

JADE - Java Agent DEvelopment Framework

Présentation plateforme Java du 21/10/19

Guillaume Laroyenne, Paul Lemettre, Hervé Meyer, Kévin Ouahmad, Adrien Schmuck



1

Le concept SMA (Système Multi Agents)

C'est quoi un agent ?

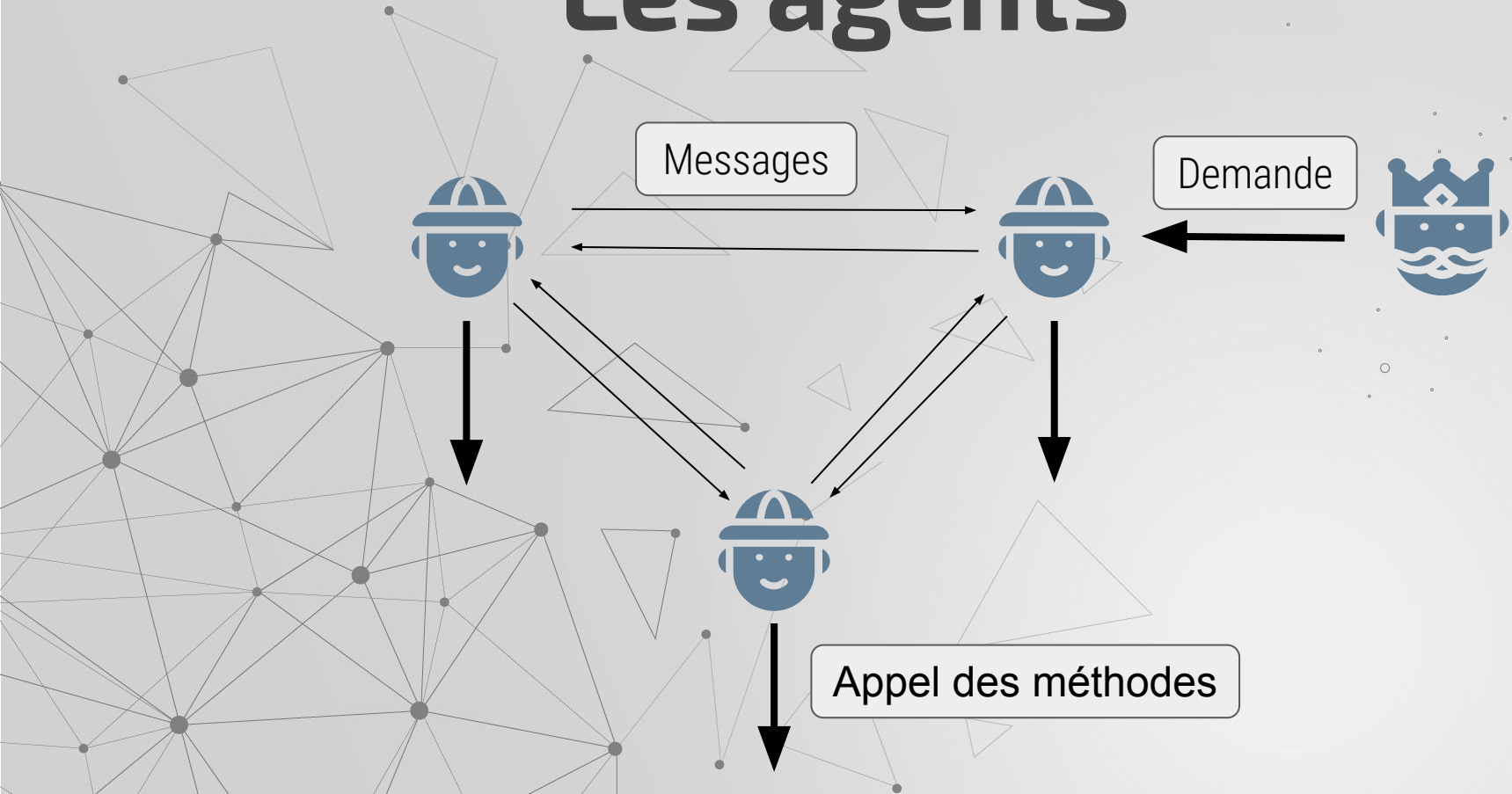
Définition

Propriétés

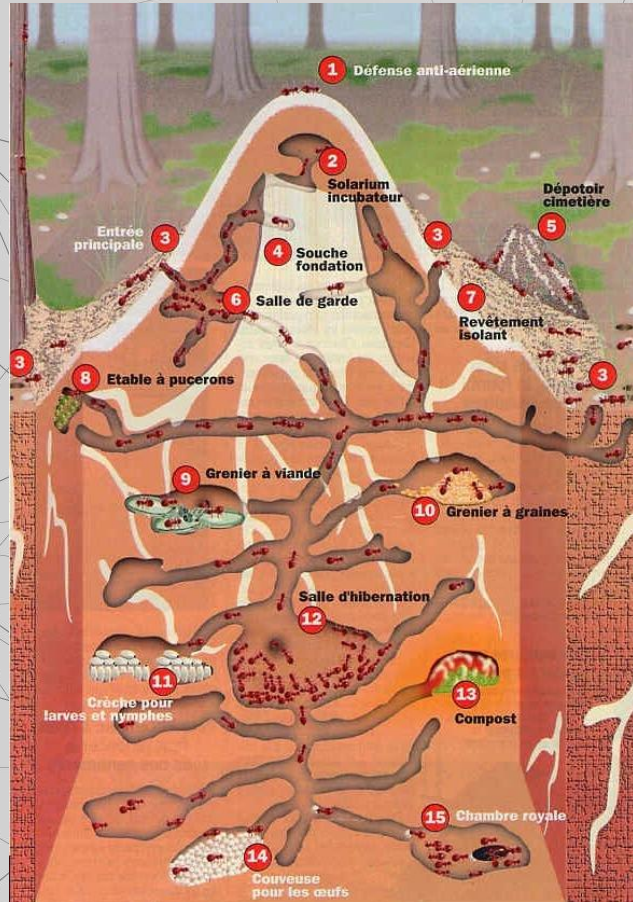
Avantages



Les agents



Les agents

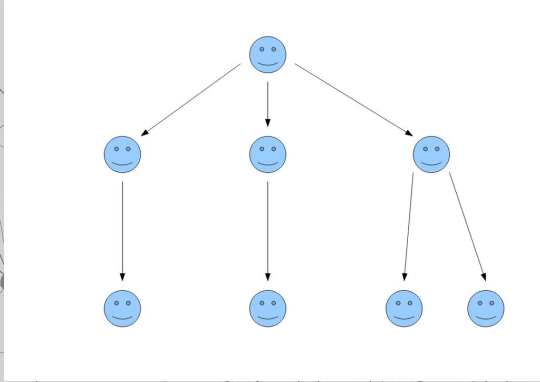


Un agent = Une fourmi

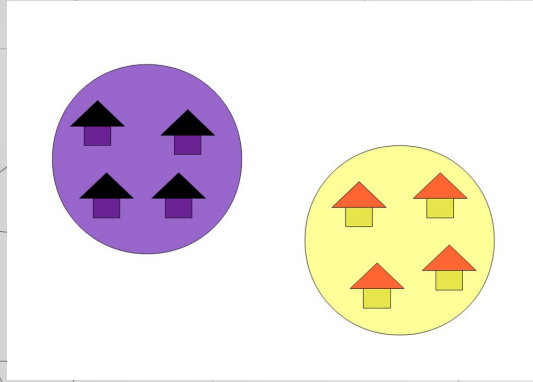
Un système = Une fourmilière

<http://www.antzzz.org/fourmis.php?fourmis=fourmis-fourmiliere&fourmiliere=dome>

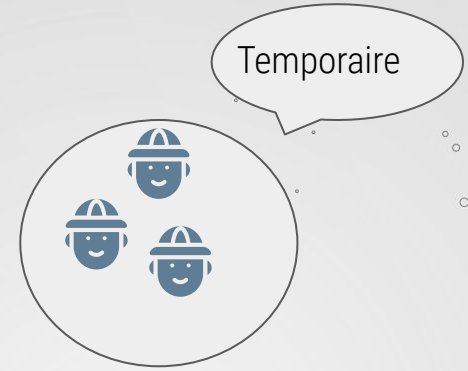
Organisation des agents



Hiéarchies



Holarchies

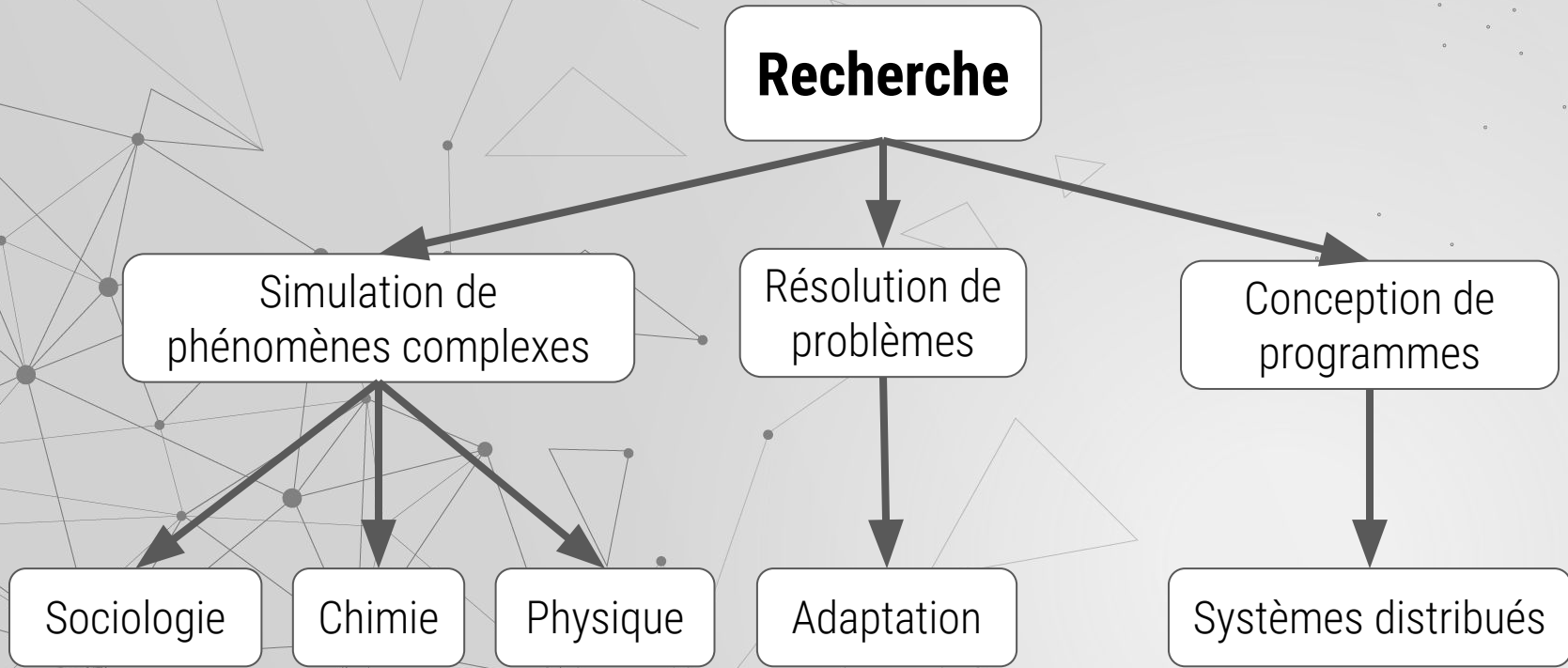


Coalitions

Algorithmes

Problème du voyageur
de commerce

Utilisation de nos jours



Utilisation de nos jours

Industrie



Seigneur des anneaux

Finance





2

Présentation de l'API JADE

Présentation de l'API JADE

Les origines



Développée depuis
1998, libérée en 2000
sous LGPL.



Telecom Italia Lab, R&D de Telecom
Italia.

Plus quelques
membres du comité
directeurs :



MOTOROLA



WHITESTEIN
Technologies



Présentation de l'API JADE

Le but



Présentation de l'API JADE

La FIPA

Foundation for
Intelligent Physical
Agents



- IEEE Computer Society standards organization
Association de professionnels pour normaliser les SMA
- Établissement de protocoles de communication multi-agents
- Maximiser l'interopérabilité des applications SMA
Définition de technologies génériques pour différent champs d'application

