

# Systemmodellierung mit SysML

## Inhalt

- **Grundlagen:**

- Aggregation & Komposition

- **Kompositionsstrukturdiagramm:**

- Motivation
- Rolle
- Konnektor

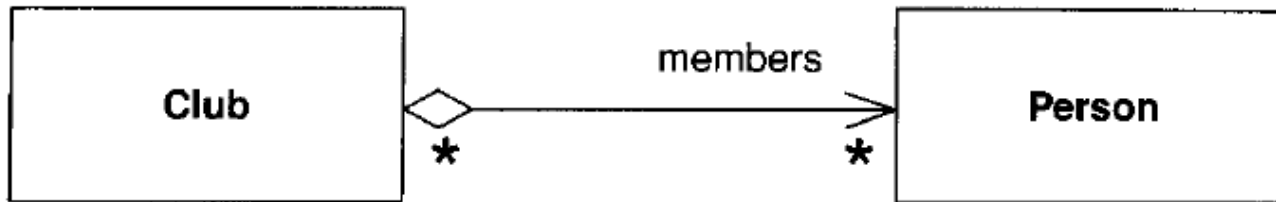
- **Erweiterte Konzepte:**

- Schnittstelle
- Signal
- Port

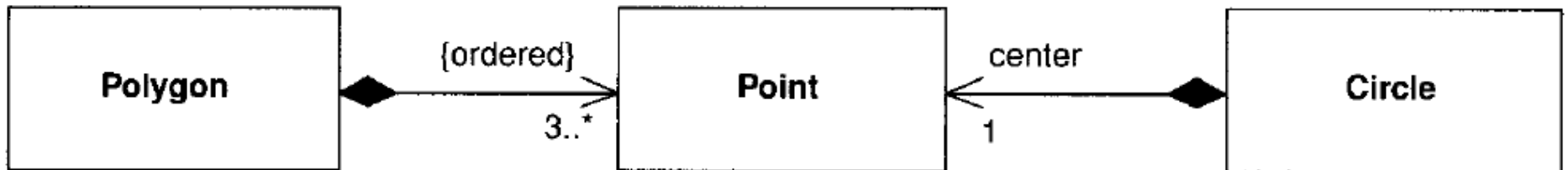
Oliver Stadie

# Aggregation & Komposition (1)

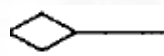

**Aggregation:**



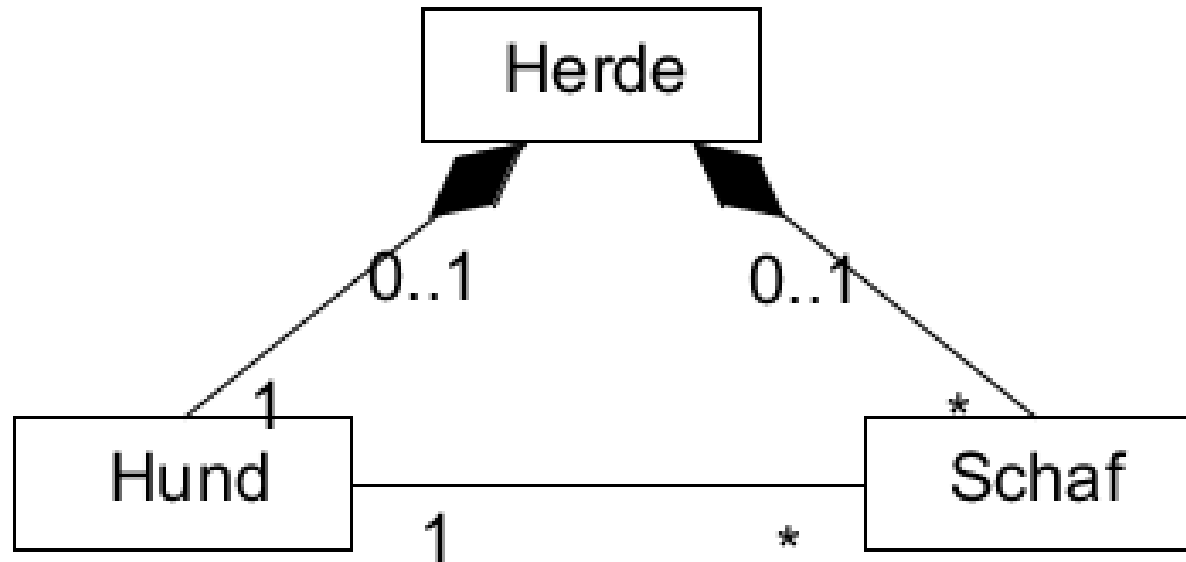
**Komposition:**



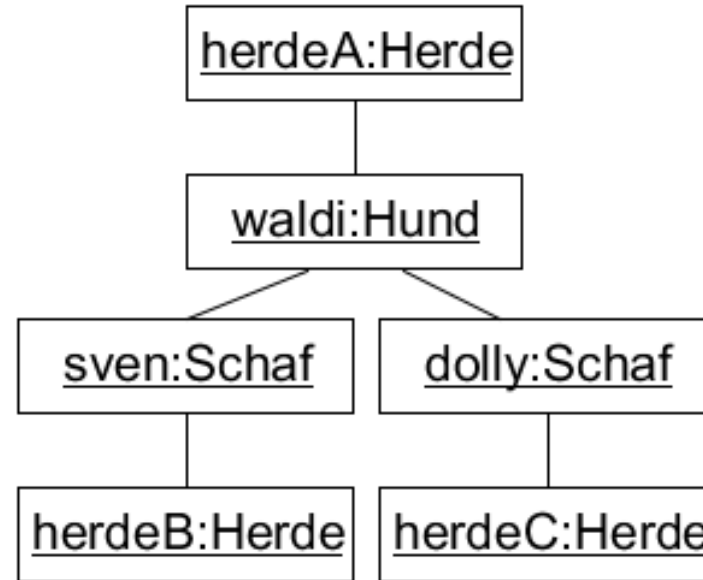
