



*Élaboration 1 du :*

# META-MOTEUR DE RECHERCHE SUR LE WEB ORIENTE COMMUNAUTE

***Équipe :***

M. Jérémie FRÉCHARD

Mlle Cécile GIRARD

Mlle Aysel GUNES

M. Pierre RAMOS

*Étudiants en Master Informatique première année*

***Client :***

M. Rushed KANAWATI

***Encadrant :***

Mme Sophie TOULOUSE

*(15/03/2006)*

*Version 2.0*

# *Sommaire*

<b>I/ Cas d'utilisation.....</b>	<b>3</b>
1)Auteurs et liste des cas d'utilisation.....	3
2)Diagramme de cas d'utilisation.....	5
3)Développement des cas d'utilisation.....	7
4)Diagrammes de séquence.....	13
 <b>II/ Architecture globale.....</b>	 <b>15</b>
1)Présentation générale.....	15
2)Messages Serveur / Client .....	16
3)Messages Agent / Agent.....	16
4)Messages Serveur / Modules de tris.....	17
5)Messages Clients / Modules de tris.....	17

# I/ Cas d'utilisation

## **1) Acteurs et liste des cas d'utilisation**

### **Acteurs :**

- *Utilisateur* : Acteur demandant une recherche d'une certaine requête à l'aide du système composé d'agents et MoteurRecherche.
- *Administrateur* : Utilisateur particulier étant administrateur et ayant la possibilité de garder les messages échangés, internes et externes au système (les logs du système).

### **Cas d'utilisation :**

- Faire une recherche,
- Installer un agent,
- Désinstaller un agent,
- Se connecter à un agent,
- Se déconnecter à un agent,
- Etablir une liste de résultats pertinente,
- Remplir le formulaire.

<b><i>Résumé</i></b>	<b><i>But de l'utilisateur</i></b>	<b><i>Sous fonction</i></b>
- Faire une recherche	- Etablir une liste de résultats pertinente - Se connecter à un agent - Se déconnecter à un agent - Remplir le formulaire	- Installer un agent - Désinstaller un agent

Notre système peut se décomposer en sous cas d'utilisation.

Dans ce cas, les acteurs seront :

- *Agent* : Rôle intermédiaire entre l'utilisateur et les autres agents ainsi que le MoteurRecherche. Il fait la demande de la recherche, il transmet les mots-clés et les résultats aux autres agents. Puis il recevra le tri de chaque autre agent et effectuera le tri final et renverra la liste des résultats à l'utilisateur.

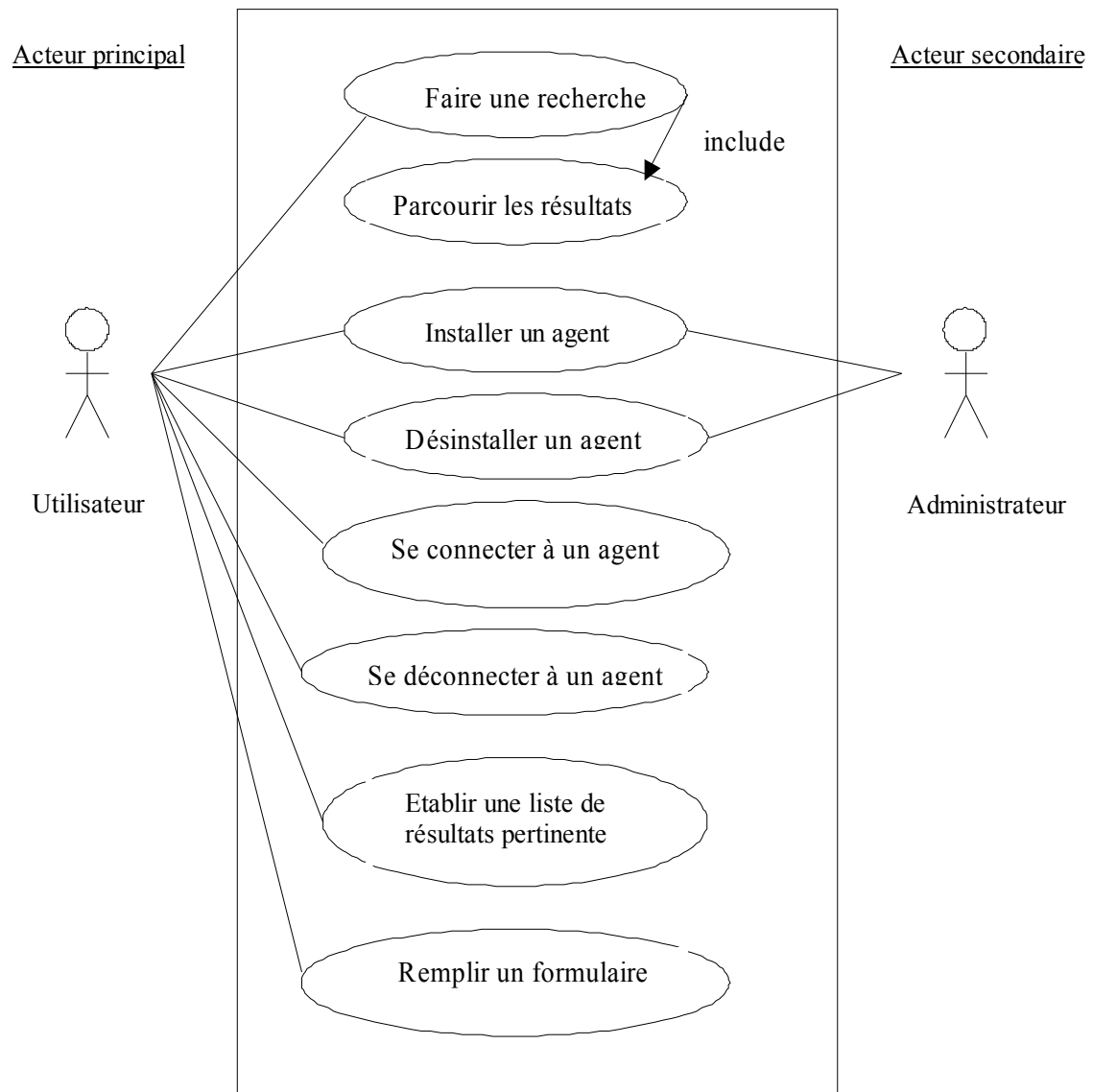
NB : le registry informera quels sont les agents connectés.

