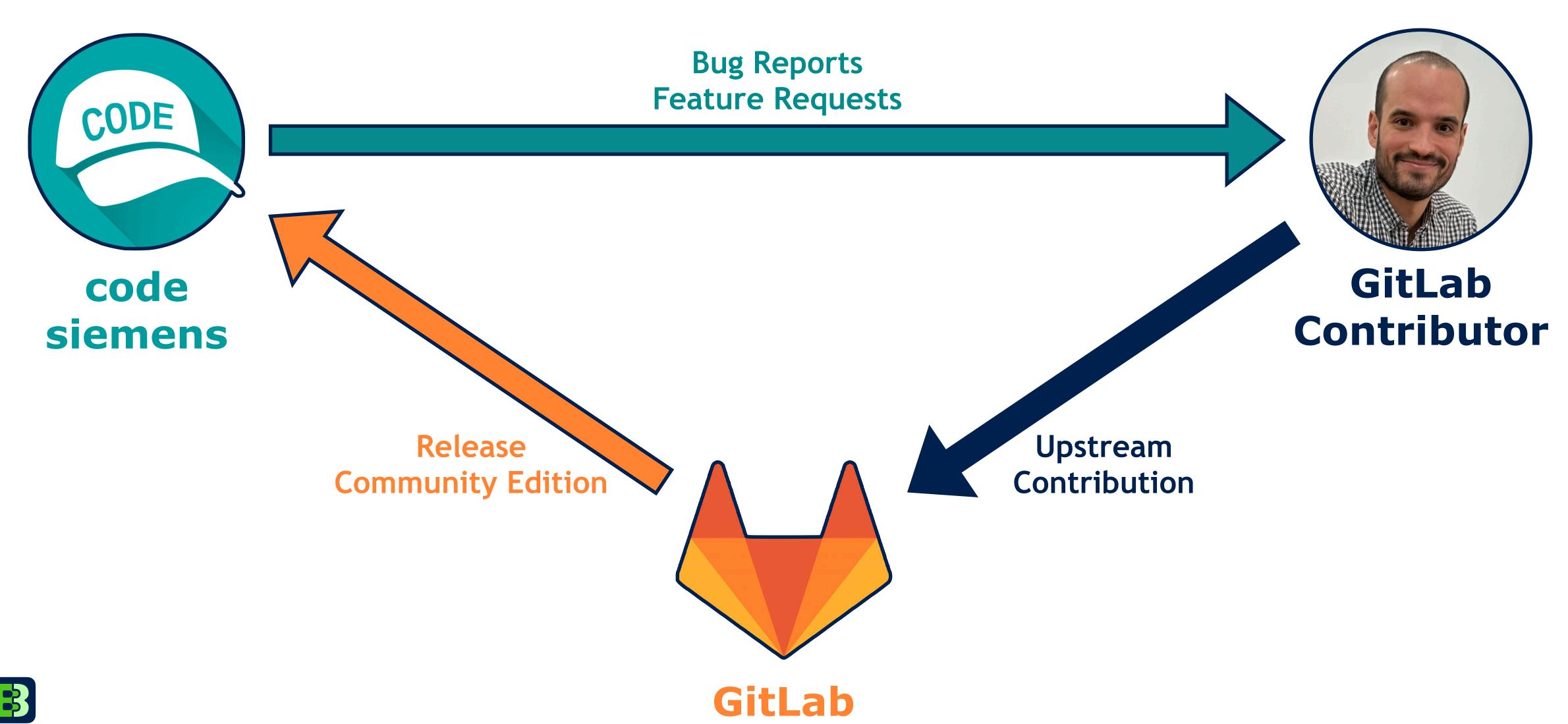
Contributing to GitLab: Protected Packages

Gerardo Navarro (B310 Digital)





Contributing on behalf of Siemens



Step #1: Issue

- Use Google and GitLab to find existing issues or epics
- Unterstand and research the scope of the issue
- Contribute your opinion, thoughts and questions to the discussions

Identify packages as protected to prevent accidental deletion or updates

Open Epic created 3 years ago by Tim Rizzi

Context

You can use the GitLab Package Registry to publish and store packages right alongside your source code and pipelines. A Developer in the respective project may publish new versions of a package using GitLab CI or the command line. However, similar to protected branches, there are many cases where only a Project Admin should be allowed to update or delete a given package.

