

---

# Rapport d'Analyse

Projet de Développement Informatique

---



IGN

Interface d'administration du GéoCaptcha

**Réalisé par :** Fofana Ibrahima, Posado Bañuls Diego, Quillat Maël et Rathana Clément  
**Commandité par :** Mohad Mélodia

10 mars 2025

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Contexte . . . . .	3
1.2	Aspects sociaux et financiers . . . . .	3
<b>2</b>	<b>Objectifs de l'étude - Reformulation du besoin</b>	<b>5</b>
2.1	Objectifs . . . . .	5
2.2	Contraintes . . . . .	5
2.3	Recueil du besoin - Les acteurs . . . . .	6
<b>3</b>	<b>Analyse Fonctionnelle</b>	<b>7</b>
3.1	Fonctionnalités . . . . .	7
3.2	Méthode envisagée . . . . .	8
<b>4</b>	<b>Étude Technique : Choix des Logiciels et Langages - Architecture</b>	<b>9</b>
4.1	Choix du Framework Frontend . . . . .	9
4.2	Choix de l'Architecture . . . . .	9
4.2.1	Diagramme de Composants . . . . .	11
4.2.2	Diagramme de Déploiement . . . . .	12
4.3	Choix des Outils de Conception . . . . .	12
4.4	Choix des Outils de Diagrammes . . . . .	13
4.5	Choix des Outils de Tests et Intégration Continue . . . . .	14
<b>5</b>	<b>Suivi du projet</b>	<b>14</b>
5.1	Planning prévisionnel . . . . .	15
5.2	Risques . . . . .	15
5.3	Attentes . . . . .	16
<b>6</b>	<b>Conclusion</b>	<b>16</b>
<b>7</b>	<b>Annexes</b>	<b>17</b>

## Glossaire

- **API (Application Programming Interface)** : Ensemble de fonctions et de protocoles permettant à différentes applications de communiquer entre elles.
- **Captcha** : Test automatique permettant de différencier un utilisateur humain d'un robot. Le GéoCaptcha repose sur des données géographiques pour cette vérification.
- **DSFR (Design System de l'État Français)** : Système de conception graphique et ergonomique permettant d'assurer la cohérence et l'accessibilité des interfaces numériques des services publics français.
- **Jeu Captcha** : Un test ou une question basée sur des données géographiques que l'utilisateur doit jouer afin d'affirmer qu'il est humain. Nous avons choisi d'utiliser "jeu" et non "défi" pour ne pas déshumaniser cette action qui consiste à prouver que nous sommes un humain.
- **GitHub** : Plateforme en ligne de gestion de versions basée sur Git, permettant la collaboration et l'hébergement du code source.
- **IGN (Institut National de l'Information Géographique et Forestière)** : Organisme public français chargé de la production et de la diffusion des données géographiques de référence.
- **Interface d'administration** : Interface graphique permettant aux administrateurs de gérer les jeux Captcha, les utilisateurs et les métriques associées.
- **MARS (Mission Architecture Réseau et Sécurité)** : Service de l'IGN ayant développé le GéoCaptcha.
- **Métriques** : Ensemble d'indicateurs statistiques permettant d'évaluer la performance des jeux Captcha, notamment le taux de réussite, le nombre de sollicitations et les comportements suspects.
- **Open Source** : Logiciel dont le code source est librement accessible et modifiable par la communauté, favorisant la transparence et la collaboration.
- **ReactJS** : Bibliothèque JavaScript open source développée par Facebook, utilisée pour créer des interfaces utilisateur interactives et performantes.
- **VueJS** : Framework JavaScript progressif permettant de développer des interfaces utilisateur dynamiques et réactives.
- **UX/UI (Expérience utilisateur / Interface utilisateur)** : Ensemble des techniques visant à améliorer l'ergonomie, la navigation et l'expérience globale d'un utilisateur sur une interface numérique.
- **Versioning** : Processus de gestion des différentes versions du code source dans un projet informatique, souvent réalisé avec Git et GitHub.

# 1 Introduction

## 1.1 Contexte

Lorsque vous arrivez sur un site internet, il vous est souvent demandé si vous êtes un humain ou un robot. Pour prouver votre humanité vous devez résoudre un captcha, la plupart du temps sous la forme de déchiffrement de texte ou bien de séquence d'images. L'IGN, par la Mission Architecture Réseau et Sécurité (MARS), a conçu un système novateur dans ce domaine, les GéoCaptcha. Ces captchas reposent sur des données géographiques, l'objectif est de proposer aux utilisateurs un nouveau type de captcha qui se veut ludique et respectueux de la vie privée tout en sensibilisant le public à la donnée géospatiale.

Dans le contexte du projet de développement informatique pour 2024-2025, notre équipe est chargée d'améliorer et d'optimiser l'interface d'administration du GéoCaptcha. À l'heure actuelle, une API administrateur offre la possibilité de gérer les jeux, les clés d'accès et d'accéder à diverses métriques. Toutefois, l'interface liée présente des contraintes en matière d'ergonomie, de design et de fonctionnalités. Pour assurer une expérience utilisateur plus aisée et intuitive, une refonte de l'interface est indispensable.