- *AutresAgents* : Chacun fait un tri sur le couple (Mots-clés, Résultats) envoyé par l'agent. Puis chacun envoie la liste de résultats triée en fonction de sa liste.
- *MoteurRecherche* : Donne le résultat initial (comme Google).

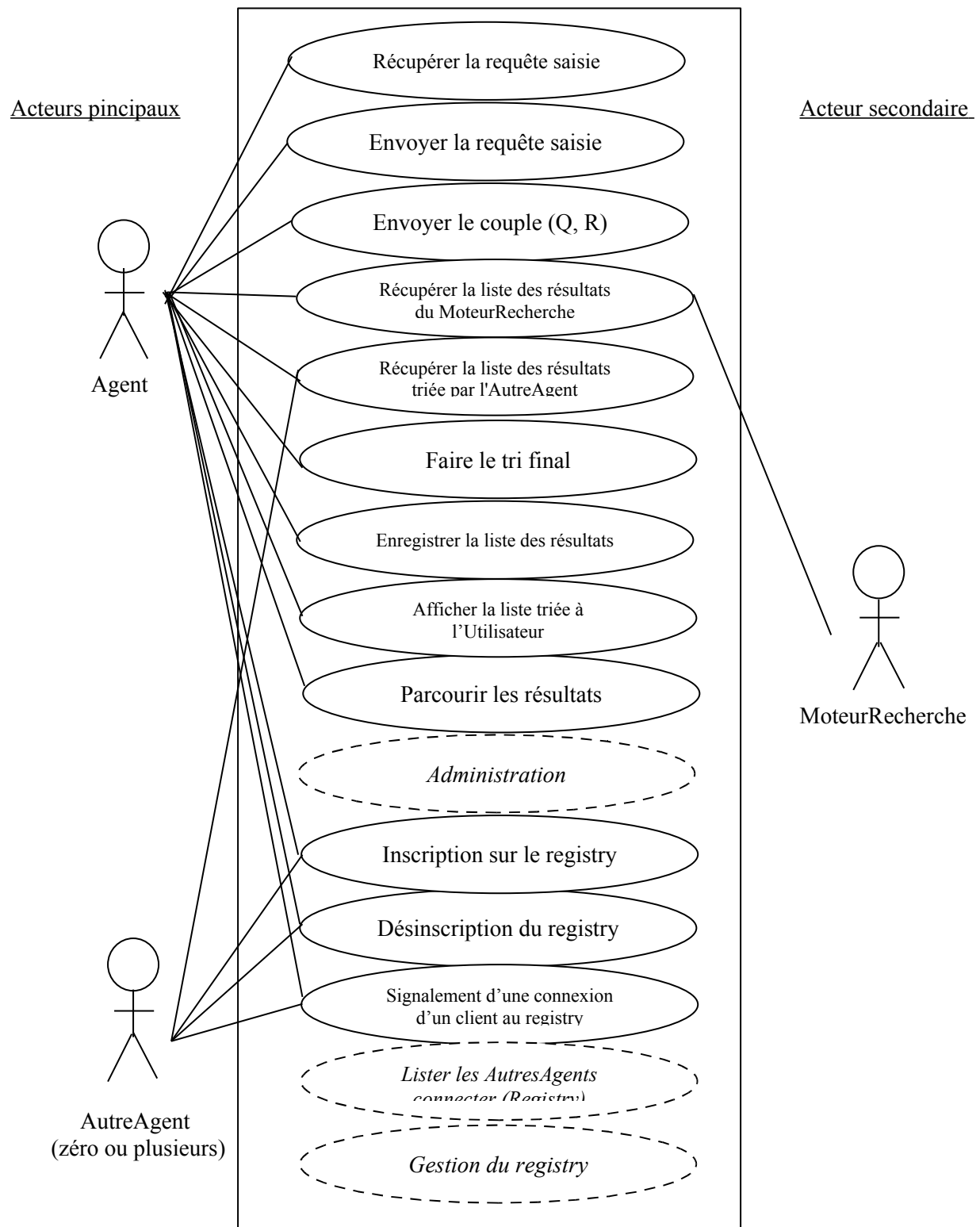
#### **Sous cas d'utilisation :**

- Récupérer la requête saisie,
- Envoyer la requête saisie,
- Envoyer le couple (Q, R) (correspondant aux mots clés et à la liste des résultats du MoteurRecherche),
- Récupérer la liste des résultats du MoteurRecherche,
- Récupérer la liste des résultats triée par l'AutreAgent,
- Faire le tri final,
- Enregistrer la liste des résultats (dans la base de données),
- Afficher la liste triée à l'Utilisateur,
- Parcourir les résultats (Clics sur les liens puis alimentation de la Base de Données),
- Administration (accès aux fichiers de log et de configuration),
- Inscription sur le registry,
- Désinscription du registry,
- Signalement d'une connexion d'un client au registry,
- Lister les AutresAgents connecter (réponse du registry),
- Gestion du registry (timeout et autres).

## 2) Diagramme des cas d'utilisation



Reprenons les sous cas d'utilisation précédents et effectuons le diagramme de ces sous cas d'utilisation.



### **3) Développement des cas d'utilisation**

Etant pauvre en cas d'utilisation, nous développerons tous les cas d'utilisation touchent l'utilisateur.

