



**UNIVERSITÉ D'ALGER 1**  
**DÉPARTEMENT INFORMATIQUE**

---

FBPM :

**MINI-PROJET**  
**Conception d'un réseau social d'échange des documents Avec la**  
**méthode SCRUM et Implémentation a l'aide d'un système**  
**multi-Agents**

Docteur : N.chekkai

---

- Réalisé par :
- ACHELI Mahdi
  - ZERMANI Abdelhadi Zakaria

Anee universitaire 2019/2020

<b>1</b>	<b>Conception Avec SCRUM [scrum]</b>	<b>3</b>
1.1	Generalité . . . . .	3
1.1.1	Definition : . . . . .	3
1.1.2	Pré requis recommandés . . . . .	3
1.1.3	Les Rôles en bref . . . . .	3
1.1.4	product backlog . . . . .	4
1.1.5	Réunion de planification de sprint . . . . .	4
1.1.6	Mêlée quotidienne ou « stand-up meeting » . . . . .	4
1.2	Modélisation et analyse avec SCRUM . . . . .	6
1.2.1	Diagramme de Cas d'utilisation . . . . .	6
1.2.2	Diagramme d'interaction . . . . .	7
1.2.3	Diagramme de Séquence Système . . . . .	7
1.2.4	Diagramme de Classes . . . . .	7
1.2.5	Rôles . . . . .	7
1.2.6	Product Backlog . . . . .	7
1.2.7	Description des Sprint . . . . .	7
<b>2</b>	<b>L IMPLIMENTATION DU SYSTEME MULTI-AGENT</b>	<b>9</b>
2.1	Définition . . . . .	9
2.1.1	Agent . . . . .	9
2.1.2	typologie des Agents . . . . .	9
2.1.3	Communication entre les agents . . . . .	9
2.1.4	Système Multi-agent . . . . .	10
2.1.5	Les avantages des SMA . . . . .	10
2.1.6	JADE (Java Agent Développement Framework) . . . . .	10
2.2	Modélisation . . . . .	11
2.2.1	description des entités . . . . .	11
2.2.2	Architecture du système . . . . .	11
2.2.3	Fonctionnement . . . . .	11

<b>A</b>	<b>Guide d'exécution du projet</b>	<b>21</b>
A.1	lancement du mainContainer . . . . .	21
A.2	Lancement des Agents . . . . .	22

# CHAPITRE 1

## CONCEPTION AVEC SCRUM [SCRUM]

### 1.1 Generalité

#### 1.1.1 Definition :

La méthode Scrum (« Scrum » signifie « Mêlée » en anglais), ou plus exactement le cadre méthodologique Scrum est de loin la méthode Agile la plus utilisée dans le monde. Expérimentée en 1993, elle bénéficie aujourd'hui de nombreux retours d'expérience. Les conférences, communautés, formations, blogs, outils et ouvrages à son sujet ne manquent pas. Parler d'une « méthode » concernant Scrum n'est pas ce qu'il y a de plus approprié. Scrum ne se considère pas comme une méthode mais comme un cadre méthodologique. Une méthode dit généralement « comment » faire les choses. Scrum ne dit pas comment réussir son logiciel, comment surmonter les obstacles, comment développer, comment spécifier, etc. Il se contente d'offrir un cadre de gestion de projet Agile (et c'est déjà beaucoup) : des rôles, un rythme itératif, des réunions précises et limitées dans le temps, des artefacts (product backlog, sprint backlog, graphique d'avancement) et des règles du jeu.

#### 1.1.2 Pré requis recommandés

- Blocs de post-it et marqueurs.
- L'ouvrage « Scrum depuis les tranchées ».
- Jeu de cartes ou logiciel de Planning

#### 1.1.3 Les Rôles en bref

1. Le Product Owner : DR. chekkai noura
2. Le Scrum Master :
  - M.Acheli – Z.zermani
3. l'équipe de développement
  - M.Acheli – Z.zermani

#### **1.1.4 product backlog**

La première étape consiste à formaliser la vision du produit (logiciel) que l'on souhaite réaliser. Cette vision décrit les principaux objectifs, jalons, utilisateurs visés. Elle contribuera à guider et fédérer les acteurs du projet. La suite consiste à établir la liste des exigences fonctionnelles et non fonctionnelles du produit. Chaque exigence est ensuite estimée par l'équipe de développement avec la technique de Planning Poker. A la lueur des estimations, la liste ainsi complétée est ordonnancée. Les exigences seront converties en fonctionnalités utilisables selon cet ordonnancement. Le principe étant de convertir en premier les exigences qui apportent le plus de valeur ajoutée (ou ROI) au commanditaire. Il s'agit donc de faire remonter les exigences fonctionnelles de la plus haute valeur ajoutée (ou dont le ROI est le plus élevé) en haut de la liste. Cette liste est appelée le Product Backlog. Le Product Backlog servira à piloter l'équipe de développement et pourra évoluer tout au long du projet. Le changement est non seulement autorisé mais encouragé afin de pouvoir éliminer les idées de départ qui s'avéreront mauvaises et de prendre en compte les nouvelles idées qui arriveront en cours de route. Cette activité de construction du Product Backlog est collaborative, elle implique le Product Owner et l'équipe de développement.

#### **1.1.5 Réunion de planification de sprint**

**Phase 1 : Le « Quoi »**

**Phase 2 : Le « Comment »**

**Graphique d'avancement (Burndown Chart)**

#### **1.1.6 Mêlée quotidienne ou « stand-up meeting »**

c'est une réunion d'équipe organisée quotidiennement pour présenter une mise à jour de l'état à tous les membres d'une équipe de développement. Cette mise à jour d'état semi-temps réel soulève des problèmes potentiels et synchronise les efforts pour éliminer les problèmes difficiles.

