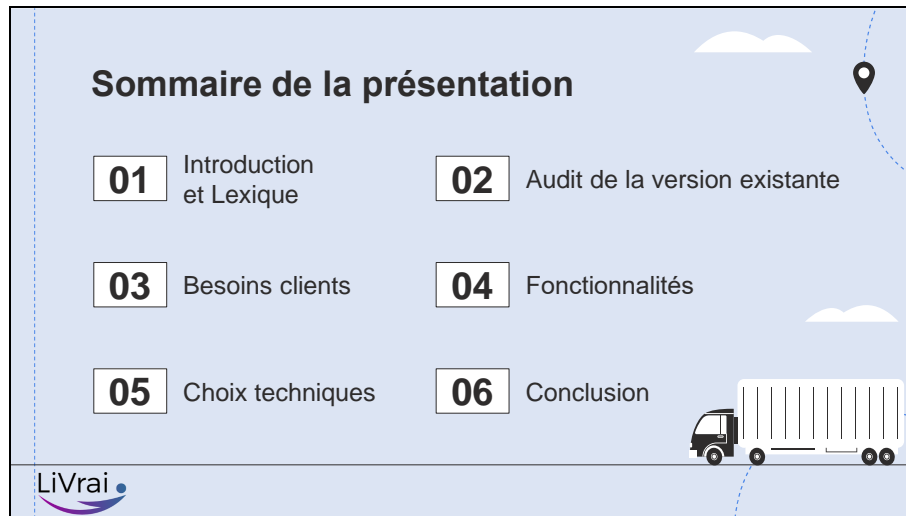




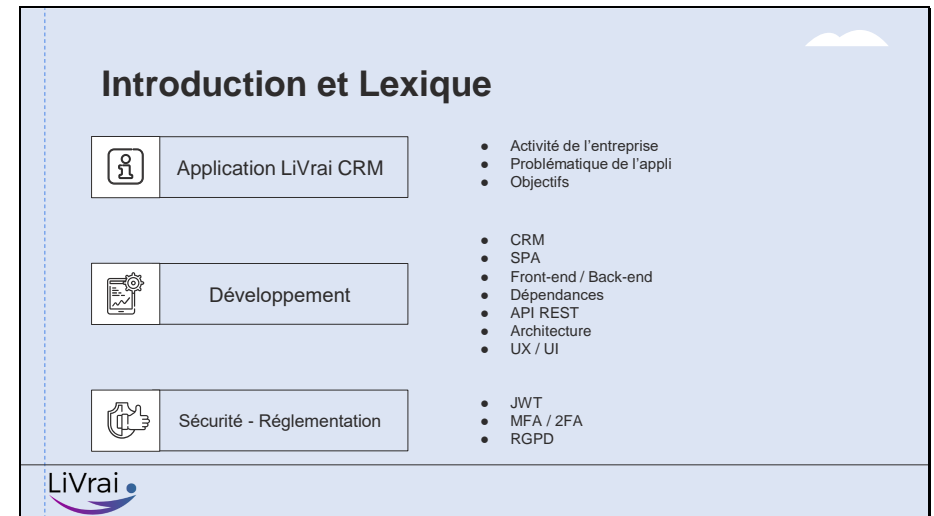
## Présentation du projet

Cahier des charges fonctionnel et technique  
pour une application CRM de service de livraison.

Réponse à un appel d'offre pour la refonte de l'application de l'entreprise LiVrai.



- 1- Introduction rapide de l'entreprise et rappel des termes techniques.
- 2- Compte-rendu de l'audit de la version existante.
- 3- Rappel des besoins décrits dans la fiche de mission.
- 4- Cahier des charges fonctionnel.
- 5- Cahier des charges technique.
- 6- Remarques, propositions et conclusion



