

# Verteilte Systeme und Komponenten

Zusammenfassung Frühlingssemester 2018

Patrick Bucher

17.05.2018

## Inhaltsverzeichnis

<b>1</b>	<b>Komponenten</b>	<b>1</b>
1.1	Definition . . . . .	1
1.2	Eigenschaften . . . . .	2
1.3	Komponentenmodelle . . . . .	2

## 1 Komponenten

Herkunft: *componere* (lat.) = zusammensetzen

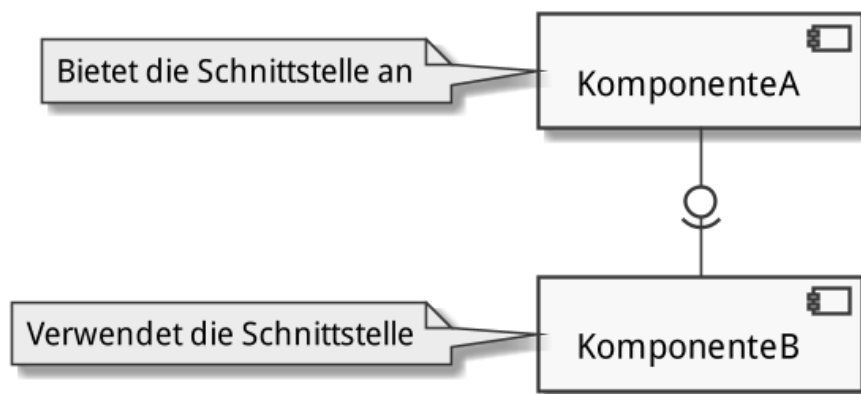


Abbildung 1: Komponentendiagramm (UML2)

### 1.1 Definition

Eine Software-Komponente

1. ist ein Software-Element
2. passt zu einem bestimmten Komponentenmodell
3. folgt einem bestimmten *Composition Standard*
4. kann ohne Änderungen mit anderen Komponenten verknüpft und ausgeführt werden

## 1.2 Eigenschaften

### Software-Komponenten

1. sind eigenständig ausführbare Softwareeinheiten
2. sind über ihre Schnittstellen austauschbar definiert
3. lassen sich unabhängig voneinander entwickeln
4. können kunden- und anwendungsspezifisch oder anwendungsneutral und wiederverwendbar sein
  - COTS (Commercial off-the-shelf): Software «von der Stange»
5. können installiert und deployed werden
6. können hierarchisch verschachtelt sein

## 1.3 Komponentenmodelle

### Komponentenmodelle

- sind konkrete Ausprägungen des Paradigmas der komponentenbasierten Entwicklung
- definieren die genaue Form und Eigenschaften einer Komponente
- definieren einen *Interaction Standard*
  - wie können die Komponenten miteinander über Schnittstellen kommunizieren (Schnittstellenstandard)
  - wie werden die Abhängigkeiten der Komponenten voneinander festgelegt
    - \* von der Komponente verlangte Abhängigkeiten: *Required Interfaces*
    - \* von der Komponente angebotene Abhängigkeiten: *Provided Interfaces*
- definieren einen *Composition Standard*
  - wie werden die Komponenten zu grösseren Einheiten zusammengefügt
  - wie werden die Komponenten ausgeliefert (Deployment)