The following features are already available to protect the release process:

- Protected Branches
- Protected Tags
- Protected Environments
- Protecting Pipeline Settings by hosting .gilab-ci.yml in a separate project

However, there is no way to do the same for packages. This means that we are protecting all of the components responsible for creating a package but not the output.

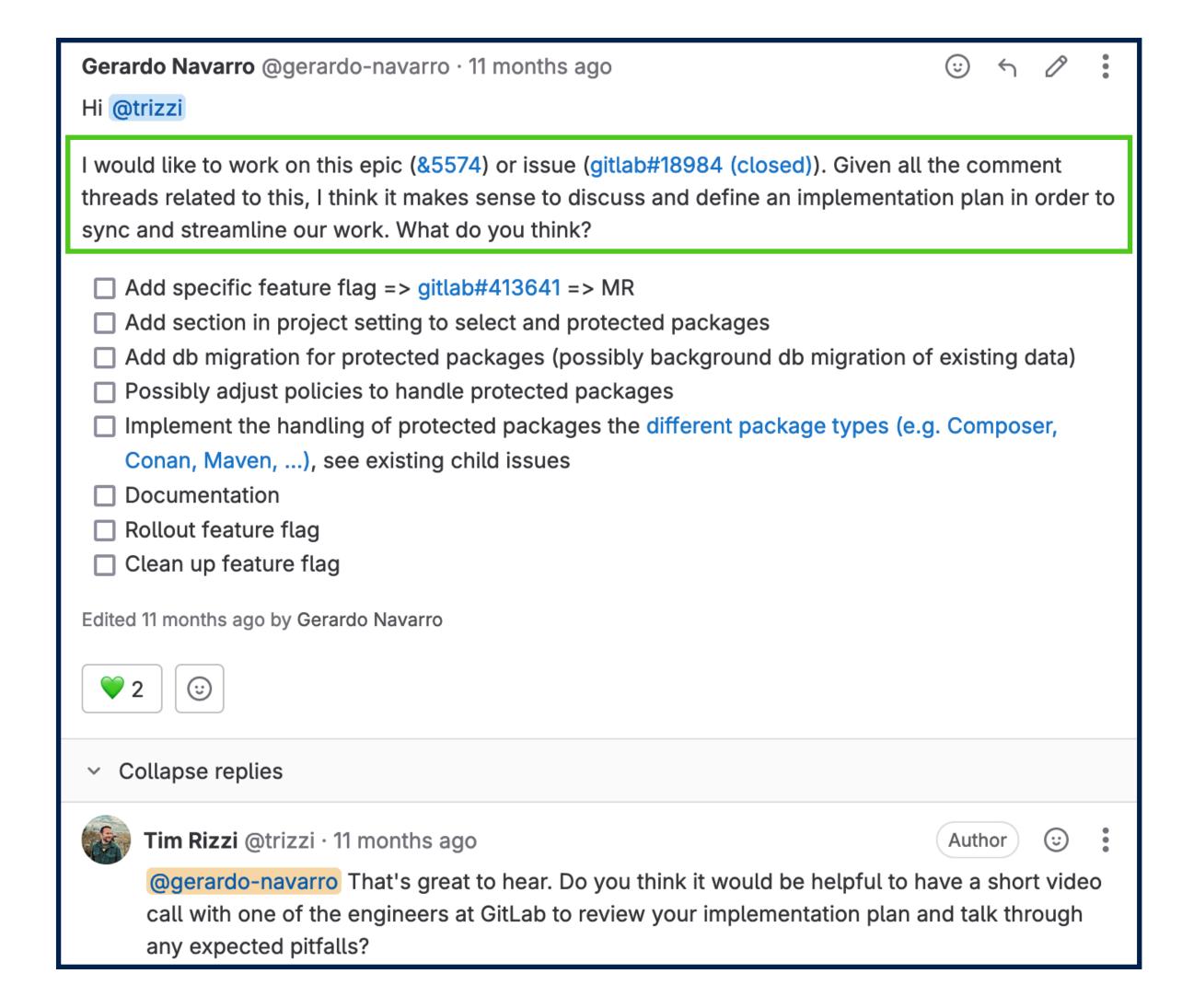
Proposal

For each package manager format add configuration options for restricting all actions except read (create, update and delete) similar to the existing Protected Branches, Protected Tags and Protected Environments.



