Cible de sécurité CSPN

Odisia Broker & Desktop

Version 1.1

Technopole de l'Aube en Champagne – CS 90601 – 10901 Troyes Cedex 9
Tél. : +33 (0)3 25 43 90 78 – Fax : +33 (0)9 81 40 30 08 – www.lex-persona.com
SARL au capital de 30 000 Euros – R.C.S. Troyes 480 622 257 – Code APE 5829C – TVA FR01480622257

Etat de de validation

	Vérifié par	Approuvé par
Nom	Benoit Schweblin	Julien Pasquier
Fonction	P.O Odisia	Directeur technique

Historique des révisions

Version	Date	Auteurs	Commentaires
1.0	08/06/2020	Oppida	Version initiale
1.1	04/09/2020	Oppida	Mise à jour suite aux commentaires de l'ANSSI

Table des matières

1 Product identification

Editeur	Lex Persona
Lien vers l'éditeur	www.lex-persona.com
Nom commercial du produit	ODISIA Broker & Odisia Desktop
Version évaluée	Odisia Broker 1.0.15 Odisia Desktop 1.3.7
Catégorie de produit	API de signature numérique

2 Versions de composants

Apache Tomcat	9.0.35
Java SE	8u251

3 Références

[#api-xxxxxxxx] Odisia Broker API documentation – version du 11/04/19. (la fonction est désignée après le #api dans la référence).

[INSTALL] Odisia Broker API installation – version du 11/04/19.

4 Argumentaire (Description) du produit

4.1 Description générale du produit

Le produit soumis à évaluation est ***Odisia Broker utilisé avec le produit Odisia Desktop***. ***Odisia Broker*** est, un webservice REST permettant l'intermédiation entre une application métier qui nécessite une signature électronique et un signataire utilisateur de l'application métier. ***Odisia Desktop*** est une application cliente qui permet de réaliser la signature électronique depuis le poste de l'utilisateur final.

Odisia Broker et ***Odisia Desktop*** s'inscrivent dans une solution plus globale développée par LEX Persona et intitulée ***Odisia Entreprise***.

Odisia Enterprise propose une solution adaptée aux applications Web qui nécessitent la mise en œuvre d'une signature électronique à l'aide d'un certificat sur support cryptographique.
Cette solution permet de s'affranchir des contraintes liées aux applets Java, en proposant une solution de signature indépendante du navigateur Web de l'utilisateur, téléchargée préalablement ou lors de la première signature.

Odisia Enterprise propose une architecture de signature électronique ouverte et novatrice, qui découpe le processus de signature en tâches indépendantes mais organisées de manière sécurisée, ce qui permet de simplifier et fiabiliser la signature sur le poste de travail avec tout type de navigateur, sans nécessiter d'applet Java.

Les capacités de génération de signature de ***Odisia Entreprise*** sont les suivantes :

- avancées ou qualifiées conformes au règlement eIDAS ;
- conformes au référentiel général de sécurité de l'administration française ;
- conformes au format PES V2 de la DGFIP ;
- conformes aux formats requis par l'ACPR pour la signature des remises réglementaires des domaines banque et assurance ;
- conformes au format INTEROP de la Banque de France.

Les fonctionnalités de ***Odisia Entreprise*** :

- Sont compatibles avec toutes versions récentes de Edge, Chrome, Firefox, Internet Explorer, Safari et Opéra
- Utilisent le format de certificats X.509 V3
- Permettent un horodatage conforme à la RFC 3161
- Supportent des porte-clés au standard PKCS#12 et sur dispositifs cryptographiques conformes au standard PKCS#11, MS CAPIet Keychain
- Permettent l'utilisation de plusieurs formats de signatures : XAdES (ETSI TS 101 903), CAdES (ETSI TS 101 733), PAdES (ETSI TS 102 778-2 & -3)
- Prennent en charge des types de signature enveloppante, enveloppée, détachée
- Offrent Possibilité de compléter et de valider la chaine de certification de manière standard ou personnalisée
- Effectuent un Filtrage des certificats en fonction de l'AC, du subjectDN, de l'empreinte du certificat, etc.

