

# ingo

CHRISTIAN HUEMER

MARION SCHOLZ

**Objektorientierte Modellierung mit UML**  
**Teil V - Anwendungsfalldiagramm**

# Anwendungsfalldiagramm

## Das Anwendungsfalldiagramm



- Use Cases (= Anwendungsfälle) sind Ausgangspunkt vieler objekt-orientierter Entwicklungsmethoden.
- Oft auch Basiskonzept, das sich über den kompletten Analyse- und Designprozess hinweg spannt.
- Ausgangsfragen für den Einsatz von Anwendungsfällen:
  - Warum verwendet man Anwendungsfälle?
  - Wie sehen Anwendungsfälle aus?
  - Was macht man mit Anwendungsfällen, wenn man sie einmal definiert hat?
- Anwendungsfälle konzentrieren sich auf das fundamentale Problem bei der Entwicklung eines Systems: Entwicklung einer Lösung für Kunden bzw. AnwenderInnen, die sich diese auch gewünscht haben.

# Use Case (2/3)



- Anwendungsfälle repräsentieren die Anforderungen der Kund\*innen
- „Ein Anwendungsfall ist eine Sequenz von Transaktionen innerhalb eines Systems, deren Aufgabe es ist, einen für einzelne Akteur\*innen (Anwender\*innen) identifizierbaren Nutzen zu erzeugen.“ [Ivar Jacobson]
- Akteur\*innen interagieren mit dem System im Kontext der Anwendungsfälle
- Akteur\*in:
  - Rolle, die jemand oder etwas einnimmt und die in Beziehung zum Geschäftsbereich steht, oder
  - Alles, das mit dem System interagiert

# Use Case (3/3)



- Transaktionen innerhalb eines Systems implizieren, dass Akteur\*innen eine Reihe von Möglichkeiten geboten wird um mit dem System zu kommunizieren und dass durch sie ein messbarer Nutzen erzeugt wird.
- Ein messbarer Nutzen impliziert, dass die Ausführung einer Transaktion eine sichtbare, quantifizierbare und/oder qualifizierbare Auswirkung auf jene Dinge hat, die außerhalb des Systems liegen, im speziellen auf Akteur\*innen.

# Bsp.: Studienabteilung

- **System** (Was wird beschrieben?)

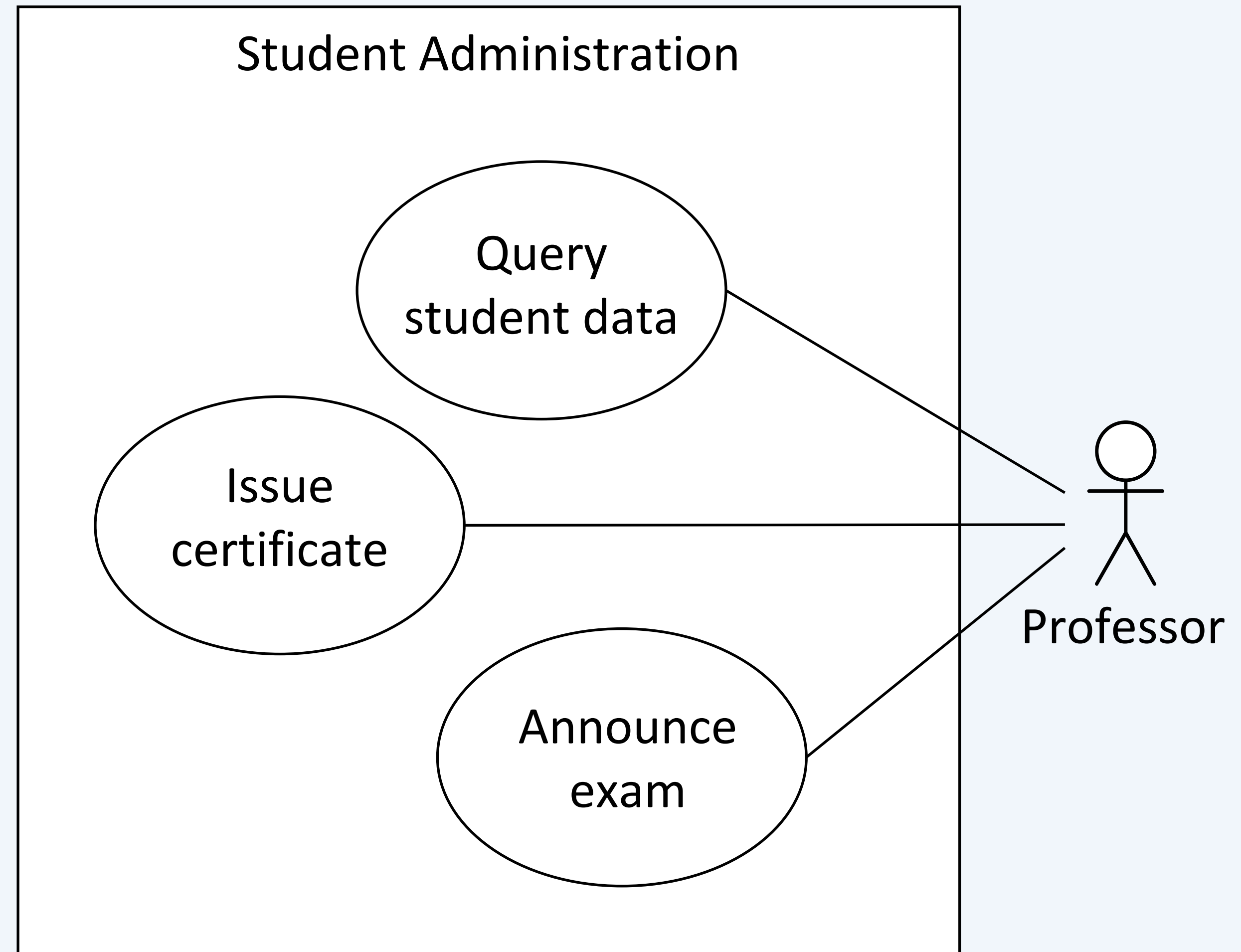
- Student Administration

- **Akteur\*in** (Wer benutzt das System?)

- Professor

- **Anwendungsfälle des Professor**  
(Was machen die Akteur\*innen?)

- Abfragen von Studierendendaten
  - Ausstellung von Zeugnissen
  - Prüfungsankündigungen



# Anwendungsfalldiagramm

## Beispiel Kaugummiautomat





DOLLY

FRUITS

10 Cent

FRUIT SHAPED BUBBLE GUM FILLED WITH  
SUGAR FREE SHERBET POWDER!

© 1995 ZED CANDY CO. ALL RIGHTS RESERVED. MADE IN CANADA. PS45-144





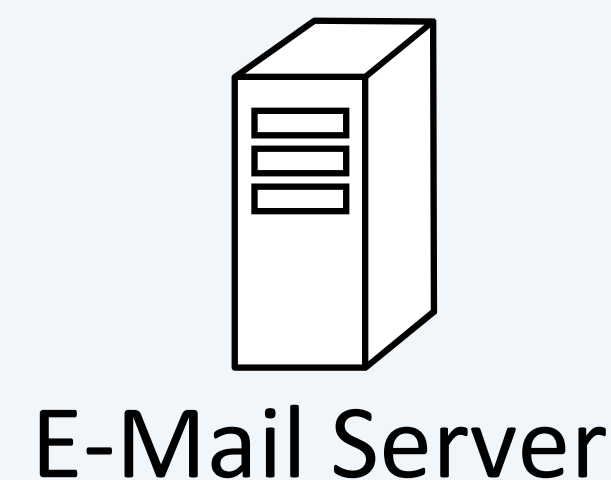
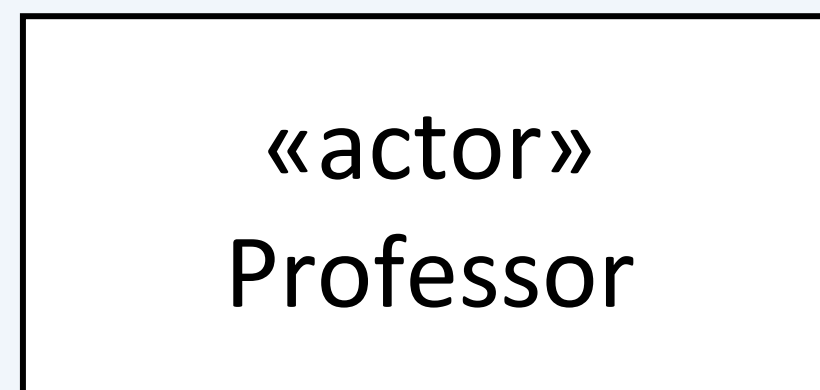
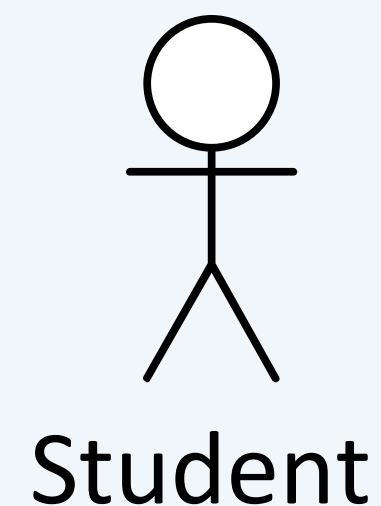
Anwendungsfalldiagramm

**Der Akteur bzw. die Akteurin und der Anwendungsfall**





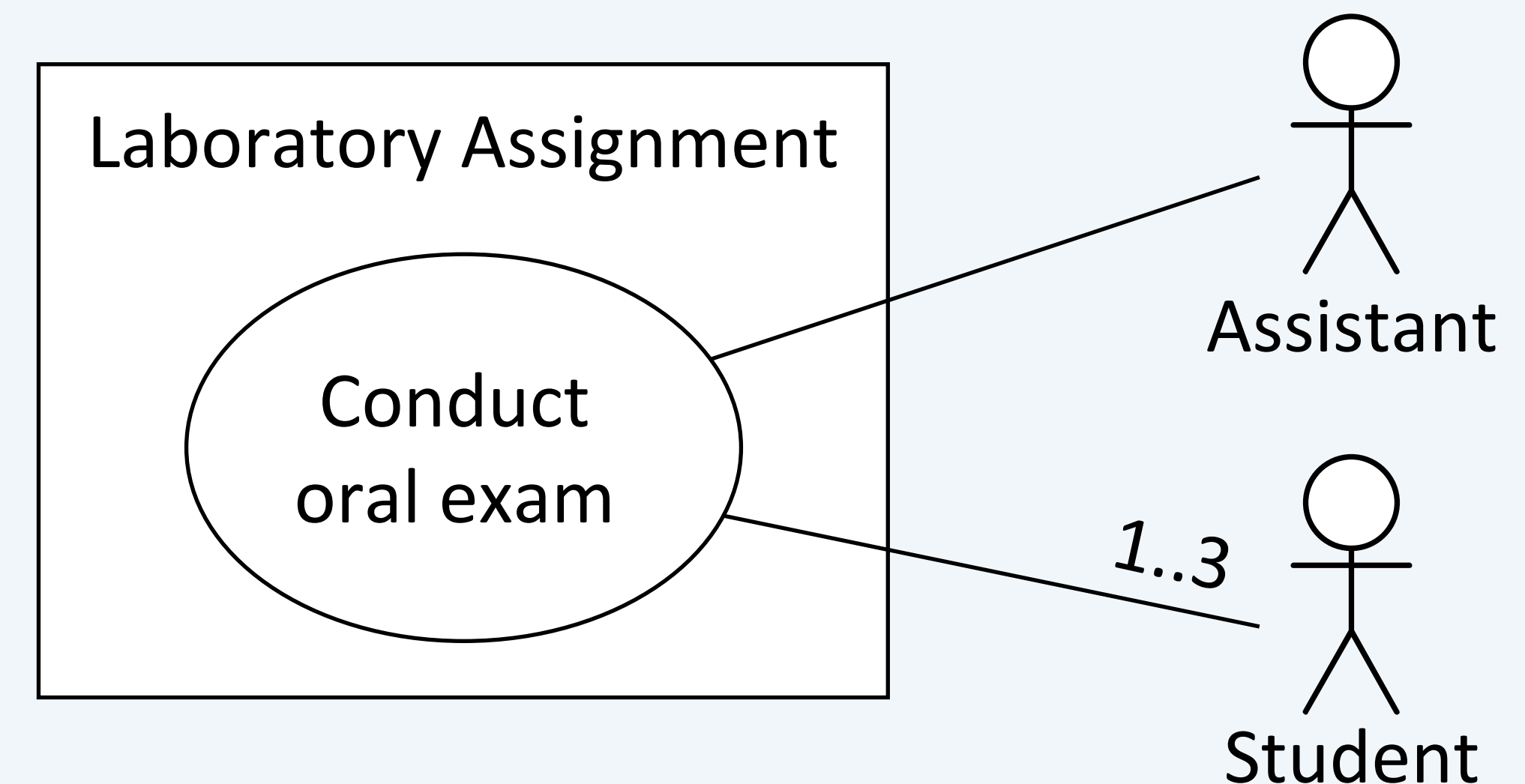
- Akteur\*innen **interagieren** mit dem System...
  - indem sie das **System benutzen**,
  - indem sie **vom System benutzt werden**
- Akteur\*in wird **durch Assoziationen mit Anwendungsfällen verbunden**, d.h. er/sie „kommuniziert“ mit dem System
- Jede Akteurin bzw. jeder Akteur muss mit **mindestens einem Anwendungsfall** kommunizieren
- Die Assoziation ist binär und kann **Multiplizitäten** aufweisen
- Notationsvarianten:





# Akteur\*in - Eigenschaften

- Akteur\*innen repräsentieren **Rollen der Benutzer**
  - Konkrete Benutzer können gleichzeitig mehrere Rollen spielen, annehmen und ablegen
- Akteur\*innen befinden sich **klar außerhalb** der Systemgrenzen
- Üblicherweise werden Benutzerdaten auch innerhalb des Systems verwaltet. Diese werden als Objekte bzw. Klassen innerhalb des Systems modelliert.
- Beispiel: **Laboratory Assignment**
  - Akteur\*in **Assistant** interagiert mit dem System **Laboratory Assignment** indem er bzw. sie es benutzt
  - Die Klasse **Assistant** umfasst Objekte, welche die Userdaten beinhalten (Name, SozVersNr, ...)





# Akteur\*in - Klassifikation

- **Menschlich**

- z.B. Student, Professor

- **Nicht-menschlich**

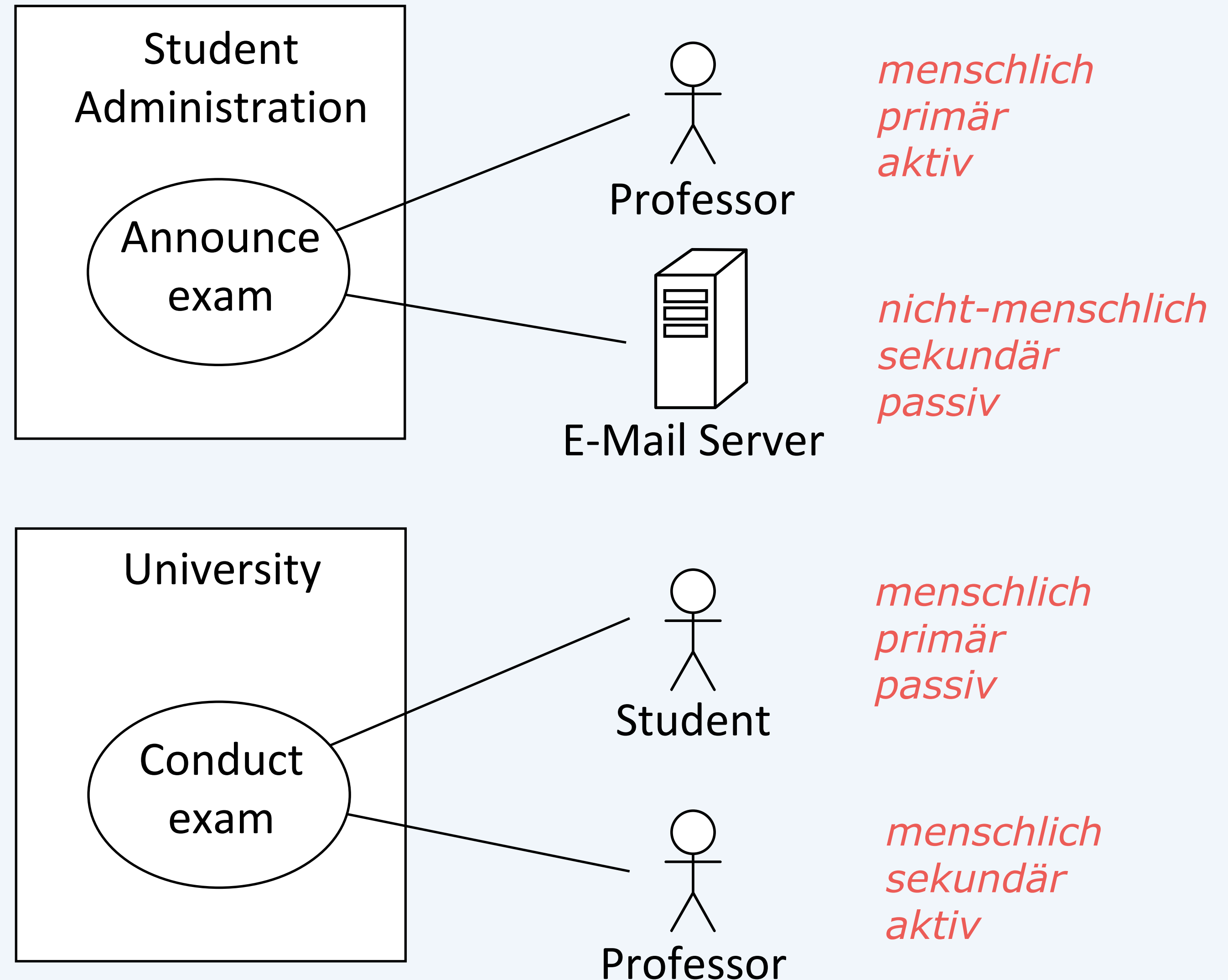
- z.B. E-Mail Server

- **Primär:** Hauptnutznießer\*in der Anwendung

- **Sekundär:** notwendig für das Funktionieren des Systems

- **Aktiv:** stößt selbst Anwendungsfälle an

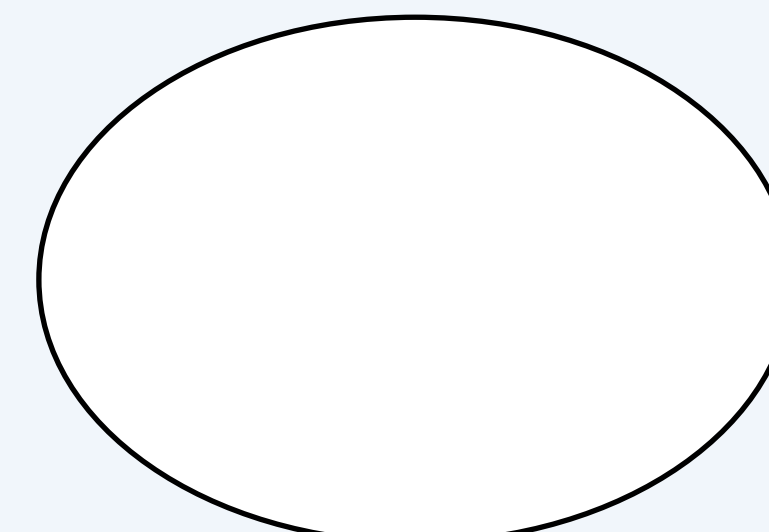
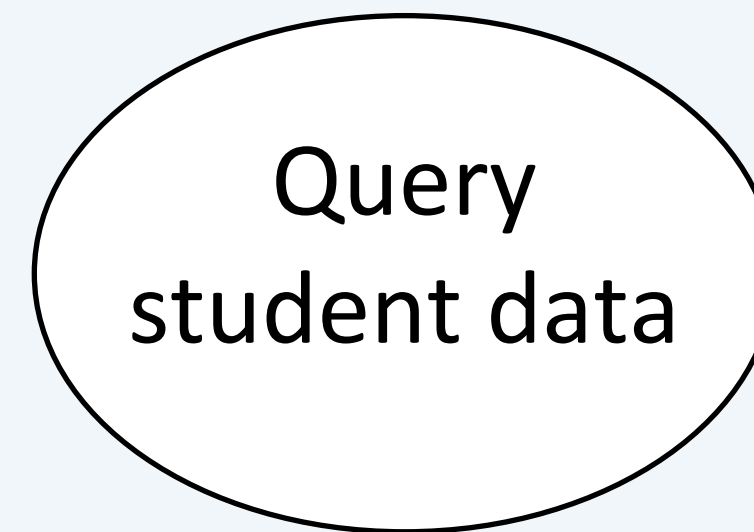
- **Passiv:** stößt selbst keine Anwendungsfälle an



