

Automatisation du réseau AMAP



FreeMAP

Médéric HURIER
Omar EDDASSER

★ Sommaire

1. Présentation du sujet
2. Objectifs attendus
3. Module client
4. Module producteur
5. Module adhérent
6. Choix d'implantation
7. Choix techniques
8. Conclusion

★ Présentation du sujet

AMAP = Association pour le Maintien d'une
Agriculture Paysanne

1. Mise en relation Client/Producteur
2. Composition du panier
3. Historique

★ Objectifs attendus

Modernisation & Automatisation

- Enregistrement d'abonnement (client)
- Envoi de propositions (producteur)
- Composition assistée (adhérent)

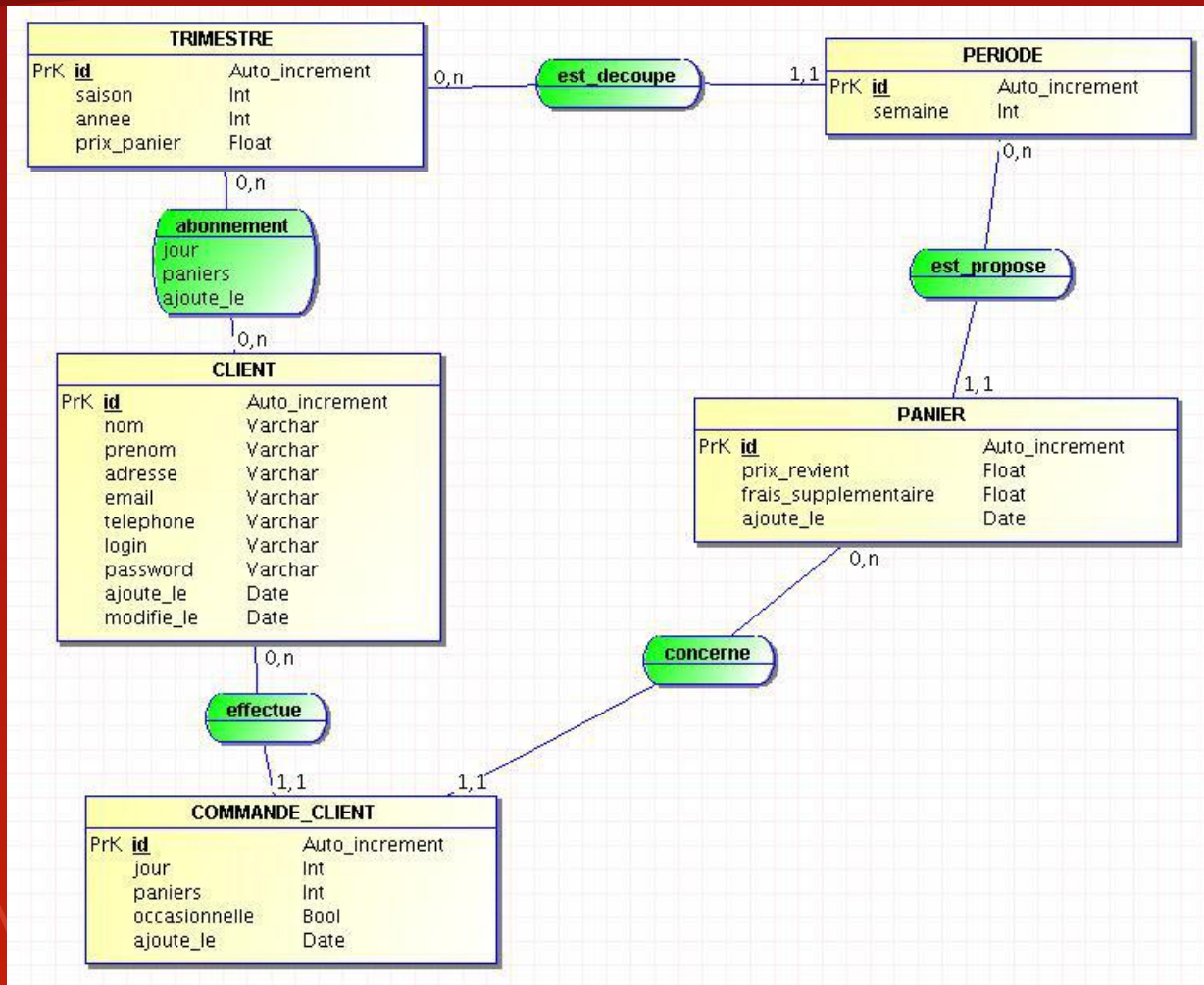
➡ réduire les interactions entre acteurs

