

# PROJET 2A

*INTERCEPTION SYSTÈME POUR  
L'ÉMULATION D'APPLICATIONS*



**Tuteurs laboratoire**  
Lucas NUSSBAUM  
Martin QUINSON

**Responsable école**  
Phuong Le Hong

# PLAN

## I. Gestion de projet

- 1) Contexte et objectif
- 2) Diagramme de Gantt
- 3) Difficultés rencontrées

## II. Réalisations

- 1) Des applications distribuées...
- 2) ... espionnées à leur insu : l'intercepteur

## III. Apport à notre formation

- 1) En compléments des enseignements
- 2) Expérience de recherche

# PLAN

## I. Gestion de projet

- 1) Contexte et objectif
- 2) Diagramme de Gantt
- 3) Difficultés rencontrées

## II. Réalisations

- 1) Des applications distribuées...
- 2) ... espionnées à leur insu : l'intercepteur

## III. Apport à notre formation

- 1) En compléments des enseignements
- 2) Expérience de recherche

# GESTION DE PROJET

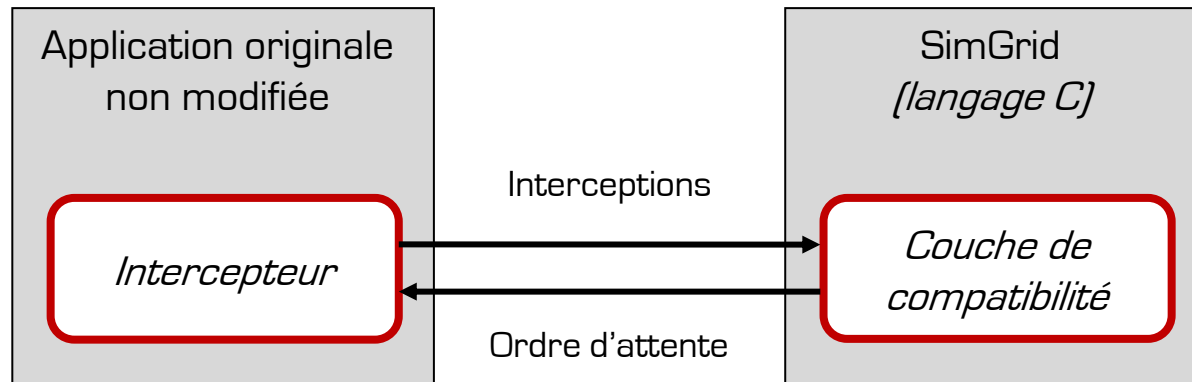
## → CONTEXTE ET OBJECTIFS

- Utilité des applications distribuées
- Inconvénients des expériences *in situ*
  - Nécessité des simulations
- Inconvénients des simulations
  - Emuler des applications

