

Distributed Systems (DSy)

Admin

Thomas Bocek

14.02.2023

About Me

- OST, Lecturer
 - Distributed systems and blockchains
 - Previously (VSS/DSA/DS1/DS2,WED1,3)
- Co-founder **Axelra AG**
 - May 2019
 - Tech Delivery, MVP in 100 days - Digitalization / Blockchain
 - E.g., **THORWallet** (DeFi app for cross chain assets)
 - CTO, operative
- General interests: distributed systems, blockchain, startups



DSy Administration

- Module Description
- Website: <https://dsl.i.ost.ch/>
- Exercises: Teams
 - Team code: 2tbddgw (works only for OST students)
- Lectures: pre-recorded, videos available on <https://dsl.i.ost.ch/lect/fs23/> and Youtube
 - Videos and PDFs will be uploaded 1 week before the lecture
 - Youtube will only have the lectures videos, admin related videos & PDFs only via <https://dsl.i.ost.ch/lect/fs23/> (VPN)
 - Same as in previous semester FS22

Distributed Systems

Kurzzeichen:

M_DSy

Durchführungszeitraum:

FS/22-FS/23

ECTS-Punkte:

4

Lernziele:

Verteilte Systeme sind allgegenwärtig in beruflichem und privatem Alltag; sie führen Konzepte und Technologien aus unterschiedlichen Bereichen der Informatik zusammen. Die Themengebiete umfassen:

- Protocols for Distributed Systems
- Fully Distributed Systems and NAT issues
- BitTorrent, Tor, Rsync
- RPC and Queues
- Blockchain Introduction
- Smart Contracts and Cryptocurrencies

Das Lernziel in diesem Modul ist, dass Studenten ein verteiltes System selbständig entwickeln können.

Verantwortliche Person:

Bocek Thomas

Standort (angeboten):

OST Rapperswil

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

Kurzzeichen:

DSy

Lernziele:

Plan und Lerninhalt:

Die Themengebiete können variieren und werden in der ersten Vorlesung bekanntgegeben.

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

Nr	Date	Lecture*	Lecturer
01	21.02.2023	Admin, Introduction / Motivation	Bocek
02	28.02.2023	Introduction / Categorization	Bocek
03	06.03.2023	Distributed Applications: Tor, WebSockets, Protocols 1	Bocek
-	14.03.2023	-	-
04	21.03.2023	Monorepos / Polyrepos, Protocols 2	Bocek
05	28.03.2023	Containers and VMs	Bocek
06	04.04.2023	Load Balancing	Bocek
-	11.04.2023	-	-
07	18.04.2023	Application Protocols and RPC, Deployment (Docker Swarm, Kubernetes)	Bocek
08	25.04.2023	Classic Consensus, CRDT, Deployment (Docker Swarm, Kubernetes)	Bocek
09	02.05.2023	Guest lecture	tba
10	09.05.2023	Blockchain, Bitcoin	Bocek
11	16.05.2023	Exam preparation, Ethereum Introduction, Smart Contracts	Bocek
12	23.05.2023	Challenge Task Presentations I	you
13	30.05.2023	Challenge Task Presentations II	you
14	06.06.2023	Challenge Task Award Winner Announcement, Q&A	Bocek

* topics may change

DSy Exercises

- Challenge Task
 - Starts 28.02.2023 (lecture 2)
 - **No exercises in lecture 1!**
- Goal: practical application of distributed systems concepts and tools
- Online: via MS Teams
 - Tuesday exercise u14: 10:10 – 11:50 / 1.211
 - Tuesday exercise u11-u13: 13:10 – 18:40 / 1.252
 - Work on challenge task
 - You can come in at any time – remote call possible as well

Plan und Lerninhalt:

Die Themengebiete können variieren und werden in der ersten Vorlesung bekanntgegeben.

In den Übungen entwickeln Studentengruppen eine einfache verteilte Applikation (Challenge Task I), welche diverse Requirements erfüllen muss. Das Erfüllen dieser Requirements ist Voraussetzung für das Bestehen der Übungen und für die Zulassung zur Prüfung. Ein Beispiel eines Challenge Tasks könnte ein vollständig verteiltes Tic-tac-toe sein.

