GitHub workshop

Bastien Brugger

Ishan Mishra

Ryan Petersburg

Cornell University

ERES V

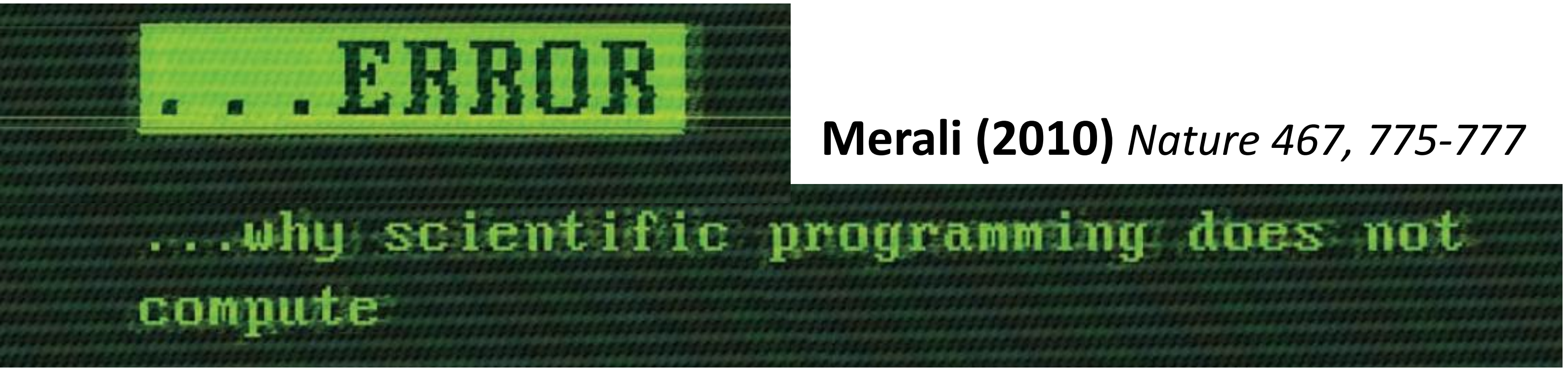
Day 3 – 2019/06/19

...ERROR

....why scientific programming does not compute

Merali (2010) *Nature 467, 775-777*

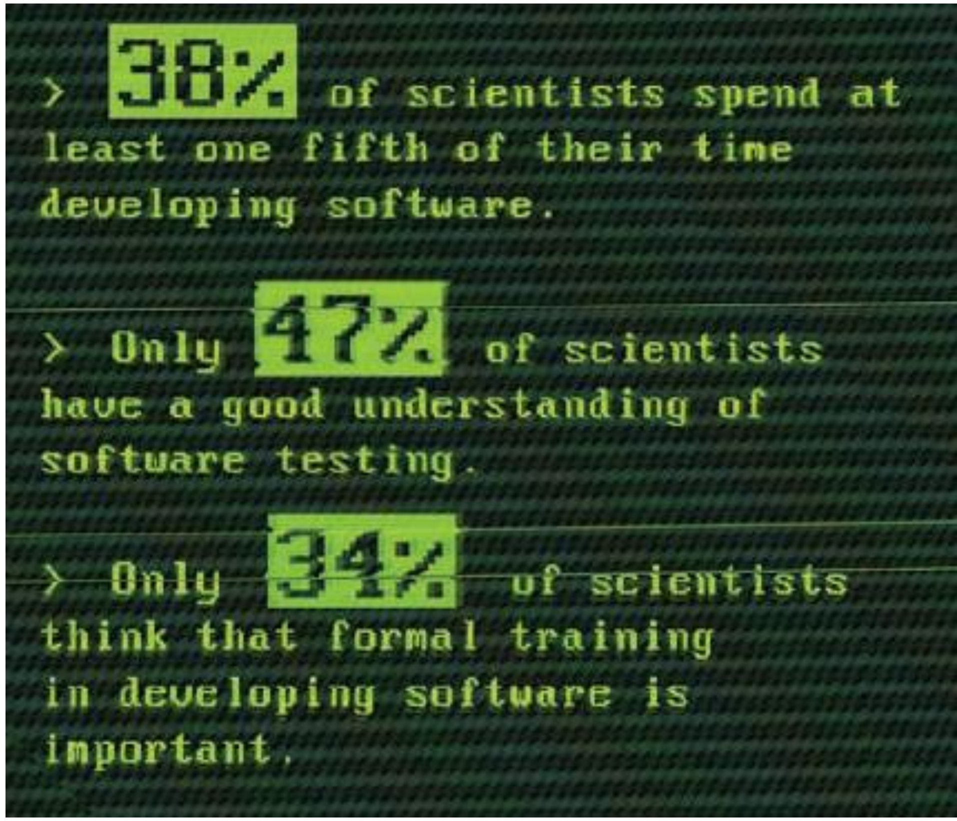
*"Problems created by bad documentation are further amplified when successful codes are modified by others to fit new purposes. The result is the bane of many graduate students or postdocs' life: the "**monster code**". Sometimes decades old, these codes are notoriously messy and become progressively more nightmarish to handle"*



...ERROR

...why scientific programming does not compute

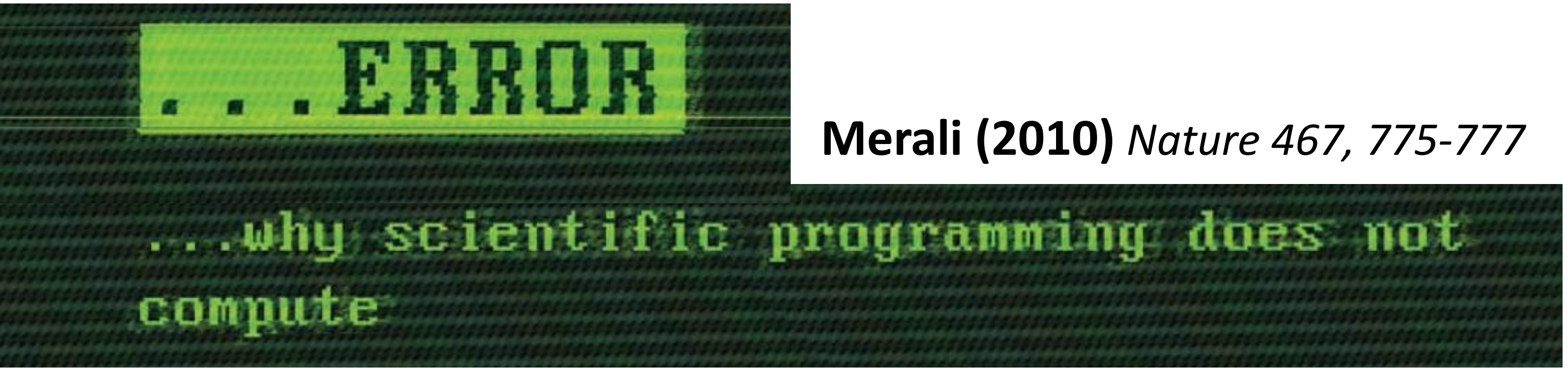
Merali (2010) *Nature 467, 775-777*



> **38%** of scientists spend at least one fifth of their time developing software.

> Only **47%** of scientists have a good understanding of software testing.

> Only **34%** of scientists think that formal training in developing software is important.

