**Permettre aux développeurs de produire du code sécurisé**

Toute entreprise soucieuse de la sécurité de ses applications doit fournir à ses développeurs les éléments dont ils ont besoin pour produire du code sécurisé. Beaucoup d’entreprises attendent de leurs développeurs des applications avec un certain degré de sécurité sans faire grand-chose pour permettre à ces développeurs d’en construire.

La première chose à faire est de mettre à la disposition des développeurs un ensemble de contrôles de sécurité disponibles dans leur environnement. Chaque organisation par défaut dispose d’un certain nombre de contrôles de sécurité dans son infrastructure, tels que des bibliothèques de chiffrement, des serveurs de logs, des serveurs d'authentification, etc. Les développeurs ont besoin d'un accès facile à ces contrôles et cela permet à la longue d’avoir une manière standard de prendre en compte la sécurité durant le cycle de développement logiciel.

Une fois ces contrôles de sécurité disponibles, il faut élaborer un ensemble de directives de codage sécurisé pour l’organisation. C'est un ensemble de règles que les développeurs doivent suivre lors du développement d'applications. Ces directives doivent être spécifiques à l’entreprise et contenir de nombreux extraits de code et des exemples de codage sécurisé. En outre, la directive doit être adaptée à l’environnement, aux règles et aux technologies utilisés. En effet, les stratégies théoriques (polices de sécurité, recommandation…) ne sont souvent pas très parlant aux développeurs. Si vous voulez suivre une pratique de programmation particulière, il vaut mieux l'afficher en code.

La dernière chose à faire pour aider les développeurs est de leur donner un peu de formation en codage sécurisé. Le cours devrait couvrir quand et comment utiliser tous les principaux contrôles de sécurité, en donnant des exemples des failles de sécurité courantes associées à chaque contrôle et comment suivre les directives de codage sécurisé afin d'utiliser les contrôles pour éviter ces vulnérabilités.

Le Top 10 Owasp, tout en présentant les failles de sécurité les plus communes, indique, pour chacune de ces failles, un ensemble de contrôles de sécurité pour s’en prévenir. Ainsi, durant cette phase, nous nous sommes penchés sur le Top 10 en explorant chacun de ces points afin de prélever les contrôles de sécurité adéquats. Chaque point fait l’objet d’un paquetage de contrôles de sécurité.

1. **Package Injection**

Dans ce package, il s’agit de relever toutes les fonctions de sécurité permettant de se protéger des injections.

* 1. Analyse du risque « Injection »
     1. Définition

Une attaque par injection cible les applications utilisant des interpréteurs de commandes. Lors d’une attaque par injection, l’attaquant fournit un/des segments malicieux de code en tant qu’input transformant la requête initiale en une autre souvent dangereuse. Les attaques par injection sont le plus souvent utilisées pour dérober des informations sensibles. Par une injection, l’attaquant, selon ses capacités peut obtenir des informations confidentielles ou même arriver à modifier ces informations.

* + 1. Causes

La cause principale d’une injection est le fait qu’une application accepte des données provenant de sources non sures comme l’utilisateur par exemple (input) ou le navigateur de l’utilisateur (cookies ou head ers des requêtes http)

* + 1. Vecteurs principaux
* HTTP POST
* HTTP GET
* Cookies
* HTTP Headers
* Champs de formulaire
  + 1. Préventions

Principales :

