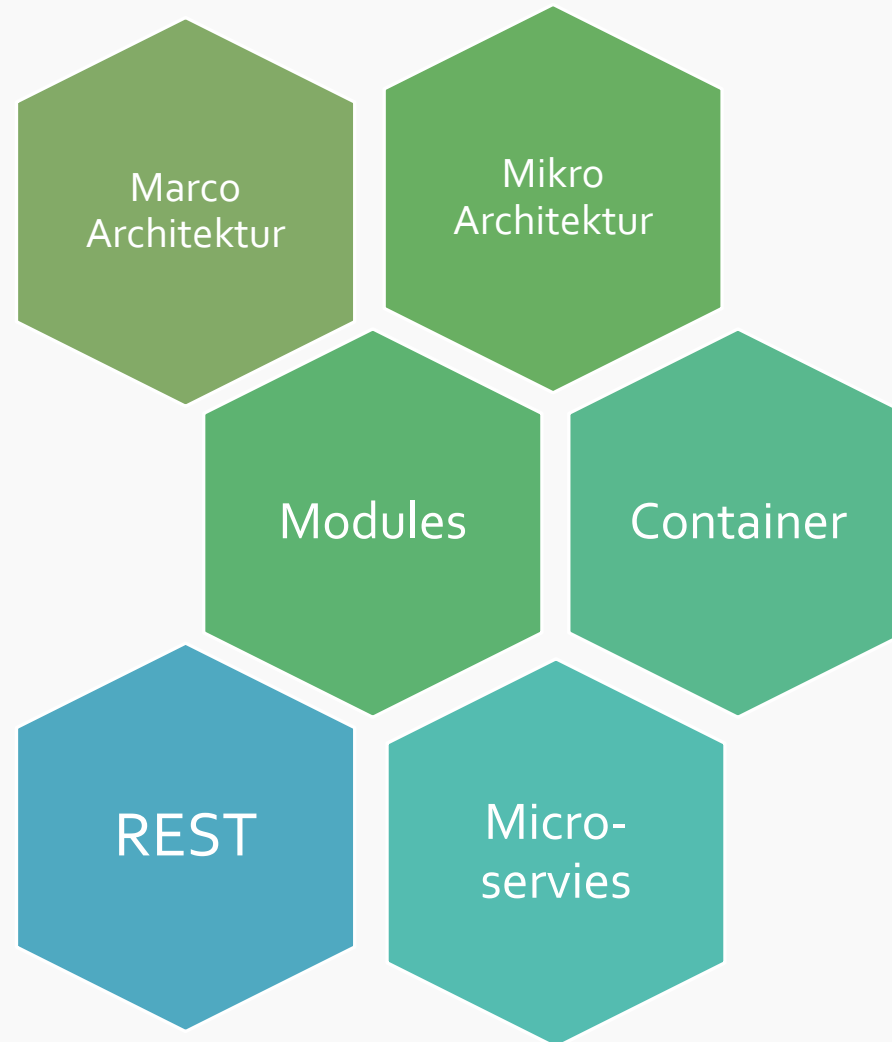


# Einführung Modul Softwarearchitektur



# Begriffliche Einordnungen



## Definitionen von Softwarearchitektur<sup>[1]</sup>

### Definition 1:

- *Software architecture is the set of design decisions which, if made incorrectly, may cause your project to be cancelled.* - Eoin Woods [sei]

In dieser Definition tritt weder das Wort Komponente, Microservice oder Schnittstelle auf, beschreibt einen wesentlichen Teil von Architektur aber sehr gut: In der Architektur geht es um **Entscheidungen**, die hohe Kosten verursachen können und sich später schwer rückgängig machen lassen.

