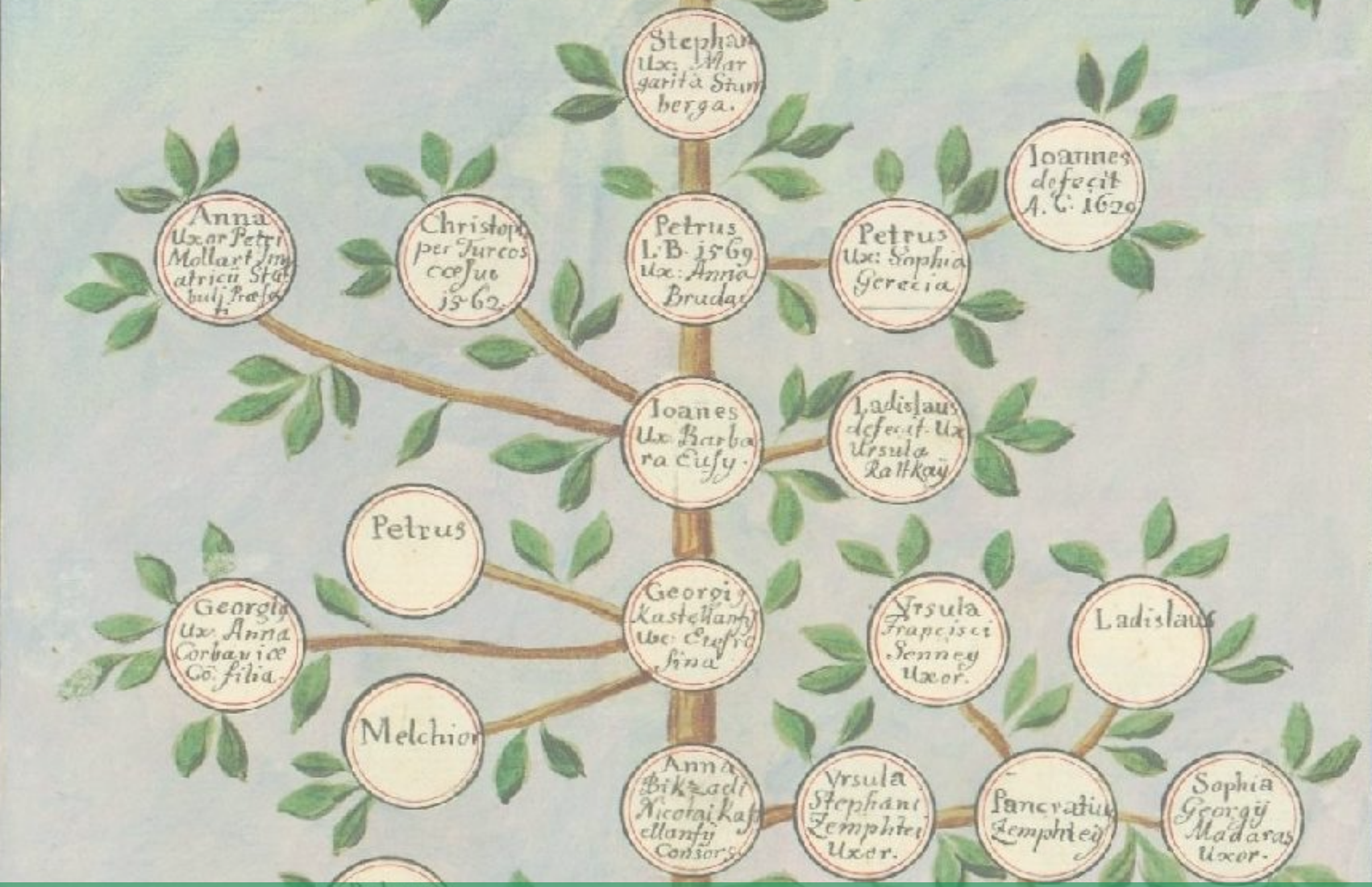
Git for Everybody

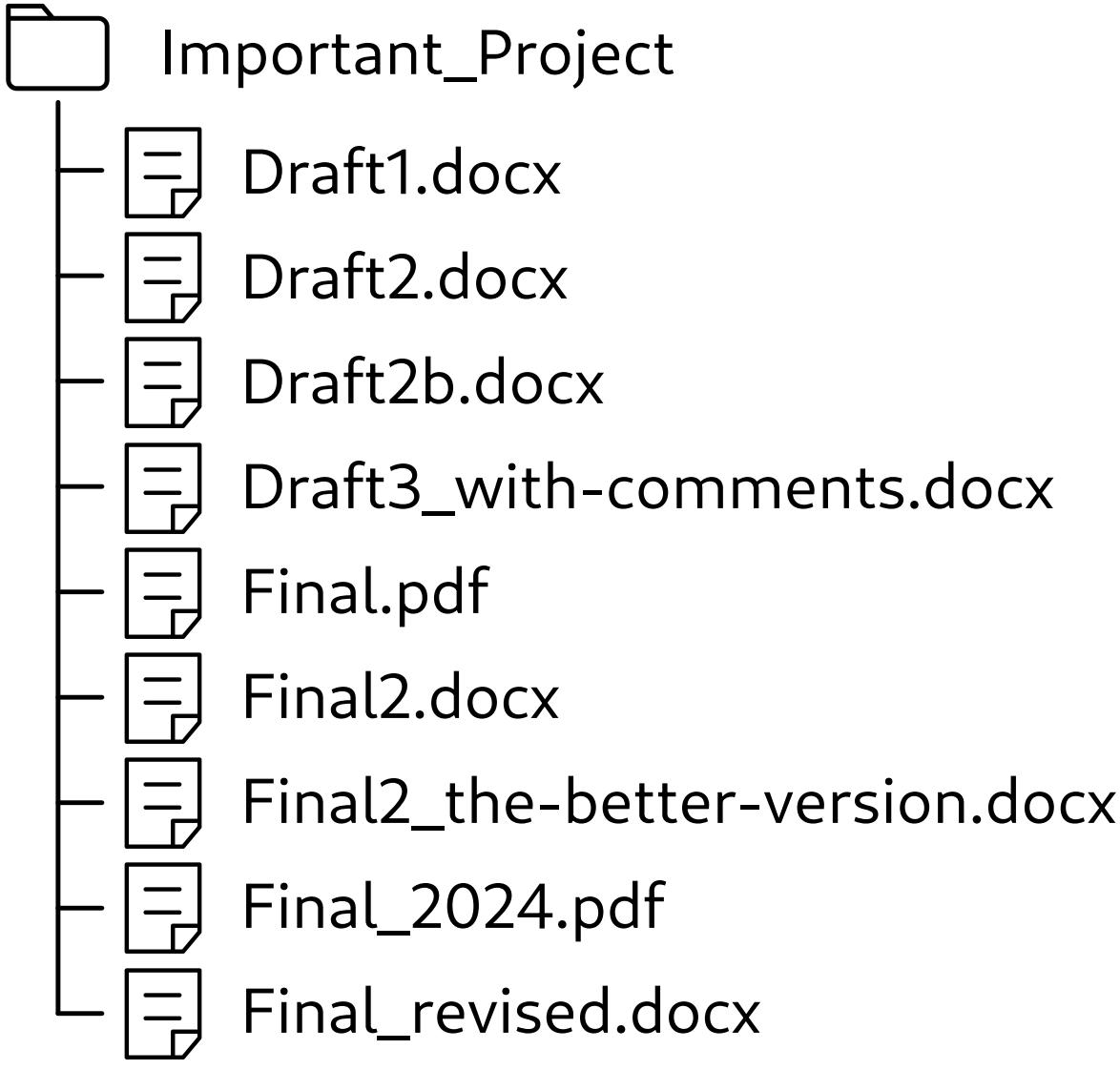
Intro to Git and GitHub for Non-Coders

github.com/tiago-rorke/git-for-everybody

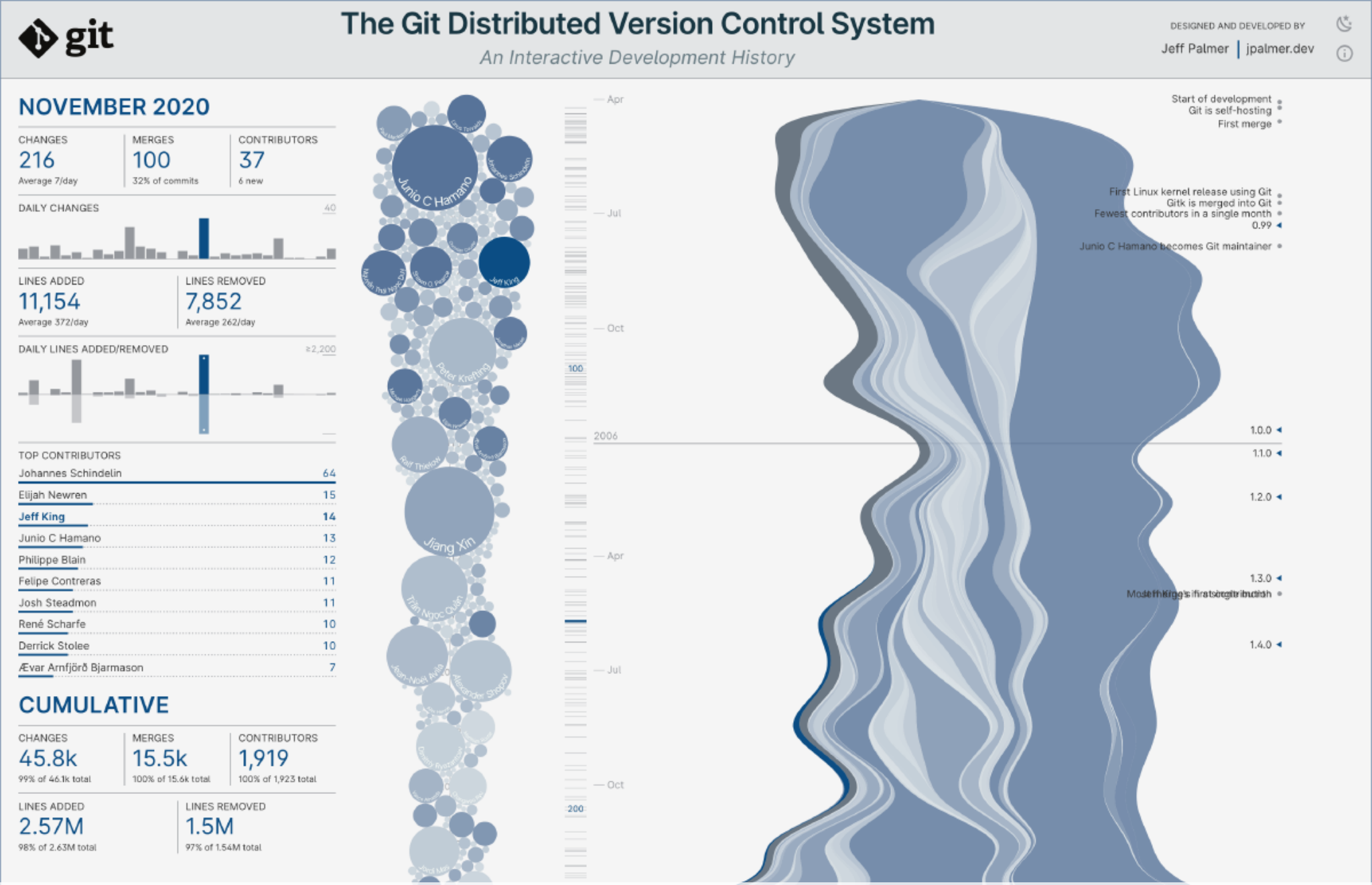


What is Git, and what can you do with it?

