Intégration du raffinement formel dans le processus de conception des SoCs

Hocine Mokrani, Rabéa Ameur-Boulifa, Sophie Coudert, Emmanuelle Encrenaz-Tiphene

LABSOC

(Telecom ParisTech – Sophia Antipolis)

hmokrani@telecom-paristech.fr

MSR Novembre 2011

Objectif

Notre travail vise l'amélioration de la phase d'exploration d'architecture en vue de produire des systèmes SoC corrects par construction en utilisant des méthodes formelles.

Plan

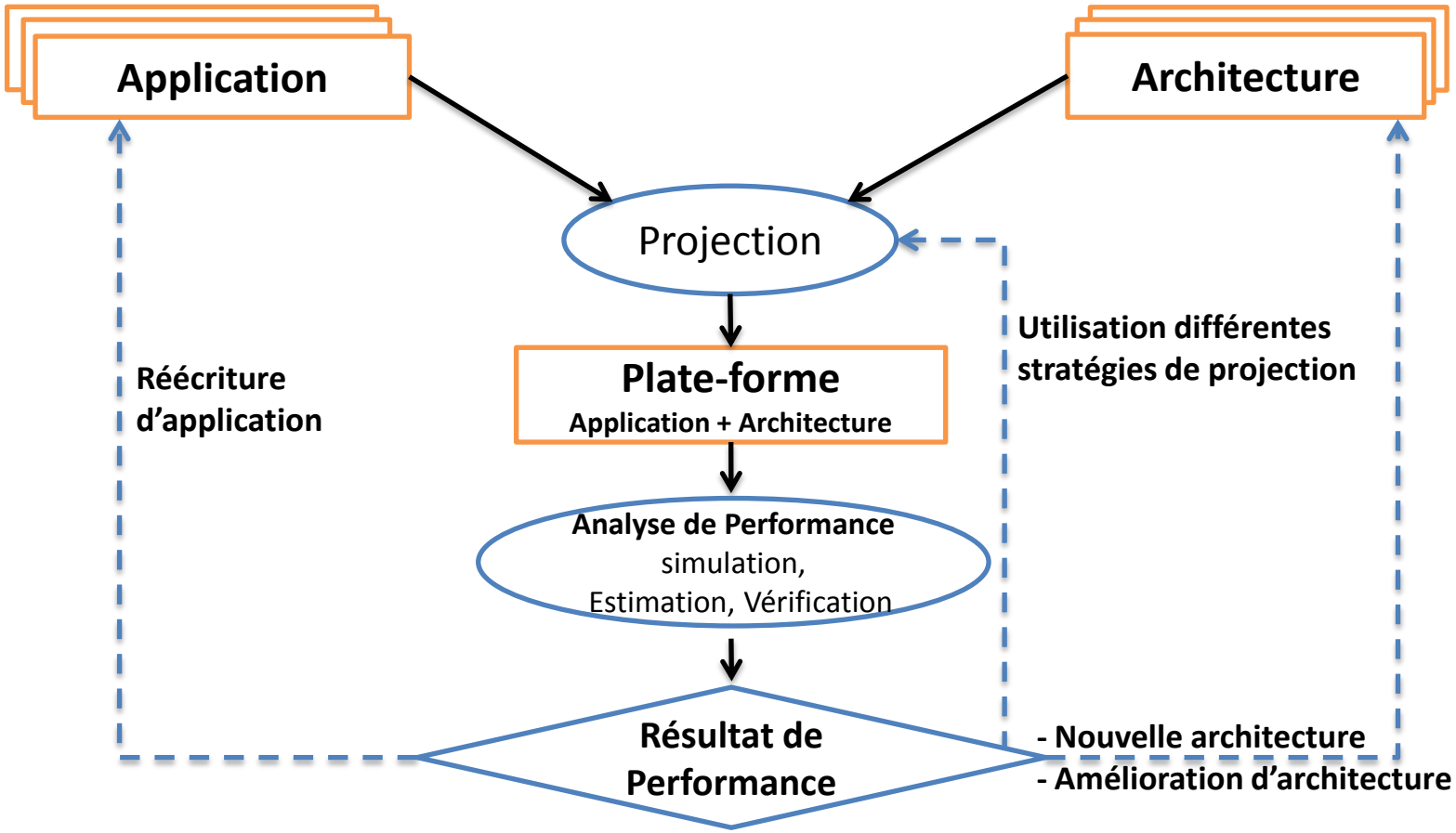
- Exploration d'architecture.
- Notre approche.
- Illustration.
- Conclusion.

Exploration d'architecture

Sélectionner parmi les différentes alternatives d'architecture celle qui répond le mieux aux objectifs et aux exigences fixés.

- Exigences fonctionnelles: non-blocage, exclusion mutuelle, …
- Exigences non-fonctionnelles: consommation, dimensionnement,… .

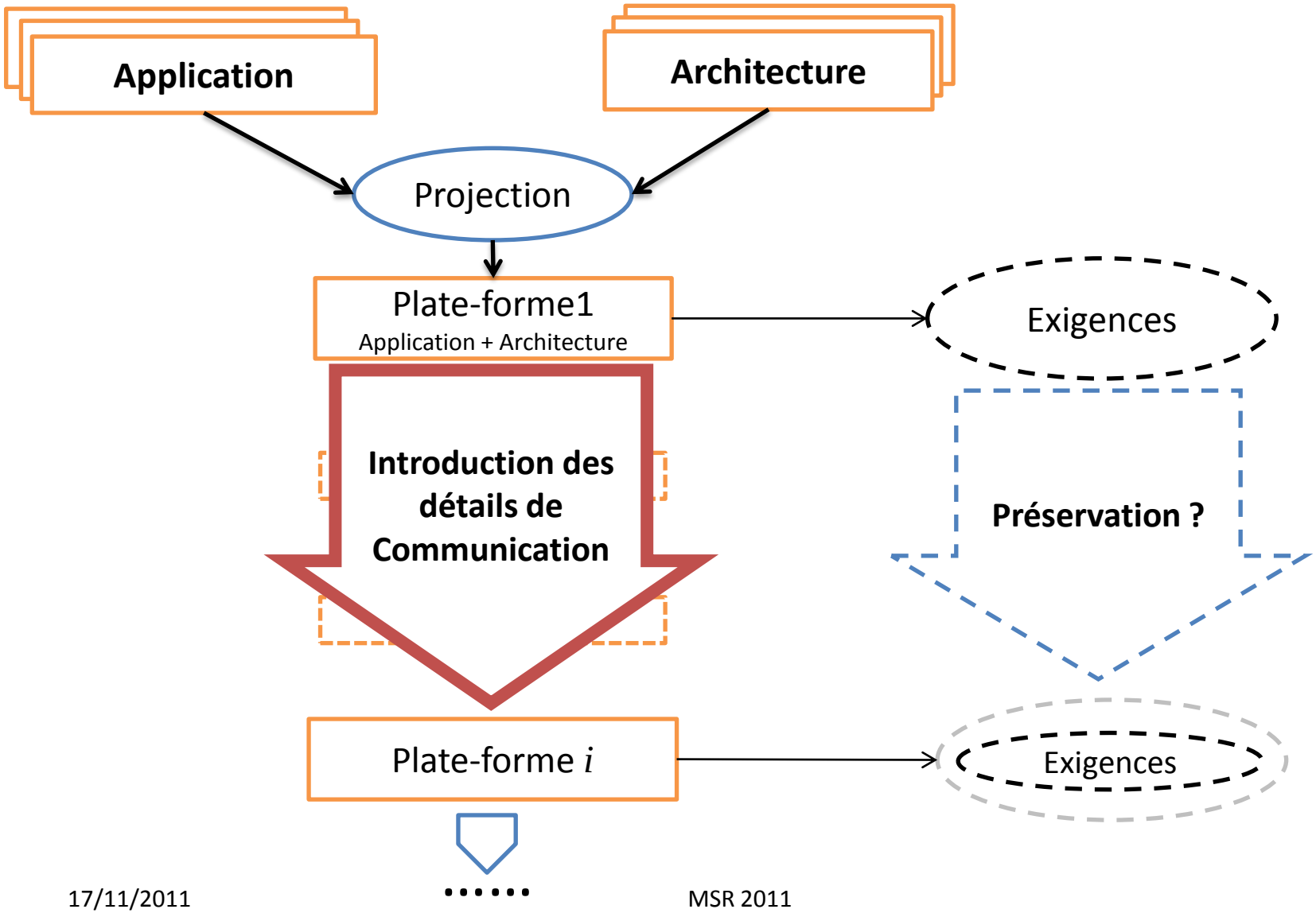
Méthodologie d'exploration d'architecture



