



Faculté des Sciences Agadir  
Master d'excellence-Analytique des données  
et intelligence artificielle



**Module :**  
**SYSTEMS REPARTIS & DISTRIBUES**

**Projet N°7**  
***Moteur de recherche distribué***  
***(Conception UML)***

Réalisé par :

- KINAD Kawtar
- MOUSTIKE Imane
- KHAIR Latifa
- KENNOUZ Ayoub

Encadré par :

- Pr. Idraiss Jaafar

Année académique : 2024-2025



## **Table de Matières**

<b>1. Introduction .....</b>	<b>5</b>
<b>2. Diagramme de Cas d'Utilisation .....</b>	<b>5</b>
<b>3. Diagramme de Classes .....</b>	<b>6</b>
<b>4. Diagramme de Séquence .....</b>	<b>9</b>
<b>6. Diagramme de Déploiement .....</b>	<b>10</b>
<b>7. Diagramme d'Activité .....</b>	<b>11</b>
<b>8. Conclusion .....</b>	<b>12</b>

## Table de Figures

<b>Figure 1:Diagramme de Cas d'Utilisation .....</b>	<b>5</b>
<b>Figure 2:Diagramme de Classes.....</b>	<b>8</b>
<b>Figure 3:Diagramme de Séquence .....</b>	<b>9</b>
<b>Figure 4:Diagramme de Déploiement .....</b>	<b>10</b>
<b>Figure 5:Diagramme d'Activité .....</b>	<b>11</b>

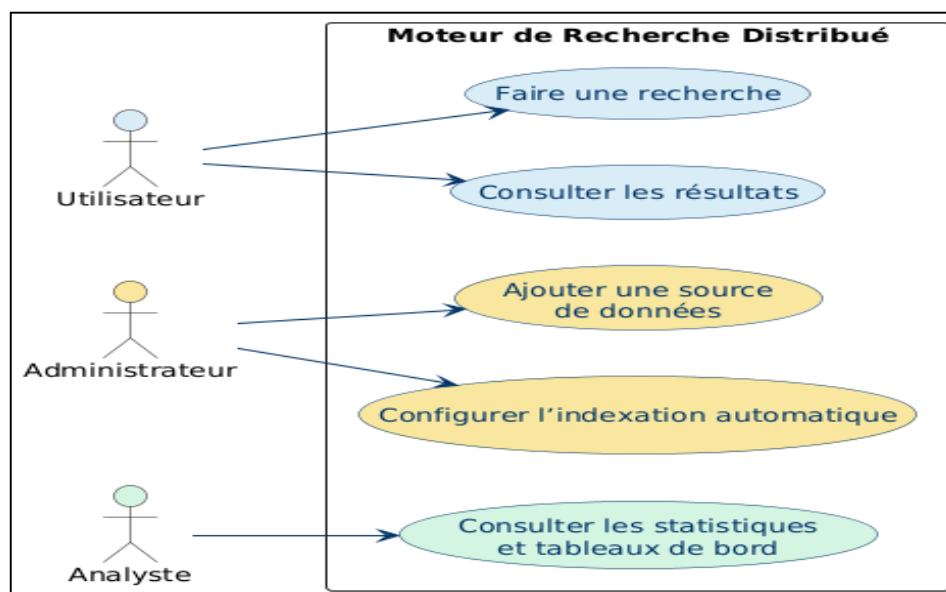
## 1. Introduction

Dans ce projet, nous avons conçu un système de moteur de recherche distribué utilisant plusieurs composants technologiques. L'objectif principal est d'illustrer les différentes interactions entre ces composants et leurs comportements à travers des diagrammes UML. Ce rapport comprend l'analyse et la description de ces diagrammes qui suivent un ordre logique permettant de comprendre le système de manière progressive.

## 2. Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation (Use Case Diagram) présente les **différents utilisateurs** du système (acteurs) ainsi que les **fonctionnalités principales** qu'ils peuvent utiliser. Il met en lumière les interactions entre l'extérieur (utilisateurs, administrateurs, analystes) et le moteur de recherche distribué.

- **L'utilisateur** interagit principalement avec l'**API** ou **Kibana** pour soumettre des requêtes et consulter les résultats.
- **L'analyste** consulte les statistiques via **Kibana**.
- **L'administrateur** supervise l'indexation et configure l'ingestion de données.



**Figure 1: Diagramme de Cas d'Utilisation**

### 3. Diagramme de Classes

Le diagramme de classes décrit la **structure interne du système** : les classes principales, leurs attributs, méthodes et les relations qu'elles entretiennent (association, dépendance...). Il offre une **vue statique** du système.

