

Distributed Systems (DSy)

Admin

Thomas Bocek

12.02.2026

About Me

- Professor (2018-2019, 2025-...)
 - Distributed systems [\[link\]](#) and blockchains [\[link\]](#)
 - Application Architecture [\[link\]](#)
 - Focus: practical implementation
- Previous
 - Co-founder / CTO
 - [Axelra](#) AG, Tech Venture Builder



DSy Administration

- Module Description
- Website: <https://dsl.i.ost.ch/>
- Exercises / Admin Slides & Videos: Teams
 - Team code: 2tbddgw (works only for OST students)
- Lectures: pre-recorded, videos available on <https://dsl.i.ost.ch/lect/fs26/> and Youtube
- Videos and PDFs will be uploaded on Friday evening
- Youtube will only have the lectures videos, admin related videos & PDFs only via <https://dsl.i.ost.ch/lect/fs26/> (VPN) or Teams
- Same as in previous semester FS25

Kürzel:

M_DSy

Durchführungszeitraum:

FS/22-FS/27

ECTS-Credits:

4

Lernziele:

Die Studierenden sollen die grundlegenden Konzepte und Kategorien verteilter Systeme verstehen und erklären können. Sie werden in der Lage sein, Load-Balancing-Strategien, Container-Technologien und virtuelle Maschinen effektiv einzusetzen, die für die Skalierbarkeit und Effizienz moderner verteilter Systeme entscheidend sind. Zudem lernen sie, verschiedene Protokolle wie HTTP, WebSockets oder RPC in verteilten Systemen angemessen zu nutzen und sichere Authentifizierungsmechanismen zu integrieren. Ein grundlegendes Verständnis für Blockchain-Technologien, einschliesslich Bitcoin und Ethereum, wird ebenfalls vermittelt.

Ein zentraler Bestandteil der Vorlesung ist die Implementierung eines verteilten Systems von Grund auf, das in einem praktischen Projekt Anwendung findet. Dabei werden typische Herausforderungen identifiziert und durch geeignete Lösungsstrategien bewältigt, während die Studierenden auch die Vor- und Nachteile verschiedener Architekturentscheidungen abwägen.

Die genauen Themengebiete können variieren und werden in der ersten Vorlesung bekanntgegeben. Die Vorlesung wird im Voraus aufgezeichnet und jeweils vor der Vorlesungsstunde online zur Verfügung gestellt. Die Themengebiete vergangener Vorlesungen sind ebenfalls online abrufbar: <https://dsl.i.ost.ch/lect/>

Verantwortliche Person:

Prof. Dr. Bocek Thomas (BOTH)

Standort (angeboten):

Rapperswil-Jona, St.Gallen (Informatik Raster)

Empfohlene Module:

Betriebssysteme 1 (M_Bsys1, HS/19-HS/20), Software-Engineering 1 (M_SE1, HS/17-HS/20)

Zusätzlich vorausgesetzte Kenntnisse:

Englisch

(Dieses Modul wird teilweise in Englisch gehalten.)

DSy Administration

Kürzel:

DSy

Semester:

1

Plan und Lerninhalt:

Die groben Lerninhalte umfassen die Einführung in verteilte Systeme, einschliesslich ihrer grundlegenden Konzepte und Kategorien. Ein zentraler Bestandteil sind die praktischen Übungen, in denen die Studierenden eine einfache verteilte Applikation implementieren müssen. Die genauen Lerninhalte können variieren und werden in der ersten Vorlesung bekanntgegeben. Lerninhalte vergangener Vorlesungen sind online abrufbar: <https://dsl.i.ost.ch/lect/>.

- If there is interest in further topics (distributed systems and blockchain related), please talk to me.

