

Code Versioning

Olivier Mattelaer (CISM/CP3)

based on slides from

Damien Francois (CISM)

Juan Cabrera (UNAMUR)

Jonathan Lambrechts (IMMC)

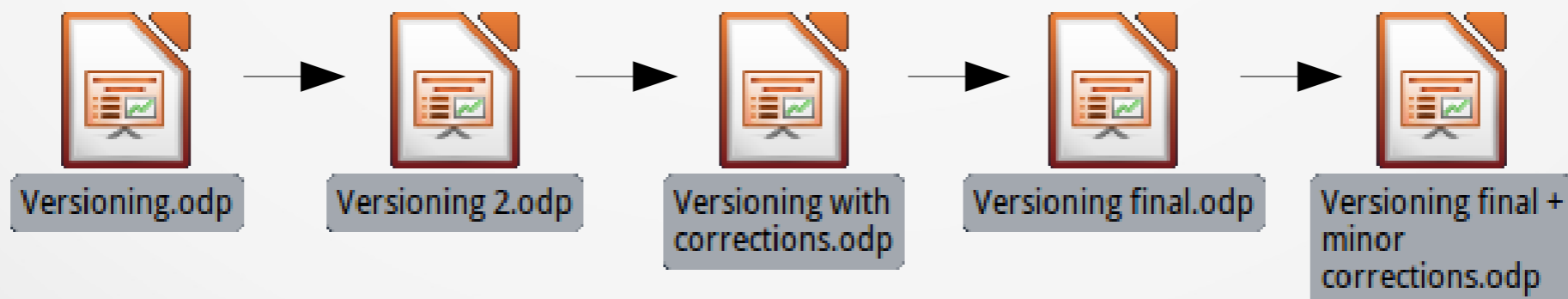
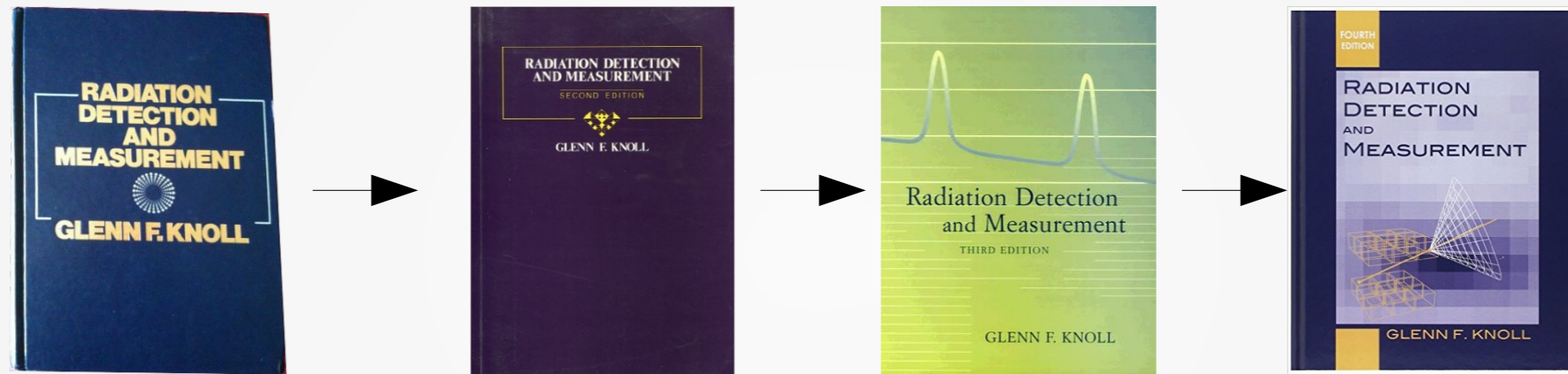
Scott Chalcon (git)

Game based on gameshell from

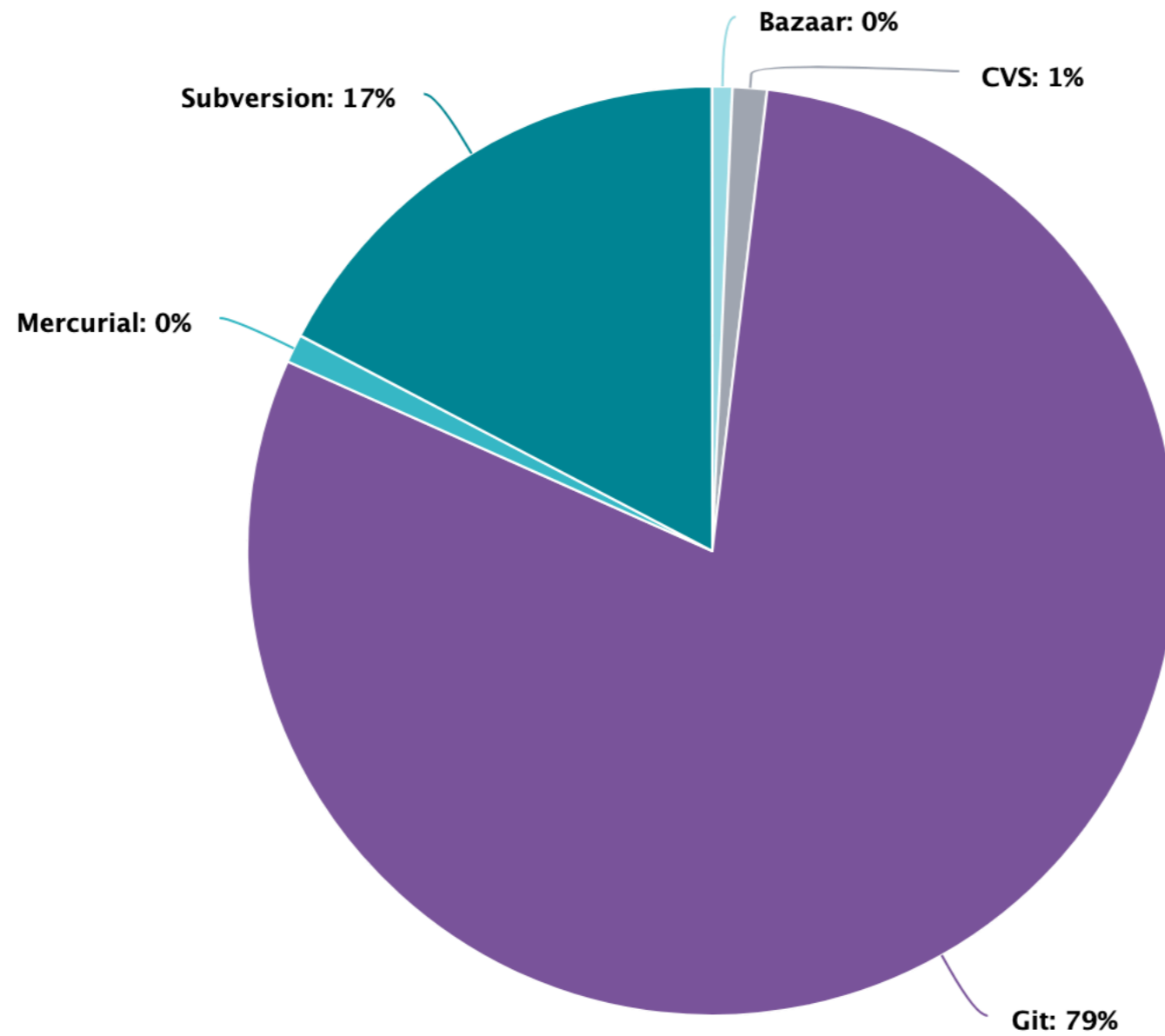
Pierre Hyvernats (SMB)

Rodolphe Lepigre (SMB)

What is code versioning



Why git?



TODAY

Mix of "Theory" and game...

