W3WI_110.2 - Distributed Systems

Lecturer: Prof. Dr. Michael Eichberg
Contact: michael.eichberg@dhbw.de, Raum 149B
Version: 2.0 (24SCA und 24EG/EH)

Core Topics

- Terminology, concepts, architectures, requirements profiles and architecture models for distributed systems
- Design and implementation approaches
- Comparison of different middleware concepts
- Synchronous and asynchronous communication, remote method calls
- Asynchronous communication and messaging systems
- Security aspects in distributed systems

Previous Knowledge

- Who is familiar with Java?
- Who is familiar with Python?
- Who is familiar with JavaScript/Typescript?
- Who knows what a RESTful API is?
- Who has ever logged in to a server via SSH?
- Who has ever done any administration of a Linux server?
- Who is familiar with Unix/Linux/Mac OS command line tools?
- Who has done any development outside of university projects?

Examination - Portfolio

Background

- the module *Developing Distributed Systems* has 55 lecture hours
- the lecture *Distributed Systems* has 22 lecture hours

  ⇒ Distributed Systems will contribute up to **50** points (out of 120) to the final grade for the module. (Please, don't do the math.)

2 Parts

01 Presentations

02 Programming Exercise

Presentations - General Conditions

- Each person should present for 10 Minutes sharp!
- The presentations should deal in particular with the core content of the lecture and *be of a conceptual nature*.

  This means, after briefly presenting the overall purpose of the framework/technology/protocol, the architecture/the details must be presented. I. e., how errors are dealt with, which services are offered, which guarantees/security aspects are implemented, how scalability is achieved, etc.

  No promotional presentations!
- Presentations have to be in English.
- The speakers should not rotate several times during the presentation. I. e. the first speaker presents first, then the second, and so on. This is necessary for the grading.

Presentations - Submissions

Attention!

You have to upload your presentation to Moodle 36 hours before you give the presentation.

Attention!

Further Requirements and Checklist (in Ger.)

The checklist hast to be signed and uploaded together with the presentation. (As a second PDF file.)

Presentations - Available Topics

※ Hint

Students giving presentations belonging to the same block have to coordinate with each other to avoid any overlap. If you need a specific topic to be covered by another student but are not sure whether it will be presented sufficiently, create a backup slide for your presentation that covers this topic as well and mark it as a backup slide. This backup slide will not be counted towards the time limit.

Virtualization and Virtualization Platforms

01 Introduction to Virtualization & Use Cases
- What is virtualization? Historical context and motivation
- Different types/levels of virtualization
- Key use cases: server consolidation, cloud computing, development/testing, isolation
- Benefits and trade-offs

02 Hypervisors - Architecture & Types
- What is a hypervisor?
- Type 1 (bare-metal) vs Type 2 (hosted) - architectural differences
- Full virtualization vs paravirtualization approaches
- Examples and when to use each type

03 Virtual Machines - Implementation & Management
- VM structure and components (virtual hardware, guest OS)
- VM lifecycle: creation, running, pause/resume, snapshots
- VM migration (live migration concepts)
- Resource allocation and isolation

04 Containers & OS-level Virtualization
- Container concept and how it differs from VMs
- Namespaces and cgroups (conceptual)
- Container images and layering
- Use cases and comparison with VMs

05 Memory Virtualization
- The address translation problem (guest virtual → guest physical → host physical)
- Shadow page tables approach
- Hardware-assisted virtualization (EPT/NPT)
- Memory management techniques (overcommitment, ballooning)

06 Network & I/O Virtualization
- Challenges of virtualizing network and I/O devices
- Emulated vs paravirtualized devices
- SR-IOV (Single Root I/O Virtualization) and device passthrough
- Virtual NICs and network bridges
- Virtual switches and network isolation

Network Protocols

01 QUIC (only available when we have ≥ 21 students)

02 HTTP/3

03 BitTorrent Protocol (only available when we have ≥ 25 students)

Modern RPC

01 Protobuf

02 Google RPC

Web-App Security

01 SOP (Same-Origin Policy), CORS (Cross-Origin Resource Sharing) (Foundations)

02 CORP / COOP / COEP (Cross-Origin Resource/Opener/Embedder Policies) (only available when we have ≥ 23 students)

03 CSP (Content Security Policy) and SRI (Subresource Integrity)

Introduction and concrete examples how they are used/specified and help prevent attacks.

