

RAPPORT DE STAGE DE TRONC COMMUN

allaw

Allaw

Travail avec l'équipe backend, et création d'un serveur de notification

Germain Stang - Co-CEO Allaw

Présenté par :
Legendre-Lemaire Malo

Promo 2026

0 Table des matières

Table des matières

1	Introduction	1
2	Présentation de l'entreprise	2
2.1	Le secteur d'activité	2
2.1.1	La concurrence	2
2.2	L'entreprise	2
2.2.1	L'équipe	2
2.2.2	Présentation	3
2.3	Le service	3
2.4	Le positionnement du stage dans les travaux de l'entreprise	3
3	Travail effectué	4
3.1	Le cahier des charges (C.D.C)	4
3.1.1	But général	4
3.1.2	Explication détaillée des résultats à obtenir	4
3.2	Compte-rendu d'activités	6
3.2.1	Axes d'étude et de recherche choisis	6
3.2.2	Déroulement concret des études, expérimentations, mises au point...	7
3.3	Développement, interprétation et critique des résultats	9
3.3.1	Bilan des choix technologiques et architecturaux	9
4	Conclusion générale	11
5	Glossaire	12
A	Annexes	13
A.1	Landing page d'Allaw	13
A.2	Connexion Google et LinkedIn	13
A.3	Notifications sur Allaw Pro	14
A.4	Recherche de professionnels	14

1 Introduction

Allaw est une jeune startup nantaise créée il y a un an par un père et son fils, dont le but est de révolutionner le monde du droit avec une ambition simple : créer le Doctolib du droit.

En février 2024, grâce à des contacts communs, j'ai eu l'opportunité de rencontrer Ludovic Stang, CEO d'Allaw. Cet échange a rapidement débouché sur une journée d'immersion au sein de l'entreprise. Cette expérience m'a permis de découvrir concrètement les missions qui m'attendraient durant mon stage. J'ai également rencontré une partie de l'équipe, qui était alors plus réduite qu'aujourd'hui. Mon intérêt pour cette entreprise vient de plusieurs points :

- Grâce à ma maîtrise des technologies utilisées par l'entreprise, telles que Node.js, TypeScript et MongoDB, je peux m'intégrer rapidement sans m'inquiéter des aspects techniques. Cela me permet de me concentrer pleinement sur la découverte du travail en entreprise.
- Le fait que l'entreprise soit petite (10 personnes), jeune mais en pleine expansion me permet de découvrir le travail en équipe de façon plus immersive et d'avoir un impact significatif sur le développement du produit. Enfin, cela permet d'appréhender une entreprise auprès de tous ses services, par exemple j'interagis souvent avec l'équipe "sales", ainsi que d'être en relation directe avec le management.
- L'entreprise évolue dans un secteur en pleine transformation numérique, ce qui offre de nombreuses opportunités d'innovation et d'apprentissage.
- La mission d'Allaw, qui vise à simplifier la prise de rendez-vous pour les particuliers, à démocratiser l'accès au droit et à simplifier le quotidien des professionnels juridiques, correspond à mes valeurs personnelles.
- L'ambiance de travail jeune et dynamique, ainsi que la culture d'entreprise orientée vers l'innovation et le partage des connaissances, correspondent parfaitement à mes attentes pour une première expérience professionnelle.
- Enfin, le fait que l'entreprise soit située à Nantes était un vrai avantage pour moi étant donné que je suis nantais.

Je suis donc au cœur de la dynamique d'une création d'entreprise où je découvre toutes les étapes importantes pour son expansion (embauche, levées de fonds, etc.). La parfaite intégration dans l'entreprise me permet de contribuer à la création de ses fondations techniques et humaines.

Je ne regrette pas d'avoir intégré cette jeune entreprise dans laquelle je me suis rapidement senti intégré, y compris sur le plan humain. En effet, les stratégies mises en place pour la cohésion des équipes sont nombreuses (des sorties, des jeux, etc.).