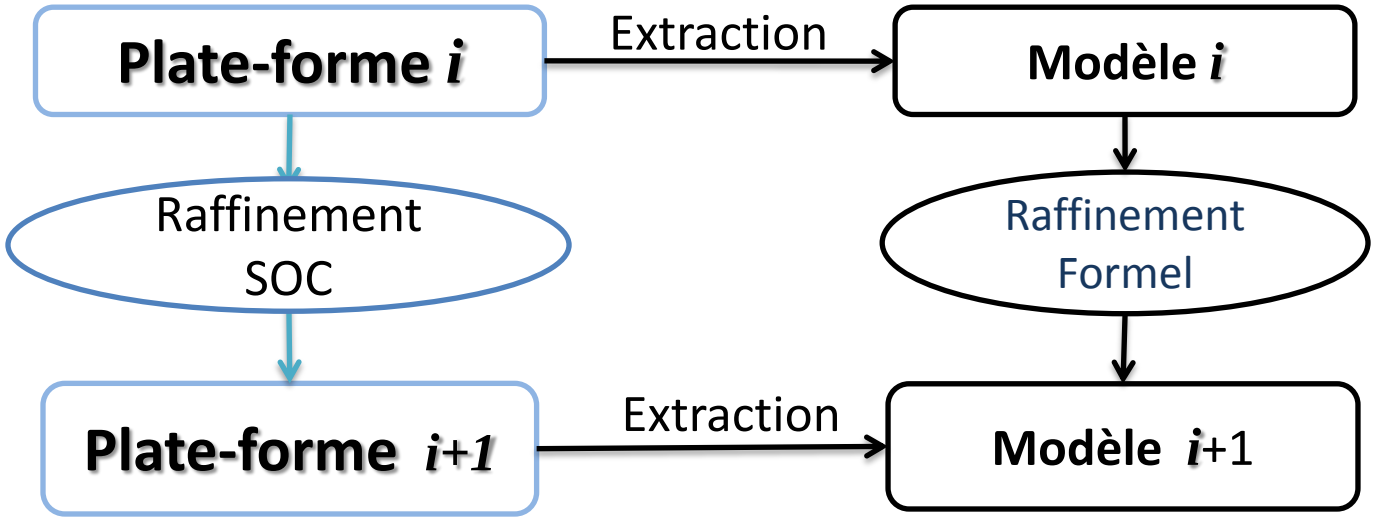
[1] Ludovic Apvrille and al. «*Abstract application modeling for system design space exploration* ». Euromicro conference on Digital System Design, 2006.

[2] Lieverse Paul and al. « *A methodology for architecture exploration of heterogeneous signal processing systems* ». Journal of VLSI Signal Processing, 2000.