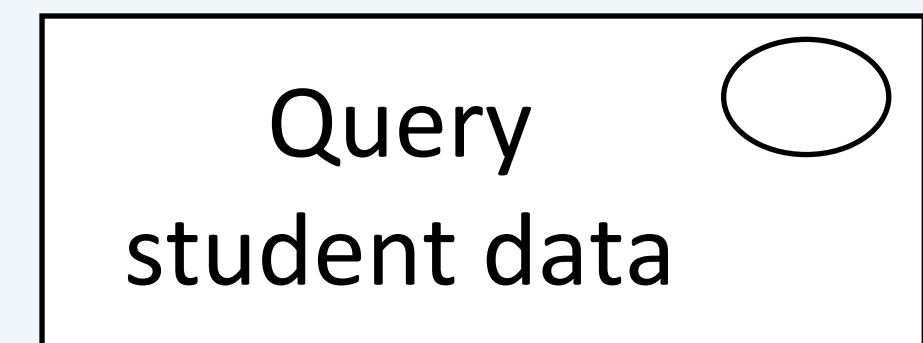


- Anwendungsfälle (use cases) beschreiben das **Verhalten**, das von dem zu entwickelnden System **erwartet** wird

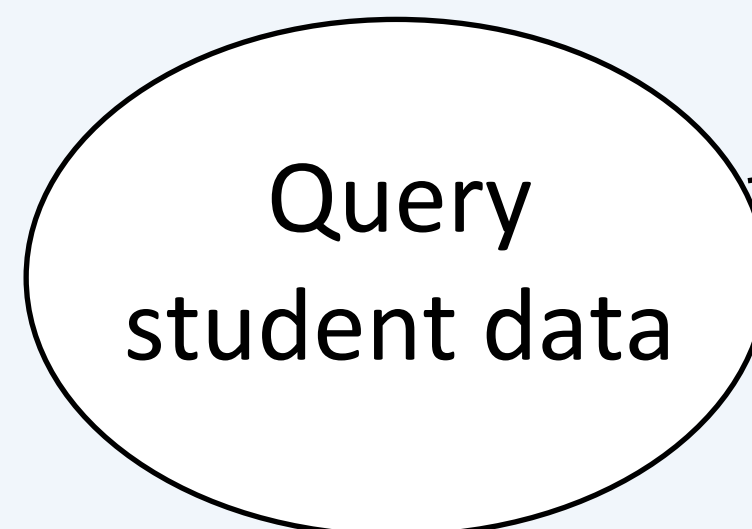
- Notationsvarianten



Query  
student data



- Kurzbeschreibung als Notiz



Berechtigte Benutzer\*innen  
können Studierendendaten  
abfragen

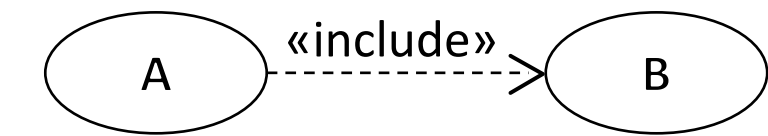


# Anwendungsfalldiagramm

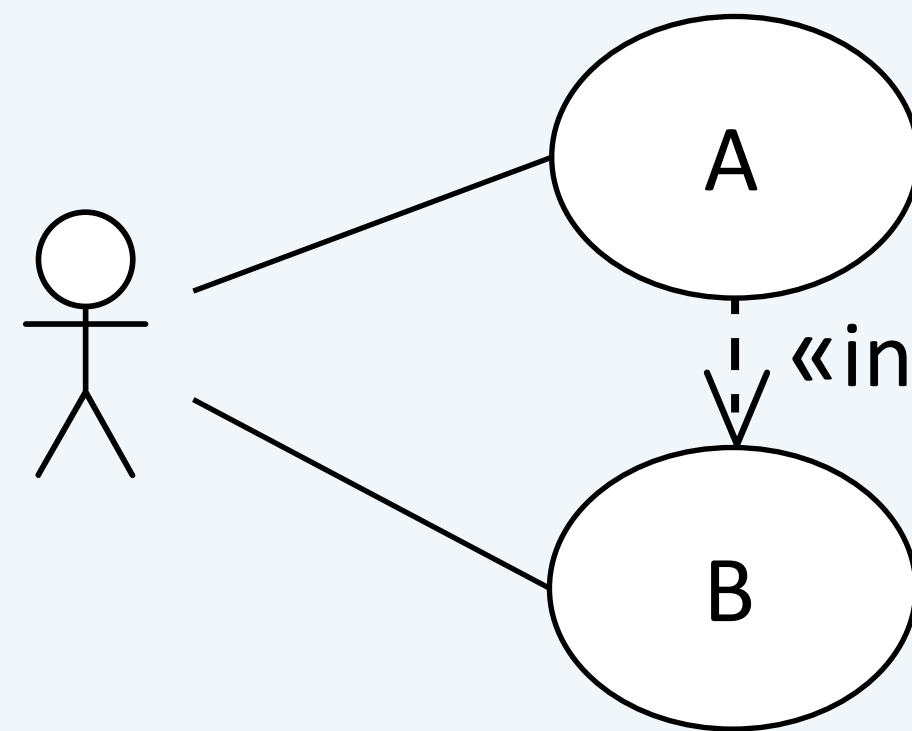
## Beziehungen zwischen Anwendungsfällen



# «include» - Beziehung



- Das **Verhalten** des benutzten Anwendungsfalls (inkludierter Anwendungsfall) wird in den benutzenden Anwendungsfall (Basis-Anwendungsfall) **eingebunden**



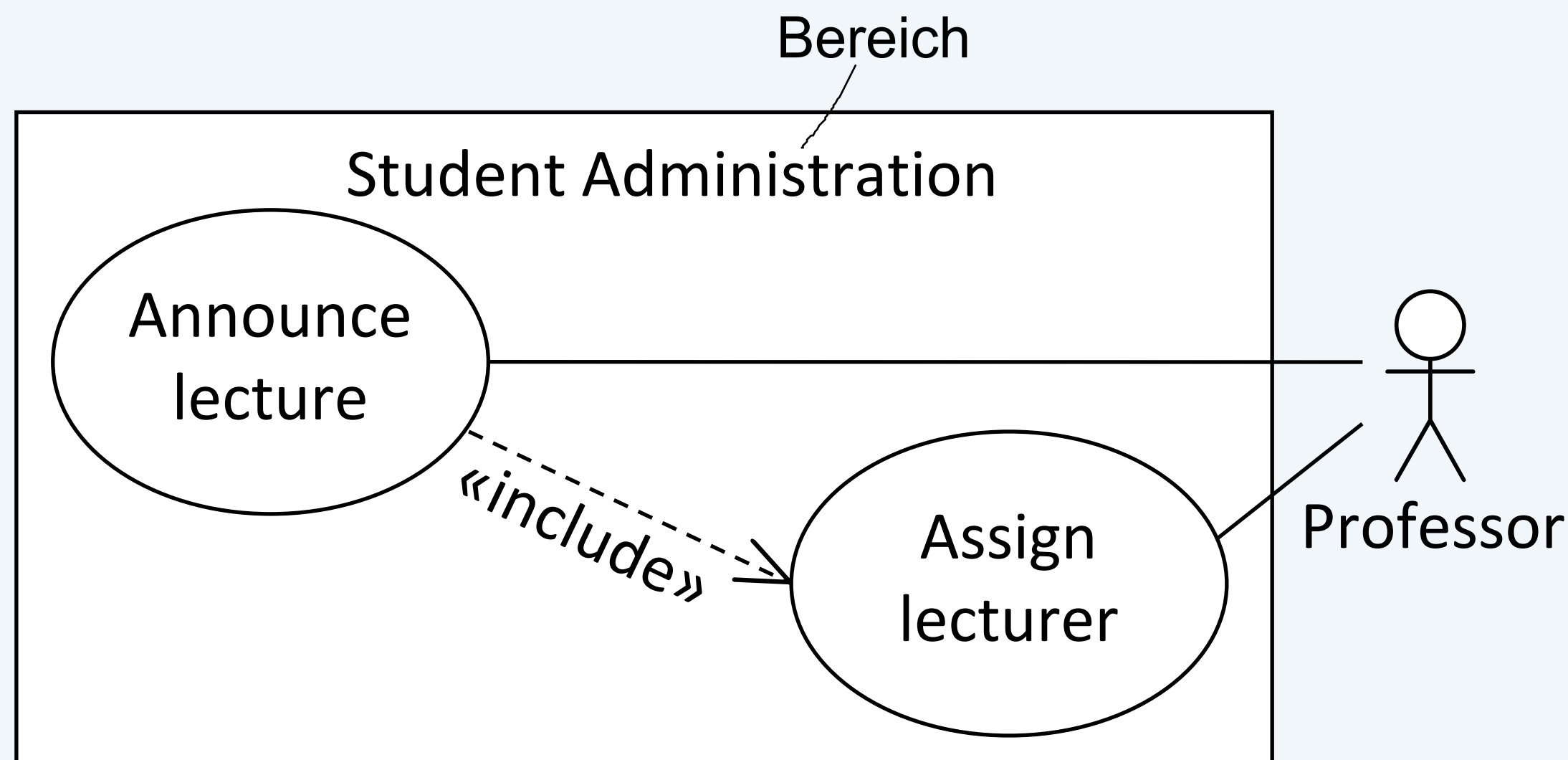
»base use case« *Basis-Anwendungsfall*

benötigt **B** um die Funktionalität sicher zu stellen

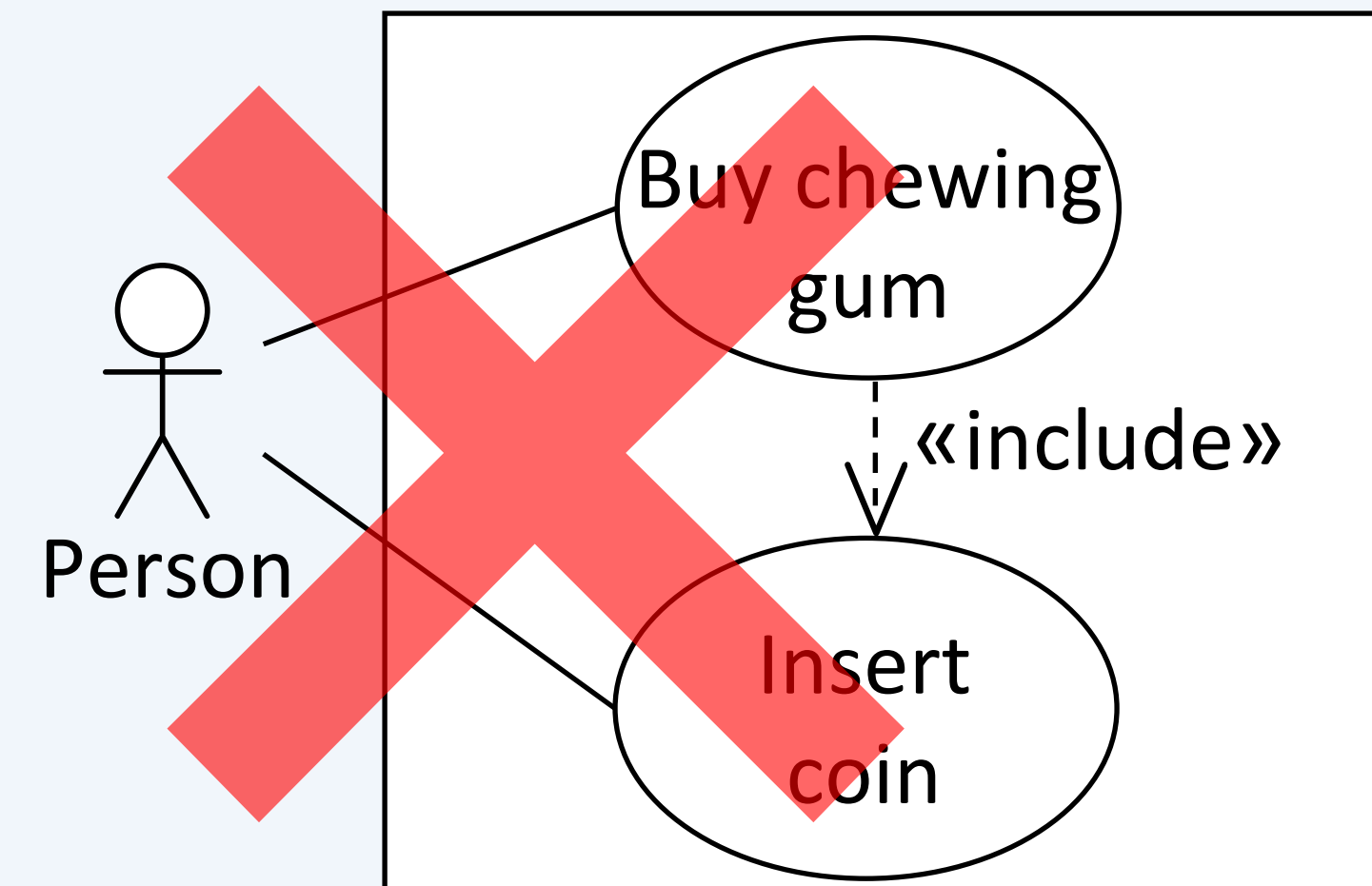
»included use case« *Inkludierter Anwendungsfall*

kann auch separat ausgeführt werden

- Bsp:

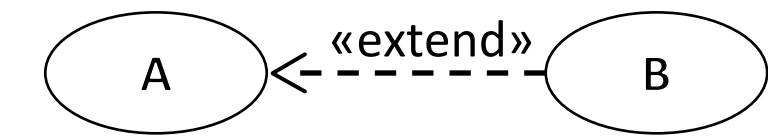


Weil "insert coin" kein Nutzen sondern bestenfalls Verlust  
(=Kein eigenständiger Use Case)



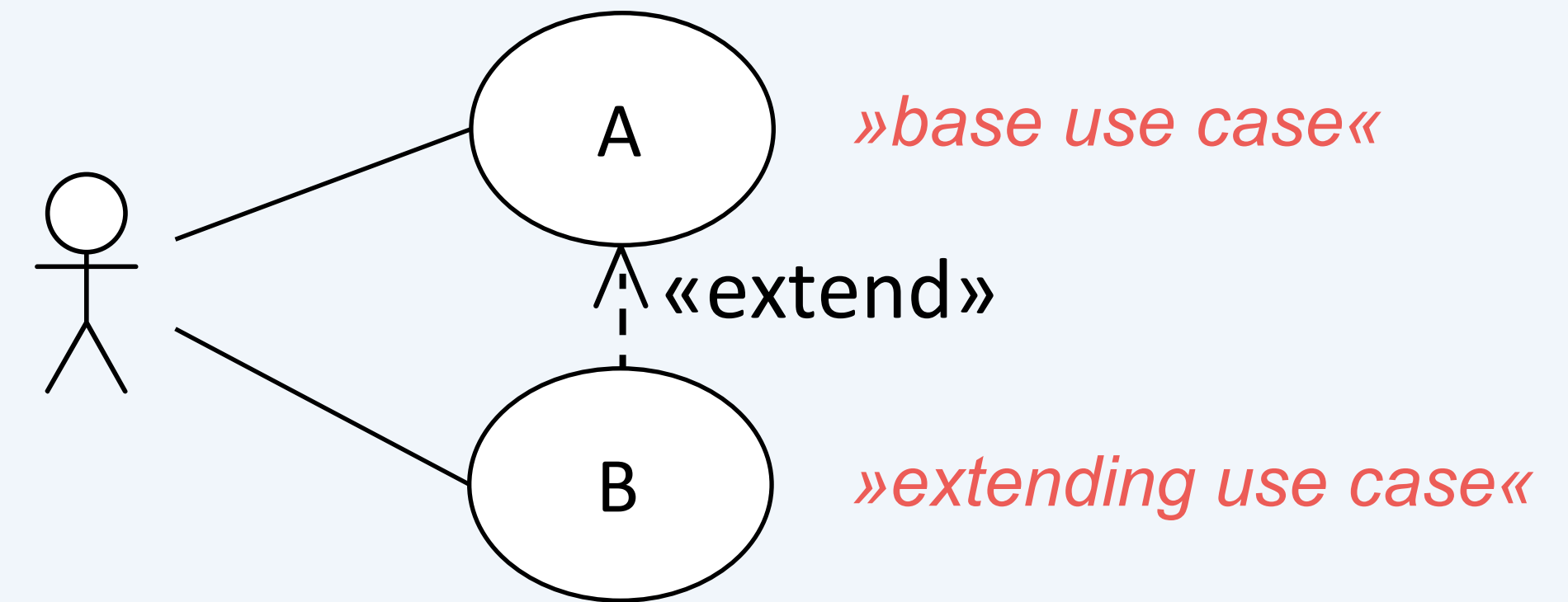


# «extend» - Beziehung

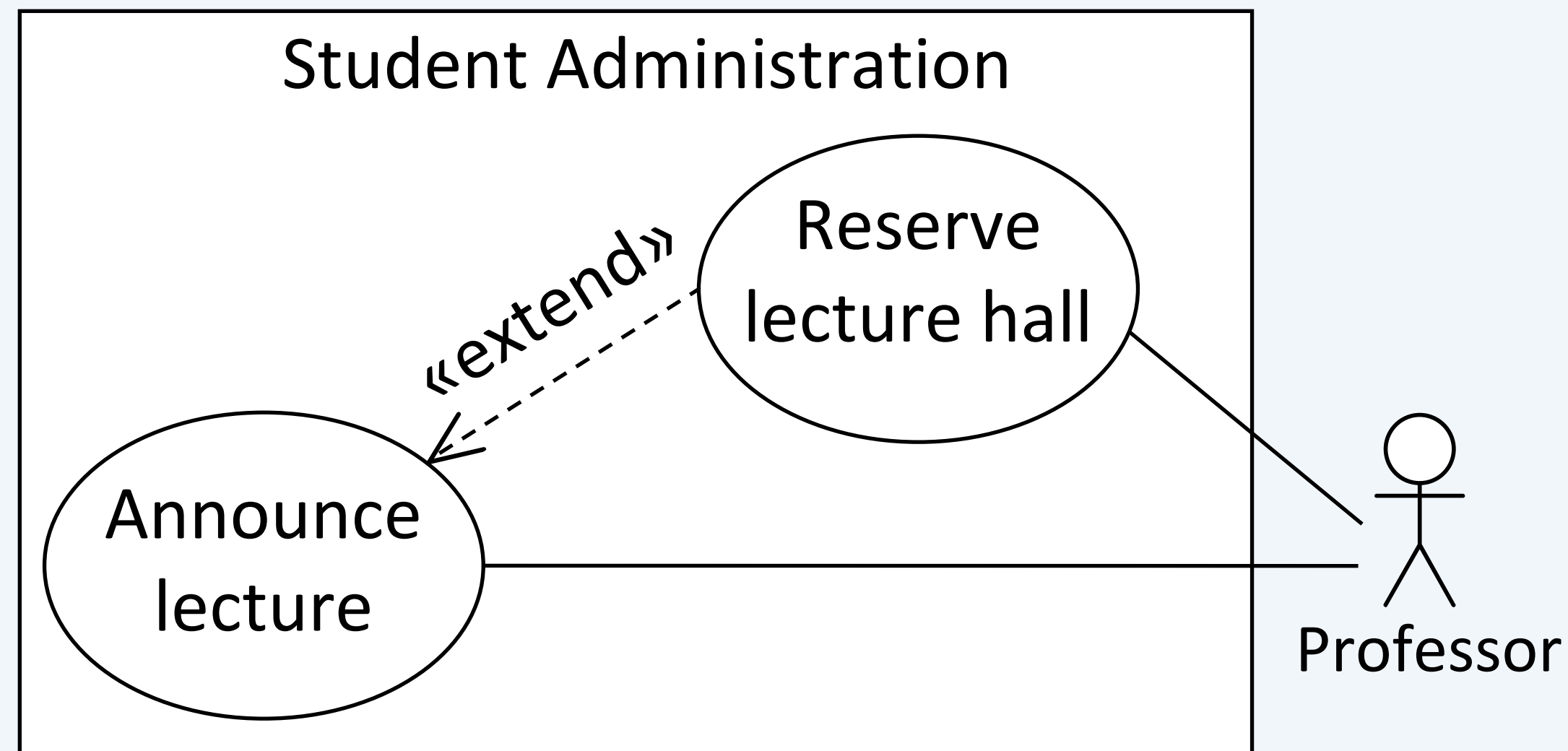


ingo

- Das Verhalten von **B** kann in **A** inkludiert werden
  - **A** entscheidet, ob **B** ausgeführt wird
- **A** bzw. **B** können auch separat ausgeführt werden
- Angabe des »Wo« durch Erweiterungsstellen in **A**
- Angabe des »Wann« durch Bedingung in **A** bzw. als Teil der «**extend**»-Beziehung



