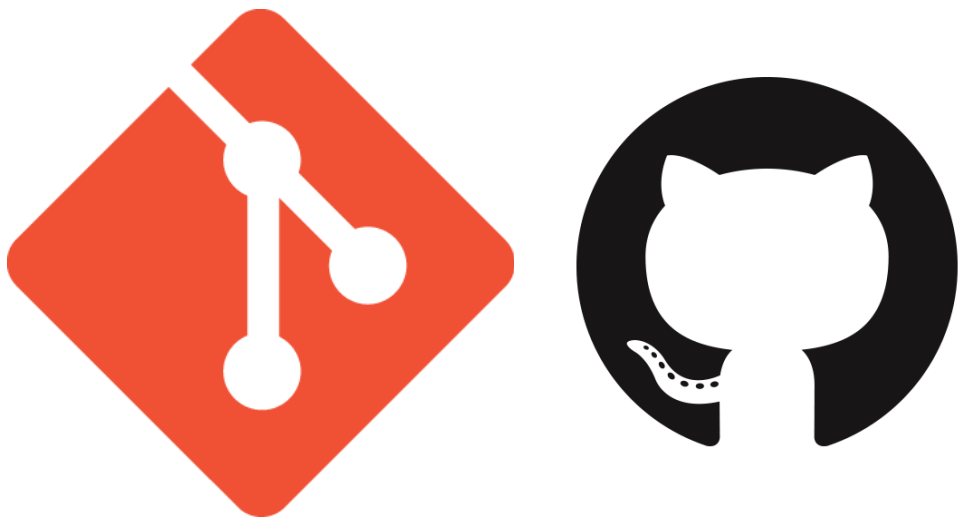
Version Control with Git & Collaborating with GitHub



Why does git exist?

- Git protects yourself and others from yourself and others
- You can modify/change/break/improve your code, secure in the knowledge that you can not ruin your work too badly
- **No** commercial software is written without Version Control!
- A ton of open-source projects are developed using git (pandas, scikit-learn, seaborn, ggplot, $\cdots$)
- You will be able to replicate old results, *always*

How does git work?

- $\longrightarrow$ manages changes to a project without overwriting any part of it
- You tell git which files to keep track of in a particular folder
- Eventually to make snapshots of your folder

Step 0: Get git

- Install git from https://git-scm.com/download
- Install GitHub Desktop (a GUI) from https://git-scm.com/downloads/guis

Step 1: Initalize a repository

- ! Choose a normal folder that shall host your project
- repository = directory git watches
- `git init` in the console, or
- ... click "Add local repository" in GitHub Desktop

Intermediate: Checking the status

- ! Create some text files (e.g. Python scripts) with some content in the directory
- `git status`
- What do you see?

Step 2: The staging area

- `git add <filename>`
- Now git includes the changes from this file
- git does not move the file; git just changes the way how it keeps track

Step 3: Creating save points

- commit = a save point/check point
- `git commit -m "<Short summary of what you did>"`
- Everything what used to be in the staging area is now in the history (the local repository)

Intermediate: Inspecting the repo

- ! Make changes to the file you just committed
- `git status`
- What do you see?

