

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A. Mira Béjaïa
Faculté des Sciences Exactes
Département d'informatique



Mémoire de fin de cycle
Pour l'obtention du diplôme de Master
Sous le thème :

**Développement d'une solution ITSM intelligente
conforme à ITIL4 avec intégration d'une intelligence
artificielle**

Réalisé PAR :

- Ryma Adrar (GL)
- Amel Abbas (ASR)

Sous l'encadrement de :

- Mme Bouadem Nassima

Soutenu le : 30/06/2025

Devant le jury composé de :

- Présidente Mme Zidani Ferroudja
- Examineur M Touazi Djoudi
- Examineur M Khenous lachemi
- Examinatrice Mme Hocini Kenza

Année Universitaire : 2024/2025

Remerciement

Ce travail n'aurait pu voir le jour sans l'aide de Dieu, et sans le soutien de personnes qui ont su, chacune à leur manière, y laisser une empreinte précieuse.

*Je remercie chaleureusement Madame **Nassima BOUADEM**, notre encadrante universitaire, pour son accompagnement bienveillant et structurant tout au long de ce mémoire. Son implication a posé les fondements académiques solides de notre démarche.*

*Ma plus profonde gratitude va à Monsieur **Yassine SAMAH**, analyste SI chez Cevital, pour son encadrement rigoureux, sa confiance, et la clarté qu'il nous a apportée dès le départ. Travailler à ses côtés dans un cadre professionnel stimulant nous a permis de gagner en autonomie, en compétence et en assurance.*

*Merci à Monsieur **Nabil DJAFRI**, responsable support, pour ses conseils avisés et son souci constant de qualité, qui ont renforcé la valeur technique de notre travail.*

*Un remerciement particulier à Monsieur **Farouk OUNECER**, directeur du système d'information, pour le suivi attentif qu'il a accordé à notre projet, et pour sa reconnaissance qui a renforcé notre confiance en nous.*

*Nos sincères remerciements à Monsieur **Samy KHERBACHI**, pour sa disponibilité, ses conseils et sa bienveillance technique.*

*Nous remercions également l'équipe Cevital pour leur accueil professionnel, avec une mention spéciale à Monsieur **Brahim BENNAI** pour sa bienveillance.*

Enfin, nous exprimons notre reconnaissance à l'ensemble de nos enseignants, qui nous ont formés tout au long de ces années.

Dédicace

Je dédie ce travail de tout cœur à

***Mes parents**, pour leurs sacrifices, leur amour inconditionnel et leur présence constante. Merci d'avoir cru en moi, même lorsque moi je doutais.*

*À mon frère **Hassen** et à ma petite sœur **Imene**, pour leur soutien, leurs encouragements et leurs sourires qui ont illuminé mes journées.*

À mes cousins, à mes amies et à toutes les personnes qui m'ont soutenue de près ou de loin, qui ont cru en moi, et qui ont contribué, d'une manière ou d'une autre, à ce parcours.

*À ma chère binôme **Amel**, mon amie précieuse, merci d'être présente dans ma vie, pour ta bienveillance, ton soutien et les moments partagés. Ce chemin n'aurait pas été le même sans toi.*

*Et enfin, à **ma grand-mère**, qui n'est plus parmi nous aujourd'hui, mais dont l'amour et la tendresse continuent de m'accompagner chaque jour. Ce travail, je le lui dédie aussi, avec toute ma pensée et ma gratitude. Et à **ma grand-mère et mon grand-père**, que Dieu les garde en bonne santé et nous accorde encore de longues années à leurs côtés.*

Merci pour tout.

Ryma

Dédicace

Je dédie ce travail de tout cœur à

*À mes parents, Merci pour vos sacrifices, votre amour et votre soutien constant. **Papa**, merci pour ta présence et tes efforts silencieux. **Maman**, merci pour ton énergie, ton cœur, et ta force partagée. C'est grâce à vous que je suis là aujourd'hui.*

*À mes frères et sœurs : **Imene, Meriem, Lamia, Hana et Zakaria**, Vous êtes ma force, mon soutien et ma motivation au quotidien.*

À mon grand-père, tu n'es plus parmi nous, mais tu vis en moi. Je t'avais promis de réussir, et aujourd'hui je t'offre le début de cette promesse. Tu es et tu resteras toujours dans mon cœur.

*À ma binôme **Ryma** , mon amie, ma sœur de cœur, Merci d'avoir partagé ce parcours avec moi. Ensemble, nous avons formé bien plus qu'une équipe.*

*À mes amies **Dalia et Kenza**, pour votre présence, votre amitié précieuse, et votre soutien sincère.*

À mes grands-parents, Que Dieu vous garde. Votre tendresse est un repère.

*Et enfin, à **moi-même**. Pour être restée fidèle à mes objectifs depuis le premier jour. Je me remercie pour l'effort, la patience et la force d'avoir traversé chaque étape avec courage Je me promets de toujours avancer et briller pour moi.*

Amel

Plan

Introduction générale	1
1 Fondements de la Gestion des Services IT et Introduction à ITIL4	3
1.1 Introduction	3
1.2 Les Services Informatiques et leur Importance	3
1.2.1 Définition des Services Informatiques	3
1.2.2 Classification des Services Informatiques	3
1.2.3 Importance Stratégique des Services Informatiques	4
1.3 Cadres et Référentiels de Bonnes Pratiques en IT	4
1.4 Présentation d'ITIL4	5
1.4.1 Définition d'ITIL4	5
1.4.2 Objectifs et Valeur Ajoutée d'ITIL4	5
1.4.3 Composants Essentiels d'ITIL4	6
1.4.4 Gestion des Services avec ITIL4	10
1.4.5 Concepts Opérationnels Clés dans ITIL4	12
1.5 L'intelligence artificielle dans la gestion des services informatiques (ITSM)	12
1.5.1 Définition d'un chatbot	12
1.5.2 Intégration d'un chatbot dans ITIL4	13
1.6 conclusion	13
2 Conception de notre solution ITSM	16
2.1 Introduction	16
2.2 Contexte de la gestion des demandes et incidents	16
2.2.1 La gestion actuelle	16
2.2.2 La gestion automatisée	17
2.3 Identification et définition des acteurs	18
2.3.1 Définition d'un acteur	18
2.3.2 Liste et rôle des acteurs identifiés	19
2.4 Spécifications des besoins	19
2.4.1 Besoins fonctionnels	19
2.4.2 Besoins non fonctionnels	20
2.5 Cas d'utilisation	21
2.5.1 Définition	21
2.5.2 Diagramme de cas d'utilisation	21
2.5.3 Description textuelle des cas d'utilisation	23
2.6 Diagramme de Séquences	24
2.6.1 Se connecter	24
2.6.2 Générer un ticket	25
2.6.3 Gérer une solution proposée par le technicien	26
2.6.4 Gérer les utilisateurs	26

2.6.5	Traiter un ticket	28
2.7	Diagramme d'activité	29
2.7.1	Création d'un ticket	29
2.7.2	Gérer une solution proposée par un technicien	29
2.8	Diagramme de classe	30
2.9	Modèle Logique de données (MLD)	30
2.10	Conclusion	33
3	Mise en œuvre technique et environnement de développement	36
3.1	Introduction	36
3.2	Environnement et outils mobilisés	36
3.2.1	Technologies du front-end	36
3.2.2	Technologies du back-end	37
3.2.3	Technologies de gestion et de stockage des données	37
3.2.4	Outils de modélisation, documentation et design	37
3.2.5	Technologies de réseau et de communication sécurisée	37
3.3	Automatisation et intelligence du système	38
3.3.1	n8n, plateforme d'automatisation	38
3.3.2	Connexion entre l'interface et la plateforme d'automatisation	40
3.3.3	Scénarios d'automatisation	42
3.4	Systèmes intelligents de gestion proactive	45
3.4.1	Système de notifications dynamiques	45
3.4.2	Système d'alerte SLA en temps réel	45
3.4.3	Système d'alerte en cas de défaillance réseau	46
3.5	Sécurisation du système et des échanges	46
3.5.1	Sécurité de l'interface utilisateur	46
3.5.2	Sécurisation des appels API et des webhooks	47
3.5.3	Sécurisation des Webhooks n8n	47
3.5.4	Sécurisation des données échangées et stockées	47
3.6	Mise en place d'un réseau privé virtuel pour la communication sécurisée	48
3.6.1	Contexte et justification	48
3.6.2	Objectifs du VPN dans l'architecture	48
3.6.3	Fonctionnement technique	48
3.6.4	Configuration réalisée	49
3.7	Conclusion	50
4	Présentation visuelle de la solution réalisée et résultats obtenus	53
4.1	Introduction	53
4.2	Identité visuelle de la solution	53
4.3	Création de valeur ITSM à travers la solution "IntelFlow"	54
4.4	Présentation fonctionnelle des interfaces	55
4.4.1	Interfaces personnalisées selon le rôle	58
4.4.2	Base de connaissances intégrée et FAQ	63
4.4.3	Statistiques de performance	64
4.5	Conclusion	66
	Conclusion générale	67

Liste des abréviations

IA	Intelligence Artificielle
ITSM	Information Technology Service Management
IT	Information Technology
ITIL	Information Technology Infrastructure Library
VPN	Virtual Private Network
API	Application Programming Interface
FAQ	Frequently Asked Questions
MLD	Modèle Logique de Données
SLA	Service Level Agreement
RAG	Retrieval-Augmented Generation

Table des figures

1.1	Service Value Chain (SVC)	8
1.2	Système de Valeur des Services (SVS)	9
1.3	Les pratiques d'ITIL4	11
2.1	Schéma global du processus automatisé	18
2.2	Besoins fonctionnels	20
2.3	Diagramme de cas d'utilisation	22
2.4	Diagramme de séquence «Se connecter»	24
2.5	Diagramme de séquence «Générer un ticket»	25
2.6	Diagramme de séquence «Gérer une solution proposée»	26
2.7	Diagramme de séquence «Gérer les utilisateurs»	27
2.8	Diagramme de séquence «Traiter un ticket»	28
2.9	diagramme d'activité "créer un ticket "	29
2.10	Diagramme d'activité "Gérer une solution proposée par un technicien "	29
2.11	Diagramme de classe	30
2.12	Recherche de voisins	32
2.13	Projection de la base	32
3.1	Outils	38
3.2	Flux de traitement des documents vers Qdrant via n8n	43
3.3	Résumé du processus d'automatisation de traitement et d'attribution des tickets	44
3.4	Supervision automatiser	46
3.5	Architecture sécurisée d'accès à la plateforme IntelFlow via VPN	49
3.6	Création et configuration du réseau ZeroTier	49
3.7	Membres connectés et autorisés dans le réseau	50
4.1	LOGO	53
4.2	Interface de connexion	55
4.3	Saisie de l'adresse mail	55
4.4	Vérification du code	56
4.5	Définition d'un nouveau mot de passe	56
4.6	Changement de rôle	57
4.7	Interface d'accueil de demandeur	58
4.8	Interface d'accueil du technicien	58
4.9	La liste des tickets	59
4.10	Ajout d'un utilisateur	59
4.11	Interface du catalogue de services	60
4.12	la gestion de catalogue de services	60
4.13	Calendrier de disponibilité	61
4.14	Interface de disponibilité des techniciens	61
4.15	Interface des équipements côté superviseur	62
4.16	Interface des équipements côté techniciens	62

4.17	Base de connaissance côté demandeur	63
4.18	Foire aux questions	63
4.19	Base de connaissance	64
4.20	Statistiques	64
4.21	Evolution des tickets	65
4.22	Répartition des tickets par type et par mode de résolution	65
4.23	Statistique côté superviseur	66

Introduction générale

Dans un monde où les systèmes d'information sont au cœur des activités des entreprises, la qualité de la gestion des services IT représente un enjeu stratégique majeur. Cependant, de nombreuses organisations rencontrent encore des difficultés à assurer une gestion fluide et efficace, en raison de processus lourds, d'un manque d'automatisation et d'une surcharge des équipes support. Ces difficultés entraînent des retards dans le traitement des demandes, une réactivité insuffisante face aux incidents, ainsi qu'une insatisfaction croissante des utilisateurs.

Face à ces enjeux, une question centrale se pose : comment améliorer la gestion des services IT afin d'optimiser à la fois l'efficacité opérationnelle et l'expérience utilisateur ?

Le cadre méthodologique ITIL4 (Information Technology Infrastructure Library) constitue une référence majeure dans la gestion des services informatiques. En alignant les processus IT sur les besoins métiers et en promouvant des principes d'amélioration continue, ITIL4 offre un cadre structurant pour moderniser la gestion des services. Toutefois, sa mise en œuvre peut s'avérer complexe, notamment pour les utilisateurs non experts.

C'est dans ce contexte que s'inscrit notre projet. Nous avons conçu et développé une solution ITSM conforme aux principes d'ITIL4, centrée sur la gestion des demandes et des incidents, et intégrant un assistant virtuel (chatbot). Ce dernier permet de créer automatiquement les tickets, de proposer des solutions adaptées, et d'attribuer les demandes au technicien le plus approprié, tout en allégeant la charge de travail des équipes techniques.

L'objectif de ce mémoire est de présenter cette solution innovante, en abordant les fondamentaux de l'ITSM et d'ITIL4, les limites des outils classiques, puis la conception et le développement de notre solution. Nous mettrons en lumière les bénéfices apportés en termes d'ergonomie, d'automatisation intelligente et d'amélioration de la qualité du service.

Afin de répondre à cette problématique, nous avons structuré notre mémoire en quatre chapitres :

- **Chapitre 1** : Fondements de la Gestion des Services IT et introduction à ITIL4
- **Chapitre 2** : Conception de la solution ITSM
- **Chapitre 3** : Mise en œuvre technique et environnement de développement
- **Chapitre 4** : Présentation visuelle de la solution réalisée et résultats obtenus

CHAPITRE 1

*Fondements de la Gestion des Services IT
et Introduction à **ITIL4***

Chapitre 1

1.1 Introduction

Dans un monde où la transformation numérique est devenue un levier essentiel de compétitivité, les services informatiques jouent un rôle central dans le fonctionnement des organisations. Qu'il s'agisse de la gestion des infrastructures, du support aux utilisateurs ou du développement d'applications métier, l'IT est désormais un pilier fondamental de la performance et de l'innovation des entreprises.

Toutefois, la complexité croissante des systèmes d'information et la nécessité d'une gestion efficace des services IT ont conduit à l'adoption de cadres méthodologiques et de bonnes pratiques visant à structurer, optimiser et aligner ces services sur les besoins stratégiques des entreprises. Parmi ces cadres, ITIL (Information Technology Infrastructure Library) s'est imposé comme une référence incontournable pour la gestion des services IT.

Dans ce qui suit, nous allons explorer les concepts fondamentaux des services informatiques, mettre en lumière leur importance stratégique, et analyser les principales approches permettant d'assurer leur efficacité. Nous nous pencherons ensuite sur le cadre ITIL4, ses objectifs, ainsi que ses composants essentiels, notamment le Système de Valeur des Services (SVS) et les quatre dimensions de la gestion des services IT, qui constituent les bases de cette approche modernisée.

1.2 Les Services Informatiques et leur Importance

1.2.1 Définition des Services Informatiques

Les services informatiques englobent l'ensemble des activités et des solutions technologiques mises en œuvre pour soutenir et optimiser les processus métier d'une organisation. Ils incluent la gestion des systèmes d'information, le développement et la maintenance des applications, le support technique aux utilisateurs, ainsi que la sécurisation des données et des réseaux. L'objectif principal est d'assurer le bon fonctionnement des opérations quotidiennes tout en facilitant l'innovation et la croissance de l'entreprise. [2]

1.2.2 Classification des Services Informatiques

Les services informatiques se déclinent en plusieurs catégories principales :

a. Services d'Infrastructure IT :

Ces services concernent la gestion et la maintenance des composants matériels et logiciels fondamentaux qui supportent l'ensemble des opérations informatiques de l'organisation. Cela inclut les serveurs, les réseaux, les dispositifs de stockage et les plateformes cloud.

Une infrastructure bien gérée assure une base solide pour toutes les applications et services métiers.

b. **Services Applicatifs :**

Cette catégorie englobe le développement, le déploiement et la maintenance des applications logicielles répondant aux besoins spécifiques de l'entreprise. Cela peut inclure des systèmes de gestion de la relation client (CRM), des plateformes de commerce électronique ou des applications mobiles dédiées. Ces services garantissent que les outils logiciels sont alignés sur les objectifs stratégiques et opérationnels de l'organisation.

c. **Services de Support Technique :**

Le support technique fournit une assistance aux utilisateurs finaux pour résoudre les problèmes liés aux équipements informatiques, aux logiciels ou aux réseaux. Cela inclut le dépannage, la résolution d'incidents, la gestion des demandes de service et la formation des utilisateurs. Un support efficace améliore la satisfaction des employés et assure une continuité opérationnelle.

d. **Services de Sécurité Informatique :**

Ces services visent à protéger les actifs informationnels de l'organisation contre les menaces internes et externes. Ils comprennent la mise en place de pare-feu, la gestion des antivirus, le chiffrement des données, la surveillance des réseaux et la formation à la sensibilisation à la sécurité. Une approche proactive en matière de sécurité est essentielle pour préserver la confidentialité, l'intégrité et la disponibilité des informations.

e. **Services Cloud et de Virtualisation :**

Avec l'évolution vers des solutions plus flexibles et évolutives, les services cloud et de virtualisation permettent aux entreprises d'héberger des applications et des données sur des plateformes externes, réduisant ainsi les coûts liés à l'infrastructure physique. Ces services offrent également une scalabilité adaptée aux besoins changeants de l'entreprise. [11]

1.2.3 Importance Stratégique des Services Informatiques

Les services informatiques jouent un rôle crucial dans la stratégie globale des entreprises modernes :

a. **Amélioration de l'Efficacité et de la Productivité :**

En automatisant les processus métier et en fournissant des outils collaboratifs, les services informatiques permettent aux employés de travailler plus efficacement, réduisant les tâches manuelles et les erreurs, ce qui conduit à une augmentation de la productivité globale.

b. **Support à la Transformation Numérique :**

Les services informatiques sont au cœur de la transformation numérique, aidant les entreprises à adopter de nouvelles technologies, à innover dans leurs offres et à rester compétitives sur le marché. Ils facilitent l'intégration de solutions numériques dans tous les aspects de l'organisation.

c. **Réduction des Coûts et Optimisation des Ressources :**

Une gestion efficace des services informatiques permet de rationaliser les opérations, de réduire les dépenses inutiles et d'optimiser l'utilisation des ressources, notamment grâce à l'adoption de solutions cloud et à l'automatisation des processus. [14]

1.3 Cadres et Référentiels de Bonnes Pratiques en IT

Les cadres et référentiels de bonnes pratiques en IT sont des méthodologies et frameworks développés pour optimiser la gestion des services informatiques, améliorer l'efficacité des proces-

sus et assurer la conformité avec les exigences réglementaires. Ils permettent aux organisations d'aligner leurs stratégies IT sur les objectifs métiers et d'adopter des pratiques éprouvées pour garantir la performance, la sécurité et l'innovation.

Il existe aujourd'hui plusieurs cadres de référence et approches méthodologiques qui visent à structurer la gestion des services informatiques; parmi eux, ITIL se distingue comme l'un des plus utilisés et les plus adaptés aux enjeux actuels des organisations. Sa version actuelle, **ITIL4**, représente une évolution majeure du cadre, introduisant une approche moderne, agile et centrée sur la co-crédation de valeur.

1.4 Présentation d'ITIL4

1.4.1 Définition d'ITIL4

ITIL (Information Technology Infrastructure Library) est un cadre de gestion des services informatiques (ITSM – IT Service Management) conçu pour aider les organisations à aligner leurs services IT avec les besoins métiers. ITIL4 est la dernière version de ce cadre, publiée en 2019 par AXELOS, et introduit une approche plus flexible, intégrant des pratiques modernes comme Agile, DevOps et Lean IT.

ITIL4 repose sur une vision holistique de la gestion des services, mettant l'accent sur la co-crédation de valeur, la collaboration entre les parties prenantes et l'optimisation continue des services.[13]

1.4.2 Objectifs et Valeur Ajoutée d'ITIL4

1.4.2.1 Objectifs d'ITIL4

ITIL4 est conçu pour moderniser la gestion des services IT en intégrant les nouvelles méthodologies agiles, DevOps et Lean IT. Son objectif principal est d'améliorer la qualité et l'efficacité des services IT tout en garantissant une gestion flexible et optimisée.

Les principaux objectifs d'ITIL4 sont :

1. Aligner les services IT avec les objectifs stratégiques de l'entreprise
 - Assurer que les services IT soutiennent directement les objectifs métier et les besoins des utilisateurs.
 - Favoriser une approche axée sur la valeur et la performance de l'entreprise.
 - Permettre une meilleure prise de décision grâce à des processus standardisés.
2. Optimiser la gestion des ressources IT (humaines, matérielles, logicielles)
 - Améliorer l'utilisation et la répartition des ressources pour éviter les gaspillages.
 - Mettre en place des bonnes pratiques pour une gestion efficace du personnel IT, des infrastructures et des logiciels.
 - Automatiser certaines tâches pour réduire la charge de travail et accroître l'efficacité.
3. Garantir la satisfaction des utilisateurs grâce à une meilleure gestion des incidents et des demandes
 - Réduire le temps moyen de résolution des incidents (MTTR - Mean Time to Repair).
 - Assurer un support client réactif et efficace pour minimiser les interruptions de service.

- Offrir des services personnalisés et adaptés aux besoins spécifiques des utilisateurs.
4. Favoriser la flexibilité et l'adaptabilité des services IT face aux évolutions technologiques et aux besoins métiers
 - Permettre une adoption rapide des nouvelles technologies (Cloud, DevOps, AI, etc.).
 - Adapter les services IT aux transformations digitales et aux nouvelles exigences du marché.
 - Encourager une culture de l'innovation et du changement continu.
 5. Réduire les coûts opérationnels tout en maintenant un haut niveau de qualité de service
 - Optimiser les coûts de gestion IT sans compromettre la qualité des services.
 - Éliminer les processus inefficaces grâce à l'automatisation et aux meilleures pratiques.
 - Réduire le nombre d'incidents et de pannes pour limiter les coûts liés aux interruptions de service.

