

Distributed Systems

Advanced Course

0. Overview



Paris Carbone

PEOPLE

Teachers



Paris
Carbone
(examiner)

- **KTH Assoc.Professor of Data Systems**
- **Director, Data Systems Lab**
- **CoFounder, CEO OrbDB Labs**

- **17yr Experience in Distr Systems R&D**
- **KTH Best Teacher Award Recipient**
- **Co-creator of Apache Flink**
- **(ACM SIGMOD Systems Award 2023)**



PEOPLE

Teachers



Paris
Carbone
(examiner)



TAs



Harald Ng

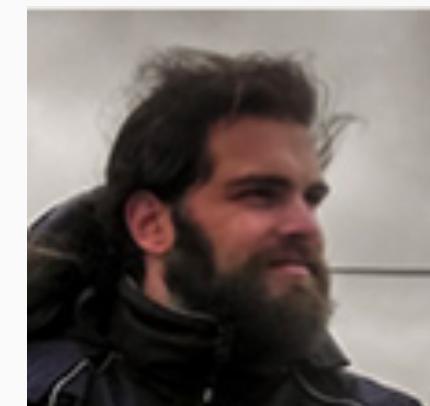


Sonia
Horchidan

Junior TAs

- Dimitrios Bakalis

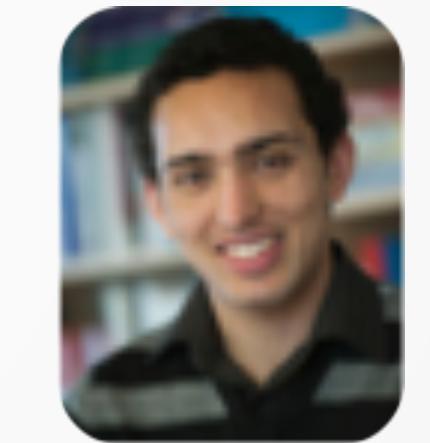
Alumni - Hall of Fame



Lars
Kroll



Cosmin
Arad



Tallat
Shafaat



Niklas
Ekström



Seif
Haridi



Ali
Ghodsi



KTH-2026

(previous teacher)

Systems @ Data Systems Lab

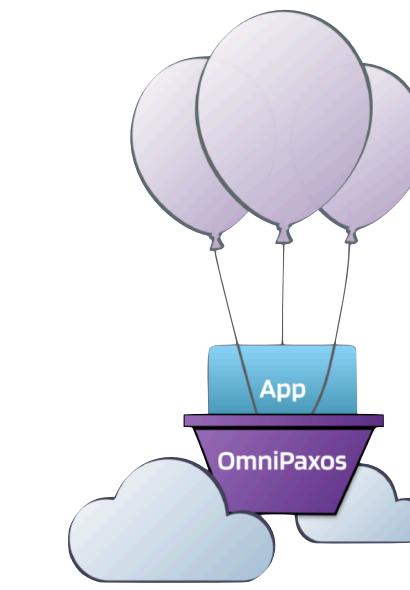
- ▶ PhDs/Research Engineers at KTH+RISE in distributed systems
- ▶ Several interns a year (depending on funding) from ID2203
- ▶ **Production-Grade Open-Source Systems**