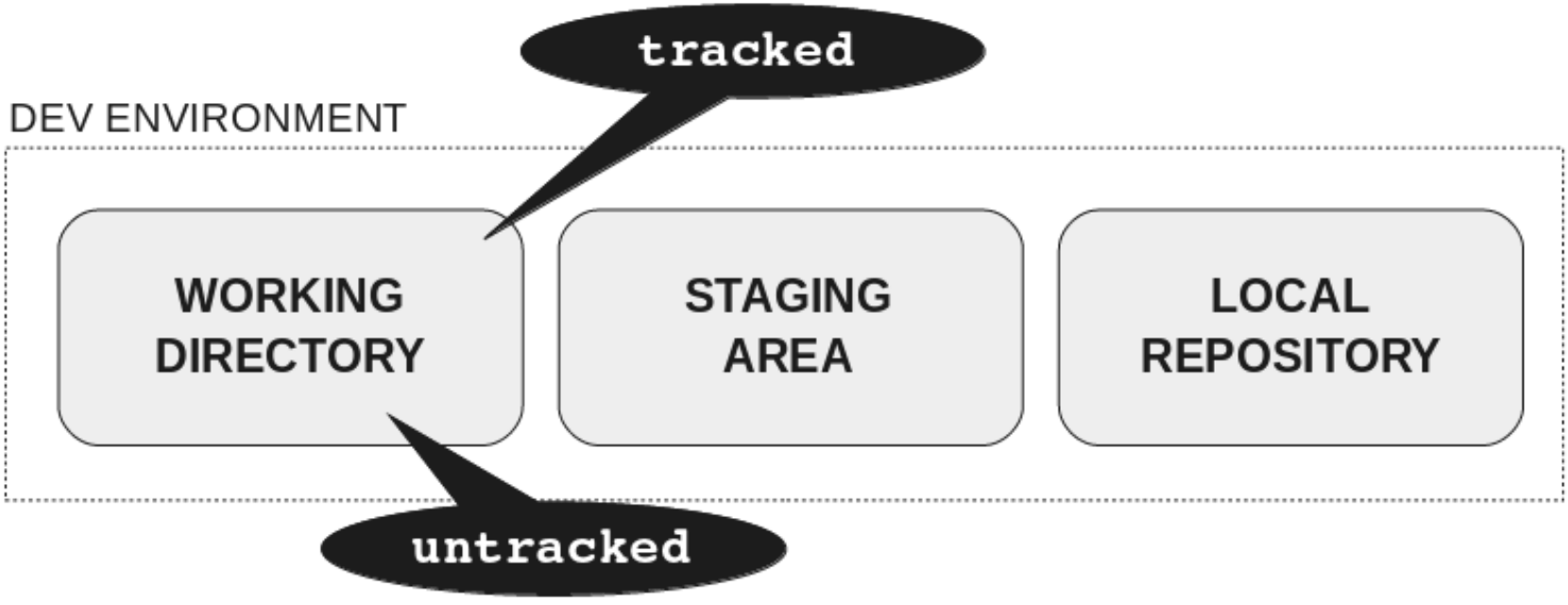
Intermediate: Inspecting the repo

- ! Make changes to the file you just committed
- `git status`
- What do you see?
- `git diff <filename>`
- What do you see?

Intermediate: Inspecting the repo

- ! Make changes to the file you just committed
- `git status`
- What do you see?
- `git diff <filename>`
- What do you see?
- `git log`
- What do you see?

Summary of git's architecture



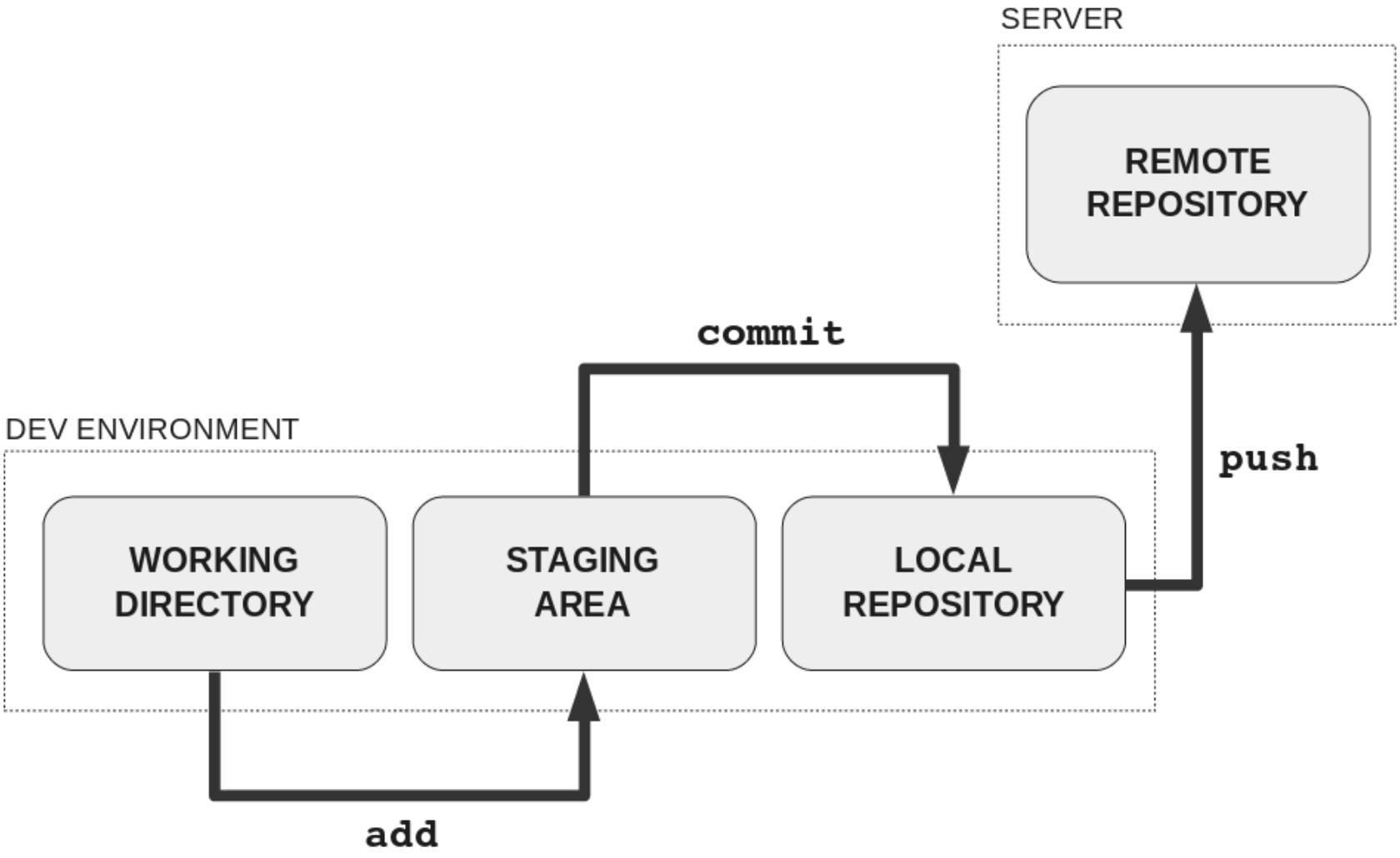
from: Rachel Carmena (2018): https://rachelcarmena.github.io/2018/12/12/how-to-teach-git.html"How to teach Git"

