# wishPIN

# (Hmac→token ou tokenCB)

Xpollens

PAYMENT INSIDE

# Evolution token ou tokenCB

## Préambule

La solution wishPIN de Xpollens doit évoluer pour être conforme aux nouveaux standards bAPI de PPS et BPCE.

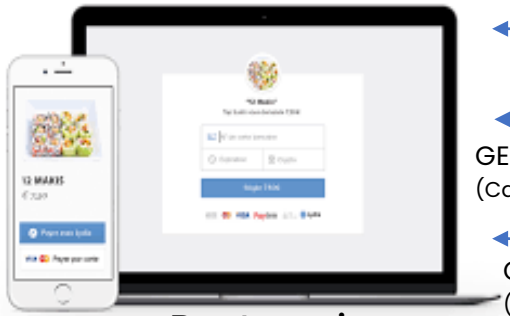
La solution **signature HMAC** actuelle perdurera le temps que nos partenaires migrent sur la solution Target en **signature token**.

**Le principe fonctionnel est le même.** Pour définir son PIN, il faut entrer à la commande de carte la même référence (cardExternalRef ou AppcardID) que lors de l'appel à la signature Token avec cette ressource et fournir le token en entrée de l'appel (qui devra être modifié ) du SDK wishPIN.

/!\ : A noter qu'il y a 2 intégrations possibles dont l'une fait changer **l'UX sur le mobile**. Lors de l'appel en Token**CB**, un clavier virtuel avec indexation et coordonnées du clavier est présenté à l'utilisateur pour saisie de son code PIN. Le clavier est configurable via des input parameter..

# WishPIN

## Schéma d'architecture




2 3 Définition et Transfert PIN en temps réel
CardID=CardExternalRef

1 2 GET TokenCB/POST Hmac
(CardExternalRef)

3 1 Commande carte
(CardExternalRef)

1 2 3 Le processus peut également suivre ce sens, soit commencer par le wishPN avant la commande carte.

