

# Pre Meeting - Open Source Lab

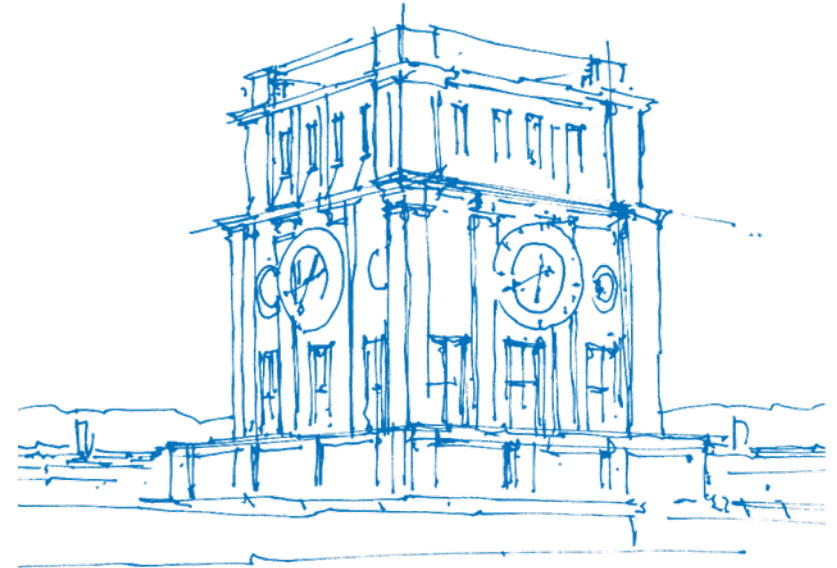
Fabian Sauter, Christian Menges, Alexander Stephan

Technical University of Munich

Department of Informatics

Chair of Connected Mobility

Garching, 07.02.2023



*TUM Uhrenturm*

# Team

## Fabian Sauter, M.Sc.

- [fabian.sauter@in.tum.de](mailto:fabian.sauter@in.tum.de)
- <https://github.com/com8>
- <https://gitlab.com/COM8>
- Popular Programming Languages
  - C++
  - C#
  - C
  - Python
- Misc
  - XMPP Dev
  - Web: <https://uwp.x.org>

## Christian Menges, M.Sc.

- [christian.menges@tum.de](mailto:christian.menges@tum.de)
- <https://github.com/Garfield96>
- <https://gitlab.com/Garfield96>
- <https://gitlab.lrz.de/ga87nad>
- Popular Programming Languages
  - C/C++
  - Go
  - Rust
  - Python
  - Ruby

## Alexander Stephan

- [alexander.stephan@tum.de](mailto:alexander.stephan@tum.de)
- Bachelor Informatik
- <https://github.com/alexanderstephan>
- <https://gitlab.com/alexanderstephan>
- <https://gitlab.lrz.de/alexanderstephan>
- Popular Programming Languages
  - C/C++
  - Go
  - Java
  - TypeScript

Contact: [opensource@cm.in.tum.de](mailto:opensource@cm.in.tum.de)

# Important Information

- **Website:** <https://www.ce.cit.tum.de/cm/teaching/summer-term-2023/open-source-lab/>
- **Registration:** 09.02.–14.02.2023 using [matching.in.tum.de](https://matching.in.tum.de)
- **Duration:**
  - Weekly theory lectures at the beginning.
  - Later biweekly meetings to check student's practical progress.
  - Time slots will be decided in cooperation with the participants (at least one session takes place in the evening (6 - 8 pm)).
  - **All lectures and meetings will be held online (virtual) using BBB, attendance is mandatory!**
- **Module ID:** IN0012 / IN2106 (Bachelor and Master practical course)
- **ECTS:** 10
- **Capacity:** 20 students (Two groups)
- **Language:** English (or German, in case all participants agree)

# Theory Lectures

The following topics will be covered:

- General introduction
- Git basics
- GitHub, GitLab, ...
- Licenses
- Utilities like Static Analyzer, Linter, etc.
- CI/CD

# Requirements for Projects

- Open Source (must be open-contribution)
- No "personal" projects
- Active user base (latest commit not older than 1 year)
- At least 50 active users (1000+ recommended)
- You must work on at least 1 project, which is not directly related to TUM (TUM-Dev, TUM-Live, ...)
- The project should be large enough to last you at least a couple of weeks.
- Contributions can be new features, bug fixes or performance improvements (PRs fixing typos are not accepted by us)
- Without previous experience working on extremely large and complex projects, such as GCC, Linux Kernel, Postgres, etc. is not recommended.
- We recommend to pick one of the projects listed below, since these projects are in widespread use and we can help you in case of problems.