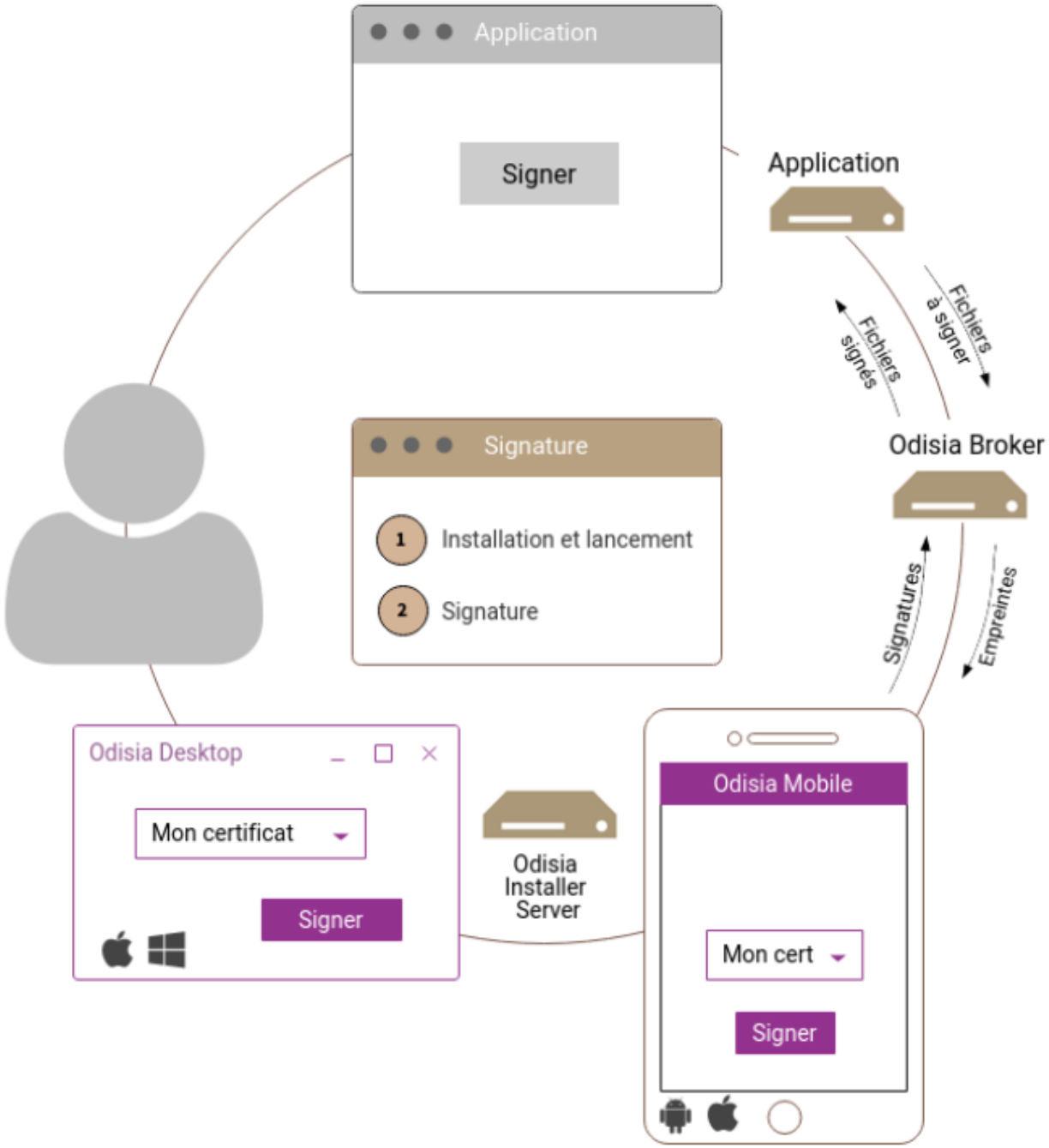
- Prennent en charge la signature d'empreinte

La suite ***Odisia entreprise*** comporte les modules suivants :

