BIIC

Spécifications des interfaces *MeekFi / ICPS / Amplitude*

- *Service temps réel*
- *APPLICATION DEBIT*
- *MAJCARD*
- *BCLEAR*

Version	*Auteur*	*Date*	*Observations*
Version 1.0	*Expert'iz*	*13/06/24*	*Création du document*

Sommaire

1. Introduction

BIIC entend émettre via son reseau ATM un service de retrait sans carte.

La demande d'autorisation de retrait sans carte est initiée depuis le GAB de BIIC, soumis à plateforme d'autorisation MeekFi via le CMS ICPS. Le dénouement du service de retrait sans carte a une forte incidence comptable sur le Core Banking Amplitude.

L'orchestration des échanges entre les plateformes ICPS, MeekFi et Amplitude doit s'opérer au travers d'un middleware à déployer en interne par la banque.

Le présent document est élaboré pour définir et valider les spécifications du middleware d'orchestration des échanges à savoir :

- API ICPS/Middle
- API Middle /MeekFi
- API Middle/Amplitude
- Interface batch ICPS/Amplitude

2. Architecture et cinématique des services à implémenter

2.1. Demande d'autorisation d'un retrait

La demande d'autorisation est préparée et émise depuis le GAB de la banque et router à la plateforme MeekFi pour traitement.

Le dénouement de la demande d'autorisation se décline au travers de la cinématique suivante :

