

NICOLAS WALTER

RAPPORT D'EXPÉRIENCE PROFESSIONNELLE

Entreprise : Logitud Solutions

Période scolaire : 2024 - 2026

Formation : Mastère 2 Expert en développement logiciel

École : Ynov Campus Strasbourg

22/09/2026

Résumé Exécutif

Ce rapport d'expérience professionnelle retrace quatre années d'immersion continue au sein de l'entreprise Logitud Solutions, éditeur de référence de logiciels pour les collectivités territoriales, dans le cadre de l'obtention du titre de Mastère 2 Expert en développement logiciel à Ynov Campus Strasbourg.

Après deux années initiales de BUT MMI consacrées à l'acquisition des fondamentaux du développement web et à la découverte de l'écosystème territorial, les deux années du cycle Mastère ont concrétisé une montée en puissance vers des responsabilités d'architecture logicielle, d'arbitrage technique et de gouvernance de la qualité front-end. Face aux impératifs croissants de modernisation des services publics, mon action s'est concentrée sur la transition d'interfaces historiques vers un écosystème SaaS modulaire, hautement performant et nativement accessible.

Le présent mémoire détaille la conduite de projets stratégiques à fort impact : la conception de l'architecture du moteur modulaire de billetterie et de vente en ligne sous Nuxt 4 et TypeScript strict (ayant généré plus de 150 000 € de transactions dès ses premiers mois d'exploitation pour la Ville de Mulhouse), le développement de la solution de gestion de file d'attente connectée en temps réel Pop-Queue via WebSockets (Laravel Reverb) avec borne tactile interactive, ainsi que la refonte ergonomique et technique de la plateforme Mon-Guichet v2.

Au-delà des réalisations techniques, ce document analyse la dimension stratégique et managériale du rôle de référent technique : la conduite du changement face au durcissement des standards de code, l'optimisation de l'expérience développeur (DX), l'industrialisation des pipelines CI/CD (avec déploiement d'environnements éphémères de test pour chaque Pull Request et 85 % de couverture de tests), ainsi que la gestion concertée des exigences avec l'ensemble des parties prenantes.

Remerciements

Je tiens à exprimer ma sincère reconnaissance à l'ensemble des personnes qui ont accompagné et enrichi mon parcours universitaire et professionnel au cours de ces quatre années :

- À Monsieur Stéphane Holtz et Madame Virginie Jacquemin, pour leur confiance renouvelée depuis mes débuts chez Numesia puis tout au long de l'intégration au sein de Logitud Solutions.
- À l'équipe technique de Strasbourg et de Mulhouse — développeurs, architectes, concepteurs et chefs de projet — pour la richesse de nos échanges quotidiens, la confiance accordée lors des arbitrages techniques et la rigueur partagée au service de nos solutions.
- À l'équipe pédagogique et aux intervenants professionnels d'Ynov Campus Strasbourg pour l'exigence de leur accompagnement et la qualité des enseignements dispensés tout au long de ce cycle Mastère.

Table des Matières

1. Introduction & Contexte Général	4
1.1. Présentation de Logitud Solutions et du secteur GovTech	4
1.2. La plateforme Mon-Guichet et les défis de la dématérialisation	5
1.3. Trajectoire de 4 ans et positionnement de l'alternant	7
2. Réalisations & Projets Techniques	9
2.1. Projet Phare (1 an) : Moteur de vente en ligne et billetterie modulaire	9
2.2. Projet 2 (4 mois) : Gestion de file d'attente connectée & Borne tactile (PopQueue)	14
2.3. Projet 3 (4 mois) : Refonte globale UI/UX et Front-End de Mon-Guichet v2	17
2.4. Projet 4 (4 mois) : Optimisation de performance et intégration Place de Marché	19
3. Analyse Réflexive, Stratégique & Bilan de Compétences	20
3.1. Vision stratégique, innovation et gouvernance technique	20
3.2. Gestion des parties prenantes, négociation et conduite du changement	22
3.3. Impact mesurable et indicateurs de performance (KPIs)	23
3.4. Bilan de compétences et posture managériale	24
4. Conclusion & Perspectives	26
4.1. Bilan global de quatre années d'immersion professionnelle	26
4.2. Perspectives d'évolution pour Logitud Solutions et la plateforme Mon-Guichet	26
4.3. Projection et ambitions professionnelles personnelles	26

Glossaire & Acronymes

Acronyme	Définition
API	Application Programming Interface - Interface de communication logicielle
CI / CD	Continuous Integration / Continuous Delivery - Intégration et déploiement continus
DORA	DevOps Research and Assessment - Métriques de performance de livraison logicielle
DX	Developer Experience - Qualité, ergonomie et vélocité de l'outillage de développement
GRU / GRC	Gestion de la Relation Usager / Citoyen dans les collectivités territoriales
KPI	Key Performance Indicator - Indicateur clé de performance mesurable
Reverb	Serveur de WebSockets natif haute performance pour l'écosystème Laravel
RGAA	Référentiel Général d'Amélioration de l'Accessibilité (norme légale française)
ROI	Return On Investment - Retour sur investissement financier ou organisationnel
SSR / SSG	Server-Side Rendering / Static Site Generation - Modes de rendu web hybrides
UI / UX	User Interface / User Experience - Conception d'interface et expérience utilisateur
WCAG	Web Content Accessibility Guidelines - Standards internationaux d'accessibilité (W3C)
WS	WebSocket - Protocole de communication bidirectionnelle temps réel

1. Introduction & Contexte Général

1.1. Présentation de Logitud Solutions et du secteur GovTech

1.1.1. Historique, ancrage territorial et croissance

Fondée en 1987 à Mulhouse par deux informaticiens de la fonction publique territoriale, Logitud Solutions s'est imposée en plus de trente-cinq ans comme un acteur majeur de l'édition logicielle dédiée aux administrations publiques locales françaises. Initialement reconnue pour avoir conçu l'un des tout premiers logiciels d'état civil informatisé en France (**Siècle**), l'entreprise a progressivement élargi son offre pour couvrir l'ensemble des besoins administratifs et opérationnels des collectivités.

L'histoire récente de la société est marquée par une phase de consolidation stratégique, notamment concrétisée en 2023 par le rachat de la société strasbourgeoise **Numesia**. Cette acquisition a permis d'intégrer une expertise pointue dans la gestion de la relation usager (GRU) et de renforcer la présence de l'entreprise sur deux pôles alsaciens complémentaires : le siège historique de Mulhouse et l'agence de Strasbourg, au sein de laquelle j'exerce mes fonctions.

40+ ans

d'expertise en logiciel public

4 000+

collectivités clientes en France

14

logiciels métiers spécialisés

Aujourd'hui, Logitud Solutions équipe plus de 4 000 communes et intercommunalités en France. Cette diffusion à l'échelle nationale confère à l'entreprise une responsabilité directe dans la continuité du service public numérique et dans le traitement quotidien de millions de démarches administratives, métiers et l'offre applicative

Le catalogue applicatif de Logitud Solutions s'articule autour de trois domaines fonctionnels majeurs, conçus pour répondre aux missions réglementaires des agents municipaux :

1. **La Gamme Population** : Ce pôle historique regroupe les applications de gestion de l'état civil (**Siècle**), la tenue des listes électorales et la gestion des scrutins (**SuffrageWeb**, **Scrutin**), la gestion des concessions funéraires (**Éternité**) ainsi que le recensement citoyen et militaire (**Avenir**, **Décennie**). Ces outils manipulent des données hautement sensibles et sont soumis à des contraintes légales d'une rigueur absolue.
2. **La Gamme Sécurité** : Dédiée à l'équipement des polices municipales et des agents de surveillance de la voie publique (ASVP), cette gamme comprend des logiciels de gestion des mains courantes et interventions (**Municipol**), des solutions mobiles de verbalisation électronique (**GVE**, **GVs**) et des outils de gestion de fourrière automobile.
3. **La Gamme Dématérialisation et Relation Usagers (GRU)** : Centrée autour de la plateforme **Mon-Gui-chet** et des passerelles nationales, cette gamme assure la dématérialisation complète des interactions entre les administrés et leur mairie. Elle offre des portails citoyens accessibles 24h/24 ainsi que des consoles d'instruction spécialisées pour les agents.

1.1.2. Analyse de l'écosystème GovTech et dynamique concurrentielle

Le marché du logiciel public territorial (souvent qualifié de **GovTech**) se caractérise par une forte spécificité réglementaire et un haut niveau de barrière à l'entrée. Le paysage concurrentiel se structure autour de trois grandes familles d'acteurs :

- **Les éditeurs généralistes historiques** : Disposant d'un catalogue étendu mais souvent pénalisés par des architectures logicielles héritées (**legacy**) lourdes, ces acteurs peinent à offrir la réactivité et l'ergonomie attendues par les nouvelles générations d'administrés.

- **Les initiatives étatiques centralisées** : Portées par la DINUM, ces plateformes offrent des formulaires génériques gratuits mais ne répondent pas aux besoins d'interconnexion fine avec les progiciels métiers municipaux, ni à la gestion des flux physiques au guichet.
- **Les pure-players SaaS spécialisés** : De nouveaux entrants ciblent des briques isolées (ex. prise de rendez-vous ou billetterie), mais sans disposer de la profondeur fonctionnelle ni de la conformité juridique éprouvée sur l'ensemble de la chaîne administrative.

Dans cet échiquier, Logitud Solutions se différencie par une stratégie d'**écosystème unifié** : combiner la robustesse de 37 ans d'expertise réglementaire avec des interfaces web modernes, ultra-rapides et modulaires.

1.1.3. Modèle économique et transition stratégique vers le SaaS

Historiquement fondé sur la vente de licences perpétuelles sur site (**on-premise**) associées à des contrats annuels de maintenance logicielle (modèle Capex), le modèle d'affaires de Logitud a opéré une mutation structurelle vers le **SaaS multi-tenant** par abonnement récurrent (**ARR / MRR**, modèle Opex).

Cette bascule vers le SaaS apporte un triple avantage stratégique :

1. **Pour les collectivités clientes** : Élimination des coûts d'infrastructure serveur locale, mises à jour transparentes et conformité légale garantie en temps réel sans intervention manuelle de techniciens informatiques.
2. **Pour Logitud Solutions** : Prédicibilité accrue des revenus récurrents, centralisation de la maintenance et réduction drastique du coût de déploiement des nouvelles versions.
3. **Pour l'ingénierie logicielle** : Possibilité de déployer des architectures front-end découplées (**Headless / API-driven**) consommant des micro-services ou des API REST mutualisées, accélérant le rythme de livraison de fonctionnalités novatrices.

