

# Distributed Systems

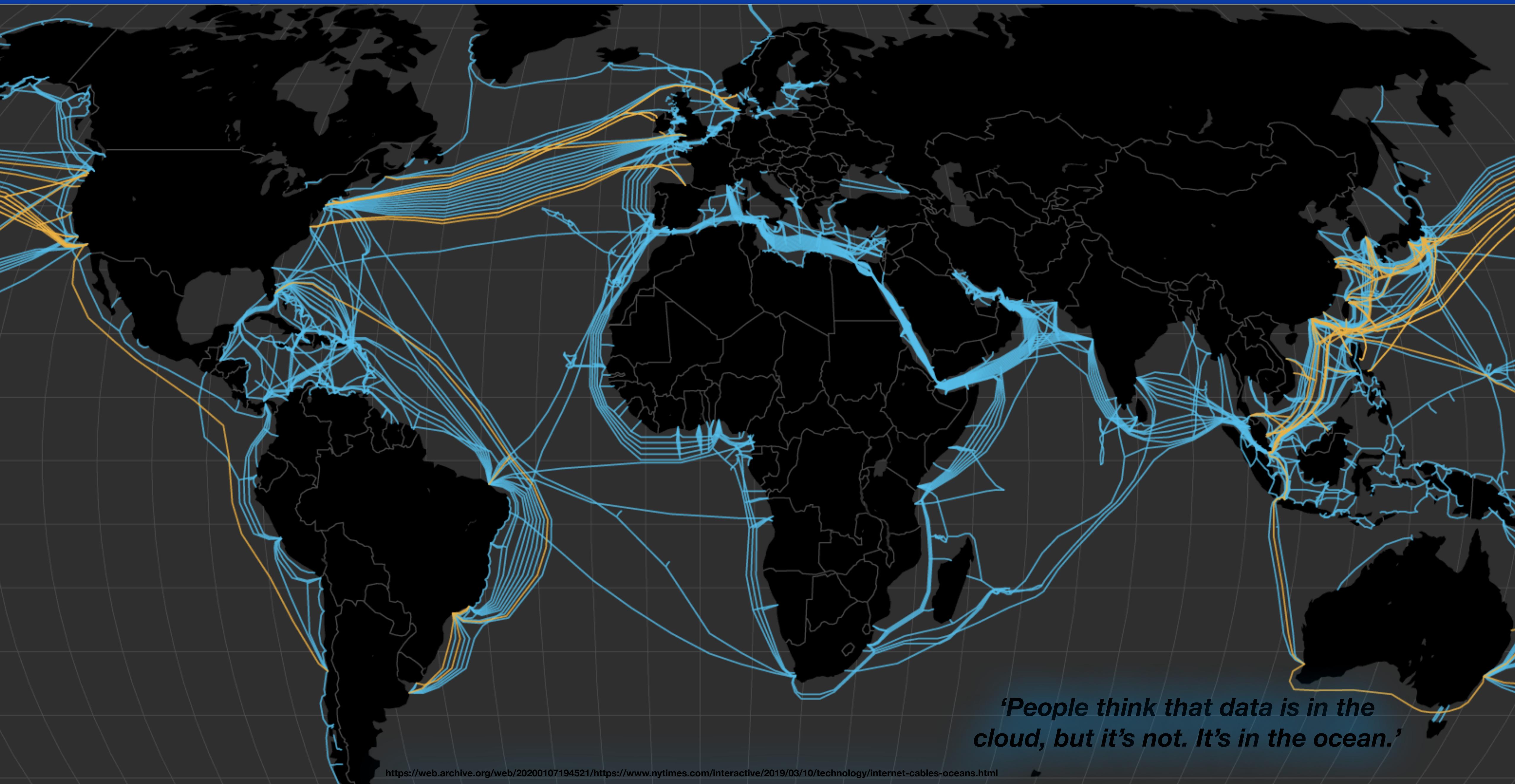
## DAT520 - Spring 2025

Course Information

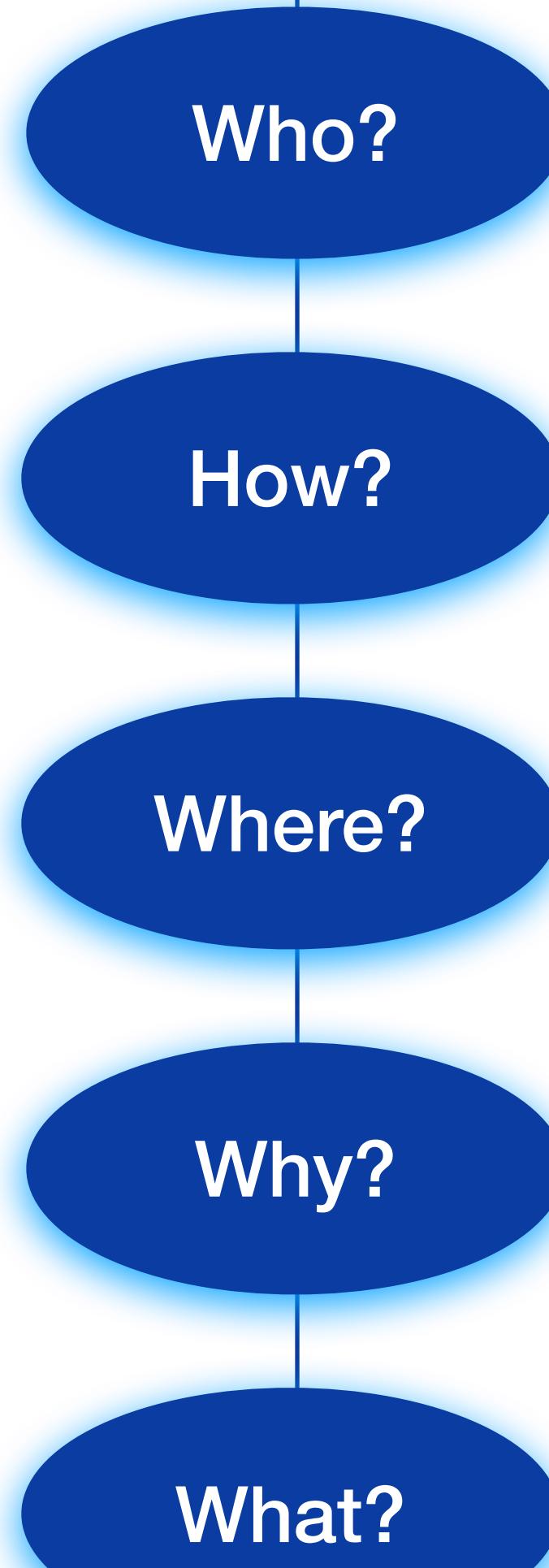
Prof. Hein Meling



# Internet Cables



*'People think that data is in the cloud, but it's not. It's in the ocean.'*



About us

Logistics and Policies...

Study Resources

Intro to Distributed Systems (DS)

Syllabus - Main topics

# About us

## Course Instructor: Prof. Hein Meling

- hein.meling@uis.no
  - Don't send messages on Canvas
- Office: KE D427 (no office hours)
  - Talk to me between lectures
  - Come by the office — I may be busy ...
  - Send email to book an appointment
- Research interests: Distributed Systems
  - Fault tolerance, availability, reliable storage, blockchain systems, etc...

## Teaching Assistants

- Daniel Osmundsen Dirdal, PhD student
- Magnus Tysdal, Master student
- Jesper Søberg, Master student



