A Brief Introduction to git + GitHub

credit: git

credit: GitHub

Adam A Miller

CIERA Summer REU + High School Research Program

27 June 2018

CIERA

Center for Interdisciplinary Exploration and Research in Astrophysics

ADLER PLANETARIUM

Important! These slides heavily borrow from a talk by Erik Tollerud

see https://github.com/LSSTC-DSFP/LSSTC-DSFP-Sessions/blob/master/Session3/Day1/Tollerud_repos_slides.pdf

What is Version Control?

From Wikipedia:

Revision control, also known as version control, source control or software configuration management (SCM), is the **management of changes to documents, programs, and other information stored as computer files.**

Why?

Peace of mind (backups)

Freedom (easy to explore with branches)

Collaboration (show & share your work)

Why?

Peace of mind (backups)

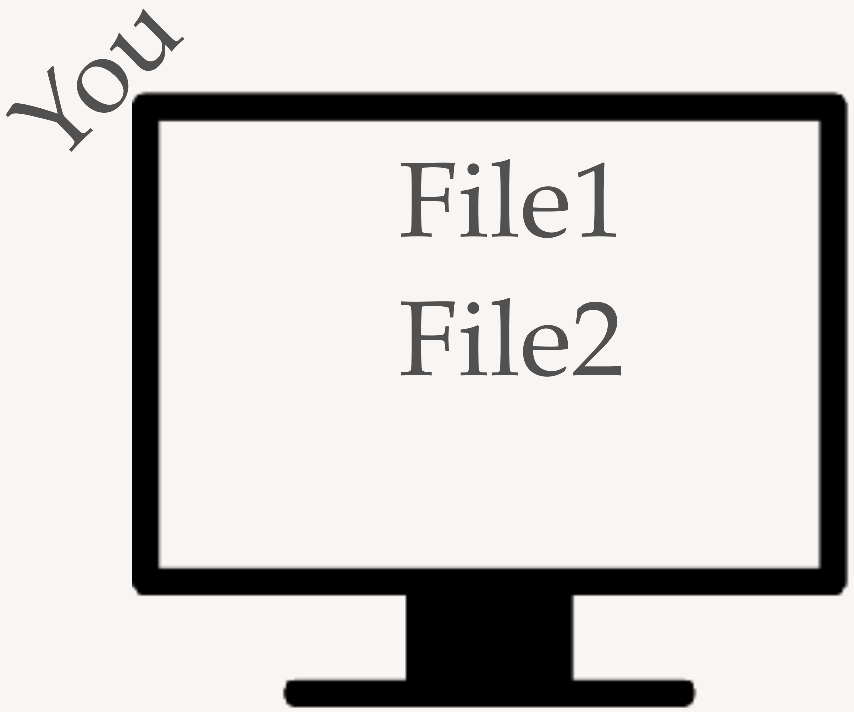
Freedom (easy to explore with branches)

Collaboration (show & share your work)

Also - you are your own worst enemy

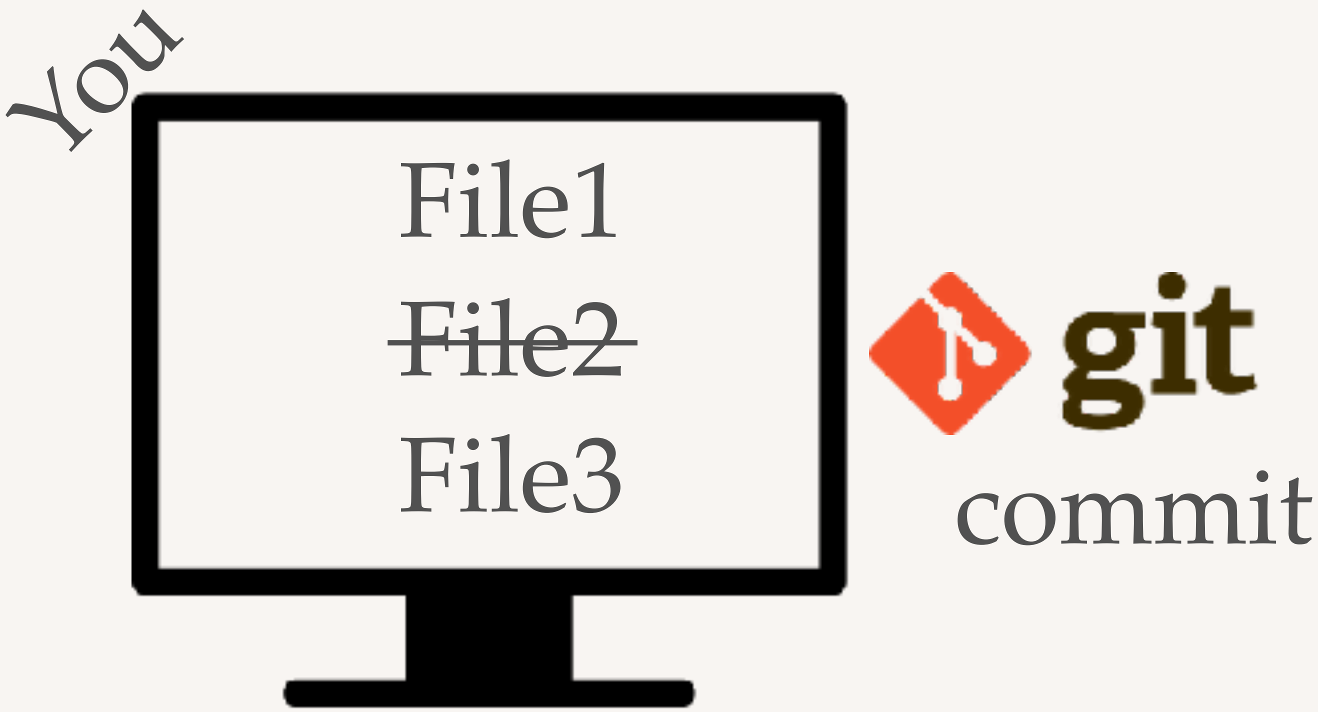
(but this is okay and part of the learning process)

