

# *DEMONEY : Modélisation en B et vérification de propriétés*

Nicolas Stouls

([nicolas.stouls@imag.fr](mailto:nicolas.stouls@imag.fr))

Laboratoire LSR de Grenoble / IMAG

*Travaux financés par une bourse CNRS / STMicroelectronics*

Le 6 mai 2004

# Objectifs

- ◆ Expression et vérification de propriétés de sécurité
- ◆ Choix méthodologiques :
  - ◆ Construction par raffinement
  - ◆ Utiliser au maximum la preuve
    - ◆ Invariants, préconditions, etc.
  - ◆ Compléter par vérification d'un automate des comportements autorisés
    - ◆ Génération automatique d'un graphe comportemental
      - ◆ Outil *GénéSyst*
    - ◆ Comparaison automatique d'un graphe comportemental avec une propriété
      - ◆ Extension à venir de *GénéSyst*

# *Plan*

I. Rappels généraux sur DEMONEY

II. Exemple de propriétés et objectifs de sécurité

III. Présentation du modèle

IV. Vérification de propriétés et politiques de sécurité

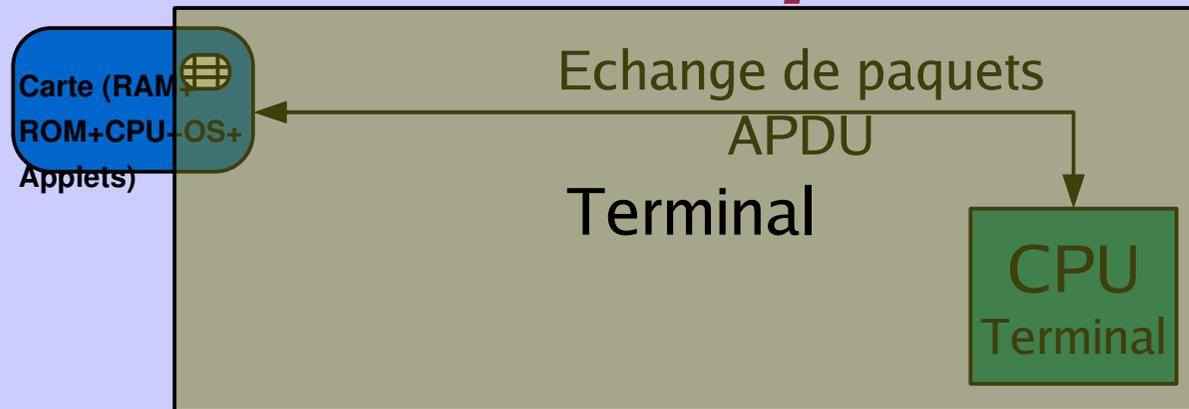
V. Conclusion

# I. Rappels généraux sur DEMONEY

# DEMONEY

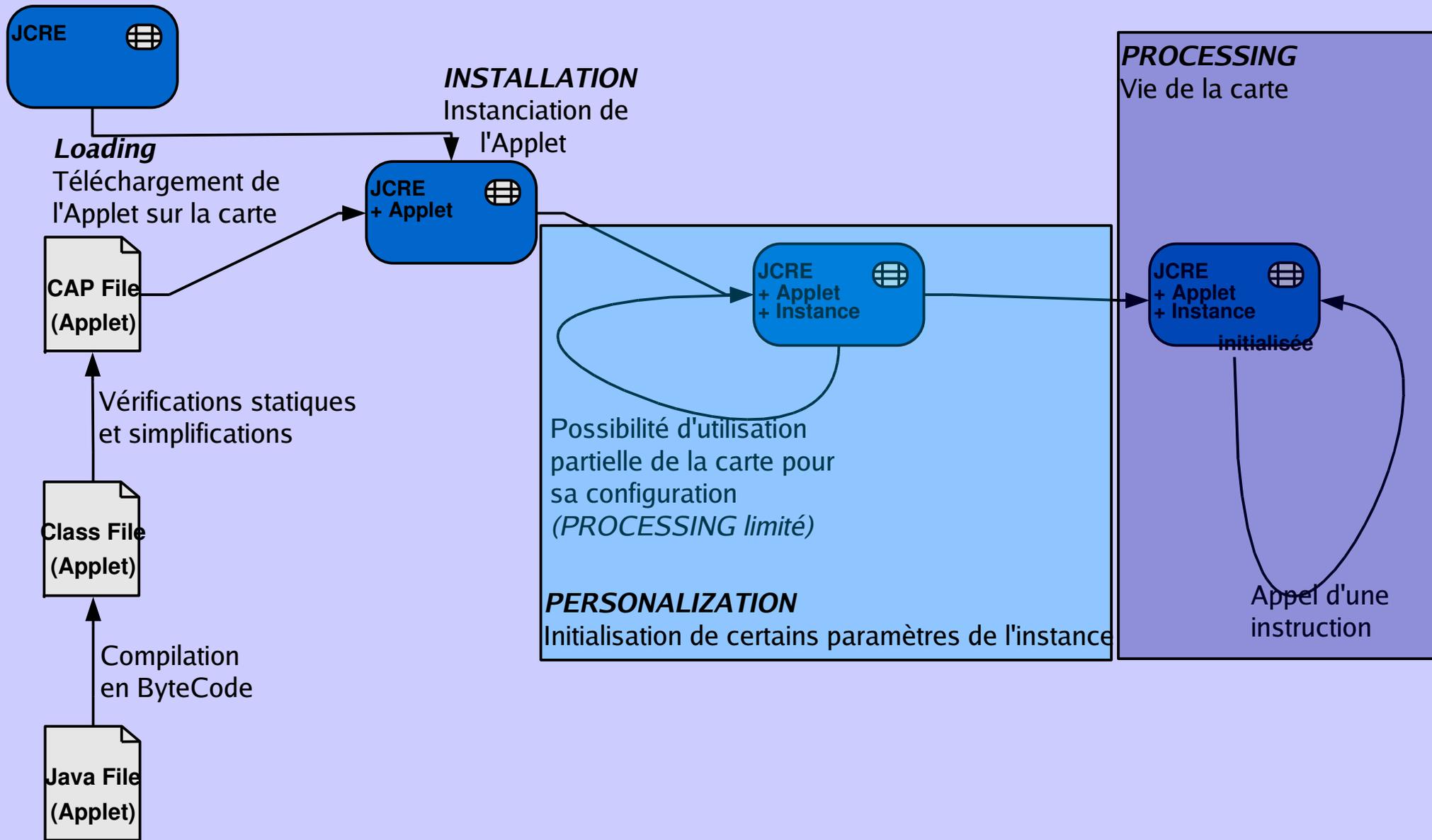
- ◆ Applet JavaCard d'un porte monnaie électronique
  - ◆ *Développée par : **Trusted Logic***
  - ◆ *Cadre : **projet SecSafe***
  - ◆ *Objectif : faire une **étude de cas** représentative des Applets JavaCard*
  - ◆ *Différentes implantations avec différentes **API** mais le même comportement*
    - ◆ *Ici, on ne traite que du cas **DEMONEY StandAlone***

# Concepts de base d'une architecture carte à puce



- ◆ La carte possède :
  - ◆ CPU, RAM, ROM, OS (*Java Card*) et programmes (*Applets*)
- ◆ Une communication est menée par le terminal
  - ◆ La carte ne fait que répondre
- ◆ Communication = échange de paquets APDU
  - ◆ APDU = Tableau de bytes contenant une commande et des données

# Cycle de vie d'une applet



## *2 interfaces différentes*

- ◆ Interface avec le système d'exploitation
  - ◆ Méthodes
- ◆ Interface avec le terminal
  - ◆ APDU reçus par la méthode process

# *Quelques méthodes externes*

## ◆ **Select** :

- ◆ *Informe l'Applet qu'elle est sélectionnée*

## ◆ **process** :

- ◆ *Transmet une commande APDU*

## ◆ **deselect** :

- ◆ *Informe l'Applet qu'elle n'est plus sélectionnée*

# Quelques commandes APDU

passées à l'Applet par la commande *process*

Commande	Effet
STORE DATA	<i>Personnalisation des données de la carte</i>
SELECT	Seul CodeOp imposé par JavaCard
INITIALIZE UPDATE	<i>Entame la procédure de sécurisation du canal</i>
EXTERNAL AUTHENTICATION	<i>Termine la procédure de sécurisation du canal</i>
PIN VERIFY	<i>Vérifie que le code PIN entré est correct</i>
INITIALIZE TRANSACTION	<i>Initialise un crédit ou un débit</i>
COMPLETE TRANSACTION	<i>Finalise un crédit ou un débit</i>
GET DATA	<i>Renvoie les informations publiques de l'instance</i>