FIGURE 1 – Exemple de captcha

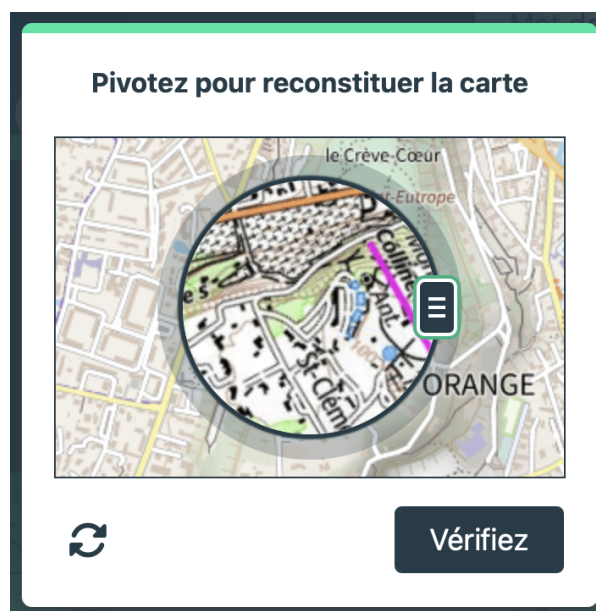


FIGURE 2 – Exemple d'un GéoCaptcha

Initialement, ce projet est conçu pour les développeurs internes de l'IGN, le but est de leur offrir un instrument efficace pour la gestion du système. Néanmoins, son objectif est de se transformer en un projet Open Source, ce qui donnera la possibilité à une communauté beaucoup plus vaste de participer et de l'utiliser dans divers contextes. L'ouverture du code permettra un perfectionnement constant et encouragera une adoption plus vaste de l'idée de GéoCaptcha.

## 1.2 Aspects sociaux et financiers

La réalisation d'un projet implique toujours une estimation du temps et des ressources nécessaires. Dans notre cas, nous sommes une équipe de quatre personnes et nous prévoyons consacrer environ **80 heures** au projet. Cette répartition se fait de la manière suivante :

- **20 heures** dédiées à l’analyse (cadrage, définition des besoins, étude de l’existant).
- **60 heures** consacrées au développement (refonte de l’interface, intégration de l’API, tests et validation).

En ce qui concerne le budget, notre projet a l’avantage de ne requérir aucun investissement en équipements ou en logiciels. Nous utilisons uniquement des technologies Open Source, y compris le framework que nous sélectionnerons (que ce soit ReactJS ou VueJS) et l’API actuelle. En outre, nous nous conformons au Design System de l’État Français (DSFR), qui est accessible gratuitement et garantit une uniformité graphique sans coût additionnel. Par conséquent, à l’exception du coût de la main-d’œuvre, la solution est sans frais.

La réussite d’un projet dépend aussi d’une solide cohésion au sein de l’équipe. Notre équipe se compose de quatre étudiants, chacun contribuant avec ses aptitudes et son savoir-faire pour garantir une avancée efficace du projet :

- **Un chef de projet**, chargé de la coordination, du suivi des tâches et de la communication avec le commanditaire et du développement. Ce rôle sera assuré par **Maël QUILLAT**.
- **Un UX/UI designer et développeur**, chargé de la conception de l’interface utilisateur avec le DSFR et de l’optimisation de l’expérience utilisateur ainsi que de son accessibilité. **Diego POSADO BAÑULS** tiendra ce rôle.
- **Un Lead Développeur Frontend**, chargé de l’expertise sur le framework choisi, du développement des composants réutilisables et de l’implémentation d’interfaces réactives. **Ibrahima FOFANA** sera notre Lead Développeur.
- **Un responsable qualité et développeur**, chargé du développement des visualisations de données et des dashboards ainsi que des tests fonctionnels, de la documentation technique et du guide utilisateur. Ce rôle sera occupé par **Clément RATHANA**.

Cette structure facilite une distribution efficace des tâches et encourage une coopération fluide au sein du groupe. Nous prévoyons d’organiser des réunions hebdomadaires, qu’elles soient organisées avec notre équipe ou avec notre commanditaire, afin de suivre et d’ajuster les tâches en fonction de l’évolution du projet.

## 2 Objectifs de l'étude - Reformulation du besoin

### 2.1 Objectifs

Pour comprendre l'objectif de notre projet il faut arriver à dissocier les différentes composantes du GéoCaptcha. Il y a le jeu GéoCaptcha (cf. figure 2), lorsqu'un utilisateur arrive sur un site internet il doit résoudre un GéoCaptcha. Il existe également une interface administrateur, que nous appelons CaptchAdmin, anciennement Tinder GéoCaptcha (cf. figure 3). Celle-ci permet de créer et de gérer les GéoCaptchas, mais aussi de consulter des métriques et des logs spécifiques à ces derniers. De plus, une distribution de clés d'accès est disponible. Le but principal de ce projet est d'améliorer l'interface CaptchAdmin, pour la rendre plus ergonomique, accessible et performante.



FIGURE 3 – Interface administrateur déjà existante

### 2.2 Contraintes

La création de l'interface CaptchAdmin doit se conformer à diverses contraintes techniques, ergonomiques et organisationnelles.

L'interface doit être élaborée en utilisant ReactJS ou VueJS, pour garantir une adéquation parfaite avec les technologies web actuelles. L'application doit aussi se conformer aux normes du Design System de l'État Français (DSFR), assurant par conséquent une uniformité visuelle et une accessibilité en accord avec les standards actuels.