## {Resilient | Reliable} Systems Lab



2018-2023

**BBChain** (UiS & UiT): An Efficient Trustworthy Distributed Document Verification System

2019-2022

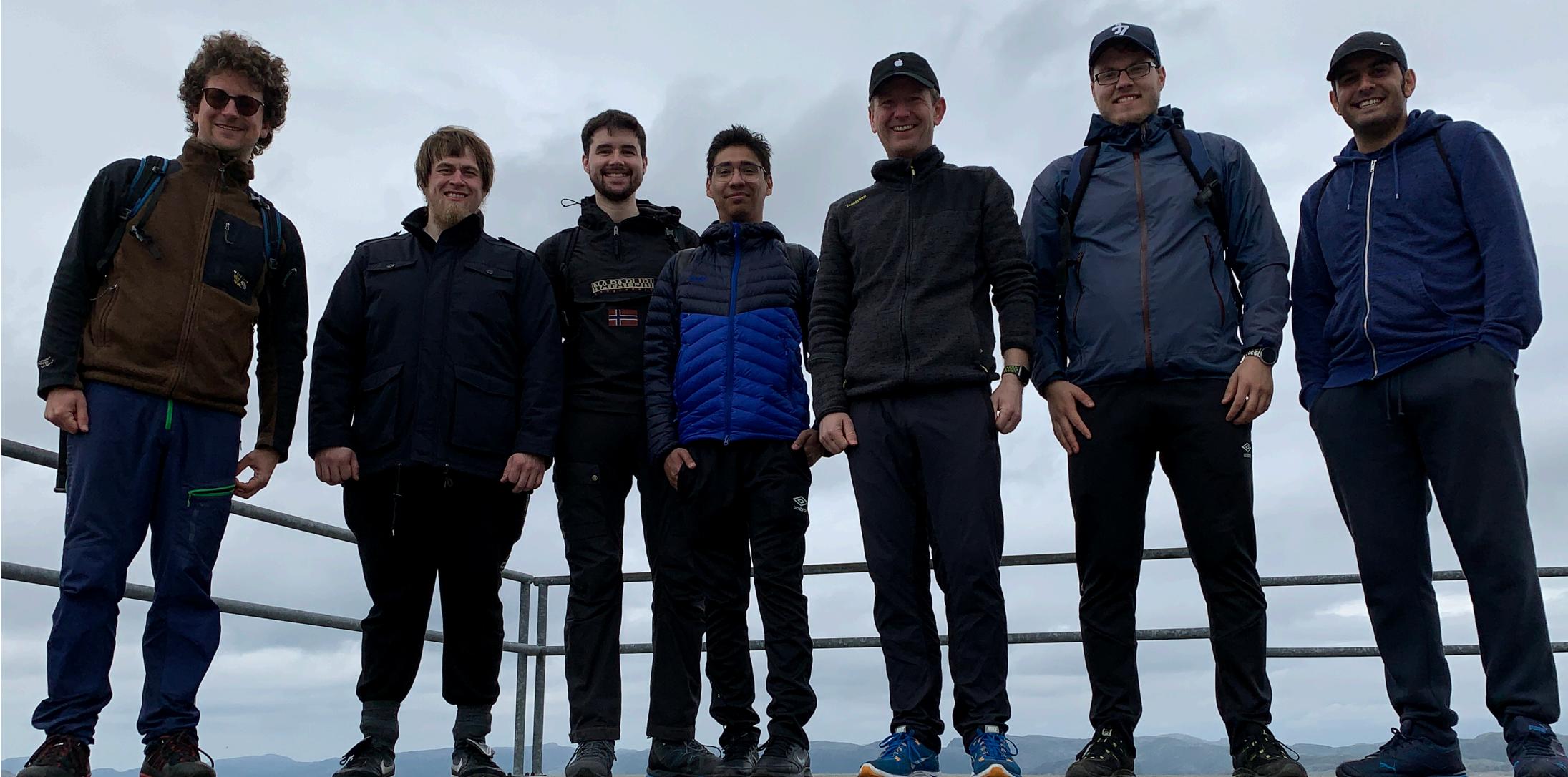
**Credence** (USA, Canada, Brazil, and Norway research institutions): Collaboration Network for Excellent Education and Research in Dependable and Secure Distributed Systems

2021-2022

Leander Sabbatical: Technische Universität Braunschweig

2022-2023

Hein Sabbatical: University of California, Berkeley



Who are you?

- What's your study program?
  - Computer Science
  - Data Science
  - Others?
- Who is working as a developer?
  - What do you do?
- Experience:
  - Developing distributed systems?
  - Go/Rust/Java/C# programming?
  - Only Python?

# Learning Outcomes

# Knowledge

*Be familiar with*

- principles for design and implementation of distributed systems.
- system architectures for distributed systems.
- techniques for solving various problems in distributed systems.

Lectures

Lectures

Lectures

## Skills

*Be able to*

- develop advanced distributed applications with fault tolerance properties.
- perform independent research in distributed systems.
- reason about problems that involve distributed components.

Labs

Master thesis

Exam

## General competency

*Know how to*

- develop distributed computer systems.

Work

# Logistics and Policies

## Recommended prerequisites

- Operating Systems
- Databases
- Computer Networking
- Advanced Programming
- Discrete Mathematics (Logic operators)
  - For all, existence, set operators, ...

## GitHub

[github.com/dat520-2025/info](https://github.com/dat520-2025/info)

- Course information
- Syllabus
- Lecture Plan
- Lab project information
- Reading material (links)

## Discord (Helpbot)

- Announcements
- Get help during labs
- Queue management
- Lab approval

<https://discord.gg/9RAXztAY>

*Register with Helpbot using your GitHub username: /register*

**PS: No Canvas!**

## QuickFeed

- Submit assignments to GitHub
- Tested by QuickFeed
- View results of lab
- Lab approval

<https://uis.itest.run/>

*Sign up using your  
GitHub account*

*Enroll in DAT520*

*Consider creating a separate  
GitHub account for course work*

# Weekly Schedule

Monday 6/1	Tuesday 7/1	Wednesday 8/1	Thursday 9/1
		<b>Laboratorium</b> 08:15 - 10:00 H. Meling KE E-456	
<b>Laboratorium</b> 14:15 - 16:00 H. Meling KE E-456		<b>Forelesning</b> 14:15 - 16:00 H. Meling KE E-166	<b>Forelesning</b> 14:15 - 16:00 H. Meling KE E-456
			<b>Laboratorium</b> 16:15 - 18:00 H. Meling KE E-456

