DOSSIER DE COMPÉTENCES

Daniel Levy

Dossier de compétences · Développeur Node.js / React & IA générative

Ce dossier de compétences présente mon travail de développeur full-stack Node.js / React / TypeScript, spécialisé dans l'intégration de l'IA générative aux applications métier. Pour chaque feature Sairen : le contexte, l'objectif, la solution, la stack et mon périmètre exact.

Mis à jour le 10 novembre 2026

5 ans	5 features	5 fournisseurs LLM	Paris
de développement full-stack	livrées chez Sairen, de l'assistant IA au chat temps réel	derrière une même connexion	sur site ou remote · disponible immédiatement

Compétences

Front-end	React, Next.js, TypeScript
Back-end	Node.js, NestJS, TypeScript, API REST, WebSocket, webhooks, workers, BullMQ, Redis, PostgreSQL
Intégration IA	Assistant IA avec appel d'outils (tool calling), serveurs MCP, permissions sur les outils, connexion LLM multi-fournisseurs (OpenAI, Claude, Mistral, Gemini, DeepSeek), traitements LLM en chaîne, transcription (speech-to-text), validation humaine des réponses
Architecture & qualité	Architecture hexagonale (ports et adaptateurs), tests unitaires (Jest, Vitest)
Mobile	Swift, SwiftUI, Firebase, RevenueCat, Amplitude
Données	PostgreSQL, MySQL, RLS, Talend, Node-RED, Metabase, RGPD

DERNIÈRE MISSION · 03/2026 – 09/2026

Sairen : 5 features, de l'assistant IA au chat temps réel

Développeur Full-Stack IA · Paris · équipe de 4

Sairen édite un SaaS d'agents IA conversationnels, en production dans des centres médicaux et des groupes hôteliers. Son produit phare est un agent vocal qui répond au téléphone et prend rendez-vous. Beaucoup de clients gardent leur CRM ou leur ERP (SAP Business One, Sage), auxquels le produit doit se connecter.

FEATURE 1 · ULYSSE

Un assistant IA qui exécute les actions du produit

CONTEXTE

Le produit compte de nombreux écrans : conversations, statistiques, tâches, agents, workflows. Une partie des données utiles aux clients se trouve aussi dans leur ERP (SAP Business One, Sage).

OBJECTIF

Permettre à l'utilisateur de faire l'essentiel en conversant : consulter, agir, configurer, interroger son ERP. L'interface ne sert plus qu'à contrôler les agents, et l'assistant ne sort jamais des droits de l'utilisateur.

SOLUTION

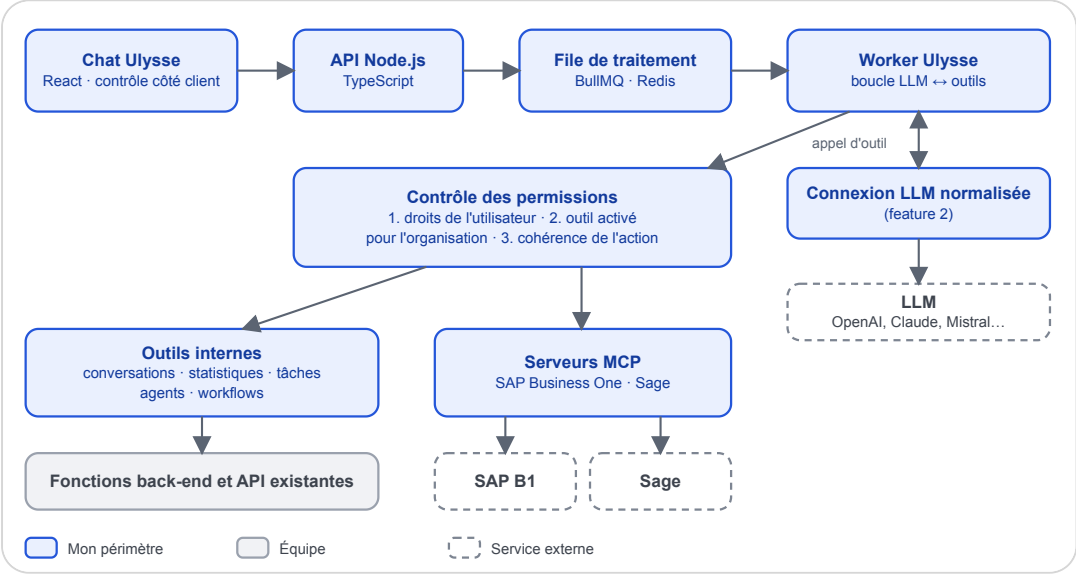
Un chatbot LLM : l'utilisateur écrit sa demande, l'assistant la traite et répond. Par exemple « combien d'appels cette semaine ? », « quelles tâches me restent ? », « crée un agent pour l'accueil ».