- ***Odisia Broker*** qui est un Web Service REST d'intermédiation de signature
- ***Odisia Installer Server*** pour permettre aux utilisateurs de télécharger et installer aisément un client léger de signature électronique
- ***Odisia Desktop*** pour PC et Mac
- ***Odisia Mobile*** pour smartphones et tablettes Android ou iOS (lecteur Bluetooth® nécessaire)
- ***Odisia Broker Signature Page***, page de contrôle du contenu des documents à signer et de mise en œuvre du protocole de consentement

4.2 Description de l'utilisation du produit

4.2.1 Principes de fonctionnement

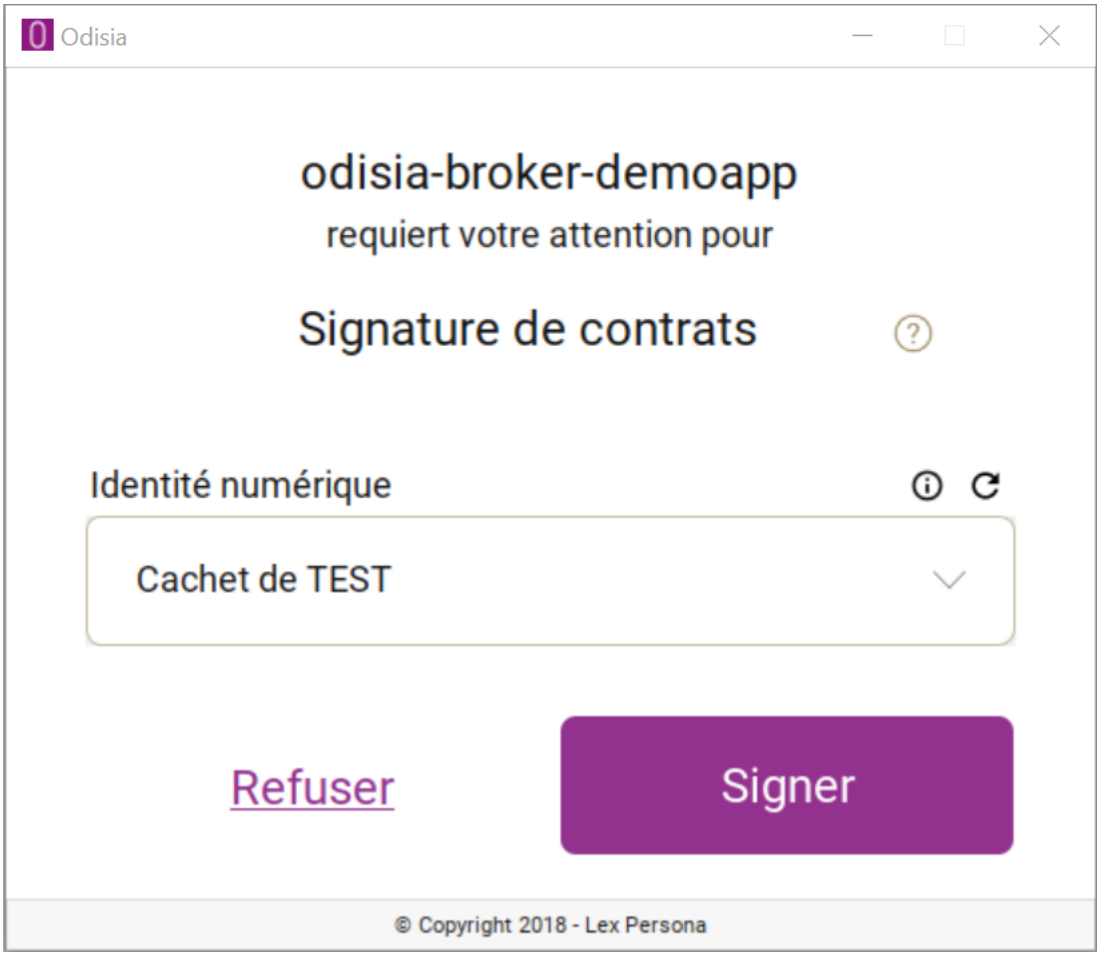


Les principales étapes de l'utilisation du produit sont détaillées ci-dessous.

Lancement

Odisia Desktop, installé sur le poste de l'utilisateur, peut être lancé depuis un navigateur Web grâce à l'URL de lancement Odisia.

