Architecture Decision Records (ADRs)

Système de gestion de parking - Flutter + Spring Boot

ADR-001 : Application Mobile et Web Flutter

Status: Accepté

Décision

Justification du choix mobile-first pour la gestion de parking

Avantages

- QR code natif
- Notifications push
- GPS
- Performance native
- Cross-platform avec un seul codebase

ADR-002: Clean Architecture + DDD

Status: Accepté

Décision

Adoption de Clean Architecture avec Domain Driven Design

Avantages

- Séparation claire des responsabilités
- Business logic indépendante du framework
- Testabilité et maintenabilité améliorées

ADR-003: Stack Technique Flutter + Spring Boot

Status: Accepté

Stack retenue

- **Frontend :** Flutter/Dart pour performance native mobile
- **Backend :** Java 17+ Spring Boot 3.x pour robustesse enterprise
- **Base de données :** MySQL
- **Message Queue :** RabbitMQ pour fiabilité
- **Cache :** Redis pour performance

ADR-004: Authentification JWT + Spring Security + Flutter Secure Storage

Status : Accepté

Solution d'authentification

- Spring Security pour le backend (battle-tested)
- Flutter Secure Storage pour sécurité mobile (Keychain/Keystore)
- Support biométrie native

ADR-005: Modular Monolith

Status: Accepté

Architecture

- Single database avec séparation par domaines
- Simplicité opérationnelle vs complexité microservices
- ACID transactions cross-domain

ADR-006: BLoC Pattern + Repository Pattern

Status: Accepté

Patterns Flutter

- State management prévisible pour Flutter
- Séparation UI/Business Logic
- Architecture testable et réactive