IUT Nantes
Pôle Sciences et technologie
Nantes Université

# R2.10 Gestion de projet & des organisations
# Partie Gestion de Projets Informatiques
# BUT Info – Semestre 2

Jean-Marie Mottu
IUT de Nantes – Département Informatique

# Organisation de la ressource

- Volume
  - 7-8 cours magistraux
  - 8-9 TDs de 2h40
- Enseignants
  - Jean-Marie Mottu : CM et TD
  - Xavier Aimé, Florent Cordeau, Saïd El Mamouni : TD
- Notation
  - Contrôle(s) continu(s)
    - QCMs en séance de CM
    - Note(s) pratique(s)
    - Contrôle fin de période

# Programme

- Enjeux
- Cas d'utilisations - UML
- Tâches
- Démarche Projet
- Organisation
- Acteurs
- Répartition
- Typologie
- Planification
- Cahier des Charges
- Cycle de développement
- Suivi de Projet
- Risques
- Gestion de version
- Outils de travail collaboratif

IUT Nantes
Pôle Sciences et technologie
Nantes Université

# GPO2
# Enjeux

Jean-Marie Mottu
IUT de Nantes – Département Informatique

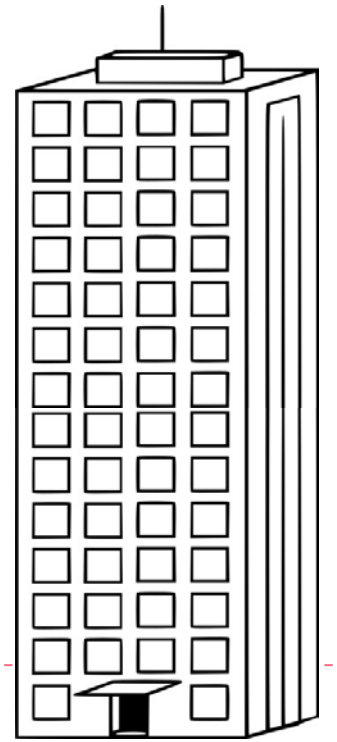
# Enjeux de la gestion de projet

- Gestion de projet & des organisations 2 est un module à part entière du programme de BUT
  - Compétences promues :
    - C4 Gérer des données de l'information
    - C5 Conduire un projet
  - Cette partie Gestion de Projets Informatiques est directement liée dans ce semestre aux :
    - R2.01 Développement orienté objets
    - R2.03 Qualité de développement
    - SAEs de Semestre 2

- Un projet c'est _____

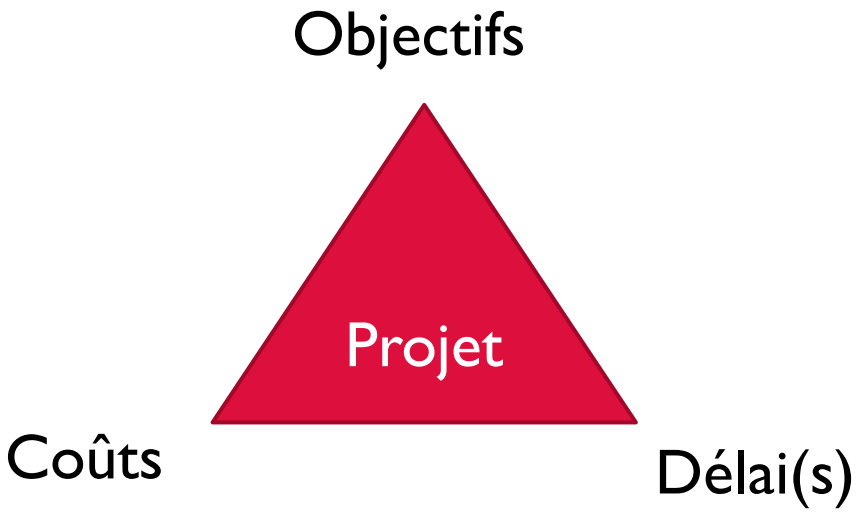
# Enjeux de la gestion de projet

- On met en place un projet quand la résolution d'un problème et/ou l'amélioration d'une activité
  - a des **objectifs** dont l'importance
  - nécessitent l'emploi de **moyens**
  - pendant un **temps** conséquent.
- Un projet est un challenge à relever
  - Plus un projet sera important
  - Plus il nécessitera de moyens coordonnés
  - Plus il sera difficile de tenir les délais



# Triangle de fer de la gestion de projet

- Interdépendance nécessitant des compromis entre
  - Coûts
    - Moyens/Ressources : humain, matériel
  - Temps
  - Objectifs
    - Fonctionnels
    - Extra-fonctionnels



