaux éléments d'un projet open-source

Le tableau 1.2 illustre les différents éléments d'un projet open source :

Élément	Description
À propos (About)	Brève description du projet
Documentation	Guides et explications (README.md).
Code source	Fichiers de code accessibles et modifiables.
Commits	Série de modifications successives apportées au code, enregistrées et suivies via Git, permettant de suivre l'évolution du projet.
Issues	Suivi des bugs et des améliorations.
Licence open-source	Définit les droits d'utilisation (par exemple : MIT, GPL, Apache).

TABLE 1.2 – Les éléments d'un projet open source

Ce tableau résume les éléments clés d'un projet open source, comme la documentation, les commits, les issues ou la licence. Chacun contribue à la transparence, la collaboration et l'ouverture du projet. Il est également possible de rechercher, au sein de ces éléments, des indices de conformité au RGPD [23].

1.4.4 Métadonnées associées à un projet open-source

Le tableau 1.3 illustre les métadonnées associées à un projet open source :

Métadonnée	Description
Nom du projet	Identifiant unique du projet.
Auteur(s) / Contributeurs	Liste des personnes ayant contribué au développement.
Topic du projet	Mots-clés permettant d'identifier le domaine d'application du projet.
Lien	URL du dépôt .

1. DevOps combine le développement logiciel (dev) et les opérations (ops). C'est une méthodologie qui favorise la collaboration et la responsabilité partagée entre les équipes de développement et d'exploitation.

À propos (About)	Brève description du projet.
Statistiques du projet	Nombre d'étoiles (popularité), forks (copies pour modification), watchers (suiveurs du projet).
Date de création / Dernière mise à jour	Historique du projet et de son activité récente.

TABLE 1.3 – Les Métadonnées d'un projet open source

Ce tableau présente les métadonnées clés d'un projet open source, telles que le nom, les contributeurs, les mots-clés (topics) et les statistiques de dépôt (étoiles, forks). Ces informations permettent d'identifier, de documenter et d'évaluer un projet dans un dépôt collaboratif [23].

1.5 Technologies avancées et étude de la conformité

Dans cette section, nous allons explorer plusieurs technologies avancées qui jouent un rôle clé dans l'étude de la conformité notamment dans le contexte des projets open source. Nous aborderons les modèles de langage de grande taille (LLMs) et leur avantages ainsi que les techniques d'exploration de code, telles que l'AST (Arbre de Syntaxe Abstraite). Enfin, nous discuterons du concept de RAG (Retrieval-Augmented Generation) et de son rôle dans la génération de réponses en s'appuyant sur des sources externes..

1.5.1 Les grands modèles de langage (LLM)

Les LLM (Les grands modèles de langage) sont des réseaux de neurones avec l'architecture de transformateur², pré-entraînés sur de vastes corpus de données, principalement textuelles. Ils sont conçus pour générer des sorties de manière probabiliste à partir d'une entrée textuelle, appelée prompt ou invite. [32]. Les LLM sont un type d'intelligence artificielle (IA) qui se sont imposés comme des outils puissants pour un large éventail de tâches, notamment le traitement du langage naturel (NLP), la traduction automatique, la génération de code et les systèmes de question-réponse [8].

Après avoir expliqué ce qu'est un LLM, examinons à présent quelques modèles de LLM :

- **MistralAI Large** : Mistral Large est un modèle de langage développé par Mistral AI, capable d'atteindre des performances de haut niveau en matière de raisonnement. Il est particulièrement adapté aux tâches complexes de compréhension et de transformation de texte multilingue, ainsi qu'à la génération de code. Sa fenêtre de contexte peut atteindre jusqu'à 128k tokens, ce qui permet de traiter de très longs documents. Le modèle obtient

2. Transformer est un type d'architecture de réseau de neurones qui excelle dans le traitement des données séquentielles, et il est principalement associé aux grands modèles de langage (LLM).

d'excellents résultats sur plusieurs benchmarks de référence. Il est accessible via l'API de Mistral AI et propose des fonctionnalités telles que la génération de texte, la création de représentations vectorielles (embeddings), ainsi que l'affinage (fine-tuning) dans certaines configurations. Une offre gratuite est disponible sur la plateforme, permettant aux utilisateurs d'expérimenter et de développer sans frais dans certaines limites [47].

- **GPT-4o** : GPT-4o est un modèle de langage développé par OpenAI, capable de traiter des entrées textuelles, audio, image et vidéo, et de générer des sorties en texte, audio et image, grâce à un réseau neuronal unique. Il offre d'excellentes performances en traitement du texte en anglais et en génération de code, tout en bénéficiant d'une meilleure compréhension des contenus visuels et audio. Le modèle n'est pas gratuit et est accessible via l'API d'OpenAI [36].
- **Llama 4 Maveric** : Développé par Meta, il prend en charge des entrées textuelles et visuelles multilingues, et peut générer des sorties textuelles et du code dans 12 langues différentes. Conçu pour des tâches de vision et de langage, ce modèle est optimisé pour adopter un comportement d'assistant, effectuer du raisonnement à partir d'images et gérer une interaction multimodale à usage général. Il est accessible via plusieurs API, notamment OpenRouter, Together AI et Novita AI, permettant ainsi son intégration dans diverses applications. Le modèle n'est pas entièrement gratuit [38].

1.5.1.1 instruction générative (Prompting)

L'instruction générative est une méthode permettant d'interroger des modèles de langage (LLM) entraînés afin de générer des réponses. Les LLM peuvent être utilisés avec divers types de prompts, dans lesquels ils peuvent s'adapter aux instructions sans nécessiter de l'affinage (fine-tuning), ou dans d'autres cas, avec un affinage sur des données contenant différents styles de prompts.

Ci-dessous, nous discuterons de quelques types de prompts largement utilisés :

- **Prompt sans exemple (Zero-Shot Prompting)** : Les LLMs sont capables de répondre à des requêtes jamais vues auparavant en s'appuyant sur leurs connaissances acquises lors de l'entraînement. Ce style de prompt demande aux LLMs de répondre à des questions d'utilisateurs sans avoir vu d'exemples dans le prompt [34].
- **Prompt avec peu d'exemples (Few-Shot Prompting)** : Ce type de prompt consiste à montrer au modèle plusieurs paires d'entrée-sortie en guise de démonstration pour qu'il comprenne le format attendu et génère la réponse appropriée [34].
- **Enchaînement de Prompt (Prompt Chaining)** : Méthodologie utilisée pour décomposer des tâches complexes en sous-tâches plus petites et plus faciles à gérer. Ce type comprend généralement plusieurs prompts séquentiels, soit spécifiques à une tâche, soit à usage général, chacun remplissant un objectif précis. La sortie de chaque étape sert d'entrée pour l'étape suivante [48].

1.5.2 Fouille de logiciels et LLMs

La fouille de logiciels (software mining) désigne l'utilisation des techniques de fouille de données (data mining) sur des données issues de l'ingénierie logicielle. Les sources de données incluent le code source, les traces d'exécution, les historiques de modifications ou encore les bases de bugs. Lorsqu'il s'agit spécifiquement du code source statique, on parle de fouille de code source (code mining) [22]. Un sous-domaine clé est la fouille de dépôts logiciels (MSR), qui extrait des informations utiles à partir de dépôts comme GitHub ou GitLab [51].

Dans ce contexte, l'utilisation des grands modèles de langage (LLMs) a révolutionné le domaine de l'ingénierie logicielle (Software Engineering), en permettant des approches plus efficaces et plus simples pour des tâches complexes. Ces modèles, entraînés sur de vastes ensembles de données, ont démontré leur capacité à assister diverses activités d'ingénierie logicielle, notamment la génération de code, la génération de documentation et l'analyse de projets logiciels [11].

1.5.3 Représentation de code

L'analyse automatisée du code source nécessite des représentations structurées qui capturent à la fois la syntaxe et la sémantique des programmes. Plusieurs techniques ont été développées pour transformer le code en structures exploitables par les algorithmes d'apprentissage automatique et les LLMs notamment :

a) **Arbre de Syntaxe Abstraite (AST)**

Un arbre de syntaxe abstraite (AST) constitue une représentation structurée fondamentale lors de l'analyse du code. Il s'agit d'une représentation arborescente d'un programme, où chaque nœud représente un élément de code spécifique. Les nœuds internes correspondent aux opérateurs (conditions, boucles, opérations logiques), tandis que les feuilles représentent les opérandes (variables, littéraux, appels de fonctions).

Pour construire ces AST, les recherches actuelles s'appuient principalement sur des analyseurs syntaxiques (parseurs) spécialisés selon le langage de programmation : JavaParser pour Java, TypedAST pour Python, Tree-sitter pour C++, entre autres [45].

b) **CST (Concrete Syntax Tree)**

Un arbre de syntaxe concrète (CST) est défini comme une structure de données qui représente la structure syntaxique du code source selon la grammaire d'un langage de programmation. Il capture la structure exacte du code source, y compris tous les tokens, la ponctuation et les espaces blancs, en préservant tous les détails syntaxiques [35].

Cette représentation détaillée permet d'effectuer une analyse et des transformations précises du code source, ce qui rend les CST particulièrement utiles pour des tâches telles que la mise en forme du code, le refactorisation et la gestion des erreurs. En conservant toutes

les informations syntaxiques, le CST offre une vue complète de la structure du code, ce qui peut être exploité pour diverses applications dans l'ingénierie logicielle [35].

Pour exploiter ces représentations automatiquement et extraire des informations pertinentes, une approche efficace doit être mise en œuvre.

1.5.4 La génération augmentée par récupération (RAG)

C'est une technique innovante de l'intelligence artificielle qui combine des modèles génératifs avec la recherche d'informations en temps réel dans des sources externes. Contrairement aux modèles traditionnels qui se basent uniquement sur les connaissances apprises lors de l'entraînement, un système de RAG permet de générer les réponses dans des données issues de sources de connaissance externes. Cette technique réduit considérablement le risque d'hallucinations, c'est-à-dire la génération d'informations incorrectes ou inventées, un problème récurrent dans les modèles génératifs classiques. Elle s'avère particulièrement précieuse dans des domaines sensibles comme la santé, le droit ou la finance, où la fiabilité de l'information et la transparence, notamment via l'indication des sources sont essentielles [49].

La figure 1.1 illustre le fonctionnement d'un système RAG, qui combine la récupération d'information à partir d'une base de connaissances externe avec la génération de la réponse par un LLM.

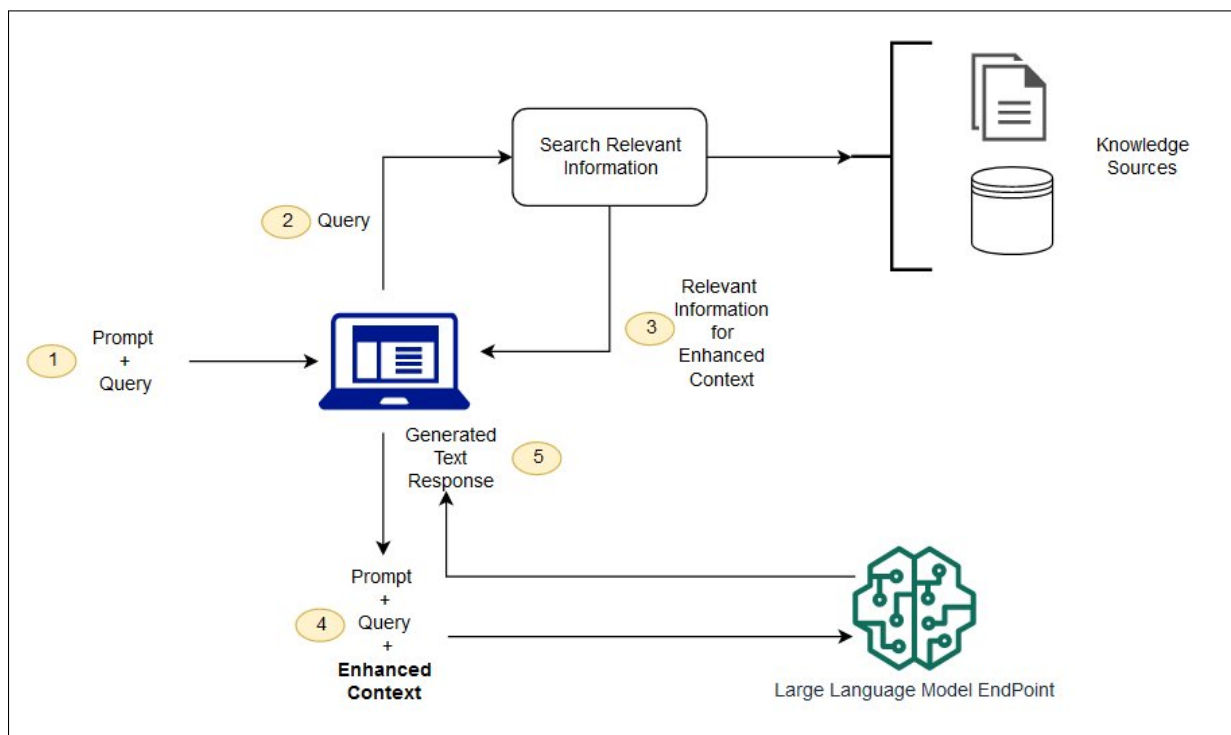


FIGURE 1.1 – Le fonctionnement d'un système RAG [3]

Pour bien comprendre l'architecture d'un système RAG, il est essentiel d'en examiner les principaux composants. De manière générale, un système RAG se compose de deux éléments

principaux :

1.5.4.1 Système de recherche d'information

Le système de recherche joue le rôle de moteur de connaissance en allant chercher les informations pertinentes dans des sources externes. Il repère et sélectionne les documents ou extraits de données les plus adaptés à la requête de l'utilisateur, afin de fournir au modèle génératif un contenu fiable, actualisé et en lien direct avec le contexte de la demande. Ce système accède dynamiquement à diverses sources de connaissances externes pour enrichir les réponses générées. Ces sources incluent des collections de documents pré-indexés (Wikipedia, dépôts scientifiques et techniques, repositories de code source, bases de connaissances d'entreprises) qui sont optimisées grâce à des représentations vectorielles (embeddings) denses pour faciliter la recherche sémantique [49].

Le système de recherche s'appuie sur différentes méthodes pour identifier et récupérer les données les plus pertinentes en fonction de la requête saisie.

- **Recherche Sparse (Techniques de Correspondance Lexicale)**

Les techniques de recherche sparse (lexical) se basent sur la correspondance des mots-clés, où les documents sont classés en fonction de la présence et de la fréquence des termes issus de la requête. Parmi les méthodes les plus courantes, on trouve la fonction de classement probabiliste BM25 (Best Matching 25) qui évalue les documents en fonction de la fréquence des termes et de la longueur des documents [49].

- **Recherche Dense (Recherche Sémantique)**

Les méthodes de recherche dense utilisent des modèles d'apprentissage profond pour créer des représentations vectorielles du texte et des requêtes, permettant une recherche sémantique plutôt qu'une simple correspondance de mots-clés. Des modèles pré-entraînés comme Sentence-BERT (SBERT) ou Dense Passage Retrieval (DPR) génèrent des embeddings capturant le sens contextuel du texte.

Une approche courante pour effectuer des recherches efficaces sur ces embeddings est l'utilisation de FAISS (Facebook AI Similarity Search), une bibliothèque permettant de réaliser des recherches de similarité rapides et évolutives sur des vecteurs denses [49].

Modèles existants pour la recherche sémantique :

- **SBERT (Sentence-BERT)** : une modification du modèle BERT pré-entraîné qui utilise des structures de réseaux siamois et triplets pour produire des représentations vectorielles de phrases (sentence embeddings) sémantiquement significatives, pouvant être comparées à l'aide de la similarité cosinus [40].
- **OpenAI Embeddings (text-embedding-ada-002, text-embedding-3)** : proposés via une API, ces modèles sont conçus pour produire des vecteurs denses à haute performance pour des cas d'usage comme le RAG. Ils sont entraînés à la fois sur des paires de documents et de requêtes, et sont multilingues [37].

1.5.4.2 Modèle Génératif

Le modèle génératif, fondé sur un LLM, est conçu pour produire des réponses cohérentes et pertinentes. Il s'appuie à la fois sur la requête de l'utilisateur et sur le contexte documentaire récupéré en amont (par exemple via FAISS). Il est capable de formuler des réponses contextualisées et adaptées aux besoins exprimés, tout en minimisant les risques de réponses génériques ou non justifiées[49].

1.5.4.3 Défis du RAG par Rapport aux Modèles Traditionnels

Bien que le RAG améliore la pertinence des réponses, cette approche soulève plusieurs défis [49] :

- La récupération des informations en temps réel peut entraîner des délais par rapport aux modèles avec réponses pré-entraînées[49].
- Si les données externes sont peu fiables, cela impacte la précision du modèle.
- La mise en place du RAG nécessite la gestion de plusieurs éléments, tels que les pipelines de récupération, l'indexation du stockage et l'intégration des API.

1.6 Travaux connexes

Cette section présente une revue des travaux existants portant sur l'intégration du Règlement Général sur la Protection des Données (RGPD) dans le contexte des projets logiciel open source (OSS). Elle met en lumière les principales approches adoptées par la recherche pour analyser la conformité au RGPD dans ces projets, les types de données étudiées (issues, commits, etc.), ainsi que les outils et méthodologies utilisés.

1.6.1 Degeling *et al.* (2019)

L'article de Martin Degeling *et al.* mesure l'impact du RGPD sur la transparence des sites web en Europe. Les auteurs ont mené une étude empirique longitudinale sur 6 759 sites web dans les 28 pays de l'UE entre décembre 2017 et octobre 2018. La méthode combine l'analyse automatique via Selenium³, suivie d'une inspection manuelle pour repérer les politiques de confidentialité et ainsi qu'une observation des bannières cookies, que les auteurs ont classées en fonction des options offertes à l'utilisateur (acceptation, refus, personnalisation, etc.). L'objectif était de mesurer les changements induits par le RGPD sur les pratiques de transparence. L'analyse a utilisé des techniques de traitement automatique du langage (NLP) pour 24 langues afin de détecter les références au RGPD. Les résultats montrent une hausse des politiques de confidentialité (+4,9%) et des bannières cookies (+16%), mais peu de sites appliquent un consentement explicite.

3. navigateur automatisé utilisé pour charger et analyser les pages web, détecter les politiques de confidentialité et les bannières cookies

Les mécanismes de consentement restent souvent incomplets ou non conformes. Les outils de consentement tiers varient fortement en termes de contrôle utilisateur et de conformité [12].

1.6.2 Abdel-Jaouad Aberkane (2021)

Dans le cadre de sa thèse, Abdel-Jaouad Aberkane (Université de Gand) propose une approche innovante pour automatiser la conformité au RGPD dans l'ingénierie des exigences (RE) à l'aide du traitement automatique du langage (TAL/NLP). L'objectif est d'intégrer le RGPD dès les premières phases du développement logiciel, en appliquant les principes de *privacy by design* et *by default*, notamment pour les PME aux ressources limitées. Sa démarche repose sur la Design Science Research Methodology, structurée en trois cycles (pertinence, rigueur, conception), permettant de concevoir des artefacts (outils d'analyse automatisée) tout en ancrant la recherche dans un problème concret. Aberkane commence par une cartographie systématique de la littérature à l'intersection du RGPD, du TAL et du RE, qui révèle un manque d'approches intégrées. Il développe ensuite des modèles d'apprentissage automatique pour analyser les politiques de confidentialité et évaluer la conformité des exigences systèmes. L'approche progresse en quatre phases : (1) cartographie des travaux, (2) évaluation de la conformité via les politiques, (3) identification des facteurs organisationnels, et (4) développement d'un modèle d'évaluation automatique. Cette recherche, à la croisée de trois domaines encore peu explorés ensemble, vise à fournir aux PME des outils concrets pour améliorer leur conformité dès les débuts des projets logiciels [1].

1.6.3 Tahaei *et al.* (2022)

Tahaei *et al.* analysent les discussions sur Stack Overflow pour comprendre comment les développeurs recherchent et fournissent des conseils en matière de confidentialité, et dans quelle mesure ceux-ci s'alignent avec les stratégies de Hoepman. L'étude porte sur 119 réponses acceptées, à partir desquelles 148 conseils ont été extraits. La majorité concernent la conformité (RGPD, CCPA) et se concentrent sur quatre stratégies : informer, masquer, contrôler et minimiser. D'autres, comme abstraire, séparer, faire respecter et démontrer, sont rarement évoquées, révélant un écart entre la recherche et la pratique. Les développeurs s'appuient souvent sur des documentations officielles ou des articles en ligne, montrant l'importance des sources formelles et informelles. Le manque de directives claires complique l'adoption de certains conseils. Les auteurs recommandent d'améliorer outils et formation, et d'étendre la recherche à des plateformes comme Reddit ou GitHub Discussions [46].

1.6.4 Sangaroonsilp *et al.* (2023)

Sangaroonsilp *et al.* proposent dans leur article une approche systématique visant à identifier, structurer et classer les exigences en matière de protection de la vie privée présentes dans

les rapports d'issues des projets logiciels (Google Chrome et Moodle). Leur objectif est de combler l'écart entre les réglementations de haut niveau (comme le RGPD, l'ISO/IEC 29100, la PDPA thaïlandaise et le cadre APEC) et les exigences concrètes de conception logicielle. Pour cela, les auteurs développent une taxonomie des exigences de confidentialité en s'appuyant sur la méthode GBRAM (Goal-Based Requirements Analysis Method), qui permet d'extraire et de raffiner les objectifs exprimés en langage naturel dans les textes réglementaires. Les exigences identifiées sont ensuite classées en 7 grandes catégories fonctionnelles (participation de l'utilisateur, notification, traitement des données, sécurité, etc.), aboutissant à une taxonomie riche de 71 exigences types. Cette taxonomie a été validée par l'analyse de 1 374 rapports d'issues issus de deux projets open-source de grande envergure (Google Chrome et Moodle), où chaque issue a été annotée manuellement pour évaluer sa conformité à la taxonomie [43].

1.6.5 Lucas Franke (2023)

Franke étudie l'impact du RGPD sur le développement logiciel, en se concentrant sur les projets open source, afin d'évaluer comment la réglementation influence les pratiques de développement. Il adopte une méthode mixte : une analyse quantitative de 998 pull requests liées au RGPD et 12217 non RGPD sur GitHub, ainsi qu'une étude qualitative fondée sur trois entretiens. Les résultats montrent que les pull requests liées au RGPD génèrent plus de commentaires, de commits et restent actives plus longtemps. Les développeurs rencontrent des difficultés liées à l'ambiguïté du RGPD et à la compréhension des exigences de conformité. L'intégration de contenus externes, comme les vidéos ou les plug-ins (ex. Google Analytics), complique le respect du règlement, notamment à cause des demandes de consentement. Certains estiment que le RGPD nuit à l'expérience utilisateur, tandis que d'autres considèrent ses principes comme du bon sens, notamment le droit à l'oubli et la gestion des données personnelles. La mise en conformité est souvent limitée par le manque de budget ou l'absence de consultation juridique [20].

1.6.6 Hennig *et al.* (2023)

Hennig *et al.* L'article étudie l'impact des règlements de la protection des données personnelles, comme le RGPD et la CCPA dans le développement des projets open source hébergés sur GitHub. L'objectif est de comprendre quels types de problèmes liés à la protection des données sont signalés, qui les signale, et comment les développeurs y réagissent. L'étude porte sur 21 608 issues provenant de 5 892 dépôts entre 2016 et 2022. Un échantillon de 650 issues est codé manuellement pour catégoriser les types de problèmes, leurs résolutions et les discussions. Les problèmes sont signalés par des acteurs internes et externes, souvent liés à la conformité, aux demandes de fonctionnalités ou aux clarifications de politique. Une analyse qualitative et des méthodes quantitatives (modèles logit multinomiaux, apprentissage automatique) permettent d'identifier les schémas de résolution. Les auteurs prévoient de publier leur jeu de données et

d'élargir leur étude à d'autres plateformes. L'étude souligne l'influence croissante des lois sur la vie privée sur le développement logiciel [27].

