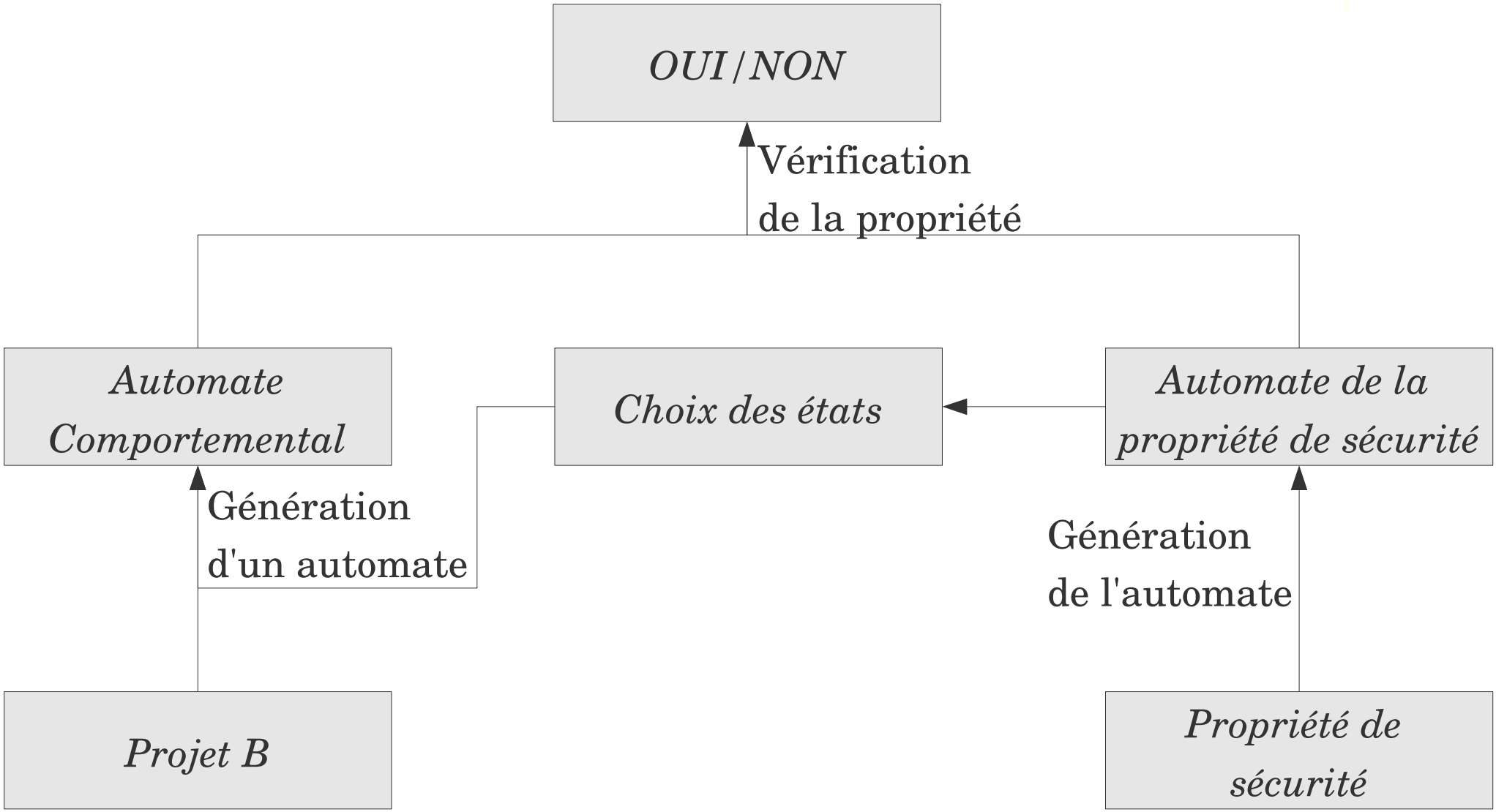
Vérification de Propriétés de sécurité

Travaux financés par une bourse CNRS / STMicroelectronics

Objectifs (1/4)



Objectifs (2/4)