1.4.2.2 Valeur ajoutée d'ITIL4

L'adoption d'ITIL4 apporte des avantages significatifs aux organisations en améliorant la gestion des services IT et en optimisant leur performance globale. Voici les principales valeurs ajoutées :

1. Meilleure efficacité opérationnelle grâce à des processus optimisés
ITIL4 propose des workflows clairs et efficaces pour rationaliser la gestion des services. Les entreprises gagnent en productivité grâce à des processus bien définis et normalisés.
2. Amélioration continue des services IT avec un modèle basé sur les retours d'expérience
Le modèle d'amélioration continue (Continual Improvement Model) encourage l'évolution constante des services. Les décisions sont prises sur la base de données réelles et d'indicateurs de performance.
3. Gestion proactive des risques et incidents pour minimiser les interruptions de service
ITIL4 favorise une approche prédictive et préventive pour identifier et corriger les problèmes avant qu'ils ne deviennent critiques. La gestion des incidents et des problèmes est optimisée pour réduire l'impact des interruptions.
4. Amélioration de la satisfaction des clients en proposant des services plus réactifs et alignés sur leurs besoins
Les utilisateurs bénéficient d'un service plus rapide, fiable et adapté à leurs attentes. La gestion des demandes est simplifiée pour assurer un support client fluide et efficace.
5. Interopérabilité avec Agile, DevOps et Lean IT, permettant une transformation digitale plus fluide
ITIL4 est conçu pour être compatible avec les approches modernes (Agile, DevOps, Lean) afin d'accélérer les cycles de développement et d'innovation. Les entreprises peuvent intégrer ITIL4 dans des environnements hybrides pour assurer une gestion cohérente des services IT et métier.

1.4.3 Composants Essentiels d'ITIL4

ITIL4 repose sur un ensemble de concepts fondamentaux qui permettent aux organisations de gérer efficacement leurs services IT. Ces concepts sont structurés autour du Système de Valeur des Services (SVS) et des quatre dimensions fondamentales qui garantissent une approche équilibrée et holistique de la gestion des services.[13]

1.4.3.1 Le Système de Valeur des Services (SVS)

Le Service Value System (SVS) est le cadre central d'ITIL4 qui englobe l'ensemble des éléments et activités nécessaires pour la création, la gestion et l'optimisation des services IT. Il permet de s'assurer que toutes les parties prenantes (clients, fournisseurs, équipes IT, direction) sont alignées vers un objectif commun : fournir de la valeur aux utilisateurs finaux. [4]

Le SVS est composé de cinq éléments clés :

1. Les Principes Directeurs :

Les principes directeurs d'ITIL4 sont des recommandations universelles conçues pour aider les organisations à adopter et adapter le cadre ITIL en fonction de leurs besoins spécifiques. Ces principes ne sont pas rigides mais servent de guides généraux pour améliorer la gestion des services IT tout en garantissant un alignement avec les objectifs stratégiques de l'organisation.

Ces principes sont inspirés de méthodologies modernes telles que Agile, Lean et DevOps, favorisant une approche flexible, efficace et centrée sur la valeur.

Parmi ces principes :

- Concentrez-vous sur la valeur : chaque service doit apporter un bénéfice mesurable à l'entreprise et aux utilisateurs.
- Commencez là où vous êtes : il est préférable d'améliorer les processus existants plutôt que de les reconstruire entièrement.
- Progressez de manière itérative avec des retours d'expérience : adopter une amélioration progressive et continue.
- Collaborez et favorisez la transparence : une gestion efficace des services IT repose sur une communication et une coopération optimales.
- Penser et travailler de manière holistique : il faut considérer l'organisation comme un tout : processus, personnes, partenaires et technologies.
- Rester simple et pratique : éviter toute complexité inutile et toujours privilégier des solutions claires et efficaces.
- Optimiser et automatiser : automatiser les tâches répétitives pour libérer du temps et améliorer les performances globales.[7]

2. La Gouvernance

La gouvernance définit la manière dont les décisions sont prises, contrôlées et appliquées au sein d'une organisation. Elle garantit que la gestion des services IT est en accord avec les objectifs stratégiques et les exigences réglementaires de l'entreprise.

Les activités de gouvernance incluent :

- Définition des politiques et des normes IT.
- Surveillance et évaluation des performances des services.
- Ajustement des processus pour répondre aux besoins métiers.

3. La Chaîne de Valeur des Services (Service Value Chain - SVC)

La Service Value Chain (SVC) représente l'ensemble des étapes et interactions nécessaires pour créer, fournir et améliorer les services IT. Elle est composée de six activités clés :

Service Value Chain - SVC	
Activité	Description
Planifier	Définir la stratégie et les objectifs des services IT.
Améliorer	Assurer une amélioration continue des services et des processus.
Engager	Gérer les interactions avec les parties prenantes (clients, fournisseurs).
Concevoir et transitionner	Développer de nouveaux services ou modifier les services existants.
Obtenir et construire	Acquérir ou développer les composants nécessaires à la prestation des services.
Fournir et supporter	Assurer la livraison des services et leur maintenance.

FIGURE 1.1 – Service Value Chain (SVC)

Chaque activité de la chaîne de valeur est interconnectée et peut être personnalisée en fonction des besoins spécifiques de l'organisation.

4. Les Pratiques ITIL (ITIL Practices)

ITIL4 propose 34 pratiques réparties en trois catégories :

- Pratiques de gestion générale
- Pratiques de gestion des services
- Pratiques de gestion technique

Ces pratiques permettent d'optimiser la gestion des services IT en s'appuyant sur des méthodes éprouvées et des bonnes pratiques adaptées à chaque situation.

5. L'Amélioration Continue

L'amélioration continue est un principe clé d'ITIL4 qui vise à optimiser en permanence la qualité des services et l'efficacité des processus IT.

Elle repose sur un cycle d'amélioration en sept étapes :

- Déterminer la vision et les objectifs.
- Évaluer l'état actuel des services.
- Définir l'état futur souhaité.
- Déterminer les actions à entreprendre.
- Mettre en œuvre les améliorations.
- Évaluer les résultats.
- Ajuster et itérer pour de nouvelles améliorations. Ce processus garantit que l'organisation s'adapte en permanence aux évolutions technologiques et aux attentes des utilisateurs.

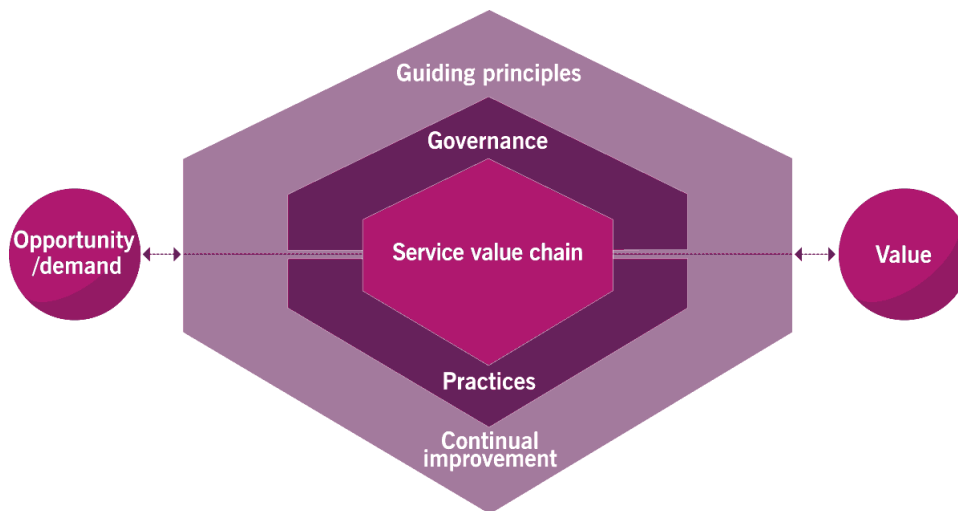


FIGURE 1.2 – Système de Valeur des Services (SVS)

1.4.3.2 Les Quatre Dimensions Fondamentales

ITIL4 définit quatre dimensions essentielles qui doivent être prises en compte pour assurer une gestion efficace et équilibrée des services IT. Ces dimensions permettent d'adopter une vision globale et cohérente pour la conception et l'exploitation des services.[4] [13]

1. Organisation et Personnes :

- Comprend les compétences, la culture d'entreprise, les rôles et responsabilités des équipes IT.
- Favorise une collaboration efficace entre les différents services et partenaires.
- Met l'accent sur le développement des talents et l'amélioration continue des compétences.

2. Information et Technologie :

- Englobe les outils, logiciels, bases de données, et infrastructures utilisés pour gérer les services IT.
- Assure que les systèmes d'information sont alignés avec les objectifs stratégiques de l'organisation.
- Intègre les nouvelles technologies comme le cloud computing, l'IA et l'automatisation.

3. Partenaires et Fournisseurs :

- Gère les relations avec les fournisseurs, prestataires et partenaires stratégiques.
- Assure une gouvernance efficace des contrats, SLA (Service Level Agreements).
- Évalue la dépendance de l'organisation vis-à-vis des fournisseurs IT externes.

4. Flux de Valeur et Processus :

- Décrit la manière dont les services IT sont conçus, fournis et améliorés.
- Permet une gestion fluide des interactions entre les différentes activités IT.
- Optimise les processus pour réduire les délais, améliorer la qualité et minimiser les coûts.

1.4.4 Gestion des Services avec ITIL4

ITIL4 propose un cadre structuré pour la gestion des services IT à travers 34 pratiques de gestion [9], réparties en trois catégories principales :

1. Pratiques de Gestion Générale (General Management Practices)
2. Pratiques de Gestion des Services (Service Management Practices)
3. Pratiques de Gestion Technique (Technical Management Practices)

Ces pratiques permettent aux organisations d'optimiser leurs services IT, de réduire les risques et d'améliorer en continu la qualité des prestations offertes aux utilisateurs.

1.4.4.1 Pratiques de Gestion Générale

Les pratiques de gestion générale sont issues des domaines traditionnels de la gestion d'entreprise et s'appliquent aux services IT comme aux autres activités de l'organisation. Elles permettent de piloter la stratégie IT, gérer les risques, améliorer les processus et garantir la conformité aux standards et réglementations (14pratiques)

Exemples de pratiques clés :

1. Gestion de la Stratégie
2. Gestion des Risques
3. Gestion de la Sécurité de l'Information

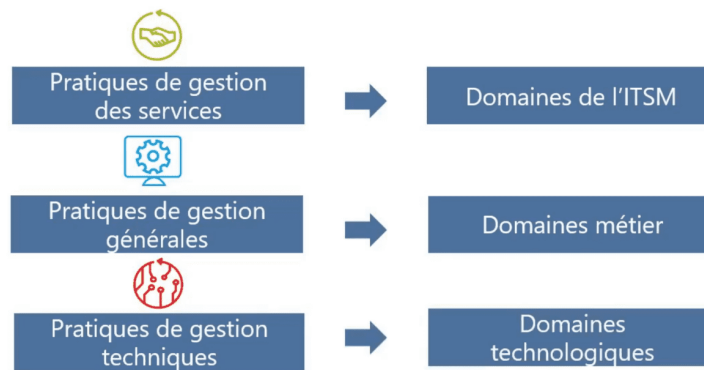


FIGURE 1.3 – Les pratiques d'ITIL4

1.4.4.2 Pratiques de Gestion des Services

Ces pratiques sont directement liées à la gestion des services IT. Elles permettent d'optimiser l'efficacité opérationnelle, la satisfaction des utilisateurs et la continuité des services. (17 pratiques)
Exemples de pratiques clés :

1. Gestion des incidents
2. Gestion des demandes
3. Gestion des problèmes
4. Gestion des changements

1.4.4.3 Pratiques de Gestion Technique

Ces pratiques concernent les aspects techniques de la gestion des services IT et garantissent l'efficacité des infrastructures et des applications. (3 pratiques)

1. Gestion des Infrastructures et des Plateformes
2. Développement et Gestion des Logiciels
3. Gestion du Déploiement

Les 34 pratiques d'ITIL4 permettent aux organisations de structurer et d'optimiser leur gestion des services IT. Elles garantissent une meilleure réactivité, une amélioration continue et une réduction des risques IT.

1.4.5 Concepts Opérationnels Clés dans ITIL4

Dans le cadre d'ITIL4, certains termes sont essentiels pour comprendre la gestion des services informatiques au quotidien :

- **Demande** : il s'agit d'une demande standard provenant d'un utilisateur pour obtenir un service, une information ou un accès spécifique. [3]
- **Incident** : un incident est un événement non planifié qui entraîne une interruption ou une dégradation de la qualité d'un service informatique. L'objectif de la gestion des incidents est de rétablir le service normal aussi rapidement que possible
- **Ticket** : un ticket est un enregistrement dans un outil de gestion des services () qui permet de tracer toute demande, incident, problème ou changement. Il contient les informations nécessaires pour traiter le cas, comme le type, la description, le demandeur , catégorie , SLA
- **SLA (Service Level Agreement)** : Dans ITIL4, un SLA est un accord formel entre le fournisseur de services et le client, précisant le niveau attendu d'un service en termes de disponibilité, de délai de réponse ou de temps de résolution. Il vise à s'assurer que les services fournis sont alignés avec les besoins métier. Le SLA est un outil clé pour mesurer la performance du service

1.5 L'intelligence artificielle dans la gestion des services informatiques (ITSM)

L'**intelligence artificielle (IA)** représente aujourd'hui un atout majeur dans la transformation numérique des entreprises. Dans le domaine de la gestion des services informatiques (ITSM), elle offre de nouvelles perspectives pour repenser les méthodes de travail, renforcer l'efficacité opérationnelle et améliorer l'expérience des utilisateurs.

L'intégration progressive de l'IA dans l'ITSM traduit une volonté des organisations de moderniser leurs outils de gestion et de tirer parti des avancées technologiques pour faire face à la complexité croissante des systèmes d'information. Elle s'inscrit également dans une démarche d'innovation continue, encouragée par les bonnes pratiques comme celles proposées par le référentiel ITIL4.

Ainsi, l'IA n'est pas seulement une technologie de rupture, mais un vecteur d'évolution des processus IT qui permet d'envisager une gestion plus intelligente, proactive et centrée sur les besoins des utilisateurs.[8]

1.5.1 Définition d'un chatbot

un chatbot est un programme informatique qui simule et traite une conversation humaine (écrite ou parlée), permettant aux humains d'interagir avec des terminaux digitaux comme s'ils communiquaient avec une personne réelle. Les chatbots peuvent être aussi simples que des programmes rudimentaires répondant à une requête simple avec une réponse sur une seule ligne, ou aussi sophistiqués que des assistants digitaux qui apprennent et évoluent pour fournir des niveaux de personnalisation croissants à mesure qu'ils collectent et traitent des informations.[14]

1.5.2 Intégration d'un chatbot dans ITIL4

1.5.2.1 Le Rôle d'un Chatbot dans ITIL4

L'intégration d'un chatbot dans ITIL4 permet d'optimiser la gestion des services informatiques en automatisant et en facilitant les interactions entre les utilisateurs et les systèmes IT. ITIL4 repose sur une approche axée sur la valeur et la flexibilité, et un chatbot constitue un outil stratégique pour améliorer l'efficacité des processus ITSM. Son rôle principal est de servir d'interface intelligente entre les utilisateurs et les services IT en répondant aux demandes courantes, en traitant les incidents et en fournissant des informations pertinentes.

1.5.2.2 Les Avantages

- Amélioration de l'expérience utilisateur (support rapide et disponible 24/7).
- Réduction de la charge de travail des équipes IT.
- Standardisation des processus et réduction des erreurs humaines.

1.6 conclusion

Dans ce chapitre, nous avons posé les fondations nécessaires à la suite de notre travail. Nous avons commencé par définir les services informatiques, en expliquant leur rôle essentiel et leur impact stratégique dans les organisations. Ensuite, nous avons présenté le cadre de bonnes pratiques ITIL4, en mettant en avant ses principes clés, ses objectifs, ainsi que ses apports dans la gestion des services.

Nous avons également mis en lumière le rôle de l'intelligence artificielle, notamment à travers les chatbots, dans le cadre d'une démarche ITIL, afin de montrer comment elle peut contribuer à l'automatisation de certaines tâches et à l'amélioration de l'expérience utilisateur.

L'ensemble de ces notions va nous permettre d'aborder, dans les chapitres suivants, l'analyse des besoins, la conception de notre solution, ainsi que son développement et sa mise en œuvre.

CHAPITRE 2

Conception de notre solution ITSM

Chapitre 2

2.1 Introduction

Ce chapitre a pour objectif de poser les bases fonctionnelles de notre système de gestion automatisée des demandes et des incidents, en s'inscrivant dans le cadre des bonnes pratiques ITIL4. Dans un premier temps, nous présentons le contexte général lié à la gestion des demandes et des incidents au sein des systèmes d'information, en mettant en évidence les limites des approches classiques.

Ensuite, nous décrivons le fonctionnement envisagé dans notre solution, qui repose sur l'intégration d'un agent intelligent capable de générer automatiquement des tickets à partir des besoins exprimés par les utilisateurs. L'automatisation du processus est illustrée par un schéma global représentant l'ensemble du cycle de traitement des demandes et des incidents.

Ce chapitre se poursuit par l'analyse des besoins, l'identification des acteurs et de leurs interactions, la modélisation des cas d'utilisation, ainsi que les différents diagrammes UML (séquence, activité, classes), permettant de structurer la conception du système.

2.2 Contexte de la gestion des demandes et incidents

2.2.1 La gestion actuelle

Dans les systèmes traditionnels de gestion des demandes et des incidents, le processus repose essentiellement sur une intervention humaine à chaque étape. Lorsqu'un demandeur rencontre un incident ou exprime un besoin, il est invité à accéder à une interface de gestion pour créer un ticket. Cela suppose de remplir un formulaire, avec des champs obligatoires tels que le titre, la catégorie, l'urgence, l'impact, la description, etc. Une fois le formulaire complété, le ticket est transmis dans le système de manière non automatisée à l'équipe de support, sans ciblage direct d'un technicien.

Cependant, cette approche montre rapidement ses limites. D'une part, tous les demandeurs ne sont pas à l'aise avec la procédure, certains rencontrent des difficultés à remplir les formulaires correctement, d'autres abandonnent avant même de terminer leur saisie. Il arrive également que des utilisateurs ne sachent pas comment initier la création d'un ticket, ce qui les pousse à contourner le système en contactant directement les techniciens par email ou téléphone.

Cela crée une multiplication des canaux de contact, souvent non centralisés, non tracés et non standardisés. Résultat : les techniciens doivent eux-mêmes reprendre les informations reçues par email ou oralement, puis enregistrer manuellement les tickets dans le système afin de garantir une traçabilité minimale. Cela engendre une surcharge de travail importante pour les techniciens, qui se retrouvent à gérer à la fois des tickets informels et leur

traitement technique.

De plus, ce processus est source d'erreurs, de duplications ou d'oublis. Le manque de rigueur dans l'enregistrement, la non-priorisation des tickets, l'absence d'analyse centralisée et la lenteur globale du traitement nuisent fortement à la qualité du service fourni.

2.2.1.1 Principales limites de la gestion actuelle :

- Difficulté ou incapacité à créer un ticket.
- Utilisation de canaux non officiels (email, téléphone).
- Charge de travail accrue pour les techniciens.
- Erreurs de saisie ou oublis.
- Manque de suivi structuré.
- Absence de traçabilité complète.
- Faible réactivité du service.

Face à ces constats, il devient essentiel de repenser le modèle en intégrant une gestion automatisée, plus fluide, plus intelligente et centrée sur l'utilisateur.

2.2.2 La gestion automatisée

La gestion automatisée des demandes et des incidents représente une avancée majeure dans la modernisation des systèmes de support. Elle repose sur l'intégration d'agents intelligents, de bases de connaissances et de workflows automatisés afin de fluidifier l'ensemble du processus, depuis la saisie du besoin jusqu'à sa résolution.

2.2.2.1 Fonctionnement général :

- L'utilisateur exprime son besoin à travers une interface conversationnelle (chatbot).
- Un agent IA analyse le message et déduit automatiquement les données clés : type, titre, description, urgence, impact, etc en s'appuyant sur un référentiel structuré de services.
- Le système affiche automatiquement le ticket généré.
- L'utilisateur peut valider ou annuler le ticket généré.
- Une solution adaptée est automatiquement suggérée à l'utilisateur, si elle est identifiée dans la base de connaissances par l'agent IA.
- L'utilisateur peut valider ou refuser cette solution.
- En cas de refus, le ticket est transmis automatiquement à un technicien en fonction de sa disponibilité, de sa charge actuelle et de son domaine de compétence.

Ce système automatise l'ensemble du processus de génération de ticket de l'analyse du besoin à l'acheminement vers le support et libère ainsi les techniciens des tâches répétitives, leur permettant de se concentrer sur les incidents nécessitant une réelle expertise.

2.2.2.2 Avantages de l'automatisation :

- Réduction des délais de création et de traitement.

- Expérience utilisateur simplifiée et guidée.
- Moins d’erreurs liées à la saisie manuelle.
- Statistiques et tableaux de bord automatisés.
- Transfert intelligent vers les techniciens en cas de besoin.

L’automatisation permet donc d’uniformiser le processus, de mieux gérer les tickets et d’offrir une réponse plus rapide et plus pertinente à l’utilisateur.

2.2.2.3 Schéma du processus de gestion automatisée des demandes

Cette image représente le processus de gestion des demandes dans une plateforme ITSM intégrant une intelligence artificielle. Le demandeur commence par définir son besoin, qui est ensuite analysé automatiquement par un agent IA. Un ticket est alors généré et peut être validé ou annulé par le demandeur. Une fois validé, l’IA cherche une solution similaire dans une base vectorielle, ici représentée par Qdrant. Si une solution existe, elle est proposée au demandeur, qui peut l’approuver pour clôturer le ticket, ou la refuser. Si aucune solution n’est trouvée, le ticket est attribué à un technicien pour traitement. Ce processus permet d’optimiser la résolution des demandes tout en réduisant l’intervention humaine aux cas les plus complexes.

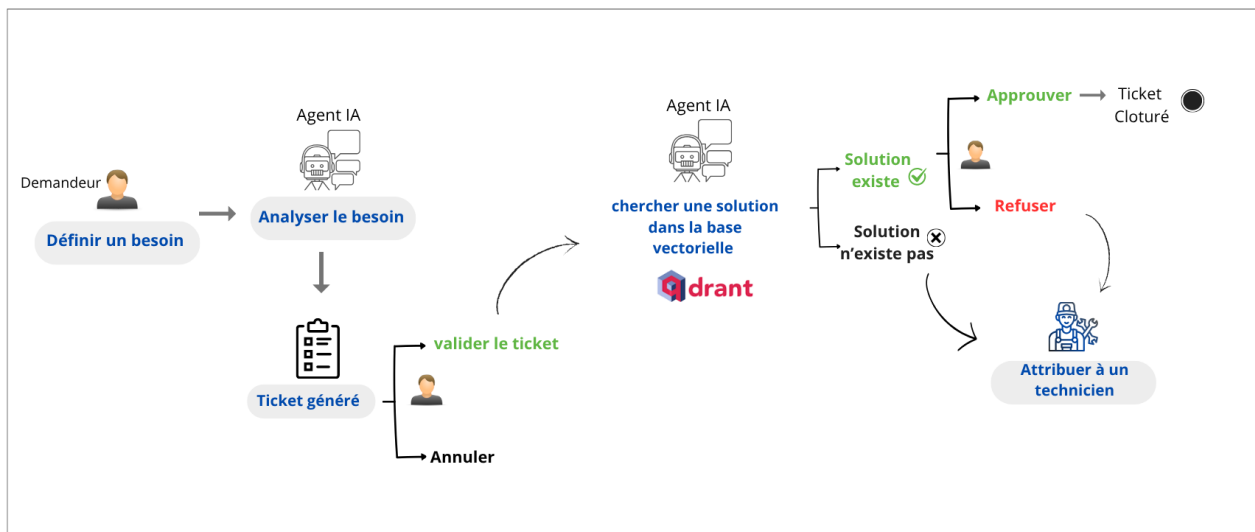


FIGURE 2.1 – Schéma global du processus automatisé

2.3 Identification et définition des acteurs

2.3.1 Définition d’un acteur

En modélisation UML, **un acteur** est défini comme toute entité (humaine ou logicielle) qui interagit avec le système pour accomplir une tâche. Un acteur peut initier une action (acteur primaire) ou simplement recevoir une réponse (acteur secondaire)[1]

2.3.2 Liste et rôle des acteurs identifiés

Les acteurs du système se répartissent entre un agent IA, chargé de l'automatisation, et des utilisateurs occupant différents rôles : demandeur, technicien, administrateur, superviseur chacun ayant des droits et fonctions spécifiques.

