



# Introduction à UML

UE RB14

---

BONHOURE Timothé

November 21, 2024

# Introduction

---

# Qu'est-ce qu'UML ?

- UML signifie *Unified Modeling Language*.

## **Definition**

Un modèle est une abstraction de la réalité. Il simplifie un système complexe pour aider à mieux le comprendre et le développer.

# Pourquoi modéliser un système informatique ?

La modélisation est essentielle pour :

- **Compréhension et communication :**
  - Facilite la compréhension des systèmes complexes.
  - Sert de langage commun entre les équipes techniques et métiers.

# Pourquoi modéliser un système informatique ?

La modélisation est essentielle pour :

- **Compréhension et communication :**
  - Facilite la compréhension des systèmes complexes.
  - Sert de langage commun entre les équipes techniques et métiers.
- **Planification et analyse :**
  - Identifie les besoins fonctionnels et non fonctionnels.
  - Préviens les erreurs dès les premières phases du projet.

# Pourquoi modéliser un système informatique ?

La modélisation est essentielle pour :

- **Compréhension et communication :**
  - Facilite la compréhension des systèmes complexes.
  - Sert de langage commun entre les équipes techniques et métiers.
- **Planification et analyse :**
  - Identifie les besoins fonctionnels et non fonctionnels.
  - Préviend les erreurs dès les premières phases du projet.
- **Conception et maintenance :**
  - Structure le système pour une conception modulaire.
  - Crée une documentation pour la maintenance future.

# Pourquoi modéliser un système informatique ?

La modélisation est essentielle pour :

- **Compréhension et communication :**
  - Facilite la compréhension des systèmes complexes.
  - Sert de langage commun entre les équipes techniques et métiers.
- **Planification et analyse :**
  - Identifie les besoins fonctionnels et non fonctionnels.
  - Préviend les erreurs dès les premières phases du projet.
- **Conception et maintenance :**
  - Structure le système pour une conception modulaire.
  - Crée une documentation pour la maintenance future.
- **Réduction des coûts et des risques :**
  - Limite les erreurs en développant des prototypes virtuels.
  - Réduit les coûts de corrections tardives.

# Pourquoi utiliser un langage unifié comme UML ?

## Les avantages d'un langage unifié pour la modélisation :

- **Uniformité et standardisation :**
  - Assure une compréhension commune entre les parties prenantes.
  - Permet de partager des modèles indépendamment des outils ou équipes.

# Pourquoi utiliser un langage unifié comme UML ?

## Les avantages d'un langage unifié pour la modélisation :

- **Uniformité et standardisation :**
  - Assure une compréhension commune entre les parties prenantes.
  - Permet de partager des modèles indépendamment des outils ou équipes.
- **Polyvalence :**
  - Couvre différents aspects des systèmes (structurels, comportementaux, etc.).
  - Peut être utilisé pour modéliser des systèmes variés (informatique, médecine, logistique...).

# Pourquoi utiliser un langage unifié comme UML ?

## Les avantages d'un langage unifié pour la modélisation :

- **Uniformité et standardisation :**
  - Assure une compréhension commune entre les parties prenantes.
  - Permet de partager des modèles indépendamment des outils ou équipes.
- **Polyvalence :**
  - Couvre différents aspects des systèmes (structurels, comportementaux, etc.).
  - Peut être utilisé pour modéliser des systèmes variés (informatique, médecine, logistique...).
- **Interopérabilité :**
  - Favorise l'intégration avec des outils logiciels de conception.
  - Facilite la réutilisation de modèles et de diagrammes.

# Pourquoi utiliser un langage unifié comme UML ?

## Les avantages d'un langage unifié pour la modélisation :

- **Uniformité et standardisation :**
  - Assure une compréhension commune entre les parties prenantes.
  - Permet de partager des modèles indépendamment des outils ou équipes.
- **Polyvalence :**
  - Couvre différents aspects des systèmes (structurels, comportementaux, etc.).
  - Peut être utilisé pour modéliser des systèmes variés (informatique, médecine, logistique...).
- **Interopérabilité :**
  - Favorise l'intégration avec des outils logiciels de conception.
  - Facilite la réutilisation de modèles et de diagrammes.
- **Amélioration de la collaboration :**
  - Réduit les ambiguïtés grâce à des notations standardisées.
  - Facilite le travail collaboratif dans des équipes multidisciplinaires.

## Conclusion de la partie 1

UML est un outil essentiel pour :

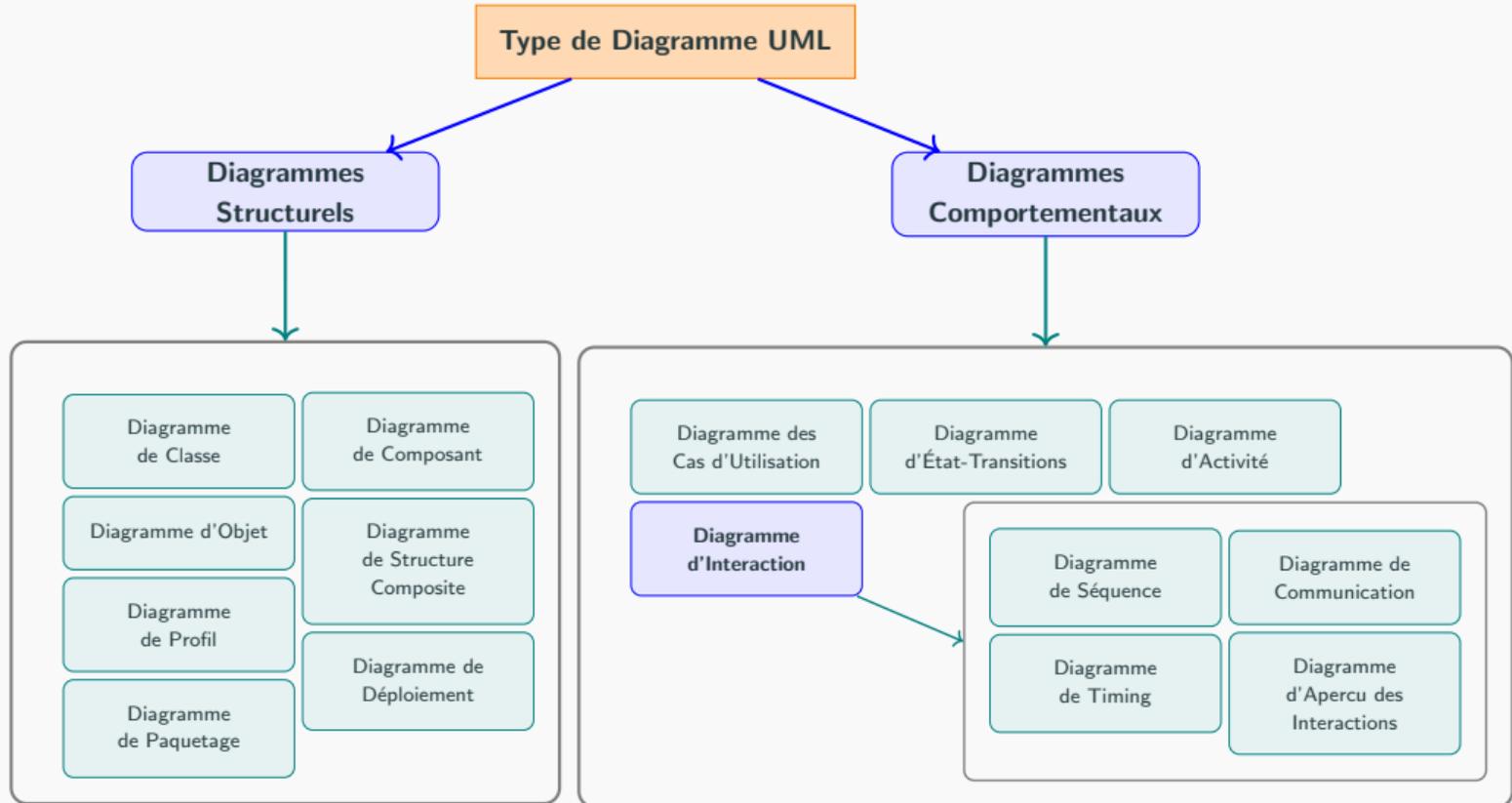
- Simplifier la communication.
- Structurer les processus médicaux.
- Travailler efficacement entre métiers différents.

Nous allons maintenant explorer les types de diagrammes UML.