# GESTION DE PROJET

## → CONTEXTE ET OBJECTIFS

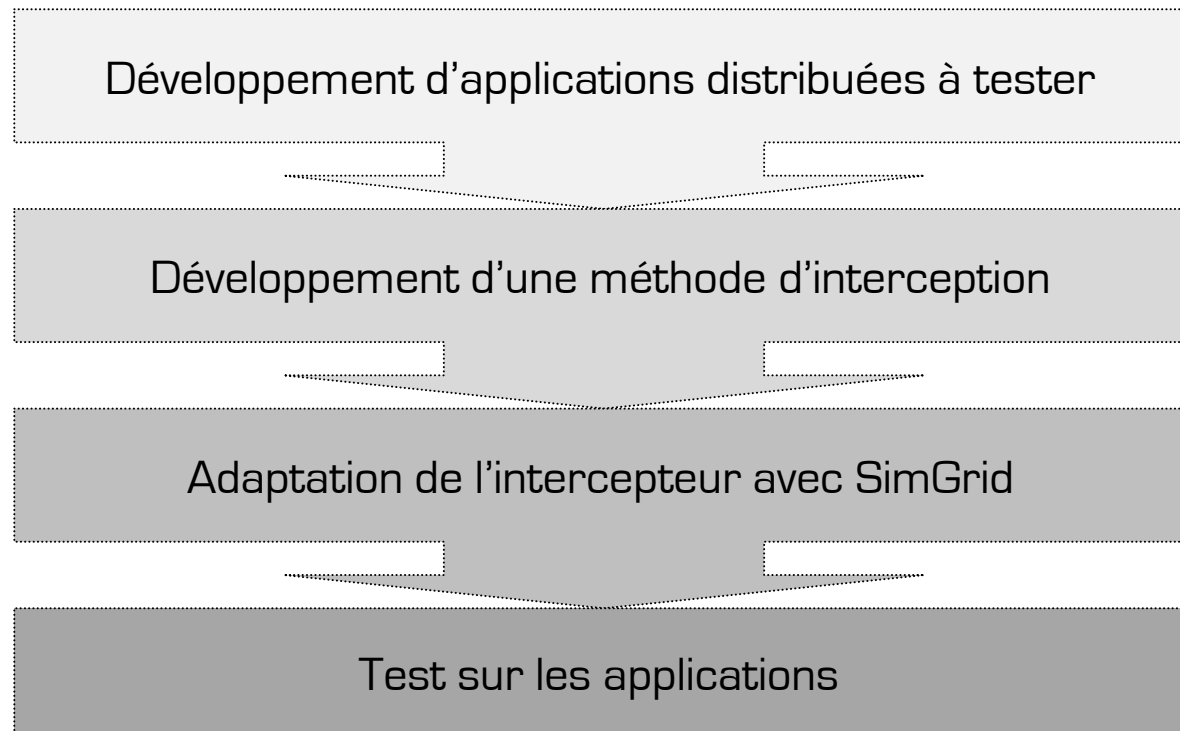
- Principe du projet



# GESTION DE PROJET

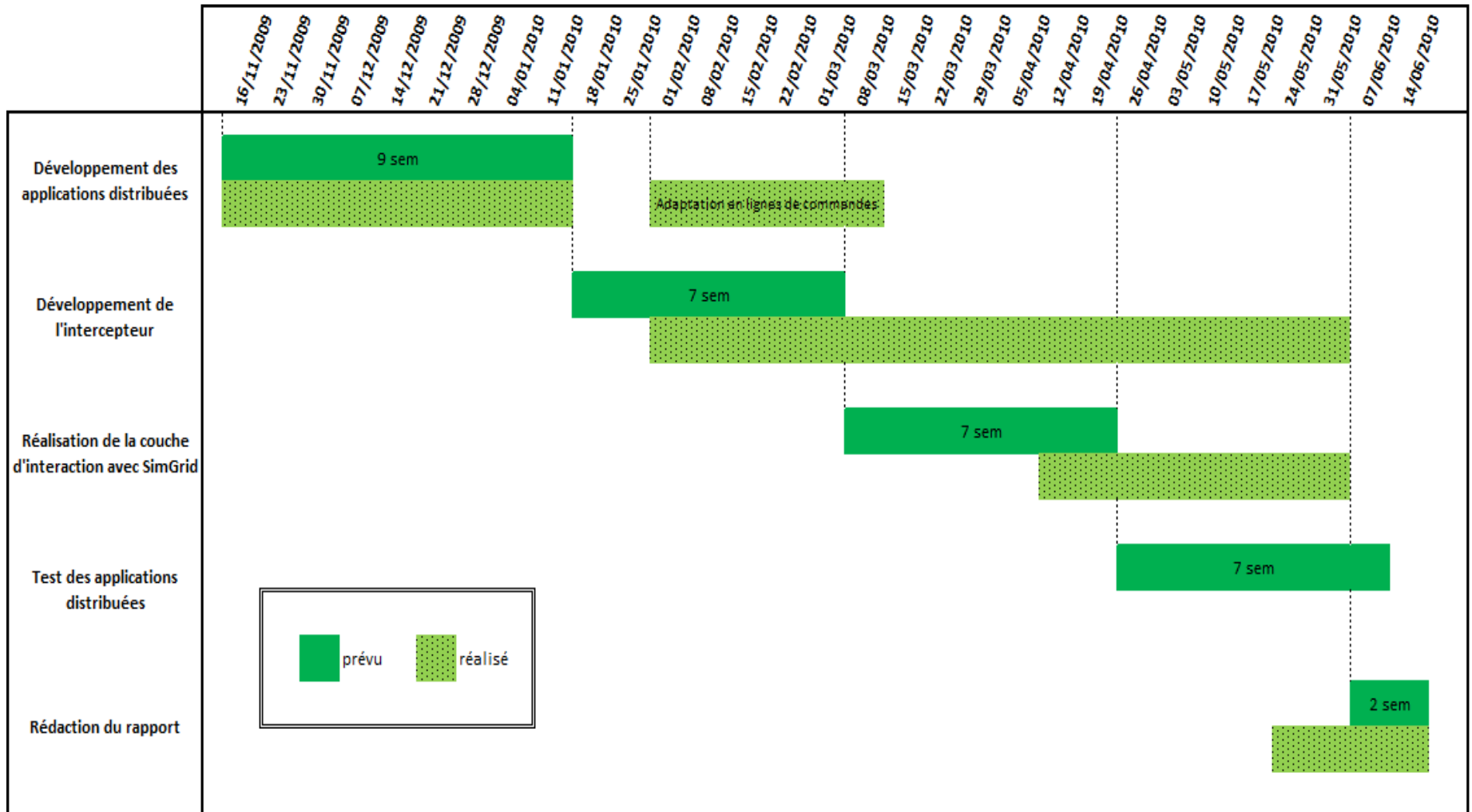
## → *CONTEXTE ET OBJECTIFS*

- Les 4 phases du Cahier des Charges



# GESTION DE PROJET

## → DIAGRAMME DE GANTT



# GESTION DE PROJET

## → *DIFFICULTÉS RENCONTRÉES*

- Des retards...
- Technicité
- Peu de documentation



# PLAN

## I. Gestion de projet

- 1) Contexte et objectif
- 2) Diagramme de Gantt
- 3) Difficultés rencontrées

## II. Réalisations

- 1) Des applications distribuées...
- 2) ... espionnées à leur insu : l'intercepteur