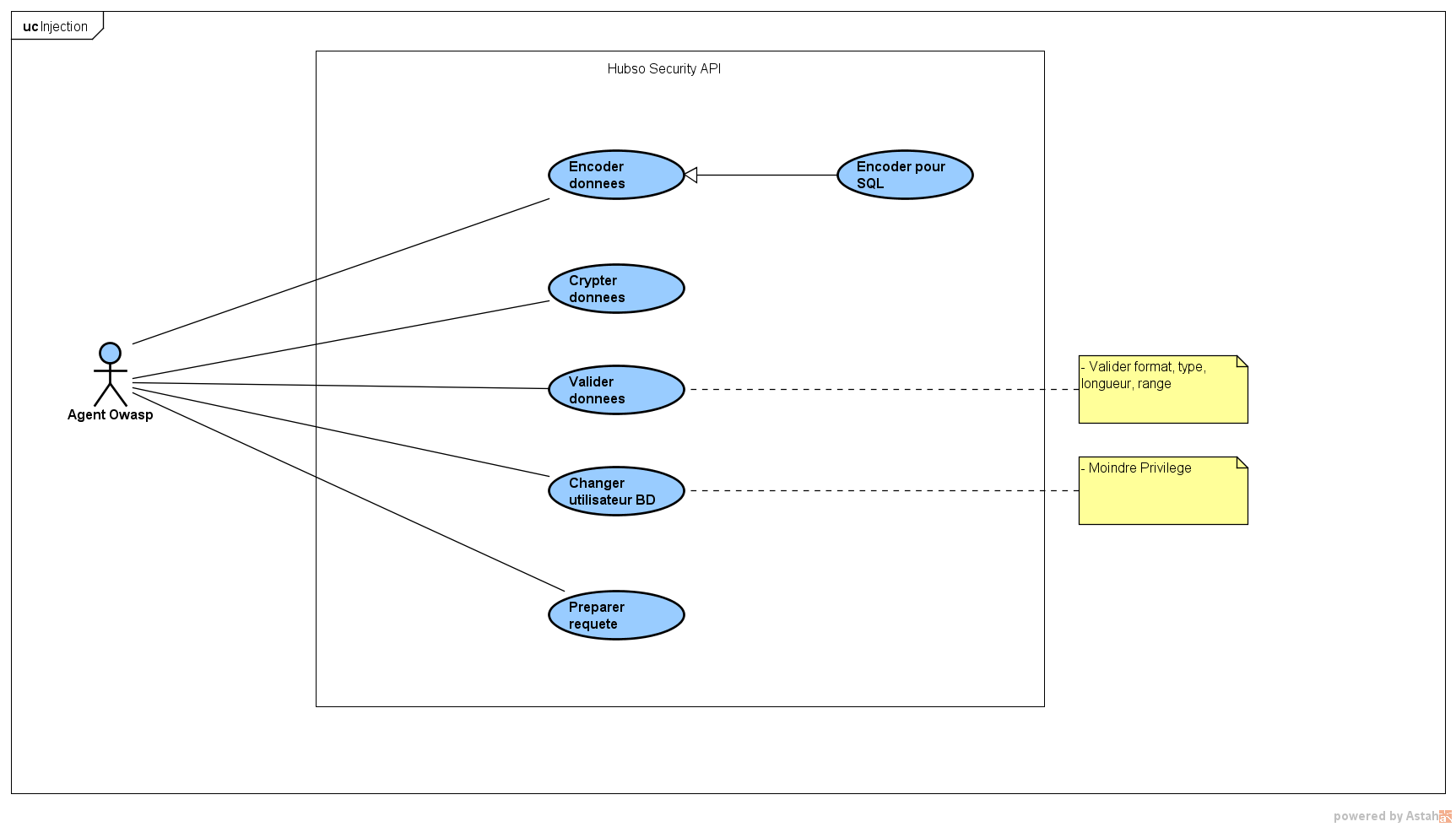
* Input Validation
* Encodage
* Requêtes préparées

Supplémentaires :

* Réduire la surface d’attaques en donnant le moins de privilèges
* Crypter toutes les données sensibles
* Se méfier des headers http

Ainsi, pour prévenir les injections, les fonctions de sécurités suivantes doivent être mises à la disposition des développeurs :

* la validation des entrées
* l’encodage des données
* la préparation de requête
* le changement d’utilisateur de base de données
* le cryptage de données
  1. Diagramme de cas d’utilisation



* 1. Fiches de description textuelle des cas d’utilisation
     1. Cas d’utilisation « Valider données »

Sommaire d’identification

**Titre** : supprimer utilisateur

**Résumé** : Ce cas d’utilisation permet au responsable pédagogique de supprimer un utilisateur qui ne fait plus parti de l’UMMISCO du système**.**

**Acteur(s)** : Responsable pédagogique

**Date de création** : 26/05/2016 **Date de mise à jour :** 07/07/2016

**Version :** 1.1 **Responsable :** groupe 4

Description des scénarios

**Précondition(s)**

* Le responsable pédagogique doit être authentifié.

**Scénario nominal**

|  |  |
| --- | --- |
| 1. Le responsable pédagogique demande la  catégorie de l'utilisateur qu'il veut supprimer. | 2. Le système affiche la liste des utilisateurs de cette catégorie. |
| 3. Le responsable pédagogique demande la suppression d'un utilisateur. | 4. Le système affiche un message de confirmation de suppression. |
| 5. Le responsable pédagogique confirme la suppression. | 1. Le système supprime l’utilisateur. 2. Le système informe le responsable pédagogique que la suppression s’est bien effectuée. |

**Enchainement(s) alternatif**(s)

*A1 : Annulation de la demande de suppression*

L’enchainement A1 démarre au point 4 du scénario nominal.

