



**1<sup>ère</sup> Année Master Sciences et Technologies de l'information et de la Communication (M1STIC)**

**TP Algorithmes distribués (ALDI)**

**TP 01 : Initiation à la plateforme JADE  
(Création et lancement des agents)**

**Année universitaire : 2023/2024**

# Compétences à acquérir

Suite à ce TP, vous serez en mesure de :

- Se familiariser avec la plateforme JADE.
- Comprendre le principe de fonctionnement d'un système multi agents.
- Installer la plateforme JADE.
- Créer un agent et lancer son exécution pour afficher son identifiant.
- Créer un agent qui reçoit des paramètres passés en arguments puis les afficher.

# Plan

- 1. Introduction à la plateforme JADE**
- 2. Système multi agents**
- 3. Installation de la plate forme JADE**
- 4. Création du premier agent avec JADE sous JAVA**

# Introduction

- ✓ JADE (Java Agent Development Framework) est un environnement de développement d'agent implémenté totalement dans le langage JAVA.
- ✓ Il facilite la mise en place d'un système multi-agent (SMA) répondant aux spécifications FIPA (Foundation for Intelligent Physical Agents) à travers un ensemble d'outils.
- ✓ La plateforme JADE inclut des composants qui contrôlent un système multi-agent :

**Un runtime Environment** : l'environnement où les agents peuvent vivre, il doit être activé pour pouvoir lancer les agents.

**Une librairie de classes** : que les développeurs utilisent pour écrire leurs programmes.

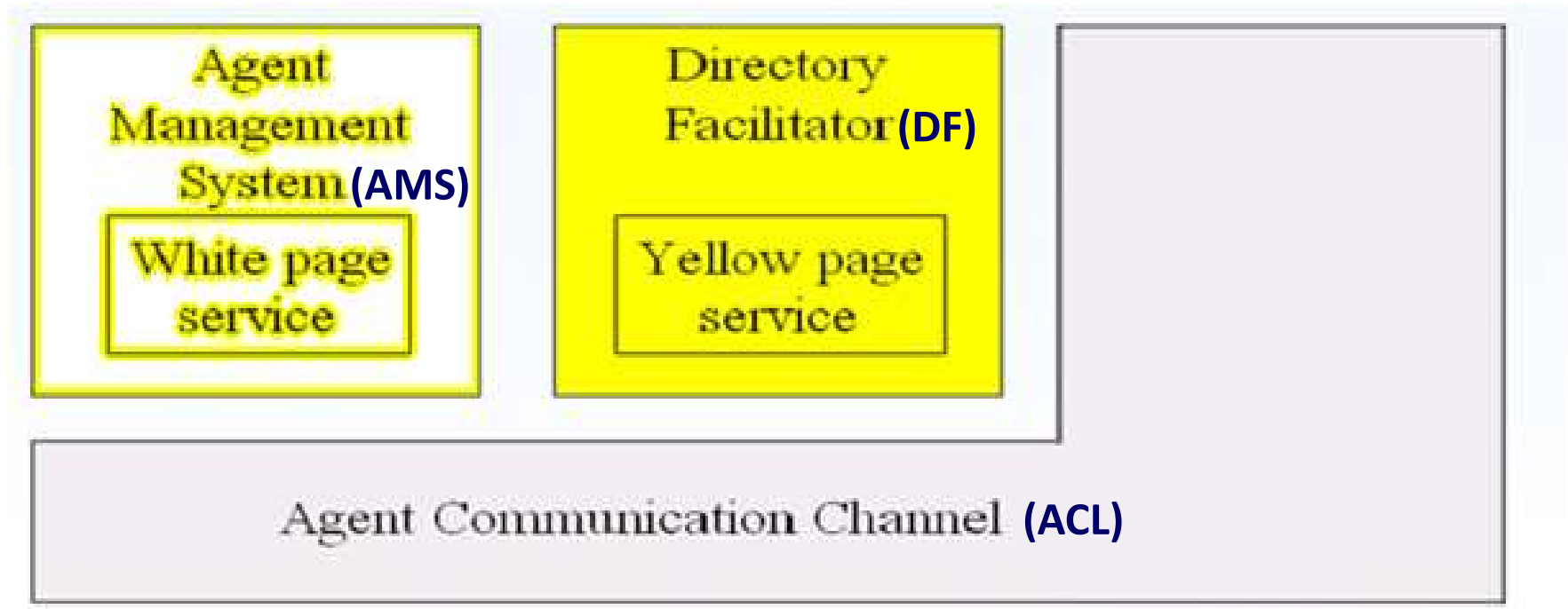
**Une suite d'outils graphiques** : qui facilitent la gestion et la supervision de la plateforme des agents.

# Introduction

- ✓ Chaque instance du JADE est appelée conteneur « container », elle peut contenir plusieurs agents.
- ✓ Un ensemble de conteneurs constituent une plateforme.
- ✓ Chaque plateforme doit contenir un conteneur spécial appelé « main-container » et tous les autres conteneurs s'enregistrent auprès de celui-là dès leur lancement.
- ✓ Un « main-container » contient trois agents spéciaux appelés **AMS**, **DF** et **ACL** qui sont créés automatiquement au lancement du « main-container »
  - **AMS** (Agent Management System)
  - **DF** (Directory Facilitator)
  - **ACL** (Agent Communication Channel)

