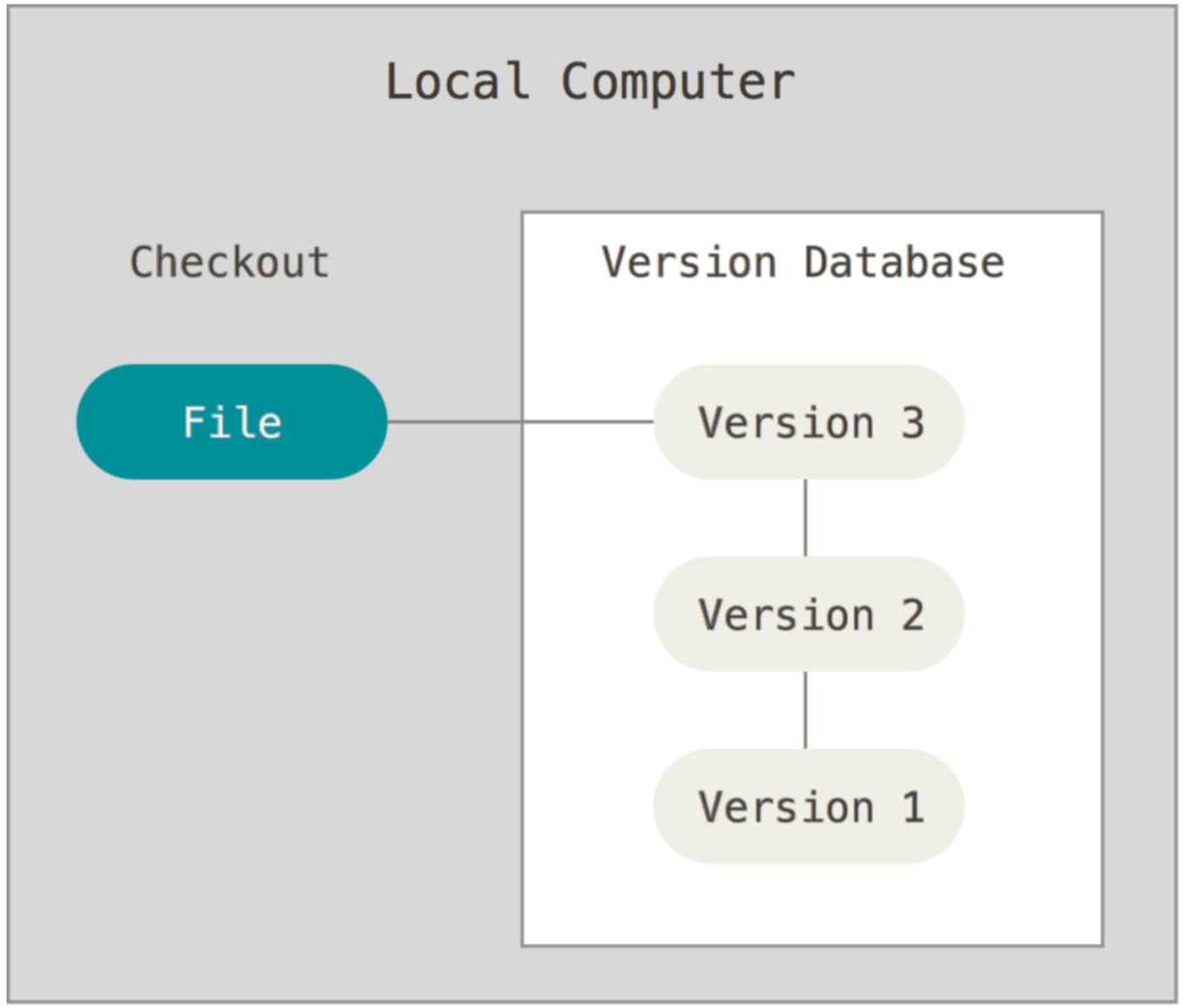
heig-vd

Git pour PRG1

Olivier Cuisenaire

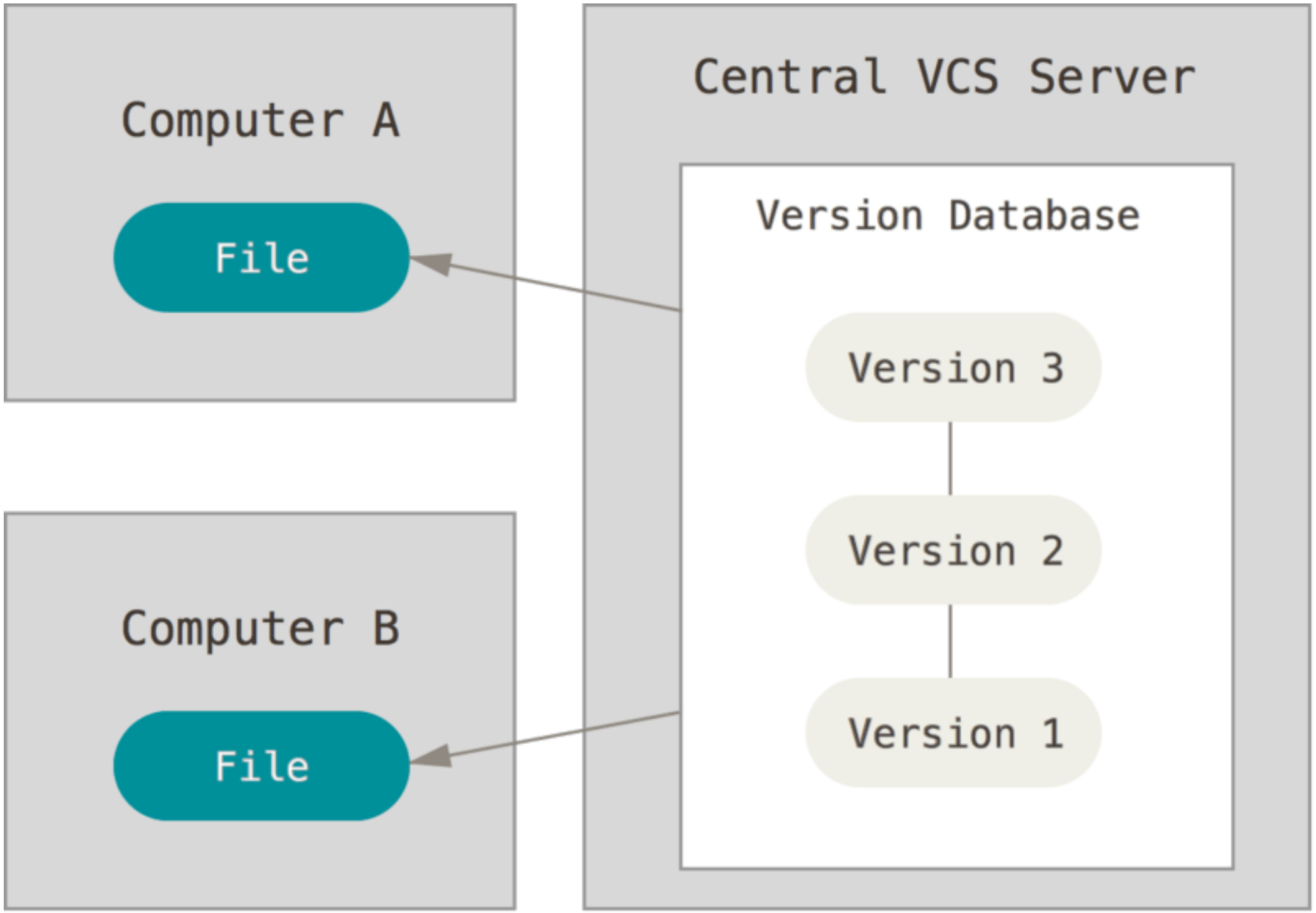
heig-vd

VCS = Version Control Sofware



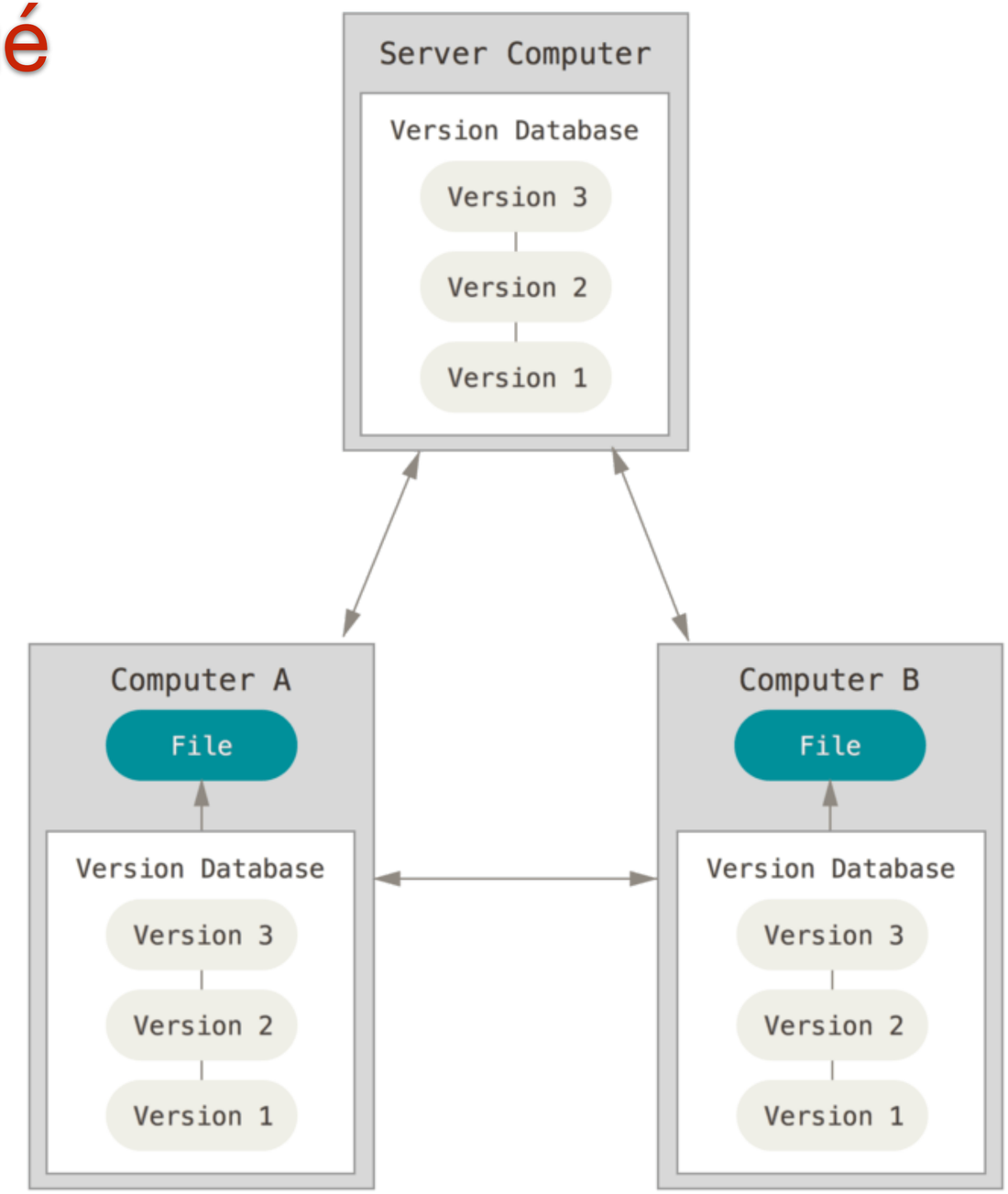
heig-vd

VCS centralisé



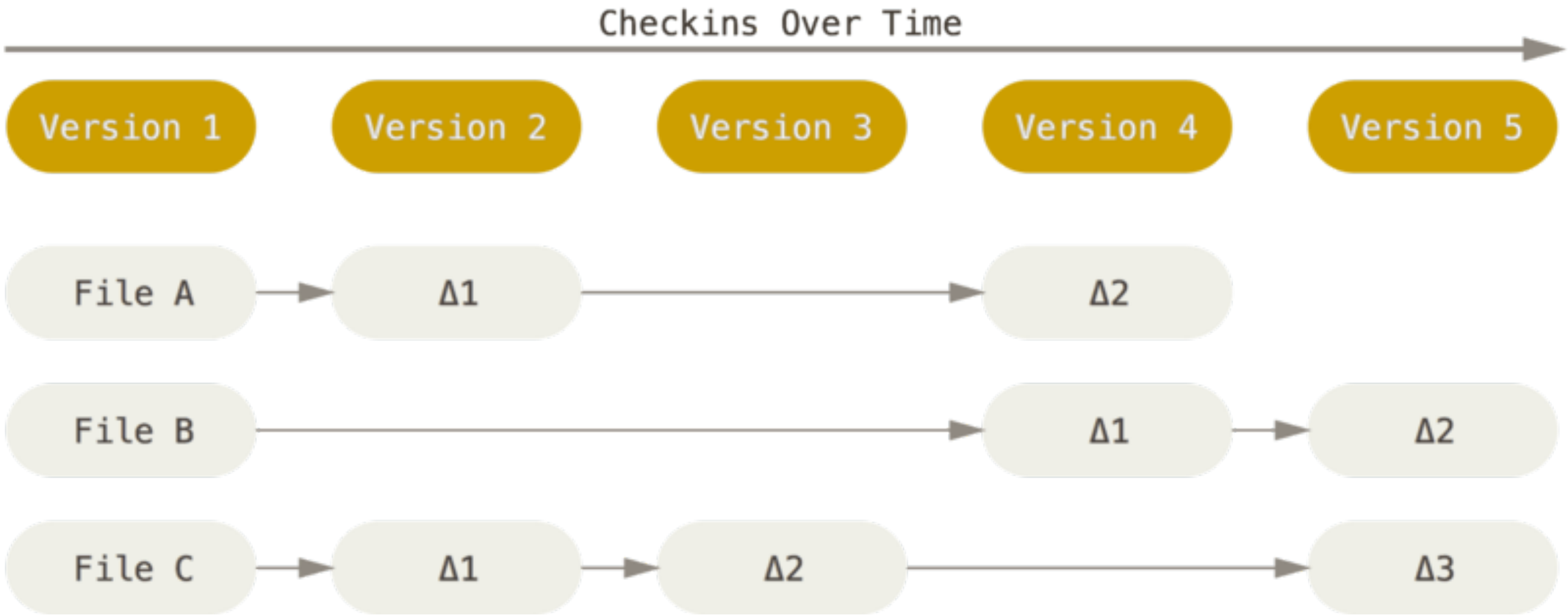
heig-vd

Git = VCS distribué



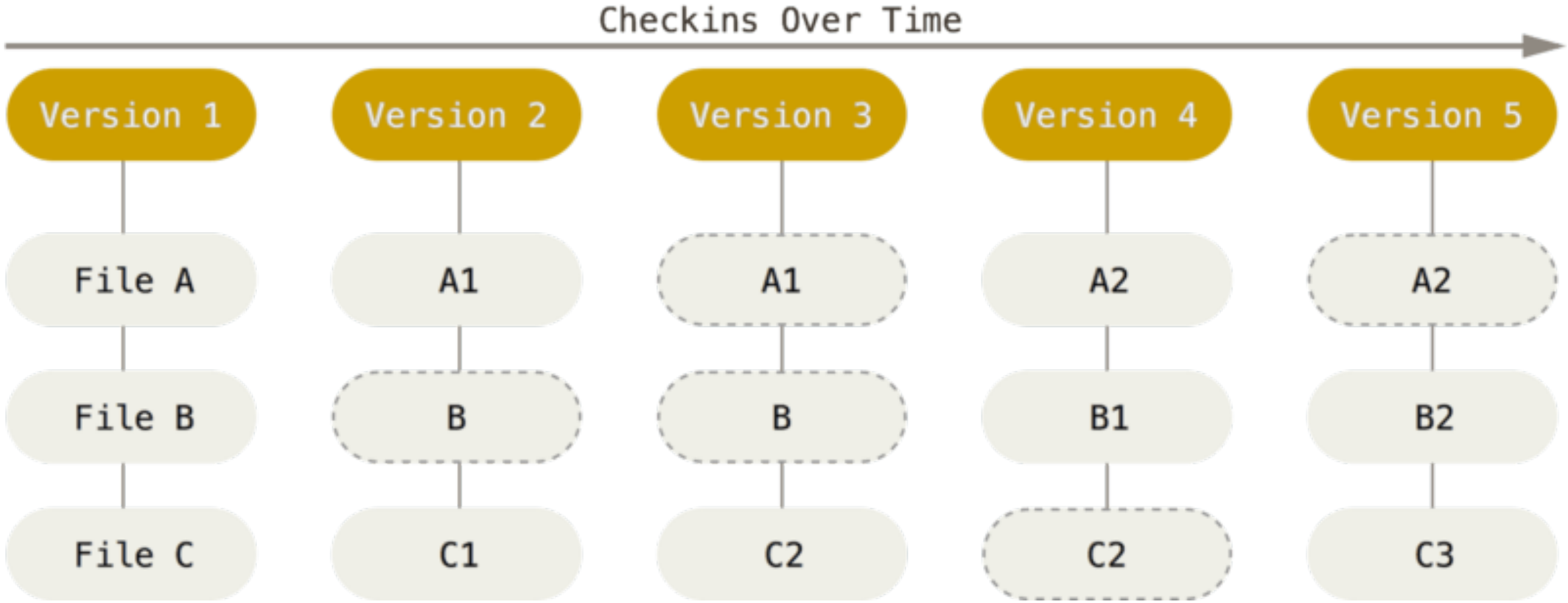
heig-vd

Les anciens VCS stockent des différences



heig-vd

Git stocke les fichiers qui ont changé



heig-vd

Démarrer un repo git

- Localement

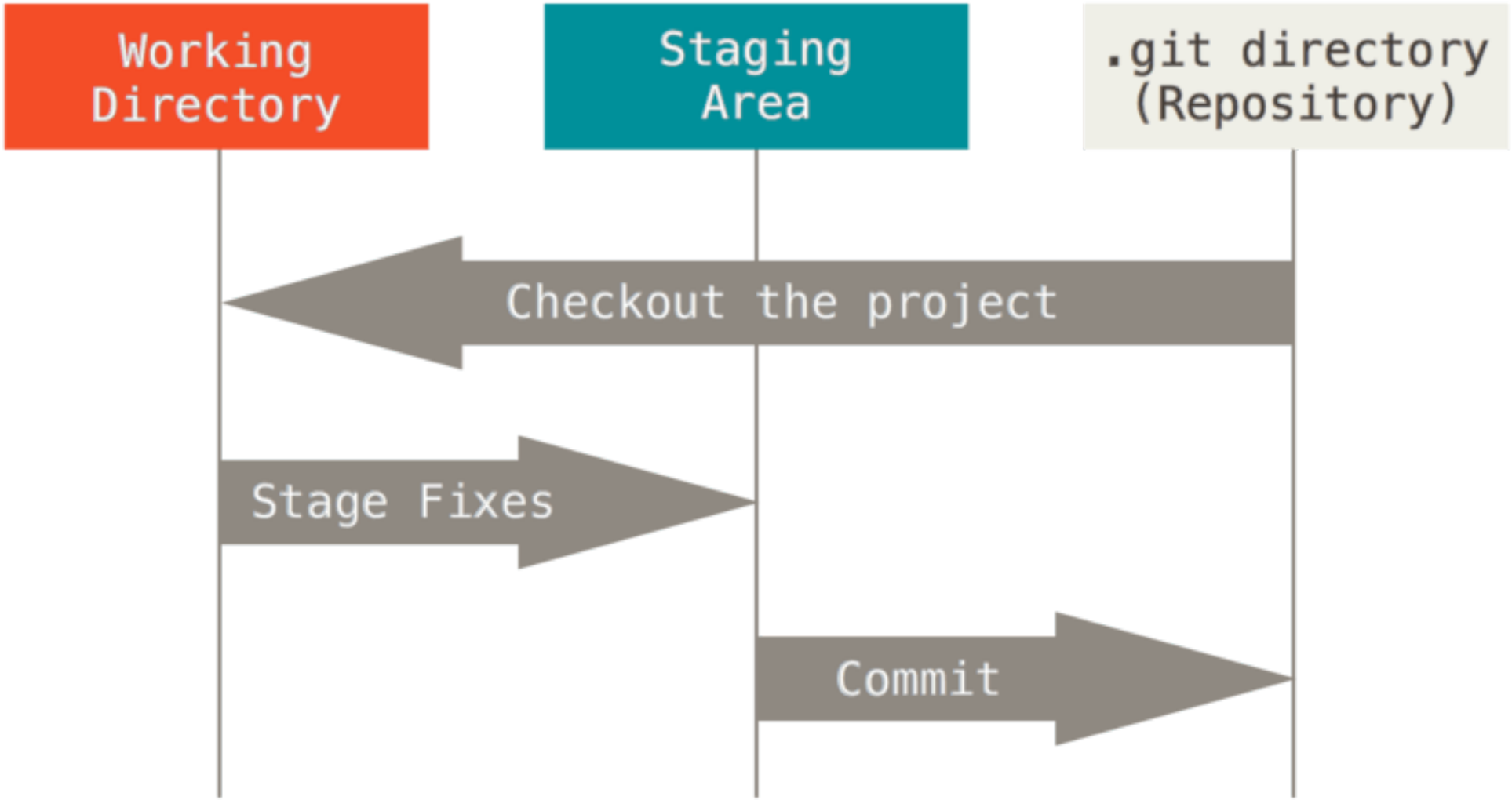
```
git init
```

- A partir d'un repo à distance

```
git clone https://github.com/ocuisenaire/PRG1-Exercices.git
```

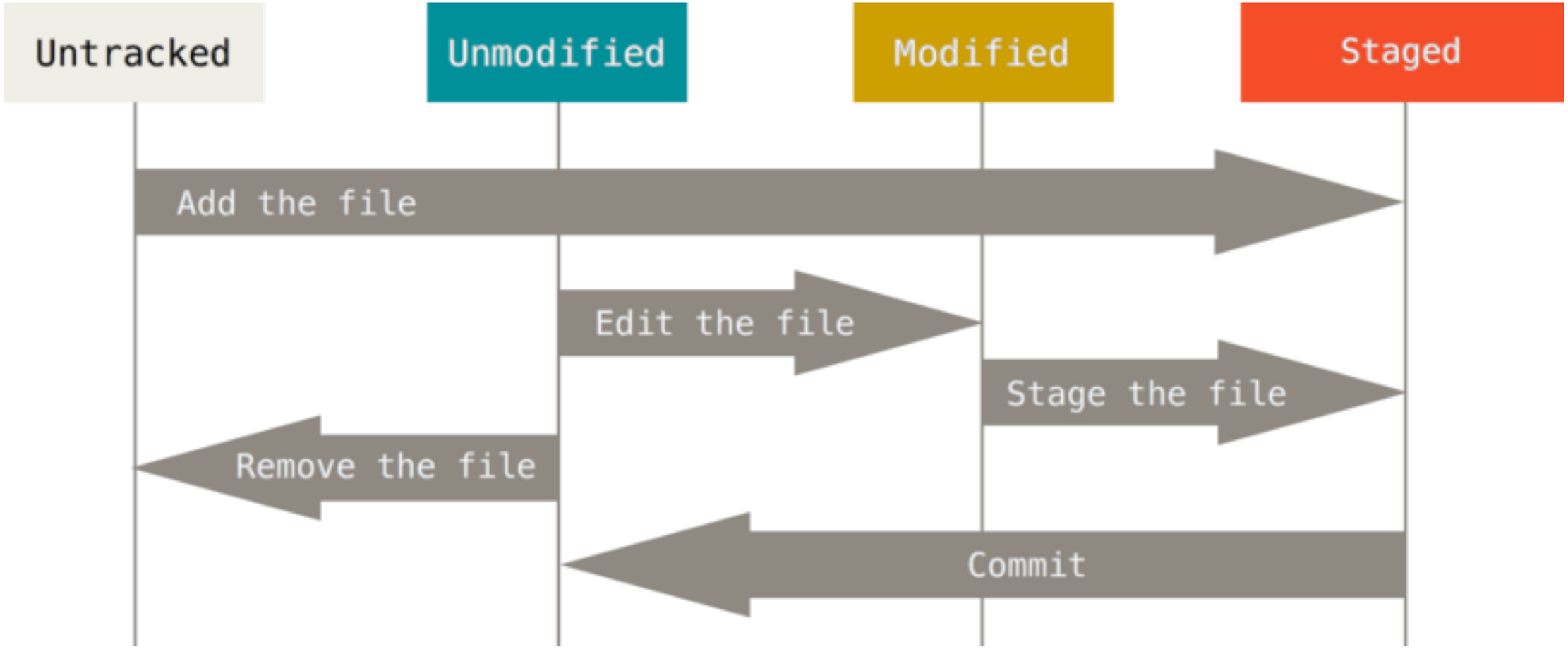
heig-vd

Sections d'un projet Git



heig-vd

Etats d'un fichiers



heig-vd

Les commandes git

- `git add [fichiers]`
  - Place les fichiers et/ou répertoires dans la staging area
  - Valide depuis untracked ou depuis modified
