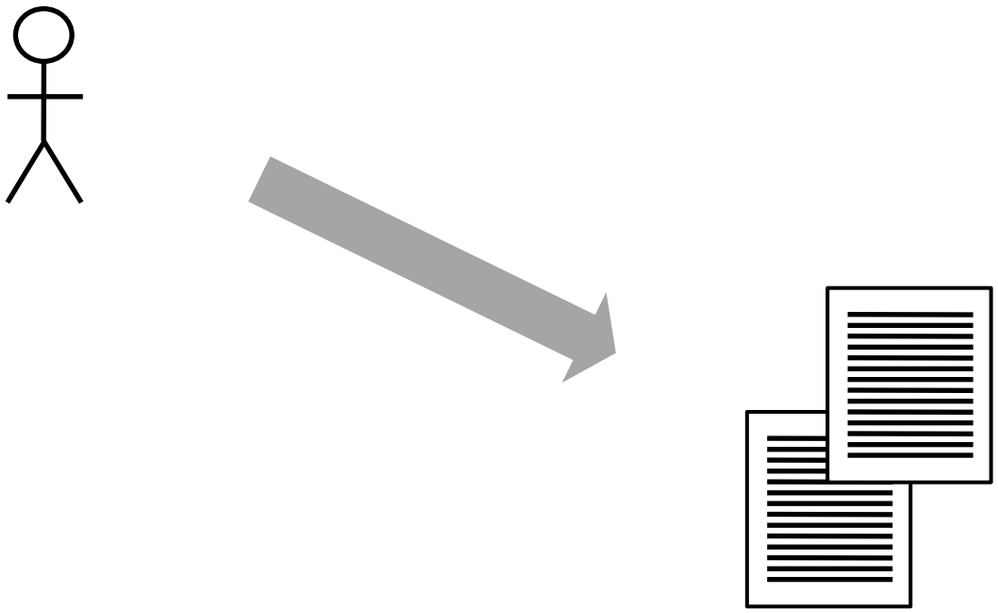
UNIVERSITÄT BERN

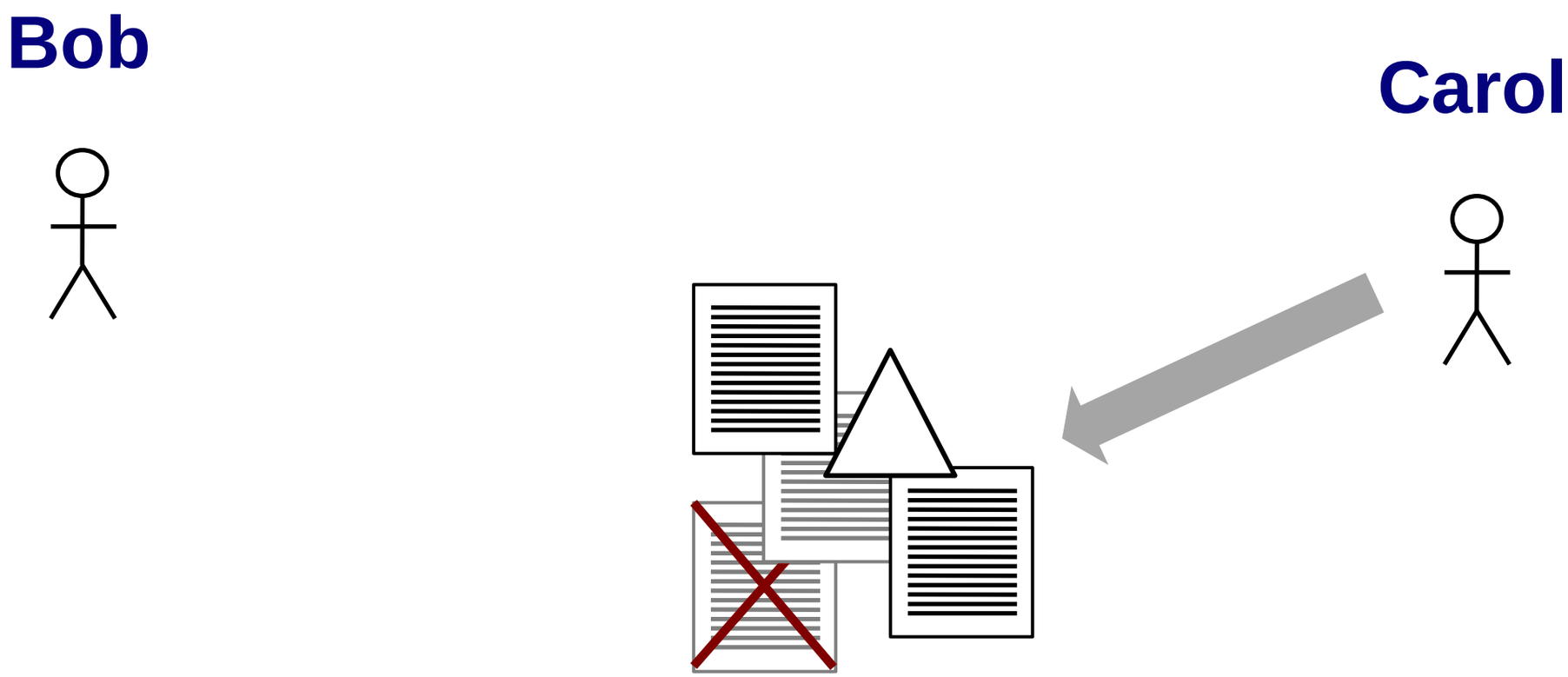
Distributed version control with git — a brief introduction

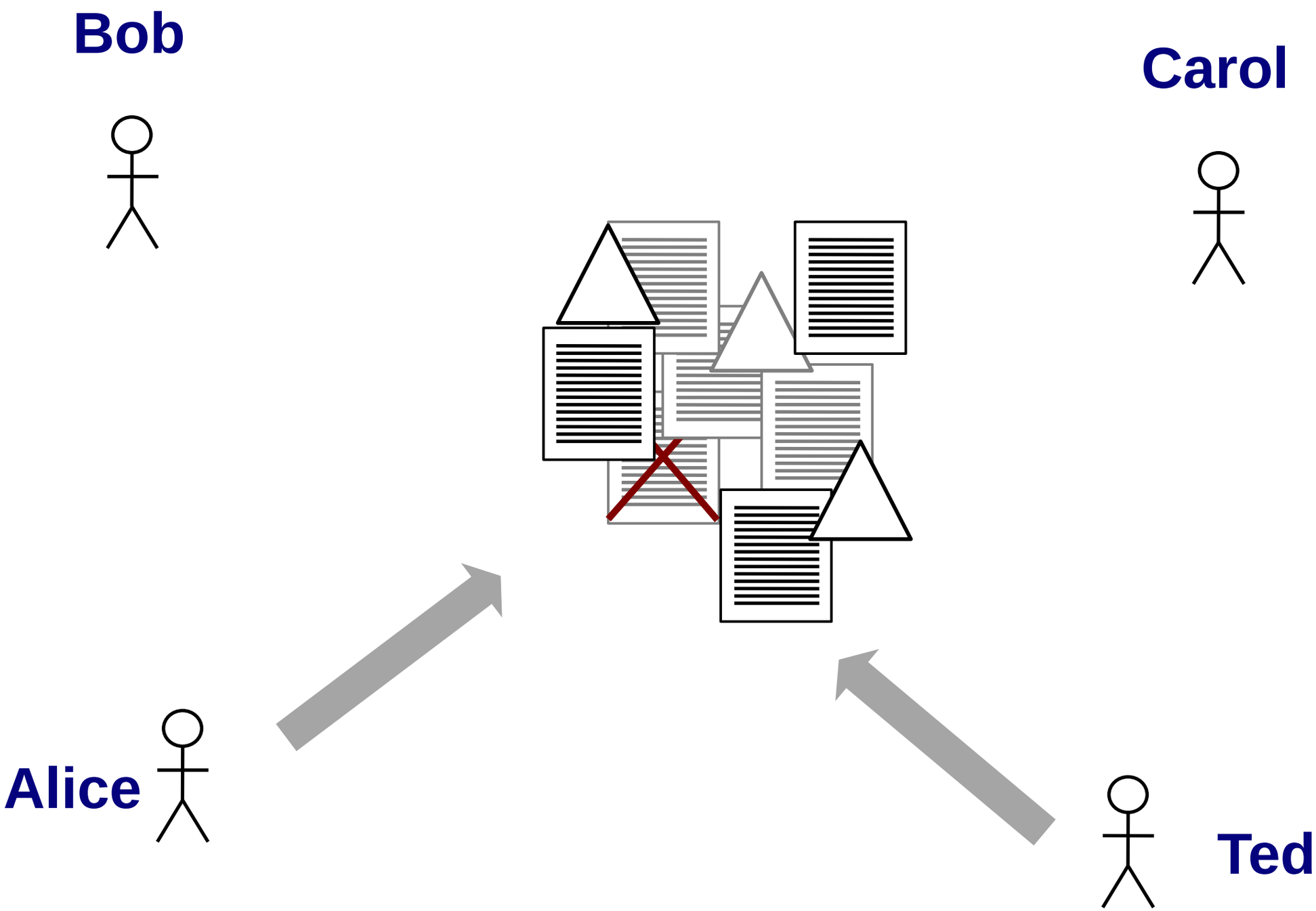
Joel Krebs
Slides by Oscar Nierstrasz

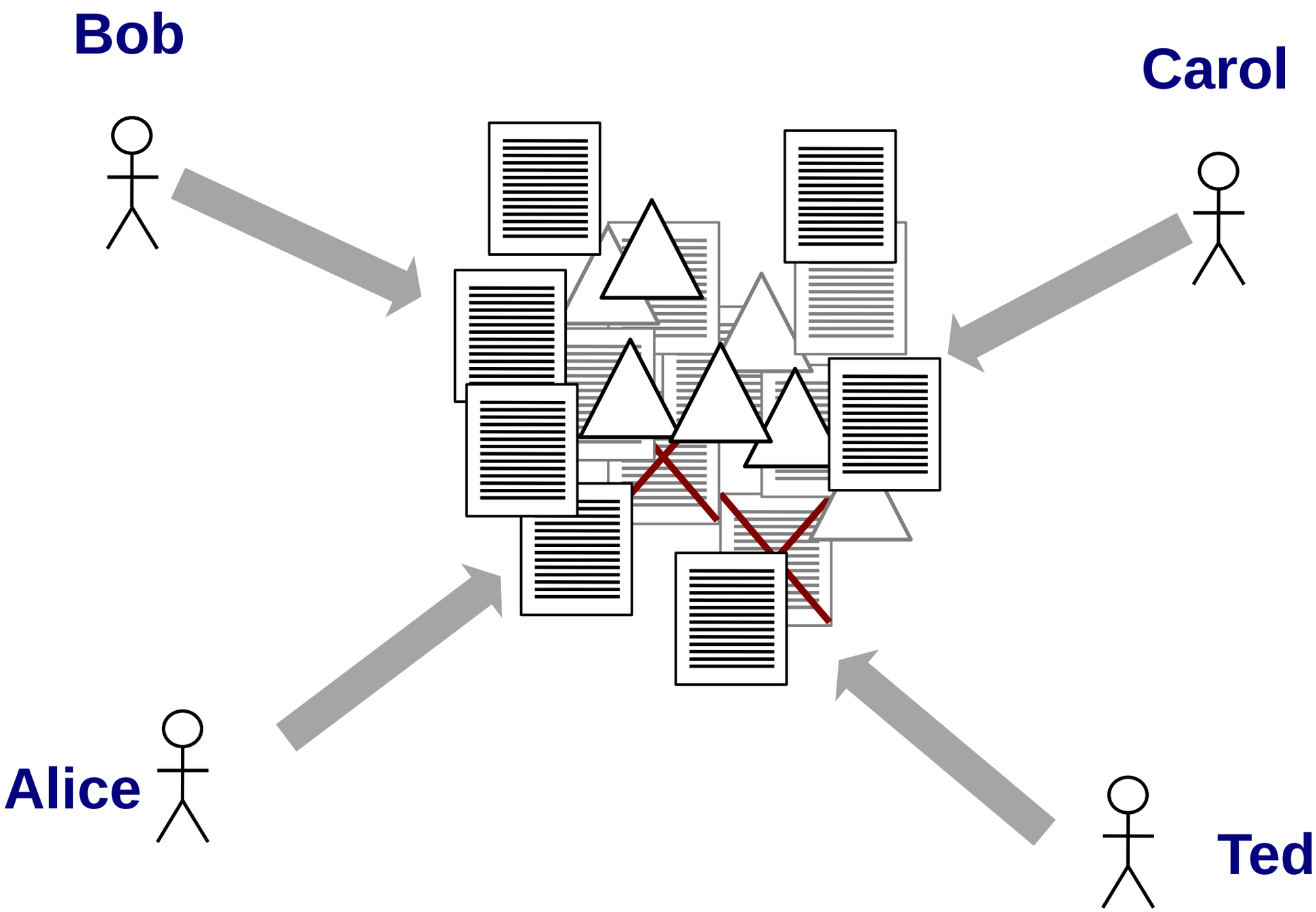
Why git?

Bob




Bob
Carol


Bob
Carol
Alice
Ted



A recipe for disaster!

