

Guide d'appui

Outil de conformité pour Proxy e-Santé

Statut : Validé | Classification : Publique | Version : v1.1



SOMMAIRE

1. CONTEXTE	2
2. OBJECTIF	2
3. L'ESPACE DE CONFIANCE DE PRO SANTE CONNECT	3
4. L'OUTIL DE CONFORMITE.....	4
5. SCENARIOS DE TESTS.....	5
6. PREREQUIS.....	6
7. CONFIGURATION DE PLATINES	7
8. EXECUTION DES TESTS ET RESULTATS	8
9. SUITE DE TESTS.....	8
9.1. Scénario 1 : Cas Nominal - Connexion PS1 LPS1 + Requête API PSC.....	8
9.1.1. <i>Objectif.....</i>	8
9.1.2. <i>Diagramme de séquence.....</i>	9
9.1.3. <i>Résultats attendus</i>	10
9.1.4. <i>Scénario de test.....</i>	11
9.2. Scénario 2 : Cas Nominal - Connexion PS1 LPS1 et PS2 LP1 + Requête API PSC	13
9.2.1. <i>Objectif.....</i>	13
9.2.2. <i>Diagramme de séquence.....</i>	14
9.2.3. <i>Résultats attendus</i>	14
9.2.4. <i>Scénario de test.....</i>	14
9.3. Scénario 3 : Cas Nominal - Connexion PS1 LPS1 et PS1 LP2 + Requête API PSC	17
9.3.1. <i>Objectif.....</i>	17
9.3.2. <i>Diagramme de séquence.....</i>	18
9.3.3. <i>Résultats attendus</i>	18
9.3.4. <i>Scénario de test.....</i>	18
9.4. Scénario 4 : Cas Nominal - Connexion PS1 LPS1 et PS2 LP2 + Requête API PSC	21
9.4.1. <i>Objectif.....</i>	21
9.4.2. <i>Diagramme de séquence.....</i>	22
9.4.3. <i>Résultats attendus</i>	22
9.4.4. <i>Scénario de test.....</i>	22
9.5. Scénario 5 : Cas Nominal - Connexion PS1 LPS1 + Requête API PSC + déconnexion	25
9.5.1. <i>Objectif.....</i>	25
9.5.2. <i>Diagramme de séquence.....</i>	26
9.5.3. <i>Résultats attendus</i>	26
9.5.4. <i>Scénario de test.....</i>	27

1. CONTEXTE

Pro Santé Connect est un fournisseur d'identité sectorielle basé sur le standard OIDC qui permet aux professionnels de santé de s'authentifier aux services en lignes par la saisie d'un code PIN qui permet soit de lire le certificat contenu dans une carte CPS, soit d'ouvrir le certificat de l'application smartphone e-CPS.

En plus de l'authentification, **Pro Santé Connect propose un espace de confiance dans lequel les services interconnectés peuvent échanger des données de façon sécurisée**. Cet espace de confiance propose une traçabilité et une sécurité poussée qui permettent de retracer tout le parcours des échanges, depuis le PS jusqu'à l'API Pro Santé Connectée en passant par un proxy e-Santé.

Afin de simplifier le processus d'adhésion à l'espace de confiance, **l'ANS propose un outil de conformité aux éditeurs de proxy e-Santé**

2. OBJECTIF

Le **Proxy e-Santé** est le serveur intermédiaire, ne disposant pas d'interface utilisateur, destiné à **sécuriser les échanges de données entre le fournisseur de service Utilisateur (FSU) et l'API Pro Santé Connectée**.

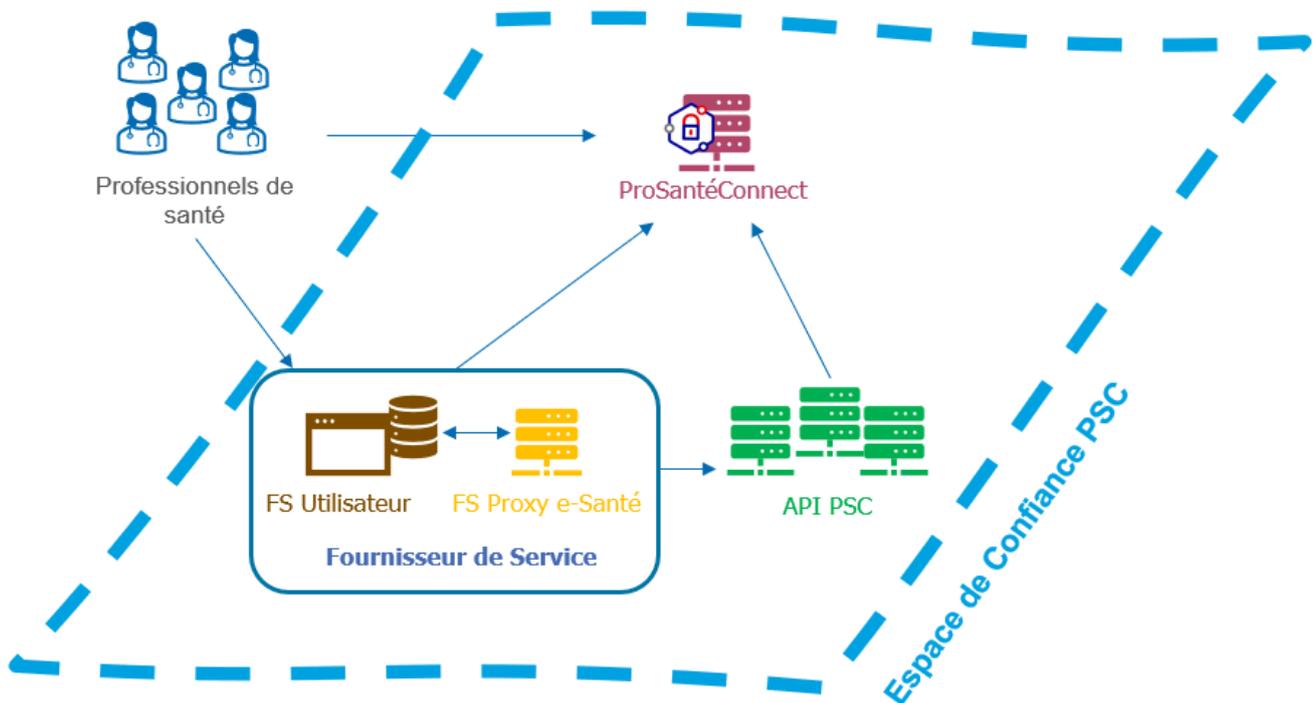
Cet outil de test permet aux éditeurs de Proxy e-Santé de valider la conformité de leur composant en testant en autonomie et à leur rythme que ce dernier réponde à un sous-ensemble d'exigences de l'espace de confiance du Proxy e-Santé garantissant une sécurité commune à tous.

3. L'ESPACE DE CONFIANCE DE PRO SANTE CONNECT

L'espace de confiance est un constitué d'un ensemble de composants qui permettent des échanges en toute sécurité aux différents intervenants dans cet espace.

Il est composé de :

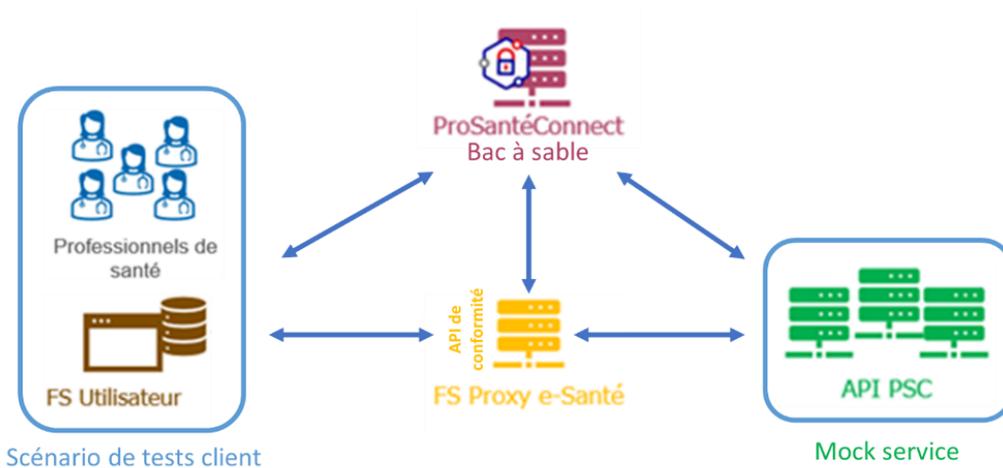
- Un mécanisme d'authentification basé sur OpenID Connect utilisant ClientID + certificat x509
- Une API Crédential, qui liste des clients ID des services qui sont habilités "espace de confiance"
- Des Proxy e-Santé qui garantissent la séparation des contextes et la traçabilité des échanges
- Des API Pro Santé Connectées qui fournissent des données métier uniquement dans cet espace



4. L'OUTIL DE CONFORMITE

Cet outil a pour objectif d'aider les éditeurs de Proxy e-Santé à tester leur système pour s'assurer qu'il répond aux spécifications du référentiel.

Il est disponible dans l'environnement Bac à Sable de Pro Santé Connect.



