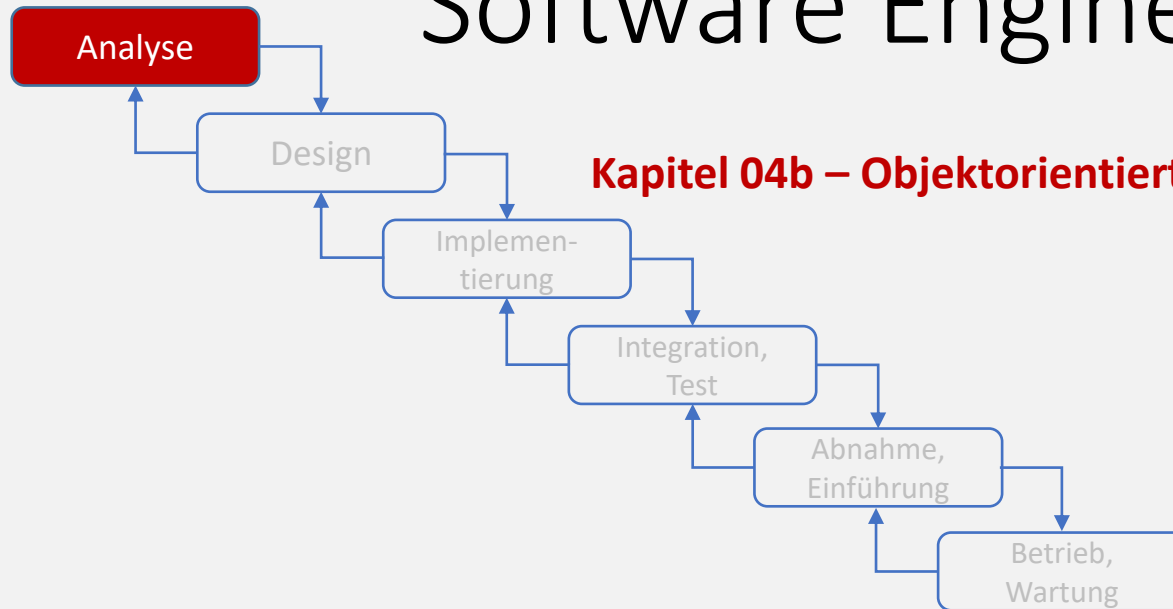




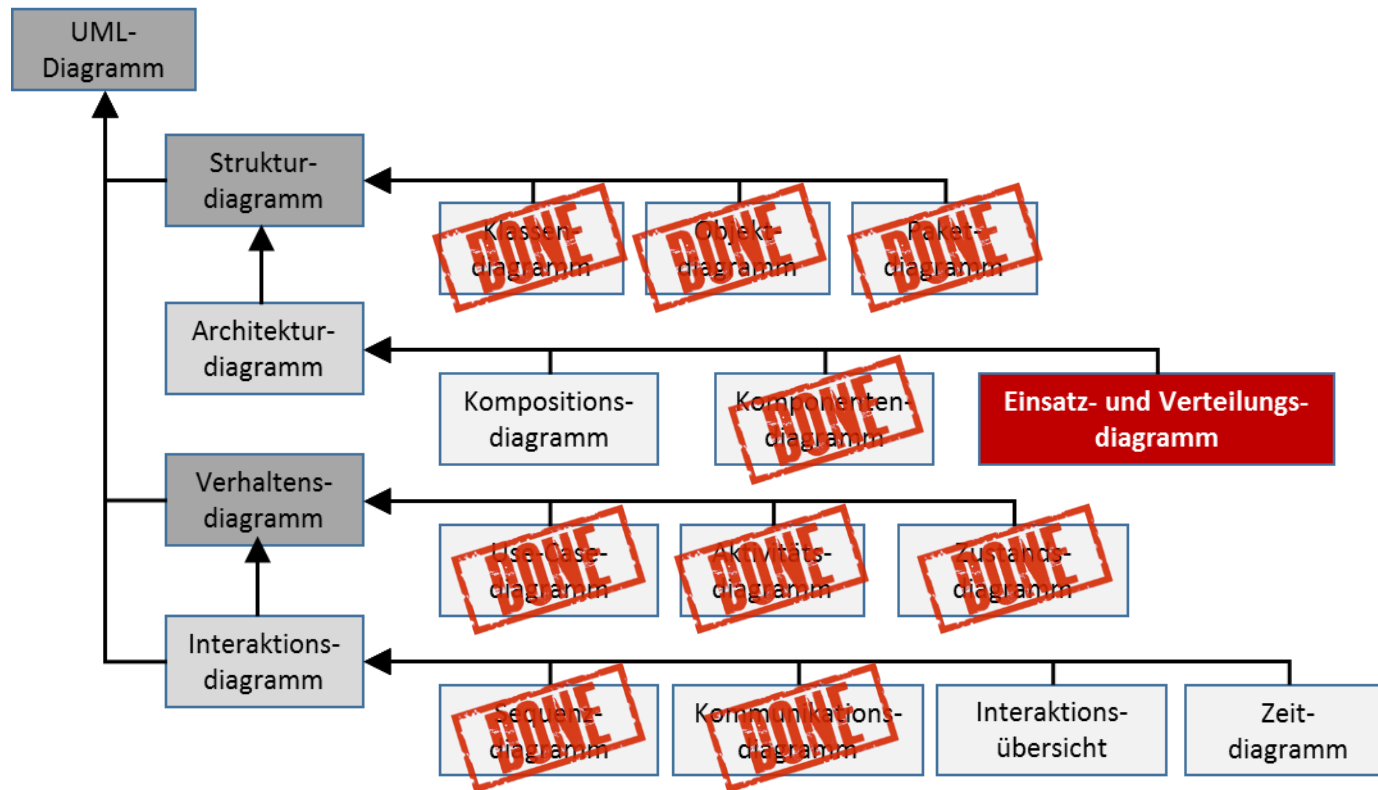
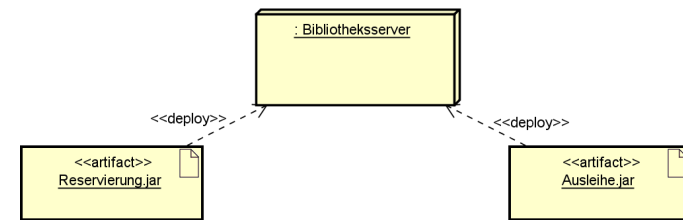
# Software Engineering