# Différents niveaux de sécurité imbriqués

- ◆ **Publique**
  - ◆ Accès aux données publiques (*GETDATA*)
  - ◆ Sécurisation du canal (*INITIALIZEUPDATE, EXTERNALUPDATE, VERIFYPIN*)
- ◆ **Débit** (*terminal de débit et canal chiffré*)
  - ◆ Débit de la carte (*INITIALIZETRANSACTION[Debit], COMPLETE TRANSACTION*)
- ◆ **Crédit** (*terminal de crédit et canal chiffré*)
  - ◆ Crédit de la carte avec des espèces (*INITIALIZETRANSACTION [Credit par espèces]*)
- ◆ **Crédit-Identifié** (*terminal bancaire de crédit et code PIN vérifié*)
  - ◆ Crédit de la carte par virement (*INITIALIZETRANSACTION[Credit par virement bancaire]*)
- ◆ **Administration** (*terminal d'administration et canal chiffré*)
  - ◆ Gestion de la carte (*STOREDATA*)

# PROCESSING

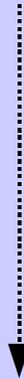
Utilisation d'une Applet

Select APDU (choix d'un AID – Applet Identifier)

Select : Renvoie les infos publiques de l'instance de l'Applet



Process APDU : exécution d'une commande

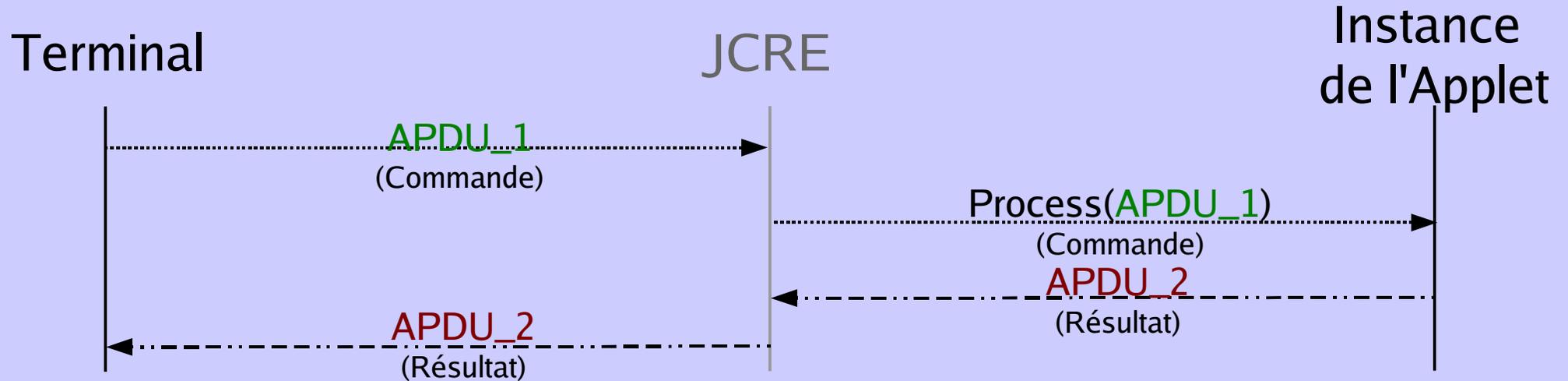


*Deselect : Par selection d'une applet.*

*Note : Reset n'appelle pas la méthode Deselect*

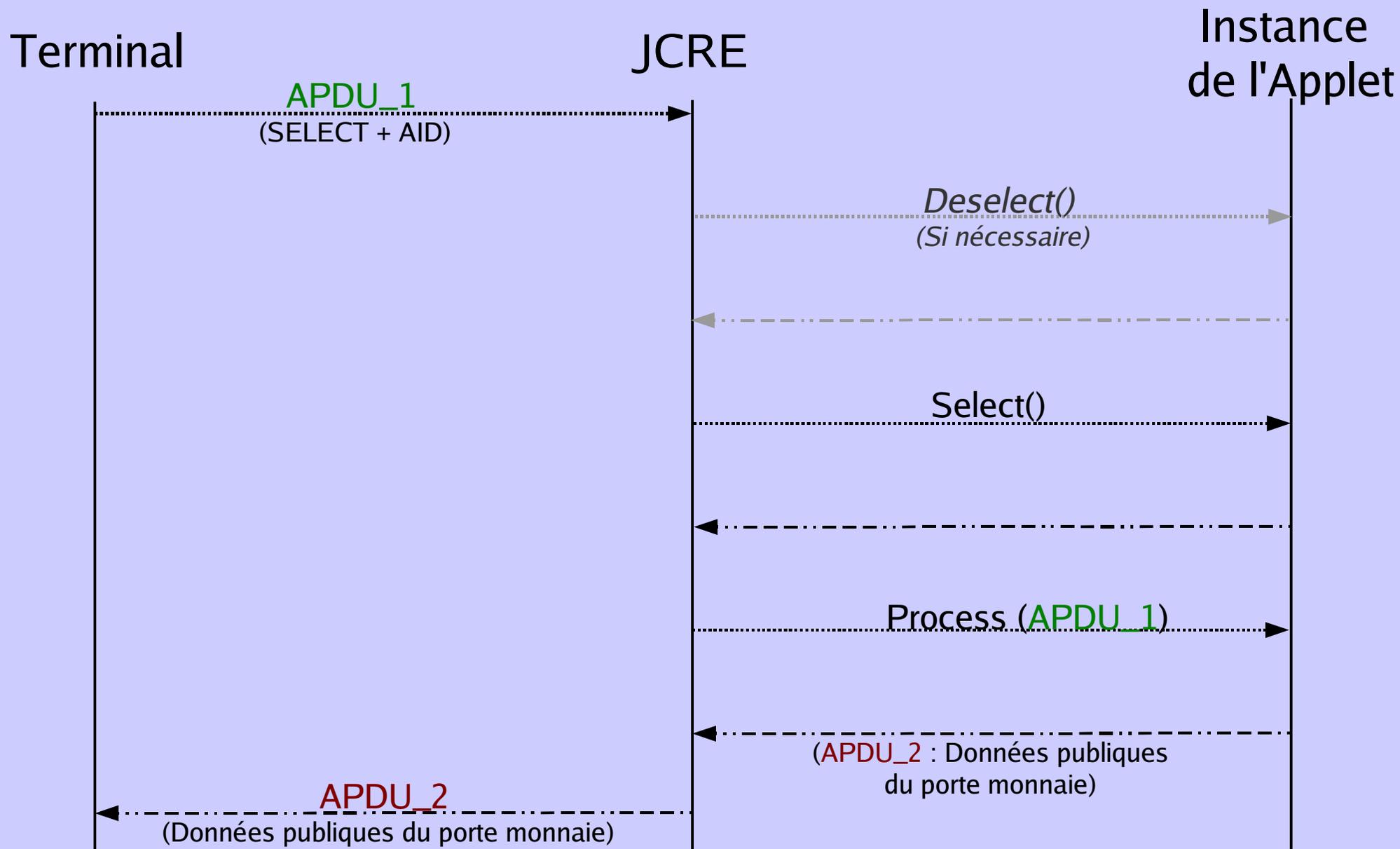
# PROCESS

Envoi d'une commande à une instance de l'Applet

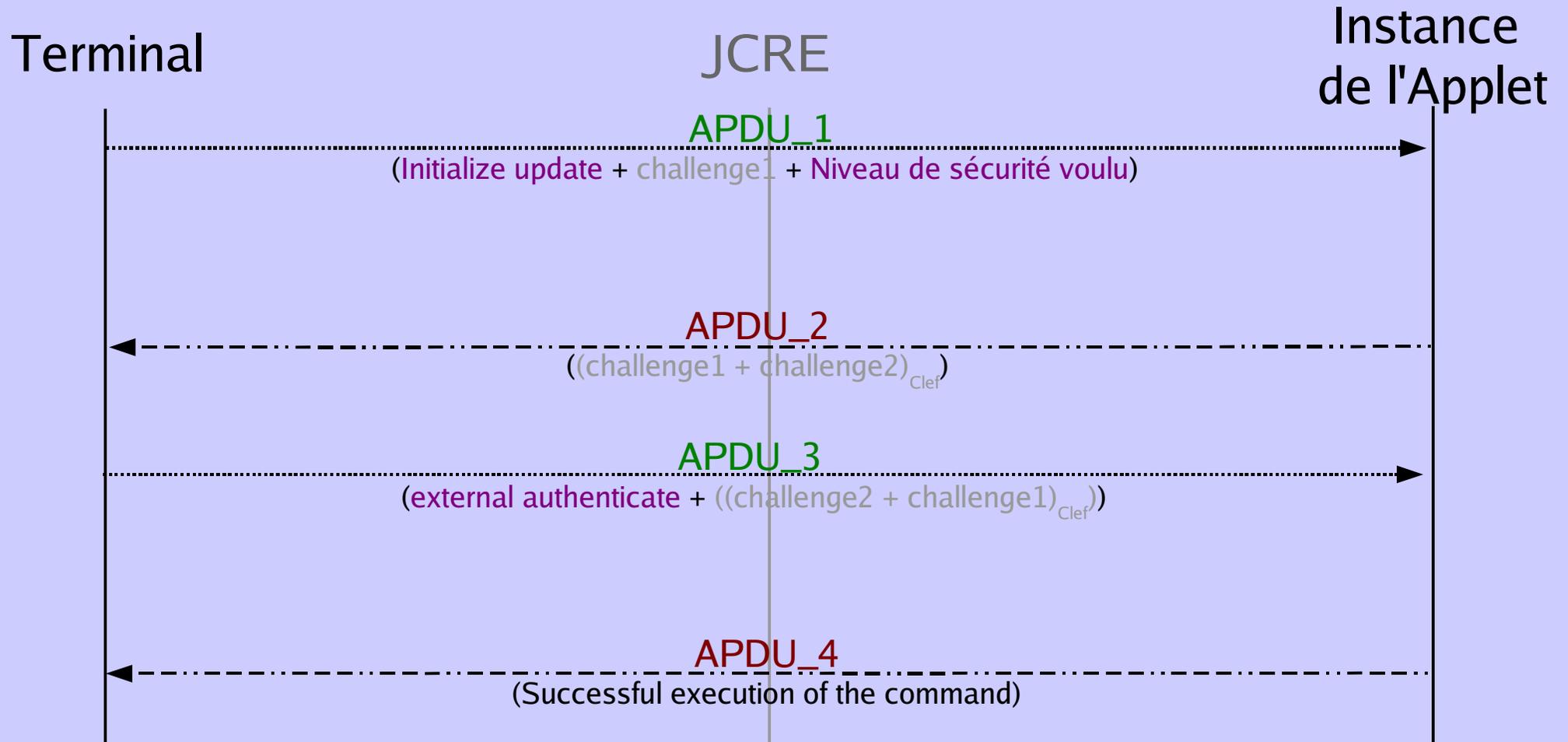