#### ➤ Classes principales :

##### a. Utilisateur et ses rôles

La classe Utilisateur représente toute personne interagissant avec le système. Un utilisateur peut soumettre plusieurs requêtes, d'où une relation 1 à N avec la classe **RequeteRecherche**.

##### b. RequeteRecherche

Cette classe représente les requêtes textuelles (ou avec filtres) que l'utilisateur saisit pour interroger le moteur. Elle contient des attributs comme :

- **Terme** : mot-clé ou expression à rechercher.
- **Date** : horodatage de la soumission.

Elle est créée par l'utilisateur, puis transmise à l'API, qui va l'envoyer à Elasticsearch pour traitement.

##### c. API

La classe API sert d'interface intermédiaire entre l'interface utilisateur et le cœur du moteur de recherche (**ElasticsearchCluster**). Elle reçoit des objets **RequeteRecherche**, les interprète et envoie des instructions **au cluster Elasticsearch**.

Elle génère également les statistiques de recherche, en lien avec la classe Statistiques.

#### d. ElasticsearchCluster et NoeudElasticsearch

Le moteur de recherche utilise une architecture distribuée, modélisée par la classe **ElasticsearchCluster**, qui regroupe plusieurs **NoeudElasticsearch**. Il s'agit d'une relation de **composition** : la destruction du cluster entraîne celle des nœuds. Chaque nœud représente un serveur du cluster et héberge des **index**.

#### e. Index et Document

Chaque *NoeudElasticsearch* peut contenir un ou plusieurs index, qui sont des structures optimisées pour la recherche rapide dans de grands volumes de données. Un index regroupe des documents, unités d'information textuelle comme des articles ou des messages.

#### f. Logstash

**Logstash** est un composant d'ingestion qui collecte, nettoie et transforme les données provenant de sources externes, puis les envoie directement au **cluster Elasticsearch** pour indexation.

#### g. Kibana et Statistiques

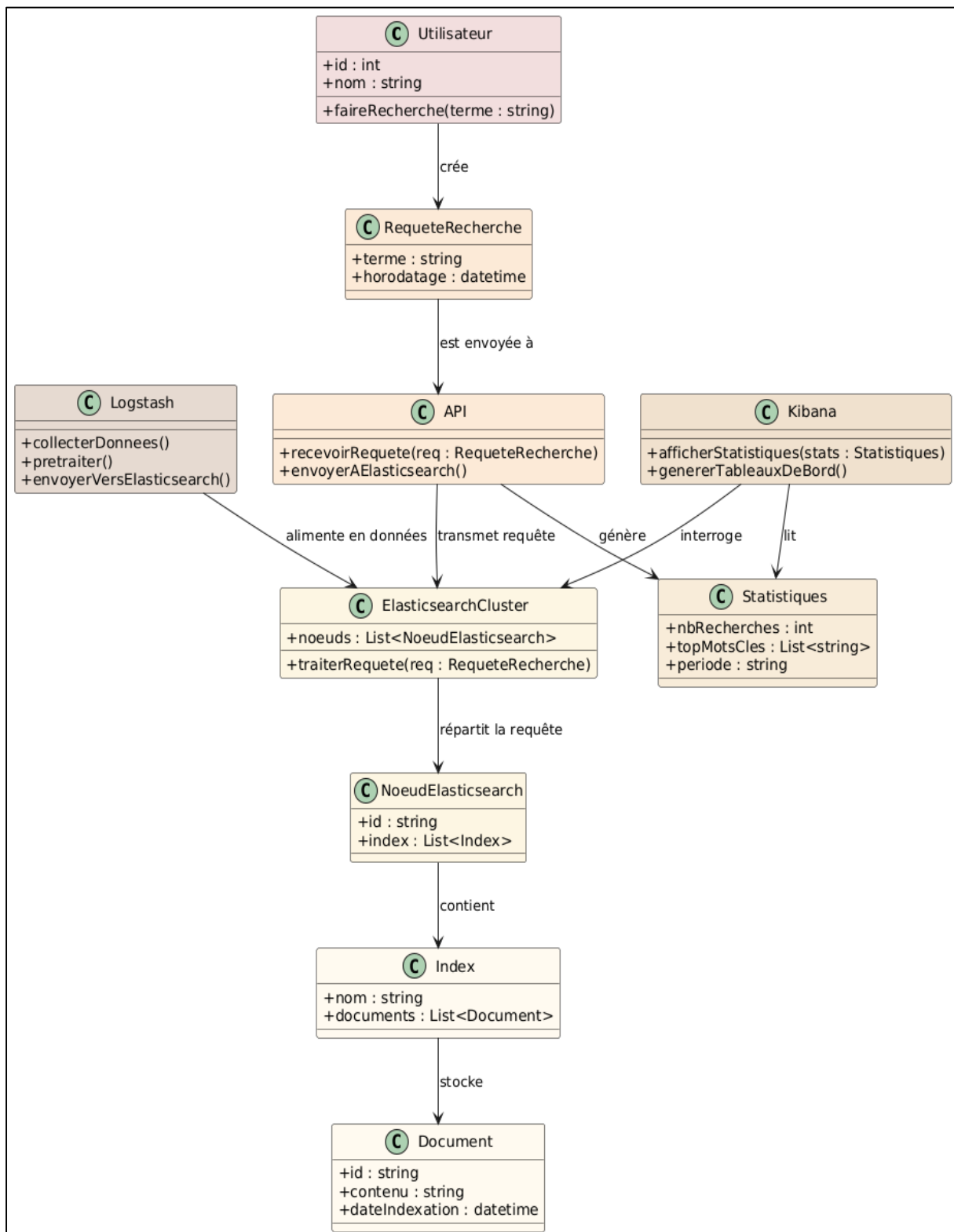
**Kibana** est l'outil de visualisation utilisé par l'admin ou l'analyste. Il affiche les Statistiques générées par le système :