## Kapitel 04b – Objektorientierte Analyse (OOA)



Einsatz-/Verteilungsdiagramme

# UML: Verteilungsdiagramme

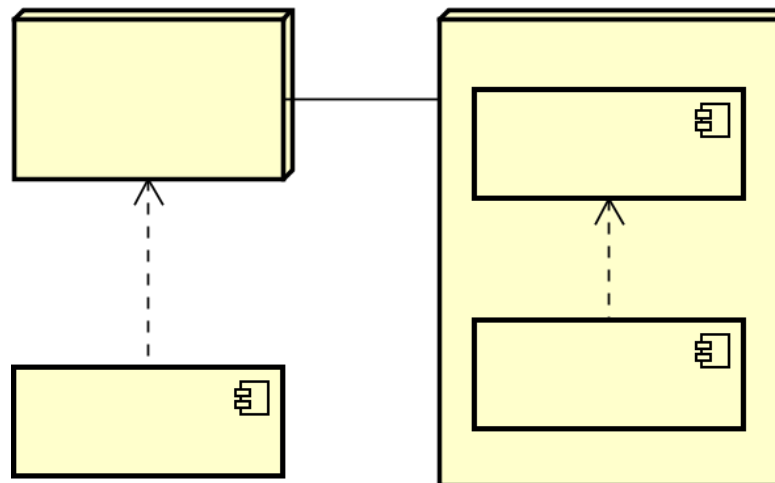


auch Einsatzdiagramm oder engl. Deployment Diagram genannt

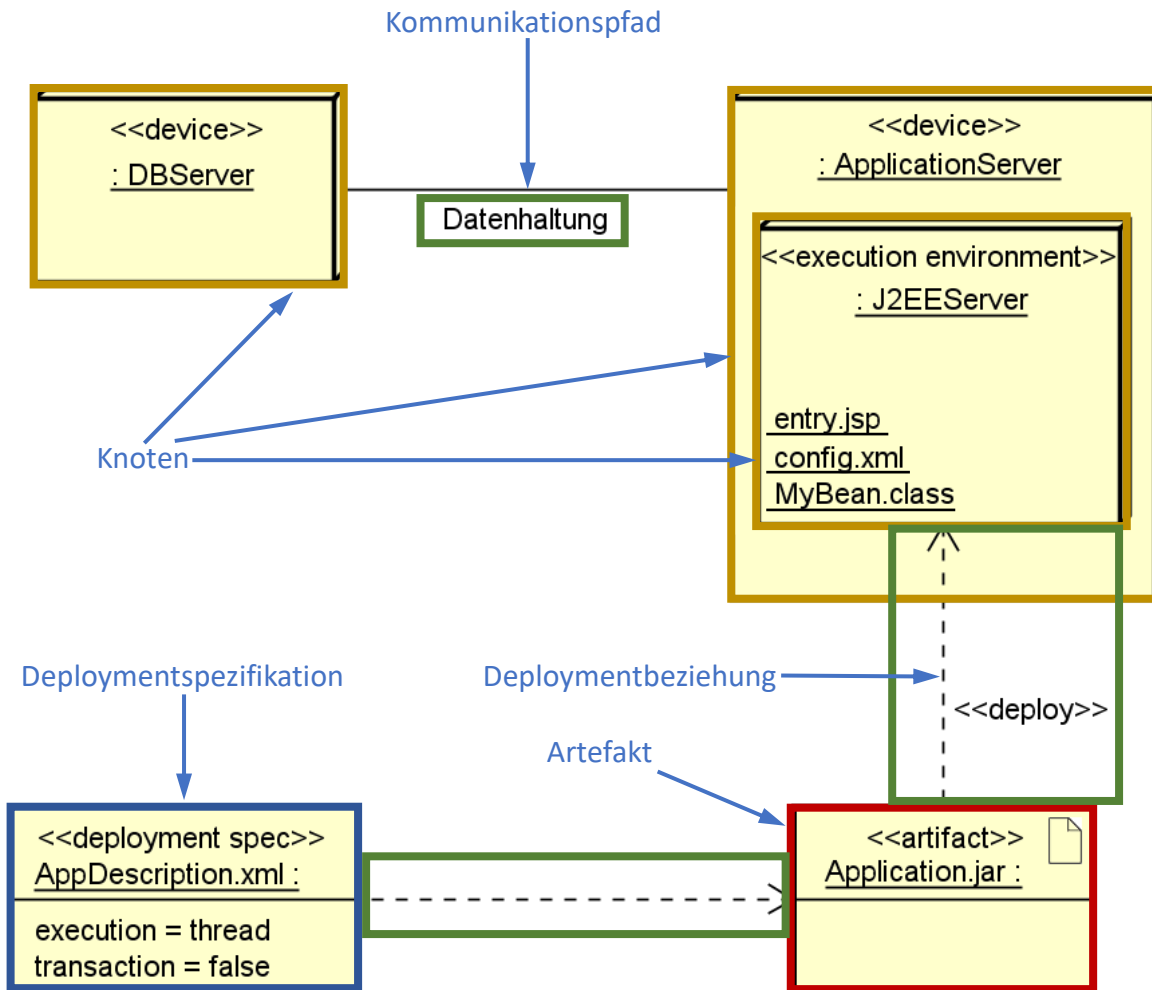
# Verteilungsdiagramm

## Wozu benötige ich das Verteilungsdiagramm?

- Es zeigt die Zuordnung von Artefakten (also u.a. Softwarekomponenten) auf Hardware-Einheiten, die als Knoten bezeichnet werden
- Stellt die Kommunikationsverbindungen und Abhängigkeiten zwischen Knoten dar
- **Es beantwortet die Frage:**  
*“Wie werden die Artefakte des Systems zur Laufzeit verteilt?”*  
(Verteilung: Installation, Konfiguration, Bereitstellung oder Ausführung)



