

Modul Softwarearchitektur



Ein paar Definitionen von Softwarearchitektur zu Beginn...¹

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⁺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):

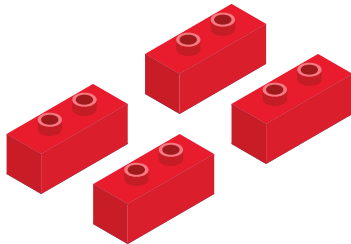
Definition 4:

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

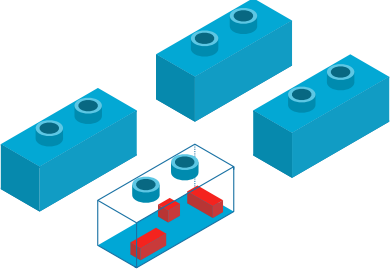
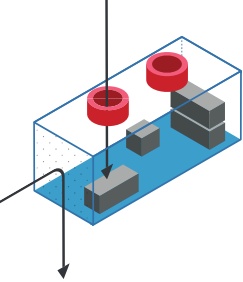
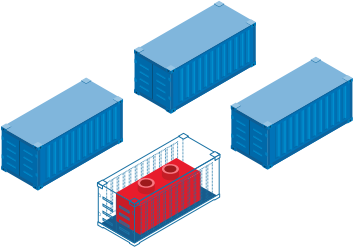
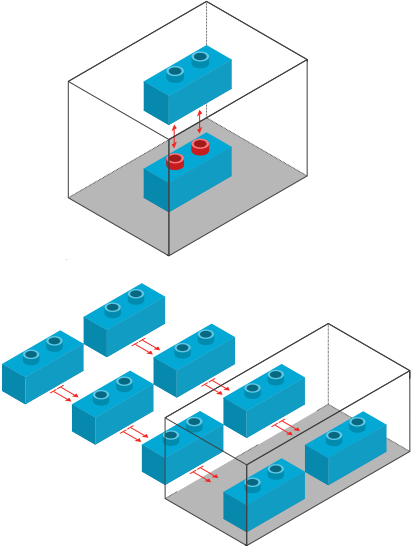
Begriff Einordnungen



Arbeitsauftrag Begriffe: Ergänzen Sie nachfolgende Übersicht mit den wesentlichen Informationen aus dem Erklärvideo.

Makroarchitektur		Entscheidungen betreffen alle Module
-------------------------	---	--------------------------------------

¹ Quelle: Takai, Daniel: Architektur für Websysteme. Hanser Verlag, S. 7f.

Mikroarchitektur		Entscheidungen betrifft nur ein Modul
Modul		<p>Elemente der Softwarearchitektur Altes Prinzip, das eine Modularisierung ermöglicht</p> <p>eine aufgabe - eng zusammenhängend enge koppelung</p> <p>zugriff nur über Interface Datenkapselung</p>
Container		module werden in container gepackt dadurch sind diese resilient
Microservice		<p>sind unabhängig deploybare Module</p> <p>schnittstelle oft mit rest</p> <p>Eine Makroarchitekturentscheidung ist es z.B., ob REST als Schnittstelle verwendet wird und welches Datenformat (z.B. JSON) genutzt wird.</p>

REST vs. SOAP



Arbeitsauftrag REST vs. SOAP: Bitte beantworten Sie basierend auf der Learning-Snack über REST vs. SOAP folgende Fragen:

1. Was ist REST?
2. Was ist SOAP?
3. Was sind zwei Unterschiede?



REST Representational State Transfer

umfasst eine Reihe von Softwarearchitektur-Prinzipien, die auf die Anforderungen schlanker Webservices und mobiler Anwendungen zugeschnitten sind. Da es sich um Richtlinien oder Empfehlungen handelt, bleibt die Implementierung letztlich den Entwicklern überlassen.

SOAP Simple Object Access Protocol

ist ein Standardprotokoll, das zunächst entwickelt wurde, damit Anwendungen, die mit verschiedenen Sprachen und auf verschiedenen Plattformen erstellt wurden, miteinander kommunizieren konnten. Da es sich um ein Protokoll handelt, umfasst es integrierte Regeln, die Komplexität und Overhead erhöhen, was zu längeren Seitenladezeiten führen kann.

Viele Altsysteme verwenden möglicherweise noch SOAP. REST wurde später eingeführt und gilt in webbasierten Szenarien häufig als die schnellere Alternative. REST besteht aus einer Reihe von Richtlinien für eine flexible Implementierung. SOAP ist ein Protokoll mit spezifischen Anforderungen wie XML-Messaging.

REST-APIs sind schlank und daher ideal für moderne Anwendungen geeignet, wie das Internet of Things (IoT), mobile Anwendungen und Serverless Computing. SOAP-Webservices bieten zwar integrierte Sicherheit und Transaktions-Compliance, die vielen Unternehmensanforderungen entspricht, sind aber aufwändiger zu erstellen. Darüber hinaus folgen auch viele öffentlich zugängliche APIs, wie die Google Maps-API, den REST-Richtlinien.