2 Présentation de l'entreprise

2.1 Le secteur d'activité

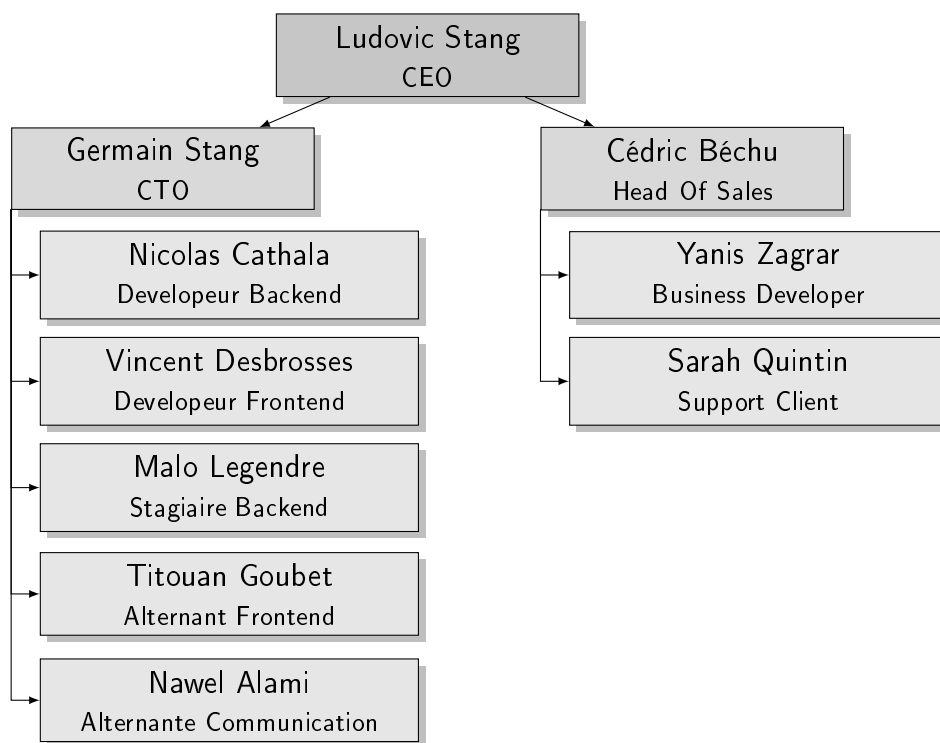
Le secteur d'activité d'Allaw se situe à l'intersection du numérique et du droit, dans le domaine de la "Legal Tech". Allaw propose son service aux notaires, aux avocats et aux commissaires de justice. Ce secteur est en pleine expansion, porté par la transformation numérique des professions juridiques. Le marché français des services juridiques représente un potentiel important, composé de plus de 70 000 avocats, 6 000 notaires et 3 000 commissaires de justice.

2.1.1 La concurrence

Allaw opère dans un secteur où plusieurs concurrents sont présents, chacun couvrant une partie de ce domaine d'activité. Cependant, Allaw se distingue par sa capacité unique à offrir une gestion complète des prises de rendez-vous. Contrairement à d'autres qui se concentrent exclusivement sur la mise en relation entre professionnels et clients ou sur la gestion administrative, Allaw ambitionne d'intégrer ces deux aspects. Cette intégration permet non seulement de faciliter la gestion des interactions, mais aussi d'améliorer l'expérience utilisateur en offrant une interface harmonieuse et cohérente.

2.2 L'entreprise

2.2.1 L'équipe



2.2.2 Présentation

Allaw est une startup nantaise fondée en 2023 par un duo père-fils, Ludovic Stang et son fils Germain. L'entreprise compte actuellement une dizaine de collaborateurs et connaît une phase de croissance active. Sa mission principale est de simplifier la mise en relation entre les professionnels du droit et leurs clients, en s'inspirant du modèle de Doctolib dans le secteur médical. Doctolib disposait d'une solide force de vente, un modèle qu'Allaw s'efforce d'imiter.

2.3 Le service

Le service développé par Allaw permet :

- Aux particuliers de prendre facilement rendez-vous avec des professionnels du droit
- Aux professionnels juridiques de gérer efficacement leur agenda et leur clientèle
- D'automatiser certaines tâches administratives pour optimiser le temps des professionnels
- De faciliter la communication entre les clients et les professionnels du droit

Elle parvient à cela à travers les fonctionnalités suivantes :

- Un référencement de tous les professionnels en France, y compris ceux non inscrits sur Allaw.
- Les agendas Outlook et Google directement synchronisés sur Allaw permettant de proposer aux particuliers des créneaux de disponibilités.
- Les visioconférences directement prises sur Allaw.
- Le paiement des prestations directement sur Allaw.
- Et bien d'autres...

Voir la page d'accueil d'Allaw dans l'annexe A.1.

2.4 Le positionnement du stage dans les travaux de l'entreprise

En tant que stagiaire au sein de l'équipe backend, ma mission s'inscrit dans le développement technique de la plateforme. Je travaille principalement sur :

- Le développement de nouvelles fonctionnalités pour le serveur
- La création d'un système de notifications
- L'amélioration des performances et de la scalabilité de l'application

Ce stage contribue directement à l'évolution du produit dans une phase critique de croissance de l'entreprise.

