

Nicolas Rouffart's landscape

Generated by [IcePanel](#) at 2025-08-04 08:53:25

Table of contents

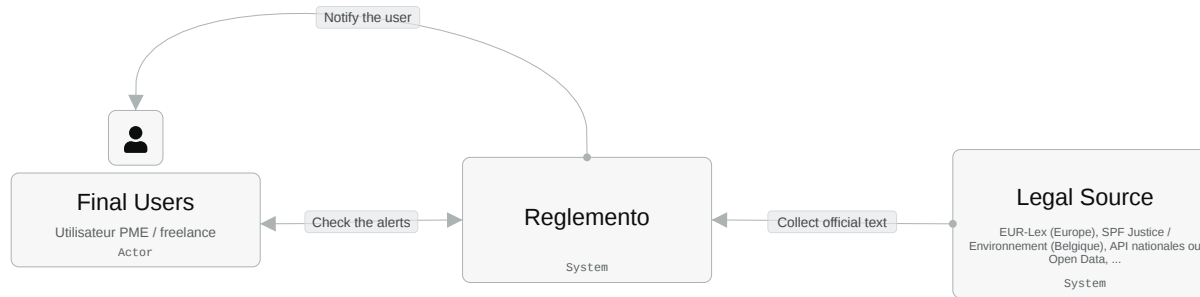
- [1. Context diagrams](#)
 - [1.1. Context Diagram](#)
- [2. App diagrams](#)
 - [2.1. Reglemento App Diagram](#)
- [3. Component diagrams](#)
 - [3.1. Back-end API Component Diagram](#)
 - [3.2. Front-end Component Diagram](#)
- [4. Actors](#)
 - [4.1. Final Users](#)
 - [4.2. OAuth0](#)
- [5. Systems](#)
 - [5.1. API Layer](#)
 - [5.2. Legal Source](#)
 - [5.3. Reglemento](#)

- [6. Apps](#)
 - [6.1. Back-end API](#)
 - [6.2. Front-end](#)
 - [6.3. Scraping](#)
- [7. Stores](#)
 - [7.1. reglemento-origonal](#)
 - [7.2. Reglemento-Storage](#)
- [8. Components](#)
 - [8.1. AlertService](#)
 - [8.2. APILayer](#)
 - [8.3. APIServices](#)
 - [8.4. AppComponent \(root\)](#)
 - [8.5. AuthModules](#)
 - [8.6. NotificationModules](#)
 - [8.7. RegulationController](#)
 - [8.8. RegulationModule](#)
 - [8.9. RegulationRepository](#)
 - [8.10. SharedModules](#)
 - [8.11. UserProfileModules](#)

1. Context diagrams

1.1. Context Diagram

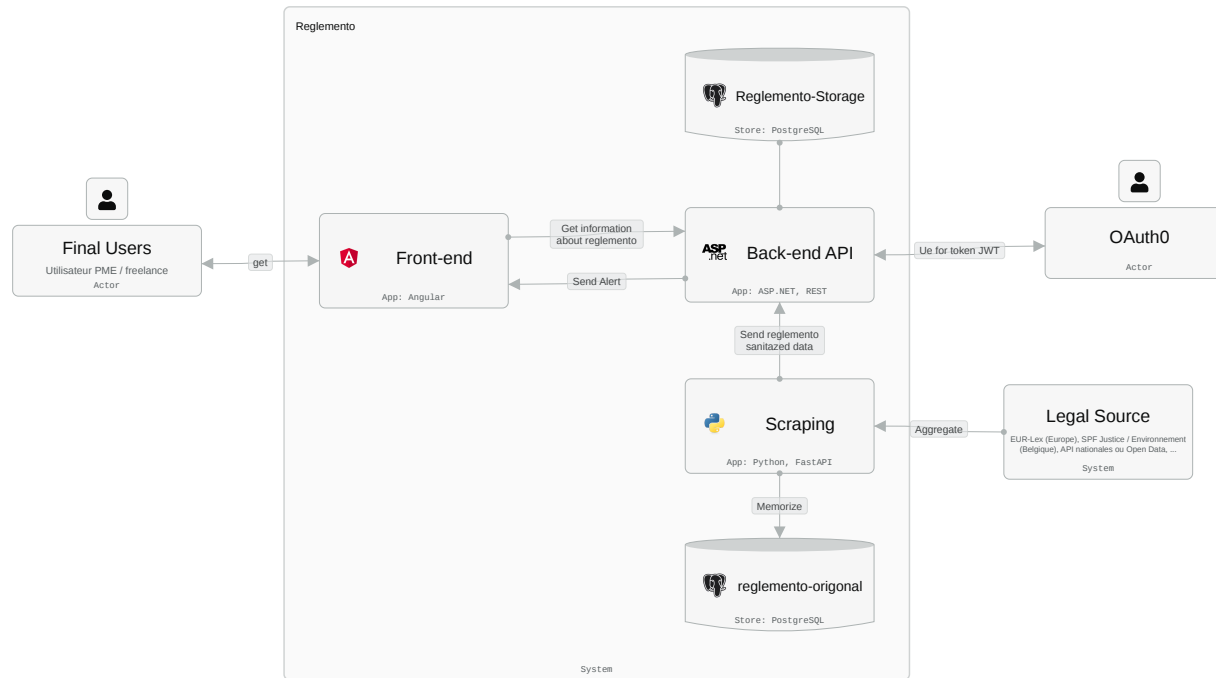
Context diagram in the Reglemento domain.