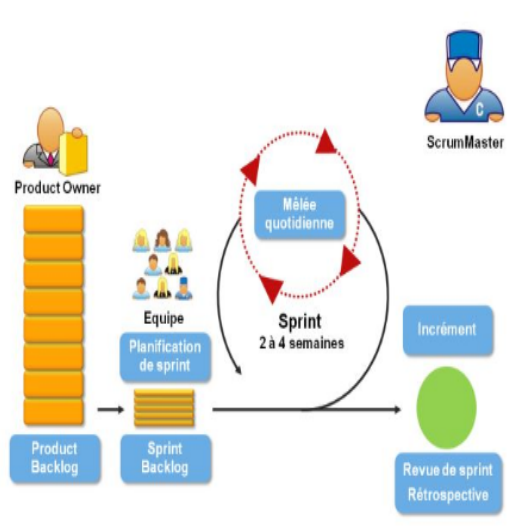


FIGURE 1.1 – Processus Scrum

cette Figure résume le processus SCRUM

## 1.2 Modélisation et analyse avec SCRUM

### 1.2.1 Diagramme de Cas d'utilisation

#### Identification des acteurs

1. User

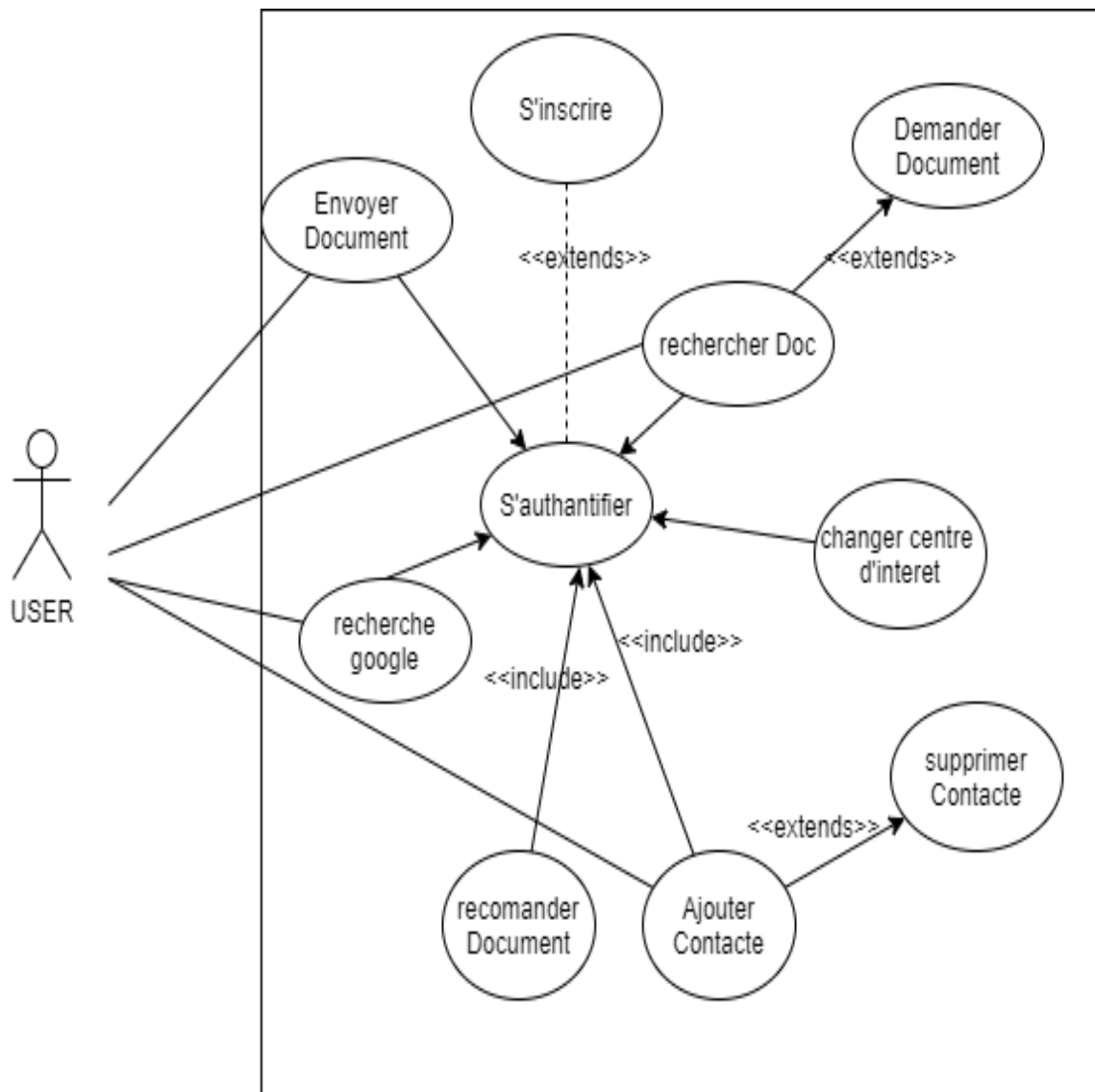


FIGURE 1.2 – Diagramme de cas d'utilisation

### 1.2.2 Diagramme d'interaction

### 1.2.3 Diagramme de Séquence Système

### 1.2.4 Diagramme de Classes

### 1.2.5 Rôles

1. **Product Owner** : DR.chekkai
2. **SCRUM Master** :  
-Zermani -Acheli
3. **Team** :  
-Zermani -Acheli

### 1.2.6 Product Backlog

1. SPRINT 1 :
  - Inscription
  - Authentification
2. SPRINT 2 :
  - Envoyer Document
  - Demander Document
  - Chercher Document
  - Recommander Document
3. SPRINT 3 :
  - Changer centre d'intérêt
  - Ajouter Contacte
  - Supprimer Contacte
4. SPRINT 4 :
  - Identifier le leader d'opinion
  - Requête google

### 1.2.7 Description des Sprint

#### Description du Sprint 1

Sprint 1	user story	en tant que	je veux	priorité
	Inscription	user	m'inscrire	1
	Authentification	user	m'authentifier	2

FIGURE 1.3 – Tableau Descriptif du Sprint 1



### Description du Sprint 2

Sprint 2	user story	en tant que	je veux	priorité
	Envoyer Document	user	Envoyer un document	3
	Demander Document	user	Demander un document	3
	Recommander Document	user	Recommander un document	4
	Rechercher Document	user	Rechercher un Document	3

FIGURE 1.4 – Tableau Descriptif du Sprint 2

### Description du Sprint 3

Sprint 3	user story	en tant que	je veux	priorité
	Changer centre d'interet	user	Changer mon centre d'interet	5
	Ajouter Contact	user	ajouter un contacte	6
	Supprimer Contact	user	supprimer un contacte	6

FIGURE 1.5 – Tableau Descriptif du Sprint 3

### Description du Sprint 4

Sprint 4	user story	en tant que	je veux	priorité
	Identifier le leader d'opinion	user	identifier le leader d'opinion	7
	Requette google	user	lancer une requete google	8

FIGURE 1.6 – Tableau Descriptif du Sprint 4

## CHAPITRE 2

# L IMPLIMENTATION DU SYSTEME MULTI-AGENT

## 2.1 Définition

### 2.1.1 Agent

Un agent est une entite reelle ou virtuelle, evoluant dans un environnement, capable de le percevoir et d'agir dessus, qui peut communiquer avec d'autres agents, qui exhibe un comportement autonome, lequel peut etre vu comme la consequence de ses connaissances, de ses interactions avec d'autres agents et des buts qu'il poursuit.

