



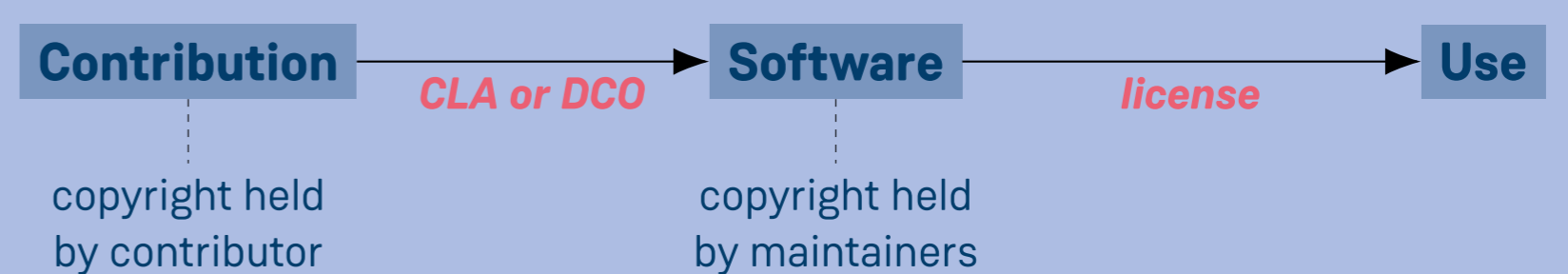
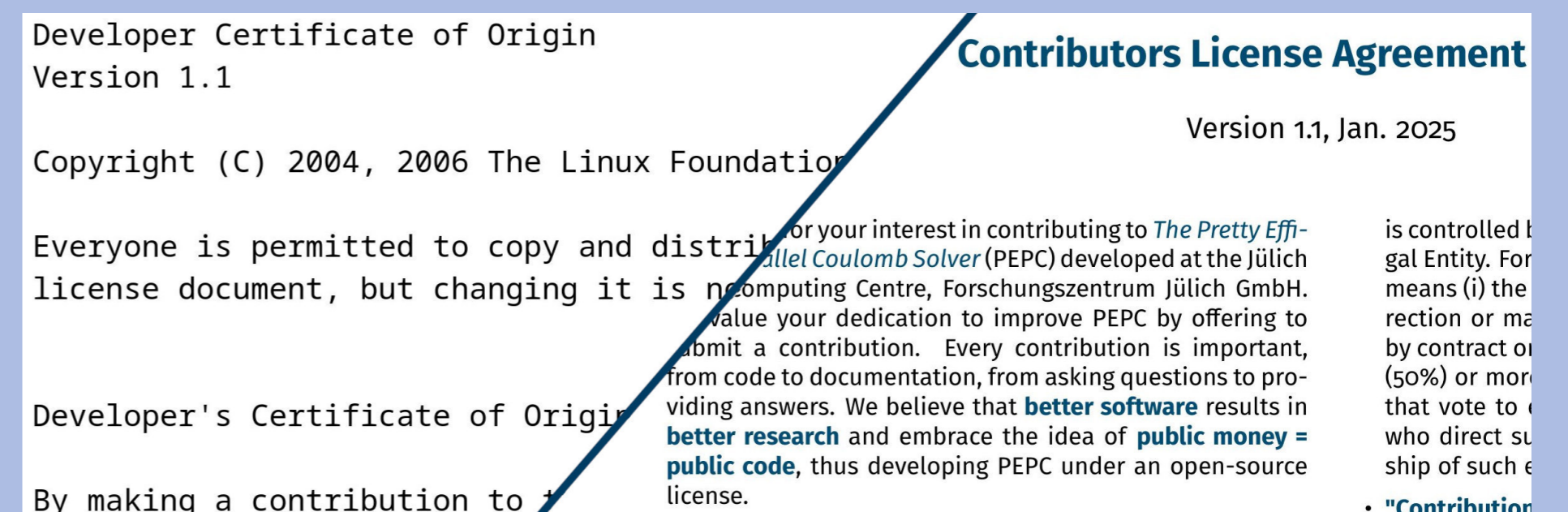
Inbound licensing with ease

Dirk Brömmel, Robert Speck | Jülich Supercomputing Centre | Forschungszentrum Jülich GmbH