Step #2: Collaborate

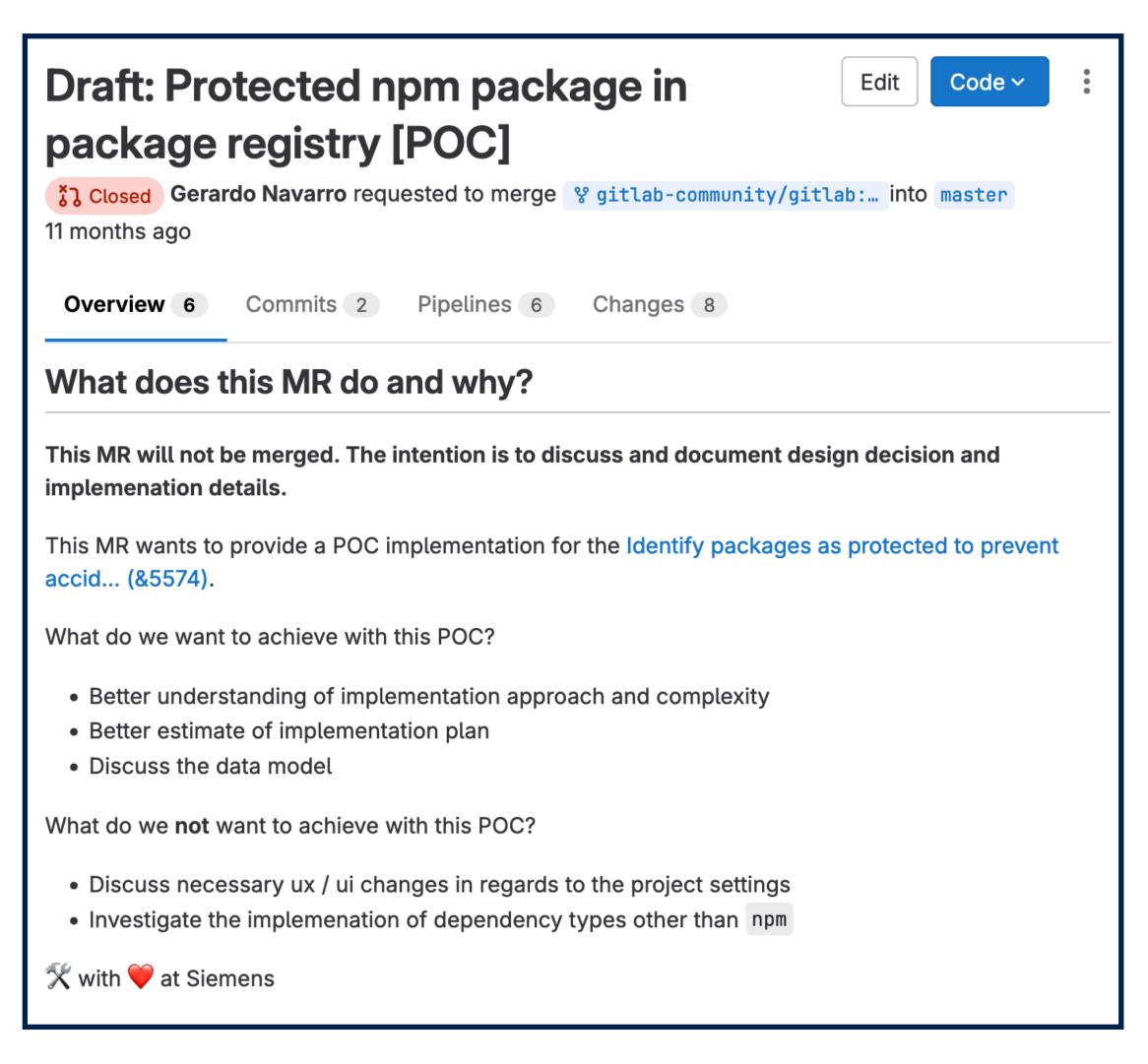
- Express your desire to contribute
- Discuss scope of issue, feature or epic
- Propose implementation plan and next steps





