

Version control and project environments

ECP 8506 - Clinical Trial Simulations

Mutaz Jaber, PharmD, PhD

Email: jaber038@umn.edu

"FINAL".doc



FINAL.doc!



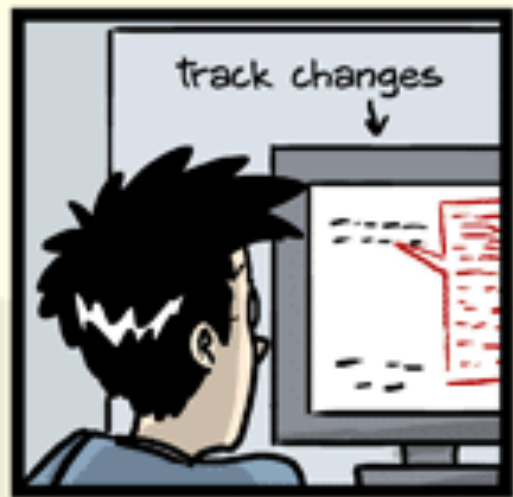
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRADSCHOOL?????.doc



**How do you record the history of
your projects?**

What is version control and why should we care?

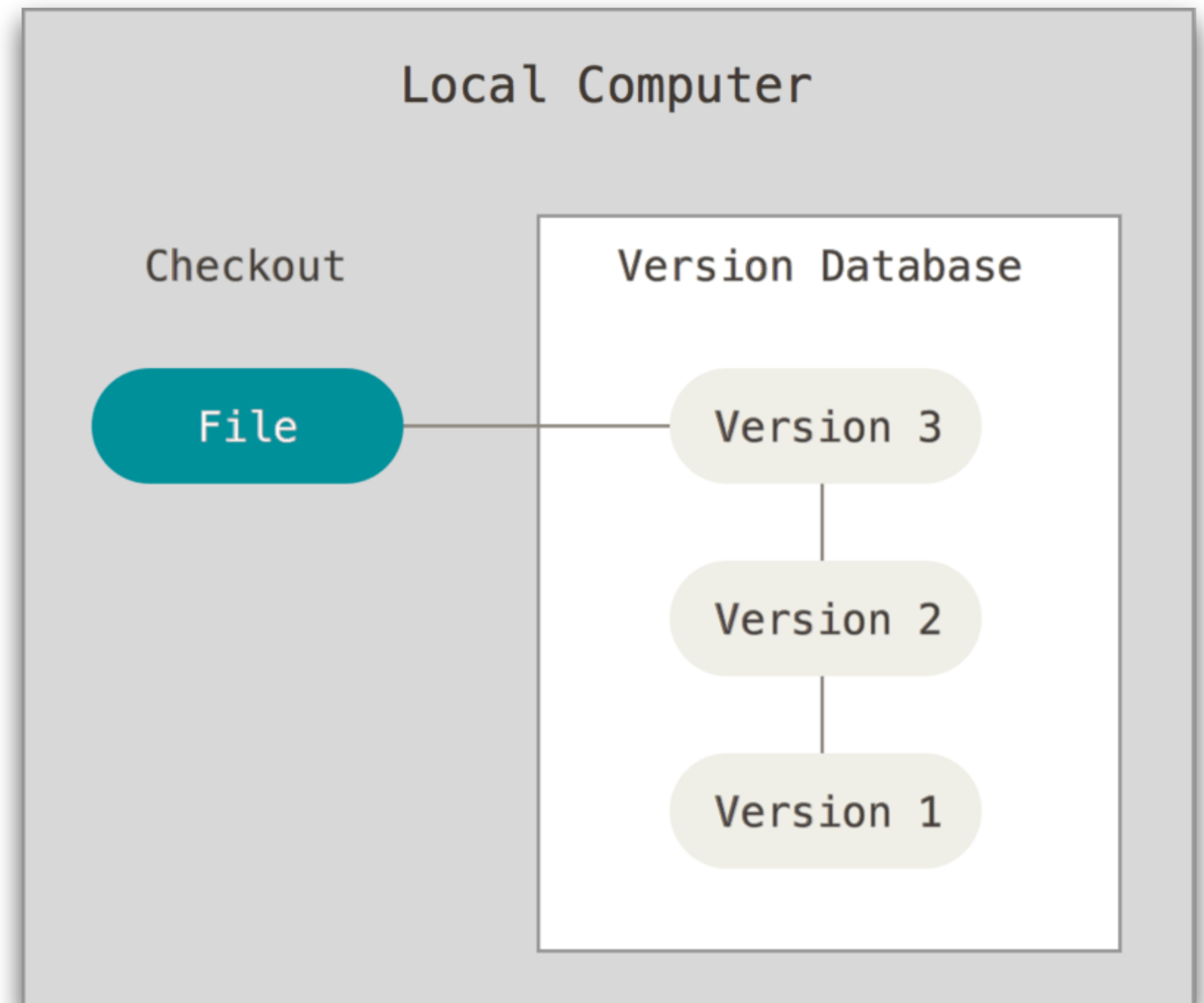
Version control is a system that records **changes** to a file or set of files over time so that you can recall specific versions later.

