**Die Modulbeschreibung sollte direkt über diesen Link in HISinOne eingepflegt werden.**

| Module code | Module title | | Category |
|---|---|---|---|
| MAIE1030 | System Design | | MA |
| | Degree program | MA Software Engineering | |
| | Faculty | Building Services Engineering and Computer Science | |

| | |
|---|---|
| **Module coordinator** | Prof. Dr. Marcel Spehr |
| **Module type** | Mandatory module |
| **Frequency** | 1x annually in SuSe |
| **Recommended semester** | 1. semester |
| **Credit (ECTS-Points)** | 5 |
| **Academic Assessment Method** | Coursework requirement<br><br>PZ = Examination requirement (N: graded)<br><br>60% Project work with final group presentation<br><br>40% Homework presentations regarding technological deep dives on selected parts of the system design stack (group work) |
| **Teaching language** | English |
| **Admission requirements for this Module** | none |
| **Module duration** | 1 Semester |
| **Required Registration** | Students enrolled in the above-mentioned degree program/standard semester will be registered automatically upon re-enrollment; all other participants, please refer to the information below.<br><br>none |

| | Course | Lecturer | Type | Group Size (max.) | Number of Groups | Contact hours per week (SWS) | Workload (in h) | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | Face-to-face | Self-study |
| 1 | Systems Architecture | Spehr | Seminar | 30 | 1 | 4 | 60 | 65 |
| 2 | Titel der Lehrveranstaltung. | Dozent*in | Wählen Sie ein Element aus. | | Wählen Sie ein Element aus. | | | |
| 3 | Titel der Lehrveranstaltung. | Dozent*in | Wählen Sie ein Element aus. | | Wählen Sie ein Element aus. | | | |
| 4 | | | Wählen Sie ein | | | | | |

Zuletzt bearbeitet: 08.02.2026

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Ele-ment aus. | | | | | |
| **5** | Titel der Lehrver-anstaltung. | Dozent*in | Wählen Sie ein Ele-ment aus. | | Wählen Sie ein Element aus. | | | |
| | | | | | Sum | **4,0** | **60** | **65** |
| | | | | | **Total Workload for Module** | | **125** | |

| | |
|---|---|
| **Learning Objectives / Learning outcomes** | Students are able to:<br>• Elicit requirements for complex systems and translate them into technical specifications<br>• Understand the current landscape of technical topics and apply appropriate tools in a problem-oriented manner<br>• Independently explore a technical topic and present it in an understandable way to a professional audience<br>• Identify typical components of a complex system landscape and apply example candidates accordingly |
| **Contents** | • **Architectural patterns for complex systems**<br>  ○ Loose coupling<br>  ○ Microservices<br>• **Kubernetes and Docker Compose for administration of the system landscape**<br>• **Introduction to and use of typical components of complex systems**<br>  ○ Data lakes<br>  ○ Databases<br>  ○ AI<br>  ○ Message-passing platforms (e.g., Kafka)<br>  ○ Credential management (e.g., LDAP)<br>• **Software as a Service (SaaS)**<br>• **Hardware and software solutions for specific tasks** |
| **Literature** | • JJ Gewax: API Design Patterns. Manning. 2021<br>• Mark Richards: Software architecture patterns. O'Reilly. 2015<br>• Gernot Starke: Effektive Software Architekturen. Carl Hanser Verlag. 2020<br>• Mark Richards: Fundamentals of software architectures. O'Reilly. 2020<br>• John Ousterhout: Prinzipien des Softwaredesigns: Entwurfs-strategien für komplexe Systeme. O'Reilly. 2021<br>• Robert C. Martin: Clean Architecture. Addison-Wesley. 2017<br>Neal Ford: Software Architecture: The Hard Parts: Modern Tradeoff Analysis for Distributed Architectures. O'Reilly. 2021 |