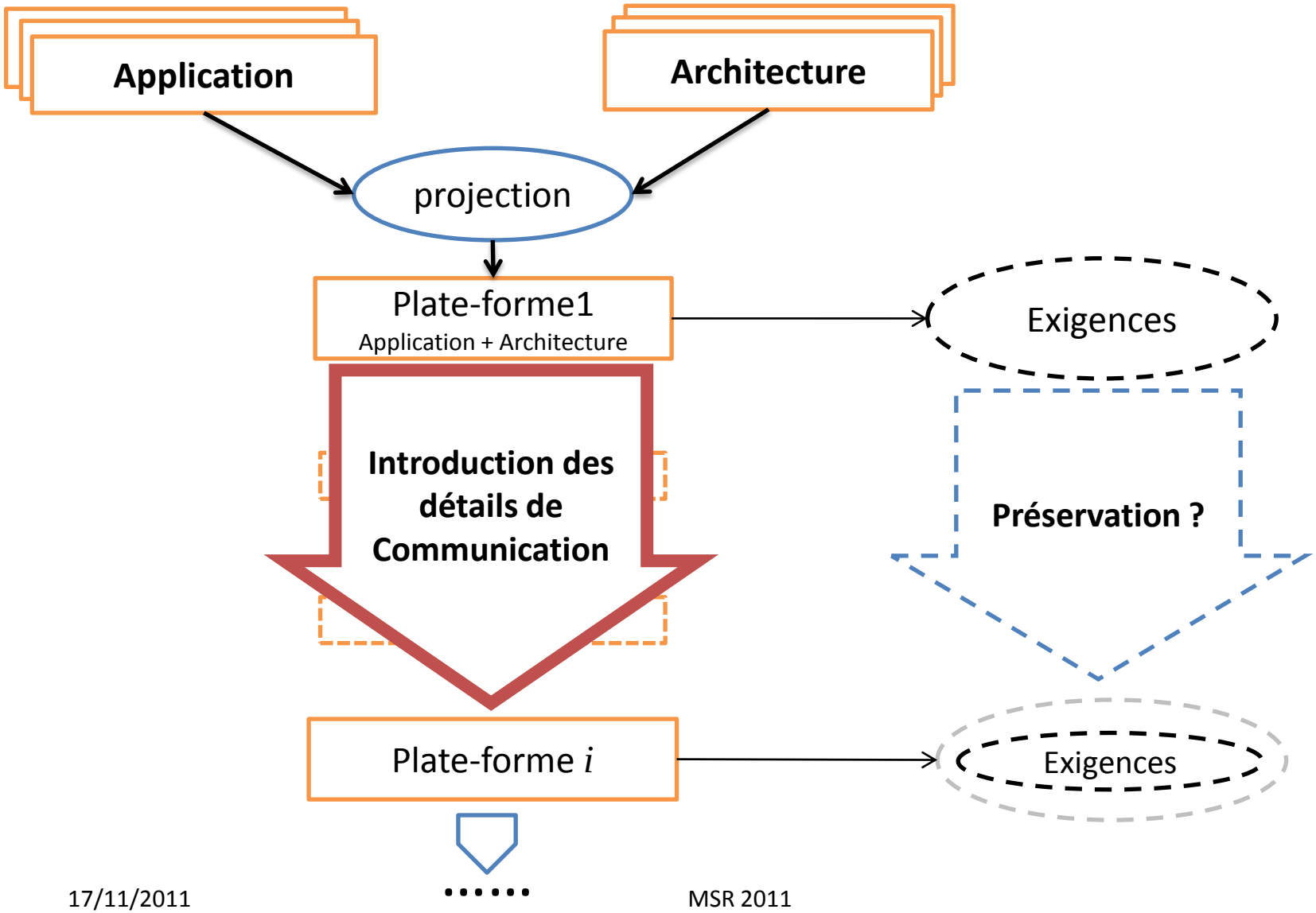
Problème de raffinement



Notre approche



Transformation en action



Task Modeling Language (TML)

- ✔ Abstraction des détails architecturaux.
- ✔ Abstraction des données de calcul.
- ✔ TML modélise une application par un réseau de tâches communiquant via des canaux.

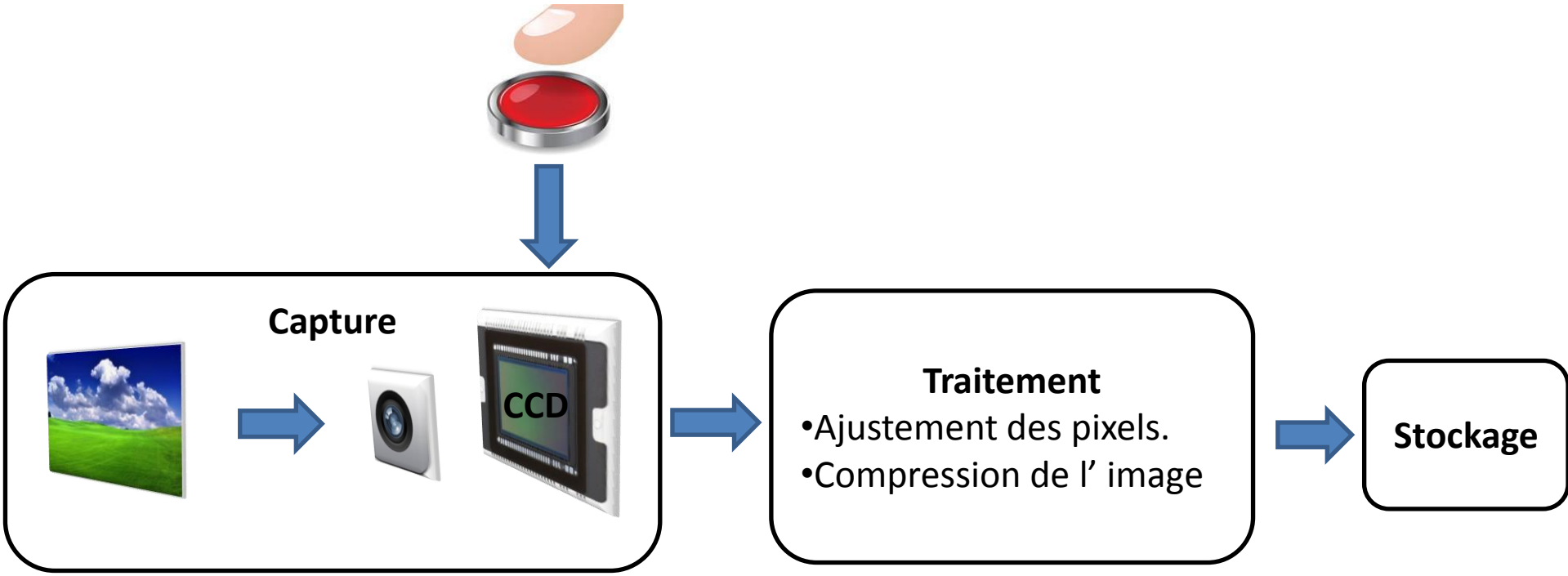
•***Tâches***: "Calculs/communications", instructions, déclaration de variables, alternatives et boucles.

•***Canaux:***

Channels, Events, Requests. (bloquant, non-bloquant, FIFO,…)

Exemple d'une application (1)

Exemple d'une application (2)



- **CCD** : Simulation de la capture de l'image. (générateur d'image)
- **CCDPP** : Ajustement des pixels.
- **CODEC**: Compression de l'image.
- **TRANS**: Simulation du transfert des données vers des périphériques externes.
- **CNTRL**: Contrôle du fonctionnement de système et l'acheminement des données.

Application TML (1)

- **Channel** *CImage1*, BRBW, CCD, CCDPP, 1;
- **Channel** *CImage2*, BRBW, CCDPP, CNTRL, 1;
- …….

- **Event** *EStart1*, FIFO, CNTRL, CCDPP, 1;
- **Event** *EStart2* , FIFO, CCDPP, CCD, 1;

- **Task** CCD {

```
Wait(EStart2);
Exec;
Write(CImage1);
}
```

- **Task** CODEC

```
{
Repeat(N times)
   Repeat(M times)
         Read(CBlock1);
         Exec;
         Write(CBlock2);
   Endrepeat
Endrepeat
}
```

- **Task** CCDPP{

```
Wait(EStart1);
Notify(EStart2);
Read(CImage1);
Exec;
Write(CImage2);
}
```

- **Task** TRANS

```
{
 Read(CImage3);
Exec;
}
```

- **Task** CNTRL

```
{
Notify (EStart1);
Read(CImage2);
Repeat(N times)
   Repeat(M times)
         Write(CBlock1);
         Read(CBlock2,);
         Exec;
   Endrepeat
Endrepeat
Write(Image3);
}
```

Application TML (2)

- **Channel** *CImage1*, BRBW, CCD, CCDPP, 1;
- **Channel** *CImage2*, BRBW, CCDPP, CNTRL, 1;
- …….