2. App diagrams

2.1. Reglemento App Diagram

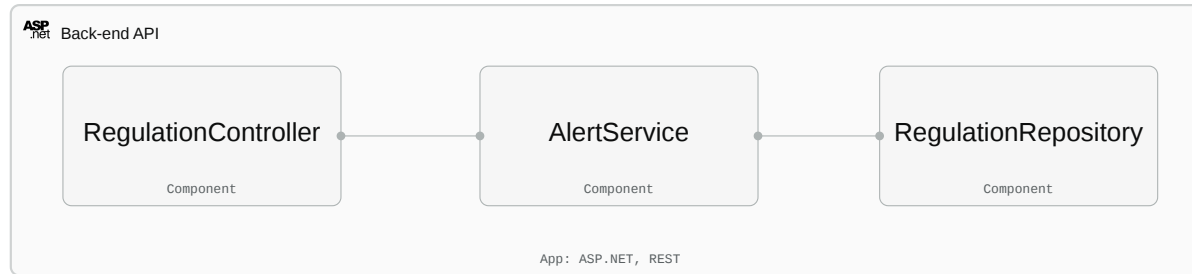
Reglemento app diagram in the Reglemento domain.



3. Component diagrams

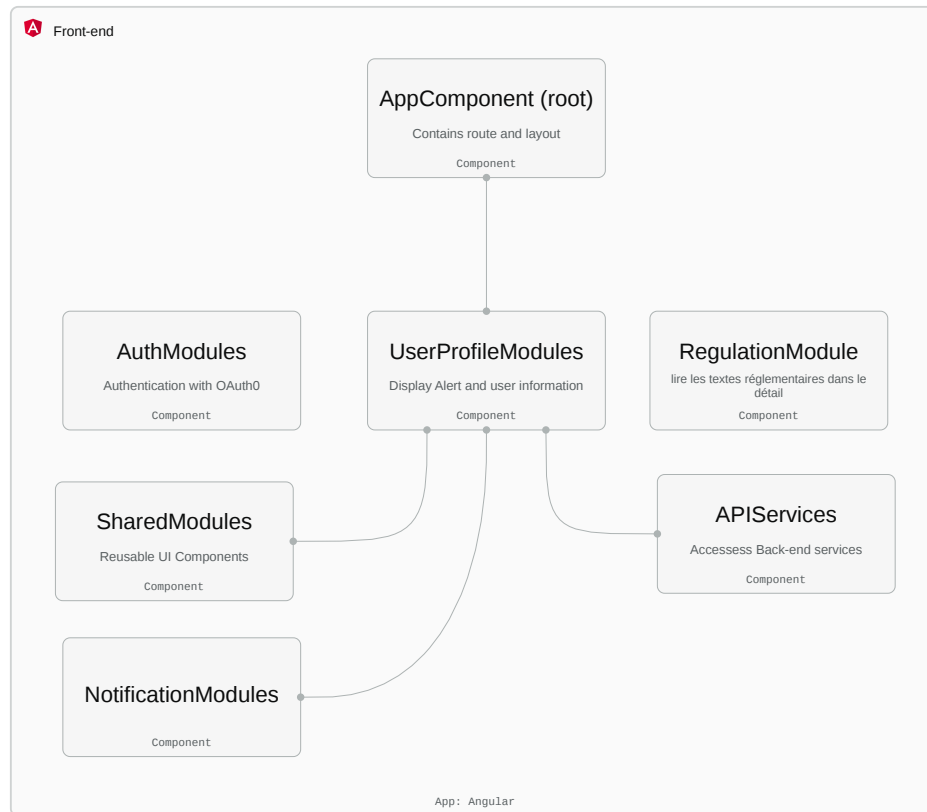
3.1. Back-end API Component Diagram

Back-end API component diagram in the Reglemento domain.



