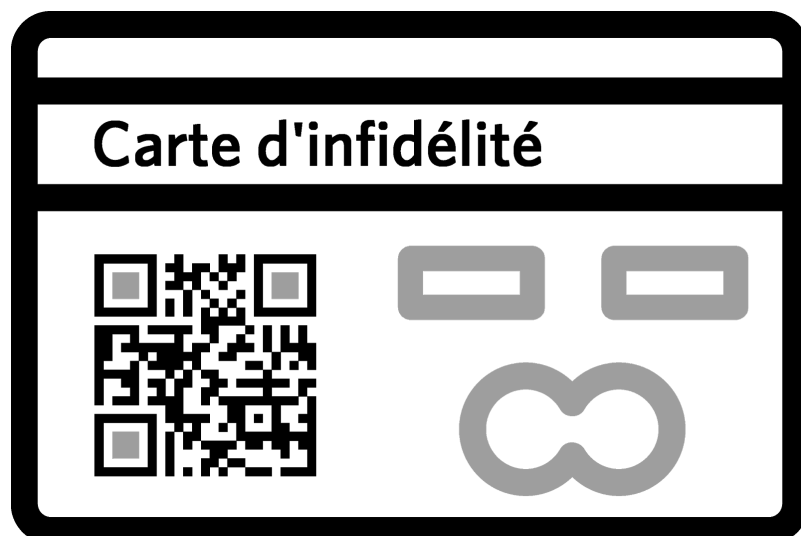


Rapport

Architecture

22 février 2017

Anthony LOROSCIO
Thomas MUNOZ
Loris FRIEDEL



Sommaire

1. Contexte	2
2. Vue fonctionnelle	4
2.1 Scénarios	2
2.2 Cas d'utilisations	7
2.3 Diagramme de composants	10
3. Vue développement	12
3.1 Diagramme de classe des objets du métier	12
3.2 Modèle relationnel de stockage	13
3.3 Explicitation du mapping objet-relationnel	15
4. Vue déploiement	16
Annexes	17

1. Contexte

Face aux centres commerciaux de grande envergure, les commerçants locaux ripostent en lançant un tout nouveau concept : **la carte d'infidélité**.

L'idée est d'inciter les consommateurs à acheter chez plusieurs commerçants d'une zone donnée. Divers avantages seront alors proposés en échange de points cumulés grâce aux achats (*infidèles*) chez les participants du programme. Le système ainsi développé devra être déployable dans plusieurs villes.

2. Vue fonctionnelle

2.1 Scénarios

Scénario 1 : achat d'un produit chez un commerçant affilié au programme

Acteur primaire : client

Acteur secondaire : commerçant

Précondition : le client possède une carte d'infidélité

Scénario primaire	Variantes
<ol style="list-style-type: none">1. Le commerçant lit la carte d'infidélité du client2. Le système crédite des points sur la carte en fonction du prix3. FIN	<ol style="list-style-type: none">3.b Le client achète des avantages avec ses points (aller en 3)

Postcondition : la carte est créditée d'un nombre de points proportionnel au prix payé chez le commerçant

Scénario 2 : adhérer au programme d'infidélité

Acteur primaire : commerçant

Acteur secondaire : la mairie

Scénario primaire	Variantes
<ol style="list-style-type: none">1. Le commerçant dépose sa demande d'inscription à la mairie2. La mairie crée son compte commerçant (inactif)3. Le commerçant alimente son catalogue d'avantages4. La mairie active son compte commerçant5. FIN	

Postcondition : le commerçant adhère au programme et possède un compte actif

Scénario 3 : diffuser les offres promotionnelles

Acteur primaire : commerçant

Acteur secondaire : mairie, client

Scénario primaire	Variantes
<ol style="list-style-type: none">1. Le commerçant effectue une demande de diffusion d'offre auprès de la mairie2. La mairie récupère les adresses de contact des clients concernés3. FIN	

Note : Nous avons déterminé qu'il n'était pas de notre responsabilité d'envoyer les mails aux clients concernés, nous renverrons donc la liste des adresses email des clients concernés par l'offre. Cependant, il sera possible d'employer un service externe qui aura pour responsabilité l'envoi des mails à la liste de clients renvoyée par notre service.

Scénario 4 : s'inscrire au service d'infidélité

Acteur primaire : Client non inscrit

Acteur secondaire : la mairie

Précondition : le client ne possède pas de carte d'infidélité

Scénario primaire	Variantes
<ol style="list-style-type: none">1. Le client s'inscrit au service d'infidélité auprès d'un commerçant2. Le commerçant donne une carte d'infidélité au client3. FIN	<ol style="list-style-type: none">1.b Le client s'inscrit au service d'infidélité chez lui2.b Le client récupère une carte grâce à la mairie (aller en 3)

Postcondition : le client possède une carte d'infidélité

Scénario 5 : se connecter au service

Acteur primaire : Client inscrit

Acteur secondaire : Cerberus

Précondition : le client n'est pas connecté au service

Scénario primaire	Variantes
<ol style="list-style-type: none">1. Le client communique ses identifiants au service d'infidélité2. Le service infidélité effectue une demande d'authentification au service Cerberus3. L'authentification réussit4. FIN	<ol style="list-style-type: none">3.b L'authentification échoue (aller en 1)

