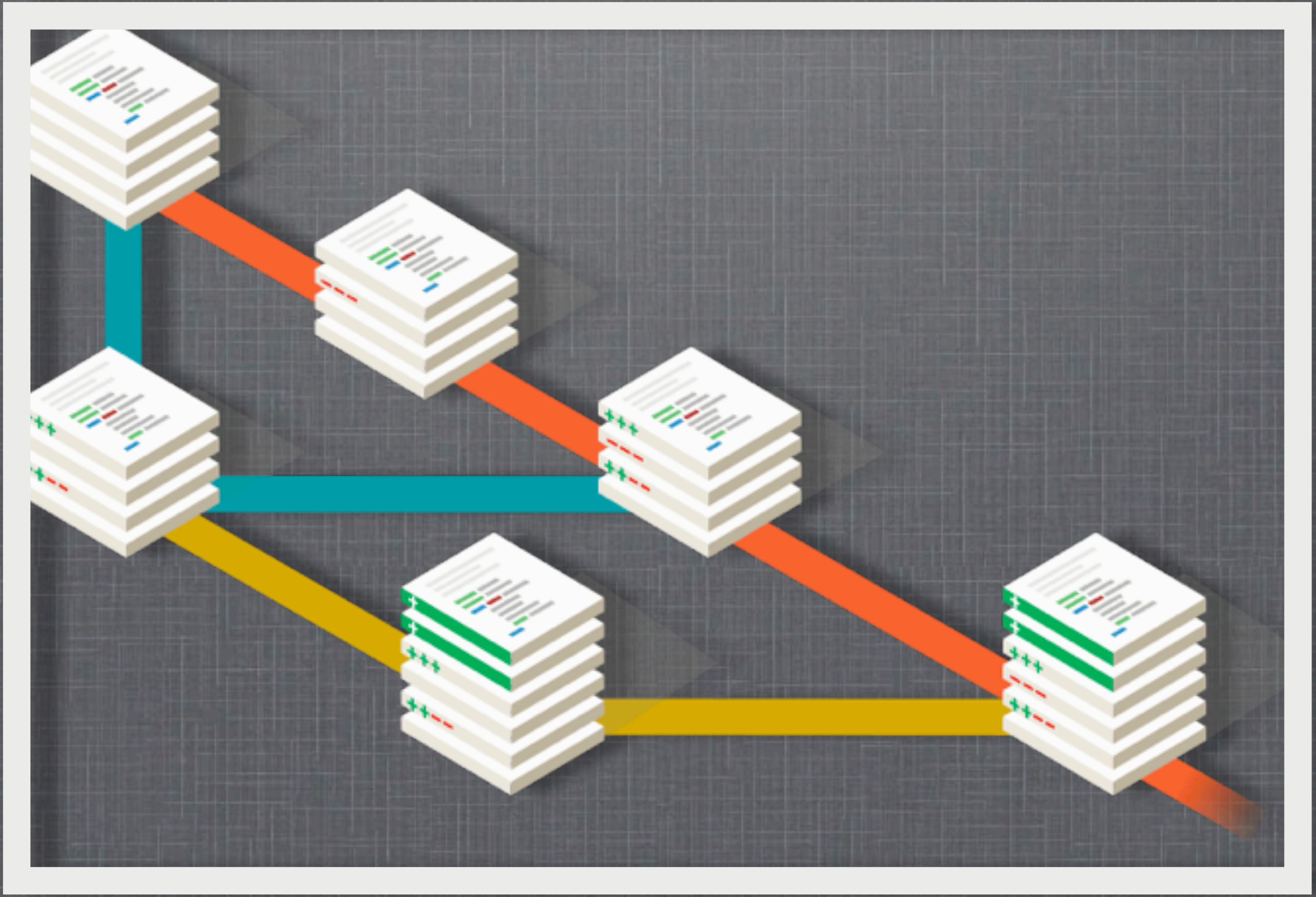
GIT

Ming-Hsien Tsai



SDM 2013

this picture is taken from http://git-scm.com

WHAT IS GIT

- Git is
  - a version control system (VCS)
  - free
  - open source
  - distributed

WHY VERSION CONTROL

version 1

WHY VERSION CONTROL

version 1

version 2

WHY VERSION CONTROL

version 1

version 3

version 2

WHY VERSION CONTROL

version 1

version 3

version 2

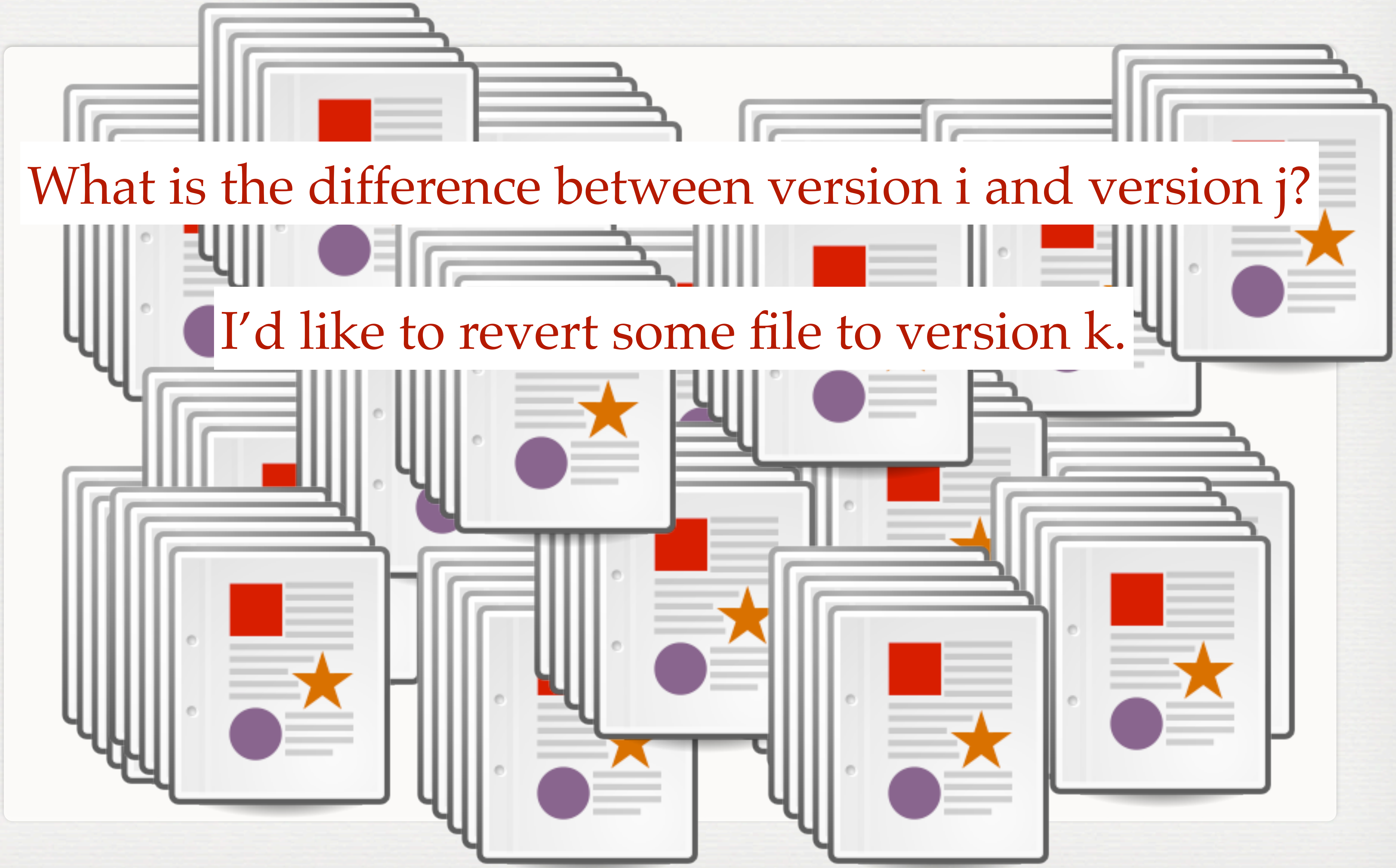
version 4

WHY VERSION CONTROL

WHY VERSION CONTROL

What is the difference between version i and version j?