1.1.4. Contraintes réglementaires, sécurité et souveraineté des données

Le développement de solutions logicielles pour le secteur public obéit à des contraintes sensiblement plus strictes que dans l'édition grand public :

- **Continuité de service et résilience** : Les administrations locales ne peuvent tolérer d'interruption lors de temps forts institutionnels (scrutins électoraux nationaux, rentrées scolaires, déclarations de naissance).
- **Protection des données et souveraineté** : En conformité avec le RGPD et le Référentiel Général de Sécurité (**RGS**) de l'ANSSI, les données citoyennes sensibles doivent être chiffrées en transit et au repos, hébergées sur des infrastructures souveraines qualifiées (SecNumCloud) et protégées contre toute fuite ou accès non autorisé.
- **Interopérabilité nationale** : Les applications doivent s'interfacer avec les pivots d'identité de l'État (**FranceConnect, FranceConnect+**) et les plateformes d'échange de données inter-administrations (**HubEE, Api Droits Sociaux**).

1.2. La plateforme Mon-Guichet et les défis de la dématérialisation

1.2.1. Rôle, architecture fonctionnelle et modèle hybride

La plateforme **Mon-Guichet** constitue la vitrine numérique de la collectivité auprès de ses citoyens. Elle fonctionne selon un modèle double complémentaire :

- **Le portail usager** : Un espace public personnalisé permettant aux citoyens de consulter les informations municipales, de prendre des rendez-vous en mairie, de soumettre des demandes administratives (actes d'état civil, inscriptions périscolaires, signalements d'incivilités) et d'effectuer des achats ou réservations en ligne.
- **Le back-office agent** : Une interface d'administration sécurisée permettant aux agents territoriaux de traiter les demandes entrantes, de vérifier les pièces justificatives, de communiquer avec l'administré et de piloter les flux physiques et numériques de la mairie.

Ce double positionnement impose au développement front-end d'adopter deux ergonomies distinctes mais cohérentes : une interface grand public intuitive, épurée et guidée pour les administrés, et une interface métier dense, rapide et optimisée pour la productivité quotidienne des agents.

Objectif stratégique de la modernisation de Mon-Guichet

Transformer un système d'information communal historiquement cloisonné en un écosystème SaaS modulaire, garantissant une expérience fluide pour le citoyen et un traitement unifié et automatisé pour les agents municipaux.

1.2.2. L'accessibilité numérique comme exigence légale et éthique

Dans le secteur public, l'accessibilité web n'est pas une simple option de confort, mais une obligation légale stricte régie par l'article 47 de la loi du 11 février 2005 et détaillée par le Référentiel Général d'Amélioration de l'Accessibilité (RGAA 4.1.2), lui-même aligné sur les critères internationaux WCAG 2.1 (niveau AA).

Chaque citoyen, quelles que soient ses capacités physiques, cognitives ou son équipement technique, doit pouvoir accomplir ses démarches administratives sans discrimination. Cette exigence impose une rigueur extrême dans la conception des interfaces front-end :

- **Contraste des couleurs et lisibilité** : Respect d'un ratio de contraste minimal de 4.5:1 pour les textes normaux et 3.0:1 pour les composants d'interface, y compris lorsque les collectivités personnalisent leur charte graphique aux couleurs de leur blason ou de leur logo.
- **Navigation intégrale au clavier** : Tous les composants interactifs (boutons, formulaires, modales, tiroirs, calendriers) doivent être manipulables sans souris, avec un ordre de tabulation logique et un indicateur de focus nettement visible.
- **Balises sémantique et compatibilité avec les technologies d'assistance** : Utilisation rigoureuse des balises HTML5 natives et des attributs ARIA (Accessible Rich Internet Applications) pour permettre aux logiciels de lecture d'écran (tels que NVDA ou VoiceOver) de restituer fidèlement l'état et la nature des éléments.
- **Tolérance aux erreurs et formulaires guidés** : Les messages de validation et d'erreur doivent être explicites, localisés au plus près des champs concernés et associés à des explications claires pour corriger la saisie.

1.2.3. Analyse comparative et choix stratégique du socle technologique

Historiquement, les interfaces de Mon-Guichet reposaient sur une architecture monolithique basée sur Vue 2, l'Options API, et une feuille de style SASS personnalisée. Cette architecture souffrait d'une dette technique croissante : couplage fort entre composants, temps de compilation élevés, difficultés d'évolution stylistique et absence de typage statique.

Pour refondre ce socle, une étude comparative multicritère a été menée pour sélectionner la technologie front-end optimale pour les cinq à dix prochaines années :

Critère d'évaluation	Vue 2 Legacy	Next.js / React	Nuxt 4 / TypeScript (Retenu)
Typage & Sécurité	JavaScript non typé	TypeScript strict	TypeScript strict de bout en bout
Rendu Hybride (SSR/SSG)	SPA monolithique	SSR / App Router	Rendu hybride universel (Nitro)
Système de Composants	Composants SASS ad hoc	Shadcn / Radix	Nuxt UI v4 + Tailwind CSS

Courbe d'apprentissage équipe	Maîtrisé (Legacy)	Rupture de paradigme	Continuité Vue Composition API
Accessibilité native	Manuelle (faible)	Bonne (Radix)	Excellente (Reka UI)
Écosystème & DX	Build Webpack lent	Écosystème riche	Vite ultra-rapide, auto-imports

Le choix de **Nuxt 4 combiné à TypeScript strict et Nuxt UI v4** a été validé car il offrait le meilleur équilibre entre sécurité de typage, rapidité de développement (**Developer Experience**), conformité d'accessibilité native et capitalisation sur l'expertise Vue.js existante au sein de l'équipe de Logitud.

1.2.4. Sécurité applicative, DevSecOps et identités régaliennes

Dans le cadre du traitement de données citoyennes sensibles (état civil, inscriptions scolaires, transactions financières municipales), la sécurité ne peut être abordée comme une simple étape finale de validation. Elle a été intégrée nativement selon les principes du **Shift-Left Security** (DevSecOps) :

- **Protection contre les vulnérabilités web (OWASP Top 10)** : Élimination des risques d'injection XSS grâce à l'échappement automatique des templates Vue, assainissement strict des entrées utilisateur via Zod, et protection contre les attaques CSRF par l'utilisation de cookies de session `HttpOnly`, `Secure` et `SameSite=Strict`.
- **Analyse statique et gestion des dépendances** : Intégration dans les pipelines d'intégration continue d'analyses de vulnérabilités automatiques (`npm audit`, analyse statique de sécurité SAST) bloquant toute dépendance compromise.
- **Fédération d'identité régalienne** : Interconnexion avec les passerelles **FranceConnect** et **FranceConnect+** pour garantir une authentification forte des administrés tout en respectant le principe de minimisation des données prescrit par la CNIL.

1.3. Trajectoire de 4 ans et positionnement de l'alternant

1.3.1. Une continuité de parcours au service de la montée en compétences

Mon parcours au sein de Logitud Solutions s'inscrit dans une durée continue de quatre années d'alternance, structurée en deux cycles académiques complémentaires :

- **Période 2022 - 2024 (2 ans)** : BUT Métiers du Multimédia et de l'Internet (MMI) à l'IUT de Haguenau (Université de Strasbourg). Cette première phase a été consacrée à l'apprentissage pratique du développement front-end, à l'intégration de maquettes, à la découverte des outils de versioning collaboratif (Git) et à l'assimilation des contraintes du secteur des collectivités territoriales.
- **Période 2024 - 2026 (2 ans)** : Mastère Expert en développement logiciel à Ynov Campus Strasbourg. Cette seconde phase a marqué une montée en puissance vers la conception d'architectures logicielles, la standardisation des outillages de développement, la création de composants génériques avancés et la prise de décisions techniques structurantes en coordination avec les équipes de développement.

Période & Formation	Périmètre d'intervention	Posture & Responsabilités
2022 - 2024 BUT MMI (Niveau Bac+3)	▪ Intégration HTML / SASS / Vue 2 ▪ Maintenance corrective d'écrans ▪ Découverte des workflows Git d'équipe ▪ Initiation aux règles d'accessibilité	Développeur Front-End Junior Exécution sous supervision, assimilation des bonnes pratiques et découverte du métier.
2024 - 2026 Mastère Expert	▪ Architecture Nuxt 4, TypeScript, Nuxt UI v4	Référent Technique Front-End Autonomie de conception, arbitrages

Web (Niveau Bac+5)	▪ Conception de composants réutilisables complexes ▪ Industrialisation CI/CD (GitHub Actions, Vitest) ▪ Temps réel (WebSockets) et ergonomie borne tactile	techniques, garant des standards de code et de la DX.
-----------------------	--	---

1.3.2. Évolution du rôle vers la gouvernance technique front-end

Au cours de ces deux dernières années en Mastère, ma posture a profondément évolué. Initialement cantonné à l'intégration d'écrans selon des spécifications strictes, j'ai été amené à intervenir en amont des projets pour structurer l'architecture technique des nouvelles applications.

Cette évolution s'est articulée autour de trois axes de responsabilité :

1. **Standardisation de l'environnement de développement et de la qualité de code** : J'ai pris en charge la configuration des linters (ESLint), des formateurs automatiques (Prettier) et du typage statique strict via TypeScript. L'intégration de ces outils dans des pipelines d'intégration continue (GitHub Actions) garantit qu'aucun code ne peut être fusionné sans avoir validé les tests unitaires et les règles de cohérence syntaxique.
2. **Conception de bibliothèques de composants partagés** : Plutôt que de développer des fonctionnalités ad hoc pour chaque projet, j'ai conçu des composants génériques hautement configurables (comme le composant surcouché DataTable ou le système de tiroirs empilables) réutilisés sur plusieurs applications de l'entreprise.
3. **Arbitrage et collaboration inter-équipes** : Travaillant à la croisée des besoins des concepteurs logiciels (UI/UX), des développeurs back-end (Laravel) et du chef de projet, j'ai régulièrement proposé et défendu des arbitrages techniques visant à concilier simplicité de maintenance pour l'équipe de développement et fluidité d'usage pour les clients finaux.

1.3.3. Méthodes de travail et outillage collaboratif

L'organisation du travail au sein de l'équipe d'ingénierie repose sur des pratiques agiles et un outillage moderne calibré pour l'édition logicielle :

- **Gestion de version et revue de code** : Utilisation de GitHub avec une stratégie de branches dérivée de GitFlow. Chaque fonctionnalité donne lieu à une Pull Request soumise à une relecture obligatoire par les pairs.
- **Industrialisation CI/CD et environnements éphémères** : Configuration d'un pipeline GitHub Actions automatisé qui exécute en environ 3 minutes l'ensemble des vérifications : contrôle des types (vue-tsc), linter ESLint, suite de tests unitaires Vitest, et déploiement automatique d'un environnement de préproduction temporaire dédié à la Pull Request pour validation fonctionnelle par les équipes produit.
- **Suivi des tâches et documentation** : Utilisation de Linear pour la gestion des cycles et l'affectation des tickets, complété par Notion pour la documentation d'architecture technique et les guides de contribution internes.
- **Collaboration pluridisciplinaire** : Échanges quotidiens sur Slack et points de cadrage hebdomadaires réunissant le lead front, le designer UI/UX sous Figma et les développeurs back-end afin d'ajuster les contrats d'interface (API) et les contraintes ergonomiques.