1. History of modification

- Mission 1->6

2. Workflow

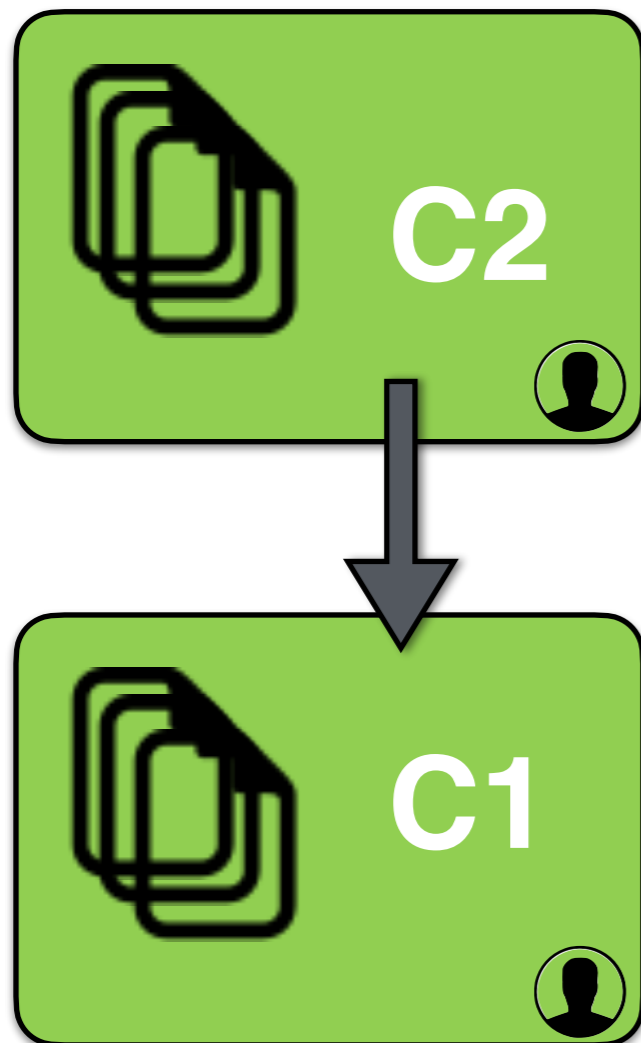
- Mission 7->11

3. Team Work

- 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 (SHA1)
 - MetaData (who created/when/info)
 - Pointer to previous(es) commit