WHY VERSION CONTROL



What is the difference between version i and version j?

I'd like to revert some file to version k.

WHY VERSION CONTROL

What is the difference between version i and version j?

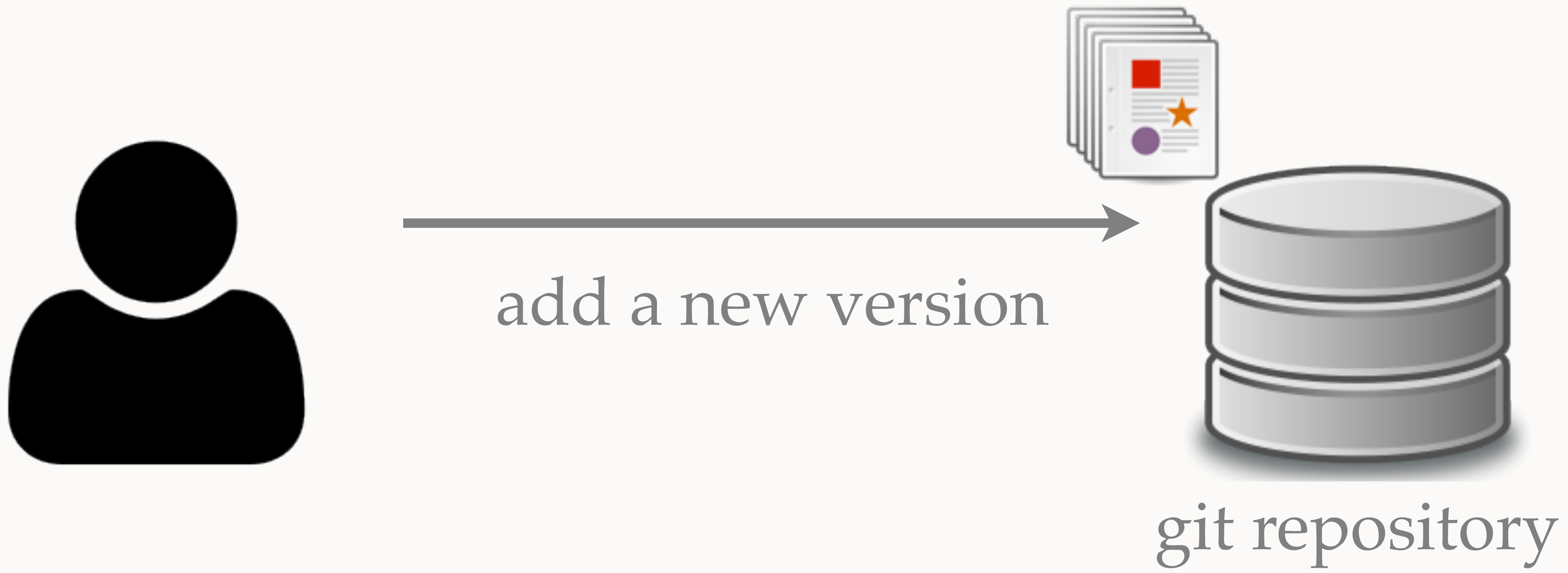
I'd like to revert some file to version k.

You need a VCS!

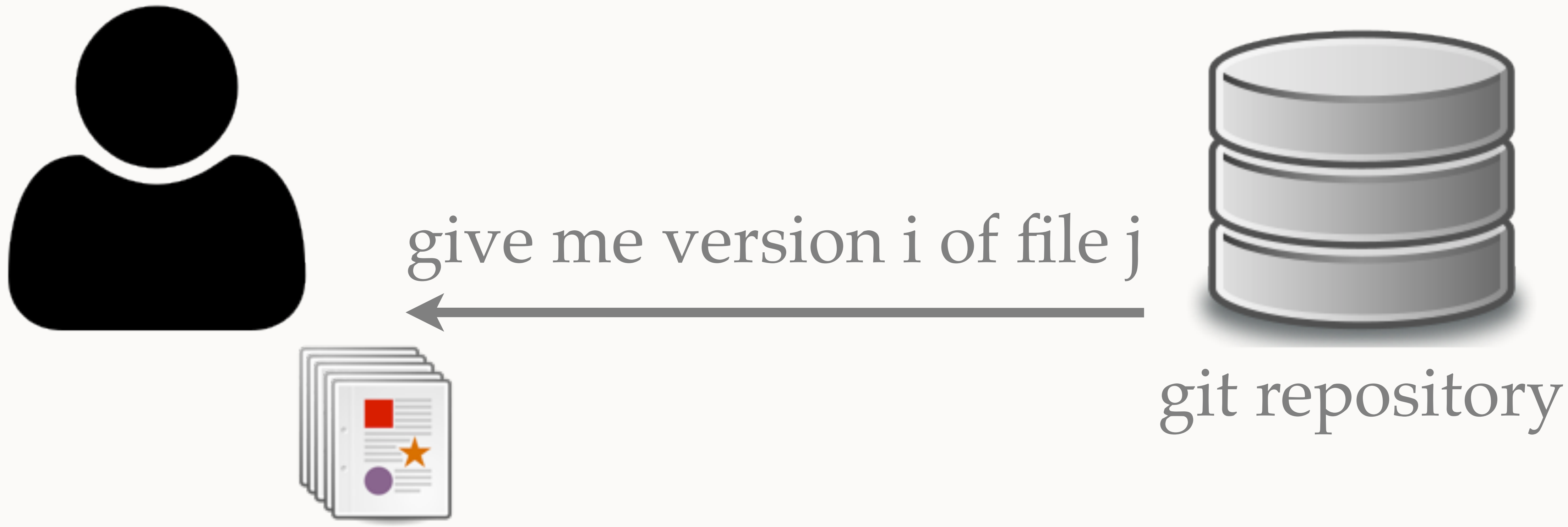
WITH GIT (1/2)

git repository

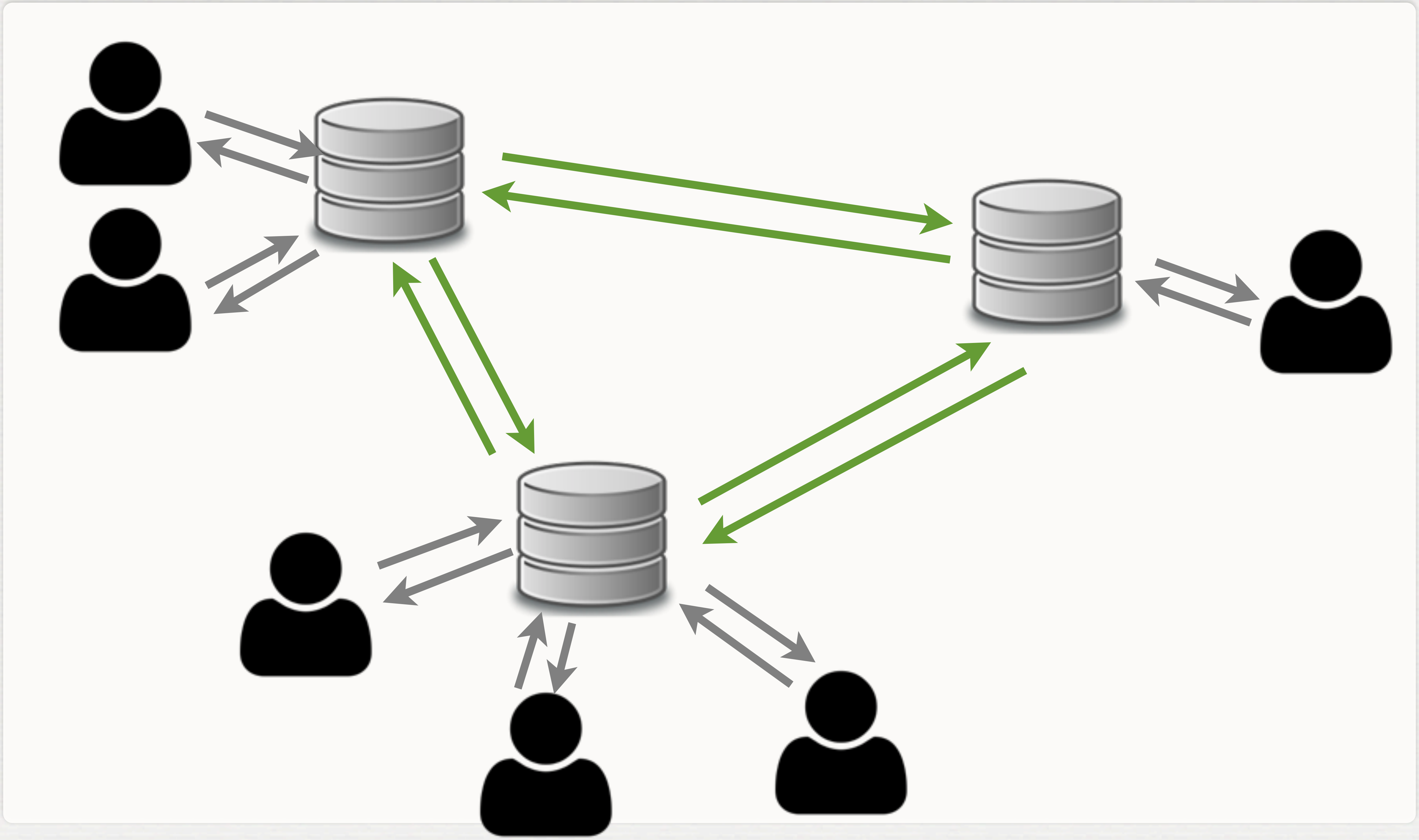
WITH GIT (1/2)