L'interface graphique d'***Odisia Desktop*** se décompose en 3 zones :
• La zone d'information et de contexte de la transaction de signature. Dans la copie d'écran ci-dessus, « odisia-broker-demoapp » est le nom de l'entité qui demande à l'utilisateur de signer. Le bouton (?) permet de voir plus de détails sur la transaction de signature (nom de la transaction et liste des documents à signer).
• La zone de sélection de l'identité numérique à utiliser pour signer. C'est une liste déroulante des certificats du signataire avec des pictogrammes indiquant les utilisations de la clé (signature, chiffrement, authentification). Le bouton (i) qui permet de voir plus de détails sur le certificat sélectionné.
• La zone des actions qui permet à l'utilisateur de Signer ou de Refuser la signature.



Odisia Desktop télécharge la transaction de signature
Ensuite, ***Odisia Desktop*** appelles le endpoint « List InputFiles » ***d'Odisia Broker*** permettant de lister les documents à signer (associés à la transaction de signature). Si le téléchargement ou le traitement de la transaction de signature échoue, une erreur est affichée à l'utilisateur ainsi qu'un bouton lui permettant de réessayer.

Signature

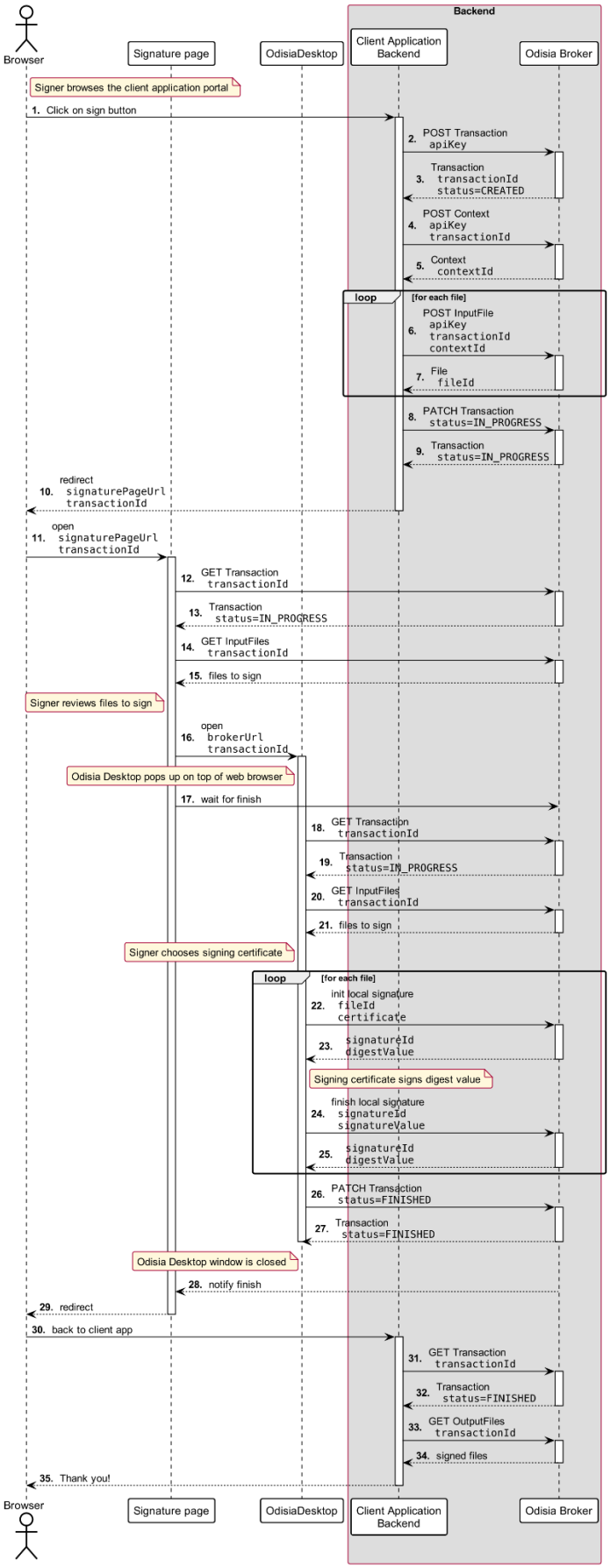
L'utilisateur doit tout d'abord sélectionner dans la liste déroulante des certificats celui qu'il souhaite utiliser pour signer les documents. Ensuite, lorsque l'utilisateur clique sur le bouton de signature, ***Odisia Desktop*** réalise les opérations suivantes pour chaque document à signer :
a) Appel du endpoint « Initialize LocalSignature » d'***Odisia Broker*** permettant de récupérer le hash du document à signer, en passant la chaîne de certification.
b) Génération de la signature du hash du document.
c) Appel du endpoint « Finish LocalSignature » d'***Odisia Broker*** permettant d'envoyer le résultat de signature à ***Odisia Broker*** afin que ce dernier puisse finalement l'intégrer dans le document concerné.

