Git Basics—Distributed Version-control

Software Engineering II
WS 2020/21

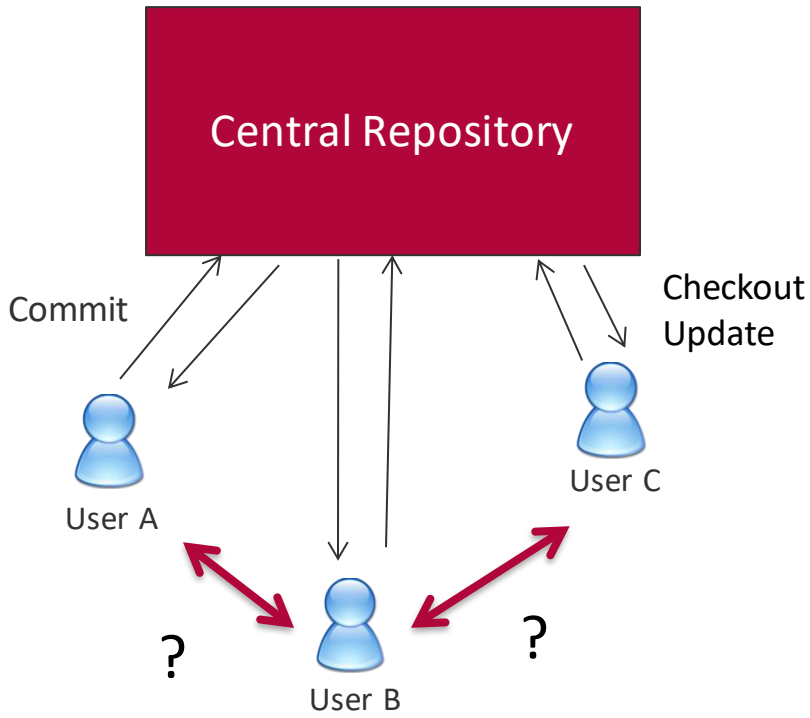
Enterprise Platform and Integration Concepts

Outline

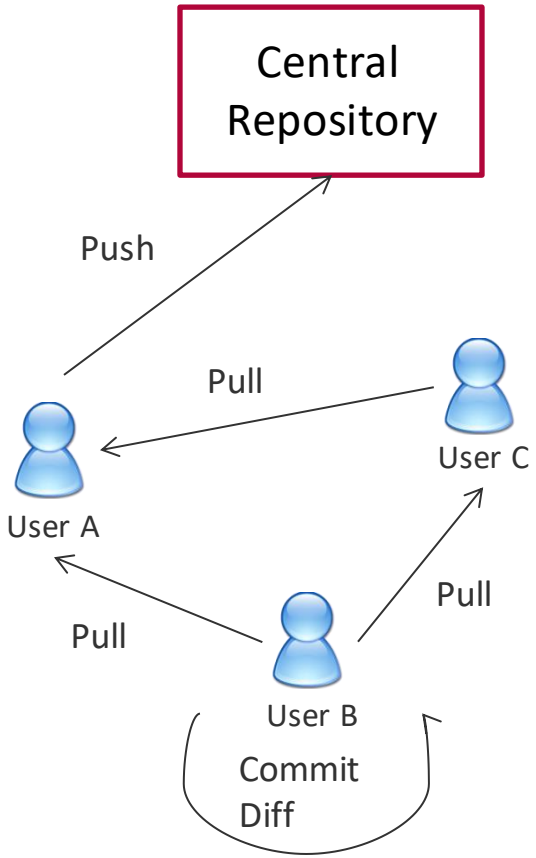
1. Basics
2. Local
3. Collaboration

https://xkcd.com/1597/

Centralized vs Distributed VCS



VS.



Centralized vs Distributed VCS

- Distributed VCS are mostly used like centralized VCS
  - Same features (branches, tags, merging)
- Local commits are a blessing and a curse
  - Commits can be made while offline
  - Higher chances of code diverging
- Pull-Requests are better than patch files