Postcondition : le client est connecté au service

Scénario 6 : activer l'avantage de stationnement

Acteur primaire : client

Précondition : le client est connecté

Scénario primaire	Variantes
<ol style="list-style-type: none">1. Le client saisit le numéro de sa plaque d'immatriculation2. Le client active l'avantage3. FIN	<ol style="list-style-type: none">2.b Le client n'est pas abonné à l'avantage récurrent du parking ou a déjà activé cet avantage aujourd'hui (aller en 3)

Postcondition : l'avantage parking est activé ou l'activation a échoué

Scénario 7 : vérifier validité stationnement

Acteur primaire : policier

Précondition : le policier est connecté au service en tant qu'utilisateur spécial

Scénario primaire	Variantes
<ol style="list-style-type: none">1. Le policier saisit le numéro de la plaque2. Le policier obtient le détail de l'avantage du stationnement ainsi que l'heure à laquelle celui-ci a été activé3. FIN	<ol style="list-style-type: none">2.b Aucune information sur la plaque n'est trouvée (<i>le conducteur a oublié d'activer son avantage ou c'est quelqu'un qui n'a pas payé le parking</i>) (aller en 3)

Scénario 8 : profiter d'un avantage

Acteur primaire : client

Acteur secondaire : partenaire commerçant, partenaire non commerçant

Précondition : le client n'a pas d'avantage récurrent actif et est connecté au service d'infidélité

Scénario primaire	Variantes
<ol style="list-style-type: none">1. Le client achète un avantage avant son passage en caisse2. La carte fidélité du client est débité du prix de l'avantage3. Le client active l'avantage4. FIN	<ol style="list-style-type: none">1.b Le client achète un avantage auprès du partenaire lors de son passage en caisse2.b Le client n'obtient pas le bonus car il n'a pas assez de points (aller en 4)

Scénario 9 : abonnement à bonus VUP récurrent

L'**acteur primaire** est le client VUP qui possède une carte d'infidélité et qui est connecté au service. **Premièrement**, le client s'abonne à un avantage récurrent, **ensuite** l'abonnement est, **soit** confirmé **soit** échoué (car il est déjà abonné à un avantage récurrent).

Scénario 10 : désabonnement à un avantage VUP récurrent

L'**acteur primaire** est le client VUP qui est connecté et qui est abonné à un avantage récurrent. **Premièrement**, le client sélectionne son avantage récurrent actuel, **puis** le client se désabonne de cet avantage.

Scénario 11 : approvisionner sa carte d'infidélité

Acteur primaire : client

Précondition : le client est authentifié sur la plateforme

Scénario primaire	Variante
<ol style="list-style-type: none">1. Le client saisit ses informations bancaires2. Le client saisit la somme souhaitée3. L'organisme bancaire valide la transaction4. Le service d'infidélité crédite la carte du client de la somme5. FIN	<ol style="list-style-type: none">3.a La transaction n'est pas validée (aller en 1)

Scénario 12 : payer avec sa carte d'infidélité

Acteur primaire : client

Acteur secondaire : commerçant

Scénario primaire	Variante
<ol style="list-style-type: none">1. Le commerçant lit la carte2. La transaction est validée3. Le montant de l'achat est débité de la carte du client.4. FIN	<ol style="list-style-type: none">2.b La transaction est rejetée (aller en 4)

Scénario 13 : alimenter son catalogue

L'**acteur primaire** est le superviseur, et il est connecté. Le superviseur peut ajouter, mettre à jour ou supprimer des avantages du catalogue.

Scénario 14 : récupérer des KPI globaux

L'acteur primaire est la mairie et il est connecté.

Le superviseur demande des KPI sur les habitudes de consommation des utilisateurs. Le système génère les KPI et les communique au superviseur.

Scénario 15 : récupérer des KPI par commerce

Acteur primaire : commerçant

Précondition : être authentifié

Scénario primaire	Variantes
<ol style="list-style-type: none">1. Le commerçant demande des KPI sur l'utilisation du service pour son commerce2. FIN	<ol style="list-style-type: none">1b. Le commerçant demande des KPI sur l'utilisation particulière des autres partenaires du programme