# Lecture Plan Spring 2025

Week	Month	Wednesday (14:15 - 16:00)	Thursday (14:15 - 16:00)	Comment
2	Jan	Course Introduction	Ch 1 Introduction	
3	Jan	Live coding: Go / gRPC	Ch 2 Basic Abstractions	Bring laptop
4	Jan	Live coding: Gorums	Ch 2 Basic Abstractions	Bring laptop
5	Jan	Ch 2 Basic Abstractions	Ch 3 Reliable Broadcast	
6	Feb	Ch 3 Reliable Broadcast	Paxos	
7	Feb	Paxos	Ch 5 Consensus	
8	Feb	Ch 6 Consensus Variants	Ch 6 Consensus Variants	
9	Feb	Ch 4 Shared Memory	Ch 4 Shared Memory	
10	Mar	Consistent Global States	Consistent Global States	
11	Mar	Consistent Global States	Consistent Global States	
12	Mar	Guest Lecture		
13	Mar	No Lecture (Only Lab)	No Lecture (Only Lab)	
1	Apr			
18	Apr	No Lecture (Only Lab)	No Lecture (Only Lab)	
22	Jun	Written exam: TBD		

“I love deadlines. I like the whooshing sound they make as they fly by.”

— Douglas Adams, *The Salmon of Doubt*

# Lab Assignments

Lab	Topic	Grading	Approval	Submission	Deadline
1	<a href="#"><u>Getting Started with Go and Network Programming</u></a>	Pass/fail	Automatic	Individually	January 24
2	<a href="#"><u>Network Programming with gRPC and Gorums</u></a>	Pass/fail	TA Approval	Individually	February 6
3	<a href="#"><u>Failure Detector and Leader Election</u></a>	Pass/fail	TA Approval	Group	February 20
4	<a href="#"><u>Single-decree Paxos and Multi-Paxos</u></a>	Pass/fail	TA Approval	Group	March 6
5	<a href="#"><u>Multi-Paxos with Gorums and Performance Evaluation</u></a>	Pass/fail	TA Approval	Group	March 27
6	<a href="#"><u>Bank Application with Reconfiguration</u></a>	Pass/fail	TA Approval	Group	April 25

## Deadlines and Slip days

[dat520-2025/info/policy.md](#)

- If you cannot make a deadline, you can use up to **7 slip days** throughout the semester, including weekends and holidays.
- It is the **date on your lab's submission**, as viewed on GitHub that counts towards the slip days.
- Slip days **cannot be used to extend** the final submission deadline.
- All submissions must be approved before the last deadline.

## Collaboration

[dat520-2025/info/policy.md](#)

- Talk to each other or the teaching staff about any of the assignments. Assistance is limited to general discussion of the problem. Each student/group must write out their own solutions.
- Consulting another student's/group's solution is prohibited. Submitted solutions must not be copied from any source.
- You **shall not** supply your work to other students in future instances of this course.



## Generative Models

[dat520-2024/info/policy.md](#)

- May use generative models such as ChatGPT or GitHub Copilot to generate code.
- Must be able to explain the code as if you had written it yourself.
- The code must obviously **solve the assignment** and pass a sufficient number of tests on QuickFeed.



## Contributions

[dat520-2025/info/policy.md](#)

- Group members must **contribute equally** to the code
- *Implies that group members should commit a similar amount of code on GitHub.*
- Each group member must **present individually** and explain their work to a TA for approval.
- Group members are expected to be able to explain the code written by other group members.



## Two Working Styles

[dat520-2025/info/policy.md](#)

- Pair programming
  - Alternate between coding vs watching
  - Make sure to switch user that writes and commits code to GitHub
- Code Review
  - Each group member creates a branch
  - Create Pull Request
  - Partner reviews your code before it is merged into the main branch

## When to Commit

[dat520-2025/info/policy.md](#)

- Commits should represent a **logical unit of work**.
  - A bug fix, a new feature, a refactoring, etc.
  - It is up to you to decide what constitutes a unit of work, but it should not be too large.
- *Unacceptable to commit an entire lab assignment in one commit.*

# Commit Messages

[dat520-2025/info/policy.md](#)

- Describe the changes you have made.
- Must follow the Conventional Commits format.

```
93fa221 Hein Meling Mon Jan 8 13:53:59 2024 +0100 fix(lab1): fixed doc format issue in errors task
90509b3 Hein Meling Mon Jan 8 13:51:13 2024 +0100 feat(lab1): implemented fibonacci sequence task
43a353c Hein Meling Mon Jan 8 13:49:57 2024 +0100 feat(lab1): implemented multiwriter task
9d5d0bf Hein Meling Mon Jan 8 13:49:39 2024 +0100 feat(lab1): implemented errors task
2693212 Hein Meling Mon Jan 8 13:44:50 2024 +0100 feat(lab1): implemented the cipher task
f6d8a83 Hein Meling Mon Jan 8 13:43:12 2024 +0100 feat(lab1): implemented stringer task
1f9a64a Hein Meling Mon Jan 8 13:40:55 2024 +0100 mc-go(lab1): answered go questions
88da89b Hein Meling Mon Jan 8 13:38:25 2024 +0100 token(lab1): ran generate_token on go1
23e522c Hein Meling Mon Jan 8 13:20:26 2024 +0100 mc-missing(lab1): answered missing semester questions
24cf9ac Hein Meling Mon Jan 8 13:19:01 2024 +0100 mc-shell(lab1): answered shell questions
31c4fa8 Hein Meling Mon Jan 8 13:17:54 2024 +0100 mc-git(lab1): fixed git answers
ce0027e Hein Meling Mon Jan 8 11:19:53 2024 +0100 mc-git(lab1): answered git questions
8133f91 Hein Meling Mon Jan 8 09:52:23 2024 +0100 lab1: first commit
```

