

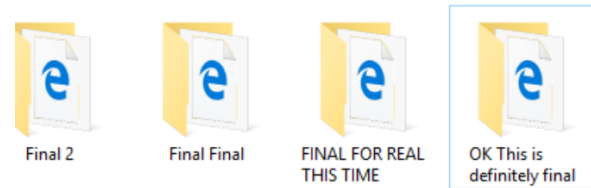
Getting Started at Hackathons

Track 1: Getting Started

Hi, I'm Eric Jiang 🖐️

- Currently, the Project Lead for monPlan -
monplan.apps.monash.edu
- Co-founded FutureYou, GeckoDM and MARIE.js
- 🐦 @lorderikir
- 🔗 <https://lorderikir.me>
- @ eric.jiang@monash.edu
- github.com/lorderikir

First of all, what is git?

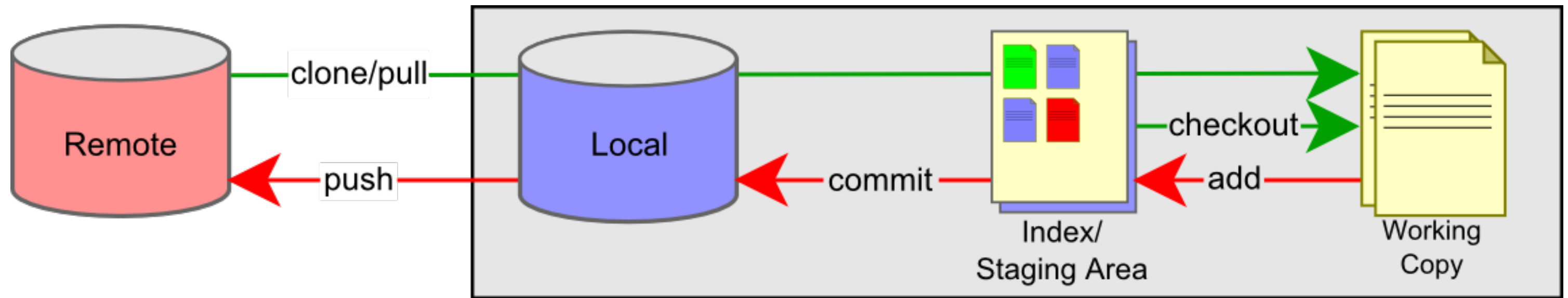


Git

Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people

— Git SCM Website

How Git Works



Some Basic Commands

Command	Description
<code>git clone</code>	Clones a repository locally
<code>git add</code>	Stages changes to file(s) for a commit
<code>git commit</code>	Creates a commit (set of changes)
<code>git push</code>	Push changes to the hosted repo

Advanced Git

Well, working with teams  may be hard. There are generally two ways you can work off a repository.