2.2 Cas d'utilisations

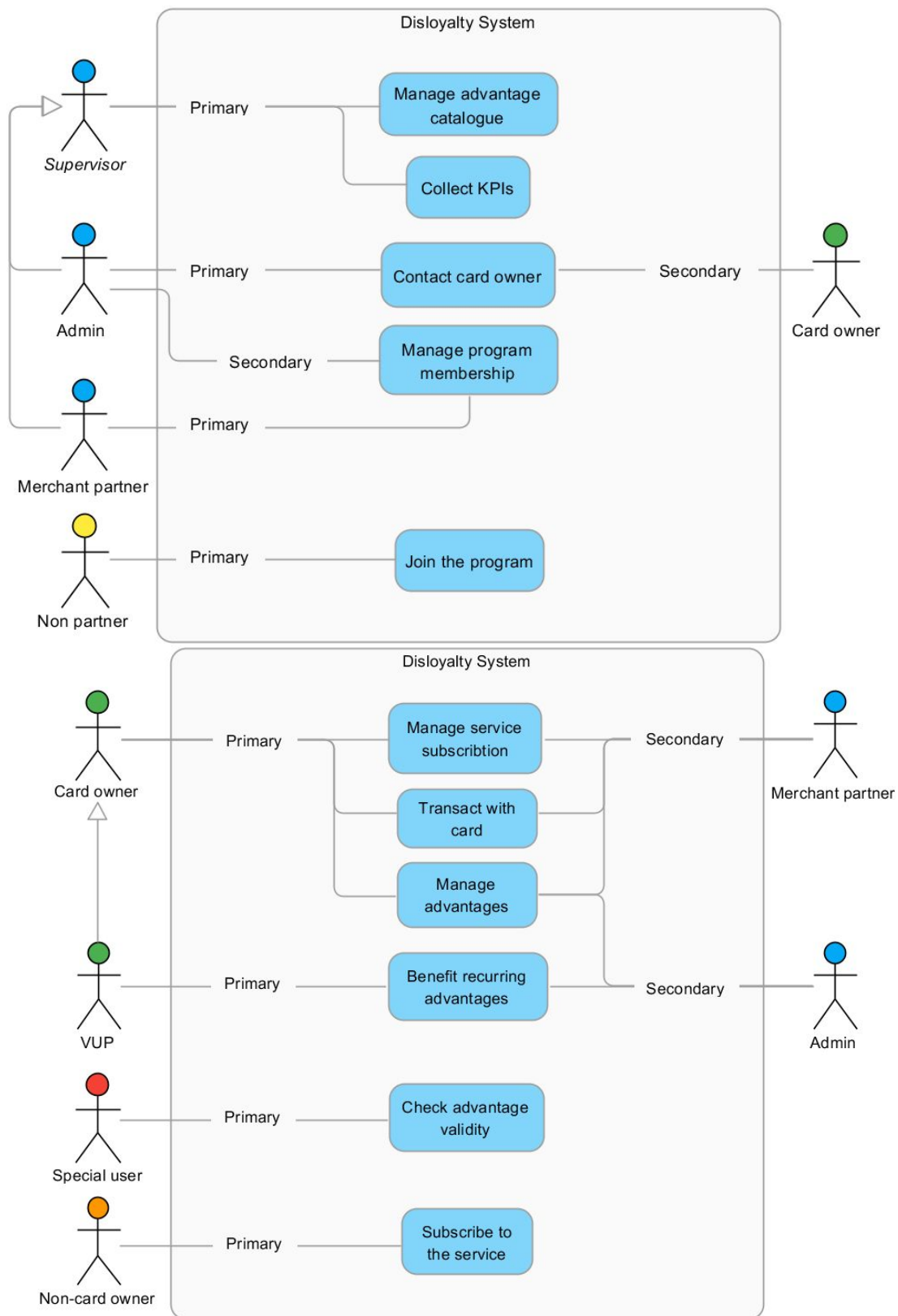


Figure 1 - Diagramme de cas d'utilisation de haut niveau

Ce diagramme de haut niveau représente les principaux cas d'utilisation et les acteurs qui y sont associés.

Le superviseur peut gérer le catalogue d'avantages et récupérer des informations relatives aux KPI.

Le partenaire marchand correspond aux commerçants, ils ont donc accès aux mêmes utilisations que le superviseur mais peuvent aussi gérer leur propre adhésion.

L'administrateur correspond à l'administrateur système de la mairie, il dispose lui aussi des mêmes capacités que le superviseur mais peut aussi contacter les membres possédants une carte. L'acteur **non partenaire** représente les commerçants non adhérents au programme, cet acteur ne peut que joindre le programme pour s'affilier.

Le détenteur d'une carte représente le client, il peut donc gérer son profile, effectuer des transactions et gérer ses avantages. L'acteur VUP correspond au client qui a obtenu le statut éponyme, il peut alors accéder au catalogue VUP ou profiter d'un avantage journalier.

L'utilisateur spécial représente concrètement le policier, mais il représente aussi tous les utilisateurs qui voudront vérifier la validité d'un avantage.

Enfin, **le non détenteur d'une carte** correspond au client qui n'a pas encore de carte, il ne peut donc que s'inscrire.

Nous détaillerons par la suite les cas d'utilisations spécifiques, en omettant ceux trop succincts.

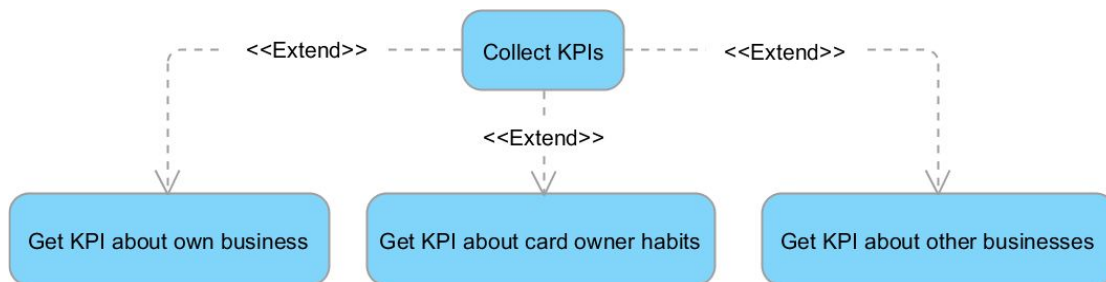


Figure 2 : Cas d'utilisation de collect KPIs

Ce cas d'utilisation représente les actions correspondantes à la collecte de KPIs. Un commerçant peut récolter les KPI relatifs à son propre commerce, mais aussi ceux d'autres commerces. Il peut aussi récupérer des KPI relatifs aux habitudes des clients.