L'objectif est de construire un produit robuste capable de supporter une augmentation significative du nombre d'utilisateurs tout en maintenant la stabilité requise au bon fonctionnement. Cela implique également de collaborer étroitement avec les équipes de développement frontend pour assurer une intégration fluide des nouvelles fonctionnalités.

3 Travail effectué

3.1 Le cahier des charges (C.D.C)

3.1.1 But général

Au sein d'Allaw, mon but principal était d'aider au développement du backend et donc d'aider Nicolas, l'unique développeur backend, dans ses tâches. Mon sujet principal pour le stage était un serveur de notifications, recevant les événements du backend et les redistribuant vers les clients connectés.

Mais j'ai aidé au développement de beaucoup d'autres fonctionnalités :

1. Relier tous les professionnels inscrits et non-inscrits de manière aléatoire sur leurs pages pour aider le référencement Google. (le "maillage")
2. La connexion des usagers passant par Google ou LinkedIn. Voir annexe A.2.
3. La recherche des professionnels sur le site, paginée et aléatoire. Voir annexe A.4.
4. Les notifications grâce au serveur de notifications. Voir annexe A.3.
5. Régler la gestion des décalages horaires sur Allaw permettant d'ouvrir l'application aux outre-mer.
6. Une barre de recherche permettant de rechercher parmi tous les objets de manière naturelle.

À noter que pour chacune de ces fonctionnalités, le développement sur le **Frontend** a été fait par d'autres collègues, je me focalisais sur le **Backend**.

Je vais concentrer ce rapport sur le serveur de notifications, comme c'était mon sujet de stage, ainsi que les décalages horaires et enfin la barre de recherche.

3.1.2 Explication détaillée des résultats à obtenir

Le "maillage"

À mon arrivée, l'équipe était en pleine refonte du site, pour améliorer leur référencement Google. Dans ces travaux, se trouvait le besoin d'afficher en dessous de tous les professionnels, une sélection de 10 autres professionnels se trouvant dans la même ville. Et ceci de manière statique, pour que les professionnels suggérés ne changent pas afin d'aider Google à référencer les pages.

Ce "maillage" devait être suffisamment interconnecté pour permettre à tous les professionnels d'être trouvés par Google en passant par un autre professionnel.

Le serveur de notification

Le serveur de notifications devait recevoir des événements du backend, comme un nouveau rendez-vous, un paiement effectué, etc. Puis d'une manière ou d'une autre les renvoyer à tous les clients en temps réel sur leur frontend.

À part ces conditions, j'étais complètement libre sur l'implémentation et les choix de technologies.

Allaw ne disposant pas de DevOps à temps plein, il était difficile pour Nicolas, le Backend, de trouver le temps de se former et de créer les services difficiles à mettre en place tels que Kafka pour la communication inter-services. Pour cette raison, le serveur de notifications n'aurait qu'une seule instance pour toute l'application, ce qui augmente les charges sur un seul serveur, contrairement au Backend qui possède trois instances. Il fallait donc un service performant et robuste.

Ce serveur de notifications permettra aux professionnels gardant Allaw ouvert dans la journée de savoir en temps réel quand un rendez-vous a été pris sans avoir à actualiser sans cesse l'application. Voir annexe A.3.

Les décalages horaires

Durant la phase de MVP d'Allaw, un support robuste des différents décalages horaires sur la plateforme n'avait pas été réfléchi. En effet, les différentes bibliothèques de l'écosystème JavaScript gèrent de manière différente les décalages horaires, et en essayant de tout accommoder, les dates étaient souvent sauvegardées et transformées dans des formats non-standards.

Mon but était donc d'uniformiser cette gestion, permettant à quelqu'un d'outre-mer de prendre rendez-vous avec un avocat se trouvant en métropole et que les bonnes dates s'affichent.

Cette gestion permettra à Allaw de commencer à débaucher des professionnels dans les DOM-TOM directement, ce qui étend largement leur secteur d'activité. De plus, les professionnels du droit en DOM-TOM ont souvent des difficultés dans la mise en relation avec leurs clients, Allaw est donc parfaitement positionné à ce niveau-là.

La barre de recherche global

La base de données utilisée par Allaw est MongoDB. Or, les capacités de recherche textuelle de cette base de données sont très simplistes et pas très performantes. Il fallait donc mettre en place Elasticsearch qui offre des capacités de recherche textuelle bien supérieures.