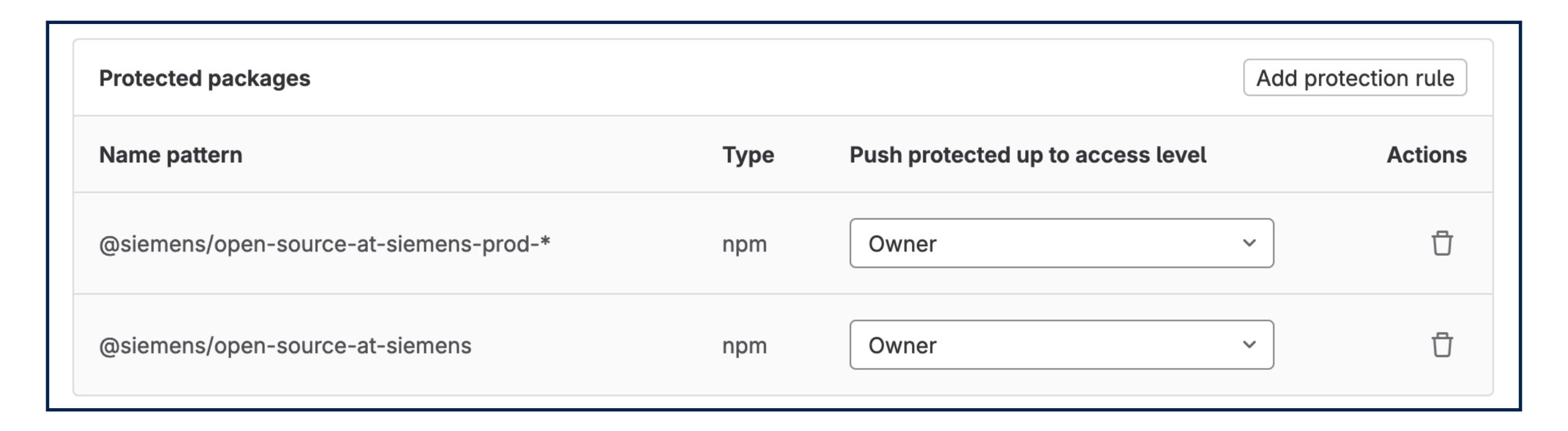
Step #3: Proof of Concept

- Understand necessary effort, expected feature scope and upcoming design decisions
- Focus on essentials functions
- Ask for feedback early on and start a discussion





