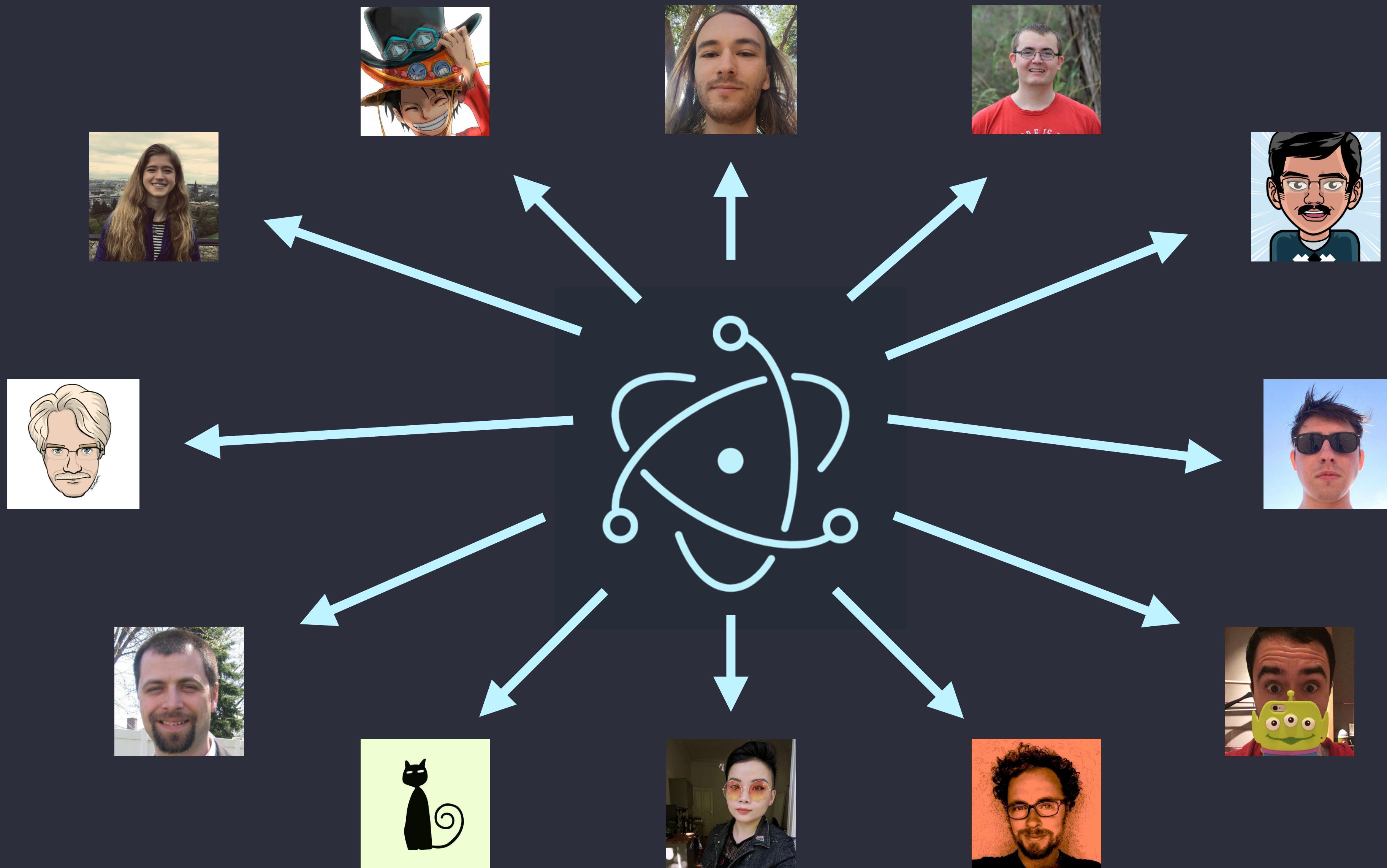




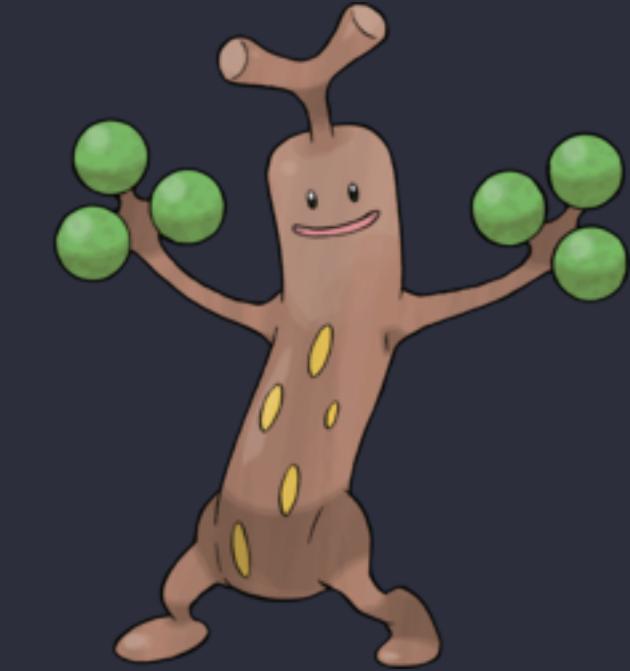
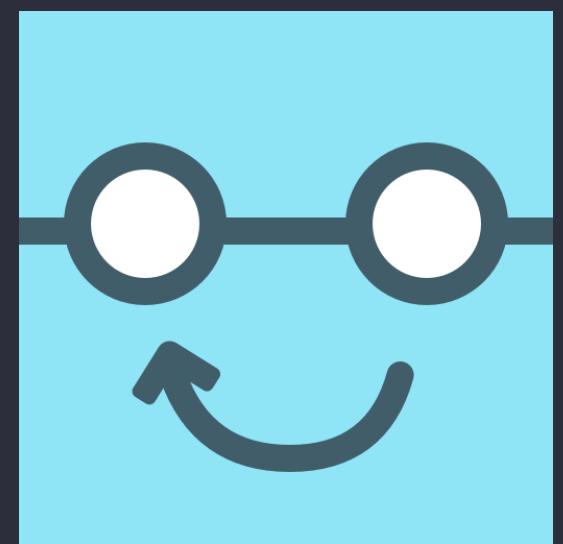
ELECTRON: MEET THE TEAM

Shelley Vohr
@codebytere

Who's On The Team?



Who's On The Team?



Quick Examples



Semantic Pull Requests

Requires that PRs/commits have their purpose as a prefix



Work-In-Progress

Prevents merging of pull requests that have WIP in the title



Release Notes Clerk

Requires that Pull Requests have release notes

How Do Bots Help Us?

