



UNIVERSITÉ
PARIS-SUD 11



Soutenance de THESE

Modélisation Hétérogène Non-Hiérarchique

présentée par

Mokhoo MBOBI

17 décembre 2004

Plan

- **Systèmes Embarqués**
 - Positionnement du marché
 - Caractéristiques
 - Outils de modélisation et Modélisation hiérarchique

- **Problématique et Objectif**

- **Approche hétérogène non-hiérarchique**

- **Modélisation :**
 - Composant à Interface Hétérogène (HIC)
 - Modèle d'Exécution Hétérogène Non-Hiérarchique

- **Intégration et Validation par Simulation dans Ptolemy II**

- **Conclusion et perspectives**

Plan

■ Systèmes Embarqués

- Positionnement du marché
- Caractéristiques
- Outils de modélisation et Modélisation hiérarchique

■ Problématique et Objectif

■ Approche hétérogène non-hiérarchique

■ Modélisation :

- Composant à Interface Hétérogène (HIC)
- Modèle d'Exécution Hétérogène Non-Hiérarchique

■ Intégration et Validation par Simulation dans Ptolemy II

■ Conclusion et perspectives

Plan

■ Systèmes Embarqués

- Positionnement du marché
- Caractéristiques
- Outils de modélisation et Modélisation hiérarchique

■ Problématique et Objectif

■ Approche hétérogène non-hiérarchique

■ Modélisation :

- Composant à Interface Hétérogène (HIC)
- Modèle d'Exécution Hétérogène Non-Hiérarchique

■ Intégration et Validation par Simulation dans Ptolemy II

■ Conclusion et perspectives

Plan

- **Systèmes Embarqués**
 - Positionnement du marché
 - Caractéristiques
 - Outils de modélisation et Modélisation hiérarchique
- **Problématique et Objectif**
- **Approche hétérogène non-hiérarchique**
- **Modélisation :**
 - Composant à Interface Hétérogène (HIC)
 - Modèle d'Exécution Hétérogène Non-Hiérarchique
- **Intégration et Validation par Simulation dans Ptolemy II**
- **Conclusion et perspectives**

Plan

- **Systèmes Embarqués**
 - Positionnement du marché
 - Caractéristiques
 - Outils de modélisation et Modélisation hiérarchique
- **Problématique et Objectif**
- **Approche hétérogène non-hiérarchique**
- **Modélisation :**
 - Composant à Interface Hétérogène (HIC)
 - Modèle d'Exécution Hétérogène Non-Hiérarchique
- **Intégration et Validation par Simulation dans Ptolemy II**
- **Conclusion et perspectives**

Plan

- **Systèmes Embarqués**
 - Positionnement du marché
 - Caractéristiques
 - Outils de modélisation et Modélisation hiérarchique

- **Problématique et Objectif**

- **Approche hétérogène non-hiérarchique**

- **Modélisation :**
 - Composant à Interface Hétérogène (HIC)
 - Modèle d'Exécution Hétérogène Non-Hiérarchique

- **Intégration et Validation par Simulation dans Ptolemy II**

- **Conclusion et perspectives**

Plan

■ Systèmes Embarqués

- Positionnement du marché
- Caractéristiques
- Outils de modélisation et Modélisation hiérarchique

■ Problématique et Objectif

■ Approche hétérogène non-hiérarchique

■ Modélisation :

- Composant à Interface Hétérogène (HIC)
- Modèle d'Exécution Hétérogène Non-Hiérarchique

■ Intégration et Validation par Simulation dans Ptolemy II

■ Conclusion et perspectives

Plan

■ Systèmes Embarqués

- Positionnement du marché
- Caractéristiques
- Outils de modélisation et Modélisation hiérarchique

■ Problématique et Objectif

■ Approche hétérogène non-hiérarchique

■ Modélisation :

- Composant à Interface Hétérogène (HIC)
- Modèle d'Exécution Hétérogène Non-Hiérarchique

■ Intégration et Validation par Simulation dans Ptolemy II

■ Conclusion et perspectives

Plan

- **Systèmes Embarqués**

- Positionnement du marché
- Caractéristiques
- Outils de modélisation et Modélisation hiérarchique

- **Problématique et Objectif**

- **Approche hétérogène non-hiérarchique**

- **Modélisation :**

- Composant à Interface Hétérogène (HIC)
- Modèle d'Exécution Hétérogène Non-Hiérarchique

- **Intégration et Validation par Simulation dans Ptolemy II**

- **Conclusion et perspectives**

Plan

- **Systèmes Embarqués**

- Positionnement du marché
- Caractéristiques
- Outils de modélisation et Modélisation hiérarchique

- **Problématique et Objectif**

- **Approche hétérogène non-hiérarchique**

- **Modélisation :**

- Composant à Interface Hétérogène (HIC)
- Modèle d'Exécution Hétérogène Non-Hiérarchique

- **Intégration et Validation par Simulation dans Ptolemy II**

- **Conclusion et perspectives**

Plan

- **Systèmes Embarqués**

- Positionnement du marché
- Caractéristiques
- Outils de modélisation et Modélisation hiérarchique

- **Problématique et Objectif**

- **Approche hétérogène non-hiérarchique**

- **Modélisation :**

- Composant à Interface Hétérogène (HIC)
- Modèle d'Exécution Hétérogène Non-Hiérarchique

- **Intégration et Validation par Simulation dans Ptolemy II**

- **Conclusion et perspectives**

Plan

- **Systèmes Embarqués**

- Positionnement du marché
- Caractéristiques
- Outils de modélisation et Modélisation hiérarchique

- **Problématique et Objectif**

- **Approche hétérogène non-hiérarchique**

- **Modélisation :**

- Composant à Interface Hétérogène (HIC)
- Modèle d'Exécution Hétérogène Non-Hiérarchique

- **Intégration et Validation par Simulation dans Ptolemy II**

- **Conclusion et perspectives**

Positionnement du marché des S.E

Positionnement du marché des S.E



Positionnement du marché des S.E

SYSTEMES PERSONNELS



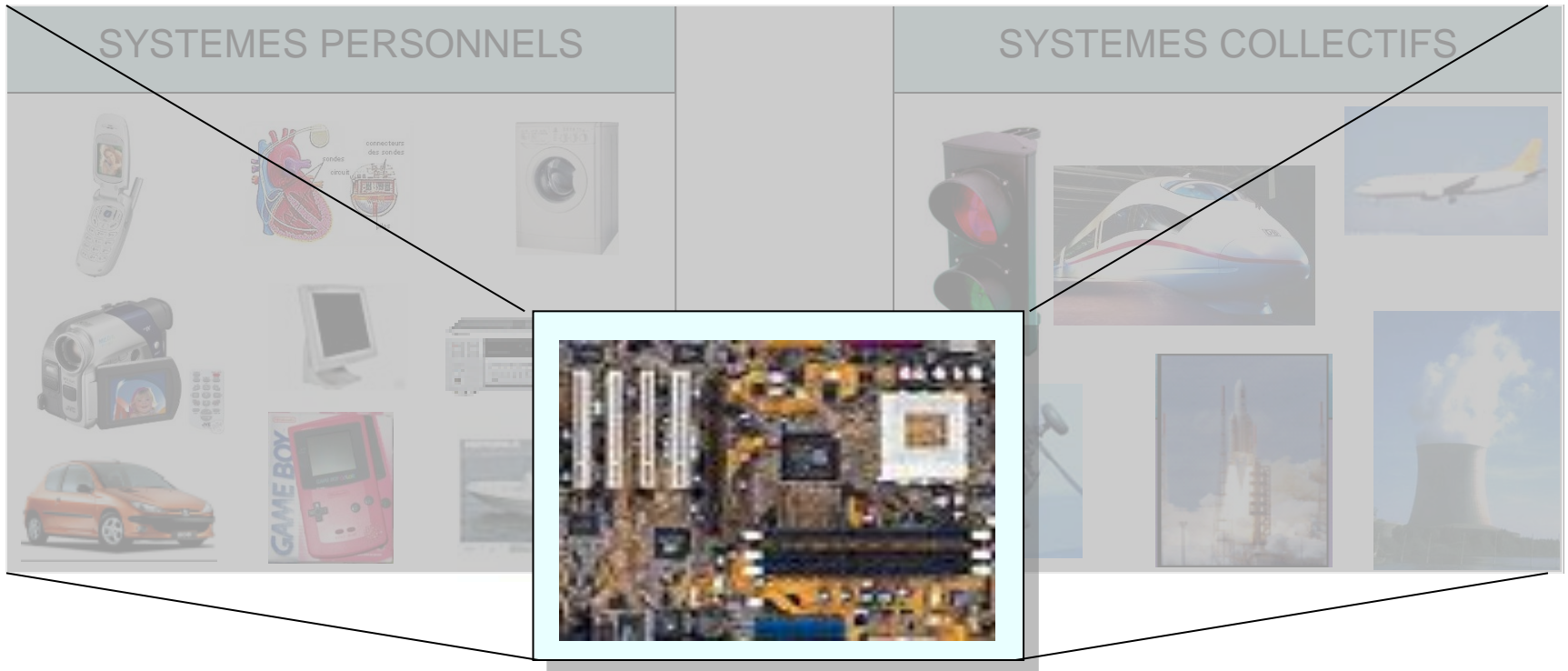
SYSTEMES COLLECTIFS



Positionnement du marché des S.E

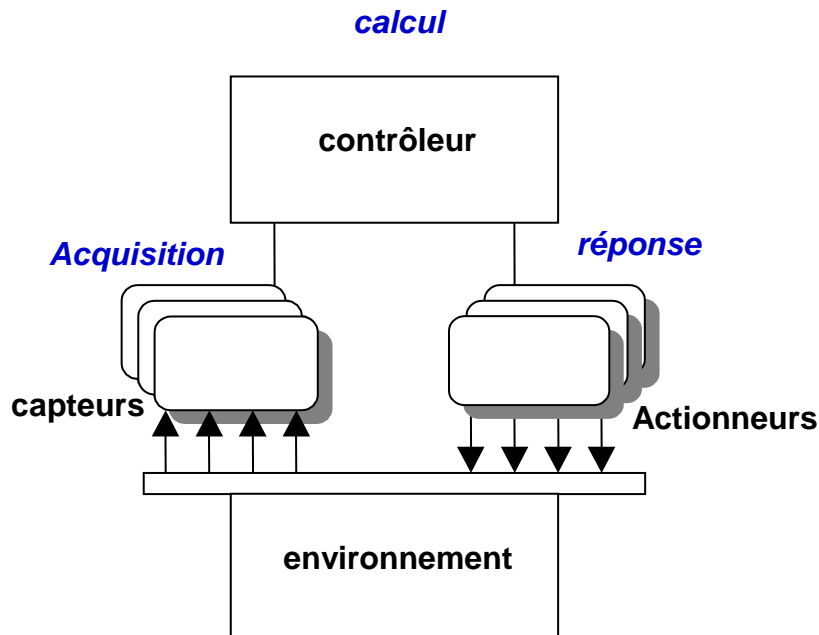


Positionnement du marché des S.E



Définition et exemples

- Dispositif matériel comportant des parties logicielles
- Utilisé pour contrôler et agir sur son environnement (système dans lequel il est logé)
- Observe les variations de son environnement grâce à des capteurs et agit sur lui grâce à des actionneurs
- Il peut varier de simple contrôleur de lave vaisselle au système complexe de guidage de missiles.



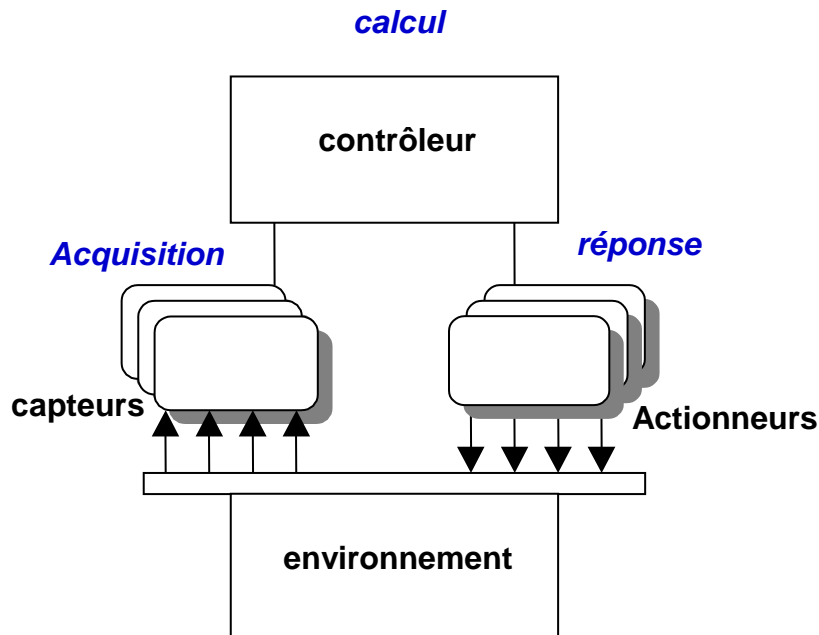
Définition et exemples

- Dispositif matériel comportant des parties logicielles
- Utilisé pour contrôler et agir sur son environnement (système dans lequel il est logé)
- Observe les variations de son environnement grâce à des capteurs et agit sur lui grâce à des actionneurs
- Il peut varier de simple contrôleur de lave vaisselle au système complexe de guidage de missiles.

Exemples



Microphone embarqué avec la sonde Mars Surveyor 98



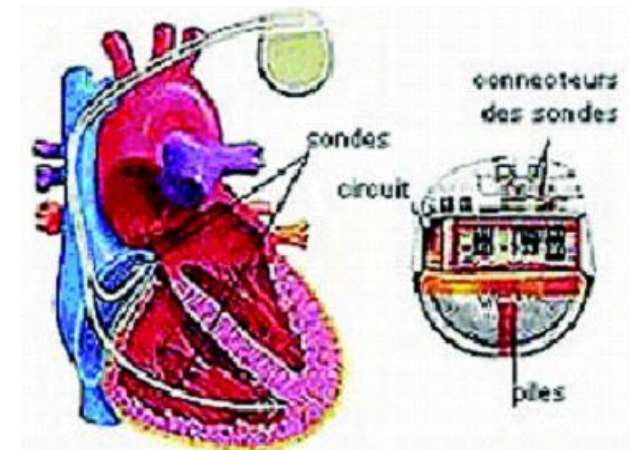
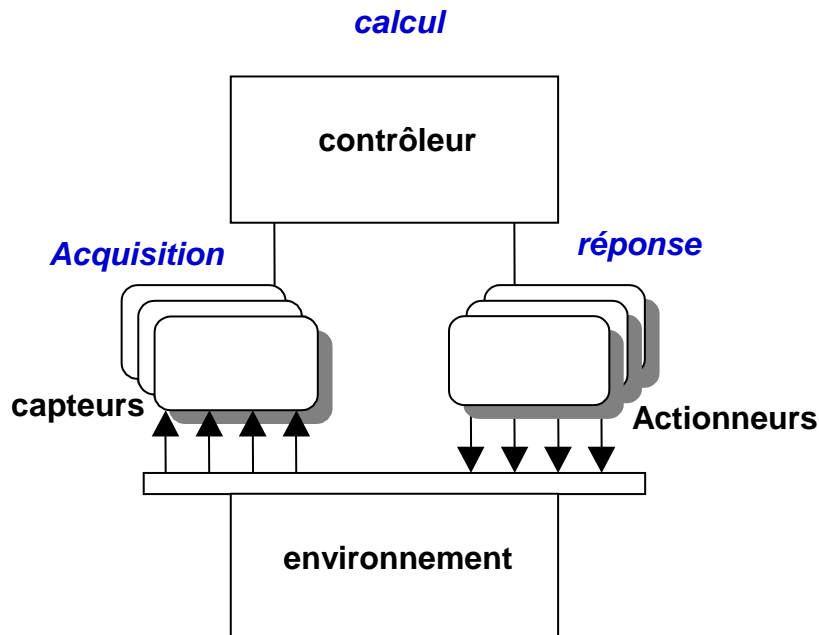
Définition et exemples

- Dispositif matériel comportant des parties logicielles
- Utilisé pour contrôler et agir sur son environnement (système dans lequel il est logé)
- Observe les variations de son environnement grâce à des capteurs et agit sur lui grâce à des actionneurs
- Il peut varier de simple contrôleur de lave vaisselle au système complexe de guidage de missiles.

■ Exemples



Microphone embarqué avec la sonde Mars Surveyor 98



Stimulateur cardiaque

Environnements de modélisation

■ Langage de modélisation

- Langages de description matérielle (VHDL, Verilog, SystemC etc)
- Langages de description des architectures (SpecC, Superlog, etc)
- Langages orientés objet pour l'analyse et la conception (UML+SDL)
- Langages pour la modélisation des systèmes réactifs (Lustre, Esterel, SDL, etc)
- Langages de programmation (OpenJ, JavaTime etc)

Environnements de modélisation

■ Langage de modélisation

- Langages de description matérielle (VHDL, Verilog, SystemC etc)
- Langages de description des architectures (SpecC, Superlog, etc)
- Langages orientés objet pour l'analyse et la conception (UML+SDL)
- Langages pour la modélisation des systèmes réactifs (Lustre, Esterel, SDL, etc)
- Langages de programmation (OpenJ, JavaTime etc)

■ Plates-formes de modélisation

- | | | |
|--------------|------------|------------|
| - SpecC | - Simulink | - Polis |
| - SystemC | - Space | - Music |
| - Rosetta | - Coware | - el Greco |
| - Ptolemy II | | |

Caractéristiques

- **Hétérogénéité** : Un seul système implique plusieurs domaines techniques
 - Optique, Mécanique, Thermodynamique, Traitement des signaux et des images
Electronique analogique ou numérique, etc.

Caractéristiques

- **Hétérogénéité** : Un seul système implique plusieurs domaines techniques
 - Optique, Mécanique, Thermodynamique, Traitement des signaux et des images
Electronique analogique ou numérique, etc.

- **Modèles de calcul**
 - Chaque domaine technique obéit à un ensemble de lois physiques ou axiomes qui expriment des contraintes sur les propriétés des composants en fonction des relations entre composants
 - Ces lois sont appelées « *Modèle de Calcul ou Model of Computation (MoC)* »

Caractéristiques

- **Hétérogénéité** : Un seul système implique plusieurs domaines techniques
 - Optique, Mécanique, Thermodynamique, Traitement des signaux et des images
Electronique analogique ou numérique, etc.

- **Modèles de calcul**
 - Chaque domaine technique obéit à un ensemble de lois physiques ou axiomes qui expriment des contraintes sur les propriétés des composants en fonction des relations entre composants
 - Ces lois sont appelées « *Modèle de Calcul ou Model of Computation (MoC)* »

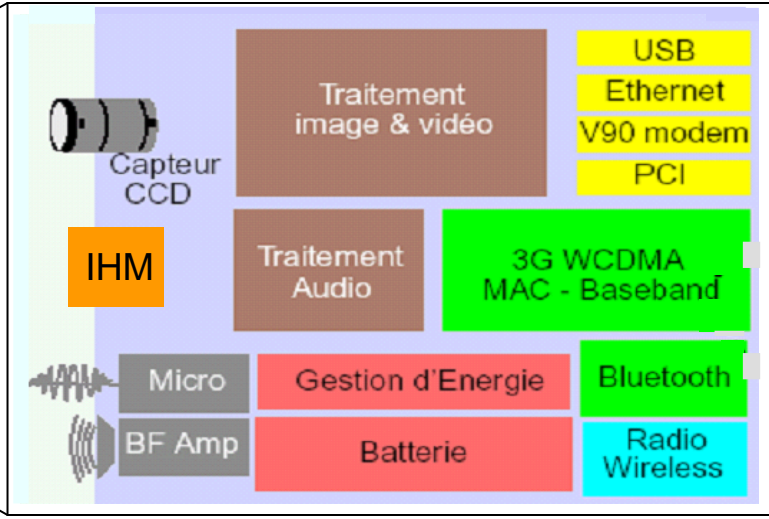
 - Exemples :
 - CT (Continuous Time) : Syst. physique, dynamique continue, équations différentielles
 - DE (Discrete Events) : Réseaux de communication, modèles statistiques
 - FSM (Finite State Machines) : Logique de contrôle
 - SDF (Synchronous Data Flow) : Flots de données, traitement du signal
 - SR (Synchronous Reactive) : Systèmes réactifs, logique de contrôle

- Exemple d'Hétérogénéité : Téléphone cellulaire multimédia 3G

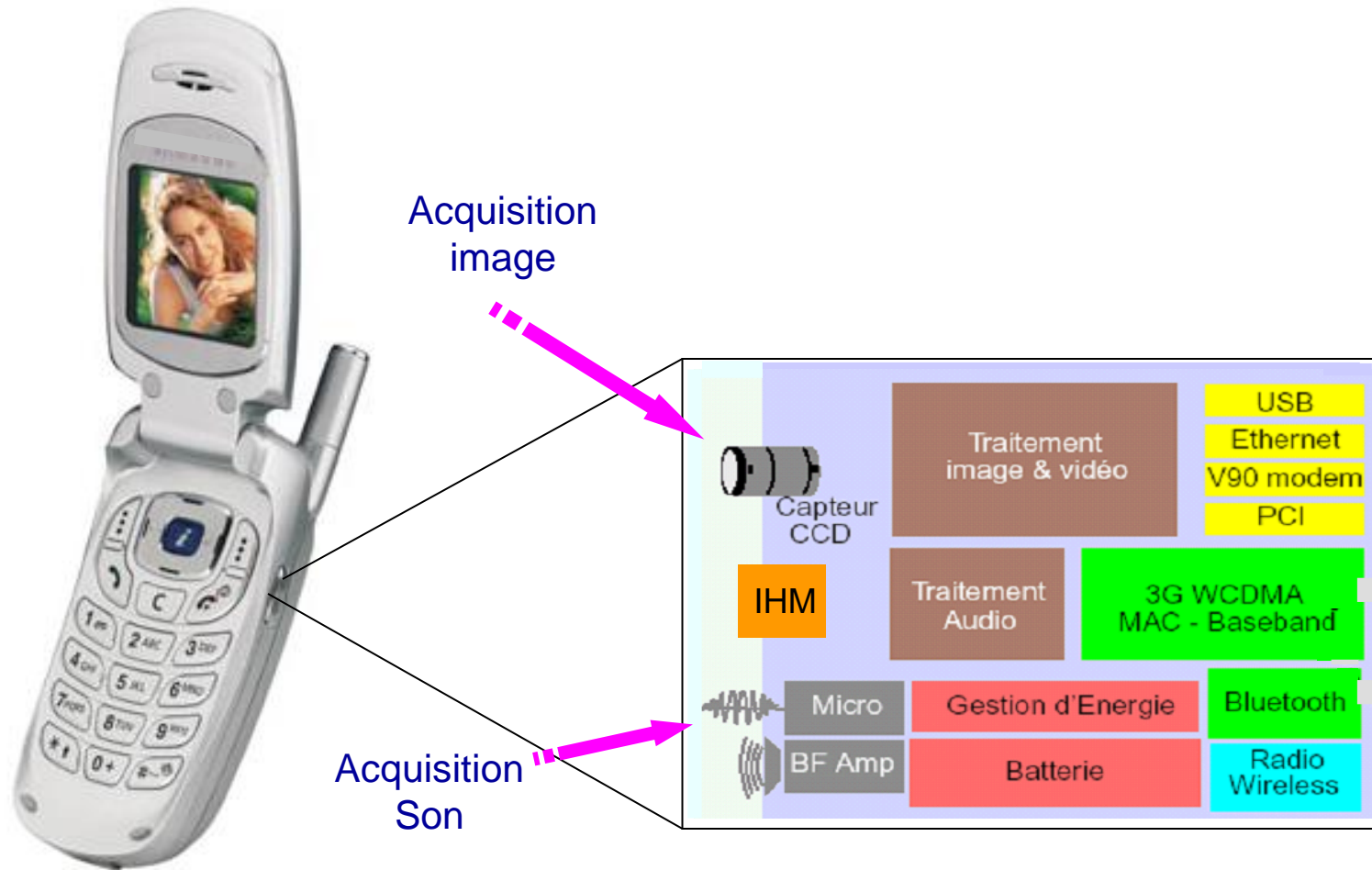
■ Exemple d'Hétérogénéité : Téléphone cellulaire multimédia 3G



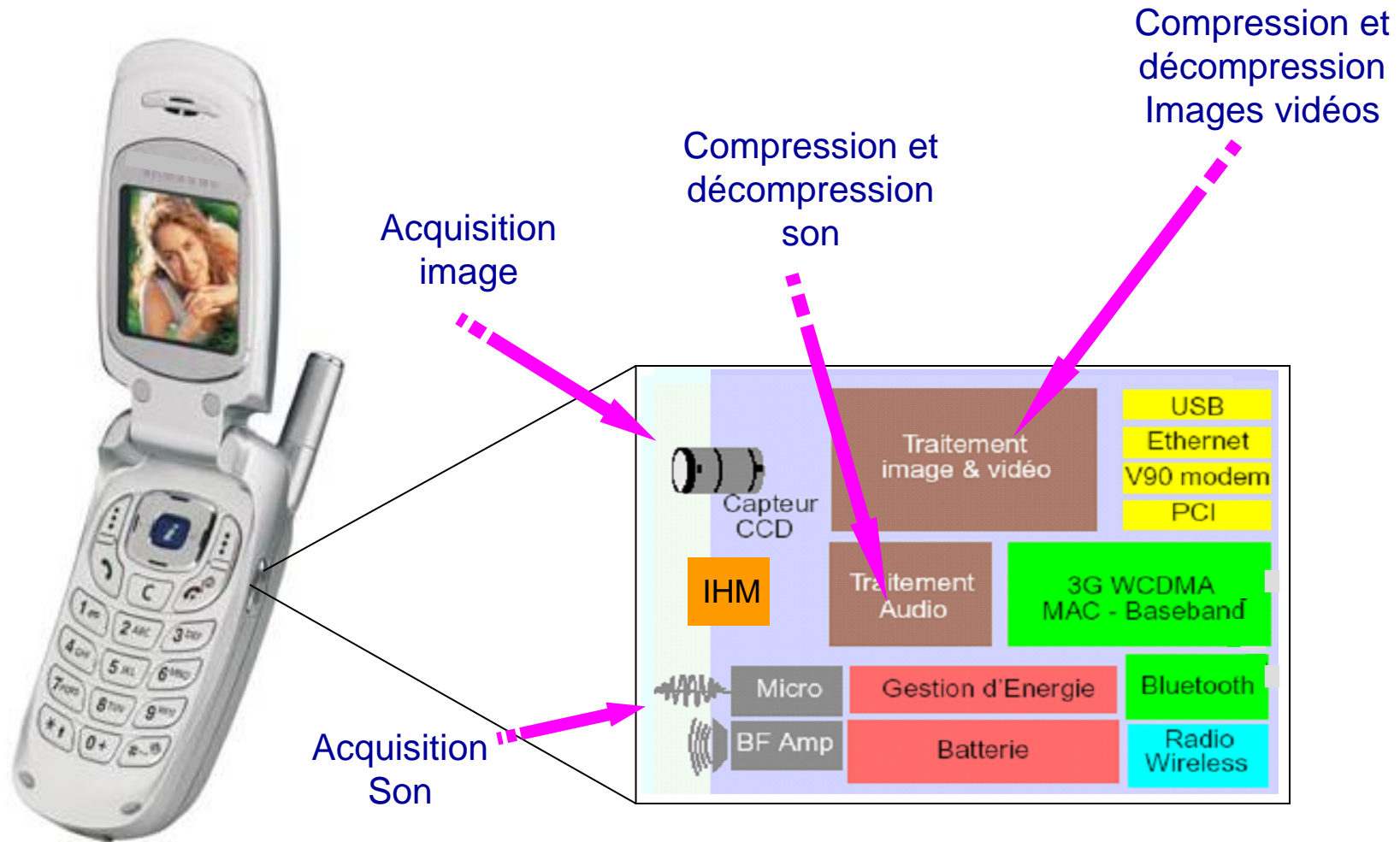
■ Exemple d'Hétérogénéité : Téléphone cellulaire multimédia 3G



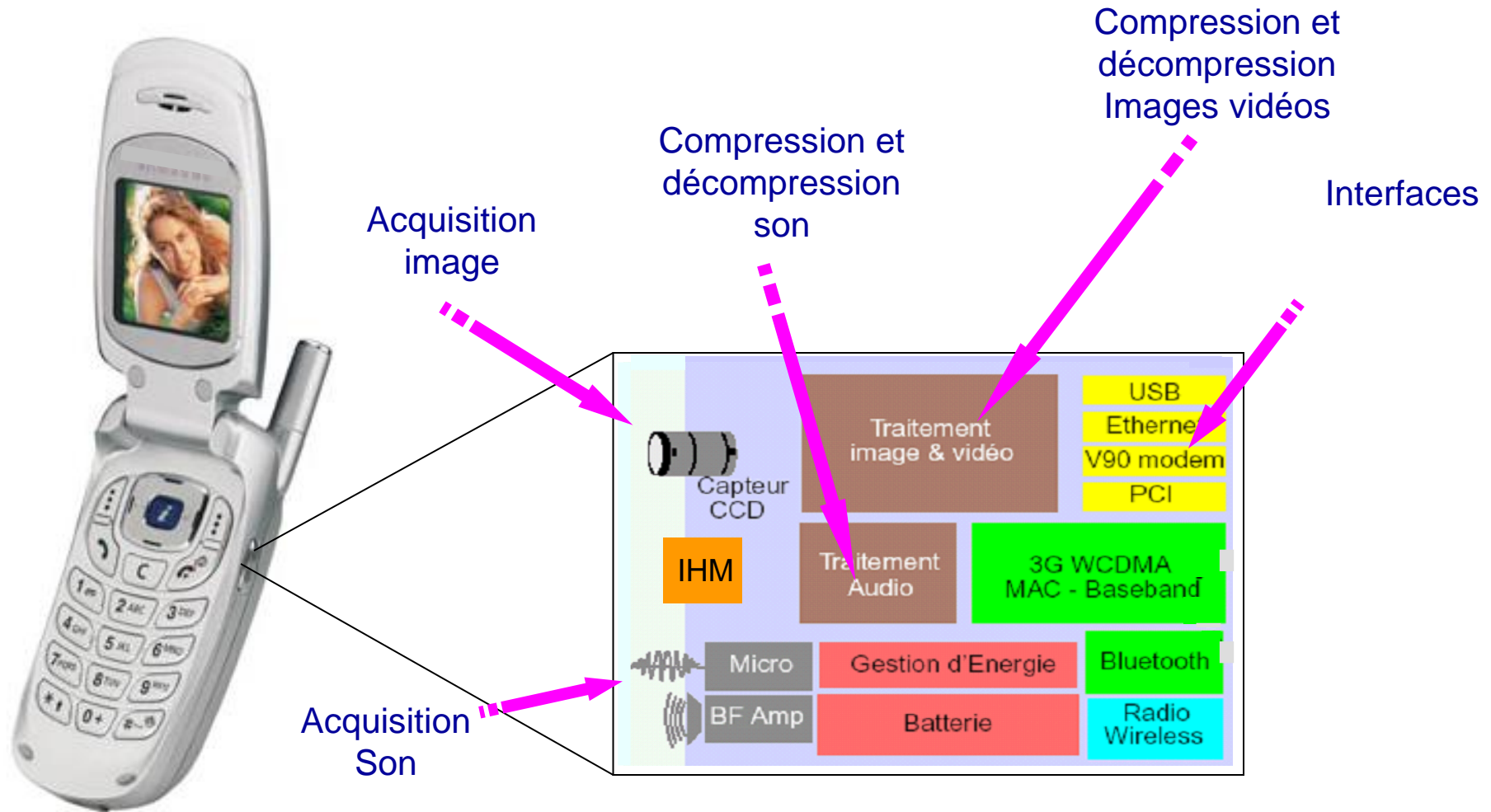
■ Exemple d'Hétérogénéité : Téléphone cellulaire multimédia 3G



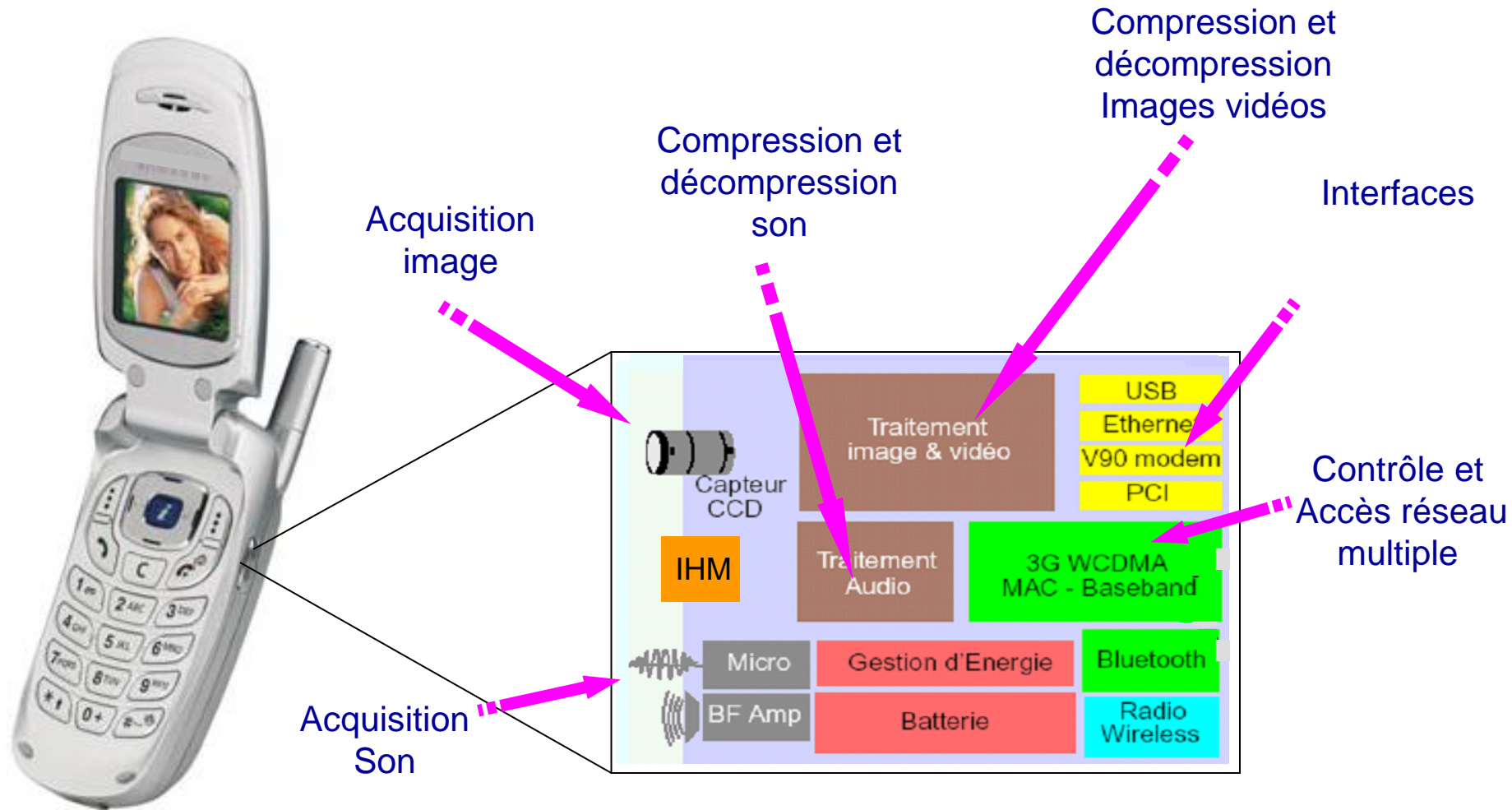
■ Exemple d'Hétérogénéité : Téléphone cellulaire multimédia 3G



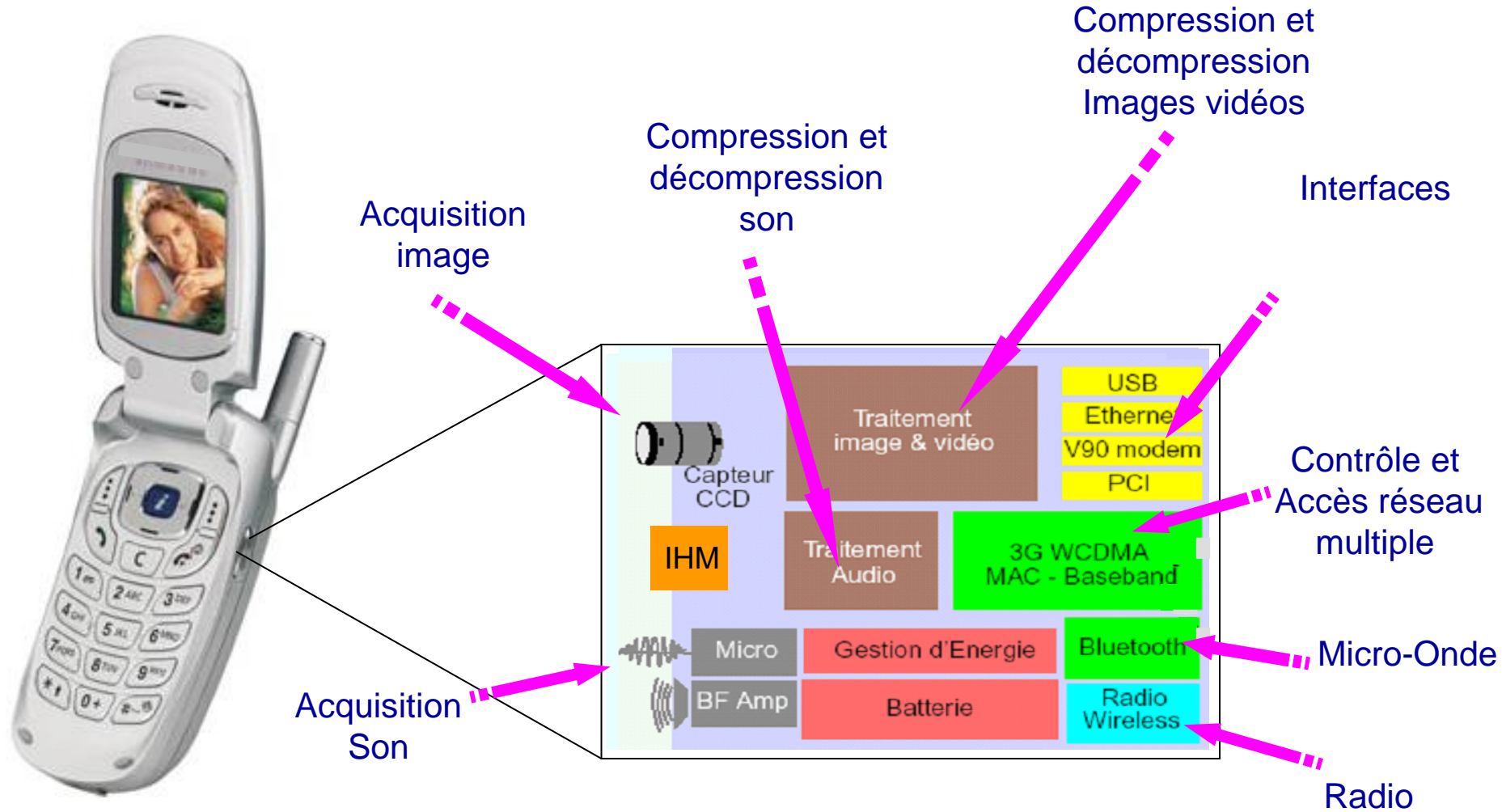
■ Exemple d'Hétérogénéité : Téléphone cellulaire multimédia 3G



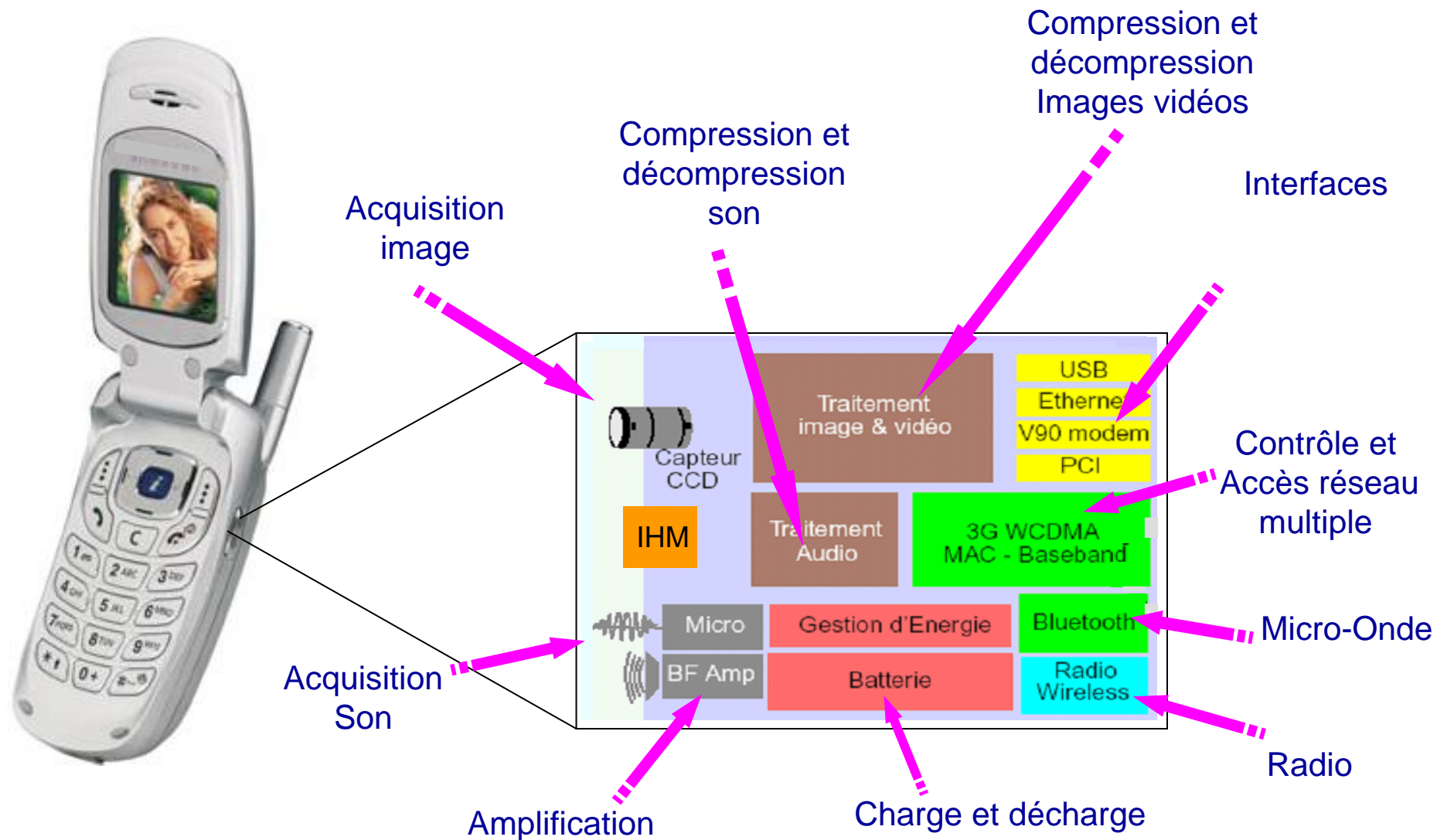
Exemple d'Hétérogénéité : Téléphone cellulaire multimédia 3G



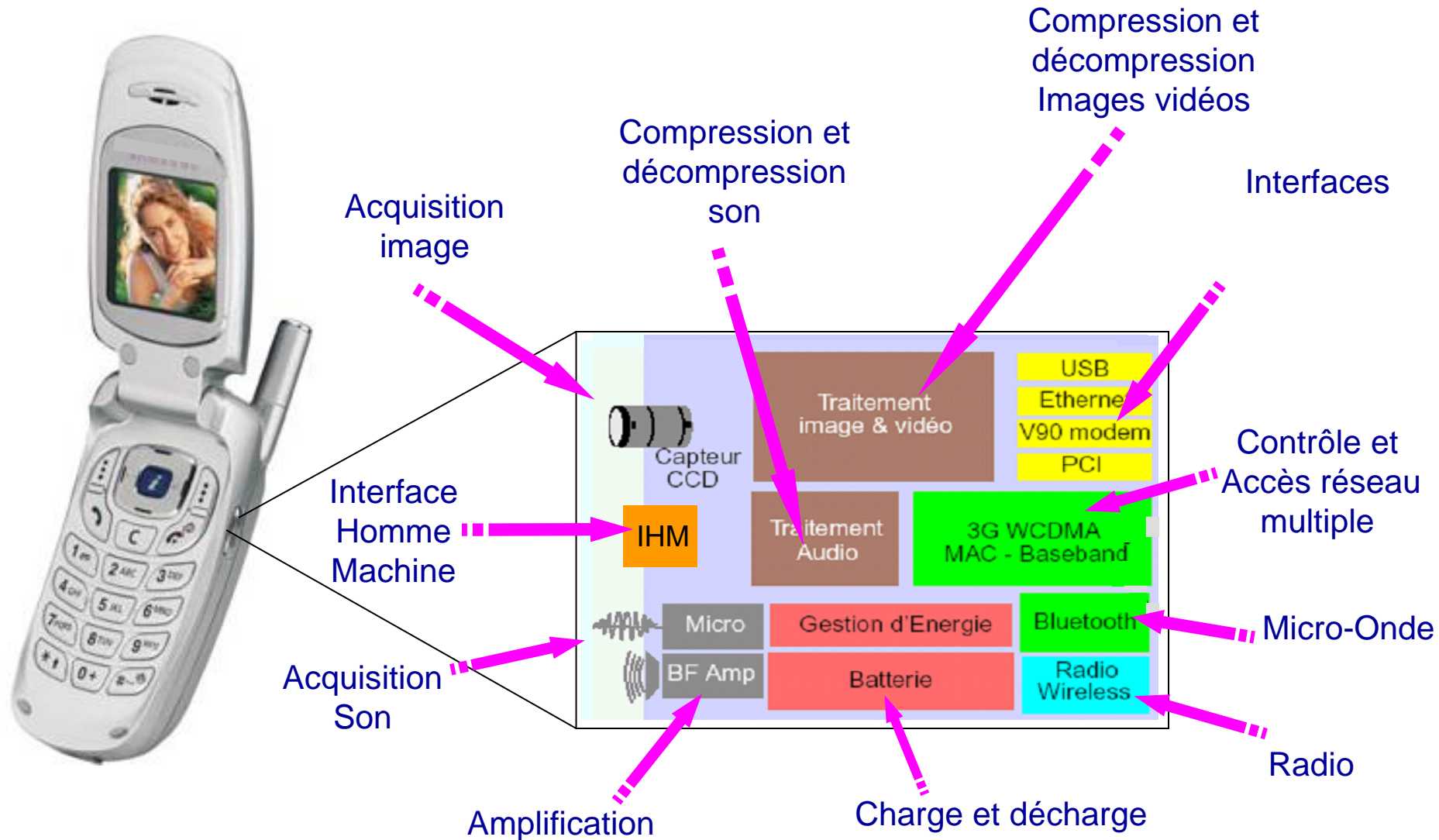
Exemple d'Hétérogénéité : Téléphone cellulaire multimédia 3G



■ Exemple d'Hétérogénéité : Téléphone cellulaire multimédia 3G



Exemple d'Hétérogénéité : Téléphone cellulaire multimédia 3G



Approches existantes de l'Hétérogénéité

■ Hétérogénéité Amorphe

(Pour un ensemble de MoCs fixé)

- Plusieurs environnements de modélisation se concentrent sur un petit ensemble donné de MoCs

- Exemple : signaux continus et discrets pour l'ingénierie électrique et des machines à états et des équations différentielles pour les systèmes hybrides

- La connaissance *a priori* des MoCs permet d'en définir l'union

- La connaissance totale des interactions entre MoCs permet une modélisation complète allant jusqu'à l'implémentation

- Exemple : CAN entrées continues et sorties discrètes.