Xpollens
PAYMENT INSIDE

Thalès/Gemalto
Processeur Externe
PIN Processing

5 PIN Block
PIN/CardID=CardExternalRef

4 Entrée Batch
PAN/CardID=CardExternalRef

**SI Producteur Natixis Payments**

6 Smart Porteur Référentiel
Demande à temporiser

7 Pegase
Réconciliation
PAN/ PIN

8 Perso Carte

G&D
Idemia
Thalès
**Ateliers de fabrication cartes**

9

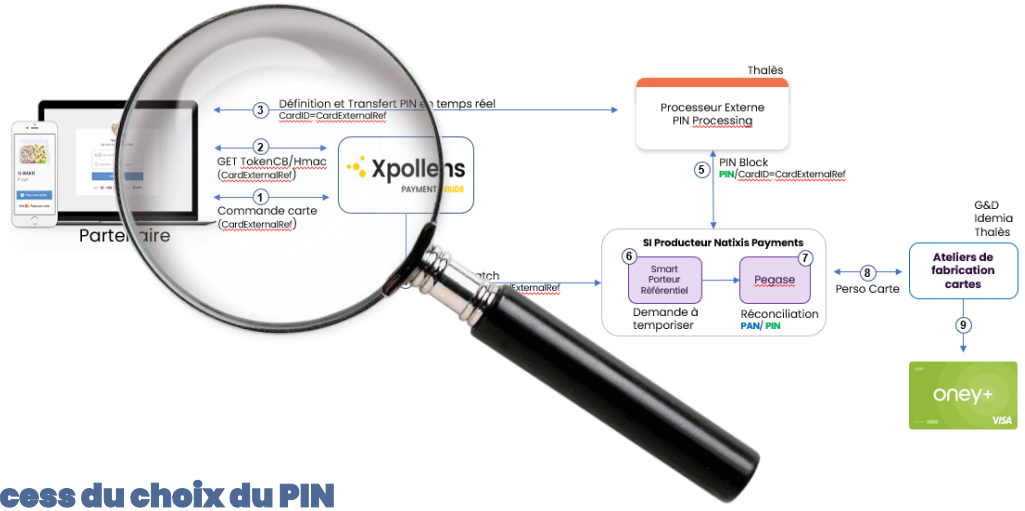

## Process du choix du PIN

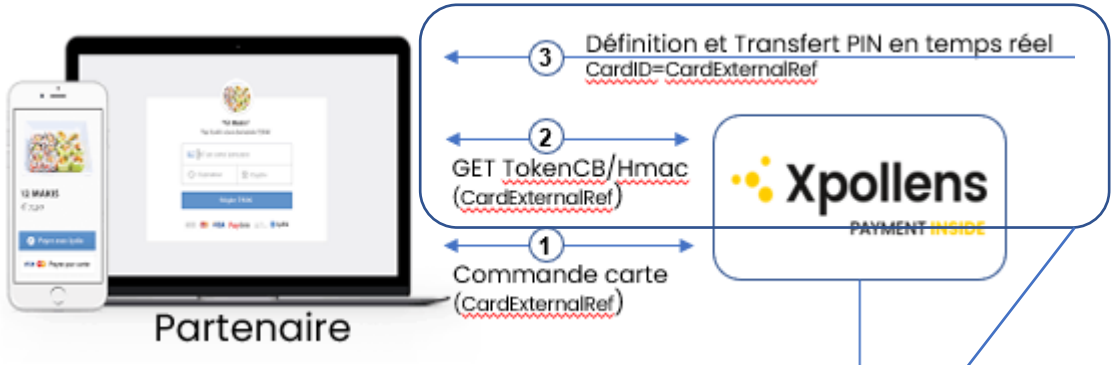
1. Le partenaire commande la carte par API.
2. Le partenaire appelle Xpollens pour récupérer une signature par API.
3. Le porteur choisit son PIN en ligne lors de la création de carte.
   Le partenaire appelle le sdk en mode define PIN
4. Xpollens transmet la demande de création de carte à Smart Porteur Référentiel.
5. En parallèle, Thalès transmet le fichier des BCC, contenant les PIN chiffrés à Pegase.
6. Smart Porteur Référentiel envoie les demandes de création carte à temporiser à Pegase
7. Pegase réconcilie PAN et PIN (via l'identifiant Porteur COJAVC) et échange avec le HSM pour transchiffrement du BCC et stockage dans son référentiel
8. Les fichiers cartes (et mailers si paramétrés) sont produits et transmis aux ateliers
9. La carte personnalisée est produite

# WishPIN : Focus token

**RECOMMANDE**

## Voici ce qui change : 1 appel API + 1 appel sdk modifiés




### Process du choix du PIN

1. Le partenaire commande la carte par API.
2. Le partenaire appelle Xpollens pour récupérer une signature **Token** par API depuis son backend.
3. *Le partenaire affiche un clavier (virtuel recommandé).*
   Le porteur saisit son PIN depuis son téléphone lors de la création de carte. Le partenaire appelle le sdk **en mode define PIN via token**
4. *Xpollens transmet la demande de création de carte à Smart Porteur Référentiel.*
5. *En parallèle, Thalès transmet le fichier des BCC, contenant les PIN chiffrés à Pegase.*
6. *Smart Porteur Référentiel envoie les demandes de création carte à temporiser à Pegase*
7. *Pegase réconcilie PAN et PIN (via l'identifiant Porteur COJAVC) et échange avec le HSM pour transchiffrement du BCC et stockage dans son référentiel*
8. *Les fichiers cartes (et mailers si paramétrés) sont produits et transmis aux ateliers*
9. *La carte personnalisée est produite*