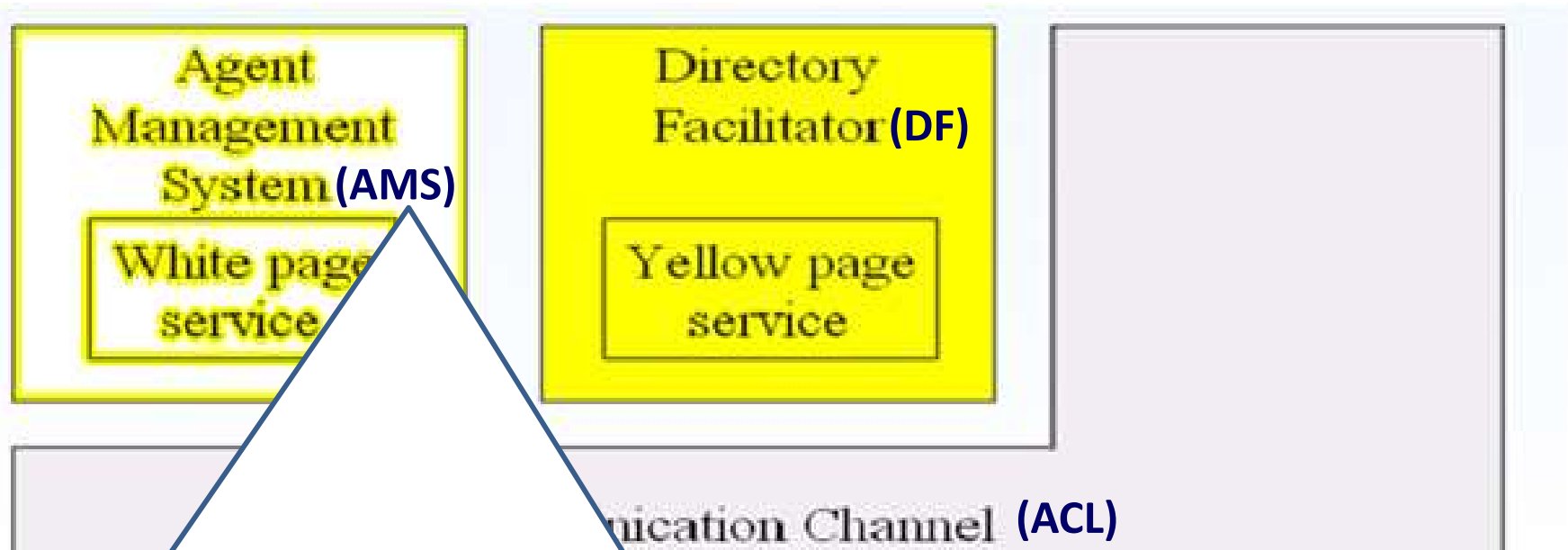
# Introduction

Architecture du main-container



# Introduction

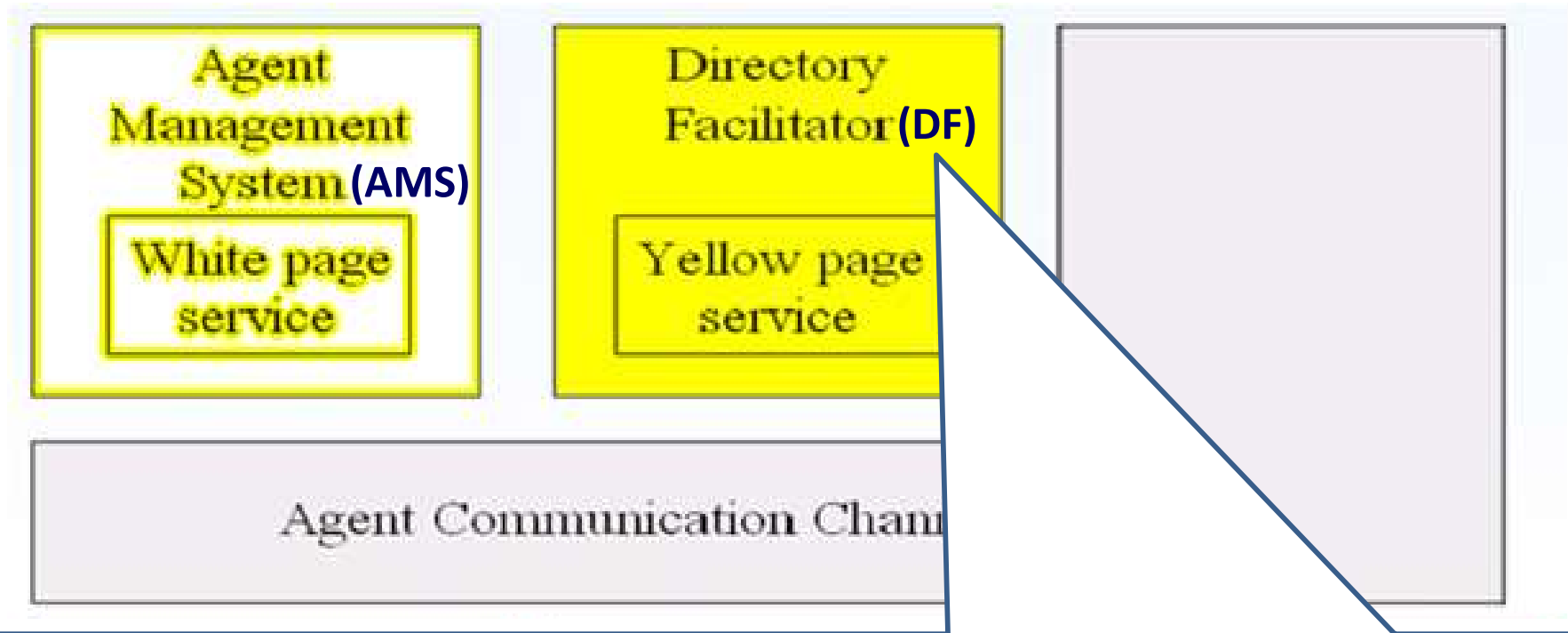
Architecture du main-container



- **AMS** (Agent Management System) permet de :
  - ✓ Gérer le cycle de vie des agents
  - ✓ Maintenir une liste de tous les agents qui résident sur la plate-forme (pages blanches)

# Introduction

Architecture du main-container



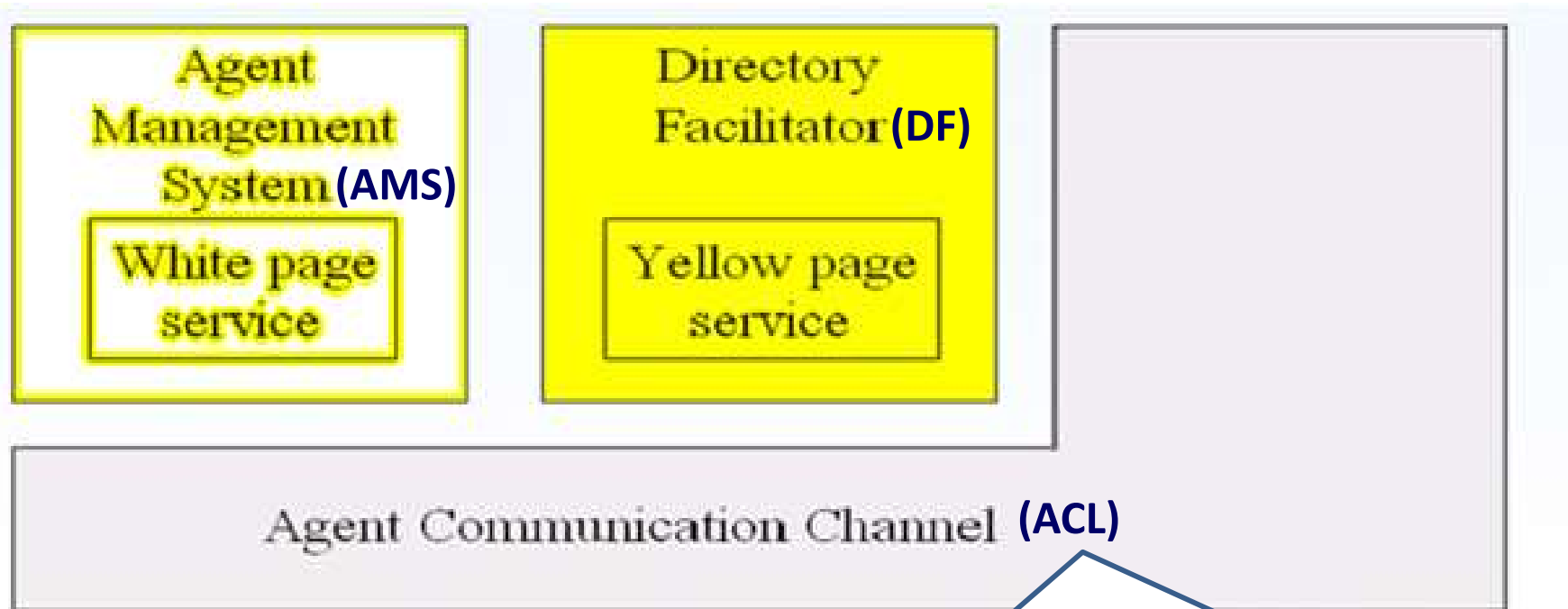
**DF** (Directory Facilitator) permet de :