Basic Idea Behind git + GitHub



credit: E. Tollerud

Basic Idea Behind git + GitHub



You

File1

~~File2~~

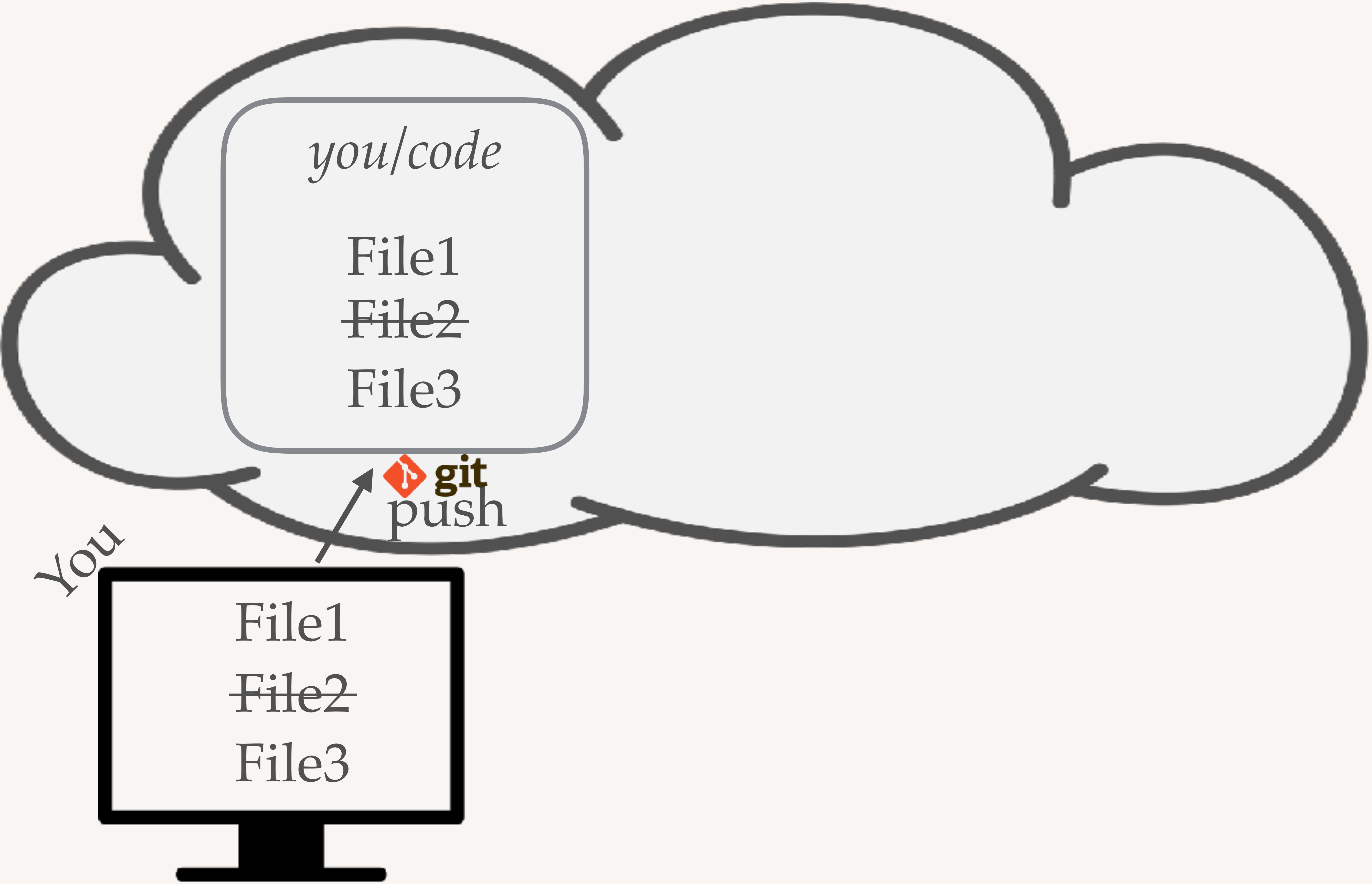
File3

git

commit

credit: E. Tollerud

Basic Idea Behind git + GitHub



you/code

File1

~~File2~~

File3

git push

You

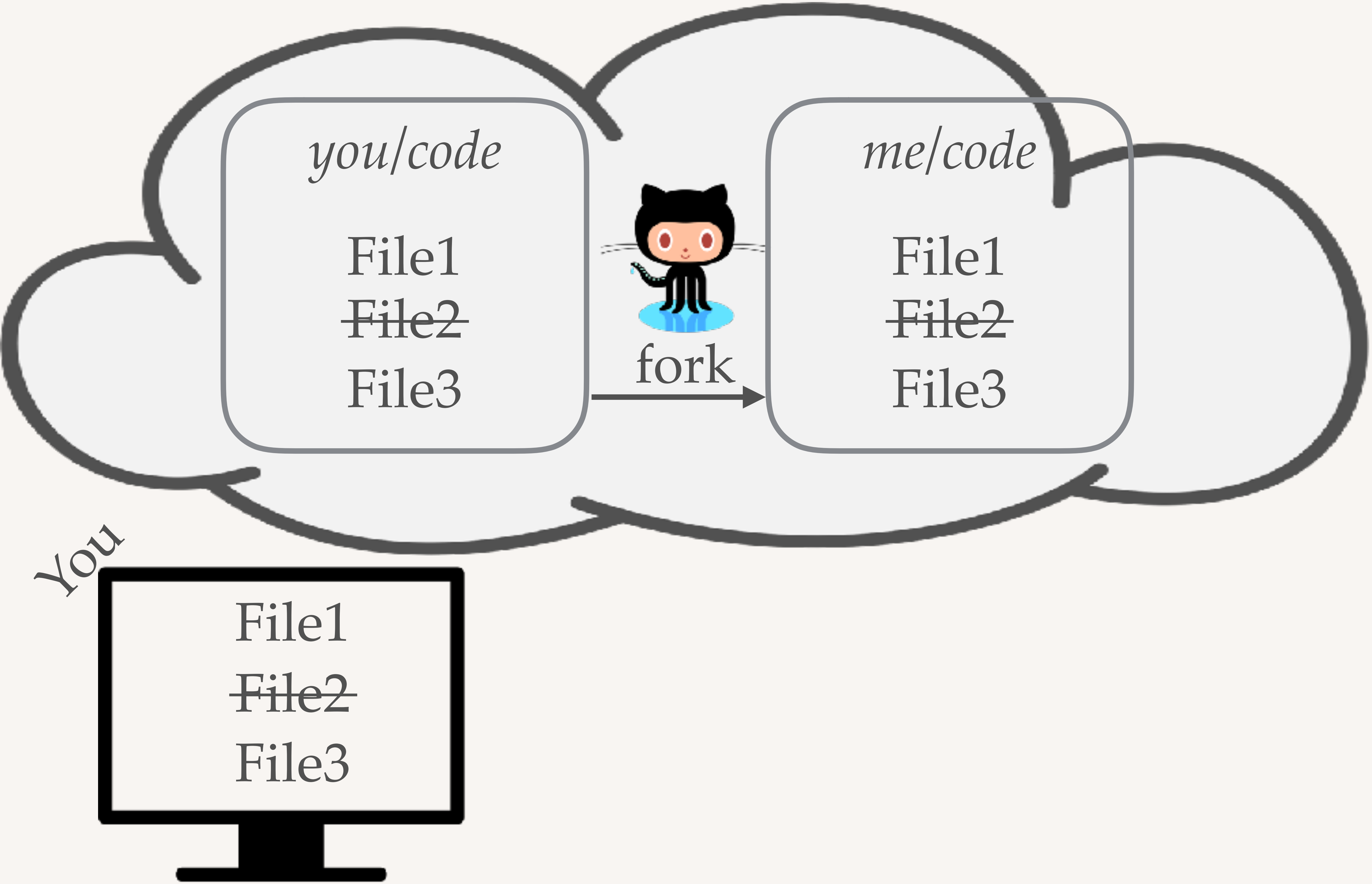
File1

~~File2~~

File3

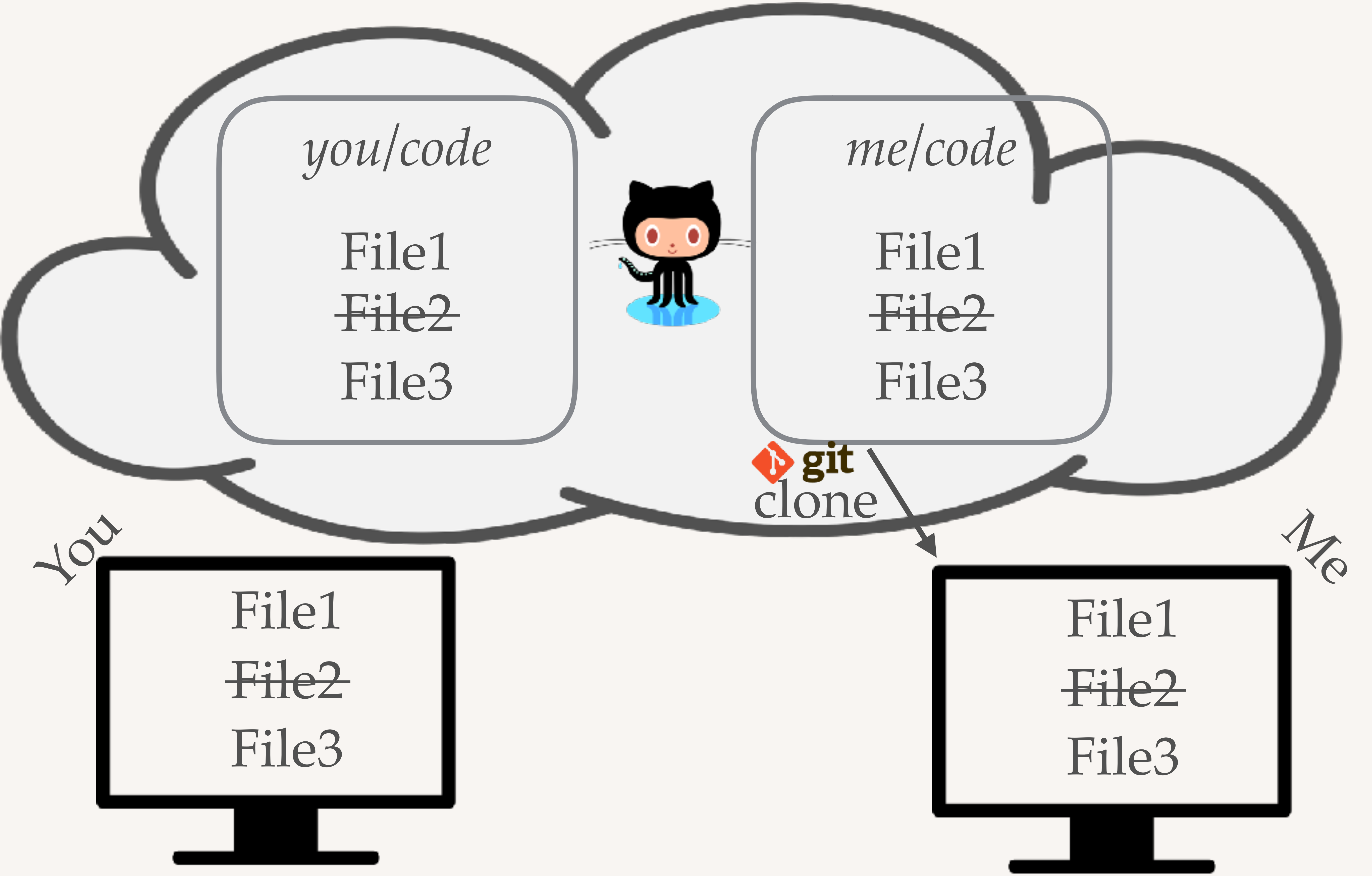
credit: E. Tollerud

Basic Idea Behind git + GitHub



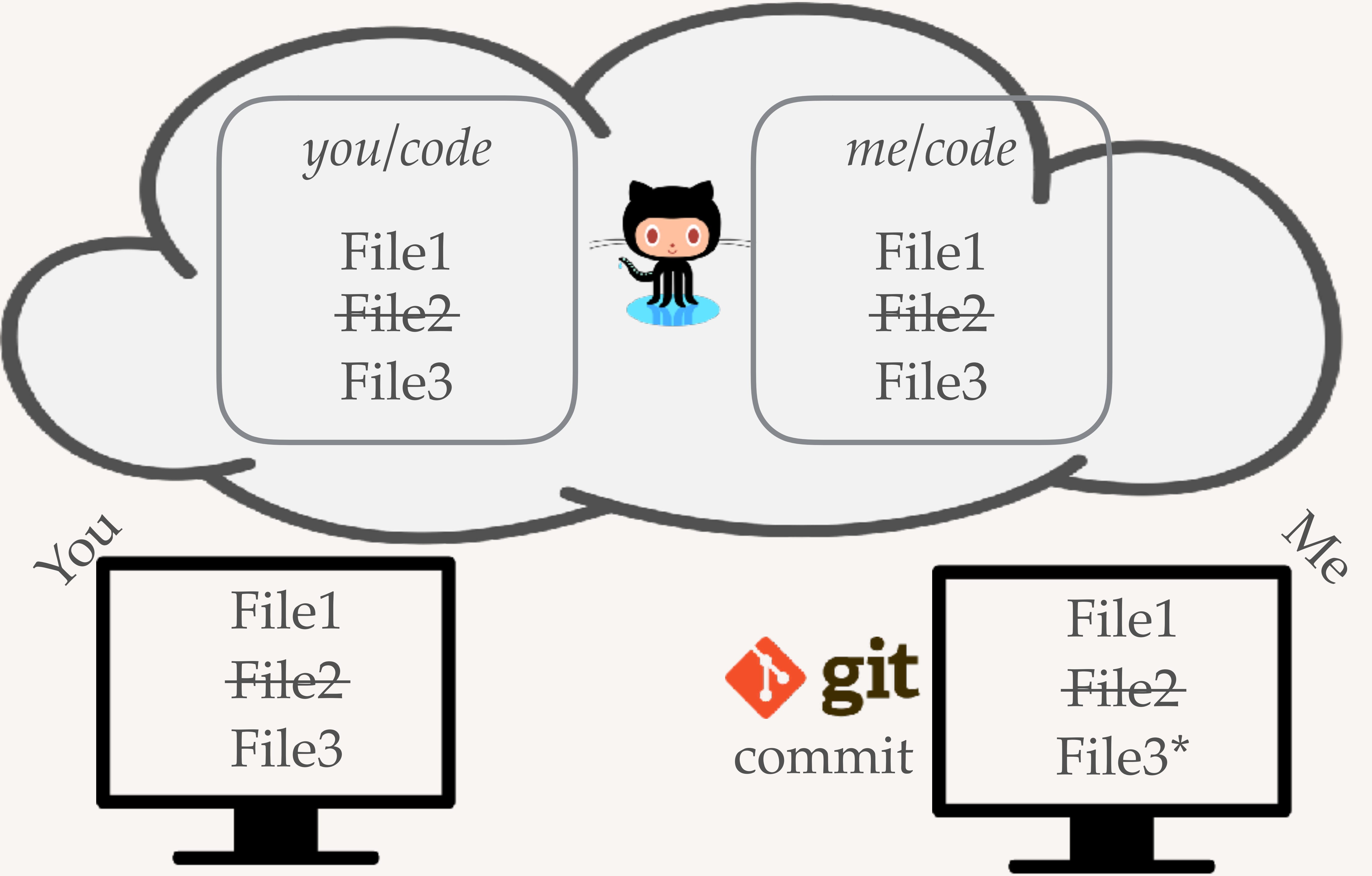
credit: E. Tollerud

Basic Idea Behind git + GitHub