Version Control



Version Control

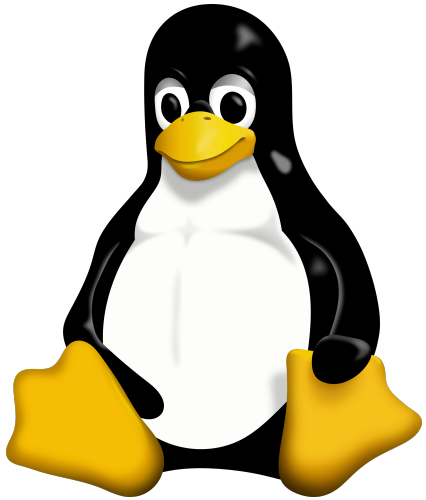


Jeff Palmer - Git: An Interactive Development History
jpalmer.dev

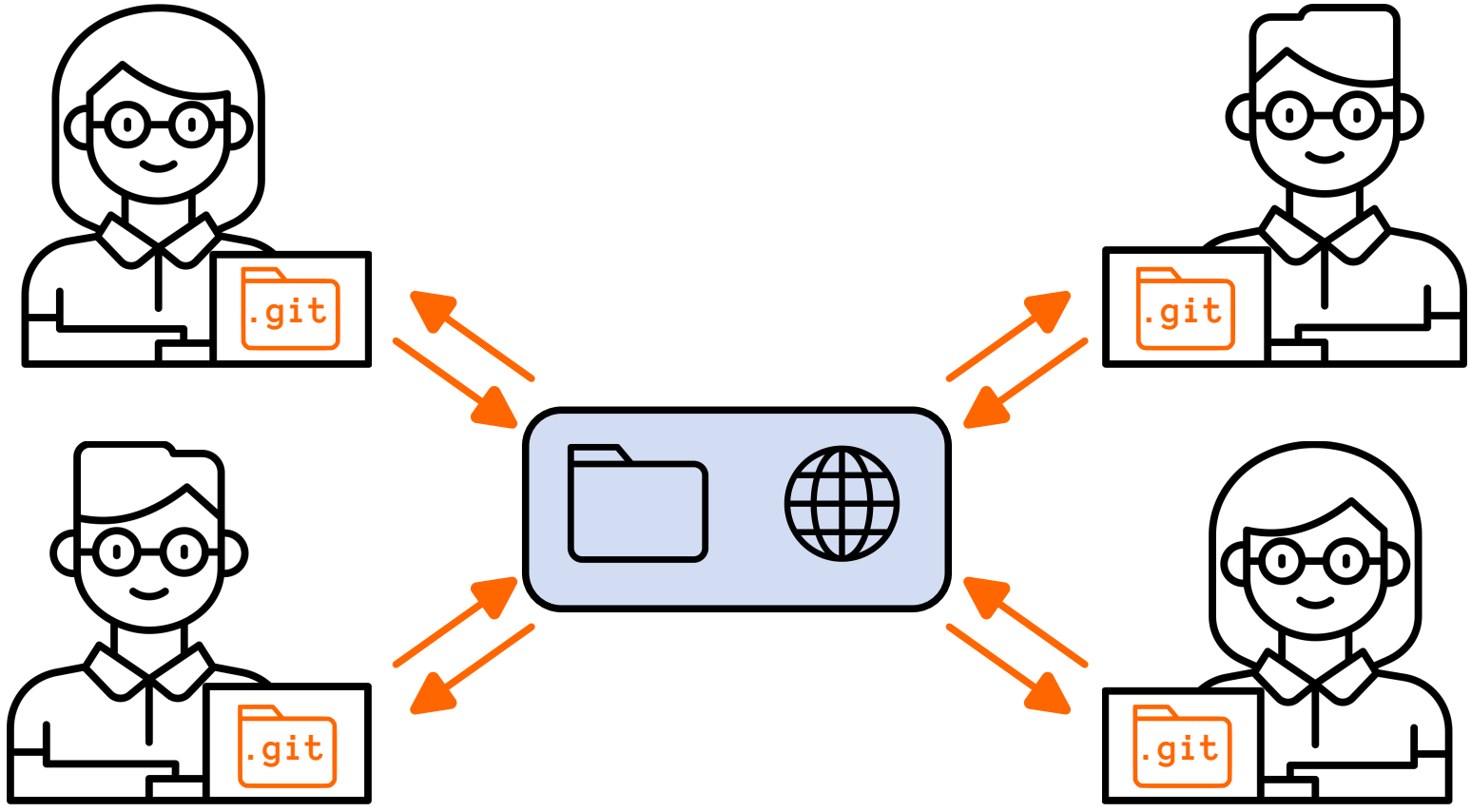
Git

An open source project

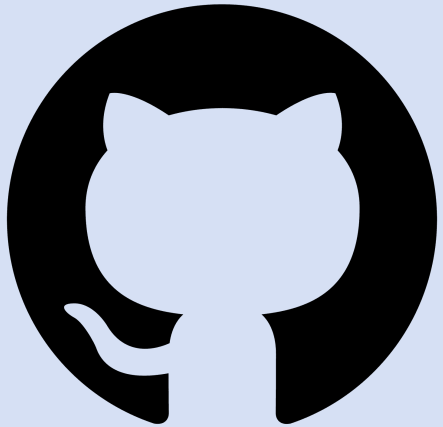
Development started by Linux creator Linus Torvalds in 2005



Distributed Version Control



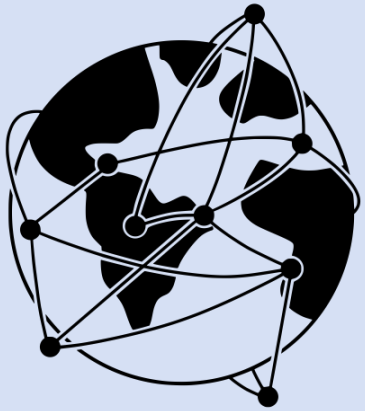
git



GitHub

git-scm.com

github.com



Other Git Hosting Platforms/Networks

GitLab

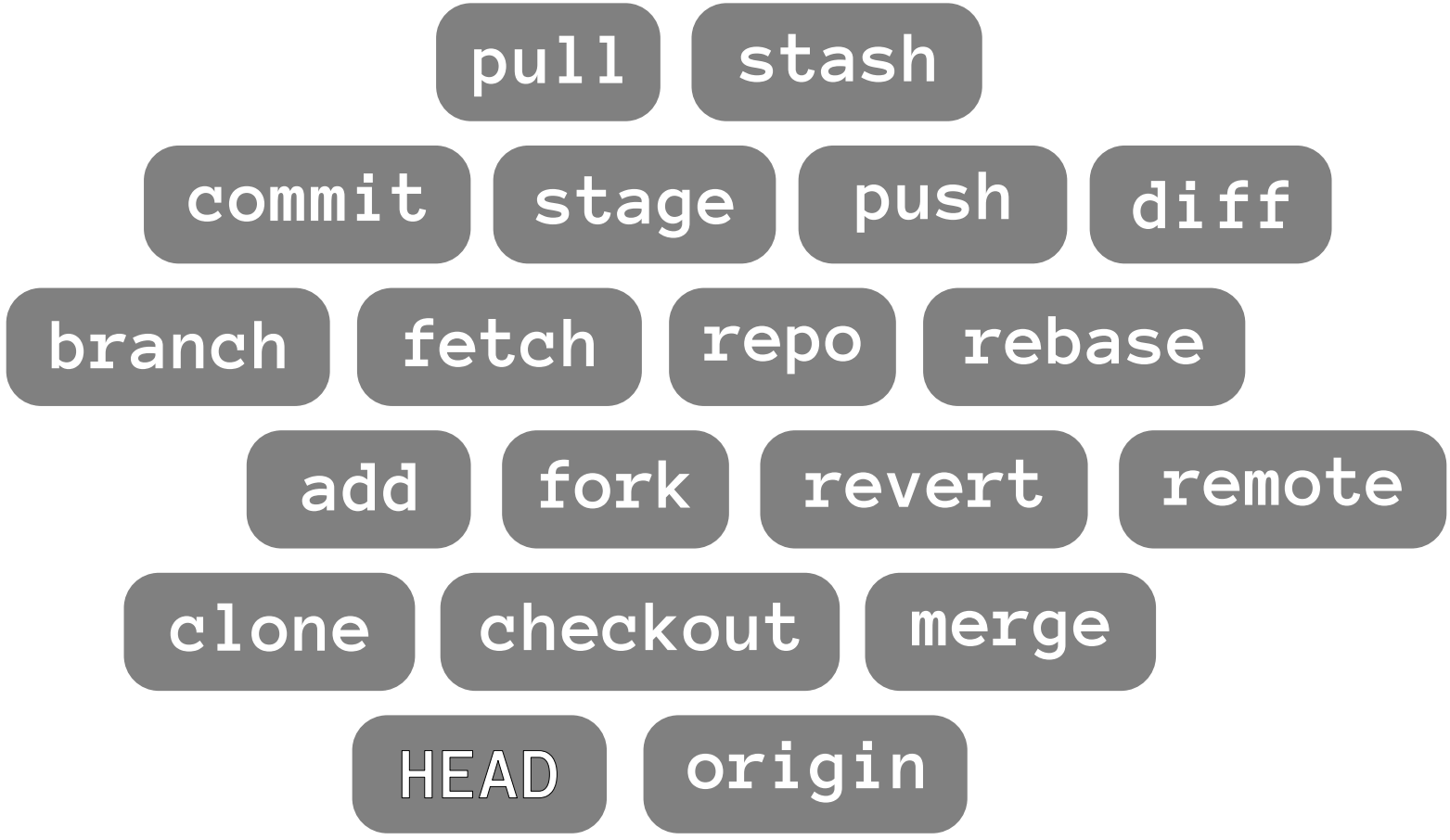
Gitea

Bitbucket

SOURCEFORGE

radicle

Git Is a Language



When to use Git?

You **could** use Git anywhere, but here are some examples of when you might **want** to use it:

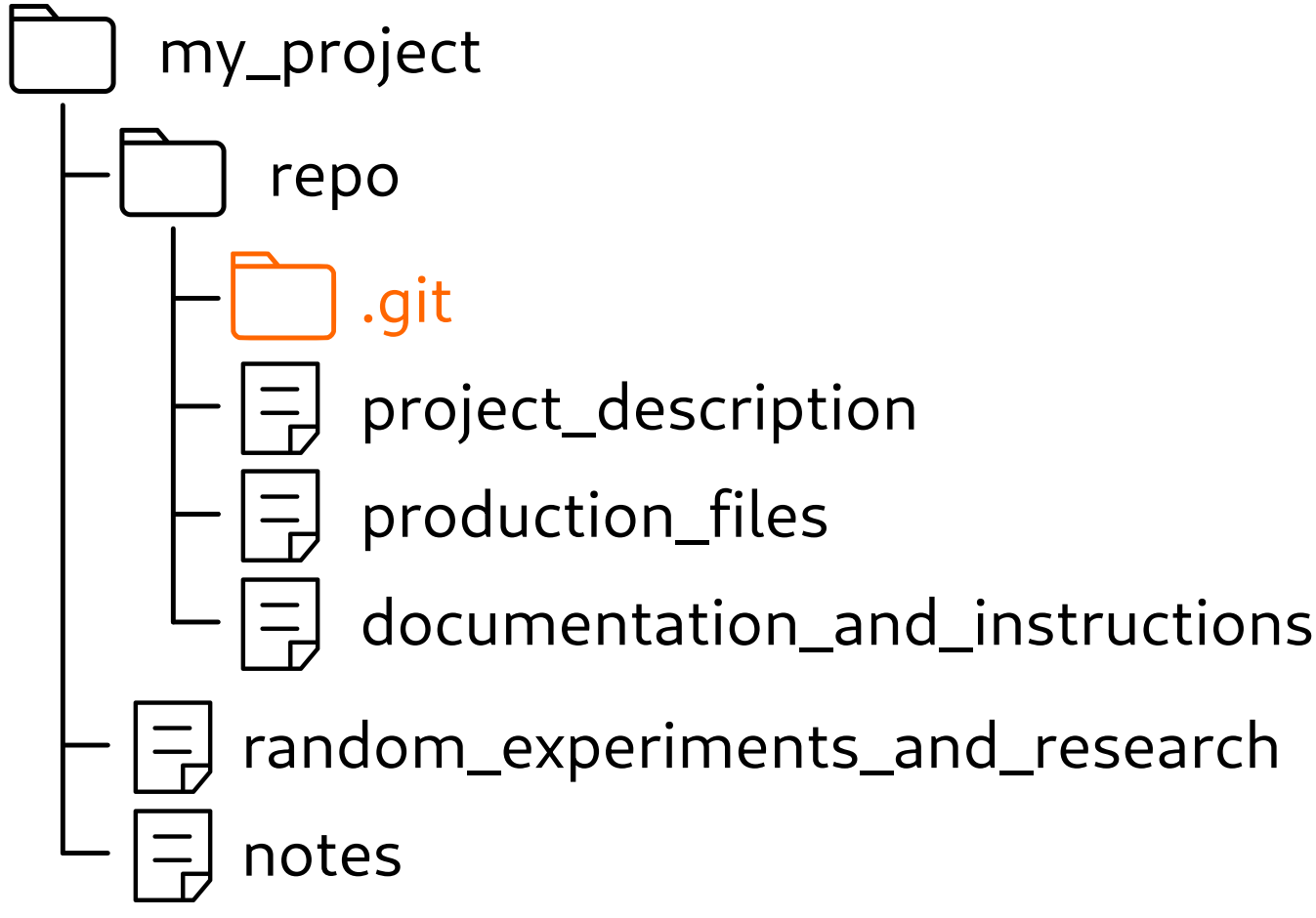
`locally`

- Working with code
- Writing
- Audio/Video compositions
- Managing assets
- Archives
- For tracking and metrics

`publicly`

- Documentation
- Collaboration
- "Open sourcing" a project
- Communication and discussion
- Data/knowledge sharing

How I use Git


my_project
repo
.git
project_description
production_files
documentation_and_instructions
random_experiments_and_research
notes

What works with Git

You **can** put any kind of file in a Git repository, but some kinds of files will work better than others:

`great`

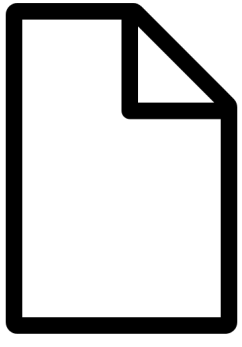
- Text-based files
- Small/lightweight files including images, PDFs, and other media
- Lightweight project files eg: .kdenlive

`awkward`

- Large media files such as audio and video
- Large archives (eg .zip)
- Heavy project files eg: photoshop .psd

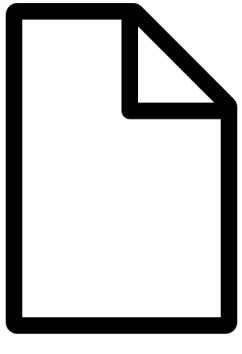
Binary vs Text

Git efficiently tracks changes in text files, less so for binary files



.md
.txt
.html
.svg

```
<?xml encoding="UTF-8" ?>
<svg
   width="105mm"
   height="148mm"
```



.png
.pdf
.docx
.mp4

```
01101000 01100001 01110110
01100101 00100000 01100001
01101110 00100000 11110000
10011111 10100101 10011010
```

Markdown

`.md`

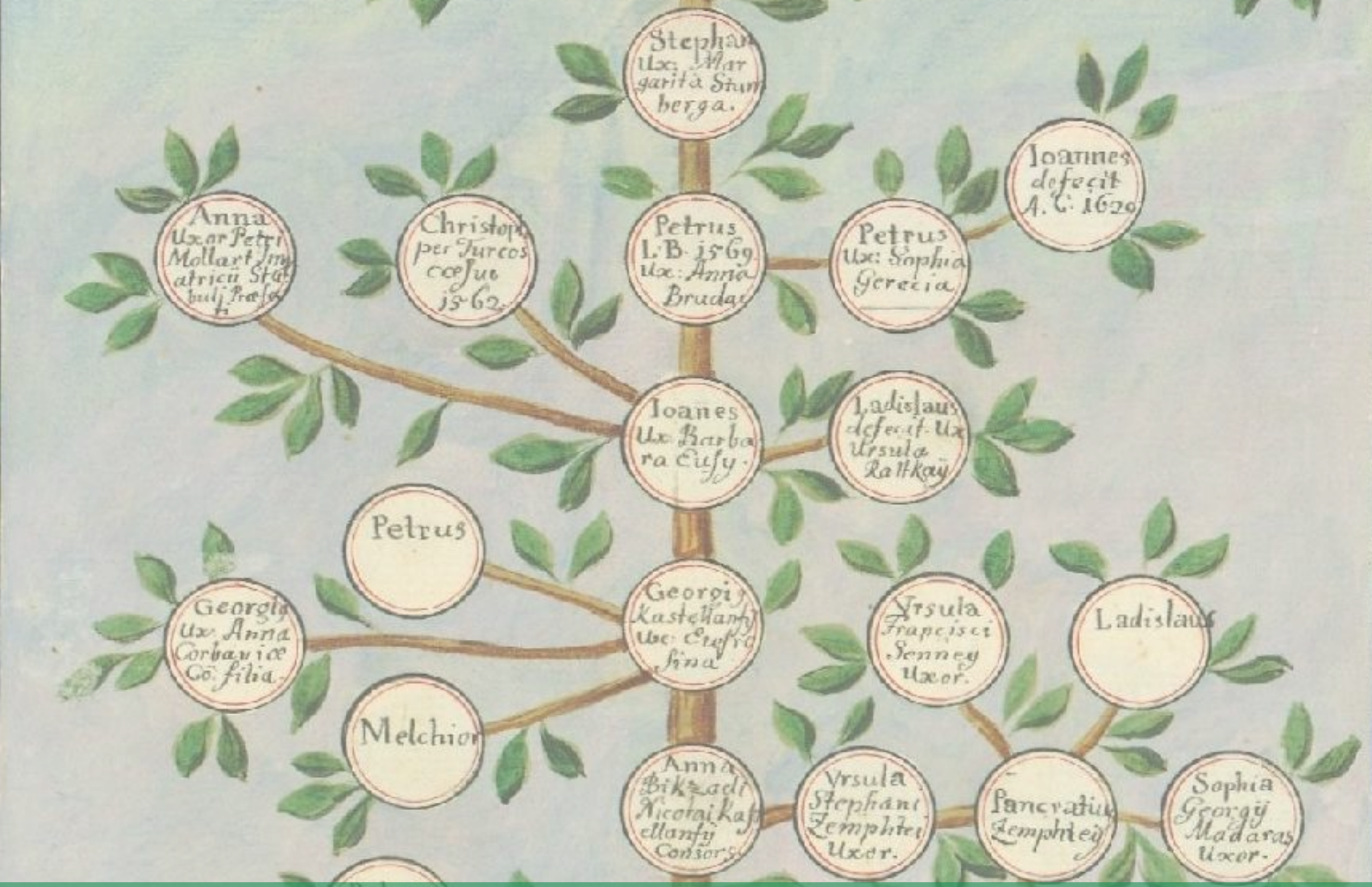
Markdown is a "markup language" for formatting text

Created by John Gruber in 2004, developed together with Aaron Swartz

Designed to be easy to read as plain text


John Gruber

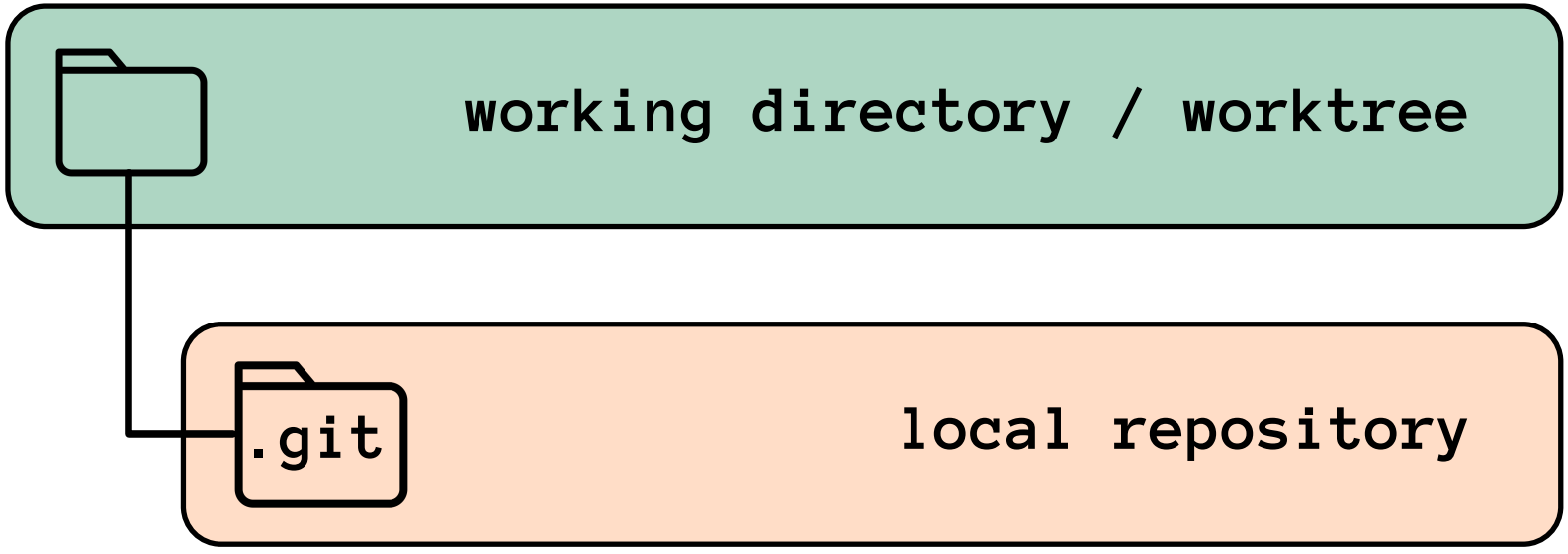

Aaron Swartz



How do you use Git?