- ✓ Enregistrer les services offerts par les agents (Pages jaunes)
- ✓ Rechercher les services offerts par les agents



# Introduction

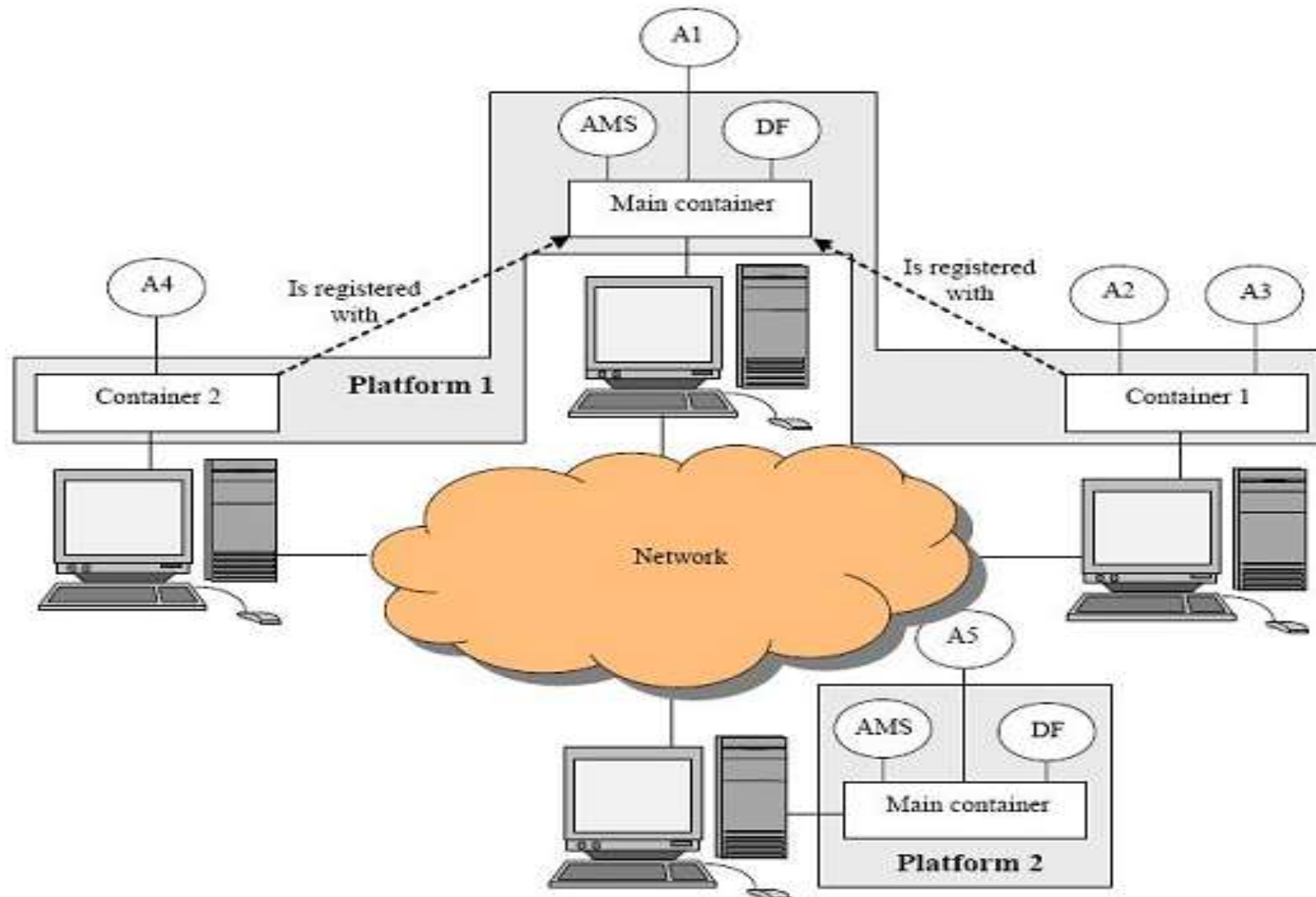
Architecture du main-container



- **ACL** (Agent Communication Channel) permet de :
  - ✓ Gérer les communications entre les agents

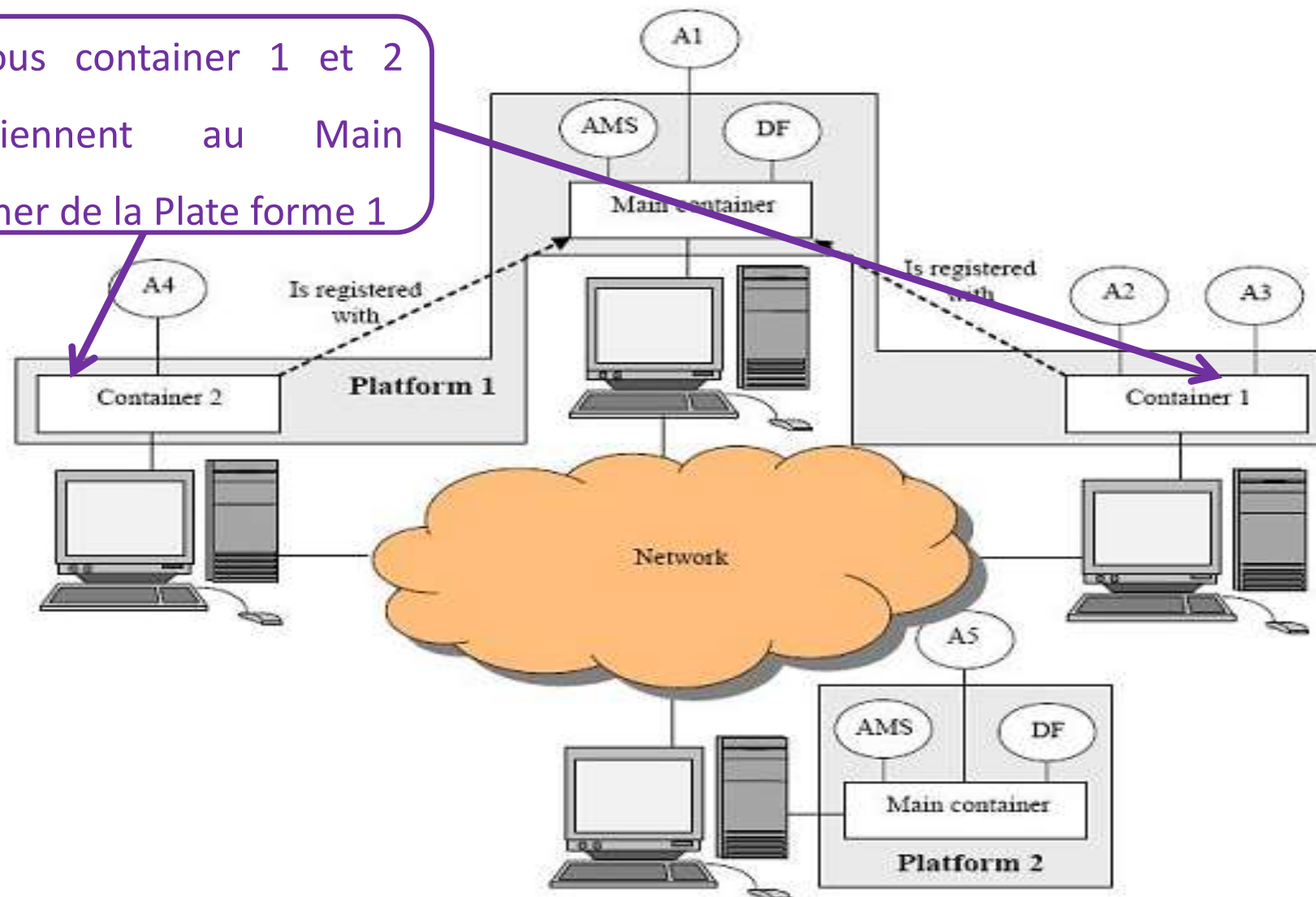
# Introduction

Exemple de Containers (Environnement d'exécution pour les agents)