### Definition 2:

- *The set of structures needed to reason about the system, which comprises software elements, relations among them, and properties of both.* - Paul Clements [CBB<sup>+</sup>10]

In dieser Definition geht es um Strukturen von Software und deren Eigenschaften. Zweifellos ist es wichtig zu wissen, aus welchen Bausteinen ein System besteht und welche Eigenschaften, auch **Qualitätsmerkmale** (3.1) genannt, diese Bausteine haben sollen. Es

ist vor allem die Dokumentation des Systems, die in dieser Definition eine tragende Rolle spielt, auch wenn sie nur implizit genannt ist.

### Definition 3:

- *Architecture is first and foremost key to achieving system understanding. As a vehicle for communication among stakeholders, it enables high-bandwidth, informed communication among developers, managers, customers, users, and others who otherwise would not have a shared language.* Paul Clements [CKK02]

Dies ist meine Lieblingsdefinition von Softwarearchitektur, weil sie die Bedeutung der gemeinsamen Sprache in der täglichen Kommunikation mit den Stakeholdern so schön hervorhebt. Für mich ist die Sprache ein subtiles, aber wertvolles Werkzeug der Entwicklung, dem ich in diesem Buch viel Platz gelassen habe. So bezieht sich dann auch die letzte Definition von Softwarearchitektur, die zu meinem Buchmodell passt, auf die **Allgemeinsprache** (5.3):

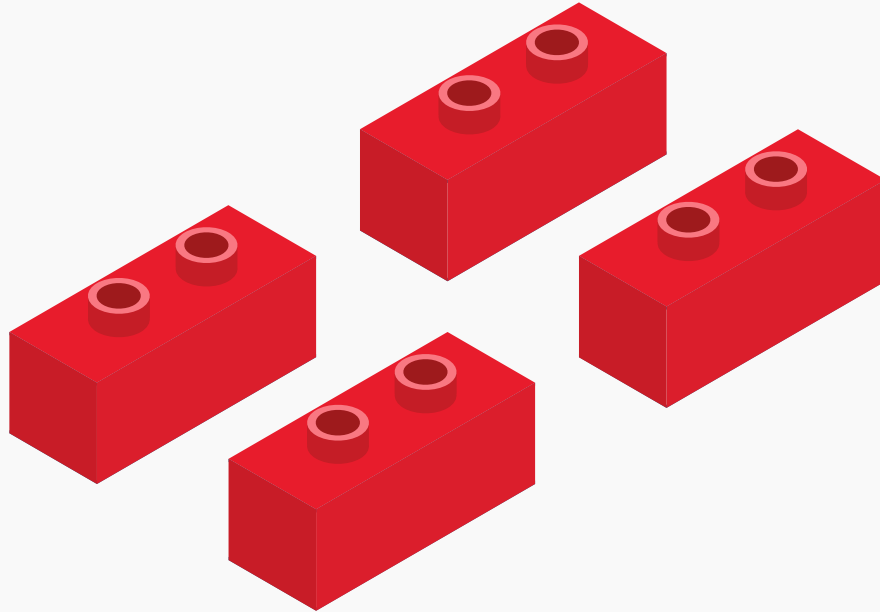
### Definition 4:

- *Softwarearchitektur ist die systematische Beschreibung der Services und Qualität eines Systems in der Allgemeinsprache.* Daniel Takai