What is git?

git

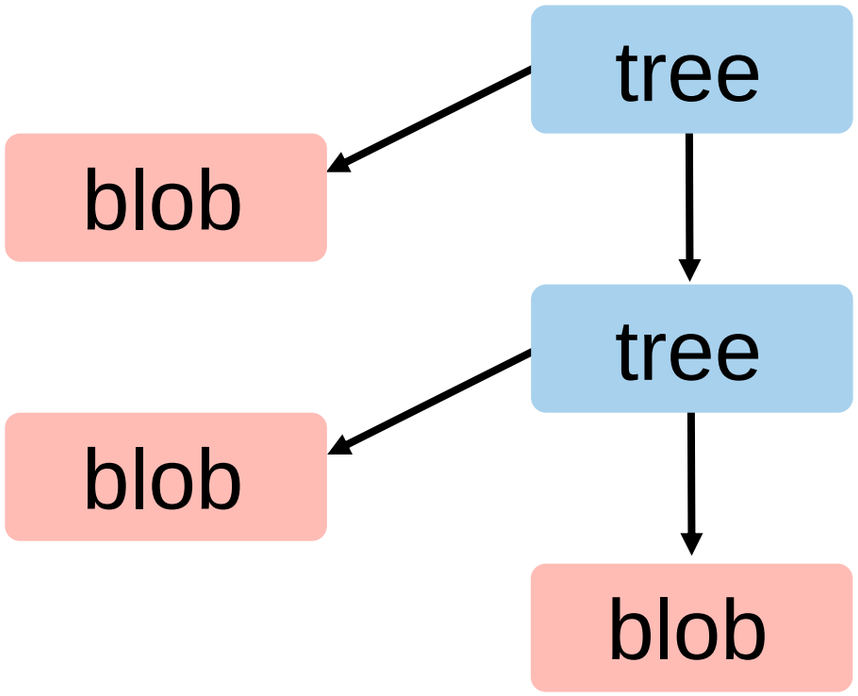
- Distributed revision control system
- Originally developed by Linus Torvalds for the development of the Linux Kernel in 2005
- Focus on speed and efficiency
- Quite a unique design and therefore sometimes a bit scary and difficult to understand

The git object model

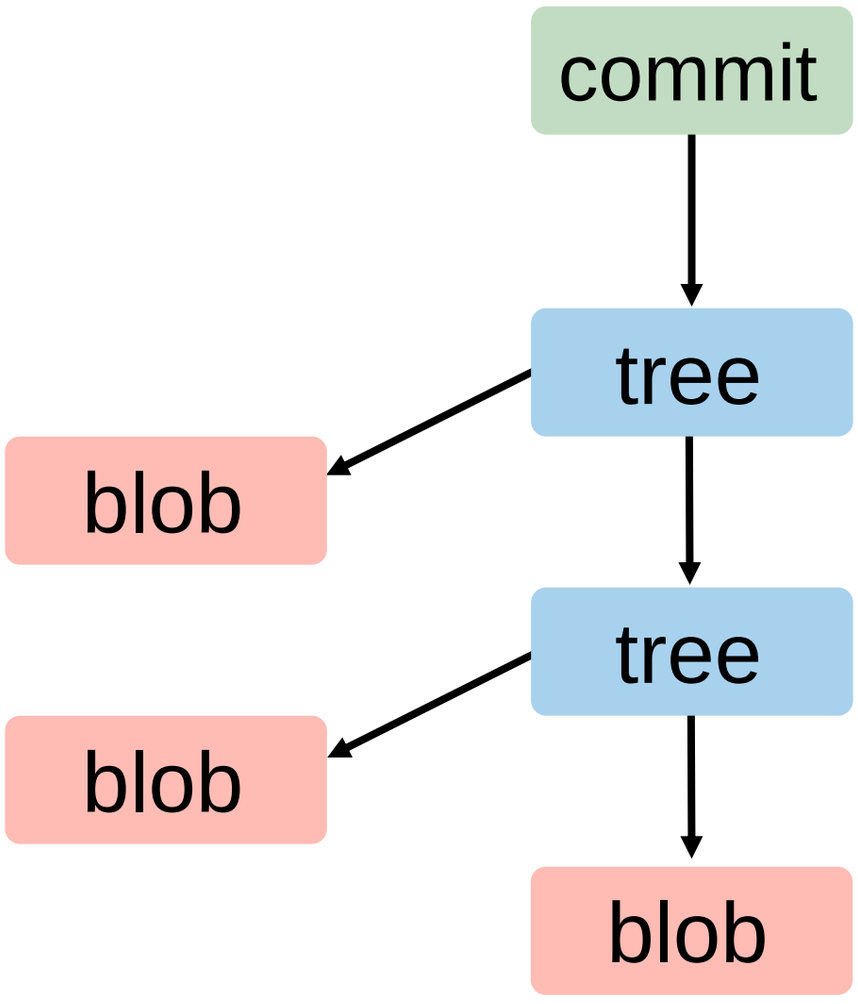
A “blob” is *content* under version control (a file)

blob

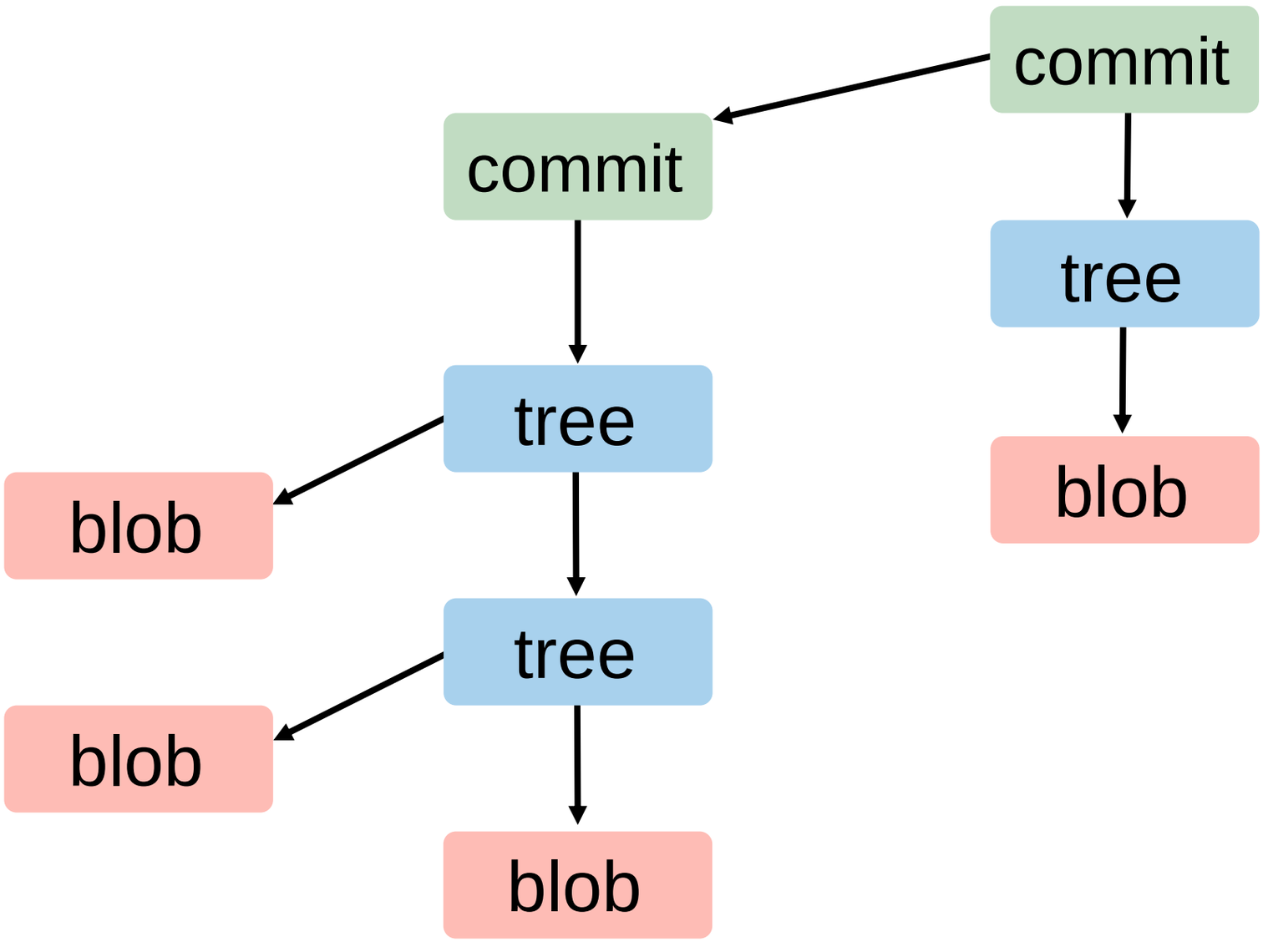
You can have *trees* of blobs
(directories of files)



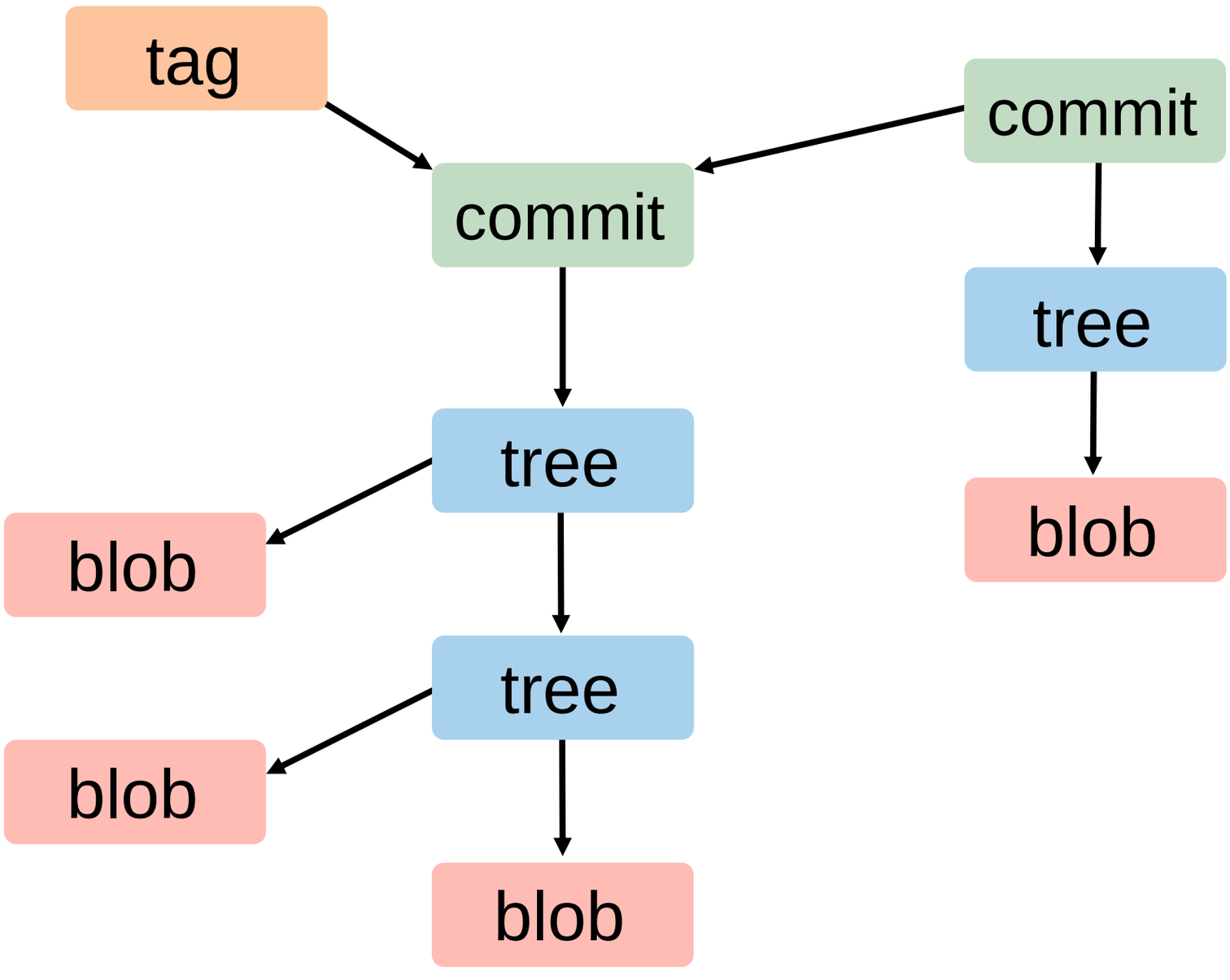
A “commit” is a tree of blobs (a set of changes)



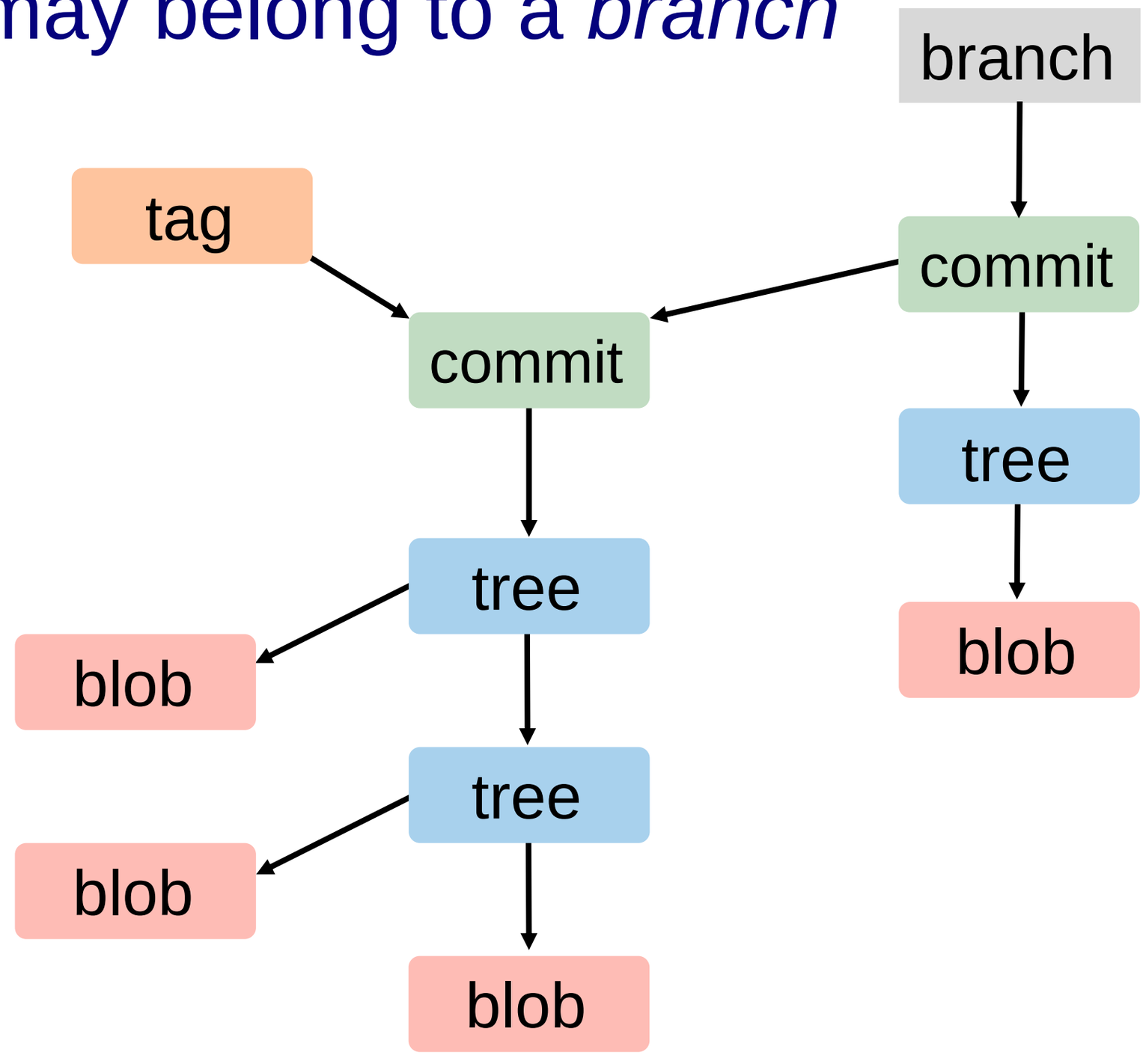
Most commits modify (or merge) earlier commits