- **Demandeur** : Utilisateur final qui interagit avec l'interface pour exprimer un besoin.
- **Technicien** : Acteur chargé du traitement opérationnel des tickets et du suivi technique.
- **Administrateur** : Responsable de la configuration générale et de l'administration du système.
- **Superviseur** : Utilisateur ayant accès aux outils de suivi, de contrôle et à certaines fonctions de gestion.
- **Agent IA (intelligence artificielle)** : Composant autonome chargé d'analyser les besoins, de superviser l'état des équipements, et de générer des alertes automatisées si nécessaire.

2.4 Spécifications des besoins

On distingue deux types de besoins : les besoins fonctionnels et les besoins non fonctionnels

2.4.1 Besoins fonctionnels

Il s'agit des fonctionnalités que doit proposer le système. Elles correspondent aux besoins exprimés en termes de comportements d'entrée et de sortie. D'un point de vue fonctionnel, notre application devra intégrer l'ensemble des fonctionnalités nécessaires pour répondre aux besoins de l'utilisateur

Les principales fonctionnalités du système sont présentées dans le tableau ci-dessous.

Acteur	Besoins fonctionnels
Administrateurs	<ul style="list-style-type: none"> -Gérer les utilisateurs. -Consulter l'ensemble des tickets créés. -Consulter les statistiques. -Gérer le catalogue de services. -Gérer la base de connaissances . -Gérer la FAQ.
Technicien	<ul style="list-style-type: none"> -Traiter ses tickets. -Consulter les statistiques. -Gérer la base de connaissances. -Gérer ses disponibilités . -Superviser les équipements connectés .
Superviseur	<ul style="list-style-type: none"> -Consulter l'ensemble des tickets créés. -Consulter les statistiques. -Gérer la base de connaissances . -Gérer la FAQ. -Gérer le catalogue de services. -Gérer les équipements connectés.
Demandeur	<ul style="list-style-type: none"> -Définir un besoin. -Valider ou annuler un ticket. -Gérer une solution proposée. -Consulter les statistique. -Consulter la base de connaissances . -Consulter la FAQ .

FIGURE 2.2 – Besoins fonctionnels

2.4.2 Besoins non fonctionnels

Les besoins non-fonctionnels décrivent les contraintes que le logiciel doit satisfaire. Ce sont des aspects techniques que le développeur doit respecter afin de produire un logiciel de qualité.

Dans ce cadre, plusieurs éléments essentiels doivent être pris en compte afin d'assurer une utilisation optimale de l'application :

- **Ergonomie** : L'interface doit être simple, intuitive et accessible.
- **Sécurité** : L'application doit garantir la confidentialité des données utilisateurs et assurer la protection contre tout accès non autorisé.
- **Performance** : Les fonctionnalités doivent être rapides à exécuter, avec un temps de réponse réduit. L'organisation de l'interface doit également permettre une navigation fluide grâce à une mise en page claire et intuitive.

2.5 Cas d'utilisation

2.5.1 Définition

Un cas d'utilisation décrit les différentes interactions entre un acteur et le système, permettant à ce dernier d'atteindre un objectif spécifique.[5]

2.5.2 Diagramme de cas d'utilisation

Dans les diagrammes UML, les cas d'utilisation sont représentés par des ellipses portant le nom de l'action ou de la fonctionnalité. Chaque acteur est relié aux cas d'utilisation qui le concernent par une ligne d'association.

Ci-dessous, nous présentons le diagramme des cas d'utilisation :

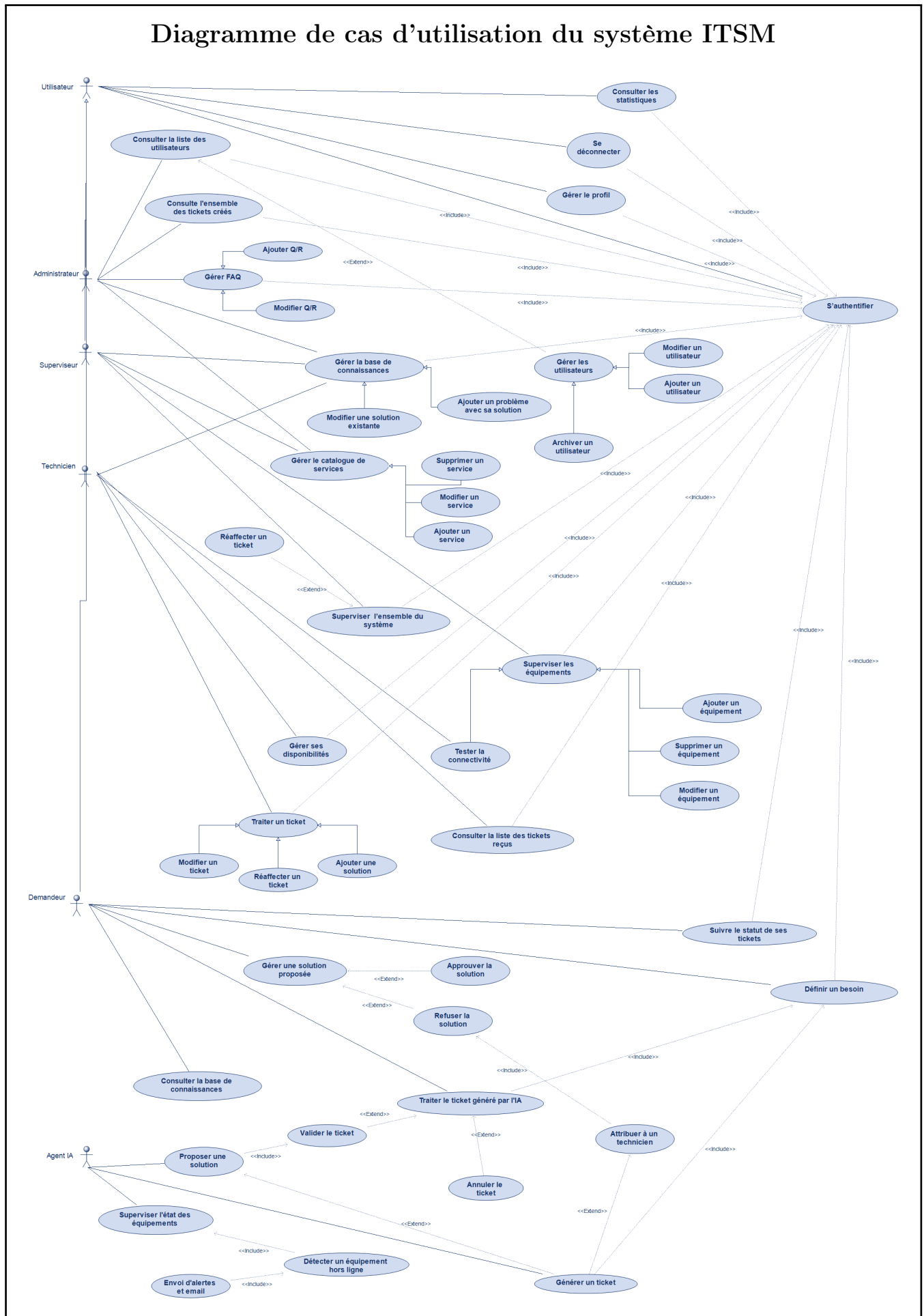


FIGURE 2.3 – Diagramme de cas d'utilisation

2.5.3 Description textuelle des cas d'utilisation

— **Description du cas d'utilisation : Création de ticket**

Nom du cas : Création de ticket

But : permettre au demandeur de formuler un besoin ou un incident afin qu'un ticket soit généré automatiquement dans le système.

Acteur Principal : demandeur

Acteur Secondaire : agent IA

Date : 13/04/2025

Pré-Condition : le demandeur doit être connecté à la plateforme.

Enchaînement Nominal :

1. Le demandeur accède à l'interface du chatbot et saisit librement son besoin.
2. L'agent IA analyse le message et déduit les informations nécessaires pour générer un ticket (type, titre, catégorie, urgence, impact, etc.) en s'appuyant sur un référentiel structuré de services.
3. Le système affiche au demandeur un formulaire pré-rempli avec les données générées par l'IA.
4. Le demandeur a la possibilité de valider ou annuler le ticket proposé.

Post-Condition :

1. Si le ticket est validé, il est enregistré dans la base de données.
2. Un message de confirmation est affiché au demandeur.
3. Si le ticket est annulé, aucune action n'est enregistrée.

Enchaînement d'Exception :

1. Si le message de demandeur est incomplet ou non compréhensible, l'agent IA demande une reformulation claire avant de proposer un ticket.

— **Description du cas d'utilisation : Validation d'une solution**

Nom du cas : Validation d'une solution

But : permettre au demandeur de valider ou refuser une solution proposée afin de décider de la clôture ou de la poursuite du traitement du ticket.

Acteur Principal : demandeur

Acteur Secondaire : agent IA

Date : 13/04/2025

Pré-Condition : une solution doit avoir été proposée à le demandeur.

Enchaînement Nominal :

1. Le demandeur consulte la solution proposée via l'interface.
2. Le demandeur a la possibilité de valider la solution s'il la juge satisfaisante.
3. En cas de validation, la solution est ajoutée au ticket et il sera clôturé.

Post-Condition :

1. La solution validée est enregistrée dans la base de données.
2. Le ticket reste actif jusqu'à la confirmation de la solution par le demandeur.
3. Une confirmation est affichée au demandeur.

Enchaînement d'Exception :

1. Si le demandeur refuse la solution proposée par l'agent IA, celle-ci transmet automatiquement le ticket à un technicien.
2. Si le demandeur refuse également la solution proposée par le technicien, ce dernier peut modifier ou proposer une autre solution.

2.6 Diagramme de Séquences

2.6.1 Se connecter

Ce diagramme montre les étapes de connexion d'un utilisateur. L'utilisateur accède à l'application, le système affiche la page de login. Il saisit ensuite son e-mail et mot de passe. Le système vérifie les informations auprès de la base de données. Si elles sont correctes, l'accès est accordé, sinon un message d'erreur s'affiche.

Le diagramme suivant illustre cet enchaînement :

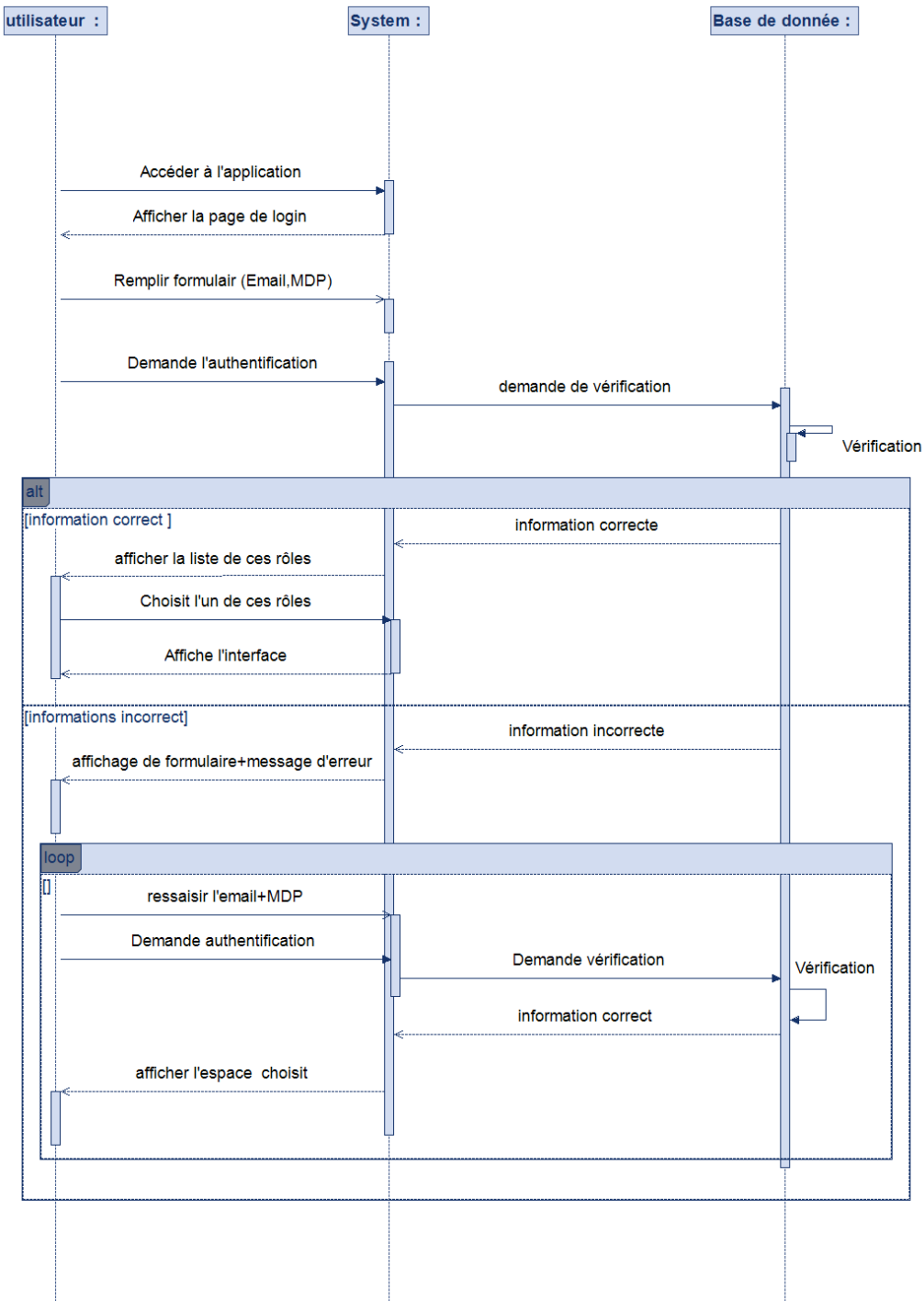


FIGURE 2.4 – Diagramme de séquence «Se connecter»

2.6.2 Générer un ticket

Le scénario débute lorsque le demandeur définit son besoin, qui est ensuite analysé par l'agent IA. Si le message est pertinent, un ticket est généré puis le demandeur peut soit le valider ou l'annuler. L'agent IA recherche une solution dans la base de connaissances si le ticket est validé.

Si une solution est trouvée, elle est proposée au demandeur, qui peut la valider ou la refuser. En cas de validation, le ticket est clôturer ; sinon, il est attribué à un technicien. Si aucune solution n'existe, le ticket est aussi attribué au bon technicien . Si le ticket n'est pas validé, le demandeur peut soit redéfinir son besoin, soit annuler le ticket.

Le diagramme suivant illustre cet enchaînement :

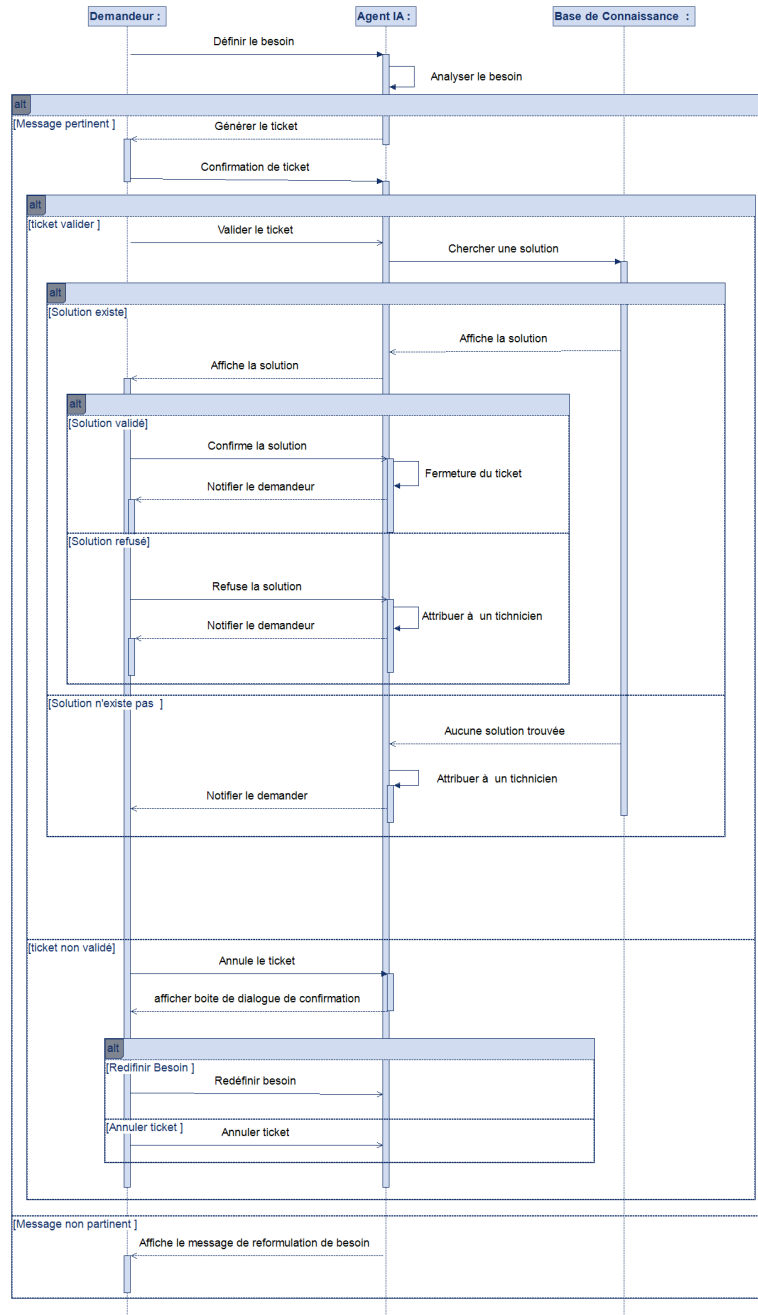


FIGURE 2.5 – Diagramme de séquence «Générer un ticket»

2.6.3 Gérer une solution proposée par le technicien

Ce diagramme de séquence montre les interactions entre le demandeur, le système et la base de données lors du traitement d'une solution liée à un ticket. Le demandeur commence par consulter la solution, que le système récupère depuis la base de données. Ensuite, deux scénarios sont possibles : si la solution est acceptée, le système clôture le ticket et affiche une confirmation de clôture ; si elle est refusée, le ticket repasse au statut "en cours".

Le diagramme suivant illustre cet enchaînement :

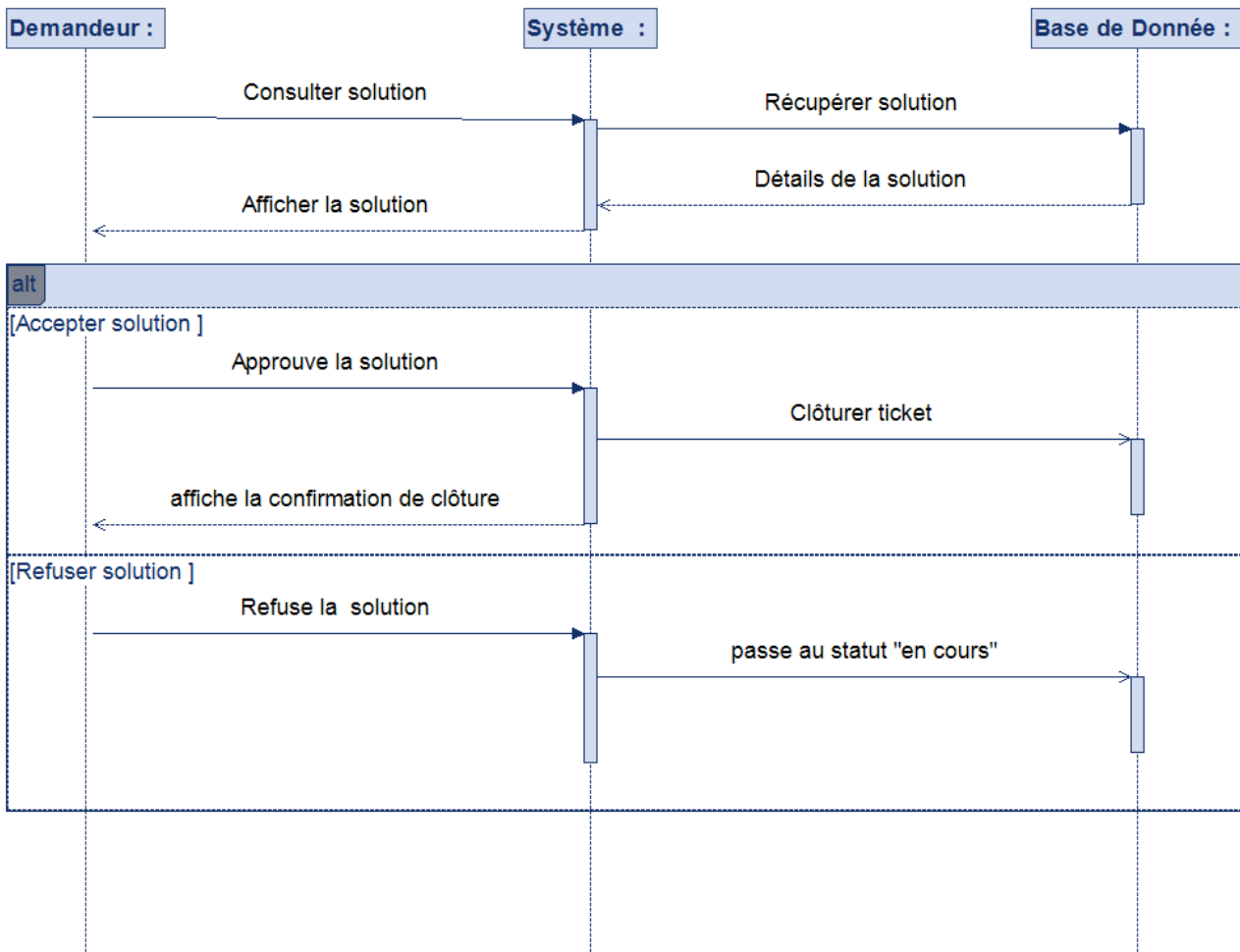


FIGURE 2.6 – Diagramme de séquence «Gérer une solution proposée»

