Informatique Repartie

Présentation Projet 2021/2022

Cecilia Zanni-Merk

cecilia.zanni-merk@insa-rouen.fr

Bureau BO B R1 04

Nouveau planning

- 31 janvier 2022 : CM + TD
  - Consignes pour le développement des projets + TD RMI
- 21 février 2022 : TD groupé + TD
  - Finalisation cahiers spéc et conception + Début préparation techno 1
- 28 février 2022 : CM + TD
  - QCM Techno 1 + TD Techno 1
- 7 mars 2022 : CM + TD
  - QCM Techno 2 + TD Techno 2

Nouveau planning

- 31 janvier 2022 : CM + TD
  - Consignes pour le développement des projets + TD RMI
- 21 février 2022 : TD groupé + TD
  - Finalisation cahiers spéc et conception + Début préparation techno 1
- 28 février 2022 : CM + TD
  - QCM Techno 1 + TD Techno 1
- 7 mars 2022 : CM + TD
  - QCM Techno 2 + TD Techno 2
- Plus d’autres surprises à venir …

Agenda

- Rappels UML
- Démarche de mise en œuvre pour le projet
- Livrables
- Présentation du projet
- Organisation et deadlines

Rappels UML

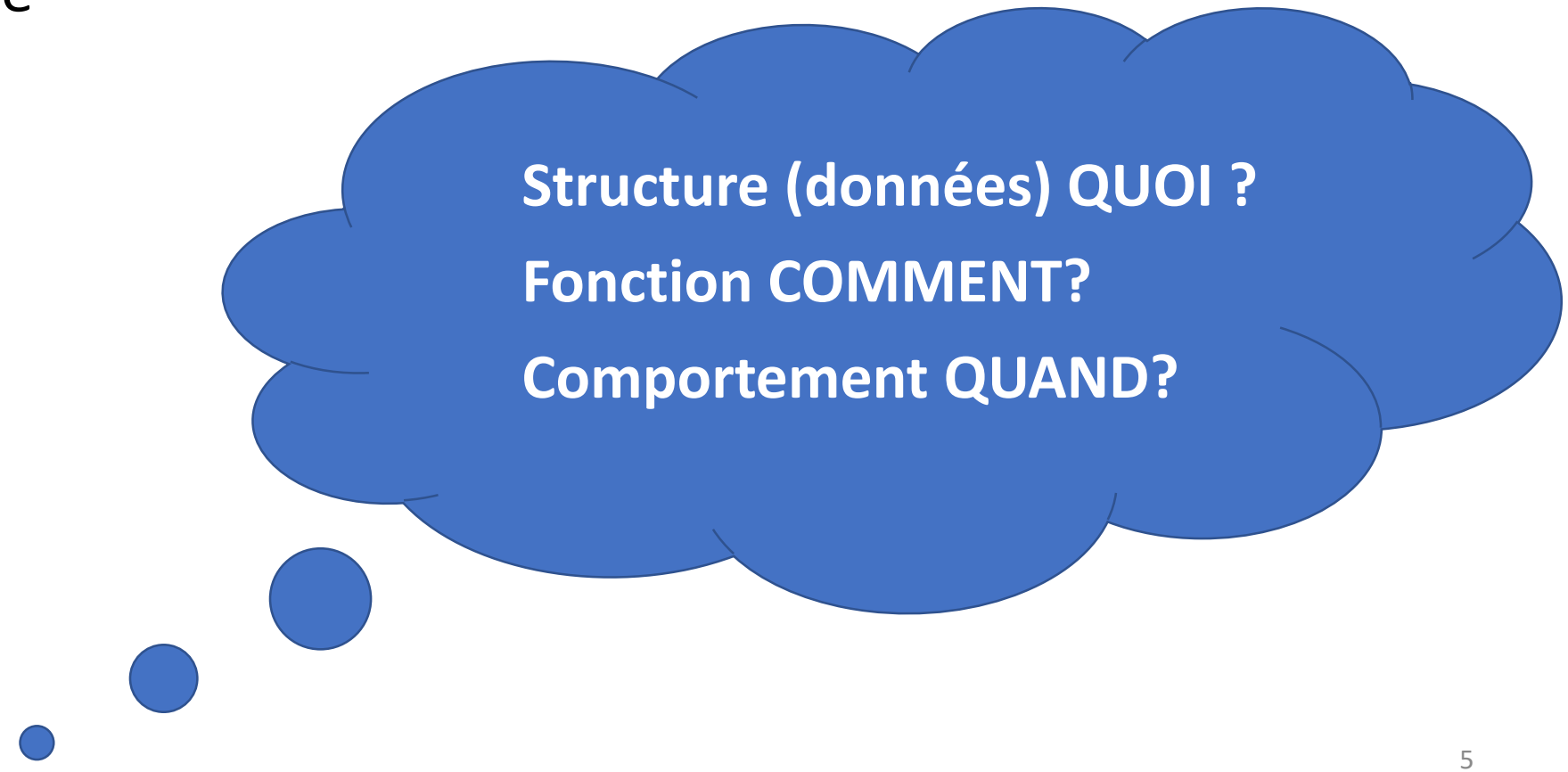
Très rapidement

Rappels UML

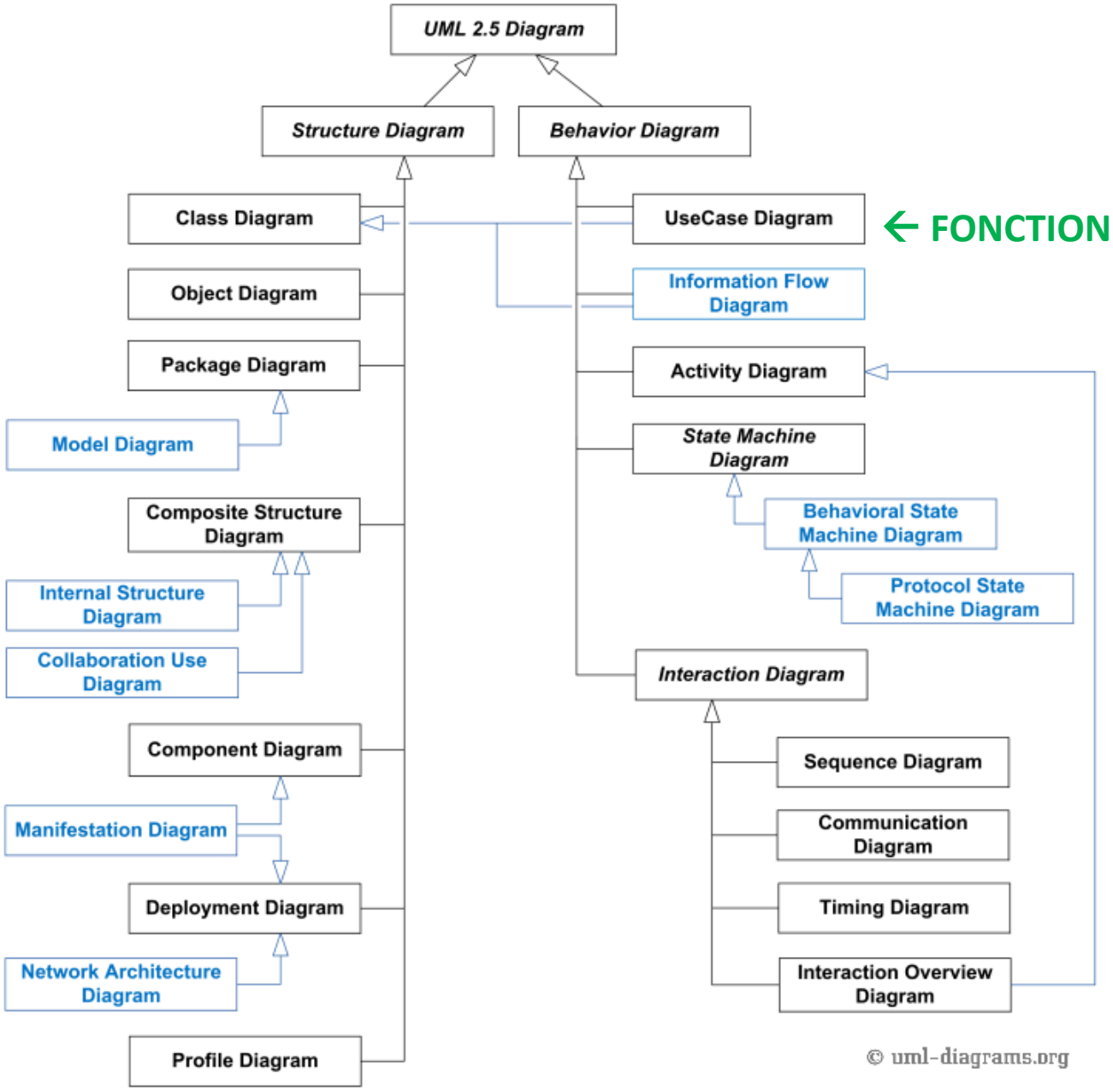
- UML permet d’ exprimer **la structure**, **la fonction** et **le comportement** d'un programme

Rappels UML

- UML permet d’ exprimer **la structure**, **la fonction** et **le comportement** d'un programme



Rappels UML 2.5



← FONCTION

UML 2.5 propose 14 diagrammes standard

- Sept diagrammes de structure
  - De classes
  - De structure composite
  - D’objets
  - De composants
  - De paquetages
  - De déploiement
  - De profils
- Sept diagrammes de comportement
  - De cas d’utilisation
  - D’interaction
  - D’activités
  - De communication
  - De séquence
  - De temps

UML 2.5 propose 14 diagrammes standard

- Sept diagrammes de structure
  - De classes ← spéc / conception
  - De structure composite
  - D'objets
  - De composants ← conception
  - De paquetages ← conception
  - De déploiement ← conception
  - De profils
- Sept diagrammes de comportement
  - De cas d'utilisation ← spéc
  - De séquences ou d'interaction ← conception
  - D'activités
  - De communication
  - De séquences système ← spéc
  - De temps