- Fréquence de recherche
- Termes les plus utilisés
- Temps de réponse

Les statistiques sont créées par l'API après chaque recherche et consultées via Kibana.

#### Explication :

- ! Le diagramme montre la structure interne du système, avec des classes représentant des entités distinctes et des relations de dépendance entre elles, comme l'Utilisateur qui crée une requête de recherche, et l'API qui la transmet à Elasticsearch pour traitement.



**Figure 2:Diagramme de Classes**

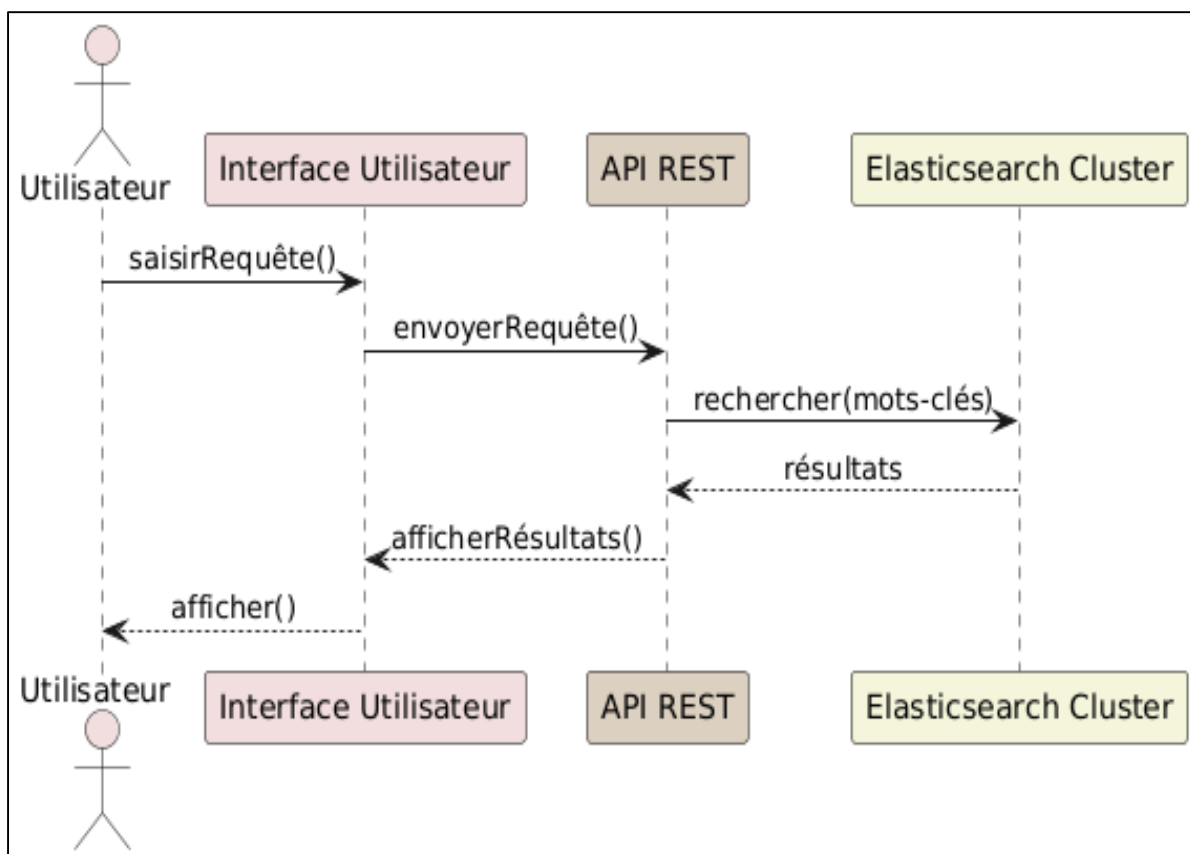


## 4. Diagramme de Séquence

Le diagramme de séquence permet de visualiser un scénario dynamique, en montrant les échanges entre les composants au fil du temps. Il représente l'ordre d'appel des méthodes et l'enchaînement logique des opérations lors d'une recherche.