Version control systems start with a base version of the document and then save just the changes you made at each step of the way.

One can review project history to find out:

- *Which* changes were made?
- *Who* made the changes?
- *When* were the changes made?
- *Why* were changes needed?

Different types of version control system out there and we will focus on **Git** utility



Git version control system

The most common version control system.



The screenshot shows the Git website homepage. At the top left is the Git logo (a red diamond with a white branching diagram) followed by the text "git --local-branching-on-the-cheap". To the right is a search bar with the placeholder text "Type / to search entire site...". Below the header, there is a navigation menu on the left with links: "About", "Documentation" (highlighted in red), "Reference", "Book", "Videos", and "External Links". To the right of the navigation menu, there is a status bar showing "Version 2.46.0" and "git last updated in 2.46.0", along with links for "Topics" and "English". The main content area features the heading "NAME" in red, followed by the text "git - the stupid content tracker". Below this is the heading "SYNOPSIS" in red, followed by a code block containing the command-line options for the git command.

git --local-branching-on-the-cheap

Type / to search entire site...

About

Documentation

Reference

Book

Videos

External Links

Downloads

Community

Version 2.46.0 git last updated in 2.46.0

Topics **English**

NAME

git - the stupid content tracker

SYNOPSIS

```
git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
    [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
    [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--no-lazy-
fetch]
    [--no-optional-locks] [--no-advice] [--bare] [--git-dir=<path>]
    [--work-tree=<path>] [--namespace=<name>] [--config-env=<name>=
<envvar>]
    <command> [<args>]
```