Git Three area

Workspace



./WORKDIR

Index



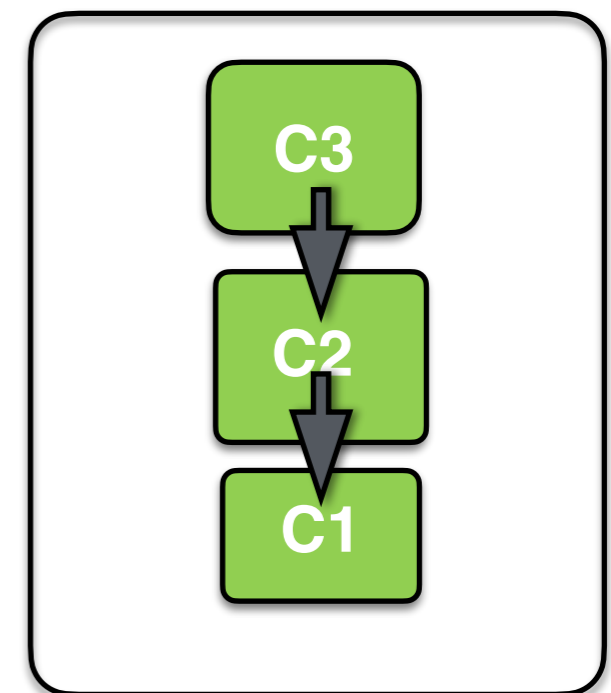
.git/index

Staging area

Repository

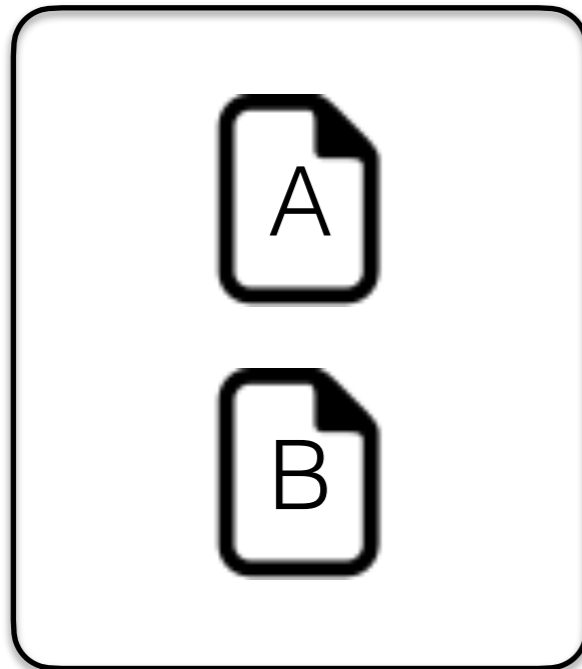


.git/

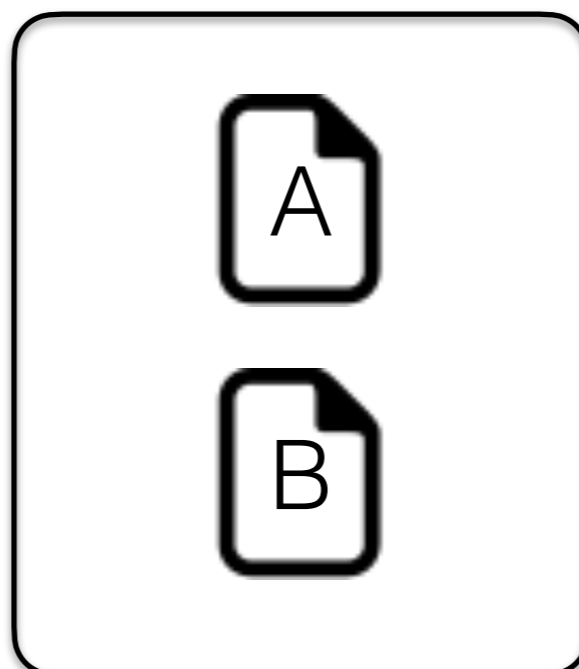


Git Three area

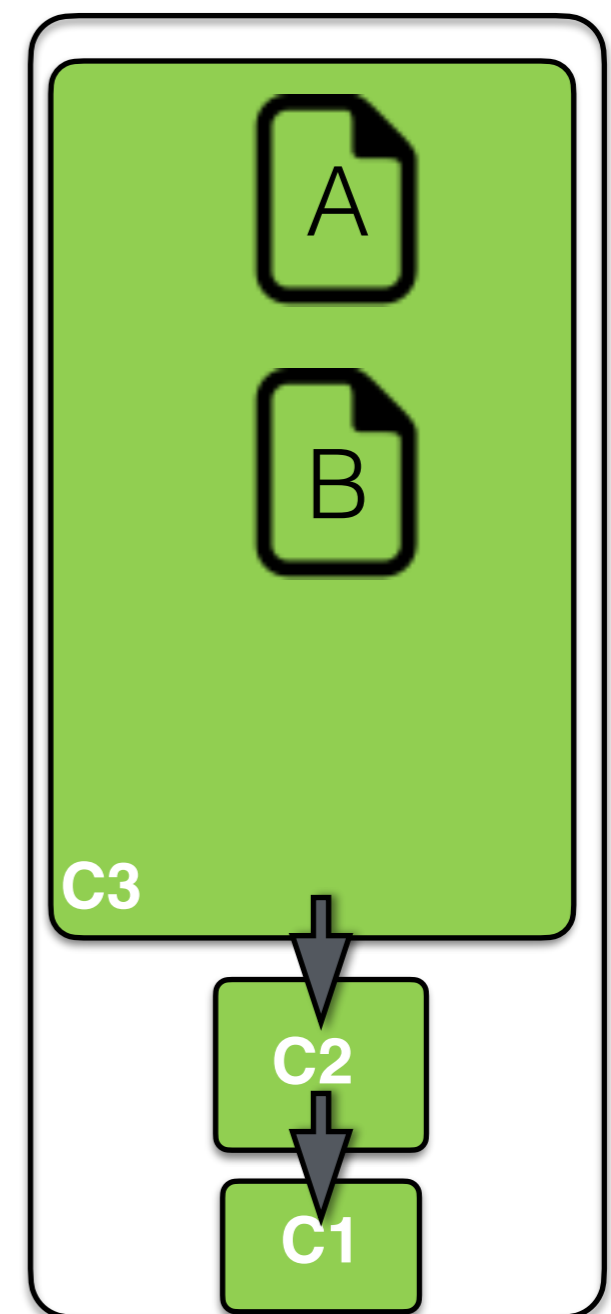
Workspace



Index

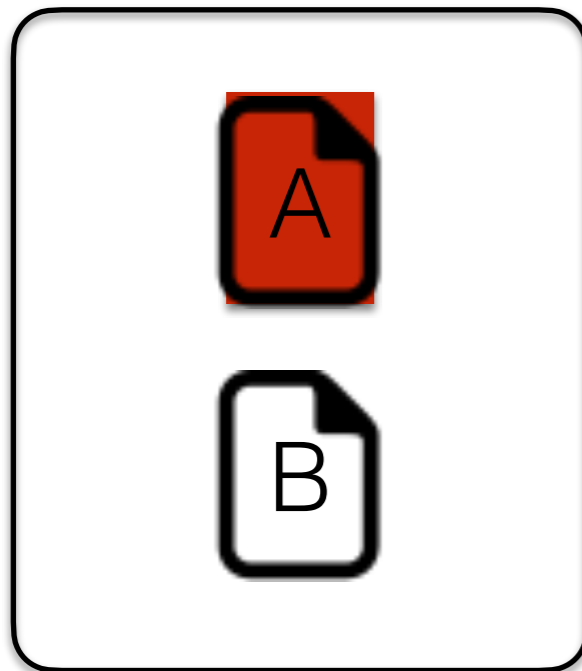


Repository

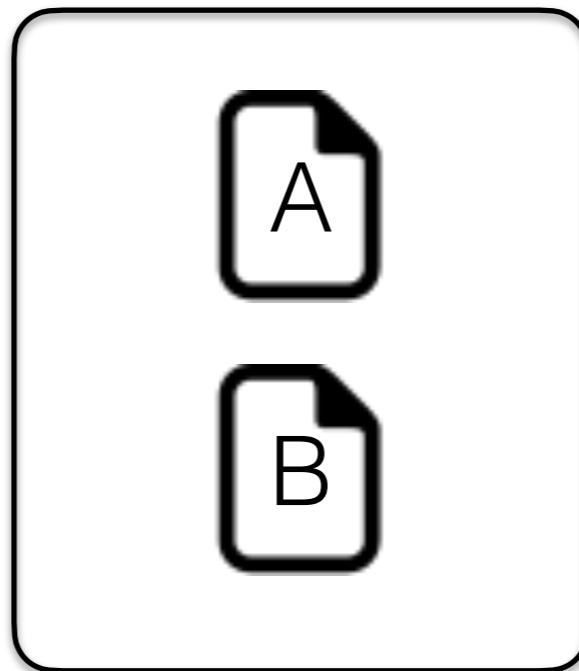


Git Three area

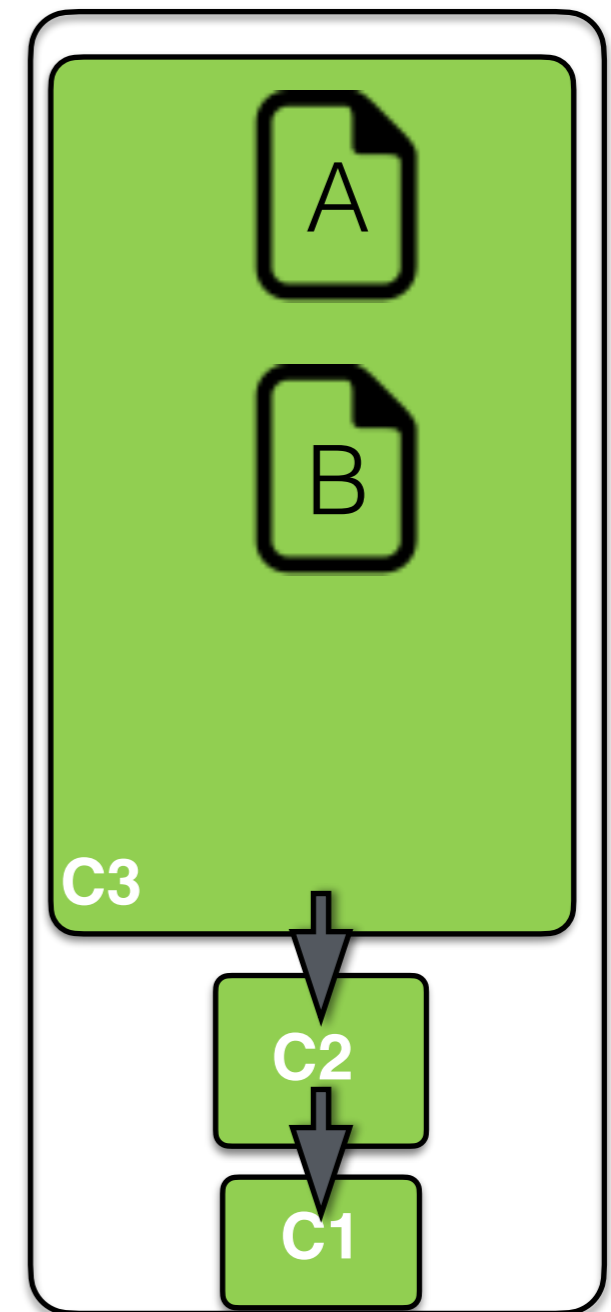
Workspace



Index



Repository

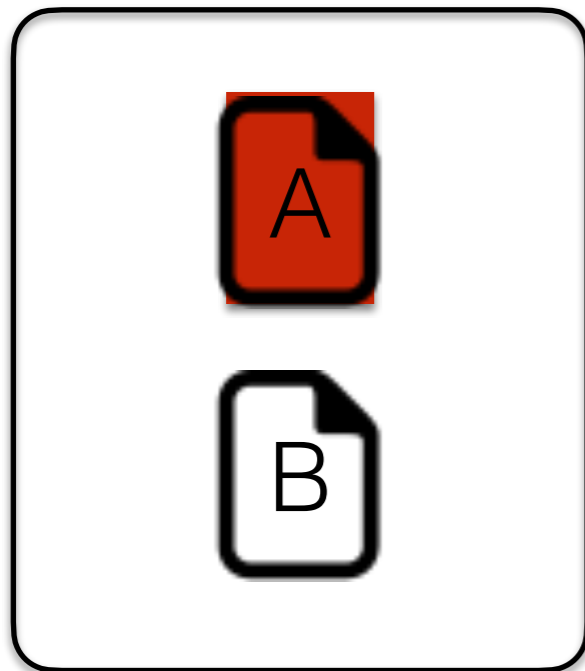


Action:

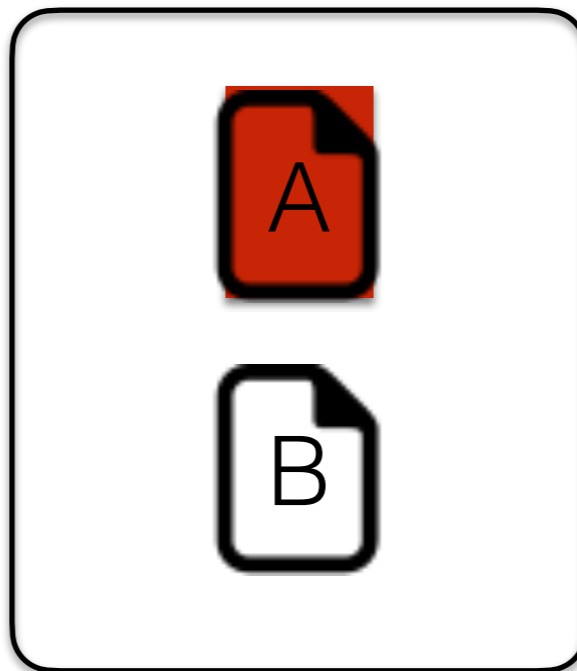
**Modifying file A
-> add a line**

Git Three area

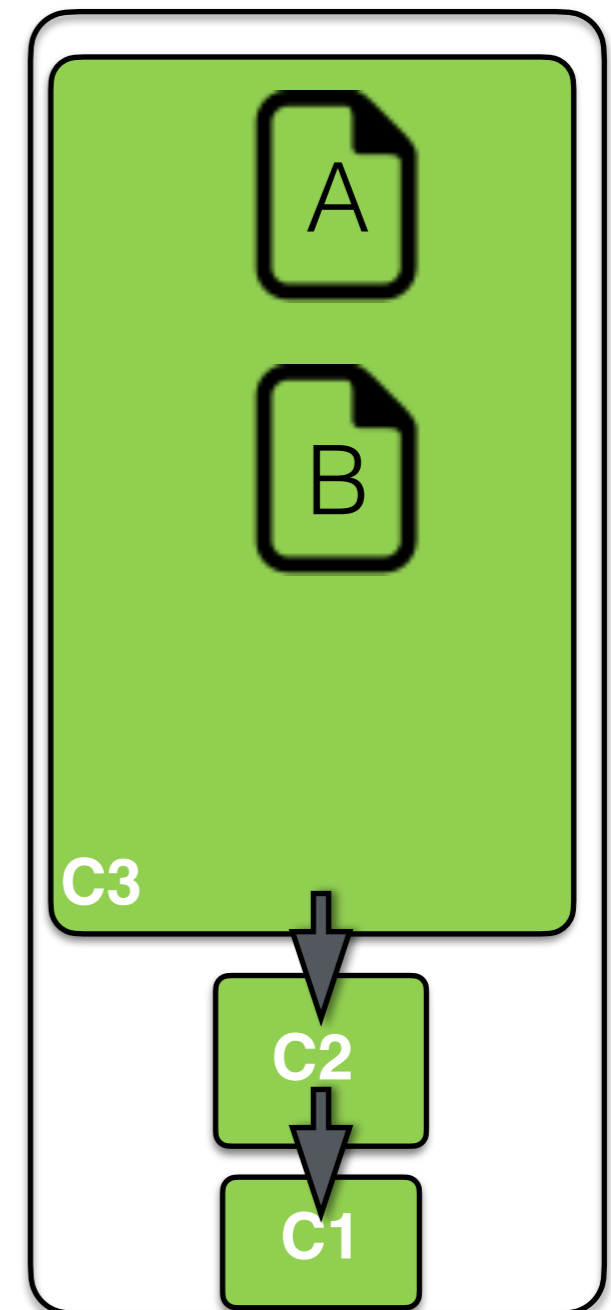
Workspace



Index



Repository



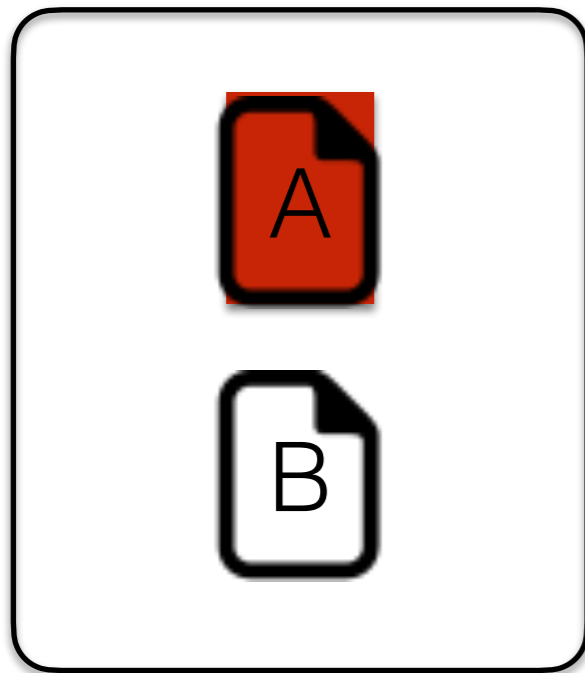
Action:

git add A

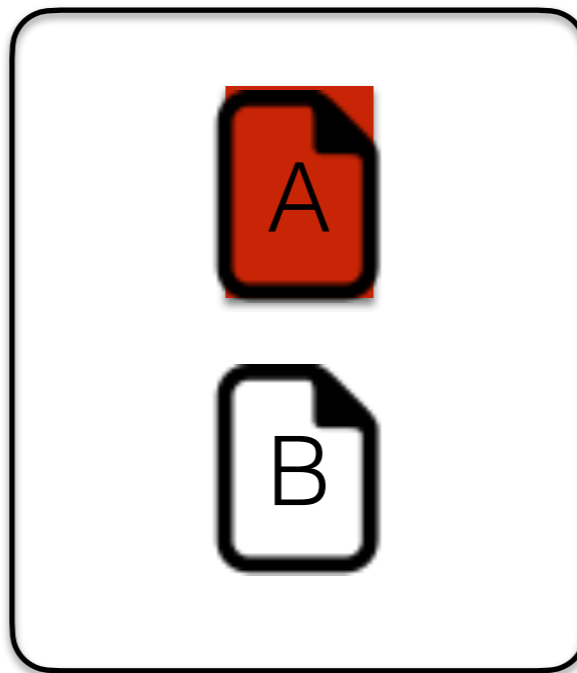
- > modify file moves to the index
- > inside the box
- > ready for a commit

Git Three area

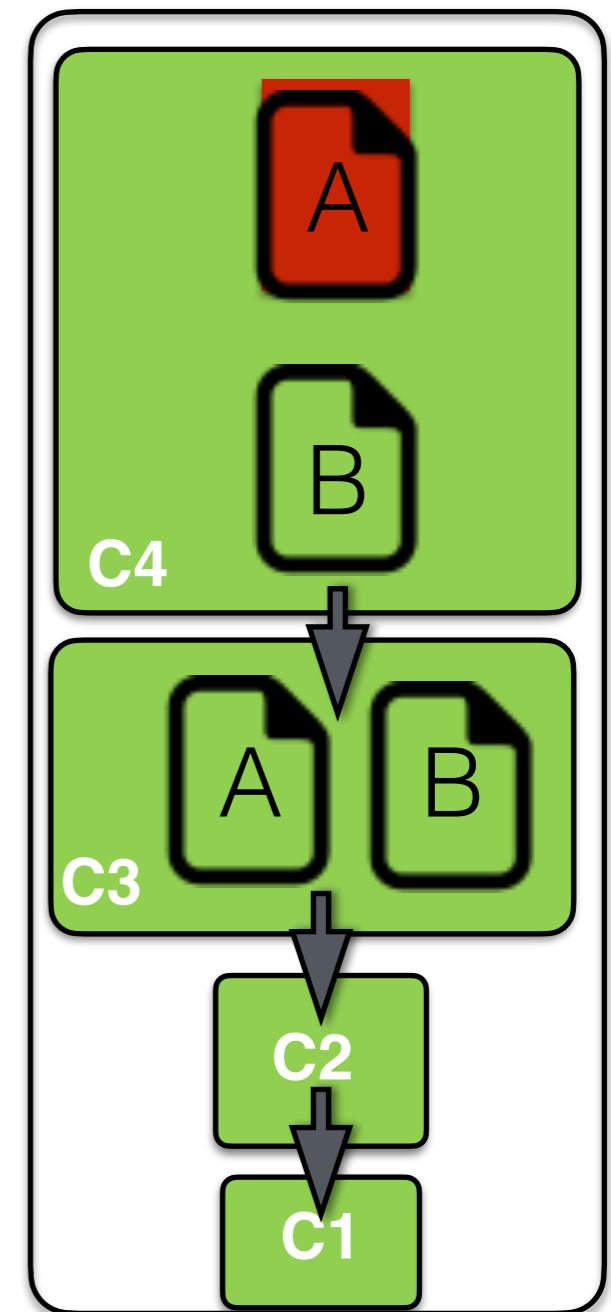
Workspace



Index



Repository



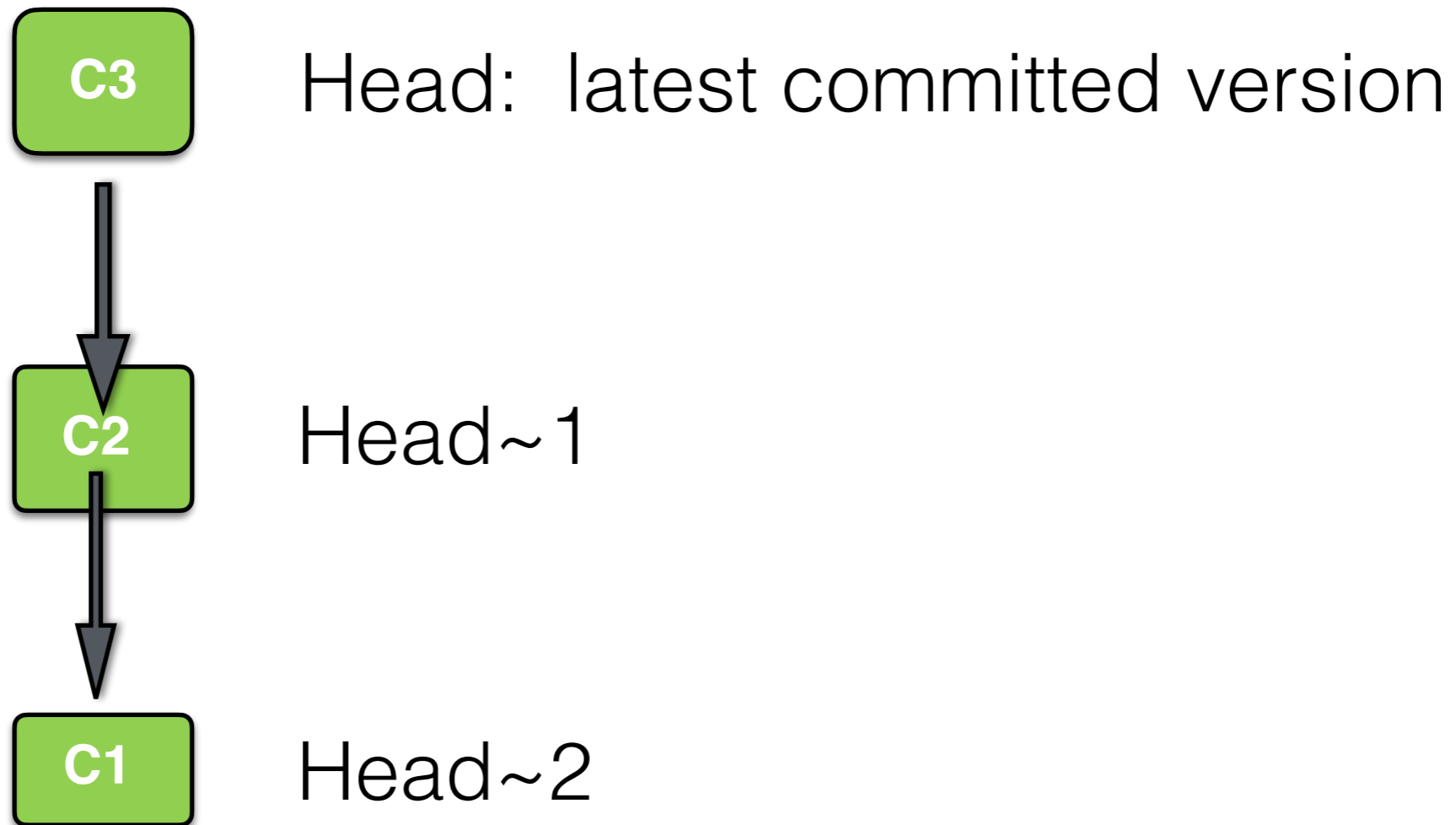
Action:

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



Let's play

Game installed on lemaitre4 and manneback
module load gameshell/git

Basic command of the game

gsh goal
gsh check

Issue with a level: restart it with
gsh reset

Exercise #1

Stop at mission #8

You have ~45 min

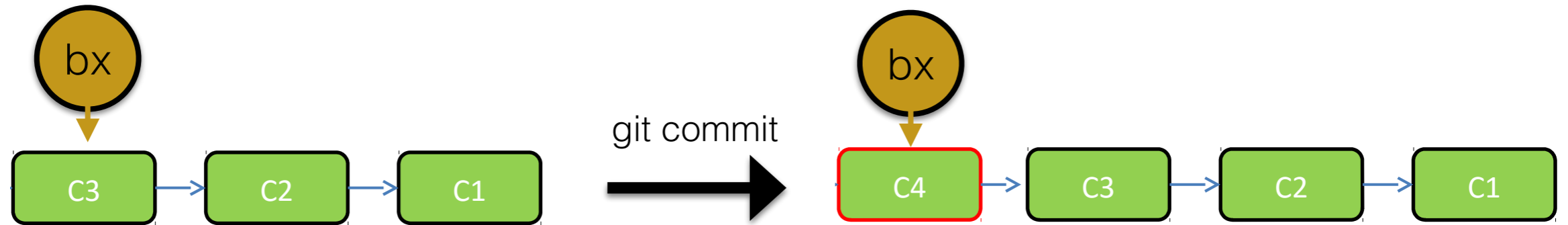
Workflow

branch in git

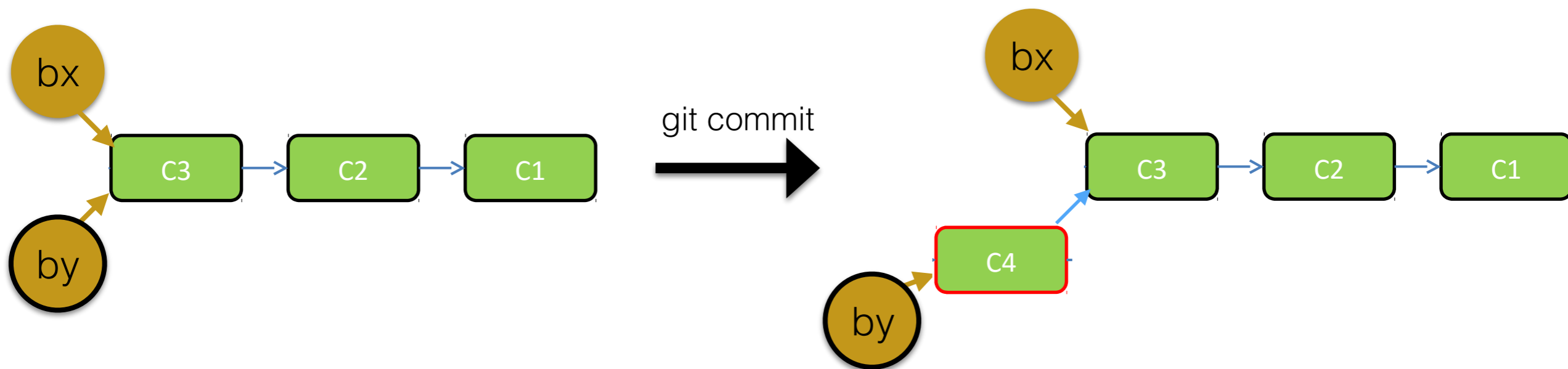
- Branch is **pointer** to a commit (represent an history) which represent a **line of development**
- A branch can point at other commit, it **can move!**
- A branch is a way to organise your work and working histories

branches

- default branch: master (or main)
- When doing a commit, the branch moves to the new commit

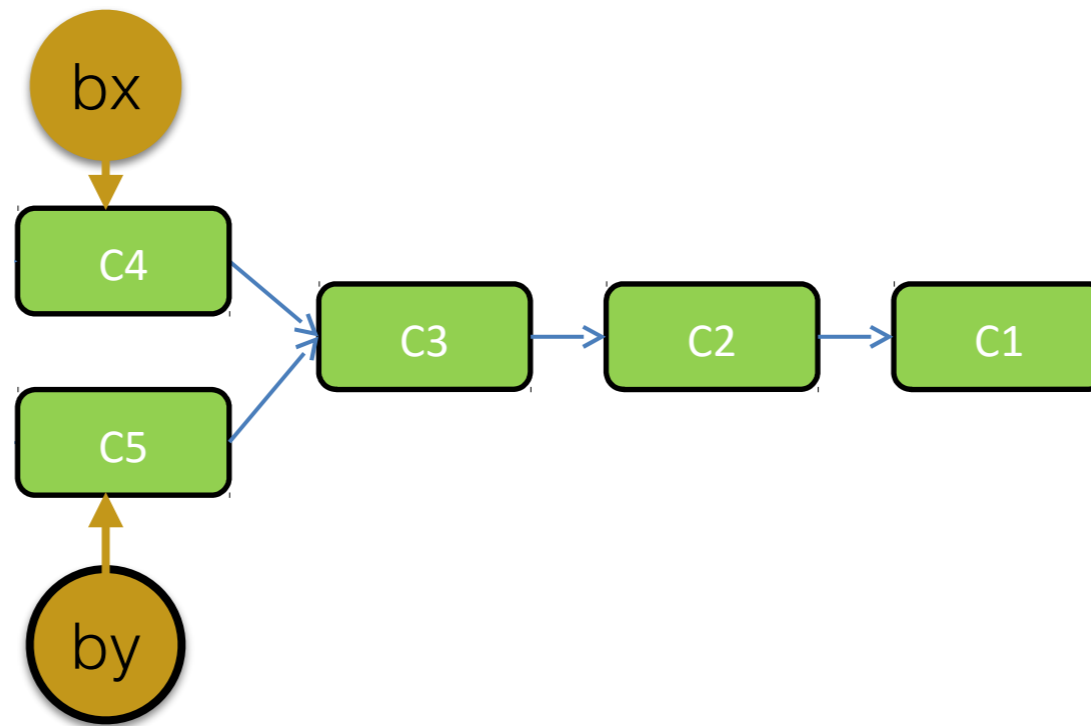


- creating a new branch: add a pointer (git switch -c by)
- only selected branch affected by commit!



merging

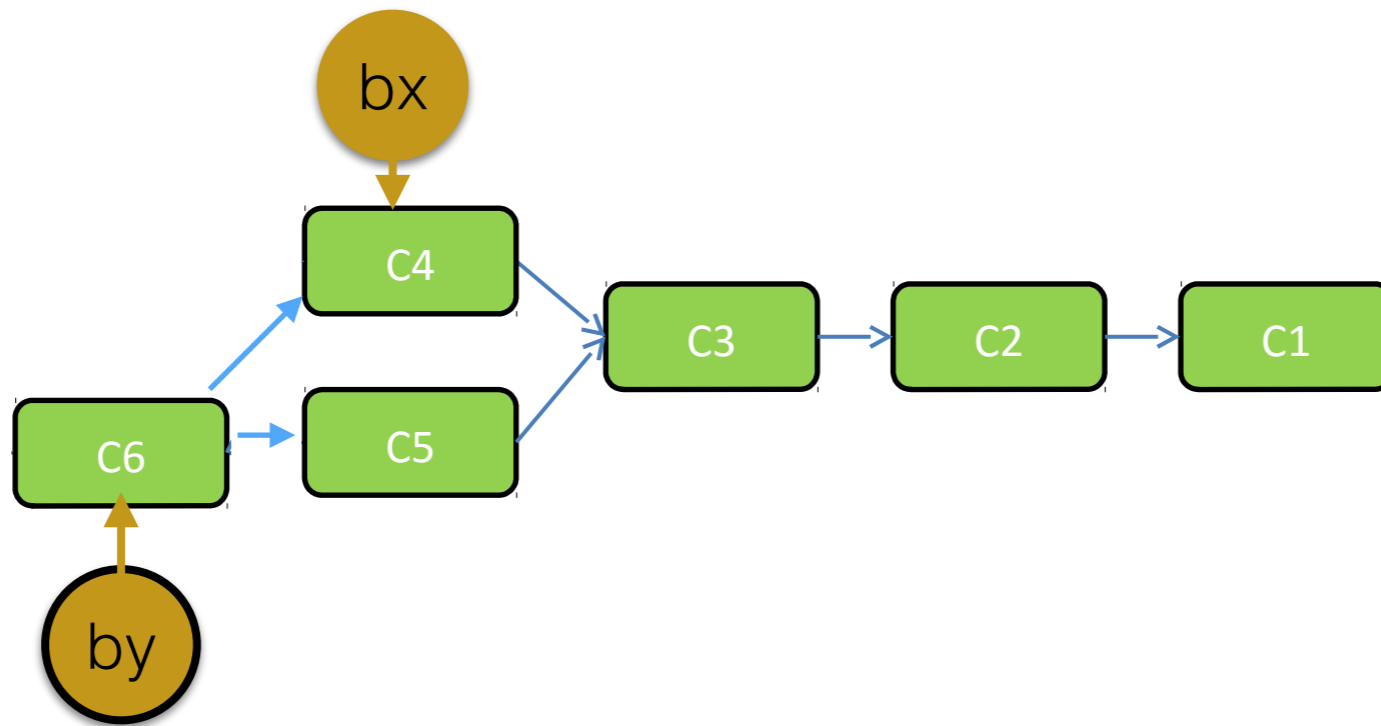
- The interest of branch is that you can **merge** them
- Include in one (branch) file the modification done somewhere else