Il s'articule autour de **3 composants** :

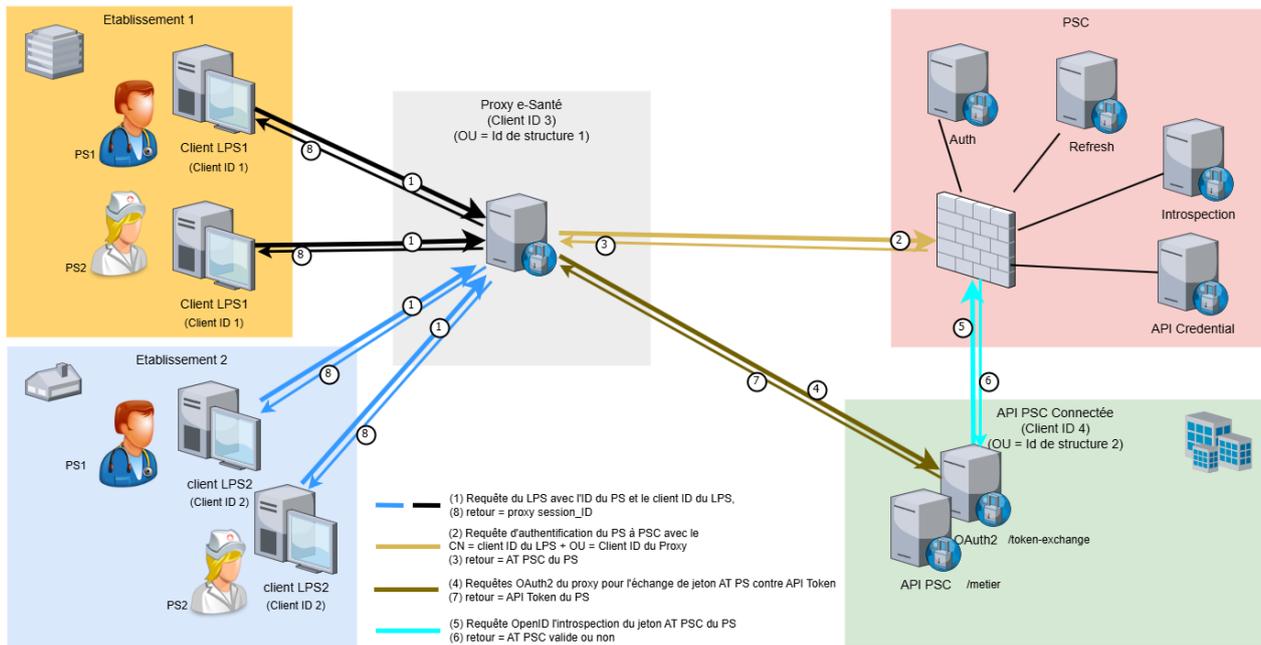
- **Le scénario de Tests client** :
Ce composant envoie les requêtes prévues par le plan de tests et simule les actions des PS utilisant des logiciels PS (LPS) ou toutes autres solutions impliquant des appels à des API Pro Santé Connectées via un Proxy e-Santé donné.
- **L'API de conformité** :
Ce composant permet d'obtenir par simple requête API, les informations présentes au niveau du Proxy e-Santé afin de valider que la gestion des informations répond aux exigences. Afin de collecter ces informations (contexte des requêtes « id PS, id LPS » et des connexions), une API spécifique aux tests de conformité et imposée par l'ANS. Elle être exposée par le Proxy e-Santé pour permettre :
 - ✓ D'initialiser la connexion entre le PS-LPS et le Proxy e-Santé (endpoint /connect)
 - ✓ D'invalider la session en cours (endpoint /disconnect)
 - ✓ De transmettre certaines requêtes à l'API PSC (endpoint /send)
 - ✓ De retourner les traces stockées au niveau du Proxy e-Santé (endpoint /traces)
- **Le mock service API PSC** :
Ce composant simule un fournisseur de données, qui doit répondre aux différents cas de tests du scénario, en retournant des données prévues par le scénario. Pour cela il expose une API Pro Santé Connectée au sens du CI-SIS.

5. SCENARIOS DE TESTS

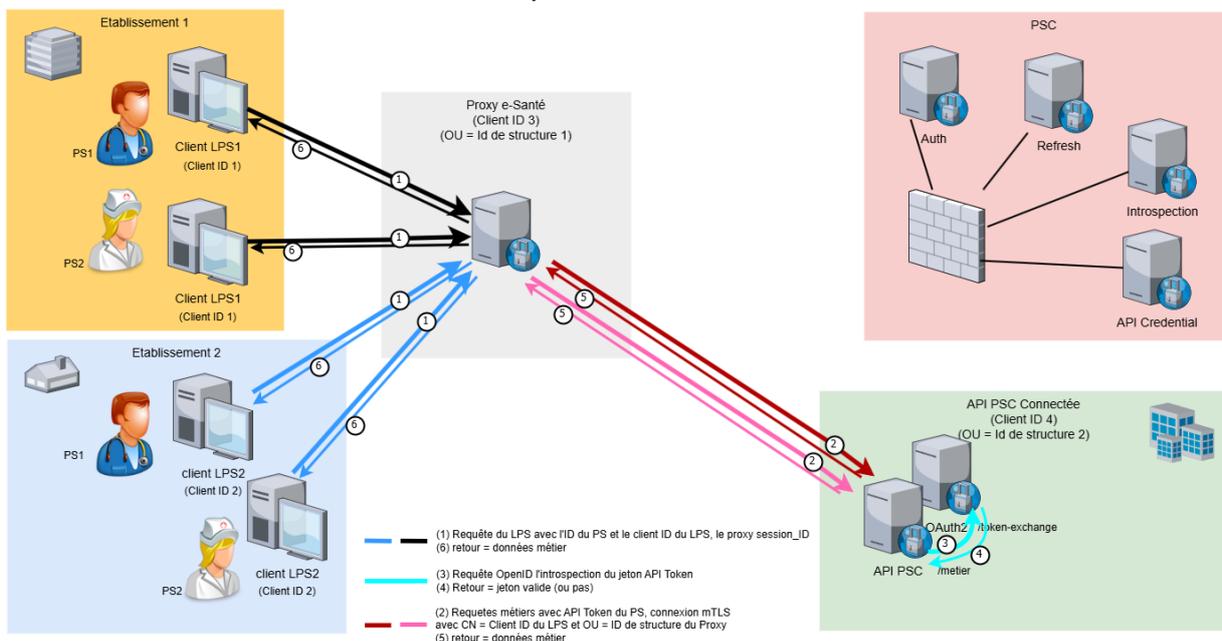
Pour les besoins des tests le système impliquera :

- 2 Professionnels de santé de test, ayant chacun activé leur e-CPS respectives
- 2 Logiciels de professionnel de santé (LPS)
- Le Proxy e-Santé à tester
- 1 API Pro Santé Connectée
- L'environnement Pro Santé Connect Bac à Sable

Espace de confiance PSC - Tests de conformité Connexion



Espace de confiance PSC - Tests de conformité Requêtes métier



Afin de valider la conformité du Proxy e-Santé aux exigences de l'espace de confiance, la suite de cas métier comprendra les cinq scénarios suivants :

- Scénario 1 : Cas Nominal - Connexion PS1 LPS1 + Requête API PSC
- Scénario 2 : Cas Nominal - PS1 LPS1 et PS2 LPS1+ Requête API PSC
- Scénario 3 : Cas Nominal - PS1 LPS1 et PS1 LPS2 + Requête API PSC
- Scénario 4 : Cas Nominal - PS1 LPS1 et PS2 LPS2 + Requête API PSC
- Scénario 5 : Cas Nominal - PS1 LPS1 + Requête API PSC + déconnexion

L'outil de test qui est utilisé pour exécuter ces scénarios s'appelle "Platines" c'est un outil de test d'interopérabilité déjà utilisé par le ROR, TOM, et le RASS.

6. PREREQUIS

Comme vu dans le paragraphe précédent avant de pouvoir exécuter les tests de conformité, l'éditeur doit avoir réalisés les points suivants :

- ✓ Disposer de 2 CPS de test ou e-CPS de test (deux téléphones)
- ✓ Avoir configuré l'outil de test Platines
 - Création de la chaine de confiance
 - Création de l'application « consommateur de données » qui utilise la chaine de confiance
 - Création de l'application « fournisseur de données »
- ✓ Disposer des certificats d'Authentification IGC-SANTE ELEMENTAIRE ORGANISATION de test pour la connexion des PS auprès de PSC bac-à-sable :
 - Certificat 1 : CN=Client ID du LPS 1 (ans-odc-lps1-edc-bas) + OU=id de structure de test
 - Certificat 2 : CN=Client ID du LPS 2 (ans-odc-lps2-bas) + OU=id de structure de test
 - Les certificats sont utilisés par le proxy e-Santé pour la connexion CIBA des PS 1 et 2 à Pro Santé Connect "au nom" des LPS 1 et 2
 - Ces certificats sont également utilisés pour les requêtes SSL entre le Proxy e-Santé et l'API PSC

Remarque : Le processus de commande de certificats est disponible à cette adresse :
<https://industriels.esante.gouv.fr/produits-et-services/certificats-logiciels>

7. CONFIGURATION DE PLATINES

Un guide de l'utilisateur de Platines est disponible [ici](#), néanmoins il est important de comprendre que l'outil Platines est avant tout un outil de test d'interopérabilité.