### 2.1.2 typologie des Agents

1. agent simple re exe : Ce type d'agent choisit ses actions en se basant uniquement sur le percept courant, en ignorant les percepts precedents.
2. agent a etats : plus de ce que fait l'agent a simple re exe, ce type d'agent a un historique ou il enregistre tous ces etats .
3. agent a but : l'agent essaye d'atteindre un "but" donne, sans necessairement connaitre le pourquoi du comment il faut faire pour l'atteindre.
4. agent utilite : c'est une variante de l'agent fonde sur le but, ou ce type d'agent essaie de trouver le plus court chemin.
5. agent apprenant : l'apprentissage permet :
  - De simplifier la conception .
  - a l'agent d'avoir plus de flexibilite.
  - a l'agent d'agir dans des environnements inconnus et de devenir meilleur avec le temps.

### 2.1.3 Communication entre les agents

il existe deux type de communication entre les agents qui sont :

1. communication directe : par envoi de message ou Chaque agent JADE possede une sorte de boite aux lettres qui contient les messages qui lui sont envoyes par les autres agents. Ces boites aux lettres sont sous forme d'une liste qui contient les messages selon l'ordre chronologique de leur arrivee.
2. communication indirecte : avec une zone memoire partagee (Black Board) entre les agents ou si quelqu'un d'eux veut lire ou ecrire il doit a chaque fois consulter cette zone.

#### **2.1.4 Systeme Multi-agent**

Dans la recherche sur l'intelligence artificielle, la technologie des systemes a base d'agents a ete saluee comme un nouveau paradigme pour la conceptualisation, la conception et la mise en oeuvre de sysemes logiciels. Les agents sont des programmes informatiques sophistiques qui agissent de maniere autonome pour le compte de leurs utilisateurs, dans des environnements ouverts et distribues, an de resoudre un nombre croissant de problemes complexes. Cependant, les applications necessitent de plus en plus plusieurs agents pouvant fonctionner ensemble. Un systeme Multi-agents (SMA) est un reseau faiblement couple d'agents logiciels qui interagissent pour resoudre des problemes depassant les capacites individuelles ou la connaissance de chaque systeme expert.

#### **2.1.5 Les avantages des SMA**

une conception d'un logiciel multi-agent a plusieurs avantage que ce soit dans la qualite du logiciel qui est distribue, ou dans la qualite de la solution produit par le logiciel en termes de temps et de precision.

- Un SMA distribue des ressources et des capacites de calcul sur un reseau d'agents interconnectes. Alors qu'un systeme centralise peut etre a ecte par des ressources limitées, des goulots d'etranglement ou des defaillances critiques, un systeme SMA est decentralise et ne soure donc pas du probleme de point de defaillance unique associe aux systemes centra- lises.
- un framework parfait pour la modelisation de certaines meta-heuristique comme les algorithmes de colonie de fourmis.

#### **2.1.6 JADE (Java Agent Développement Framework)**

est une organisation logiciel entièrement implémenté en langage Java. Il simplie la mise en oeuvre des systèmes multi-agents grâce a un middleware conforme aux specifications FIPA et a un ensemble d'outils graphiques prenant en charge les phases de debogage et de deploiement. Un système base sur JADE peut etre distribue sur des machines (qui n'ont même pas besoin de partager le meme systeme d'exploitation) et la configuration peut être contrôlée via une interface graphique distante. La configuration peut même être modifiée au moment de l'exécution en déplaçant les agents d'une machine a une autre, selon les besoins[9].

## 2.2 Modélisation

### 2.2.1 description des entités

1. **User** : l'acteur principale du système qui a la possibilité d'envoyer des documents ou d'effectuer simplement une recherche d'un document
2. **Serveur** : est un Agent tres utile qui gère l'authentification des User
3. **DF** : est un Agent qui permet a chaque user de publier ses Documents
4. **AMS** : est un agent implicite de la plateforme JADE qui Attribue un AID unique a chaque Agent

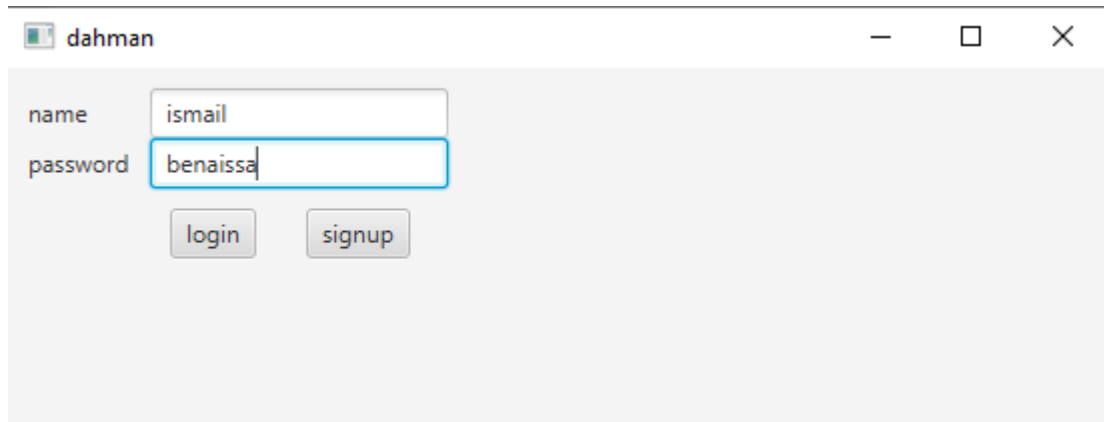
### 2.2.2 Architecture du systeme

### 2.2.3 Fonctionnement

#### Description de quelques scénarios réels

##### 1. Scénario 1 :Authentification

un agent veux d'authentifier le user rempli le formulaire d'authentification , et envoi une requête d'authentification au serveur, a la réception de la requête le serveur vérifie le nom et le mot de passe et renvoi un message soit "echec" soit "accept", le message "accept" contient aussi la liste de ses contacts (selon le centre d'intérêt choisi) on remarque ici que l'agent peut avoir plusieurs comptes



The image shows a graphical user interface for authentication. The window has a title bar with the name 'dahman' and standard minimize, maximize, and close buttons. The main area contains a form with two text input fields. The first field is labeled 'name' and contains the text 'ismail'. The second field is labeled 'password' and contains the text 'benaissa'. Below these fields are two buttons: 'login' and 'signup'.

