

# Objektorientierte Analyse

## 36. Analyse mit querschnittlichen Verfeinerungen: Beispiel EU-Rent

Prof. Dr. rer. nat. habil. Uwe Aßmann

Institut für Software- und  
Multimediatechnik

Lehrstuhl Softwaretechnologie

Fakultät für Informatik

TU Dresden

Version 16-0.3, 25.06.16

- 1) Beispiel EU-Rent
- 2) Erster Versuch
- 3) Zweite Verfeinerung
- 4) Dritte Verfeinerung



DRESDEN  
concept  
Exzellenz aus  
Wissenschaft  
und Kultur

# Überblick Teil III: Objektorientierte Analyse (OOA)

2

Softwaretechnologie (ST)

1. Überblick Objektorientierte Analyse
  1. Strukturelle Modellierung mit CRC-Karten
2. Strukturelle metamodelgetriebene Modellierung mit UML
  - Analyse des Domänenmodells: Strukturelle metamodelgetriebene Modellierung
    1. Modellierung von komplexen Objekten
      1. Systemanalyse: Strukturelle Modellierung für Kontextmodell und Top-Level-Architektur
3. Analyse von funktionalen Anforderungen (Verhaltensmodell)
  1. Funktionale Verfeinerung: Dynamische Modellierung von Lebenszyklen mit Aktionsdiagrammen
  2. Funktionale querschneidende Verfeinerung: Szenarienanalyse mit Anwendungsfällen, Kollaborationen und Interaktionsdiagrammen



## 4. Beispiel Fallstudie EU-Rent

# Die Autoverleihfirma “EU-Rent”

- ▶ EU-Rent ist eine Fallstudie, die von der internationalen Forschungsgemeinschaften zur Demonstration von Analyse eingesetzt wird
  - Formulierung von Geschäftsregeln (business rules in business model, ST-II)
  - Domänenmodell
  - Anwendungsfalldiagramme
  - Aktivitätsdiagramme und Statecharts für Arbeitsabläufe, auch zur Szenarioanalyse von Anwendungsfalldiagrammen
  - Kontextmodelle
- ▶ z.B. <http://www.kdmanalytics.com/sbvr/EU-Rent.html>

# Aktivitäten von EU-Rent

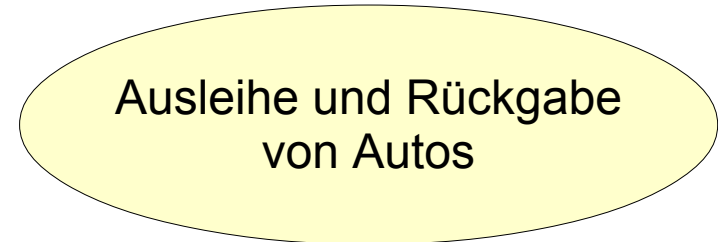
4

Softwaretechnologie (ST)

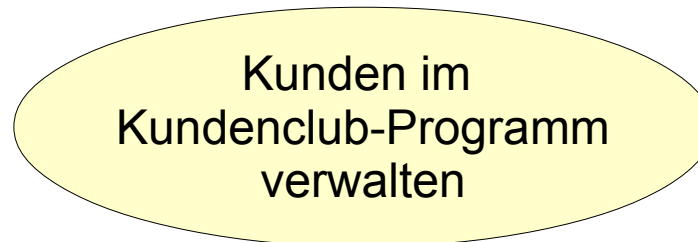
- ▶ Bei der Analyse geht man i.d.R. von textuellen “user stories” aus (z.B. Interviews, Tonbandmitschnitte, Protokolle, ...). Diese gilt es zu analysieren
- ▶ Wir betrachten 3 Gruppen von Aktivitäten. Ein einzelnes Anwendungsfalldiagramm reicht nicht aus, das ganze Informationssystem zu modellieren



vorbestellt  
ad-hoc abgeholt



Abholung, Rückgabe, auch  
an anderer Zweigstelle  
Rabatte und Sonderangebote



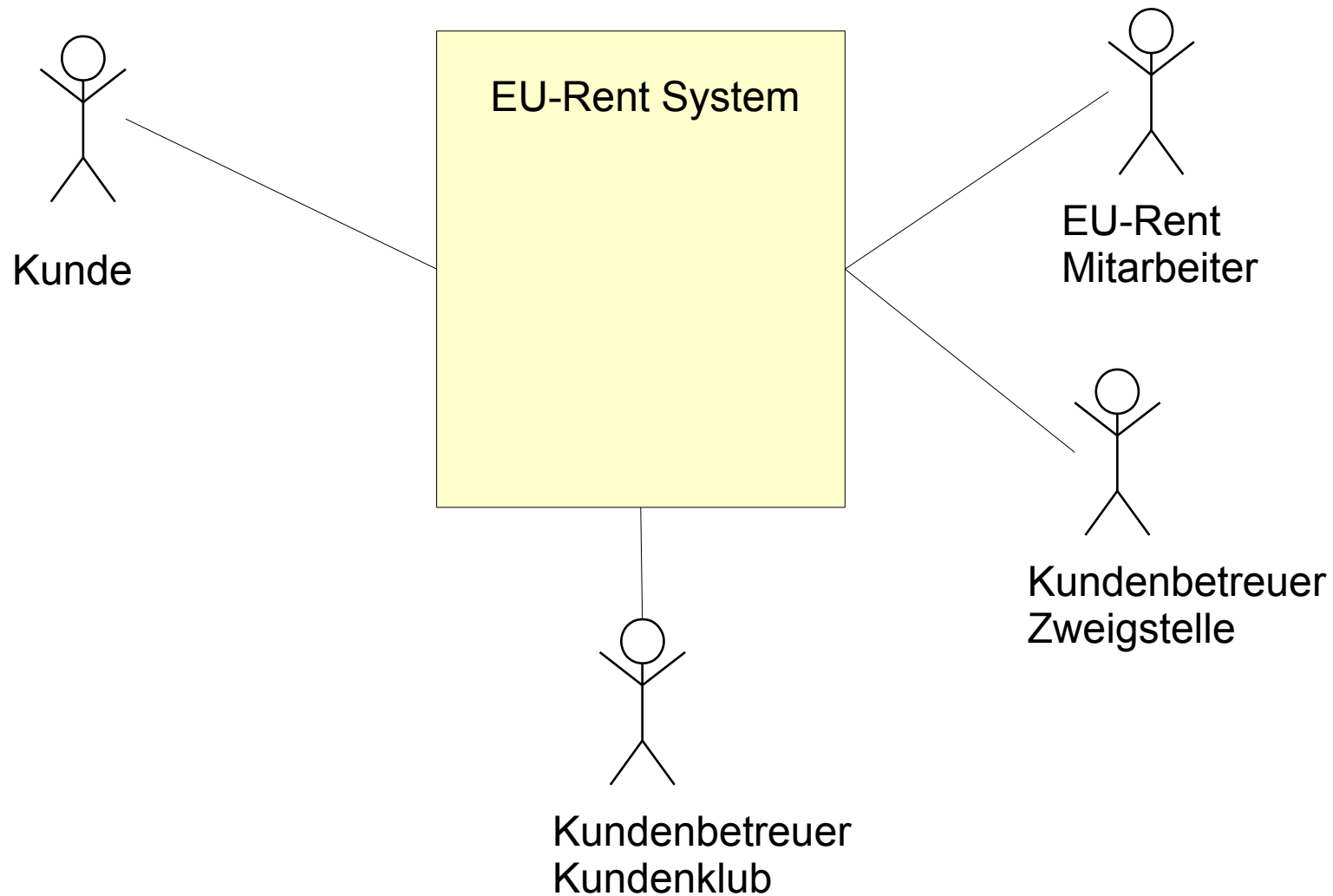
Rabattpunkte-Programm

# Stakeholder (hier: Akteure)

5

Softwaretechnologie (ST)

- ▶ Akteure stehen mit dem System in Beziehung. Noch keine Festlegung von Systemfunktionen