Ce qui signifie qu'il teste soit

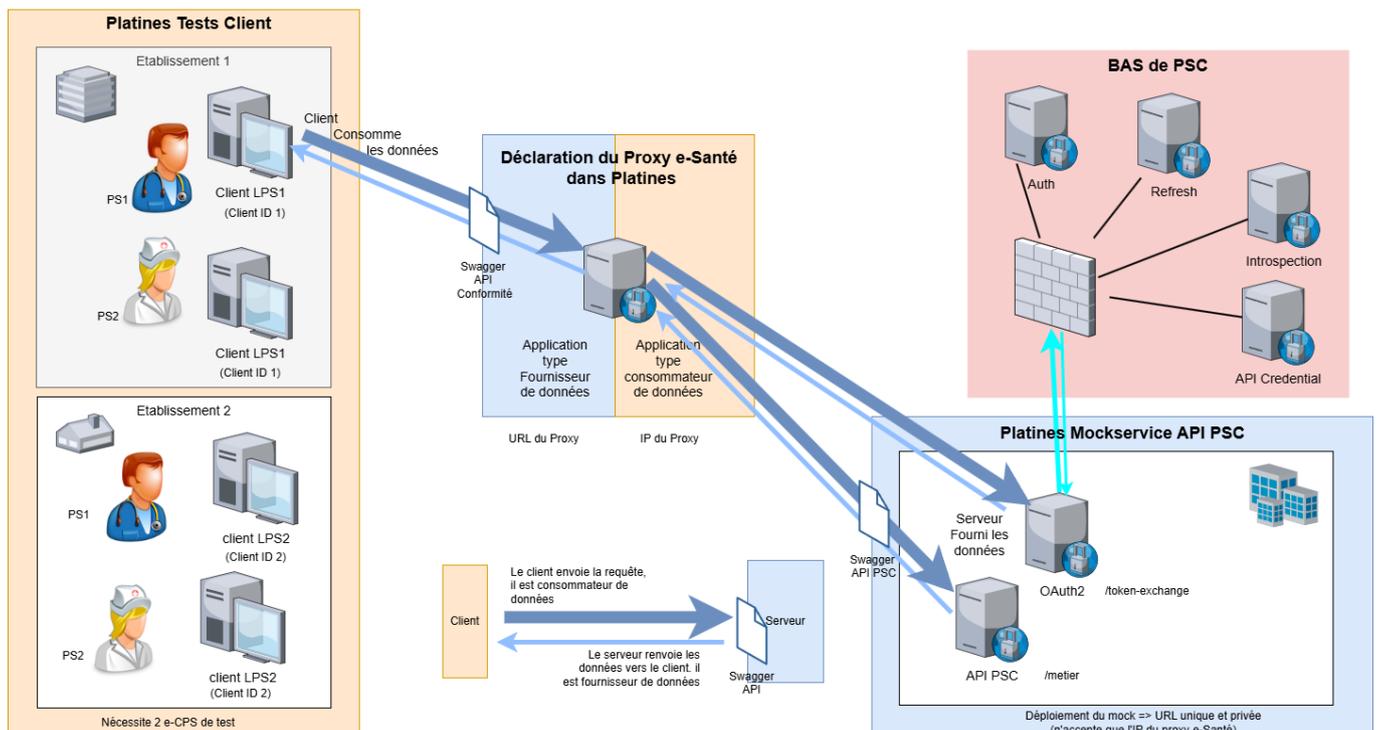
- le comportement d'un client qui effectue des requêtes vers un serveur (conformité technique et métier des requêtes)
- le comportement d'un serveur qui envoie des données à un client (conformité technique et métier des données envoyées par le serveur)

Dans le cadre de l'Espace de Confiance, le Proxy e-Santé joue les deux rôles,

Donc il faut configurer deux applications dans l'outil Platines :

- Une application de type "Fournisseur de données" : le côté API de conformité qui renvoie les données au client qui réalise les requêtes du scénario de test. Cette application doit avoir une URL accessible depuis internet.
- Une application de type "consommateur de données" : le côté du proxy e-Santé qui interroge l'API PSC et qui consomme ses données. Cette application doit présenter une adresse IP qui est enregistrée en liste blanche pour accéder au mock API PSC.

Intégration du Proxy e-Santé dans l'outil Platines PLAteforme de Test d'INteropérabilité de la e-Santé



8. EXECUTION DES TESTS ET RESULTATS

L'exécution des tests s'effectue dans l'ordre suivant :

Dans Platines

- ✓ Déployer le Mock Service API PSC
- ✓ S'assurer que la session est active (durée de session à définir lors du déploiement)

Dans le Proxy e-Santé

- ✓ Enregistrer l'URL du Mock Service dans le Proxy e-Santé (URL reçue par e-mail)
- ✓ Enregistrer l'URL du endpoint d'échange de jeton de l'API PSC dans le Proxy e-Santé

Dans Platines

- ✓ Déployer le projet des tests API dans Platines

Sur Smartphone

- ✓ Accepter les demandes d'authentifications reçues sur les deux e-CPS de test pendant le déroulement du plan de tests

A l'issue du test, les résultats OK ou KO sont directement visibles dans Platines, et un rapport est téléchargeable depuis la session de test, ainsi que les logs.

9. SUITE DE TESTS

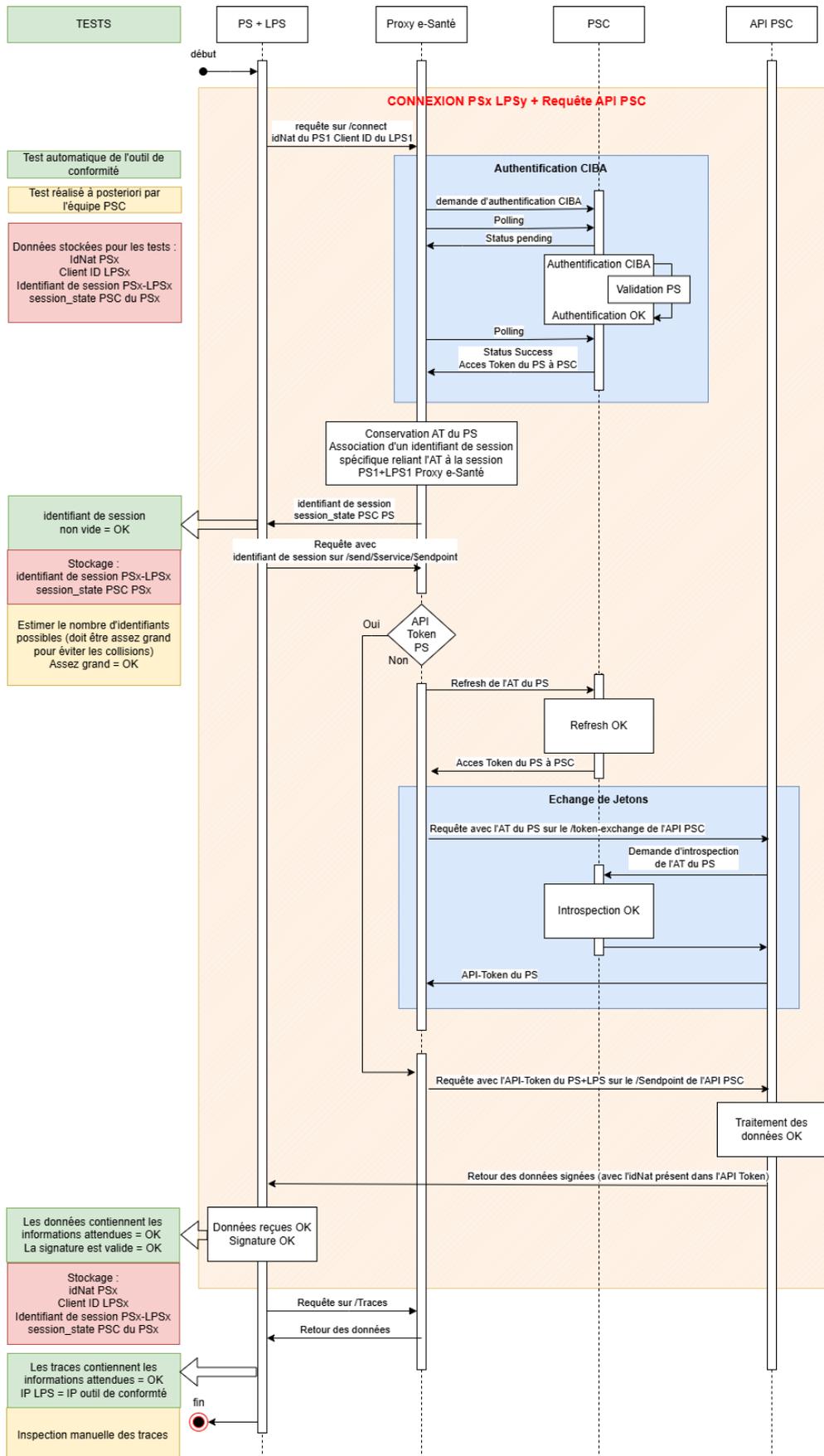
9.1. Scénario 1 : Cas Nominal - Connexion PS1 LPS1 + Requête API PSC

9.1.1. Objectif

Ce scénario de test doit :

- Simuler la connexion d'un seul PS via un LPS au Proxy e-Santé
- Effectuer une requête (demande de signature) à l'API PSC
- Effectuer une requête (récupération des traces) au Proxy e-Santé

9.1.2. Diagramme de séquence



9.1.3. Résultats attendus

1) Requête de connexion (passant) :

- Le Proxy e-Santé doit **initier une connexion** du PS1 à PSC au nom du LPS1 (cette connexion doit être en succès pour continuer, avec notamment un contrôle du certificat présenté (AC émettrice, Expiration et Révocation))
- Le Proxy e-Santé doit **retourner** :
 - un **identifiant de session unique** à la connexion « PS1+LPS1-Proxy e-Santé » (relié à l'Access Token PSC du PS au niveau du Proxy e-Santé),
 - ainsi que le « **session_state** » PSC du PS

2) Requête de signature (passant) :

- L'API PSC vérifie le jeton PSC (AT) via un appel au endpoint d'introspection de PSC
- Les données retournées au LPS1 par l'API PSC doivent contenir **les données envoyées + une signature** par l'API PSC
- Vérifier que les données signées qui transfèrent par le proxy n'ont pas été altérées

3) Requête de récupération des traces (passant) :

- Appel au endpoint /traces afin de vérifier que le proxy collecte bien les traces demandées (plusieurs formats sont supportés : text/plain, application/json, application/xml, application/zip, application/octet-stream)
- Vérifier le format de la réponse : La pièce jointe avec les traces fournit par le proxy doit contenir l'entête "Content-Disposition" à sa réponse

4) Requête de connexion (non passant) :

- Le Proxy e-Santé doit **initier une double connexion** du PS1 à PSC via le LPS1
- Le Proxy e-Santé doit **retourner** :
 - un **code 304**,

5) Requête de connexion (non passant) :

- Le Proxy e-Santé doit **initier une connexion** du PS1 à PSC via le LPS1 avec un **identifiant de LPS invalide**
 - Le Proxy e-Santé doit **retourner** :
 - un **code 404**,
- ainsi que le message « **User National ID or Software Client ID Not Found** »

6) Requête sur /send mais PS déconnecté :

- Le Proxy e-Santé doit **initier une connexion** du PS1 à PSC via le LPS1
- Le Proxy e-Santé doit déconnecter le PS
- A la réception de la requête /send, le Proxy e-Santé doit **retourner** :
 - un **code 401**,
 - ainsi que le message « **No session found** »

9.1.4. Scénario de test

- nationalId = idNat d'une e-CPS de test
- clientId = Identifiant du LPS (ans-odc-lps1-edc-bas)
- proxy_session_id = Identifiant de la session Proxy
- session_state = Identifiant de session PSC

Remarque :