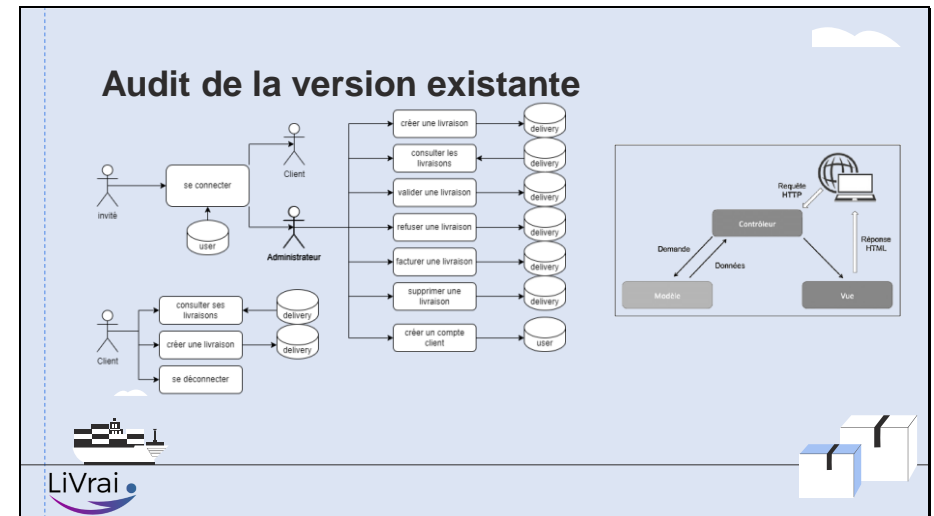
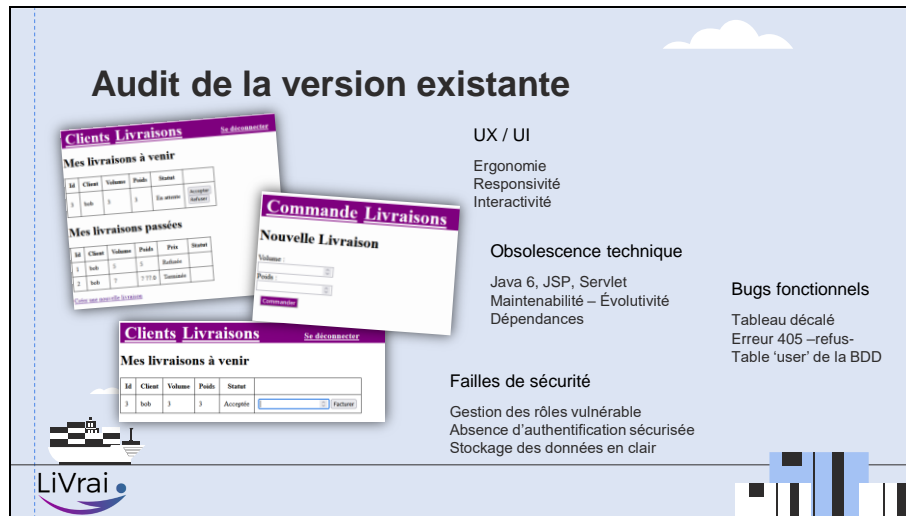
**La société LiVrai : activité**

**Application CRM actuelle : problématique**

**Objectifs** du développement d'une nouvelle application :

- autonomie client
- libération de charge administrative des commerciaux
- Accès aux livraisons et à la facturation par le service logistique.