FIGURE 2.1 – l'interface d'authentificationn sous notre App

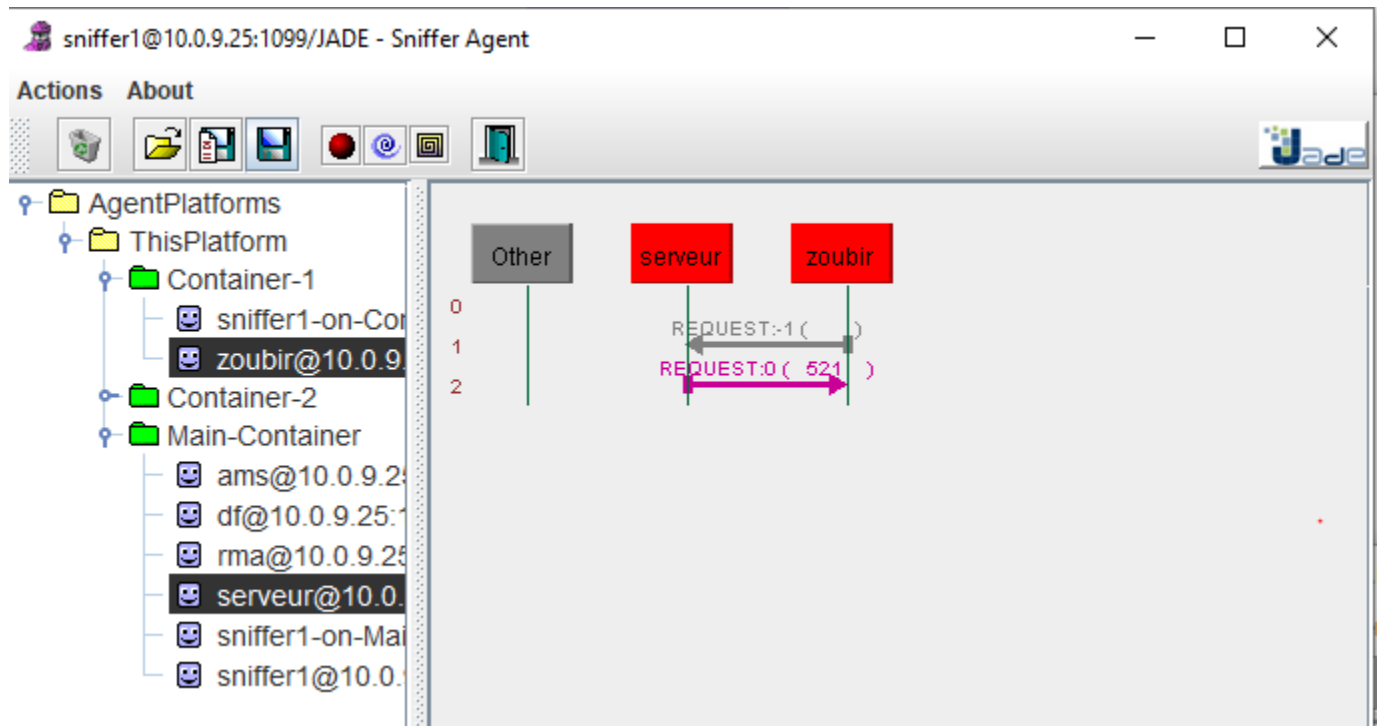


FIGURE 2.2 – visualisation du sniffer

## 2. Scénario 2 :ajouter un document

pour ajouter un document l'agent va utiliser le DF, il va publier un service "le nom du fichier" dans la page jaune du df.

