

Modulcode (1.)	Modulbezeichnung (2.)	Zuordnung (3.)
MAI1030	System Architecture (SA)	
	Studiengang (4.)	Master Angewandte Informatik
	Fakultät (5.)	Gebäudetechnik und Informatik

Modulverantwortlich (6.)	Prof. Dr. Marcel Spehr
Modulart (7.)	Pflicht
Angebotshäufigkeit (8.)	SS
Regelbelegung / Empf. Semester (9.)	MA1
Credits (ECTS) (10.)	5 CP
Leistungsnachweis (11.)	T(N) (SL) + M(B) (SL)
Unterrichtssprache (12.)	Deutsch
Voraussetzungen für dieses Modul (13.)	Kenntnisse und Kompetenzen entsprechend Modulen BAI1040: Grundkonzepte der Programmierung BAI2040: Objektorientierte Programmierung BAI3020: Softwaretechnik 1 BAI4020: Softwaretechnik 2
Modul ist Voraussetzung für (14.)	
Moduldauer (15.)	1 Semester
Notwendige Anmeldung (16.)	-
Verwendbarkeit des Moduls (17.)	-

(18.)	Lehrveranstaltung	(19.)	Dozent/in	(20.)	Art	(21.)	Teilnehmer (maximal)	(22.)	Anzahl Gruppen	(23.)	Workload		
											(24.)	Präsenz	Selbst- studium (25.)
1	System Architecture		Spehr		S		15		1		4	60	65
Summe										4	60	65	
Workload für das Modul (26.)											125		

Qualifikationsziele (27.)	<p>Die Studierenden können:</p> <ul style="list-style-type: none"> • Anforderungen an komplexe Systeme aufnehmen und in technische Vorgaben übersetzen • die aktuelle technische Themenlandschaft überblicken und problemadaptiv geeignete Werkzeuge einsetzen • selbständig ein technisches Thema erschließen und es für ein Fachauditorium verständlich aufbereiten • typische Komponenten einer komplexen Systemlandschaft benennen und exemplarisch Kandidaten einsetzen
Inhalte (28.)	<ul style="list-style-type: none"> • Architekturmuster für komplexe System <ul style="list-style-type: none"> ○ Loose-Coupling ○ Microservices • Kubernetes und Docker-Compose für Administration der Systemlandschaft • Kennenlernen und Verwendung typischer Komponenten komplexer Systeme <ul style="list-style-type: none"> ○ Data Lakes ○ Datenbanken ○ KI ○ Message-Passing-Plattform (bspw. Kafka) ○ Verwaltung von Anmeldeinformationen (bspw. LDAP) • Software-As-A-Service • Hard- und Softwarelösungen für Teilaufgaben
Vorleistungen und Modulprüfung (29.)	<ul style="list-style-type: none"> • Vorleistungen: <ul style="list-style-type: none"> ○ Keine • Modulprüfung: <ul style="list-style-type: none"> ○ Klausur im Prüfungszeitraum ○ Präsentation einer Hausaufgabe im Semesterverlauf
Literatur (30.)	<ul style="list-style-type: none"> • 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