- Cas d'utilisation** : Faire une recherche  
**Portée** : Méta-Moteur de recherche sur le Web orienté communauté  
**Niveau** : Résumé  
**Le contexte d'intention** : L'intention de l'Utilisateur est d'obtenir une liste de résultats pertinente et communautaire d'une recherche via une requête particulière  
**Acteur primaire** : Utilisateur  
**Intérêts principaux** : L'Utilisateur veut obtenir un résultat le plus pertinent possible  
**Préconditions** : Avoir installer un agent sur la machine  
**Minimum de garanties** : Obtenir un résultat dans les délais  
**Garanties du succès** : Obtenir un résultat optimisé et trié avec la participation des autres agents en un minimum de temps  
**Déclencheur** : L'Utilisateur fait une requête (remplir le formulaire)  
**Scénario principal** : L'Utilisateur va faire sa recherche  
L'Utilisateur répète les étapes 1 à 6 autant de fois qu'il le désire  
1.L'Utilisateur demande le formulaire via l'adresse URL (se connecter à un agent)  
2.Le système lui envoie la page correspondant au formulaire (affichage)  
3.L'Utilisateur remplit le formulaire  
4.Le système établit une liste de résultats pertinente (tri...)  
5.Le système affiche des résultats à l'Utilisateur  
**Extensions** : 2a. Le système ne peut lui fournir la page correspondant au formulaire  
2a1. Le système informe l'Utilisateur ;  
le cas d'utilisation échoue.

**Cas d'utilisation** : Installer un agent  
**Portée** : Méta-Moteur de recherche sur le Web orienté communauté  
**Niveau** : Sous fonction  
**Le contexte d'intention** : L'intention de l'Utilisateur est d'installer un agent sur sa machine  
**Acteur primaire** : Utilisateur  
**Intérêts principaux** : L'Utilisateur veut installer un agent  
**Préconditions** : Avoir une machine virtuelle Java d'installée sur sa machine et avoir les droits d'installation  
**Minimum de garanties** : Au pire, affichage du problème détecté  
**Garanties du succès** : Agent est installé et en écoute  
**Scénario principal** : L'Utilisateur souhaite installer un agent sur sa machine  
     1.L'Utilisateur se connecte sur son compte et exécute le fichier de compilation et d'installation  
     2.Le système compile et installe  
     3.Le système informe l'Utilisateur que l'agent est bien installé et en écoute sur un certain port  
  
**Extensions** : 2a. Le système ne compile pas  
     2a1. Le système informe l'Utilisateur du problème;  
         le cas d'utilisation échoue  
     2b. Le script d'installation ne fonctionne pas  
     2b1. Le système informe l'Utilisateur du problème;  
         le cas d'utilisation échoue  
     3a. Le système n'arrive pas à se mettre en écoute sur le port  
     3a1. Le système l'informe ;  
         le cas d'utilisation échoue.

**Cas d'utilisation** : Désinstaller un agent  
**Portée** : Méta-Moteur de recherche sur le Web orienté communauté  
**Niveau** : Sous fonction  
**Le contexte d'intention** : L'intention de l'Utilisateur est de désinstaller un agent sur sa machine  
**Acteur primaire** : Utilisateur  
**Intérêts principaux** : L'Utilisateur veut désinstaller un agent se trouvant sur sa machine  
**Préconditions** : Avoir une les droits de suppression  
**Minimum de garanties** : Au pire, affichage du problème détecté  
**Garanties du succès** : Agent désinstallé de la machine  
**Scénario principal** : L'Utilisateur souhaite désinstaller un agent installé sur sa machine  
     1.L'Utilisateur se connecte sur son compte et exécute le fichier désinstallation  
     2.Le système désinstalle l'agent  
     3.Le système informe l'Utilisateur que l'agent est bien désinstallé  
  
**Extensions** : 2a. Le système n'arrive pas à désinstaller l'agent  
     2a1. Le système informe l'Utilisateur du problème ;  
         le cas d'utilisation échoue.



**Cas d'utilisation** : Se connecter à un agent  
**Portée** : Méta-Moteur de recherche sur le Web orienté communauté  
**Niveau** : But de l'utilisateur  
**Le contexte d'intention** : L'intention de l'Utilisateur est de se connecter à un agent  
**Acteur primaire** : Utilisateur  
**Intérêts principaux** : L'Utilisateur veut se connecter à l'agent installé sur sa machine  
**Préconditions** : Installation préliminaire de l'agent sur sa machine et avoir un navigateur Web  
**Minimum de garanties** : Au pire, affichage du problème détecté  
**Garanties du succès** : Connexion à l'agent  
**Scénario principal** : L'Utilisateur souhaite se connecter à l'agent installé sur sa machine  
     1) L'Utilisateur entre l'adresse URL dans son navigateur Web  
     2) Le système envoie le formulaire  
  
**Extensions** : 2a. Le navigateur Web ne peut lui envoyer la page correspondant au formulaire  
     2a1. Le navigateur Web informe l'Utilisateur ;  
     le cas d'utilisateur échoue.