1.6.7 Lucas Franke *et al.* (2024) (1)

Franke *et al.* ont mené une méthodologie exploratoire fondée sur des données qualitatives concernant la conformité au RGPD dans les projets open-source, afin de mieux comprendre les perceptions et les expériences des développeurs open-source vis-à-vis de cette réglementation. Pour cela, ils ont conçu un questionnaire, diffusé auprès de développeurs identifiés via des pull requests liées au RGPD sur GitHub, ainsi que sur Twitter et Reddit. 47 développeurs y ont répondu. L'étude met en évidence plusieurs défis récurrents dans la mise en œuvre du RGPD : besoin de conseil juridique, complexité de la conception logicielle, et difficulté à évaluer la conformité. Les auteurs soulignent le besoin pressant d'outils et de ressources adaptés pour aider les développeurs OSS à se conformer à la réglementation en matière de protection des données. [18].

1.6.8 Lucas Franke *et al.* (2024) (2)

Franke *et al.* étudient l'impact du RGPD sur le développement de logiciels open source en combinant une enquête qualitative (nombre de développeurs = 56) et une analyse quantitative de 31 462 pull requests sur GitHub. Ils utilisent également une analyse de sentiment automatisée sur le contenu des pull requests. Leur objectif est de comprendre la perception des développeurs OSS, les défis techniques rencontrés et les méthodes d'évaluation de conformité. Les résultats montrent une augmentation notable de l'activité liée au RGPD, mais aussi des difficultés récurrentes : exigences floues, manque de ressources et coûts élevés. La plupart des développeurs s'auto-évaluent, évitant de consulter des experts juridiques par contrainte. L'étude met en évidence un besoin fort d'outils et de guides adaptés pour faciliter la mise en conformité dans les projets OSS [19].

1.6.9 Kapitsaki et Papoutsoglou (2024)

Kapitsaki et Papoutsoglou étudient comment le RGPD est représenté dans la communauté open source via l'analyse des messages de commits sur GitHub. Leur étude porte sur 69 905 commits effectués depuis 2016, examinant l'évolution des références au RGPD dans les commits, leur volume, la durée d'implémentation et les termes clés utilisés. En 2018, 47% des commits mentionnaient le RGPD, signe d'un pic d'adaptations cette année-là, mais leur nombre a diminué ensuite. Près de 30% des commits ont été réalisés avant l'entrée en vigueur de la loi, en anticipation. En moyenne, les modifications ont pris 100,9 jours, cependant, 63,8% des dépôts ont été modifiés en une seule journée, tandis que 6 dépôts ont nécessité plus de 1 900 jours. Enfin, les auteurs ont identifié et extrait les termes les plus couramment présents dans les messages de

commits. Pour la suite, ils prévoient d'analyser le code source pour mieux cerner la nature des changements et d'élargir leur étude à d'autres lois comme le CCPA et le CPRA [30].

1.6.10 Tableau de comparaison

Le tableau 1.4 présente une comparaison détaillée des travaux connexes, en mettant en évidence leurs objectifs, les données analysées et les méthodes utilisées.

AQ : Analyse Qualitative. AQN : Analyse Quantitative. PR : pull request. DA : Données Analysées.

Article	Année	Objectif	Source de données	DA	Nbr de DA	Méthode	Analyse de code	IA
Degeling <i>et al.</i> [12]	2019	Mesurer l'impact du RGPD sur la transparence des sites web	Sites web de pays de l'UE	Politiques de confidentialité, bannières cookies	6 759 sites	analyse automatisée, NLP	✗	✓
Aberkane [1]	2021	Intégrer le RGPD dès les premières phases du développement logiciel	Articles scientifiques	articles analysés	448	TALN, design science	✗	✓
Tahaei <i>et al.</i> [46]	2022	Étudier les conseils des développeurs sur la protection de la vie privée	Stack Overflow	Discussions	148 conseils	AQ	✗	✗
Franke [20]	2023	Évaluer l'impact du RGPD sur la communauté open source	Github, interviews	PR, développeurs	13215 PR, 3 reponses	AQ+AQN	✗	✗
Hennig <i>et al.</i> [27]	2023	Étudier l'impact de CCPA et RGPD sur le développement logiciel	Github	Issues	21 608	AQ+AQN	✗	✓
Sangaroonsilp <i>et al.</i> [43]	2023	Construire une taxonomie des exigences liées à la protection de la vie privée.	Google Chrome, Moodle	Rapports d'issues	1 374	GBRAM, taxonomie, annotation manuelle	✗	✗

Articles	Année	Objectif	Source de données	DA	Nbr de DA	Méthode	Analyse de code	IA
Kapitsaki & Papoutsoglou [30]	2024	Étudier l'évolution des mentions RGPD	Github	Messages de commits	69905	AQN	✗	✗
Franke et al. (1) [18]	2024	Comprendre l'impact du RGPD sur le développement open source	développeurs OSS	Perceptions liées au RGPD	47 réponses	AQ	✗	✗
Franke et al. (2) [19]	2024	Comprendre comment le RGPD affecte le développement open source	Github, questionnaire	PR et réponses de développeurs	31000 PR, 56 réponses	AQ+AQN	✗	✓

TABLE 1.4 – Comparaison entre les travaux

1.6.11 Synthèse des travaux ultérieurs

Plusieurs travaux s'intéressent à la conformité au RGPD dans des contextes variés, notamment les projets open source, mais avec des approches différentes selon les objectifs, les données analysées et les méthodes utilisées.

Certains travaux ont mesuré l'impact du RGPD sur les projets open-source. Franke [20] analyse des pull requests et réalise des entretiens avec des développeurs pour comprendre leur perception. Franke *et al.* [19], quant à eux, combinent des questionnaires avec une analyse automatique des pull requests à l'aide des techniques de NLP, ce qui en fait un travail mixte, à la fois déclaratif et computationnel. Hennig *et al.* [27] analysent également l'effet du RGPD, cette fois à travers un large corpus d'issues GitHub, en utilisant aussi des techniques de NLP.

D'autres travaux se concentrent sur l'analyse des éléments issues des projets logiciels : Kapitsaki et Papoutsoglou [30] explorent les messages de commits pour repérer les mentions du RGPD, en combinant une analyse quantitative automatisée basée sur des mots-clés avec une analyse qualitative manuelle du contenu. Sangaroonsilp et al. [43] développent une taxonomie à partir des issues de projets comme Chrome, par annotation manuelle. Franke *et al.* [18], quant à eux, adoptent une approche qualitative basée sur un questionnaire diffusé à des développeurs OSS afin de comprendre leurs expériences et besoins en matière de conformité au RGPD.

Sur un autre plan, Aberkane [1] propose une extraction d'exigences RGPD à partir d'articles scientifiques en utilisant le NLP. De même, Degeling *et al.* [12] utilisent l'IA pour analyser les politiques de confidentialité de sites web. Enfin, Tahaei *et al.* [46] étudient des discussions sur Stack Overflow à travers une approche qualitative, sans recours à l'automatisation.

Dans l'ensemble, ces travaux s'appuient sur des artefacts textuels (issues, commits, pull request, interviews) et n'analysent pas directement le code source. De plus, peu intègrent l'intelligence artificielle, et ceux qui le font l'utilisent uniquement pour traiter des textes.

Notre approche comble ces lacunes en combinant des LLM, des AST et une architecture RAG en analysant directement le code des projets open source. Cela permet de proposer un cadre plus précis, automatisé et interprétable pour l'analyse de conformité, rendant ainsi notre contribution originale et complémentaire aux travaux existants.

1.7 Conclusion

Dans ce chapitre, nous avons abordé la notion de protection des données personnelles dans un monde numérique en constante évolution. Qu'il s'agisse du RGPD en Europe, du CCPA en Californie ou de la loi 18-07 en Algérie, ces textes législatifs visent à redonner aux individus un certain contrôle sur leurs informations personnelles. Mais au-delà des principes juridiques, ce sont surtout les pratiques concrètes des développeurs, des entreprises et des plateformes qui comptent. L'univers open-source, avec ses milliers de projets et ses communautés engagées, joue un rôle majeur dans cette dynamique. Nous avons également présenté certaines technologies avancées d'intelligence artificielle, pouvant être utilisées dans le cadre de la fouille de logiciels afin d'analyser la conformité aux exigences réglementaires.

Dans le chapitre suivant, nous présenterons notre approche proposée pour l'analyse de la conformité et l'identification des pratiques liées à la protection des données personnelles.

Chapitre 2

Analyse de de la conformité RGPD dans le code source via une approche RAG

2.1 introduction

Dans ce chapitre, nous présentons la conception de notre étude d'analyse de conformité dans le développement logiciel. Cette approche repose sur une série d'étapes structurées, débutant par la collecte des projets logiciels jusqu'à l'analyse fine de leur code source. Nous décrivons ensuite les deux approches principales que nous avons mises en place pour la classification automatique : l'analyse des commits d'une part, et l'analyse des descriptions de projets et fichiers README d'autre part. Nous abordons également la technique utilisée pour représenter le code source, à savoir l'AST (Abstract Syntax Tree). Enfin, nous introduisons la conception de l'approche RAG (Retrieval-Augmented Generation) appliquée à l'analyse de code.

2.2 Processus de notre travail

Le processus que nous avons mis en œuvre pour analyser la conformité RGPD des projets open source s'appuie sur plusieurs étapes clés, représentées dans le schéma 2.1 :

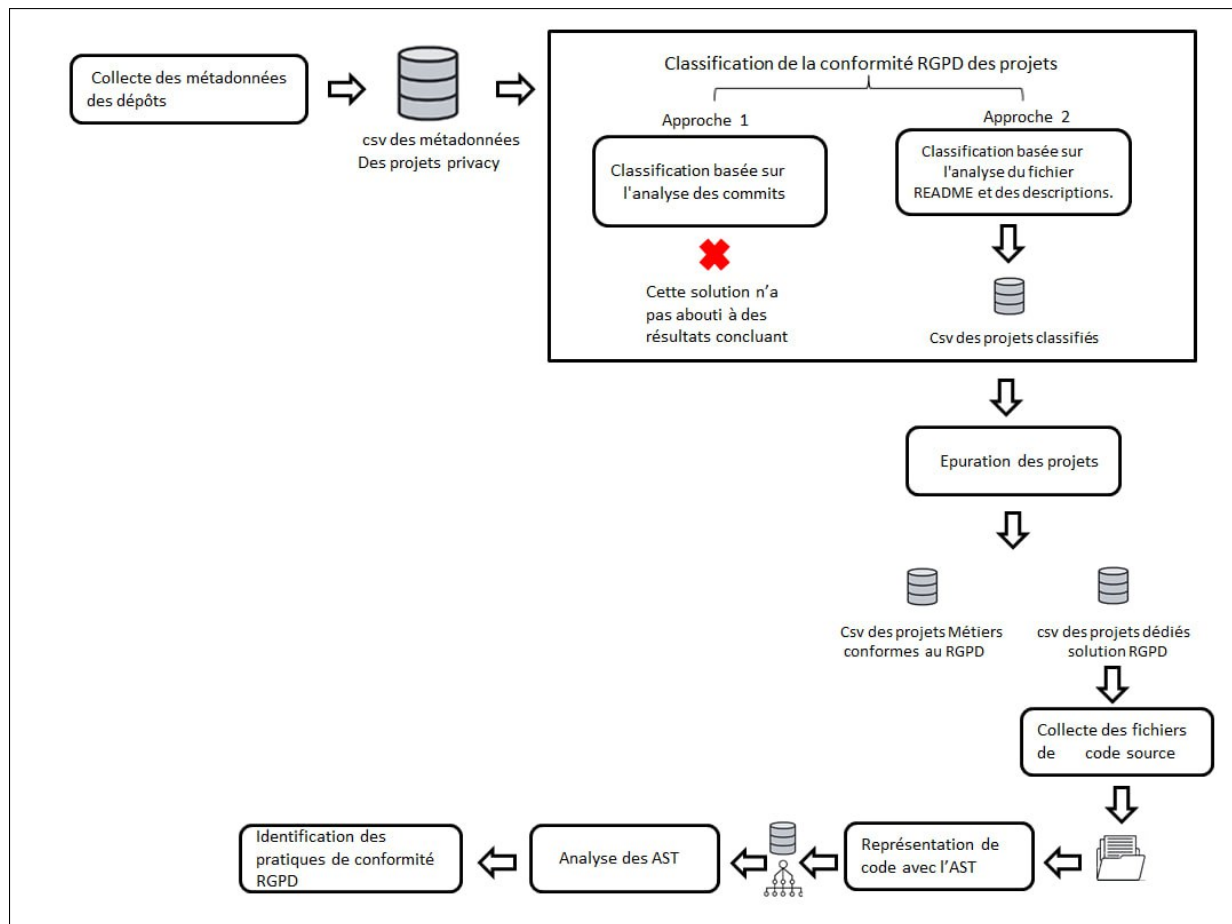


FIGURE 2.1 – Processus d’analyse de la conformité RGPD des projets open source

2.3 Collecte des projets

L’identification des projets open source intégrant des pratiques liées au Règlement Général sur la Protection des Données (RGPD) représente une étape essentielle dans le cadre de cette étude.

Comme il n’existe pas de dataset prêt à l’emploi, nous avons dû concevoir un processus spécifique pour collecter et préparer les projets concernés. L’organigramme 2.2 présente le processus de collecte des projets :

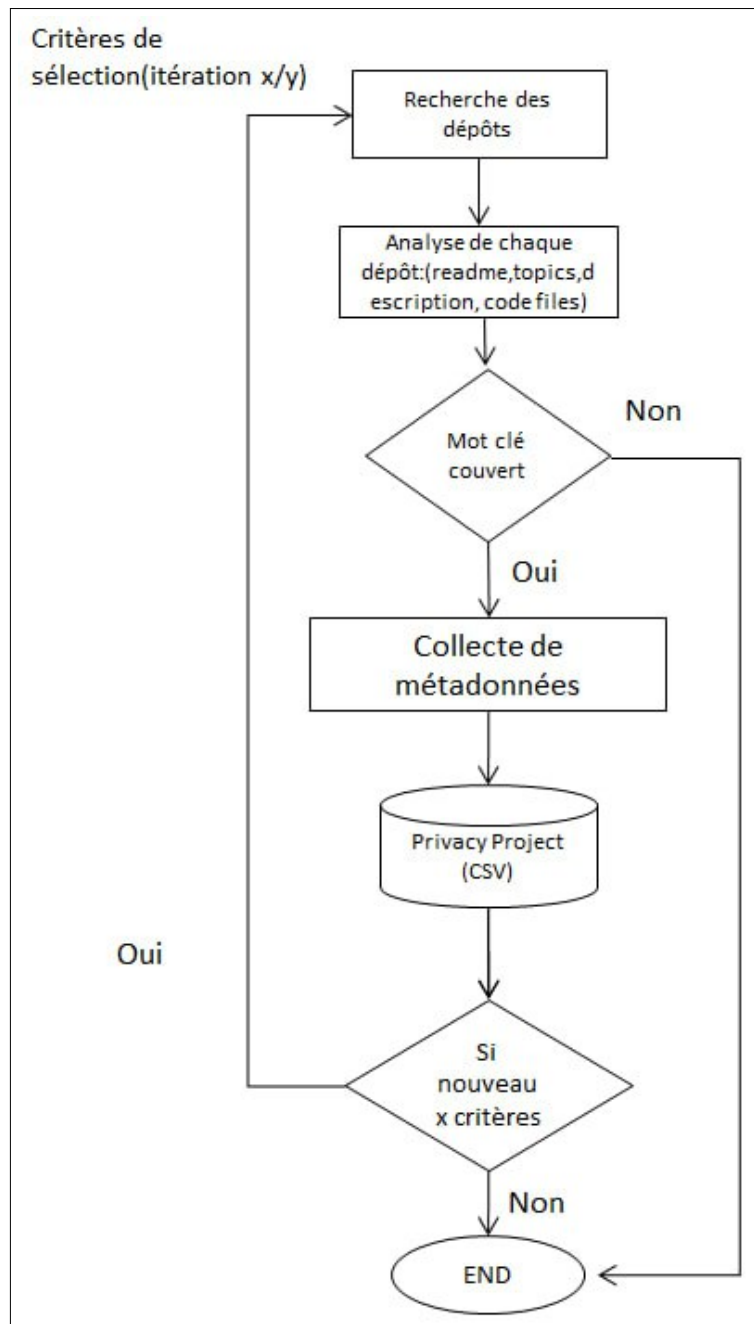


FIGURE 2.2 – Processus de collecte des projets

Pour collecter un maximum de projets en lien avec le RGPD, nous mettons en place une démarche itérative de collecte, reposant sur deux listes de mots-clés liées à la protection des données privées utilisées dans chaque recherche lancée.

Une analyse est ensuite effectuée sur les différents éléments présents dans les dépôts : les fichiers racine du code source des projets, la documentation accompagnant les projets (le fichier read-me ou le manuel d'utilisation s'il existe), la description du projet ainsi que les "topics".

Cette démarche nous permet d'identifier à la fois les projets évoquant explicitement le RGPD lorsque des mots-clés directement associés au RGPD comme "RGPD" ou "GDPR" (la première liste utilisée pour la recherche) sont présents dans le dépôt, et ceux dont la conformité est suggérée

de manière plus implicite en utilisant des termes plus généraux en lien avec la protection des données personnelles, tels que "privacy", "compliance" ou encore "data protection" (mots-clés ajoutés à la première liste pour construire une liste plus complète). Ces mots-clés, bien que moins spécifiques, reflètent souvent des préoccupations ou des mécanismes conformes au RGPD.

Une fois les projets identifiés, les métadonnées illustrées dans le tableau 2.1 sont extraites afin d'enrichir l'analyse.

Métadonnée	Exemple
repo_name	cookieconsent
repo_url	github.com/osano/cookieconsent
description	A free solution to the EU, GDPR, and California Cookie Laws
stars	3437
language	JavaScript
forks	576
open_issues	93
pull_requests	45
commits	693
license	MIT
collaborators	43
created_at	2015-02-02 17 :55 :24+00 :00
updated_at	2025-02-09 13 :52 :49+00 :00
topics	cookie-consent, eu-cookie, javascript, cookies, gdpr, legal, ccpa
selection_criteria	description, readme
keyword_used	GDPR

TABLE 2.1 – Un exemple des métadonnées enregistrées

2.4 Classification de la conformité RGPD des projets

Lors de l'étape précédente, la collecte des projets s'est appuyée sur la détection de mots-clés contenus dans les éléments des dépôts (README, descriptions, etc.) et liés généralement au RGPD. Cependant, cette collecte n'est pas toujours fructueuse car la présence de ces mots-clés peut nous conduire à la récupération de projets traitant seulement la protection des données de manière générale, sans pour autant être spécifique au RGPD ou encore n'ayant aucun lien avec la confidentialité et la protection des données personnelles, les mots-clés pouvant être mentionnés dans un autre contexte, voire pour signaler explicitement l'absence de lien avec ces mots-clés (ex. : "ce projet n'a aucun rapport avec le RGPD").

Pour affiner notre collecte et épurer cette sélection, nous allons classifier les différents projets collectés en menant une analyse approfondie et sémantique fondée sur les Grands Modèles de Langage (Large Language Model). Cette analyse constitue une phase de classification automa-

tique, elle peut se baser sur les commits ou encore sur les fichiers annexes des projets (readme, description, etc.).

Pour assurer un meilleur rendement (résultat), nous essayerons de proposer une analyse avec les commits et une autre avec les fichiers annexes aux projets. Les deux approches seront détaillées dans cette section.

2.4.1 Classification basée sur l'analyse des commits

Nous collectons d'abord les commits d'un échantillon de projets issus de la collecte initiale. L'analyse des commits permet d'étudier les évolutions des projets, car ils reflètent les modifications apportées au code.

Nous effectuons ensuite une classification des commits afin de filtrer ceux directement liés à la mise en conformité avec le RGPD, tout en écartant ceux sans rapport avec cette réglementation. Cette classification comprend également un regroupement des commits selon des catégories conceptuelles liées au RGPD (par exemple, gestion du consentement, sécurité des données, droit d'accès), permettant ainsi une meilleure compréhension des pratiques concrètes mises en œuvre dans le code.

Limites :

Malgré sa pertinence, cette approche présente plusieurs limites. D'une part, la forte présence de commits non pertinents, ambigus ou mal formulés rend leur exploitation difficile. D'autre part, le fait de fonder l'analyse sur un nombre restreint de commits (par projet) peut biaiser la compréhension globale en réduisant l'observation à certains concepts ou pratiques spécifiques, sans refléter l'ensemble des efforts de mise en conformité.

Donc une exploration plus approfondie des projets eux-mêmes peut être plus intéressante pour l'étude.

2.4.2 Classification basée sur l'analyse des fichiers README & description

Une analyse préliminaire manuelle est effectuée, basée sur la lecture attentive des fichiers de documentation README et des descriptions de projet, considérés comme des éléments représentatifs. Ces contenus textuels permettent d'accéder directement à l'intention déclarée des développeurs, à la finalité des projets, ainsi qu'aux fonctionnalités générales. À partir de cette analyse, plusieurs typologies de projets ont pu être identifiées, fournissant des indications précieuses sur les finalités, les fonctionnalités et les intentions de projet :

- **Projets dédiés aux solutions RGPD :** Ces projets sont spécifiquement conçus pour répondre aux exigences du RGPD, comme des bibliothèques de gestion de la confidentialité ou des outils de conformité. Une analyse approfondie du code source est nécessaire pour identifier les pratiques et les implémentations concrètes.

- **Projets métiers intégrant le RGPD** : Il s'agit de projets ayant une finalité métier (applications web, services, etc.) qui incluent également des pratiques de conformité au RGPD. Ce type de projet aborde implicitement la protection des données sans toujours mettre l'accent sur la conformité réglementaire. Dans ce cas, l'analyse des commits s'avère particulièrement pertinente pour suivre les ajustements et améliorations liés à la conformité réglementaire.
- **Projets axés sur la protection de la vie privée en général** : Ces projets se concentrent sur la confidentialité et la sécurité des données de manière générale, sans traitement explicite des concepts RGPD.

Ces premières observations justifient un approfondissement de l'analyse sémantique des fichiers README et des descriptions associées aux projets collectés. L'objectif est d'aller au-delà de la simple détection de mots-clés, en évaluant de manière plus fine la relation réelle entre le projet et les exigences du RGPD. Cette approche permet notamment de distinguer les projets ayant mis en œuvre des pratiques conformes au RGPD de ceux qui ne font qu'évoquer le RGPD de manière opportuniste, parfois sans véritable implémentation technique. Elle contribue ainsi à épurer la base de projets, en excluant ceux qui se réfèrent de manière vague ou détournée à la protection des données personnelles, sans implication directe avec la réglementation.

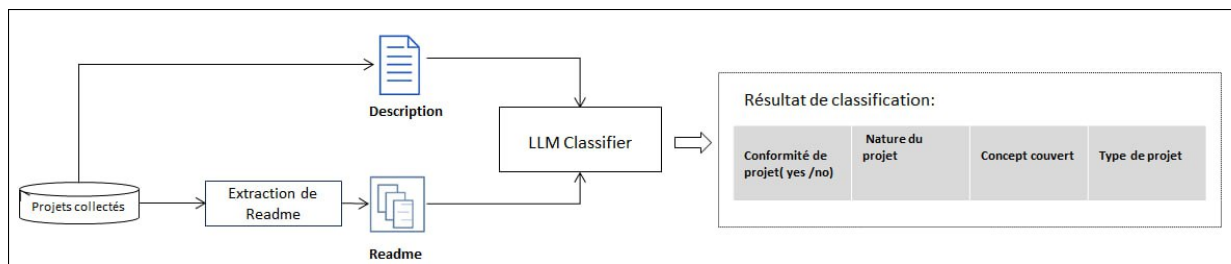


FIGURE 2.3 – Processus de classification des projets à l'aide d'un modèle LLM, basé sur le README et la description

Afin d'automatiser cette phase d'analyse sémantique à l'échelle de l'ensemble des projets collectés, nous avons conçu un pipeline de classification automatique s'appuyant sur un LLM. Cette chaîne de traitement présentée à la figure 2.3 illustre les étapes générales de cette phase. Les descriptions de projets ainsi que les fichiers README sont extraits automatiquement et servent d'entrée au classifieur. Ces éléments textuels, considérés comme représentatifs des finalités et des intentions des projets, sont ensuite analysés par le LLM. Ce dernier permet d'évaluer la conformité du projet au RGPD, de déterminer sa nature, les concepts de protection des données abordés, ainsi que son type (outil, bibliothèque, service, etc.). Cette étape joue donc un rôle essentiel dans la réduction du périmètre d'analyse technique, en ciblant les projets dont le contenu mérite une étude approfondie du code source, notamment ceux identifiés comme solution dédiées au RGPD. Afin d'identifier les pratiques concrètes liées à la protection des données personnelles.

