

# TDDD25: Distributed Systems Programming Project

Petru Eles and Adrian Horga

Computer and Information Science  
Linköping University

January 2019

# Contact

**Adrian Horga**

adrian.horga@liu.se

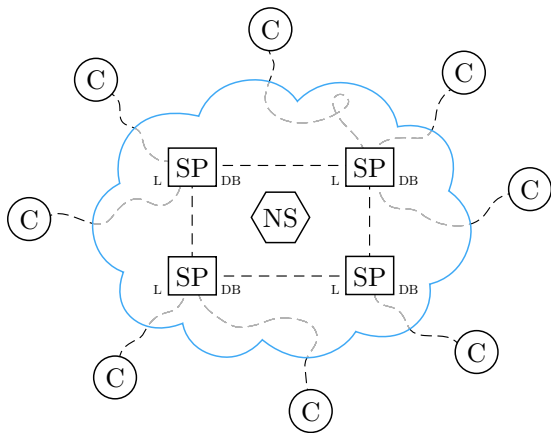
Office 329:198, Building B

# Organization

<https://www.ida.liu.se/~TDDD25/>

- 1 teaching session
- 7 lab sessions
- 1 + 5 labs
- 2 groups : Group *A* and Group *B*
- Registration deadline: February 17
- Completion deadline: March 10

# Distributed Database



**C** — client  
**SP** — server/peer  
**NS** — name service  
**DB** — database  
**L** — lock

# Source Code

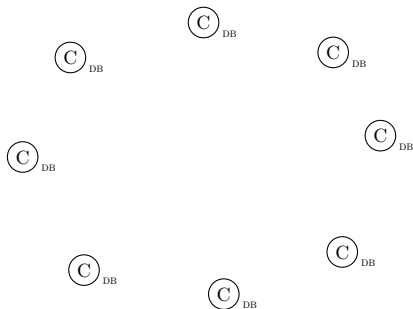
<https://gitlab.ida.liu.se/TDDD25/labs/>

- doc/
- src/
  - lab0/
  - lab1/
  - lab2/
  - lab3/
  - lab4/
  - lab5/
  - modules/
    - Common/
    - Server/

# Implementation

- Multi-threaded object-oriented code in Python 3
- Communication via objects serialized in JSON
- Data transfer through TCP sockets

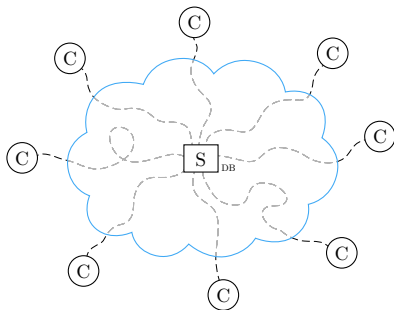
## Lab 0: Standalone Database



- Local database for each client (C)
- **TODO**: complete the implementation of the read and write operations of the database (DB)

<https://gitlab.ida.liu.se/TDDD25/labs/raw/master/doc/lab0.pdf>

# Lab 1: Client-Server Database

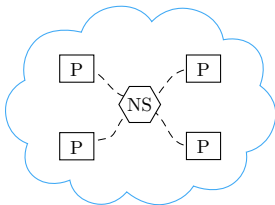


- Centralized database (DB)
- **TODO**: complete the implementation of the client (C) and server (S) communication mechanism

<https://gitlab.ida.liu.se/TDDD25/labs/raw/master/doc/lab1.pdf>



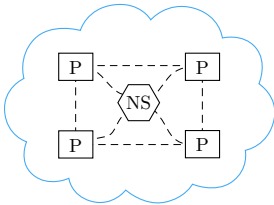
## Lab 2: Object Request Broker



- Name service (NS) and object request broker (ORB)
- Abstract away the communication part of the functionality
- **TODO:** complete the implementation of the ORB

<https://gitlab.ida.liu.se/TDDD25/labs/raw/master/doc/lab2.pdf>

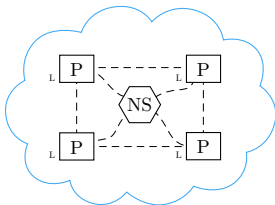
## Lab 3: Peer-to-Peer Communication



- Smart mechanism for keeping track of peers (P)
- **TODO**: complete the functionality dealing with the peers joining or leaving the system

<https://gitlab.ida.liu.se/TDDD25/labs/raw/master/doc/lab3.pdf>

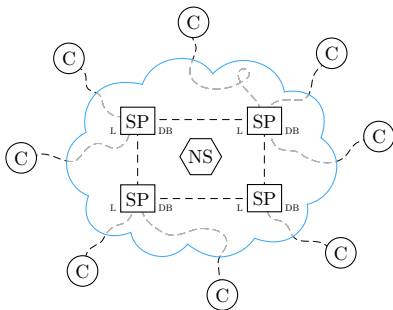
## Lab 4: Distributed Locks



- Distributed mutual exclusion to control concurrent operations
- **TODO**: complete the implementation of the second Ricart–Agrawala algorithm (L)

<https://gitlab.ida.liu.se/TDDD25/labs/raw/master/doc/lab4.pdf>

## Lab 5: Client-Server Database with Replicas



- Everything together
- **TODO**: complete the implementation of the server/peer (SP) using all the previously developed components

<https://gitlab.ida.liu.se/TDDD25/labs/raw/master/doc/lab5.pdf>

# Submission

- No written reports are needed
- Make your code available on GitLab
- Add your teaching assistant as a Reporter
- Demonstrate your solutions in class
- **Attend your group's lab sessions**

Good luck!