



Technische Hochschule
Ingolstadt

Fakultät für Elektrotechnik
und Informatik

*Zukunft in
Bewegung*

Komponentenarchitektur

Software Engineering

Prof. Dr. Bernd Hafenrichter



Was ist eine Softwarearchitektur (Struktursicht

Struktursicht Sicht

- Fokus: Beschreibt die statische Struktur der Software in Form von Subsystemen und Komponenten

Multimedia-Device

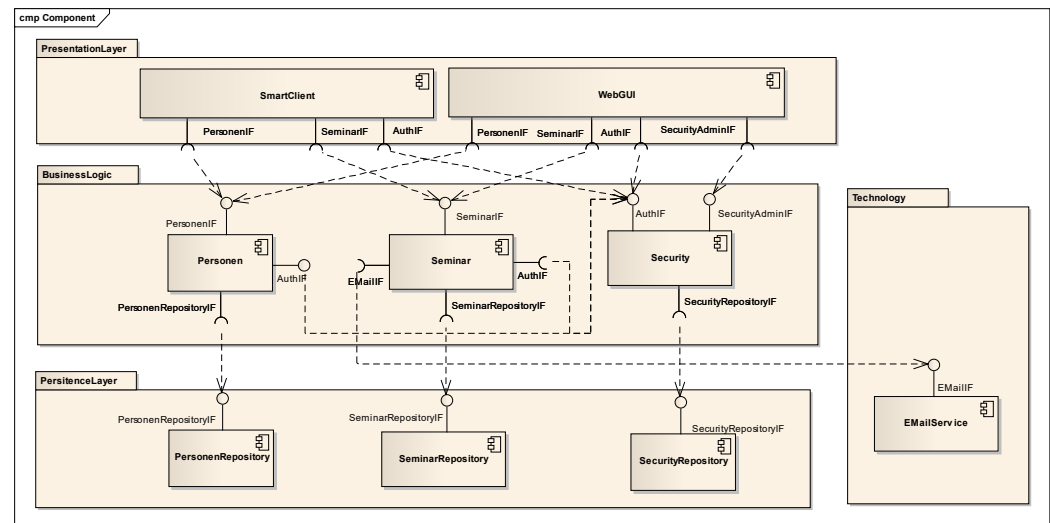
Telefon

Audio

Radio

Navigations

HW-Abstraktion-Layer

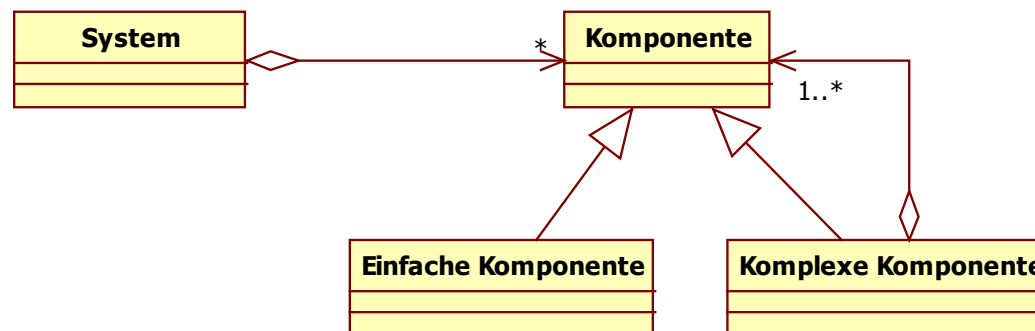


Grobe
Komponentenarchitektur

Detaillierte Komponentenarchitektur

Was ist eine Softwarearchitektur (Struktursicht)

- Beschreibung des Systems aus der Vogelperspektive
- Oft getrieben durch die nicht-funktionalen Anforderungen
- Zerlegung des Systems in kleine, von einander möglichst unabhängige Teile (= Komponenten)
- Bei großen Systemen können diese Komponenten wieder in Teile zerlegt. Dann stellt das Subsystem quasi selbst wieder ein System dar
- Abstraktion der spezifischen Aufgaben der Subsysteme → keine Details der Umsetzung innerhalb der Subsysteme