Fermeture

Odisia Desktop peut être fermé manuellement par l'utilisateur (qui peut refuser explicitement de signer ou simplement fermer son ***Odisia Desktop***) ou bien automatiquement après la signature des documents. Si l'utilisateur refuse explicitement de signer, alors ***Odisia Desktop*** mets à jour la transaction dans ***Odisia Broker*** en mettant son statut à « REFUSED ». Si l'utilisateur ferme ***Odisia Desktop*** directement en cliquant sur le bouton (X), alors cela provoquera l'envoi automatique d'une requête vers ***Odisia Broker*** afin de le notifier de la fermeture.

4.2.2 Détail des échanges entre les composants

Pour une meilleure compréhension des mécanismes mis en jeu, le schéma ci-après détaille les différents échanges entre les composants du produit et de son environnement d'exécution.


Backend
Browser
Signature page
OdisiaDesktop
Client Application Backend
Odisia Broker
Signer browses the client application portal
1. Click on sign button
2. POST Transaction apiKey
3. Transaction transactionId status=CREATED
4. POST Context apiKey transactionId
5. Context contextId
loop
[for each file]
6. POST InputFile apiKey transactionId contextId
7. File fileId
8. PATCH Transaction status=IN_PROGRESS
9. Transaction status=IN_PROGRESS
10. redirect signaturePageUrl transactionId
11. open signaturePageUrl transactionId
12. GET Transaction transactionId
13. Transaction status=IN_PROGRESS
14. GET InputFiles transactionId
15. files to sign
Signer reviews files to sign
16. open brokerUrl transactionId
Odisia Desktop pops up on top of web browser
17. wait for finish
18. GET Transaction transactionId
19. Transaction status=IN_PROGRESS
20. GET InputFiles transactionId
21. files to sign
Signer chooses signing certificate
loop
[for each file]
22. init local signature fileId certificate
23. signatureId digestValue
Signing certificate signs digest value
24. finish local signature signatureId signatureValue
25. signatureId digestValue
26. PATCH Transaction status=FINISHED
27. Transaction status=FINISHED
Odisia Desktop window is closed
28. notify finish
29. redirect
30. back to client app
31. GET Transaction transactionId
32. Transaction status=FINISHED
33. GET OutputFiles transactionId
34. signed files
35. Thank you!
Browser
Signature page
OdisiaDesktop
Client Application Backend
Odisia Broker

1 Le signataire clique un bouton « signer » dans une application métier distante.

2 L'application métier initialise une transaction de signature en demandant une création de transaction sur ***Odisia Broker*** (B.TRANSACTION_ID). Voir en annexe la fonction [#api-Transaction-TransactionCreation] pour plus d'information. A noter que l'application métier s'authentifie auprès d'***Odisia Broker*** en fournissant sa clé unique d'identification (B.APIKEY).

3 ***Odisia Broker*** retourne un objet transaction contenant le B. TRANSACTION_ID).

4 L'application métier créé dans ***Odisia Broker*** un ou plusieurs contextes de signature associés au B.TRANSACTION_ID. Un contexte (B.CONTEXT_SIGN) de signature est un ensemble de paramètres de signatures pour un type particulier de fichier à signer. Voir la fonction Create Context [#api-Context-ContextCreation] pour plus d'information.

5 ***Odisia Broker*** retourne un objet contexte qui contient le B.CONTEXT_SIGN.