Les informations « A Valoriser » (en vert) sont renseignées :

- Par l'utilisateur technique lors de la création de la session de tests. Exemple : "channel" : "\${value_authentMode}", prend l'une des deux options : **CARD, MOBILE** selon le choix de l'utilisateur.
- Par l'outil de test en utilisant les informations déjà reçues lors des requêtes précédentes. Exemple : "session_state" : "\${sessionstate}" prend la valeur reçue par la requête /connect

Nom du test	Modèle d'URL	Réponse attendue
Requête de connexion (Passant)	<p>Méthode : POST</p> <p>Requête : https://\${BASE_URL}/connect</p> <p>Json body :</p> <pre>{ "nationalId": "\${value_nationalId}", "bindingMessage": "99", "clientId": "\${value_clientId}", "channel": "\${value_authentMode}", }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/connect</p>	<ul style="list-style-type: none"> - Code : 200 - Le corps de la réponse (.json) contient : <ul style="list-style-type: none"> - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur
Requête de signature (Passant)	<p>Méthode : POST</p> <p>Requête : https://\${BASE_URL}/send/apips/ signsessiondata</p> <p>Json body :</p> <pre>{ "nationalId": "\${value_nationalId}", "clientId": "\${value_clientId}", "proxy_session_id": "\${value_proxysessionid}", "session_state": "\${sessionstate}" }</pre>	<ul style="list-style-type: none"> - Code : 200 - Le corps de la réponse (.json) contient : <ul style="list-style-type: none"> - key "nationalId" avec une valeur - key "clientId" avec une valeur - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur - key "signature" avec une valeur

	<p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre>{ "nationalId" : "899700539499", "clientID" : " ans-odc-lps1-edc-bas", "proxy_session_id" : "F24A1675730D0BD9D7191CA78592143C", "session_state" : "9ee0fd18-4823-4972-b6c3-847bbb3cbe8a" }</pre>	<p>- Les valeurs des variables (« nationalId », « clientId », « Proxy_session_id », « session_state », « signature ») sont identiques à celles envoyées dans la requête</p>
<p>Requête de récupération des traces</p> <p>(Passant)</p>	<p>Méthode : GET</p> <p>Requête : https://\${BASE_URL}/traces?start=\${value_start}&end=\${value_end}</p> <p>Format de la date : AAAA-MM-JJT00:00:00Z</p> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/traces?start=2024-10-02T09:00:00Z&end=2024-10-02T11:32:00Z</p> <p>Paramètres :</p> <ul style="list-style-type: none"> - start = début de la période de recherche choisie (Obligatoire) - end = fin de la période de recherche choisie (facultatif) 	<p>- Code : 200</p> <p>- La réponse peut être au format texte ou pièce jointe, et doit contenir les informations suivantes :</p> <ul style="list-style-type: none"> • client_id • id_nat • session_state • timestamp • source_ip • source_port • proxy_session_id • certificats <ul style="list-style-type: none"> ○ cn ○ ou • error_code • request
<p>Requête de connexion</p> <p>(Non Passant) (PS1 déjà connecté au Proxy)</p>	<p>Méthode : POST</p> <p>Requête : Envoyer deux fois la requête de connexion pour le PS1 : https://\${BASE_URL}/connect</p> <p>Json body :</p> <pre>{ "nationalId" : "\${value_nationalId}", "bindingMessage" : "99", "clientID" : "\${value_clientId}", "channel" : "\${value_authentMode}", }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/connect</p>	<p>- Code : 304</p>
<p>Requête de connexion</p> <p>(Non Passant) (Identifiant PS invalide)</p>	<p>Méthode : POST</p> <p>Requête : Envoyer une requête de connexion avec un identifiant PS invalide : https://\${BASE_URL}/connect</p> <p>Json body :</p> <pre>{ "nationalId" : "\${value_nationalId}", "bindingMessage" : "99", "clientID" : "\${value_clientId}", "channel" : "\${value_authentMode}", }</pre>	<p>- Code : 404</p> <p>- Un message : "User National ID or Software Client ID Not Found"</p>

	<pre>} En vert = A valoriser En bleu = Nom du endpoint Exemple : Json body : { "nationalId" : "\${value_nationalId}", "bindingMessage" : "99", "clientID" : "ans-odc-lps3-bas", "channel" : "MOBILE", }</pre>	
--	--	--

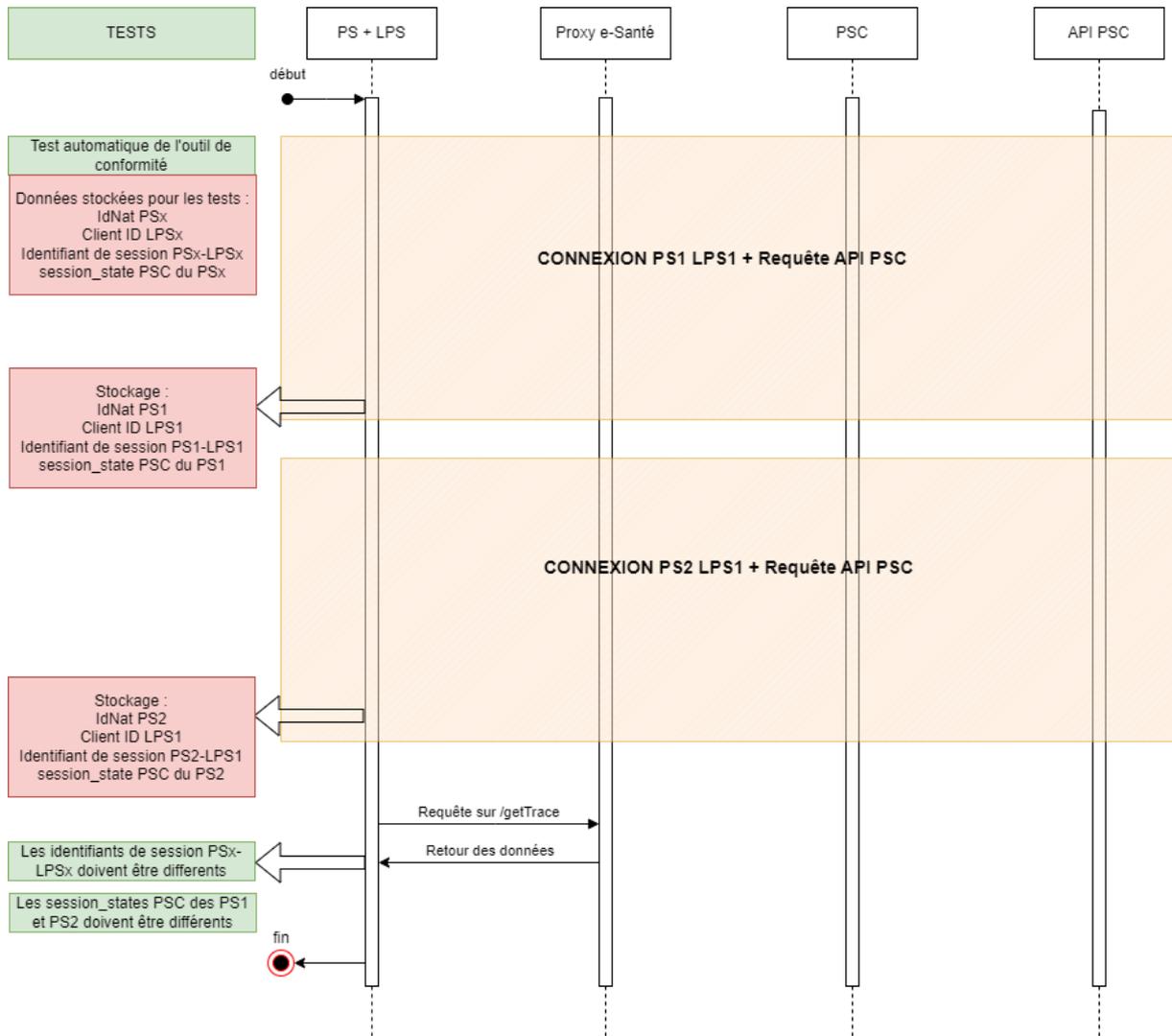
9.2. Scénario 2 : Cas Nominal - Connexion PS1 LPS1 et PS2 LP1 + Requête API PSC

9.2.1. Objectif

Ce scénario de test doit :

- Simuler la connexion du PS1 via le LPS1 au Proxy e-Santé (aucun AT n'est envoyé au Proxy e-Santé)
- Effectuer une requête (1) à l'API PSC Connectée
- Simuler la connexion du PS2 via le LPS1 au Proxy e-Santé (aucun AT n'est envoyé au Proxy e-Santé)
- Effectuer une requête (2) à l'API PSC Connectée
- Récupérer les traces correspondantes

9.2.2. Diagramme de séquence



9.2.3. Résultats attendus

- Le Proxy e-Santé doit initier une connexion du PS1 à PSC (cette connexion doit être en succès pour continuer)
- Le Proxy e-Santé doit retourner un identifiant de session unique à la connexion PS1+LPS1-Proxy e-Santé et relié à l'AT PSC du PS
- Les données retournées doivent contenir les données envoyées + un aléa, le tout signé par l'API PSC
- Le Proxy e-Santé doit initier une connexion du PS2 à PSC (cette connexion doit être en succès pour continuer)
- Le Proxy e-Santé doit retourner un identifiant de session unique à la connexion PS2+LPS1-Proxy e-Santé et relié à l'AT PSC du PS (qui doit être différent de celui du point 2)
- Les données retournées doivent contenir les données envoyées + un aléa, le tout signé par l'API PSC
- Les traces collectées (via le endpoint /traces) doivent contenir les informations attendues

9.2.4. Scénario de test

- nationalId = idNat d'une e-CPS de test

- clientId = Identifiant du LPS (ans-odc-lps1-edc-bas)
- proxy_session_id = Identifiant de la session Proxy
- session_state = Identifiant de session PSC