Diagrammes UML utiles en InfoRep

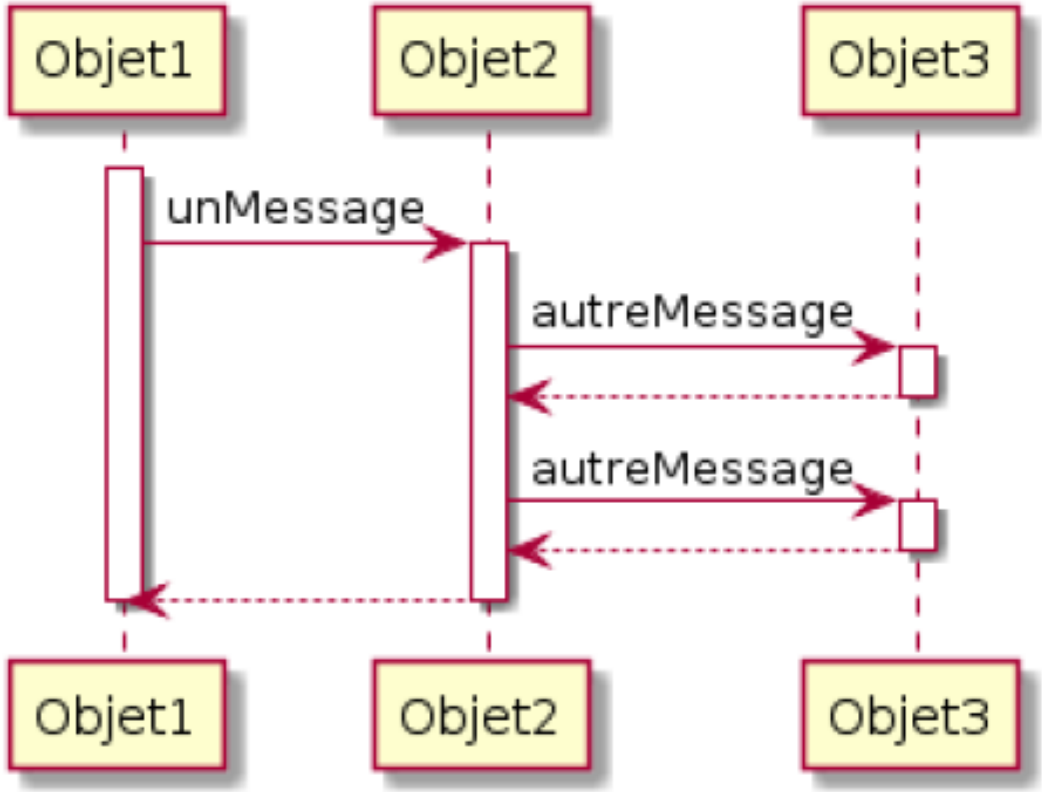
- **De composants :** liste les composants logiciels
- **De déploiement** : définit la répartition des composants sur une architecture matérielle
- **D’interaction**
  - **De séquence système**: représente les scenarios d'interactions entre acteurs et le système, un par cas d’utilisation
  - **De séquence** : représente les séquences entre objets du système par l’échange de messages plus détaillé
- **Global d'interaction** : représente les enchaînements entre scenarios (identifiés dans différents diagrammes de séquence)

Diagrammes de séquence (rappel)

- Diagramme qui représente les interactions entre classes par chronologie d'appels de méthode
- Un diagramme de séquence illustre un cas d'utilisation
- Le temps est représente par un axe vertical
- Une période d’activation correspond au temps pendant lequel un objet est en cours d’exécution

Diagrammes de séquence (rappel)

- Diagramm
  chronolo
- Un diagra
- Le temps
- Une péric
  objet est



classes par

sation

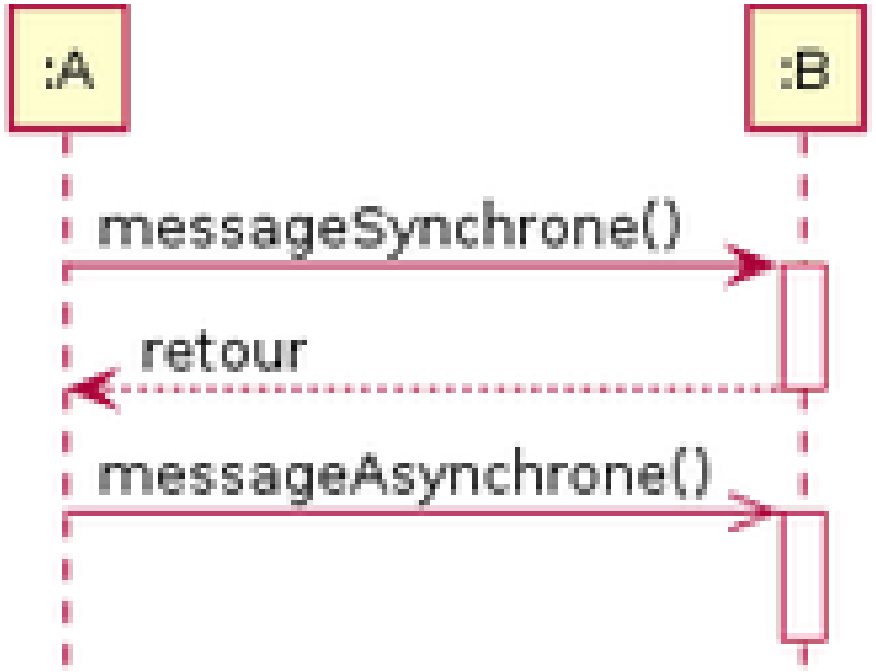
endant lequel un

Messages synchrones ou asynchrones

- Un message *synchrone* bloque l'expéditeur jusqu'à la réponse du destinataire. Le flot de contrôle passe de l'émetteur au récepteur.
  - Si un objet A envoie un message synchrone à un objet B, A reste bloqué tant que B n'a pas terminé.
  - On peut associer aux messages d'appel de méthode un message de retour (en pointillés) marquant la reprise du contrôle par l'objet émetteur du message synchrone.
- Un message *asynchrone* n'est pas bloquant pour l'expéditeur. Le message envoyé peut être pris en compte par le récepteur à tout moment ou ignoré.

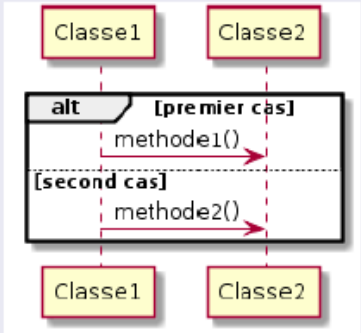
Messages synchrones ou asynchrones

- Un message *s* u'à la réponse du destinataire. L teur au récepteur.
  - Si un objet A et B, A reste bloqué tant que B n'a pa
  - On peut assc un message de retour (en pointillés) m t émetteur du message synchrone.
- Un message *a* r l'expéditeur. Le message envo récepteur à tout moment ou ig

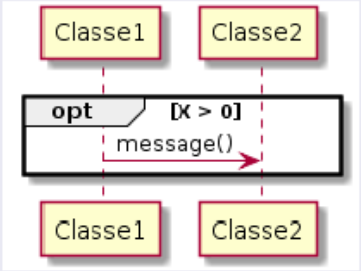


Fragments d’interaction (rappel)

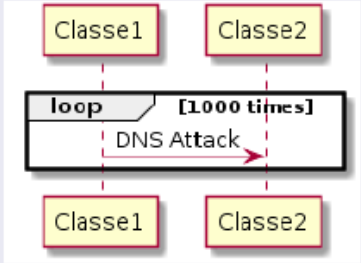
Alternatives



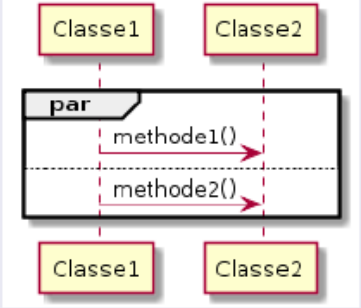
Options



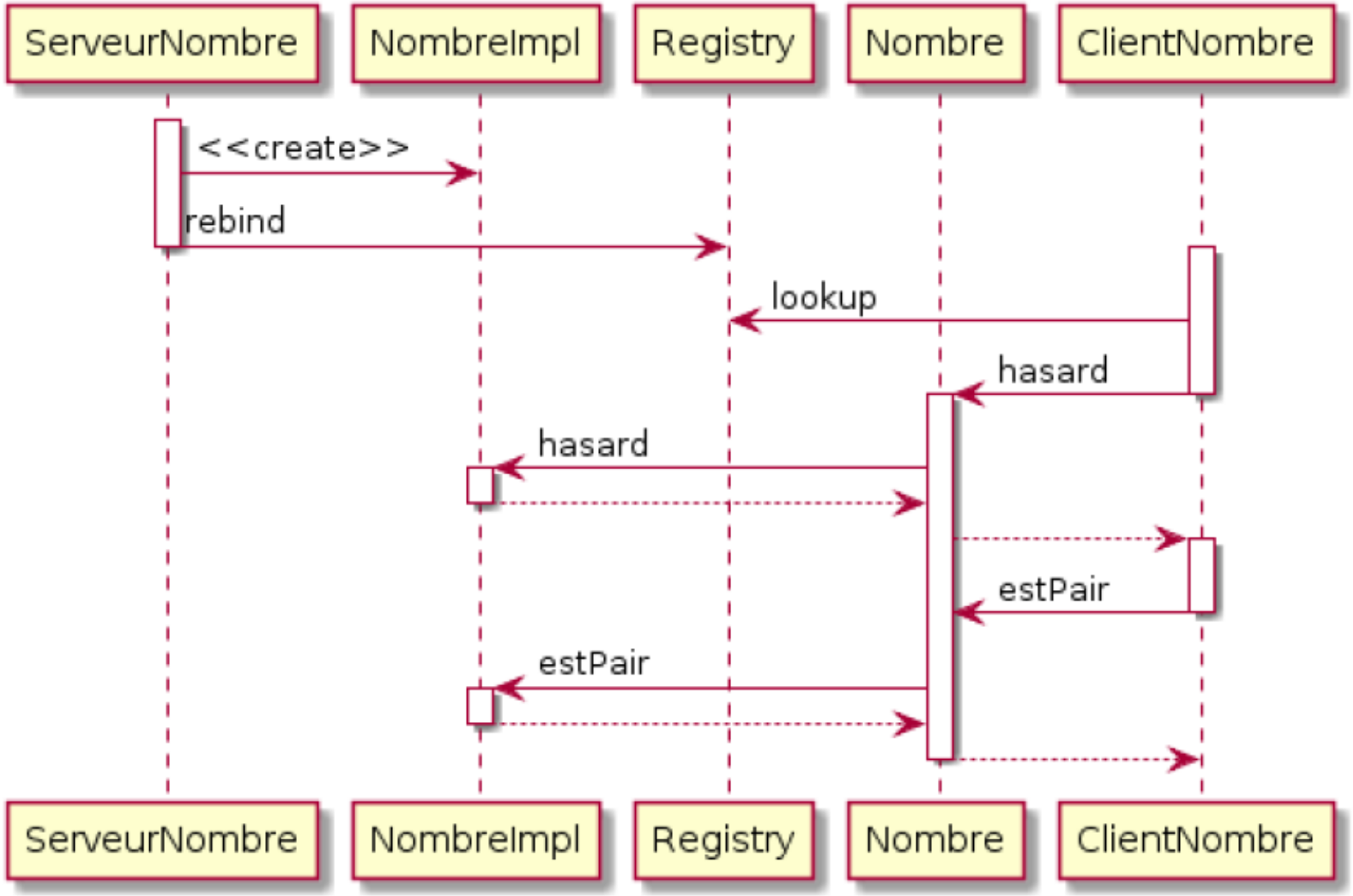
Boucles



Parallélisme



Diagrammes de séquence (rappel)



Diagrammes de composants (rappel)

- Composant : Elément physique représentant une partie de l'implémentation du système
  - code (source, binaire ou exécutable),
  - Script, fichier de commande, …
  - Fichier de données, table, . . .
- Un composant implante des services utilisables par d'autres composants et regroupe un ensemble de ressources de l'application pour en faire une entité réutilisable et/ou à déployer sur un support matériel.

