

Modul: Cloud-native Architekturen

Niveau	Master	Kürzel	CloudArch
Modulname englisch	Cloud-native Architectures		
Modulverantwortliche	Nane Kratzke		
Fachbereich	Elektrotechnik und Informatik		
Studiengang	Informatik/Softwaretechnik für verteilte Systeme, Master		
Verpflichtungsgrad	Pflicht	ECTS-Leistungspunkte	5
Fachsemester	2	Semesterwochenstunden	4
Dauer in Semestern	1	Arbeitsaufwand in Stunden	150
Angebotshäufigkeit	SoSe	Präsenzstunden	60
Lehrsprache	Deutsch/Englisch	Selbststudiumsstunden	90

Der folgende Abschnitt ist nur ausgefüllt, wenn es **genau eine** modulabschließende Prüfung gibt.

Prüfungsleistung	Projektarbeit	Prüfungsprache	Deutsch/Englisch
Dauer PL in Minuten		Bewertungssystem PL	Drittelnoten
Lernergebnisse	<ul style="list-style-type: none"> • Die Studenten kennen die NIST Definition des Cloud Computing und die CNCF Definition von Cloud-native. • Die Studenten kennen die CNCF Landscape und nutzen diese (oder vglb. Rahmenwerke) zielgerichtet zur Auswahl, Beurteilung und Entwicklung von Komponenten für Cloud-native Applikationen. • Die Studenten wenden das IDEAL-Prinzip zielgerichtet zur Entwicklung Cloud-nativer Architekturen an und berücksichtigen dabei die beiden dominanten Cloud-nativen Architekturstile Microservices und Serverless Architectures. • Die Studenten können elastische und resiliente Ansätze für das Scheduling & Orchestration, die Service Coordination & Service Discovery, das Traffic Management, die Observability & Analysis, das Continuous Integration & Deployment, sowie des Service Provisioning die im Rahmen Cloud-nativer Architekturen eingesetzt werden, einschätzen, abwägen und für eigene Architekturen zielgerichtet einsetzen. • Die Studenten kennen die Vor- und Nachteile statusloser und statusbehafteter Komponenten vor dem Hintergrund von Skalierbarkeit und Elastizität. • Die Studenten kennen die Besonderheiten verteilter Systeme, des CAP Theorems, sowie die Prinzipien der Daten-, Event- und Call-basierten Kopplung und nutzen diese zielgerichtet zur Entwicklung entsprechender Cloud-nativer Architekturen. • Die Studenten sind sich der Gesetzmäßigkeiten des Pay-as-you-go Kostenmodells bewusst und können die oben genannten Ansätze zielgerichtet dazu einsetzen, möglichst Ressourcen- 		

	schonende Architekturen für Cloud-native Applikationen und Dienste zu entwerfen.
Teilnahmevoraussetzungen	Das Modul Cloud-native Programmierung sollte bestanden und dessen Voraussetzungen erfüllt sein.
Der vorige Abschnitt ist nur ausgefüllt, wenn es genau eine modulabschließende Prüfung gibt.	
Berücksichtigung von Gender- und Diversity-Aspekten	<ul style="list-style-type: none"> ✘ Verwendung geschlechtergerechter Sprache (THL-Standard) ✘ Zielgruppengerechte Anpassung der didaktischen Methoden ✘ Sichtbarmachen von Vielfalt im Fach (Forscherinnen, Kulturen etc.)
Verwendbarkeit	
Bemerkungen	Die Berücksichtigung von Gender- und Diversity-Aspekten erfolgt nicht nach formalen Vorgaben, sondern wird im Rahmen des allgemein üblichen und gesellschaftlich akzeptierten Miteinanders praktiziert.

Lehrveranstaltung: Cloud-native Architekturen (Vorlesung)

(zu Modul: Cloud-native Architekturen)

Lehrveranstaltungsart	Vorlesung	Lernform	Präsenz
LV-Name englisch	Cloud-native Architectures (Lecture)		
Anwesenheitspflicht	nein	ECTS-Leistungspunkte	2
Teilnahmebeschränkung	200	Semesterwochenstunden	3
Gruppengröße	200	Arbeitsaufwand in Stunden	60
Lehrsprache	Deutsch/Englisch	Präsenzstunden	30
Studienleistung		Selbststudiumsstunden	30
Dauer SL in Minuten		Bewertungssystem SL	

Der folgende Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

Prüfungsleistung		Prüfsprache	
Dauer PL in Minuten		Bewertungssystem PL	
Lernergebnisse			
Teilnahmevoraussetzungen			

Der vorige Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

Lehrinhalte	<ul style="list-style-type: none"> • Die NIST-Definition des Cloud Computing • Das Pay-as-you-go Kostenmodell des Cloud Computing und dessen Auswirkung auf wirtschaftliche Architekturen für Cloud-native Applikationen und Dienste • Elastizität, Skalierbarkeit und Resilienz in Cloud-nativen Applikationen und Diensten • Die CNCF Definition von Cloud-native und die CNCF Landscape • Das IDEAL Prinzip des Service Computing und die dominierenden Architekturstile im Cloud Computing: Microservices und Serverless Architectures • Service Scheduling & Orchestration, Service Coordination & Service Discovery, Traffic Management, Observability & Analysis, Service Provisioning, Continuous Integration & Deployment • Stateless vs. Stateful • (Lose) Kopplung von Diensten mittels Daten-, Event- oder Call-basierter Kopplung vor dem Hintergrund des CAP-Theorems in verteilten Systemen
Literatur	<ul style="list-style-type: none"> • Nane Kratzke, Cloud-native Computing, Hanser, 2021 • Christoph Fehling, Frank Leymann, Ralph Retter, Walter Schupeck, Peter Arbitter, Cloud Computing Patterns, Springer • Cornelia Davis, Cloud-native – Designing change-tolerant software, Manning

	<ul style="list-style-type: none">• CNCF Landscape, https://landscape.cncf.io
Bemerkungen	

Lehrveranstaltung: Cloud-native Architekturen (Praktikum)

(zu Modul: Cloud-native Architekturen)

Lehrveranstaltungsart	Praktikum	Lernform	Präsenz
LV-Name englisch	Cloud-native Programming (Practical Course)		
Anwesenheitspflicht	ja	ECTS-Leistungspunkte	3
Teilnahmebeschränkung	24	Semesterwochenstunden	1
Gruppengröße	12	Arbeitsaufwand in Stunden	90
Lehrsprache	Deutsch/Englisch	Präsenzstunden	30
Studienleistung		Selbststudiumsstunden	60
Dauer SL in Minuten		Bewertungssystem SL	

Der folgende Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

Prüfungsleistung		Prüfsprache	
Dauer PL in Minuten		Bewertungssystem PL	
Lernergebnisse			
Teilnahmevoraussetzungen			

Der vorige Abschnitt ist nur ausgefüllt, wenn es eine lehrveranstaltungsspezifische Prüfung gibt.

Lehrinhalte	Siehe Vorlesung (diese werden in einem praktischen Projekt vertieft und angewendet)
Literatur	Siehe Vorlesung
Bemerkungen	