5. Le système affiche la liste des utilisateurs de cette catégorie.

Le scénario nominal reprend au point 3.

**Enchainement(s) d’erreur**

*E2 : code d*’*identification définitivement erroné*

L’enchaînement E2 démarre au point 5 du scénario nominal.

6. Le GAB indique au Porteur de carte que le code est erroné, pour la troisième fois.

7. Le GAB confisque la carte.

8. Le Système d’autorisation est informé ; le cas d’utilisation se termine en échec.

**Post condition :**

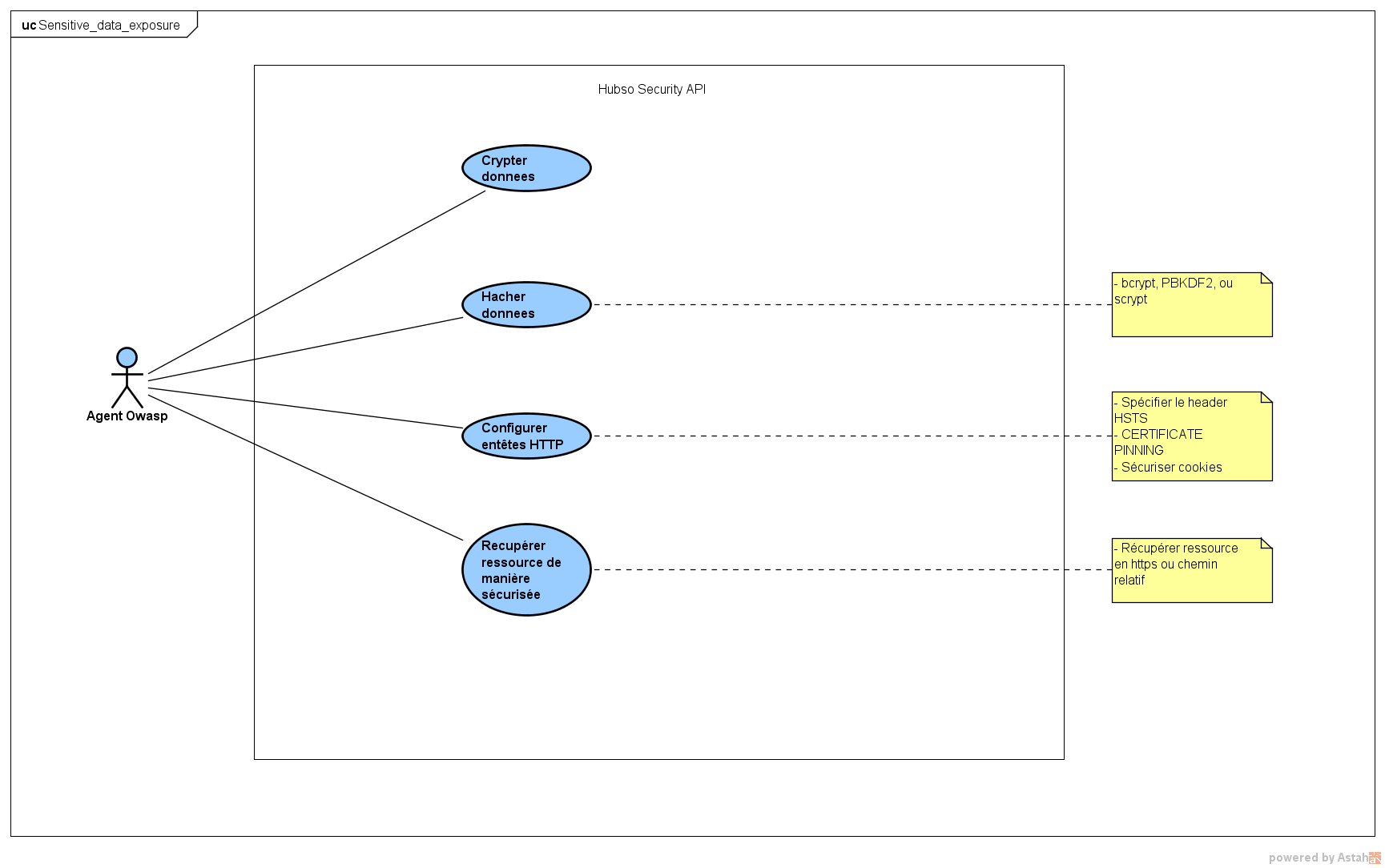
L’utilisateur en question a été supprimé du système.