2.6.4 Gérer les utilisateurs

Ce diagramme de séquence illustre le cas "Gérer les utilisateurs" par un administrateur. Après connexion, l'administrateur accède à l'interface de gestion et peut :Ajouter ,Modifier et Archiver

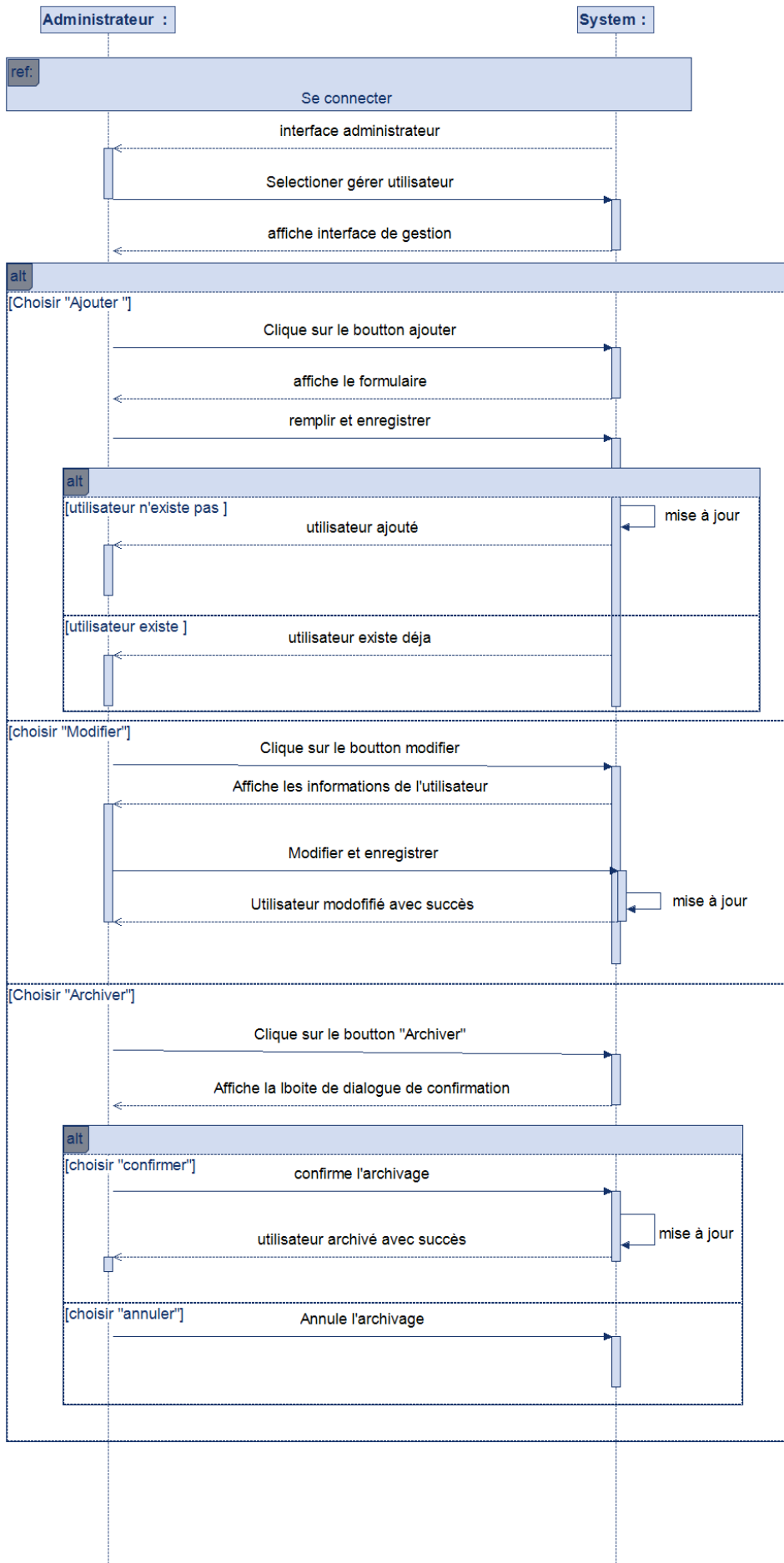


FIGURE 2.7 – Diagramme de séquence «Gérer les utilisateurs»

2.6.5 Traiter un ticket

Ce diagramme de séquence illustre le processus de traitement d'un ticket par un technicien. Le processus débute par l'accès à la liste des tickets disponibles, suivi de la sélection d'un ticket spécifique. Le système interroge alors la base de données pour en récupérer les détails, qui sont ensuite affichés à l'interface du technicien.

Le diagramme présente ensuite plusieurs alternatives possibles selon les actions effectuées par le technicien : Modification du ticket , Réaffectation du ticket ou Ajout d'une solution

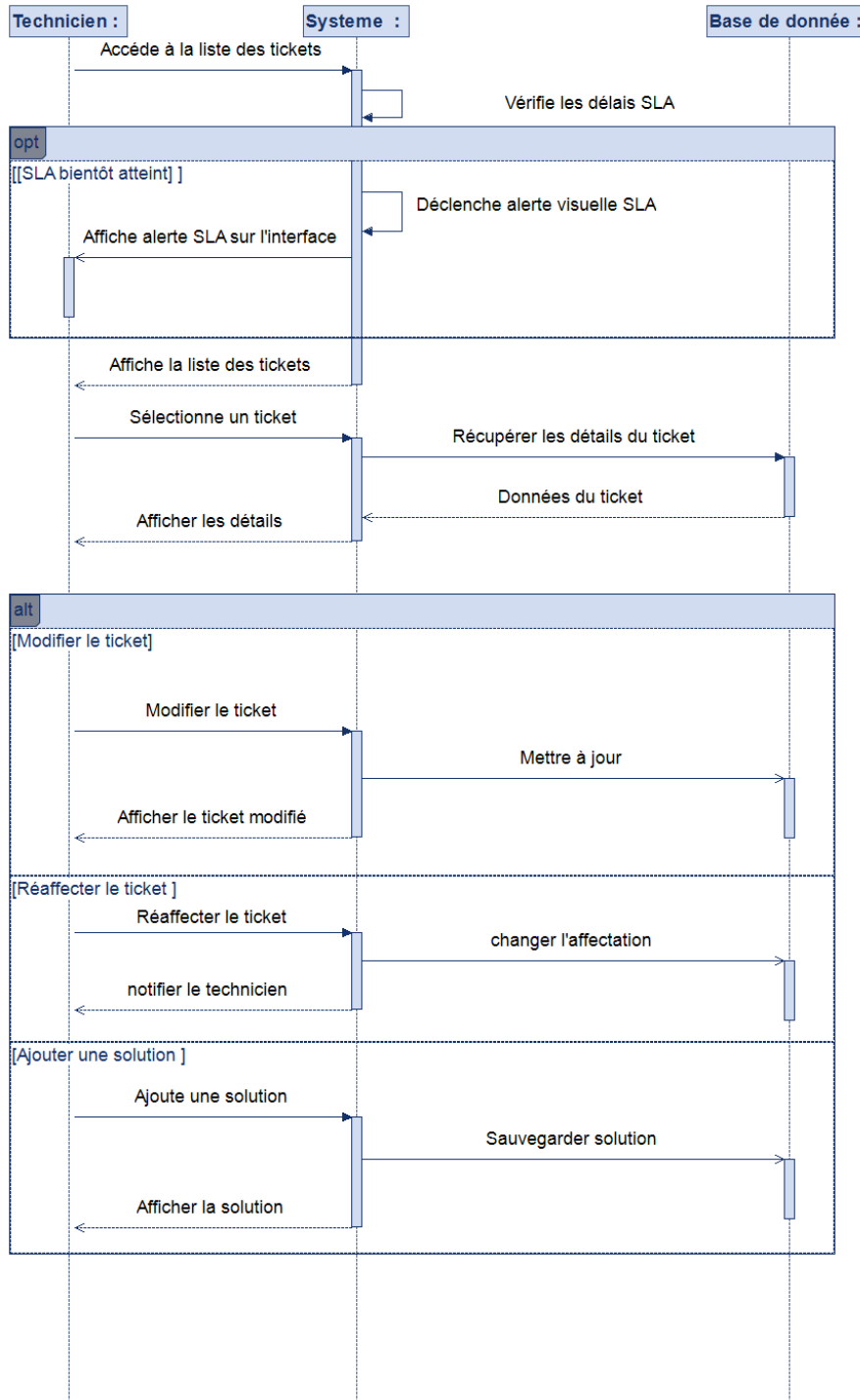


FIGURE 2.8 – Diagramme de séquence «Traiter un ticket»

2.7 Diagramme d'activité

2.7.1 Création d'un ticket

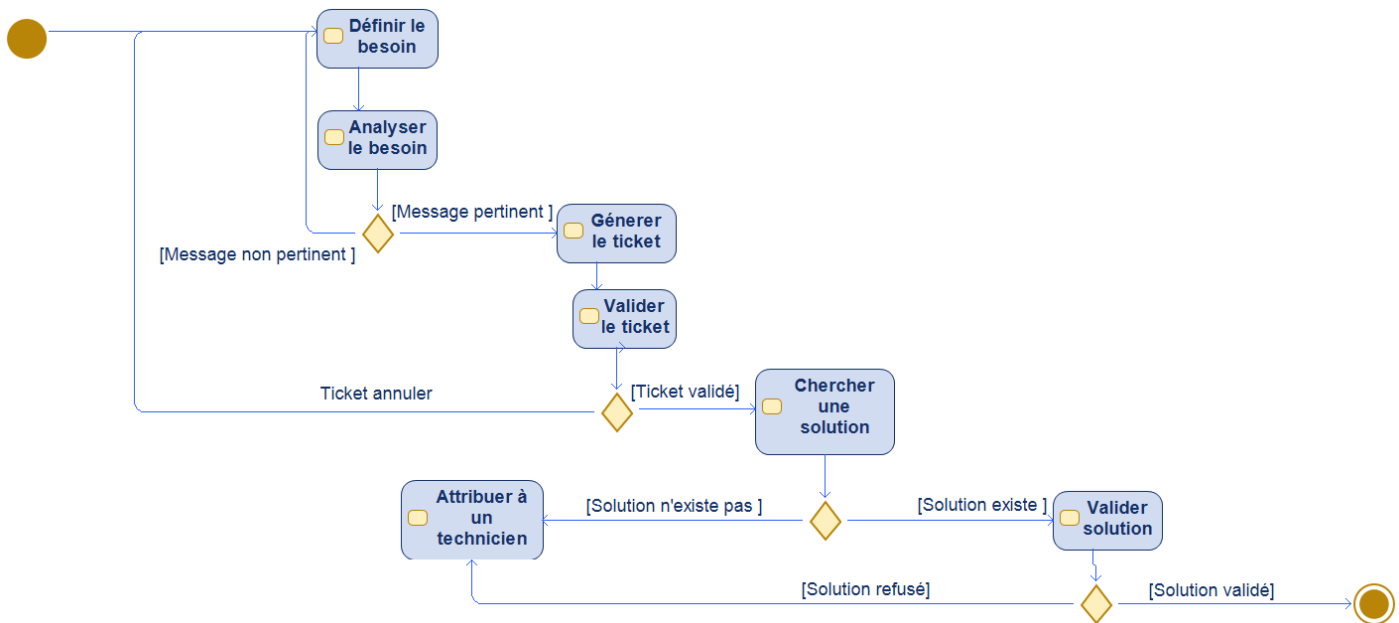


FIGURE 2.9 – diagramme d'activité "créer un ticket "

2.7.2 Gérer une solution proposée par un technicien

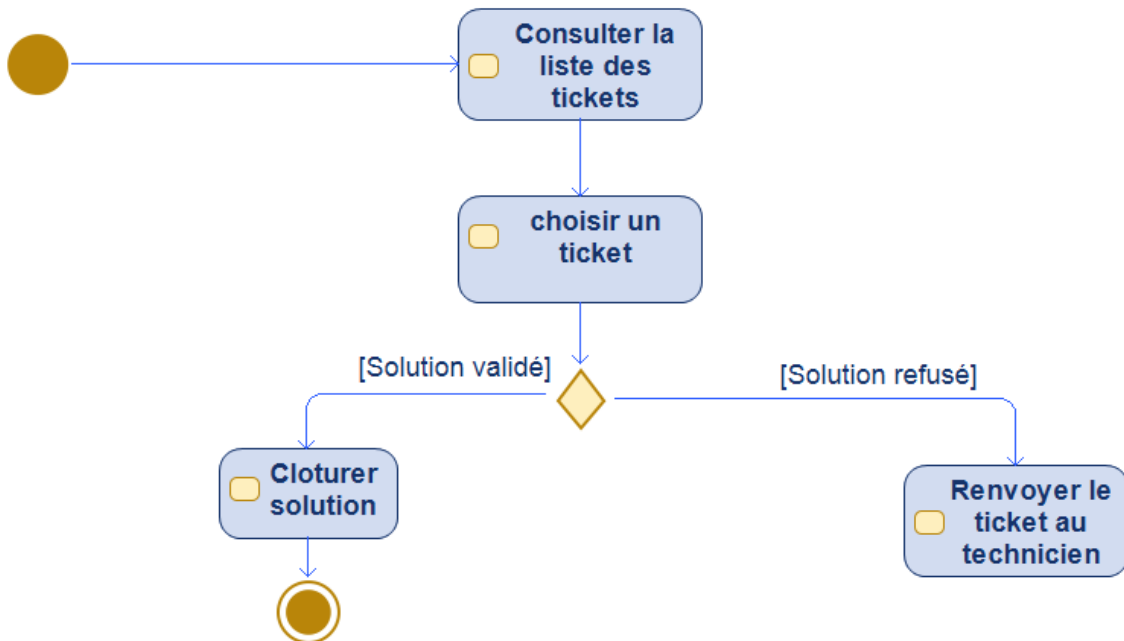


FIGURE 2.10 – Diagramme d'activité "Gérer une solution proposée par un technicien "

2.8 Diagramme de classe

Le diagramme de classes est un outil utilisé en génie logiciel pour représenter les classes et interfaces d'un système, ainsi que les relations qui les lient. Il fait partie des diagrammes structuraux d'UML, car il se concentre sur l'organisation statique du système, sans prendre en compte les aspects dynamiques ou temporels.[6]

Le diagramme de classe de notre projet est représenté comme suit :

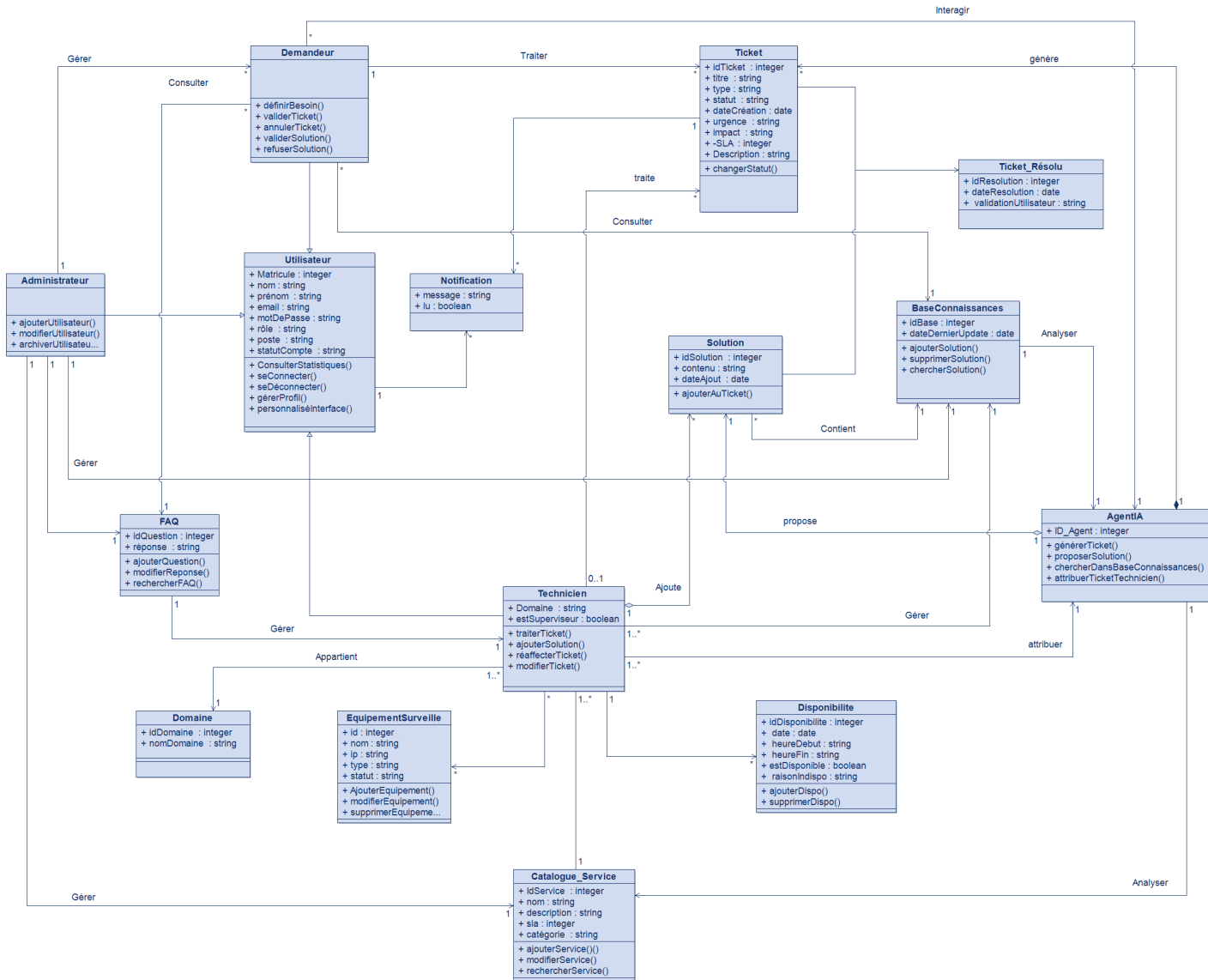


FIGURE 2.11 – Diagramme de classe

2.9 Modèle Logique de données (MLD)

Ce modèle représente le passage du modèle conceptuel de données (MCD) ver un modèle logique adapté à une ase de données relationnelle , en vue de son implémentation dans MySQL :

- **Demandeur** (idDemandeur, nom, prénom, email, motDePasse, rôle, poste, statut-Compte)
- **Administrateur** (idAdmin, nom, prénom, email, motDePasse, rôle, poste, statut-Compte)
- **Technicien** (idTechnicien, nom, prénom, email, motDePasse, rôle, poste, statut-Compte, #idDomaine)
- **Domaine** (idDomaine, nomDomaine, #idSuperviseur)
- **Ticket** (idTicket, titre, type, statut, dateCreation, urgence, impact, SLA, description, etat, domaine, raisonAttente, #idDemandeur, #idTechnicien, #idSuperviseur)
- **Solution** (idSolution, contenu, dateAjout, raisonAttente, #idTechnicien)
- **FAQ** (idQuestion, question, réponse, #idDemandeur, #idTechnicien)
- **AgentIA** (idAgent, #idDemandeur, #idTicket, #idTechnicien)
- **TicketRésolu** (idResolution, dateResolution, validationUtilisateur, #idTicket, #idSolution)
- **disponibilites** (idDisponibilite, date, heureDebut, heureFin, estDisponible, raisonIndispo, #idTechnicien)
- **equipementssurveilles** (id, nom, ip, type, statut, #idTechnicien, #idSuperviseur)

Base de données vectorielle

En complément de la base de données relationnelle implémentée sous MySQL, nous avons intégré une base de données vectorielle, en l'occurrence **Qdrant**, afin de gérer des informations non structurées telles que les documents de la base de connaissances ou le catalogue des services. Ces données, initialement en format JSON pour le catalogue, sont transformées en vecteurs via des modèles d'embedding issus de LLM (Large Language Models). Cette vectorisation permet la mise en place d'une recherche sémantique intelligente, exploitée notamment par l'agent IA pour analyser les similarités entre les requêtes utilisateurs et les ressources disponibles. Grâce à ce système, l'agent intelligent est en mesure de fournir des réponses contextuellement pertinentes et d'interagir de manière plus naturelle avec les services proposés.

Les figures ci-dessous illustrent deux vues complémentaires de cette base vectorielle :

- **Visualisation locale :**

Représentation des voisins les plus proches d'un vecteur. Ce graphe interactif permet d'explorer les services similaires à partir d'un point de référence. Il reflète la capacité de l'agent IA à naviguer dans l'espace vectoriel pour proposer des services pertinents en fonction de la requête.

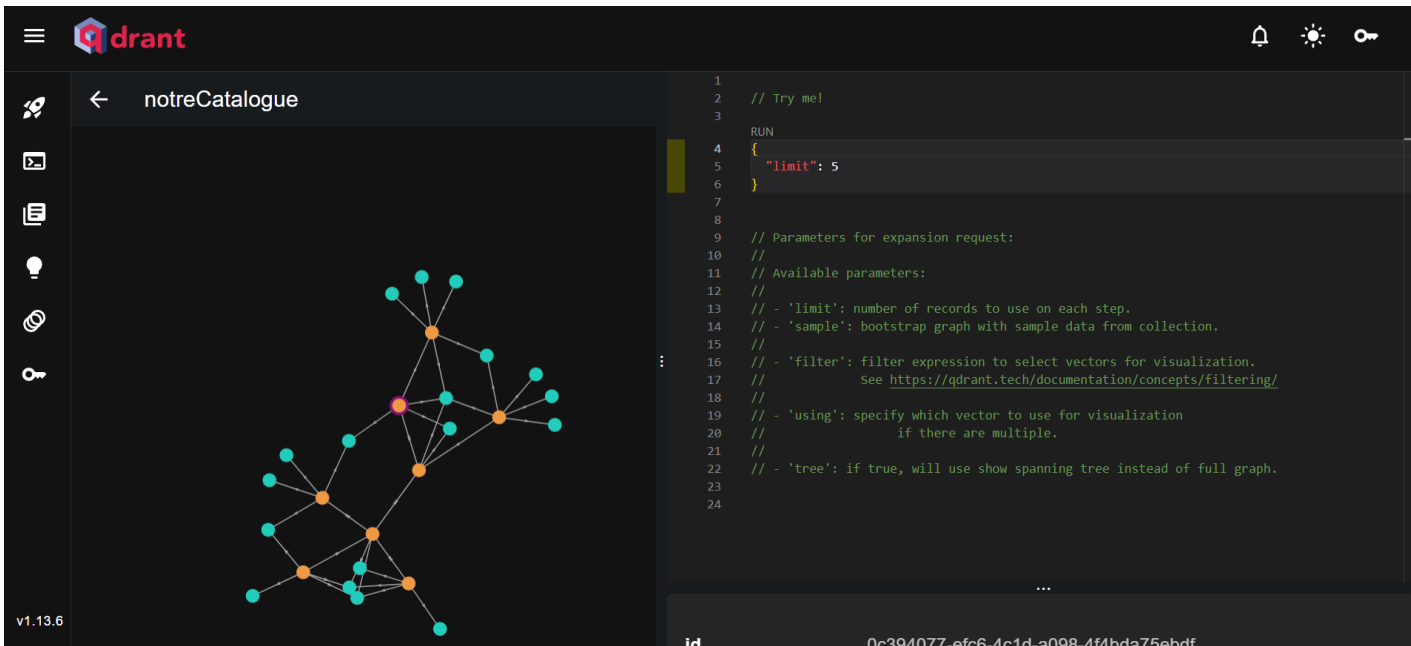


FIGURE 2.12 – Recherche de voisins

- **Visualisation globale :**

Vue d'ensemble de l'espace vectoriel regroupant l'ensemble des services encodés. Chaque point représente une entrée vectorielle issue du catalogue. Cette projection permet de visualiser la structure globale des données et de détecter des clusters sémantiques utiles pour les suggestions automatiques de l'agent IA.

