# Git

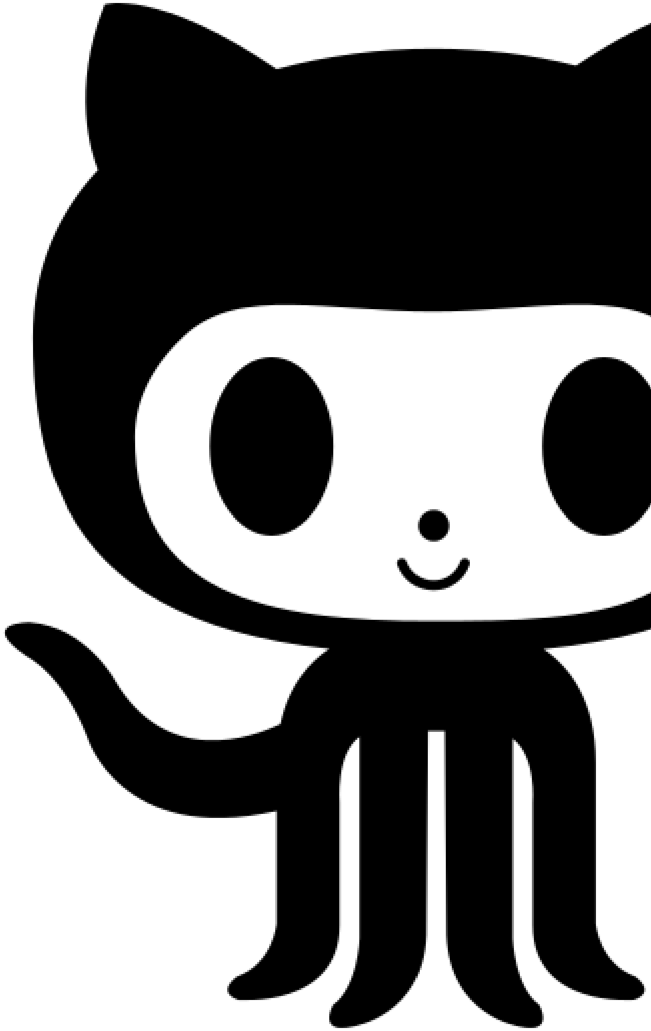
Arian Treffer
arian.treffer@hpi.de

Prof. Plattner, Dr. Uflacker
Enterprise Platform and Integration Concepts group

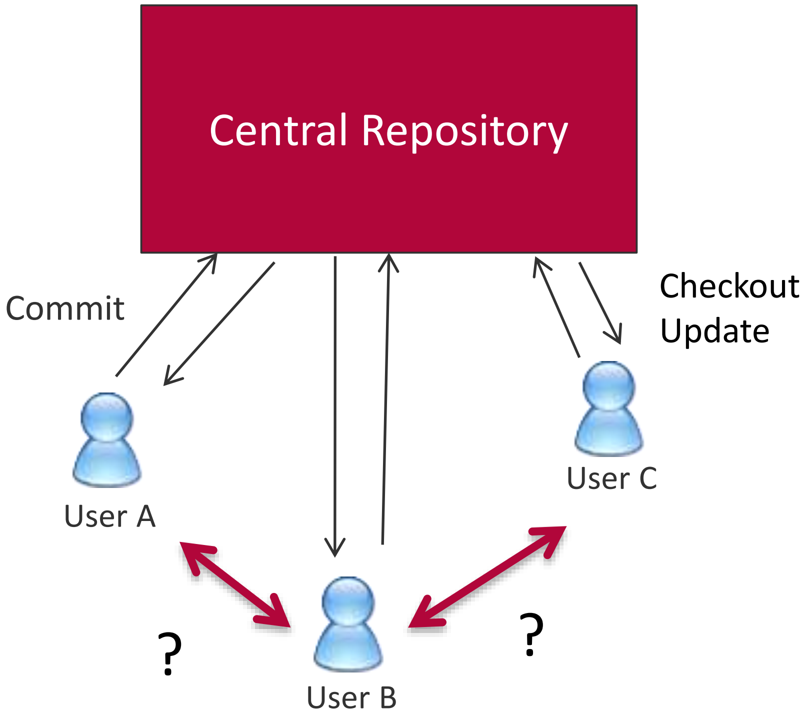
Software Engineering II
WS 2015/16

# Outline

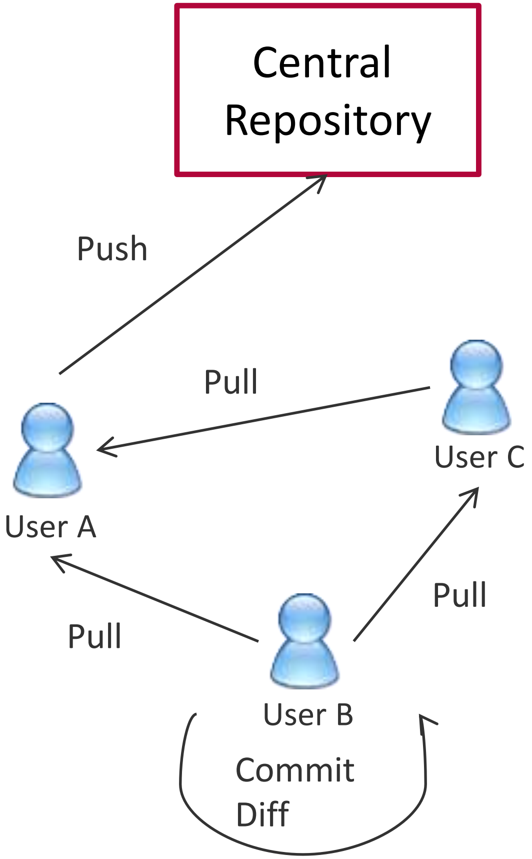
1. Basics
2. Local
3. Collaboration



# 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
  - Why?
- Pull-Requests are better than patch files
- Every clone creates a full copy of the repository
  - Good or bad?

# Git Objects

Blob

- Content of a file
- Nothing else

Tree

- File structure
- References Blobs

Office Checklist 31.03.11 20:45

Collapse sunshade

Take all pillows to the conference table

Close all windows and doors

Clean up your used glasses and plates

Turn off Nespresso machine

Turn off all lights

Close the door and lock both!!! locks

COMMIT!