Exemples d'environnements :

- SIMULINK et VHDL-AMS

Approches existantes de l'Hétérogénéité

■ Hétérogénéité Amorphe (Pour un ensemble de MoCs fixé)

- Plusieurs environnements de modélisation se concentrent sur un petit ensemble donné de MoCs
- Exemple : signaux continus et discrets pour l'ingénierie électrique et des machines à états et des équations différentielles pour les systèmes hybrides
- La connaissance *a priori* des MoCs permet d'en définir l'union
- La connaissance totale des interactions entre MoCs permet une modélisation complète allant jusqu'à l'implémentation
- Exemple : CAN entrées continues et sorties discrètes.

Exemples d'environnements :

- SIMULINK et VHDL-AMS

■ Hétérogénéité Hiérarchique (Pour un ensemble ouvert de MoCs)

- On ne peut pas construire l'union des MoCs *a priori*
- Un composant n'obéit qu'à un seul MoC.
- L'abstraction hiérarchique permet d'utiliser un Moc interne à un composant différent du MoC externe
- Les changements de MoC ne peuvent s'opérer qu'à la frontière d'un composant.

Exemple d'environnement : Ptolemy II

Approches existantes de l'Hétérogénéité

■ Hétérogénéité Amorphe (Pour un ensemble de MoCs fixé)

- Plusieurs environnements de modélisation se concentrent sur un petit ensemble donné de MoCs
- Exemple : signaux continus et discrets pour l'ingénierie électrique et des machines à états et des équations différentielles pour les systèmes hybrides
- La connaissance *a priori* des MoCs permet d'en définir l'union
- La connaissance totale des interactions entre MoCs permet une modélisation complète allant jusqu'à l'implémentation
- Exemple : CAN entrées continues et sorties discrètes.

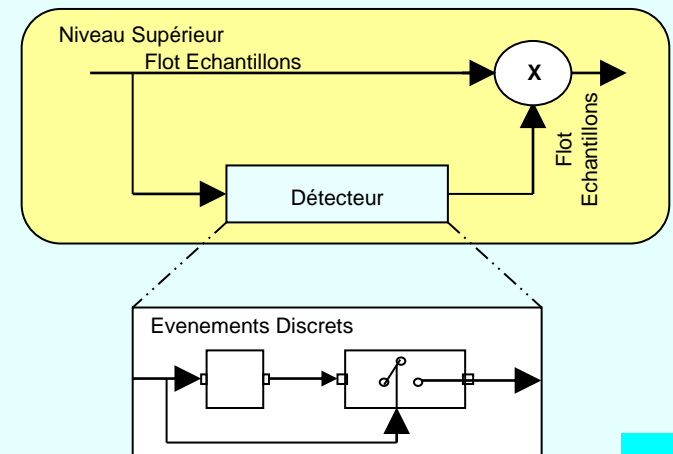
Exemples d'environnements :

- SIMULINK et VHDL-AMS

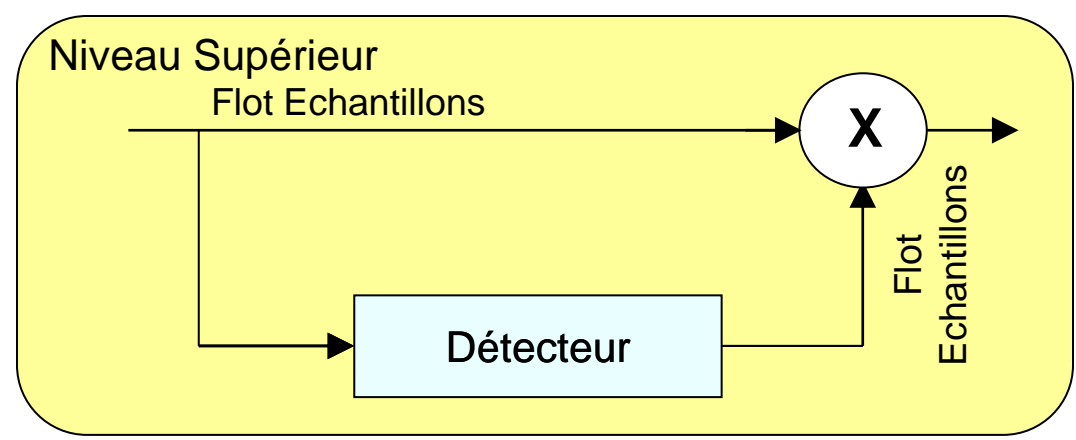
■ Hétérogénéité Hiérarchique (Pour un ensemble ouvert de MoCs)

- On ne peut pas construire l'union des MoCs *a priori*
- Un composant n'obéit qu'à un seul MoC.
- L'abstraction hiérarchique permet d'utiliser un MoC interne à un composant différent du MoC externe
- Les changements de MoC ne peuvent s'opérer qu'à la frontière d'un composant.

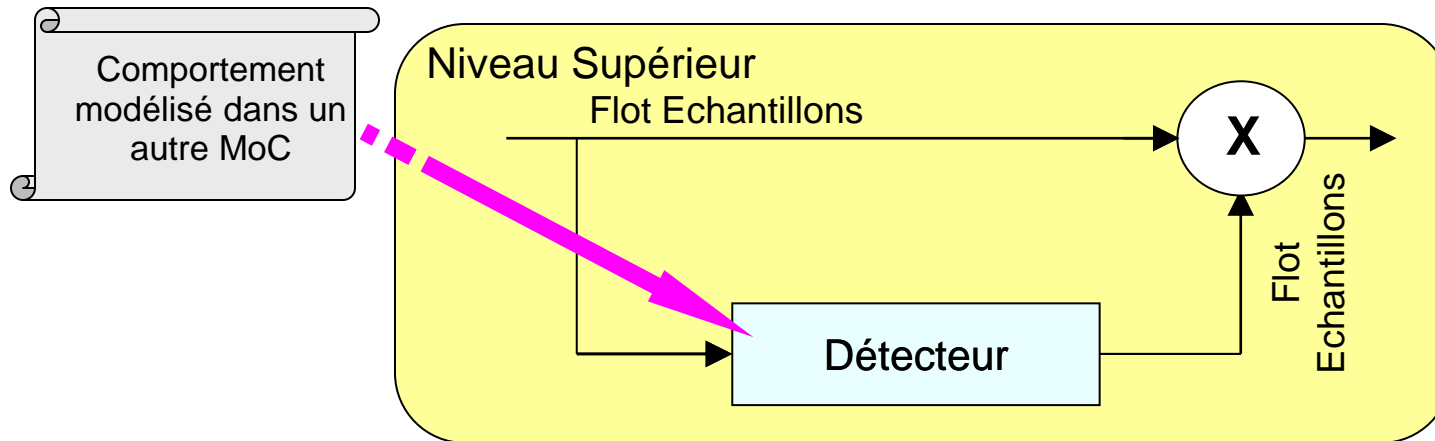
Exemple d'environnement : Ptolemy II



Problématique



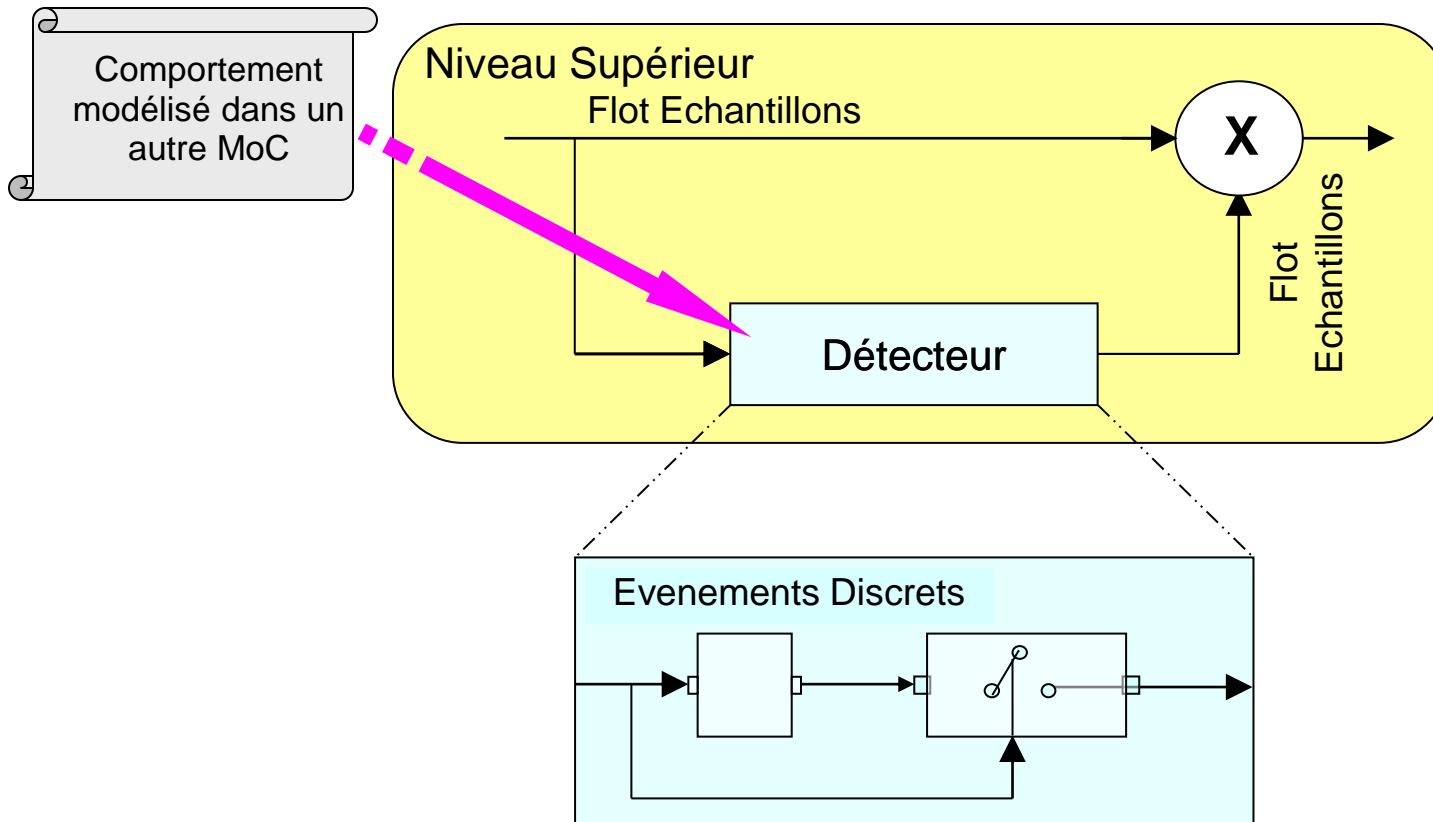
Problématique



Conséquences

- Apparition de constructions ad hoc à la frontière entre deux MoCs
- Interdiction des composants fonctionnant à la frontière de plusieurs MoCs
- Les transformations des données à la frontière entre deux domaines dépendent de l'environnement de modélisation

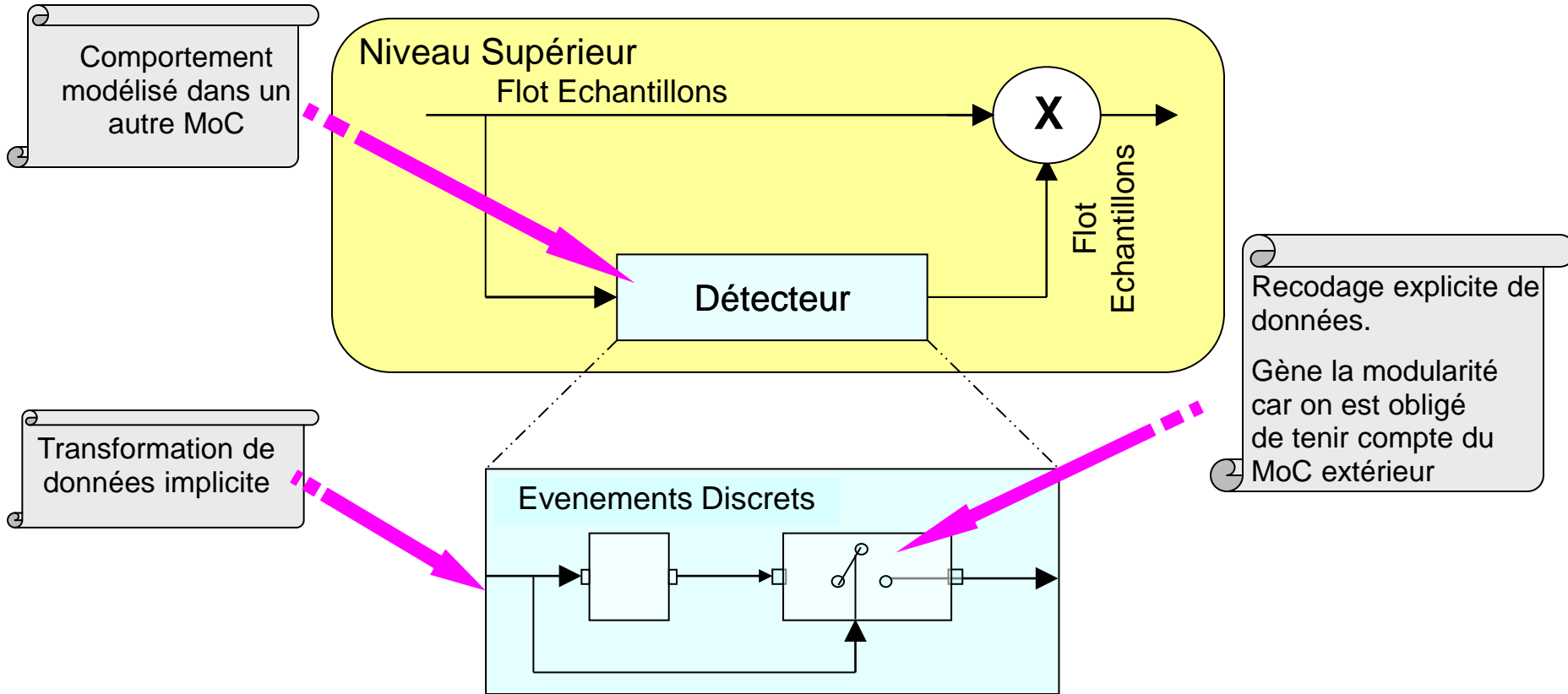
Problématique



Conséquences

- Apparition de constructions ad hoc à la frontière entre deux MoCs
- Interdiction des composants fonctionnant à la frontière de plusieurs MoCs
- Les transformations des données à la frontière entre deux domaines dépendent de l'environnement de modélisation

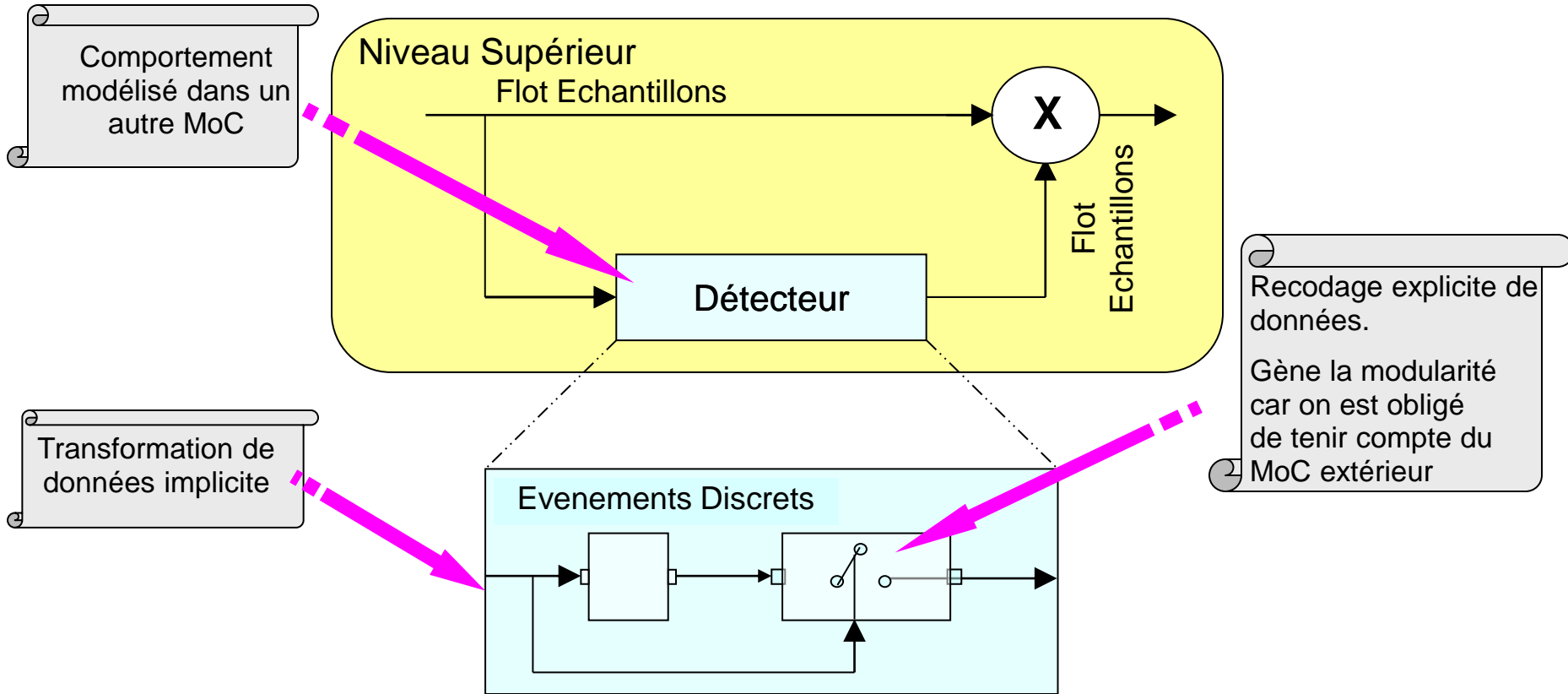
Problématique



Conséquences

- Apparition de constructions ad hoc à la frontière entre deux MoCs
- Interdiction des composants fonctionnant à la frontière de plusieurs MoCs
- Les transformations des données à la frontière entre deux domaines dépendent de l'environnement de modélisation

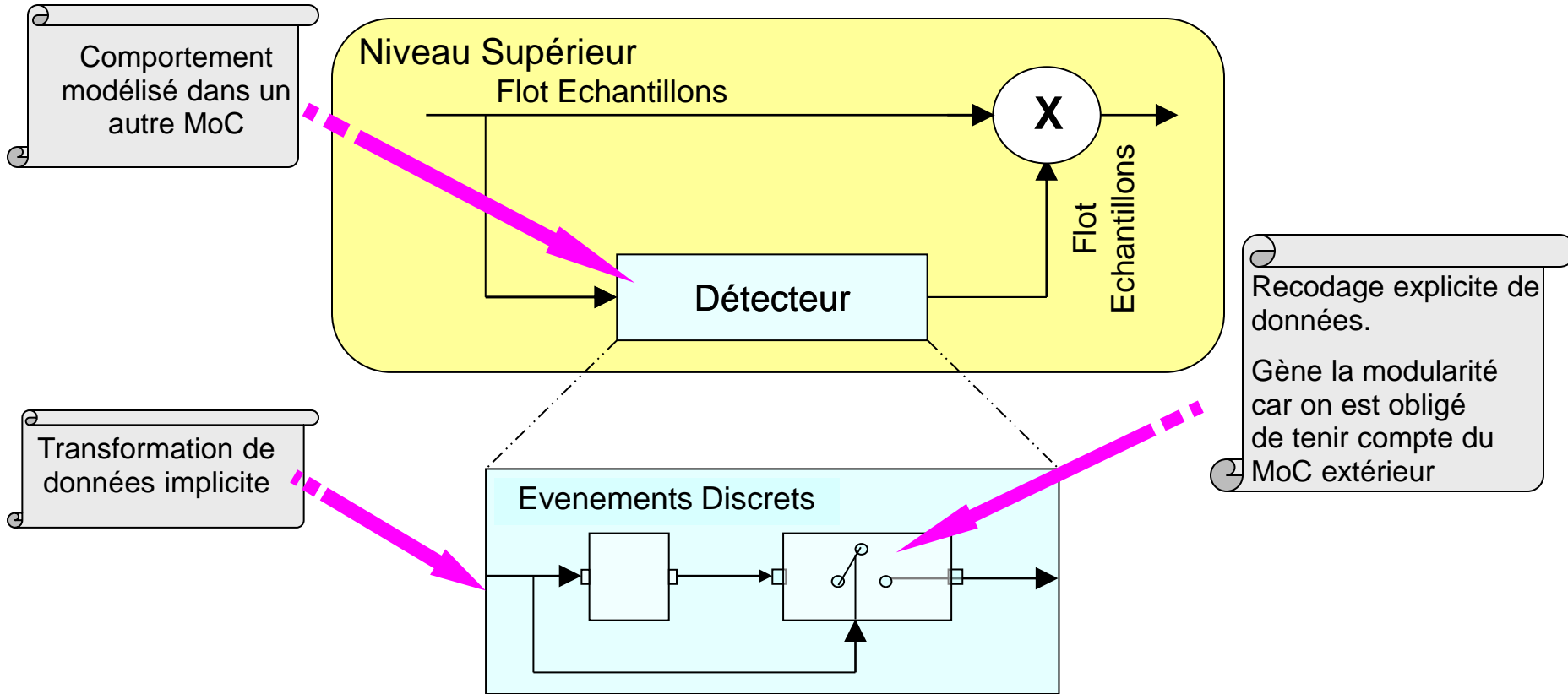
Problématique



Conséquences

- Apparition de constructions ad hoc à la frontière entre deux MoCs
- Interdiction des composants fonctionnant à la frontière de plusieurs MoCs
- Les transformations des données à la frontière entre deux domaines dépendent de l'environnement de modélisation

Problématique



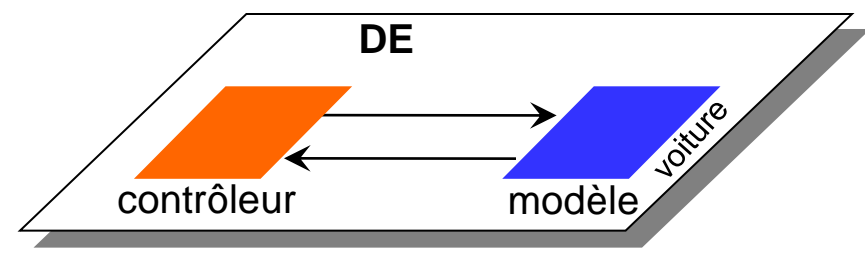
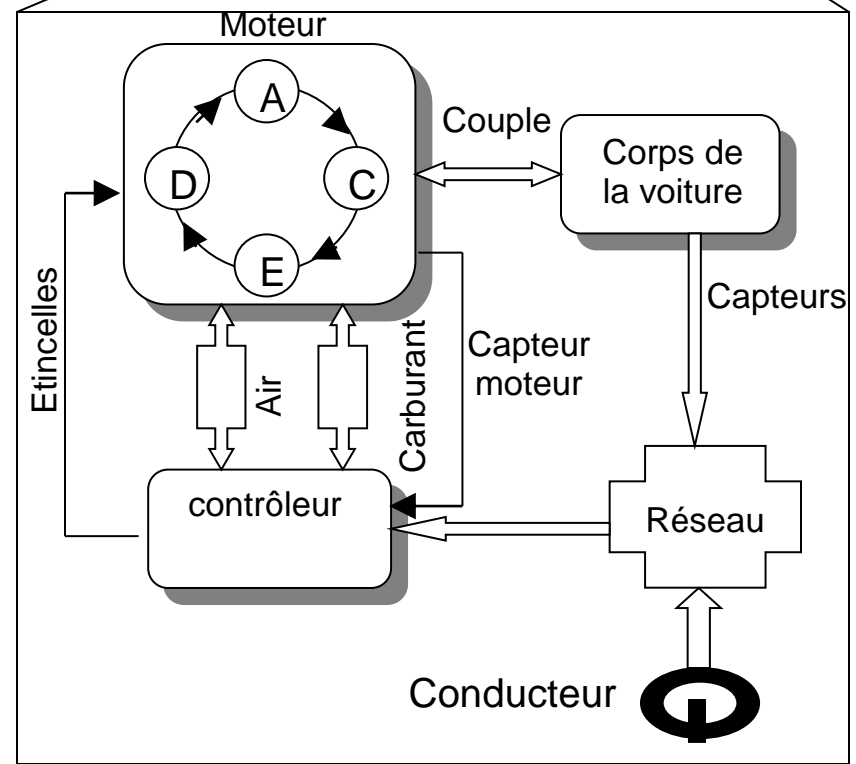
Conséquences

- Apparition de constructions ad hoc à la frontière entre deux MoCs
- Interdiction des composants fonctionnant à la frontière de plusieurs MoCs
- Les transformations des données à la frontière entre deux domaines dépendent de l'environnement de modélisation

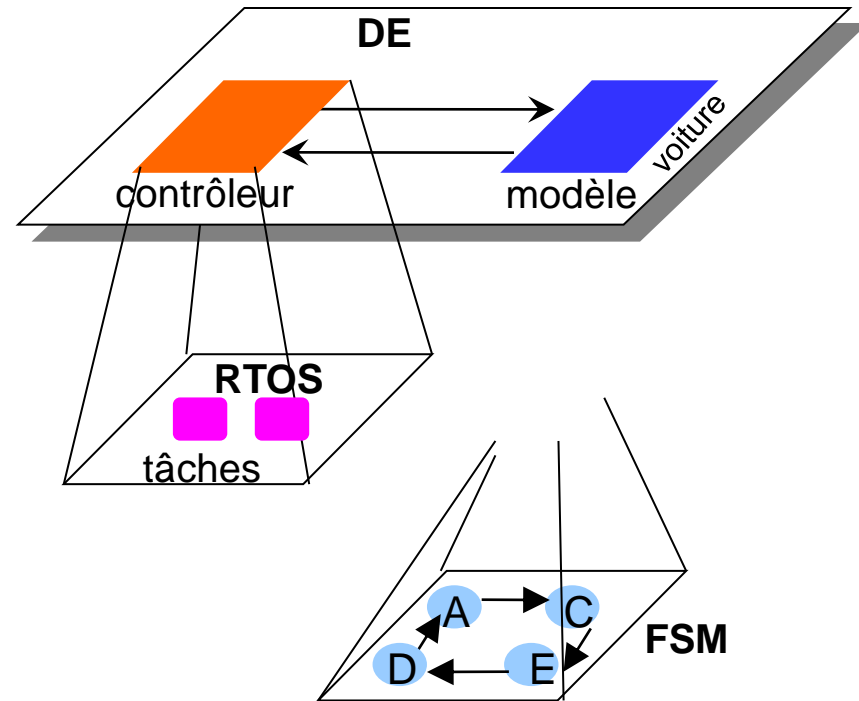
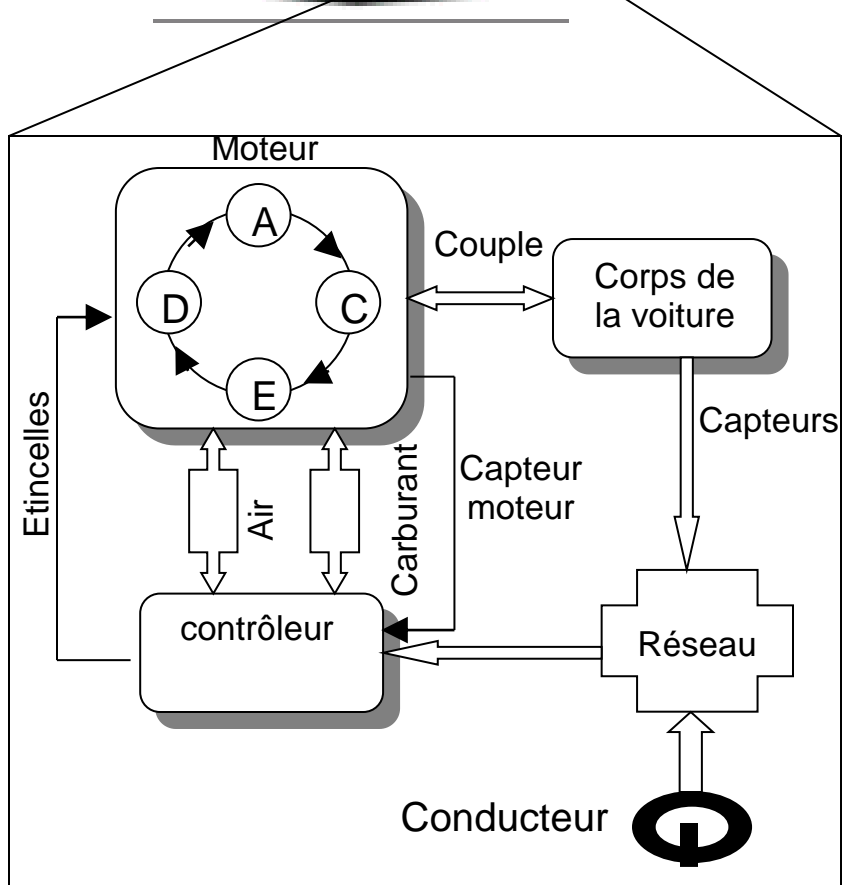
■ Exemple d'Hétérogénéité Hiérarchique



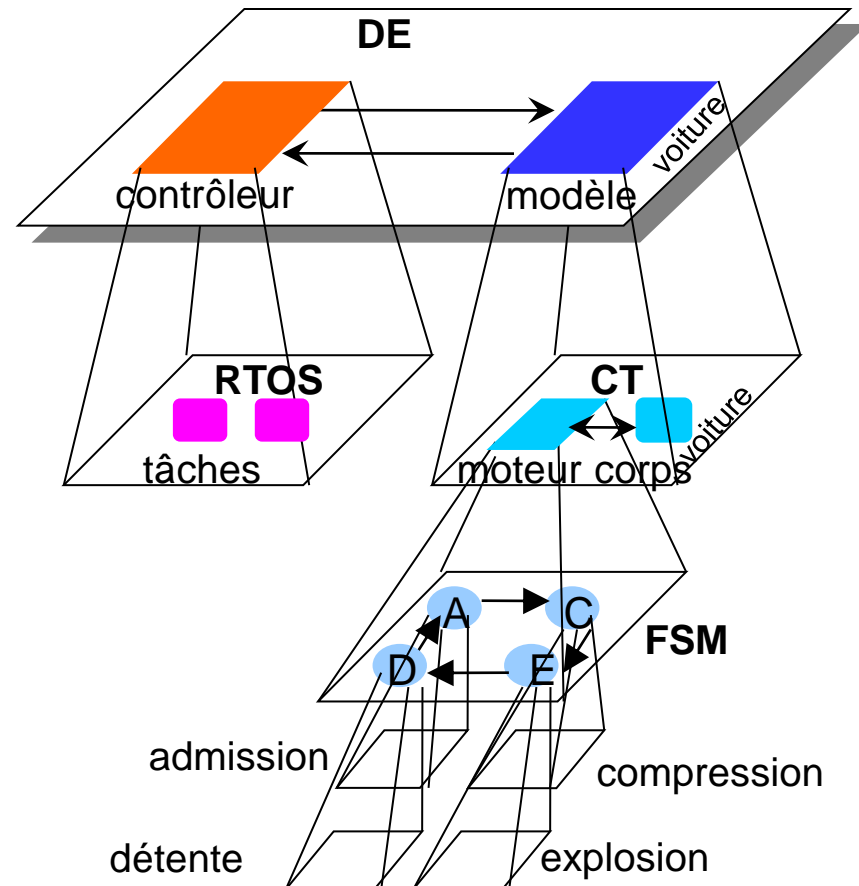
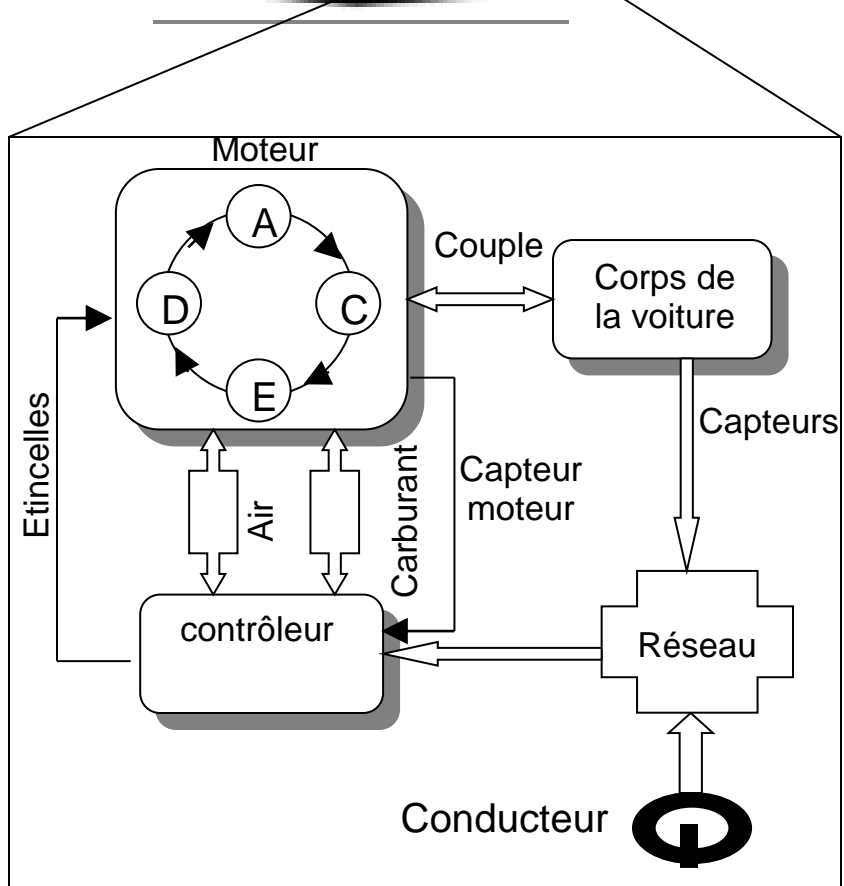
■ Exemple d'Hétérogénéité Hiérarchique



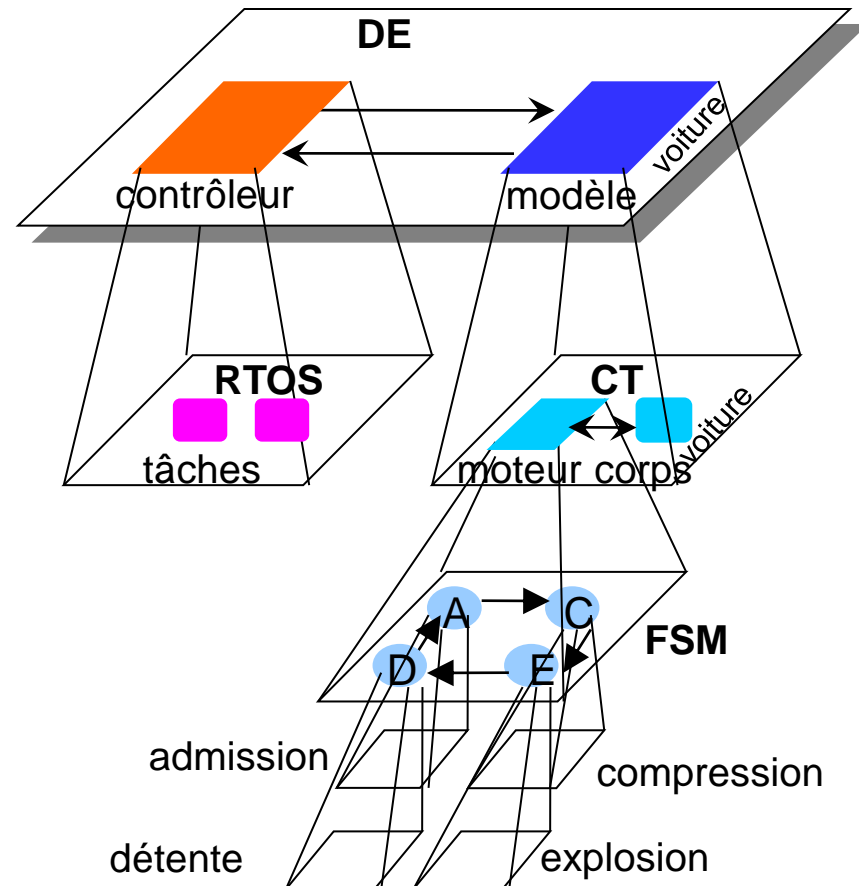
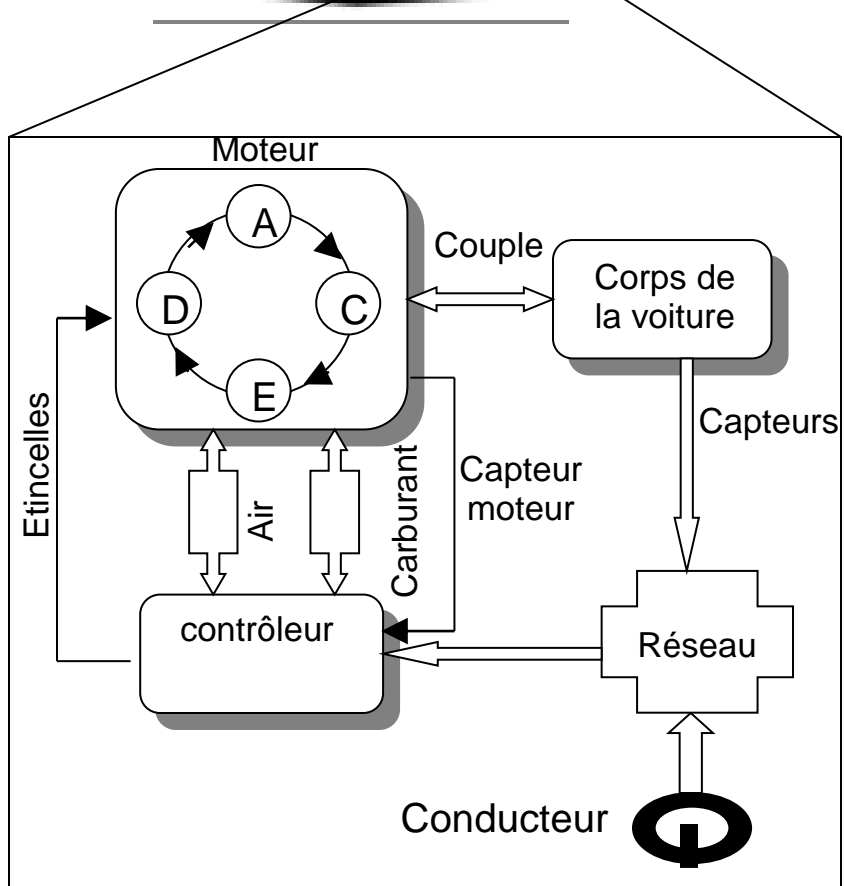
■ Exemple d'Hétérogénéité Hiérarchique