2.5 Épuration des projets

Parmi les différentes catégories de projets identifiées, ceux portant sur les solutions RGPD apparaissent comme les plus pertinents pour une analyse du code source, dans la mesure où ils sont spécifiquement conçus et largement développés selon des pratiques axées sur la conformité réglementaire.

2.6 Analyse du code source

Cette section présente le processus que nous avons suivi pour analyser le code source des projets dédiés aux solutions RGPD et extraire les pratiques liées à la protection des données privées.

2.6.1 Collecte des fichiers de code source

Au cours de cette étape, nous avons procédé à la sélection de plusieurs projets et à la collecte des fichiers de code source correspondants. Ces projets ont été choisis en fonction de leur mise en œuvre de pratiques de conformité diversifiées et de leur utilisation de différents langages de programmation, afin de garantir une variété représentative dans l'analyse des pratiques étudiées.

2.6.2 Représentation des fichiers de code

L'analyse automatisée du code source nécessite d'abord une étape de prétraitement, au cours de laquelle les fichiers doivent être préparés. Toutefois, se baser uniquement sur une représentation textuelle du code pose plusieurs défis majeurs. Cette forme ne donne pas accès directement à la structure logique du programme comme les fonctions, les classes ou les attributs et elle varie considérablement selon le langage de programmation, ce qui rend l'analyse multilingue plus complexe et exhaustive.

Pour dépasser ces limites, nous avons opté pour une représentation structurée du code, plus adaptée à l'analyse fine. Cette transformation permet de capturer la structure syntaxique et logique du code, ce qui facilite sa manipulation, son analyse, et l'extraction ciblée d'éléments pertinents.

Nous avons ainsi choisi d'utiliser les arbres de syntaxe abstraite (AST pour Abstract Syntax Trees) comme représentation intermédiaire du code, afin de structurer l'information et d'enrichir notre analyse [50].

Contrairement à une approche basée sur texte brut, l'AST exploite la structure syntaxique et sémantique des langages de programmation, permettant ainsi de capturer des relations plus riches entre les éléments du programme. L'analyse de ces AST s'est faite selon plusieurs étapes, comme illustré par la figure 2.4 :

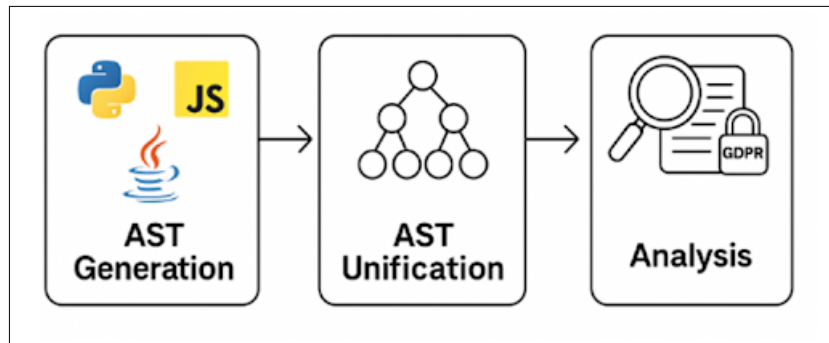


FIGURE 2.4 – Génération, unification et analyse des AST

2.6.2.1 Génération de l'AST

Pour chaque projet, nous avons généré un AST afin de représenter de manière structurée et formelle le code source. Cette étape nous permet d'accéder à la structure syntaxique du programme (fonctions, classes, instructions, etc.) de manière plus lisible et exploitable. Il est important de noter que les AST sont propres à chaque langage de programmation, car ils dépendent des règles syntaxiques et sémantiques spécifiques à ce langage. Ainsi, la génération d'AST a été réalisée à l'aide d'outils adaptés pour chaque langage utilisé dans les projets analysés.

2.6.2.2 Unification de l'AST

Les AST générés pour chaque projet présentent des structures différentes selon le langage de programmation utilisé. Chaque langage ayant ses propres spécificités syntaxiques et sémantiques, les arbres produits ne sont pas directement exploitables de manière uniforme. Cette hétérogénéité constitue un obstacle majeur à l'analyse des projets multi-langages. Face à cette problématique, nous avons conçu une stratégie d'unification des AST visant à produire une représentation commune des éléments syntaxiques, ce qui permet une manipulation homogène, en un format commun lors de l'analyse automatisée quelle que soit le langage de programmation .

L'unification réalisée consiste à extraire, pour chaque langage, un ensemble commun d'éléments clés (classes, méthodes, fonctions, attributs, etc.) et à les représenter dans un format homogène. Les informations spécifiques à un langage sont conservées lorsqu'elles sont disponibles. Dans le cas contraire, elles sont laissées à null afin de garantir une structure de données cohérente entre les projets.

Cette étape rend possible l'analyse automatique sur des corpus multi-langages en uniformisant les représentations et facilite l'étape suivante, qui consiste à manipuler les AST pour plusieurs tâches.

2.6.3 Analyse des AST : LLM

Cette étape consiste à analyser les AST des projets afin d'identifier les pratiques liées à la protection des données privées. Il s'agit d'une phase cruciale, car la pertinence des résultats

conditionne directement la validité des conclusions formulées par la suite.

Dans un premier temps, nous avons envisagé d'utiliser directement les AST comme entrée dans le prompt d'un LLM. Toutefois, cette approche s'est rapidement révélée limitée pour plusieurs raisons :

- **Limites de la fenêtre de contexte :** Les représentations textuelles des AST peuvent être particulièrement volumineuses, surtout lorsqu'il s'agit de projets de grande taille. Cela pose un véritable défi, car les grands modèles de langage (LLM) ne peuvent traiter qu'une quantité limitée d'informations à la fois et leur capacité contextuelle restreinte nous oblige donc à être sélectifs dans les données que nous leur soumettons dans une seule requête.
- **Pertinence non garantie :** Lorsque l'on soumettait un AST complet au modèle, celui-ci avait parfois du mal à identifier les parties vraiment utiles pour répondre à une requête précise. Cela pouvait réduire la pertinence des résultats, en masquant les éléments réellement pertinents dans un volume trop important de données.

Face à ces limitations, nous avons opté pour une approche plus robuste et évolutive : le RAG (Retrieval Augmented Generation). Il s'agit d'un processus dans lequel des informations pertinentes sont récupérées depuis des bases de connaissances externes avant d'être utilisées pour enrichir le prompt adressé au LLM.

Ce paradigme a démontré son efficacité en améliorant significativement la précision des réponses, en réduisant les hallucinations des modèles, notamment dans les tâches à forte densité de connaissances [21].

2.6.4 Analyse des AST : Approche de RAG

Le schéma 2.5 illustre l'architecture du pipeline RAG depuis la construction de la base de connaissances jusqu'à la génération de réponses enrichies.

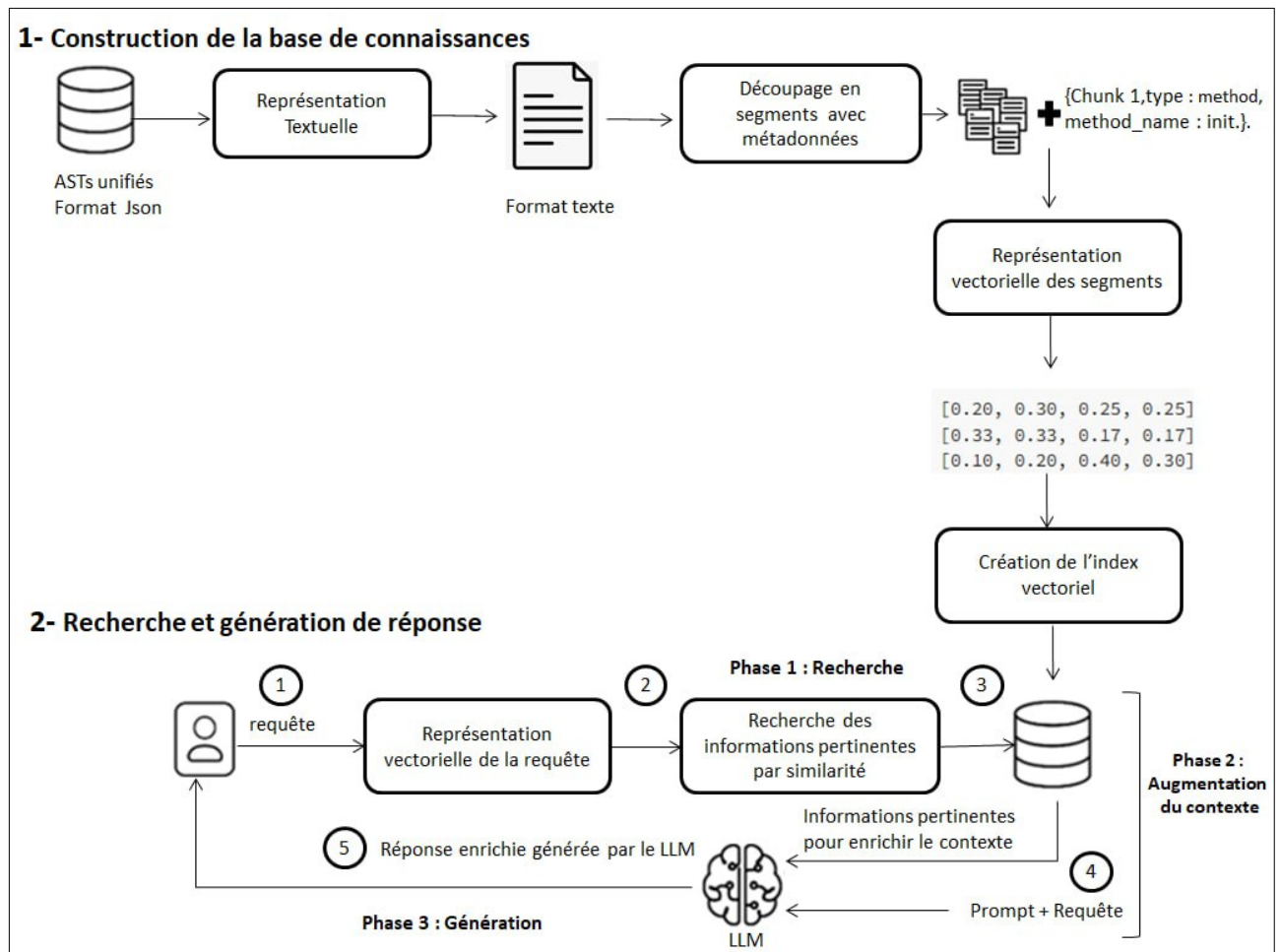


FIGURE 2.5 – Architecture du pipeline RAG

Ce processus est divisé en deux grandes phases :

Phase 1 : Construction de la base de connaissances (Offline Processing)

Cette phase est effectuée avant toute interaction utilisateur. Dans notre cas, les AST unifiés constituent la matière première de notre base de connaissances. Ces AST sont transformés en représentations textuelles structurées qui sont ensuite vectorisées et stockées. Les étapes de ce processus sont détaillées ci-dessous :

1. Représentation textuelle

Cette étape vise à transformer les AST en un format textuel compatible avec les modèles utilisés pour l'analyse dans les étapes suivantes. À partir des fichiers unifiés, nous avons converti les différents éléments de l'AST en texte selon une approche différenciée :

- **Éléments complexes et riches en logique :**

Certains éléments, comme les blocs de code contenant une forte densité sémantique, sont conservés sous forme brute, mais formatés de manière lisible (avec indentation et sauts de ligne préservés). Cette préservation est essentielle pour maintenir l'intégralité de l'information sémantique et permettre une analyse fine de la logique du programme.

- **Éléments simples ou structurellement redondants :**

D'autres composants, dont la structure hiérarchique ne fournit pas de valeur ajoutée significative pour l'analyse, sont convertis en texte clair et structuré. Cette représentation simplifiée permet de réduire la complexité tout en conservant les informations essentielles à la compréhension du code.

Cette approche hybride maximise la conservation d'informations pertinentes tout en optimisant la représentation pour les modèles utilisés ensuite. Nous avons privilégié une représentation riche uniquement lorsqu'elle apporte une réelle valeur ajoutée à l'analyse du comportement du programme.

2. Découpage en segments (chunking)

La segmentation consiste à découper de grands documents ou ensembles de données en unités plus petites et plus faciles à manipuler, appelées segments (chunks). Cette étape est essentielle dans les systèmes de RAG où ces segments sont transmis à un modèle de recherche (avec un format compatible), permettant à ce dernier de localiser efficacement les informations pertinentes. En effet, les représentations internes comme les AST sont souvent trop volumineuses pour être encodés directement dans un vecteur dense (embedding). Cela peut dépasser les limites de contexte imposées par les modèles et entraîner une perte de détails significatifs. En divisant les documents en segments plus petits tels que des paragraphes, des blocs de code ou des sections fonctionnelles le système gagne en précision et peut mieux localiser les passages les plus pertinents lors d'une requête [42].

- **Stratégie de découpage basée sur la structure syntaxique**

Une méthode efficace de découpage consiste à exploiter la structure hiérarchique fournie par les AST. Ces arbres représentent les éléments clés d'un code source sous forme de nœuds bien identifiés, tels que les classes, les méthodes, les attributs, etc. Cette organisation logique rend possible un découpage intelligent, centré sur ces unités fonctionnelles. Ainsi, chaque segment extrait garde une cohérence sémantique forte, ce qui améliore considérablement la lisibilité et la pertinence des réponses générées par les systèmes RAG [44].

- **Contrôle de la granularité via les limites de tokens**

Afin de garantir la compatibilité avec les modèles de représentation vectorielle et d'éviter les dépassements de capacité, une limite maximale de tokens par segment est souvent imposée. Ce seuil, ajustable selon les besoins (par exemple 512 ou 1024 tokens), permet de générer des segments à la fois informatifs et exploitables. Il n'existe cependant pas de taille de segment optimale universelle. Le choix dépend de plusieurs facteurs : la nature du contenu (texte long ou court), les capacités du modèle de représentation vectorielle (plus ou moins à l'aise avec des textes longs), la complexité attendue des requêtes (précises ou exploratoires), et les objectifs de l'application (extraction ciblée vs compréhension globale). Un segment trop court pourrait manquer de sens, tandis qu'un segment trop

long risquerait de dépasser la capacité du modèle. Cette contrainte aide donc à trouver un équilibre entre compacité et richesse contextuelle [31].

À chaque segment sont associées des métadonnées permettant de l'identifier, telles que le type de segment (par exemple : une méthode, un champ), le fichier dans lequel il se trouve, etc.

3. Représentation vectorielle (Embedding)

Chaque segment est converti en un vecteur dense, une représentation numérique visant à capturer la signification sémantique du texte. Ces vecteurs sont généralement produits par des modèles d'apprentissage automatique entraînés sur de vastes corpus textuels, constituant la « mémoire » du système RAG. Cela permet au système de retrouver des informations pertinentes en se basant sur le contexte, et non simplement sur des mots-clés, même lorsque les requêtes utilisent des formulations différentes.

4. Indexation

Après la génération des embedding, ces derniers sont extraits et indexés dans une base de données vectorielle comme FAISS. Cette étape associe chaque vecteur à son contenu d'origine afin de permettre, lors de la phase de requête, de retrouver les éléments les plus pertinents selon la similarité sémantique avec une requête donnée [33].

Exemple d'architecture de l'indexation dans FAISS

— Association "ID-Embedding" dans FAISS

FAISS maintient une association fondamentale entre des identifiants numériques et des vecteurs embedding. Chaque vecteur reçoit un ID séquentiel lors de son insertion :

ID_FAISS <-> Vecteur d'Embedding

0 <-> [0.2, -0.1, 0.8, ...]

1 <-> [0.5, 0.3, -0.2, ...]

2 <-> [-0.1, 0.7, 0.4, ...]

...

— Association ID-Métadonnées

Parallèlement, une structure de données (dataset) associe les mêmes ID aux métadonnées complètes :

Position Dataset <-> Métadonnées Complètes

0 <-> {text : "...", metadata : {file : "...", class : "..."} }

1 <-> {text : "...", metadata : {file : "...", method : "..."} }

2 <-> {text : "...", metadata : {file : "...", type : "..."} }

...

Techniques d'indexation

Plusieurs techniques existent, offrant différents compromis entre précision, vitesse et utilisation de la mémoire :

- **Flat Indexing**

Flat indexing consiste à stocker chaque vecteur tel quel. Lors d'une recherche, le système compare le vecteur de requête à tous les vecteurs de l'index, ce qui garantit une bonne précision, mais au prix d'un temps de calcul élevé pour des grands corpus. Cette approche est donc adaptée lorsque l'on cherche la meilleure précision possible et que la taille du jeu de données modérées, ou que la vitesse n'est pas une contrainte majeure.

- **IndexFlatIP**

Avec des vecteurs normalisés permet une recherche basée sur la similarité cosinus, équivalente au produit scalaire. Cela simplifie et accélère considérablement les calculs dans des systèmes d'indexation comme FAISS [10].

- **IndexFlatL2**

Mesure la similarité entre vecteurs en calculant la distance euclidienne. Plus la distance est faible, plus les vecteurs sont similaires. Utilisé quand la norme des vecteurs est significative [10].

- **LSH (Locality Sensitive Hashing)**

LSH est une méthode plus rapide qui repose sur une fonction de hachage conçue pour que des vecteurs proches soient envoyés dans le même "bucket". Lors d'une requête, le vecteur est haché, et la recherche s'effectue uniquement parmi les vecteurs du même bucket, ce qui réduit fortement le nombre de comparaisons. Cela rend la recherche beaucoup plus rapide que l'approche flat, mais moins précise, car on risque de rater des vecteurs proches qui sont tombés dans d'autres buckets [10].

- **Inverted File Indexing (IVF)**

Inverted File Indexing (IVF) suit un principe similaire à LSH, mais utilise le regroupement (clustering souvent via K-means) pour diviser l'espace des vecteurs en centroïdes. Chaque vecteur est affecté à un cluster. Lors d'une recherche, le vecteur de requête est comparé au centroïde le plus proche, puis à tous les vecteurs de ce cluster uniquement. Cela permet d'accélérer la recherche tout en restant plus souple que LSH. Cependant, si la requête est proche de la frontière entre deux clusters, il peut être nécessaire de chercher dans plusieurs clusters pour éviter des erreurs de correspondance [10].

Phase 2 : Recherche et génération de réponse (Online Querying)

Cette phase se déclenche lorsqu'un utilisateur soumet une question au système. Elle repose sur une série d'étapes clés visant à fournir une réponse contextualisée et pertinente à partir de la base de connaissances existante.

1. Requête – Représentation vectorielle de la question

La question formulée (peut cibler une pratique de RGPD) par l'utilisateur est d'abord transformée en un vecteur à l'aide du même modèle d'encodage sémantique que celui utilisé pour la base de connaissances. Cette cohérence dans le choix du modèle est essentielle pour garantir que la question et les documents soient projetés dans le même espace vectoriel.

2. Recherche – Recherche des segments similaires

Le vecteur de la question est comparé par similarité à l'ensemble des vecteurs présents dans la base de données, implémentée efficacement par l'index. Le système sélectionne ensuite les k segments les plus proches du vecteur de la requête (par exemple en identifiant les segments AST les plus proches de la pratique demandée), un choix judicieux de k est essentiel pour assurer un bon équilibre entre :

- Une couverture contextuelle suffisante (k trop petit = information manquante)
- Une précision ciblée (k trop grand = bruit et dilution du signal)
- Une taille de prompt raisonnable pour le LLM lors de la génération de la réponse.

Une fois les k vecteurs identifiés, leurs segments textuels associés sont récupérés, accompagnés de leurs métadonnées.

3. Augmentation – Enrichissement du prompt

Les segments récupérés sont ensuite traités pour former un contexte cohérent :

- a) Tri basé sur la pertinence (score de similarité) .
- b) Intégration des segments afin d'y ajouter des informations contextuelles.

Ce contexte enrichi est utilisé pour construire un prompt structuré qui sera envoyé au modèle de langage, accompagné de la question de l'utilisateur. Le prompt contient généralement :

- Une instruction claire sur la tâche à accomplir.
- Les segments pertinents.
- La requête de l'utilisateur.
- Des directives sur le format de réponse attendu.

4. Génération – Production de la réponse par le modèle

Enfin, le prompt enrichi est transmis au LLM, qui génère une réponse basée sur le contexte fourni et ses connaissances générales. Grâce à l'ajout de contexte pertinent, la réponse produite est plus précise, cohérente et directement adaptée à la demande de l'utilisateur (La réponse peut contenir des exemples d'implémentation de la pratique demandée dans la requête), tout en étant capable de citer les sources spécifiques issues de la base de code.

2.7 Conclusion

Dans ce chapitre, nous avons présenté les étapes clés de notre approche d'analyse de conformité dans le développement logiciel. Deux méthodes ont été explorées : l'analyse des commits, qui n'a pas permis de répondre à nos besoins d'analyse, et l'analyse de la documentation des projets, plus pertinente. Cette dernière nous a permis d'identifier deux types de projets : ceux intégrant des pratiques de conformité et ceux explicitement dédiés à la conformité RGPD. Nous avons choisi de nous concentrer sur ces derniers, car ces projets ciblent directement les problématiques de protection des données privées. Nous avons également justifié l'usage de l'AST pour représenter le code source, facilitant ainsi l'analyse automatique, et conclu par la présentation de l'approche RAG, utilisée comme solution finale pour extraire les pratiques de conformité. Dans le prochain chapitre, nous allons implémenter et mettre en œuvre les solutions proposées, en détaillant les outils utilisés et les choix techniques effectués.

Chapitre 3

Implémentation du système GDPR assistant

3.1 Introduction

Dans ce chapitre, nous passons à la mise en œuvre concrète de l'approche d'analyse de conformité présentée précédemment. Nous détaillons le choix des outils, bibliothèques, API et environnements de développement, ainsi que les décisions prises en matière de traitement des données, d'architecture du système et d'intégration des modèles de langage. Nous présentons également les pipelines mis en place et les ajustements nécessaires pour assurer la fiabilité et l'efficacité de l'ensemble du processus.

3.2 Environnement de Développement

Dans cette section, nous allons présenter le processus, les outils, les bibliothèques ainsi que les spécifications de l'environnement de développement utilisés pour la mise en œuvre de notre système.

3.2.1 Environnement matériel

Le projet a été réalisé sur un ordinateur portable sous Windows 11 Professionnel, équipé d'un processeur Intel Core i5-8350U (1,70–1,90 GHz) et de 8 Go de RAM. Cette configuration milieu de gamme permet de mener efficacement les tâches courantes de développement et de tests.

3.2.2 Environnement logiciel

Dans le tableau 3.1 ci-dessous, nous listons les outils, langages et les APIs utilisés pour créer notre solution ainsi que leur description.

Outils et API	Description
Python 	Python est un langage de haut niveau interprété, orienté objet. Nous l'avons choisi pour sa syntaxe flexible, son typage dynamique et ses types de données puissants, qui le rendent idéal pour développer rapidement des applications et intégrer divers composants logiciels. C'est l'un des langages les plus utilisés dans le domaine de l'intelligence artificielle [17].
Google Co-laboratory 	C'est un notebook Jupyter hébergé dans le cloud, permet de coder et d'exécuter Python dans le navigateur, sans configuration, avec accès gratuit (limité) aux GPU et un partage facile [25].
Google Drive 	Un service de stockage basé sur le cloud développé par Google. adapté à tous les collaborateurs et aux équipes de toutes tailles. Il permet d'importer, ouvrir, partager et modifier plusieurs types de fichiers à partir de n'importe quel appareil [26].
Pandas 	Pandas est une bibliothèque open source conçue pour la manipulation et l'analyse de données. Elle est reconnue pour sa rapidité, sa polyvalence et sa facilité d'utilisation, et elle s'appuie sur le langage de programmation Python [15].
Numpy 	NumPy est la bibliothèque fondamentale pour le calcul scientifique en Python. Elle offre des tableaux multidimensionnels puissants, des fonctions mathématiques avancées, et des outils pour l'intégration de code C/C++ ou Fortran [13].
Faiss	Faiss (Facebook AI Similarity Search) est une bibliothèque pour la recherche de similarité et le regroupement de vecteurs denses. Elle propose des algorithmes capables de traiter des ensembles de vecteurs, même très grands [14].
PyGithub	PyGithub est une bibliothèque Python permettant d'utiliser l'API Github v3. Elle permet de gérer les ressources GitHub (dépôts, profils utilisateurs, organisations, etc.) directement depuis des scripts Python [29].
Sentence Transformer	Est une bibliothèque Python permettant de transformer des phrases, des paragraphes ou des documents en vecteurs numériques appelés embeddings. Ces vecteurs capturent le sens sémantique du texte, de sorte que les textes similaires soient représentés par des vecteurs proches dans l'espace vectoriel. La bibliothèque est basée sur les modèles Transformers et facilite des tâches telles que la recherche sémantique [41].
GitHub API 	Est une interface qui permet d'interagir de manière automatisée avec GitHub. Elle offre des fonctionnalités essentielles pour accéder aux dépôts, récupérer des commits, analyser les contributions, et bien plus encore. Elle impose une limite de 5 000 requêtes par heure pour les utilisateurs authentifiés (rate limite) ¹ [22].
MistralAI API 	Est une interface de programmation qui permet aux développeurs d'intégrer les modèles avancés de Mistral AI dans leurs workflows de production. Elle donne accès à des fonctionnalités telles que la génération de texte, les embeddings et la génération de code [2].