  - Inopérant depuis unmodified
- `git commit -m "message à choix"`
  - Déplace les fichiers de la staging area au repository
  - Les fichiers commités sont maintenant unmodified

heig-vd

Les commandes git (2)

- `git status`
  - Indique les fichiers untracked, modified ou staged dans votre répertoire
- `git reset [fichier]`
  - Annule un `git add` en retirant le fichier modifié de la staging area
- `git diff [fichier optionnel]`
  - Affiche la différence entre le(s) fichiers courants et la version commitée

heig-vd

Les commandes git (3)

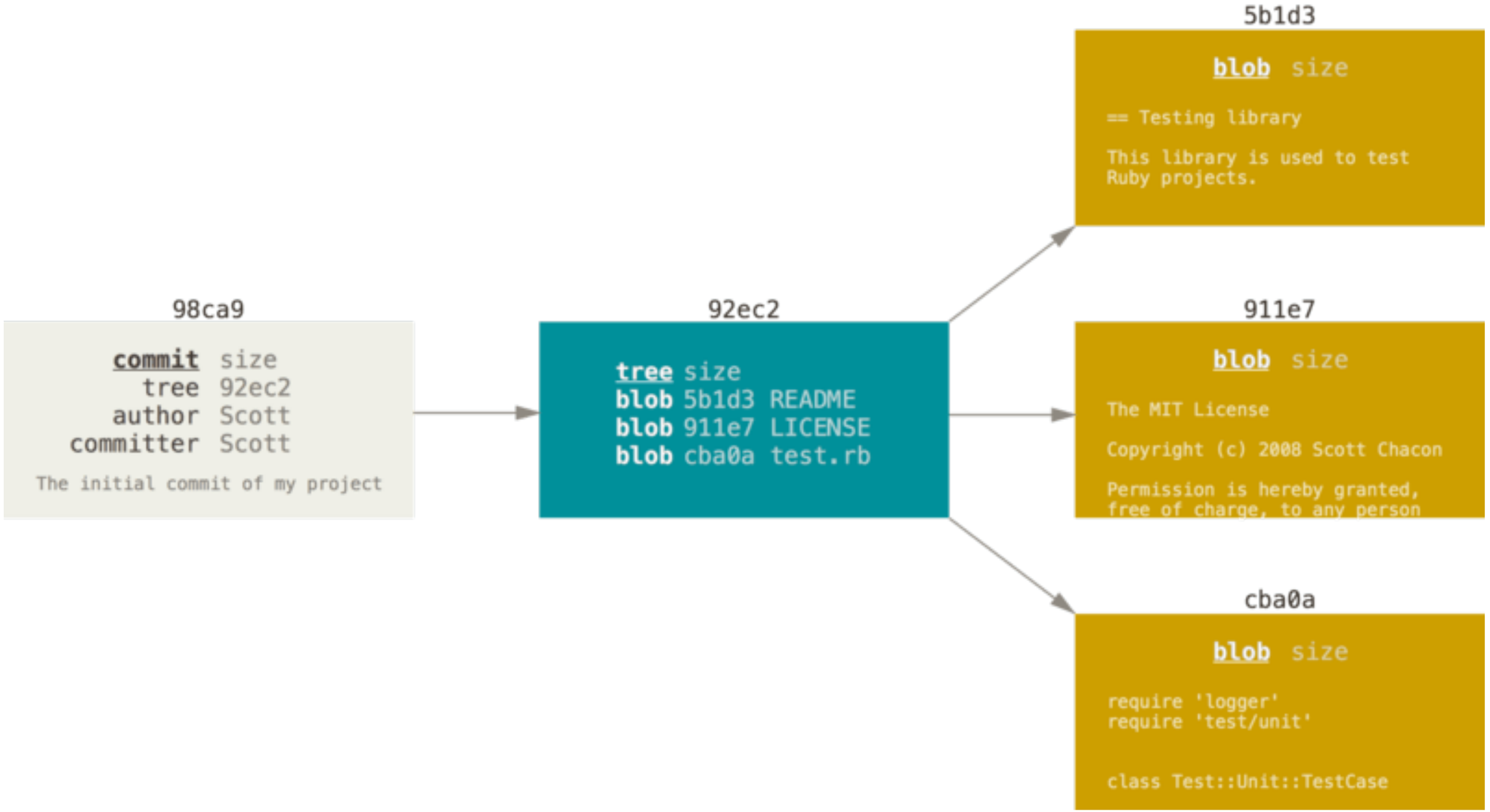
- `git diff --staged`
  - Affiche la différence entre les fichiers courants et la staging area
- `git checkout [fichier]`
  - Ecrase le fichier courant avec la version stockée dans le repo
- `git rm [fichier]`
  - Efface le fichier et le « stage » pour le prochain commit
- `git mv [fichier]`
  - Renomme le fichier et l’annonce pour le prochain commi