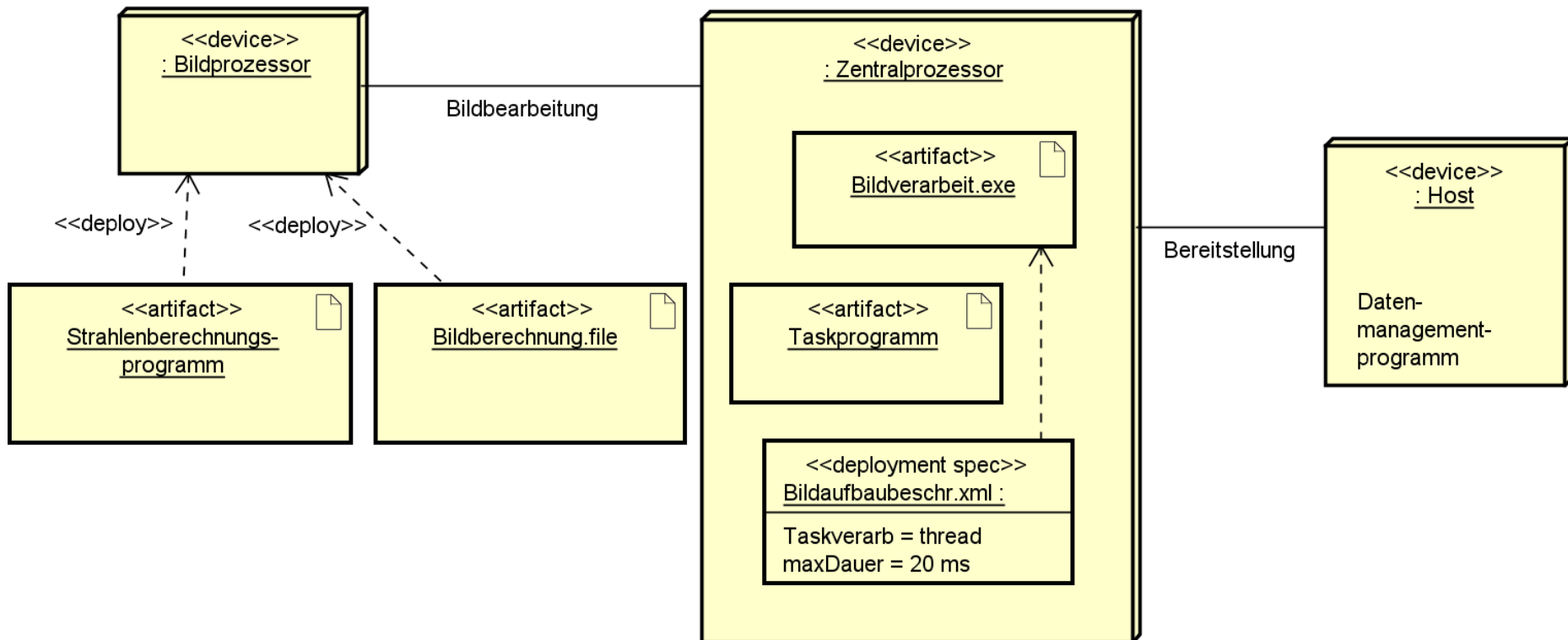
# Verteilungsdiagramm: Überblick



## Elemente:

- **Artefakte**
- **Knoten** (Rechnerressourcen wie z.B. Prozessoren)
- **Einsatzspezifikationen**
- **Beziehungen zwischen Knoten**

# Verteilungsdiagramm: Anwendungsbeispiel

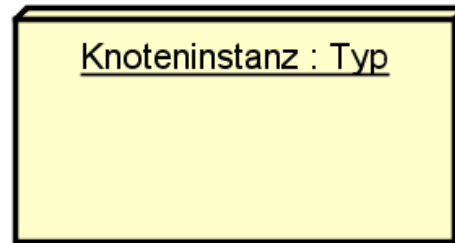
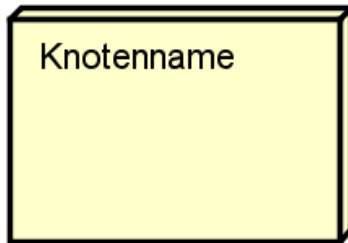


# Verteilungsdiagramme: Anwendung im Projekt

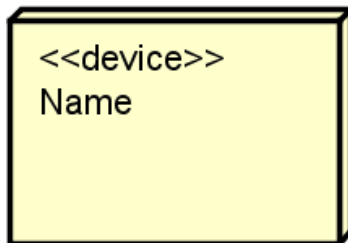
- Grundsätzlich: nur sinnvoll, wenn es auch wirklich etwas zu verteilen gibt
- Dann ist es an drei Stellen sinnvoll:
  - Systemkontext  
“Wie kommuniziert mein System mit Nachbarsystemen”
  - Design-Phase:  
“Wie sieht eine Verteilung auf Umgebungen im Prinzip aus?”
  - Übergabe in den Betrieb:  
“Wie sieht die konkrete Verteilung aus (Rechnernamen, IP-Adressen, Portnummern, etc.)”