Step #4: Implement



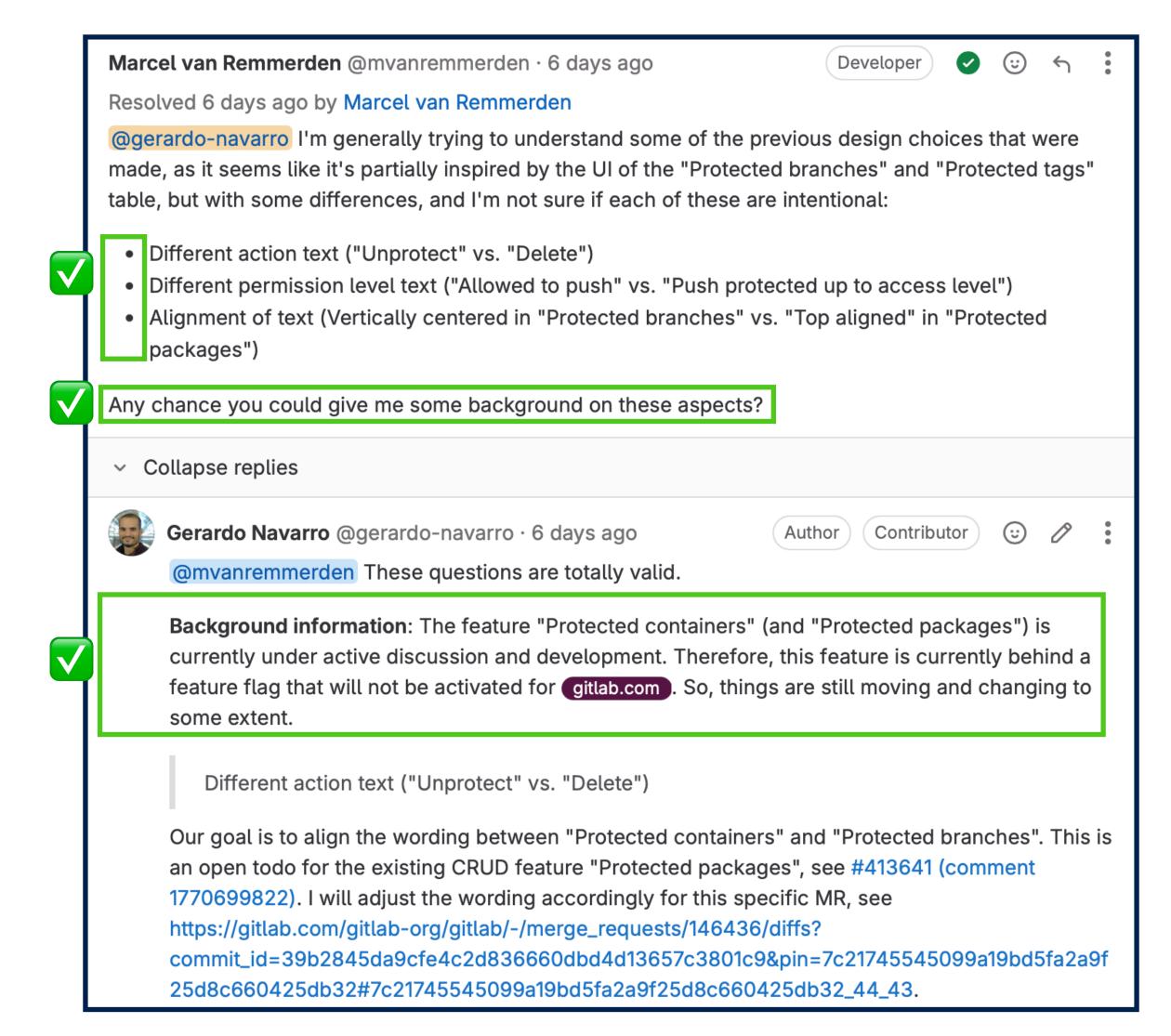
```
npm notice Publishing to http://gdk.test:3000/api/v4/projects/34/packages/npm/ with tag latest and default access
npm ERR! code E403
npm ERR! 403 403 Forbidden - PUT http://gdk.test:3000/api/v4/projects/34/packages/npm/@siemens%2fopen-source-at-siemens-prod-variant-1 - Package protected.
npm ERR! 403 In most cases, you or one of your dependencies are requesting
npm npm ERR! 403 a package version that is forbidden by your security policy, or
npm ERR! 403 on a server you do not have access to.
```



Hint #1: Communication

- Use screenshot, screencasts or other visuals
- Apply text formatting for visual structure
- Include "Call-To-Action", e.g. question