6 L'application métier dépose un ou plusieurs fichiers (B.DOC_SIGN) sur le serveur ***Odisia Broker***. Les fichiers sont associés à un contexte de signature (B.CONTEXT_SIGN). Voir la fonction Upload InputFile [#api-File-InputFileCreation] pour plus d'information.

7 Pour chaque fichier transféré par l'application métier, ***Odisia Broker*** retourne un objet fichier contenant un identifiant de fichier.

8 Après avoir créé les contextes de signature et envoyé les fichiers à ***Odisia Broker***, l'application métier doit démarrer la transaction de signature en positionnant le statut de la transaction à la valeur IN_PROGRESS. Voir la fonction Patch Transaction [#api-Transaction-TransactionPatch] pour plus d'information.

9 ***Odisia Broker*** retourne l'objet transaction avec le nouveau statut.

10 Une fois la transaction démarrée, l'application métier redirige le signataire vers la page de signature d'***Odisia Broker*** en fournissant le B.TRANSACTION_ID en paramètre. L'URL de signature est de la forme :

`https://{host}/odisia-broker/web/sign?transactionId={id}.`

11 Le navigateur du signataire ouvre la page de signature.

12 à 15 Le signataire récupère depuis ***Odisia Broker*** les fichiers à signer, associés à la B.TRANSACTION_ID. Le signataire est en mesure de visualiser les fichiers qu'il va pouvoir signer.

16 Une fois les fichiers revus par le signataire, ***Odisia Desktop*** s'affiche afin de récupérer les fichiers à signer depuis ***Odisia Broker***.

17 à 21 ***Odisia Desktop*** télécharge les fichiers à signer associés à la transaction en cours. A l'issue du téléchargement, le signataire est invité à choisir le certificat de signature qu'il souhaite utiliser.

22 à 25 Pour chaque document à signer, la signature est initialisée entre ***Odisia Desktop*** et ***Odisia Broker*** (voir la fonction Initialize Local Signature [#api – Initialize Local Signature] pour plus d'information. La signature est réalisée sur le hash des documents calculés par ***Odisia Broker.***

26 à 27 Une fois les documents signés, la transaction est clôturée auprès d'***Odisia Broker***.

28 ***Odisia Desktop*** est fermé et la page de signature est notifiée de la fin de transaction par ***Odisia Broker***.

29 à 30 Le signataire quitte la page de signature et est redirigé vers l'application métier.

31 à 24 L'application métier récupère auprès de ***Odisia Broker*** les documents signés. ***Odisia Broker*** notifie la fermeture de la transaction à l'application métier

35 L'application métier notifie au signataire la fin de la transaction.

4.3 Fonctionnalités de sécurité du produit

La ToE fourni les fonctions de sécurité suivantes pour la partie ***Odisia Broker*** :

- Protection des communications avec l'application métier ;
- Protection des communications avec ***Odisia Desktop ;***
- Identification des application métiers appelantes ;
- Protection des transactions contre le rejeu.
- Calcul de hash des fichiers à signer

Concernant le module ***Odisia Desktop*** la ToE fournit les fonctions de sécurité suivantes :

- Protection des communications avec ***Odisia Broker ;***
- Protection des communications avec l'application métier ;
- Signature de documents.

4.4 Définition du périmètre d'évaluation

Le périmètre d'évaluation comprend les fonctions de sécurité de ***Odisia Broker*** et ***Odisia Desktop*** présentées dans le chapitre précédent.

Sont exclus du périmètre :

- L'application métier.
- Les éléments cryptographiques (ex : token) qui stockent les bi-clés des utilisateurs.

4.5 Définition de la plate-forme d'évaluation

Les systèmes d'exploitation pris en charge par ***Odisia Broker*** sont Windows et Linux
Les systèmes d'exploitation pris en charge par ***Odisia Desktop*** sont Windows, Mac OS X, Android et iOS

Dans le cadre de l'évaluation les systèmes d'exploitation utilisés sont Windows pour ***Odisia Desktop*** et ***Odisia Broker.***

La plate-forme d'évaluation comporte en plus des deux composants soumis à évaluation :

- Une application métier type
- Un token Gemalto IDPrime MD 940

La version Windows utilisée est Windows 10.