Definition „Komponente“

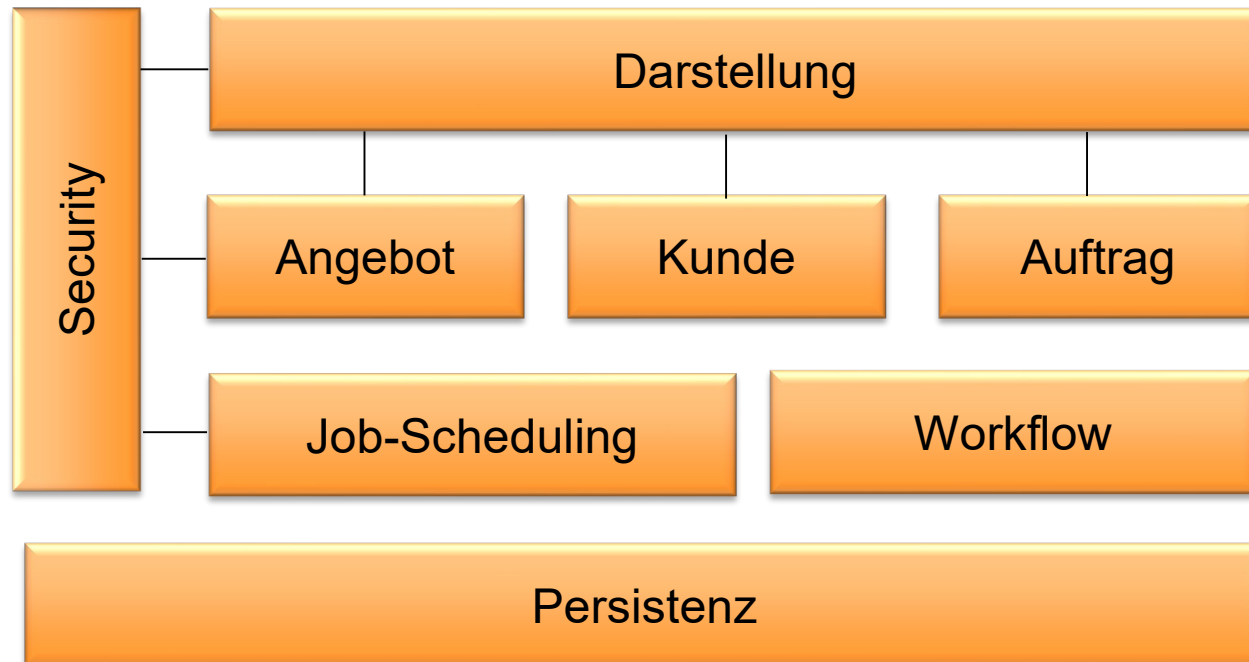
Motivation „Komponente“

- Komponenten stellen die Grundpfeiler einer Anwendung dar
- Die Zerlegung in Komponenten entspricht dem Prinzip der „Gliederung“.
- Die Festlegung von Abhängigkeit zwischen Komponenten entspricht der Lösungsstruktur
- Komponenten werden neutral, Implementierungsunabhängig definiert.
- Eine gute Komponentenstruktur ist der Ausgangspunkt für weitere Aktivitäten:
 - Entwurfsaktivitäten
 - Codierung
 - Test

Definition „Komponente“

Motivation „Komponente“

- Beispiel: Mögliche Komponenten einer Web-Anwendung



Definition „Komponente“

Die 6-Merkmale einer Komponente

- (1) Eine Komponente exportiert eine oder mehrere Schnittstellen
 - Schnittstellen werden als Verträge aufgefasst die neben der Syntax auch die Verarbeitungssemantik exakt definieren
 - Jede Komponente die eine Schnittstelle S exportiert stellt gleichzeitig eine Implementierung von S dar