**Lexique**

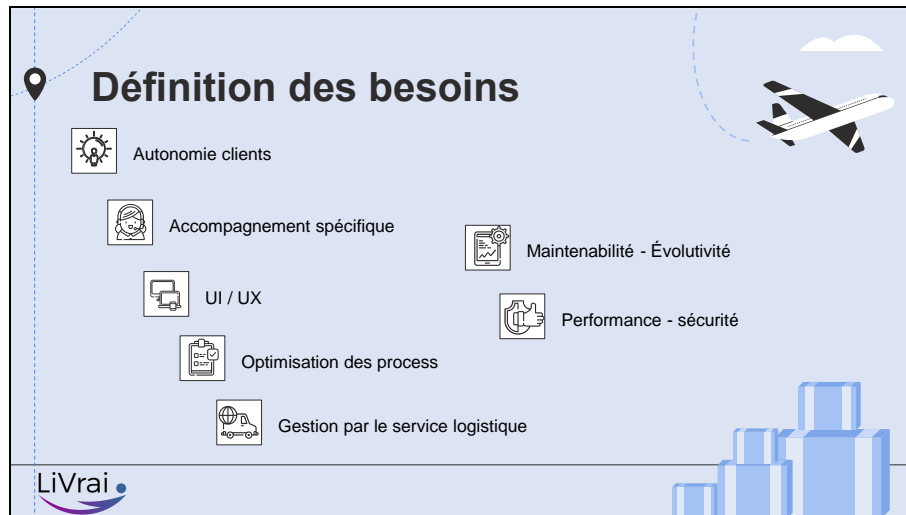


## Déficiences :

- **Problèmes UX/UI** : Ergonomie faible, non responsive, interactions limitées.
- **Déficiences techniques** : Technologie obsolète (Java 6, JSP, Servlet), difficultés de maintenance et évolutivité.
- **Faibles de sécurité** : Mots de passe stockés en clair, gestion des privilèges vulnérable.
- **Bugs majeurs** : Anomalies sur le suivi des livraisons et erreurs récurrentes.

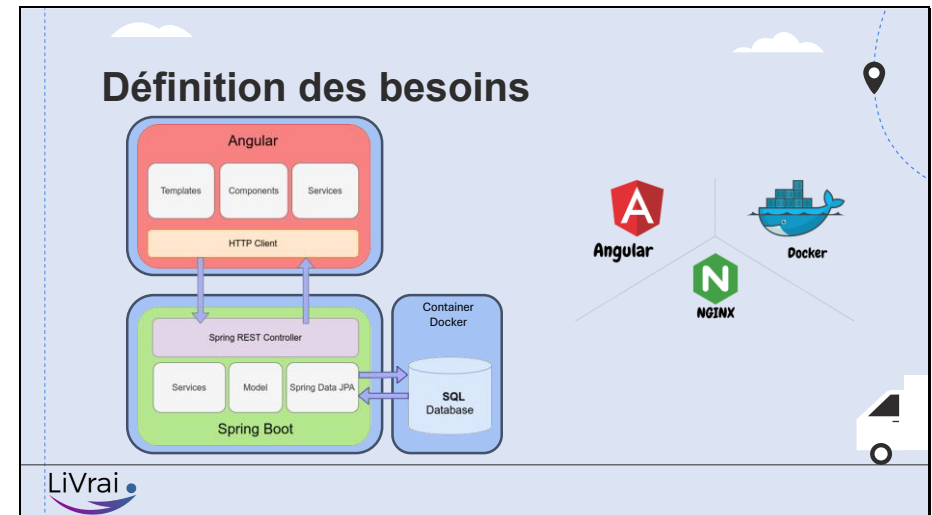
## Points forts :

- **Sécurité** : protection contre les injections à niveau des années 2000.
- **Stabilité - Robustesse** : Technologie standard.
- **Structure – cas d'usage** : identification des scénarios clés.
- **Architecture** : DAO-JSP-Servlet (MVC).



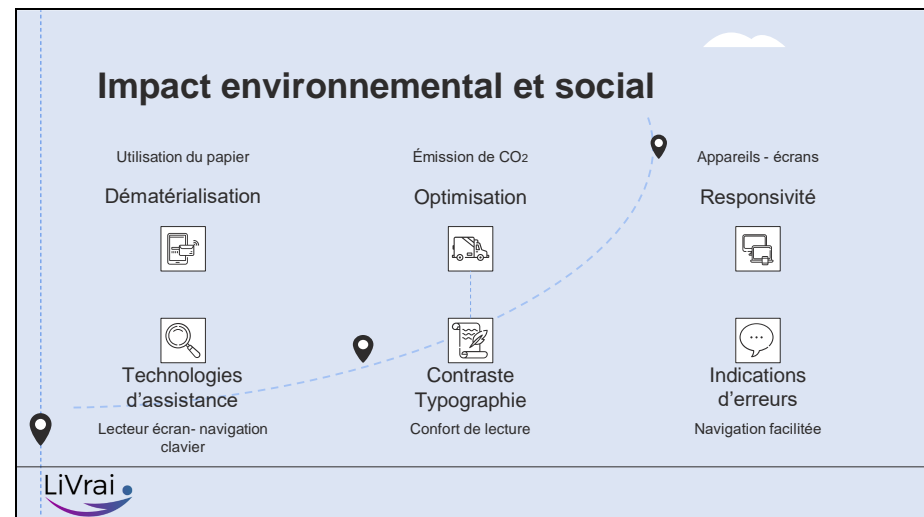
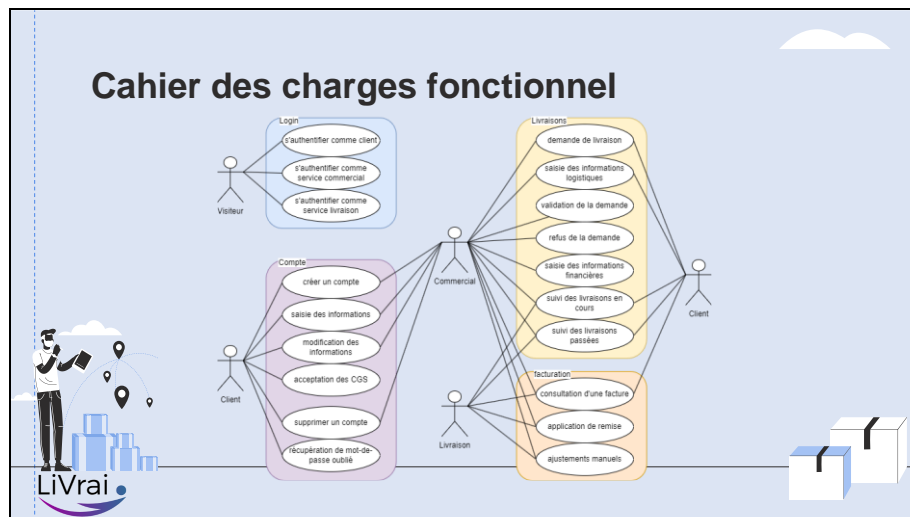
### Synthèse des besoins :