Add helpful (necessary) context or background information



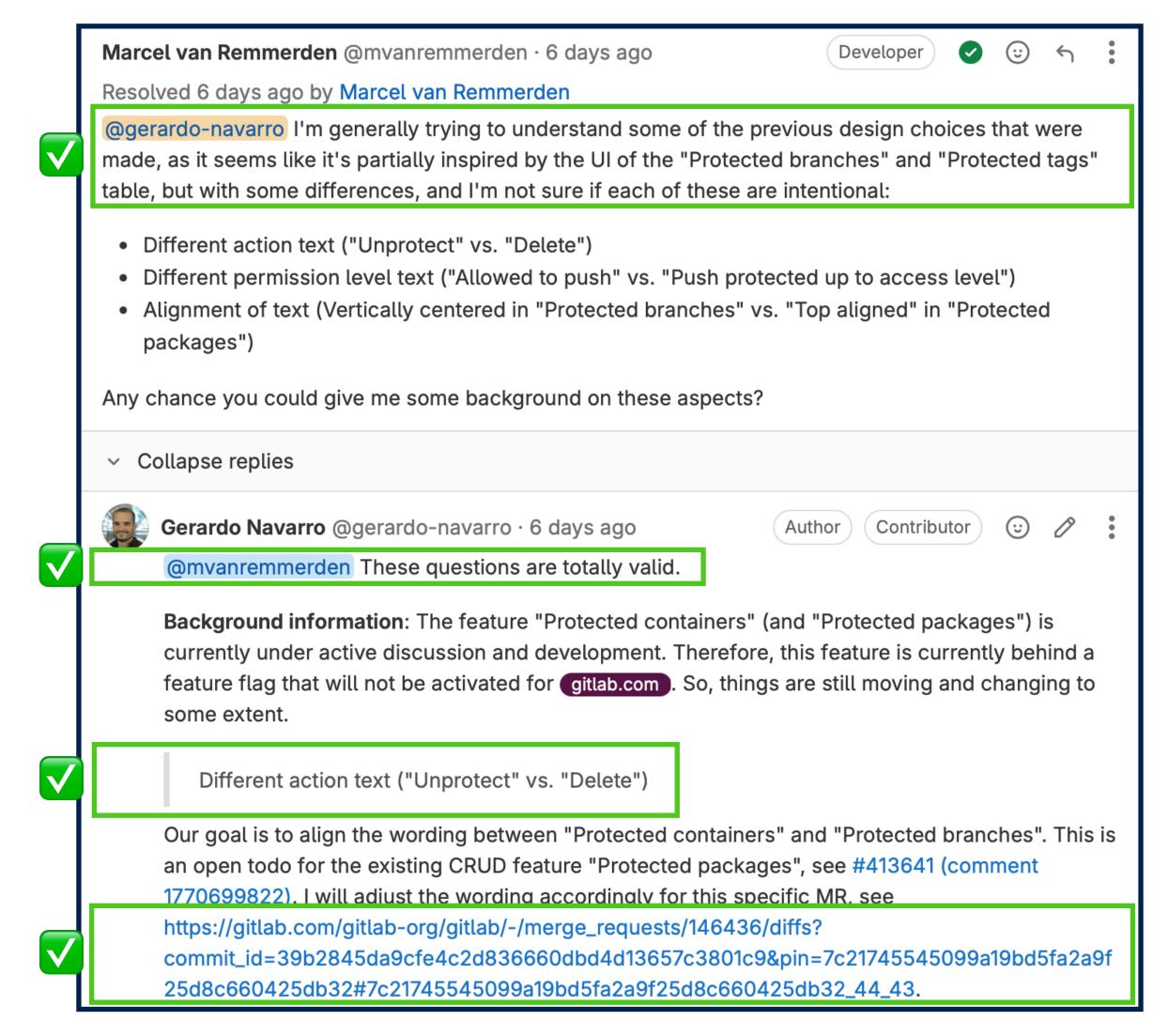


Hint #2: Communication

- Assume positive intent in every interaction
- Friendly, responsible and inclusive communication is key