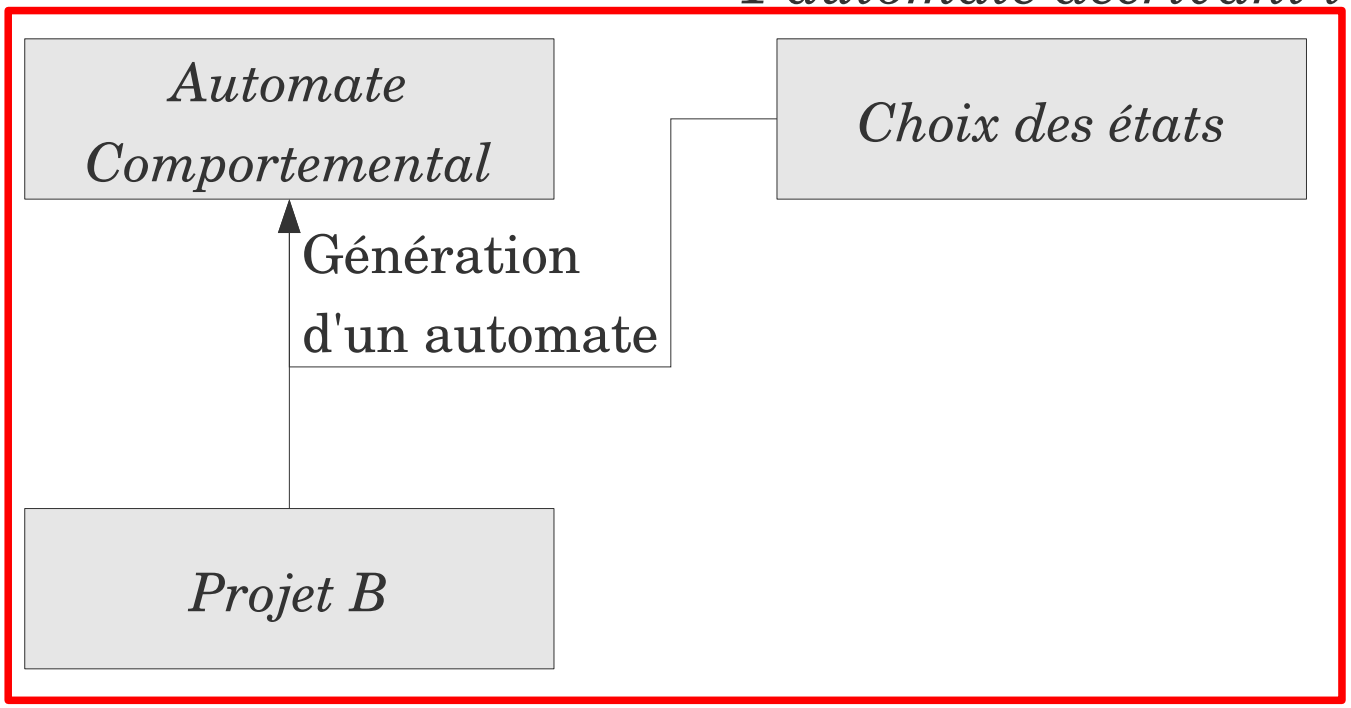
Déjà réalisé par l'outil GénéSyst

Entrées :

- *1 machine et les états attendus de l'automate*
- *(1 raffinement et ses états de représentation)*

Sortie :

- *1 automate décrivant l'ensemble des comportements possibles de la spécification.*



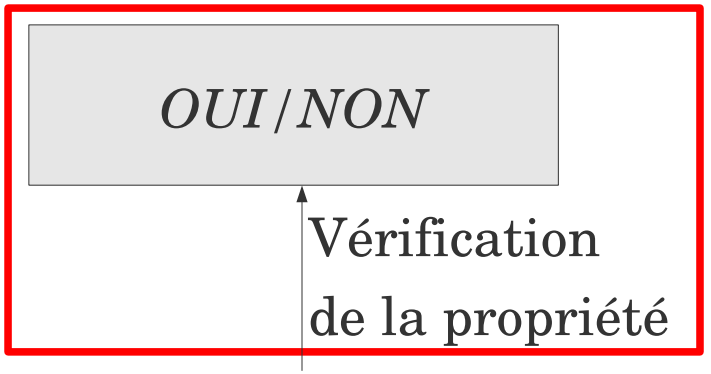
Objectifs (3/4)