BlCh 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 / RP (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.
- Dockerized
- Use of HTTP/3
- Database (persistent storage)

• Challenge Task FS 2023

- Design and implementation of a simple distributed system (of your choice), with at least one service storing data in a database, and where a service instance can fail

• Requirements

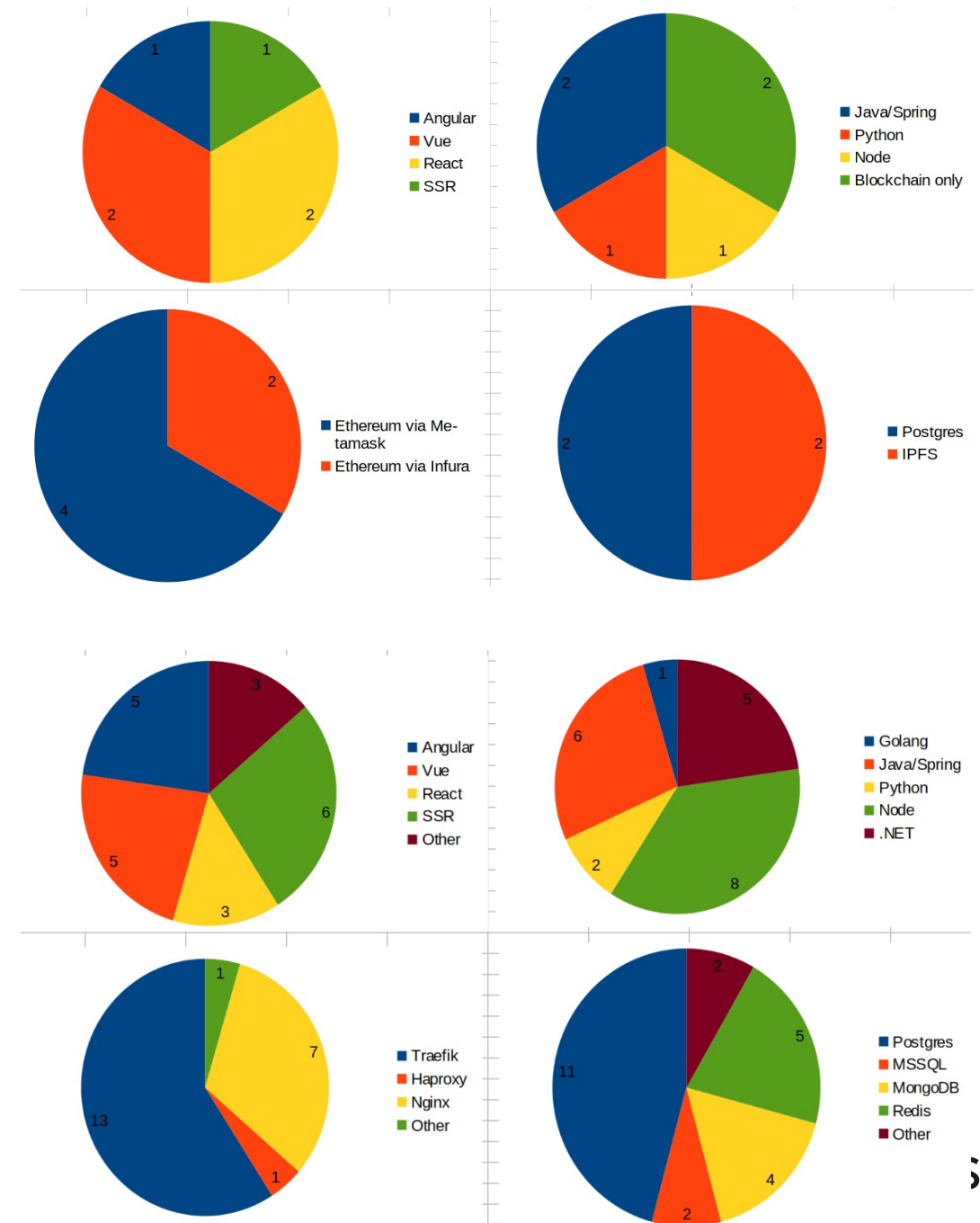
- Load balancing with scalable service
- Failover of a service instance
- Dockerized and Frontend connects via HTTP/3
- Persistent storage (storage does not need to be scalable, but you can build it scalable if you want)
- Use latest stable releases of chosen libraries and frameworks