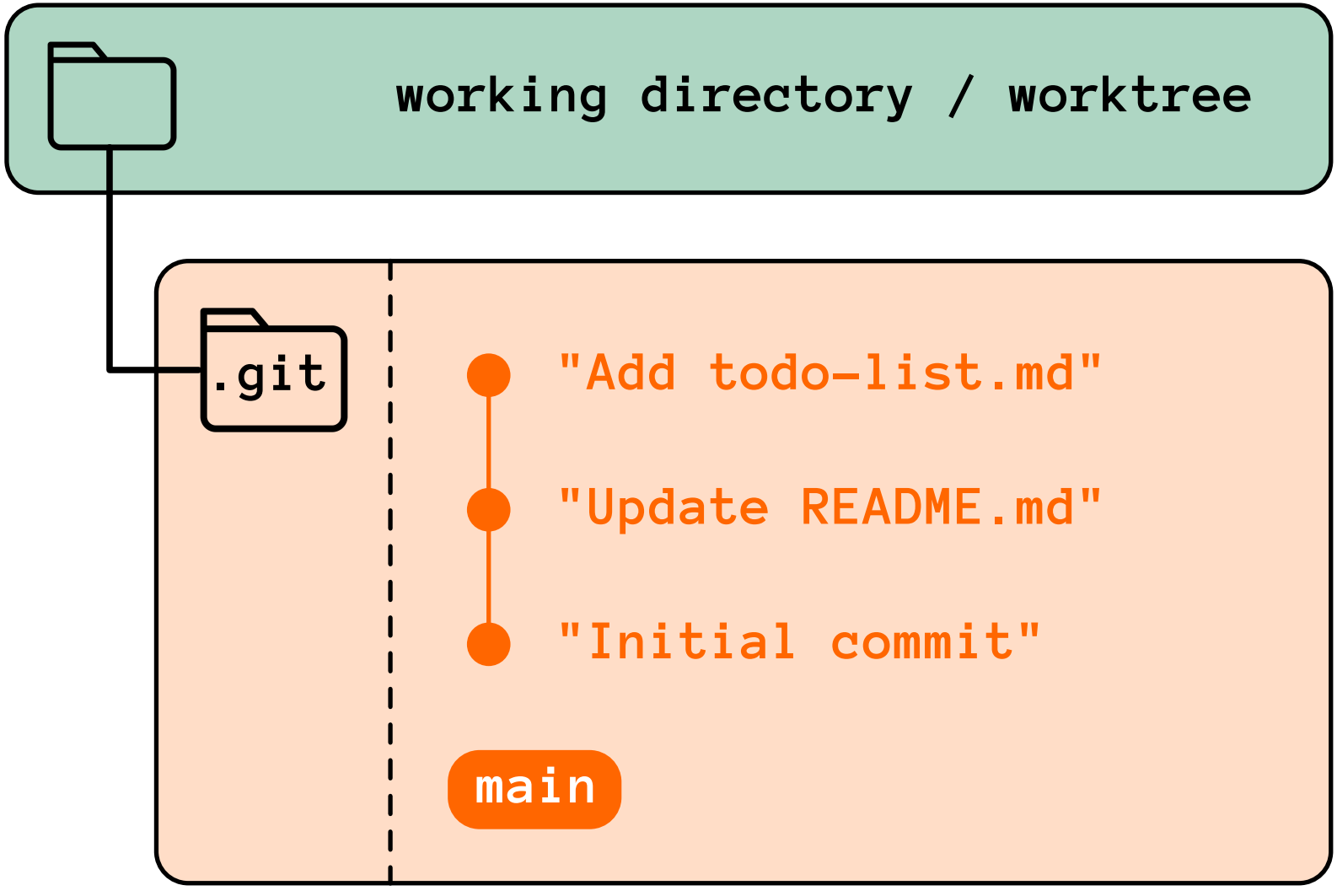
Init will turn a folder into a local git repository, creating a hidden **.git** folder

`git init`



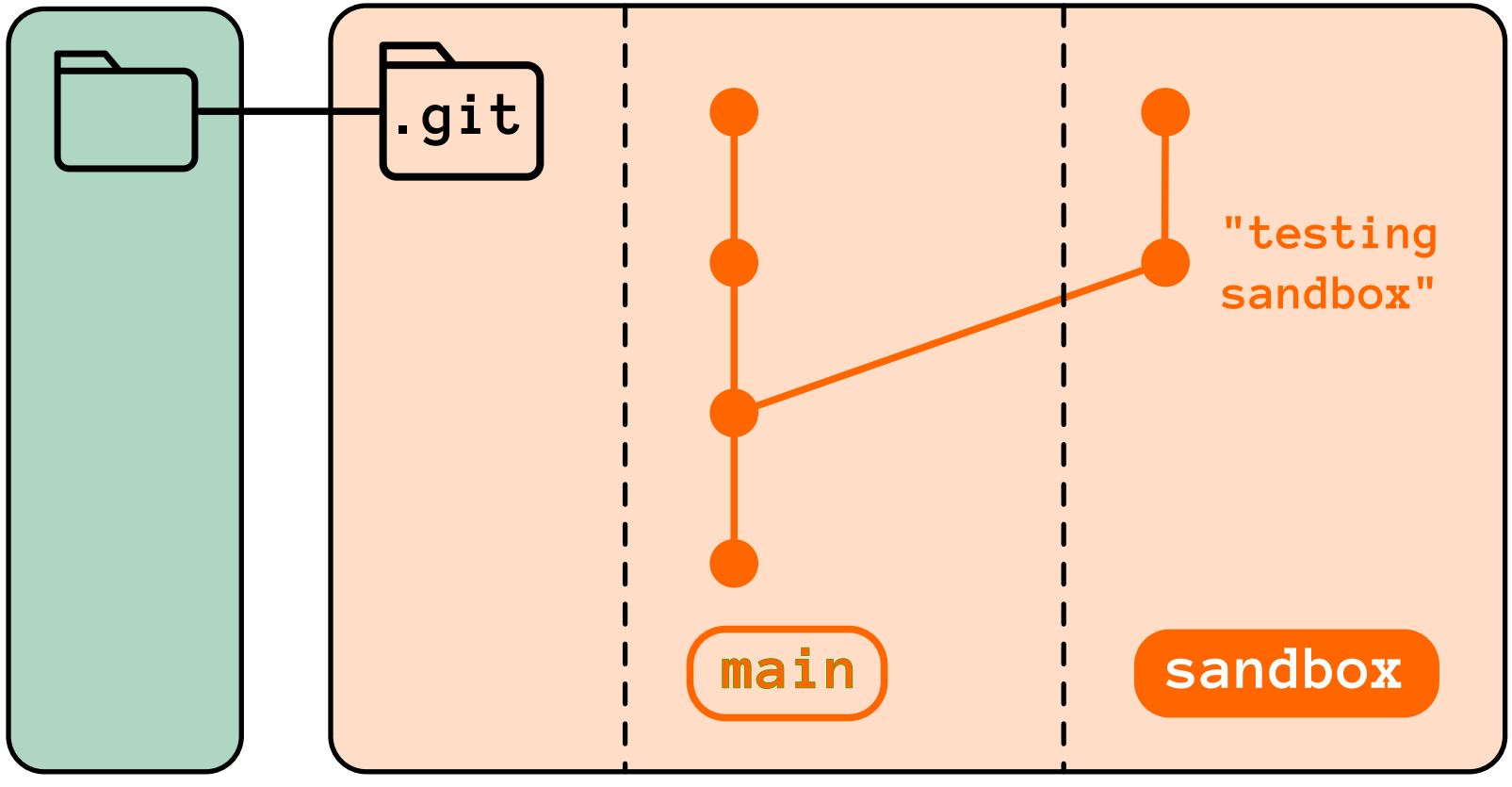
Commits are like save points,
that store versions of your files

`git commit`



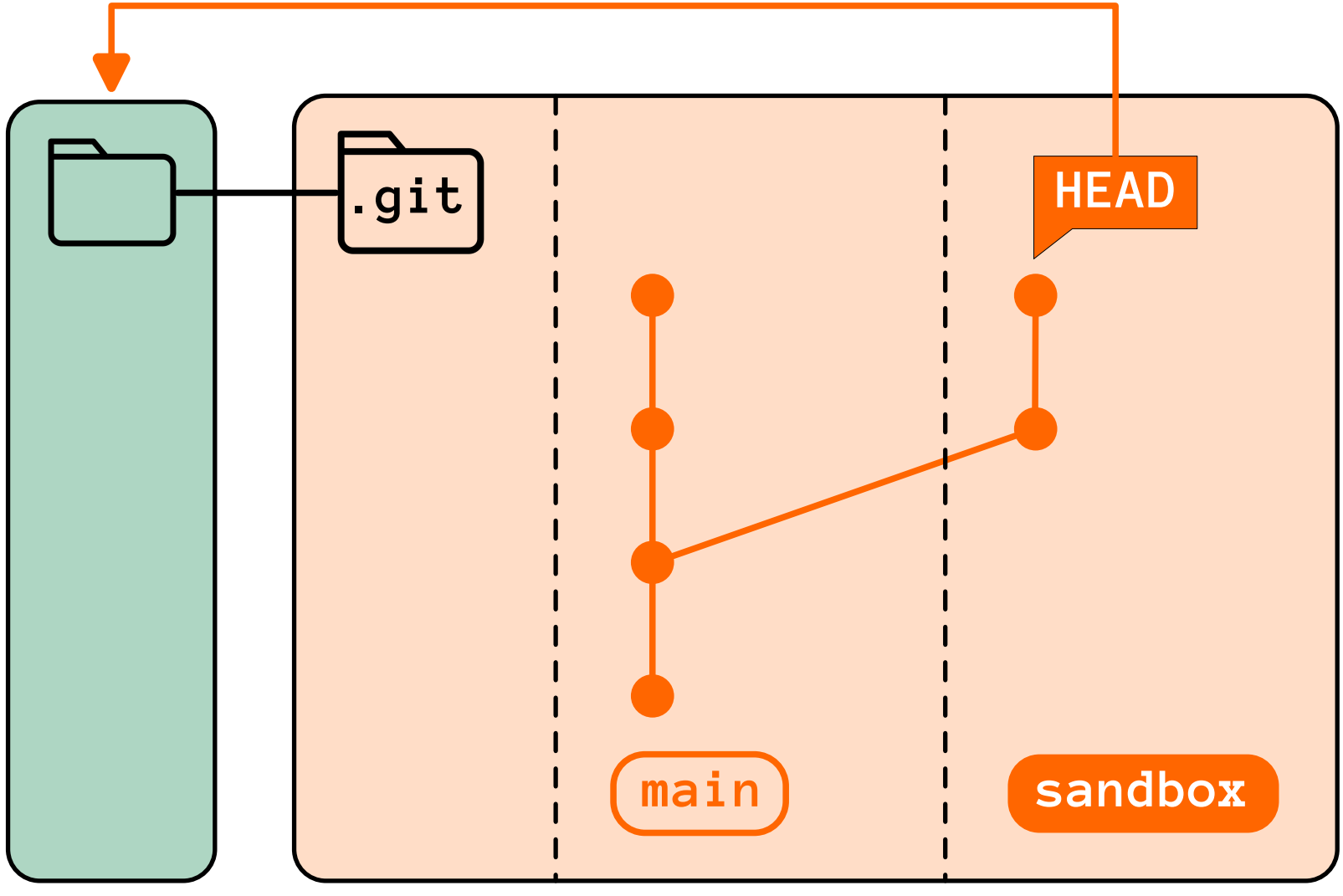
A **Branch** is an alternate version of your files which you can work on separately from the originals

`git branch`



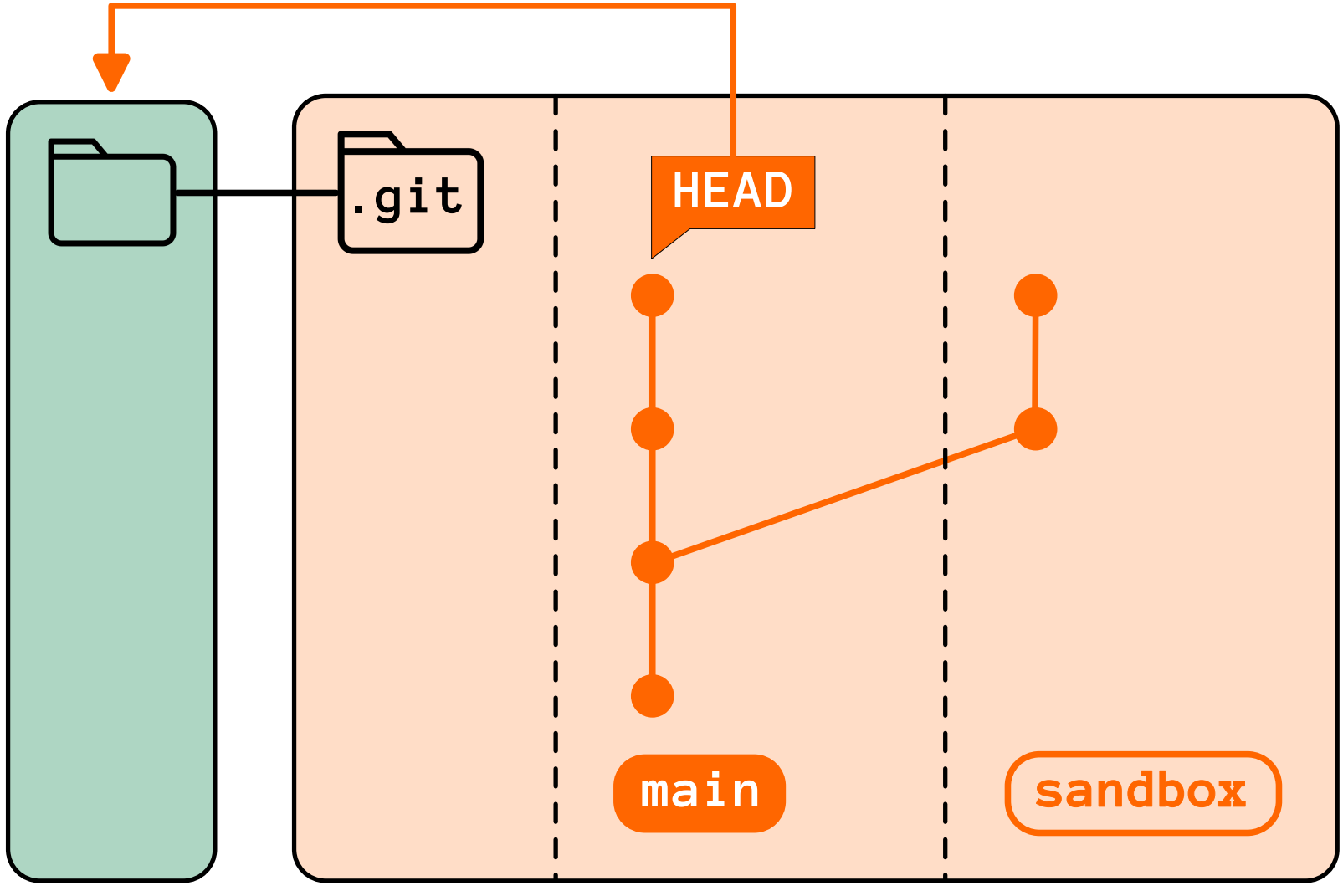
Checkout is used to switch between branches