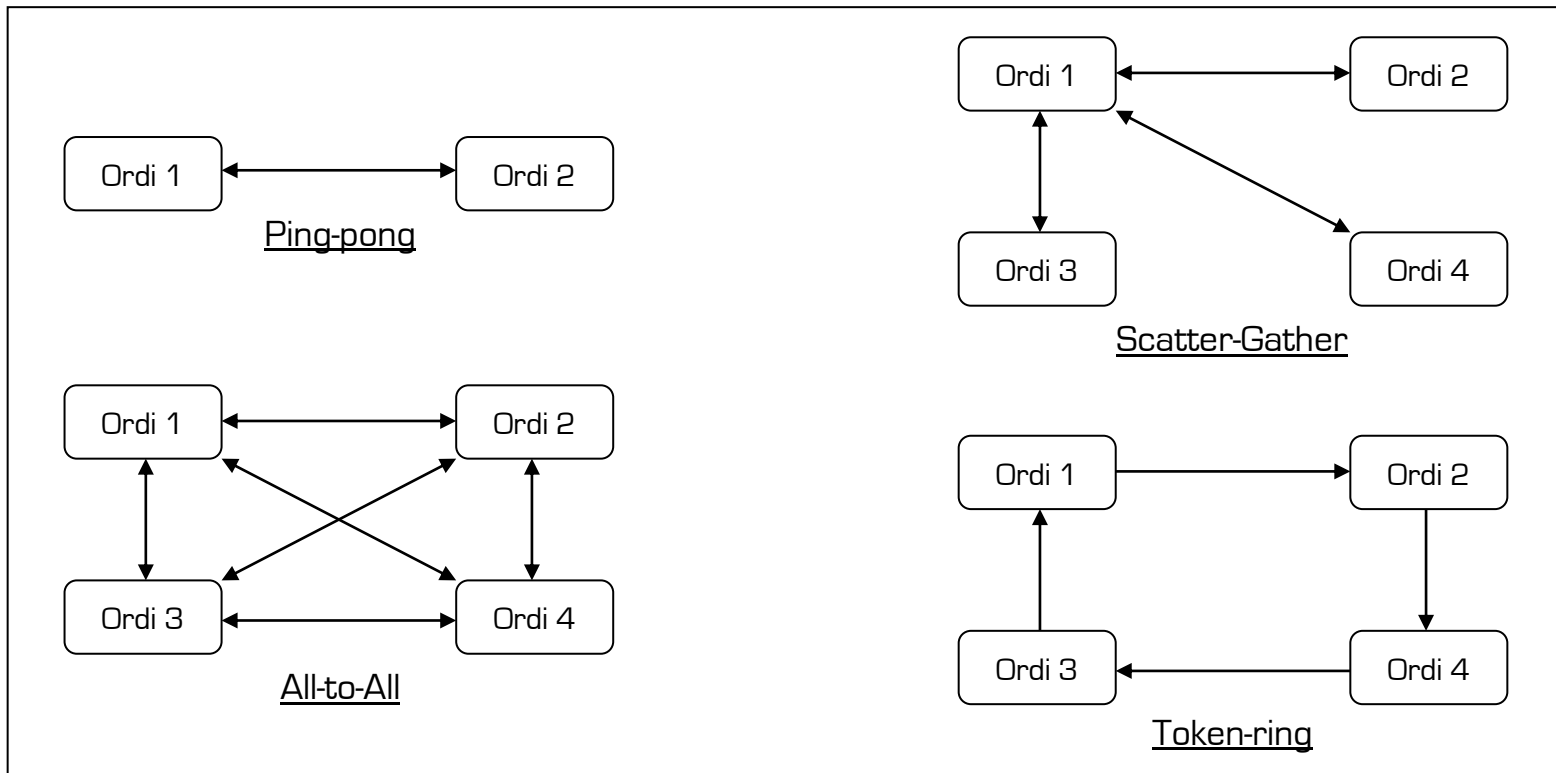
## III. Apport à notre formation

- 1) En compléments des enseignements
- 2) Expérience de recherche

# RÉALISATIONS

## → APPLICATIONS DISTRIBUÉES

- 4 exemples différents



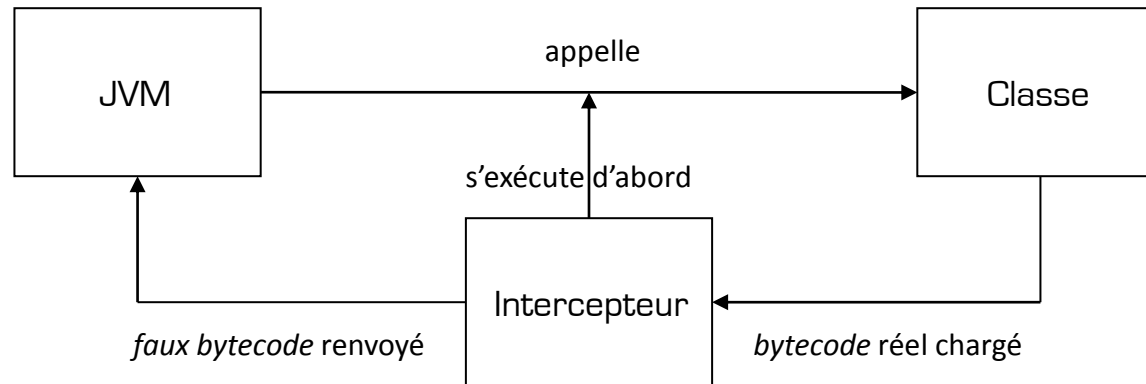
# RÉALISATIONS

## → INTERCEPTEUR

- Intercepter : pourquoi ?
  - Dimensionnement des plateformes
  - Connaitre le temps de calcul
- Intercepter : pour qui ?
  - Invisibilité et transparence d'exécution
  - Portabilité, robustesse
- Intercepter : comment ?
  - Les agents Java : modifier un *bytecode* à la volée
  - LibASM : un décompilateur

