

# CANEVAS TECHNIQUE

## CONSEILS DE RÉDACTION

- Langage : clair, professionnel, sans jargon inutile.
- Style : ton institutionnel, orienté résultats.
- Inclure : tableaux, schémas, encadrés explicatifs pour vulgariser la technique.
- Format final : .docx + version PDF pour diffusion.

[ TITRE ]

## RAPPORT TECHNIQUE DE DÉVELOPPEMENT

**Plateforme de Synchronisation Totale entre Instances DHIS2**

### Page de garde

- **Titre complet du rapport :**

*RAPPORT TECHNIQUE DE DÉVELOPPEMENT — Plateforme de Synchronisation Totale entre Instances DHIS2*

- **Auteur / équipe :** (Nom de l'ingénieur, Équipe E-Santé)
- **Date et période :** (ex. Octobre 2025)
- **Logos :** Logo Santé Intégré
- **Confidentialité :** “Document interne à la Direction SEAQ – Diffusion restreinte”

# Résumé exécutif (1 page maximum)

## Objectif :

Donner une vision claire et stratégique du projet pour les décideurs.

## À inclure :

- Objectif global du développement (répliquer intégralement une instance DHIS2 source vers une destination vierge, manuellement ou automatiquement).
- Importance stratégique (interopérabilité, sauvegarde, duplication, mise à l'échelle DHIS2).
- Résultats attendus (solution robuste, flexible, automatisée).
- Technologies principales utilisées.
- Valeur ajoutée pour le système de santé numérique.

## 1. Contexte et justification

### Contenu attendu :

- Contexte général du projet dans la stratégie nationale e-Santé (ex : interopérabilité, fiabilité des données, duplication d'environnement).
- Problèmes observés avant la solution (copie manuelle longue, erreurs humaines, perte d'intégrité, non-compatibilité des versions).
- Raison du développement de cette plateforme (besoin de duplication rapide d'instances DHIS2 pour tests, formation ou extension).
- Acteurs concernés (équipe E-Santé, SE, partenaires techniques).

## 2. Objectif général et objectifs spécifiques

### Objectif général :

Créer une application capable de **synchroniser dynamiquement et intégralement** deux instances DHIS2 (source et destination), couvrant toutes les composantes de données et de métadonnées.

## Objectifs spécifiques :

- Synchroniser automatiquement et manuellement tout le contenu d'une instance DHIS2 source vers une destination.
- Garantir la cohérence, l'intégrité et la traçabilité des données transférées.
- Gérer la compatibilité entre différentes versions DHIS2.
- Offrir une supervision complète des processus (progression, logs, alertes).
- Permettre la planification ou l'exécution à la demande des synchronisations.
- Faciliter la duplication intégrale d'une instance pour des fins de **sauvegarde, migration ou réPLICATION**.

## 3. Description générale de la solution

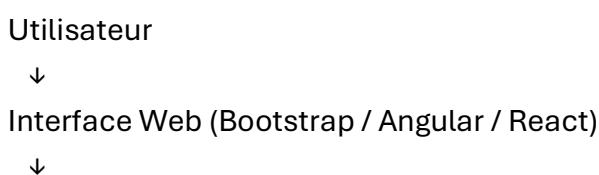
### À inclure :

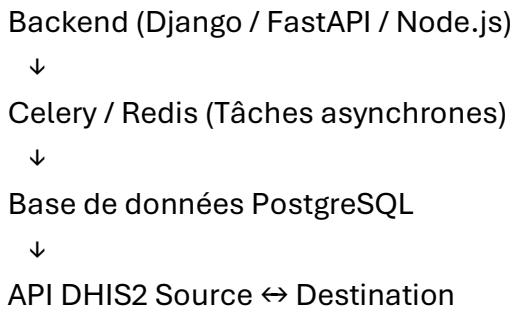
- Présentation fonctionnelle du système (rôle de la source et de la destination).
- Type de synchronisation supportée :
  - Métadonnées (structures, catégories, programmes, etc.)
  - Données (événementielles, agrégées, tracker)
  - Objets analytiques (tableaux de bord, visualisations, cartes, etc.)
- Fonctionnement général :
  - Mode manuel : synchronisation à la demande.
  - Mode automatique : exécution planifiée ou déclenchée par détection de changement.
- Exemple d'utilisation : duplication complète d'une instance nationale vers une instance régionale pour formation ou tests.