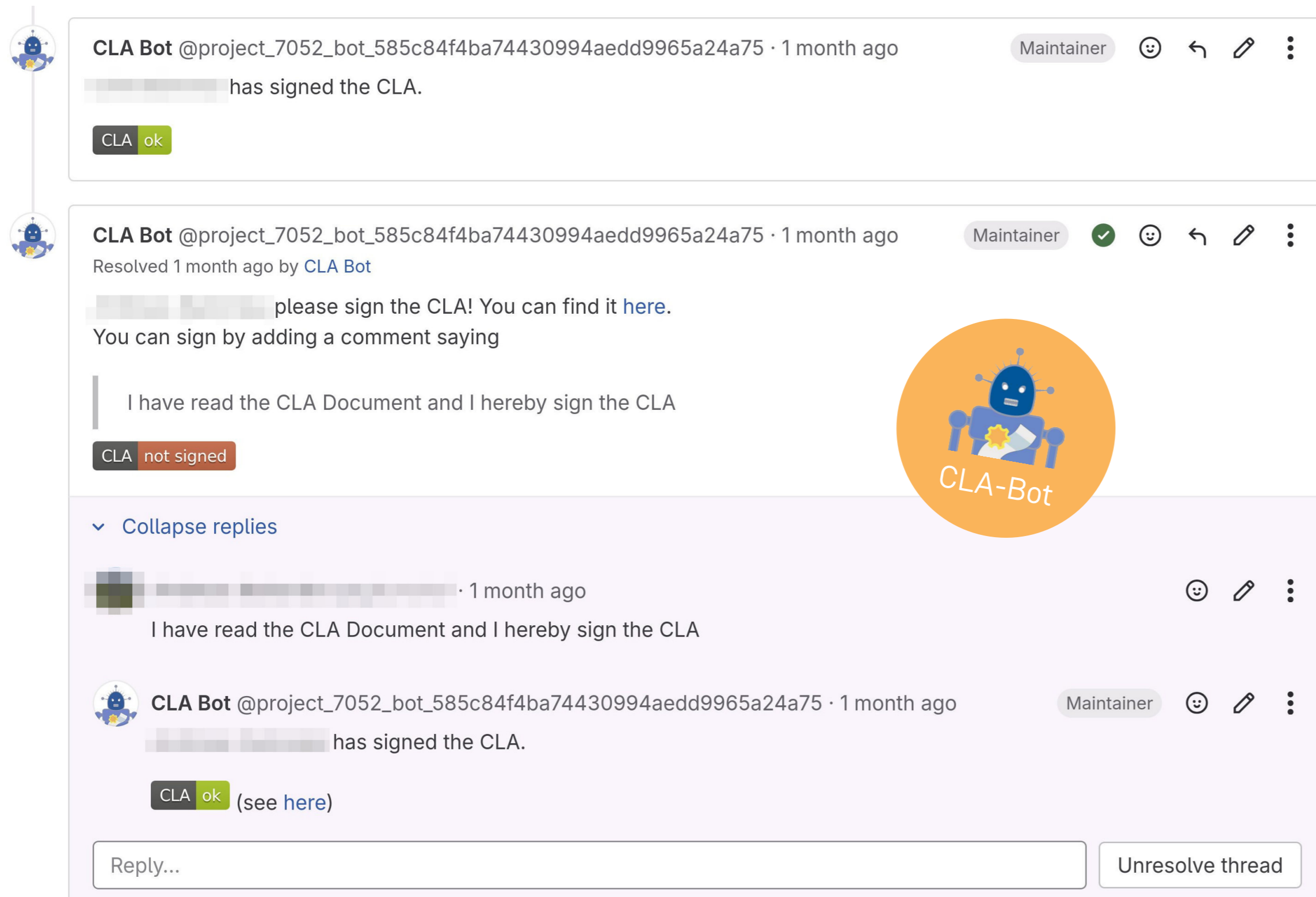
Contact: {d.broemmel, r.speck, rse}@fz-juelich.de | Website: https://go.fzj.de/CLA-Bot



- You have a **license** for your code
 - are contributions properly included?
- Tired of keeping track and reminding **contributors** to **sign agreements**, checking up on them?
- You are not on GitHub and miss e.g. the cla-assistant to **keep things simple**?
- Does your GitLab tier support the necessary capabilities?



Legal disclaimer: We are referring to 'technical ease' here. For your legal requirements, check with your institution's legal department!