JADE implémente les spécifications pour que les agents FIPA puissent :

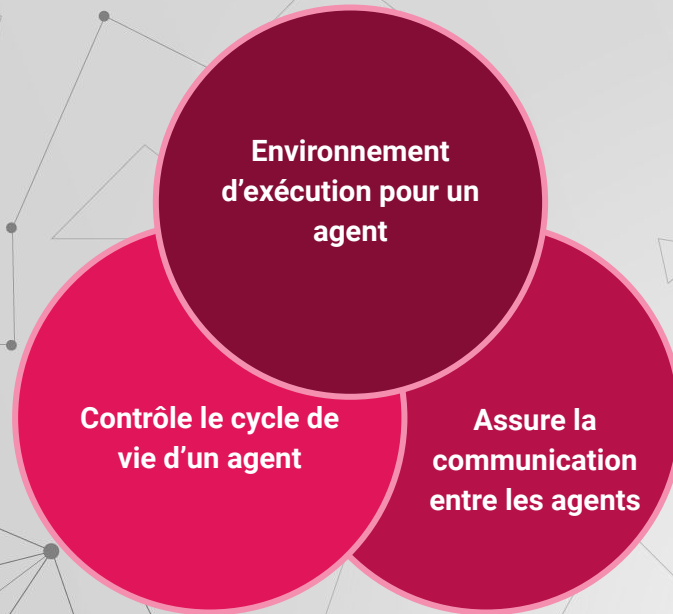
- Exister
- Fonctionner
- Communiquer



3

Mise en oeuvre avec JADE

Les conteneurs



```
Runtime runtime = Runtime.instance();
```

```
ProfileImpl profile = new ProfileImpl(false); // Pas un conteneur principal  
profile.setParameter(Profile.MAIN_HOST, "localhost");
```

```
AgentContainer container = runtime.createAgentContainer(profile);
```

Les agents

Entité exécutable
au sein d'un
conteneur

Communiquent entre
eux via des messages

Possibilité de
créer plusieurs
instances d'un
même agent

```
public class BuyerAgent extends Agent {  
  
    @Override  
    public void setup() {  
  
    }  
  
    @Override  
    protected void takeDown() {  
  
    }  
  
}
```

```
AgentController agent = container.createNewAgent("Nom de l'instance", BuyerAgent.class.getName(), new Object[0]);
```

```
container.start();  
agent.start();
```


Les messages ACL

- Permettent l'échange d'information entre agents
- Exemple : demander le prix d'un produit à un vendeur

- PROPOSE
- REQUEST
- CANCEL
- AGREE
- CONFIRM
- DISCONFIRM
- INFORM
- FAILURE
-

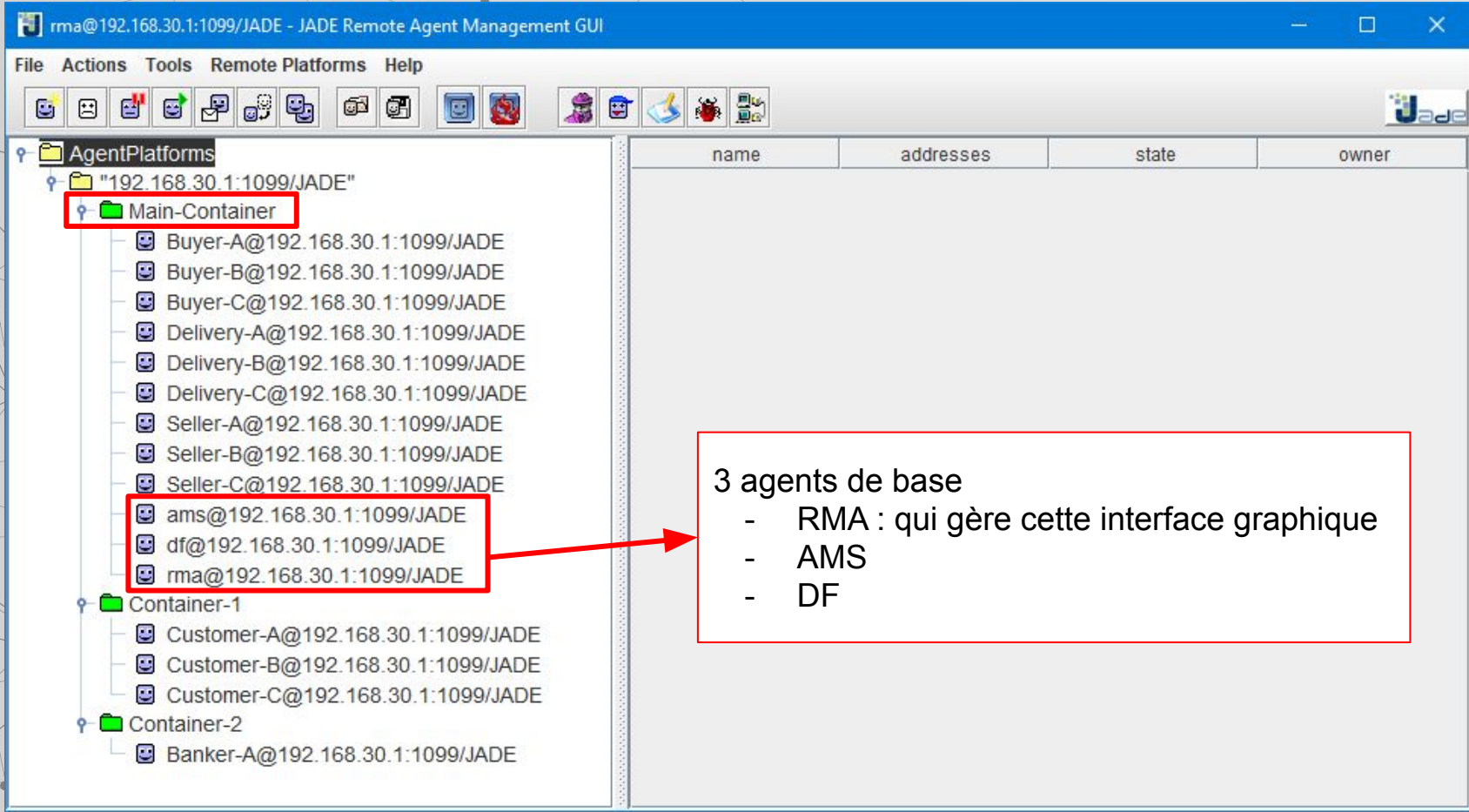
```
Product product = new Apple();
```

```
ACLMessage message = new ACLMessage(ACLMessage.REQUEST);  
message.setOntology("myOntology");  
message.setProtocol("requestPrice");  
message.setConversationId("1");  
message.setContentObject(product); // ou setContent("une chaine")  
message.addReceiver(...);
```

```
myAgent.send(message);
```

Problème : on ne connaît pas les vendeurs !