credit: E. Tollerud

Basic Idea Behind git + GitHub



you/code

File1
~~File2~~
File3

me/code

File1
~~File2~~
File3

You

File1
~~File2~~
File3

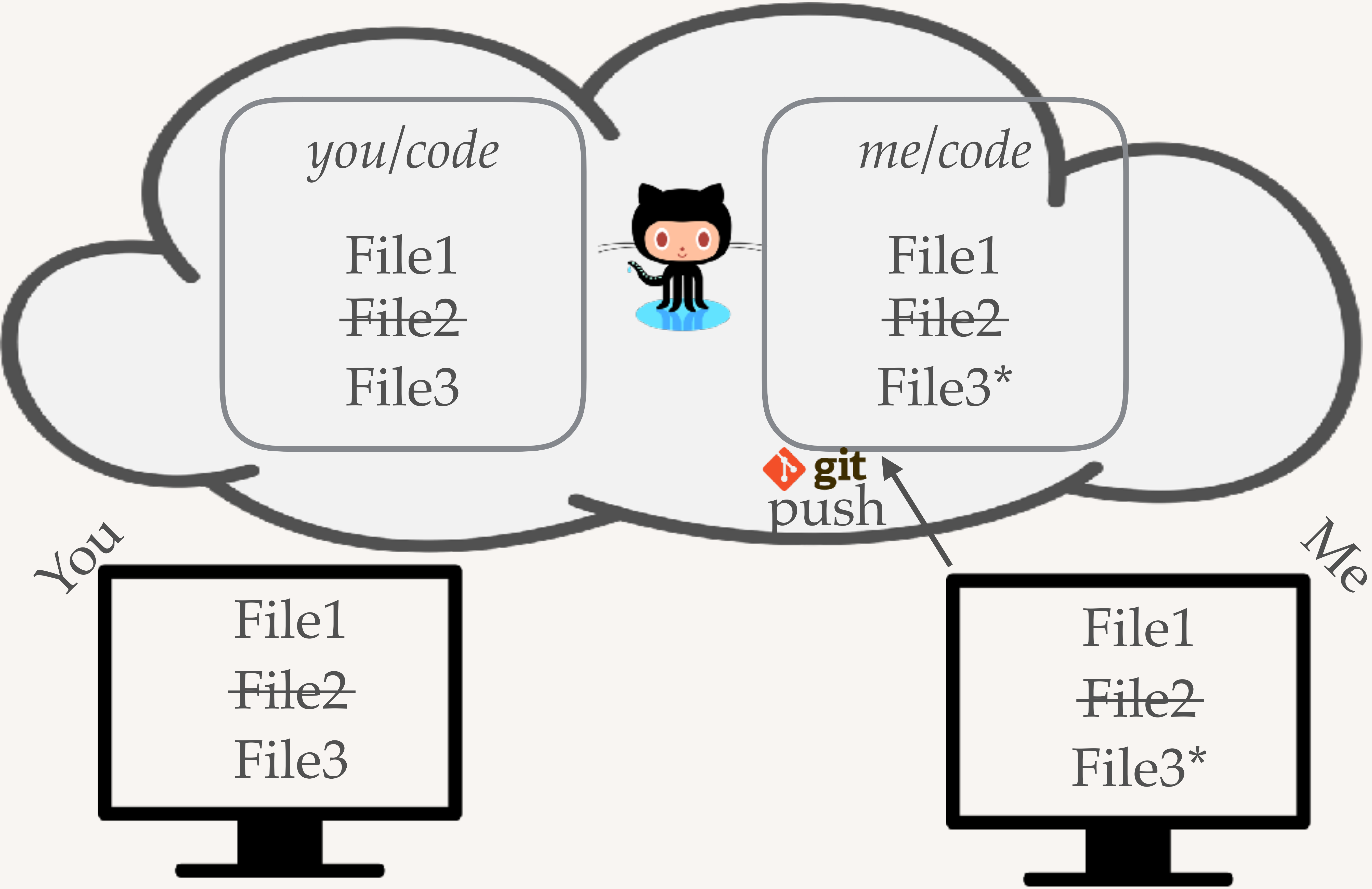
git
commit

Me

File1
~~File2~~
File3*

credit: E. Tollerud

Basic Idea Behind git + GitHub



you/code

File1
~~File2~~
File3

me/code

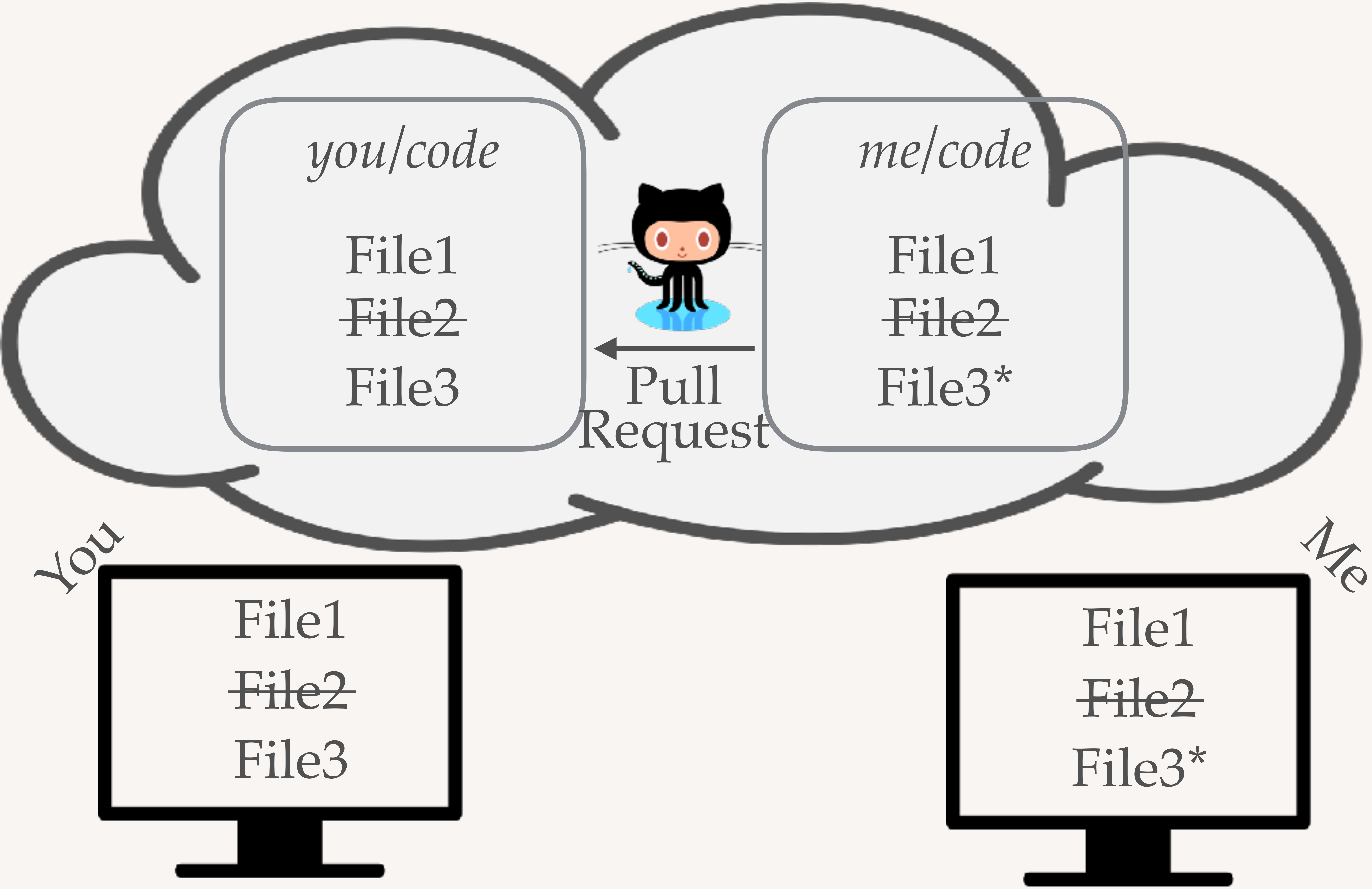
File1
~~File2~~
File3*

File1
~~File2~~
File3

File1
~~File2~~
File3*

credit: E. Tollerud

Basic Idea Behind git + GitHub



you/code

File1
~~File2~~
File3