Exigences non fonctionnelles

|  |  |
| --- | --- |
| Contraintes | Descriptif |
| Temps de réponse | L’interface du GAB doit réagir en l’espace de 2 secondes au  maximum. Une transaction nominale de retrait doit durer moins  de 2 minutes. |

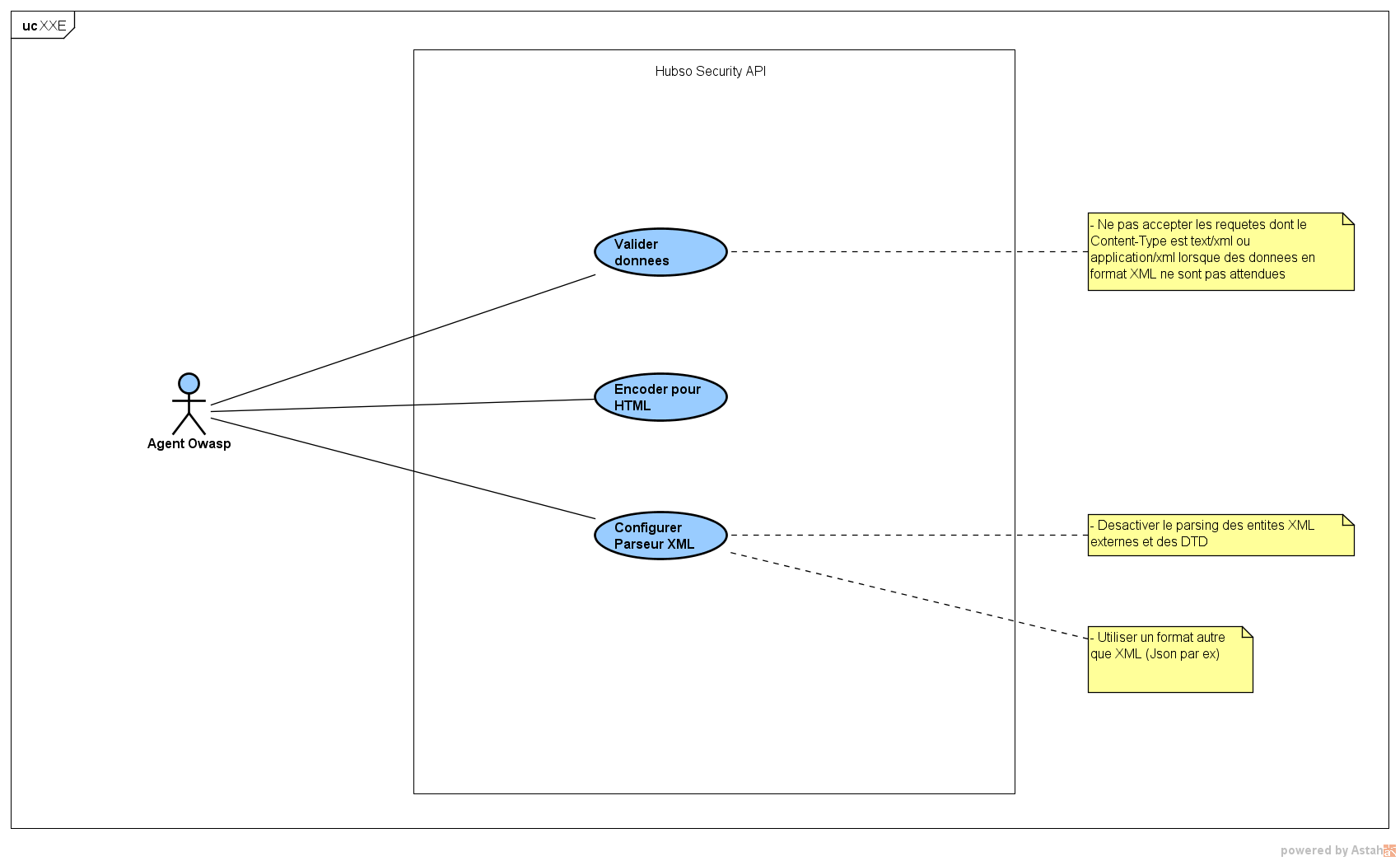
* 1. Diagrammes d’activités des cas d’utilisation