Definition „Komponente“

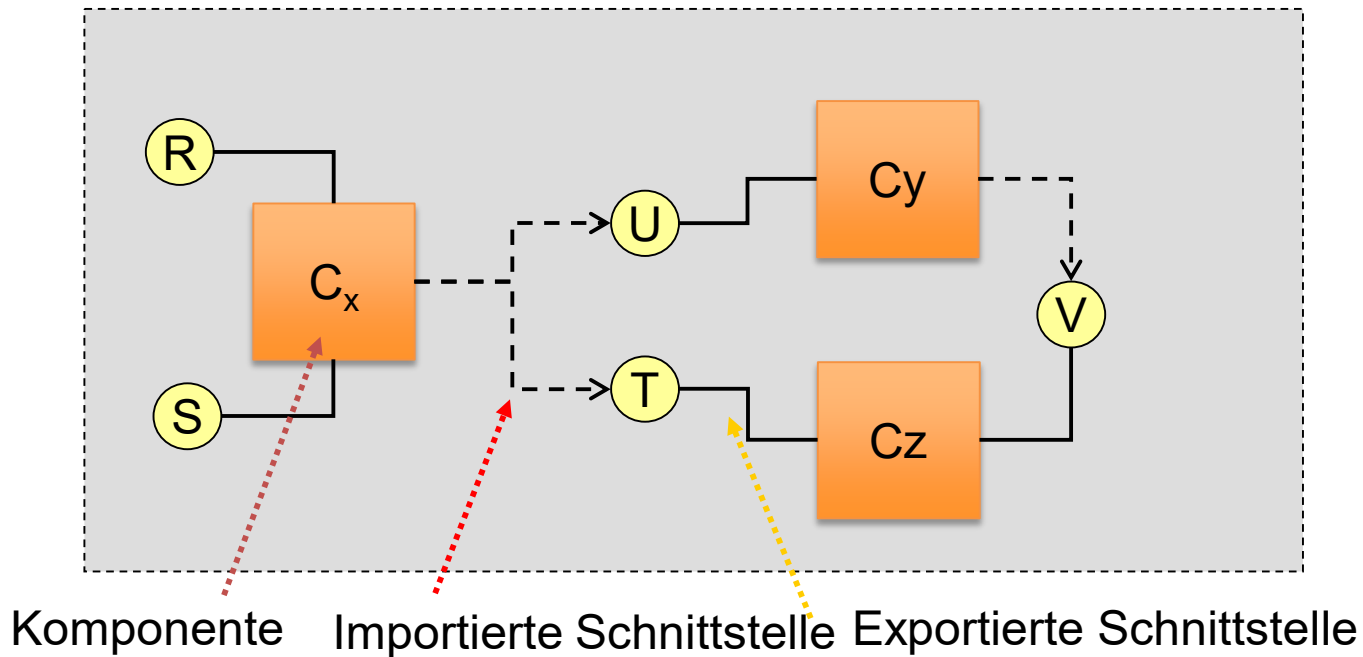
Die 6-Merkmale einer Komponente

- (2) Eine Komponente importiert andere Schnittstellen
 - Die Komponente benutzt die Methoden der importierten Schnittstelle
 - Eine Komponente ist erst lauffähig wenn alle importierten Schnittstellen zur Verfügung stehen
 - Die Komponente ist unabhängig von der Implementierung der importierten Schnittstellen

Definition „Komponente“

Die 6-Merkmale einer Komponente

- (1 + 2) Komponenten sind über Schnittstellen gekoppelt



Definition „Komponente“

Die 6-Merkmale einer Komponente

- (3) Eine Komponente versteckt die Implementierung
 - Eine Komponente kann durch andere Komponenten ersetzt werden
- (4) Eine Komponente definiert eine Einheit der Wiederverwendung
 - Eine Komponente kennt nicht die Umgebung in der Sie läuft
 - Sie macht nur minimale Angaben über die Umgebung

Definition „Komponente“

Die 6-Merkmale einer Komponente

- (5) Komponenten können andere Komponenten enthalten
 - Man kann neue Komponenten aus bestehenden Komponenten zusammensetzen (komponieren)
 - Die Komposition ist über beliebige Stufen möglich
- (6) Die Komponente spielt eine zentrale Rolle
 - für den Entwurf
 - für die Planung
 - und für die Implementierung



Definition „Komponente“

Weitere Begriffe

- Komponenten die nicht weiter unterteilt sind werden einfache Komponenten oder Modul genannt
- Alle anderen Komponenten werden als zusammengesetzte Komponenten (=Komposition) bezeichnet. Die Teile einer Komposition werden als Subkomponenten bezeichnet