Choix d'une logique temporelle.
Contraintes :

- Permet d'exprimer au moins le *Leads to*
- Permet de décorer les modalités avec des événements

Propriété de sécurité

Objectifs (4/4)



Plusieurs techniques complémentaires :

- Comparaison d'automates
- Model-Checking
- vérification des obligations de preuve de la propriété sur la spécification

Plan

I. GénéSyst

II. Prise en compte de projets modulaires

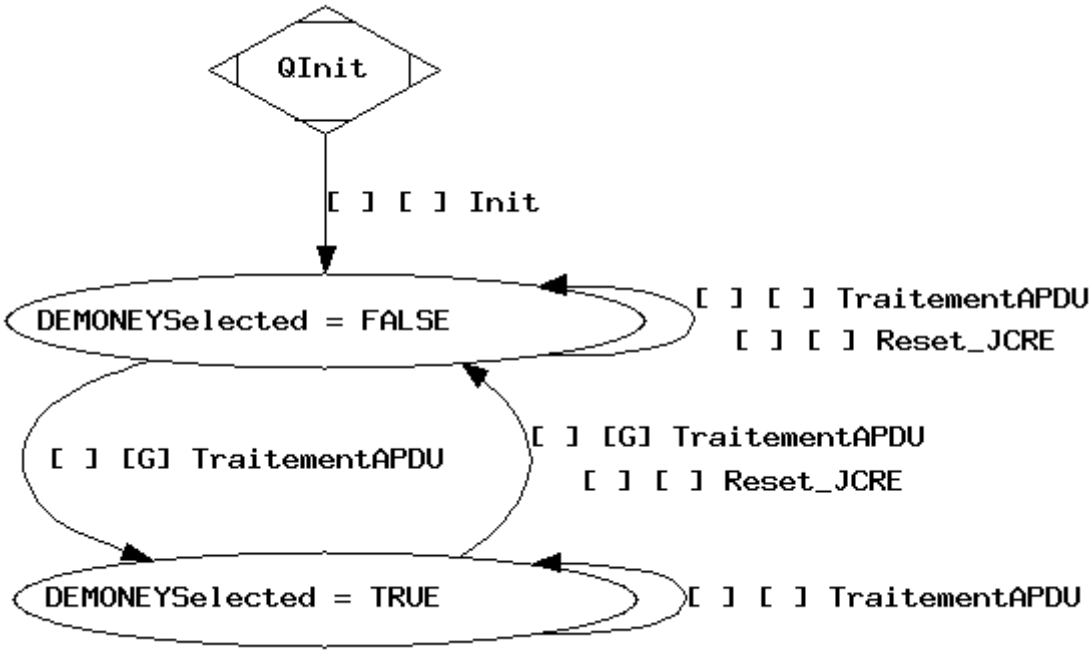
III. Expression de propriétés

IV. Perspectives

I. GénéSyst

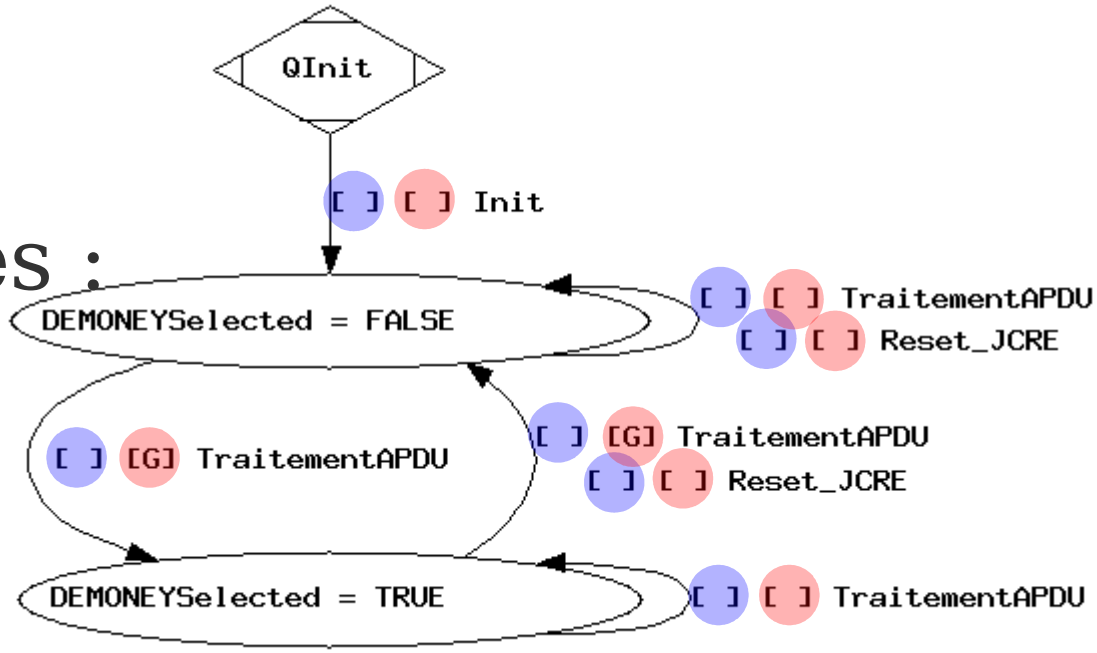
I.1. Généralités