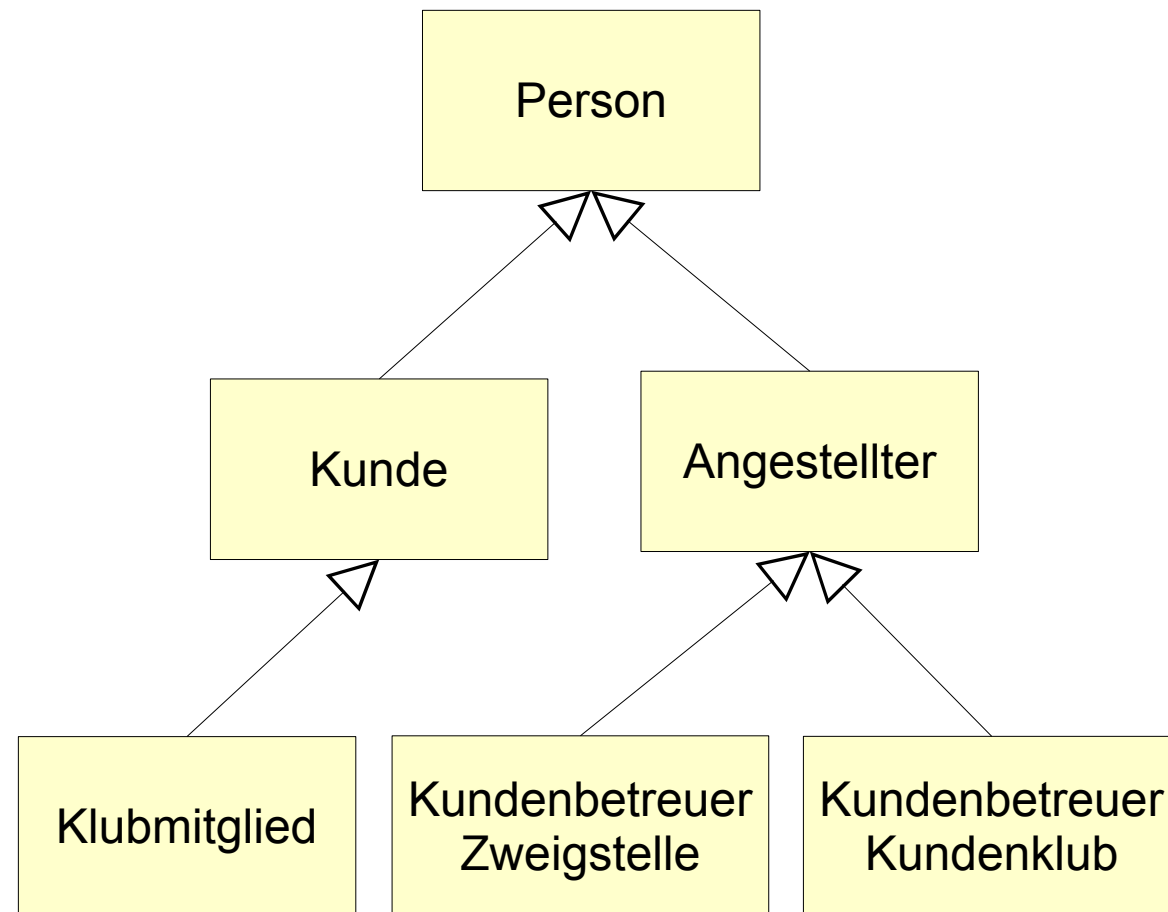
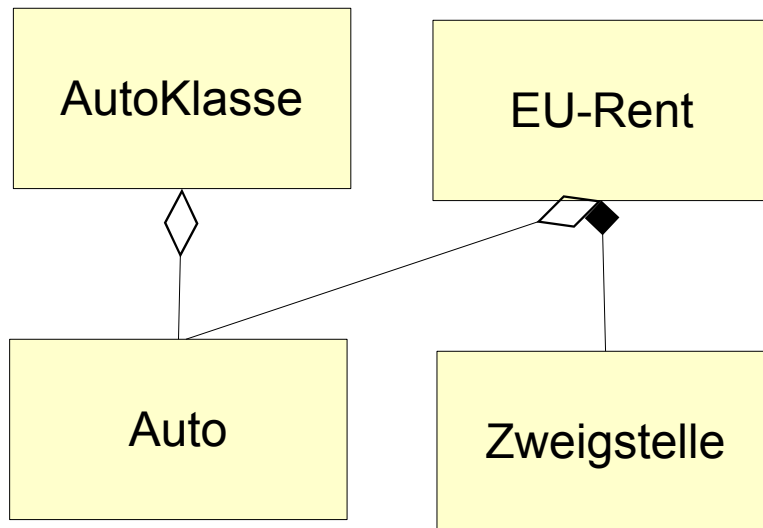


# Strukturelle Analyse für Domänenmodell, 1. Streich

6

Softwaretechnologie (ST)

- ▶ Domänenmodelle müssen die Organisation, Dinge, Geschäftsvorgänge, -objekte modellieren. Die involvierten Personen werden mitmodelliert
- ▶ Noch ohne Kardinalitäten und Attribute