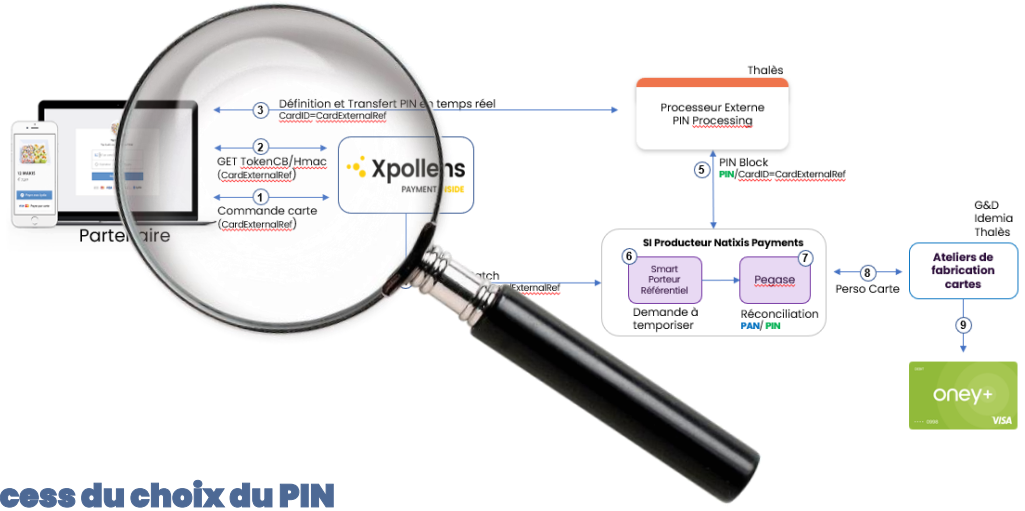
2. Le partenaire appelle Xpollens (nouvelle API) pour récupérer une signature **Token** par API depuis son backend.

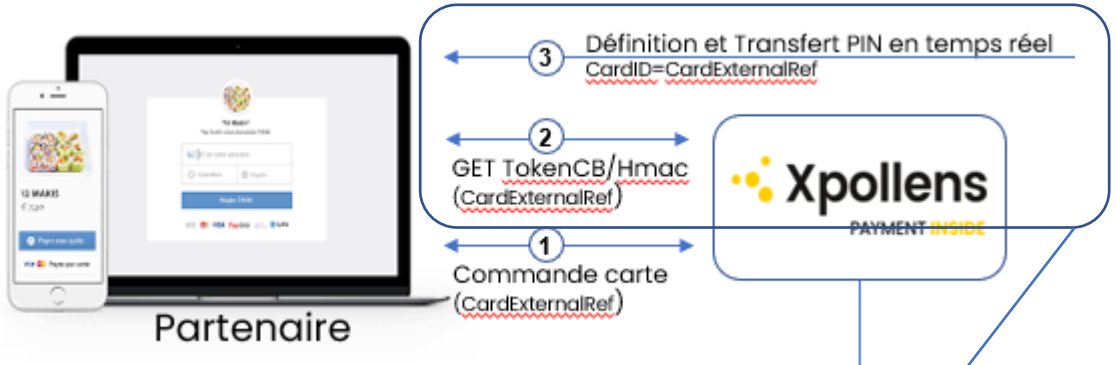
3. *Le partenaire affiche un clavier (recommandé). Le porteur saisit son PIN depuis son téléphone lors de la création de carte.*
   ➔ **Inchangé**

   Le partenaire appelle le sdk **en mode define PIN via token et non plus en hmac**

# WishPIN : Focus tokenCB (imposé GIE CB)

## Voici ce qui change : idem token + clavier de coordonnées




### Process du choix du PIN

1. Le partenaire commande la carte par API.
2. Le partenaire appelle Xpollens pour récupérer une signature **Token** par API depuis son backend.
3. **Le partenaire affiche un clavier (virtuel obligatoire). Le porteur saisit son PIN et ce sont les coordonnées qui sont transmises à l'atelier.** Le partenaire appelle le sdk **en mode define PIN via token**
4. *Xpollens transmet la demande de création de carte à Smart Porteur Référentiel.*
5. *En parallèle, Thalès transmet le fichier des BCC, contenant les PIN chiffrés à Pegase.*
6. *Smart Porteur Référentiel envoie les demandes de création carte à temporiser à Pegase*
7. *Pegase réconcilie PAN et PIN (via l'identifiant Porteur COJAVC) et échange avec le HSM pour transchiffrement du BCC et stockage dans son référentiel*
8. *Les fichiers cartes (et mailers si paramétrés) sont produits et transmis aux ateliers*
9. *La carte personnalisée est produite*

2. Le partenaire appelle Xpollens (nouvelle API) pour récupérer une signature **Token** par API depuis son back-end.

3. Le partenaire affiche un clavier à l'aide de coordonnées. Le porteur saisit son PIN depuis ce clavier lors de la création de carte**. Les coordonnées (et non le PIN) sont ensuite transmises à l'atelier pour déchiffrage du PIN choisi.**

   Le partenaire appelle le sdk **en mode define PIN via token (comme un token classique) et non plus en hmac**