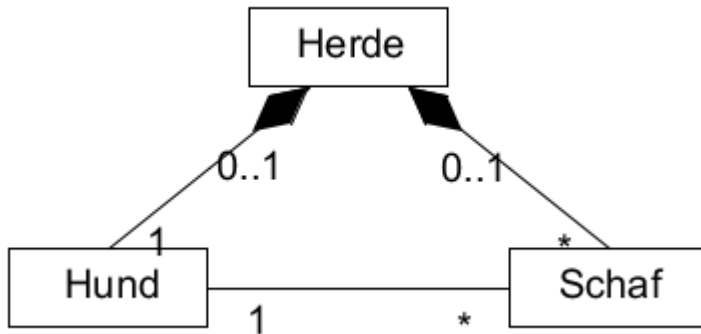
# Aggregation & Komposition (2)

- **Beide** beschreiben eine Teil-Ganzes-Beziehung
  - Raute auf der Seite des Ganzen
- **Aggregation** (  ):
  - Wie Assoziation, mit besonderer Auszeichnung eines Endes als „Ganzes“
  - „Think of it as a modeling placebo“ - Rumbaugh
  - (Keine Zyklen erlaubt)
- **Komposition** (  ):
  - „no sharing“: Jedes Teil hat maximal ein Ganzes  
→ Multiplizität [0..1] oder [1]
  - Wenn Ganzes vernichtet wird, werden auch seine Teile vernichtet