En plus de ce nouveau service, la recherche devait permettre de trouver des rendez-vous grâce à une recherche naturelle. Par exemple, entrer "rendez-vous avec Malo Legendre le 5 janvier 2025" devrait retourner tous les rendez-vous avec "Malo Legendre" qui ont eu lieu le 5 janvier 2025.

Une autre fonctionnalité était de proposer de créer un rendez-vous, une note ou autre objet si ceux-ci n'existent pas déjà.

3.2 Compte-rendu d'activités

3.2.1 Axes d'étude et de recherche choisis

Mon stage portait sur le développement d'un serveur de notifications performant et robuste. L'objectif principal était de fournir aux utilisateurs des informations **en temps réel**, ce qui a imposé des contraintes spécifiques sur le choix des technologies et l'architecture du système. Plusieurs axes d'étude ont été définis pour atteindre cet objectif.

Le premier enjeu était le choix des technologies. J'étais libre de choisir, mais je devais considérer la **dette technique** potentielle. Utiliser des technologies peu maîtrisées par l'équipe aurait compliqué la maintenance. Il fallait trouver un équilibre entre **innovation** et **familiarité**.

De plus, la **communication** entre le Backend, le Frontend et mon service devait être simple. L'utilisation de technologies similaires aux services existants permettrait de réutiliser les mêmes interfaces TypeScript et ainsi réduire les risques d'incompatibilité.

Un autre axe important était de déterminer le **protocole de communication** entre les clients et le serveur de notifications. La nécessité d'une communication **en temps réel** a immédiatement réduit le champ des possibles. Les requêtes HTTP traditionnelles, basées sur un modèle requête-réponse, ne sont pas adaptées à ce besoin car elles obligeraient les clients à interroger le serveur périodiquement (polling) pour vérifier la présence de nouvelles notifications. Cette approche est inefficace en termes de consommation de ressources (bande passante, CPU) et introduit une latence importante entre l'émission d'une notification et sa réception par le client.

Pour garantir une communication en temps réel, il était nécessaire de se tourner vers des technologies permettant une communication **bidirectionnelle** ou **unidirectionnelle persistante** entre le serveur de notification et les clients. Deux technologies principales se sont dégagées : **WebSockets** et **Server-Sent Events (SSE)**.

- **WebSockets** établit une connexion bidirectionnelle et persistante entre le client et le serveur. Cela permet au serveur de "pousser" des notifications vers le client dès qu'elles sont disponibles, sans que le client n'ait à les demander.
- **SSE**, quant à lui, permet une communication unidirectionnelle du serveur vers le client via une connexion HTTP persistante. Le serveur peut ainsi envoyer des données au client en temps réel.

Le choix entre ces deux technologies devait se faire selon la facilité d'implémentation, la performance et la compatibilité avec l'architecture existante, tout en gardant à l'esprit le besoin fondamental de **temps réel**. C'est-à-dire que quand un usager prend rendez-vous sur le site, il faut qu'**instantanément** le professionnel reçoive une notification pour le prévenir de la prise de rendez-vous.

Enfin, le dernier axe de recherche était de concevoir une **architecture générique** pour le serveur de notifications. Le serveur devait être capable de transmettre n'importe quel type de notification **sans en interpréter le contenu**. Cela posait la question de la structuration des données et de leur gestion efficace. Comment garantir la flexibilité du système sans connaître la structure interne des notifications? Permettre au serveur de

notifications de lire les notifications poserait le problème de devoir lui permettre aussi de lire la base de données principale, et donc partager la structure de celle-ci, augmentant la difficulté de changement.

3.2.2 Déroulement concret des études, expérimentations, mises au point...

Les technologies que j'ai finalement utilisées sont :

- **Fastify**, à l'opposé de **Express** utilisé pour le Backend. Ces technologies gèrent les requêtes HTTP. La différence entre mon service et le Backend ne pose pas de problèmes majeurs. **Fastify** est plus rapide qu'**Express**, ce qui est crucial pour un serveur de notification. De plus, il offre une meilleure validation des données, améliorant la robustesse du système.
- **MongoDB** comme base de données, la même que le Backend. Cela simplifie le déploiement et la gestion des données.

Pour la connexion des clients, j'ai choisi les **Server-Sent Events (SSE)** plutôt que les **WebSockets**. Ce choix a été motivé par :

- **Simplicité** : Les SSE sont nativement supportés par les navigateurs et simples à implémenter côté serveur.
- **Adaptation au besoin unidirectionnel** : Les notifications vont du serveur vers les clients, les SSE sont donc parfaitement adaptés.
- **Reconnexion automatique** : En cas de perte de connexion, les navigateurs se reconnectent automatiquement au flux SSE.