De plus, puisque la solution doit être open source, le code devra être correctement organisé et documenté pour encourager les apports de contributeurs externes. Le projet sera géré en utilisant une approche collaborative sur GitHub, ce qui inclut l'utilisation de branches, de demandes de tirage et d'examen de code réguliers.

Ces contraintes sont imposées par notre commanditaire. À celles-ci s'ajoute la contrainte du temps. En effet, le projet doit être rendu au 1er Avril 2025 ce qui nous laisse une dizaine de séances pour mener à bien ce projet.

## 2.3 Recueil du besoin - Les acteurs

Comme évoqué précédemment, plusieurs acteurs sont amené à utiliser les GéoCaptchas.

Il y a l'utilisateur final, celui qui va jouer au GéoCaptcha pour prouver son humanité, nous ne nous occupons pas de cet utilisateur pour notre projet.

Il y a ensuite les utilisateurs principaux de l'interface CaptchAdmin, ce sont les administrateurs du GéoCaptcha. Ils sont chargés de superviser les jeux GéoCaptcha, de contrôler leur contenu et d'examiner les métriques. Ils requièrent un instrument performant et facile à utiliser.

Les développeurs internes de l'IGN participent à la maintenance et à l'évolution du projet. Bien qu'à court terme ils soient amenés à administrer l'interface, donc générer des GéoCaptchas, générer des clés d'accès et visualiser les différentes métriques, à long terme ils seront amenés à simplement gérer les éventuels bugs de l'interface.

Enfin, comme le projet est destiné à se transformer en une solution open source, il sera possible d'impliquer à long terme une communauté élargie de contributeurs externes. Il sera donc essentiel de fournir un code bien documenté et accessible pour favoriser la contribution et l'évolution constante du projet.

**DIAGRAMME CAS D'UTILISATION DU GEOCAPTCHA**

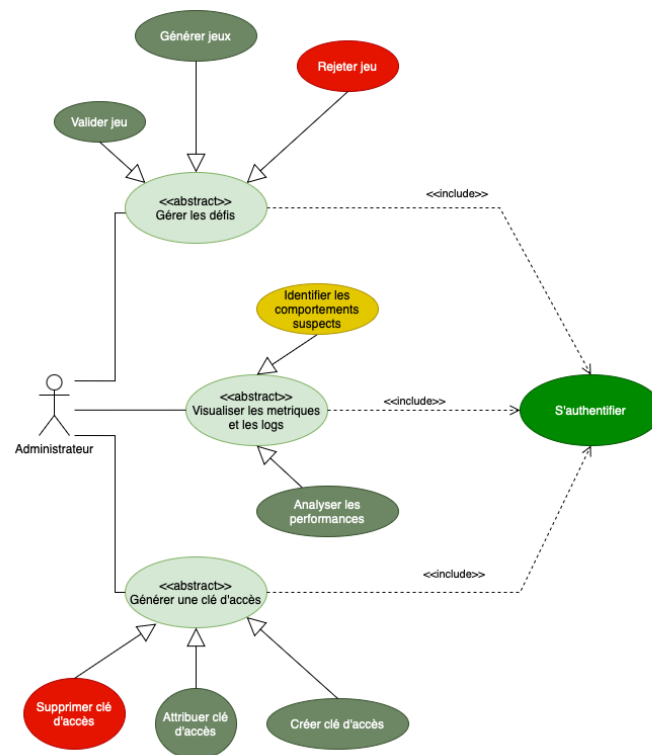


FIGURE 4 – Diagramme cas d'utilisation

La figure 4 présente un cas d'utilisation de l'interface CaptchAdmin où un administrateur peut utiliser plusieurs fonctionnalités qui seront décrites dans la section suivante.

### 3 Analyse Fonctionnelle

#### 3.1 Fonctionnalités

Comme évoqué précédemment, la demande de notre commanditaire est d'obtenir une maquette d'interface fonctionnelle plus ergonomique et plus efficace. Cette interface doit notamment proposer les fonctionnalités suivantes.

##### Gestion des clés d'accès

L'interface devra permettre la **création** et **suppression de clés d'accès**.

##### Gestion des jeux GéoCaptcha

L'outil devra inclure :

- La **création de nouveaux jeux**, définis manuellement ou générés aléatoirement. (Exemple : choisir une zone géographique et ne générer des GéoCaptchas que dans cette zone)
- La **validation ou rejet des jeux soumis**.

##### Analyse des Métriques et Logs

L'interface devra offrir des outils d'analyse pour :

- Suivre le **taux de réussite** des jeux.
- Visualiser les **zones géographiques les plus sollicitées**.
- Identifier le **temps moyen de résolution** des défis.