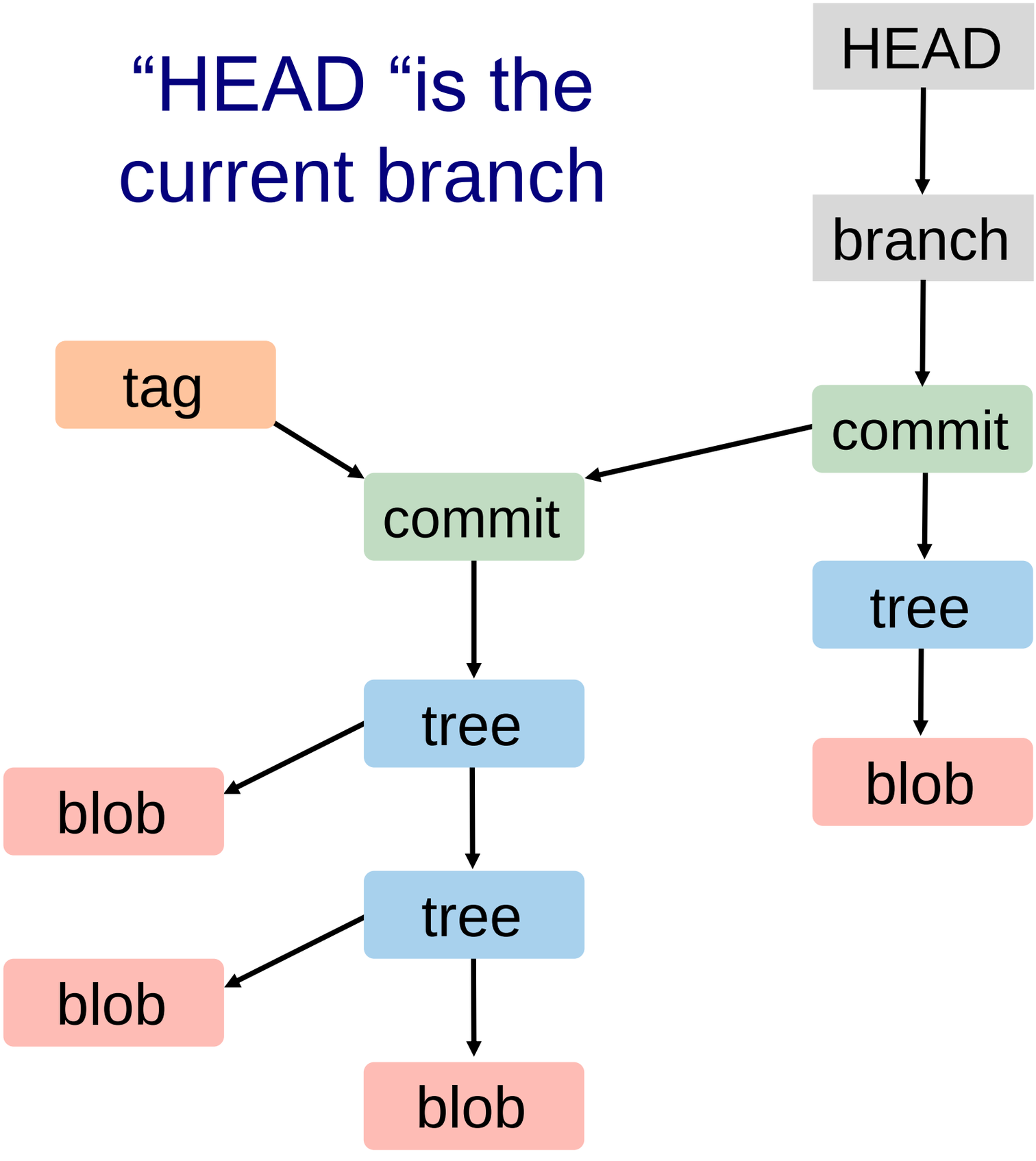
You can “tag” an interesting commit

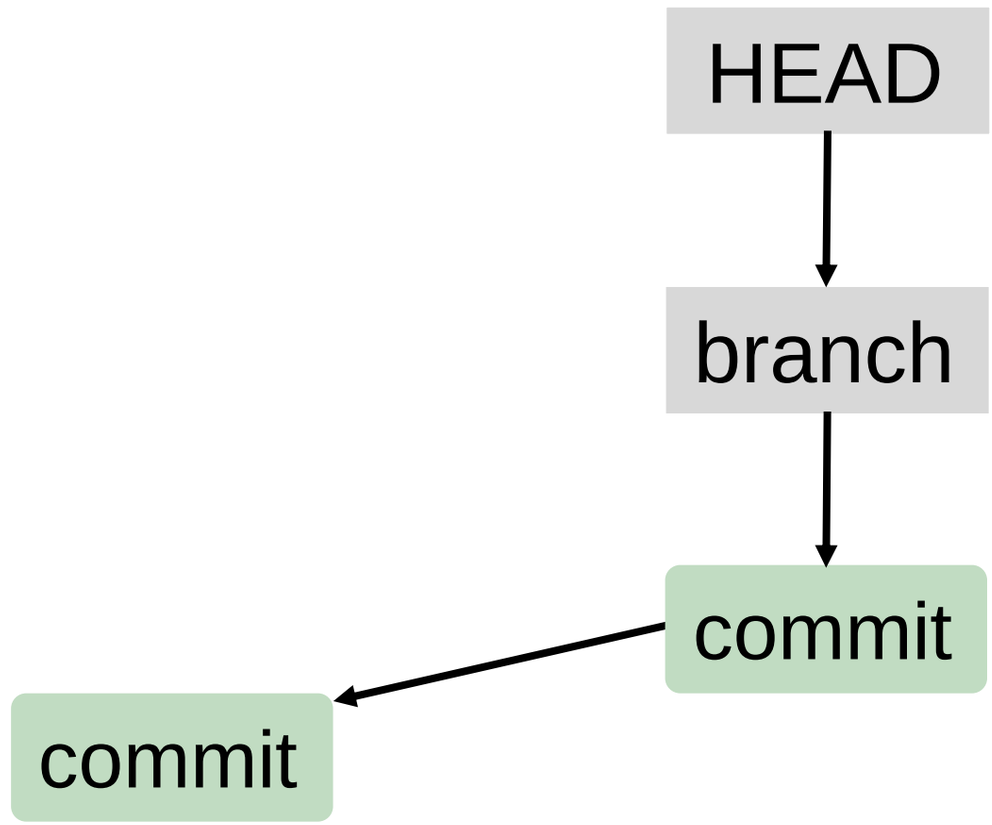


A graph of commits may belong to a *branch*



“HEAD “is the current branch



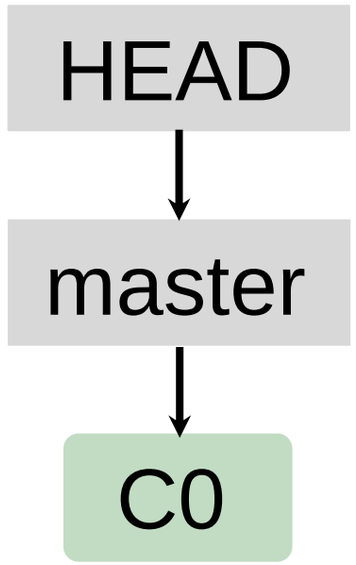


Let’s focus on *commits*
and *branches*

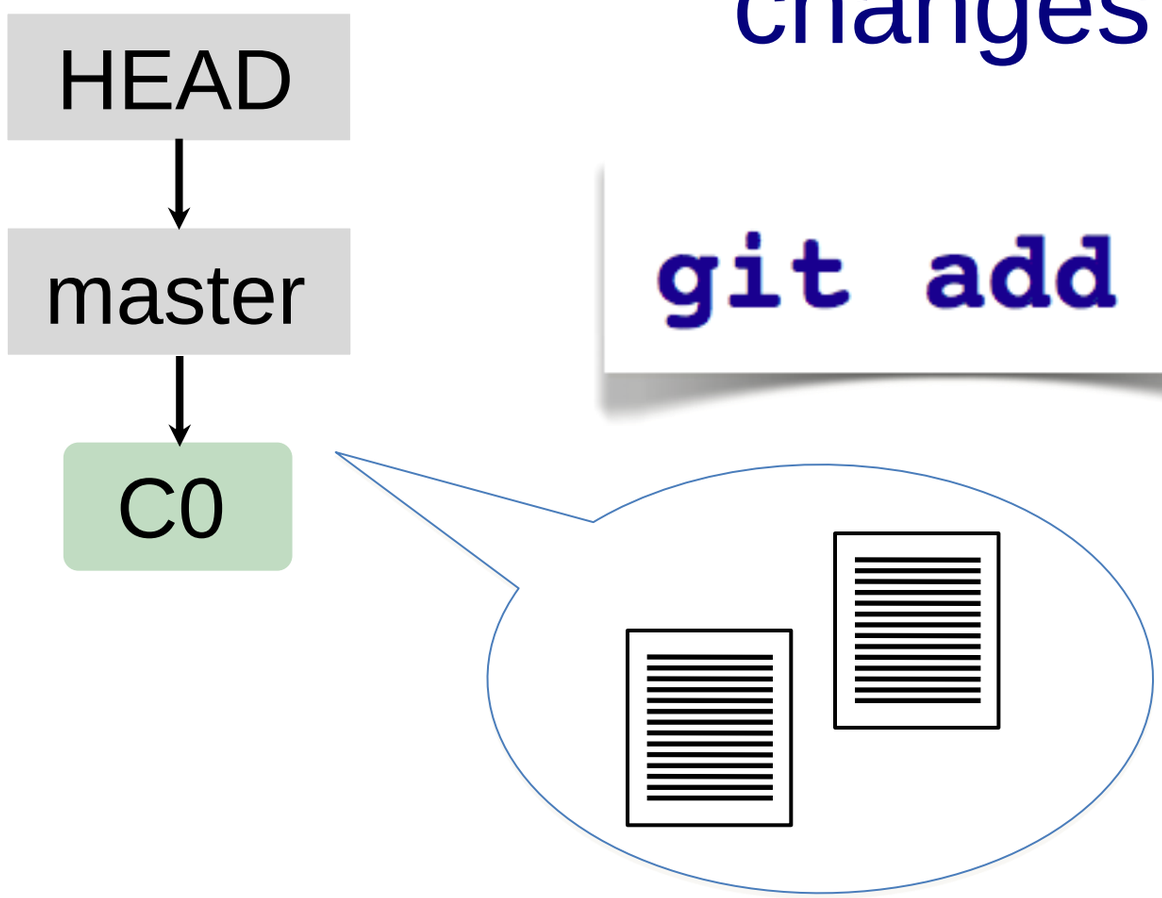
Basic git

Create a git repo

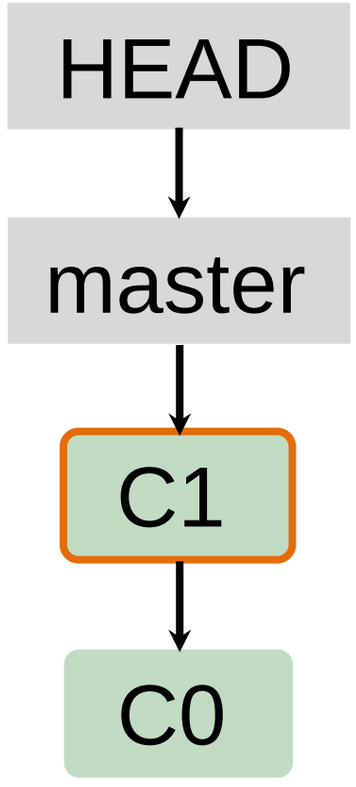
```
mkdir repo
cd repo
git init
```