---

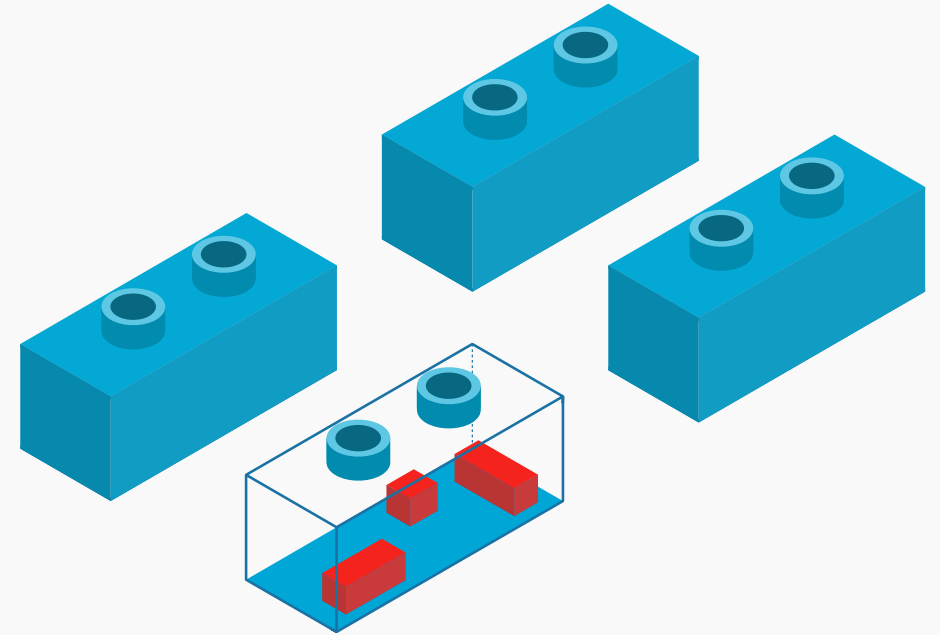
Quelle: Takai, Daniel: Architektur für Websysteme. Hanser Verlag, S. 7f.

# Makroarchitektur



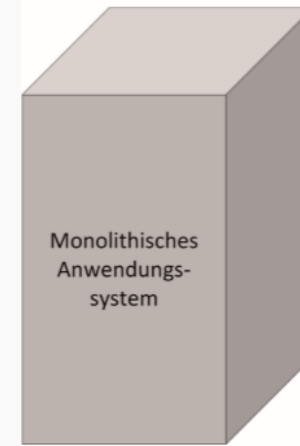
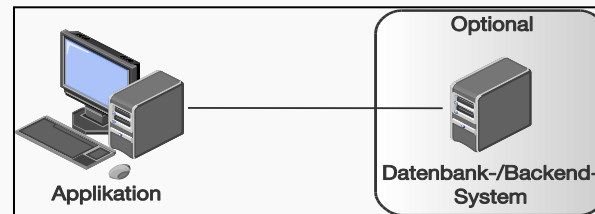
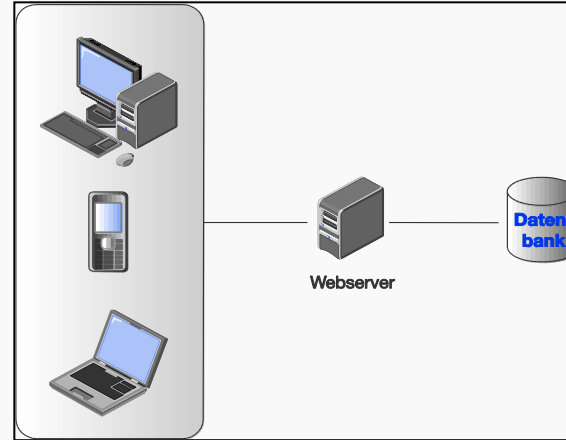
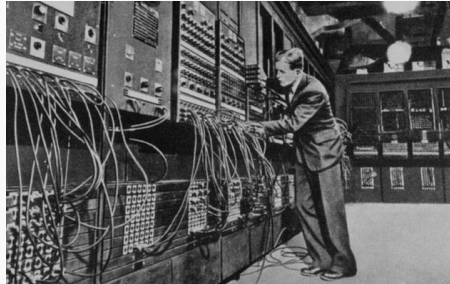
- Entscheidungen betreffen alle Module