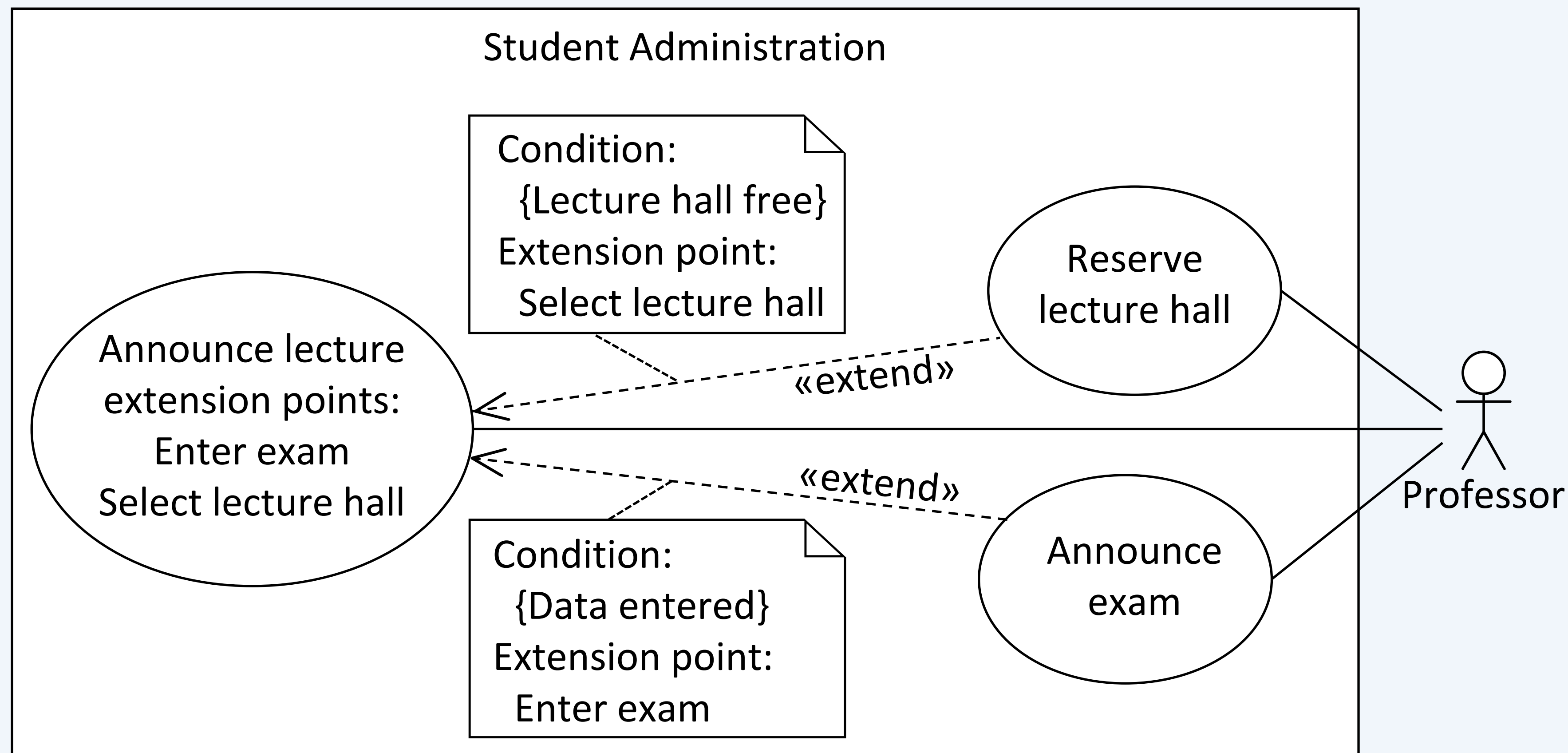
## ■ Bsp.:



# «extend» - Beziehung: Erweiterungsstellen

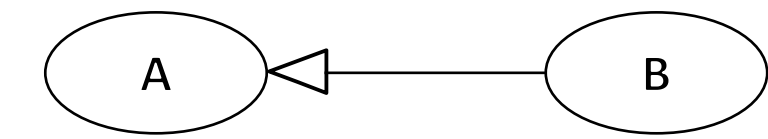
- Mehrere Erweiterungsstellen (extension points) je Anwendungsfall möglich
- Namen von Erweiterungsstellen
  - müssen eindeutig sein

■ Bsp.:



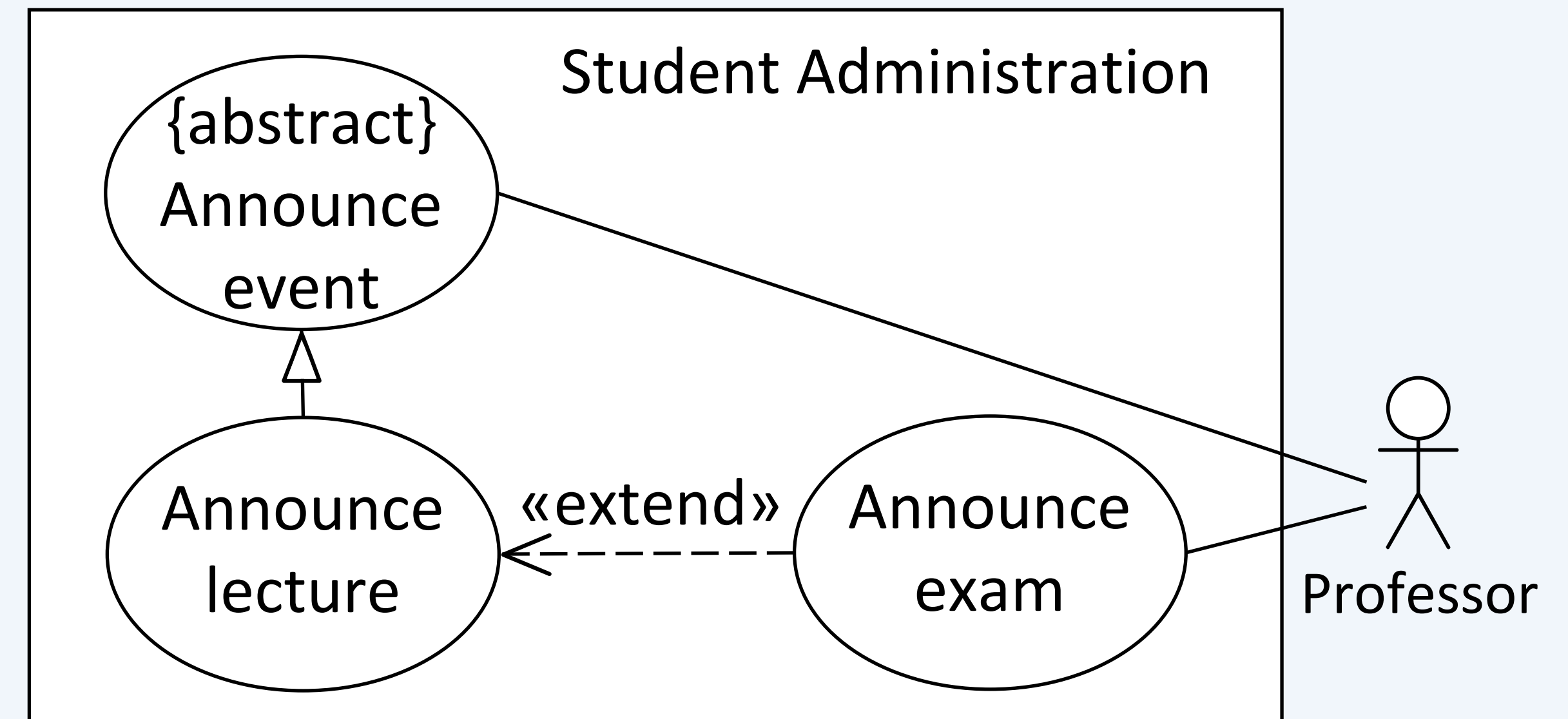
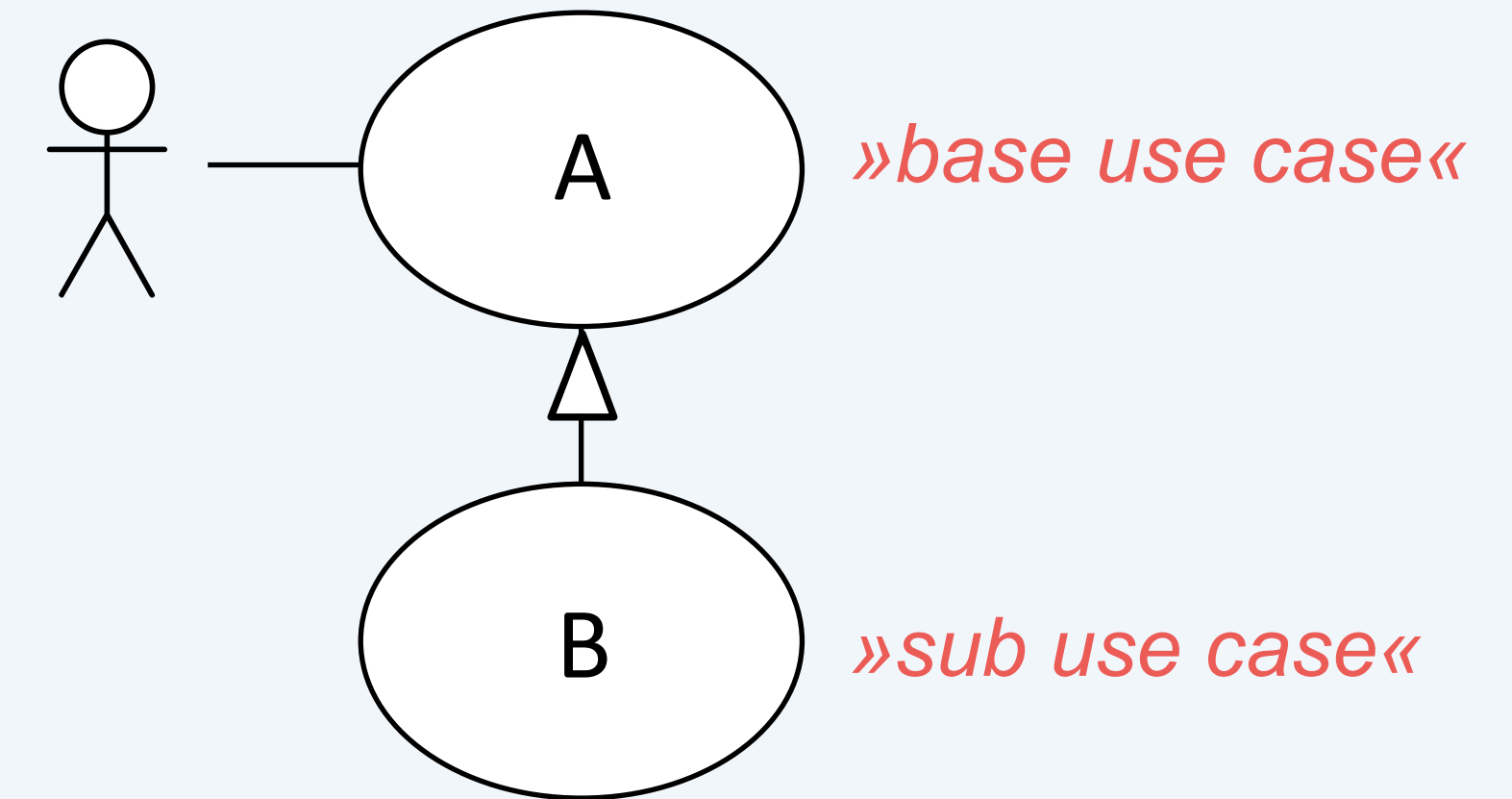


# Generalisierung bei Anwendungsfällen



ingo

- B erbt das Verhalten von A und kann dieses überschreiben oder ergänzen
- B erbt alle Beziehungen von A
- B übernimmt Grundfunktionalität von A
- B entscheidet, was von A ausgeführt bzw. geändert wird
- Modellierung abstrakter Anwendungsfälle möglich: {abstract}
  - abstrakte Anwendungsfälle sind nicht ausführbar!



ingo

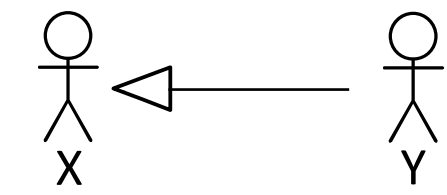
## Anwendungsfalldiagramm Beziehungen zwischen Akteuren



ingo

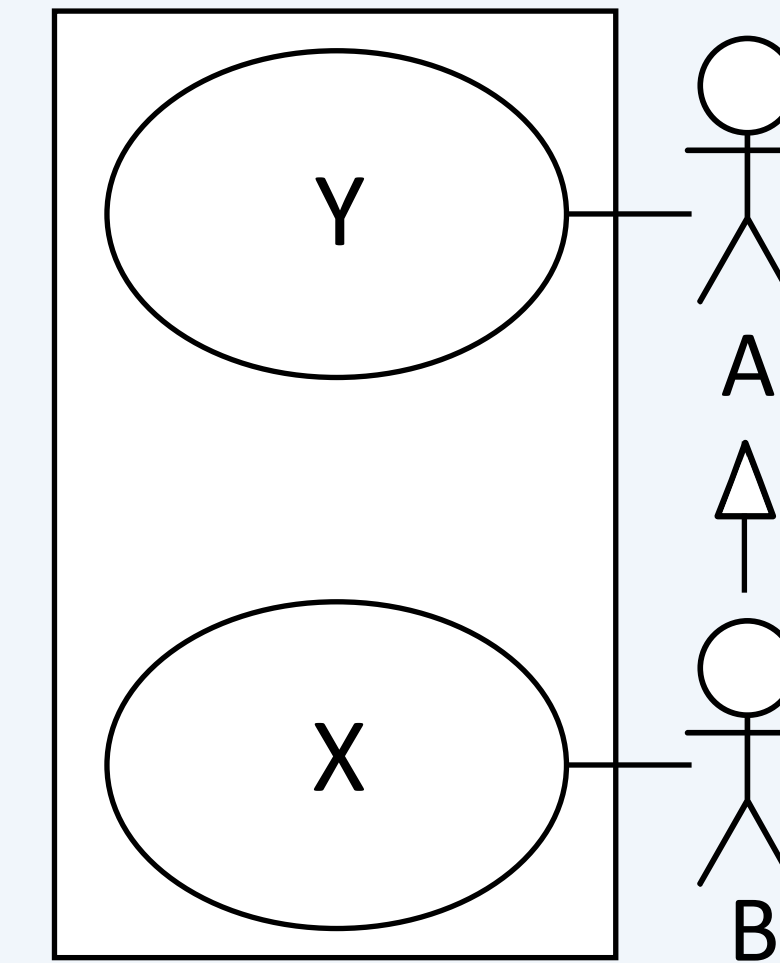


# Generalisierung bei Akteur\*innen (1/2)

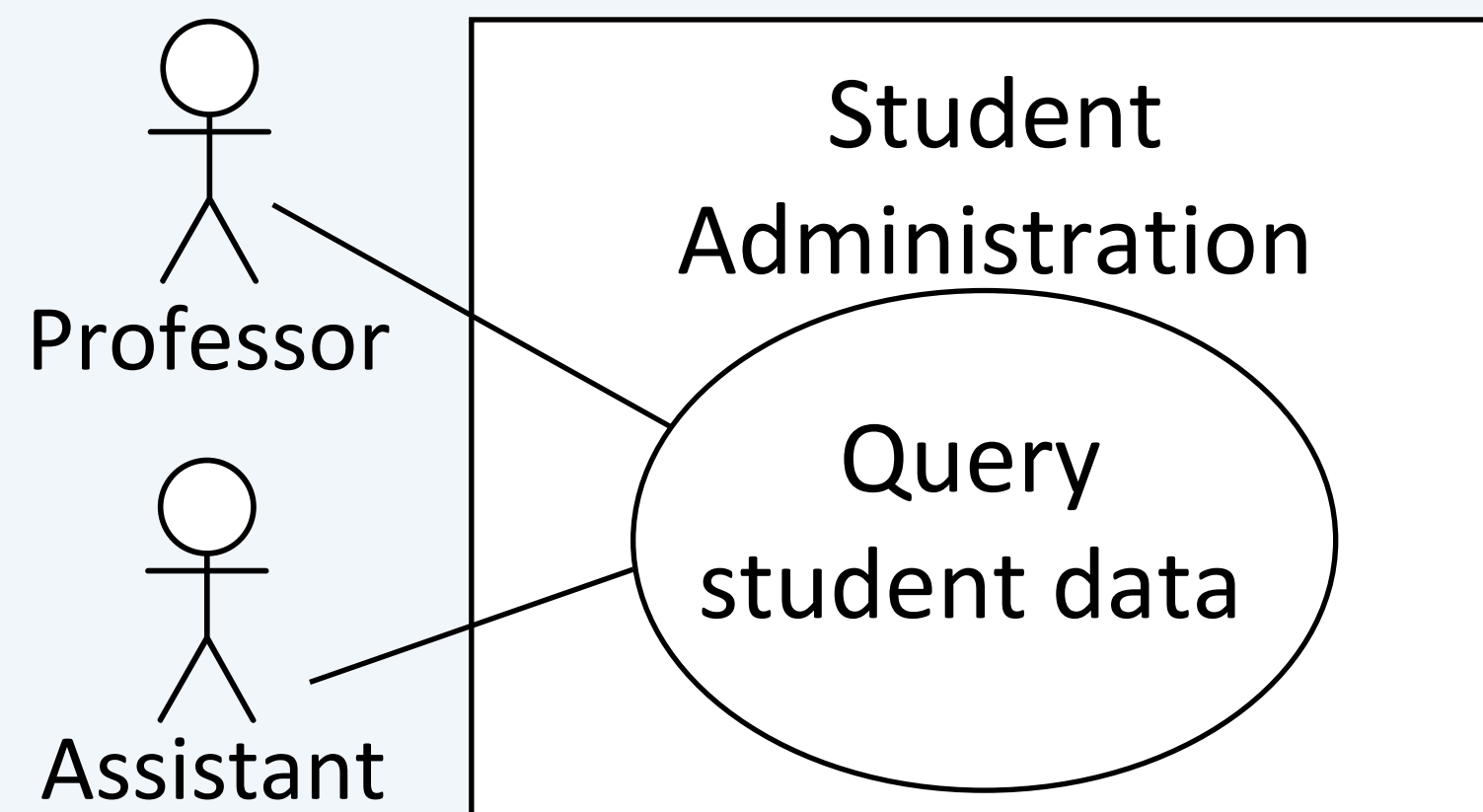


ingo

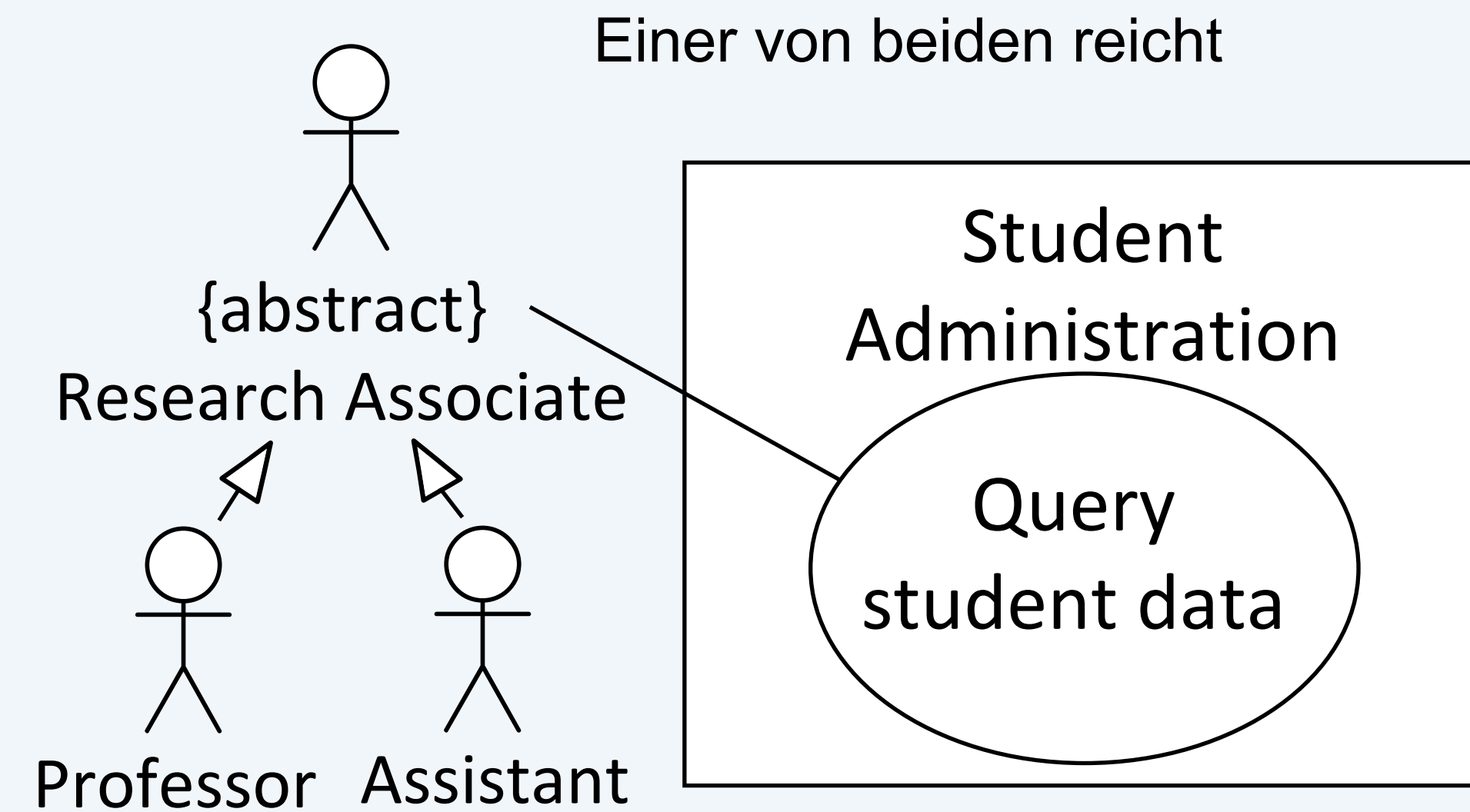
- Akteur\*in **B** erbt von Akteur\*in **A**
- **B** kann mit den Anwendungsfällen **X** und **Y** kommunizieren
- **A** kann nur mit **Y** kommunizieren
- Mehrfachvererbung ist erlaubt