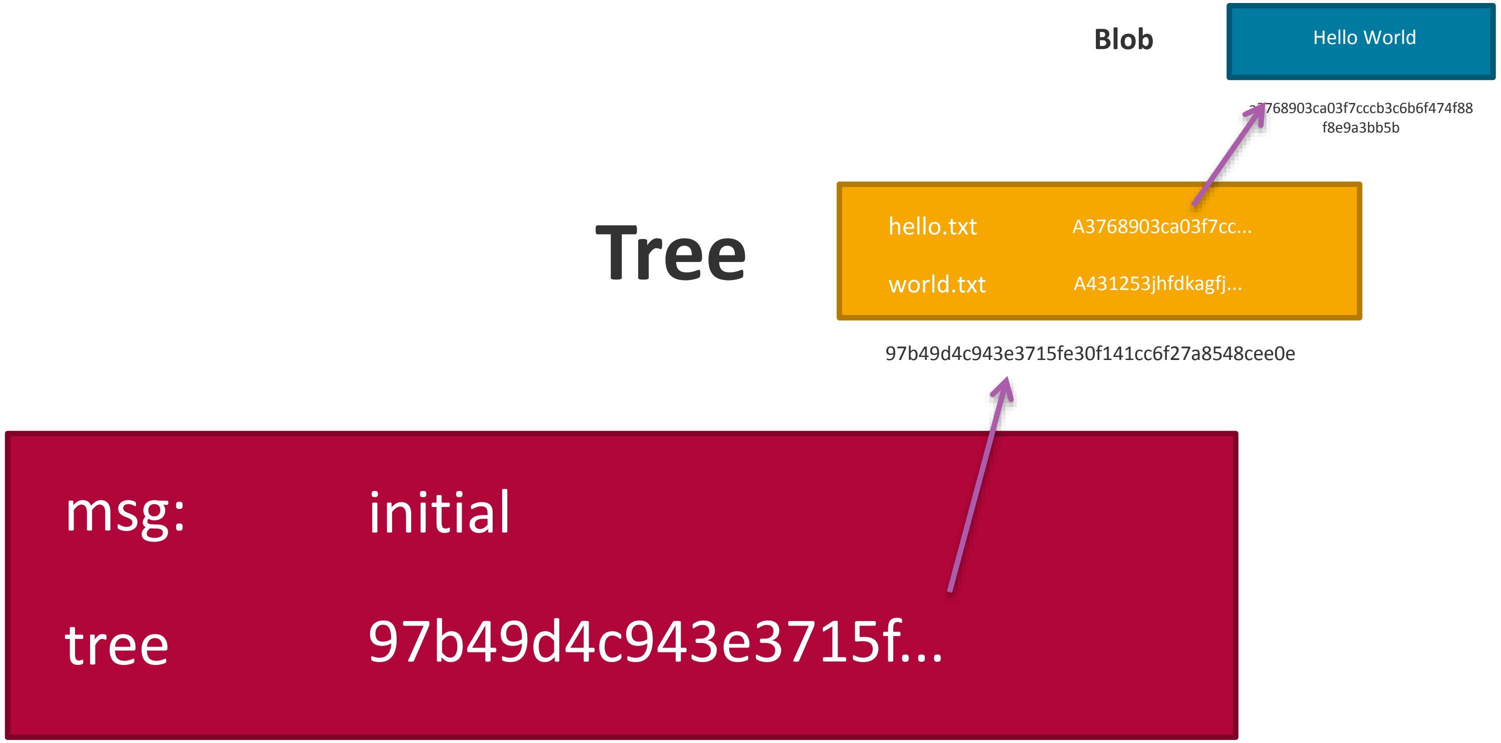
Commit

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

Tag

- Reference to other object

# Commit



Blob

Hello World

a3768903ca03f7cccb3c6b6f474f88
f8e9a3bb5b

Tree

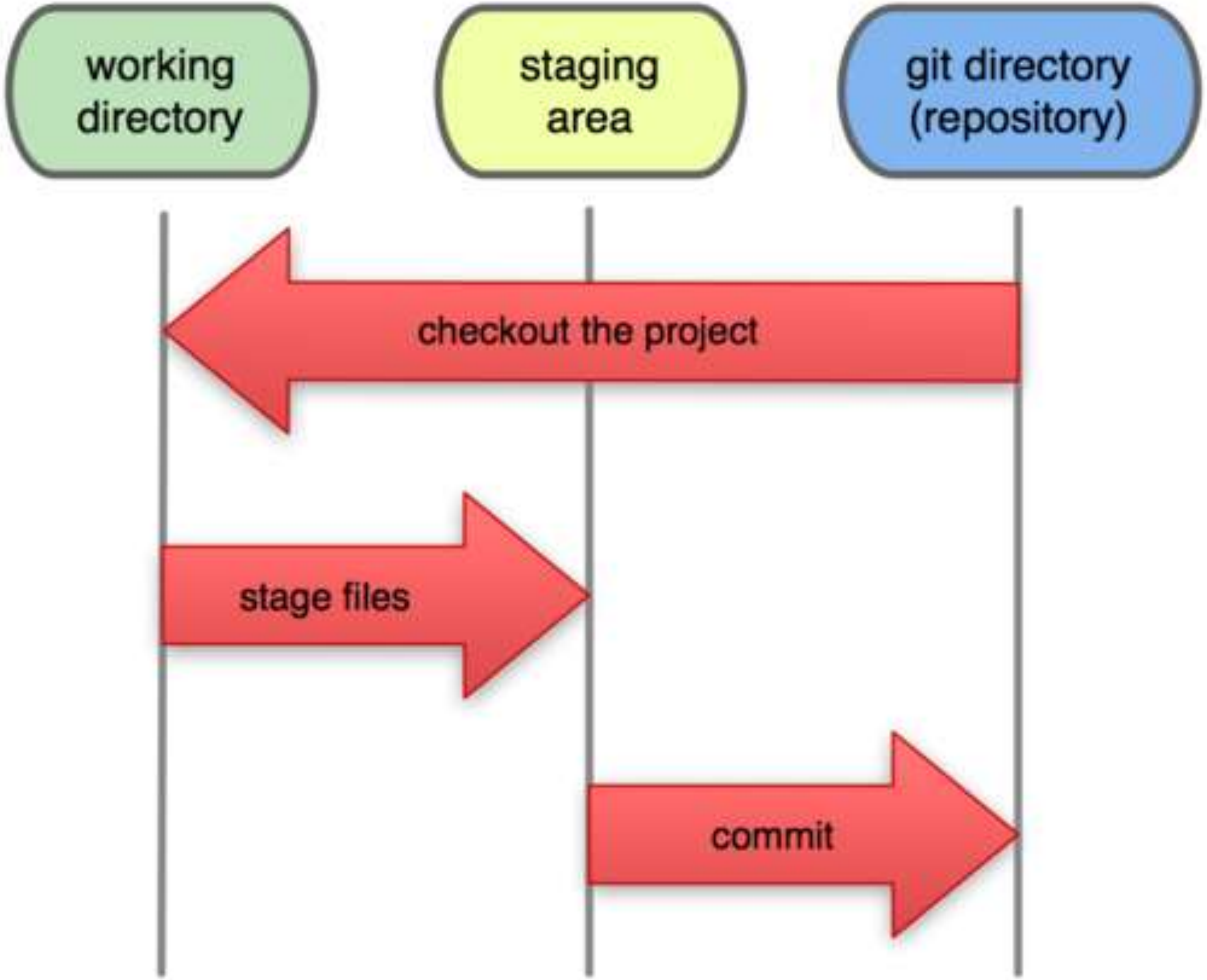
| | |
|---|---|
| hello.txt | A3768903ca03f7cc... |
| world.txt | A431253jhfdkagfj... |

97b49d4c943e3715fe30f141cc6f27a8548cee0e

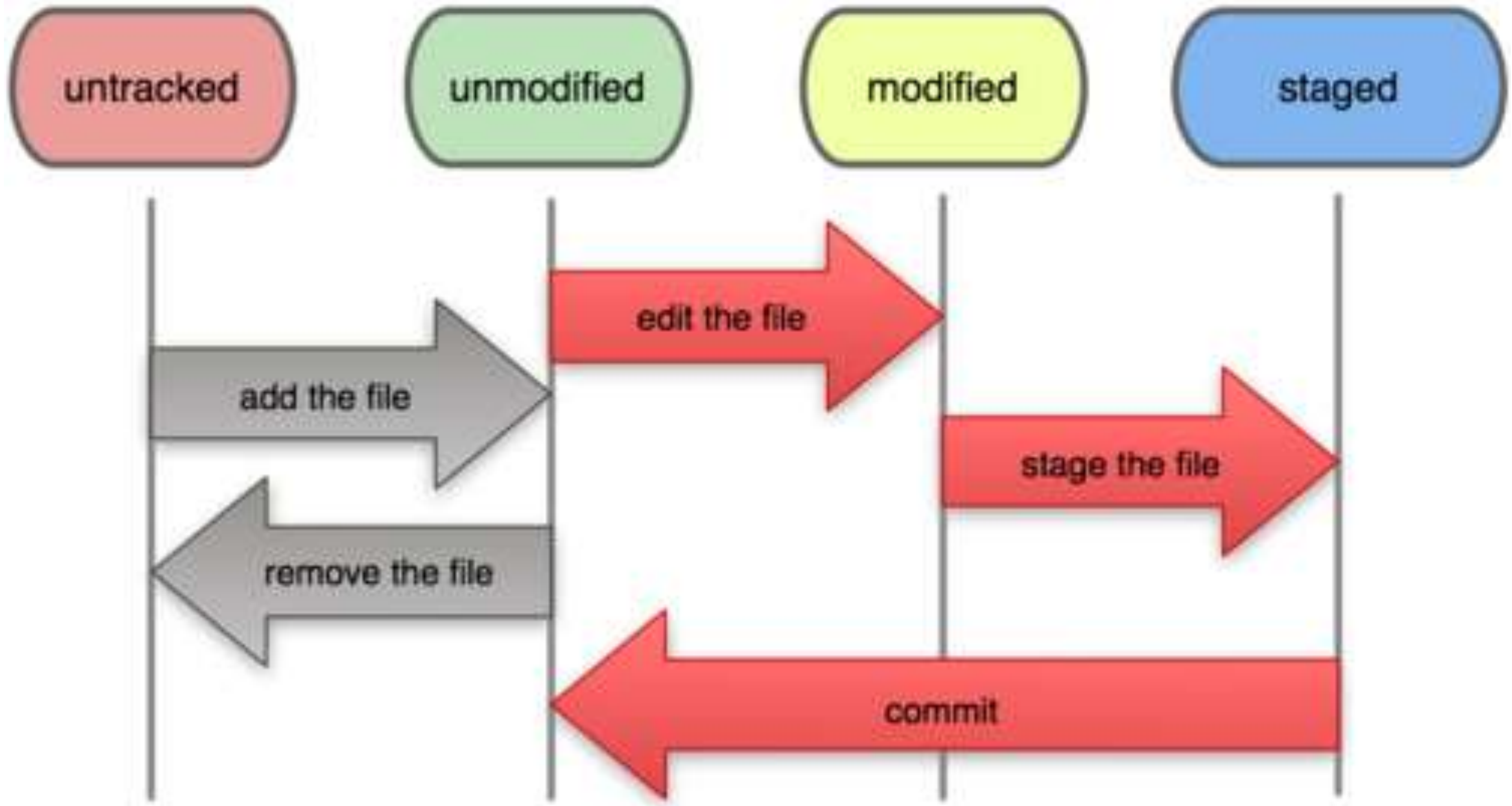
| | |
|---|---|
| msg: | initial |
| tree | 97b49d4c943e3715f... |

a3768903ca03f7cccb3c6b6f474f88f8e9a3bb5b

# Local Operations

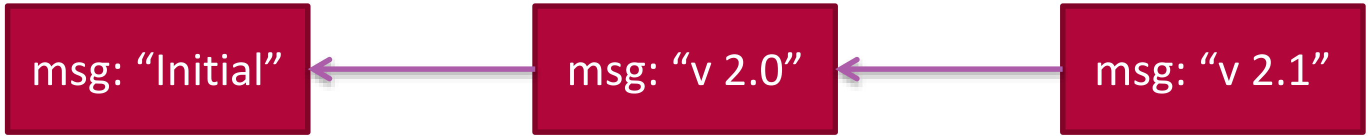


# File Status Lifecycle



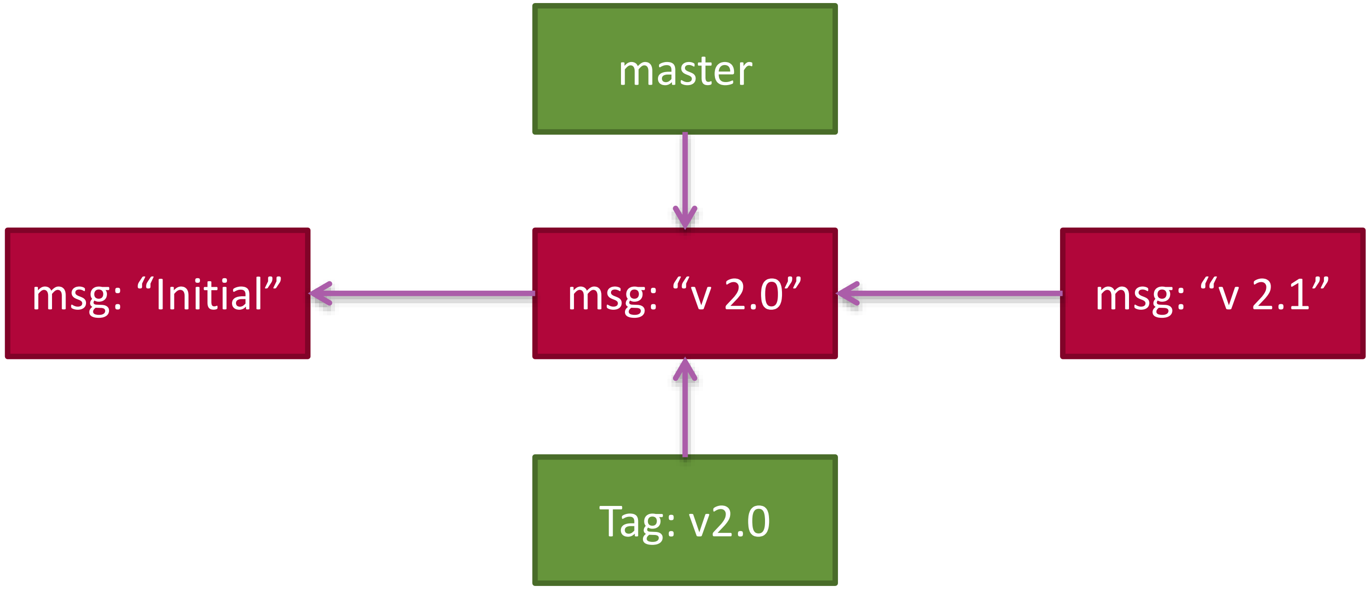
# Commit Parent

```
$ git commit -m “v 2.1”
```