WITH GIT (1/2)



WITH GIT (2/2)



PROJECTS USING GIT

- Linux kernel
- Android
- Egit/jgit
- Fedora
- FFmpeg
- gcc
- jQuery
- ......

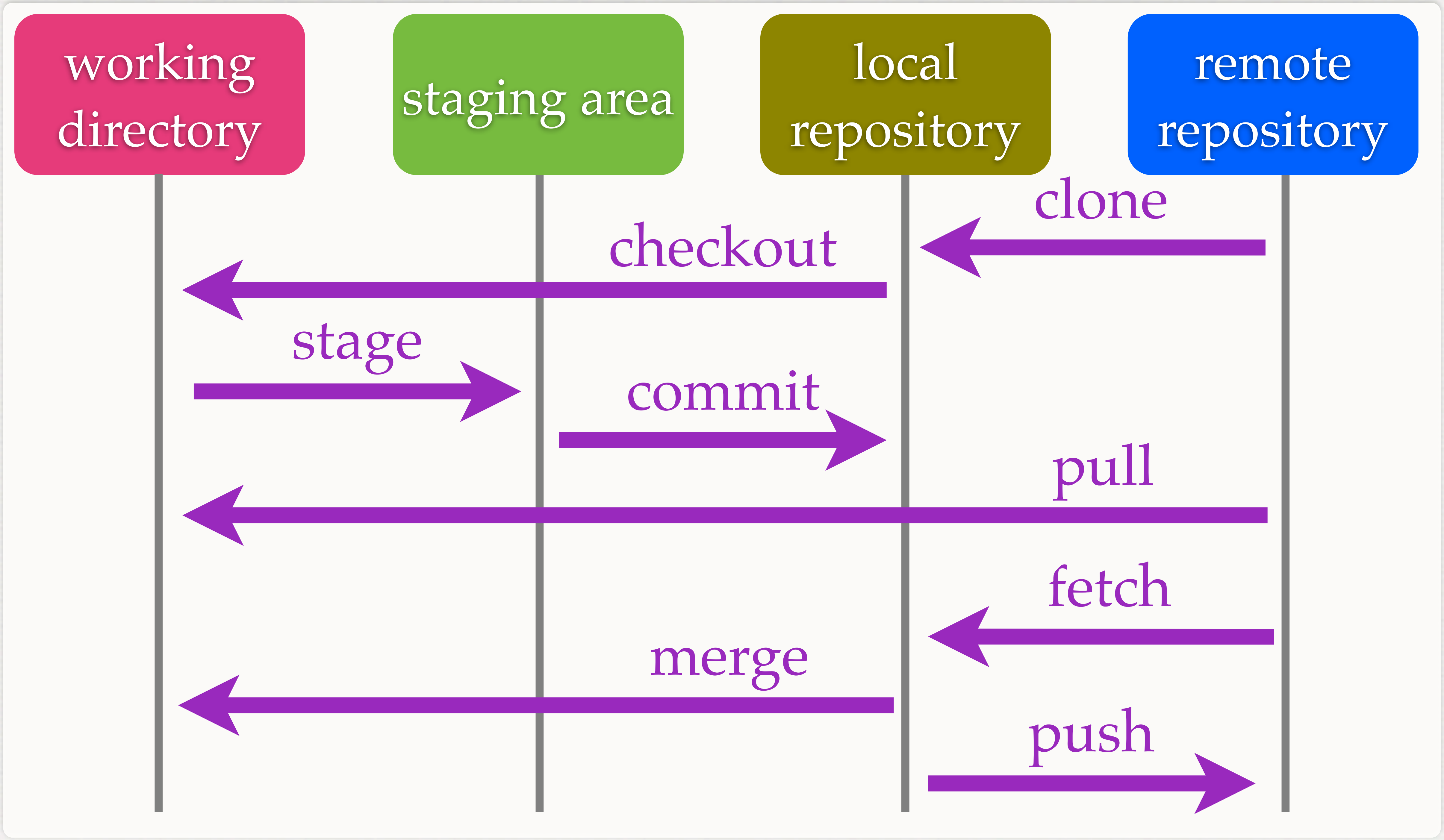
OTHER VCS

- CVS
- Subversion (SVN)
- Mercurial
- Rational Team Concert
- Visual SourceSafe
- ...