Nom du test	Modèle d'URL	Réponse attendue
Requête de connexion PS1 LPS1 (Passant)	<p>Méthode : POST</p> <p>Requête : https://\${BASE_URL}/connect</p> <p>Json body :</p> <pre> { "nationalId": "\${value_nationalId_PS1}", "bindingMessage": "99", "clientId": "\${value_clientId}", "channel": "\${value_authentMode}", }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/connect</p>	<ul style="list-style-type: none"> - Code : 200 - Le corps de la réponse (.json) contient : <ul style="list-style-type: none"> - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur
Requête de signature (Passant)	<p>Méthode : POST</p> <p>Requête : https://\${BASE_URL}/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre> { "nationalId": "\${value_nationalId}", "clientId": "\${value_clientId}", "proxy_session_id": "\${value_proxysessionid}", "session_state": "\${sessionstate}" }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre> { "nationalId": "899700539499", "clientId": "ans-odc-lps1-edc-bas", "proxy_session_id": "F24A1675730D0BD9D7191CA78592143C", "session_state": "9ee0fd18-4823-4972-b6c3-847bbb3cbe8a" }</pre>	<ul style="list-style-type: none"> - Code : 200 - Le corps de la réponse (.json) contient : <ul style="list-style-type: none"> - key "nationalId" avec une valeur - key "clientId" avec une valeur - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur - key "signature" avec une valeur - Les valeurs des variables (« nationalId », « clientId », « Proxy_session_id », « session_state », « signature ») sont identiques à celles envoyées dans la requête
Requête de récupération des traces (Passant)	<p>Méthode : GET</p> <p>Requête : https://\${BASE_URL}/traces?start=\${value_start}&end=\${value_end}</p> <p>Format de la date : AAAA-MM-JJT00:00:00Z</p>	<ul style="list-style-type: none"> - Code : 200 - La réponse peut être au format texte ou pièce jointe, et doit contenir les informations suivantes : <ul style="list-style-type: none"> • client_id • id_nat

	<p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/traces?start=2024-10-02T09:00:00Z&end=2024-10-02T11:32:00Z</p> <p>Paramètres :</p> <ul style="list-style-type: none"> - start = début de la période de recherche choisie (Obligatoire) - end = fin de la période de recherche choisie (facultatif) 	<ul style="list-style-type: none"> • session_state • timestamp • source_ip • source_port • proxy_session_id • certificats <ul style="list-style-type: none"> ○ cn ○ ou • error_code • request
<p>Requête de connexion PS2 LPS1 (Passant)</p>	<p>Méthode : POST</p> <p>Requête : https://\${BASE_URL}/connect</p> <p>Json body :</p> <pre>{ "nationalId": "\${value_nationalId_PS2}", "bindingMessage": "99", "clientId": "\${value_clientId}", "channel": "\${value_authentMode}", }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/connect</p>	<ul style="list-style-type: none"> - Code : 200 - Le corps de la réponse (.json) contient : <ul style="list-style-type: none"> - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur
<p>Requête de signature (Passant)</p>	<p>Méthode : POST</p> <p>Requête : https://\${BASE_URL}/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre>{ "nationalId": "\${value_nationalId}", "clientId": "\${value_clientId}", "proxy_session_id": "\${value_proxysessionid}", "session_state": "\${sessionstate}" }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre>{ "nationalId": "899700539500", "clientId": "ans-odc-lps1-edc-bas", "proxy_session_id": "F24A1675730D0BD9D7191CA78592143C", "session_state": "9ee0fd18-4823-4972-b6c3-847bbb3cbe8a" }</pre>	<ul style="list-style-type: none"> - Code : 200 - Le corps de la réponse (.json) contient : <ul style="list-style-type: none"> - key "nationalId" avec une valeur - key "clientId" avec une valeur - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur - key "signature" avec une valeur - Les valeurs des variables (« nationalId », « clientId », « Proxy_session_id », « session_state », « signature ») sont identiques à celles envoyées dans la requête
<p>Requête de récupération des traces (Passant)</p>	<p>Méthode : GET</p> <p>Requête : https://\${BASE_URL}/traces?start=\${value_start}&end=\${value_end}</p> <p>Format de la date : AAAA-MM-JJT00:00:00Z</p>	<ul style="list-style-type: none"> - Code : 200 - La réponse peut être au format texte ou pièce jointe, et doit contenir les informations suivantes : <ul style="list-style-type: none"> • client_id

	<p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/traces?start=2024-10-02T09:00:00Z&end=2024-10-02T11:40:00Z</p> <p>Paramètres :</p> <ul style="list-style-type: none"> - start = début de la période de recherche choisie (Obligatoire) - end = fin de la période de recherche choisie (facultatif) 	<ul style="list-style-type: none"> • id_nat • session_state • timestamp • source_ip • source_port • proxy_session_id • certificats <ul style="list-style-type: none"> ○ cn ○ ou • error_code <p>request</p>
<p>Déconnexion PS1 & PS2</p>	<p>Méthode : DELETE</p> <p>Requête : https://\${BASE_URL}/disconnect</p> <p>Http header : Cokkie : proxy_session_id=\${proxy_session_id_value}</p> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/disconnect</p>	<p>- Code : 200</p> <p>- 'proxy_session_id' de PS1 et différent que celui de PS2</p> <p>- 'session_state' value de PS1 et différent que celui de PS2</p>

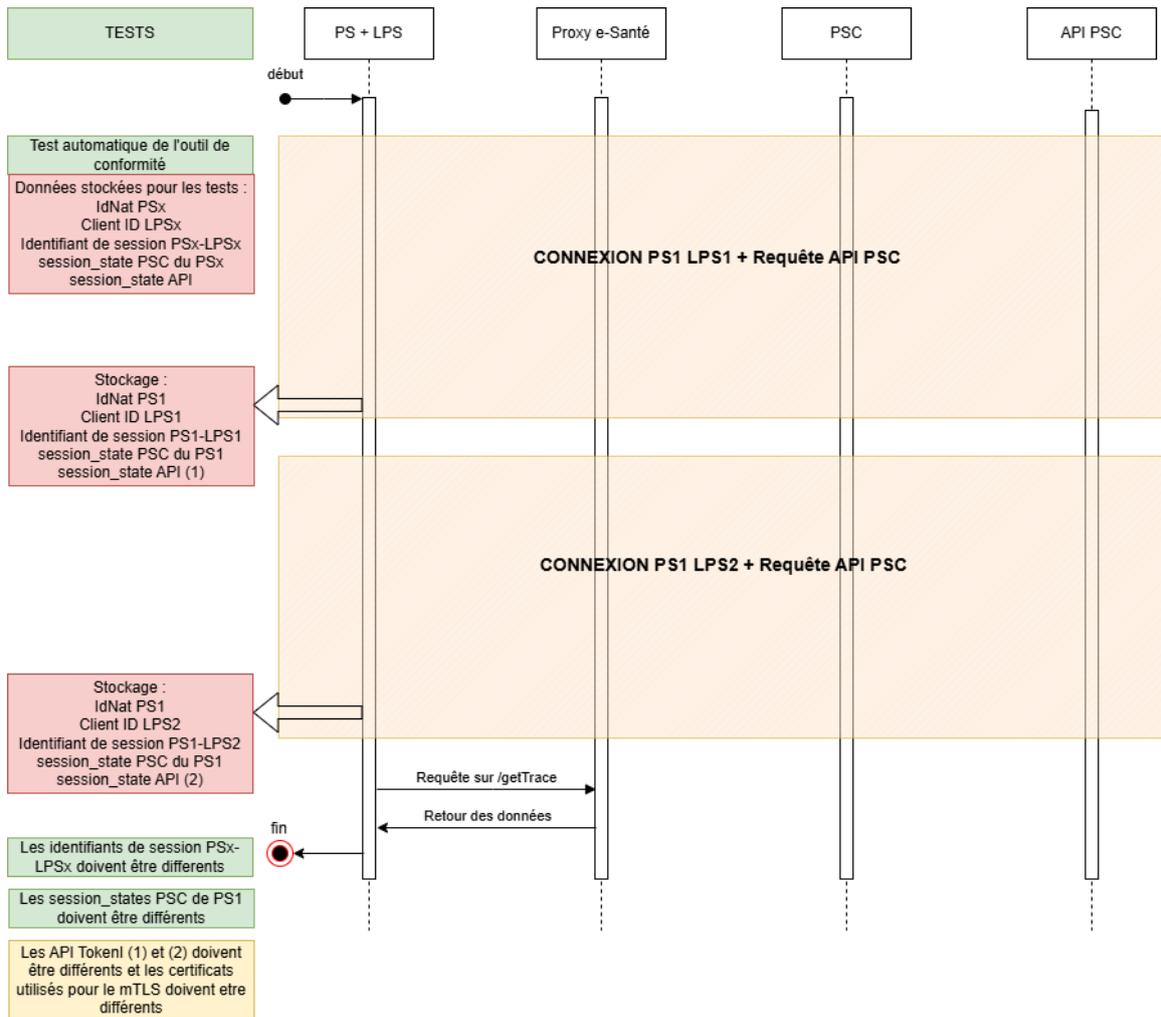
9.3. Scénario 3 : Cas Nominal - Connexion PS1 LPS1 et PS1 LP2 + Requête API PSC

9.3.1. Objectif

Ce scénario de test doit :

- Simuler la connexion du PS1 via le LPS1 au Proxy e-Santé (aucun AT n'est envoyé au Proxy e-Santé)
- Effectuer une requête (1) à l'API PSC Connectée
- Simuler la connexion du PS1 via le LPS2 au Proxy e-Santé (aucun AT n'est envoyé au Proxy e-Santé)
- Effectuer une requête (2) à l'API PSC Connectée
- Récupérer les traces correspondantes

9.3.2. Diagramme de séquence



9.3.3. Résultats attendus

- Le Proxy e-Santé doit initier une connexion du PS1 à PSC (cette connexion doit être en succès pour continuer)
- Le Proxy e-Santé doit retourner un identifiant de session unique à la connexion PS1+LPS1-Proxy e-Santé et relié à l'AT PSC du PS
- Les données retournées doivent contenir les données envoyées + un aléa, le tout signé par l'API PSC
- Le Proxy e-Santé doit initier une connexion du PS1 à PSC (cette connexion doit être en succès pour continuer)
- Le Proxy e-Santé doit retourner un identifiant de session unique à la connexion PS1+LPS2-Proxy e-Santé et relié à l'AT PSC du PS (qui doit être différent de celui du point 2)
- Les données retournées doivent contenir les données envoyées + un aléa, le tout signé par l'API PSC
- Les traces collectées (via le endpoint /traces) doivent contenir les informations attendues