Git Objects

Blob

- Content of a file
- Nothing else

Commit

- References Tree object
- Metadata
- 0..* parent commits

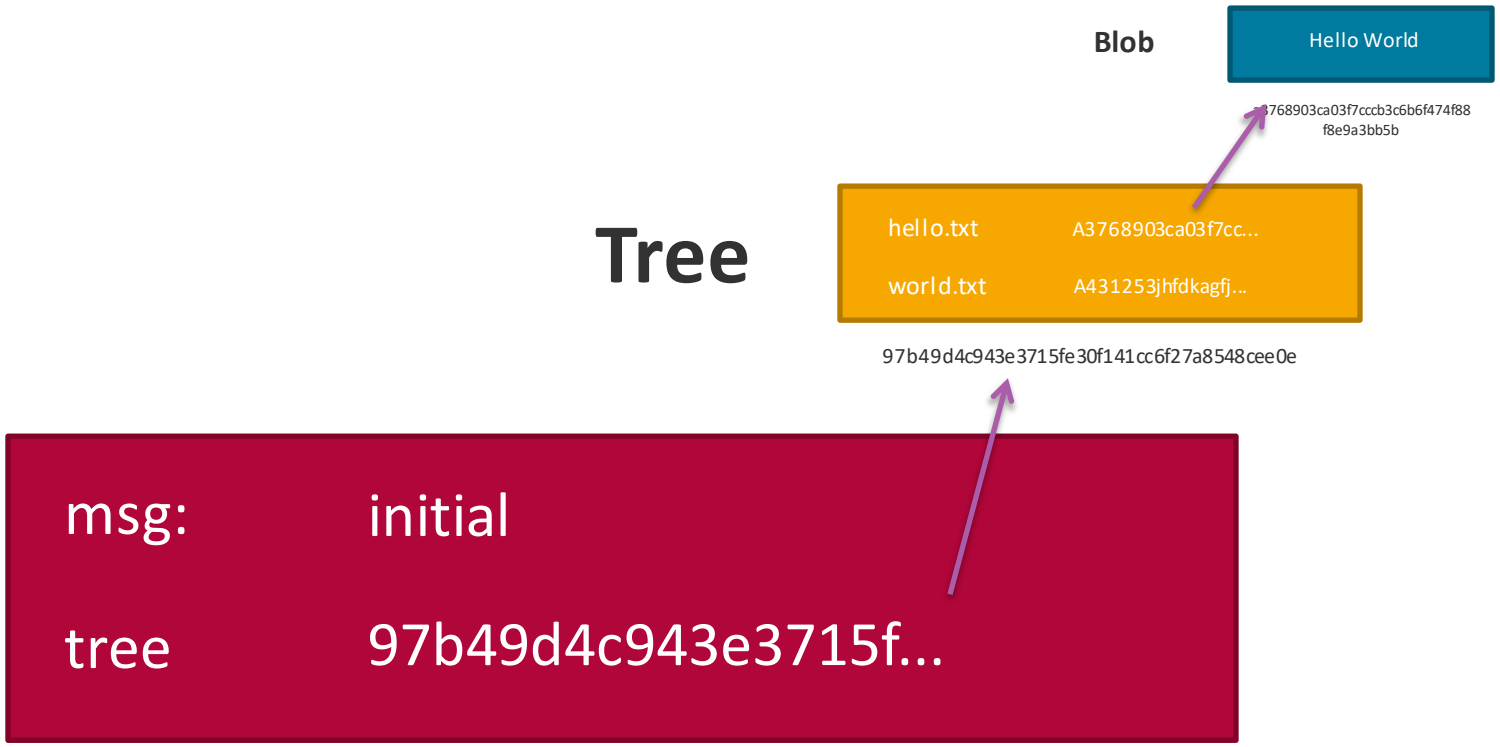
Tree

- File structure
- References Blobs

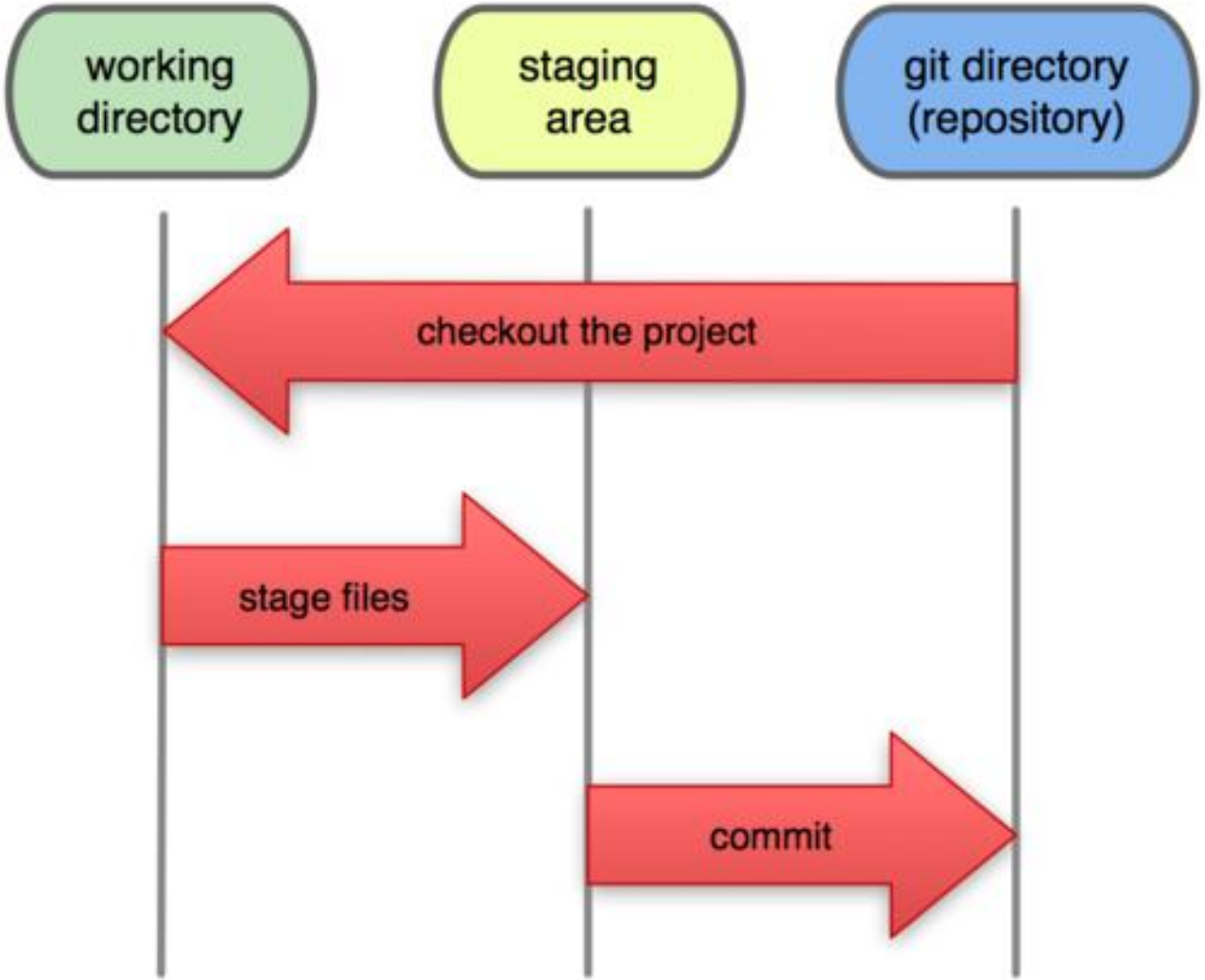
Tag

- Reference to other object

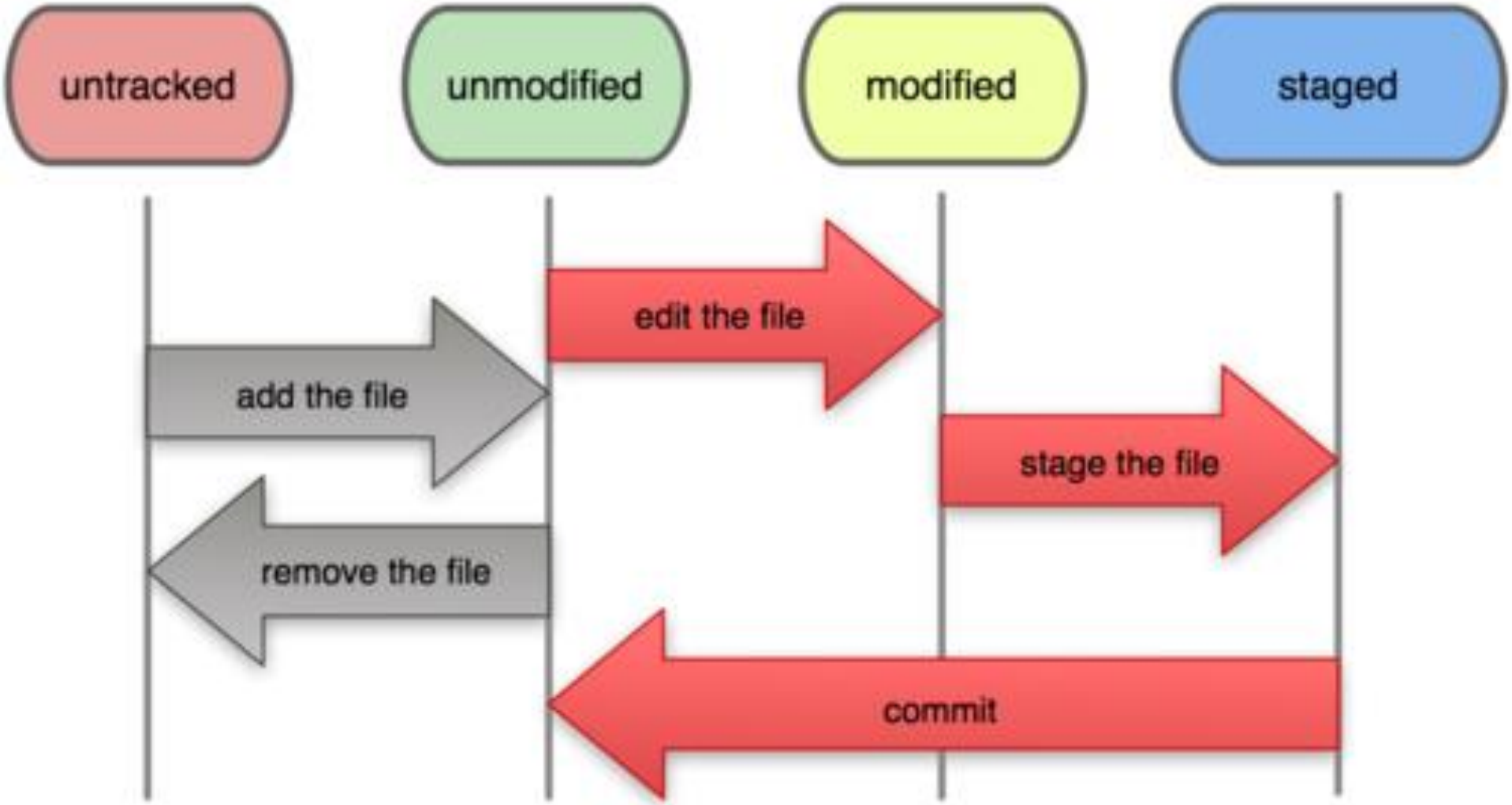
Commit


Blob
Hello World
A3768903ca03f7cccb3c6b6f474f88
f8e9a3bb5b
Tree
hello.txt A3768903ca03f7cc...
world.txt A431253jhfdkagfj...
97b49d4c943e3715fe30f141cc6f27a8548cee0e
msg: initial
tree 97b49d4c943e3715f...
a3768903ca03f7cccb3c6b6f474f88f8e9a3bb5b