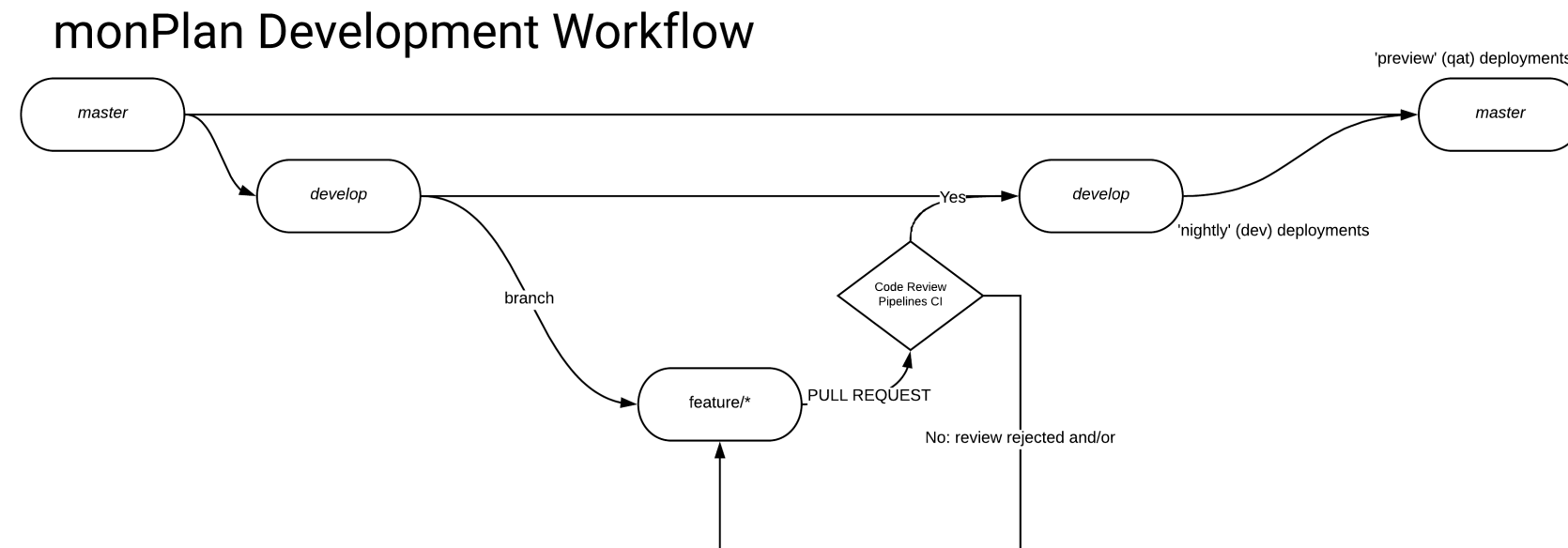
- Using Branches
- Using Forks

Option 1: Use Branches 🌳 for Versioning Control

1. Make a branch with the feature name or your own username
2. Every time you commit and push up
3. Make a Pull Request
4. Merge the pull request

One of the best workflows is known as *GitFlow*

GitFlow - Used with monPlan Git Workflow



- **master**: branch is the key branch, everytime for release
- **develop**: *unstable*, most of the PRs should go here
- **'feature/*', 'fix/*', etc.**: are 'for purpose' branches, these branches are for development
- **deploy** (not shown), is for **manual** deployments to prod

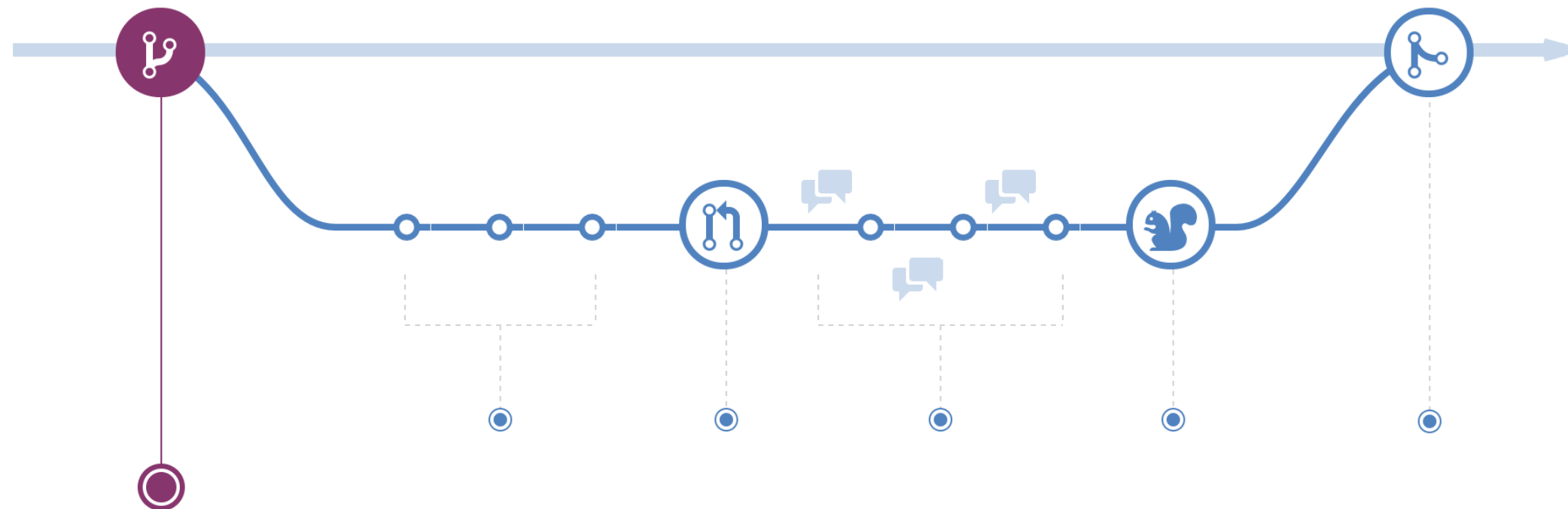
This slide has been adapted from my CI-CD talk