heig-vd

Les branches

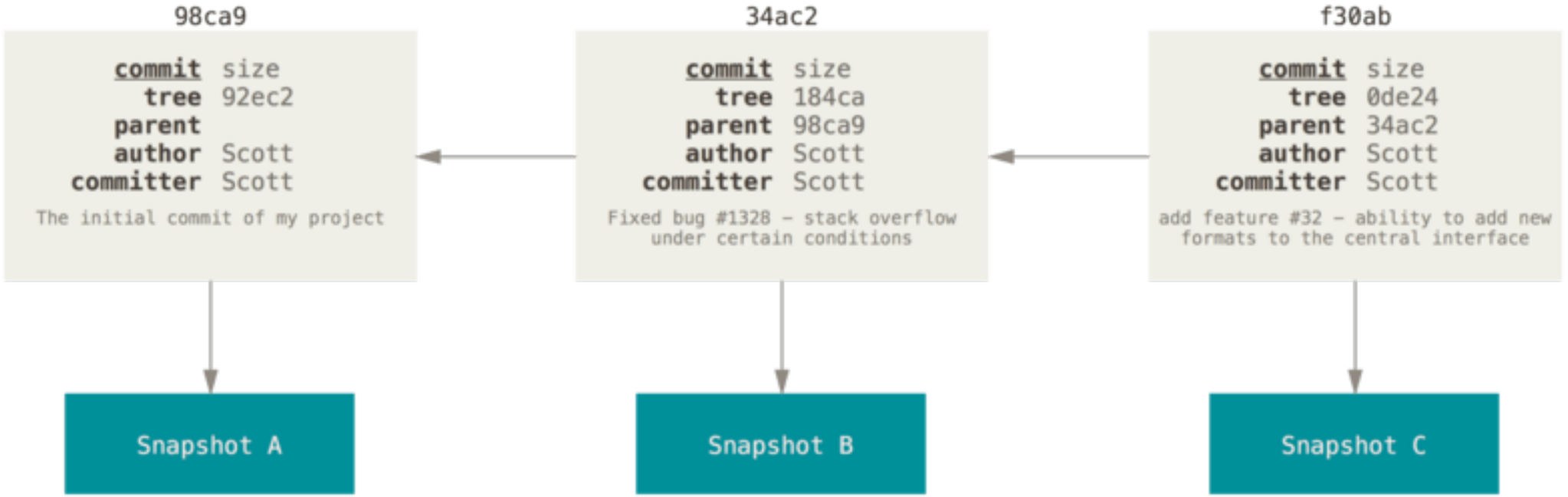
heig-vd

Un commit dans le repository



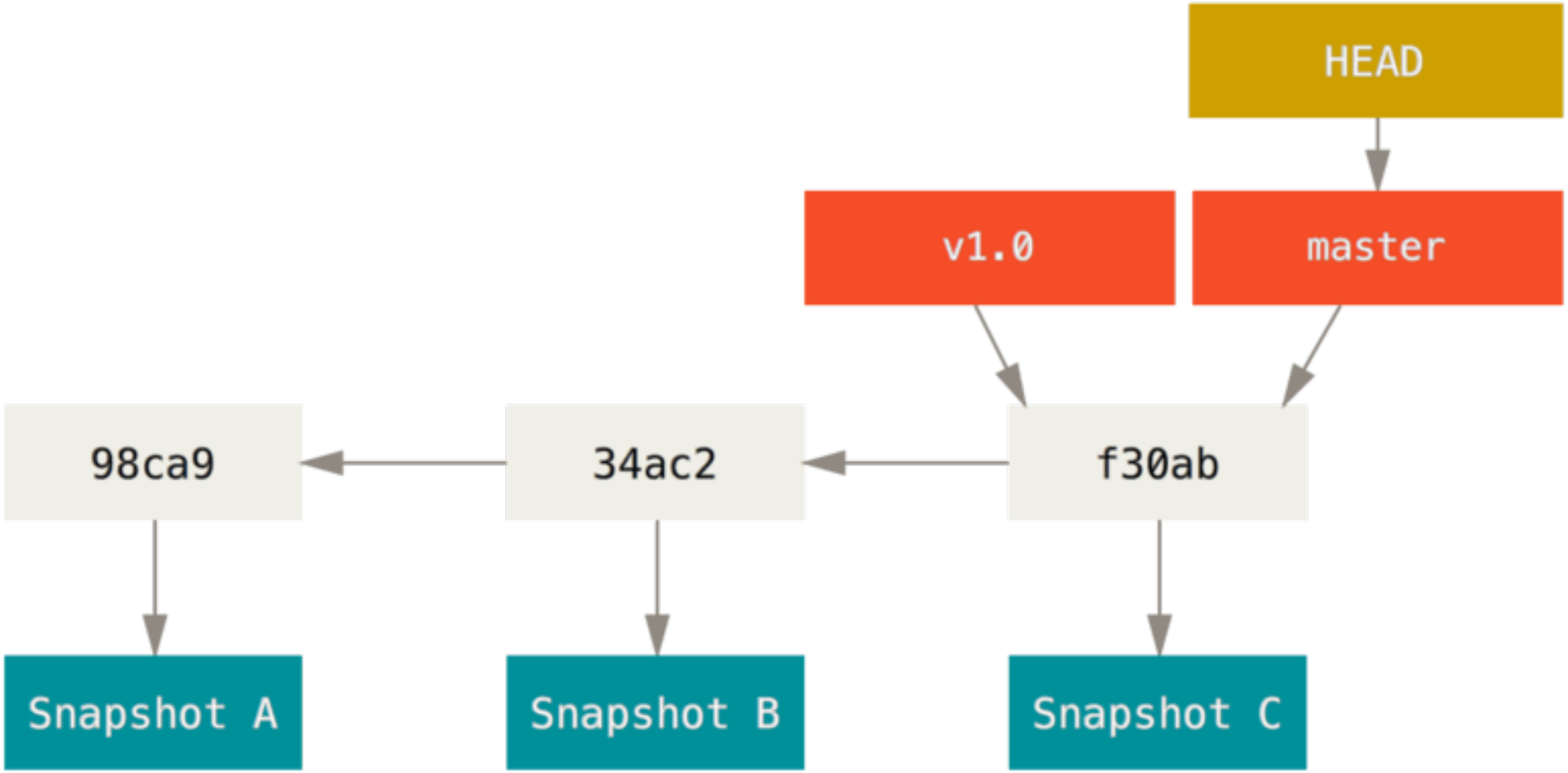
heig-vd

Liste chainée de commits



heig-vd

Branche = pointeur vers un commit



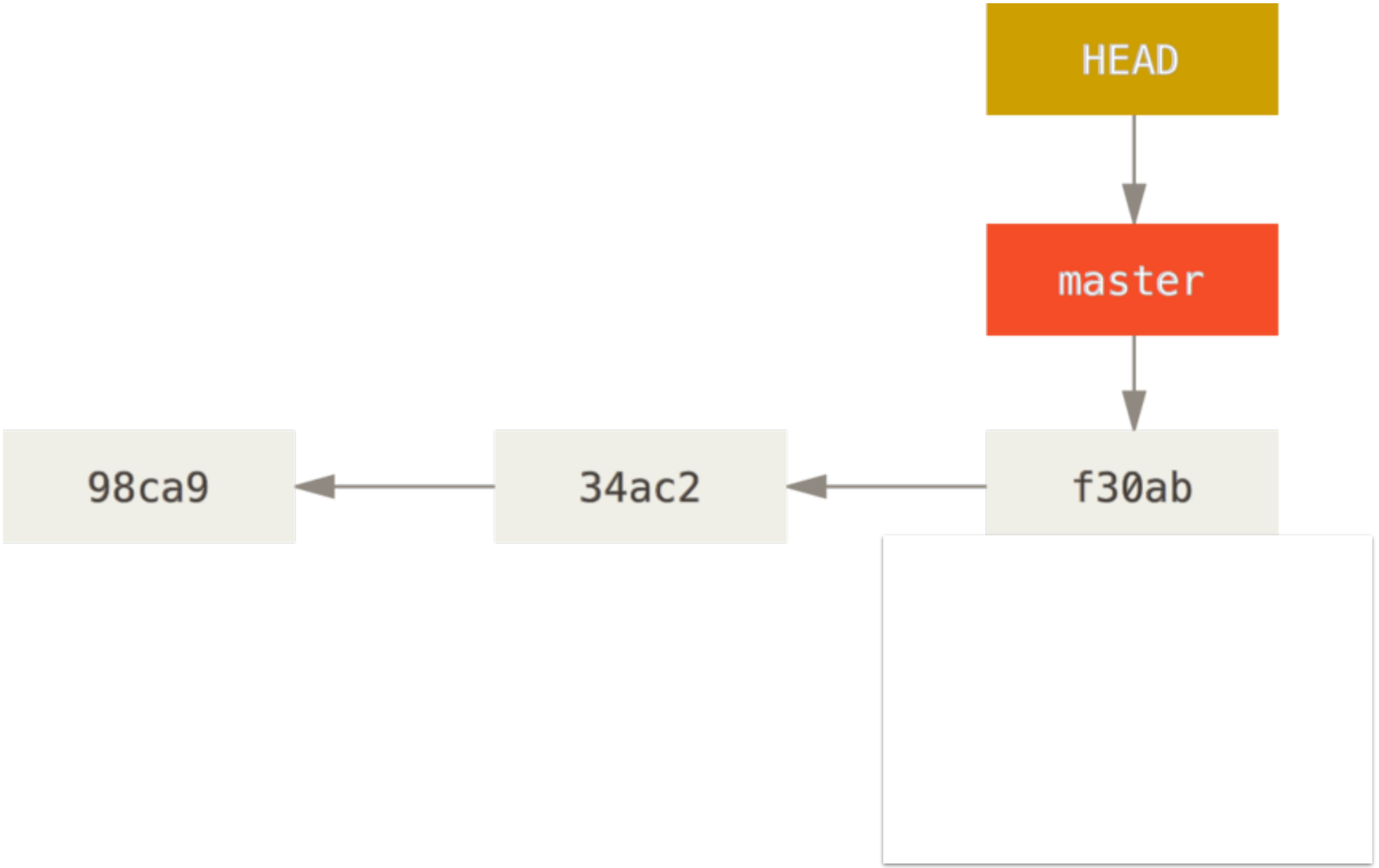
heig-vd

Les commandes git (4)

- `git branch`
  - Liste les branches de votre repo
- `git branch [branch-name]`
  - Crée une nouvelle branche depuis l'emplacement courant dans le repo
- `git checkout [branch-name]`
  - Aller dans la branche donnée et modifier le contenu local en conséquence