- ◆ AID non précisé
- ◆ Message traité par l'Applet sélectionnée

# SELECT APDU

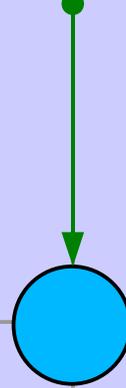


# Sécurisation canal



# *Comportements possibles dans la phase de PROCESSING*

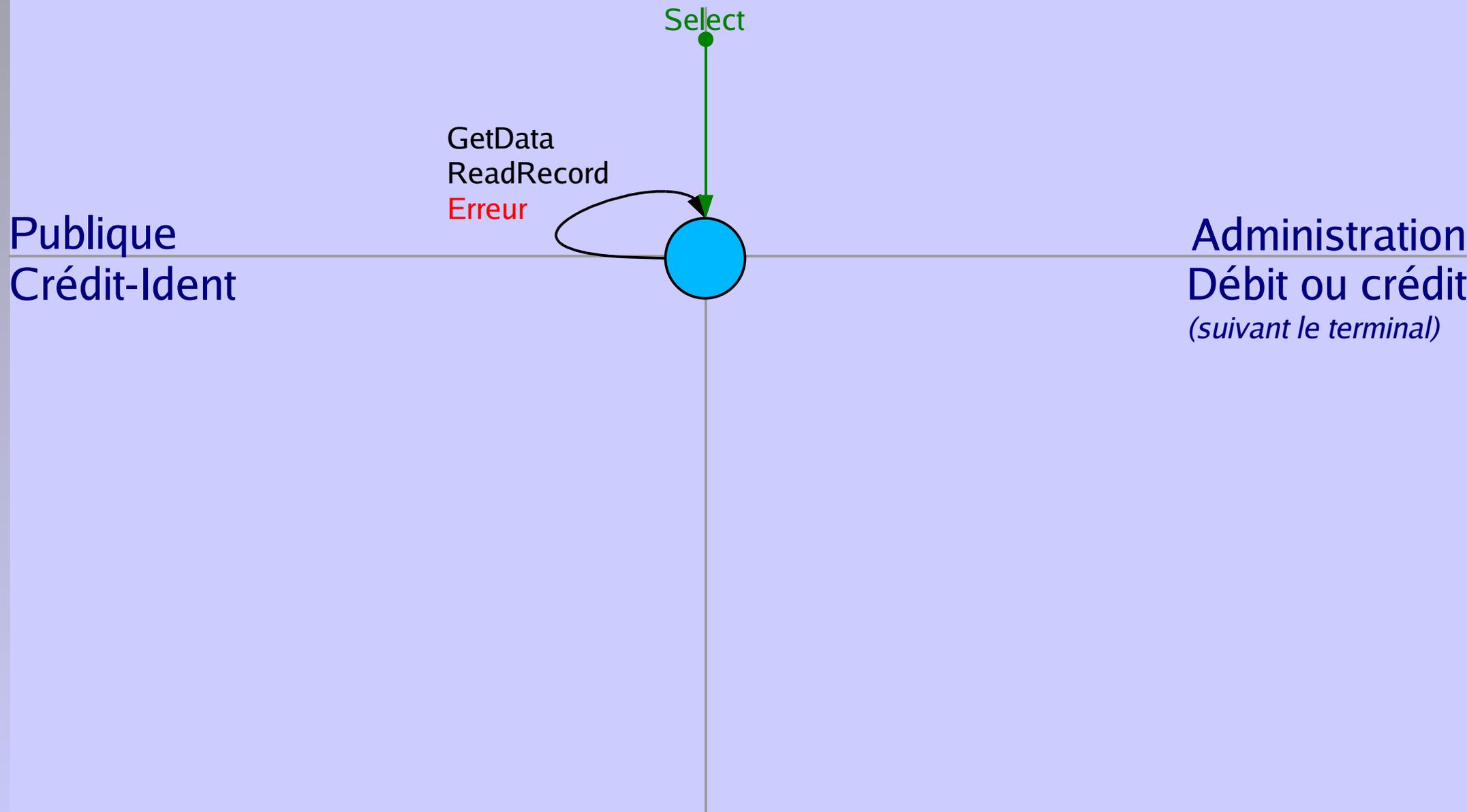
Select



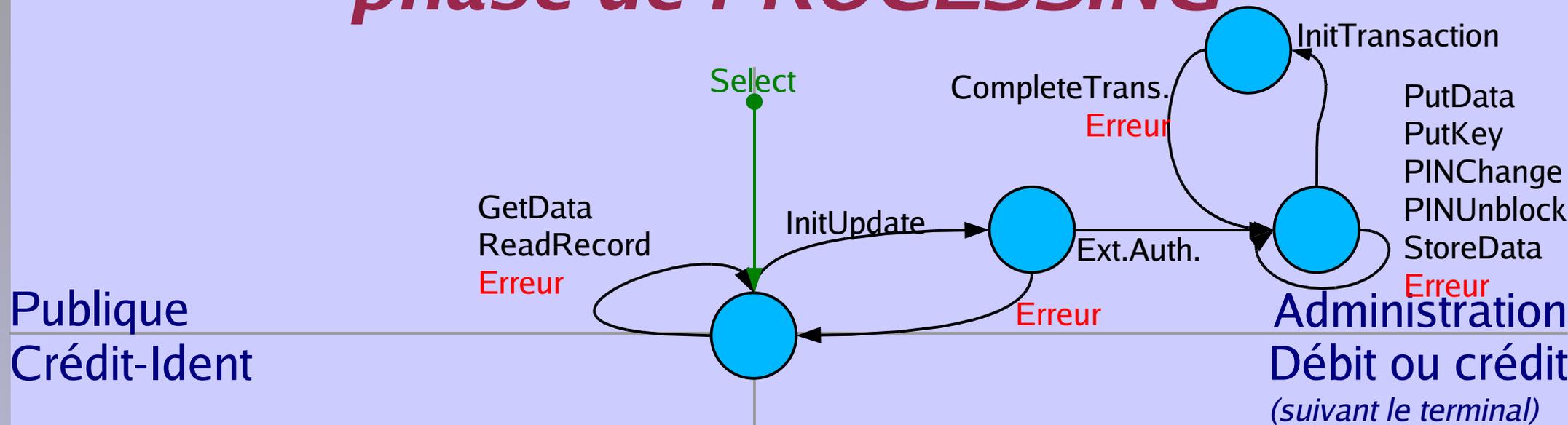
Publique  
Crédit-Ident

Administration  
Débit ou crédit  
*(suivant le terminal)*

# Comportements possibles dans la phase de *PROCESSING*

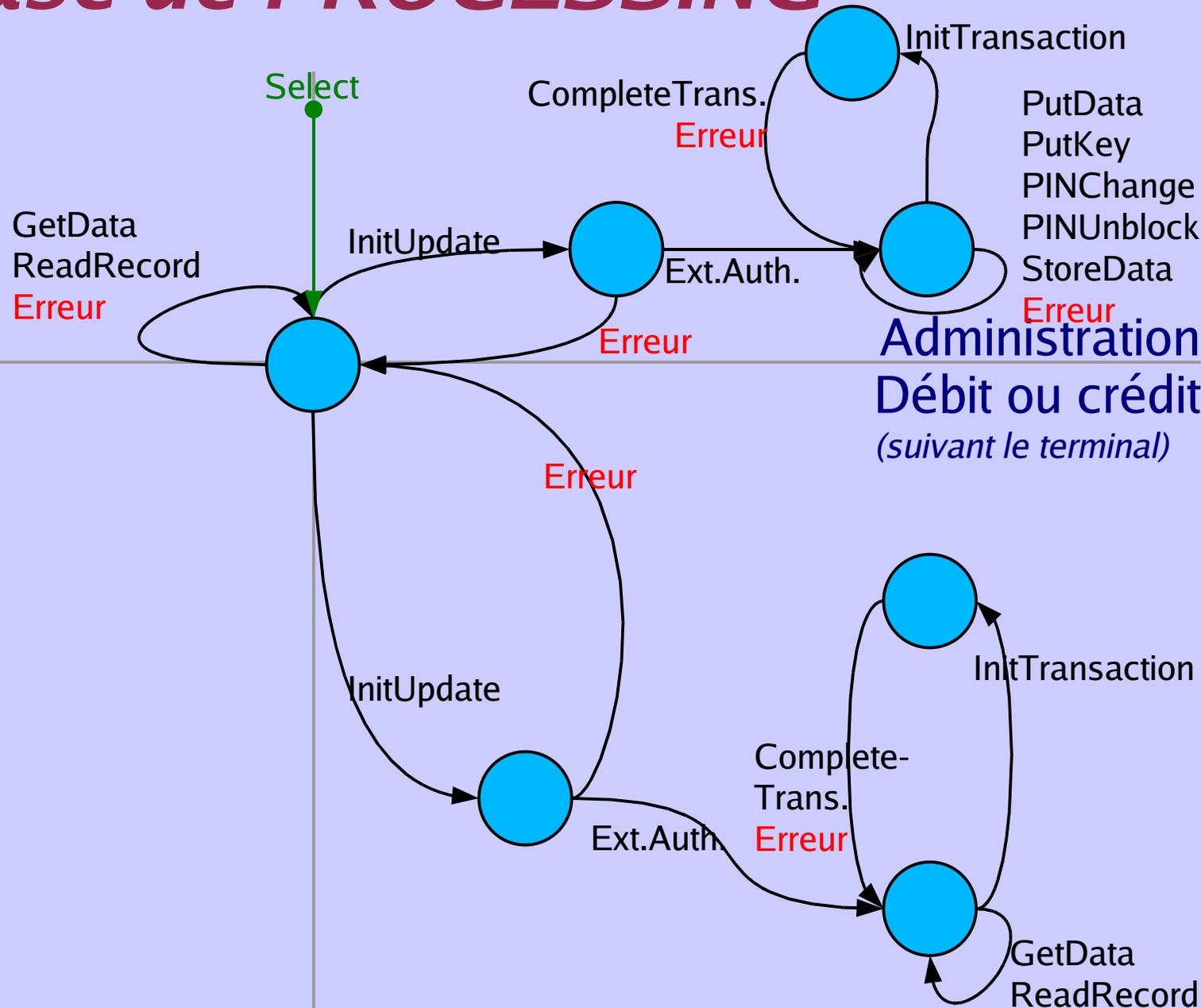


# Comportements possibles dans la phase de PROCESSING



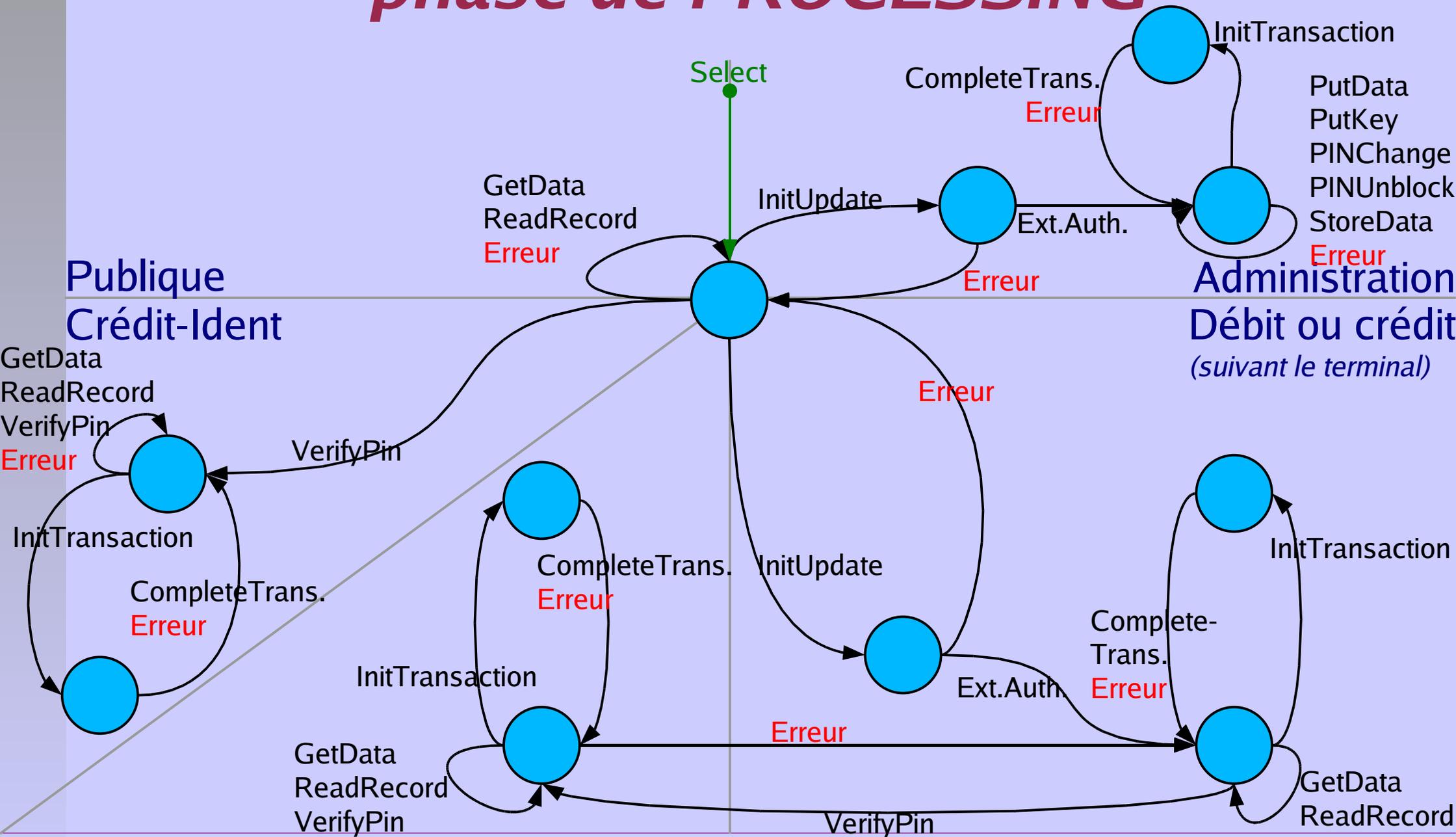
# Comportements possibles dans la phase de PROCESSING