TABLE 3.1 – Description des outils et des APIs utilisés

1. Rate Limit : Une restriction qui limite le nombre total de requêtes pouvant être effectuées par une application ou un utilisateur sur une période donnée.

3.3 Collecte des projets GitHub

Nous avons collecté les projets à partir de la plateforme GitHub, étant une plateforme très populaire pour les projets open source et qui dispose d'une API facilitant la collecte des projets, tout en choisissant des critères de sélection selon nos besoins. Le processus de collecte a été conçu en trois étapes : une recherche visant à élargir progressivement le périmètre des projets tout en assurant leur pertinence, une analyse du contenu des dépôts, puis une collecte des projets jugés pertinents. Le tableau 3.2 illustre le processus de collecte :

Phase	Description	Critères de sélection	Éléments d'analyse	Résultats	Limites
Phase 1 Recherche initiale restreinte aux projets à forte visibilité	Les projets populaires (>1000 étoiles) sont mieux documentés et susceptibles d'implémenter le RGPD	- 1ère Liste de mots-clés : RGPD, GDPR - Opérateur logique OU - Nombre d'étoiles > 1000	Topics des projets uniquement	30 projets identifiés	- Biais de représentativité (projets moins étoilés mais pertinents exclus) - Recherche limitée aux topics - Problème de mots-clés non explicites
Phase 2 Recherche élargie	Visé à dépasser les limites de la phase 1 en élargissant le périmètre de recherche	- Suppression du critère de popularité - 2ème Liste de mots-clés étendus : RGPD, GDPR, General Data Protection Regulation, privacy, data protection, compliance	- README - Description - Fichiers de code root	~400 projets identifiés (avant interruption)	Interruption due au rate limit de l'API GitHub
Phase 3 Stratégie par intervalles d'étoiles	Permet la poursuite de la collecte sans dépasser les limites de l'API	Tranches d'étoiles (ex. 400-600, etc.) Mêmes mots-clés que la phase 2	Mêmes éléments d'analyse que la phase 2	+6200 projets collectés	Temps de collecte important

TABLE 3.2 – Processus de collecte des projets GitHub liés au RGPD

Pour chaque phase, l'API recherche des projets répondant à certains critères (étoiles, mots-clés). Ensuite, pour chaque dépôt trouvé, une analyse est effectuée pour vérifier la présence des mots-clés dans différents éléments du dépôt (topics, README, description, fichiers source). Lors de cette analyse, chaque projet a également été annoté selon deux critères :

- L'emplacement du mot-clé lié au RGPD (dans les topics, le README, la description ou le code).
- Le mot-clé déclencheur de la sélection (ex. "RGPD", "GDPR", "privacy", etc.).

Cette annotation permet d'évaluer à la fois la visibilité du mot-clé dans le dépôt et l'intention de son utilisation, qu'il s'agisse d'une simple mention indicative ou d'une réelle mise en œuvre de pratiques de protection des données.

Enfin, les métadonnées des projets identifiés sont enregistrées dans un fichier CSV. Ce fichier constitue une base de données de projets liés à la protection des données personnelles, utilisable pour des analyses ultérieures.

Sélection des projets : exclusion de la tranche [0–200] étoiles

Afin de garantir la qualité et la pertinence des projets analysés, nous avons choisi d'exclure les dépôts GitHub ayant un nombre d'étoiles compris entre 0 et 200. Bien que cette tranche puisse contenir certains projets intéressants, une analyse préliminaire a révélé qu'elle contenait de nombreux dépôts non pertinents, souvent bruyants, peu actifs, expérimentaux ou à usage personnel. Comme le soulignent Kalliamvakou et al. [29], GitHub héberge en effet un grand nombre de dépôts de ce type, susceptibles d'altérer la qualité des résultats obtenus. Donc cet intervalle nécessite un filtrage des projets pour se concentrer sur les projets les plus pertinents. Ce choix nous permet ainsi de concentrer notre étude sur des projets plus pertinents et susceptibles de refléter des pratiques de développement alignées avec les exigences du RGPD.

3.4 Utilisation des LLM (Large Language Model) via une API

Pour l'analyse de la conformité dans les projets collectés lors de l'étape précédente, nous avons choisi d'utiliser un modèle LLM, en l'occurrence Mistral, plutôt que des modèles classiques. Ce choix repose sur plusieurs avantages des LLM. D'une part, leur capacité à comprendre des entrées en langage naturel complexes, leur permet de traiter des contenus variés sans nécessiter une phase intensive de prétraitement ou d'annotation. D'autre part, avec l'Ingénierie des instructions générative (prompt engineering), ces modèles s'adaptent facilement à différentes tâches, ce qui les rend flexible.

La figure 3.1 illustre l'initialisation d'un client Mistral pour l'interaction avec le modèle.

```
# Initialiser le client Mistral
client = Mistral(api_key=MISTRAL_API_KEY)
```

FIGURE 3.1 – Initialisation de client Mistral

Un LLM comme Mistral n'est pas utilisé localement, mais via un service distant, accessible à travers une clé d'API sécurisée fournie par l'API Mistral. Il est nécessaire de faire plusieurs choix et réglages pour garantir un traitement stable et conforme aux limites imposées par l'API. Voici les composantes essentielles de cette utilisation :

3.4.1 Choix du modèle

Le modèle utilisé pour l'ensemble de cette étude est mistral-large-latest, développé par Mistral AI. Ce choix repose sur plusieurs critères : il s'agit d'un modèle open source, offrant d'excellentes capacités de raisonnement, ce qui le rend particulièrement adapté aux tâches complexes [2]. Par ailleurs, sa fenêtre de contexte étendue à 128 000 tokens permet d'analyser efficacement de grands volumes de données, notamment des contenus longs tels que les fichiers README.

3.4.2 Construction des prompts

Pour chaque appel à l'API, un prompt structurée est envoyée afin d'orienter le modèle vers la tâche à réaliser, comme la classification automatique. Ce prompt est soigneusement formulée pour guider le modèle à générer des réponses pertinentes répondant au besoin. Elle commence par définir clairement le rôle du modèle, en le positionnant par exemple, comme un expert du RGPD. Elle précise ensuite le contenu à analyser, qu'il s'agisse de commits, de documentation de projet, etc.

Le type de prompt est choisi en fonction du besoin : cela peut être une instruction de type zero-shot, few-shot ou encore chain-of-thought, selon le niveau de contexte ou d'exemples nécessaires pour guider le modèle.

Par ailleurs, le format de sortie est explicitement spécifié afin d'obtenir des réponses bien structurées et facilement exploitables automatiquement.

Enfin, le prompt peut être ajusté de manière itérative afin d'améliorer la qualité des réponses.

3.4.3 Paramètres de l'appel au modèle

L'appel au modèle (voir fig 3.2) repose sur une configuration standardisée précisant trois paramètres essentiels.

```
response = client.chat.complete(  
    model="mistral-large-latest",  
    messages=[{"role": "user", "content": prompt}],  
    temperature=0  
)
```

FIGURE 3.2 – Appel au modèle Mistral via l'API

Le modèle utilisé est mistral-large-latest, tel que défini précédemment. La température est fixée à 0, ce qui permet de garantir des réponses déterministes, sans aléa dans la génération.

Enfin, la structure des messages respecte le format attendu par l'API, avec un rôle défini (role : "user") et un contenu (content) correspondant au prompt.

3.4.4 Gestion des appels API

L'interaction avec une API de modèle de langage nécessite une gestion rigoureuse pour la cohérence des résultats et le respect des limites imposées par l'API. Pour cela, plusieurs mécanismes ont été mis en place :

- Traitement par lots : permet d'envoyer plusieurs éléments (par exemple plusieurs commits ou descriptions) en un seul appel API. Cela réduit le nombre total d'appels effectués et optimise l'utilisation des ressources disponibles.
- Délai entre les lots : introduit un temps d'attente entre deux appels consécutifs. Cette mesure évite une sollicitation excessive de l'API, ce qui pourrait entraîner des erreurs ou un blocage temporaire du service.
- Mécanisme de "retry" : en cas d'erreur temporaire (par exemple un problème réseau), le système relance automatiquement la requête, dans la limite d'un nombre défini de tentatives.
- Temps d'attente initial avant chaque nouvelle tentative : ce délai entre deux essais successifs réduit la pression sur le service et la probabilité d'échecs répétés.
- Stratégie de backoff exponentiel : après chaque échec, le délai d'attente est augmenté de façon exponentielle. Cette stratégie évite d'aggraver une situation de surcharge (notamment lors d'erreurs 429 "Too Many Requests") et laisse au service distant le temps de se rétablir.

La figure 3.3 montre un exemple de gestion des appels API, illustrant la manière dont les requêtes sont traitées.

```
max_retries = 5
initial_sleep_time = 2
max_sleep_time = 32 # Temps d'attente maximum en secondes

# Paramètres de traitement par lots
batch_size = 5 # Nombre de dépôts à traiter par lot
delay_between_batches = 10 # Délai en secondes entre les lots
```

FIGURE 3.3 – Exemple de gestion des appels api

Les valeurs définies résultent d'une série d'ajustements progressifs réalisés au fil des expérimentations, ainsi dépend de plusieurs critères, notamment le type de contenu traité et le volume de données à analyser. L'ensemble de ces mécanismes a été mis en œuvre progressivement en réponse aux erreurs rencontrées lors des premières interactions avec l'API, dans le but de trouver un équilibre optimal entre la pertinence des résultats et le temps de traitement.

3.5 La classification automatique pour l'analyse de conformité

Après avoir présenté les principes méthodologiques encadrant l'utilisation des LLM dans notre étude, nous passons à leur mise en œuvre sur l'approche de classification. Notre objectif à travers cette classification est de cibler les projets les plus pertinents pour notre étude, en identifiant ceux qui présentent un lien clair avec le RGPD afin d'analyser les pratiques de protection des données personnelles mises en œuvre dans leur code. Deux approches de classification ont été entamées :

- L'analyse des commits des projets.
- L'analyse des descriptions de projets et de leurs fichiers README.

3.5.1 Analyse de conformité à travers les commits

Cette section décrit la manière dont les commits ont été collectés, et soumis au modèle afin d'identifier les commits relatifs à la conformité au RGPD.

3.5.1.1 Collecte des commits

Dans un premier temps, nous avons sélectionné un échantillon de projets populaires, reconnus pour leur qualité et susceptibles de contenir des commits bien rédigés. Cette sélection a conduit à un ensemble de 41 projets à analyser. La figure 3.4 illustre les métadonnées des commits collectés, présentées sous forme de tableau.

repo_url	repo_name	commit_sha	author	date	message
https://github.com/SecUSo/privacy-friendly-pedometer	SecUSo/privacy-friendly-pedometer	6c71dfc5b82887365db0edc4db7162ddad419e47	morckx	2024-02-09 19:58:31+00:00	Add GPL license notices to new files And clean up code and imports.
https://github.com/SecUSo/privacy-friendly-pedometer	SecUSo/privacy-friendly-pedometer	cd35d4d9b3c2b1328290496dc100891387a5e0ab	morckx	2023-01-07 22:47:38+00:00	Introduce instrumented espresso CI tests * all provided locales on all fragments and preferences sections * csv export * step count test
https://github.com/SecUSo/privacy-friendly-pedometer	SecUSo/privacy-friendly-pedometer	f938ed3afb13f80b38b7108834556feb5a41044	morckx	2024-01-31 21:13:40+00:00	Add dependabot workflow For * android gradle build * github ci workflows
https://github.com/SecUSo/privacy-friendly-pedometer	SecUSo/privacy-friendly-pedometer	bc46aa185b06c53b3f68afb25156d9166aa8290d	udenr	2024-01-23 12:30:21+00:00	Merge pull request #139 from SecUSo/changelog-"v3.1.0" Update changelog for "v3.1.0"
https://github.com/SecUSo/privacy-friendly-pedometer	SecUSo/privacy-friendly-pedometer	e0ad4387d79ea12e742f93c31e880279f9fc595	udenr	2024-01-23 11:46:44+00:00	update changelog for "v3.1.0" changes This commit was created by changelog-from-release in "Changelog Generation" CI workflow
https://github.com/SecUSo/privacy-friendly-pedometer	SecUSo/privacy-friendly-pedometer	e3343a1e05e756f344d946b9f52a5e5a6cc0b5d4	udenr	2024-01-23 11:39:47+00:00	Update changelog.yml
https://github.com/SecUSo/privacy-friendly-pedometer	SecUSo/privacy-friendly-pedometer	6a061d779b0f833a7843f1b001b9e09f2a2cd843	udenr	2024-01-23 11:39:54+00:00	Merge pull request #135 from jahway803/documentation documentation: Added info re: Privacy Friendly Backup to README.md
https://github.com/SecUSo/privacy-friendly-pedometer	SecUSo/privacy-friendly-pedometer	13d0c10049d8b3c3e1335aeeb80b1ab7a67d93c5	udenr	2024-01-23 11:36:00+00:00	Merge pull request #138 from SecUSo/development Update to version 3.1.0
https://github.com/SecUSo/privacy-friendly-pedometer	SecUSo/privacy-friendly-pedometer	90e5cd8eb817b805840c24d2988ed2314478f0c0	udenr	2024-01-23 11:28:14+00:00	Bump version to 3.1.0 (16)
https://github.com/SecUSo/privacy-friendly-pedometer	SecUSo/privacy-friendly-pedometer	4e1a791faac75ba0549d415f57297b812fb8044	udenr	2024-01-23 11:24:13+00:00	Merge pull request #137 from SecUSo/add-ci-workflow Add ".ci.yml" workflow

FIGURE 3.4 – Tableau des métadonnées des commits collectés

Le nombre élevé de commits représentait un inconvénient majeur lors de la phase de collecte, notamment en raison de la limitation du nombre de requêtes imposée par l'API de GitHub. Pour optimiser cette étape, nous avons donc récupéré les 500 derniers commits de chaque projet. Les commits ont été stockés (14823 commits) avec leurs métadonnées associées (la date de modification, la clé SHA, etc.). Cette stratégie visait à accélérer la collecte tout en poursuivant

notre objectif initial : identifier les pratiques de mise en conformité à partir de l'analyse des commits.

3.5.1.2 Classification des commits

Un prompt a été conçu pour guider le modèle de manière précise dans la tâche de classification en zero-shot (sans exemples fournis). Il s'articule autour de plusieurs règles :

- Classification stricte selon une liste de catégories RGPD prédéfinies (proposée par un expert de domaine).
- Proposition de nouvelle catégorie si le commit est lié au RGPD mais ne rentre dans aucune catégorie existante.
- Analyse par mots-clés : RGPD, GDPR, data protection, compliance, etc.
- Analyse sémantique même en l'absence de mots-clés explicites.
- Gestion des commits hors sujet : tout commit non lié à la conformité est classé comme "Non-Compliant".

3.5.1.3 Résultats de classification

La classification des messages de commits (voir fig 3.5) a permis de distinguer deux grandes catégories : les commits "Non-Compliant" et ceux appartenant à "Other Categories", liés au RGPD.

5b263ed2cc417140fd5a69fc180e5377073ea6fb	ComplianceBoard Banner (#72) ComplianceBoard Banner	Compliance
ab6636bec12879e06f1c880557a43f3d799677e	ComplianceBoard Banner	Compliance
b3143a80eb89a8673abfb538142f58f2e0cb28f5	ccpa-checklist-tools (#71) ccpa-checklist-tools	Compliance
afa462281a37359e49c447c8dc97071b4109c40c	ccpa-checklist-tools	Compliance
127dc91dedfb9c5049e7ca61fa56a96c8163af48	Add missing footer	Non-Compliant
f34364f7b95f1ea3ca099b98cee17d7b4484971	Remove page query	Non-Compliant
4a421b3c15ccb2f2c6e69258df68c22e40bd2633	Remove footer and add PR link	Non-Compliant
6dfed7f95ede0de6f832bd8eeac7c47226683c6	Add Privacy Radius experience	Privacy
e24f71df795224cd72d2462a7fe3756e45098c00	Add schema tags	Non-Compliant
000c88ed49e2a233bd313e23ea2838f7454ee762	Update GDPR Form links & copy	GDPR

FIGURE 3.5 – Tableau des commits classifiés

Commits "Non-Conforme" (Non-Compliant)

Une majorité des messages de commits ont été classés comme "Non-Compliant", commits jugés non liés à la conformité RGPD (14001 commit). Ces messages sont souvent peu informatifs ou mal rédigés, rendant leur interprétation difficile. D'autres messages sont complètement hors sujet par rapport à la thématique de la conformité. Ces commits décrivent des actions génériques sur le projet.

Commits "Autres Catégories" (Other Categories)

À l'inverse, une minorité de messages de commits considérés comme liés à la conformité, appartenant soit à une catégorie RGPD prédéfinie, soit à une nouvelle catégorie définie par le modèle (825 commits). Plusieurs d'entre eux font référence explicitement à des principes

de RGPD. Par exemple, les commits "java docs for consent enums added" et "added GDPRConsent.AUTOMATIC_PERSONAL_CONSENT" témoignent d'implémentation ou de documentation autour de la gestion du consentement des utilisateurs.

D'autres messages, comme "added an isInitialised function to the GDPR class", font référence à des éléments directement associés à des modules de conformité. Ces messages révèlent une intention explicite d'aligner le développement du projet avec les exigences de protection des données personnelles imposées par le RGPD.

3.5.1.4 Limites

La manipulation d'un volume important de commits a représenté un défi majeur, tant lors de la phase de collecte que durant le traitement via les LLM. Le traitement de cet échantillon de projets s'est avéré particulièrement long, et les résultats obtenus ont montré que seule une minorité des commits était véritablement exploitable pour l'analyse de conformité. Face à ces limites, notamment le rapport peu favorable entre temps de traitement et qualité des résultats, nous avons décidé d'explorer une approche alternative.

3.5.2 Analyse de conformité à travers la documentation (README) et les descriptions de projets

Cette seconde approche de classification vise à élargir la portée de notre analyse à des informations contextuelles plus générales et souvent plus explicites que les messages de commit. Avant de détailler les résultats issus de cette seconde analyse, nous présentons tout d'abord les étapes spécifiques de collecte et de traitement de ces contenus.

3.5.2.1 Collecte des données

Dans cette étape, nous avons récupéré les descriptions des projets à partir des métadonnées précédemment collectées. Ensuite, nous avons extrait les fichiers README correspondants en utilisant l'API GitHub. Ces contenus ont été transmis au LLM, afin d'être utilisés comme entrée pour la classification.

3.5.2.2 Classification avec LLM

Afin d'orienter efficacement le modèle de langage dans sa tâche de classification, un prompt spécifique a été conçu. On trouve ci-dessous (figure 3.6) un exemple de prompt utilisé pour interagir avec le modèle dans le cadre de notre analyse.

```

prompt = f"""
You are an expert in analyzing projects related to GDPR (General Data Protection Regulation).
Analyze the following information (description and README) and answer the questions:

### Project Description:
{description}

### README Content:
{readme_text}

### Questions:
-Is the project specifically designed for GDPR compliance? (Yes / No)
-What is the type of project? (Framework, Tool, Guide, Approach, Other)
-If the project is GDPR-related, which concepts does it address? (e.g., Consent, Security, DPO, Data Portability, etc.)
-Is the project a dedicated GDPR solution, a business project compliant with GDPR, or not related to GDPR at all?
(Answer with: "Dedicated GDPR solution", "Business project with GDPR compliance", or "Not related to GDPR")
-Explain why the project is classified as GDPR-compliant or not. (Provide a brief explanation based on the description and README.)

### Respond in this strict format:**
"Classification: Yes/No | Type: <Project Type> | Concepts: <List of GDPR concepts or N/A>
| Project Nature: <Dedicated GDPR solution / Business project with GDPR compliance> | Explanation: <Brief explanation>"

Instructions:
1. Carefully read the README and description before answering.
2. Provide a clear and concise explanation for your classification.
3. If the project is not GDPR-related, explain why.
"""

```

FIGURE 3.6 – Exemple d'un prompt

Celui-ci positionne le modèle comme un expert en projets liés au RGPD. En se basant sur le contenu d'entrée, le modèle est invité à répondre à cinq questions précises, concernant notamment :

- L'adéquation du projet au RGPD.
- Le type du projet (framework, outil, guide, approche, etc.).
- Les concepts RGPD abordés.
- La nature globale du projet (solution dédiée, projet métier conforme, ou projet non lié au RGPD).
- Une explication argumentée basée sur les informations fournies.
- La réponse attendue doit suivre un format strict pour faciliter l'extraction automatique des résultats.

3.5.2.3 Résultats

Les projets ont été répartis selon leur classification par rapport à la conformité aux concepts RGPD abordés, ainsi qu'à la nature des projets.

La figure 3.7 présente le tableau des projets classifiés selon leur conformité au RGPD.

repo_name	repo_url	classification	type_project	concepts_covered	project_nature	explanation
DreamClear	https://github.com/shallowdream204/DreamClear	No	Approach	N/A	Not related to GDPR	The project "DreamClear"
farside	https://github.com/benbusby/farside	No	Tool	N/A	Not related to GDPR	The project is a smart redirecting gateway for various frontend services, focusing on privacy-oriented alternative frontends. It does not specifically address GDPR concepts such as consent, security, DPO, or data portability. The README and description do not mention any GDPR-related features or compliance measures.
Awesome-Security-Gists	https://github.com/Hack-with-Github/Awesome-Security-Gists	No	Guide	N/A	Not related to GDPR	The project is a collection of GitHub gists aimed at hackers, penetration testers, and security researchers. It focuses on various security topics such as privacy-conscious browsing, malware fact sheets, payloads, and penetration testing tips. There is no mention of GDPR compliance, data protection, or any related concepts in the project description or README. Therefore, it is not related to GDPR.
offen	https://github.com/offen/offen	Yes	Tool	Consent, Security, Data Portability, Data Minimization, Transparency	Dedicated GDPR solution	The project is specifically designed to comply with GDPR guidelines. It emphasizes user consent for data collection, end-to-end encryption for data security, and allows users full access to their data, including the ability to review and delete it. The tool is self-hosted, ensuring no third-party involvement, and uses first-party cookies only, which aligns with GDPR principles of data minimization and transparency.
pirsch	https://github.com/pirsch-analytics/pirsch	Yes	Tool	Privacy, Security, Data Minimization	Business project with GDPR compliance	The project is classified as GDPR-compliant because it emphasizes privacy-friendly analytics, does not use cookies, and does not store personal information. It generates a unique fingerprint for each visitor using a hash of non-personal data, ensuring data minimization and privacy. The project is hosted on EU servers, further aligning with GDPR requirements for data localization.

FIGURE 3.7 – Tableau des projet classifiés

Examinons les résultats de cette classification : sur l'ensemble des projets analysés, 647 ont été classés comme « non conformes » au RGPD, tandis que 104 ont été considérés comme « conformes ».

Parmi les 104 projet, 55 correspondent à des projets métiers conformes, c'est-à-dire des projets dont l'objectif principal n'est pas de fournir une solution RGPD, mais qui intègrent des pratiques de conformité aux concepts du RGPD dans leur fonctionnement. Leur étude permet de comprendre comment des applications métiers adaptent leur architecture ou leurs fonctionnalités pour répondre aux exigences réglementaires.