- Bsp.: Ich brauche Professor UND Assistent um Use Case auszuführen

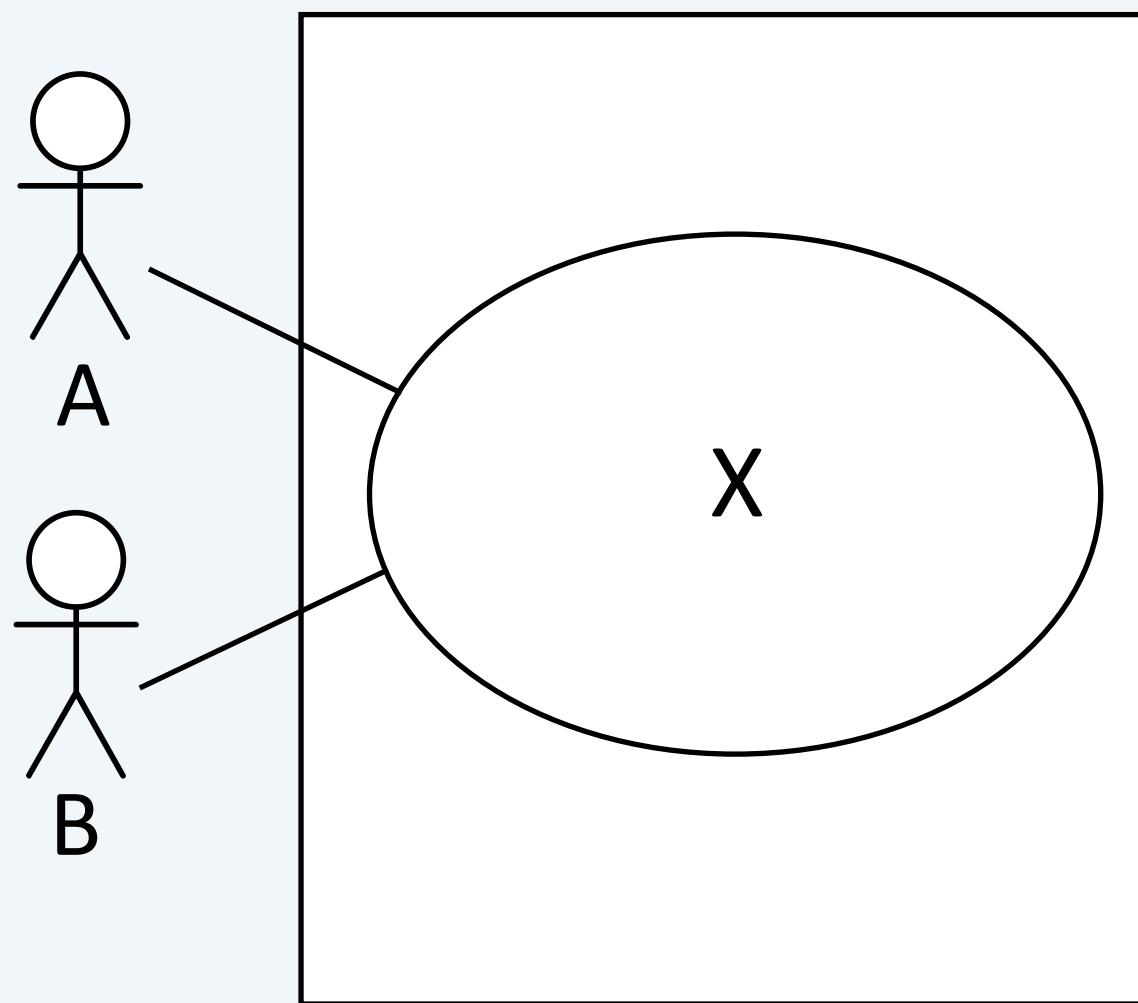


≠

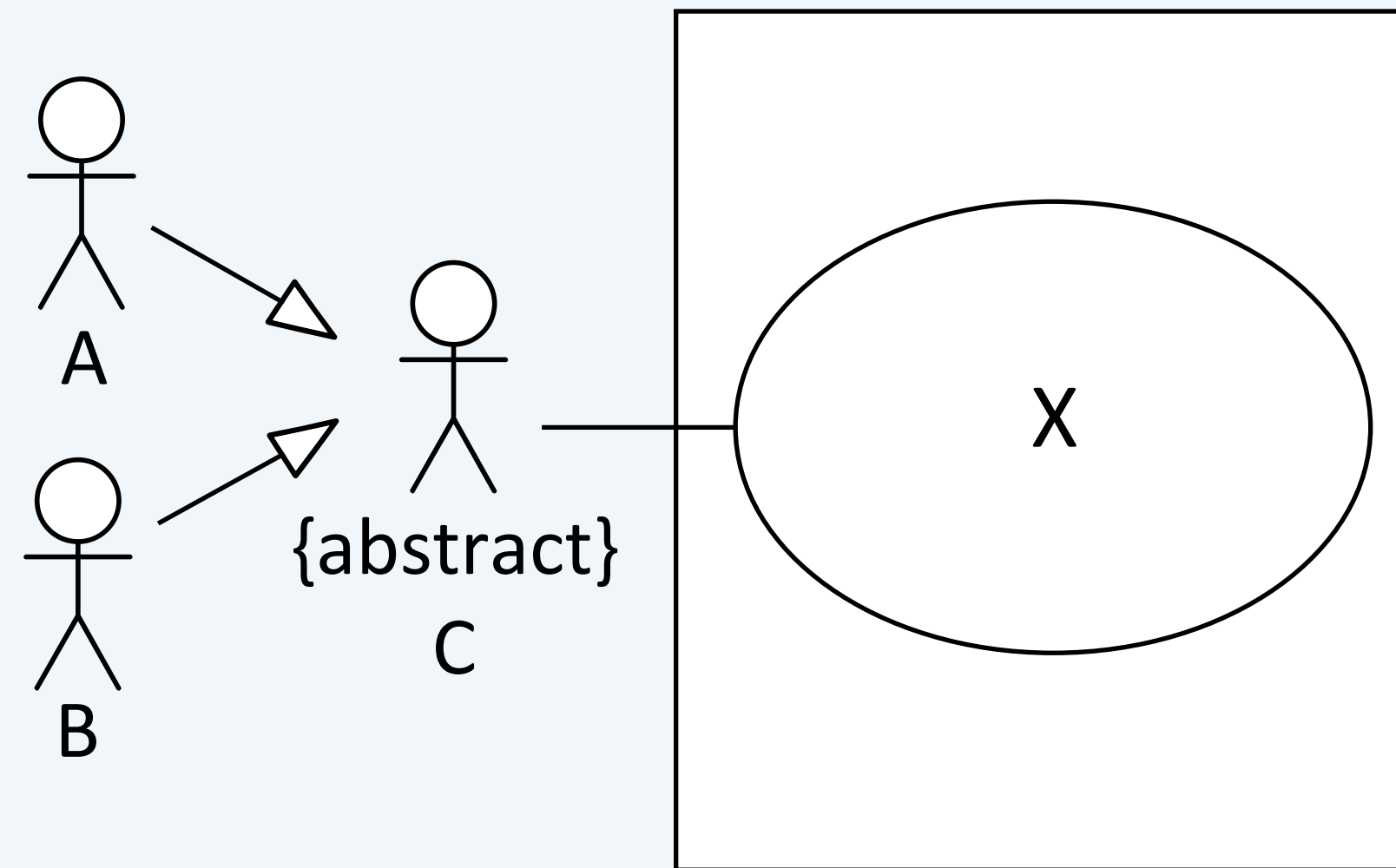


# Generalisierung bei Akteur\*innen (2/2)

- Unterscheidung, ob mehrere Akteur\*innen gemeinsam mit einem Anwendungsfall kommunizieren können oder müssen.