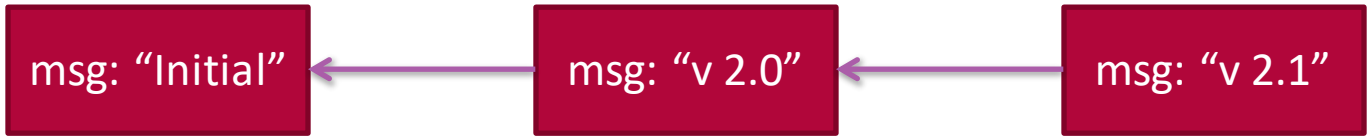
Local Operations



File Status Lifecycle

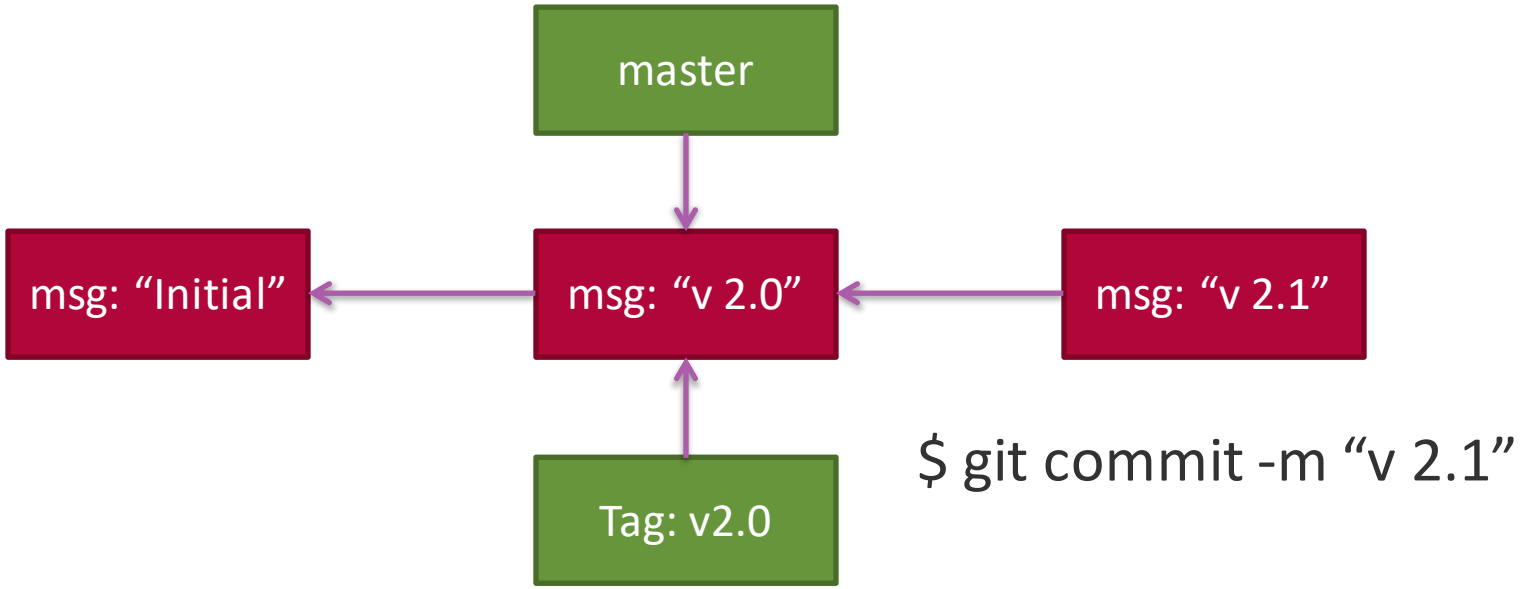


Commit Parent



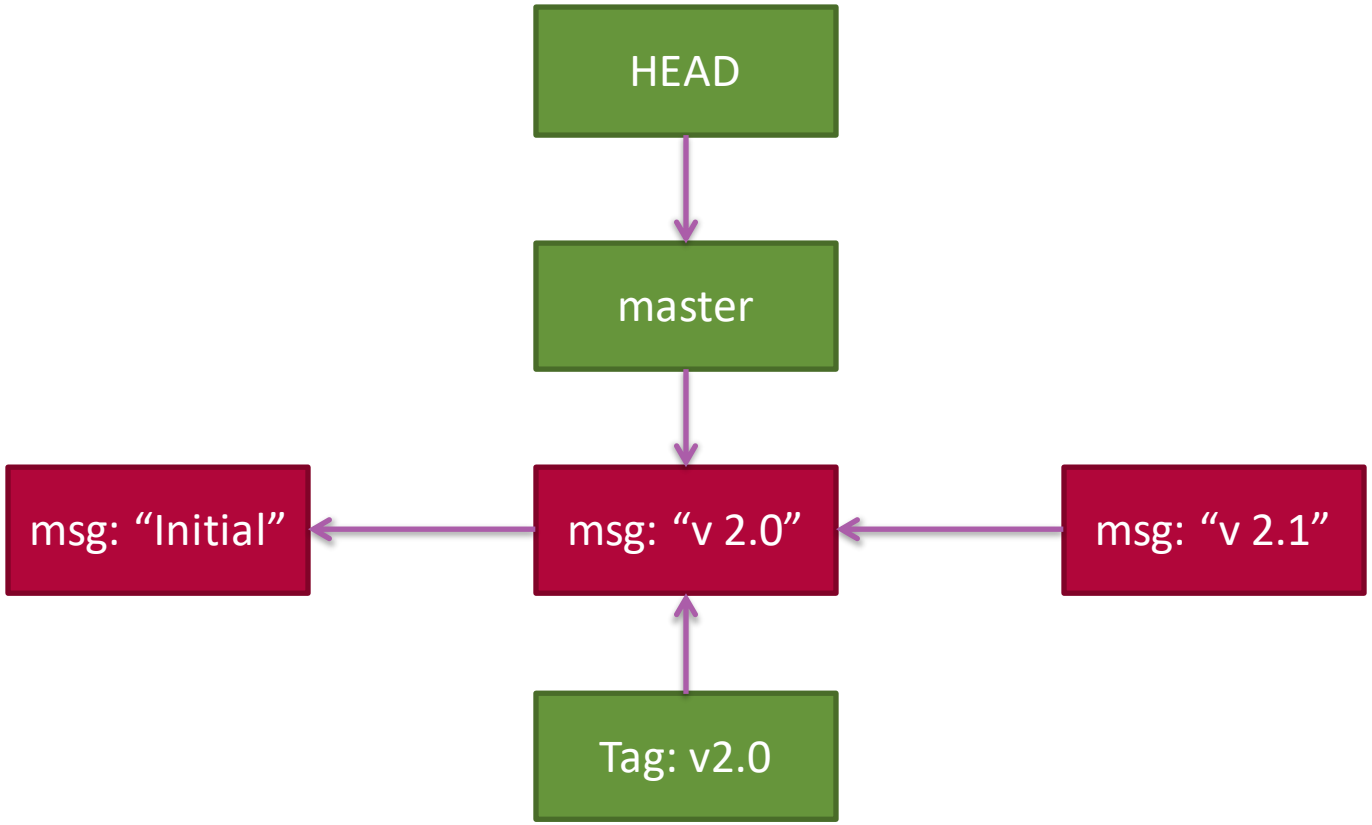
$ git commit -m “v 2.1”

Branches & Tags



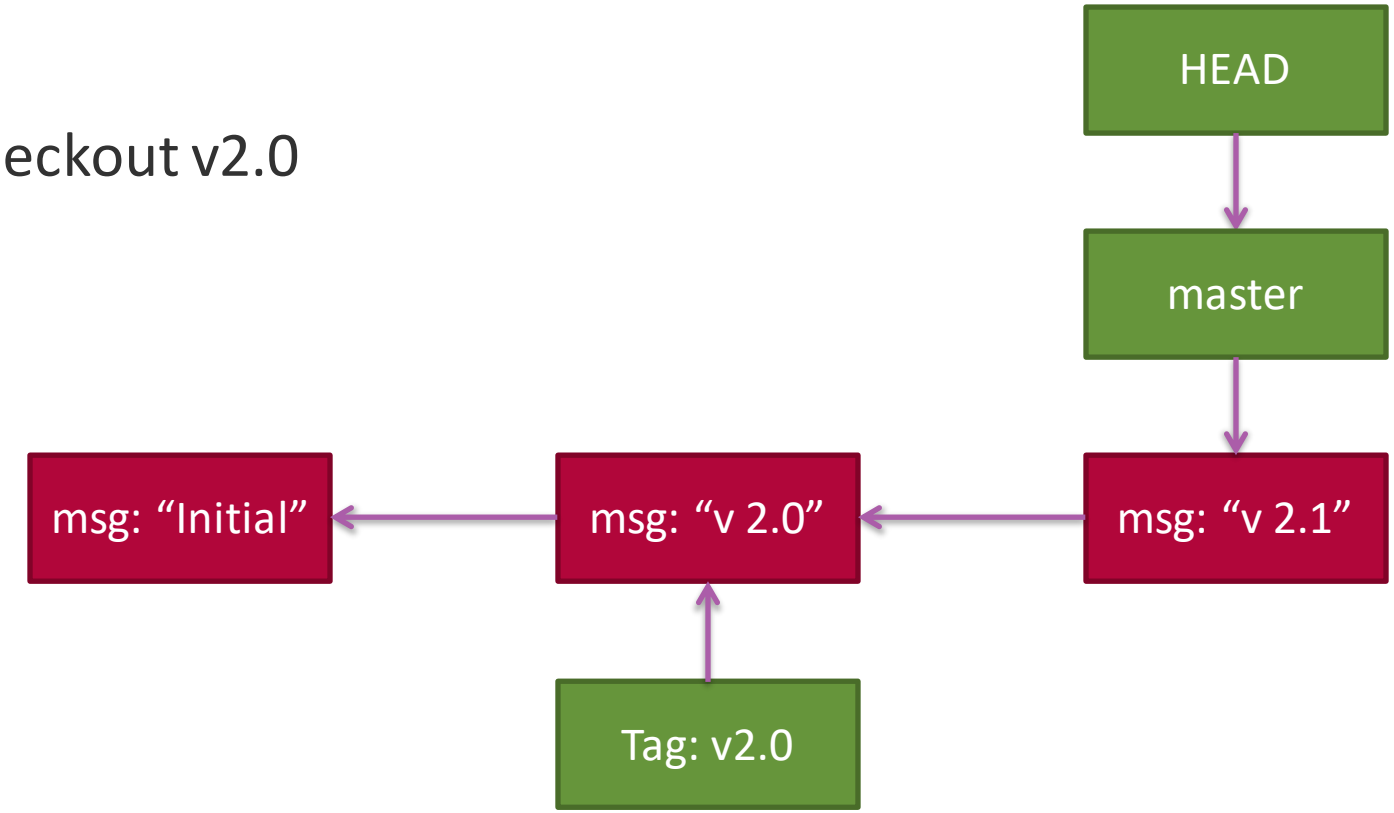
$ git commit -m “v 2.1”