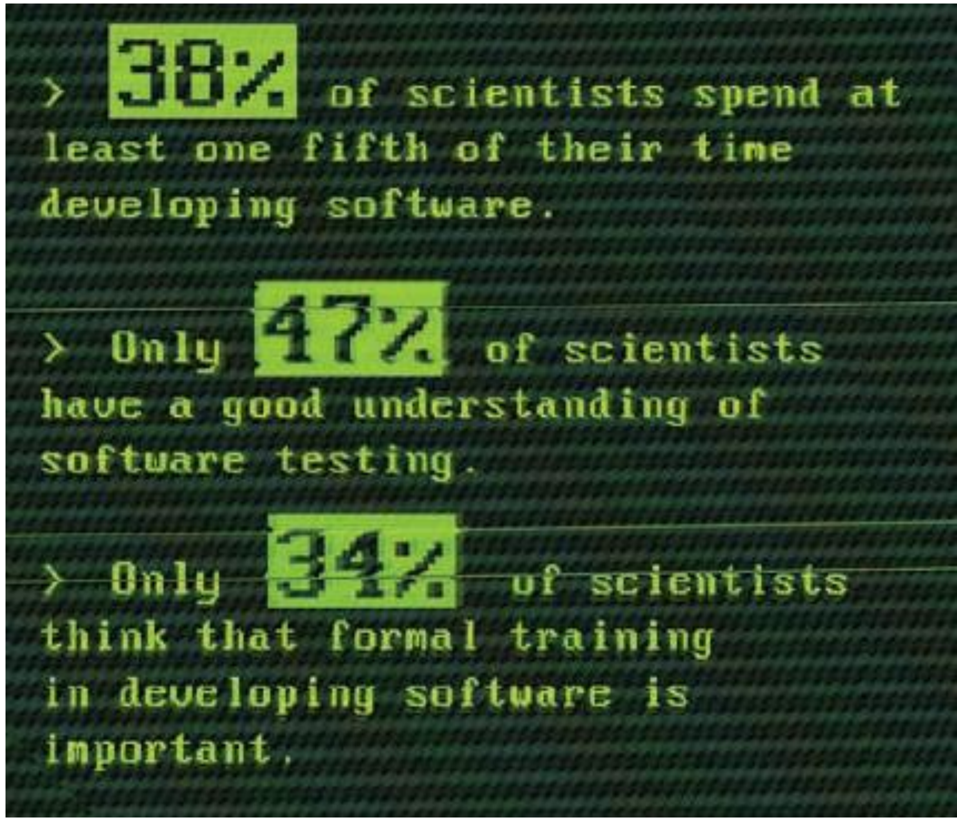

...ERROR

....why scientific programming does not compute

Merali (2010) *Nature 467, 775-777*

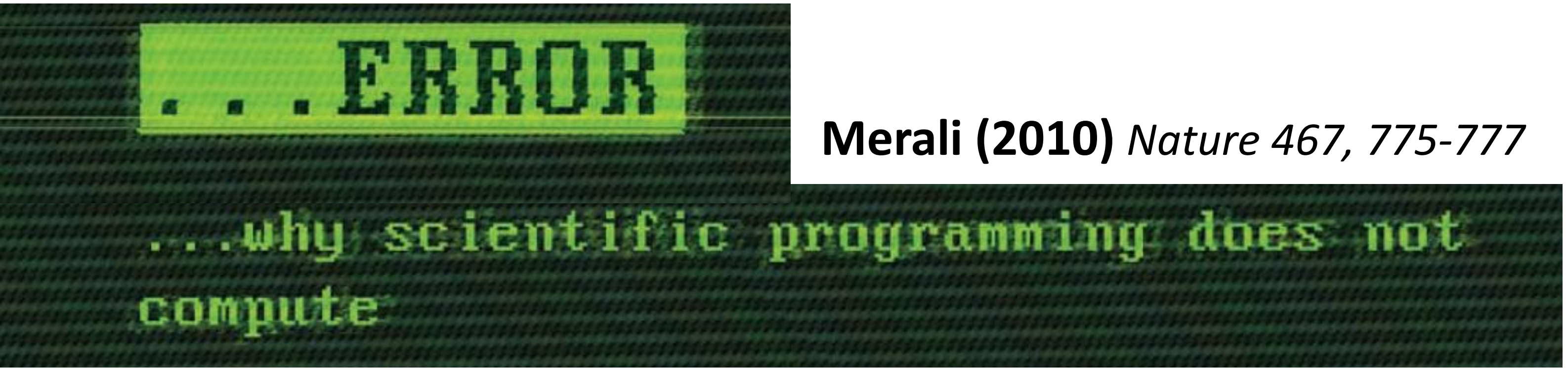
Five tips to make scientific code more robust:

- *Track your material*
- *Write testable software*
- *Test the software*
- *Encourage sharing of software*
- *Use a version-control system* (VCS)


> 38% of scientists spend at least one fifth of their time developing software.

> Only 47% of scientists have a good understanding of software testing.

> Only 34% of scientists think that formal training in developing software is important.



...ERROR

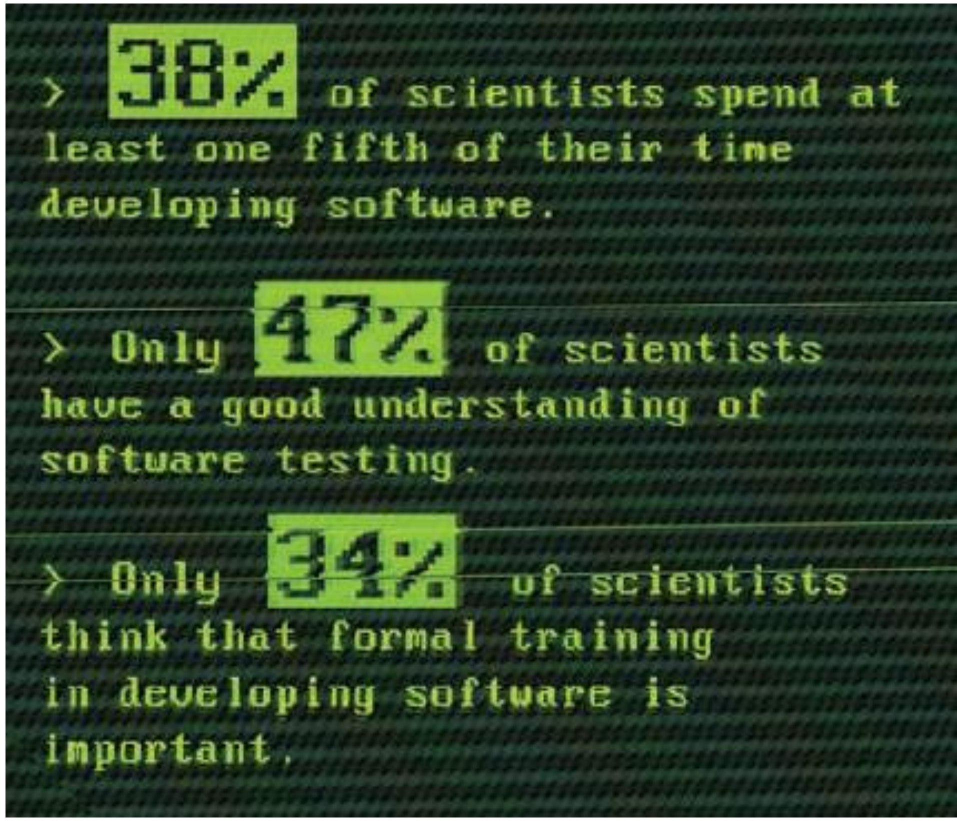
....why scientific programming does not compute

Merali (2010) *Nature 467, 775-777*

Five tips to make scientific code more robust:

- *Track your material*
- *Write testable software*
- *Test the software*
- *Encourage sharing of software*
- *Use a version-control system* (VCS)

GitHub



> 38% of scientists spend at least one fifth of their time developing software.

> Only 47% of scientists have a good understanding of software testing.

> Only 34% of scientists think that formal training in developing software is important.

Summary

1. What is **VCS** and why use it?
2. Creating and understanding your first **GitHub** repository
3. Managing an actual **test** case
4. Best practices for code **accessibility** and **documentation**
5. Creating your **website** with GitHub Pages
6. Other **resources**