Le conteneur principal



The screenshot displays the JADE Remote Agent Management GUI. The title bar reads "rma@192.168.30.1:1099/JADE - JADE Remote Agent Management GUI". The menu bar includes "File", "Actions", "Tools", "Remote Platforms", and "Help". The toolbar contains various icons for agent management. The left pane shows a tree structure under "AgentPlatforms":

- AgentPlatforms
 - "192.168.30.1:1099/JADE"
 - Main-Container
 - Buyer-A@192.168.30.1:1099/JADE
 - Buyer-B@192.168.30.1:1099/JADE
 - Buyer-C@192.168.30.1:1099/JADE
 - Delivery-A@192.168.30.1:1099/JADE
 - Delivery-B@192.168.30.1:1099/JADE
 - Delivery-C@192.168.30.1:1099/JADE
 - Seller-A@192.168.30.1:1099/JADE
 - Seller-B@192.168.30.1:1099/JADE
 - Seller-C@192.168.30.1:1099/JADE
 - ams@192.168.30.1:1099/JADE
 - df@192.168.30.1:1099/JADE
 - rma@192.168.30.1:1099/JADE
 - Container-1
 - Customer-A@192.168.30.1:1099/JADE
 - Customer-B@192.168.30.1:1099/JADE
 - Customer-C@192.168.30.1:1099/JADE
 - Container-2
 - Banker-A@192.168.30.1:1099/JADE

The right pane shows a table with columns: "name", "addresses", "state", and "owner". It is currently empty.

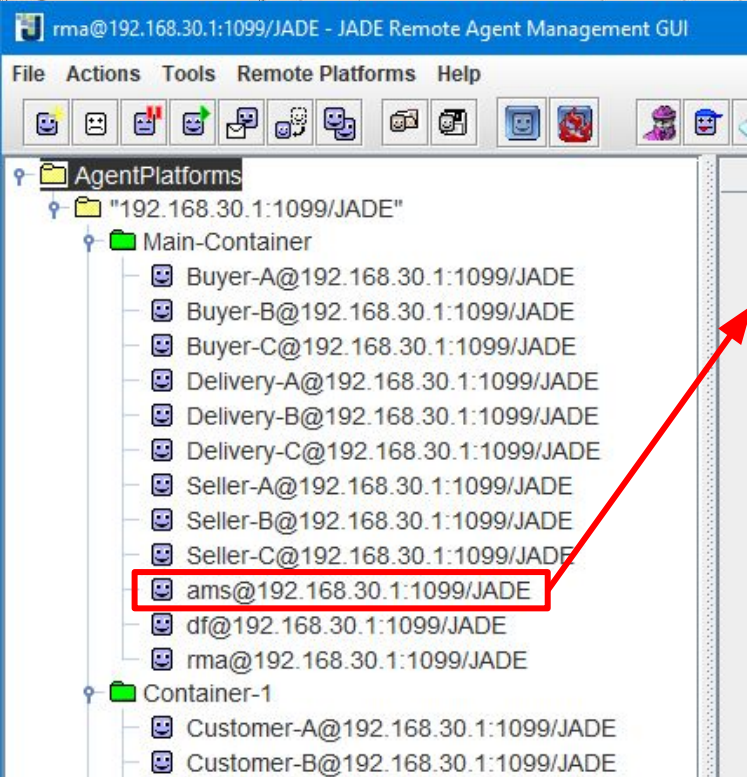
A red box highlights the "Main-Container" and its sub-agents. A red arrow points from the "ams@192.168.30.1:1099/JADE", "df@192.168.30.1:1099/JADE", and "rma@192.168.30.1:1099/JADE" agents to a text box on the right.

3 agents de base

- RMA : qui gère cette interface graphique
- AMS
- DF

L'agent AMS

Agent Management System



- Gérer le cycle de vie des agents
- Chaque agent
 - S'enregistre auprès de l'AMS avant *setup()*
 - Se désenregistre auprès de l'AMS après *takeDown()*
- Il y a un seul AMS par plateforme
- L'AMS
 - Liste d'identifiant de tous les agents
 - Où ils se trouvent

Cycle de vie d'un agent

Géré par AMS

INITIATED : l'agent est lancé mais non enregistré auprès de l'AMS, aucun nom, aucune adresse.

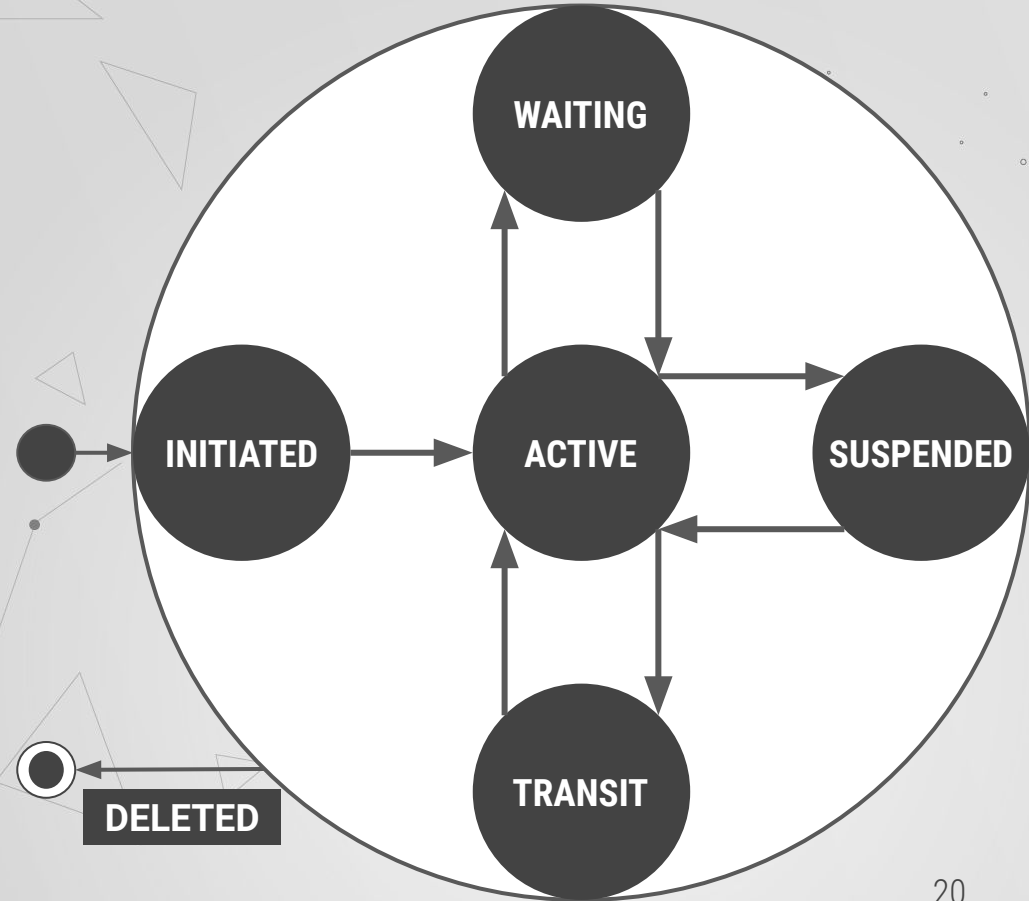
ACTIVE : l'agent est répertorié auprès de l'AMS et peut accéder aux services.

SUSPENDED : tous les comportements de l'agent sont suspendus.

