



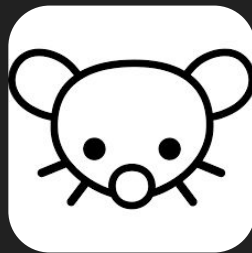
Works on My Machine

Team 3

<https://www.youtube.com/watch?v=UkW-vXXCAt4>

Overview

- What?
 - Application that *Blends* posts from different Mastodon and Lemmy instances into a single webpage.
- Why?
 - Easy place to find some trending posts, for people who need a place to start their Fediverse journey.

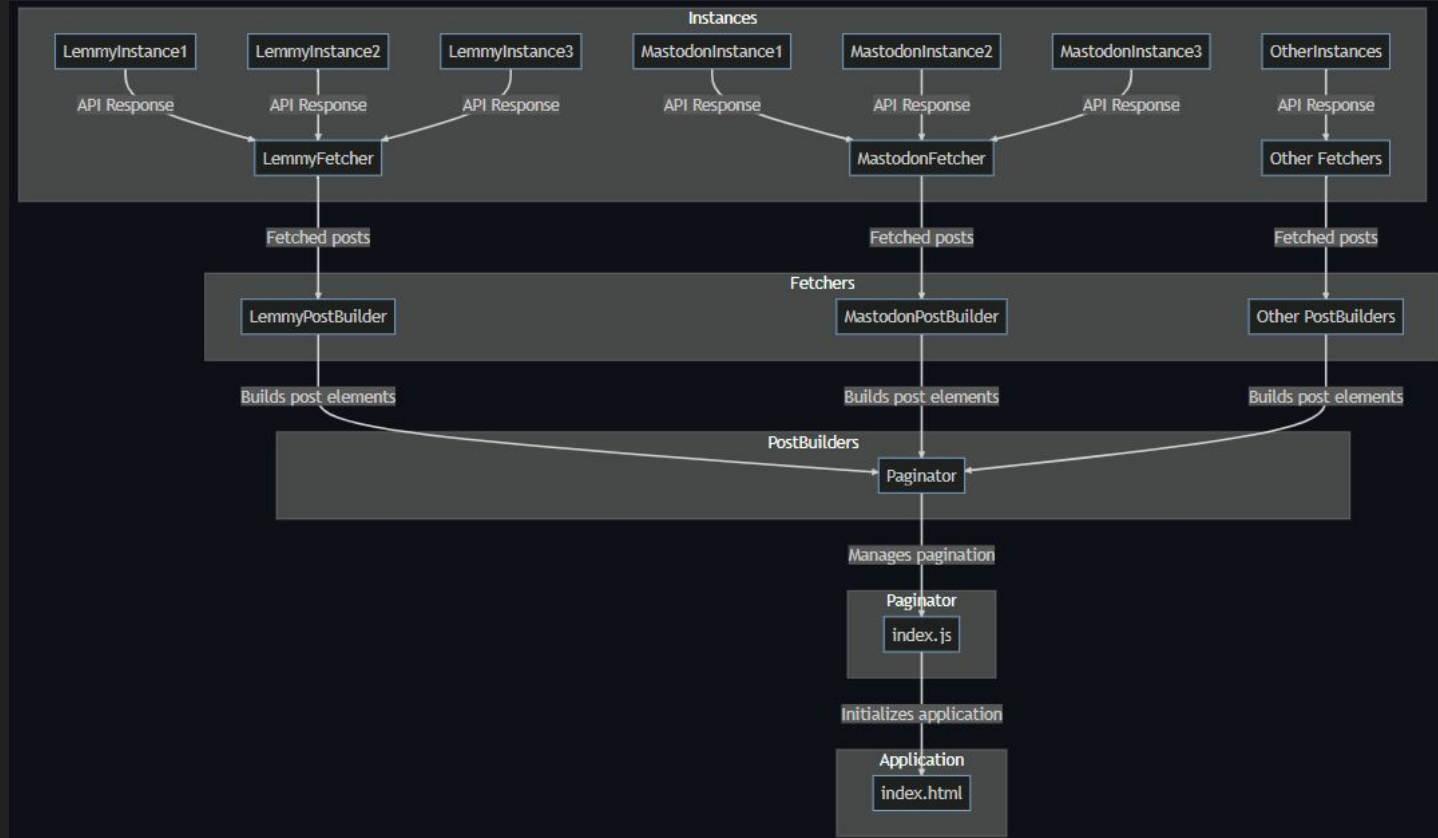


Processes adopted

- Agile - Sprints lasting ~1 week
- Weekly meetings
 - Policy was to make it to at least $\frac{2}{3}$ meetings a week
- Feature branching*
- Architectural Decision Record
- Meeting notes
- “Well” documented code
- Retrospectives