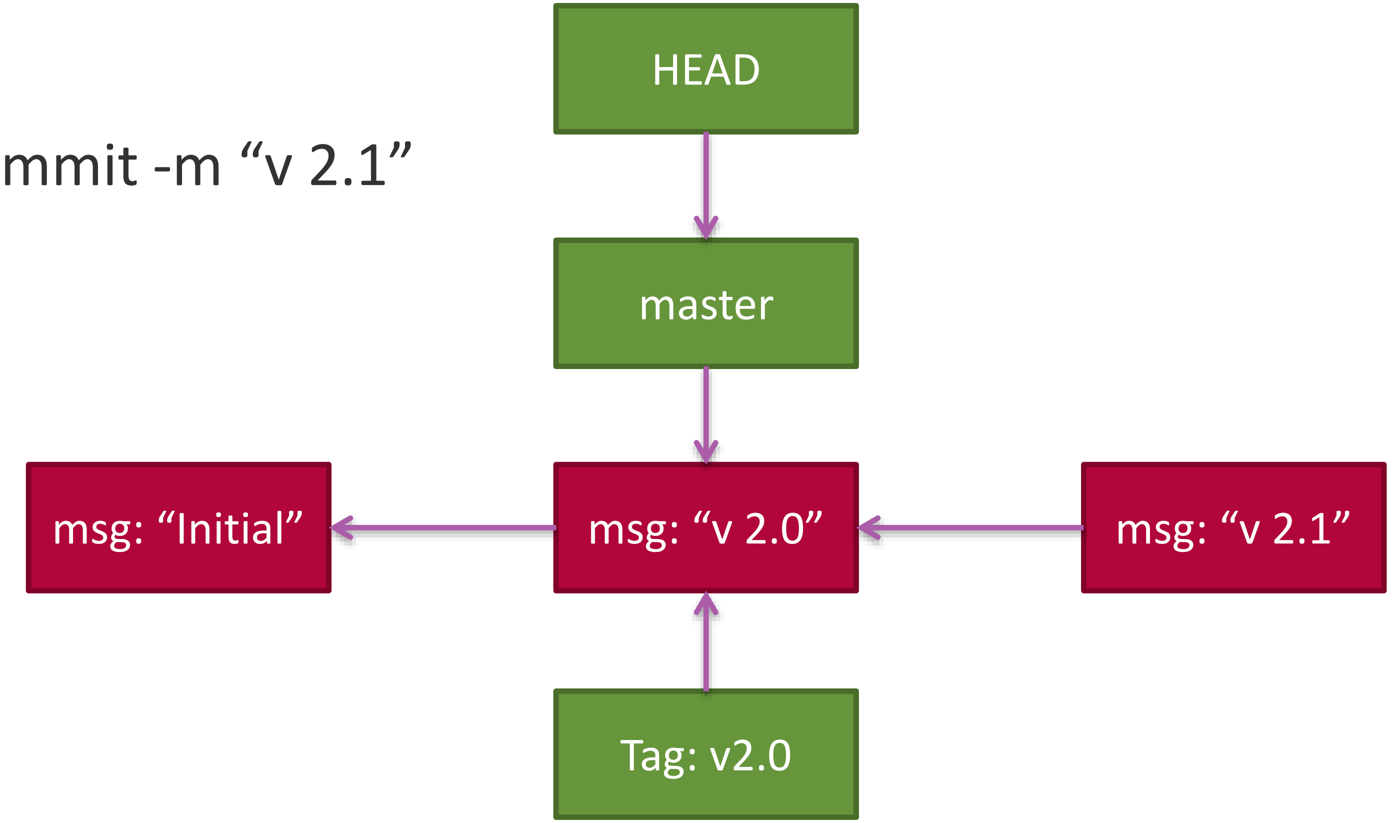
# Branches & Tags

```
$ git commit -m "v 2.1"
```