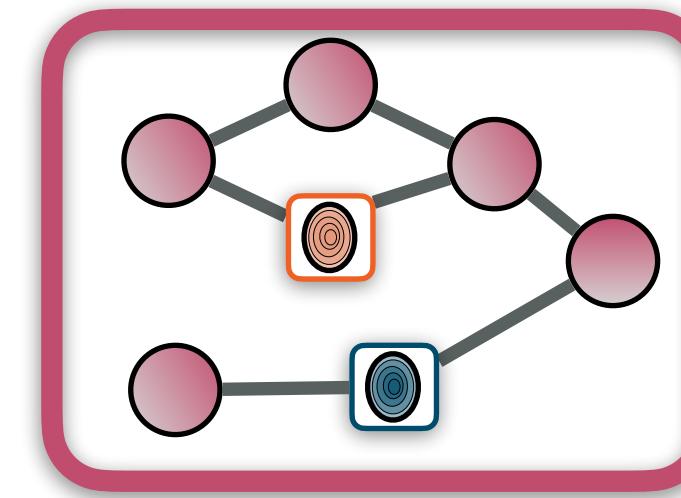
[2013-]
Flink



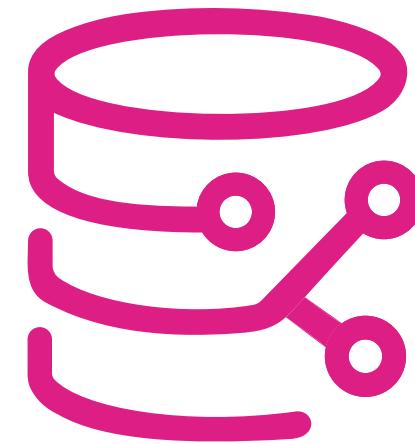
[2018-]
Aqua



[2021-]
OmniPaxos



[2022-]
Portals