## 4. Architecture technique

### 4.1. Présentation globale

Inclure un schéma ou diagramme (exemple ci-dessous) :





## 4.2. Couches techniques

- **Frontend** : technologies choisies (Bootstrap, React, Angular...) et justification (ergonomie, simplicité).
- **Backend** : framework (Django/Flask/FastAPI), logique métier, sécurité, API REST.
- **Base de données** : PostgreSQL (structuration des logs, des configurations et des historiques).
- **Tâches asynchrones** : Celery + Redis (exécution automatique sans bloquer le serveur).
- **Intégration DHIS2** : utilisation de dhis2.py ou API REST officielle.

## 4.3. Sécurité et accès

- Authentification des instances DHIS2.
- Gestion des accès à la plateforme (administrateur, utilisateur simple).
- Cryptage des mots de passe et des tokens DHIS2.

# 5. Fonctionnalités principales

## 5.1. Configuration des instances

- Ajout, suppression et test de connexion DHIS2.
- Enregistrement des paramètres (URL, version, identifiants).
- Détection automatique de la version DHIS2.

## 5.2. Types de synchronisation

- **Métadonnées** : structures, programmes, options, indicateurs, data elements, datasets.

- **Données** : agrégées, tracker, événements.
- **Objets analytiques** : visualisations, dashboards, cartes, rapports.
- **Système** : attributs, paramètres système, constantes.

### 5.3. Modes d'exécution

- **Manuel** : déclenchement à la demande par un administrateur.
- **Automatique** : synchronisation planifiée ou déclenchée par détection de changements (via lastUpdated).

### 5.4. Orchestration de synchronisation

- Exécution dans un ordre hiérarchique (métadonnées avant données).
- Gestion des dépendances entre objets.
- Suivi d'état : pending, running, completed, failed.
- Gestion des erreurs avec “retry automatique” et journalisation.

### 5.5. Monitoring et logs

- Tableau de bord temps réel.
- Visualisation des synchronisations en cours.
- Historique complet avec logs horodatés et codes d'erreur.

## 6. Interface utilisateur (UX/UI)

- Capture(s) ou description des écrans :
  - **Accueil** : résumé des synchronisations et instances actives.
  - **Configuration** : formulaire de paramétrage des instances.
  - **Synchronisation** : bouton de lancement, suivi de progression, barre d'état.
  - **Logs** : affichage détaillé des erreurs et succès.
  - **Administration** : gestion des accès et supervision.
- Principes : design épuré, intuitif, compatible mobile, multilingue (si prévu).

## **7. Tests et validation**

### **Types de tests :**

- Test de connectivité DHIS2.
- Test de compatibilité inter-version.
- Test de synchronisation complète (temps, volume, erreurs).
- Test de robustesse (interruption réseau, reprise, timeout).

### **Indicateurs de performance :**

- Temps moyen de synchronisation.
- Taux d'erreurs ou de retries.
- Volume de données transféré avec succès.

### **Résultats attendus :**

- Duplication fidèle et complète d'une instance source vers destination.
- Aucun échec d'intégrité.
- Journalisation complète de tous les processus.

## **8. Sécurité, conformité et traçabilité**

- Gestion des identifiants sensibles (chiffrement AES/Hash).
- Audit trail : historique des actions utilisateur.
- Conformité aux standards DHIS2 API et sécurité serveur (HTTPS, tokens).
- Protection contre les interruptions réseau et erreurs de synchronisation.

## **9. Perspectives et évolutions**

- Intégration avec d'autres systèmes d'information (OpenHIE, FHIR, SmartCare, etc.).
- Extension vers des synchronisations sélectives par module ou orgUnit.
- Ajout de notifications (emails, Slack, SMS).
- Support multi-instance (plus de 2 DHIS2 simultanément).
- Monitoring avancé via tableaux de bord analytiques.

## **10. Conclusion**

- Rappel des objectifs atteints.
- Importance stratégique pour la transformation numérique du système de santé.
- Vision d'avenir (interopérabilité régionale, durabilité technique).

## **Annexes**

Inclure :

- Extraits de code commentés (services de synchronisation, classes principales).
- Diagramme d'architecture.
- Exemples de logs réels.
- Paramètres de configuration utilisés.
- Table des dépendances DHIS2 (ordre de synchronisation).