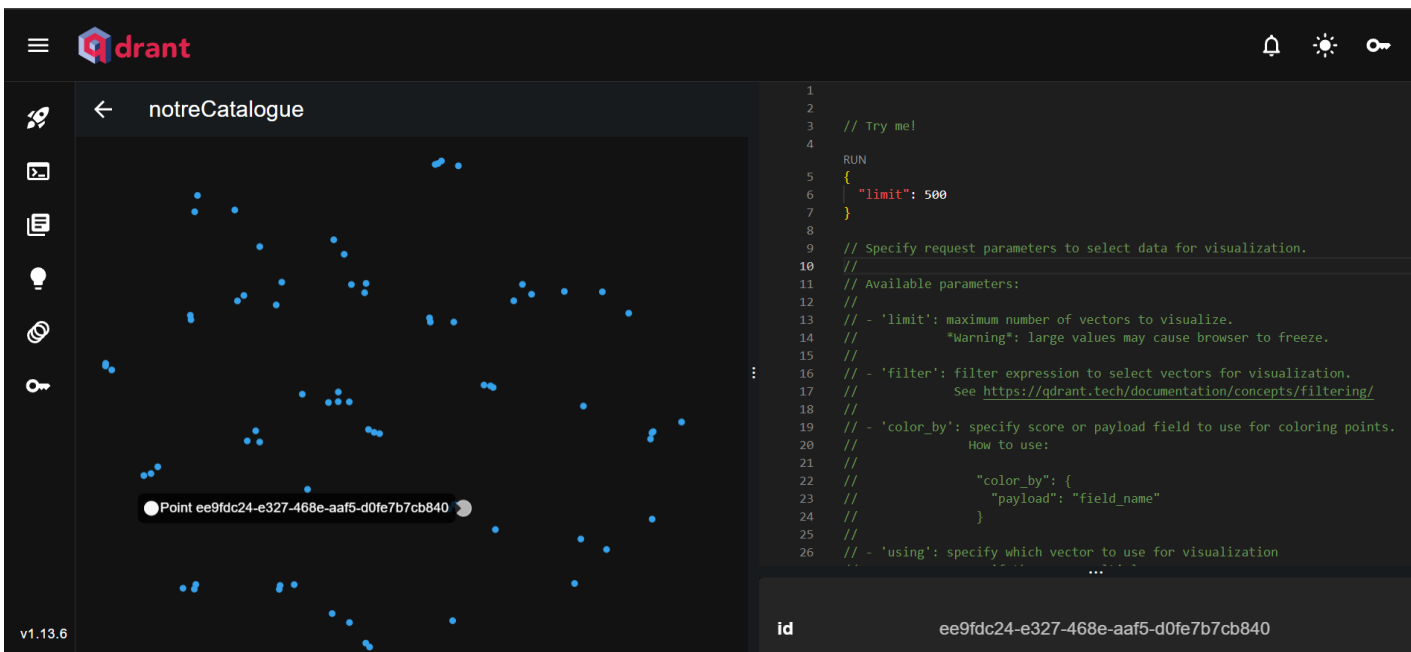


FIGURE 2.13 – Projection de la base

2.10 Conclusion

Dans ce chapitre, nous avons présenté l'ensemble des éléments nécessaires à la conception de notre solution. Nous avons décrit le contexte de gestion des demandes et incidents, identifié les acteurs et défini leurs rôles. Nous avons ensuite exprimé les besoins fonctionnels et non fonctionnels, et illustré les interactions du système à travers des cas d'utilisation, des diagrammes de séquence, d'activité et de classes. Nous avons également élaboré le Modèle Logique de Données (MLD), qui constitue une étape essentielle pour assurer la structuration cohérente et efficace de la base de données.

Cette phase de conception nous a permis de structurer notre réflexion et de poser une base solide pour le développement de la solution, en veillant à répondre aux exigences du projet tout en assurant sa cohérence avec les principes d'ITIL4.

CHAPITRE 3

*Mise en œuvre technique et environnement
de développement*

Chapitre 3

3.1 Introduction

Dans cette section, nous présentons l'ensemble des technologies et outils mobilisés pour concevoir et implémenter notre solution de gestion intelligente des demandes et incidents. Le choix des langages, des frameworks, des plateformes d'automatisation ainsi que des outils de modélisation a été guidé par les exigences de performance, de sécurité, d'évolutivité et de conformité avec les bonnes pratiques ITIL4. Ce chapitre détaille d'abord les environnements techniques adoptés, tant côté client (front-end) que serveur (back-end), ainsi que les formats d'échange utilisés. Il aborde ensuite l'automatisation des processus à travers la plateforme n8n et l'intégration d'un agent intelligent capable de traiter, structurer et enrichir les demandes utilisateur. Un accent particulier est mis sur la gestion automatisée des alertes en temps réel, afin d'assurer une réactivité optimale face aux événements critiques. Enfin, une attention particulière est portée aux mécanismes de sécurisation des flux de données. Enfin, une attention particulière est portée aux mécanismes de sécurisation des flux de données et des échanges, afin de garantir l'intégrité et la confidentialité des informations manipulées tout au long du processus.

3.2 Environnement et outils mobilisés

Le développement de notre solution s'appuie sur une combinaison de technologies modernes, d'outils de modélisation et d'automatisation, ainsi que sur un ensemble de formats standards pour assurer l'interopérabilité entre les différents composants du système. Cette section présente les outils utilisés selon leur rôle dans le projet, en distinguant les technologies front-end, back-end, les bases de données, les outils de modélisation et de documentation.

3.2.1 Technologies du front-end

L'interface utilisateur a été développée avec **ReactJS**, une bibliothèque JavaScript puissante conçue pour créer des interfaces interactives, dynamiques et réactives. Elle permet de construire l'application sous forme de composants modulaires, favorisant la réutilisabilité du code et la séparation claire des responsabilités. Ce choix a également été motivé par la forte intégration de React avec des services externes via les API, et sa compatibilité avec des environnements modernes comme n8n.

3.2.2 Technologies du back-end

- **Node.js** : environnement d'exécution JavaScript utilisé côté serveur pour gérer les opérations asynchrones, les API, et la logique métier. Il permet une communication fluide avec le front-end et s'intègre parfaitement avec les services d'automatisation.
- **n8n** : plateforme centrale d'automatisation dans notre architecture. Elle permet de connecter tous les outils entre eux, d'orchestrer les scénarios de traitement automatisé, et d'assurer la communication entre l'agent IA, la base de données et l'interface utilisateur. Grâce à ses workflows visuels, elle centralise les échanges entre les différentes composantes du système

3.2.3 Technologies de gestion et de stockage des données

- **PHPMyAdmin** : interface web permettant la gestion d'une base de données MySQL utilisée comme base centrale du système.
- **NoCoDB** : outil no-code transformant une base de données en interface collaborative, utilisé ici pour héberger un catalogue de services et une base de connaissances, consulté par l'agent IA via n8n.
- **Qdrant** : base de données vectorielle spécialisée dans la recherche sémantique, utilisée par l'agent IA pour enrichir ses réponses via la technique de RAG (Retrieval-Augmented Generation).

3.2.4 Outils de modélisation, documentation et design

- **Modelio** : outil de modélisation UML utilisé pour créer les diagrammes de cas d'utilisation, de classes, de séquence, etc.
- **XMind** : logiciel de création de cartes mentales utilisé pour organiser les idées, fonctionnalités et scénarios du projet.
- **Canva** : outil de conception graphique utilisé pour produire des visuels intégrés dans le mémoire.
- **LaTeX** : langage de composition de documents utilisé pour rédiger le mémoire avec une mise en forme académique.
- **Visual Studio Code (VS Code)** : éditeur de code utilisé pour développer et organiser l'ensemble des fichiers du projet.

3.2.5 Technologies de réseau et de communication sécurisée

- **ZeroTier** : outil de mise en réseau virtuelle permettant d'assurer la communication sécurisée entre machines distantes dans une architecture distribuée.[15]













Outil / Technologie	Rôle dans le projet	Justification du choix
ReactJS 	Développement de l'interface utilisateur	Interface dynamique, composants réutilisables, grande compatibilité avec les API
Node.js 	Traitement serveur, API, logique métier	Environnement léger, performant, unification du code avec JavaScript
n8n 	Automatisation des processus	Plateforme visuelle open source, intégration simple avec IA, webhook, bases de données
PHPMysqlAdmin 	Gestion de la base principale (tickets, utilisateurs...)	Interface web intuitive, compatibilité avec MySQL, facile à administrer
NoCoDB 	Catalogue de services exploité par l'agent IA via N8N	Interface no-code flexible, lecture directe par webhook/API
Qdrant 	Recherche vectorielle utilisée par l'agent IA	Requêtes sémantiques performantes, base optimisée pour RAG
Modelio 	Création des diagrammes UML	Outil professionnel et adapté à la modélisation logicielle
XMind 	Planification et organisation des idées	Visualisation claire des structures logiques
Canva 	Conception de visuels pour le mémoire	Création rapide de schémas et éléments graphiques
LaTeX 	Rédaction du mémoire	Qualité typographique, rigueur académique
VS Code 	Éditeur de développement	Léger, puissant, adapté à React, Node.js et JSON
Zerotier 	Assure la communication sécurisée entre client et serveur dans une architecture distribuée	Création d'un VPN local sécurisé, encapsulation du trafic via tunnel chiffré, accès restreint aux machines autorisées

FIGURE 3.1 – Outils

3.3 Automatisation et intelligence du système

3.3.1 n8n, plateforme d'automatisation

3.3.1.1 Définition

n8n (Node for Node) est une plateforme open-source d'automatisation des flux de travail (workflow automation). Elle permet de concevoir des scénarios automatisés en connectant visuellement différents services, applications ou bases de données, à travers une interface graphique intuitive fondée sur le principe des nœuds interconnectés. Chaque nœud représente une action, une condition ou une transformation.[12]

3.3.1.2 Historique et évolution de n8n

- **2019 – Lancement du projet**

n8n est créé par **Jan Oberhauser** avec pour objectif de proposer une alternative open-source, flexible et auto-hébergeable aux outils d'automatisation propriétaires tels que **Zapier** ou **Integromat**. Dès sa création, n8n repose sur une architecture modulaire basée sur des nœuds interconnectés, permettant de concevoir des flux d'automatisation visuellement.

- **2020–2021 – Développement communautaire et stabilité**

Le projet commence à attirer l'attention de nombreux développeurs et passionnés de solutions libres. Une communauté open-source active se constitue rapidement, contribuant à l'ajout progressif de nouveaux connecteurs (nœuds) et à l'amélioration de l'interface. Durant cette période, n8n gagne en maturité fonctionnelle et devient progressivement utilisable dans des contextes de développement avancés.

- **2022–2023 – Adoption croissante en entreprise**

L'automatisation des processus devient une priorité pour de nombreuses organisations. n8n commence à être intégré dans des environnements professionnels, notamment dans les domaines de la gestion de données, du support technique ou de l'administration interne. Sa capacité à être hébergé en local et personnalisé selon les besoins spécifiques des entreprises contribue fortement à son adoption.

- **2024–2025 – Explosion de la visibilité avec l'essor de l'IA et du cloud hybride**

Avec l'avènement des agents intelligents, de l'intelligence artificielle générative (comme ChatGPT, RAG...) et des systèmes connectés à distance, n8n devient un pivot technologique dans les architectures modernes. Les entreprises recherchent de plus en plus de solutions personnalisables, souveraines et interopérables, ce qui renforce l'attrait de n8n. [10].

3.3.1.3 Caractéristiques techniques de n8n

La plateforme n8n se distingue par une architecture technique moderne, modulaire et flexible, qui permet de construire des workflows automatisés tout en s'adaptant à différents environnements d'hébergement et de développement. Ses caractéristiques sont pensées pour répondre aux exigences de robustesse, de compatibilité et d'évolutivité dans les systèmes d'automatisation.

- **Technologie**

n8n est développé en TypeScript, un langage basé sur JavaScript qui introduit le typage statique. Cela renforce la fiabilité du code, facilite le débogage, et permet une meilleure évolutivité du projet, notamment lors de l'intégration de nouveaux nœuds ou de l'ajout de fonctionnalités personnalisées.

- **Interface utilisateur**

L'interface de n8n est entièrement visuelle, conçue autour d'un système de glisser-déposer (drag-and-drop). Elle permet aux utilisateurs de concevoir leurs flux sans avoir à écrire de code, ce qui rend la plateforme accessible aussi bien aux profils techniques qu'aux utilisateurs non développeurs.

- **Structure des workflows**

Les workflows sont composés de nœuds interconnectés, chacun représentant une tâche spécifique (action, condition, transformation, etc.). La structure des workflows repose sur une exécution séquentielle ou conditionnelle des nœuds, permettant de gérer des scénarios simples comme complexes avec logique conditionnelle, boucles et filtrage.

- **Déclenchement**

Un workflow n8n peut être déclenché de plusieurs façons :

- Manuellement depuis l'interface.
- Par horaire programmé.
- Par évènement système (données reçues, état d'un fichier, etc.).
- À l'appel d'un point d'entrée externe (comme un webhook).

- **Hébergement**

La plateforme est conçue pour être auto-hébergeable :

- sur une machine locale (développement ou tests).
- sur un VPS (serveur distant).
- dans un conteneur Docker.
- ou via des services cloud privés ou publics.

- **Personnalisation**

n8n offre un haut degré de personnalisation :

- Des nœuds “Function” permettent d’injecter du code JavaScript personnalisé pour traiter ou manipuler les données.
- Il est également possible de développer des nœuds sur mesure et de les intégrer à la plateforme pour étendre ses capacités.

3.3.2 Connexion entre l’interface et la plateforme d’automatisation

La connexion entre l’interface utilisateur développée en ReactJS et la plateforme d’automatisation n8n repose sur l’utilisation de webhooks et d’API REST. Ces deux mécanismes permettent d’assurer une communication fluide, dynamique et sécurisée entre les composants du système, tout en garantissant leur indépendance structurelle.

3.3.2.1 Les Webhooks

Un webhook est une URL d’écoute configurée pour recevoir des requêtes HTTP POST dès qu’un événement spécifique est déclenché. Contrairement aux appels API classiques, le webhook n’attend pas une interrogation régulière mais agit immédiatement.

Dans le cadre de notre projet, les webhooks sont utilisés pour :

- Recevoir les besoins utilisateur saisis dans l’interface ;
- Déclencher des workflows automatisés dans n8n selon l’action de l’utilisateur (valider, refuser, etc.).

L’avantage principal des webhooks est leur réactivité immédiate : dès qu’une action est effectuée côté client, le traitement s’enclenche automatiquement sans requête périodique ou chargement inutile.

3.3.2.2 Les API REST

Une API REST (Application Programming Interface) permet la communication entre n8n et des services externes (ou internes) via des appels HTTP. Les échanges se font généralement au format JSON, ce qui facilite l’intégration et la lecture des données.

Dans notre système, les APIs sont utilisées pour :

- Connecter n8n à des services d’intelligence artificielle externes (ex. : Google Gemini),
- Interroger ou enrichir la base vectorielle (Qdrant),
- Lire ou écrire dans la base de données NoCoDB, où sont stockés des fichiers JSON (catalogue de services, base de connaissances...).

Grâce à ces API, n8n agit comme un orchestrateur central, capable d’exploiter des services tiers, d’analyser des données, et de mettre à jour dynamiquement les bases internes du système.

3.3.2.3 Les avantages

L'architecture fondée sur les webhooks et API présente plusieurs atouts majeurs :

- **Réactivité** : les actions sont exécutées instantanément dès l'envoi d'une requête, sans attente.
- **Modularité** : chaque composant du système (React, n8n, IA, base de données) peut évoluer indépendamment.
- **Extensibilité** : de nouveaux services peuvent être facilement intégrés via des API sans refondre tout le système.
- **Sécurité** : les webhooks sont protégés, et les API contrôlent les accès aux ressources.
- **Maintenance facilitée** : les échanges sont traçables, les workflows sont visuellement modifiables dans n8n.

3.3.3 Scénarios d'automatisation

Dans cette section, nous décrivons les grands principes qui structurent l'automatisation du système. Après avoir présenté la plateforme d'automatisation (n8n) et ses mécanismes d'interconnexion (webhooks et APIs), nous abordons ici les scénarios intelligents mis en place pour traiter les besoins des utilisateurs, générer des tickets, et proposer des solutions pertinentes à l'aide de l'intelligence artificielle, ainsi que superviser les équipements afin de détecter les pannes et d'envoyer automatiquement des alertes et des e-mails.

Mais avant cela, nous introduisons brièvement les concepts fondamentaux sur lesquels repose cette intelligence automatisée.

3.3.3.1 Notions de base : agent IA, LLM, agent RAG

1. Agent IA

Un agent IA est un module autonome qui reçoit un besoin, le traite de manière intelligente, et fournit un résultat structuré, sans intervention humaine. Il agit comme un traducteur entre le langage utilisateur et les données techniques attendues par le système.

2. LLM – Large Language Model

Un LLM (Large Language Model) est un modèle d'intelligence artificielle entraîné sur de grands volumes de textes. Il est capable de comprendre le langage humain, d'extraire des informations, et de générer des textes structurés. Ce type de modèle est utilisé pour analyser les besoins exprimés par l'utilisateur, souvent formulés de façon floue ou non technique, et de produire des sorties structurées.

Dans notre projet nous avons utilisé Google Gemini, un LLM :

- Accessible gratuitement.
- Simple à intégrer via API.
- Particulièrement performant en français, ce qui représente un atout dans notre contexte.

Gemini nous a permis d'obtenir des résultats cohérents, rapides et précis, sans coût d'abonnement, ce qui en fait un choix stratégique et pertinent.

3. RAG – Retrieval-Augmented Generation

Un agent RAG combine la recherche de contenus pertinents dans une base vectorielle (Qdrant) avec la génération linguistique par un LLM. Il permet à l'IA de s'appuyer sur un historique réel et validé, pour répondre de façon enrichie et contextualisée.

3.3.3.2 Alimentation intelligente des données

Objectif :

Créer un environnement intelligent pour permettre à l'agent IA d'accéder à des données enrichies, vectorisées et consultables sémantiquement.

Fonctionnement global :

1. Extraction des ressources depuis NoCoDB : Les fichiers JSON contenant le catalogue de services et la base de connaissances sont récupérés.
2. Préparation des données : Les contenus sont lus, analysés et découpés en segments (chunks) grâce à des nœuds dédiés, afin de faciliter leur vectorisation.
3. Conversion en vecteurs sémantiques : Les segments de texte sont transformés en vecteurs numériques via l'Embedding Google Gemini, une technologie d'encodage sémantique.

4. Injection dans Qdrant : Les vecteurs sont stockés dans Qdrant Vector Store, où ils deviennent consultables de manière sémantique et rapide par l'agent IA.

L'intérêt ici est d'alimenter une base de mémoire intelligente : Qdrant permet d'indexer les documents selon leur signification, et non leur forme. Cette couche vectorielle donne à l'IA un accès immédiat à des données enrichies, précises, et contextuelles, ce qui augmente la pertinence et la qualité des tickets générés.

Le schéma suivant illustre le flux de travail implémenté dans n8n

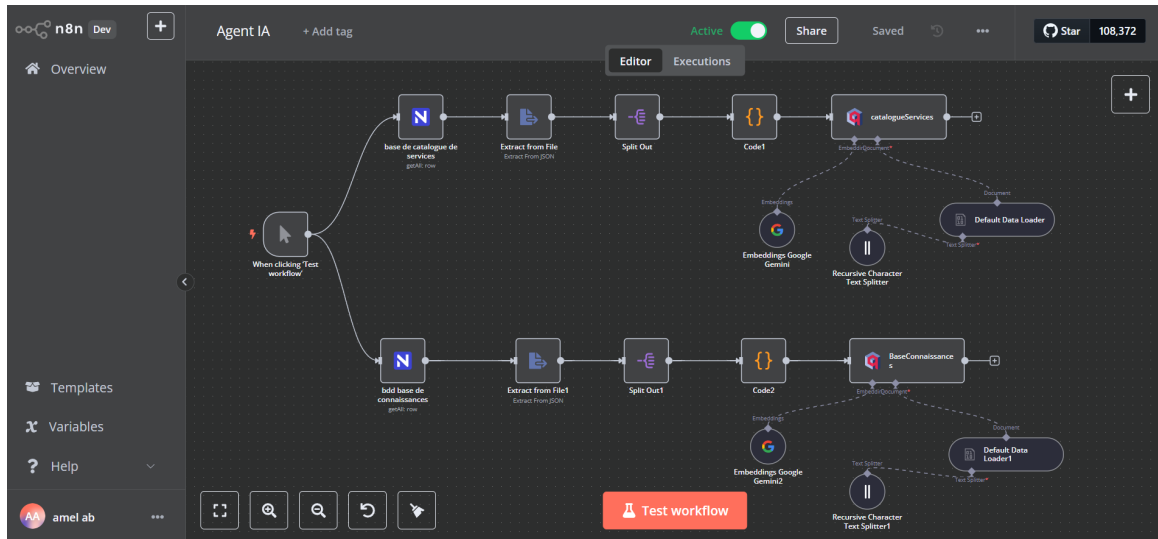


FIGURE 3.2 – Flux de traitement des documents vers Qdrant via n8n

3.3.3.3 Automatisation du traitement du demandeur

A Automatisation de la création du ticket demandeur :

Objectif :

Générer automatiquement un ticket structuré à partir d'un besoin utilisateur exprimé librement.

Fonctionnement global

- 1 L'utilisateur envoie son besoin via l'interface React.
- 2 Le besoin est transmis à n8n via un webhook.
- 3 L'agent IA analyse la demande et en déduit :
 - le type (demande ou incident),
 - le titre, la catégorie, l'impact, l'urgence, le SLA, etc.