Pull
Request

me/code

File1
~~File2~~
File3*

You

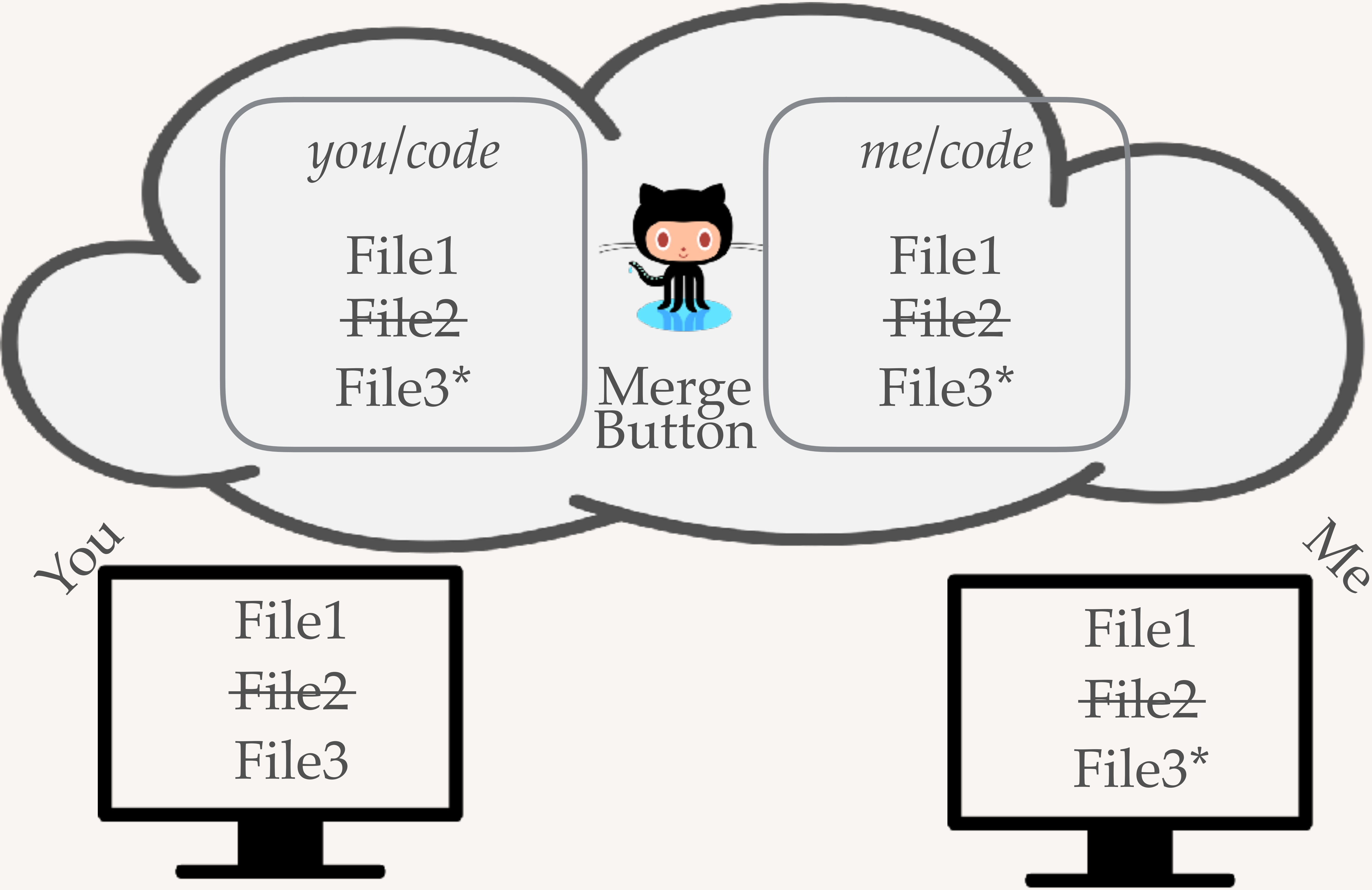
File1
~~File2~~
File3

Me

File1
~~File2~~
File3*

credit: E. Tollerud

Basic Idea Behind git + GitHub



you/code

File1

~~File2~~

File3*

Merge Button

me/code

File1

~~File2~~

File3*

You

File1

~~File2~~

File3

Me

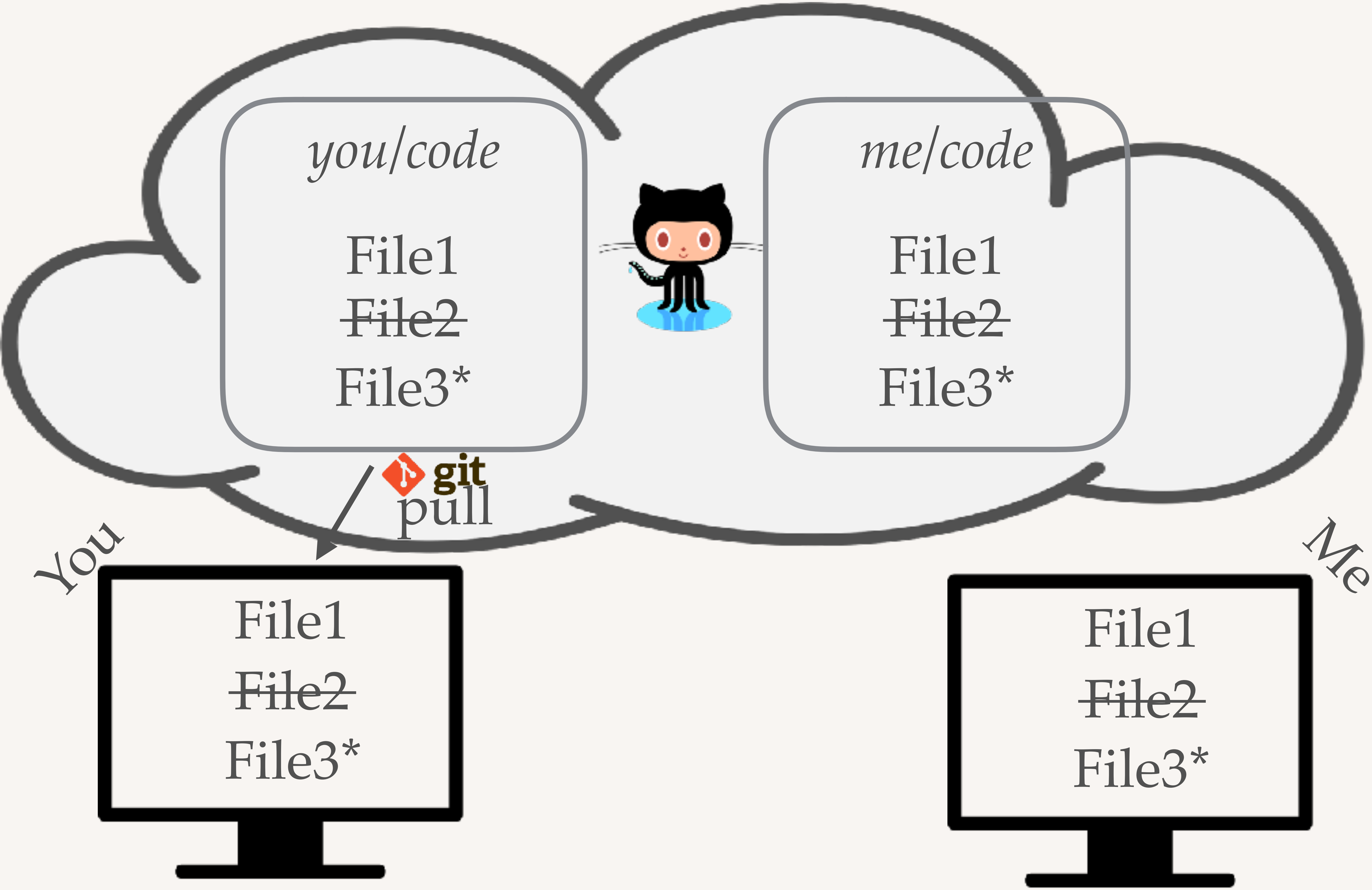
File1

~~File2~~

File3*

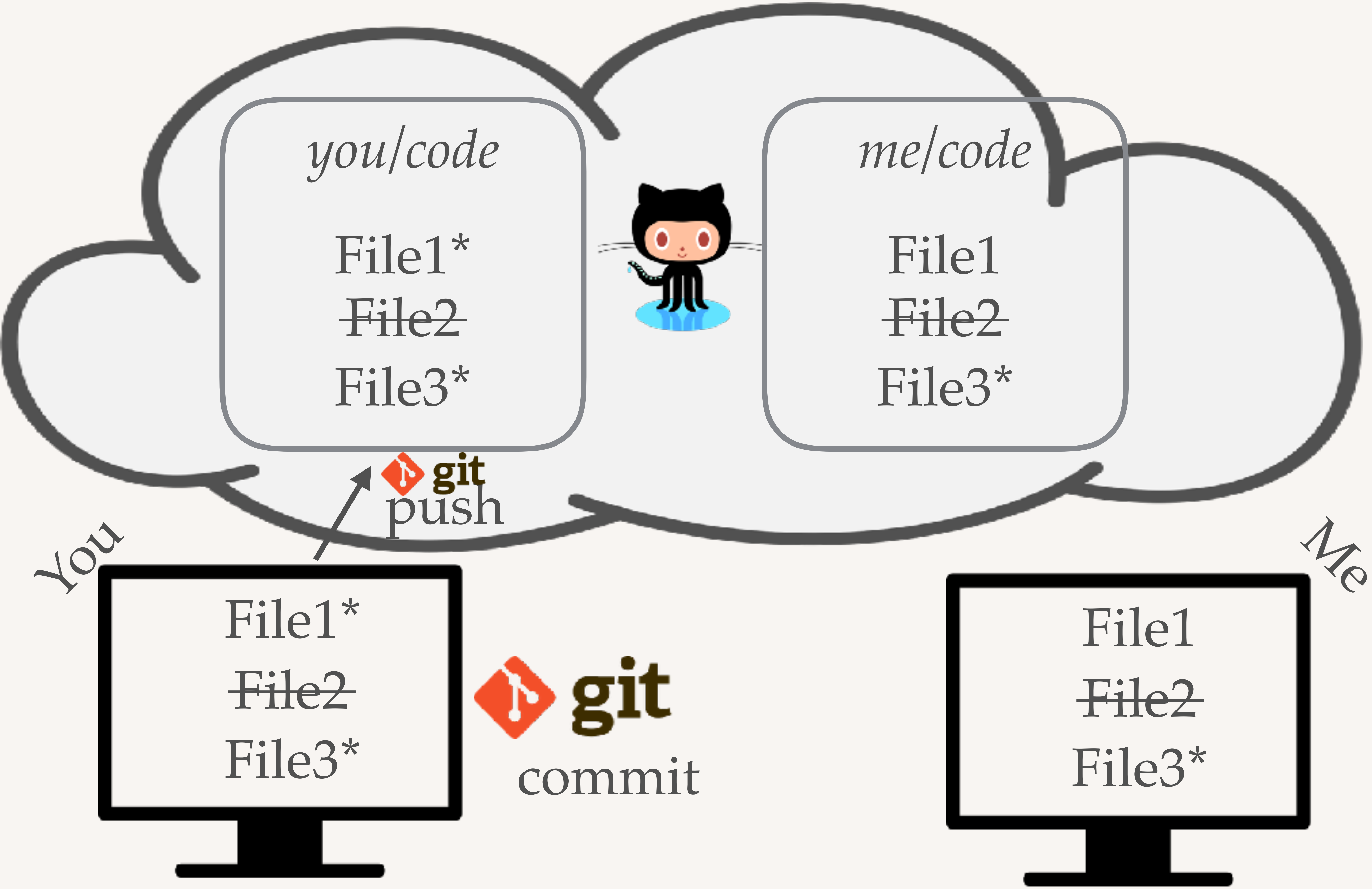
credit: E. Tollerud

Basic Idea Behind git + GitHub



credit: E. Tollerud

Basic Idea Behind git + GitHub