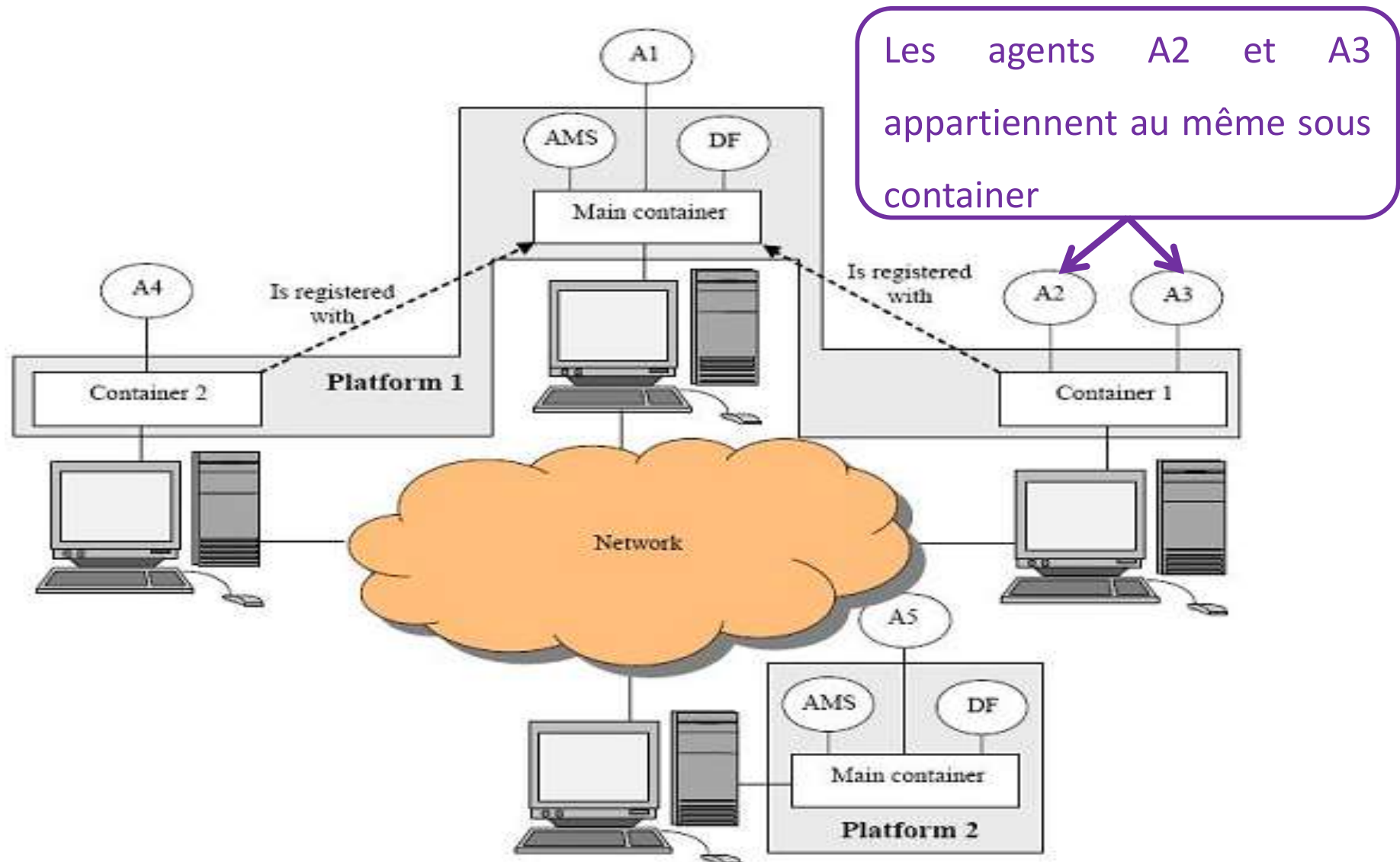
# Introduction

Exemple de Containers (Environnement d'exécution pour les agents)