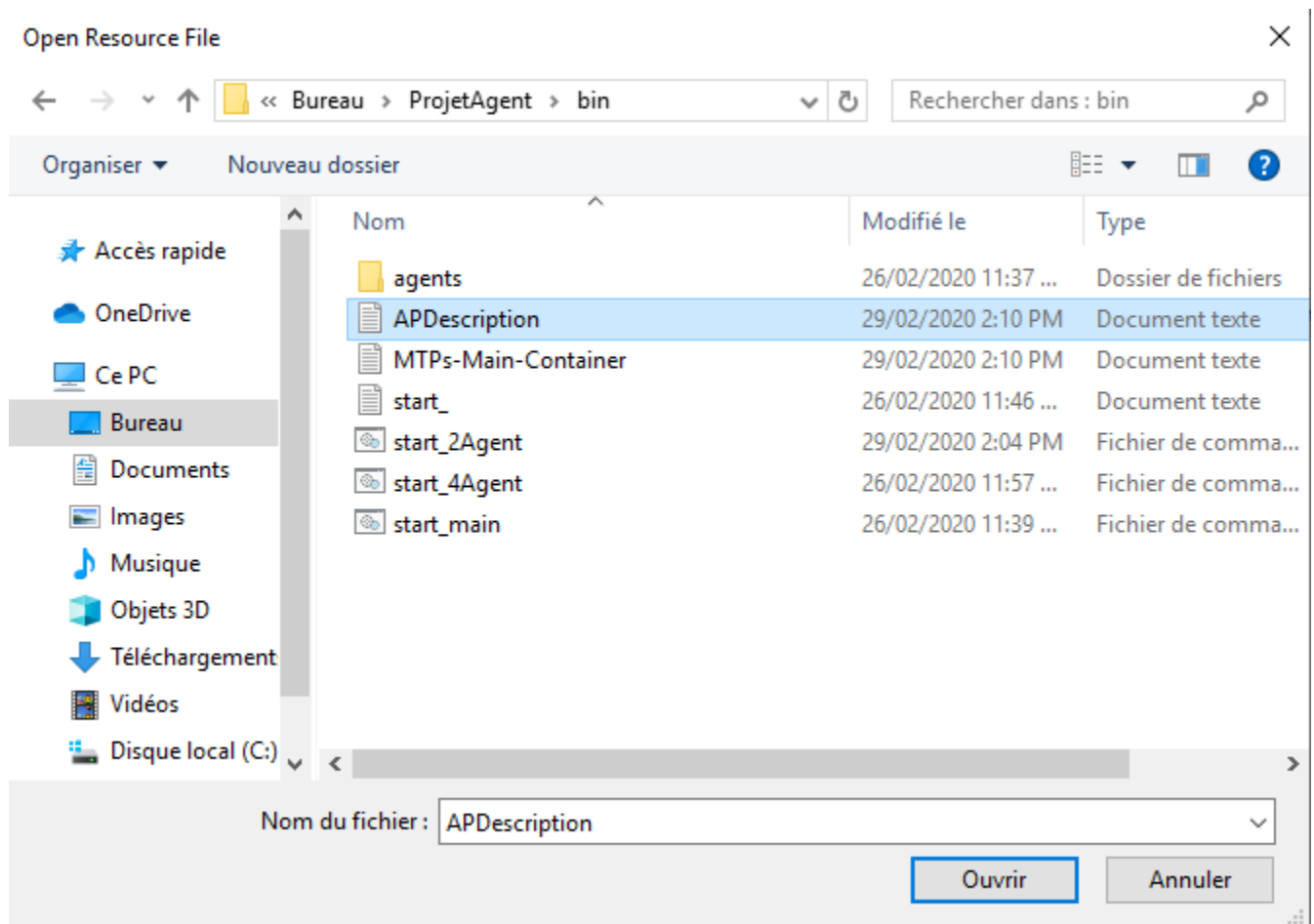


FIGURE 2.3 – Selection du document a ajouter

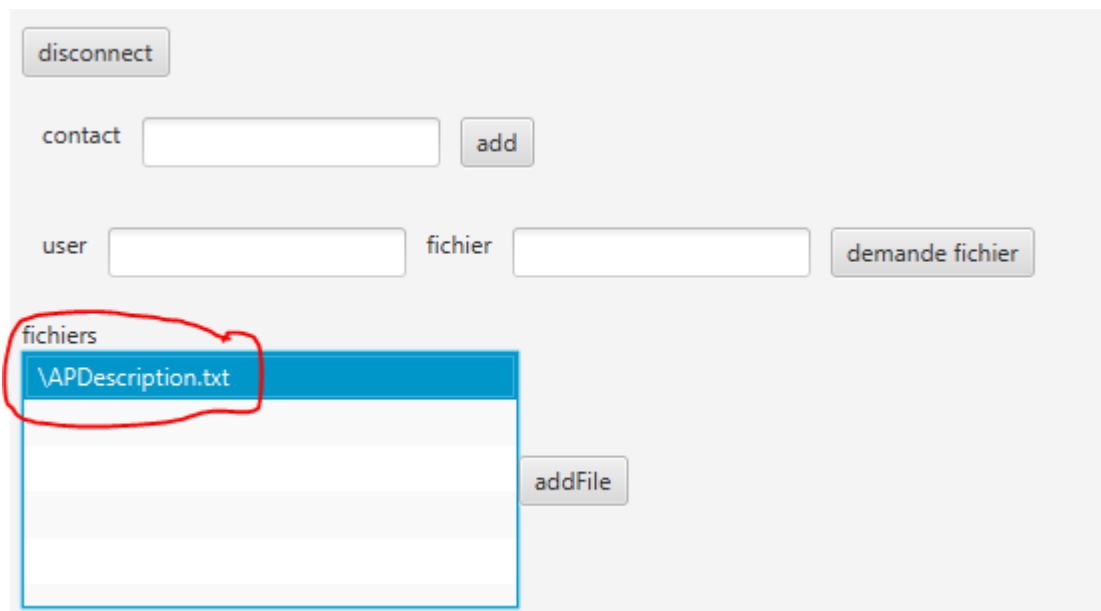


FIGURE 2.4 – le document a ete ajouté

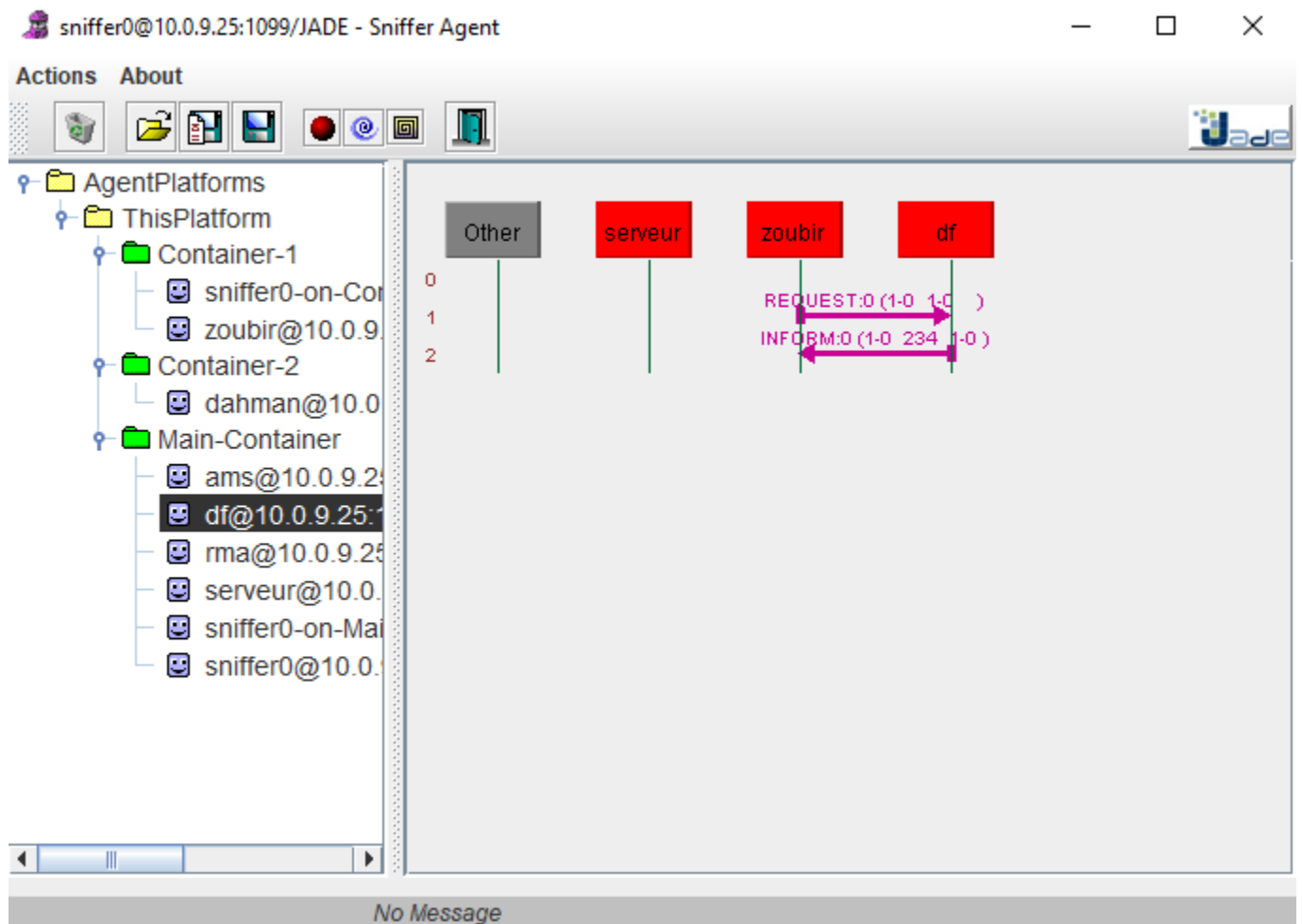


FIGURE 2.5 – visualisation du sniffer

### 3. Scénario 3 : un agent veut envoyer un document

le user sélectionne un document a envoyer qu'il a déjà ajouter, et sélectionne un contacte dans sa liste pour l'envoyer, il envoi en premier une requête au serveur , en lui donnant le nom complet du contacte, il va lui rendre le AID de l'agent connecté a partir de ce compte, a la réception de l'AID , l'agent envoi le document directement a l'agent cible.