Nr	Date	Lecture*	Lecturer
01	17.02.2026	Admin, Introduction / Motivation, part 1	Bocek
02	23.02.2026	Introduction / Motivation, part 2	Bocek
03	02.03.2026	Monorepos / Polyrepos, Containers and VMs	Bocek
04	09.03.2026	Docker, Debugging Containers	Bocek
05	16.03.2026	Load Balancing	Bocek
06	23.03.2026	Web Architectures	Bocek
07	30.03.2026	Categorization	Bocek
-	-	-	-
08	13.04.2026	Authentication	Bocek
09	20.04.2026	Protocols	Bocek
10	27.04.2026	Application Protocols	Bocek
11	04.05.2026	Exam preparation	Bocek
12	11.05.2026	Deployment, Performance, Blockchain, Bitcoin	Bocek
13	18.05.2026	Challenge Task Presentations	Bocek
-	-	-	you
14	01.06.2026	Challenge Task Award Winner Announcement, Q&A	Bocek

* topics may change

DSy Exercises

- Challenge Task
 - Goal: practical application of distributed systems concepts and tools
 - Change: **individual work**
 - With availability of generative AI tools, a single student can now design, build, and implement what previously required a group of 2-3
- Offline: Rapperswil
 - Monday afternoon: 13:10 – 18:40 / 1.257
- Offline: St.Gallen
 - Monday morning: 08:10 – 09:50 / FZ.0332
- Exercises: work on challenge task
 - Monday: all day available
- Online: via MS Teams (on request)

In den Übungen arbeiten die Studierenden in Gruppen an einer verteilten Applikation (Challenge Task DSy), die verschiedene Anforderungen erfüllen muss. Das erfolgreiche Umsetzen dieser Anforderungen ist Voraussetzung für das Bestehen der Übungen und die Zulassung zur Prüfung. Die Studierenden haben während der Übungszeit entweder vor Ort oder von Zuhause die Möglichkeit, an der Challenge Task zu arbeiten. Beispiele aus früheren Semestern sind ebenfalls online verfügbar: <https://dsl.i.ost.ch/lect/>.

DSy Challenge Task

- Leistungsbewertung

Während der Prüfungssession:

Schriftliche Prüfung, 60 Minuten

Zulassungsbedingungen zur Prüfung:

Nach Ankündigung, Abgabe erfolgreich gelöste Übungsaufgaben.

- Simple Frontend (e.g., HTML, Vue, React, Svelte)
- Loadbalancer(e.g., traefik, nginx, HAproxy, Caddy)
- Two instances of a service (your choice), and during the challenge task presentation, one instance will be shut off.
- Database (persistent storage, does not need to be scalable)