Diagrammes de composants (rappel)

- Composant : Elément physique représentant une partie de l'implémentation du système
  - code (source, binaire ou exécutable),
  - Script, fichier de commande, …
  - Fichier de données, table, . . .
- Un composant implante des services utilisables par d'autres composants et regroupe un ensemble de ressources de l'application pour en faire une entité réutilisable et/ou à déployer sur un support matériel.

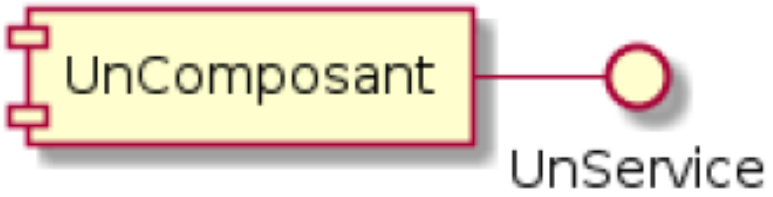


Diagramme de composants (rappel)

- Il est décrit par des interfaces fournies et requises
  - Une interface fournie est une fonctionnalité implémentée par le composant que d'autres composants peuvent appeler.
  - Une interface requise est une fonctionnalité que le composant doit pouvoir utiliser et implémentée par d'autres.

Diagramme de composants (rappel)

- Il est décrit par des interfaces fournies et requises
  - Une interface fournie est une fonctionnalité implémentée par le composant que d'autres composants peuvent appeler.
  - Une interface requise est une fonctionnalité que le composant doit pouvoir utiliser et implémentée par d'autres.

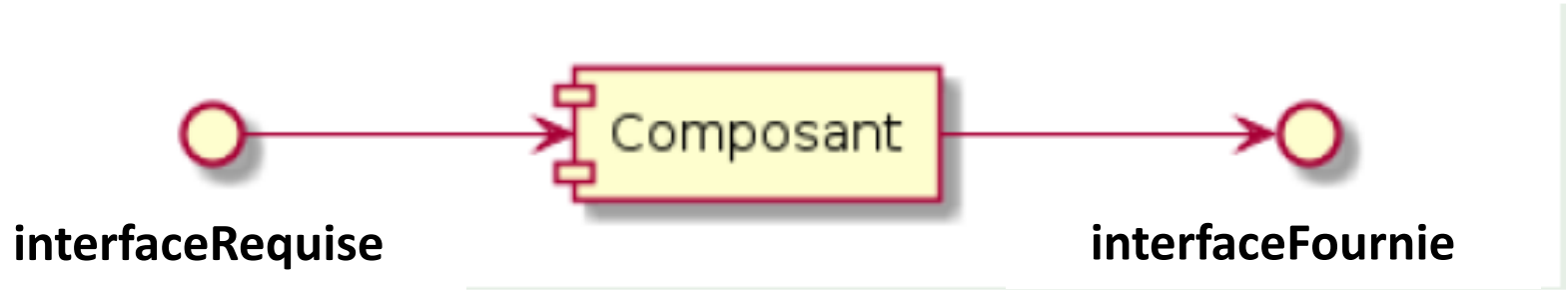


Diagramme de composants (rappel)

- Des relations de dépendance peuvent être exprimées entre composants. Elles représentent l'utilisation des services d'un composant par un autre

Diagramme de composant

- Des relations de dépendance peuvent ê
  composants. Elles représentent l'utilisat
  composant par un autre

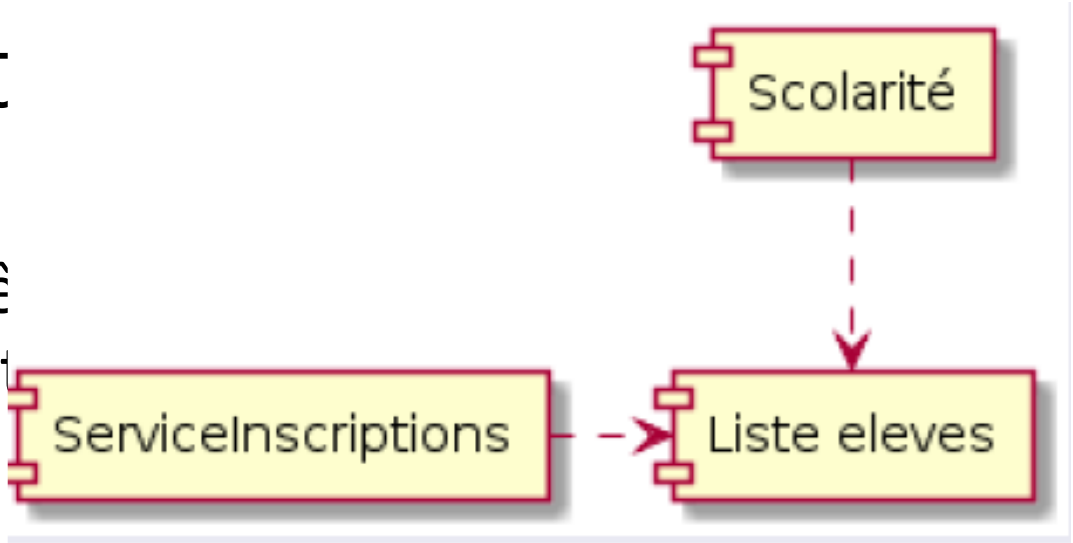
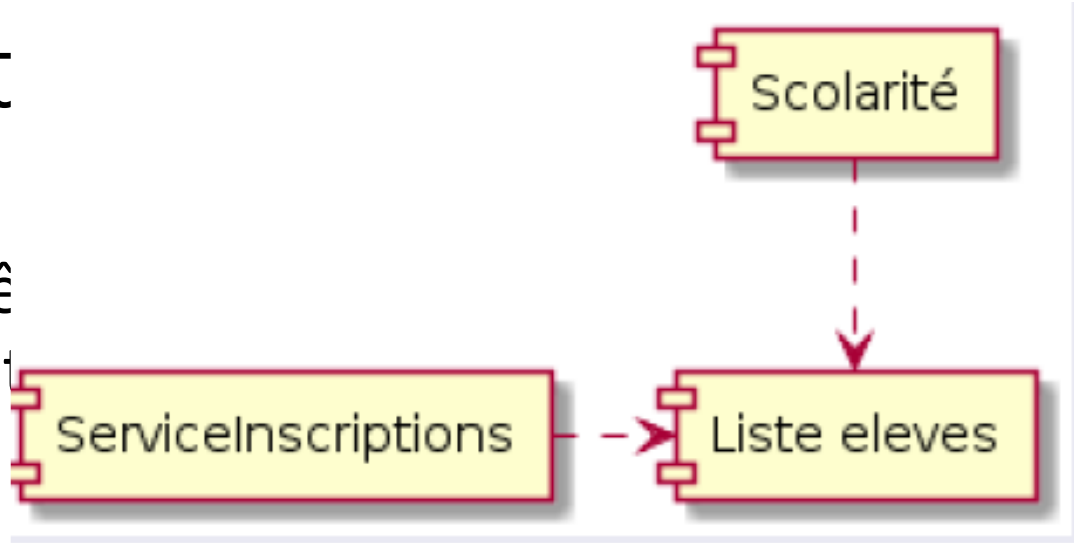