Nr	Date	Lecture*	Suggested Milestone
01	21.02.2023	Admin, Introduction / Motivation	Group forming, idea for app
02	28.02.2023	Introduction / Categorization	Setup dev environment
03	06.03.2023	Distributed Applications: Tor, WebSockets, Protocols 1	Information flow
-	14.03.2023	-	
04	21.03.2023	Monorepos / Polyrepos, Protocols 2	
05	28.03.2023	Containers and VMs	
06	04.04.2023	Load Balancing	
-	11.04.2023	-	
07	18.04.2023	Application Protocols and RPC, Deployment (Docker Swarm, Kubernetes)	Hand-in 1 initial (17.04.2023, 23:59 CET)
08	25.04.2023	Classic Consensus, CRDT, Deployment (Docker Swarm, Kubernetes)	
09	02.05.2023	Guest lecture	
10	09.05.2023	Blockchain, Bitcoin	Connecting everything
11	16.05.2023	Exam preparation, Ethereum Introduction, Smart Contracts	Testing, testing, testing
12	23.05.2023	Challenge Task Presentations I	Hand-in 2 final (22.05.2023, 23:59 CET)
13	30.05.2023	Challenge Task Presentations II	
14	06.06.2023	Challenge Task Award Winner Announcement, Q&A	

* topics may change

DSy Challenge Task

- Hand-in 1: 17.04.2022, 23:59 (CET) - initial version of your project
- Final hand-in: 22.05.2022, 23:59 (CET)
 - Well documented code and infrastructure
 - 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 (see FS22, HS22)
- If you **fail** the CT, you **fail** this course



DSA Challenge Task 2018

- Design and implementation of a fully-distributed, P2P messaging application with an integrated, blockchain-based notary service
- Requirements
 - Full decentralization and P2P
 - Send messages between peers
 - Quickly and easily verify messages
- Each winner: 0.01311 BTC (50CHF)
 - They used the IOTA blockchain as a signaling layer for WebRTC.
 - Everything decentralized



DSA Challenge Task 2019

- Design and implementation of a fully-distributed, P2P gambling game
- Requirements:
 - Fully decentralization and P2P communication
 - Waiting room with chat
 - Fully decentralized gambling
- Each winner: 0.0075 BTC (~50CHF)
 - Simple yet efficient design
 - Blockchain-based four-in-a-row
 - Technology: Ethereum with web3.js and WebRTC



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.



