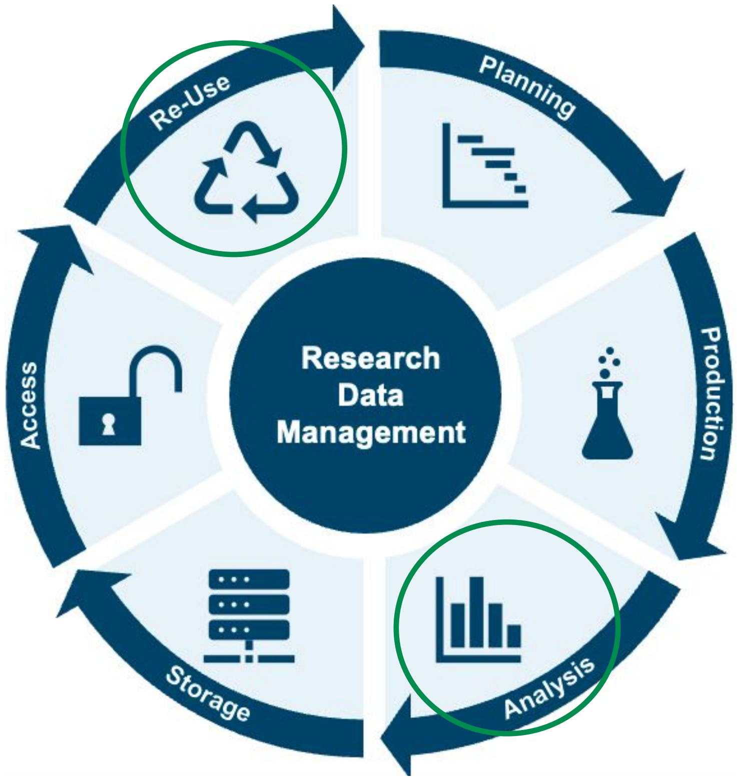
Betty‘s (Re)Search Engine

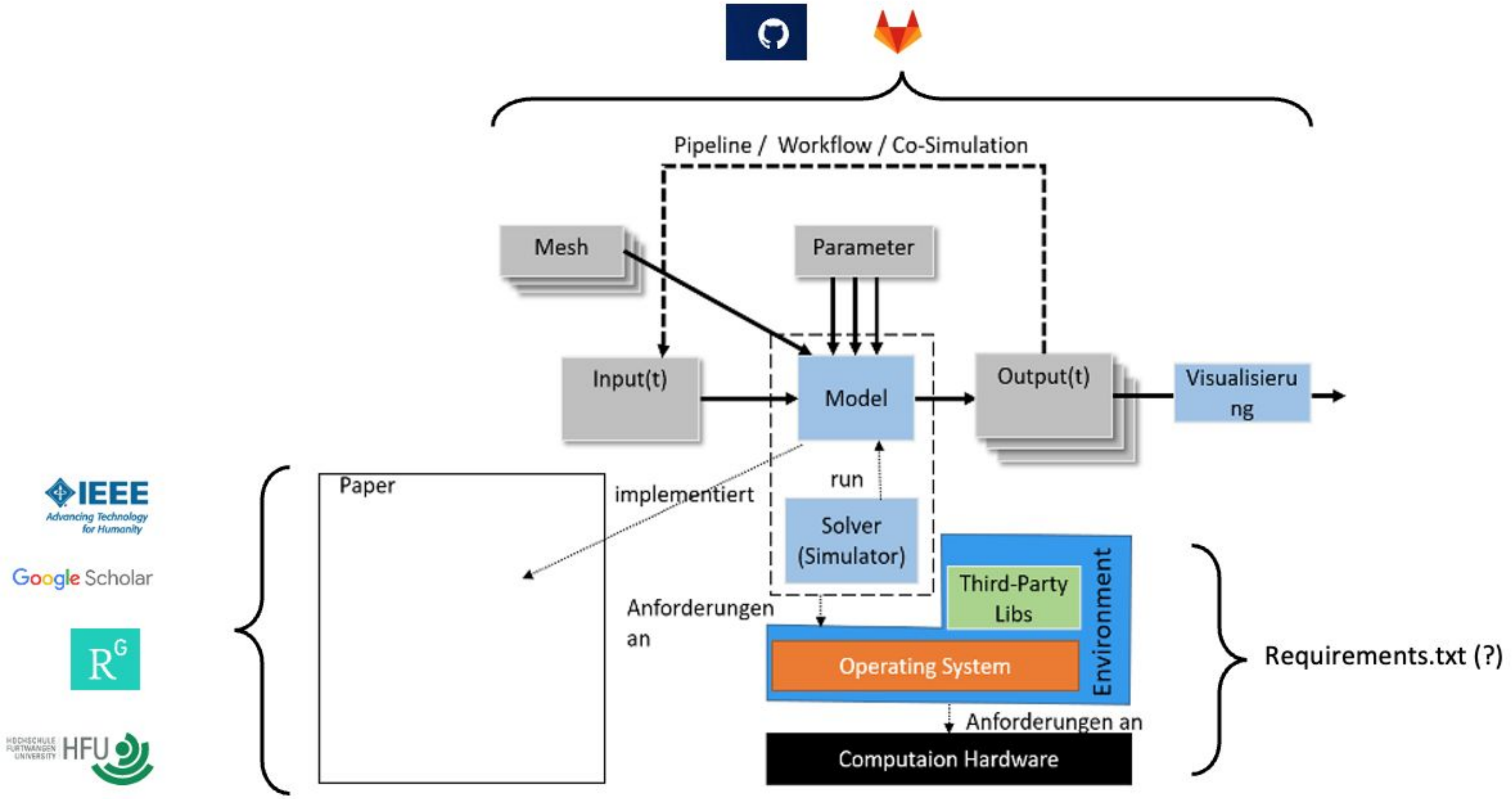
A client-based search engine for research software stored in repositories. - Demonstration