# Modell einer Schafherde



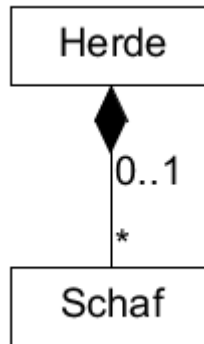
# Probleme bisheriger Methoden



Legale Objekt-Konstellation, aber nicht in unserem Sinne

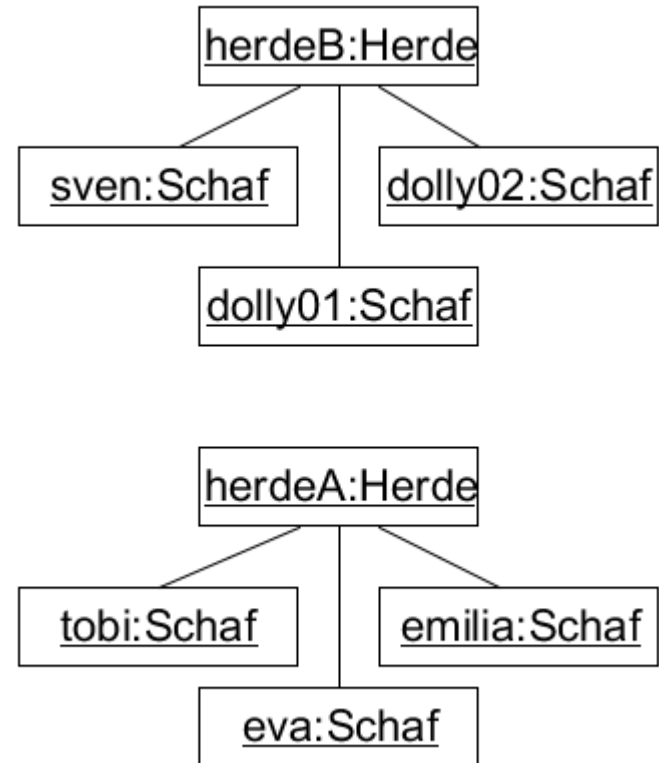
# Klassen & Objekte

Typ-Ebene  
mit Klassen



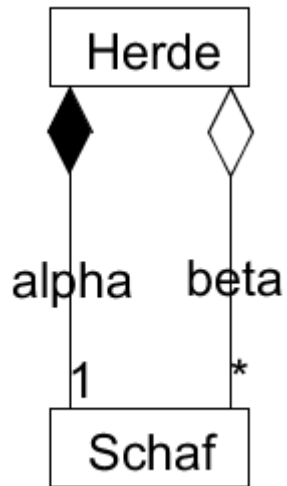
Klasse als Menge seiner Objekte

Instanz-Ebene  
mit Objekten