Des outils (tool calling) : chaque action est un outil qui appelle une fonction back-end ou une API existante. L'assistant passe par les mêmes services que le reste de l'application, jamais directement par les données.

Des serveurs MCP vers SAP Business One et Sage : l'utilisateur demande sa dernière commande ou son tableau de livraison, et l'assistant va chercher l'information dans l'ERP.

La sécurité sur trois niveaux, contrôlée côté back-end, et aussi côté client : l'assistant ne fait que ce que l'utilisateur a le droit de faire, un outil n'est disponible que s'il est activé pour l'organisation, et des contrôles de cohérence bloquent les actions impossibles.

Des traitements asynchrones : les demandes passent par une file BullMQ / Redis, traitée par des workers.



Stack	TypeScript, Node.js, React, LLM via la connexion normalisée, tool calling, MCP, BullMQ, Redis
Mon périmètre	Conçu et développé de bout en bout : le chatbot, la couche d'outils, les permissions, les serveurs MCP, les workers et une partie des outils. Les autres outils sont écrits par l'équipe selon une convention commune : qui crée une fonction back-end crée ensuite l'outil Ulysse correspondant.

FEATURE 2 · CONNEXION LLM

Une connexion LLM normalisée, multi-fournisseurs

CONTEXTE

Les clients n'ont pas tous les mêmes préférences : certains veulent Claude, d'autres ChatGPT, d'autres un modèle moins cher. Et chaque fournisseur a sa propre API, avec son propre format.

OBJECTIF

Choisir le modèle par usage, selon la qualité et le coût en tokens, sans toucher au code métier. Et permettre au client de brancher sa propre connexion LLM.

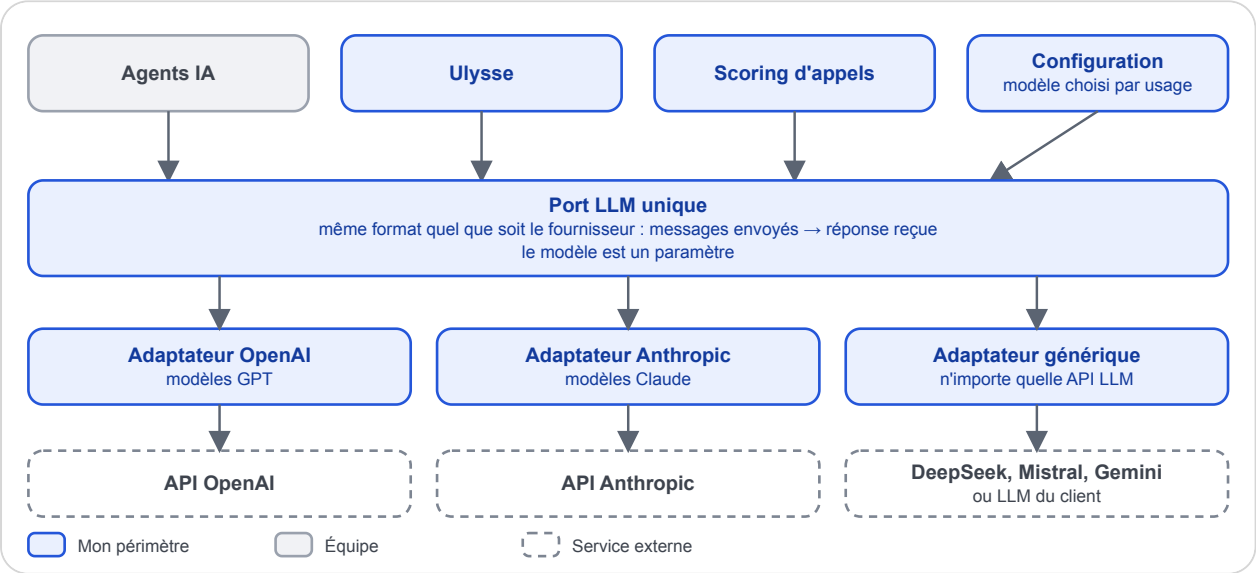
SOLUTION

Un port LLM unique : quel que soit le fournisseur, l'envoi d'un message et la réception de la réponse suivent toujours le même format.

