

20/10/2025

# DH-CNS

Réunion technique hôpitaux

# Mentions légales

## Informations de nature générale

Les informations reprises dans cette présentation sont de nature générale. Elles ne sont pas adaptées à des circonstances personnelles ou spécifiques. Elles ne peuvent pas être considérées comme des conseils personnels, professionnels ou juridiques. Si vous avez besoin de conseils personnels ou spécifiques, consultez un des services de la CNS ou de vos caisses de maladie du secteur public, en fonction du thème de votre demande.

## Informations officielles

Seuls les textes officiels publiés au Mémorial sont considérés comme authentiques. Les informations et les documents disponibles au sein de cette présentation ne peuvent pas être considérés comme une reproduction authentique des textes officiels. En cas de différences, le texte officiel publié au Mémorial prime toujours.

## Contenu et responsabilité

La CNS consent de gros efforts pour que les informations mises à disposition au sein de cette présentation soient complètes et correctes. Ces informations n'ont cependant pas une vocation d'exhaustivité ou de constituer un engagement de la part de la CNS. Si vous désirez poser des questions au sujet de cette présentation, vous pouvez nous contacter à [cns@secu.lu](mailto:cns@secu.lu).

## Informations d'autres autorités, instances et organisations

Cette présentation peut comporter des informations d'autres autorités, instances et organisations sur lesquels la CNS n'exerce aucun contrôle. La CNS n'offre aucune garantie quant au caractère exhaustif ou exact du contenu de ces informations. La CNS décline toute responsabilité pour les dommages directs ou indirects résultant de l'utilisation des ces informations et de leurs contenus.

# Sommaire

- Introduction/contexte – CNS
- Authentification AeS – Agence eSanté
- eConnector SaaS – Incert
- Roadmap
- Questions/réponses