# Grading

- No final report required, all interesting topics should be described in the documentation of the projects or the corresponding PR.
- LOC not relevant
- Intermediate presentation (no fancy slides required)
- Code quality
- Interaction with the community
- Communication with advisors

**Note:** Spamming or creating other unnecessary burden to the community will result in failing the course immediately. Remember, **you are representing TUM.**

# Important Information - Application

**Only apply** if you

- ... have a good understanding of at least two programming languages
- ... have strong problem solving skills
- ... are willing to invest a significant amount of time in programming and problem solving, also during the semester break

# Important Information - Interested?

Send a mail to [opensourcelab@cm.in.tum.de](mailto:opensourcelab@cm.in.tum.de) containing...

- ... a few lines, why you are interested in doing this lab (~100 words)
- ... your previous programming experience (bullet points)
- ... your GitHub/GitLab/. . . profile (if available) or a description why you don't need an account, e.g. because you want to work on project using mailing lists
- ... an outline of projects you would like to work on in the first six weeks of the lab. Please provide links to issues if available.

**Mails sent from external mail services get ignored** (exception: exchange students without TUM mail account)



# Example Projects - TUM Developers

## Die TUM Campus App

Dein Begleiter für den Uni-Alltag



Die TUM Campus App ist der mobile Begleiter für alle Studierenden und Mitarbeitenden der Technischen Universität München. Sie hilft dabei, schnell und einfach an aktuelle Informationen zu kommen und so den Uni-Alltag besser zu organisieren:

- Mensa-Speisepläne
- U-Bahn-Abfahrtszeiten
- Kalender mit Vorlesungsterminen
- Chatrooms
- Suche nach Personen und Räumen
- Noten
- Studiengebühren
- News von der TUM
- Lernräume

Die App ist für alle TUM-Mitglieder gedacht, kann aber auch ohne TUM-Kennung verwendet werden.

### eat-api

Forked from srehowald/eat-api  
Simple static API for some (student) food places in Munich.

● HTML MIT 🍷 8 ☆ 7 🕒 4 (1 issue needs help) 🛠️ 1  
Updated 5 hours ago

### CalendarProxy

Proxy for the TUM iCal export to remove clutter

php tum proxy tum-ical-export  
📦 1 package ● PHP 📄 GPL-2.0 🍷 8 ☆ 18 🕒 0 🛠️ 3  
Updated 6 days ago

### RBGreaterAgain

📦 1 package ● PHP 🍷 3 ☆ 5 🕒 7 🛠️ 1 Updated 14 days ago

#### Top languages

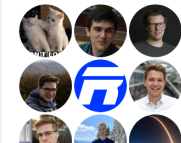
● PHP ● JavaScript  
● Kotlin ● HTML  
● Ruby

#### Most used topics

tum students  
tum-campus

#### People

13 >



<https://github.com/TUM-Dev>

# Example C/C++ Projects

**GNU coreutils** Possible tasks:

- Implement parallel shuffle to allow shuffling of huge data sets (30GB+) in an acceptable amount of time
- Improve performance of sort, etc.

**KDE** [https://community.kde.org/Get\\_Involved/development](https://community.kde.org/Get_Involved/development)

Possible tasks:

- 15 minute bug initiative

**Kompute** - Vulkan Compute wrapper

<https://github.com/KomputeProject/kompute>

Possible Tasks:

- Expose dst/src offsets in recordCopyFrom tensor function
- Support multi-dimensional Tensors

# Example C++ Projects - CPR

```
#include <cpr/cpr.h>

int main(int argc, char** argv) {
    cpr::Url url{"https://..."};
    cpr::Response r = cpr::Get(url,
    cpr::Authentication{"user", "pass"},
    cpr::Parameters{{"anon", "true"},
                    {"key", "value"}});
    r.status_code; // 200
    r.text; // JSON text string
}
```