**A und B** kommunizieren  
mit **X**



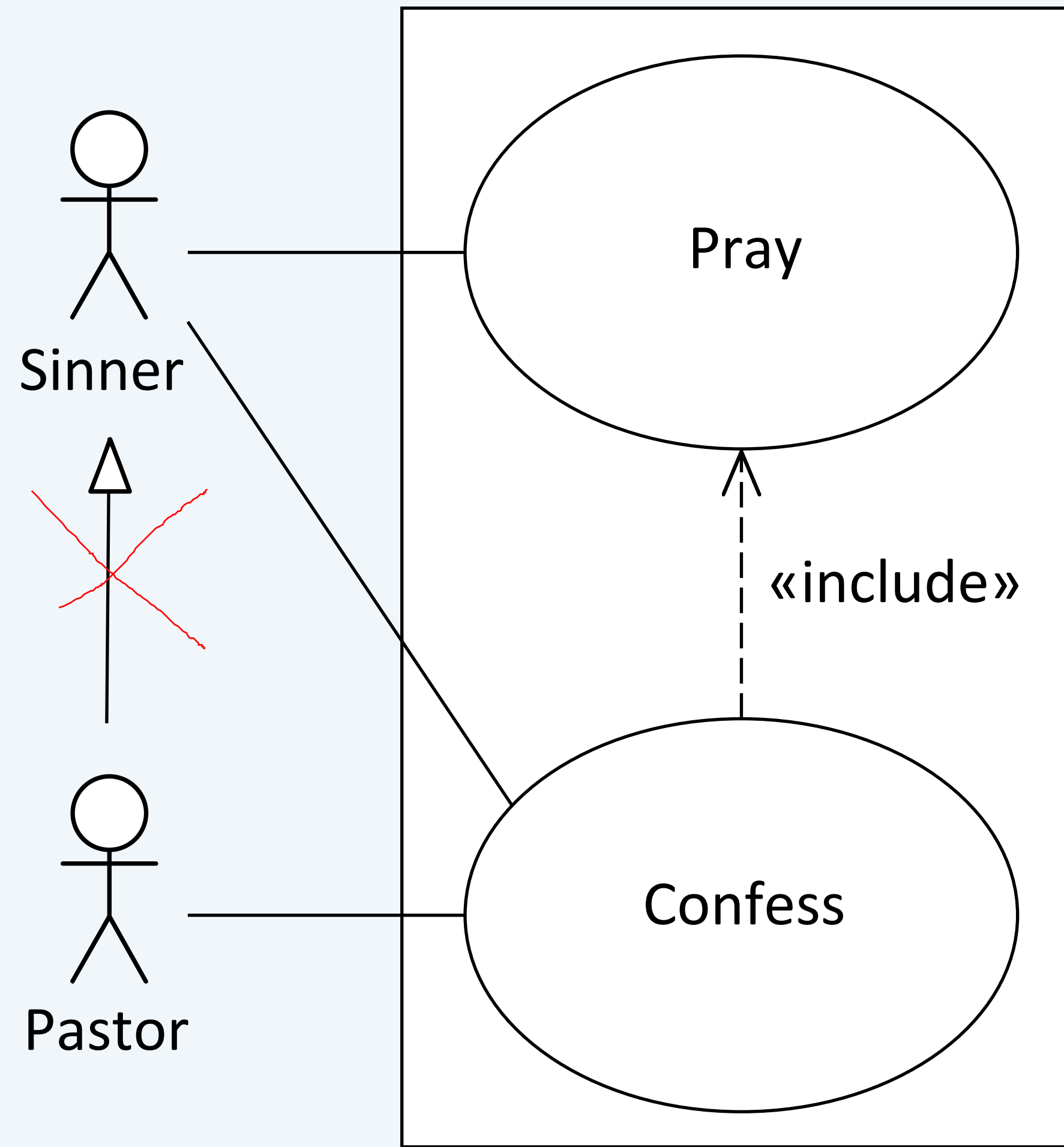
**A oder B** kommunizieren  
mit **X**



# Anwendungsfalldiagramm Beispiele

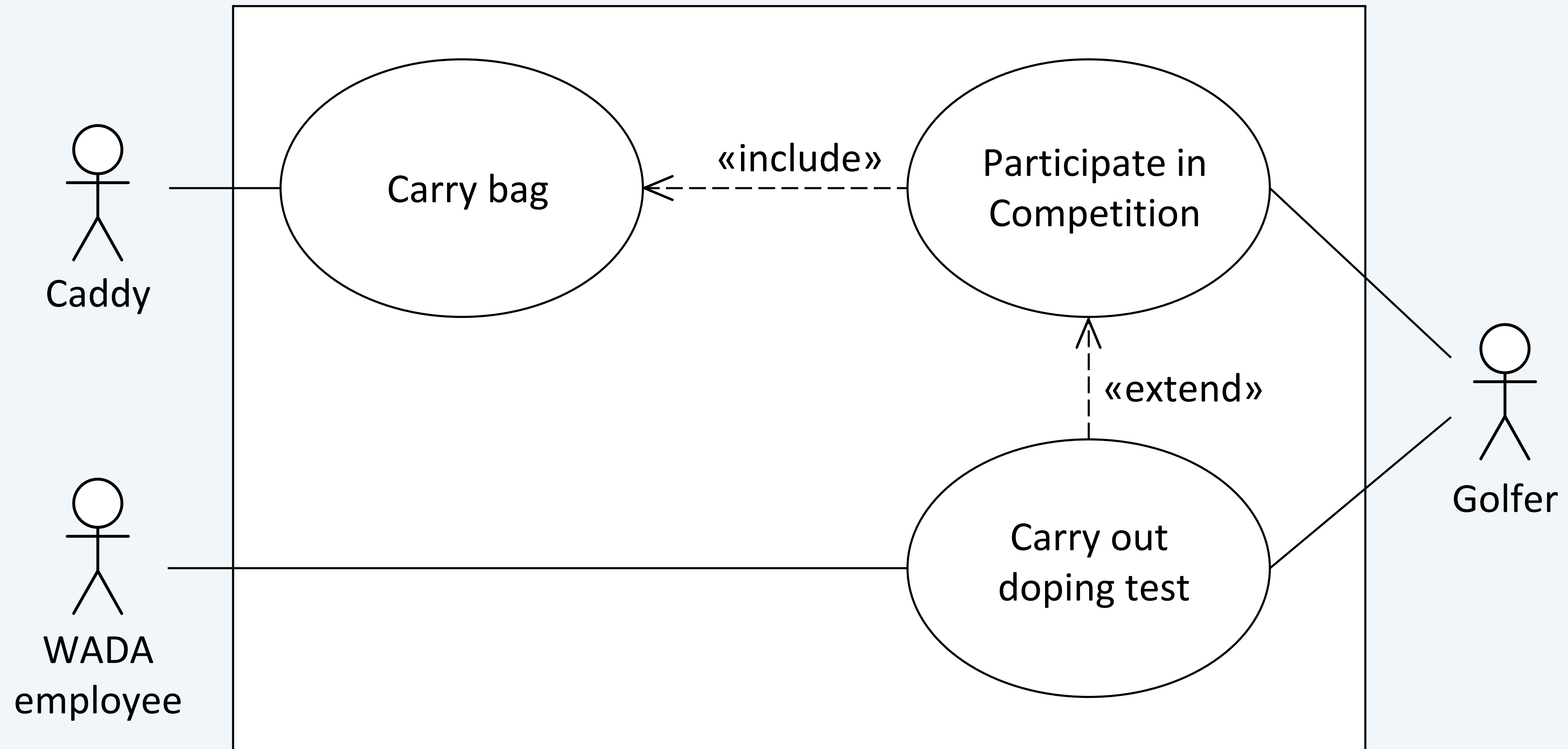


# Bsp.: Beichten

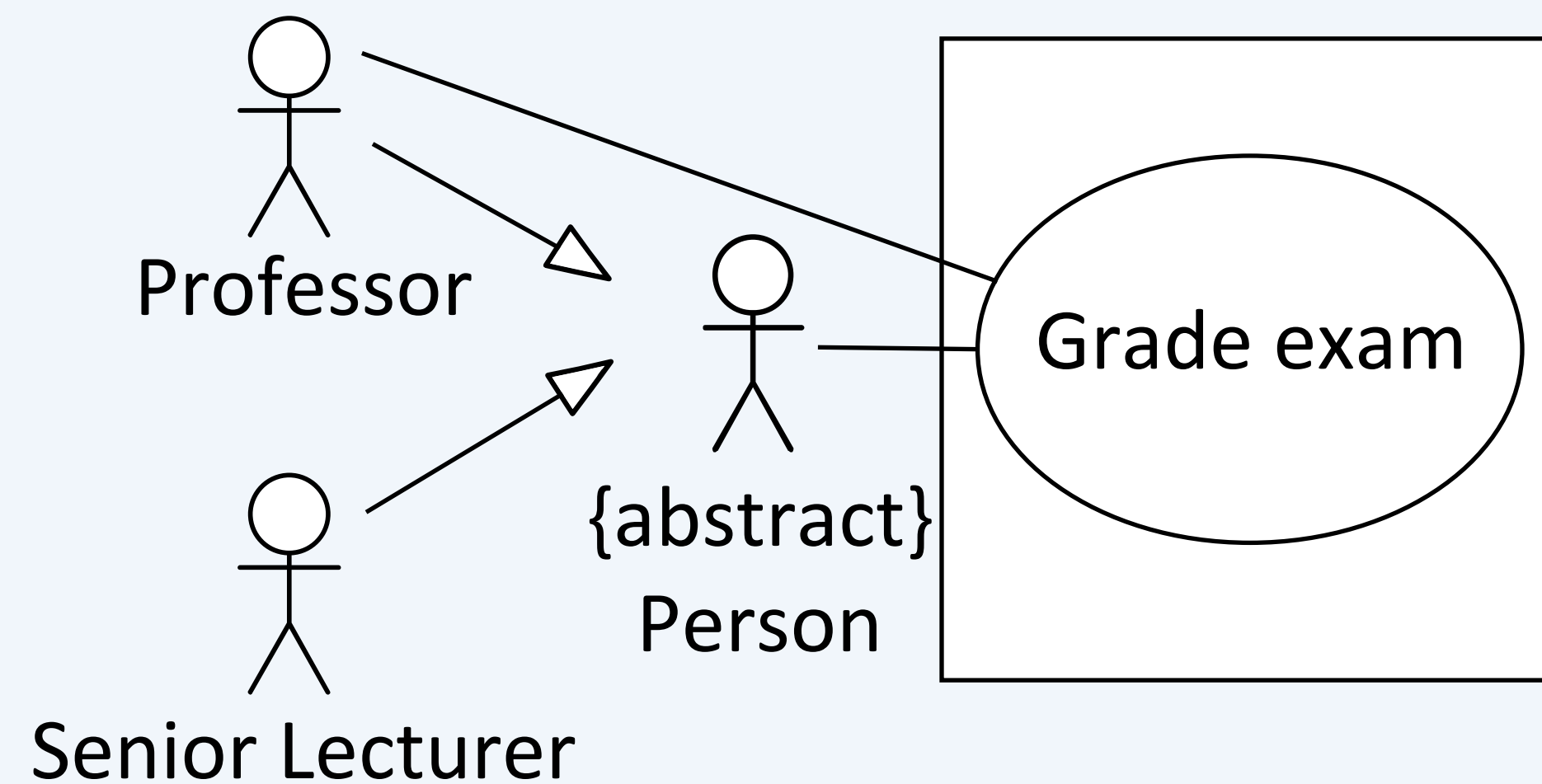
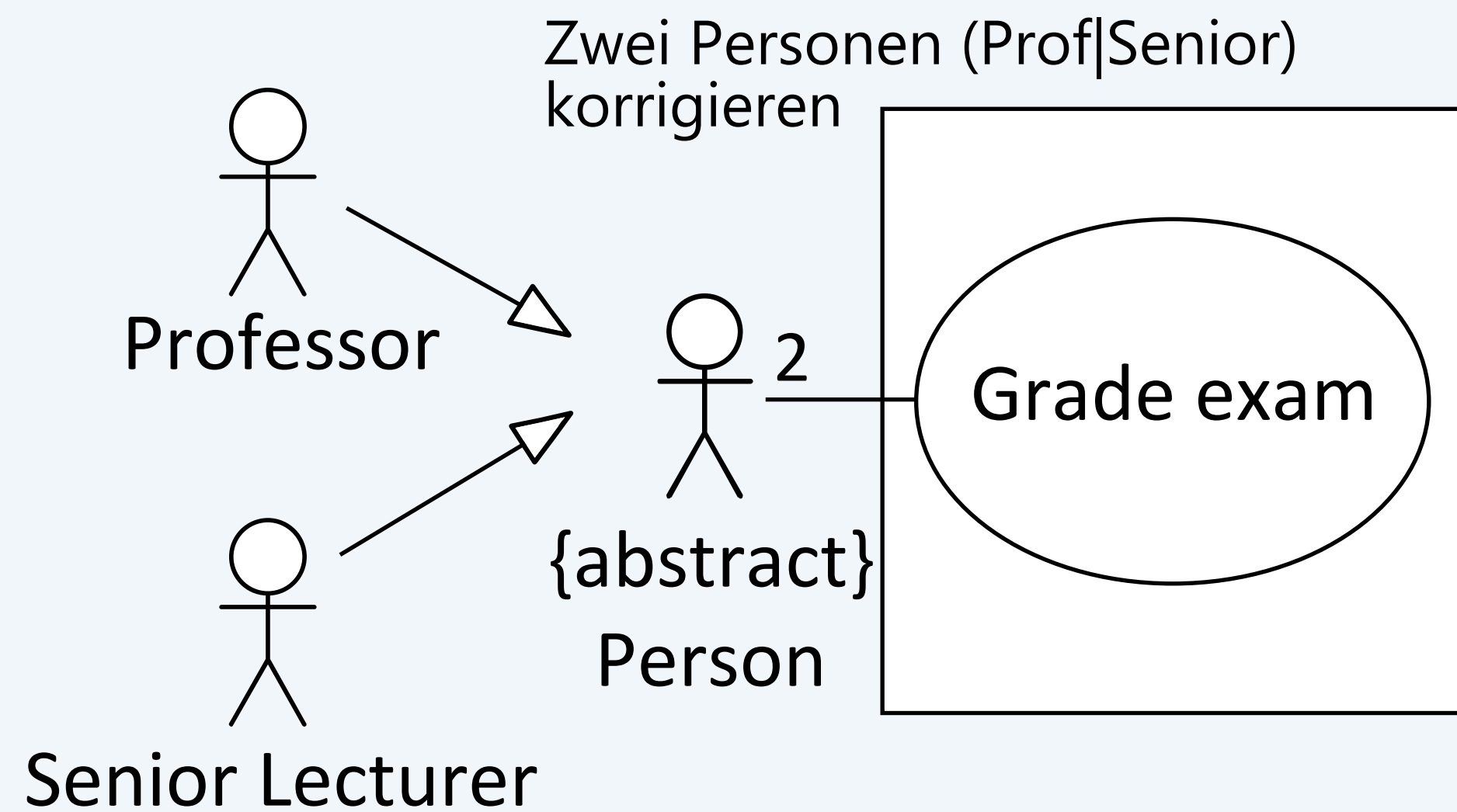
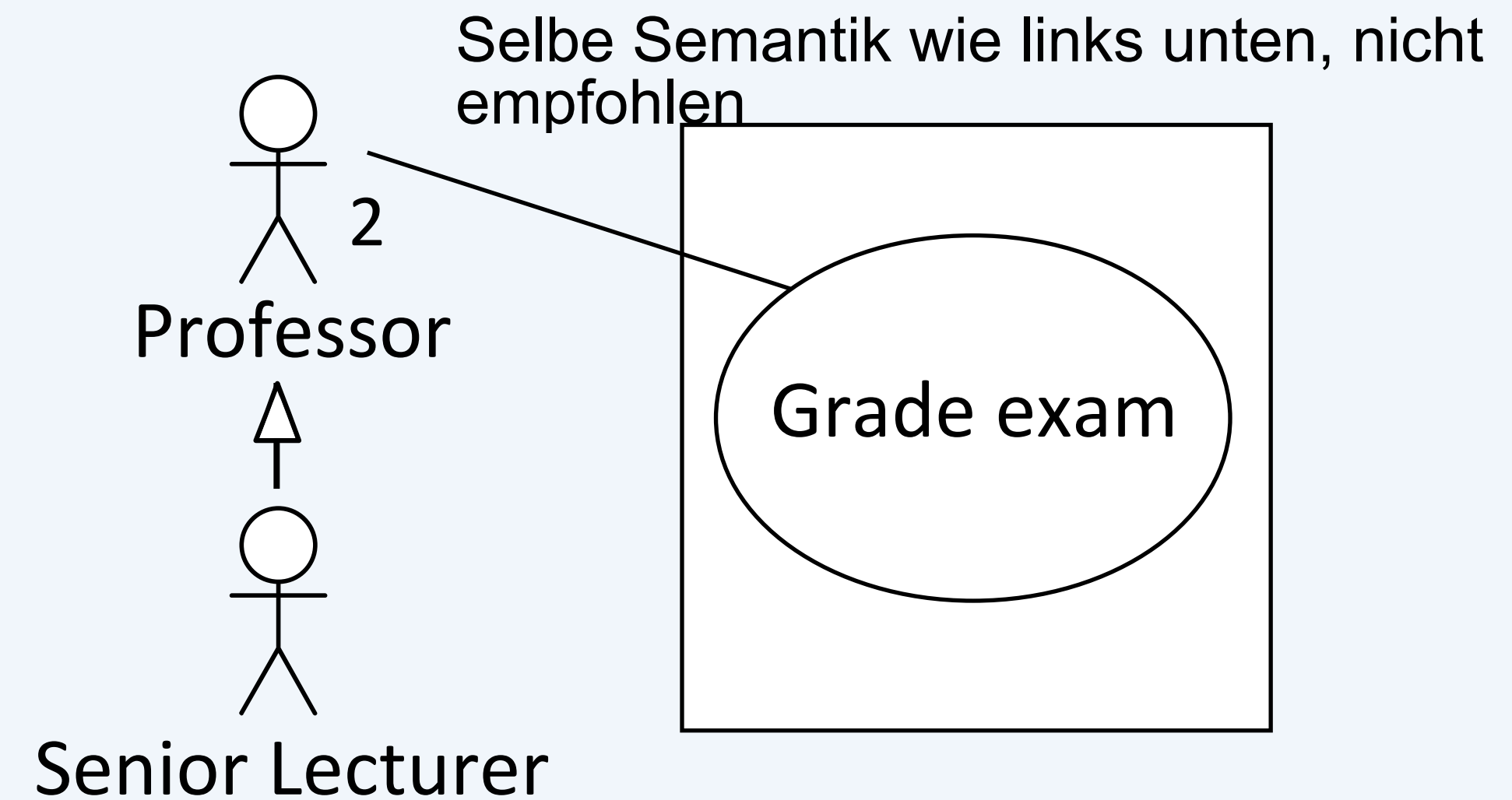
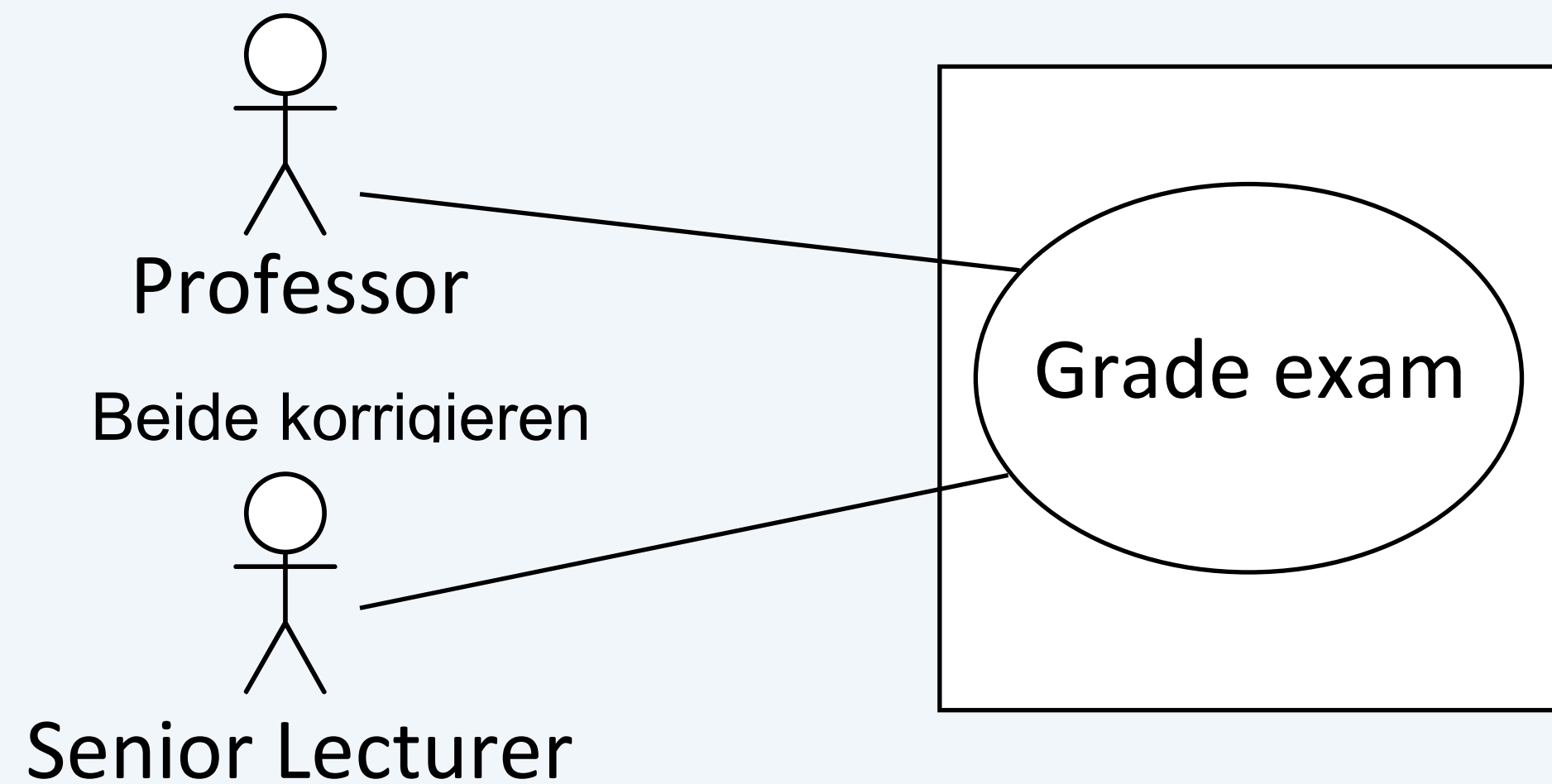




# Bsp.: An Wettbewerb teilnehmen



# Bsp.: Verschiedene Arten der Prüfungskorrektur





# Anwendungsfalldiagramm

## Die Anwendungsfallmodellierung in der Praxis



# Identifikation von Akteur\*innen

- Wer **benutzt** die wesentlichen Anwendungsfälle?
- Wer braucht Systemunterstützung für die **tägliche Arbeit**?
- Wer ist für die **Systemadministration** zuständig?
- Mit welchen externen **Geräten / (Software-) Systemen** muss das System kommunizieren können?
- Wer oder was interessiert sich für die **Ergebnisse** des Systems?



# Identifikation von Anwendungsfällen

- Nach der Identifikation der Akteur\*innen
- Trigger für Anwendungsfälle suchen
  - Trigger = Ereignisse, die eintreten müssen, damit das System veranlasst wird ein Ergebnis zu produzieren.
  - Der Aufruf des Systems erfolgt oft durch Akteur\*in, der bzw. die damit Akteur\*in des Anwendungsfalles wird.
  - Es werden folgende Trigger unterschieden:
    - Intern
    - Extern
    - Zeitlich

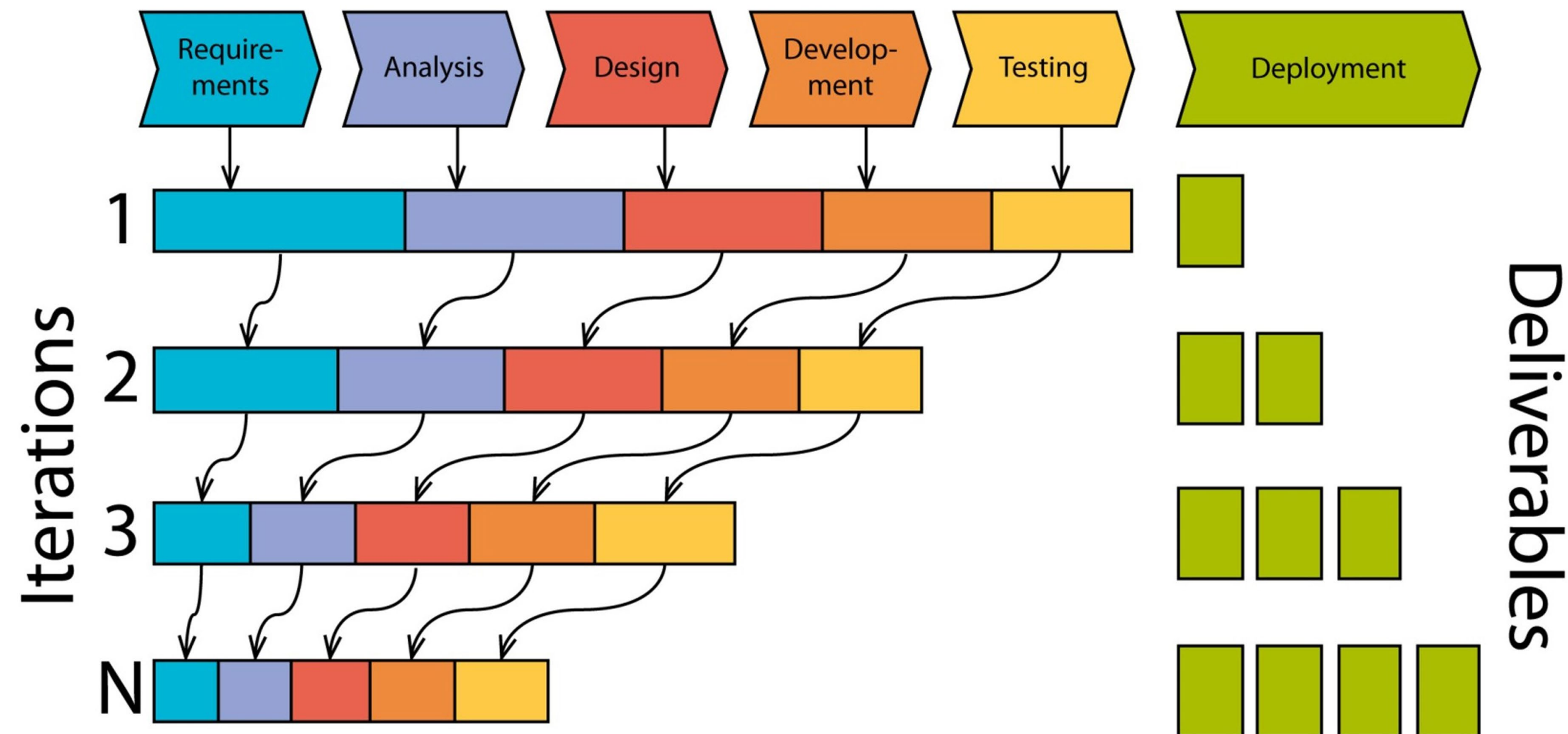
- Name
- Kurzbeschreibung
- Vorbedingung: Voraussetzung für erfolgreiche Ausführung
- Nachbedingung: Systemzustand nach erfolgreicher Ausführung
- Fehlersituationen: nur problembereichsrelevante Fehler
- Systemzustand im Fehlerfall
- Akteur\*innen, die mit dem Anwendungsfall kommunizieren
- Trigger: auslösende Ereignisse für den Anwendungsfall
- Standardablauf: einzelne Schritte
- Alternativabläufe: Abweichungen vom Standardablauf

# Regeln zur Anwendungsfallmodellierung

- Die wichtigsten funktionalen Anforderungen müssen in den Anwendungsfällen festgehalten werden.
- Ein Anwendungsfall....
  - beschreibt eine Transaktion für die Auftraggeber\*innen bezahlen.
  - beschreibt einen typischen Fall, ein System zu verwenden und nicht mehr.
  - ist wie ein Theaterstück. Die Anwendungsfallbeschreibung enthält die Choreographie.
  - hat eine Einleitung, einen Hauptteil und einen Schluss.
  - soll so einfach wie möglich beschrieben sein.
  - muss präzise definiert sein.
  - ist dann fertig beschrieben, wenn Kund\*innen, Anwender\*innen und Softwareentwickler\*innen ihn akzeptieren.
  - stellt die Grundlage für einen Systemtest dar.
  - sollte mit maximal zwei Seiten beschrieben werden.



