

# Collaborative text and code editing on high-resolution displays

“Kollaborative Text- und Codebearbeitung auf hochauflösenden Displays”

**Prüfer:** Prof. Dr. Thomas Ertl  
**Betreuer:** Florian Frieß, (Christoph Müller)  
**Bearbeiter:** TBD  
**Beginn:** TBD

## Background

Source code of large software projects is often difficult to understand and to explore since it is distributed over many documents. Displays of laptops and workstations usually do not have enough space to display them all at once. There are already tools and extensions that help explore and understand source code, e.g. the *Code Map* in *Visual Studio*, but none of them uses large high-resolution displays. Such displays can show many more documents at once, but have limited input possibilities.

## Task Description

The main goal of the thesis is to develop a collaborative code editor; it needs to work with standard workstations, mobile devices and high-resolution displays. A central component is the underlying text editor that is required to be able to handle multiple concurrent interactions (insertions, deletions, etc.). For code editing it requires extensions like syntax highlighting, word completion, go-to functions, auto formatting, renaming, and more. Additionally, assistive in-situ visualisation, e.g. overlays like call graphs or dependency graphs should be displayed to help the user understanding the code base. Reaching this goal includes several sub-tasks.

The first objective is the development of the text editor. It consists of two parts, the first part being the central server that stores the documents and applies the changes to them in real time. The second part is the client that needs to be flexible, so that it can be used on many different devices, ranging from mobile devices to high-resolution displays.

The second objective is to extend the text editor by adding extensions for multiple programming languages. These extensions should support syntax highlighting, word completion, go-to functions, auto formatting, renaming, and more. For example, this can be done with the Language Server Protocol<sup>1</sup> (LSP) server and client.

The last objective is to utilise the large screen space of the high-resolution display to show assistive visualisations like call graphs or dependency graphs. The data for these graphs needs to be extracted from the documents on the server.

---

<sup>1</sup> <https://langserver.org/>

## Goals

The basic goals of the thesis are:

- Create a client/server text editor that allows multi-user changes in real time.
  - o All documents should be stored on a central server.
  - o Should allow multiple session (editing of different documents) at the same time.
- Create a client that shows the document and sends local changes to the server.
  - o The client needs to run on different devices ranging from mobile devices to high-resolution displays. High-resolution displays might be driven by a cluster of multiple computers, i.e. the client needs to be able to display its content as multiple tiles.
- Extend the server to support extensions for one or more programming languages.
  - o Support for C++ is a requirement.
  - o Support of syntax highlighting is a requirement.
  - o Support of at least one of word completion, go-to-functions, auto formatting and renaming is a requirement.
  - o The extensions need to be usable for every client in real time.
- Add assertive in-situ visualisations to the client.
  - o Extract data from the document on the server and send it to the client in real time.
  - o Add overlay visualisation for call graphs.

Once the basic goals are reached, the following tasks may be dealt with:

- Add additional assistive features like word completion, go-to-functions, auto formatting and renaming.
- Add additional visualisation overlays like dependency graphs or highlighting of variable definitions and variable uses.
- Add support for more languages.