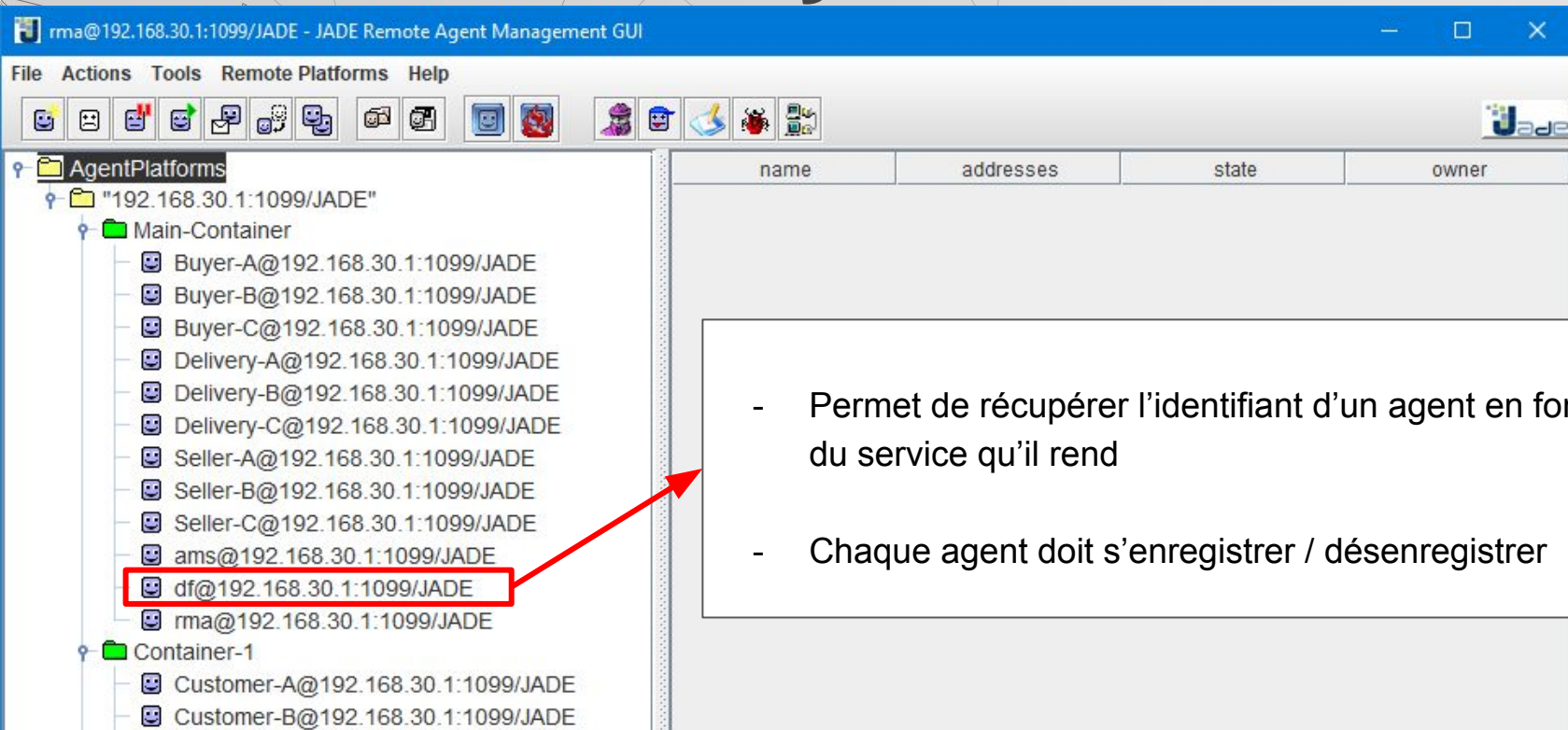
TRANSIT : l'agent migre vers une autre plateforme.

WAITING : tous les comportements de l'agent sont temporairement interrompus.

DELETED : l'exécution de l'agent est terminée et n'est plus répertorié au sein de l'AMS.



L'agent DF Directory Facilitator



The screenshot shows the JADE Remote Agent Management GUI. The title bar reads 'rma@192.168.30.1:1099/JADE - JADE Remote Agent Management GUI'. The menu bar includes 'File', 'Actions', 'Tools', 'Remote Platforms', and 'Help'. The toolbar contains various icons for agent management. The left pane shows a tree view of 'AgentPlatforms' with a sub-folder '192.168.30.1:1099/JADE' containing a 'Main-Container' and a 'Container-1'. The 'Main-Container' lists several agents, with 'df@192.168.30.1:1099/JADE' highlighted by a red box. A red arrow points from this box to a callout box on the right.

name	addresses	state	owner
Buyer-A@192.168.30.1:1099/JADE			
Buyer-B@192.168.30.1:1099/JADE			
Buyer-C@192.168.30.1:1099/JADE			
Delivery-A@192.168.30.1:1099/JADE			
Delivery-B@192.168.30.1:1099/JADE			
Delivery-C@192.168.30.1:1099/JADE			
Seller-A@192.168.30.1:1099/JADE			
Seller-B@192.168.30.1:1099/JADE			
Seller-C@192.168.30.1:1099/JADE			
ams@192.168.30.1:1099/JADE			
df@192.168.30.1:1099/JADE			
rma@192.168.30.1:1099/JADE			

- Permet de récupérer l'identifiant d'un agent en fonction du service qu'il rend
- Chaque agent doit s'enregistrer / désenregistrer

Echange de messages entre agents

Plateforme

Transfert de message entre plateforme

Main Container

ACC

DF

AMS

A1

A2

A3

Message transport system

Utilisation de RMI pour transférer les messages

Container 1

A4

A5

A6

Message transport system

Container 2

A7

A8

A9

Message transport system

Ajout d'un agent au DF

```
public class SellerAgent extends Agent {
```

```
    @Override
```

```
    public void setup() {
```

```
        try {
```

```
            DfAgentDescription dfAgentDescription = new DfAgentDescription();  
            dfAgentDescription.setName(this.getAID());
```

```
            ServiceDescription serviceDescription = new ServiceDescription();  
            serviceDescription.setName(getLocalName());  
            serviceDescription.setType("sellerService");
```

```
            dfAgentDescription.addServices(serviceDescription);
```

```
            DfService.register(this, dfAgentDescription);
```

```
        } catch (FIPAException e) {
```

```
            e.printStackTrace();
```

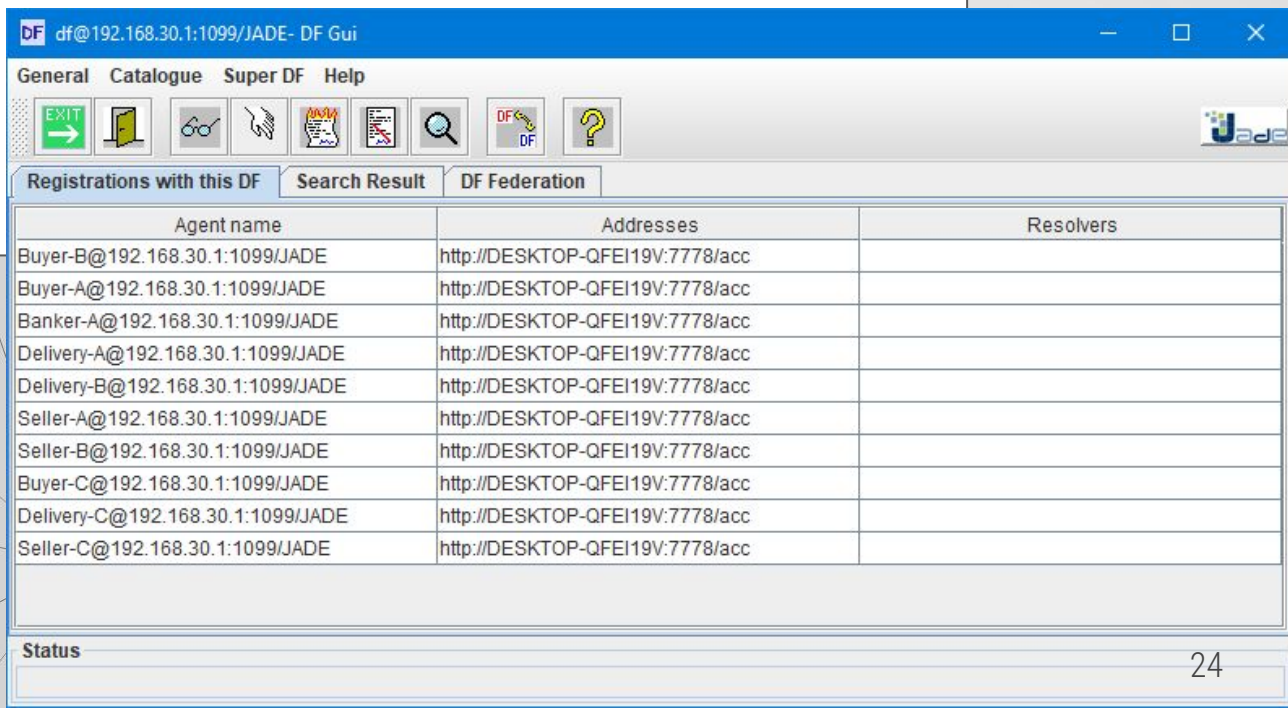
```
        }
```

```
    }
```


Suppression d'un agent du DF

@Override

```
protected void takeDown() {  
    try{  
        DFService.deregister(this);  
    } catch (FIPAException e) {  
        e.printStackTrace();  
    }  
}
```



The screenshot shows the 'df@192.168.30.1:1099/JADE- DF Gui' window. The 'Registrations with this DF' tab is active, displaying a table of registered agents. The table has three columns: 'Agent name', 'Addresses', and 'Resolvers'. The agents listed are Buyer-B, Buyer-A, Banker-A, Delivery-A, Delivery-B, Seller-A, Seller-B, Buyer-C, Delivery-C, and Seller-C, all with the same address and no resolvers.