Summary

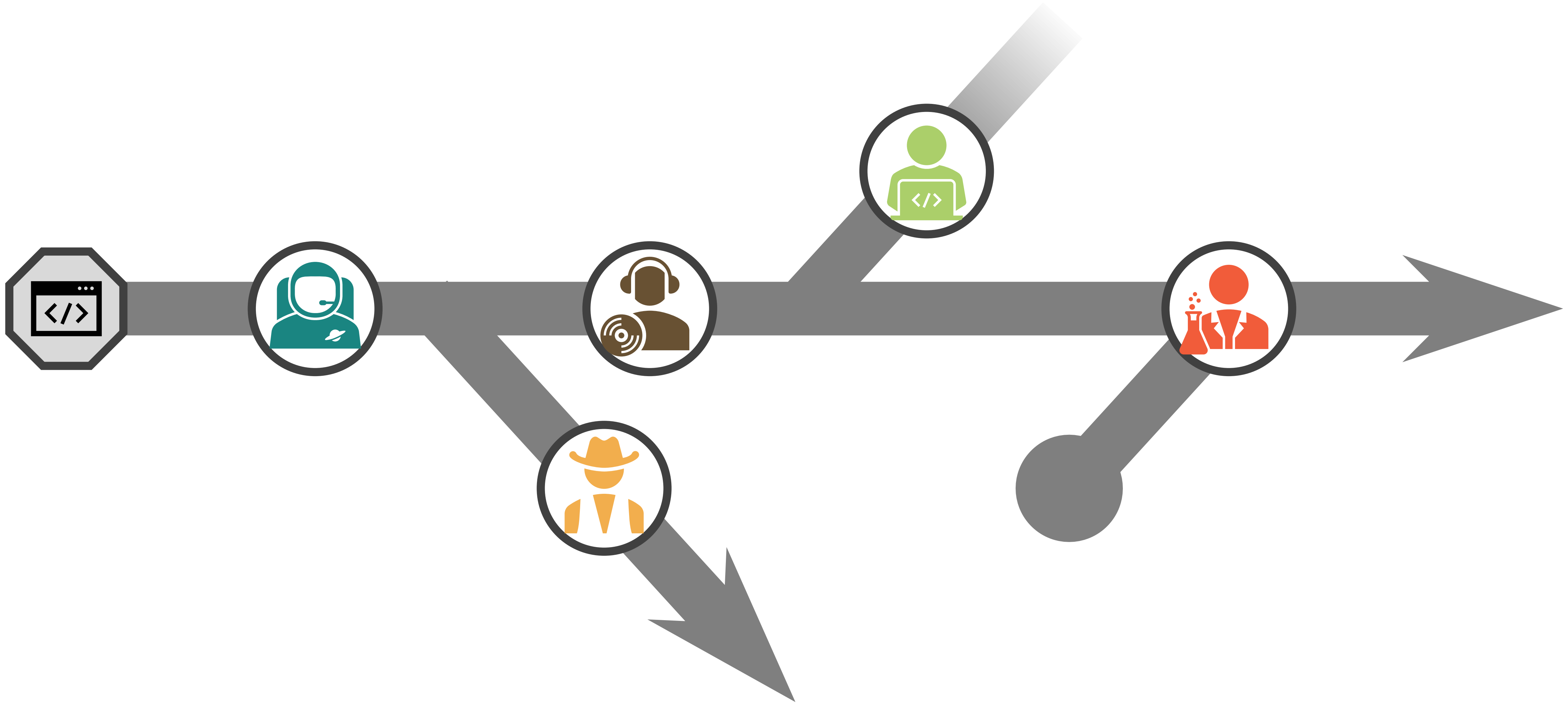
All material can be found at

`github.com/BastienBrugger/ERESV-github`

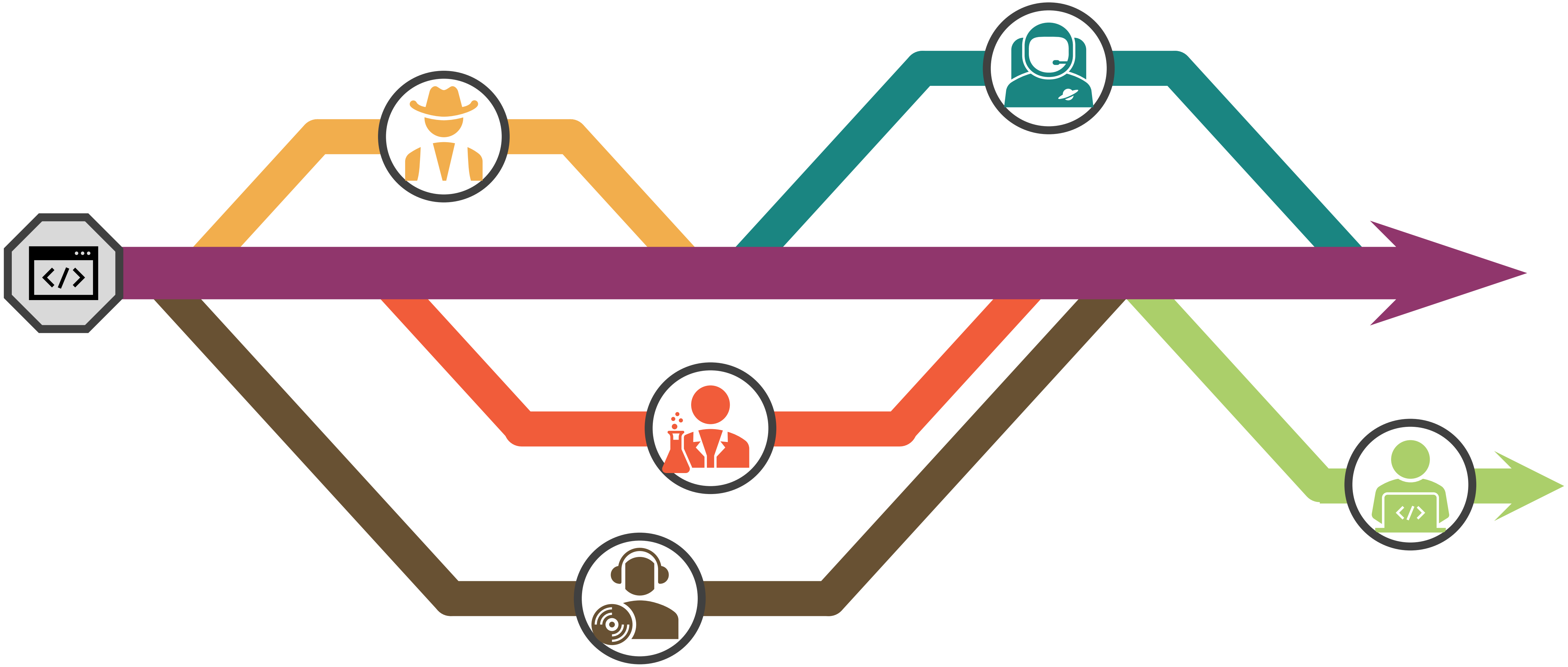
Disclaimer:

`Git`	*open-source VCS, the **tool** that manages your code history*
`GitHub`	*hosting **service** for `Git` repositories*