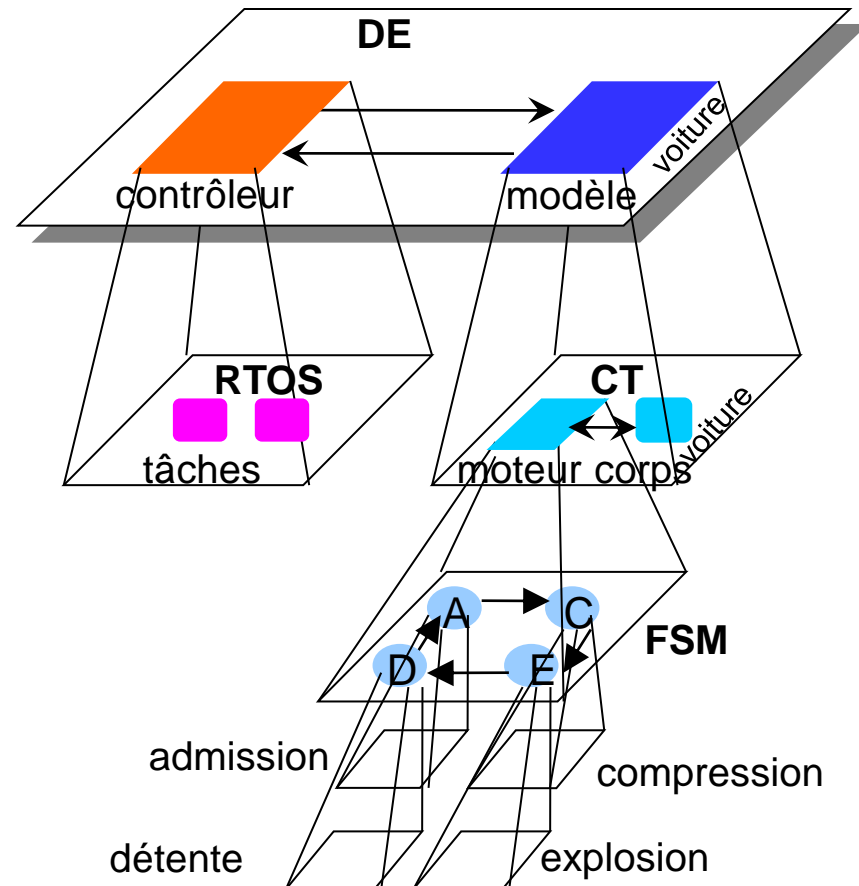
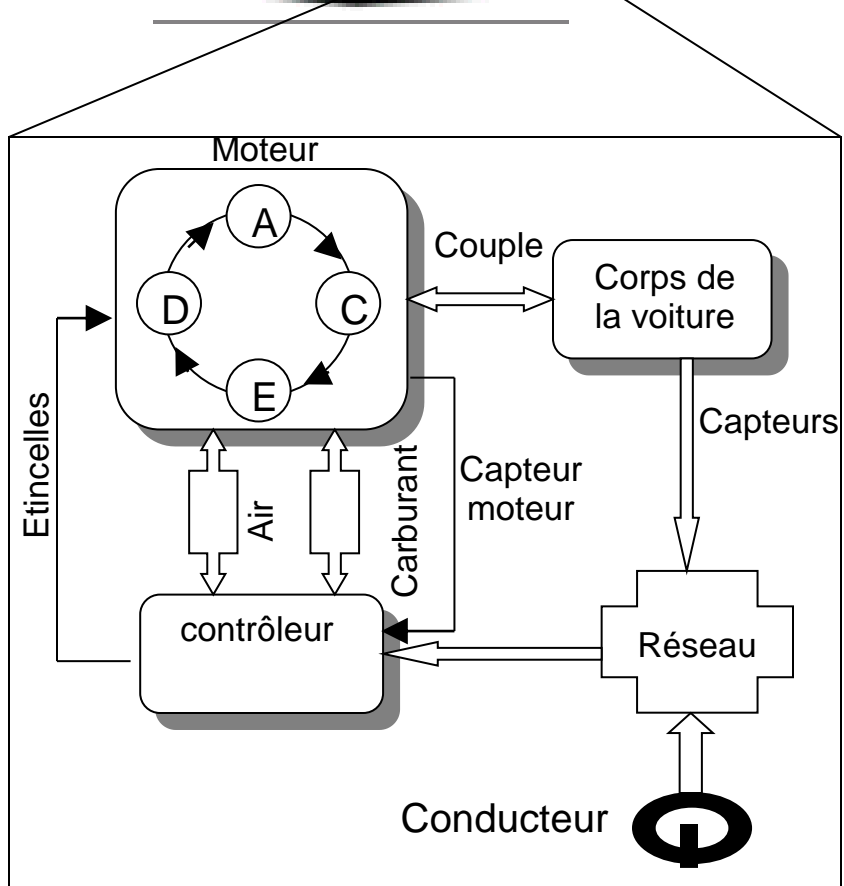
■ Exemple d'Hétérogénéité Hiérarchique



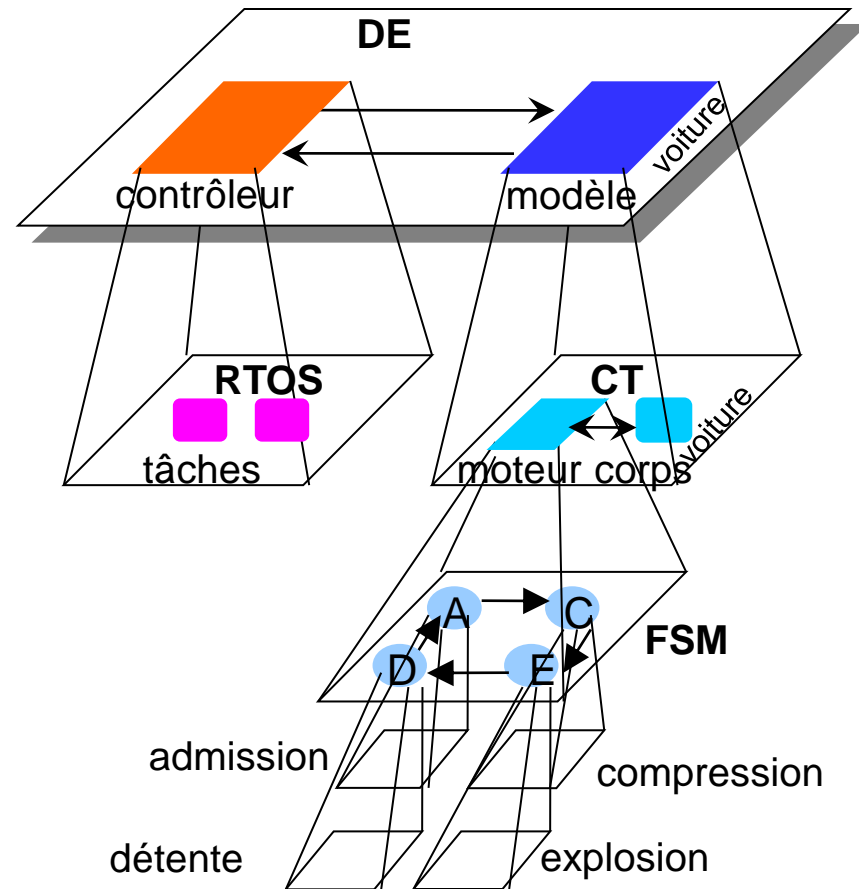
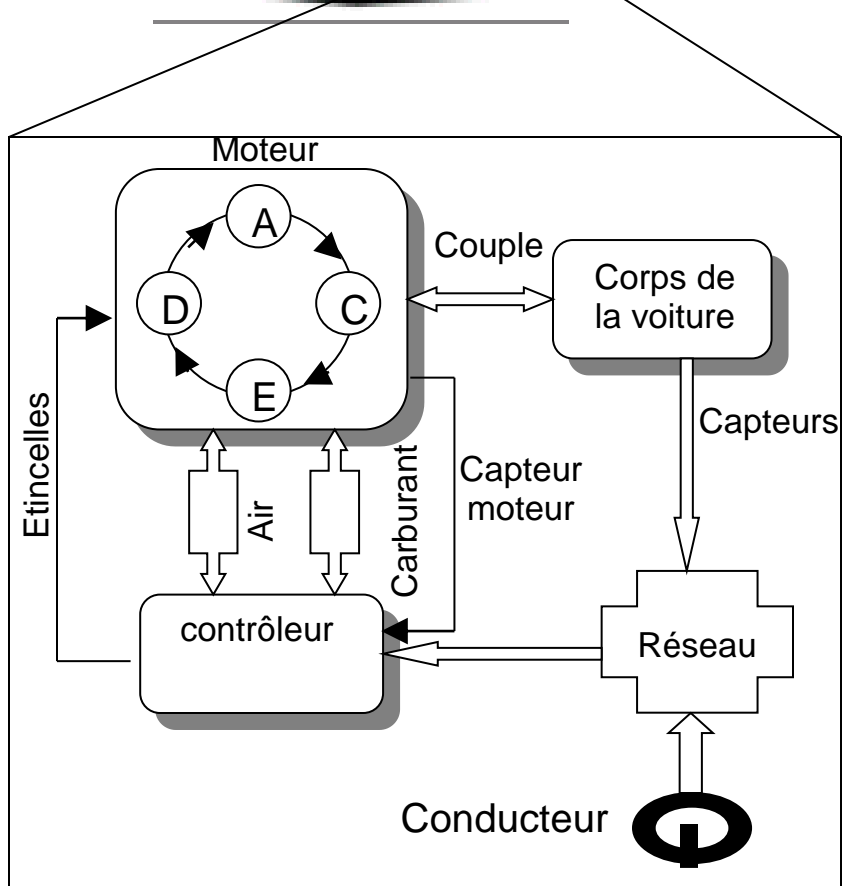
Exemple d'Hétérogénéité Hiérarchique



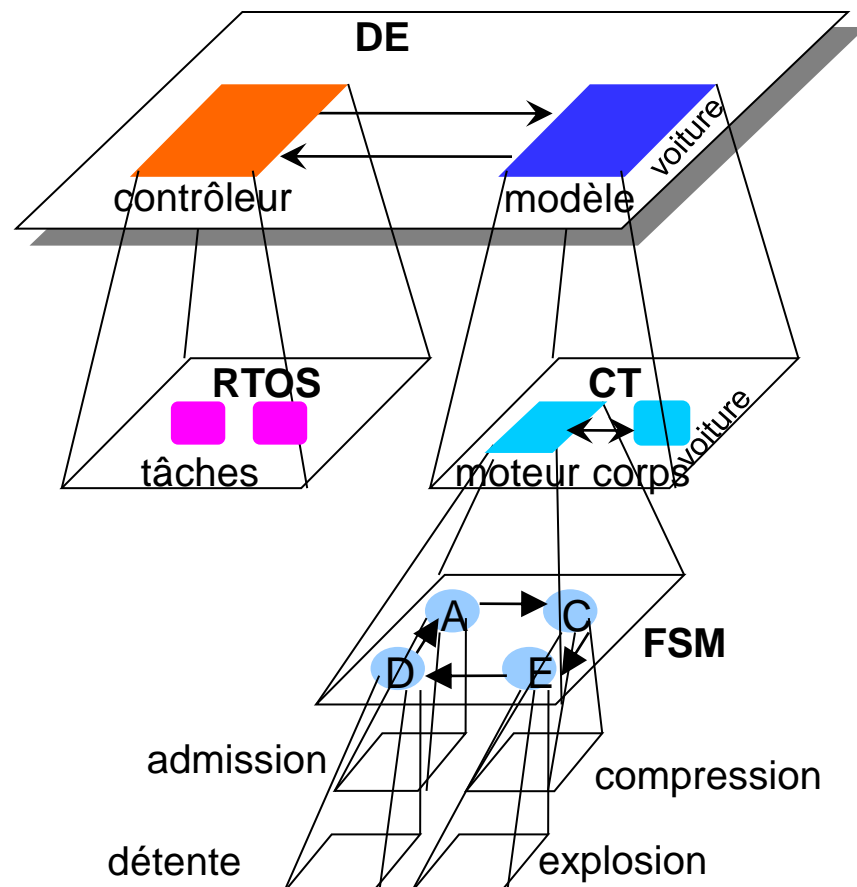
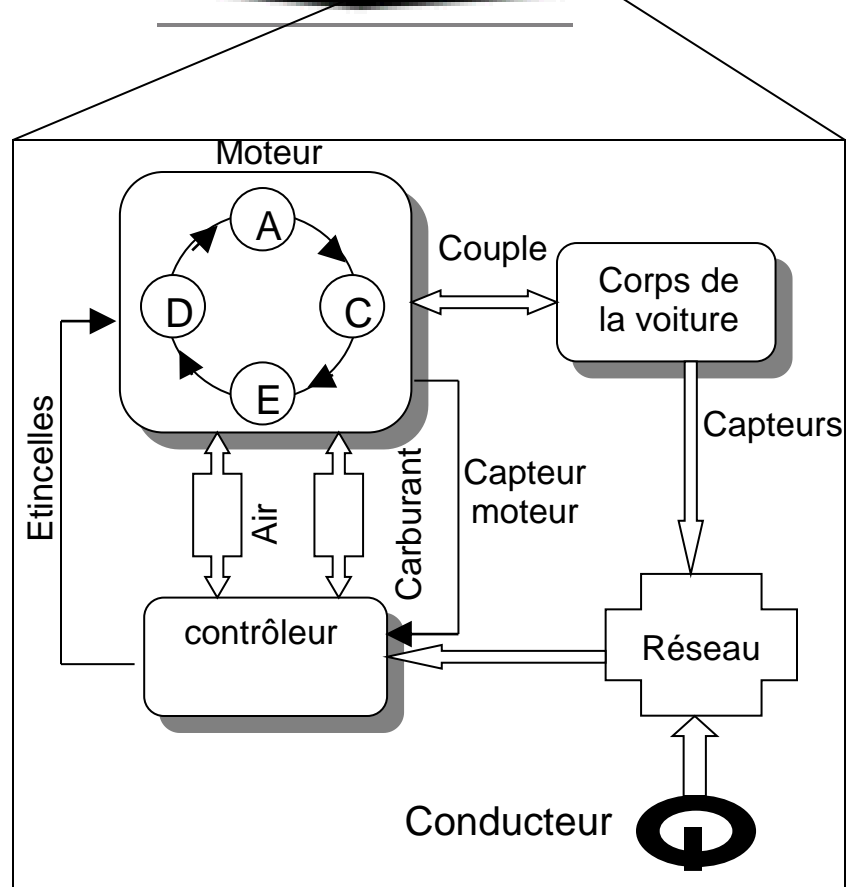
■ Exemple d'Hétérogénéité Hiérarchique



Exemple d'Hétérogénéité Hiérarchique



Exemple d'Hétérogénéité Hiérarchique

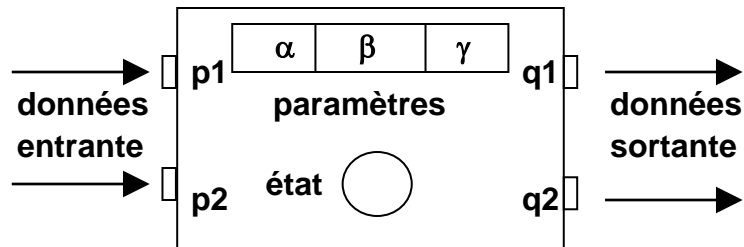


- Dans chaque couche, le MoC est bien défini.
- L'interface entre les couches est prise en compte par les concepteurs de plate-formes.

Plan

- **Systèmes Embarqués**
 - Positionnement du marché
 - Caractéristiques
 - Outils de modélisation et Modélisation hiérarchique
- **Problématique et Objectif**
- **Approche hétérogène non-hiérarchique**
- **Modélisation :**
 - Composant à Interface Hétérogène (HIC)
 - Modèle d'Exécution Hétérogène Non-Hiérarchique
- **Intégration et Validation par Simulation dans Ptolemy II**
- **Conclusion et perspectives**

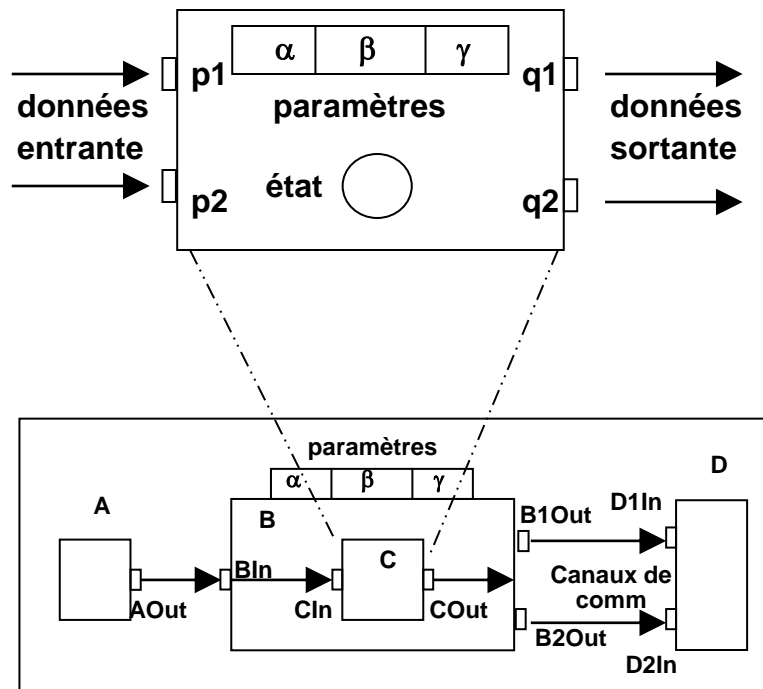
Choix de la Méthodologie orientée Acteurs



■ Acteur = Composant

- Interface = (Ports + Paramètres) et état
- Opérations : flot de données et de contrôle

Choix de la Méthodologie orientée Acteurs



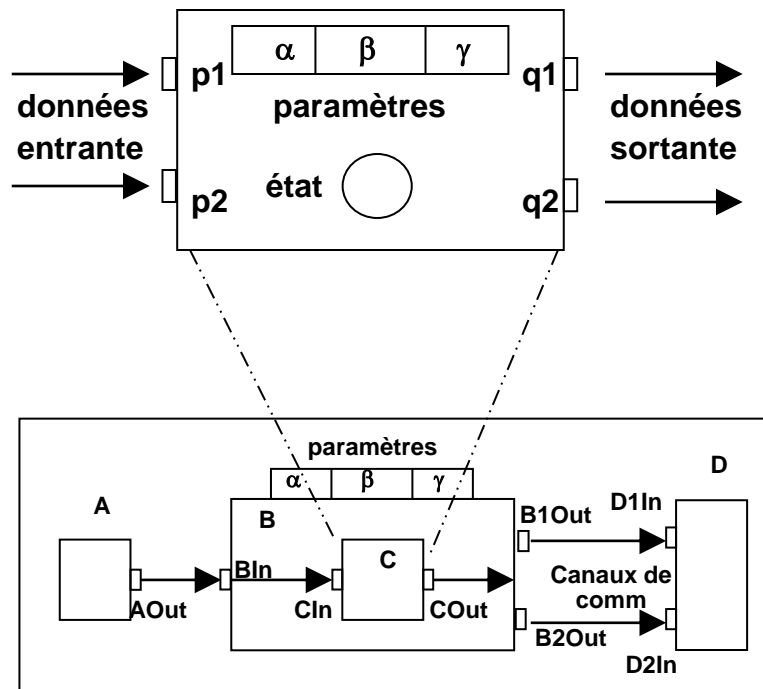
■ Acteur = Composant

- Interface = (Ports + Paramètres) et état
- Opérations : flot de données et de contrôle

■ Modèle

- Implémente le MoC : définit la sémantique de communication entre les acteurs
- Interface externe = Ports + Paramètres externes : Abstraction hiérarchique

Choix de la Méthodologie orientée Acteurs



■ Acteur = Composant

- Interface = (Ports + Paramètres) et état
- Opérations : flot de données et de contrôle

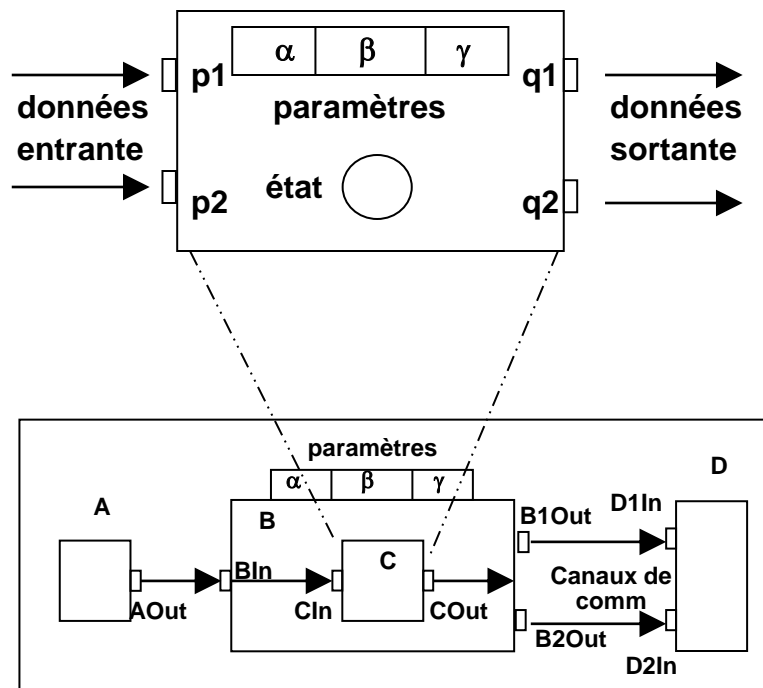
■ Modèle

- Implémente le MoC : définit la sémantique de communication entre les acteurs
- Interface externe = Ports + Paramètres externes : Abstraction hiérarchique

■ Structuration et conceptualisation des systèmes

- Système = structure de sous-systèmes (met en évidence la structure causale du système)

Choix de la Méthodologie orientée Acteurs



■ Acteur = Composant

- Interface = (Ports + Paramètres) et état
- Opérations : flot de données et de contrôle

■ Modèle

- Implémente le MoC : définit la sémantique de communication entre les acteurs
- Interface externe = Ports + Paramètres externes : Abstraction hiérarchique

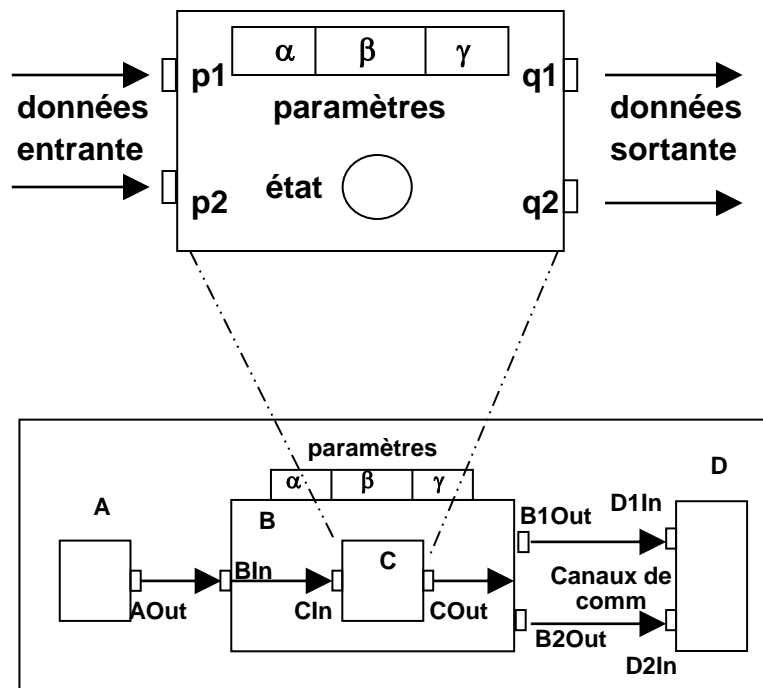
■ Structuration et conceptualisation des systèmes

- Système = structure de sous-systèmes (met en évidence la structure causale du système)

■ Décomposabilité et composabilité modulaire

- Décomposition des systèmes en composants interagissants et recomposition avec MoCs

Choix de la Méthodologie orientée Acteurs



■ Acteur = Composant

- Interface = (Ports + Paramètres) et état
- Opérations : flot de données et de contrôle

■ Modèle

- Implémente le MoC : définit la sémantique de communication entre les acteurs
- Interface externe = Ports + Paramètres externes : Abstraction hiérarchique

■ Structuration et conceptualisation des systèmes

- Système = structure de sous-systèmes (met en évidence la structure causale du système)

■ Décomposabilité et composabilité modulaire

- Décomposition des systèmes en composants interagissants et recomposition avec MoCs

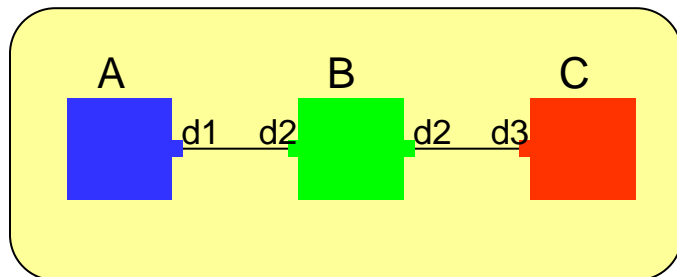
■ Séparation des préoccupations

- Complète l'OO + découplage (transmission de données et le transfert de contrôle :
séparation flot des données - flot de contrôle)

But de notre Approche :
Pouvoir utiliser différents MoCs à un même niveau hiérarchique

But de notre Approche :
Pouvoir utiliser différents MoCs à un même niveau hiérarchique

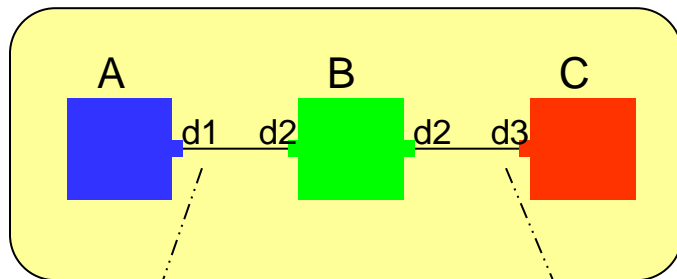
- ◆ Un composant n'obéit qu'à un seul MoC, mais, on peut établir des connexions entre les ports des domaines différents
- ◆ Les connexions ne sont autorisées que entre les ports du même domaine, mais, un composant peut obéir à plusieurs MoCs



But de notre Approche :
Pouvoir utiliser différents MoCs à un même niveau hiérarchique

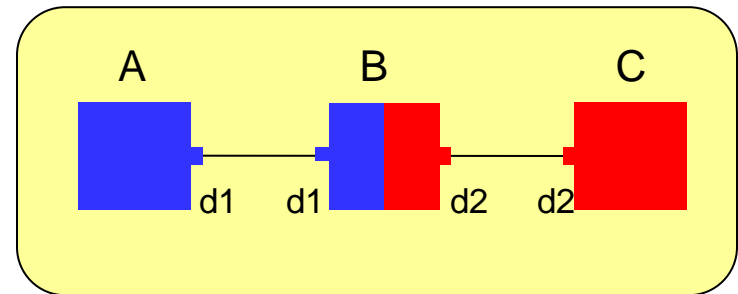
◆ Un composant n'obéit qu'à un seul MoC, mais, on peut établir des connexions entre les ports des domaines différents

◆ Les connexions ne sont autorisées que entre les ports du même domaine, mais, un composant peut obéir à plusieurs MoCs



Changement de sémantique

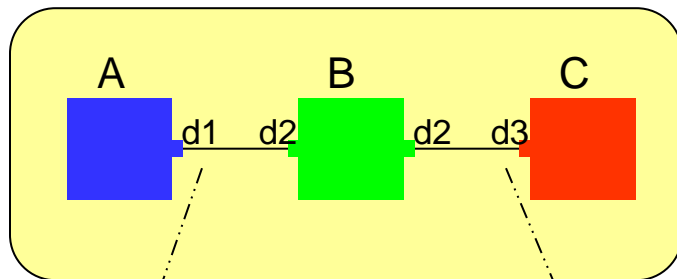
- Conversion de protocole
- transformation format de donnés



But de notre Approche :
Pouvoir utiliser différents MoCs à un même niveau hiérarchique

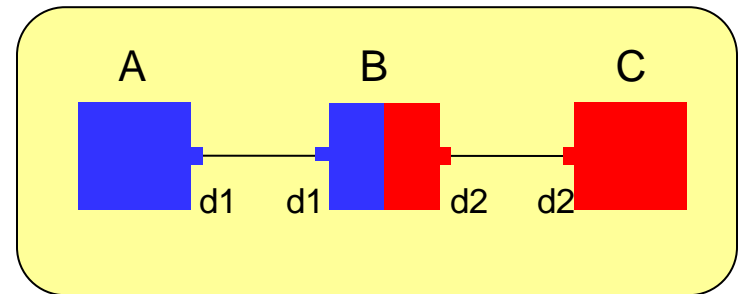
◆ Un composant n'obéit qu'à un seul MoC, mais, on peut établir des connexions entre les ports des domaines différents

◆ Les connexions ne sont autorisées que entre les ports du même domaine, mais, un composant peut obéir à plusieurs MoCs



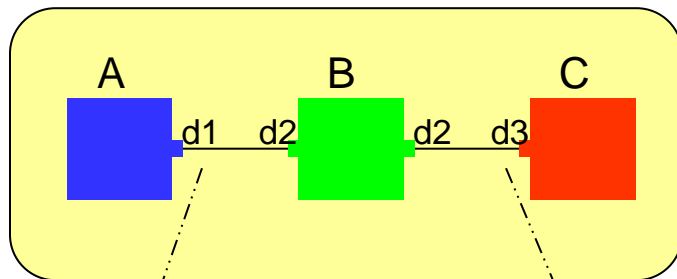
Changement de sémantique

- Conversion de protocole
- transformation format de donnés



But de notre Approche :
Pouvoir utiliser différents MoCs à un même niveau hiérarchique

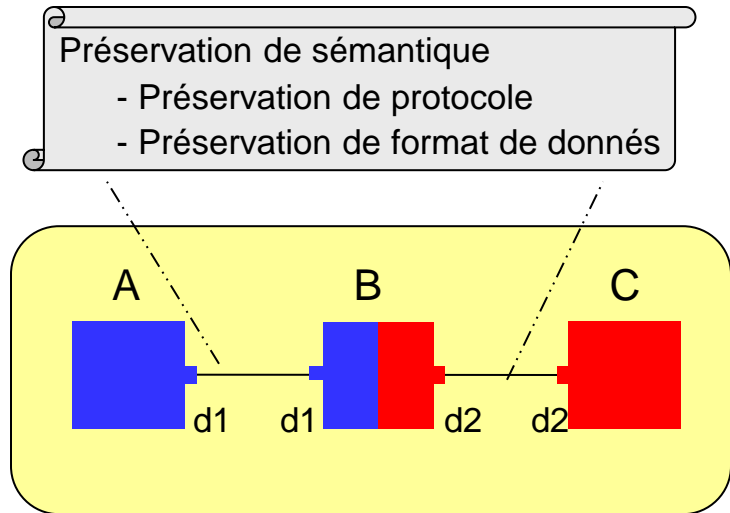
- ◆ Un composant n'obéit qu'à un seul MoC, mais, on peut établir des connexions entre les ports des domaines différents



Changement de sémantique

- Conversion de protocole
- transformation format de donnés

- ◆ Les connexions ne sont autorisées que entre les ports du même domaine, mais, un composant peut obéir à plusieurs MoCs

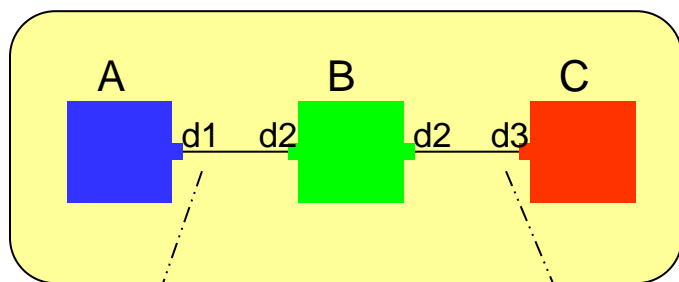


Préservation de sémantique

- Préservation de protocole
- Préservation de format de donnés

But de notre Approche :
Pouvoir utiliser différents MoCs à un même niveau hiérarchique

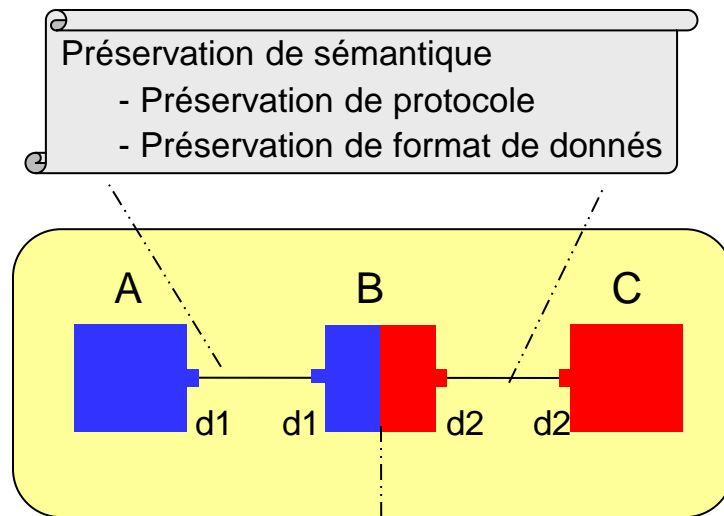
◆ Un composant n'obéit qu'à un seul MoC, mais, on peut établir des connexions entre les ports des domaines différents



Changement de sémantique

- Conversion de protocole
- transformation format de donnés

◆ Les connexions ne sont autorisées que entre les ports du même domaine, mais, un composant peut obéir à plusieurs MoCs



Préservation de sémantique

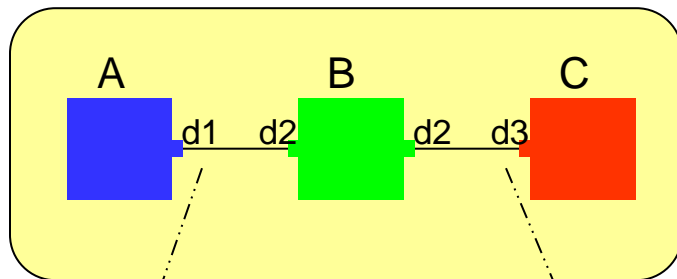
- Préservation de protocole
- Préservation de format de donnés

Changement de sémantique

- Conversion de protocole
- transformation format de donnés

But de notre Approche :
Pouvoir utiliser différents MoCs à un même niveau hiérarchique

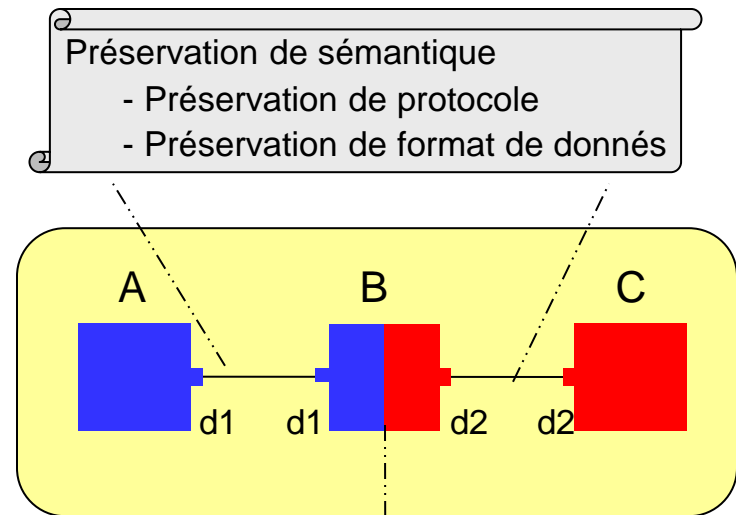
- ◆ Un composant n'obéit qu'à un seul MoC, mais, on peut établir des connexions entre les ports des domaines différents



Changement de sémantique

- Conversion de protocole
- transformation format de donnés

- ◆ Les connexions ne sont autorisées que entre les ports du même domaine, mais, un composant peut obéir à plusieurs MoCs



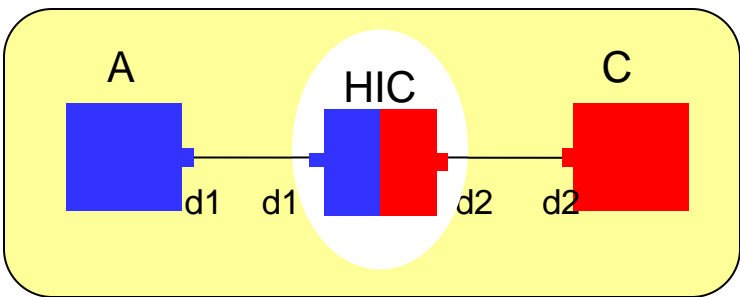
Préservation de sémantique

- Préservation de protocole
- Préservation de format de donnés

Changement de sémantique

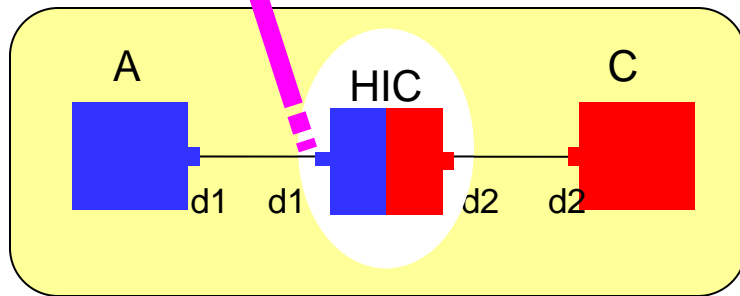
- Conversion de protocole
- transformation format de donnés

■ Composant à Interface Hétérogène (HIC)



■ Composant à Interface Hétérogène (HIC)

Lorsqu'il interprète une entrée, il traduit sa signification dans le MoC associé en sa signification interne.

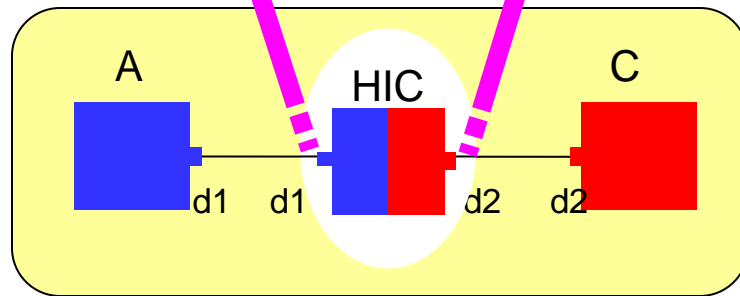


Son comportement peut être décomposé en autant de comportements secondaires qu'il y a de MoCs.
Ces CS sont couplés et chacun d'eux est un « pont » entre un MoC et son comportement global

■ Composant à Interface Hétérogène (HIC)

Lorsqu'il interprète une entrée, il traduit sa signification dans le MoC associé en sa signification interne.

Lorsqu'il produit une sortie, il traduit l'information recueillie de ses entrées dans la sémantique de cette sortie selon son MoC.



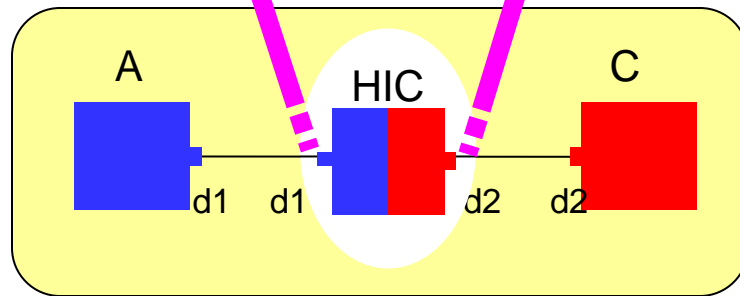
Son comportement peut être décomposé en autant de comportements secondaires qu'il y a de MoCs.

Ces CS sont couplés et chacun d'eux est un « pont » entre un MoC et son comportement global

■ Composant à Interface Hétérogène (HIC)

Lorsqu'il interprète une entrée, il traduit sa signification dans le MoC associé en sa signification interne.

Lorsqu'il produit une sortie, il traduit l'information recueillie de ses entrées dans la sémantique de cette sortie selon son MoC.



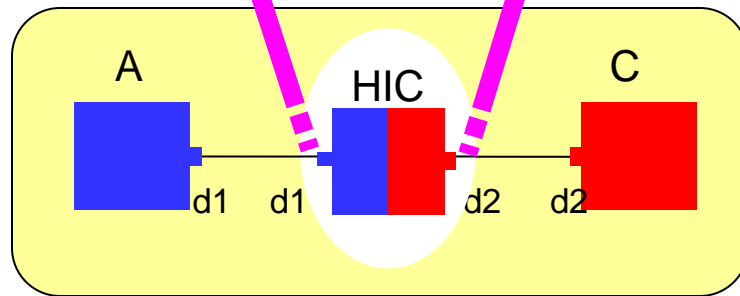
Son comportement peut être décomposé en autant de comportements secondaires qu'il y a de MoCs.

Ces CS sont couplés et chacun d'eux est un « pont » entre un MoC et son comportement global

■ Composant à Interface Hétérogène (HIC)

Lorsqu'il interprète une entrée, il traduit sa signification dans le MoC associé en sa signification interne.

Lorsqu'il produit une sortie, il traduit l'information recueillie de ses entrées dans la sémantique de cette sortie selon son MoC.



Son comportement peut être décomposé en autant de comportements secondaires qu'il y a de MoCs.

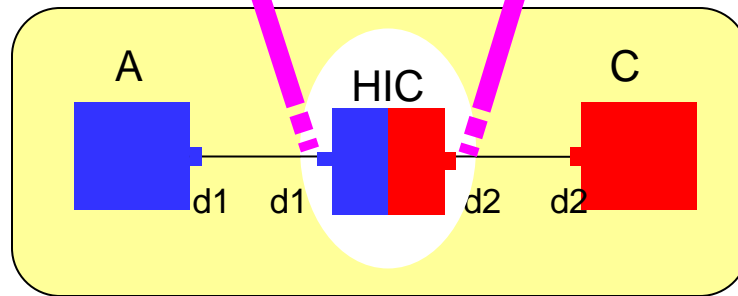
Ces CS sont couplés et chacun d'eux est un « pont » entre un MoC et son comportement global

■ Interprétation du modèle plat dans l'approche hiérarchique

■ Composant à Interface Hétérogène (HIC)

Lorsqu'il interprète une entrée, il traduit sa signification dans le MoC associé en sa signification interne.

Lorsqu'il produit une sortie, il traduit l'information recueillie de ses entrées dans la sémantique de cette sortie selon son MoC.



Son comportement peut être décomposé en autant de comportements secondaires qu'il y a de MoCs.