# Use Cases



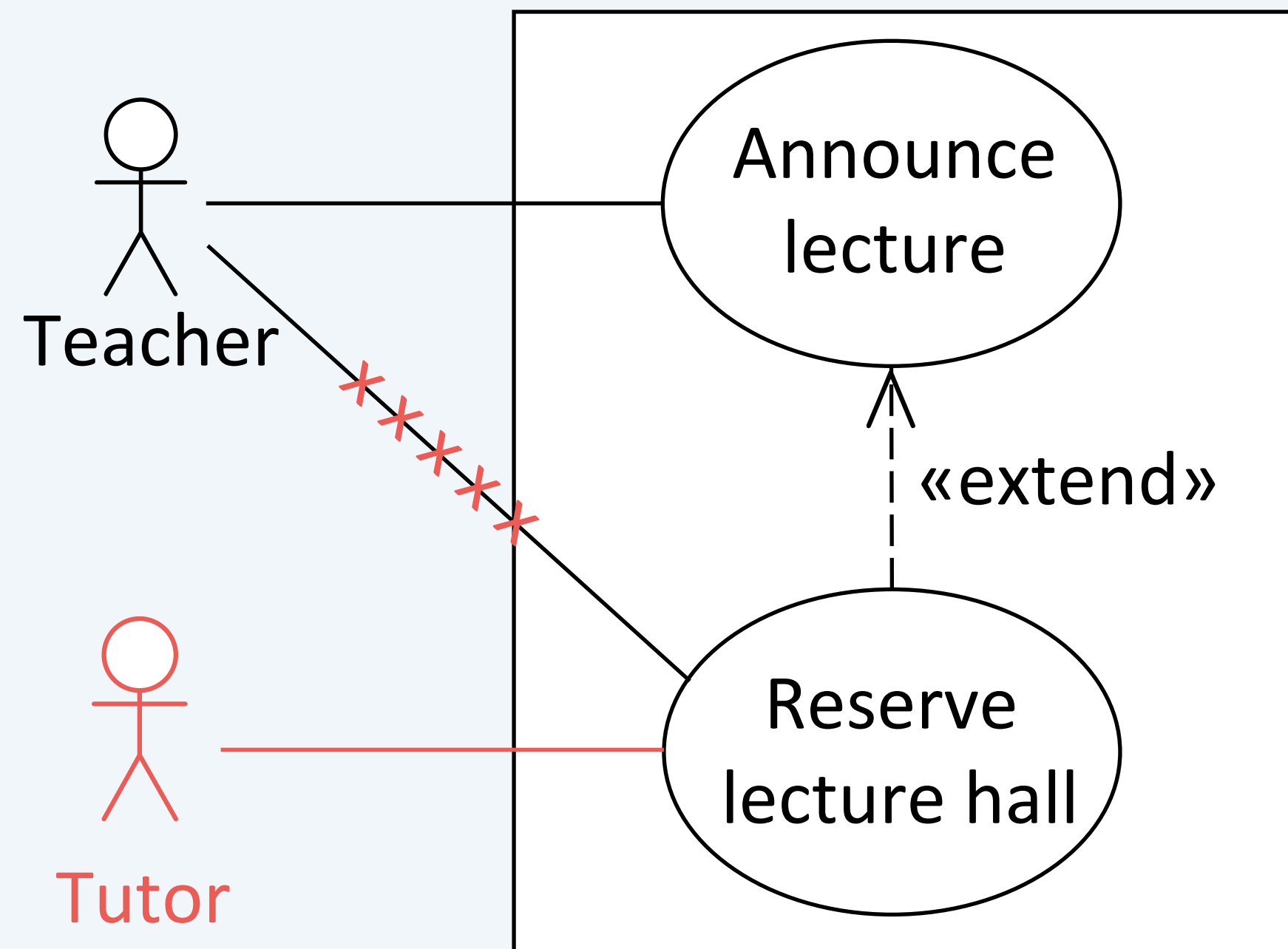
# Anwendungsfalldiagramm

## Typische Modellierungsfehler

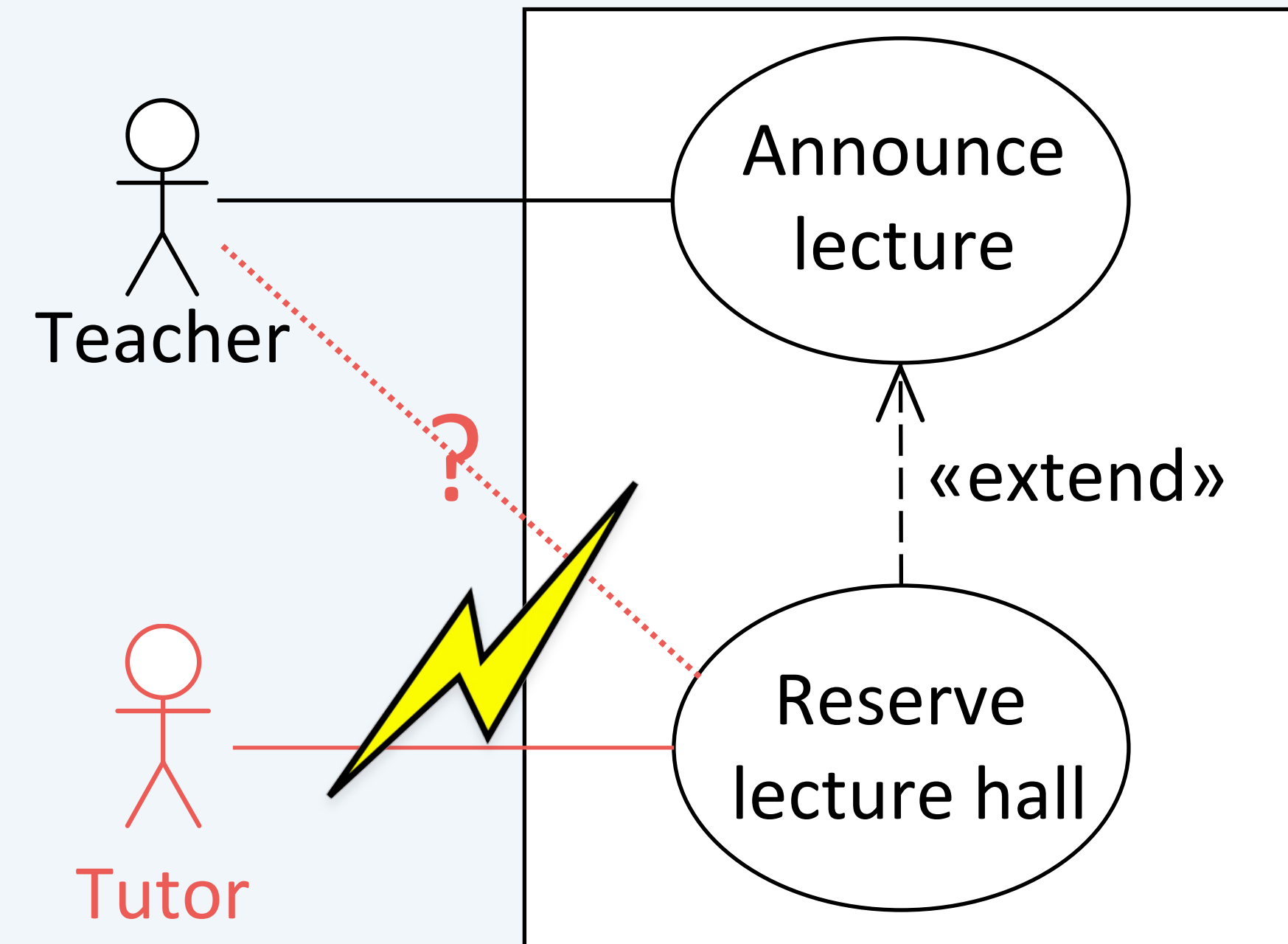


# «extend»

UML Standard..



..in der Praxis oft so:



unklar, wer den Hörsaal reserviert!

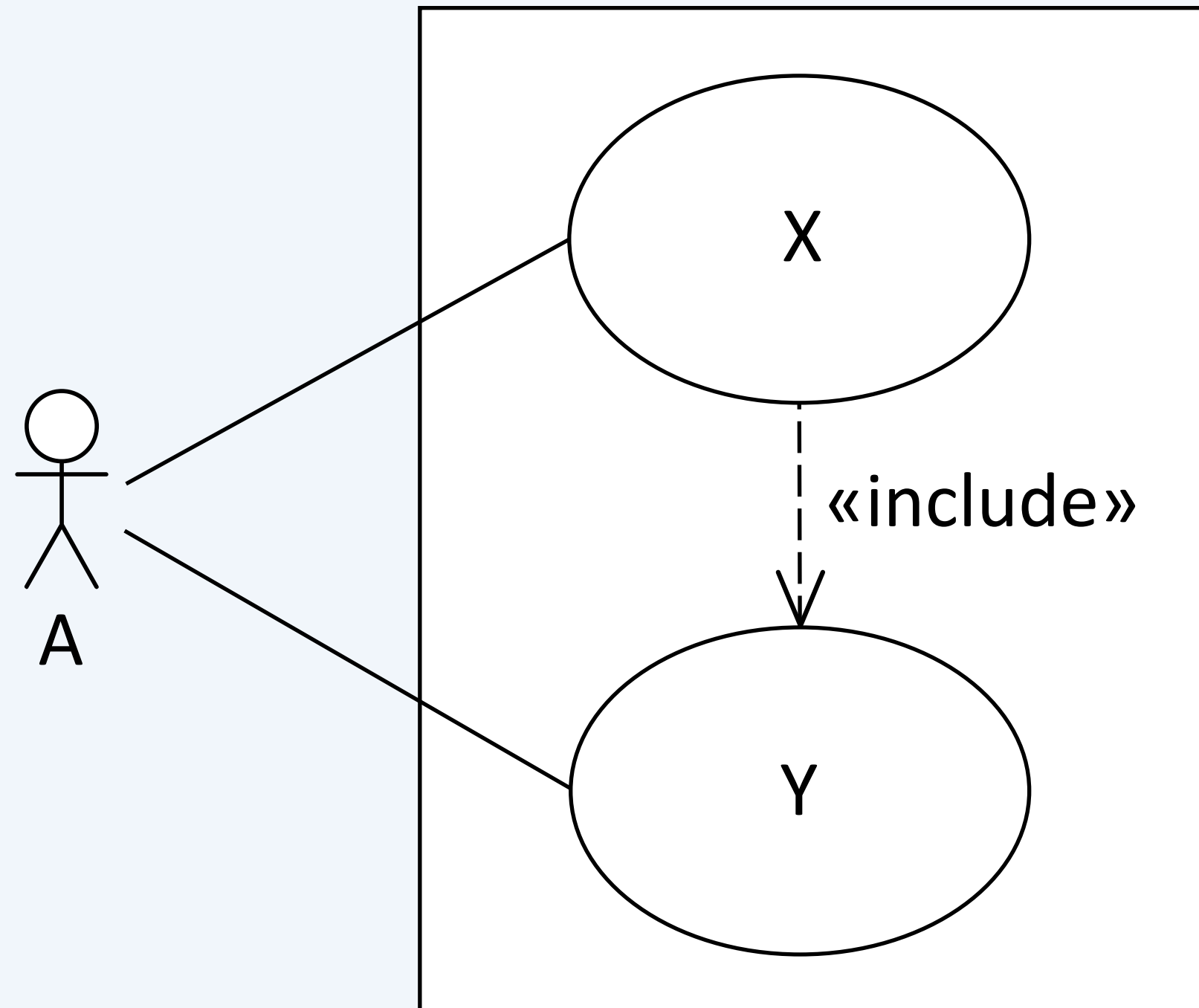
*Wie abbilden, dass jemand anders (alleine) den Hörsaal reserviert?*



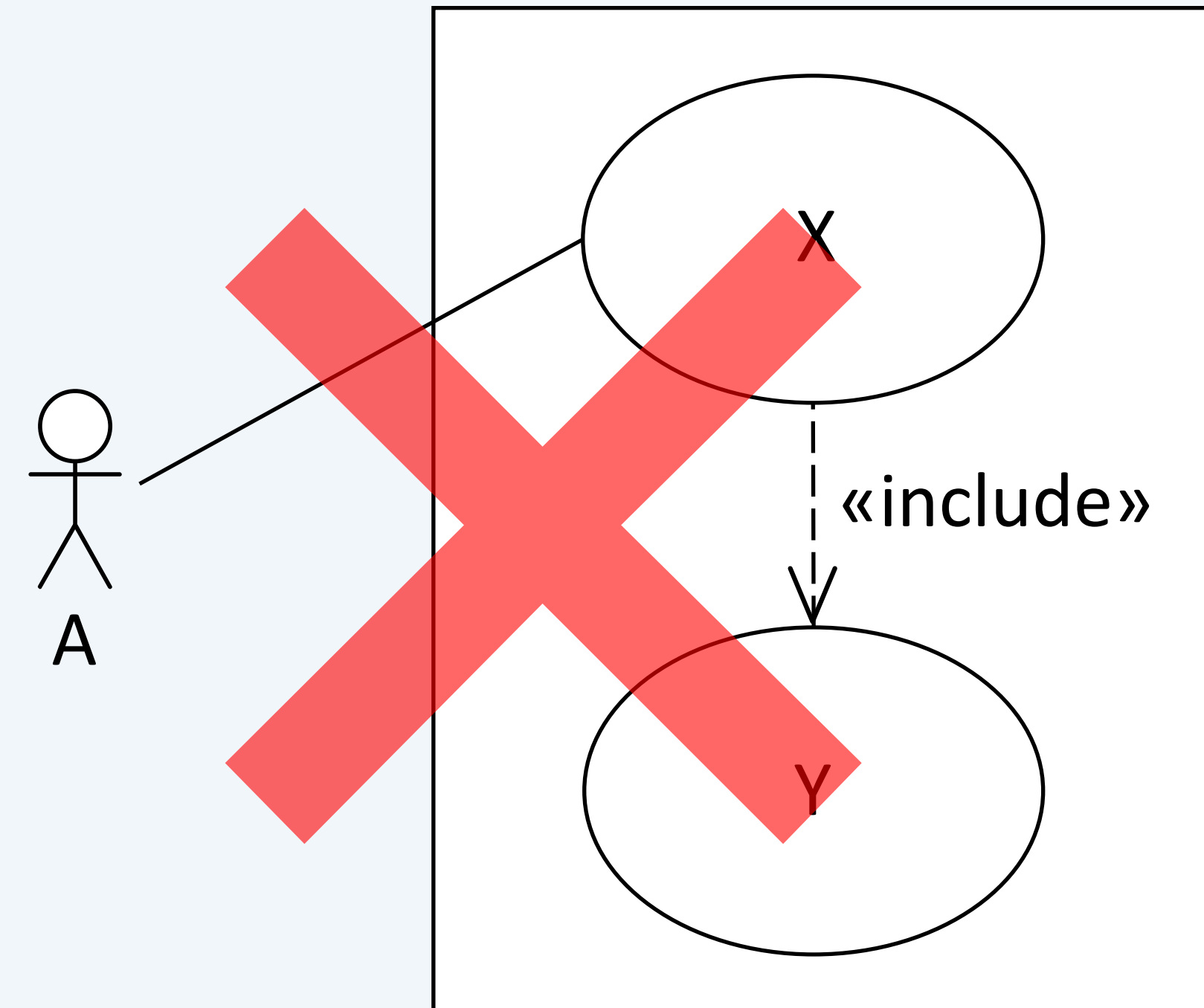
# «include»

- analog zu «extend»

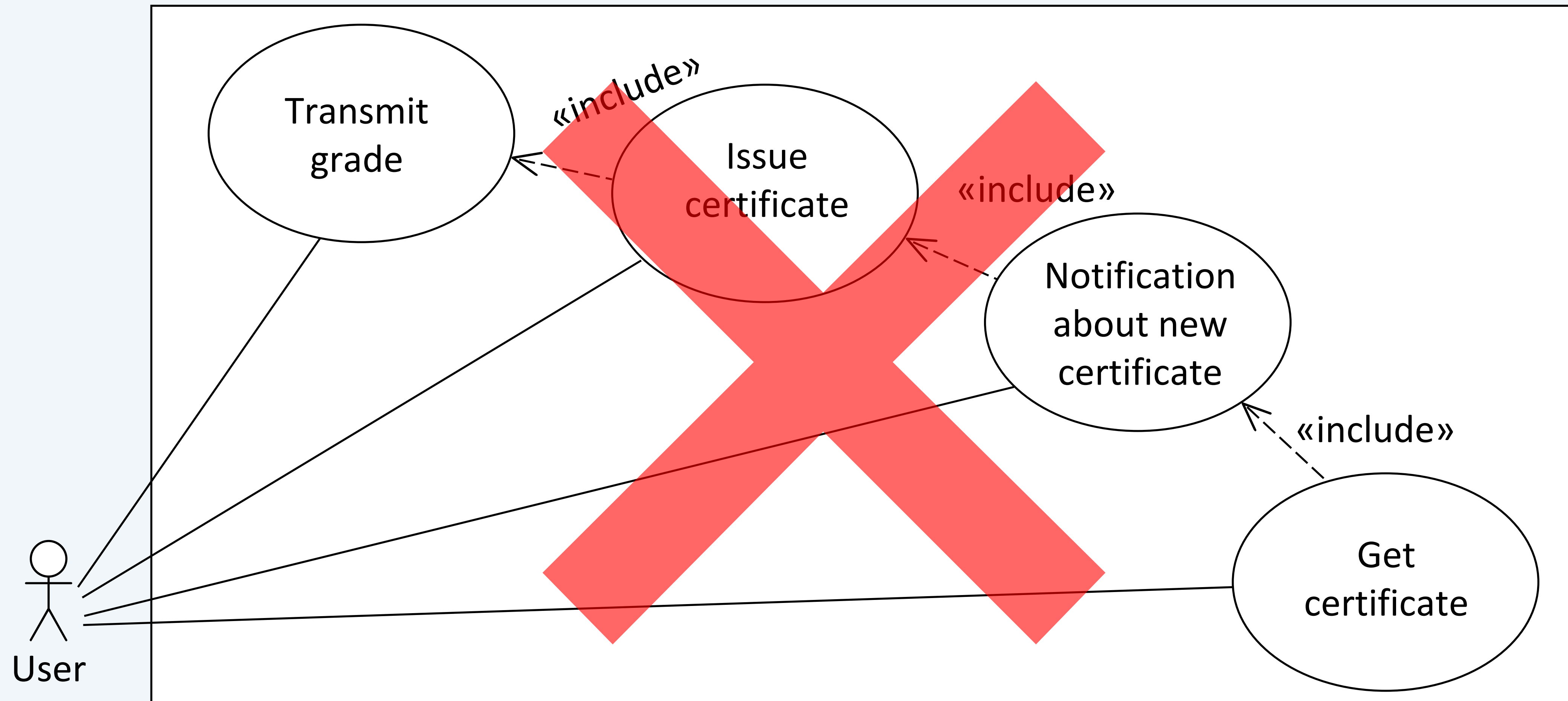
UML Standard..



..in der Praxis oft so:

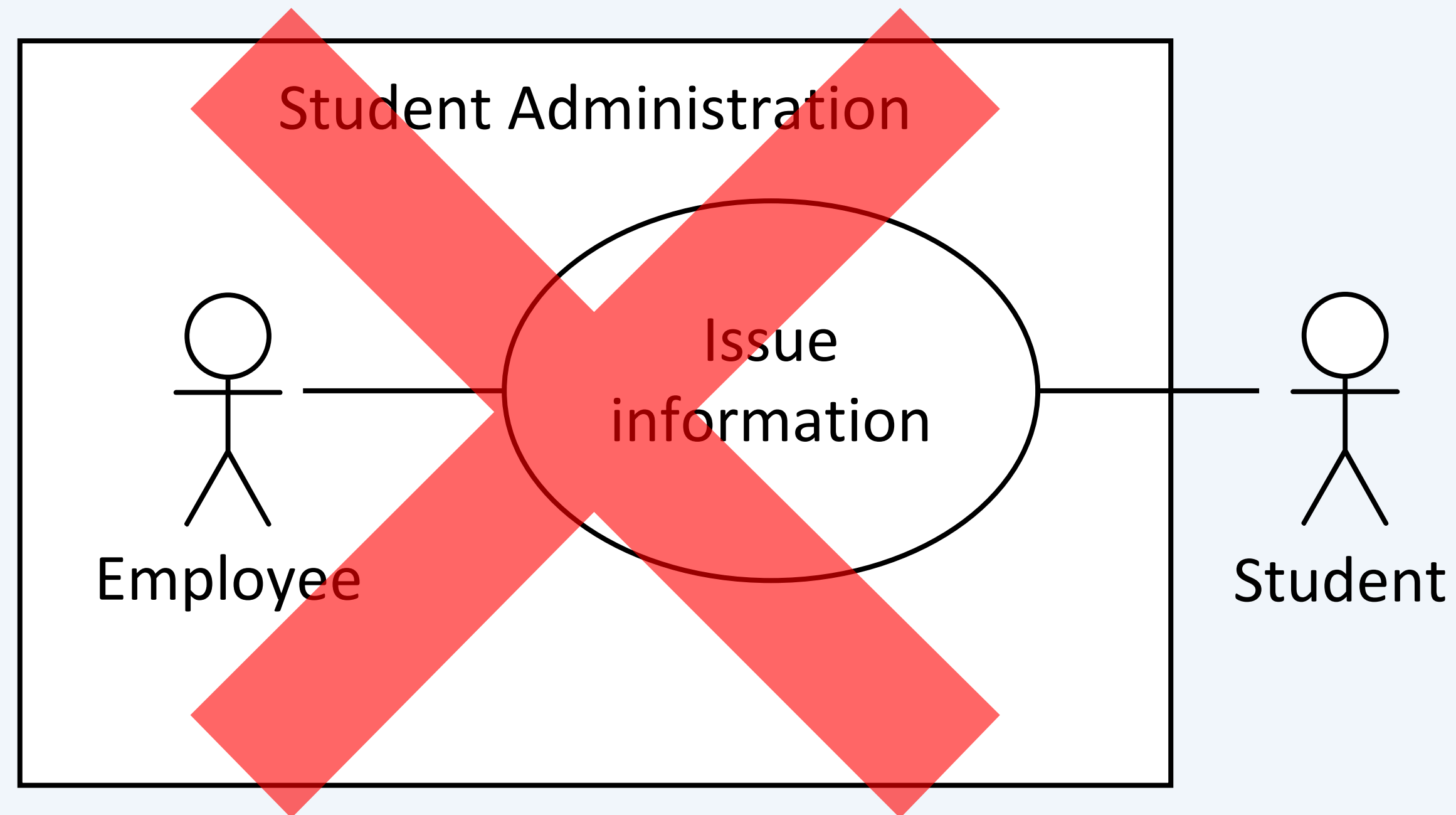


- Anwendungsfalldiagramme modellieren **keine Abläufe!**



# Systemgrenzen

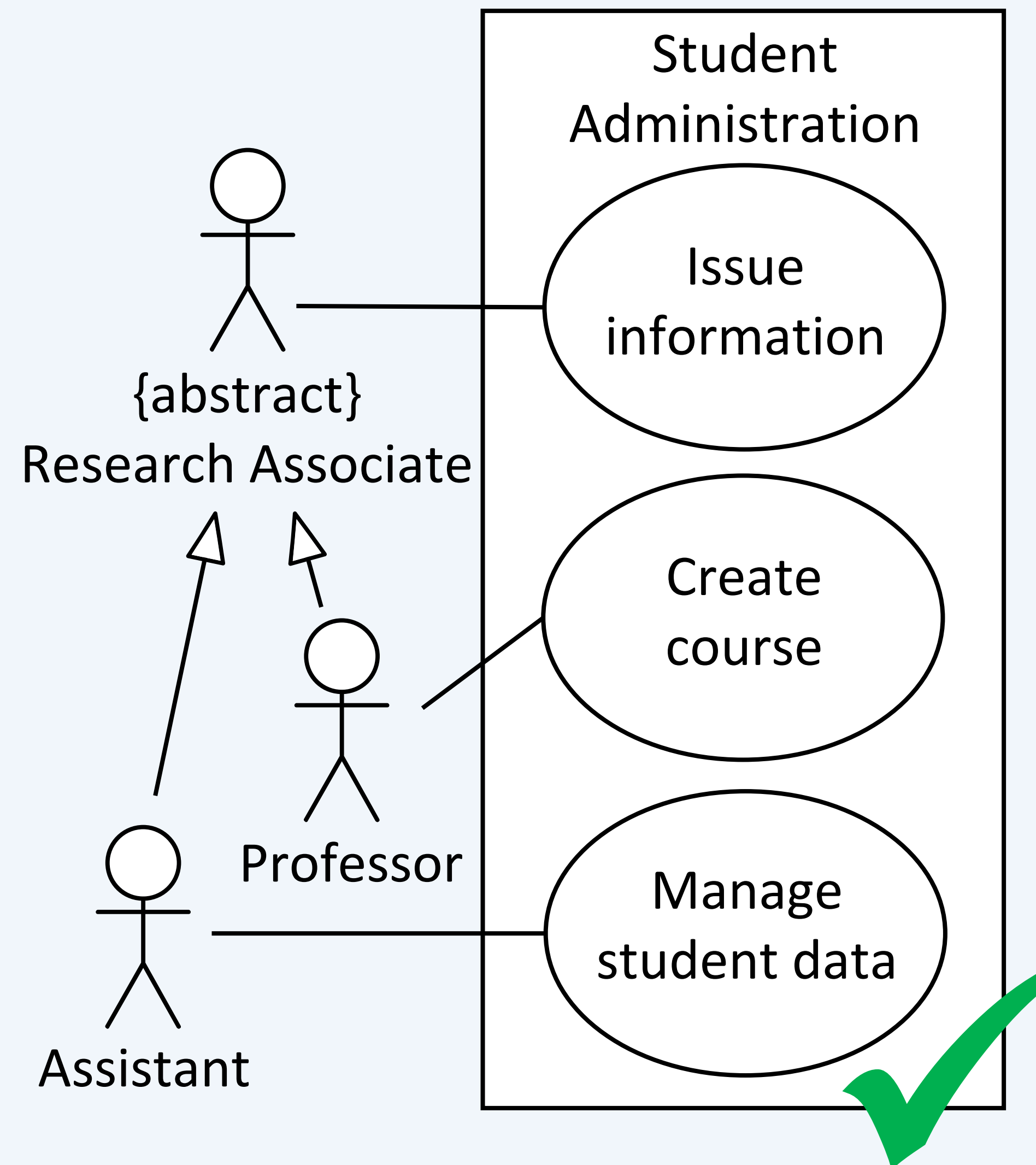
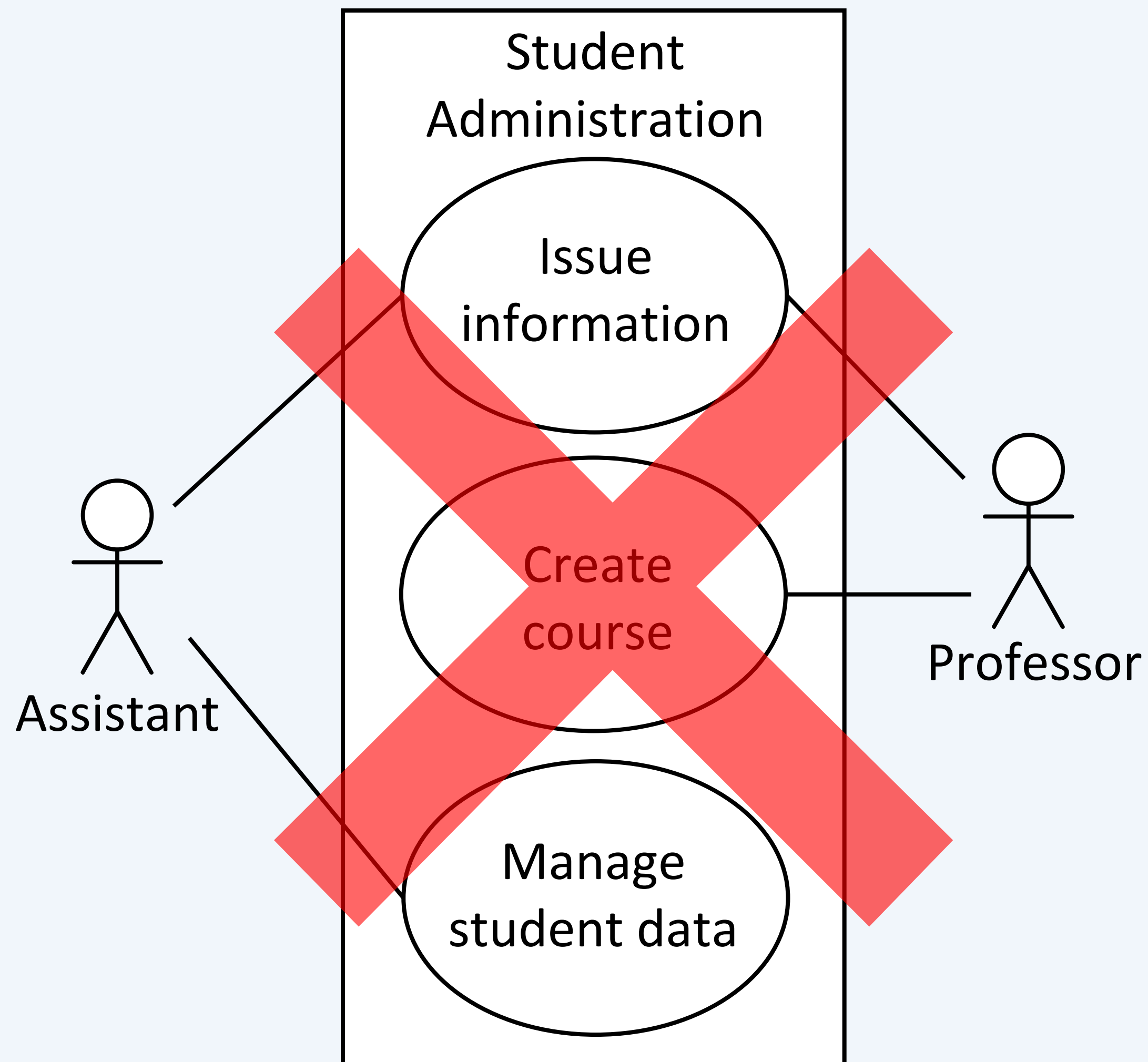
- Akteur\*innen sind immer außerhalb der Systemgrenzen!





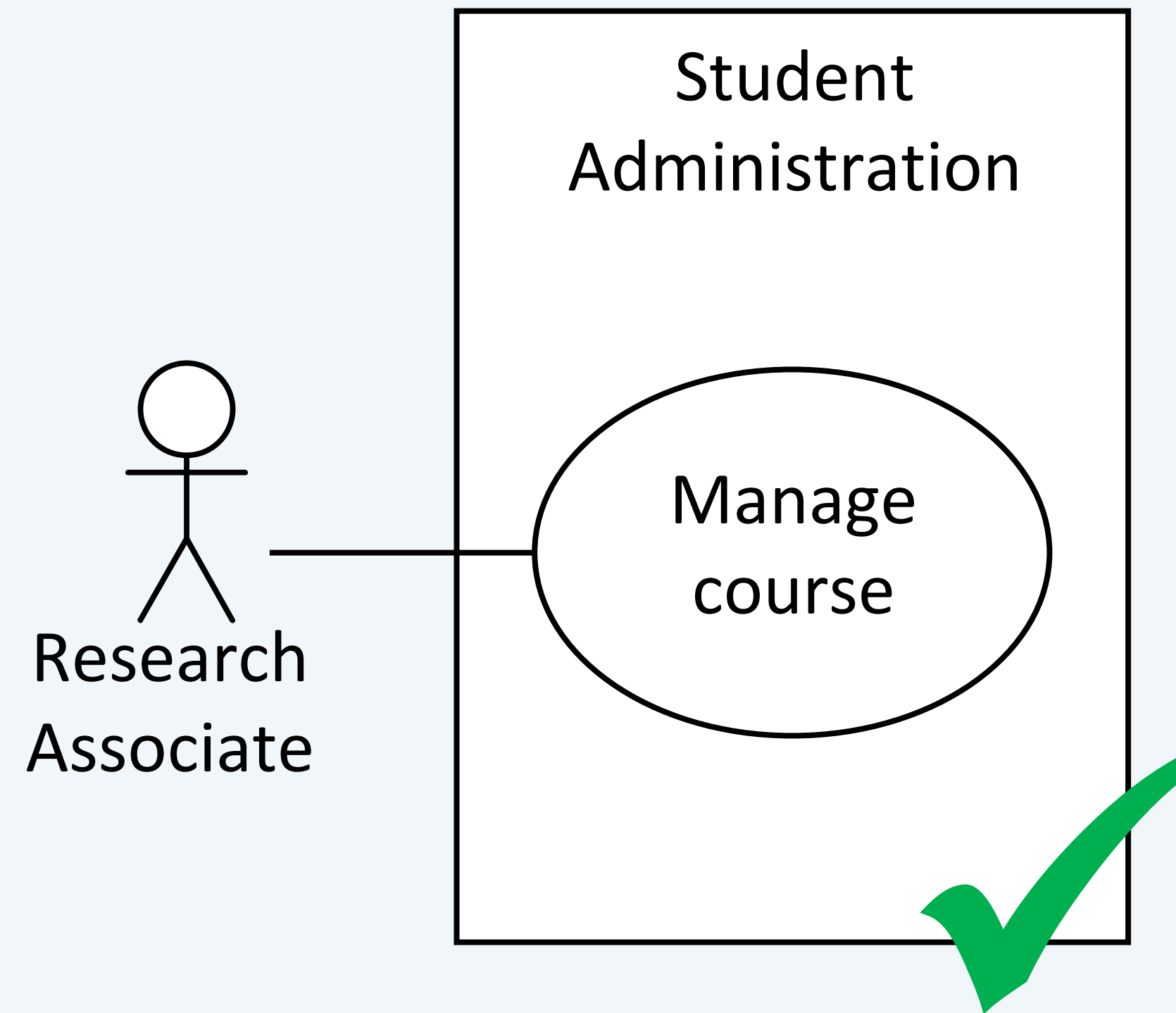
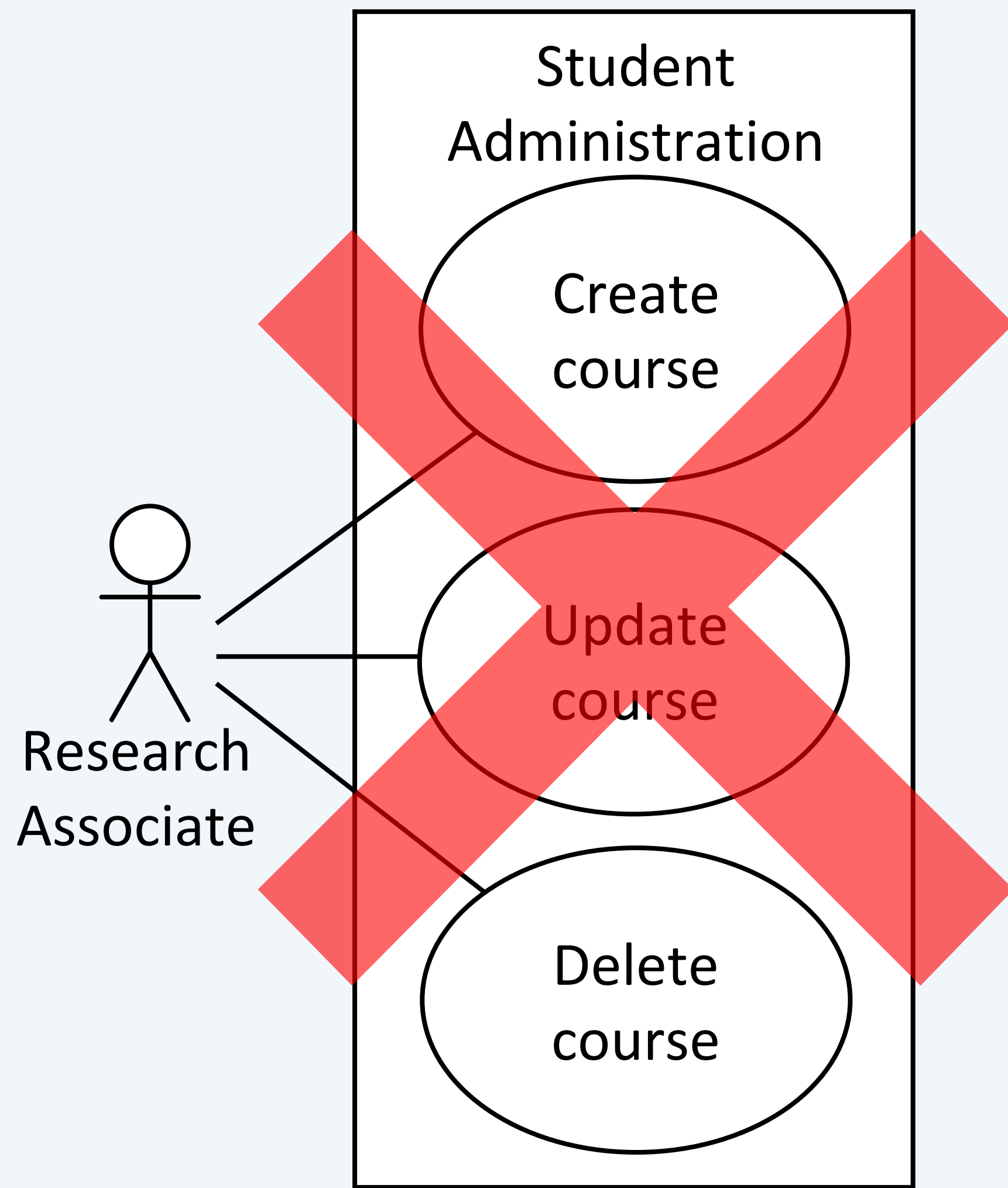
# „und“ bzw. „entweder oder“

- An **Issue information** sind entweder **Assistant** oder **Professor** beteiligt



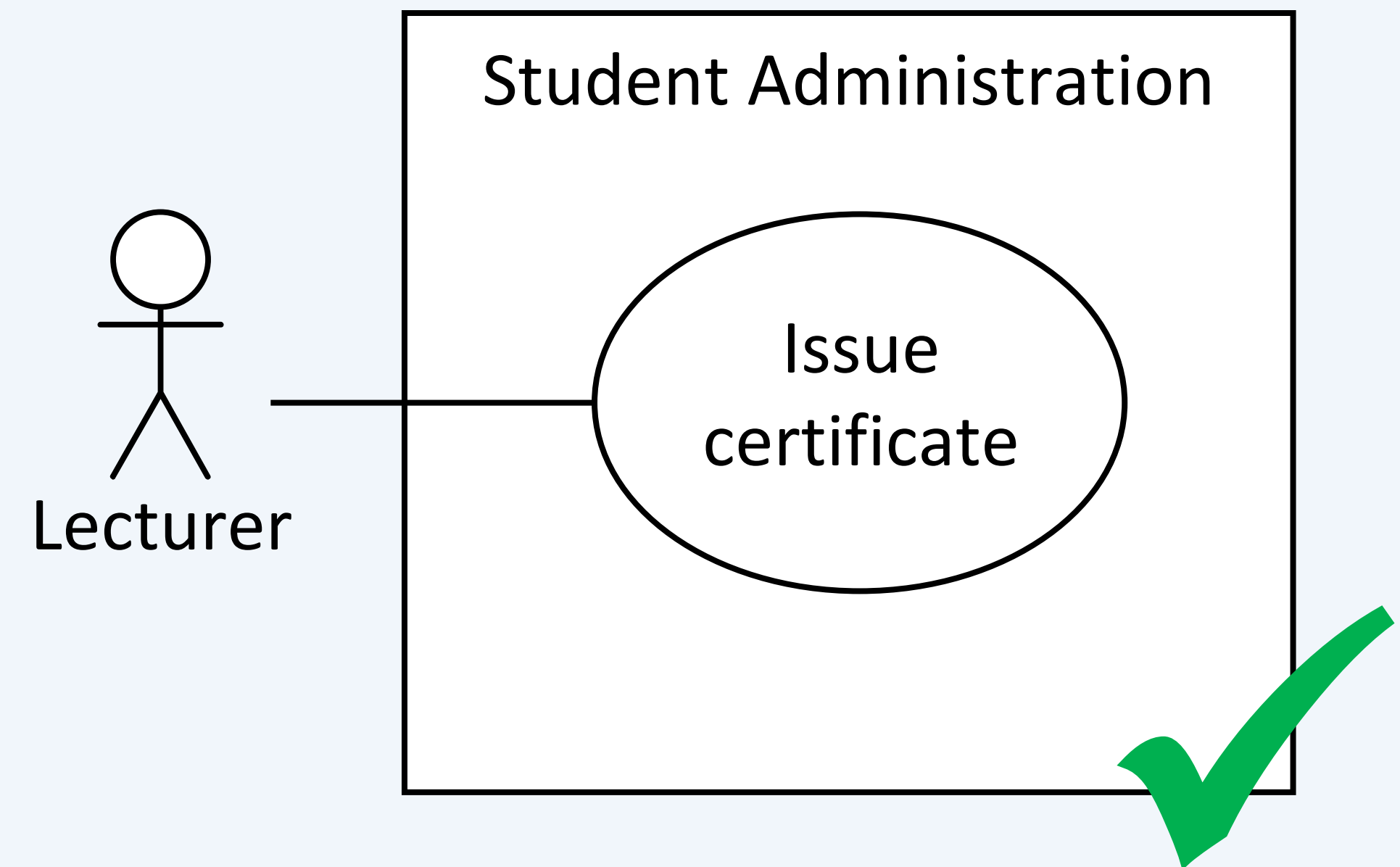
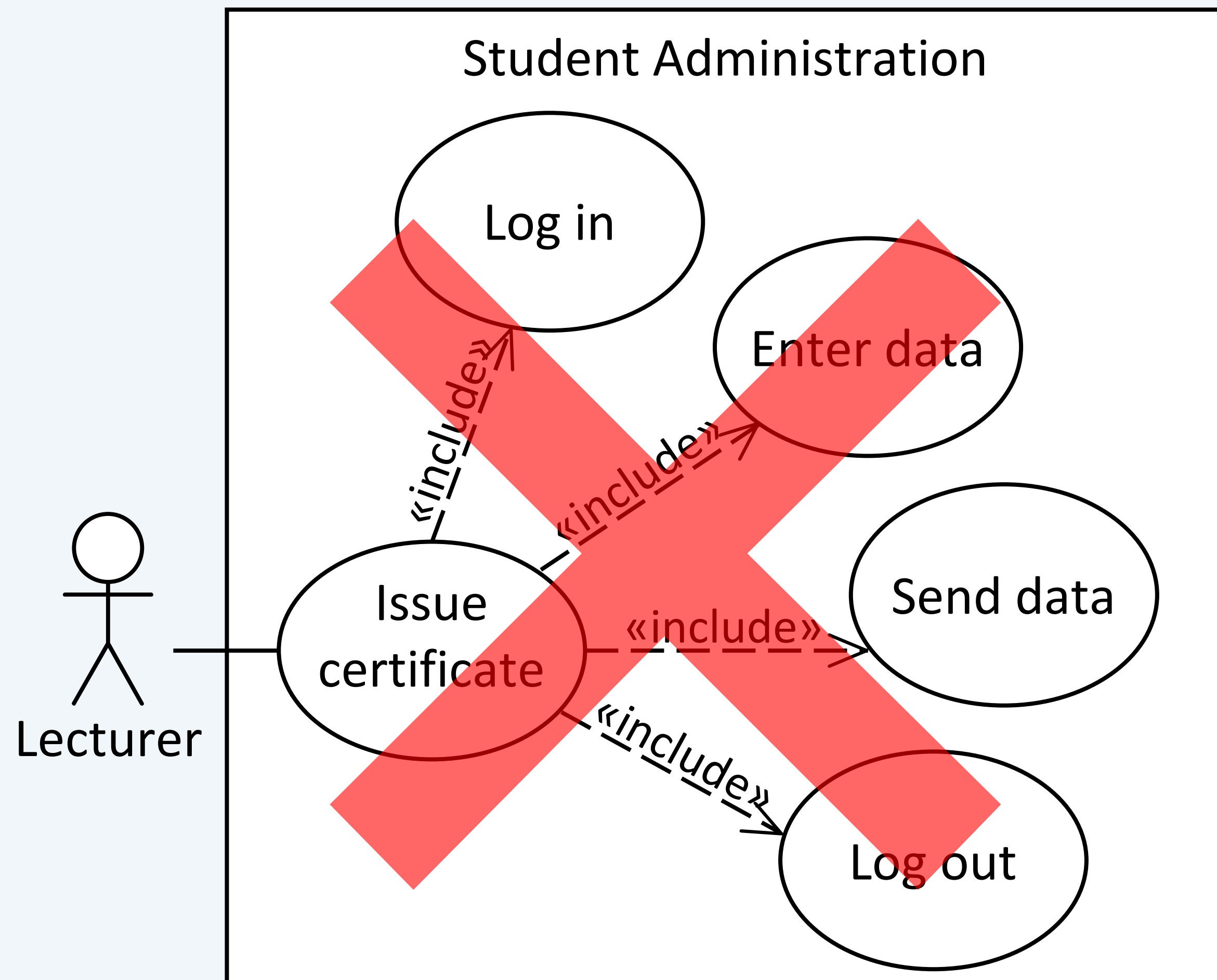
# Granularitätslevel

- Viele kleine Anwendungsfälle werden zu einem Anwendungsfall zusammengefasst, der die unterschiedlichen Möglichkeiten enthält



# Funktionale Zerlegung

- Die einzelnen Schritte werden immer gemeinsam in einer vordefinierten Reihenfolge ausgeführt, daher handelt es sich nur um **einen** Anwendungsfall  $\Rightarrow$  keine funktionale Zerlegung





# Anwendungsfalldiagramm

## Beispiel Studienabteilung



# Bsp.: Die Studienabteilung



- **Ziel:** vereinfachte Darstellung des Systems "Studienabteilung" einer Universität

## Verbale Beschreibung

- Viele wichtige Verwaltungstätigkeiten einer Universität werden über die Studienabteilung abgewickelt. Studierende können hier immatrikulieren und inskribieren, sowie sich aber auch wieder vom Studium abmelden.
- Studierende erhalten hier Zeugnisse. Mitarbeiter\*innen drucken die Zeugnisse aus. Zeugnisrelevante Daten werden durch Lehrende an die Studienabteilung übermittelt. Die Studierenden werden dann automatisch durch das Benachrichtigungssystem informiert.
- Es wird zwischen 2 Arten von Mitarbeiter\*innen unterschieden: a) solche, die sich ausschließlich mit der Verwaltung von Studierendendaten befassen (**service employee**) und b) jene, die alle restlichen Aufgaben erfüllen (**administration employee**), wobei aber beide Gruppen Auskünfte geben.
- **administration employee** stellen Zeugnisse aus, sobald Studierende diese abholen. Weiters erstellen sie Lehrveranstaltungen. Bei der Erstellung einer Lehrveranstaltung kann gleich ein Hörsaal reserviert werden.