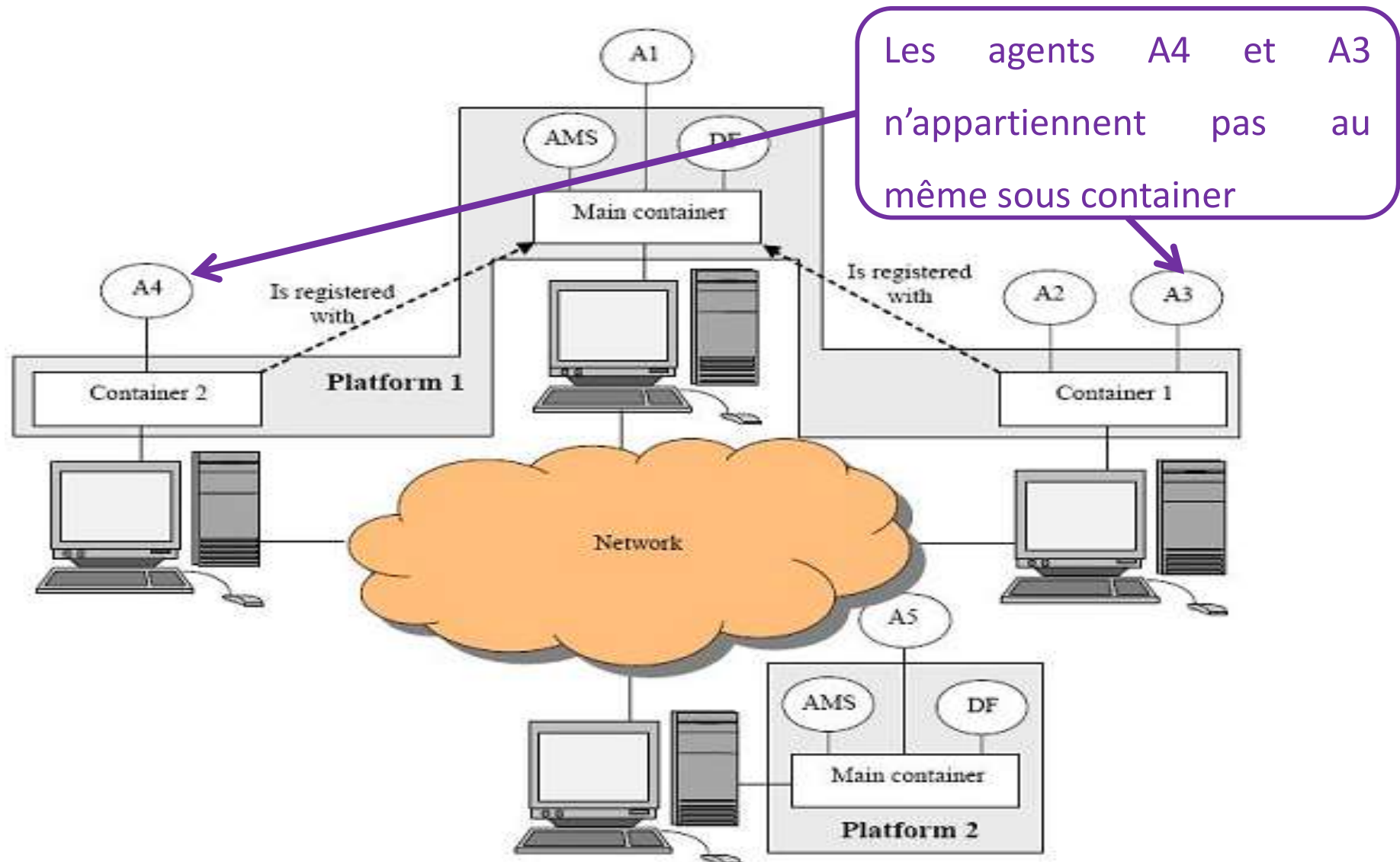
# Introduction

Exemple de Containers (Environnement d'exécution pour les agents)



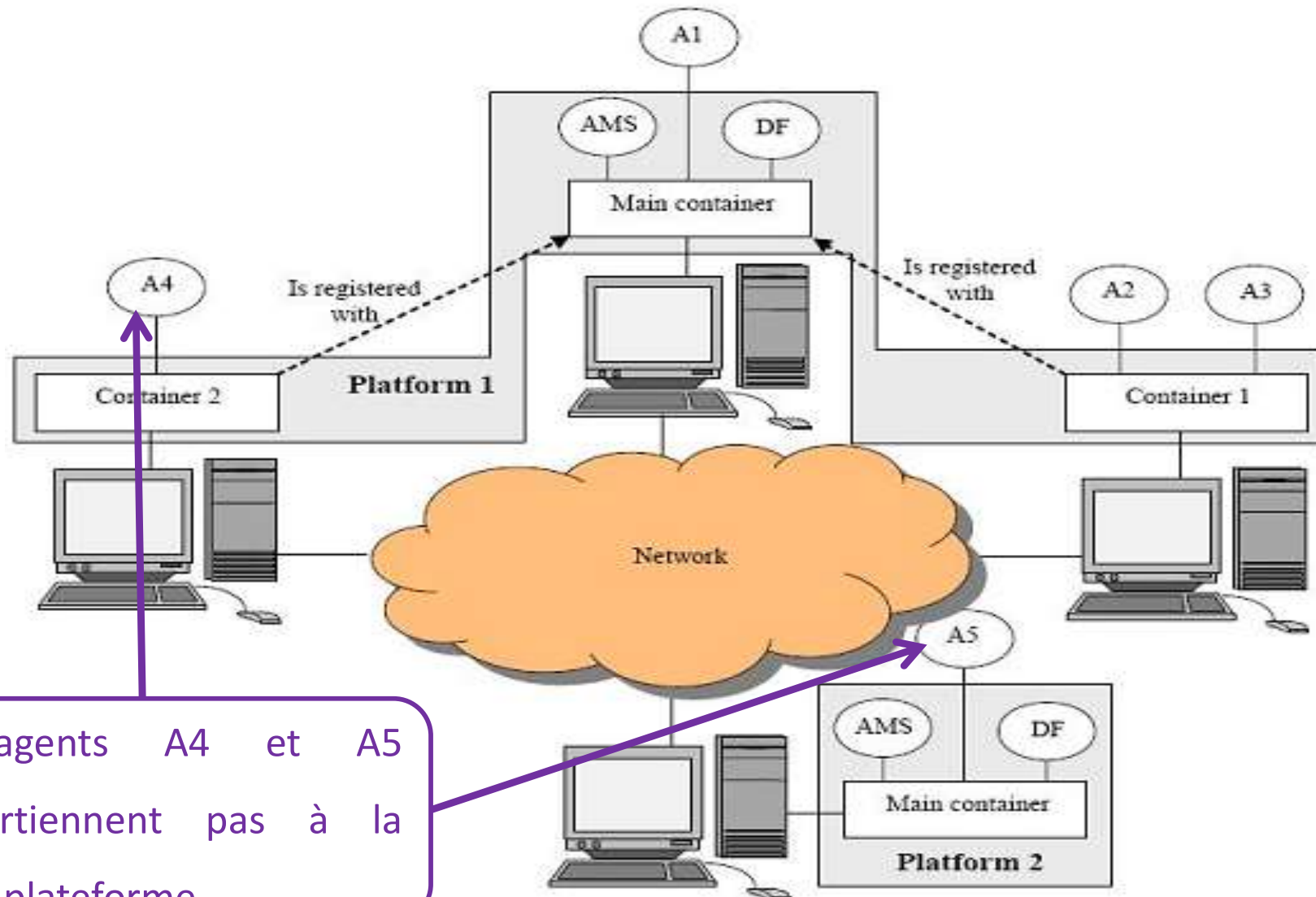
# Introduction

Exemple de Containers (Environnement d'exécution pour les agents)



# Introduction

Exemple de Containers (Environnement d'exécution pour les agents)