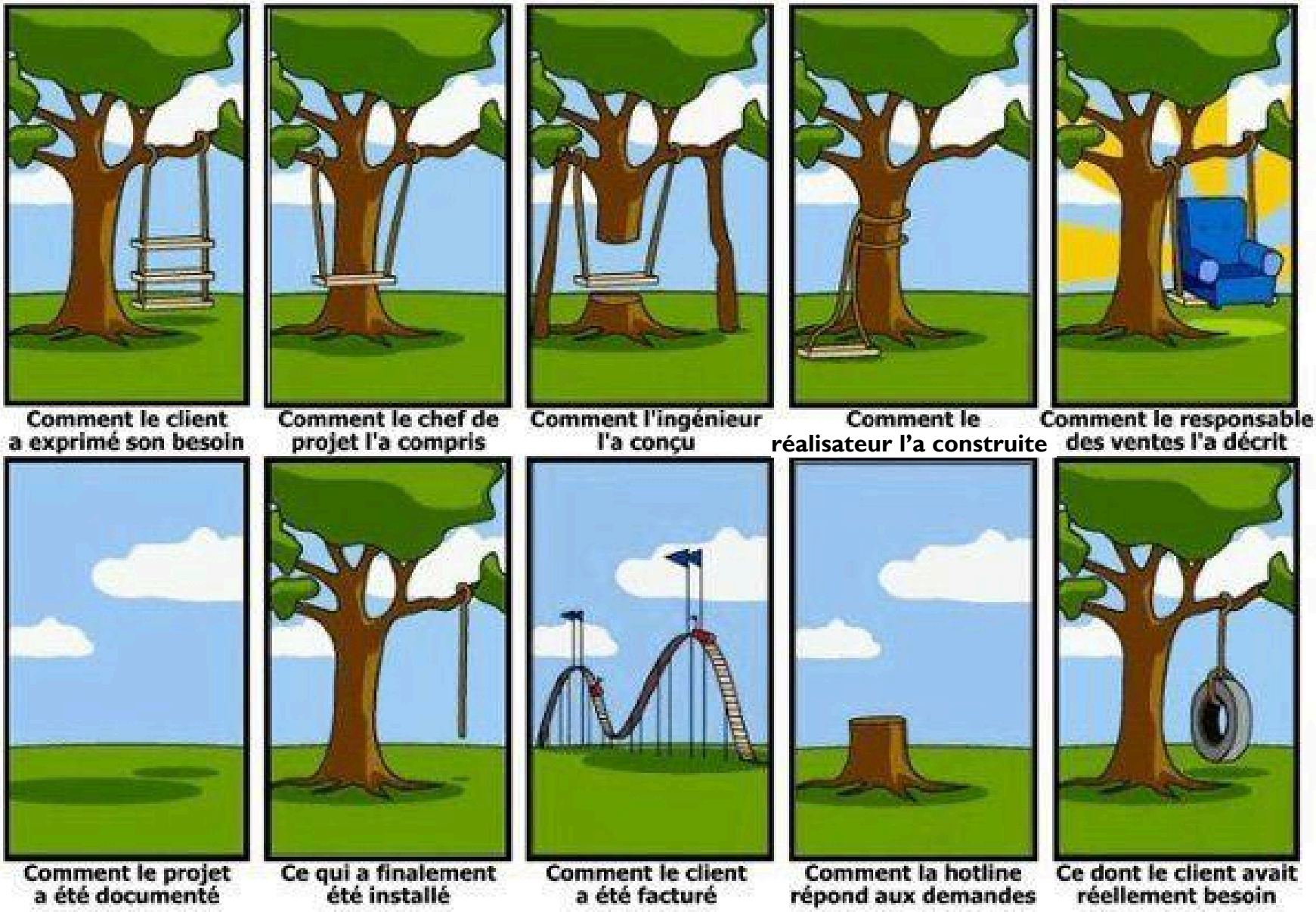
# Mise en œuvre de projets

- Des ressources sont mobilisées (Acteurs, Outils, Matériaux, etc.)
- pour effectuer des tâches
- afin d'obtenir une réalisation
  - Rendre un service
  - Produire un bien
    - Matériel, immatériel

- Dans un projet de construction d'une maison
  - L'architecte dessine un plan
  - Le bureau d'étude simule la consommation énergétique
  - Le maçon avec sa bétonnière construit un mur
  - Le diagnostiqueur avec une porte soufflante vérifie la perméabilité à l'air.

# Les mêmes difficultés partout

- Génies civil, chimique, électrique, logiciel …



# Gestion de projets informatiques

- S'intègre dans le Génie Logiciel
  - « Règles de l'art » de l'ingénierie de la réalisation de systèmes manipulant de l'information ou système à logiciel prépondérant
  - Ensemble des méthodes permettant de réaliser un projet de développement logiciel
    - Théorique
    - Empirique
  - Organisation des métiers du logiciel

# Les acteurs d'un projet informatique

- Client
- Chef de projet
- Ingénieur
- Développeur
- Responsable des ventes
- Installateur
- Client
- Support
- etc.

Principales ressources d'un projet informatique

# Les livrables, dépendances

- Le Produit
  - Logiciel
    - Professionnel ou pas
    - Desktop
    - Embarqué
    - On-the-shelf
  - Sa spécification, sa documentation, ses tests, etc.
- L'environnement de déploiement
  - Physique ou pas
- L'environnement de développement
  - Station de travail
  - Simulateur
  - Maquette

# Réussite et échecs

- Réussite si le triangle de fer a maintenu son équilibre
- Échecs
  - Abandon ou Difficultés (hors délai, hors budget, objectifs non atteints)
- Conséquences
  - Financière
    - Gaspillage
    - Faibles ventes
    - Pénalités
    - Condamnations de justice
  - Humaine
    - Emploi
    - Condamnations de justice
    - Accidents
  - Autre
    - Réputation
    - Etc.

# Exemple d'échec avec Incidence financière

- Louvois : logiciel de paye des soldats français
  - Retards et plus de *573* millions € de primes trop perçues
    - 50% des personnels de l'armée de terre concernés
    - ~100 millions ne seront pas récupérés
  - Un logiciel au coût indirect estimé 470M€
  - Début projet 1996, déploiement (partiel) 2011, abandon 2013, déploiement sur plusieurs années de son remplaçant (2019)
  - Multiples raisons
    - Un objectif louable mais une complexité sous-estimée
      - Fusion de plusieurs systèmes, multiples scénarios
      - Par exemple, une estimation de 15 000 dossiers à traiter devient concrètement 140 000.