Improve developer workflow

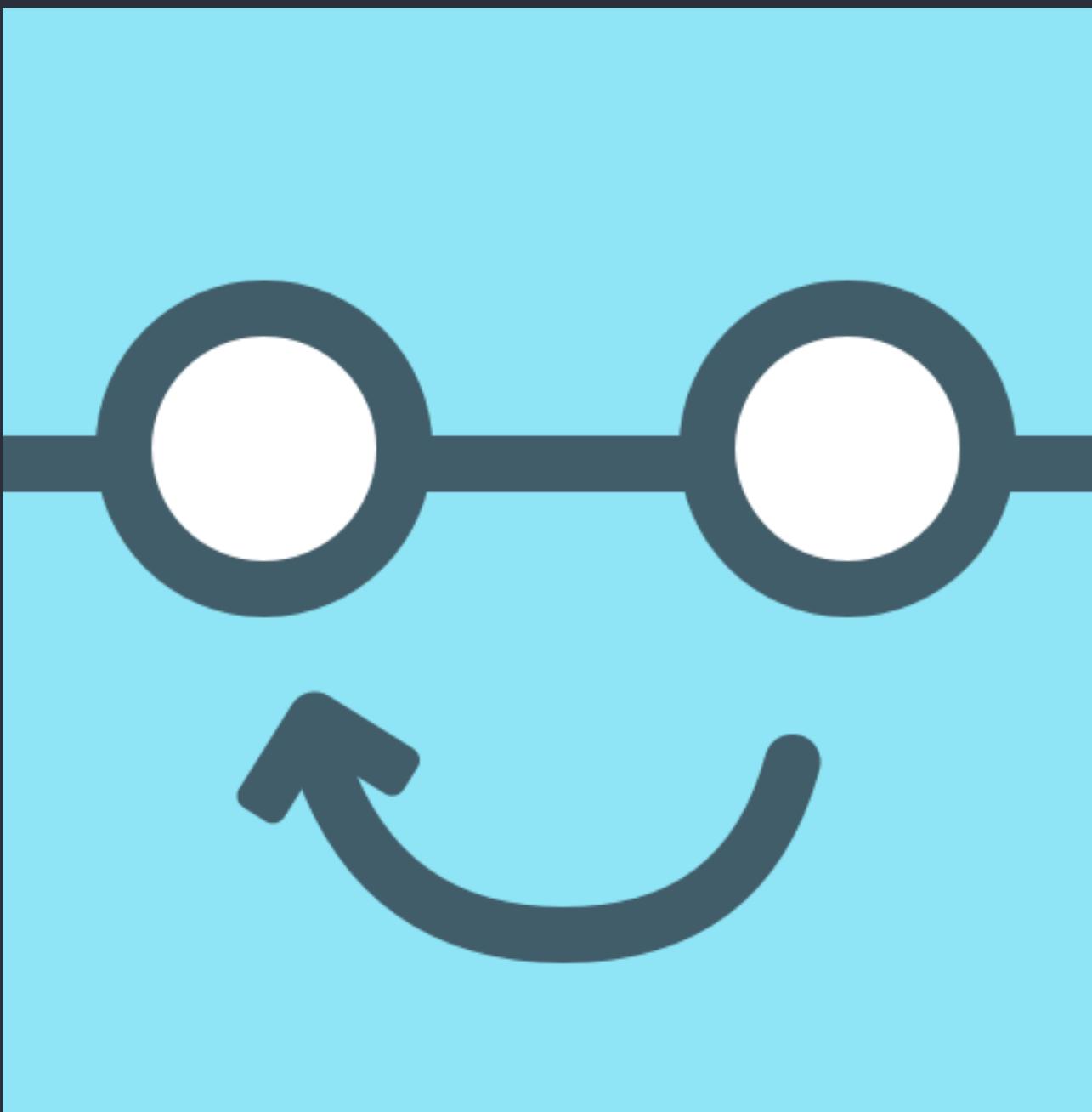
Enforce standard requirements

Free developers' time

Remove ambiguity of responsibility



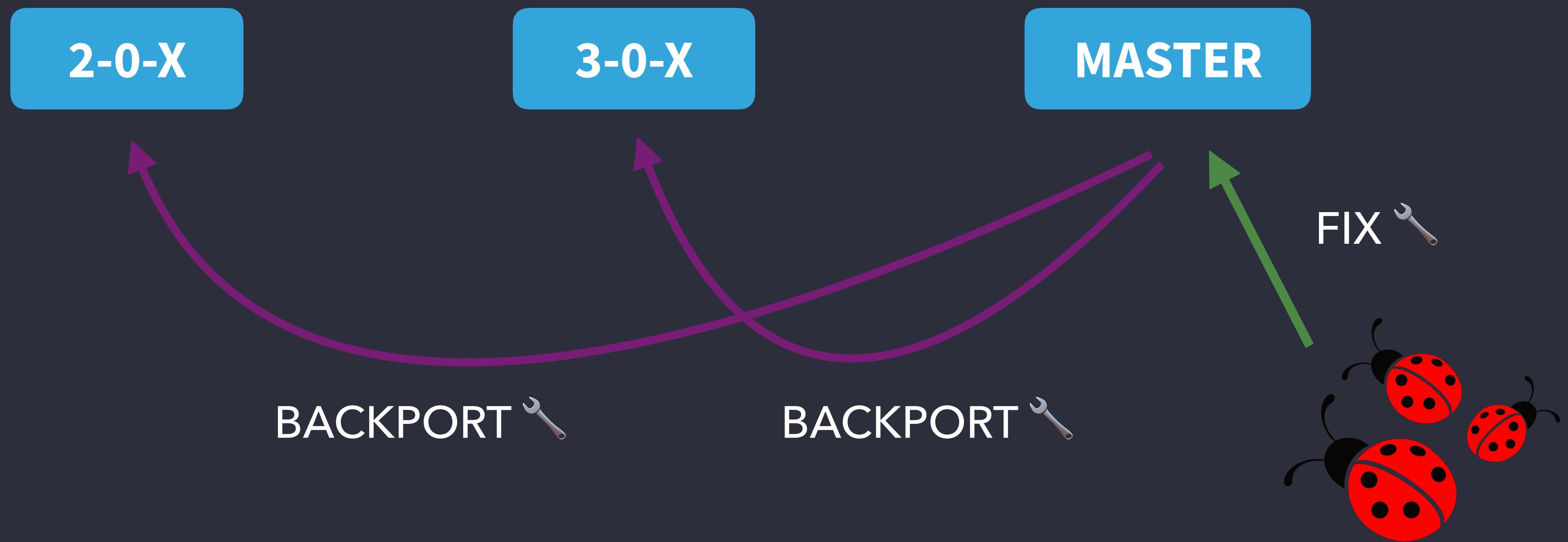
Case Study #1: Trop



<https://github.com/electron/trop>

```
[ ... 'backport'].slice(4)].reverse().join``
```

What's Backporting?



What's the Problem?

Backporting is:

- REPETITIVE
- TIME-CONSUMING
- EASY TO FORGET ABOUT

Enter Trop

Trop does this for us!

Backport: A commit or series of commits applied to a branch

Flow

- 1) Listen for merge event OR manual trigger
- 2) Cherry pick commits to a copy branch
- 3) Open a PR against the release branch

Real-World Example

The screenshot shows a GitHub pull request timeline for a merge into the master branch. The timeline includes the following events:

- MarshallOfSound merged commit `26925e6` into `master` 8 hours ago. 3 of 22 checks passed. [View details](#) [Revert](#)
- MarshallOfSound deleted the `autorelease-allthethings` branch 8 hours ago. [Restore branch](#)
- trop bot referenced this pull request 8 hours ago. **chore: remove autorelease check logic (backport: 2-0-x) #14221** [Merged](#)
- trop bot commented 8 hours ago: We have automatically backported this PR to "2-0-x", please check out [#14221](#)
- trop bot added `merged/2-0-x` and removed `target/2-0-x` labels 8 hours ago
- trop bot referenced this pull request 8 hours ago. **chore: remove autorelease check logic (backport: 3-0-x) #14222** [Merged](#)
- trop bot commented 8 hours ago: We have automatically backported this PR to "3-0-x", please check out [#14222](#)
- trop bot added `merged/3-0-x` and removed `target/3-0-x` labels 8 hours ago