# Group Registration on QuickFeed

- Groups of two (max three)
- If you cannot find a group partner
  - Ask on the #group-maker channel on Discord

## Partner Problems

[dat520-2025/info/policy.md](#)

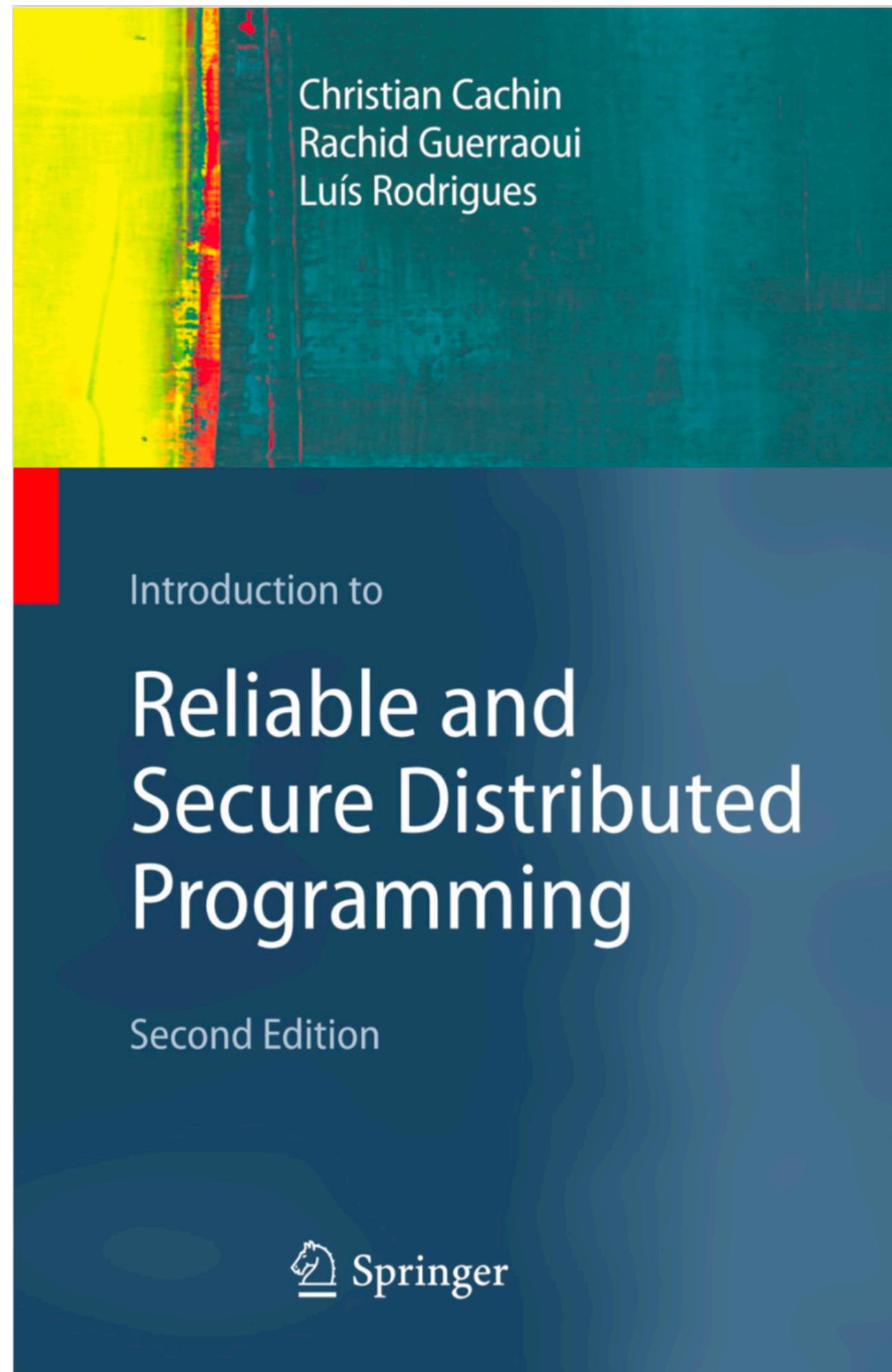
- Please **tell us** and **your partner** early
  - decide to drop out of the course
  - want to dissolve the group and work alone
- To avoid problems, we suggest to organize regular
  - joint work sessions and
  - coordination meetings