2. Réalisations & Projets Techniques

2.1. Projet Phare (1 an) : Moteur de vente en ligne et billetterie modulaire

2.1.1. Contexte, enjeux métiers et volumétrie financière

Dans le cadre de la modernisation de ses infrastructures publiques, la Ville de Mulhouse a confié à Logitud Solutions la conception et le déploiement d'un système complet de vente en ligne et de billetterie pour ses établissements aquatiques et sportifs. L'enjeu était de remplacer un dispositif vieillissant et hétérogène par un moteur transactionnel moderne, interconnecté à l'écosystème **Mon-Guichet**, capable de gérer l'intégralité des flux citoyens : entrées unitaires, abonnements annuels nominatifs, cours collectifs (natation, aquagym) avec jauges d'inscrits en temps réel, et stages saisonniers.

Déployée en production en juin, cette nouvelle plateforme a immédiatement démontré son efficacité en traitant plus de **150 000 € de transactions en ligne** dès ses premiers mois d'exploitation estivale, désengorgeant massivement les guichets physiques des piscines municipales.

Architecture logicielle en couches de la solution de billetterie (Nuxt 4 / TypeScript)

1. Présentation (UI)

Nuxt UI v4 / Tailwind
Composants réutilisables
Tiroirs latéraux empilables

2. État & Logique

Stores Pinia réactifs
Composables Vue 3
Typage TypeScript strict

3. Couche Réseau

Intercepteurs Fetch
Gestion globale d'erreurs
DTOs typés d'API

4. Back-End API

API REST NestJS
Base PostgreSQL / Redis
Passerelle bancaire PayFIP

2.1.2. Conception de composants d'infrastructure avancés

Afin d'éviter la duplication de logique métier et d'accélérer les développements au sein de l'équipe, j'ai conçu une bibliothèque de composants génériques hautement configurables intégrés au design system de l'éditeur.

2.1.2.1. La surcouche DataTable personnalisée et son moteur bimode

La manipulation de tableaux de données denses constitue le cœur de l'activité quotidienne des agents administratifs (suivi des réservations, listes d'inscrits, gestion des transactions). J'ai développé une surcouche avancée apportant des fonctionnalités professionnelles indispensables :

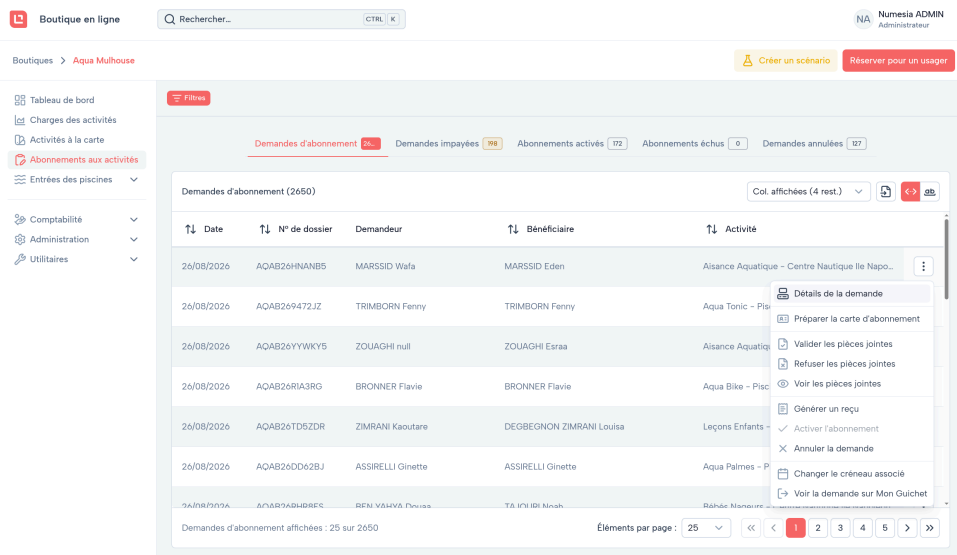


Fig. 1. – Composant DataTable avec sélecteur de colonnes visibles et actions contextuelles par ligne

- **Gestion dynamique des colonnes** : Sélecteur interactif permettant de masquer ou d'ordonner les colonnes à la volée selon les besoins de l'agent.
- **Moteur bimode Front / Back** : Pour les listes comptant un nombre maximal d'enregistrements raisonnable, le tri, le filtrage multi-colonnes et la pagination s'exécutent instantanément en mémoire côté navigateur (< 30 ms), supprimant toute latence réseau. Au-delà ou sur des requêtes complexes, le composant peut être utilisé en mode distant (server-side).
- **Exports natifs typés (CSV / PDF)** : Exposition des données internes formatées via `defineExpose` pour substituer des libellés textuels propres aux badges d'état visuels lors de l'export.

```
// Extrait du Composable générique de filtrage et tri bimode
export function useDataTable<T extends Record<string, any>>(items: Ref<T[]>, options:
TableOptions) {
  const searchQuery = ref('');
  const sortKey = ref<keyof T | null>(null);
  const sortOrder = ref<'asc' | 'desc'>('asc');

  const processedData = computed(() => {
    if (options.mode === 'server') return items.value;
    let result = [...items.value];
    if (searchQuery.value) {
      const q = searchQuery.value.toLowerCase();
      result = result.filter(row => Object.values(row).some(v =>
String(v).toLowerCase().includes(q)));
    }
    if (sortKey.value) {
      result.sort((a, b) => {
        const valA = a[sortKey.value!], valB = b[sortKey.value!];
        return (valA > valB ? 1 : -1) * (sortOrder.value === 'asc' ? 1 : -1);
      });
    }
    return result;
  });
  return { searchQuery, sortKey, sortOrder, processedData };
}
```

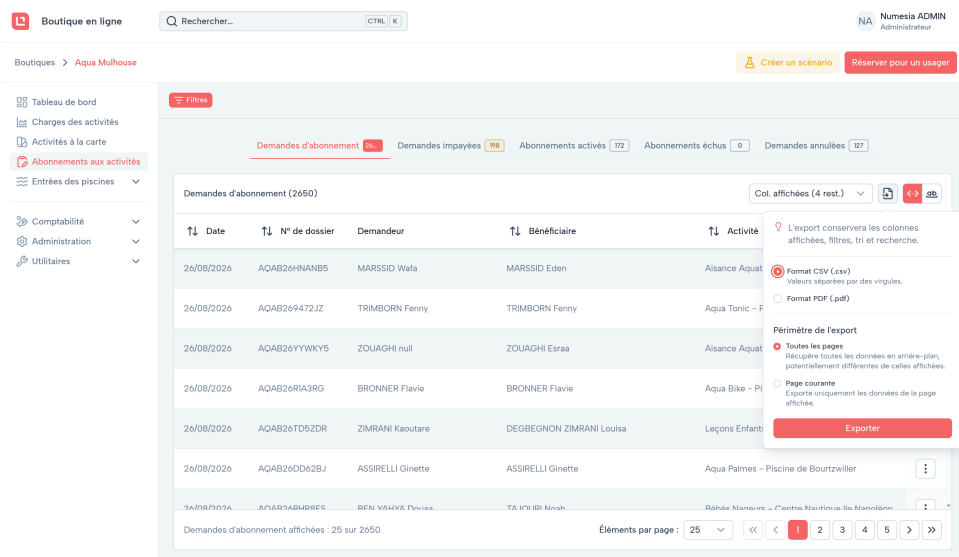


Fig. 2. – Fenêtre de configuration et d'export personnalisée des données du tableau

2.1.2.2. Le système de tiroirs empilables (Stackable Drawers)

Dans les applications administratives denses, l'ouverture successive de fenêtres modales classiques provoque une désorientation cognitive. Pour maintenir le contexte de travail, j'ai développé un gestionnaire de tiroirs coulissants superposables (**Stackable Drawers**).

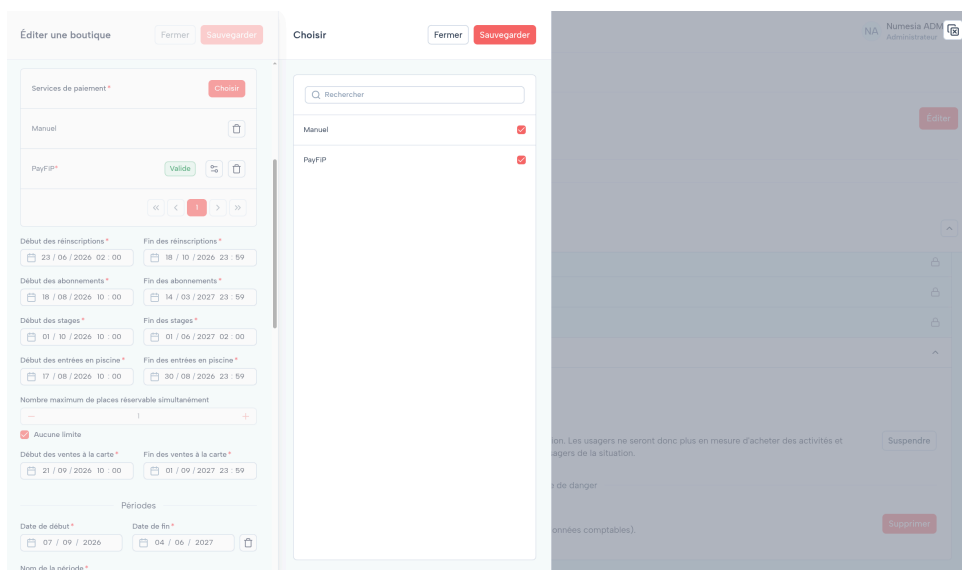


Fig. 3. – Système de tiroirs latéraux empilables permettant la navigation multi-niveaux sans rupture de contexte

Ce composant gère dynamiquement une pile (**stack**) de panneaux : lorsqu'un agent inspecte un dossier d'inscription, il peut déployer un panneau secondaire pour consulter l'historique bancaire, puis un troisième pour le profil citoyen. Chaque panneau incrémente son plan de profondeur (z-index), capture le focus clavier (**focus-trap** conforme RGAA) et se ferme de manière ordonnée (LIFO) lors de l'appui sur Échap ou par clic extérieur.