Vasiliy Seibert,
vasiliy.seibert@tu-clausthal.de

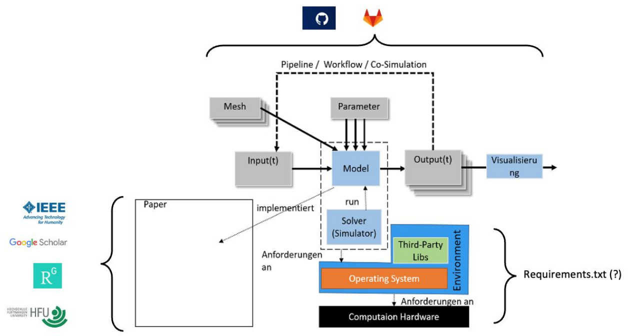
13.09.22



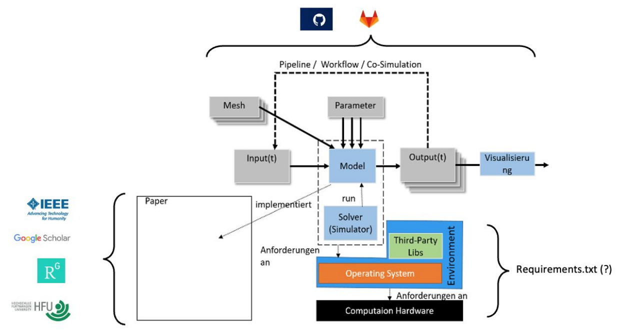
Motivation:


Pipeline / Workflow / Co-Simulation
Mesh
Parameter
Input(t)
Model
Output(t)
Visualisierung
Paper
implementiert
run
Solver (Simulator)
Third-Party Libs
Environment
Operating System
Anforderungen an
Anforderungen an
Computaion Hardware
Requirements.txt (?)
IEEE
Google Scholar
HFU

Motivation:

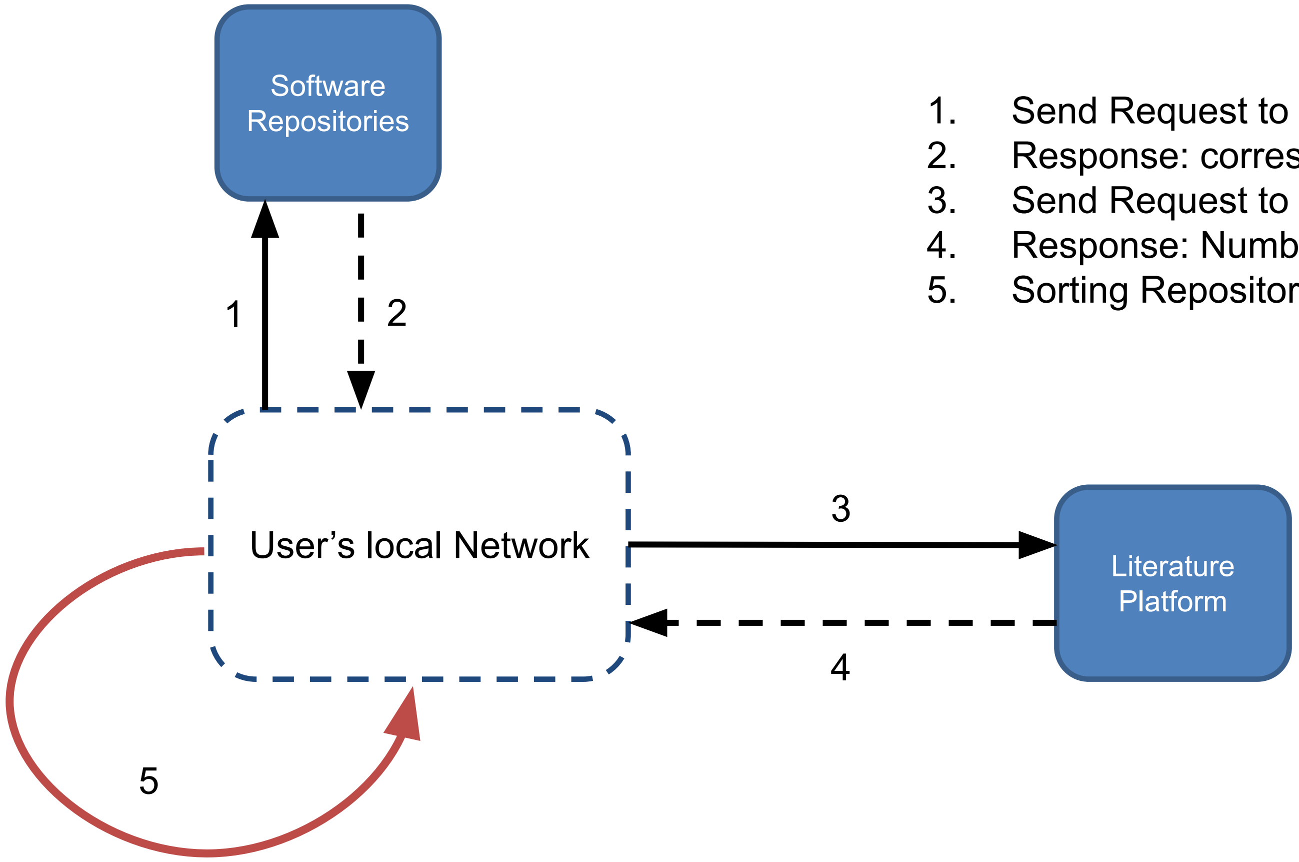


(…)