Transparence des notifications

Le serveur de notification a été pensé depuis le début pour être complètement indépendant des données sauvegardées à l'intérieur des notifications.

Cela a plusieurs avantages :

- **Découplage fort** : Évolutions indépendantes des services, pas de dépendance au format des notifications.
- **Sécurité renforcée** : Accès limité aux données sensibles, réduit les risques.
- **Flexibilité et évolutivité** : Gère tout type de notification, ajout facile de nouveaux types.
- **Maintenance simplifiée** : Logique de traitement déportée, focus sur la transmission.
- **Réutilisabilité** : Composant générique, utilisable dans d'autres projets.
- **Indépendance de la base de données principale** : Pas d'accès, pas de dépendance à sa structure.
- **Simplicité** : Code plus clair et facile à comprendre, focus sur l'essentiel.

Pour cela la plus grande difficulté vient de savoir quelle notification transmettre à quel professionnel. Comme le serveur de notification n'a pas accès aux données de la notifications il ne peut pas savoir à qui cette notification est destinée.

Pour résoudre ce problème, j'ai choisi un système peu conventionnel :

Quand un professionnel se connecte à Allaw, le Front va faire une requête au Backend, et celui-ci va lui donner un **Token**. Ce token sera spécialement créé par le Backend pour contenir un 'filtre', ce filtre décrit quelles notifications il est autorisé à recevoir. C'est avec ce token qu'il pourra se connecter directement au serveur de notifications, qui pourra, grâce à ce filtre dont il ne connaît pas la logique pour le créer, savoir quelles notifications seront renvoyées.

Ce token est constitué d'une **expression** en JSON similaire aux **query** MongoDB, ce qui permet une grande flexibilité. Ces expressions que j'ai surnommées des expressions **Alang** sont un véritable langage de programmation permettant d'exprimer de nombreuses fonctionnalités complètement dans ce token.

Marquage comme lu des notifications

Pour pouvoir savoir quelles notifications ont été vues par le professionnel, un système similaire de token a été mis en place, permettant au front de marquer comme vue n'importe quelle notification directement sur le serveur de notifications.

Ces tokens contiennent une liste d'actions qui peuvent être prises, que le front peut choisir, suivie d'une liste de notifications sur lesquelles effectuer l'action.

Gestion des données

Le Backend fait une requête authentifiée au serveur de notification pour publier une nouvelle notification, quand par exemple un nouveau rendez-vous a été pris.

Le serveur de notifications fait passer par deux systèmes toutes les notifications reçues de cette manière :

- Un dont le but est de transférer les notifications vers toutes les connexions actives.
- Un autre qui sauvegarde la notification dans la base de données pour pouvoir la renvoyer aux pros qui n'étaient pas en ligne lors de la prise de rendez-vous.

Les notifications sont vérifiées envers tous les filtres de tous les professionnels connectés actuellement, et sont envoyées vers ceux qui correspondent. Puis elle est enregistrée dans la base de données, avec des données supplémentaires aux données de la notification, comme la date d'envoi. Cette enregistrement permet de renvoyer la notifications à la connexion d'un professionnel si jamais il ne l'avait pas vu avant.

3.3 Développement, interprétation et critique des résultats

Le développement du serveur de notifications a été une expérience riche en enseignements, me confrontant à la complexité de la conception d'un système temps réel performant et robuste.

3.3.1 Bilan des choix technologiques et architecturaux

Les choix technologiques et architecturaux effectués ont permis d'atteindre les objectifs initiaux : fournir aux utilisateurs des informations en temps réel de manière transparente, sans que le serveur n'ait à interpréter le contenu des notifications.

— **Technologies utilisées :**

- **Fastify :** Gestion des requêtes HTTP. Sa rapidité et sa robustesse, en comparaison avec Express, ont contribué à la performance du serveur. La validation des données intégrée a renforcé la fiabilité du système.
- **MongoDB :** Base de données NoSQL. Son utilisation, en cohérence avec le reste de l'infrastructure, a simplifié le déploiement et la gestion des données.
- **Server-Sent Events (SSE) :** Protocole de communication unidirectionnelle. Ce choix s'est avéré judicieux pour la diffusion efficace des notifications et la reconnexion automatique en cas de perte de connexion.

— **Architecture :**

- **Système de filtres (expressions Alang) :** Intégré dans des tokens générés par le Backend, il a permis une distribution ciblée des notifications sans exposer la **logique métier** au serveur.
 - **Flexibilité et performance :** Gestion d'une grande variété de scénarios de notification.
 - **Simplicité pour le serveur :** Pas d'interprétation du contenu des notifications.
- **Marquage des notifications comme lues :** Utilisation de tokens pour une gestion efficace de l'état des notifications côté client, tout en préservant l'indépendance du serveur.