En parallèle, 49 projets ont été identifiés comme des solutions dédiées au RGPD, c'est-à-dire conçues spécifiquement pour traiter des problématiques liées à la protection des données personnelles. L'analyse de ces projets permet d'étudier des approches spécialisées et des bonnes pratiques mises en œuvre pour répondre aux différentes obligations imposées par le RGPD. L'étude s'est concentrée sur ces projets dédiés au RGPD en réalisant une analyse complète du code.

3.6 Analyse du code source

L'analyse du code source repose sur plusieurs étapes clés visant à extraire une représentation exploitable par les modèles de traitement automatique. Le processus mis en place est le suivant :

3.6.1 Analyse des concepts couverts

Une analyse des concepts RGPD couverts dans ces projets a été réalisée. Cela permet d'anticiper les concepts abordés dans les pratiques de développement par les développeurs, et d'étudier par la suite comment ces concepts sont effectivement implémentés dans le code.

La figure 3.8 met en évidence les concepts du RGPD les plus fréquemment couverts dans les projets analysés.

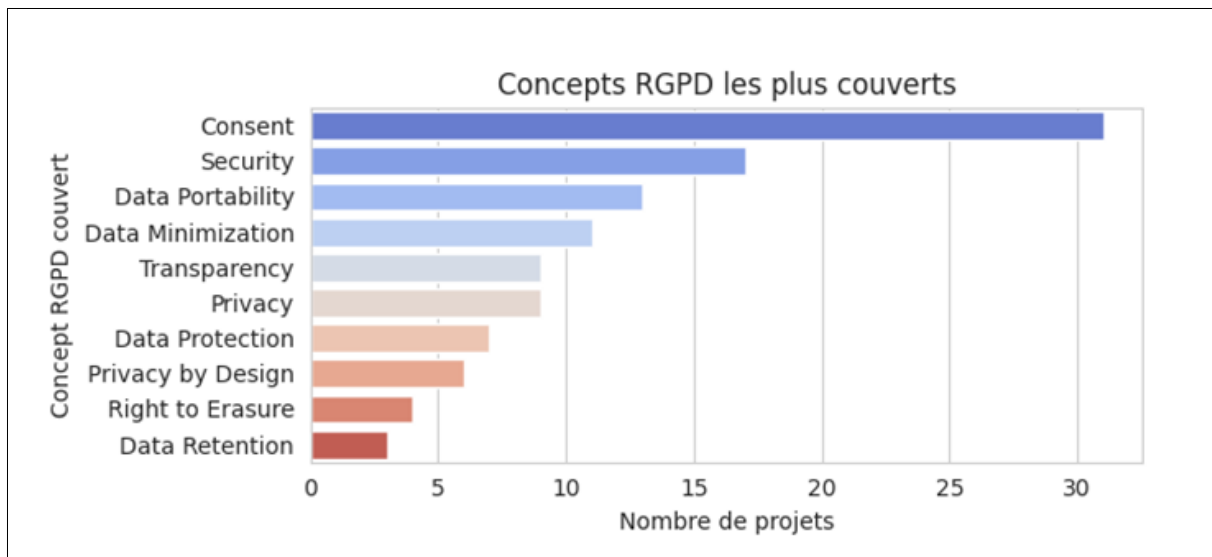


FIGURE 3.8 – Concepts du RGPD les plus couverts dans les projets

L'importance donnée aux concepts de consentement, de sécurité et de portabilité des données est compréhensible compte tenu des exigences centrales du RGPD. Cependant, la faible couverture de certains concepts comme la rétention des données et la protection de la vie privée dès la conception (privacy by design) pourrait indiquer un manque de prise en compte de certains aspects de la conformité, ou simplement une spécialisation des projets étudiés.

3.6.2 Sélection des projets

Le choix de se limiter à un nombre restreint de projets s'explique par les contraintes de ressources inhérentes à l'environnement de développement utilisé, notamment Google Colab, dont la mémoire et les capacités de calcul sont limitées. Néanmoins, l'objectif principal de ce travail est de démontrer la pertinence et la faisabilité de l'approche proposée. Voici dans le tableau 3.3 quelques projets que nous avons sélectionnés pour l'analyse.

repo name	repo_url	concepts_covered	project_nature	language
GDPR-Dialog	https://github.com/MFlisar/GDPRDialog	Consent	Dedicated GDPR solution	Java
cookie-consent	https://github.com/klaxit/cookie-consent	Consent	Dedicated GDPR solution	JavaScript

repo name	repo_url	concepts_covered	project_nature	language
amazon-s3-find-and-forget	https://github.com/aws-labs/amazon-s3-find-and-forget	Data Erasure, Security, Data Minimization	Dedicated GDPR solution	Python
ml_privacy_meter	https://github.com/privacytrustlab/ml_privacy_meter	Data protection Impact Assessment, Privacy, Security	Dedicated GDPR solution	Python

TABLE 3.3 – Tableau des projets sélectionnés pour l’analyse

- **GDPRDialog (Java)** : ce projet propose une boîte de dialogue réutilisable pour la gestion du consentement utilisateur dans les applications Android. Il intègre des mécanismes de vérification de consentement, de stockage dans les préférences, et de mise à jour du statut de consentement.
- **cookie-consent (JavaScript)** : cette bibliothèque JavaScript permet d’afficher une bannière de gestion des cookies sur les sites web. Elle inclut des options de personnalisation, de catégorisation des cookies et de stockage de l’état de consentement via cookies.
- **amazon-s3-find-and-forget (Python)** : outil AWS pour automatiser la suppression des données personnelles dans Amazon S3. Il gère notamment la localisation, l’anonymisation et l’effacement des objets associés à un utilisateur dans une logique de conformité à l’article 17 du RGPD (droit à l’effacement).
- **ml privacy meter (python)** : est un outil permettant d’évaluer la vulnérabilité des modèles de machine learning aux fuites de données personnelles. En simulant des attaques comme le Membership Inference Attack, il permet aux développeurs et aux chercheurs de tester la confidentialité de leurs modèles et de réduire les risques de violation du RGPD.

3.6.3 Génération de l’AST

Pour chaque projet, les fichiers de code source ont été collectés via l’API GitHub. Chaque projet étant associé à un langage unique, les fichiers ont été filtrés selon leur extension (.py, .java, .js, etc.) afin d’identifier uniquement les fichiers pertinents.

Ensuite, chaque fichier a été analysé à l’aide d’un parseur adapté à son langage de programmation. Cette analyse permet de produire une représentation structurée du code : un arbre syntaxique abstrait (AST). Par exemple, les parseurs utilisés incluent :

- la bibliothèque native ast pour Python.
- Esprima pour JavaScript.

- JavaParser pour Java.

Les AST produits pour tous les fichiers d'un projet sont ensuite convertis en dictionnaires JSON-compatibles et rassemblés dans un fichier JSON unique par projet. Ce fichier regroupe l'ensemble des structures syntaxiques extraites, prêtes à être manipulées ou analysées automatiquement.

La figure 3.9 illustre un exemple de la structure hiérarchique produite pour un fichier source.



FIGURE 3.9 – Exemple de la structure hiérarchique AST

On y observe la structure hiérarchique de l'AST, qui reflète l'organisation logique du code : chaque élément (déclaration de classe, méthode, annotation, etc.) est représenté sous forme de nœud, avec ses sous-éléments imbriqués. Cette représentation rend visibles les relations structurelles internes du code, tout en facilitant l'extraction d'informations ciblées (ex. : appels de méthodes, types déclarés, etc.).

3.6.4 Unification des AST

Les AST générés pour chaque langage de programmation (Java, Python, JavaScript) présentent des structures syntaxiques différentes. Le tableau ci-dessous illustre l'étape d'unification réalisée afin de produire une structure commune (quelques éléments ont été montrés dans le tableau 3.4 pour but d'illustration). Il compare les représentations natives des AST de Java, Python et JavaScript pour plusieurs éléments clés (tels que les classes, méthodes, constructeurs, imports, etc.) avec leur équivalent dans le modèle unifié.

Par exemple :

Élément extrait	AST Java	AST Python	AST JavaScript	AST unifié
Classe	<code>{"node" : "ClassDeclaration", "name" : "MyClass", ...}</code>	<code>{"_type" : "ClassDef", "name" : "MyClass", ...}</code>	<code>{"type" : "ClassDeclaration", "id" : {"name" : "MyClass"}, ...}</code>	<code>"class_name" : "MyClass", "language" : "...", ...</code>
Méthode	<code>{"node" : "MethodDeclaration", "name" : "doSomething", ...}</code>	<code>{"_type" : "FunctionDef", "name" : "do_something", ...}</code>	<code>{"type" : "MethodDefinition", "key" : {"name" : "doSomething"}, ...}</code>	<code>"methods" : [{"name" : "...", "parameters" : [...], ...}]</code>
Champ / Attribut	<code>{"node" : "FieldDeclaration", "declarators" : [{"name" : "x", ...}], ...}</code>	<code>{"_type" : "Assign", "targets" : [{"_type" : "Name", "id" : "x"}], ...}</code>	<code>{"type" : "VariableDeclaration", "declarations" : [...]}</code>	<code>"fields" : [{"name" : "x", ...}]</code>
Constructeur	<code>{"node" : "ConstructorDeclaration", "name" : "MyClass", ...}</code>	Méthode <code>__init__</code>	<code>{"type" : "MethodDefinition", "kind" : "constructor", ...}</code>	<code>"constructors" : [...]</code>
Fonction globale	<code>null</code>	<code>{"_type" : "FunctionDef", "name" : "...", ...}</code> (hors classe)	<code>{"type" : "VariableDeclaration", "init" : {"type" : "FunctionExpression"}}</code>	<code>"functions" : [{"name" : "...", ...}]</code>
Import	<code>{"path" : "java.util.List"}</code>	<code>{"_type" : "Import"}</code> ou <code>{"_type" : "ImportFrom"}</code>	<code>{"type" : "ImportDeclaration", "source" : {"value" : ...}}</code>	<code>"imports" : [{"java.util.List"} ou "os"</code>
Documentation	<code>"comment" : "Javadoc"</code>	<code>docstring</code> en première ligne : <code>{"_type" : "Expr", "value" : {"_type" : "Str", ...}}</code>	Non standard (non extrait)	<code>documentation</code>

TABLE 3.4 – Comparaison des représentations AST pour Java, Python, JavaScript en un format unifié

- Une déclaration de classe (ClassDeclaration en Java et JavaScript, ClassDef en Python) est transformée dans le format unifié en un nœud "class" avec des champs standardisés (class_name, language, etc.).
- Les constructeurs, bien que représentés différemment selon les langages (méthode `__init__` en Python, ConstructorDeclaration en Java), sont regroupés sous une structure commune "constructors".
- Les éléments spécifiques à un langage sont également pris en compte (ex. : fonctions globales en Python et JavaScript), afin d'assurer une représentation complète.

Cette unification est une étape essentielle pour permettre des analyses ultérieures sur des projets multi-langages. La figure 3.10 montre un exemple du format d'AST unifié :

```

{
  "class": {
    "class_name": "NomClasse",
    "language": "// Java, Python, etc.",
    "constructors": [
    ],
    "methods": [
    ],
    "fields": [
    ],
    "imports": [
    ]
  }
}
}functions": {
}

```

FIGURE 3.10 – Exemple du format d'un AST unifié

3.6.5 Analyse des AST : Approche de RAG

L'implémentation de notre processus d'analyse des ASTs repose sur une architecture modulaire conforme au paradigme RAG. Cette section présente les choix d'implémentation des différentes phases du système.

3.6.5.1 Phase 1 : Construction de la base de connaissances

1. Prétraitement des AST

À partir des AST unifiés, nous avons transformés le format JSON en un format textuel compatible avec l'analyse par des modèles en aval.

Cette transformation inclut l'ajout de séparateurs entre les documents (voir figure 3.11), permettant ainsi une structuration par sections (correspondant aux fichiers) et sous-sections (correspondant aux éléments du fichier).

```

for item in data:
    output.append(f"\n--- Document ---\n")
    output.append(f"In file '{item['file_path']}', the following elements are defined:")

```

FIGURE 3.11 – Ajout de séparateurs entre les documents

La représentation textuelle résultante peut adopter soit une structure plate, soit conserver une organisation hiérarchique, en fonction du niveau de granularité requis pour l'analyse. Par exemple, les bibliothèques importées sont représentées dans l'AST sous forme de nœuds hiérarchiques dans la section «imports». Dans notre format textuel, nous les convertissons en une liste plate afin de simplifier leur traitement.

La figure 3.12 ci-dessous présente un exemple de représentation d'AST pour les imports.

```
Imports:
  • decimal.Decimal
  • logging
  • os
  • sys
  • collections.Counter
  • io.BytesIO
  • numpy
  • pyarrow
  • pyarrow.parquet
```

FIGURE 3.12 – Exemple de représentation d’AST pour les imports

En revanche, les fonctions et méthodes, qui contiennent des informations plus riches, conservent une structure hiérarchique. Le nom de la fonction et ses paramètres sont clairement explicite afin d’en améliorer la lisibilité.

La figure 3.13 ci-dessous présente un exemple de représentation d’AST pour une fonction.

```
Functions:
  • with_logging(handler)
    Documentation:
      Decorator which performs basic logging and makes logger available on context
    Body:
      {
        "_type": "Expr",
        "value": {
          "_type": "Constant",
          "value": "\n    Decorator which performs basic logging and makes logger available on context\n    ",
          "kind": null
        }
      }
}
```

FIGURE 3.13 – Exemple de représentation d’AST pour une fonction

2. Découpage des AST

Les AST en format textuel sont ensuite découpés en segments (chunks). Le découpage est effectué par structure de code : champs, fonctions, classes, méthodes, etc. Cette approche structurelle préserve la cohérence sémantique de chaque élément de code, contrairement à un découpage arbitraire par taille fixe.

La taille d’un segment est fixée à 480 tokens, ce choix répond à un équilibre entre plusieurs contraintes structurelles et fonctionnelles. D’abord, ce seuil reste en dessous de la limite maximale de 512 tokens imposée par le modèle de représentation vectorielle utilisé dans l’étape suivante, ce qui permet d’éviter les dépassements et de conserver une marge pour l’ajout éventuel de métadonnées. Ensuite, cette taille est adaptée à la nature du contenu analysé principalement du code source structuré et relativement dense en conservant une richesse contextuelle suffisante pour chaque segment. Par ailleurs, dans le cadre d’un usage

exploratoire et d'extraction de pratiques de confidentialité, il est essentiel que les segments soient suffisamment longs pour contenir des blocs de logique complets, tout en restant courts pour permettre une recherche rapide et ciblée.

Techniquement, nous utilisons un "tokenizer" compatible avec notre modèle pour mesurer précisément la taille d'un segment en nombre de tokens (et non en caractères), afin de respecter les limites imposées.

Dans notre implémentation, le découpage est effectué en utilisant les expressions régulières pour détecter les sections et les sous sections comme suit :

2.1 Découpage en sections principales

Le document est d'abord segmenté en sections globales à l'aide d'un séparateur reconnaissable (— Document —) indiquant le début d'un nouveau fichier.

2.2 Découpage en sous sections

Chaque section est ensuite divisée en sous-sections selon les en-têtes comme Imports :, Functions :, Methods :, etc.

— Méthodes et fonctions :

Sont traitées avec une attention particulière :

Si le corps d'une méthode est trop long et dépasse la limite imposé par le modèle représentation vectorielle choisit, il est découpé ligne par ligne tout en conservant systématiquement l'en-tête (nom de la méthode, signature) dans chaque morceau. Cette duplication de l'en-tête dans chaque segment permet de maintenir le contexte tout au long des segments, ce qui est crucial pour la compréhension du comportement logique. Le découpage s'effectue de préférence entre des blocs logiques (après des accolades fermantes ou des points-virgules) pour éviter de couper brutalement en plein milieu d'une structure syntaxique comme une condition ou une boucle.

— Autres sections :

(imports, fields, ou blocs explicatifs) sont soit conservées intégralement (si leur taille est inférieure au seuil maximal), soit découpées par paragraphes logiques.

Pour chaque segment extrait, nous conservons les métadonnées suivantes :

- Le type de contenu (method_body, method_header, import_section, field_declaration, etc.)
- Le nom de la méthode ou fonction associée.
- La position du segment dans la méthode (par exemple "2/4" pour indiquer qu'il s'agit du deuxième segment sur quatre).
- Le chemin du fichier source.

Ces métadonnées permettent d'identifier et d'enrichir les segments, ainsi de reconstituer une unité complète à partir de ses parties, lors de la génération des réponses.

3. Représentation vectorielle (Embedding)

Nous avons utilisé le modèle all-MiniLM-L6-v2 (open source) de la bibliothèque Sentence-Transformers, un choix motivé par plusieurs critères. Tout d’abord, ce modèle présente un compromis entre performance et consommation de ressource. De plus, il est spécifiquement optimisé pour les tâches de similarité sémantique : il a été entraîné pour maximiser la similarité cosinus entre des phrases de sens proche, ce qui correspond parfaitement à notre besoin de retrouver des segments de code sémantiquement pertinents par rapport à une requête utilisateur. Enfin, ses vecteurs de 384 dimensions offrent une richesse sémantique.

4. Choix relatifs à l’indexation

Pour assurer une recherche précise dans un temps raisonnable, nous avons choisi FAISS avec l’index IndexFlatIP. Ce choix repose sur plusieurs considérations :

- IndexFlatIP avec des vecteurs normalisés permet une recherche basée sur la similarité cosinus, plus adaptée à la comparaison sémantique que l’approche euclidienne (IndexFlatL2) qui se base sur la magnitude des vecteurs.
- L’index plat garantit des résultats exactes, contrairement aux index approximatifs (IVF, HNSW) qui sacrifient la précision pour la vitesse.

Les représentations vectorielles proviennent de fichiers JSON multi-langages (Java, Python, JavaScript, etc.) et sont regroupés dans une base unifiée, permettant des recherches efficaces, extensibles et précises sur l’ensemble du corpus de code.

3.6.5.2 Phase 2 : Recherche et génération de réponses

1. Traitement des requêtes utilisateur

La requête utilisateur (fig 3.14) déclenche le système, elle est transformée en représentation vectorielle avec le même modèle que celui utilisé pour les segments :

```
user_query = "How do I save the current user consent status?"
query_embedding = model.encode([user_query], normalize_embeddings=True)
```

FIGURE 3.14 – Exemple d’une requête utilisateur

2. Recherche des segments pertinents

Lors de cette étape, la base de données vectorielle est interrogée pour retourner les k segments pertinents.

La figure 3.15 illustre une requête de recherche vectorielle effectuée à l’aide de la méthode `index.search`.

```
D, I = index.search(np.array(query_embedding, dtype=np.float32), k)
```

FIGURE 3.15 – Requête de recherche vectorielle avec index.search

- `index.search(...)` : interroge l'index FAISS pour trouver les k vecteurs les plus proches selon la métrique produit scalaire.
- D : matrice contenant les scores de similarité pour chaque résultat.
- I : matrice contenant les indices des vecteurs correspondants dans la base.
- k : nombre de segments plus similaires à la requête que FAISS doit retourner.

La figure 3.16 montre le processus de récupération des résultats après une requête de recherche vectorielle.

```
# 2. Récupération des résultats FAISS (chunk + score FAISS)
top_faiss_results = [(extracted_data[i], D[0][j]) for j, i in enumerate(I[0])]
```

FIGURE 3.16 – Récupération des résultat

- $I[0]$: liste des k indices retournés par FAISS.
- $D[0][j]$: score associé à chaque résultat (j étant la position dans la liste).
- `extracted_data[i]` : accède au segment de texte correspondant à l'index i .
- Le résultat est une liste de tuples contenant :
 - le segment pertinent.
 - son score de similarité avec la requête.

Les résultats sont fournis à un LLM pour générer une réponse contextualisée, en se basant uniquement sur les segments de données jugés pertinents.

3. Génération de réponses contextualisées

À cette étape, le LLM est intégré à l'approche de récupération pour générer des résultats contextualisés. C'est à ce moment que nous avons extrait les pratiques liées à la protection des données privées dans le contexte du RGPD.

Pour cela, nous avons structuré un prompt destiné à guider le LLM dans la génération des réponses, tel que le modèle est positionné comme un expert en conseil RGPD.

La génération repose sur une approche de "multi-step prompting" (aussi appelée prompt chaining), où trois étapes successives sont utilisées pour extraire les pratiques techniques à partir des segments pertinents et les transformer en un guide RGPD structuré, ainsi qu'en une évaluation de conformité :

- **Extraction des pratiques** : Un premier prompt identifie les éléments pertinents dans les k segments en rapport avec la requête utilisateur. Cette approche "multi-prompts" permet de mieux cibler les informations.

- **Analyse détaillée** : Un second prompt prend en entrée les pratiques extraites et les structure sous forme de guide technique à destination des développeurs. Ce guide peut inclure des méthodes, des extraits de code, des bibliothèques utilisées, ou encore des instructions d'intégration.
- **Évaluation de la conformité RGPD** : Enfin, un troisième prompt est appliqué lorsque la requête cible un projet spécifique. Il a pour objectif de déterminer si le projet met en œuvre les pratiques RGPD attendues, d'identifier d'éventuelles lacunes, et de proposer des améliorations concrètes.

3.7 Conclusion

Dans ce chapitre, nous avons présenté notre approche d'analyse de conformité en détaillant les différentes étapes techniques mises en œuvre. Nous avons présenté les choix technologiques adoptés et les outils utilisés pour collecter, traiter et analyser les données des projets logiciels. L'intégration des modèles de langage et la structuration des instructions génératives ont également été abordées. Cette phase d'implémentation constitue une base solide pour la suite de notre travail.

Dans le chapitre suivant, nous allons valider les résultats de classification obtenus à partir des deux approches explorées, puis évaluer l'efficacité du système RAG à extraire les pratiques de conformité. L'analyse de ces pratiques constituera la phase finale de notre étude, dont l'objectif est de mieux comprendre l'implémentation de ces pratiques dans le développement logiciel.

Chapitre 4

Tests expérimentaux et validation

4.1 Introduction

Dans ce chapitre, nous présentons l'évaluation finale de notre approche d'analyse de conformité. Nous commençons par valider les résultats obtenus à partir des deux méthodes de classification explorées. Ensuite, nous évaluons les performances du système RAG selon deux dimensions : la récupération des documents pertinents et la génération de réponses. Cette évaluation permet de valider l'efficacité de l'ensemble de la chaîne de traitement, depuis l'identification des sources d'information jusqu'à la production de réponses cohérentes et fidèles au contenu extrait. L'objectif est de mesurer la capacité du système à identifier et transformer les documents collectés en pratiques de conformité claires, exploitables et utiles dans le cadre de notre étude.

4.2 Validation manuelle des résultats de classification

Afin de garantir la fiabilité des résultats issus des classifications automatiques réalisées par les grands modèles de langage (LLM), une validation manuelle a été mise en place pour les deux approches utilisées dans ce travail : l'analyse des commits et l'analyse des descriptions/README des projets. Cette validation repose sur l'examen d'un échantillon représentatif des données, selon le processus décrit ci-dessous.

4.2.1 Méthodologie de validation

La méthodologie de validation repose sur deux étapes principales : la sélection de l'échantillon à valider, puis l'analyse des résultats obtenus.

4.2.1.1 Étape 1 – Sélection de l'échantillon

Un échantillon représentatif des éléments analysés automatiquement (commits ou projets) a été sélectionné à l'aide d'un calculateur de taille d'échantillon (fig 4.1).

Population Size ⓘ

Confidence Level (%) ⓘ

Margin of Error (%) ⓘ

1000

95

5

Sample size

278

FIGURE 4.1 – Calculateur de taille d'échantillon

Par exemple, avec une population de 1000 éléments, un niveau de confiance de 95 % et une marge d'erreur de 5 %, la taille de l'échantillon requise pour obtenir des résultats statistiquement fiables est de 278 éléments.

4.2.1.2 Étape 2 – Analyse des résultats

Chaque élément de l'échantillon est examiné manuellement afin de déterminer si la classification fournie par le modèle est correcte ou non. Un expert du domaine a validé certains résultats. Cette analyse permet d'évaluer la précision du modèle et d'identifier les erreurs.