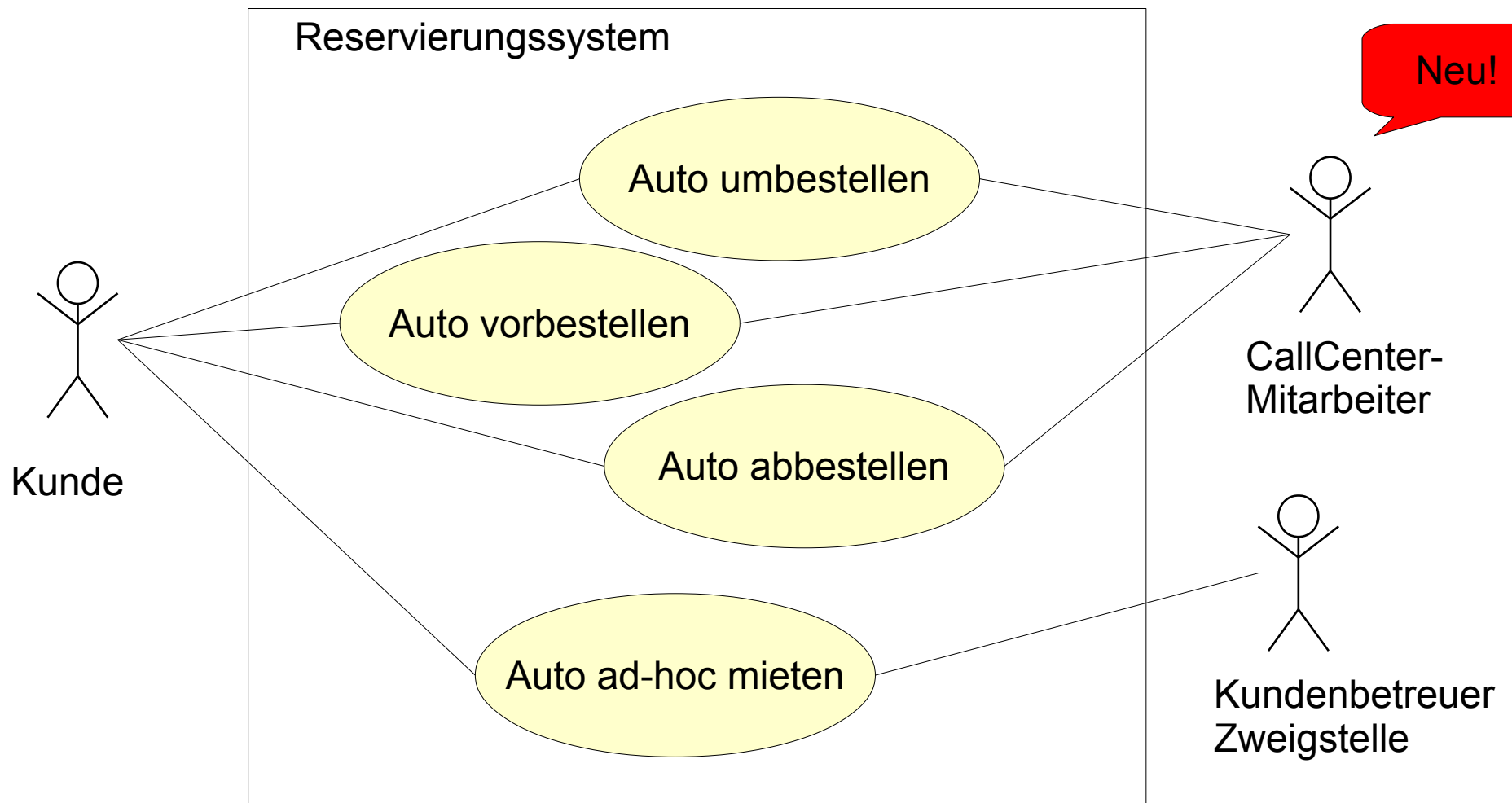


# A) Reservierungssystem von EU-Rent

7

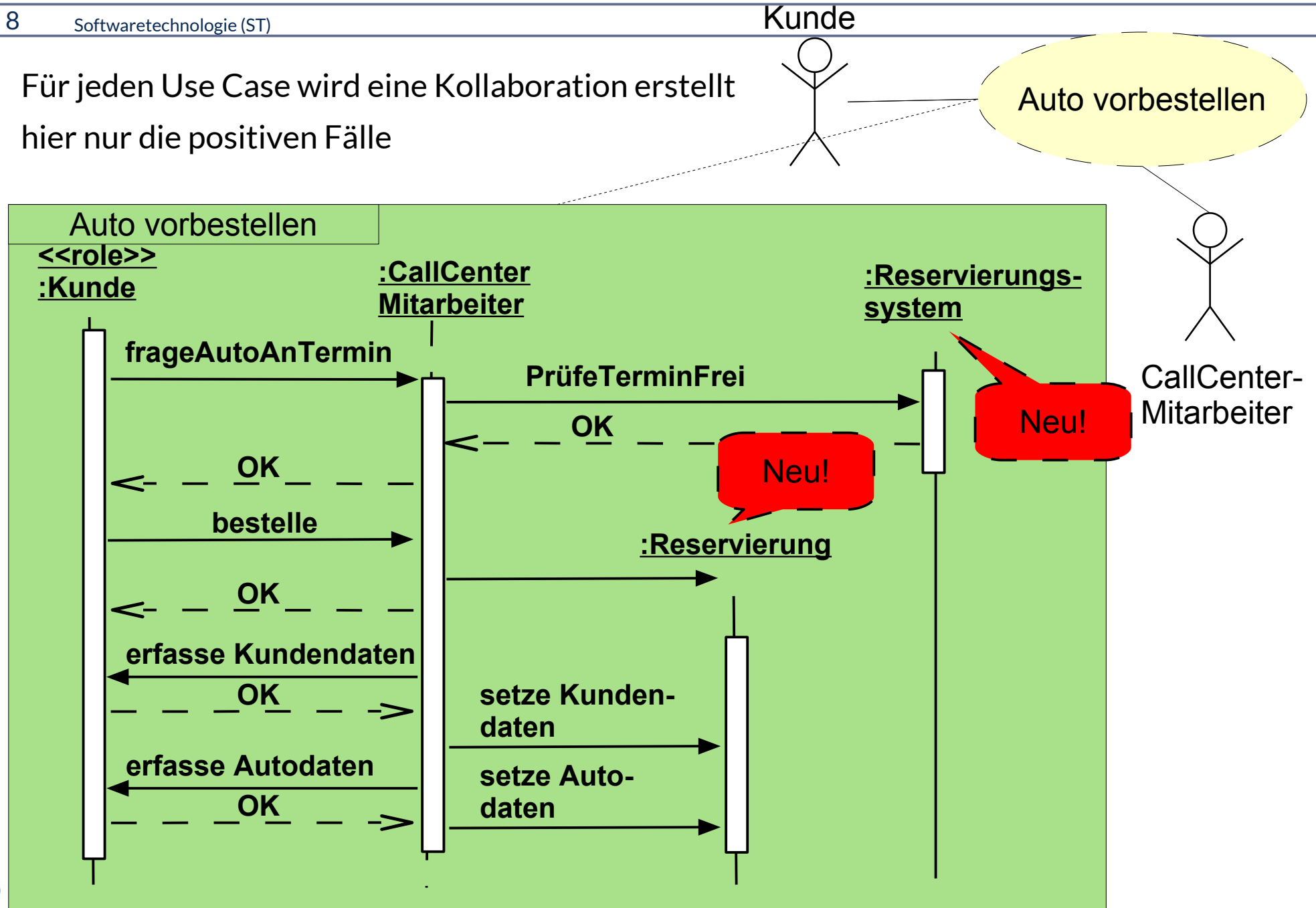
Softwaretechnologie (ST)

- ▶ Jeder Use Case führt zu einer querschnittlichen Verfeinerung der Analysemodelle
- ▶ Beim Ausfüllen (Elaboration) kommen neue Elemente hinzu



# Erste Szenarienanalyse mit Sequenzdiagramm "Auto vorbestellen"

- ▶ Für jeden Use Case wird eine Kollaboration erstellt
- ▶ hier nur die positiven Fälle