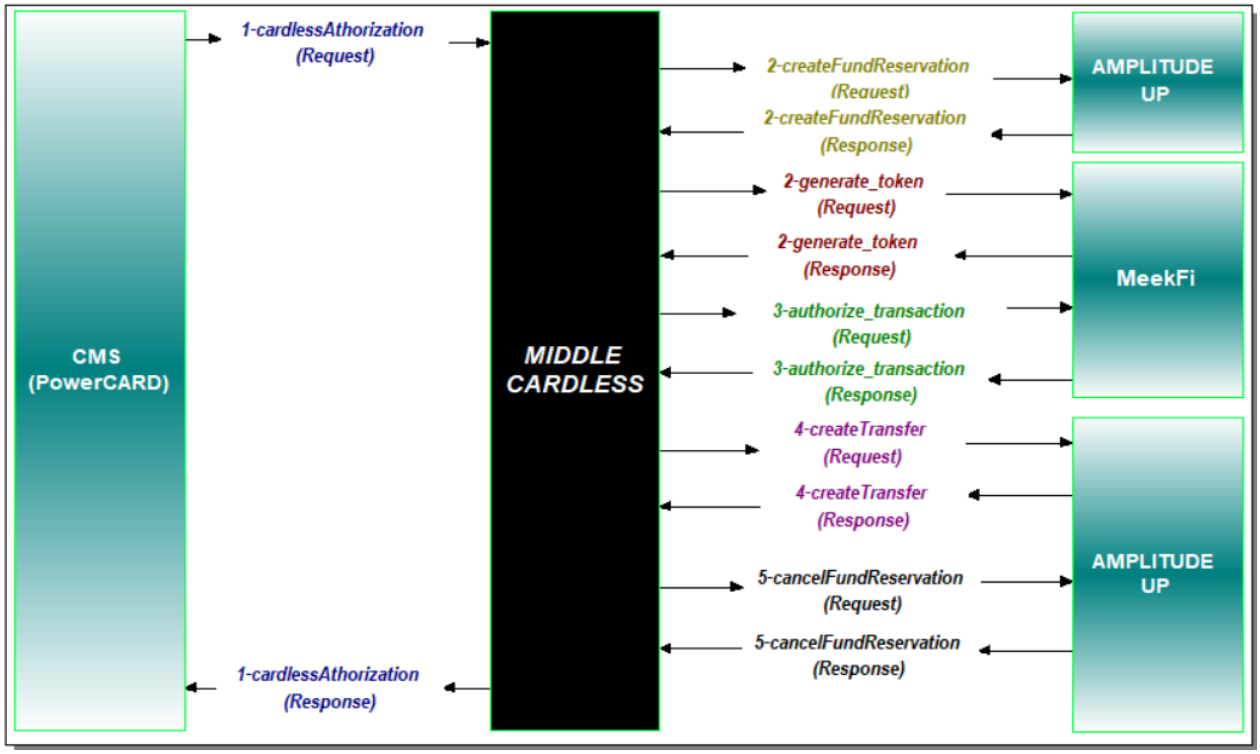

Figure 1 : Cinématique de la demande d'autorisation

On déroule l'étape 2 telque prévu et ici l'etape 3 serait plutôt createFundReservation (initiée par Meekfi qui appelle un service exposé au niveau du middle cardless). Une fois la réservation de fonds est confirmée on passe à l'etape 4 qui serait donc maintenant authorize_transaction, ensuite on aurait une étape étape 5 qu'on pourrait appeler confirm_transaction (initiée par Meekfi, et exposé au niveau du middle Cardless dont le rôle sera de cancel la reservation de fonds et de passer le virement du compte opérateur vers le compte de transit). On pourrait donc avoir une etape 6 finale confirm_tx_atmaccount (initiee au niveau du middle cardless) dont l'objectif est de faire un createTransfer entre le compte de transit précédemment crédité et le compte GAB

2.2. Annulation d'une demande d'autorisation

En cas d'incident de paiement par le GAB, une demande d'annulation est soumise à la plateforme MeekFi par le CMS PowerCARD.

Le dénouement de la demande d'annulation se décline au travers de la cinématique suivante :

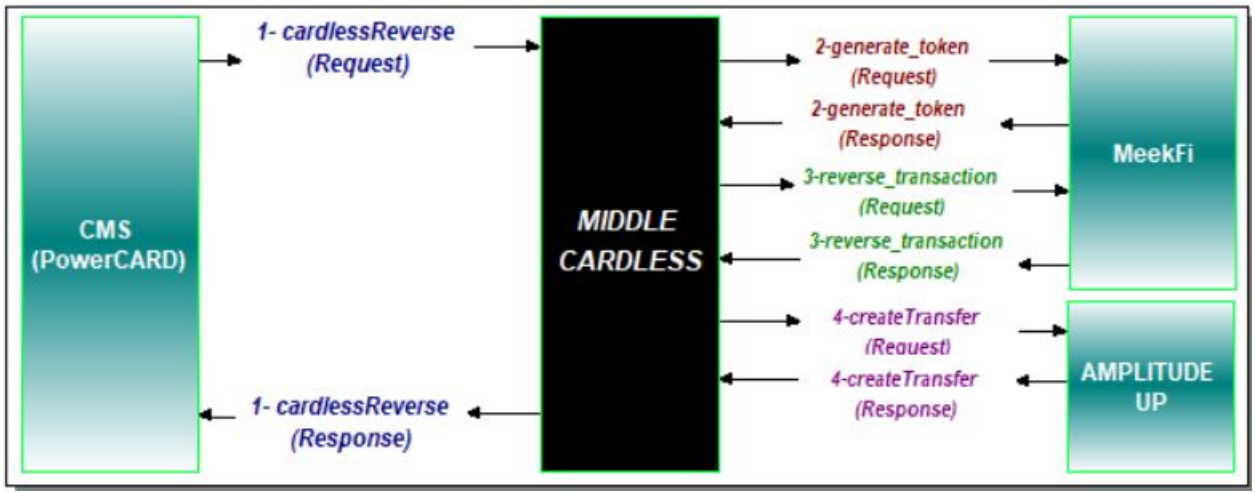

Figure 2 : Cinématique de la demande d'annulation

En cas de reversal (sous reserve que CMS puisse transmettre en real time les cas de transactions échouées) alors on suivrait le workflow défini juste que l'etape 4 serait plutôt un cancelFundReservation en lieu et place du createTransfer et cette transaction devra être initiée par Meekfi qui appellerait un service exposé pour la circonstance au niveau du middle Cardless

N.B : les specs des Apis createFundreservation, confirm_transaction, et cancelFundReservation seront mis à disposition de Meekfi dans un document séparé de ce dernier

2.3. Traitement batch

En fin de journée, un fichier de transaction (DAP Transaction) est généré par PowerCARD et mis à la disposition de MIDDLE CARDLESS pour réconciliation.