4.2.2 Validation des résultats pour l'approche basée sur les commits

Pour cette approche, un échantillon de 375 commits a été sélectionné pour validation parmi 14826 commits. Chaque commit a été lu attentivement afin de vérifier si la classification automatique était correcte. L'évaluation se base sur les critères suivants :

- **Classification** : le commit est-il correctement classé comme Non-conforme (Non-compliant) ou, s'il est Conforme (Compliant), dans la bonne catégorie thématique ?
- **Justification** : l'étiquette attribuée est-elle justifiée par le contenu du commit (message et modification de code) ?

Le tableau 4.1 illustre les résultats de la validation manuelle appliquée à la classification des commits effectuée par le modèle de langage (LLM).

Message de commit	Classification par LLM	Classification correct ?	Justification
Add an option for a soft consent mode where services are not loaded until consent #1286	Consent	Oui	Mention explicite du recueil de consentement préalable
"Update IAB Tech Lab - Consent string and vendor list formats v2.md Update dataRetention text to include ... in days ..."	data retention	Oui	Le message indique une modification de la durée de conservation des données, ce qui est une exigence du RGPD.

Message de commit	Classification par LLM	Classification correct ?	Justification
:zap : perf : replace 'FuturesUnordered' with 'tokio : :JoinSet' & relocate random user agent generation to search route (#630) Relocates random user agent to search route so that it is initialized only once per search request. Additionally, rayonizes and idiomatizes some code.	non compliant	oui	aucune mention explicite de pratiques liées au RGPD.
Add fallback for the multiple gtag service Fix #1247	consent	Non	aucune mention explicite ou implicite de consentement.

TABLE 4.1 – Validation de la classification des messages de commit

Le tableau présente les résultats de la validation manuelle appliquée aux commits classés automatiquement par le modèle de langage (Mistral). Au cours du processus de classification automatique des commits, nous avons observé que certains messages pouvaient être mal classés, mal rédigés ou trop peu informatifs, ce qui pouvait nuire à la précision des résultats.

La phase de validation humaine a joué un rôle dans ce processus. Chaque fois qu'une erreur de classification était identifiée, le prompt a été réajusté dans le but d'améliorer la qualité des résultats fournis par le modèle. Cette démarche nous a permis de renforcer la pertinence des commits extraits. Toutefois, malgré ces ajustements, certains messages de commit restaient peu pertinents ou trop génériques pour permettre une classification fiable. Cette limite reflète les contraintes inhérentes à la qualité des messages disponibles dans les dépôts étudiés.

4.2.3 Validation des résultats pour l'approche basée sur les README et descriptions

Dans cette seconde approche, un échantillon de 255 projets a été sélectionné. La validation repose sur la comparaison entre les résultats produits automatiquement et les informations disponibles sur les dépôts GitHub des projets (README et description). Chaque projet est évalué selon les critères suivants :

- **Classification ("Yes/No")** : La classification est-elle correcte (présence ou non d'une finalité RGPD) ?
- **Type de projet (Framework, Tool, Guide, etc.)** : Le type identifié est-il approprié au contenu du projet ?

- **Concepts RGPD couverts** : Les concepts listés sont-ils pertinents, complets et bien extraits ?
- **Nature du projet** : Le projet est-il correctement identifié comme une "Dedicated GDPR solution" ou un "Business project with GDPR compliance" ?
- **Explication / justification** : La justification fournie par le modèle est-elle claire, cohérente et fidèle au contenu du projet ?

Les résultats sont illustrés dans le tableau 4.2 :

Nom de depot	Classification	Type_project	Concept couvert	Nature de projet	Classification correct ?	Commentaire
ClearGDPR	oui	outil	consentement	projet dédié rgpd	oui	Implémente clairement le consentement selon le RGPD.
GDPR-Developer-Guide	oui	guide	Security, Data Minimization, User Rights, Legal Basis, Data Retention, Privacy by Design, Informed Consent	projet dédié rgpd	oui	Traite directement les principaux concepts du RGPD.
minidenticons	oui	outil	Data Minimization, Security	projet métier	oui	une bibliothèque permettant de générer des identicons SVG et qui intègre l'rgpd
deep_privacy	no	outil	/	Non lié au RGPD	oui	Aucun lien direct avec le RGPD observé.

TABLE 4.2 – Validation manuelle de la classification des documentations

La classification automatique basée sur les README et descriptions s'est révélée globalement fiable, comme le montre le tableau 4.2. La majorité des classifications proposées par le LLM ont été jugées correctes lors de la validation manuelle, ce qui reflète la pertinence des résultats. Cela s'explique par la bonne structuration des README, qui présentent clairement les objectifs et fonctionnalités des projets, ainsi que par les performances des modèles LLM en traitement de texte. Ces derniers ont su exploiter les documentations pour extraire des projets pertinents liés au RGPD et distinguer ceux qui ne l'étaient pas directement.

4.3 Validation des résultats du système RAG

Cette section détaille le processus de validation du système RAG, en commençant par la méthodologie d'évaluation, suivie de l'analyse de la récupération, de l'amélioration par reranker, et se concluant par une conclusion des résultats.

4.3.1 Méthodologie d'évaluation du système RAG

Le système RAG fonctionne selon le principe suivant : un utilisateur pose une question, le système récupère des passages pertinents dans la base de connaissances, puis combine ces passages avec la question pour générer une réponse à l'aide d'un LLM. L'évaluation du RAG repose sur deux dimensions essentielles :

1. **Évaluation du retrieval** : Les passages récupérés sont-ils réellement pertinents par rapport à la question posée ?
2. **Évaluation de la génération** : La réponse générée par le LLM, à partir des passages récupérés, est-elle correcte, complète et bien formulée ? [5]

Ces deux volets constituent ensemble d'évaluation complète du système RAG. Le premier vise à juger la qualité de la recherche d'information, tandis que le second mesure la pertinence et la cohérence des réponses produites.

4.3.2 Évaluation de la récupération (retrieval)

Pour évaluer la capacité du système à récupérer les bons documents avant génération, nous avons interrogé la base vectorielle FAISS avant l'intégration du LLM. L'objectif est de vérifier si les documents pertinents sont correctement identifiés pour chaque requête.

4.3.2.1 Étape 1 – Préparation des données d'évaluation

Nous avons constitué un dictionnaire `ground_truth` associant chaque requête à l'ensemble des documents (segments de méthode) attendus. Chaque partie d'une méthode est considérée comme un document à part entière. La figure 4.2 illustre la structure de ce dictionnaire.

```

ground_truth = {
  "How do I save the current user consent status?": [
    "setConsent 1/6", "setConsent 2/6", "setConsent 3/6",
    "setConsent 4/6", "setConsent 5/6", "setConsent 6/6"
  ],
  "How to get the current user consent setting?": [
    "getConsentState 1/6", "getConsentState 2/6", "getConsentState 3/6",
    "getConsentState 4/6", "getConsentState 5/6", "getConsentState 6/6"
  ],
  "method that retrieves accepted Categories from cookie object": [
    "acceptedCategories 1/1"
  ],
  "Which method computes attack metrics from MIA scores?": [
    "compute_attack_results 1/8", "compute_attack_results 2/8", "compute_attack_results 3/8",
    "compute_attack_results 4/8", "compute_attack_results 5/8", "compute_attack_results 6/8",
    "compute_attack_results 7/8", "compute_attack_results 8/8"
  ],
  "which method can be used delete rows from a Parquet file based on a list of values to remove from specific columns?": [
    "get_row_indexes_to_delete 1/7", "get_row_indexes_to_delete 2/7", "get_row_indexes_to_delete 3/7",
    "get_row_indexes_to_delete 4/7", "get_row_indexes_to_delete 5/7", "get_row_indexes_to_delete 6/7",
    "get_row_indexes_to_delete 7/7"
  ],
  "How can I determine if a user needs to be asked for GDPR consent?": [
    "checkIfNeedsToBeShown 1/9", "checkIfNeedsToBeShown 2/9", "checkIfNeedsToBeShown 3/9",
    "checkIfNeedsToBeShown 4/9", "checkIfNeedsToBeShown 5/9", "checkIfNeedsToBeShown 6/9",
    "checkIfNeedsToBeShown 7/9", "checkIfNeedsToBeShown 8/9", "checkIfNeedsToBeShown 9/9"
  ]
}

```

FIGURE 4.2 – Dictionnaire "ground_truth"

Par exemple pour la requête “How do I save the current user consent status?”, la méthode setConsent, comporte 6 parties. Ainsi, les 6 documents setConsent 1/6, ..., setConsent 6/6 doivent être récupérés.

4.3.2.2 Étape 2 – Récupération des documents

Pour chaque requête, nous avons demandé au système de retourner les k documents les plus similaires selon FAISS. Avec un k relativement élevé pour ne pas rater des documents pertinents, dans ce cas k = 15. Chaque document retourné est accompagné de son score de similarité par rapport à la requête.

4.3.2.3 Étape 3 – Évaluation des résultats

Les performances sont mesurées à l’aide des métriques classiques [39] :

- **Precision@K** : proportion de documents récupérés qui sont réellement pertinents.

$$\text{Precision@K} = \frac{\text{Nombre d'éléments pertinents dans les K top recommandations}}{K}$$

- **Recall@K** : proportion de documents pertinents qui ont été retrouvés.

$$\text{Recall@K} = \frac{\text{Nombre d'éléments pertinents dans les K top recommandations}}{\text{Nombre total d'éléments pertinents}}$$

— **F1-score@K** : moyenne harmonique entre la précision et le rappel.

$$\text{F1-score@K} = 2 \times \frac{\text{Precision@K} \times \text{Recall@K}}{\text{Precision@K} + \text{Recall@K}}$$

Voici les résultats dans la figure 4.3 et 4.4 :

Query: How do I save the current user consent status? Expected: ('setConsent 2/6', 'setConsent 1/6', 'setConsent 6/6', 'setConsent 5/6', 'setConsent 4/6', 'setConsent 3/6') Retrieved: (setConsent 3/6 (score=0.5312), setConsent 4/6 (score=0.5259), setConsent 1/6 (score=0.5243), setConsent 6/6 (score=0.5122), setConsent 5/6 (score=0.5085), setConsent 2/6 (score=0.5008), getConsentState 2/6 (score=0.4618), checkIfNeedsToShow 8/9 (score=0.4562), getConsentState 3/6 (score=0.4553), getConsentState 4/6 (score=0.4501), checkIfNeedsToShow 9/9 (score=0.4495), checkIfNeedsToShow 2/9 (score=0.4384), getConsentState 1/6 (score=0.4366), checkIfNeedsToShow 6/9 (score=0.4351), checkIfNeedsToShow 3/9 (score=0.4321)) Precision@15: 0.40 Recall@15: 1.00 F1@15: 0.57 Query: How to get the current user consent setting? Expected: ('getConsentState 2/6', 'getConsentState 1/6', 'getConsentState 3/6', 'getConsentState 6/6', 'getConsentState 5/6', 'getConsentState 4/6') Retrieved: (getConsentState 2/6 (score=0.4876), getConsentState 3/6 (score=0.4742), setConsent 3/6 (score=0.4729), getConsentState 4/6 (score=0.4621), setConsent 4/6 (score=0.4538), setConsent 1/6 (score=0.4529), getConsentState 1/6 (score=0.4461), getConsentState 5/6 (score=0.4431), setConsent 6/6 (score=0.4423), getConsentState 6/6 (score=0.4384), setConsent 2/6 (score=0.4314), setConsent 5/6 (score=0.4306), checkIfNeedsToShow 8/9 (score=0.4278), checkIfNeedsToShow 9/9 (score=0.4277)) Precision@15: 0.40 Recall@15: 1.00 F1@15: 0.57	3/8 (score=0.5326), compute_attack_results 5/8 (score=0.5304), compute_attack_results 2/8 (score=0.5254)) Precision@15: 0.33 Recall@15: 0.62 F1@15: 0.43 Query: which method can be used delete rows from a Parquet file based on a list of values to remove from specific columns? Expected: ('get_row_indexes_to_delete 6/7', 'get_row_indexes_to_delete 1/7', 'get_row_indexes_to_delete 5/7', 'get_row_indexes_to_delete 7/7', 'get_row_indexes_to_delete 3/7', 'get_row_indexes_to_delete 4/7', 'get_row_indexes_to_delete 2/7') Retrieved: (get_row_indexes_to_delete_for_composite 1/12 (score=0.7412), get_row_indexes_to_delete_for_composite 7/12 (score=0.7317), get_row_indexes_to_delete_for_composite 10/12 (score=0.7258), get_row_indexes_to_delete_for_composite 6/12 (score=0.7254), delete_matches_from_parquet_file 8/8 (score=0.7233), get_row_indexes_to_delete_for_composite 8/12 (score=0.7227), get_row_indexes_to_delete_for_composite 5/12 (score=0.7212), delete_matches_from_parquet_file 3/8 (score=0.7190), get_row_indexes_to_delete 2/7 (score=0.7164), get_row_indexes_to_delete_for_composite 11/12 (score=0.7123), get_row_indexes_to_delete 1/7 (score=0.7109), get_row_indexes_to_delete_for_composite 3/12 (score=0.7105), get_row_indexes_to_delete 7/7 (score=0.7089), delete_from_table 3/5 (score=0.7077), get_row_indexes_to_delete_for_composite 12/12 (score=0.7076)) Precision@15: 0.20 Recall@15: 0.43 F1@15: 0.27
--	--

FIGURE 4.3 – Résultats d'évaluation (FAISS) (1)

Query: method that retrieves accepted Categories from cookie object Expected: ('acceptedCategories 1/1') Retrieved: (acceptedCategories 1/1 (score=0.5661), dump 7/7 (score=0.4529), dump 2/7 (score=0.4565), load 3/3 (score=0.4416), dump 6/7 (score=0.4408), getCookie 3/3 (score=0.4355), getCookie 1/3 (score=0.4306), load 1/3 (score=0.4261), _getCookie 2/3 (score=0.4260), dump 5/7 (score=0.4196), status 1/1 (score=0.4074), dump 4/7 (score=0.4054), dump 3/7 (score=0.3930)) Precision@15: 0.07 Recall@15: 1.00 F1@15: 0.12 Query: Which method computes attack metrics from MIA scores? Expected: ('compute_attack_results 8/8', 'compute_attack_results 3/8', 'compute_attack_results 4/8', 'compute_attack_results 2/8', 'compute_attack_results 6/8', 'compute_attack_results 1/8', 'compute_attack_results 7/8', 'compute_attack_results 5/8') Retrieved: (compute_abstain_attack_results_for_k_pos_k_neg 1/9 (score=0.5837), compute_abstain_attack_results_for_k_pos_k_neg 7/9 (score=0.5818), compute_abstain_attack_results_for_k_pos_k_neg 8/9 (score=0.5807), compute_abstain_attack_results_for_k_pos_k_neg 4/9 (score=0.5801), compute_abstain_attack_results_for_k_pos_k_neg 2/9 (score=0.5782), compute_abstain_attack_results_for_k_pos_k_neg 5/9 (score=0.5759), compute_abstain_attack_results_for_k_pos_k_neg 6/9 (score=0.5728), compute_abstain_attack_results_for_k_pos_k_neg 9/9 (score=0.5675), compute_attack_results 1/8 (score=0.5622), compute_abstain_attack_results_for_k_pos_k_neg 3/9 (score=0.5594), compute_abstain_attack_results 1/22 (score=0.5555), compute_attack_results 8/8 (score=0.5396), compute_attack_results	Query: How can I determine if a user needs to be asked for GDPR consent Expected: ('checkIfNeedsToShow 3/9', 'checkIfNeedsToShow 7/9', 'checkIfNeedsToShow 4/9', 'checkIfNeedsToShow 2/9', 'checkIfNeedsToShow 9/9', 'checkIfNeedsToShow 1/9', 'checkIfNeedsToShow 6/9', 'checkIfNeedsToShow 8/9', 'checkIfNeedsToShow 5/9') Retrieved: (checkIfNeedsToShow 8/9 (score=0.5746), checkIfNeedsToShow 9/9 (score=0.5709), checkIfNeedsToShow 2/9 (score=0.5614), checkIfNeedsToShow 6/9 (score=0.5482), checkIfNeedsToShow 3/9 (score=0.5406), checkIfNeedsToShow 5/9 (score=0.5401), checkIfNeedsToShow 1/9 (score=0.5380), checkIfNeedsToShow 7/9 (score=0.5332), checkIfNeedsToShow 4/9 (score=0.5222), setConsent 3/6 (score=0.5035), getConsentState 3/6 (score=0.4993), withExplicitNonPersonalisedConfirmation 1/1 (score=0.4956), getConsentState 2/6 (score=0.4842), setConsent 6/6 (score=0.4822)) Precision@15: 0.60 Recall@15: 1.00 F1@15: 0.75 --- Moyennes sur toutes les requêtes --- Precision@15: 0.33 Recall@15: 0.84 F1@15: 0.45
---	--

FIGURE 4.4 – Résultats d'évaluation (FAISS) (2)

Les requêtes bien formulées ont permis d'obtenir un rappel moyen de 0.84, ce qui indique que la majorité des documents pertinents étaient bien inclus dans les résultats initiaux de la recherche avec FAISS. Toutefois, la précision moyenne restait relativement faible (0.33), en raison de la présence de nombreux faux positifs.

Dans certains cas, les documents pertinents sont noyés parmi des faux positifs, ce qui nuit à la qualité globale du classement initial. Une analyse qualitative de ces faux positifs révèle que certains obtiennent des scores de similarité relativement élevés (souvent > 0.43), car ils sont sémantiquement proches de la requête.

À l'inverse, une autre partie des faux positifs affiche des scores de similarité faibles, ce qui confirme leur éloignement réel du sens de la requête.

Ces résultats justifient l'intérêt d'un raffinement des résultats obtenus par FAISS.

4.3.3 Amélioration via reranker

Afin d'améliorer la précision tout en éliminant les faux positifs, une étape de re-ranking (re-classement) a été introduite après la recherche initiale. Cette étape consiste à appliquer un modèle cross-encoder¹ sur les k segments retournés par FAISS, en évaluant chaque paire (requête, document). Cela permet de réorganiser les documents selon leur pertinence réelle et de garder uniquement les segments ayant une similarité élevée avec la requête, d'un point de vue sémantique. Pour cela, nous avons utilisé le modèle cross-encoder/ms-marco-MiniLM-L-6-v2.

Résultats après Reclassement :

Après l'application du "reranker", une nette amélioration a été observée sur les performances du système de retrieval. Les documents réellement pertinents ont été priorisés, avec des scores élevés dépassant souvent 0.90. À l'inverse, les documents moins pertinents ont reçu des scores plus faibles et donc peuvent être filtré en ajoutant un seuil ce qui va permis de réduire significativement le nombre de faux positifs.

Les figures 4.5 et 4.6 illustrent les résultats de la récupération après l'application de reranker.

<pre>Query: How do I save the current user consent status? Expected: {'setConsent 2/6', 'setConsent 1/6', 'setConsent 6/6', 'setConsent 5/6', 'setConsent 4/6', 'setConsent 3/6'} Retrieved (avec scores reranker): (setConsent 6/6 (score=0.9174), setConsent 5/6 (score=0.8015), setConsent 1/6 (score=0.7836), setConsent 2/6 (score=0.7739), setConsent 4/6 (score=0.7053), setConsent 3/6 (score=0.6969), getConsentState 1/6 (score=0.5327), getConsentState 4/6 (score=0.4232), getConsentState 3/6 (score=0.4036)) Precision@9: 0.67 Recall@9: 1.00 F1@9: 0.80 Query: How to get the current user consent setting? Expected: {'getConsentState 2/6', 'getConsentState 1/6', 'getConsentState 3/6', 'getConsentState 6/6', 'getConsentState 5/6', 'getConsentState 4/6'} Retrieved (avec scores reranker): (getConsentState 4/6 (score=0.9595), getConsentState 6/6 (score=0.9568), getConsentState 2/6 (score=0.9557), getConsentState 3/6 (score=0.9523), getConsentState 1/6 (score=0.9512), getConsentState 5/6 (score=0.9259), setConsent 4/6 (score=0.6867), setConsent 2/6 (score=0.6545), setConsent 3/6 (score=0.6103)) Precision@9: 0.67 Recall@9: 1.00 F1@9: 0.80</pre>	<pre>Query: method that retrieves accepted Categories from cookie object Expected: {'acceptedCategories 1/1'} Retrieved (avec scores reranker): (acceptedCategories 1/1 (score=0.8930), load 3/3 (score=0.6675), status 1/1 (score=0.1207), dump 7/7 (score=0.0827), dump 6/7 (score=0.0759), _getCookie 2/3 (score=0.0714), dump 5/7 (score=0.0682), dump 3/7 (score=0.0581), dump 4/7 (score=0.0564)) Precision@9: 0.11 Recall@9: 1.00 F1@9: 0.20</pre>
--	---

FIGURE 4.5 – Résultats d'évaluation (reranker) (1)

1. Un cross-encoder est un modèle de réseau de neurones, généralement basé sur les Transformers qui traite des paires d'entrées ensemble et produit un score indiquant leur relation ou leur similarité.

Query: Which method computes attack metrics from MIA scores? Expected: ('compute_attack_results 8/8', 'compute_attack_results 3/8', 'compute_attack_results 4/8', 'compute_attack_results 2/8', 'compute_attack_results 6/8', 'compute_attack_results 1/8', 'compute_attack_results 7/8', 'compute_attack_results 5/8') Retrieved (avec scores reranker): ('compute_attack_results 8/8 (score=0.9987)', 'compute_attack_results 5/8 (score=0.9987)', 'compute_attack_results 3/8 (score=0.9984)', 'compute_attack_results 2/8 (score=0.9983)', 'compute_abstain_attack_results_for_k_pos_k_neg 9/9 (score=0.9982)', 'compute_attack_results 1/8 (score=0.9981)', 'compute_abstain_attack_results 1/22 (score=0.9977)', 'compute_abstain_attack_results_for_k_pos_k_neg 7/9 (score=0.9973)', 'compute_abstain_attack_results_for_k_pos_k_neg 1/9 (score=0.9968)') Precision@9: 0.56 Recall@9: 0.62 F1@9: 0.59 Query: which method can be used delete rows from a Parquet file based on a list of values to remove from specific columns? Expected: ('get_row_indexes_to_delete 6/7', 'get_row_indexes_to_delete 1/7', 'get_row_indexes_to_delete 5/7', 'get_row_indexes_to_delete 7/7', 'get_row_indexes_to_delete 3/7', 'get_row_indexes_to_delete 4/7', 'get_row_indexes_to_delete 2/7') Retrieved (avec scores reranker): ('get_row_indexes_to_delete 7/7 (score=0.9993)', 'get_row_indexes_to_delete 1/7 (score=0.9992)', 'get_row_indexes_to_delete_for_composite 12/12 (score=0.9990)', 'get_row_indexes_to_delete 2/7 (score=0.9990)', 'get_row_indexes_to_delete_for_composite 1/12 (score=0.9989)',	get_row_indexes_to_delete_for_composite 7/12 (score=0.9986), get_row_indexes_to_delete_for_composite 10/12 (score=0.9986), get_row_indexes_to_delete_for_composite 11/12 (score=0.9985), get_row_indexes_to_delete_for_composite 3/12 (score=0.9985)') Precision@9: 0.33 Recall@9: 0.43 F1@9: 0.37 Query: How can I determine if a user needs to be asked for GDPR consent Expected: ('checkIfNeedsToShow 3/9', 'checkIfNeedsToShow 7/9', 'checkIfNeedsToShow 4/9', 'checkIfNeedsToShow 2/9', 'checkIfNeedsToShow 9/9', 'checkIfNeedsToShow 1/9', 'checkIfNeedsToShow 6/9', 'checkIfNeedsToShow 8/9', 'checkIfNeedsToShow 5/9') Retrieved (avec scores reranker): ('checkIfNeedsToShow 6/9 (score=0.9846)', 'checkIfNeedsToShow 7/9 (score=0.9836)', 'checkIfNeedsToShow 5/9 (score=0.9826)', 'checkIfNeedsToShow 2/9 (score=0.9775)', 'checkIfNeedsToShow 1/9 (score=0.9774)', 'checkIfNeedsToShow 3/9 (score=0.9770)', 'checkIfNeedsToShow 4/9 (score=0.9754)', 'checkIfNeedsToShow 9/9 (score=0.9751)', 'checkIfNeedsToShow 8/9 (score=0.9694)') Precision@9: 1.00 Recall@9: 1.00 F1@9: 1.00 --- Moyennes sur toutes les requêtes --- Precision@9: 0.56 Recall@9: 0.84 F1@9: 0.63
--	--

FIGURE 4.6 – Résultats d'évaluation (reranker) (2)

L'application d'un reranker permet d'écarter les faux positifs les plus éloignés tout en conservant ceux qui présentent une sémantique proche des documents cibles. Cette amélioration du classement s'est traduite par une hausse de la précision moyenne, qui est passée de 0.33 à 0.56. Par conséquent, le score F1 global a connu une progression notable, passant de 0.45 à 0.63, confirmant l'intérêt de cette étape de reclassement pour améliorer la qualité des documents fournis au système de génération.