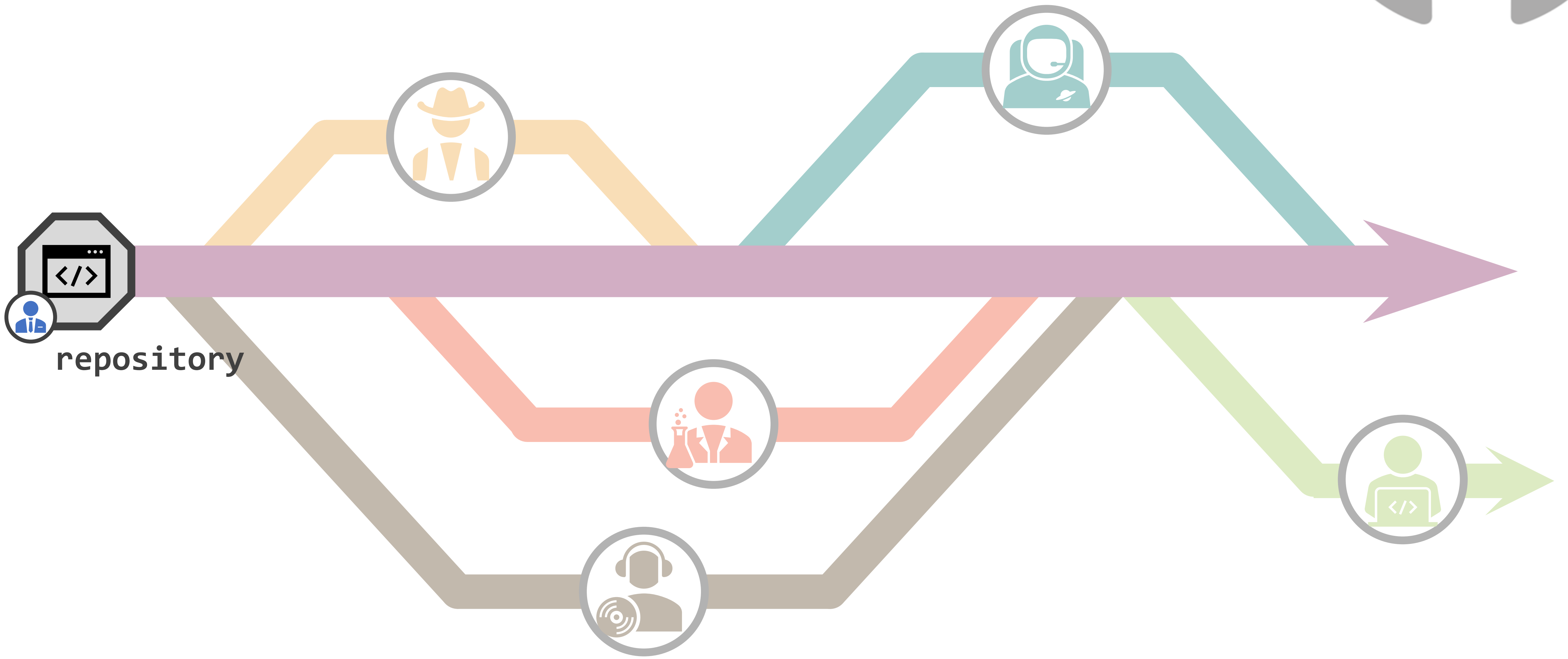
How VCS works



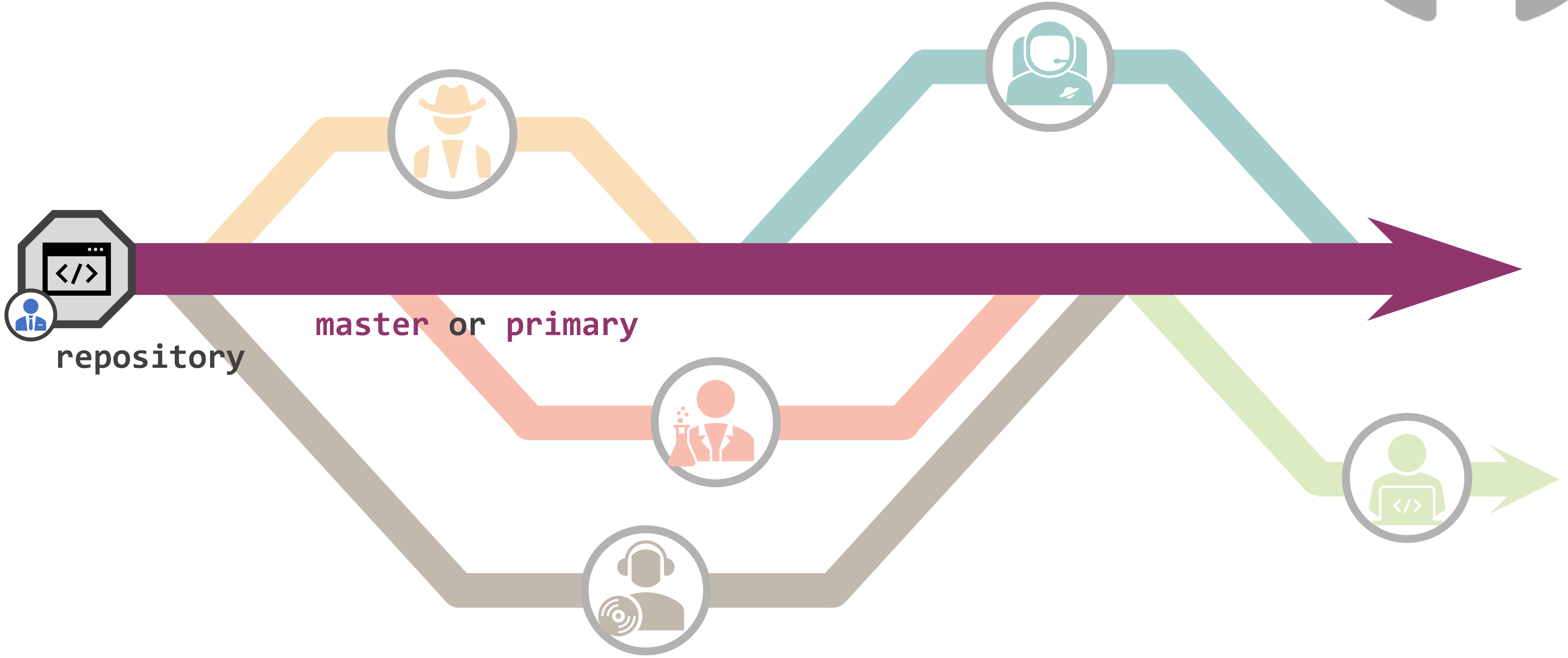
How VCS works



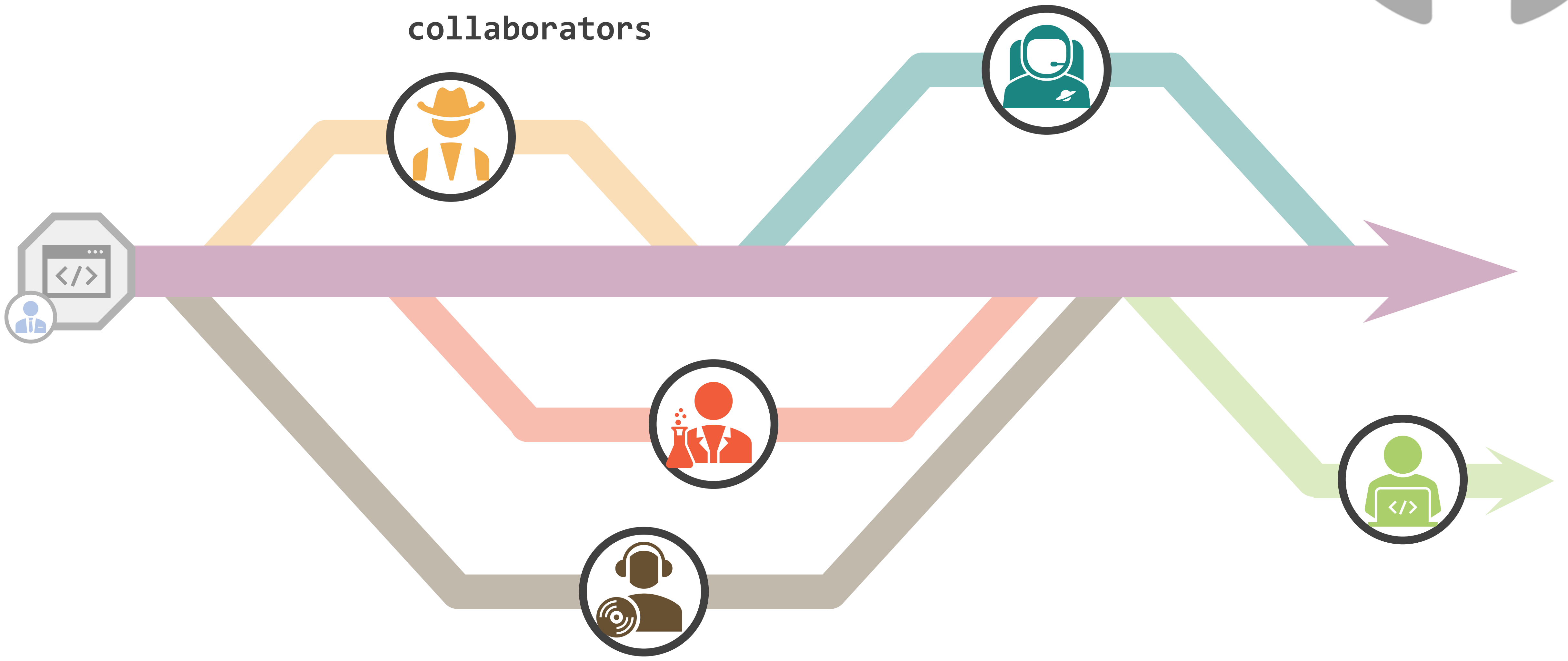
How GitHub works



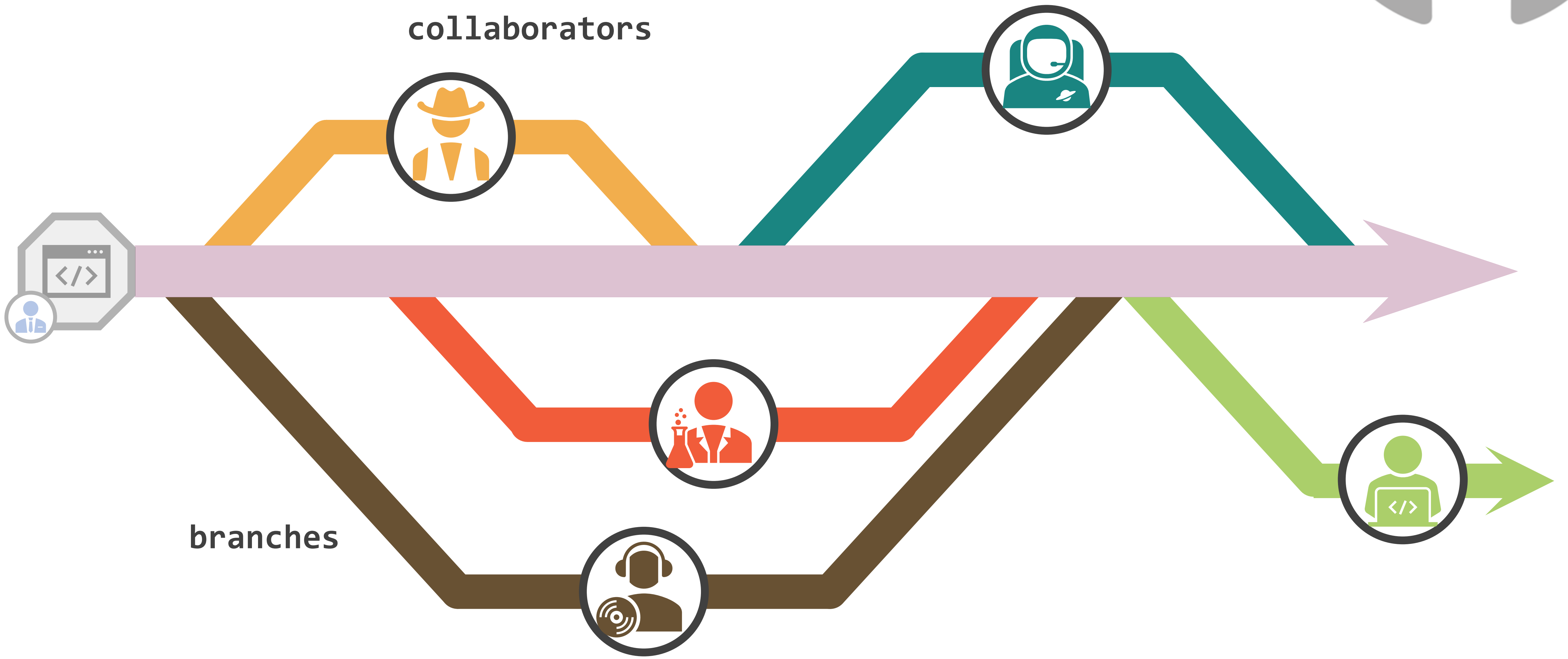
How GitHub works



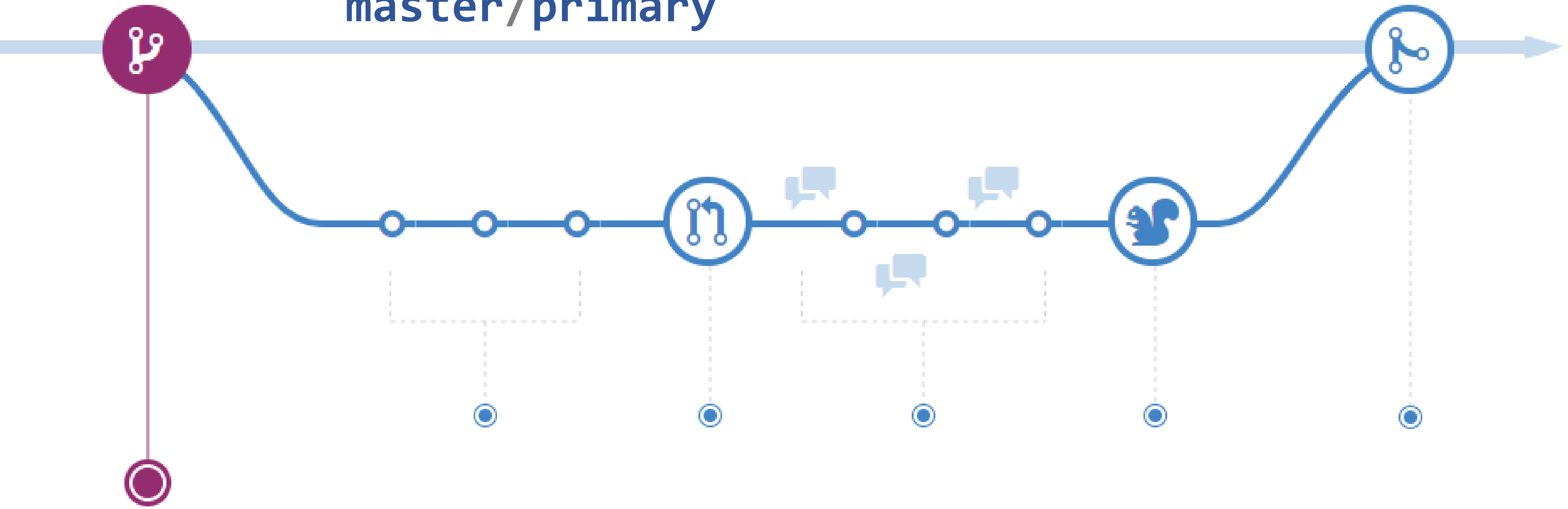
How GitHub works



How GitHub works



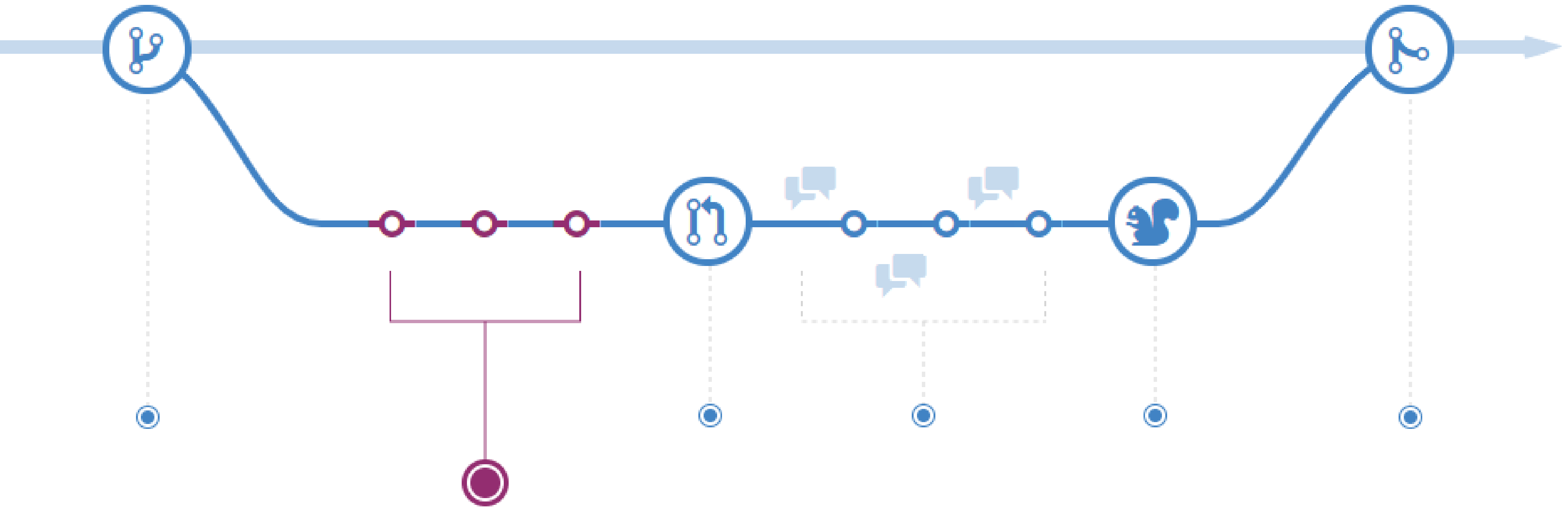
GitHub flow



Create a **branch**

*Copy files in an environment where you can **experiment** new ideas, without affecting the* **`master`**