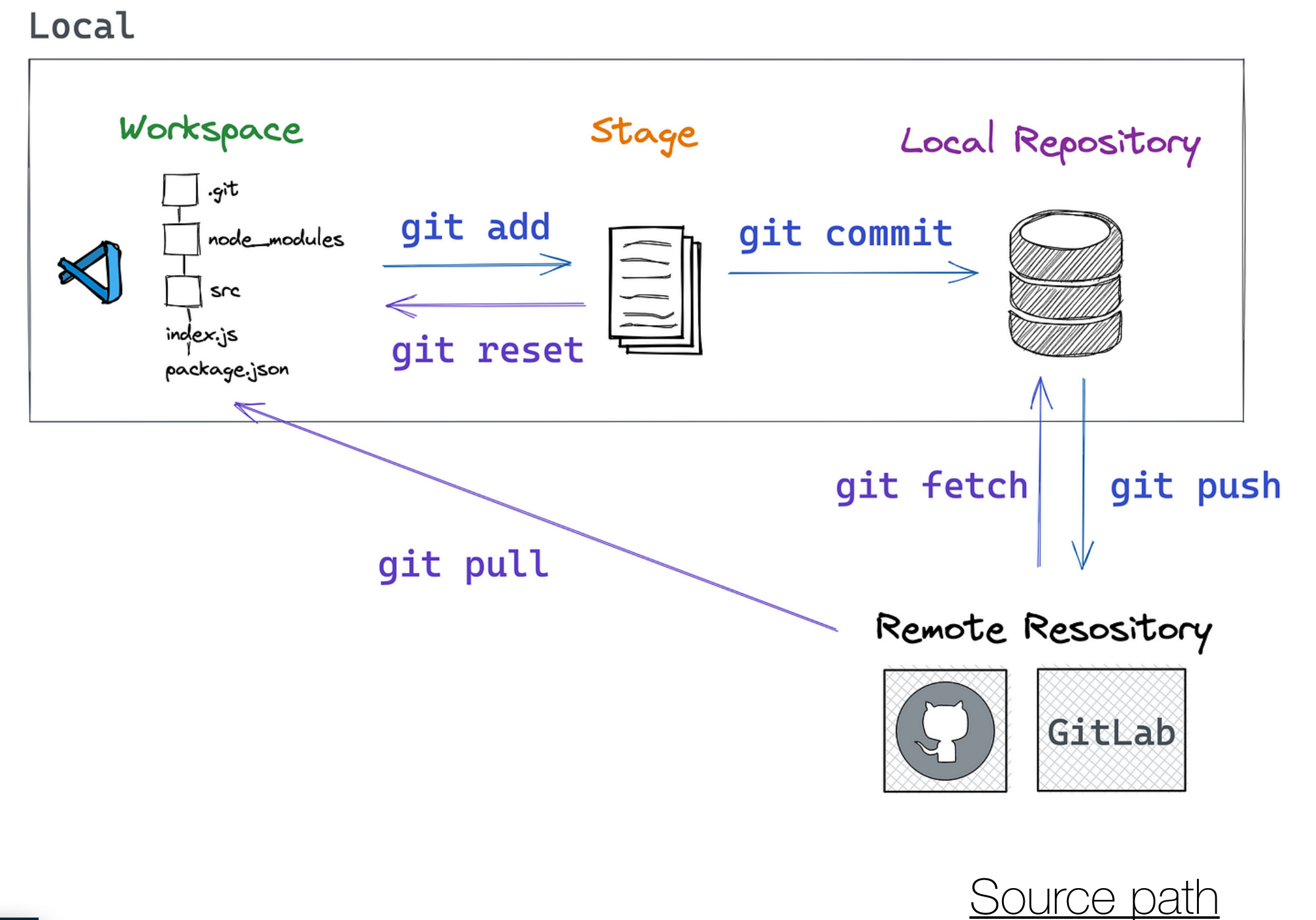
Git Repository

A Git repository is the `.git/` folder inside a project. This repository tracks all changes made to files in your project, building a history over time. Meaning, if you delete the `.git/` folder, then you delete your project's history.

