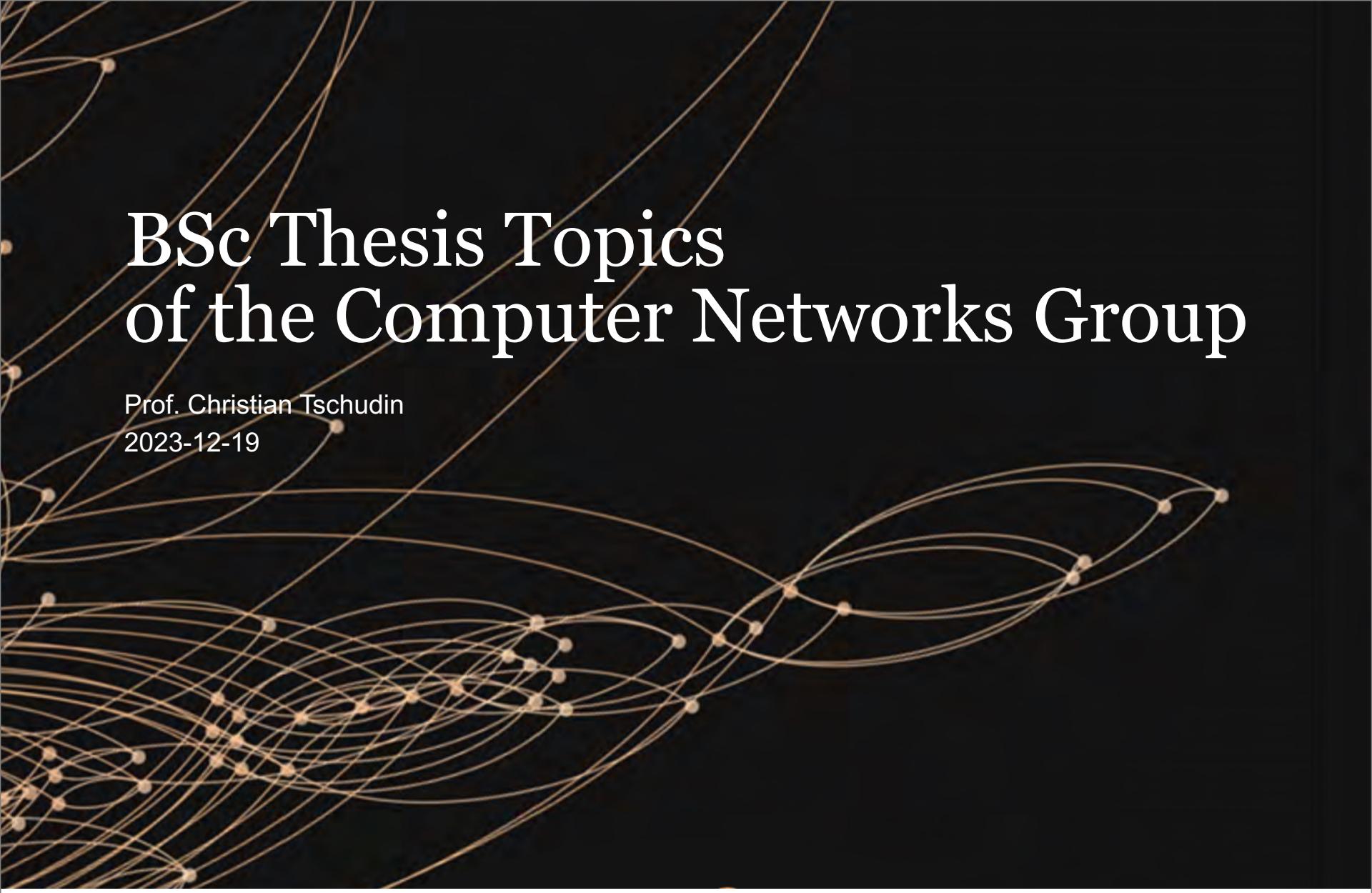


# BSc Thesis Topics of the Computer Networks Group

Prof. Christian Tschudin

2023-12-19

University of Basel

# Members of the Computer Networks Group

Christian Tschudin
*computer networks*

Erick Lavoie
*peer-to-peer*

Osman Biçer
*cryptography*

Ali Ajorian
*compilers*

# Teaching

*"modulo current sabbatical"*

**Fall Semester**

- Computer Architecture (formerly CATC) / HS23 by Prof Wagner
- (MSc) Foundations of Distributed Systems / HS23 by Dr Lavoie

**Spring Semester**

- Distributed Applications and Internet Architecture (formerly IaS)
- (MSc) Computer Networks (MSc)

**Other involments and topics of past seminars**

- Scientific Writing (MSc), Erick Lavoie
- Seminars:
  . Conflict-free Replicated Data Types (CRDT)
  . Programming with Monads
  . Programming with LISP
  . "101 things I learned in Computer Science"