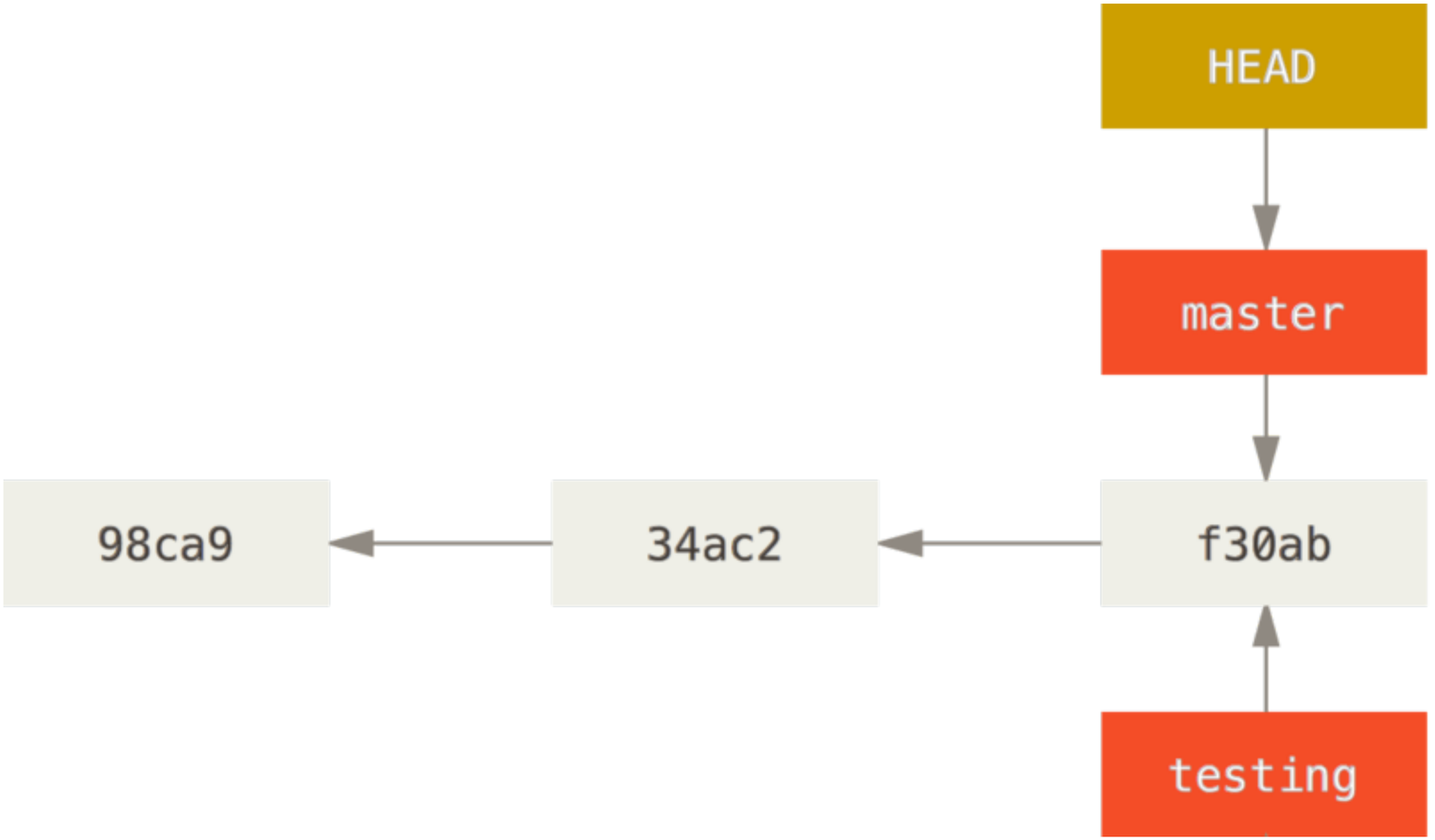
heig-vd

git branch testing



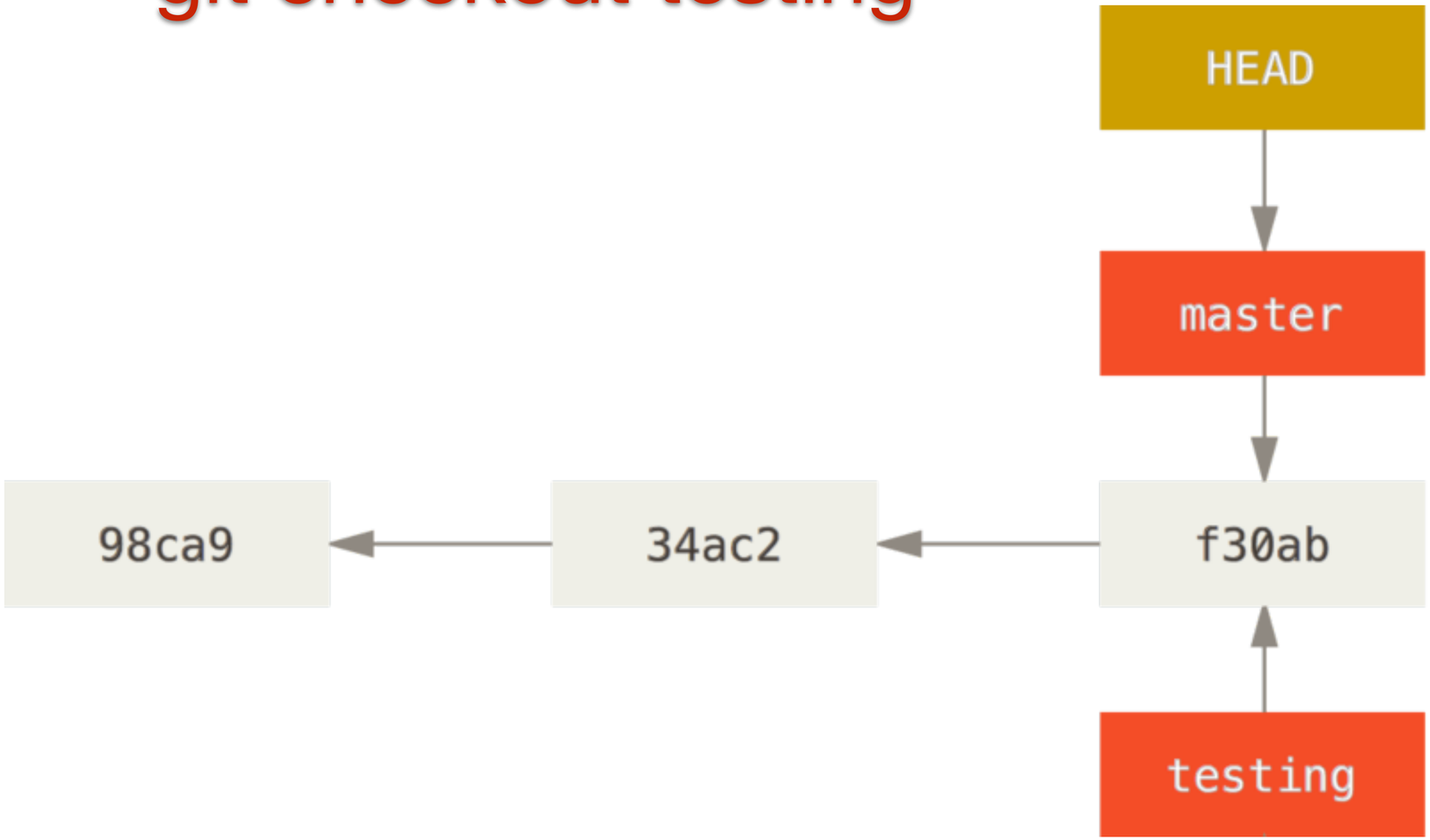
heig-vd

git branch testing



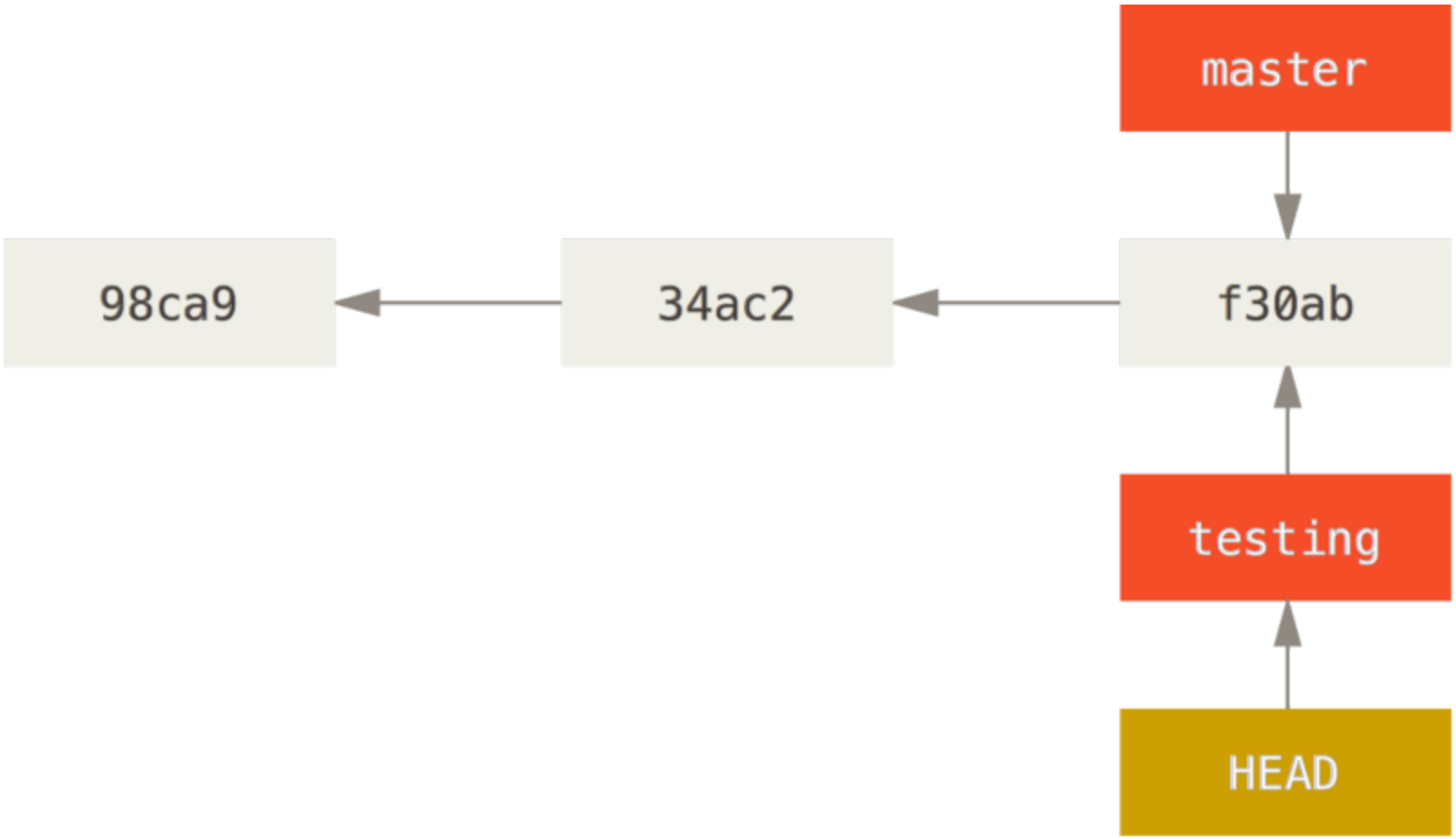
heig-vd

git checkout testing



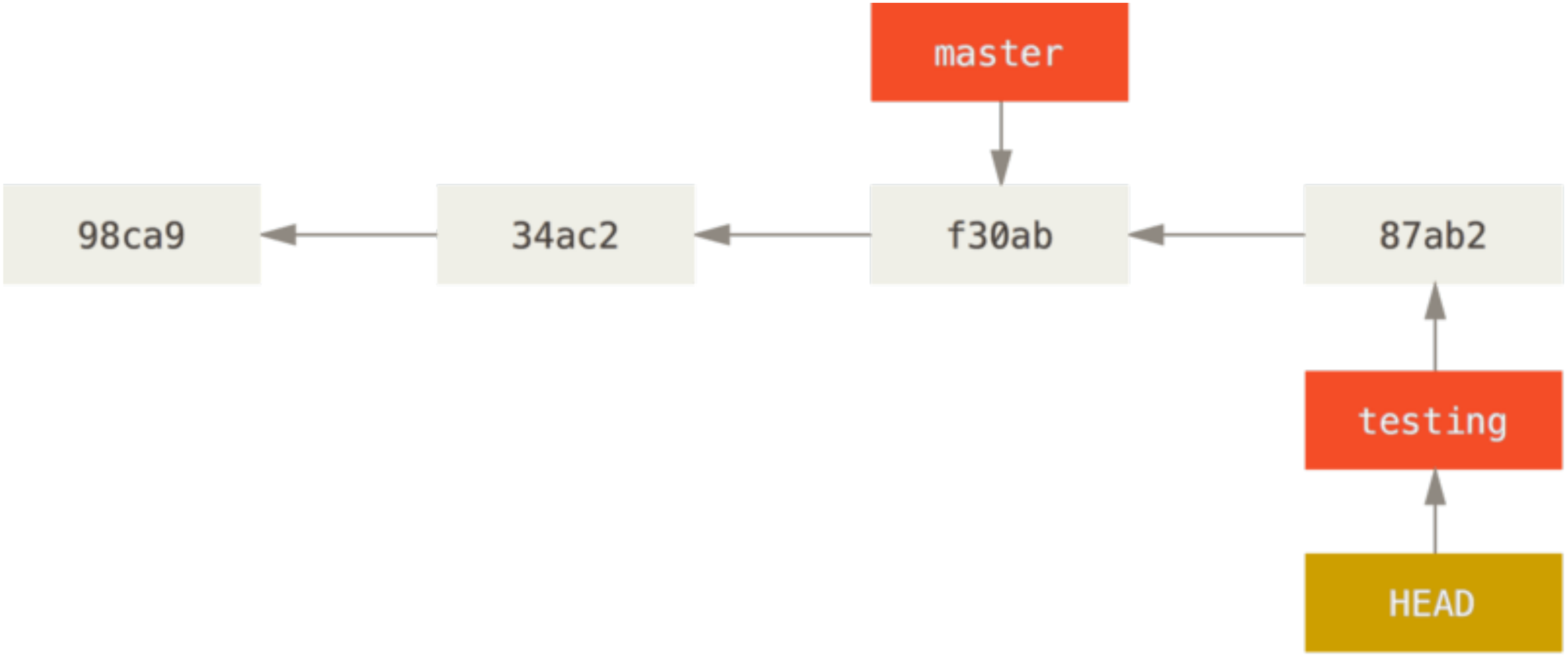
heig-vd

git checkout testing



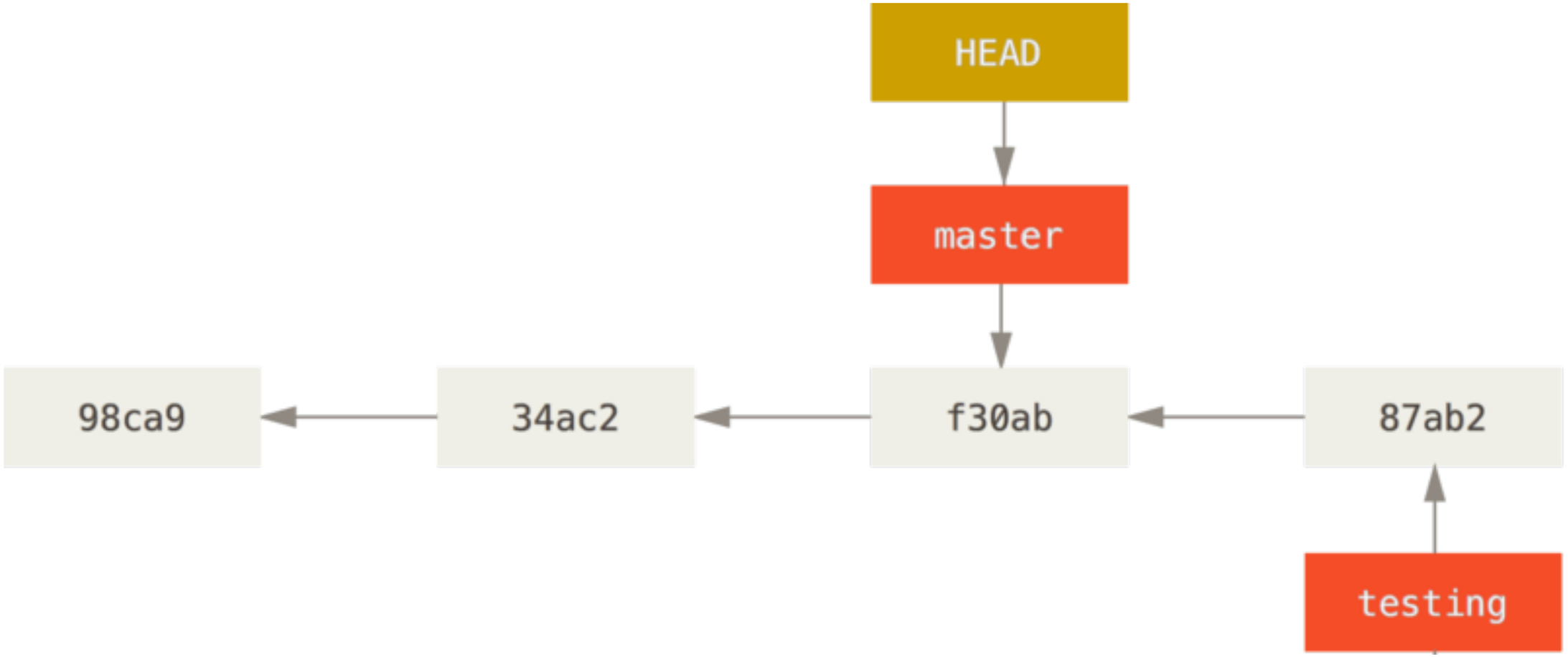
heig-vd

vim test.rb
git commit -a -m 'made a change'