### Processus:

1. L'**Utilisateur** saisit une requête via l'interface utilisateur.
2. La requête est envoyée à l'**API**.
3. L'**API** transmet la requête à **Elasticsearch**.
4. **Elasticsearch** traite la requête et renvoie les résultats à l'**API**.
5. Enfin, l'**API** affiche les résultats à l'**Utilisateur** via l'interface.



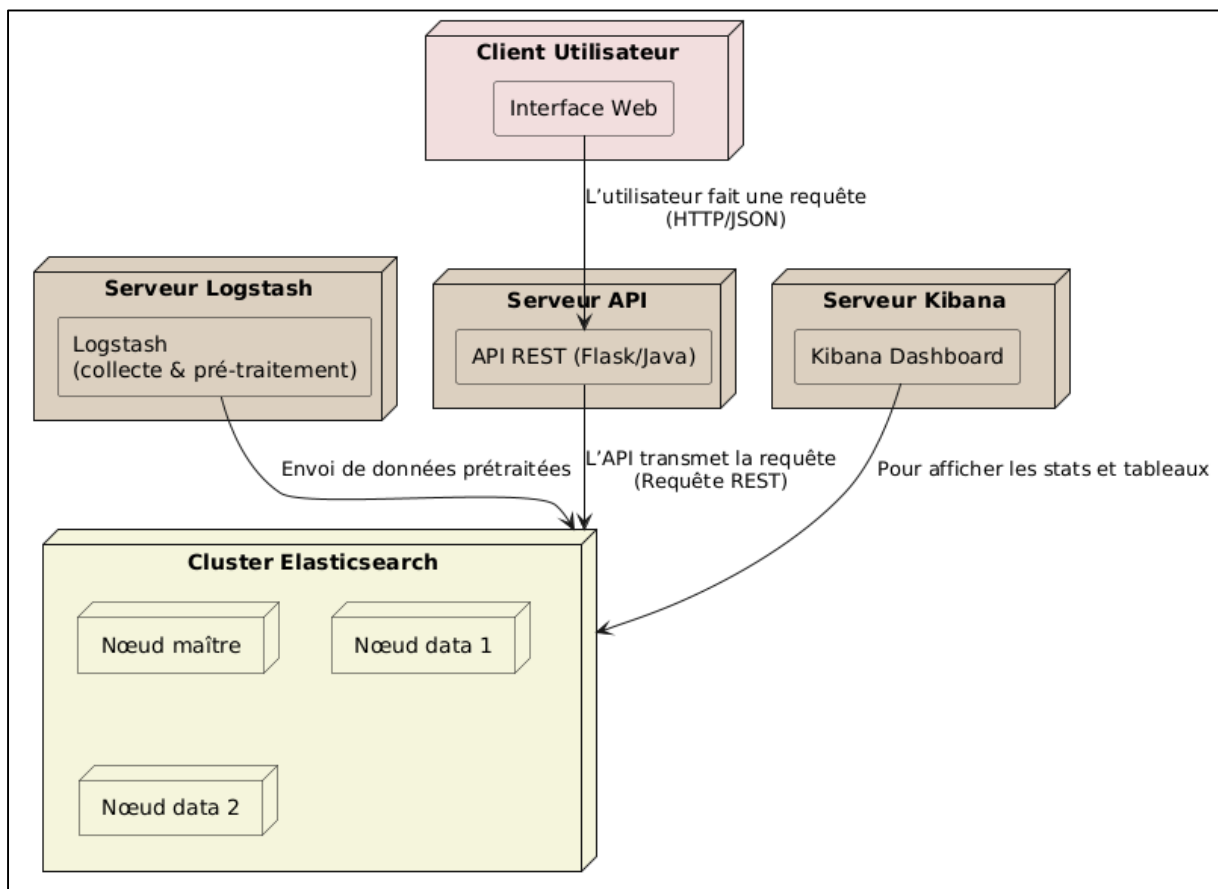
**Figure 3:Diagramme de Séquence**

## 6. Diagramme de Déploiement

Le diagramme de déploiement décrit la configuration physique du système : où se trouvent les composants logiciels, sur quelles machines ou serveurs ils s'exécutent, et comment ils communiquent.

