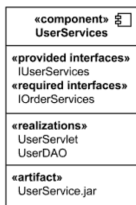


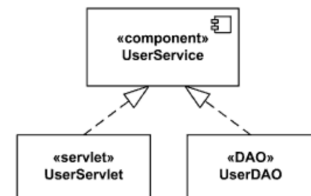
# UML

## Komponentendiagramme

Modellierung von Komponenten, deren bereitgestellte und benötigte Schnittstellen, Ports und Beziehungen zwischen ihnen



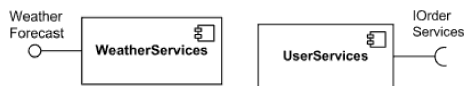
**Logische Komponenten** (z.B. Geschäftskomponenten)



**Physische Komponenten** (Programme, Maven-Packages)

Realisieren **interne Bestandteile** (z.B. Klassen) eine Komponente, können diese innerhalb Komponente in einem separaten **Abschnitt** ( <realization> compartment) notiert werden

### Schnittstellen

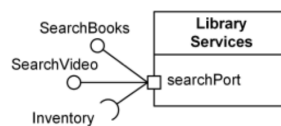


**Bereitgestellte** Schnittstelle wird **durch die Komponente** ealisiert.

**Benötigte** Schnittstelle definiert durch **Verwendungsabhängigkeit** ( <use> dependency) .

### Ports

Über Port kann Element mit Umgebung, mit anderen Objekten oder mit seinen internen Bestandteilen kommunizieren.

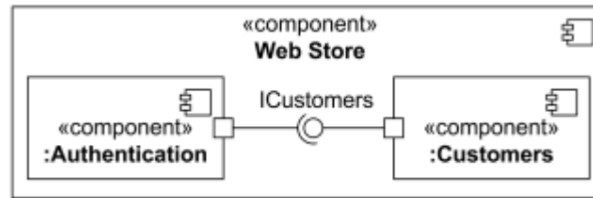


→ Standardsichtbarkeit: öffentlich (public)

→ Angabe von Multiplizitäten optional, z.B. [1..6]

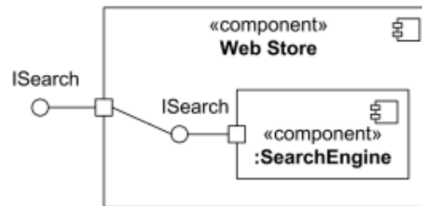
→ Simple port: Port mit nur einer Schnittstelle

## Assembly Connector



**Verbindung** zwischen min. zwei Komponenten, die Dienstbereitstellung und -nutzung zwischen Komponenten darstellt.

## Delegationskonnector



**Verbindung** eines externen Ports bzw. Schnittstelle einer Komponente mit ihren internen Bestandteilen

Komponenten sind Paketen ähnlich: Sie definieren Grenzen und gruppieren und gliedern Modellelemente

- Aber Komponenten definieren die Laufzeitsicht:
- Kapselung der enthaltenen Modellelemente
- Expliziter Export von Eigenschaften und Fähigkeiten der enthaltenen Komponenten
- Können Binärcode, Bibliotheken oder ausführbare Programme darstellen
- Können direkt auf Hardware oder Virtualisierungsinfrastruktur ausgeführt werden

Pakete: Logische Sicht auf Codestruktur zur Entwicklungszeit | Komponenten: Physische Sicht zur Laufzeit

Revision #3

Created 28 September 2022 13:05:08 by Merith Holtmann

Updated 28 September 2022 16:06:51 by Merith Holtmann