9.3.4. Scénario de test

- nationalId = idNat d'une e-CPS de test
- clientId = Identifiant du LPS ("ans-odc-lps1-edc-bas" ou "ans-odc-lps2-edc-bas")
- proxy_session_id = Identifiant de la session Proxy

- session_state = Identifiant de session PSC

Nom du test	Modèle d'URL	Réponse attendue
Requête de connexion PS1 LPS1 (Passant)	<p>Méthode : POST</p> <p>Requête : https://\${BASE_URL}/connect</p> <p>Json body :</p> <pre>{ "nationalId": "\${value_nationalId_PS1}", "bindingMessage": "99", "clientId": "\${value_clientId_lps1}", "channel": "\${value_authentMode}", }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/connect</p>	<p>- Code : 200</p> <p>- Le corps de la réponse (.json) contient :</p> <ul style="list-style-type: none"> - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur
Requête de signature (Passant)	<p>Méthode : POST</p> <p>Requête : https://\${BASE_URL}/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre>{ "nationalId": "\${value_nationalId_PS1}", "clientId": "\${value_clientId_lps1}", "proxy_session_id": "\${value_proxysessionId}", "session_state": "\${sessionstate}" }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre>{ "nationalId": "899700539499", "clientId": "ans-odc-lps1-edc-bas", "proxy_session_id": "F24A1675730D0BD9D7191CA78592143C", "session_state": "9ee0fd18-4823-4972-b6c3-847bbb3cbe8a" }</pre>	<p>- Code : 200</p> <p>- Le corps de la réponse (.json) contient :</p> <ul style="list-style-type: none"> - key "nationalId" avec une valeur - key "clientId" avec une valeur - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur - key "signature" avec une valeur <p>- Les valeurs des variables (« nationalId », « clientId », « Proxy_session_id », « session_state », « signature ») sont identiques à celles envoyées dans la requête</p>
Requête de récupération des traces (Passant)	<p>Méthode : GET</p> <p>Requête : https://\${BASE_URL}/traces?start=\${value_start}&end=\${value_end}</p> <p>Format de la date : AAAA-MM-JJT00:00:00Z</p> <p>En vert = A valoriser En bleu = Nom du endpoint</p>	<p>- Code : 200</p> <p>- La réponse peut être au format texte ou pièce jointe, et doit contenir les informations suivantes :</p> <ul style="list-style-type: none"> • client_id • id_nat • session_state • timestamp • source_ip

	<p>Exemple : https://10.3.8.165:8080/traces?start=2024-10-02T09:00:00Z&end=2024-10-02T11:32:00Z</p> <p>Paramètres :</p> <ul style="list-style-type: none"> - start = début de la période de recherche choisie (Obligatoire) - end = fin de la période de recherche choisie (facultatif) 	<ul style="list-style-type: none"> • source_port • proxy_session_id • certificats <ul style="list-style-type: none"> ○ cn ○ ou • error_code • request
<p>Requête de connexion PS1 LPS2 (Passant)</p>	<p>Méthode : POST</p> <p>Requête : https://10.3.8.165:8080/connect</p> <p>Json body :</p> <pre> { "nationalId": "\${value_nationalId_PS1}", "bindingMessage": "99", "clientId": "\${value_clientId_LPS2}", "channel": "\${value_authentMode}", }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/connect</p>	<p>- Code : 200</p> <p>- Le corps de la réponse (.json) contient :</p> <ul style="list-style-type: none"> - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur
<p>Requête de signature (Passant)</p>	<p>Méthode : POST</p> <p>Requête : https://10.3.8.165:8080/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre> { "nationalId": "\${value_nationalId_PS1}", "clientId": "\${value_clientId_IPS2}", "proxy_session_id": "\${value_proxysessionid}", "session_state": "\${sessionstate}" }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre> { "nationalId": "899700539500", "clientId": "ans-odc-lps2-edc-bas", "proxy_session_id": "dc3d6338-a841-4ccc-9a20-1774e09185bc", "session_state": "b6e716e1-00b7-41ce-a62d-0f653cd57471" }</pre>	<p>- Code : 200</p> <p>- Le corps de la réponse (.json) contient :</p> <ul style="list-style-type: none"> - key "nationalId" avec une valeur - key "clientId" avec une valeur - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur - key "signature" avec une valeur <p>- Les valeurs des variables (« nationalId », « clientId », « Proxy_session_id », « session_state », « signature ») sont identiques à celles envoyées dans la requête</p>
<p>Requête de récupération des traces (Passant)</p>	<p>Méthode : GET</p> <p>Requête : https://10.3.8.165:8080/traces?start=\${value_start}&end=\${value_end}</p> <p>Format de la date : AAAA-MM-JJT00:00:00Z</p> <p>En vert = A valoriser En bleu = Nom du endpoint</p>	<p>- Code : 200</p> <p>- La réponse peut être au format texte ou pièce jointe, et doit contenir les informations suivantes :</p> <ul style="list-style-type: none"> • client_id • id_nat • session_state • timestamp

	<p>Exemple : https://10.3.8.165:8080/traces?start=2024-10-02T09:00:00Z&end=2024-10-02T11:40:00Z</p> <p>Paramètres :</p> <ul style="list-style-type: none"> - start = début de la période de recherche choisie (Obligatoire) - end = fin de la période de recherche choisie (facultatif) 	<ul style="list-style-type: none"> • source_ip • source_port • proxy_session_id • certificats <ul style="list-style-type: none"> ○ cn ○ ou • error_code <p>request</p>
<p>Déconnexion PS1 LPS1 & PS1 LPS2</p>	<p>Méthode : DELETE</p> <p>Requête : https://\${BASE_URL}/disconnect</p> <p>Http header : Cokkie : proxy_session_id=\${proxy_session_id}</p> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/disconnect</p>	<p>- Code : 200</p> <p>- 'proxy_session_id' de PS1 LPS1 et différent que celui de PS1 LPS2</p> <p>- 'session_state' value de PS1 LPS1 et différent que celui de PS1 LPS2</p>

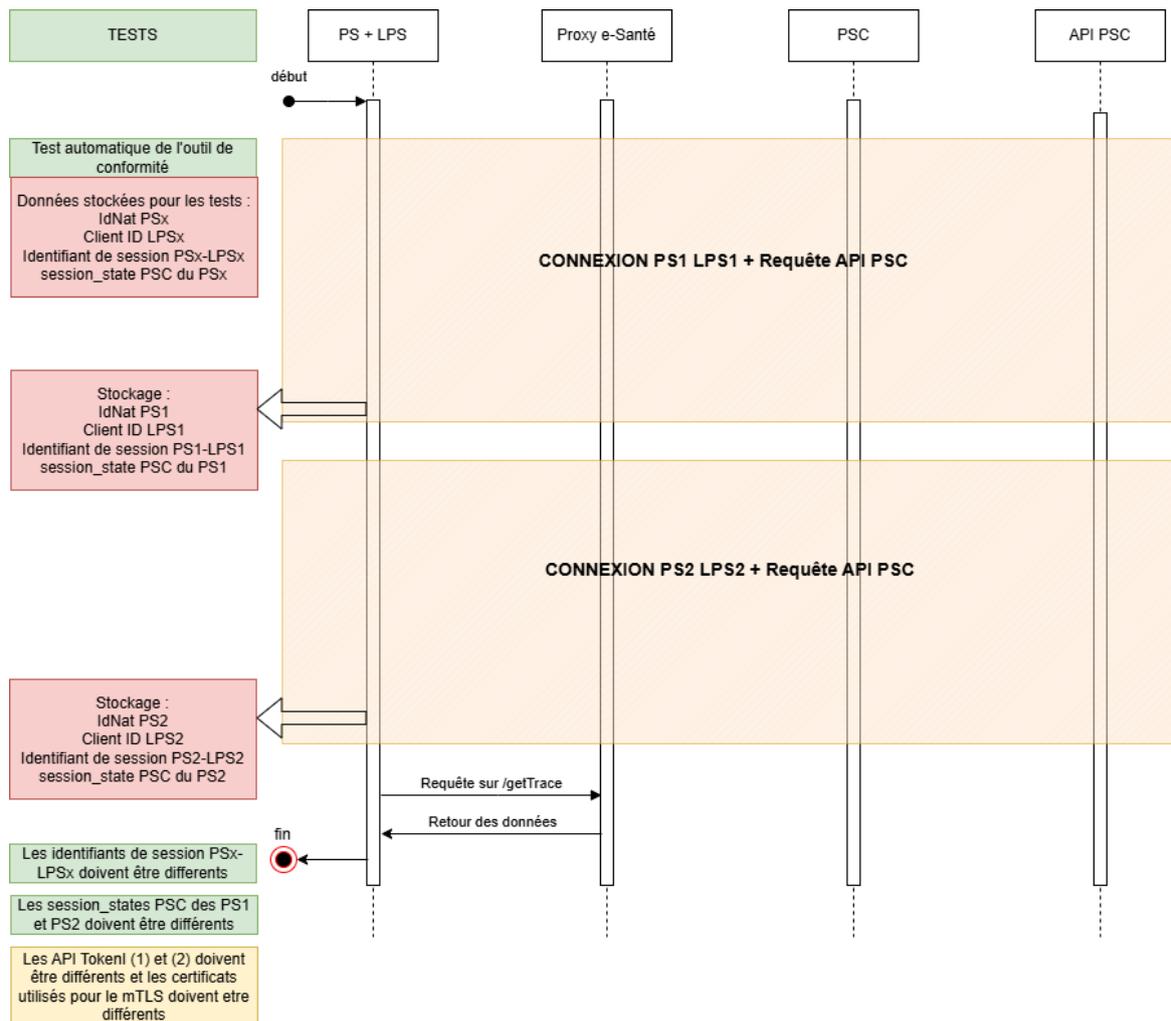
9.4. Scénario 4 : Cas Nominal - Connexion PS1 LPS1 et PS2 LP2 + Requête API PSC