Pour cela, il s'appuie sur deux ressources essentielles :

- Le catalogue de services, pour garantir que les valeurs générées soient conformes aux standards internes .
- La base de connaissances, afin d'identifier des cas similaires déjà traités.

C'est ici que l'usage d'une base vectorielle prend tout son sens : elle permet à l'agent IA de ne pas se limiter à une correspondance exacte de mots, mais d'exploiter une recherche sémantique. Grâce à l'embedding, chaque besoin est converti en vecteur, et comparé à ceux existants. Cette approche permet une recommandation intelligente, fondée sur la proximité du sens, et non sur des mots-clés rigides.

B Attribution automatique à un technicien

Objectif :

Dans le cas où l'utilisateur demande l'intervention d'un technicien, un nouveau processus se déclenche :

- L'agent IA identifie la catégorie du problème.
- Il interroge une base SQL contenant la liste des techniciens et leurs spécialités.
- Le ticket est attribué automatiquement au technicien le plus pertinent.
- Des notifications sont envoyées :
 - à l'utilisateur pour l'informer de la prise en charge,
 - au technicien concerné avec les détails du ticket.

Le schéma suivant résume visuellement l'ensemble du processus d'automatisation, depuis la soumission de la demande par l'utilisateur jusqu'à l'attribution automatique du ticket à un technicien via l'agent IA orchestré par n8n :

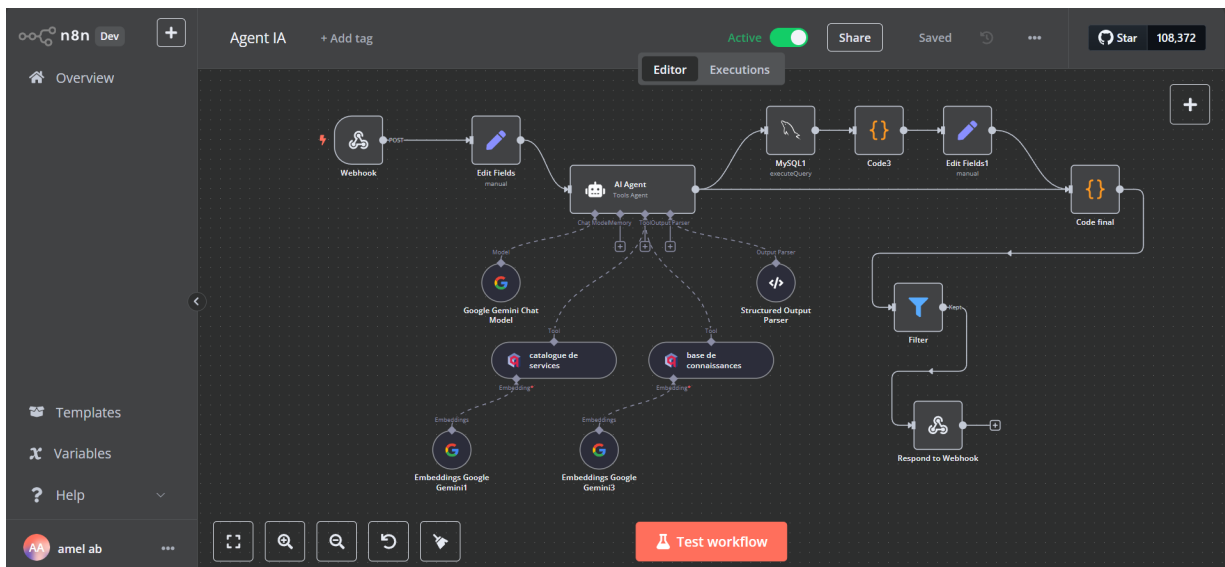


FIGURE 3.3 – Résumé du processus d'automatisation de traitement et d'attribution des tickets

3.3.3.4 Nœuds clés utilisés

Parmi les nœuds essentiels qui structurent ce scénario :

- **AI Agent** : centralise l'appel au LLM, la mémoire et les bases.
- **Google Gemini Chat** : moteur de génération.
- **Embedding Gemini** : convertit les textes en vecteurs pour Qdrant.
- **Catalogue de services (NoCoDB)** : sert de référentiel à l'IA.
- **Base de connaissances** : permet la recherche de cas similaires.
- **Structured Output Parser** : transforme la sortie en JSON structuré.
- **SQL Query** : Récupère les techniciens correspondants depuis la base SQL.
- **Respond to Webhook** : envoie la réponse à l'interface.

Chaque nœud a été configuré stratégiquement pour renforcer la cohérence du raisonnement de l'IA, sans jamais alourdir le processus.

3.4 Systèmes intelligents de gestion proactive

3.4.1 Système de notifications dynamiques

Intégration d'un système de notifications intelligent (**Technicien & Demandeur**) Dans le cadre de notre solution, un système de notifications dynamique a été conçu afin de renforcer la communication en temps réel entre les différents acteurs du système d'assistance (techniciens et utilisateurs demandeurs).

- **Côté technicien**, les notifications permettent d'informer l'agent lorsqu'un nouveau ticket lui est attribué. Le message indique explicitement le numéro et le titre du ticket concerné. En cliquant sur la notification, le technicien est redirigé vers l'interface de gestion détaillée du ticket, où il peut ajouter une solution ou effectuer les traitements nécessaires. Cela favorise une réaction rapide, en particulier pour les tickets urgents ou ceux proches de l'échéance SLA.
- **Côté demandeur**, les notifications informent l'utilisateur dès qu'un de ses tickets est résolu par un technicien. Le message inclut également le numéro du ticket concerné, et un clic sur la notification redirige directement vers le détail du ticket, assurant une transparence et une traçabilité complète.

Le badge de notification se met à jour dynamiquement en temps réel, assurant une expérience fluide. Cette réactivité améliore l'ergonomie et l'efficacité de l'interface.

Ce module s'intègre dans une logique de gestion proactive des services IT, en assurant une remontée immédiate des événements. Il incarne une automatisation intelligente alignée sur les principes ITIL4, notamment la transparence, la réactivité et la création de valeur pour l'utilisateur.

3.4.2 Système d'alerte SLA en temps réel

Dans le cadre de l'optimisation du suivi des tickets côté technicien, un système d'alerte en temps réel a été mis en place pour détecter les tickets dont le délai de résolution (**SLA – Service Level Agreement**) approche de l'expiration. Cette fonctionnalité repose sur un calcul dynamique du temps restant à partir de la date de création du ticket et de la durée SLA attribuée. Lorsqu'un ticket a moins de 2 heures restantes avant le dépassement du SLA, une alerte visuelle s'affiche automatiquement à l'écran pour avertir le technicien.

Ce mécanisme relève de la surveillance temps réel, un principe fondamental dans les systèmes critiques. Il implique un traitement périodique des données en local et l'affichage de l'information pertinente sans nécessiter de rechargement manuel. Ce choix technologique est essentiel dans les environnements IT sensibles où le respect des délais conditionne la qualité de service.

Cette approche contribue à :

- Garantir la réactivité des équipes techniques.
- Réduire les risques d'incidents non traités à temps.
- Optimiser la gestion proactive des ressources réseau en évitant l'accumulation des délais critiques.

3.4.3 Système d'alerte en cas de défaillance réseau

Le système développé intègre une double alerte en cas de déconnexion d'un équipement critique. D'une part, une notification visuelle s'affiche directement sur l'interface superviseur, permettant une réaction rapide. D'autre part, une alerte est automatiquement envoyée par email via n8n, contenant les informations essentielles (nom de l'équipement, adresse IP, date, heure). Cette double couche d'alerte renforce la supervision proactive et garantit une disponibilité continue du réseau.

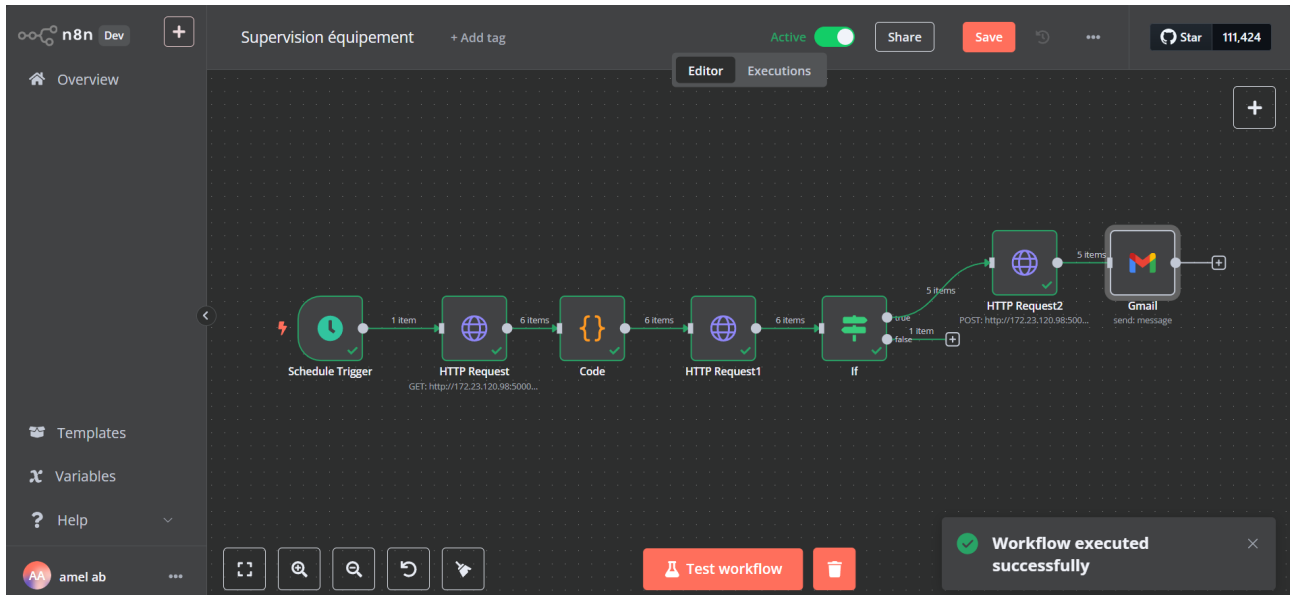


FIGURE 3.4 – Supervision automatiser

3.5 Sécurisation du système et des échanges

La sécurité a été intégrée à plusieurs niveaux de l'architecture, depuis l'interface utilisateur jusqu'à l'échange des données entre les services, en passant par la gestion des accès et des communications API. Cette section détaille les principales pratiques mises en œuvre.

3.5.1 Sécurité de l'interface utilisateur

1. Validation des entrées utilisateur

- Vérification de la structure de l'email et de la longueur du mot de passe avant d'envoyer la requête.
- Affichage de messages d'erreur personnalisés pour guider l'utilisateur (Email invalide, Mot de passe trop court).
- Gestion des erreurs réseau ou serveur (Erreur réseau, veuillez réessayer).

2. Transmission sécurisée des données

- Utilisation du Content-Type application/json pour garantir une transmission structurée des données.
- Envoi des données via la méthode POST uniquement, évitant les expositions dans l'URL.

3. Gestion du token JWT côté client

Après authentification réussie, le token JWT est stocké dans le localStorage sous le

nom `authToken`, permettant à l'utilisateur d'être reconnu lors des prochaines actions protégées.

4. Navigation sécurisée après authentification

La redirection vers l'interface principale est bloquée tant que l'utilisateur n'a pas été authentifié avec succès.

3.5.2 Sécurisation des appels API et des webhooks

1. Authentification et gestion des sessions avec JWT

- Nous avons mis en place une authentification basée sur JWT, généré après vérification des identifiants.
- Le JWT intègre l'email et le rôle de l'utilisateur, signé avec une clé secrète avec une durée limitée .

2. Détection dynamique des rôles

- Une analyse automatique de l'email permet d'attribuer dynamiquement le rôle (admin, technicien ou utilisateur).
- Le rôle est inclus dans le token JWT, permettant de restreindre l'accès selon les privilèges.

3. Hachage sécurisé des mots de passe

Tous les mots de passe sont hachés avec Bcrypt avant stockage, garantissant une non-exposition en cas de fuite de données.

4. Filtrage des origines avec CORS

Seules les requêtes provenant du client React sont acceptées, empêchant toute tentative externe non autorisée.

3.5.3 Sécurisation des Webhooks n8n

Nous avons intégré la protection des Webhooks n8n avec les pratiques suivantes :

1. Authentification des Webhooks par JWT

Chaque webhook est protégé par un JWT obligatoire, permettant à n8n de :

- Valider l'identité de la source de la requête.
- Empêcher les appels externes non autorisés.
- Sécuriser l'accès aux workflows critiques.

2. Restriction des traitements

Seules les requêtes authentifiées par JWT sont acceptées pour lancer les automatisations, empêchant toute exécution abusive.

3.5.4 Sécurisation des données échangées et stockées

1. Sécurisation des bases de données

- L'accès aux données (tickets, utilisateurs, techniciens) est contrôlé via Sequelize, protégeant contre les injections SQL.
- Les bases NoCoDB et Qdrant sont consultées uniquement par API sécurisée, sans exposition directe.

2. Structure des échanges en JSON

Toutes les communications entre services se font en format JSON, garantissant une interopérabilité et une traçabilité des échanges.

3. **Journalisation des erreurs** Chaque erreur serveur est capturée et enregistrée dans les logs, facilitant le suivi et le diagnostic sans exposer d'informations sensibles.

3.6 Mise en place d'un réseau privé virtuel pour la communication sécurisée

Dans le cadre de la mise en œuvre de notre solution, il a été essentiel de garantir un environnement de communication sécurisé entre les composants distribués du système. Pour cela, nous avons mis en place un **réseau privé virtuel (VPN)** permettant d'établir une connexion point-à-point entre le poste client distant (utilisateur demandeur) et le poste serveur hébergeant l'ensemble des services critiques (interface React, n8n, base de données).

3.6.1 Contexte et justification

Le VPN joue un rôle fondamental dans la sécurisation des échanges, en créant un canal chiffré encapsulé entre les deux machines à travers Internet, comme s'il s'agissait d'un réseau local. L'objectif est de simuler un environnement LAN sécurisé entre client et serveur, tout en contournant les limitations d'adresse IP publique .

L'outil utilisé pour établir ce VPN applique des mécanismes de chiffrement avancé (**chiffrement AES-256**) assurant confidentialité, intégrité et authentification dans toutes les communications. Une fois le réseau configuré, chaque machine obtient une adresse IP privée virtuelle permettant de lancer les services sur des ports accessibles uniquement aux membres autorisés du réseau VPN.

3.6.2 Objectifs du VPN dans l'architecture

- **Établir une communication sécurisée** entre une machine distante (client) et la machine locale (serveur).
- **Permettre l'accès distant à l'interface utilisateur React**, tout en gardant N8N, la base de données et les fichiers sensibles non exposés publiquement.
- **Assurer l'isolation du réseau** : seules les machines autorisées accèdent aux ressources internes.

3.6.3 Fonctionnement technique

- Chaque machine reçoit une adresse IP virtuelle privée via le VPN.
- La communication passe par un tunnel sécurisé utilisant un chiffrement AES-256.
- Le serveur agit comme nœud central, où tournent les services : React, n8n, backend et base de données.
- Le client est autorisé manuellement depuis une interface d'administration, ce qui renforce le contrôle d'accès.

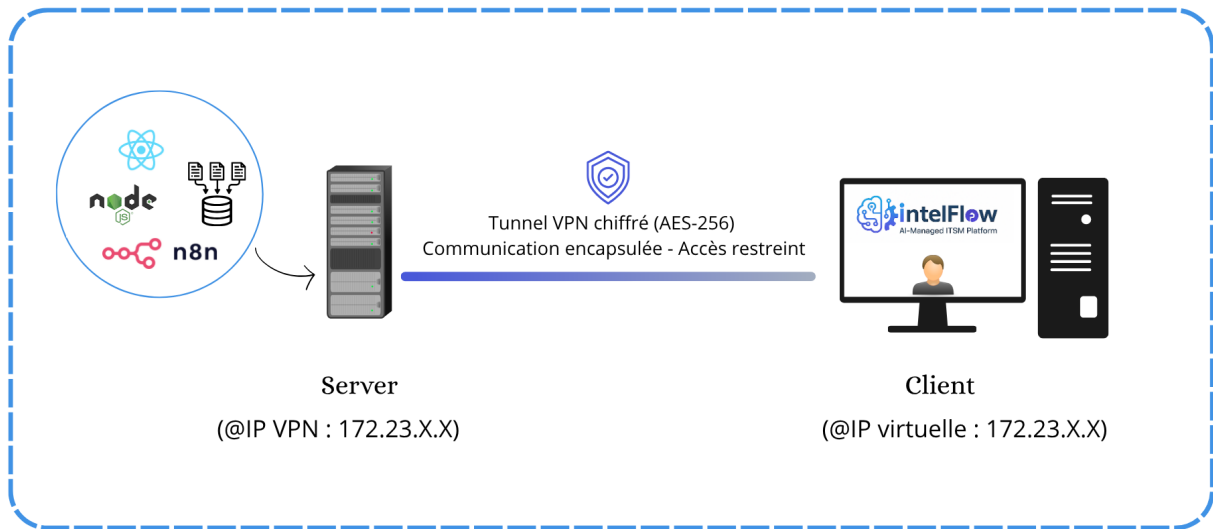


FIGURE 3.5 – Architecture sécurisée d'accès à la plateforme IntelFlow via VPN

3.6.4 Configuration réalisée

Afin de permettre la communication entre les différents composants de notre application via un VPN, nous avons utilisé ZeroTier, une solution de réseau virtuel simplifiée.

- **Création du réseau ZeroTier**

La capture ci-dessous affiche l'interface de gestion du réseau créé sur ZeroTier. Un identifiant unique (35c19XXXXXX) y est généré, permettant aux clients de rejoindre le réseau .

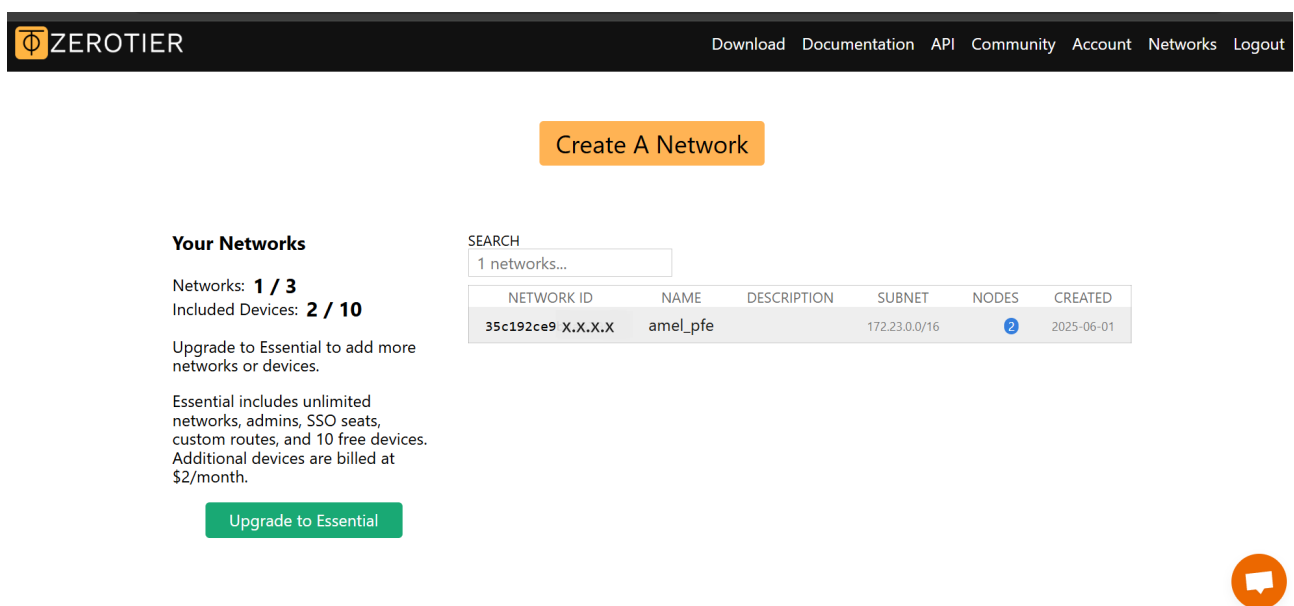


FIGURE 3.6 – Création et configuration du réseau ZeroTier

- **Ajout des membres au réseau**

La capture suivante montre les deux membres connectés au réseau : ServerMoi et postClient, chacun ayant reçu une adresse IP virtuelle attribuée par ZeroTier. Ces adresses permettent la communication sécurisée entre les deux machines via le VPN.

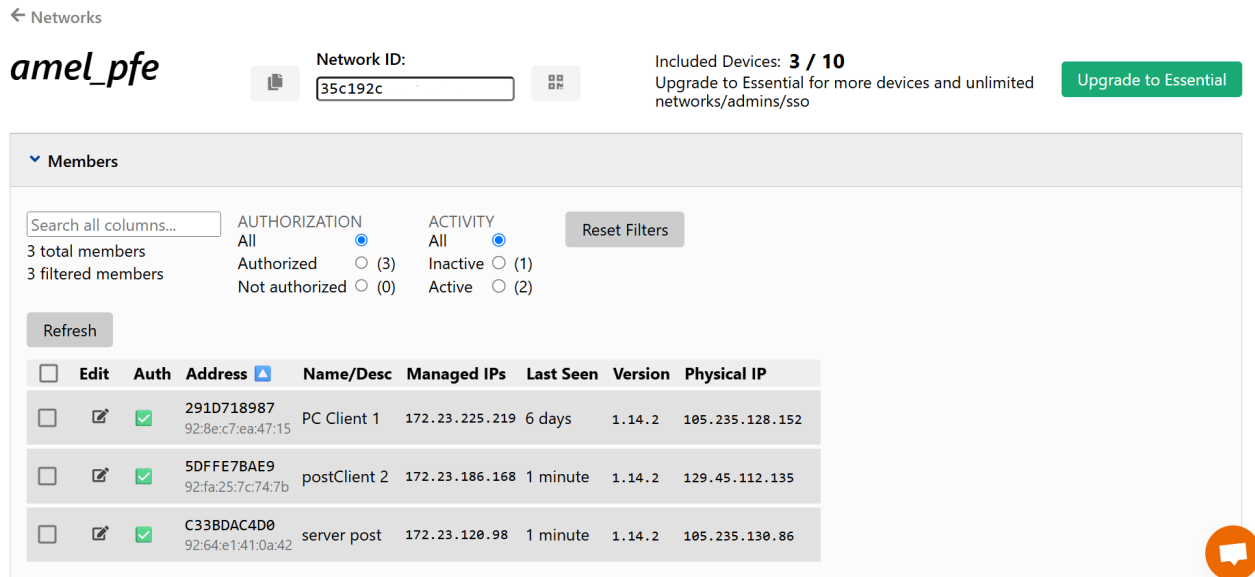


FIGURE 3.7 – Membres connectés et autorisés dans le réseau

3.7 Conclusion

Ce chapitre a présenté l'ensemble des technologies, des outils, et des mécanismes d'intégration technique déployés pour la mise en œuvre de notre solution. De la modélisation des workflows d'automatisation avec n8n à l'exploitation intelligente des données via les agents IA et RAG, chaque choix technologique a été justifié pour répondre aux besoins spécifiques du projet. La connexion entre l'interface utilisateur et la plateforme d'automatisation, la gestion intelligente des tickets, l'alimentation des bases vectorielles, la supervision en temps réel des équipements, l'automatisation d'alertes intelligentes en fonction de l'état du système et des délais contractuels, ainsi que la sécurisation des échanges et des accès, ont permis de concevoir un système fiable, cohérent et sécurisé. Cette architecture pose ainsi les fondations d'une solution évolutive et adaptée aux exigences professionnelles de gestion des services informatiques.