# Exemple d'échec avec Incidence vitale

- Printemps 2019 : 2 crashs de Boeing 737 MAX
  - 346 victimes
  - ~400 appareils interdit de voler, des milliers de vols annulés
  - Arrêt de la production, ~400 appareils pas livrés/payés
- Un système de sécurité se déclenchait à tort et les pilotes ne savaient pas l'éteindre
- Multiple raisons
  - Un problème mécanique sensé être pallié par un logiciel
    - mais un manque de redondance de capteurs défaillants
  - 2 voyants d'alerte mais en option
  - Auto-certification
    - Alors que certaines personnes à Boeing se doutaient d'un problème
  - Pas de formation des pilotes à cette nouvelle version MAX
    - De toute façon, le logiciel des simulateurs n'avait pas été adapté
  - Manque de coopération: la veille un pilote avait évité un tel crash
- Boeing n'a mis que 2 mois à mettre à jour le logiciel et les simulateurs mais le premier vol n'a eu lieu que le 29/12/2020

# Facteurs de succès des projets

| | Facteurs de succès des Projets "The Chaos Report" – Standish Group 1994 | Facteurs de succès des Projets "Chaos Report" – Standish Group 2014 |
|---|---|---|
| 1er | Implication des utilisateurs | Implication des utilisateurs |
| 2e | Soutien de la direction | Soutien de la direction |
| 3e | Demande claire et actée | Objectifs métier clairs |
| 4e | Planning bien géré | Planification appropriée |
| 5e | Attentes réalistes | Attentes réalistes |
| 6e | Jalons rapprochés | Itérations courtes |
| 7e | Equipe compétente | Equipe compétente |
| 8e | Maîtrise d'ouvrage bien identifiée | Participation du client |
| 9e | Vision et objectifs clairs | Projection et objectif clairs |
| 10e | Equipe motivée et « au travail » | Equipe impliquée et dédiée |

# La Maîtrise comme motivation permanente

- Sans maitrise tout projet est voué à l'échec

- La maîtrise de l'élaboration de logiciel se décline en
  - Programming-in-the-Small
    - Construire des briques de logiciels corrects
  - Programming-in-the-Large
    - Assembler des briques diverses
  - Programming-in-the-Variability
    - S'adapter aux changements inhérents au Génie Logiciel

- Proscrire _____

# Enjeux de la gestion de projet

- Erreur récurrente mais extrêmement coûteuse:

---

  - Aussi bien du client que du fournisseur
    - Le client ne connait pas l'informatique
      - Il croit connaitre l'informatique parce qu'il passe la journée sur un ordi
      - Il croit faire construire un immeuble alors qu'il veut une ville entière
    - Le fournisseur veut remporter le marché
      - En sous-estimant ses coûts, il n'obtient pas assez de moyens pour réussir
      - Le management ne connait pas réellement l'informatique
  - C'est un extrême car cela conduit à l'abandon des projets
    - Avant ou après avoir causé des dégâts

# Bénéfice de la gestion de projet

- Minimise les risques
  - Augmente les chances de succès
- Organise globalement des activités organisées à différentes échelles
  - Chaque équipe de programmeurs s'organise
  - Le client s'organise
  - Etc.
- La gestion de projet englobe toute l'organisation
  - évite que chacun travaille dans son coin

IUT Nantes
Pôle Sciences et technologie
Nantes Université

# GPO2
# Axe fonctionnel : cas d'utilisations

Jean-Marie Mottu
IUT de Nantes – Département Informatique

# Axe fonctionnel avant conceptuel

- La phase d'analyse et de conception, avec la production de diagramme UML a tendance à dériver vers le « conceptuel »

- Les cas d'utilisations se focalisent sur l'axe fonctionnel
- Les objectifs étant
  - Que les parties prenantes du projet se comprennent
  - De représenter le système en insistant sur son périmètre
  - D'exprimer le service rendu
  - D'identifier les conditions du fonctionnement du système

# Objectif : Que les parties prenantes du projet se comprennent

- Parties prenantes doivent se mettre d'accord
  - **Client**
    - Le client doit définir son **besoin**
  - **Fournisseur**
    - Le fournisseur doit comprendre quelle **réponse apportée**

- Avant de penser à la conception des fonctionnalités du systèmes, il faut savoir quelles utilisateurs vont s'en servir:
  - **Acteurs**

**Qui parle de qui ?**

# Objectif : Représenter le système en insistant sur son périmètre

- Le système développé, même à différentes échelles (de taille, de complexité, etc.), est des limites
  - Il ne fait pas tout seul
  - Il est dans un environnement
- Il faut définir quel est le **système développé** et quel est son **périmètre**.
  - Nommer, définir, caractériser

**De quoi parle-t-on ?**

# Objectif : exprimer le service rendu

- Identifier les cas d'utilisation
  - **Abstraction** d'une manière d'utiliser le système
  - Expression fonctionnel du **besoin**
  - Correspondent aux **fonctionnalités** du système accessibles aux acteurs
- Nombre raisonnable : 7 +/- 2 (H. A. Miller, 1958)
- Relation entre les cas d'utilisation
  - S'utilisent entre eux
  - S'étendent entre eux

**Quelles sont les fonctionnalités ?**

# Objectif : Identifier les conditions du fonctionnement du système

- Les cas d'utilisation ne sont pas actionnés n'importe comment par les acteurs, mais selon certaines conditions
  - Les cas d'utilisation
    - peuvent se spécialiser
    - peuvent s'étendre
    - peuvent s'inclure
  - Leur actionnement respecte des conditions d'entrée, de sortie, d'invariance