GitHub flow

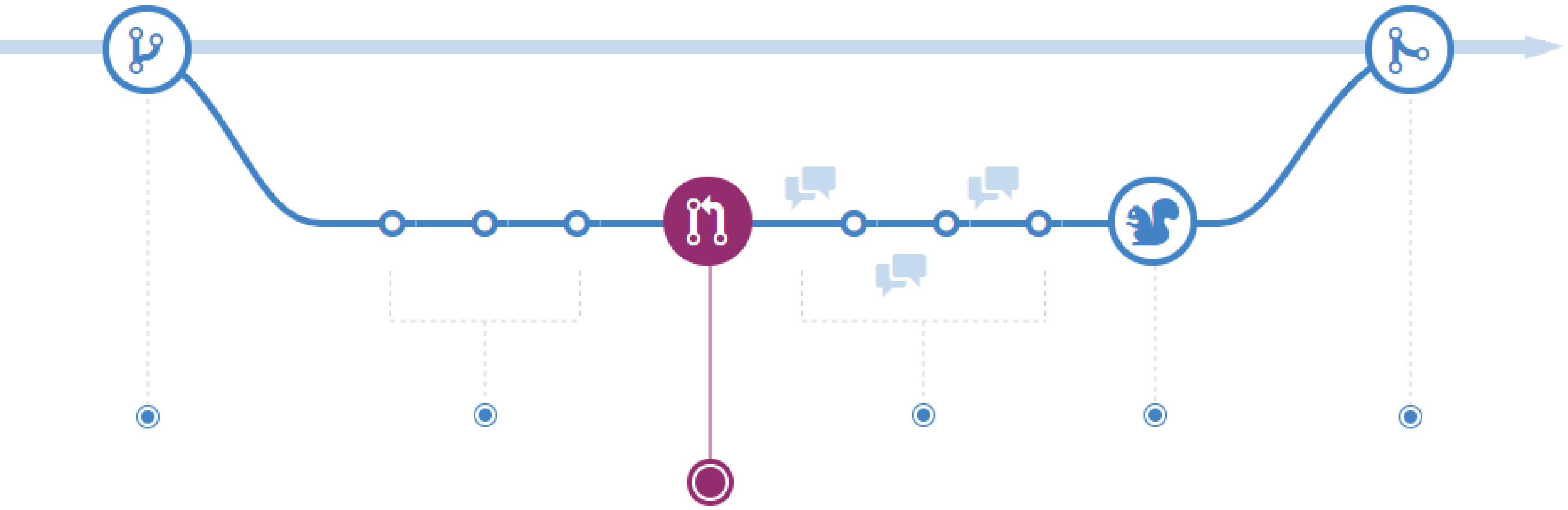


Add **commits**

Adding, editing or deleting files

*Creates a **transparent history** of your work: each* **commit** *has an associated message*

GitHub flow

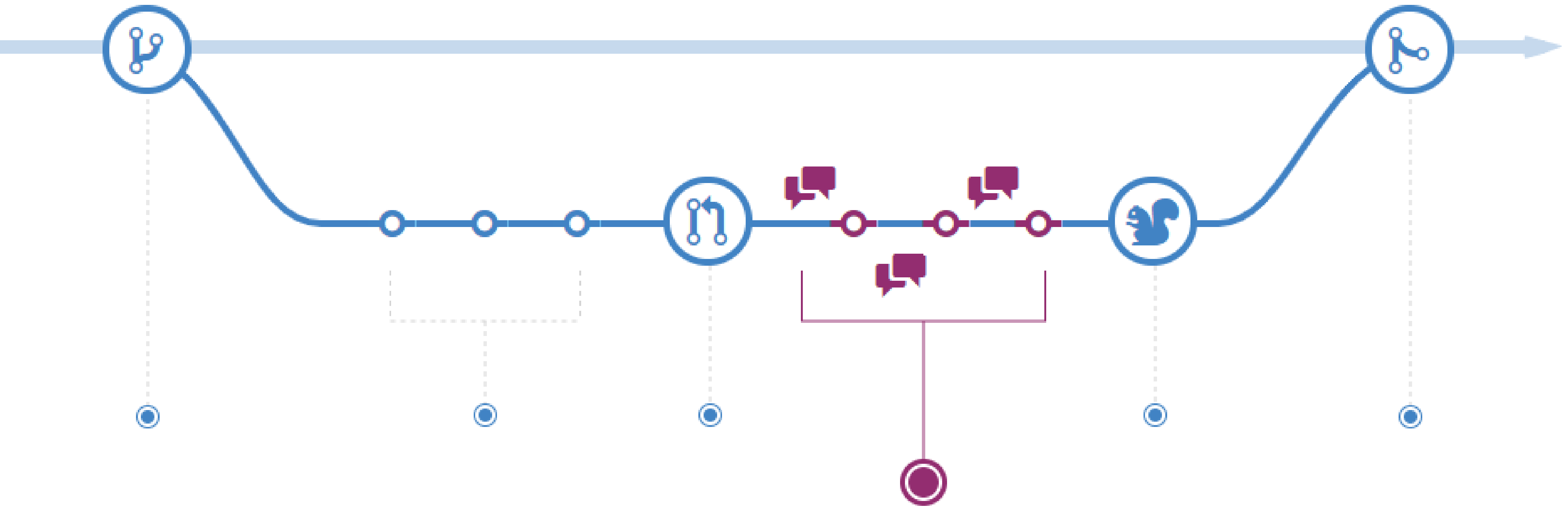


Open a `pull request`

Show your changes to other `collaborators` *and initiate discussion*

*When you're ready to **add your work**, when you want to **share ideas**, when you're stuck and **need help**…*