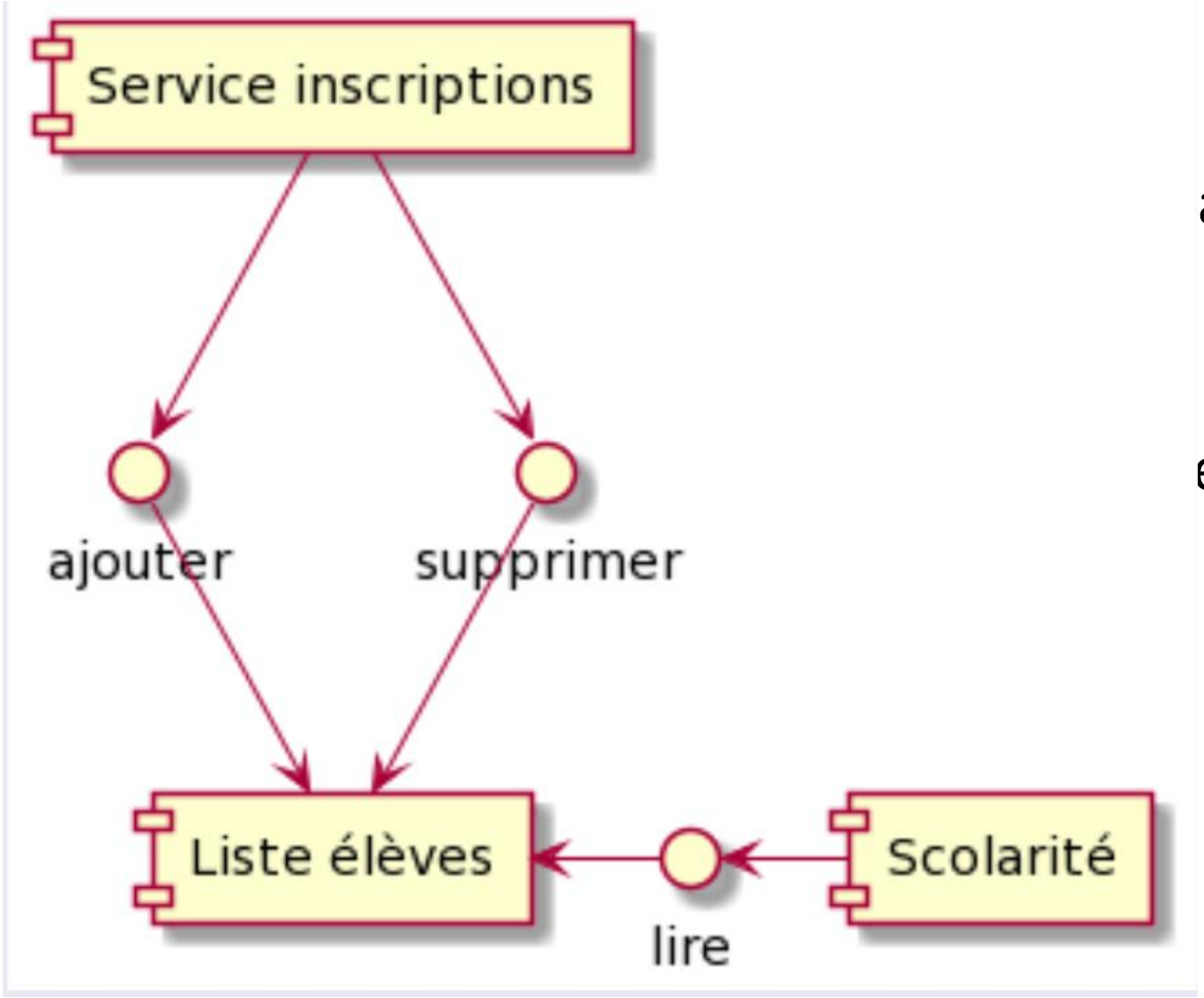
Diagramme de composant

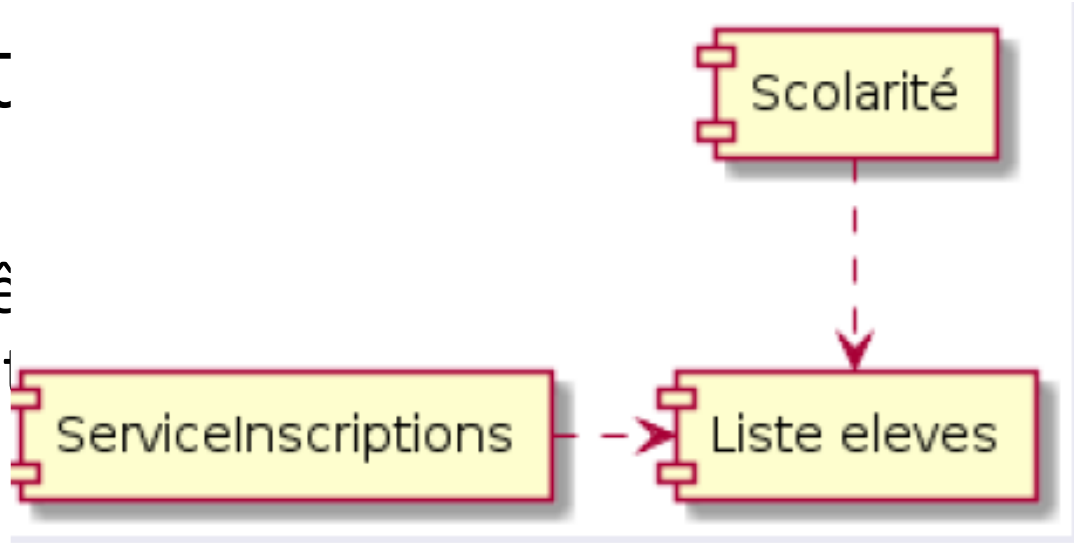


- Des relations de dépendance peuvent ê
  composants. Elles représentent l'utilisa
  composant par un autre

- Les dépendances entre composants peuvent s'exprimer plus finement au niveau de leurs interfaces

Diagramme de composant





euvent s’exprimer plus finement

Diagramme de déploiement

- Un diagramme de déploiement est la description de la configuration matérielle par l'emplacement des composants logiciels et matériels sur les nœuds de l'architecture.
- Il relie les interfaces requises et fournies des composants pour définir la manière dont ils communiquent.
- Les ressources matériels (ordinateur, périphériques, . . .) sont représentes par des nœuds.

Diagramme de déploiement

- Un diagramme de déploiement est la description de la configuration matérielle par l'emplacement des composants logiciels et matériels sur les nœuds de l'architecture.
- Il relie les interfaces requises et fournies des composants pour définir la manière dont ils communiquent.
- Les ressources matériels (ordinateur, périphériques, . . .) sont représentes par des nœuds.

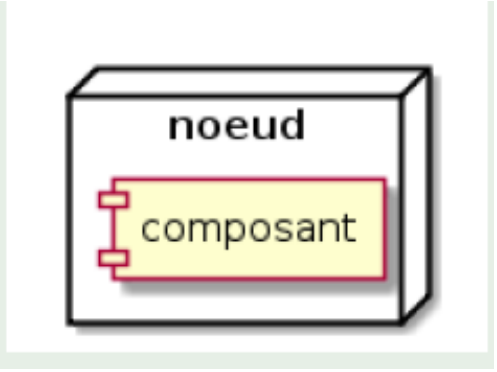
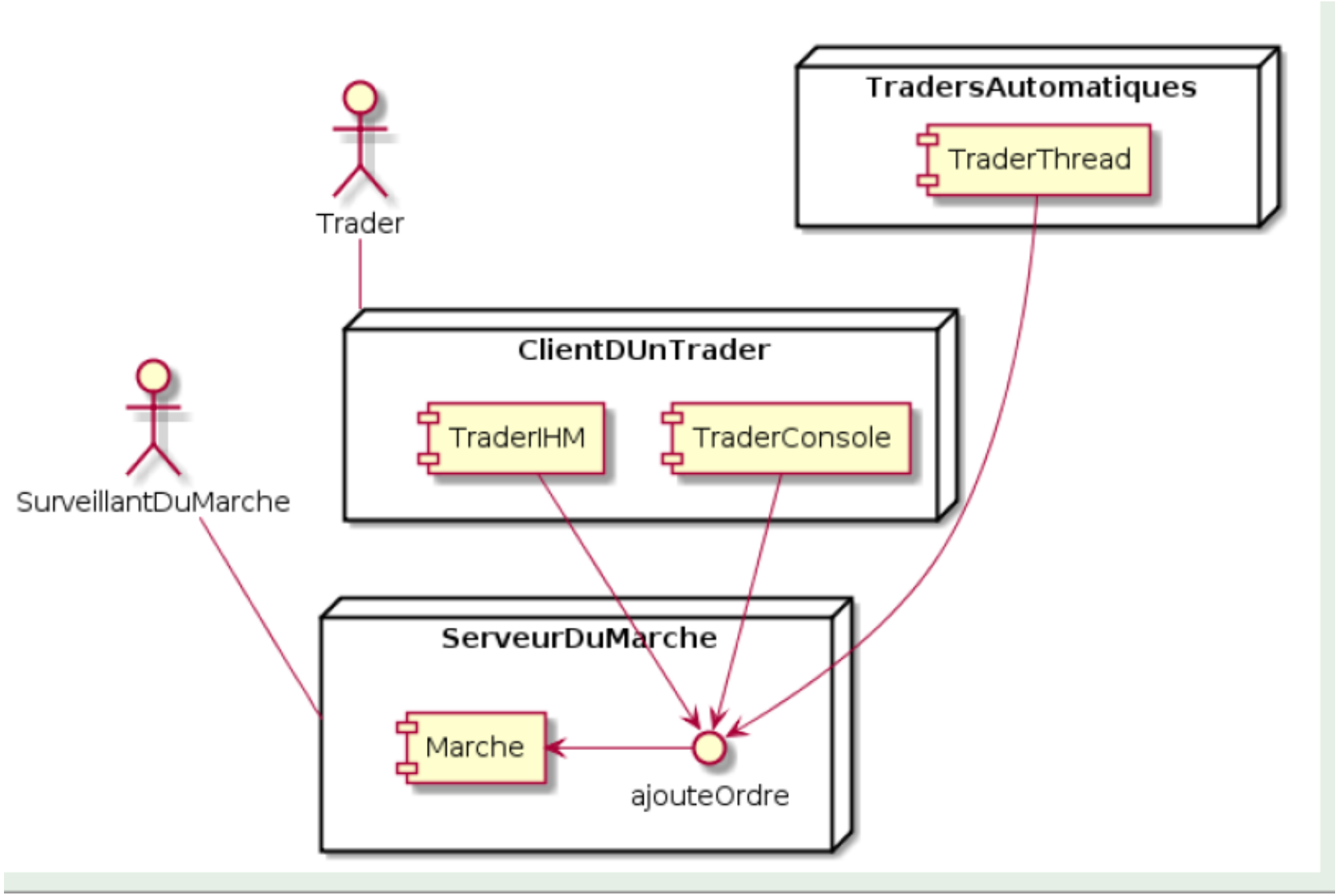