Tell git to “stage” changes



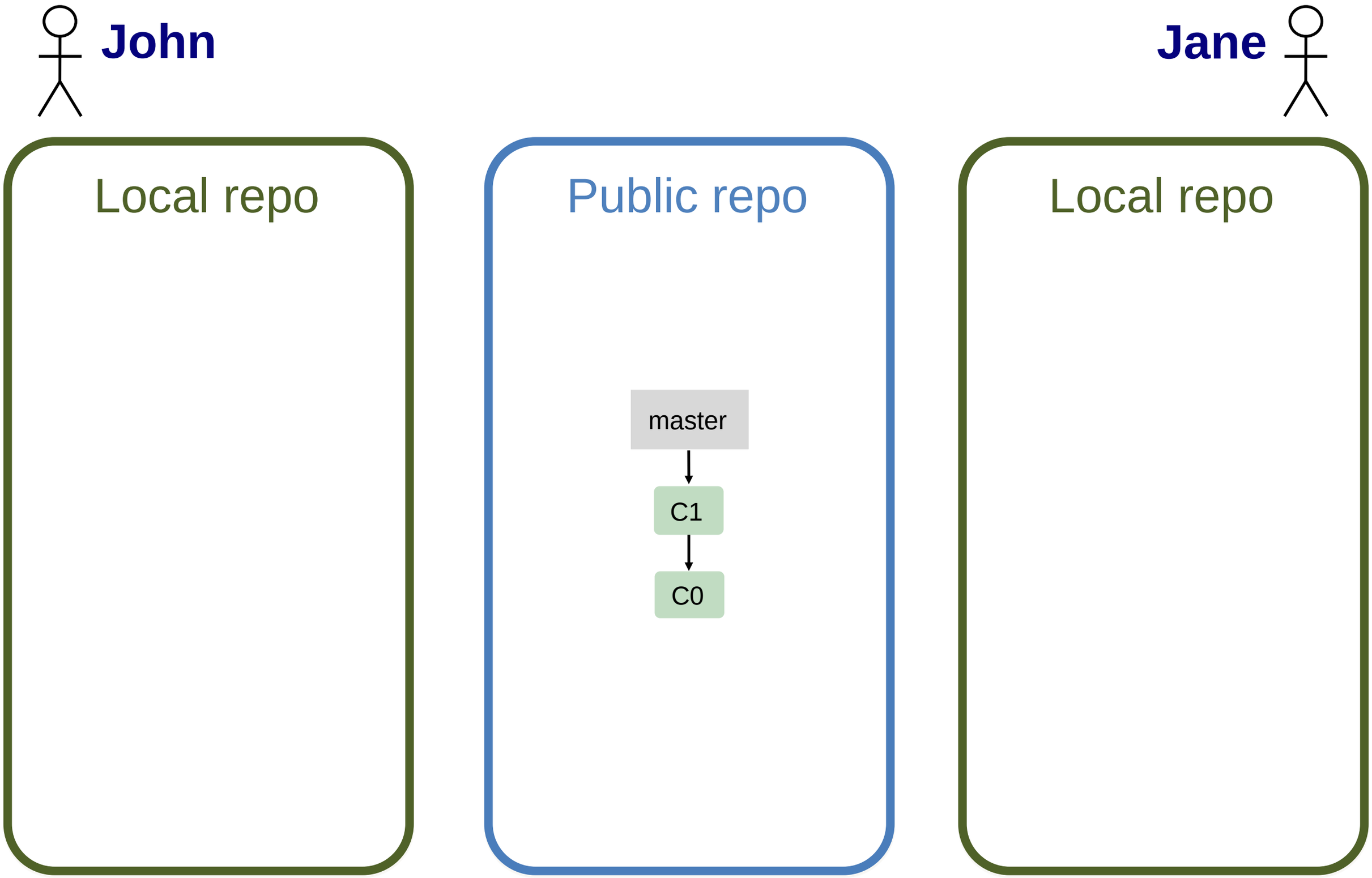

```
git add …
```

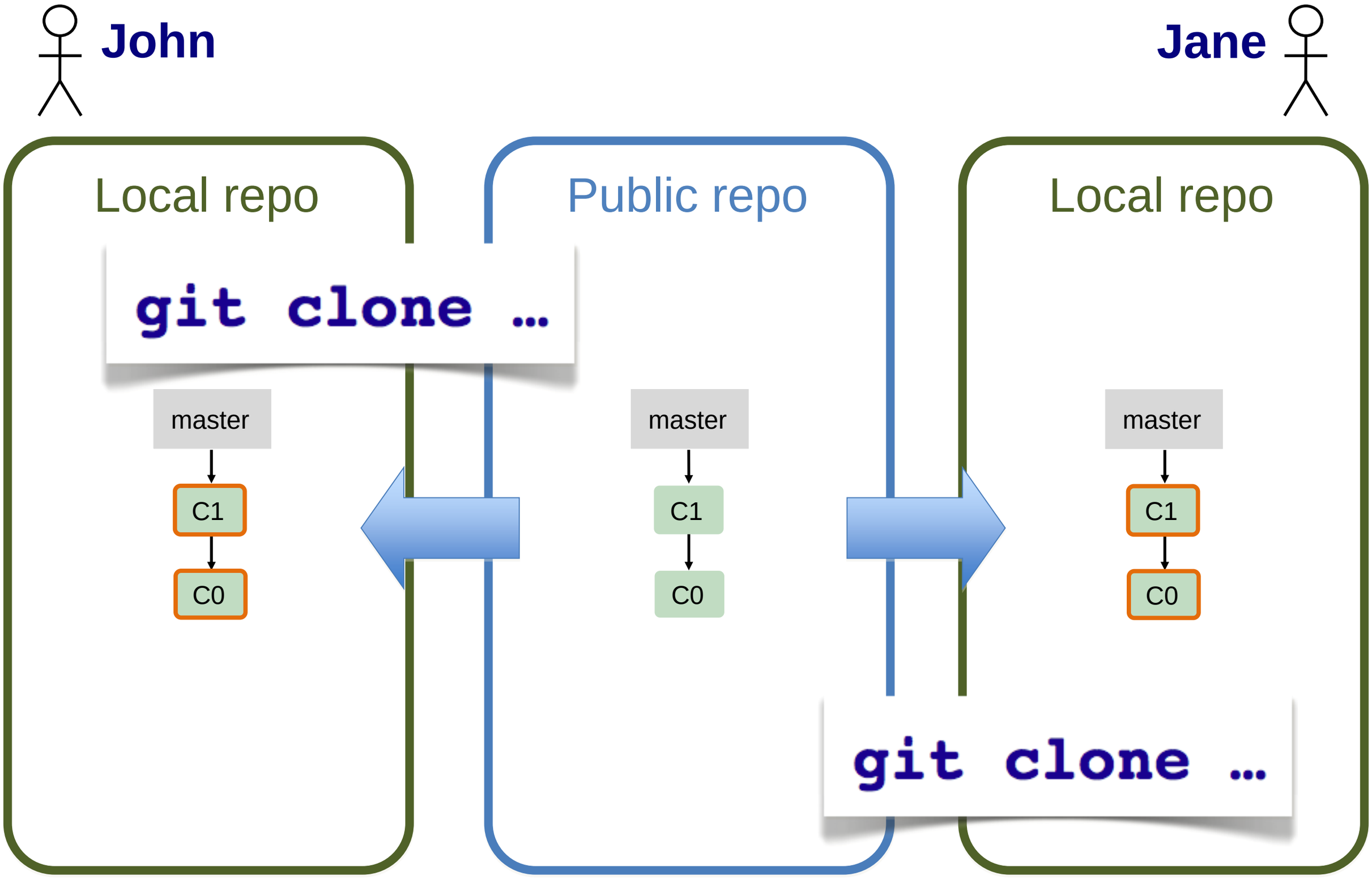


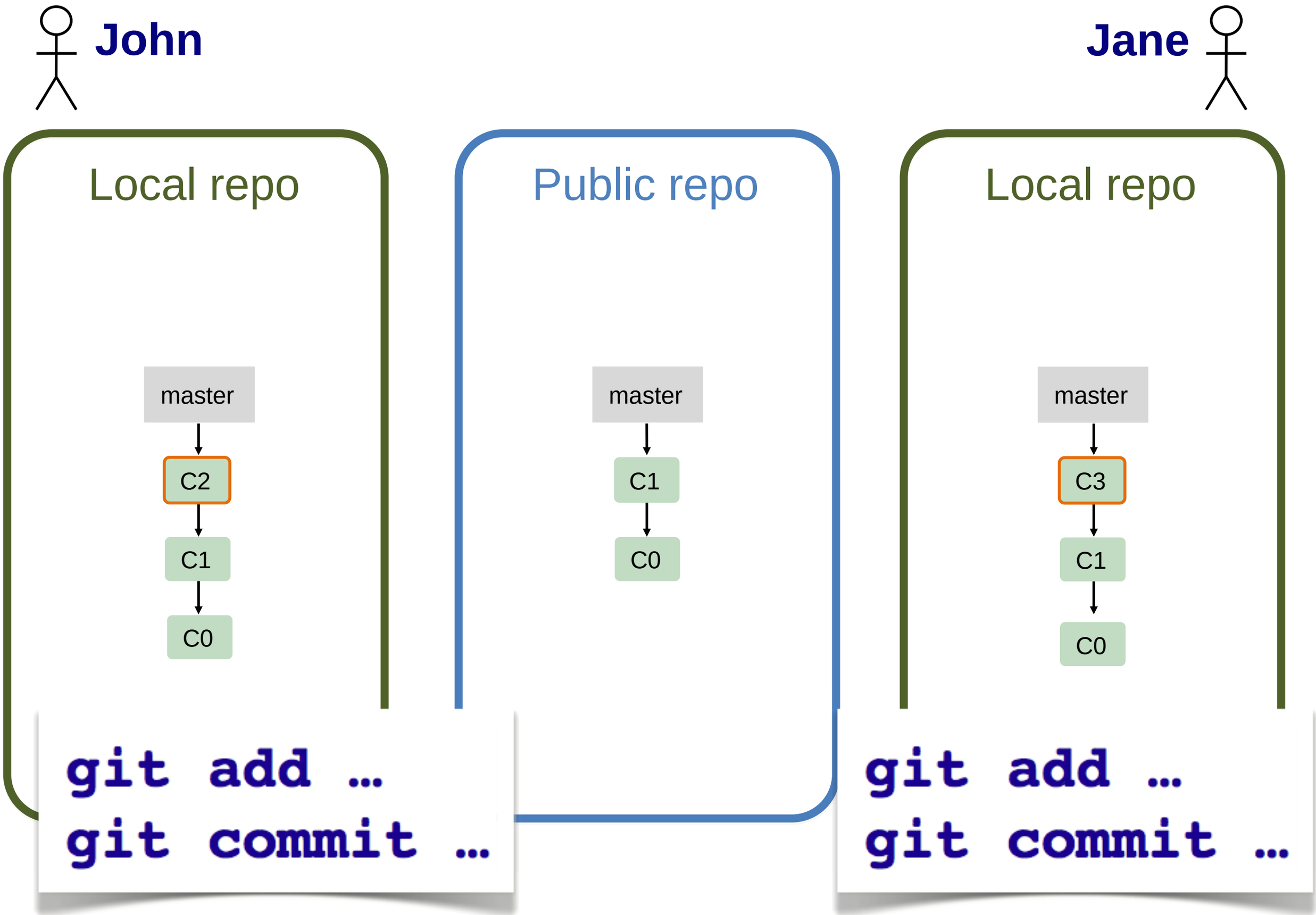
Commit your changes

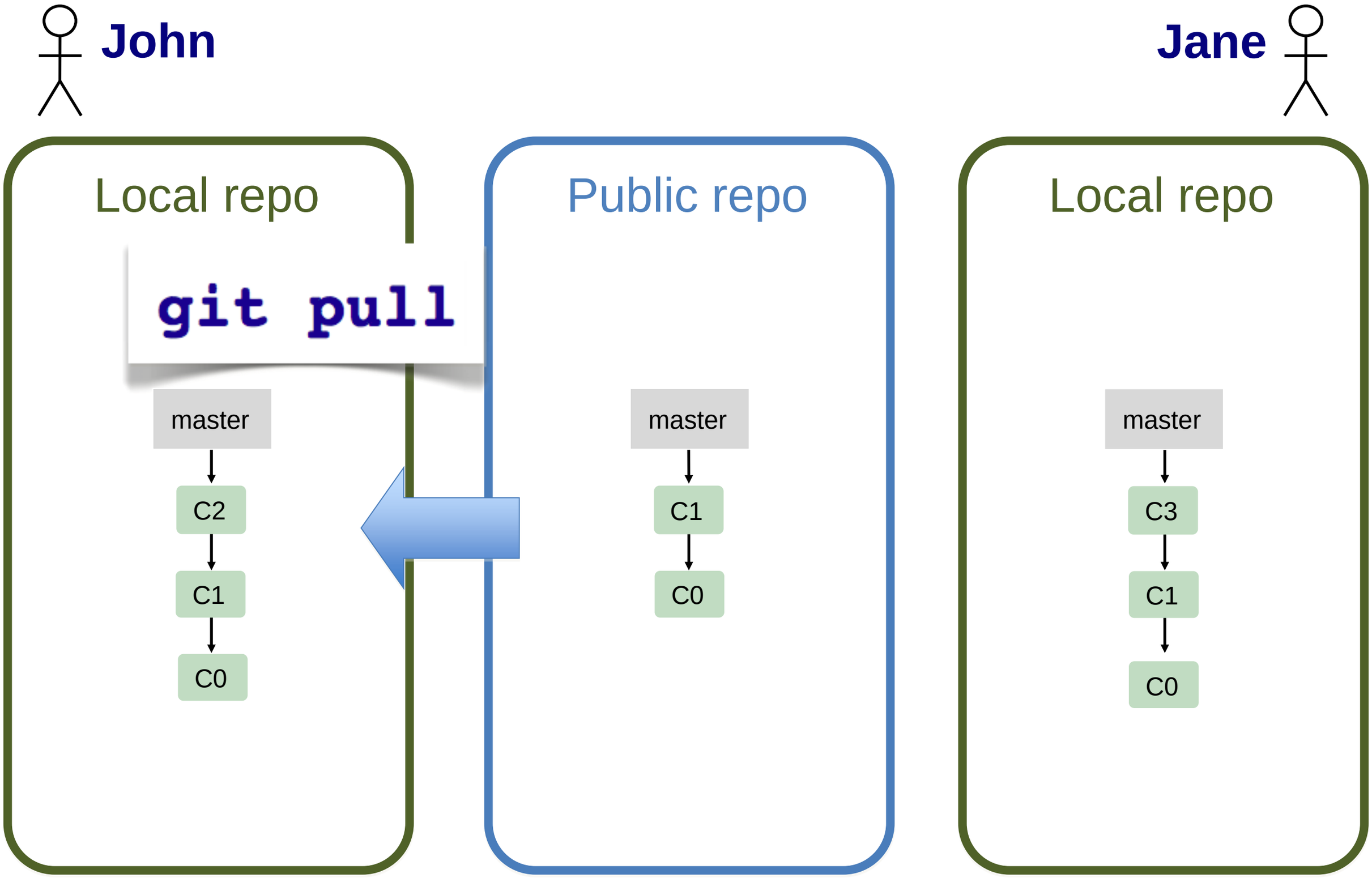
```
git commit …
```

Collaborating

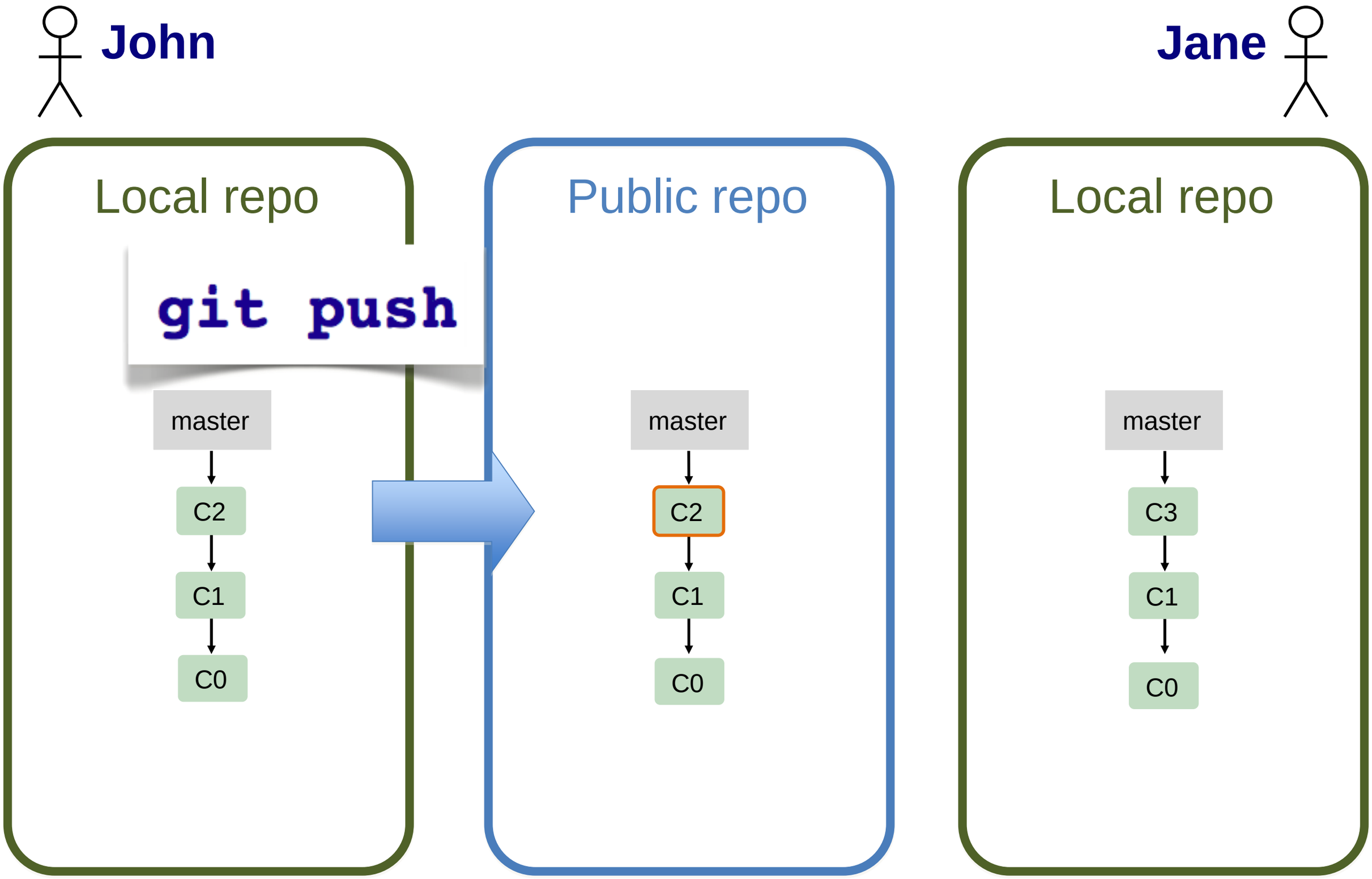

John
Jane
Local repo
Public repo
Local repo
master
C1
C0

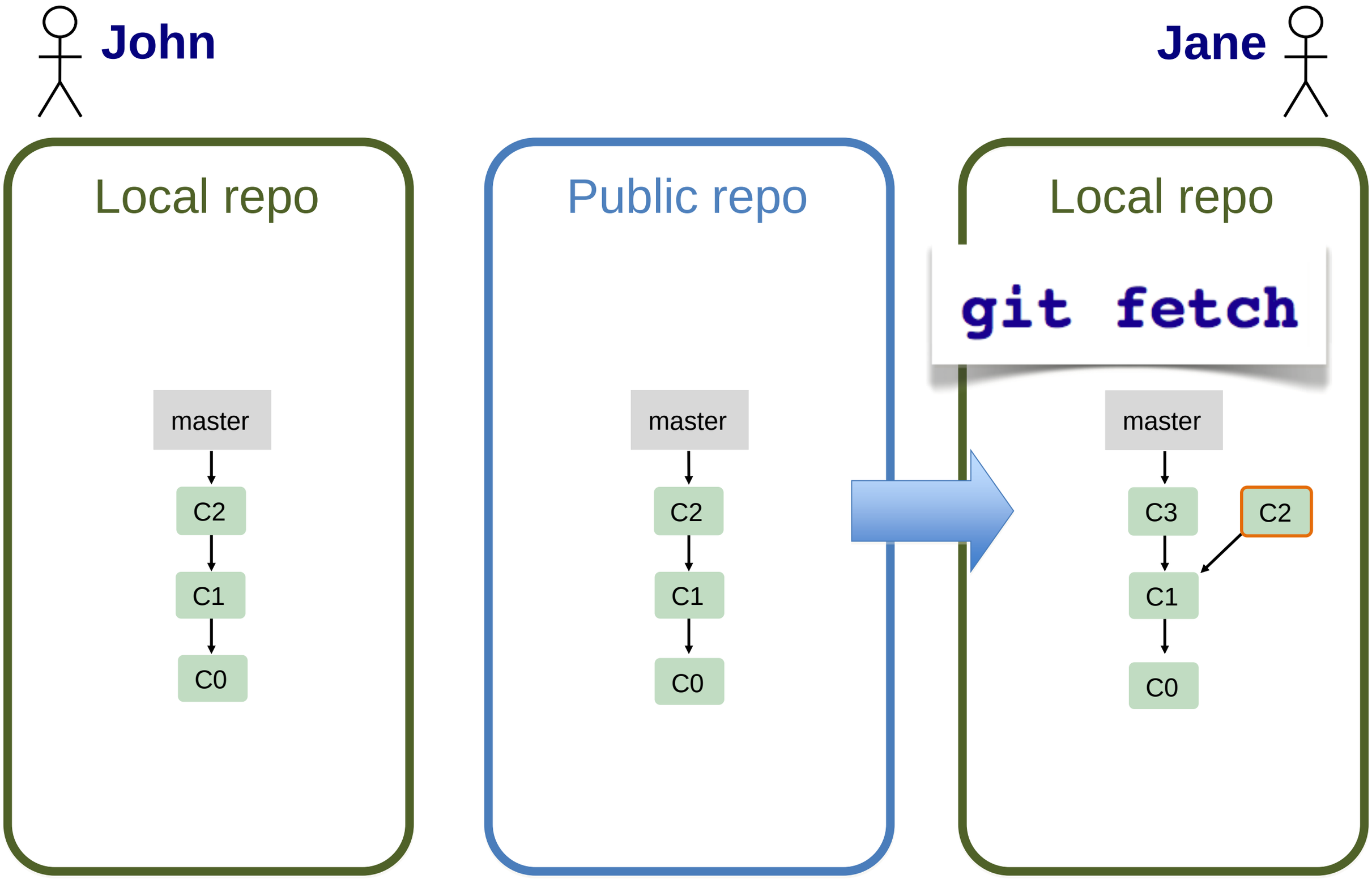

John
Jane
Local repo
Public repo
Local repo