# General Areas for BSc Projects

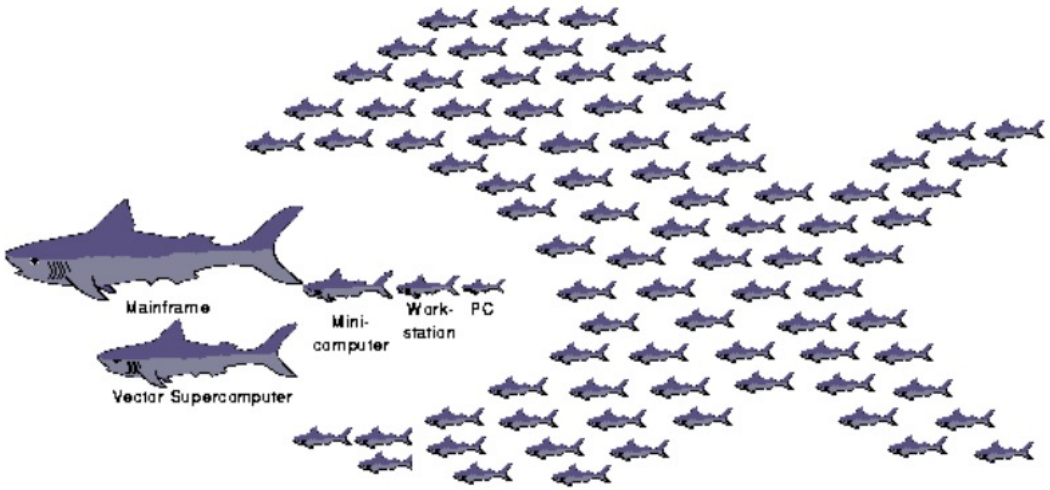
A. Distributed Applications / Peer-to-Peer

B. Hostile Environments (like the Internet or your CPU)

C. BYO

# A) Distributed Applications

Aristotle: *«The whole is more than the sum of its parts»*



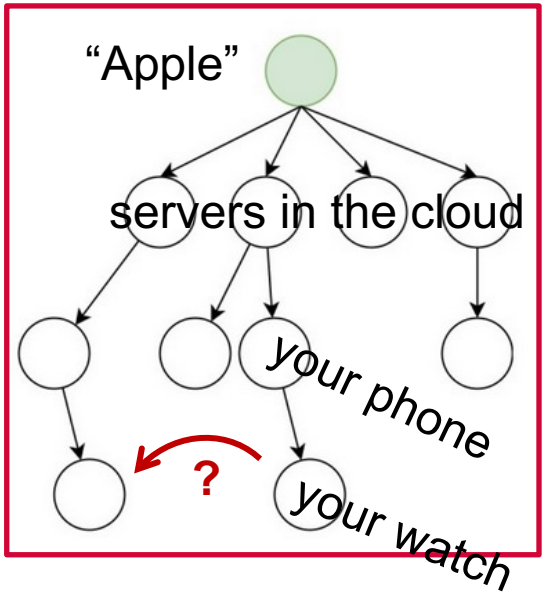

"scale out" instead of "scale up"

Despite the cloud: statement is not obvious in Computer Science, as server-based solutions dominate

Science question: What «DNA» for successful peer-to-peer applications?

## A) Distributed Applications:

### a decentralized scenario




**Today's distribution economics:**
- buy a smart watch
- buy a smartphone
  (to connect your smart watch)
- buy a mobile plan
  (to connect your smartphone to the cloud)
- buy a cloud subscription
  (to access Apple's services)