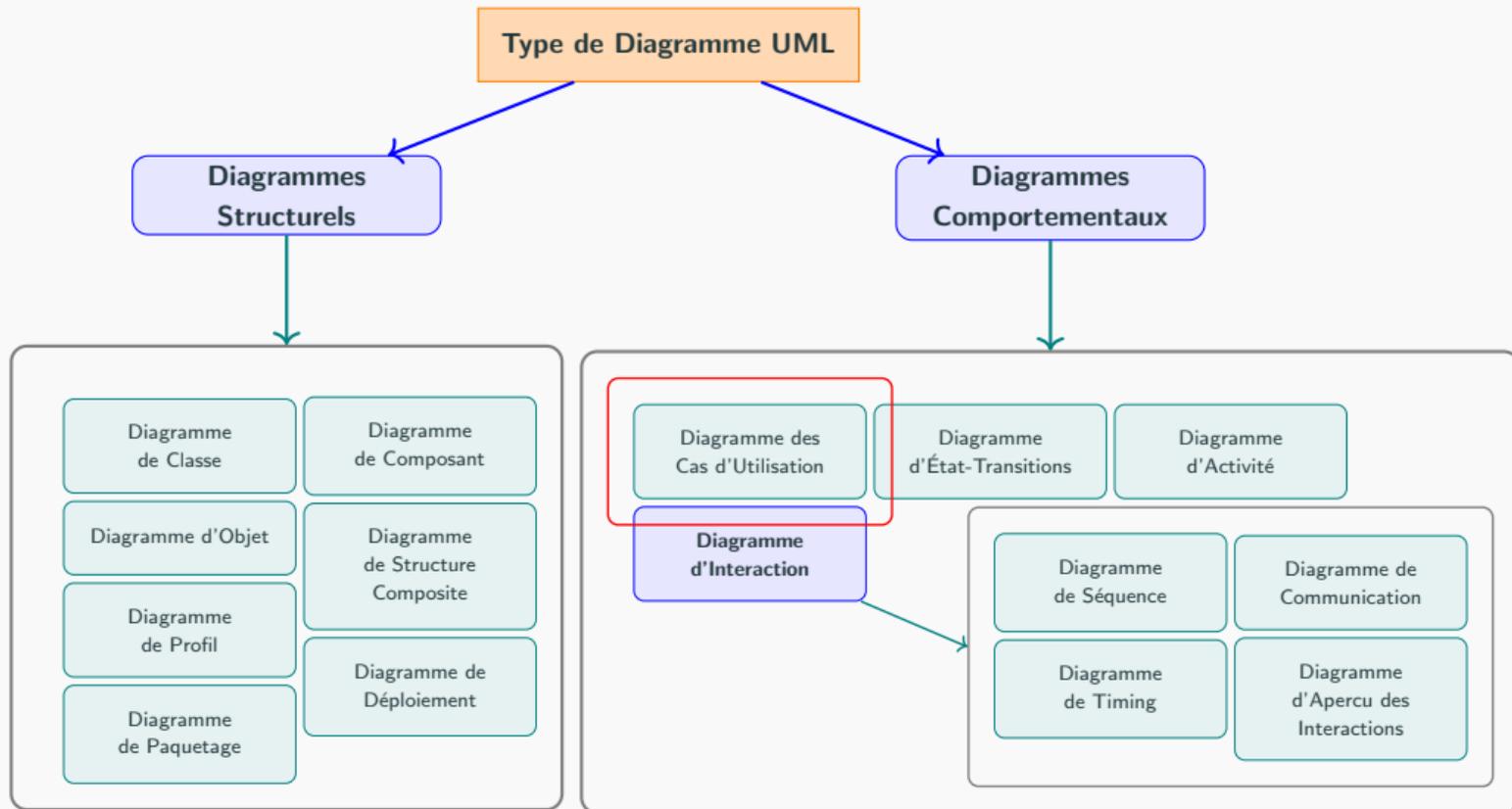
# Survol des types de diagrammes UML

---

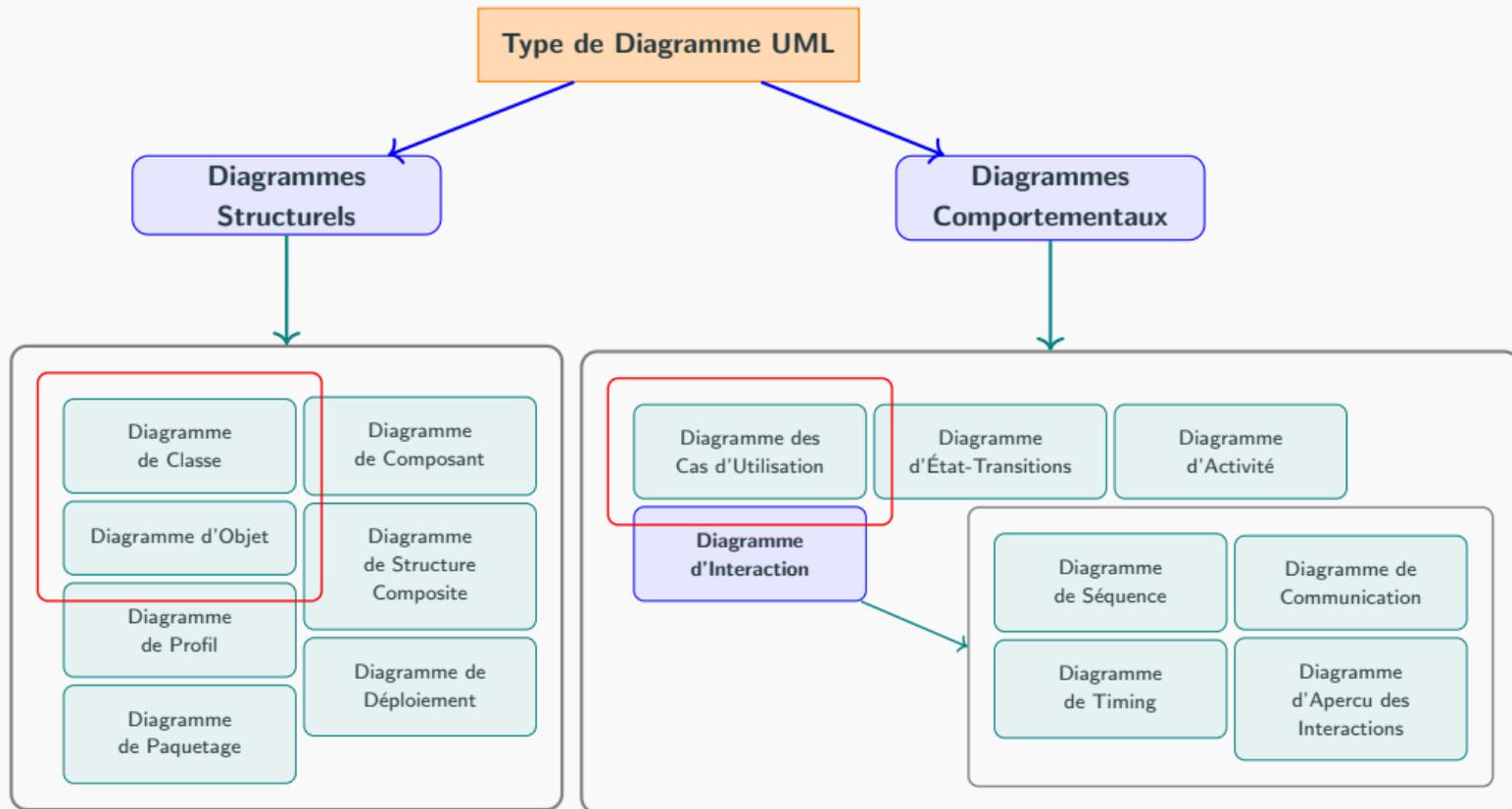
# Les types de diagrammes UML : Vue d'ensemble



# Les types de diagrammes UML : Vue d'ensemble



# Les types de diagrammes UML : Vue d'ensemble



## Diagramme de cas d'utilisation

---

## Le Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation est un type de diagramme UML qui décrit les interactions entre un système et ses utilisateurs (ou d'autres systèmes). Il permet de capturer les exigences fonctionnelles du système et d'illustrer comment les utilisateurs interagissent avec celui-ci.

## Le Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation est un type de diagramme UML qui décrit les interactions entre un système et ses utilisateurs (ou d'autres systèmes). Il permet de capturer les exigences fonctionnelles du système et d'illustrer comment les utilisateurs interagissent avec celui-ci.

- **Acteurs** : Représentent les utilisateurs ou systèmes externes qui interagissent avec le système.
- **Cas d'utilisation** : Décrivent les actions ou services fournis par le système aux acteurs.
- **Relations** : Montrent les interactions entre acteurs et cas d'utilisation.

