



Cloud Native Architecture and Engineering

Module title:

Cloud Native Architecture and Engineering

Credits:

6

Responsible person:

Tai, Stefan

Office:

EN 14

Contact person:

Hummel, Anita

Website:

<http://www.ise.tu-berlin.de>

Display language:

Englisch

E-mail address:

lehre@ise.tu-berlin.de

Learning Outcomes

Students understand the principles of architecting and engineering cloud-native software systems. They are familiar with architectural opportunities and constraints that the cloud offers and imposes and are able to competently assess, select, and apply fitting technologies to meet given functional and non-functional requirements.

Content

Cloud-native architecture is about tailoring software architectures to embrace cloud computing services (e.g., AWS Lambda or Google Cloud Spanner) for designing scalable and resilient software solutions. The cloud provides different services that serve both as building blocks and target environment (including, but not limited to serverless computing, container orchestration, and virtual machine clusters as execution environments and as part of microservice compositions). Different programming and runtime models present different opportunities and constraints. A well-architected cloud-native system is elastic, observable, cost-efficient, and easily manageable through continuous software engineering methods and tools. This course covers respective principles, methods and technologies, along with real-world application examples. Emphasis is also set on relevant engineering principles related to DevOps and chaos engineering.

Module Components

Course Name	Type	Number	Cycle	SWS
Cloud Native Architecture and Engineering	IV	0435 L 336	WiSe	4

Workload and Credit Points

Cloud Native Architecture and Engineering (Integrierte Veranstaltung)	Multiplier	Hours	Total
Preparation and rework	15.0	8.0h	120.0h
Lecture and Lab Time (may be offered online)	15.0	4.0h	60.0h
			180.0h

The Workload of the module sums up to 180.0 Hours. Therefore the module contains 6 Credits.

Description of Teaching and Learning Methods

The course is offered as "integrated classroom learning" comprising lectures, reading assignments, student presentations, and lab-based implementation. Students are required to acquire in-depth knowledge on select aspects through thorough self-study of current literature.

Requirements for participation and examination

Desirable prerequisites for participation in the courses:

- Distributed systems fundamentals,
- Software engineering fundamentals,
- Solid programming skills

Mandatory requirements for the module test application:

keine Angabe

Module completion

Grading:

graded

Type of exam:

Portfolioprüfung
100 points in total

Language:

English

Grading scale:

Note:	1.0	1.3	1.7	2.0	2.3	2.7	3.0	3.3	3.7	4.0
Punkte:	95.0	90.0	85.0	80.0	75.0	70.0	65.0	60.0	55.0	50.0

Test description:

In total 100 portfolio points can be achieved. The overall grade is determined according to §47 (2) AllgStuPO, grading scheme 2 of the Faculty IV.

Test elements	Categorie	Points	Duration/Extent
(Deliverable Assessment) Design document (Software architectural design)	practical	20	ca. 5 pages
(Deliverable Assessment) Practical Assignment (Programming/code)	practical	20	ca. 40 h
(Deliverable Assessment) Reading Assignment and Presentation (Presentation of findings)	oral	20	ca. 15 min
(Learning Process Review) Oral consultation (may be in groups)	oral	20	ca. 15 min per student
(Deliverable Assessment) Report (Assessment and discussion of practical assignment)	practical	20	ca. 5 pages

Duration of the Module

The following number of semesters is estimated for taking and completing the module:

1 Semester

This module may be commenced in the following semesters:

Sommersemester

Maximum Number of Participants

The maximum capacity of students is 50

Registration Procedures

Will be announced during the first lecture. www.tu.berlin/ise/studium-lehre

Recommended reading, Lecture notes

Lecture notes:

unavailable

Electronical lecture notes :

available

Assigned Degree Programs

This moduleversion is used in the following modulelists:

Computer Engineering (Master of Science)
StuPO 2015
Modullisten der Semester: SoSe 2023
Computer Science (Informatik) (Master of Science)
StuPO 2015
Modullisten der Semester: SoSe 2023
Elektrotechnik (Master of Science)
StuPO 2015
Modullisten der Semester: SoSe 2023
Information Systems Management (Wirtschaftsinformatik) (Master of Science)
StuPO 2017
Modullisten der Semester: SoSe 2023
Medieninformatik (Master of Science)
StuPO 2017
Modullisten der Semester: SoSe 2023

Miscellaneous

No information