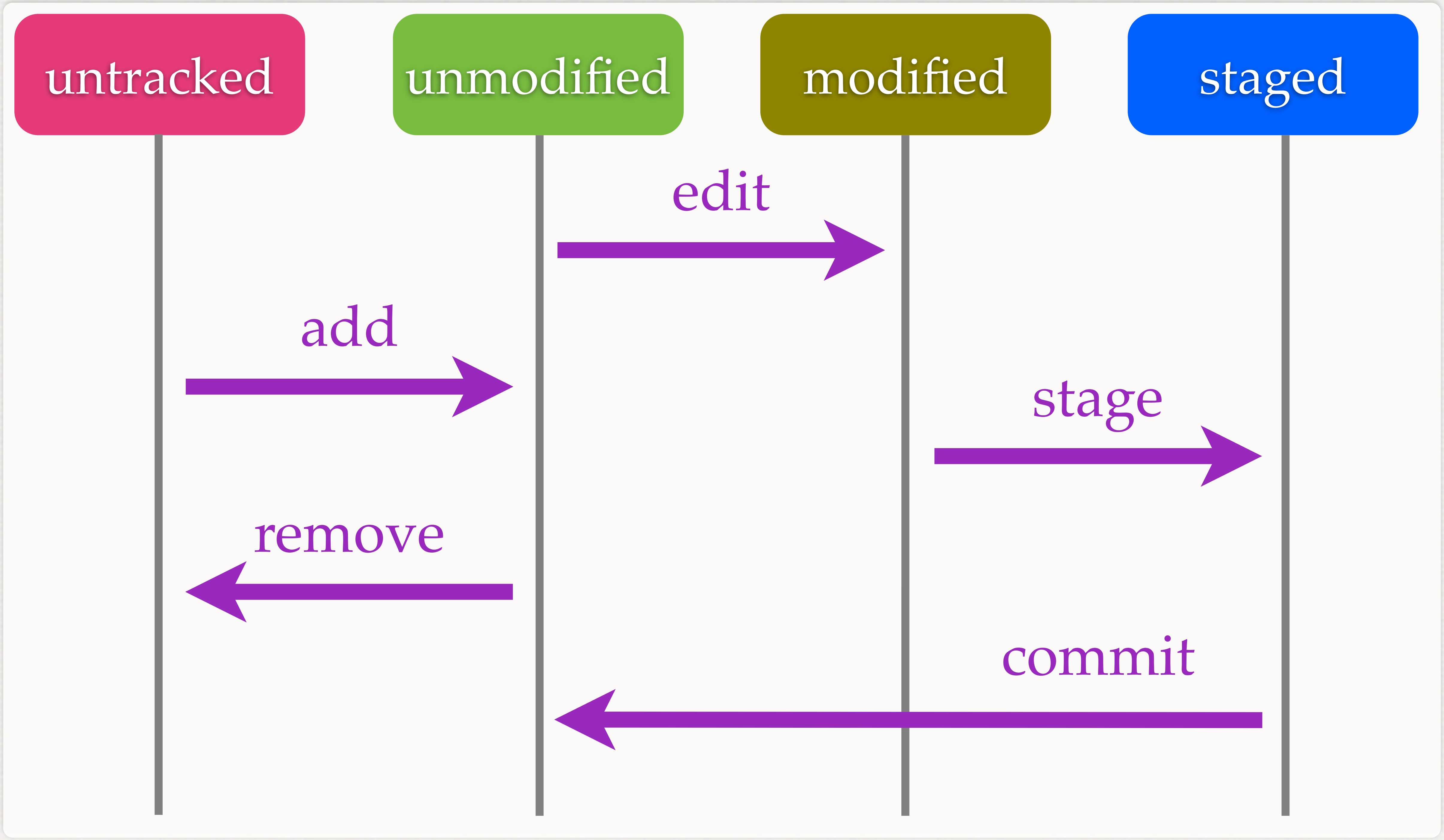
PROJECT HOSTING

- GitHub (http://github.com/):
  - git
- Bitbucket (http://gitbucket.org/)
  - git, mercuial
- Google Code (http://code.google.com/)
  - svn