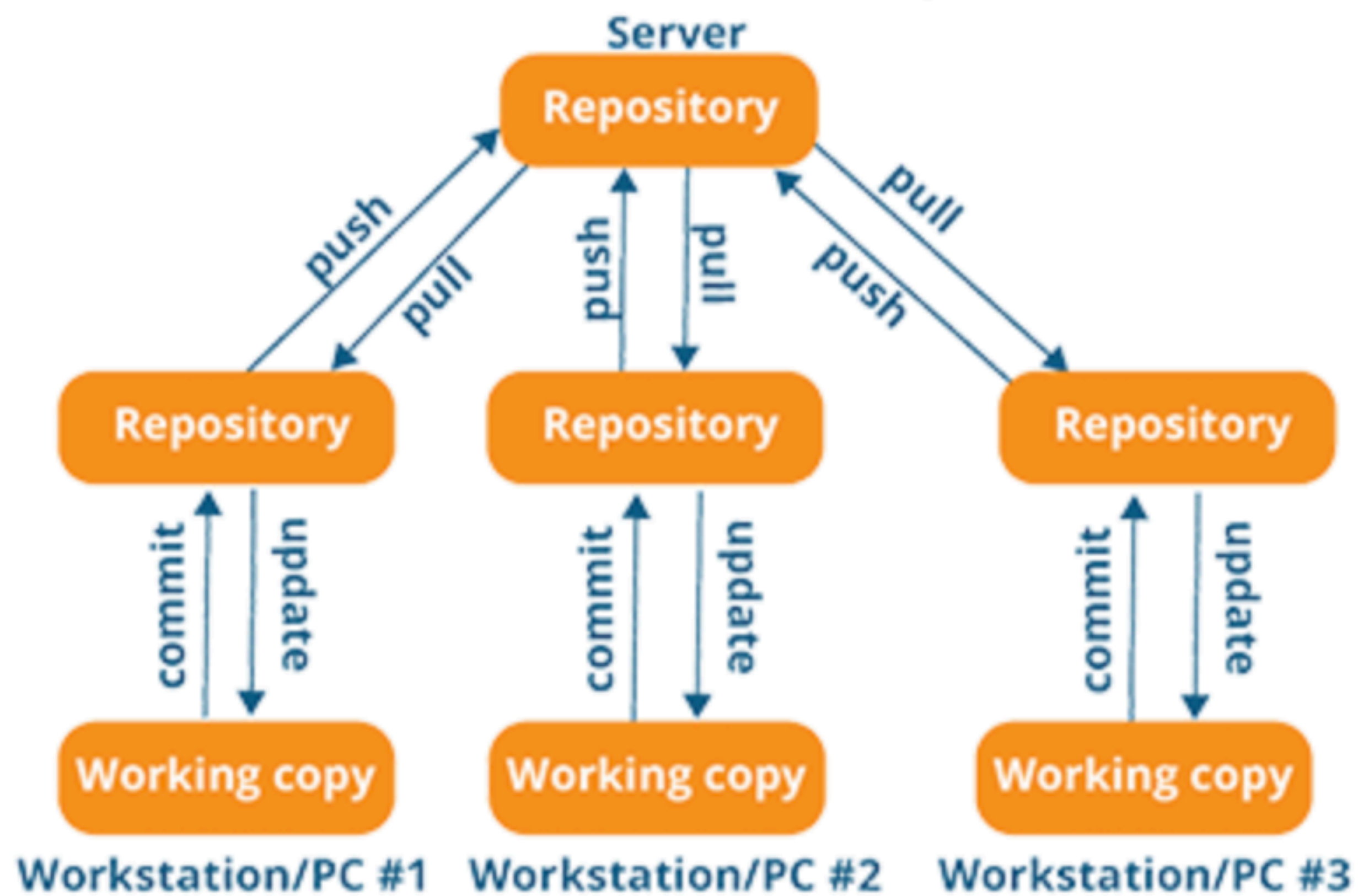
To initialize a repository (a.k.a repo) in your local device:

> `git init`

```
▶ (base) P7G4X9X3VM courses/ECP8506 » git init
Initialized empty Git repository in /Users/mjaber/courses/ECP8506/.git/
```