`git checkout`



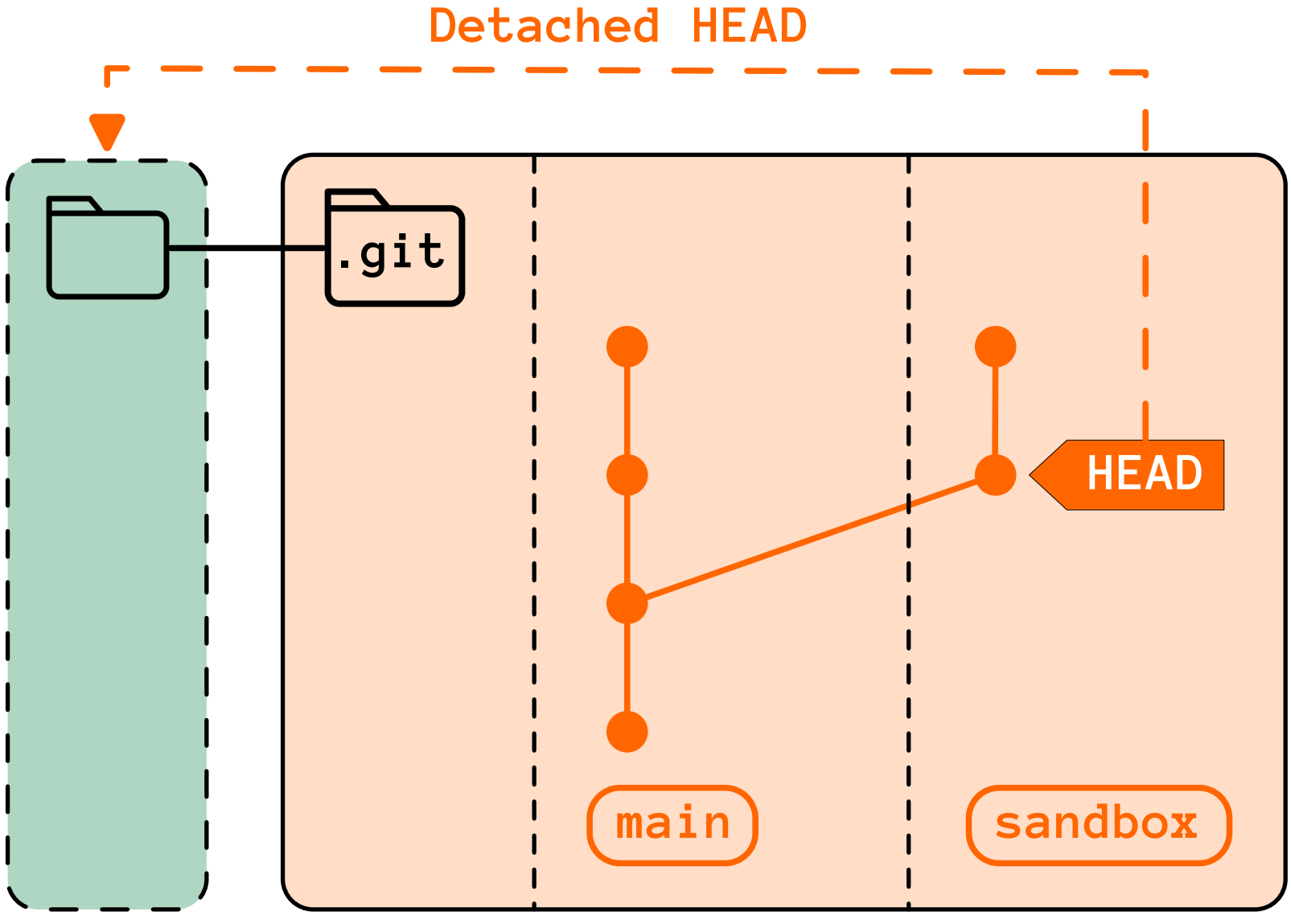
Checkout is used to switch between branches

`git checkout`



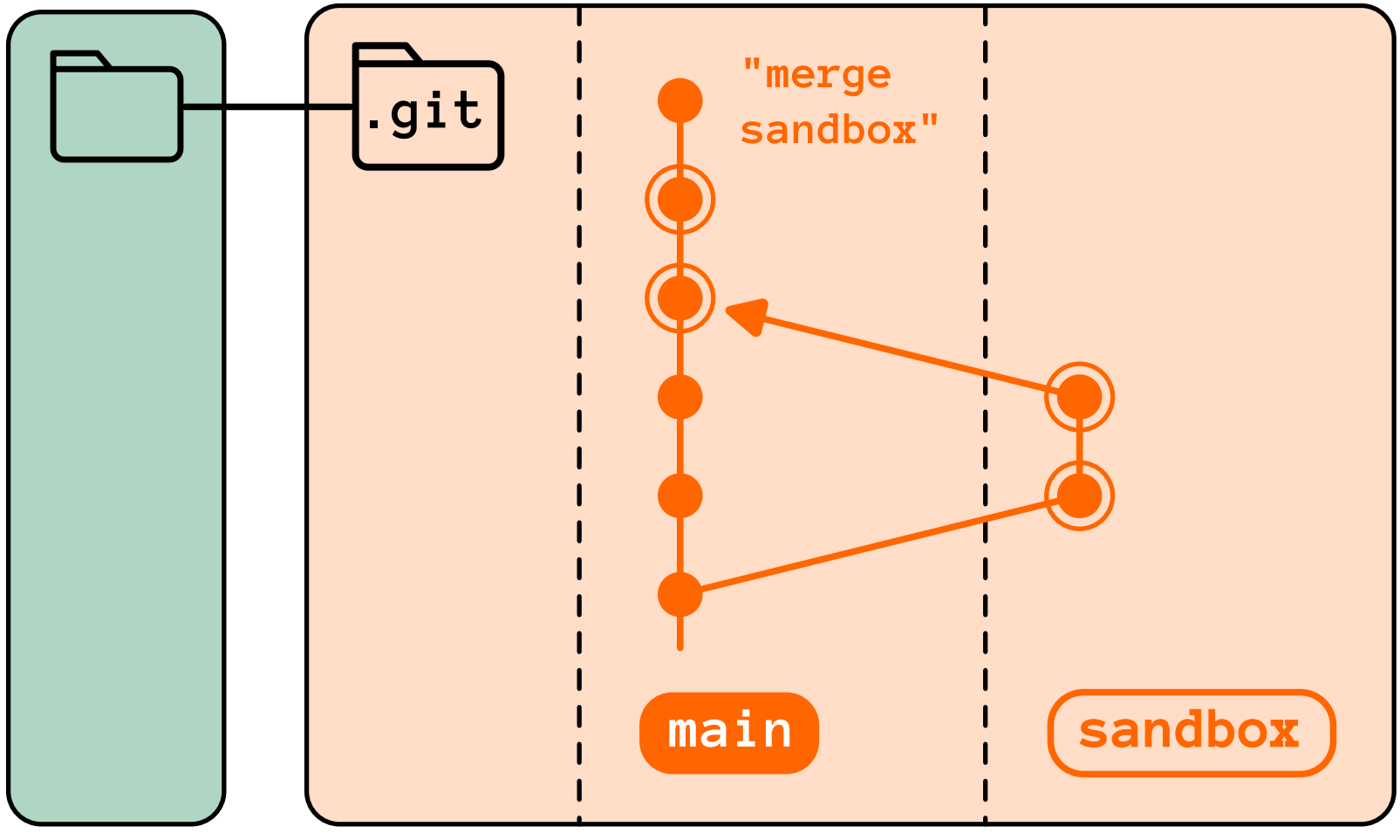
If you checkout a previous commit
in the history, it will "detach" the **HEAD**

git checkout



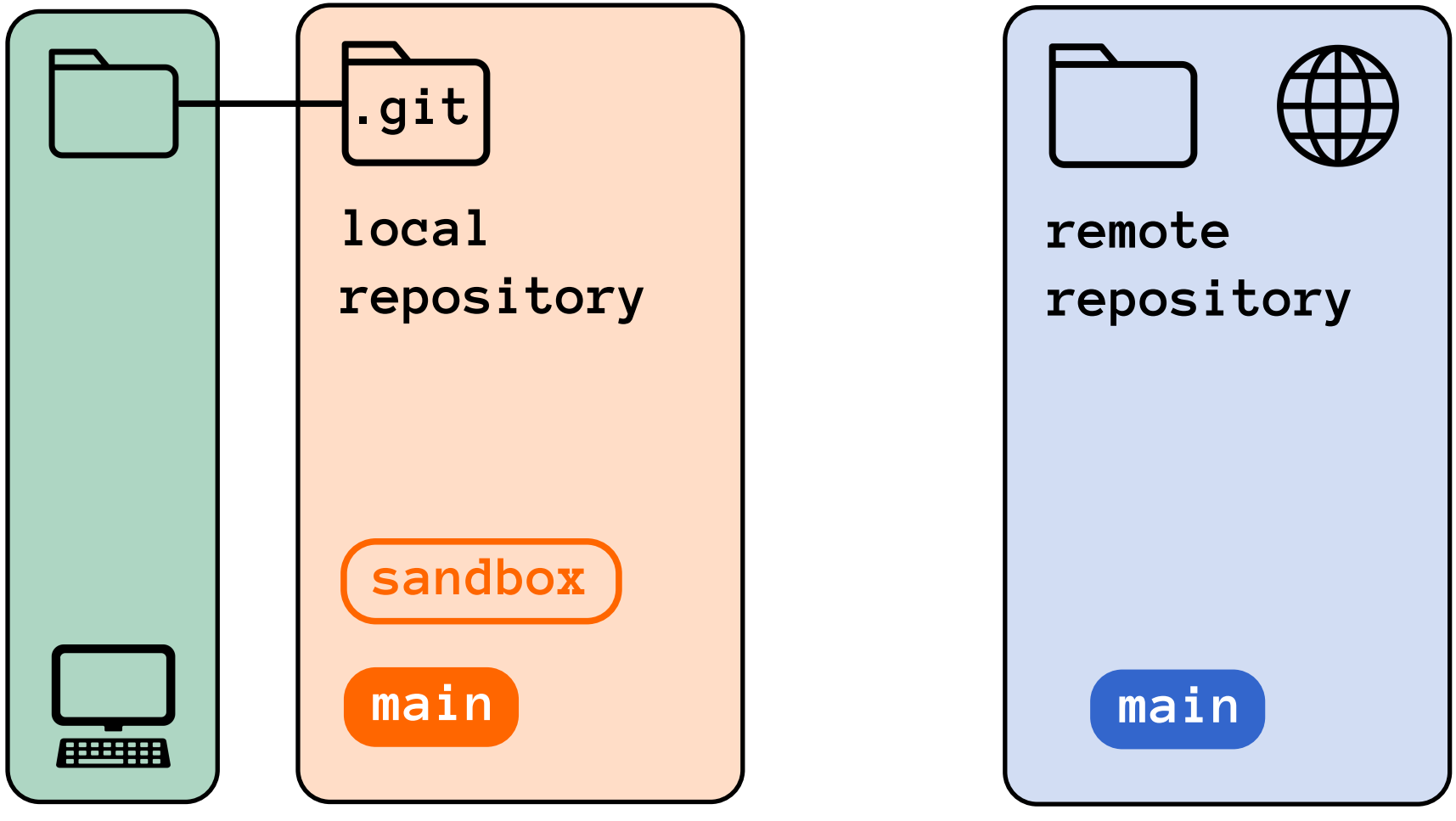
Merge will bring changes from another branch into the current branch