Remise en question : *From scratch* vs. Solutions Cloud

Le développement *from scratch* a été formateur mais a représenté un investissement conséquent. La question se pose : **une solution cloud existante n'aurait-elle pas été plus judicieuse ?**

Analyse des solutions Cloud alternatives

De nombreux services managés de gestion de notifications existent :

- **Exemples :** Firebase Cloud Messaging (FCM), Amazon SNS, Azure Notification Hubs.

- **Fonctionnalités** : Infrastructures scalables, gestion de gros volumes, différents protocoles (WebSockets, SSE...), fonctionnalités avancées (segments d'utilisateurs, personnalisation, analyses), haute disponibilité et sécurité.

Avantages potentiels d'une solution Cloud

- **Concentration sur le métier** : Se focalise sur les aspects fonctionnels.
- **Temps de développement réduit** : Mise sur le marché plus rapide.
- **Coûts potentiellement réduits** : Facturation à l'usage, avantageux au démarrage.
- **Gestion de l'infrastructure déléguée** : Sécurité, scalabilité et haute disponibilité gérées par le fournisseur.

Avantages du développement *from scratch*

- **Personnalisation totale** : Adaptation précise aux besoins.
- **Maîtrise complète** : Contrôle de l'architecture et du code.
- **Flexibilité accrue** : Ajout de fonctionnalités et optimisation facilitées.
- **Expertise interne** : Développement de compétences précieuses et différenciantes.
- **Indépendance technologique** : Pas de verrouillage fournisseur.

Inconvénients du développement *from scratch* et dette technique

- **Maintenance et évolution** : Dépendent de l'expertise de l'équipe, risque en cas de départ.
- **Responsabilité accrue** : Charge de travail pour la sécurité, la scalabilité et la haute disponibilité.
- **Coût de développement élevé** : Investissement en temps et en ressources humaines conséquent. Le calcul du coût par rapport à une solution cloud n'est pas certain d'être avantageux.

Conclusion : Un choix stratégique à pondérer

Le choix dépend des priorités du projet, des ressources et de la stratégie à long terme.

- **Solution Cloud** : Gain de temps, réduction de la dette technique, concentration sur le métier.
- *From scratch* : Maîtrise totale, personnalisation, développement de compétences.

4 Conclusion générale

Le stage que j'ai effectué chez Allaw m'a permis de mettre en œuvre les compétences acquises durant ma formation et de les appliquer à des projets concrets. Les objectifs principaux de mon stage, tels que définis dans mon cahier des charges, ont été atteints avec succès.

J'ai notamment travaillé sur le développement d'un serveur de notifications performant et robuste, en utilisant des technologies telles que Fastify et SSE. Ce projet m'a permis d'approfondir mes connaissances en matière de développement backend et de communication en temps réel. J'ai également contribué à d'autres projets, tels que la mise en place du maillage pour améliorer le référencement Google et la gestion des décalages horaires, ce qui m'a permis de diversifier mes compétences et de découvrir de nouvelles problématiques.

Ce stage chez Allaw, bien plus qu'une simple expérience professionnelle, a été une véritable aventure humaine et formatrice. J'ai eu la chance de plonger au cœur d'une jeune startup en pleine croissance, où chaque membre de l'équipe, par sa passion et son engagement, contribue à façonner le futur de l'entreprise.

J'ai particulièrement apprécié l'ambiance de travail dynamique et collaborative. L'équipe, soudée et bienveillante, a su m'intégrer rapidement et me faire sentir comme un membre à part entière. J'ai pu échanger avec des professionnels passionnés, apprendre de leurs expériences et bénéficier de leurs conseils.

Sur le plan professionnel, j'ai pu mettre en pratique les connaissances acquises durant ma formation dans un environnement stimulant. J'ai également découvert les défis et les réalités du monde de l'entreprise, notamment en matière de gestion de projet, de communication et de résolution de problèmes.

Mais au-delà des aspects professionnels, c'est surtout sur le plan humain que ce stage a été enrichissant. J'ai gagné en confiance en moi, en autonomie et en capacité à travailler en équipe.

Je suis reconnaissant envers toute l'équipe d'Allaw pour m'avoir accueilli et accompagné durant ces semaines. J'ai grandi tant sur le plan professionnel que personnel, et je suis convaincu que cette expérience me sera précieuse pour la suite de mon parcours.