Ces fonctionnalités peuvent être amenées à évoluer. On pourrait par exemple imaginer de nouvelles métriques à visualiser

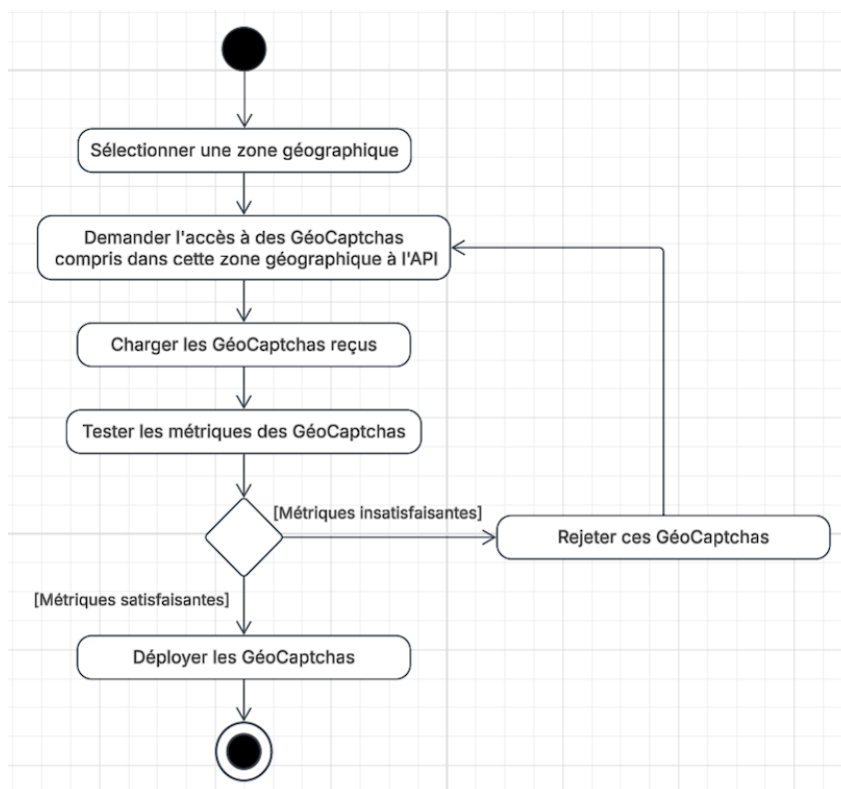


FIGURE 5 – Diagramme d'activité



La figure 5 présente un diagramme d'activité montrant comment l'utilisateur de l'interface d'administration peut choisir de valider ou invalider le déploiement d'un GéoCaptcha selon une zone géographique particulière.

### 3.2 Méthode envisagée

Pour mener à bien ce projet nous avons réfléchi à quelle stratégie mettre en place. L'objectif principal étant de mettre en place une maquette visuelle ergonomique et performante il est important de faire évaluer ces deux critères. Pour ce qui est de la performance, nous mettrons en place des tests et nous demanderons avis à notre commanditaire ainsi qu'à l'équipe de développeur de l'IGN qui travaille sur ce projet. Concernant l'ergonomie, nous prévoyons de créer plusieurs maquettes visuelles afin d'effectuer des sondages auprès des étudiants de l'école, des professeurs et bien sûr de l'équipe de développeur de l'IGN. Nous pourrons grâce à cela évaluer quel type de bouton parle au plus grand nombre, quelle forme globale semble la plus appropriée ou tout autre type d'élément d'ergonomie susceptible d'améliorer l'expérience utilisateur.

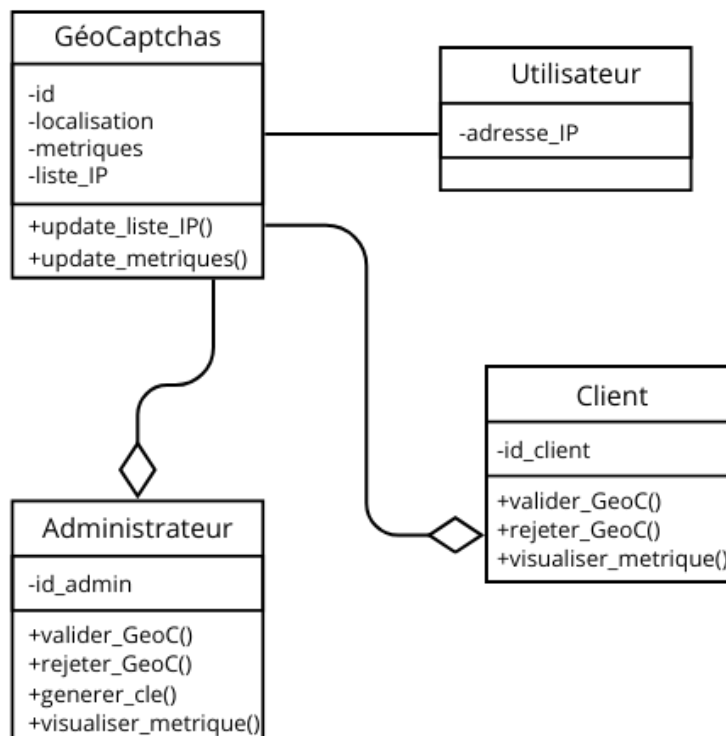


FIGURE 6 – Diagramme de classe

La figure 6 présente un diagramme de classe pour notre projet. Bien que nous n'ayons pas à gérer l'utilisateur final, il est présenté sur ce diagramme afin de présenter tous les acteurs qui interagissent avec les GéoCaptchas. Ici le client représente alors quelqu'un qui souhaiterait utiliser l'interface CaptchAdmin pour générer des GéoCaptchas pour son site internet. L'administrateur lui possède déjà une clé d'accès et peut en générer de nouvelles.