Head



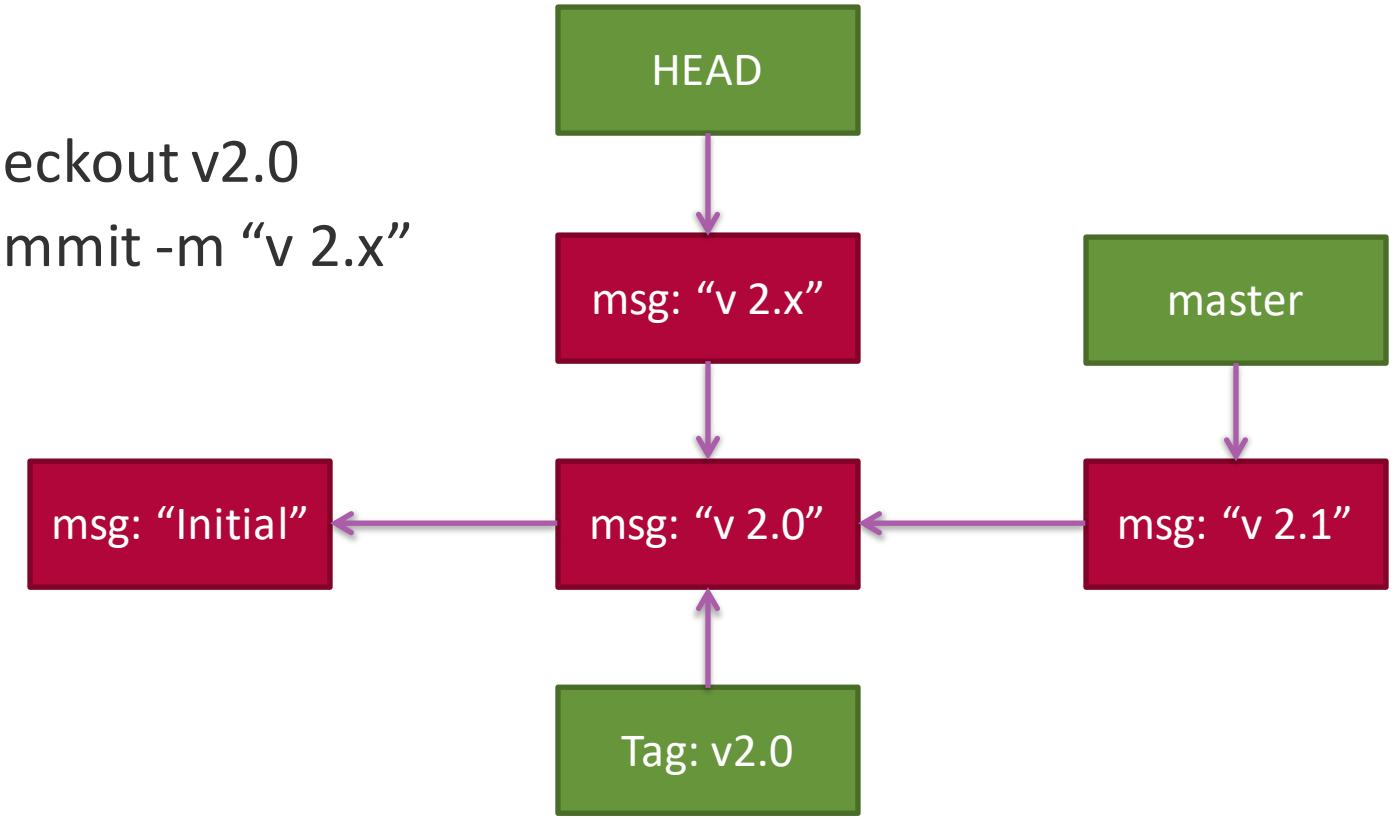
Detached Head

$ git checkout v2.0



Detached Head

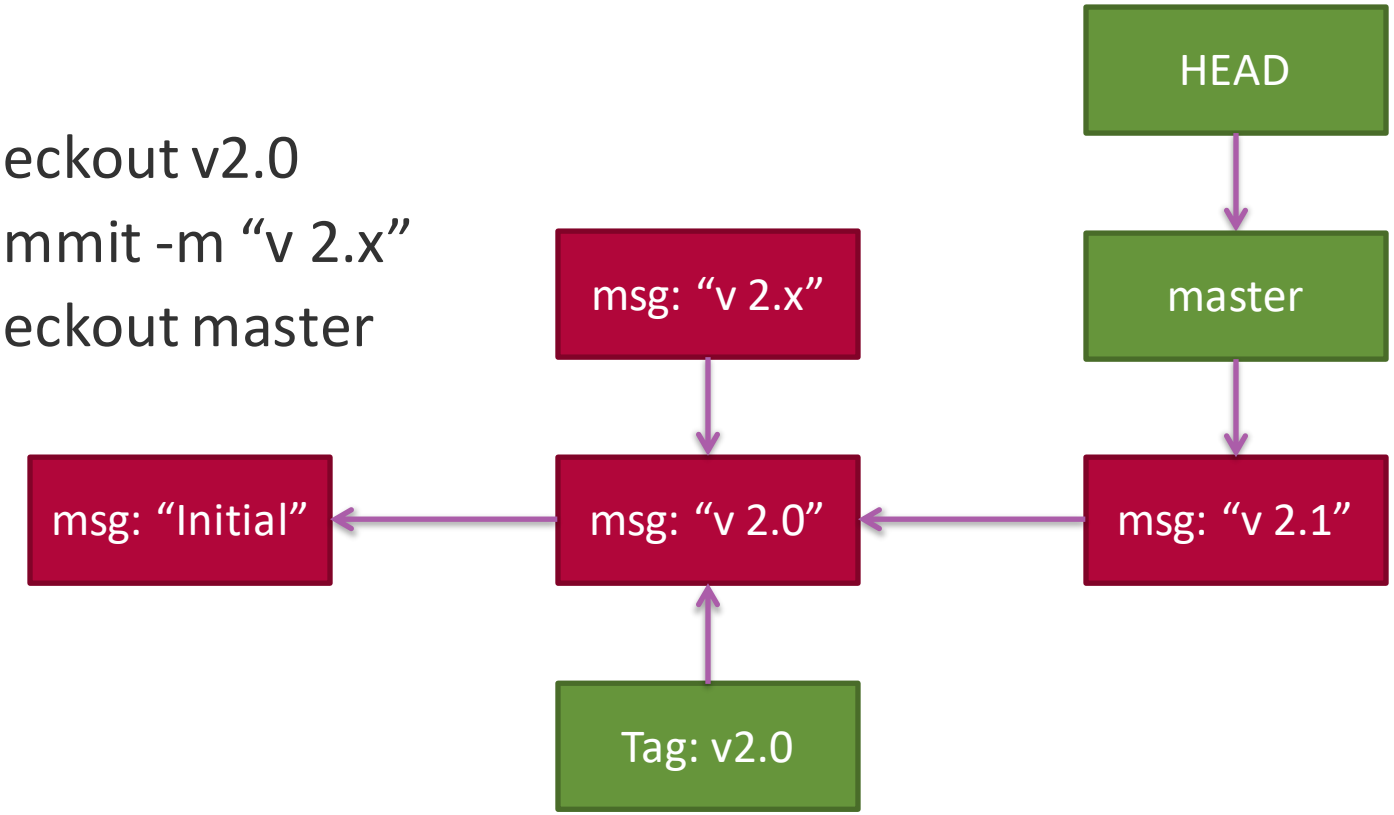
$ git checkout v2.0
$ git commit -m “v 2.x”



Detached Head

HPI

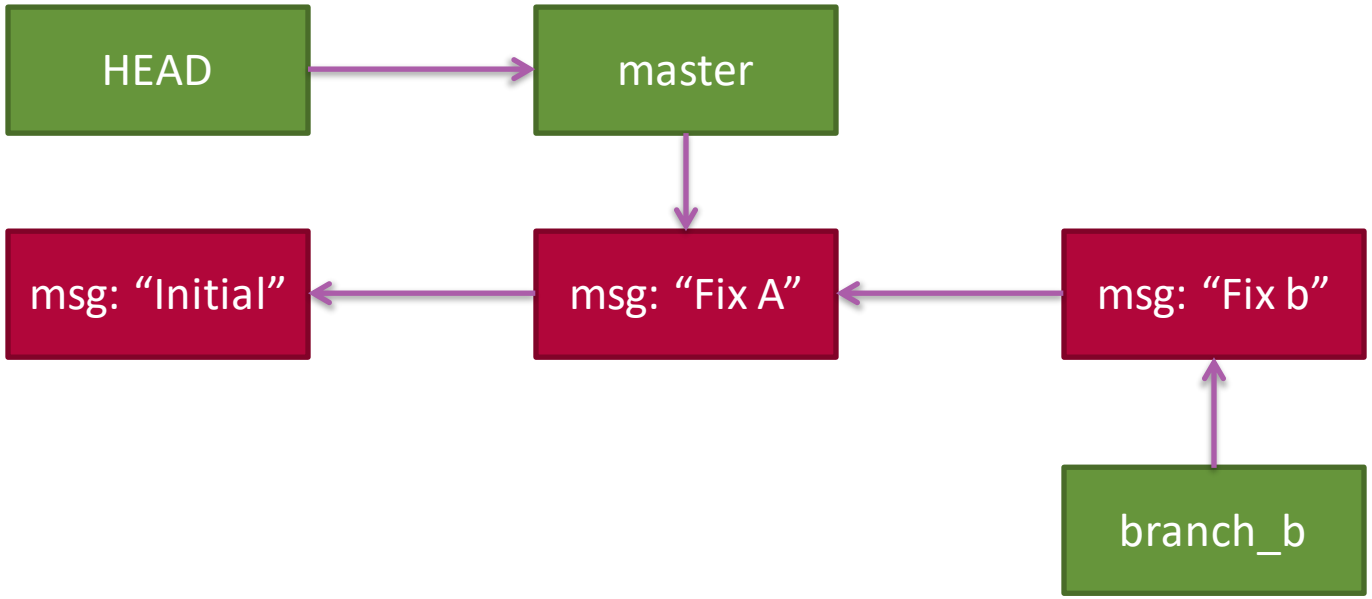
```
$ git checkout v2.0
$ git commit -m “v 2.x”
$ git checkout master
```