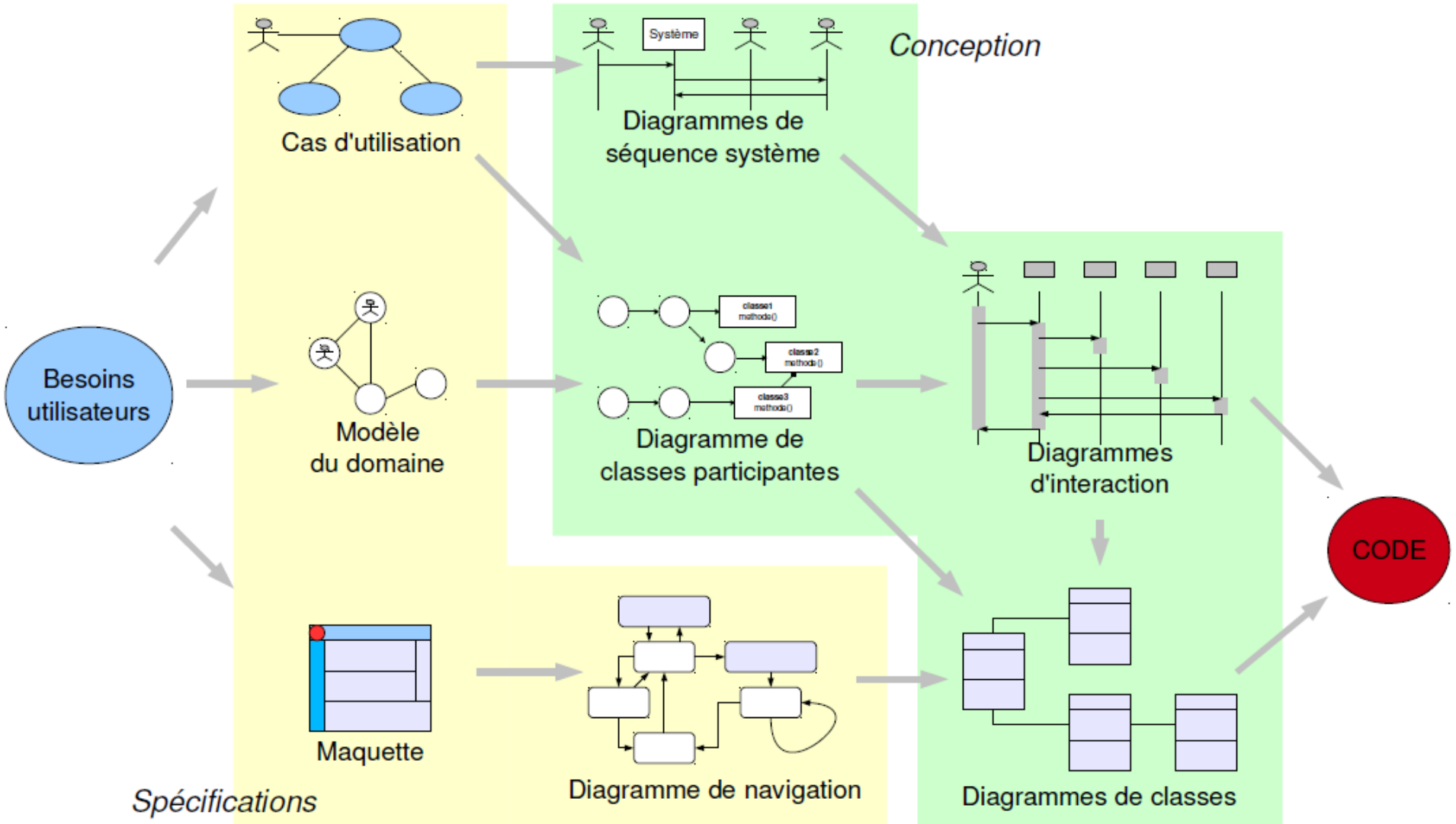
Diagramme de déploiement



Démarche de mise en oeuvre

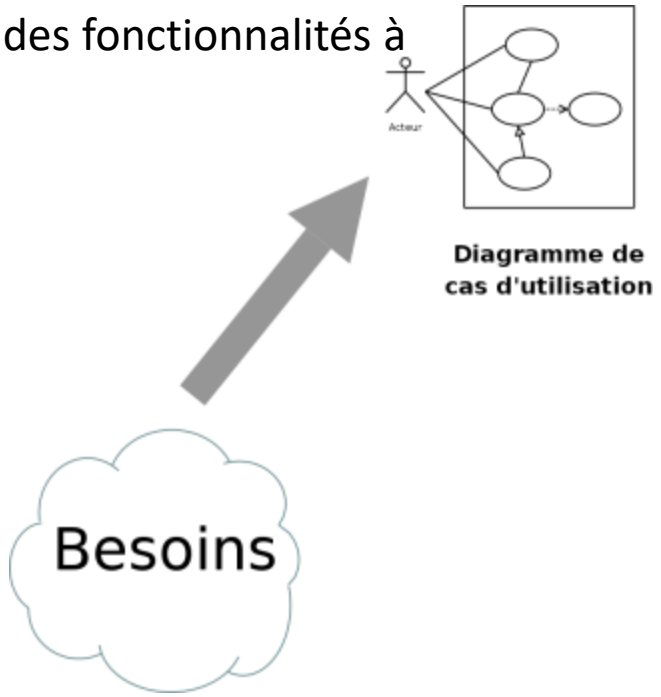
https://laurent-audibert.developpez.com/Cours-UML/?page=mise-en-oeuvre-uml

Démarche de GL simplifiée

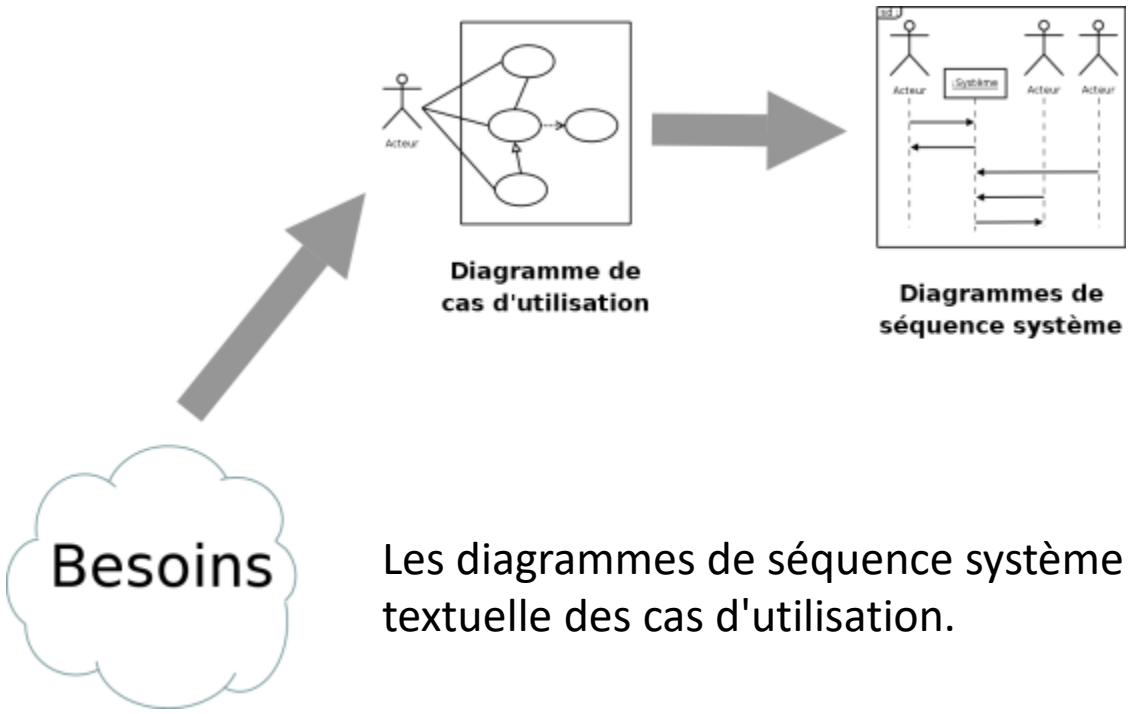


Identification des besoins

Les besoins sont modélisés par un diagramme de cas d'utilisation, à partir de la liste exhaustive des fonctionnalités à développer

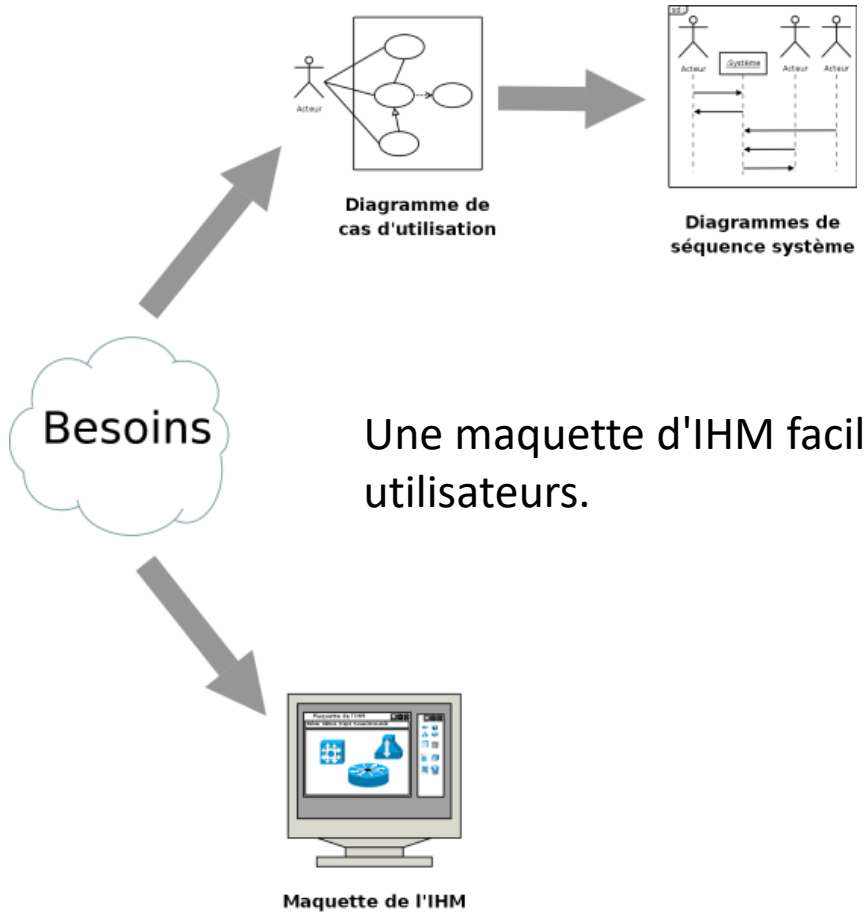


Identification des besoins



Les diagrammes de séquence système illustrent la description textuelle des cas d'utilisation.