3.2. Front-end Component Diagram

Front-end component diagram in the Reglemento domain.



4. Actors

4.1. Final Users

Internal actor in the Reglemento domain.

Status: live

Utilisateur PME / freelance

In diagrams:

- [Context Diagram](#)
- [Reglemento App Diagram](#)

Description :

Personne physique ou entreprise qui souhaite recevoir des alertes juridiques ciblées selon son secteur et ses activités.

Responsibilities :

- Crée un compte sur Reglemento
- Définit son profil réglementaire (secteur, pays, obligations)
- Consulte les alertes

Technical Decisions : *(vide)*

4.2. OAuth0

Internal actor in the Reglemento domain.

Status: live

In diagrams:

- [Reglemento App Diagram](#)

Description : Fournisseur SaaS utilisé pour gérer l'authentification et l'autorisation des utilisateurs de manière sécurisée.

Responsibilities :

- Connexion, inscription, gestion des tokens
- Federation d'identité (optionnel)

Technical Decisions :

- Utilisation du SDK SPA (frontend)
- Intégration des JWT validés dans .NET backend

5. Systems

5.1. API Layer

Internal system in the Reglemento domain.

Status: live

5.2. Legal Source

Internal system in the Reglemento domain.

Status: live

EUR-Lex (Europe), SPF Justice / Environnement (Belgique), API nationales ou Open Data, ...

In diagrams:

- [Context Diagram](#)
- [Reglemento App Diagram](#)

EUR-Lex

- **Description :** Base de données officielle du droit de l'Union Européenne.

- **Responsibilities** : Fournit les textes et règlements européens.
- **Technical Decisions** :
 - Utilisation de leur moteur de recherche (scraping) ou RSS

☐☐ **SPF Justice / Environnement**

- **Description** : Portails officiels belges publiant les lois, arrêtés royaux, etc.
- **Responsibilities** : Source des réglementations nationales belges.
- **Technical Decisions** :
 - Données extraites via scraping ou formats XML/HTML

☐☐ **LegiFrance API**

- **Description** : Portail officiel du droit français.
- **Responsibilities** : Fournit lois, décrets, codes, etc.
- **Technical Decisions** :
 - API REST officielle disponible
 - Gestion des jetons et pagination

5.3. Reglemento

Internal system in the Reglemento domain.

Status: live

In diagrams:

- [Context Diagram](#)
- [Reglemento App Diagram](#)

Description :

Plateforme centralisée qui collecte, filtre, et notifie des évolutions réglementaires pertinentes pour les utilisateurs.

Responsibilities :

- Centraliser des sources officielles européennes et nationales
- Filtrer les données en fonction des profils
- Notifier les utilisateurs

Technical Decisions :

- Application web Angular / .NET
- Données issues de scrapers Python ou APIs
- Auth via Auth0

6. Apps

6.1. Back-end API

Internal app in the Reglemento domain.

Status: live

Technologies:

- [ASP.NET](#)
- [REST \(Representational State Transfer\)](#)

In diagrams:

- [Back-end API Component Diagram](#)
- [Reglemento App Diagram](#)

Description :

Point d'entrée pour toutes les communications entre le frontend, les données métier, et les services de traitement comme le scraping ou les notifications.

Responsibilities :

- Exposer des endpoints REST sécurisés
- Orchestration des appels vers le module de scraping
- Gestion des profils, règlements, sources, et alertes
- Déclenchement des notifications

Technical Decisions :

- Architecture modulaire, DDD
- Authentification par JWT via Auth0
- Swagger / OpenAPI
- Middleware de logging, validation, sécurité

6.2. Front-end

Internal app in the Reglemento domain.

Status: live

Technologies:

- [Angular](#)

In diagrams:

- [Front-end Component Diagram](#)
- [Reglemento App Diagram](#)

Description :

Application monopage permettant aux utilisateurs d'interagir avec la plateforme Reglemento via une interface moderne.

Responsibilities :

- Authentification via Auth0
- Affichage des alertes, règlements, et contenu personnalisé
- Gestion du profil utilisateur (secteur, pays, filtres)

Technical Decisions :

- Angular 17+
- TypeScript / RxJS
- Consommation de l'API REST exposée par le backend .NET
- Intégration de la SDK Auth0 pour l'authentification (SPA)

6.3. Scraping

Internal app in the Reglemento domain.

Status: live

Technologies:

- [Python](#)
- [FastAPI](#)

In diagrams:

- [Reglemento App Diagram](#)

7. Stores

7.1. reglemento-origonal

Internal store in the Reglemento domain.