1. **Package Violation de Gestion d’Authentification**
   1. Analyse du risque « Violation de Gestion d’Authentification »
   2. Diagramme de cas d’utilisation
   3. Fiches de description textuelle des cas d’utilisation
   4. Diagramme d’activités des cas d’utilisation
2. **Package Exposition de données sensibles**
   1. Analyse du risque « Exposition de données sensibles »
   2. Diagramme de cas d’utilisation



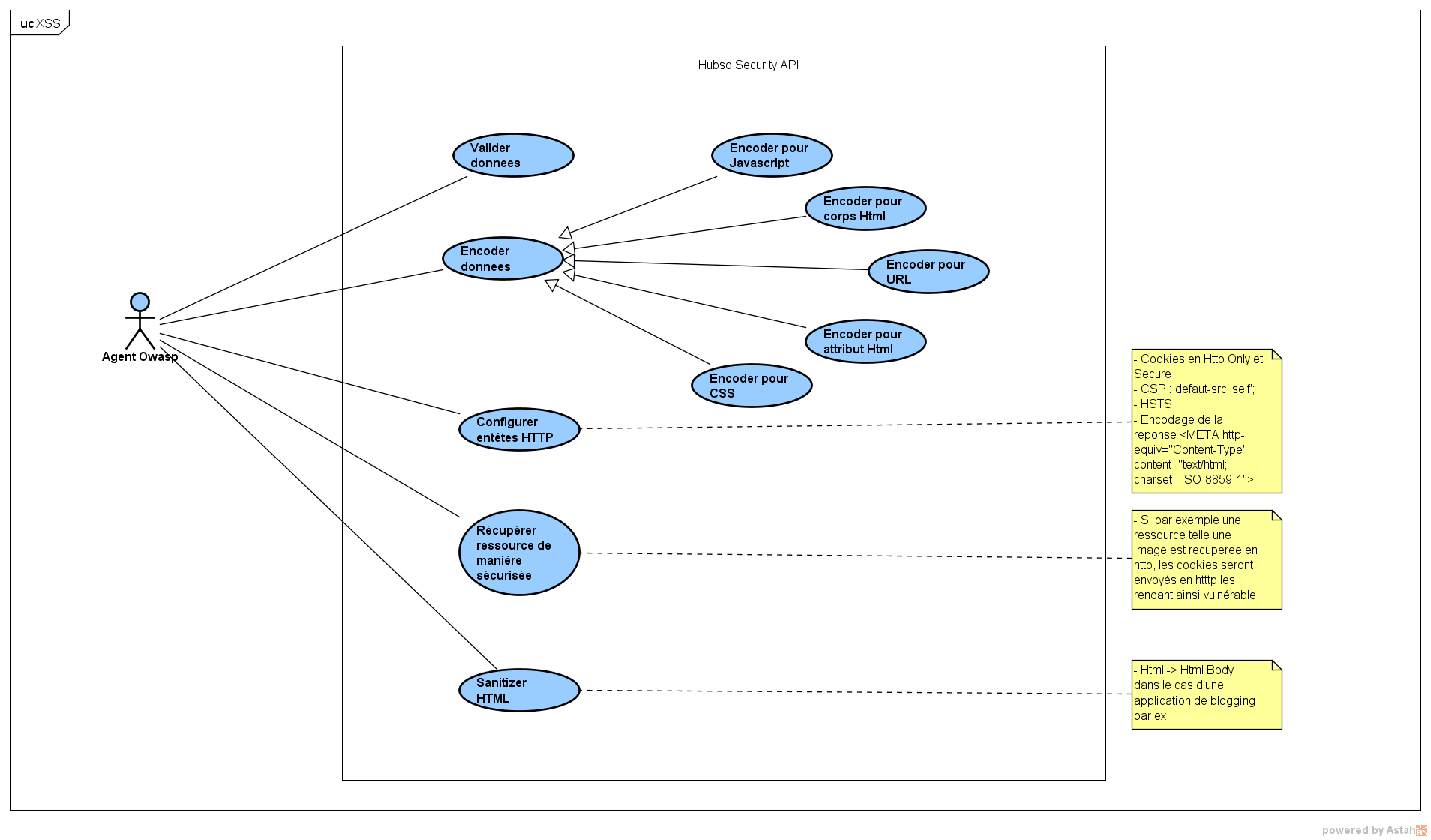
* 1. Fiches de description textuelle des cas d’utilisation
  2. Diagramme d’activités des cas d’utilisation