9.4.1. Objectif

Ce scénario de test doit :

- Simuler la connexion du PS1 via le LPS1 au Proxy e-Santé (aucun AT n'est envoyé au Proxy e-Santé)
- Effectuer une requête (1) à l'API PSC Connectée
- Simuler la connexion du PS2 via le LPS2 au Proxy e-Santé (aucun AT n'est envoyé au Proxy e-Santé)
- Effectuer une requête (2) à l'API PSC Connectée
- Récupérer les traces correspondantes

9.4.2. Diagramme de séquence



9.4.3. Résultats attendus

- Le Proxy e-Santé doit initier une connexion du PS1 à PSC (cette connexion doit être en succès pour continuer)
- Le Proxy e-Santé doit retourner un identifiant de session unique à la connexion PS1+LPS1-Proxy e-Santé et relié à l'AT PSC du PS
- Les données retournées doivent contenir les données envoyées + un aléa, le tout signé par l'API PSC
- Le Proxy e-Santé doit initier une connexion du PS2 à PSC (cette connexion doit être en succès pour continuer)
- Le Proxy e-Santé doit retourner un identifiant de session unique à la connexion PS2+LPS2-Proxy e-Santé et relié à l'AT PSC du PS (qui doit être différent de celui du point 2)
- Les données retournées doivent contenir les données envoyées + un aléa, le tout signé par l'API PSC
- Les traces collectées (via le endpoint /traces) doivent contenir les informations attendues

9.4.4. Scénario de test

- nationalId = idNat d'une e-CPS de test
- clientId = Identifiant du LPS ("ans-odc-lps1-edc-bas" ou "ans-odc-lps2-edc-bas")
- proxy_session_id = Identifiant de la session Proxy

- session_state = Identifiant de session PSC

Nom du test	Modèle d'URL	Réponse attendue
Requête de connexion PS1 LPS1 (Passant)	<p>Méthode : POST</p> <p>Requête : https://\${BASE_URL}/connect</p> <p>Json body :</p> <pre> { "nationalId": "\${value_nationalId_PS1}", "bindingMessage": "99", "clientId": "\${value_clientId_lps1}", "channel": "\${value_authentMode}", }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/connect</p>	<p>- Code : 200</p> <p>- Le corps de la réponse (.json) contient :</p> <ul style="list-style-type: none"> - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur
Requête de signature (Passant)	<p>Méthode : POST</p> <p>Requête : https://\${BASE_URL}/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre> { "nationalId": "\${value_nationalId_PS1}", "clientId": "\${value_clientId_lps1}", "proxy_session_id": "\${value_proxysessionid}", "session_state": "\${sessionstate}" }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre> { "nationalId": "899700539499", "clientId": "ans-odc-lps1-edc-bas", "proxy_session_id": "F24A1675730D0BD9D7191CA78592143C", "session_state": "9ee0fd18-4823-4972-b6c3-847bbb3cbe8a" }</pre>	<p>- Code : 200</p> <p>- Le corps de la réponse (.json) contient :</p> <ul style="list-style-type: none"> - key "nationalId" avec une valeur - key "clientId" avec une valeur - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur - key "signature" avec une valeur <p>- Les valeurs des variables (« nationalId », « clientId », « Proxy_session_id », « session_state », « signature ») sont identiques à celles envoyées dans la requête</p>
Requête de récupération des traces (Passant)	<p>Méthode : GET</p> <p>Requête : https://\${BASE_URL}/traces?start=\${value_start}&end=\${value_end}</p> <p>Format de la date : AAAA-MM-JJT00:00:00Z</p> <p>En vert = A valoriser En bleu = Nom du endpoint</p>	<p>- Code : 200</p> <p>- La réponse peut être au format texte ou pièce jointe, et doit contenir les informations suivantes :</p> <ul style="list-style-type: none"> • client_id • id_nat • session_state • timestamp • source_ip

	<p>Exemple : https://10.3.8.165:8080/traces?start=2024-10-02T09:00:00Z&end=2024-10-02T11:32:00Z</p> <p>Paramètres :</p> <ul style="list-style-type: none"> - start = début de la période de recherche choisie (Obligatoire) - end = fin de la période de recherche choisie (facultatif) 	<ul style="list-style-type: none"> • source_port • proxy_session_id • certificats <ul style="list-style-type: none"> ○ cn ○ ou • error_code • request
<p>Requête de connexion PS2 LPS2 (Passant)</p>	<p>Méthode : POST</p> <p>Requête : https://10.3.8.165:8080/connect</p> <p>Json body :</p> <pre> { "nationalId" : "\${value_nationalId_PS2}", "bindingMessage" : "99", "clientID" : "\${value_clientId_LPS2}", "channel" : "\${value_authentMode}", }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/connect</p>	<p>- Code : 200</p> <p>- Le corps de la réponse (.json) contient :</p> <ul style="list-style-type: none"> - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur
<p>Requête de signature (Passant)</p>	<p>Méthode : POST</p> <p>Requête : https://10.3.8.165:8080/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre> { "nationalId" : "\${value_nationalId_PS2}", "clientID" : "\${value_clientId_IPS2}", "proxy_session_id" : "\${value_proxysessionid}", "session_state" : "\${sessionstate}" }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre> { "nationalId" : "899700539500", "clientID" : "ans-odc-lps2-edc-bas", "proxy_session_id" : "dc3d6338-a841-4ccc-9a20-1774e09185bc", "session_state" : "b6e716e1-00b7-41ce-a62d-0f653cd57471" }</pre>	<p>- Code : 200</p> <p>- Le corps de la réponse (.json) contient :</p> <ul style="list-style-type: none"> - key "nationalId" avec une valeur - key "clientId" avec une valeur - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur - key "signature" avec une valeur <p>- Les valeurs des variables (« nationalId », « clientId », « Proxy_session_id », « session_state », « signature ») sont identiques à celles envoyées dans la requête</p>
<p>Requête de récupération des traces (Passant)</p>	<p>Méthode : GET</p> <p>Requête : https://10.3.8.165:8080/traces?start=\${value_start}&end=\${value_end}</p> <p>Format de la date : AAAA-MM-JJT00:00:00Z</p> <p>En vert = A valoriser En bleu = Nom du endpoint</p>	<p>- Code : 200</p> <p>- La réponse peut être au format texte ou pièce jointe, et doit contenir les informations suivantes :</p> <ul style="list-style-type: none"> • client_id • id_nat • session_state • timestamp

	<p>Exemple : https://10.3.8.165:8080/traces?start=2024-10-02T09:00:00Z&end=2024-10-02T11:40:00Z</p> <p>Paramètres :</p> <ul style="list-style-type: none"> - start = début de la période de recherche choisie (Obligatoire) - end = fin de la période de recherche choisie (facultatif) 	<ul style="list-style-type: none"> • source_ip • source_port • proxy_session_id • certificats <ul style="list-style-type: none"> ○ cn ○ ou • error_code <p>request</p>
<p>Déconnexion PS1 LPS1 & PS2 LPS2</p>	<p>Méthode : DELETE</p> <p>Requête : https://\${BASE_URL}/disconnect</p> <p>Http header : Cokkie : proxy_session_id=\${proxy_session_id}</p> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/disconnect</p>	<p>- Code : 200</p> <p>- 'proxy_session_id' de PS1 LPS1 et différent que celui de PS2 LPS2</p> <p>- 'session_state' value de PS1 LPS1 et différent que celui de PS2 LPS2</p>

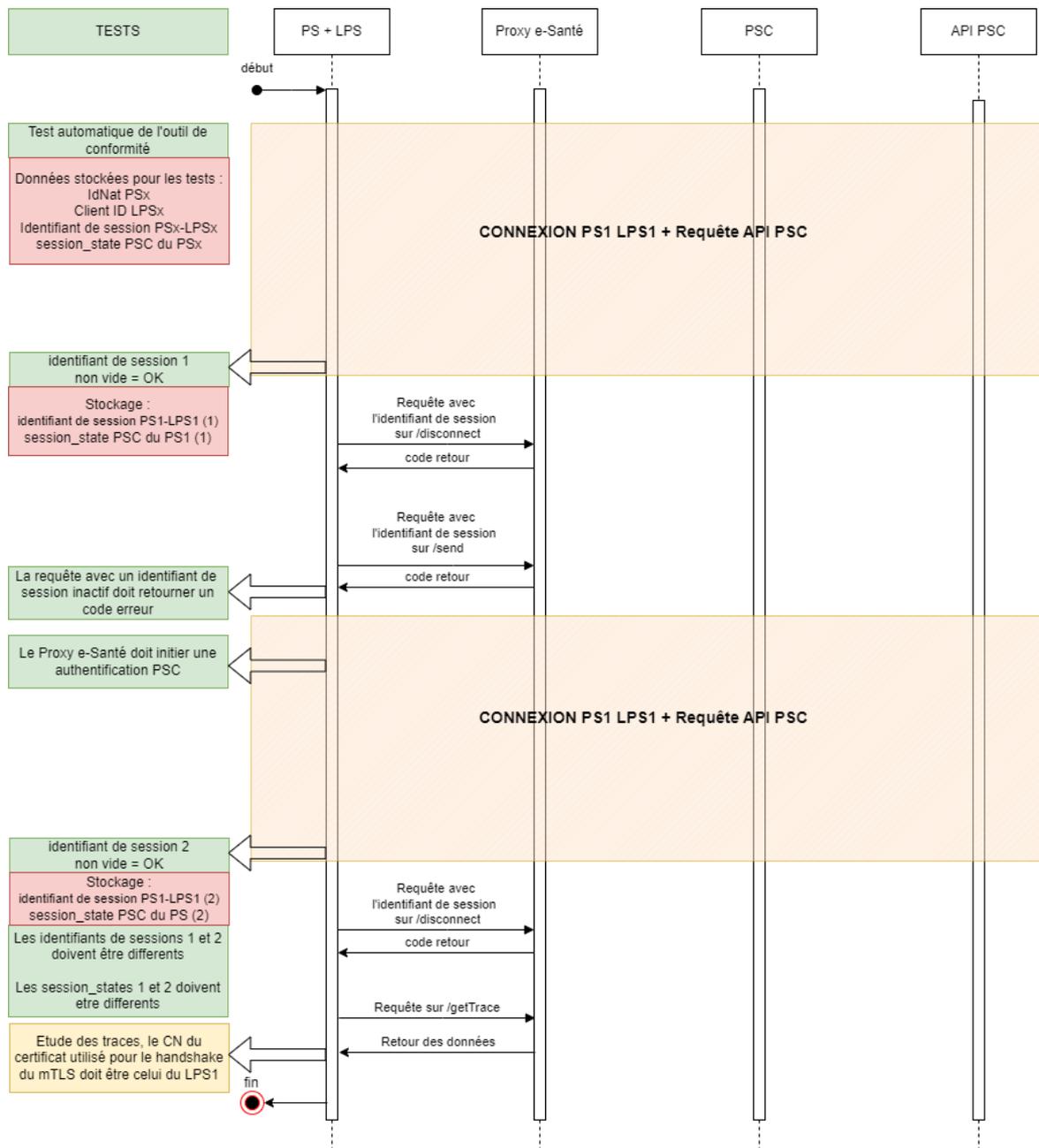
9.5. Scénario 5 : Cas Nominal - Connexion PS1 LPS1 + Requête API PSC + déconnexion

