# Cahier des Charges - Système d'Information Géographique (SIG) BADGE

## **1. Introduction**

La Base de Données Géographique Environnementale nationale (BADGE) est un système d'information géographique destiné à centraliser, analyser et diffuser efficacement des données géospatiales environnementales. L’objectif est d’ améliorer l'accès aux informations environnementales pour une prise de décision efficace et une meilleure gestion des ressources naturelles.

Le projet BADGE s'inscrit dans une robuste démarche de modernisation et de mutualisation des données géographiques, favorisant la transparence et la collaboration entre les différents acteurs du domaine environnemental. Il repose sur une architecture modulaire, scalable et sécurisée, garantissant une performance optimale et une haute disponibilité des données .

**2. Bénéficiaires**

La BADGE est destinée à plusieurs catégories d’utilisateurs, chacune ayant des besoins spécifiques clairement définis pour optimiser leur interaction avec le système.

### **2.1 Utilisateurs Primaires**

* **Gestionnaires de projet** : accès intégral à la base de données avec droits étendus d'édition, d'analyse et de gestion quotidienne ;
* **Partenaires institutionnels** : consultation régulière, génération de rapports et téléchargement sélectif des données ;
* **Décideurs politiques** : accès à des tableaux de bord synthétiques actualisés périodiquement et rapports décisionnels automatisés ;

### **2.2 Utilisateurs Secondaires**

* **Bailleurs de fonds** : suivi des projets, indicateurs, analyse des performances
* **Chercheurs et étudiants** : téléchargement de données, analyses spatiales complexes, création de requêtes avancées ;
* **Grand public** : consultation simplifiée via cartes interactives, facilitant la compréhension des enjeux environnementaux ;

## 

## **3. Objectifs**

La BADGE répond à des objectifs précis, mesurables, assurant performance et fiabilité à long terme. En effet, elle vise à faciliter la prise de décisions stratégiques en matière de gestion durable des ressources naturelles, tout en renforçant la collaboration et la transparence entre acteurs institutionnels et privés.

### **3.1 Objectifs Spécifiques**

* Centralisation des données : Intégrer 100 % des données géospatiales environnementales nationales disponibles ;
* Performance d'accès aux données :
* Latence API inférieure à 100 ms pour les requêtes standards Garantir une disponibilité globale du système à 99,9 %
  + Assurer une reprise rapide du service en cas d'incident majeur
* Compatibilité et interopérabilité :
  + Support complet des standards internationaux tels que WMS, WFS, WCS et GeoTIFF
  + Développement **100 % API-first** pour faciliter l’intégration avec des systèmes externes (FAO, NASA, ESA, plateformes Open Data).
* Gestion efficace des utilisateurs :
  + Gérer jusqu'à 1000 utilisateurs simultanés avec montée en charge automatique
  + Sécuriser les accès selon des rôles et permissions précis grâce à JWT/OAuth2
* Actualisation des données :
  + Mise à jour des cartes et données en temps réel
  + Synchronisation régulière et automatisée des sources externes
* Analyse avancée, intelligence artificielle (IA) et reporting automatisé :
* Intégration de **modèles prédictifs** avec **TensorFlow ou PyTorch**.
* Utilisation de **serveurs GPU dédiés** pour l'entraînement et le scoring en temps réel.
  + Automatiser la production de rapports et de tableaux de bord personnalisables
* Documentation exhaustive et multilingue :
  + Fournir une documentation technique complète et détaillée (avec schémas et exemples)
  + Proposer des tutoriels interactifs et une aide contextuelle en plusieurs langues (FR, EN)
* **Portail** : Vitrine de tous les Systèmes d'Information du Centre de Suivi Écologique de Dakar (CSE), intégration avec les plateformes open data

## **4. Fonctionnalités**

Le système propose des fonctionnalités avancées, conçues pour maximiser l’efficacité et l’expérience utilisateur.