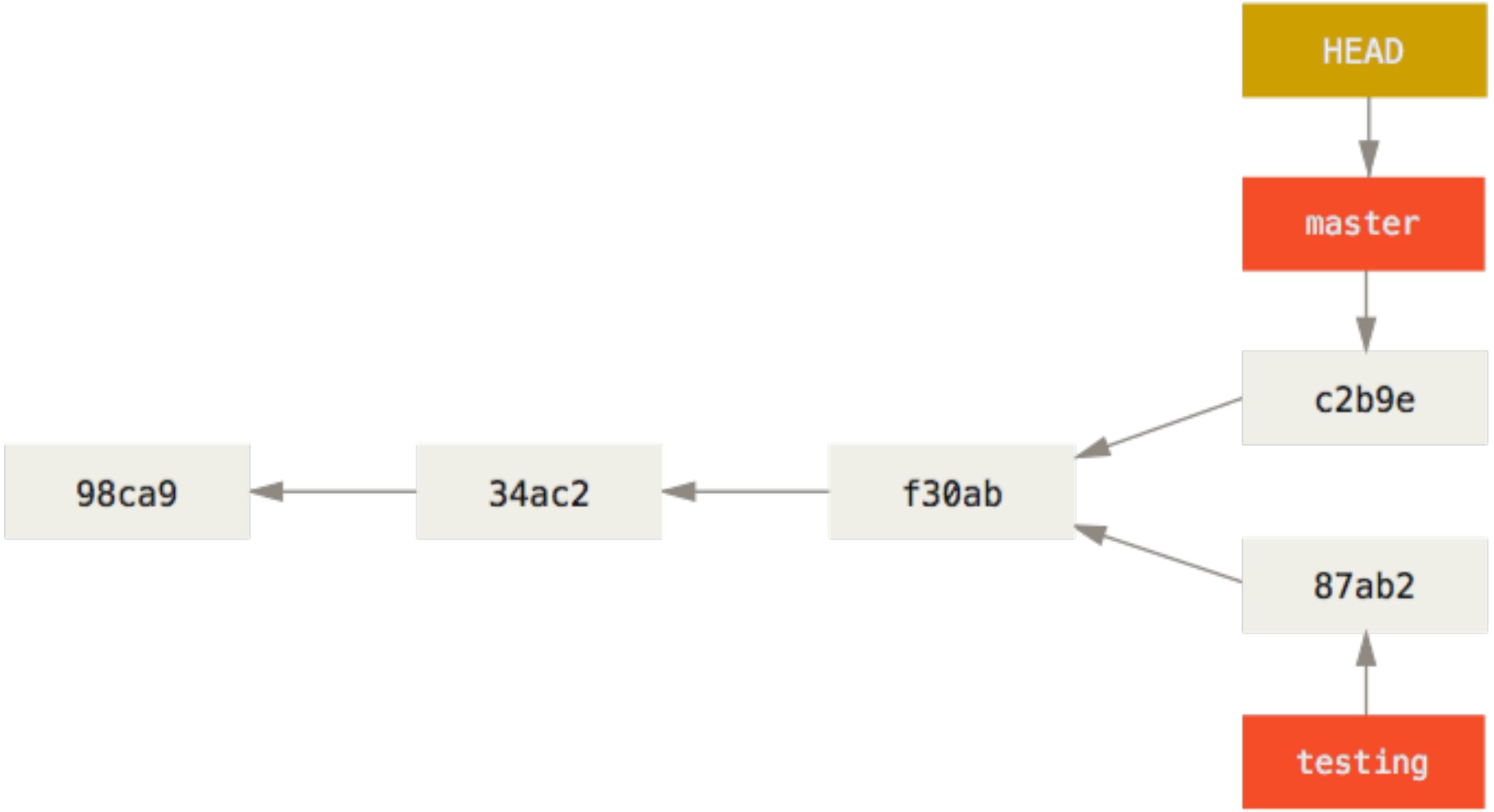
heig-vd

git checkout master



heig-vd

vim test.rb
git commit -a -m 'made other changes'



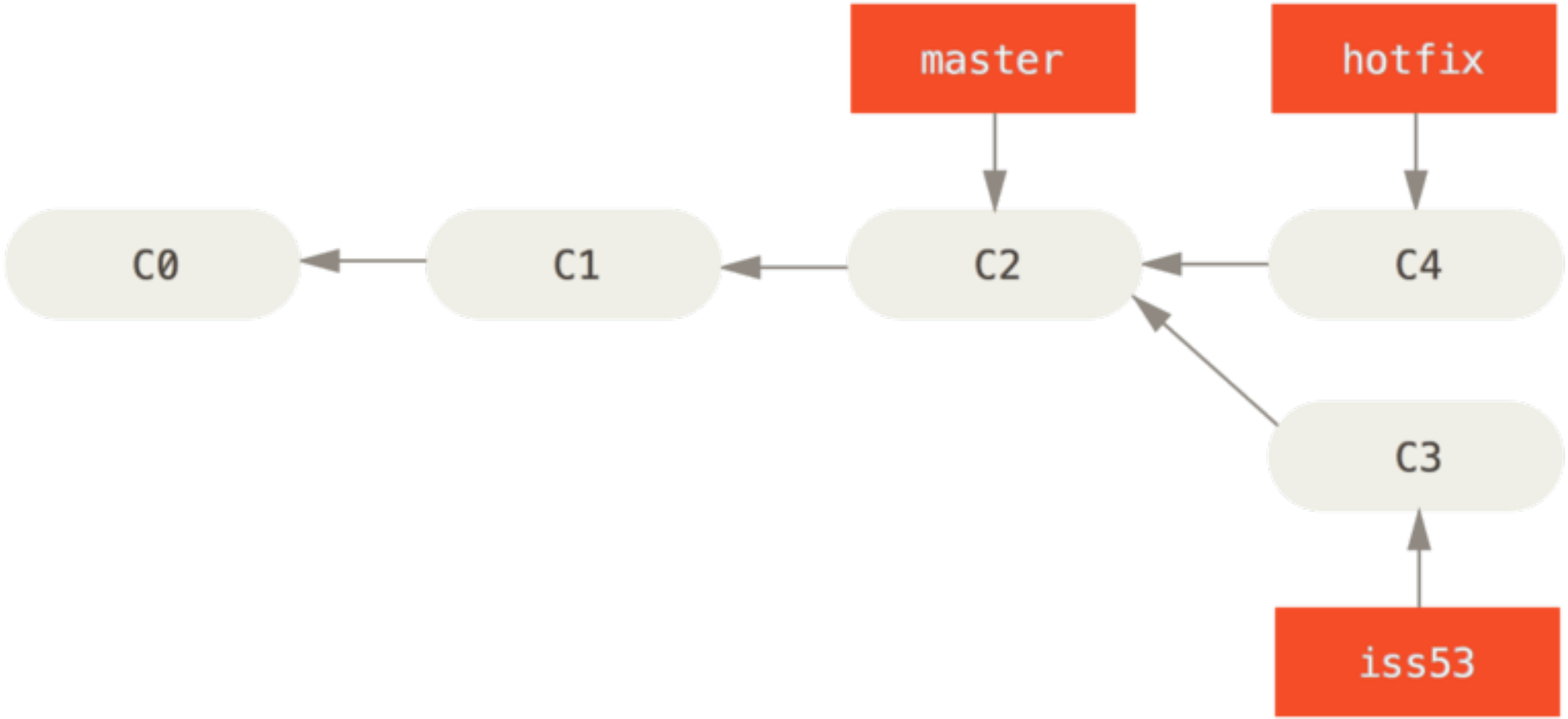
heig-vd

Les commandes git (5)