## 4 Étude Technique : Choix des Logiciels et Langages - Architecture

Cette partie détaille les choix techniques effectués pour le projet **GéoCaptcha**, en justifiant chaque décision en fonction des besoins fonctionnels, des contraintes techniques et des objectifs du projet. Les choix portent sur les frameworks frontend, le backend, la base de données, l'architecture globale, ainsi que les outils de conception et de modélisation. Pour réaliser cette étude nous avons créé plusieurs tables comparatives.

### 4.1 Choix du Framework Frontend

**Options : ReactJS vs VueJS**

Critères	ReactJS	VueJS
Philosophie	Bibliothèque JavaScript centrée sur les composants	Framework progressif, facile à intégrer
Facilité d'apprentissage	Courbe d'apprentissage plus complexe (JSX, hooks)	Plus simple à prendre en main (syntaxe intuitive)
Performance	Très performant grâce au Virtual DOM	Léger et rapide
Réactivité	Unidirectionnel (flux de données strict)	Réactif avec un binding bidirectionnel
Écosystème	Nécessite des bibliothèques tierces (Redux, React Router)	Outils intégrés (Vue Router, Vuex)
Taille de la communauté	Très grande communauté et support solide	Moins grande, mais en pleine croissance
Adoption dans l'industrie	Très utilisé par les grandes entreprises (Meta, Airbnb)	Moins adopté dans les grandes entreprises
Interopérabilité	Plus centré sur l'écosystème React	Facile à intégrer avec d'autres projets

TABLE 1 – Comparaison entre ReactJS et VueJS

**Choix : VueJS**

**Justification :** VueJS est plus facile à prendre en main, ce qui correspond aux besoins du projet GéoCaptcha qui doit se faire dans un délai restreint. De plus, il est facile à intégrer avec d'autres projets ce qui encourage le déploiement open source du projet.

### 4.2 Choix de l'Architecture

**Options : Monolithique vs N-Tiers vs Microservices**

Architectures	Avantages ✓	Inconvénients ✗
<b>Monolithique (Tout en un)</b>	Déploiement et développement plus simples Facilité de debugging et de test Moins de dépendances externes	Difficulté à évoluer et à scaler Maintenance plus compliquée sur le long terme Risque de panne totale (tout est lié)
<b>N-Tiers (3-Tiers ou plus) (Séparation Frontend, Backend, Base de données)</b>	Meilleure organisation et modularité Plus sécurisé (API backend protégée) Possibilité d'évolution progressive	Déploiement un peu plus complexe Communication entre les couches (API, latence) Gestion des versions plus délicate
<b>Microservices (Services indépendants communiquant via API)</b>	Très évolutif et scalable  Haute disponibilité (un service peut tomber sans affecter les autres) Développement parallèle possible	Complexité accrue (orchestration avec Kubernetes/-Docker Swarm)  Risque de latence (communication entre services)  Plus coûteux en infrastructure

TABLE 2 – Comparaison des architectures logicielles

### Choix : Architecture N-Tiers

**Justification :** L'architecture N-Tiers (séparation frontend, backend, base de données) offre une meilleure modularité et organisation, ce qui est essentiel pour un projet comme GéoCaptcha où les fonctionnalités sont clairement séparées. De plus, cette architecture permet une meilleure scalabilité et une maintenance plus facile. Elle est également plus sécurisée, car chaque couche peut être isolée et protégée individuellement.