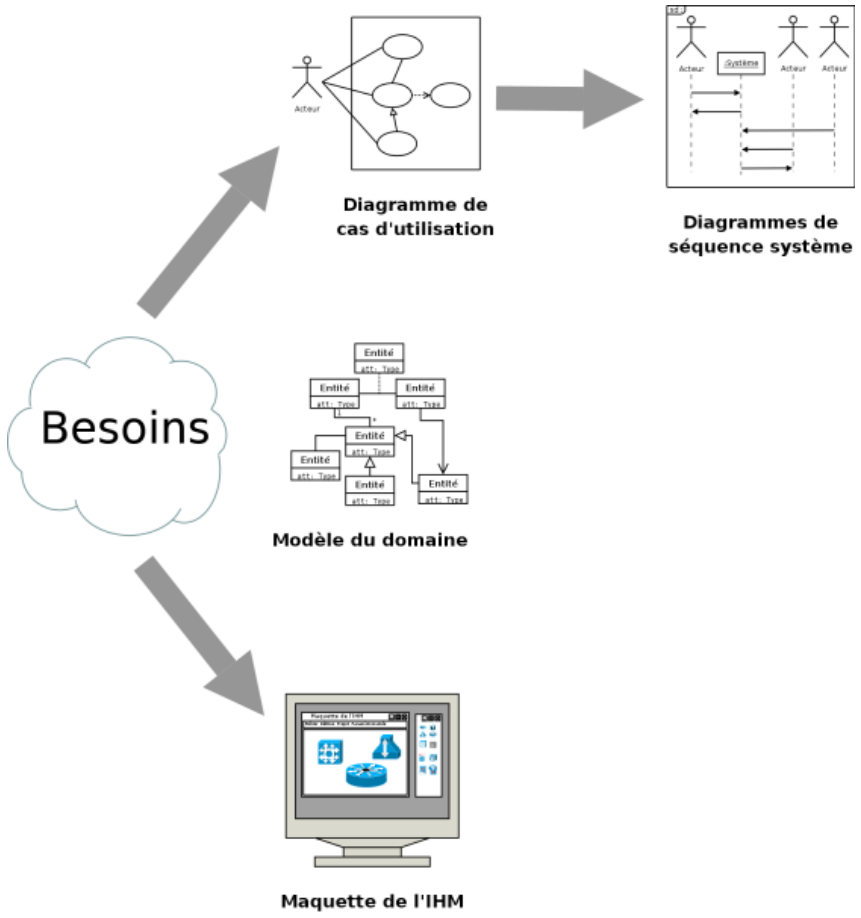
Identification des besoins



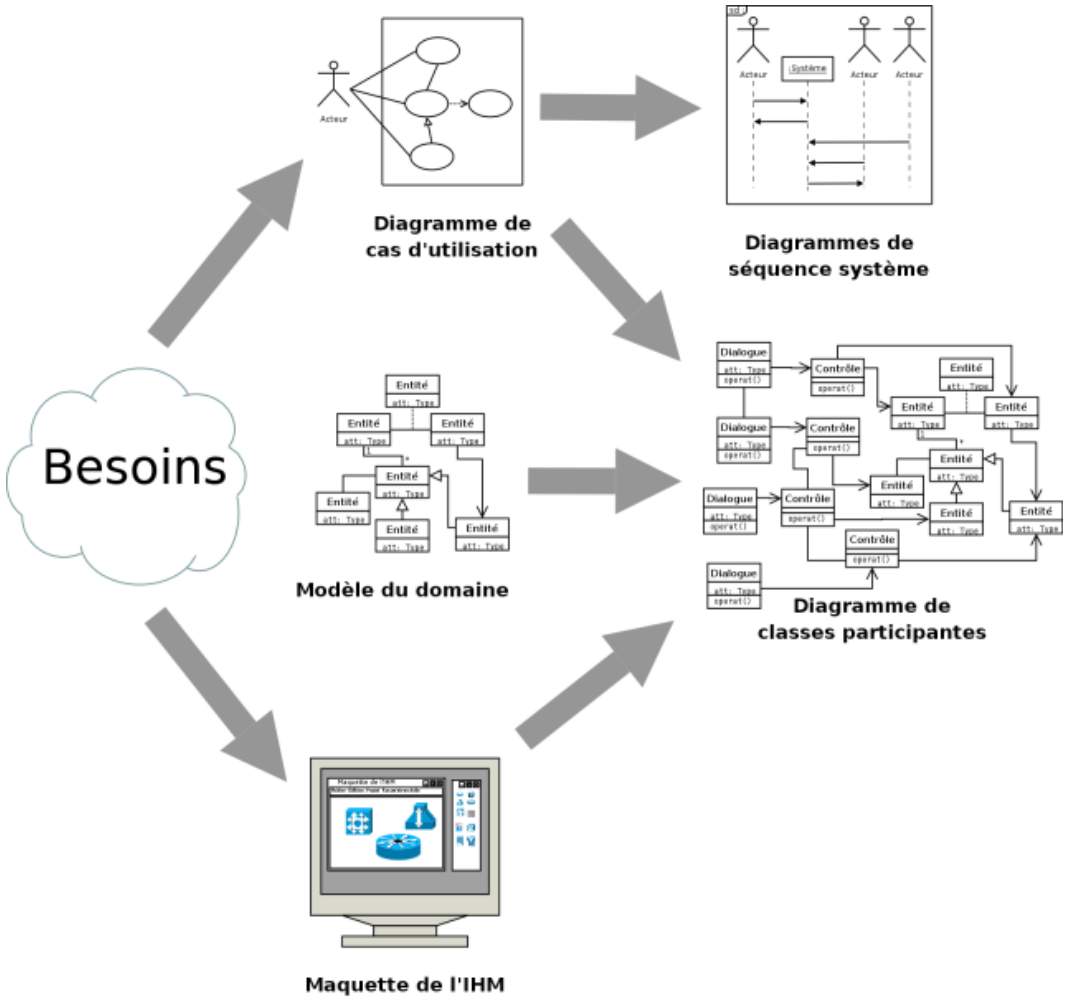
Une maquette d'IHM facilite les discussions avec les futurs utilisateurs.

Phase d’analyse

La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes.

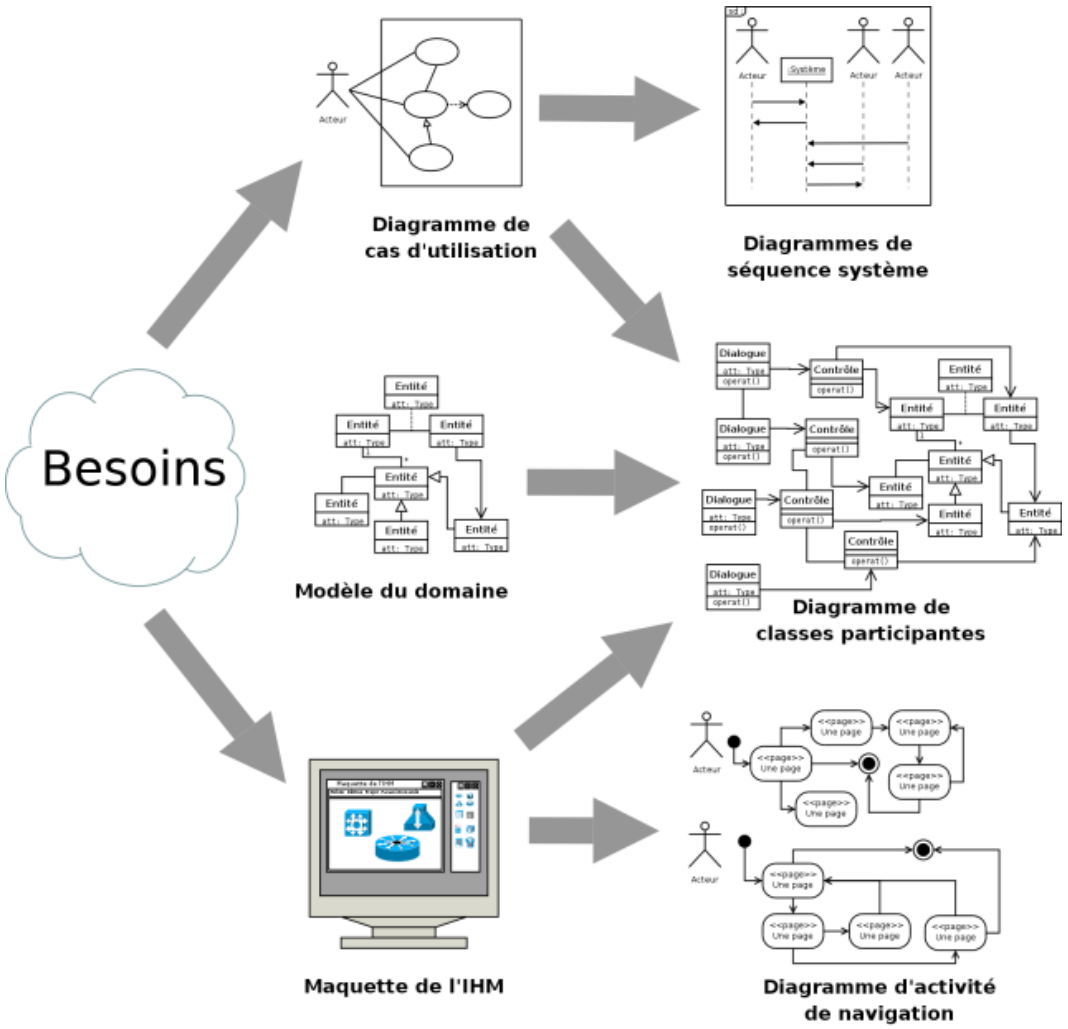


Phase d’analyse



Le diagramme de classes participantes effectue la jonction entre les cas d'utilisation, le modèle du domaine et les diagrammes de conception logicielle.
On inclura ici les éléments issus du modèle MVC … les dialogues, les contrôles et les entités (qui proviennent directement du modèle du domaine), ainsi que leurs relations