# RÉALISATIONS

## → INTERCEPTEUR : PRINCIPE



- Fichier : Test.class
  - Chargée puis immédiatement interceptée
- Fichier : MyTest.class
  - Classe réellement chargée

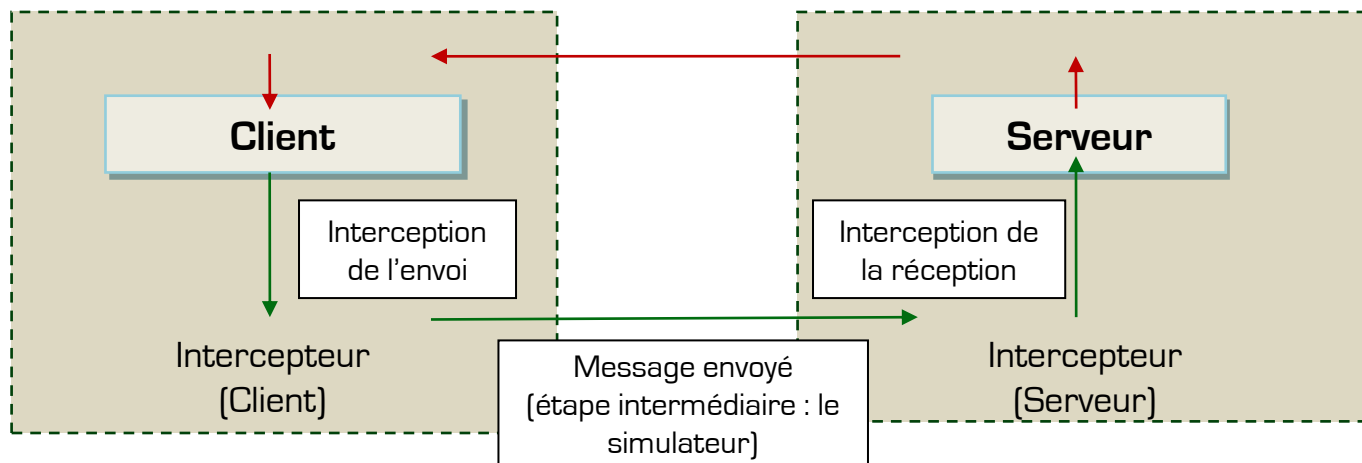
# RÉALISATIONS

## → INTERCEPTEUR : PRINCIPE

Fichier	Test.class	MyTest.class	RealTest.class
Nom de classe	Test	Test	RealTest
Bytecote chargé	MyTest.class	MyTest.class	Test.class
Appelle		RealTest.class	Test.class renommée

# RÉALISATIONS

## → INTERCEPTEUR : CAS PRATIQUE



- Détermination du temps de calcul
- SimGrid intervient entre les 2 interceptions
- Temps virtuel modifié

# PLAN

## I. Gestion de projet

- 1) Contexte et objectif
- 2) Diagramme de Gantt
- 3) Difficultés rencontrées

## II. Réalisations

- 1) Des applications distribuées...
- 2) ... espionnées à leur insu : l'intercepteur

## III. Apport à notre formation

- 1) En compléments des enseignements
- 2) Expérience de recherche

# APPORT À NOTRE FORMATION

→ *EN COMPLÉMENTS DES ENSEIGNEMENTS*

- Compétences techniques
- Gestion d'un projet
  - Estimations des besoins
  - Gestion et évolution des priorités
- Raisonnement professionnel
  - Portabilité et robustesse des applications
  - Transmission des savoirs



# APPORT À NOTRE FORMATION

## → *EXPÉRIENCE DE RECHERCHE*

- Pas de garantie de délais
  - Difficulté de prévoir un échéancier
- 1<sup>ère</sup> expérience de recherche
  - Méthode de travail différente (documentation)

# CONCLUSION

**Merci de votre attention !**