Agent name	Addresses	Resolvers
Buyer-B@192.168.30.1:1099/JADE	http://DESKTOP-QFEI19V:7778/acc	
Buyer-A@192.168.30.1:1099/JADE	http://DESKTOP-QFEI19V:7778/acc	
Banker-A@192.168.30.1:1099/JADE	http://DESKTOP-QFEI19V:7778/acc	
Delivery-A@192.168.30.1:1099/JADE	http://DESKTOP-QFEI19V:7778/acc	
Delivery-B@192.168.30.1:1099/JADE	http://DESKTOP-QFEI19V:7778/acc	
Seller-A@192.168.30.1:1099/JADE	http://DESKTOP-QFEI19V:7778/acc	
Seller-B@192.168.30.1:1099/JADE	http://DESKTOP-QFEI19V:7778/acc	
Buyer-C@192.168.30.1:1099/JADE	http://DESKTOP-QFEI19V:7778/acc	
Delivery-C@192.168.30.1:1099/JADE	http://DESKTOP-QFEI19V:7778/acc	
Seller-C@192.168.30.1:1099/JADE	http://DESKTOP-QFEI19V:7778/acc	

Status

Envoyer un message aux vendeurs

```
ServiceDescription serviceDescription = new ServiceDescription();  
serviceDescription.setType("sellerService");  
  
DFAgentDescription dfAgentDescription = new DFAgentDescription();  
dfAgentDescription.addServices(serviceDescription);  
  
DFAgentDescription[] agents = DFService.search(myAgent, dfAgentDescription);
```

Recherche des agents fournissant le service "vendeurs" dans le Directory Facilitator

```
ACLMessage message = new ACLMessage(ACLMessage.REQUEST);  
message.setOntology("myOntology");  
message.setProtocol("requestPrice");  
...
```

```
for (DFAgentDescription agent : agents) {  
    message.addReceiver(agent.getName());  
}
```

Envoie du message à tous les vendeurs

```
myAgent.send(message);
```

4


Conception d'un agent

Description d'un agent

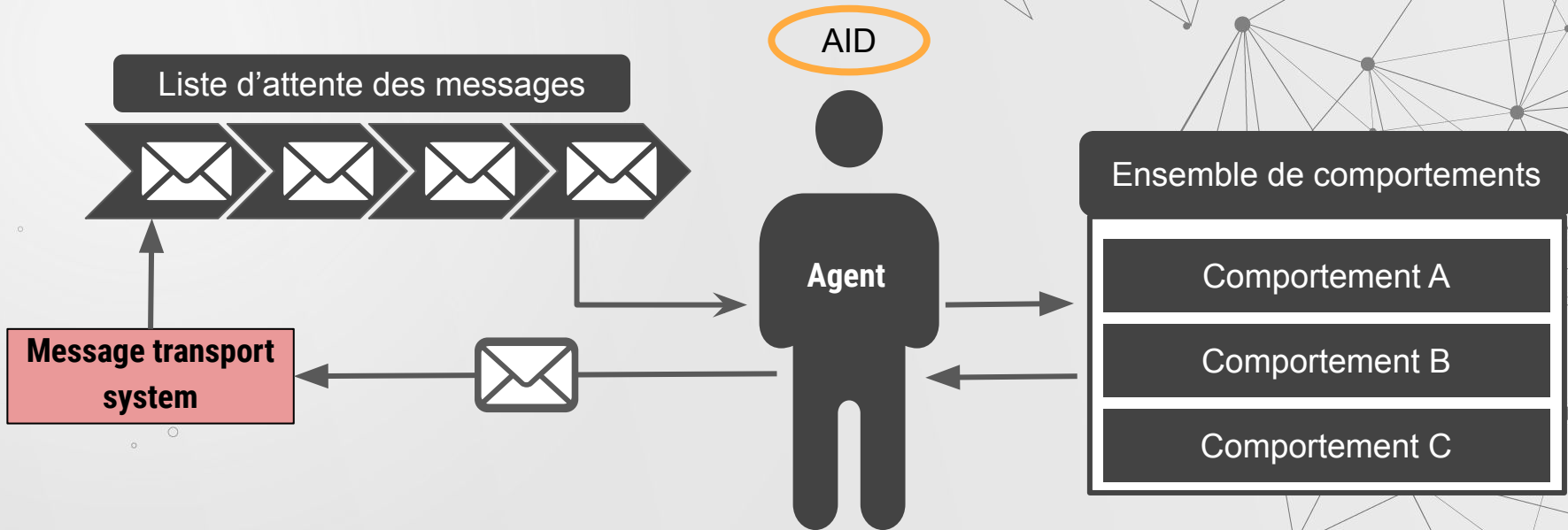
Sur plateforme JADE

Comment traiter les messages ?

Comment traiter plusieurs tâches dans un thread ?

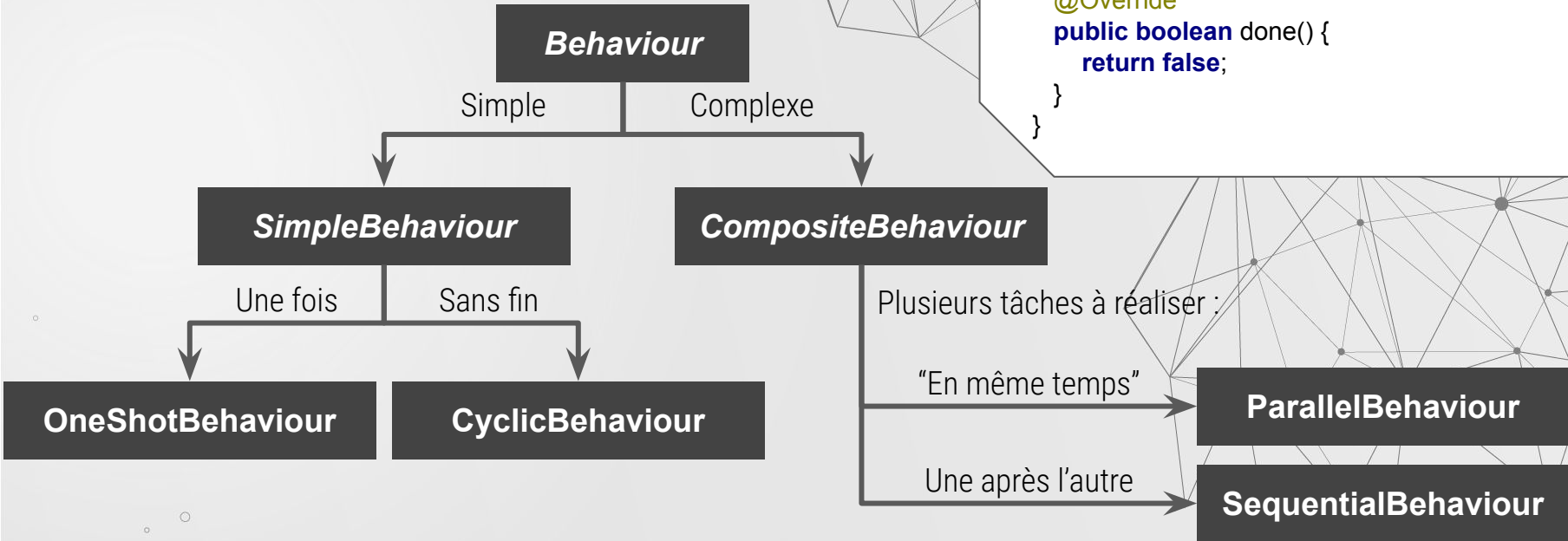
1  = 1 Thread

Tout doit être
sérialisable



Création d'un comportement

```
public class MyBehaviour extends Behaviour {  
    public MyBehaviour(Agent agent) {  
        super(agent);  
    }  
    @Override  
    public void action() {  
        System.out.println("Processing !");  
    }  
    @Override  
    public boolean done() {  
        return false;  
    }  
}
```



Modèle d'exécution d'un agent

@Override

```
protected void setup() {  
    System.out.println("Activation l'agent");  
    addBehaviour(new MyBehaviour(this));  
}
```

@Override

```
protected void action() {  
    System.out.println("Appel de l'action");  
}
```

@Override

```
protected boolean done() {  
    return false;  
}
```

@Override

```
protected void takeDown() {  
    System.out.println("Fin de l'agent");  
}
```

Ne pas faire de
comportement
bloquant !

Initialisation de l'agent

Agent actif ?

Non

Oui

Récupérer le comportement
actif au dessus de la file

Exécuter le comportement

Comportement
terminé ?

Non

Oui

Retirer le comportement

Arrêt de l'agent

Traitement des messages

Recevoir un message

Dans la classe d'un agent

```
ACLMessage aclMessage = this.blockingReceive();  
System.out.println("New message : " + aclMessage);
```

Dans la classe d'un comportement

```
ACLMessage aclMessage = myAgent.receive();  
if (aclMessage != null) {  
    System.out.println("New message : " + aclMessage);  
} else {  
    block();  
}
```

Comment filtrer les messages ?