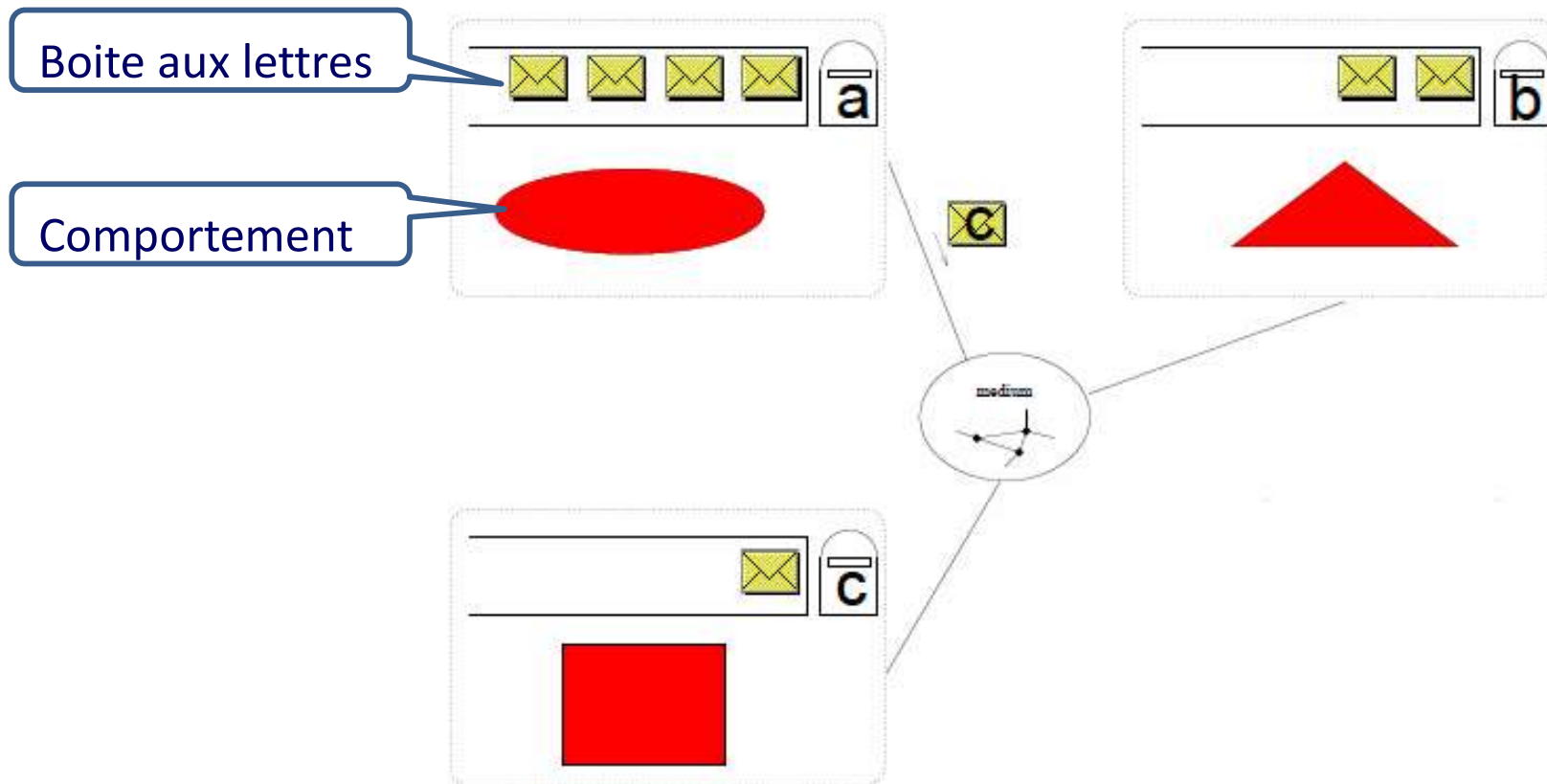
Les agents A4 et A5  
n'appartiennent pas à la  
même plateforme

# Systeme multi agents

- ✓ Un système multi agents est un système composé d'un ensemble d'agent.
- ✓ Un agent est une entité autonome réelle ou abstraite, possède un cycle de vie et peut **communiquer avec d'autres agents au moyen de messages**, les **messages** reçus sont stockés dans une **boîte aux lettres (file d'attente)**.
- ✓ Un agent possède des compétences et offres des services
- ✓ Un agent est caractérisé par un ou plusieurs **comportement** qui décrit la **réaction** de l'agent à un message reçu ou à une observation de ses connaissances. Ce **comportement** peut changer pendant son exécution.
- ✓ Un agent est identifié par un nom unique qui est l'**AgentIdentifier (AID)**
- ✓ Chaque agent peut joindre ou quitter librement la plateforme et rentrer en contact avec chacun des autres agents.

# Systeme multi agents

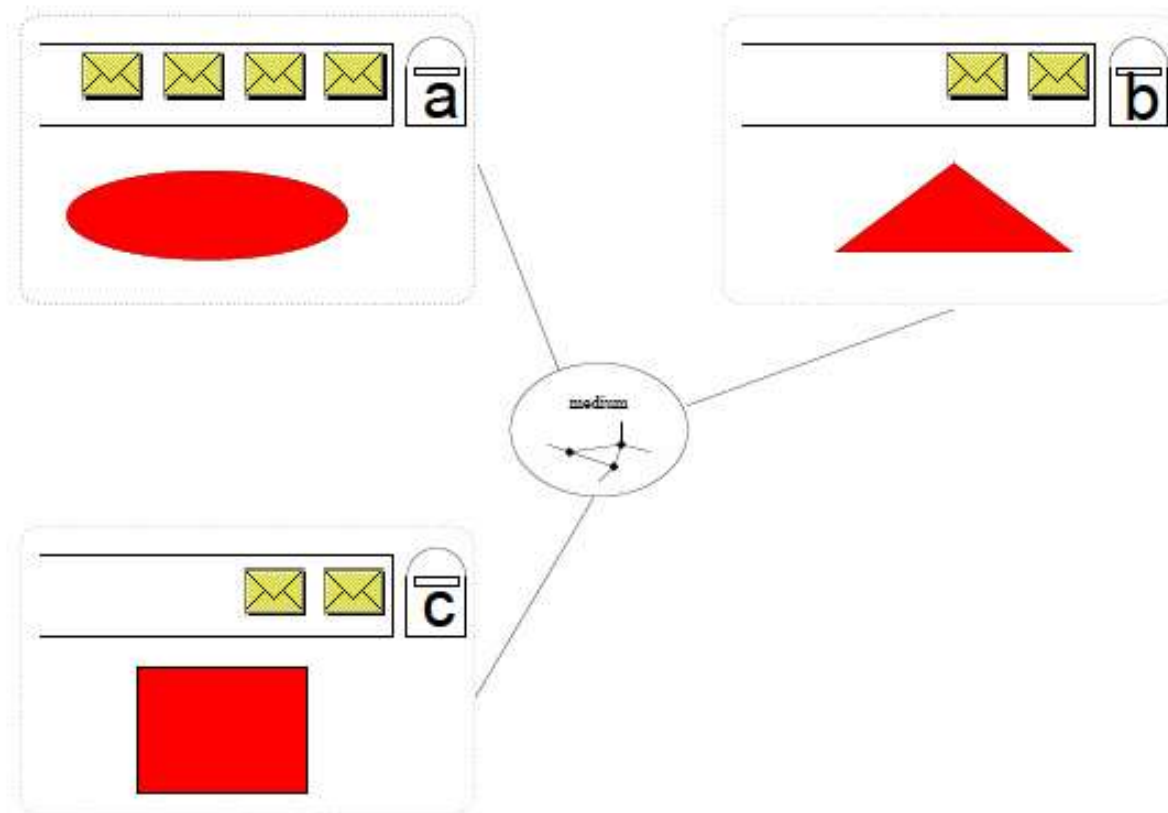
- ✓ Dans cet exemple, nous avons 3 agents « a », « b » et « c »
- ✓ On suppose que l'agent « a » a envoyé un message à l'agent « c »