List all commits:
`git reflog`

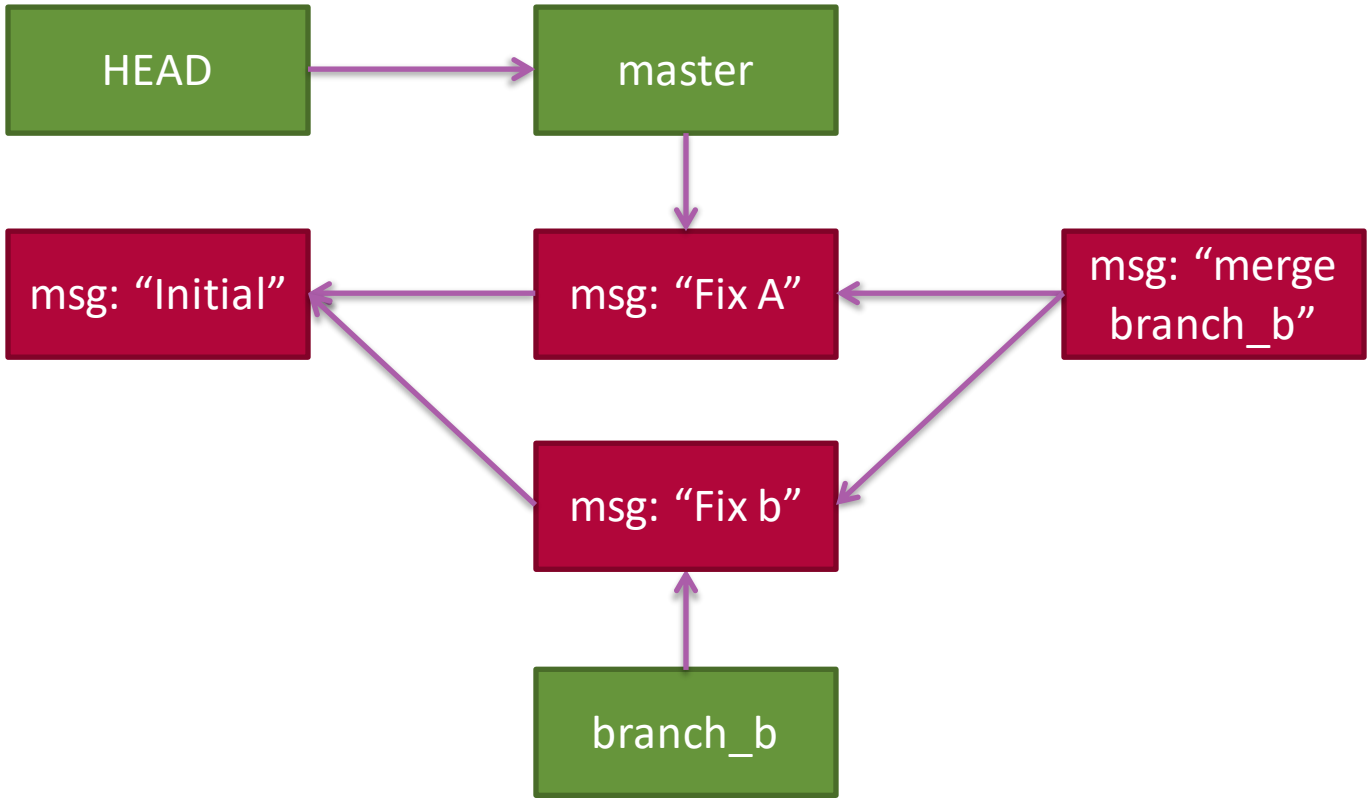
Fast-forward

$ git merge branch_b

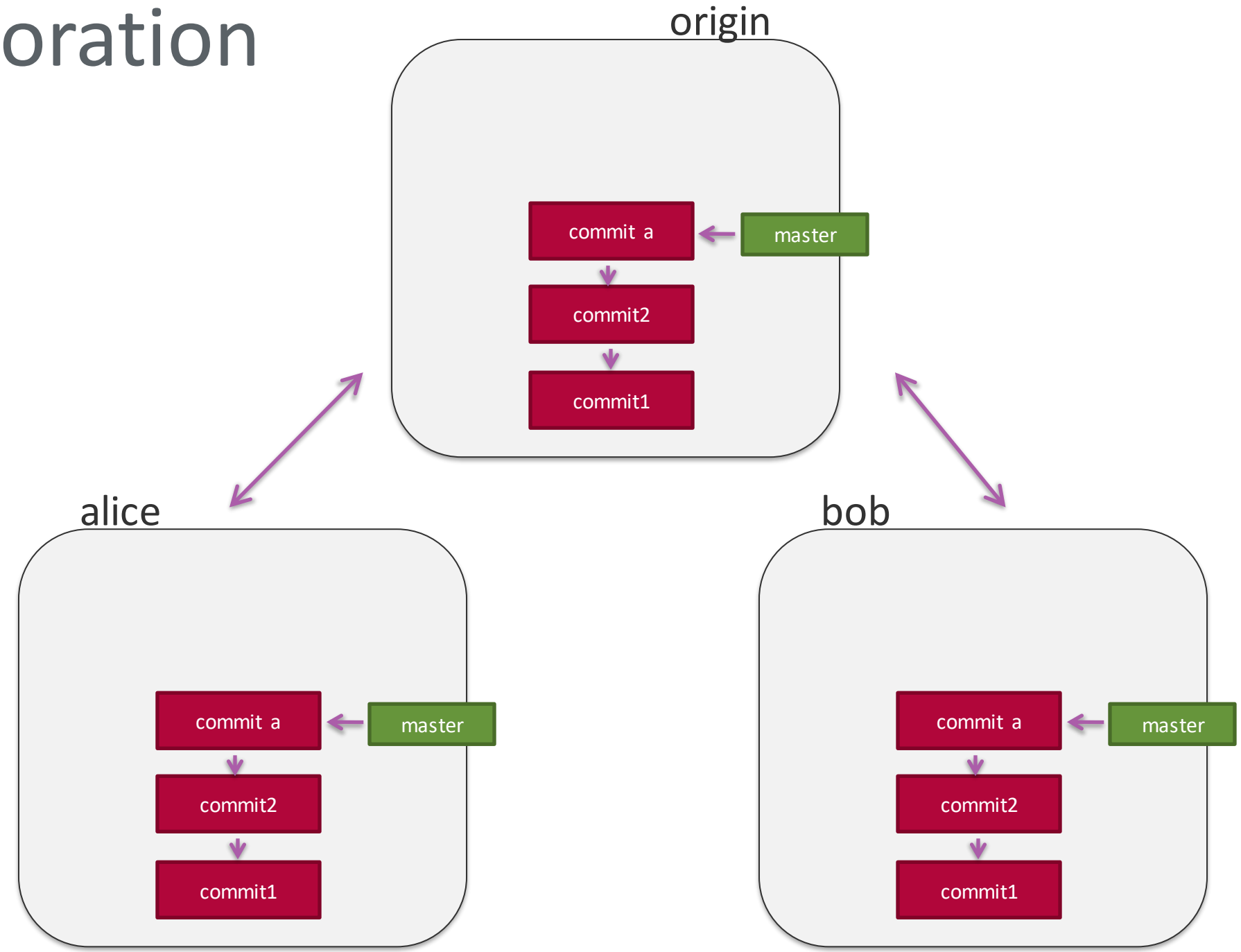


Merge

$ git merge branch_b

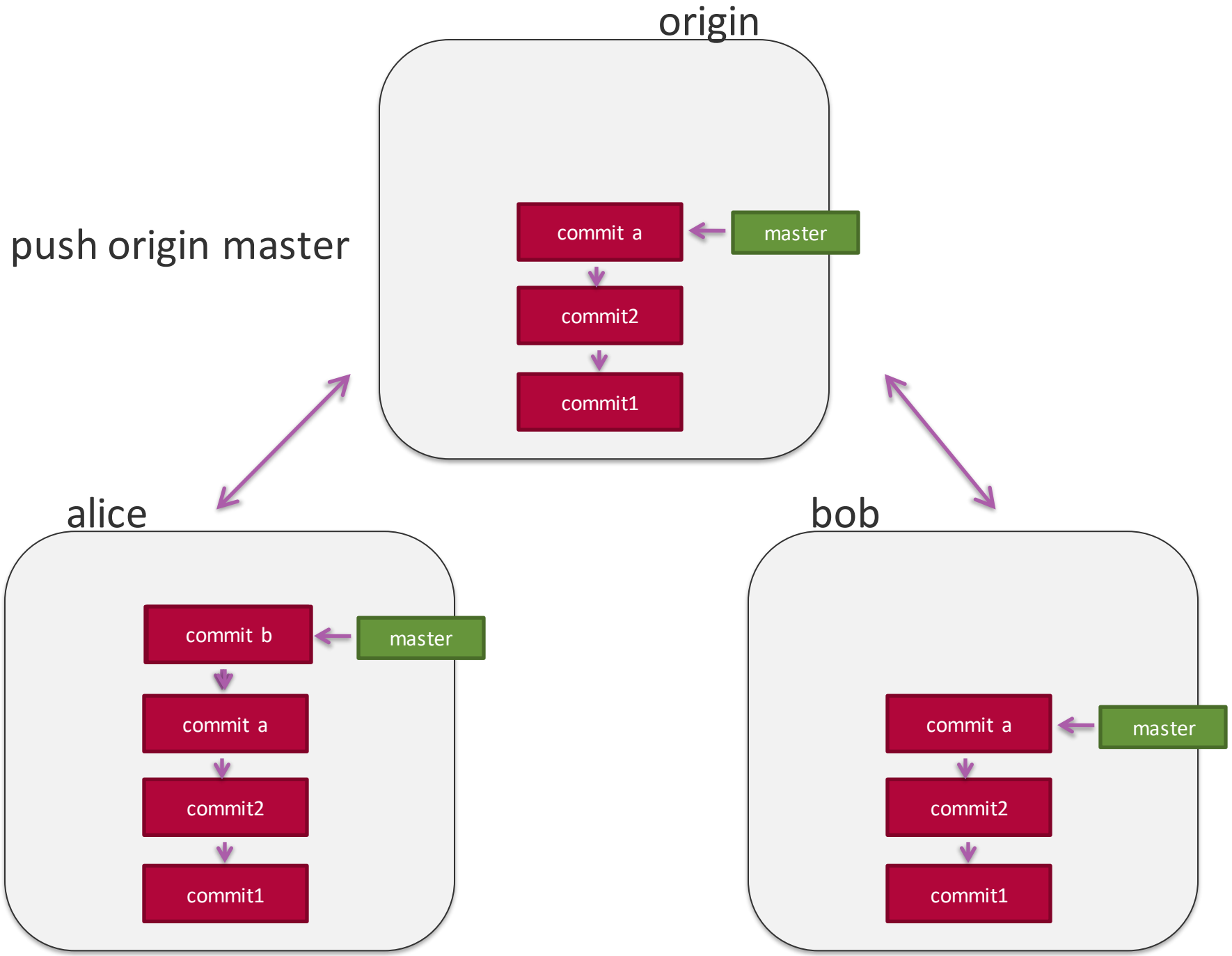


Collaboration



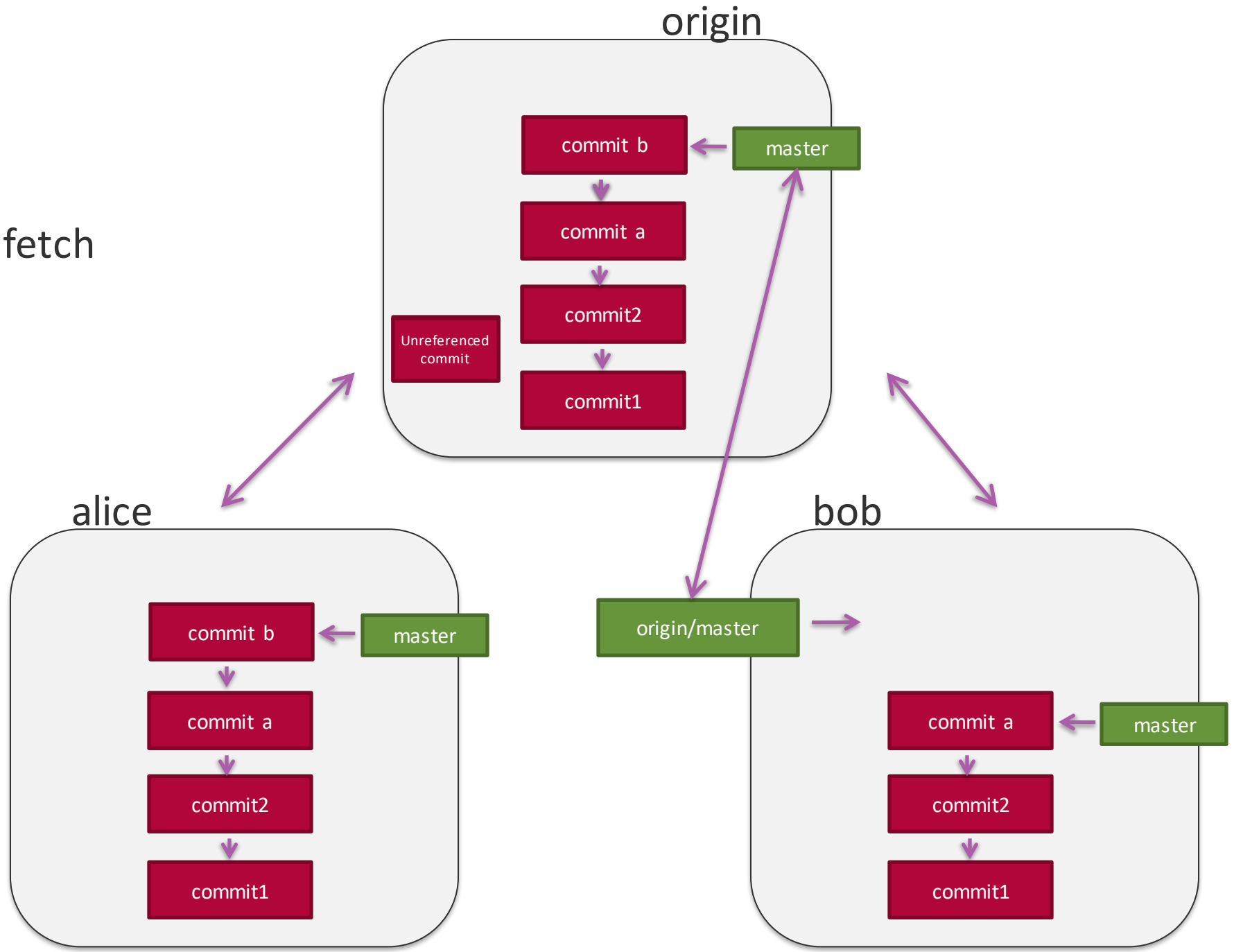
Push

alice$ git push origin master



Fetch