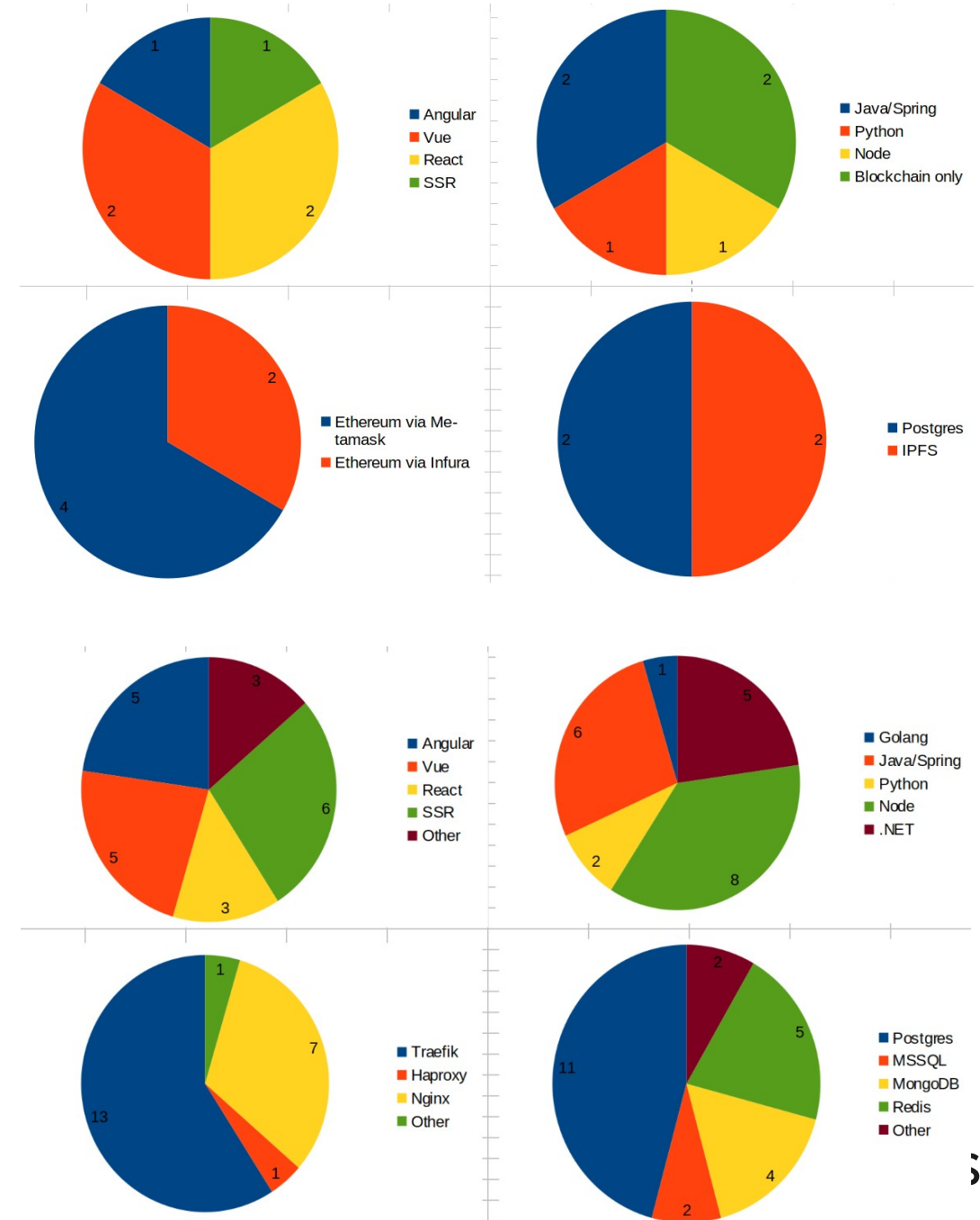
- Challenge Task FS 2026
 - This semester's challenge task (CT) is the design and implementation of a distributed system (of your choice) with JWT authentication where a service instance can fail.
- Requirements
 - Load balancing
 - Dockerized
 - Simple frontend
 - JWT-based authentication
 - Persistent storage
 - Use latest stable releases of chosen libraries and frameworks
 - Single command startup
 - Performance analysis

Milestones

- 02.03.2026 first hand-in of your initial plan and project idea. Invitation to github/gitlab repository
- 30.03.2026 second hand-in of your initial commit of your challenge task and current progress (initial state, does not need to run)
- 04.05.2026 third hand-in of your current progress (what works, what is still open)
- 18.05.2026: Final hand-in / presentation
 - You are expected to understand and explain your code
 - Presentations will take place during the week starting 18.05.2026, with slots available throughout the week in both St.Gallen and Rapperswil for higher number of presentations
 - You should show the architecture, components, and design decisions in 5 min, a demo in 5 min, and the Q&A will be 5 min
 - After your presentation, you need to hand in your PDF presentation

DSy Challenge Task

- Final hand-in
 - Source code you understand
 - Presentation (slides) and demo of the application, also showing the architecture and design decisions
- Libraries and tools
 - Allowed to use any language, framework, and platforms. However, the supervisors are familiar with those: Java, Golang, JavaScript, Linux
- If you **fail** the CT, you **fail** this course