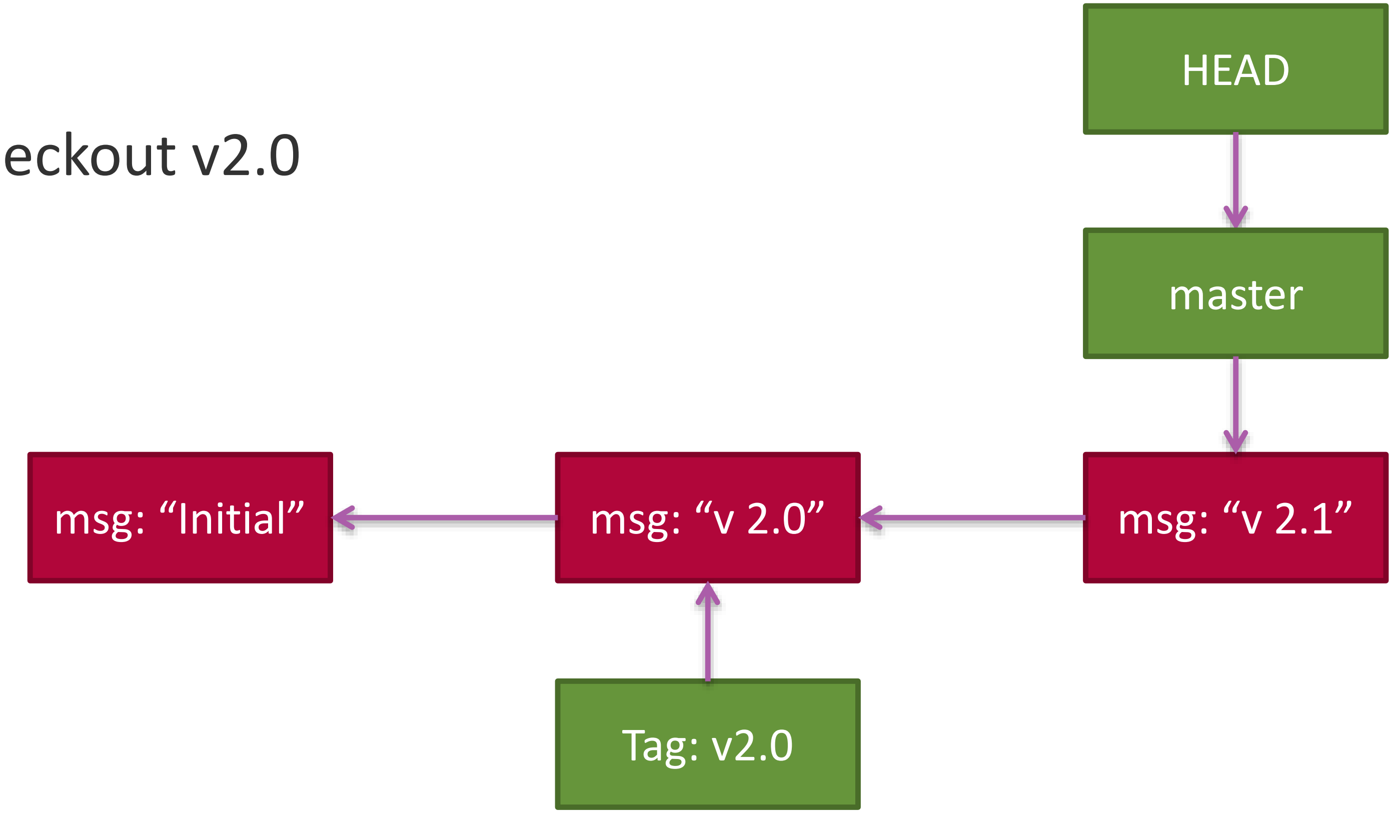
# Head

```
$ git commit -m “v 2.1”
```



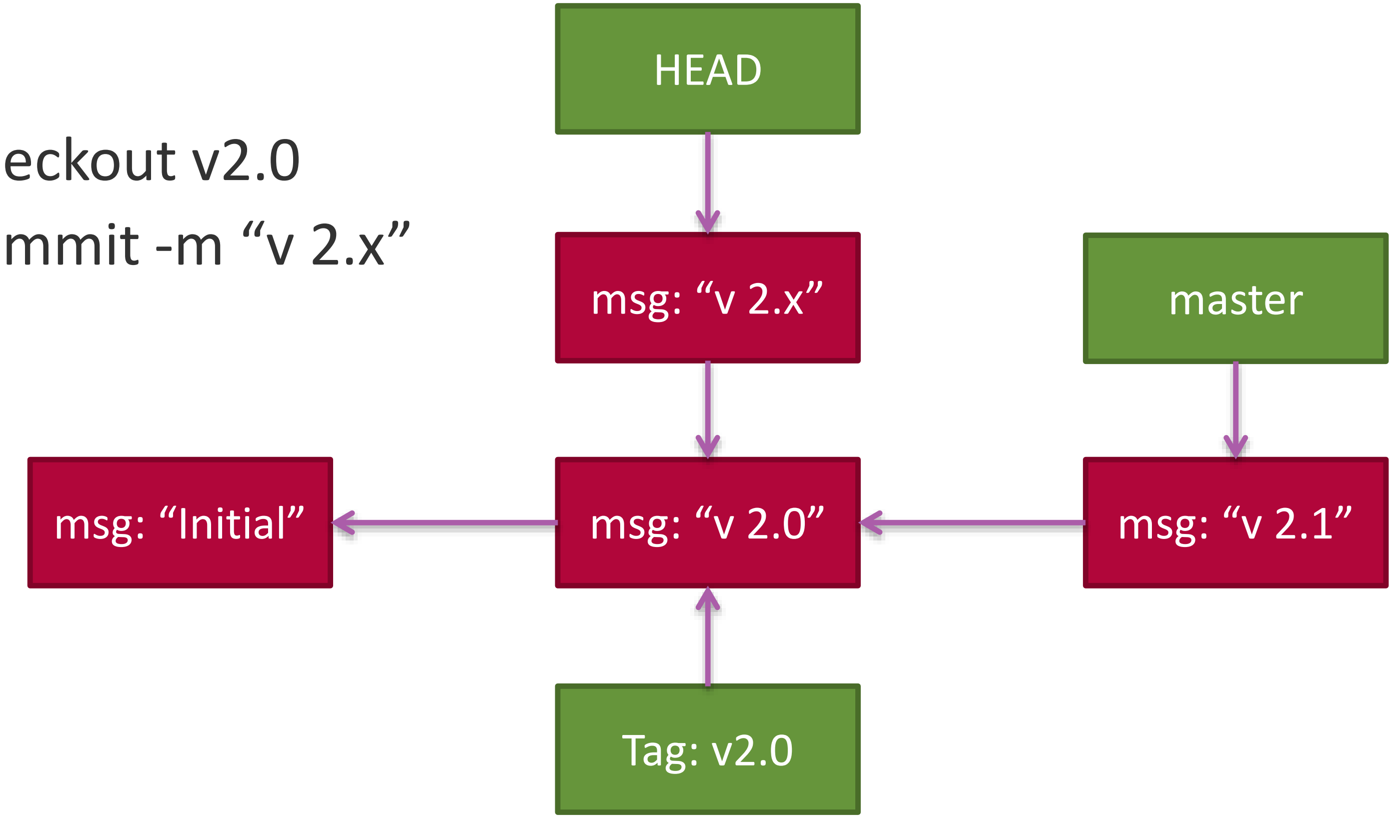
# Detached Head

```
$ git checkout v2.0
```



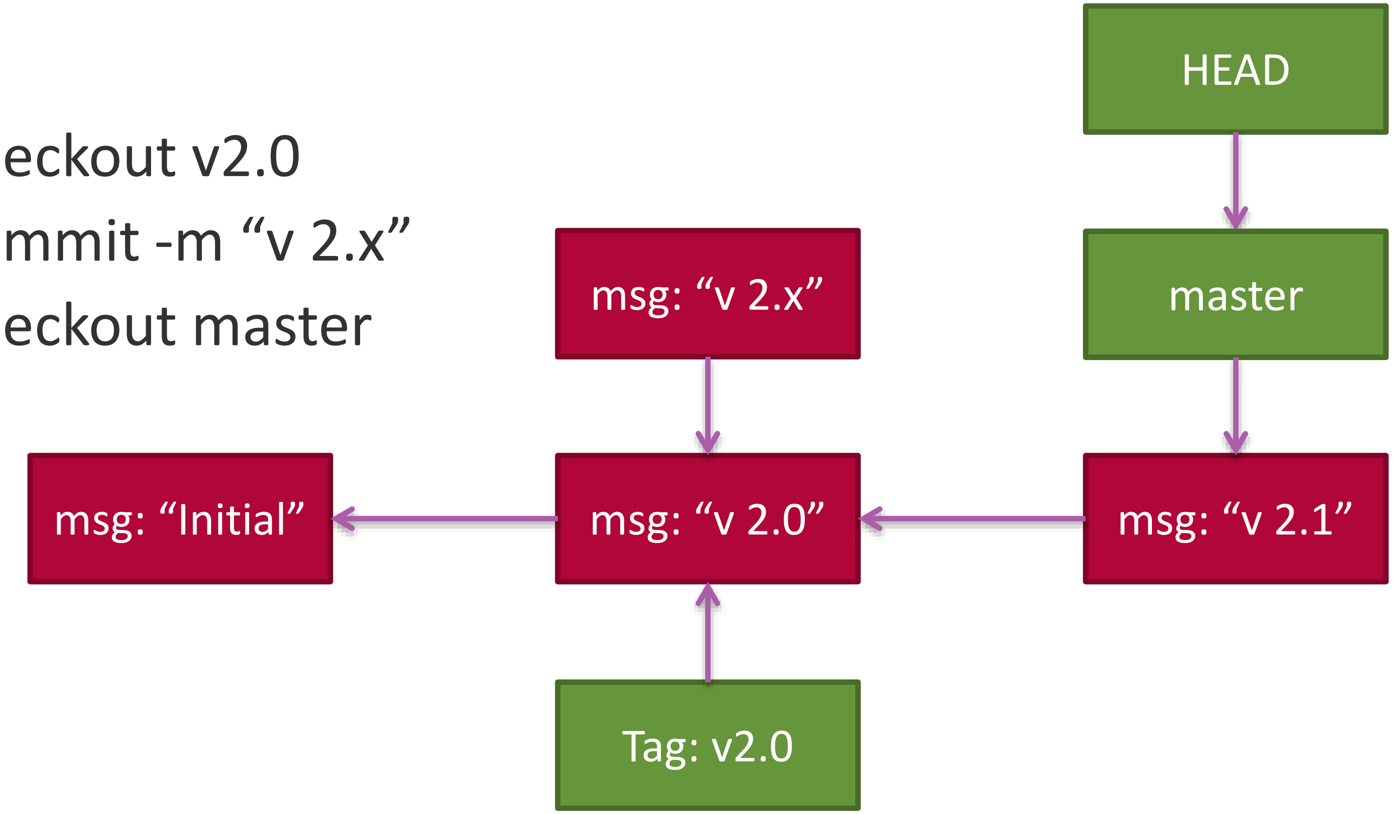
# Detached Head

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



# Detached Head

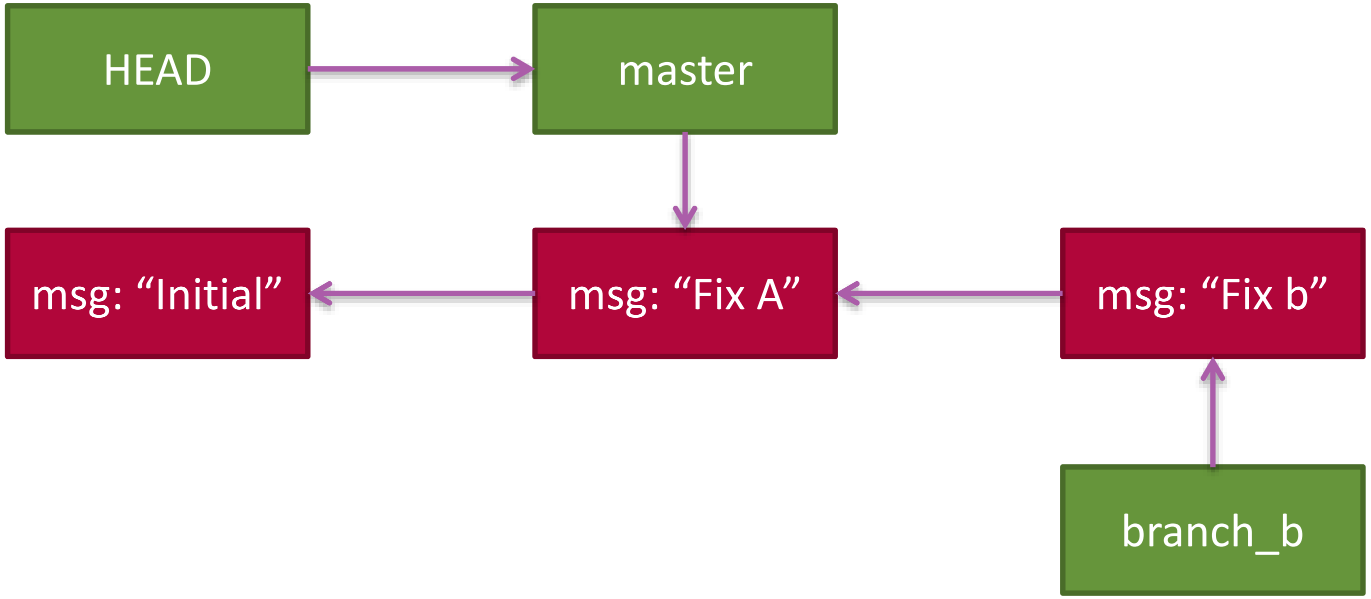
```
$ git checkout v2.0
$ git commit -m "v 2.x"
$ git checkout master
```