So how does it work?

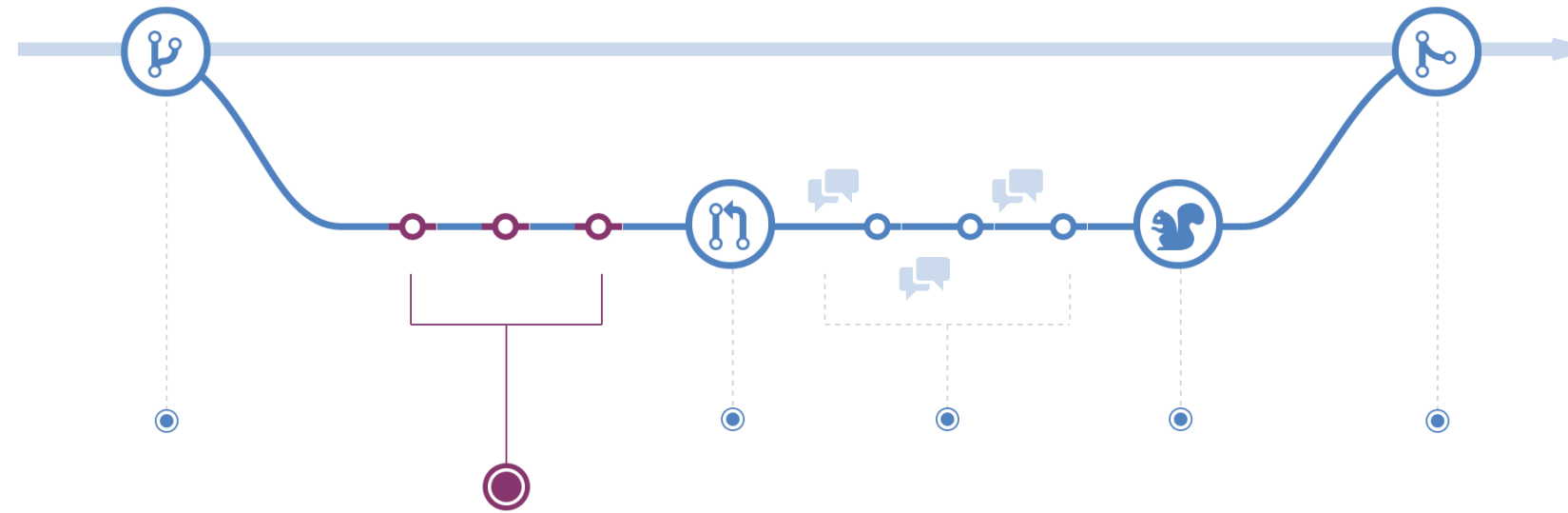
Create a branch

ProTip: It's easier for your branch name to follow a simple pattern, typically we use something like *fix/*, *feature/*, *refactor/**

```
# we create a new branch  
git branch feature/demo-branch  
# we make the new branch the new working branch  
git checkout feature/demo-branch
```