# Le Diagramme de Cas d'Utilisation

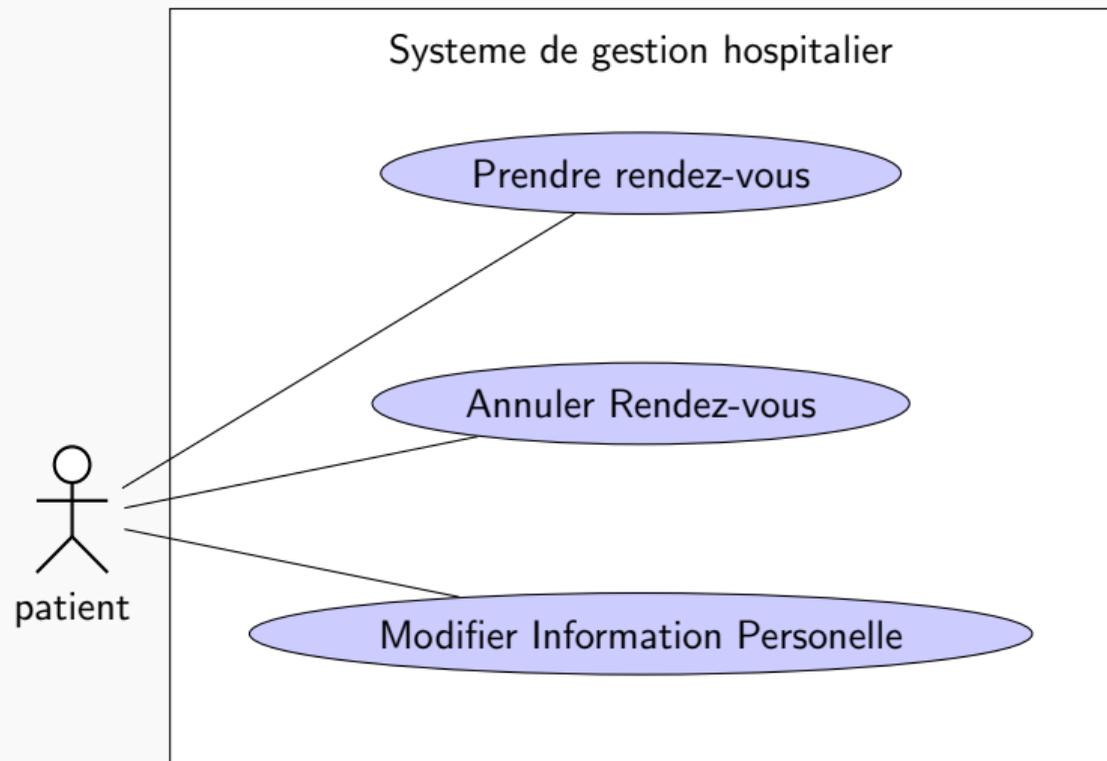
Le diagramme de cas d'utilisation est un type de diagramme UML qui décrit les interactions entre un système et ses utilisateurs (ou d'autres systèmes). Il permet de capturer les exigences fonctionnelles du système et d'illustrer comment les utilisateurs interagiront avec celui-ci.

- **Acteurs** : Représentent les utilisateurs ou systèmes externes qui interagissent avec le système.
- **Cas d'utilisation** : Décrivent les actions ou services fournis par le système aux acteurs.
- **Relations** : Montrent les interactions entre acteurs et cas d'utilisation.

## But principal

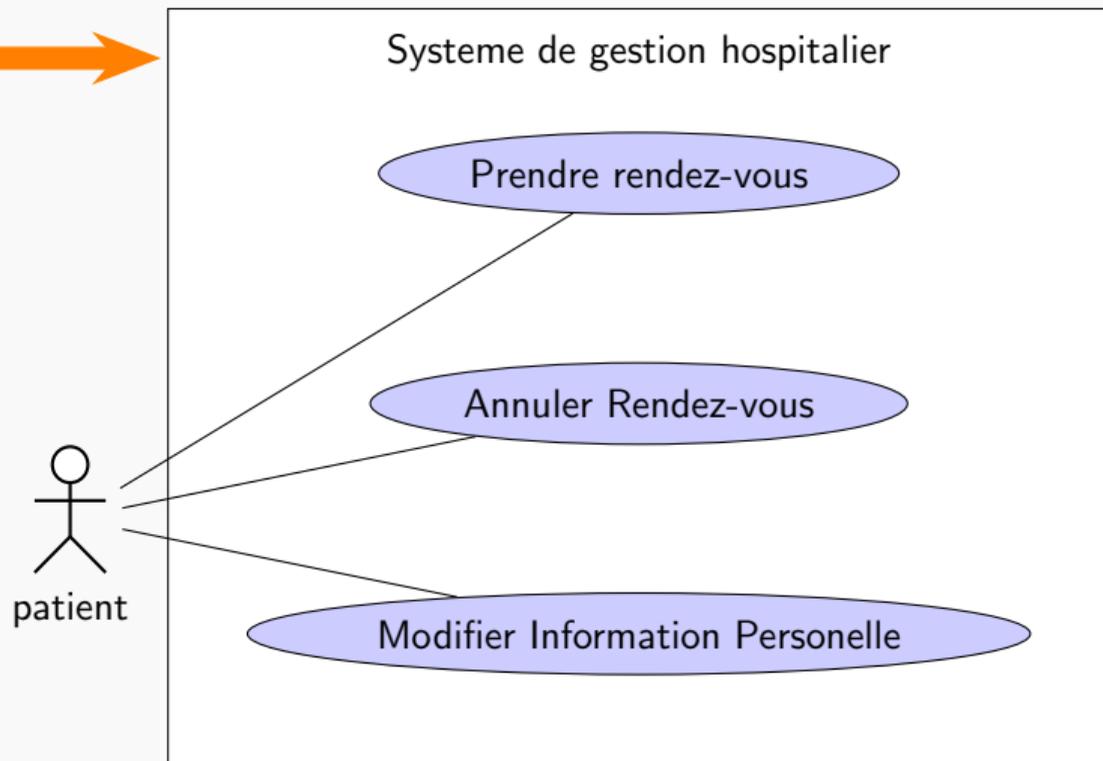
Le diagramme de cas d'utilisation sert à **définir les fonctionnalités du système** du point de vue de l'utilisateur, ce qui permet de clarifier les besoins du système et de servir de base pour la conception et les tests.