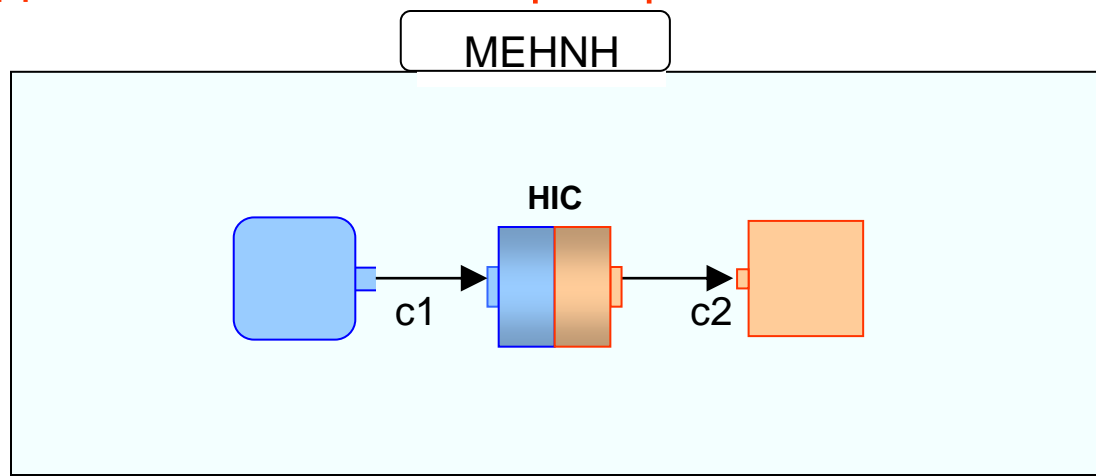
Ces CS sont couplés et chacun d'eux est un « pont » entre un MoC et son comportement global

■ Interprétation du modèle plat dans l'approche hiérarchique

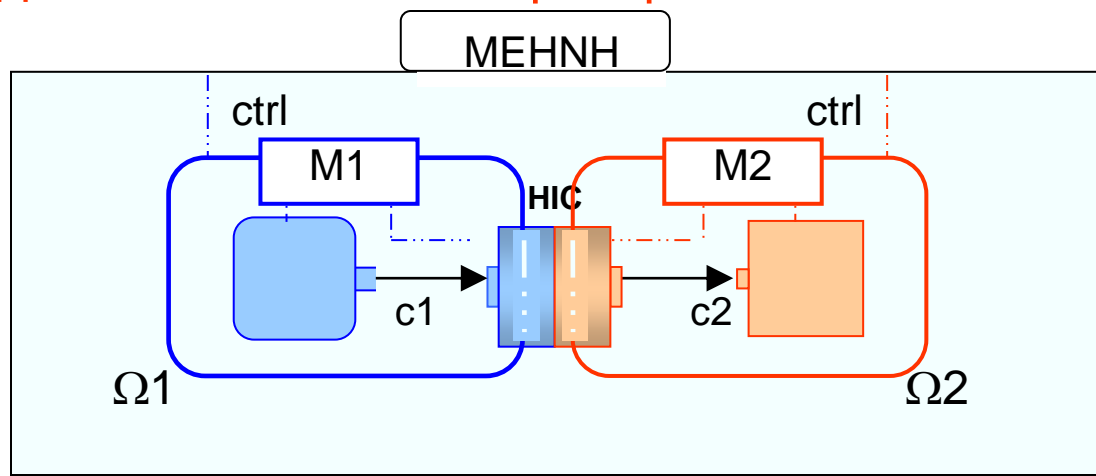
➡ Modèle d'Exécution Hétérogène Non-Hiérarchique

- Gestion du HIC
- Partitionnement du système en sous-systèmes homogènes
- Délégation du calcul du comportement des sous-systèmes à leur modèle
- Ordonnancement statique des sous-systèmes
- Coordination des communications entre sous-systèmes

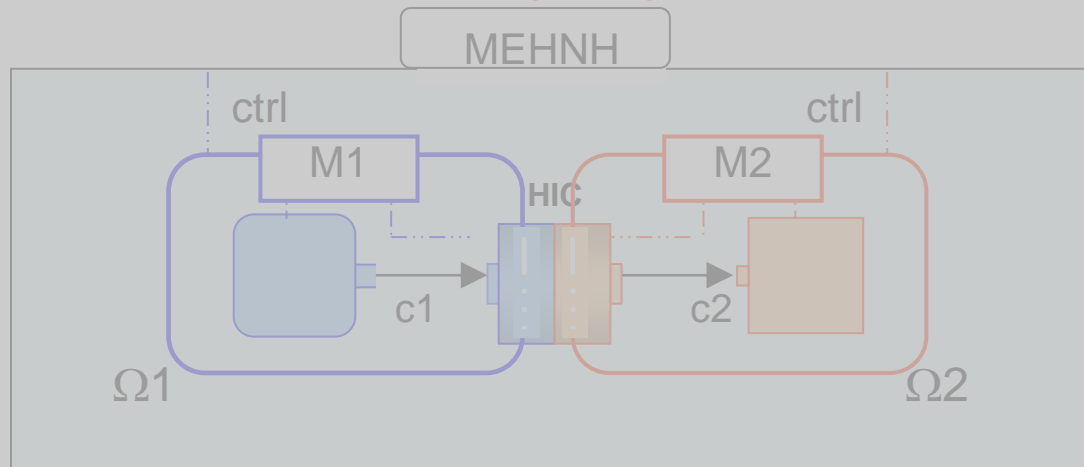
■ Approche Non-Hiérarchique : première tentative



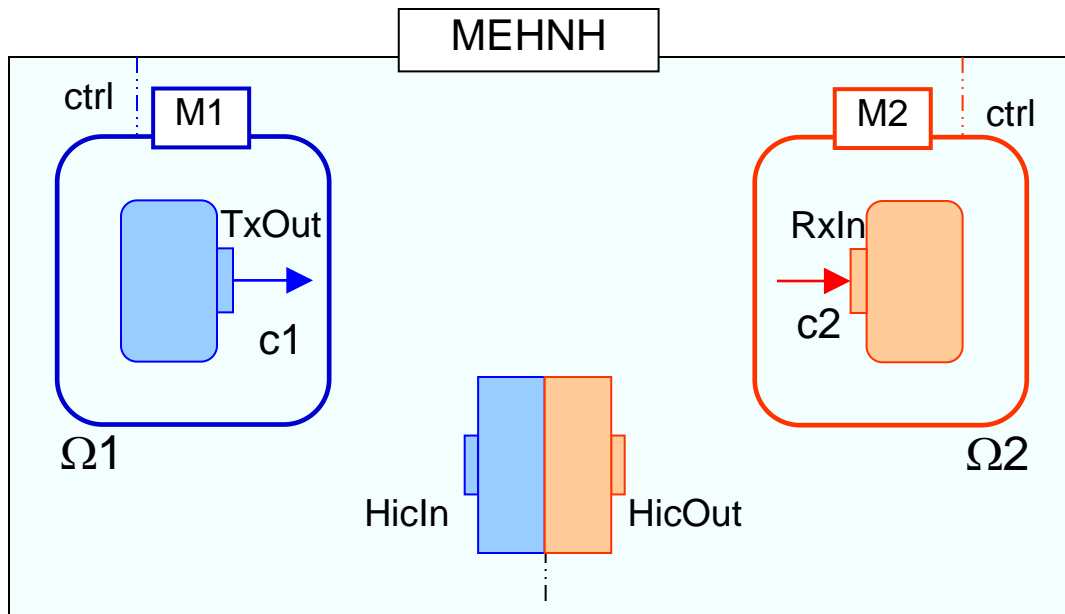
■ Approche Non-Hiérarchique : première tentative



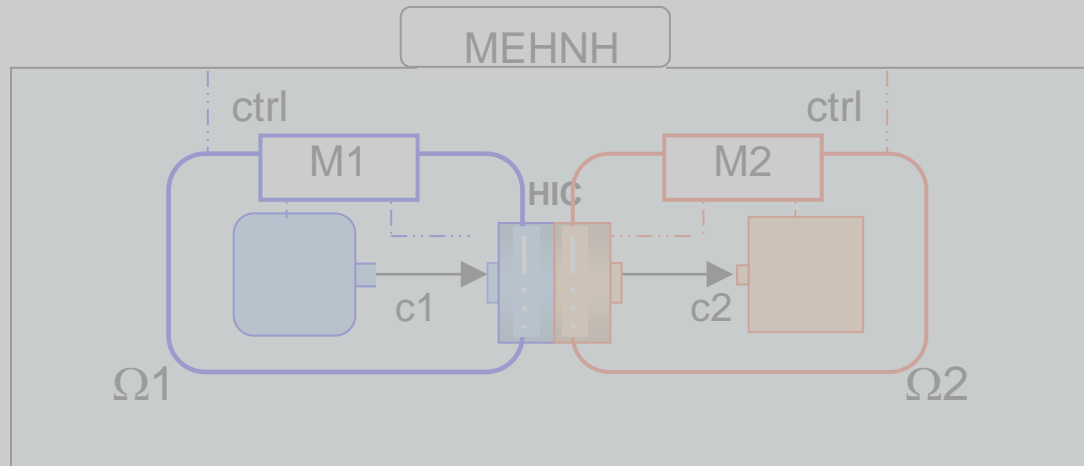
■ Approche Non-Hiérarchique : première tentative



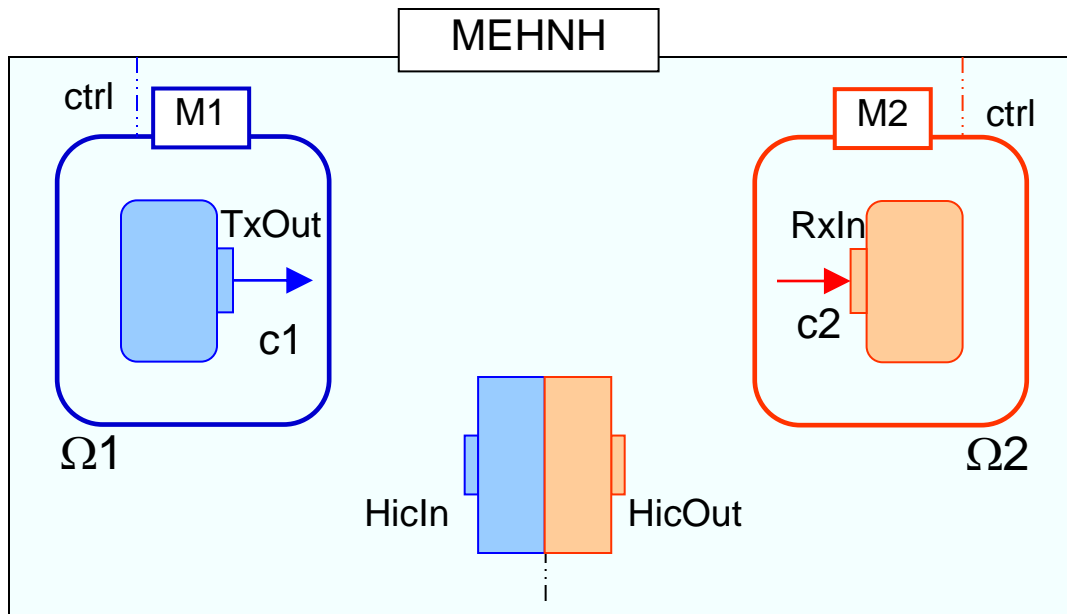
■ Non-Hiérarchie par l'Abstraction Hiérarchique



■ Approche Non-Hiérarchique : première tentative



■ Non-Hiérarchie par l'Abstraction Hiérarchique

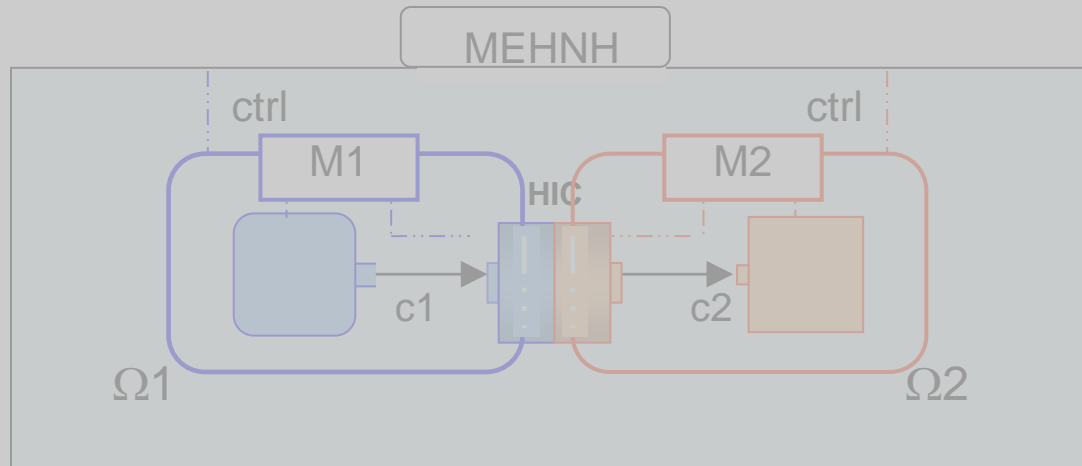


Inconvénient

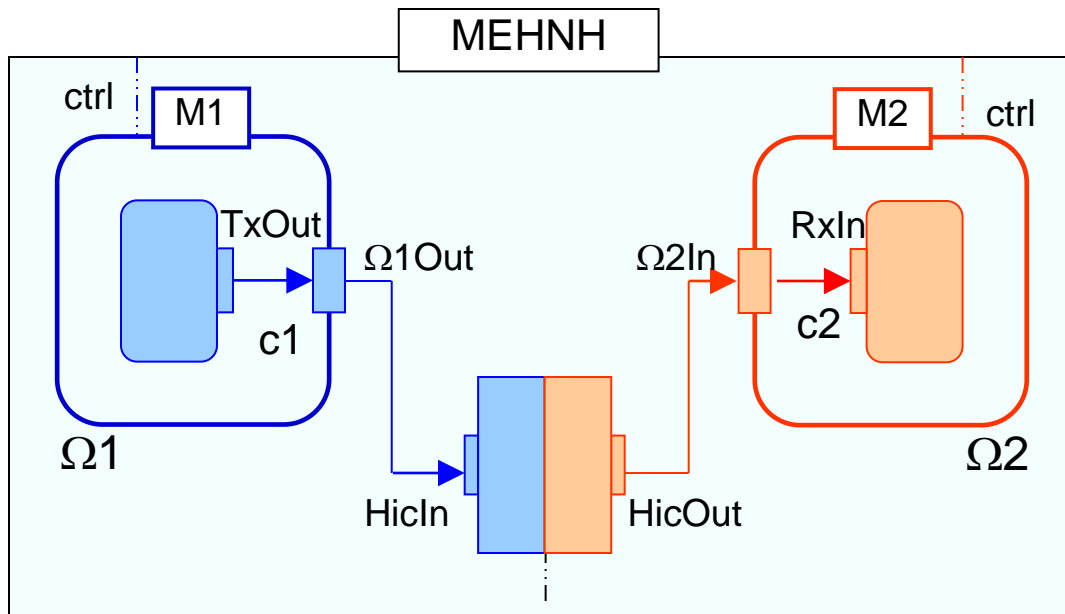
Modèle d'exécution dépend des Mocs utilisés par $\Omega 1$ et $\Omega 2$

Incompatibilité avec un ensemble ouvert de modèles

■ Approche Non-Hiérarchique : première tentative



■ Non-Hiérarchie par l'Abstraction Hiérarchique

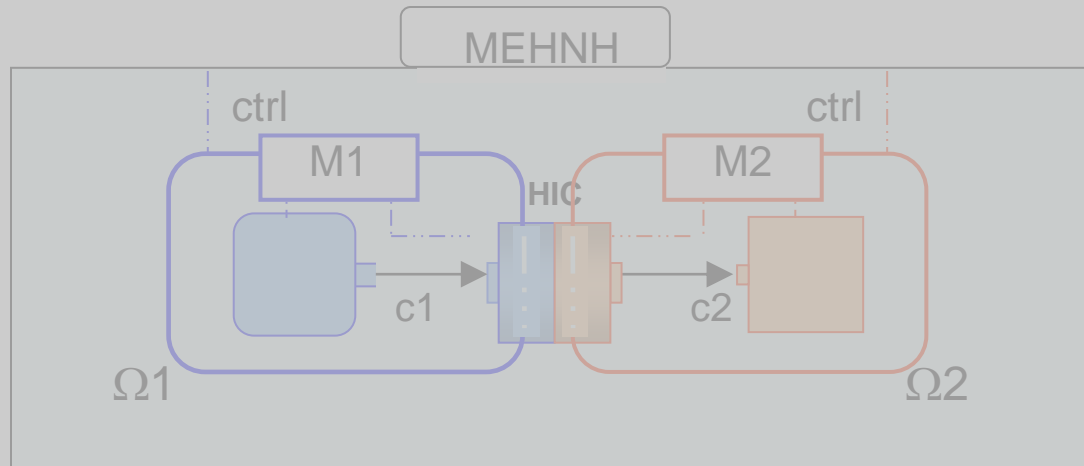


Inconvénient

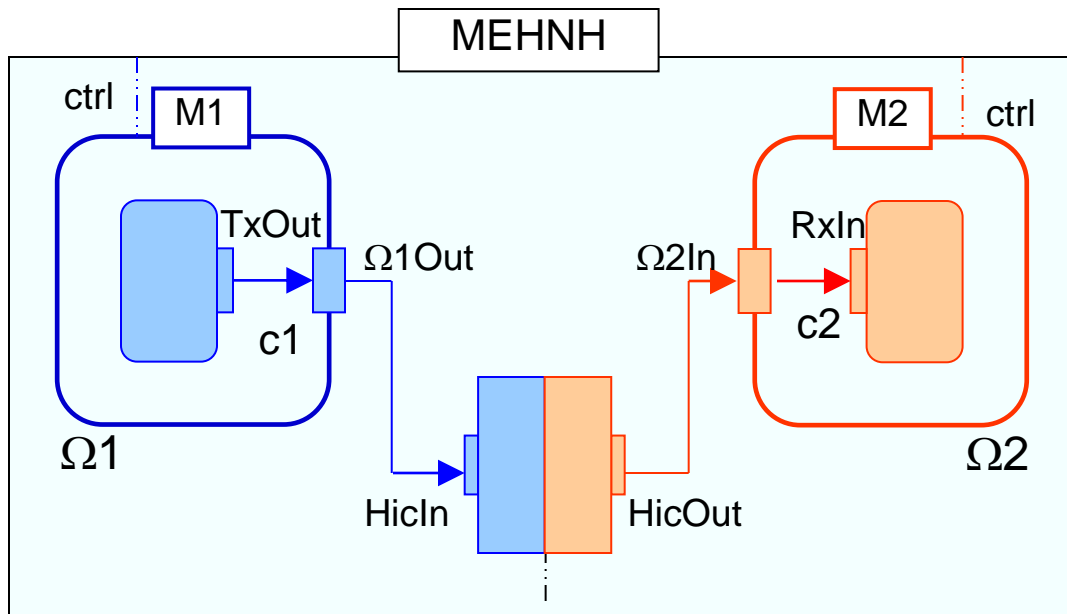
Modèle d'exécution dépend des Mocs utilisés par Ω_1 et Ω_2

Incompatibilité avec un ensemble ouvert de modèles

■ Approche Non-Hiérarchique : première tentative



■ Non-Hiérarchie par l'Abstraction Hiérarchique

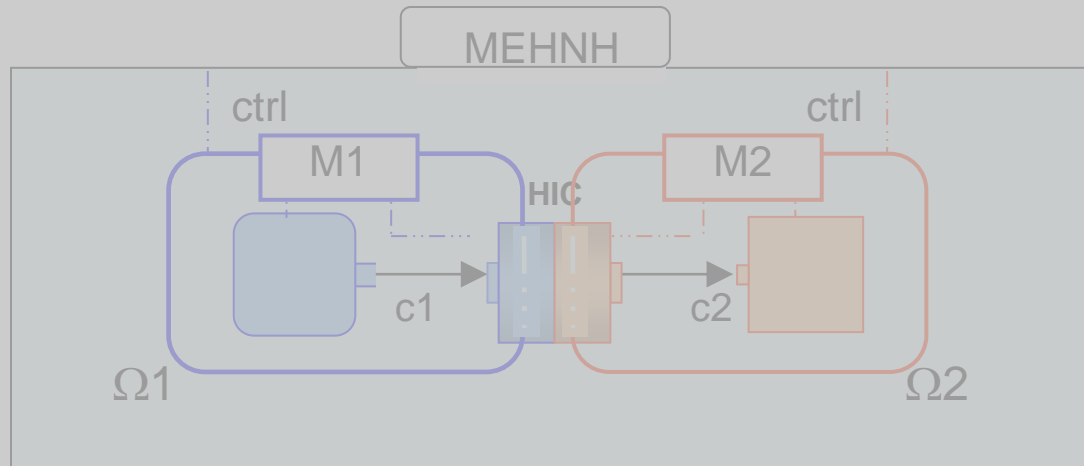


Inconvénient

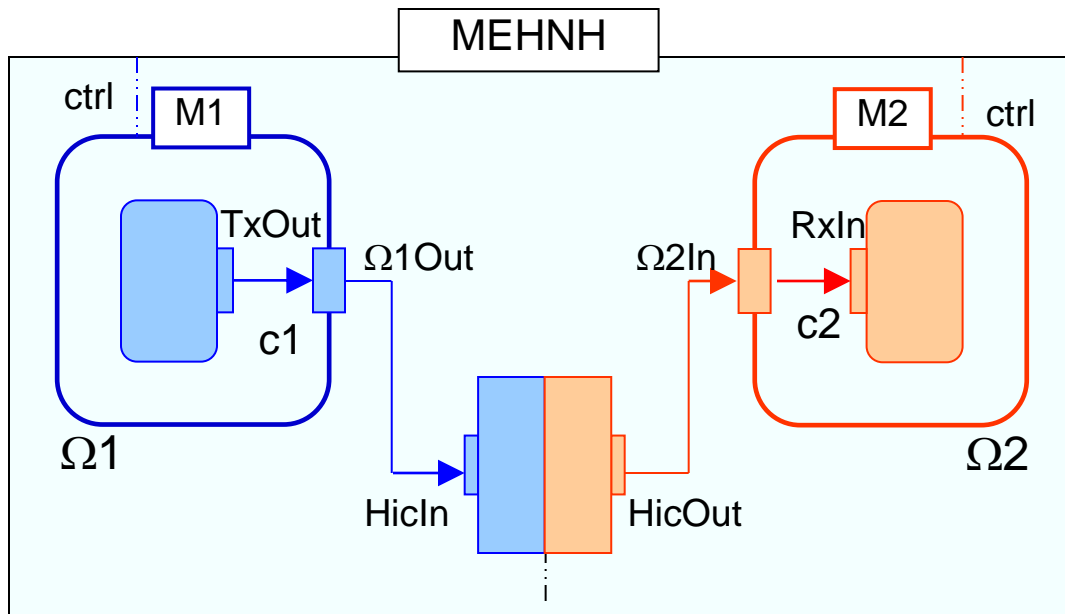
Modèle d'exécution dépend des Mocs utilisés par Ω_1 et Ω_2

Incompatibilité avec un ensemble ouvert de modèles

■ Approche Non-Hiérarchique : première tentative



■ Non-Hiérarchie par l'Abstraction Hiérarchique

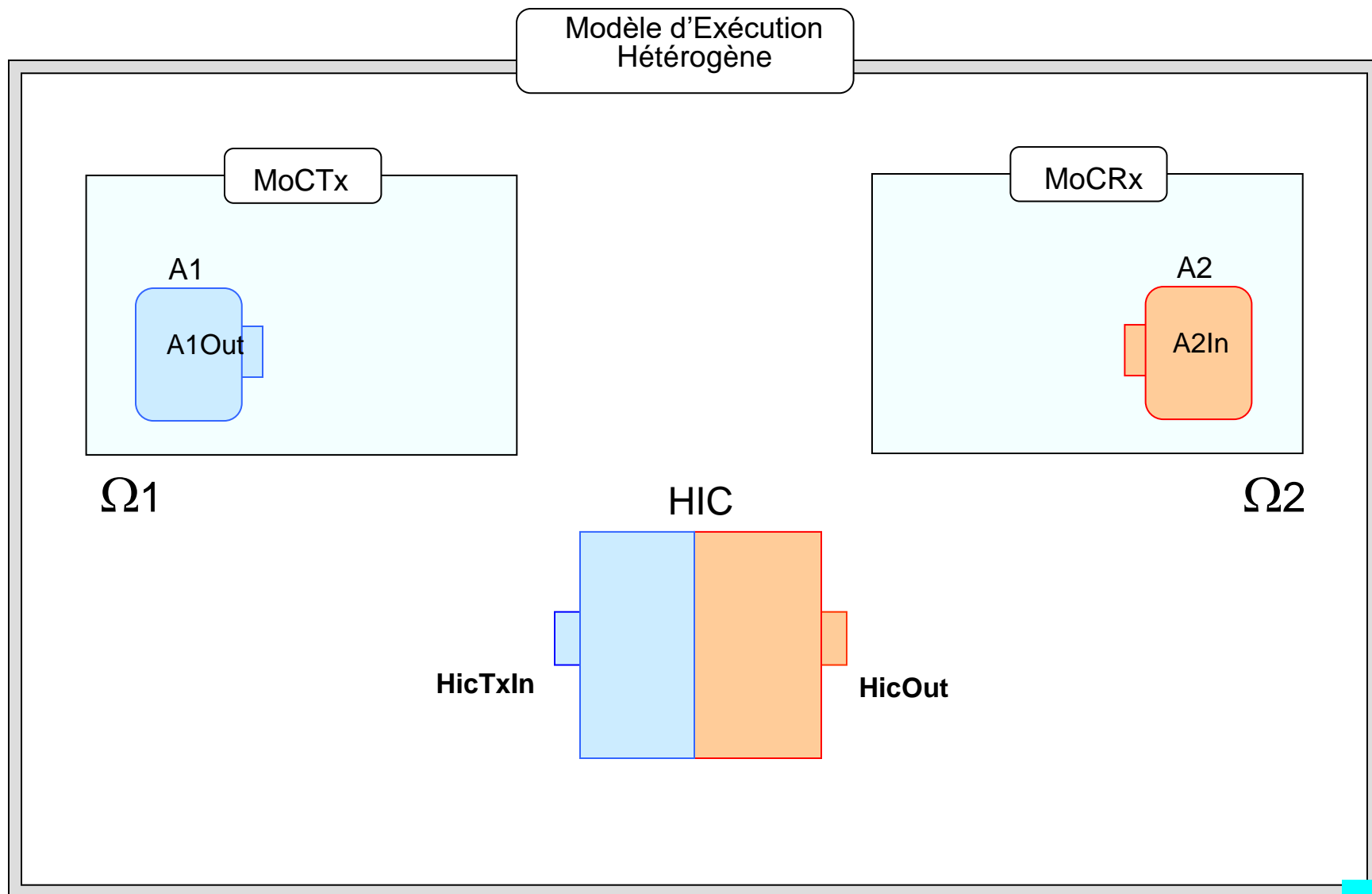


Inconvénient

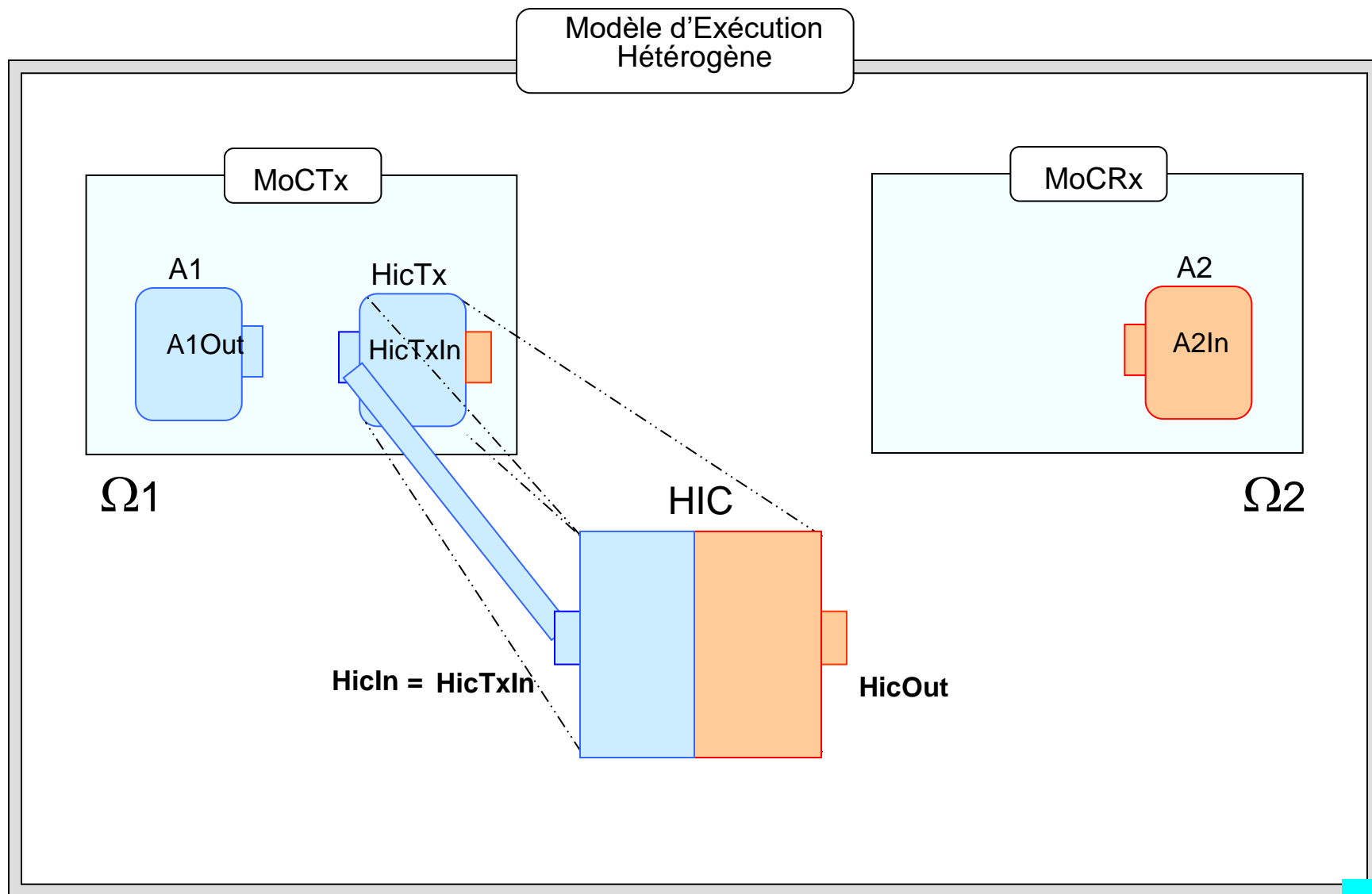
Modèle d'exécution dépend des Mocs utilisés par Ω_1 et Ω_2

Incompatibilité avec un ensemble ouvert de modèles

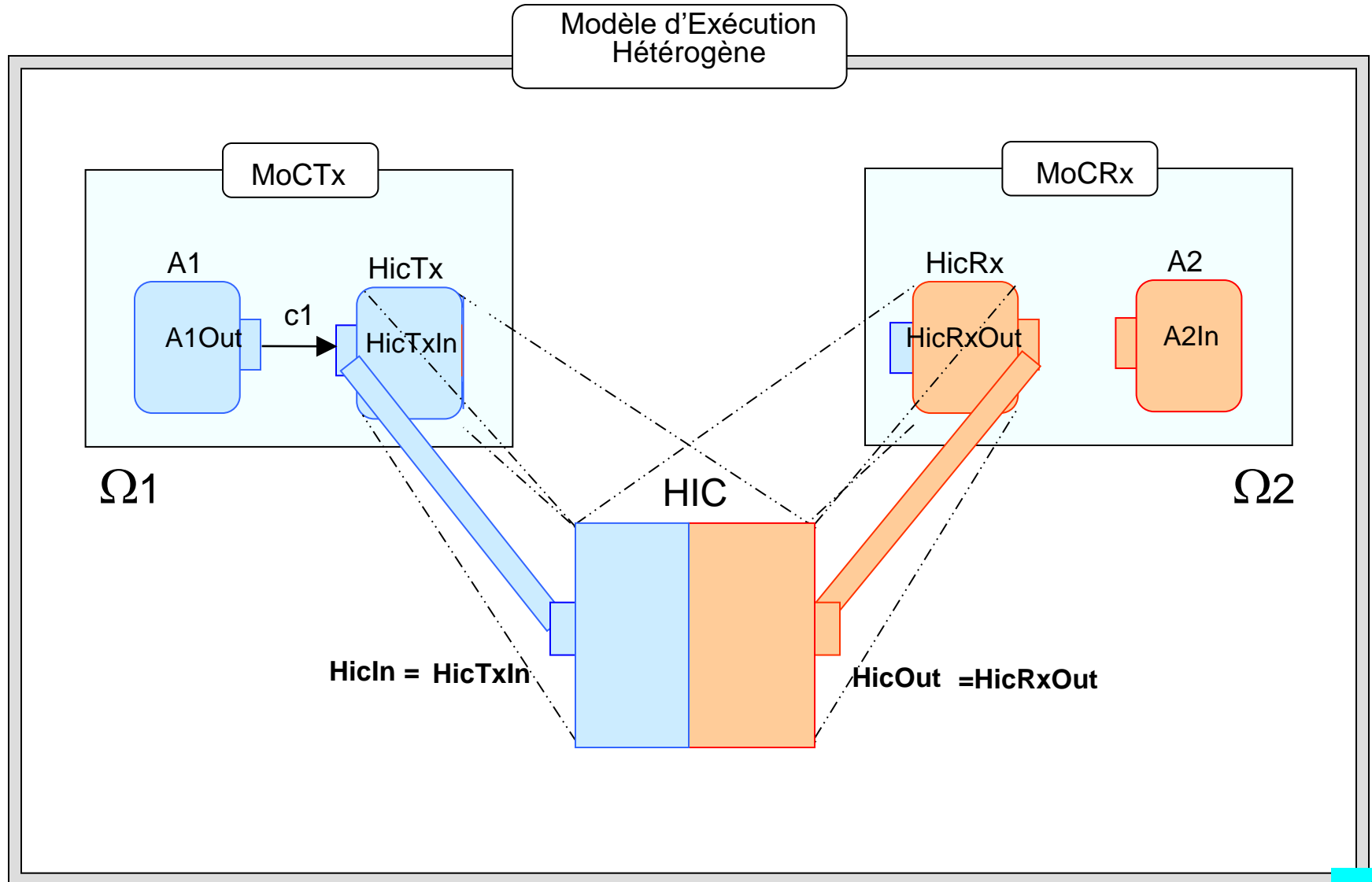
■ Non-Hiérarchie par l'Abstraction Non-Hiérarchique : Projection de HIC et Canal Abstrait Hétérogène



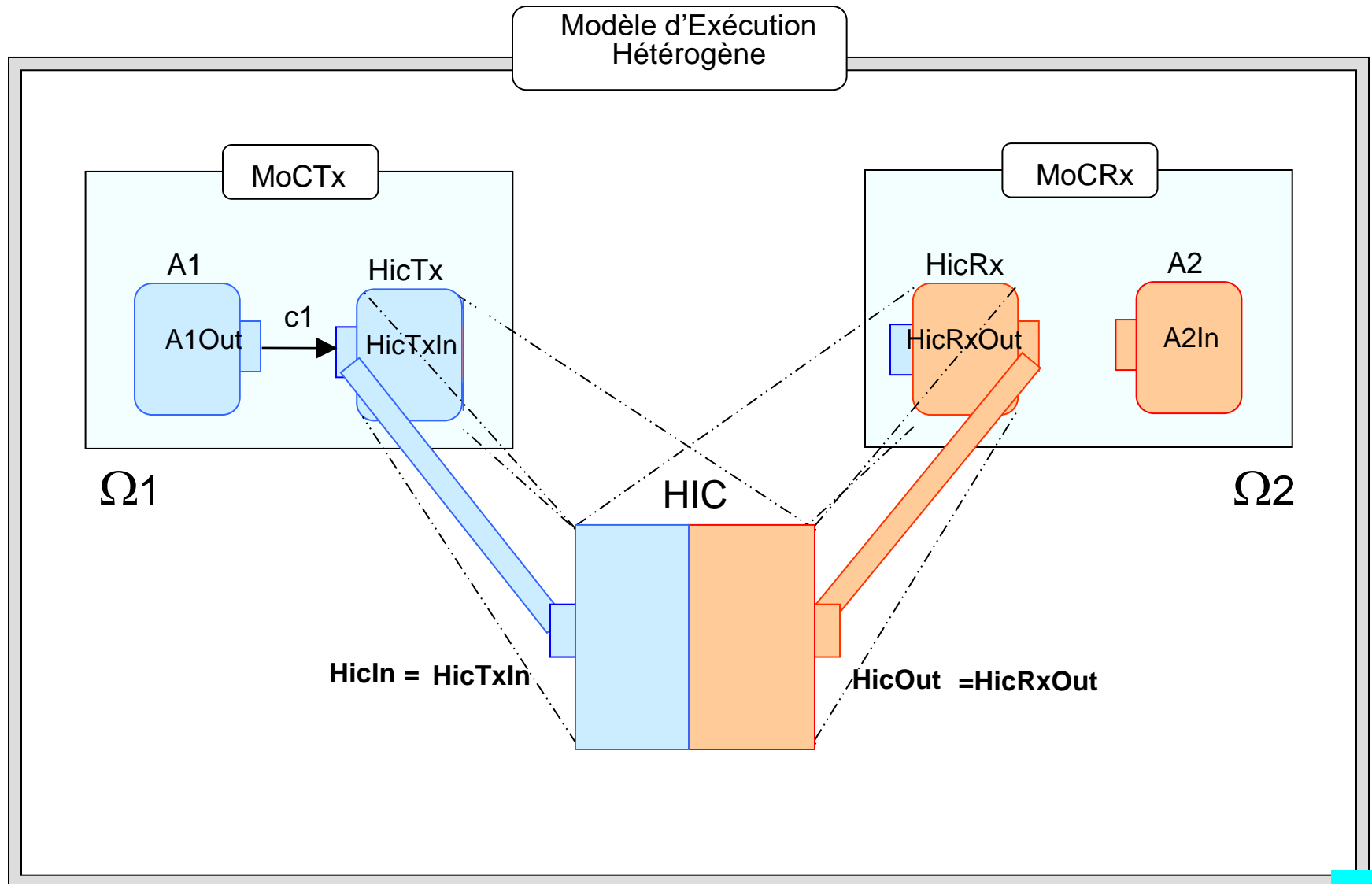
■ Non-Hiérarchie par l'Abstraction Non-Hiérarchique : Projection de HIC et Canal Abstrait Hétérogène



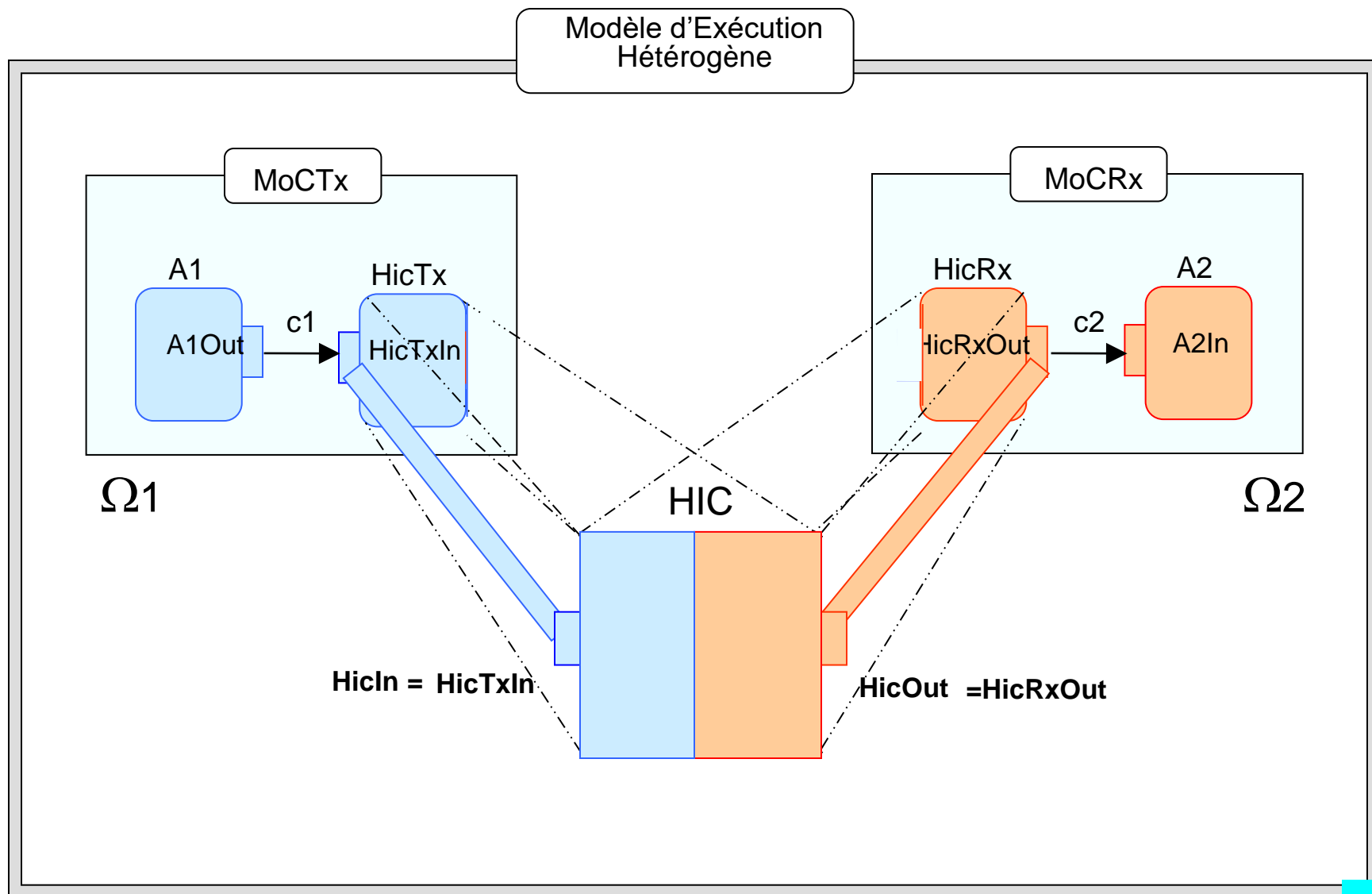
■ Non-Hiérarchie par l'Abstraction Non-Hiérarchique : Projection de HIC et Canal Abstrait Hétérogène