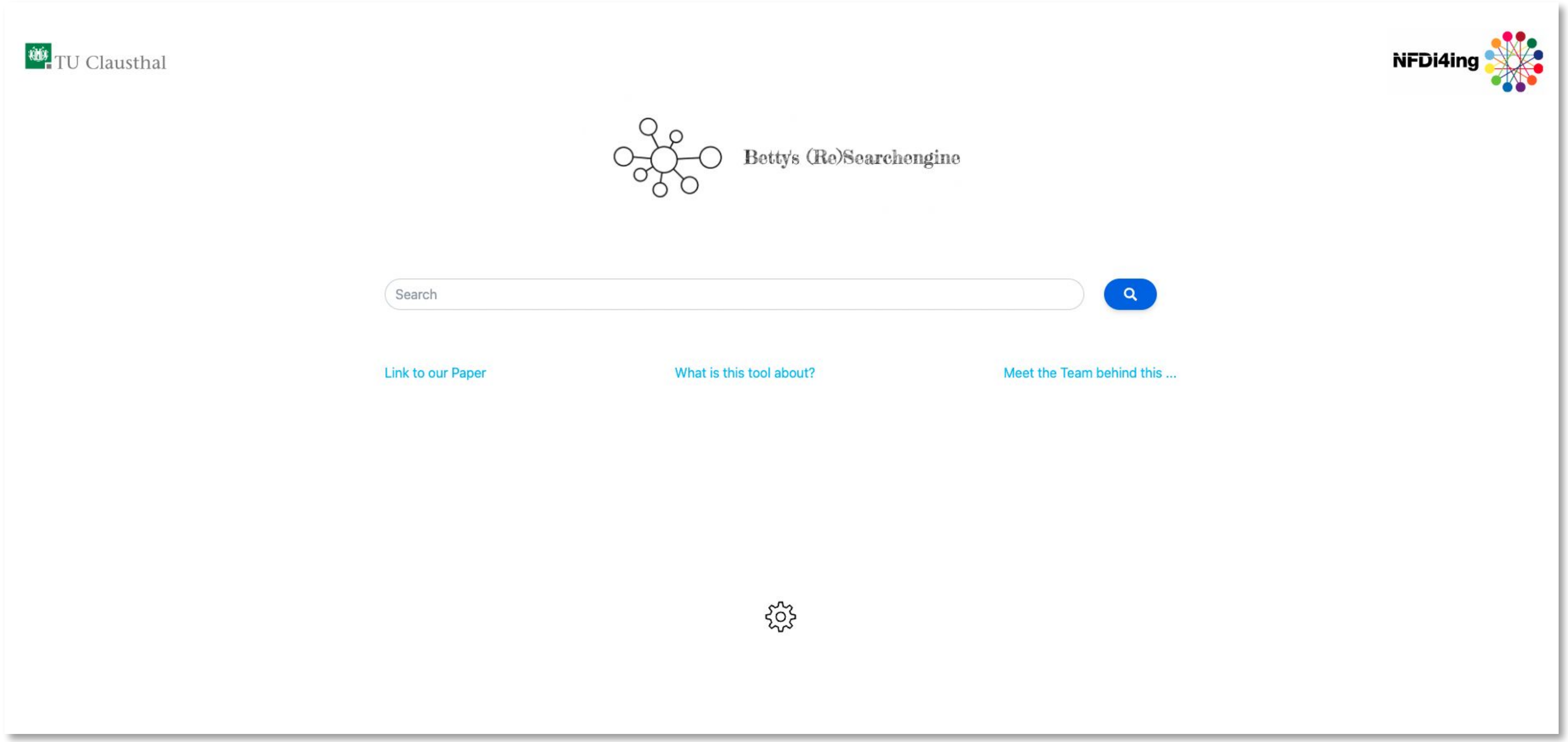
- The process of finding research software for a specific purpose is inefficient.
- One must look through a publication before being able to hope for a corresponding link to a repository where the research software is then stored.
- The described way of searching also does not allow applying preferences (e.g., only searching for research software written in Python).

Betty’s (Re)Searchengine



1. Send Request to Software Repositories
2. Response: corresponding repositories to a given Search Query
3. Send Request to Literature Platform
4. Response: Number of citations for a Repository
5. Sorting Repositories according to their number of citations