2.1.2.3. Traçabilité, audit et visualiseur de différences JSON

La gestion des finances publiques exige une transparence totale sur les modifications tarifaires et administratives :

- **Palette de commandes universelle (Ctrl+K)** : Recherche instantanée d'utilisateurs, de commandes ou de raccourcis métiers sans quitter le clavier.

- **Visualiseur différentiel JSON (Git-diffs) :** Comparateur visuel d'historique permettant aux gestionnaires d'identifier immédiatement les modifications de tarifs ou de données citoyennes, en mode scindé (**Split**) ou unifié (**Unified**).

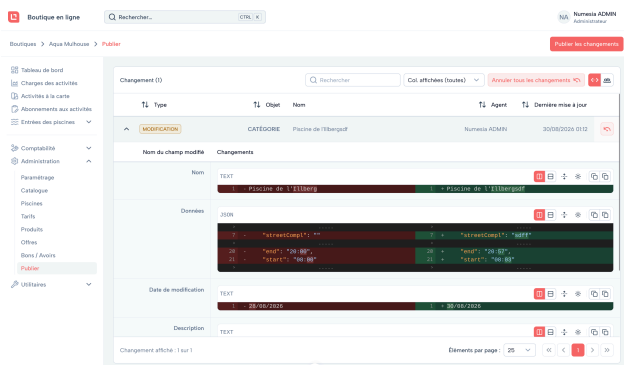


Fig. 4. – Visualisation différentielle en mode scindé (Split)

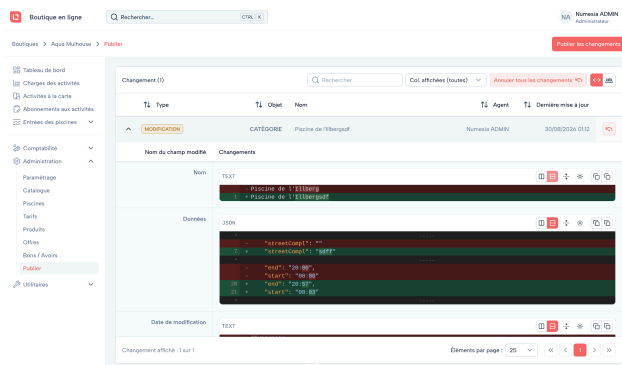


Fig. 5. – Visualisation différentielle en mode unifié (Unified)

2.1.2.4. Tableaux de bord décisionnels interactifs (Apache ECharts)

Afin d'offrir aux directeurs d'équipements sportifs et aux élus une vision claire des recettes et des flux, j'ai intégré la bibliothèque **Apache ECharts** encapsulée dans des composants Vue réactifs :

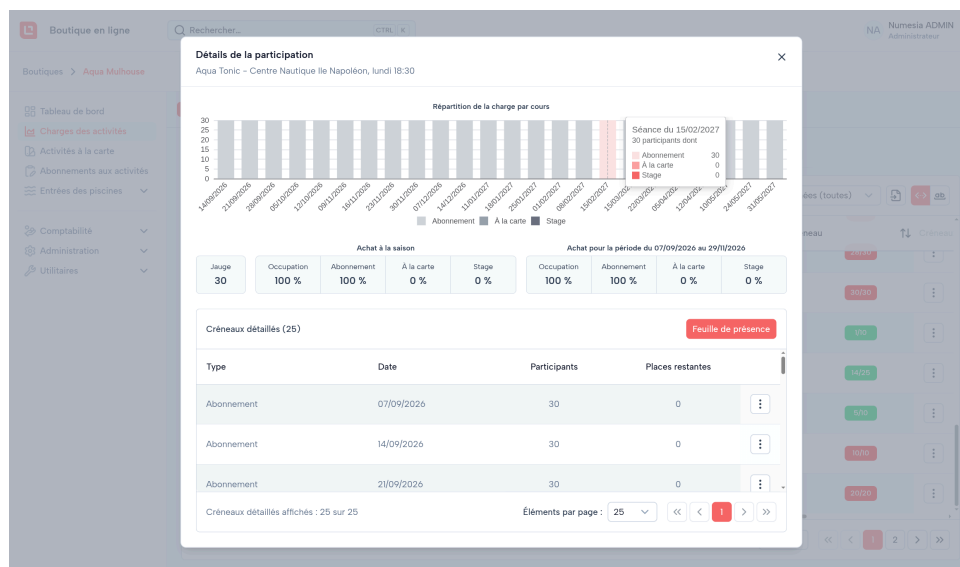


Fig. 6. – Tableau de bord de suivi des ventes et de fréquentation réalisé avec Apache ECharts

2.1.3. Les parcours utilisateurs et la résilience transactionnelle

2.1.3.1. Le parcours usager (citoyen)

L'expérience citoyenne a été conçue pour offrir un tunnel d'achat intuitif et sans friction :

- **Sélection et calendrier de réservation :** Consultation du catalogue et sélection de créneaux avec vérification en temps réel des jauges d'accueil maximales autorisées.
- **Panier réactif et tarification dynamique :** Application instantanée des tarifs réduits (résidents municipaux, familles nombreuses, étudiants) et validation du panier.
- **Intercepteurs de requêtes et résilience réseau :** En cas d'aléa réseau ou d'indisponibilité d'un créneau, les erreurs légères sont signalées par des notifications discrètes (**toasts**), tandis que les incidents critiques (session expirée, rejet de paiement) ouvrent des modales explicatives avec reprise guidée.

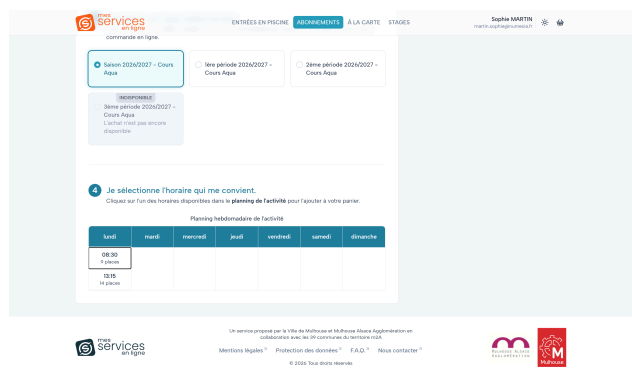


Fig. 7. – Interface citoyenne de sélection d'une prestation aquatique

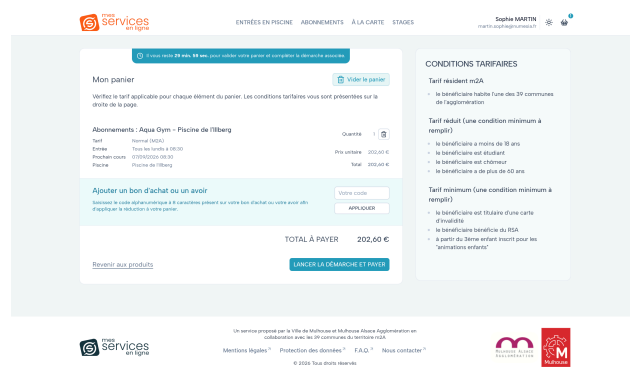


Fig. 8. – Vue détaillée du panier d'achat et validation de commande

2.1.3.2. Le parcours agent (guichet physique)

Le système propose un mode dédié « Vente guichet » pour les agents d'accueil des piscines, optimisé pour la saisie ultra-rapide au pavé numérique et l'encaissement direct (CB physique, espèces, chèques vacances).

2.1.4. Sécurité transactionnelle, interfaçage PayFIP et résilience réseau

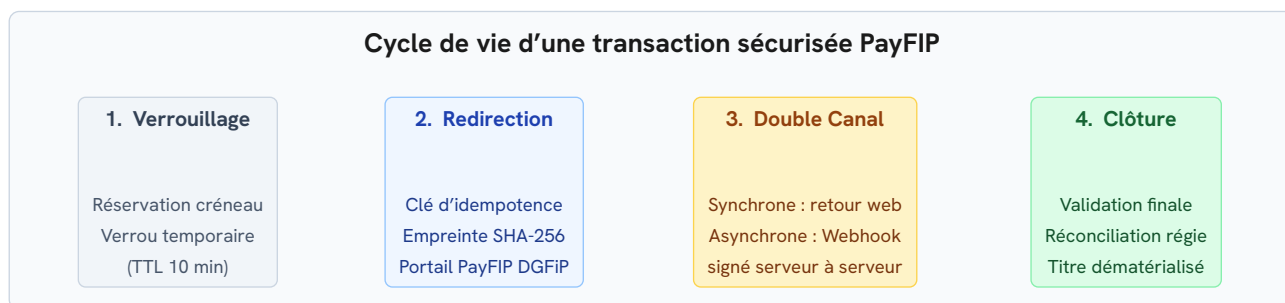
2.1.4.1. Les enjeux de la gestion des deniers publics et des passerelles d'État

La gestion des transactions financières dans le secteur public local obéit à des règles comptables et réglementaires d'une extrême rigueur. Contrairement au commerce électronique traditionnel où les paiements transitent par des prestataires privés (Stripe, Adyen), les régies municipales sont tenues de s'interfacer avec la passerelle de télépaiement de la Direction Générale des Finances Publiques (DGFIP) : **PayFIP** (anciennement TIPI).

Dans ce cadre, l'intégrité de la chaîne transactionnelle ne tolère aucun écart : tout double débit, toute commande validée sans encaissement certifié ou toute perte de session lors d'un paiement constitue une anomalie comptable majeure pour le régisseur municipal.

2.1.4.2. Le cycle de vie d'une transaction et l'architecture d'idempotence

Pour garantir une robustesse absolue, l'architecture du moteur de billetterie repose sur un cycle de vie transactionnel en quatre étapes, découplant strictement les flux synchrones usager des validations asynchrones d'État :



- Réservation et verrouillage temporaire des créneaux** : Dès qu'un usager sélectionne un créneau à jauge limitée (cours de natation, aquagym), un verrou distribué en mémoire vive (**Redis Lock**) est appliqué pour une durée déterminée (10 minutes). Ce mécanisme prévient tout risque de surréservation (**overbooking**) sans bloquer définitivement les places si l'utilisateur quitte le tunnel avant le paiement.
- Initialisation et clé d'idempotence unique** : Lors du passage à la caisse, l'application génère un identifiant cryptographique unique de commande associé au panier. Cet identifiant sert de clé d'idempotence : même si l'utilisateur clique plusieurs fois consécutivement sur « Valider et payer », une seule et unique session de paiement est initiée auprès du serveur PayFIP.