1. **Package Attaque sur les entités XML externes (XXE)**
   1. Analyse du risque « Attaque XXE »
   2. Diagramme de cas d’utilisation



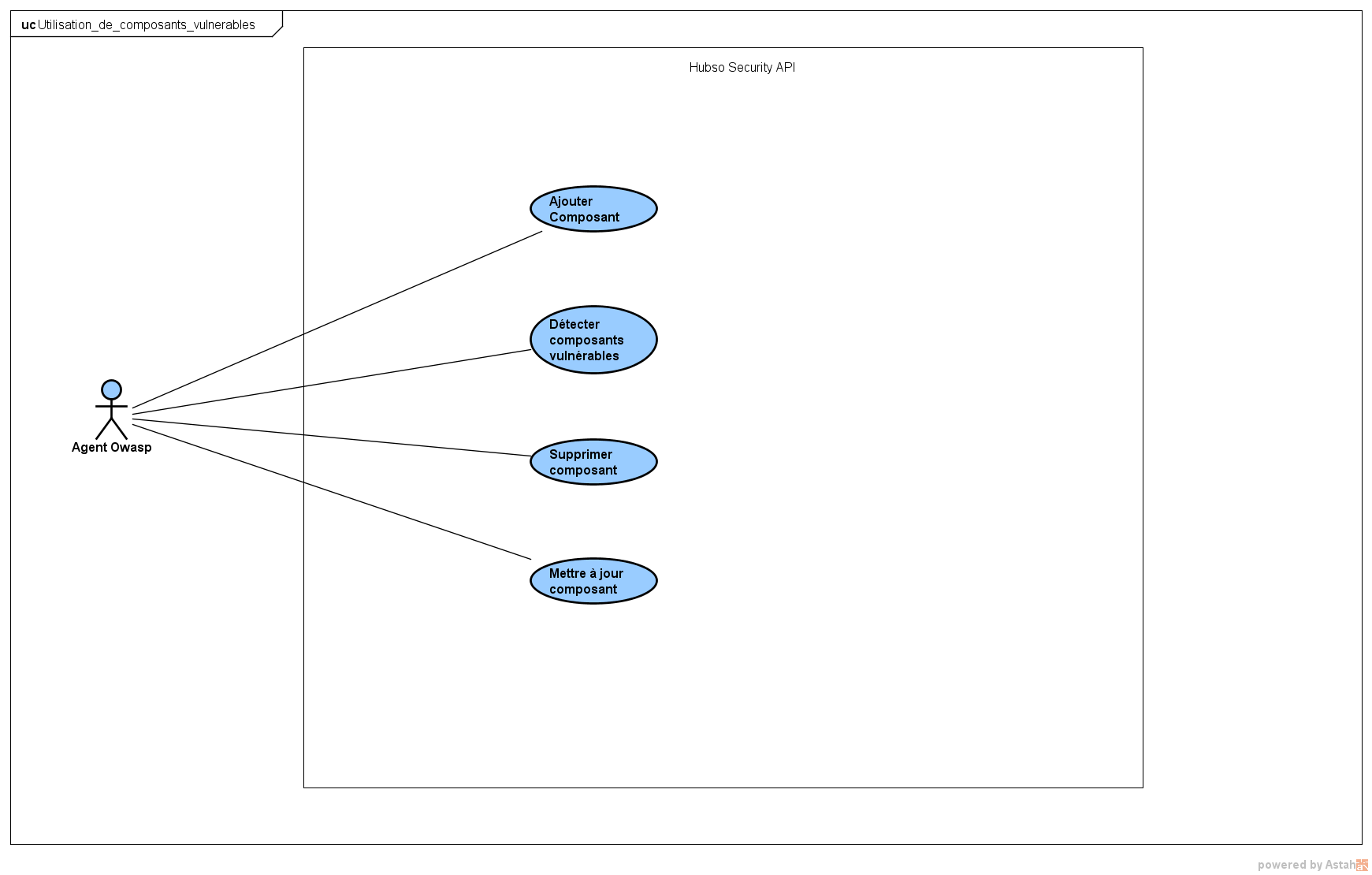
* 1. Fiches de description textuelle des cas d’utilisation
  2. Diagramme d’activités des cas d’utilisation