- Rôle :
  - *Extraction du contrôle d'un système B*
- Entrées :
  - *Spécification*
  - *Etats*
- Sortie :
  - *Automate comportemental*
- Méthode :
  - *Génération puis vérification d'obligations de preuve*

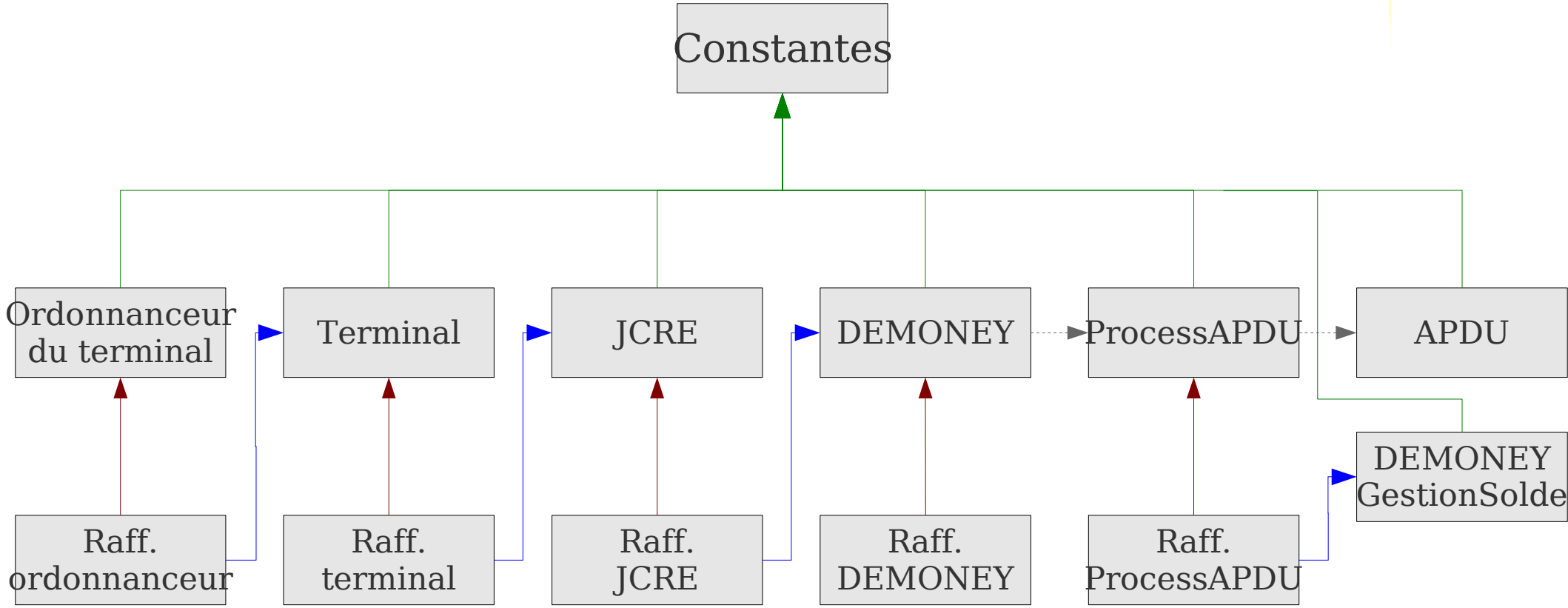


I.2. Particularités

- Automate exact (grâce aux gardes)
- Gardes en 2 parties :
  - *Déclenchabilité*
  - *Atteignabilité*
- Conditions simplifiées :
  - *Condition vraie* [ ]
  - *Condition fausse*
  - *Condition non vraie :* [G]
- Prise en compte du raffinement
  - *Représentation hiérarchique [Stouls-Potet-AFADL04]*