# Study Resources

## Official Book

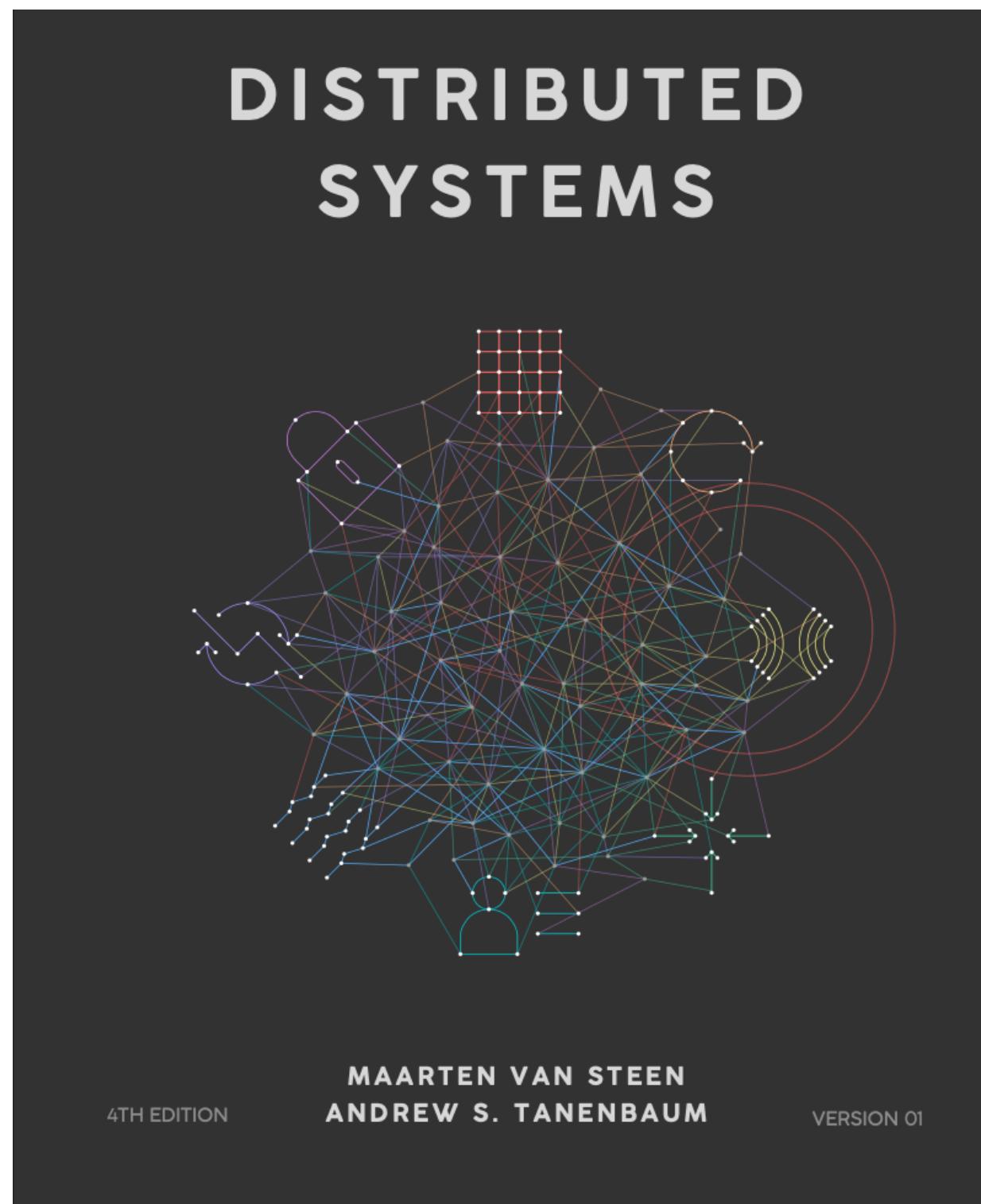
<https://link.springer.com/book/10.1007%2F978-3-642-15260-3>