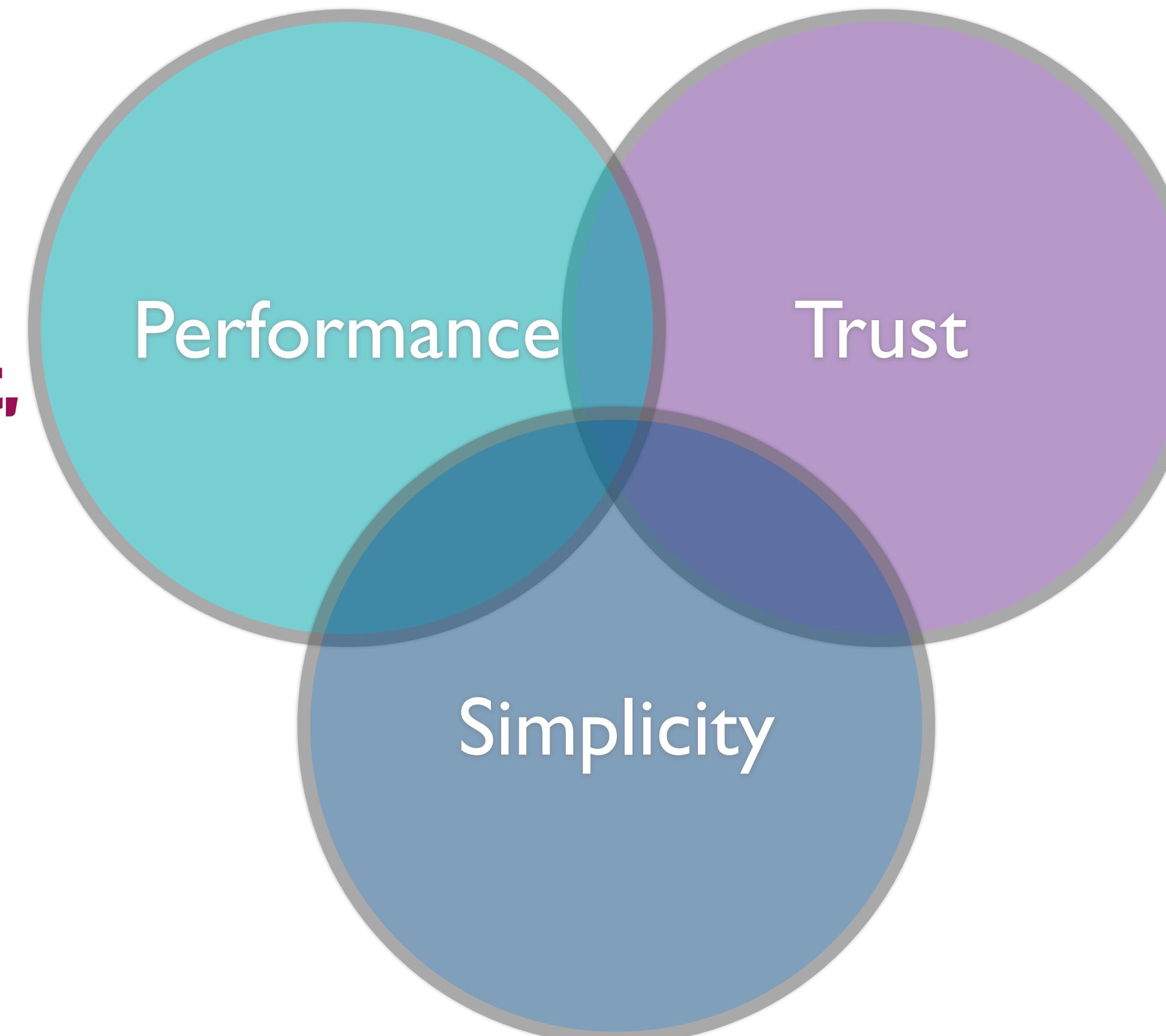


[2023-]
Orb DB

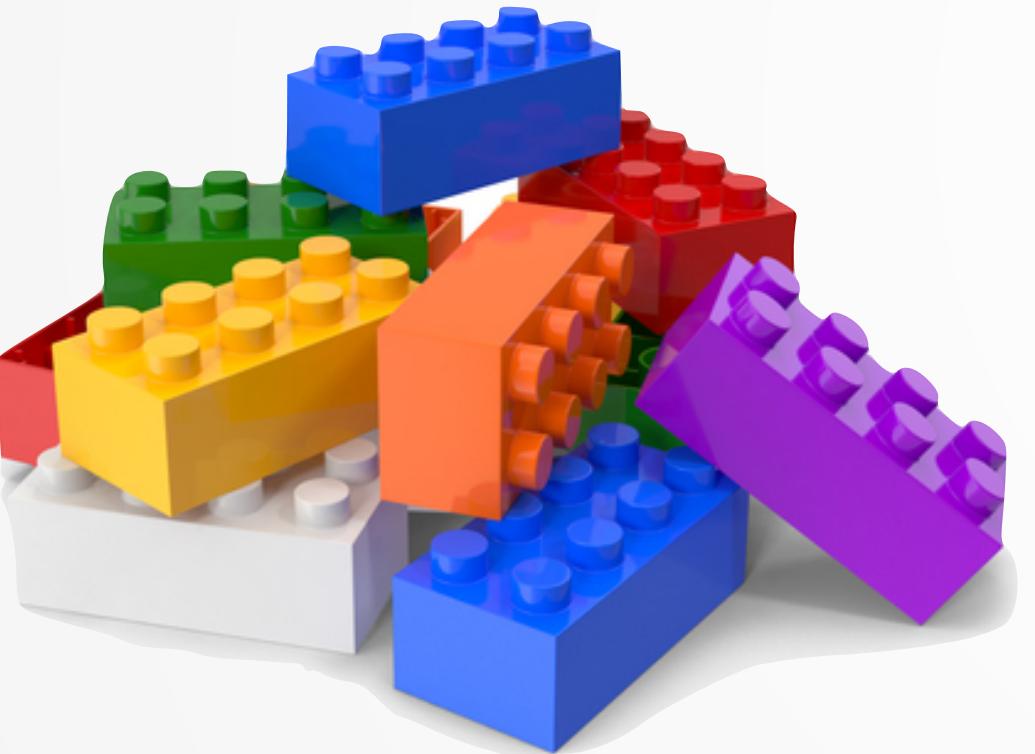
▶ **Your course project can result into a system contribution**