- `git merge [branch]`
  - Fusionne la branche nommée avec la branche courante

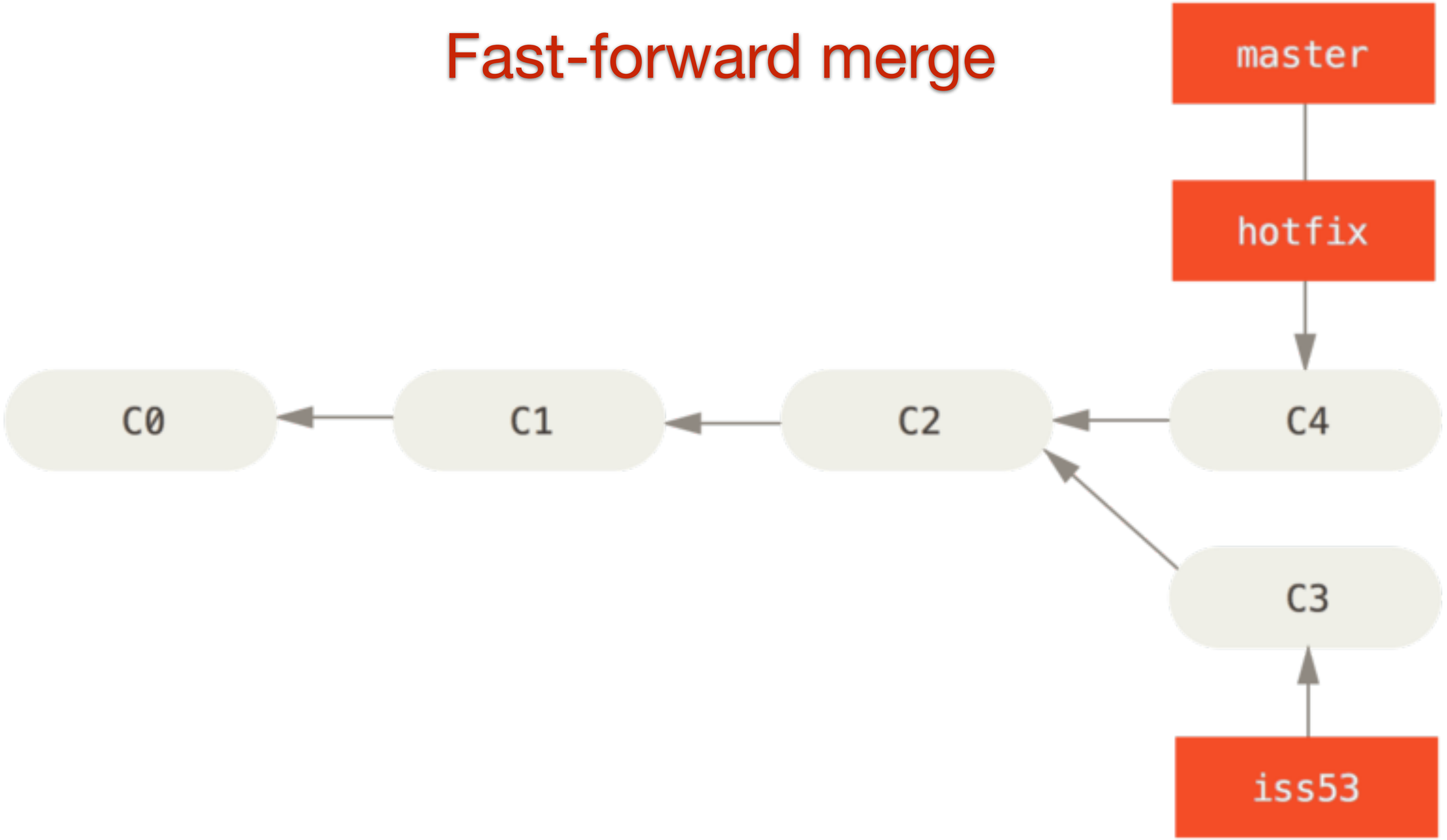
heig-vd

Fast-forward merge



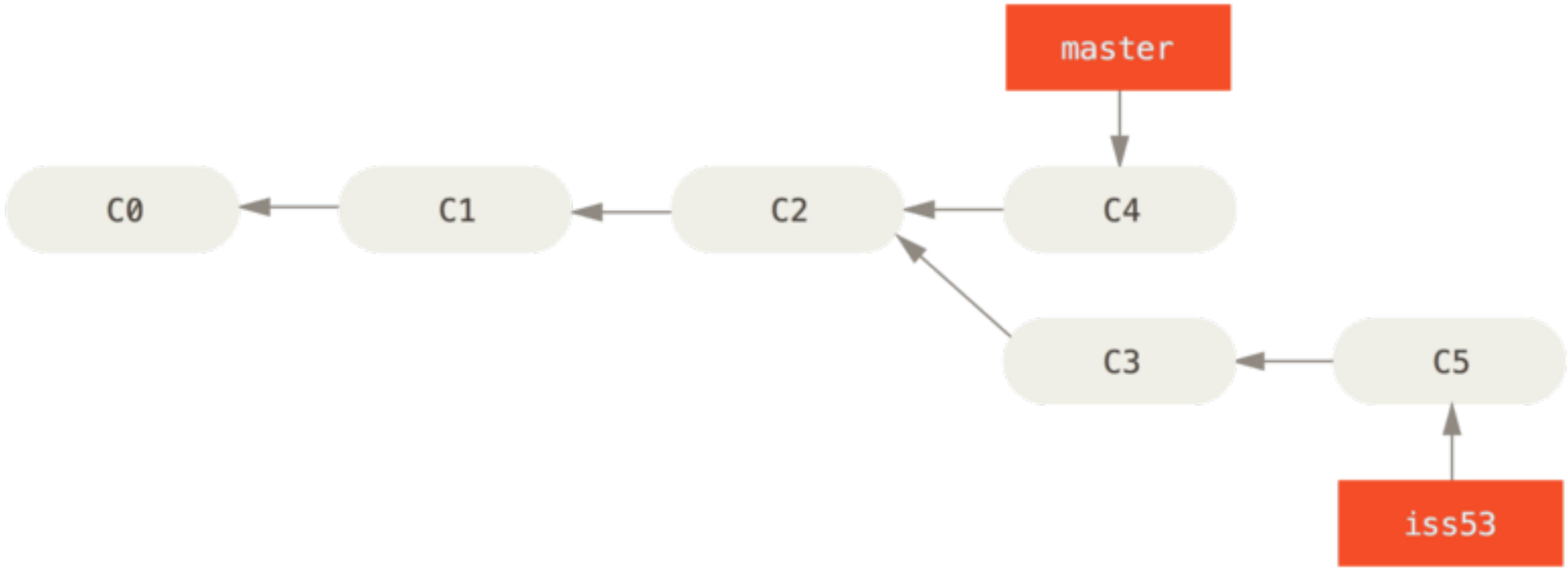
heig-vd

Fast-forward merge



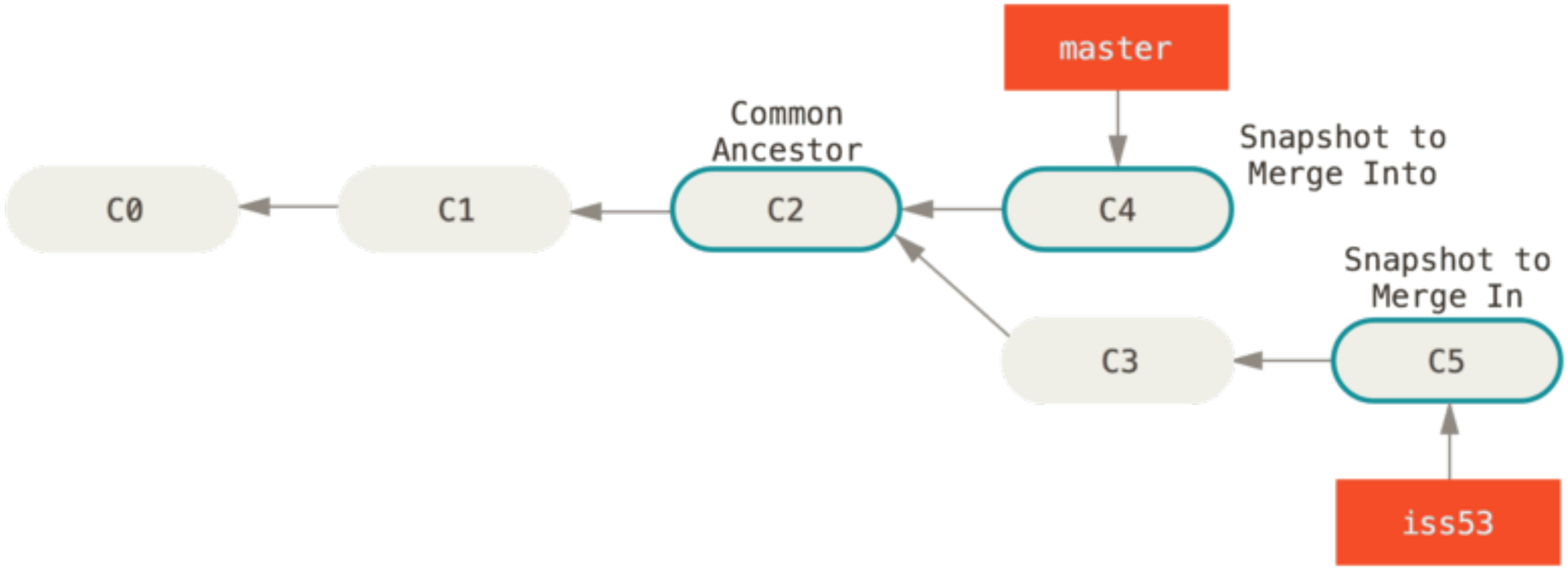
heig-vd

Basic merge



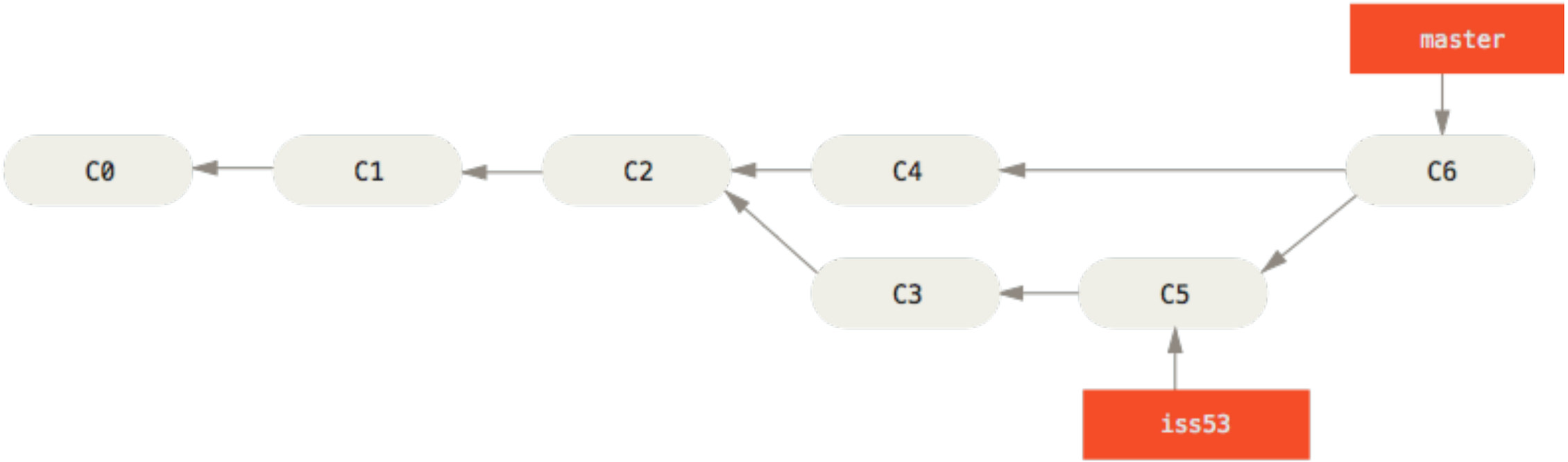
heig-vd

Basic merge



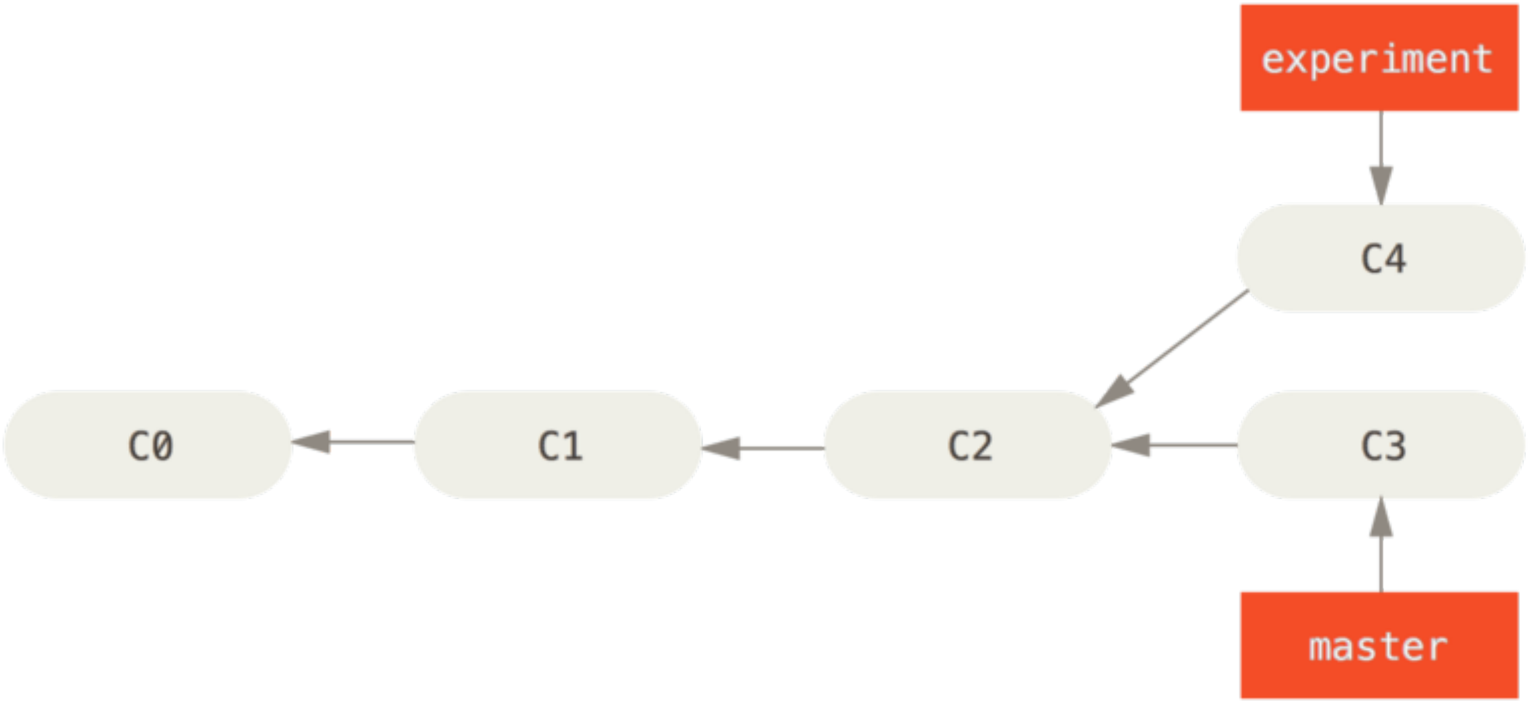
heig-vd

Basic merge



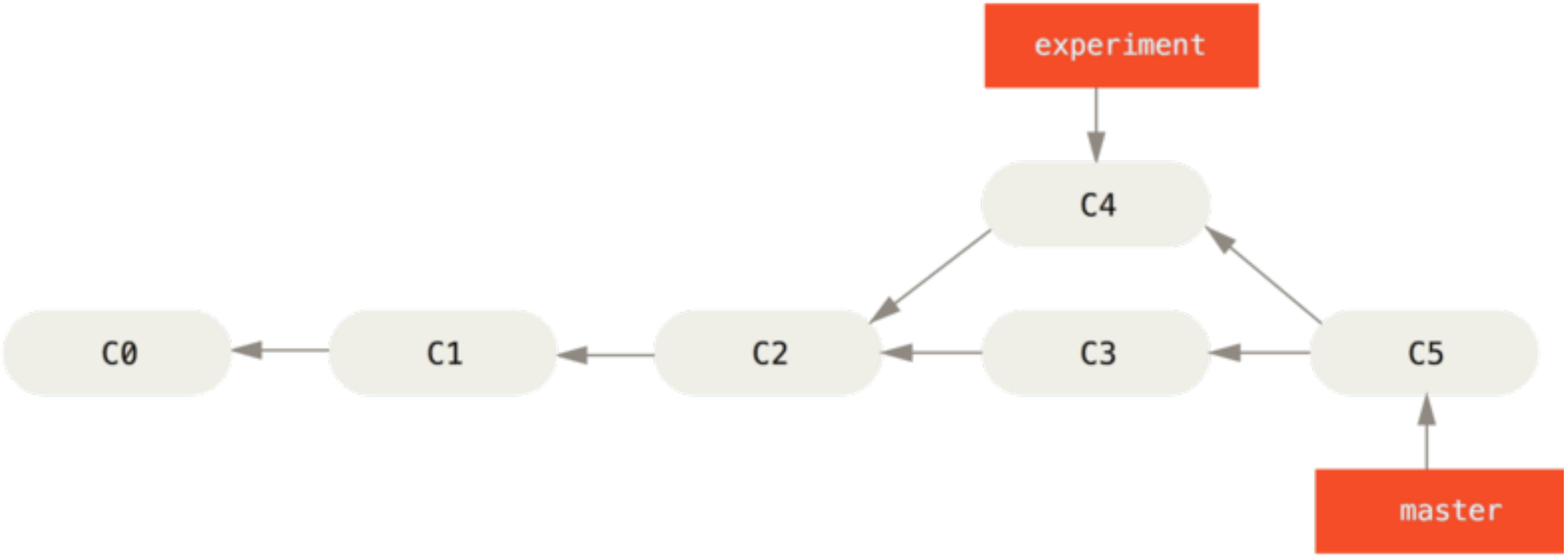
heig-vd

merge vs. rebase