git clone …
master
C1
C0
master
C1
C0
master
C1
C0
git clone …

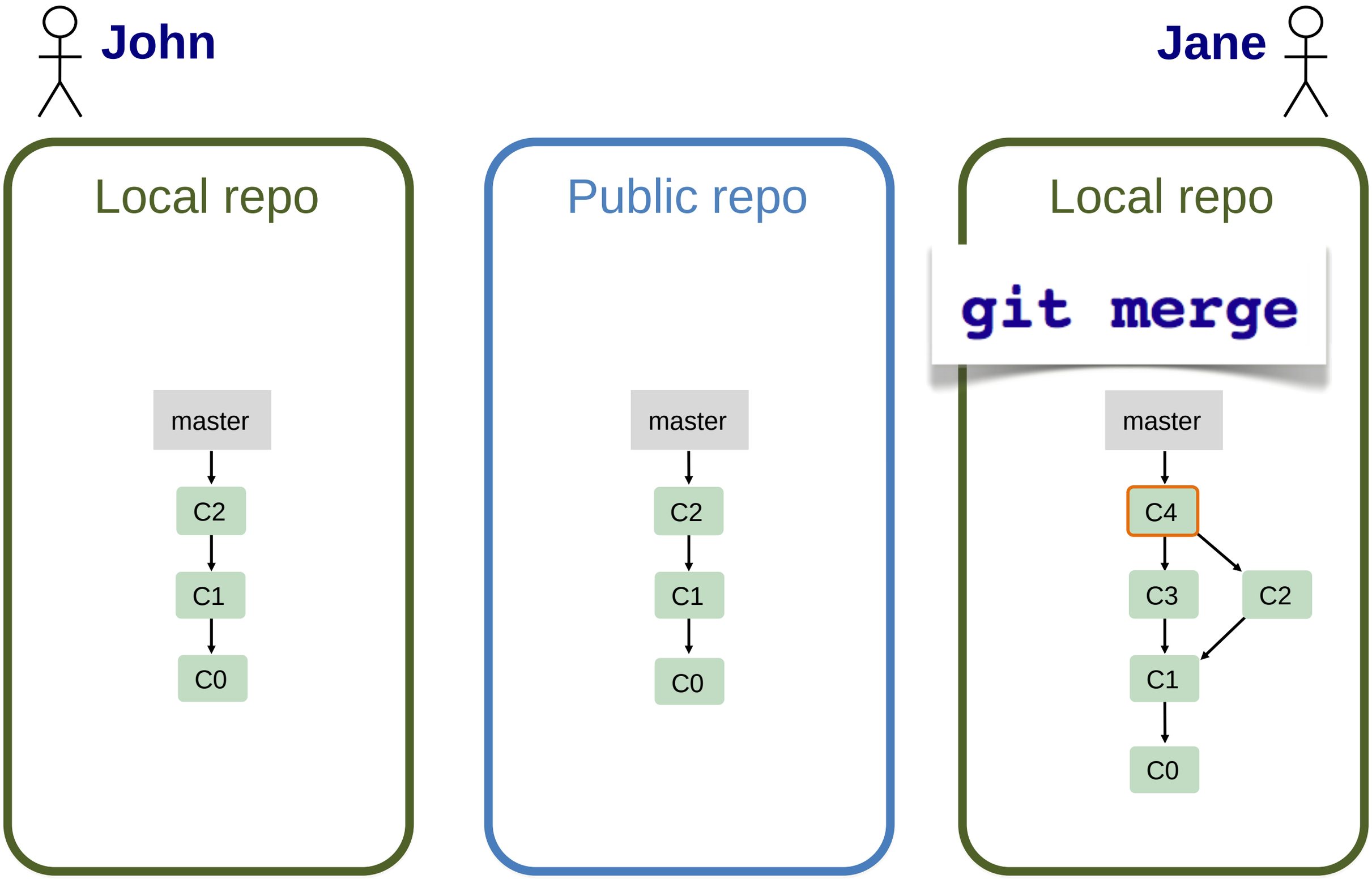

John
Jane
Local repo
Public repo
Local repo
master
C2
C1
C0
master
C1
C0
master
C3
C1
C0
git add …
git commit …
git add …
git commit …



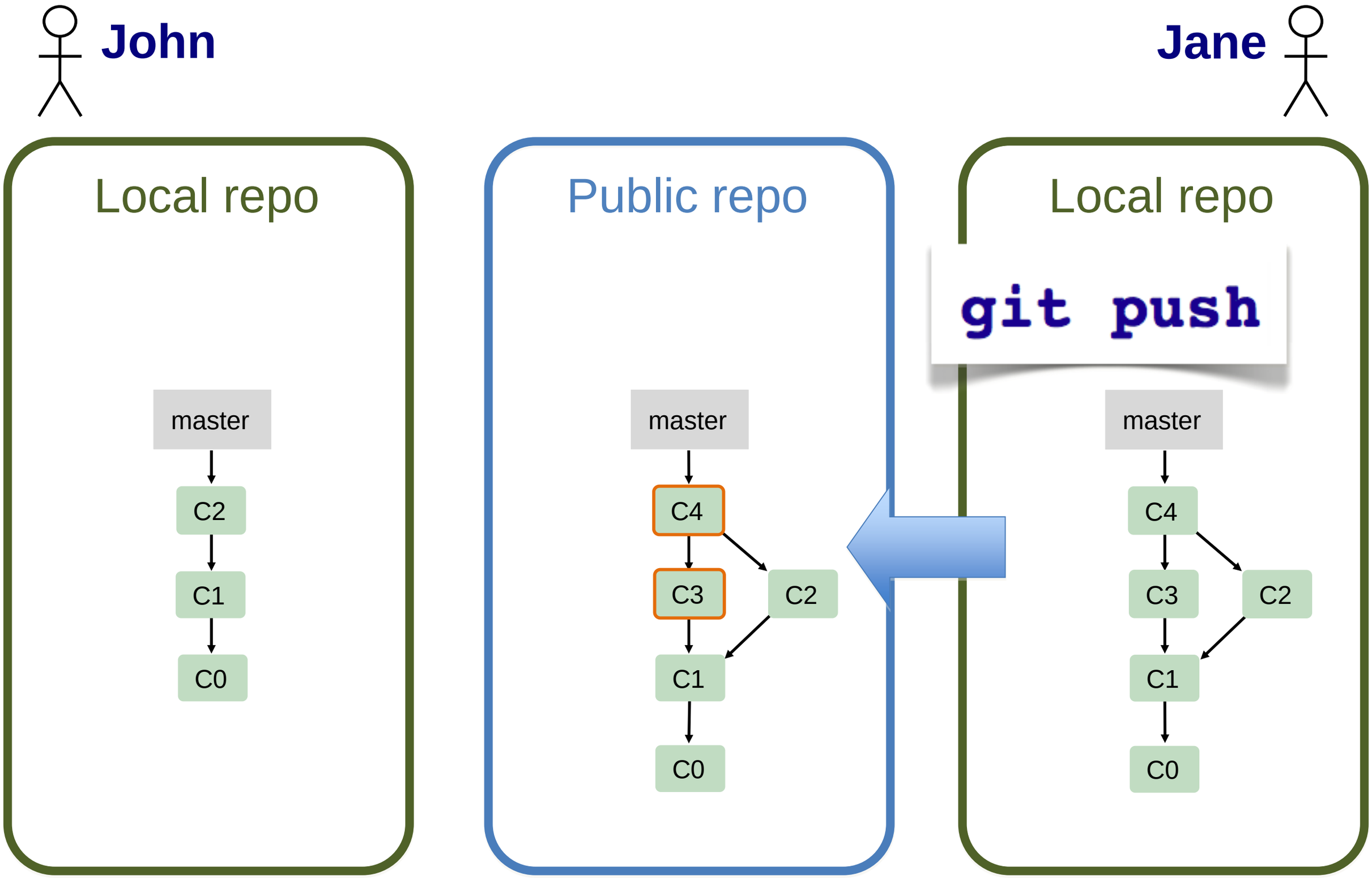
(nothing new to pull)

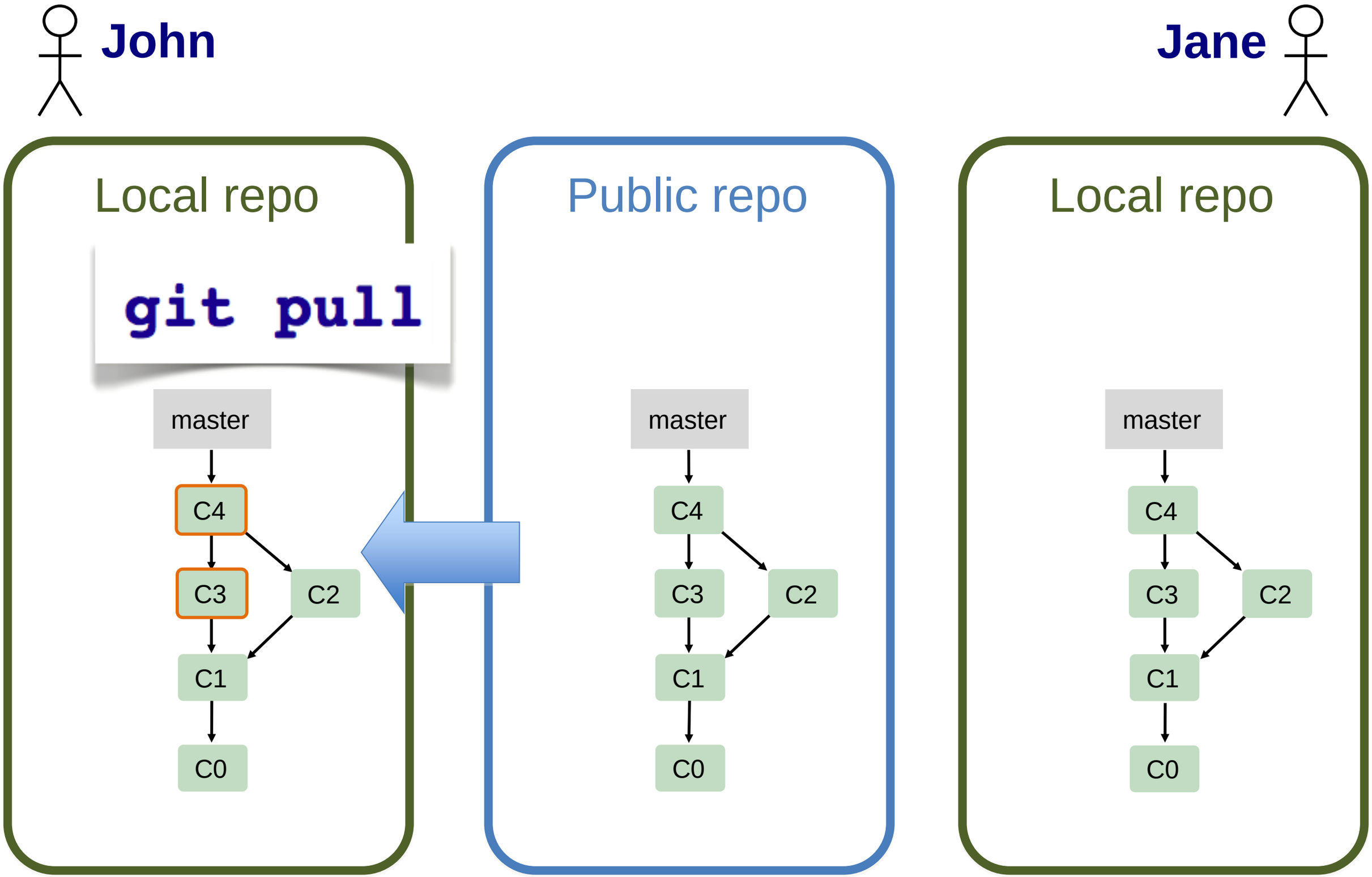

John
Jane
Local repo
Public repo
Local repo
git push
master
C2
C1
C0
master
C2
C1
C0
master
C3
C1
C0


