

# Architecture Technique – MVP Système de Gestion Hospitalière

## Objectif du document

Ce document présente l'architecture technique retenue pour le MVP du système de gestion hospitalière. L'architecture est volontairement simple, robuste et évolutive afin de permettre une implémentation rapide en 7 jours tout en servant de base solide pour les évolutions futures.

## 1. Principes d'architecture

- Simplicité : limiter la complexité technique au strict nécessaire.
- Lisibilité : architecture compréhensible par toute l'équipe.
- Stabilité : choix de technologies éprouvées.
- Évolutivité : possibilité d'ajouter des fonctionnalités sans refonte.

## 2. Architecture globale

L'architecture repose sur un modèle **client–serveur monolithique** en réseau local (LAN). Les clients desktop communiquent avec un serveur central via une API REST.

Couche	Technologies	Rôle
Client	Electron + React + TypeScript	Interface utilisateur par rôle
Serveur	Node.js + NestJS	Logique métier et API REST
Données	PostgreSQL + Prisma ORM	Stockage des données relationnelles

## 3. Architecture Backend

Le backend est implanté sous forme de monolithe NestJS structuré par modules fonctionnels. Chaque module correspond à une responsabilité métier unique.

- Auth : authentification et gestion des rôles.
- Users : gestion des utilisateurs.
- Patients : création et consultation des patients.
- Appointments : gestion des rendez-vous.
- Prescriptions : création des prescriptions médicales.
- Results : saisie et consultation des résultats de laboratoire.

## 4. Architecture Frontend

Le frontend est une application Electron basée sur React et TypeScript. L'interface est organisée par rôle utilisateur afin de refléter les responsabilités métier.

- Une interface dédiée par rôle (médecin, biologiste, secrétariat).
- Pages orientées actions métier (prescrire, saisir résultat).
- Communication avec le backend via API REST.

## 5. Architecture des données

La base de données PostgreSQL est structurée autour d'un modèle relationnel simple. Les relations garantissent l'intégrité des données médicales.

- User
- Patient
- Appointment
- Prescription
- Result

## 6. Sécurité – Niveau MVP

- Mots de passe chiffrés avec bcrypt.
- Contrôle d'accès basé sur les rôles.
- Accès limité au réseau local.
- Sécurité avancée reportée aux phases ultérieures.

## 7. Évolutions prévues post-MVP

- Ajout de notifications temps réel (WebSocket).
- Audit et traçabilité des accès.
- Gestion avancée des documents médicaux.
- Renforcement de la sécurité et conformité RGPD.