# Intégration

Xpollens
PAYMENT INSIDE

## 2021 Signature HMAC

### API

**API signature hashMac**

POST api/v1.0/cards/{appcardid}/hmac}

- appCardID

### SDK

**PIN Definition - HMAC**

postDefinePINRequest

- Bank ID
- Unique ID
- Timestamp
- hashMac
- PIN
- hostUrl

## Q2 2022 Target Solution

### API

**API signature token**

GET /api/v2.0/tokensignature/{cardExternalRef}

- cardExternalRef

### SDK

**PIN Definition - token**

**postDefinePINECDHRRequest**

- Token
- signature
- PIN
- hostUrl

# Recette communautaire

## Etapes

**Afin de vous accompagner dans le changement, nous proposons une recette communautaire**

### *1 – Recette bipartite Xpollens - NPS*

- Commande carte Entrée Batch avec choix du code PIN (CEBCC SMPREF)
- Intégration API de signature TokenCB *

### *2 – Recette bipartite Xpollens - Thalès*

- ** Prérequis Intégration API de signature TokenCB*
- Intégration API/SDK de choix du PIN Thalès
- Réception des BCC (PINBLOCK chiffrés) pour vérification par NPS

### *3 – Recette tripartite Xpollens – Thalès – NPS*

- Commande carte Entrée Batch avec choix du code PIN (CEBCC SMPREF)
- Choix du PIN (API/SDK Thalès + API de signature Token CB)
- Traitement NPS de réconciliation PAN-PIN
- Scénario de recette avec validation des paramétrages NPS (table PCC044) : délai de rétention BCC et délai de choix du code

### *4 – Recette OBD (bout en bout)*

- Commande carte Entrée Batch avec choix du code PIN (CEBCC SMPREF)
- Choix du PIN (API/SDK Thalès + API de signature Token CB)
- Traitement NPS de réconciliation PAN-PIN
- Scénario de recette avec validation des paramétrages NPS (table PCC044) : délai de rétention BCC et délai de choix du code