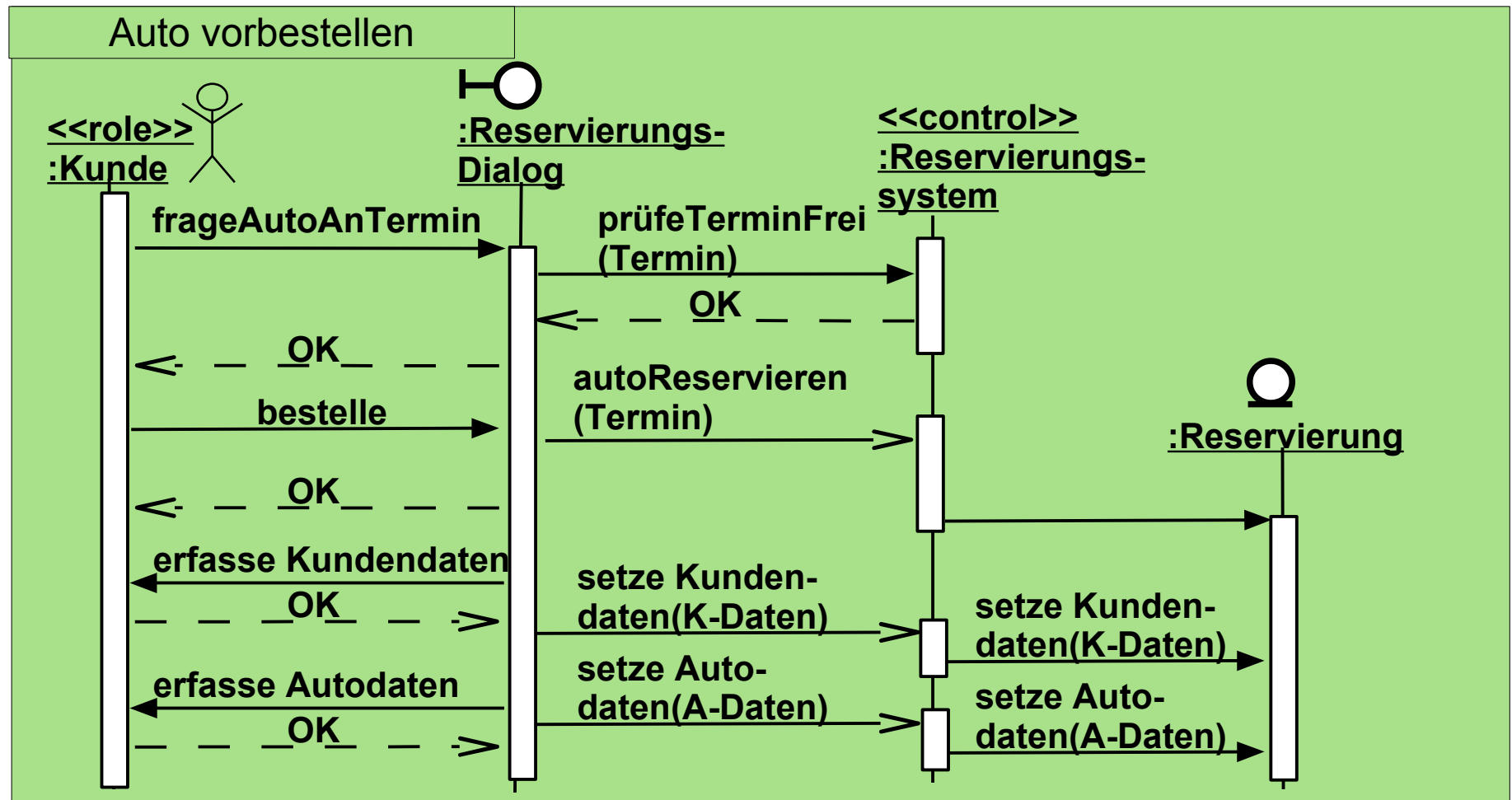


# BCD-Einteilung für Erste Szenarienanalyse "Auto vorbestellen"

9

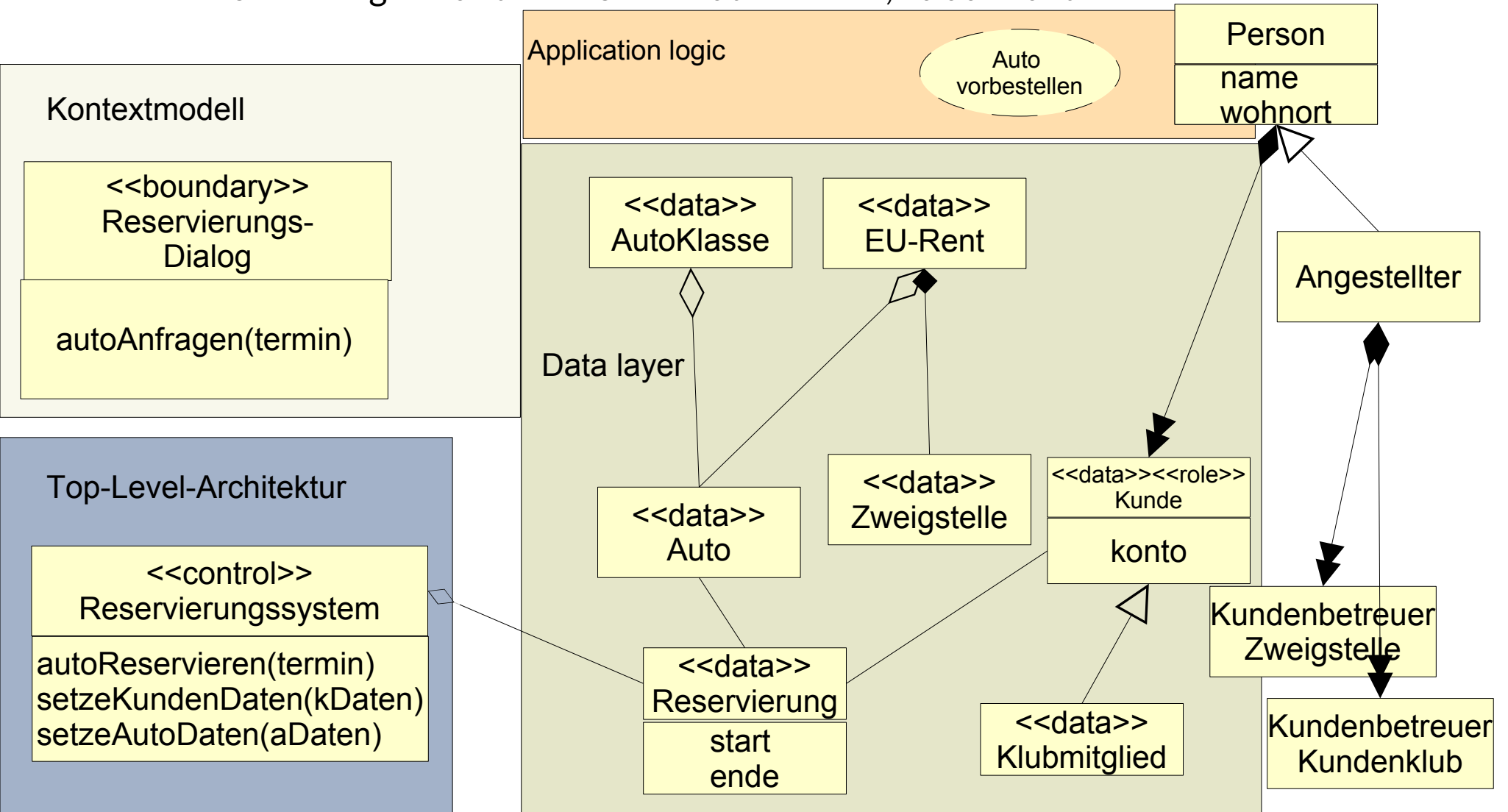
Softwaretechnologie (ST)

- ▶ Erstellen einer Kollaboration
- ▶ Identifizieren von Konnektorobjekten, die Kollaboration managen
- ▶ Einziehen einer Boundary-Klasse, Vergabe von BCD-Stereotypen
- ▶ Umschichten der Kommunikation der Daten-Klasse