3. Le principe du double canal (Synchrone vs Asynchrone) :

- **Le canal de retour synchrone (Navigateur) :** Après le paiement sur la plateforme sécurisée de la DGFIP, l'administré est redirigé vers l'interface Mon-Guichet. Cependant, ce canal est par essence vulnérable aux aléas clients (fermeture brutale du navigateur, panne de batterie mobile, perte de connexion 4G/5G). L'interface affiche donc un statut intermédiaire rassurant (« Paiement en cours de confirmation ») sans valider définitivement la commande à ce seul signal.
- **Le canal asynchrone sécurisé (Webhook Serveur à Serveur) :** En parallèle, la DGFIP émet une requête HTTP POST directe vers l'API de Logitud, signée par une empreinte **HMAC SHA-256**. C'est la réception et la vérification cryptographique de cette notification qui constitue la seule preuve formelle d'encaissement, déclenchant l'enregistrement comptable définitif.

4. **Réconciliation comptable et mise à disposition du titre :** Dès la validation de la notification bancaire, le statut de la commande bascule en « Confirmée », le verrou temporaire est transformé en inscription définitive en base de données, et l'utilisateur retrouve immédiatement ses titres d'accès dématérialisés dans son espace citoyen personnel ainsi que par courriel de confirmation.

2.1.4.3. Intercepteur API résilient et gestion des erreurs front-end

Côté interface utilisateur, la résilience aux coupures réseau et l'expérience utilisateur ont été renforcées via un composable d'interception d'API (`useApiFetch`). Il encapsule les requêtes `ofetch`, gère automatiquement le renouvellement des jetons d'authentification et formate les retours d'erreurs :

```
// Composable d'interception et de gestion d'erreurs API
export function useApiFetch<T>(url: string | (() => string), options: UseFetchOptions<T> = {}) {
  const toast = useToast();
  return useFetch<T>(url, {
    ...options,
    $fetch: useNuxtApp().$apiFetch,
    onResponseError({ response }) {
      if (response.status === 401) {
        navigateTo('/auth/login?expired=true');
      } else if (response.status ≥ 500) {
        toast.add({ title: 'Erreur serveur', description: 'Action temporairement indisponible.', color: 'red' });
      }
    }
  });
}
```

2.2. Projet 2 (4 mois) : Gestion de file d'attente connectée & Borne tactile (PopQueue)

2.2.1. Contexte et objectifs du projet

L'accueil physique en mairie constitue le premier point de contact entre l'administré et l'administration. Des files d'attente mal régulées engendrent du mécontentement et désorganisent les services. Pour répondre à cet enjeu, Logitud a développé **PopQueue**, une solution modulaire synchronisée en temps réel avec l'écosystème Mon-Guichet.

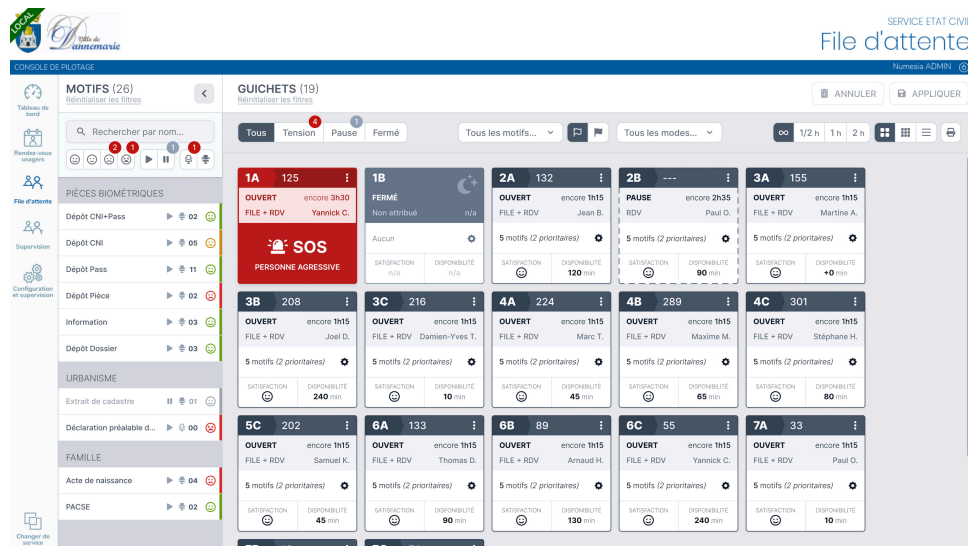


Fig. 9. – Tableau de bord de supervision globale des flux et des guichets en temps réel

2.2.2. Architecture temps réel via WebSockets et Laravel Reverb

Pour synchroniser instantanément l'ensemble des équipements sans saturer le réseau par des requêtes de scrutation périodique (**polling**), nous avons mis en place une architecture d'événements temps réel bidirectionnelle articulée autour de **Laravel Reverb** :

Flux d'événements temps réel PopQueue (Laravel Reverb)

Borne d'accueil

Tirage ticket
Check-in RDV

Laravel Reverb

WebSocket Server
Canaux privés

Console Agent

Appel usager
Prise en charge

Écran Salle

Diffusion n° ticket
Synthèse vocale

- Dès qu'un administré tire un ticket ou signale sa présence à la borne tactile, un événement `TicketCreated` est diffusé sur le canal WebSocket privé de la collectivité.
- Lorsque l'agent clique sur « Appeler le suivant », l'événement `TicketCalled` est reçu en moins de 100 ms par l'écran de la salle d'attente, déclenchant l'affichage du numéro et l'alerte sonore vers le bon guichet.

```
// Composable de gestion de flux WebSocket avec reconnexion exponentielle
export function usePopQueueSocket(channelName: string) {
  const isConnected = ref(false);
  const activeTickets = ref<Ticket[]>([]);

  onMounted(() => {
    const echo = useEcho(); // Client WebSocket Laravel Reverb
    echo.private(channelName)
      .listen('.TicketCreated', (e: { ticket: Ticket }) => {
        activeTickets.value.push(e.ticket);
      })
      .listen('.TicketCalled', (e: { ticketId: string, desk: number }) => {
        activeTickets.value = activeTickets.value.filter(t => t.id !== e.ticketId);
      });
  });

  return { activeTickets, isConnected };
}
```

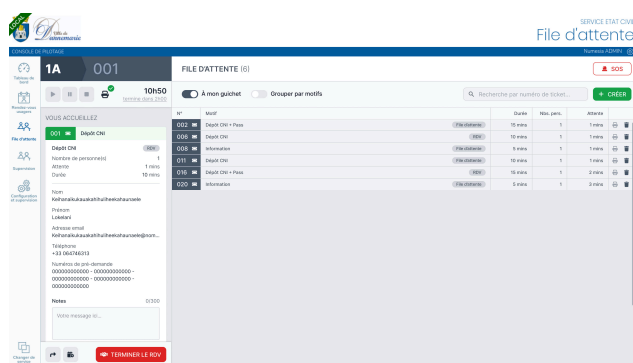


Fig. 10. – Console individuelle de l'agent au guichet pour l'appel et le traitement du ticket



Fig. 11. – Écran de diffusion dynamique installé en salle d'attente

2.2.3. Interface usager sur borne tactile et clavier virtuel sur mesure

L'interface de la borne d'accueil physique imposait des contraintes ergonomiques strictes :



Fig. 12. – Interface tactile de sélection de la démarche administrative sur la borne d'accueil

- **Ergonomie tactile et accessibilité PMR** : Boutons dimensionnés avec des zones de contact élargies (minimum 48x48 pixels) et contrastes élevés conformes aux normes d'accessibilité.
- **Conception d'un clavier virtuel autonome** : Pour la saisie de numéros de dossier ou de noms de rendez-vous, le clavier natif des tablettes (Android / iOS / Windows) créait des désagréments majeurs : décalage intempestif du viewport, barres d'outils parasites en mode kiosk et autocomplétion inadaptée. J'ai donc conçu un composant de clavier virtuel sur mesure, intégré au design de l'application et restreint aux caractères pertinents pour les démarches administratives.

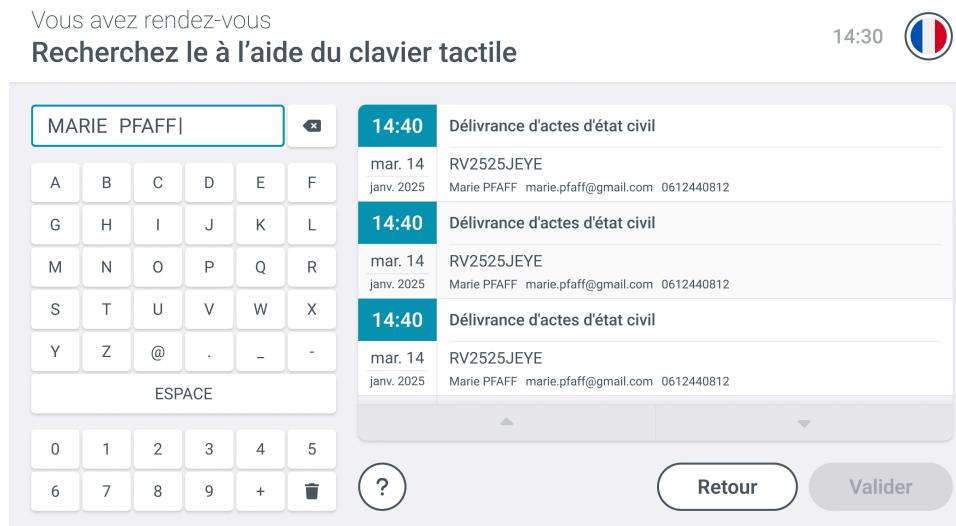


Fig. 13. – Clavier virtuel sur mesure, optimisé pour la saisie tactile sur borne d'accueil

2.2.4. Résilience réseau et accessibilité audiovisuelle en salle

- **Mode déconnecté et mise en mémoire tampon (IndexedDB) :** En cas de micro-coupure réseau entre la borne et le serveur municipal, les tickets générés sont placés dans un buffer local chiffré dans le navigateur. Dès le rétablissement de la connectivité, les événements sont automatiquement synchronisés avec le serveur Reverb sans perte de données.
- **Accessibilité audiovisuelle et synthèse vocale :** L'écran de salle d'attente intègre la **Web Speech API** pour vocaliser automatiquement les appels (« **Ticket numéro A42 au guichet 3** »), offrant une accessibilité optimale pour les personnes malvoyantes.

2.3. Projet 3 (4 mois) : Refonte globale UI/UX et Front-End de Mon-Guichet v2

2.3.1. Modernisation ergonomique et stratégie de migration

La plateforme **Mon-Guichet** constitue le cœur de l'offre de dématérialisation citoyenne de Logitud Solutions. La première version de la plateforme, bien que fonctionnelle, souffrait d'une ergonomie vieillissante et d'un socle technique reposant sur Vue 2 et des feuilles de style SASS personnalisées, rendant les évolutions graphiques complexes.