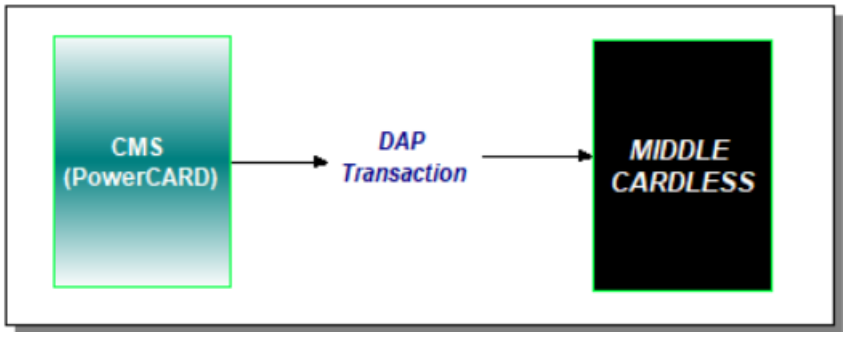

Figure 3 : Fichier de transaction sans carte

2.4. Architecture technique

Les services d'émission et d'autorisation des retraits sans carte englobent des composants matériels et des plateformes de paiement.

L'architecture technique qui expose les services se déploient sur 3 sites différents.

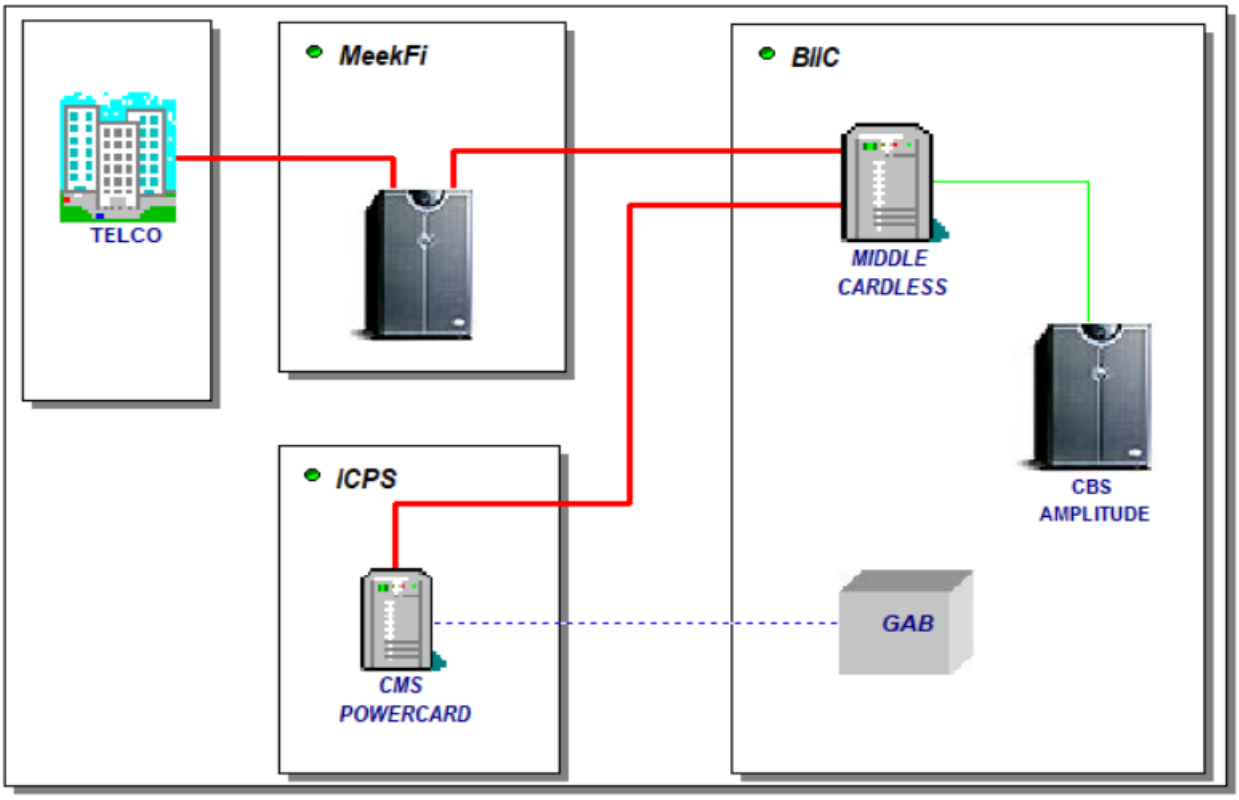

Figure 3 : Architecture technique déployée derrière le retrait sans carte

3. Spécifications des flux

Dans la mise en œuvre de ses services de paiement de retrait sans carte, les flux échangés entre les applications sont de 3 ordres à savoir :

- Les flux PowerCARD/BIIC (Middle CardLess)
- Les flux BIIC (Middle CardLess)/MeekFi
- Les flux BIIC (Middle CardLess)/Amplitude

Ci-après les spécifications des API à exploiter dans le cadre de la mise en œuvre des services de retrait sans carte.