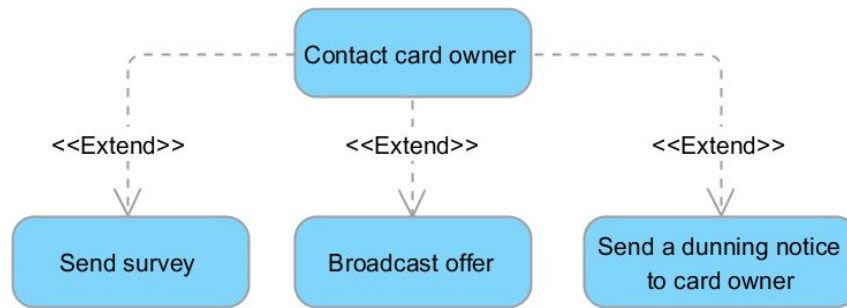


Figure 3 : Contacter les membres du service

L'administrateur peut contacter les membres du service en leur envoyant un sondage, en diffusant une offre ou en relançant les consommateurs qui ont perdu leur statut de VUP.

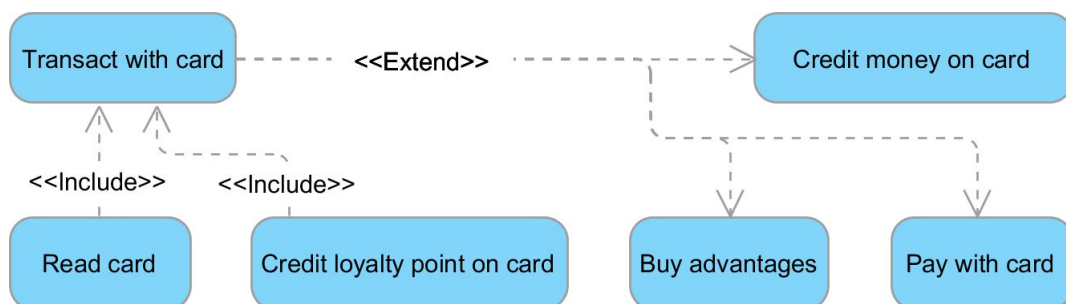


Figure 4 : Effectuer une transaction avec la carte

Lorsqu'un abonné effectue un achat dans une boutique qui adhère au programme, il faut que sa carte soit lue et que les points correspondants soient crédités sur son compte. Il peut aussi acheter des avantages ou payer avec sa carte d'infidélité.

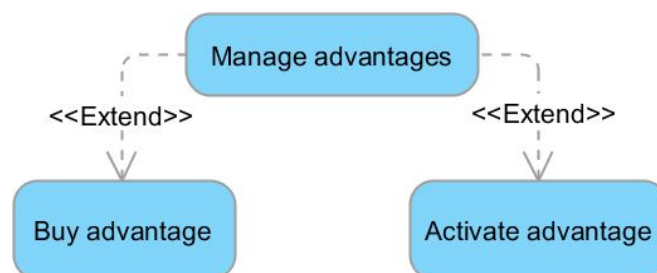


Figure 5 : gérer ses avantages

Un abonné peut gérer ses avantages. Il doit pour cela acheter des avantages, mais il peut aussi activer un avantage (activer le stationnement gratuit par exemple).

2.3 Diagramme de composants

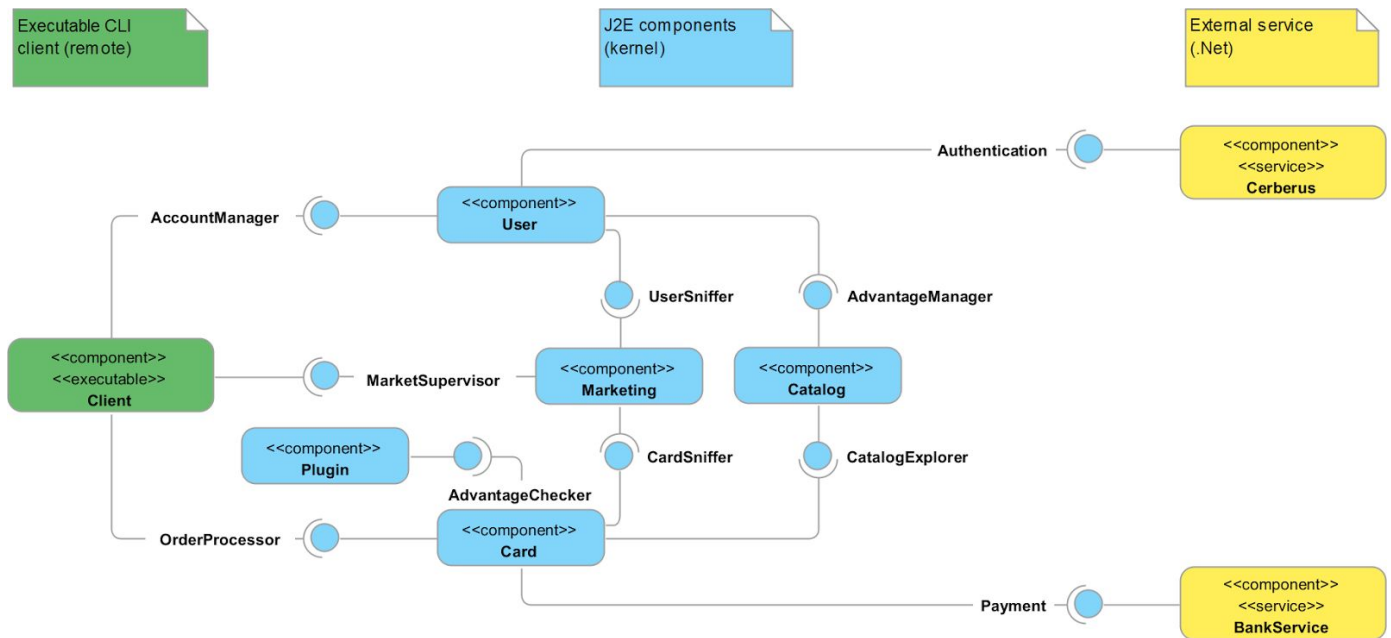


Figure Y-Wing : diagramme de composants de notre système d'infidélité

Interfaces exposées

AccountManager

```
void registerMerchant(Merchant)
void activate(Merchant)
void registerClient(Client)
void link(Client, Card)
void authenticate(User)
void updateClient(Client)
void updateMerchant(Merchant)

void addAdvantage(Supervisor, Advantage)
void updateAdvantage(Advantage)
void deleteAdvantage(Advantage)
```