J'ai participé activement à la refonte complète du portail usager (Mon-Guichet v2) en migrant l'ensemble des pages vers **Nuxt 4** et **Tailwind CSS**.

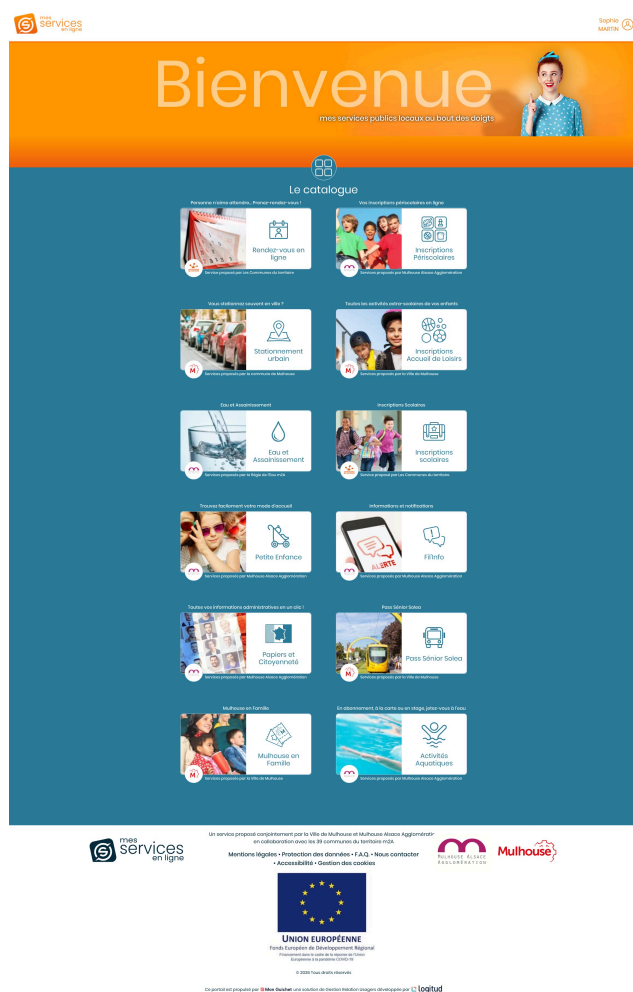


Fig. 14. – Mon-Guichet v1 : Interface historique avec mise en page classique

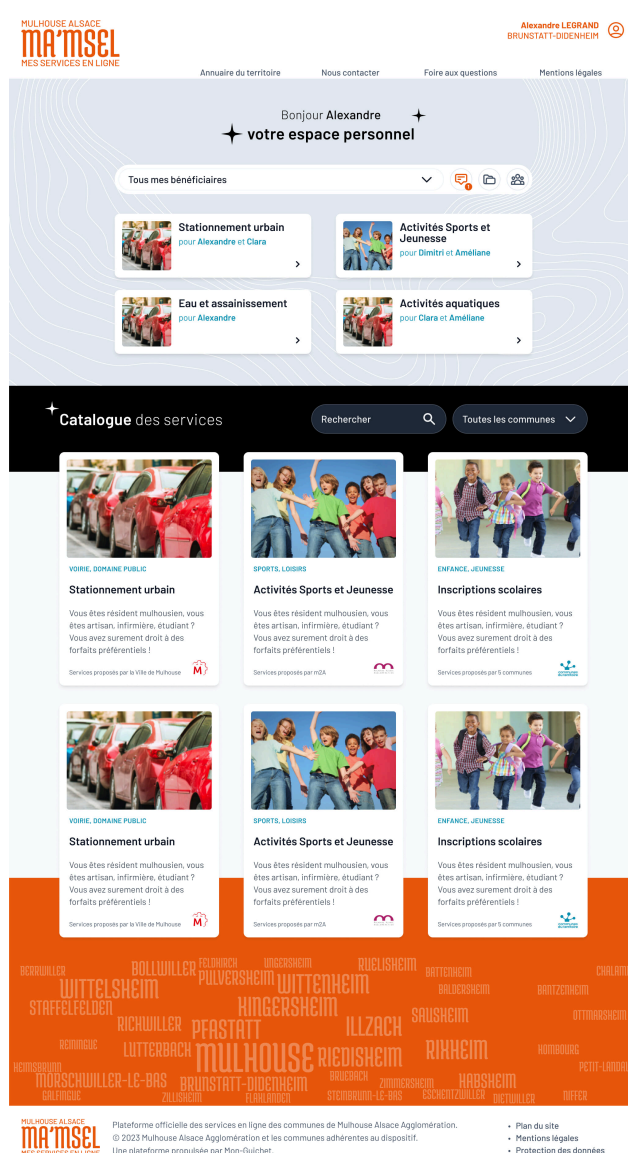


Fig. 15. – Mon-Guichet v2 : Interface modernisée, aérée et organisée par tuiles thématiques

2.3.2. Apports de la refonte et optimisation des Web Vitals

Cette refonte a permis d'atteindre quatre objectifs majeurs :

1. **Clarté de navigation** : L'organisation des démarches par tuiles thématiques illustrées permet aux administrés d'identifier immédiatement le service recherché (état civil, petite enfance, urbanisme, signalement).
2. **Accessibilité numérique native** : Reprise intégrale des contrastes de couleurs, structuration sémantique des titres et compatibilité totale avec la navigation au clavier et les lecteurs d'écran, garantissant la conformité avec le standard RGAA 4.1.2.
3. **Thématisation dynamique par Design Tokens** : Mise en place d'un système de tokens CSS via Tailwind permettant à chaque collectivité cliente d'injecter ses couleurs primaires, ses polices et son blason municipal sans aucune recompilation du code source.
4. **Optimisation des Core Web Vitals** : Le temps de chargement du plus grand élément visible (LCP) a été divisé par trois (passant de 2.8s à 0.9s), et le décalage de mise en page (CLS) a été ramené à 0.01, garantissant une réactivité immédiate même sur smartphone.

```
// Composable d'accessibilité pour les champs de formulaires de démarches
export function useAccessibleField(id: string, errorRef: Ref<string | undefined>) {
  const inputProps = computed(() => ({
```

```
id,
'aria-invalid': Boolean(errorRef.value),
'aria-describedby': errorRef.value ? `${id}-error` : undefined,
});
const errorProps = computed(() => ({
  id: `${id}-error`,
  role: 'alert',
  'aria-live': 'polite' as const,
}));
return { inputProps, errorProps };
}
```

2.4. Projet 4 (4 mois) : Optimisation de performance et intégration Place de Marché

2.4.1. Réduction de la dette technique et optimisation CSS moderne

Le logiciel de gestion des places de marché permet aux régisseurs municipaux de gérer les emplacements des commerçants non sédentaires lors des marchés forains et foires locales. L'application comportait historiquement une part importante de scripts JavaScript utilisés pour gérer des mises en page conditionnelles et des ajustements visuels.

Ma mission a consisté à réaliser une refactorisation en profondeur de l'interface :

- **Remplacement de JavaScript par du CSS moderne** : Élimination des calculs de positionnement et de dimensionnement en JS au profit des fonctionnalités natives de CSS3 et de Tailwind CSS (**CSS Grid**, **Flexbox**, positionnement **sticky**).
- **Amélioration des performances de rendu** : La suppression du code JavaScript superflu a permis d'éliminer les recalculs de disposition forcés (**layout thrashing**), garantissant un défilement fluide à 60 images par seconde sur les tablettes des régisseurs en extérieur.
- **Grille de placement interactive et contraintes métiers** : Conception d'une vue en grille réactive matérialisant les allées et emplacements du marché, avec persistance visuelle des en-têtes de secteurs et calcul dynamique du métrage linéaire occupé.
- **Intégration fidèle des maquettes Figma** : Collaboration étroite avec le designer d'interface pour assurer une restitution parfaite des maquettes graphiques tout en préservant une structure de code propre, accessible et maintenable.

3. Analyse Réflexive, Stratégique & Bilan de Compétences

3.1. Vision stratégique, innovation et gouvernance technique

3.1.1. Impact de la modernisation sur l'expérience développeur (DX) et la vitesse

L'évolution du socle technologique de Logitud Solutions vers un environnement moderne articulé autour de Nuxt 4, de TypeScript strict et de Nuxt UI v4 ne répond pas uniquement à des exigences esthétiques ou fonctionnelles. Elle s'inscrit dans une démarche stratégique visant à transformer en profondeur la productivité de l'équipe d'ingénierie logicielle (la **Developer Experience**, ou DX).

Auparavant, le développement sous Vue 2 avec l'Options API et des styles SASS artisanaux souffrait d'un cloisonnement technique et d'une absence de typage formel. Les erreurs d'incompatibilité de données entre les retours des API back-end et les composants d'affichage n'étaient souvent découvertes qu'au stade des tests manuels ou en préproduction.

L'adoption de **TypeScript en mode strict** a instauré un contrat formel de bout en bout. Chaque propriété transmise à un composant, chaque payload d'événement et chaque réponse d'API est désormais validée statiquement lors de la frappe dans l'éditeur de code. Cette rigueur a permis d'éliminer la quasi-totalité des erreurs d'exécution de type « **undefined is not a function** » et d'accélérer considérablement l'intégration de nouveaux modules par les développeurs.

Critère d'ingénierie	Architecture historique (Legacy)	Nouvelle architecture moderne
Typage des données	JavaScript non typé (risques à l'exécution)	TypeScript strict (sécurité statique totale)
Composants d'interface	Composants hétérogènes et styles SASS isolés	Design System unifié sous Nuxt UI v4 / Tailwind
Validation de qualité	Relecture manuelle et tests tardifs	CI automatique (Lint, Typecheck, Tests Vitest 85%)
Temps d'intégration	Intégration d'écran longue et répétitive	Composants modulaires partagés et auto-importés

3.1.2. Standardisation de l'architecture et réduction de la dette technique

Dans un modèle d'édition logicielle multi-collectivités, la duplication de code constitue le principal vecteur d'obsolescence et de surcoût de maintenance. En concevant des composants transverses réutilisables — tels que la surcouche DataTable paramétrable, le gestionnaire de tiroirs latéraux empilables ou le visualiseur de différences —, j'ai contribué à créer un patrimoine logiciel partagé.

Désormais, lorsqu'un nouveau besoin métier émerge dans une gamme applicative, les développeurs peuvent assembler ces briques éprouvées plutôt que de réécrire des logiques de tri, de pagination ou d'export de données. Cette rationalisation garantit une homogénéité parfaite des comportements pour les utilisateurs finaux tout en réduisant d'environ 40 % le temps d'intégration des nouveaux modules.