1. **Violation de Contrôle d’accès**
   1. Analyse du risque « Violation de Contrôle d’accès »
   2. Diagramme de cas d’utilisation
   3. Fiches de description textuelle des cas d’utilisation
   4. Diagramme d’activités des cas d’utilisation
2. **Mauvaise Configuration de Sécurité**
   1. Analyse du risque « Mauvaise Configuration de Sécurité »
   2. Diagramme de cas d’utilisation
   3. Fiches de description textuelle des cas d’utilisation
   4. Diagramme d’activités des cas d’utilisation
3. **Cross-Site Scripting (XSS)**
   1. Analyse du risque « XSS »
   2. Diagramme de cas d’utilisation



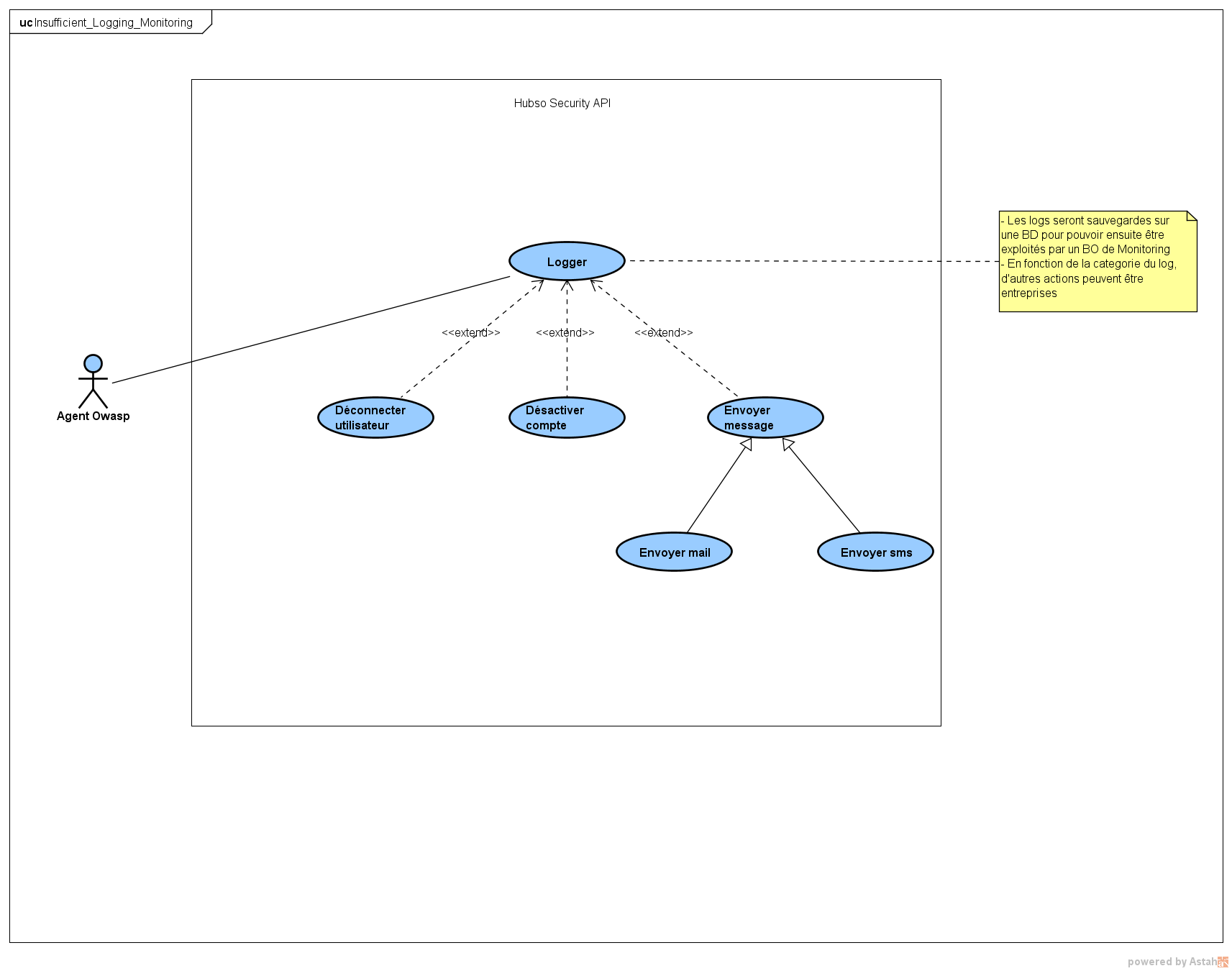
* 1. Fiches de description textuelle des cas d’utilisation
  2. Diagramme d’activités des cas d’utilisation