- **Clients** : Création et gestion de comptes en autonomie.
- **Commerciaux** : Intervention ciblée pour certains clients.
- **Interface** : Amélioration UI/UX.
- **Process** : optimisations, automatisation.
- **Logistique** : Suivi et gestion des livraisons optimisé.
- **Administration** : Automatisation et fluidification de la facturation.
- **Développement** : Maintenabilité, évolutivité, performance, sécurité.



### Synthèse des besoins :

- **Front-end** : SPA Angular.
- **Back-end** : Java 21 LTS.
- **Docker** : Déploiement.

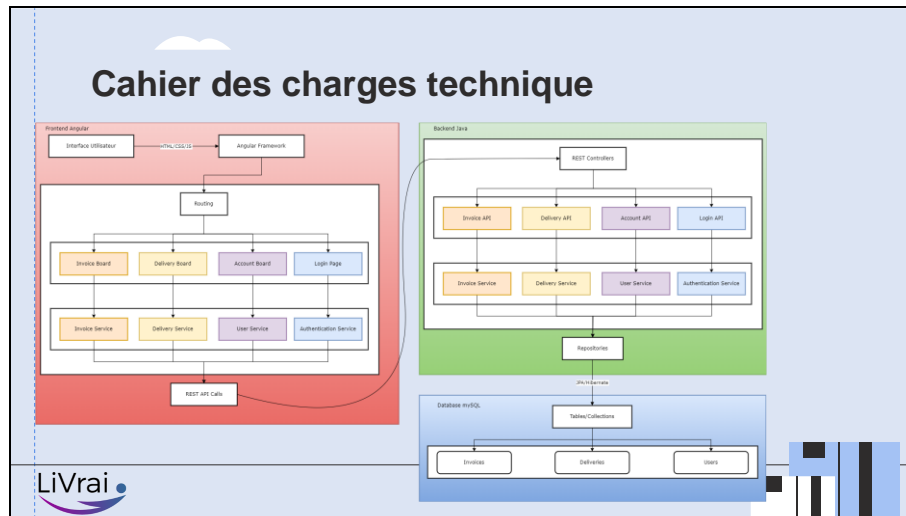


## Cahier des charges fonctionnel

- **Authentification et sécurité** : Gestion des rôles et permissions, JWT, RGPD.
- **Comptes clients** : Inscription, modification, gestion des profils.
- **Gestion des livraisons** : Suivi, automatisation, validation, historique.
- **Facturation** : Création, modification, consultation des factures.