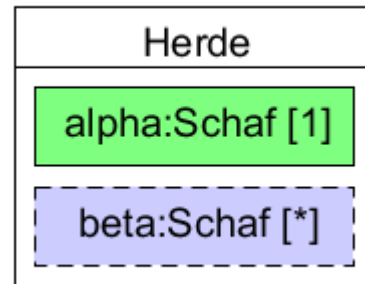


# Klassen, Rollen & Objekte

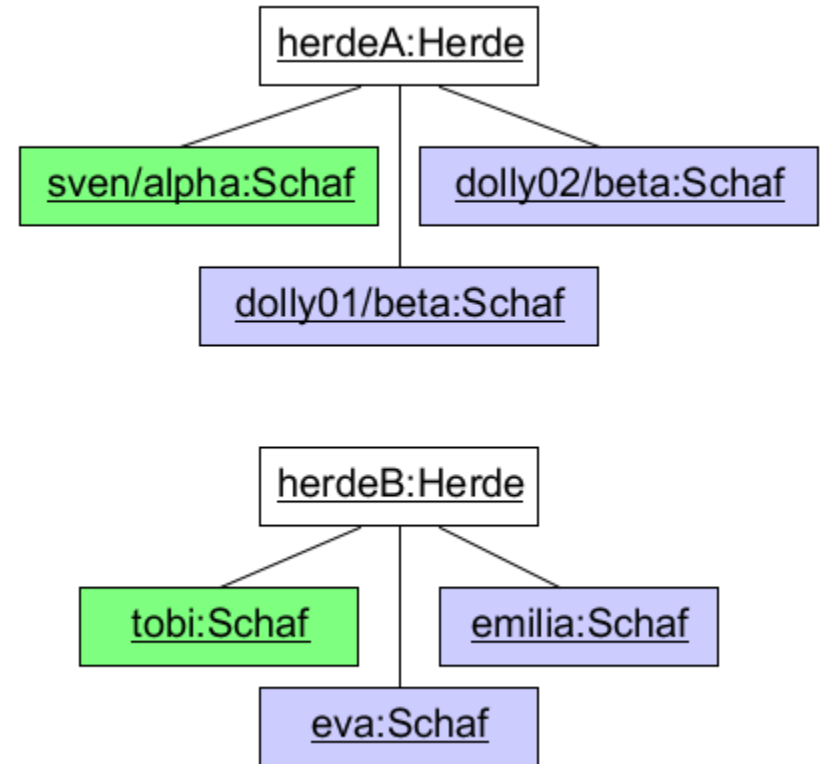
Typ-Ebene  
mit Klassen



Rollen-Ebene  
mit Rollen/Parts



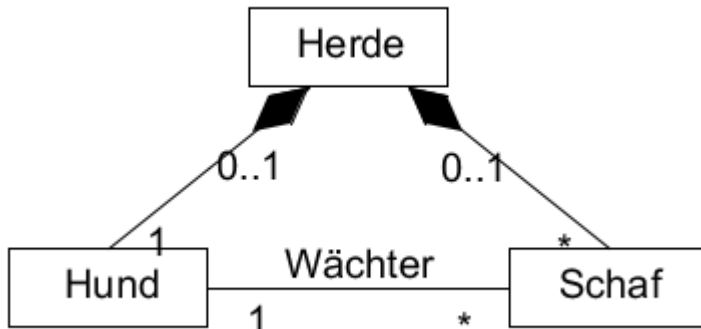
Instanz-Ebene  
mit Objekten



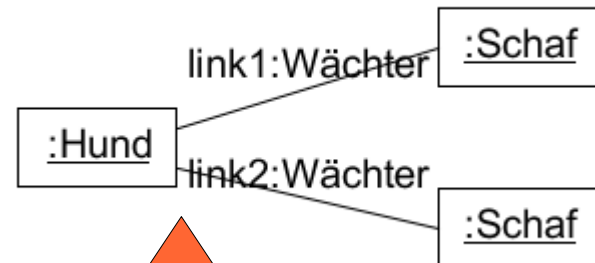
- Rolle als Teilmenge aller Objekte einer Klasse
- Parts sind im Kontext von Herde definiert (s.u.)