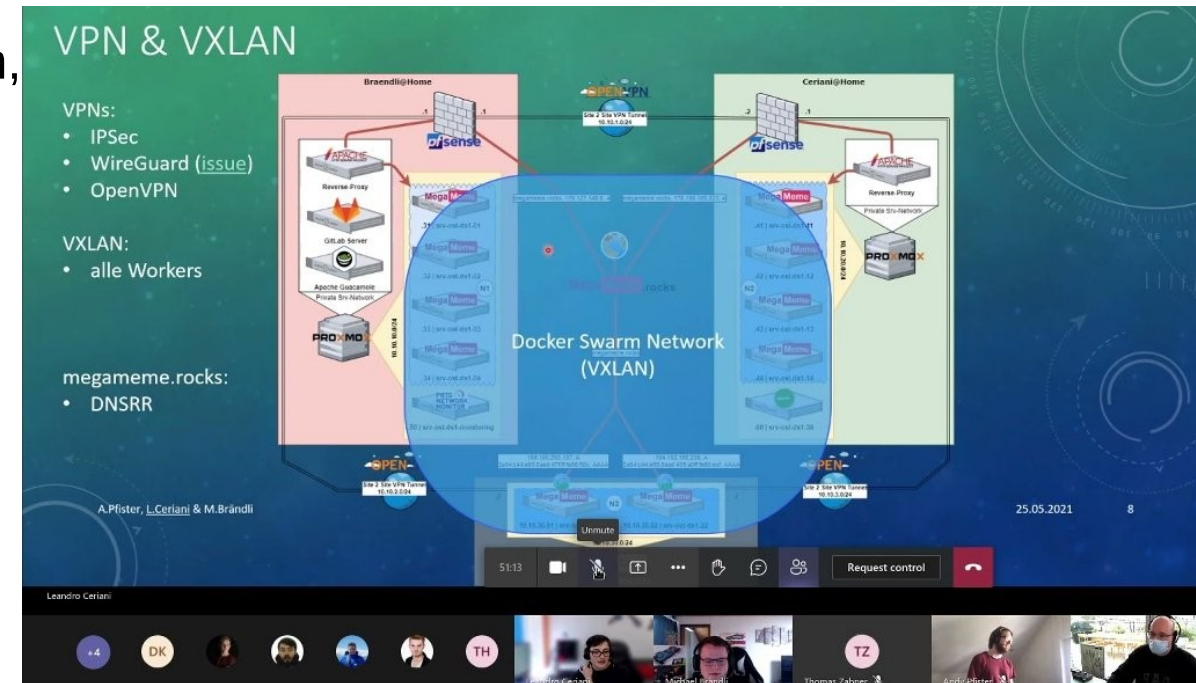
DS1/VSS Challenge Task FS 2020

- Free choice project
- Requirements
 - Use at least one text protocol, at least one binary protocol
 - Load balancing
 - Scalable service with authentication
- Each winner: 5.27mBTC (0.00527BTC)
 - They gathered weather data from multiple locations, and they used many components: ActiveMQ, Eureka discovery services, Kibana, Docker compose, running on self-hosted gitlab.



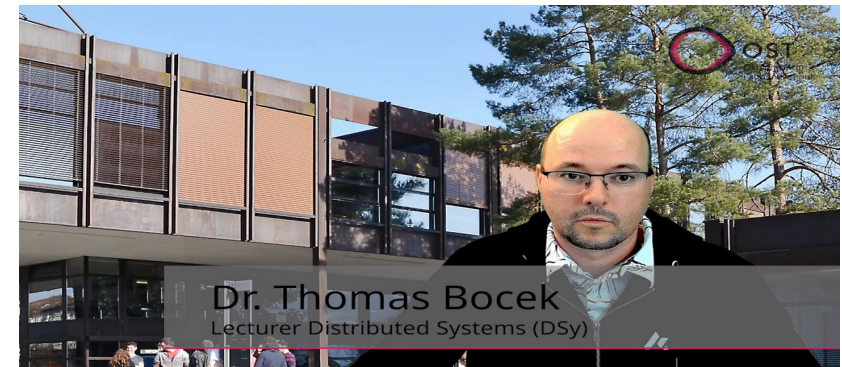
DS1 Challenge Task FS 2021

- This semester's challenge task (CT) is the design and implementation of a simple distributed system, with the following components:
 - Simple Frontend (e.g., HTML, Vue, React, Svelte)
 - Loadbalancer(e.g., traefik, nginx, HAproxy, Caddy)
 - Two instances of a service (your choice)
 - One storage backend (Postgres, simple KV memory storage)
 - JWT Authentication
- Each winner: 1.35 mBTC (0.00135 BTC)
 - Voting meme platform with Svelte and Ruby on Rails with docker swarm, GlusterFS, and CockroachDB