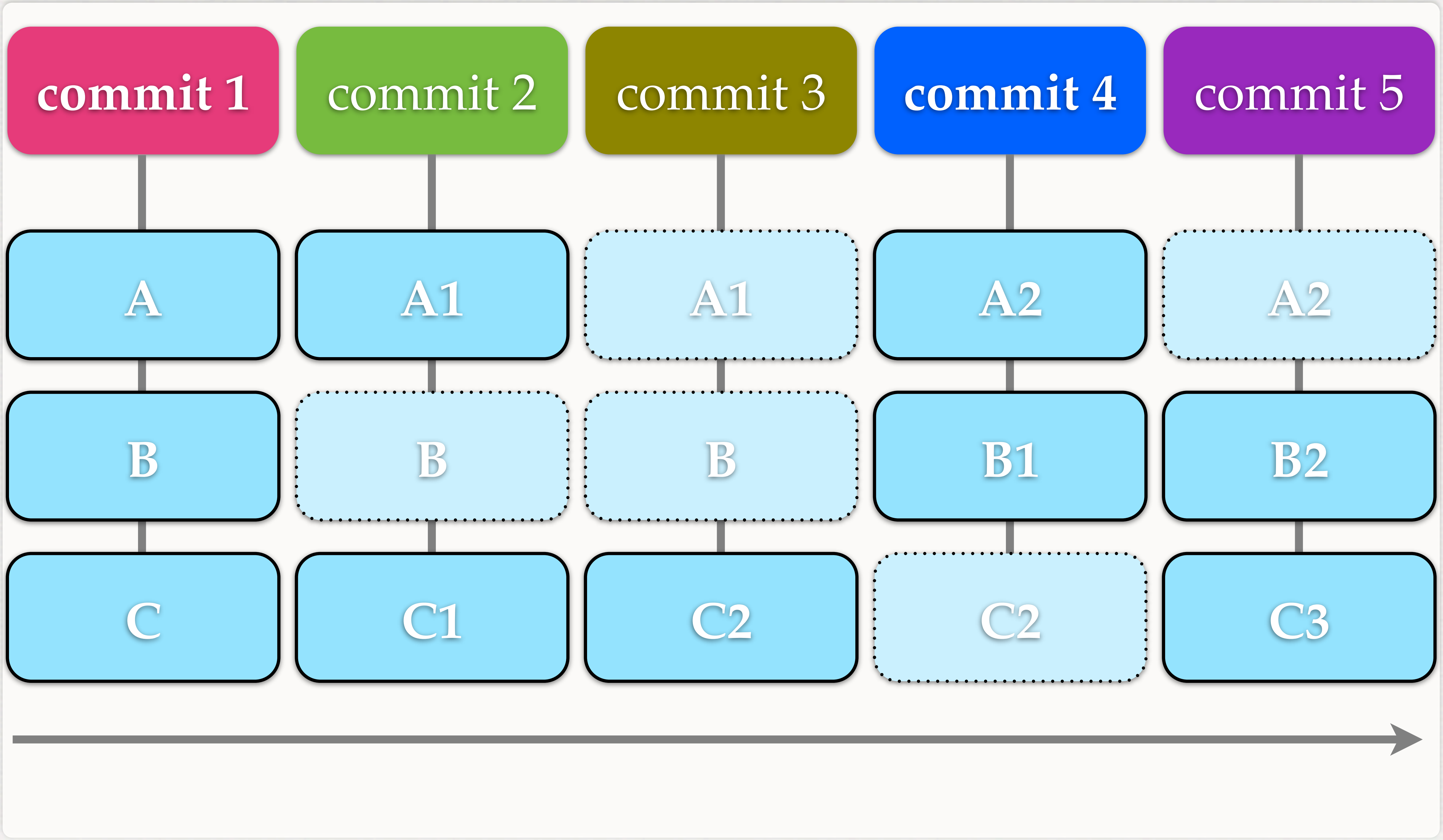
WORKING WITH GIT



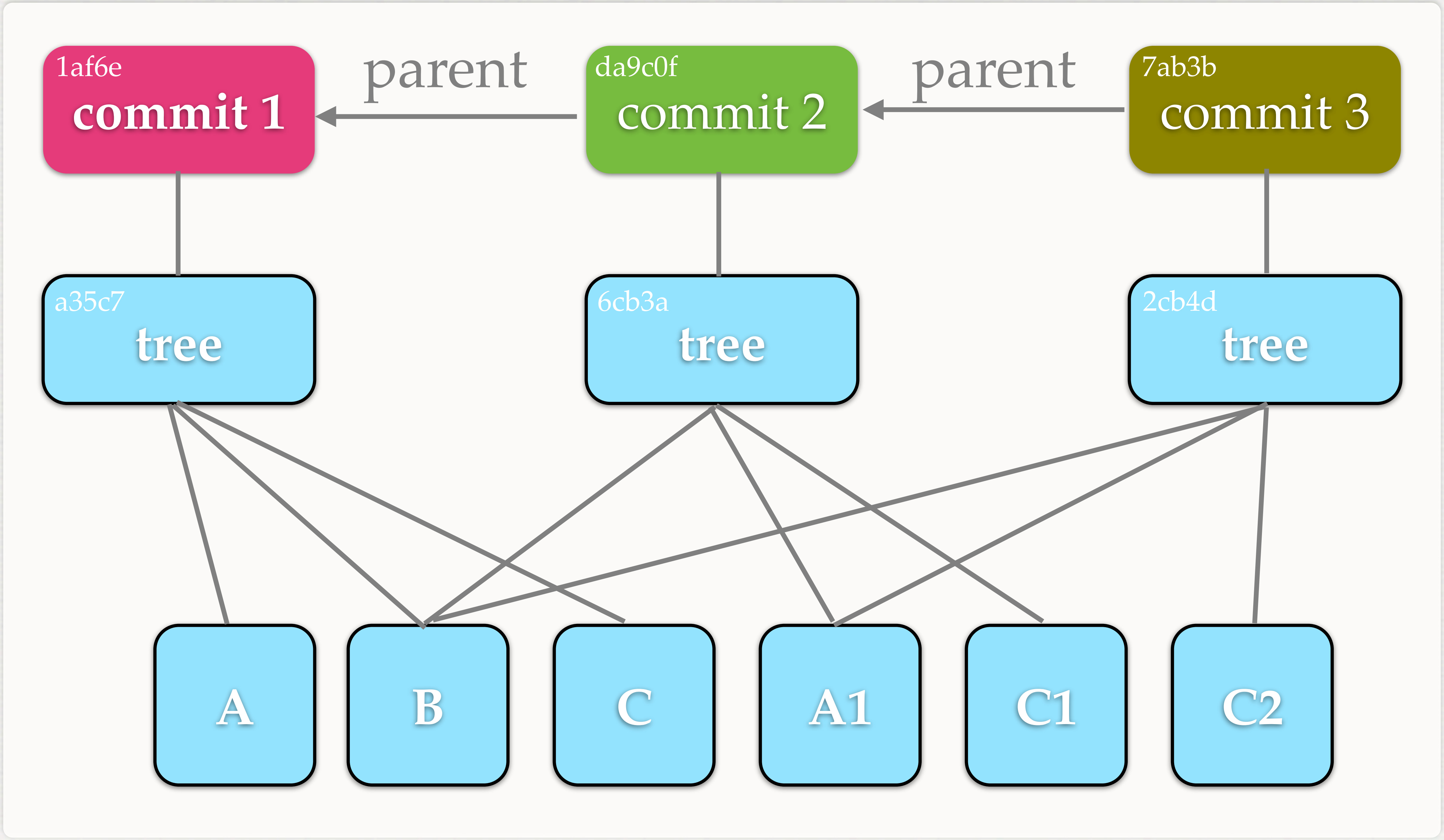
FILE STATUS LIFECYCLE



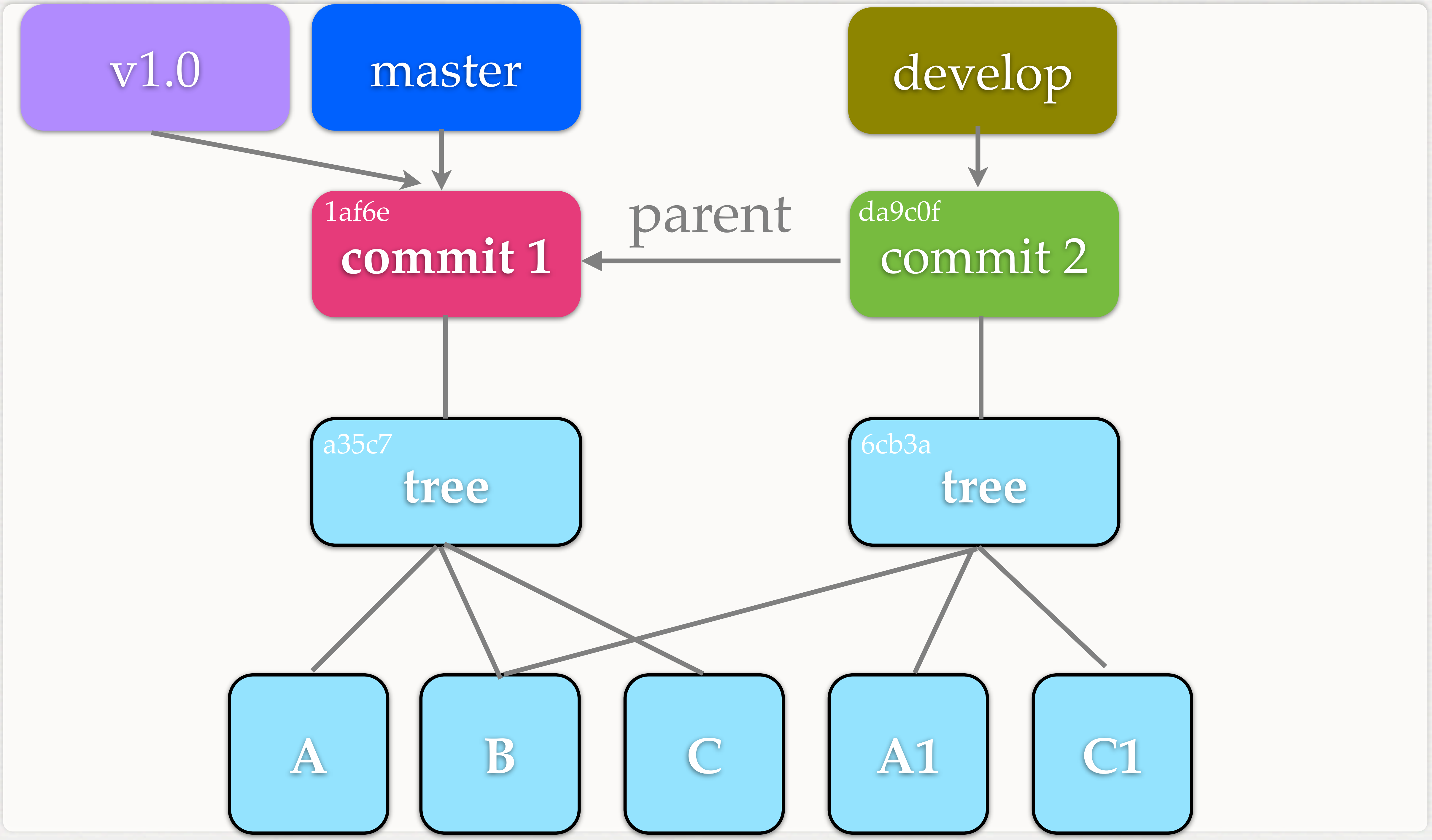
SNAPSHOTS


commit 1
commit 2
commit 3
commit 4
commit 5
A
A1
A1
A2
A2
B
B
B
B1
B2
C
C1
C2
C2
C3

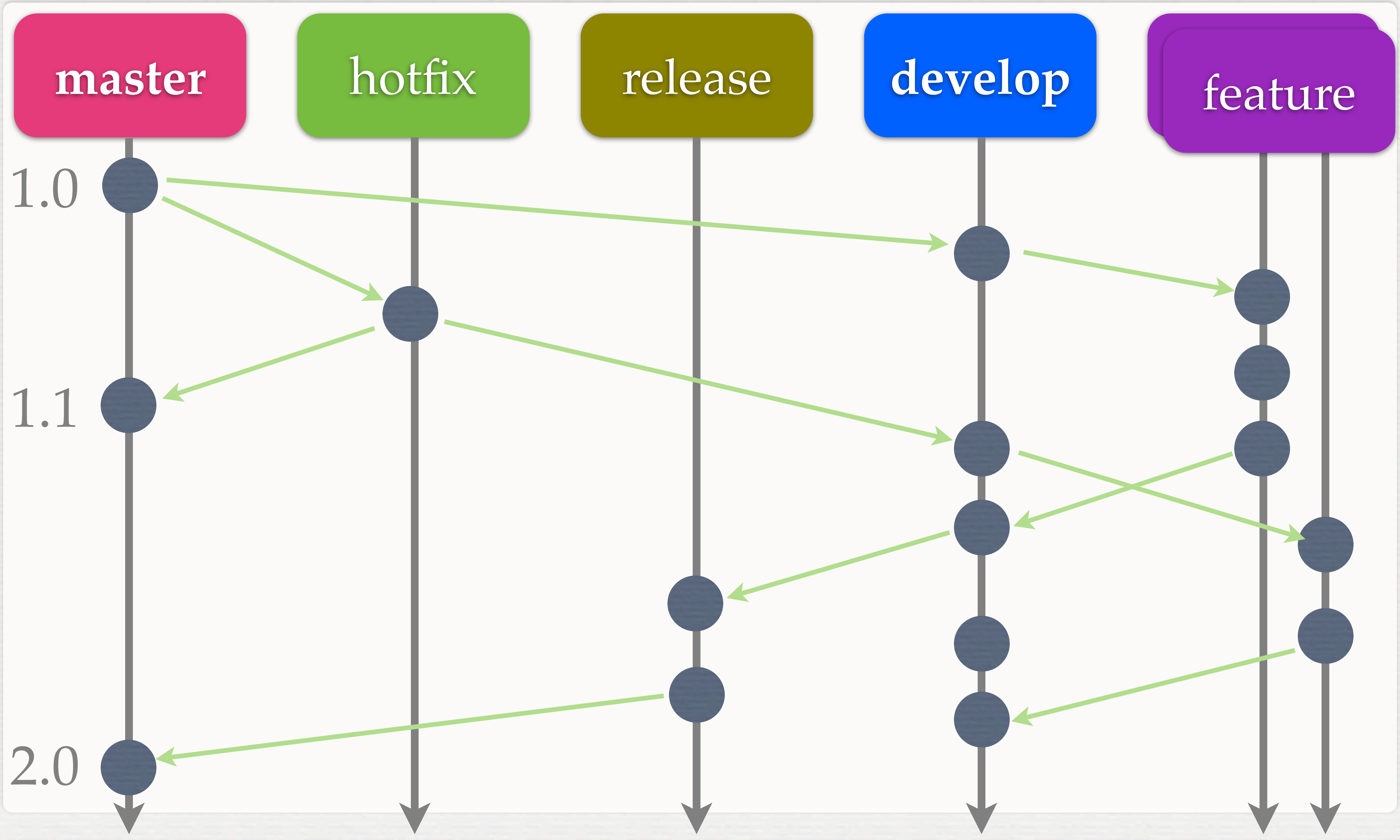
DATA MODEL


1af6e
commit 1
parent
da9c0f
commit 2
parent
7ab3b
commit 3
a35c7
tree
6cb3a
tree
2cb4d
tree
A
B
C
A1
C1
C2

BRANCHES & TAGS

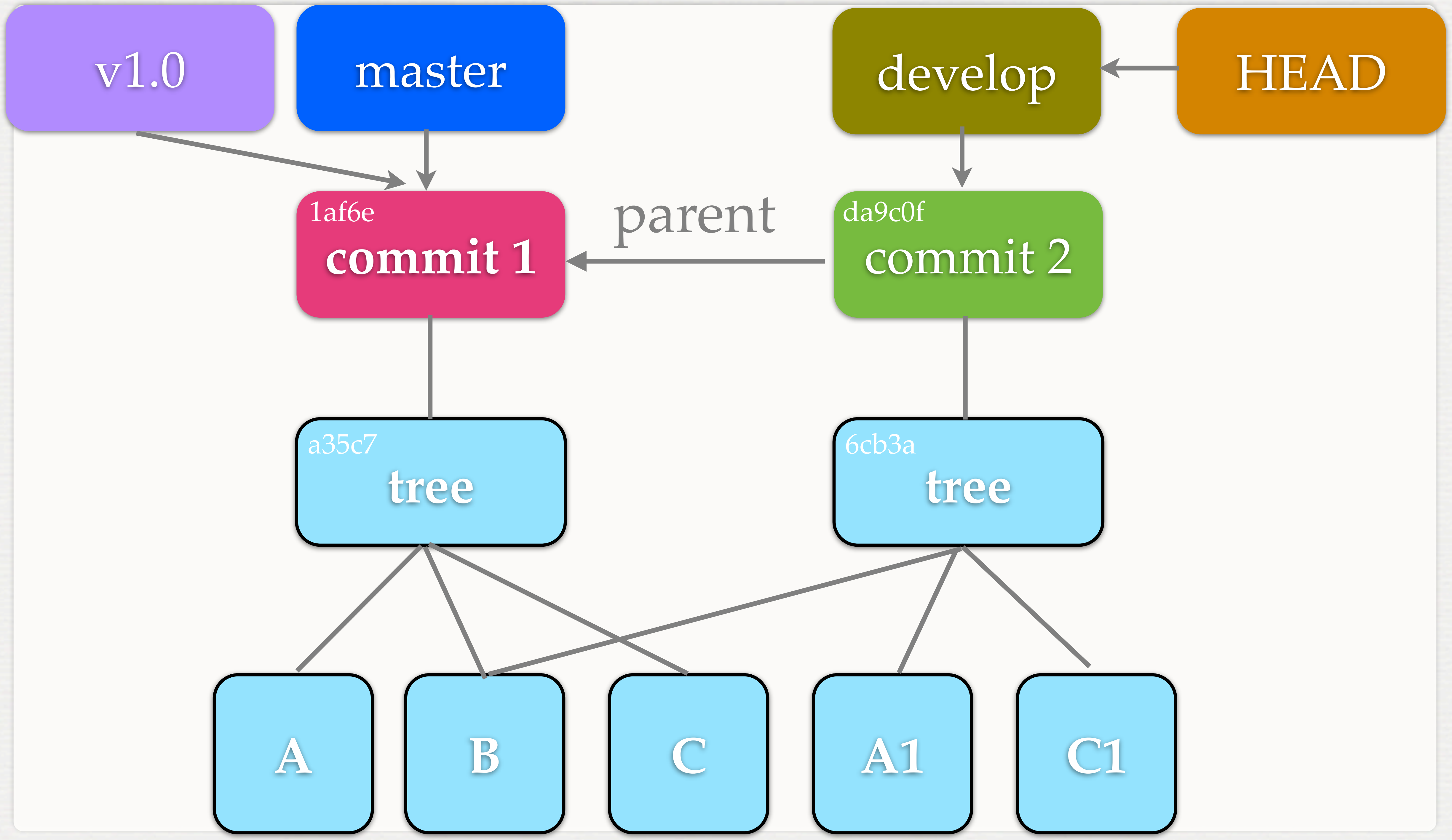


BRANCHING MODEL

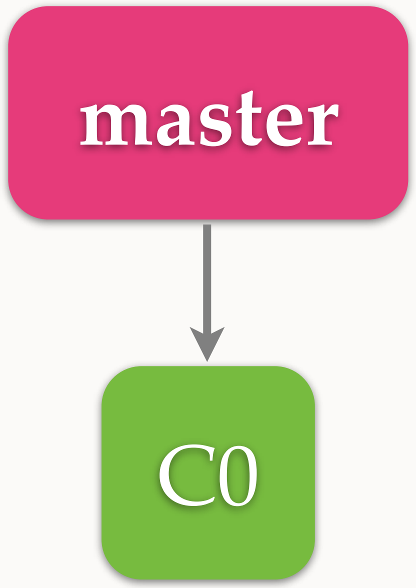


http://nvie.com/posts/a-successful-git-branching-model/

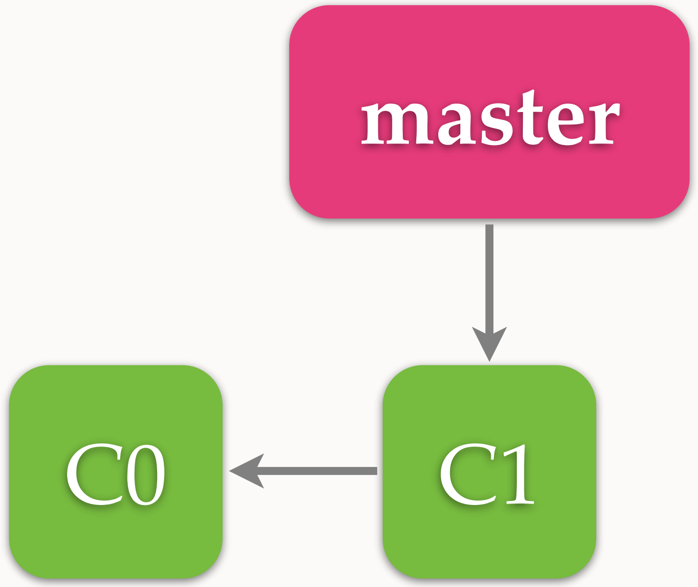
HEAD


v1.0
master
develop
HEAD
1af6e
commit 1
parent
da9c0f
commit 2
a35c7
tree
6cb3a
tree
A
B
C
A1
C1