**Tip:**
List all commits:
`git reflog`

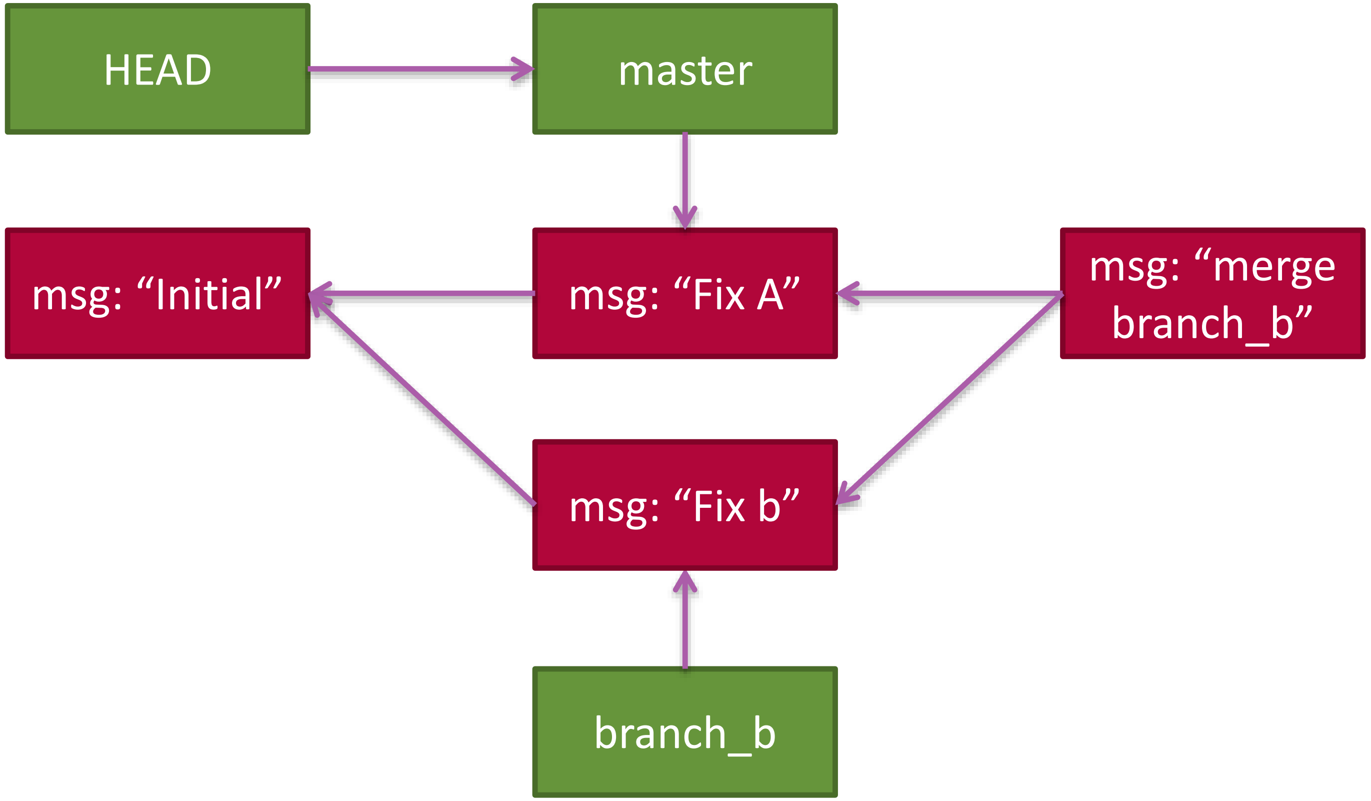
# Fast-forward

```
$ git merge branch_b
```