git merge bx

merging

- merging two different modifications



git merge bx



working-dir

Repository

Do it yourself

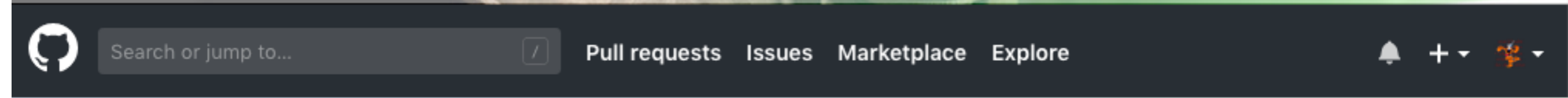
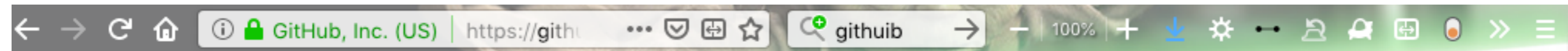
- Mission #8-14

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.

[Read the guide](#)

[Start a project](#)

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

Repositories

[New repository](#)

Find a repository...

[oliviermattelaer/singularity-recipe](#)

[oliviermattelaer/MGISR-1](#)

Browse activity

[Discover repositories](#)



dcolignon starred [oliviermattelaer/Singularity-Tutorial](#) 7 days ago

[oliviermattelaer/Singularity-Tutorial](#)

[★ Star](#)

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

My Machine

The Server

origin/master

master

master

C1

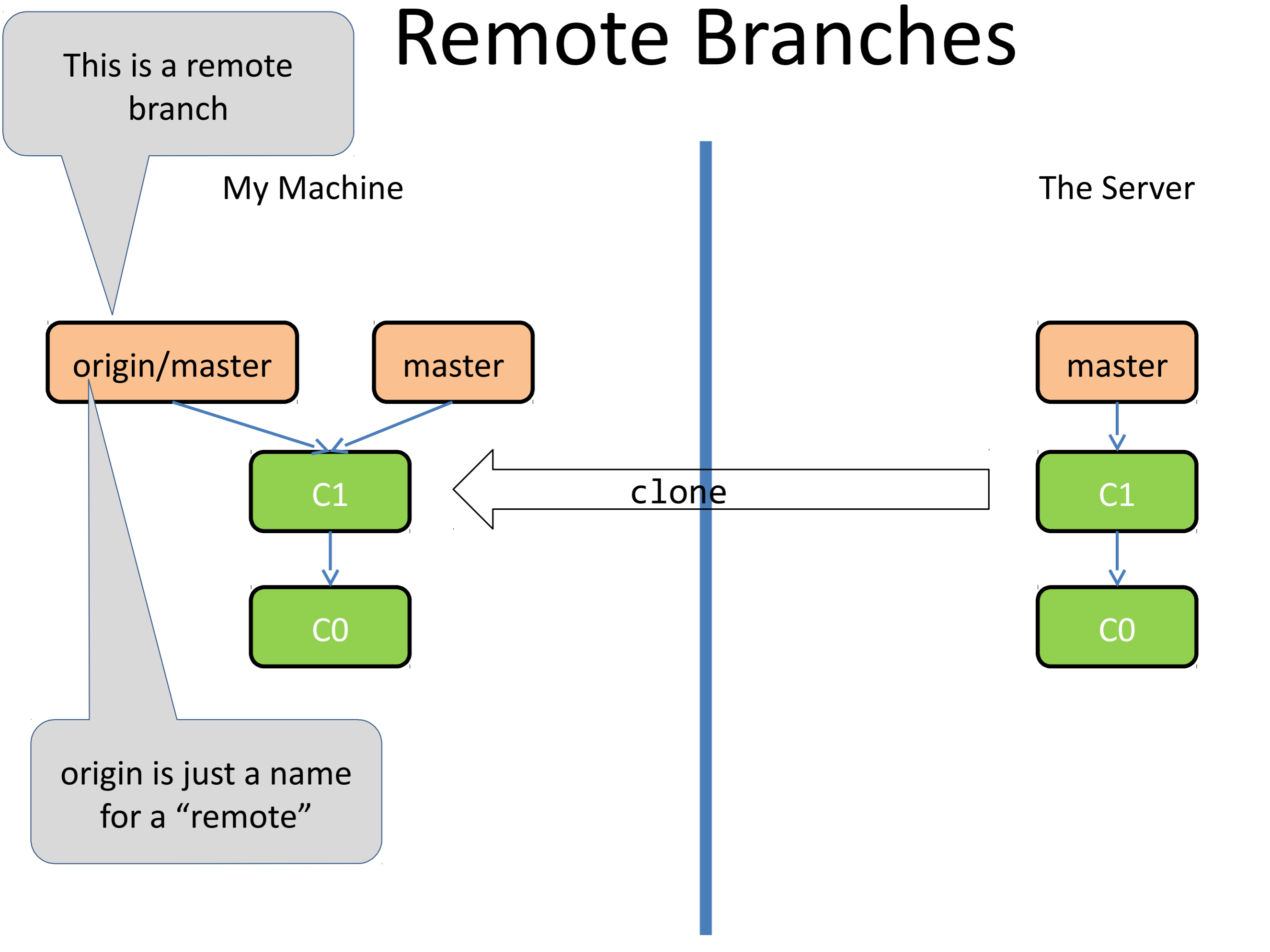
← clone

C1

C0

C0

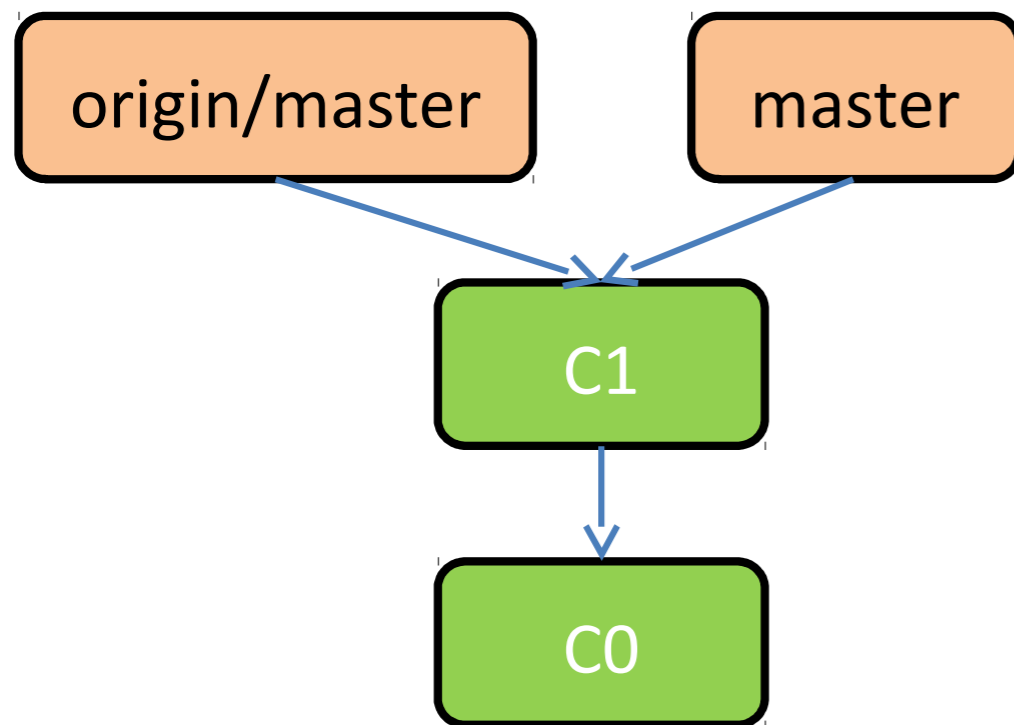
origin is just a name for a "remote"



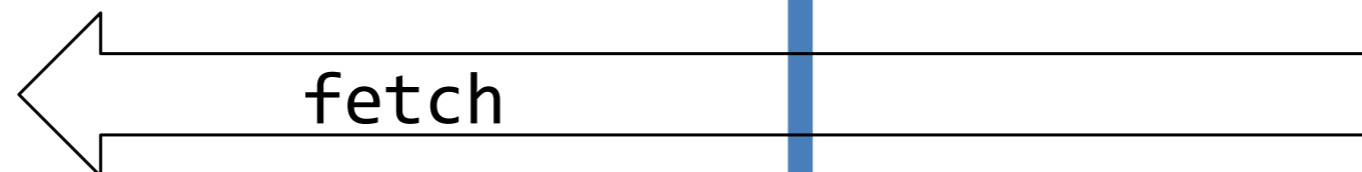
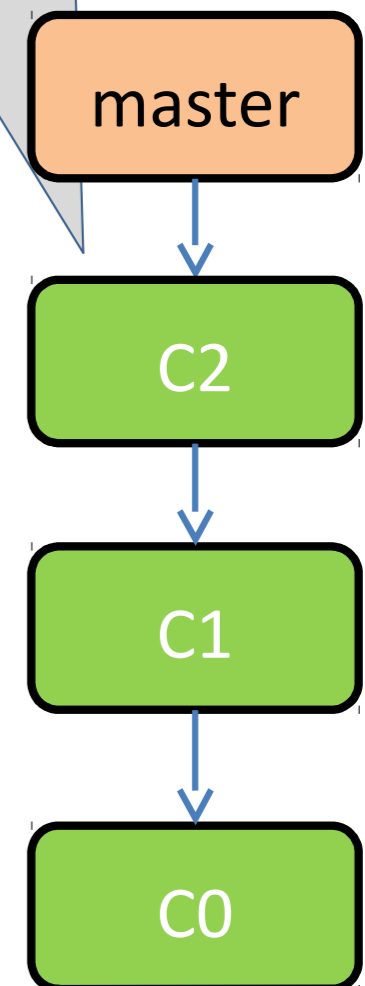
Remote Branches - fetch