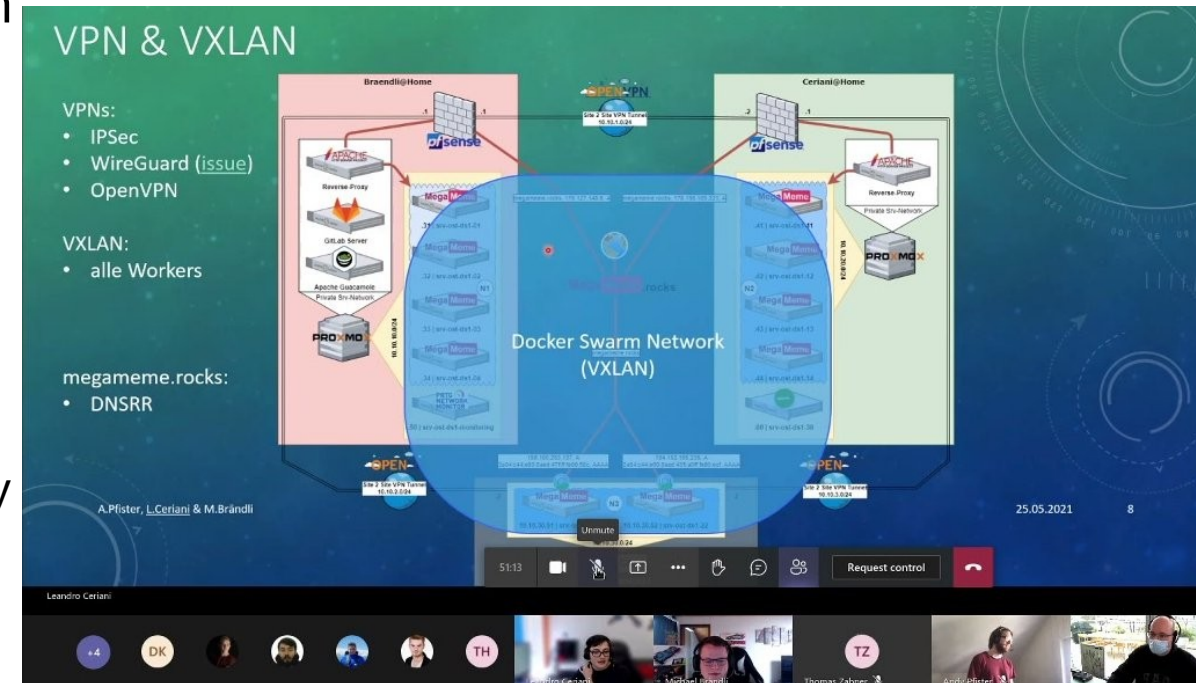
ADSB Challenge Task HS 2020

- Design and implementation of a decentralized financial application
- Requirements
 - Full decentralization and P2P mechanisms for the financial product (e.g., WebRTC, TomP2P, libp2p) or use scalable mechanisms ready to be deployed on the cloud
 - Graphical (GUI/Web-based) frontend
 - Public blockchain based
- Each winner: 2.625 mBTC (0.002625 BTC)
 - Donation platform and used the blockchain to keep track of donations