Publique  
Crédit-Ident

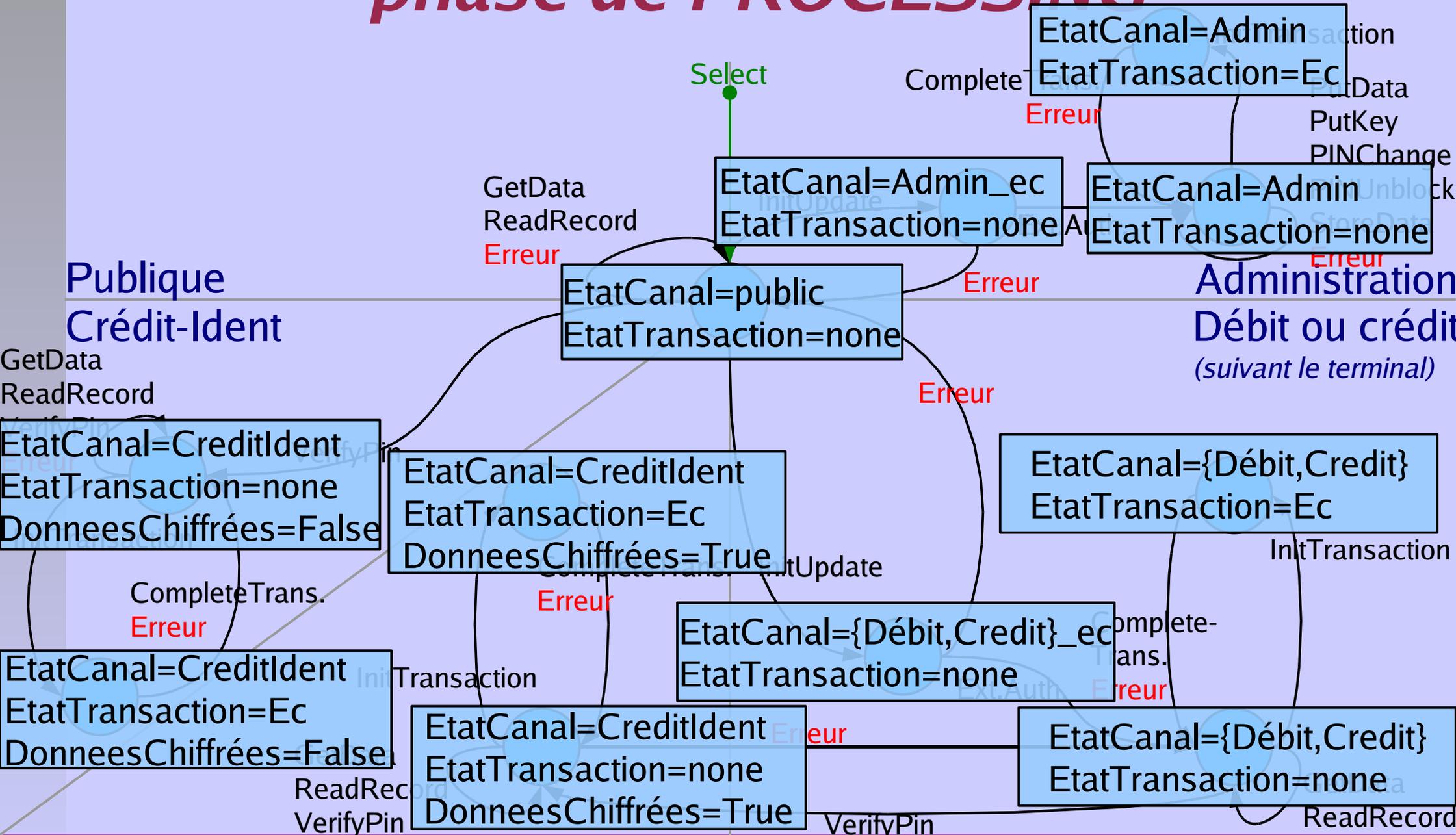


Administration  
Débit ou crédit  
(suivant le terminal)

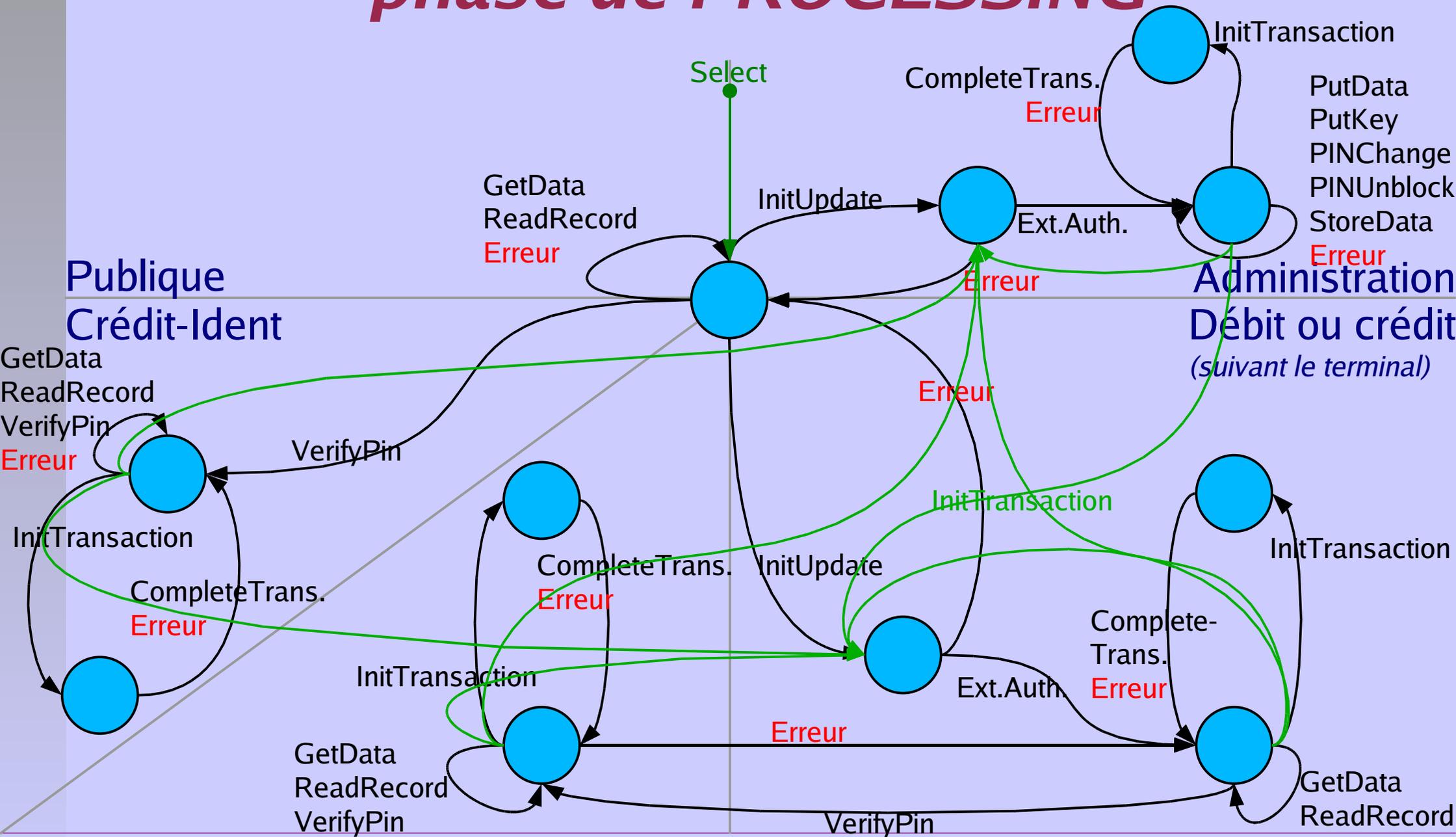
# Comportements possibles dans la phase de PROCESSING



# Comportements possibles dans la phase de PROCESSING



# Comportements possibles dans la phase de PROCESSING



## II. Exemples de propriétés et objectifs de sécurité

# *Propriété de DEMONEY*

- ◆ **Atomicité :**
  - ◆ *Les opérations de mise à jour du solde ne doivent pas être interrompues*
  - ◆ *Les opérations de sécurisation du canal ne doivent pas être interrompues*