# Mikroarchitektur



- Entscheidungen betrifft nur ein Modul

# Softwarearchitektur im Wandel der Zeit



Frontendframeworks

SOAP/SOA

Microservice

REST

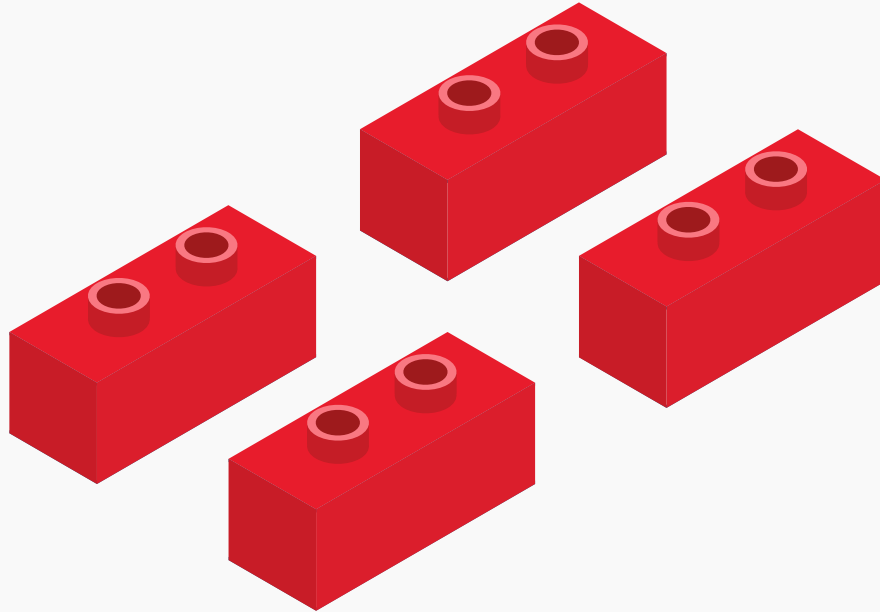
1960

1980

1990

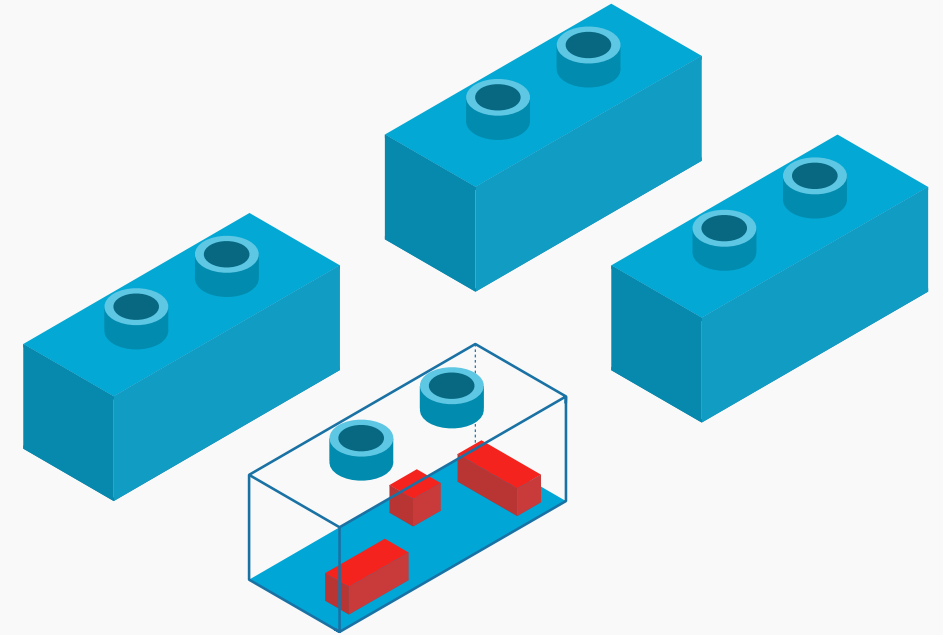
2000

# Makroarchitektur



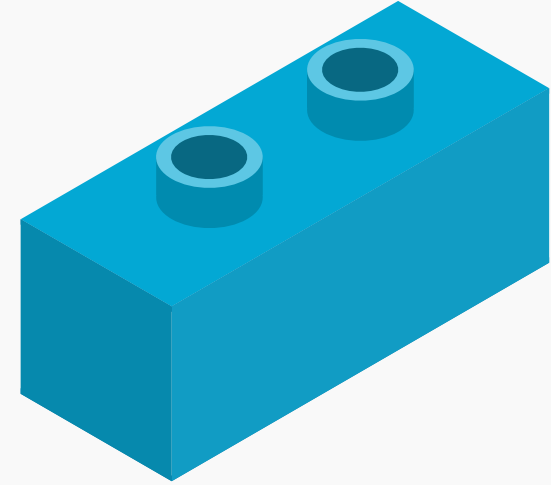
- Entscheidungen betreffen alle Module

# Mikroarchitektur



- Entscheidungen betrifft nur ein Modul

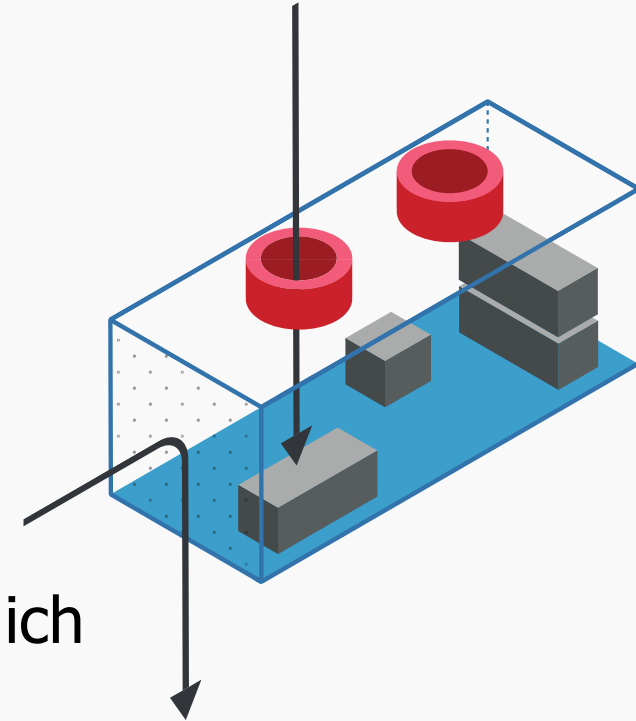
# Module



- Sind die Elemente der Mikroarchitektur
- Altes Prinzip, das eine Modularisierung ermöglicht
- Module bieten hohe Kohäsion bei gleichzeitig geringer Koppelung