Definition „Schnittstelle“

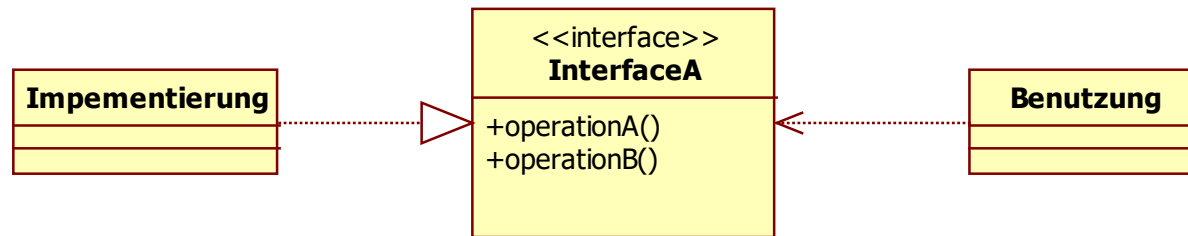
Schnittstellen sind das Bindeglied zwischen den Komponenten

- Die Definition einer Schnittstelle besteht aus einer Menge von verschiedenen Operationen.
 - Alle Operationen einer Schnittstelle werden über ein technisches Protokoll (z.B. RMI, WebServices, Methodenaufruf) abgebildet
- Eine Operation wird durch folgende Eigenschaften definiert
 - Syntax (Rückgabewerte, Argumenten, in/out, Typen)
 - Semantik (Was bewirkt die Methode?)
 - Nichtfunktionale Eigenschaften (Performance, Robustheit, Verfügbarkeit)