# Verteilungsdiagramm: Elemente

- **Knoten und Knoteninstanz:**

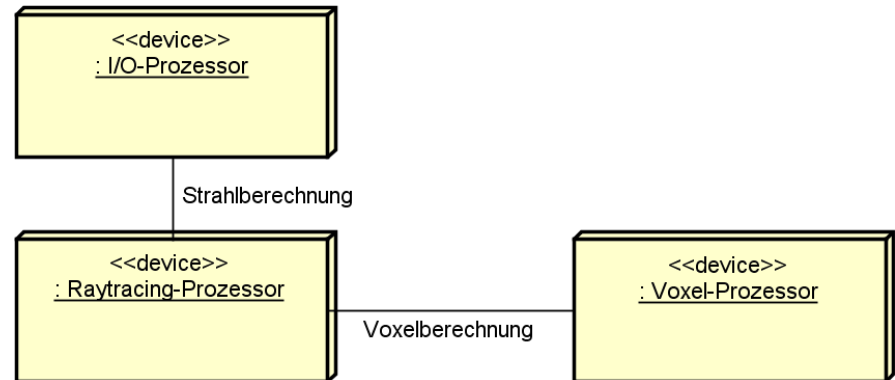
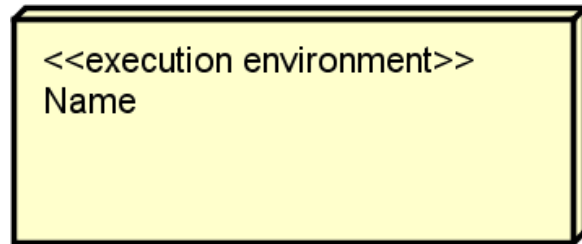


