

Course size (nominal values; actual values may depend on programme)

Credits	3.0	Study time	80 h	Contact hrs	30.0 h
---------	-----	------------	------	-------------	--------

Course offerings and teaching methods in academic year 2018-2019

A (semester 1)	Dutch	lecture	12.0 h
		seminar: practical PC room classes	18.0 h

Lecturers in academic year 2018-2019

Simoens, Pieter	TW05	lecturer-in-charge
-----------------	------	--------------------

Offered in the following programmes in 2018-2019

	crdts	offering
Master of Science in Information Engineering Technology	3	A

Teaching languages

Dutch

Keywords

system design, cloud, big data, distributed systems, microservices

Position of the course

This course covers design and implementation techniques for large-scale software project. In particular, we target applications that need to be executed on more than 1 server node for reasons of performance.

Covered topics: distributed systems, cloud computing, microservices, big data.

Contents

- view models of a software architecture
- reactive applications and system design
- container technology and container orchestration (Docker, Kubernetes)
- dimensions and models of cloud computing (IaaS/PaaS/CaaS/SaaS)
- principles of domain-driven design (aggregates, domain services, etc.)
- elastic scaling of cloud-based applications
- design principles for cloud-native applications
- the microservice architectural pattern
- communication between micro-services
- relational and non-relational data models
- storage, consistency, replication and partitioning of data in distributed systems
- processing models for big data: batch, streaming

Initial competences

Final competences

- 1 To be able to apply the principles of advanced software design with a view to production, maintenance and quality.
- 2 To be able to understand and to apply the techniques of organized software design, as used in software companies and bigger computer departments.
- 3 To be able to design, research, analyze and diagnose.

Conditions for credit contract

Access to this course unit via a credit contract is determined after successful competences assessment

Conditions for exam contract

This course unit cannot be taken via an exam contract

Teaching methods

Lecture, seminar: practical PC room classes

Extra information on the teaching methods

Lectures involve both theoretical concepts as practical applications.

6 weeks 2u theory

Lab exercises: hands-on experience with discussed technologies.

6 weeks 3u lab

Learning materials and price

Annotated slides.

References

- R. Wattenhofer, principles of Distributed Computing (lecture collection)
- D. Thain, Foundations of Distributed Systems
- D. McCreary, Making Sense of NoSQL
- R. STrickland, Cassandra High Availability
- N. Marz and J. Warren, "Big Data"
- Hadoop MapReduce v2 Cookbook
- N. Garg, Learning Apache Kafka (2nd edition)
- S. Saxena, Real-time analytics with Storm and Cassandra
- C. Fehling, Cloud Computing Patterns
- B. Wilder, Cloud Architecture Patterns
- S. Newman, Building Microservices

Course content-related study coaching

Evaluation methods

end-of-term evaluation

Examination methods in case of periodic evaluation during the first examination period

Written examination, simulation

Examination methods in case of periodic evaluation during the second examination period

Written examination, skills test

Examination methods in case of permanent evaluation

Possibilities of retake in case of permanent evaluation

not applicable

Extra information on the examination methods

Written exam: theory and paper exercises.

Practical exam: exercises on lab PC using available documentation and course material.

Calculation of the examination mark

Score is the average of both parts of the examination. To calculate the final score, each part is weighted for 1/2.

If a student obtains less than 8/20 for at least one part of the exam, the aggregate score is capped to 8/20.