I.3. Rappel de la modélisation DEMONEY utilisée



Sees : *Vue des constantes communes.*

Includes : *Recopie du code. Les invariants sont préservés.*

Extends : *Inclusion avec exportation des opérations.*

Refines : *Raffinement d'une machine par une autre.*

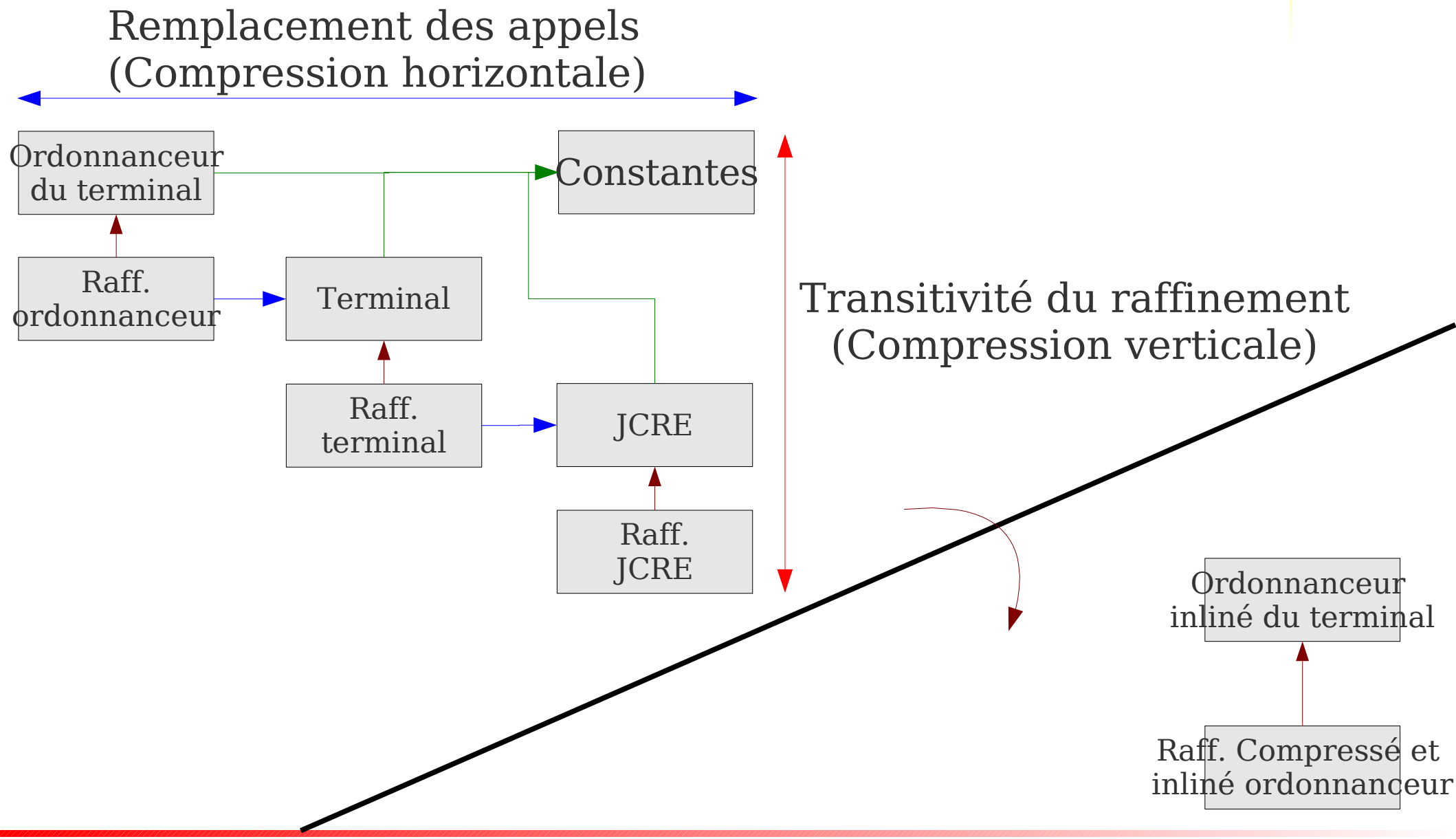
I.4. Contraintes

- GénéSyst ne supporte pas les projets
  - *1 seul raffinement autorisé*