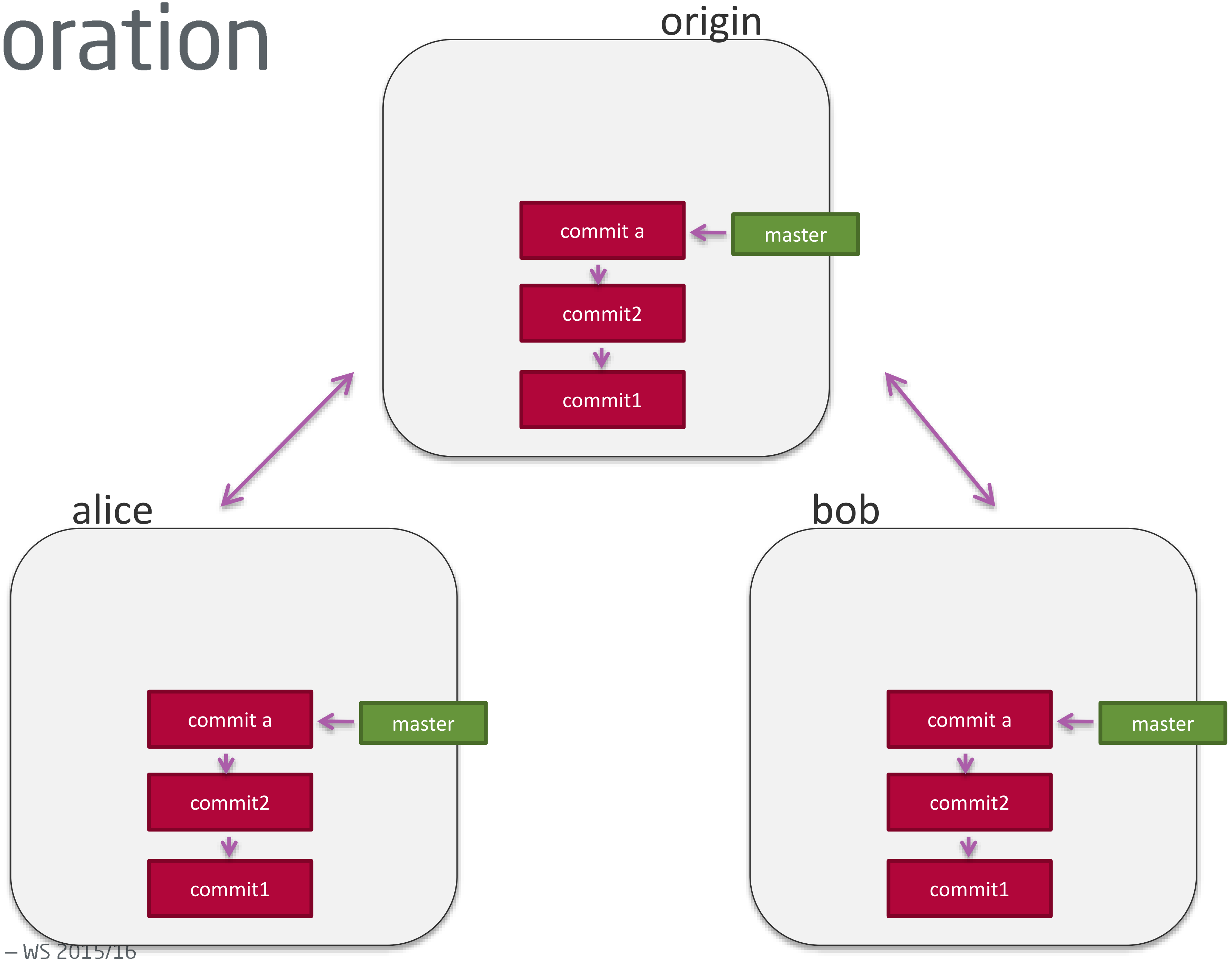


# Merge

$ git merge branch_b

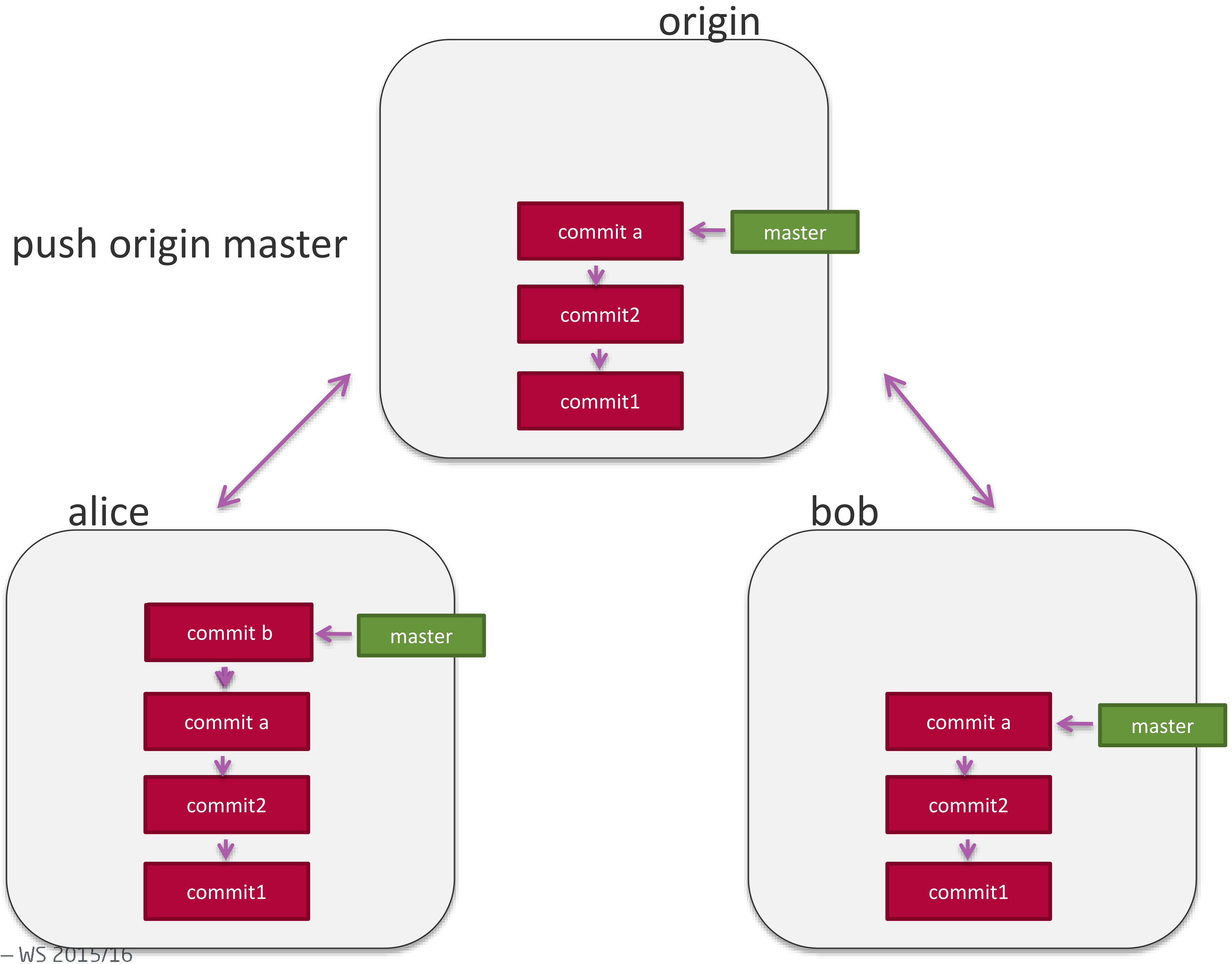


# Collaboration



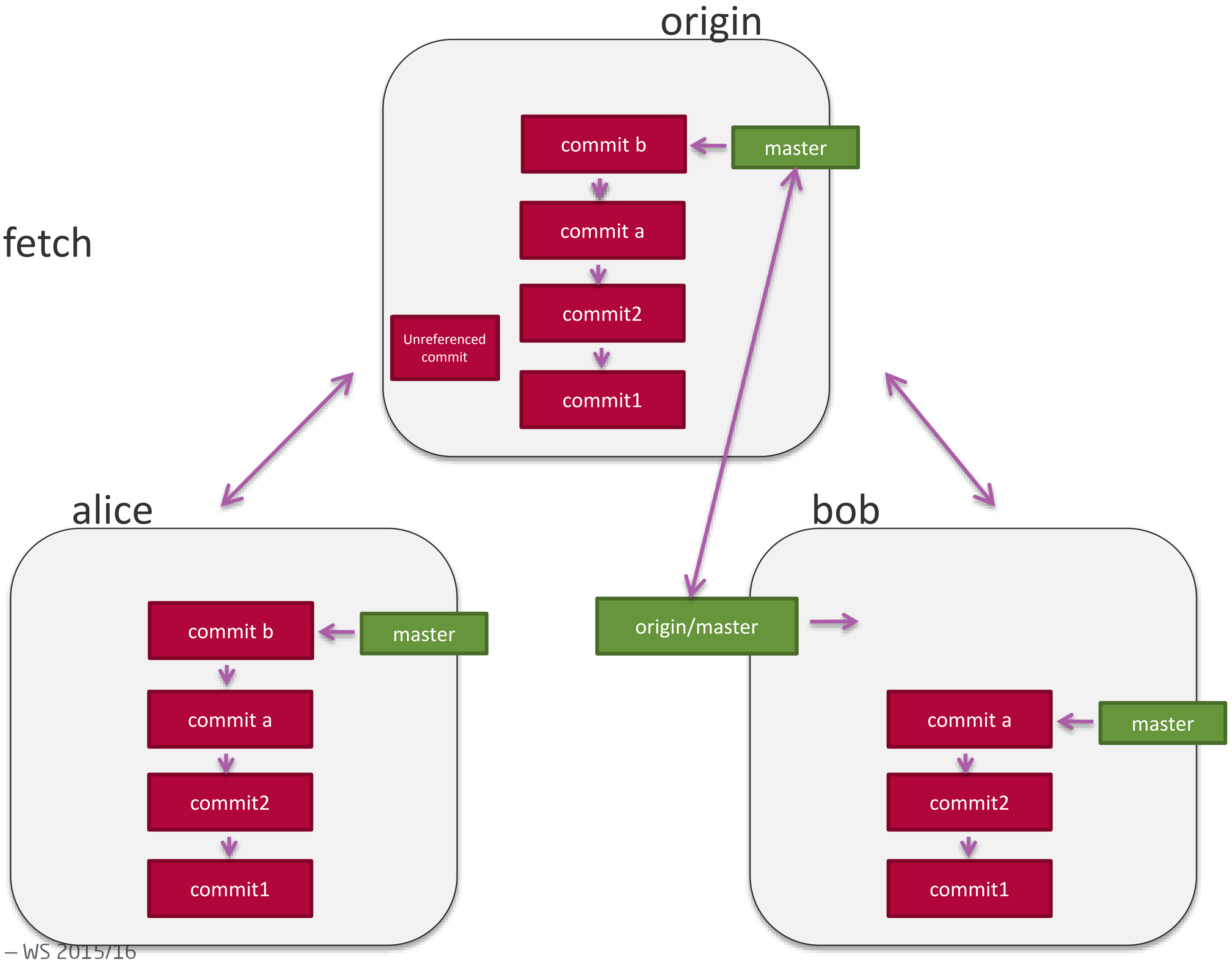
# Push

alice$ git push origin master



# Fetch

bob$ git fetch

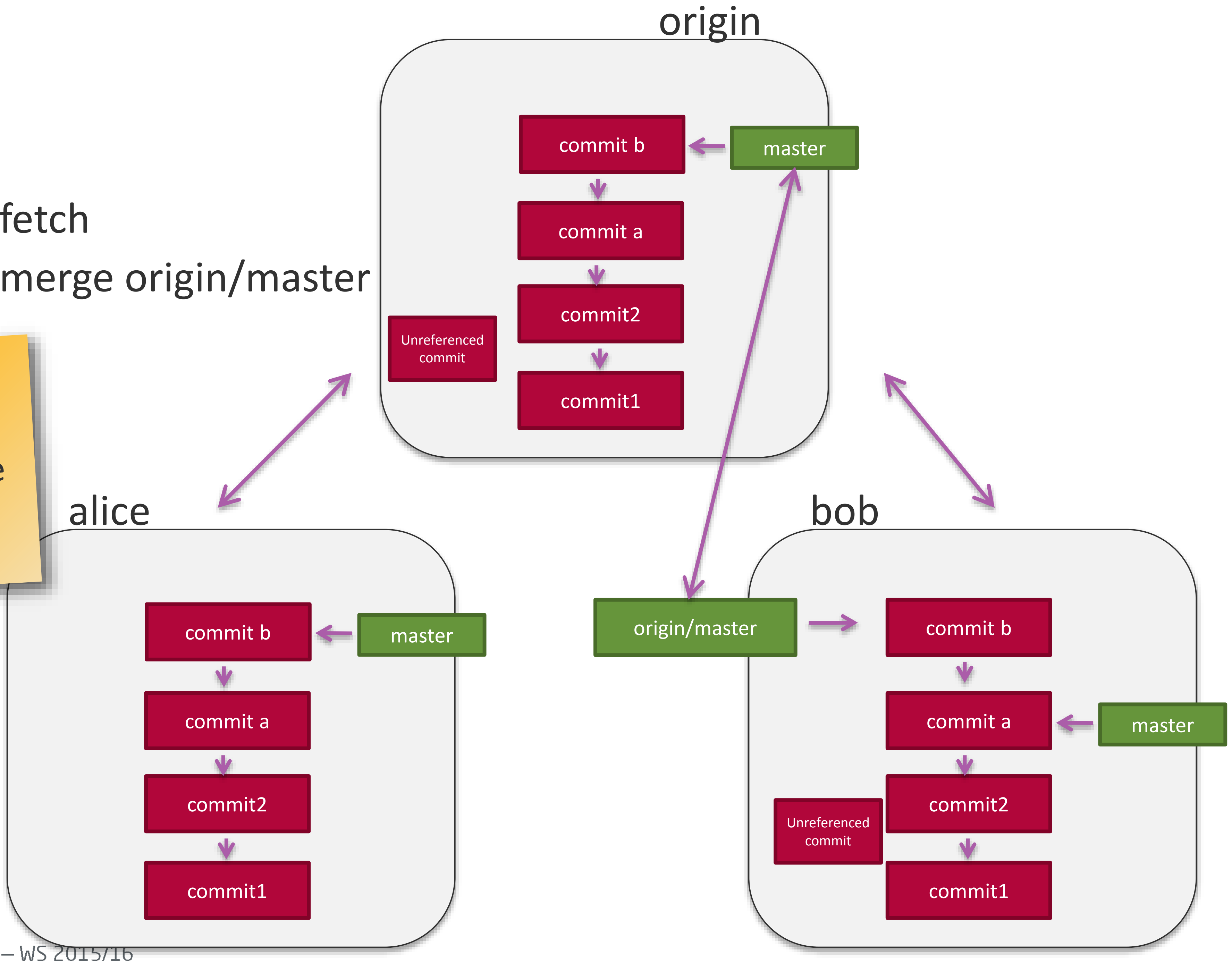


