

# **Architecture Technique du Système de Cartographie**

## **Stack Technologique**

### **Backend et Traitement Bas Niveau**

- **Langage Principal:** C++
- **Frameworks et Bibliothèques**
  - OpenCV (Vision par ordinateur)
  - OpenGL/Vulkan (Rendu graphique)
  - ARCore/ARKit (Réalité augmentée)
  - Eigen (Calculs mathématiques)
  - Boost (Utilitaires C++)

### **Intelligence Artificielle et Machine Learning**

- **Langage:** Python
- **Frameworks**
  - TensorFlow
  - PyTorch
  - Scikit-learn
  - NumPy
  - Pandas
- **Cas d'Usage**
  - Modèles prédictifs
  - Traitement des données
  - Entraînement des algorithmes
  - Analyse géospatiale

## **Frontend Mobile**

- **Technologie:** React Native
- **Langage:** TypeScript
- **Bibliothèques**
  - React Navigation
  - Redux/MobX (Gestion d'état)
  - Expo (Outils de développement)
  - React Native Vision Camera
  - React Native AR

## **Architecture Logicielle**

### **Composants C++**

## 1. Moteur de Réalité Augmentée

- Tracking de position
- Reconstruction 3D
- Détection d'objets
- Rendu graphique

## 2. Processeur de Vision par Ordinateur

- Analyse d'images
- Segmentation
- Reconnaissance d'objets

## **Composants Python**

### 1. Services d'IA

- Modèles de prédiction
- Analyse de données
- Apprentissage automatique

### 2. Backend de Traitement

- Prétraitement des données
- Entraînement des modèles
- API de services IA

## **Composants React Native**

## 1. Interface Utilisateur

- Écrans de navigation
- Visualisation de données
- Interactions RA

## 2. Pont de Communication

- Appels aux services C++
- Requêtes aux modèles Python
- Gestion des états

## Flux de Données

 Copier

```
[Capteurs] → C++ (Traitement Bas Niveau)  
           → Python (Analyse IA)  
           → React Native (Visualisation)
```

## Défis Techniques

- Intégration entre C++, Python et React Native
- Performances et latence
- Synchronisation des données
- Compatibilité multiplateforme

## Recommandations d'Implémentation

- Utiliser gRPC ou ZeroMQ pour communication inter-langages
- Conteneurisation avec Docker
- Déploiement cloud ou edge computing
- Gestion des versions avec Git