### 4.2.1 Diagramme de Composants

DIAGRAMME DE COMPOSANT DE L'ARCHITECTURE N-TIERS DU GEOCAPTCHA

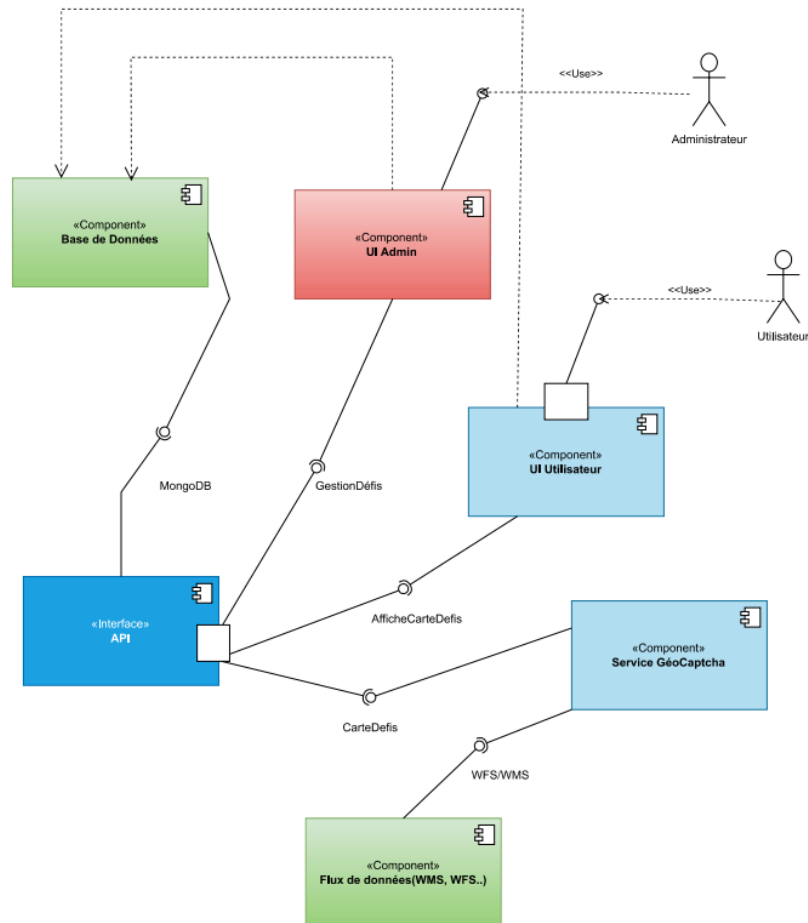


FIGURE 7 – Diagramme de Composants de l'Architecture N-Tiers du GéoCaptcha

#### Explication :

- **Base de Données** : MongoDB est utilisé pour stocker les données des utilisateurs, des défis et des métriques.
- **UI Admin** : L'interface d'administration, développée en ReactJS, permet la gestion des utilisateurs et des défis.
- **UI Utilisateur** : L'interface utilisateur, également développée en ReactJS, permet aux utilisateurs de résoudre les défis Captcha.
- **Service GéoCaptcha** : Le backend expose une API REST pour gérer les interactions entre le frontend et la base de données.
- **Flux de Données** : Les flux WMS/WFS sont utilisés pour afficher les cartes et les données géographiques.

## 4.2.2 Diagramme de Déploiement

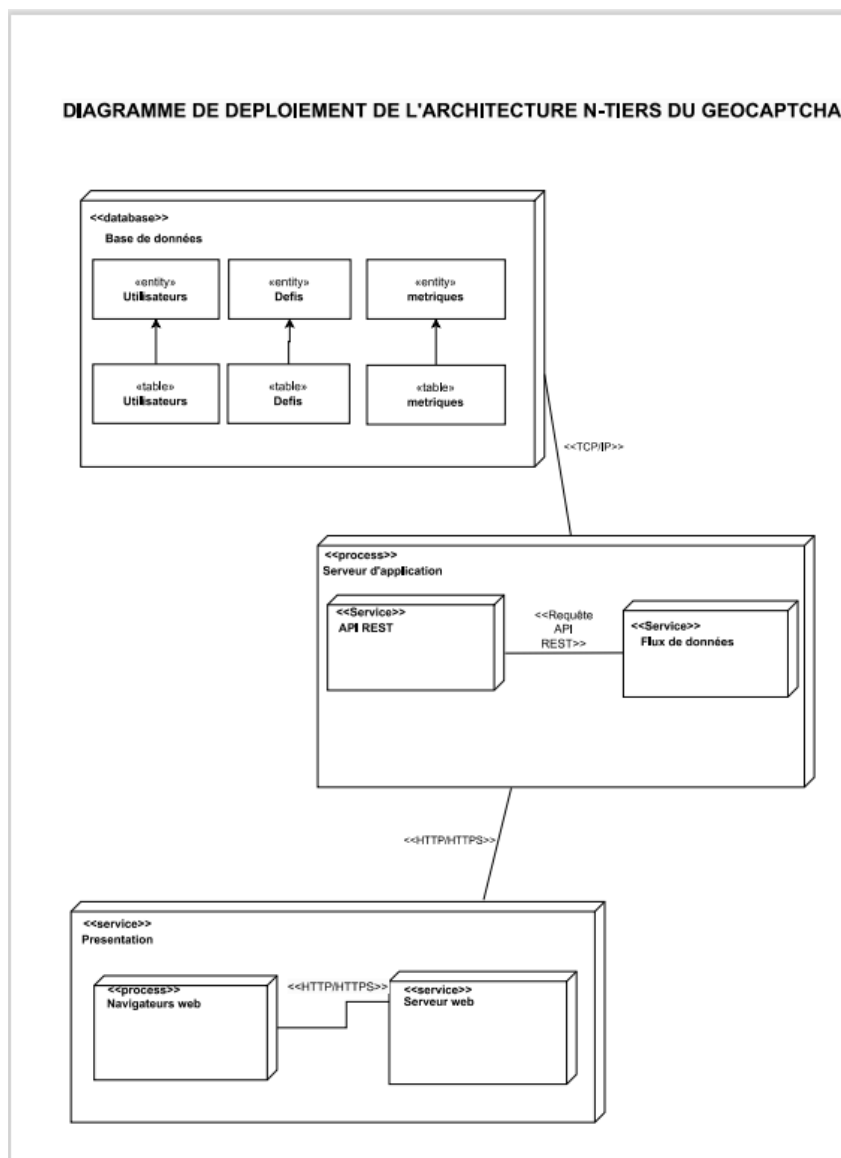


FIGURE 8 – Diagramme de Déploiement de l'Architecture N-Tiers du GéoCaptcha

### Explication :

- **Base de Données :** MongoDB est déployé sur un serveur dédié, avec des tables pour les utilisateurs, les défis et les métriques.
- **Serveur d'Application :** Il est utilisé pour héberger l'API REST, qui gère les requêtes entre le frontend et la base de données.
- **Serveur Web :** Il est utilisé pour servir les fichiers statiques (HTML, CSS, JS) du frontend.
- **Navigateurs Web :** Les utilisateurs accèdent à l'interface via un navigateur web, qui communique avec le serveur web via HTTP/HTTPS.