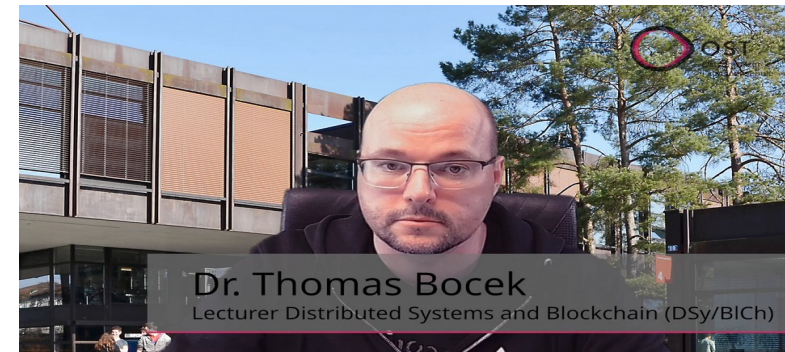
DSy Challenge Task FS 2022

- This semester's challenge task (CT) is the design and implementation of a simple distributed system (of your choice), with at least one service with a websocket, and where a service instance can fail.
- Requirements
 - Load balancing with scalable service
 - Failover of a service instance
 - A websocket needs to send data from a service to the frontend. You need to implement (consume) a websocket (using e.g., Blazor that integrates websockets does not fulfill this requirement). You can produce websocket data in your backend or use 3rd party websocket services (e.g., crypto market data)
 - Dockerized
 - Persistent storage (storage does not need to be scalable, but you can build it scalable if you want)
 - Use latest stable releases of chosen libraries and frameworks
- Amy Gdala, a web application to memorize information by talking about your day (writing a diary) and to search for all your stored information with natural language queries. Their techstack included GPT-3, Kibana/Elastic Search, Coqui-STT/TTS, Python for micro-services, Kubernetes, and Istio for gRPC load balancing.



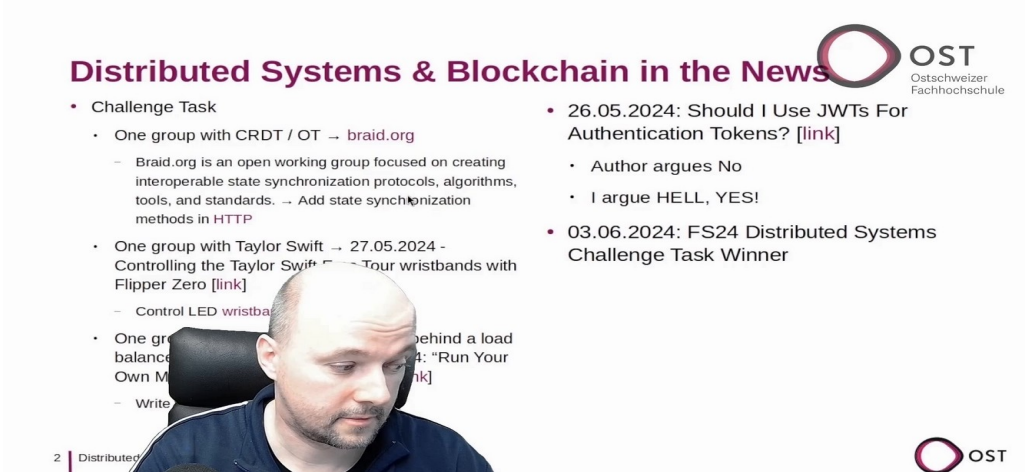
DSy Challenge Task FS 2023

- This semester's challenge task (CT) is the design and implementation of a simple distributed system (of your choice) where a service instance can fail.
 - 1) Load balancing with scalable service
 - 2) Failover of a service instance
 - 3) Dockerized and Frontend connects via HTTP/3
 - 4) Persistent storage (storage does not need to be scalable, but you can build it scalable if you want)
 - 5) Use latest stable releases of chosen libraries and frameworks
- Winners built a web application, designed to share collections of pictures, featured automated tagging through PhotoPrism. For storage, they used ScyllaDB, a high-performance, fault-tolerant database compatible with Cassandra. Everything is self-hosted with their own Kubernetes cluster using MetalLB.



DSy Challenge Task FS 2024

- This semester's challenge task (CT) is the design and implementation of a simple distributed system (of your choice) where a service instance can fail.
 - Load balancing with multiple instances with failover of a service instance
 - Dockerized
 - Simple frontend
 - Scheduled task
 - Persistent storage
 - Use latest stable releases of chosen libraries and frameworks
- Winners built Co-Puzzle Web App, which is a collaborative puzzle using websockets. For the load balancer they used Traefik, everything dockerized of course, for the frontend/backend they used React with next, and for the database, they used MongoDB.