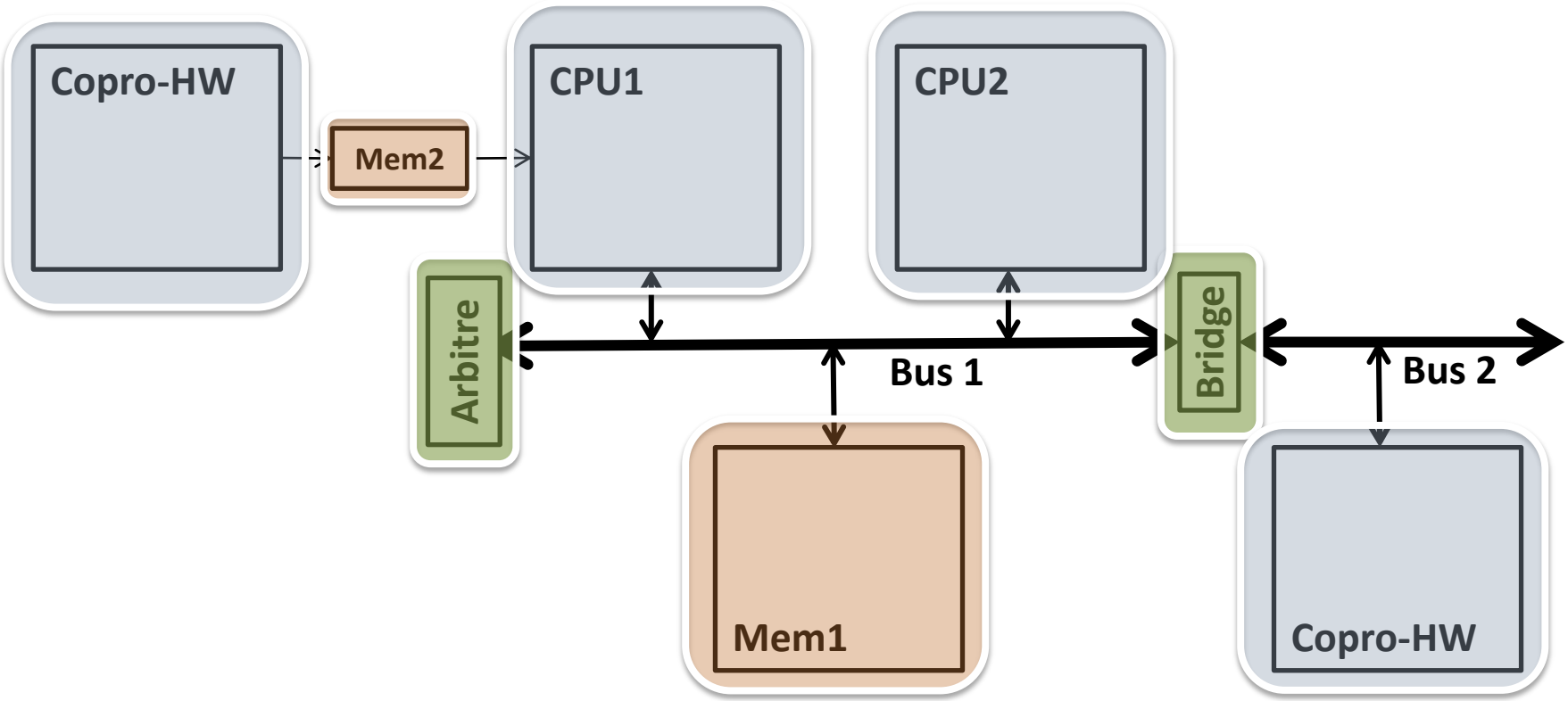
- **Event** *EStart1*, FIFO, CNTRL, CCDPP, 1;
- **Event** *EStart2* , FIFO, CCDPP, CCD, 1;

- **Task** CCD {
  Wait(EStart2);
  Exec;
  Write(CImage1);
  }
- **Task** CODEC
  {
  Repeat(N **times**)
  Repeat(M **times**)
  Read(CBlock1);
  Exec;
  Write(CBlock2);
  Endrepeat
  Endrepeat
  }

- **Task** CCDPP{
  Wait(EStart1);
  Notify(EStart2);
  Read(CImage1);
  Exec;
  Write(CImage2);
  }

Task CNTRL
{
Notify (EStart1);
Read(CImage2);
Repeat(N **times**)
Repeat(M **times**)
Write(CBlock1);
Read(CBlock2,);
Exec;
Endrepeat
Endrepeat
Write(Image3);
}

Architecture



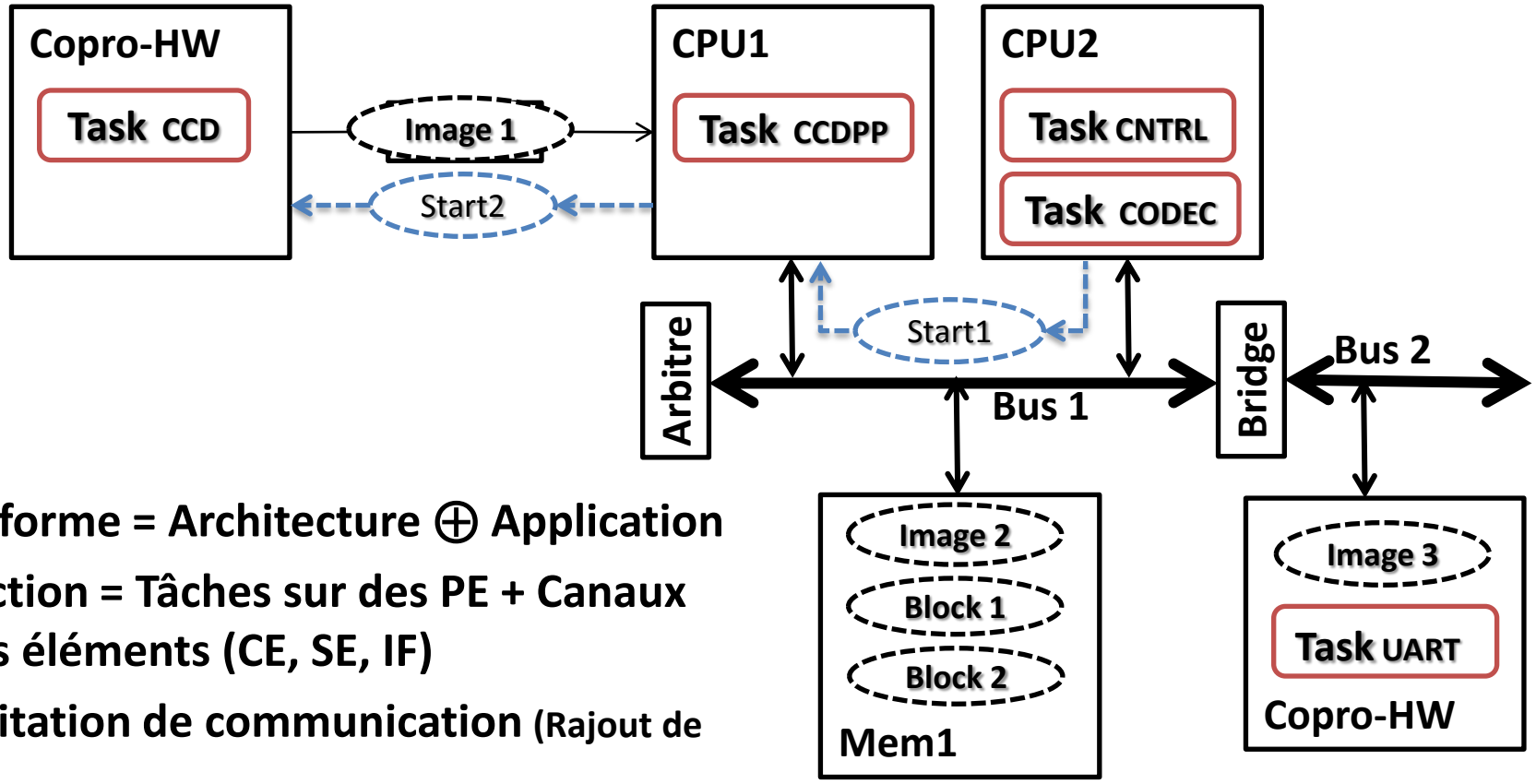
PE (Elément de calcul)