3.1. Flux PowerCARD / BIIC (Middle CardLess)

3.1.1. getToken

L'API `getToken` permet à une application d'obtenir un jeton d'accès (access_token) en utilisant les informations d'identification du client (client_id et client_secret) via le flux de type client_credentials, ce qui lui permet d'accéder à des ressources protégées avec une portée spécifiée, dans ce cas "openid".

API : getToken			
Client	PowerCARD (ICPS)		
Serveur	BIIC (MIDDLE CARDLESS)		
REQUEST			
Champ	**Type**	**M/O**	**Description**
grant_type	String	M	Le type de flux : **client_credentials**
client_id	String	M	L'identifiant du client. **ICPS**
client_secret	String	M	Code secret du client. **Biic&icps2024$**
Scope	String	M	Spécifie les autorisations. **openid**
RESPONSE			
access_token	String	M	Le jeton d'Access
token_type	String	M	Le type de jeton : **bearer**
refresh_token	String	M	Le jeton de rafraîchissement
expires_in	Integer	M	**Indique la durée de validité du jeton d'access**

Exemple :

Request : https://10.10.104.181:48003/cardless/atm/v1/getToken

```
{
   "grant_type": "client_credentials",
   "client_id": "ICPS",
   "client_secret": "Biic&icps2024$"
  "scope": "openid"
}
```

Response:

{"access_token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJJQ1BTIiwiZXhwIjoxNzQxNTIxNjI5fQ.ESDCqftdmTlWfutEVB0hFL7G-Cv-aGGFAyNB6SiuVX0","token_type":"bearer","refresh_token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJJQ1BTIiwiZXhwIjoxNzQxNTIzMTI5fQ.8w86-z85mD1YFY7sEbqRZTo__buyj0_zUj6UqvmPNa0","expires_in":300}

3.1.2. cardlessAuthorization

L'API cardlessAuthorization implémente le traitement la demande d'autorisation émise par le client depuis le terminal GAB de la banque.

API : cardlessAuthorization			
Client	PowerCARD (ICPS)		
Serveur	BIIC (MIDDLE CARDLESS)		
REQUEST			
Champ	**Type**	**M/O**	**Description**
countryCode	String	M	Code pays pour la transaction. Pour le Benin : **BJ**
currencyCode	String	M	Devise de la transaction. **XOF**
transType	String	M	Type de transaction. **C2W**
amount	Integer	M	Montant de la transaction.
tranRef	String	M	Reference de la transaction. **RRN** (Retrieval Reference Number)
cardAcceptorID	Integer	M	Numéro du code marchant du GAB (champ 42 du ISO8583)
terminalID	Integer	M	Numéro unique du terminal GAB (champ 41 du ISO8583)
payCode	Integer	M	Numéro du Voucher (Valeur entrée par le client)
Pin	String	M	Numéro de pin encrypté
RESPONSE			
transid	String	M	ID de la transaction fourni par MeekFi
requestId	String	M	Reference de la transaction (tranRef du REQUEST)
countryCode	String	M	Code pays pour la transaction. Pour le Benin : **BJ**
statusCode	String	M	**000** Success **011** Generic error **012** Request body invalid **013** Duplicate request id **014** Token is invalid **015** Invalid details 301 Authorization Failed 999 Default Error

Exemple :

```
curl http://ip:port/atm/v1/authorize_transaction
-d '{ "countryCode": "BJ", "currencyCode": "XOF", "tranRef":"2993696128621", "transType":"C2W", "amount": "4000", "tranRef":"2319629261", "cardAcceptorID": "10332230025810", "payCode":"1835416211","pin":"xahleh98329h32kgkgaksg" }'
-H 'Content-Type': application/json'
-H 'Authorization: Bearer QYRpOM614NbKb2JHtjqN.oL1o11xPSwEve6P3TP4Ud6NkQSyIvWPRhVPr5gdk'
```

```
{
"countryCode": "BJ",
"currencyCode": "XOF",
"transType":"C2W",
"amount": "4000",
"tranRef":"2319629261",
"cardAcceptorID": "10332230025810",
"terminalID": "111111111",
"payCode":"1835416211",
"pin":"xahleh98329h32kgkgaksg"
```

}
{
"statusCode": "000",
"statusMessage": "Successful",
"requestId": "2319629261",
"countryCode": "BEN",
"transid": "983748682"
}

3.1.3. acknowledge

L'API acknowledge permet à l'application de confirmer la réception d'une transaction en envoyant un identifiant de transaction (transaction_id) et un identifiant d'accusé de réception (acknowledgment_id), et retourne une réponse indiquant que l'accusé de réception a été reçu avec succès.