# Introduction / contexte



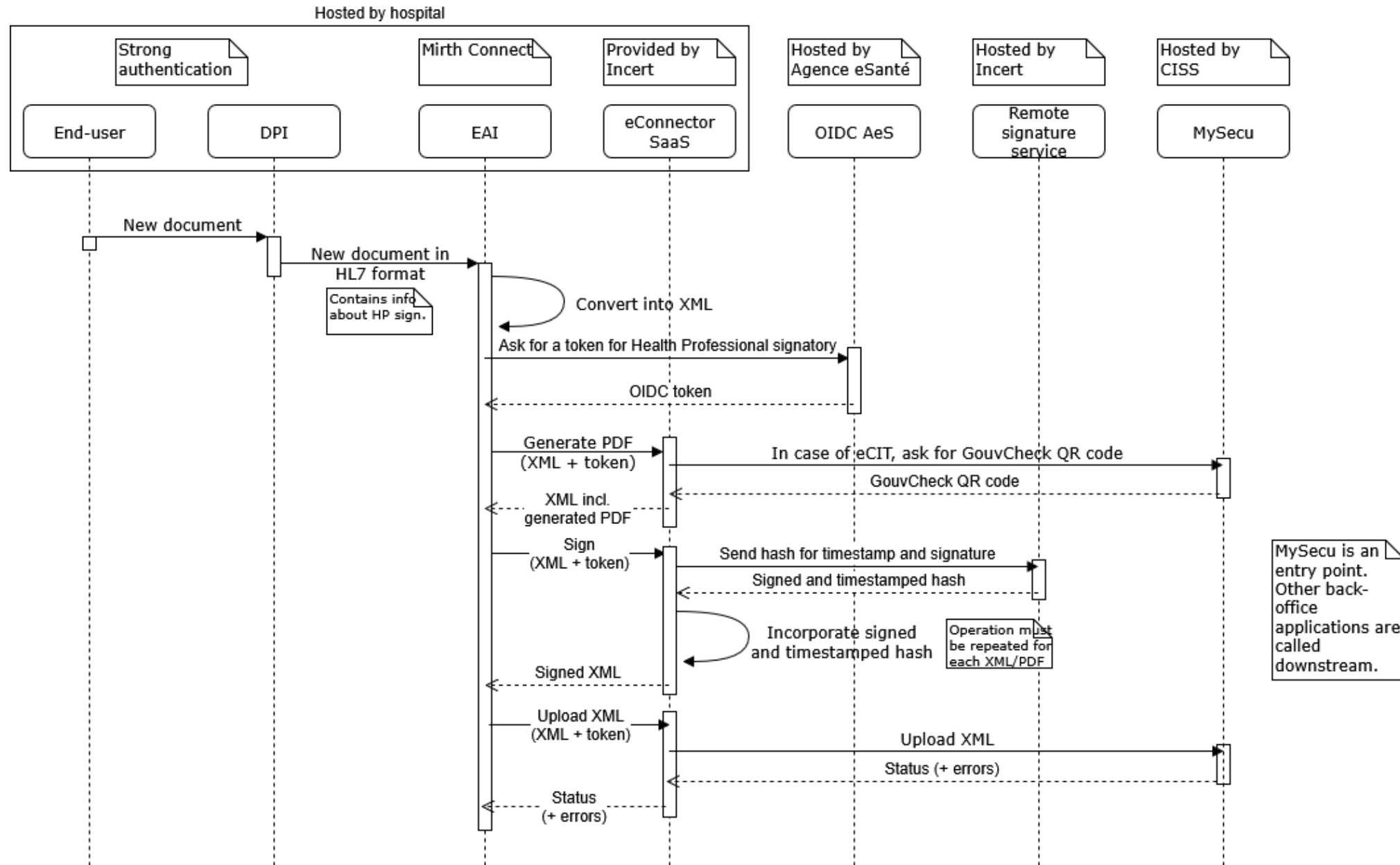
# Introduction

- Solution dédiée aux hôpitaux
  - Utilisation d'un système d'authentification forte (badges hôpitaux)
- Conception de la solution basée sur
  - eConnector SaaS
    - Hébergé dans le SI des hôpitaux
  - Echanges avec CIO des hôpitaux pour l'authentification AeS et les services offerts par l'eConnector SaaS

# Contexte

- Documents produits par SI hôpitaux
  - Pas ceux produits par "cabinets libéraux" au sein des hôpitaux
- Requêtes toujours faites par un médecin
  - Ou au nom d'un médecin dans le cas d'un mémoire d'honoraires
- Développements dans le DPI complexes à réaliser
- Par contre, possibilité de sortir les informations du DPI en HL7
- Ensuite, prise en charge par l'EAI (Mirth Connect)
  - Conversion HL7 ==> XML prise en charge par l'EAI
  - Appels vers eConnector SaaS réalisés par l'EAI
  - Rupture dans la chaîne d'authentification
    - Message HL7 ne contient pas le jeton d'authentification original de l'utilisateur
    - C'est l'EAI qui va s'authentifier et demander un jeton au nom du médecin

# Diagramme de séquence



# Authentication AeS



# Prérequis

## ■ Authentification

- Besoin d'un compte applicatif (client\_id, client\_secret)
- => Adresser demande à [support\\_labelisation@incert.lu](mailto:support_labelisation@incert.lu) - en utilisant procédure communiquée (voir slide 26)

## ■ Besoin de l'eHealthId du professionnel de santé qui fera l'envoi

## ■ Audience cible: <https://eadmin.lu/>

# Requête d'impersonalisation

POST <https://xxx.esante.lu/auth/realms/default/protocol/openid-connect/token>

Content-Type: application/x-www-form-urlencoded

client\_id = <source-client> &

client\_secret = <source\_client\_secret> &

grant\_type = urn:ietf:params:oauth:grant-type:token-exchange &

requested\_subject = <eHealthId-Physician> &

audience = <target-client>

## URLs:

**Intégration:** <https://www-integration.esante.lu>

**Production:** <https://www.esante.lu>

# Réponse d'impersonalisation - OK

HTTP 200 OK

....

```
{  
  "access_token": "eyJhbGciOiJSUzI1Ni.....",  
  "expires_in": 300,  
  "refresh_expires_in": 1800,  
  ....  
}
```

