Introduction

=> L’environnement

=> Lancement du projet

=> Cahier des charges

=> La planification

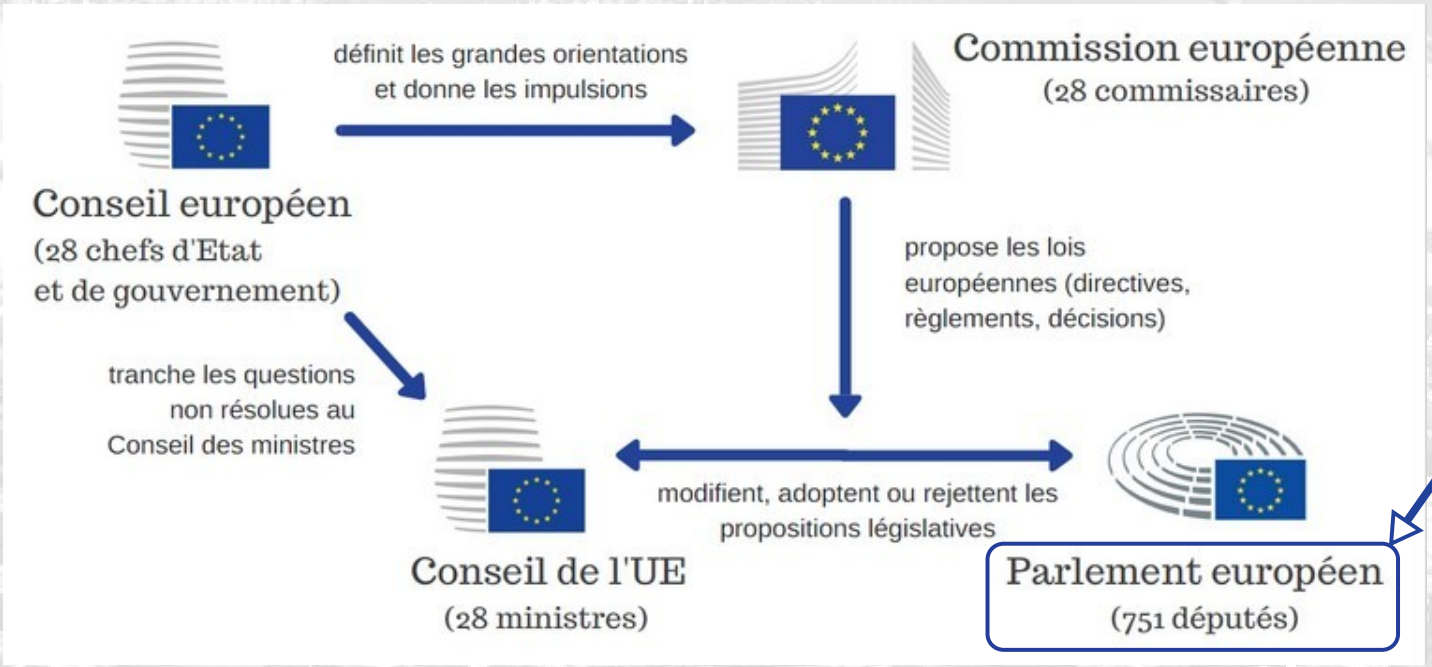
=> Une phase de recherche

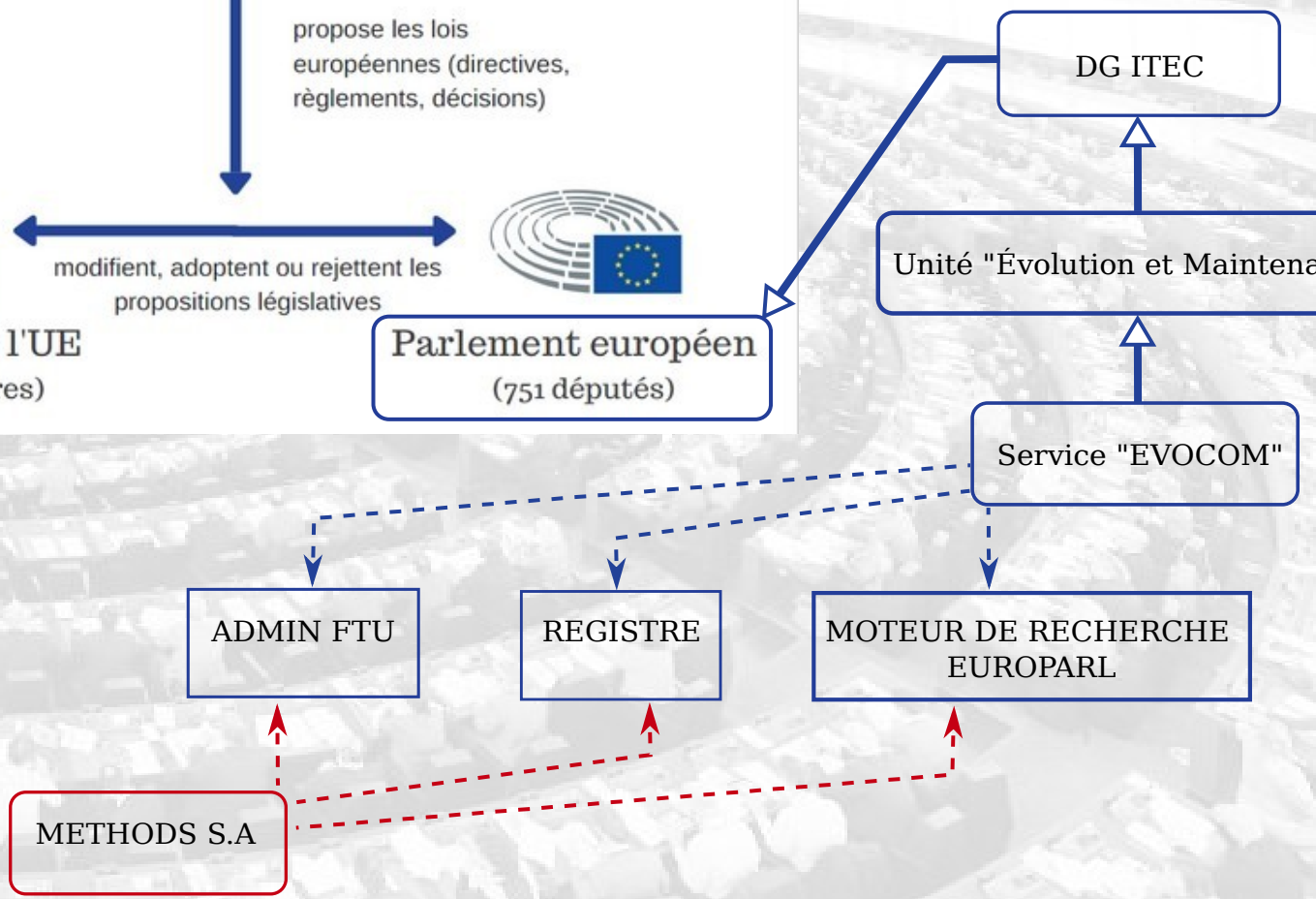
=> La réalisation

=> La validation et la documentation

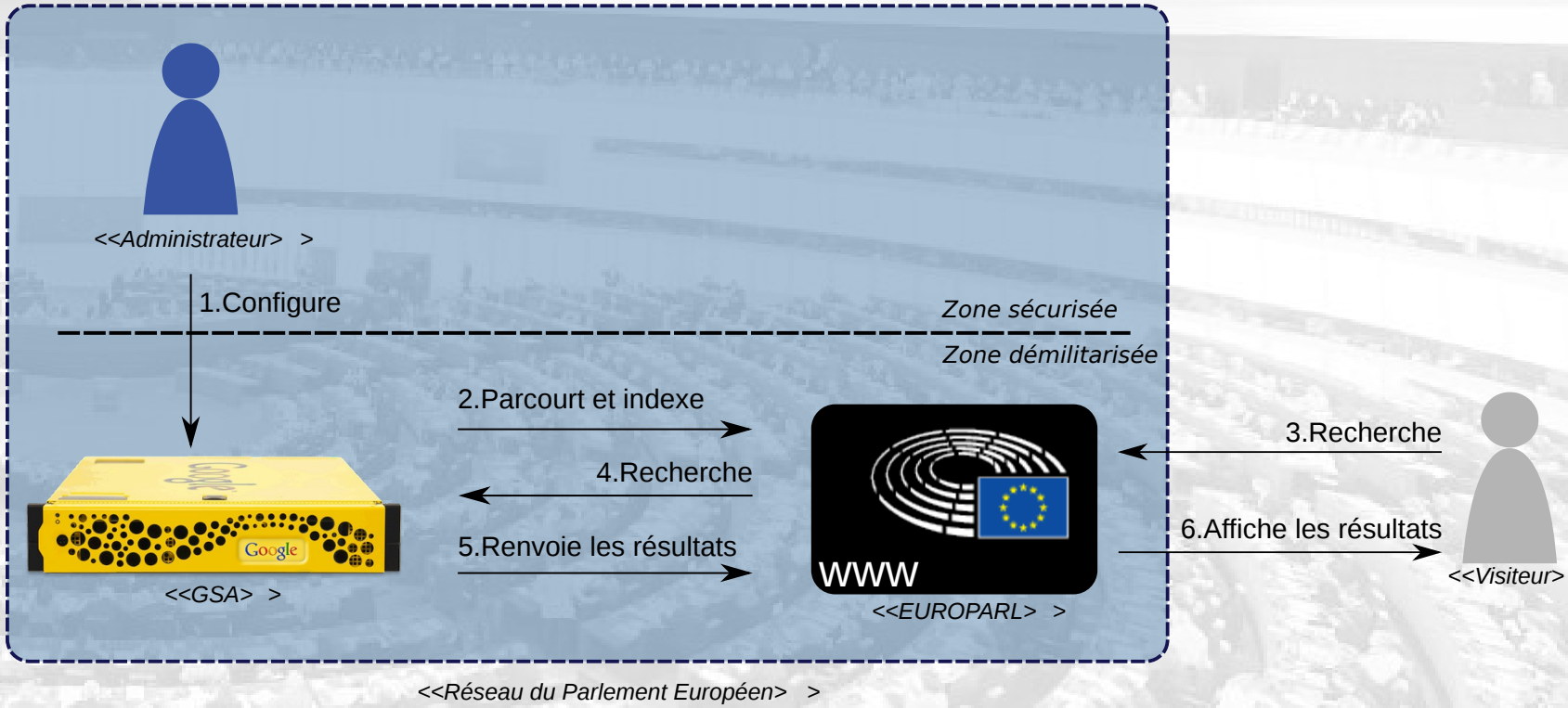
=> Retour d’expérience

Statut et mission au Parlement Européen





Situation initiale et lancement du projet



LES LACUNES :

Solution « tout en un » => hébergement par l'unité « Opérations »

=> Méconnaissance du système

=> Aucune optimisation **=> Résultat Insatisfaisants <=** AUDIT DE LA DGCOM

FIN DES LICENCES ET ARRÊT DU SUPPORT => Lancement du projet en novembre 2016

Cahier des charges : expression de besoins