What 'it' does

- A 'bot' will check contributors are ok to contribute
- Contributors will be reminded how to agree to your guidelines
- Once they comply, the bot will clearly mark agreement and close discussions
- If set up to do so, the bot will record agreements for future reference

What we strive for

- No additional servers, based on CI/CD pipelines
- Used via CI/CD components to keep it simple
- Users should be in control of where data is kept
- Open development – we are looking for interested parties to test or contribute

Where we are

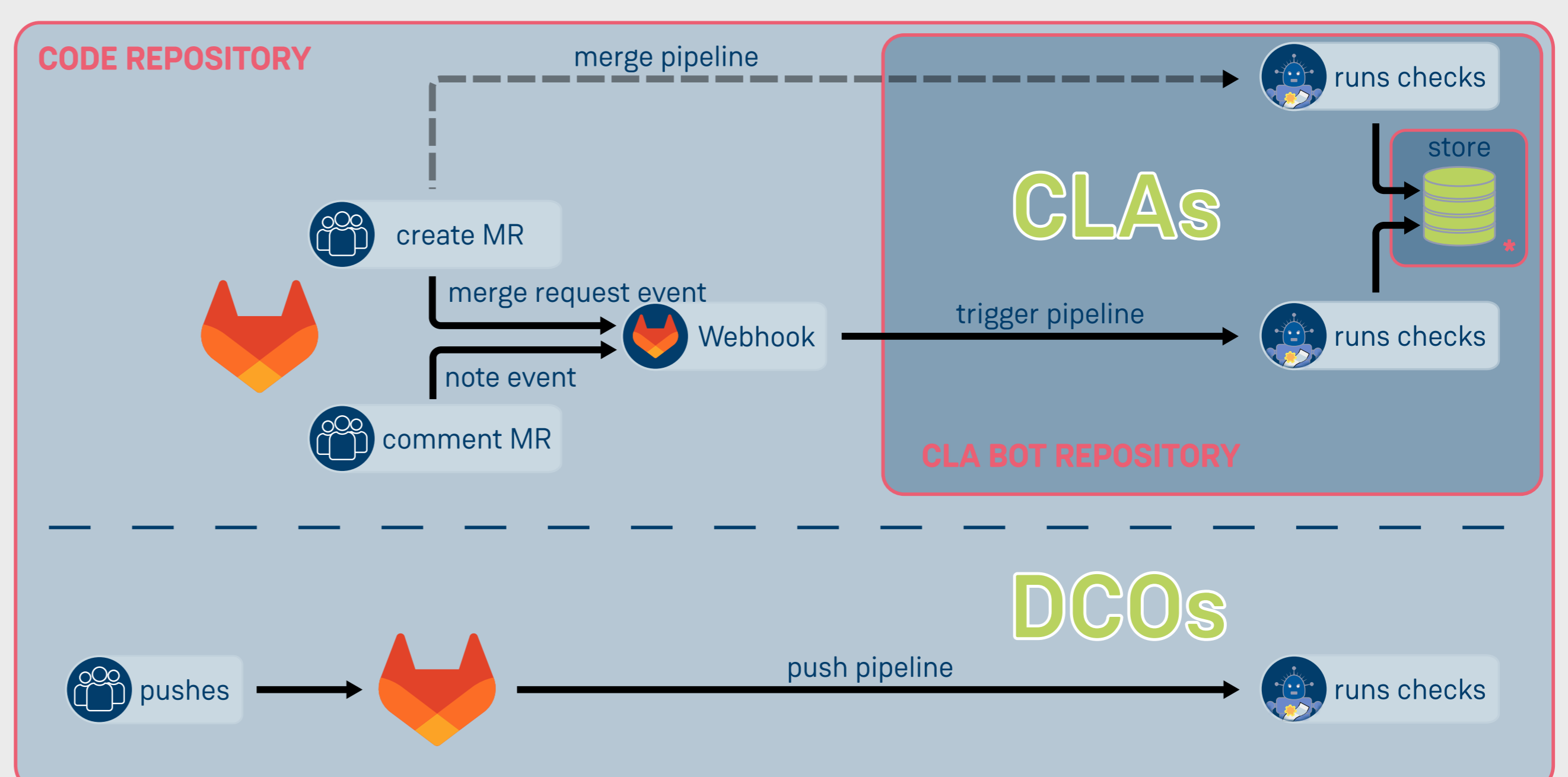
- First working version ready for testing
- Still some way to go, e.g. documentation and tests
- Not feature complete – what would be required?

How does it work?

- CI/CD pipelines on GitLab
- Webhooks to react to events
- Pipeline triggers to start the pipeline
- The bot relies on Python, leveraging gitlab-python and other modules

Constraints

- Cannot block MRs (on GitLab-CE)
- Lots of pipelines, extra commits
- Token management
- Repository setup
- Multiple workflows



* STORAGE REPOSITORY