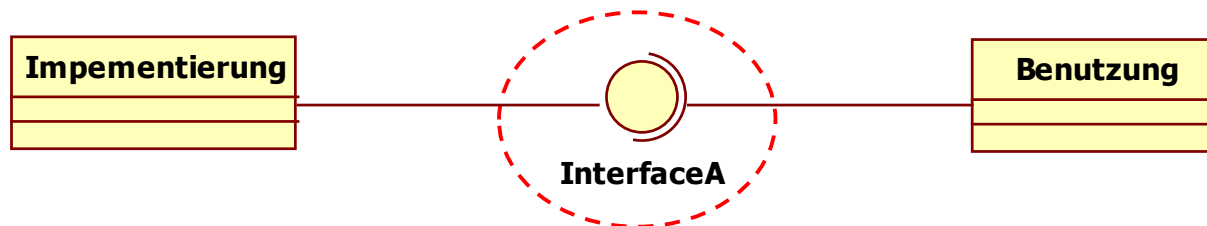
Definition „Schnittstelle“

Schnittstellen sind das Bindeglied zwischen den Komponenten

Darstellung einer Schnittstelle mit Stereotyp <<interface>>



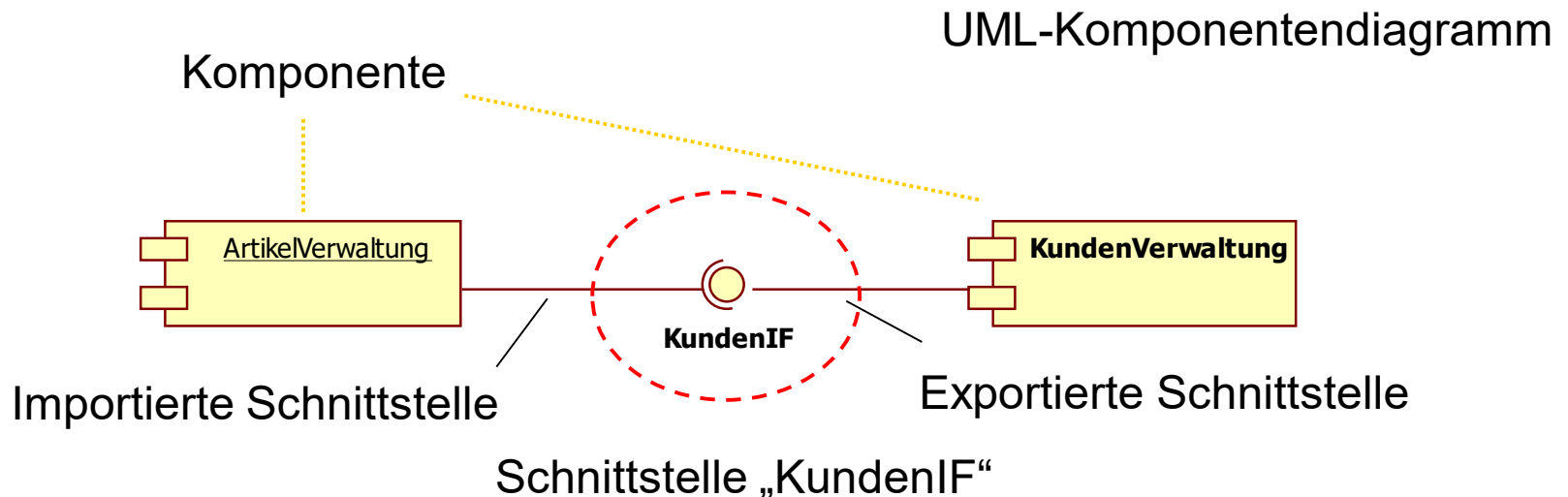
Darstellung einer Schnittstelle mit der Lollipop-Notation der UML



Definition „Schnittstelle“

Beispiel aus einem Onlineshop

- Komponente Artikelverwaltung importiert die Schnittstelle „KundenIF“
- Komponente Kundenverwaltung exportiert die Schnittstelle „KundenIF“
- Die Schnittstelle „KundenIF“ ist das Bindeglied zwischen den Komponenten



Definition „Schnittstelle“

Dokumentation einer Schnittstelle

- Die Syntax einer Schnittstelle kann in UML bzw. der jeweiligen Programmiersprache definiert werden
- Für die Semantik existiert keine allgemeingültige Möglichkeit der Definition
 - Informell durch zugehörige Schnittstellendokumentation
 - Mögliche Ansätze (wie z.B. QSL) verwenden Vor- und Nachbedingungen sowie Invarianten um die Semantik einer Schnittstelle zu beschreiben
 - Design by Contract
 - Semantik kann auch durch vordefinierte Testsuiten abgedeckt werden.

Zerlegung in Komponenten (Struktur)

Motivation: Warum Zerlegung in einzelne Komponenten

- Reduktion der Komplexität (Kleine Teile können leichter verstanden und entwickelt werden als große) → Divide and Conquer!
- Parallelisierung bei der Entwicklung (die einzelnen Komponenten können parallel verfeinert und entwickelt werden)
- Komponenten können von Zulieferern entwickelt werden, die Spezialisten in dem Bereich sind
- Komponenten können fertig gekauft werden, sog. Commercial off-the-shelf (COTS) Produkte → kommerzielle Produkte aus dem Regal (z.B. Datenbanken)
- Erhöhung der Wartbarkeit und Portierbarkeit (Komponenten sollen leicht austauschbar sein)

Zerlegung in Komponenten (Struktur)

Voraussetzung für eine sinnvolle Zerlegung

- Schnittpunkte zwischen den Komponenten müssen so gering wie möglich gehalten werden
- Schnittstellen zwischen den Komponenten müssen klar definiert sein
- Verantwortlichkeiten müssen sinnvoll auf die Komponenten aufgeteilt werden → starke Kohäsion anstreben
- Möglichkeit für Kompromisse sollte gegeben sein (besonders wichtig für die Nutzung von COTS-Komponenten)

Zerlegung in Komponenten (Struktur)

Wie findet man eine gute Komponentenstruktur

- Erfahrung
- Betrachtung von Architekturen ähnlicher Systeme
- Per Hand
- Separation of Concerns
- Trennung von Technologie und Fachlogik
- Betrachtung und Kombination von Architekturmustern/Referenzarchitekturen
 - Die Muster stellen hierbei Strukturierungsansätze dar, wie ein System gegliedert werden kann.
 - Die dargestellten Strukturen können mit Hilfe von Komponenten realisiert werden.