**Sous quelles conditions fonctionnent le système ?**

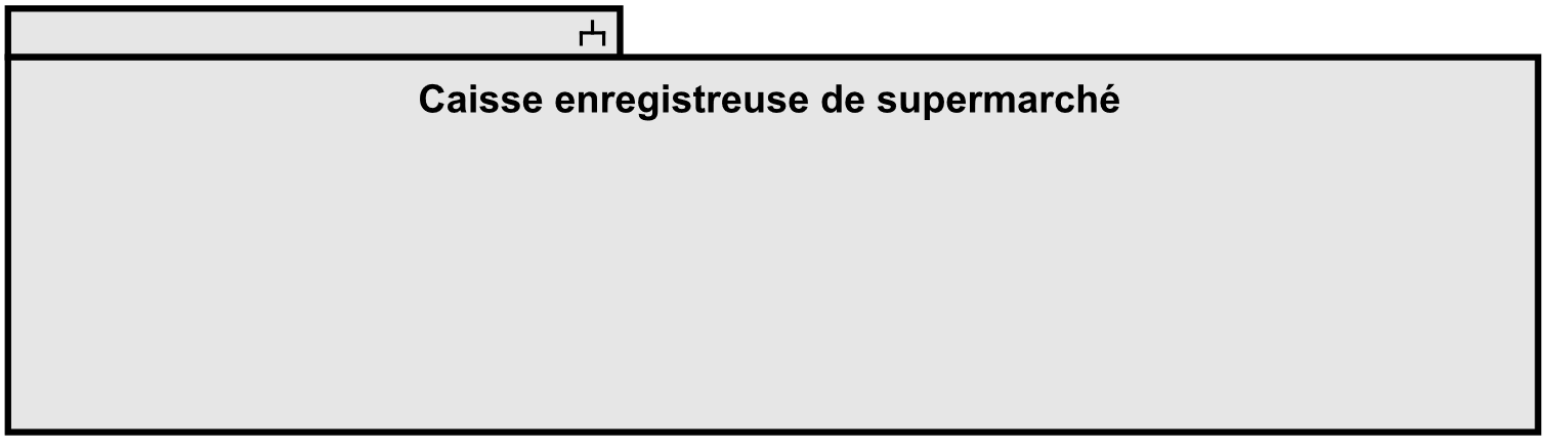
# Diagramme de cas d'utilisation DCU

- Le diagramme de cas d'utilisation est le premier diagramme UML
  - C'est un des diagrammes principaux d'UML
  - Diagrammes de comportement
    - Fonctionnel (plutôt que conceptuel)

- Créé par les différentes parties prenantes
  - Nombreuses méthodes (cf TD1 et TD2) pour le construire
  - Le nombre de concepts du diagramme reste limité
    - Cf. ~15 pages sur ~800 de la norme UML 2.5.1 de l'OMG
    - on peut itérer pour réifier les parties du système en éléments du modèle

# DCU : Représenter le système en insistant sur son périmètre

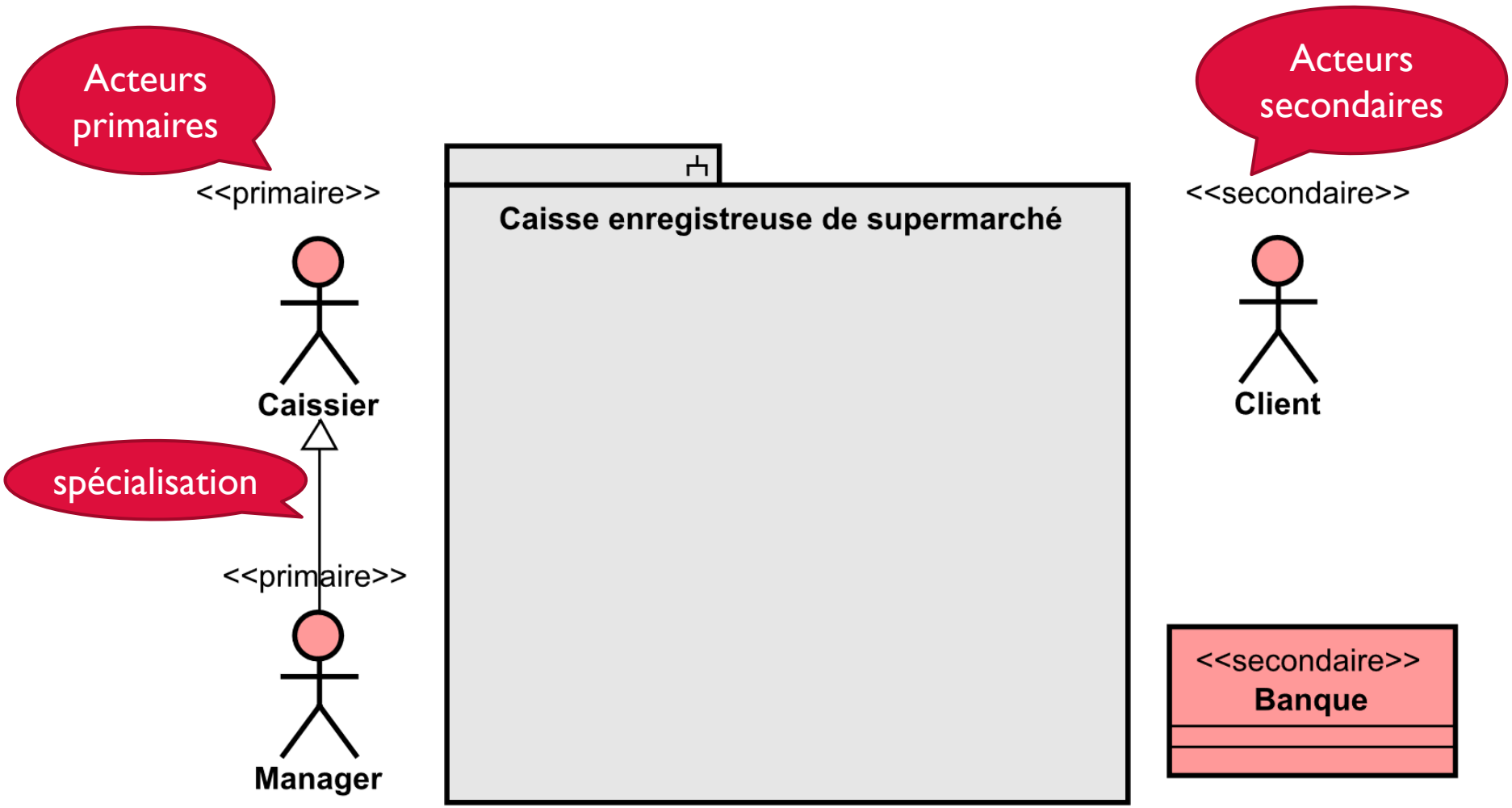
- On dessine d'abord le (sous)système
  - en le nommant
  - En identifiant ce qui est dedans, dehors
- Exemple : considérer une caisse enregistreuse d'un supermarché, tenue par des caissiers, qui s'occupe de traiter le passage en caisse et de faire payer chaque client, le superviseur les remplacent et peut annuler les erreurs:



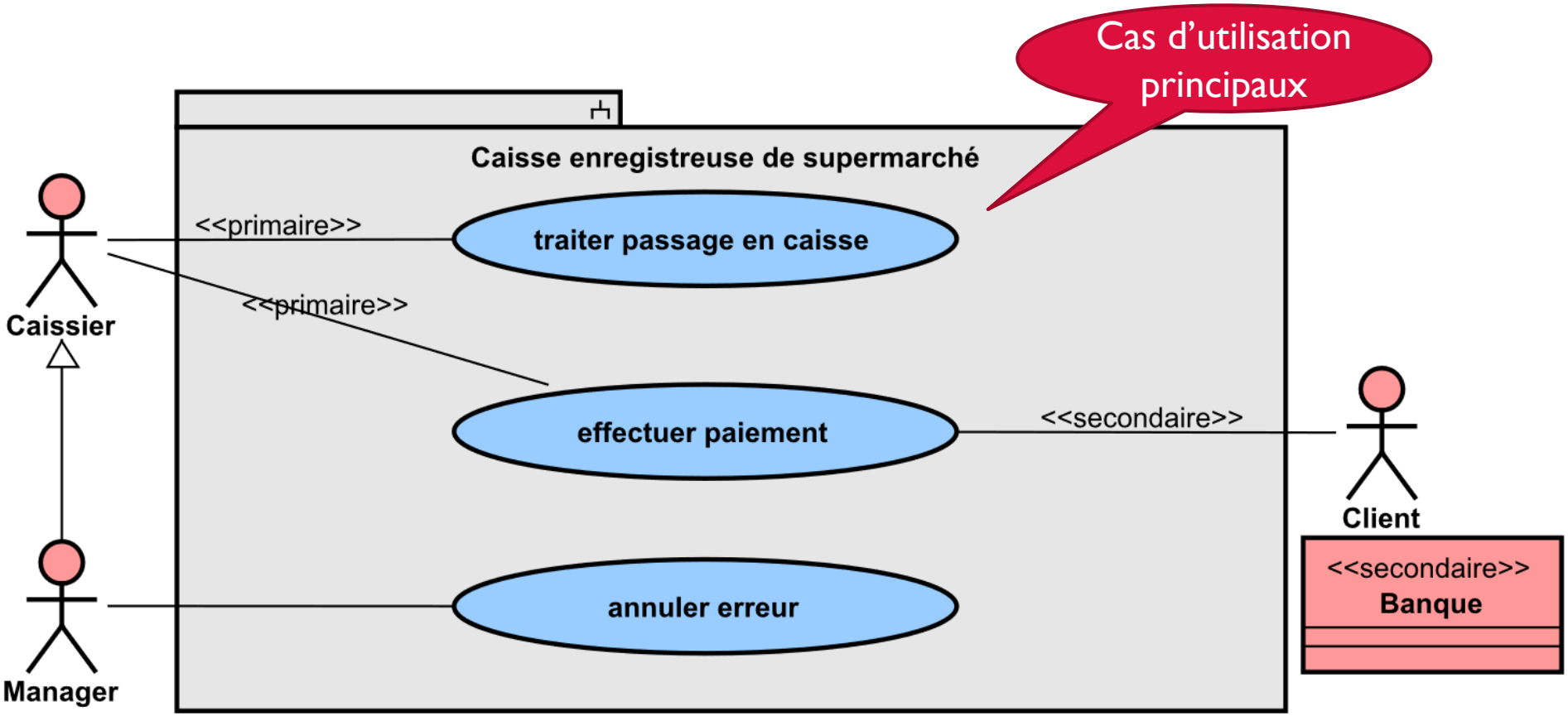
# DCU : Représenter les acteurs

- Les acteurs interagissent avec le système
  - en actionnant les cas d'utilisation
  - en étant sollicité par les cas d'utilisation
- Acteurs physiques comme logiques
  - Des humains
  - D'autres systèmes
- Peuvent être
  - principaux, secondaires en eux-mêmes et
  - avoir des relations d'héritage