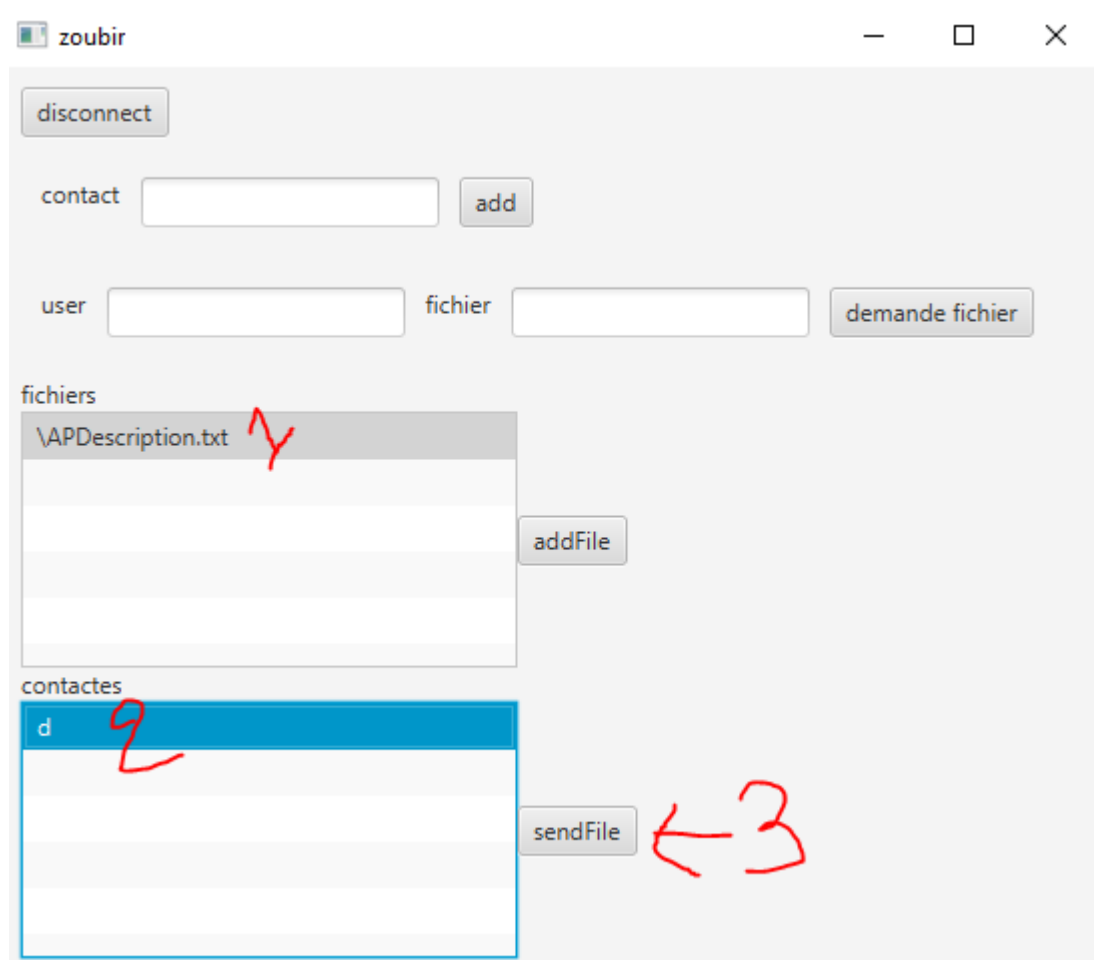


FIGURE 2.6 – l'interface d'envoi d un document

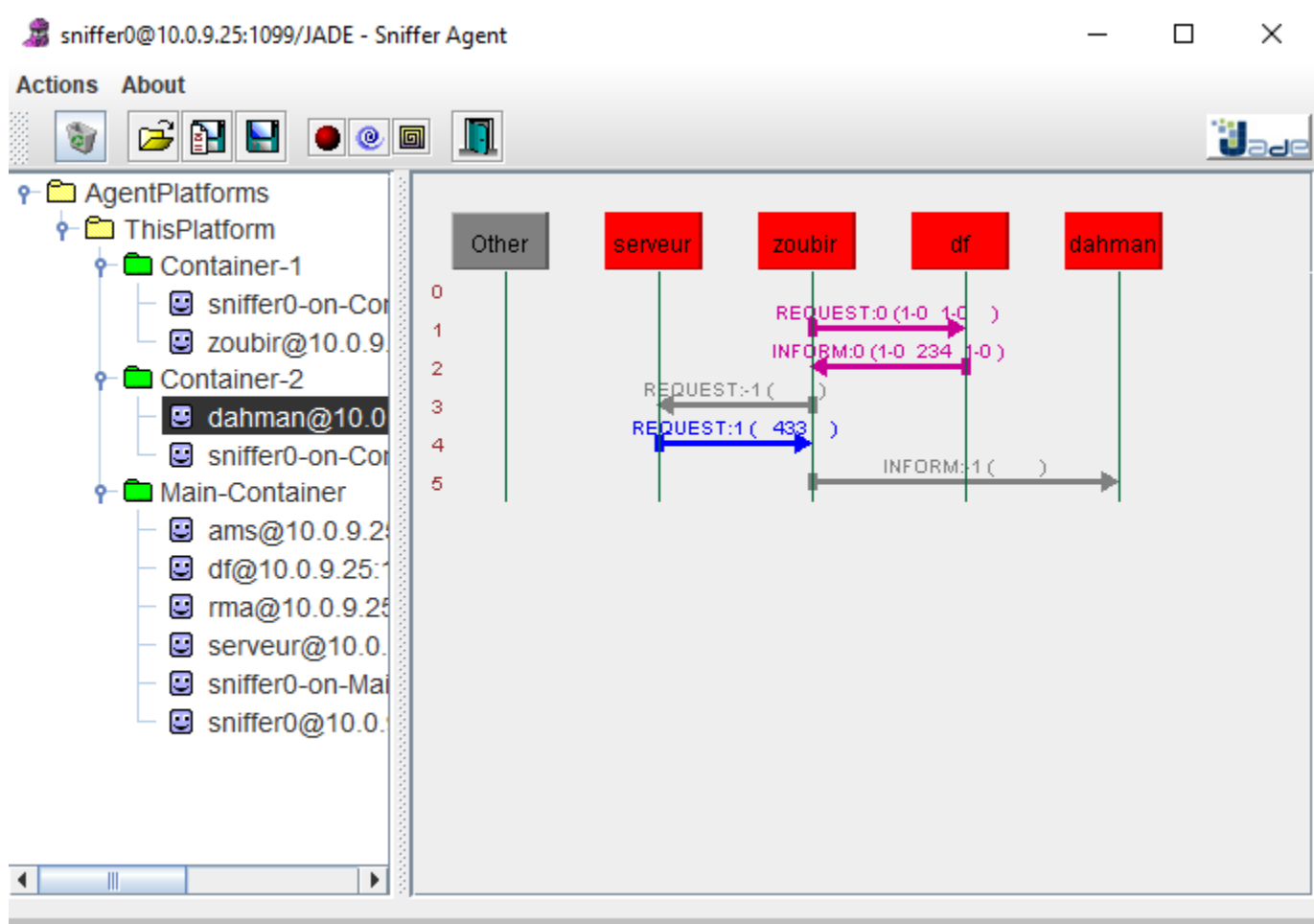


FIGURE 2.7 – visualisation du sniffer

#### 4. Scénario 4 : demander un document

le user ecrit le nom du fichier à demander et le nom du contacte à qui il veut demander, il envoi en premier une requête au serveur , en lui donnant le nom complet du contacte, il va lui rendre le AID de l'agent connecté a partir de ce compte, a la réception de l'AID , l'agent envoi la demande directement a l'agent cible.