Trois adaptateurs derrière ce port : OpenAI pour les modèles GPT, Anthropic pour les modèles Claude, et un adaptateur générique capable de se connecter à n'importe quelle API de LLM (DeepSeek, Mistral, Gemini ou la connexion du client).

Le modèle est un simple paramètre, choisi par usage : par exemple Claude Sonnet pour un agent, DeepSeek pour Ulysse afin de réduire le coût en tokens.

Les agents, Ulysse et le scoring d'appels passent tous par ce port : aucune brique ne dépend d'un fournisseur.



Stack	TypeScript, Node.js, ports et adaptateurs, API OpenAI, API Anthropic, adaptateur générique pour toute API de LLM
Mon périmètre	Conçu et développé de bout en bout : le port commun, le format unique d'échange, les adaptateurs OpenAI et Anthropic et l'adaptateur générique.

FEATURE 3 · CANAUX

Six canaux de messagerie branchés sur les agents IA

CONTEXTE

Les entreprises reçoivent des messages sur de nombreux canaux (réseaux sociaux, messageries, e-mail), chacun avec sa propre API. Elles doivent répondre vite, avec le bon ton et dans leurs horaires.

OBJECTIF

Normaliser la conversation, quel que soit le canal, pour y brancher les agents IA, tout en gardant le contrôle humain sur les réponses.

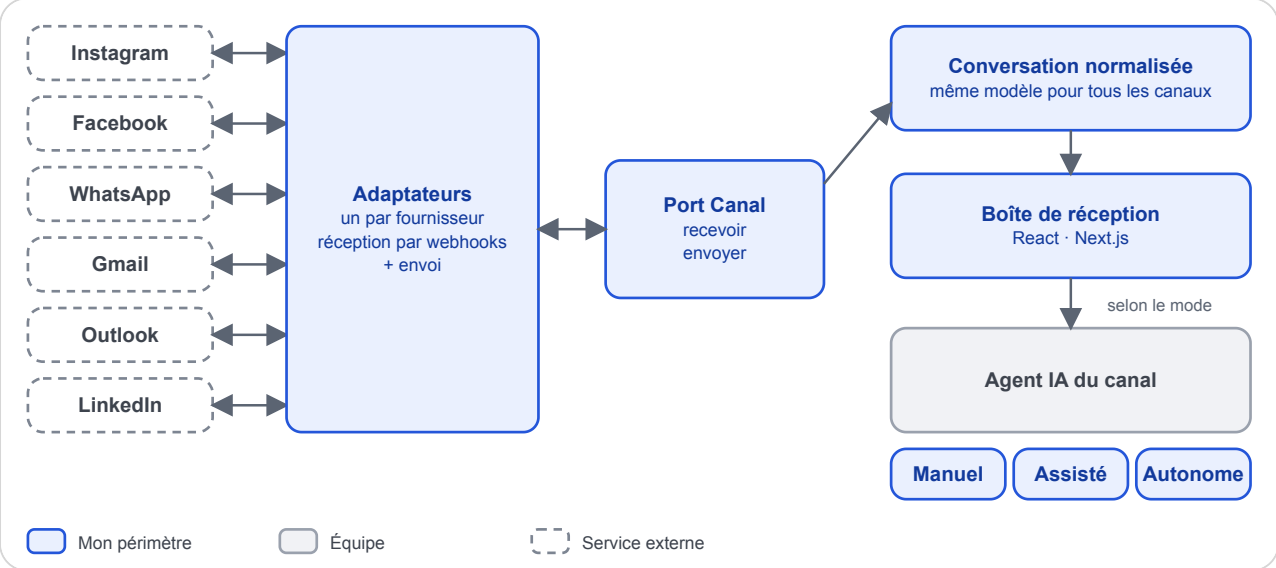
SOLUTION

Un port « canal » avec deux capacités, recevoir et envoyer un message, et un adaptateur par fournisseur : Instagram, Facebook et WhatsApp (Meta), Gmail, Outlook, LinkedIn.