Filtrer des messages

Utilisation de la classe :
MessageTemplate

```
MessageTemplate messageTemplate =  
MessageTemplate.and(  
    MessageTemplate.MatchOntology("MyOntology"),  
    MessageTemplate.MatchProtocol("MyProtocol"));
```

```
myAgent.receive(messageTemplate);
```

Attention il ne faut pas oublier de faire la gestion des messages indésirables .
→ Utilisation de l'opérateur logique *not*.

GUI Agent

Marquer l'interface graphique
comme *transient*

Lancer l'interface graphique
dans la méthode *setup*

Supprimer l'interface
graphique dans *takeDown*

Redéfinir les méthodes :
beforeMove
afterMove
afterClone

Redéfinir la méthode
onGuiEvent

```
GuiEvent guiEvent = new GuiEvent(this, 2);  
myAgent.postGuiEvent(guiEvent);
```

```
public class MyGuiAgent extends GuiAgent {
```

```
    private transient JFrame jFrame;
```

```
    public MyGuiAgent() { jFrame = null; }
```

```
    @Override
```

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

```
        try {
```

```
            SwingUtilities.invokeLater(() -> jFrame = new JFrame());
```

```
            jFrame.setVisible(true);
```

```
        } catch (InterruptedException | InvocationTargetException e) {
```

```
            e.printStackTrace();
```

```
            doDelete();
```

```
        }
```

```
    }
```

```
    @Override
```

```
    protected void takeDown() {
```

```
        jFrame.setVisible(false);
```

```
        jFrame.dispose();
```

```
    }
```

```
    @Override
```

```
    protected void onGuiEvent(GuiEvent guiEvent) {
```

```
        System.out.println("Type : " + guiEvent.getType());
```

```
        System.out.println("Parameters : " + guiEvent.getAllParameter());
```

```
    }
```

```
}
```



5

Inspection / débogage des agents

Actions sur les agents

 **Start New Agent**

 **Kill**

 **Suspend**

 **Resume**

 **Send Message**

 **Migrate Agent**

 **Clone Agent**

 **Load Agent**

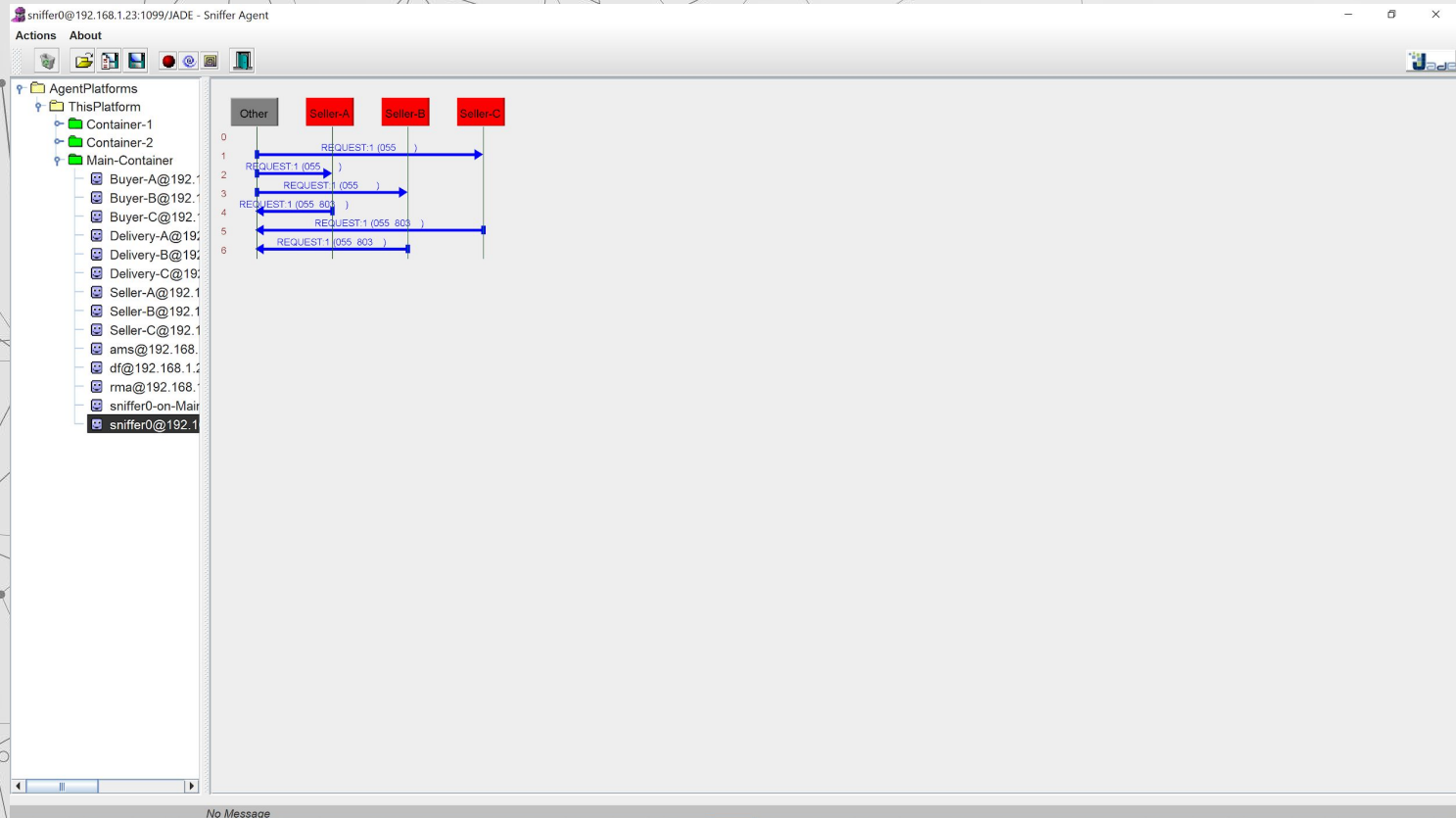
 **Save Agent**

 **Freeze Agent**

 **Thaw Agent**

- Ajouter / supprimer des agents
- Mettre en pause / reprendre des agents
- Envoyer un message à un agent
- Migrer un agent vers un autre container
- Cloner un agent
- Charger / sauvegarder des agents
- Congeler / décongeler des agents

Utilisation du Sniffer



Utilisation du Sniffer

sniffer0@192.168.1.23:1099/JADE - Sniffer Agent

Actions About



entPlatforms

ThisPlatform

Container-1

Customer-A@192.168.1

Customer-B@192.168.1

Customer-C@192.168.1

sniffer0 on Container 1@

Container 2

Banker A @ 102.168.1.22

[illegible]

shinero on container 2@

■ Main Container

Buyer-A@192.168.1.25.

Buyer-B@192.168.1.23.

Buyer-C@192.168.1.23:

Delivery-A@192.168.1.2

Delivery-B@192.168.1.2

Delivery-C@192.168.1.2

☺ Seller-A@192.168.1.23:

☺ Seller-B@192.168.1.23:

☺ Seller-C@192.168.1.23:


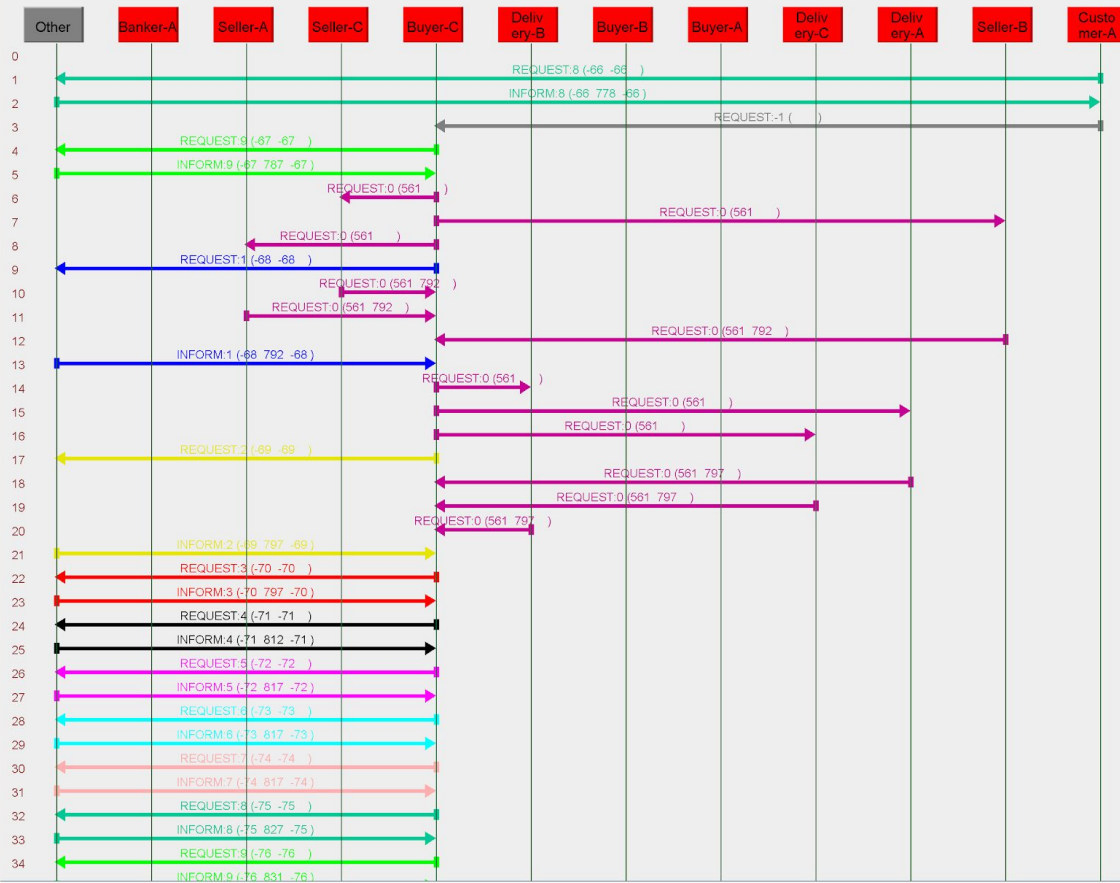
ams@192.168.1.23:1095



```
df@192.168.1.23:1099/
```