`git merge`



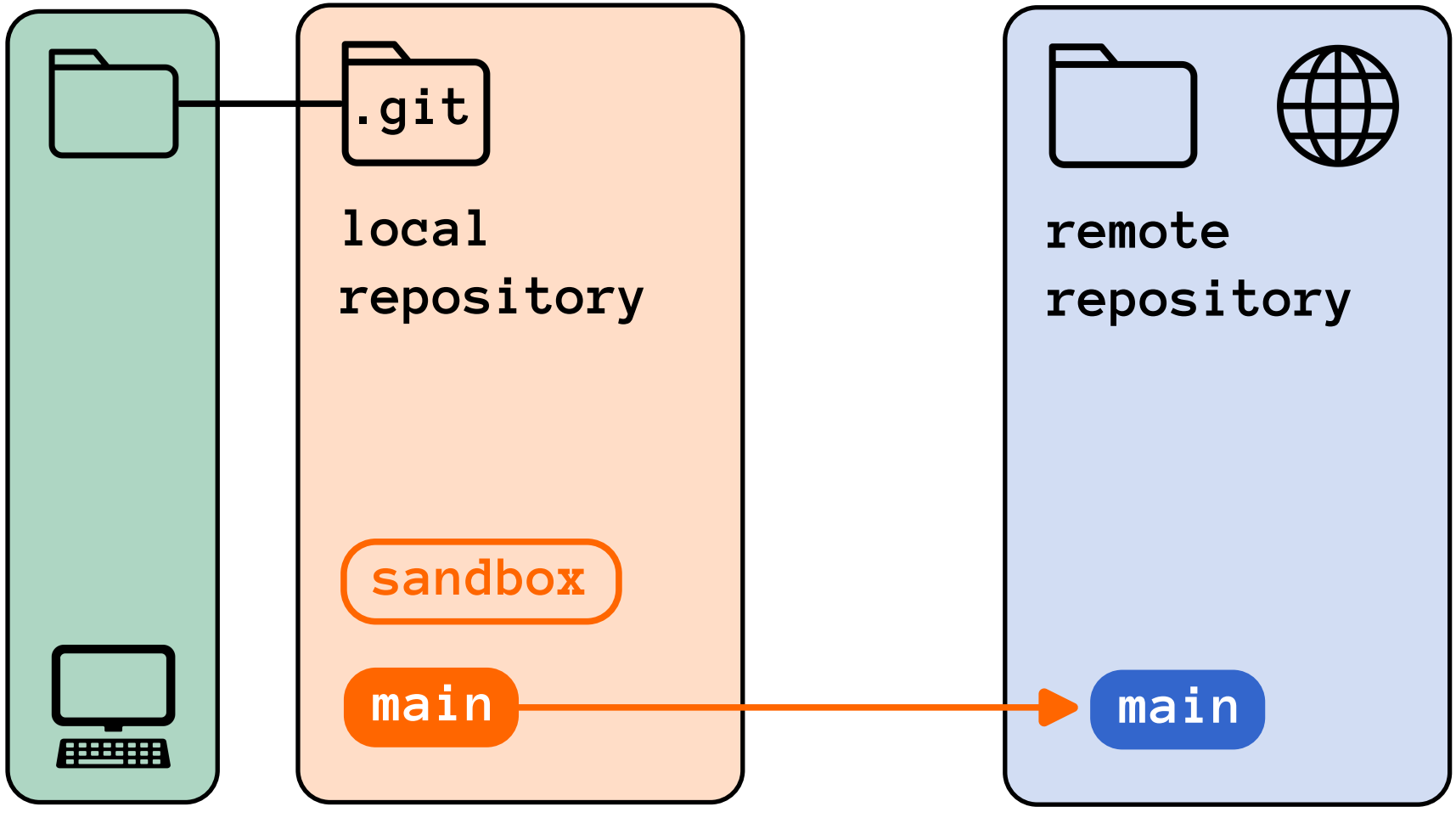
A **Remote** is a copy of the repository which is somewhere else, like on an online git hosting platform

`git remote`



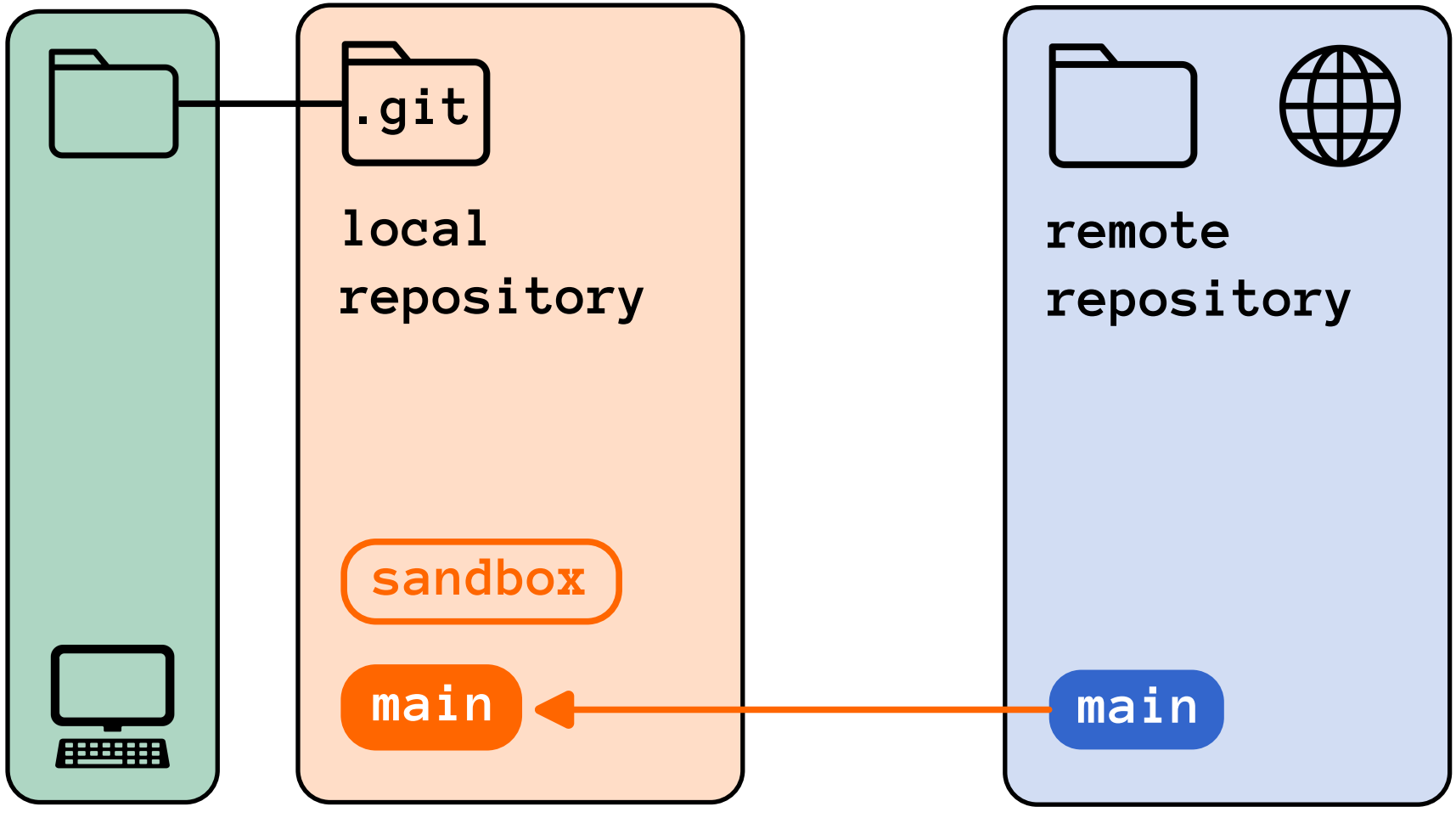
Push will copy new commits
from your **local** repository
to a **remote** repository
(usually) only for your current branch

`git push`



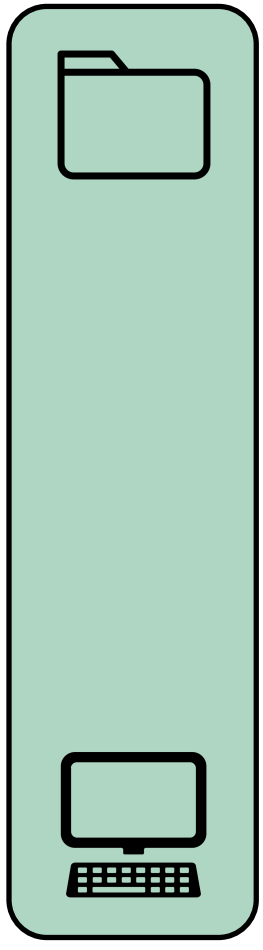
Pull will copy new commits
from the remote repository
to your local repository
(usually) only for your current branch