5 Utilisateurs de la ToE

Les utilisateurs de la ToE sont les personnes disposant de l'outil ***Odisia Desktop*** sur l'ordinateur personnel et qui signent des données en relation avec une application métier.

Des administrateurs sont également présents dans le périmètre de la ToE. Il s'agit des personnels qui administrent le module ***Odisia Broker.***

6 Hypothèses sur l'environnement

H.APPLI_METIER_SURE
L'application métier détermine les paramètres de signature, y compris les algorithmes. L'application métier est réputée de confiance et ne fait pas partie du périmètre d'évaluation.

H.ODISIA_DESKTOP_SURE
L'application ***Odisia Desktop*** est signée par une clé LEX PERSONA certifiée par l'autorité de certification DIGICERT. Il est considéré que la signature est valide et qu'elle est vérifiée par l'utilisateur au moment de l'installation de l'application.

H.LIB_TOKEN_SURE
Les librairies des différents "token" utilisés pour stocker les clés de signature des utilisateurs de la ToE sont réputées intègres et authentiques. Les tokens utilisés doivent être qualifiés. Les cartes à puce suivantes peuvent être utilisées (mais cette liste n'est pas exhaustive) :

Nom de la solution	Numéro de qualification
Solution IAS Classic sur un composant Infineon	576
Solution IAS Classic sur un composant Samsung	1383
Ideal Citiz	4247
ID One Cosmo	15210

H.MACHINE_SURE
La machine sur laquelle est installée l'application ***Odisia Desktop*** est considérée comme saine et mise à jour régulièrement.

H.SERVEUR_SUR
Le serveur qui héberge la partie ***Odisia Broker*** de la ToE est considéré comme sain et mis à jour régulièrement.

H.ADMIN_SUR
L'administrateur de la machine qui héberge ***Odisia Broker*** est considéré comme formé et non hostile.

H.GENERATION_CLE
La génération de la clé de signature est considérée robuste et de confiance.

7 Biens sensibles de la ToE

B.CONTEXTE_SIGN
Il s'agit du contexte de signature qui est fourni par l'application métier et qui permet de définir les paramètres cryptographiques de la signature.
Besoin de sécurité : intégrité (lors de la transmission du contexte)

B.DOC_SIGN
Il s'agit des documents à signer, produits par l'application métier. Ces documents peuvent être de n'importe quel type (Pdf, XML ou tout autre format binaire avec un format de signature CAdES
Besoin de sécurité: intégrité et authenticité

B.TRANSACTION_ID
Ce bien caractérise l'accès à la transaction de signature par le signataire et l'accès au document. C'est un Identifiant qui est partagé entre **Odisia Broker**, l'application métier et le signataire.
Besoin de sécurité : Confidentialité (pour le temps de la transaction)

B.APIKEY :
Il s'agit d'une clé d'identification qui permet de provisionner des transactions et de provisionner des B.TRANSACTION_ID. Chaque clé d'identification est unique par application cliente afin de pouvoir l'identifier.
Besoin de sécurité : intégrité (dans le cadre des échanges)

B.URL_SIGN
Il s'agit de l'URL de signature, générée pour permettre à l'utilisateur d'être redirigé pour signature vers une URL spécifique à la transaction. L'URL contient le B.TRANSACTION_ID permettant d'identifier la transaction.
Besoin de sécurité : Confidentialité (pour le temps de la transaction)

B.HASH_DOCUMENT
Calcul cryptographique effectué par un élément cryptographique sur les documents à signer et qui est transmis à ***Odisia Broker***.
Besoin de sécurité : Intégrité

8 Menaces

M.ALTERATION_DOC_TRANSMIS
Un attaquant altère les documents transmis au signataire par l'application métier.

M.ALTERATION_DOC_SIGNES
Un attaquant modifie les documents signés par le signataire

M.CONTREFAIT_TRANSAC
Un attaquant contrefait un identifiant de transaction (B.TRANSACTION_ID) pour se substituer au signataire légitime ou pour fournir au signataire légitime des mauvais documents à signer.