➡ augmenter l'autonomie des adhérents

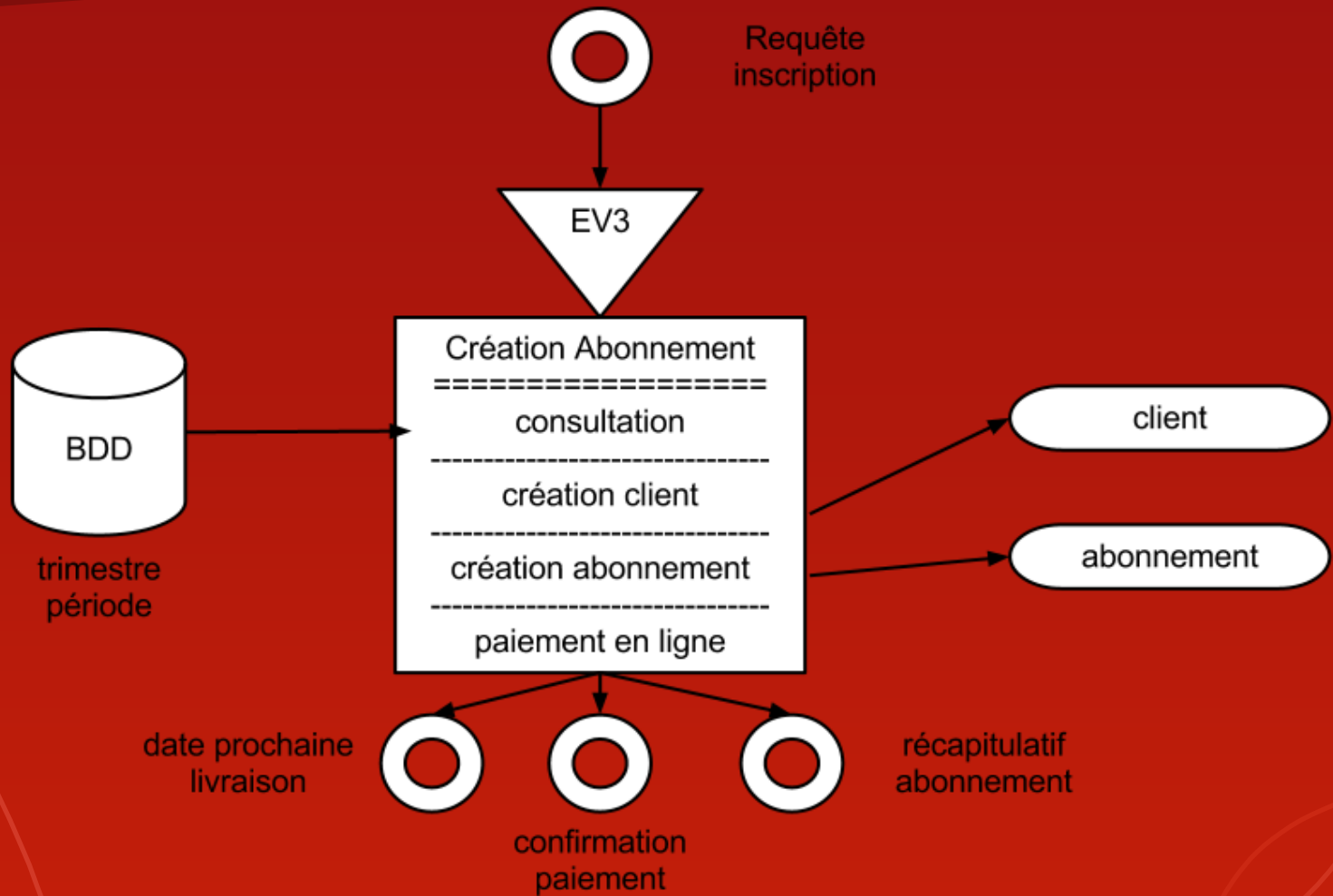


Module Client

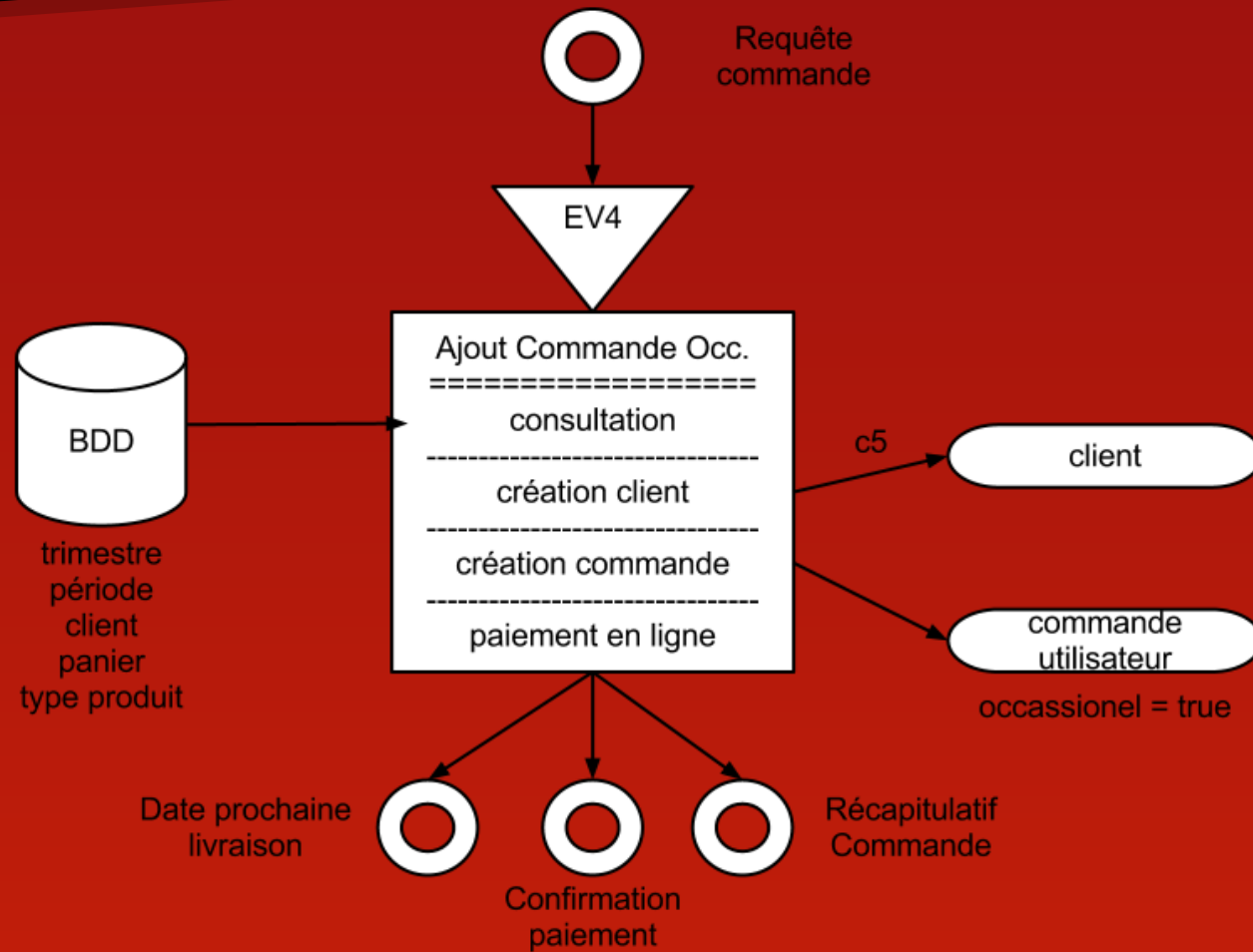