UserSniffer

```
ClientEvent[] retrieveClientEvent(Date, Date)
```

CardSniffer

```
Transaction[] findTransaction(Date, Date)
```

Authentication

```
bool authenticate(User)
```

OrderProcessor

```
creditPointFor(Card, Supervisor, amount:int) // amount en euro
bool buyAdvantage(Card, Advantage, amount:int) // amount en point d'infidélité
bool buy(Merchant, Card, amount:int) // amount en euro
void activate(Card, Advantage)
void activateRecurring(Card, Advantage)
```

Payment

```
bool supply(bankCardNumber:String, validity:String, cvv:String, amount:int) //
amount en euro
```

MarketSupervisor

```
Client[] findByMerchant(Supervisor[], Date, Date)
Client[] findLostVUP(Date, Date)

Transaction[] getGivenAdvantages(Supervisor[], begin:Date, end:Date)
Transaction[] getGrossPurchases(Supervisor[], begin:Date, end:Date)
```

AdvantageManager

```
void add(Advantage)
void delete(Advantage)
void update(Advantage)
```

AdvantageChecker

```
AdvantageData getAdvantageData(Advantage)
```

CatalogExplorer

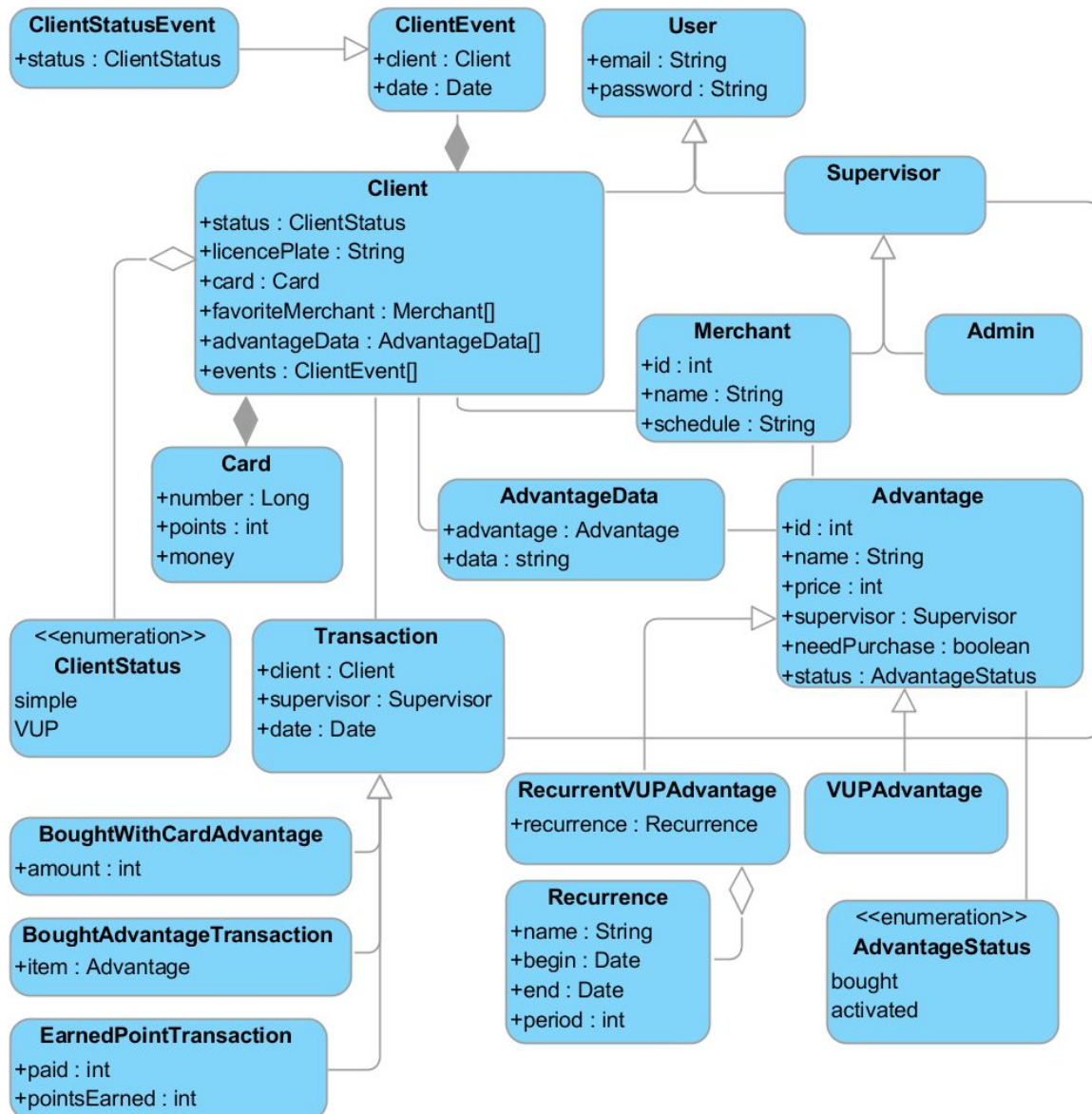
```
Advantage[] getCatalog(Supervisor)
```