SE (Eléments de Stockage)

IF (Elément d'Interfaçage)

CE (Element de Communication)

Projection



- **Plate-forme = Architecture ⊕ Application**
- **Projection = Tâches sur des PE + Canaux sur les éléments (CE, SE, IF)**
- **Explicitation de communication (Rajout de détails)**

Modèles

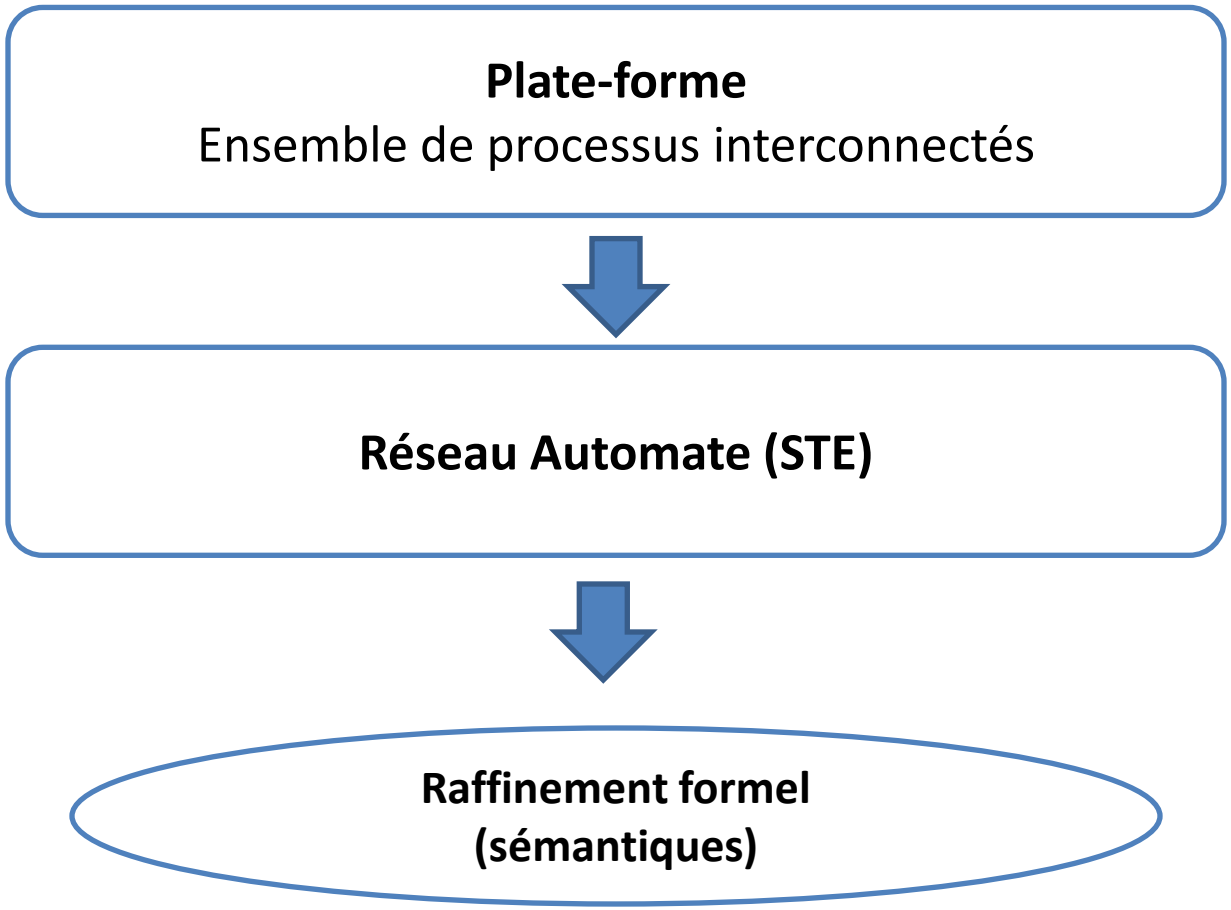


Plate-forme
Ensemble de processus interconnectés
Réseau Automate (STE)
Raffinement formel
(sémantiques)

Sémantiques

1. • Trace
   • P1 $\sqsubseteq_{Trace}$ P2 $\Leftrightarrow$ Trace(P2) $\subseteq$ Trace(P1)
2. • Complete Trace
   • P1 $\sqsubseteq_{CTrace}$ P2 $\Leftrightarrow$ Ctrace(P2) $\subseteq$ Ctrace(P1)
3. • Failure
   • P1 $\sqsubseteq_{Failure}$ P2 $\Leftrightarrow$ Failure(P2) $\subseteq$ Failure(P1)
4. • Readiness
   • P1 $\sqsubseteq_{Readiness}$ P2 $\Leftrightarrow$ Readiness(P2) $\subseteq$ Readiness(P1)

[3] J.R van Glabbeek, *The linear time-branching time spectrum I; the semantics of concrete, sequentiel processes. In Handbook of process Algebra, chapitre1, pages 3-99, Elsevier, 2001*