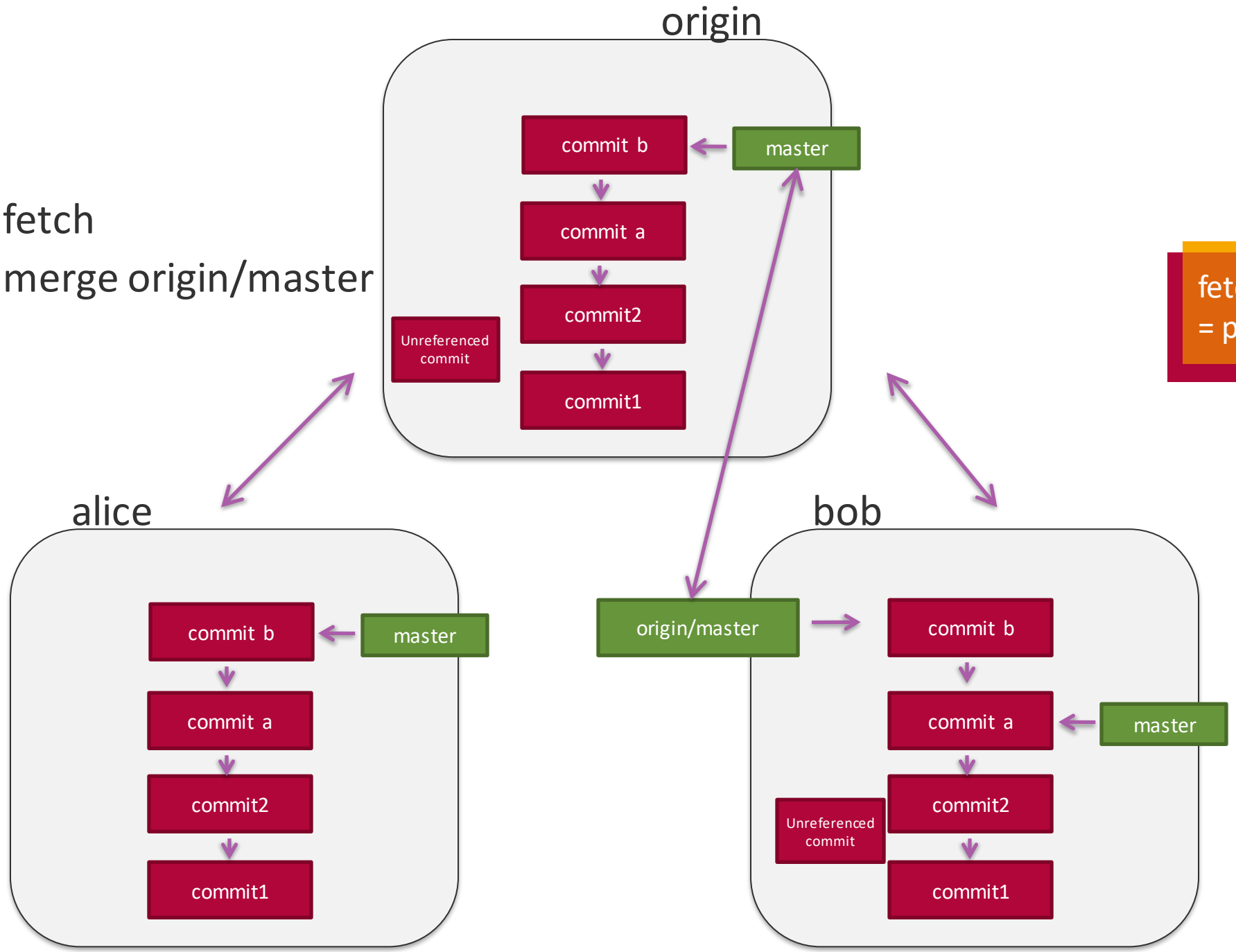
bob$ git fetch



Pull

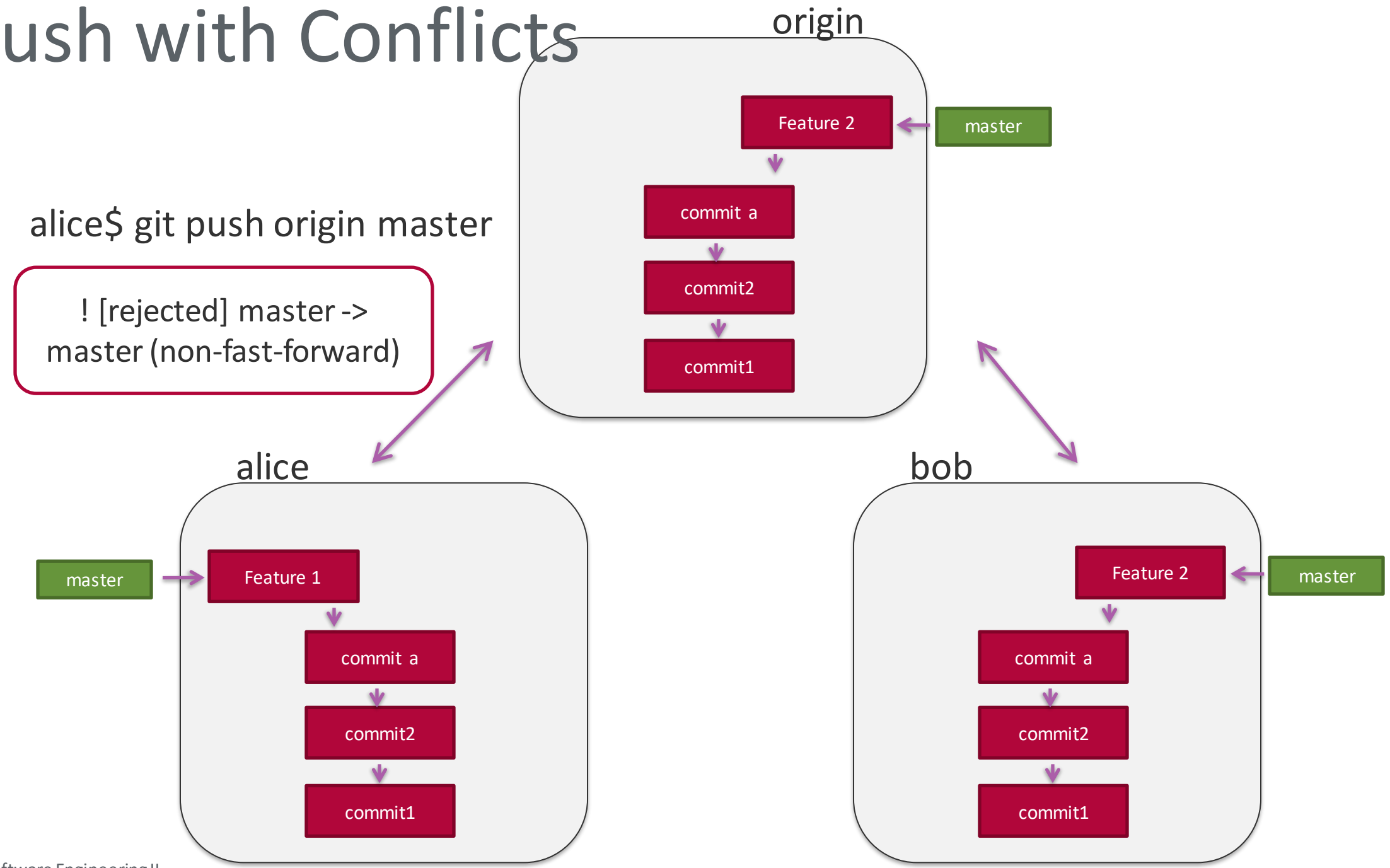
bob$ git fetch

bob$ git merge origin/master

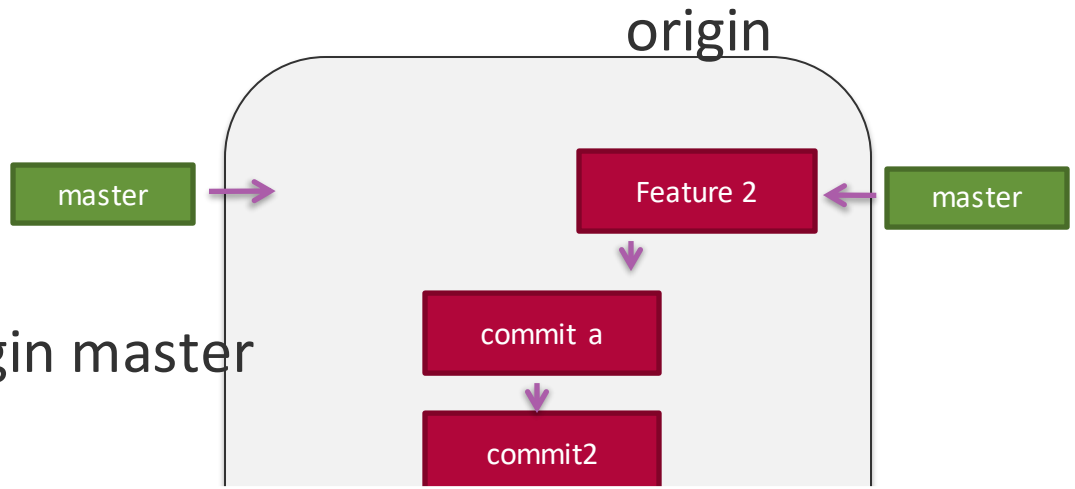


fetch + merge
= pull

Push with Conflicts

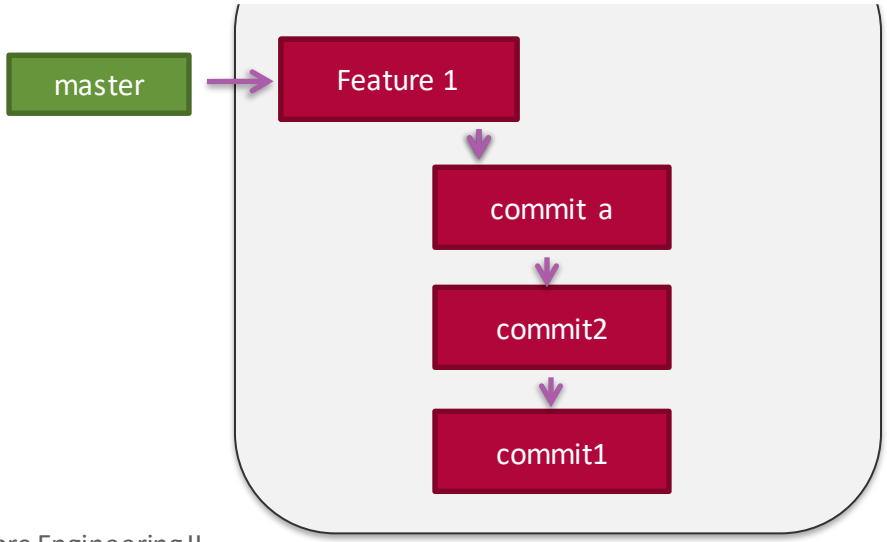


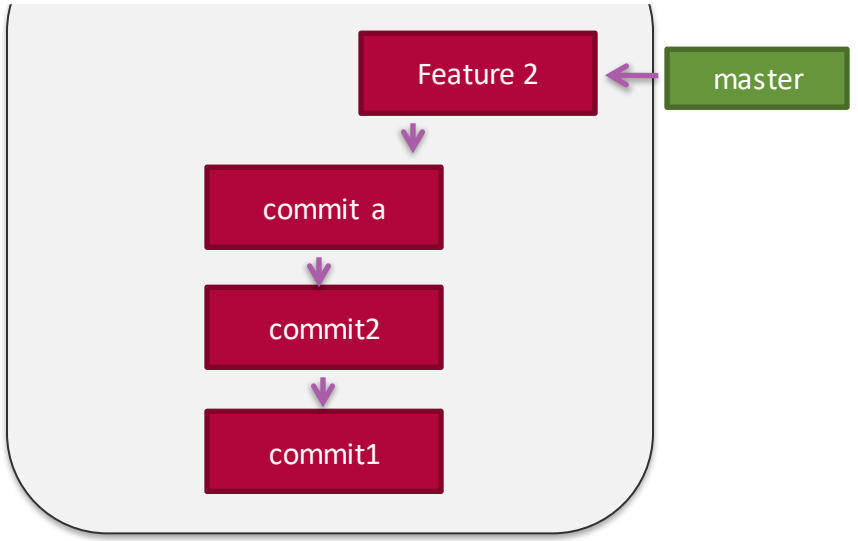
Push --force



alice$ git push --f origin master