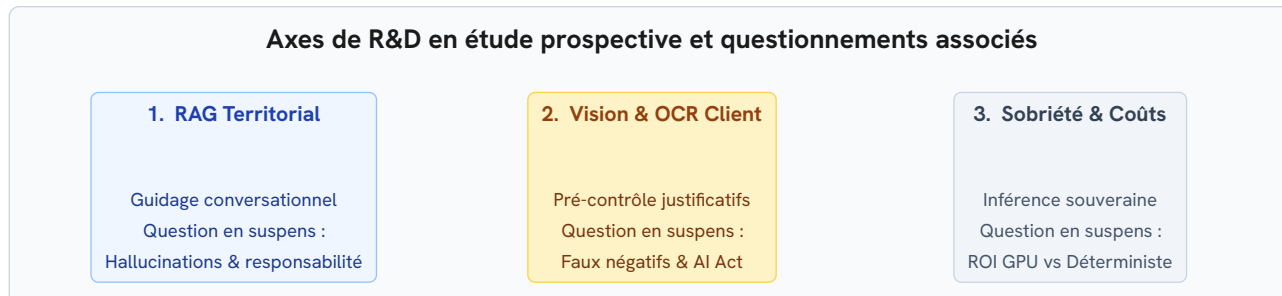
3.1.3. Prospective technologique : Étude exploratoire de l'IA et questions en suspens

3.1.3.1. Une démarche de R&D exploratoire et responsable

L'émergence rapide des technologies d'intelligence artificielle générative et de vision par ordinateur ouvre des opportunités de modernisation considérables pour la relation citoyenne. Au sein de Logitud

Solutions, l'intégration potentielle de ces briques au cœur de la plateforme **Mon-Guichet** fait actuellement l'objet d'une **étude prospective approfondie en R&D pour les versions futures**.

Toutefois, le déploiement d'intelligences artificielles dans le secteur public local ne peut être guidé par un simple effet de mode technologique : il impose une grille d'analyse lucide, mesurant les bénéfices attendus à l'aune des nombreuses questions techniques, juridiques, éthiques et financières qui demeurent aujourd'hui en suspens.



3.1.3.2. Piste 1 en étude : L'orientation citoyenne par RAG territorial

Le premier cas d'usage à l'étude concerne la mise en place d'un assistant d'orientation citoyenne s'appuyant sur une architecture de génération augmentée par récupération (**Retrieval-Augmented Generation** ou RAG). L'objectif serait de permettre aux administrés d'exprimer leurs besoins en langage naturel (« *Je viens d'emménager, comment inscrire mon enfant à la cantine et faire ma carte de stationnement ?* ») et de les orienter instantanément vers les démarches administratives appropriées.

Cependant, plusieurs verrous majeurs font l'objet d'arbitrages critiques :

- **Risque d'hallucination et engagement de la responsabilité publique** : Dans l'administration, une information erronée (ex: délais de recours contentieux, conditions de ressources erronées pour une aide sociale) engage directement la responsabilité juridique de la mairie. Comment garantir un taux d'exactitude factuelle de 100 % sur des corpus administratifs complexes ?
- **Souveraineté des données et conformité SecNumCloud** : Les requêtes citoyennes peuvent contenir des éléments de vie privée. Il est exclu d'utiliser des APIs d'acteurs commerciaux extra-européens. Le déploiement de modèles de langage **open-weights** (ex: Mistral AI) sur des infrastructures souveraines qualifiées SecNumCloud représente un coût d'inférence GPU élevé, posant la question de l'accessibilité financière pour les communes de taille intermédiaire.
- **Volatilité de la réglementation locale** : Les règlements intérieurs, tarifs municipaux et délibérations évoluent à chaque conseil municipal. Le maintien d'un index vectoriel parfaitement synchronisé sans introduire d'anachronismes documentaires constitue un défi d'ingénierie non résolu.

3.1.3.3. Piste 2 en étude : Le pré-contrôle des pièces justificatives par vision

Le second axe d'étude porte sur l'analyse automatisée des pièces justificatives téléversées par les usagers (justificatif de domicile, pièce d'identité, attestation de quotient familial) afin d'assister les agents communaux dans leur travail d'instruction quotidien.

Ce chantier soulève également d'importantes questions en suspens :

- **Inférence côté client (WebAssembly / On-Device) vs Inférence serveur** : L'exécution locale de modèles légers de vision par ordinateur via WebAssembly (Tesseract WASM) au sein du navigateur de l'administré présenterait l'avantage de préserver une confidentialité absolue (aucun document n'est envoyé à un serveur tiers avant validation) et d'économiser de la bande passante. En contrepartie, les disparités d'équipements (smartphones d'entrée de gamme) posent un risque d'échec d'exécution ou de ralentissement sévère pour l'utilisateur.
- **Risque de faux négatifs et rupture d'accès aux droits** : Le rejet algorithmique automatisé d'un document froissé ou mal cadré, mais pourtant parfaitement exploitable par un œil humain, risque de créer de la frustration et d'exclure les administrés les plus éloignés du numérique. La décision humaine de rejet doit impérativement rester la seule règle applicable.

- **Conformité avec l'AI Act européen** : L'Union Européenne classe les systèmes d'IA intervenant dans l'accès aux services publics et aux prestations sociales comme des systèmes à haut risque, imposant des obligations draconiennes d'auditabilité, d'évaluation des biais et de documentation technique.

3.1.3.4. Arbitrage fondamental : Sobriété numérique vs IA probabiliste

L'ensemble de ces réflexions prospectives conduit à une interrogation centrale d'architecture logicielle : *l'apport d'une brique d'IA probabiliste compense-t-il la complexité opérationnelle, la consommation énergétique et les coûts d'infrastructure par rapport à une automatisation déterministe rigoureuse ?*

Pour l'heure, Logitud Solutions privilégie la perfection ergonomique de ses formulaires guidés, la validation statique stricte des données de saisie et l'accessibilité universelle, tout en maintenant cette cellule de veille technologique active pour intégrer l'IA lorsque le cadre réglementaire, technique et économique offrira toutes les garanties de maturité.

3.2. Gestion des parties prenantes, négociation et conduite du changement

3.2.1. Collaboration interdisciplinaire et contrats d'interface avec le Back-End

Le développement d'applications d'envergure telles que le moteur de vente aquatique de Mulhouse ou le système de file d'attente PopQueue nécessite une coordination permanente entre des profils aux expertises variées.

La relation avec les développeurs back-end (spécialisés sous Laravel) a constitué un axe central de mon activité. Afin d'éviter les blocages réciproques, nous avons instauré une méthodologie de travail axée sur la co-définition des contrats d'interface (API) en amont des développements. En formalisant la structure exacte des données JSON échangées (ressources, réservations, statuts de paiement) dès la phase de conception, j'ai pu construire et tester les interfaces front-end à l'aide de données simulées (**mocks** typés sous TypeScript) pendant que l'équipe back-end implémentait la logique de base de données et les règles métiers. Cette approche a supprimé les temps d'attente intermédiaires et rendu l'intégration finale fluide et sans surprise.

Partie prenante	Nature des interactions et arbitrages
Lead Développeur Front	Validation des choix architecturaux, alignement sur les standards de code et revues de Pull Requests.
Équipe Back-End (Laravel)	Définition concertée des contrats d'API, gestion des formats d'erreurs et des flux temps réel Reverb.
Designer UI/UX (Figma)	Traduction des intentions graphiques en composants accessibles, conciliation des contraintes RGAA et d'ergonomie.
Chef de Projet / Métier	Priorisation des fonctionnalités dans Linear, respect des délais de livraison et démonstrations intermédiaires.
Agents communaux (retours)	Prise en compte des retours d'usage en situation réelle pour ajuster la densité d'information et les raccourcis.

Tableau 1. – Cartographie des parties prenantes et modalités de collaboration au quotidien

3.2.2. Conduite du changement et résolution de conflits : Le durcissement des standards de code

L'élévation des exigences de qualité logicielle s'accompagne inévitablement de défis organisationnels. L'un des moments charnières de mon rôle de référent technique sur le projet de billetterie a concerné le durcissement progressif de la gouvernance de code.

En concertation avec les leads techniques, j'ai progressivement introduit des règles de linting ESLint plus strictes, l'interdiction du type any, le tri automatique des imports et des plugins d'analyse statique poussés (typescript-eslint, règles de scoping de templates Vue). Initialement, cette démarche a suscité

des réticences de la part de plusieurs développeurs de l'équipe, qui percevaient ces contraintes comme un carcan rigide ralentissant leur rapidité d'écriture et bridant leur flexibilité.

Face à ces frictions, j'ai adopté une posture managériale constructive et pédagogique :

1. **Démonstration par la preuve** : Lors de sessions de cadrage technique, j'ai présenté des cas concrets de bugs détectés instantanément par l'IDE avant même l'exécution, évitant des heures de débogage ultérieur.
2. **Fluidification des revues de code** : J'ai démontré que l'automatisation du formatage et des règles de style libérait les revues de Pull Requests des débats de forme stériles pour se concentrer exclusivement sur la valeur métier et l'architecture.
3. **Accompagnement individuel** : J'ai assisté les développeurs dans la configuration optimale de leur environnement de travail (VS Code / WebStorm) pour que les corrections s'opèrent automatiquement à l'enregistrement du fichier.

Cette approche concertée a permis de désamorcer les craintes : l'équipe a unanimement reconnu le gain en sérénité, en cohérence globale et en lisibilité du code, faisant de cette rigueur une fierté partagée.

3.2.3. Arbitrage technique : Le cas du tri et du filtrage bimode Front / Back

Un autre arbitrage marquant a concerné l'architecture du composant `DataTable`. L'approche conventionnelle consistait à déléguer chaque tri ou filtre à l'API back-end. J'ai défendu et fait valider le mode bimode : pour les volumétries courantes, le traitement s'opère instantanément en mémoire côté navigateur (< 30 ms), procurant un confort exceptionnel aux agents tout en déchargeant significativement les serveurs applicatifs.

3.3. Impact mesurable et indicateurs de performance (KPIs)

Afin d'évaluer concrètement la création de valeur pour l'éditeur Logitud Solutions et pour les collectivités territoriales clientes, un ensemble d'indicateurs quantifiés a été suivi :

Indicateur Clé (KPI)	Valeur mesurée	Impact opérationnel et création de valeur
Volume d'affaires billetterie	150 000 €	Transactions en ligne traitées dès les premiers mois post-lancement à Mulhouse.
Digitalisation des achats	60 % en ligne	Désengorgement massif des caisses physiques des piscines dès le 1er trimestre.
Couverture de tests unitaires	85 % (Front) / 90 % (Back)	Sécurisation des parcours critiques et élimination quasi-totale des régressions.
Temps d'exécution CI/CD	3 minutes	Pipeline GitHub Actions complet avec déploiement d'environnements éphémères de PR.
Conformité légale RGAA	> 95 %	Score Lighthouse garantissant l'accessibilité universelle et l'éligibilité aux marchés publics.
Poids du bundle JS initial	-20 %	Optimisation par code splitting Nuxt 4 et chargement dynamique des dépendances.
Temps de tri / recherche table	< 30 ms	Fluidité instantanée pour les agents municipaux.
Gain de temps d'intégration	40 %	Réutilisation des briques partagées (Drawers, DataTables, Clavier tactile).