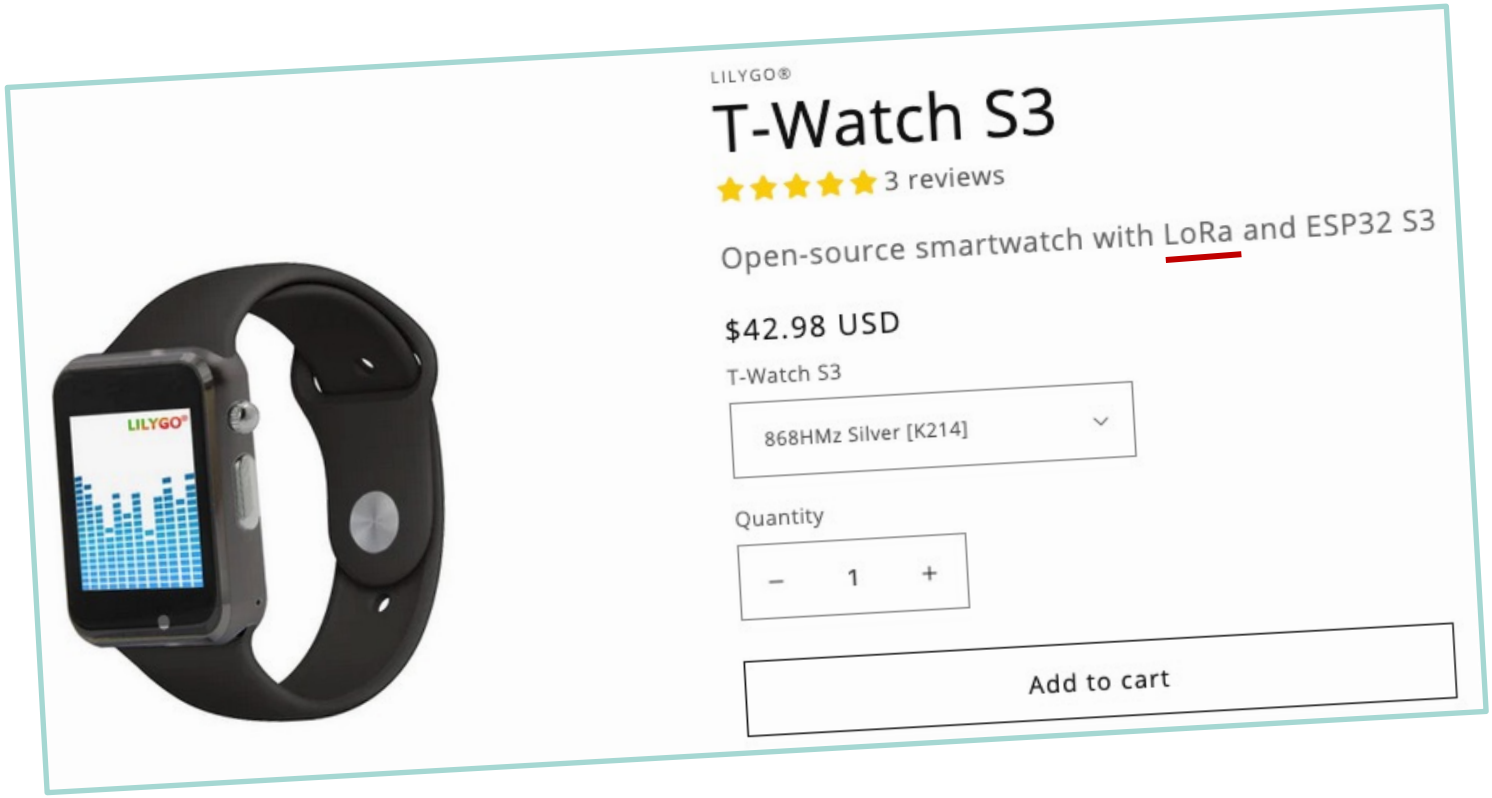
**An alternate economic model:**
- buy some device
- let it talk to its peers

There is a market for P2P knowhow, startups

## A) Distributed Applications:

### abundance of devices and connectivity

Long-range connectivity is available today




LoRa (Long-Range radio): 100m to multiple kilometers

## A) Distributed Applications (contd):

### re-structuring «the stack»

A post-Internet architecture for distributed applications:

distributed applications based on CRDTs *)

data replication via trustable append-only logs

peer-to-peer connectivity

CRDT=«Conflict-free Replicated Data Types», discovered 2011

## B) Hostile Computing Environments

Most cryptography relies on you being able to trusting your computing device e.g., when en- and decrypting

Unfortunately, this assumption is more and more voided:

- not «your» smartphone, or laptop
- forced updates of OS, apps
  (Google playstore: re-updates once a week)
- not blockable scanning of your content
- a big science problem:
  no reproduciable experiments anymore

*«Den Teppich unter den Füssen wegziehen»*

## B) Hostile Computing Environments
### science question: how to safely use a computer post-compromise?

Cryptographic solutions exist in the client/server model.

What about peer-to-peer?

First theory result in our group, «oblivious homomorphic encryption»

awaits exploration with implementations

## BYO (bring your own)

Many ways «to do distribution»

If you have an idea or use case: come and talk to us!

## Some Titles of Past/Ongoing/Scheduled BSc Theses

- *Security Bubbles for Trust Scoping*
- *Decentralized Kanban Board using Secure Scuttlebutt and CRDTs*
- *Managing and Distributing Software Updates Using Append-Only Logs*
- *Managing Resources of Network Nodes Using Append-Only-Logs*
- *ED25519 for Micropython on the ESP32*
- *Implementing the Double Ratchet algorithm in Tremola, a Scuttlebutt based messaging app for Android*
- *NetShell Network: An identity-centric store-and-forward network*
- *A Discovery Protocol for Secure Scuttlebutt based on chat app Tremola*

## Some Potential Titles for BSc theses

- A decentralized **app store**
- Porting Lokens (a decentralized **crypto currency** without mining) to embedded processors
- **Peer-to-peer replication over Git** (!= Github)
- A code **obfuscation engine** in C++
- LoRa **packet scheduling**
- Audio waveforms for **packetized shortwave transmission**
- Using **Fully Homomorphic Encryption** for PIR (Priv. Inform. Retrieval)

**Programming languages used:**

- JavaScript, Kotlin, C, C++, asm, Python, and why not LISP?

# Some References

**P2P Economies:**

«Designing P2P Systems as Closed Knowledge Commons», Dec 2023

https://openreview.net/pdf?id=w4ZrjzLj1f

**Lokens:**

- https://arxiv.org/abs/2305.16976, May 2023
- https://lokens.net/ (Swiss non-profit association)

**Oblivious Homomorphic Encryption:**

https://eprint.iacr.org/2023/1699, Nov 2023

**Thank you** for your attention. **Questions?**