9.5.1. Objectif

Ce scénario de test doit :

- Simuler la connexion du PS1 via le LPS1 au Proxy e-Santé (aucun AT n'est envoyé au Proxy e-Santé)
- Effectuer une requête (1) à l'API PSC Connectée
- Déconnecter le PS1 sur le endpoint /disconnect (désactiver de la session PS1+LPS1)
- Effectuer une requête (1) à l'API PSC Connectée (identique à celle du point 2)
- Simuler la connexion du PS1 via le LPS1 au Proxy e-Santé (aucun AT n'est envoyé au Proxy e-Santé)
- Effectuer une requête (2) à l'API PSC Connectée
- Récupérer les traces correspondantes

9.5.2. Diagramme de séquence



9.5.3. Résultats attendus

- Le Proxy e-Santé doit initier une connexion du PS1 à PSC (cette connexion doit être en succès pour continuer)
- Le Proxy e-Santé doit retourner un identifiant de session unique à la connexion PS1+LPS1-Proxy e-Santé et relié à l'AT PSC du PS
- Les données retournées doivent contenir les données envoyées + un aléa, le tout signé par l'API PSC
- Le PS1 + LPS1 est déconnecté du Proxy e-Santé
- L'identifiant de session doit être inactif
- La requête envoyée sur le endpoint /send avec l'id de session doit retourner un code erreur
- Le Proxy e-Santé doit initier une connexion du PS1 à PSC (cette connexion doit être en succès pour continuer)

- Le Proxy e-Santé doit retourner un identifiant de session unique à la connexion PS1+LPS1-Proxy e-Santé et relié à l'AT PSC du PS (qui doit être différent de celui du point 2)
- Les données retournées doivent contenir les données envoyées + un aléa, le tout signé par l'API PSC
- 1Les traces collectées (via le endpoint /traces) doivent contenir les informations attendues
- 1Les identifiants de session PS1 + LPS des points 2 et 8 doivent être différents

9.5.4. Scénario de test

- nationalId = idNat d'une e-CPS de test
- clientId = Identifiant du LPS (ans-odc-lps1-edc-bas)
- proxy_session_id = Identifiant de la session Proxy
- session_state = Identifiant de session PSC

Nom du test	Modèle d'URL	Réponse attendue
Requête de connexion PS1 LPS1 (Passant)	<p>Méthode : POST</p> <p>Requête : https://\${BASE_URL}/connect</p> <p>Json body :</p> <pre>{ "nationalId": "\${value_nationalId_PS1}", "bindingMessage": "99", "clientId": "\${value_clientId_lps1}", "channel": "\${value_authentMode}", }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/connect</p>	<p>- Code : 200</p> <p>- Le corps de la réponse (.json) contient :</p> <ul style="list-style-type: none"> - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur
Requête de signature (Passant)	<p>Méthode : POST</p> <p>Requête : https://\${BASE_URL}/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre>{ "nationalId": "\${value_nationalId_PS1}", "clientId": "\${value_clientId_lps1}", "proxy_session_id": "\${value_proxysessionid}", "session_state": "\${sessionstate}" }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre>{ "nationalId": "899700539499", "clientId": " ans-odc-lps1-edc-bas",</pre>	<p>- Code : 200</p> <p>- Le corps de la réponse (.json) contient :</p> <ul style="list-style-type: none"> - key "nationalId" avec une valeur - key "clientId" avec une valeur - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur - key "signature" avec une valeur <p>- Les valeurs des variables (« nationalId », « clientId », « Proxy_session_id », « session_state », « signature ») sont identiques à celles envoyées dans la requête</p>

	<pre>"proxy_session_id" : "F24A1675730D0BD9D7191CA78592143C", "session_state" : "9ee0fd18-4823-4972-b6c3-847bbb3cbe8a" }</pre>	
Déconnexion PS1 LPS1	<p>Méthode : DELETE</p> <p>Requête : https://{{BASE_URL}}/disconnect</p> <p>Http header : Cokkie : proxy_session_id={{proxy_session_id}}</p> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/disconnect</p>	<p>- Code : 200</p>
Requête de signature (Non Passant)	<p>Méthode : POST</p> <p>Requête : https://{{BASE_URL}}/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre>{ "nationalId" : "{{value_nationalId_PS1}}", "clientID" : "{{value_clientId_ips1}}", "proxy_session_id" : "{{value_proxysessionid}}", "session_state" : "{{sessionstate}}" }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre>{ "nationalId" : "899700539499", "clientID" : " ans-odc-lps1-edc-bas", "proxy_session_id" : "F24A1675730D0BD9D7191CA78592143C", "session_state" : "9ee0fd18-4823-4972-b6c3-847bbb3cbe8a" }</pre>	<p>- Code : 401</p> <p>- Un message : "User National ID or Software Client ID Not Found"</p>
Requête de récupération des traces (Passant)	<p>Méthode : GET</p> <p>Requête : https://{{BASE_URL}}/traces?start={{value_start}}&end={{value_end}}</p> <p>Format de la date : AAAA-MM-JJT00:00:00Z</p> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/traces?start=2024-10-02T09:00:00Z&end=2024-10-02T11:32:00Z</p> <p>Paramètres :</p> <ul style="list-style-type: none"> - start = début de la période de recherche choisie (Obligatoire) - end = fin de la période de recherche choisie (facultatif) 	<p>- Code : 200</p> <p>- La réponse peut être au format texte ou pièce jointe, et doit contenir les informations suivantes :</p> <ul style="list-style-type: none"> • client_id • id_nat • session_state • timestamp • source_ip • source_port • proxy_session_id • certificats <ul style="list-style-type: none"> ○ cn ○ ou • error_code • request

<p>Requête de connexion PS1 LPS1 (Passant)</p>	<p>Méthode : POST</p> <p>Requête : https://\${BASE_URL}/connect</p> <p>Json body :</p> <pre>{ "nationalId" : "\${value_nationalId_PS1}", "bindingMessage" : "99", "clientId" : "\${value_clientId_LPS1}", "channel" : "\${value_authentMode}", }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/connect</p>	<ul style="list-style-type: none"> - Code : 200 - Le corps de la réponse (.json) contient : <ul style="list-style-type: none"> - key "proxy_session_id" avec une valeur. Cette valeur est différente que la première requête de « connexion » - key "session_state" avec une valeur. Cette valeur est différente que la première requête de « connexion »
<p>Requête de signature (Passant)</p>	<p>Méthode : POST</p> <p>Requête : https://\${BASE_URL}/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre>{ "nationalId" : "\${value_nationalId_PS1}", "clientId" : "\${value_clientId_IPS1}", "proxy_session_id" : "\${value_proxysessionid}", "session_state" : "\${sessionstate}" }</pre> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/send/apipsc/signsessiondata</p> <p>Json body :</p> <pre>{ "nationalId" : "899700539500", "clientId" : " ans-odc-lps1-edc-bas", "proxy_session_id" : "dc3d6338-a841-4ccc-9a20-1774e09185bc", "session_state" : "b6e716e1-00b7-41ce-a62d-0f653cd57471" }</pre>	<ul style="list-style-type: none"> - Code : 200 - Le corps de la réponse (.json) contient : <ul style="list-style-type: none"> - key "nationalId" avec une valeur - key "clientId" avec une valeur - key "proxy_session_id" avec une valeur - key "session_state" avec une valeur - key "signature" avec une valeur - Les valeurs des variables (« nationalId », « clientId », « Proxy_session_id », « session_state », « signature ») sont identiques à celles envoyées dans la requête
<p>Déconnexion PS1 LPS1</p>	<p>Méthode : DELETE</p> <p>Requête : https://\${BASE_URL}/disconnect</p> <p>Http header : Cokkie : proxy_session_id=\${proxy_session_id}</p> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/disconnect</p>	<ul style="list-style-type: none"> - Code : 200 - 'proxy_session_id' valeurs sont différents entre les 2 sessions - 'session_state' valeurs sont différent entre les 2 sessions
<p>Requête de récupération des traces</p>	<p>Méthode : GET</p> <p>Requête : https://\${BASE_URL}/traces?start=\${value_start}&end=\${value_end}</p>	<ul style="list-style-type: none"> - Code : 200 - La réponse peut être au format texte ou pièce jointe, et doit

<p>(Passant)</p>	<p>Format de la date : AAAA-MM-JJT00:00:00Z</p> <p>En vert = A valoriser En bleu = Nom du endpoint</p> <p>Exemple : https://10.3.8.165:8080/traces?start=2024-10-02T09:00:00Z&end=2024-10-02T11:32:00Z</p> <p>Paramètres :</p> <ul style="list-style-type: none">- start = début de la période de recherche choisie (Obligatoire)- end = fin de la période de recherche choisie (facultatif)	<p>contenir les informations suivantes :</p> <ul style="list-style-type: none">• client_id• id_nat• session_state• timestamp• source_ip• source_port• proxy_session_id• certificats<ul style="list-style-type: none">○ cn○ ou• error_code• request
-------------------------	---	--