## 4.3 Choix des Outils de Conception

**Options :** Figma vs Sketch vs Photoshop

Critères	Figma	Sketch	Photoshop
<b>Collaboration</b>	Excellente collaboration en temps réel via le cloud	Pas de collaboration native en temps réel	Pas conçu pour la collaboration en direct
<b>Compatibilité</b>	Fonctionne sur Windows, macOS et Linux (via navigateur)	Disponible uniquement sur macOS	Compatible Windows et macOS
<b>Facilité d'utilisation</b>	Interface intuitive et facile à prendre en main	Simple pour les designers UI/UX	Interface plus complexe pour le design UI/UX
<b>Prototypage intégré</b>	Oui, avec interactions avancées	Nécessite un outil externe	Très limité pour le prototypage
<b>Gestion des composants et design system</b>	Excellente gestion des composants réutilisables	Bonne gestion des symboles, mais moins flexible que Figma	Pas conçu pour les design systems UI/UX
<b>Performance</b>	Fluide car basé sur le cloud	Très rapide	Peut être lourd pour le design UI/UX
<b>Prix</b>	Gratuit pour les bases, payant pour les équipes	Payant	Payant

TABLE 3 – Comparaison des outils de conception

### Choix : Figma

**Justification** : Figma est un outil de design UI/UX basé sur le cloud, offrant une excellente collaboration en temps réel et une gestion des composants réutilisables. Il est compatible avec tous les systèmes d'exploitation et permet un prototypage avancé, ce qui est essentiel pour concevoir une interface utilisateur intuitive et accessible. Figma est également gratuit pour les projets de base, ce qui correspond aux contraintes budgétaires du projet.

## 4.4 Choix des Outils de Diagrammes

**Options** : Draw.io vs StartUML vs Enterprise Architect

Critères	Draw.io	StarUML	Enterprise Architect
Type	Web et Desktop (Gratuit)	Desktop (Free-mium)	Desktop (Payant)
Facilité d'apprentissage	Très intuitif	Moyennement intuitif	Complexe mais puissant
Support UML	Basique mais efficace	Très complet	Ultra-complet avec modèles avancés
Collaboration	Via Google Drive et GitHub	Pas de collaboration native	Travail en équipe possible
Personnalisation	Basique (formes simples)	Bonne personnalisation	Très personnalisable
Formats d'export	PNG, SVG, XML	PDF, HTML, Markdown	Tous formats (PDF, XMI, Code)
Intégration avec le code	Non	Partielle (JSON, Java)	Génération automatique
Prix	Gratuit	Gratuit/Payant	Payant

TABLE 4 – Comparaison des outils de diagrammes

#### Choix : Draw.io

**Justification :** Draw.io est un outil gratuit, intuitif et facile à utiliser pour créer des diagrammes UML. Il offre une collaboration via Google Drive et GitHub, ce qui est essentiel pour un projet open source comme GéoCaptcha. Bien qu'il soit basique, il est suffisamment puissant pour les besoins du projet, notamment pour la création de diagrammes de déploiement et de composants.

### 4.5 Choix des Outils de Tests et Intégration Continue

#### Choix : Jest + Cypress + GitHub Actions

##### Justification :

- **Jest** est utilisé pour les tests unitaires et d'intégration, offrant une couverture rapide et efficace du code.
- **Cypress** est choisi pour les tests end-to-end (E2E), permettant de valider le comportement de l'application dans son ensemble.
- **GitHub Actions** est utilisé pour l'intégration continue (CI), automatisant les builds et les tests à chaque push ou pull request, garantissant ainsi la qualité du code.

## 5 Suivi du projet

Le suivi du projet se base sur divers aspects clés, comme la création d'un calendrier prévisionnel décomposant les différentes étapes de l'évolution, la reconnaissance des dangers susceptibles de nuire à la bonne marche du projet, tout comme l'établissement des objectifs et produits attendus à chaque phase. Une organisation minutieuse du temps et des

activités sera effectuée pour garantir le respect des échéances, tandis qu'une évaluation régulière des risques aidera à prévoir les obstacles techniques ou organisationnels potentiels. Finalement, des rencontres de suivi avec les commanditaires et des bilans d'avancement internes assureront une évolution constante en fonction des exigences du projet et une amélioration des ressources disponibles.

## 5.1 Planning prévisionnel

Pour faciliter la lecture des tâches à venir nous avons réalisé un planning prévisionnel.

	05/02/2025	12/02/2025	26/02/2025	05/03/2025	10/03/2025	12/03/2025	18/03/2025	19/03/2025	25/03/2025	26/03/2025	31/03/2025	01/04/2025	02/04/2025
Compréhension des enjeux													
Phase d'analyse													
Réalisation d'un premier design de l'interface													
Réfléchir à une métrique pour valider les géocapcha													
Interface visuelle pour l'administrateur													
Sondage ergonomie													
Rapport et livrables													
Préparation de la présentation													
Valider/invalidé en fonction de l'emplacement géographique													
Echéances													

FIGURE 9 – Planning prévisionnel

Ce planning est voué à évoluer en fonction de l'avancement du projet. De plus, concernant l'organisation de notre projet, nous avons prévu de faire des points hebdomadaires avec notre commanditaire. Pour faciliter l'échange et le partage de données nous avons également créer un Google Drive commun, cela permet de ne pas polluer le GitHub avec des documents inutiles au moment du rendu du projet.

## 5.2 Risques

Pour gérer au mieux notre projet, il faut évaluer les risques potentiels.

Nature du risque	Probabilité	Gravité	Conséquences	Solution corrective	Solution Préventive
Code non fonctionnel	Moyenne	Forte	Pas de livrable	Regarder les commentaires, lancer des séries de tests unitaires pour repérer les erreurs	Commenter le code, lancer régulièrement des tests unitaires
Manque d'organisation au sein du groupe	Faible	Moyenne	Avancée du projet entravée, peut engendrer des retards	Faire des réunions pour faire un point sur l'avancée de chacun et savoir sur quoi travaille chaque membre	Faire des points réguliers, attribuer clairement les tâches, favoriser la communication au sein de l'équipe
Mauvaise évaluation du temps	Forte	Moyenne	Prise de retard sur les différentes tâches engendrant un retard global	Adapter le diagramme de Gantt	Communiquer régulièrement le niveau d'avancement, prévenir les potentiels retards voire l'avance
Mauvaise connaissance des outils imposés	Moyenne	Faible	Perte de temps de travail puisqu'il faudra l'allouer à l'apprentissage de ces outils	Se renseigner d'emanère autodidacte grâce à des documentations en ligne, tutoriels, etc...	Se renseigner le plus en amont possible sur l'utilisation de ces outils

FIGURE 10 – Matrice des risques

On remarque que le plus gros risque est lié à un code non fonctionnel, ce risque a tout de même une faible probabilité de se réaliser, notamment grâce à l'aide de notre commanditaire. Enfin une mauvaise évaluation du temps est le risque le plus probable compte tenu du court délai pour réaliser ce projet.



### 5.3 Attentes

Afin de totalement comprendre les attentes de notre commanditaire nous lui avons demandé de classer par ordre de priorité et de noter les objectifs majeurs du projet.

OBJECTIF	DÉTAIL	PRIORITÉ (/10)	ÉTAT
Réaliser une maquette de l'interface	Proposer une design ergonomique et intuitif	10	Non démarré
Gérer les rôles	Offrir une gestion des rôles et des niveaux d'accès	9	Non démarré
Gérer les défis	Offrir des outils pour créer de nouveaux défis	9	Non démarré
Visualiser les métriques	Proposer une interface pour analyser les performances des défis	6	Non démarré

FIGURE 11 – Tableau des attentes du commanditaire

Ces attentes n'ont pas démarré au moment du rendu de ce rapport d'analyse mais sont maintenant notre priorité pour la phase de développement.

## 6 Conclusion

Ce rapport a permis de définir les objectifs du projet, les contraintes techniques et l'organisation de l'équipe. Les prochaines étapes incluent le développement de l'interface et la mise en place des tests.

## 7 Annexes

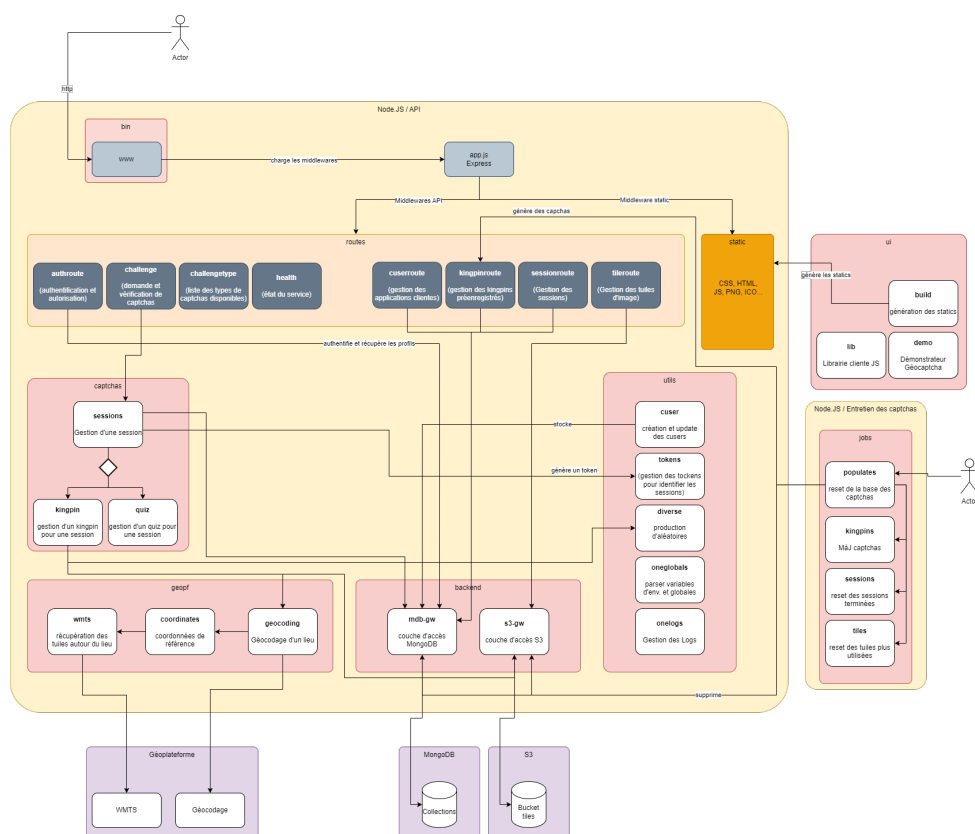


FIGURE 12 – Diagramme d'architecture



FIGURE 13 – Site de démonstration pour le GeoCaptcha