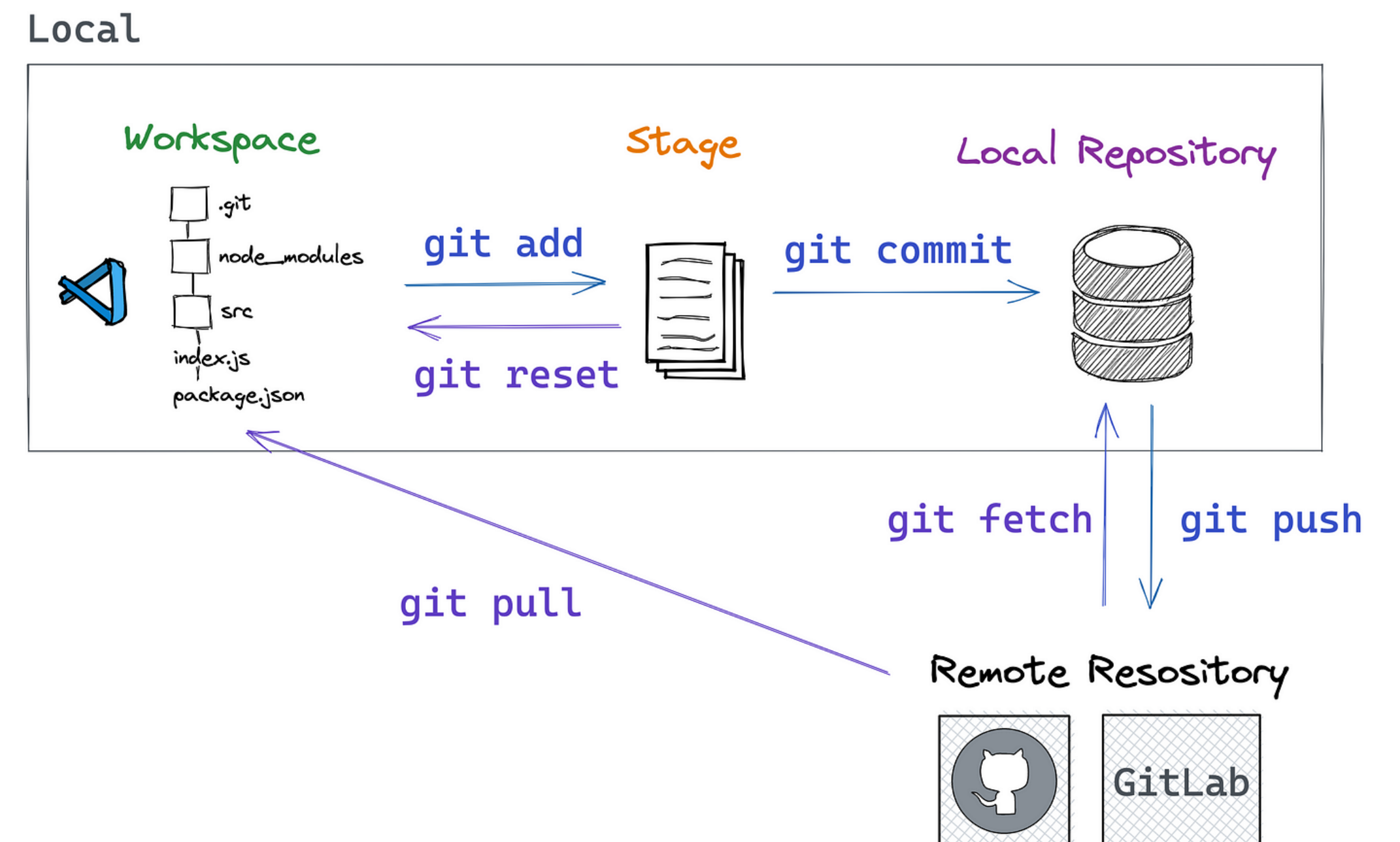
Distributed version control system



GitHub and others

Git Workflow

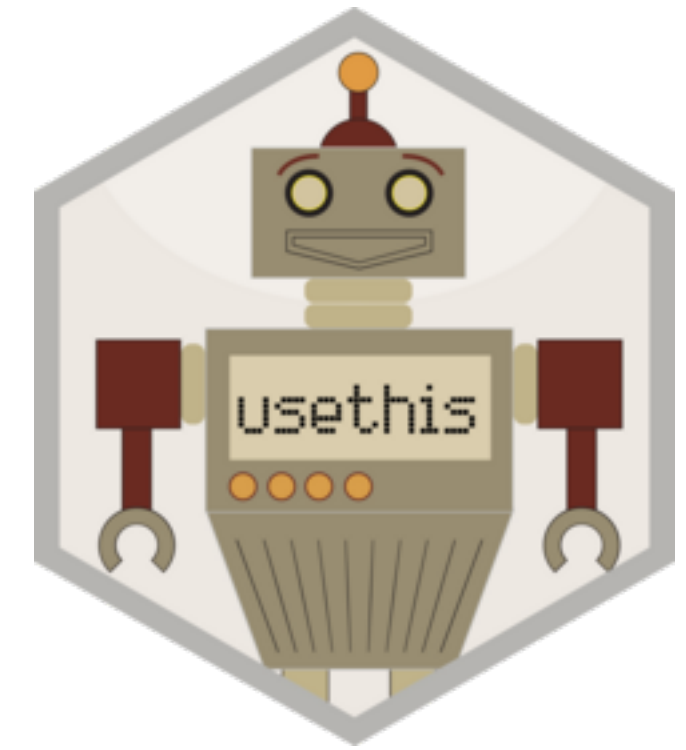
- Initializing a repository: **git init**
- Adding files to staging area: **git add**
- Creating a commit: **git commit -m "Commit message"**
- Checking status: **git status**
- Viewing commit history: **git log**



Connect Git/GitHub on MSI through RStudio

```
usethis::create_github_token(description=...)
```

```
gitcreds::creds_set()
```