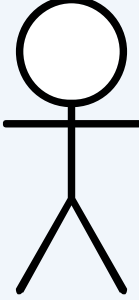
## Wer sind die Benutzer\*innen?

- Viele wichtige Verwaltungstätigkeiten einer Universität werden über die Studienabteilung abgewickelt. **Studierende** können hier immatrikulieren und inskribieren, sowie sich aber auch wieder vom Studium abmelden.
- Studierende erhalten hier Zeugnisse. Mitarbeiter\*innen drucken die Zeugnisse aus. Zeugnisrelevante Daten werden durch **Lehrende** an die Studienabteilung übermittelt. Die Studierenden werden dann automatisch durch das **Benachrichtigungssystem** informiert.
- Es wird zwischen 2 Arten von **Mitarbeiter\*innen** unterschieden: a) solche, die sich ausschließlich mit der Verwaltung von Studierendendaten befassen (**service employee**) und b) jene, die alle restlichen Aufgaben erfüllen (**administration employee**), wobei aber beide Gruppen Auskünfte geben.
- **administration employee** stellen Zeugnisse aus, sobald Studierende diese abholen. Weiters erstellen sie Lehrveranstaltungen. Bei der Erstellung einer Lehrveranstaltung kann gleich ein Hörsaal reserviert werden.

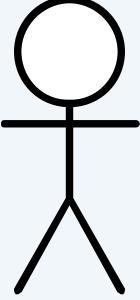


# Bsp.: Akteur\*innen (2/2)

- Wer sind die Benutzer\*innen?

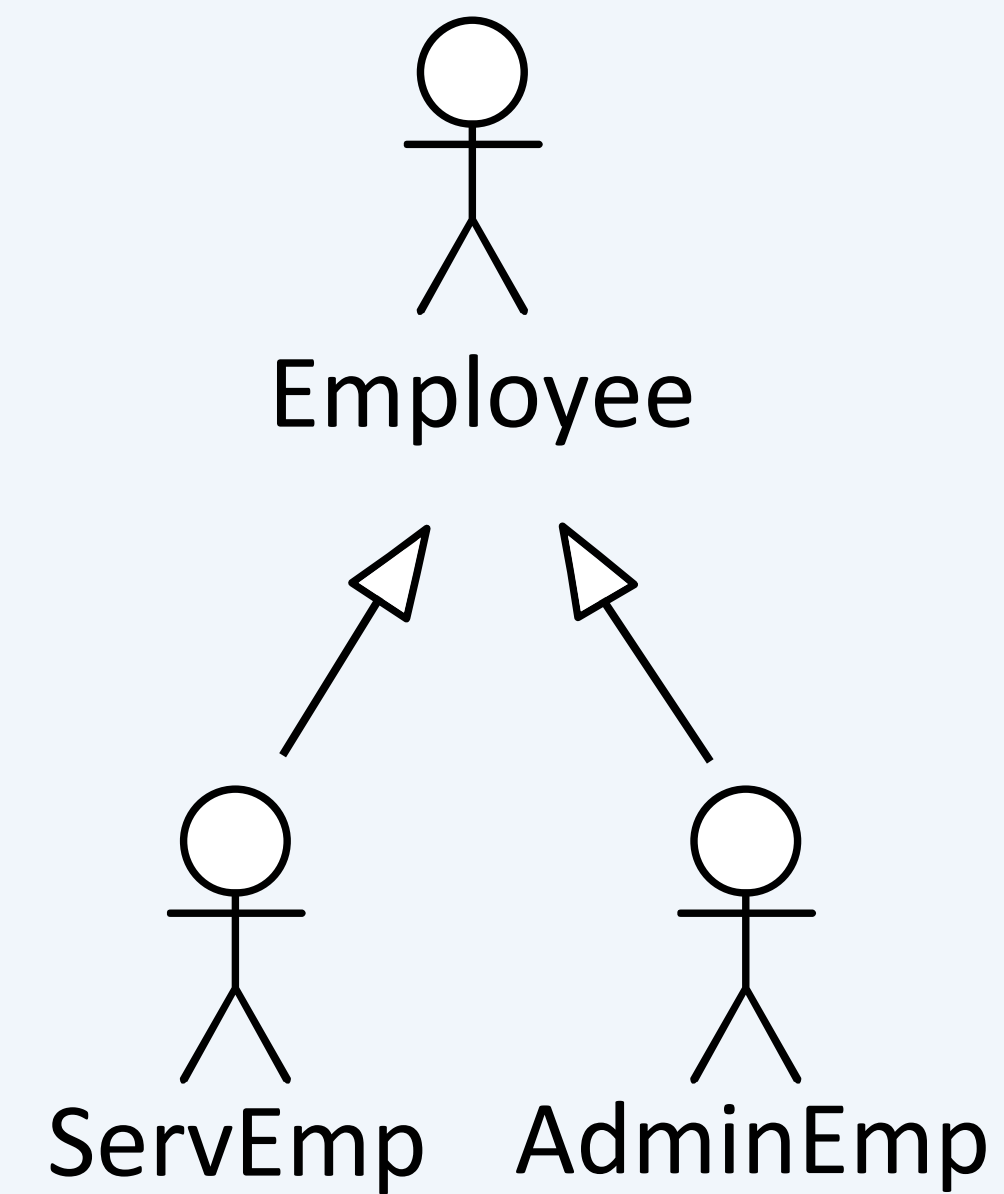


Lecturer



Student

«actor»  
Notification System

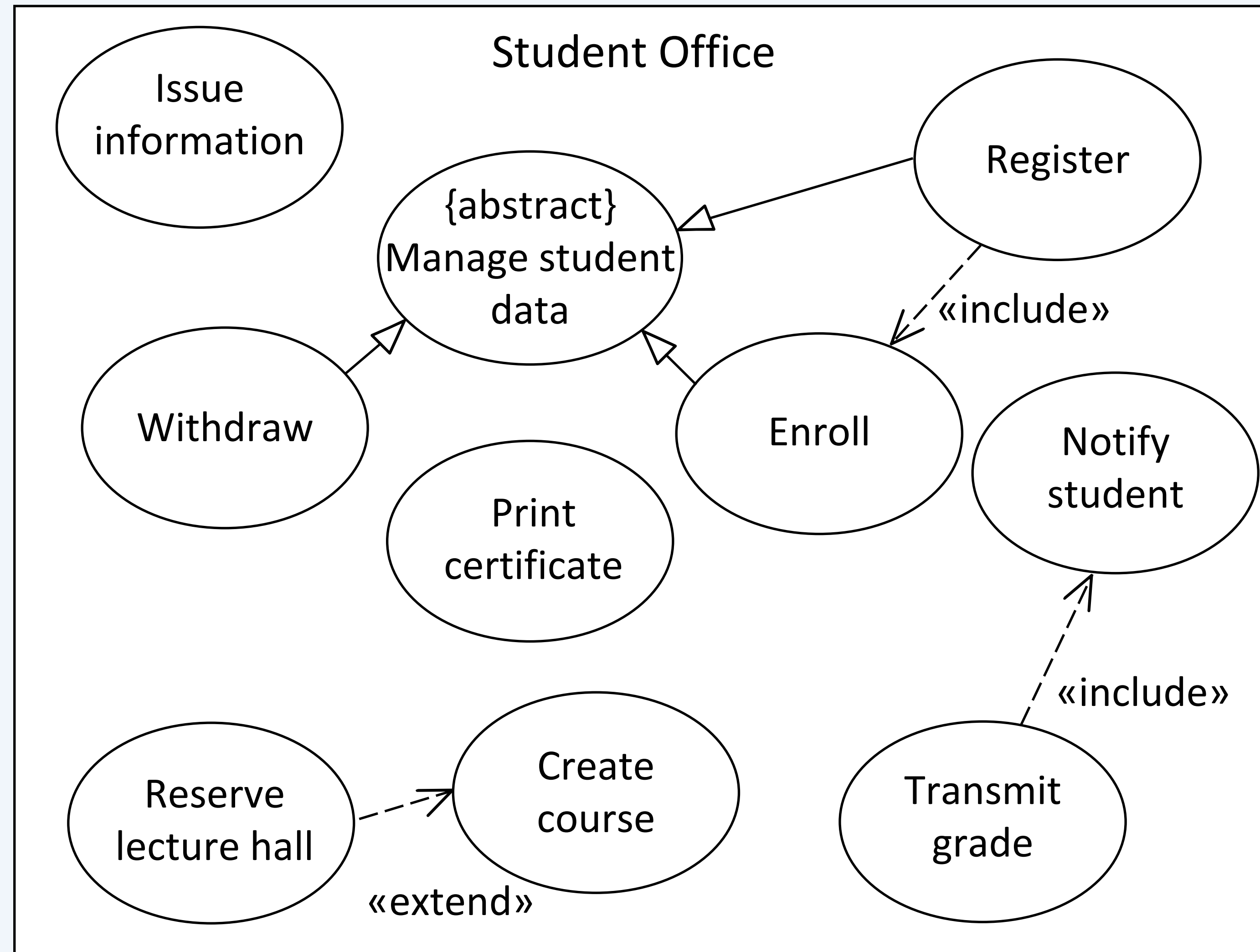


## Welche Funktionen bietet die Studienabteilung?

- Viele wichtige Verwaltungstätigkeiten einer Universität werden über die Studienabteilung abgewickelt. Studierende können hier **immatrikulieren** und **inskribieren**, sowie sich aber auch wieder vom Studium **abmelden**.
- Studierende erhalten hier Zeugnisse. Mitarbeiter\*innen **drucken die Zeugnisse** aus. **Zeugnisrelevante Daten** werden durch Lehrende an die Studienabteilung **übermittelt**. Die **Studierenden** werden dann automatisch durch das Benachrichtigungssystem **informiert**.
- Es wird zwischen 2 Arten von Mitarbeiter\*innen unterschieden: a) solche, die sich ausschließlich mit der Verwaltung von Studierendendaten befassen (**service employee**) und b) jene, die alle restlichen Aufgaben erfüllen (**administration employee**), wobei aber beide Gruppen **Auskünfte geben**.
- **administration employee** stellen Zeugnisse aus, sobald Studierende diese abholen. Weiters erstellen sie Lehrveranstaltungen. Bei der **Erstellung einer Lehrveranstaltung** kann gleich ein **Hörsaal reserviert** werden.

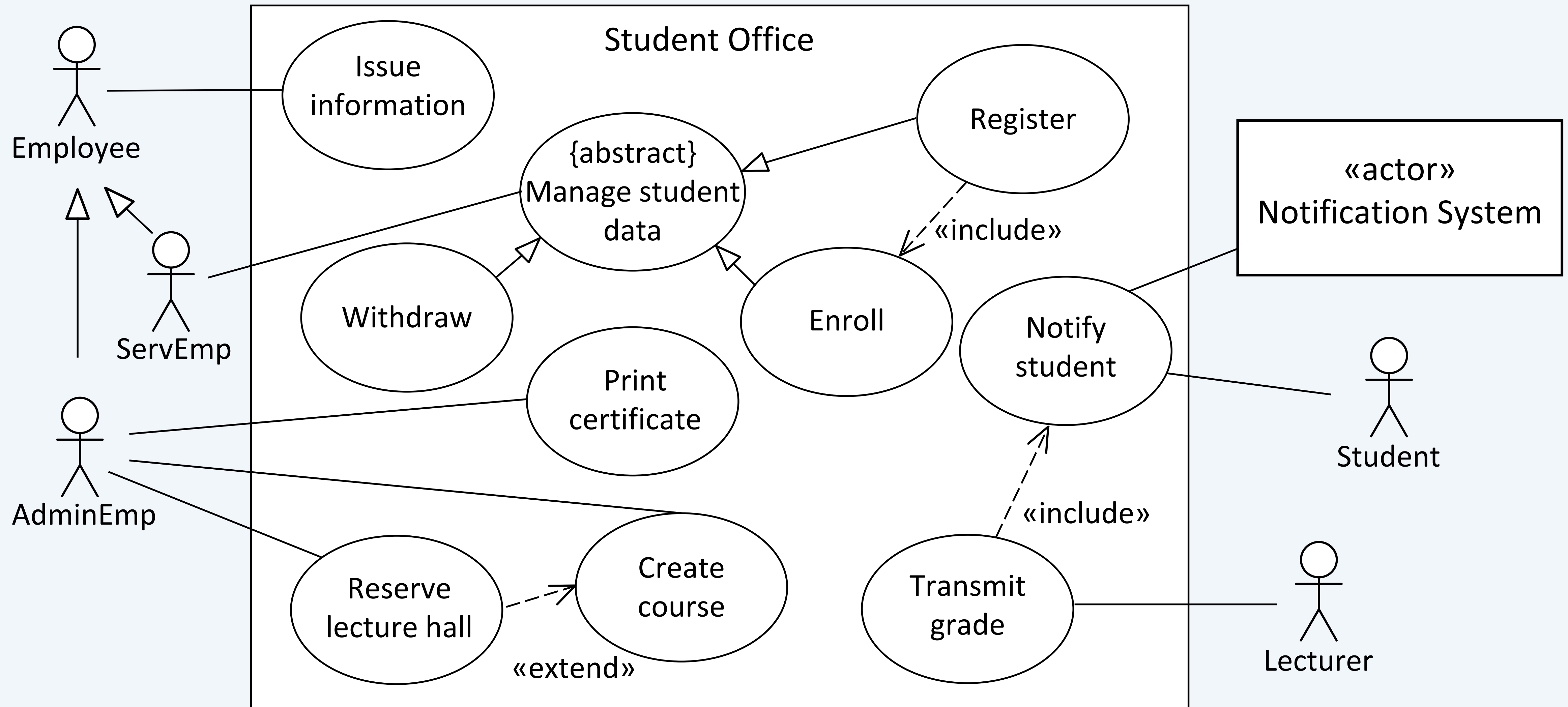
# Bsp.: Anwendungsfälle (2/2)

- Welche Funktionen bietet die Studienabteilung?



# Bsp.: Assoziationen

- Welche Akteur\*innen benutzen welche Anwendungsfälle?





# Bsp.: Beschreibung (1/2)

- **Name:** Zeugnis drucken
- **Kurzbeschreibung:** Auf Anfrage eines/einer Studierenden druckt ein `AdminEmp` das Zeugnis auf Papier aus.
- **Vorbedingung:** Studierende\*r wurde bereits benotet und alle relevanten Daten wurden bereits übermittelt.
- **Nachbedingung:** Zeugnis befindet sich in ausgedruckter Form bei Studierender/Studierendem
- **Fehlersituationen:** Der Drucker funktioniert nicht
- **Systemzustand im Fehlerfall:** Zeugnis wurde nicht gedruckt
- **Akteur\*innen, die mit dem Anwendungsfall kommunizieren:** `AdminEmp`
- **Trigger:** Studierende\*r möchte ein ausgedrucktes Zeugnis

# Bsp.: Beschreibung (2/2)



## ■ Standardablauf:

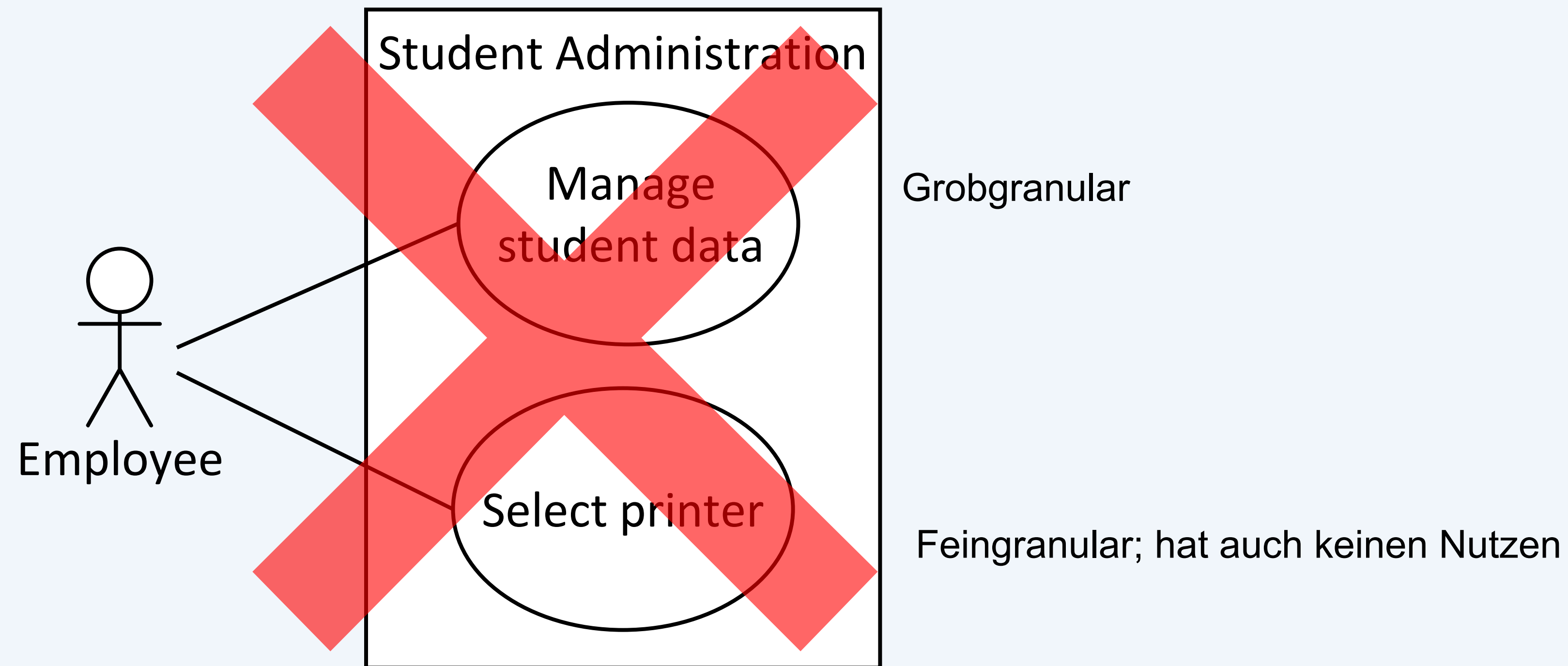
- (1) Student\*in geht in Studienabteilung und bittet um ein gedrucktes Zeugnis.
- (2) **AdminEmp** gibt die Matrikelnummer ein.
- (3) **AdminEmp** wählt das Zeugnis aus.
- (4) **AdminEmp** wählt „drucken“ aus.
- (5) System bestätigt dass das Zeugnis gedruckt wurde
- (6) Zeugnis wird Student\*in übergeben.

## ■ Alternativabläufe:

- (1') Student\*in bittet via Email um das gedruckte Zeugnis.
- (2-5) wie oben
- (6') Zeugnis wird per Post zugeschickt

# Bsp.: Modellierungstipps (1/2)

- Die Anwendungsfälle sollten auf der gleichen Abstraktionsebene sein.



# Bsp.: Modellierungstipps (2/2)

- Es sollen nur Anwendungsfälle und Akteur\*innen inkludiert werden, die relevant sind.
- Natürlich gibt es noch andere Personen, die etwas mit der Studienabteilung zu tun haben, die aber in diesem Kontext der Modellierung vernachlässigbar sind.

■ Bsp.:

