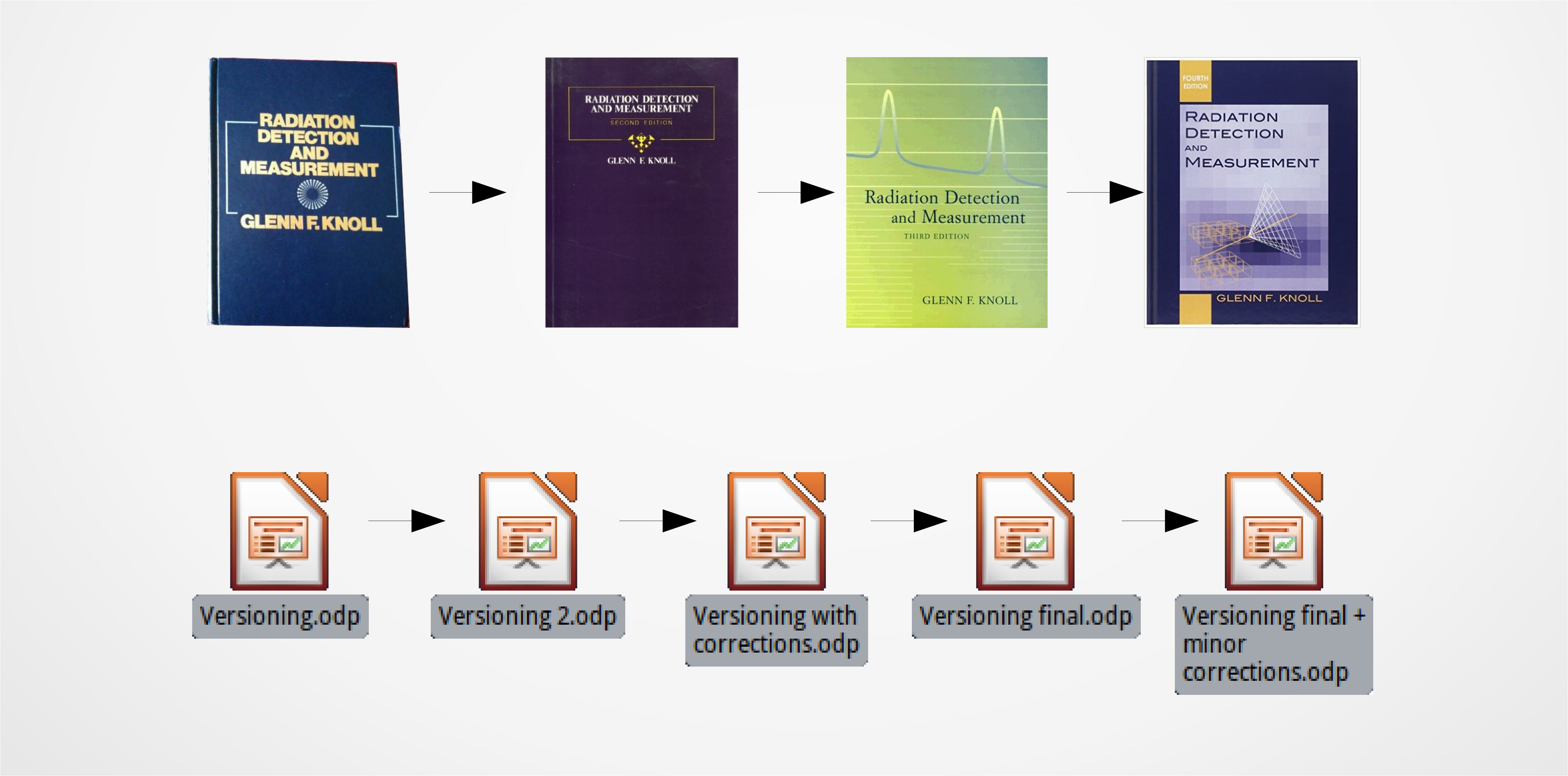
Code Versioning

Olivier Mattelaer (CISM/CP3)

based on slides from
Damien Francois (CISM)
Juan Cabrera (NAMUR)
Jonathan Lambrechts (IMMC)
Scott Chalcon (git)

What is code versioning

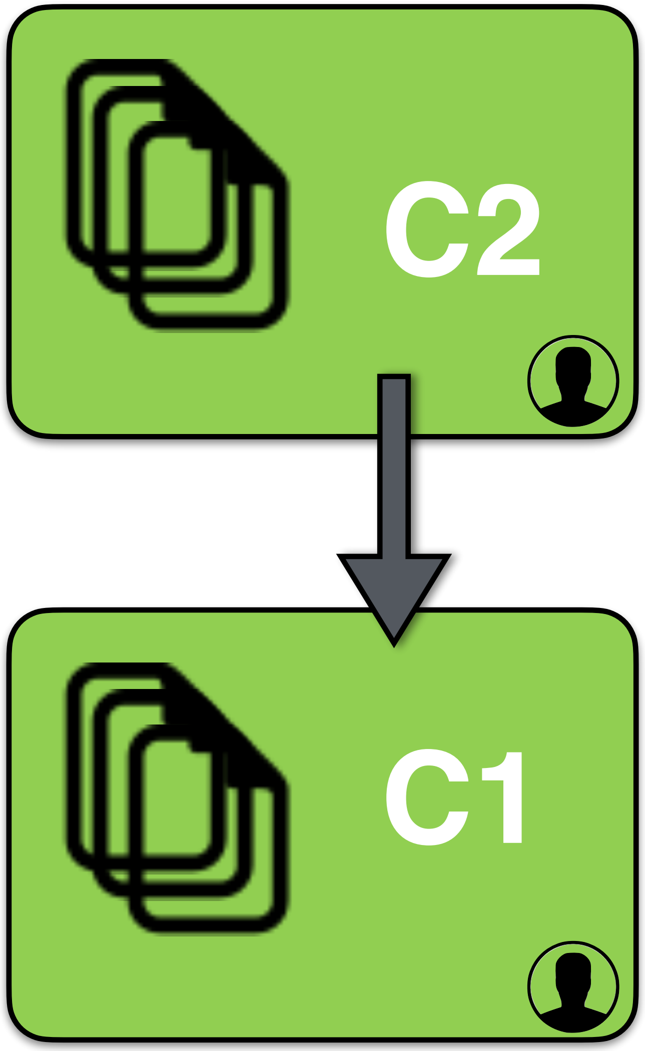

RADIATION DETECTION AND MEASUREMENT
GLENN F. KNOLL
RADIATION DETECTION AND MEASUREMENT
SECOND EDITION
GLENN F. KNOLL
Radiation Detection and Measurement
THIRD EDITION
GLENN F. KNOLL
FOURTH EDITION
RADIATION DETECTION AND MEASUREMENT
GLENN F. KNOLL
Versioning.odp
Versioning 2.odp
Versioning with corrections.odp
Versioning final.odp
Versioning final + minor corrections.odp

Goal of code versioning

1. History of modification
    1. Mission 1->6
2. Workflow
    1. Mission 7->11
3. Team Work
    1. Mission 12-> 16

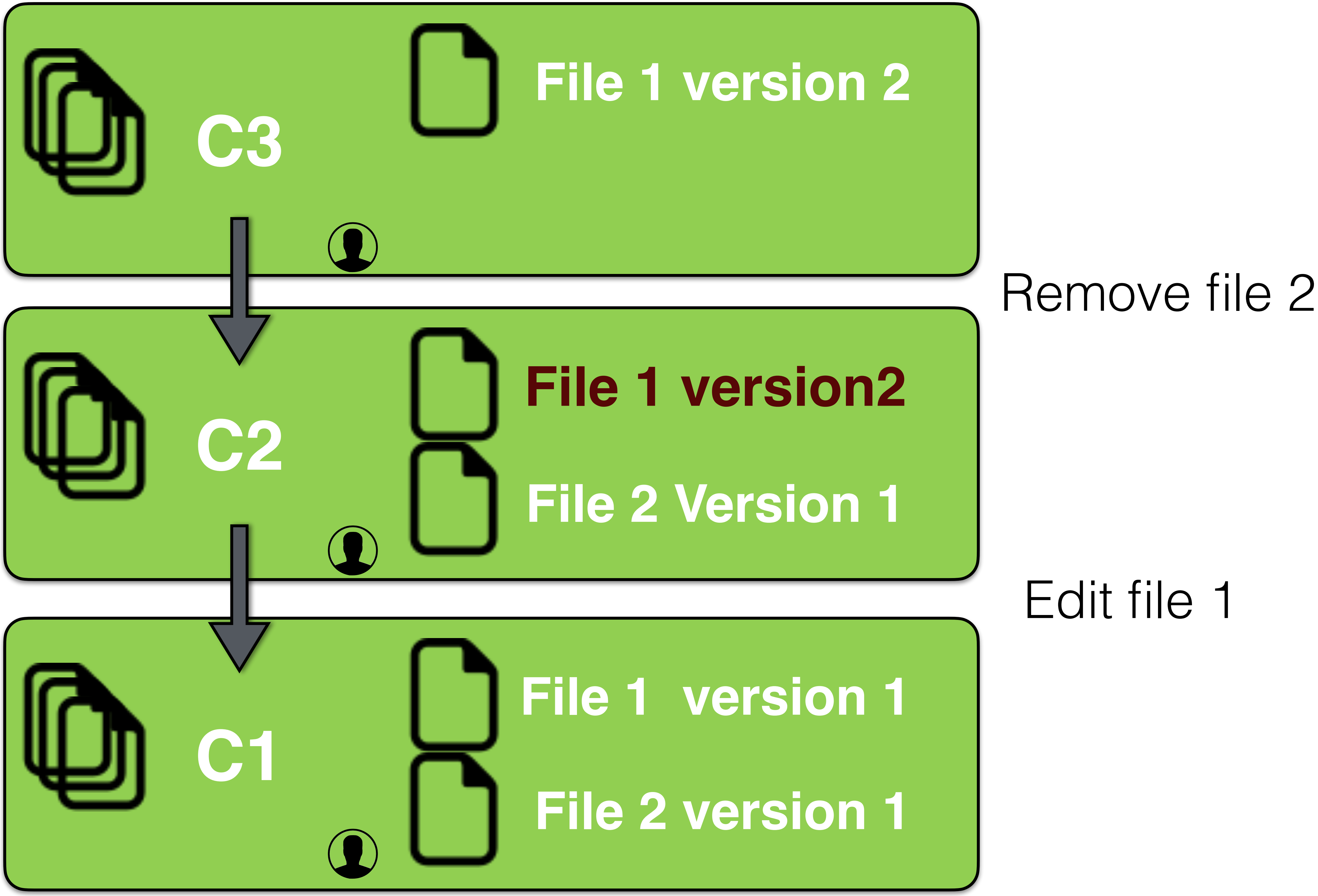
1. Commit

- An history: Is a **succession** of **snapshot** of your files at key time of their development
  - Each **snapshot** is called **COMMIT**



- Commit is
  - All the files at a given time
  - A unique name (SLHA1)
  - MetaData (who created/when/info)
  - Pointer to previous(es) commit

1. Commit



C3

File 1 version 2

Remove file 2

C2

File 1 version2

File 2 Version 1

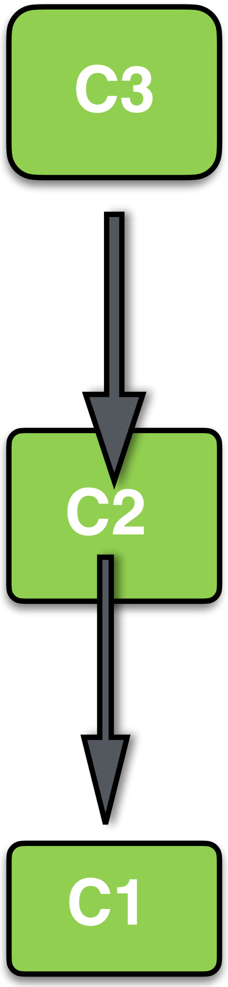
Edit file 1

C1

File 1 version 1

File 2 version 1

1. Commit



1. Simplify representation of commit/history

Git Three area

Workspace

./WORKDIR

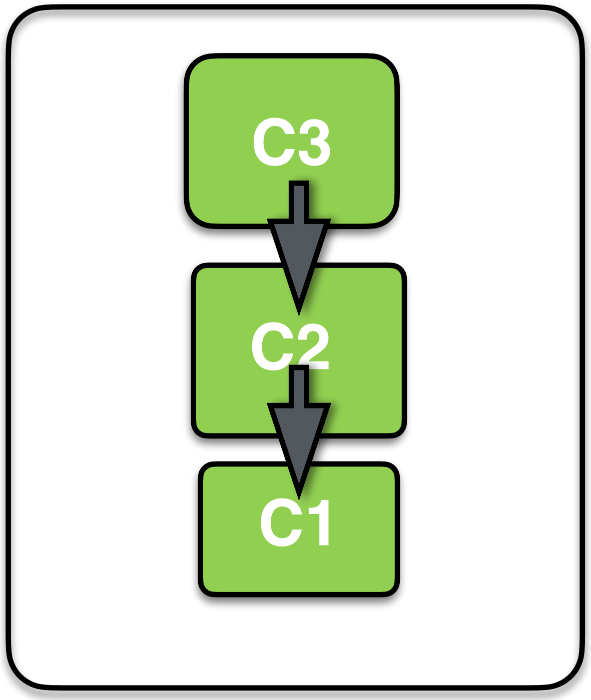
Index

.git/index

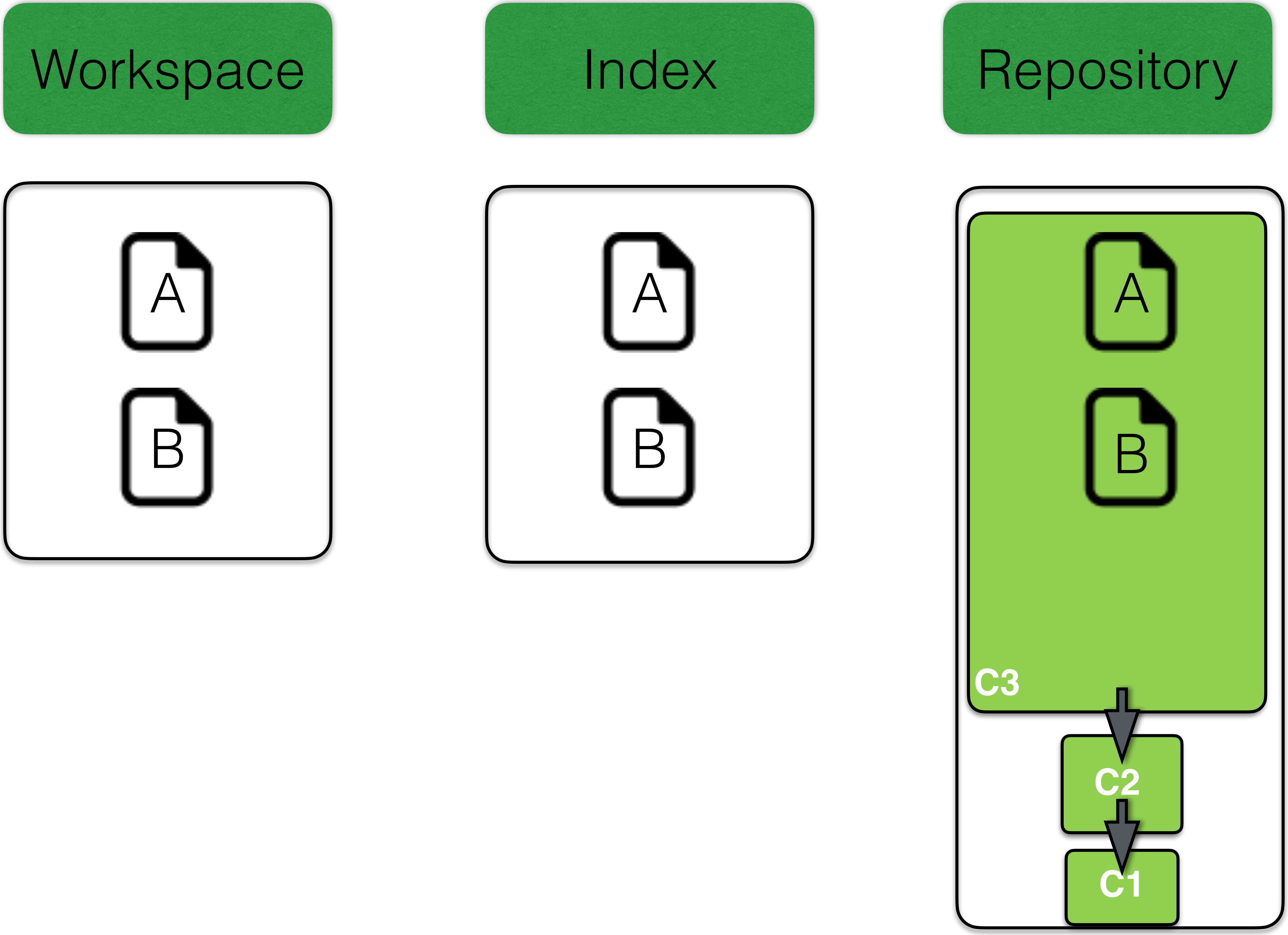
Staging area

Repository

.git/



Git Three area

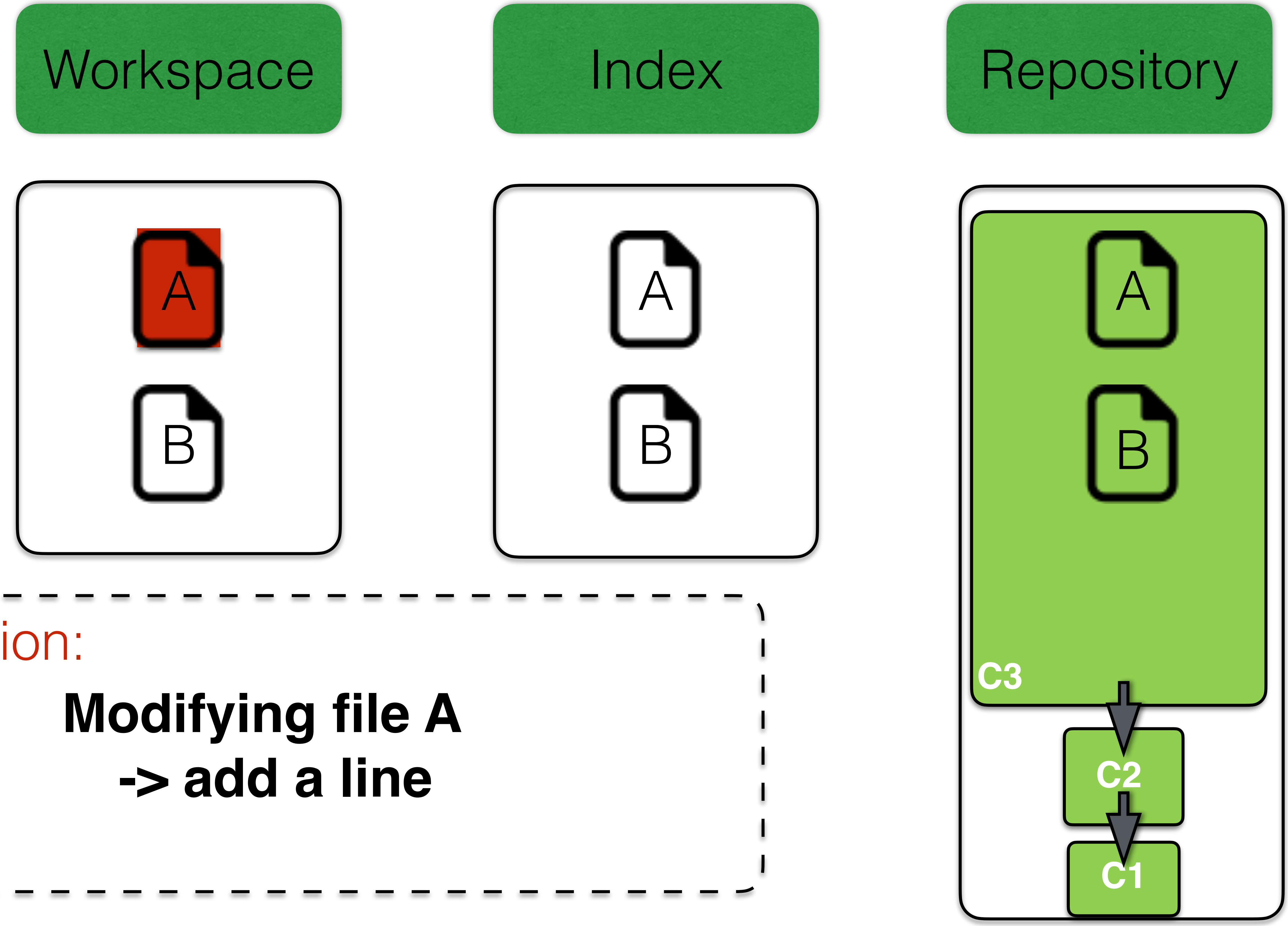


Workspace

Index

Repository

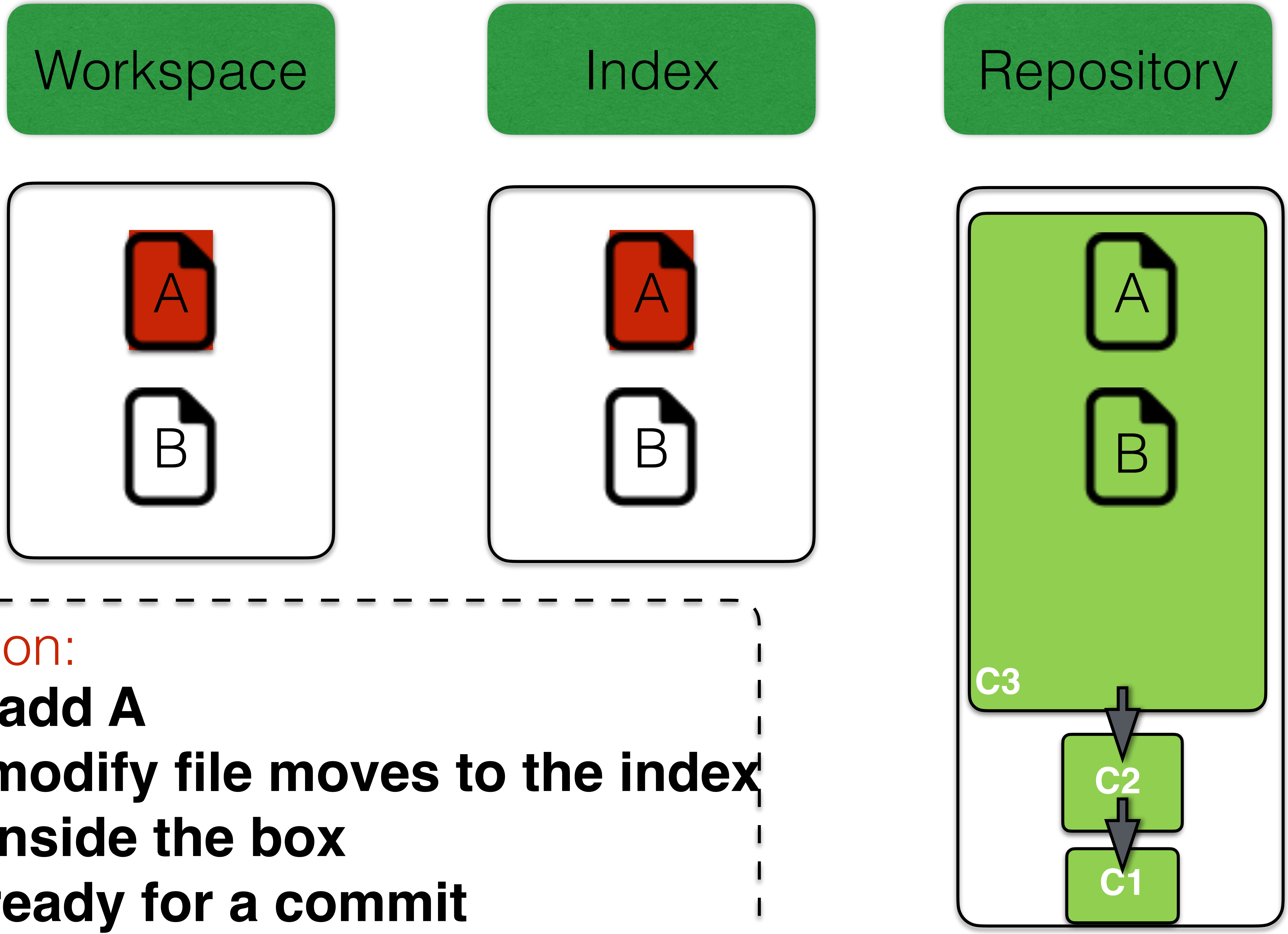
Git Three area



Action:

Modifying file A
-> add a line

Git Three area



Action:

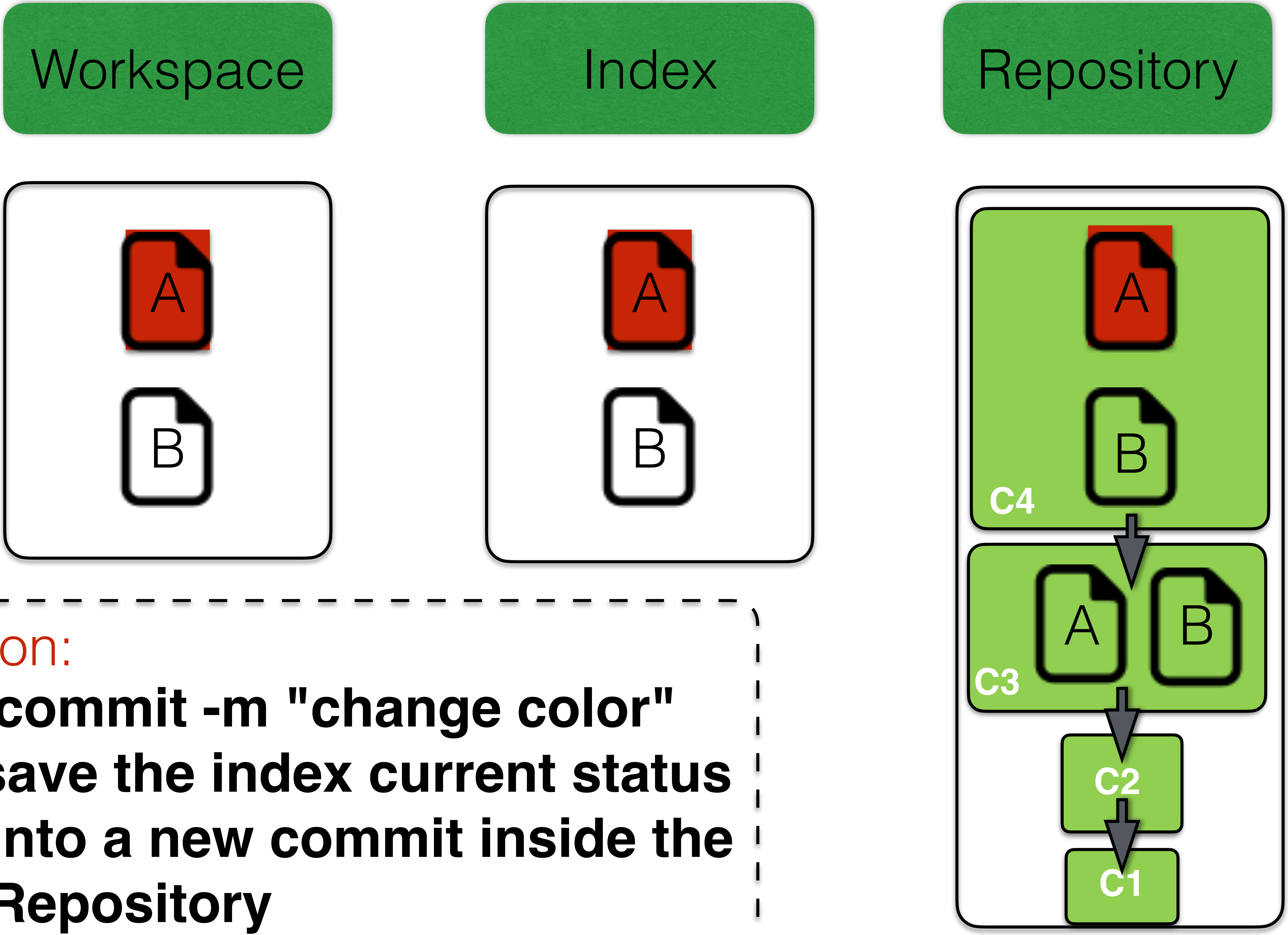
git add A

-> modify file moves to the index

-> inside the box

-> ready for a commit

Git Three area



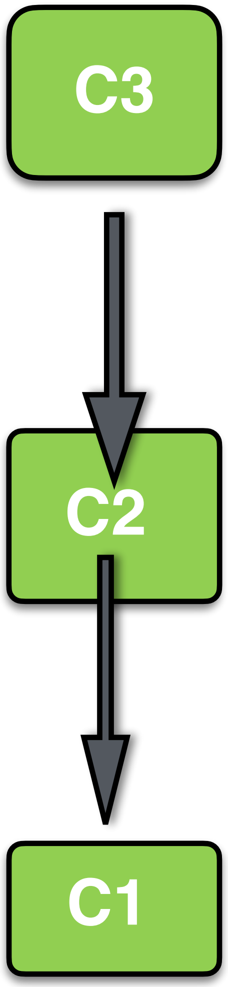
Action:

git commit -m "change color"

-> save the index current status Into a new commit inside the Repository

1. Commit

Head: place where the new commit will be attach



Head: latest committed version

Head~1

Head~2

Let's play

Game installed on lemaitre3 and manneback
module load gitshell

Basic command of the game
gsh goal
gsh check

Messed a level: restart it with
gsh reset

Exercise #1
Stop at mission #7
You have ~45 min

Workflow

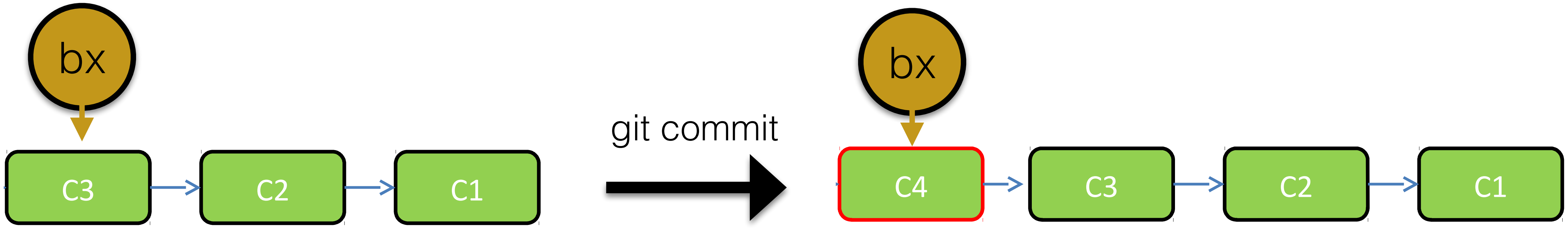
branch in git

- Branch is **pointer** to a commit (represent an history) which represent a **line of development**
- A branch can point at other commit, it **can move**!
- A branch is a way to organise your work and working histories

branches

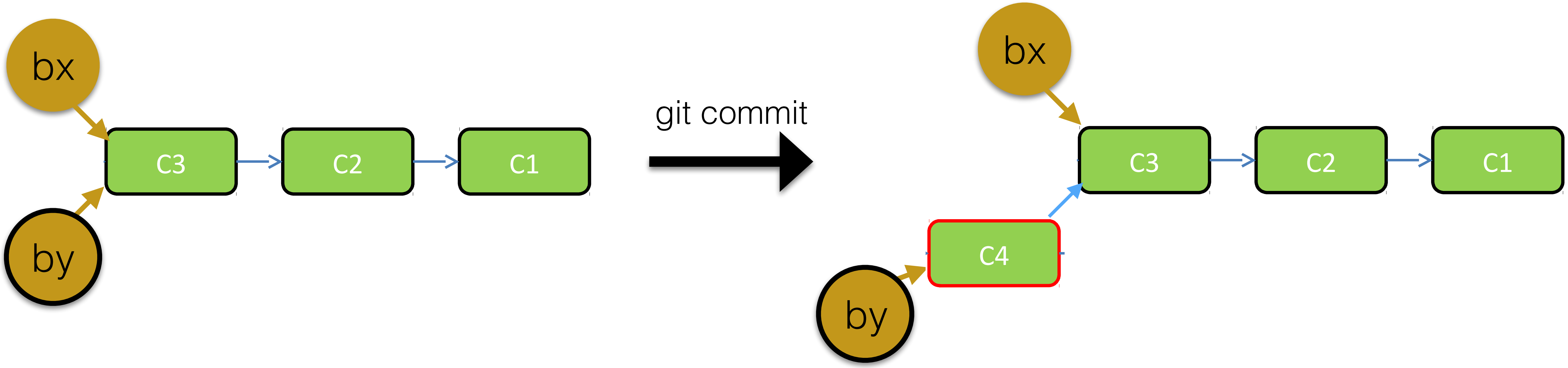
- default branch: master (or main)

- When doing a commit, the branch moves to the new commit



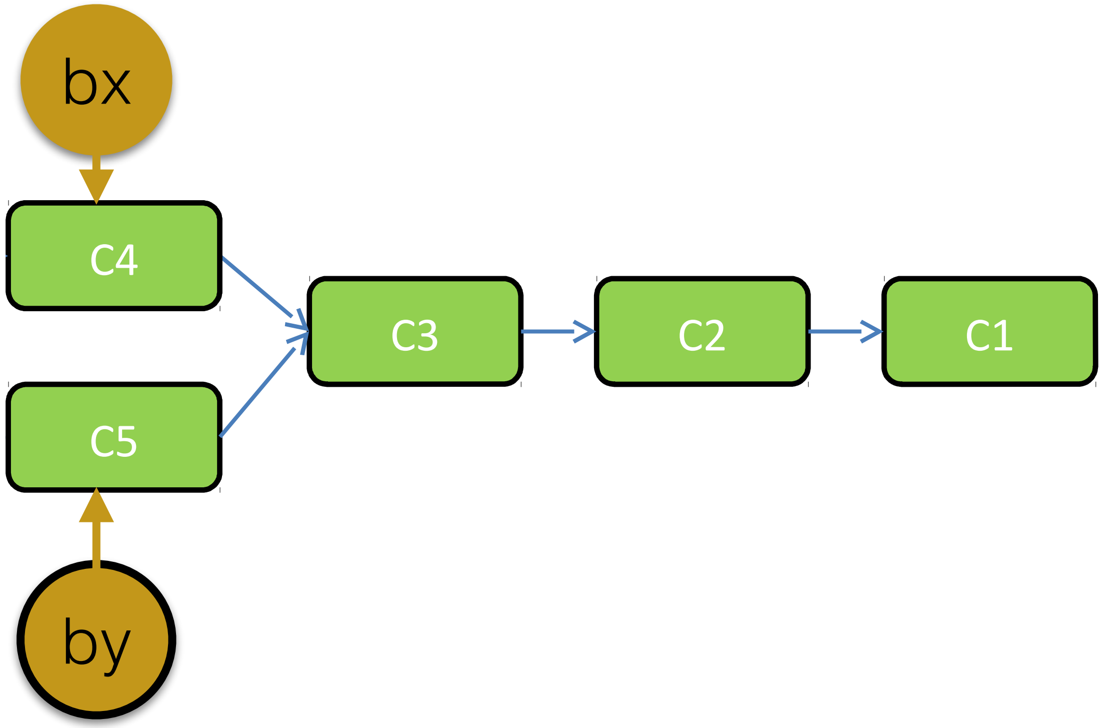
- creating a new branch: add a pointer (git switch -c by)

  - only selected branch affected by commit!



merging

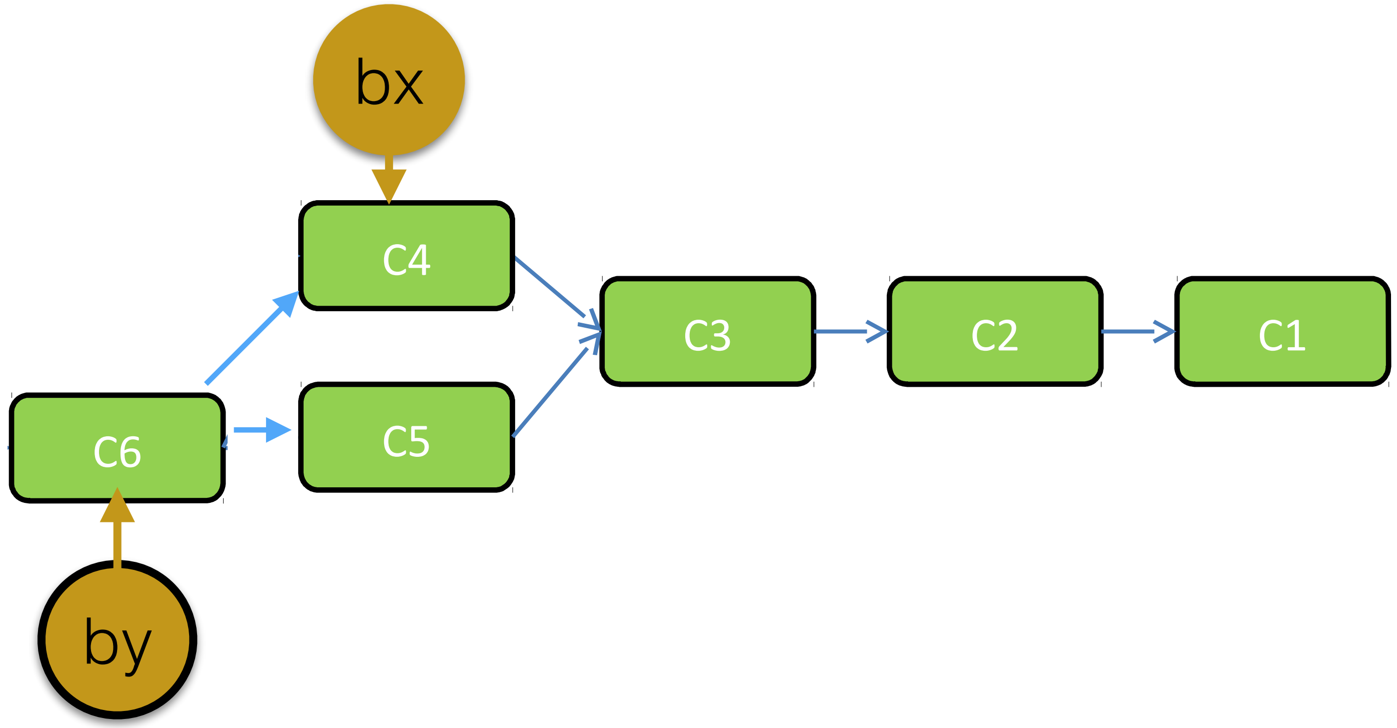
- The interest of branch is that you can **merge** them
    - Include in one (branch) file the modification done somewhere else



git merge bx

merging

- merging two different modifications



git merge bx

Do it yourself

- Mission #7-13

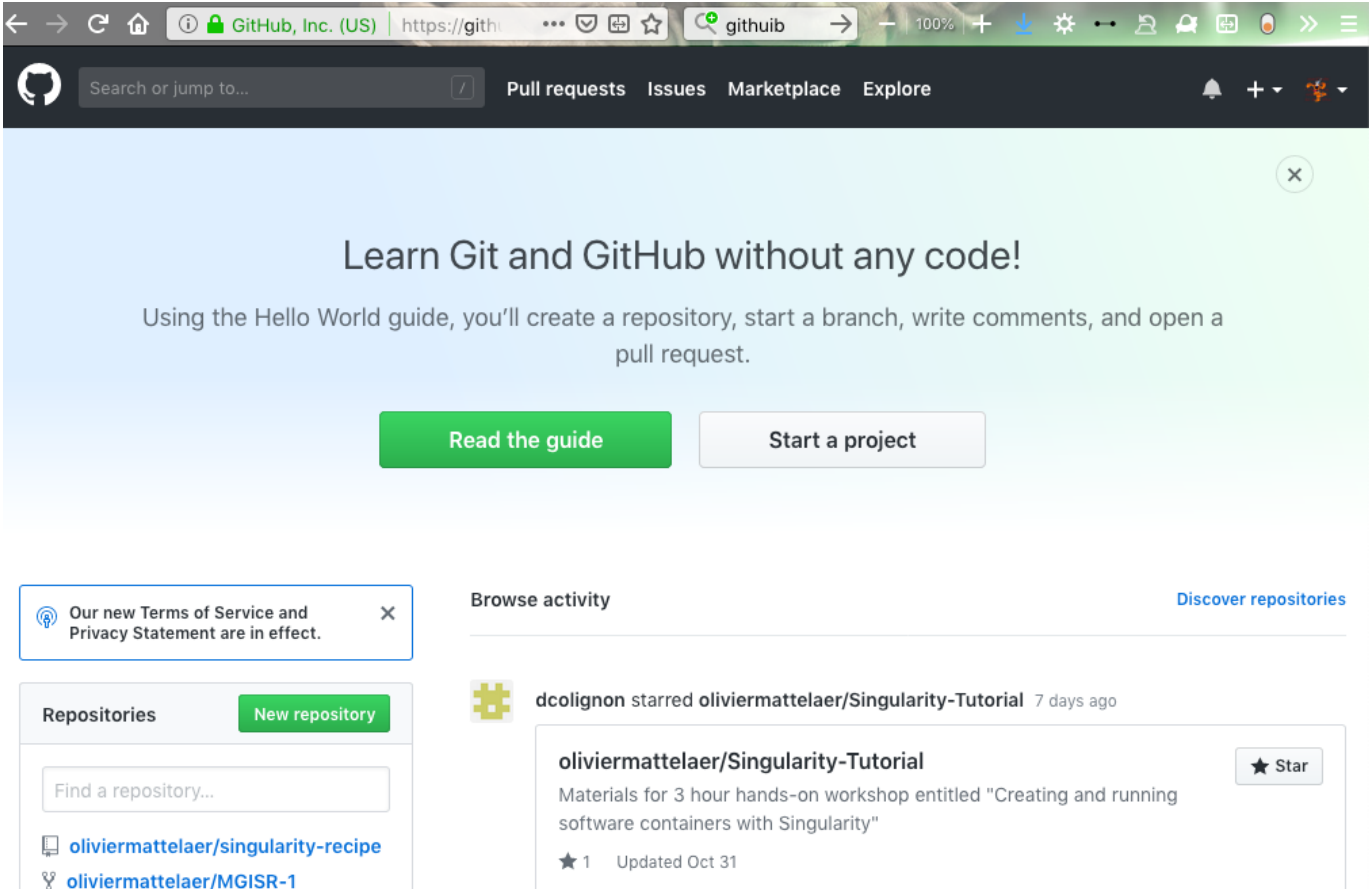
Conflict

- **Multiple version of files** are great
  - Not always easy to know how to merge them
  - Conflict will happen (same line modify by both user)

- Conflict need to be resolved manually!
  - Boring task
  - need to understand why a conflict is present!
- **Do not be afraid of conflict!** Do not try to avoid them at all cost!
- stay in sync as most as possible and keep line short

Team Work

GitHub/Gitlab



Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Our new Terms of Service and Privacy Statement are in effect.

oliviermattelaer/singularity-recipe

oliviermattelaer/MGISR-1

Browse activity

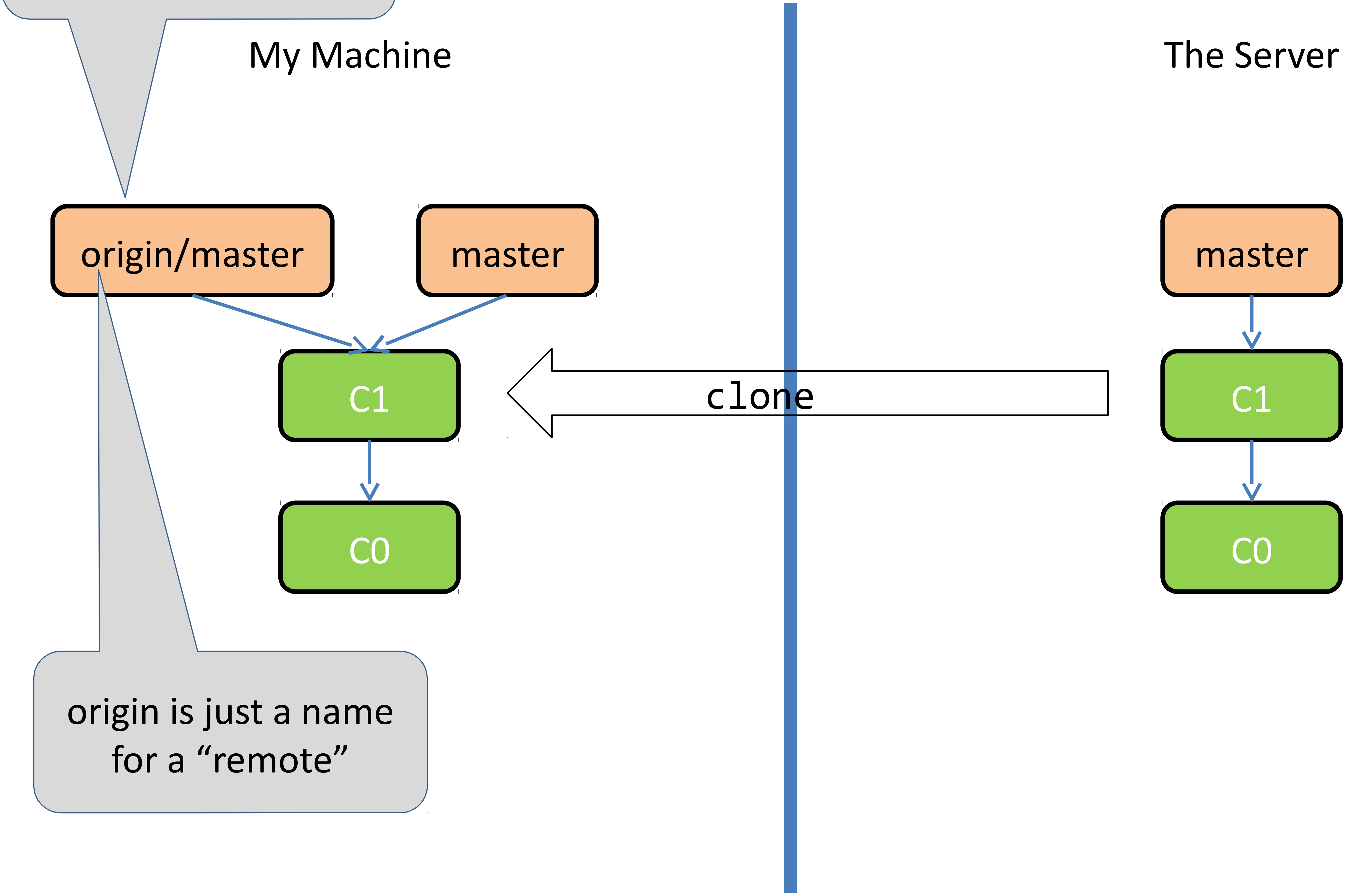
dcolignon starred oliviermattelaer/Singularity-Tutorial 7 days ago

oliviermattelaer/Singularity-Tutorial

Materials for 3 hour hands-on workshop entitled "Creating and running software containers with Singularity"

★ 1 Updated Oct 31

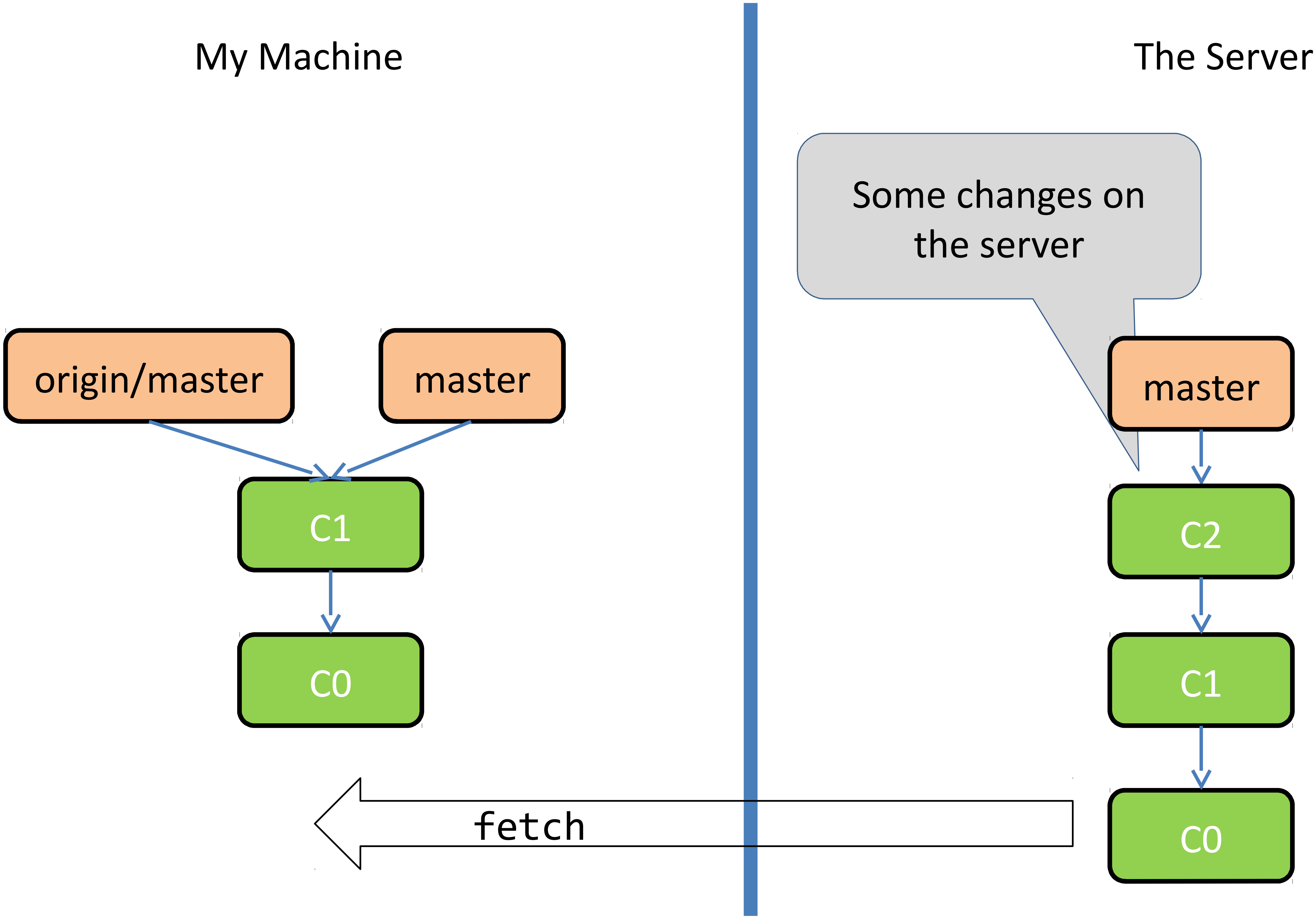
Remote Branches



This is a remote branch

origin is just a name for a “remote”

Remote Branches - fetch

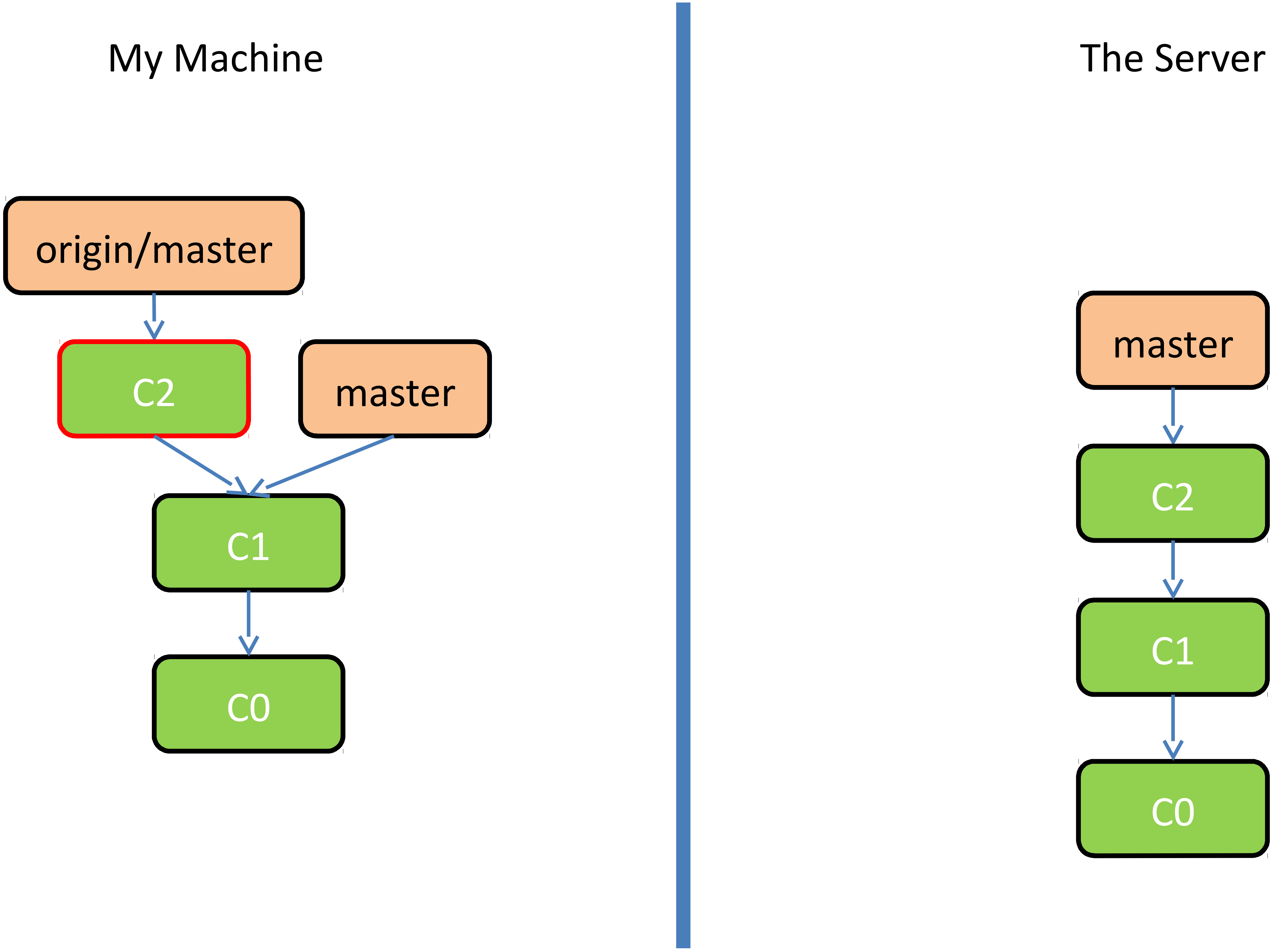


My Machine

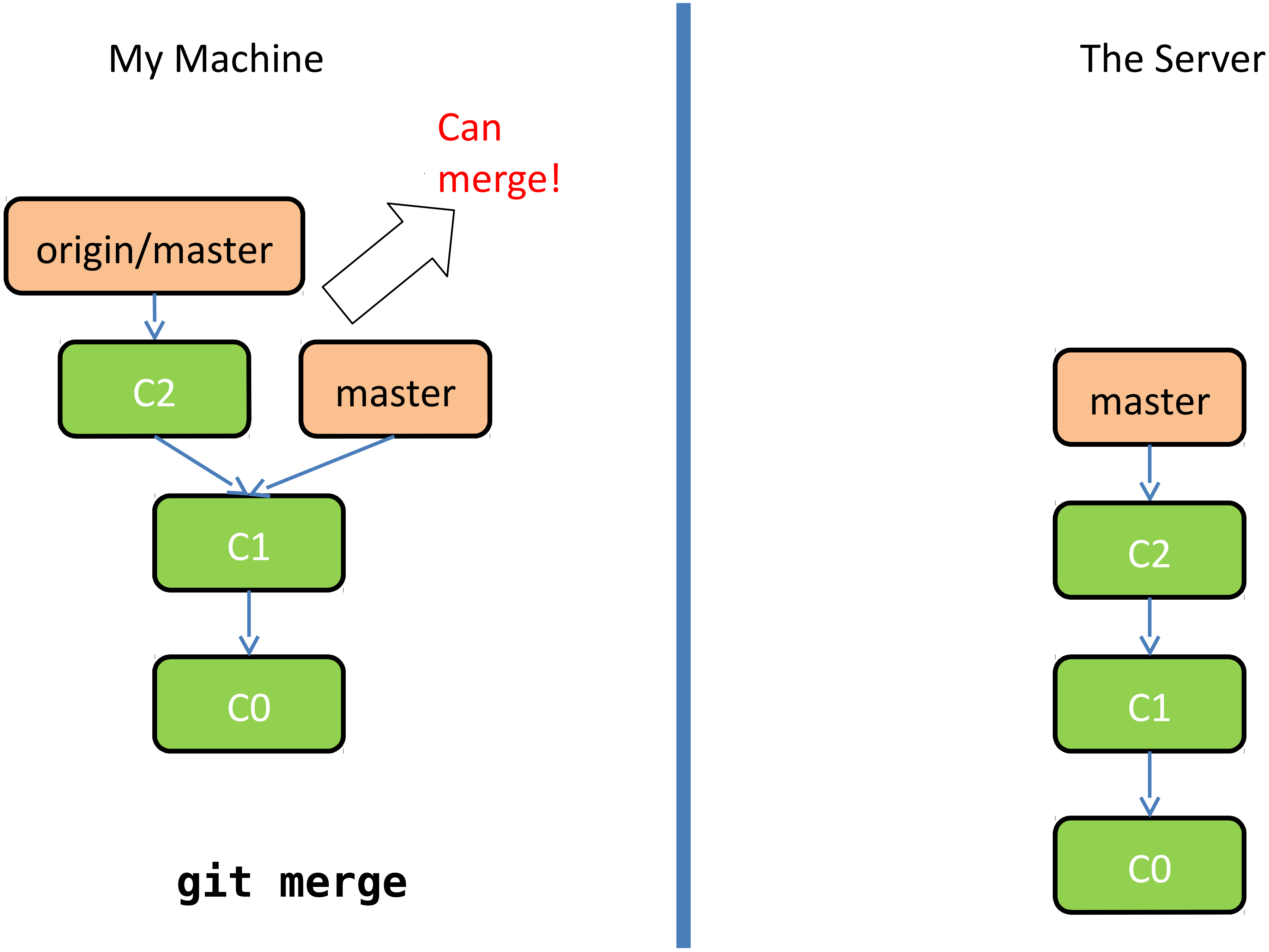
The Server

Some changes on the server

Remote Branches - fetch



Remote Branches - fetch



My Machine

The Server

Can merge!