# Pull

```
bob$ git fetch
bob$ git merge origin/master
```

**Tip:**

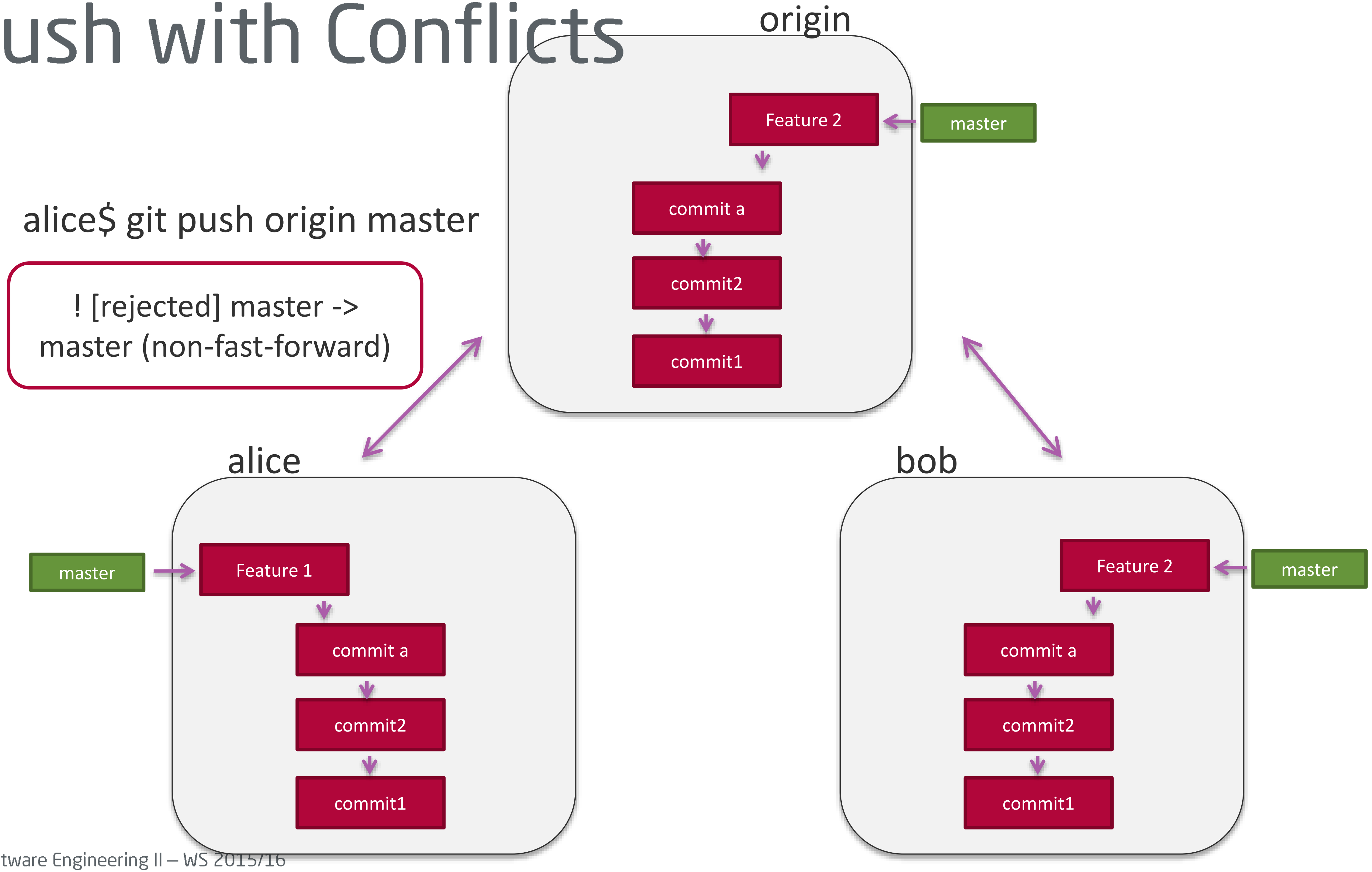
fetch + merge = pull



# Push with Conflicts

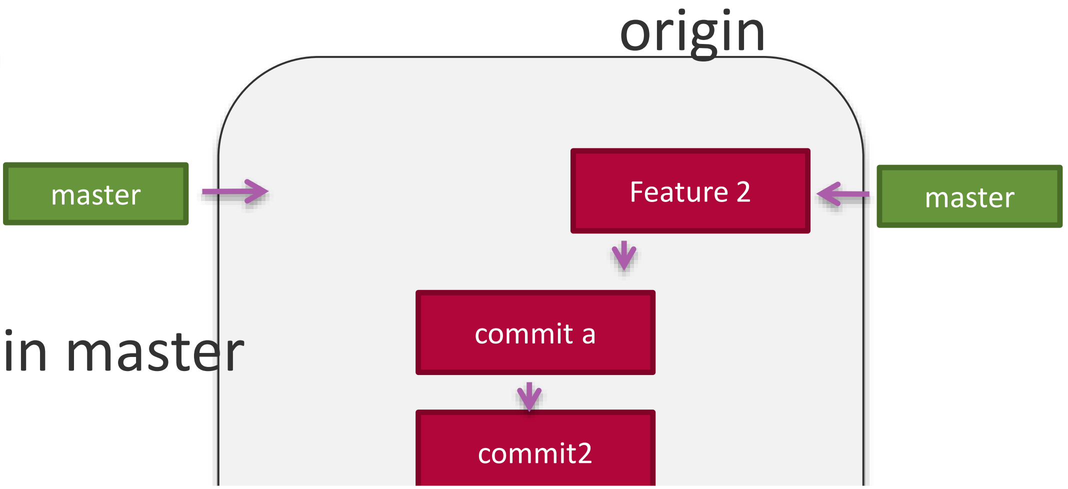
alice$ git push origin master

! [rejected] master -> master (non-fast-forward)