**Cas d'utilisation** : Se déconnecter d'un agent  
**Portée** : Méta-Moteur de recherche sur le Web orienté communauté  
**Niveau** : But de l'utilisateur  
**Le contexte d'intention** : L'intention de l'Utilisateur est de se déconnecter d'un agent installé sur sa machine  
**Acteur primaire** : Utilisateur  
**Intérêts principaux** : L'Utilisateur veut se déconnecter de l'agent se trouvant sur sa machine  
**Préconditions** : L'Utilisateur doit déjà être connecté à l'agent sur sa machine  
**Minimum de garanties** : Au pire, affichage du problème détecté  
**Garanties du succès** : Déconnexion de l'agent  
**Scénario principal** : L'Utilisateur souhaite se déconnecter d'un agent auquel il est déjà connecté sur sa machine  
     1. L'Utilisateur ferme le navigateur Web  
     2. Le navigateur Web se ferme  
  
**Extensions** : 2a. La fermeture échoue ; le cas d'utilisation échoue.

- Cas d'utilisation** : Remplir le formulaire  
**Portée** : Méta-Moteur de recherche sur le Web orienté communauté  
**Niveau** : But de l'utilisateur  
**Le contexte d'intention** : L'intention de l'Utilisateur est de remplir le formulaire pour obtenir une liste de résultats.  
**Acteur primaire** : Utilisateur  
**Intérêts principaux** : L'Utilisateur souhaite obtenir le formulaire et le remplir  
**Préconditions** : Avoir installé un agent et être connecté (avoir en affichage le formulaire)  
**Minimum de garanties** : L'Utilisateur complète le formulaire  
**Garanties du succès** : L'Utilisateur complète le formulaire et le système le valide  
**Scénario principal** : L'Utilisateur souhaite remplir le formulaire  
     1. L'Utilisateur remplit les champs du formulaire  
     2. Le système vérifie et valide les champs du formulaire  
     3. Le système affiche que les champs sont corrects et que la recherche et en cours
- Extensions** : 1a. L'Utilisateur ne veut plus remplir le formulaire ;  
                   le cas d'utilisation échoue.  
                   2a. Le système ne peut pas valider le formulaire, le système a remarqué que les informations fournies sont incorrectes  
                   2a1. Le système informe l'Utilisateur;  
                   le cas d'utilisation continue à l'étape 1.

- Cas d'utilisation** : Etablir une liste de résultats pertinente  
**Portée** : Méta-Moteur de recherche sur le Web orienté communauté  
**Niveau** : But de l'utilisateur  
**Le contexte d'intention** : L'intention de l'Utilisateur est d'obtenir une liste de résultats pertinente  
**Acteur primaire** : Utilisateur  
**Intérêts principaux** : L'Utilisateur souhaite obtenir le formulaire et le remplir  
**Préconditions** : Avoir installé un agent et être connecté  
**Minimum de garanties** : Obtenir une liste de résultats  
**Garanties du succès** : Obtenir une liste de résultats optimale  
**Scénario principal** : L'Utilisateur souhaite obtenir une liste de résultats pertinente  
     1. Le système récupère la liste des résultats du MoteurRecherche  
     2. Le système fait le tri  
     3. Le système enregistre la liste des résultats (alimentation de la base de données)
- Extensions** : 2a. Le système est incapable d'obtenir un résultat  
                   2a1. Il ne peut établir une liste de résultats à cause de l'indisponibilité des moteurs de Recherche (timeout...) ;  
                   le cas d'utilisation échoue  
                   2a2. Il ne peut établir une liste de résultats à cause du temps de réponse trop important des autres agents (timeout...) ;  
                   le cas d'utilisation échoue  
                   2a3. Il ne peut établir une liste de résultats optimale à cause du peu de temps qui lui reste à l'agent pour effectuer son tri final ;  
                   le cas d'utilisation échoue  
                   3a. Impossibilité d'accéder à la base de données;  
                   le cas d'utilisation échoue.

*Reprenons les sous cas d'utilisation et tentons d'y développer les principaux.*

**Cas d'utilisation** : Récupérer la liste des résultats du MoteurRecherche  
**Portée** : Méta-Moteur de recherche sur le Web orienté communauté  
**Niveau** : Sous fonction  
**Le contexte d'intention** : L'intention de l'Agent est de récupérer la liste de résultats établie par le MoteurRecherche  
**Acteur primaire** : Agent  
**Intérêts principaux** : L'Agent souhaite récupérer la liste des résultats du MoteurRecherche  
**Préconditions** : L'Agent a effectué une recherche  
**Minimum de garanties** : Obtenir une liste de résultats du MoteurRecherche  
**Scénario principal** : L'Agent souhaite obtenir la liste des résultats du MoteurRecherche  
1. L'Agent envoie la requête au MoteurRecherche  
2. Le MoteurRecherche fournit une liste de résultats  
  
**Extensions** : 2a. Le MoteurRecherche est incapable de fournir une liste de résultats (timeout...); le cas d'utilisation échoue.

**Cas d'utilisation** : Récupérer la liste des résultats triée par l'AutreAgent  
**Portée** : Méta-Moteur de recherche sur le Web orienté communauté  
**Niveau** : Sous fonction  
**Le contexte d'intention** : L'intention de l'Agent est de récupérer le tri d'un AutreAgent sachant qu'il lui a fourni au préalable la liste des résultats établie par le MoteurRecherche et les mots clés.  
**Acteur primaire** : Agent  
**Intérêts principaux** : L'Agent souhaite obtenir le tri d'un AutreAgent  
**Préconditions** : Avoir la liste des AutreAgents connectés (par le biais du registry : lister les AutreAgents connectés) et avoir fourni à ces AutreAgents la liste des résultats et des mots clés  
**Minimum de garanties** : Obtenir la liste de résultats du MoteurRecherche  
**Garanties du succès** : Obtenir une liste de résultats triée  
**Scénario principal** : L'Agent souhaite obtenir une liste de résultats triée par l'AutreAgent  
1. L'AutreAgent vérifie dans sa base de données la présence de la liste des mots clés fournie par l'Agent  
2. L'AutreAgent fait son tri par rapport à ses données  
3. L'AutreAgent envoie la liste triée à l'Agent  
  
**Extensions** : 1a. L'AutreAgent ne possède pas ces mots clés dans sa base de données  
1a1. L'AutreAgent informe l'Agent;  
le cas d'utilisation échoue  
2a. L'AutreAgent est incapable de faire un tri;  
le cas d'utilisation échoue  
3a. Impossibilité d'envoyer la liste triée (déconnexion de l'Agent ou de l'AutreAgent, ou autre);  
le cas d'utilisation échoue.