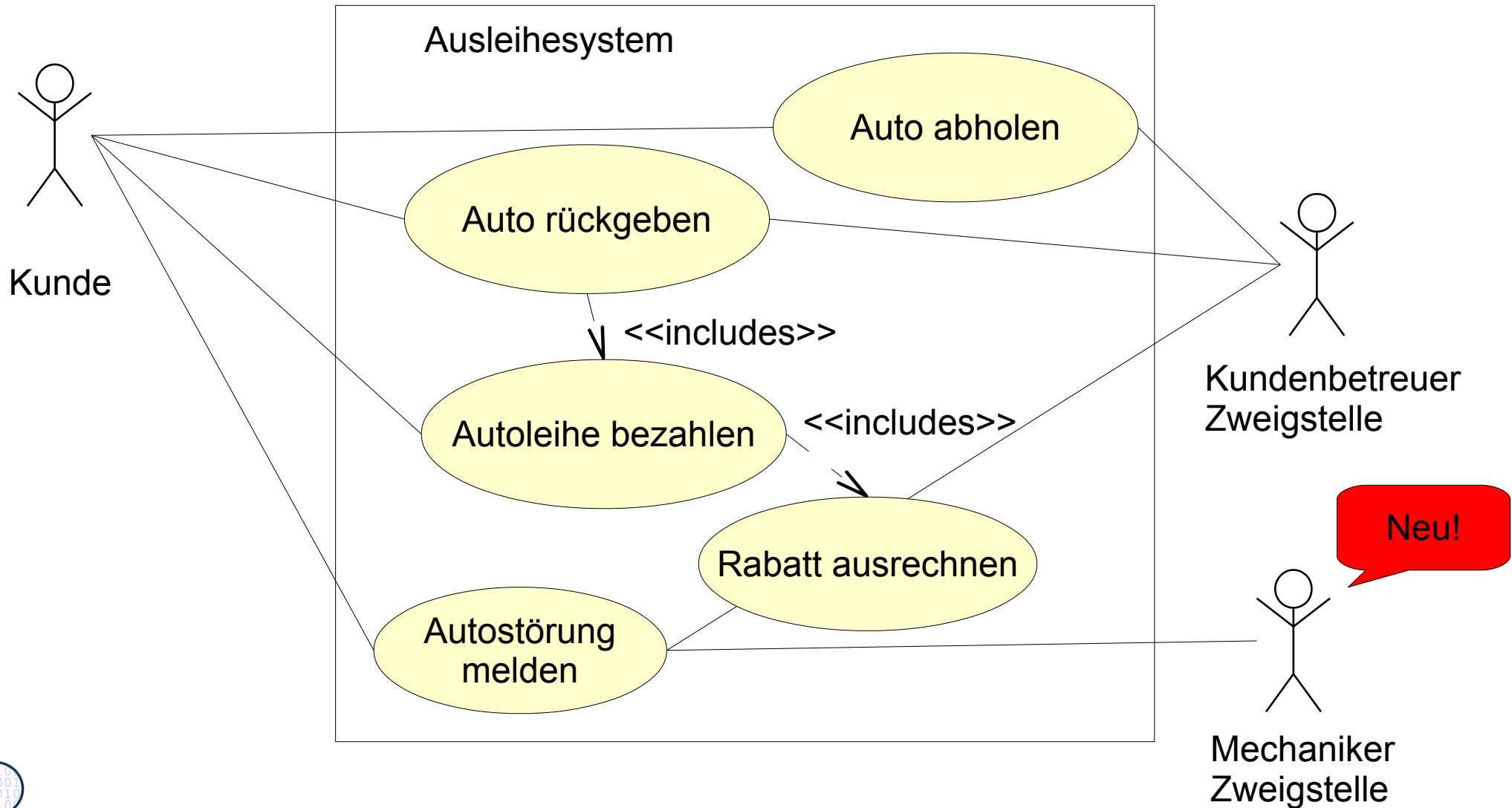


# 36.2: 2. Runde Domänenmodell

- ▶ Mehr Attribute, mehr Klassen. Ergibt erste Teile des Kontextmodells, der Top-Level-Architektur. des Datenmodells
- ▶ Personen liegen nur teilweise im Datenmodell, da auch aktiv

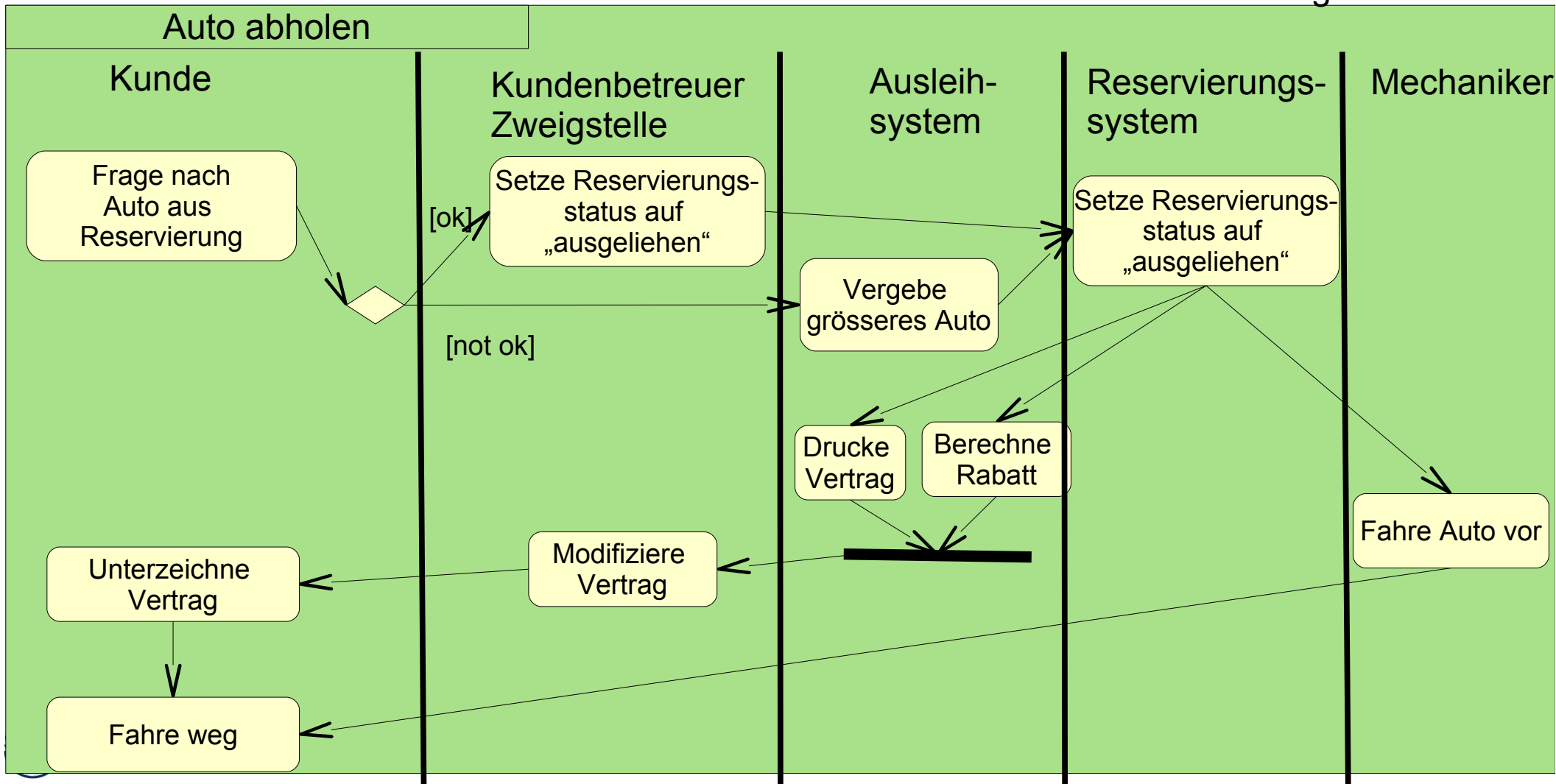
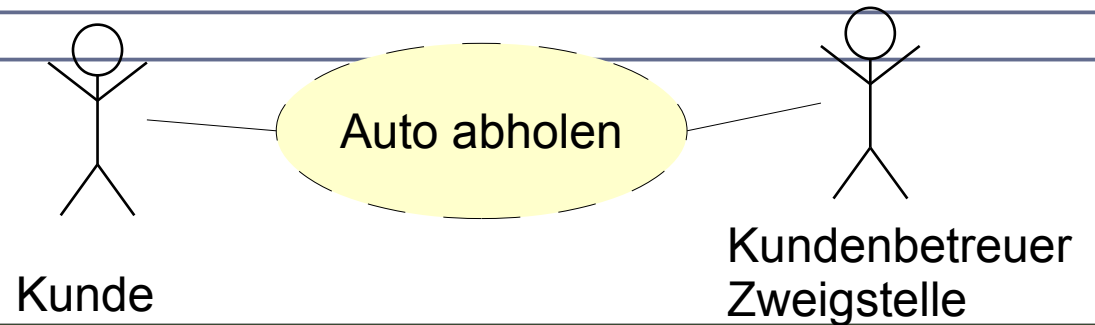


# B) Ausleihsystem von EU-Rent



# 2. Szenarienanalyse mit Aktivitätendiagramm: Auto ausleihen

- ▶ Rabatte werden vom Mitarbeiter persönlich bestimmt



# BCD für 2. Szenarienanalyse: Auto ausleihen, Methoden einführen

## Auto abholen

<<actor>>  
Kunde

<<boundary>>  
Ausleihdialog

<<control>>  
Ausleih-  
system

<<control>>  
Reservierungs-  
system

<<control>>  
Mechaniker

Frage nach  
Auto aus  
Reservierung

[ok]

rentCar(reservation)

[not ok]

CarClass.  
adjust()

Reservierungs-  
system.  
setStatus(  
„ausgeliehen“)

Contract.  
compute()