# Mots clefs

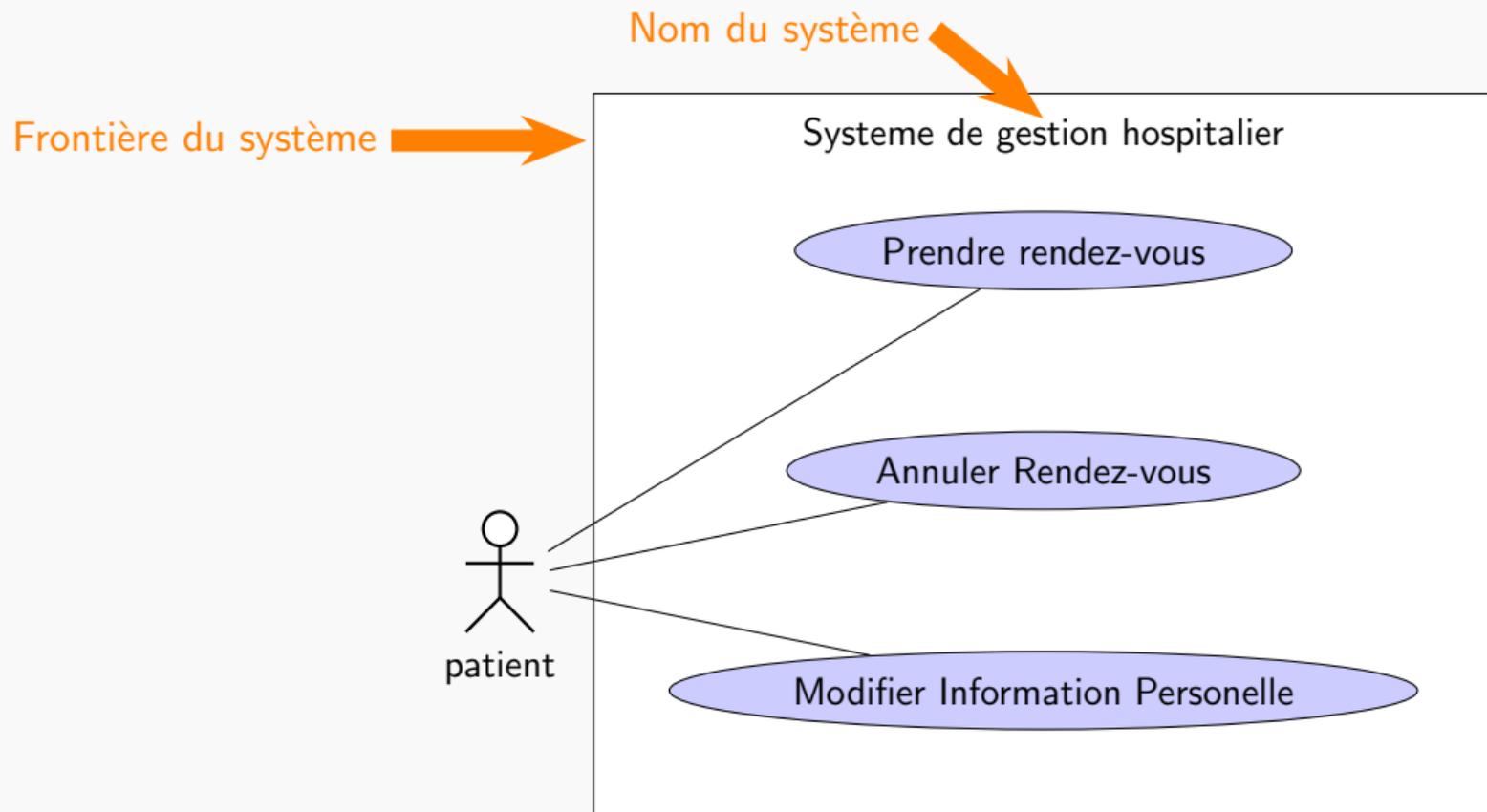


# Mots clefs

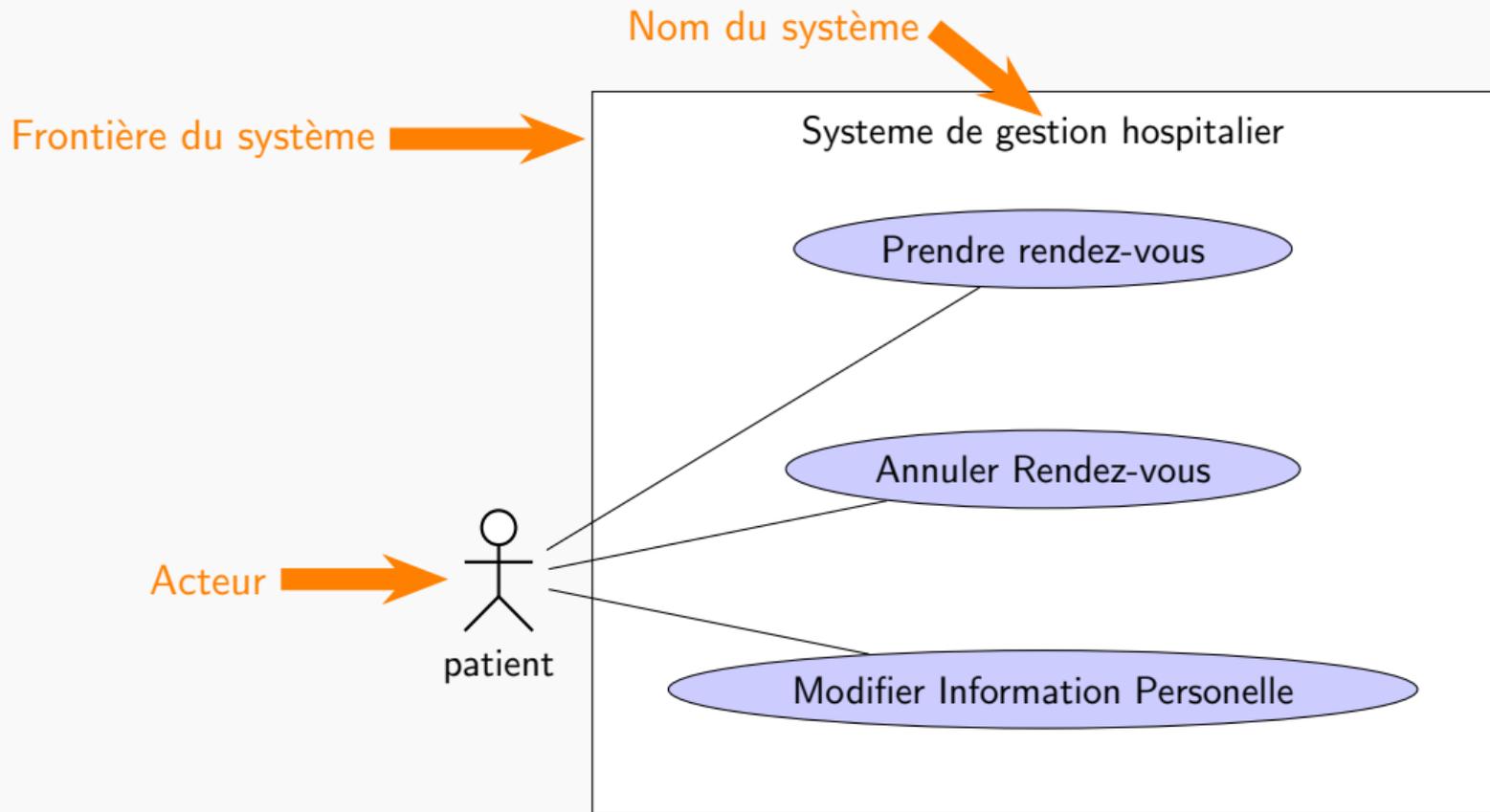
Frontière du système 



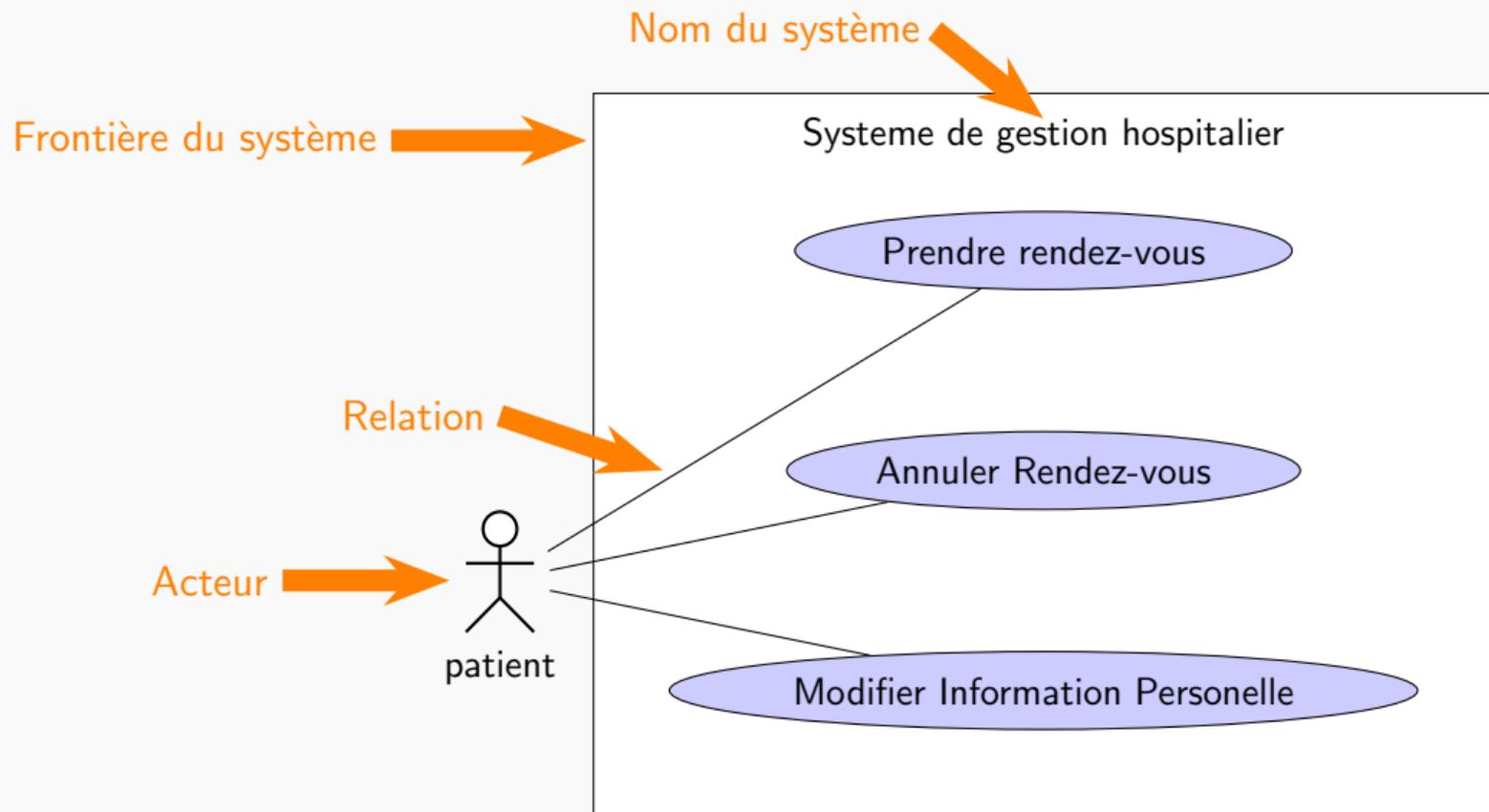
# Mots clefs

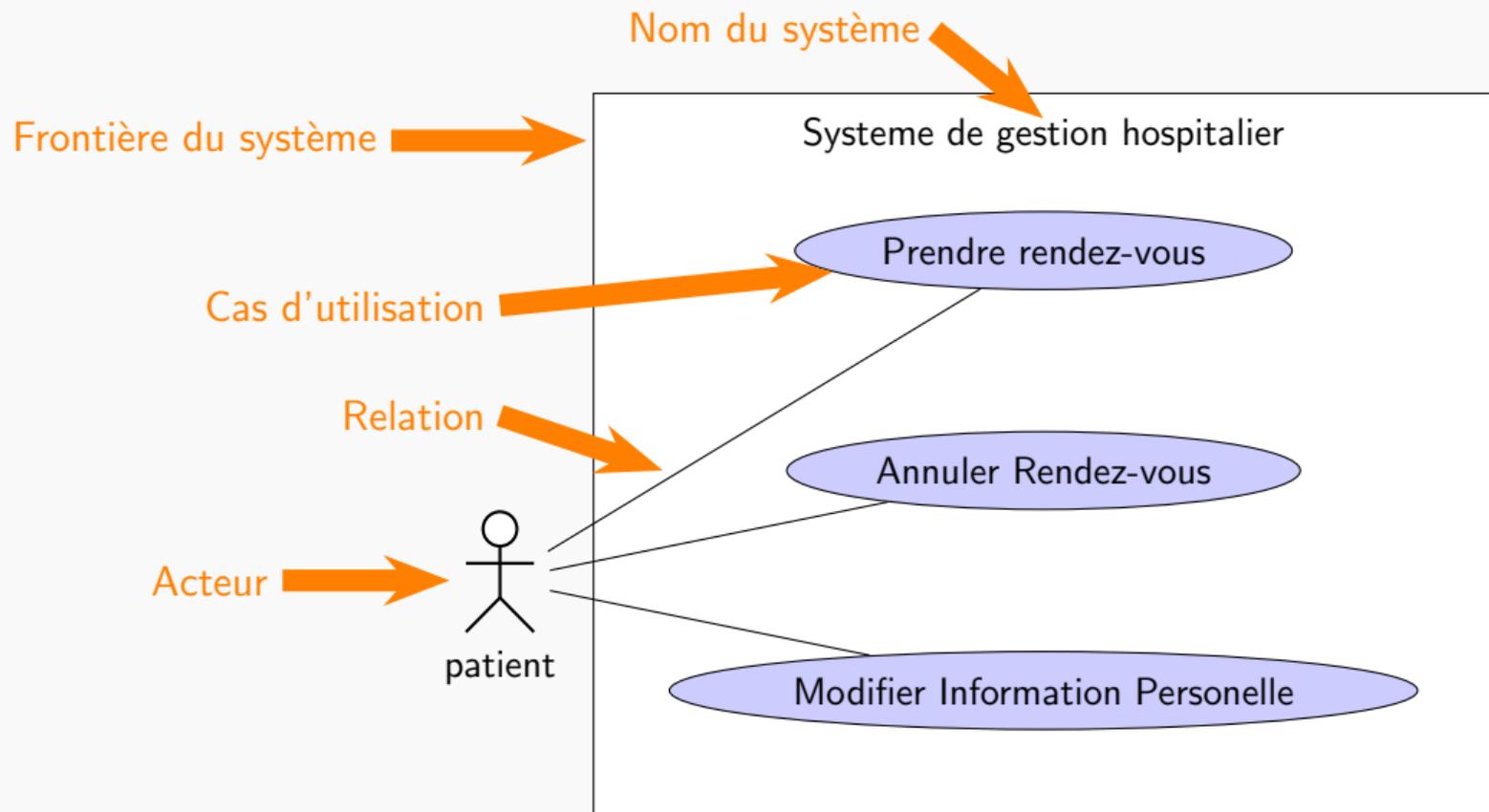


# Mots clefs

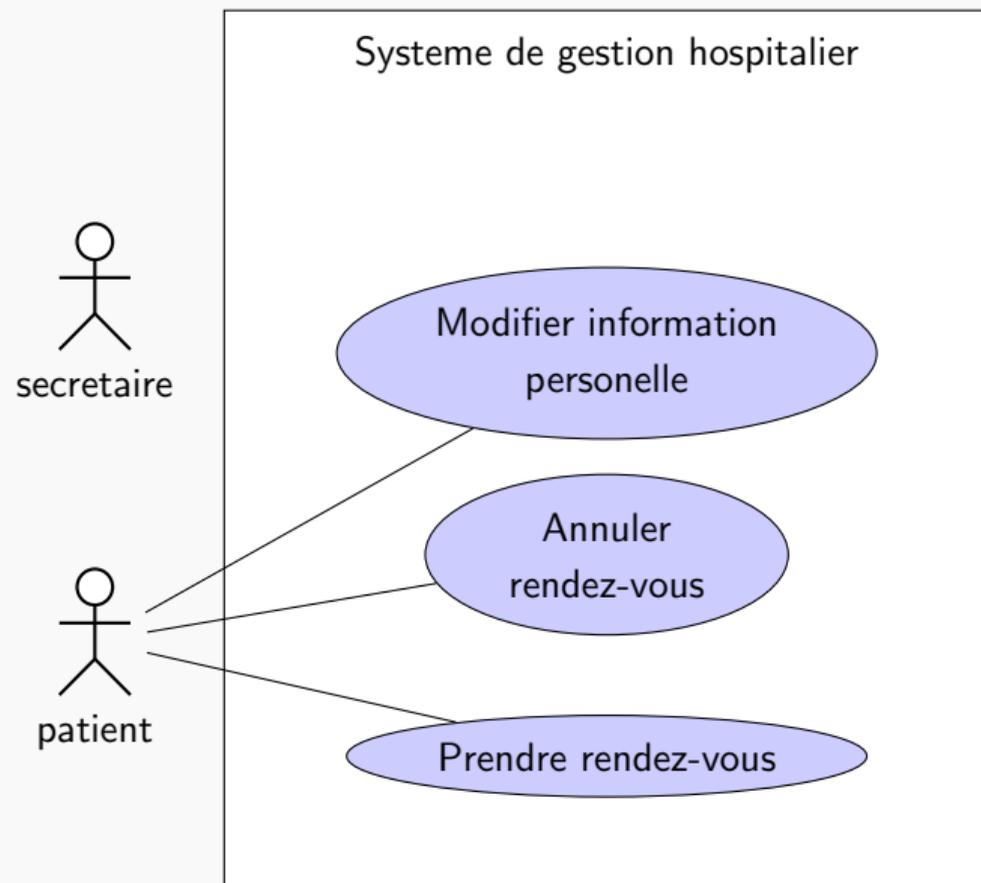


# Mots clefs

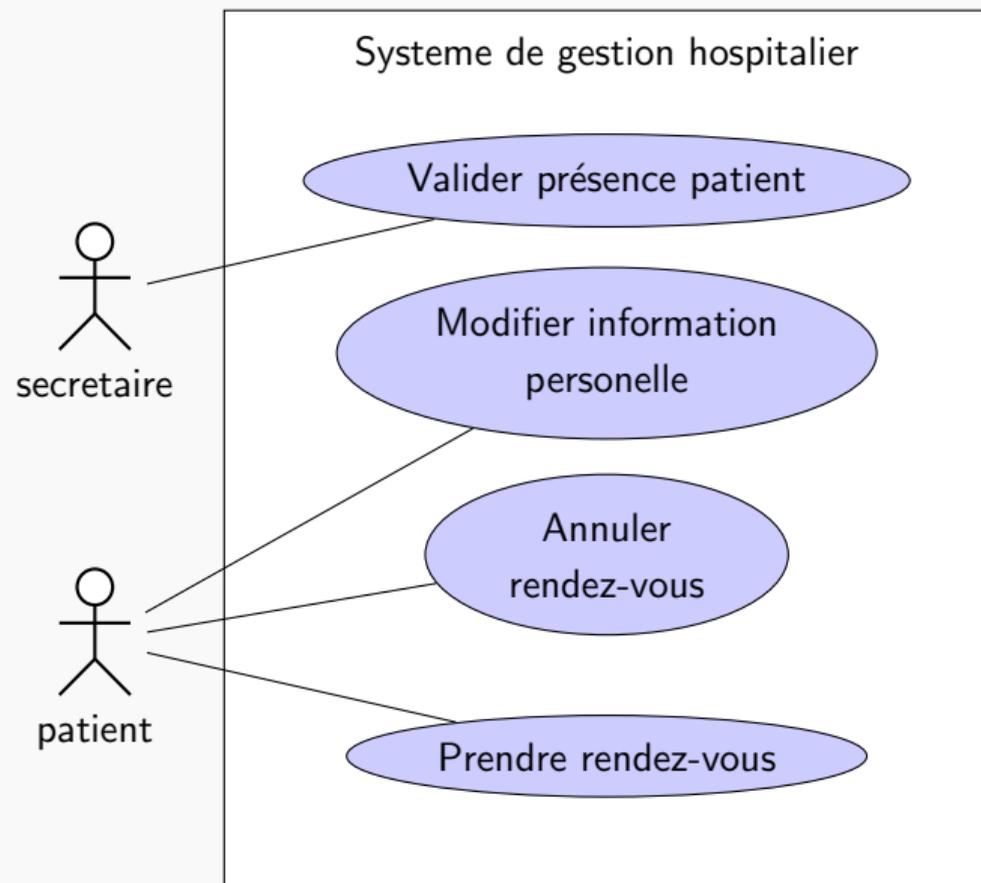




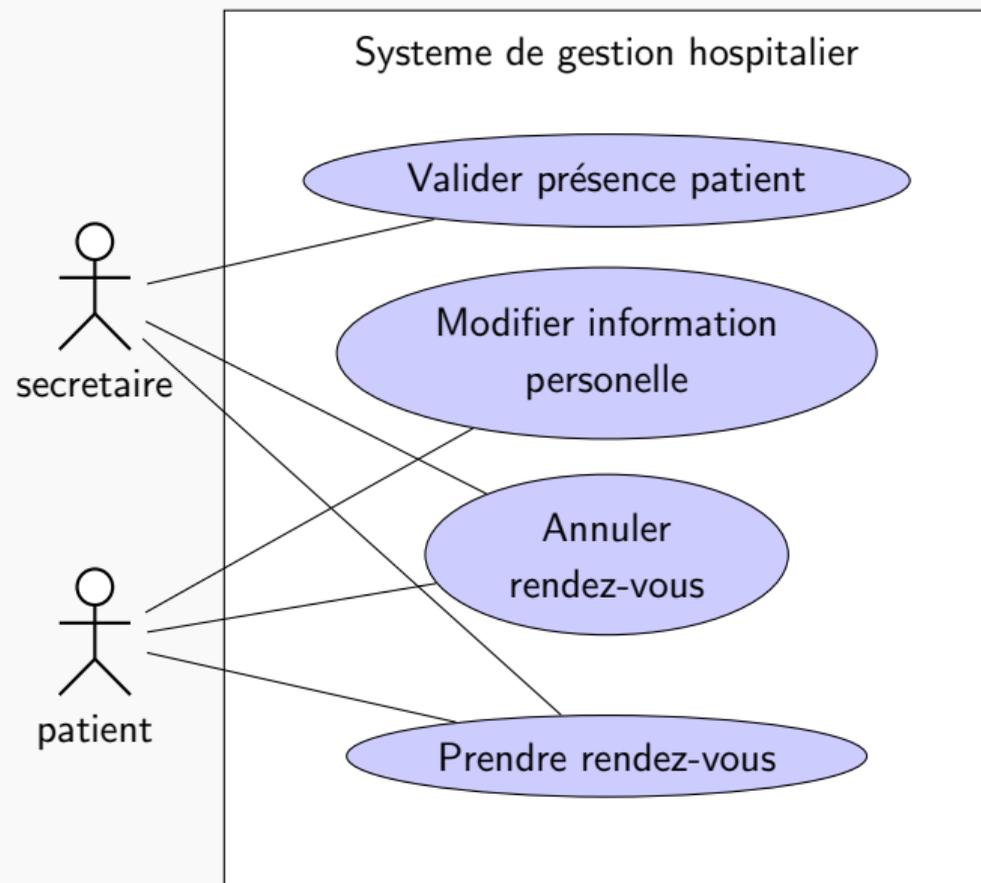
# Acteurs multiples



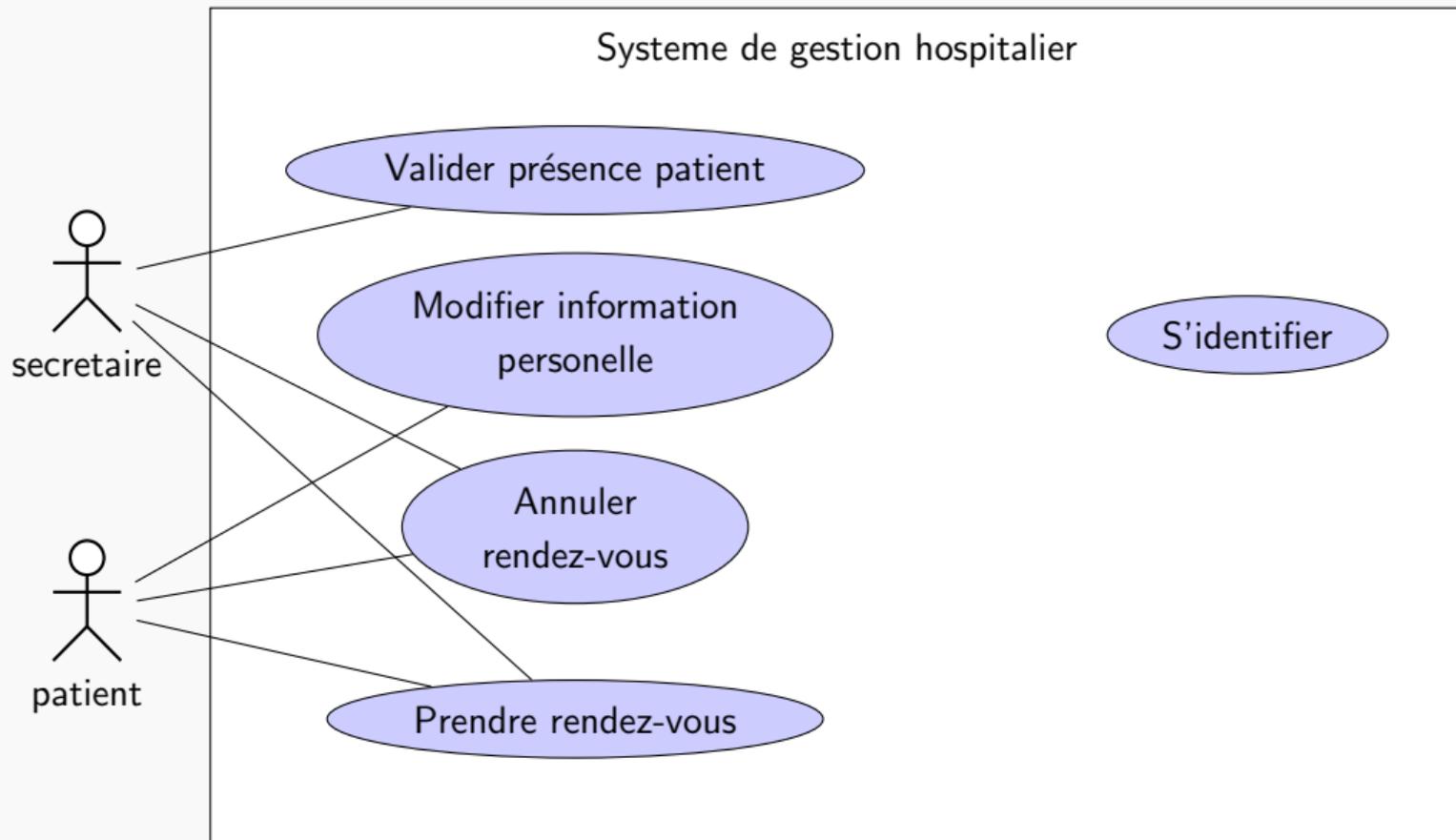
# Acteurs multiples



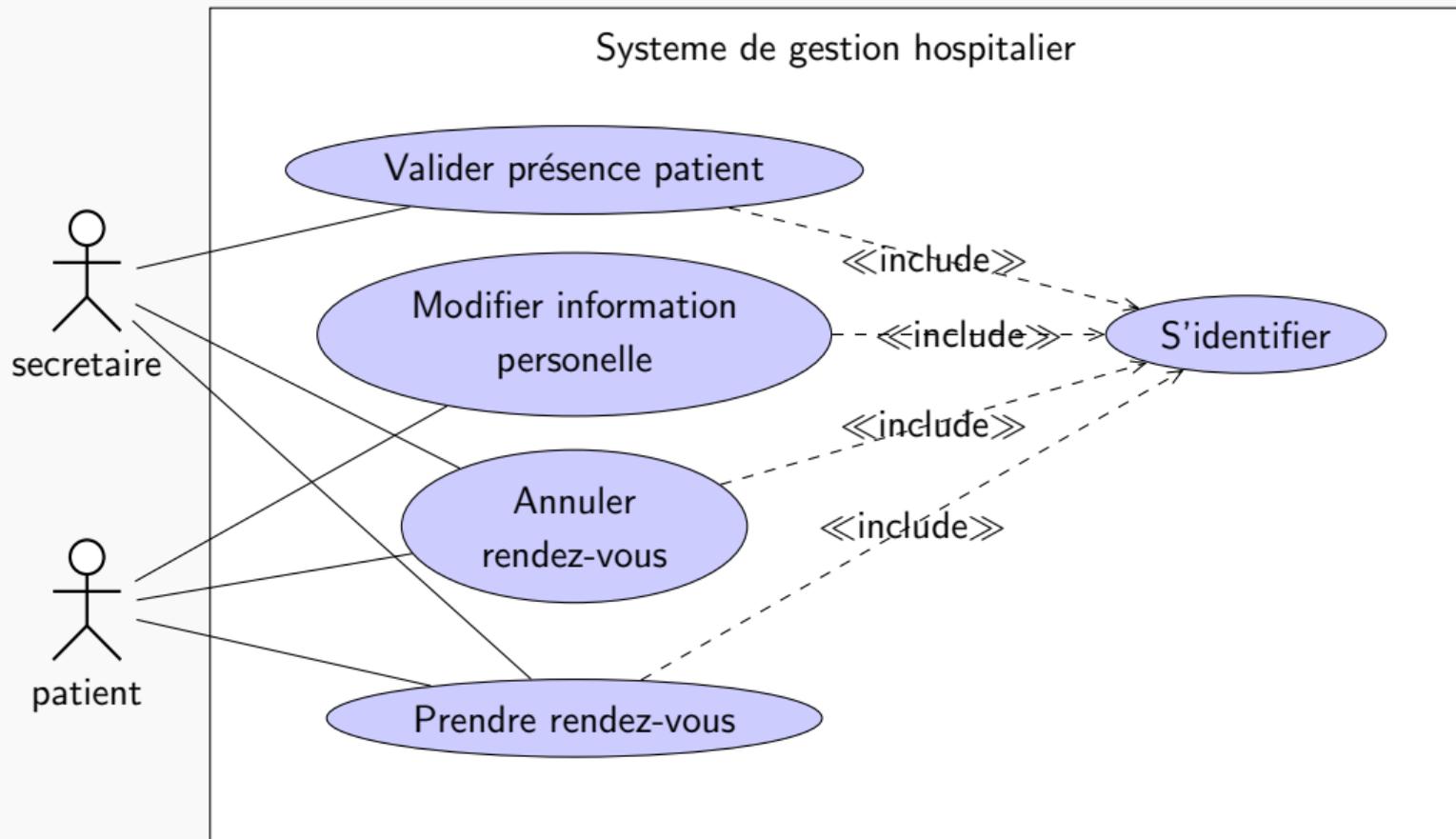
# Acteurs multiples



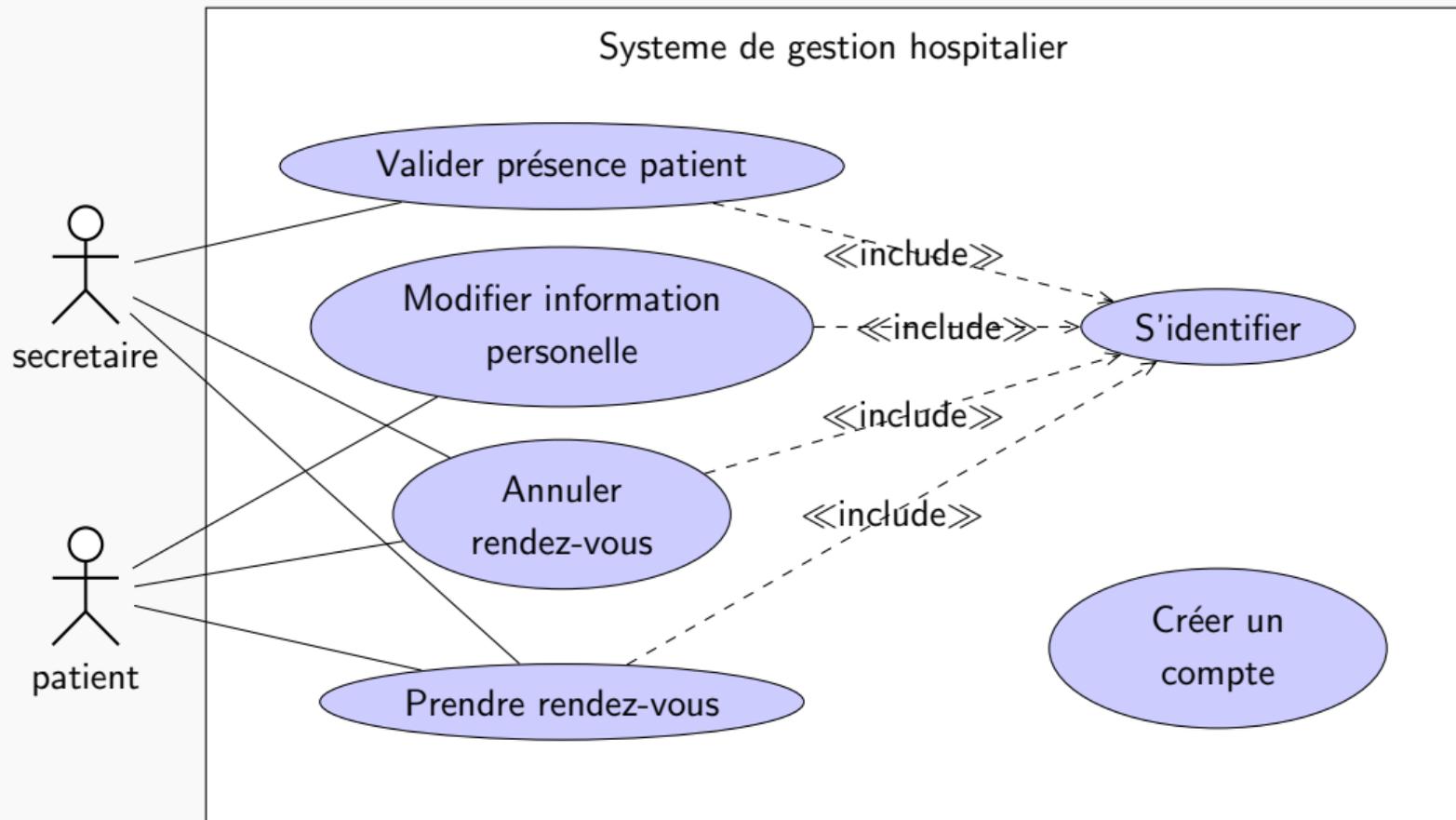
# Inclusion



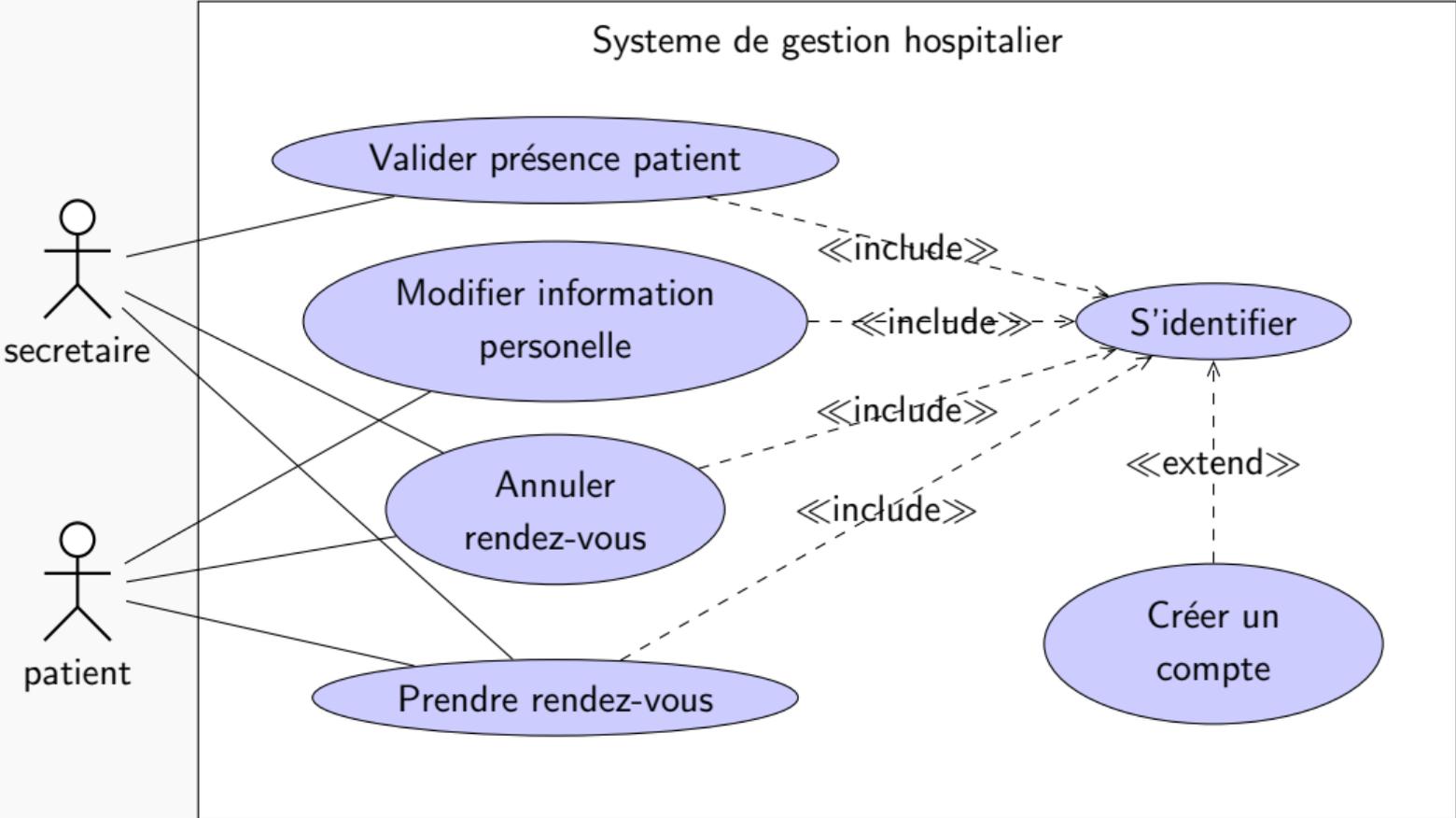
# Inclusion



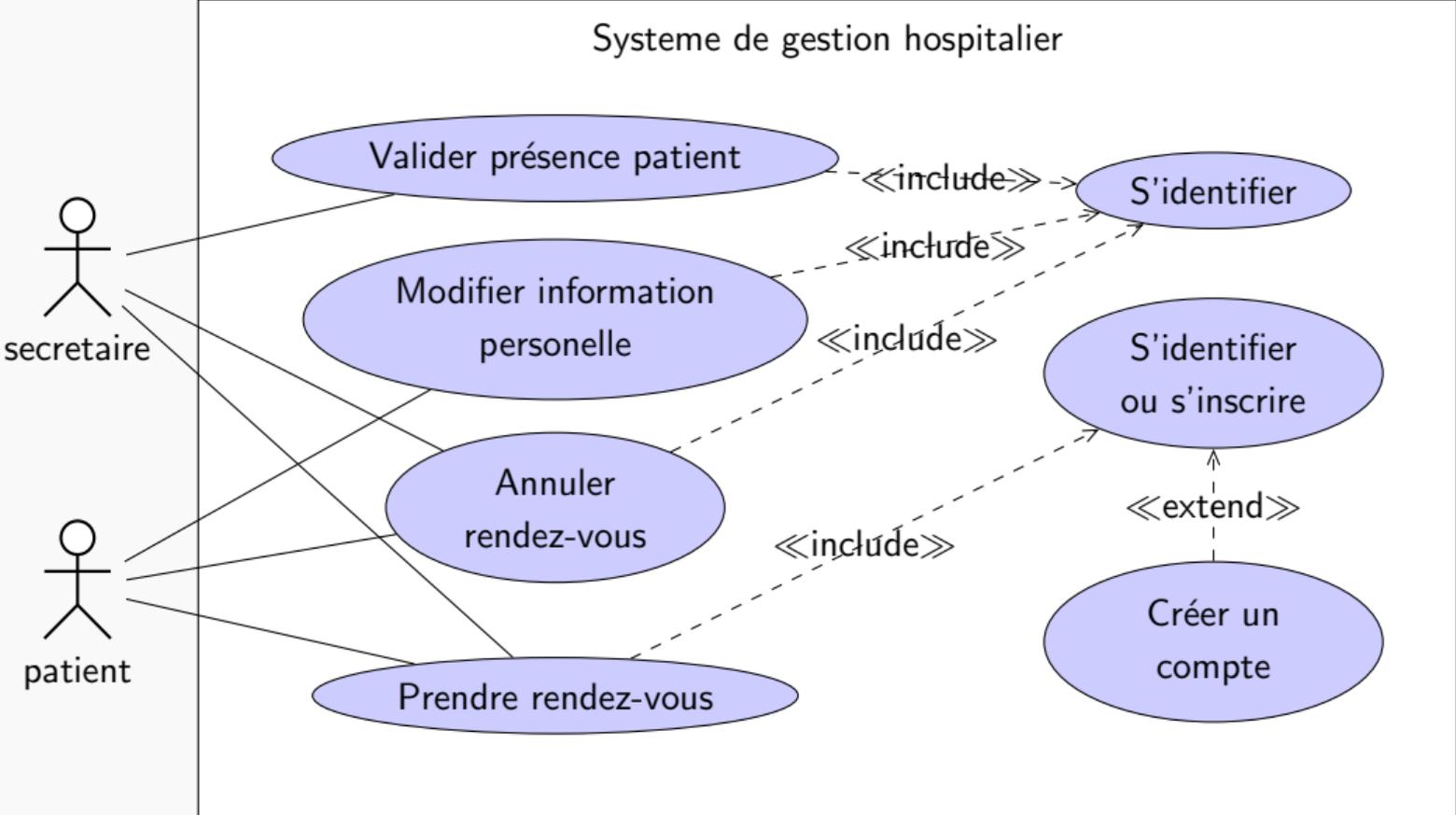