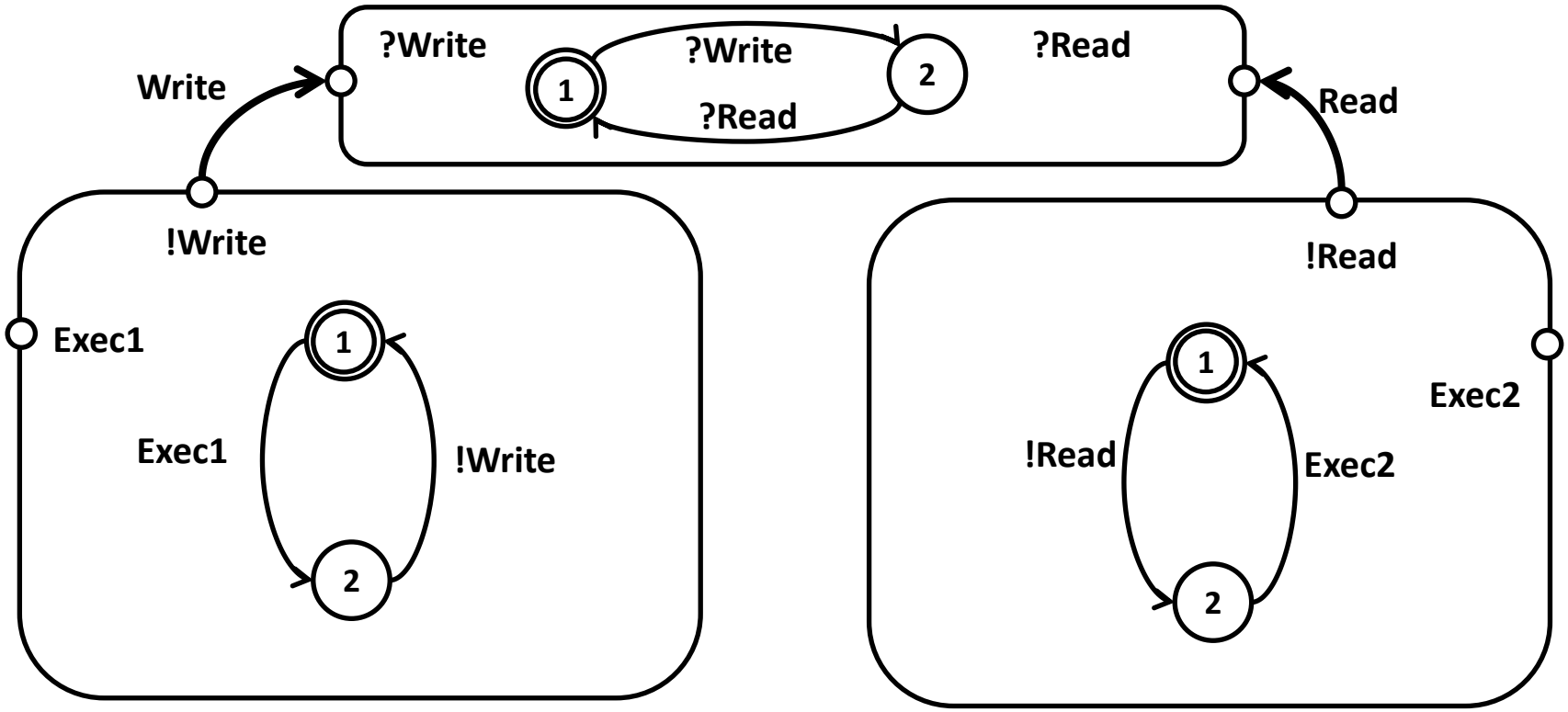
Illustration

```
CHANNEL Buffer, BRBW, Task1, Task2, 1
```

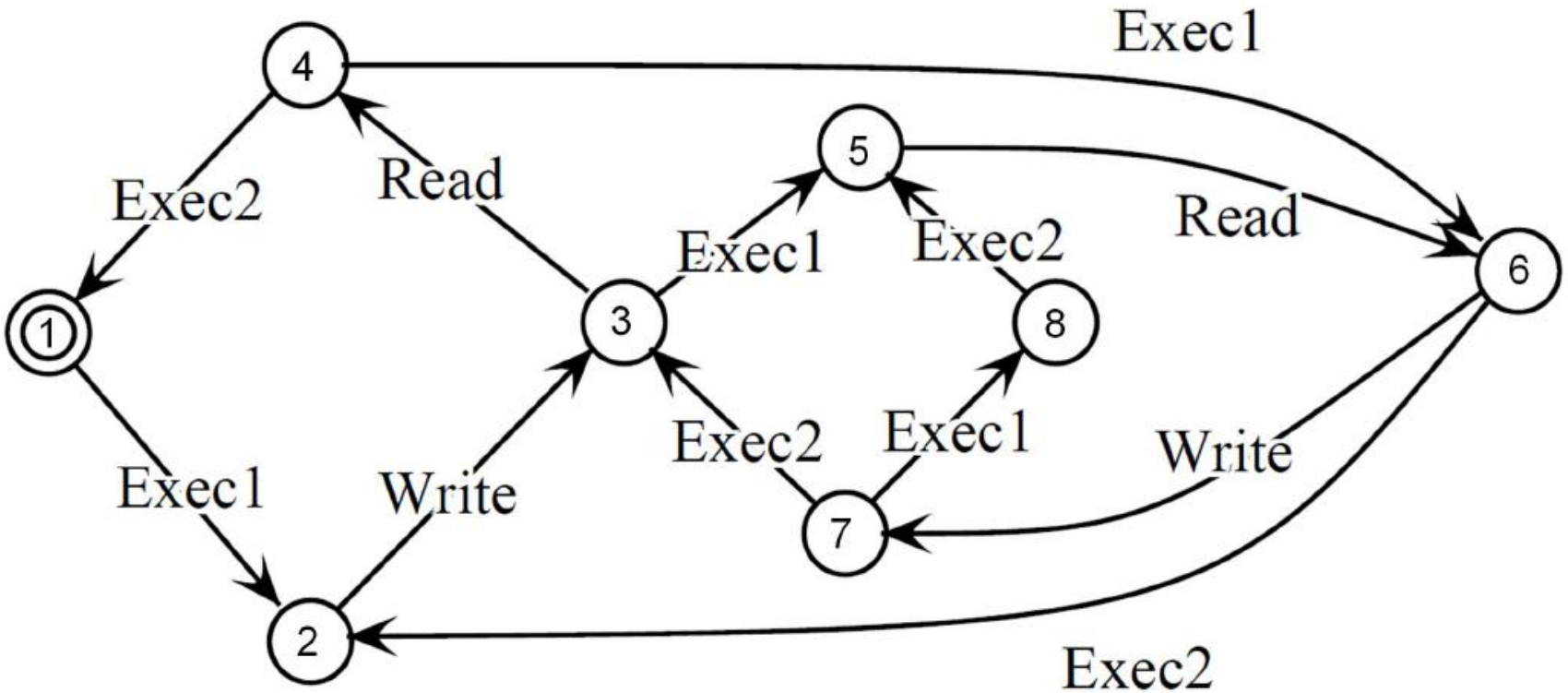
```
TASK Task1{
  REPEAT
    EXEC
    WRITE Buffer
  ENDREPEAT
}
```

```
TASK Task2{
  REPEAT
    READ Buffer
    EXEC
  ENDREPEAT
}
```