you/code

File1*
~~File2~~
File3*

me/code

File1
~~File2~~
File3*

git push

You

File1*
~~File2~~
File3*

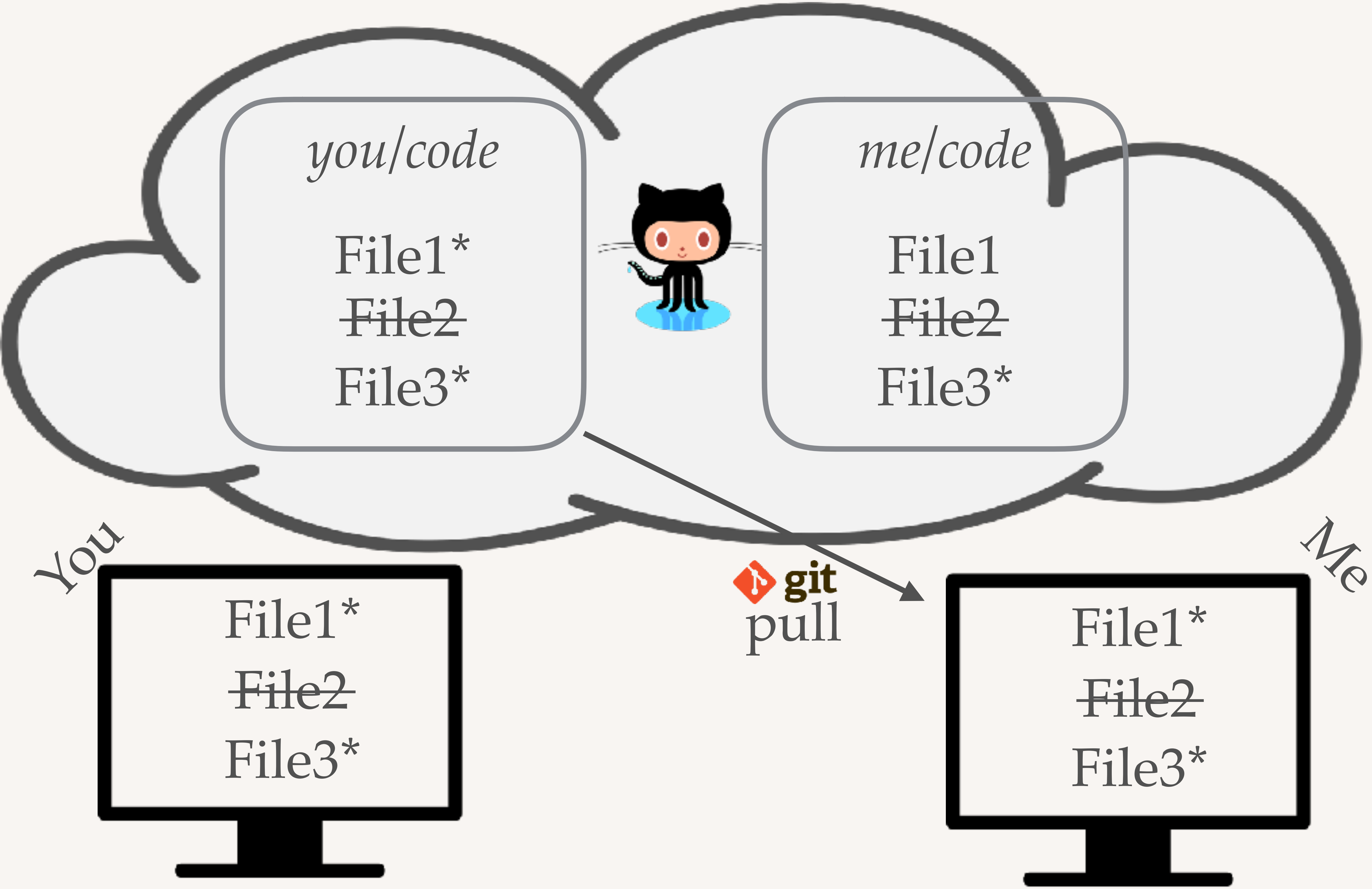
git commit

Me

File1
~~File2~~
File3*

credit: E. Tollerud

Basic Idea Behind git + GitHub



you/code

File1*
~~File2~~
File3*

me/code

File1
~~File2~~
File3*

You

File1*
~~File2~~
File3*

git
pull

Me

File1*
~~File2~~
File3*

credit: E. Tollerud

Basic Idea Behind git + GitHub

Ultimately, you only really need to know 3 commands:

```
git add <filename>