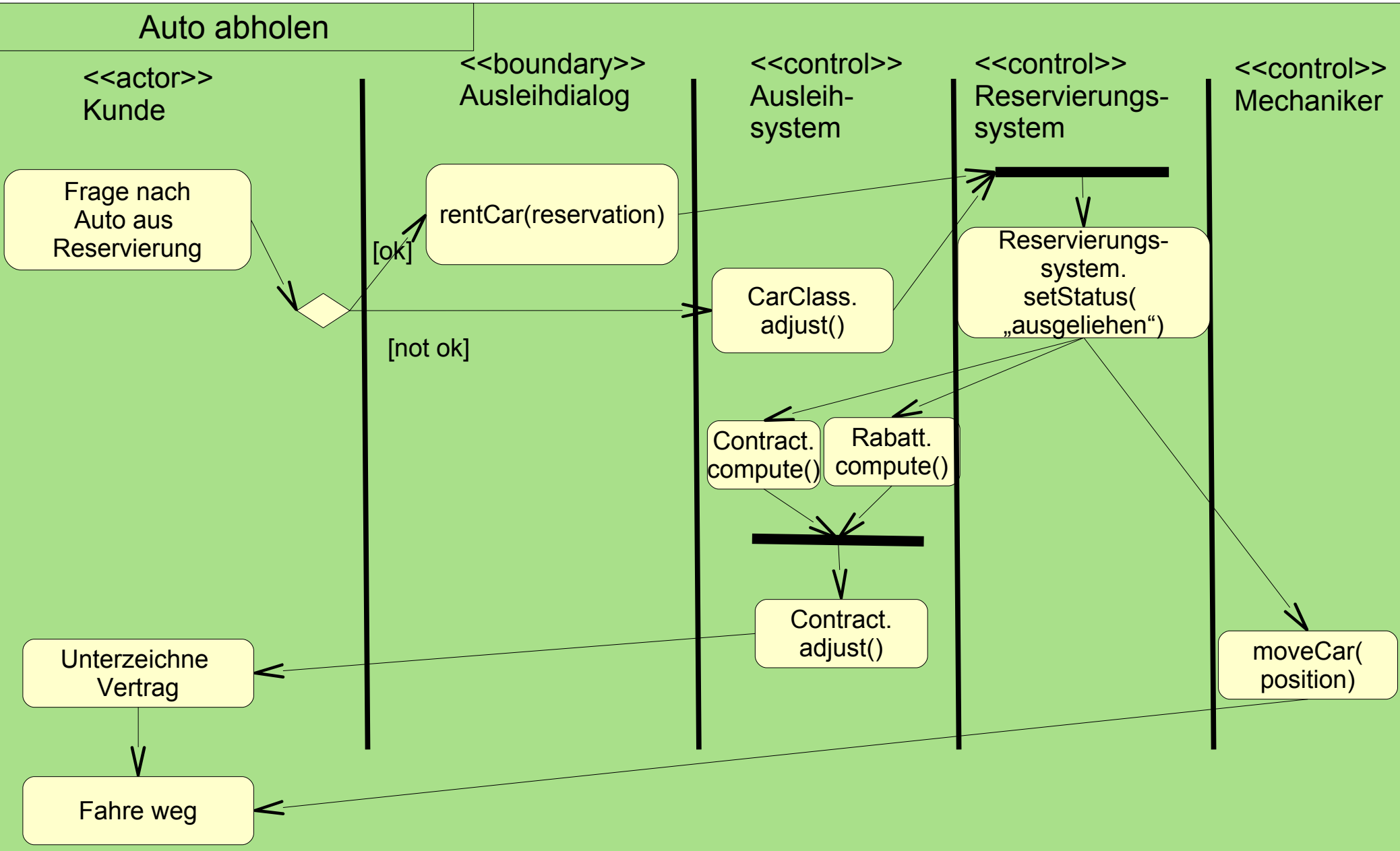
Rabatt.  
compute()

Contract.  
adjust()

Unterzeichne  
Vertrag

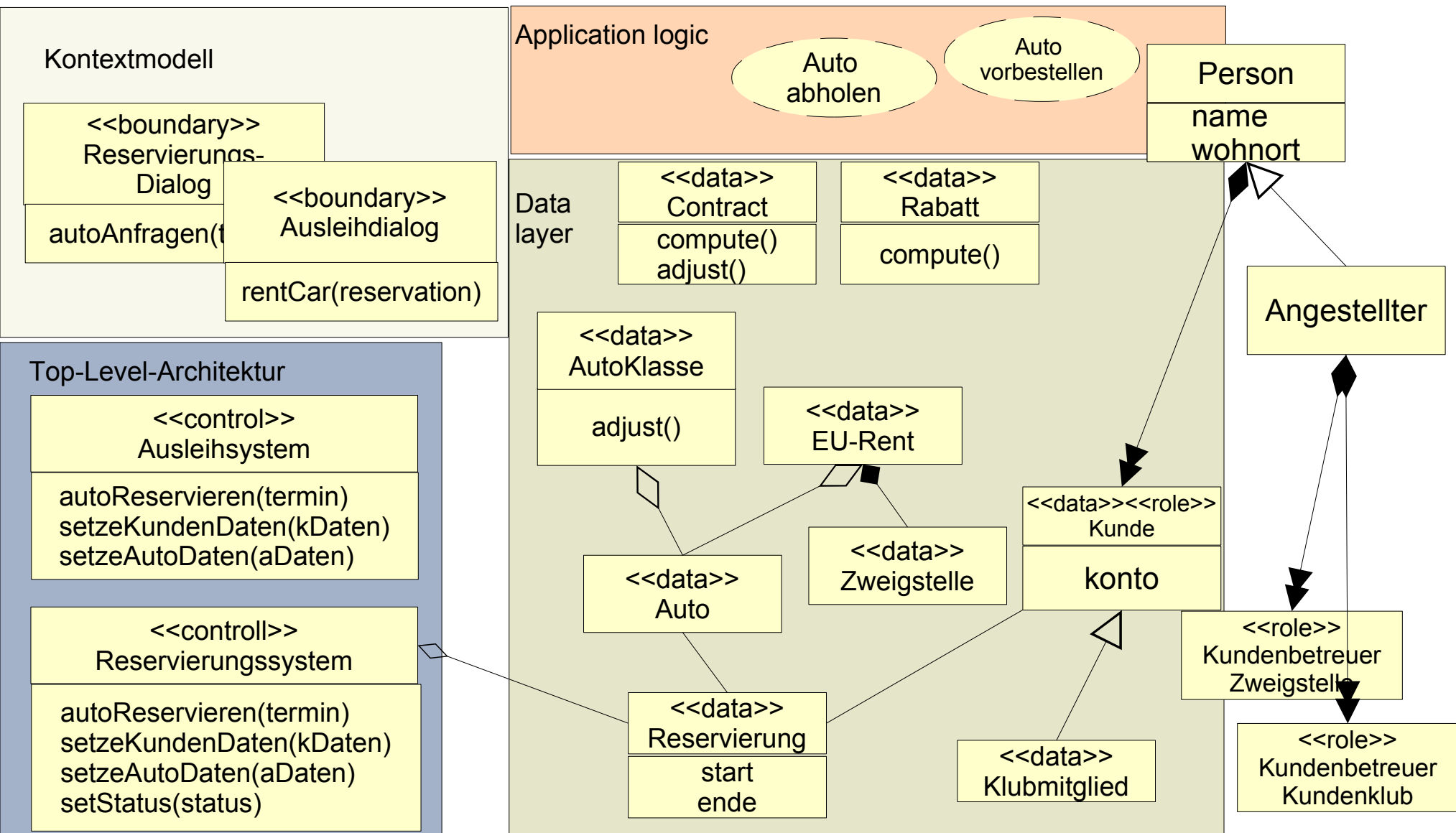
moveCar(  
position)