# Réponse d'impersonalisation – Cas d'erreurs

| Code erreur | Interprétation                                                                                        |
|-------------|-------------------------------------------------------------------------------------------------------|
| HTTP-401    | Erreur d'authentification (« client_id » et « client_secret » incorrect)                              |
| HTTP-403    | Client non autorisé pour token exchange<br>Professionnel non trouvé (« requested_subject » non connu) |

# Réponse d'impersonalisation - NOK

HTTP 403 NOK

....

```
{  
  "error": "access_denied",  
  "error_description": "Client not allowed to exchange"  
}
```

# eConnector SaaS



# Prérequis

## ■ mTLS

- Chaque entité hospitalière doit disposer d'un certificat d'authentification pour s'interfacer avec mySecu :
  - Il est nécessaire de générer une requête de signature de certificat (CSR).
    - Profil clé : RSA 4096 bits
  - Transmettre au CISS via Incert ([support\\_labelisation@incert.lu](mailto:support_labelisation@incert.lu)) la CSR pour recevoir le certificat correspondant.

## ■ Environnement autorisant des déploiements d'image Docker.

## ■ Récupération de l'image depuis iHUB

- Test url :  
<https://ihub-test.cns.secu.lu/downloads/eConnector-saas/1.1.6/eConnector-saas-1.1.6-integration.tar>

# Prérequis

## ■ Dimensionnement des machines hôtes

| Environnement                    | vCPU   | RAM    | Justification                                                          |
|----------------------------------|--------|--------|------------------------------------------------------------------------|
| Développement / test             | 1 vCPU | 2 Go   | Suffisant pour tester la génération PDF et signature ponctuelle.       |
| Préproduction                    | 2 vCPU | 4 Go   | Permet de simuler un usage réaliste et plusieurs requêtes simultanées. |
| Production (charge <50 req/min)  | 2 vCPU | 4–6 Go | Si peu d'appels simultanés.                                            |
| Production (charge > 50 req/min) | 4 vCPU | 8 Go   | Bon équilibre pour supporter plusieurs signatures/PDF en parallèle.    |

# Prérequis

## ■ Ouverture réseau sur les URL suivantes :

### ■ Intégration :

- <https://ihub-test.cns.secu.lu/>\* (service de mise à jour des templates, optionnel)
- <https://hoki-uat-tsu.incert.lu/>\* (service de signature)
- <https://ws.mysecu.lu:7443/>\* (dépôt sur mySecu)

### ■ Production :

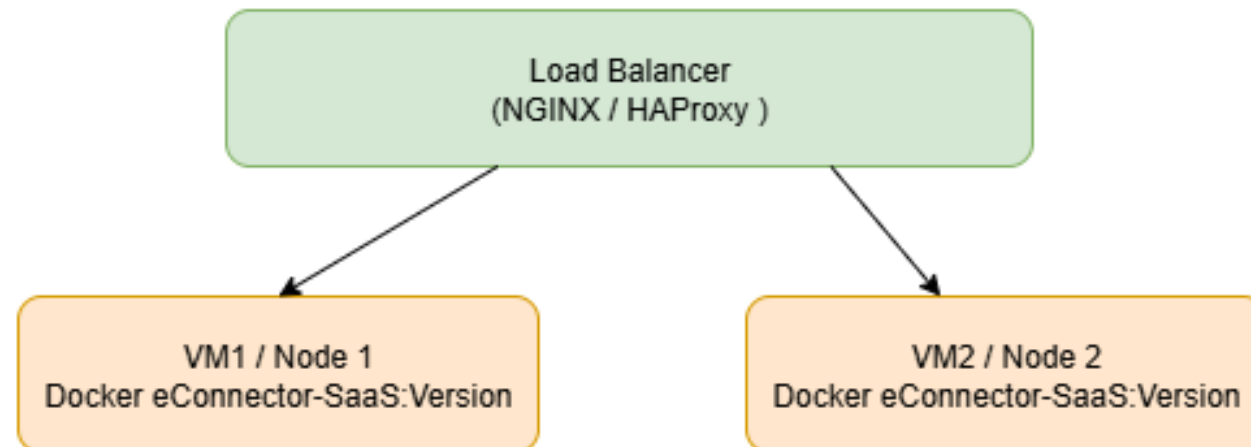
- <https://ihub.cns.secu.lu/>\* (service de mise à jour des templates, optionnel)
- <https://hoki-tsu-kms.incert.lu/>\* (service de signature)
- <https://ws.mysecu.lu/>\* (dépôt sur mySecu)

## ■ Enrôlement des professionnels de santé sur mySecu :

- Signature électronique au nom du PS
- → Obtention d'un jeton pour le PS
- → Le PS doit être connu dans mySecu
- Utilisation du code facturier possible pour les associations de PS
- URL par environnement
  - Intégration : <https://mysecu-test.services-publics.lu>
  - Production : <https://secu.services-publics.lu/>

# Installation

- Faisabilité de la haute disponibilité (HA)
  - L'architecture de l'eConnector SaaS présente plusieurs avantages facilitant la mise en place d'une HA horizontale :
    - Aucune base de données requise
    - Traitements *stateless* (chaque requête est indépendante)
    - Exécution dans un conteneur Docker
    - Utilisation d'un *Spring Boot Actuator* pour la supervision et le monitoring



# Installation – configuration de l'image

| Variable                          | Description                                                                                                          | UAT Example                                                 | Production Example                                          |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|-------------------------------------------------------------|
| <b>DEBUG_LOG</b>                  | Enables or disables debug logs (true / false).                                                                       | true                                                        | false                                                       |
| <b>KMS_TIMESTAMP_URL</b>          | Timestamp service URL.                                                                                               | https://hoki-uat-tsu.incert.lu/api/v3/sign-timestamp        | https://hoki-tsu-kms.incert.lu/api/v3/sign-timestamp        |
| <b>KMS_SIGN_URL</b>               | Remote signature service URL.                                                                                        | https://hoki-uat-tsu.incert.lu/api/v3/sign-data             | https://hoki-tsu-kms.incert.lu/api/v3/sign-data             |
| <b>KMS_GET_CERTIFICATE_URL</b>    | Certificate retrieval URL.                                                                                           | https://hoki-uat-tsu.incert.lu/api/v3/get-certificate       | https://hoki-tsu-kms.incert.lu/api/v3/get-certificate       |
| <b>TRUSTSTORE_KMS_FILENAME</b>    | Truststore filename in resource                                                                                      | truststore-integration.p12                                  | truststore-prod.p12                                         |
| <b>GENERIC_KEYSTORE_FILE</b>      | Path or filename of the generic keystore (generated from the CSR fo CCSS).                                           | keystore-integration.p12                                    | keystore-prod.p12                                           |
| <b>GENERIC_KEYSTORE_PWD</b>       | Generic keystore password                                                                                            | xxxxxx                                                      | xxxxxx                                                      |
| <b>CCSS_URL</b>                   | mySecu authentication service.                                                                                       | https://ws.mysecu.lu:7443/ws/soap/trust?wsdl                | https://ws.mysecu.lu:443/ws/soap/trust?wsdl                 |
| <b>CCSS_SYNC_URL</b>              | mySecu business service endpoint.                                                                                    | https://ws.mysecu.lu:7443/ws/soap/espinst/syncexchange      | https://ws.mysecu.lu:443/ws/soap/espinst/synceexchange      |
| <b>SYNC_EXCHANGE_WSDL</b>         | mySecu business WSDL.                                                                                                | https://ws.mysecu.lu:7443/ws/soap/espinst/syncexchange?wsdl | https://ws.mysecu.lu:443/ws/soap/espinst/synceexchange?wsdl |
| <b>IHUB_URL</b>                   | iHub update & template REST service.                                                                                 | https://ihub-test.cns.secu.lu/ws/rest/ihub/v1               | https://ihub.cns.secu.lu/ws/rest/ihub/v1                    |
| <b>SAAS_KEYSTORE</b>              | Keystore file that stores the server's private key and certificate, enabling secure HTTPS connections for the image. | saas-keystore-integration.p12                               | saas-keystore-prod.p12                                      |
| <b>SAAS_KEYSTORE_PASSWORD</b>     | Keystore password.                                                                                                   | xxxxxxx                                                     | xxxxxxx                                                     |
| <b>SAAS_KEYSTORE_KEY_PASSWORD</b> | Private key password.                                                                                                | xxxxxxx                                                     | xxxxxxx                                                     |
| <b>SIGNING_CA_ALIAS</b>           | Alias of the signing CA in the truststore.                                                                           | test_signing_ca                                             | signing_ca                                                  |
| <b>TIMESTAMP_CA_ALIAS</b>         | Alias of the timestamp CA in the truststore.                                                                         | test_timestamp_ca                                           | timestamp_ca                                                |
| <b>ROOT_SIGNING_CA_ALIAS</b>      | Alias of the root signing CA in the truststore.                                                                      | test_root_signing_ca                                        | root_signing_ca                                             |
| <b>SERVER_PORT</b>                | Server HTTPS port (default: 8443).                                                                                   | 8443                                                        | 8443                                                        |

# Installation

## ■ Lancement du conteneur:

```
docker run -d \  
-e DEBUG_LOG=true \  
-e KMS_TIMESTAMP_URL=https://hoki-uat-tsu.incert.lu/api/v3/sign-timestamp \  
-e KMS_SIGN_URL=https://hoki-uat-tsu.incert.lu/api/v3/sign-data \  
-e KMS_GET_CERTIFICATE_URL=https://hoki-uat-tsu.incert.lu/api/v3/get-certificate \  
-e TRUSTSTORE_KMS_FILENAME=truststore-integration.p12 \  
-e GENERIC_KEYSTORE_FILE=keystore-integration.p12 \  
-e GENERIC_KEYSTORE_PWD=xxxxx \  
-e CCSS_URL=https://ws.mysecu.lu:7443/ws/soap/trust?wsdl \  
-e CCSS_SYNC_URL=https://ws.mysecu.lu:7443/ws/soap/espinst/syncexchange \  
-e SYNC_EXCHANGE_WSDL=https://ws.mysecu.lu:7443/ws/soap/espinst/syncexchange?wsdl \  
-e IHUB_URL=https://ihub-test.cns.secu.lu/ws/rest/ihub/v1 \  
-e JAVA_TOOL_OPTIONS="-Dapp.jasper.path=/opt/jasper/templates" \  
-e SAAS_KEYSTORE=saas-keystore-integration.p12 \  
-e SAAS_KEYSTORE_PASSWORD=xxxxxx \  
-e SAAS_KEYSTORE_KEY_PASSWORD=xxxxxx \  
-e SIGNING_CA_ALIAS=test_signing_ca \  
-e TIMESTAMP_CA_ALIAS=test_timestamp_ca \  
-e ROOT_SIGNING_CA_ALIAS=test_root_signing_ca \  
-e SERVER_PORT=8443 \  
-p 8443:8443 \  
econnector-saas:latest
```

- Le lancement de l'image nécessite l'acceptation des CGU CNS, autrement le lancement échouera.

Vous pouvez ajouter argument suivant pour les accepter

```
-e ACCEPT_LICENSE=true
```

Si l'argument n'est pas présent, le lancement échoue, les CGU sont affichées dans le terminal avec le moyen de les récupérer.

# Utilisation de l'eConnector

## ■ Vérification de l'état du eConnector

- Pour s'assurer que l'image est bien déployée et que les services sont accessibles , il suffit de faire un call sur l'endpoint suivant :