✦ Données - Client



✦ Traitement 1 - Client



✦ Traitement 2 - Client

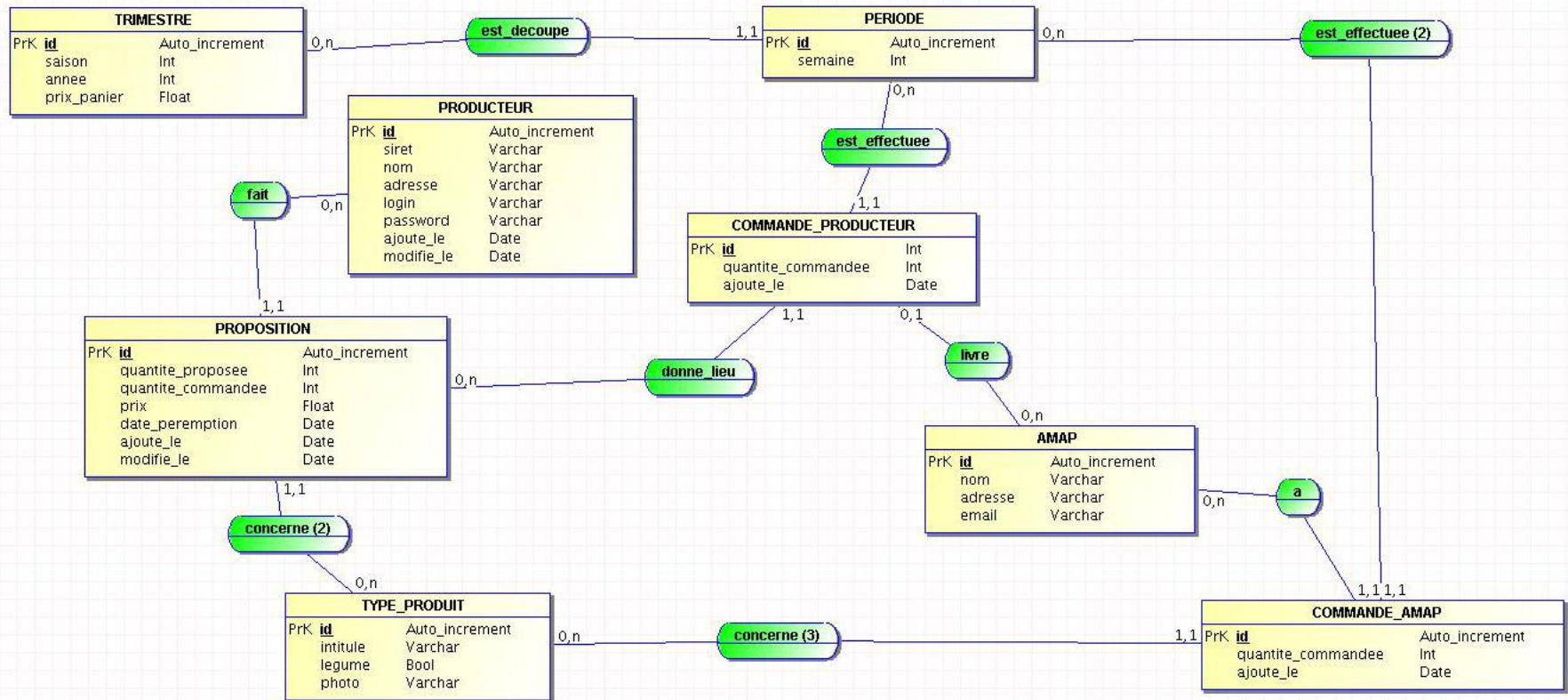