# Extension



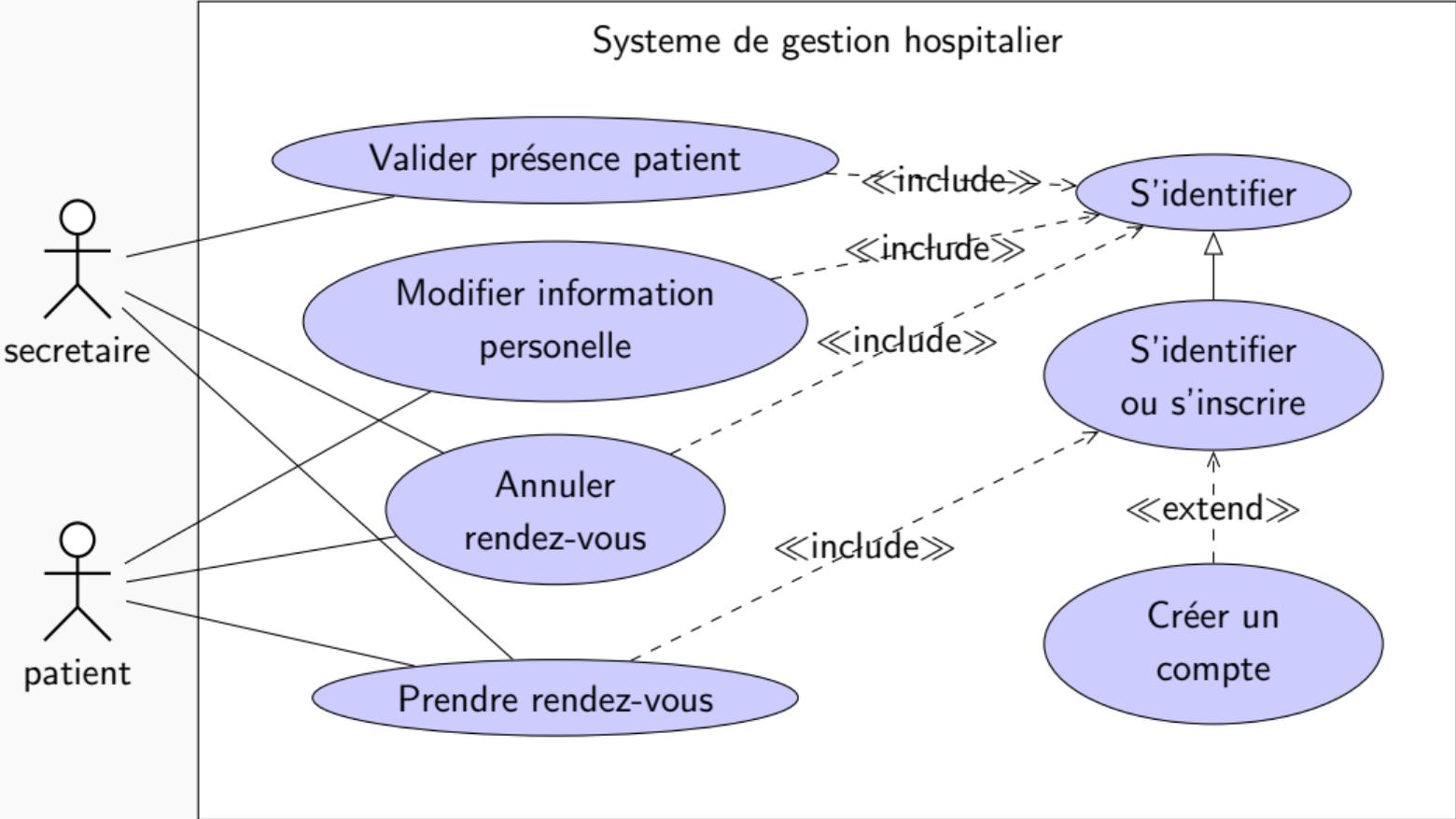
# Extension



# Généralisation



# Généralisation



## Description du cas d'utilisation : S'identifier

- **Nom du cas d'utilisation** : S'identifier.
- **Acteurs** : Utilisateur
- **But** : Permettre à l'utilisateur de se connecter à son compte existant
- **Préconditions** : L'utilisateur doit avoir un compte.
- **Postconditions** : L'utilisateur accède à son espace personnelle.
- **Scénario principal** :
  1. L'utilisateur accède à la page de connexion.
  2. Point d'extension: s'inscrire.
  3. L'utilisateur entre son identifiant et mot de passe.
  4. Le système vérifie la validité des informations.
  5. Si valides, l'utilisateur accède à son tableau de bord.