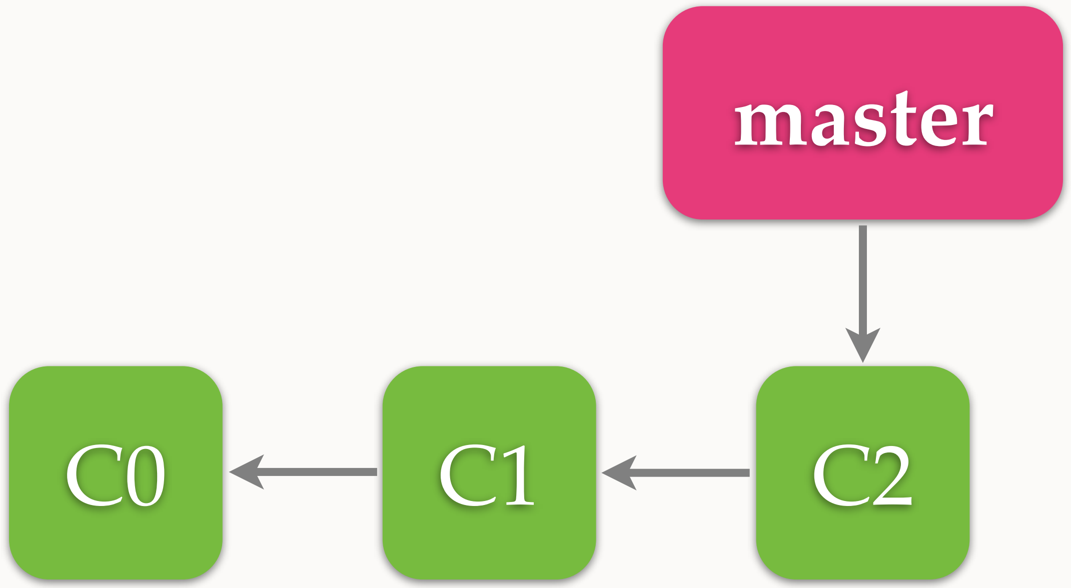
COMMITS



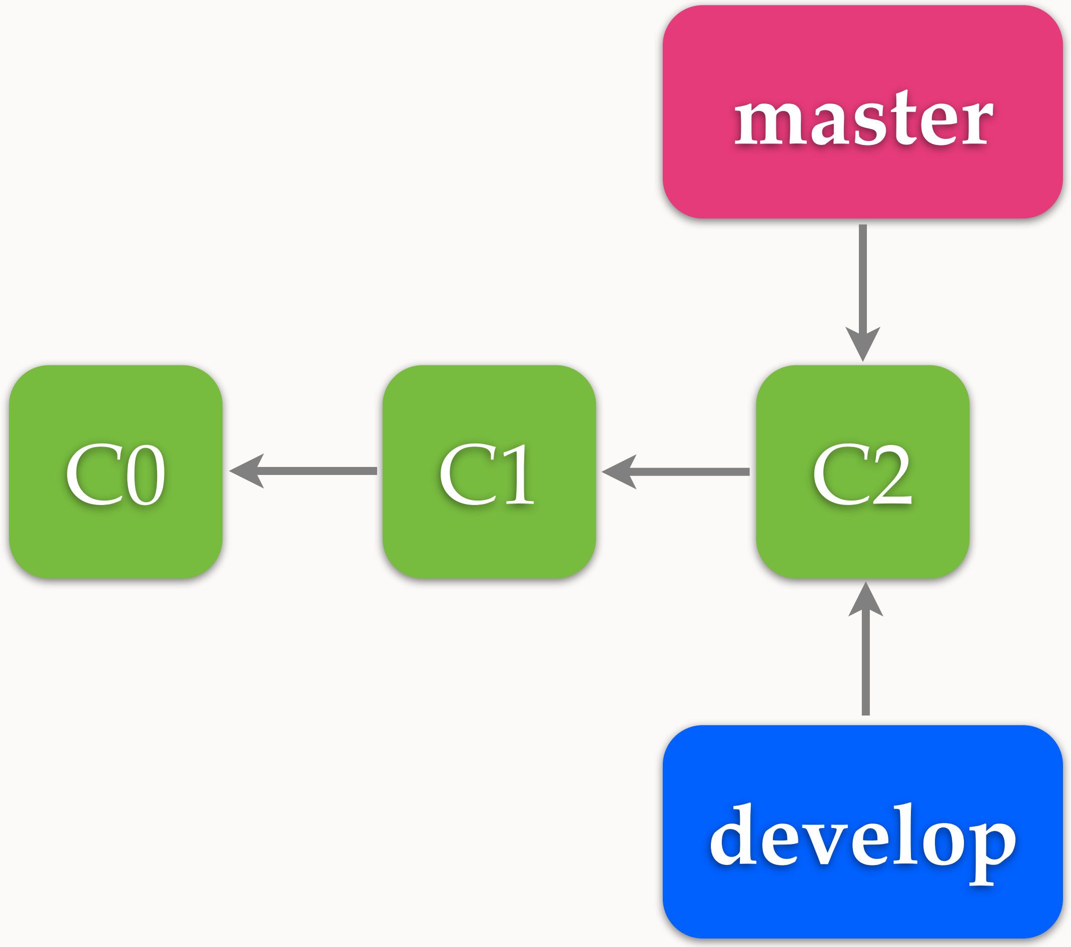
COMMITS



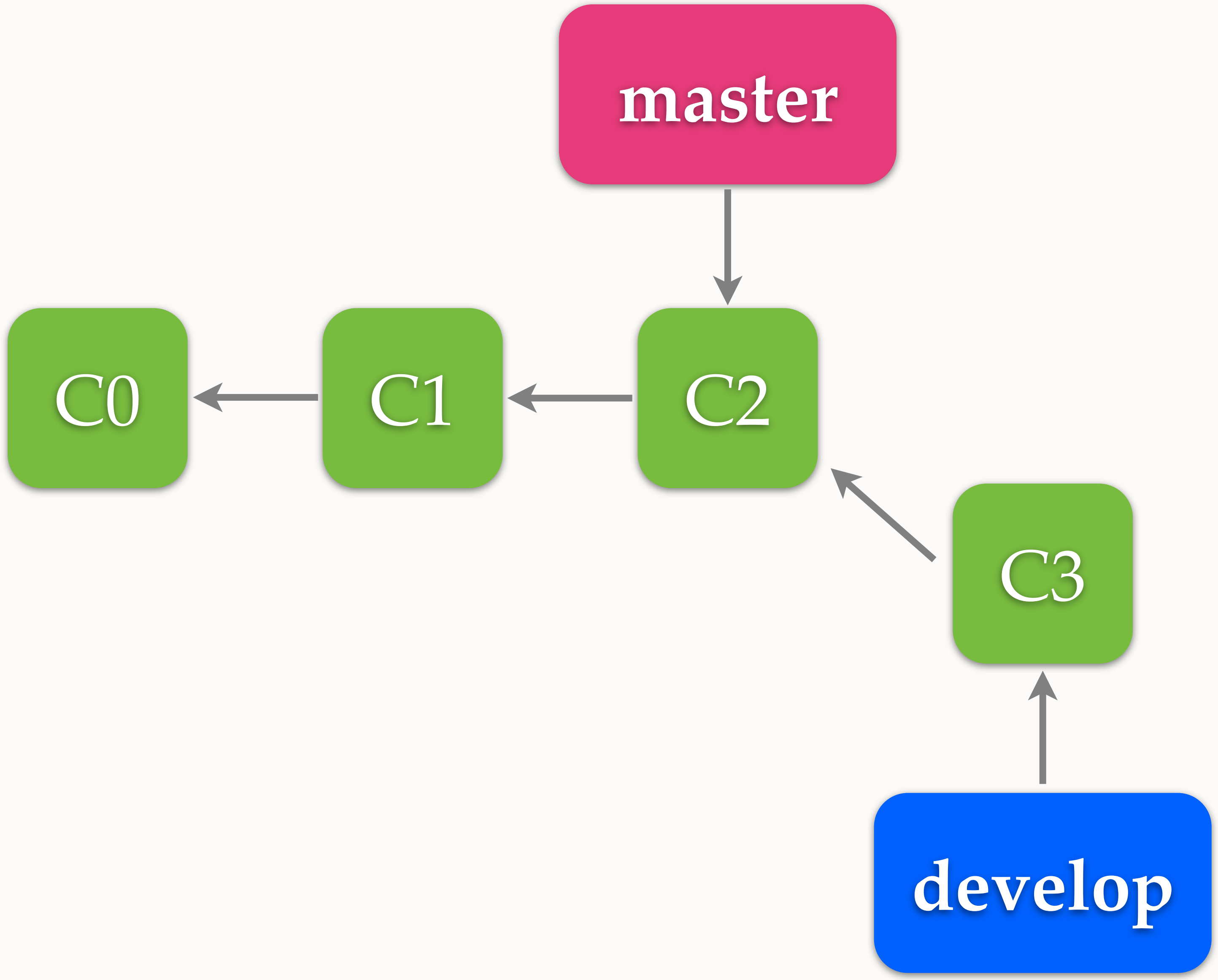