Réception par webhooks (ou interrogation toutes les 5 secondes quand le fournisseur n'en propose pas), puis conversion dans un modèle de conversation unique, centralisé dans une boîte de réception.

Trois modes, réglables par conversation pour tester, puis sur tout le canal une fois l'agent éprouvé : manuel (un humain répond), assisté (l'agent prépare un brouillon que l'humain valide) et autonome (l'agent répond seul).

La réponse s'adapte au canal (un e-mail ne s'écrit pas comme un WhatsApp) et à l'entreprise : secteur, ton, horaires.



Stack	TypeScript, Node.js, React, Next.js, ports et adaptateurs, webhooks, API Meta (Instagram, Facebook, WhatsApp), Gmail, Outlook, LinkedIn
Mon périmètre	Toute la partie canaux, conçue et livrée de bout en bout, puis branchée sur les agents IA. Les agents eux-mêmes (modèle, contexte, persona, outils) ont été développés par un autre membre de l'équipe.

FEATURE 4 · ANALYSE D'APPELS

Analyser automatiquement des campagnes d'appels et relancer les clients

CONTEXTE

Un client grand compte de la relation client mène des campagnes d'appels pour un opérateur télécom. Une campagne compte par exemple une centaine d'enregistrements de 1 à 3 minutes, trop pour être analysés à la main.

OBJECTIF

Mesurer automatiquement la qualité d'une campagne (résolution, satisfaction, risque de départ, qualité de la prise en charge) et relancer les clients dont le problème n'est pas résolu.

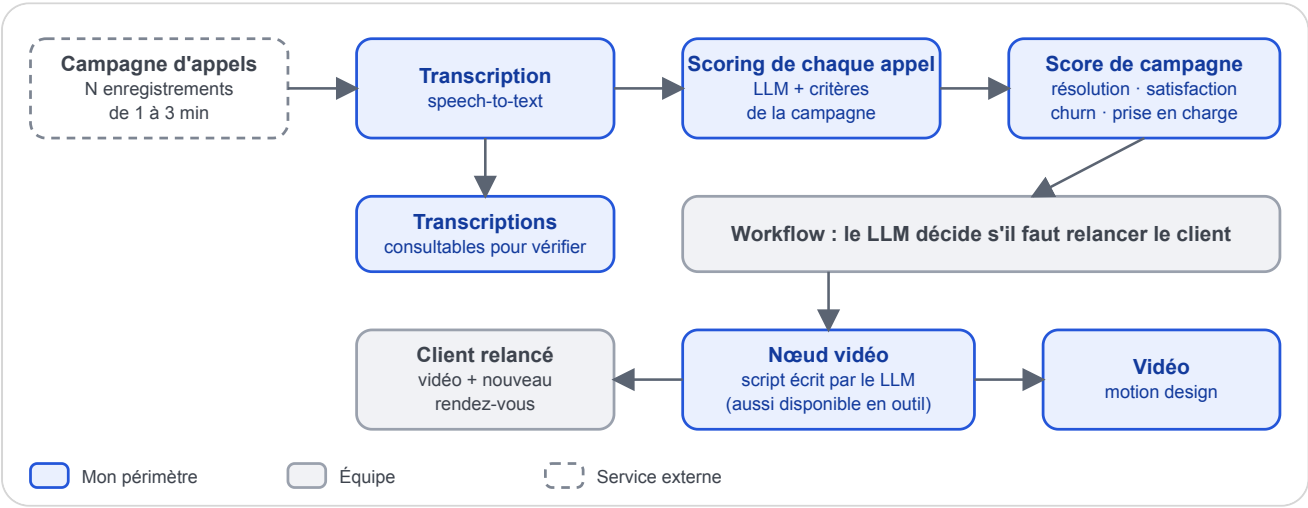
SOLUTION

Transcription de chaque enregistrement (speech-to-text). Les transcriptions restent consultables pour vérifier le scoring.

Scoring de chaque appel par un LLM selon des critères définis pour la campagne, puis score global de la campagne : sur 100 appels, combien de résolutions, de clients satisfaits, de clients à risque.