3.4. Bilan de compétences et posture managériale

3.4.1. Posture de référent technique, mentorat et transmission des savoirs

Au-delà de la production de code, mon rôle au cours de ce cycle de Mastère 2 s'est enrichi d'une véritable dimension de leadership technique et d'apprentissage organisationnel :

- **Formalisation des guides de contribution** : Rédaction sur Notion de la documentation d'architecture, des conventions de nommage TypeScript et des bonnes pratiques de composition de composants Vue 3.
- **Mentorat et accompagnement des pairs** : Animation d'ateliers de prise en main de Nuxt 4 et revues de code pédagogiques orientées sur la maintenabilité et l'accessibilité RGAA.
- **Veille technologique active** : Partage régulier avec l'équipe des évolutions de l'écosystème web (mises à jour Nuxt, RFCs Vue.js, nouvelles normes W3C).

Domaine de compétences	Acquis et réalisations concrètes en situation professionnelle
Compétences Techniques & Architecture	▪ Maîtrise avancée de l'écosystème Nuxt 4, Vue 3 (Composition API) et TypeScript strict. ▪ Conception d'architectures front-end modulaires et de bibliothèques de composants partagés. ▪ Implémentation de flux temps réel bidirectionnels via Laravel Reverb et WebSockets. ▪ Maîtrise des standards d'accessibilité numérique (RGAA 4.1.2 / WCAG 2.1 niveau AA). ▪ Intégration de bibliothèques graphiques complexes (Apache ECharts) et manipulation de diffs JSON.
Méthodologie, Qualité & DevOps	▪ Configuration de pipelines CI/CD sous GitHub Actions (tests Vitest, typage strict, déploiement d'environnements de test). ▪ Mise en place et gouvernance des règles de formatage et de typage statique (ESLint, Prettier, vue-tsc). ▪ Rédaction et automatisation de tests unitaires avec une couverture de 85 % sur le front-end. ▪ Maîtrise de la gestion de branches Git en équipe (GitFlow) et revues de code pédagogiques.
Posture Professionnelle & Soft Skills	▪ Capacité d'analyse stratégique, de négociation et de conduite du changement technique en équipe. ▪ Communication efficace avec des profils variés (développeurs, designers, chefs de projet, clients territoriaux). ▪ Leadership technique : formalisation de conventions de code, documentation Notion et partage de connaissances. ▪ Prise de recul critique sur l'expérience utilisateur et anticipation des contraintes d'exploitation à long terme.

3.4.2. Analyse critique de progression et perspectives d'expertise

Cette expérience prolongée de quatre années a confirmé ma vocation pour l'ingénierie front-end et l'architecture logicielle. Si mon autonomie technique est aujourd'hui solidement établie, j'identifie deux axes d'approfondissement majeurs pour la suite de mon parcours :

1. **L'élargissement des tests automatisés E2E (End-to-End)** : Si la couverture unitaire de 85 % sécurise la logique des composants isolés, l'automatisation systématique de scénarios de bout en bout (via Playwright) renforcera encore la fiabilité des tunnels d'achat face aux évolutions futures.

2. **L'exploration des architectures micro-frontends** : À mesure que le portefeuille applicatif de Logitud s'étend, l'adoption de la fédération de modules (**Module Federation**) offrira une indépendance de déploiement maximale entre les différentes gammes métiers.

4. Conclusion & Perspectives

4.1. Bilan global de quatre années d'immersion professionnelle

L'accomplissement de ces quatre années d'alternance au sein de l'entreprise Logitud Solutions — deux années de BUT MMI suivies de deux années de Mastère Expert en développement logiciel à Ynov Campus Strasbourg — représente une étape déterminante dans la construction de mon identité professionnelle.

Cette continuité rare au sein d'une même structure m'a offert l'opportunité d'observer et d'accompagner une transformation technologique majeure. D'une position initiale d'apprentissage centrée sur l'intégration HTML/CSS et la maintenance corrective, j'ai progressivement accédé à un niveau d'expertise me permettant de concevoir des architectures logicielles complètes, de piloter l'industrialisation des outillages de développement et d'influer directement sur les choix techniques structurants de l'entreprise.

Cette expérience m'a également permis d'acquérir une compréhension intime des spécificités du secteur public. Développer des logiciels pour les collectivités territoriales m'a appris que la qualité d'une interface ne se mesure pas uniquement à son élégance visuelle, mais avant tout à sa robustesse, à son accessibilité universelle pour tous les citoyens et à son efficacité au service du bien commun.

4.2. Perspectives d'évolution pour Logitud Solutions et la plateforme Mon-Guichet

L'écosystème **Mon-Guichet** et les nouveaux modules développés au cours de ces deux dernières années ouvrent des perspectives prometteuses pour Logitud Solutions :

- **Généralisation de la modularité applicative** : Le succès du moteur de billetterie déployé pour la Ville de Mulhouse (150 000 € de transactions dès les premiers mois) démontre la pertinence de l'architecture en modules indépendants et interconnectés. Ce modèle a vocation à être décliné sur d'autres services municipaux (réservation d'équipements sportifs et culturels, gestion des cantines scolaires, inscriptions événementielles).
- **Poursuite de l'engagement pour l'accessibilité** : Avec le renforcement continu des obligations légales pesant sur le secteur public, la démarche d'accessibilité native intégrée dans nos composants (RGAA 4.1.2 / WCAG 2.1 AA) constituera un atout concurrentiel majeur lors des futurs appels d'offres territoriaux.
- **Intégration d'outils d'assistance et d'automatisation** : L'évolution future de la plateforme s'orientera vers une assistance accrue aux agents communaux, notamment par la pré-validation automatique des pièces justificatives et l'optimisation des plannings de rendez-vous en mairie.

4.3. Projection et ambitions professionnelles personnelles

À l'issue de cette formation de Mastère 2, mon projet professionnel est clairement défini : je souhaite poursuivre ma carrière en tant que **Lead Développeur Front-End** ou **Architecte d'Applications Web** avec une approche centrée sur l'accessibilité et le DevSecOps.

Mon ambition est de continuer à concevoir des architectures front-end modernes, exigeantes et pérennes, en alliant la rigueur du typage statique, l'efficacité des méthodes DevOps et la créativité dans la conception d'interfaces utilisateur intuitives. Cette alternance de quatre ans m'a apporté la maturité technique, méthodologique et relationnelle indispensable pour aborder avec confiance et détermination les défis de l'ingénierie logicielle de demain.

Webographie & Références Techniques

1. **Documentation officielle Nuxt 4 :**
<https://nuxt.com/docs> — Guide d'architecture, Composition API, SSR et structure applicative.
2. **Documentation officielle Nuxt UI v4 & Tailwind CSS :**
<https://ui.nuxt.com> & <https://tailwindcss.com/docs> — Système de composants et utilitaires CSS.
3. **TypeScript Handbook & Strict Mode Guidelines :**
<https://www.typescriptlang.org/docs/handbook> — Typage statique, inférence et sécurité de code.
4. **Référentiel Général d'Amélioration de l'Accessibilité (RGAA 4.1.2) :**
<https://accessibilite.numerique.gouv.fr> — Critères et règles de conformité légale pour le secteur public français.
5. **Web Content Accessibility Guidelines (WCAG 2.1) - W3C :**
<https://www.w3.org/TR/WCAG21> — Standards internationaux d'accessibilité numérique de niveau AA.
6. **Laravel Reverb Documentation :**
<https://laravel.com/docs/reverb> — Serveur de WebSockets temps réel haute performance pour l'écosystème Laravel.
7. **Vitest - Next Generation Testing Framework :**
<https://vitest.dev/guide> — Framework de tests unitaires pour l'écosystème Vue/Nuxt.
8. **Apache ECharts Documentation & Examples :**
<https://echarts.apache.org/en/index.html> — Visualisation de données réactive et interactive.
9. **Plateforme Logitud Solutions :**
<https://www.logitud.fr> — Présentation des gammes logicielles et solutions pour collectivités territoriales.

Annexes : Environnement Technique & Référentiel RNCP

Annexe A : Matrice de l'Environnement Technique

Catégorie	Technologies & Outils	Rôle et domaine d'application
Frameworks & UI	Nuxt 4, Vue 3, Nuxt UI v4, Tailwind CSS	Socle applicatif moderne, Composition API et design system unifié
Langages & Typage	TypeScript (mode strict), JavaScript ESNext	Sécurité de typage statique de bout en bout et logique front
Composants avancés	Headless UI, Apache ECharts, Diff JSON	Composants accessibles, visualisations et audit
Temps réel & API	Laravel Reverb, WebSockets (WS), REST / JSON	Synchronisation instantanée des files d'attente et communication back-end
Qualité & Tests	ESLint, Prettier, Vitest (85% cov), vue-tsc	Formatage, détection d'erreurs statiques et tests unitaires
DevOps & Outils	GitHub Actions (CI 3 min), Docker, Linear, Figma, Notion	Intégration continue, environnements de test éphémères et gestion agile

Annexe B : Couverture des Compétences du Titre RNCP Niveau 7

Bloc de compétences RNCP	Démonstration et réalisations dans le cadre professionnel
1. Analyse des besoins & Architecture logicielle	Conception de l'architecture modulaire Nuxt 4 / TypeScript strict pour le moteur de billetterie et le système temps réel PopQueue.
2. Développement & Sécurisation applicative	Implémentation de composants accessibles (RGAA 4.1.2 / WCAG 2.1 AA), gestion transactionnelle PayFIP, intercepteurs API et DevSecOps.
3. Industrialisation, Qualité & DevOps	Configuration des pipelines GitHub Actions avec déploiement d'environnements éphémères de PR, couverture de tests à 85 % sous Vitest.
4. Gestion de projet, Collaboration & Conduite du changement	Arbitrages techniques concertés, durcissement des standards ESLint, co-définition des contrats d'API avec les équipes Laravel et mentorat.

Merci de votre lecture.

Candidat : Nicolas Walter

Évaluation : Rapport d'expérience professionnelle

Certification : Mastère 2 Expert en développement logiciel

Ce document est open-source et peut être consulté sur [GitHub](#).

Il a été compilé avec le moteur de composition [Typst](#), gratuit et open-source sous licence Apache 2.0.