A screenshot of a web interface with a light gray background. At the top left is a button labeled "disconnect". Below it is a "contact" label followed by an empty text input field and an "add" button. Further down is a "user" label followed by a text input field containing the letter "z", with a red handwritten "1" and an arrow pointing to it. To the right of the "user" field is a "fichier" label followed by a text input field containing "image.png", with a red handwritten "2" and an arrow pointing to it. To the right of the "fichier" field is a button labeled "demande fichier", with a red handwritten "3" and an arrow pointing to it.

FIGURE 2.8 – LE fichier a demandé

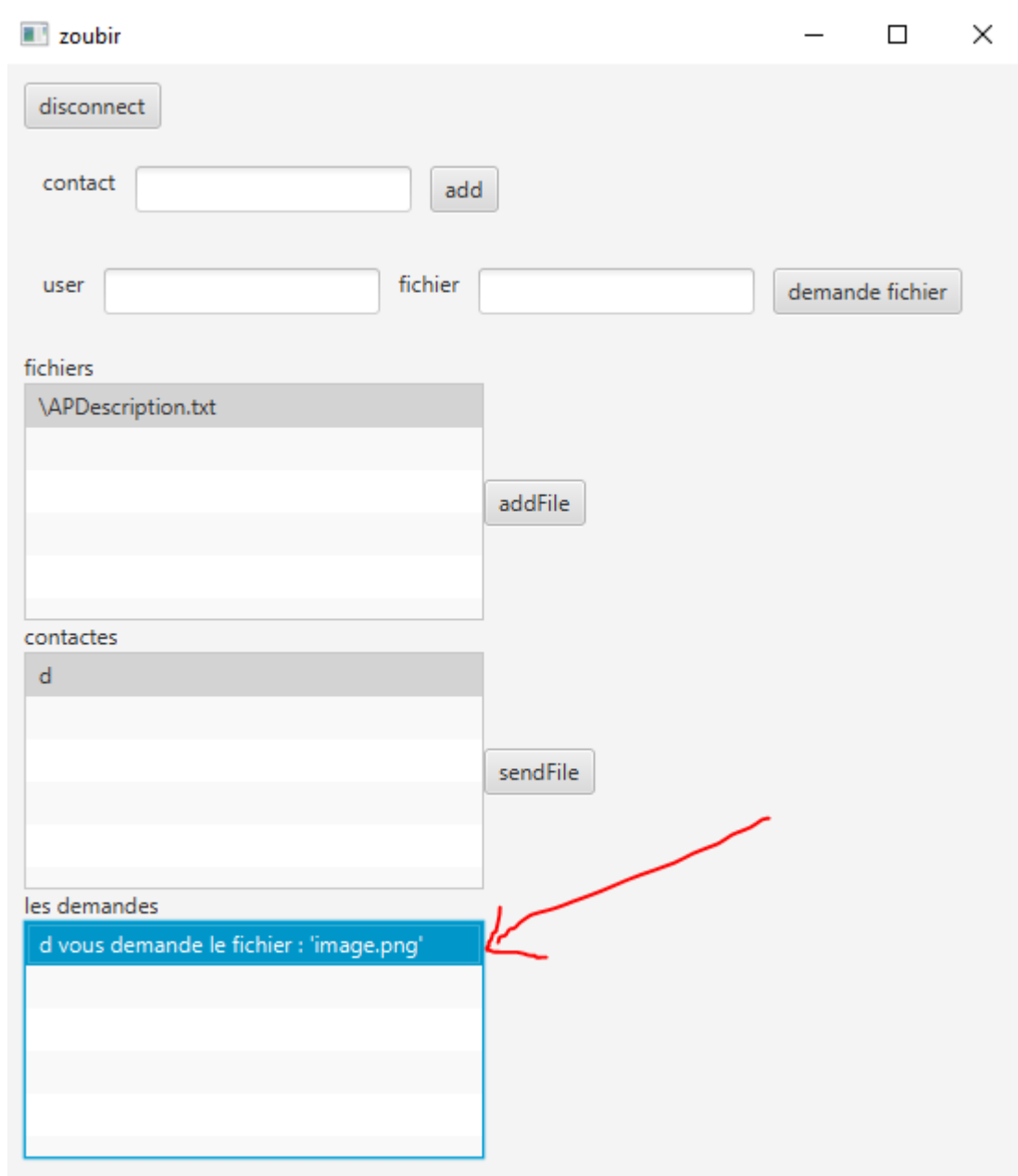


FIGURE 2.9 – la demande est reçu par le destinataire

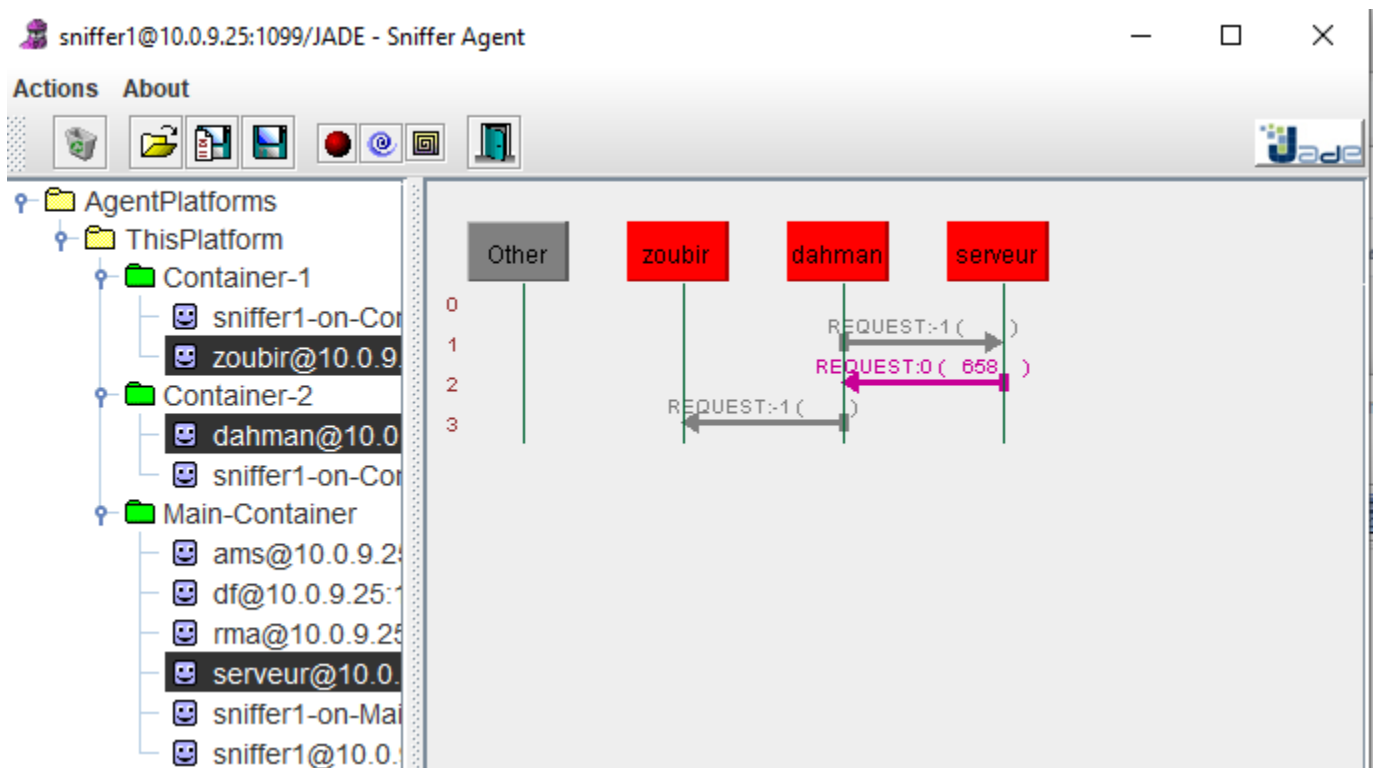