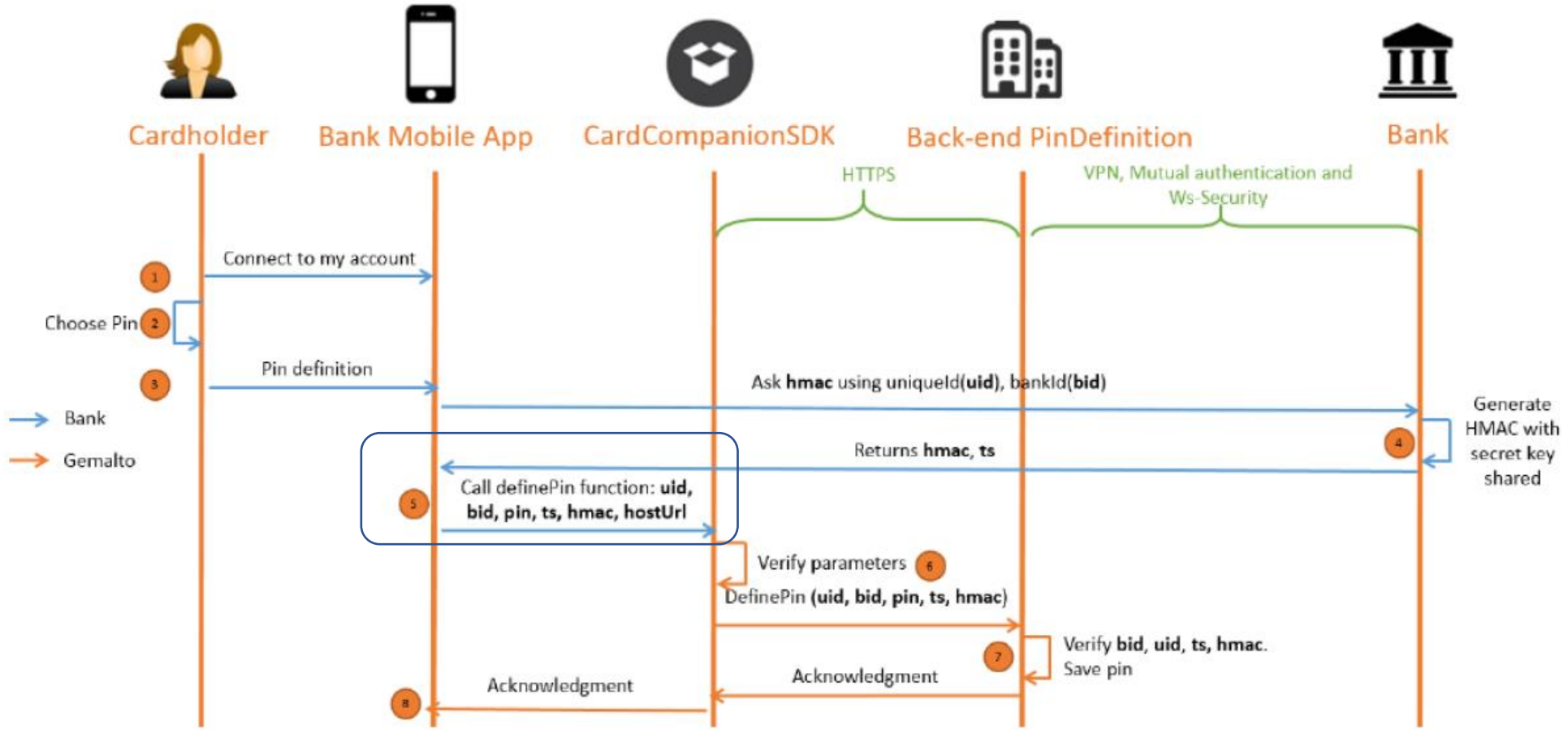
# ANNEXES

Xpollens

PAYMENT INSIDE

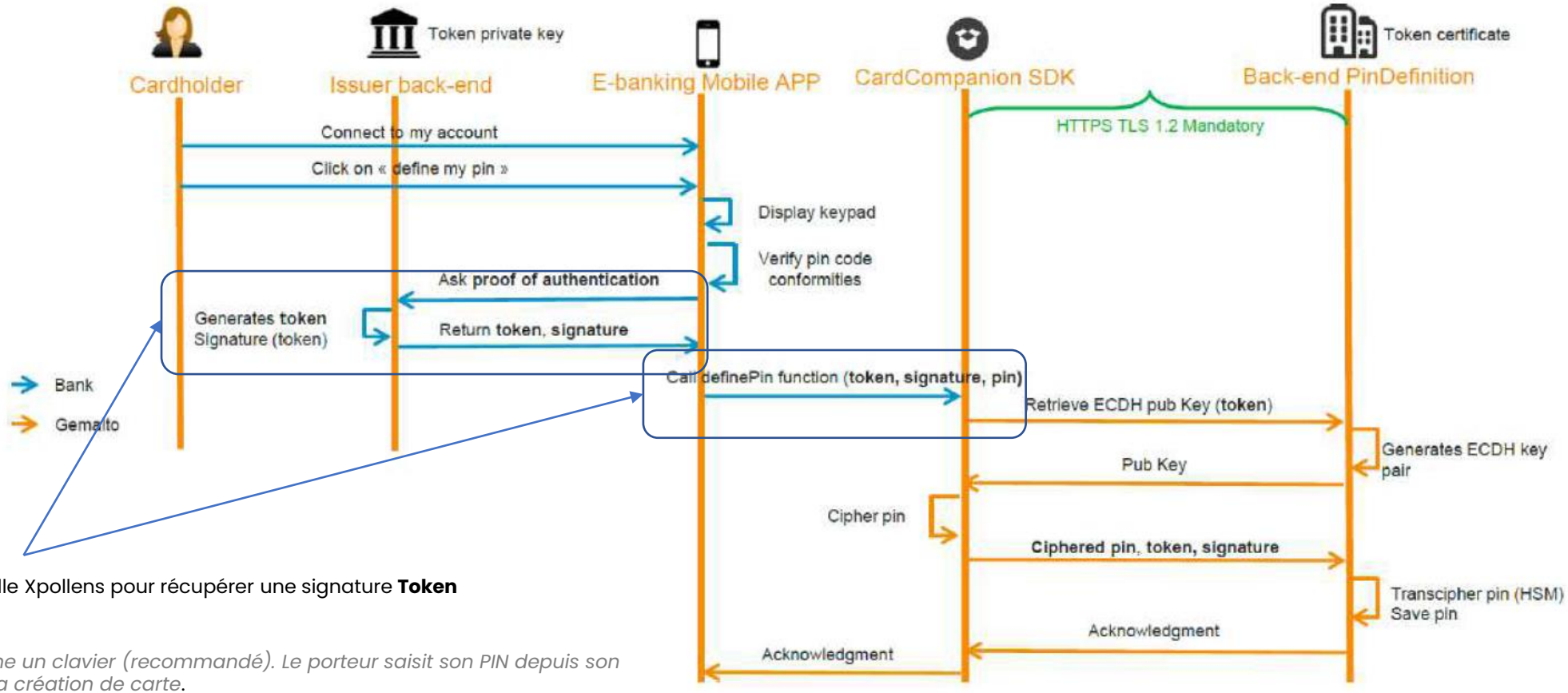
# Signature actuelle : Hmac

## Parcours