BalanceUpdater

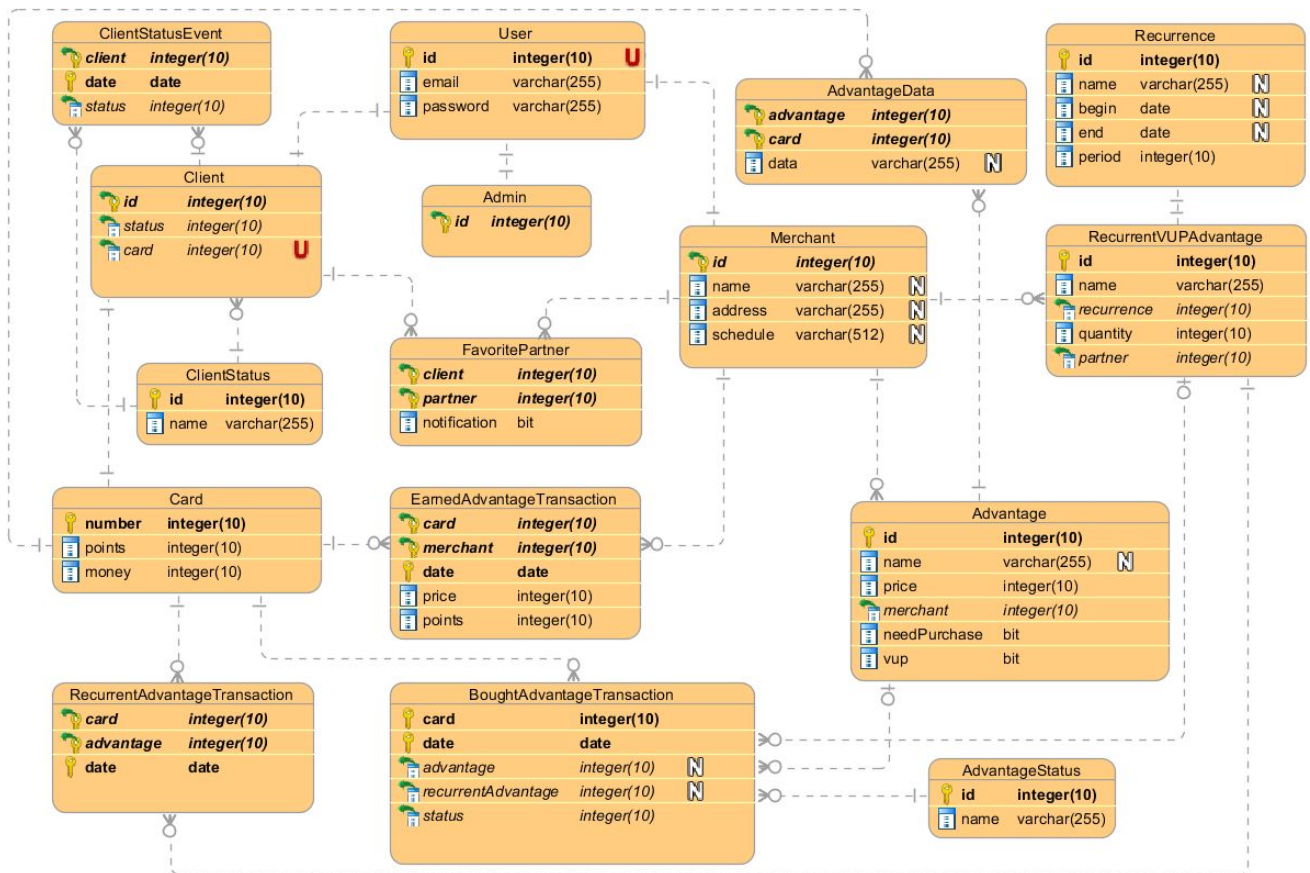
```
void debitPoint(Card, amount:int)
void debitMoney(Card, amount:int)
void creditPoint(Card, amount:int)
void creditMoney(Card, amount:int)
```

3. Vue développement

3.1 Diagramme de classe des objets du métier



3.2 Modèle relationnel de stockage



Ce modèle relationnel, a comme avantage de :

- Simplifier les échanges essentiels à l'application (utilisation de la carte, gestion d'avantages)
- Séparer les différentes formes d'utilisateurs du service dans des entités différentes afin qu'ils puissent interagir avec les fonctionnalités qui leurs sont propres (gestion d'avantages pour des commerçant, rechargement de carte pour un client par exemple)
- Il permet l'ajout de "plugins", éléments associés à un avantage particulier qui nécessite certaines données (c'est le cas des 20 minutes de parking offerts qui nécessitent de connaître le numéro de plaque d'immatriculation du client)

Cependant, pour le moment, aucune "hiérarchie" n'est possible pour les commerçants, cela rend donc difficile la définition d'avantages pour plusieurs commerçants (dans le cas d'une chaîne de magasins qui souhaitent certains avantages globaux). Pour proposer cette fonctionnalité, il faudrait ajouter une relation qui associe un avantage avec de multiples commerçants. Ainsi, il sera possible de proposer un même avantage pour plusieurs commerçants.