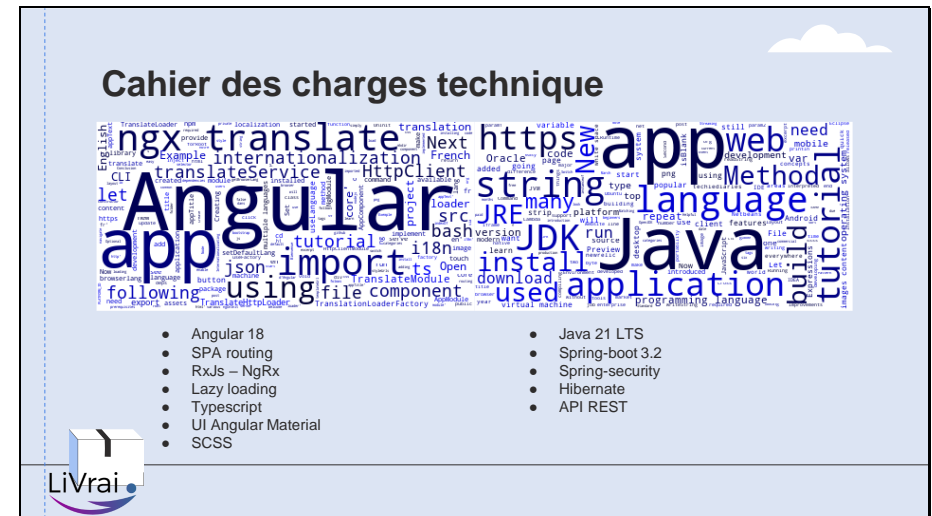
## Impacts environnementaux et sociétaux :

- **Réduction papier** : Dématérialisation des documents.
- **Optimisation des trajets** : Réduction des émissions de CO2.
- **Responsivité** : Usage sur tous types d'appareil/ toutes tailles d'écran.
- **Accessibilité WCAG 2.1** : Conception inclusive pour tous les utilisateurs.



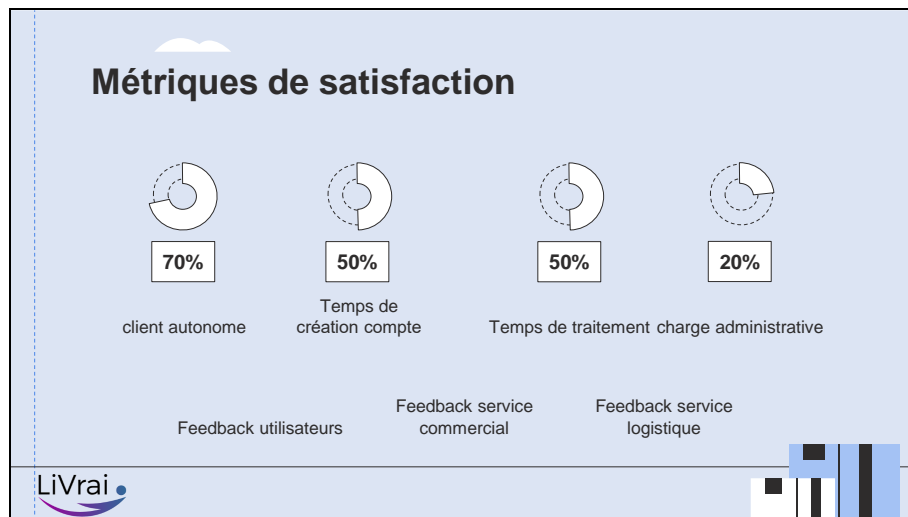
## Architecture du projet

- **MVC et API REST** : Modélisation modulaire pour évolutivité et maintenabilité.
- **Conteneurisation Docker** : Isolation des services, sécurité et mise à jour simplifiée.



## Cahier des charges technique

- **Front-end** : Angular 18, SPA, responsive, UI Material Design.
- **Back-end** : Java 21 LTS, Spring Boot 3.2, API REST.
- **Base de données** : mySQL (PostgreSQL).



## Indicateurs de réussite (6 mois)

- **Clients** autonomes sur le portail **>70%**.
- Temps de création de **compte -50%**.
- Temps de traitement de prise de **commande -50%**.
- Charge administrative des **commerciaux -20%**.
- **Feedbacks**.

## Conclusion

- **Solution adaptable et scalable** : évolutions futures (signature électronique, optimisation IA).
- **Engagement sur la qualité et la performance**.