c5: le client n'existe pas

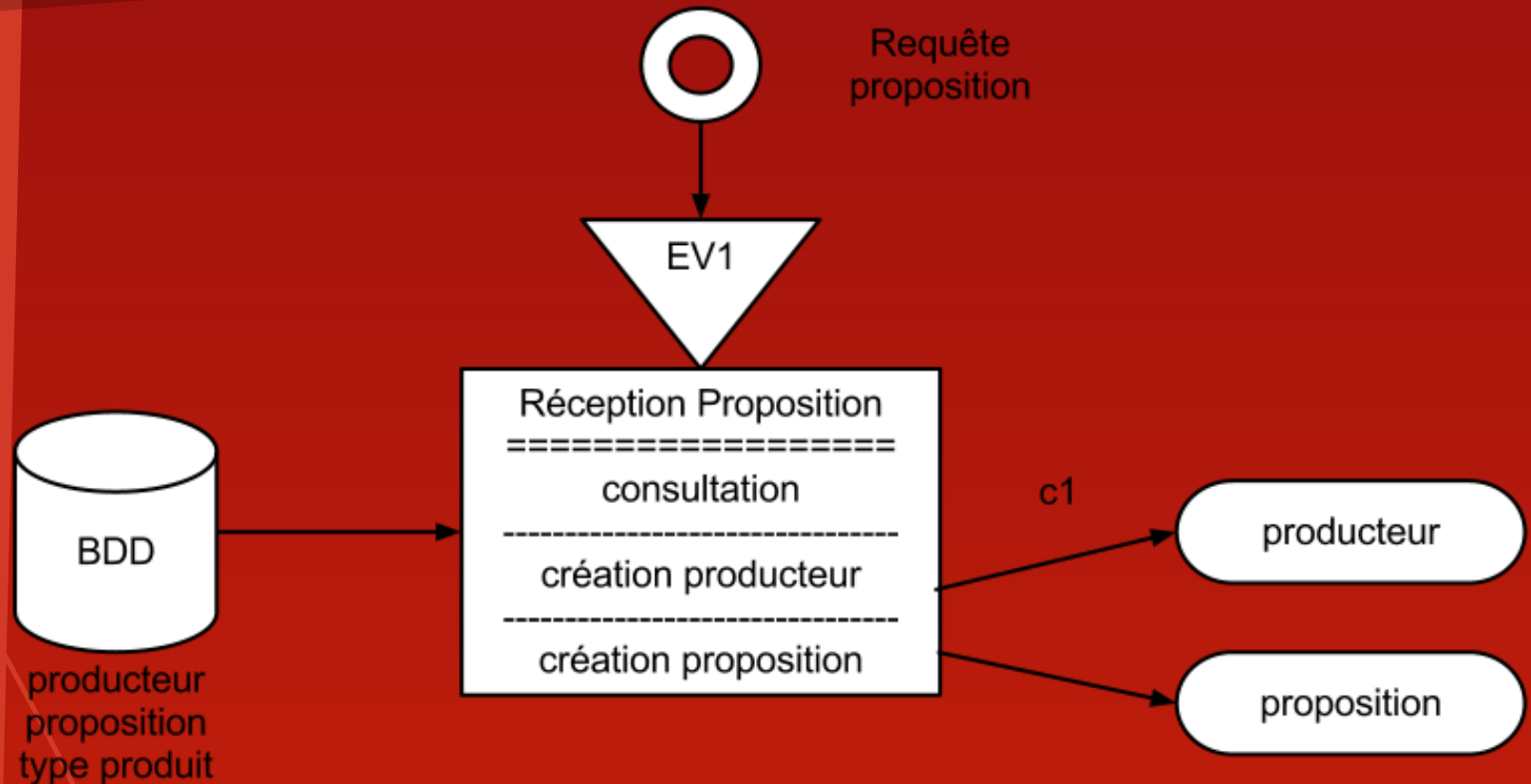


Module Producteur

Données - Producteur