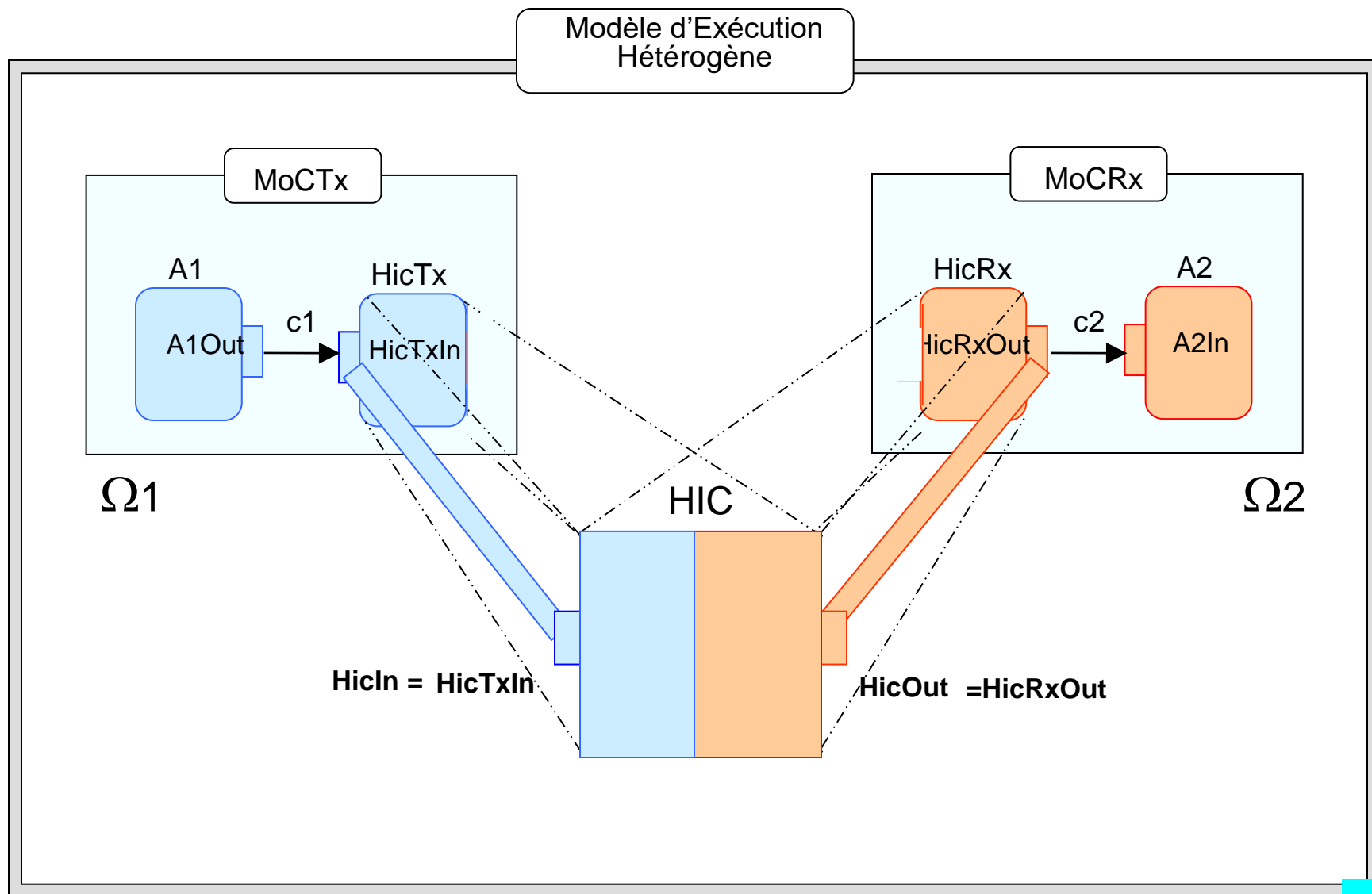
■ Non-Hiérarchie par l'Abstraction Non-Hiérarchique : Projection de HIC et Canal Abstrait Hétérogène



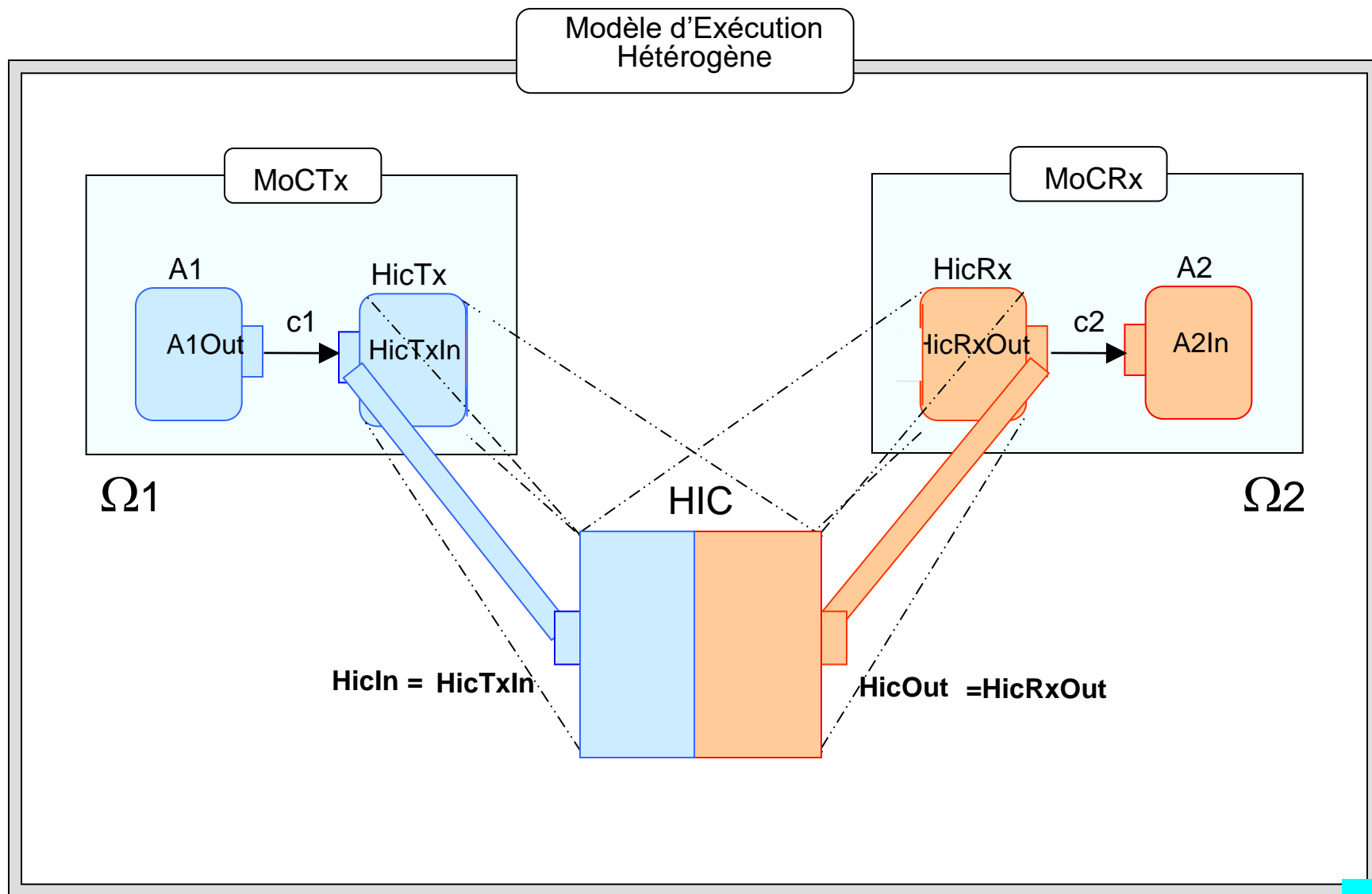
■ Non-Hiérarchie par l'Abstraction Non-Hiérarchique : Projection de HIC et Canal Abstrait Hétérogène



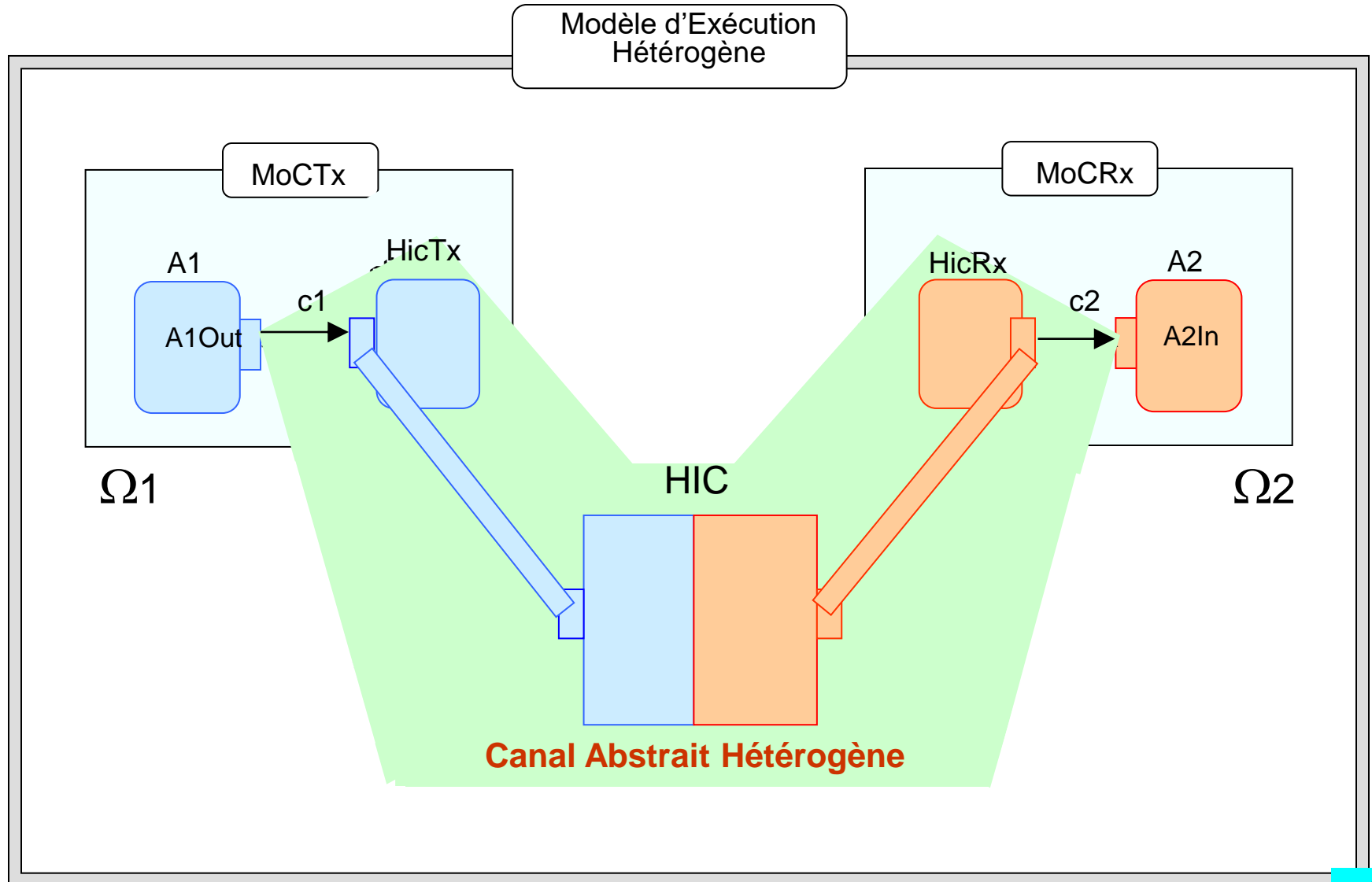
■ Non-Hiérarchie par l'Abstraction Non-Hiérarchique : Projection de HIC et Canal Abstrait Hétérogène



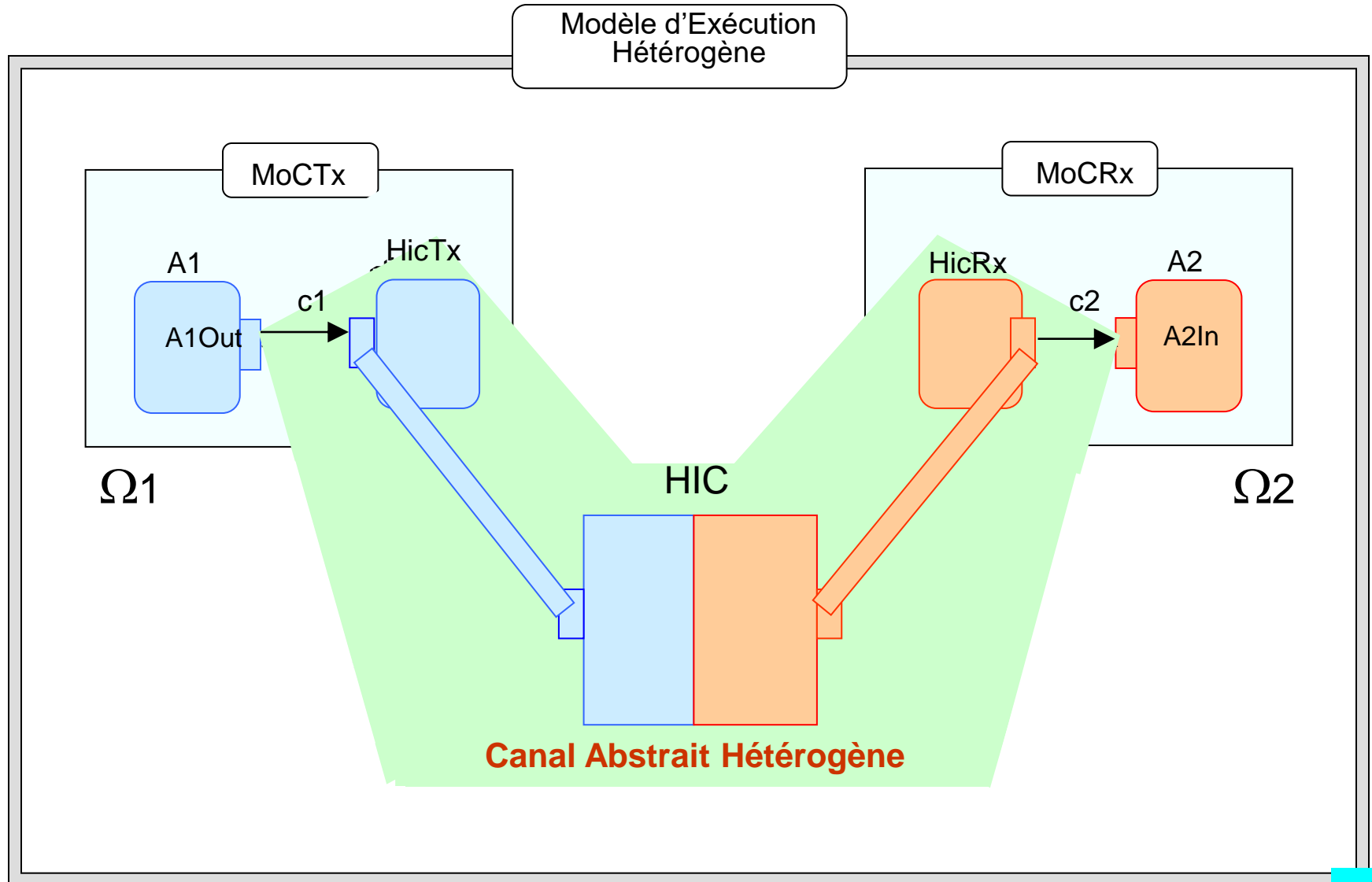
■ Non-Hiérarchie par l'Abstraction Non-Hiérarchique : Projection de HIC et Canal Abstrait Hétérogène



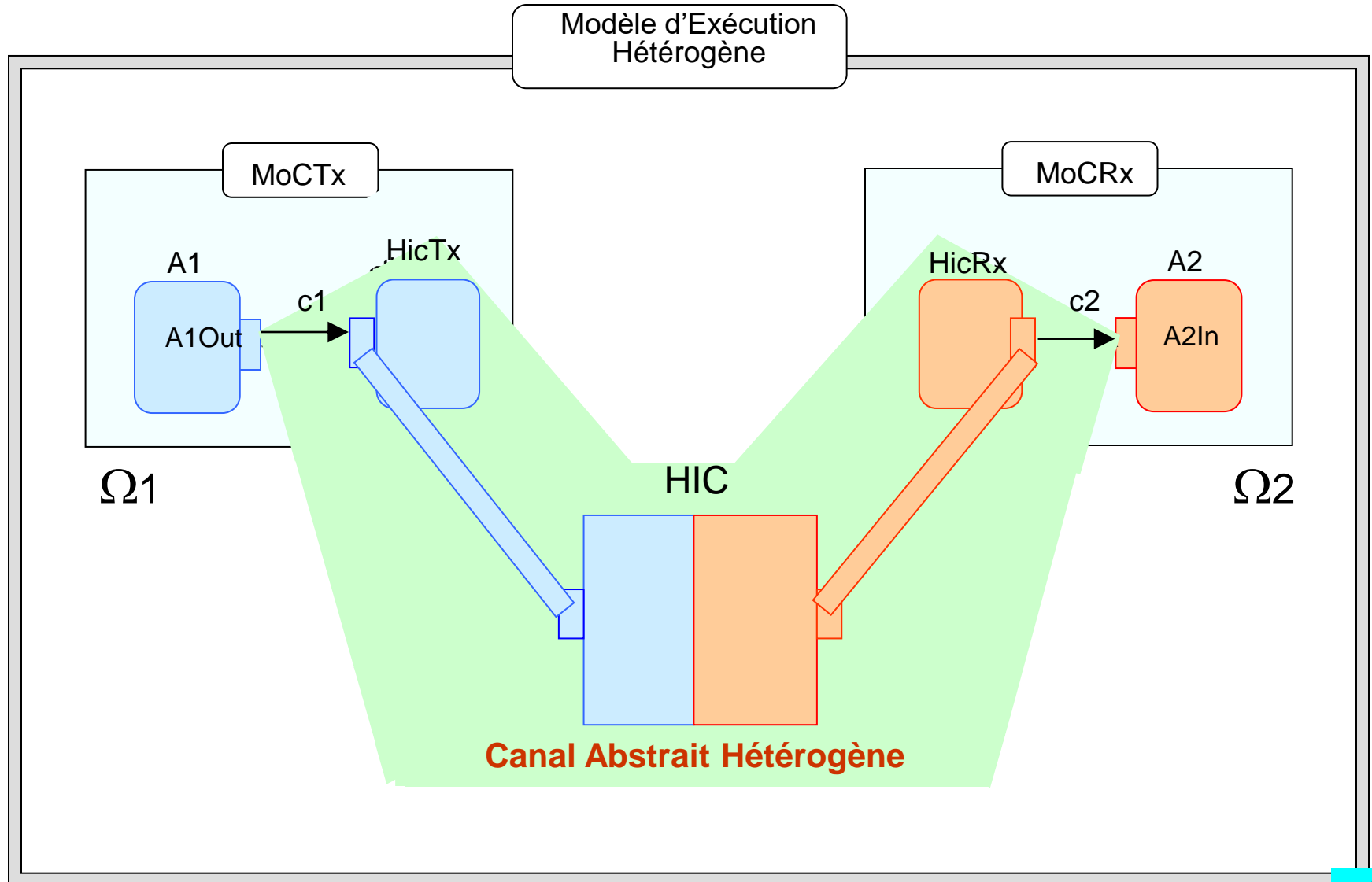
■ Non-Hiérarchie par l'Abstraction Non-Hiérarchique : Projection de HIC et Canal Abstrait Hétérogène



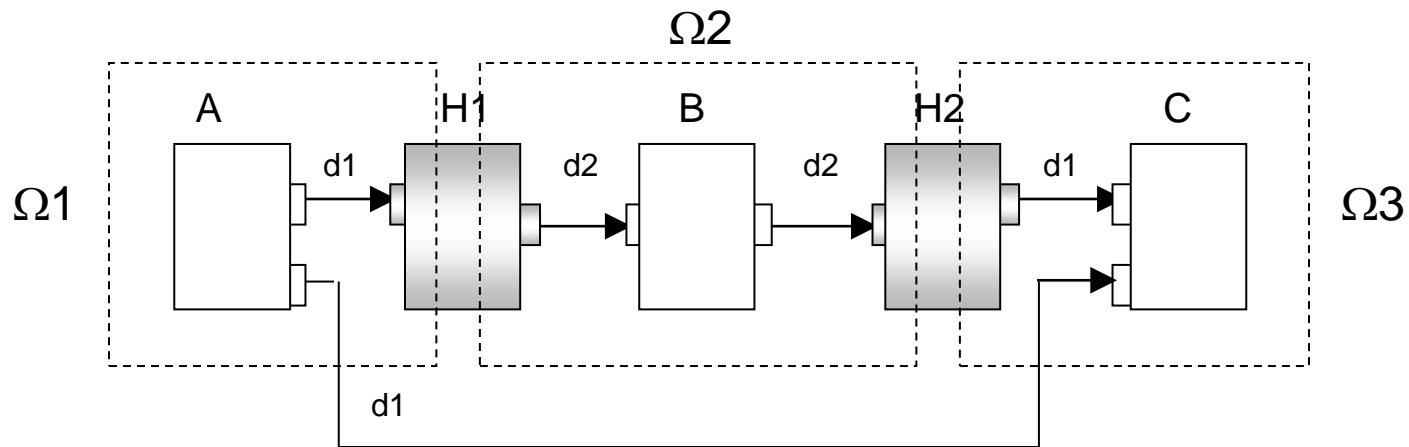
■ Non-Hiérarchie par l'Abstraction Non-Hiérarchique : Projection de HIC et Canal Abstrait Hétérogène



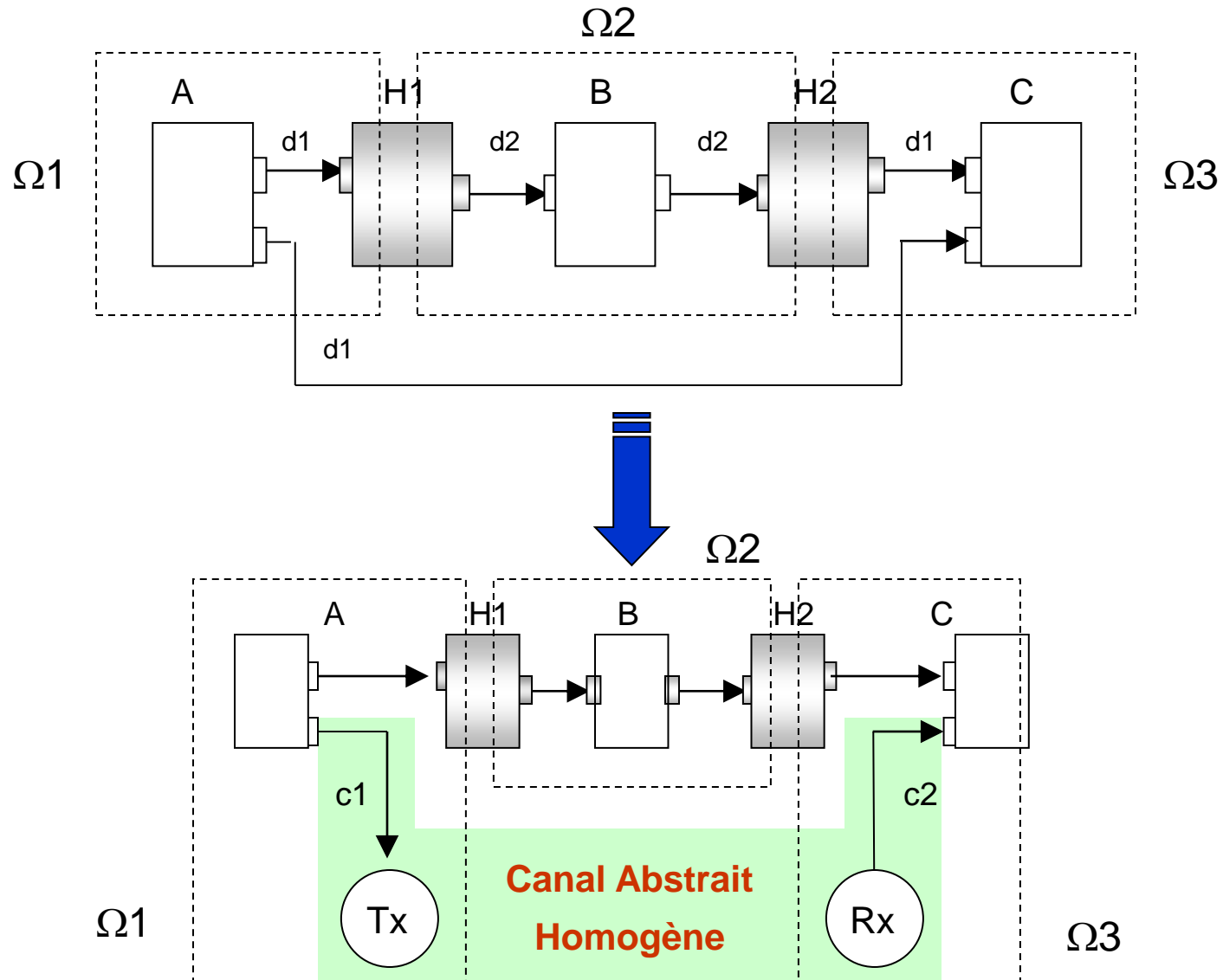
■ Non-Hiérarchie par l'Abstraction Non-Hiérarchique : Projection de HIC et Canal Abstrait Hétérogène



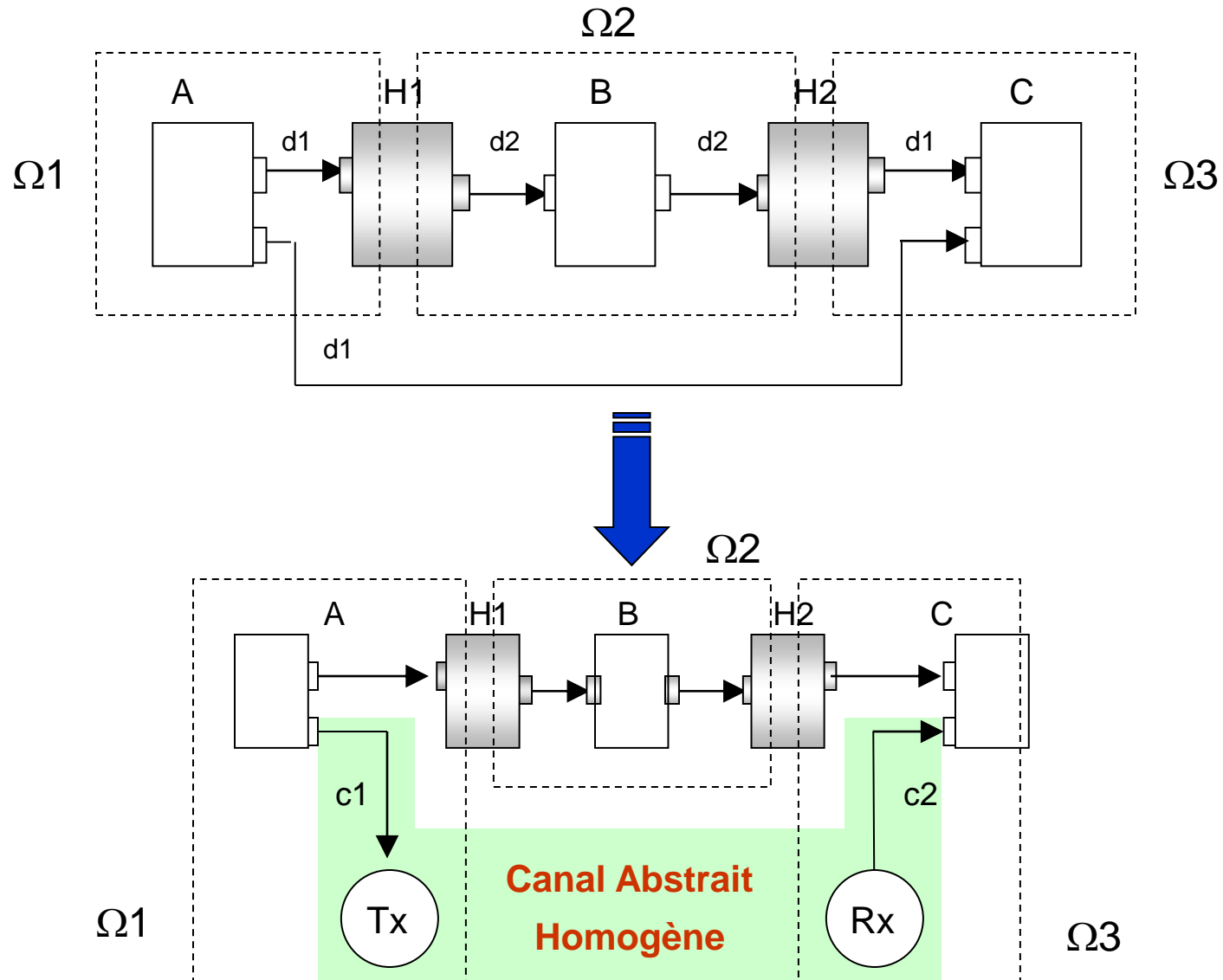
■ Projection de HIC et Canal Abstrait Homogène



■ Projection de HIC et Canal Abstrait Homogène



■ Projection de HIC et Canal Abstrait Homogène



Plan

- **Systèmes Embarqués**
 - Positionnement du marché
 - Caractéristiques
 - Outils de modélisation et Modélisation hiérarchique

- **Problématique et Objectif**

- **Approche hétérogène non-hiérarchique**

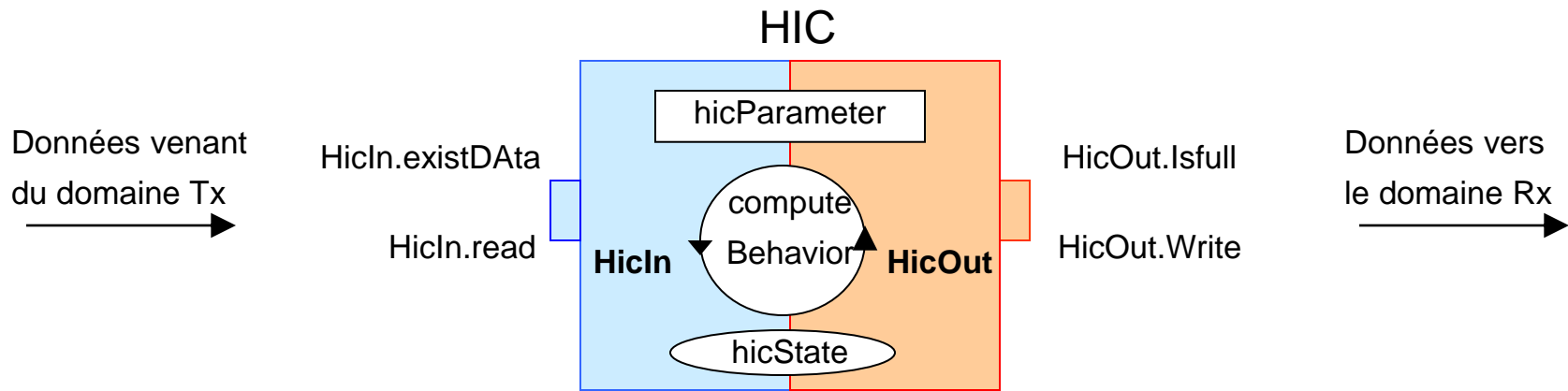
- **Modélisation :**
 - Composant à Interface Hétérogène (HIC)
 - Modèle d'Exécution Hétérogène Non-Hiérarchique

- **Intégration et Validation par Simulation dans Ptolemy II**

- **Conclusion et perspectives**

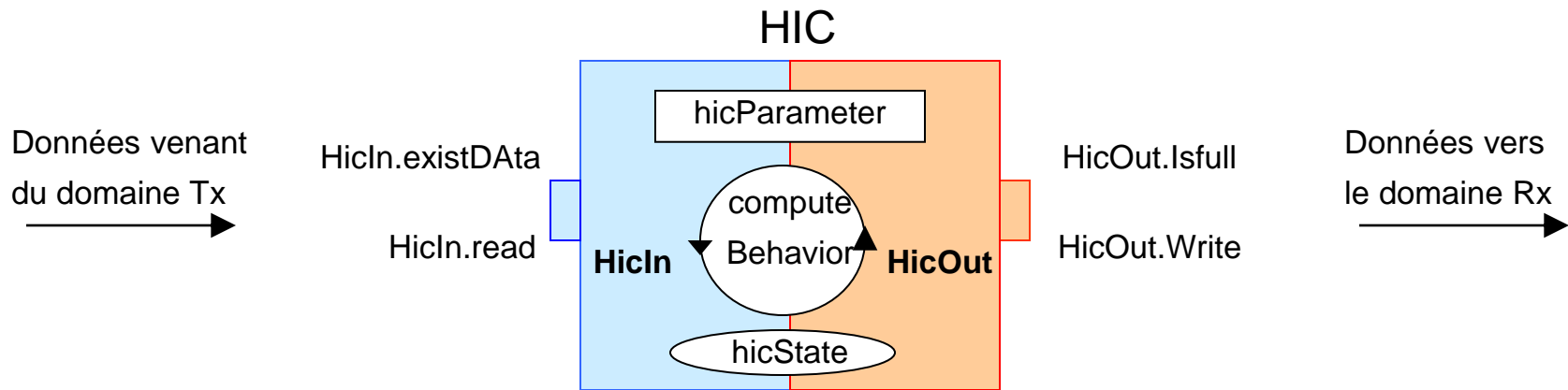
Modélisation du HIC

■ Structure de HIC



Modélisation du HIC

■ Structure de HIC

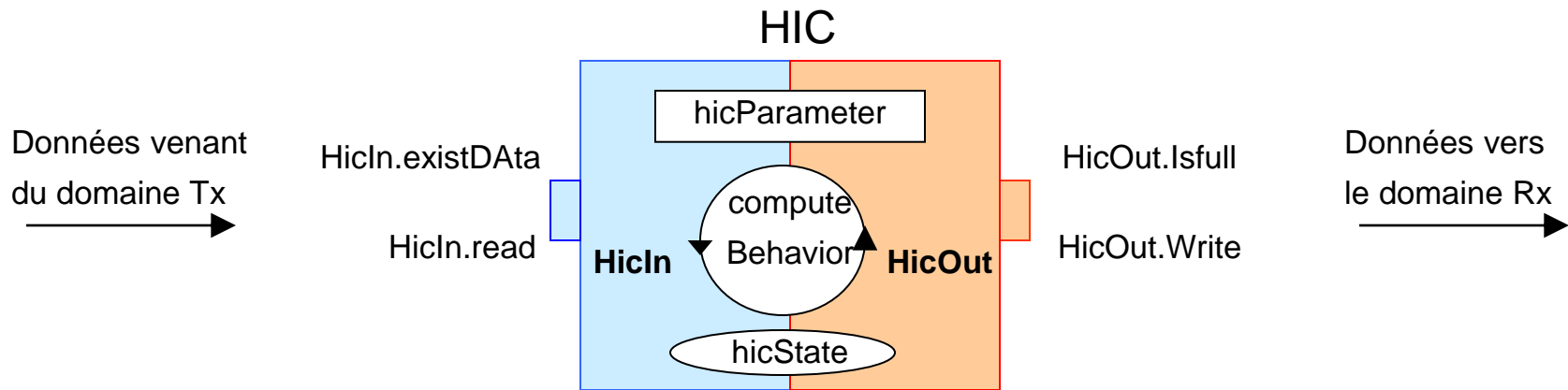


■ Opérations du modèle Acteur

Flot de données	
Comportement	Communication
computeBehavior	read write isFull existData

Modélisation du HIC

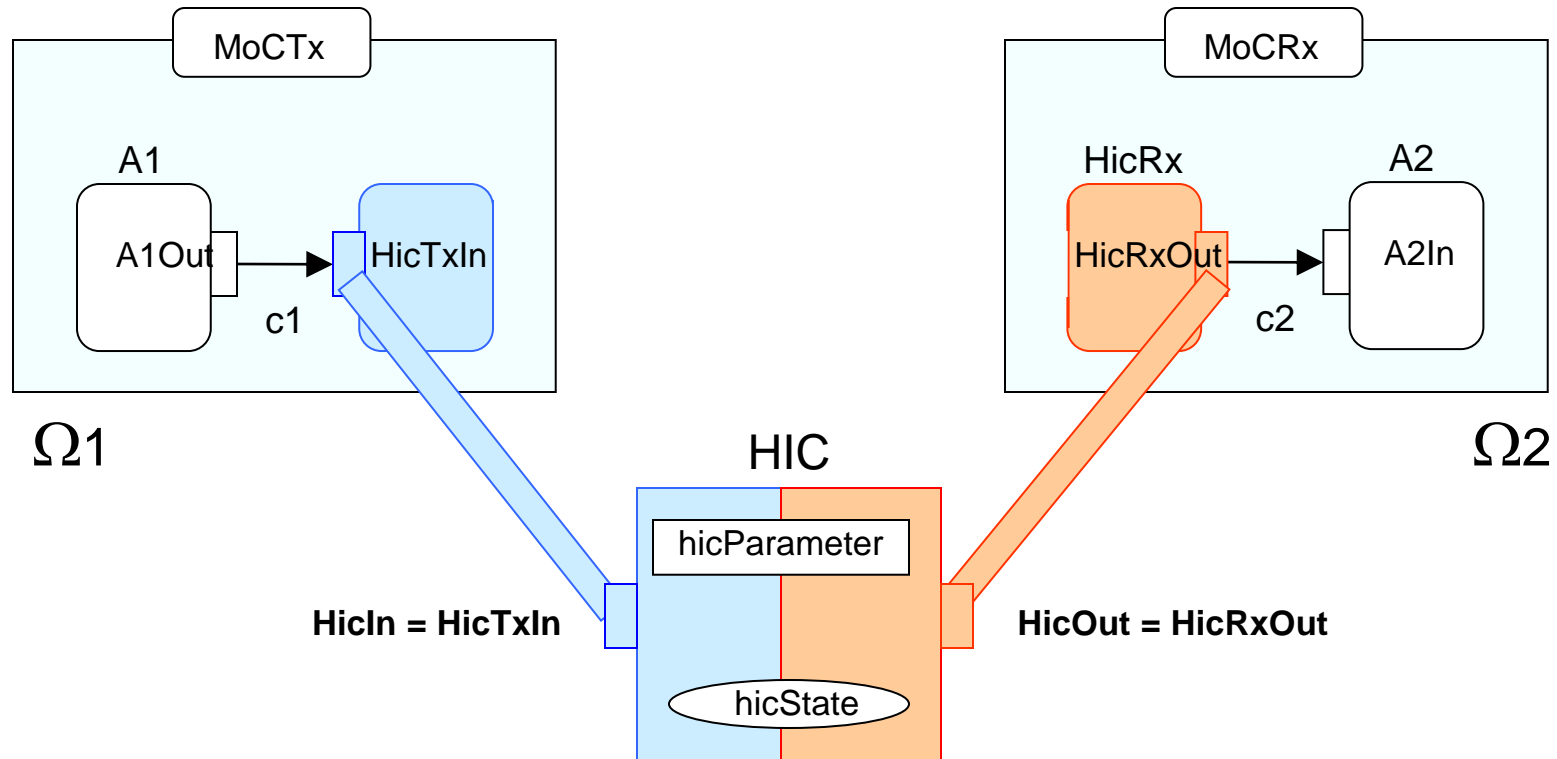
■ Structure de HIC



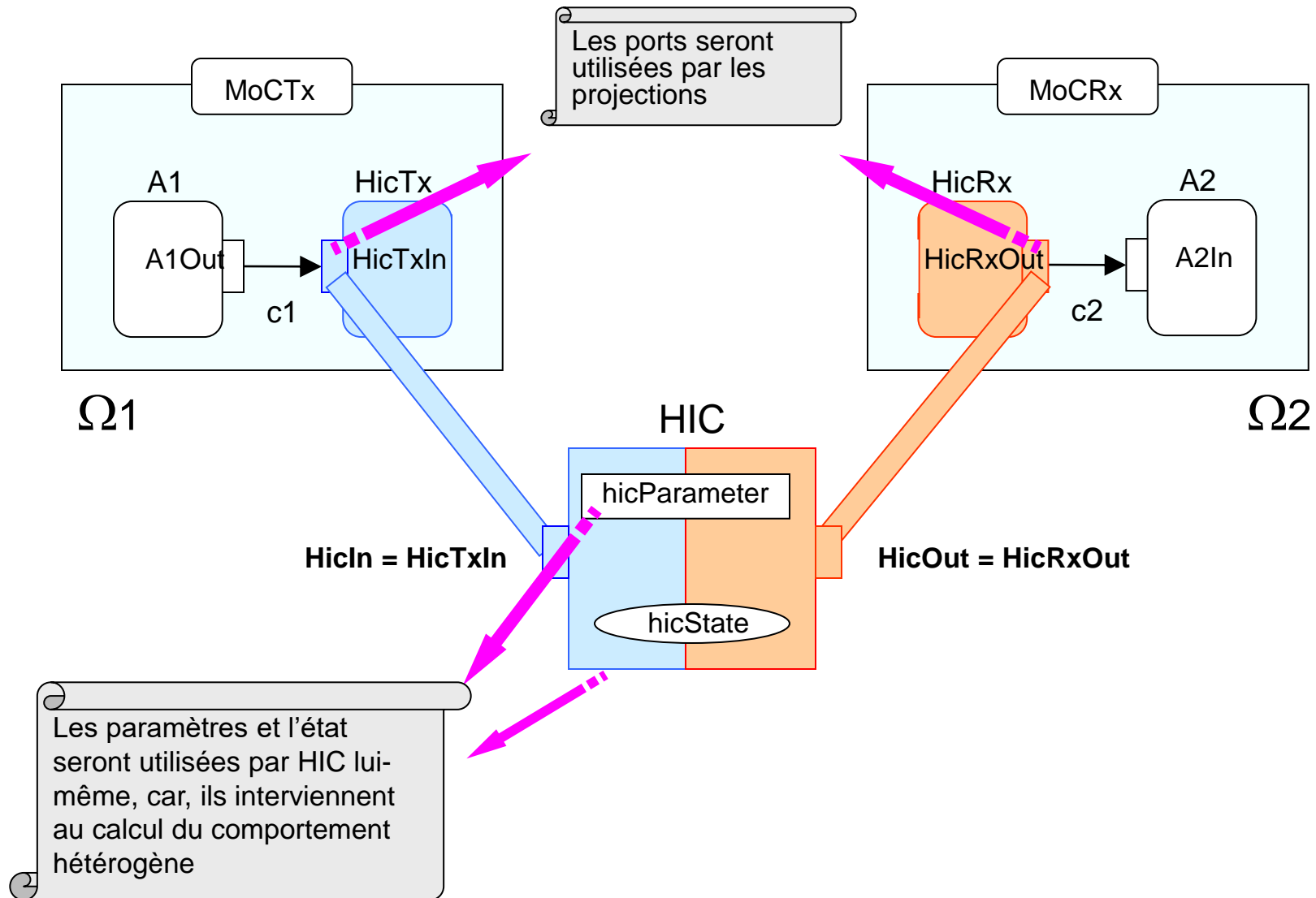
■ Opérations du modèle Acteur

Flot de données		Flot de Contrôle
Comportement	Communication	initialisation
computeBehavior	read	preCondition
	write	trigger
	isFull	postCondition
	existData	finish