- **Extension (S'inscrire)** :
  1. Si l'utilisateur n'a pas de compte, il peut s'inscrire.
  2. Le système redirige vers le formulaire d'inscription.
  3. L'utilisateur peut se connecter immédiatement après l'inscription.

## Description du cas d'utilisation : S'identifier avec extension S'inscrire

- **Nom du cas d'utilisation** : S'identifier ou s'inscrire
- **Acteurs** : Utilisateur
- **But** : Permettre à l'utilisateur de se connecter à son compte existant ou le créer.
- **Préconditions** : L'utilisateur doit avoir un compte ou choisir de s'inscrire.
- **Postconditions** : L'utilisateur accède à son espace personnelle.
- **Scénario principal** :
  - Ajout entre 1 et 2 : Point d'extension: s'inscrire.
- **Extension (S'inscrire)** :
  1. Si l'utilisateur n'a pas de compte, il peut s'inscrire.
  2. Le système redirige vers le formulaire d'inscription.
  3. L'utilisateur peut se connecter immédiatement après l'inscription.

## Cas d'Utilisation - Prendre Rendez-vous

**Nom** : Prendre Rendez-vous

**Acteur principal** : Utilisateur

**Objectif principal** : Permettre à un utilisateur de prendre un rendez-vous.

**Préconditions** :

- L'utilisateur et le patient sont enregistré dans le système.
- Le médecin est disponible.

**Postconditions** : Un rendez-vous est enregistré, le patient reçoit une confirmation.

**Description** :

1. L'utilisateur se connecte au système.
2. L'utilisateur sélectionne un médecin, un patient et une date.
3. Le système confirme la validité du rendez-vous.
4. Le patient reçoit une confirmation du rendez-vous par mail.

## Cas d'Utilisation - Annuler Rendez-vous

**Nom** : Annuler Rendez-vous

**Acteur principal** : Utilisateur

**Objectif principal** : Annuler un rendez-vous existant.

**Préconditions** :

- Le patient a un rendez-vous enregistré.
- L'utilisateur est enregistré.
- L'utilisateur a accès au rendez-vous du patient.

**Postconditions** : Le rendez-vous est annulé, le créneau est libéré.

**Description** :

1. L'utilisateur s'identifie.
2. L'utilisateur sélectionne le rendez-vous à annuler.
3. Le rendez-vous est annulé et une confirmation est envoyée au patient.