A terminal window titled "nc -l -p 1099". The prompt is "nc:". A blue smiley face icon is visible next to the IP address "192.168.1.23". The user input "rma@" is shown, followed by a carriage return character "\n".

rma@192.168.1.23:1099

 sniffer0-on-Main-Contain sniffer0@192.168.1.23:1

Dummy Agent

rma@192.168.1.23:1099/JADE - JADE Remote Agent Management GUI

File Actions Tools Remote Platforms Help

AgentPlatforms

- "192.168.1.23:1099/JADE"
 - Container-1
 - Customer-A@192.168.1.23:1099/JADE
 - Customer-B@192.168.1.23:1099/JADE
 - Customer-C@192.168.1.23:1099/JADE
 - Container-2
 - Banker-A@192.168.1.23:1099/JADE
 - Main-Container
 - Buyer-A@192.168.1.23:1099/JADE
 - Buyer-B@192.168.1.23:1099/JADE
 - Buyer-C@192.168.1.23:1099/JADE
 - Delivery-A@192.168.1.23:1099/JADE
 - Delivery-B@192.168.1.23:1099/JADE
 - Delivery-C@192.168.1.23:1099/JADE
 - Seller-A@192.168.1.23:1099/JADE
 - Seller-B@192.168.1.23:1099/JADE
 - Seller-C@192.168.1.23:1099/JADE
 - ams@192.168.1.23:1099/JADE
 - da1@192.168.1.23:1099/JADE
 - df@192.168.1.23:1099/JADE
 - rma@192.168.1.23:1099/JADE

name	addresses	state	owner
da1@192.168.1.23:1099/JADE - DummyAgent			
General Current message Queued message			
ACLMessage Envelope			
Sender: Set da1@192.168.1.23:1099/JADE			
Receivers:			
Reply-to:			
Communicative act: accept-proposal			
Content:			
Language:			
Encoding:			
Ontology:			
Protocol: Null			
Conversation-id:			
In-reply-to:			
Reply-with:			
Reply-by: Set			
User Properties:			

36

DF GUI

DF df@192.168.1.23:1099/JADE- DF Gui

General Catalogue Super DF Help



Registrations with this DF Search Result DF Federation

Agent name	Addresses	Resolvers
Buyer-A@192.168.1.23:1099/JADE	http://DESKTOP-8NJLKIU.home:7778/acc	
Seller-C@192.168.1.23:1099/JADE	http://DESKTOP-8NJLKIU.home:7778/acc	
Buyer-C@192.168.1.23:1099/JADE	http://DESKTOP-8NJLKIU.home:7778/acc	
Buyer-B@192.168.1.23:1099/JADE	http://DESKTOP-8NJLKIU.home:7778/acc	
Seller-B@192.168.1.23:1099/JADE	http://DESKTOP-8NJLKIU.home:7778/acc	
Seller-A@192.168.1.23:1099/JADE	http://DESKTOP-8NJLKIU.home:7778/acc	
Banker-A@192.168.1.23:1099/JADE	http://DESKTOP-8NJLKIU.home:7778/acc	
Delivery-B@192.168.1.23:1099/JADE	http://DESKTOP-8NJLKIU.home:7778/acc	
Delivery-A@192.168.1.23:1099/JADE	http://DESKTOP-8NJLKIU.home:7778/acc	
Delivery-C@192.168.1.23:1099/JADE	http://DESKTOP-8NJLKIU.home:7778/acc	

Status

Introspector Agent

Introspector0@192.168.1.23:1099/JADE

File About

- Buyer-B@192.168.1.23:
- Buyer-C@192.168.1.23:
- Delivery-A@192.168.1.23:
- Delivery-B@192.168.1.23:**
- Delivery-C@192.168.1.23:
- Introspector0-on-Main-CC:
- Introspector0@192.168.1.23:

AgentPlatforms.ThisPlatform.Main-Container.Deliv

Delivery-B@192.168.1.23:1099/JADE

View State Debug

Current State

Active

Suspended

Idle

Waiting

Moving

Dead

Change St...

Suspend

Wait

Wake Up

Kill

Incoming Messages

Pending

Received

Incoming Messages -- Received

REQUEST

REQUEST

Outgoing Messages

Pending

Sent

Outgoing Messages -- Sent

REQUEST

REQUEST

Behaviours

DeliveryProposalBehavior

Name: DeliveryProposalBehavior

Class: collaborative.platform.behaviors.Deliv

Kind: DeliveryProposalBehavior

Log Manager Agent

JADE Log Manager Agent (log0)

Settings Logs

AgentPlatforms

- "192.168.1.23:1099/JADE"
 - Container-1
 - Container-2
 - Main-Container

Main-Container

Logging system: Java Util Logging

Logger Name	Level	Handlers	Log file
Introspector0@192.168.1.23:1099/JADE	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
_ROOT_LOGGER	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
ams	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
collaborative.platform.agents.BankerAgent	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
collaborative.platform.agents.BuyerAgent	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
collaborative.platform.agents.CustomerAgent	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
collaborative.platform.agents.DeliveryAgent	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
collaborative.platform.agents.SellerAgent	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
df	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
global	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.lang.sl.SL0Ontology	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.lang.sl.SL1Ontology	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.lang.sl.SL2Ontology	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.lang.sl.SL3Ontology	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.onto.BasicOntology	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.onto.Ontology	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.onto.SerializableOntology	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.schema.AgentActionSchema	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.schema.AggregateSchema	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.schema.ConceptSchema	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.schema.ContentElementListSchema	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.schema.ContentElementSchema	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.schema.IRESchema	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.schema.ObjectSchemaImpl	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.schema.PredicateSchema	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.schema.PrimitiveSchema	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.schema.TermSchema	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.content.schema.VariableSchema	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.core.AgentContainerImpl	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.core.ContainerTable	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.core.InternalMessageQueue	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.core.MainContainerImpl	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.core.MainDetectionManager	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.core.MulticastMainDetectionListener	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.core.PlatformManagerImpl	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.core.Runtime	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.core.ServiceManagerImpl	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.core.TimerDispatcher	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.core.VersionManager	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.core.behaviours.FSMBehaviour	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.core.event.Notification	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	
jade.core.management.AgentManagement	INFO	java.util.logging.ConsoleHandler, java.util.logging.Con...	