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

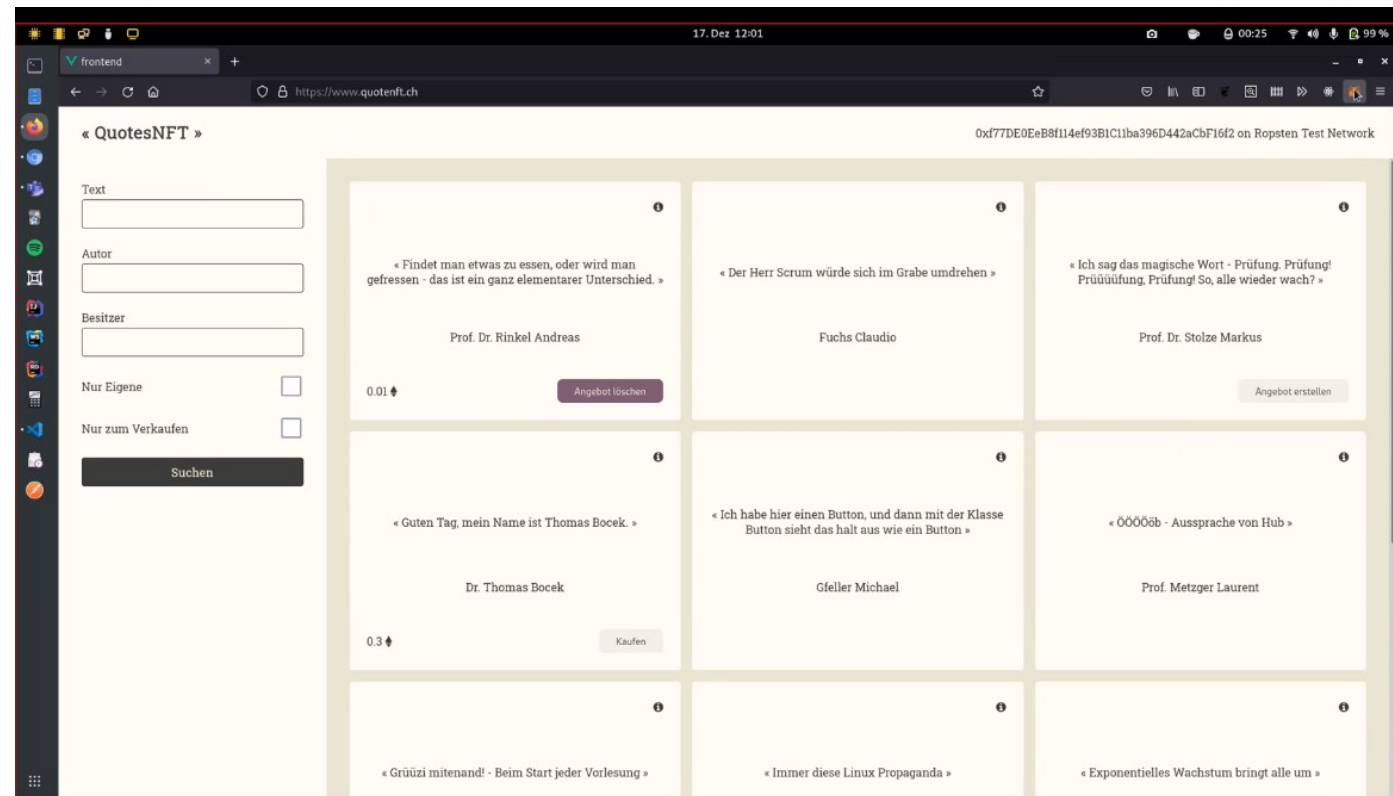


BlCh Challenge Task HS 2021

- This semester's challenge task (CT) is the design and implementation of an application including NFTs. Ideally, the application is a decentralized application (Dapp). Requirements:

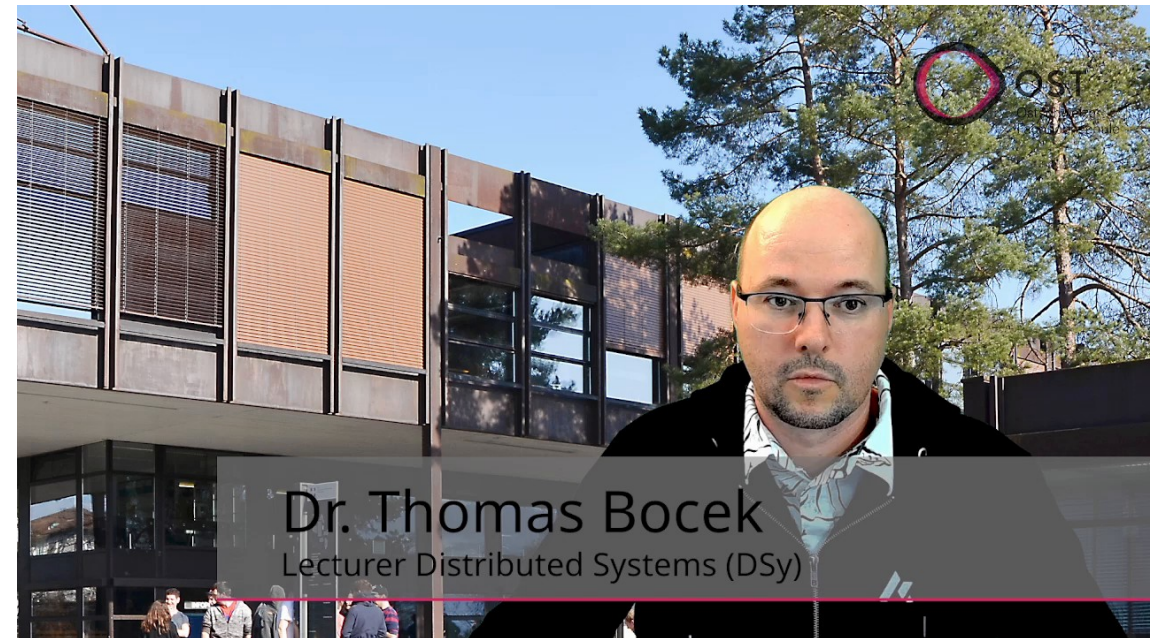
- A working prototype for your use-case
- Use latest stable releases of chosen libraries and frameworks
- An NFT stored on a public blockchain (can be testnet)
- Transfer of an NFT triggered directly or indirectly via the frontend/backend (can be testnet)
- Each winner: 1 mBTC (0.001 BTC)

- QuoteNFT is a platform to trade NFT quotes. The trading happens on a smart contract and is fully decentralized. Their project uses Vue.js for the frontend, node and express with TypeScript for the backend, and MetaMask to connect to Ethereum. Their QuoteNFT platform uses Traefik as a reverse proxy and docker-compose to orchestrate the services