★ Traitement 1 - Producteur

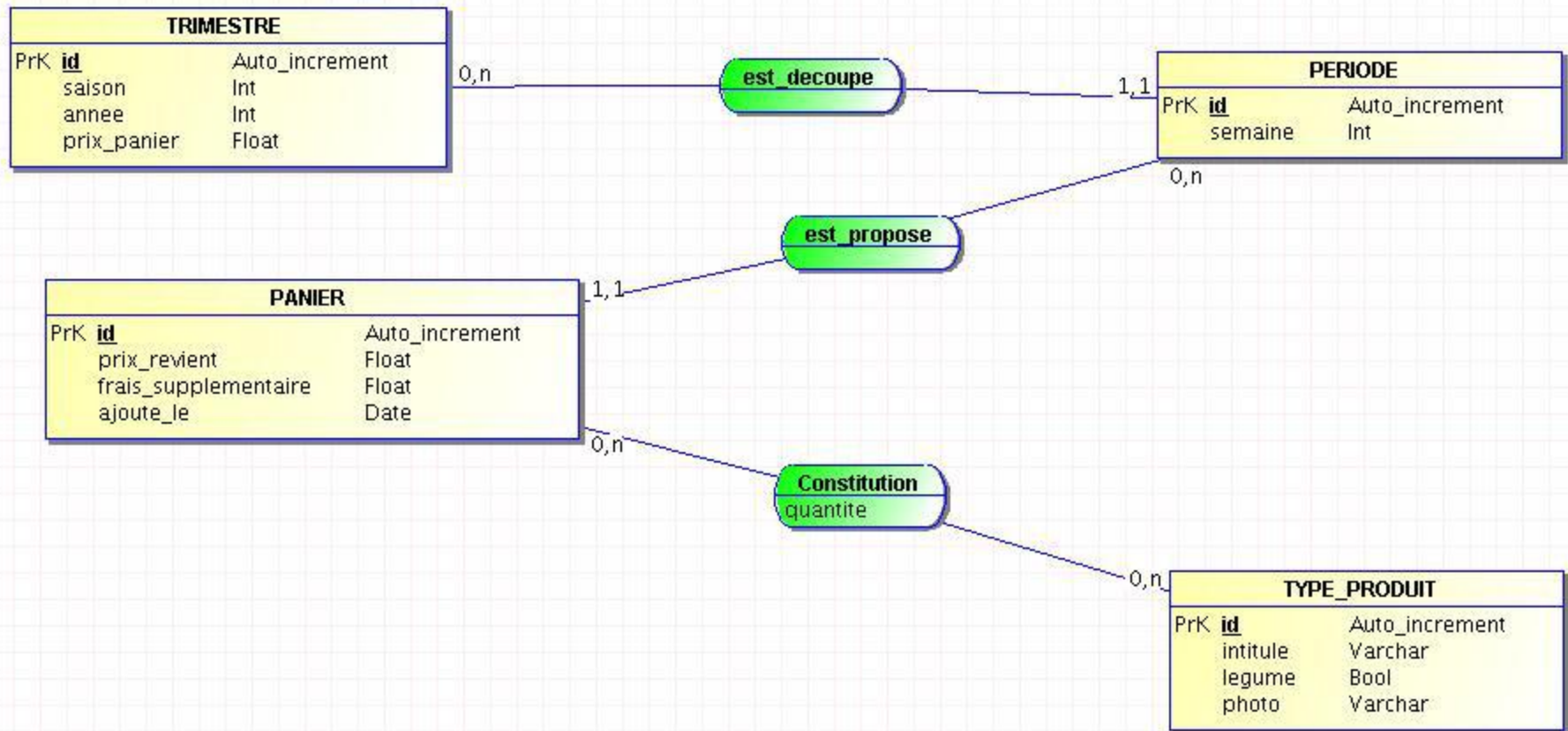


c1: le producteur n'existe pas

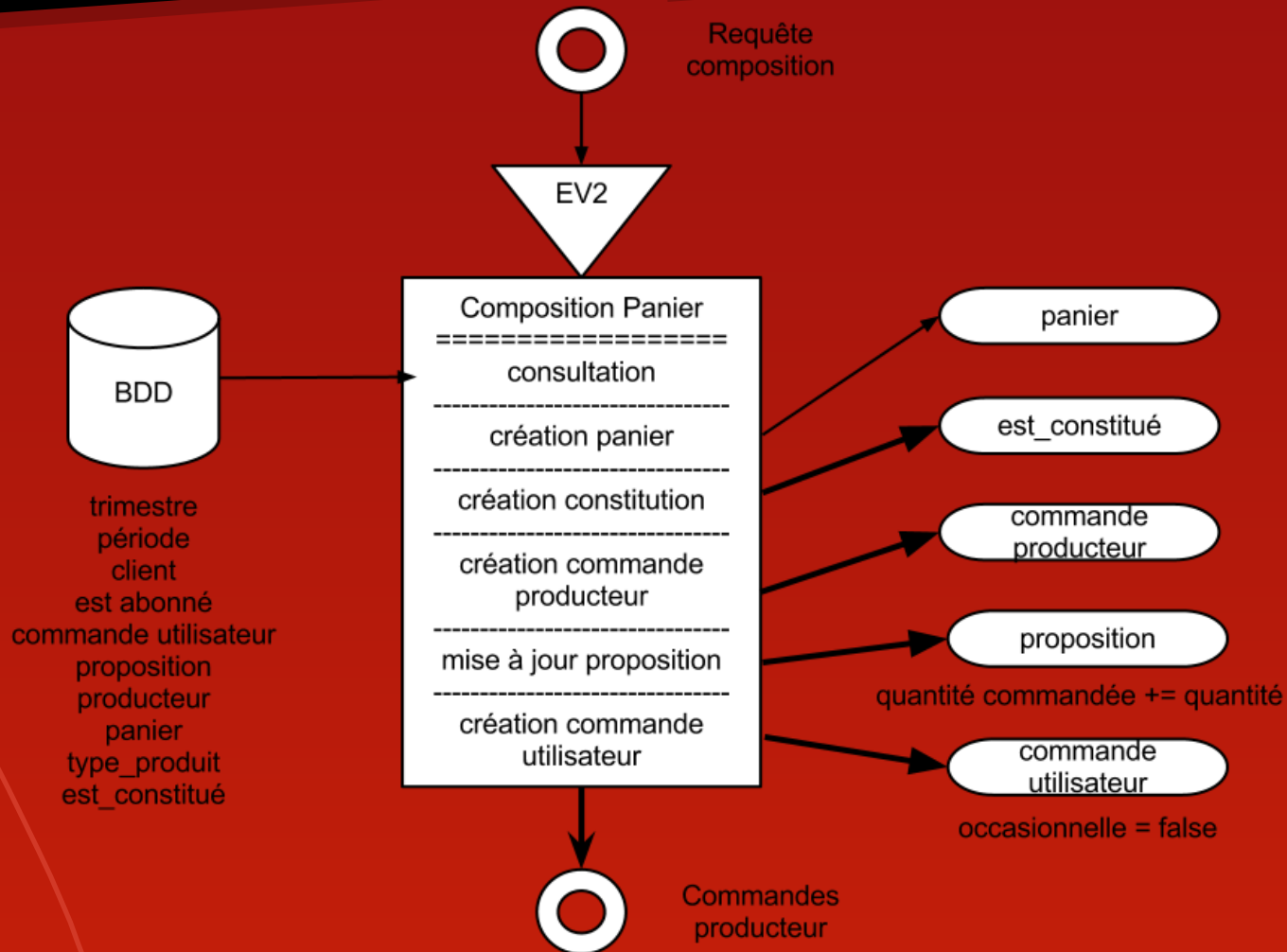