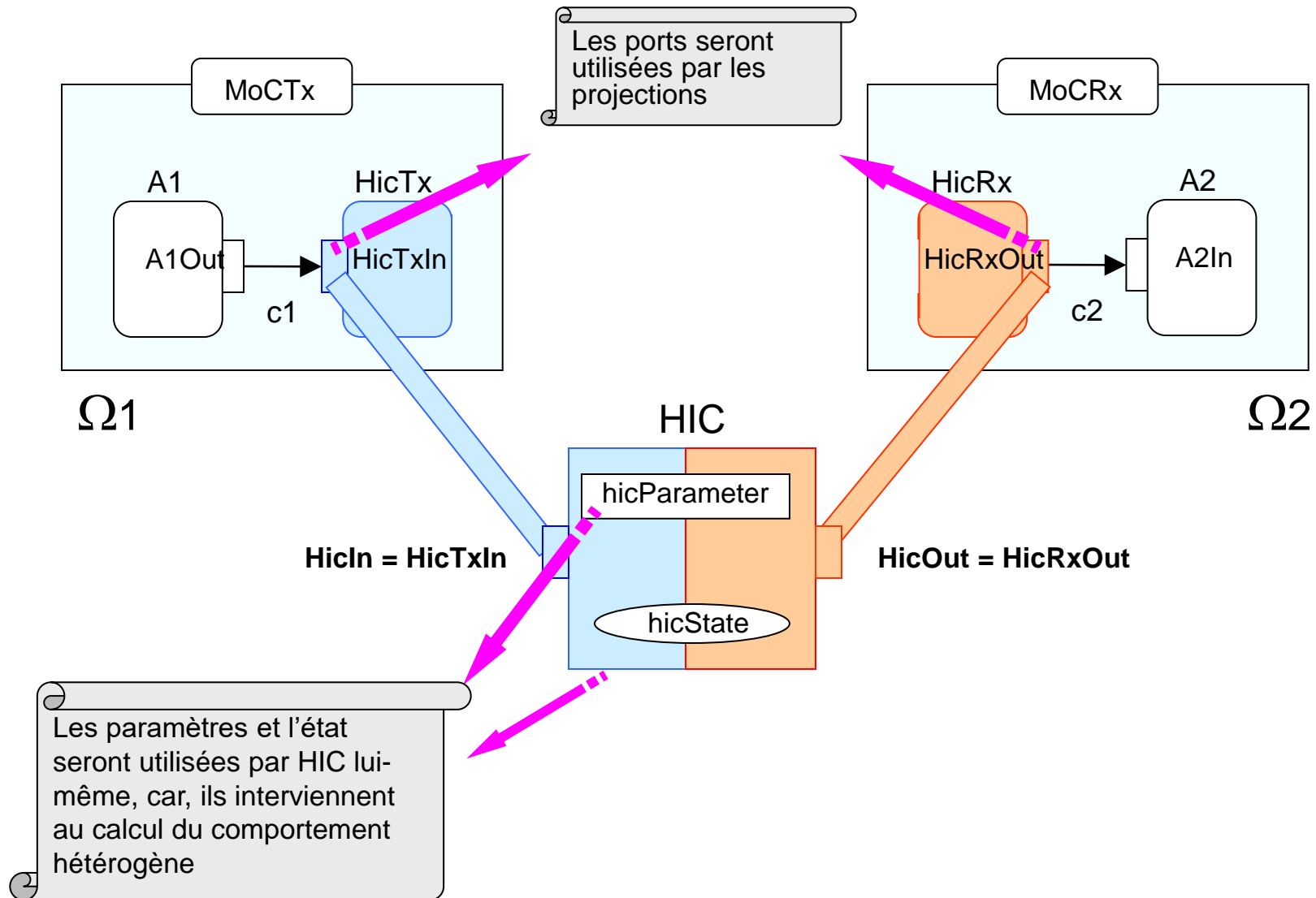
■ Partitionnement Structurel de HIC



■ Partitionnement Structurel de HIC

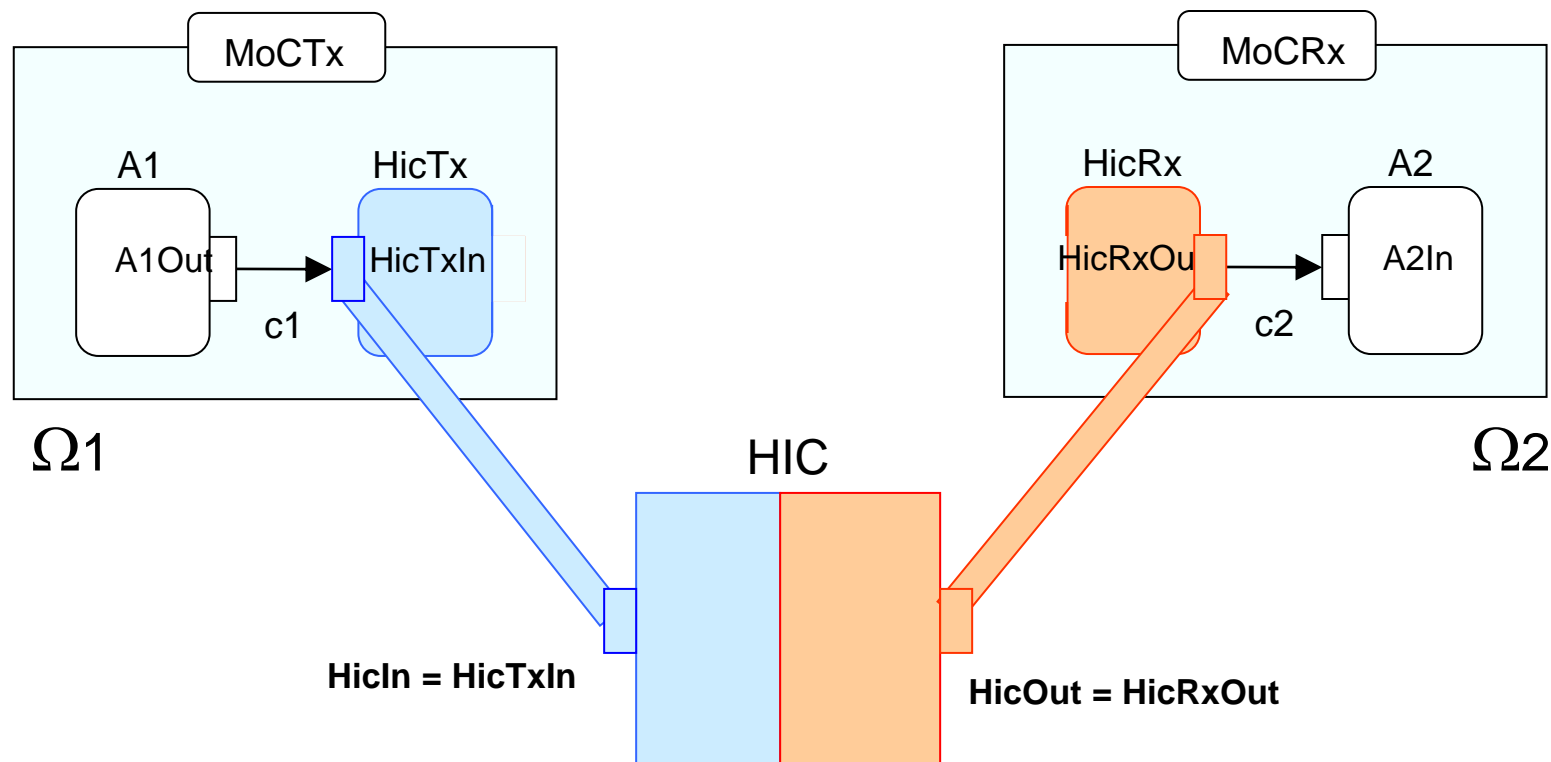


■ Partitionnement Structurel de HIC



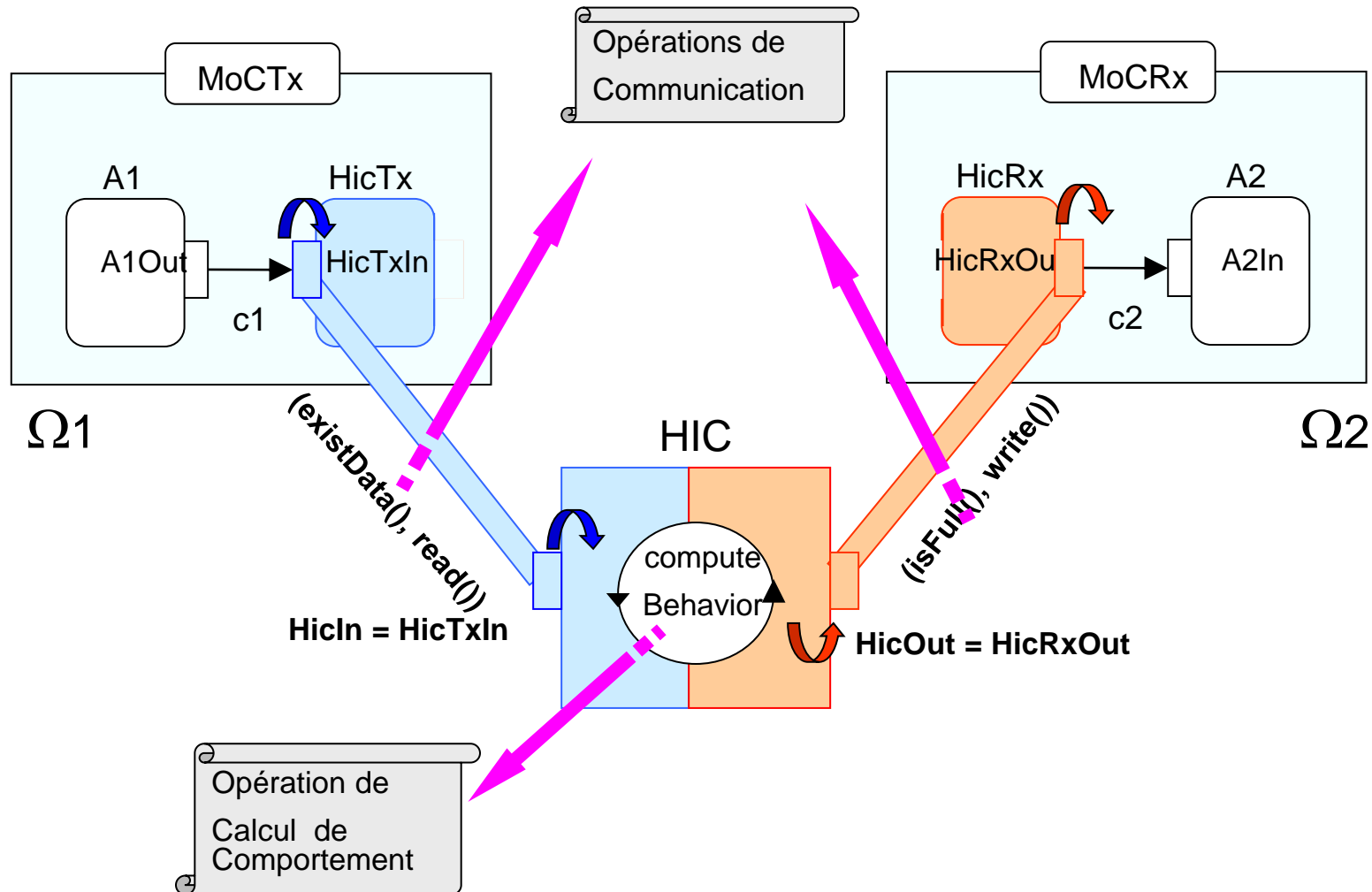
■ Partitionnement Opérationnel de HIC :

- Opérations de Communication et de Calcul de Comportement



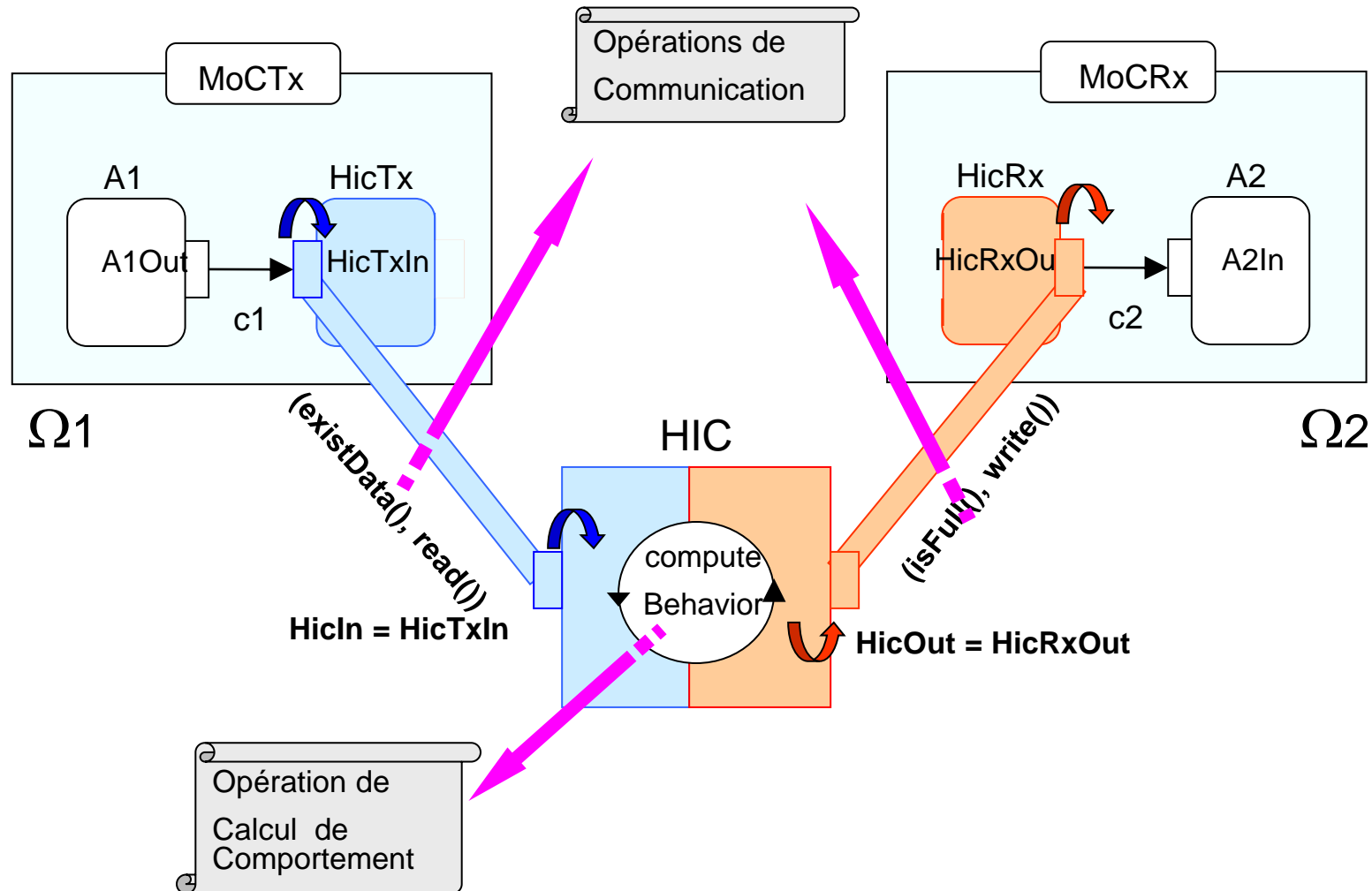
■ Partitionnement Opérationnel de HIC :

- Opérations de Communication et de Calcul de Comportement



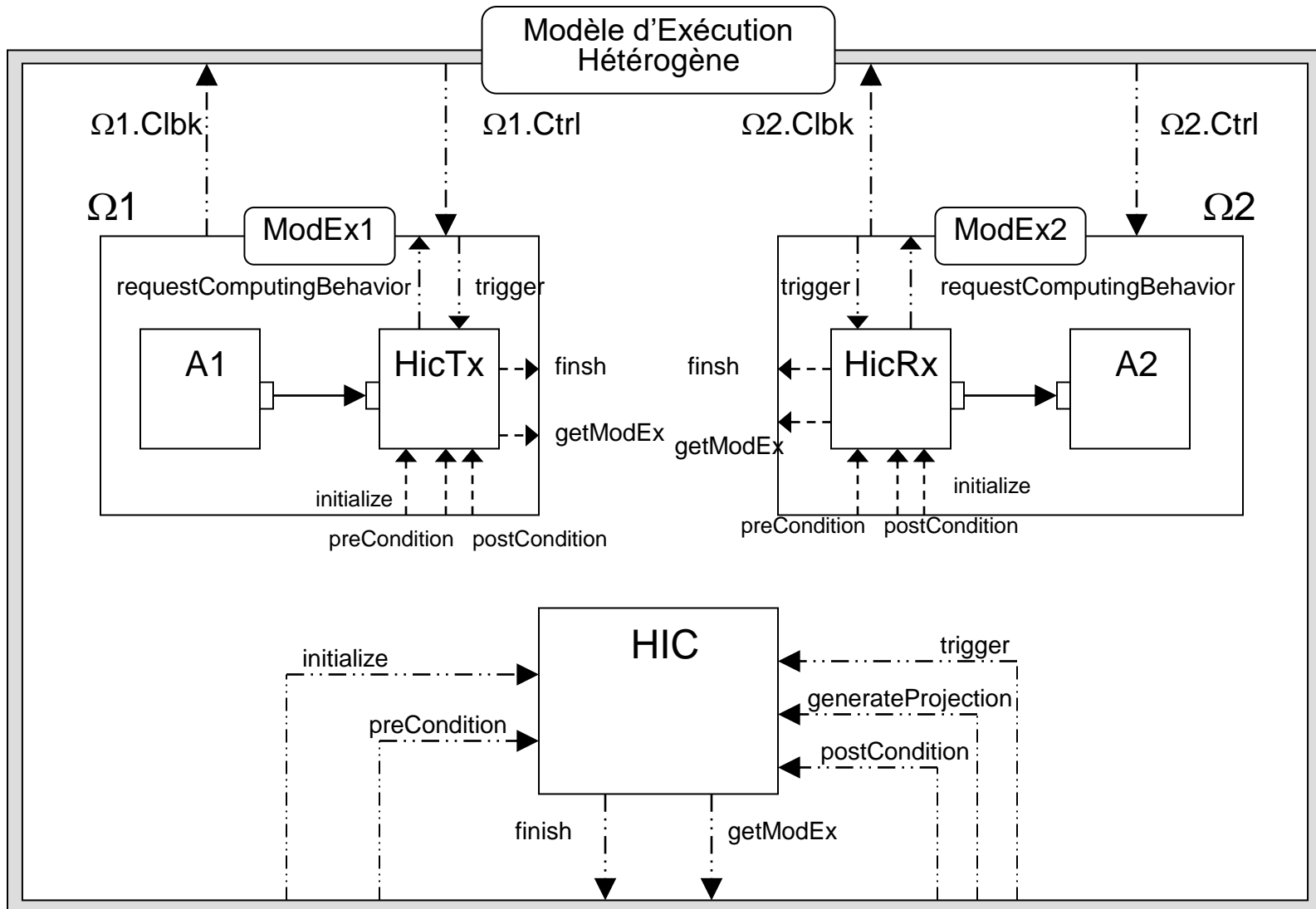
■ Partitionnement Opérationnel de HIC :

- Opérations de Communication et de Calcul de Comportement

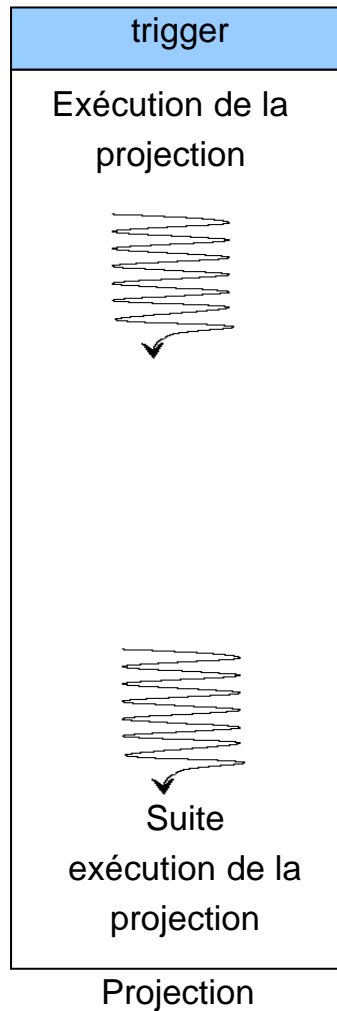


■ Partitionnement Opérationnel de HIC :

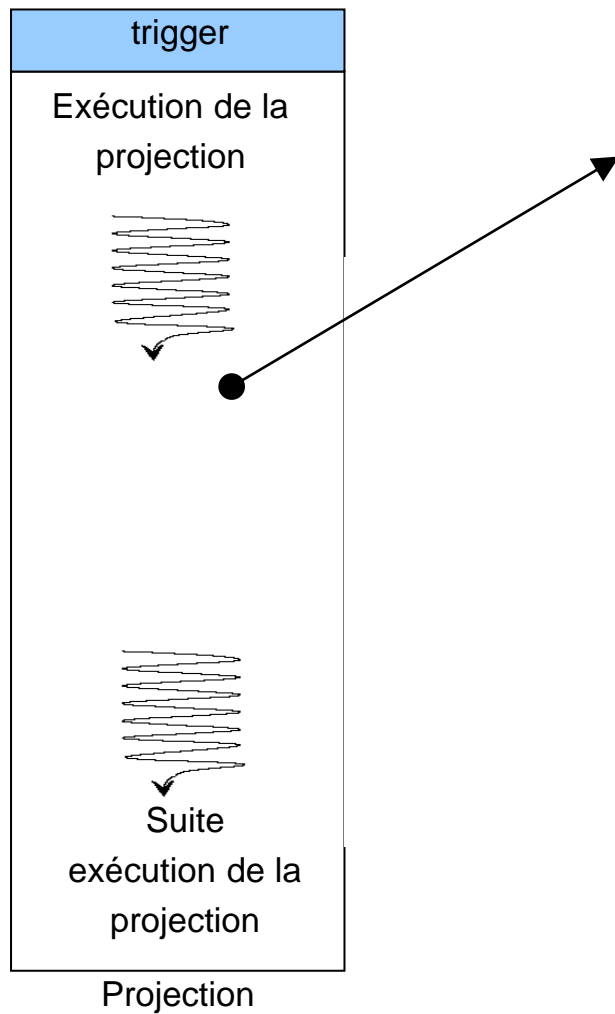
- Opérations de Contrôle



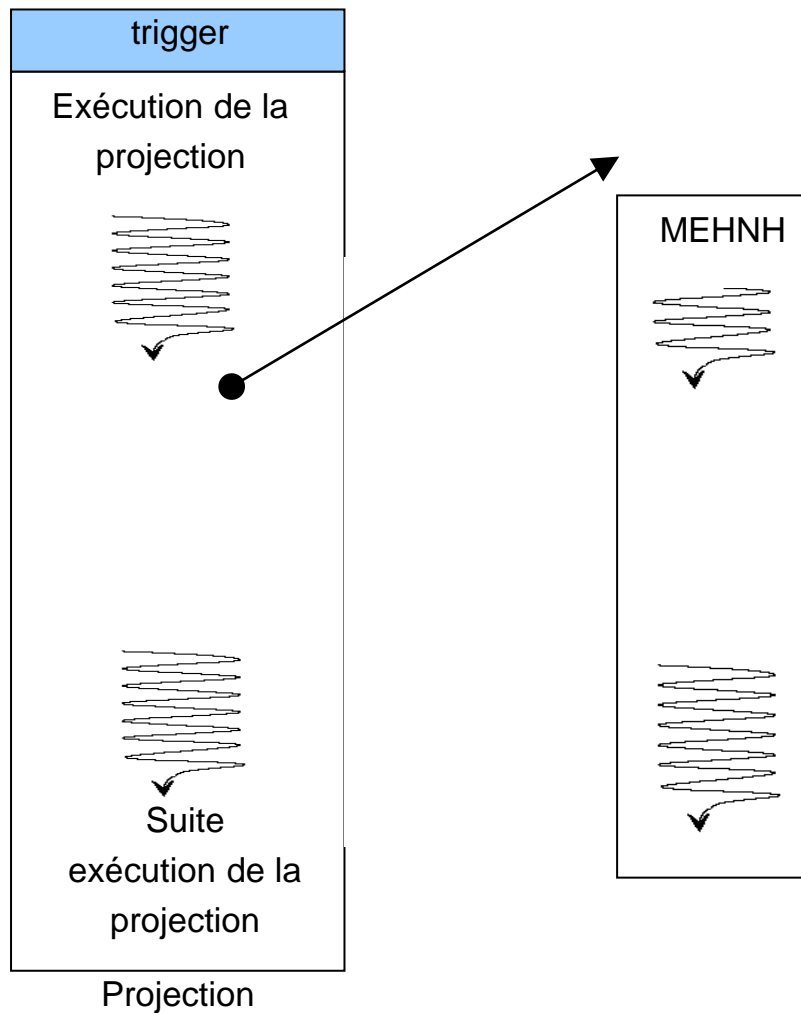
■ Calcul du comportement de HIC



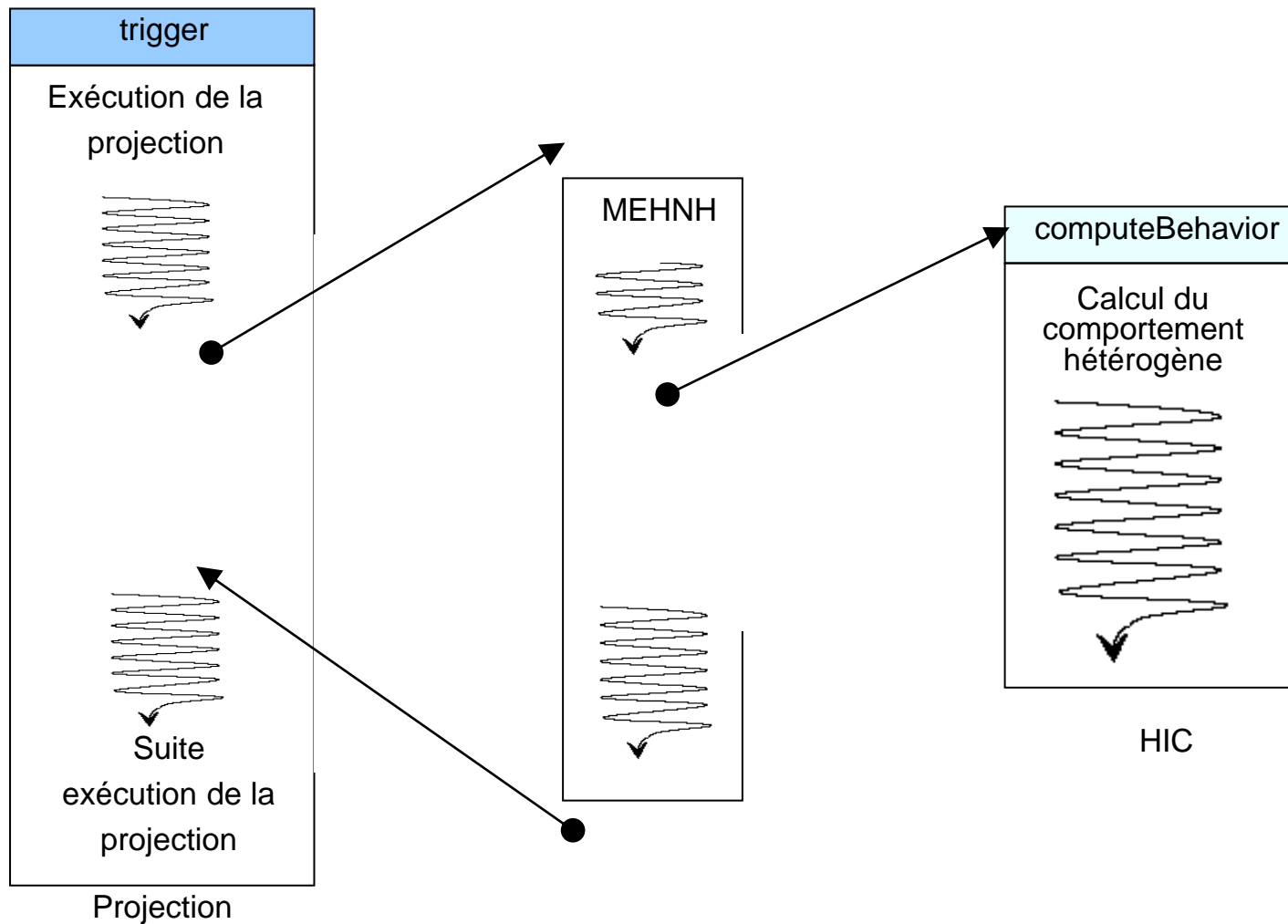
■ Calcul du comportement de HIC



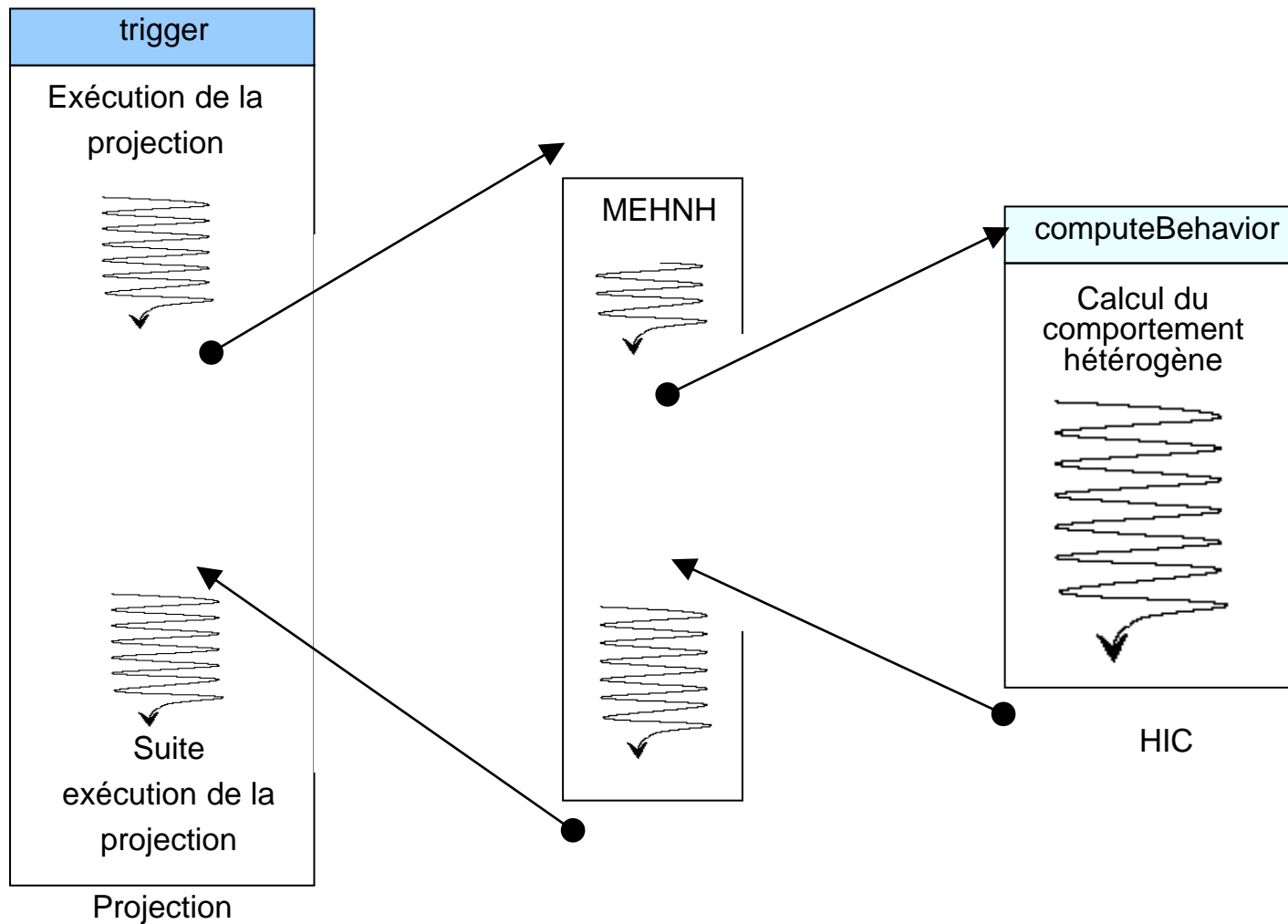
■ Calcul du comportement de HIC



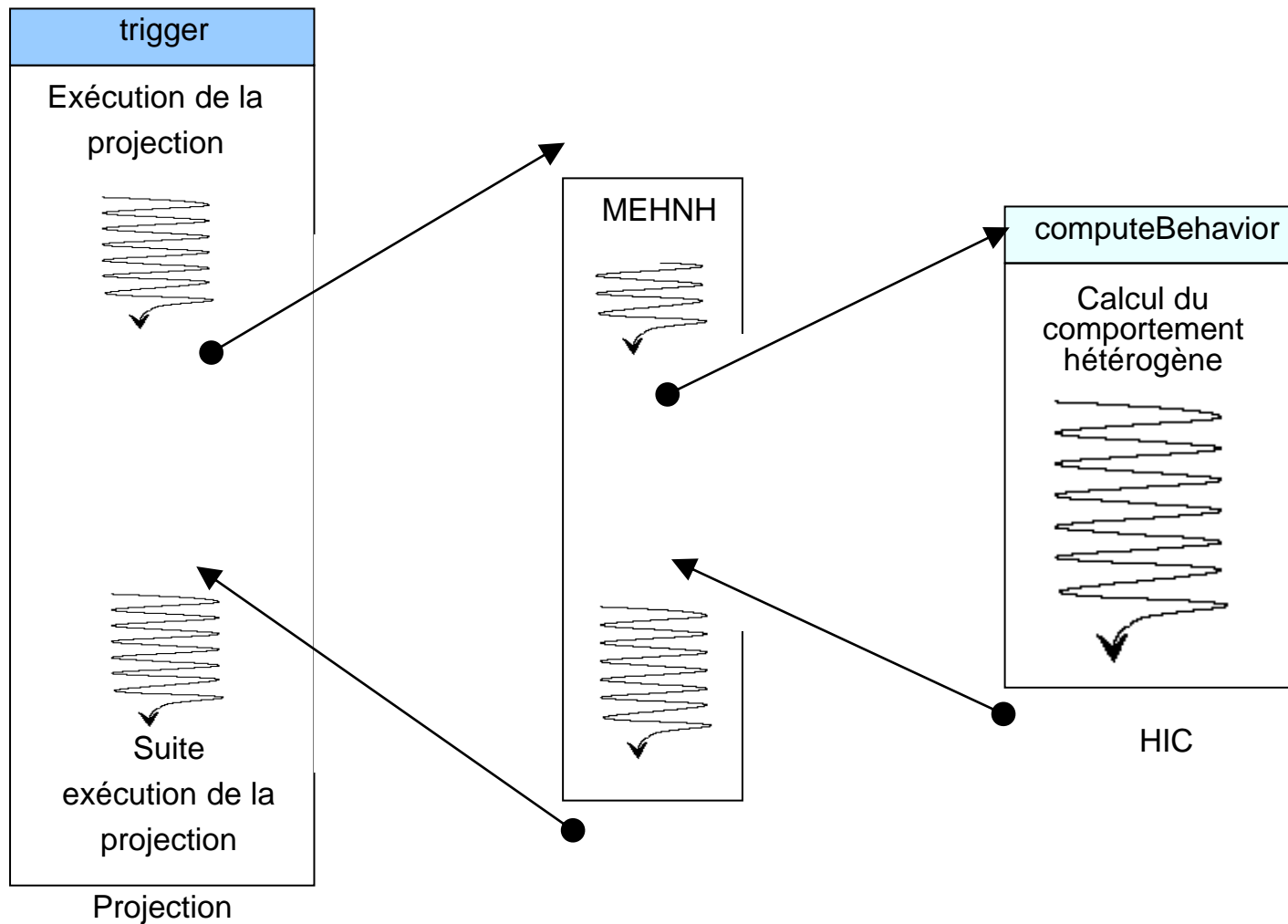
■ Calcul du comportement de HIC



■ Calcul du comportement de HIC



■ Calcul du comportement de HIC



■ Synthèse du partitionnement Opérationnel de HIC

	Tx	HIC	Rx
Communication	existData read	read write isFull existData	isFull write

■ Synthèse du partitionnement Opérationnel de HIC

	Tx	HIC	Rx
Communication	existData read	read write isFull existData	isFull write
Comportement		computeBehavior	

■ Synthèse du partitionnement Opérationnel de HIC

	Tx	HIC	Rx
Communication	existData read	read write isFull existData	isFull write
Comportement		computeBehavior	
Contrôle	initialisation preCondition Trigger getModEx requestBehaviorComputing postCondition finish	initialisation preCondition Trigger getModEx postCondition finish	initialisation preCondition Trigger getModEx requestBehaviorComputing postCondition finish

Modélisation du Modèle d'Exécution Hétérogène Non-Hiérarchique

■ Partitionnement du système

- Génération des sous-systèmes et placement des acteurs
- Déconnexion et Reconnexion des ports
- Création des Ports virtuels

Modélisation du Modèle d'Exécution Hétérogène Non-Hiérarchique

■ Partitionnement du système

- Génération des sous-systèmes et placement des acteurs
- Déconnexion et Reconnection des ports
- Création des Ports virtuels

■ Ordonnancement statique des sous-systèmes

Modélisation du Modèle d'Exécution Hétérogène Non-Hiérarchique

■ Partitionnement du système

- Génération des sous-systèmes et placement des acteurs
- Déconnexion et Reconnexion des ports
- Création des Ports virtuels

■ Ordonnancement statique des sous-systèmes

■ Exécution

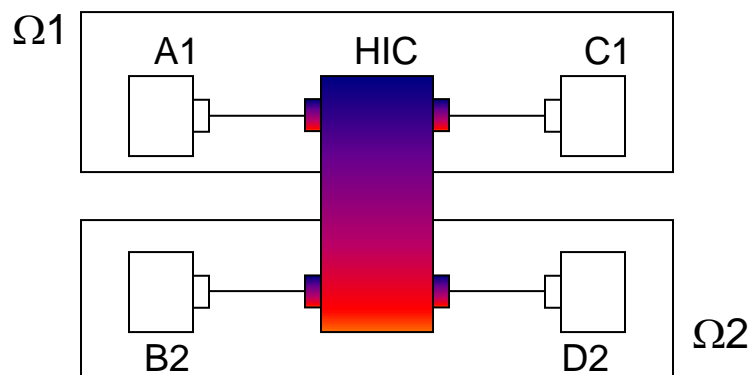
- Délégation du calcul du comportement des sous-systèmes à leur modèle
- Coordination des communications entre sous-systèmes

■ Partitionnement des sous-systèmes

- Tri topologique
- Pour chaque acteur A
- S'il existe un sous-système S utilisant le MoC de A alors
 - On place A dans S uniquement s'il n'existe pas de chemin entre A et un quelconque acteur de S passant par un HIC
 - Sinon, on crée un nouveau sous-système et on y place A
- Sinon, on crée un nouveau sous-système et on y place A
- L'absence de chemin passant par un HIC garantit l'absence de dépendances cycliques entre sous-systèmes

■ Partitionnement des sous-systèmes

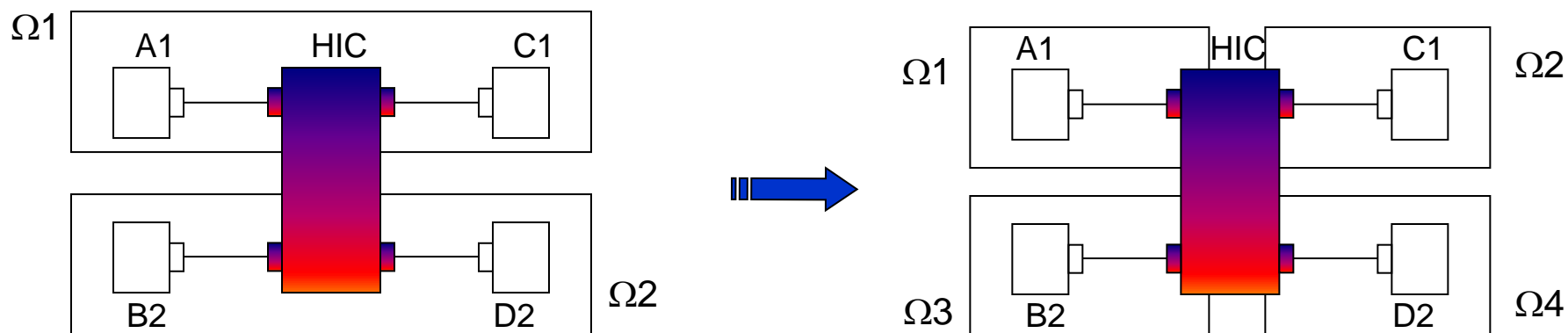
- Tri topologique
- Pour chaque acteur A
- S'il existe un sous-système S utilisant le MoC de A alors
 - On place A dans S uniquement s'il n'existe pas de chemin entre A et un quelconque acteur de S passant par un HIC
 - Sinon, on crée un nouveau sous-système et on y place A
- Sinon, on crée un nouveau sous-système et on y place A
- L'absence de chemin passant par un HIC garantit l'absence de dépendances cycliques entre sous-systèmes



HIC/ Ω_1 doit être activé après l'activation B2 et
HIC/ Ω_2 doit être activé après l'activation de A1
D'où l'impossibilité d'ordonnancer Ω_1 et Ω_2 .

■ Partitionnement des sous-systèmes

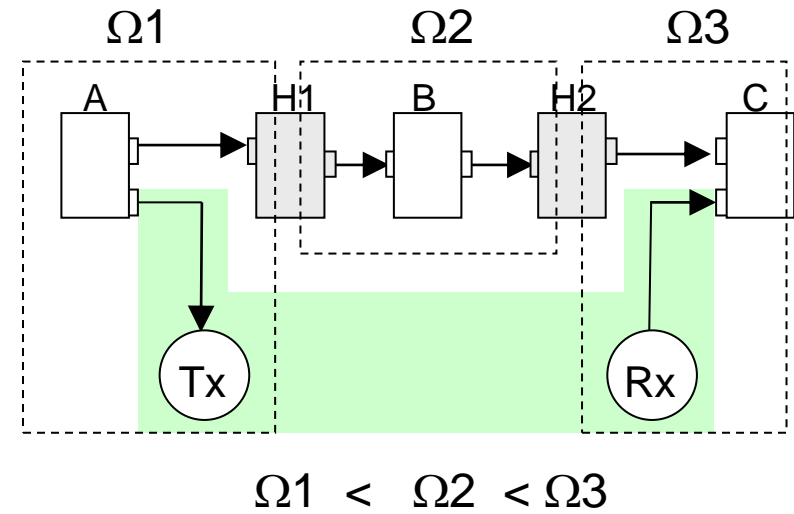
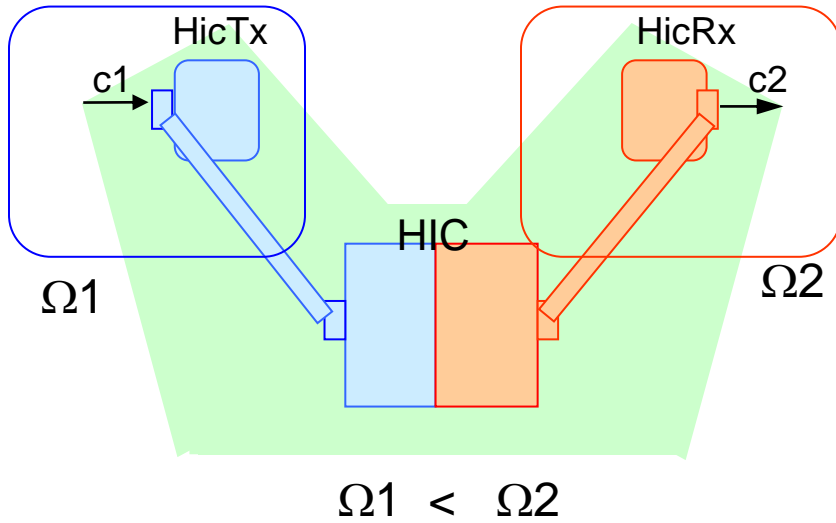
- Tri topologique
- Pour chaque acteur A
 - S'il existe un sous-système S utilisant le MoC de A alors
 - On place A dans S uniquement s'il n'existe pas de chemin entre A et un quelconque acteur de S passant par un HIC
 - Sinon, on crée un nouveau sous-système et on y place A
 - Sinon, on crée un nouveau sous-système et on y place A
- L'absence de chemin passant par un HIC garantit l'absence de dépendances cycliques entre sous-systèmes



HIC/ $\Omega1$ doit être activé après l'activation B2 et
 HIC/ $\Omega2$ doit être activé après l'activation de A1
 D'où l'impossibilité d'ordonnancer $\Omega1$ et $\Omega2$.