Monitoring & Debugging Distributed Systems

01 Log Aggregation with a particular focus on correlation of log entries

Leader Election

01 Bully Algorithm and/or Ring Algorithm

Quorum Systems

01 Majority voting (i. e., quorum-distributed computing)

Consensus Algorithms and Fault Tolerance

01 Consensus Fundamentals & Problem Definition

- What is consensus and why is it hard in distributed systems?
- The FLP impossibility result (conceptual understanding)
- Fault models: crash faults vs Byzantine faults
- Safety vs liveness properties
- Real-world motivation: replicated state machines, distributed databases

02 (Practical) Byzantine Fault Tolerance

- When do we need BFT?
- Modern developments
- Real-world usage

03 Paxos Family

- Basic Paxos algorithm (conceptual overview, roles: proposers, acceptors, learners)
- Why Paxos is correct but complex
- Multi-Paxos for practical systems
- Real-world usage

04 Raft - Understandable Consensus

- Motivation
- Leader election, log replication, safety
- How Raft differs from Paxos (design philosophy)

■ Real-world usage

Eventual Consistency

01 Eventual Consistency and Gossip Protocol

02 CRDTs (Conflict-free Replicated Data Types) (only available when we have ≥ 22 students)

Distributed File Systems

01 Ceph

02 HDFS (only available when we have ≥ 24 students)

Topic Assignment

The presentations need to be given in the order listed below.

Topic	Constraint	Student Name
01. Introduction to Virtualization & Use Cases		
02. Hypervisors - Architecture & Types		
03. Virtual Machines		
04. Containers & OS-level Virtualization		
05. Memory Virtualization		
06. Network & I/O Virtualization		
07. QUIC	≥ 21 students	
08. HTTP/3		
09. BitTorrent Protocol	≥ 25 students	
10. Protobuf		
11. Google RPC		
12. SOP, CORS (Foundations)		
13. CORP / COOP / COEP	≥ 23 students	
14. CSP and SRI		
15. Log Aggregation		
16. Leader Election (Bully/Ring Algorithm)		
17. Quorum Systems (Majority Voting)		
18. Consensus Fundamentals & Problem Definition		
19. Byzantine Fault Tolerance (PBFT)		
20. Paxos Family		
21. Raft - Understandable Consensus		
22. Eventual Consistency and Gossip Protocol		
23. CRDTs	≥ 22 students	
24. Ceph		
25. HDFS	≥ 24 students	

Moderator Assignment

Topic Id	Moderator Id
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

Topic Id	Moderator Id
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	

Presentations - Grading (max 25 Points)

- Presentation (max 20 Points)
- Checklist (max 1 Point)
- Moderation (max 4 Points)

A Moderator has three main tasks:

1. (Introduce the speaker) and *motivate the topic*.
2. Keep track of time and *give time warnings to the presenter* (2 minutes left, time is up); abort the presentation if necessary (-1 minute).
3. Lead a short Q&A session (2-3 questions) after the presentation; if there are no questions from the audience, *the moderator should have two to three questions*.

Programming Task (max 25 Points)

See Moodle April 7th for task description and grading details.

Lecture - Schedule

3. Mar 2026:	Lecture
17. Mar 2026:	Lecture Assignment of moderators to presentation topics
7. Apr 2026:	Presentations (~8 students)
	Programming Task Explanation
	Lecture
20. Apr 2026 (5VL):	Presentations (~8 students)
	Lecture
24. Apr 2026 (5VL):	Presentations (~8 students)
	Lecture
8. May 2026 (Event):	
	Programming Task Submission Deadline (see Moodle for details)