Distributed Systems & Blockchain in the News OST
Ostschweizer Fachhochschule

- Challenge Task
 - One group with CRDT / OT → braid.org
 - Braid.org is an open working group focused on creating interoperable state synchronization protocols, algorithms, tools, and standards. → Add state synchronization methods in HTTP
 - One group with Taylor Swift → 27.05.2024 - Controlling the Taylor Swift → Four wristbands with Flipper Zero [link]
 - Control LED wristbands
 - One group behind a load balancer → 24: "Run Your Own M... [link]
 - Write
- 26.05.2024: Should I Use JWTs For Authentication Tokens? [link]
 - Author argues No
 - I argue HELL, YES!
- 03.06.2024: FS24 Distributed Systems Challenge Task Winner

2 | Distributed

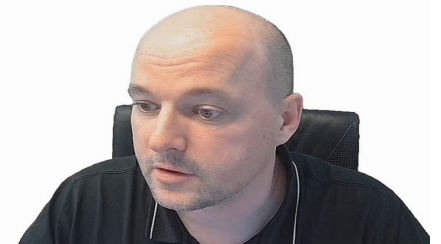
DSy Challenge Task FS 2025

- This semester's challenge task (CT) is the design and implementation of a distributed system (of your choice) with Server-Sent Events (SSE) where a service instance can fail.
- Load balancing with multiple instances with failover of a service instance
- Dockerized
- Simple frontend
- Server-Sent Events from the server to the client
- Persistent storage
- Use latest stable releases of chosen libraries and frameworks
- Winners built EduCDN, an educational content delivery network. The system provides hands-on experience in CDN concepts including L3 load balancing with BGP anycast, HTTP caching, and real-time monitoring through a dashboard.

Distributed Systems & Blockchain in the News



- 24.05.2025: Reinvent the Wheel [\[link\]](#)
 - Challenge Task FS25, admin lecture 1: Trying to understand it theoretically is not enough – hands-on is key!
 - "Reinvent for insight. Reuse for impact."
- 25.05.2025: Pakistan allocates 2,000MW power for Bitcoin mining and AI centers [\[link\]](#)
 - Cryptocurrency Adoption [\[link\]](#), Switzerland: 55
 - Bitcoin Energy Consumption Index [\[link\]](#)



DSy Recommendation

- Follow the lecture closely
 - Lecture slides contain relevant
 - Reading further literature is recommended
 - Link are provided in the slides
- Post-processing the lecture
 - If topics or statements from the lecture are not clear, research and resolve them
- Actively participate in the challenge task
 - Trying to understand it theoretically is not enough – hands-on is key!
 - Do not miss the opportunity of discussing details with your supervisor
- 12 lectures
 - 24h, ca. 2h / week
 - ~1.5h recordings
 - ~1/2 h post-processing
- 13 exercises / challenge task
 - (78h, ca. 6h / week)
 - ~6h challenge task
- Exam
 - ~18h exam preparation
- 4 ECTS = 120h
- Prüfung:
 - Closed book 60min, no coding, Hilfsmittel: none

Summary

- Lectures Videos online Friday evening
- Challenge task ideas: send to thomas.bocek@ost.ch
 - Individual work on challenge task during exercise and/or home
- 8 requirements to be met
 - 1) Load balancing, 2) Dockerized, 3) Simple frontend, 4) JWT Authentication, 5) Persistent storage, 6) Use latest stable releases of chosen libraries and frameworks, 7) Single command setup, 8) Perform load test
- All 8 requirements must be fulfilled to pass this course

- 4 Milestones

- 02.03.2026 - 1st hand-in of your initial plan and project idea. Invitation to github/gitlab repository
- 30.03.2026 - 2nd hand-in of your initial commit of your challenge task and current progress (initial state, does not need to run)
- 04.05.2026 - 3rd hand-in of your current progress (what works, what is still open)
- **Week of 18.05.2026** - final hand-in, presentation week