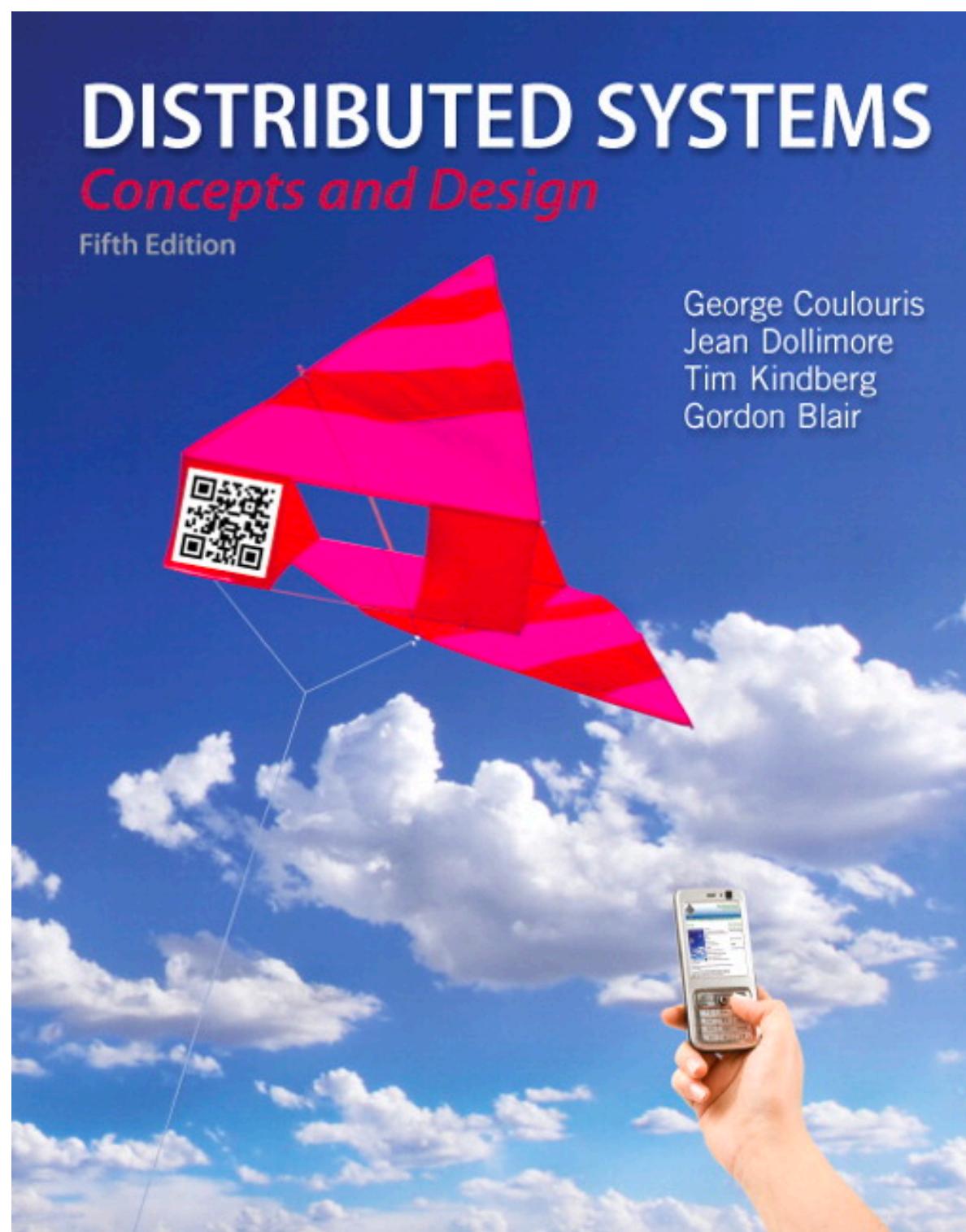
- Available for free download when on UiS campus
- Available in print in the bookstore
- RSDP