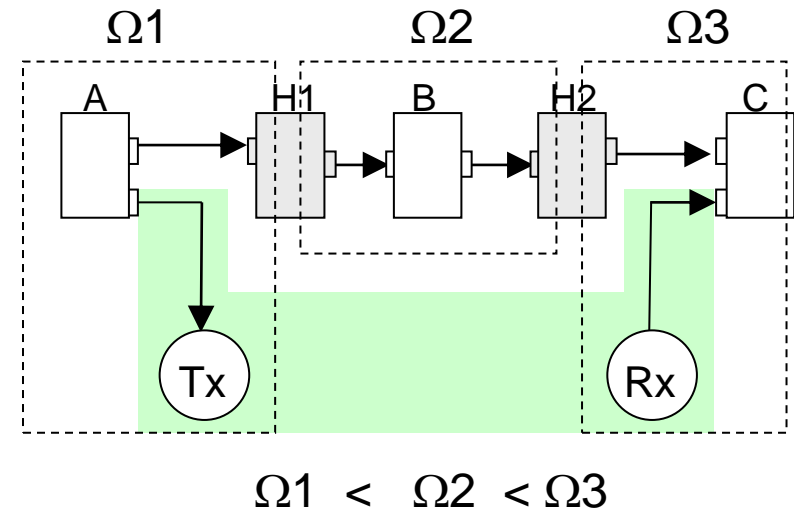
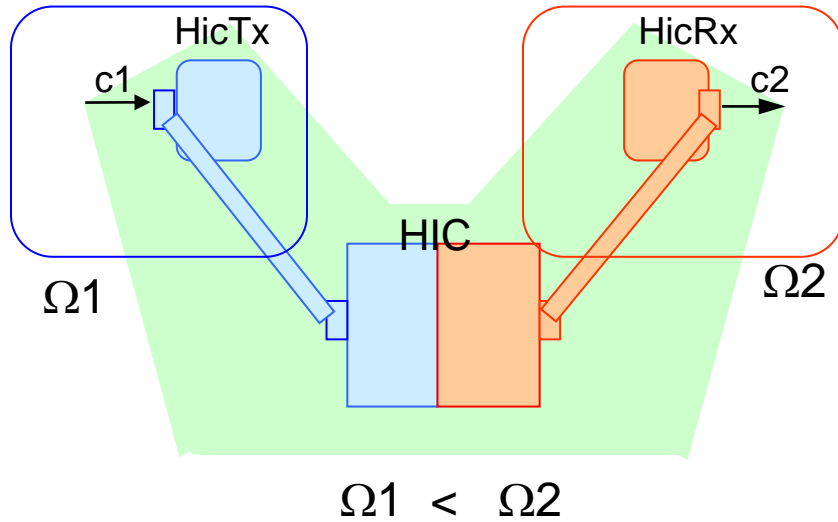
■ Ordonnancement des sous-systèmes

- Basé sur des précédences induites par les HICs.



■ Ordonnancement des sous-systèmes

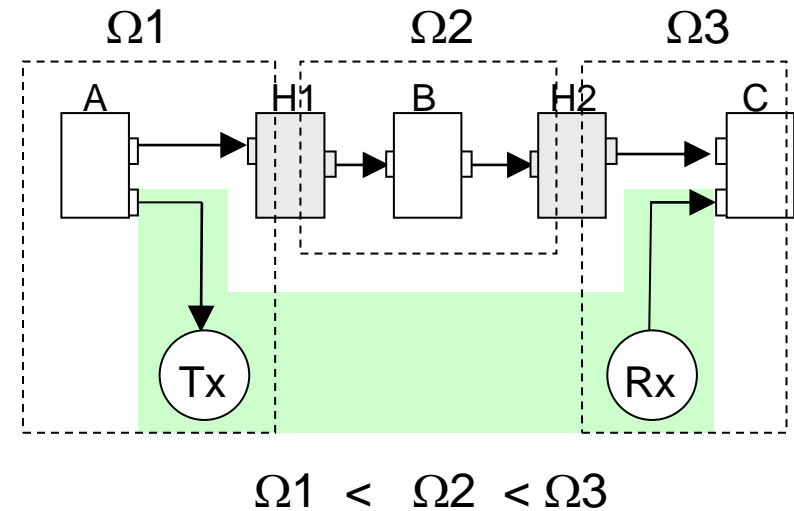
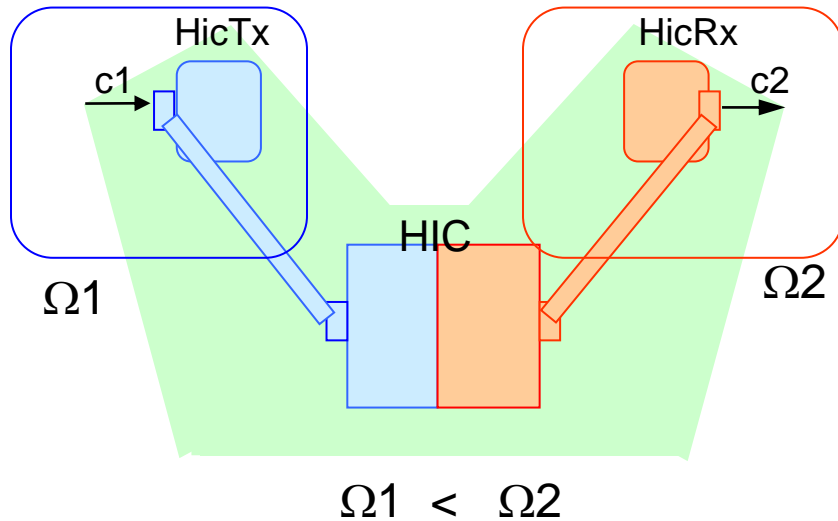
- Basé sur des précédences induites par les HICs.



-Construction du squelette du système partitionné, exclusivement composé des projections des HICs et de leurs dépendances.

■ Ordonnancement des sous-systèmes

- Basé sur des précédences induites par les HICs.

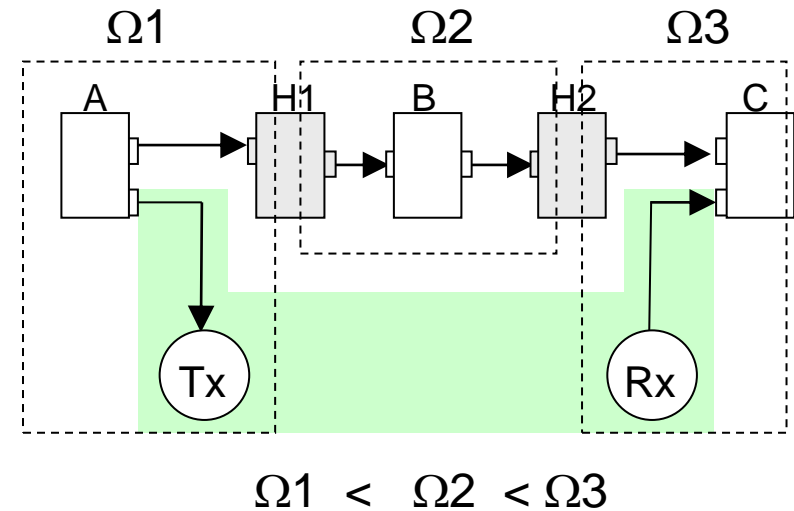
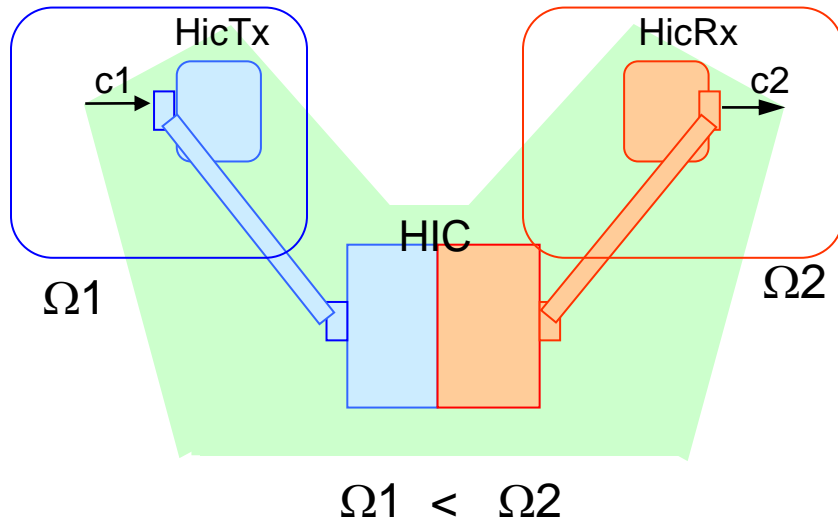


-Construction du squelette du système partitionné, exclusivement composé des projections des HICs et de leurs dépendances.

-Utilisation du tri topologique du squelette pour déterminer les précédences sur les sous-systèmes

■ Ordonnancement des sous-systèmes

- Basé sur des précédences induites par les HICs.

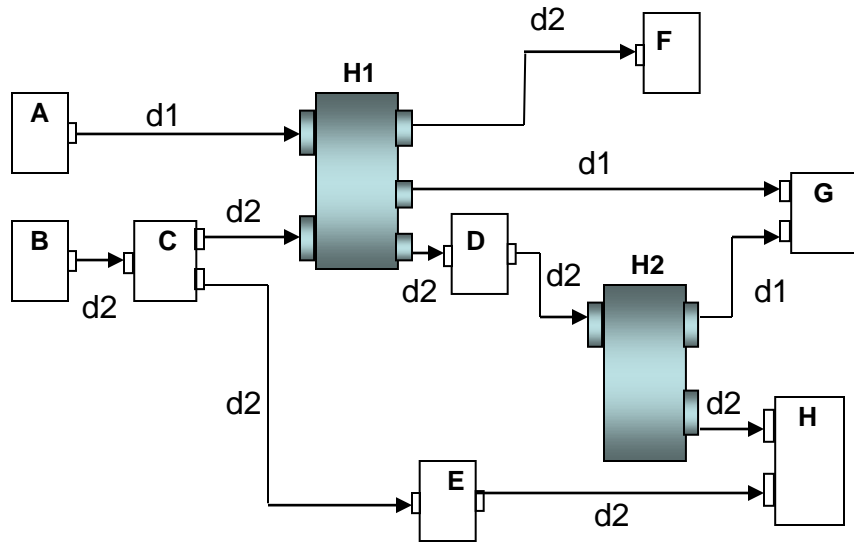


-Construction du squelette du système partitionné, exclusivement composé des projections des HICs et de leurs dépendances.

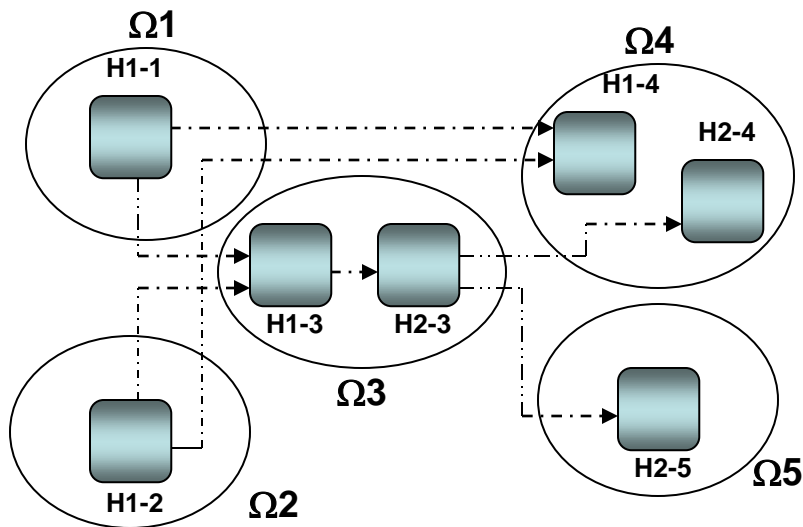
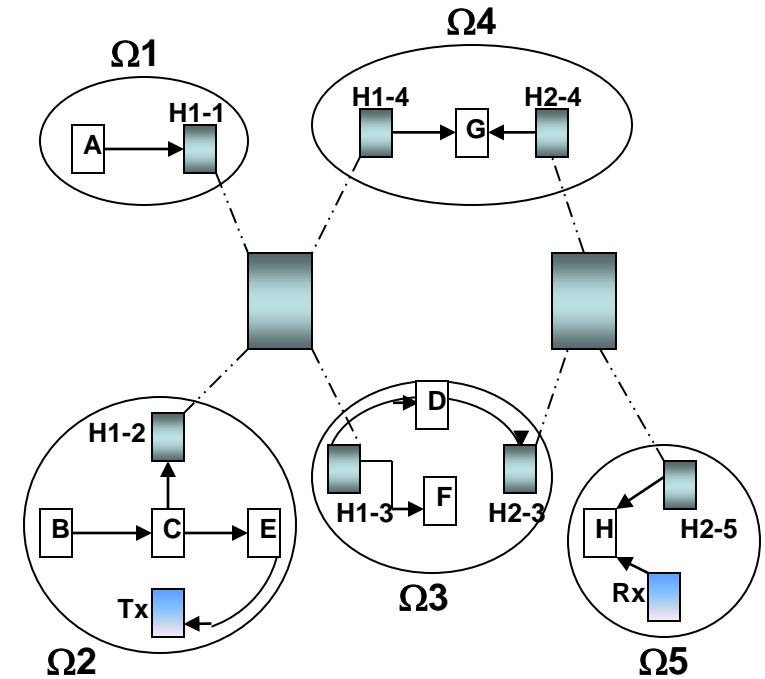
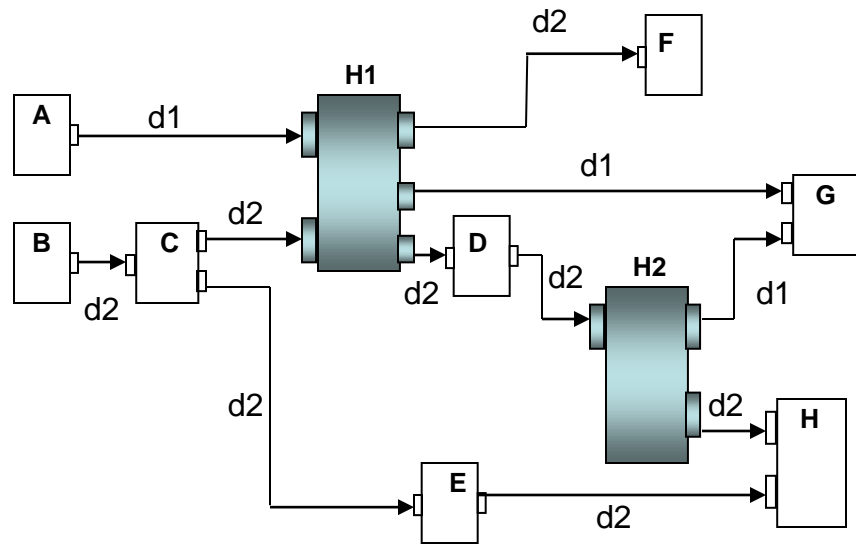
-Utilisation du tri topologique du squelette pour déterminer les précédences sur les sous-systèmes

-Tout ordre compatible avec cette relation de précedence est un ordonnancement possible des sous-systèmes.

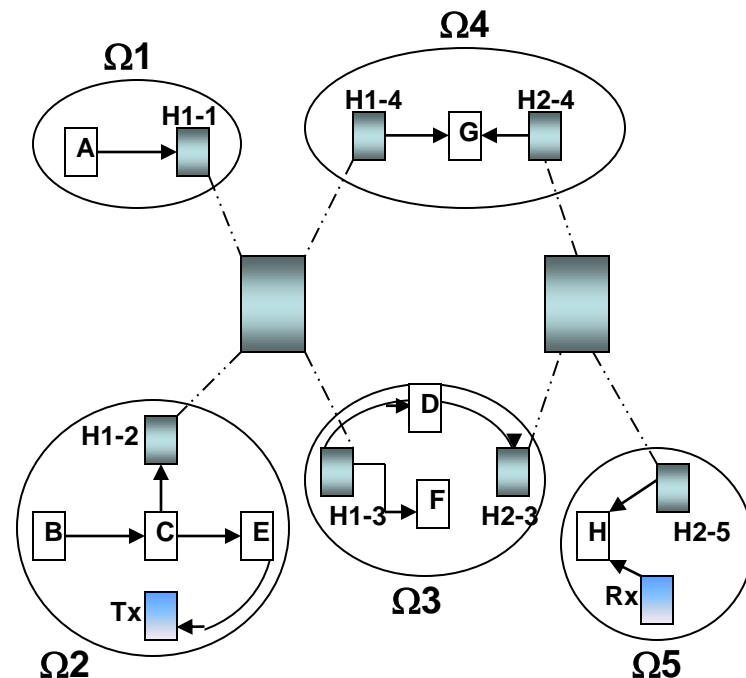
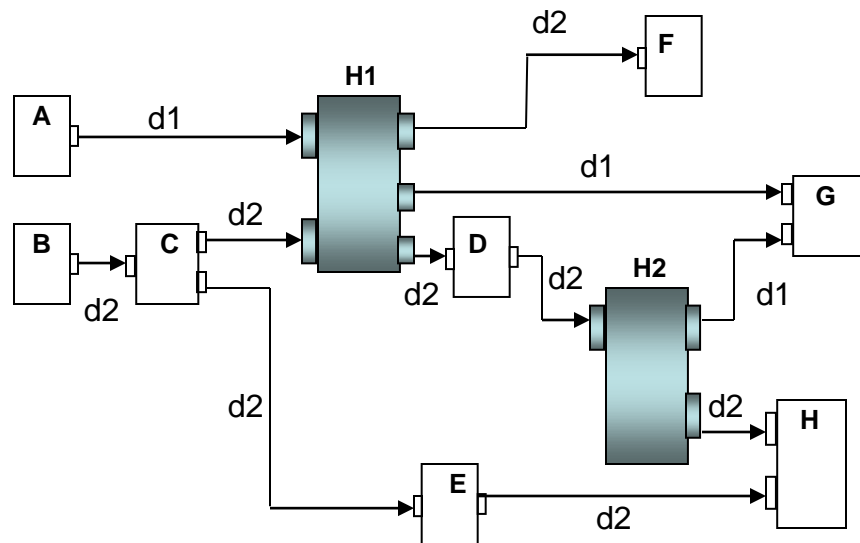
■ Exemple : Partitionnement et ordonnancement d'un système hétérogène



Exemple : Partitionnement et ordonnancement d'un système hétérogène



Exemple : Partitionnement et ordonnancement d'un système hétérogène



$\Omega 1 \triangleleft \Omega 3$

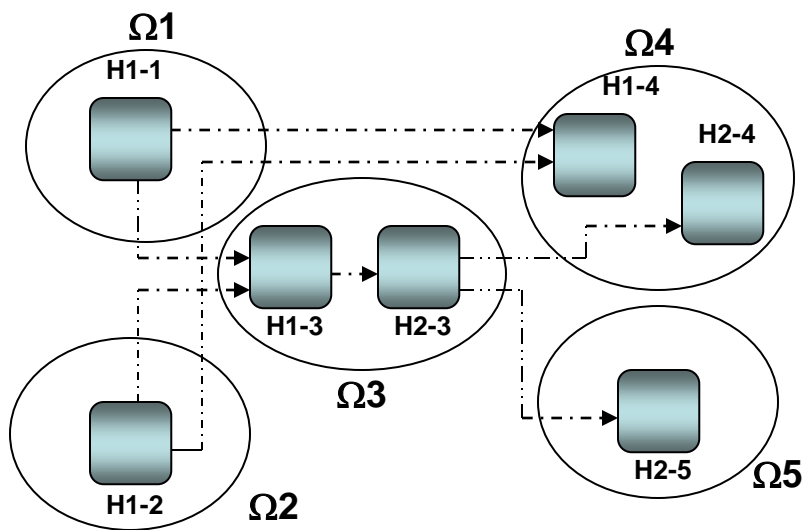
$\Omega 2 \triangleleft \Omega 3$

$\Omega 3 \triangleleft \Omega 4$

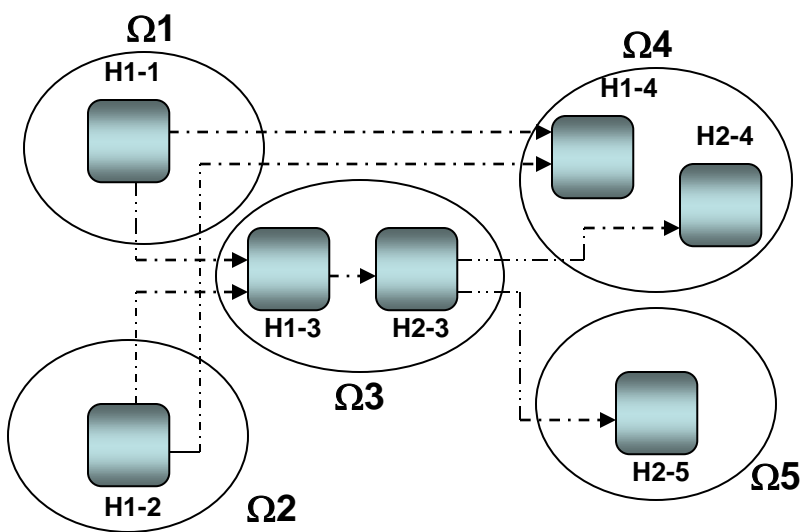
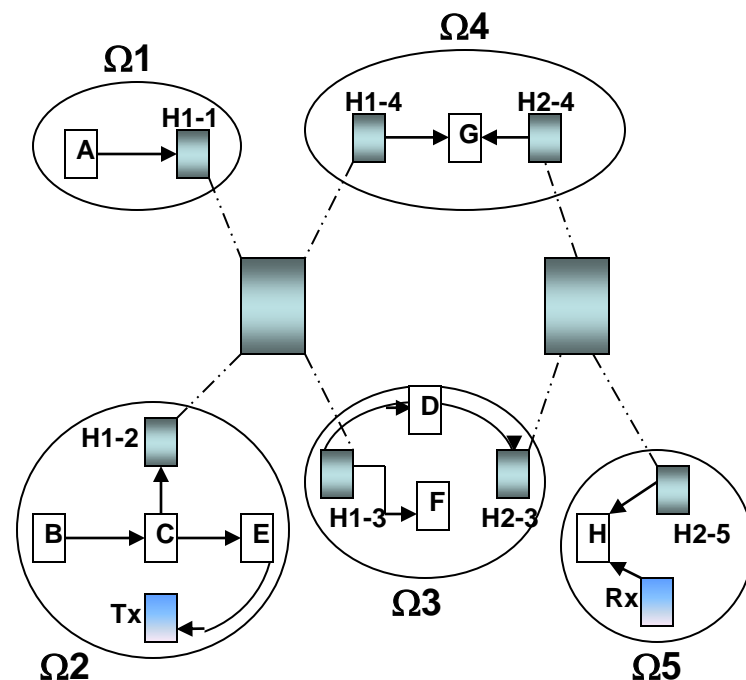
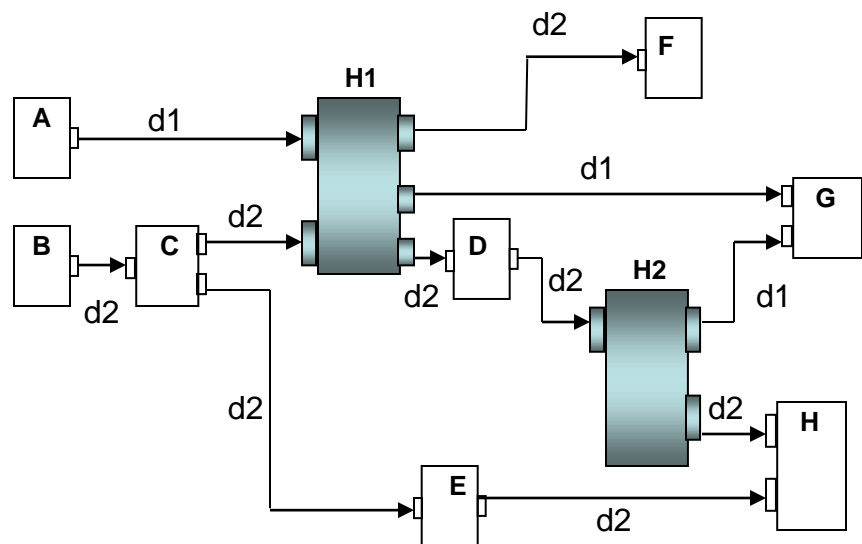
$\Omega 1 \triangleleft \Omega 4$

$\Omega 2 \triangleleft \Omega 4$

$\Omega 3 \triangleleft \Omega 5$



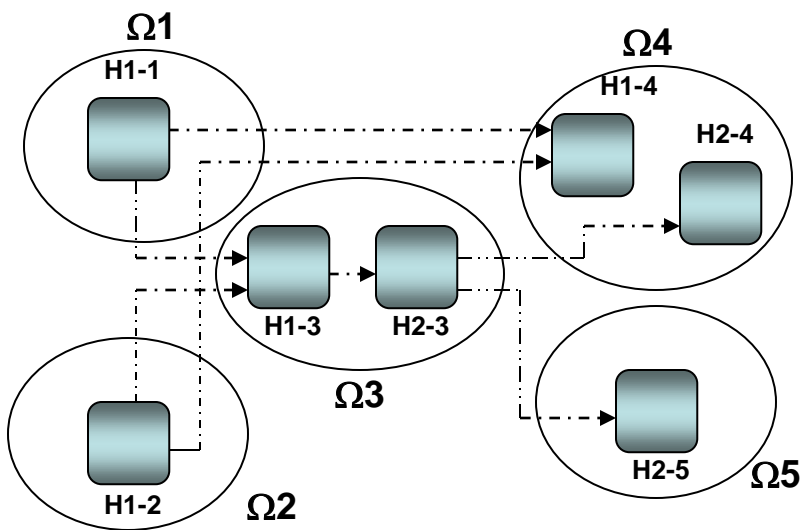
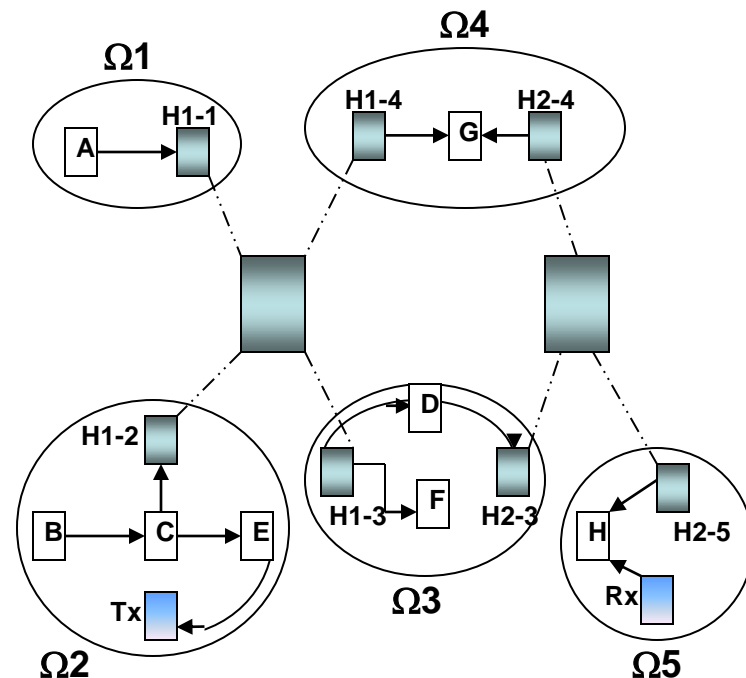
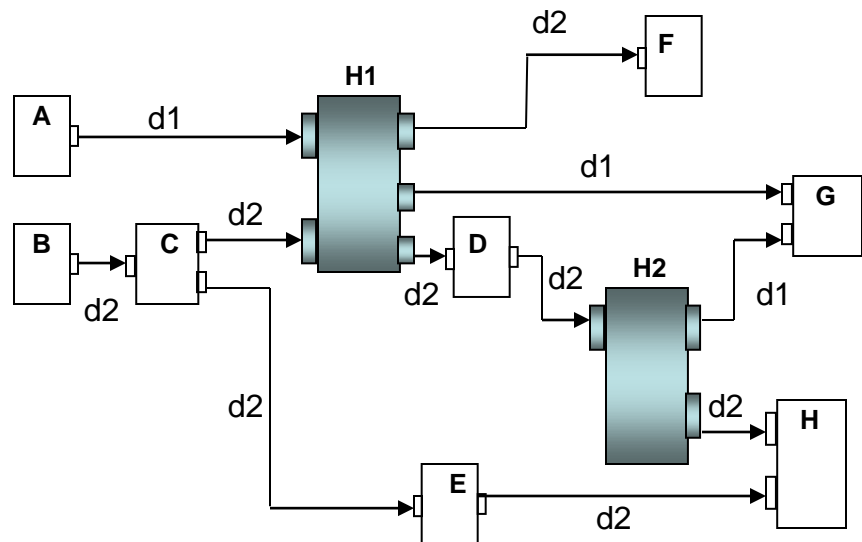
Exemple : Partitionnement et ordonnancement d'un système hétérogène



$\Omega 1 \triangleleft \Omega 3$ $\Omega 2 \triangleleft \Omega 3$ $\Omega 3 \triangleleft \Omega 4$
 $\Omega 1 \triangleleft \Omega 4$ $\Omega 2 \triangleleft \Omega 4$ $\Omega 3 \triangleleft \Omega 5$

$\Omega 1$	$\Omega 2$	$\Omega 3$	$\Omega 4$	$\Omega 5$
$\Omega 1$	$\Omega 2$	$\Omega 3$	$\Omega 5$	$\Omega 4$
$\Omega 2$	$\Omega 1$	$\Omega 3$	$\Omega 4$	$\Omega 5$
$\Omega 2$	$\Omega 1$	$\Omega 3$	$\Omega 5$	$\Omega 4$

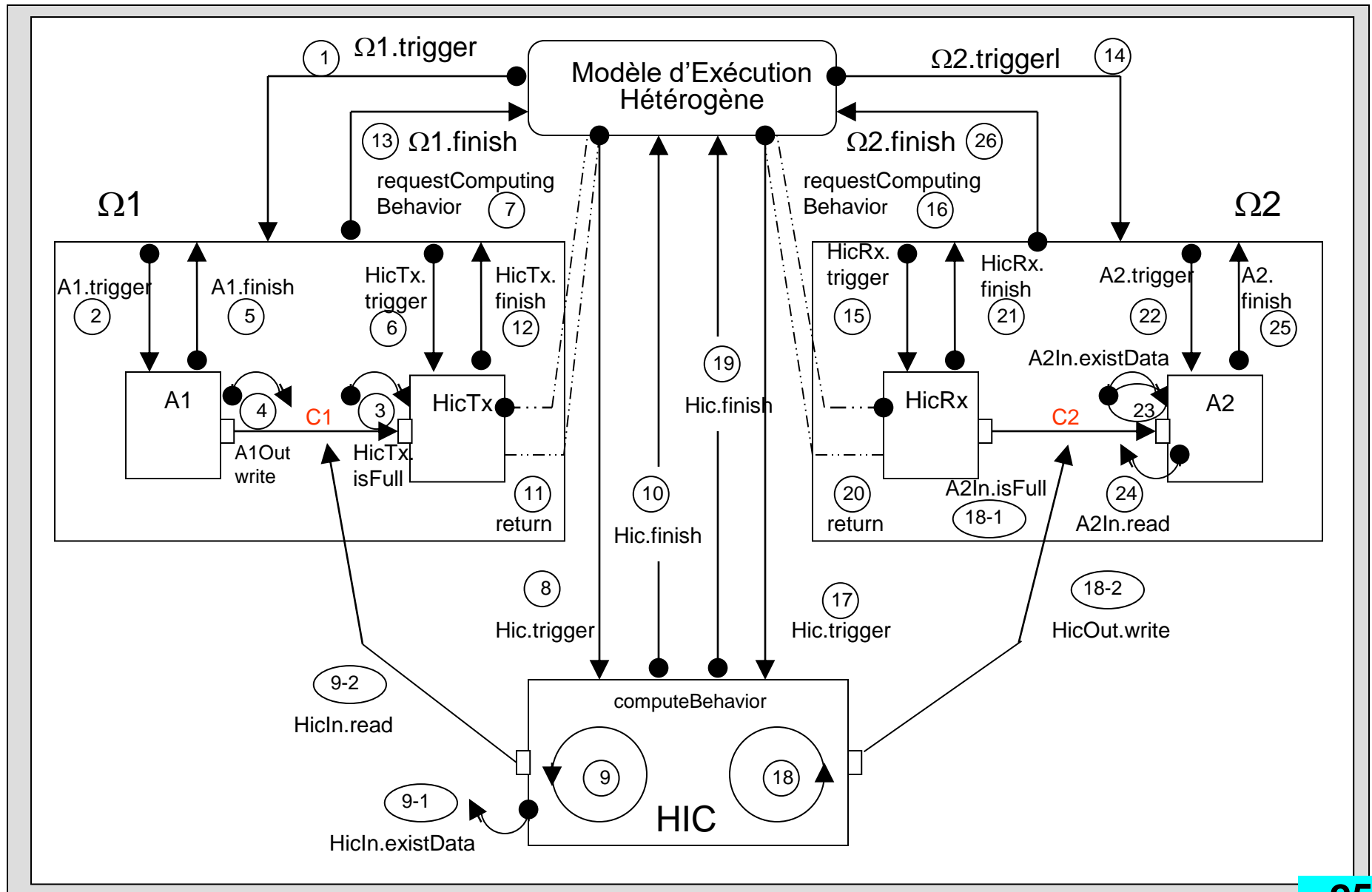
Exemple : Partitionnement et ordonnancement d'un système hétérogène



$\Omega 1 \triangleleft \Omega 3$ $\Omega 2 \triangleleft \Omega 3$ $\Omega 3 \triangleleft \Omega 4$
 $\Omega 1 \triangleleft \Omega 4$ $\Omega 2 \triangleleft \Omega 4$ $\Omega 3 \triangleleft \Omega 5$

$\Omega 1$	$\Omega 2$	$\Omega 3$	$\Omega 4$	$\Omega 5$
$\Omega 1$	$\Omega 2$	$\Omega 3$	$\Omega 5$	$\Omega 4$
$\Omega 2$	$\Omega 1$	$\Omega 3$	$\Omega 4$	$\Omega 5$
$\Omega 2$	$\Omega 1$	$\Omega 3$	$\Omega 5$	$\Omega 4$

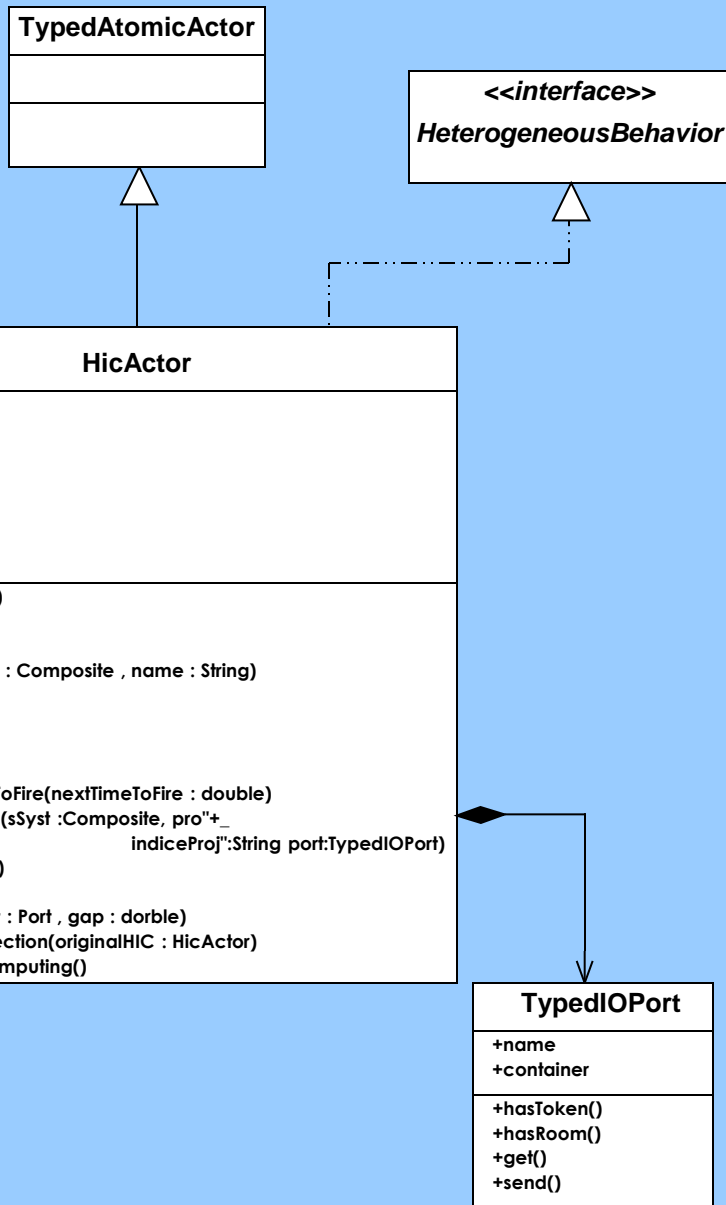
■ Exécution



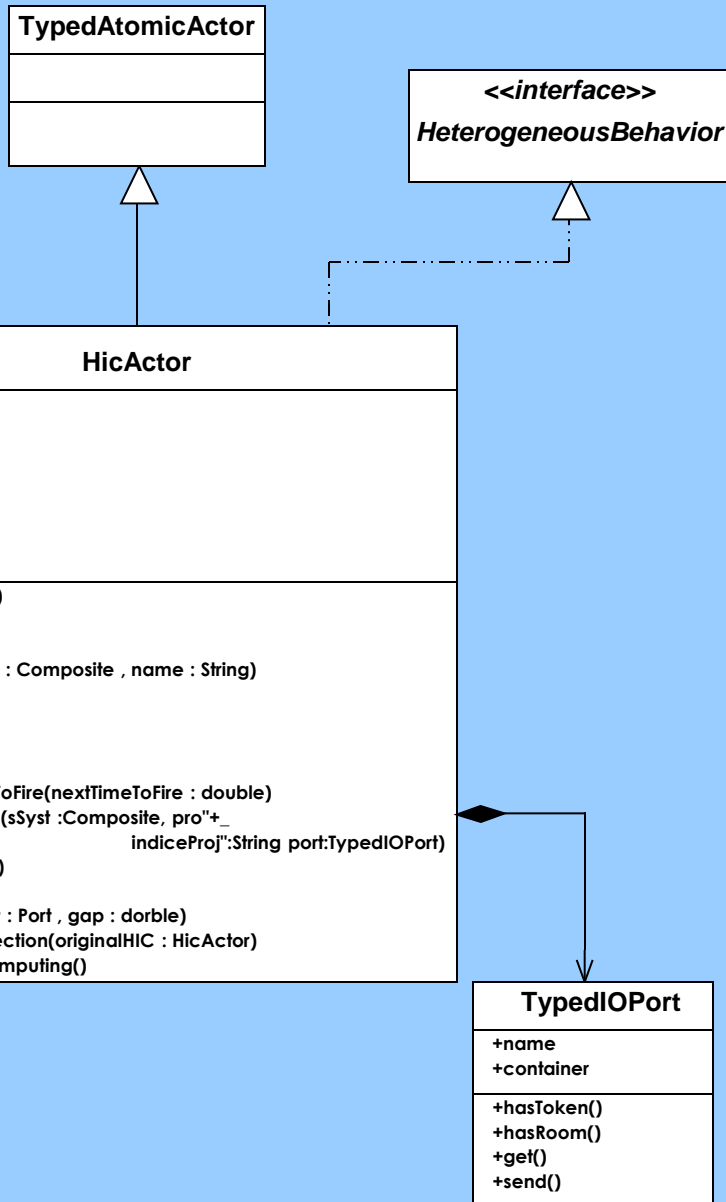
Plan

- **Systèmes Embarqués**
 - Positionnement du marché
 - Caractéristiques
 - Outils de modélisation et Modélisation hiérarchique
- **Problématique et Objectif**
- **Approche hétérogène non-hiérarchique**
- **Modélisation :**
 - Composant à Interface Hétérogène (HIC)
 - Modèle d'Exécution Hétérogène Non-Hiérarchique
- **Intégration et Validation par Simulation dans Ptolemy II**
- **Conclusion et perspectives**

