

Decentralized Systems Engineering

CS-438 – Fall 2025

DEDIS

Bryan Ford, Pierluca Borsò-Tan

EPFL

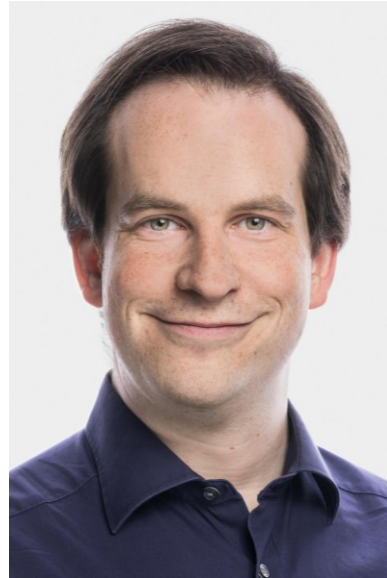
Credits: P. Tennage, C. Basescu, et al.

Your Lecturers

Prof. Bryan Ford



Pierluca Borsò-Tan



What are Distributed and Decentralized Systems?

- *Distributed system:*

a system of multiple computers (nodes) communicating over a network to achieve a common goal.

- *Decentralized system:*

a distributed system without a single point of control or authority.

different nodes or subsets of the network may be owned or controlled by different people, organizations or interests.

Some examples

- Centralized distributed systems
 - Google
 - Netflix
 - Facebook
 - WeChat
- Decentralized distributed systems
 - “The Internet”
 - E-mail
 - Usenet
 - BitTorrent, IPFS
 - Bitcoin, Ethereum
 - Avionics, control systems
 - E-Voting
 - Tor, I2P

Course Goals

- Understand the fundamental and practical challenges inherent in designing and building decentralized systems.
- Get a feel for the (limited) body of techniques and solutions
- Examine a number of real systems, past and present: how they work, and why they succeeded ...or failed
- Become better engineers: solidify this knowledge by applying it !
 - Build a small, but working, usable and robust, decentralized system

Why Study Decentralized Systems?

- Devise solutions when there is no common authority everyone trusts
- Add new tools to your engineering toolbox
- Learn to think and question your assumptions
- The world is full of decentralized systems

Course Organization

Course Organization

Lecturers: Prof. Bryan Ford, Pierluca Borsò-Tan
TAs: Shailesh Mishra, Georgiy Lebedev
AEs: Roxanne Chevalley, Konstantinos Chasialis, Mahdi Atallah
E-mail: cs438@groupes.epfl.ch

Material: Moodle (lectures, readings): <https://go.epfl.ch/CS-438>
Ed (forums, announcements)

Assessment: 40% homework 30% project 30% written exam

Class participation is not graded, but fundamental to success !

Weekly workload

- Lectures: Monday 10:15-12:00 (INF 1) **+ this afternoon !**
13:15-15:00 (MEB 331)
- Q&A w/ TAs: Friday 15:15-17:00 (SG 0211)
- Help understanding concepts, homework requirements & architecting
- *The TAs will not debug your code for you !*
- Self-guided: Monday 13:15-15:00 (MEB 331)
- Room open to hack, discuss design/problems, and test with classmates !
- Homework: 8-12h depending on your understanding & development skills
No sharing or copying code; everyone implements their own Peerster!

Course Syllabus

Week	Content	
1	Course intro UseNet and gossip	Foundations for Decentralization
2	Flooding search and routing	
3	Jeune fédéral exercise session only	
4	Structured search and compact routing	
5	Distributed storage	
6	Replication and consensus	
7	Threat modeling and threshold crypto	Attacks & Defenses
8	Anonymous communication	
9	Sybil attacks and defenses	
10	Blockchains and cryptocurrencies	Blockchains
11	Smart contracts	
12	Advanced blockchain architectures	
13	Testing & chaos engineering	Engineering
14	Decentralized democracy	E-Voting

Homework & Project

Build a decentralized system

Individually:

- Network layer (introduction to Go)
- Gossiping
- File sharing
- Consensus & blockchain

In groups of 3~4 students:

- One of many predefined projects,
- ... or your own !



Course Syllabus

Homework & Project Schedule

Week	Content	HW0	HW1	HW2	HW3	Project
1	Course intro UseNet and gossip	Sep 8>				
2	Flooding search and routing	<i>Network</i>				
3	Jeûne fédéral exercise session only	>Sep 23	Sep 23 >			
4	Structured search and compact routing		<i>Gossiping</i>			
5	Distributed storage		> Oct 7	Oct 7 >		
6	Replication and consensus			<i>FileSharing</i>		
7	Threat modeling and threshold crypto			> Oct 28	Oct 28>	
8	Anonymous communication				<i>Consensus & Blockchain</i>	
9	Sybil attacks and defenses					
10	Blockchains and cryptocurrencies				> Nov 18	Nov 18 >
11	Smart contracts					
12	Advanced blockchain architectures					<i>Your Project</i>
13	Testing & chaos engineering					
14	Decentralized democracy					> Dec 20

Homework : Your Path to Success !