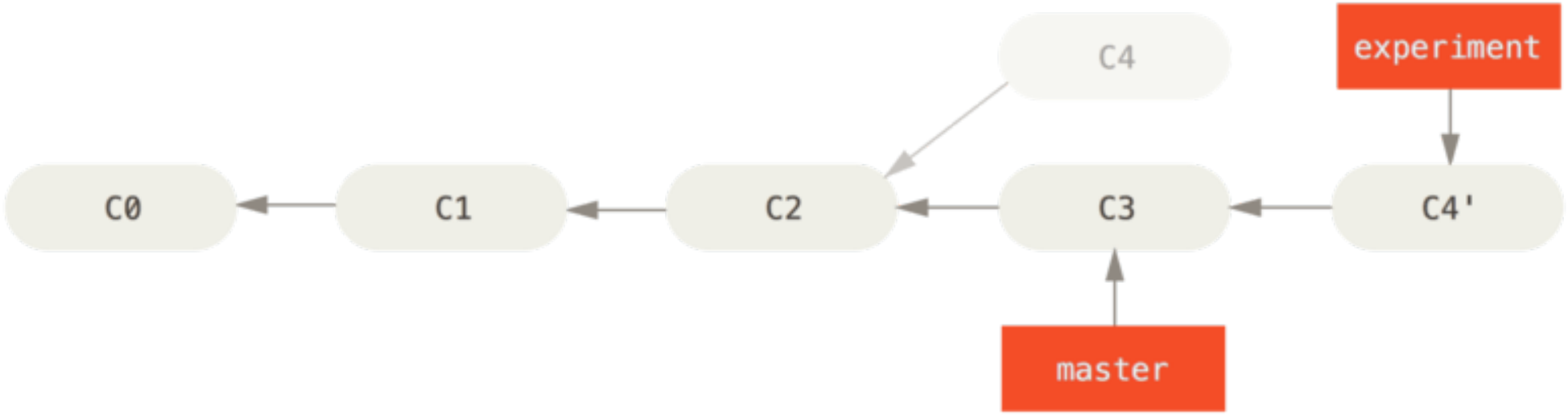
heig-vd

merge vs. rebase



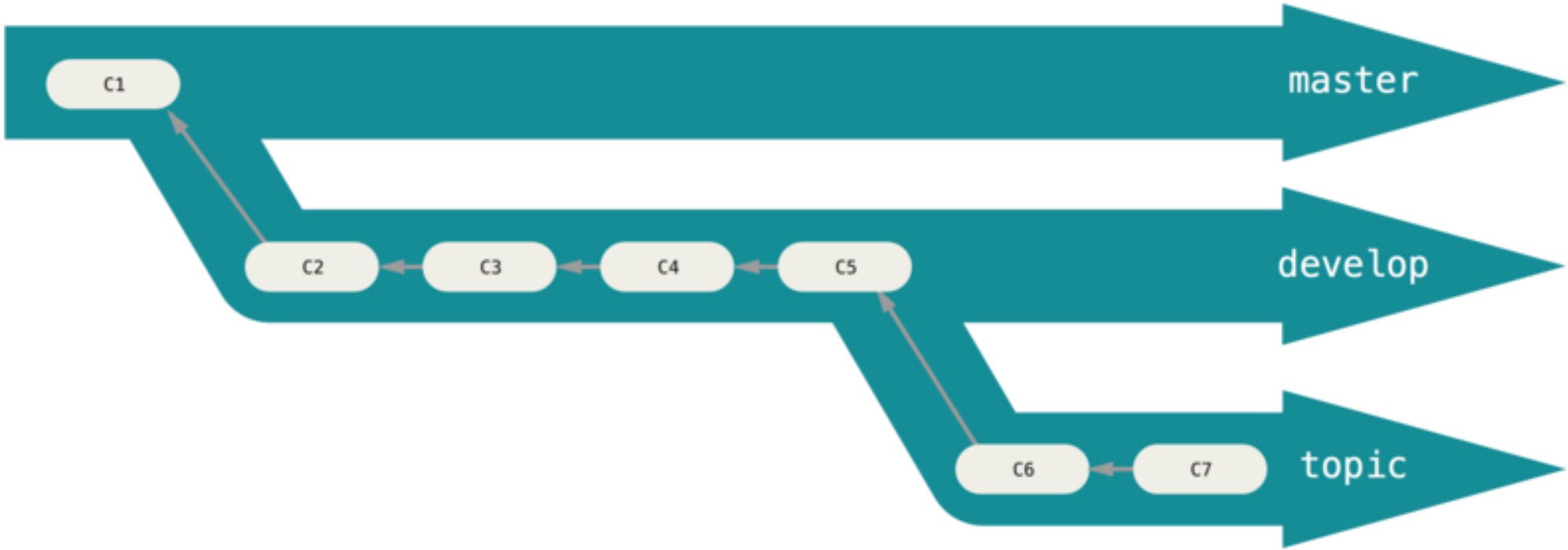
heig-vd

merge vs. rebase



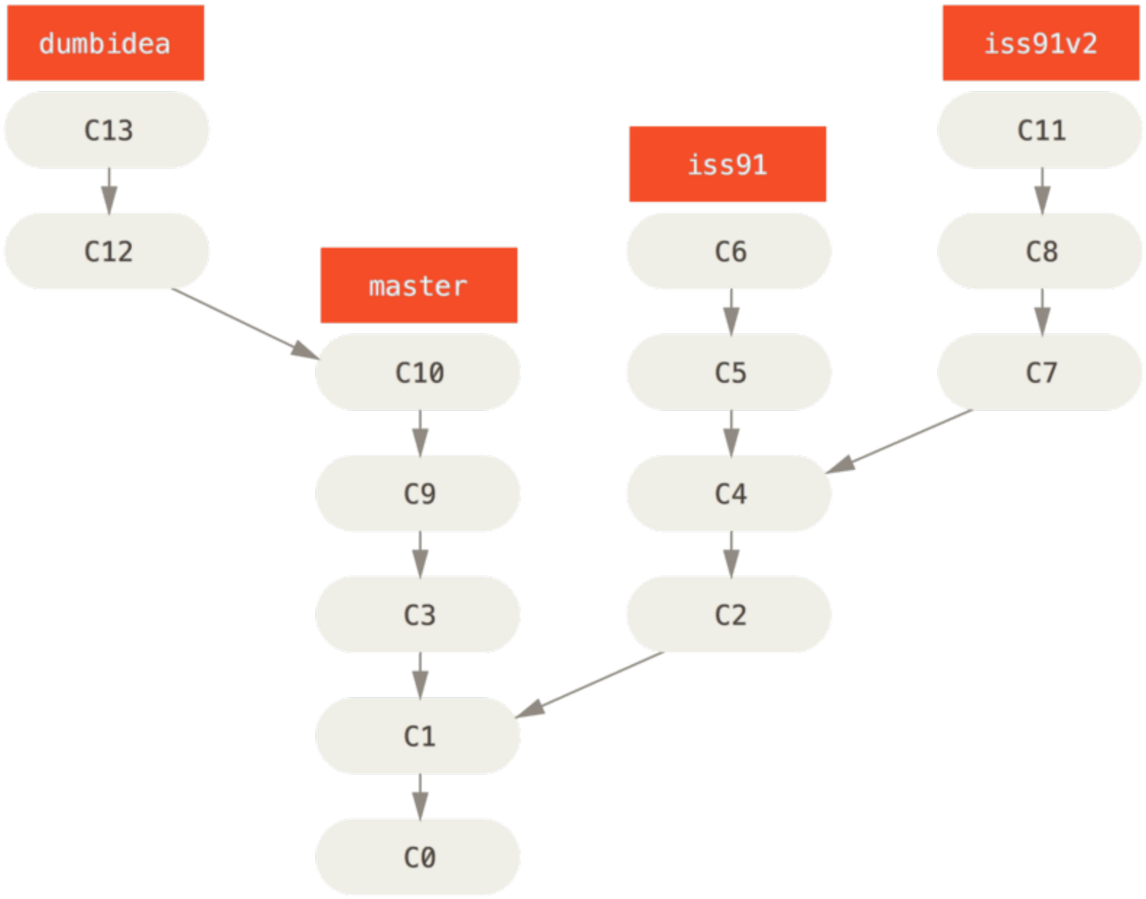
heig-vd

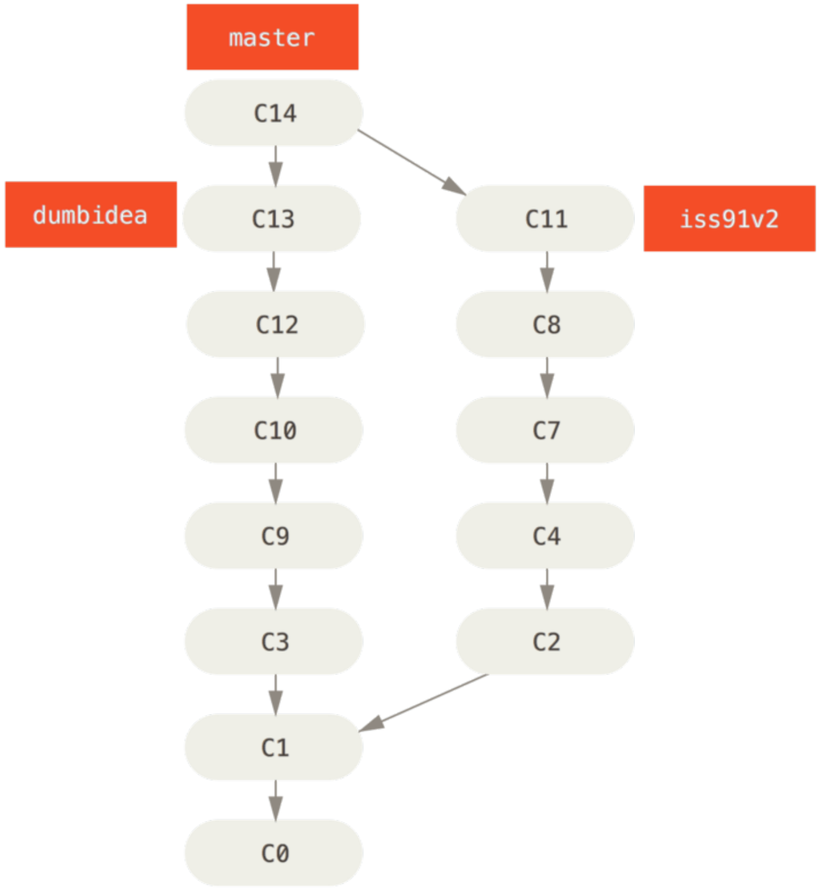
Branches aux long court



heig-vd

Branches thématiques



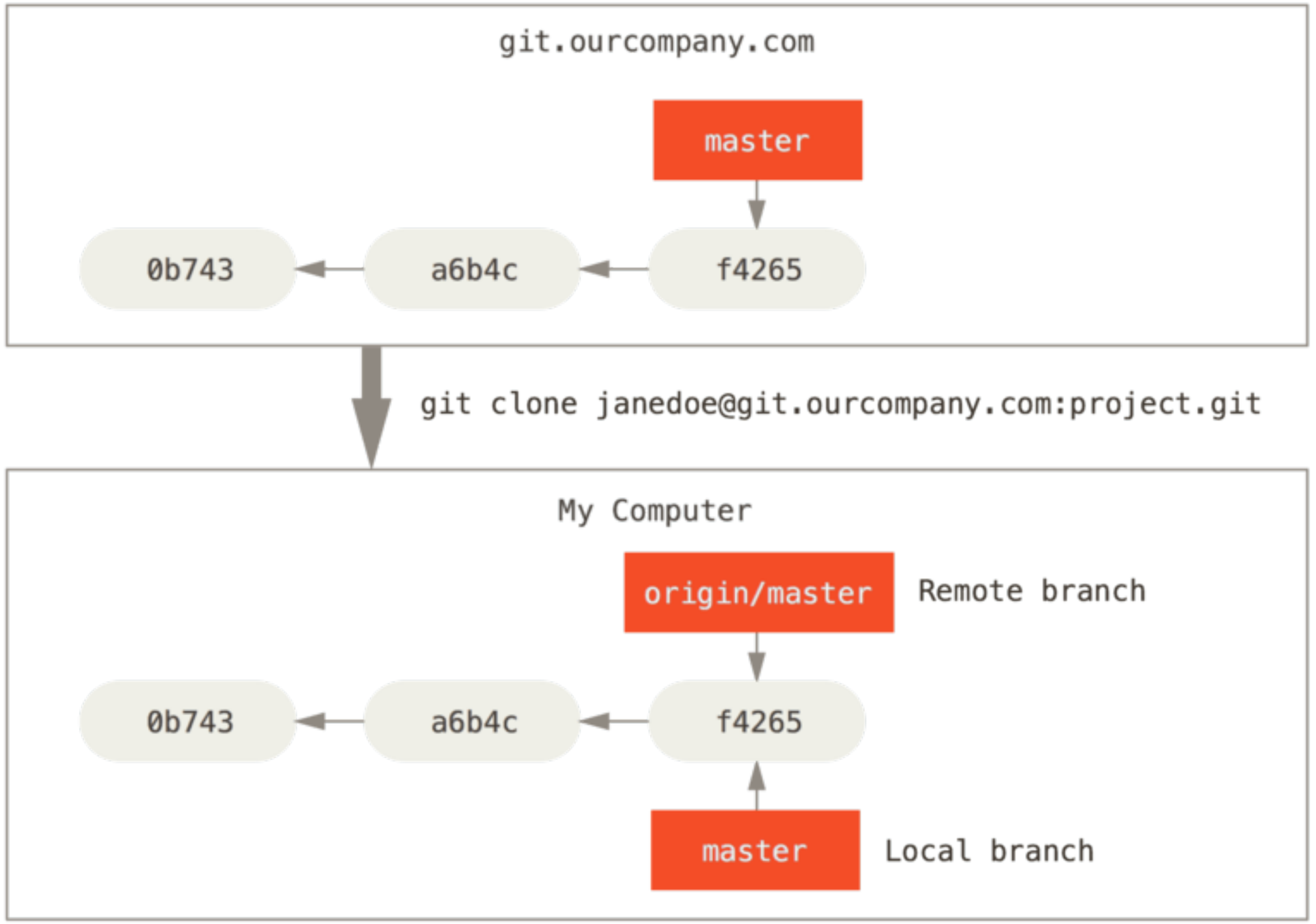


heig-vd

À distance…

heig-vd

Branches de suivi à distance



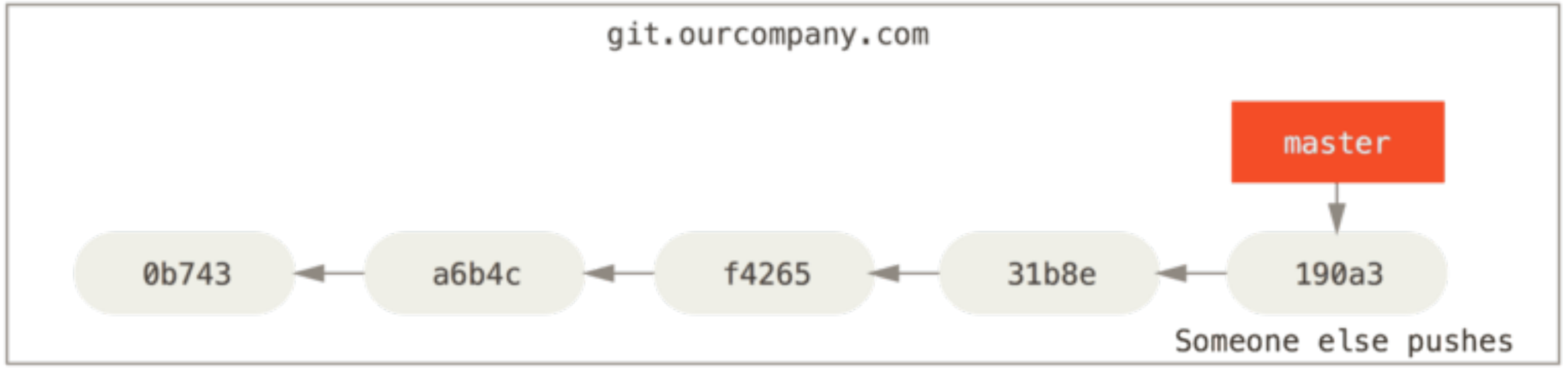
heig-vd

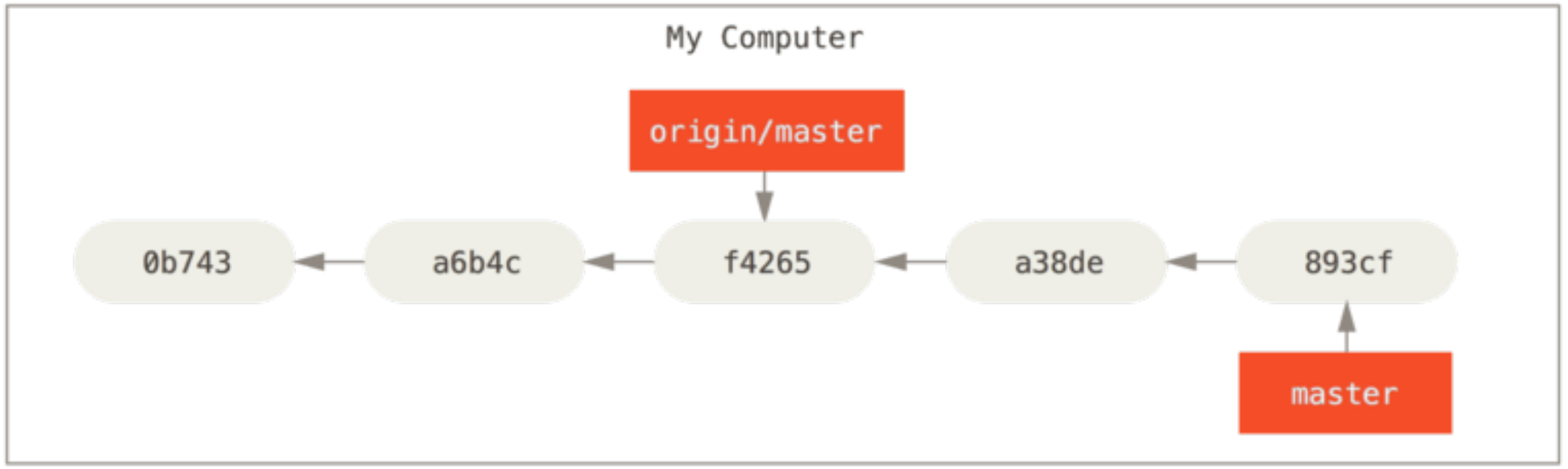
Les commandes git (6)