Our 3 Systems Principles

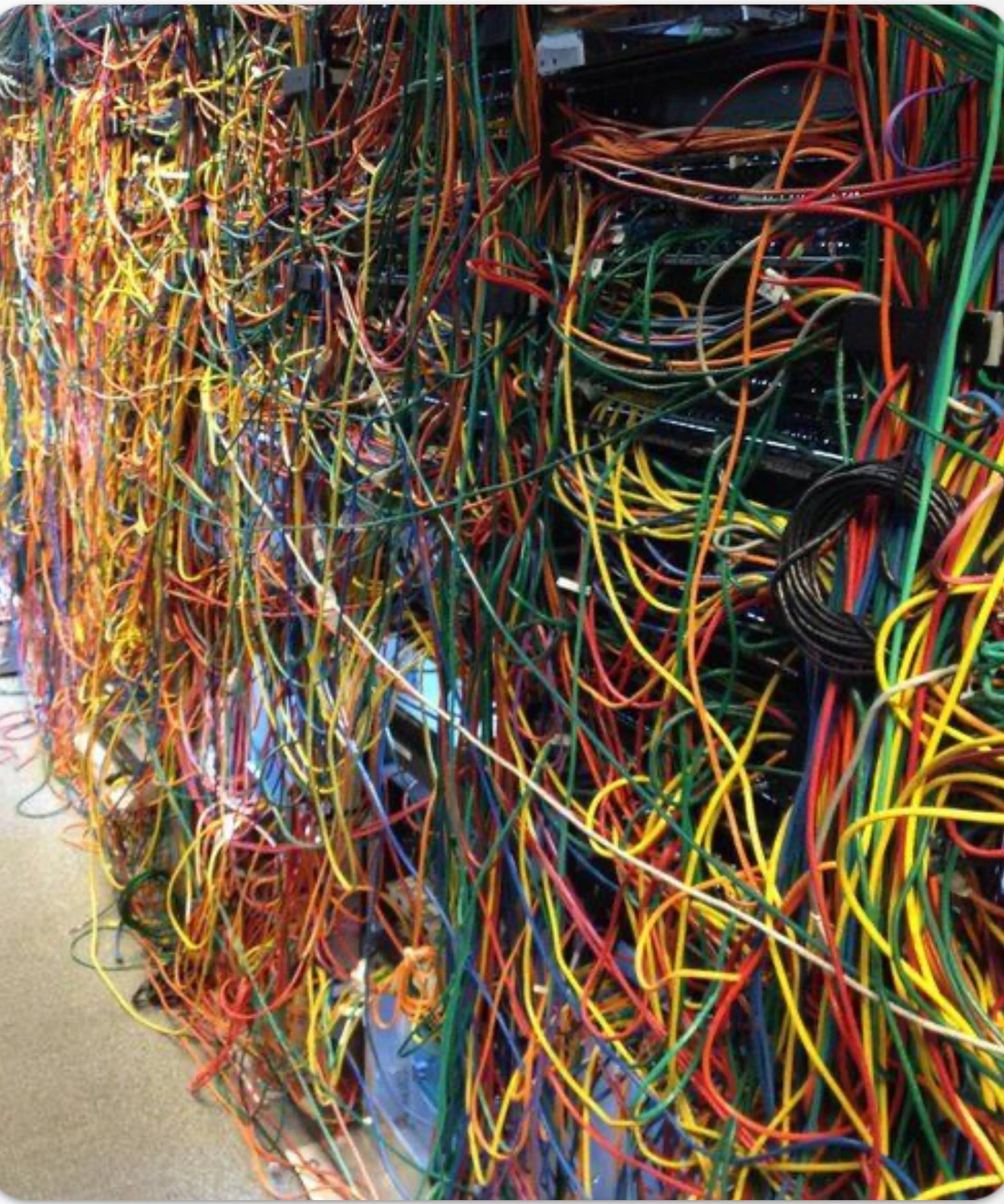
Compromise none

- 
- ▶ **Low Latency**
 - ▶ **Energy Efficiency**
 - ▶ **Data Management, HPC, Hardware acceleration**
 - ▶ **Fault Tolerance**
 - ▶ **Consistency Guarantees**
 - ▶ **Distributed Systems Theory**
 - ▶ **Intuitive User Interface/Ease-of-Use**
 - ▶ **Generative AI, LLMs in action**
 - ▶ **Engineering, Programming Languages**