## C++ Requests: Curl for People

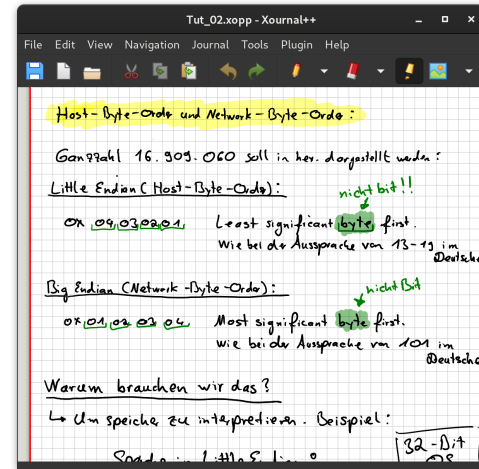
<p><b>C++17/C++20 support</b> <span>Feature</span></p> <p>#382 opened on May 20, 2020 by COM8 <span>🔗</span> CPR 1.7.0</p>	8
<p><b>Separate host and peer verification for VerifySsl option</b> <span>Feature</span></p> <p>#310 opened on Apr 14, 2019 by Liastre <span>🔗</span> CPR 1.7.0</p>	1
<p><b>System provided proxy is ignored by setting CURLOPT_PROXY to "" when no proxy is set</b> <span>Bug</span> <span>Feature</span></p> <p>#300 opened on Nov 30, 2018 by EliSchleifer</p>	1
<p><b>Please make a NuGet package. That can help add your library to project easier</b> <span>Build</span> <span>Feature</span></p> <p>#297 opened on Nov 5, 2018 by CODER591</p>	4
<p><b>Is there a way to get the full URL for a request?</b> <span>Feature</span></p> <p><span>Question</span></p> <p>#270 opened on Apr 29, 2018 by p0fi <span>🔗</span> CPR 1.7.0</p>	6
<p><b>Better Session::SetHeader() implementation</b> <span>Feature</span></p> <p>#265 opened on Mar 28, 2018 by kaplomat</p>	
<p><b>compress transfer failed</b> <span>Feature</span> <span>Tests Needed</span></p> <p>#211 opened on Aug 3, 2017 by guifengcheng</p>	2

<https://github.com/libcpr/cpr>

# Example C++ Projects - Xournal++

- Written in C/C++ and GTK3
- Possible Tasks:
  - One click copy of version information
  - Pressure filtering: Ignore pressure smaller than some set amount
  - Enhancement: add 'Duplicate' as option under 'Journal' menu
  - Upgrade to GTK4 (advanced)

Xournal++ is a handwriting notetaking software with PDF annotation support.



<https://github.com/xournalpp/xournalpp>

# Example Go Projects

**Go** compiler, stdlib, etc.

<https://github.com/golang/go>

**CNCF** The cloud native computing foundation hosts a lot of interesting programs, many of them written in Go

<https://www.cncf.io/projects/>

- Kubernetes

E.g. enhance kubectl top command: <https://github.com/kubernetes/kubectl/issues/1214>

**Traefik** A cloud native application proxy

<https://github.com/traefik/traefik>

**Awesome Go** A list of interesting Go projects for all kinds of applications

<https://github.com/avelino/awesome-go>

# Example Rust Projects

[librsvg](#) SVG rendering library

Used by Wikimedia project, ffmpeg, ...

Possible tasks:

- implement features of SVG 2.0
- enhance performance

<https://gitlab.gnome.org/GNOME/librsvg>

[uutils/coreutils](#) Rewrite of GNU coreutils in Rust

<https://github.com/uutils/coreutils>

# Example Ruby Projects

**GitLab** Possible tasks:

- More than 43,000 issues waiting for you

<https://gitlab.com/gitlab-org/gitlab>

**Nokogiri** XML, HTML, SVG, etc. processing library

One of the most used Ruby libraries (in Ruby terminology: gem)

<https://github.com/sparklemotion/nokogiri>

# Previous Projects

- TUM-Dev: eat-api
- TUM Campus App - Android/iOS
- TUM-Live
- Zulip
- Synapse (Matrix-Server)
- Element (Matrix-Client)
- Librsvg
- BigBlueButton
- spot (Spotify client)
- Cataclysm: Dark Days Ahead
- CPR
- Swift-DocC
- Mealie
- j-lawyer-org
- calibre-web
- Xournal++
- Haaukins
- Jina
- raylib
- Gnome Project
- utils/coreutils
- Chromium
- Groovy
- Atrium
- NewPipe
- Owncast
- Anki-Android
- Django
- scikit-learn
- ...



# Next steps

1. Drop us an email to increase your chance of being matched.
2. Register the course in the matching system.
3. If you are successfully matched to our course, we will send you an email with further information.
4. Familiarize yourself with the projects during the semester break, especially for larger projects.

Contact: [opensourcelab@cm.in.tum.de](mailto:opensourcelab@cm.in.tum.de)

Remember to use TUM mail address!