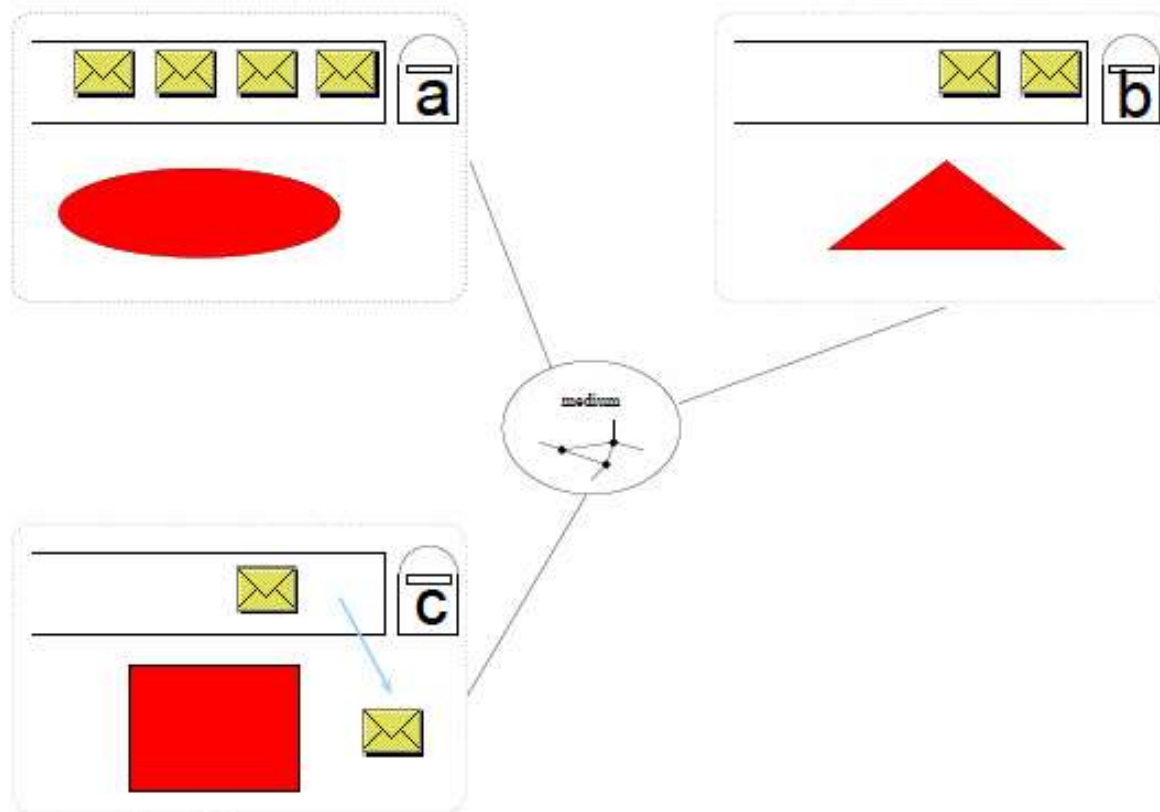
# Systeme multi agents

- ✓ L'agent « c » reçoit le message et le stocke dans sa boite aux lettres



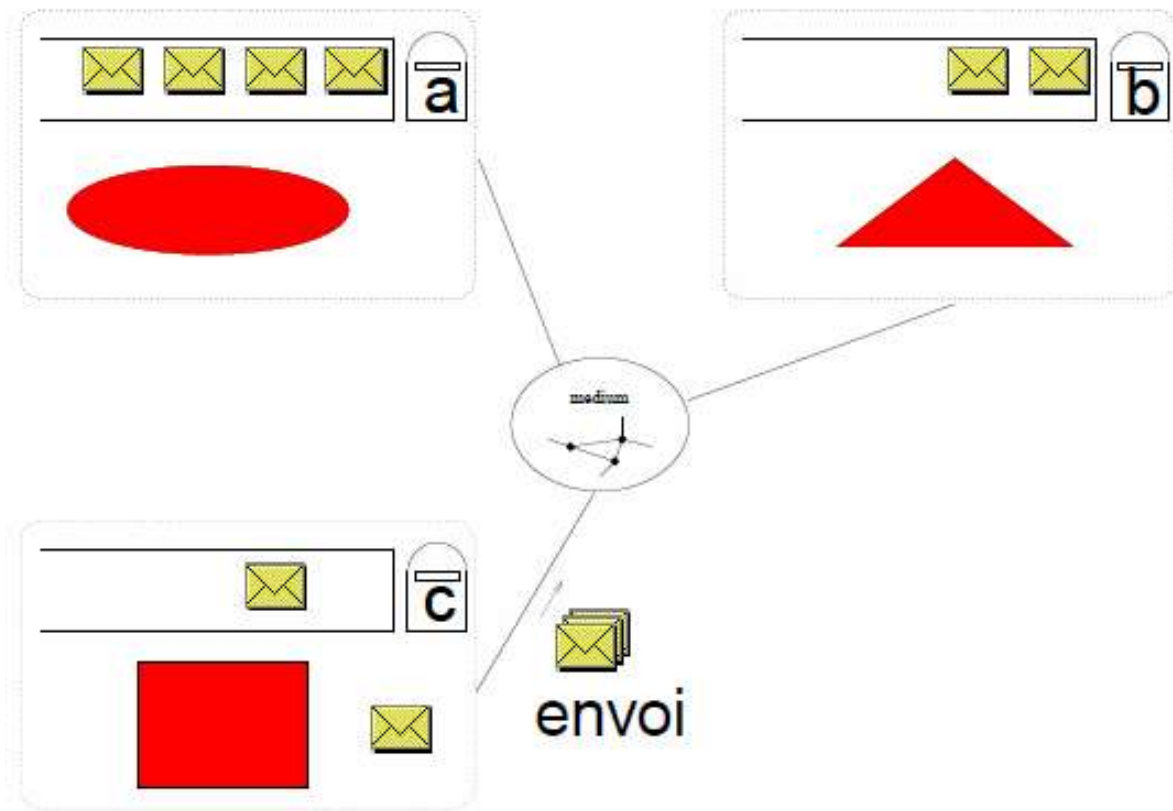
# Systeme multi agents

- ✓ L'agent « c » consulte le 1<sup>er</sup> message de sa boîte aux lettres



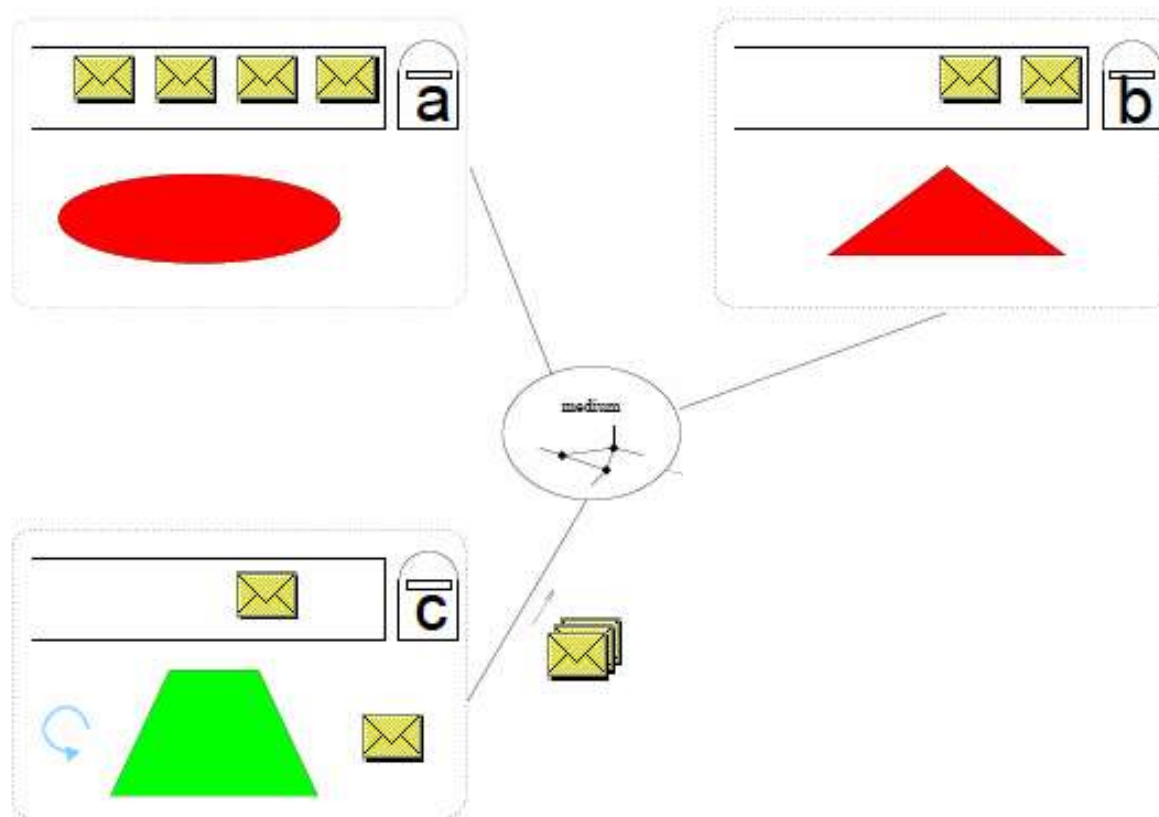
# Systeme multi agents

- ✓ L'agent « c » après la consultation du 1<sup>er</sup> message, il peut
  - Envoyer des messages aux autres agents



# Systeme multi agents

- ✓ L'agent « c » après la consultation du 1<sup>er</sup> message, il peut
  - Envoyer des messages aux autres agents
  - Changer son comportement



# Installation de la plate forme JADE

- Télécharger le fichier JADE-all-4.5.0.zip à partir du site : <http://www.jade.tilab.com>



- Décompresser le fichier JADE-all-4.5.0.zip.

(Clique droit sur le nom du fichier puis cliquer sur **Extraire vers JADE-all-4.5.0\**)

- Décompresser les fichiers (JADE-bin-4.5.0.zip, JADE-doc-4.5.0.zip, JADE-examples-4.5.0.zip, JADE-src-4.5.0.zip).

# Création du premier agent avec JADE sous JAVA

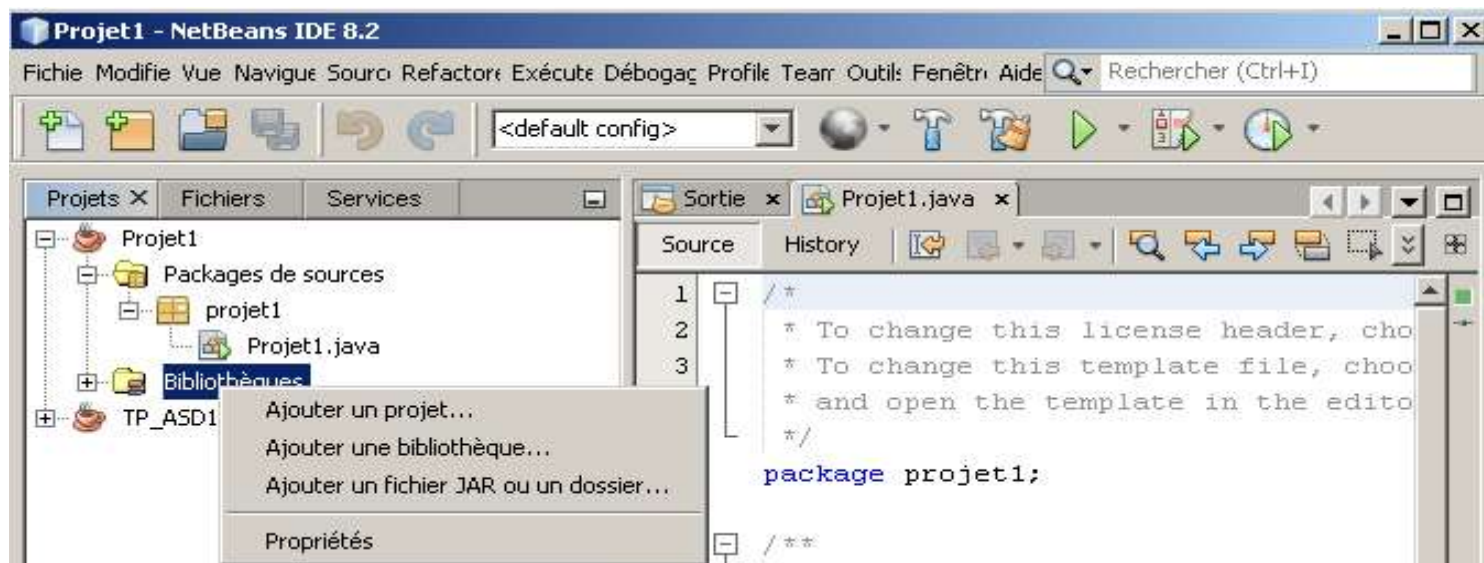
## Sous Eclipse :

- Lancer Eclipse
- Créer un nouveau projet : **File/New/Java Project** (Projet1 par exemple)
- Cliquer sur **Next**, puis sur l'onglet **Libraries** , ensuite sur le bouton **Add External JARs**, puis rajouter le fichier jade.jar (C:\JADE-all-4.5.0\JADE-bin-4.5.0\jade\lib)
- Cliquer sur le bouton **Finish**
- Créer une nouvelle classe : **File/New/Class** (AgentTest par exemple)
- Cliquer sur le bouton **Finish**
- Taper le code suivant:

# Création du premier agent avec JADE sous JAVA

## Sous NetBeans :

- Lancer NetBeans
- Cliquer sur **Fichier/Nouveau projet.../Java**