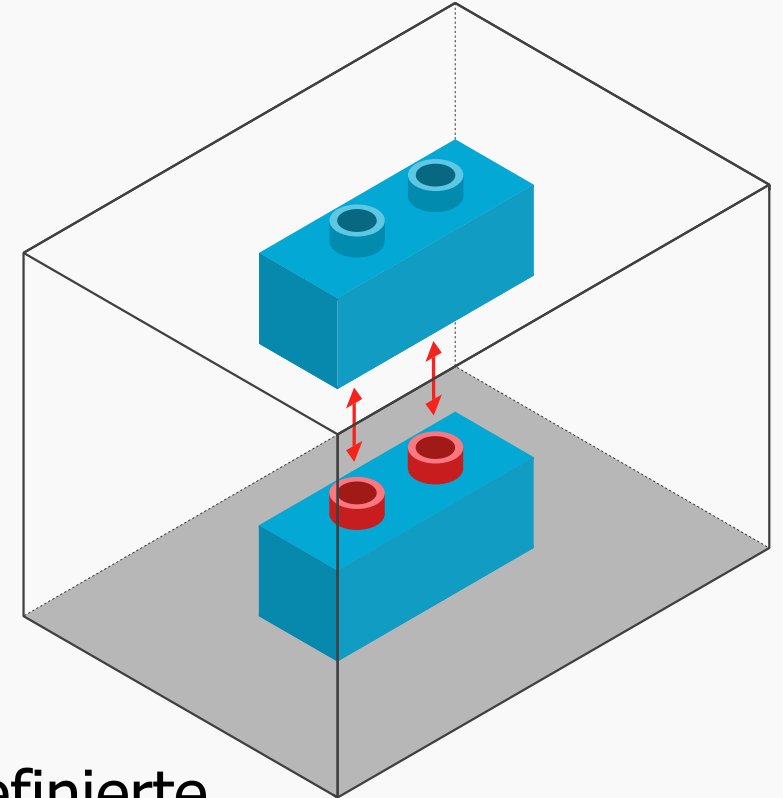
# Module

- Module bieten ein Interface an.
- Der Zugriff ist nur über das Interface möglich
- Datenkapselung auf Modulebene