Demonstration



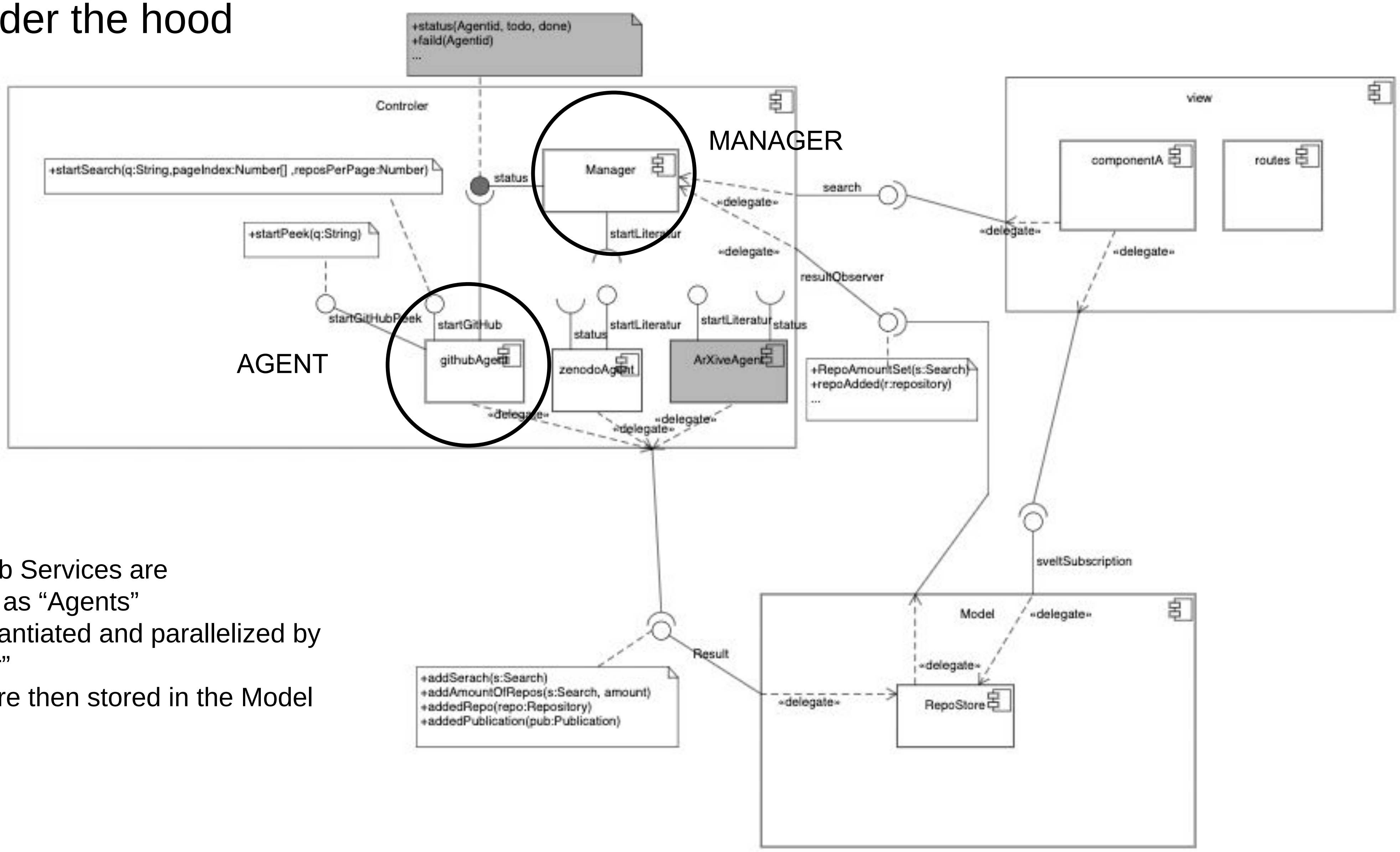
TU Clausthal

Institut für Software and Systems Engineering

Challenges (1 / 2)