# Assoziationen & Links

Typ-Ebene mit  
Assoziationen



Instanz-Ebene  
mit Links



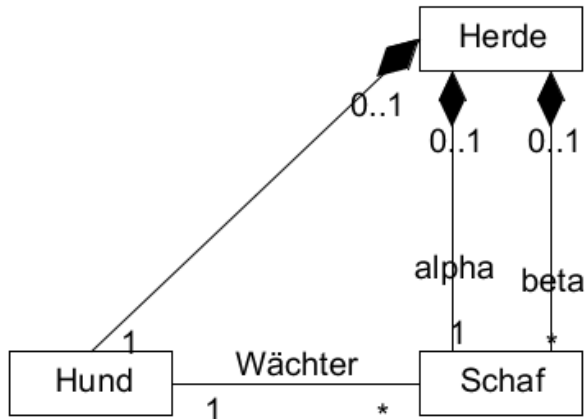
**!**  
**non-**  
**normative**

Assoziation als Menge seiner Links

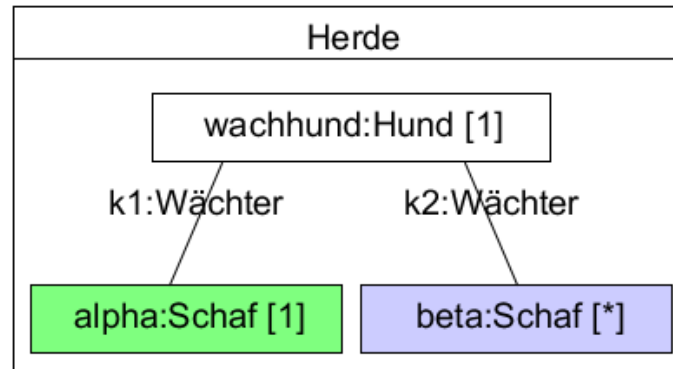


# Assoziationen, Konnektoren & Links

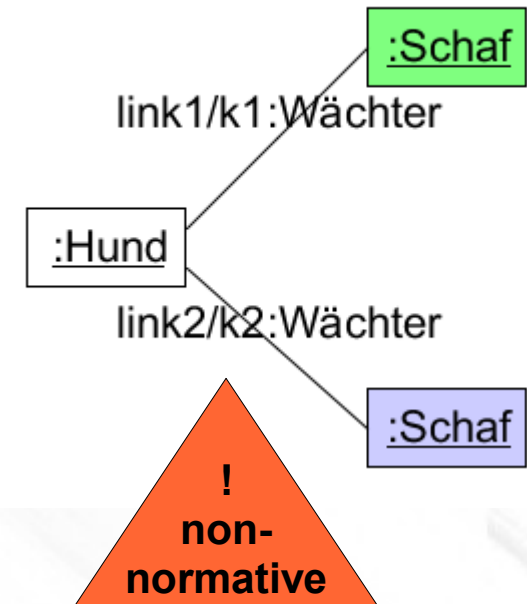
Typ-Ebene mit  
Assoziationen



Rollen-Ebene  
mit Konnektoren



Instanz-Ebene  
mit Links

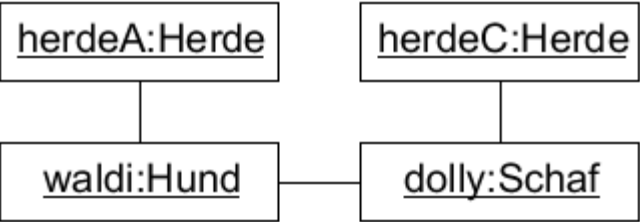


- Konnektor als Teilmenge aller Links einer Assoziation
- Konnektoren sind im Kontext von Herde definiert (s.u.)

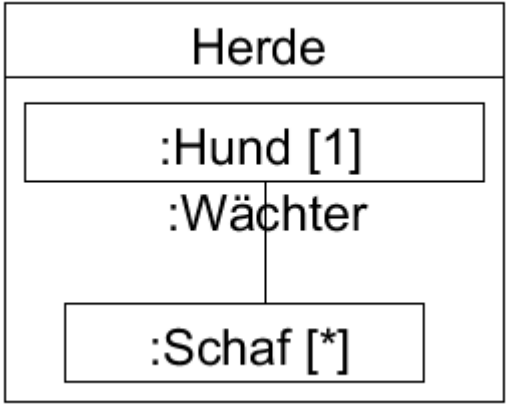
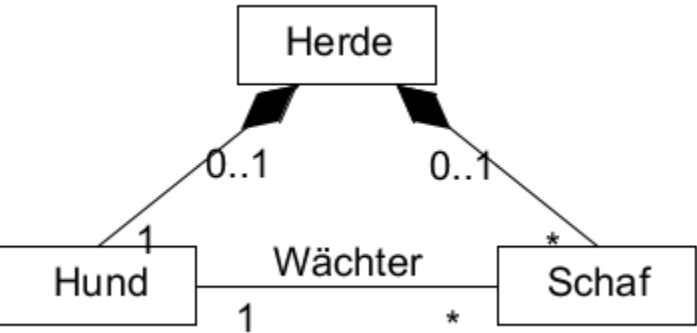
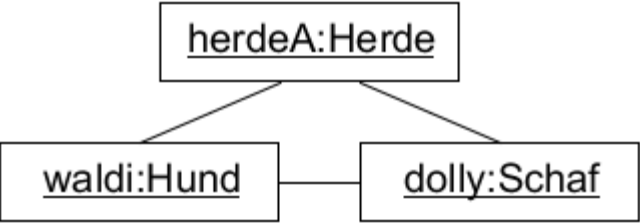
# Kompositionsstrukturdiagramm

Was bedeutet nun „im Kontext von Herde definiert“?