My Machine

The Server

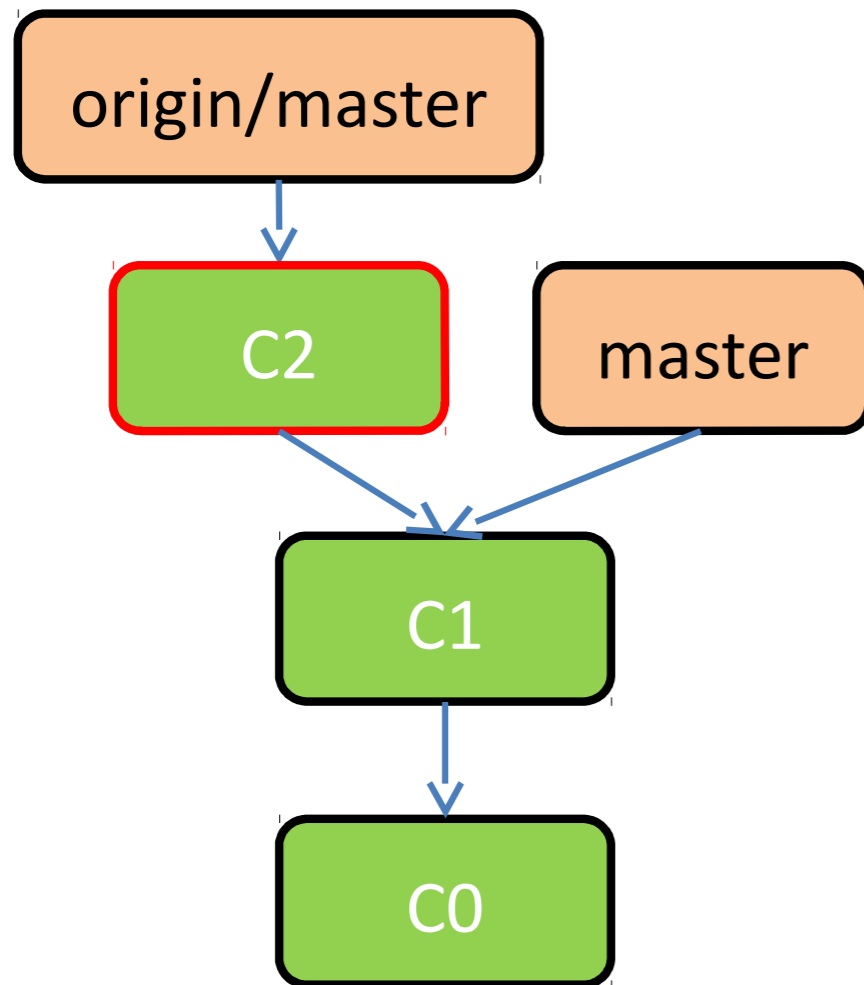


Some changes on the server

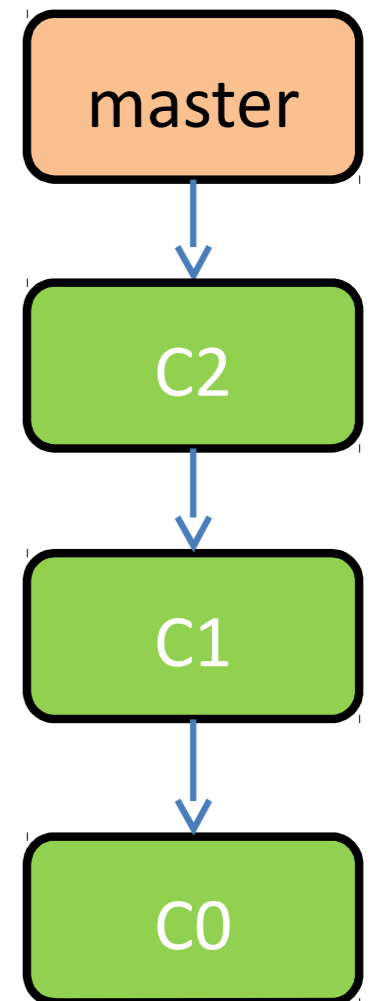


Remote Branches - fetch

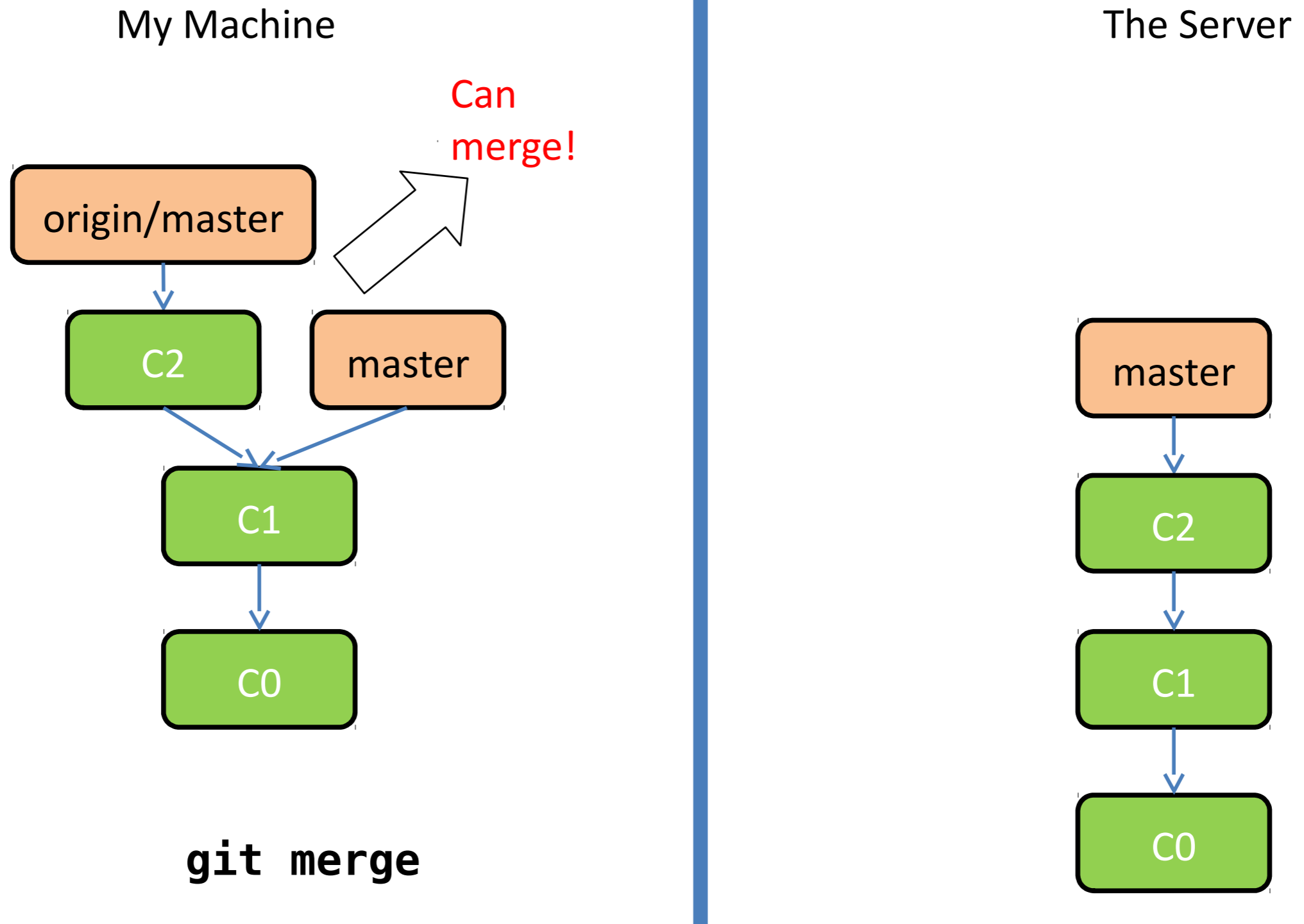
My Machine



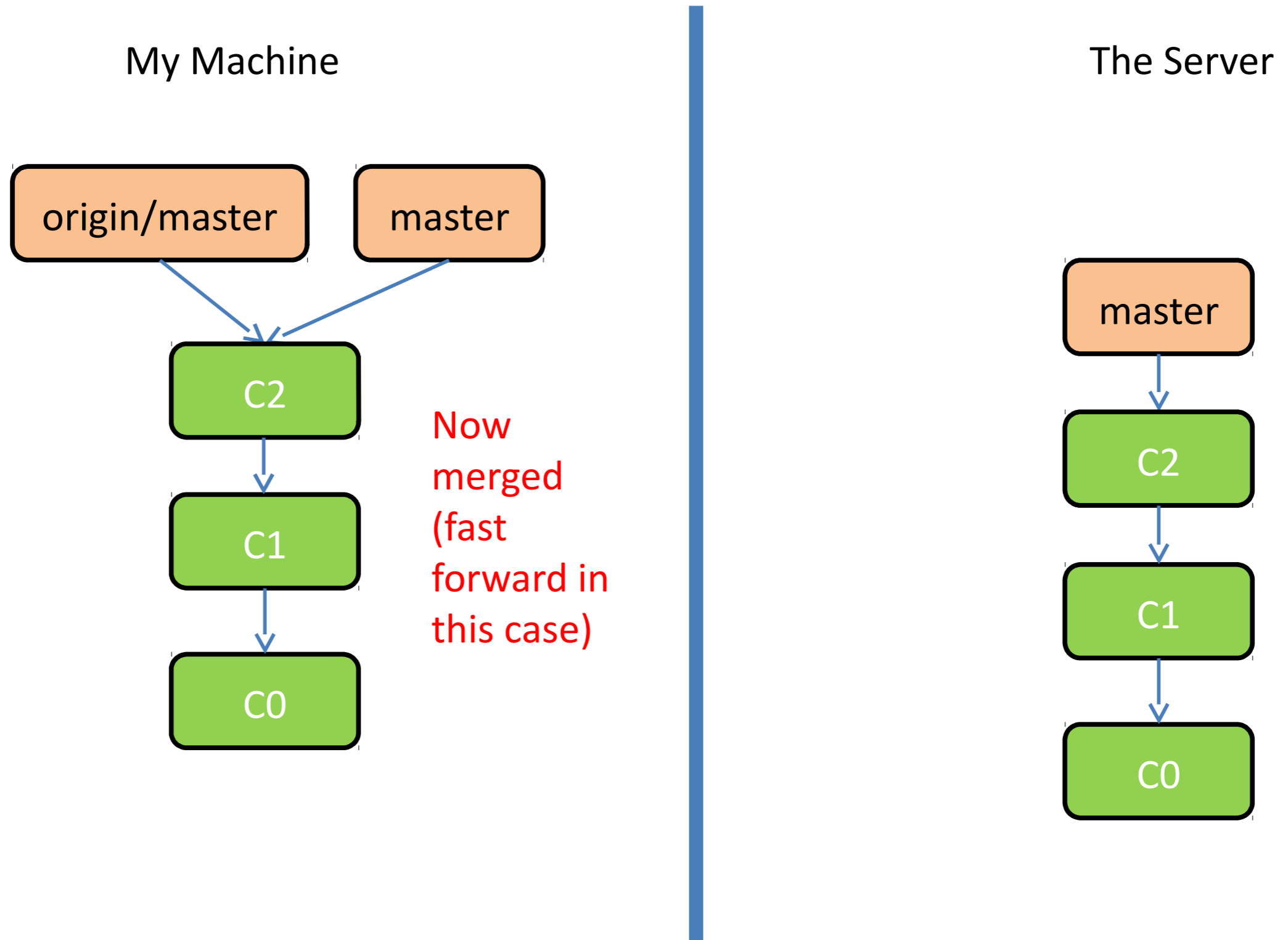
The Server



Remote Branches - fetch



Remote Branches - fetch



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”

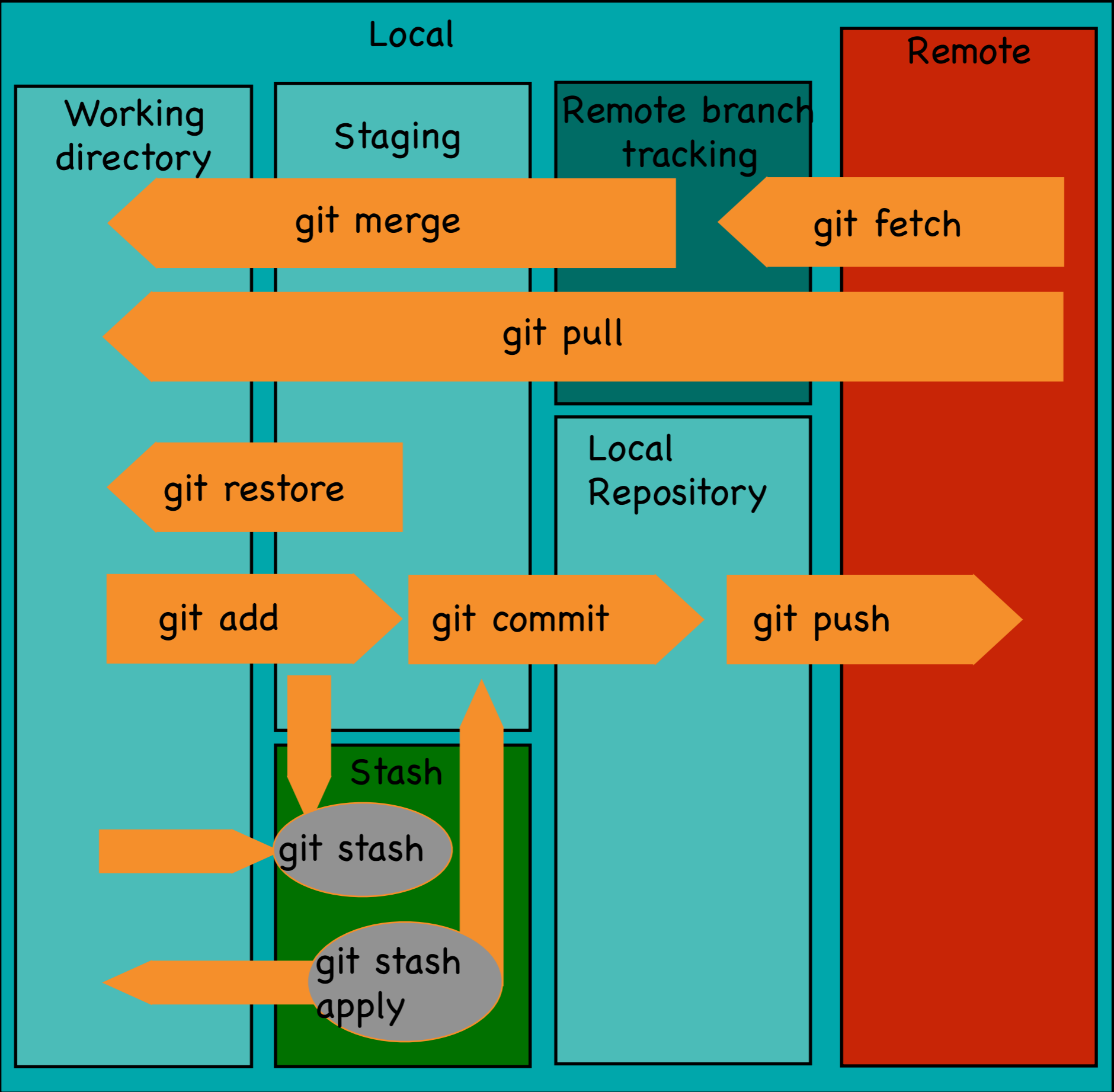


working-dir

Repository

Do it yourself

- Mission #13-17



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

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>