### **4.1 Catalogue des données**

* Visualisation interactive avec outils de filtrage
* Système de versionnage des données
* Métadonnées conformes à ISO 19115 et INSPIRE
* Formats d'export : Shapefile, GeoJSON, KML, CSV
* Prise en charge des projections cartographiques multiples
* Recherche avancée par attributs et emprise géographique
* Gestion des droits granulaire avec niveaux d'accès différenciés

### **4.2 API d'accès**

* REST API
* Notifications et mises à jour
* Documentation OpenAPI avec exemples interactifs
* Monitoring en temps réel des appels API
* Authentication JWT et OAuth2 pour sécurisation des accès
* Accès différencié selon le rôle de l'utilisateur

### **4.3 Connexion à des bases de données externes**

* Cache intelligent pour accélérer les requêtes fréquentes
* Synchronisation programmable et automatique des sources externes
* Validation automatique des données entrantes avec journalisation
* Gestion des conflits en cas de mises à jour concurrentes
* Sources : FAO, DOPA, FIRMS, CSE, E-STATION, NASA, ESA, NOAA, données locales et régionales
* Historique des synchronisations avec logs détaillés

### **4.4 Cartographie interactive**

* Mode en ligne/hors ligne avec synchronisation différée
* Édition collaborative avec gestion avancée des versions
* Historique des modifications avec possibilité de restauration
* Visualisation 3D des reliefs et bâtiments
* Intégration d'images satellites en temps réel (Sentinel, Landsat)
* Annotations collaboratives et outils de dessin avancés
* Export haute résolution pour impressions et publications

### **4.5 Analyse et Rapports**

* Rapports automatisés avec génération de graphiques
* Analyse prédictive basée sur des modèles statistiques et IA
* Tableaux de bord personnalisables avec widgets dynamiques
* Export multi-formats : PDF, CSV, Excel, JSON
* Alertes configurables pour indicateurs clés et anomalies

## **5. Architecture Technique**

### **5.1 Frontend**

* HTML5, CSS3, JavaScript (ES6+)
* Framework : Angular, avec Angular Material pour une interface plus ergonomique.
* Bibliothèque cartographique : Leaflet.js, OpenLayers
* Bootstrap et Tailwind CSS pour un design responsive et adaptatif

### **5.2 Backend**

* Décomposition en **microservices** pour une scalabilité optimale :
* Framework : Django / Django Rest Framework
* Base de données : PostgreSQL avec extension PostGIS
* Serveur cartographique : GeoServer pour la diffusion des données
* Moteur d’indexation spatiale : Elasticsearch pour recherche rapide
* File d’attente : Celery pour le traitement asynchrone
* Authentification et autorisation : Auth0 pour gestion des accès

## **6. Sécurité**

* Gestion des accès par rôles et permissions
* Chiffrement des communications (SSL/TLS)
* Sauvegardes automatiques et redondance des données
* Détection et prévention des intrusions (IDS/IPS)

## **7. Performance**

### **7.1 Objectifs**

* Latence API < 100ms pour les requêtes standards
* Disponibilité 99.9% avec reprise après sinistre
* Support de 1000+ utilisateurs simultanés avec montée en charge automatique

### **7.2 Optimisation**

* Cache distribué (Varnish) pour accélérer les accès
* Optimisation des requêtes spatiales et indexation avancée
* Compression des données pour limiter la bande passante

## **8. Données & Performances**

* Quantification du volume de données à traiter pour ajuster l'infrastructure (prévision du stockage en To et croissance annuelle).
* Gestion des **pics de charge** avec une architecture adaptée (cache intelligent, file d’attente asynchrone, scaling horizontal).

## **9. Hébergement et Infrastructure**

* Déploiement sur **une infrastructure conteneurisée** avec **Docker et Kubernetes** :
  + Scalabilité automatique.
  + Haute disponibilité des services.
  + Sécurisation et isolation des composants.
* Intégration d’un **API Gateway (Kong, Traefik, Nginx)** pour :
  + **Centraliser la gestion des API**.
  + **Optimiser l’authentification et le monitoring**.
* Monitoring complet avec **Prometheus, Grafana et ELK** pour surveiller les performances et les logs applicatifs.

**10. Validation & Proof of Concept (POC)**

* **Mise en place d’un POC** avant le déploiement final pour :
  + Valider **l’architecture microservices**.
  + Tester les performances des **API et de l’IA**.
  + Ajuster l’**orchestration des conteneurs** et le **monitoring**.

## **11. Documentation**

* Guide utilisateur détaillé avec tutoriels interactifs
* Documentation technique complète avec schémas d’architecture
* Référence API avec exemples d'utilisation
* Support multilingue (FR/EN) avec possibilité d’ajout de langues supplémentaires

## **12. Maintenance et Support**

* Assistance technique assurée par l’équipe IT du CSE
* Mises à jour régulières et correctifs de sécurité
* Forum communautaire et FAQ pour auto-support