Status: live

Technologies:

- [PostgreSQL](#)

In diagrams:

- [Reglemento App Diagram](#)

Description :

Microservice chargé d'extraire les données réglementaires à partir de sites gouvernementaux (Belgique, France, UE).

Responsibilities :

- Scraper EUR-Lex, Legifrance, SPF Justice, etc.
- Nettoyer, normaliser et envoyer les données au backend via une API

Technical Decisions :

- FastAPI
- Scrapy / BeautifulSoup
- API REST exposée au backend .NET
- Déploiement possible via Docker

7.2. Reglemento-Storage

Internal store in the Reglemento domain.

Status: live

Technologies:

- [PostgreSQL](#)

In diagrams:

- [Reglemento App Diagram](#)

Description :

Stocke toutes les données structurées : utilisateurs, profils, sources, alertes, règlements, logs.

Responsibilities :

- Persistance des données applicatives

- Support des recherches par filtre (ex. secteur, pays)

Technical Decisions :

- PostgreSQL 15+
- Entity Framework Core (code-first)
- Tables partitionnées (option futur)
-

8. Components

8.1. AlertService

Internal component in the Reglemento domain.

Status: live

In diagrams:

- [Back-end API Component Diagram](#)

8.2. APILayer

Internal component in the Reglemento domain.

Status: live

8.3. APIServices

Internal component in the Reglemento domain.

Status: live

Accessess Back-end services

In diagrams:

- [Front-end Component Diagram](#)

Description :

Service unique d'accès à l'API backend.

Responsibilities :

- Abstraction des appels HTTP
- Gestion des erreurs et retry logic

Technical decisions :

- HttpClient + interceptors
- Centralisation des endpoints

8.4. AppComponent (root)

Internal component in the Reglemento domain.

Status: live

Contains route and layout

In diagrams:

- [Front-end Component Diagram](#)

Description : Composant racine qui encapsule toute l'application Angular.

Responsibilities :

- Contient la structure de layout (navbar, footer)
- Point d'entrée pour le routing

Technical decisions :

- Utilise Angular Router
-

- Chargement du `UserService` dès l'initialisation

8.5. AuthModules

Internal component in the Reglemento domain.

Status: live

Authentication with OAuth0

In diagrams:

- [Front-end Component Diagram](#)

Description : Ensemble de composants et services dédiés à l'authentification avec Auth0.

Responsibilities :

- Connexion, déconnexion, gestion du token
- Redirection post-login

Technical decisions :

- Utilisation du SDK Auth0 Angular
- Stockage du token dans `localStorage` via interceptor

8.6. NotificationModules

Internal component in the Reglemento domain.

Status: live

In diagrams:

- [Front-end Component Diagram](#)

Description :

Composant affichant les dernières notifications (non lues ou urgentes).

Responsibilities :

- Popover ou toast des notifications
- Liens vers les règlements concernés

Technical decisions :

- Utilisation d'un `MatSnackBar` ou `ToastService` custom
-

8.7. RegulationController

Internal component in the Reglemento domain.

Status: live

In diagrams:

- [Back-end API Component Diagram](#)

8.8. RegulationModule

Internal component in the Reglemento domain.

Status: live

lire les textes réglementaires dans le détail

In diagrams:

- [Front-end Component Diagram](#)

Description : Composants permettant de lire les textes réglementaires dans le détail.

Responsibilities :

- Affichage HTML/Markdown enrichi
- Historique / versioning des textes (future feature)

Technical decisions :

- Intégration possible d'un lecteur Markdown / HTML Sanitize

8.9. RegulationRepository

Internal component in the Reglemento domain.

Status: live

In diagrams:

- [Back-end API Component Diagram](#)

8.10. SharedModules

Internal component in the Reglemento domain.

Status: live

Reusable UI Components

In diagrams:

- [Front-end Component Diagram](#)

Description :

Contient les composants UI réutilisables.

Responsibilities :

- Boutons, menus, alert-box, etc.
- Pipes et directives personnalisées

Technical decisions :

- Angular Material ou Tailwind UI (selon choix du projet)

8.11. UserProfileModules

Internal component in the Reglemento domain.

Status: live

Display Alert and user information

In diagrams:

- [Front-end Component Diagram](#)

Description :

Composants chargés d’afficher les alertes réglementaires pertinentes à l’utilisateur.

Responsibilities :

- Affichage en liste ou carte
- Filtres dynamiques
- Marquage comme lu

Technical decisions :

- Pagination + Lazy loading
- Pipe personnalisé pour le formatage du texte