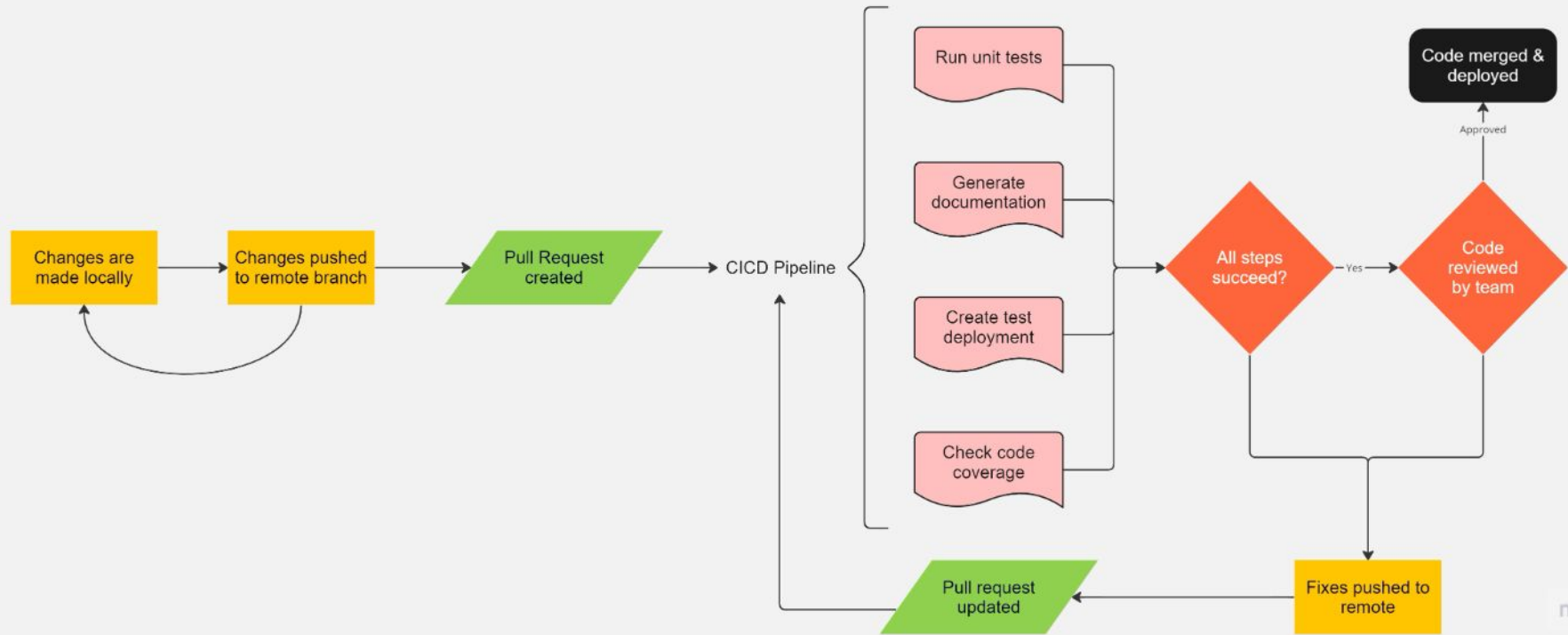
Architecture



Application workflow

- Fetchers get the posts from each of the instances through API calls, depending on the type of the network.
- The fetched posts from each of the instances is then parsed and a custom HTML element “fedi-post” is constructed for each post.
- These custom elements are then passed to the Paginator function, which divides the posts into separate pages to avoid doom scrolling.

CI Pipeline





Lessons Learned

What went well 😊

- We have something that works
- We scoped the project well from the beginning
- A lot of things got done each meeting and they were consistent
- We did a lot of documentation things right from the start (e.g. issues and meeting minutes)
- Our pipeline was set up early and works well

Lessons Learned

What didn't go well 🙄

- Failed to implement daily standups
- Schedule conflicts with other projects during weeks 10-11 due to finals
- Had to settle for feature branching instead of CI
- We extended work beyond our planned internal deadline by a fair bit, leading to stress
- Had to spend extra time and have a lot of meetings towards the end, sort of a final push

Lessons Learned

What we would do differently in the future 🧐

- Decide subteams a lot sooner, more time to get used to the routine and process
- Have a consistent meeting time a lot sooner
- Remember to make ADRs from the start

For Future Devs

- CACHING. Caching is the key
 - Drastically reduce latency
 - Didn't have time to cover here
- More comprehensive tests
 - 95% code coverage, some areas not well suited to unit tests; we would say that integration testing and end-to-end testing would also be good to look into

Thank you!

Team Works on my Machine