# Signature Target : Token

## Parcours



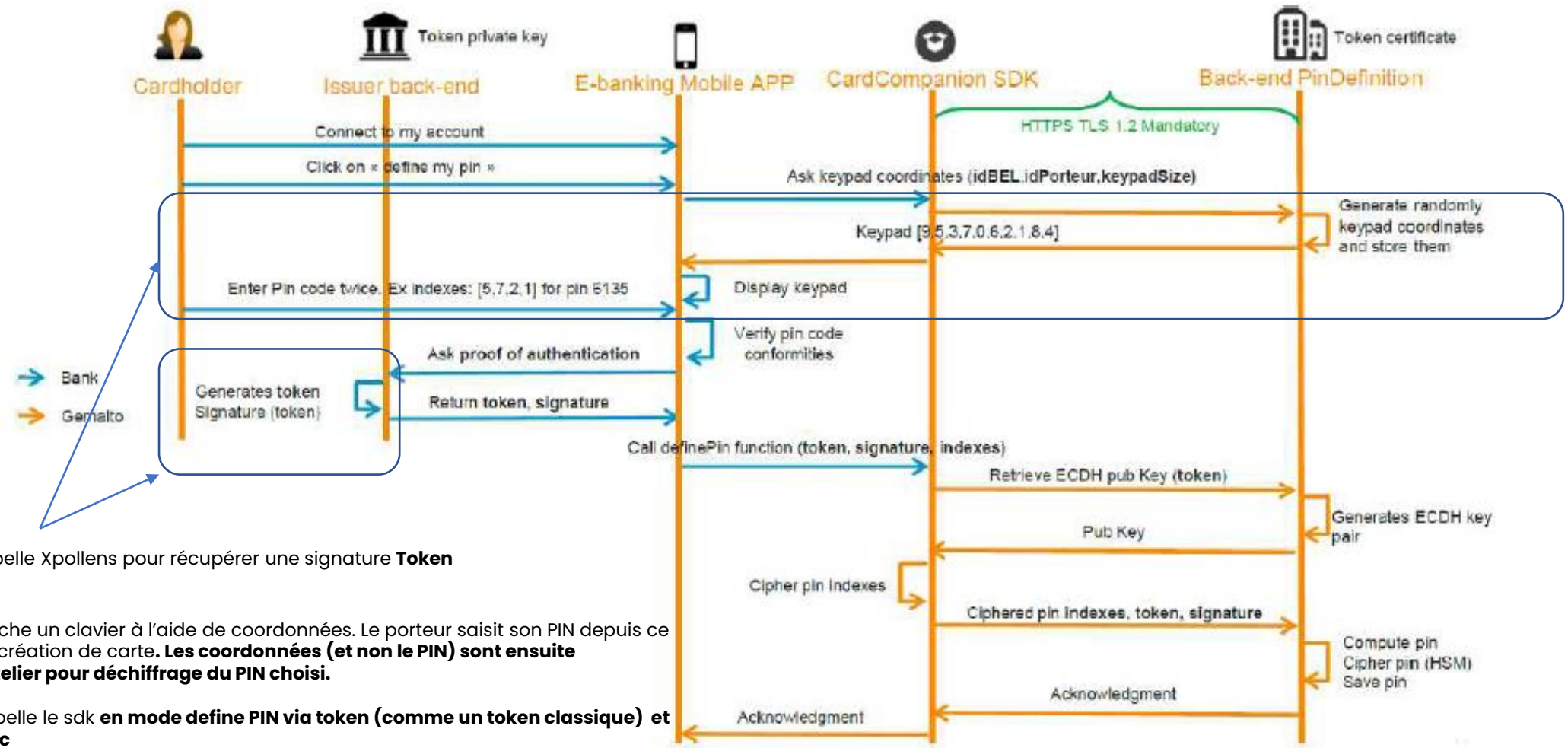

② Le partenaire appelle Xpollens pour récupérer une signature **Token** par API.