git commit -m “<a descriptive commit message>”

git push origin <branch name>
```

(you will need more while working with collaborators, but this covers > 90% of what you will do this summer)

Basic Idea Behind git + GitHub

```
git add <filename>
```

This “stages” the file to commit to the repo

<filename> = file to be committed in the repo

(at this point nothing major has occurred)

Basic Idea Behind git + GitHub

```
git commit -m “<a descriptive commit message>”
```

This “commits” the file to the repo

<message> = messages should be short **and** descriptive

e.g., “fixed” doesn’t describe

someone should be able to read commit messages and understand code

(at this point file is “logged”)

Basic Idea Behind git + GitHub

```
git push origin <branch name>
```

This updates the cloud to reflect recent commits

<branch name> = branch on which work is being done

most likely *master* or *dev*

(at this point your code is safe - even if you drop your laptop)

Other Commands Worth Knowing

As you build up your toolbox you may also want:

```
git branch
```

report current branch within the repo

```
git status
```

summary of current branch, files staged for commit, & untracked files

```
git checkout -b <branch name>
```

create a new branch called <branch name> & switch to that branch

```
git checkout <branch name>
```

switch to branch <branch name>

```
git merge <branch name>
```

merge changes in <branch name> to the **current** branch

```
git log
```

history of recent commits

```
git log --oneline --topo-order --graph
```

nice graphical log of branches and changes in repo