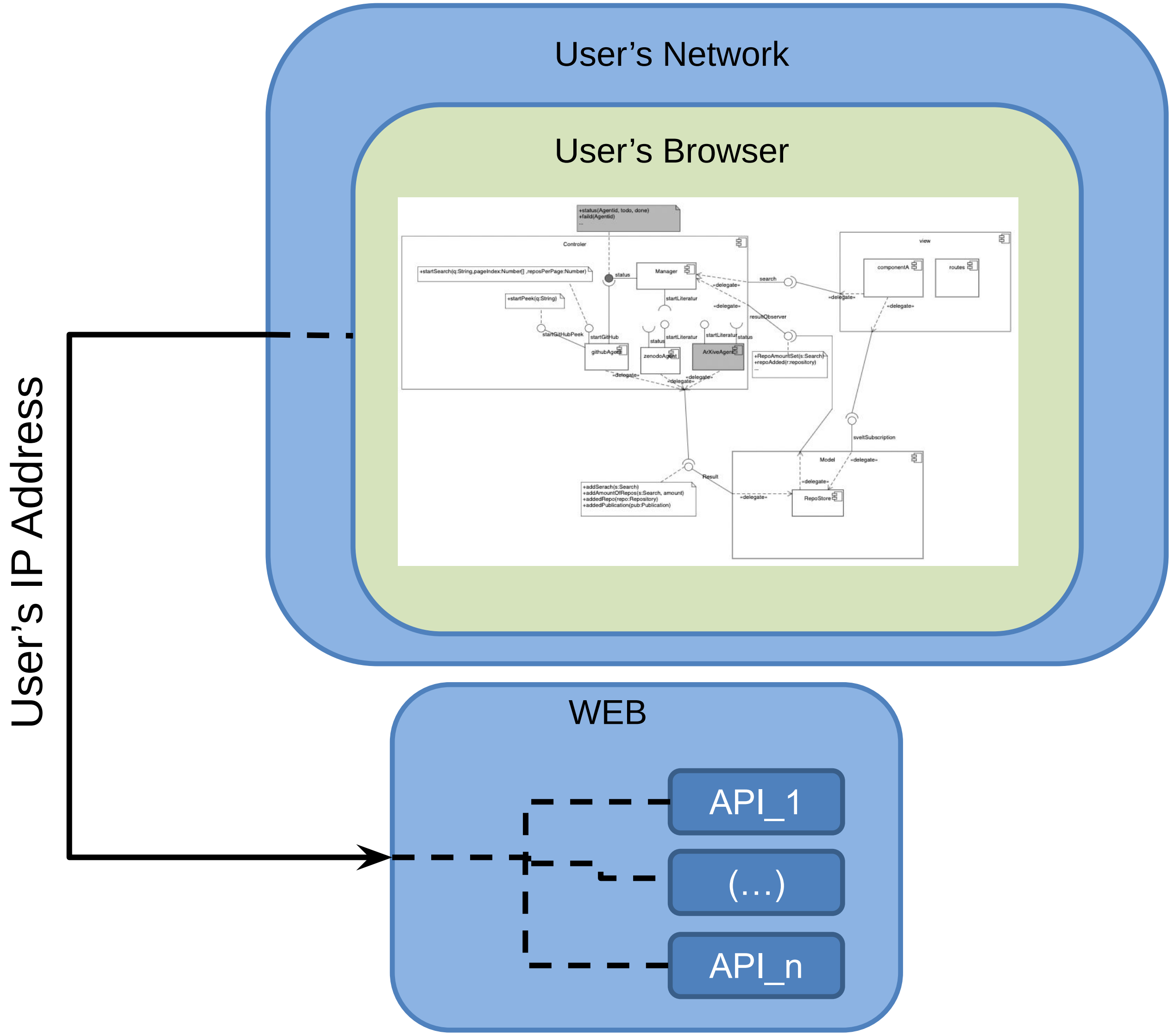
- Rate Limitations:
  - Every Service defines its own Rate Limitations.
  - Betty’s (Re)Search Engine needs to deal with all of them while maintaining a sound workflow.
- Scalability:
  - Betty’s (Re)Search Engine needs to be extendable with additional Services
- Parallelization:
  - A Search Process cannot be too time consuming, or else the user would lose interest