- **Gerät:**

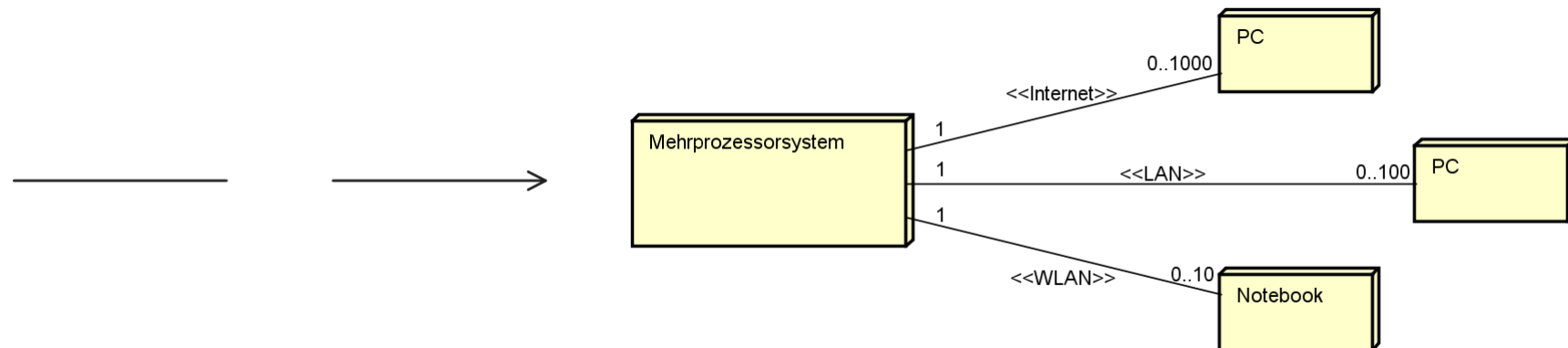


# Verteilungsdiagramm: Elemente

- **Ausführungsumgebung:**



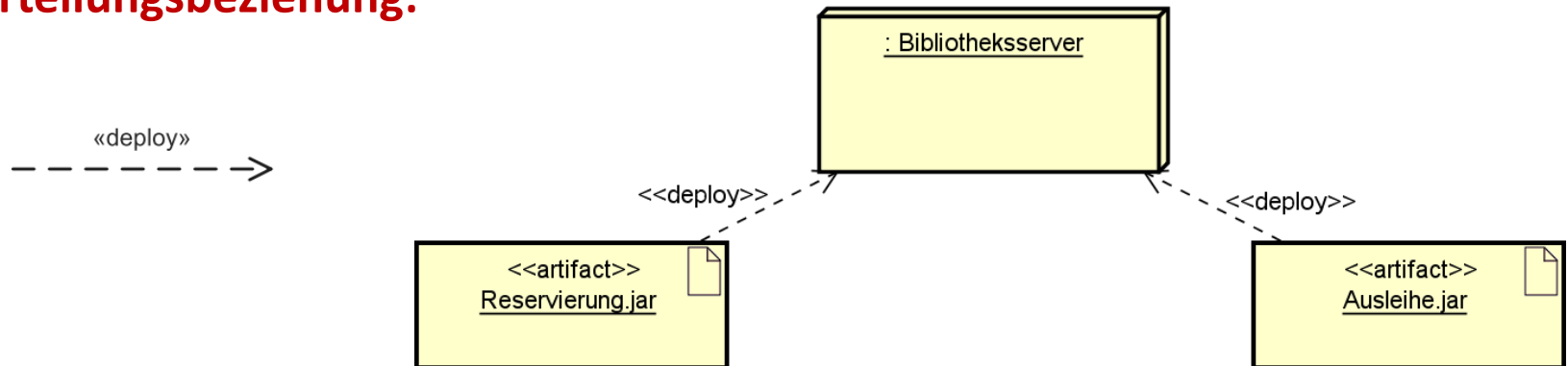
- **Kommunikationspfad (gerichtet und ungerichtet):**



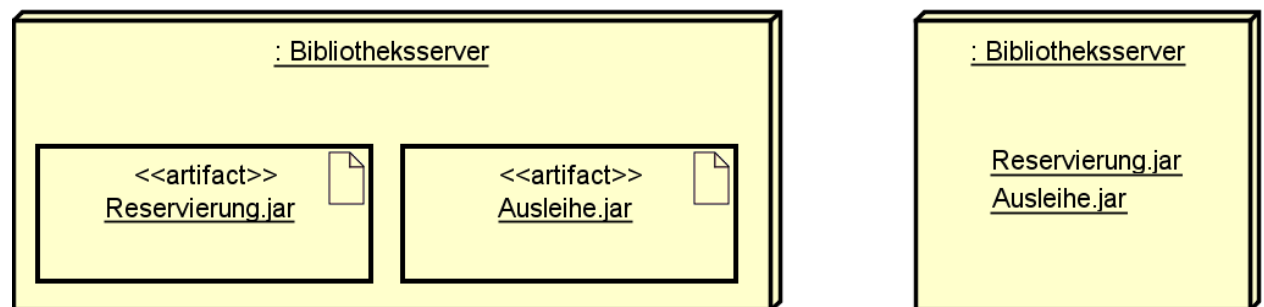


# Verteilungsdiagramm: Elemente

- **Verteilungsbeziehung:**

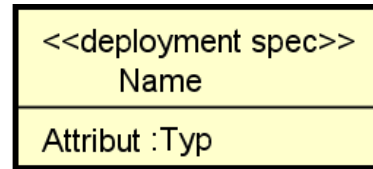
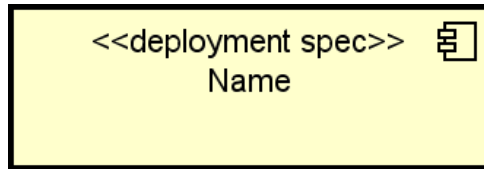


Zwei weitere Darstellungsmöglichkeiten:

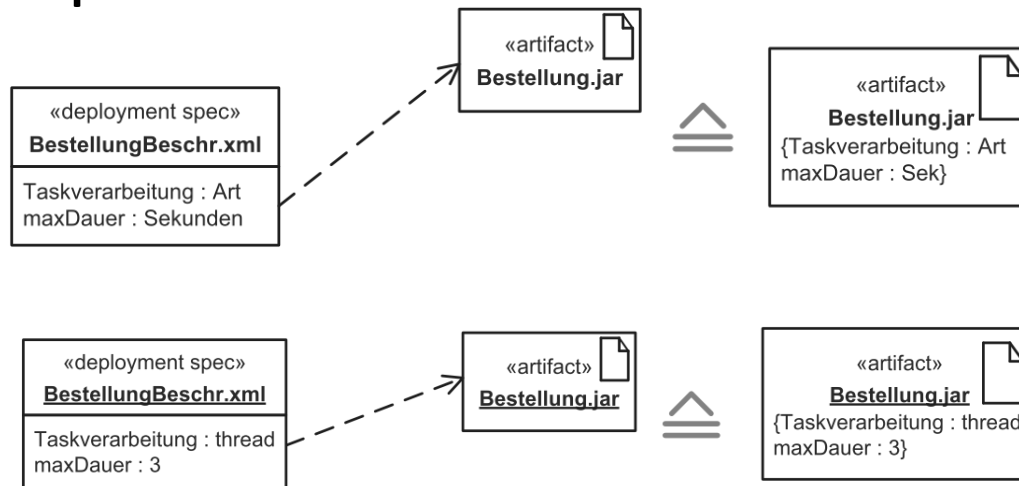


# Verteilungsdiagramm: Elemente

- **Einsatzspezifikation** (ohne und mit Eigenschaften):



## Beispiel:



# Verteilungsdiagramm: Beispiel



**Modellieren Sie die Zubereitung einer Tiefkühlpizza mithilfe eines einfachen Verteilungsdiagramms:**

# UML: Rückblick und Zusammenfassung

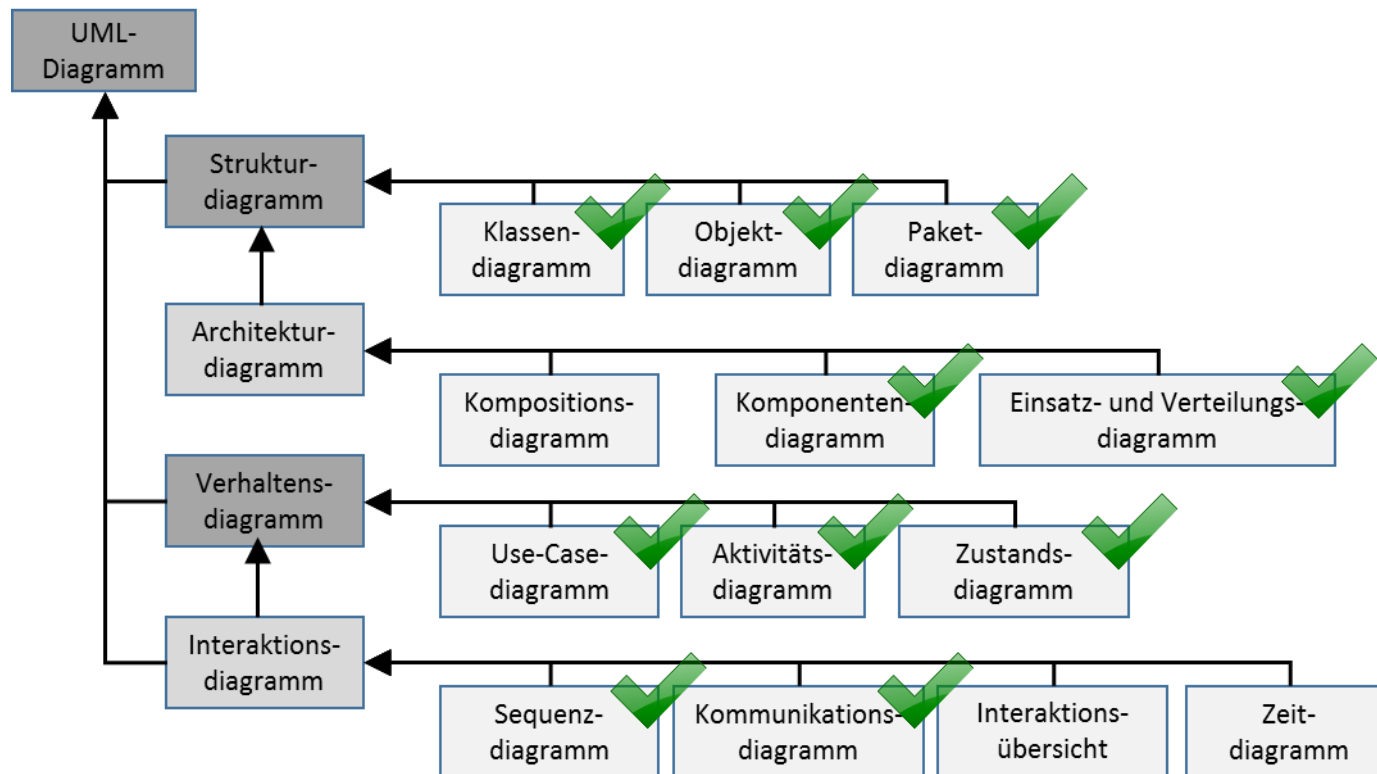
# UML: Rückblick und Zusammenfassung

## Basiskonzepte

---

- UML-Basiskonzepte:
  - Objekt
  - Klasse
  - Assoziation
  - Aggregation
  - Komposition
  - Attribut
  - Operation
  - Vererbung
  - Paket
  - ...