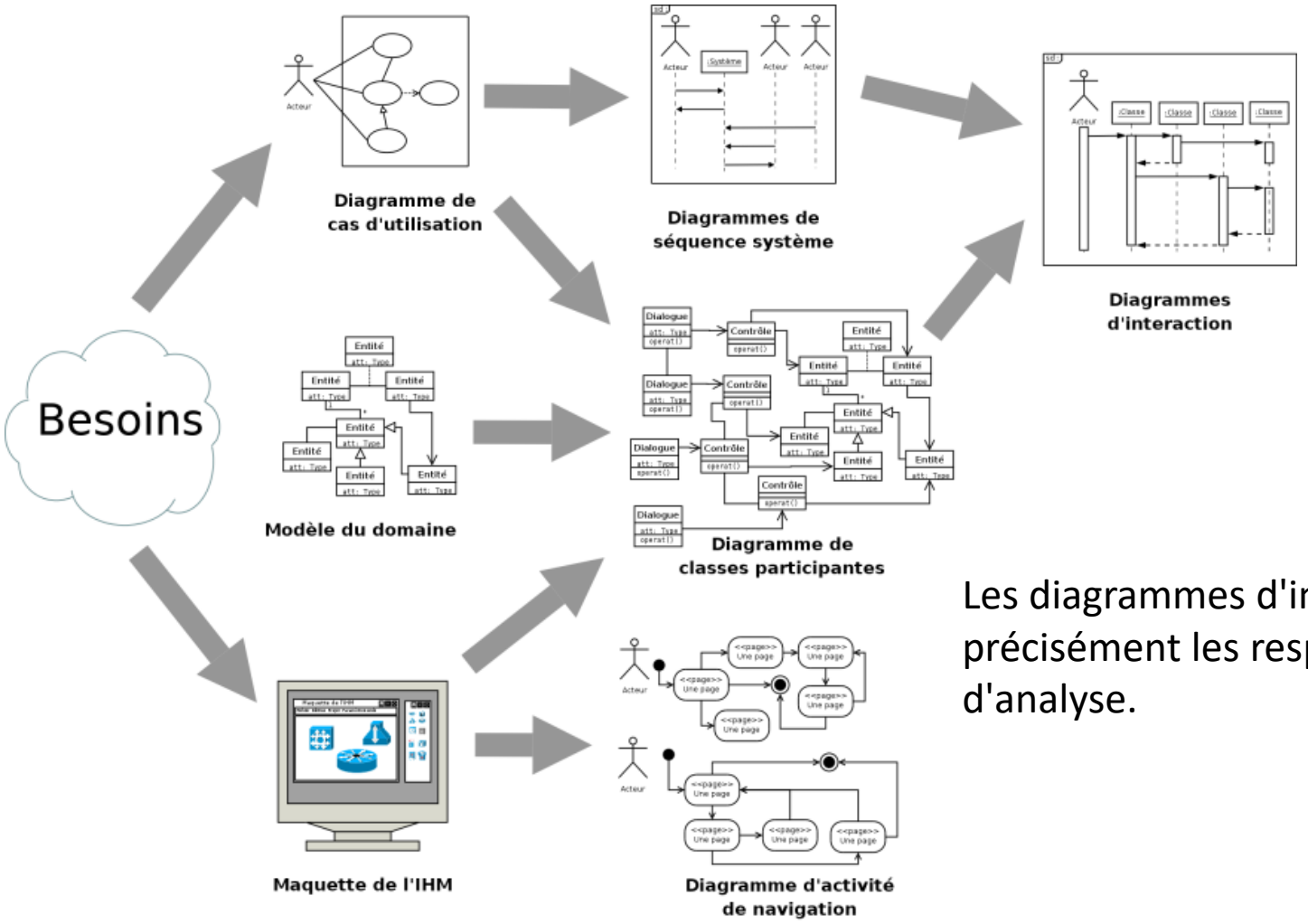
Phase d’analyse



Les diagrammes d'activités de navigation représentent graphiquement l'activité de navigation dans l'IHM.

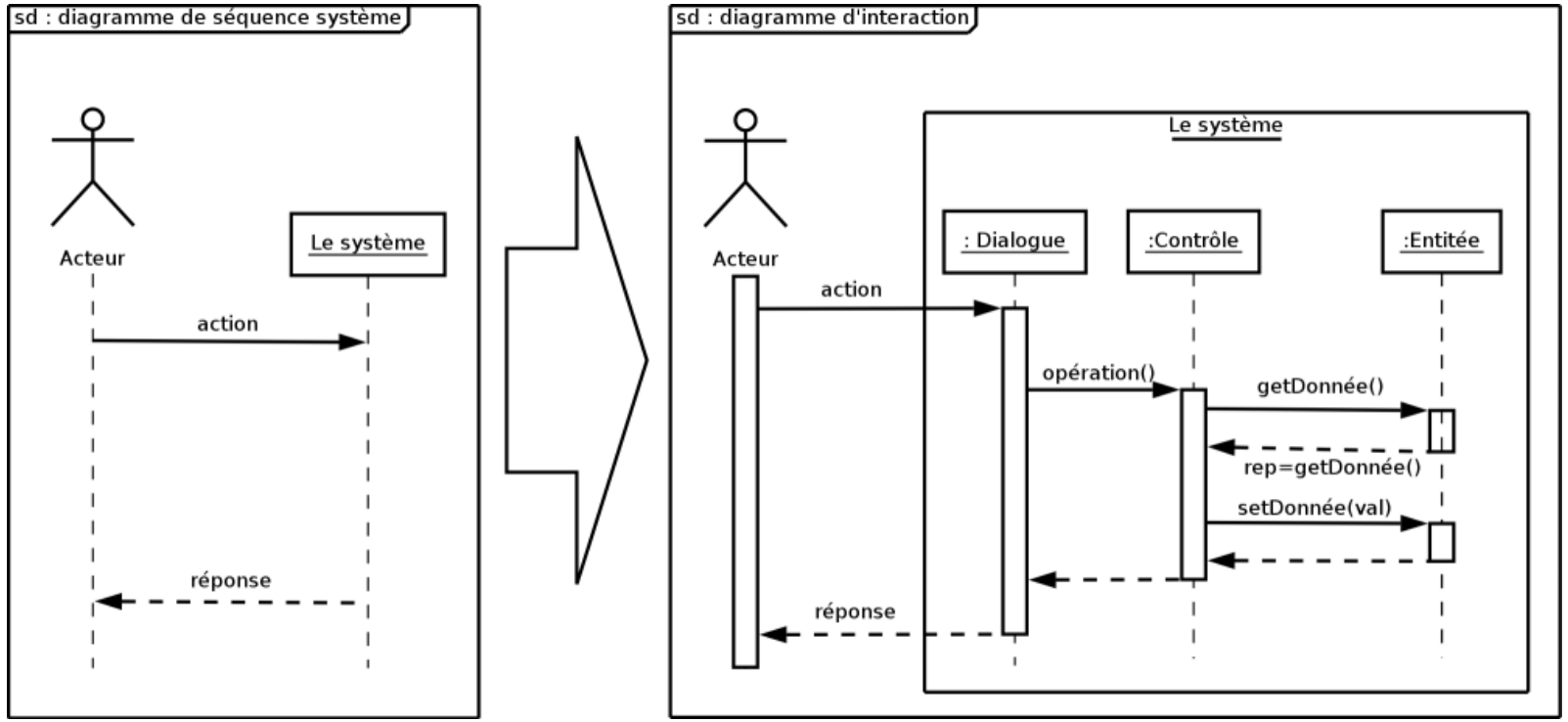
La modélisation de la navigation devra être structurée par acteur

Phase de conception



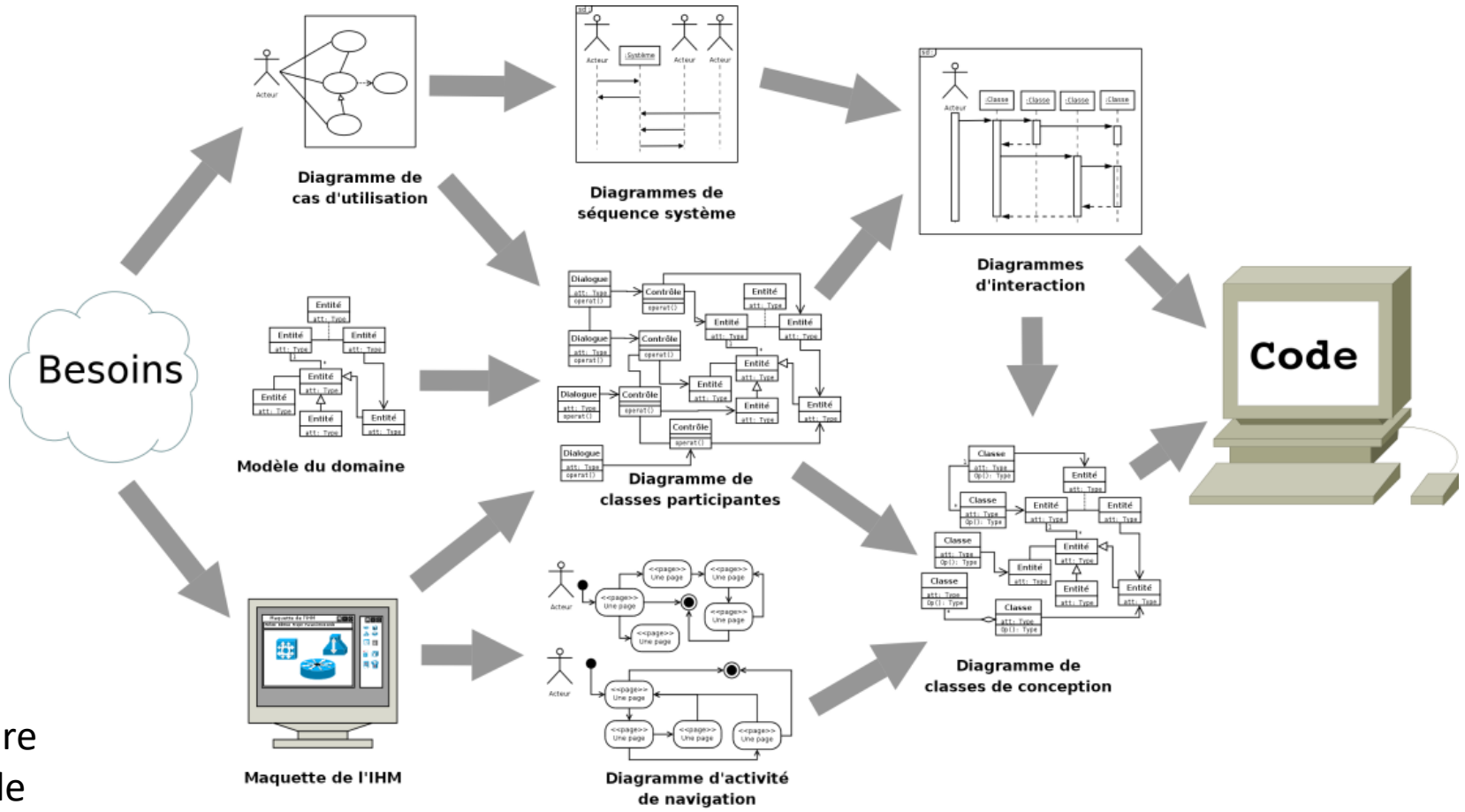
Les diagrammes d'interaction permettent d'attribuer précisément les responsabilités de comportement aux classes d'analyse.

Phase de conception



Le système des diagrammes de séquences système, vu comme une boîte noire, est remplacé par un ensemble d'objets en collaboration.

Phase de conception



Le diagramme de classes de conception servira pour l'implémentation. Il faudra prendre en compte les choix techniques (le langage de programmation, les bibliothèques utilisées …)

Livrables pour projet

Document de spécifications

- Introduction
  - Il s'agit d'une description informelle du projet et de son contexte.
  - On doit y trouver notamment comme informations :
    - la liste des fonctions principales,
    - Les différents utilisateurs et leurs caractéristiques,
    - les contraintes matérielles et logicielles.
- Besoins détaillés
  - C'est la partie contractuelle à proprement parler puisqu'elle formalise le besoin.
  - Elle consiste 3 parties distinctes :
    - les spécifications fonctionnelles ***(la liste exhaustive des fonctionnalités à développer, en français ☺)***
    - les spécifications d'interfaces ***(avec d'autres logiciels, utiles dans un cadre d'InfoRep)***
    - les spécifications opérationnelles (performance, sécurité, ...)
  - Ces différents éléments peuvent s'appuyer en UML sur des diagrammes de cas d'utilisation, des diagrammes de séquence système, d'un diagramme de modèle du domaine, de maquettes et d'un diagramme de navigation, en fonction des besoins.