4.3.4 Conclusion des résultats

La combinaison de FAISS et d'un reranker basé sur un cross-encoder s'est révélée particulièrement efficace pour améliorer la qualité du système de récupération d'information dans le cadre du pipeline RAG. Dans un premier temps, l'utilisation de FAISS avec une valeur de k relativement élevée permet de récupérer une grande diversité de documents, ce qui garantit que les documents pertinents sont bien inclus parmi les résultats, même si certains faux positifs peuvent également apparaître.

Ensuite, l'application du reranker sur ces k documents permet de réordonner les segments en fonction de leur pertinence réelle vis-à-vis de la requête. Cette étape filtre les segments moins pertinents (généralement associés à un score de similarité plus faible) et priorise ceux qui sont les plus informatifs, souvent avec des scores proches de 1. Cette méthode maximise ainsi la précision, tout en maintenant un rappel élevé, ce qui est essentiel pour garantir une couverture complète de la réponse.

Enfin, les documents reclassés sont transmis au LLM, ce qui permet de générer des réponses contextualisées, précises et ancrées dans les bons segments du corpus.

4.3.5 Évaluation de la génération des réponses

Après l'étape de récupération des documents pertinents à l'aide de FAISS et du reranker cross-encoder, nous avons évalué la capacité du LLM (Mistral) à générer une réponse structurée, fidèle et exhaustive à partir des segments fournis.

L'évaluation visait à vérifier trois aspects essentiels : la pertinence de la réponse vis-à-vis de la requête, la fidélité de la réponse au contenu des documents sans hallucination, et la couverture multi-projet avec une gestion appropriée des requêtes hors sujet. Le prompt injecté incluait des instructions claires sur l'objectif de la requête, la structuration attendue de la réponse, et la base documentaire sur laquelle s'appuyer.

4.3.5.1 Étape 1 : Analyse de la réponse générée

La réponse produite par le modèle a démontré une compréhension approfondie des documents fournis. Par exemple elle identifie la méthode `setConsent()` et son usage avec `SharedPreferences` pour stocker le consentement utilisateur dans le contexte Android/Java. La figure 4.7 présente un exemple de réponse générée par le modèle à partir d'un projet Java.

```
 **Chunk source**:  
File: library/src/main/java/com/michaelflisar/gdprdialog/GDPR.java  
Type: method_body  
Method: setConsent  
  
**Practice Label**: Setting and Persisting Consent State  
  
**Purpose**:  
This practice ensures that the current consent state is set and persisted in the preferences. It logs the consent state and success status  
  
**GDPR Reference**:  
Article 7 (Conditions for consent)  
  
**Implementation Details**:  
```java  
public boolean setConsent(GDPRConsentState consentState) {
 mCachedConsent = consentState;
 boolean success = mPreferences.edit()
 .putLong(mContext.getString(R.string.gdpr_preference_date), consentState.getDate())
 .putInt(mContext.getString(R.string.gdpr_preference_app_version), consentState.getVersion())
 .putInt(mContext.getString(R.string.gdpr_preference_is_in_eea_or_unknown), consentState.getLocation().ordinal())
 .putInt(mContext.getString(R.string.gdpr_preference), consentState.getConsent().ordinal())
 .commit();

 mLogger.debug("GDPR", String.format("consent saved: %s, success: %b", consentState.logString(), success));
 return success;
}
```
```

FIGURE 4.7 – Réponse basée sur un projet java

Chaque pratique identifiée est accompagnée d'un extrait de code réel, d'un guide de mise en œuvre pratique, et d'un rappel du principe RGPD correspondant, tel que l'Article 7.

La figure suivante présente un guide pour la gestion de l'état de consentement des utilisateurs.

```

**Guide**:
1. **Consent State Assignment**: Assign the current consent state to a cached variable.
2. **Preferences Editing**: Edit the shared preferences to store the consent state details.
3. **Commit Changes**: Commit the changes to the shared preferences.
4. **Logging**: Log the consent state and success status for verification.

**Recommended Libraries / Tools or Technical Mechanisms**:
- **SharedPreferences**: For storing and retrieving consent states.
- **Logger**: For logging consent-related information.

```

FIGURE 4.8 – Exemple d'un guide pratique

L'analyse s'étend également à la configuration de default_options.js pour l'affichage et la catégorisation des cookies dans l'environnement Web/JavaScript. La figure 4.9 montre une réponse extraite dans le contexte d'un projet JavaScript.

```

**Chunk source**:
File: src/default_options.js
Type: properties

**Practice Label**: Cookie Consent Dialog Configuration

**Purpose**:
This practice ensures that the cookie consent dialog is configured with default options, including the title, description, buttons, and categories of cookies.

**GDPR Reference**:
Article 7 (Conditions for consent)

**Implementation Details**:
...
javascript
const defaultOptions = {
  title: "We use cookies",
  description: "Click "Accept" to enable us to use cookies to personalize
this site. Customize your preferences in your
Cookie Settings or click "Reject" if you do not want us
to use cookies for this purpose. Learn more in our


```

FIGURE 4.9 – Réponse basée sur un projet js

4.3.5.2 Étape 2 : Validation de la fidélité et de la pertinence

Chaque pratique identifiée dans la réponse est explicitement associée à un segment source spécifique. Par exemple, la méthode setConsent dans un fichier Java (GDPR.java), et l'objet defaultOptions est dans un fichier JavaScript (default_options.js). Cette traçabilité garantit la transparence du processus de génération.

Le contenu des extraits générés, incluant le code et les explications, correspond directement aux documents fournis sans aucune information inventée. Le modèle évite de générer des pratiques génériques ou hors sujet et s'appuie clairement sur les extraits fournis lors de la phase de récupération.

La génération démontre une capacité à couvrir des fichiers issus de différents langages de programmation. Elle intègre un projet Android en Java (selon le fichier GDPR.java) avec un

projet en JavaScript (cookie_consent.js, default_options.js). Cette approche multi-projet illustre que le LLM a été capable de fusionner les informations provenant de plusieurs projets pour formuler une réponse adaptée. Enfin, dans le cas de requêtes hors sujet, le système informe l'utilisateur que sa demande ne peut être prise en charge.

4.3.5.3 Étape 3 : Validation d'expert en RGPD

Afin d'évaluer la fidélité juridique des résultats produits par le système, une validation a été réalisée par une experte du domaine, doctorante en informatique, justifiant de quatre années d'expérience dans le domaine de la protection des données personnelles (RGPD).

Cette validation a porté sur la pertinence des pratiques détectées, leur conformité aux articles du Règlement général sur la protection des données, ainsi que sur la cohérence des recommandations générées. L'experte a confirmé que les pratiques identifiées (telles que la gestion du consentement, le droit à l'effacement ou la sécurisation des données) sont bien alignées avec les exigences du RGPD, et que les extraits de code proposés étaient représentatifs de leur implémentation effective. Ce retour d'expert renforce la crédibilité des résultats et la validité du système proposé.

4.4 Test du système : Analyse des Pratiques RGPD dans des Projets Open Source

Cette section présente les résultats de notre étude empirique sur l'implémentation des concepts du RGPD dans le code source d'un échantillon de projets collectés. À partir d'une base vectorielle alimentée par des représentations AST de quatre projets dédiés à la conformité RGPD, nous avons testé notre système RAG afin d'identifier des pratiques techniques concrètes.

Les projets utilisés sont :

- GDPRDialog (Java) : gestion du consentement dans les applications Android.
- cookie-consent (JavaScript) : gestion du consentement via cookies pour les sites web.
- amazon-s3-find-and-forget (Python) : effacement de données, chiffrement et sécurité des données.
- ml-privacy-meter (Python) : protection des données, évaluation, impact.

4.4.1 Partie 1 : Identification des pratiques de conformité

Pour chaque concept identifié dans les projets, nous avons formulé des requêtes ciblées. Les résultats sont présentés sous forme de tableaux synthétiques, suivis de conclusions pour chaque concept du RGPD.

(a) Gestion du consentement

Requête : *What practices are used for consent management and cookie management*

Cette requête a permis de couvrir les différentes étapes du cycle de gestion du consentement, aussi bien côté mobile que web. Le tableau 4.3 synthétise les pratiques extraites.

| Pratique Implémentée | Éléments utilisés | Mécanisme technique | Référence RGD |
|--|--|--|--------------------|
| Vérification et demande de consentement | Méthode <code>checkIfNeedsToBeShown</code> | Vérification de l'état courant et déclenchement d'un dialogue conditionnel | Art. 7, Art. 4(11) |
| Stockage du consentement utilisateur | Méthode <code>setConsent.SharedPreferences.Editor</code> | Sérialisation et écriture des choix utilisateur (consentement, date, version, etc.) dans le stockage persistant de l'application | Art. 7 |
| Lecture du consentement utilisateur | Méthode <code>getConsentState.SharedPreferences</code> | Récupération du consentement stocké, et construction d'un objet représentant l'état utilisateur | Art. 7 |
| Réinitialisation de l'état de consentement | Méthode <code>resetConsent</code> , Constructeur <code>new GDPRConsentState()</code> | Réinitialisation explicite de l'état de consentement vers un état vide | Art. 7 |
| Confirmation explicite pour données non personnalisées | Méthode <code>withExplicitNonPersonalisedConfirmation</code> | Configuration fluide du consentement explicite | Art. 7 |
| Gestion du consentement cookie | Classe <code>CookieConsent</code> | Interface configurable pour la collecte du consentement | Art. 7 |

TABLE 4.3 – Pratiques d'implémentation de la gestion du consentement

Les projets étudiés mettent en œuvre des pratiques de gestion du consentement qui interviennent à différents niveaux, intégrant à la fois des mécanismes techniques (stockage, callbacks, interfaces) et des bonnes pratiques UX (dialogues explicites, gestion granulaire des cookies).

(b) Droit à l'effacement

Requête : *What are the technical practices implemented in the code to support the right to erasure (deleting user data) under GDPR ?*

La figure 4.10 illustre un exemple des pratiques identifiées en lien avec les droits d'effacement.

```


**Chunk source**: `downloaded_files/s3.py`

**Practice Label**: Deleting S3 Objects

**Purpose**:
This practice ensures that specified objects can be deleted from an S3 bucket, adhering to the GDPR's right to erasure.

**GDPR Reference**:
Article 17 (Right to erasure)

**Implementation Details**:

```
python
def delete_s3_objects(client, bucket, objects):
 """
 Deletes specified objects from an S3 bucket.
 """
 return client.delete_objects(Bucket=bucket, Delete={'Objects': objects, 'Quiet': True})

```

**Guide**:
1. **Initialize S3 Client**: Ensure you have an initialized S3 client.
2. **Call Delete Method**: Use the `delete_objects` method of the S3 client to delete the specified objects.

**Recommended Libraries / Tools or Technical Mechanisms**:
- **Boto3**: For interacting with AWS S3.

```

FIGURE 4.10 – Exemple de pratiques liées aux droits d’effacement

Cette requête a ciblé les mécanismes liés à l’effacement des données.

| Pratique Implémentée | Élément principal utilisé | Mécanisme technique | Référence RGPD |
|--|---|---|-----------------------|
| Identification des lignes à supprimer | Méthode <code>get_row_indexes_to_delete</code> (Python) | Itération sur une colonne d'ids pour trouver les identifiants d'utilisateur à supprimer | Art. 17 |
| Suppression par clé composite | Méthode <code>get_row_indexes_to_delete_for_composite</code> (Python) | Construction de clés composites (ex : prénom+nom) et filtrage par correspondance dans une liste cible | Art. 17 |
| Suppression d'objets dans un bucket S3 | Fonction <code>delete_s3_objects</code> | Appel API <code>delete_objects</code> via boto3 pour suppression définitive dans Amazon S3 | Art. 17 |

TABLE 4.4 – Pratiques d'implémentation de l'effacement des données

Le tableau 4.4 présente des pratiques techniques concrètes d'effacement de données, en lien avec l'article 17 du RGPD. Il illustre des approches variées selon le contexte technique : suppression ciblée de lignes via identifiants, filtrage par clé composite, et suppression définitive d'objets dans un bucket S3. Ces pratiques montrent une adaptation aux différents types de structures de données tout en respectant les exigences de conformité.

(c) **Intégrité et Confidentialité**

Requête : *What practices are implemented to ensure the confidentiality and integrity of personal data ?*

La figure 4.11 illustre le résultat d'une requête portant sur les pratiques liées à la confidentialité et à l'intégrité des données personnelles.

```

 ** Chunk source**:
File: downloaded_files/cse.py
Type: method_body
Method: encrypt

** GDPR Principle**: Integrity and Confidentiality

** What should be done according to the principle**:
Implement encryption to protect personal data from unauthorized access and ensure data integrity.

** Practice Label**: Encrypting S3 Objects with KMS-based CSE

** Purpose**:
The `encrypt` method encrypts S3 objects using KMS-based Client-Side Encryption (CSE). It uses the object's metadata to fetch the encryption envelope,

** GDPR Reference**:
Article 32 (Security of processing)

** Implementation Details**:
```python
def encrypt(buf, s3_metadata, kms_client):
 logger.info("Encrypting Object with CSE-KMS")
 content = buf.read()
 alg = s3_metadata.get(HEADER_ALG, None)
 matdesc = json.loads(s3_metadata[HEADER_MATDESC])
 aes_key, matdesc_metadata, key_metadata = get_encryption_aes_key(matdesc["kms_cmek_id"], kms_client)

 if alg == ALG_GCM:
 result, iv = encrypt_gcm(aes_key, content)
 s3_metadata[HEADER_TAG_LEN] = str(AES_BLOCK_SIZE)
 else:
 result, iv = encrypt_cbc(aes_key, content)

 s3_metadata[HEADER_IV] = base64.b64encode(iv).decode()
 s3_metadata[HEADER_WRAP_ALG] = "kms"
 s3_metadata[HEADER_KEY] = key_metadata
 s3_metadata[HEADER_UE_LENGTH] = str(len(content))

 return BytesIO(result), s3_metadata
```

```

FIGURE 4.11 – Exemple des pratiques liées à l'intégrité et Confidentialité

Cette requête a permis d'identifier un ensemble de pratiques liées au chiffrement, à la gestion de clés et au journal des opérations sensibles (voir tab 4.5).

| Concept RGPD | Pratique Implémentée | Élément utilisé principal | Mécanisme technique | Référence RGPD |
|-----------------------------|----------------------------------|--|--|----------------|
| Intégrité & Confidentialité | Chiffrement côté client avec KMS | Fonction <code>get_encryption_aes_key</code> | Génération d'une clé AES via AWS KMS et chiffrement du contenu avant envoi | Art. 32(1)(a) |
| Intégrité & Confidentialité | Chiffrement AES en mode CBC | Fonction <code>encrypt_cbc</code> | Chiffrement symétrique via AES (PyCryptodome), avec IV aléatoire et padding PKCS7 | Art. 32(1)(a) |
| Intégrité & Confidentialité | Déchiffrement AES en mode CBC | Fonction <code>decrypt_cbc</code> | Déchiffrement symétrique à l'aide d'une clé AES et d'un vecteur IV | Art. 32(1)(a) |
| Intégrité & Confidentialité | Encodage Base64 | Fonction <code>encode_base64</code> / <code>decode_base64</code> | Transformation des données sensibles (clé, IV) en format base64 pour stockage sécurisé | Art. 32(1)(a) |

| Concept RGPD | Pratique Implémentée | Élément utilisé principal | Mécanisme technique | Référence RGPD |
|-----------------------------|--|-------------------------------|---|----------------|
| Intégrité & Confidentialité | Gestion des métadonnées pour le chiffrement | Fonction get_metadata | Extraction des clés d'enveloppe depuis les métadonnées S3 (algorithme, clé, IV) | Art. 32(1)(a) |
| Intégrité & Confidentialité | Journalisation des opérations de chiffrement | Fonction log_encryption_start | Logging via logging.info() pour tracer les opérations de sécurité | Art. 32(1)(a) |
| Intégrité & Confidentialité | Journalisation des opérations de déchiffrement | Fonction log_decryption_start | Logging pour les processus de déchiffrement | Art. 32(1)(a) |

TABLE 4.5 – Pratiques d'implémentation de l'intégrité et Confidentialité

Les projets mettent en œuvre des pratiques avancées de sécurité. L'intégration de mécanismes comme le chiffrement client, l'encodage, et la journalisation illustre une conformité au principe d'intégrité et de confidentialité.

(d) Principe de Responsabilité

Requête : *What practices are used to evaluate the privacy risks of model predictions or data exposure ?*

La figure 4.12 présente un exemple des pratiques identifiées par la requête posée :

```

**Chunk source**:
File: ml_privacy_meter/audit.py
Type: method_body
Method: audit_models

**Practice Label**: Auditing Models

**Purpose**:
The 'audit_models' method audits target models using a Membership Inference Attack algorithm. It takes various inputs such as signal values, membership

**GDPR Reference**:
Article 5(2) - Accountability: The controller shall be responsible for, and be able to demonstrate compliance with, paragraph 1 ('accountability').

**Implementation Details**:
```python
import numpy as np
import logging

def audit_models(report_dir, target_model_indices, all_signals, population_signals, all_memberships, num_reference_models, logger, configs):
 mia_score_list = []
 membership_list = []

 for target_model_idx in target_model_indices:
 # Perform MIA using the specified algorithm
 if configs['audit']['algorithm'] == 'LOSS':
 mia_scores = run_loss(all_signals[:, target_model_idx])
 elif configs['audit']['algorithm'] == 'RMIA':
 mia_scores = run_rmia(target_model_idx)
 else:
 raise NotImplementedError(f"{configs['audit']['algorithm']} is not implemented")

 target_memberships = all_memberships[:, target_model_idx]
 mia_score_list.append(mia_scores.copy())
 membership_list.append(target_memberships.copy())

 logger.info(f"Auditing the privacy risks of target model {target_model_idx} costs {time.time() - baseline_time} seconds")

 return mia_score_list, membership_list

```

FIGURE 4.12 – Exemple des pratiques identifiées

Ce tableau présente une synthèse complète des pratiques techniques identifiées pour

assurer la conformité RGPD dans les systèmes d'apprentissage automatique. Il se concentre spécifiquement sur le principe de responsabilité (Accountability) tel que défini par l'Article 5(2) du RGPD, qui exige des contrôleurs de données qu'ils soient en mesure de démontrer leur conformité.

Concept RGPD	Pratique Implémentée	Éléments utilisés	Mécanisme technique	Référence RGPD
Responsabilité (Accountability)	Débiaisage des Prédictions	Signaux MIA et drapeaux d'appartenance des modèles de référence	Utilise les TPR/FPR des modèles de référence pour corriger les prédictions biaisées sur la présence de données dans l'entraînement.	Article 5(2)
Responsabilité (Accountability)	Audit des Modèles	Algorithme d'Attaque par Inférence d'Appartenance (MIA)	Analyse des signaux pour détecter si certaines données d'entraînement peuvent être inférées.	Article 5(2)
Responsabilité (Accountability)	Exécution d'Algorithmes MIA	Algorithme RMIA (Relative Membership Inference Attack)	Attaque pour tester si un échantillon faisait partie ou non de l'entraînement.	Article 5(2)
Responsabilité (Accountability)	Réglage de Paramètres Hors Ligne	Paramètre 'a' hors ligne	Optimisation de la précision de l'évaluation des risques de confidentialité.	Article 5(2)
Responsabilité (Accountability)	Journalisation des Résultats d'Audit	Scores MIA et étiquettes d'appartenance	Enregistre les scores et risques détectés pour vérifiabilité.	Article 5(2)

TABLE 4.6 – Pratiques d'implémentation de responsabilité

Toutes ces pratiques visent à implémenter des mécanismes d'évaluation des risques de confidentialité des prédictions de modèles ou de l'exposition de données, garantissant que les activités de traitement des données sont transparentes et responsables conformément au principe de responsabilité du RGPD.

### Synthèse des résultats

Les pratiques extraites à partir des segments de code issus de projets dédiés au RGPD constituent une ressource précieuse pour les développeurs souhaitant aligner leurs systèmes avec les exigences de protection des données personnelles. Le système permet d'identifier des exemples concrets d'implémentation (stockage du consentement, affichage de fenêtres de cookies, mécanismes de minimisation, etc.) déjà intégrés dans des solutions conformes. Ces pratiques peuvent ensuite être réutilisées ou adaptées dans d'autres projets afin de

faciliter leur mise en conformité. Ainsi, le système est capable de guider les développeurs dans l'intégration de ces pratiques au sein de leur code.

Afin d'évaluer l'apport de notre système par rapport à une approche classique, nous proposons une comparaison entre un LLM généraliste (Mistral utilisé seul) et notre système RAG spécialisé pour le RGPD.

(e) **Comparaison entre un LLM généraliste (Mistral) et notre système RAG spécialisé RGPD**

Après avoir analysé les réponses générées par notre système RAG, nous avons mené une comparaison entre ces réponses et les réponses générées par un modèle de langage généraliste, Mistral. L'objectif est d'évaluer la capacité du système à fournir des pratiques concrètes, alignées avec les exigences du RGPD, et potentiellement intégrables dans des projets logiciels.

**Requête :** *“What are the practices used for consent management ?”*

### Réponse du LLM généraliste (Mistral)

Le modèle Mistral fournit une réponse théorique et descriptive (voir figure 4.13) couvrant les bonnes pratiques en matière de gestion du consentement.

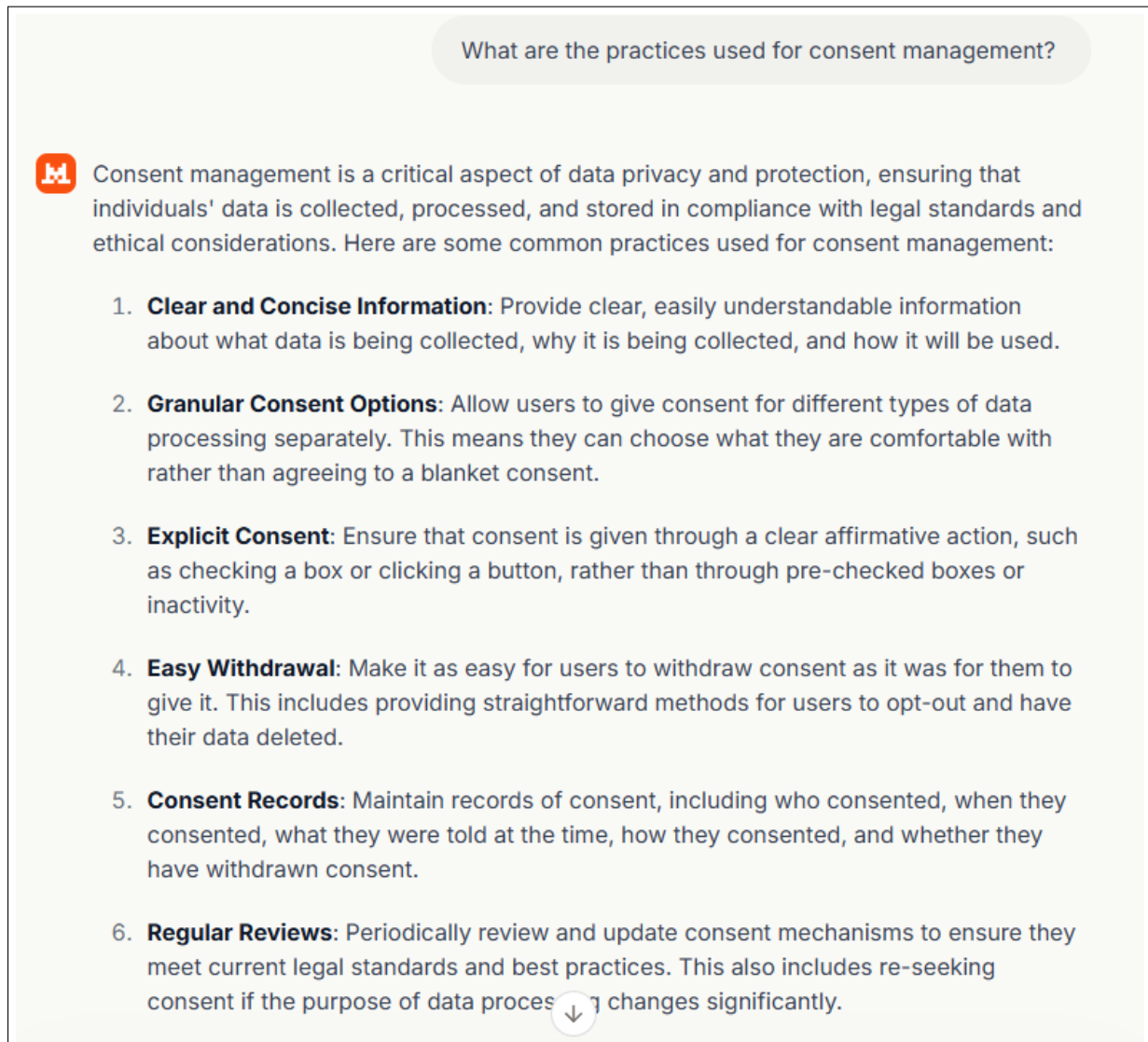


FIGURE 4.13 – Principales pratiques de gestion du consentement identifiées à l’aide du Mistral

Le modèle Mistral fournit une réponse théorique et générale, centrée sur des recommandations classiques de gestion du consentement :

- Fournir une information claire et concise.
- Offrir des options granulaires de consentement.
- Permettre un retrait facile du consentement.
- Tenir des registres des consentements.
- Vérification d’âge, gestion des tiers, etc.

Ces éléments sont pertinents et conformes aux recommandations réglementaires, mais ils ne sont pas accompagnés de preuves concrètes, d’exemples de mise en œuvre ni de

liens avec des cas réels issus du développement logiciel. Même après avoir demandé des exemples de code, la réponse était basée sur un seul concept : l'enregistrement de consentement.

Le système nécessite beaucoup d'interactions avec l'utilisateur pour fournir une réponse qui peut l'aider, ce qui prend du temps. De plus, les résultats obtenus posent question quant à leur source et leur validation par des développeurs expérimentés. Par contre dans un système RAG dédié à la conformité, les développeurs peuvent obtenir la réponse dès la première requête, accompagnée d'exemples de code basés sur des projets conformes.

Cette comparaison met en lumière l'intérêt d'un système RAG spécialisé, qui permet non seulement de fournir une réponse pertinente sur le plan réglementaire, mais orientée vers la mise en œuvre concrète. L'intégration d'une base de données de projets conformes au RGPD apporte une valeur significative au modèle, en lui offrant une capacité à faire le lien entre la conformité théorique et sa mise en œuvre concrète dans le code.

#### **4.4.2 Partie 2 : Évaluation de la conformité par projet**

Afin d'évaluer la conformité des projets aux principes du RGPD, nous avons étendu notre système RAG pour qu'il prenne en charge l'analyse ciblée d'un projet spécifique mentionné dans la requête. Lorsque la requête utilisateur contient le nom d'un projet (par exemple : gdprdialog, cookieConsent, etc.), le système filtre les segments de code associés à ce projet, puis évalue s'il implémente ou non le principe RGPD mentionné dans la requête.

##### **a. Analyse contextualisée**

Le système fournit ensuite une analyse contextualisée, en précisant :

- Les pratiques réellement implémentées dans ce projet.
- Les écarts de conformité éventuels, c'est-à-dire les pratiques attendues mais absentes.
- Les propositions d'améliorations possibles pour couvrir les manques.
- Une évaluation qualitative de la couverture du principe demandé.

**Exemple :** Pour la requête *"Does gdprdialog implement consent management practices ?"*

**Évaluation :**

Détection des pratiques implémentées avec des détails techniques (figure 4.14) :