**Cas d'utilisation** : Faire le tri final  
**Portée** : Méta-Moteur de recherche sur le Web orienté communauté  
**Niveau** : Sous fonction  
**Le contexte d'intention** : L'intention de l'Agent est d'effectuer le tri final avec éventuellement le tri des AutresAgents sur la requête formulée par l'Utilisateur.  
**Acteur primaire** : Agent  
**Intérêts principaux** : L'Agent souhaite faire le tri final  
**Préconditions** : L'Agent a envoyé à tous les AutresAgents connectés la liste des résultats du MoteurRecherche et les mots clés  
**Minimum de garanties** : Avoir la liste de résultats du MoteurRecherche  
**Garanties du succès** : Obtenir une liste de résultats triée par rapport aux listes des AutresAgents  
**Scénario principal** : L'Agent souhaite faire le tri final
 

1. L'Agent récupère la liste des résultats triée de chaque AutreAgent connecté
2. L'Agent, par rapport à la (les) liste(s) qu'il détient désormais, fait un tri.

**Extensions** :
 

- 1a. L'Agent est incapable de récupérer les listes des AutresAgents.  
L'Agent ne considérera que la liste de résultats établie par le MoteurRecherche ;  
le cas d'utilisation continue à l'étape 2.
- 2a. L'Agent est incapable de faire le tri, de fournir une liste de résultats à la sortie (timeout ou autre) ;  
le cas d'utilisation échoue.

**Cas d'utilisation** : Enregistrer la liste des résultats (dans la base de données)  
**Portée** : Méta-Moteur de recherche sur le Web orienté communauté  
**Niveau** : Sous fonction  
**Le contexte d'intention** : L'intention de l'Agent est d'enregistrer la liste des résultats triée et les mots clés dans la base de donnée locale de l'Utilisateur.  
**Acteur primaire** : Agent  
**Intérêts principaux** : L'Agent souhaite alimenter la base de données  
**Préconditions** : Avoir les mots clés et la liste des résultats établie  
**Minimum de garanties** : Enregistrer les mots clés et la liste des résultats dans la base de données  
**Scénario principal** : L'Agent souhaite alimenter la base de données
 

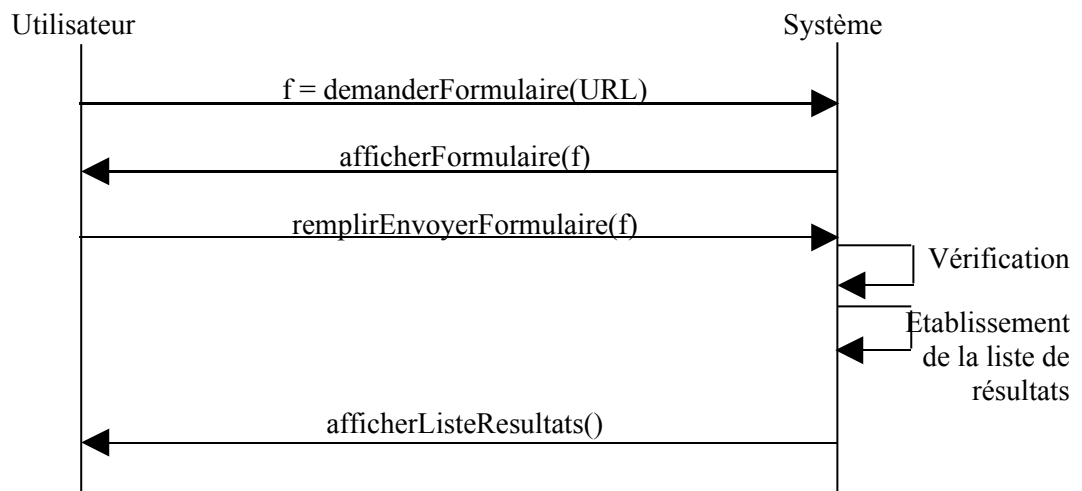
1. L'Agent accède en écriture à la base de donnée locale de l'Utilisateur
2. L'Agent enregistre et valide (« commit ») les données dans la base

**Extensions** :
 

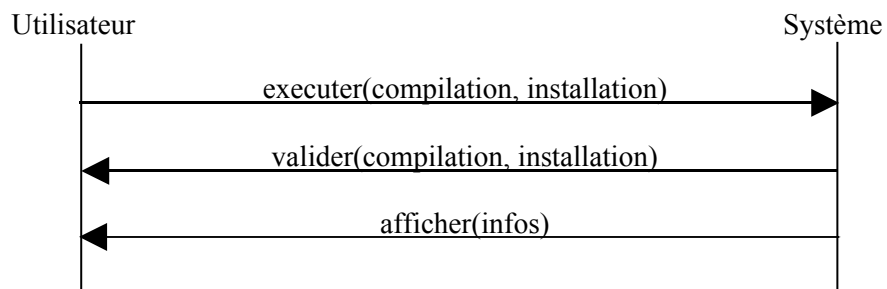
- 1a. L'Agent est incapable d'accéder à la base de données ;  
le cas d'utilisation échoue
- 2a. L'Agent est incapable d'écrire les données dans la base de données ;  
le cas d'utilisation échoue.

#### 4) Diagrammes de séquence

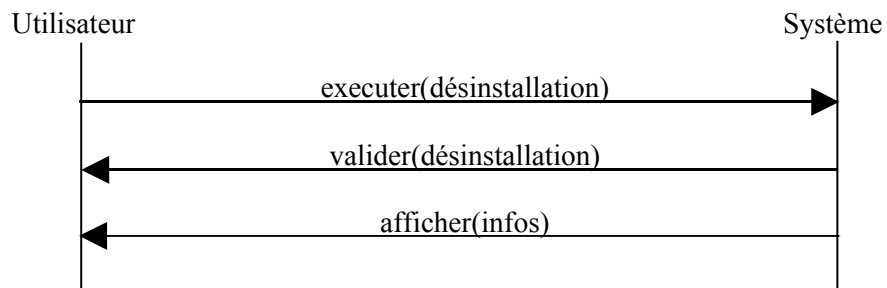
➤ *Faire une recherche*



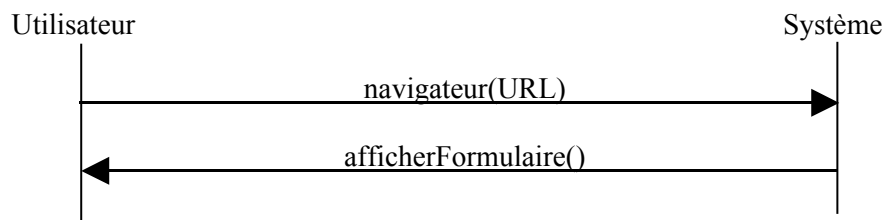
➤ *Installer un agent*



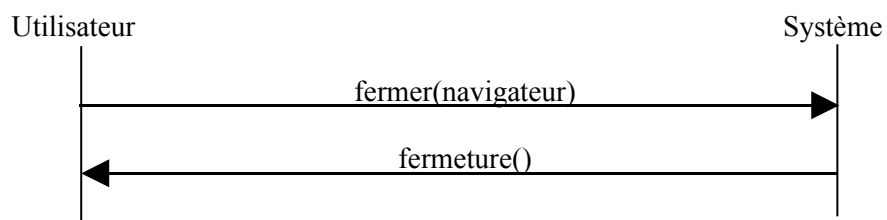
➤ *Désinstaller un agent*



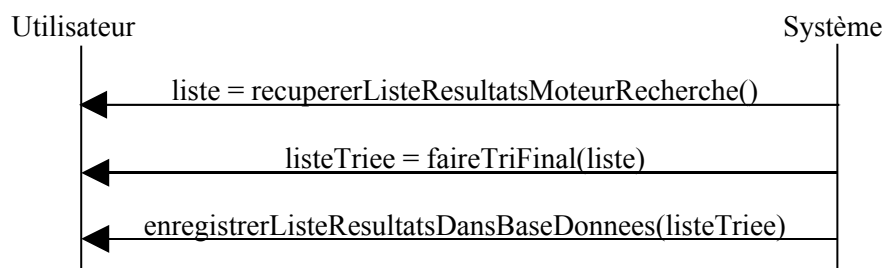
➤ *Se connecter à un agent*



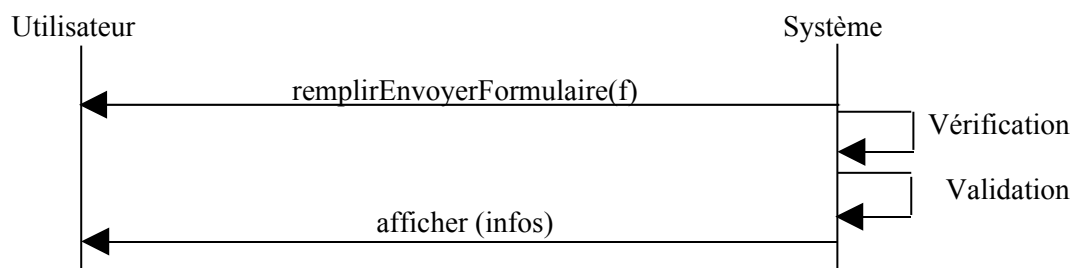
➤ *Se déconnecter d un agent*



➤ *Etablir une liste de résultats pertinente*



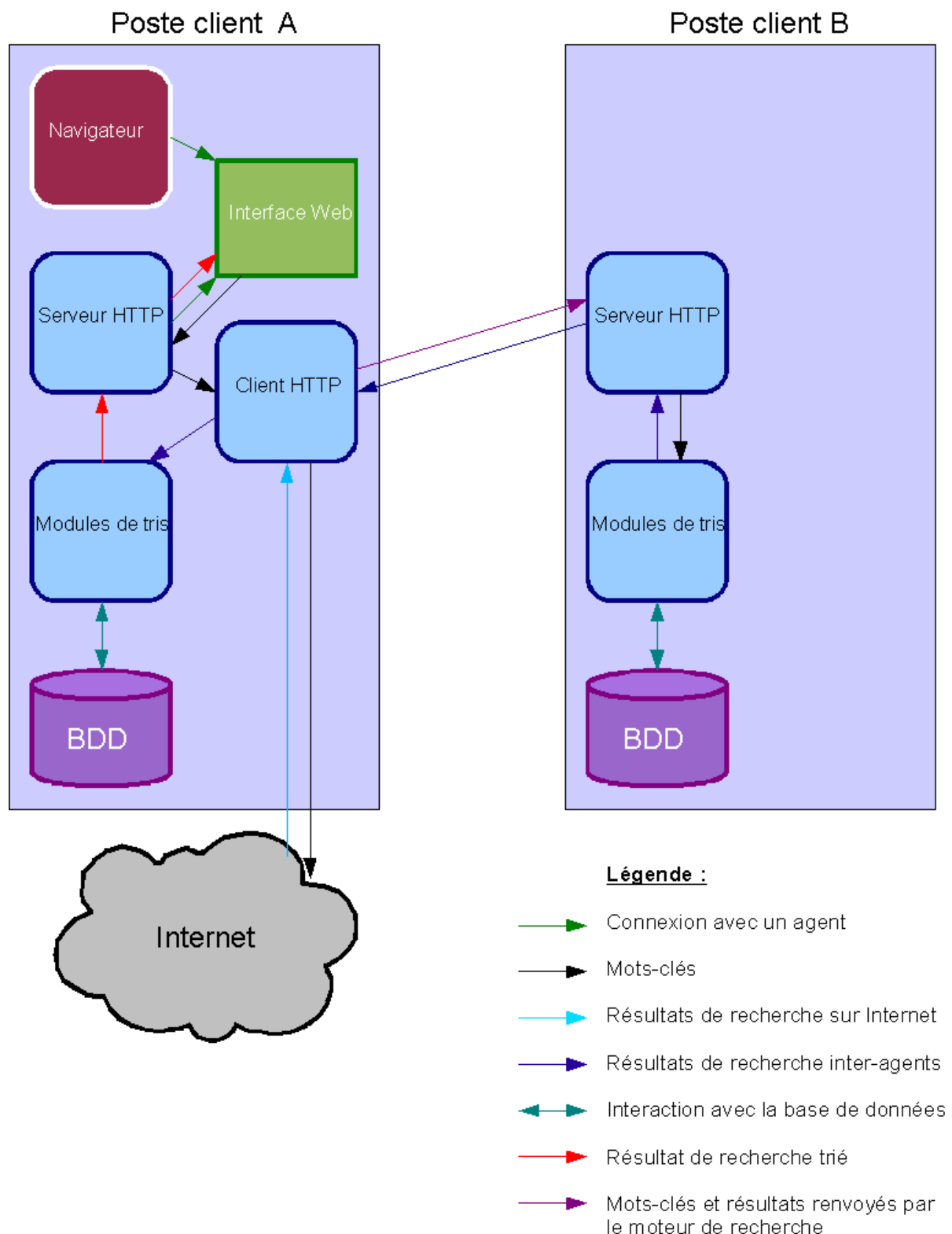
➤ *Remplir le formulaire*



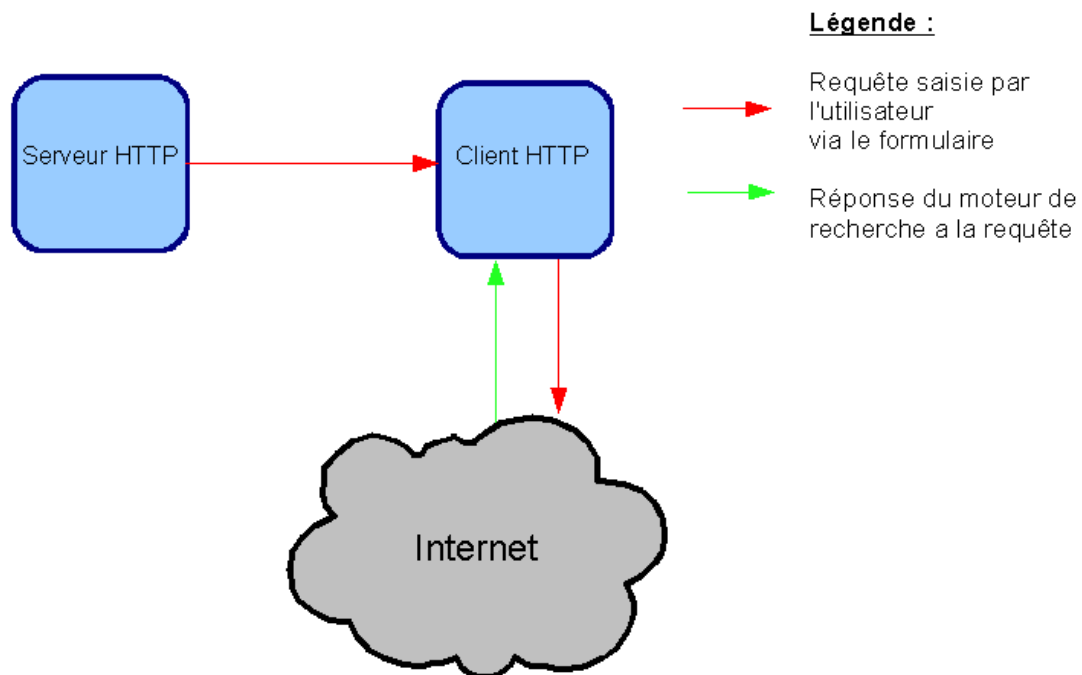
# II/ Architecture globale

## 1)Présentation générale

La suite de ce document présente les aspects liés à l'implémentation du projet, d'un point de vue global le plus indépendant des techniques de codage utilisées. Il décrit succinctement la conception globale du Méta-moteur. Seront donc exposés l'architecture logicielle de l'application, les flux de données et les différentes interfaces (messages, structures de données) qui lient les modules de l'application.

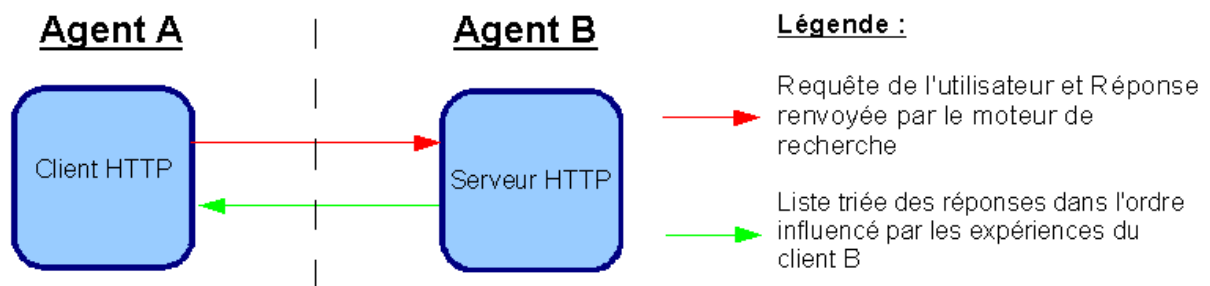


## 2) Messages Serveur / Client



Une fois la requête récupérée par le Serveur HTTP, celui-ci la transmet au client HTTP. Le client HTTP envoie la requête au moteur de recherche choisi par l'utilisateur (cela fera partie de la requête adressée par le serveur) et récupère la réponse de celui-ci.

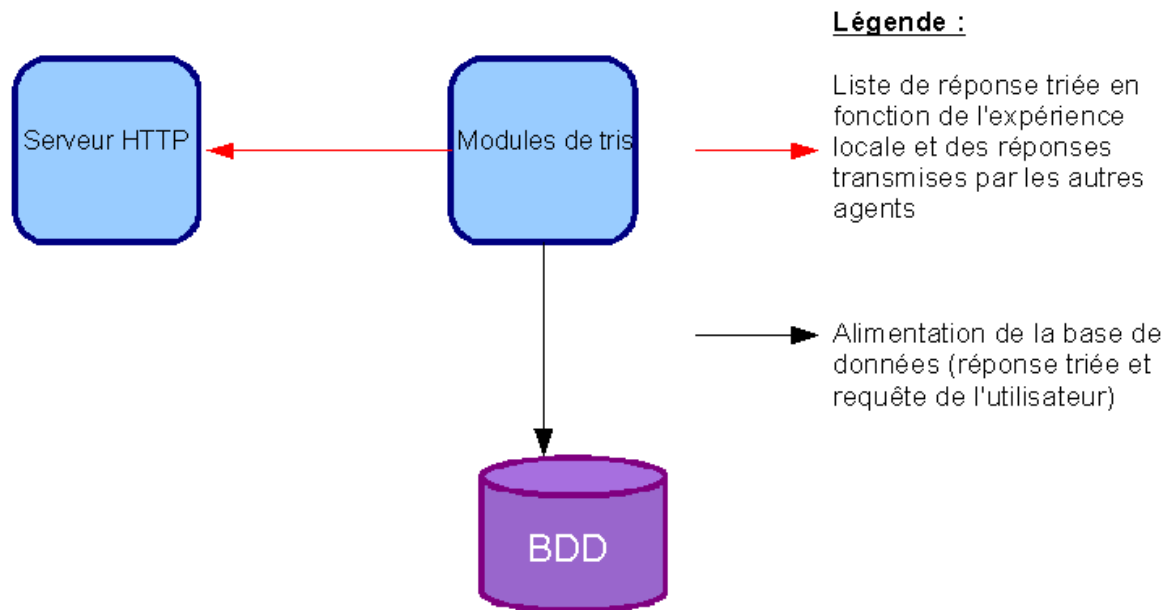
## 3) Messages Agent / Agent



L'agent A, via son client HTTP envoie aux autres agents la requête de l'utilisateur ainsi que la réponse du moteur de recherche, les autres agents reçoivent tout ceci via leur serveur HTTP puis, après tri via leurs modules de tri, leur serveur HTTP renvoient les réponses triées.

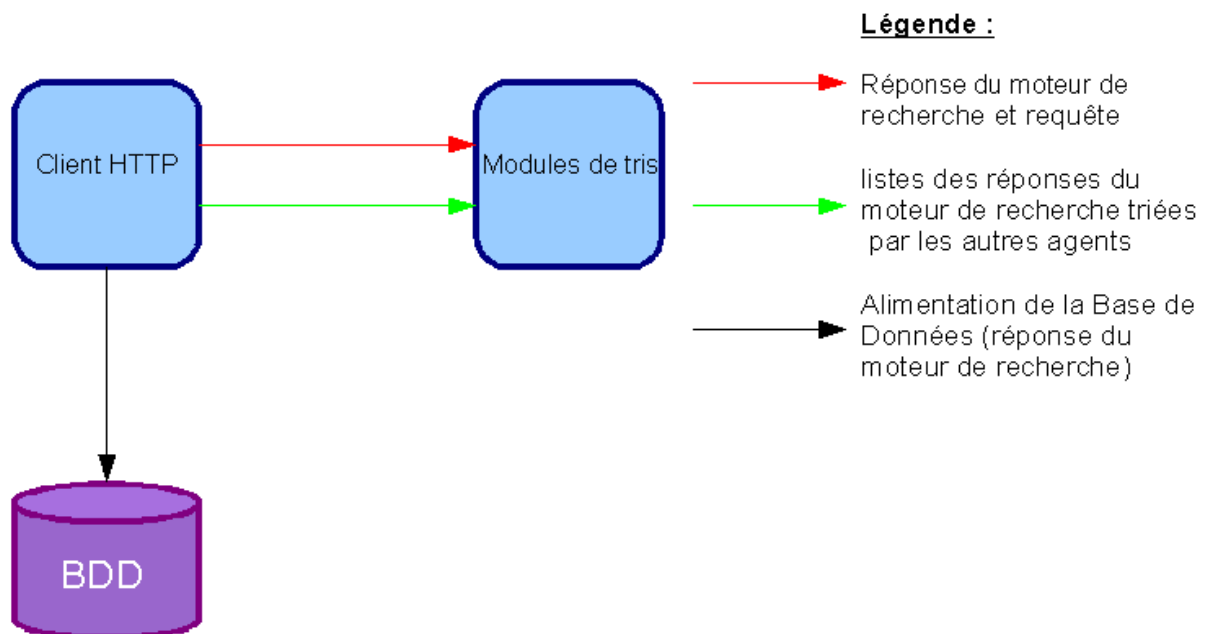


#### 4) Messages Serveur / Modules de tris



Après avoir fait leurs tris (le tri sur la requête et le résultat ou le tri en fonction des réponses des autres agents, ou les 2), les modules de tris renvoient au serveur HTTP la réponse triée, le serveur construit ensuite une page HTML afin d'envoyer cette réponse à l'utilisateur.

#### 5) Messages Clients / Modules de tris



Le client HTTP envoie aux modules de tris la réponse du moteur de recherche à la requête ainsi que la requête afin que ceux-ci puissent démarrer le tri local puis, après récupération des listes triées des réponses par les autres agents, le client HTTP les envoie aux modules de tri pour qu'ils puissent intégrer ces réponses au tri à renvoyer à l'utilisateur.