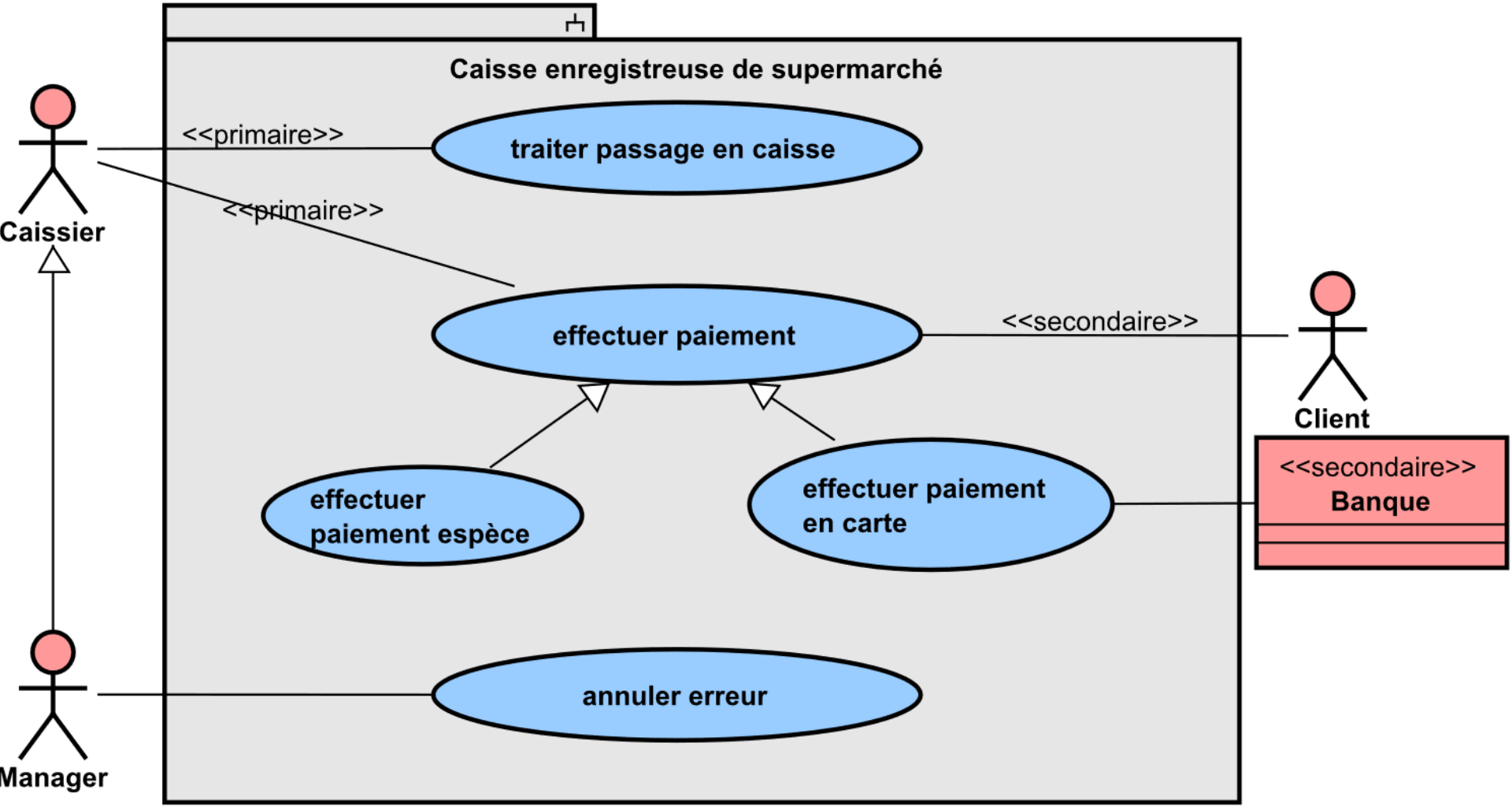
# DCU : Représenter les acteurs



# DCU : identifier les cas d'utilisation et les associer aux acteurs

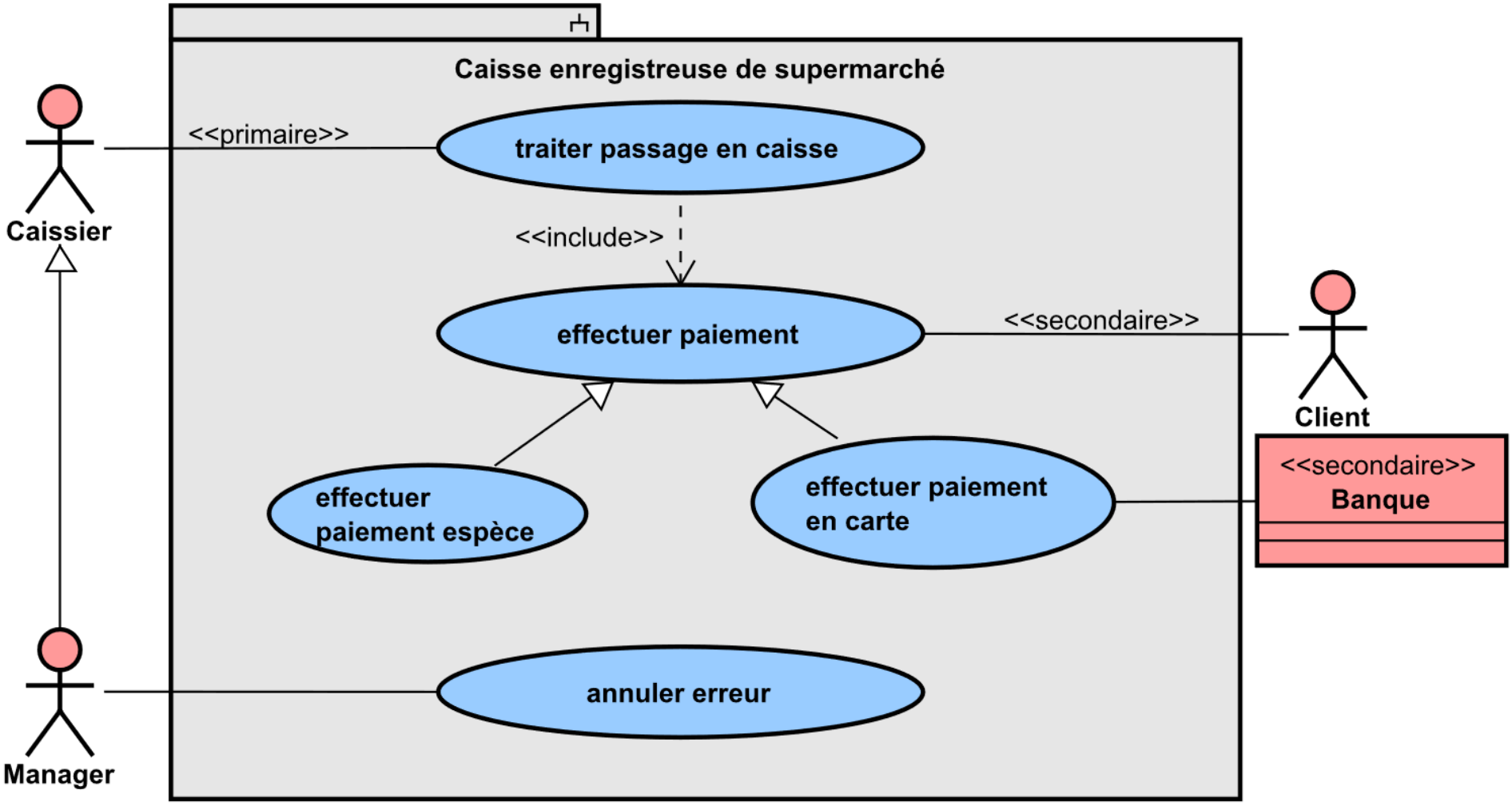


# DCU : affiner l'ensemble de cas d'utilisation
# Relation de généralisation/spécialisation

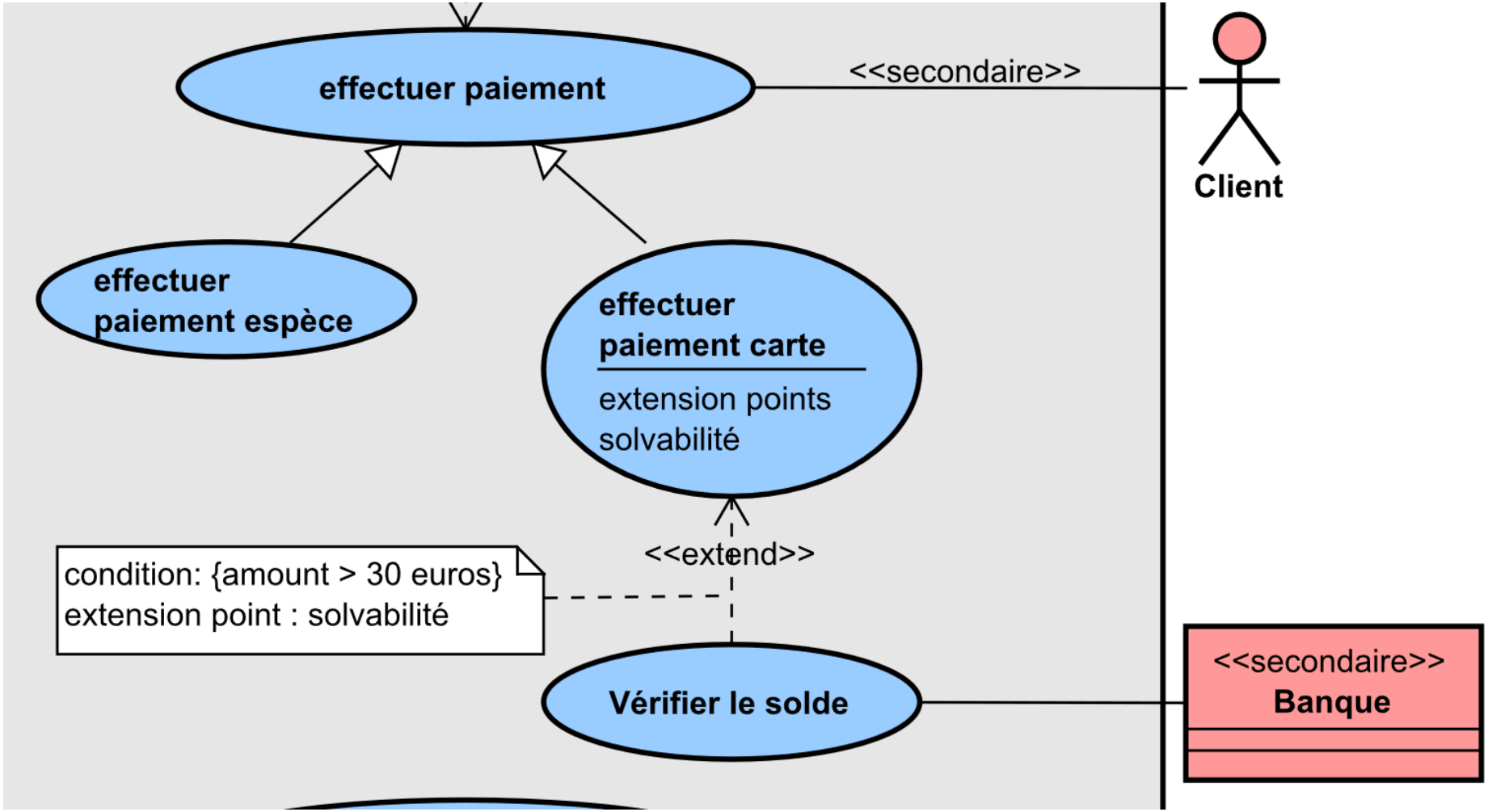


# DCU : affiner l'ensemble de cas d'utilisation
# Relation d'inclusion

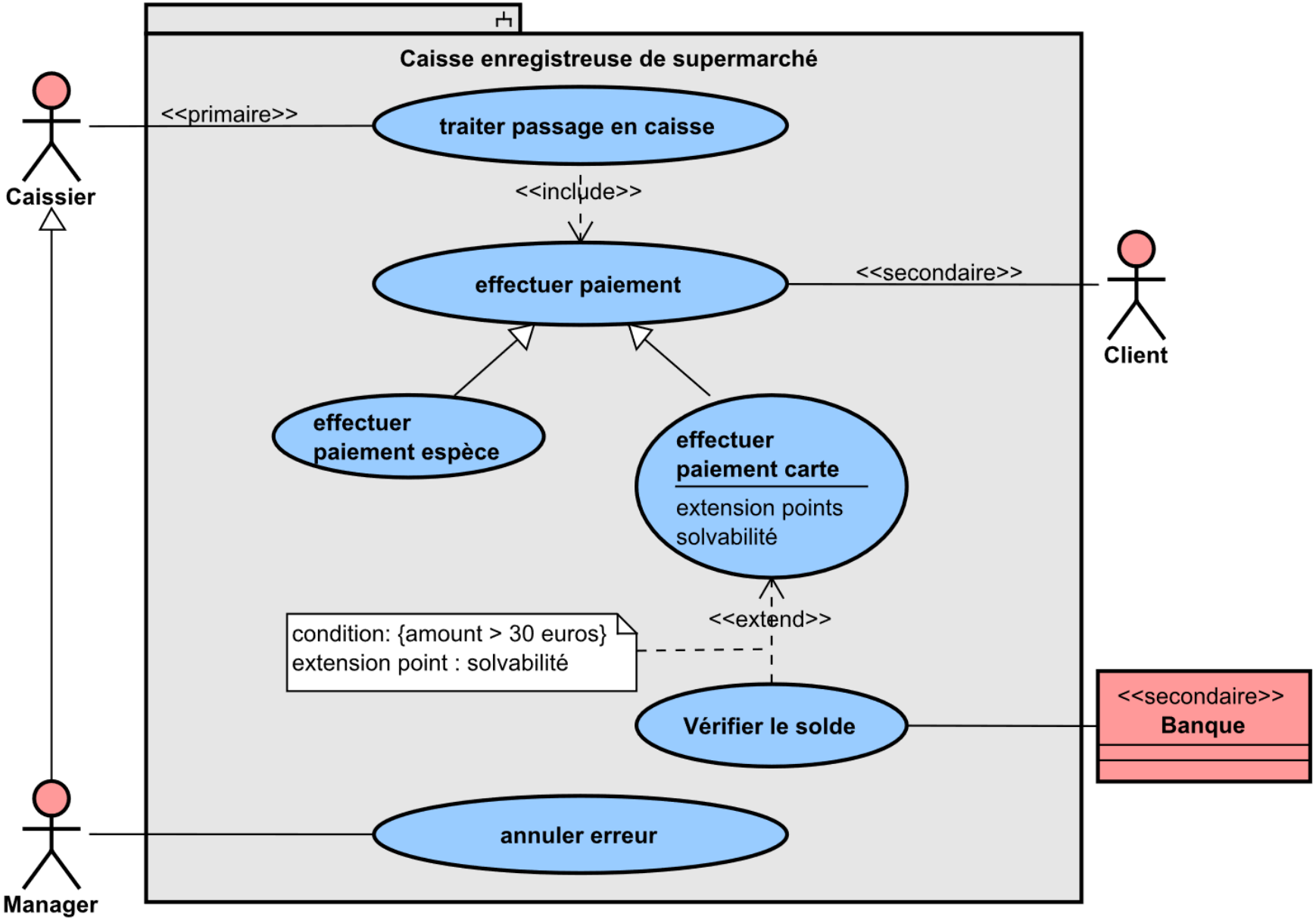
- Obligation d'inclusion d'un autre cas d'utilisation (permet d'organiser, de factoriser)