```

 ### Compliance Evaluation

Does the 'gdprdialog' project implement consent management practices?

Yes, the 'gdprdialog' project does implement consent management practices.

Mechanisms and Practices Found

1. **Consent Checking**:
 - The 'checkIfNeedsToShow' method determines if consent needs to be requested from the user.
 - It uses callbacks ('IGDPRCallback#onConsentNeedsToBeRequested' and 'IGDPRCallback#onConsentInfoUpdate') to handle consent-related actions.

2. **Consent Setting**:
 - The 'setConsent' method sets and persists the current consent state in shared preferences.
 - It logs the consent state and success status.

3. **Consent Retrieval**:
 - The 'getConsentState' method retrieves the current consent state from shared preferences.
 - It constructs a 'GDPRConsentState' object based on the retrieved data.

4. **Consent Resetting**:
 - The 'resetConsent' method resets the current consent state to an undefined state.

5. **Consent Dialog**:
 - The 'showDialog' method displays a dialog to the user to obtain consent.

6. **Explicit Consent for Non-Personalised Data**:
 - The 'withExplicitNonPersonalisedConfirmation' method in 'GDPRSetup' forces the user to give explicit consent for non-personalised data usage.

7. **Allow No Consent**:
 - The 'withAllowNoConsent' method in 'GDPRSetup' allows the user to use the app without giving consent.

```

FIGURE 4.14 – Exemple de détection des pratiques implémentées avec des détails techniques

Détection des pratiques manquantes (voir figure 4.15) :

```

Missing Parts or Gaps

1. **User Interface for Consent Management**:
 - While the project includes methods to show a consent dialog, the actual user interface and user experience for managing consent are not fully detailed in the provided code.

2. **Consent Withdrawal**:
 - There is no explicit method for users to withdraw their consent after it has been given. This is a crucial aspect of GDPR compliance.

3. **Consent Versioning**:
 - While the project stores the version of the consent, it is not clear how it handles updates to the consent policy and obtaining new consent from users when the policy changes.

4. **Consent Logging and Auditing**:
 - The project logs consent states, but there is no detailed mechanism for auditing consent changes over time, which is important for compliance and accountability.

Suggestions for Improvement

1. **Implement Consent Withdrawal**:
 - Add a method that allows users to withdraw their consent easily. This could be a button or option in the user interface that triggers the 'resetConsent' method.

2. **Enhance Consent Versioning**:
 - Implement a mechanism to notify users of changes to the consent policy and obtain new consent when the policy is updated.

3. **Improve Consent Logging and Auditing**:
 - Enhance the logging mechanism to include more detailed information about consent changes, including timestamps and user actions.
 - Implement an auditing system that allows for the review of consent history.

4. **User Interface for Consent Management**:
 - Develop a user-friendly interface for users to manage their consent preferences, including options to give, withdraw, and review their consent.

Compliance Score

Compliance Score: 7/10

```

FIGURE 4.15 – Exemple de détection des pratiques manquantes

Le projet implémente partiellement la gestion du consentement, mais ne permet pas aux utilisateurs de le retirer explicitement, ce qui représente un point de non-conformité important au regard de l'article 7 du RGPD. Le système propose ainsi des améliorations possibles pour une conformité complète au consentement selon le RGPD.

### Détection de l'absence complète d'implémentation

En cas où le projet n'implémente aucune pratique liée au concept donné, le système peut le détecter (figure 4.16) :

```

Evaluation of Consent Management in the 'downloaded_files' Project

Does the 'downloaded_files' project implement consent management?

No, the project does not appear to implement the GDPR principle of consent management based on the available code structure.

Mechanisms and Practices Found:
- None. The provided code snippets do not include any mechanisms or practices related to consent management.

Missing Parts or Gaps:
1. **User Consent Collection**: There are no mechanisms for collecting user consent.
2. **Consent Storage**: No indication of how user consent is stored or managed.
3. **Consent Withdrawal**: No mechanisms for users to withdraw their consent.
4. **Consent Verification**: No processes for verifying user consent before processing data.
5. **Consent Documentation**: No documentation or logging of user consent actions.

Suggestions for Improvement:
1. **Implement Consent Collection**:
 - Add mechanisms to collect user consent, such as consent forms or dialog boxes.
 - Example: A consent dialog box in `src/consent_box.html.js` or `src/consent_box.js`.
2. **Store Consent Information**:
 - Implement a database or storage solution to record user consent.
 - Example: A table in a database to store user consent status and timestamps.
3. **Provide Consent Withdrawal**:
 - Allow users to withdraw their consent easily.
 - Example: A user interface element to withdraw consent and update the consent status in the database.
4. **Verify Consent Before Data Processing**:
 - Ensure that user consent is verified before any data processing activities.
 - Example: Check the consent status in the database before processing user data.
5. **Document Consent Actions**:
 - Maintain logs of consent actions, including when consent was given or withdrawn.
 - Example: Log entries in a database or log file for consent actions.

Compliance Score:
Compliance Score: 0/10

The project does not implement any consent management mechanisms, which is a critical requirement under GDPR. Implementing the suggested improvements will significantly enhance

```

FIGURE 4.16 – Détection de l’absence complète d’implémentation

Cette évaluation par projet permet une analyse granulaire et ciblée de la conformité RGPD. En se concentrant sur des projets spécifiques, le système offre plusieurs avantages :

- Précision contextuelle : L’analyse se base sur le code réellement implémenté dans le projet concerné.
- Identification des lacunes : Le système détecte automatiquement les pratiques manquantes par rapport aux exigences RGPD.
- Recommandations : Les propositions d’amélioration sont concrètes et adaptées au contexte du projet.
- Évaluation graduée : Le système distingue entre implémentation partielle, complète ou absente.

Cela constitue un outil précieux pour les équipes de développement souhaitant auditer et améliorer la conformité RGPD de leurs projets de manière systématique et automatisée.

### 4.4.3 Interface du système

Nous avons mis en place une interface utilisateur permettant à l’utilisateur d’interagir avec le système. La figure 4.17 présente cette interface.

**GDPR Privacy Assistant**

Enter a privacy-related question (e.g. consent, cookies...) and get implementation practices from real code.

query

what practices are used for consent and cookie management?

Clear
Submit

output

```
Technical Practice Guide for GDPR-Compliant Consent and Cookie Management

GDPR Principle: Consent

What should be done according to the principle:
Implement user consent collection, storage, verification, and revocation before any data processing. Consent must be freely given, informed, specific, unambiguous, and revocable.

Practice Label: Consent Check and Request

Chunk source:
File: library/src/main/java/com/michaelffisar/gdprdialog/GDPR.java
Method: checkIfNeedsToBeShown

Purpose:
This practice ensures that the system checks if user consent is required before processing any data. It uses a callback mechanism to either request consent or update the consent information based on the current state.

GDPR Reference:
Article 7 (Conditions for consent)
```

Flag

FIGURE 4.17 – Interface utilisateur

Il s'agit d'une interface simple qui permet à l'utilisateur de saisir sa requête et de consulter les résultats générés en sortie.

## 4.5 Conclusion

Dans ce chapitre, nous avons validé la fiabilité des résultats issus de la classification automatique. Nous avons également évalué la capacité de notre système RAG à récupérer les segments les plus pertinents, similaires à une requête donnée, et à générer des résultats cohérents. Par la suite, nous avons procédé à des tests pratiques afin d'analyser les solutions mises en œuvre par les développeurs pour se conformer aux principes du RGPD. Les résultats ont mis en évidence que différentes pratiques peuvent être adoptées pour un même concept. Par exemple, la gestion du consentement varie selon qu'il s'agisse d'un projet Web ou Android, chaque environnement technique impliquant des méthodes spécifiques. Nous avons également évalué la conformité à l'échelle des projets, et il a été constaté que ceux-ci abordent généralement certains aspects des principes du RGPD, mais peuvent en négliger d'autres. Cette observation met en évidence la nécessité d'améliorations pour combler ces lacunes. L'évaluation proposée permet ainsi aux développeurs de mesurer la conformité de leurs projets et d'identifier les éléments manquants. En conclusion, notre système représente un premier pas vers un outil d'aide à la conformité, capable de guider les développeurs à travers des exigences fonctionnelles tout en respectant le cadre réglementaire du RGPD.

---

# Conclusion et perspectives

---

La question de la protection des données personnelles occupe aujourd’hui une place centrale dans le développement logiciel, notamment avec l’entrée en vigueur de réglementations strictes telles que le Règlement Général sur la Protection des Données (RGPD). Dans ce contexte, les développeurs sont amenés à réviser et améliorer leurs pratiques de développement pour assurer la conformité avec les exigences en matière de protection de la vie privée.

Ce mémoire s’inscrit dans cette problématique en étudiant concrètement comment les exigences du RGPD sont prises en compte dans le code des projets open-source. Nous avons commencé par explorer les concepts clés, le cadre légal, ainsi que les enjeux liés à la conformité. Cela nous a permis de mieux cerner l’environnement dans lequel évoluent les développeurs.

Par la suite, nous avons abordé des technologies avancées telles que les LLMs, l’analyse syntaxique abstraite (AST) et le Retrieval-Augmented Generation (RAG), qui ont constitué des éléments essentiels dans notre approche. Une revue des travaux existants nous a également permis de situer notre contribution dans la recherche actuelle.

Nous avons ensuite conçu une méthodologie complète pour analyser automatiquement des projets open-source. Cette méthodologie repose sur la collecte de données, la classification automatique, la représentation du code source, et enfin, sur notre approche d’analyse du code en vue d’identifier les pratiques liées à la conformité au RGPD.

Lors de la phase d’implémentation, nous avons présenté les différentes bibliothèques, outils et APIs utilisés pour automatiser la collecte, l’analyse et la classification de données issues des commits, des fichiers README et des structures de code (via l’AST), ainsi que les différents choix d’implémentation des modèles et des techniques.

Enfin, la phase de validation a confirmé la pertinence de notre démarche. Bien que certaines améliorations soient encore possibles, notre système a démontré sa capacité à fournir une première évaluation automatisée, pertinente et utile du respect des principes du RGPD dans les projets open-source.

## Perspectives

Ce travail constitue une première étape vers l’automatisation de l’analyse de la conformité des projets logiciels open source au RGPD, en exploitant les capacités des modèles de langage

(LLM) et les mécanismes de récupération d'information (RAG). Toutefois, plusieurs pistes d'amélioration et de prolongement peuvent être envisagées pour enrichir ce système et le rendre plus robuste et exploitable à grande échelle.

- Tout d'abord, l'amélioration des performances du système RAG pourrait passer par l'enrichissement de la base de connaissances utilisée pour la récupération d'information. L'intégration de l'ensemble des projets collectés permettrait d'élargir la couverture des pratiques de conformité détectables. De plus, l'utilisation de modèles plus performants, tant pour la représentation que pour la génération, ainsi que le recours à une base vectorielle externe, plus scalable et optimisée pour la gestion de millions de vecteurs avec une faible latence, contribuerait à renforcer l'efficacité du système.
- Par ailleurs, la validation du système par des experts du domaine est essentielle afin de confirmer la fiabilité des résultats obtenus par rapport au contexte juridique, en particulier le RGPD, d'identifier les éventuelles lacunes et de proposer des améliorations adaptées.
- Les projets collectés, enrichis de leurs métadonnées et des résultats de classification, seront rendus accessibles pour des études et recherches futures, constituant ainsi un référentiel de projets liés à la confidentialité.
- Les pratiques extraites à partir de projets conformes au RGPD pourraient également être réutilisées dans d'autres contextes juridiques, notamment pour se conformer à la loi algérienne 18-07 relative à la protection des données à caractère personnel. En effet, bien que les cadres réglementaires diffèrent, ces deux textes reposent sur des principes communs tels que le consentement, la sécurité, la minimisation des données ou encore les droits des personnes concernées. Les mécanismes techniques identifiés (journalisation, gestion du consentement, suppression des données, etc.) pourraient ainsi être adaptés, moyennant une contextualisation aux spécificités juridiques locales.
- Explorer l'approche d'analyse par commit en proposant une méthode de reformulation des messages de commit tout en préservant leur sens, et étudier les évolutions à travers les modifications du code, en s'appuyant sur des infrastructures robustes.

Ces perspectives ouvrent la voie à de futures améliorations et intégrations, renforçant ainsi l'impact de notre travail dans le domaine de l'analyse automatisée de la conformité logicielle.

---

# Bibliographie

---

- [1] Aberkane, A.-J. (2021). Automated gdpr-compliance in requirements engineering. Consulté le 9-03-2025.
- [2] AI, M. (2025). Mistral ai documentation. Consulté le 15-05-2025.
- [3] Amazon Web Services (2025). What is retrieval-augmented generation? <https://aws.amazon.com/what-is/retrieval-augmented-generation/>. Consulté le 12-05-2025.
- [4] Anderson, R. (2020). *Security Engineering : Third Edition*. Wiley, Hoboken, NJ, 3rd edition. Consulté le 20-02-2025.
- [5] autorag (2024). How to make good rag evaluation dataset ? Consulté le 26-05-2025.
- [6] Autorité Nationale de Protection des Données à caractère Personnel (2025). Site officiel de l'anpdp. Consulté le 16-02-2025.
- [7] California State Legislature (2018). California civil code § 1798.140 – definitions. [https://leginfo.legislature.ca.gov/faces/codes\\_displaySection.xhtml?lawCode=CIV&sectionNum=1798.140](https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?lawCode=CIV&sectionNum=1798.140). Consulté le 16-02-2025.
- [8] Chang, Y., Wang, X., Wu, Y., Wang, J., Yang, L., Wang, C., Zhu, K., Chen, H., Yi, X., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., and Xie, X. (2024). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3) :1–45. Consulté le 22-03-2025.
- [9] Commission Nationale de l'Informatique et des Libertés (CNIL) (2024). The cnil in a nutshell. Consulté le 26-05-2025.
- [10] DataStax (2025). What is a vector index ? an introduction to vector indexing. <https://www.datastax.com/guides/what-is-a-vector-index>. Consulté le 24-05-2025.

- [11] de Martino, V., Castaño, J., Palomba, F., Franch, X., and Martínez-Fernández, S. (2024). A framework for using llms for repository mining studies in empirical software engineering. Consulté le 12-05-2025.
- [12] Degeling, M., Utz, C., Lentzsch, C., Hosseini, H., Schaub, F., and Holz, T. (2019). We value your privacy ... now take some cookies : Measuring the gdpr's impact on web privacy. Consulté le 12-03-2025.
- [13] Developers, N. (2025). Numpy. Consulté le 15-05-2025.
- [14] Douze, M. (2024). Faiss. Consulté le 20-05-2025.
- [15] et al., J. R. (2020). pandas. Consulté le 20-05-2025.
- [16] European Parliament and Council of the European Union (2016). Regulation (eu) 2016/679 of the european parliament and of the council. <https://eur-lex.europa.eu/eli/reg/2016/679/oj>. Consulté le 15-02-2025.
- [17] Foundation, P. S. (2025). *Python : A dynamic, open source programming language*. Consulté le 26-05-2025.
- [18] Franke, L., Liang, H., Brantly, A., Davis, J. C., and Brown, C. (2024a). A first look at the general data protection regulation (gdpr) in open-source software. Consulté le 9-03-2025.
- [19] Franke, L., Liang, H., Farzanehpour, S., Brantly, A., Davis, J. C., and Brown, C. (2024b). An exploratory mixed-methods study on general data protection regulation (gdpr) compliance in open-source software. In *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '24)*, pages 325–336, New York, NY, USA. ACM. Consulté le 21-02-2025.
- [20] Franke, L. J. (2023). Understanding the impact of data privacy regulations on software and its stakeholders. Technical report. Consulté le 8-03-2025.
- [21] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. (2024). Retrieval-augmented generation for large language models : A survey. Consulté le 23-05-2025.
- [22] GitHub, Inc. (2022). Github rest api documentation. Consulté le 15-05-2025.
- [23] GitHub, Inc. (2025). Github. Consulté le 25-02-2025.
- [24] GitLab Inc. (2025). Gitlab. Consulté le 27-02-2025.
- [25] Google (2025a). Google colaboratory. <https://colab.research.google.com>. Consulté le 12-03-2025.

- [26] Google (2025b). Google drive. Consulté le 20-05-2025.
- [27] Henning, A., Schulte, L., Herbold, S., Kulyk, O., and Mayer, P. (2023). Understanding issues related to personal data and data protection in open source projects on github. Consulté le 8-03-2025.
- [28] IBM (2023). Qu'est-ce que la conformité à la ccpa ? Consulté le 17-02-2025.
- [29] Jacques, V. (2025). Pygithub documentation. Consulté le 20-05-2025.
- [30] Kapitsaki, G. and Papoutsoglou, M. (2024). Gdpr indications in commits messages in github repositories. In *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering : Companion Proceedings (ICSE-Companion '24)*, pages 350–351. ACM. Consulté le 13-03-2025.
- [31] Kutumbe, K. (2024). Mastering chunking for better semantic retrieval with large language models. <https://kshitijkutumbe.medium.com/mastering-chunking-for-better-semantic-retrieval-with-large-language-models-6a6f1fa5f220>. Consulté le 26-05-2025.
- [32] Miranda, J. W. P., Bruneliere, H., Tisi, M., and Sunyé, G. (2024). Towards an in-context llm-based approach for automating the definition of model views. page 14. Consulté le 12-03-2025.
- [33] MyScale (2024). Understanding vector indexing : A comprehensive guide. <https://medium.com/@myscale/understanding-vector-indexing-a-comprehensive-guide-d1abe36ccd3c>. Consulté le 24-05-2025.
- [34] Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., and Mian, A. (2024). A comprehensive overview of large language models. Consulté le 3-05-2025.
- [35] Nikolay, H. and Elena, S. (2021). The expressive power of the statically typed concrete syntax trees. Consulté le 22-03-2025.
- [36] OpenAI (2024). Gpt-4o system card. Consulté le 08-05-2025.
- [37] OpenAI (2025). Openai embeddings guide. Consulté le 24-05-2025.
- [38] OpenRouter (2025). Meta : Llama 4 maverick. Consulté le 08-05-2025.
- [39] Pullakandam, K. (2024). Understanding precision, recall, and fscore at k in recommender systems. *Medium*. Consulté le 13-05-2025.
- [40] Reimers, N. and Gurevych, I. (2019a). Sentence-bert : Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv :1908.10084*. Consulté le 26-05-2025.

- [41] Reimers, N. and Gurevych, I. (2019b). Sentence transformers : Multilingual sentence embeddings using bert. <https://huggingface.co/docs/hub/sentence-transformers>. Consulté le 19-05-2025.
- [42] Saeed, T. (2024). Chunking and embedding strategies in rag : A guide to optimizing retrieval-augmented generation. Consulté le 26-05-2025.
- [43] Sangaroonsilp, P., Dam, H. K., Choetkiertikul, M., Ragkhitwetsagul, C., and Ghose, A. (2021). A taxonomy for mining and classifying privacy requirements in issue reports. Consulté le 21-03-2025.
- [44] Shamon, J. (2024). Mastering code chunking for retrieval augmented generation. Consulté le 26-05-2025.
- [45] Spirin, E., Bogomolov, E., Kovalenko, V., and Bryksin, T. (2021). Psiminer : A tool for mining rich abstract syntax trees from code. Consulté le 28-03-2025.
- [46] Tahaei, M., Li, T., and Vaniea, K. (2022). Understanding privacy-related advice on stack overflow. *Proceedings on Privacy Enhancing Technologies*, 2022(2) :114–131. Consulté le 18-03-2025.
- [47] Team, M. A. (2024). Mistral large : Our flagship model with top-tier reasoning capacities. Consulté le 22-05-2025.
- [48] Trautmann, D. (2023). Large language model prompt chaining for long legal document classification. Consulté le 8-05-2025.
- [49] Youvan, D. C. (2025). (pdf) retrieval-augmented generation (rag) : Advancing ai with dynamic knowledge integration. Technical report. Consulté le 8-04-2025.
- [50] Zhang, J., Wang, X., Zhang, H., Sun, H., and Liu, X. (2020). Retrieval-based neural source code summarization. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pages 1385–1397. IEEE. Consulté le 15-04-2025.
- [51] Zimmermann, T., Weibgerber, P., Diehl, S., and Zeller, A. (2025). A retrospective on mining version histories to guide software changes. *IEEE Transactions on Software Engineering*. Consulté le 9-05-2025.