EUROPARL contient 240 VLP (Version Linguistique d'une Planète)

Fonctions d'usages :

- Proposer à l'utilisateur une recherche textuelle sur la VLP courante qui fournisse des résultats « pertinents » .

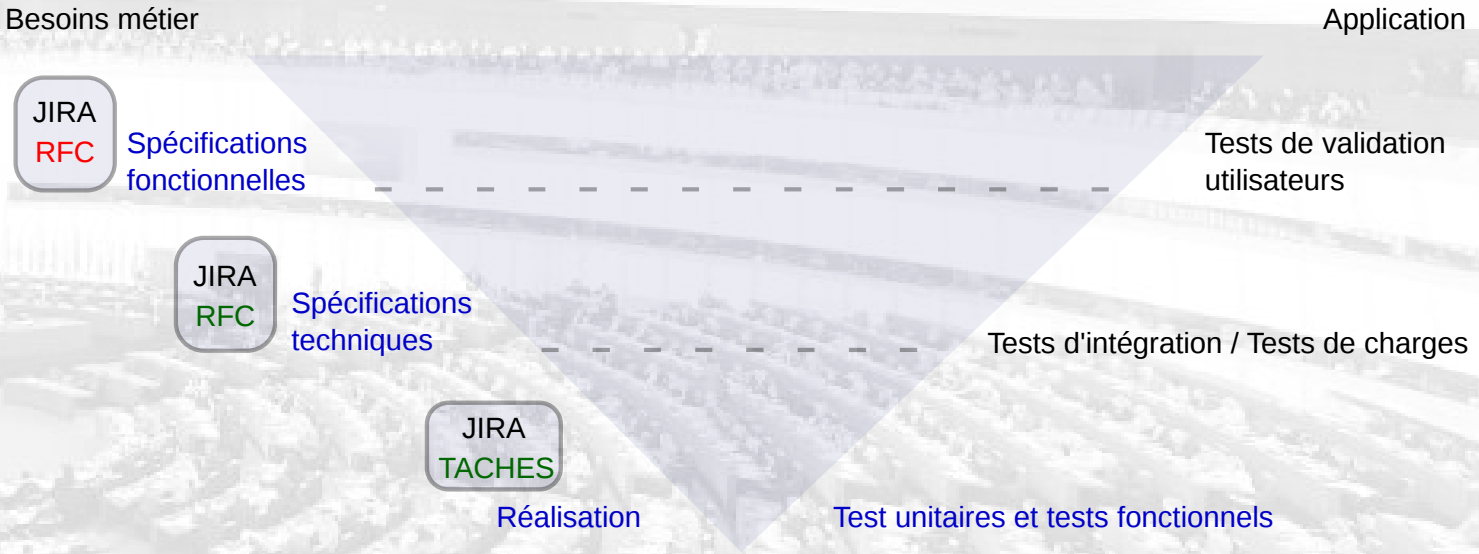
- Recherche par mots-clés et catégories
- Résultats triables et surlignés
- Auto-complétion

Fonctions contraintes :

- Étendre une recherche à toutes les VLP de la langue sélectionnée.
- Configuration : fréquences d'indexation, gestion des pages obsolètes, structure des pages
- Ajouter ou supprimer un VLP de façon « dynamique »

Cahier des charges : contexte du projet

ASTRA :



note *: Les tâches en bleu, incombent à l'auditeur*

Standards ALSA :