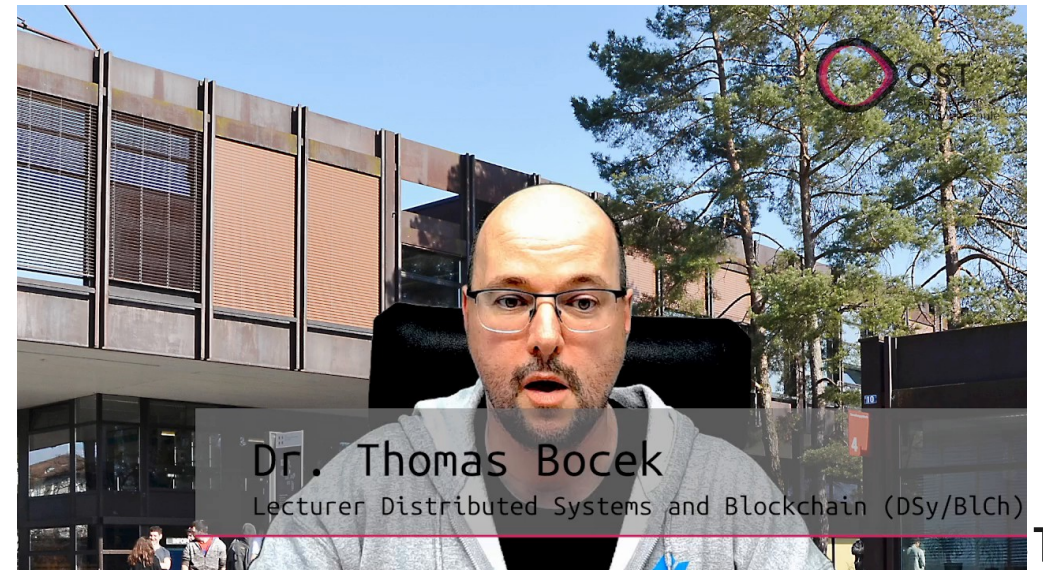
DSy Challenge Task FS 2022

- 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.



BlCh Challenge Task HS 2022

- This semester's challenge task (CT) is the design and implementation of an onchain voting application. Ideally, the application is a decentralized application (Dapp).
- Requirements
 - A working prototype for your use-case
 - Use latest stable releases of chosen libraries and frameworks
 - Submission of a voting proposal, with previously known consequences and recipients (can be testnet)
 - Onchain voting on proposal with at least 2 participants / identities (can be testnet)
 - Successful vote needs to trigger something onchain. Can be an ERC20 or an NFT transfer (can be testnet)
 - Voting status and process need to be shown in the frontend
 - The solution may use existing libraries and code, but those must open software software
- Winners built a decentralized autonomous organization (DAO), using soul-bound tokens (SBT) to vote on documents and its changes. They used Solidity with the OpenZeppelin Governor, for the backend next.js, PostgreSQL to store metadata, and to connect to the Ethereum, they used ethers.js.



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!
 - If its not working, **read logs**, try to figure it out by yourself. If its still not working, ask
 - Do not miss the opportunity of discussing details with your supervisor
- 10 lectures
 - (25h = 20h + 5h; ca. 2.5h / week)
 - ~2h frontal teaching
 - ~1/2 h post-processing
- 12 exercises / challenge task
 - (72h = 60h + 12h ca. 6h / week)
 - ~5h challenge task
 - ~1h post-processing
- Exam
 - ~23h exam preparation
- 4 ECTS = 120h
- Prüfung:
 - Closed book 60min, no coding, Hilfsmittel: none

Summary

- Lectures Videos online one week before
- Challenge task, group members: send to thomas.bocek@ost.ch until next week
 - Work on challenge task during exercise hours and at home
- 6 requirements to be met
 - Load balancing with scalable service
 - Failover of a service instance
 - HTTP/3
 - 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
- All 6 requirements must be fulfilled to pass this course
- Deliverables
 - 17.04.2023, 23:59 (CET) first hand-in of your initial version
 - 22.05.2023, 23:59 (CET) - well documented infrastructure, source code, and presentation
- Presentations during on the 23.05.2023 and 30.05.2023 on-site
- OST DSy Challenge Task Winner FS 2023
 - 06.06.2023