From git to GitHub

- git: Version control *on your machine*
- GitHub: Cloud storage accessible from git

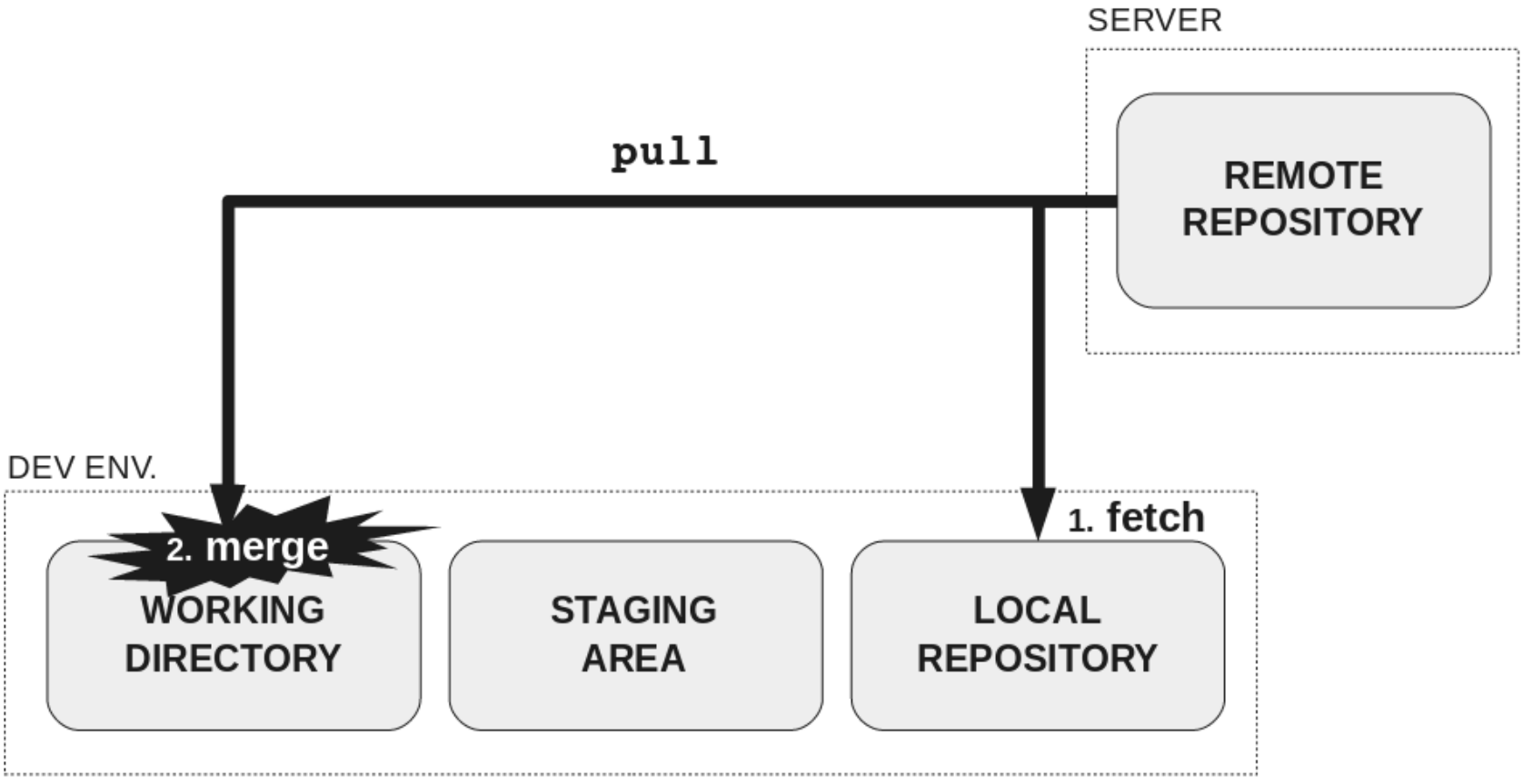
Create an account on GitHub! (Also get free, unlimited private accounts using https://education.github.com/studentsGitHub Student Developer Pack)

How do your changes make it to GitHub?



from: Rachel Carmena (2018): https://rachelcarmena.github.io/2018/12/12/how-to-teach-git.html"How to teach Git"

How do others' changes make it to your computer?



from: Rachel Carmena (2018): https://rachelcarmena.github.io/2018/12/12/how-to-teach-git.html"How to teach Git"

If you want to become git master...

https://lab.github.com/:

- https://www.youtube.com/watch?time_continue=8&v=9S0p8YMQzsMVideo
- 15 courses covering
  - Markdown
  - GitHub pages
  - Continuous integration
  - Pull Requests
  - GitHub Apps
  - ...