# UML: Rückblick und Zusammenfassung Diagramme



# UML: Rückblick und Zusammenfassung Diagramme

## Welche Fragen beantworten die UML-Diagramme?

- **Use-Case-Diagramm:**

“Was leistet mein System für seine Umwelt (Nachbarsysteme, Stakeholder)?”

- **Aktivitätsdiagramm:**

“Wie läuft ein bestimmter flussorientierter Prozess oder Algorithmus ab?”

- **Zustandsdiagramm:**

“Welche Zustände kann ein Objekt, eine Schnittstelle, ein Use Case, ...bei welchen Ereignissen annehmen?”

# UML: Rückblick und Zusammenfassung Diagramme

## Welche Fragen beantworten die UML-Diagramme?

- **Klassendiagramm:**

“Aus welchen Klassen besteht mein System und wie stehen diese untereinander in Beziehung?”

- **Paketdiagramm:**

“Wie kann ich mein Modell so schneiden, dass ich den Überblick bewahre?”

- **Objektdiagramm:**

“Welche innere Struktur besitzt mein System zu einem bestimmten Zeitpunkt zur Laufzeit (Klassendiagramm-Schnappschuss)?”



# UML: Rückblick und Zusammenfassung Diagramme

## Welche Fragen beantworten die UML-Diagramme?

- **Sequenzdiagramm:**

“Wer tauscht mit wem welche Informationen in welcher Reihenfolge aus?”

- **Kommunikationsdiagramm:**

“Wer kommuniziert mit wem? Wer „arbeitet“ im System zusammen?”

- **(Timingdiagramm:)**

“Wann befinden sich verschiedene Interaktionspartner in welchem Zustand?”

- **(Interaktionsübersichtdiagramm:)**

“Wann läuft welche Interaktion ab?”

# UML: Rückblick und Zusammenfassung Diagramme

---

## Welche Fragen beantworten die UML-Diagramme?

- **Komponentendiagramm:**

“Wie werden meine Klassen zu wiederverwendbaren, verwaltbaren Komponenten zusammengefasst und wie stehen diese miteinander in Beziehung?”

- **Verteilungsdiagramm:**

“Wie sieht das Einsatzumfeld (Hardware, Server, Datenbanken, ...) des Systems aus?  
Wie werden die Komponenten zur Laufzeit wohin verteilt?”

# UML: Rückblick und Zusammenfassung Diagramme

---

## Aktuell nicht besprochen:

- (Kompositionsdiagramm:)

“Wie sieht das Innenleben einer Klasse, einer Komponente, eines Systemteils aus?”

- (Interaktionsübersichtsdiagramm:)

“Wann läuft welche Interaktion ab?”

- (Zeitdiagramm:)

“Wann befinden sich verschiedene Interaktionspartner in welchem Zustand?”

# UML: Rückblick und Zusammenfassung

## Fazit

### UML Fazit:

Welche Vor- und Nachteile hat UML aus Ihrer Sicht?

- Teilweise (v. a. Aktivitätsdiagramm) auch ohne Informatik-Kenntnisse einfach zu verstehende, graphische Darstellung aus verschiedenen Sichtweisen
- Normung verschiedener Darstellungsweisen
- Auch automatische Abbildungen (Java Klassen + Klassendiagramm und umgekehrt) möglich und in diverse IDEs integriert
- Komplexe Zusammenhänge werden strukturiert dargestellt, Abläufe untergliedert und nach Zeit oder Voraussetzungen modelliert
- Nachteile: Rein visuell, teilweise uneingängige Symbolik, Schwierigkeit, die richtige Darstellung zu finden (es existieren Spracherweiterungen, z.B. OCL)

# Literatur

---

- [C. Rupp]: Chris Rupp et al., UML 2 glasklar
- [H. Balzert]: Heide Balzert, Lehrbuch der Objektmodellierung