origin/master

C2

master

C1

C0

master

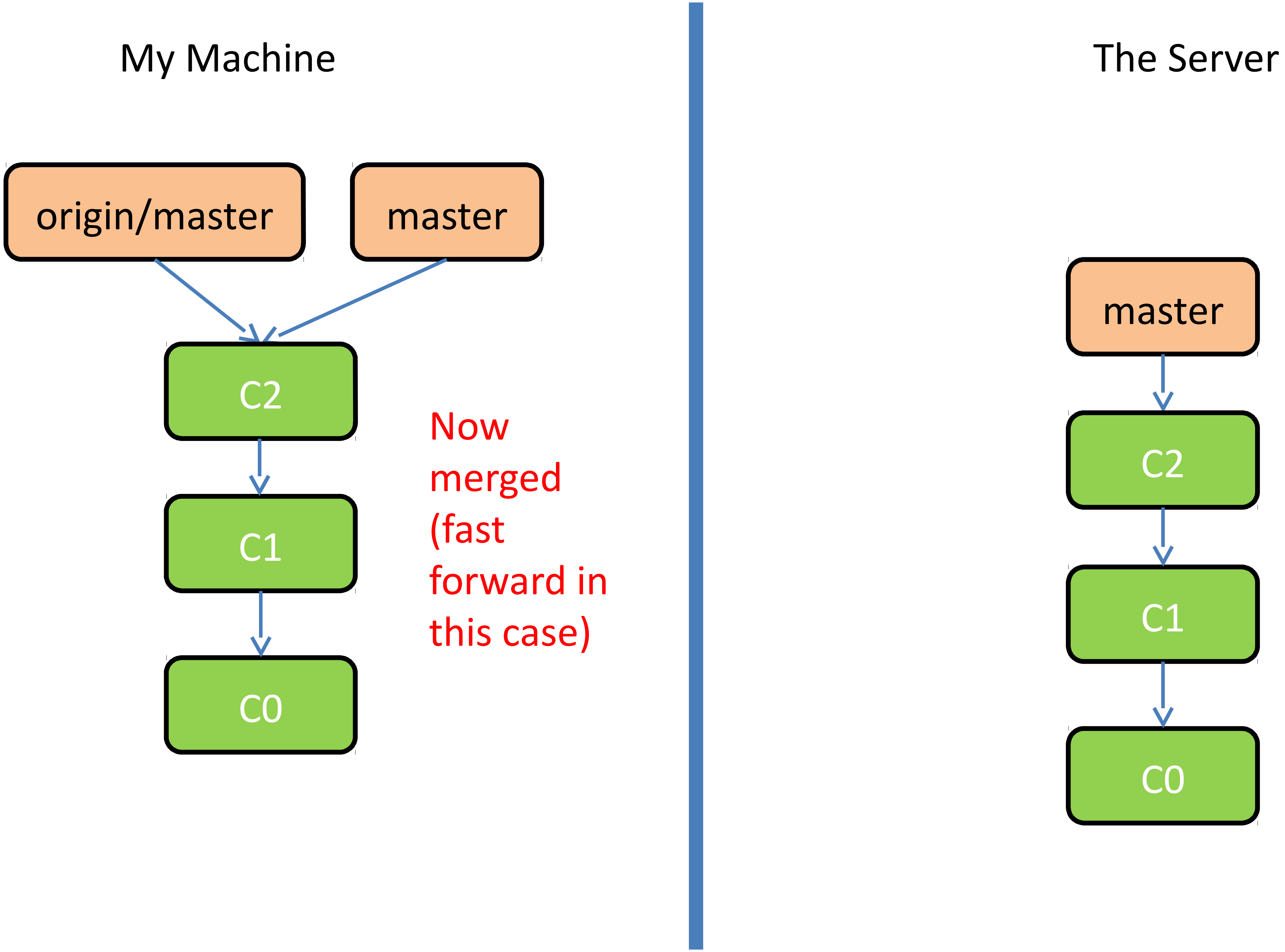
C2

C1

C0

```
git merge
```

Remote Branches - fetch



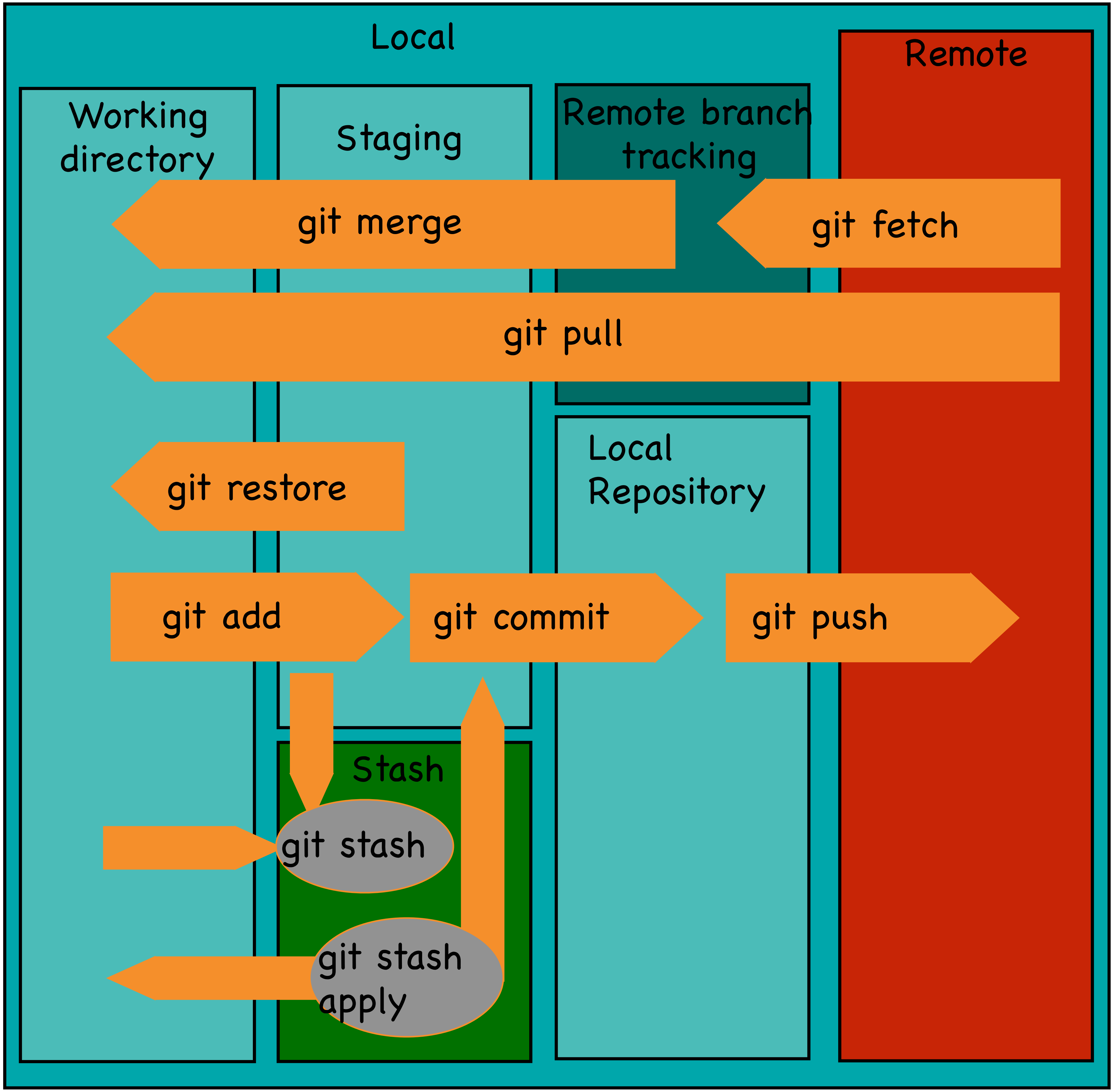
Now merged (fast forward in this case)

Remote Branches

- Reminder - Remote branches represent a branch on a remote repository
- The branch origin/master for example is a local pointer to the “master” on “origin”
- It reflects what the **local** repository **currently knows** about the state of “master” on “origin”

Do it yourself

- Mission #12-16


Local
Remote
Working directory
Staging
Remote branch tracking
git merge
git fetch
git pull
Local Repository
git restore
git add
git commit
git push
Stash
git stash
git stash apply

Information

- On cluster, use "module load git"
  - The command you learned need version 2.22
- Be careful with "restore" (no safety net)
- Restrict yourself to edit the history of commit (especially when shared)
  - Git reset # especially with --hard
  - Git rebase (only useful case is git pull --rebase)

Conclusion

- Versioning is crucial both for small/large project
  - Avoid dropbox for paper / project
- make meaningful commit
  - logical block
  - meaningful message
- Be safe avoid

More information

- Why an index: http://gitolite.com/uses-of-index.html
- technical tutorial on git (details on storage structure): https://www.youtube.com/watch?v=xbLVvrb2-fY
- https://git-scm.com/doc