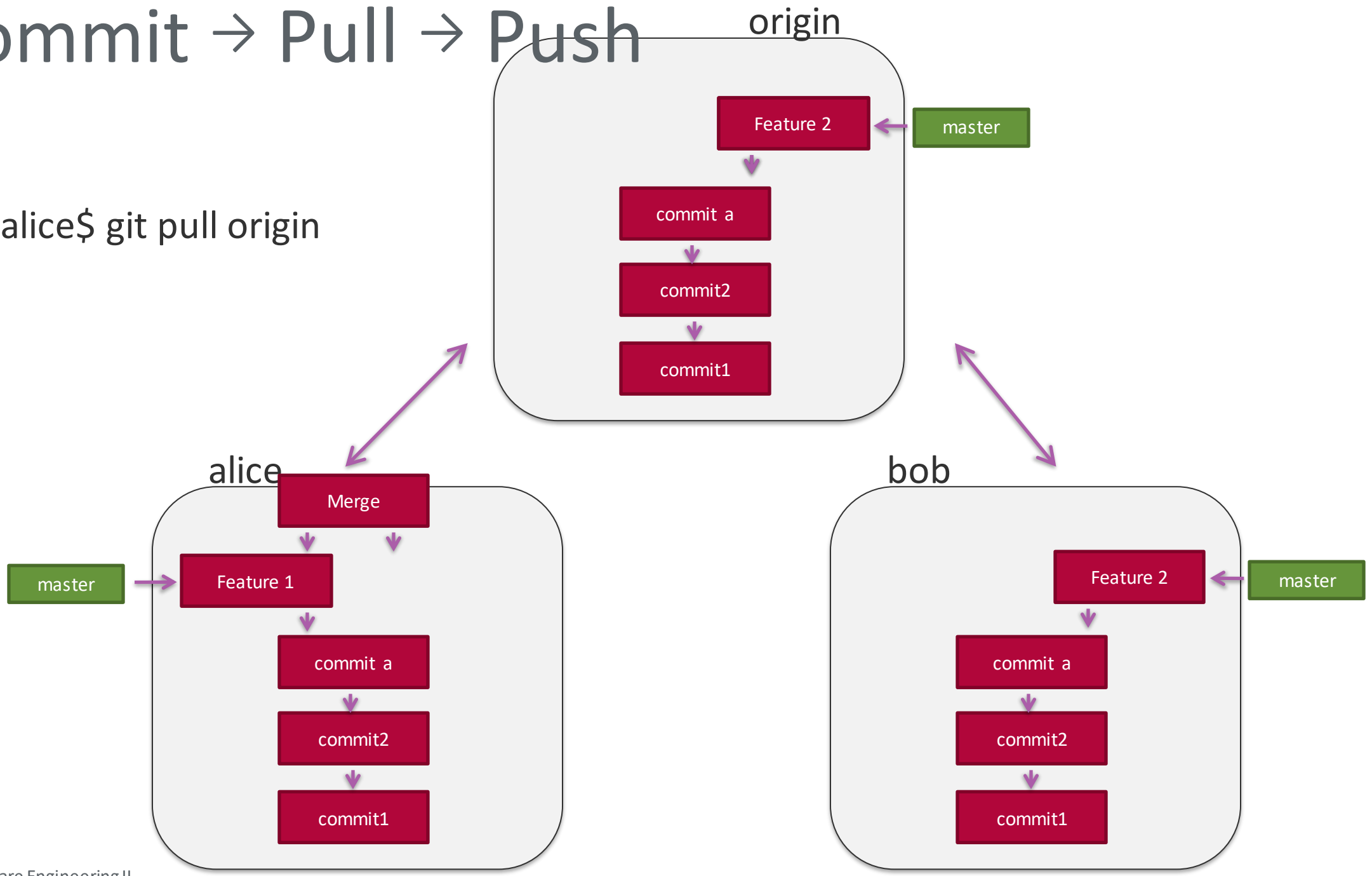
Never **EVER** push --force





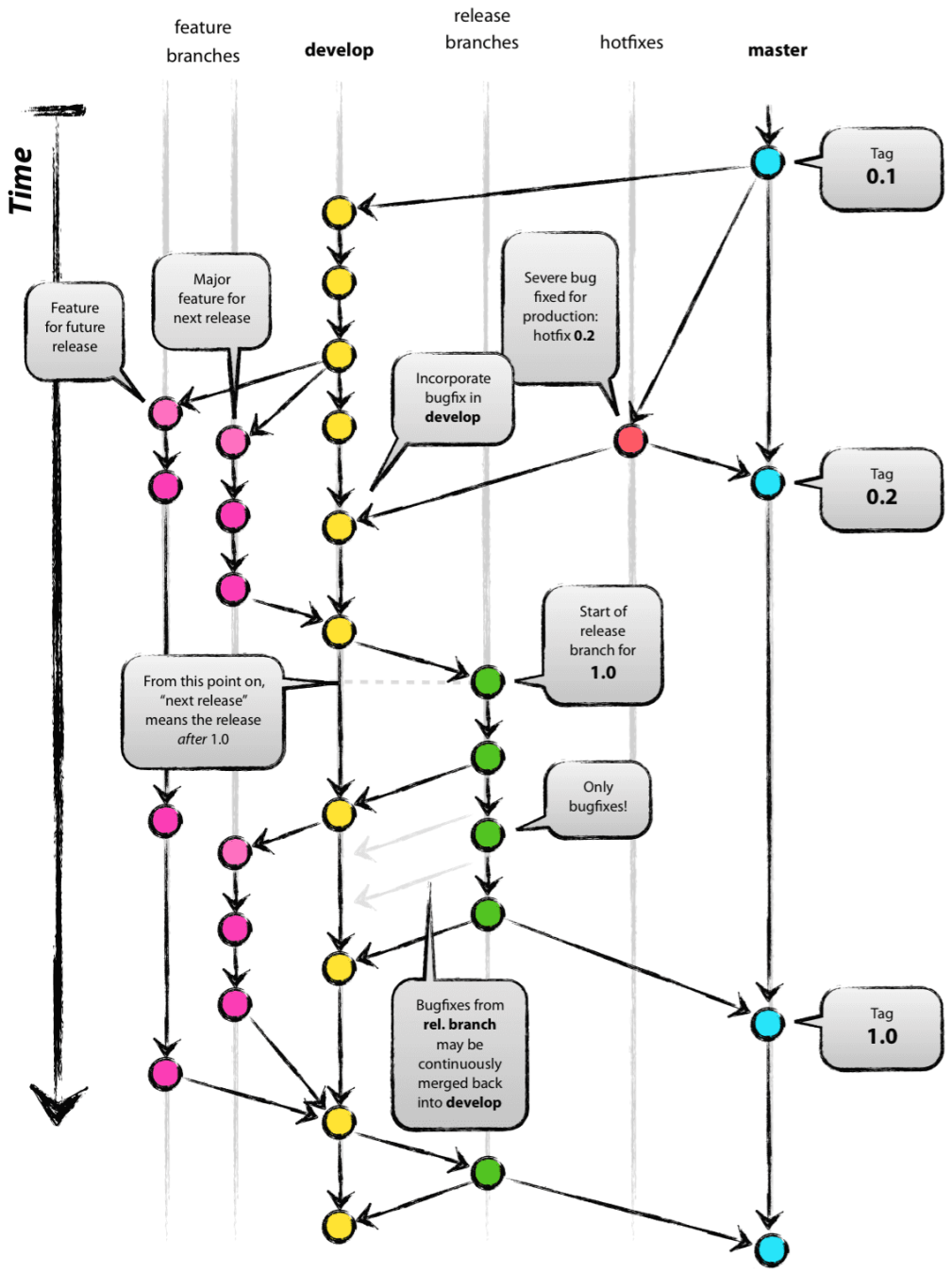
Commit → Pull → Push

alice$ git pull origin



Branching

- Many ways to structure branches
- Some helpful tips:
  - Never merge in master or release branches
  - Never break build in shared branches



http://nvie.com/posts/a-successful-git-branching-model

What happened?

- git log
- git diff
- git blame

Learn some more

Git exercises and training (go find some more), e.g.

- https://gitexercises.fracz.com/
- https://github.com/benthayer/git-gud

Summary

1. Basics
   - Objects
2. Local
   - Checkout
   - Add
   - Commit
3. Collaboration
   - Pull
   - Push