  - *Clauses sees/includes/extends/... non reconnues*
- GénéSyst nécessite du B événementiel
- Déclaration d'invariants fortement dépendant de l'architecture
  - *Ex : $x+y \leq cst$*

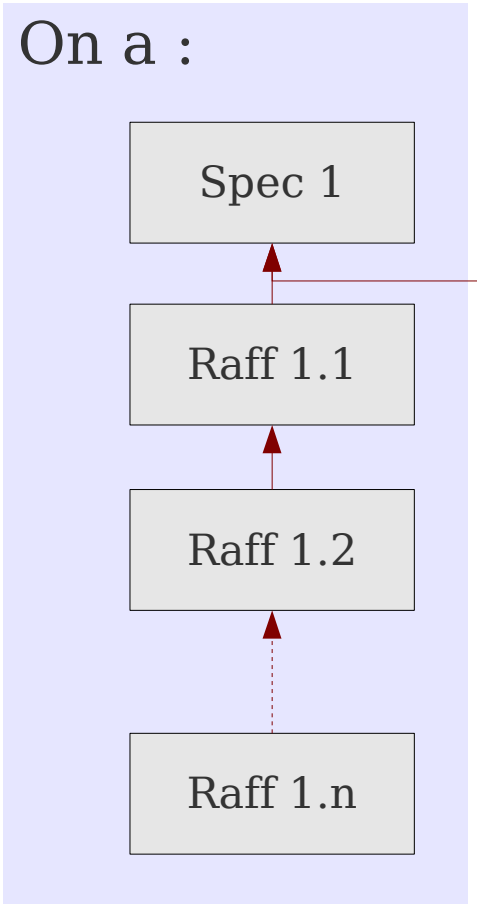
    où x et y des variables de Terminal et DEMONEY_GestionSolde

II. Prise en compte de projets modulaires

II.1. Objectif



II.2. Transitivité du raffinement



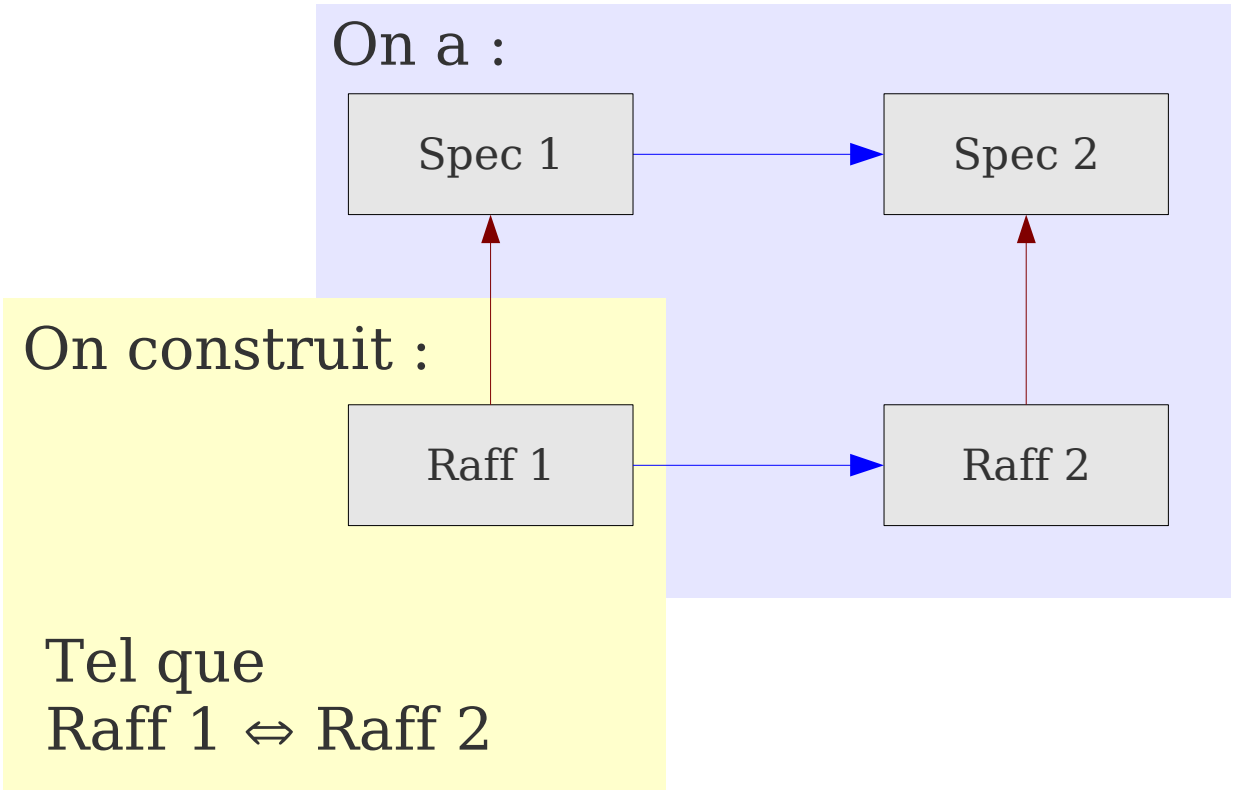
On construit :