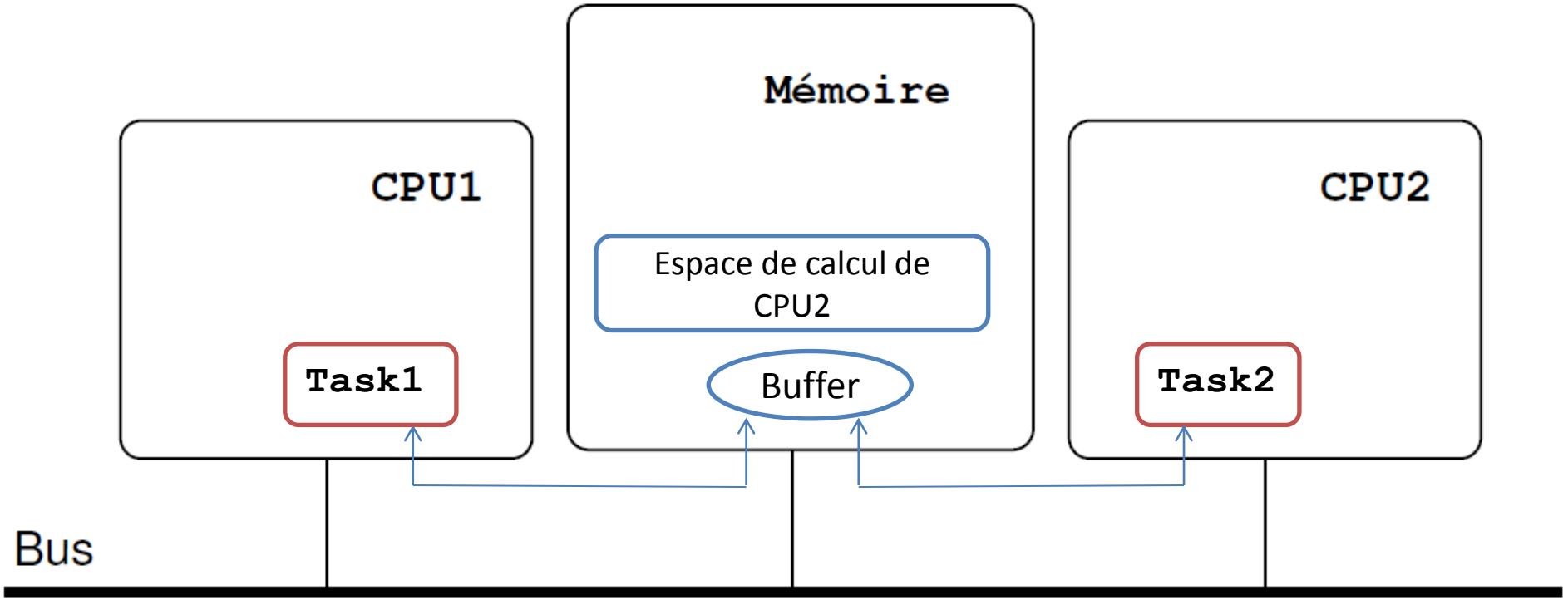
Modèle des composants



Modèle globale de l'application



Architecture et projection



Contrainte: L'espace de calcul de CPU2 est projeté sur la mémoire partagée.

Modèle abstrait VS Modèle concret

1. **Action Abstraite ≜ (Une/Séquence) d'action Concrète**

   Read ≜ CheckSignalData → LoadData → SignalRoom.

2. **Combinaison d'action abstraite ≜ Combinaison d'action concrète**

   Read; Exec ≜ CheckSignalData → LoadData → SignalRoom

   ? ? ?