③ *Le partenaire affiche un clavier (recommandé). Le porteur saisit son PIN depuis son téléphone lors de la création de carte.*
→ **Inchangé**

Le partenaire appelle le sdk **en mode define PIN via token et non plus en hmac**

# Signature Target : TokenCB

## Parcours




② Le partenaire appelle Xpollens pour récupérer une signature **Token** par API.

③ Le partenaire affiche un clavier à l'aide de coordonnées. Le porteur saisit son PIN depuis ce clavier lors de la création de carte**. Les coordonnées (et non le PIN) sont ensuite transmises à l'atelier pour déchiffrage du PIN choisi.**

Le partenaire appelle le sdk **en mode define PIN via token (comme un token classique) et non plus en hmac**

# Intégration API Xpollens + Appel SDK1.7

Xpollens
PAYMENT INSIDE

## HMAC Actuel

**API Xpollens**

**API Hmac Signature**

POST api/v1.0/cards/{appcardid}/hmac

- Input : appCardId
- Output : Signature hMac

**SDK Thalès**

**PIN Definition - token**

**postDefinePINECDHRRequest**

- BankId[1] (30007)
- uniqueId
- Timestamp
- hashMac[2]
- pin[3]
- hostUrl
- Context

Code banque/code guichet

- **Exemple uniqueId :** {"IdPorteur":"1286900002Ft6Fxy4gqkWfClRmRLqPnQ"}
- **Exemple SIGNATURE hMac[2] :**
  DxFMpPGROqMcb5QsNnovjBXFjqpClMnmXNP5q5yJ5i5ahSvCJF2XLjKRH0CtYfm1f0zaFAO6HA1ArVD+RQH0UQh5ImwSWD7r1KMgkwMwOVcIrP7kn9wv7+5VblH9XPw95V6yXckQDAb5XDCToz9xH+OBLweD0tf78uShM6l5shIJZCJDvWDWloDpU54Az28gGdCAtm43Qjnd8h82hrtZS5Ea+aqa7/oe4RP6cC+nEyPpeTUYcCcxbNhhxIXsnAu+b1x7l+DxIdeQjcXrlShLqHN/HrGvBsgHzSmcsw/wW380CtzeRctiEij9Lvc2kxCq+GR1FJMm/8oDrnHBXQLIzw==
- **Exemple PIN[3] :** 3488

# Intégration API Xpollens + Appel SDK1.7

Xpollens
PAYMENT INSIDE

**Token Actuel**

**Version actuelle et mauvais fonctionnement, voir slide suivant sur le correctif apporté et le bon usage**

**API Xpollens**

**API signature token**

GET /api/v2.0/tokensignature/{cardExternalRef}

- Input : cardExternalRef
- Output : certificat + Signature

**SDK Thalès**

**PIN Definition - token**

**postDefinePINECDHRRequest**

- Token[1]
- signature[2]
- PIN[3]
- hostUrl

**Oney appelle le SDK d'une mauvaise manière car pas d'output de l'api lui permettant de faire autrement** : PinCodeRequest.postDefinePinRequestECDH(token, pin:pin, signature:signature, hostURL:url), avec *token => tokenSignature depuis Signature et signature => certificateAlias*.