- Chapters from RSDP
- Additional papers
- Lab project
- Lecture slides

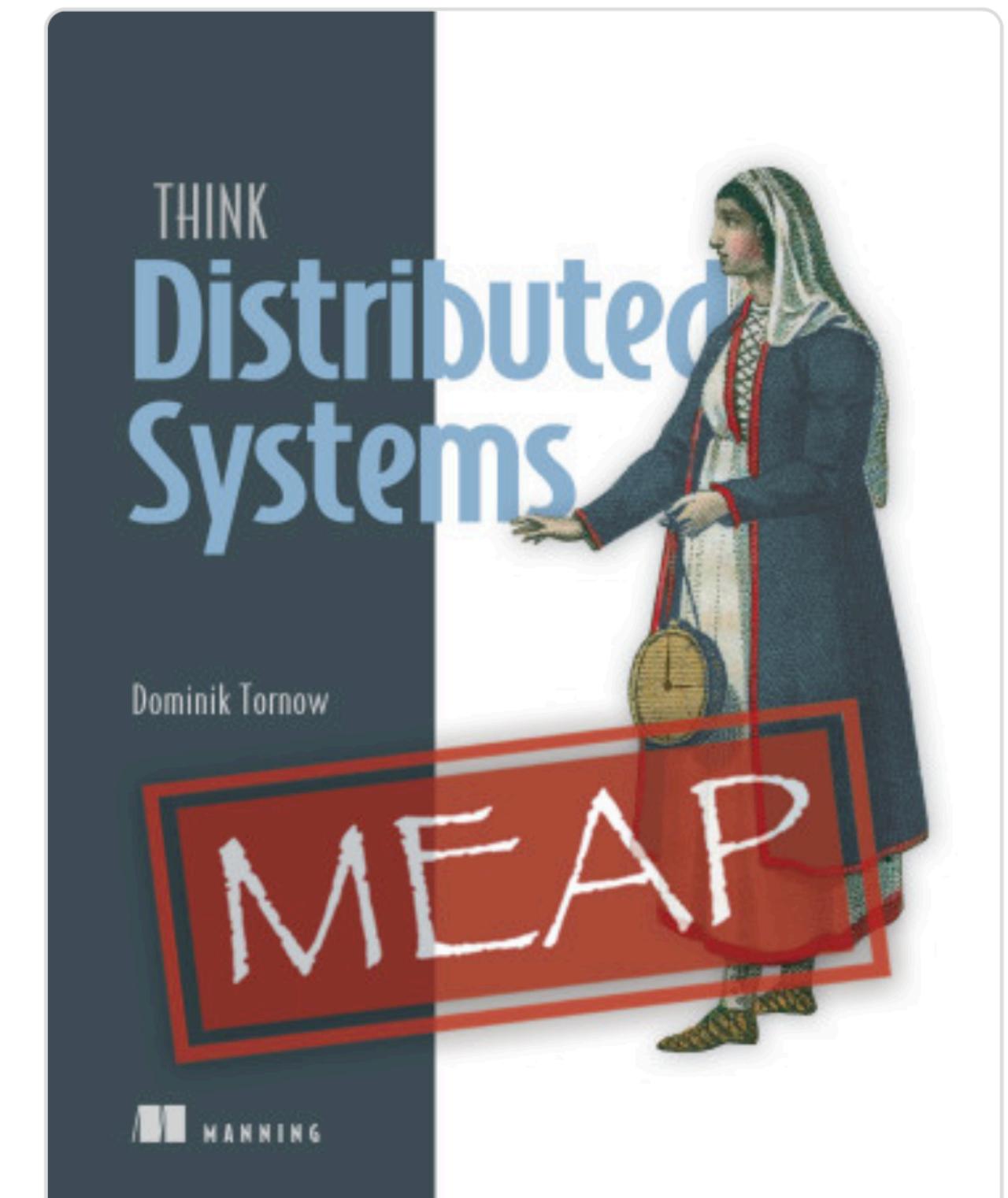
- Lab exercises includes some theory
- Theory exercises and solutions in textbook
  - Recommended practice
  - Exam questions may be picked from textbook exercises
- Previous exams
  - Will post on Canvas



Available free online



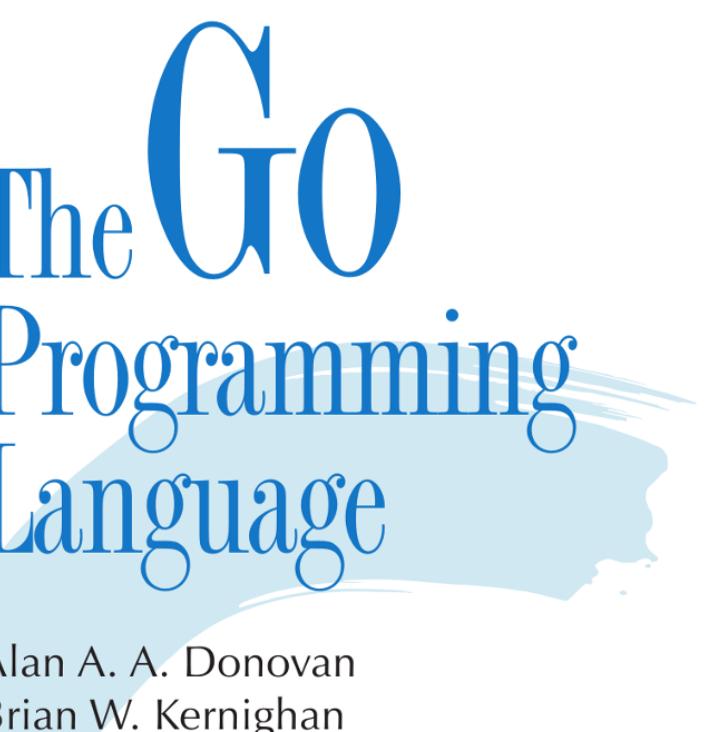
Print copy available  
at UiS library



Partially available online  
(Chapters 1-7 of 14 planned)

## Links

- Official web page: [go.dev](https://go.dev)
- [A Tour of Go](#)
- [Effective Go](#)
- Other [Learning resources](#)
- [Standard library](#) and [Style guide](#)
- Video: [Go Tutorial](#) from freeCodeCamp.org (7 hours)
  - Beware: some information may be outdated!!



# Questions?