`https://domain:{port}/econnector/status`

- Un retour 200 indique que l'eConnector est utilisable

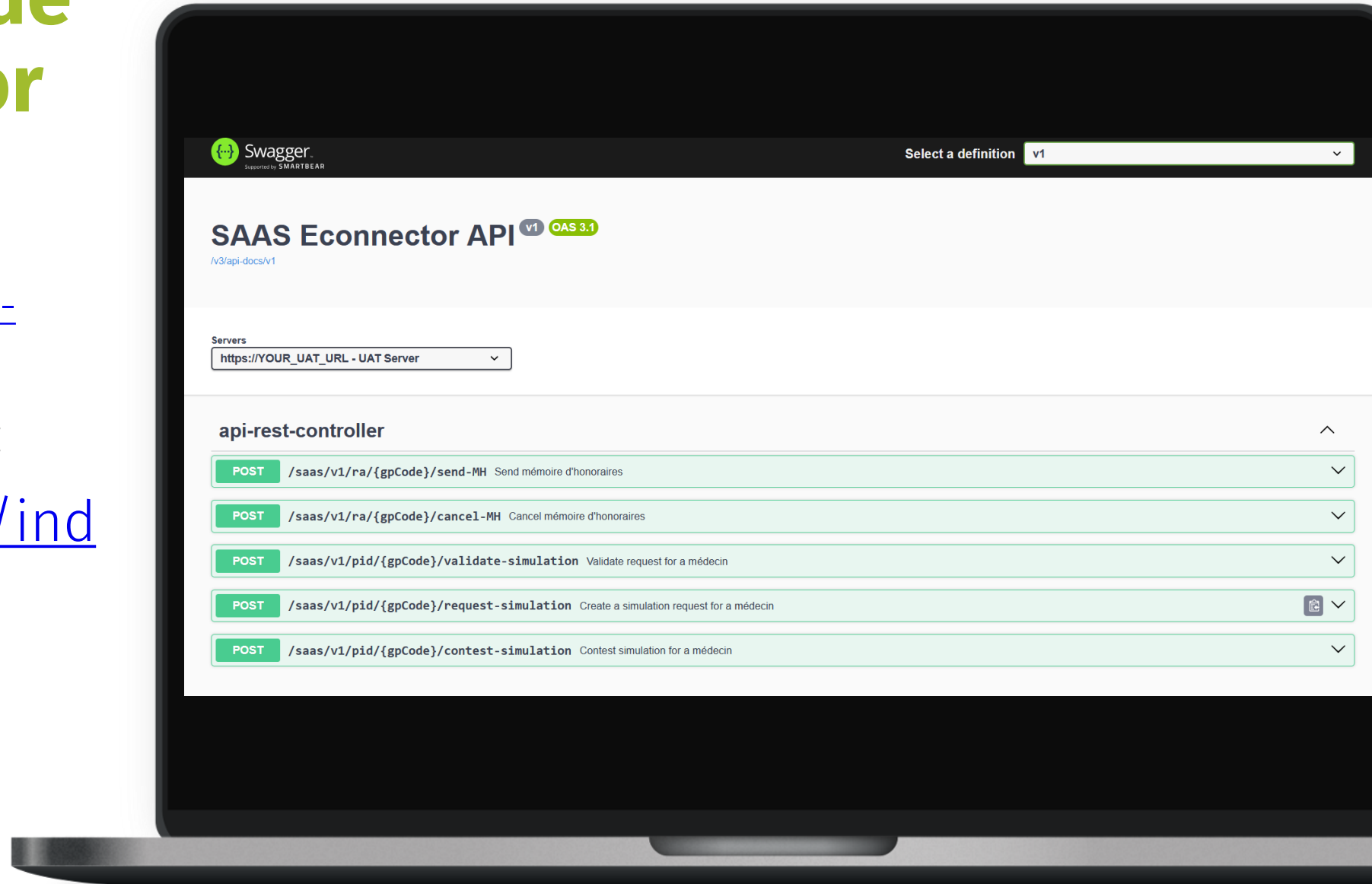
# Utilisation de l'eConnector

■ Api Swagger:

<https://domain:{port}/swagger-ui/index.html>

■ Colletion Postaman:

<https://drive.incert.lu/index.php/f/14220>

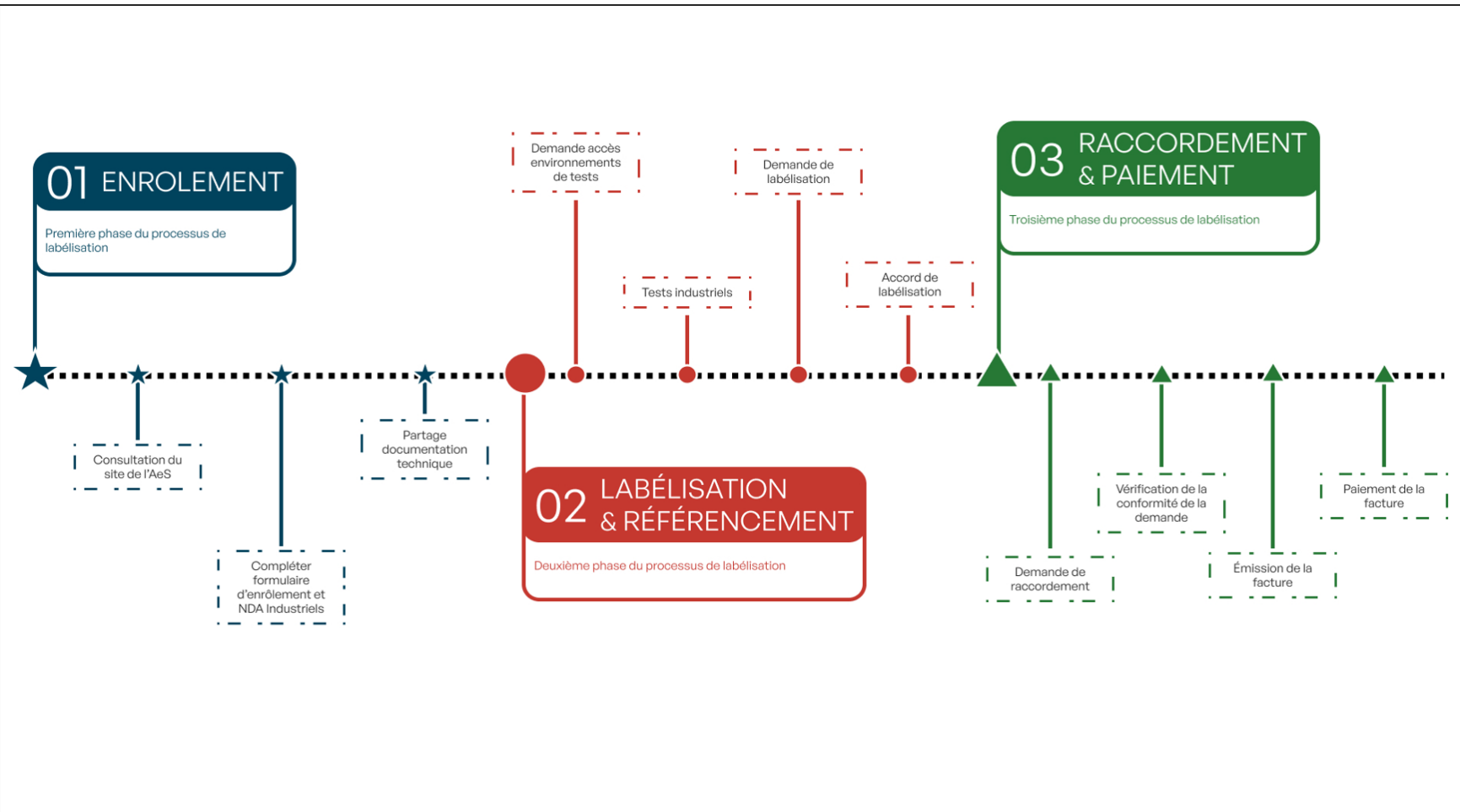


# Formulaires dans le cadre de la labélisation

Objectifs :

- Partager les formulaires
- A quelles étapes les renseigner dans le cadre de la labélisation
- Les adapter pour qu'ils répondent aux besoins

# Labélisation et formulaires



Accès environnements de tests : **Formulaire de demande d'accès aux env. de tests**

Labélisation : **Formulaire de labélisation**

Raccordement : **Formulaire de demande de raccordement**

Facture : **Formulaire de facturation**

# Formulaire demande d'accès environnements de tests

- **Objectif :** accéder aux environnements de tests
- **Informations :**
  - Pourront être renseignés par INCERT les éléments relatifs à l'établissement hospitalier et au représentant légal
  - Téléphone Mobile obligatoire pour gérer la double authentification
  - 4 cas pour la demande d'accès aux environnements de tests
- **Après la réussite des tests :**
  - Une attestation de réussite signée par les entités INCERT, AeS, CNS, CISS est délivrée.
  - Formation à la gestion des incidents