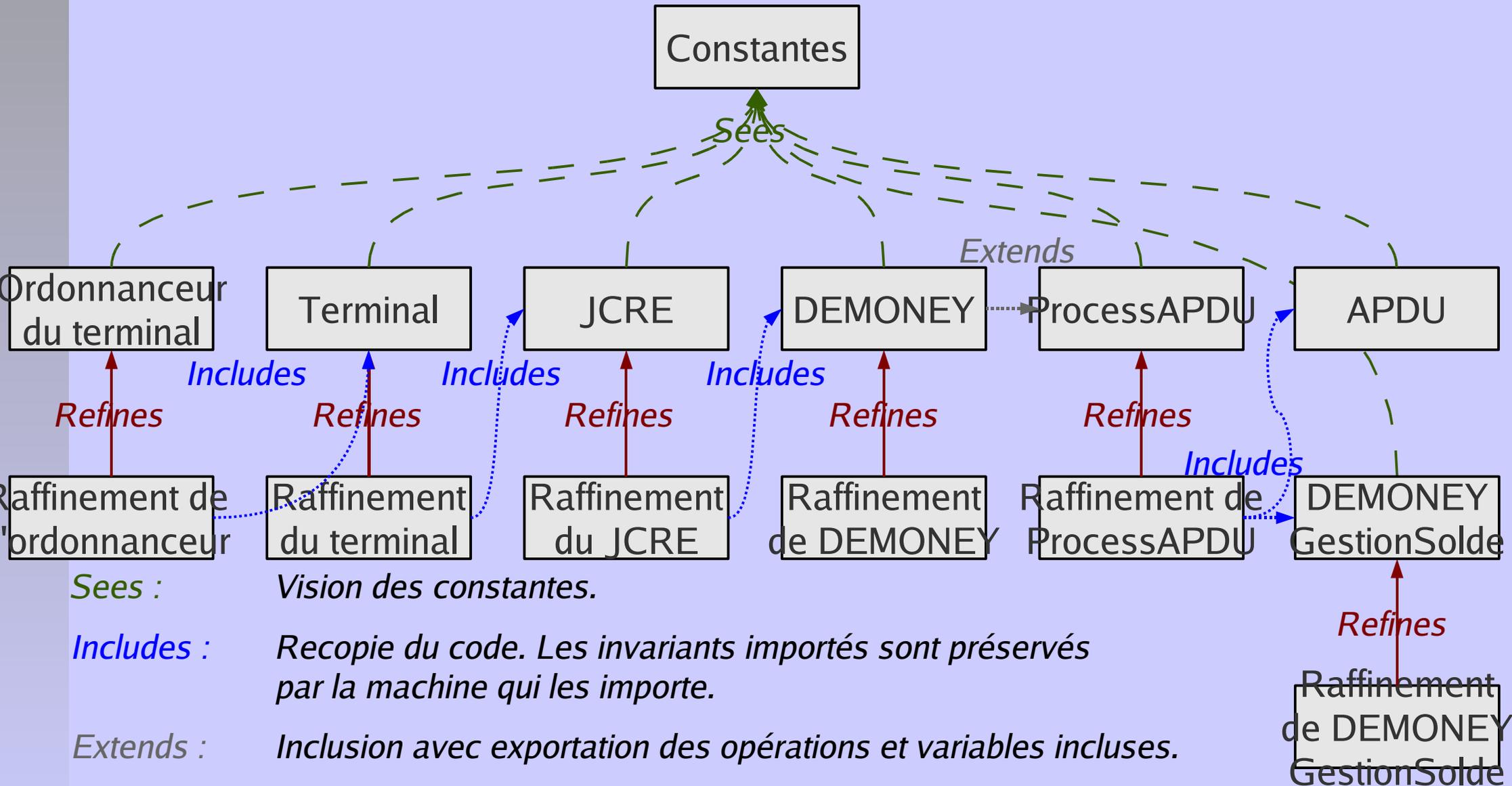
# Objectifs et politiques de sécurité

Sans prise en compte de la cryptographie

- ◆ **On ne peut pas créer d'argent :**
  - ◆ *Crédit : récupérer de l'argent avant de créditer*
  - ◆ *Solde  $\geq 0$*
- ◆ **Il est difficile de perdre de l'argent :**
  - ◆ *Solde mis à jour seulement si débit possible*
  - ◆ *Crédit : récupérer de l'argent avant de créditer*
- ◆ **Seul le titulaire du compte peut créditer la carte par virement :**
  - ◆ *Code PIN avec nombre d'essais limité*

# III. Présentation du modèle

# Organisation du modèle



**Sees :**

*Vision des constantes.*

**Includes :**

*Recopie du code. Les invariants importés sont préservés par la machine qui les importe.*

**Extends :**

*Inclusion avec exportation des opérations et variables incluses.*

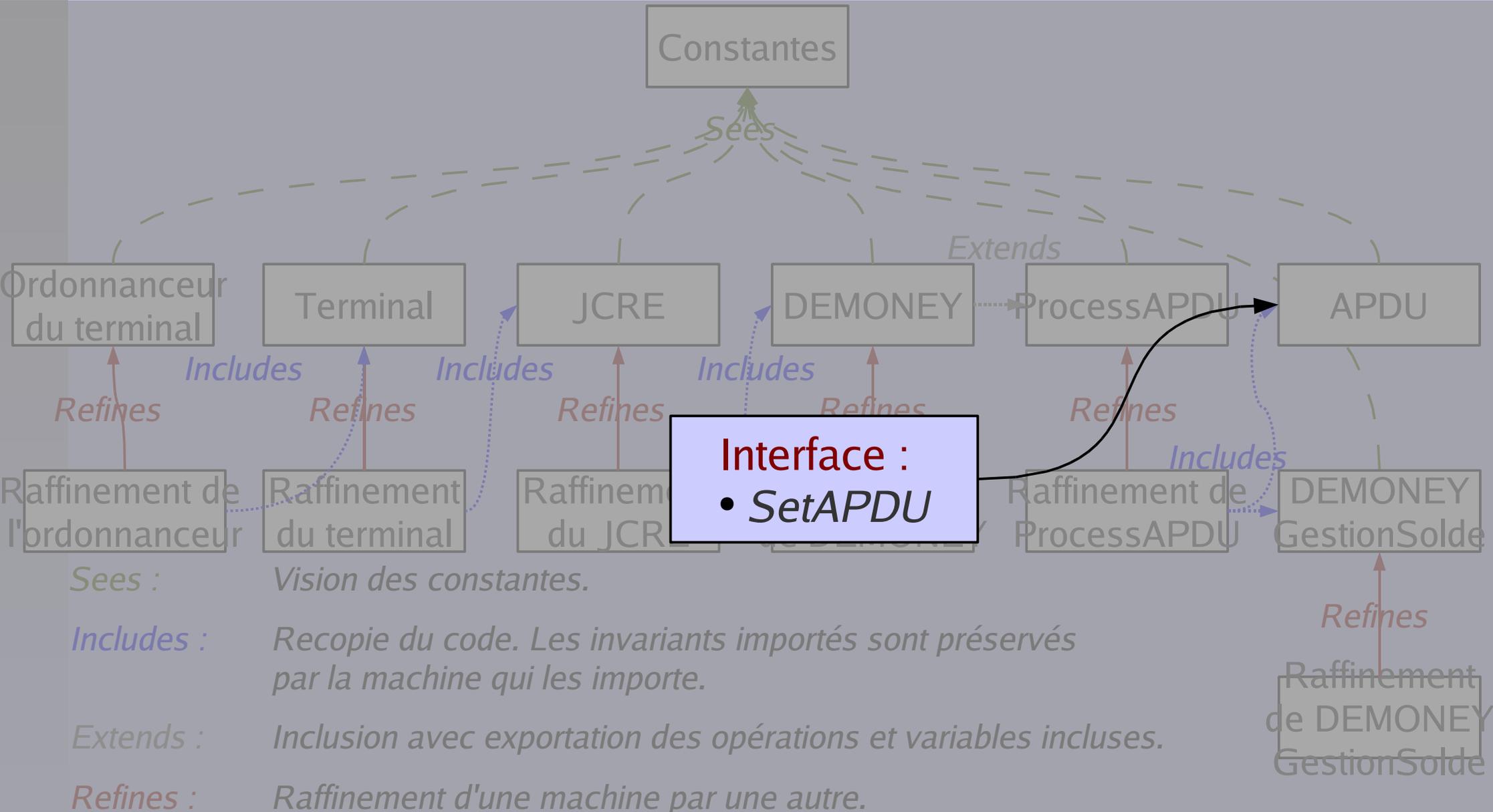
**Refines :**

*Raffinement d'une machine par une autre.*

Raffinement  
de DEMONEY  
GestionSolde

**Refines**

# Organisation du modèle



# Organisation du modèle

## Interface :

- *Reset\_ProcessAPDU*
- *ResetSecuLevel*
- *INS\_APDU\_Select*
- *INS\_APDU\_GetData*
- *INS\_APDU\_StoreData*
- *INS\_APDU\_InitializeUpdate*
- *INS\_APDU\_ExternalAuthentification*
- *INS\_APDU\_InitializeTransaction*
- *INS\_APDU\_InitializeTransactionFromBank*
- *INS\_APDU\_CompleteTransaction*
- *INS\_APDU\_VerifyPIN*