Whats under the hood



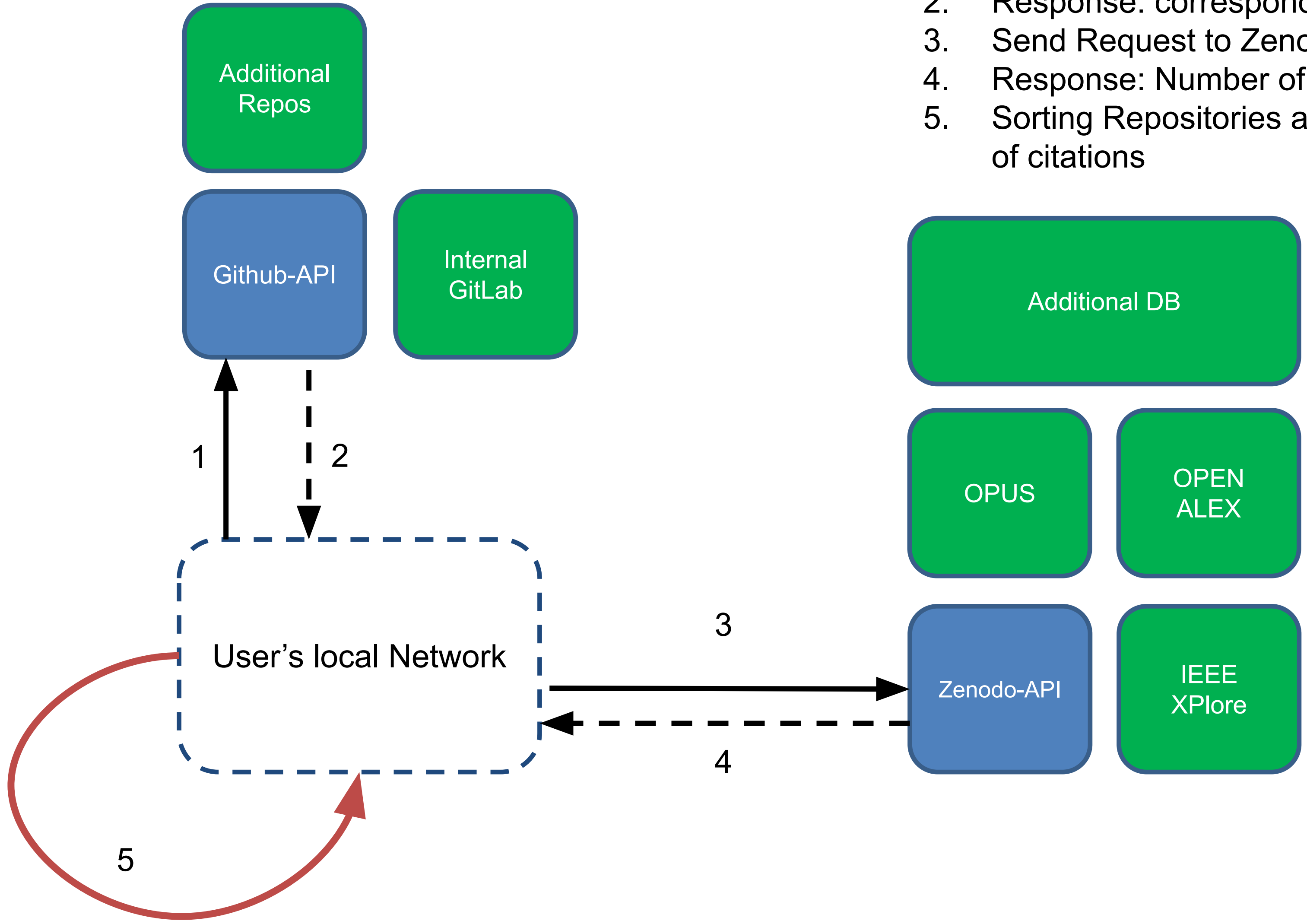
- APIs and Web Services are implemented as “Agents”
- They are instantiated and parallelized by the “Manager”
- The results are then stored in the Model

Betty’s (Re)Searchengine



Whats next?

1. Send Request to Github API
2. Response: corresponding repositories
3. Send Request to Zenodo-API
4. Response: Number of citations for a Repository
5. Sorting Repositories according to their number of citations



Challenges (2 / 2)

- Computational Cost:
  - Multiple Processes are running within the user’s browser.
- Linking Repositories to Research Papers:
  - Currently we’re linking repositories to their corresponding research by searching explicitly for repositories that have a DOI in their README.
  - This approach however results in a limited amount of Repositories that we receive from GitHub.
  - Possible Solution: Conducting Full Text Search on external Databases
    - But then again we would have to deal with a great number of repositories that got nothing to do with research

TU Clausthal

Institut für Software and Systems Engineering

Literatur

[1] M. Wilkinson et al., "The FAIR Guiding Principles for scientific data management and stewardship", Scientific Data, vol. 3, no. 1, 2016. Available: 10.1038/sdata.2016.18 [Accessed 19 August 2022].
[2] "Papers with Code - The latest in Machine Learning", Paperswithcode.com, 2022. [Online]. Available: https://paperswithcode.com/ ↗. [Accessed: 19- Aug- 2022].
[3] "Zenodo - Research. Shared.", Zenodo.org, 2022. [Online]. Available: https://zenodo.org/ ↗. [Accessed: 19- Aug- 2022].
[4] "Semantic Scholar – AI Powered Research Tool", semanticscholar.org, 2022. [Online]. Available: https://www.semanticscholar.org/ ↗. [Accessed: 19- Aug- 2022].
[5] "IEEE XPLORE", ieeexplore.ieee.org, 2022. [Online]. Available: https://ieeexplore.ieee.org/Xplore/home.jsp ↗ [Accessed: 19- Aug- 2022].