1. **Désérialisation non sécurisée**
   1. Analyse du risque « Désérialisation non sécurisée »
   2. Diagramme de cas d’utilisation
   3. Fiches de description textuelle des cas d’utilisation
   4. Diagramme d’activités des cas d’utilisation
2. **Utilisation de Composants Vulnérables**
   1. Analyse du risque « Utilisation de composants vulnérables »
   2. Diagramme de cas d’utilisation



* 1. Fiches de description textuelle des cas d’utilisation
  2. Diagramme d’activités des cas d’utilisation

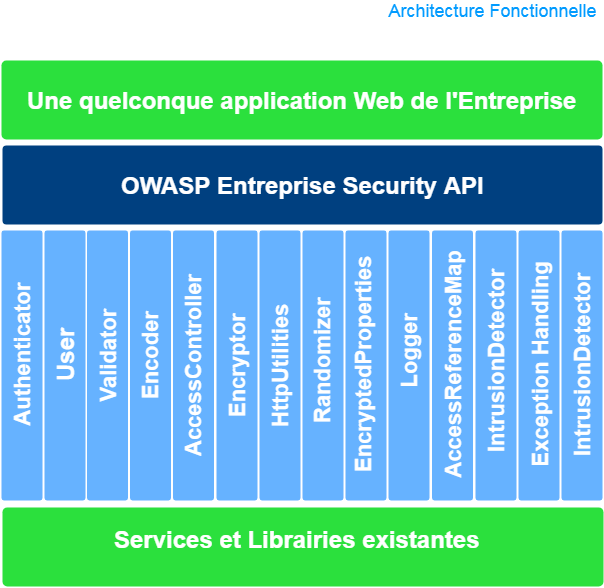
1. **Logging et Monitoring insuffisants** 
   1. Analyse du risque « Logging et Monitoring insuffisants »
   2. Diagramme de cas d’utilisation



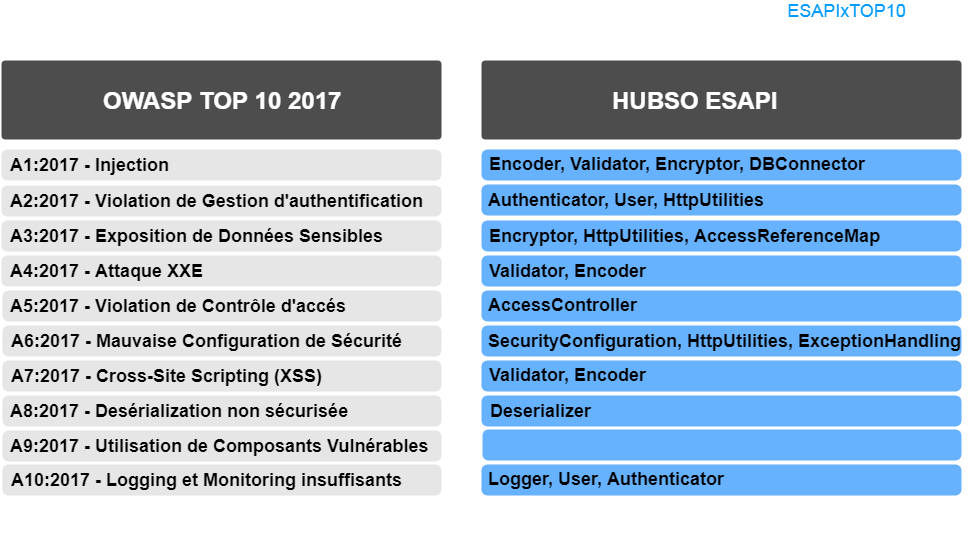
* 1. Fiches de description textuelle des cas d’utilisation
  2. Diagramme d’activités des cas d’utilisation

**Conception**

1. Architecture Fonctionnelle



1. Architecture Fonctionnelle détaillée



1. Architecture Technique