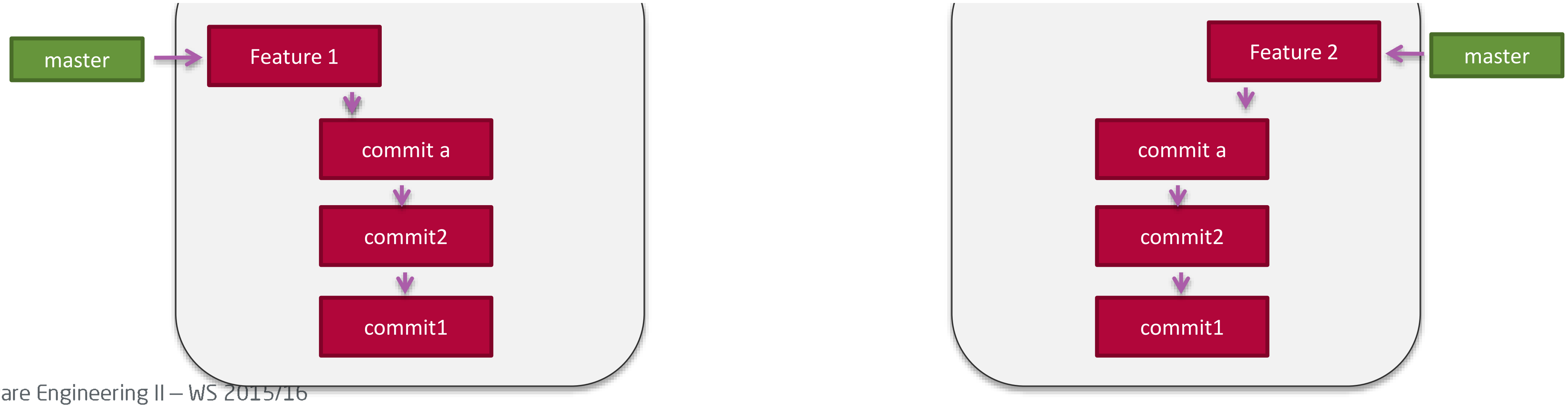


# Push --force

alice$ git push --f origin master

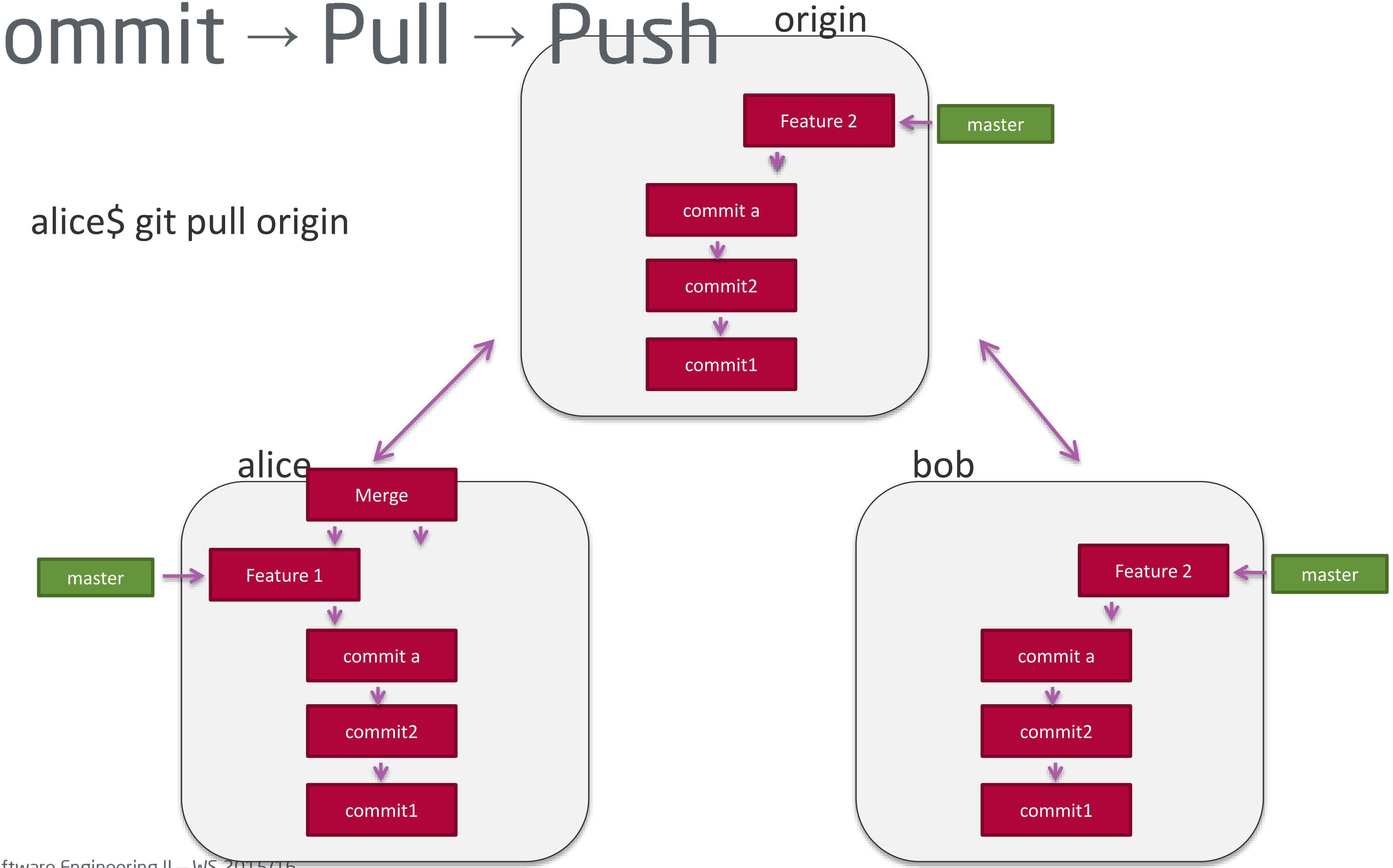


**Never EVER push --force**



# Commit → Pull → Push

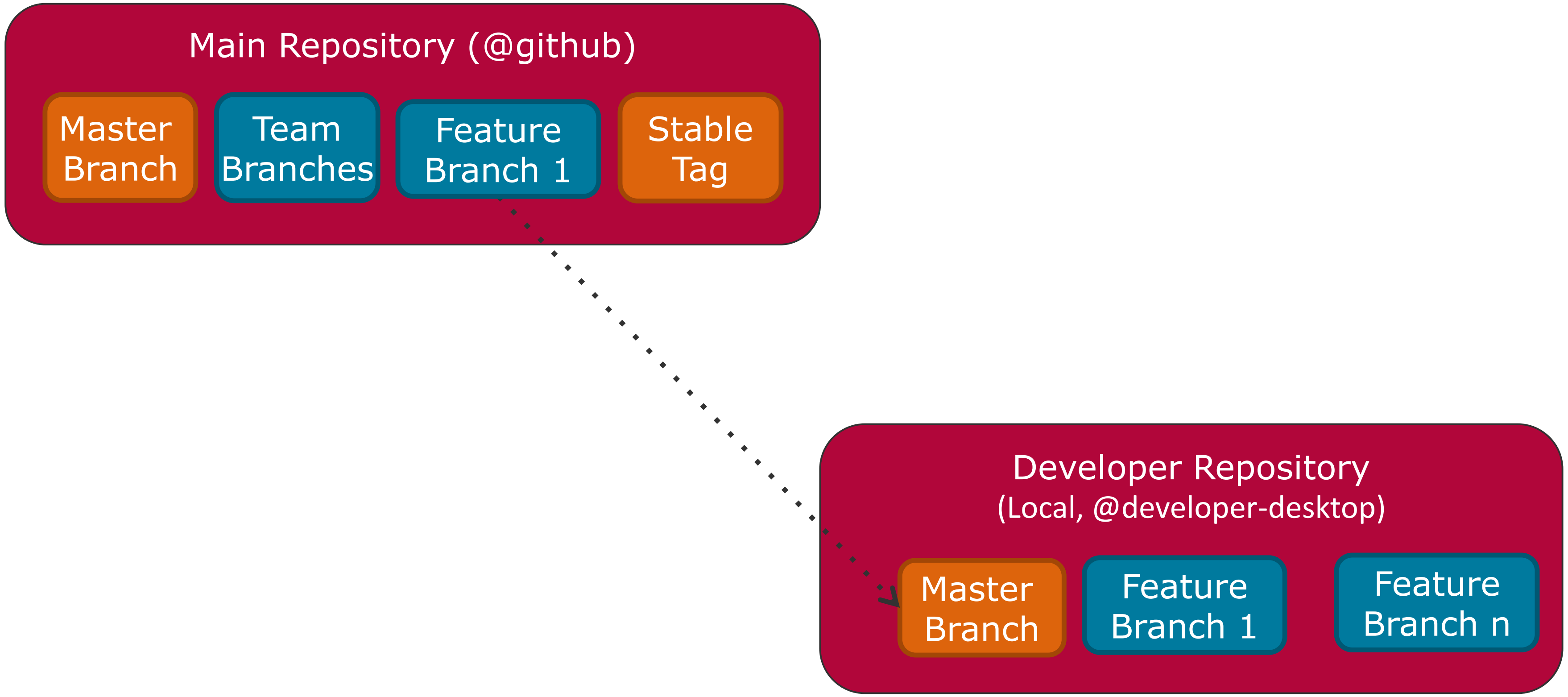
alice$ git pull origin



# What happened?

- git log
- git diff
- git blame

# Project Repository Setup



Main Repository (@github)

Master Branch

Team Branches

Feature Branch 1

Stable Tag

Developer Repository

(Local, @developer-desktop)

Master Branch

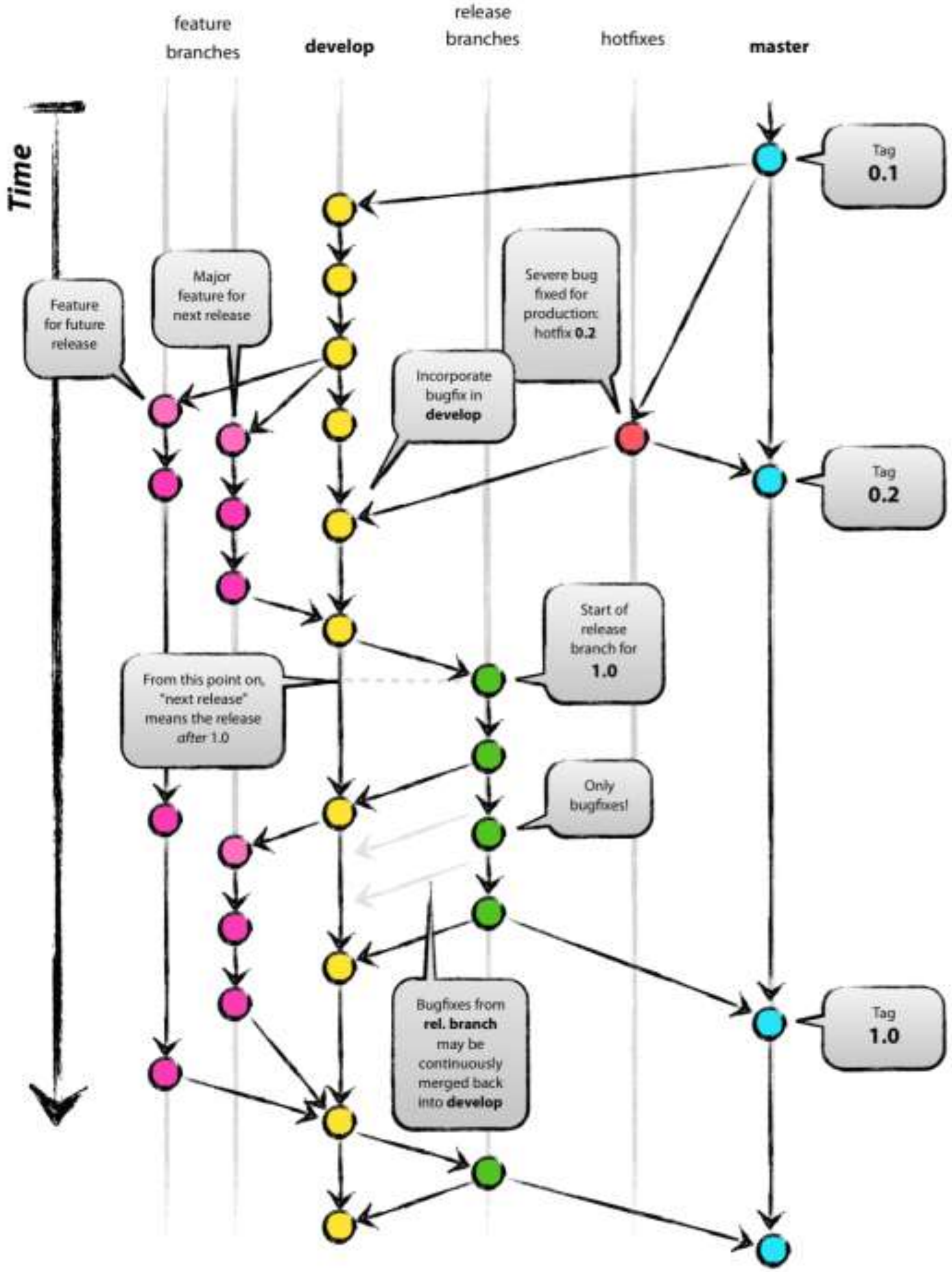
Feature Branch 1

Feature Branch n

# Branching

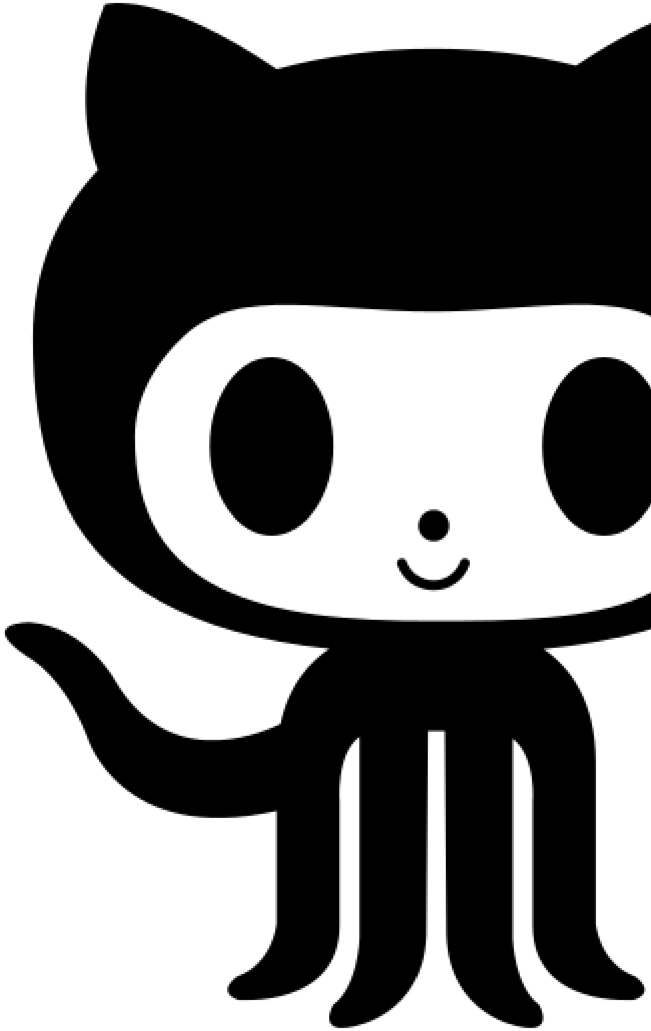
- http://nvie.com/posts/a-successful-git-branching-model/

- Never merge in master or release branches
- Never break build in shared branches


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

# Outline

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



# Next Weeks' Schedule

Week 1 (Oct 12 – Oct 16)

- Introduction lectures

Week 2 (Oct 19 – Oct 23)

- Find teams, **enroll**!
- Code School exercise
- Lecture on Scrum
  - Exercise after lunch!

Week 3 (Oct 26 – Oct 30)

- POs: Customer meeting
- Ruby on Rails exercise
- Lecture on Git and testing

Week 4 (Nov 2 – Nov 6)

- Kick-off presentation
- Start of project

# Image Credits

- https://www.flickr.com/photos/98701585@N02/9351589556/
- "Olive tree Karystos". Licensed under Public domain via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Olive_tree_Karystos.jpg#mediaviewer/File:Olive_tree_Karystos.jpg
- http://christmasstockimages.com/free/ideas_concepts/slides/gift_tag.htm