Ces étapes sont des nœuds d'un workflow : à partir du score, le LLM décide s'il faut relancer le client.

Relance par vidéo : le LLM écrit le script, une vidéo de motion design est générée automatiquement et envoyée au client avec la proposition d'un nouveau rendez-vous. La génération vidéo est aussi disponible comme outil.



Stack	TypeScript, Node.js, speech-to-text, LLM via la connexion normalisée, moteur de workflow, génération de vidéos de motion design
Mon périmètre	Le nœud d'analyse (transcription, scoring par appel, score de campagne) et le nœud vidéo avec l'outil associé. Le moteur de workflow et le service LLM qui décide des relances ont été réalisés par l'équipe.

FEATURE 5 · DESK

Le chat interne qui prend le relais quand l'IA s'arrête

CONTEXTE

Quand l'agent vocal ne peut pas résoudre un appel, il crée une tâche dans Desk, l'espace de Sairen qui centralise les tâches. Elles doivent ensuite être traitées par des humains, qui communiquaient jusque-là sur des outils externes.

OBJECTIF

Permettre aux équipes de se coordonner sur les tâches sans quitter Sairen : discuter, se transférer une tâche, la résoudre, être alerté quand on est sollicité.

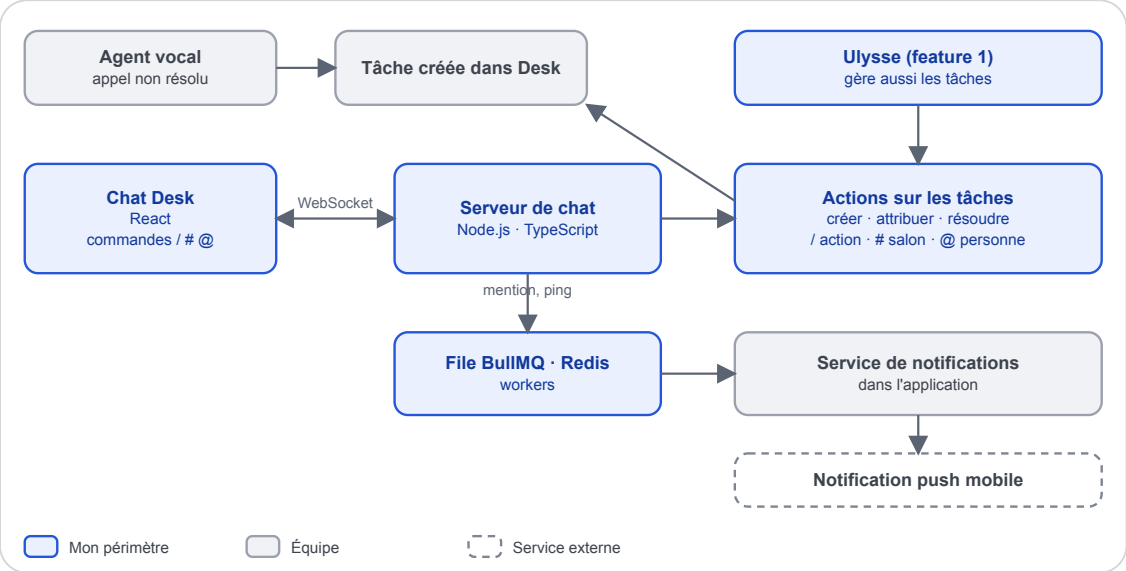
SOLUTION

Un chat interne sur le modèle de Slack, intégré à Desk, en temps réel par WebSocket entre l'application React et le serveur Node.js.

Des commandes dans le chat : « / » déclenche une action sur une tâche (la créer, l'attribuer, la résoudre), « # » mentionne un salon, « @ » mentionne une personne.

Les mentions et les relances passent par une file BullMQ / Redis traitée par des workers, qui alimentent le service de notifications : dans l'application et en push sur mobile.

Les mêmes actions sur les tâches sont aussi disponibles en conversant avec Ulysse.



Stack	TypeScript, Node.js, React, WebSocket, BullMQ, Redis, workers, notifications push
Mon périmètre	Le chat, conçu et développé de bout en bout, front et back : temps réel, commandes « / », « # » et « @ », actions sur les tâches, envoi des mentions vers les notifications. L'agent vocal, le modèle des tâches et le service de notifications ont été réalisés par l'équipe.