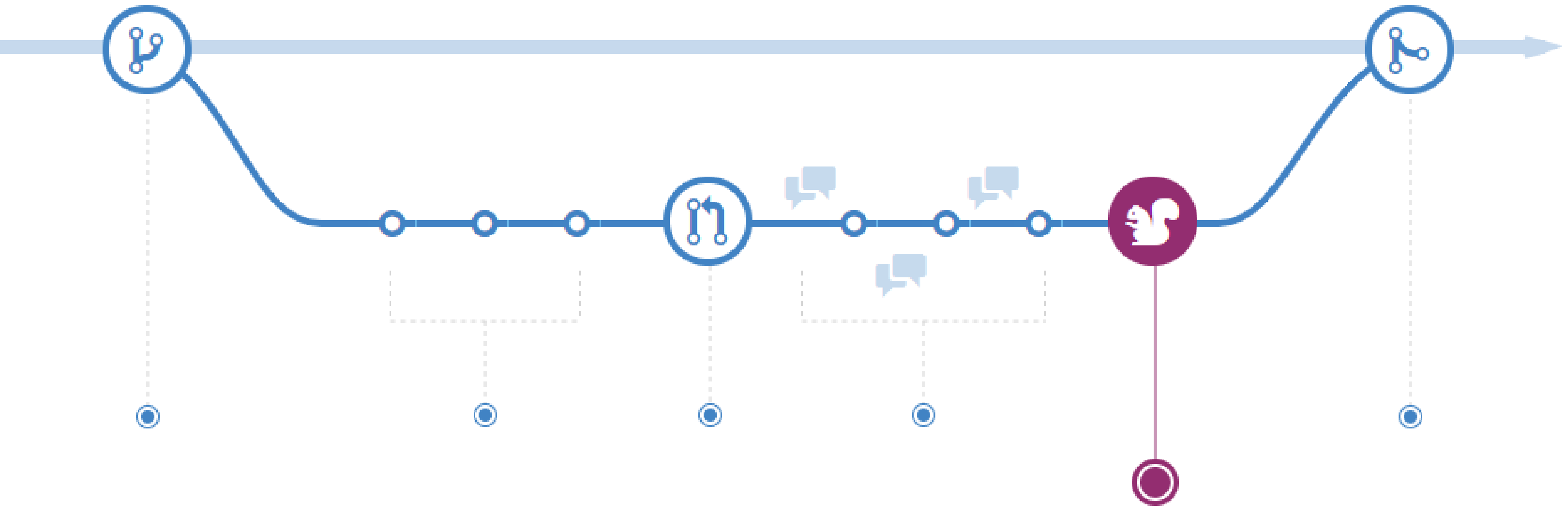
GitHub flow



Discuss and review code

Check if everything is fine

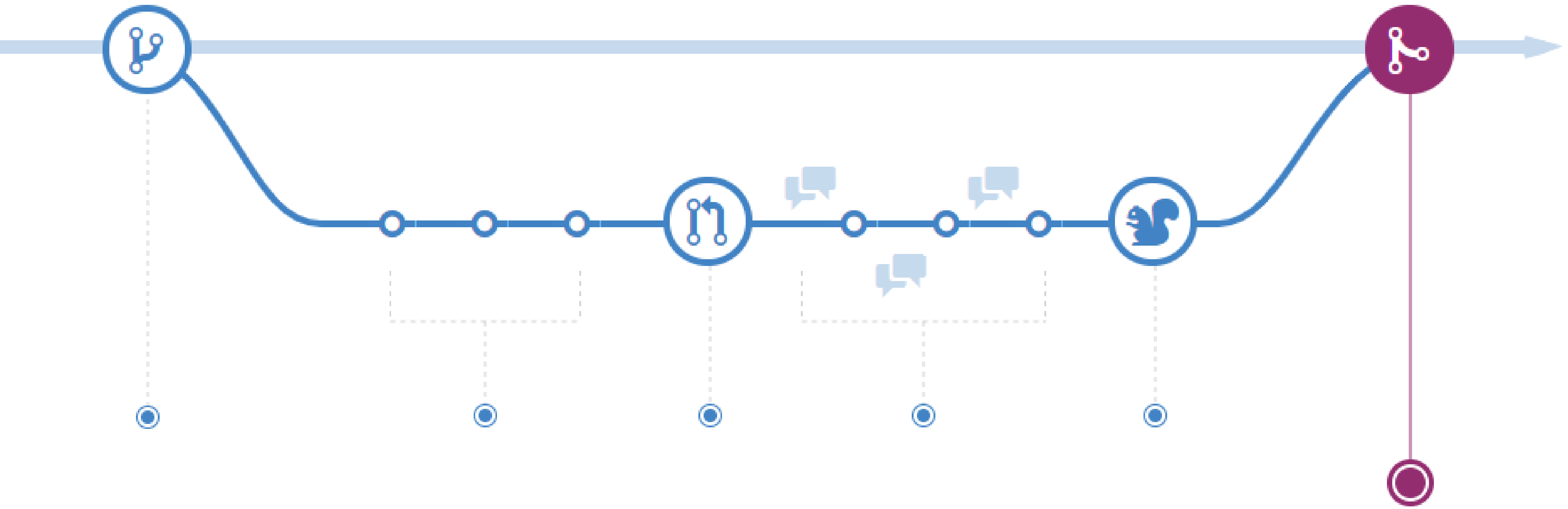
GitHub flow



Deploy

For final testing

GitHub flow

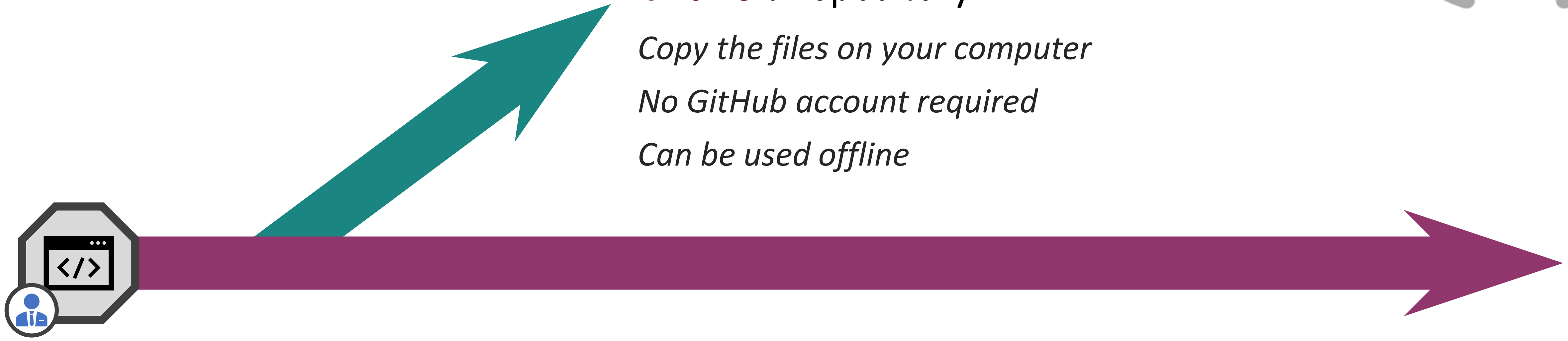


Merge

Merge your code into the **`master`** *branch*

Record is preserved

How GitHub works



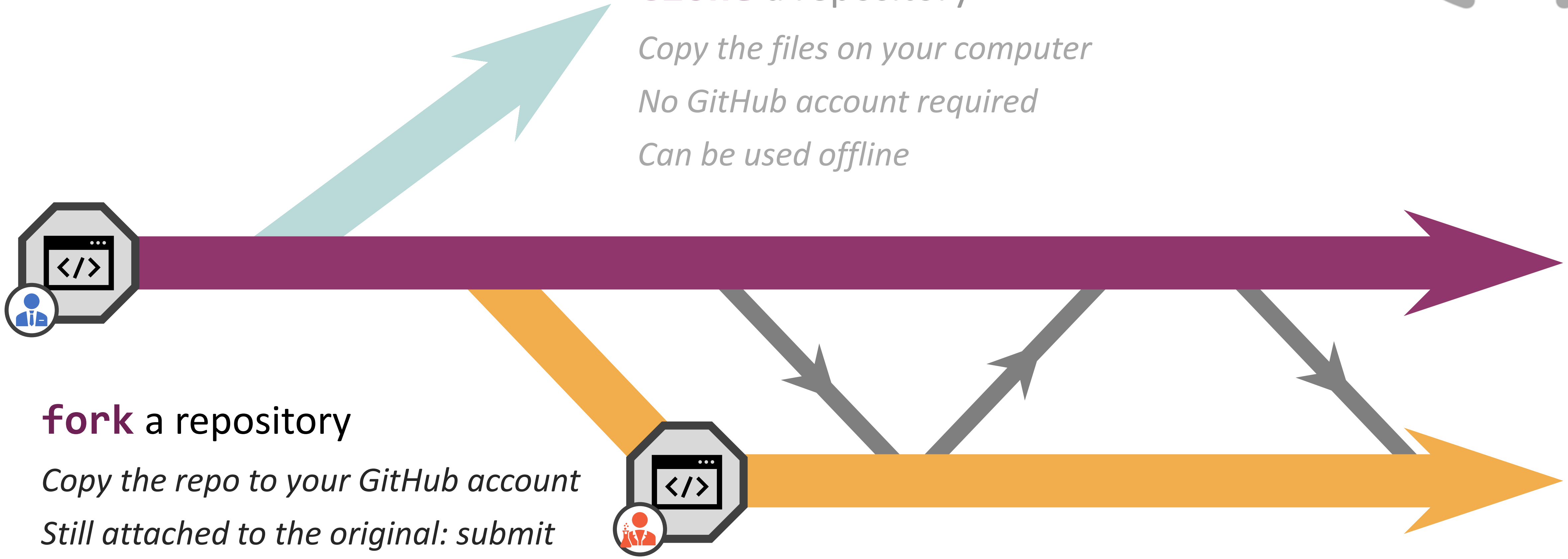
`clone` a repository

Copy the files on your computer

No GitHub account required

Can be used offline

How GitHub works



`clone` a repository

Copy the files on your computer

No GitHub account required

Can be used offline

`fork` a repository

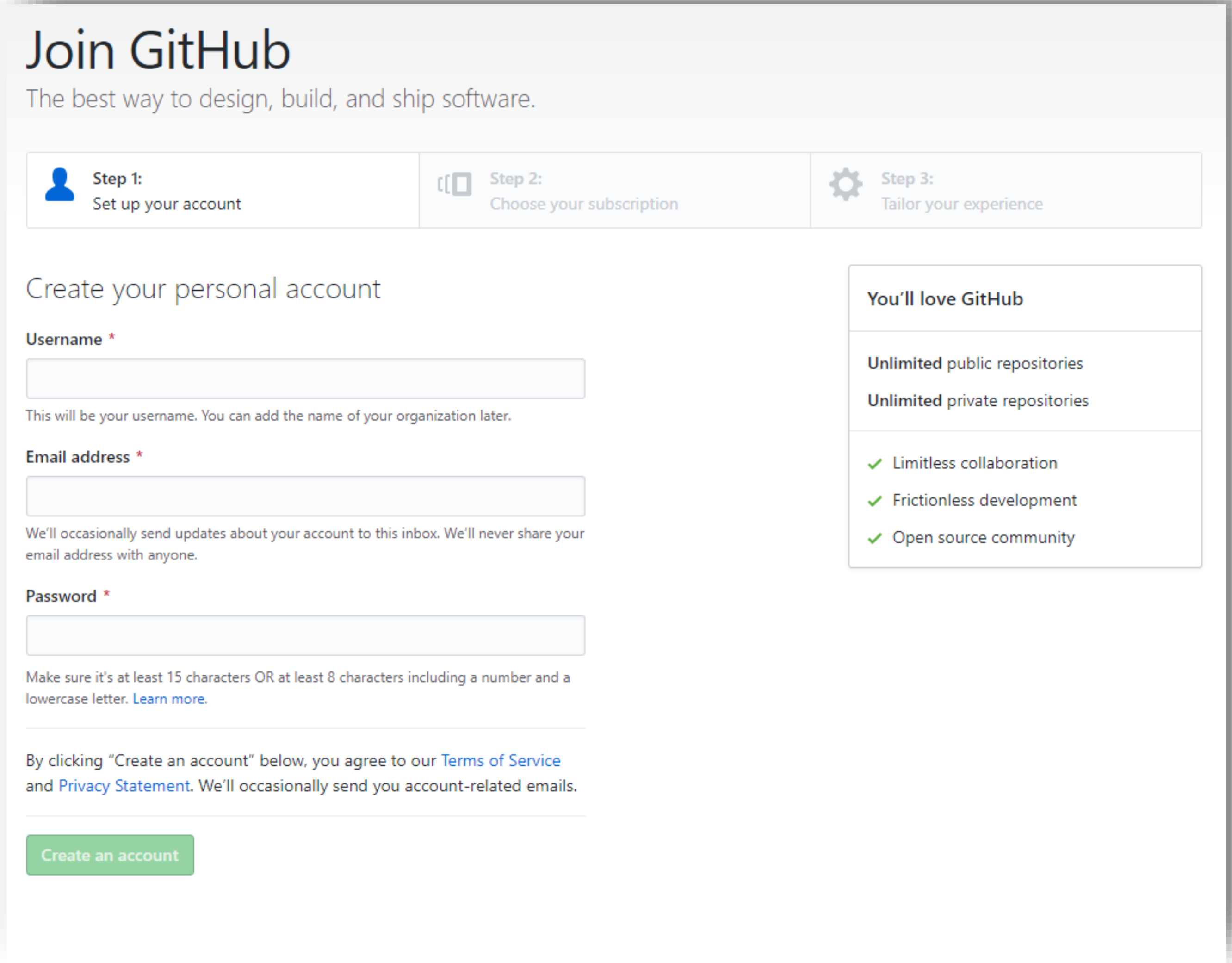
Copy the repo to your GitHub account

Still attached to the original: submit pull requests to it, import updates from it

Setting up

A new account

Go to
github.com/join



Join GitHub

The best way to design, build, and ship software.

Create your personal account

Username *

This will be your username. You can add the name of your organization later.

Email address *

We'll occasionally send updates about your account to this inbox. We'll never share your email address with anyone.

Password *

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