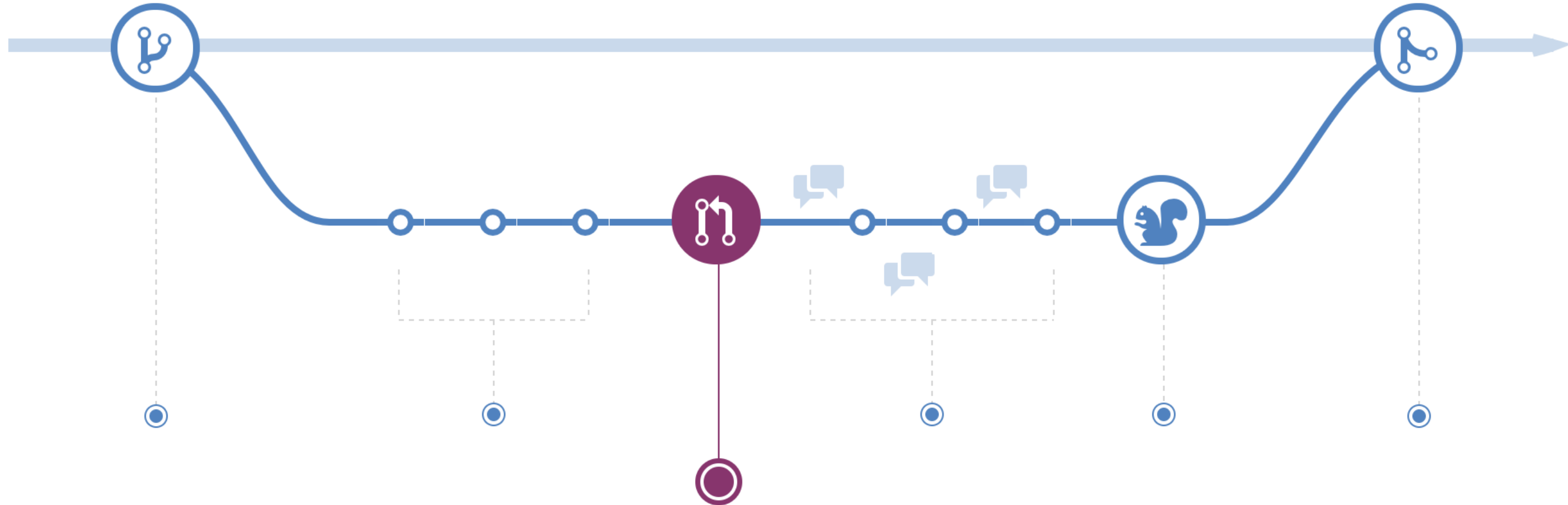
Add Commits



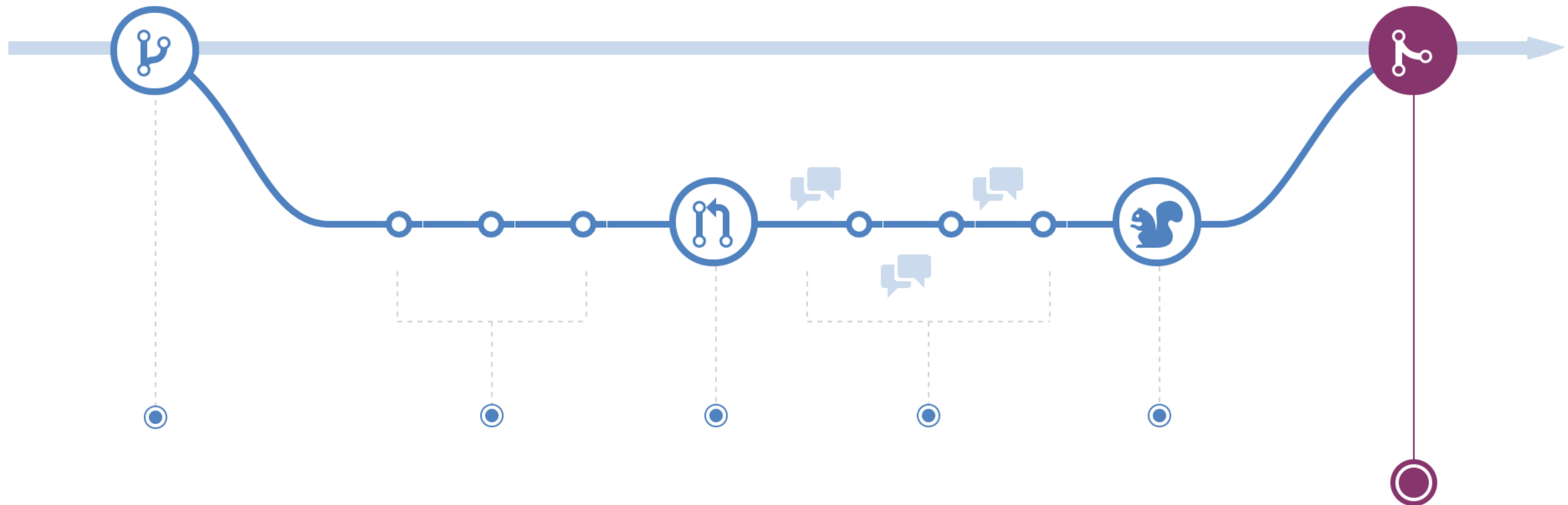
We make our changes and then

```
git add .  
git commit -m "new commit"  
git push
```