Code banque/code guichet

Input data utilisées pour créer la signature et inconnues du Partner pour appeler le sdk

- **Exemple TOKEN[1] :**
{"IdBEL":"30007","IdPorteur":"I286900002Ft6Fxy4gqkWfClRmRLqPnQ","IdFournisseur":"TH","IdTransaction":"699210410","Timestamp":"1615816121","Type":"02","SignatureCertAlias":"natixis.dev"}
- **Exemple SIGNATURE[2] :**
DxFMpPGROqMcb5QsNnovjBXFjqpClMnmXNP5q5yJ5i5ahSvCJF2XLjKRH0CtYfm1f0zaFAO6HA1ArVD+RQH0UQh5ImwSWD7r1KMgkwMwOVcIrP7kn9wv7+5VblH9XPw95V6yXckQDAb5XDCToz9xH+OBLweD0tf78uShM6l5shIJZCJDvWDWloDpU54Az28gGdCAtm43Qjnd8h82hrtZS5Ea+aqa7/oe4RP6cC+nEyPpeTUYcCcxbNhhxIXsnAu+b1x7l+DxIdeQjcXrIShLqHN/HrGvBsgHzSmcsw/wW380CtzeRctiEij9Lvc2kxCq+GR1FJMm/8oDrnHBXQLIzw==
- **Exemple PIN[3] :** 3488

# Intégration API Xpollens + Appel SDK1.7

Xpollens
PAYMENT INSIDE

## Token Solution

**API Xpollens**

**API signature token**

GET /api/v2.0/tokensignature/{cardExternalRef}

- Input : cardExternalRef
- Output : certificat + Signature + data bleue

**SDK Thalès**

**PIN Definition - token**

**postDefinePINECDHRRequest**

- Token[1]
- signature[2]
- PIN[3]
- hostUrl

- **Exemple TOKEN[1] :**
{"IdBEL":"30007","IdPorteur":"1286900002Ft6Fxy4gqkWfClRmRLqPnQ","IdFournisseur":"TH","IdTransaction":"699210410","Timestamp":"1615816121","Type":"02","SignatureCertAlias":"natixis.dev"}
- **Exemple SIGNATURE[2] :**
DxFMpPGROqMcb5QsNnovjBXFjqpClMnmXNP5q5yJ5i5ahSvCJF2XLjKRH0CtYfm1f0zaFAO6HA1ArVD+RQH0UQh5ImwSWD7r1KMgkwMwOVcIrP7kn9wv7+5VblH9XPw95V6yXckQDAb5XDCToz9xH+OBLweD0tf78uShM6l5shIJZCJDvWDWloDpU54Az28gGdCAtm43Qjnd8h82hrtZS5Ea+aqa7/oe4RP6cC+nEyPpeTUYcCcxbNhhxIXsnAu+b1x7l+DxIdeQjcXrIShLqHN/HrGvBsgHzSmcsw/wW380CtzeRctiEij9Lvc2kxCq+GR1FJMm/8oDrnHBXQLIzw==
- **Exemple PIN[3] :** 3488

# Intégration API Xpollens + Appel SDK2.0

Xpollens
PAYMENT INSIDE

## Token Solution

**API Xpollens**

**API signature token**

GET /api/v2.0/tokensignature/{cardExternalRef}

- Input : cardExternalRef
- Output : certificat + Signature + data bleue

**SDK Thalès**

**PIN Definition - token**

**PINDefinition.definePINToken**

- Token[1]
- signature[2]
- PIN[3]
- hostUrl

- **Exemple TOKEN[1] :**
{"IdBEL":"30007","IdPorteur":"1286900002Ft6Fxy4gqkWfClRmRLqPnQ","IdFournisseur":"TH","IdTransaction":"699210410","Timestamp":"1615816121","Type":"02","SignatureCertAlias":"natixis.dev"}
- **Exemple SIGNATURE[2] :**
DxFMpPGROqMcb5QsNnovjBXFjqpClMnmXNP5q5yJ5i5ahSvCJF2XLjKRH0CtYfm1f0zaFAO6HA1ArVD+RQH0UQh5ImwSWD7r1KMgkwMwOVcIrP7kn9wv7+5VblH9XPw95V6yXckQDAb5XDCToz9xH+OBLweD0tf78uShM6l5shIJZCJDvWDWloDpU54Az28gGdCAtm43Qjnd8h82hrtZS5Ea+aqa7/oe4RP6cC+nEyPpeTUYcCcxbNhhxIXsnAu+b1x7l+DxIdeQjcXrIShLqHN/HrGvBsgHzSmcsw/wW380CtzeRctiEij9Lvc2kxCq+GR1FJMm/8oDrnHBXQLIzw==
- **Exemple PIN[3] :** 3488