- Cliquer sur **Suivant**
- Donner un nom à votre projet (**Projet1** par exemple)
- Dans l'arborescence du projet, faire une clique droit sur **Bibliothèques**



# Création du premier agent avec JADE sous JAVA

## Sous NetBeans :

- Cliquer sur **Ajouter un fichier JAR ou un dossier....**
- Rajouter le fichier jade.jar (C:\JADE-all-4[1].0.1\JADE-bin-4.5.0\jade\lib)
- Cliquer sur le bouton **Ouvrir**
- Cliquer sur Créer une nouvelle classe : **Fichier/Nouveau fichier..../Java Class**
- Cliquer sur **Suivant**
- Donner un nom à votre classe (**AgentTest** par exemple)
- Cliquer sur le bouton **Terminer**
- Taper le code suivant:



# Création du premier agent avec JADE et ECLIPSE

```
import jade.core.Agent;
```

```
public class AgentTest extends Agent{ //l'agent hérite de la classe  
jade.core.Agent
```

```
protected void setup() {
```

```
    System.out.println("Je suis l'agent : "+this.getLocalName());
```

```
    System.out.println("Je suis l'agent : "+this.getName());
```

```
}
```

```
}
```

# Création du premier agent avec JADE sous JAVA

- Pour tester votre premier agent, taper le code suivant :

```
public class Test {  
    public static void main(String[] args) {  
        String [] commande = new String[3];  
        String argument = "";  
        argument = argument+"agent1:AgentTest";  
        commande [0]="-cp";  
        commande [1]="jade.boot";  
        commande [2]= argument;  
        jade.Boot.main(commande);  
    }  
}
```

Nom local de l'agent

Nom de la classe de l'agent

- Cliquer sur **Run** pour exécuter la méthode main de la classe Test
- **Remarque 1** : Si vous utilisez un package alors utilisez l'instruction suivante :

argument = argument+"agent1:**NomPackage**.AgentTest";