Module Adh rent

Données - Adhérent



✦ Traitement 1 - Adhèrent



★ Échange entre AMAP ?

2 solutions envisagées:

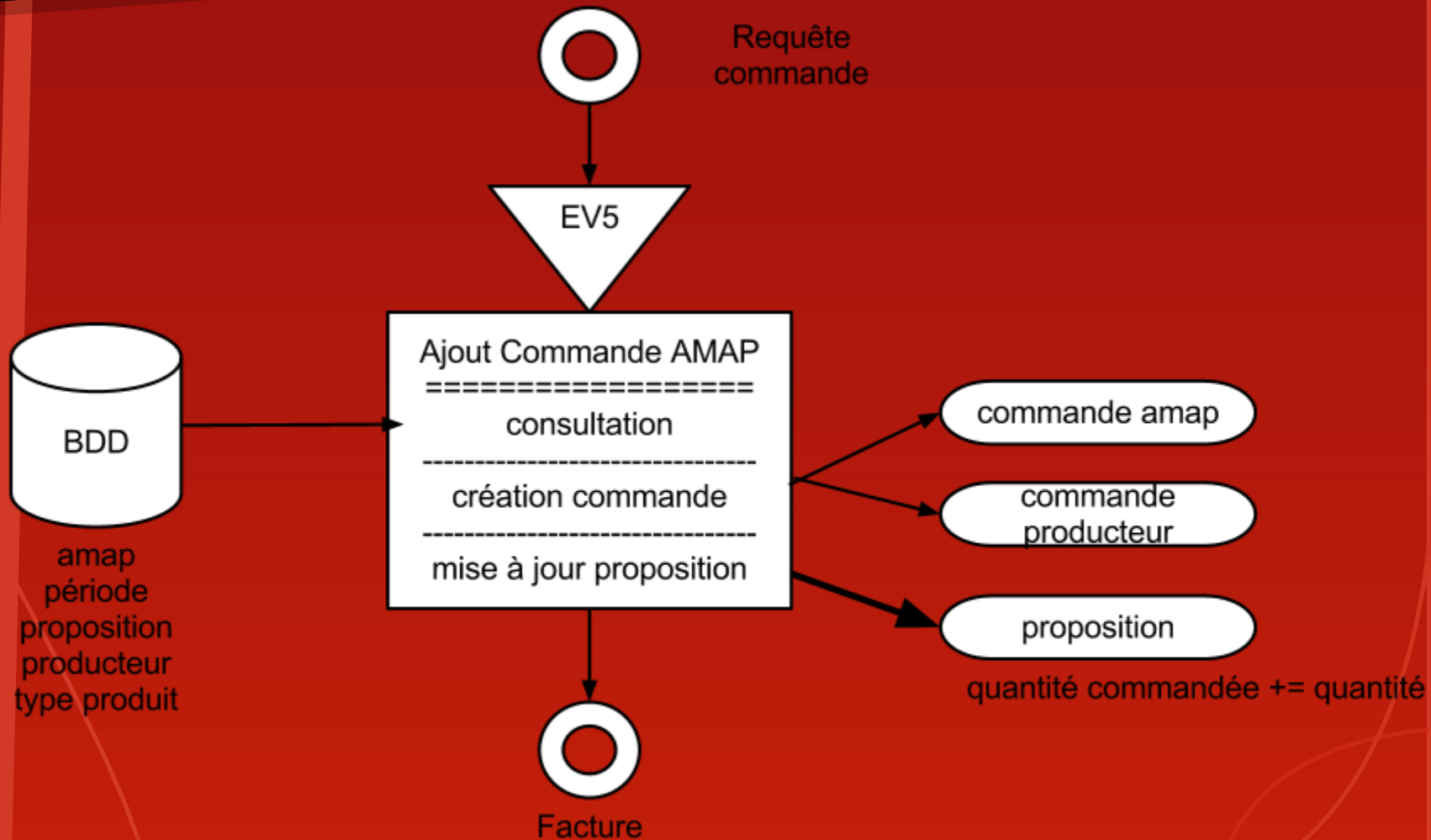
- utiliser les produits après composition
- renseigner manuellement le surplus

