

Client/Server Architecture

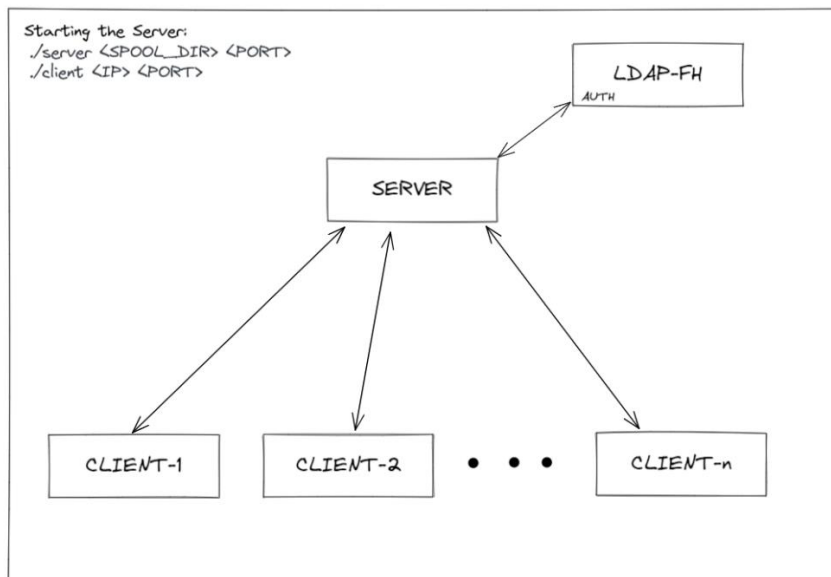
Server:

Our server supports multiple clients at the same time, we have solved this by running each client in its own process.

For a client to enter a command it must be authenticated once. The authentication works with the data of the UAS, we do this via LDAP check, if a client fails to provide the right credentials within 3 tries, the IP will be blocked for one minute.

Client:

In the client only the commands are entered and the corresponding output is displayed. The whole logic is built into the server. If the client is not authenticated only the commands "QUIT" and "LOGIN" can be used.



Used technologies

We used following tools

- git (version control)
- VS Code LiveSharing (Pair programming)
- C/C++
- Make

Development Strategy

Most of the time spent developing was via Pair Programming. It made us more efficient than when solo developing which also sped up development time. Since we were able to find the root cause of errors significantly faster.

Adaption

- Further abstraction, more header files
- Currently there can not be 2 E-Mails with the same subject => use unique ids
- There is a workaround for the 3-Login attempts, an IP-can disconnect after 2 tries and reconnect again as a new client