 @Mention / Ping reviewers or collaborators directly to get their attention

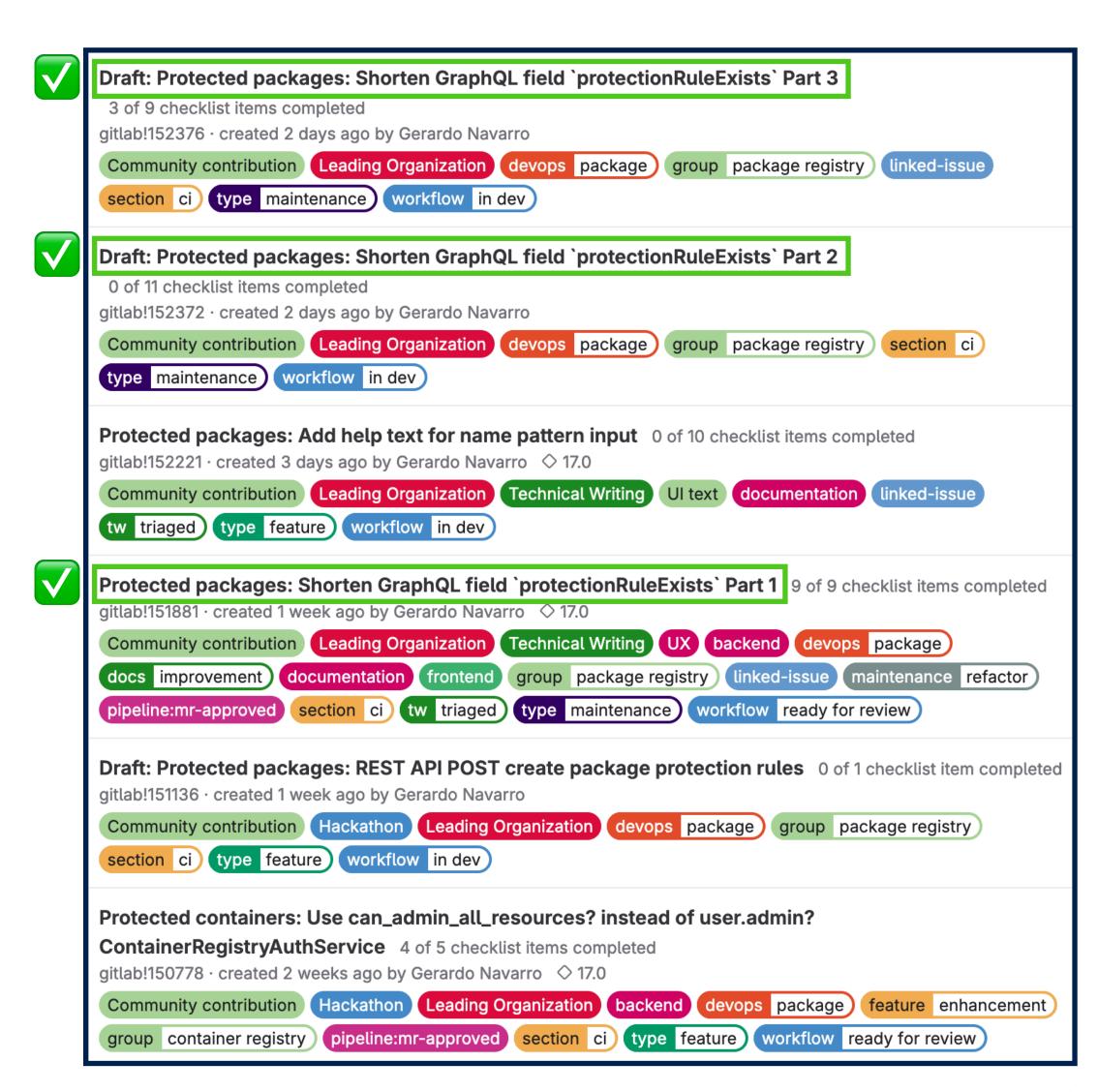
- Quote previous questions in your answer
- Use links to directly reference resources





Hint #3: Work in Parallel

- Work on multiple MR to avoid idle time
- "Stacking" MRs with separate scopes when building on larger features
- Work in different areas of the code base to engage with different GitLab teams
- Keep track of follow-up todos







Try out protected packages on code.siemens.com

Gerardo Navarro

Full-Stack / DevOps Software Engineer

gerardo@b310.de

https://gitlab.com/gerardo-navarro