Constantes

Sees

Extends

ProcessAPDU

APDU

Refines

Raffinement de  
ProcessAPDU

Includes

DEMONEY  
GestionSolde

Refines

Raffinement  
de DEMONEY  
GestionSolde

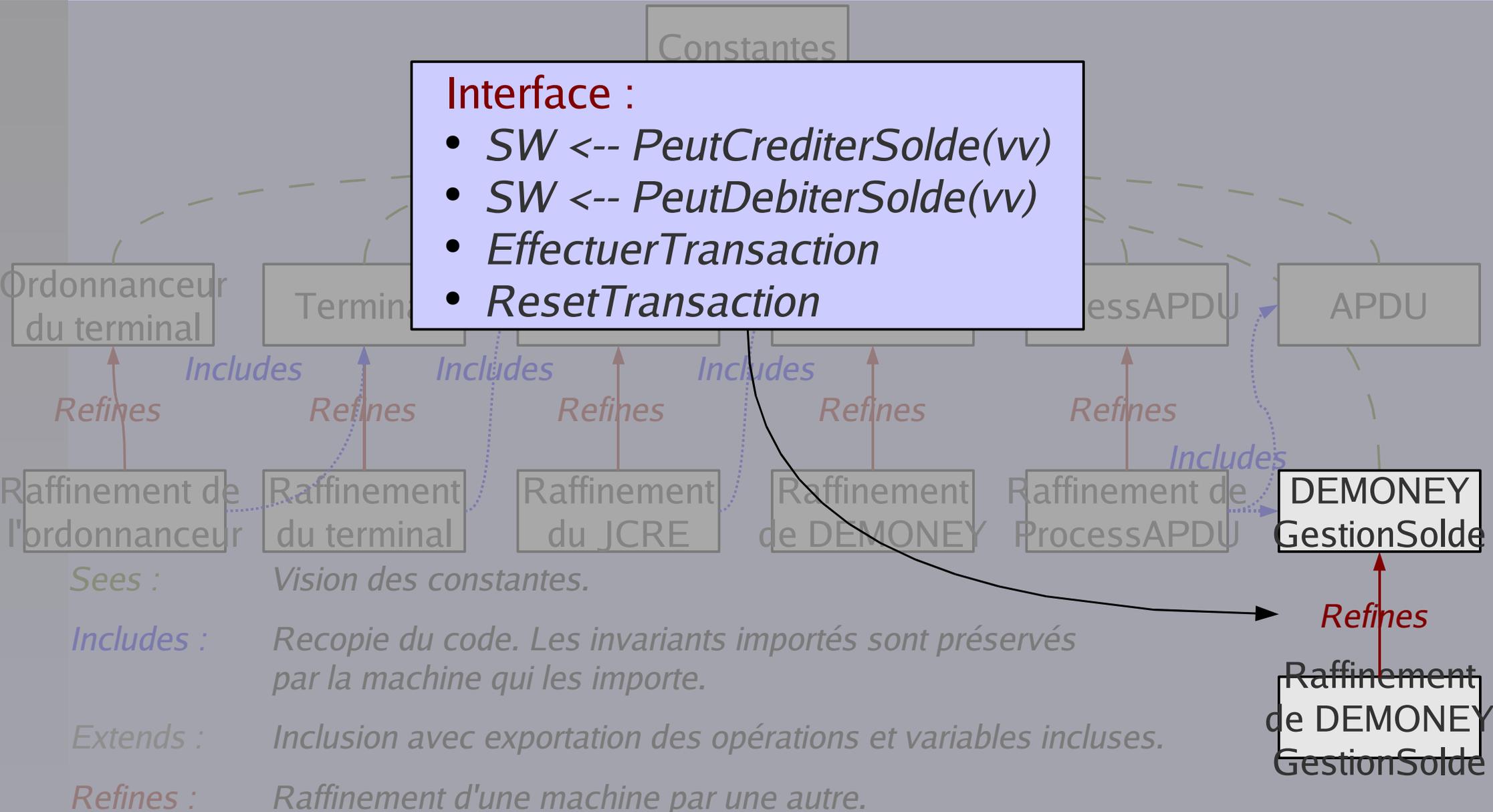
es

ment  
ONEY

ont préservés

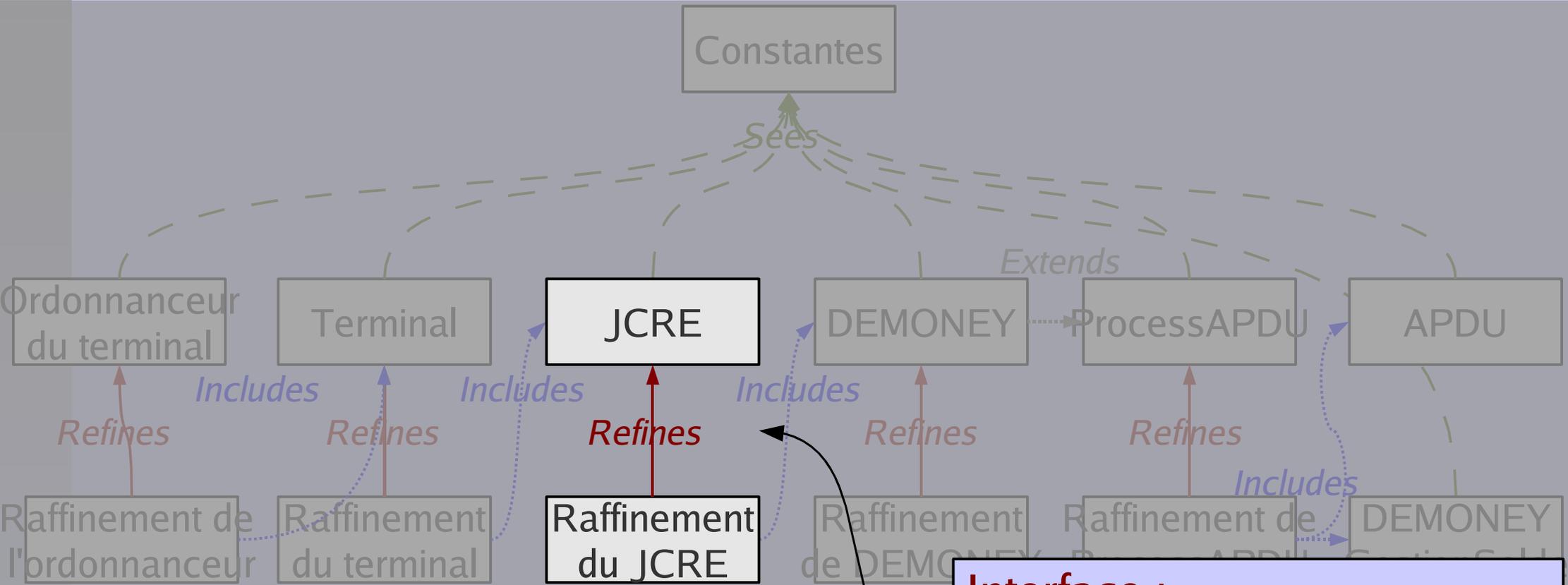
variables incluses.

# Organisation du modèle





# Organisation du modèle



*Sees :*

*Vision des constantes.*

*Includes :*

*Recopie du code. Les invariants importés sont par la machine qui les importe.*

*Extends :*

*Inclusion avec exportation des opérations et*

*Refines :*

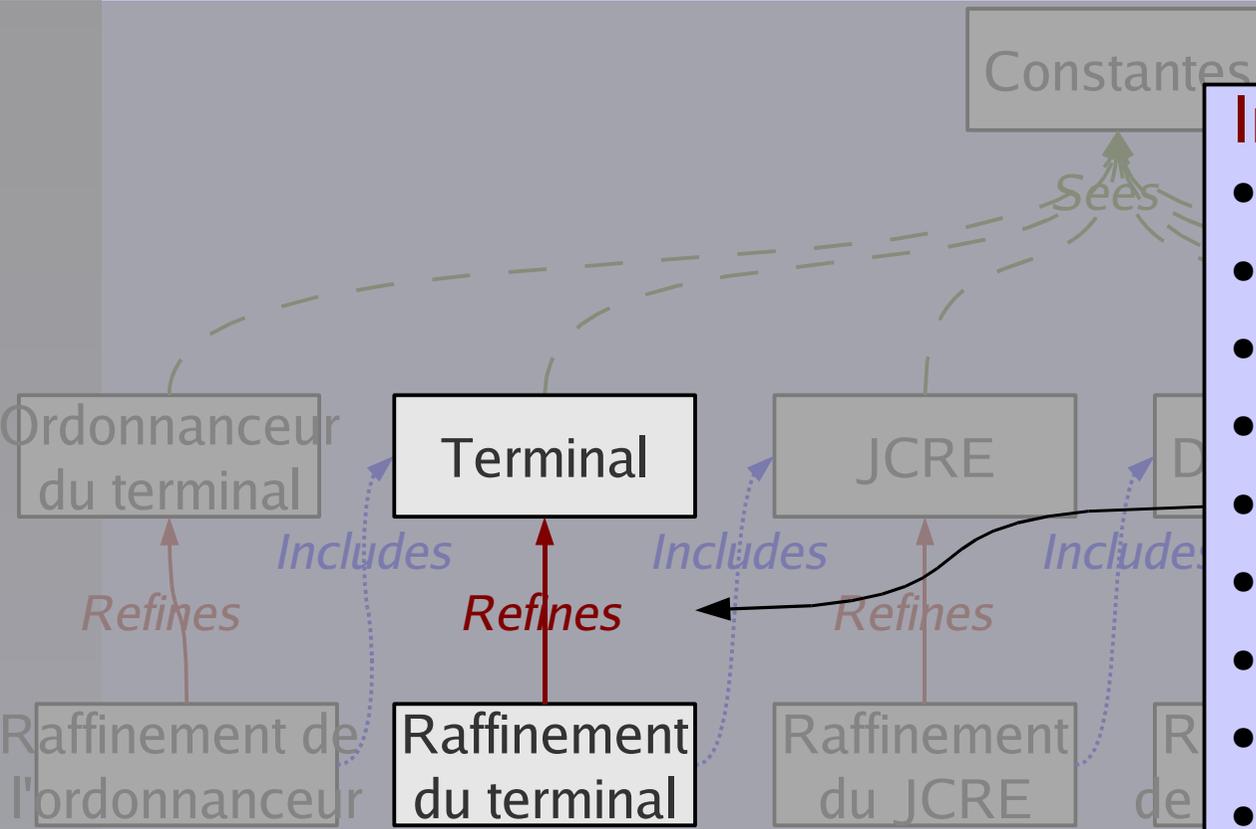
*Raffinement d'une machine par une autre.*