Case Study #2: Roller

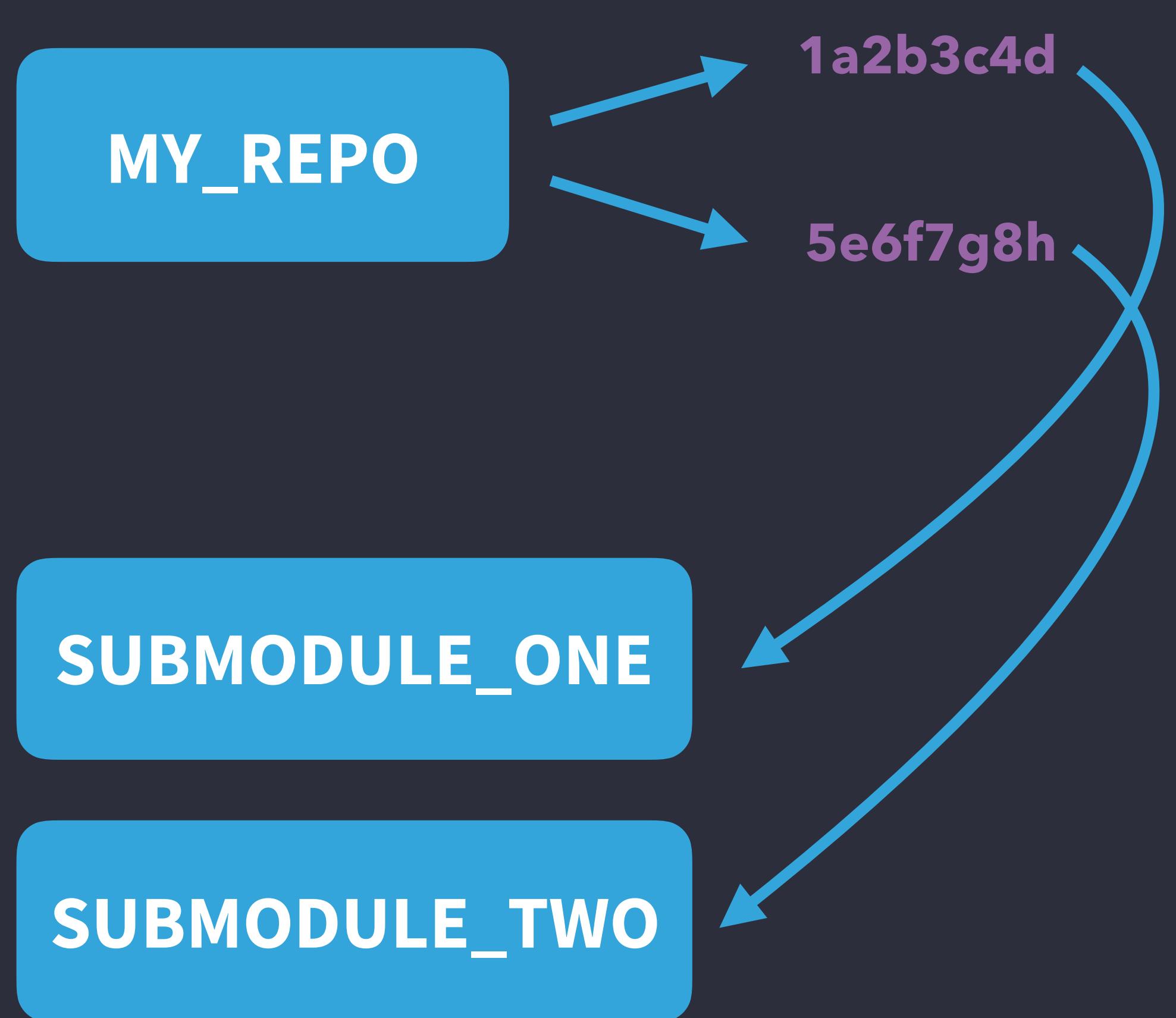


<https://github.com/electron/roller>

rollin' on upstream

What's the Problem?

Submodule Vendoring



- 1) When we make changes to a submodule, we need to update Electron
- 2) The update process is repetitive and time-consuming over a long period
 - Need to update the reference hash in each branch