John
Jane
Local repo
Public repo
Local repo
git fetch
master
C2
C1
C0
master
C2
C1
C0
master
C3
C2
C1
C0



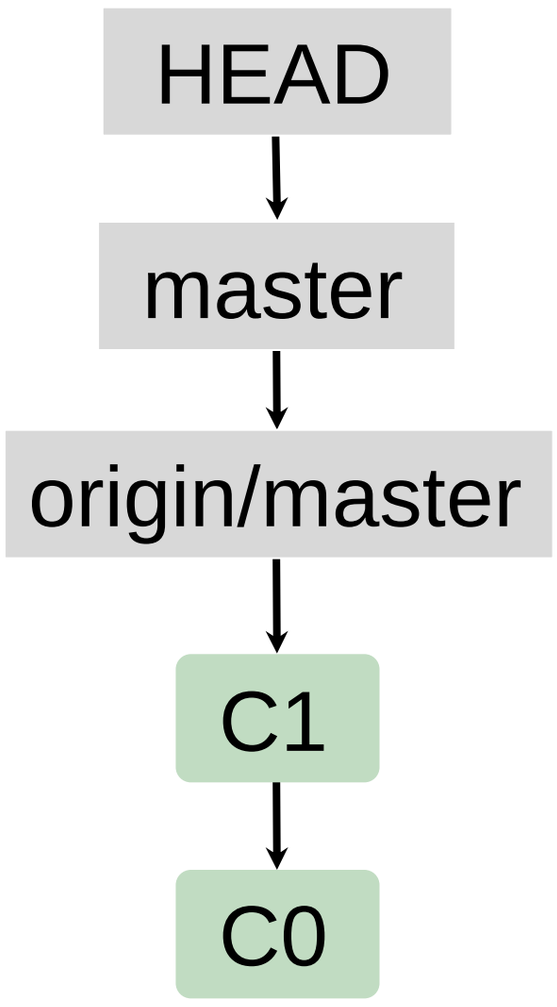
NB: **git pull** = fetch + merge

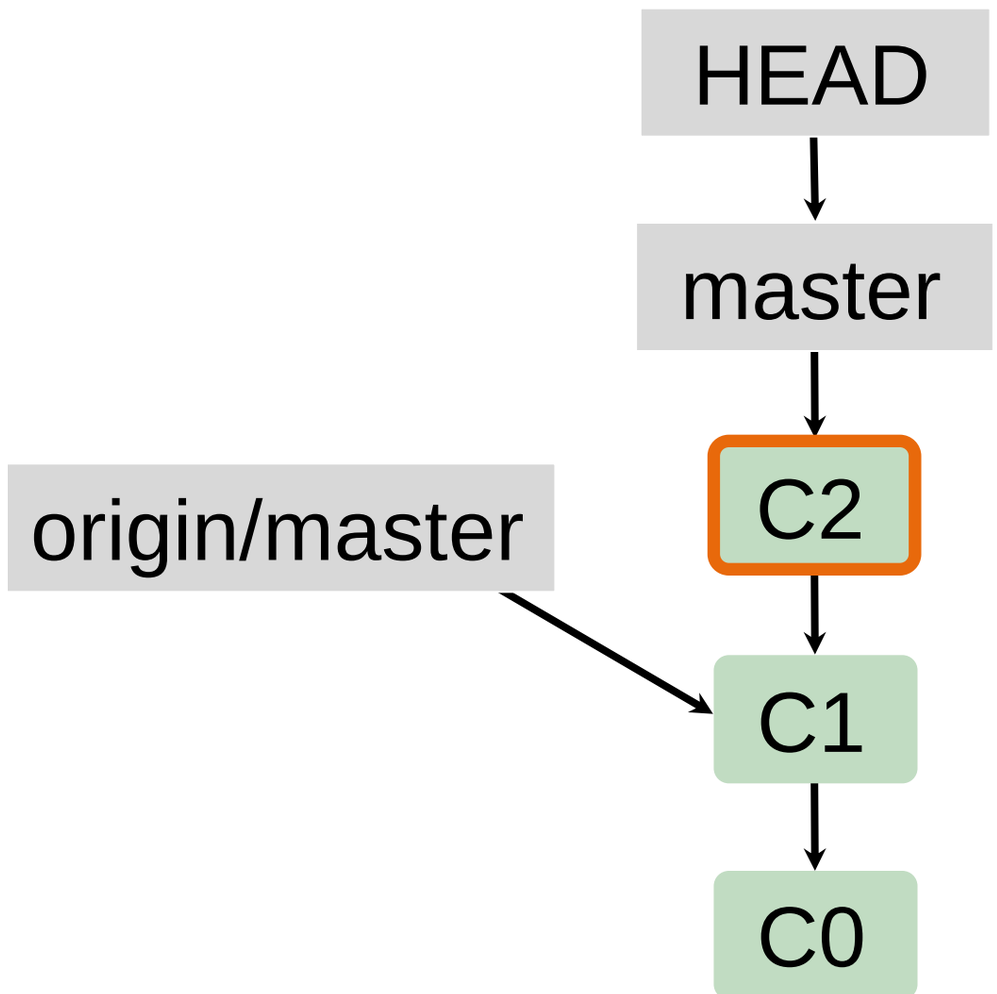

John
Jane
Local repo
Public repo
Local repo
git push
master
C2
C1
C0
master
C4
C3
C2
C1
C0
master
C4
C3
C2
C1
C0


John
Jane
Local repo
git pull
master
C4
C3
C2
C1
C0
Public repo
master
C4
C3
C2
C1
C0
Local repo
master
C4
C3
C2
C1
C0