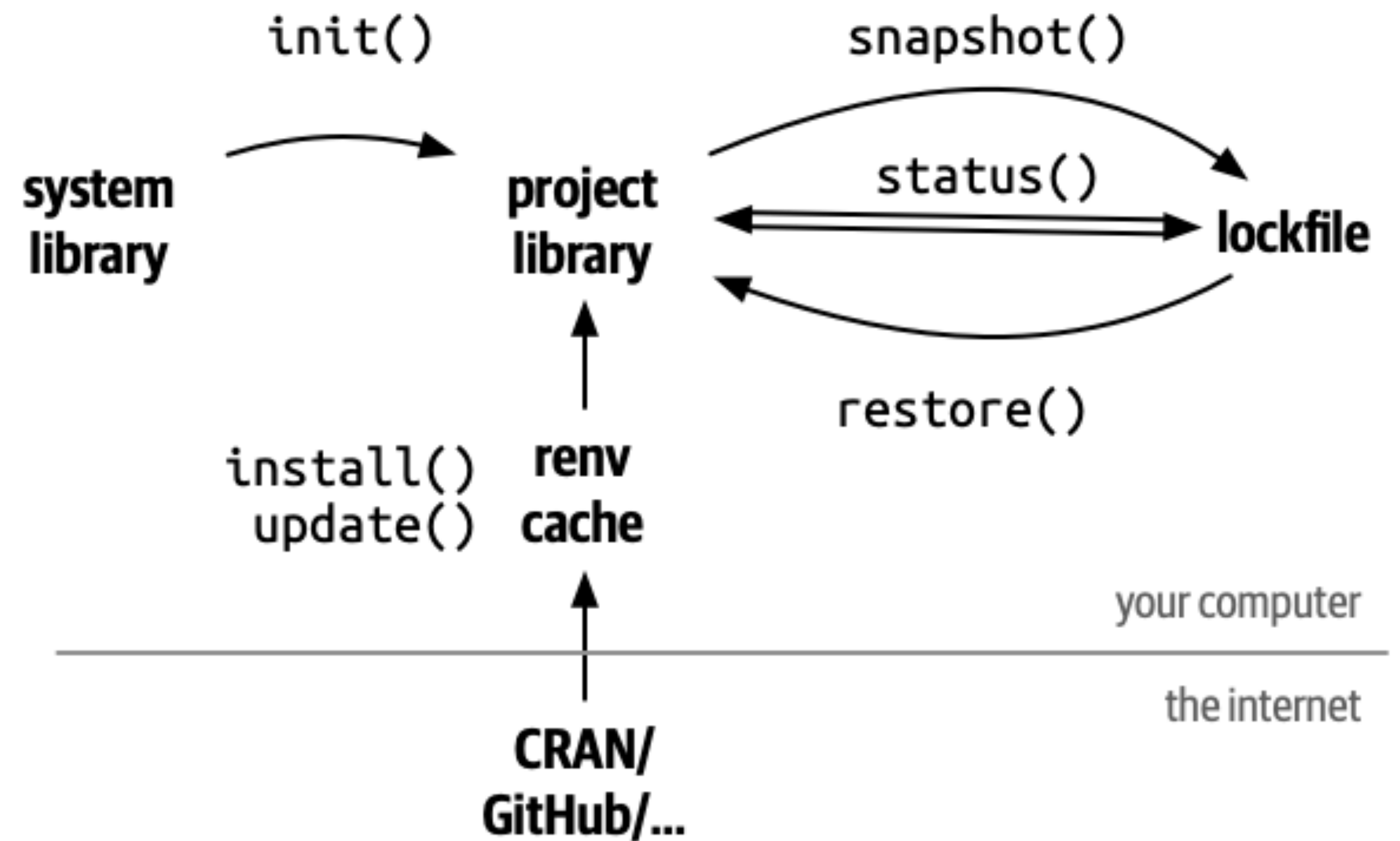
<https://usethis.r-lib.org/index.html>

<https://gitcreds.r-lib.org/index.html>

Reproducible environments using renv



“**renv** is an R package help create snapshots of your R projects”



<https://rstudio.github.io/renv/>

<https://rstudio.github.io/renv/articles/renv.html>

Common **renv** commands

```
renv::init()
```

```
renv::install()
```

```
renv::snapshot()
```

```
renv::restore()
```