Enter Roller

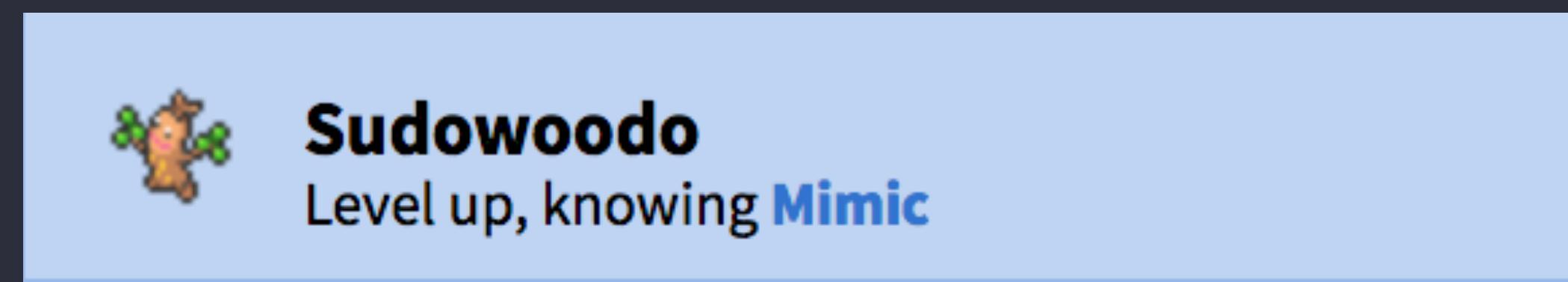
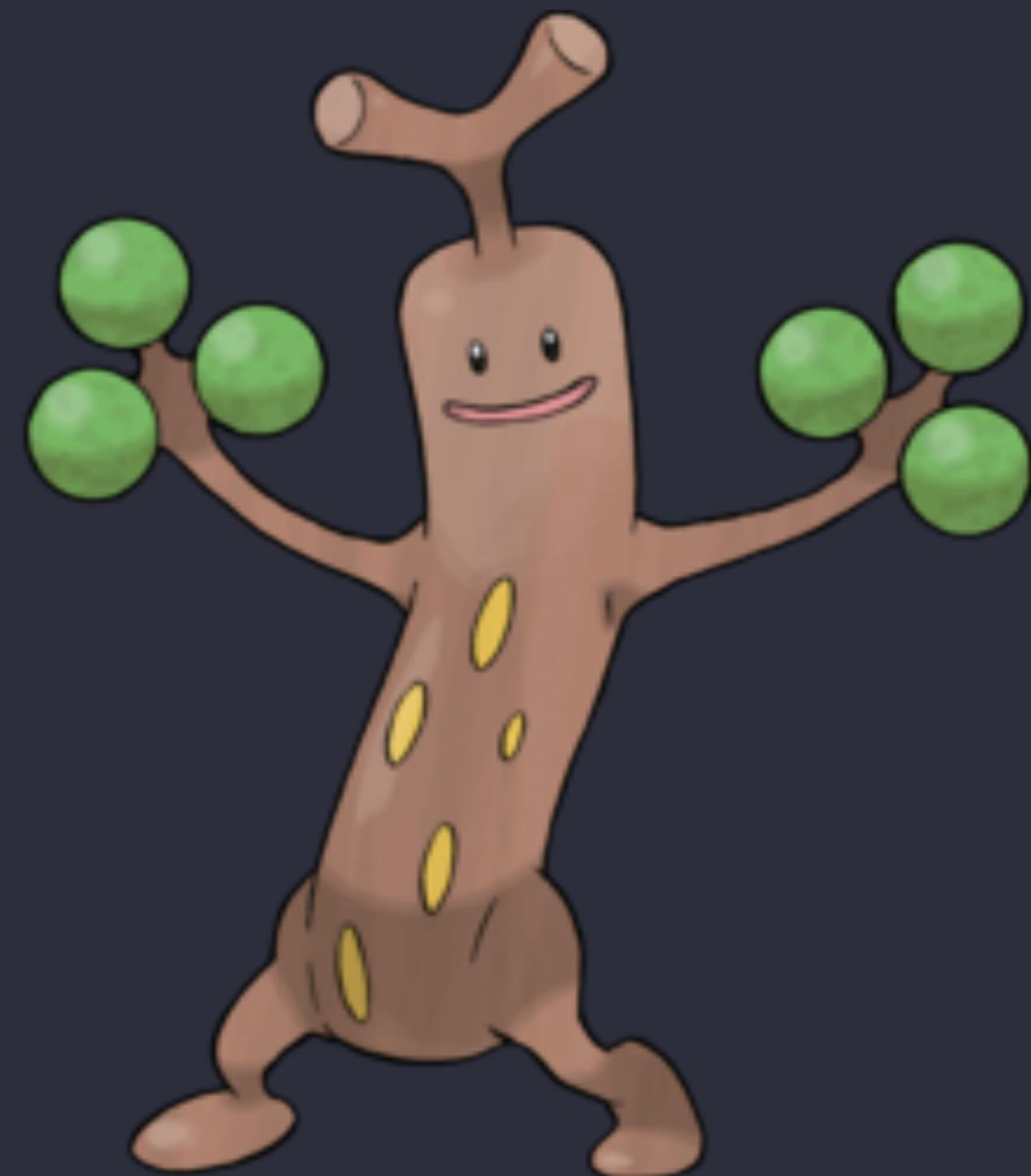
- A PR lands into release branches of libchromiumcontent
- Roller automatically opens up a PR into electron/electron
- Monitors and accounts for superseding changes



A screenshot of a GitHub pull request interface. At the top, a comment from 'roller-bot' is shown, stating: 'Updating libcc reference to latest. Changes since the last roll:' followed by two commits. Below this, a commit history shows three commits added by 'roller-bot':

- chore: bump libcc submodule to 00cce7a094c5ded91d52d2ac2aac2ab70dc66d02 (36d6209)
- chore: bump libcc in DEPS to 00cce7a094c5ded91d52d2ac2aac2ab70dc66d02 (170b10c)
- requested a review from **electron/reviewers** as a code owner a day ago