- They build on top of each other
 - Don't skip !
 - Improving previous HW is rewarded
- Defensive programming
- Manage your time
 - Don't wait for the last minute – you will fail !
 - How much is that perfect score worth to you ?

Homework Assessment (40%)

	Code Quality	HW0	HW1	HW2	HW3	Total grade
HW0 due	1%	3%				4%
HW1 due	1%	1%	9%			11%
HW2 due	1%	1%	9%	14%		25%
HW3 due	4%	1%	11%	15%	29%	60%
Total	7%	6%	29%	29%	29%	100%

- You can talk with each other and exchange ideas, **not** share code
Plagiarism will be checked for and dealt with severely

Homework Assessment (40%)

Public tests (90%):

- Unit tests 55%
- Integration tests 30%
- Performance (benchmarks) 5%

Hidden tests (10%):

- Verify robustness / defensive programming

Project: Organization

- Group Project (3~4 members)
 - Build on top of (some part of) Peerster
- Guided topic selection, with clear goals
... or you can be creative !
- Example project topics
 - Secure routing
 - Anonymous reputation
 - E-Voting
 - CRDTs



Project: Organization (cont'd)

- Regular meetings with TAs (week 10+)
- Final project report (December 19)

(More detailed information will follow later in the semester)

Project Assessment (30%)

- Group grade (50%)
 - Report
 - Project quality
 - Goals achieved
- Individual grade (50%)
 - Contribution to the project
 - (1 page) individual report
- **No winners in a losing team !**

Final Exam (30%)

- Format: short & long questions, multiple-choice questions
- Assessment of:
 - your understanding of fundamental concepts
 - your ability to reason about decentralized systems
 - your ability to engineer such systems
(design, evaluate, test, make trade-offs)
- Non-goal:
 - You learning the slides by heart – it won't help.

Support structure

Questions ?

Lecturer: Bryan Ford, Pierluca Borsò-Tan
TAs: Shailesh Mishra, Georgiy Lebedev
AEs: Roxanne Chevalley, Konstantinos Chasialis, Mahdi Atallah
E-mail: cs438@groupes.epfl.ch

Questions: Exercise sessions (Fridays)
Ed Discussion & FAQs

Engineering Recording: Go thinking & best practices
Lectures: Week 13: testing & chaos engineering

Decentralized Systems Engineering

CS-438 – Fall 2025

DEDIS

Bryan Ford, Pierluca Borsò-Tan

EPFL

Credits: P. Tennage, C. Basescu, et al.

Why Build a Decentralized System?

- Sometimes a basic requirement
- Availability, reliability and safety
- Lower resource usage (in specific scenarios)
- Enabler for resilient ecosystems
- Nature has shown they can work incredibly well !

Major Topics & Applications

- Communication: messaging, chat, voice/video
- Data: storing, sharing, searching and mining
- Collaboration mechanisms
- Social networking
- Deliberation, e-voting, reputation
- Blockchains, cryptocurrencies, and smart contract systems

Related, but (mostly) out of scope in this class:

- Decentralized control systems, industrial automation
- Intelligent agents, self-organizing robotics
- Military & civilian ad-hoc networks (MANET, VANET, FANET, etc.)

Recurrent Issues and Themes

- Identity (real, sybil) versus location
- Information integrity and privacy
- Behavior accountability
- Denial-of-service
- Protocol efficiency, in the normal case and under load or attack

Why might we prefer centralized systems?

Let's think !

~ 2 min, write down your ideas

~ 1 min, discuss with your neighbor

Then share with the class !

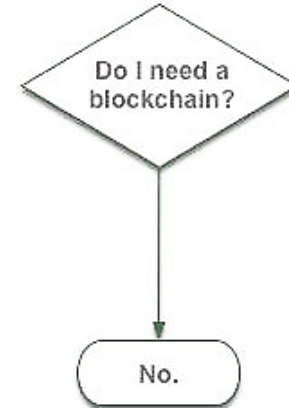
Why might we prefer centralized systems?

Simplicity:

- Engineering
- Management
- Security model
- Version management

Thereof:

- Performance / Efficiency
- Cost



Decentralized Communication

An introduction

Communicating with a (known) peer

- Same machine
a file in a shared directory
- Local networking
shared drive, file transfer
- Global networking, centralized trust
a file on a shared server (FTP? Dropbox?)
- Decentralized
e-mail (signed, encrypted)

or write command on
UNIX/Linux

Communicating with (many, unknown) peers

- Same machine
a file in a shared directory, or Linux `wall` command
- Local networking
shared drive, intranet website
- Global networking, centralized trust
mailing lists, forums, Reddit, ...
- Decentralized
??? → this afternoon's lecture

Communicating with (many, unknown) peers

- Decentralized
 ??? → this afternoon's lecture

Many open questions:

- How do we reach unknown peers ?
- How do they find out about us / our node ?
- How do we eventually reach *every* peer ?
- How can we communicate reliably ?
 - Are they online ?
 - Is the network « stable » ?