API : acknowledge			
Client	PowerCARD (ICPS)		
Serveur	BIIC (MIDDLE CARDLESS)		
REQUEST			
Champ	**Type**	**M/O**	**Description**
transaction_id	Integer	M	L'identifiant de la transaction
acknowledgment_id	String	M	L'identifiant de l'accusé de réception
RESPONSE			
statusCode	Integer	M	Retour du statut : 000
statusMessage	String	M	Le message de la réponse
transaction_id	Integer	M	L'identifiant de la transaction

Exemple :

Request : curl -X POST https://10.10.104.181:48003/cardless/atm/v1/acknowledge \

-H 'Content-Type: application/json' \

-H'Authorization:Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJJQ1BTIiwiZXhwIjoxNzQxNTIxNjI5fQ.ESDCqftdmTlWfutEVB0hFL7G-Cv-aGGFAyNB6SiuVX0' \

-d '{

"transaction_id": "983748682",

"acknowledgment_id": "ACK-a1b2c3d4e5f6"

}'

Response:

{

"statusCode": "000",

"statusMessage": "Acknowledgment received successfully",

"transaction_id": "983748682"

}

3.1.4. cardlessReverse

L'API cardlessReverse implémente le traitement de la demande d'annulation du a un incident technique (timeout, reseau, etc.) ou matériel (GAB, etc.) d'une autorisation qui n'a pas pu émise par le client depuis le terminal GAB de la banque.

API : cardlessReverse			
Client	PowerCARD (ICPS)		
Serveur	BIIC (MIDDLE CARDLESS)		
REQUEST			
Champ	**Type**	**M/O**	**Description**
countryCode	String	M	Code pays pour la transaction. Pour le Benin : **BJ**
currencyCode	String	M	Devise de la transaction. **XOF**
amount	Integer	M	Montant de la transaction.
tranRef	String	M	Reference de la transaction. **RRN** (Retrieval Reference Number)
transid	String	M	ID de la transaction fourni par MeekFi
payCode	Integer	M	Numéro du Voucher
pin	String	M	Numéro de pin encrypté
RESPONSE			
statusCode	String	M	**000** Success **011** Generic error **012** Request body invalid **013** Duplicate request id **014** Token is invalid **015** Invalid details 301 Authorization Failed 999 Default Error
ReqId	String	M	Reference de la transaction (tranRef du REQUEST)
transid	String	M	ID de la transaction fourni par MeekFi
statusMessage	String	M	Message de confirmation. Exemple : « **Reversal successful »**

Exemple :

```
curl http://ip:port/atm/v1/reverse_transaction
-d '{"countryCode": "BJ", "currencyCode": "XOF", "walletId":"2993696128621", "amount": "4000",
"tranRef":"2319629261", "transId":"12423691629692","payCode":"1835416211","pin":"xahleh98329h32kgkga"
}'
-H 'Content-Type': application/json'
-H 'Authorization: Bearer QYRpOM614NbKb2JHtjqN.oL1o11xPSwEve6P3TP4Ud6NkQSyIvWPRhVPr5gdk'
{
"countryCode": "BJ",
"currencyCode": "XOF",
"transType":"C2W",
"amount": "4000",
"tranRef":"2319629261",
"cardAcceptorID": "10332230025810",
"payCode":"1835416211",
"pin":"xahleh98329h32kgkgaksg"
}
{
```

"statusCode": "000",
"statusMessage": "Successful",
"requestId": "2319629261",
"countryCode": "BEN",
"transid" : "983748682"
}

3.2. Flux BIIC (Middle CardLess) / Amplitude

Voir les spécifications Sopra sur les API suivants :

- createFundReservation
- cancelFundReservation
- createTranfer

3.3. Flux BIIC (Middle CardLess) / MeekFi

Voir les spécifications dans le document « API Document » du partenaire MeekFi.