Les autres expériences

Staymakom

01/2026 – 03/2026

Développeur Full-Stack TypeScript · Réservation d'expériences : une nouvelle façon de penser ses vacances, l'expérience d'abord, la destination ensuite. Plateforme livrée en 3 mois.

- Reconstruit de zéro en React / Next.js une plateforme démarrée sur un prototype généré par IA, sans tests ni architecture : nouvelle base de données, dépôt git, migration complète.
- Recherche, disponibilités et parcours jusqu'au paiement en ligne avec Revolut, avec un test isolé du service de paiement pour diagnostiquer un incident en quelques minutes.
- Synchronisé automatiquement les disponibilités avec un PMS existant : si une synchronisation échoue, la réservation n'aboutit pas, pour ne jamais vendre une disponibilité fausse.
- Construit 3 espaces : client, administration (expériences, réservations, remboursements), hôtelier (inscription et visibilité de l'établissement).

TypeScript, Next.js, React, Node.js, API REST, PostgreSQL, Vitest

> **Mon périmètre** · Seul développeur, de la conception à la mise en production. Périmètre du MVP priorisé avec la fondatrice pour livrer en 3 mois.

Butter

02/2025 – 12/2025

Développeur iOS · Product Builder · Les meilleurs restaurants de Paris réunis dans une app.

- Lancé et publié l'app du MVP à la production : 14 000 utilisateurs, 4,9/5 sur l'App Store, sans budget publicitaire.
- Construit l'app en SwiftUI, avec un domaine métier isolé des écrans et de Firebase.
- Intégré Mapbox pour la carte, RevenueCat pour les abonnements (20 % de conversion premium) et Amplitude pour mesurer les usages.

Swift, SwiftUI, Firebase, RevenueCat, Amplitude, Mapbox

> **Mon périmètre** · Seul développeur : architecture, choix des services et développement de l'app. Firebase retenu pour livrer vite au stade MVP. Le produit était pensé à trois, avec les deux fondatrices.

Groupe SYG Énergie

05/2024 – 02/2025

Développeur Full-Stack React / Node.js · Un acteur du BTP et de l'énergie qui gère de nombreux chantiers, installateurs, commerciaux et clients, et voulait réunir tous ses process dans un seul outil.

- Livré un CRM métier (prospection, signatures, suivi de chantier) utilisé par plus de 200 personnes.
- Un parcours par rôle : direction, commerciaux, installateurs, secrétariat, clients.
- Un tableau de bord de pilotage : chantiers par étape, chiffre d'affaires du mois, coût des installateurs, performance des commerciaux, points de blocage.

TypeScript, JavaScript, React, Node.js, NestJS, SQL, Jest

> **Mon périmètre** · Projet d'équipe. À moi : le parcours installateur et une partie du tableau de bord de pilotage.

Groupe Socoda

03/2023 – 06/2024

Développeur Full-Stack / Data Engineer · Centrale d'achat d'acteurs du BTP : chaque adhérent reçoit un e-shop généré automatiquement à partir de son plan de vente.

- Conçu la base de données et les flux d'import qui alimentent 5 e-shops à partir de 2 millions de produits fournisseurs.
- Un format pivot unique : quel que soit le fichier reçu (Excel, JSON, TXT), il est normalisé puis intégré par des jobs Talend.
- Rapprochement avec la base produits de référence (PIM) par EAN, ou à défaut par une clé marque + référence commerciale, puis enrichissement (images, sources) avant la création automatique de l'e-shop.

MySQL, Talend, Node-RED, Metabase

> **Mon périmètre** · Équipe de 3 : j'ai conçu le process d'import de bout en bout et développé les flux avec un second développeur ; le chef de projet testait et validait.

Natixis (Groupe BPCE)

09/2021 – 03/2023

Développeur Full-Stack Java / Angular

- Réalisé les API REST et les interfaces de la plateforme crypto du groupe, dans une équipe de 15 en Agile à l'échelle (SAFe).

Java, Spring Boot, Angular, TypeScript, RxJS, JUnit

Formation

2024	Mastère Développeur Full-Stack · Cloud Campus
2022	Bachelor Chef de projet digital · EEMI
2026	MOOC CNIL — L'atelier RGPD · 6 modules validés
Langues	Français natif · anglais professionnel