Nicht möglich, da auf eine Herde beschränkt

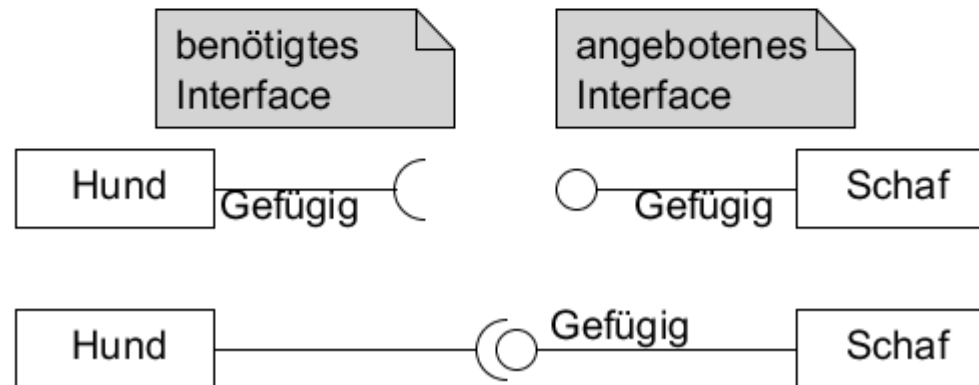
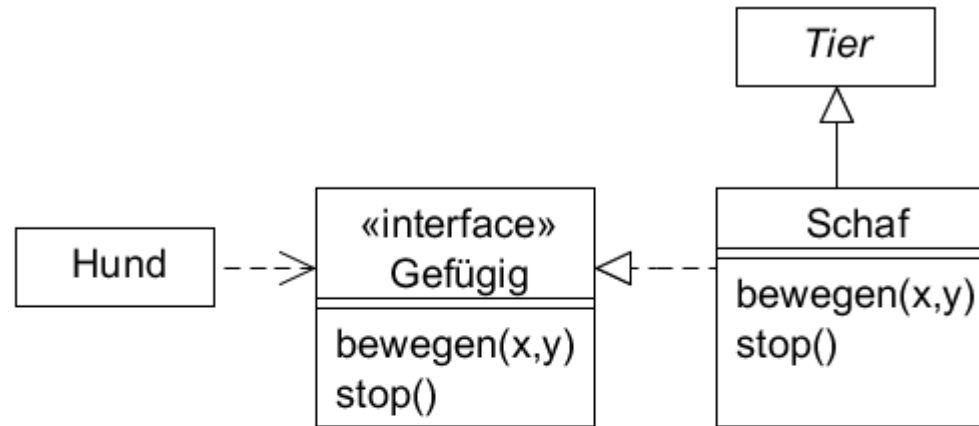