### Explication :

- Le **Serveur Logstash** est responsable de la collecte et du prétraitement des données avant qu'elles ne soient envoyées vers **Elasticsearch**.
- Le **Serveur API** héberge l'API REST qui interagit avec l'interface utilisateur et transmet les requêtes à Elasticsearch.
- **Kibana** est déployé sur un serveur distinct pour afficher les résultats et les statistiques.
- **Elasticsearch** est déployé sur un cluster avec plusieurs nœuds pour gérer les requêtes de recherche.



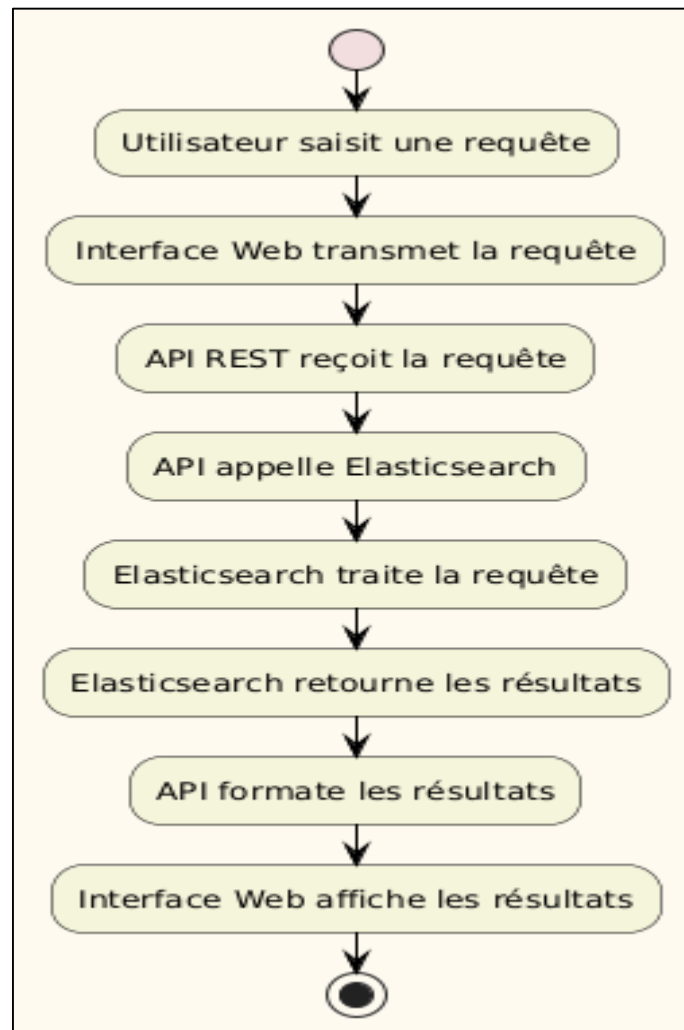
**Figure 4: Diagramme de Déploiement**

## 7. Diagramme d'Activité

Le diagramme d'activité montre le flux des actions dans un scénario d'exécution, en détaillant les étapes impliquées dans le traitement d'une requête de recherche.

**Processus :**

1. **Utilisateur saisit une requête** : L'utilisateur initie le processus de recherche.
2. **Transmission de la requête** : L'interface web transmet la requête à l'API.
3. **Traitement par Elasticsearch** : Elasticsearch traite la requête et renvoie les résultats à l'API.
4. **Affichage des résultats** : Les résultats sont affichés à l'utilisateur via l'interface.



**Figure 5:Diagramme d'Activité**

## **8. Conclusion**

Le moteur de recherche distribué présenté dans ce rapport repose sur une architecture modulaire où chaque composant joue un rôle crucial. Les diagrammes UML ont permis de décrire et d'organiser le système de manière visuelle, facilitant ainsi sa compréhension. Ce rapport fournit une vue d'ensemble claire du fonctionnement interne du système, avec des interactions détaillées entre les différents acteurs et composants logiciels.