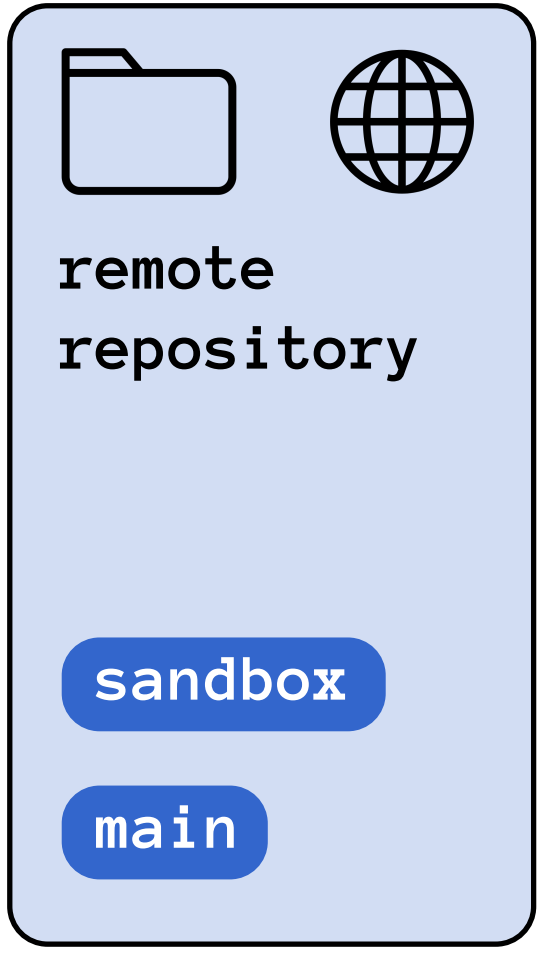
`git pull`



Clone will make a new local copy of a remote repository

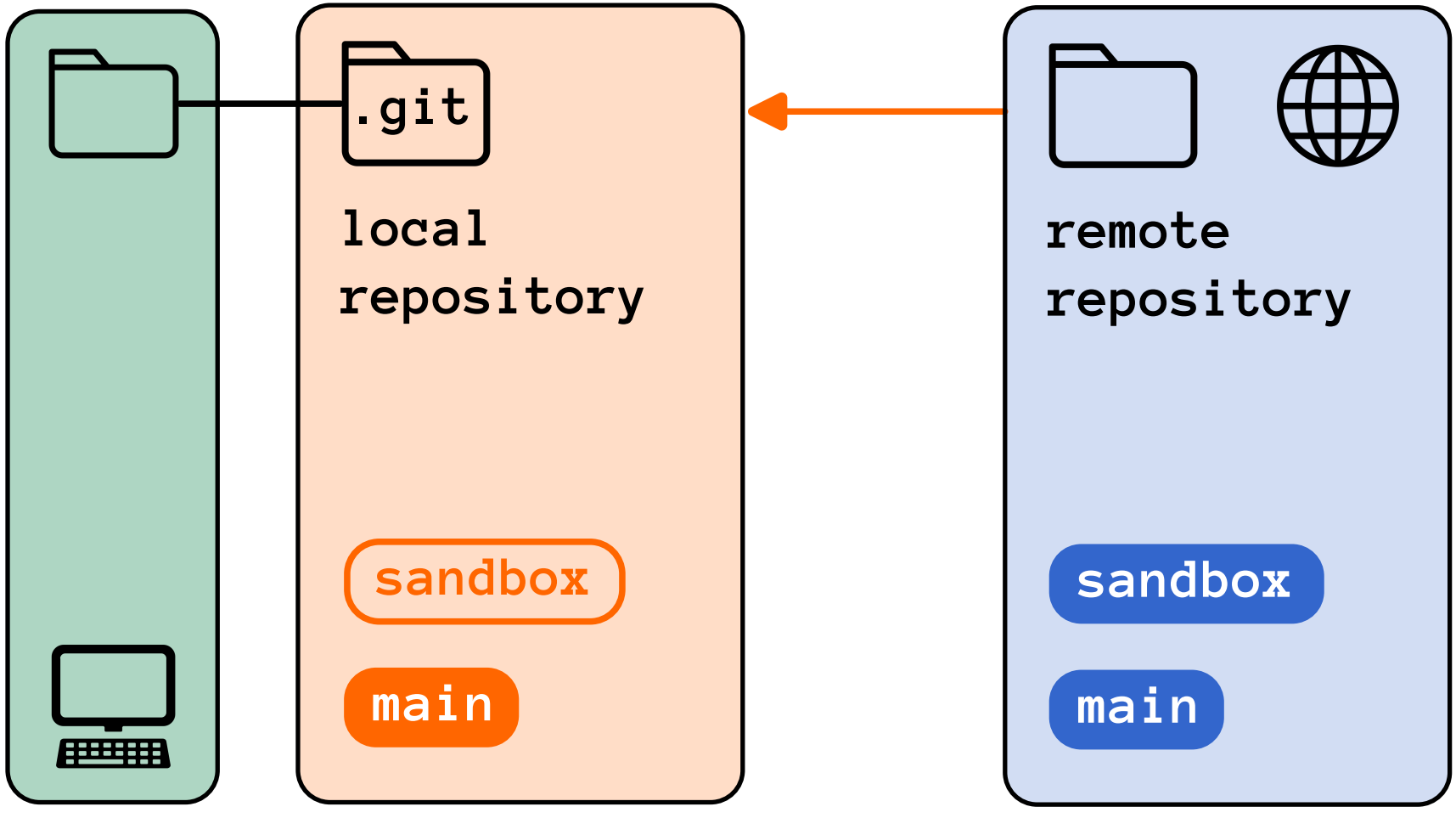
`git clone`





Clone will make a new local copy of a remote repository

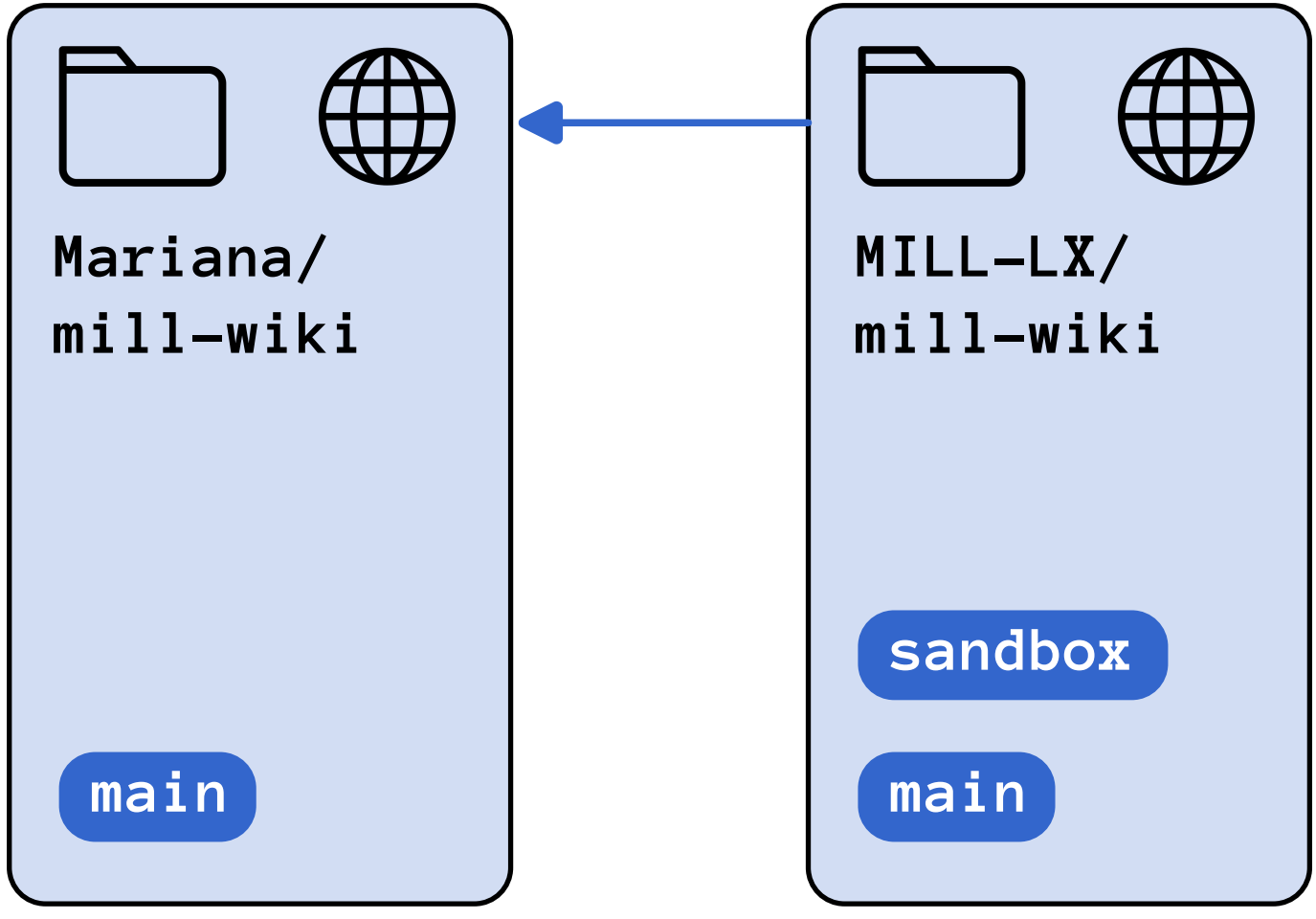
`git clone`



Fork will copy a remote repository
from someboy else's account into your account

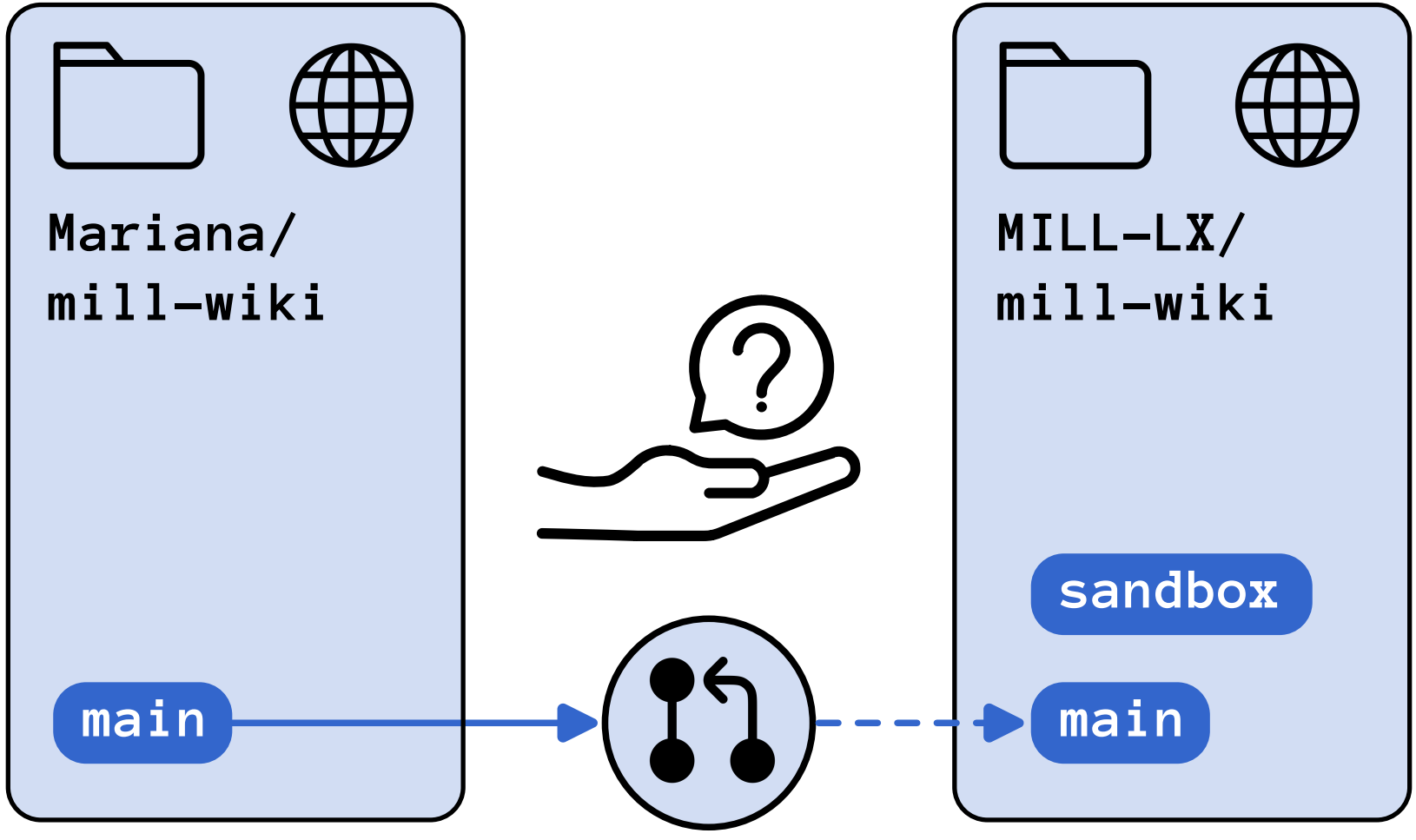
fork

You can choose to copy only the default branch



A **Pull Request** lets you propose changes from your remote repository to the original you created a fork from