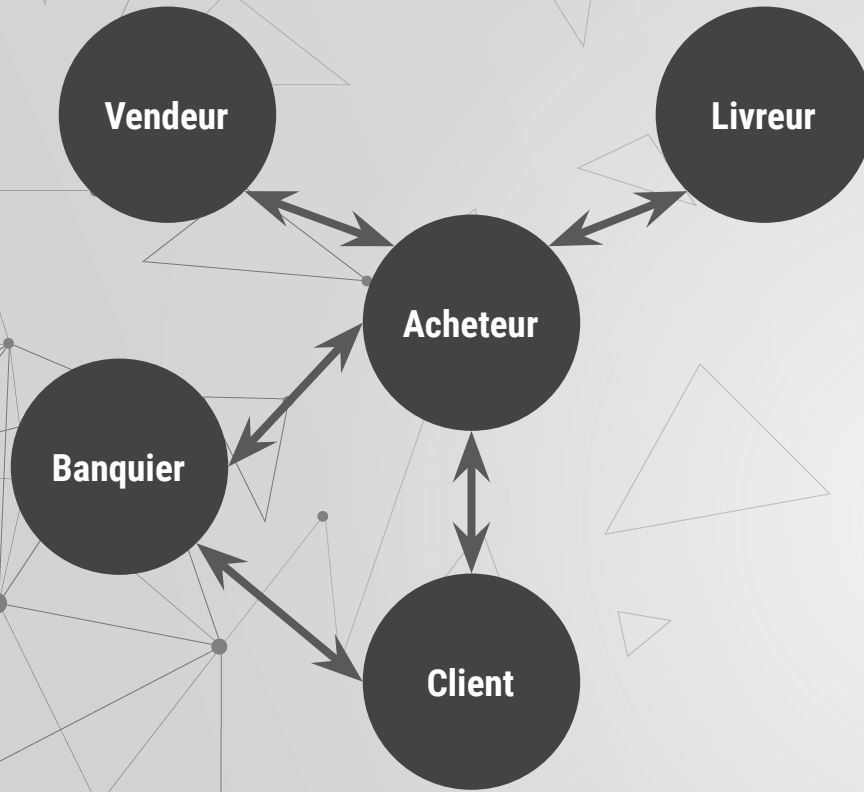


6

Démonstration

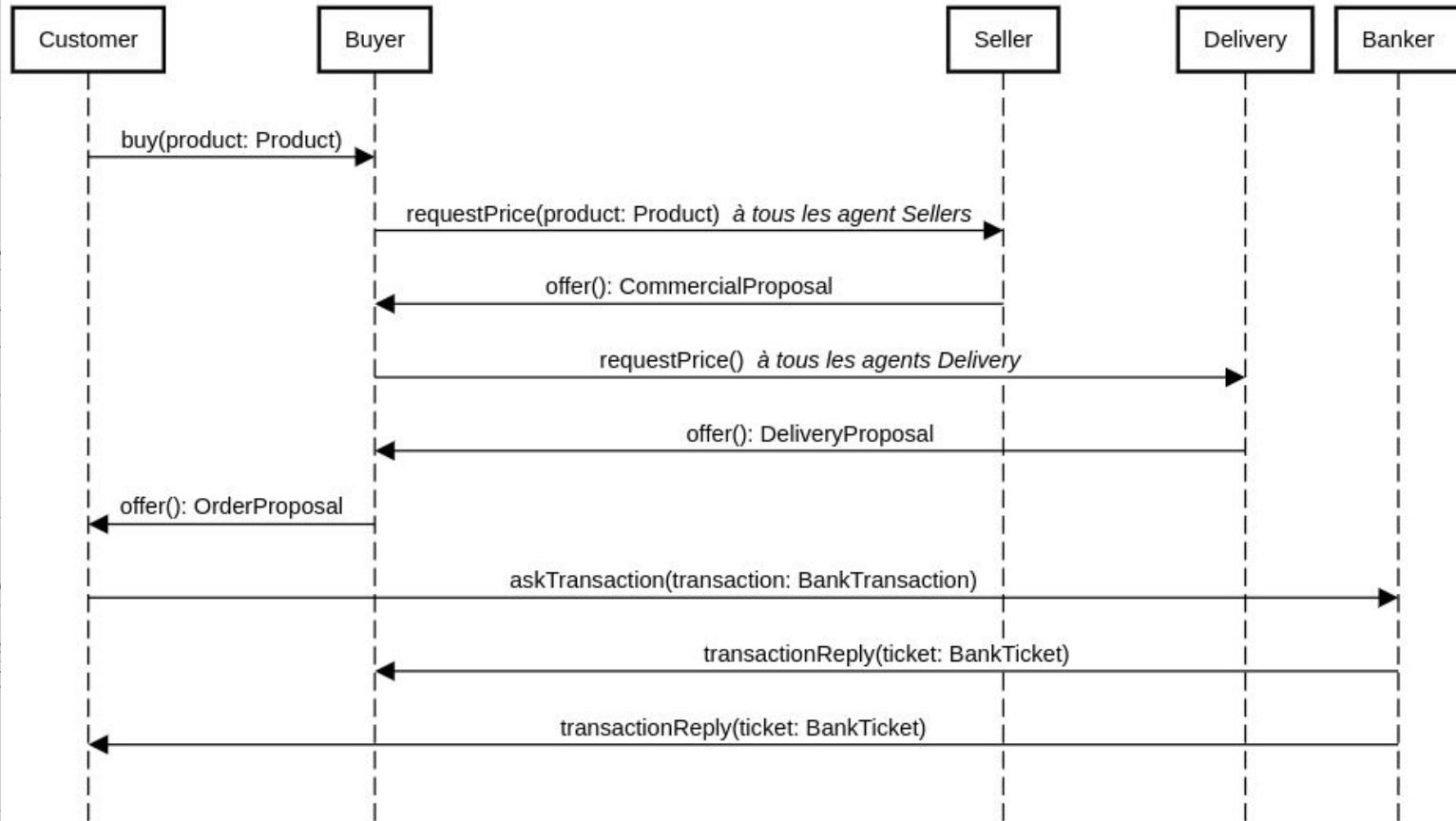
Démonstration d'un SMA

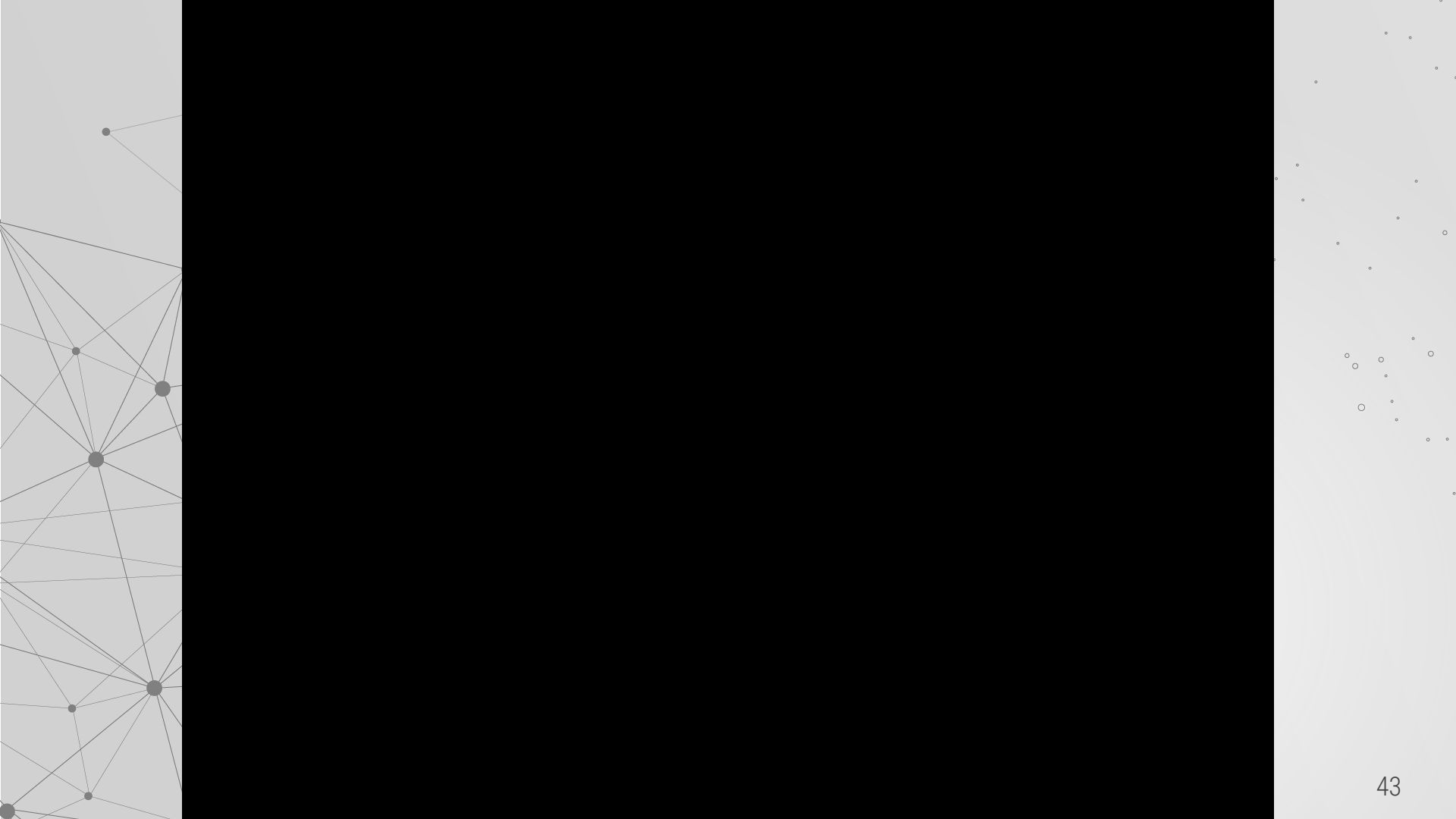
Collaborative Platform



Démonstration d'un SMA

Collaborative Platform





The background is a light gray with abstract geometric patterns. On the left, there is a complex network of thin gray lines connecting small dark gray dots, forming a web-like structure. Scattered across the slide are several thin-lined triangles of various sizes and orientations. In the top right corner, there are small, faint circles. The text "FIN" is centered in the upper half, and "Avez vous des questions ?" is centered below it in a larger font.

FIN

Avez vous des questions ?

Sources

- Developing Multi-Agent Systems with JADE, (31 juillet 2009), Giovanni Caire, Dominic Greenwood.
- <https://jade.tilab.com/>
- https://en.wikipedia.org/wiki/Java_Agent_Development_Framework
- FIPA IEEE Org. <http://fipa.org/index.html>