Raff 1.n'

Tel que
Raff 1.n' ⇔ Raff 1.n

Outil :
Magistère de Pierre Bontron

II.3. Monotonie du raffinement



Outil :
Stage d'Evelyne Altariba

III. Expression de propriétés

III.1. Exemple de propriétés de sécurité visées *(Marieke H.)*

- Atomicité
  - *Pas de transition oubliée ou en plus*
  - *Tentatives d'authentification limitées*
- Cycle de vie
  - *Personnalisation authentifiée*
  - *La personnalisation n'a lieu qu'une seule fois*
- Contrôle d'accès
  - *Seul le possesseur de la carte peut effectuer un virement*

III.2. Choix de la logique

- Utiliser des logiques existantes
  - *P(L)TL (linéaire – Manna & Pnuelli 1977)*
  - *CTL (arborescente – Clark, Emerson et Sistla 1986)*
  - *CTL* (linéaire et arborescente) (Directed Computational tree logic)*
  - *μ-Calcul (arborescente – Kozen 1983)*
  - *TLA (linéaire - Lamport 1994)*

- *Utiliser des systèmes de transitions étiquetées*

III.3. Exemples de propriétés vérifiables

- *Tentatives d'authentification limitées*

  ✸* $\forall N.(N=NbEssais \Rightarrow (O_{\{authentification\}}\ (Erreur=FALSE \Rightarrow NbEssais<N)))$

  Rappel de l'invariant du système : $NbEssais \in NATURAL$

- *Personnalisation authentifiée*

  ✸*$(CanalSecurise=FALSE \Rightarrow (O_{\{StoreData\}}\ Erreur=TRUE))$

- *La personnalisation n'a lieu qu'une seule fois*

  ✸*$(Personnalise=TRUE \Rightarrow O_{\gamma}\ Personnalise=TRUE)$

III.4. Vérification de propriétés

- Par vérification d'obligations de preuve
  - *Next, Until et Leads to introduis par J.R. Abrial et L. Mussat*
- Par model checking
  - *Simplifié car les mêmes états entre spec et propriété*
    - ↳ *Comparaison des étiquettes uniquement*

IV. Perspectives

IV.1. Perspectives

- Finir les outils (stage d'automne)
- Expérimenter sur le modèle DEMONEY
  - *Construire des automates en fonction des propriétés attendues*
  - *Différentes techniques de vérification (Projet MODESTE)*
- Définir le langage d'expression des propriétés de sécurité.
  - *Langage de propriétés actionnelles défini par M.H. Sur JML ?*

IV.2. Démo

Merci de votre ~~courage~~ attention.

Des questions ?