Case Study #3: Sudowoodo

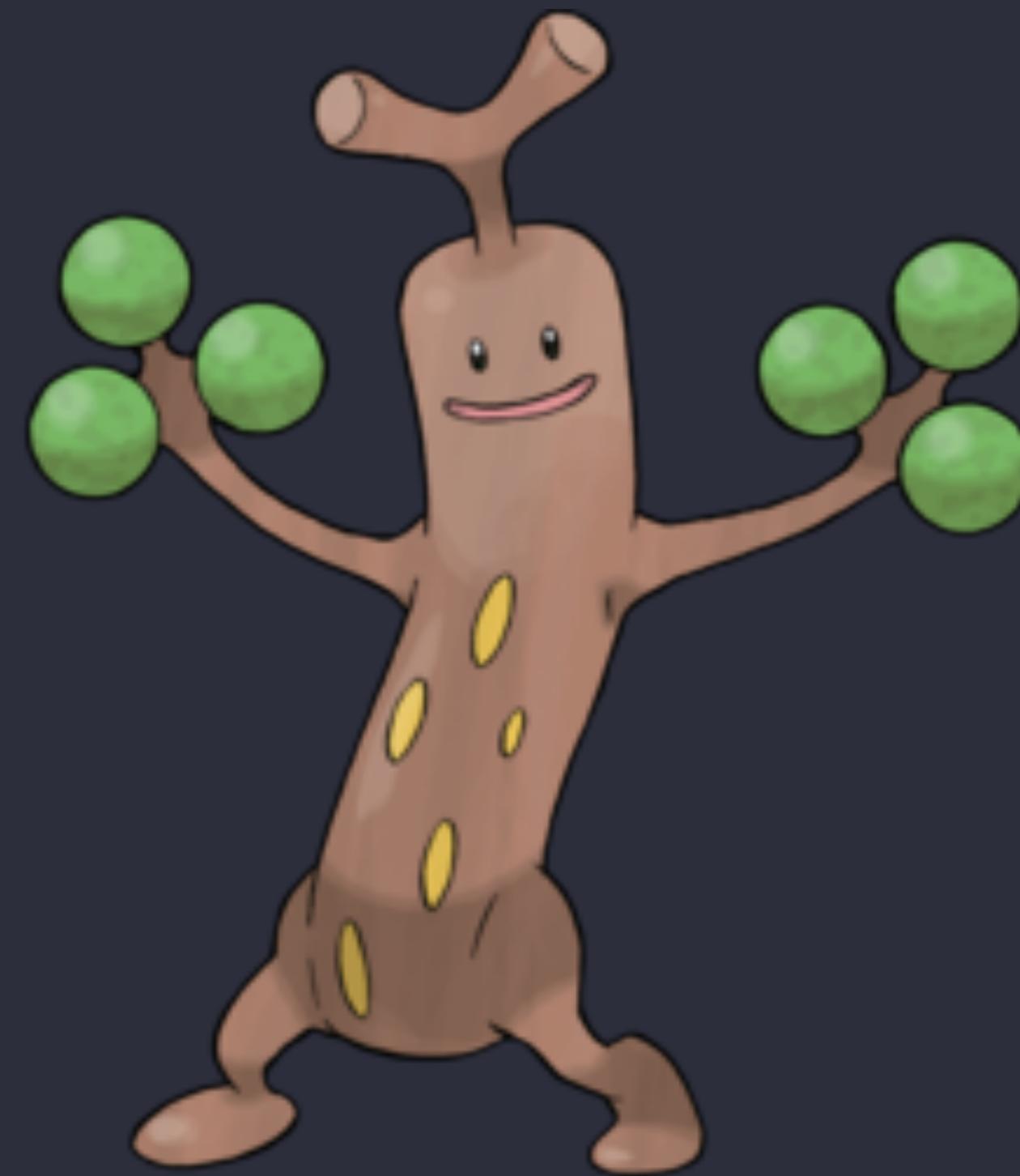


End-to-End Release Pipeline Automation

What's the Problem?