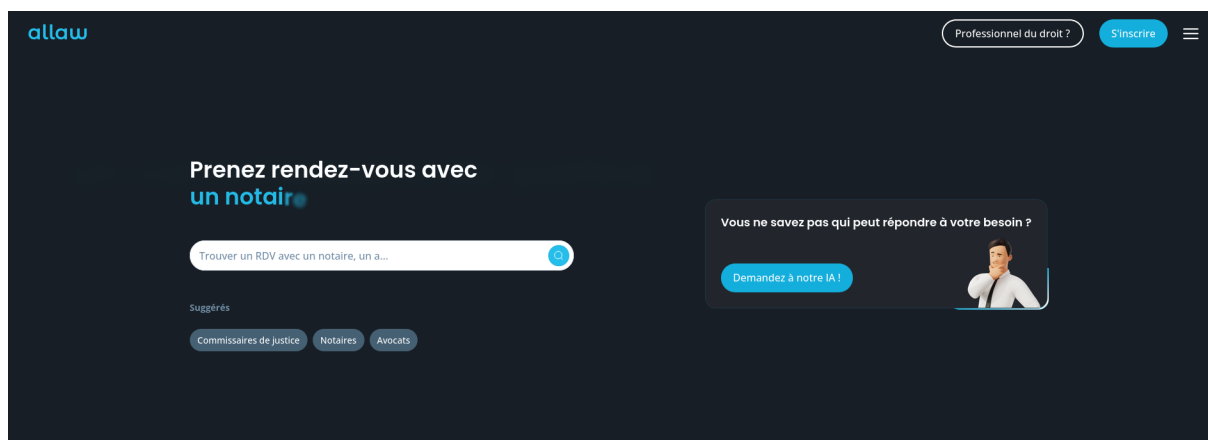
Je suis heureux d'avoir fait mes premiers pas dans le monde professionnel au sein d'une structure à taille humaine comme Allaw. J'ai pu y découvrir une culture d'entreprise forte, axée sur l'innovation, le partage et l'humain. Je suis convaincu que ces valeurs me seront utiles tout au long de ma carrière.

5 Glossaire

- **Alang** : Langage d'expressions JSON développé pour le serveur de notifications d'Allaw, permettant de filtrer les notifications à transmettre aux utilisateurs.
- **Backend** : Partie d'une application qui gère le traitement des données et la logique métier côté serveur.
- **Business Developer** : Professionnel chargé du développement commercial et de la croissance de l'entreprise.
- **Dette technique** : Conséquence de l'adoption de solutions techniques rapides mais imparfaites, nécessitant un travail supplémentaire futur.
- **DevOps** : Pratiques et outils combinant le développement logiciel (Dev) et l'administration système (Ops).
- **Doctolib** : Plateforme de prise de rendez-vous médicaux en ligne, servant de modèle à Allaw pour le secteur juridique.
- **Elasticsearch** : Moteur de recherche distribué, utilisé pour améliorer les capacités de recherche textuelle.
- **Express** : Framework web pour Node.js utilisé pour créer des applications web et des API.
- **Fastify** : Framework web Node.js alternatif à Express, optimisé pour la performance.
- **Frontend** : Partie d'une application visible et interactive pour l'utilisateur dans le navigateur.
- **JavaScript** : Langage de programmation utilisé principalement pour le développement web.
- **JSON** : Format de données textuelles utilisé pour l'échange d'informations.
- **Kafka** : Plateforme de traitement de flux de données en temps réel.
- **Legal Tech** : Technologies appliquées au domaine juridique pour moderniser et optimiser les services légaux.
- **Logique métier** : Règles et processus spécifiques au fonctionnement d'une entreprise ou d'un secteur d'activité.
- **MongoDB** : Base de données NoSQL orientée documents.
- **MVP (Minimum Viable Product)** : Version minimale d'un produit permettant de tester sa viabilité sur le marché.
- **Node.js** : Environnement d'exécution JavaScript côté serveur.
- **NoSQL** : Type de base de données non relationnelle, alternative aux bases de données SQL traditionnelles.
- **Polling** : Technique consistant à interroger régulièrement un serveur pour obtenir des mises à jour.
- **Scalabilité** : Capacité d'un système à s'adapter à une charge croissante.
- **Server-Sent Events (SSE)** : Technologie permettant à un serveur d'envoyer des mises à jour à un client en temps réel.
- **Service Support** : Équipe chargée de l'assistance aux utilisateurs.
- **Token** : Jeton d'authentification ou d'autorisation permettant l'accès sécurisé à des ressources.
- **TypeScript** : Extension de JavaScript ajoutant le typage statique.
- **WebSocket** : Protocole de communication bidirectionnelle en temps réel entre un navigateur et un serveur.