Make a PR

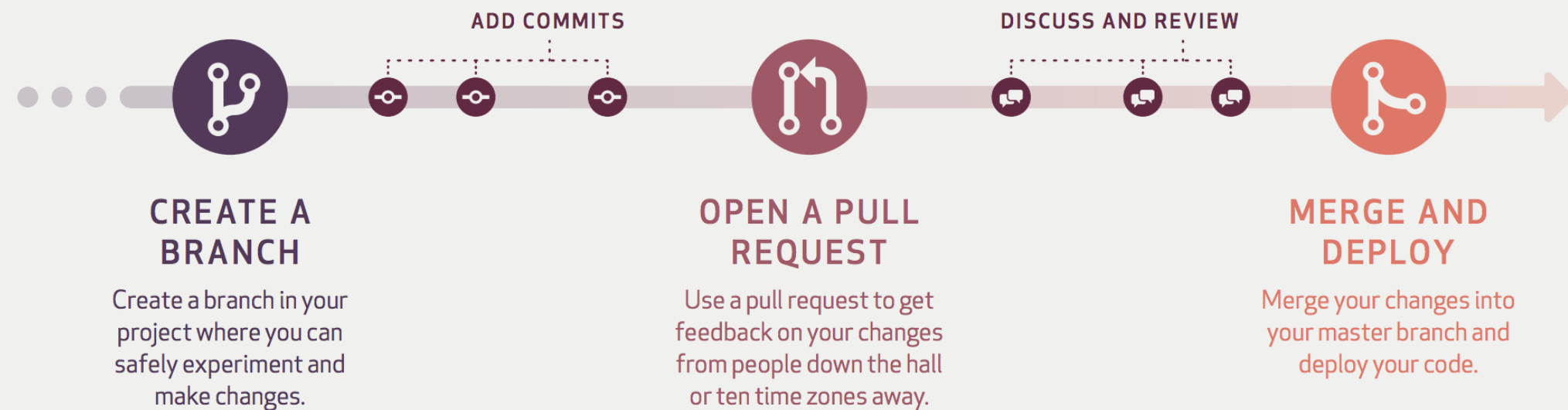


Discuss and merge



WORK FAST WORK SMART THE GITHUB FLOW

The GitHub Flow is a lightweight, branch-based workflow that's great for teams and projects with regular deployments. Find this and other guides at <http://guides.github.com/>.



GitHub is the best way to build software together.

GitHub provides tools for easier collaboration and code sharing from any device. Start collaborating with millions of developers today!

Option 2: Using Forks 🍴 for Versioning Control

The best way to image a fork, is image a copy of the main repository that you own that you can *pull, merge* and apply changes to.

Questions?



Goodbye 🖐️

Track 2: Firebase + ReactJS
for Hacks coming soon