WAY OF THINKING



VS

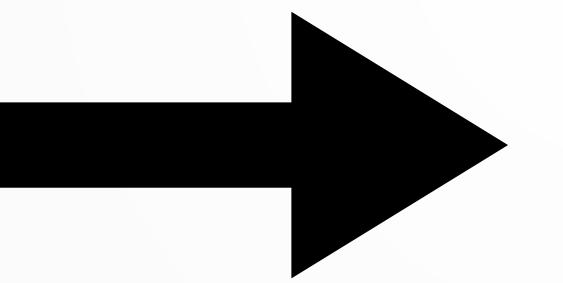
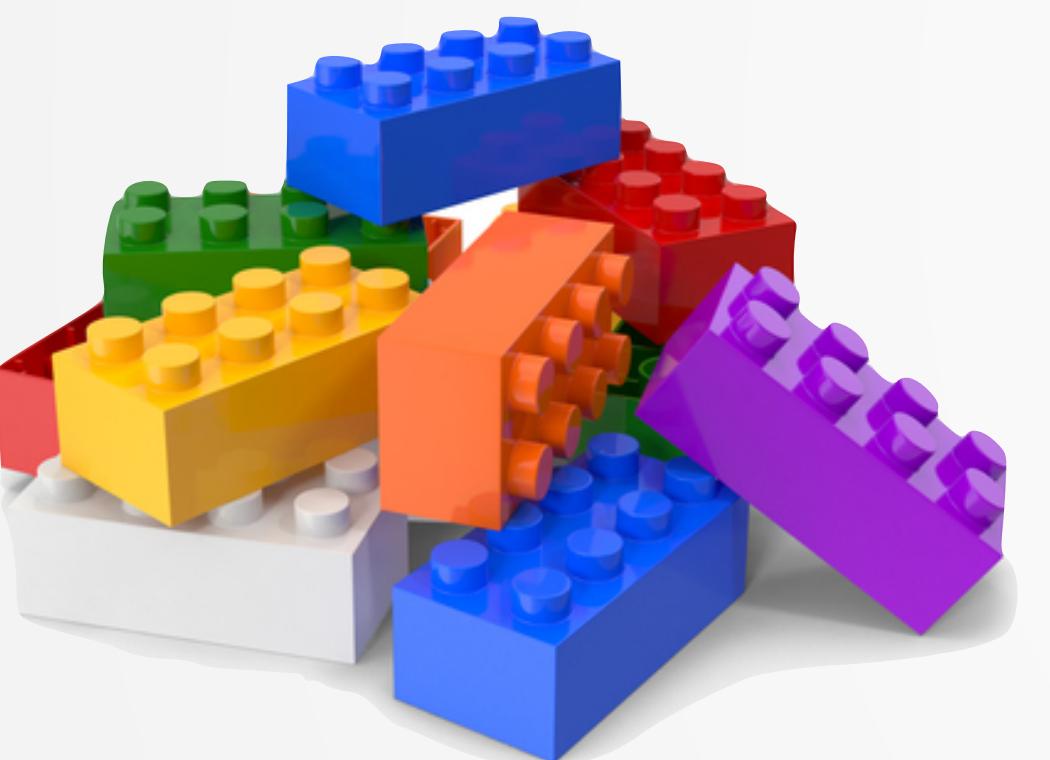


ID2203



KTH-2026

WAY OF THINKING



ID2203



KTH-2026

COURSE TOPICS



- ▶ Intro to Distributed Systems
 - ▶ Fundamental Abstractions and Failure Detectors
 - ▶ Reliable and Causal Order Broadcast
 - ▶ Consensus (Paxos)
 - ▶ Replicated State Machines (OmniPaxos, Raft, Zab etc.)
 - ▶ Distributed Shared Memory & CRDTs
 - ▶ Real-Time Abstractions (Spanner, Atomic, Quantum clocks)
 - ▶ Consistent Snapshotting (Data Management)
 - ▶ Distributed ACID Transactions (Cloud DBs)
- ▶ Basic Components
- ▶ Advanced Systems

EXCLUSIVE NEW CONTENT IN 2026

- ▶ A new model of mixed consistency (CIDR 2026)
- ▶ Quantum-entangled clocks @ KTH