A Annexes

A.1 Landing page d'Allaw



Simplifiez vos démarches juridiques avec Allaw

-  Un accès unique à plus de 90 000 professionnels du droit.
-  Prise de rendez-vous en quelques clics.

A.2 Connexion Google et LinkedIn

allaw

Rejoignez Allaw

Créez votre compte gratuitement et entrez en relation avec des centaines de professionnels.

 S'inscrire avec Google

 S'inscrire avec LinkedIn

OU

Adresse Email*

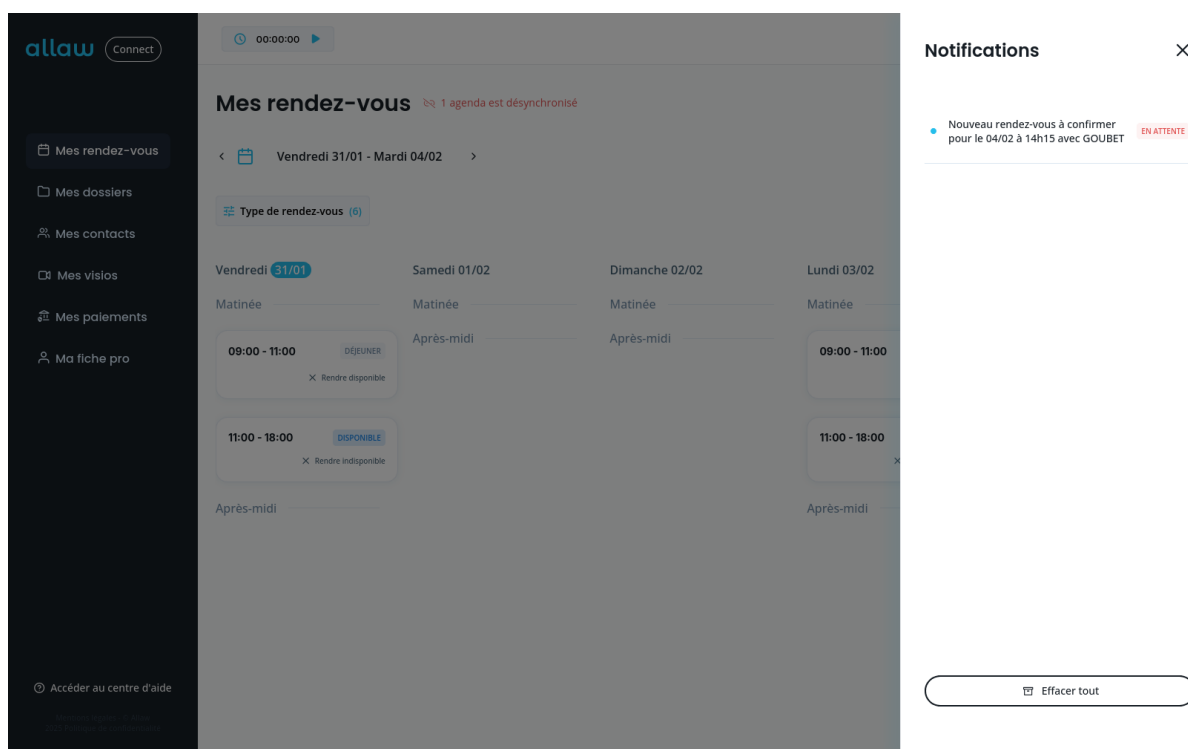
Entrez votre adresse email

Continuer →

Vous avez déjà un compte ? [Se connecter](#)

Dans le cadre de votre inscription, ALLAW traite vos données afin de vous faire bénéficier de ses services (création de compte, prise de rendez-vous avec des professionnels du droit...). Nous avons mis en place des garanties de sécurité élevées afin d'assurer la confidentialité de vos échanges. Conformément au RGPD, vous disposez d'un droit d'accès, de rectification, d'opposition, d'effacement, de portabilité et de limitation des traitements. Vous pouvez également retirer à tout moment votre consentement. Pour exercer vos droits, vous pouvez nous contacter par email à privacy@allaw.fr ou par courrier à ALLAW - Service DPD, 4 boulevard Gabriel, Guitteau 44000 Nantes. Envie d'en savoir plus ? Consultez notre [politique de confidentialité](#)

A.3 Notifications sur Allaw Pro



A.4 Recherche de professionnels