By clicking "Create an account" below, you agree to our Terms of Service and Privacy Statement. We'll occasionally send you account-related emails.

You'll love GitHub

Unlimited public repositories

Unlimited private repositories

✓ Limitless collaboration

✓ Frictionless development

✓ Open source community

Setting up

Your dashboard

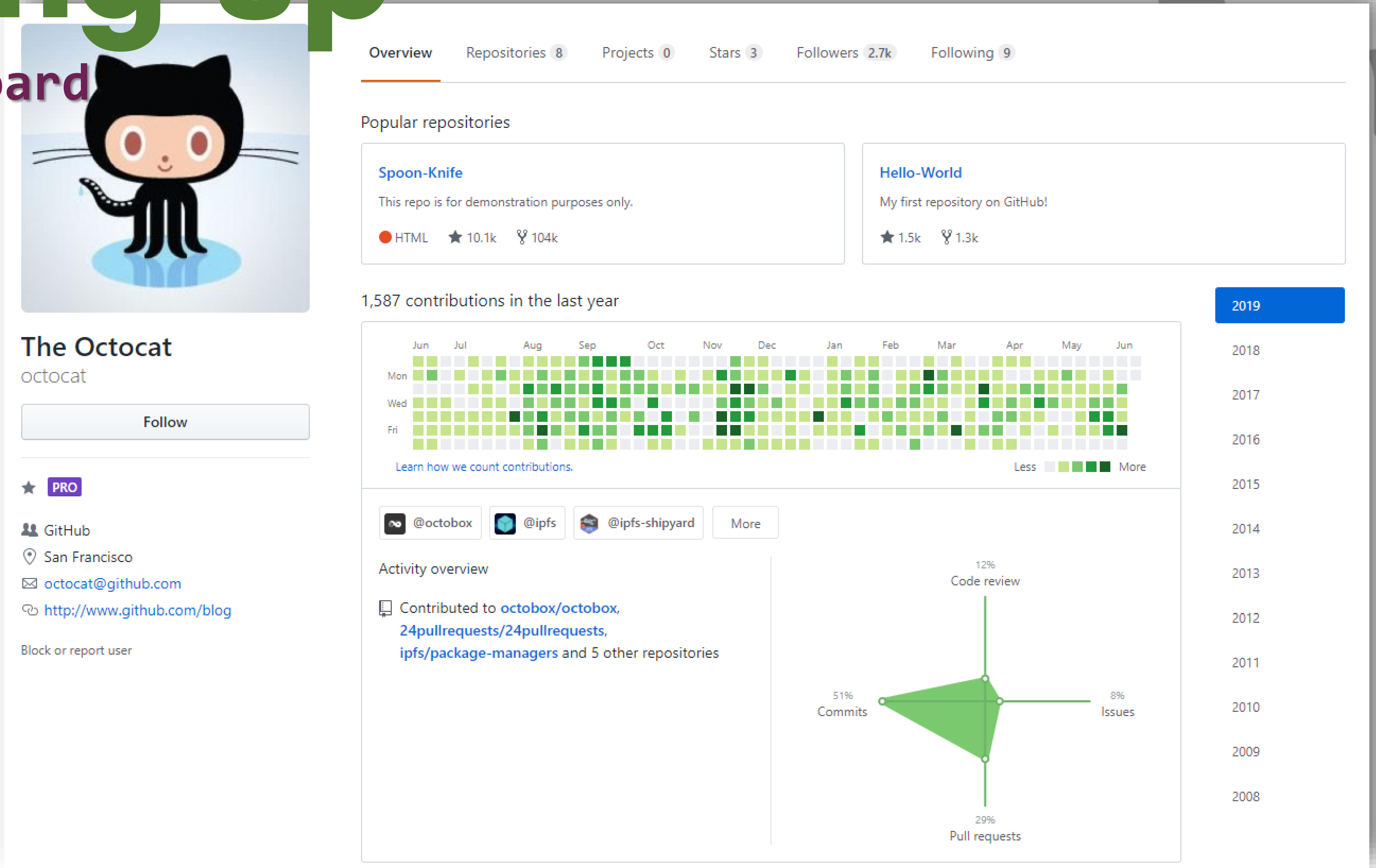
Main hub for your activities

Public

profile info,

repositories,

contributions…



The Octocat
octocat

Follow

PRO

GitHub
San Francisco
octocat@github.com
http://www.github.com/blog

Block or report user

Overview · Repositories 8 · Projects 0 · Stars 3 · Followers 2.7k · Following 9

Popular repositories

Spoon-Knife
This repo is for demonstration purposes only.
HTML ★ 10.1k 104k

Hello-World
My first repository on GitHub!
★ 1.5k 1.3k

1,587 contributions in the last year

Learn how we count contributions.

@octobox @ipfs @ipfs-shipyard More

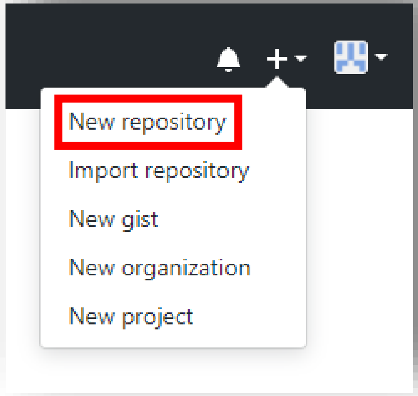
Activity overview

Contributed to octobox/octobox, 24pullrequests/24pullrequests, ipfs/package-managers and 5 other repositories

Setting up

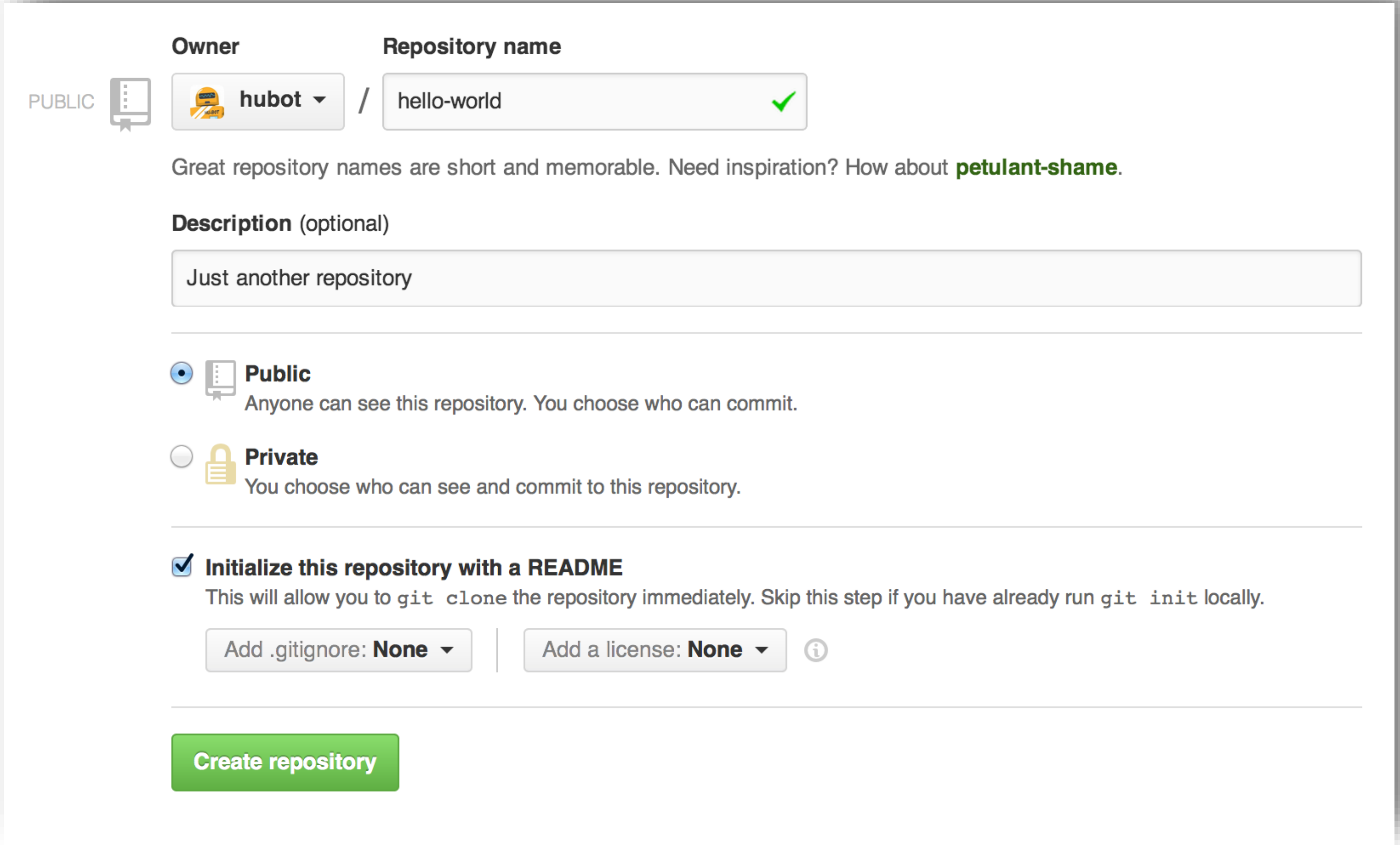
Your first repository

Upper right corner:



Choose:

- name
- description
- status *(public/private)*
- README



Owner: hubot / Repository name: hello-world ✓

Great repository names are short and memorable. Need inspiration? How about **petulant-shame**.

Description (optional)

Just another repository

[✓ **Public**] Anyone can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

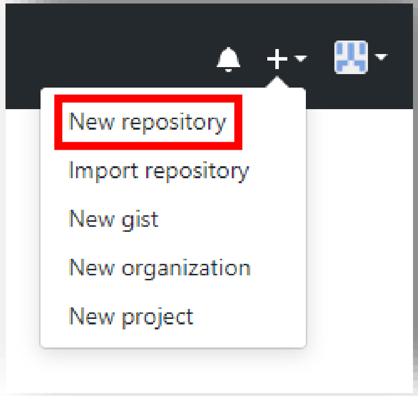
[✓] **Initialize this repository with a README**
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: **None** | Add a license: **None**

Setting up

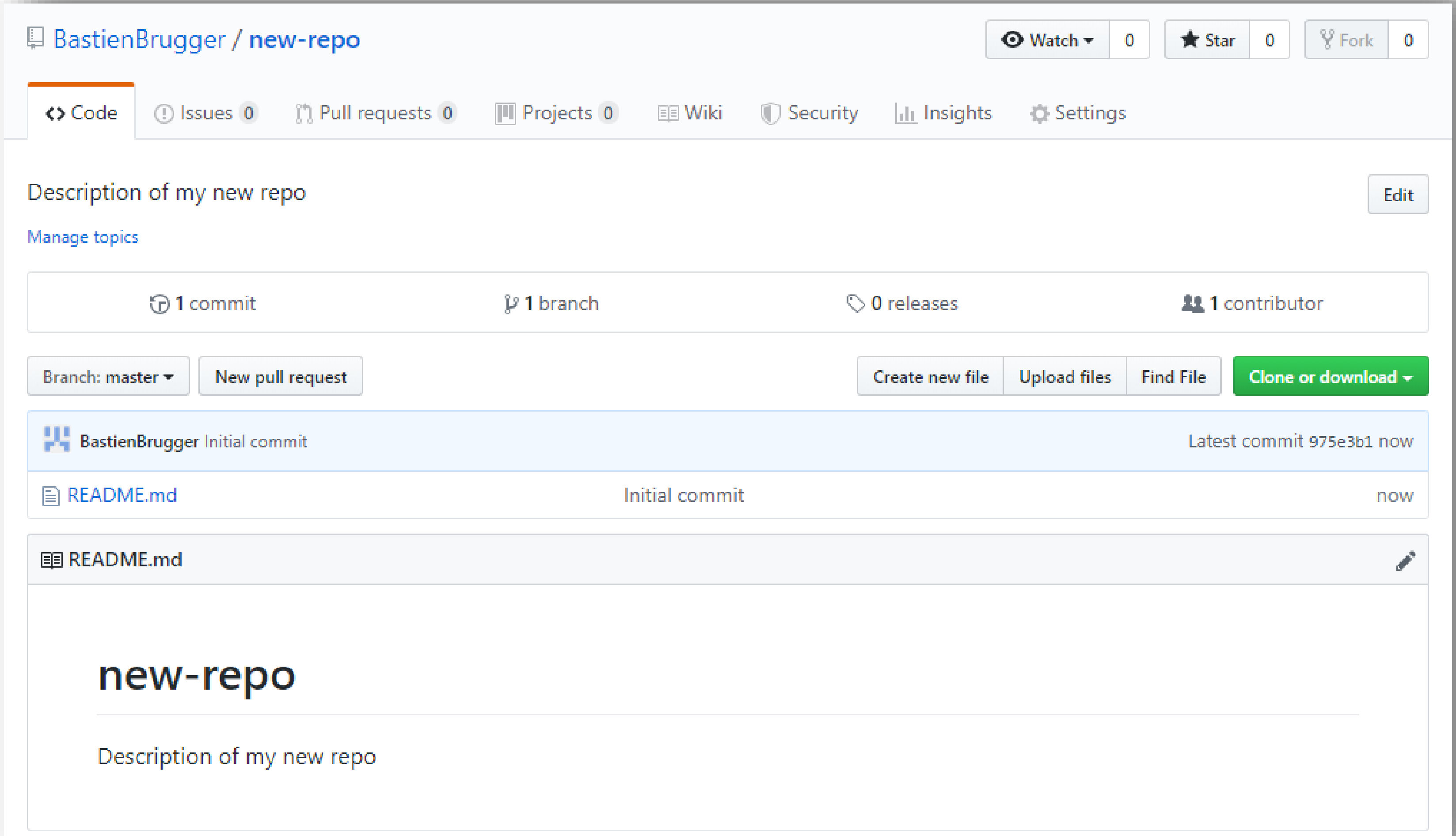
Your first repository

Upper right corner:



Choose:

- name
- description
- status *(public/private)*
- README



Description of my new repo

BastienBrugger Initial commit — Latest commit 975e3b1 now

README.md — Initial commit — now

new-repo

Description of my new repo

Managing repos

Three options:

1 Browser

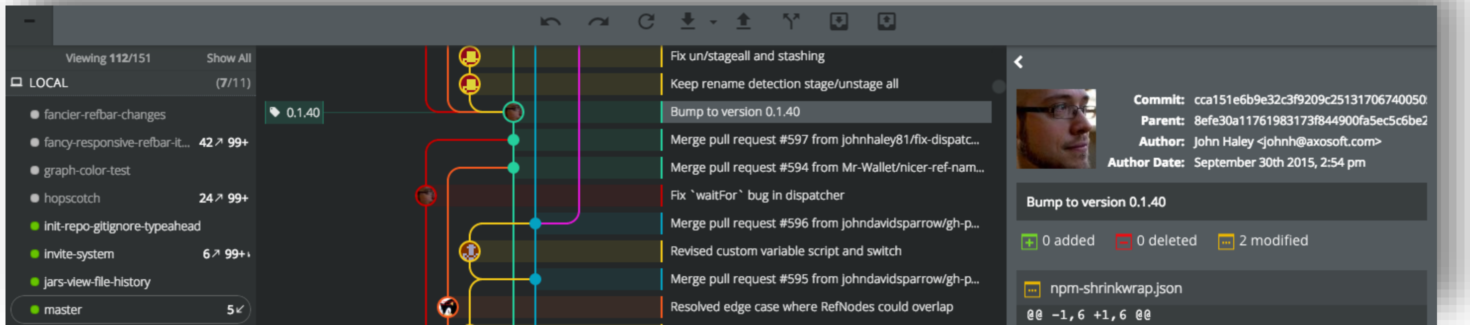
- Available everywhere, no compatibility issues
- Create/fork repos, manage files
- Be social and discover existing repos
- *Impossible to run your program online*

Managing repos

Three options:

2 Graphical User Interface (GUI)

- Multiple software options available
- Combines text editor + Git functions *(commit, pull request…)*
- User-friendly + graphic representation of collaboration



Viewing 112/151 Show All
LOCAL (7/11)

- fancier-refbar-changes
- fancy-responsive-refbar-it... 42↗ 99+
- graph-color-test
- hopscotch 24↗ 99+
- init-repo-gitignore-typeahead
- invite-system 6↗ 99+
- jars-view-file-history
- master 5↙

0.1.40

- Fix un/stageall and stashing
- Keep rename detection stage/unstage all
- Bump to version 0.1.40
- Merge pull request #597 from johnhaley81/fix-dispatc...
- Merge pull request #594 from Mr-Wallet/nicer-ref-nam...
- Fix `waitFor` bug in dispatcher
- Merge pull request #596 from johndavidsparrow/gh-p...
- Revised custom variable script and switch
- Merge pull request #595 from johndavidsparrow/gh-p...
- Resolved edge case where RefNodes could overlap

Commit: cca151e6b9e32c3f9209c25131706740050
Parent: 8efe30a11761983173f844900fa5ec5c6be2
Author: John Haley <johnh@axosoft.com>
Author Date: September 30th 2015, 2:54 pm

Bump to version 0.1.40

0 added 0 deleted 2 modified

npm-shrinkwrap.json

```
@@ -1,6 +1,6 @@
```

Managing repos

Three options:

3 Command Line Interface

- Control over everything you do
- Automation via custom scripts
- Most documented method online

Managing repos

Three options:

1	Browser	github.com/[username]
2	GUI	git-scm.com/downloads/guis
3	Command line	git-scm.com/download/[linux/mac/win]