Möglich, da auf eine Herde beschränkt

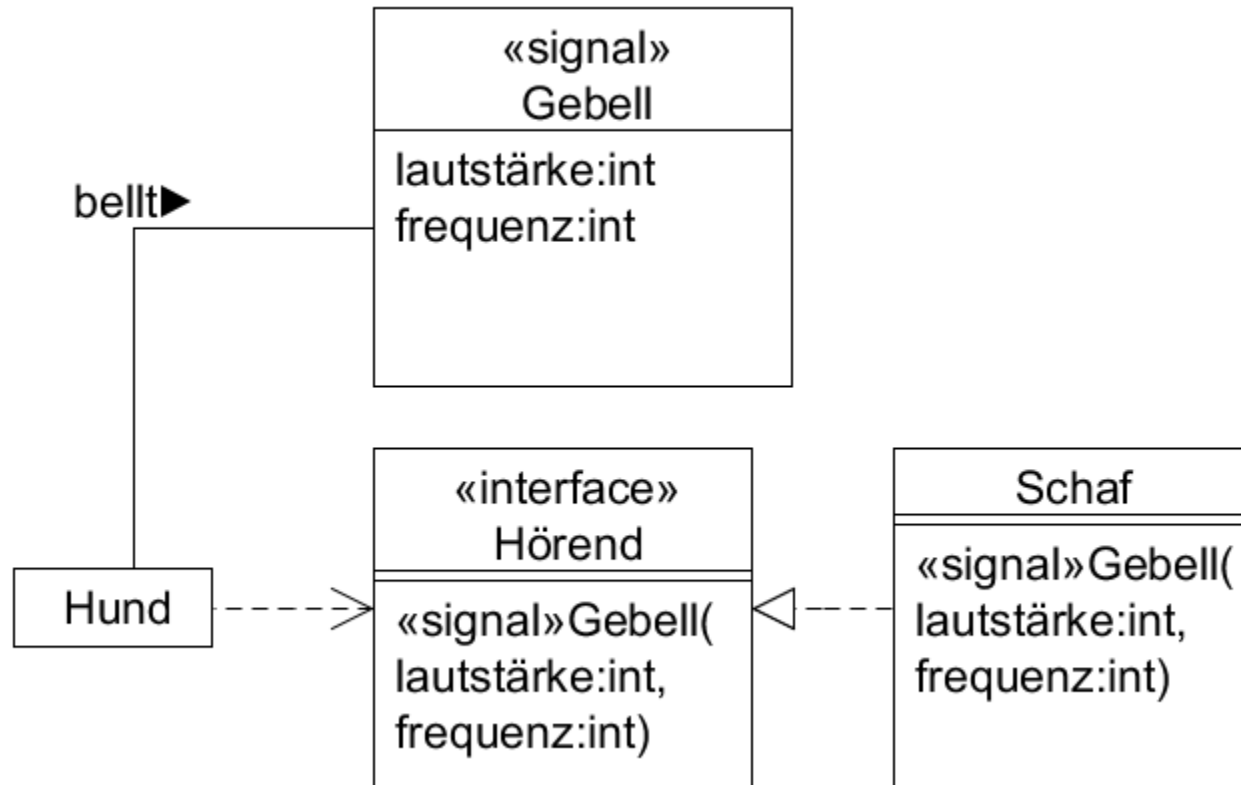


Die zwei Objekte am Ende des Links eines Konnektors müssen der gleichen Aggregat-Klasse angehören.

# Schnittstellen



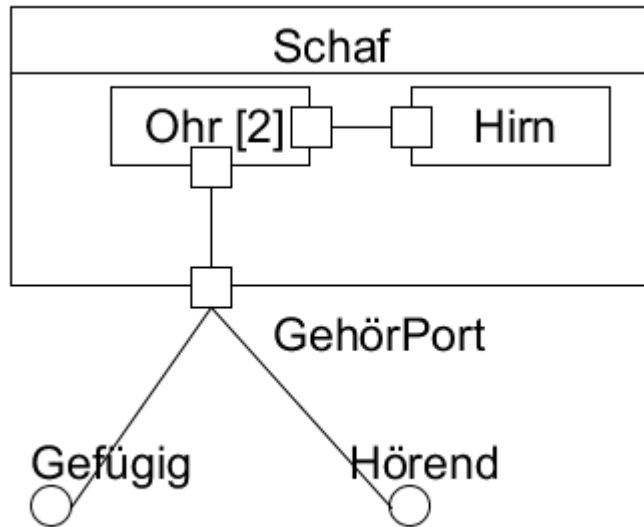
# Signale



# Signal vs. Methodenaufruf

- **Methodenaufruf:** Sender entscheidet, was Empfänger zu tun hat
  - z.B: Hund sagt dem Schaf, dass es sich nach (x,y) zu bewegen hat.
- **Signal:** Empfänger entscheidet, was mit dem Signal des Senders zu tun ist.
  - z.B: Hund bellt und Schaf entscheidet, ob es sich bewegt oder das Bellen ignoriert.

# Ports



- Bündeln und benennen Kommunikationsschnittstellen
- Definieren Kommunikationspunkte der Parts und Aggregatklasse
- Konnektoren, die Ports verbinden, bedeuten Kommunikation über die Port-Interfaces
- Konnektoren ohne Ports bedeuten direkte Kommunikation ohne Interface