CHAPITRE 4

*Présentation visuelle de la solution réalisée
et résultats obtenus*

Chapitre 4

4.1 Introduction

Ce chapitre met en lumière la version finale de notre solution telle qu'elle a été concrètement mise en œuvre. Il expose la structure globale du système développé, son identité visuelle, son mode de fonctionnement, ainsi que les résultats obtenus à travers les différentes interfaces et services intégrés. L'ensemble repose sur une architecture cohérente, sécurisée et pensée pour répondre aux besoins d'une gestion moderne, automatisée et contrôlée des demandes et incidents.

4.2 Identité visuelle de la solution

Pour renforcer la cohérence globale du système développé, nous avons conçu une identité visuelle spécifique autour du nom IntelFlow, qui incarne à la fois l'intelligence (Intel) et la fluidité des processus métiers (Flow). Cette appellation reflète fidèlement les ambitions de la solution : allier automatisation intelligente, gestion des services IT structurée et expérience utilisateur fluide.

Interprétation du logo

Le logo ci-dessous illustre cette vision : Chaque élément visuel a été pensé pour traduire



FIGURE 4.1 – LOGO

les fondements techniques et fonctionnels du projet :

- **Le cerveau connecté**, à gauche, représente l'intégration de l'intelligence artificielle, avec des circuits imprimés rappelant les réseaux neuronaux des modèles LLM (Large Language Models). Il évoque également la capacité du système à comprendre, analyser et réagir intelligemment aux besoins des utilisateurs.

- **La roue dentée** intégrée dans le cerveau, symbolise l'automatisation des workflows (n8n), la gestion technique des demandes et l'orchestration des services selon une logique ITSM.
- **Slogan intégré :**
Le slogan "AI-Managed ITSM Platform" est positionné sous le nom. Il renforce l'identité fonctionnelle de la solution, en soulignant que la gestion des services IT repose ici sur une architecture intelligente, conforme aux principes d'ITIL4.
- **Choix des couleurs :**
Le dégradé bleu-violet renforce la perception :
de fiabilité (**bleu**)
d'innovation et d'intelligence numérique (**violet**).

4.3 Création de valeur ITSM à travers la solution "IntelFlow"

Conformément aux principes directeurs d'ITIL4, notre solution **IntelFlow** a été développée pour garantir une gestion des services performante, sécurisée et centrée sur la valeur. Grâce à une intégration fluide de composants intelligents et à une automatisation rigoureuse, **IntelFlow** assure une orchestration cohérente entre les acteurs, les flux de traitement et les données critiques du système. Cette synergie opérationnelle se traduit par :

L'automatisation de la création des tickets via un agent conversationnel IA
→ Standardisation des entrées, rapidité de traitement, meilleure expérience utilisateur.

L'attribution intelligente et instantanée des tickets à un technicien
→ Répartition efficace des ressources, réduction des délais, traitement ciblé.

L'accès à une base de connaissances vectorielle dynamique
→ Capitalisation sur les solutions passées, appui aux techniciens, amélioration continue.

La centralisation des échanges et la traçabilité complète des actions
→ Gouvernance renforcée, supervision claire, conformité aux bonnes pratiques ITSM.

À travers ces mécanismes, **IntelFlow** incarne les valeurs fondamentales d'ITIL4 :

- Orientation vers la valeur.
- Amélioration continue.
- Automatisation intelligente.
- Transparence des services.

4.4 Présentation fonctionnelle des interfaces

Avant d'accéder aux interfaces personnalisées, l'utilisateur passe par une page de connexion sécurisée. Il doit saisir son adresse email au format **nomprenom@entreprise.com** et son mot de passe personnel. Une fois l'authentification réussie, il sélectionne le rôle avec lequel il souhaite se connecter (**demandeur, technicien, superviseur ou administrateur**), en fonction de ses habilitations.

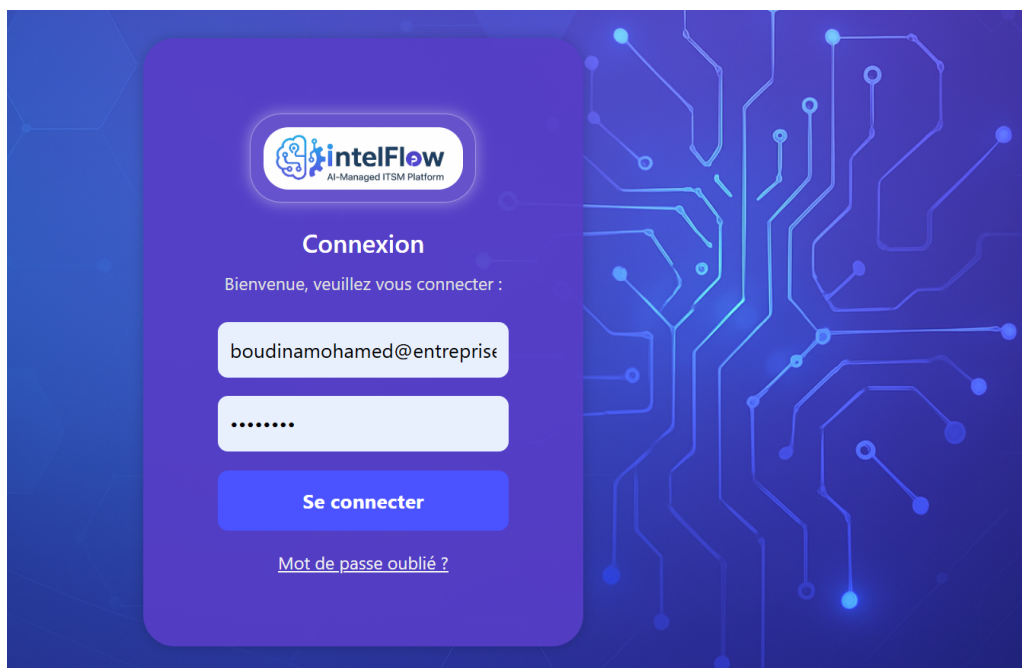


FIGURE 4.2 – Interface de connexion

Pour renforcer la sécurité de la plateforme tout en offrant une assistance efficace aux utilisateurs, nous avons conçu une interface dédiée à la réinitialisation du mot de passe. Celle-ci repose sur un processus en trois étapes :

- Saisie de l'adresse email, avec envoi automatique d'un code de vérification à usage unique .

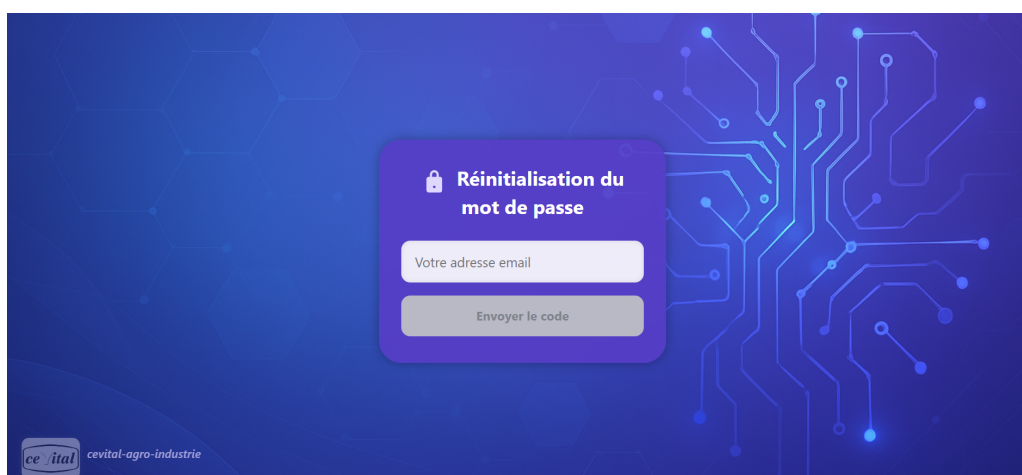


FIGURE 4.3 – Saisie de l'adresse mail

- Vérification du code par l'utilisateur (valide seulement 60 secondes), garantissant une authentification forte et temporaire.

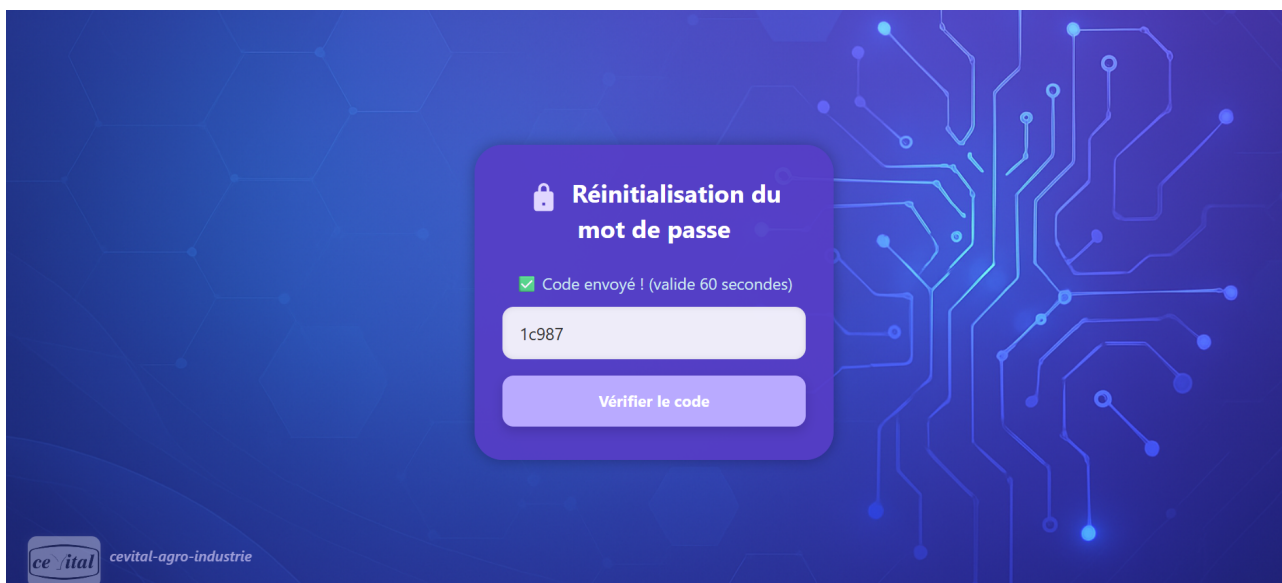


FIGURE 4.4 – Vérification du code

- Définition d'un nouveau mot de passe sécurisé, qui est immédiatement chiffré (haché) avant d'être enregistré dans la base.

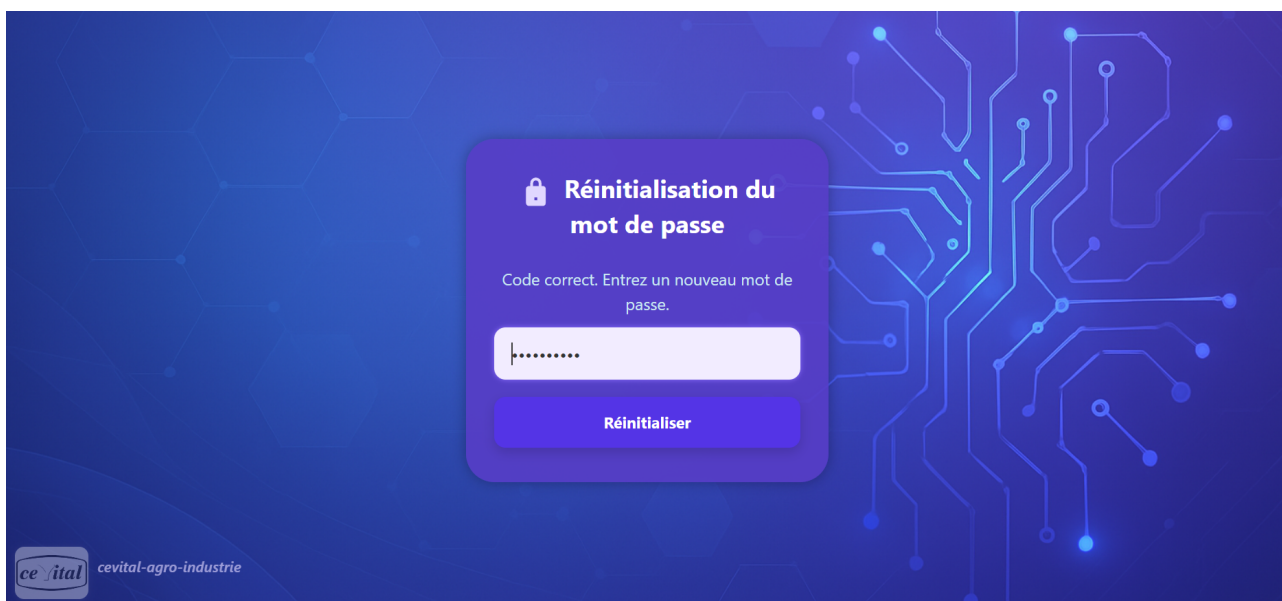


FIGURE 4.5 – Définition d'un nouveau mot de passe

Ce mécanisme permet une double validation (par email et par code temporaire), tout en restant simple d'utilisation. L'interface prend également en charge tous les rôles utilisateurs du système (demandeur, technicien, administrateur), garantissant ainsi une prise en charge complète et homogène.

La solution **IntelFlow** repose sur une architecture multi-profil, où chaque utilisateur accède à une interface adaptée à ses responsabilités dans le système. Ces profils principaux sont : demandeur, technicien, superviseur et administrateur. L'ergonomie de chaque espace a été pensée pour répondre à des besoins spécifiques tout en garantissant cohérence, fluidité d'usage et efficacité opérationnelle.



FIGURE 4.6 – Changement de rôle

4.4.1 Interfaces personnalisées selon le rôle

Le demandeur accède à un environnement centré sur la saisie intuitive des besoins via un chatbot IA, le suivi en temps réel de ses tickets, et des indicateurs personnels de traitement.



FIGURE 4.7 – Interface d'accueil de demandeur

Le technicien dispose d'un tableau de bord listant les tickets qui lui sont attribués automatiquement, avec possibilité de traitement direct et accès à des outils d'analyse.



FIGURE 4.8 – Interface d'accueil du technicien

Le superviseur bénéficie d'une vision étendue sur l'ensemble du système, incluant le suivi des techniciens et les tickets tous statuts confondus.

ID	Titre	Demandeur	Technicien	Type	Domaine	Statut	Urgence	Date Ouverture	Date Résolution	Solution
291	Problème d'accès VPN	Mohamed Boudina	Adel Cherif	Incident	Gestion des Accès	en cours	élevée	12/06/2025 11:26:43	-	-
290	Problème de connexion réseau RH	Mohamed Boudina	Mohamed Boudina	Incident	Infrastructure Réseau	en cours	moyenne	12/06/2025 11:22:08	-	-
289	Blocage ERP	Mohamed Boudina	Automatique (IA)	Incident	Applications Métier	clôturé	Élevée	11/06/2025 21:59:36	11/06/2025 21:59:52	1. Fermez complètem l'appli...
288	Ajout d'imprimante réseau	ryma adrar	Lina Mansouri	Demande	Infrastructure Impression	en cours	Faible	11/06/2025 21:20:39	-	-

FIGURE 4.9 – La liste des tickets

L'administrateur assure la gestion globale, incluant la création des comptes, l'attribution des rôles et la supervision consolidée des données du système.

Ajouter un utilisateur

Enregistrer
Annuler

FIGURE 4.10 – Ajout d'un utilisateur

- **Interface du catalogue de services**

L'interface du catalogue de services permet de centraliser et structurer les services proposés par l'organisation. Chaque service y est décrit à travers plusieurs champs essentiels tels que le nom, le domaine, la catégorie, le SLA, l'impact, ou encore l'urgence.

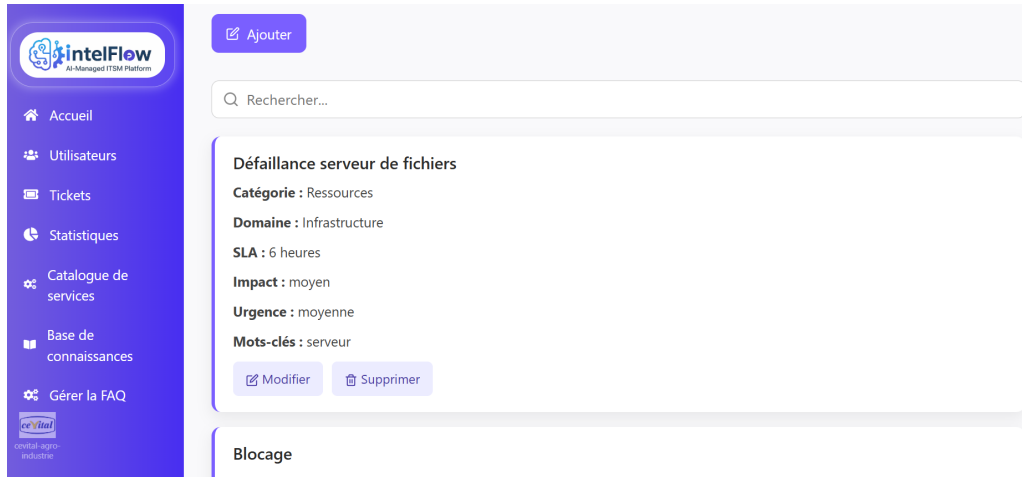


FIGURE 4.11 – Interface du catalogue de services

Sa gestion est assurée par le superviseur et l'administrateur, qui ont la possibilité d'ajouter, de modifier ou de supprimer les services selon les besoins. Le technicien, quant à lui, dispose uniquement d'un droit de consultation, lui permettant d'accéder aux informations sans les modifier.

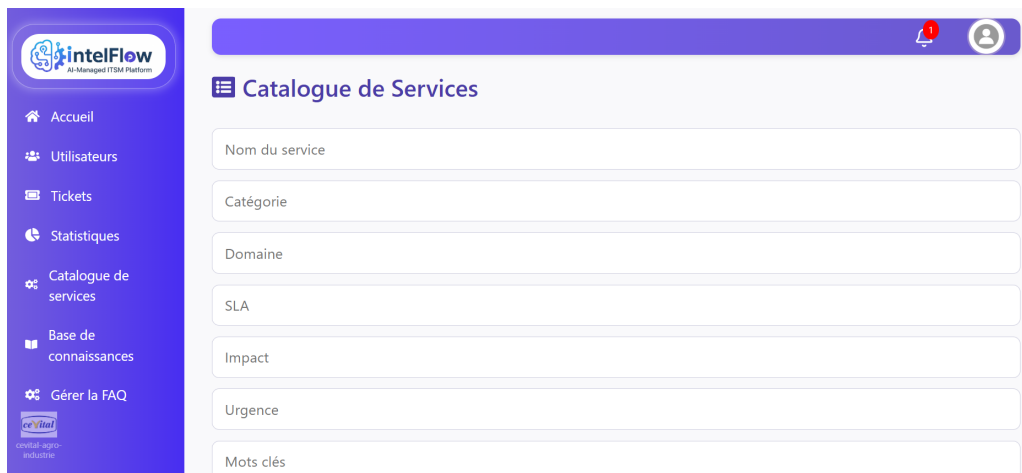


FIGURE 4.12 – la gestion de catalogue de services

Ce fonctionnement assure une meilleure lisibilité du catalogue, une mise à jour encadrée des données, ainsi qu'une répartition claire des responsabilités entre les utilisateurs du système.

- Interface de disponibilité des techniciens** Une interface dédiée a été conçue pour la gestion dynamique de la disponibilité des techniciens. Celle-ci repose sur un système de calendrier interactif, permettant à chaque technicien d'indiquer ses périodes d'indisponibilité, tout en considérant la disponibilité comme active par défaut.

The screenshot shows a form titled 'Ajouter' for adding availability. It includes a date field with a calendar icon, two time selection fields for 'Heure début' and 'Heure fin', and a checked checkbox labeled 'Je suis disponible'. A blue 'Ajouter' button is at the bottom.

FIGURE 4.13 – Calendrier de disponibilité

Cette gestion fine des disponibilités est automatiquement prise en compte par l'agent intelligent, qui évite d'assigner un ticket à un technicien indisponible. Cette approche garantit une allocation intelligente des ressources, favorise une réactivité accrue, et renforce la gestion proactive du support technique.

The screenshot shows the 'Mes disponibilités' interface. It features a table with the following data:

Date	Début	Fin	Statut	Raison	Action
2025-06-25	11:25:00	13:05:00	Indisponible	Réunion de travail	Supprimer
2025-06-25	14:30:00	15:30:00	Indisponible	maladie	Supprimer

Below the table is a blue button labeled 'Ajouter une disponibilité'.

FIGURE 4.14 – Interface de disponibilité des techniciens

- **Interface de supervision des équipements réseau**

Une interface web intuitive a été développée pour assurer une supervision centralisée des équipements du réseau. Elle permet au superviseur de gérer dynamiquement le parc machine (ajout, modification, suppression), avec une visualisation instantanée de l'état de connectivité grâce à l'utilisation du protocole ICMP via Ping.



FIGURE 4.15 – Interface des équipements côté superviseur

Le rôle du technicien est restreint à des opérations de vérification de la disponibilité réseau, sans autorisation de modification, garantissant ainsi l'intégrité du système et la séparation des privilèges.

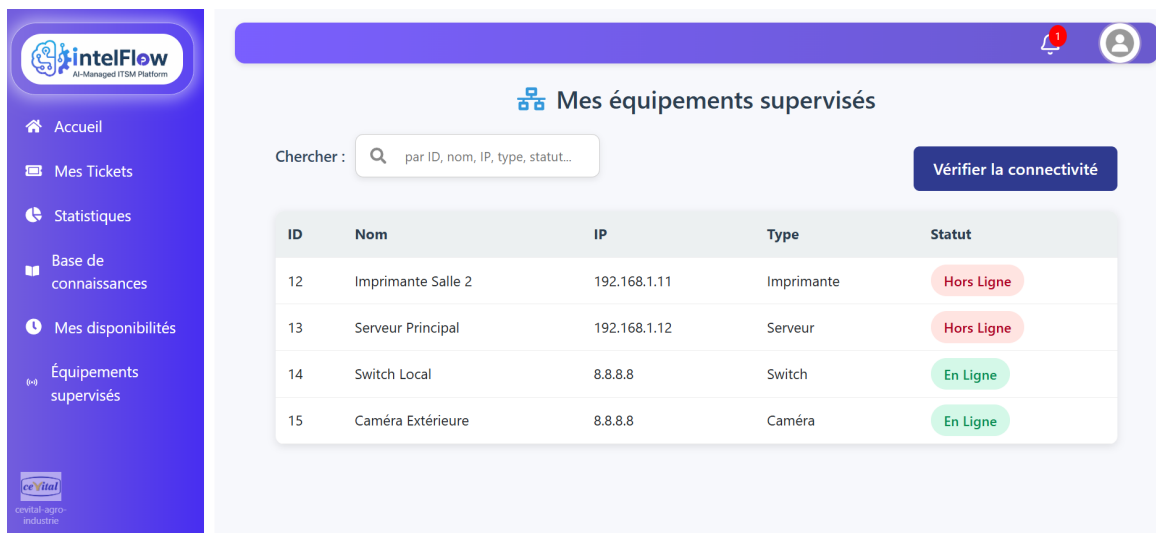


FIGURE 4.16 – Interface des équipements côté techniciens

Ce module reflète les standards d'un système de gestion proactive des infrastructures, intégrant des concepts de monitoring réseau, de haute disponibilité et de sécurité fonctionnelle.

4.4.2 Base de connaissances intégrée et FAQ

Le système intègre une base de connaissances évolutive, accessible à tous les profils :

1. Le demandeur peut la consulter pour accéder à des solutions ou FAQ



FIGURE 4.17 – Base de connaissance côté demandeur

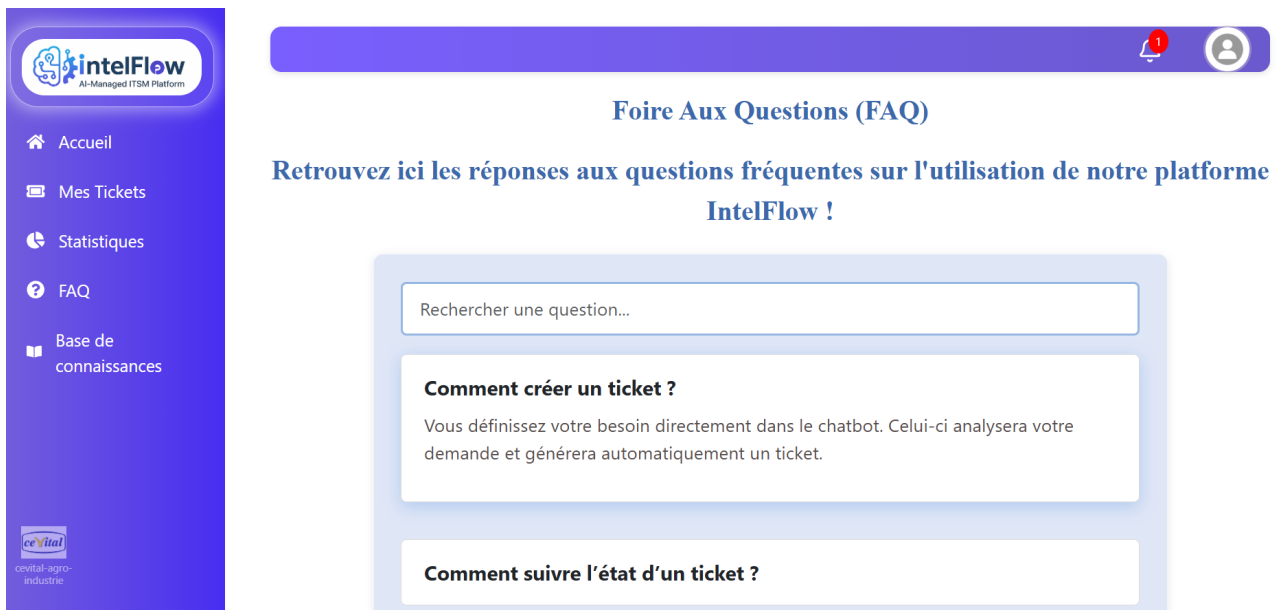


FIGURE 4.18 – Foire aux questions

2. Les techniciens, superviseurs et administrateurs peuvent faire des modifications dans la base , garantissant ainsi sa qualité et sa pertinence en continu.

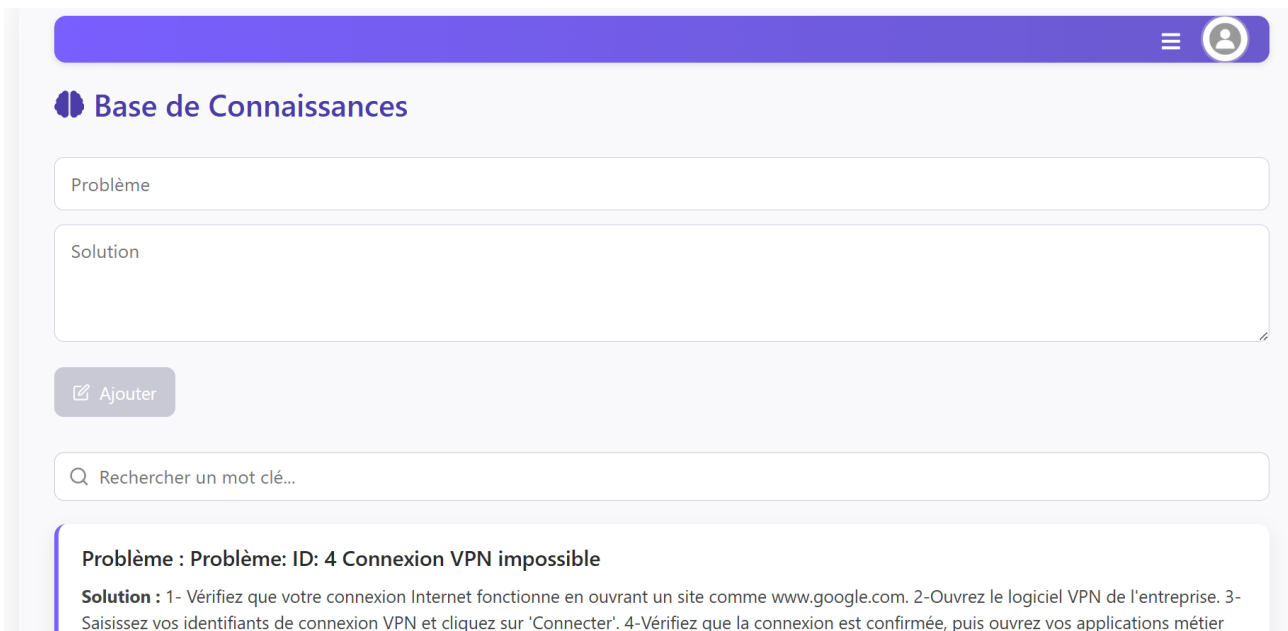


FIGURE 4.19 – Base de connaissance

4.4.3 Statistiques de performance

Chaque acteur bénéficie d'un accès à des statistiques adaptées à son périmètre :

1. Les demandeurs et techniciens disposent de statistiques personnalisées (**taux de résolution, temps de traitement, etc.**)

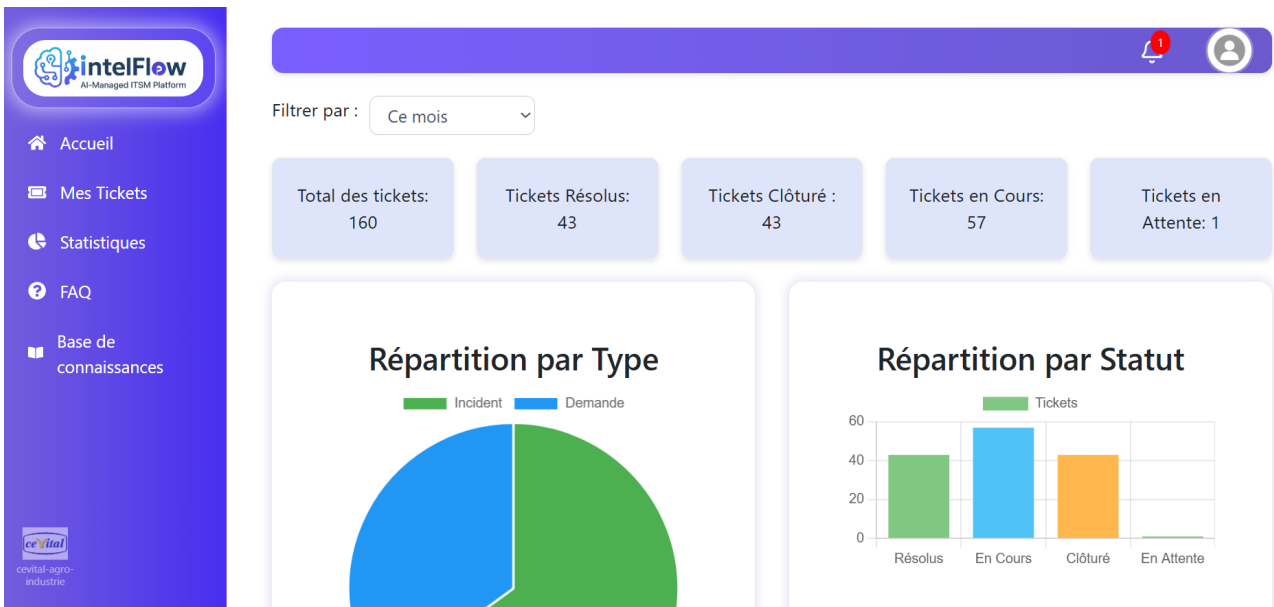


FIGURE 4.20 – Statistiques

- 2. Les superviseurs et administrateurs ont accès à des tableaux de bord globaux, avec indicateurs croisés sur l'ensemble des utilisateurs, des tickets et des performances SLA.

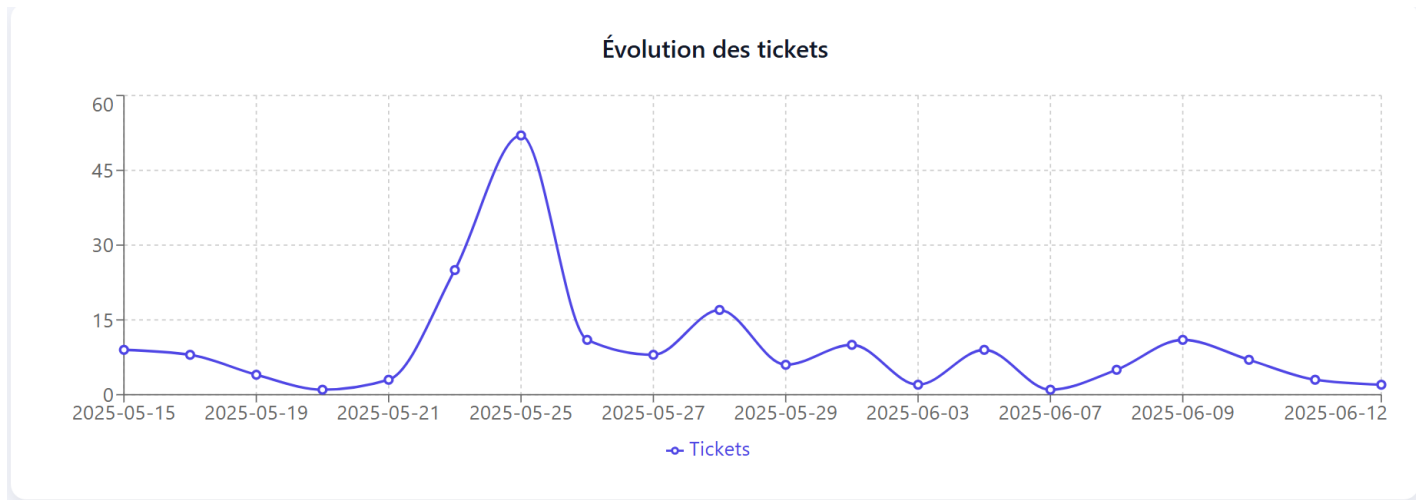
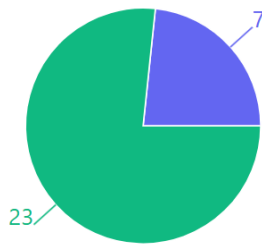


FIGURE 4.21 – Evolution des tickets

Répartition IA / Technicien



Incidents vs Demandes

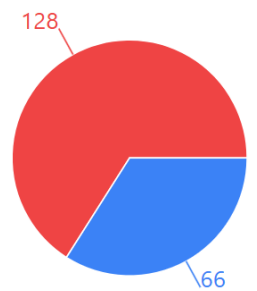


FIGURE 4.22 – Répartition des tickets par type et par mode de résolution

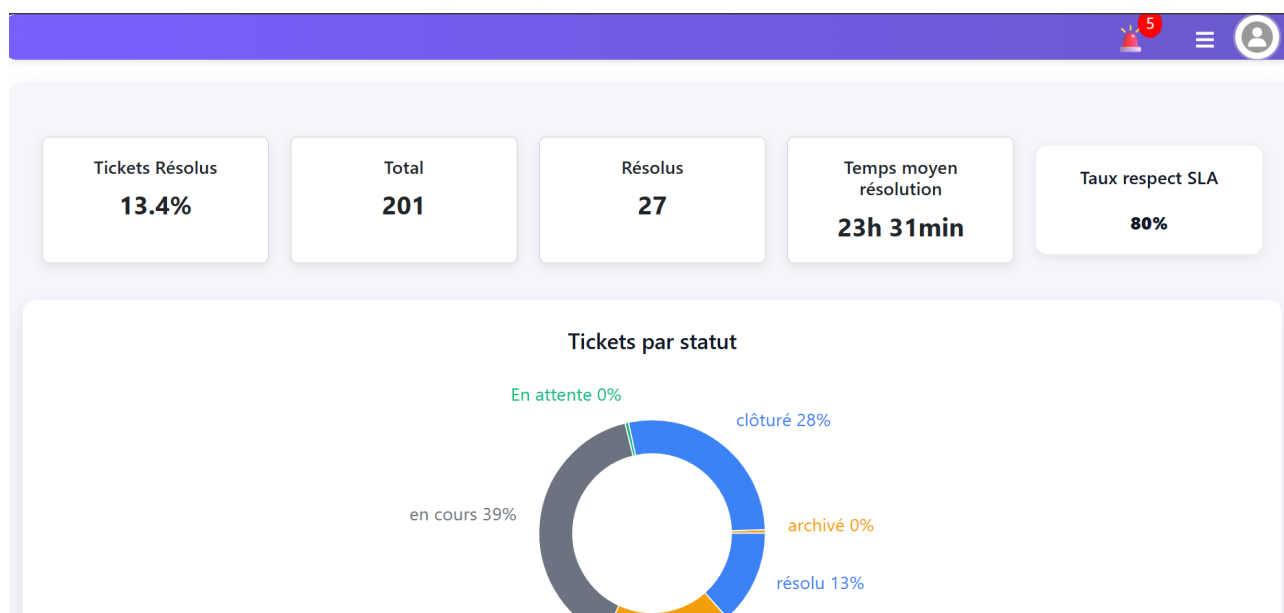


FIGURE 4.23 – Statistique côté superviseur

4.5 Conclusion

La solution IntelFlow, dans sa version finale, combine une architecture technique robuste, une identité visuelle cohérente et des interfaces personnalisées selon les profils métiers. Grâce à l'intégration intelligente des outils (**React**, **n8n**, **IA**, **VPN**), elle répond pleinement aux exigences modernes de gestion des services IT. Elle offre un environnement centralisé, sécurisé, automatisé et orienté valeur. Cette mise en œuvre opérationnelle constitue une étape clé vers la digitalisation intelligente des processus IT.

Conclusion générale

Ce mémoire avait pour objectif principal la conception et la réalisation d'une solution ITSM intelligente, conforme aux bonnes pratiques ITIL4, afin d'optimiser la gestion des demandes et incidents dans les systèmes d'information.

Pour y parvenir, nous avons adopté une démarche rigoureuse, allant de l'analyse des besoins à la mise en œuvre d'une solution opérationnelle, en intégrant des fonctionnalités avancées et en respectant les bonnes pratiques en vigueur.

La solution développée, **IntelFlow**, se distingue par son approche hybride combinant intelligence artificielle, supervision humaine et automatisation. Elle intègre notamment :

- Un chatbot conversationnel capable de créer des tickets à partir du langage naturel ;
- Une base de connaissances vectorielle pour proposer des solutions pertinentes ;
- Une affectation intelligente des tickets aux techniciens selon leur charge et disponibilité.

L'application offre également une interface riche et intuitive, adaptée à plusieurs profils :

- **Administrateur** : responsable de la gestion globale des utilisateurs, des droits d'accès, et de la configuration du système ;
- **Technicien** : traitement et suivi des tickets ;
- **Superviseur** : vision globale et affectation manuelle des cas spécifiques.

L'interface propose également un tableau de bord dynamique, affichant des statistiques en temps réel, notamment : le nombre des tickets et leur statut (en attente, en cours, résolu).

La sécurité a été soigneusement intégrée : authentification, séparation des rôles, chiffrement des mots de passe, et contrôle des accès.

Les résultats démontrent que la solution améliore la réactivité, réduit la charge des équipes support, et renforce la qualité de service, même pour des utilisateurs non techniques.

Ce travail ouvre enfin des perspectives intéressantes : extension vers d'autres processus ITSM, diagnostic du chatbot, ou enrichissement continu de la base de connaissances.

En résumé, nous avons atteint nos objectifs : proposer une solution moderne, intelligente, sécurisée et centrée utilisateur, pour une gestion IT plus fluide, plus efficace et plus accessible.

Résumé

Ce mémoire présente la conception et le développement d'une solution ITSM intelligente, conforme aux bonnes pratiques ITIL4, visant à optimiser la gestion des demandes et des incidents au sein des systèmes d'information.

Face aux limites des outils traditionnels et à la surcharge des équipes support, la solution proposée – **IntelFlow** – intègre un assistant virtuel (chatbot) capable de créer automatiquement des tickets, de suggérer des solutions pertinentes, et d'assigner les demandes aux techniciens les plus qualifiés.

Reposant sur une architecture intuitive, une base de connaissances vectorielle et une interface adaptée aux profils des utilisateurs, cette solution contribue à réduire la charge de travail, renforcer la réactivité, et améliorer la qualité des services IT.

Ce travail illustre les bénéfices d'une automatisation intelligente alignée avec ITIL4, tout en garantissant une accessibilité aux utilisateurs non techniques. Les résultats démontrent la pertinence de cette approche dans un environnement professionnel réel, et ouvrent des perspectives d'évolution, notamment l'intégration de nouveaux processus ITSM et l'usage du machine learning pour enrichir continuellement les capacités de l'assistant.

Abstract

This thesis presents the design and development of an intelligent ITSM solution, aligned with ITIL4 best practices, aimed at optimizing the management of requests and incidents within information systems.

In response to the limitations of traditional tools and the overload faced by support teams, the proposed solution – **IntelFlow** – integrates a virtual assistant (chatbot) capable of automatically creating tickets, suggesting relevant solutions, and assigning requests to the most qualified technicians.

Based on an intuitive architecture, a vector-based knowledge base, and a user-adapted interface, this solution helps reduce workload, enhance responsiveness, and improve the quality of IT services.

This work highlights the benefits of intelligent automation aligned with ITIL4, while ensuring accessibility for non-technical users. The results demonstrate the relevance of this approach in a real professional environment and open up prospects for further development, notably the integration of new ITSM processes and the use of machine learning to continuously enhance the assistant's capabilities.

Bibliographie

- [1] ACTEUR. *Acteur en UML*. Consulté en avril 2025. URL : [https://fr.wikipedia.org/wiki/Acteur_\(UML\)](https://fr.wikipedia.org/wiki/Acteur_(UML)).
- [2] ATLIASSIAN. *Comprendre les services informatiques et leur importance*. Consulté en 2025. 2023. URL : <https://www.atlassian.com/fr/itsm/it-service-management/it-services>.
- [3] ATLIASSIAN. *Service Request Management*. Consulté en juin 2025. 2025. URL : <https://www.atlassian.com/fr/itsm/service-request-management>.
- [4] AXELOS. *le Service Value System (SVS) et les dimensions d'ITIL4*. Consulté en 2025. 2023. URL : <https://www.axelos.com/resource-hub/blog/from-v3-to-4-this-is-the-new-til>.
- [5] CAS D'UTILISATION. *Diagramme de cas d'utilisation en UML*. Consulté en mai 2025. URL : <https://www.lucidchart.com/pages/fr/diagramme-de-cas-dutilisation-uml>.
- [6] *Diagramme de classe*. Consulté en mai 2025. URL : https://fr.wikipedia.org/wiki/Diagramme_de_classes.
- [7] DIGICOM CORPORATION. *Les 7 principes directeurs d'ITIL v4*. Consulté en juin 2025. 2025. URL : <https://digicomcorporation.com/fr/actualites/itil/les-7-principes-directeurs-ditil-v4/>.
- [8] *Intelligence artificielle*. Consulté en 2025. URL : https://apmg-international.com/article/ai-itsm-overview?utm_source=chatgpt.com.
- [9] MACERTIF. *34 pratiques ITIL 4 pour mieux gérer votre service IT*. Consulté en juin 2025. 2025. URL : <https://management-digital.macertif.com/ITSM/PratiquesITIL/>.
- [10] Canvas Business MODEL. *A Brief History of n8n*. Consulté en juin 2025. Oct. 2024. URL : <https://canvasbusinessmodel.com/fr/blogs/brief-history/n8n-brief-history>.
- [11] MSI NORD. *Découvrez les différents types de services informatiques disponibles pour soutenir votre entreprise*. Consulté en juin 2025. 2024. URL : <https://www.msi-nord.fr/actualites/decouvrez-les-differents-types-de-services-informatiques-disponibles-pour-soutenir-votre-entreprise/>.
- [12] n8n. Consulté en juin 2025. URL : <https://datascientest.com/n8n-tout-savoir#:~:text=n8n%20est%20une%20plateforme%20d,l'interaction%20d' applications..>
- [13] QRP INTERNATIONAL. *ITIL, c'est quoi? Définition et explication*. Consulté en juin 2025. 2025. URL : <https://www.qrpinternational.fr/blog/gestion-des-services-informatiques/itil-cest-quoi-definition-til/>.

- [14] TRIANZ. *Stratégie de gestion des services IT*. Consulté en juin 2025. 2025. URL : <https://www.trianz.com/fr/infrastructure/it-service-management-strategy>.
- [15] *zerotier*. Consulté en juin 2025. URL : <https://sparwan.com/blogs/news/zerotier-la-puissance-et-la-securite-du-vpn?srsltid=AfmB0oqrdTdxmtHSBctZ4HyJFI>