Branching and merging

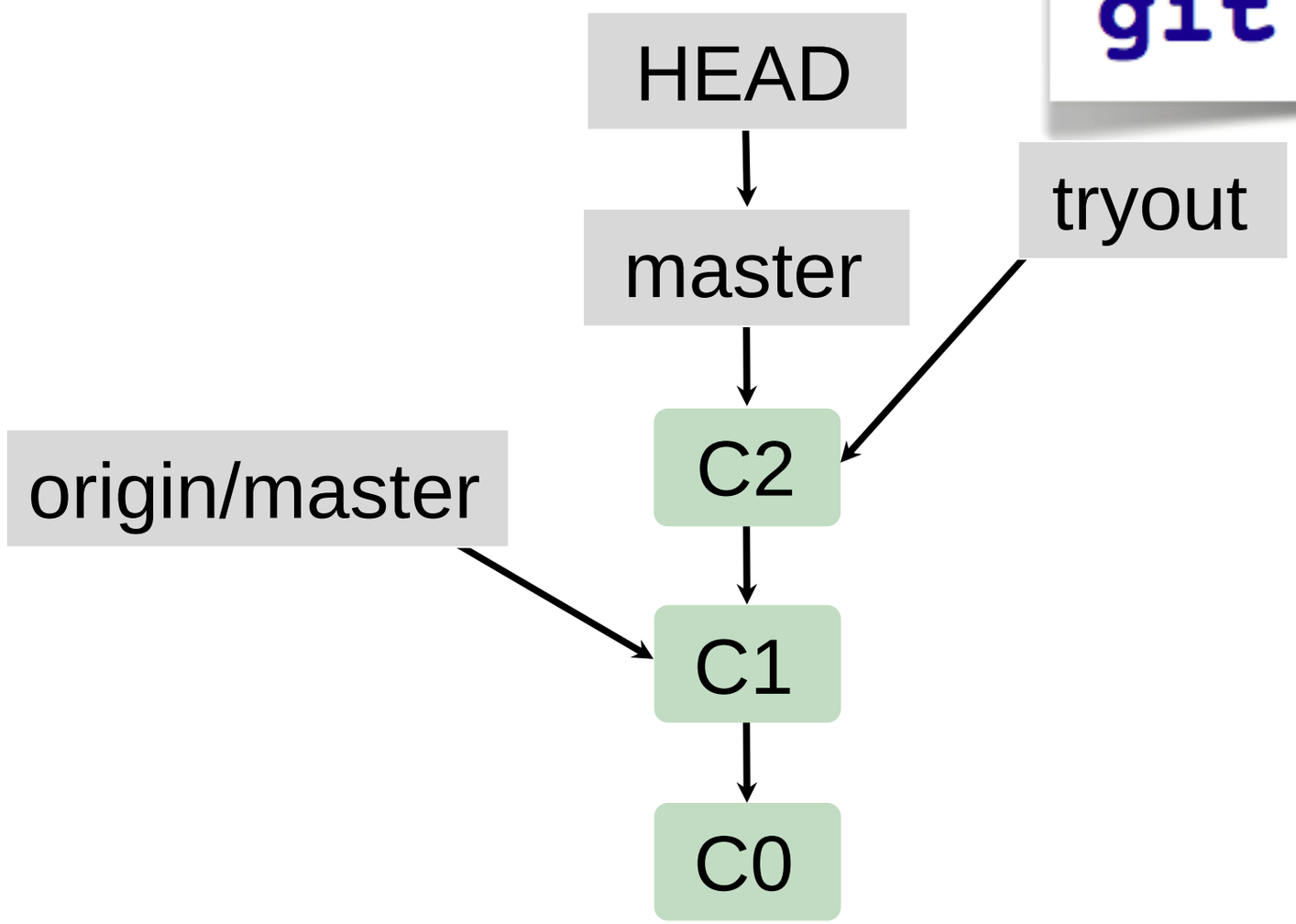
“origin” refers to the remote repo



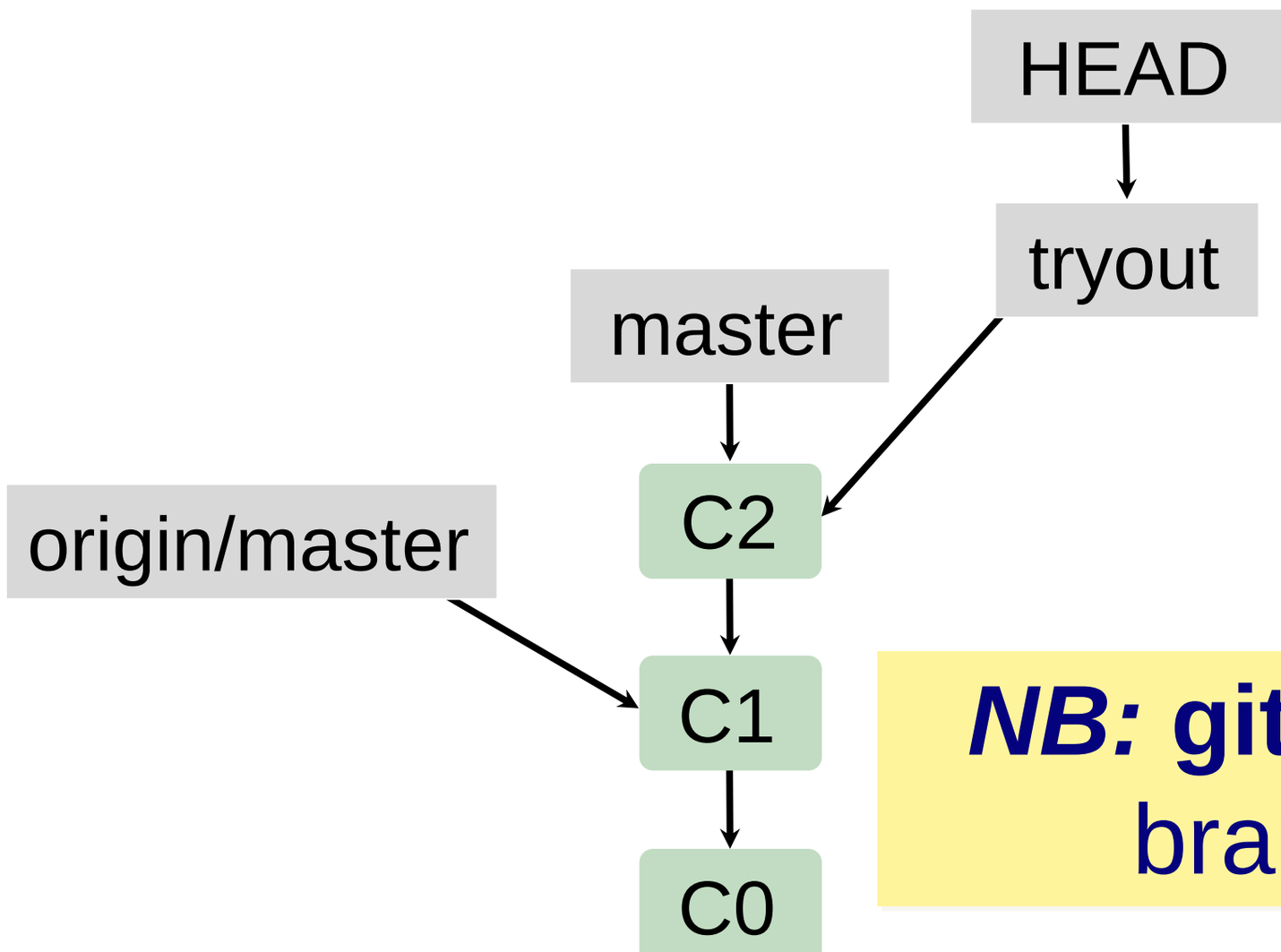



```
…
git commit …
```

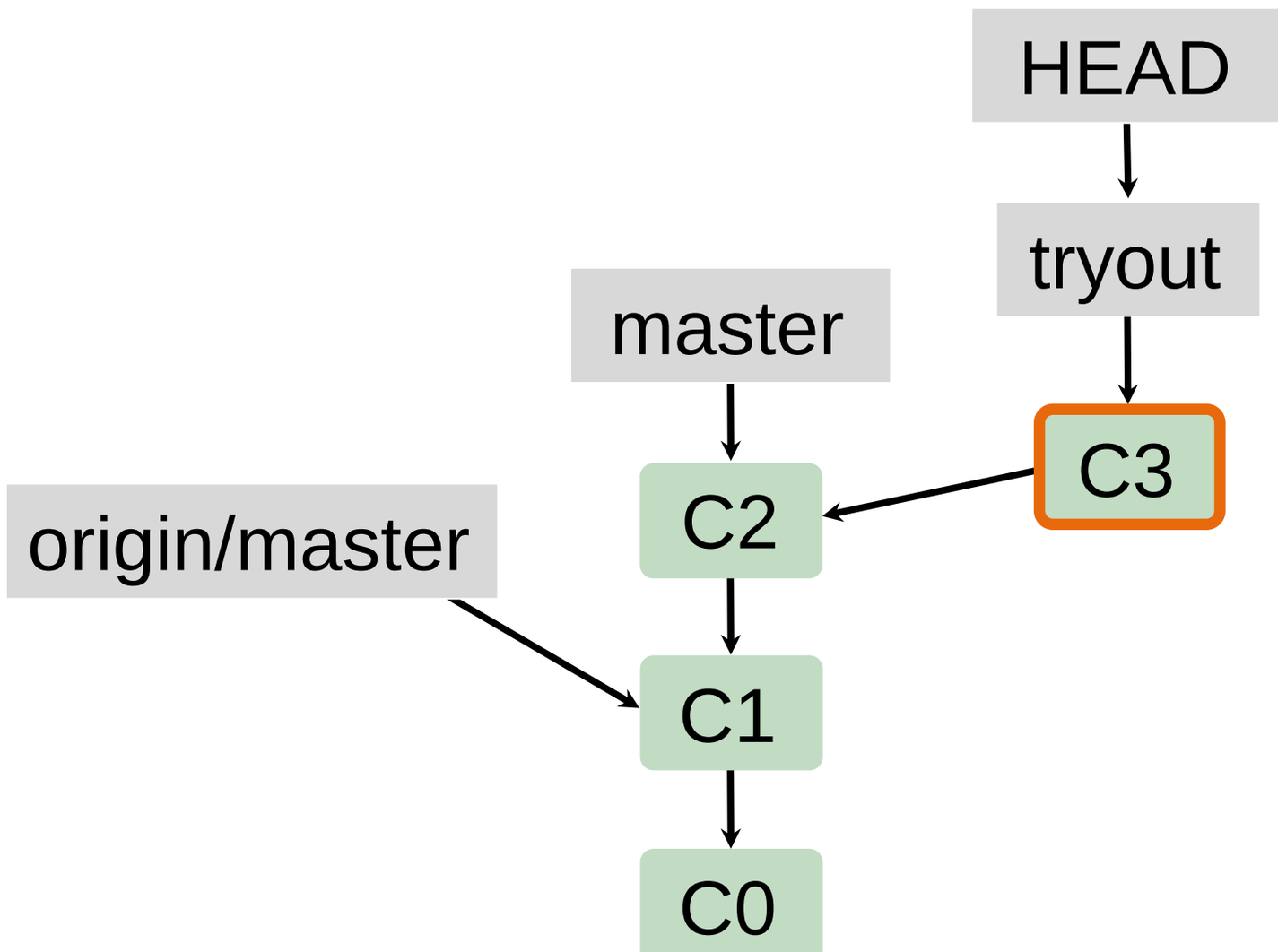
```
git branch tryout
```



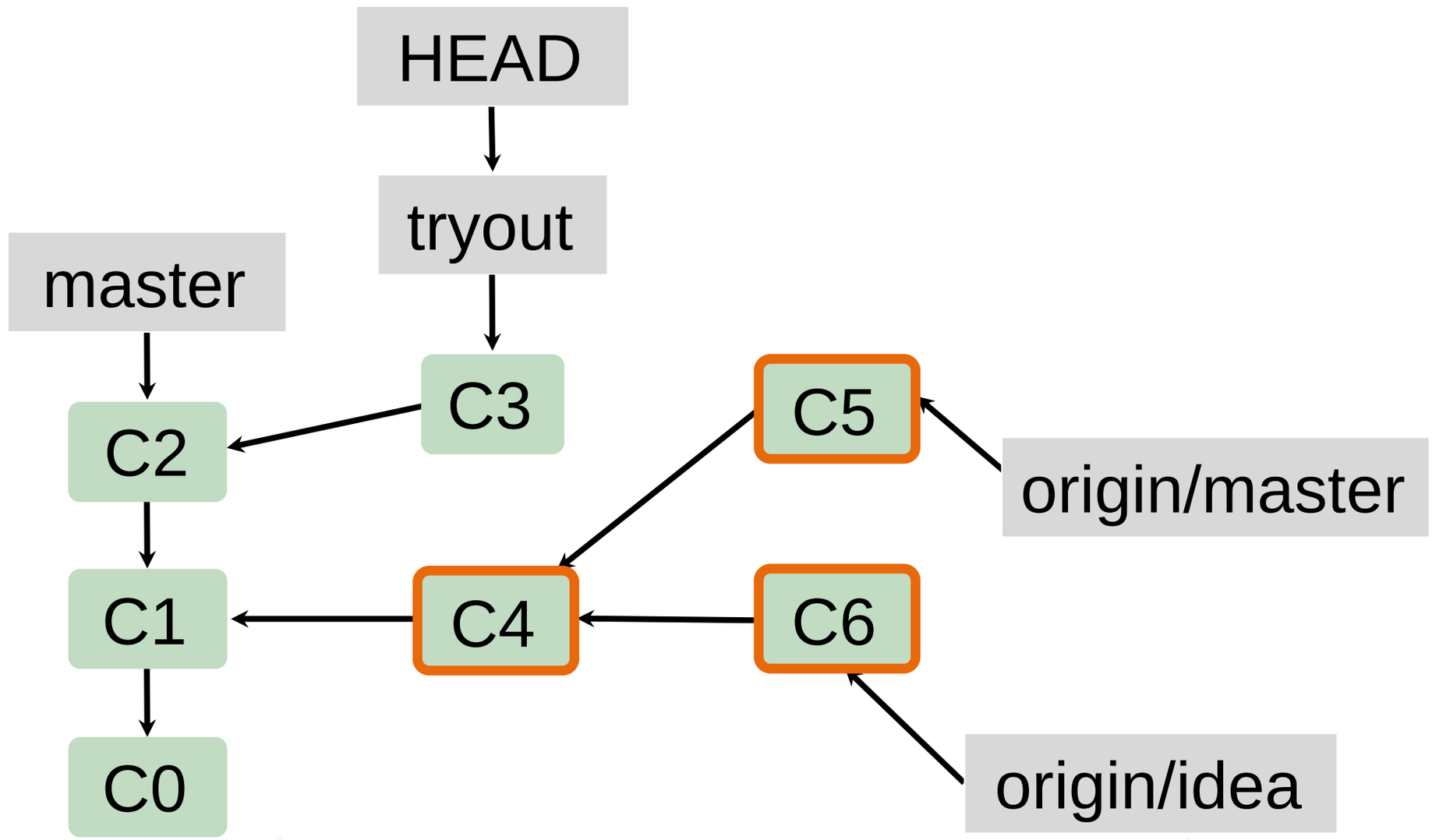
git checkout tryout



***NB:* git checkout –b … =**
branch + checkout

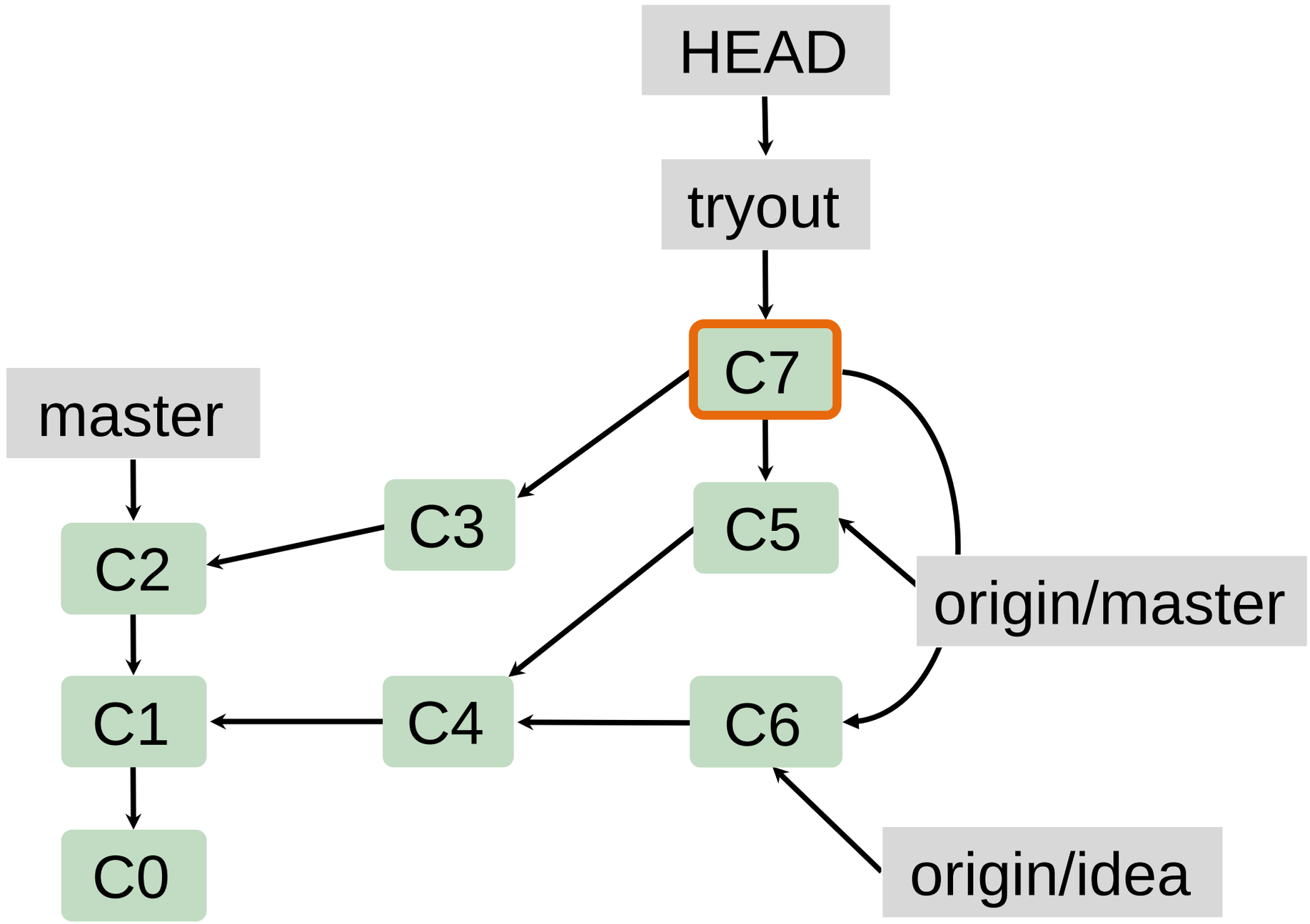



```
git commit …
```

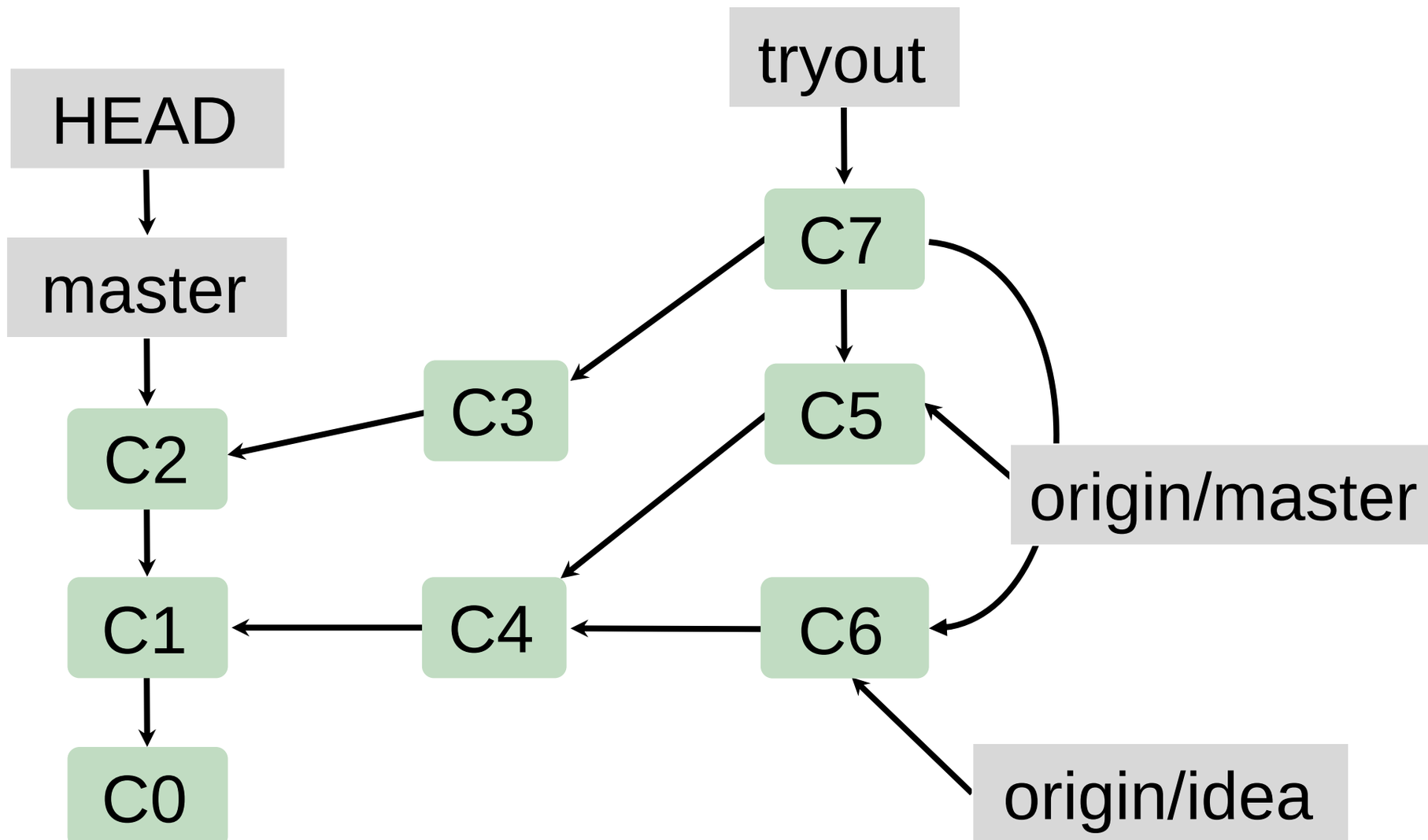



```
git fetch origin
```

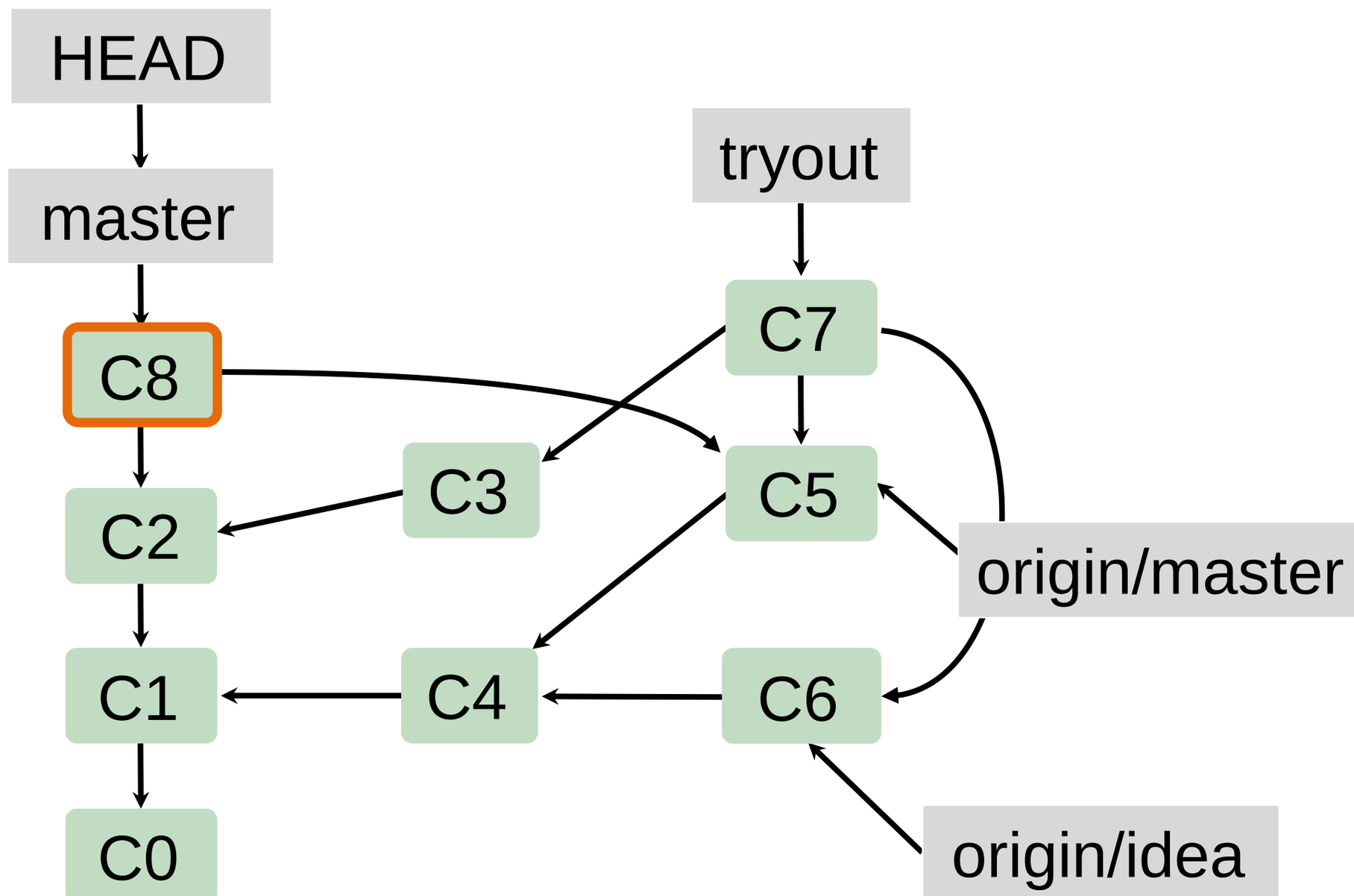
git merge origin/master origin/idea



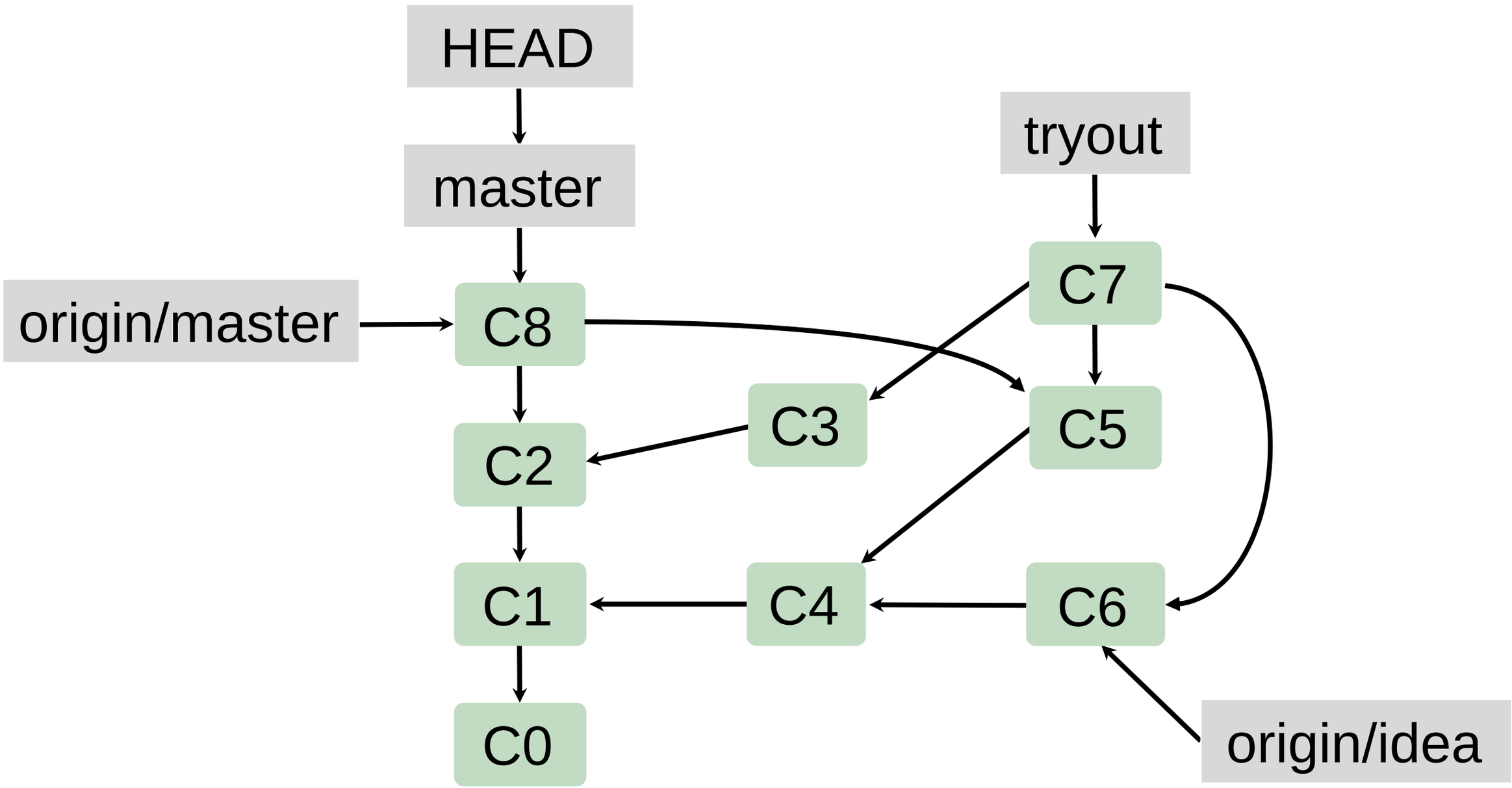
```
git checkout master
```



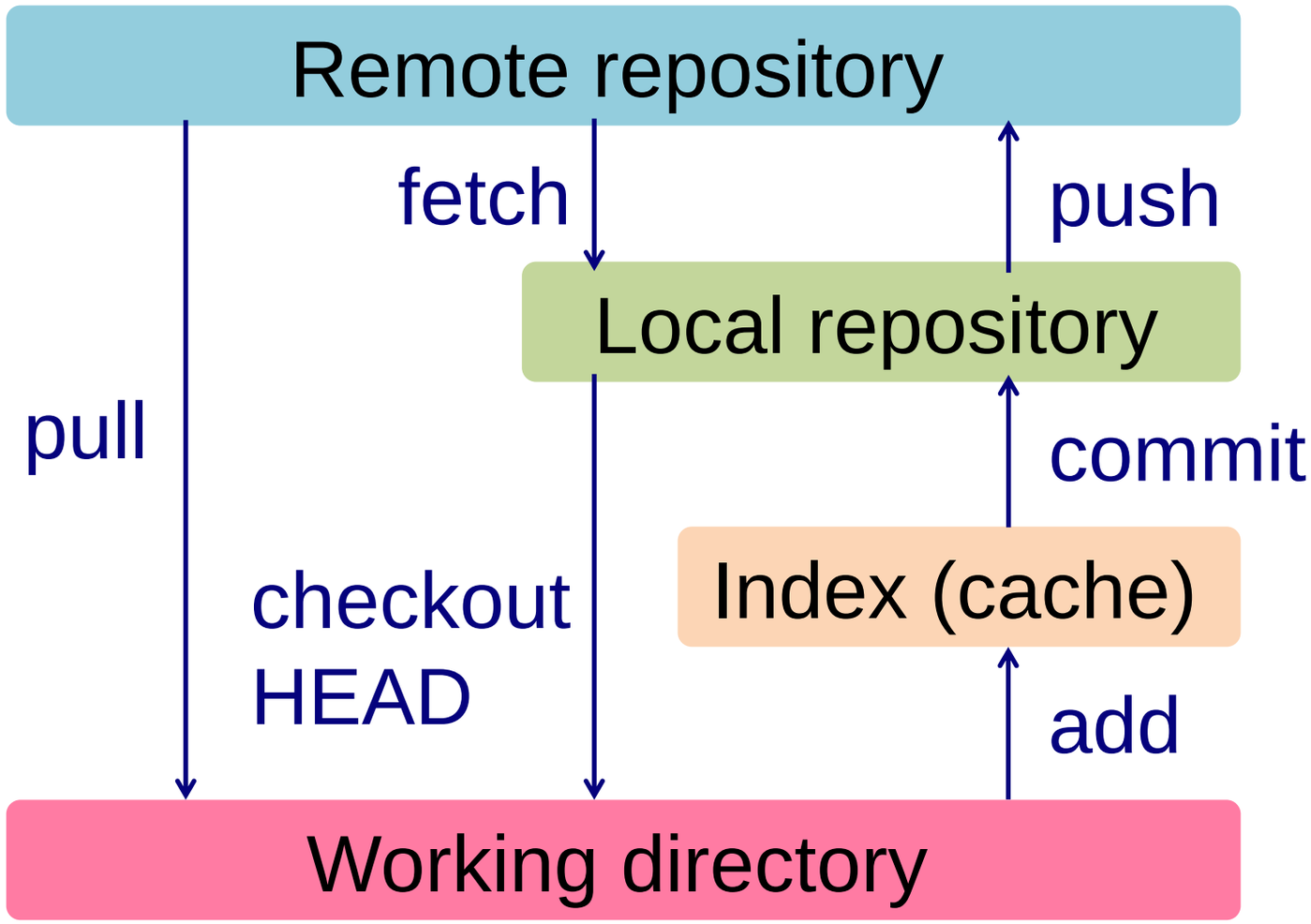
git merge


HEAD
master
tryout
C8
C7
C3
C5
C2
origin/master
C1
C4
C6
C0
origin/idea

```
git push
```


HEAD
master
origin/master
C8
C2
C1
C0
C3
C4
tryout
C7
C5
C6
origin/idea

Overview git workflow