pull request



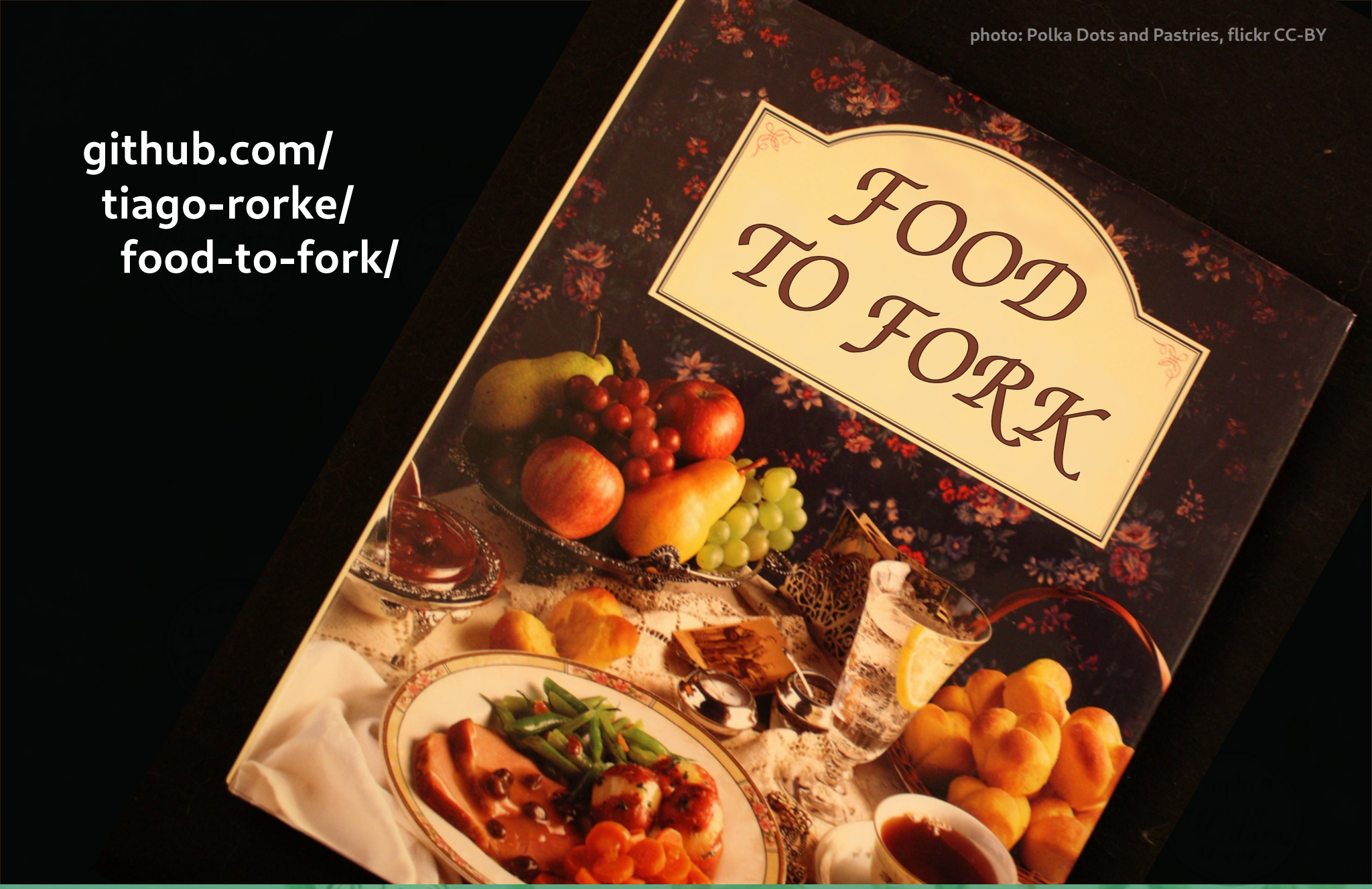


photo: Polka Dots and Pastries, flickr CC-BY

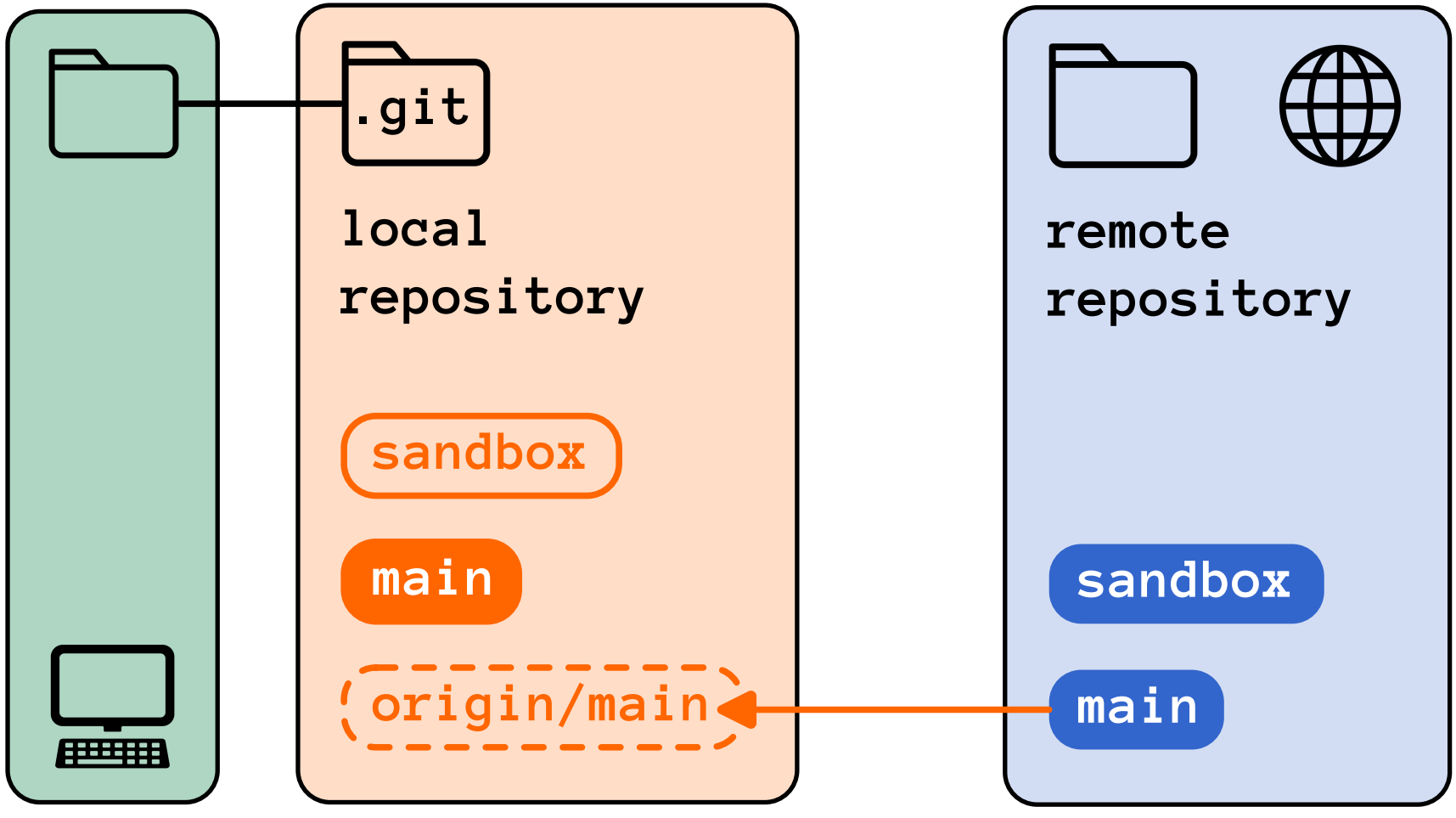
github.com/
tiago-rorke/
food-to-fork/

Try it out, by sharing a recipe

Pull actually runs two commands:

`git fetch`

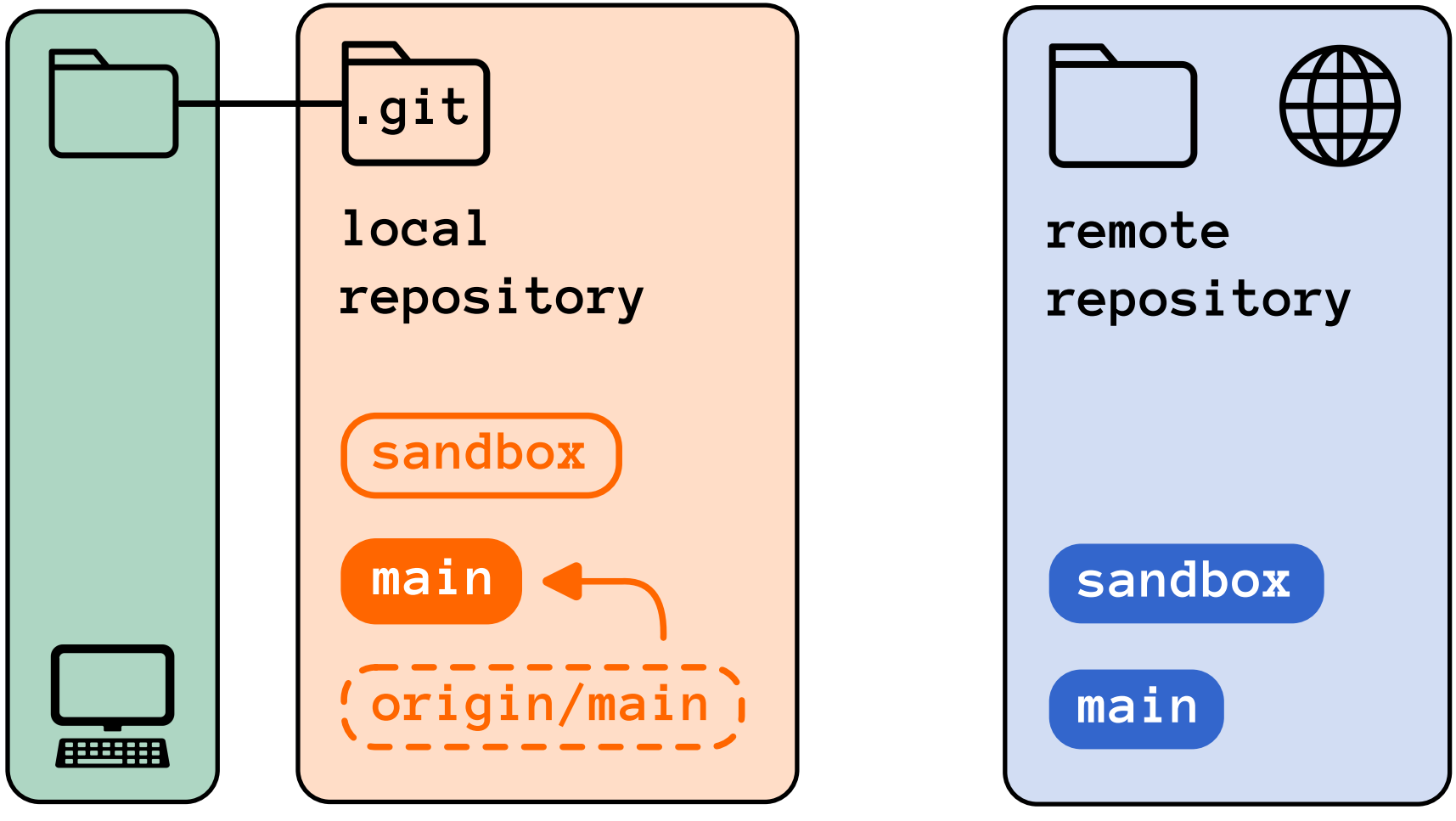
Fetch will first copy new commits
from the remote to a "remote-tracking branch"



Pull actually runs two commands:

`git merge`

Merge will then update your local branch with these new commits



GitHub Pages

Make your own recipe webpage