M.ALTERATION_CONTEXTE
Un attaquant altère le contexte de signature pour diminuer la robustesse du niveau de signature.

M.USURP_APPLI
Un attaquant tente d'usurper une application métier vis-à-vis de ***Odisia Broker*** afin de faire réaliser des transactions illégitimes.

M.CONTREFAIT_URL
Un attaquant contrefait une URL de signature afin de rediriger le signataire vers une page de signature illicite.

M.ALTERATION_HASH
Un attaquant modifie le hash des documents transmis à ***Odisia Broker*** afin de masquer une modification des données dans les documents.

9 Fonctions de sécurité

F.PROTECT_COM_APPLI_BROKER
La ToE fourni des communications sécurisées entre l'application métier et ***Odisa Broker*** permettant la protection en confidentialité et en intégrité des données échangées. La protection des communications est réalisée à l'aide du protocole HTTPS (TLS.1.2 utilisant les suites cryptographiques
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 ou
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384).

F.PROTECT_COM_BROKER_CLIENT
La ToE fourni des communications sécurisées entre ***Odisia broker*** et le poste du signataire permettant la protection en confidentialité et en intégrité des données échangées. La protection des communications est réalisée à l'aide du protocole HTTPS (TLS.1.2 utilisant les suites cryptographiques
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 ou
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384).

F.IDENTIFY_TRANSACTION
La ToE permet d'identifier de manière unique les transactions de signature en fournissant un identifiant de transaction unique (B.TRANSACTION_ID) généré à partir d'un aléa (Java Secure Random) converti en base 62 (caractères alphanumériques uniquement) puis tronqué à 64 caractères
L'identification de la transaction permet également de protéger l'URL de signature de la même manière qu'un identifiant de session.

F.IDENTIFY_API
La ToE permet d'identifier de manière unique les application métiers appelantes sur ***Odisia Broker***. ***Odisia Broker*** identifie par comparaison de la clé transmise par l'application appelante (B.APIKEY) avec une référence stockée sur ***Odisia Broker***. La clé d'identification est générée de la manière suivante : aléa (/dev/urandom) converti en base 62 (caractères alphanumériques uniquement) puis tronqué à 64 caractères

F.SIGN
La ToE permet la signature des documents via ***Odisia Desktop*** en utilisant des ressources cryptographiques stockant les bi-clés de signature. La ToE supporte les algorithmes de signature suivants : SHA-256 avec RSA-2048.

F.VERIF_SIGN
La ToE permet de réaliser sur ***Odisia Broker*** la vérification de la signature réalisée par le signataire au travers de ***Odisia Desktop***.

F.CALC_HASH

La ToE permet de réaliser sur ***Odisia Broker*** un calcul de hash (SHA256) sur chaque fichier qui est à signer par le signataire ; Le hash est transmis à ***Odisia Desktop*** pour signature par le signataire via la fonction F.SIGN.

10 Argumentaires

Le tableau ci-dessous présente la traçabilité entre les biens et les menaces

	M.ALTERATION_DOC_TRANSMIS	M.ALTERATION_DOC_SIGNES	M.CONTREFAIT_TRANSAC	M.ALTERATION_CONTEXTE	M.USURP_APPLI	M.CONTREFAIT_URL	M.ALTERATION_HASH
B.CONTEXTE_SIGN				X			
B.DOC_SIGN	X	X					
B.TRANSACTION_ID			X				
B.APIKEY :					X		
B.URL_SIGN						X	
B.HASH_DOCUMENT							X

Le tableau ci-dessous présente la traçabilité entre les menaces et les fonctions de sécurité

	M.ALTERATION_DOC_TRANSMIS	M.ALTERATION_DOC_SIGNES	M.CONTREFAIT_TRANSAC	M.ALTERATION_CONTEXTE	M.USURP_APPLI	M.CONTREFAIT_URL	M.ALTERATION_HASH
F.PROTECT_COM_APPLI_BROKER	X	X		X			
F.PROTECT_COM_BROKER_CLIENT	X	X					
F.IDENTIFY_TRANSACTION			X			X	
F.IDENTIFY_API					X		
F.SIGN		X					
F.VERIF_SIGN		X					X
F.CALC_HASH							X