Classe HicActor



Classe HicActor



Classe HDirector

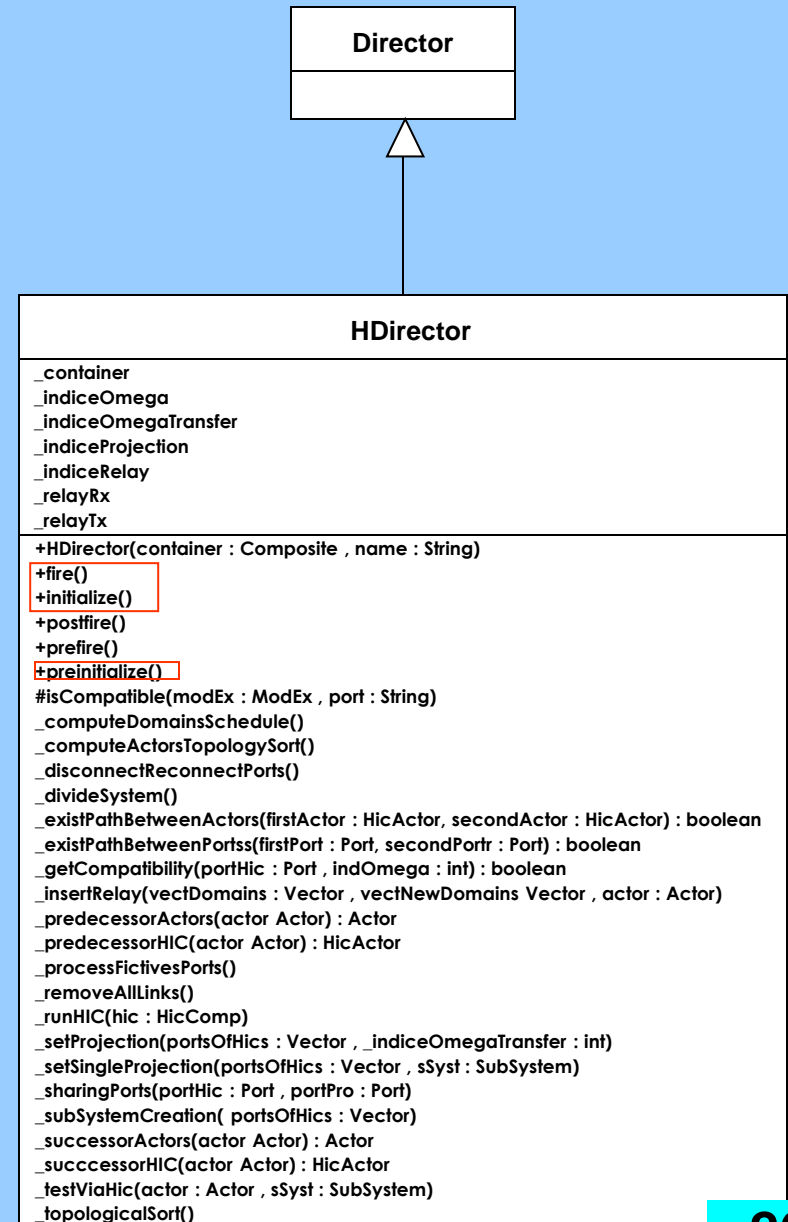


Schéma d'intégration

**Modèle d'exécution
hétérogène non hiérarchique**

HDIRECTOR

Schéma d'intégration

**Modèle d'exécution
hétérogène non hiérarchique**

Partitionnement

- génération des SS
- placement des acteurs
- déconnexion et reconnection
- création des ports virtuels

MANAGER



HDIRECTOR

Preinitialisation

- `super.preinitilize()`
- invoque même méthode sur DL
(création receivers dans projections)

Schéma d'intégration

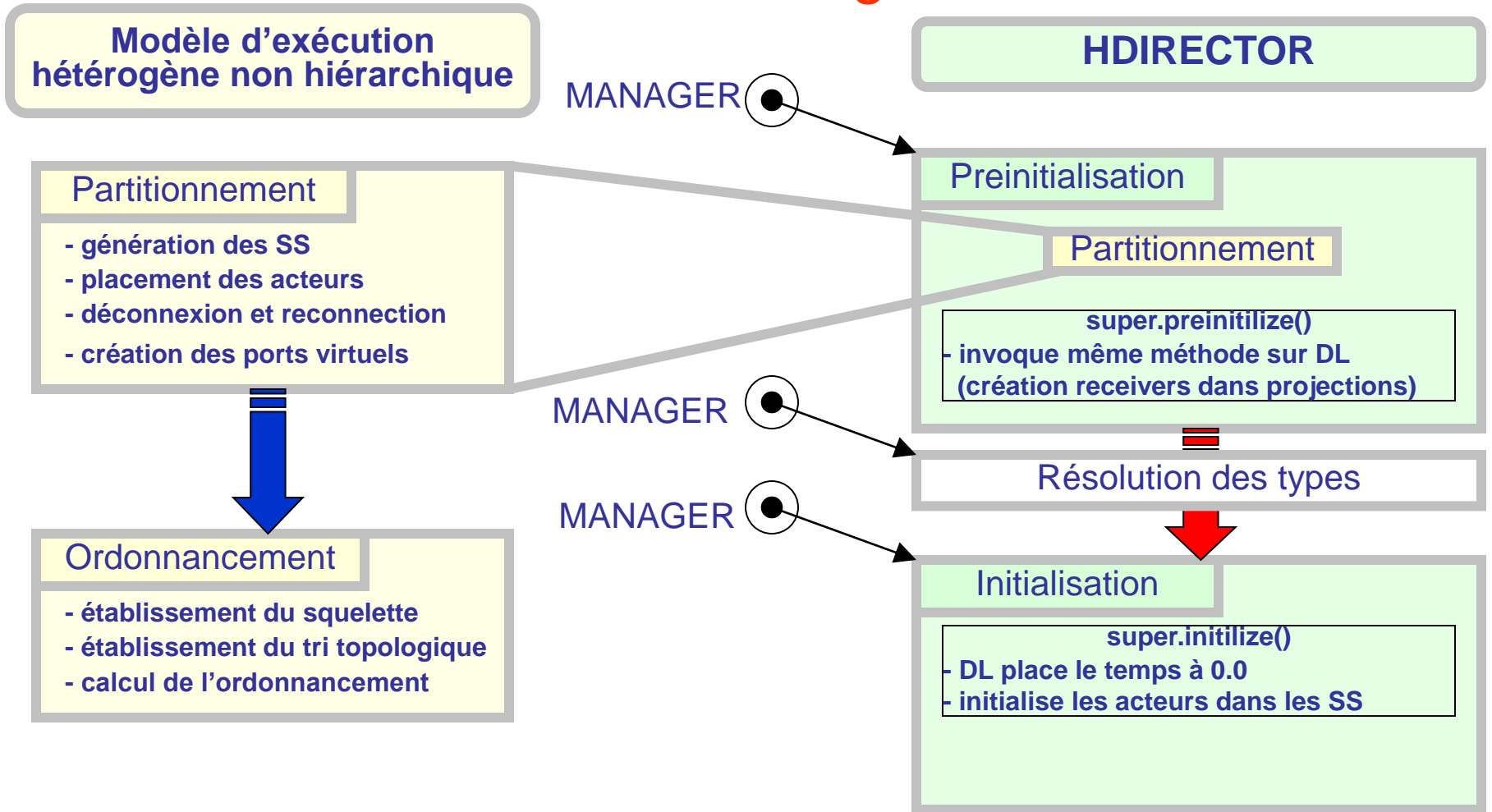


Schéma d'intégration

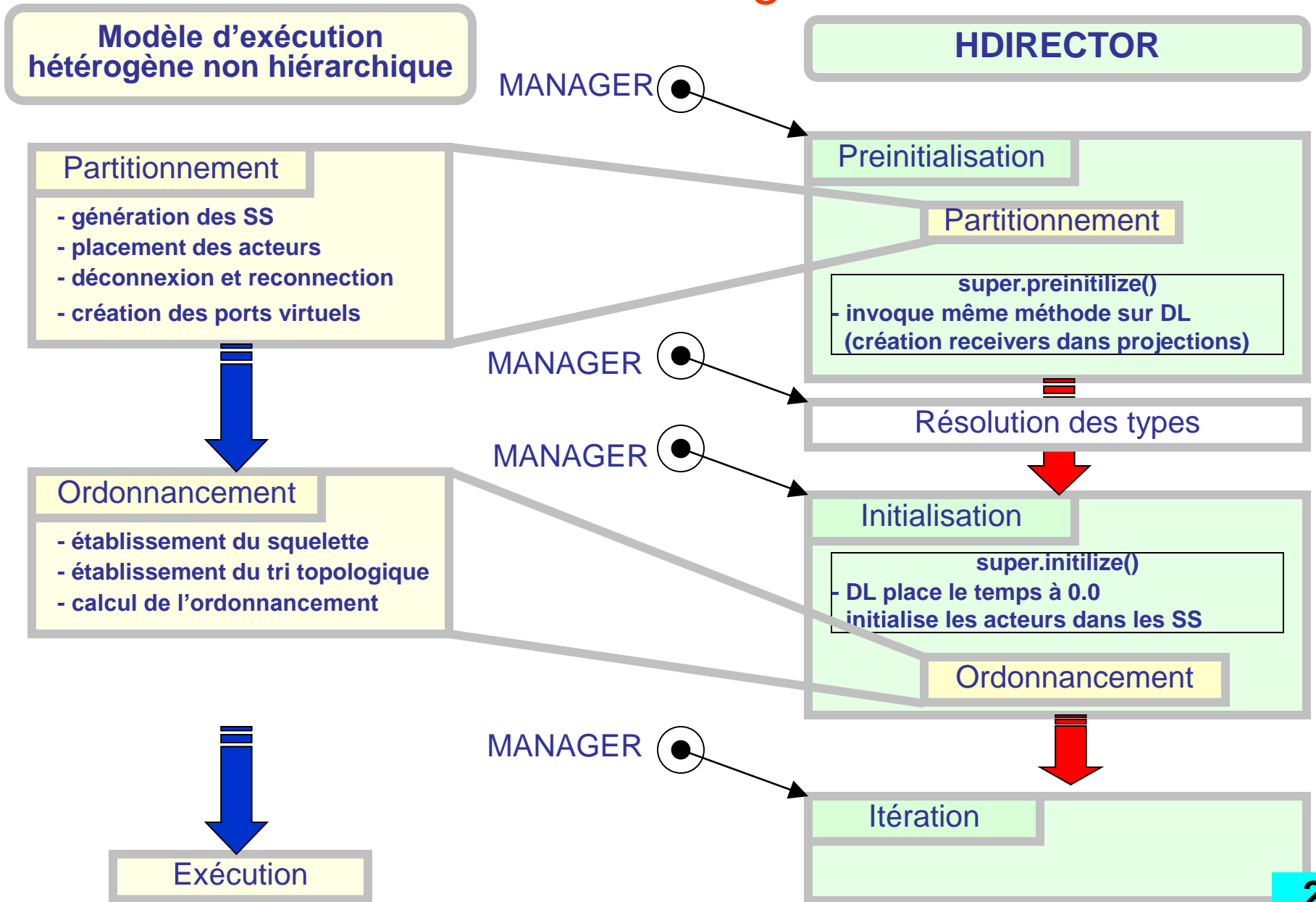


Schéma d'intégration

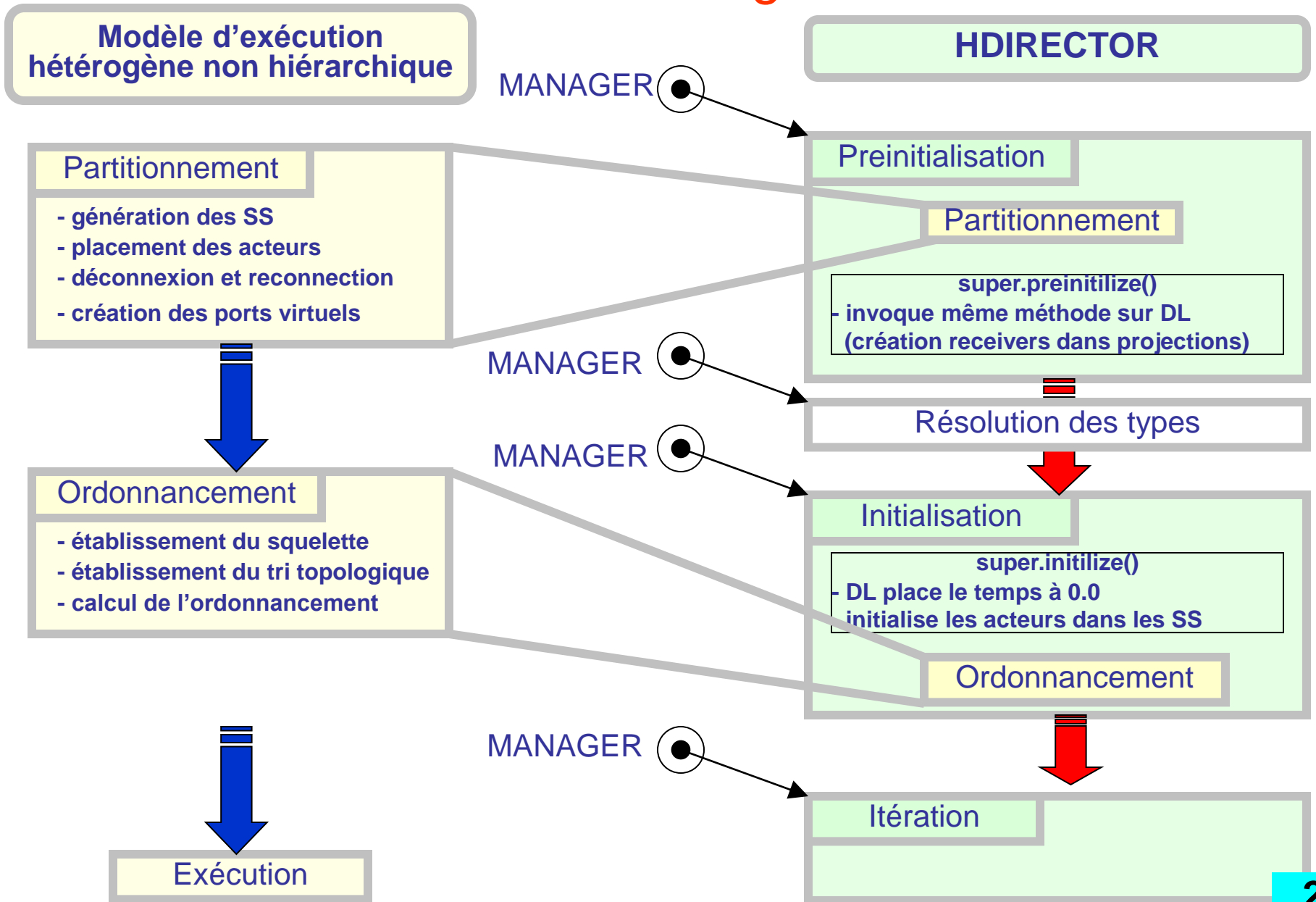


Schéma d'intégration

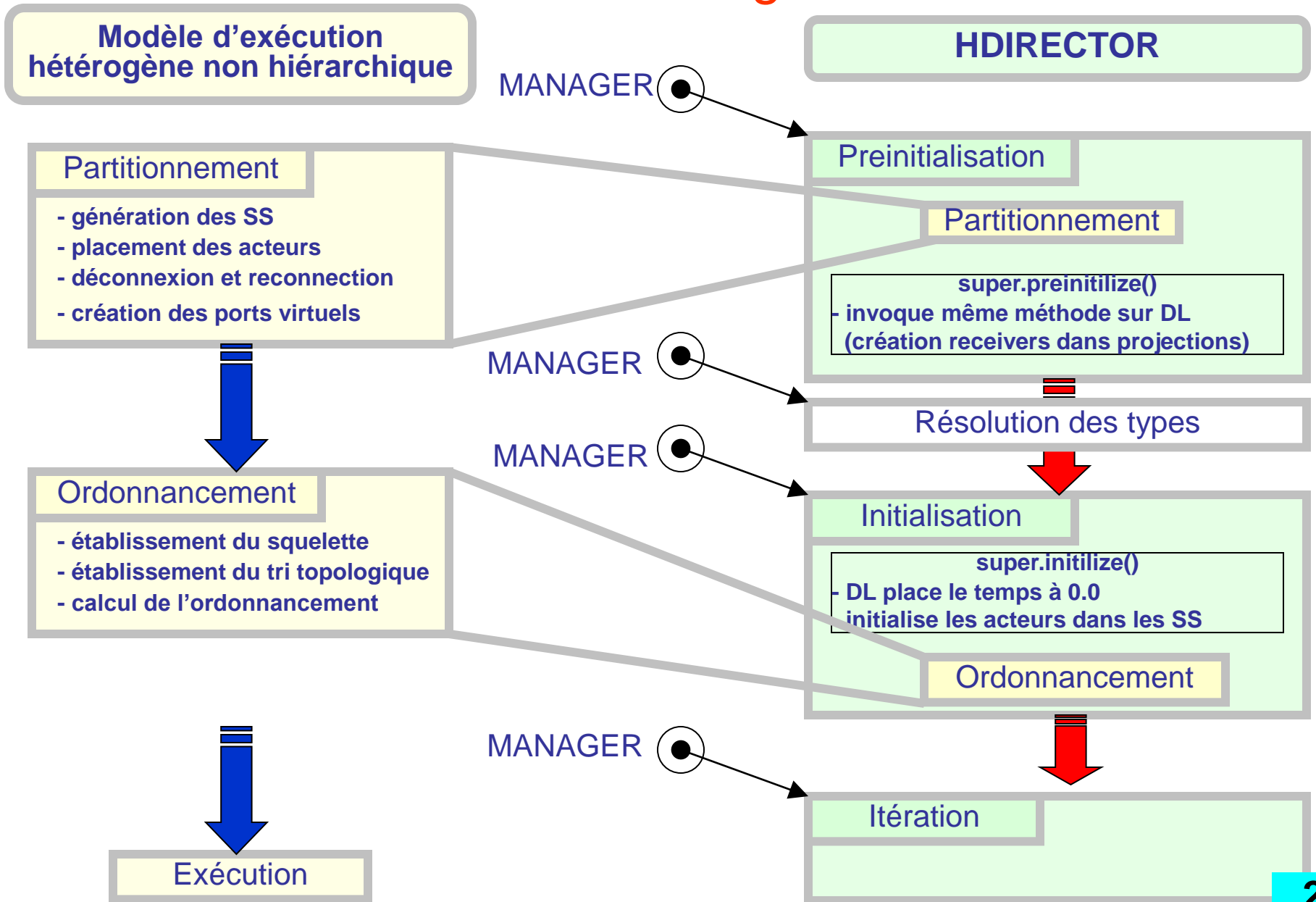


Schéma d'intégration

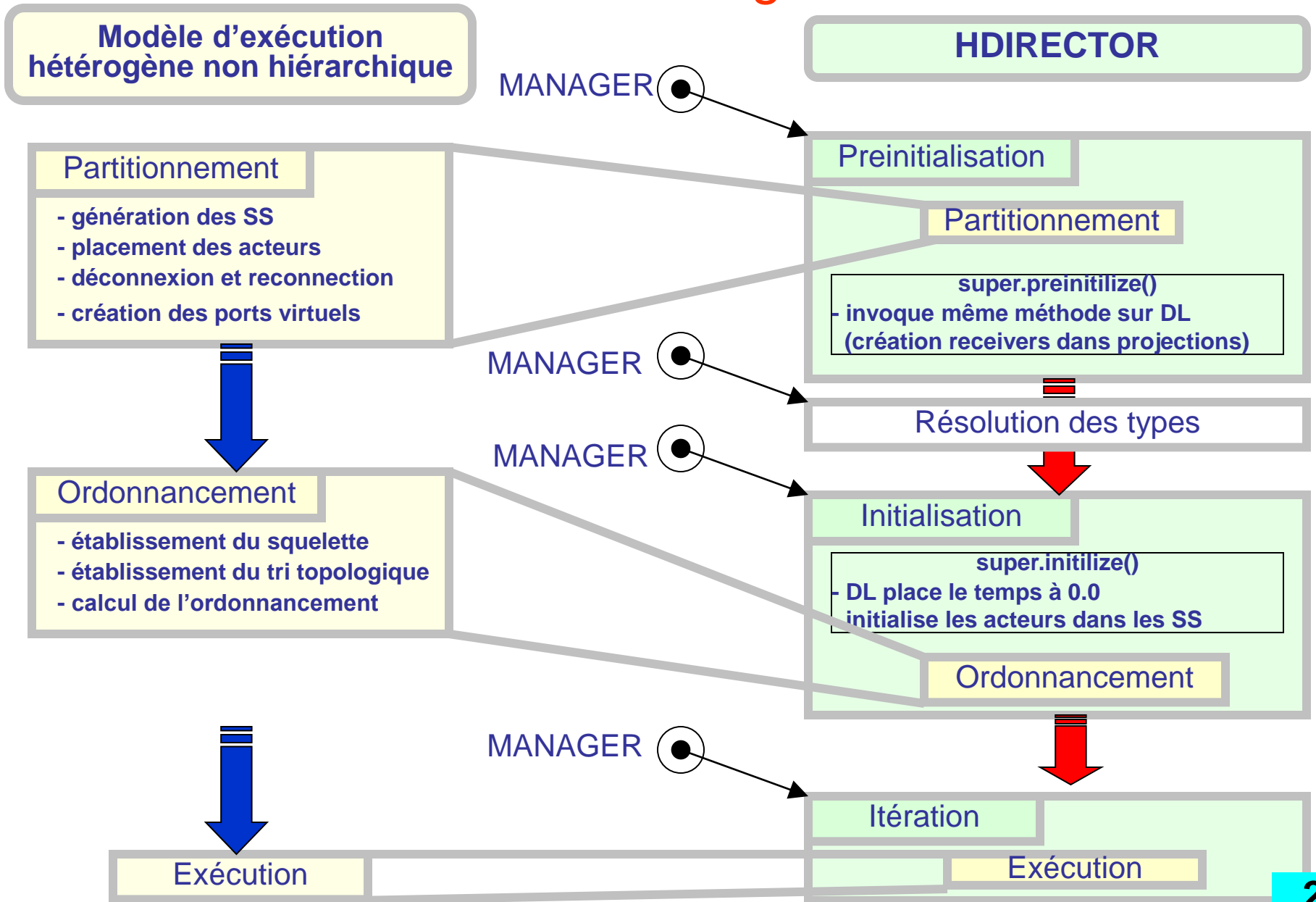
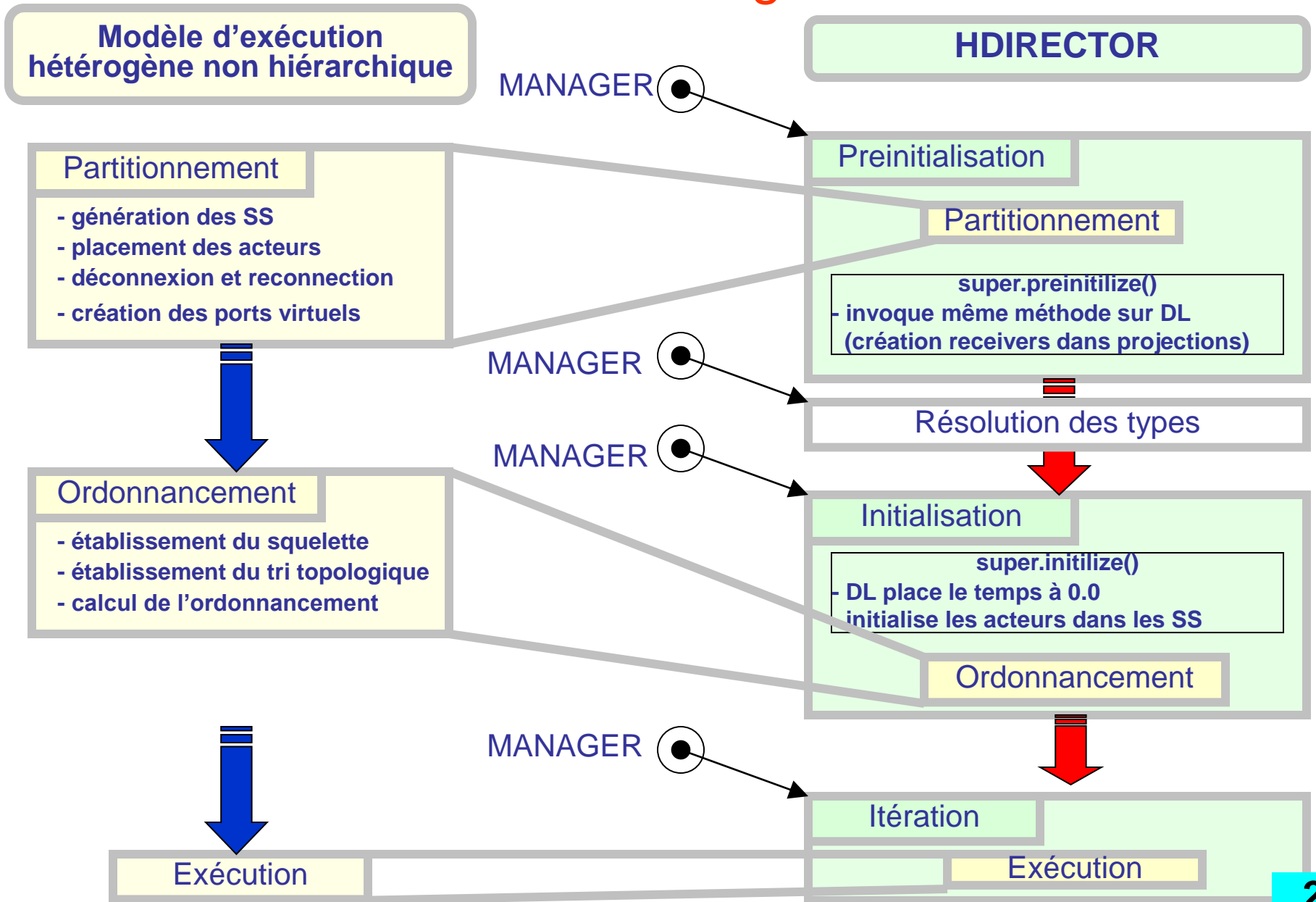
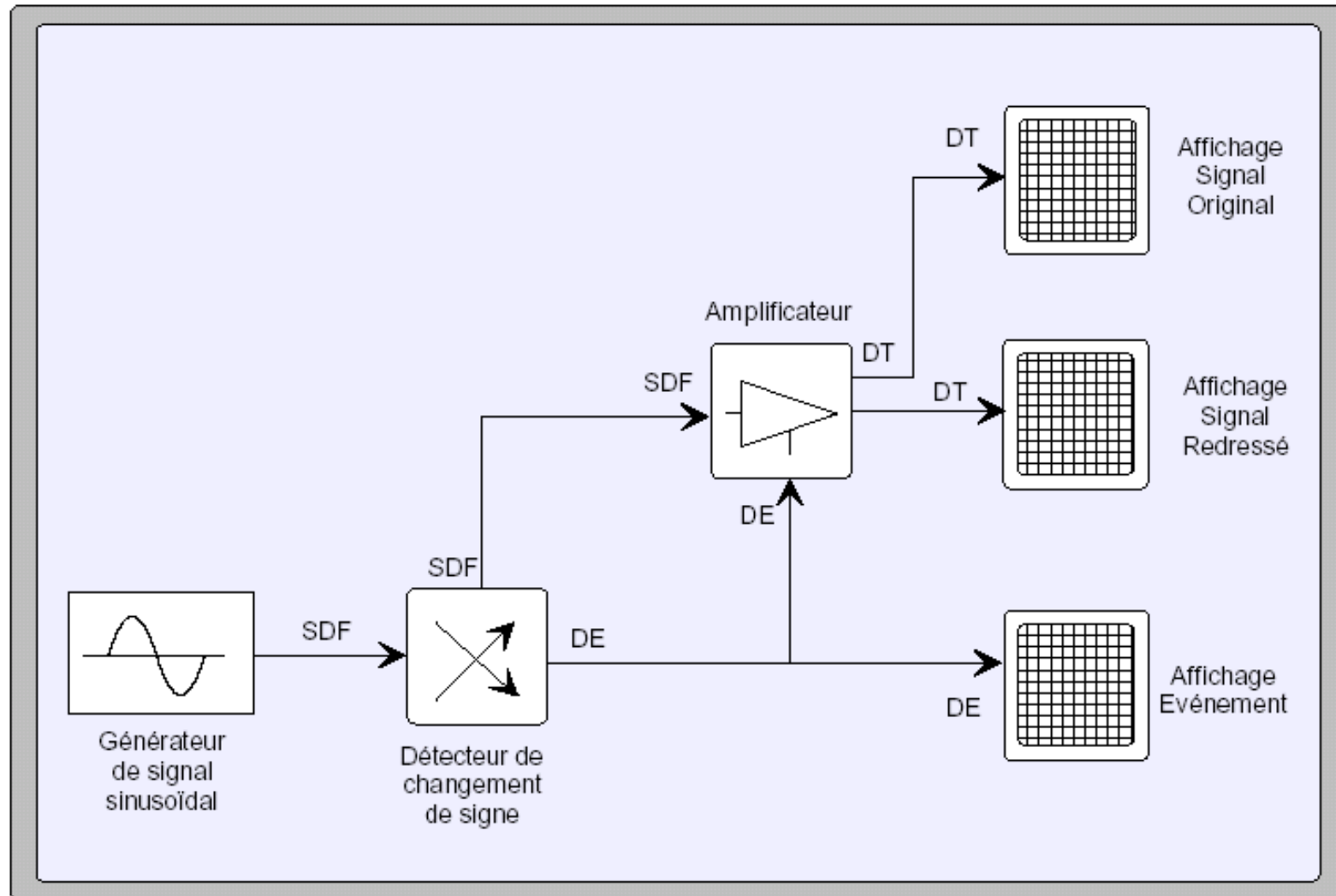


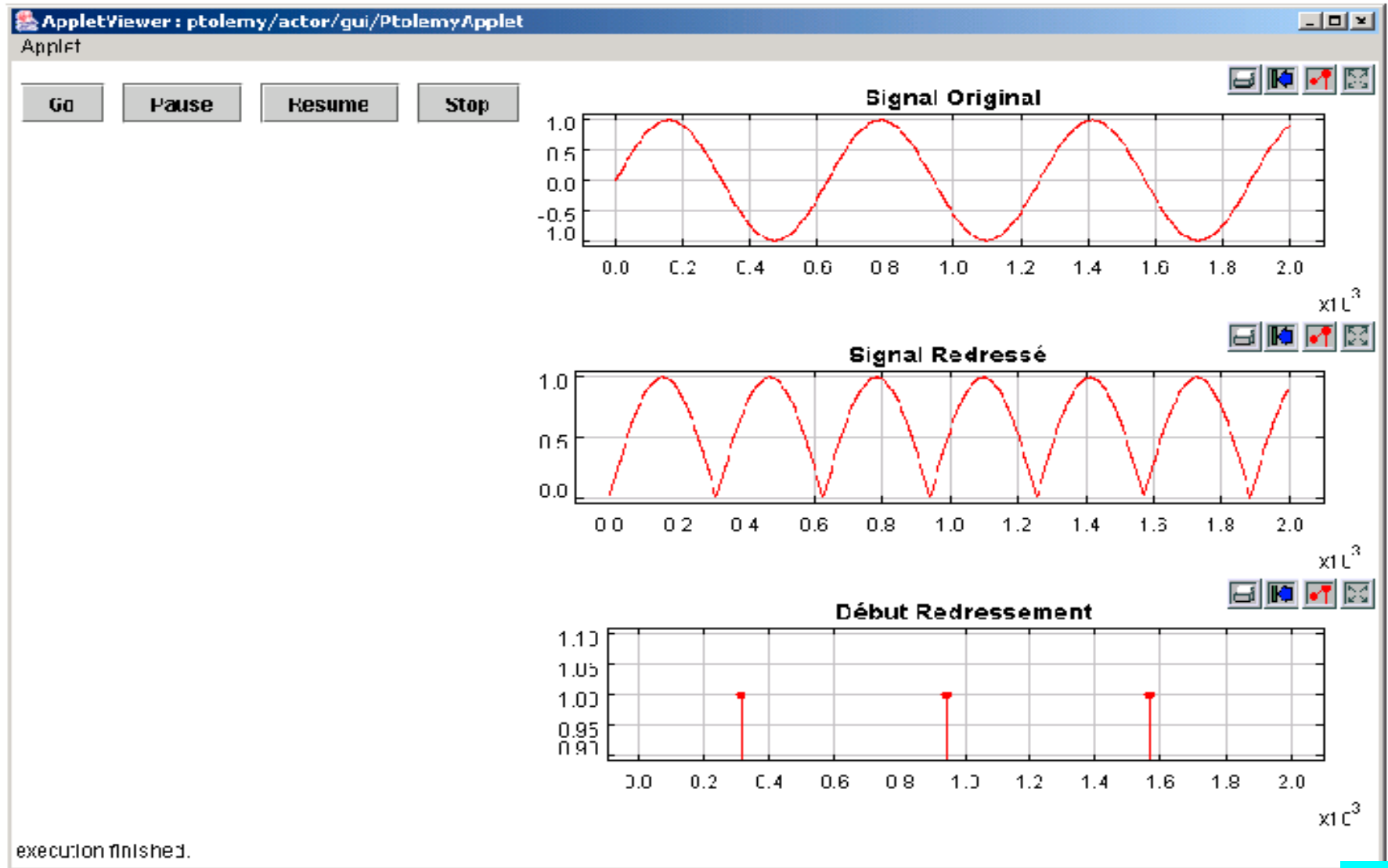
Schéma d'intégration



Simulation 1 : Redresseur de Signal



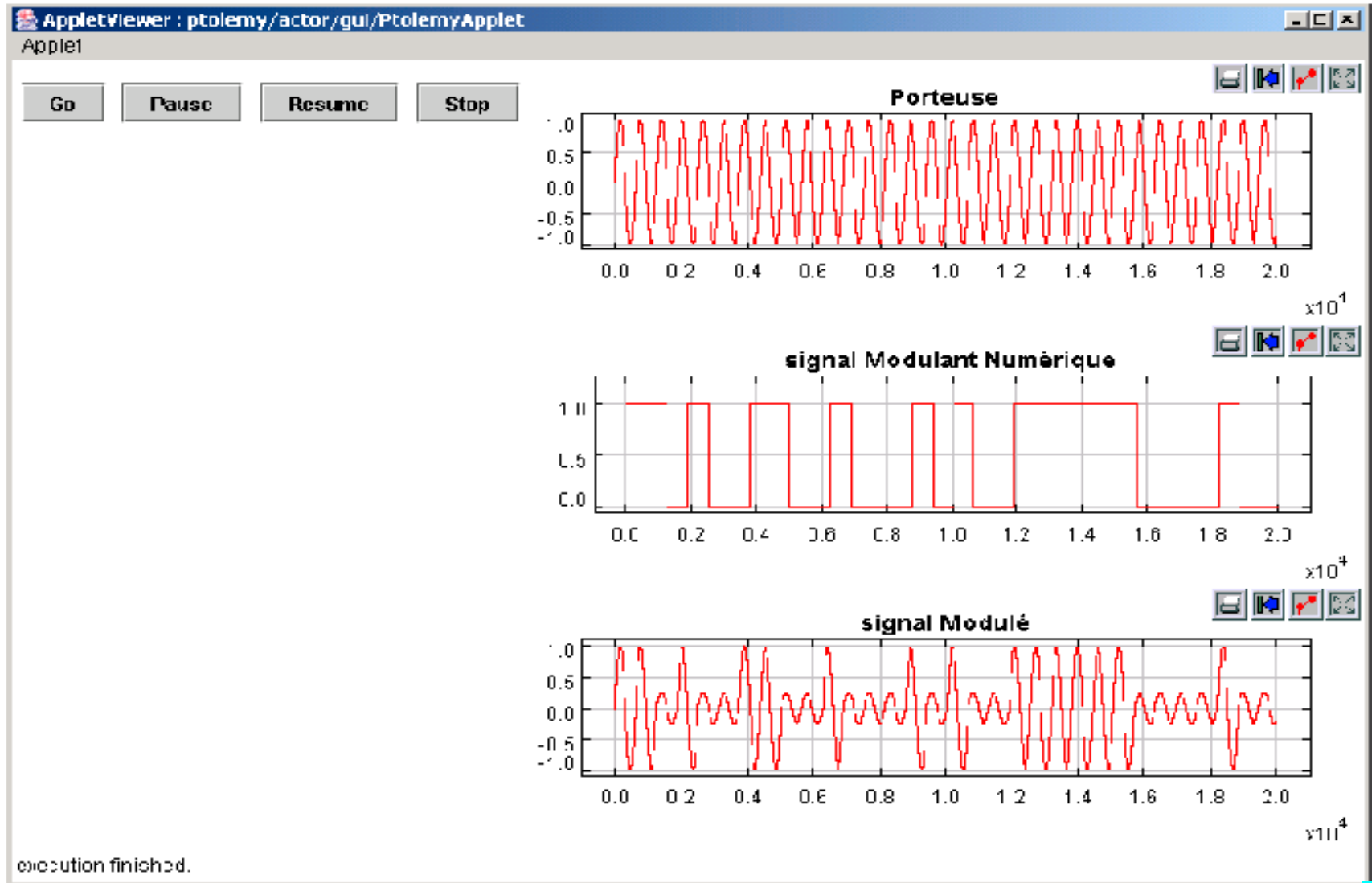
Résultat de la simulation 1



Simulation 2 : Modulation d'Amplitude



Résultat de la simulation 2



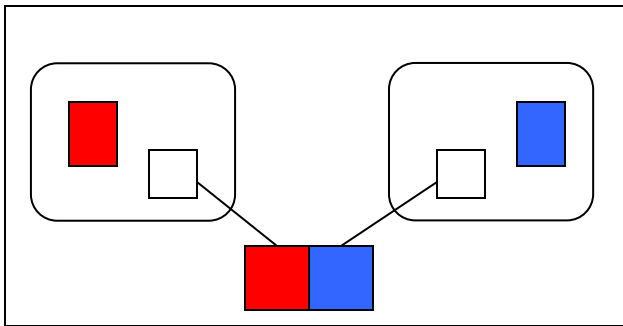
Plan

- **Systèmes Embarqués**
 - Positionnement du marché
 - Caractéristiques
 - Outils de modélisation et Modélisation hiérarchique
- **Problématique et Objectif**
- **Approche hétérogène non-hiérarchique**
- **Modélisation :**
 - Composant à Interface Hétérogène (HIC)
 - Modèle d'Exécution Hétérogène Non-Hiérarchique
- **Intégration et Validation par Simulation dans Ptolemy II**
- **Conclusion et perspectives**

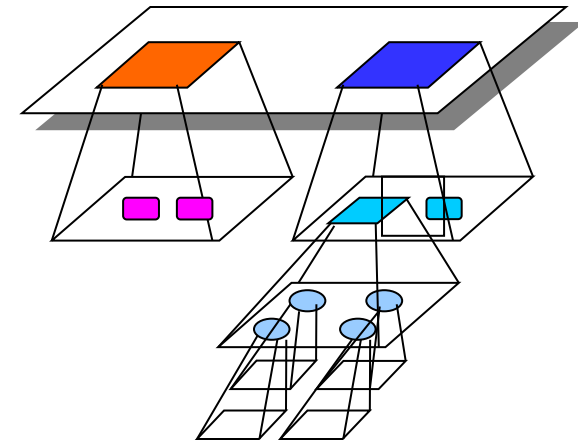
Conclusion

- Possibilité d'utiliser des Composants à Interface Hétérogène dans le cadre d'un ensemble ouvert de modèles de calcul
- Transformation d'un modèle plat en un modèle hiérarchique et exécution par un modèle de calcul générique
- Conservation de la hiérarchie pour structurer le système et gérer sa complexité

Conception



Réalisation

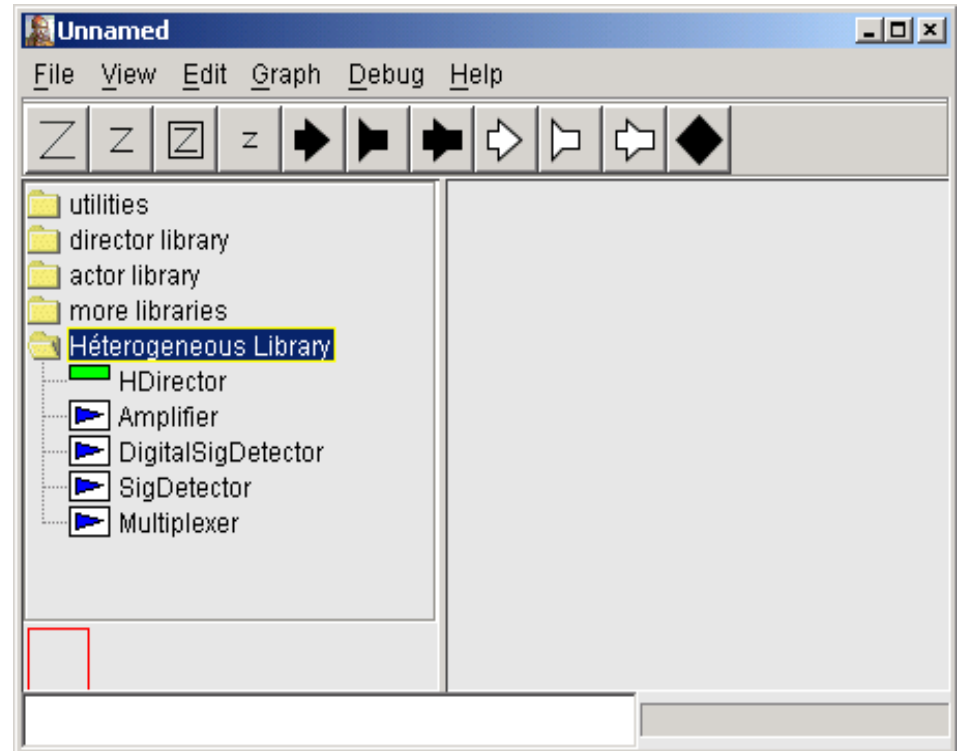


- Changer de modèle sans changer de niveau hiérarchique
- Concevoir explicitement le comportement hétérogène comme faisant partie du système
- Les boucles ne sont pas supportées au niveau hétérogène

Perspectives

■ Court terme

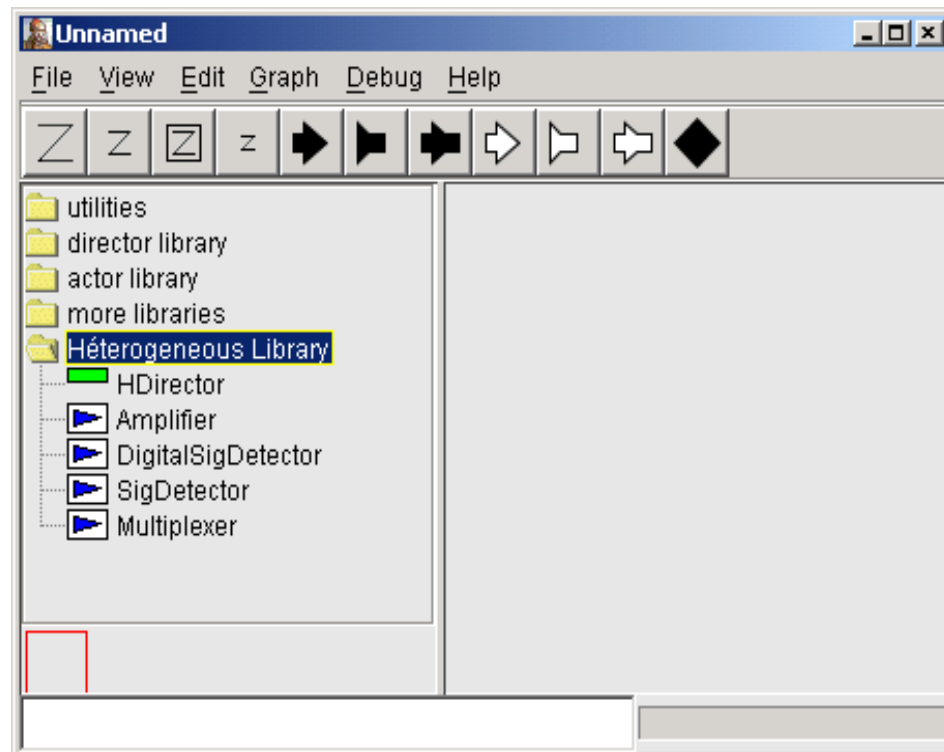
- Faciliter la construction des modèles en utilisant Vergil, l'interface graphique de Ptolemy II



Perspectives

■ Court terme

- Faciliter la construction des modèles en utilisant Vergil, l'interface graphique de Ptolemy II



■ Long terme

- Définir des méthodes de conception de HIC.
- Supporter les boucles au niveau hétérogène

Questions ?