Document de conception

- Trois sections: Introduction, Conception préliminaire, Conception détaillée
- Introduction
  - Il s'agit d'une description informelle du projet et de son contexte. Elle est souvent relativement similaire à l'introduction d'un document de spécification.
  - On doit y donc y trouver notamment comme informations :
    - la liste des fonctions principales,
    - les différents utilisateurs et leurs caractéristiques,
    - les contraintes matérielles et logicielles.

Document de conception

- Conception préliminaire
  - Cette étape consiste à réaliser une conception macroscopique, c'est-à-dire permettant de mener à un découpage en packages avec les signatures externes de chaque package. Cette étape peut s'appuyer en UML sur :
    - Un diagramme de modèle du domaine (si non spécifié) ;
    - Des diagrammes de séquence système (si non spécifiés) ;
    - Des maquettes (si non spécifiées) ;
    - Des diagrammes d'activités de navigation (si non spécifiés) ;
    - Des diagrammes d'interaction ;
    - Un diagramme des classes de conception préliminaire ;
    - Un découpage de composants avec les interfaces requises et fournies
    - Un diagramme de déploiement

Document de conception

- Conception détaillée
  - Cette étape consiste à détailler par package les éléments constituant les composants.
  - L'utilisation de diagrammes de paquetages est possible
  - Concrètement, il s'agit principalement de préciser les attributs et méthodes de classe de toutes les classes participantes et de les regrouper dans un diagramme de classes.
  - Les méthodes non triviales devront être commentées et voir leur fonctionnement détaillé par du pseudo-code.
  - **Chacun des éléments doit indiquer clairement l'auteur qui prendra en charge son développement**

Rapport final

- Doit contenir
  - Spécifications (Document de spécifications mis à jour si nécessaire)
  - Conception (Document de conception mis à jour si nécessaire),
  - Choix techniques justifiés
  - Améliorations possibles (non prévues dans les spécs)

Présentation du projet

Consignes

- Les projets sont a développer par groupes de 4 étudiants (voir Google Sheet partagé)
- Il s'agira de suivre une démarche de Génie Logiciel tout le long du projet avec des livrables intermédiaires
  - le 21/02 à 13h00 au plus tard sur Moodle : spécification et conception du projet ***(je ferai un retour sur le Google Sheet)***
  - Le 28/03 : soutenance (10 min !!!) sous la forme d'une présentation incluant une démonstration (éventuellement sous forme de vidéo).
    - Rapport, code source, manuel de l'utilisateur, et vidéo doivent être déposés sur Moodle avant le 28/03 à 8h

Consignes

Conception de Systèmes Repartis / Projets

Groupes pour Projets

Merci de renseigner ce Google Sheets
Vous vous organiserez en groupes de 4 étudiants

Projets

- Consignes développement projet
- Sujets

Livrable Spécification et Conception du Projet Final

Cahier de spécification et de conception

Date limite : le lundi 21 février 2022 à 13h00

Rendu final projet

Date limite le lundi 28 mars à 8h00

- Rapport final,
- code source,
- manuel utilisateur et
- vidéo

Conception de Systèmes Repartis / Projets

Groupes pour Projets

Merci de renseigner ce Google Sheets
Vous vous organiserez en groupes de 4 étudiants

Projets

- Consignes développement projet
- Sujets

Livrable Spécification et Conception du Projet Final

Cahier de spécification et de conception

Date limite : le lundi 21 février 2022 à 13h00

Rendu final projet

Date limite le lundi 28 mars à 8h00

ITI4 - Groupes pour Projets

#	Etudiant 1		Etudiant 2		Etudiant 3		Etudiant 4		Technologie	Commentaires	Heure de passage le 28/03
1											13h15 - 13h30
2											13h35 - 13h50
3											13h55 - 14h10
4											14h15 - 14h30
5											14h35 - 14h50
6											14h55 - 15h10
7											15h15 - 15h30
8											15h35 - 15h50
9											15h50 - 16h10
10											16h15 - 16h30

Consignes

- ***Vous n'aurez pas beaucoup de temps pour développer le projet. C'est pour cela que les phases de spécification et conception sont TRES importantes pour bien choisir ce que vous allez faire***
- L'évaluation des projets se fera sur la base du rapport final, de la présentation, d'une démo en forme de vidéo et du code
  - Les soutenances dureront 15 minutes, dont 5 de questions. Donc il faut très bien organiser ce que vous allez présenter et bien gérer la durée de la vidéo
  - Chaque méthode/package/objet devra contenir en commentaire le nom de la ou des personnes ayant participé à sa conception et à son développement, de façon à faire apparaître clairement la répartition des tâches.
  - Un manuel de l'utilisateur est demandé

THE projet

Gestion du système de santé

- Vous concevrez un ensemble d'applications distribuées proposées au sein du système de santé français.
  - Le dossier d'un patient contient une partie administrative (consultable par les employés de la CPAM) et une partie médicale (consultable par les médecins).
  - Quand un patient consulte un médecin, celui ci met à jour le dossier médical et transmet les données à la CPAM qui gère les remboursements et les informe au patient et à sa mutuelle.

Gestion du système de santé

- Il s'agira, donc, de concevoir et implémenter un serveur CPAM qui proposera des services aux médecins, aux administratifs et aux patients.
- Il faudra, en conséquence, concevoir et implémenter différents types de logiciels « client » en fonction du profil de l'utilisateur, qui viendront consommer les services proposés par le serveur.
- Votre architecture devra prévoir toutes ces interactions et être assez flexible pour pouvoir être généralisée par la suite (par exemple, gestion des interactions avec les banques pour effectuer les remboursements)

Organisation et deadlines

Organisation

- Vous travaillerez en **groupes de 4 étudiants** (***avec un groupe de 3 étudiants, si je fais confiance aux inscriptions sur Moodle***)
- Les groupes sont à renseigner sur le Google Sheet dont l'URL est publié sur Moodle
- Tous les groupes vont développer le même projet
  - Tirage au sort de la technologie à utiliser : RMI, SOAP, REST
  - *Le 31 janvier 2022 à 13h* … une fois que les groupes seront constitués … AVEC ou SANS témoins :-D

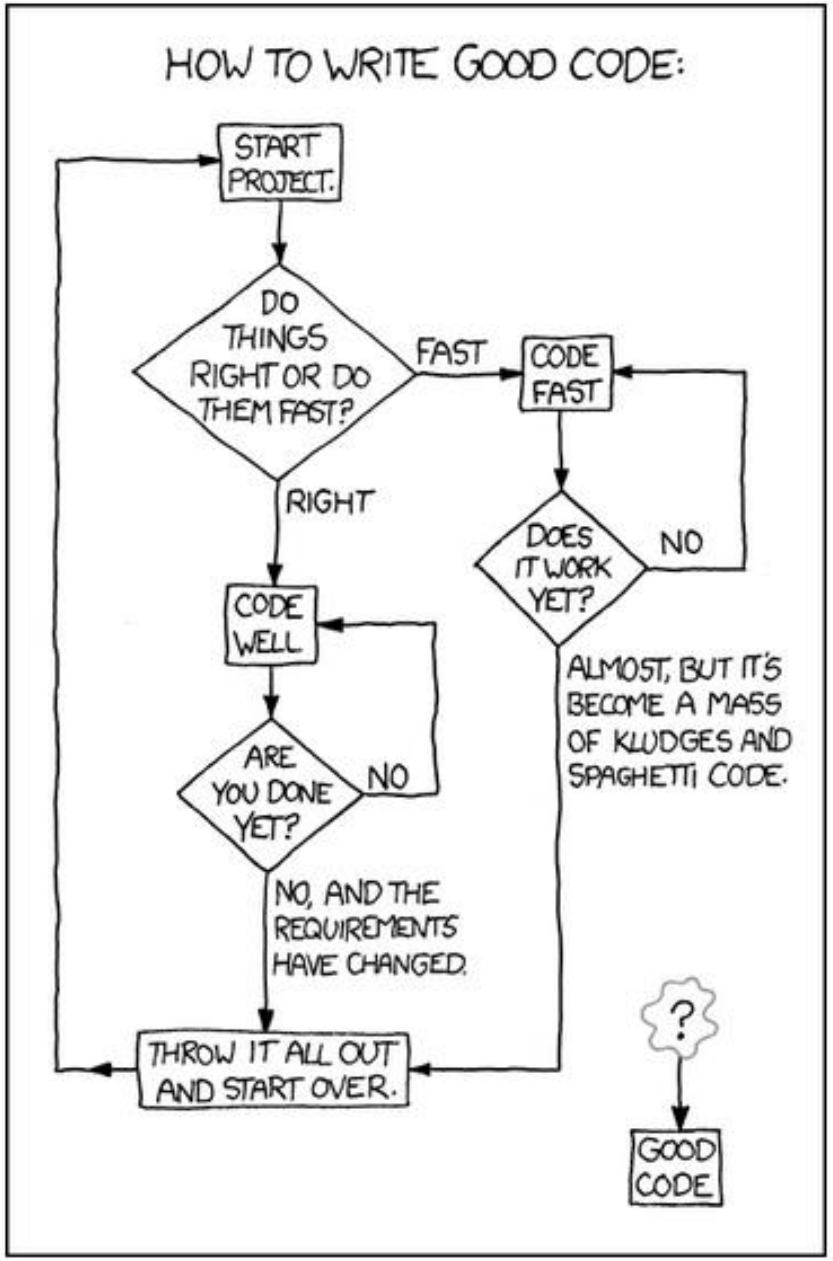
Livrables et deadlines

- Document de spécification et document de conception : **le lundi 21 février avant 13h**
  - Je ferai un retour sur ce document au fur et à mesure des dépôts sur Moodle
  - Merci de m'envoyer un message sur Discord pour me prévenir dès que le dépôt est fait
- Rapport final et code source : **le lundi 28 mars avant 8h00**
- Soutenances : **le 28 mars à partir de 13h15**

- **Attention :** QCM individuel **le 21 mars de 14h à 15h30**

Un mot sur la méthodologie …

Un mot sur la méthodologie .



- https://xkcd.com/