ID2203



KTH-2026

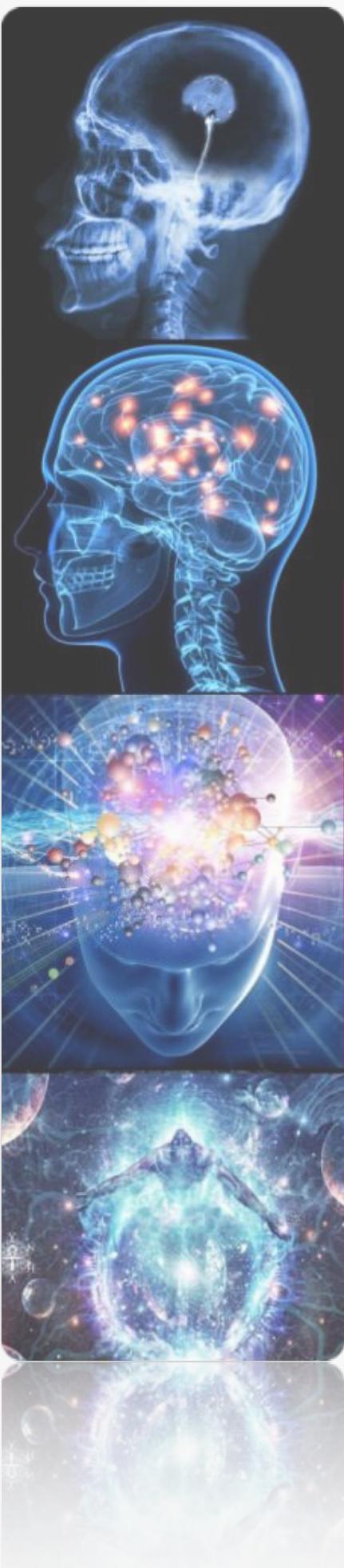
KTH PATH OF EXCELLENCE IN DATA SYSTEMS



- Data Modelling
 - Declarative Programming
 - Hardware Management
 - Transaction Management
 - Database Query Optimisation
-
- Replicated State Machines
 - Eventual Consistency Models
 - Atomic Commitment
 - Optimistic Concurrency Ctrl
 - Strict Serializability
 - Distributed Snapshotting
-
- Build your own Data System

IV1351 Data Storage Paradigms

KTH PATH OF EXCELLENCE IN DATA SYSTEMS



- Data Modelling
 - Declarative Programming
 - Hardware Management
 - Transaction Management
 - Database Query Optimisation
-
- Replicated State Machines
 - Eventual Consistency Models
 - Atomic Commitment
 - Optimistic Concurrency Ctrl
 - Strict Serializability
 - Distributed Snapshotting
-
- Build your own Data System

ID2203 Distributed Systems Advanced Course

KTH PATH OF EXCELLENCE IN DATA SYSTEMS



- Data Modelling
- Declarative Programming
- Hardware Management
- Transaction Management
- Database Query Optimisation

- Replicated State Machines
- Eventual Consistency Models
- Atomic Commitment
- Optimistic Concurrency Ctrl
- Strict Serializability
- Distributed Snapshotting