## Interface :

- *EmissionAPDUVersCarte*
- *TraitementAPDU*
- *Reset\_JCRE*

*GestionSolde*

# Organisation du modèle



## Interface :

- *EmissionReset*
- *EmissionSelect*
- *EmissionGetData*
- *EmissionStoreData*
- *EmissionInitializeUpdate*
- *EmissionExternalAuthentication*
- *EmissionInitializeTransaction*
- *EmissionCompleteTransaction*
- *EmissionVerifyPIN*
- *TraitementErreur*

*Sees :*

*Vision des constantes.*

*Includes :*

*Recopie du code. Les invariants importés sont préservés par la machine qui les importe.*

*Extends :*

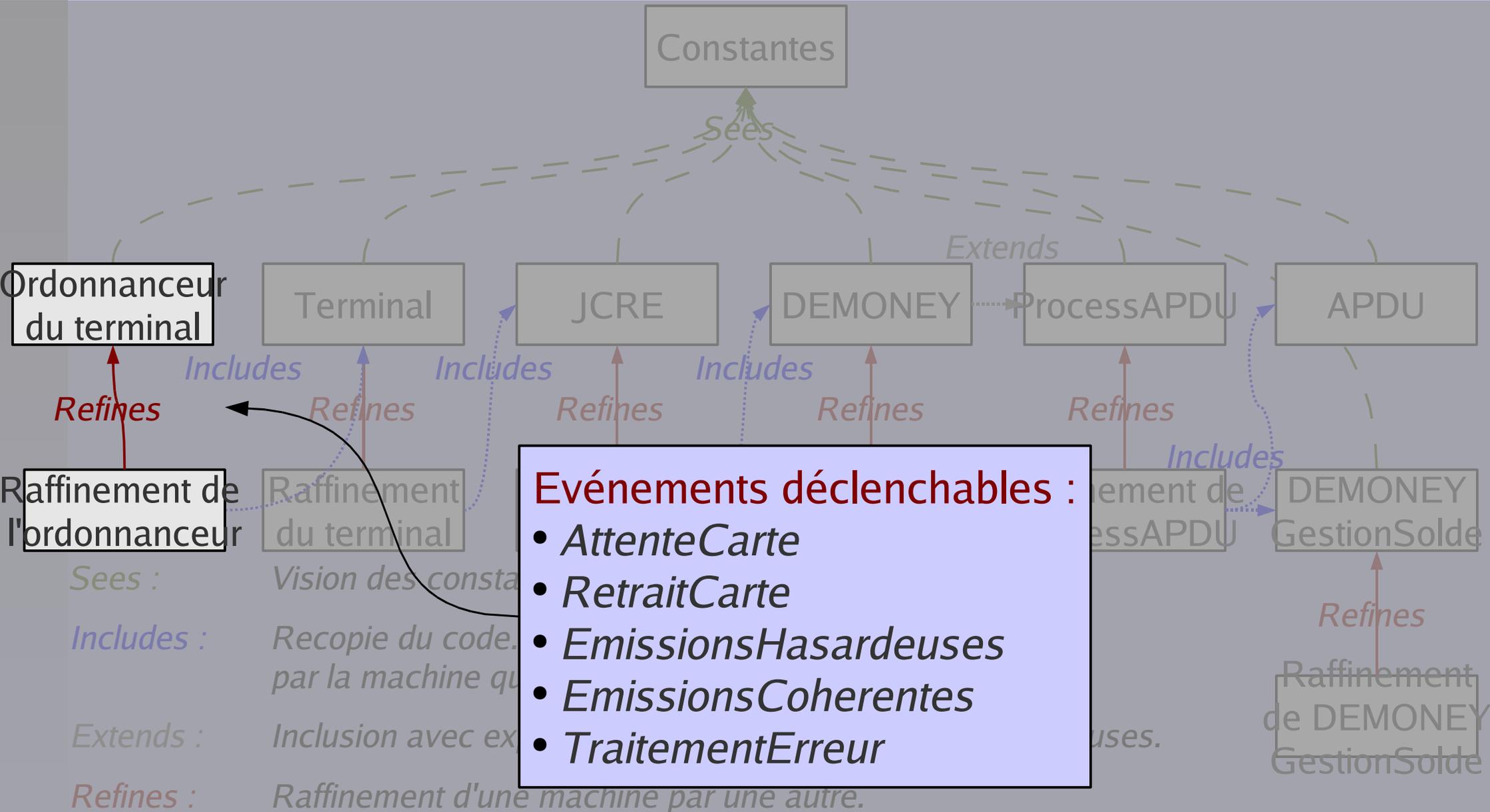
*Inclusion avec exportation des opérations et variables incluses.*

*Refines :*

*Raffinement d'une machine par une autre.*

Raffinement  
de DEMONEY  
GestionSolde

# Organisation du modèle



## IV. Vérification de propriétés et politiques de sécurité

# Vérification des propriétés de DEMONEY

## ◆ Atomicité :

◆ *Les opérations de mise à jour du solde ne doivent pas être interrompues*

◆ *Si l'on finalise une transaction alors elle était en cours :*

**PROCESS\_APDU :**

*INS\_APDU\_CompleteTransaction = PRE TransactionEc=TRUE THEN .... END*

◆ *Si l'on interprète un APDU différent de CompleteTransaction alors aucune transaction n'est en cours :*

**PROCESS\_APDU :**

*INS\_APDU\_GetData = PRE TransactionEc=FALSE THEN ... END*

*(pour toute les autres commandes)*

◆ *Vérifier sur le graphe comportemental*

◆ *Les opérations de sécurisation du canal ne doivent pas être interrompues*

**Idem.**

# Vérification des objectifs et politiques de sécurité

- ◆ On ne peut pas créer d'argent :
  - ◆ *Crédit : récupérer de l'argent avant de créditer*
  - ◆ *Vérifier sur le graphe comportemental*

**DEMONEY\_GestionSolde :**

**INVARIANT**

*SommeTransEc : Int16 &*

*SommeTransEc >= 0 &*

*((TransactionEc = Trans\_Credit) => ((SoldeCourant + SommeTransEc) : 0..SoldeMaxi))*

- ◆ *Solde >= 0*

**DEMONEY\_GestionSolde :**

**INVARIANT**

*SoldeCourant : 0..SoldeMaxi*

# Vérification des objectifs et politiques de sécurité

- ◆ Il est difficile de perdre de l'argent :
  - ◆ *Solde mis à jour seulement si débit possible*

**DEMONEY\_GestionSolde :**

**INVARIANT**

*SommeTransEc : Int16 &*

*SommeTransEc >= 0 &*

*((TransactionEc = Trans\_Debit) => ((SoldeCourant - SommeTransEc) : 0..SoldeMaxi))*

# Vérification des objectifs et politiques de sécurité

- ◆ **Seul le titulaire du compte peut créditer la carte par virement :**
  - ◆ *Code PIN avec nombre d'essais limité*

## **Process\_APDU :**

*INVARIANT*

*NbEssaisPINRestant > 0 => PINBlocked = FALSE*

*OPERATIONS*

*INS\_APDU\_VerifyPIN = PRE TransactionEc = Trans\_None & NbEssaisPINRestant > 0 THEN  
NbEssaisPINRestant := NbEssaisPINRestant - 1 ; ....  
END*

# V. Conclusion

## *Travail à venir*

- ◆ Règles systématiques d'expression de propriétés de sécurité en B.
- ◆ Méthode de conversion automatique d'un modèle B classique en un système B événementiel.
- ◆ Génésyst : la possibilité de vérifier une propriété sur un automate.
- ◆ Vérification modulaire.

# Références

- ◆ **Présentation de l'application DEMONEY** C. Paulin (01/2004)
- ◆ **DEMONEY : a démonstrative Electronic purse** R. Marlet et C. Mesnil (*Trusted Logic, 11/2002*)
- ◆ **DEMONEY : JavaCard implementation**, R. Marlet (11/2002 – 3 pages pour préciser l'annexe B)
- ◆ **Security properties and Java Card Specificities to be studied in the SacSafe Project**, R. Marlet, D. Le Metayer (8/2001 – *Spécificités des applets Java-Card. Propriétés à vérifier par analyse statique*)
- ◆ **JavaCard 2.1 Platform Specifications**
- ◆ **Manual for TL-FIT Draft**, Trusted Logic