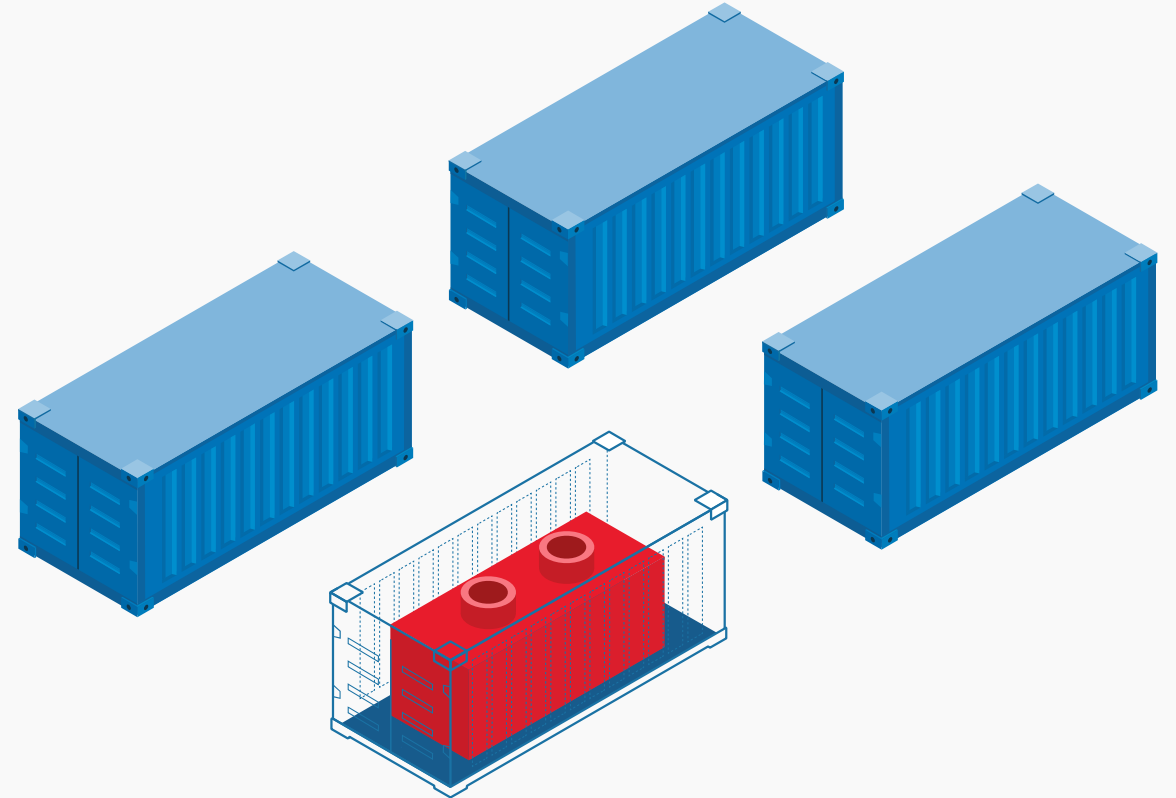


# Integration von Modulen



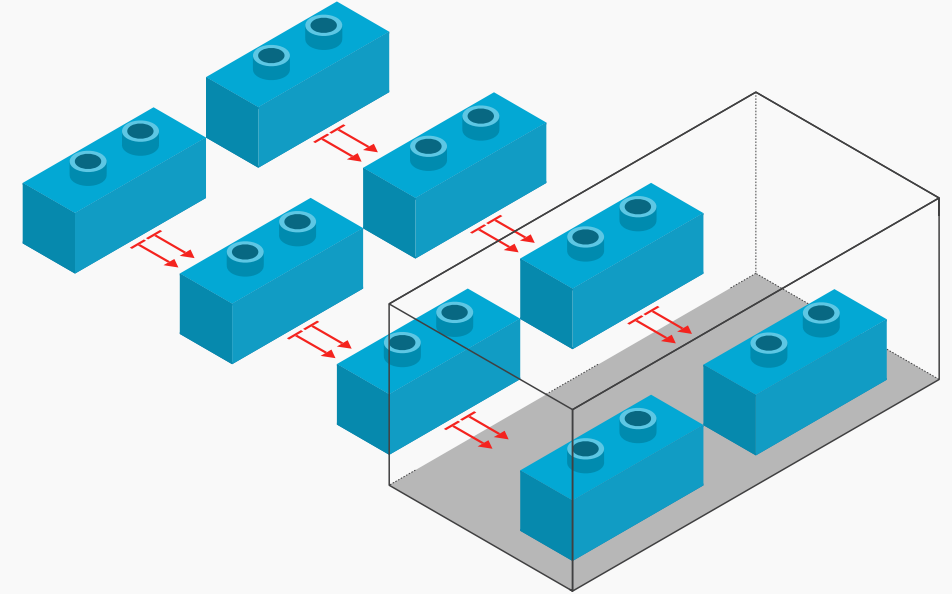
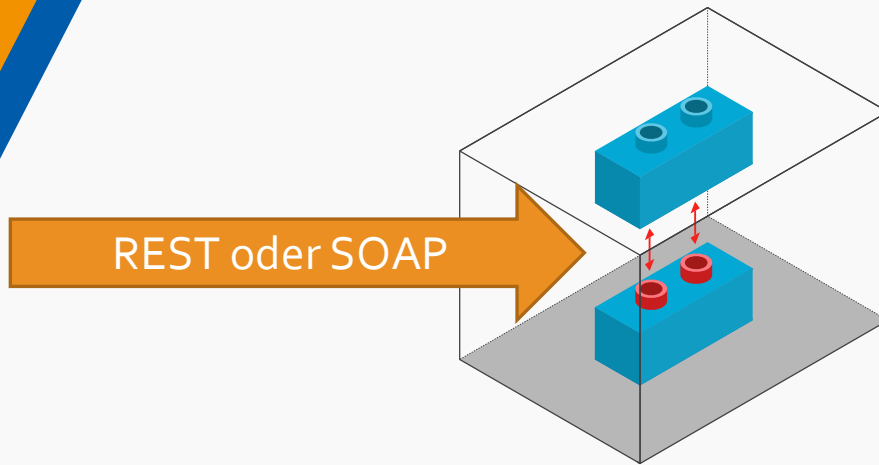
- Integration von Modulen erfolgt über definierte Schnittstellen:
  - Z.B. REST
  - Oder SOAP

# Container



- Module in Container zu packen, bietet allerlei Vorteile
- Auch hinsichtlich der Resilienz

# Microservice



- Microservices sind unabhängig deploybare Module.
- Microservices sind nur über Schnittstellen verbunden.
- Die Schnittstellen werden häufig mit REST umgesetzt.
- Welche Module es geben sollte, kann im Rahmen des Domain Driven Design überlegt werden.
- Bei der Betrachtung von Microservices sind sowohl Mikroarchitekturentscheidungen zu treffen, also auch Makroarchitekturentscheidungen.
- Eine Makroarchitekturentscheidung ist es z.B., ob REST als Schnittstelle verwendet wird und welches Datenformat (z.B. JSON) genutzt wird.