# Formulaire demande d'accès service authentification AeS

- **Objectif :** accéder au service d'authentification AeS
- **Informations :**
  - Nom de l'établissement hospitalier
  - Coordonnées de deux personnes de contact
  - Adresse email pour la réception des identifiants du client OIDC
- Formulaire en cours de rédaction

# Formulaire demande de labélisation

- **Prérequis** : signature de la convention de service de l'Agence eSanté
- **Objectifs** : la signature de l'attestation de labélisation permet le déploiement des services en production
- **Informations** :
  - La signature de la convention AeS et de la labélisation permet le référencement par l'Agence eSanté
  - Lister le nombre de professionnels raccordés
  - Partager des éléments montrant la formation des professionnels de santé.
- **Après la labélisation** : Demander le raccordement

# Formulaire demande de raccordement

- **Objectifs :** permettre de lister le nombre de professionnels raccordés
- **Informations :**
  - Renseigner le Cas n°2 du formulaire de raccordement
  - Renseigner les identifiants MH et PID en production \*
  - Lister l'ensemble des professionnels raccordés dans le fichier Liste de raccordement
- **Après le raccordement :**
  - Emettre une facture

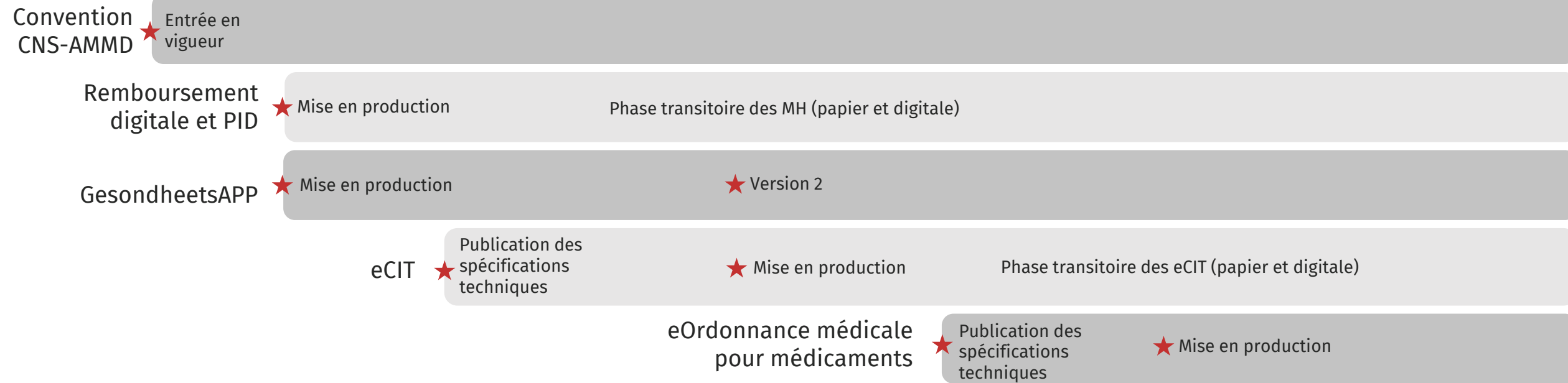
# Formulaire demande de facturation

- **Objectifs :** Permettre de lister le nombre de professionnels raccordés afin de bénéficier de l'incitation
- **Informations :**
  - A la demande de l'AeS, il sera indispensable de renseigner les noms/prénoms/spécialités et identifiants (eHealthID et Code prestataire ou GP Code) de tous les professionnels de santé raccordés en établissement hospitalier. En annexe de la demande de facturation, il faudra une liste Excel comportant toutes ces informations.
- **Après la demande de facturation :**
  - Paiement de l'incitation par l'AeS.

# Roadmap



# Calendrier site web V2



PID : Paiement Immédiat Direct

MH : Mémoires d'honoraires

eCIT : Certificat d'incapacité de travail électronique

# Questions/réponses



# Q&A

*Merci pour votre  
attention !*