Les requêtes SQL ayant du sens vis à vis des fonctionnalités demandées dans le briefing produit sont les suivantes :

Inscription client

```
INSERT INTO user (email, password) VALUES ("client@mail.com", "a16358b76266075e");
INSERT INTO client (id, card) VALUES (1, 1337);
```

Ajout d'un avantage au catalogue

```
INSERT INTO advantage (name, price, partner, needPurchase, vup) VALUES ("pain au chocolat", 100, 5, 1, 0);
```

Achat d'un avantage du catalogue par un client

```
INSERT INTO BoughtAdvantageTransaction (card, merchant, advantage) VALUES (1, 5, 18);
```

Achat d'un produit chez un commerçant (réception de point sur son compte)

```
INSERT INTO EarnedAdvantageTransaction (card, merchant, price, points) VALUES (1337, 5, 10, 100);
UPDATE card SET points = (SELECT points FROM card WHERE id = 1337) + 10 WHERE id = 1337;
```

Chargement d'argent sur la carte d'un client

```
UPDATE card SET money = (SELECT money FROM card WHERE id = 1337) + 5 WHERE id = 1337;
```

Passage au status VUP d'un client

```
UPDATE client SET status = 2 WHERE card = 1337;
```

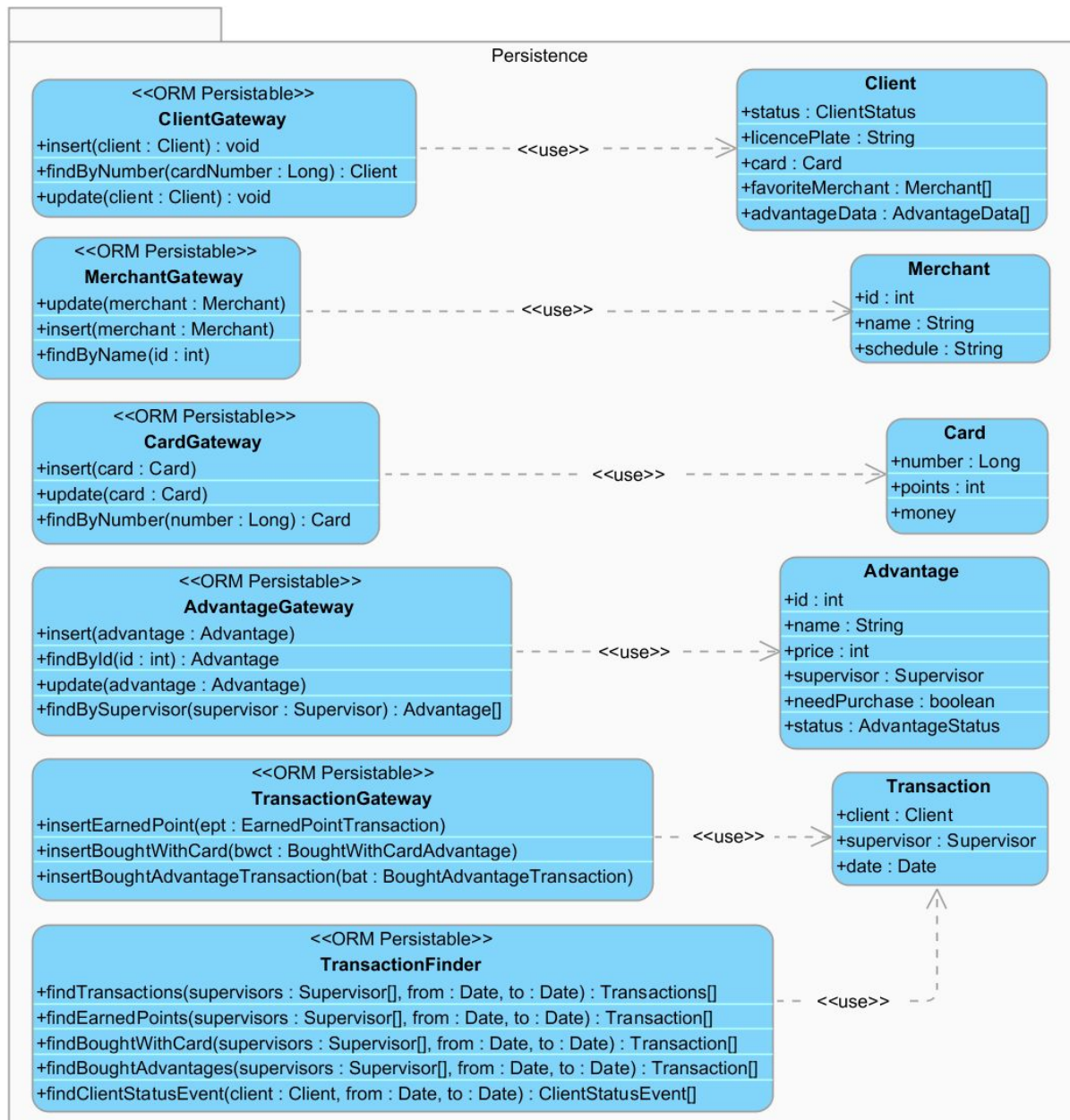
Choix d'un avantage récurrent pour un client VUP

```
INSERT INTO RecurrentAdvantageTransaction (card, advantage) VALUES (1337, 42);
```

Récupération des adresses mail des clients client d'un partenaire

```
SELECT email
FROM user
JOIN client ON user.id = client.id
WHERE client.card IN (SELECT card FROM EarnedAdvantageTransaction WHERE merchant = 42)
OR client.card IN (SELECT card FROM BoughtAdvantageTransaction WHERE merchant = 42);
```


3.3 Explicitation du mapping objet-relationnel



Pour le mapping objet-relationnel, nous avons fait le choix d'utiliser le pattern **Table Data Gateway** pour le mapping des classes *Client*, *Merchant*, *Card* et *Advantage* afin de réunir au même endroit les différentes requêtes relatives à chaque objet métier. Dans ces cas-là, le fait d'utiliser un Finder afin de séparer les usages aurait été peu pertinent.