Getting started

First steps

- Follow instructions on P2 Blog for installation
- Send your ssh public key to joel.krebs@students.unibe.ch
- Create meaningful commits with according messages
- Hints that make your life easier:
  - Create a .gitignore file
  - Always pull before you push
  - Don’t panic when merge conflicts occur

More to git

More to git …

- Merging and mergetool
- Squashing commits when merging
- Resolving conflicts
- User authentication with ssh
- gitx and other graphical tools
- git configure — remembering your name
- git remote — multiple remote repos
- github — an open source public repo
- …

Resources

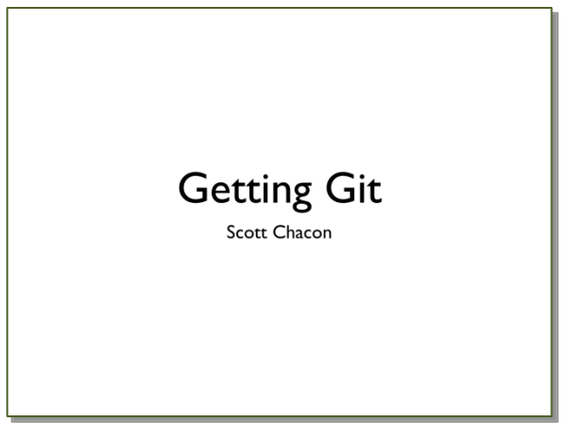
http://git-scm.com/

http://book.git-scm.com/index.html

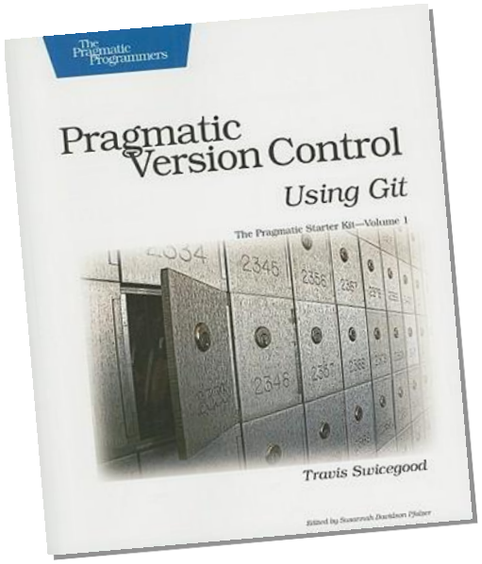
git ready

http://gitready.com/

https://github.com/



http://www.slideshare.net/chacon/getting-git



http://oreilly.com/