Fahre weg

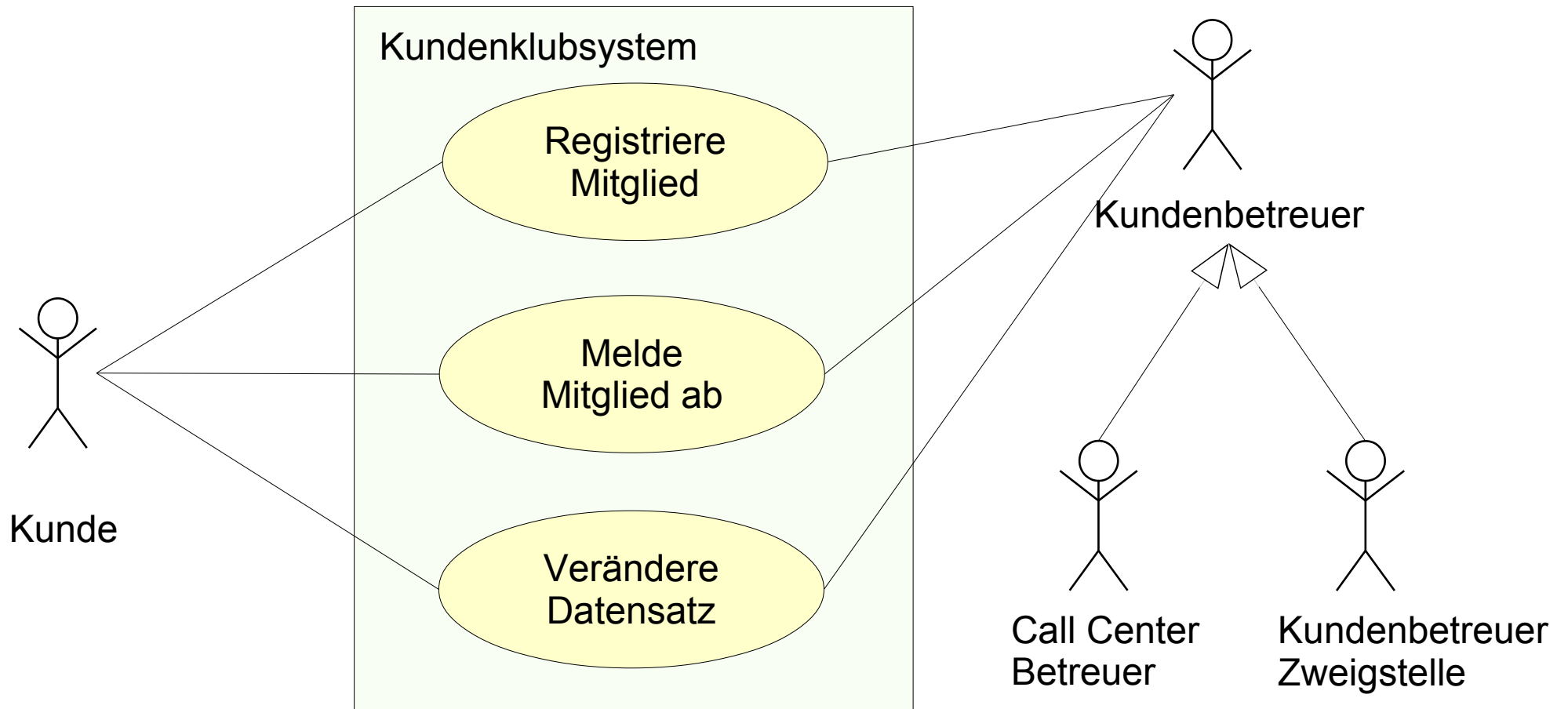


# 36.3: 3. Runde Domänenmodell, Kontextmodell, Toplevel-Architektur

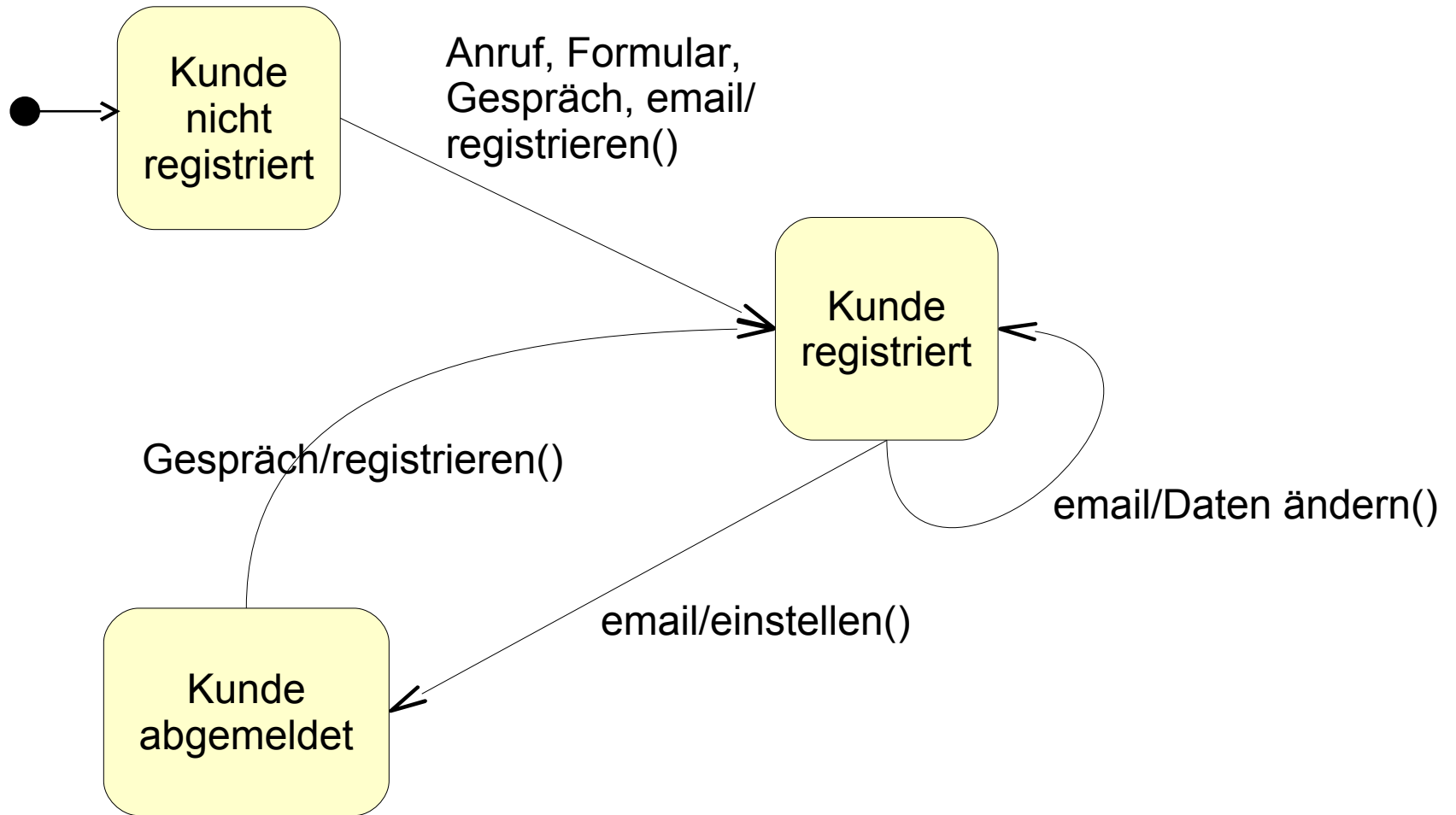
► Mehr Details



# C) Kundenclubprogramm von EU-Rent



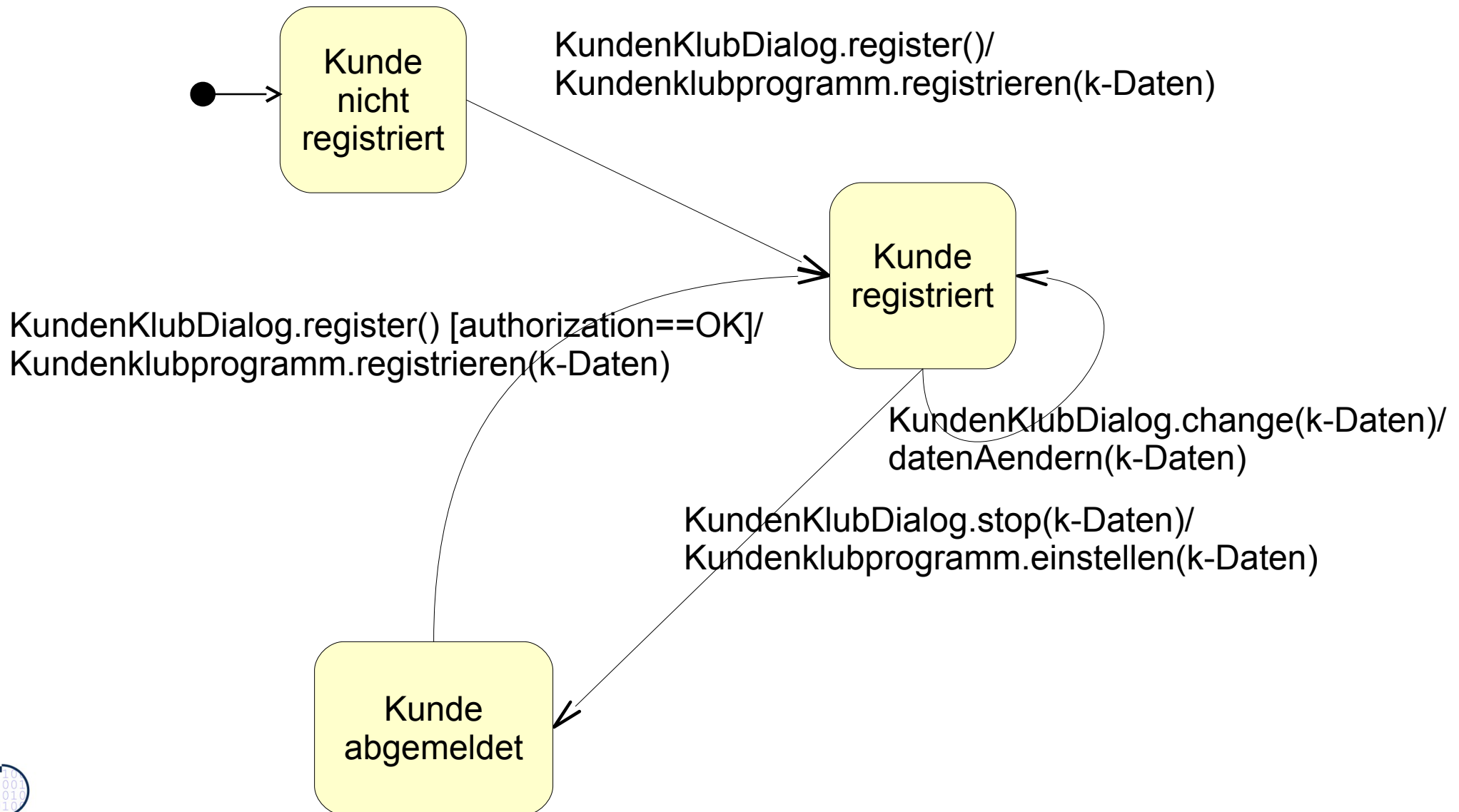
# 36.4. Szenarienanalyse mit Statecharts: Kunde als Klubmitglied verwalten



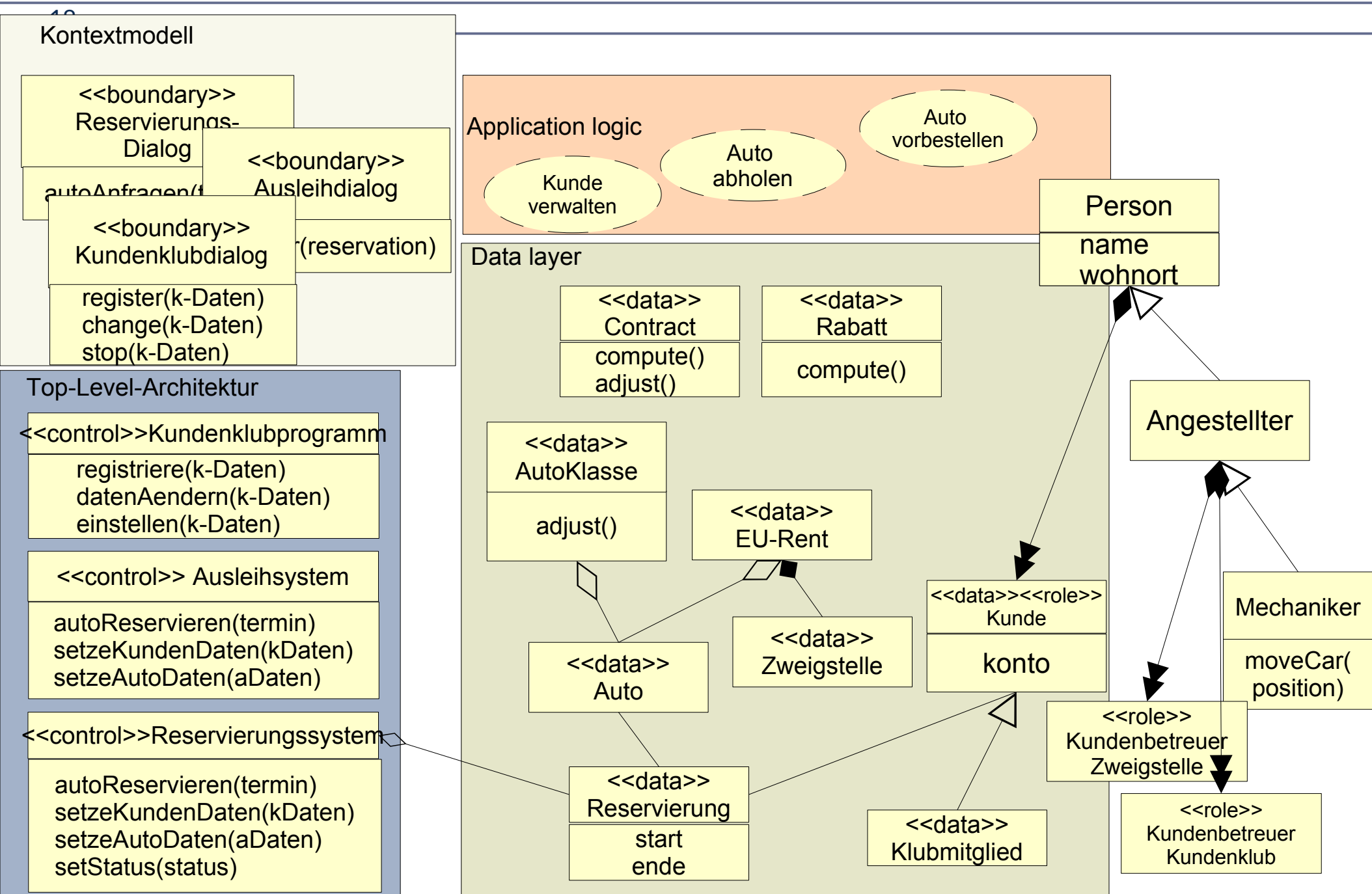


# Szenarienanalyse: Statecharts verfeinern mit Methoden

- ▶ Verfeinerung mit Methoden einer Verhaltensmaschine für das Kundenklubprogramm



# Resultat: Domänenmodell, Kontextmodell, Top-Level-Architektur



# Was haben wir gelernt?

- ▶ Szenarienanalyse hilft uns, aus Anforderungsfalldiagrammen
  - Domänenmodelle zu finden
  - Kollaborationen zu finden
  - Die Klassen des Domänenmodells nach BCD einzuteilen (hauptsächlich: boundary, control, data)
  - Weitere Modelle abzuleiten
    - Kontextmodell
    - Top-Level-Architektur (als Teil der Kontrollschicht)
    - Datenschicht
- ▶ Je nach Situation verwendet man Sequenzdiagramme, Aktivitätsdiagramme, Kommunikationsdiagramme, oder Zustandsdiagramme
- ▶ Man kann die Szenarien auch in Kollaborationen kapseln, um sie wieder zu verwenden
- ▶ Merke: In der Analyse sind noch viele Details ausgeklammert

# The End