Tomcat 8 / Spring / Elasticsearch / Plateforme d'Intégration continue

CLOUD : Projet pilote

Planification du projet

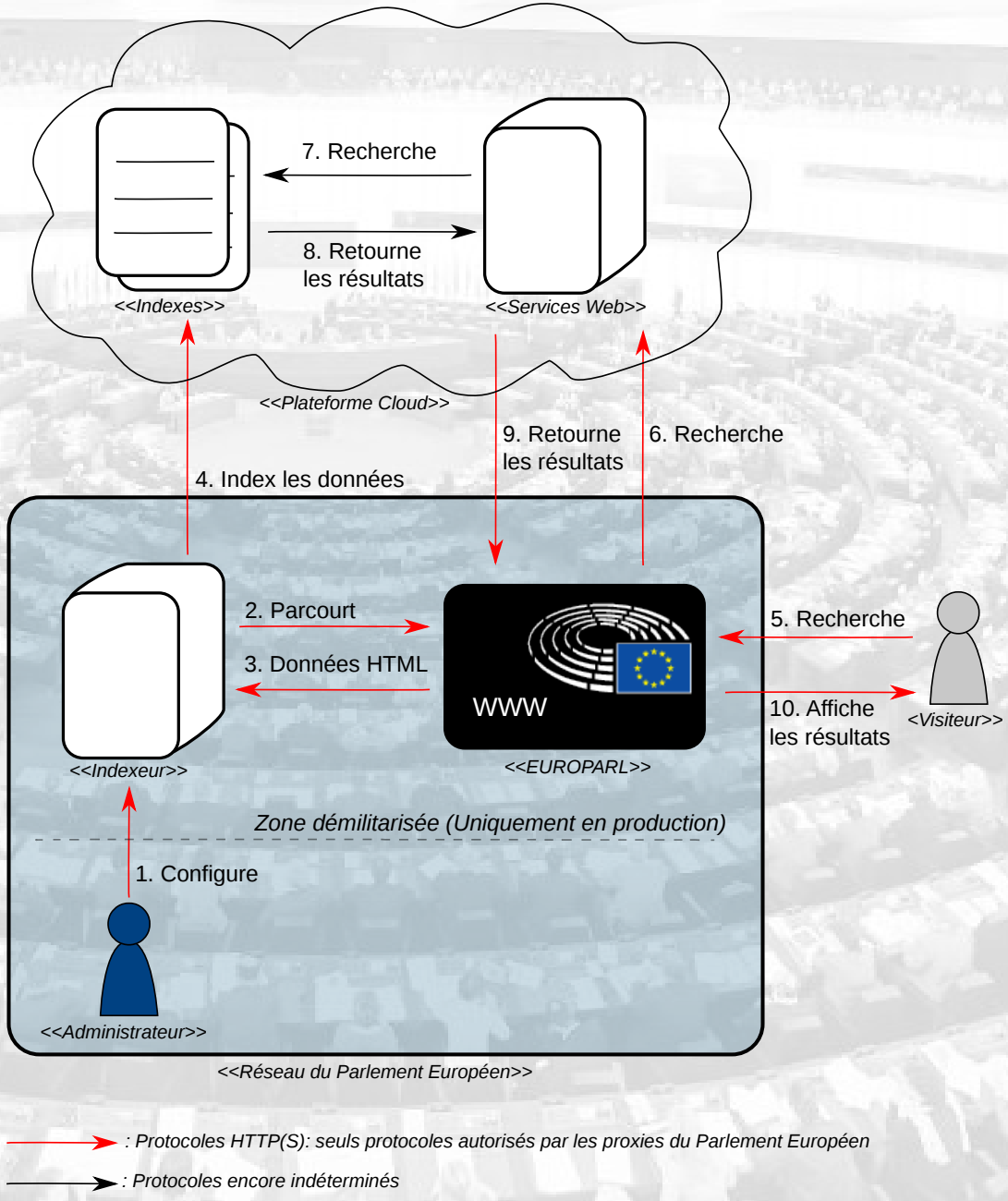
Projet 1 : POC et étude des solutions cloud (10 j.h)

Analyse fonctionnel externe	1
POC du module d'indexation	3
Tests et comparatif des solutions cloud	6

Projet 2 : Réalisation (177 j.h)

Analyse fonctionnel interne	20
Réalisation de l'indexeur	50
Réalisation des services web de recherche + librairie cliente	12
Intégration dans EUROPARL	3
Intégration continue et tests	61
Rédaction de la documentation et formation	29
Préparation et déploiement des livrables	2

Recherche : Achitecture prévisionnel



=> services web REST :

Légèreté du protocole

Facilité d'intégration dans les interfaces web

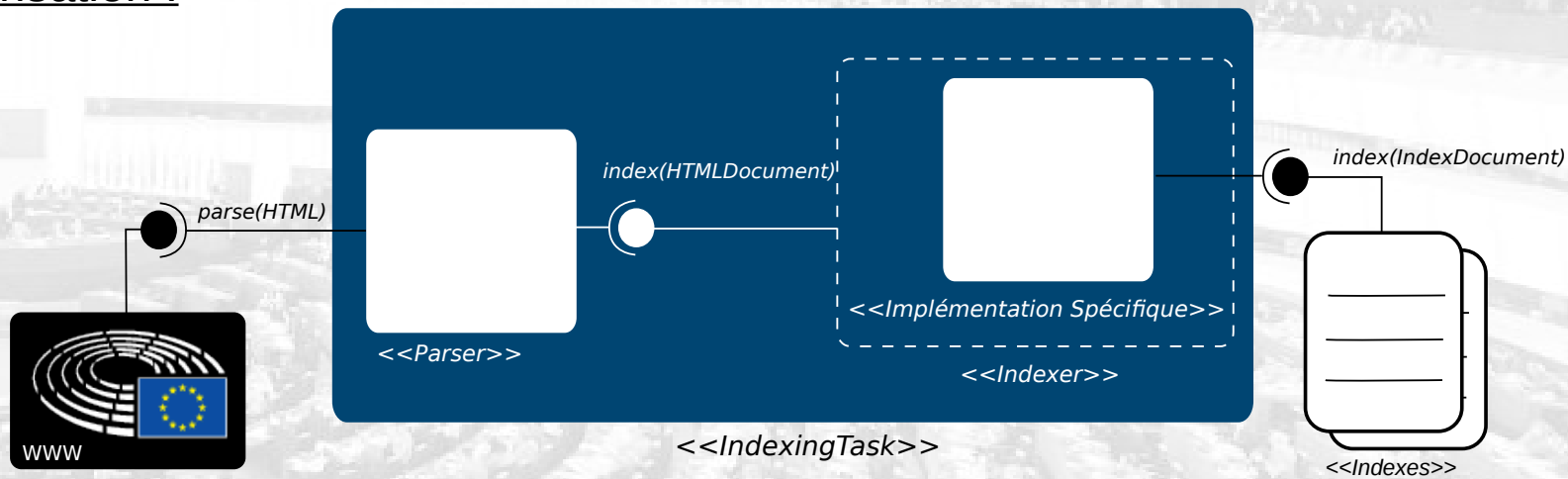
=> Interfaces : SPA en Javascript

Déplacement de la charge

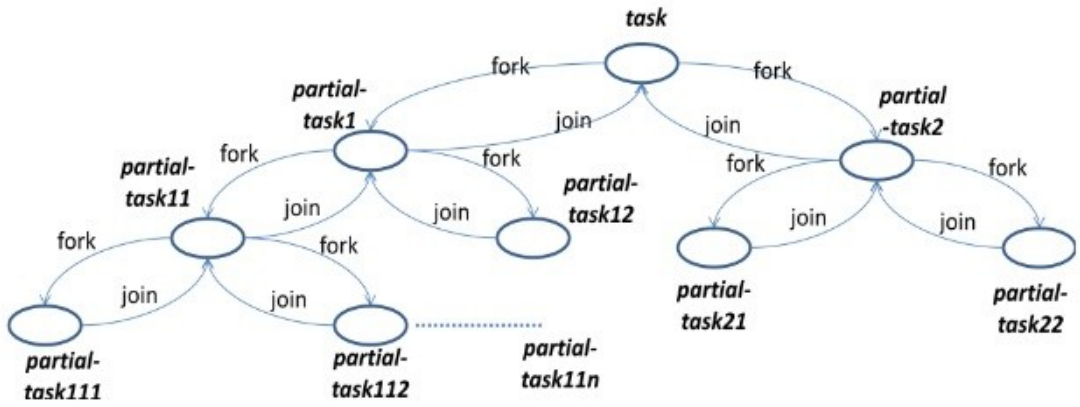
MVC

Recherche : POC du module d'indexation

Interfaces et modularisation :



Division des tâches d'indexations :



How the Fork/Join framework uses divide-and-conquer to complete the task

Recherche : Les solutions d'indexation

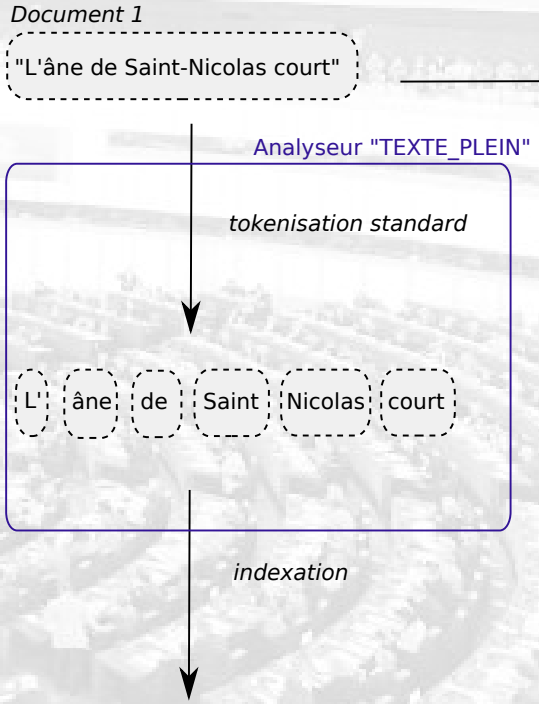
Analyse textuelle

=

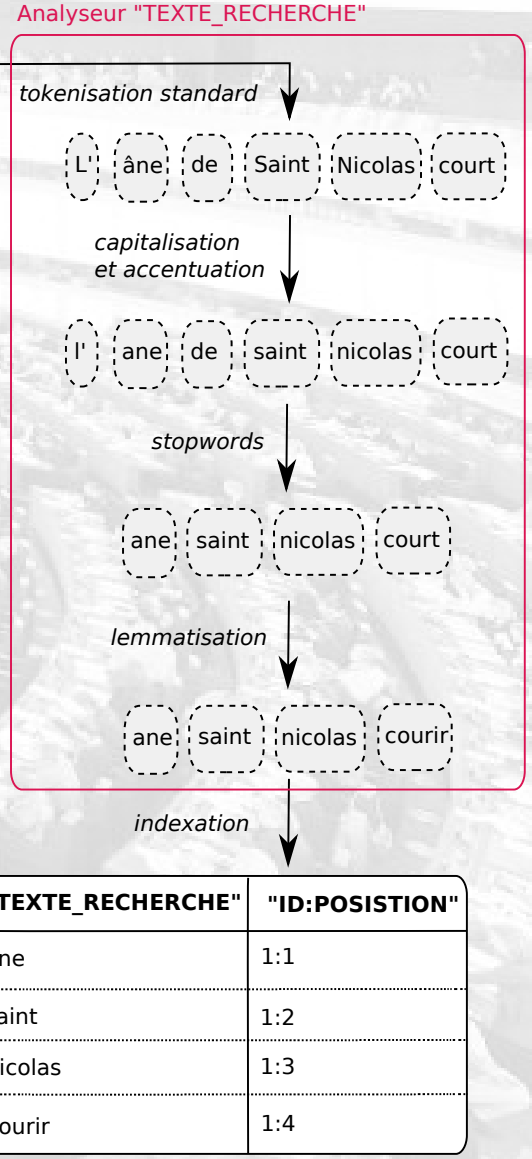
Découpe de la phrase

+

Une chaîne de traitement



"TEXTE_PLEIN"	"ID:POSISTION"
L'	1:1
âne	1:2
de	1:3
Saint	1:4
Nicolas	1:5
court	1:6



"TEXTE_RECHERCHE"	"ID:POSISTION"
ane	1:1
saint	1:2
nicolas	1:3
courir	1:4

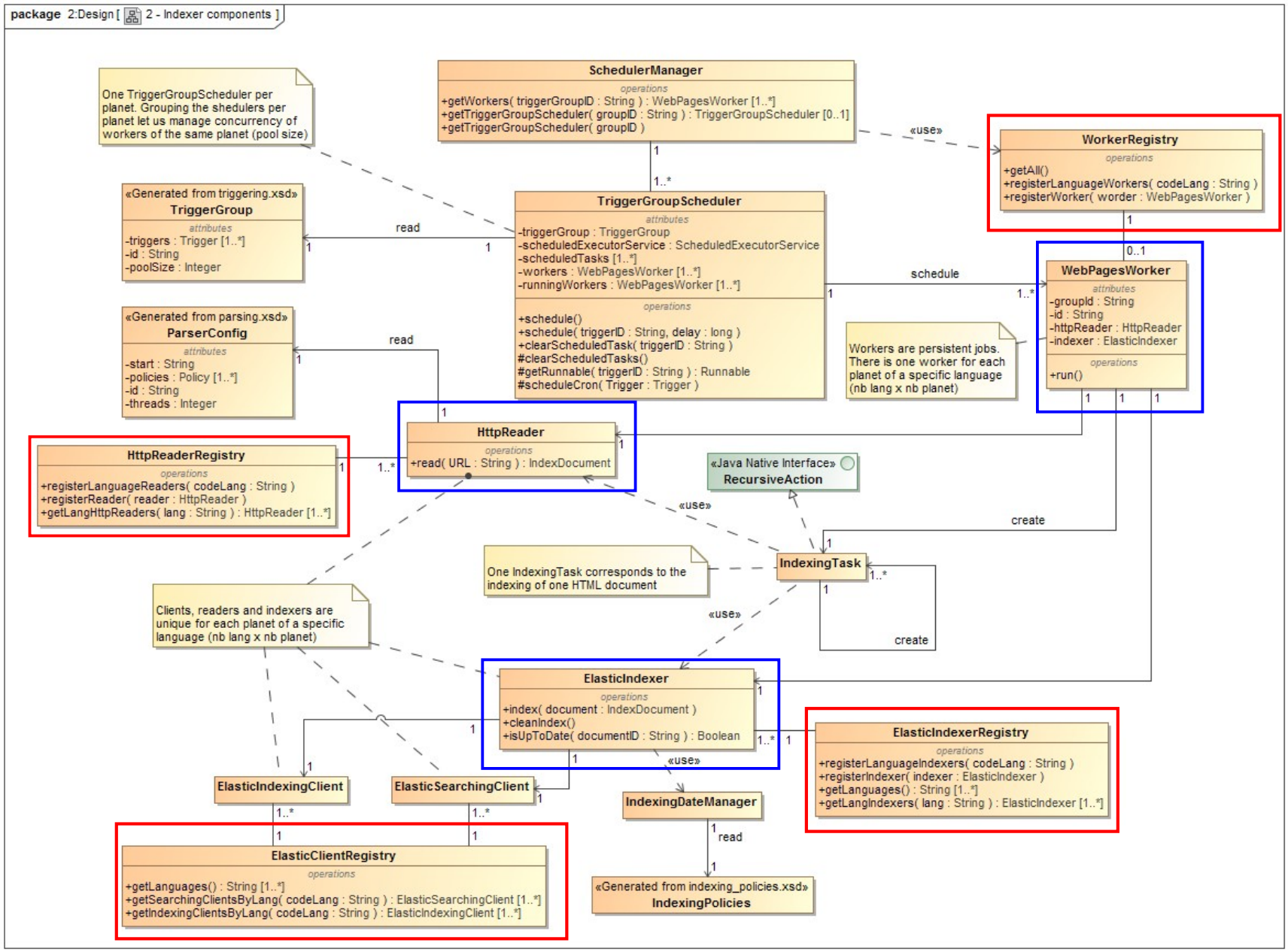
Recherche : Choix de la solution d'indexation

Critère	AzurSearch	CloudSearch	Elasticsearch Amazon	Elasticsearch Interne
Répond aux besoins non négociables (Flexibilité nulle)	OUI	OUI	OUI	OUI
Pertinence des résultats	BONNE	MOYENNE	BONNE	BONNE
Répond aux besoins négociables (Flexibilité non nulle)	EN PARTIE	EN PARTIE	OUI	OUI
Facilité d'implémentation	FACILE	FACILE	COMPLEXE	COMPLEXE
Facilité de déploiement	FACILE	FACILE	FACILE	LABORIEUSE

1er choix : Azure => Simplicité et efficacité

Solution retenue : Amazon

Réalisation : Les registres (1)



Réalisation : Les registres (2)

Permettre l'ajout « dynamique » d'une planète :

=> Déclaration dans des fichiers de configuration

=> Créer des composants spécifiques pour chaque VLP

=> Stocker en mémoire les composants dans des registres dédiés

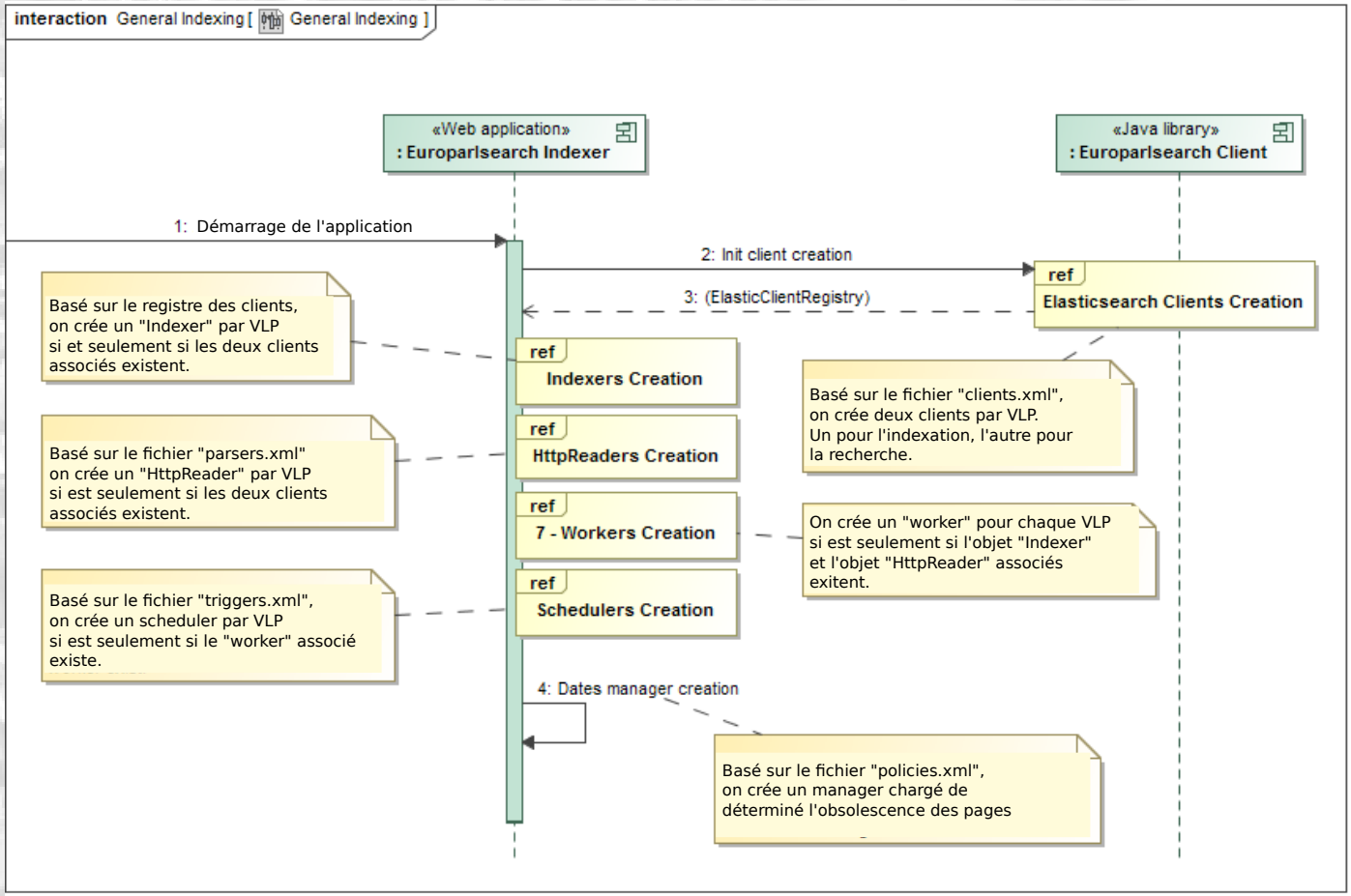
Configurations :

domains.xml

clients.xml

parsers.xml

triggers.xml



Réalisation : La planification des travaux

```
<TRIGGERING>
    [.............]
    <TRIGGER_GROUP id="news" poolsize="1">
        [.............]
        <TRIGGER id= "da">0 0 8 * * ? *</TRIGGER>
        <TRIGGER id= "de">0 0 9 * * ? *</TRIGGER>
        [.............]
    </TRIGGER_GROUP>
    <TRIGGER_GROUP id="committees" poolsize="1">
        <TRIGGER id= "bg">0 0 0 * * ? *</TRIGGER>
        <TRIGGER id= "cs">0 0 1 * * ? *</TRIGGER>
        [.............]
        <TRIGGER id= "sk">0 0 21 * * ? *</TRIGGER>
        <TRIGGER id= "sl">0 0 22 * * ? *</TRIGGER>
        <TRIGGER id= "sv">0 0 23 * * ? *</TRIGGER>
    </TRIGGER_GROUP>
    [.............]
</TRIGGERING>
```

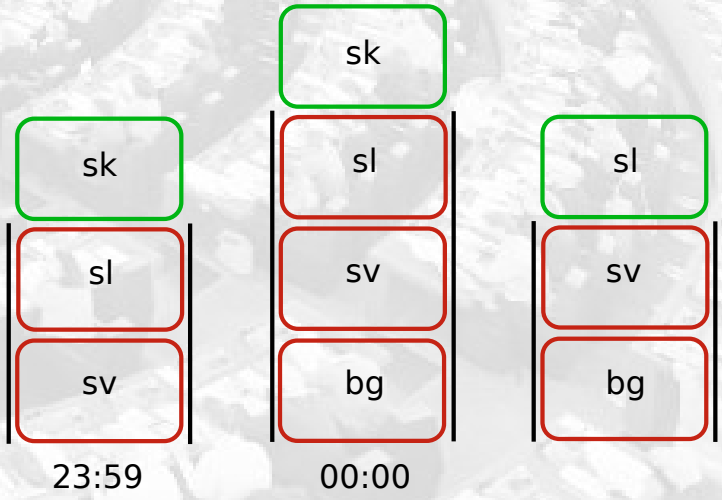
Utilisation d'expression CRON

« Misfire » instruction :

=> Problème avec la librairie Quartz

=> Ré-implémentation en FIFO

File pour "committees" :



Réalisation : Extraction des données

```
<HTML_MODEL id="delegations">
    <DOCUMENTDATE>
        <SELECTOR>meta[name=available]</SELECTOR>
        <ATTRIBUTE>content</ATTRIBUTE>
    </DOCUMENTDATE>
    [.............]
    <CONTENT>
        <ELEMENT>
            <SELECTOR>div#main_menu</SELECTOR>
        </ELEMENT>
        <ELEMENT>
            <SELECTOR>div#global_content</SELECTOR>
        </ELEMENT>
    </CONTENT>
</HTML_MODEL>
```

Localisation des données à extraire grâce aux sélecteurs CSS.

La librairie java « JSOUP » extrait le contenu textuel des balises.

Réalisation : Parcours des pages

```
<PARSER id="MEP_FR" threads="10">
    <START>http://www.europarl.europa.eu/meps/fr.html</START>
    <POLICIES>
        <POLICY>
            <CONTENT_TYPE>
                <HTML model="mep">
            </ CONTENT_TYPE>
            <KEYS>
                <PATTERN toIndex="true" asSource="true">
                    http://www.europarl.europa.eu/meps/legacy/fr/*
                </PATTERN>
                <PATTERN>
                    http://www.europarl.europa.eu/meps/fr/*
                </PATTERN>
            </KEYS>
            <EXCLUDED_LINKS>
                <PATTERN>*/popup*</PATTERN>
                <PATTERN>*/#fadeout*</PATTERN>
            <EXCLUDED_LINKS>
        </POLICY>
    <POLICIES>
<PARSER>
```

=> Utilisation des adresses canoniques

=> Volonté de d'offrir la configuration la plus souple possible

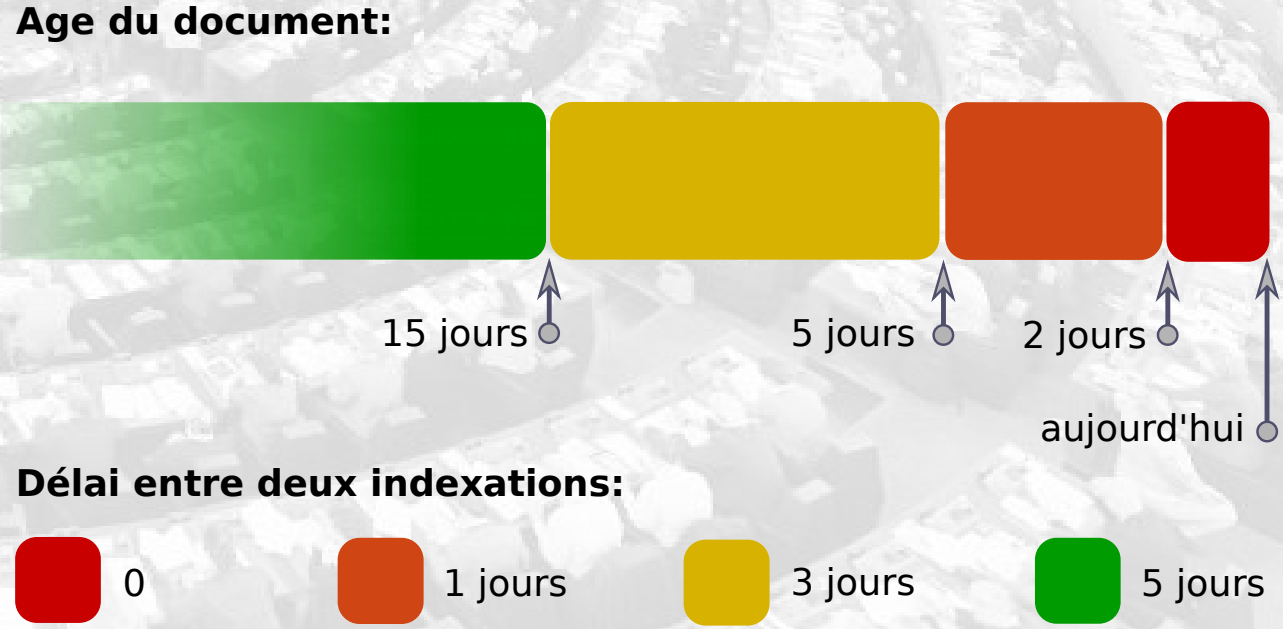
=> Une prise en compte optionnelle du robots.txt dans la prochaine version

Réalisation : Politiques d’indexation

```
<INDEXING_POLICY>
        <AGE unit="DAYS">0</AGE>
        <DELAY unit="DAYS">0</DELAY>
</INDEXING_POLICY>
<INDEXING_POLICY>
        <AGE unit="DAYS">2</AGE>
        <DELAY unit="DAYS">1</DELAY>
</INDEXING_POLICY>
<INDEXING_POLICY>
        <AGE unit="DAYS">5</AGE>
        <DELAY unit="DAYS">3</DELAY>
</INDEXING_POLICY>
<INDEXING_POLICY>
        <AGE unit="WEEKS">2</AGE>
        <DELAY unit="DAYS">5</DELAY>
</INDEXING_POLICY>
```

- Réduire la charge des travaux d’indexation

- Nettoyer les index des pages périmées



Réalisation : fonctionnalités de recherche (1)

```
"text": {
    "type": "text",
    "analyzer": "french",
    "search_analyzer": "french_search",
    "search_quote_analyzer": "french",
    "fields": {
        "withstopwords": {
            "type": "text",
            "analyzer": "french",
            "search_analyzer": "french"
        },
        "bigram": {
            "type": "text",
            "analyzer": "bigram"
            "search_analyzer": "bigram"
        }
    }
}
```

Implémentation de la « pertinence » d'un résultat :

=> Alimentation de 3 champs avec les même données

=> 3 chaînes d'analyse différentes sur les données indexées.

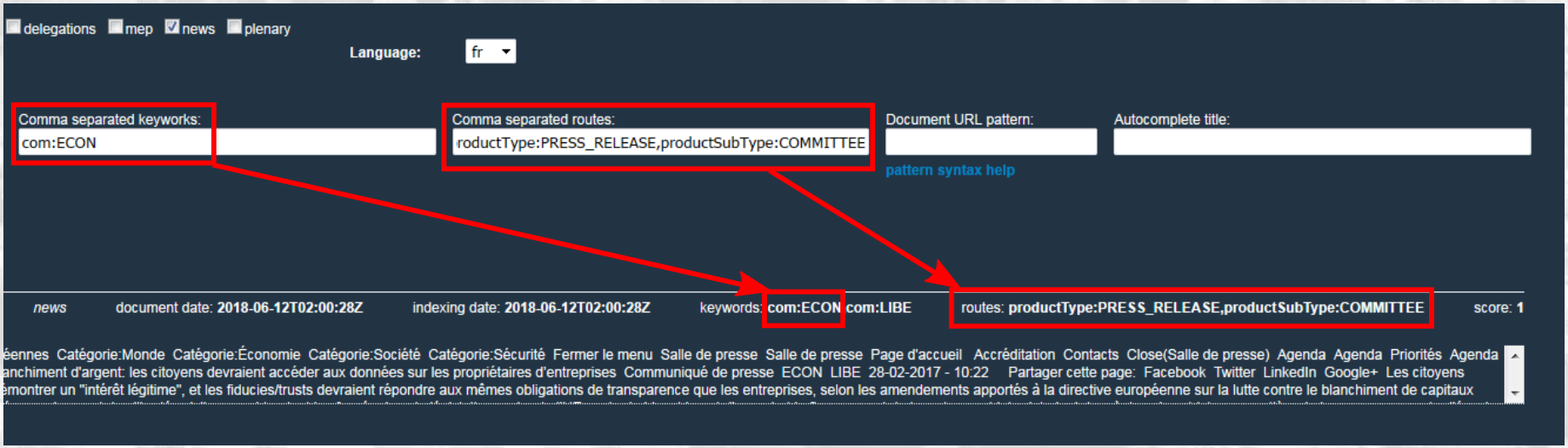
=> priorisation de certaines chaînes d'analyse dans la requête (**boost**)

=> proximité => paramètres **slop**

Réalisation : fonctionnalités de recherche (2)

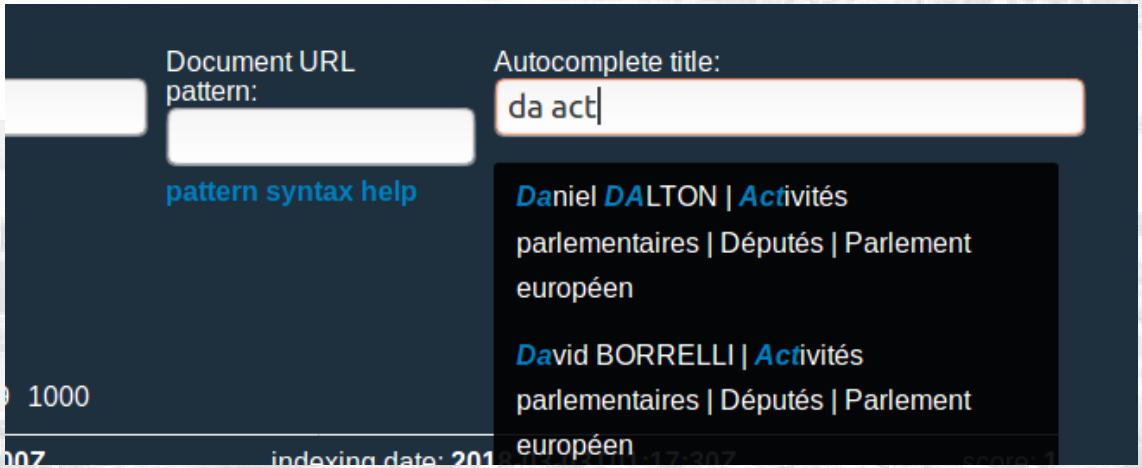
Suggestion de phrase (fonctionnalité ES : slop, fuzziness)

« keywords » et « routes » (0 traitement => recherche par thèmes « OR » et/ou par catégorie « ET »)

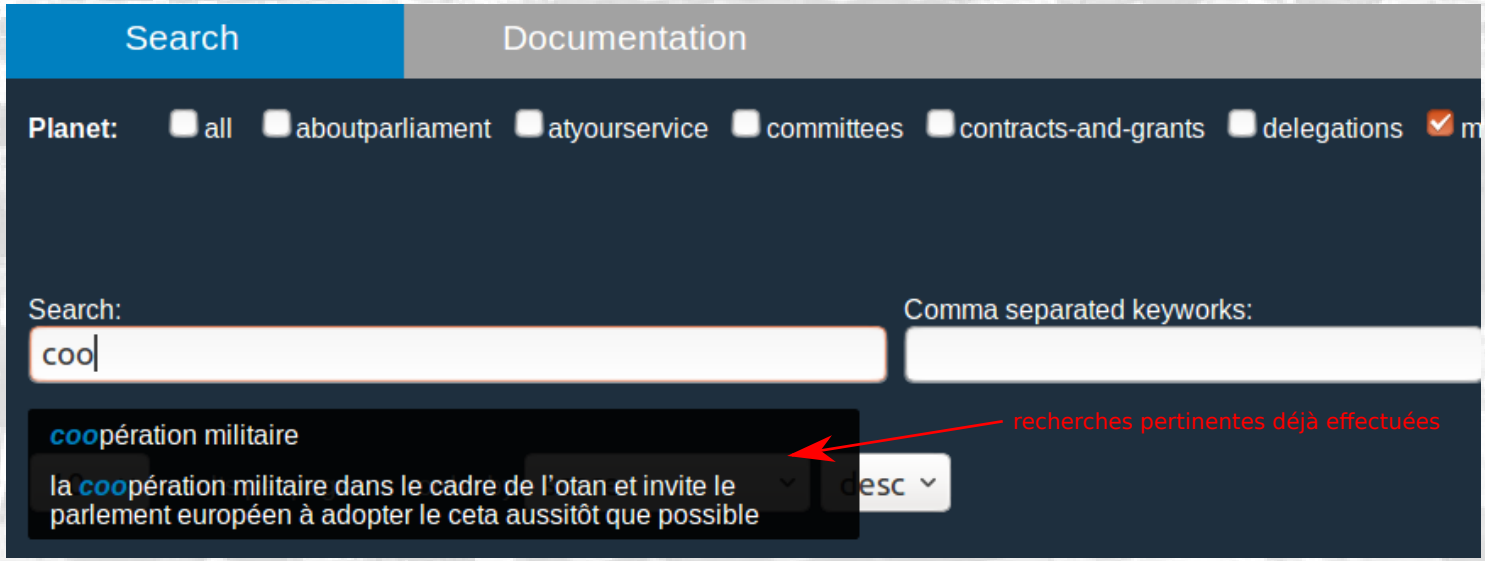


Réalisation : fonctionnalités de recherche (3)

Auto-complétion sur les titres (reconfiguration => découpe par lettre => champ de taille limitée)



Auto-complétion sur les recherches pertinentes (prévention contre l'injection)