FIGURE 2.10 – visualisation du sniffer

# ANNEXE A

## GUIDE D'EXÉCUTION DU PROJET

pour lancer le projet i suffit donc de suivre les etapes suivantes

### A.1 lancement du mainContainer

pour lancer le main container il suffit just de lancet le script\_start main.bat du dossier bin de notre projet.

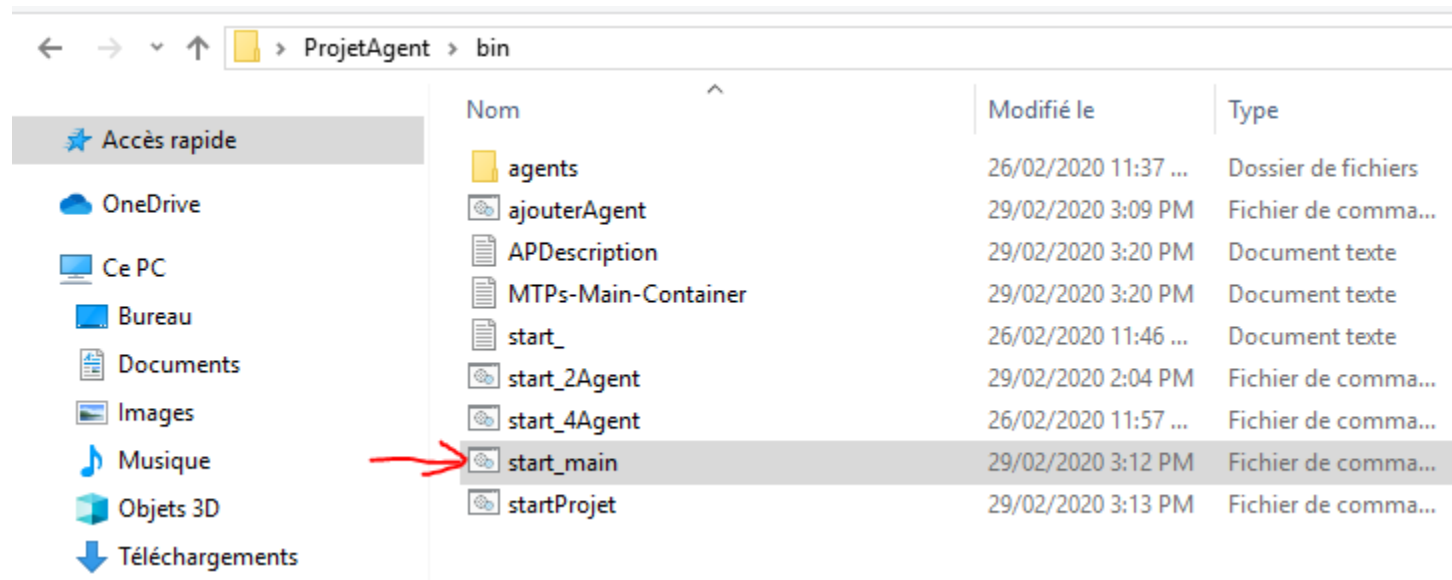
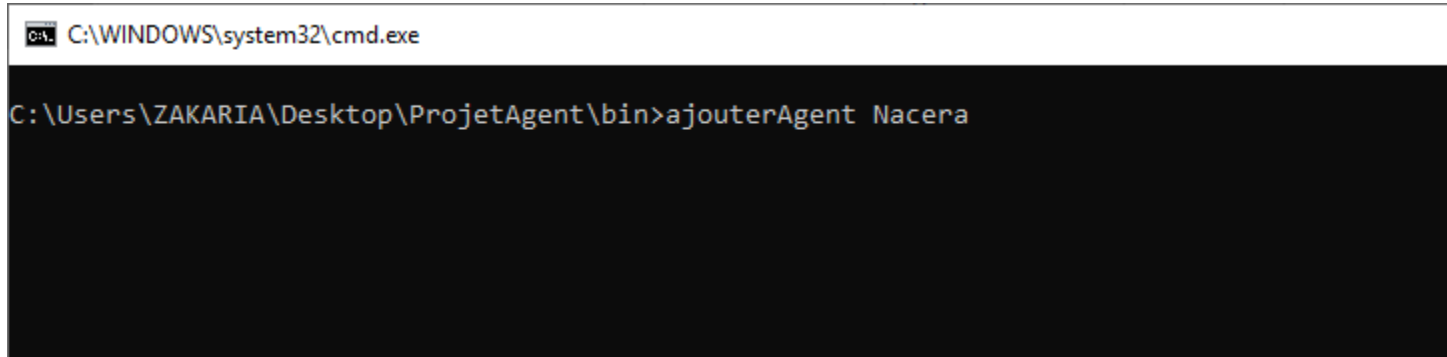


FIGURE A.1 – Lancement de main container

## A.2 Lancement des Agents

on a la possibilité de lancer autant d'agents qu'on vaudra, il suffit juste de ouvrir le cmd vers le chemin du dossier bin. Dans la fenetre de commande qui s'affiche (cmd) ecrire la ligne suivante "ajouterAgent [nom de l'agent]" autant de fois que vous voulez.



```
C:\WINDOWS\system32\cmd.exe  
  
C:\Users\ZAKARIA\Desktop\ProjetAgent\bin>ajouterAgent Nacera
```

FIGURE A.2 – lancement d'un agent