MAIS:

- étreinte fatale entre AMAP
- beaucoup d'actions supplémentaires

Solution plus simple et rapide ?

✦ Traitement - Commande AMAP





Méthodes & Techniques

★ Technologies

Cadriciel: Django

Langage: Python

Base de données: PostgreSQL

Interface client: HTML5 / CSS3

Scripts clients: jQuery, jQuery Mobile

Partage de code: Mercurial, Redmine

★ Déclencheurs

3 déclencheurs

- Après la création d'un trimestre:
 - ajoute toutes les périodes associées (13)
- Après la création d'une période
 - ajoute un panier vide associée
- Après la création d'une com. Prod.
 - vérifie que la proposition existe
 - vérifie que quantité dispo. > quantité com.
 - met à jour la quantité com. dans proposition

✦ From MCD

CLIENT		
PrK <u>id</u>		Auto_increment
nom		Varchar
prenom		Varchar
adresse		Varchar
email		Varchar
telephone		Varchar
login		Varchar
password		Varchar
ajoute_le		Date
modifie_le		Date

PANIER		
PrK <u>id</u>		Auto_increment
prix_revient		Float
frais_supplementaire		Float
ajoute_le		Date

Constitution
quantite

TYPE_PRODUIT		
PrK <u>id</u>		Auto_increment
intitule		Varchar
legume		Bool
photo		Varchar

★ To MPD

CLIENT		
PrK	<u>id</u>	Auto_increment
	nom	Varchar
	prenom	Varchar
	adresse	Varchar
	email	Varchar
	telephone	Varchar
	login	Varchar
	ajoute_le	Date
	modifie_le	Date

Constitution		
PrK	<u>id</u>	Auto_increment
	quantite	Int
FrK	<u>PANIER id</u>	Int
FrK	<u>TYPE PRODUIT id</u>	Int



Conclusion

★ Difficultés rencontrées

Gestion des logins

- Initial: login/password par acteur
- Final: utilisation des ACL Django

Gestion des clés étrangères

- Initial: clés étrangères entre bases
- Final: gestion des contraintes coté app

★ Bilan du projet

- Originalité de la production
 - utilisation de cadriciels
 - base de données open-source
- Implémentation fidèle à la modélisation
- Utilisation partielle de la démarche intégratrice

★ Démonstration

1. Les deux interfaces (util/admin)
2. Création d'un abonnement
3. Ajout d'une proposition producteur
4. Composition du panier