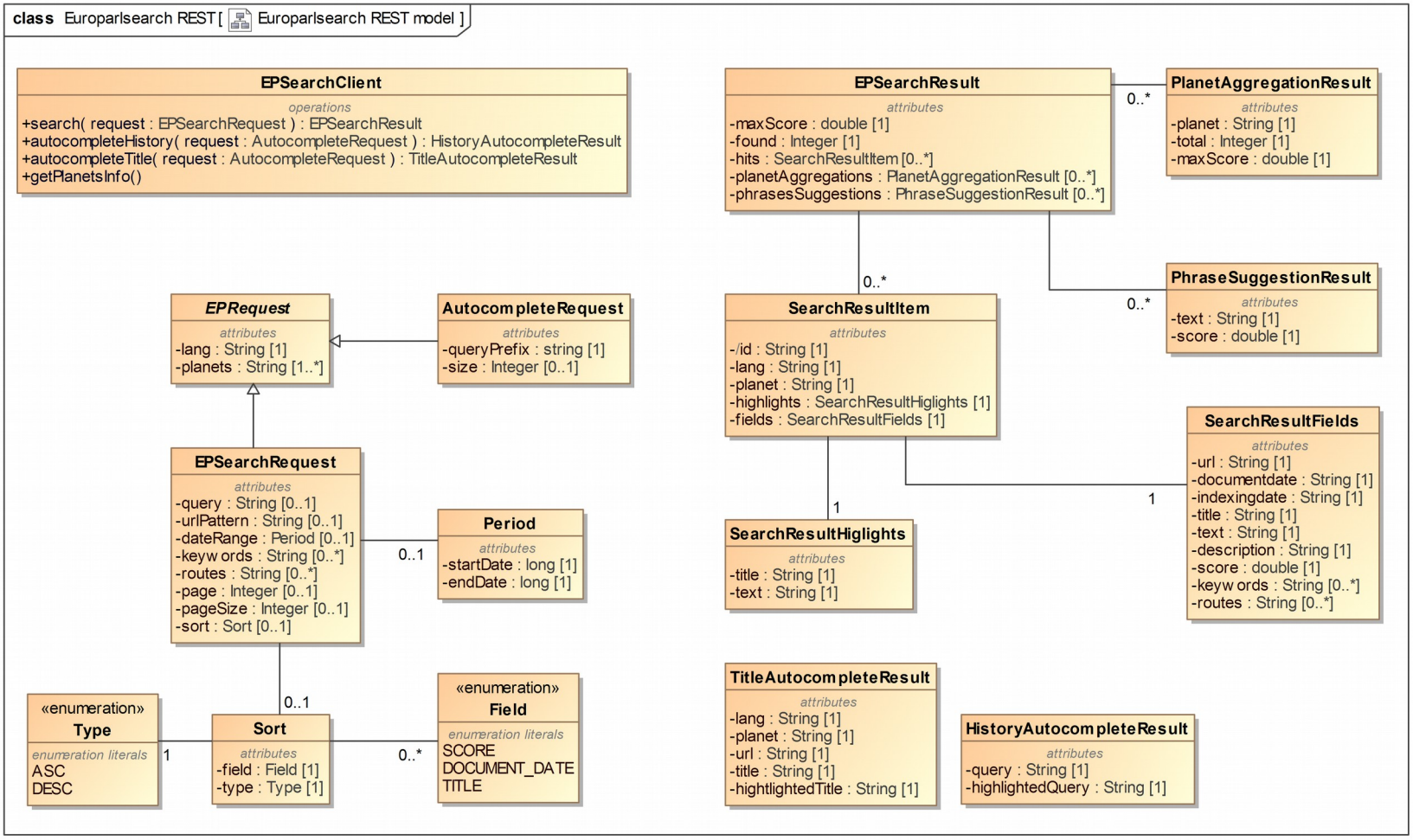


Réalisation : Les services web de recherche

=> Centraliser les accès à ES

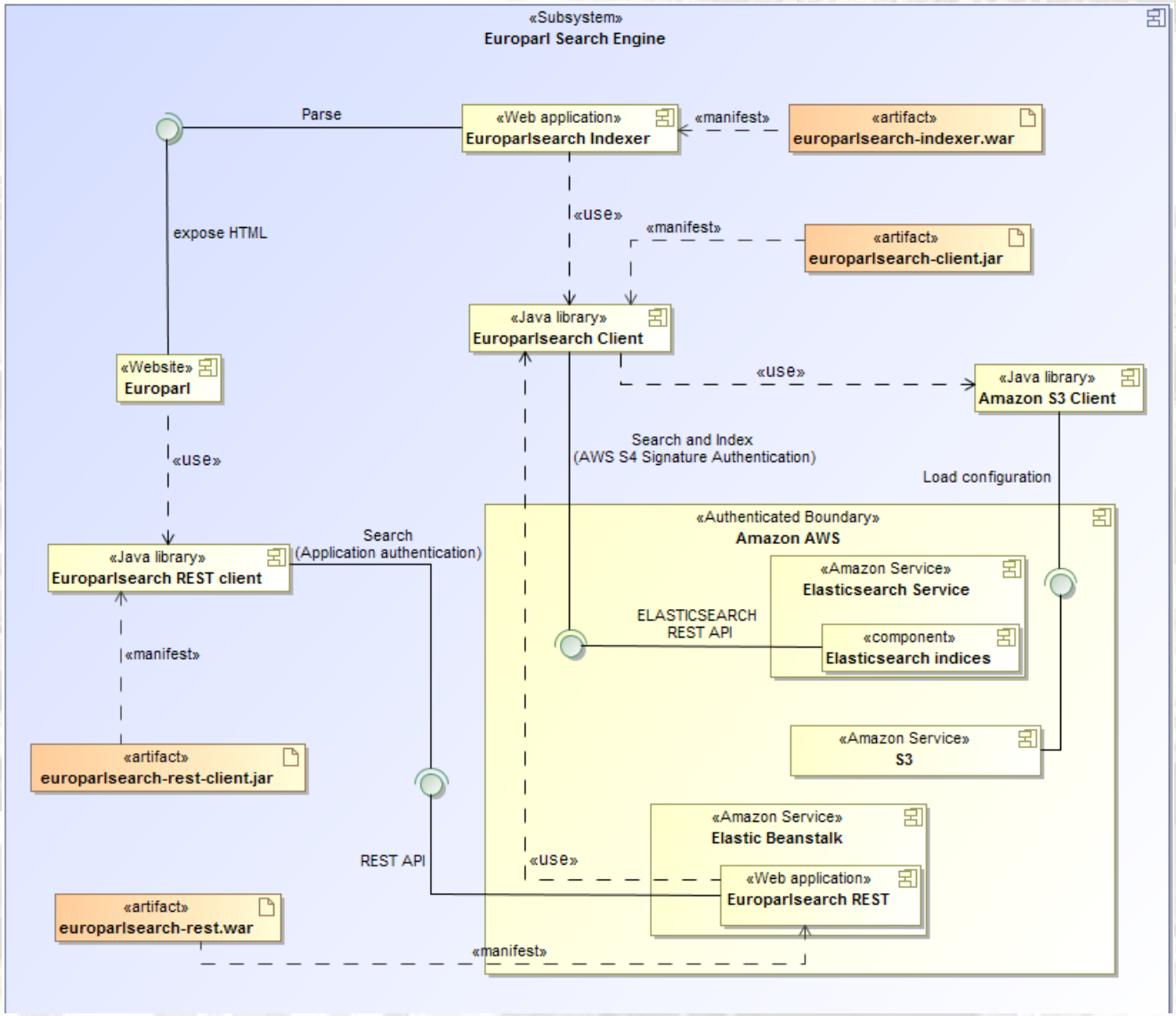
=> Simplifier la recherche aux éléments de l'analyse fonctionnelle

=> Pas de « wsdl » : (interface de programmation + documentation publique détaillée)



Réalisation : Architecture définitive

4 modules différents :



Tests, intégration continue et documentation

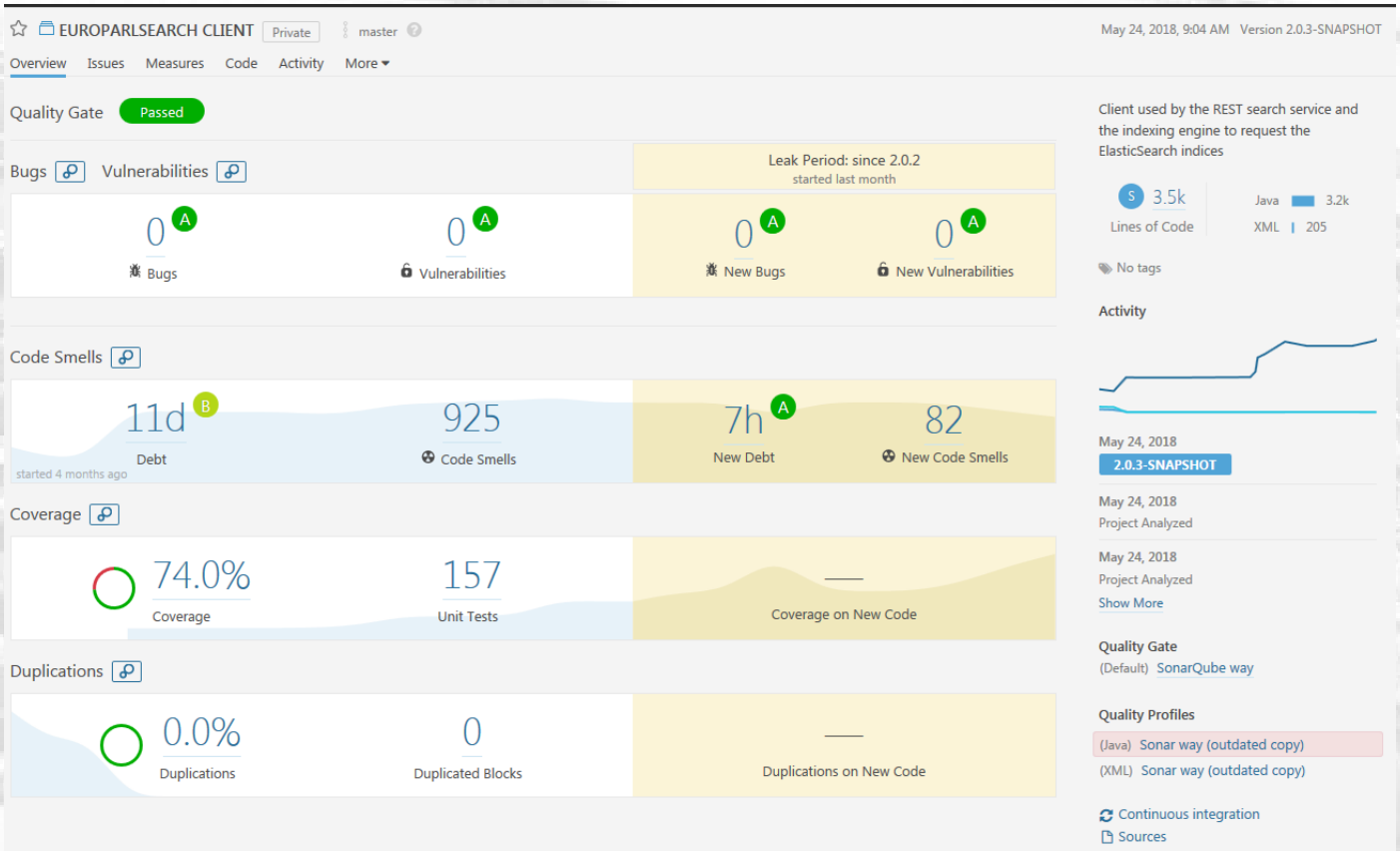
Tests :

Tests unitaires, tests d'intégration, tests de validation

Intégration continue :

Qualité du code

et couverture de tests



Documentation :

Une documentation sur l'ensemble de la chaîne, de la conception à la production

Retour d’expérience

Outil méconnu :

=> Analyse technique longue

=> « Qualité fonctionnelle» == Incertitude

=> Développement itératif : fonctionnalité / fonctionnalité

Cloud (un bilan mitigé) :

=> Expérience très satisfaisante en tant que développeur

=> Redéfinition des procédures

=> Nécessite une forte participation des services d’infrastructure

Merci