# DCU : affiner l'ensemble de cas d'utilisation
# Relation d'extension



# DCU : affiner l'ensemble de cas d'utilisation
# Relation d'extension



# DCU Préciser les conditions du fonctionnement du système

- Le diagramme de cas d'utilisation est nécessaire, précis, explicite, mais il manque de détail
  - Les informations comportementales sont finalement succinctes
  - Les diagrammes UML devraient être liées les uns aux autres
    - Cf le cours de Dev Orienté Objet
- Un cas d'utilisation offre plusieurs scénarios
  - Un scénario nominal généralement
  - Un ou plusieurs scénarios alternatifs
  - Un ou plusieurs scénarios exceptionnels (anormaux mais devant être gérés)

# DCU Préciser les conditions du fonctionnement du système : descriptif

- **Nom cas d'utilisation**
- **Responsabilité :** son objectif principal
- **Acteur(s) :** déclenchant le cas d'utilisation et impliqué(s)
- **Déclencheur :** quel(s) évènement(s) sollicite son déclenchement
- **Pré-conditions :** l'état du système nécessaire au déclenchement
- **Post-conditions :** l'état du système attendu à la fin
- **Invariant** : état interdit pendant le déroulement
- **Scénario nominal :** ce qui se passe dans le cas le plus simple
- Scénario(s) alternatif(s)
- Scénario(s) exceptionnel(s)

(Les scénarios sont directement liés aux diagrammes de séquence)