- `git push`
  - Transmet la branche locale à la branche à distance
  - Si c'est la première fois que l'on `push`, il faut écrire `git push -u origin [branch]` pour créer le lien entre la branche locale et la branche sur le repo à distance surnommé `origin`
- `git fetch [alias]`
  - Copie localement les branches du repo à distance `alias`
- `git pull`
  - Effectue un `fetch` puis un `merge`

heig-vd

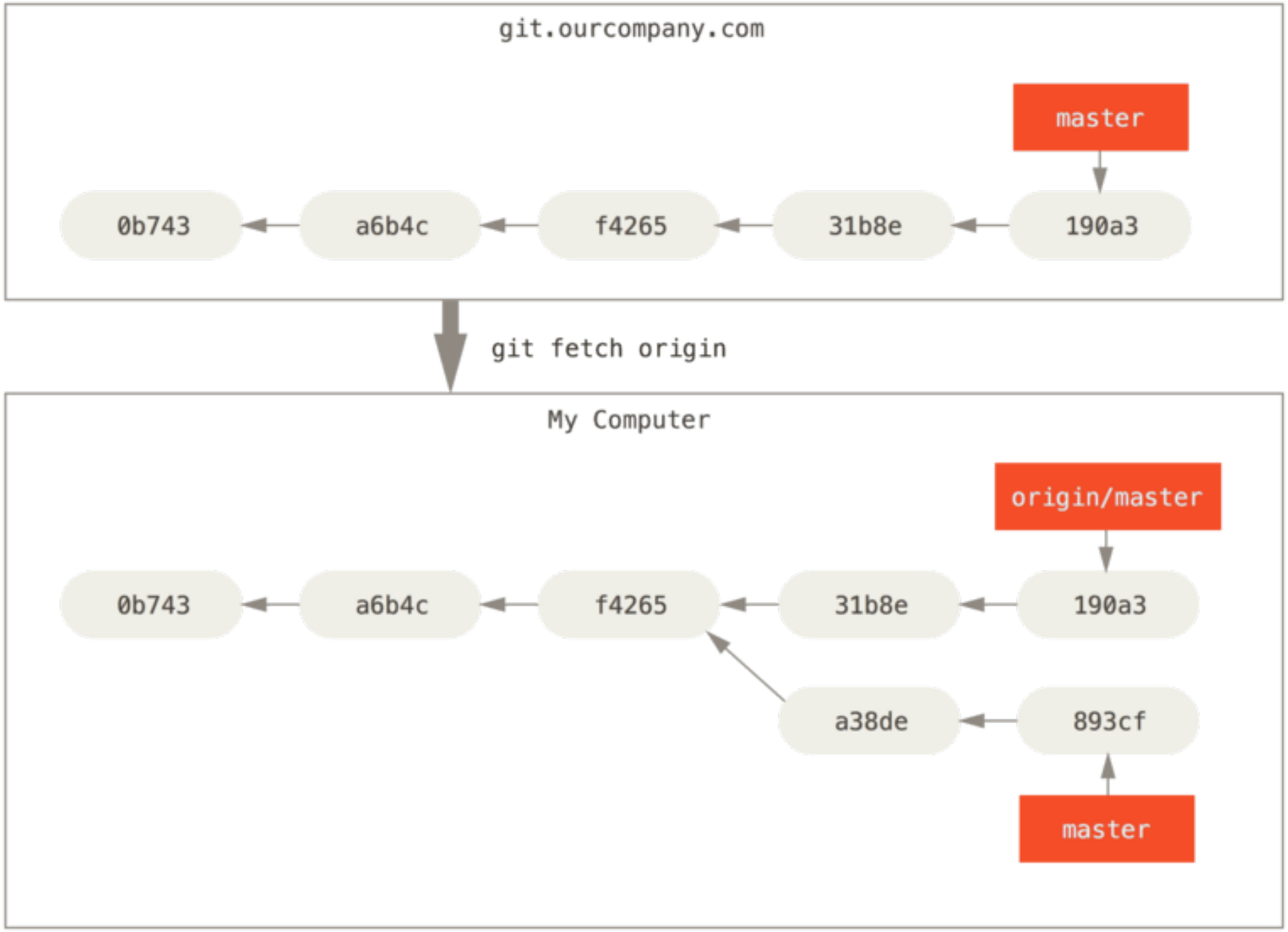
Branches de suivi à distance (2)





heig-vd

Branches de suivi à distance (3)

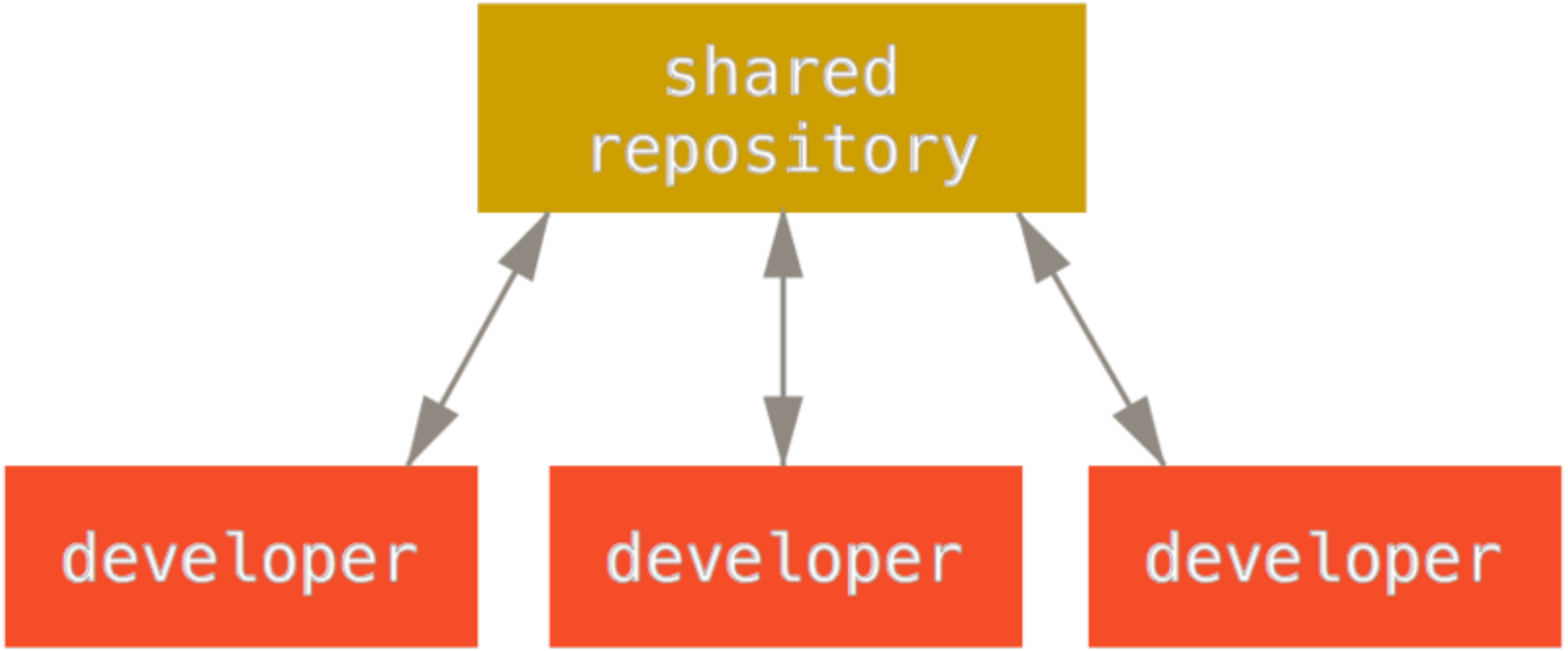


heig-vd

Git distribué

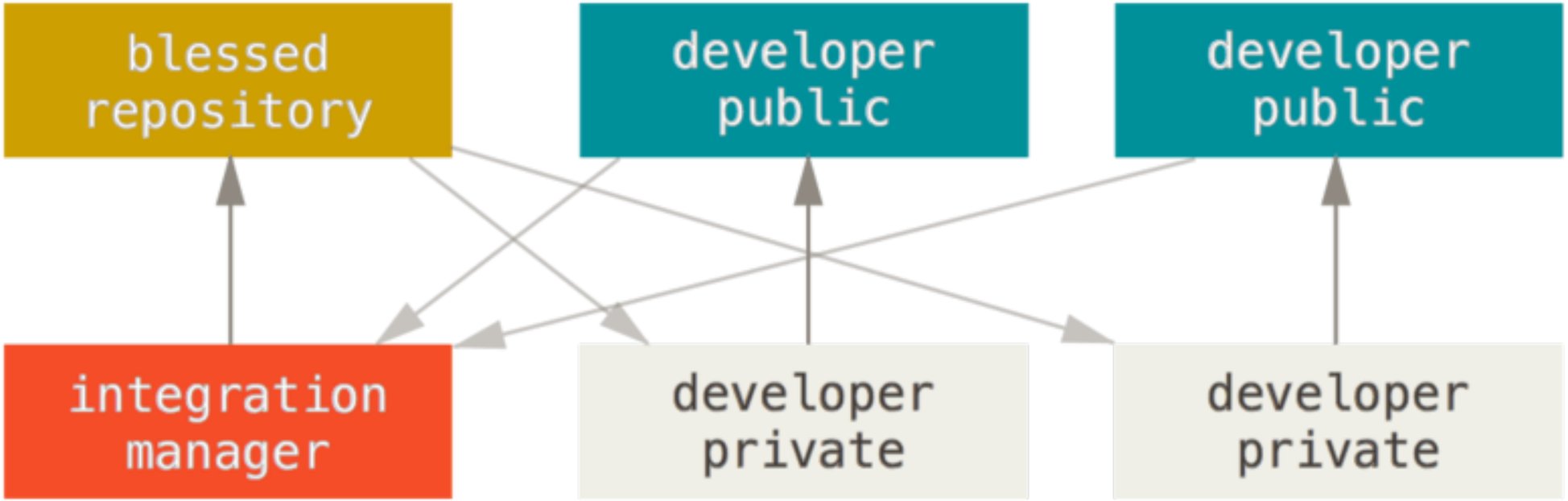
heig-vd

Gestion centralisée



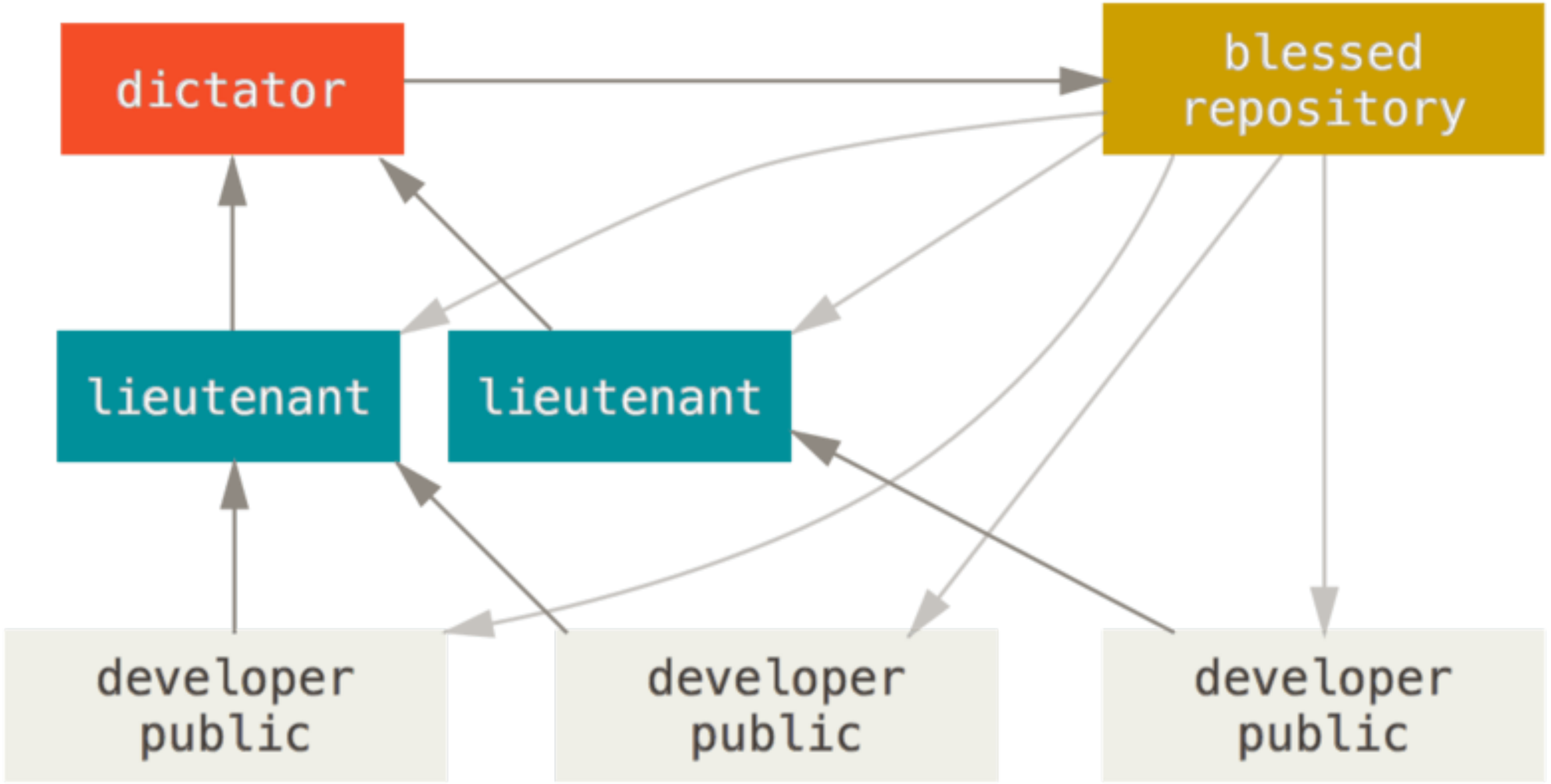
heig-vd

Gestionnaire d'intégration



heig-vd

Dictateur et lieutenants



heig-vd

Et plus encore…

- Cette présentation ne couvre pas l'utilisation de fork et de pull_request sur GitHub.

- Vous pouvez en appendre plus sur ce sujet ici …

`https://docs.github.com/en/github/collaborating-with-issues-and-pull-requests`