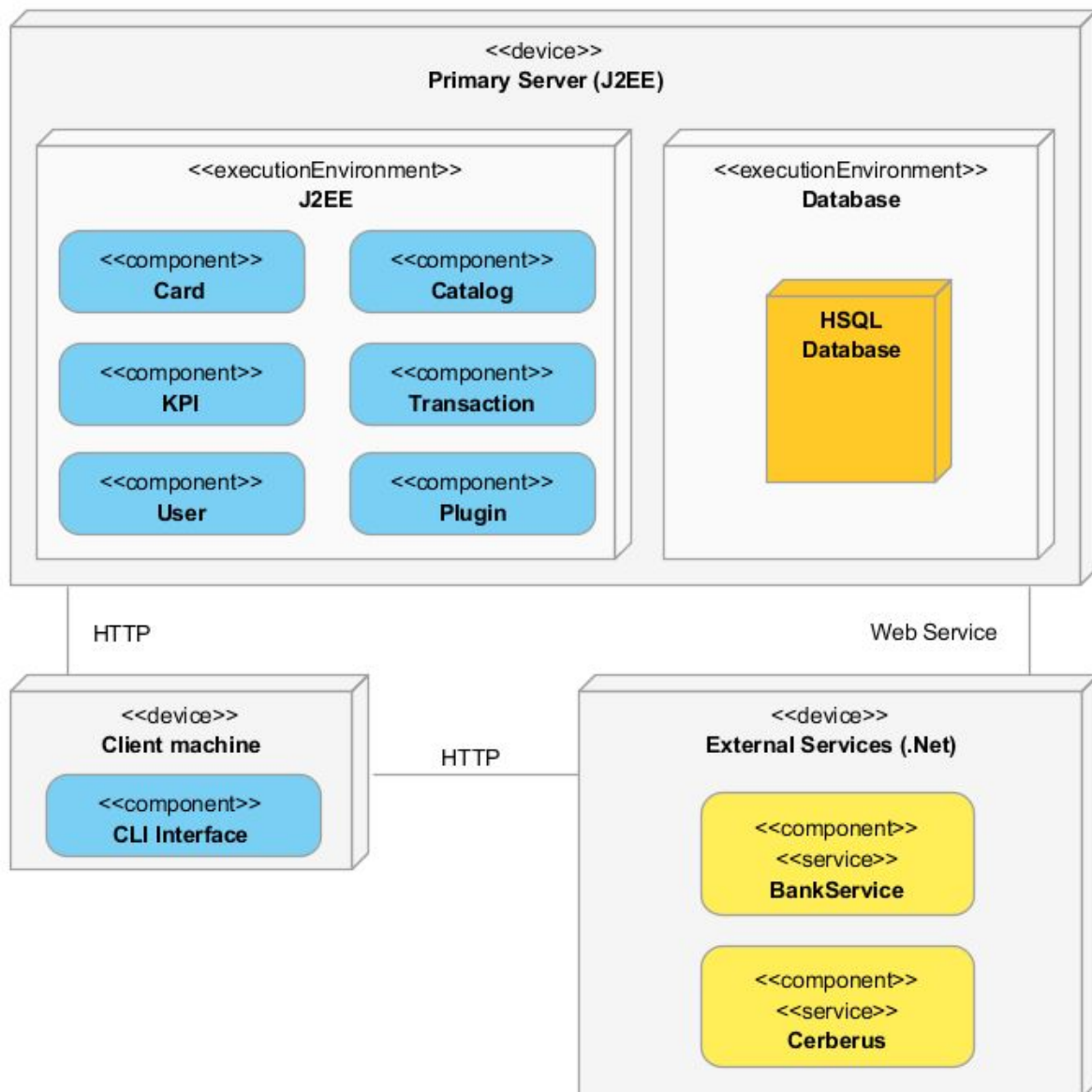
En effet, les opérations possibles sur ces classes sont peu sujettes au changement, elles correspondent au **CRU(D)** (**C**reate, **R**ead, **U**date, **D**ele~~t~~e).

En revanche, pour la classe *Transaction*, nous avons fait le choix de séparer le mapping en deux classes **Gateway** et **Finder**, cela afin de séparer les responsabilités de celles-ci.

La classe **TransactionGateway** a pour responsabilité l'insertion en base de données des transactions. La classe **TransactionFinder** a pour responsabilité d'agréger les différentes transactions selon certains critères.

Dans la pratique, **TransactionGateway** sera utilisé lors de chaque transaction définie dans notre système (achat chez un commerçant, achat d'avantage, etc.), **TransactionFinder**, en revanche, sera employé lors de la récupération des informations nécessaires au calcul des **KPI** (pour connaître le volume de vente d'un commerçant par exemple).

4. Vue déploiement



Annexes