- Build your own Data System

PhD at Data Systems Lab

COURSE CONTENT

Canvas - Recorded Lectures - <https://canvas.kth.se/courses/59258>

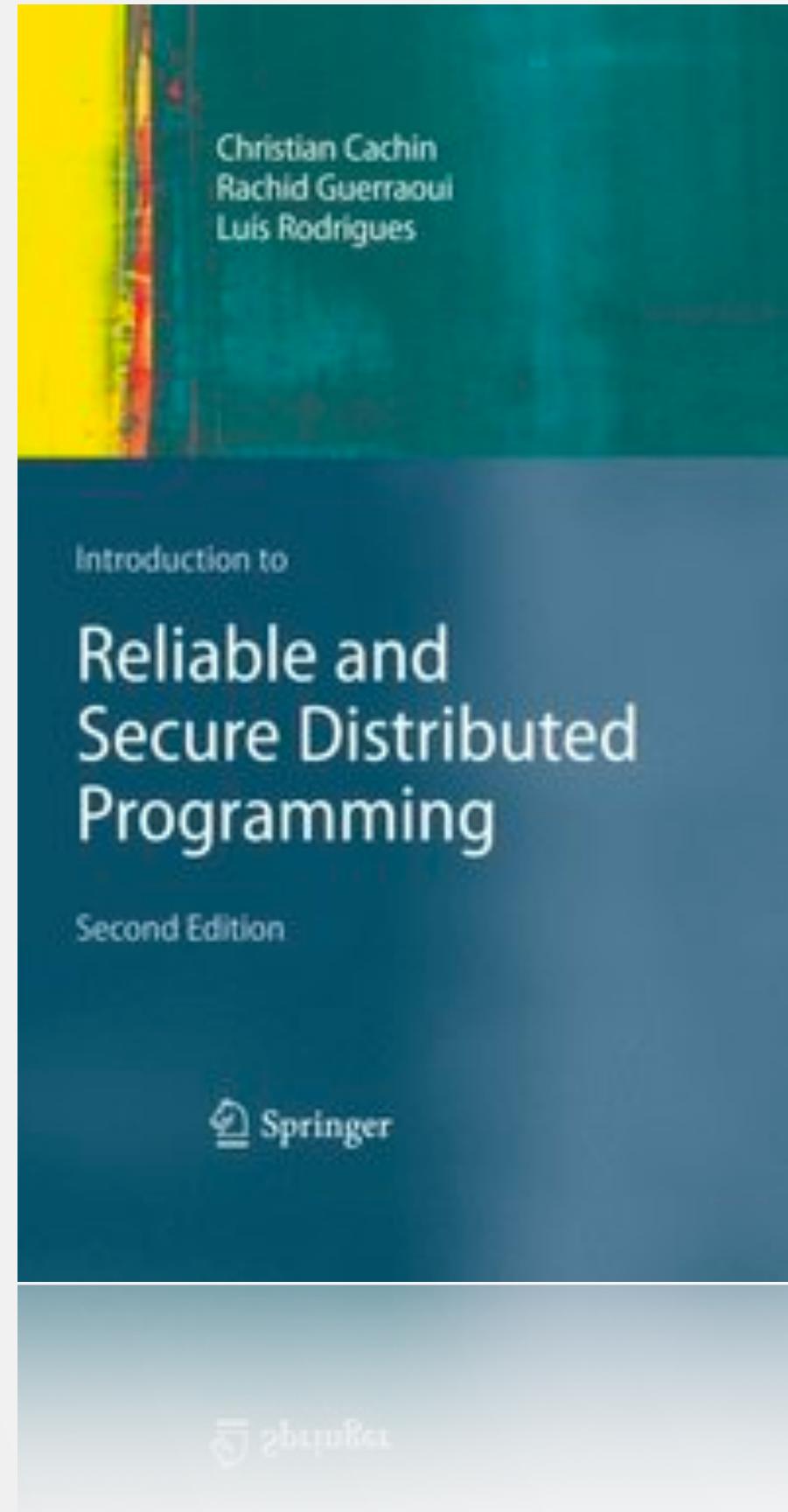
- ▶ Textbook & Algorithms
- ▶ Quizzes
- ▶ Labs & Tutorials
- ▶ Course Forum (Piazza)
- ▶ Assignments & Project
- ▶ Final Exam

LECTURES

- ▶ **Rooms** : U21, U31, U41, U51
- ▶ **Zoom** available for those who cannot make it.
- ▶ **Physical participation** : optional but recommended*
- ▶ **Recorded Videos** will be also available in canvas.