   Exec Exec Exec

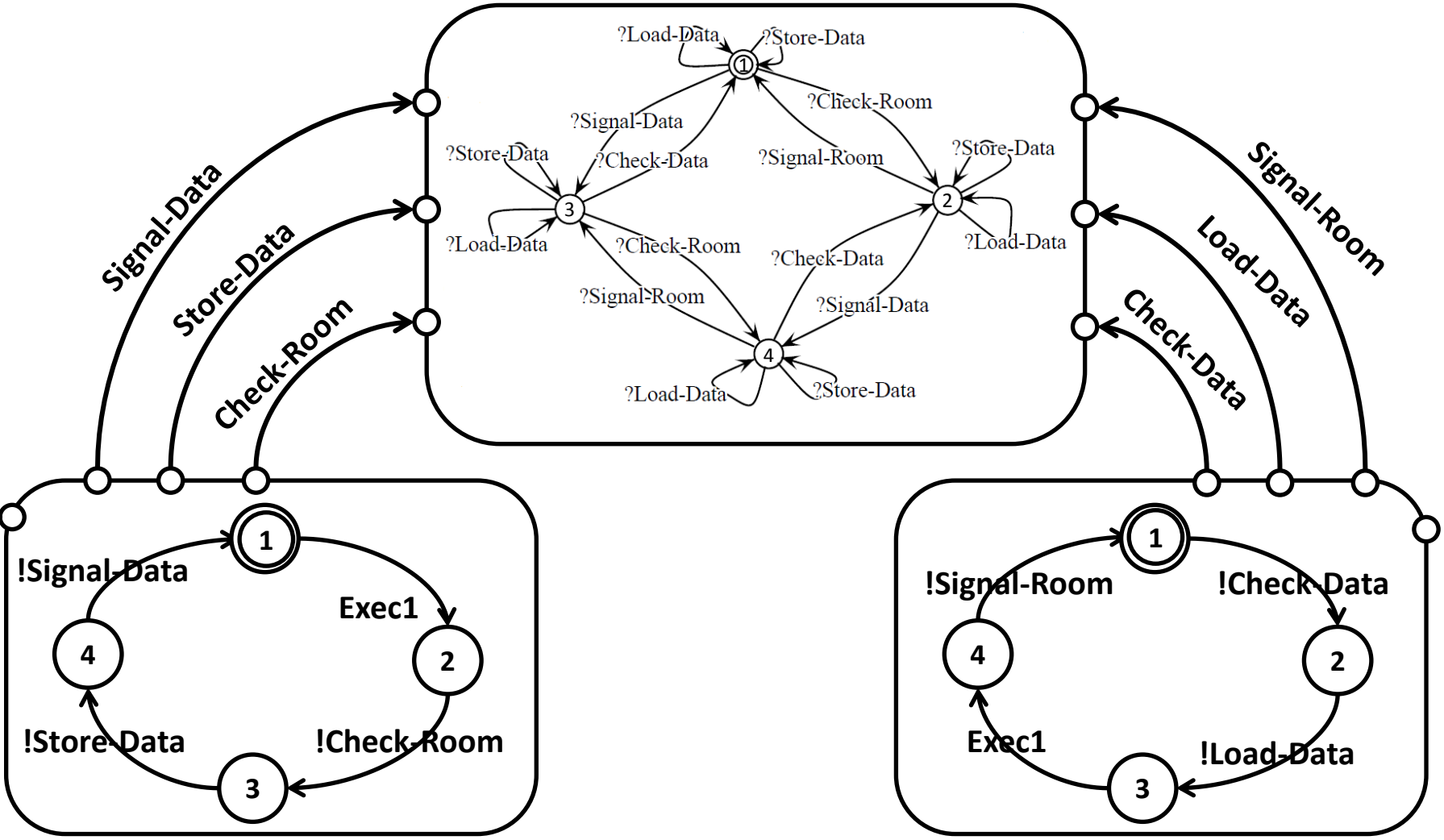
3. **Règles de correspondance**

   Read ↦ LoadData,

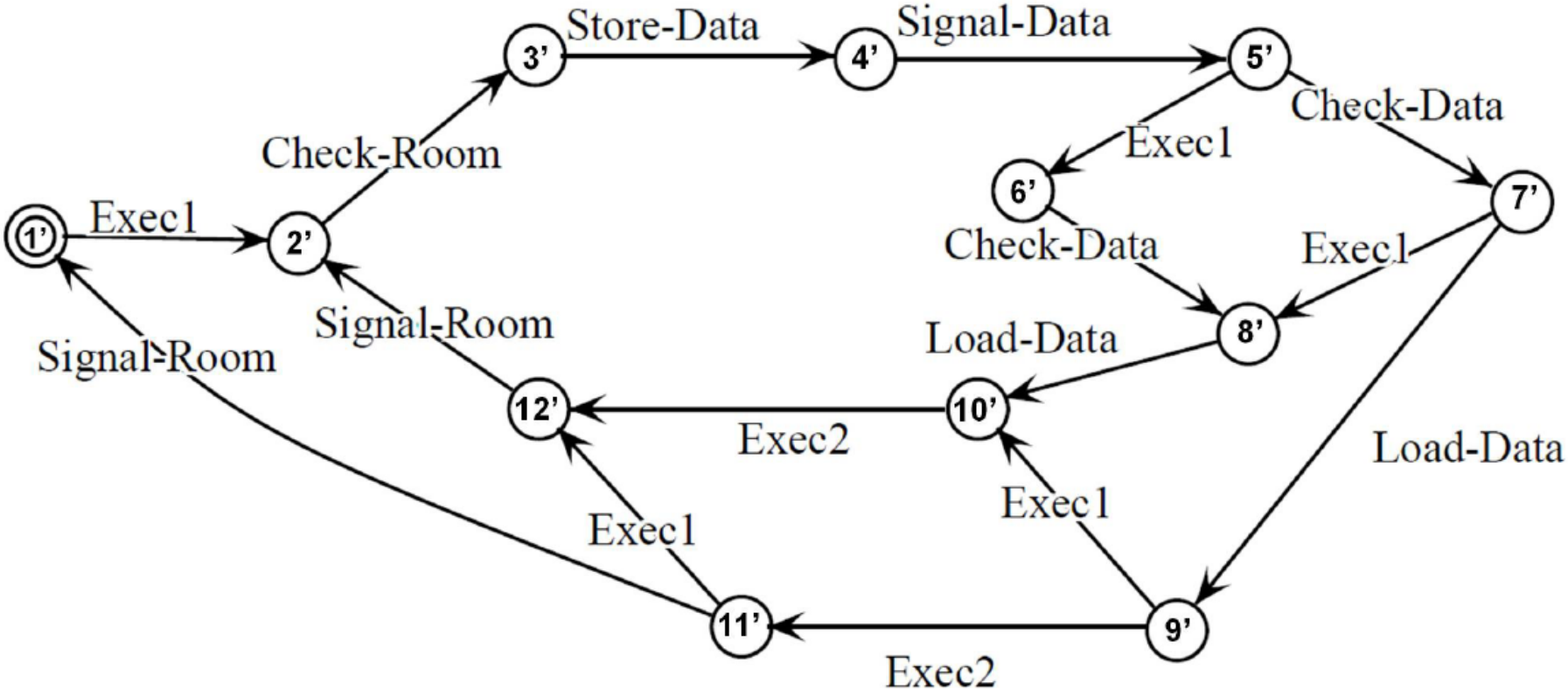
   Write ↦ StoreData,

   Exec ↦ Exec

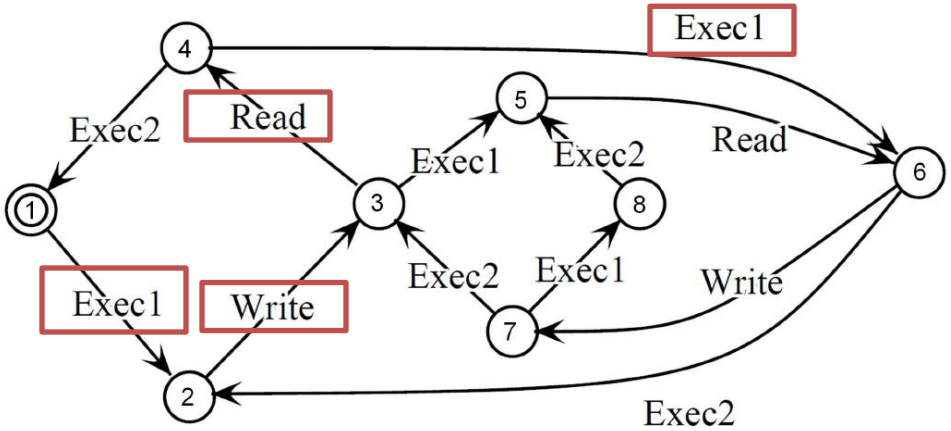
Modèle de la projection



Modèle global de la plate-forme



Analyse de raffinement



Application (Sys1)

$Sys_1 \sqsubseteq_{ICTrace} Sys_2$

$Sys_1 \not\sqsubseteq_{Failure} Sys_2$

$Sys_1 \not\sqsubseteq_{Readiness} Sys_2$

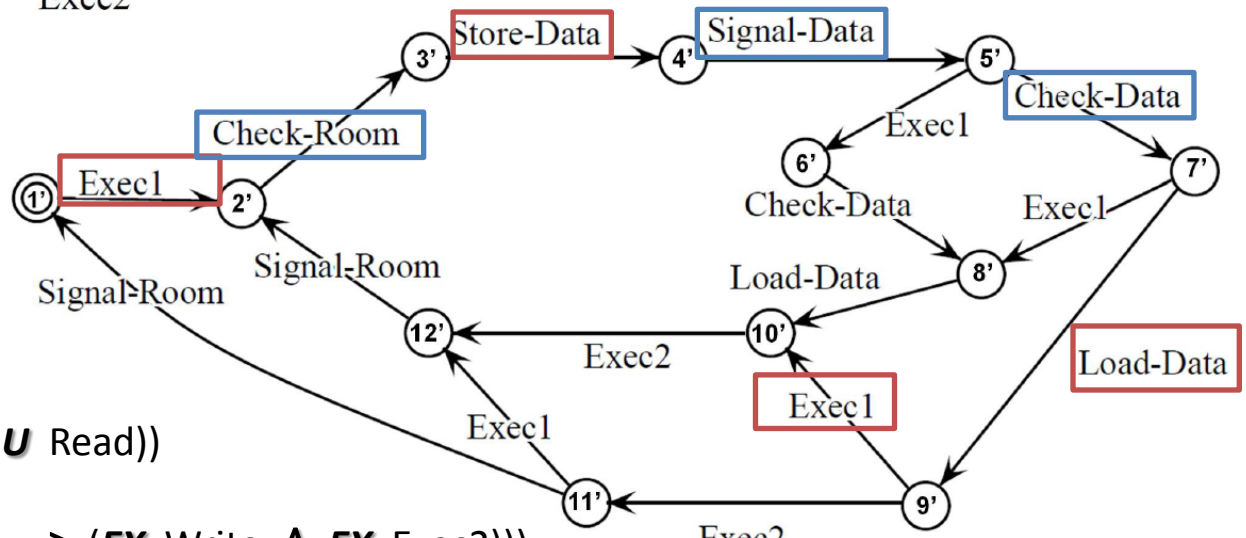


Plate-forme (Sys2)

P1: **F** (Write ⇒ **X**(¬ Write ***U*** Read))

P2: ***AF*** (Read ***∧ AX*** (Exec1 ***=>*** (***EX*** Write ***∧ EX*** Exec2)))

Conclusion

Notre objectif est de fournir un cadre rigoureux des transformations SoC effectuées entre les différents niveaux.

- Définir les transformations élémentaires de communication .
- Identifier leur ordre d'application.
- Déterminer la sémantique préservée.
- Mise en œuvre dans un outil.

Merci