

Die Modulbeschreibung sollte direkt über diesen [Link](#) in HISinOne eingepflegt werden.

Module code	Module title	Category
MAI1030	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 PZ = Examination requirement (N: graded) 60% Project work with final group presentation 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. 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					

			Element aus.					
5	Titel der Lehrveranstaltung.	Dozent*in	Wählen Sie ein Element aus.		Wählen Sie ein Element aus.			
Sum						4,0	60	65
Total Workload for Module							125	

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