*Physical participation enhances understanding and leads to top grades

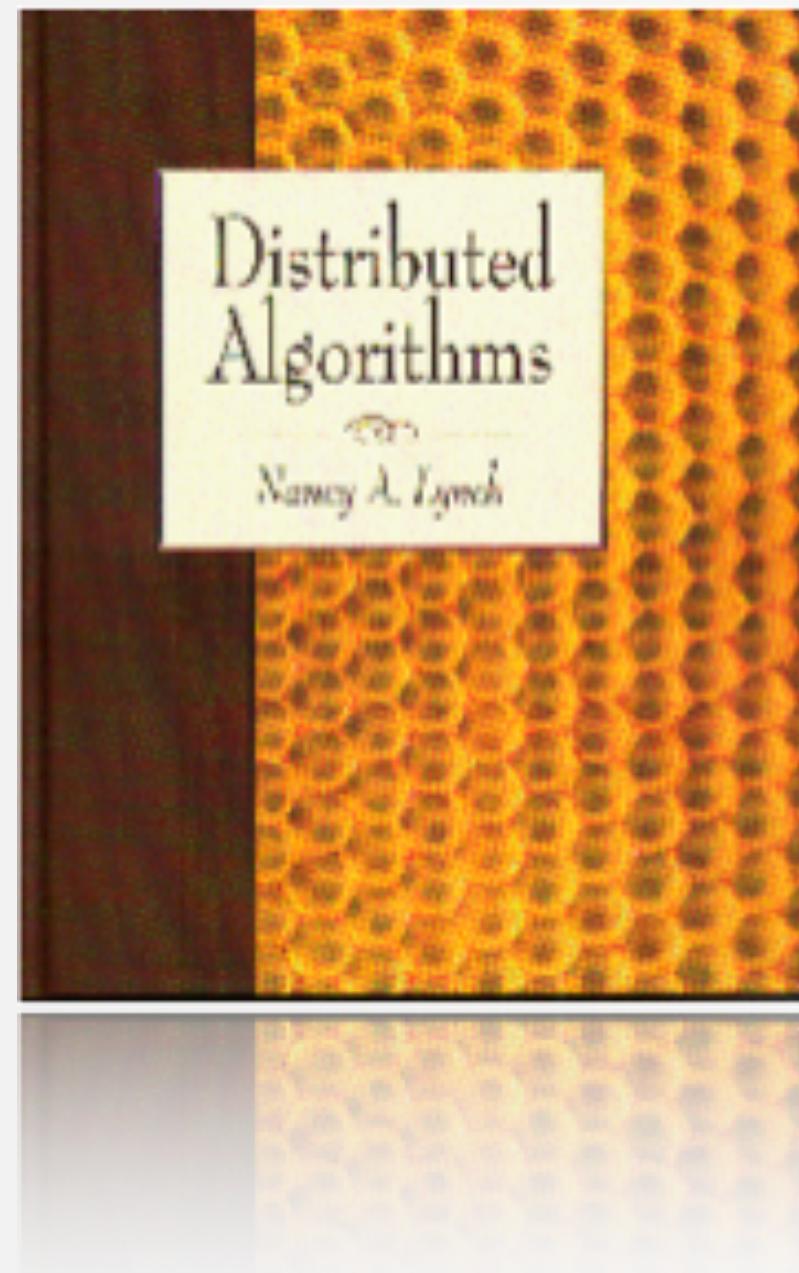
TEXTBOOK



“Reliable and Secure Distributed Programming” Cachin, Guerraoui, Rodrigues

- Main textbook of the course
- Covers most of the content presented
- Complements lectures but doesn't replace them
- E-book available at KTH Library & Canvas

TEXTBOOK



“Distributed Algorithms” by Nancy Lynch

- Another Recommended Reading
- Covers Input-Output Automata

QUIZZES

1. Non-Graded

- Complement each lecture
- Crucial for assessing understanding

2. Graded

- Graded after each module
- **13P** of the final grade

PIAZZA FORUM

- <https://piazza.com/kth.se/spring2026/id22032026/home>
- Questions & Discussions
- Anonymous posting
- All registered students will be automatically added
- Notify us if you cannot access it
- **Check your KTH inbox for the Piazza invitation**

LABS & TUTORIALS

- Live Zoom Sessions with TAs / Guests
- Recorded Uploads in Canvas
- Potential Topics (Based on Demand/Trends/TA Availability)
 - Distributed Programming Frameworks (e.g., Kompics)
 - Model Checkers (TLA+, TLC)
 - Refreshers for Math/Proof Systems
 - Exercise & Project Q&A
 - Guest lectures on specific system areas

ASSIGNMENTS & PROJECT

1. Programming Exercises - **7P (Bonus)**

- Algorithm Implementations
- Kompics (Scala) environment
- They help you get a better grade but are not required to pass course.

2. Project - **30P + 10P (Bonus)**

- Several topics and prog languages to choose (mostly Rust).
- Intermediate reports might be peer reviewed.
- Group projects are also allowed.
- **15P requirement to pass**

FINAL EXAM

- Up to **50P**
- Physical Exam as of 2023
- ~Advanced version of graded quizzes
- Tests knowledge of course topics with emphasis in reasoning
- Multiple Choice & Explanation/Proof Questions
- Pass : **25/50P** for A-F course graded part

GRADING SCHEME

For the 4.5 credit A-F graded part the grade is calculated as follows:

Graded Quizzes (max 13P)

+ **Programming Exercises (max 7P)**

+ **Project (max 40P)**

+ **Exam (max 50P)**

= **max 110P**

where **90+ A, 80+ B, 70+ C, 60+ D, 50+ E, <50 F**

PRO-TIPS

- Attend actively and question everything!
- Complexity builds up. Do not miss a step/lecture.
- Think guarantees first, performance optimisations later.
- Start learning Rust early if you plan to take a Rust project.
- Use the programming assignments simulator to put your understanding of the different properties to a test.