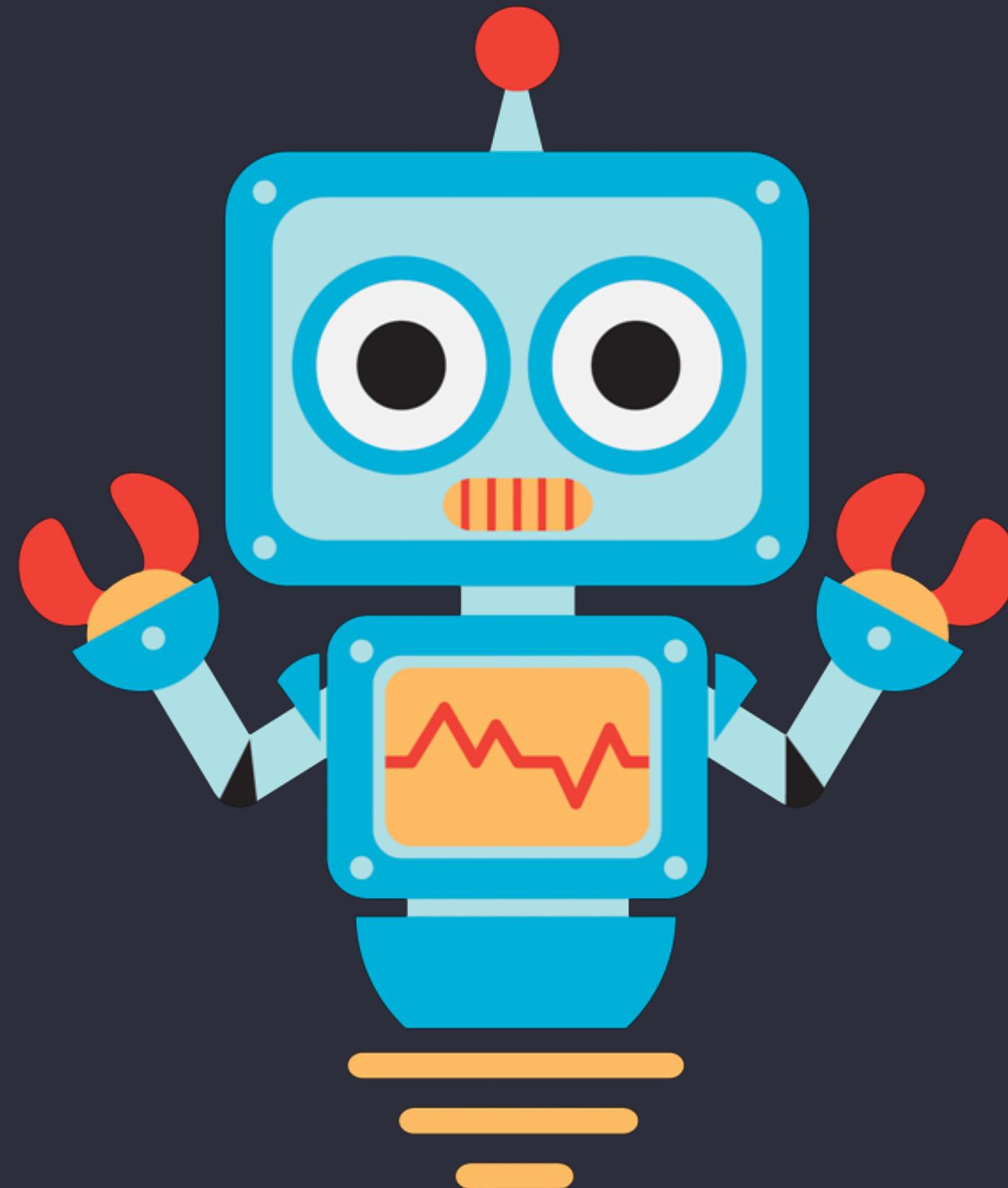
- 1) We want nightly builds for each release line + master
- 2) We want to be able to run more than one release at a time
- 3) We want to know what releases are happening asynchronously
- 4) Remove the burden of manual release work from engineers

Enter Sudowoodo



1. Clone `electron/electron`
2. Checkout the branch
3. Install dependencies using `npm i`
4. Write an `.env` file using the whole current environment
5. Run the "prepare release" script, kicking off builds on CI
6. Ask for an npm OTP from ElectronHQ on Slack
7. Run `npm publish`
8. Cleanup

Want to Make Your Own?



PROBOT

GitHub Apps to automate
and improve your workflow

Use pre-built apps to extend GitHub,
and easily build and share your own.

THANK YOU!