COMMITS



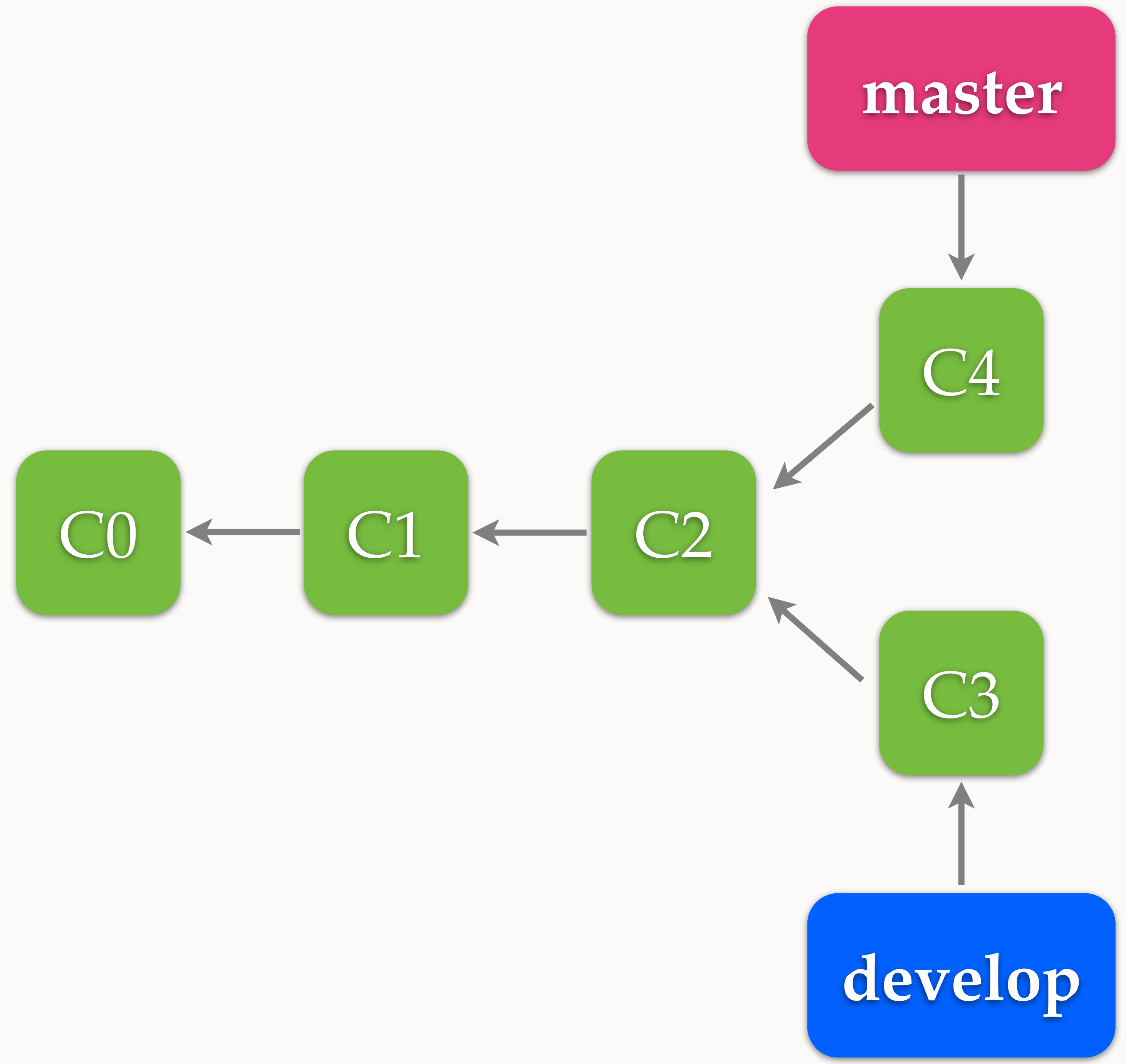
COMMITS



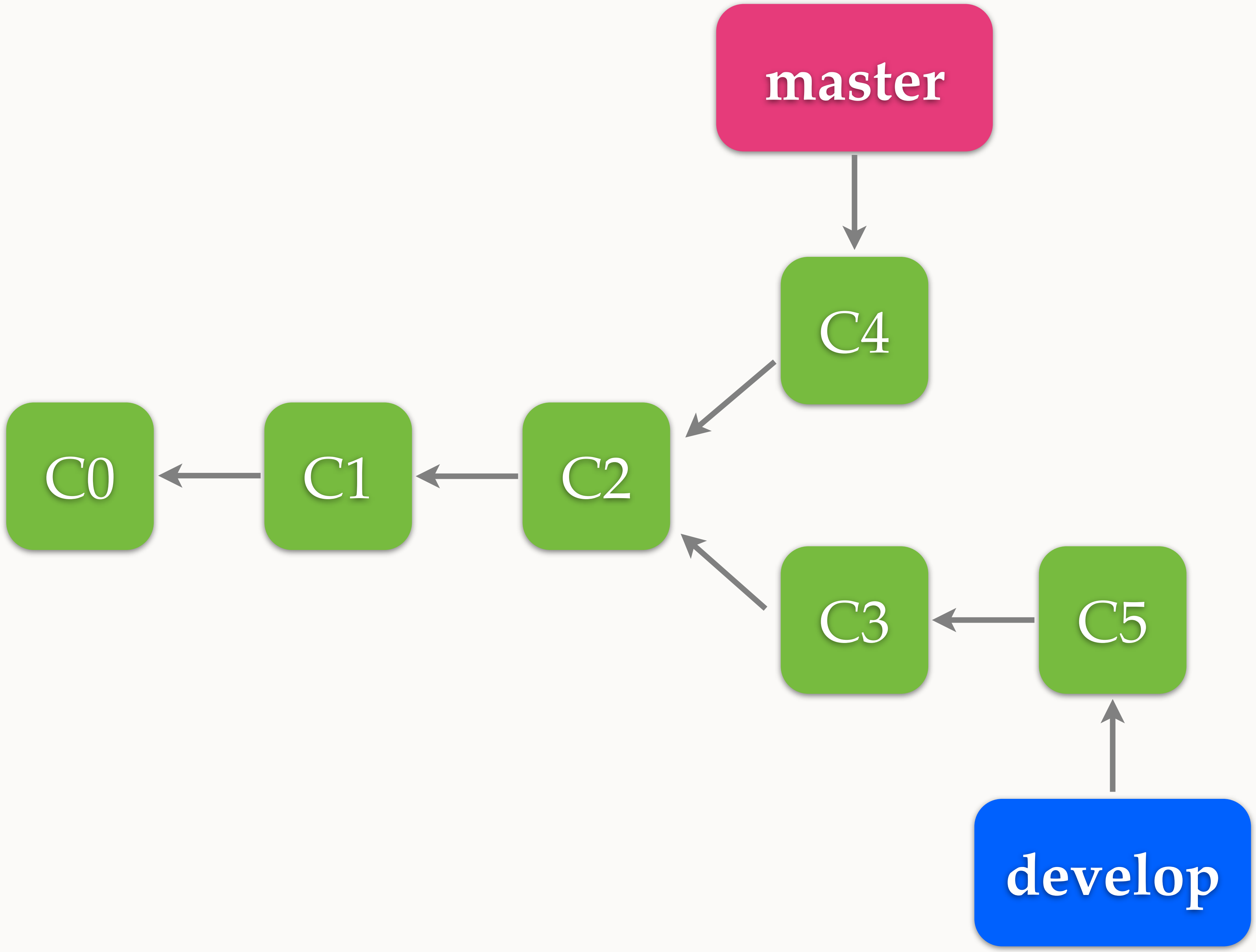
COMMITS



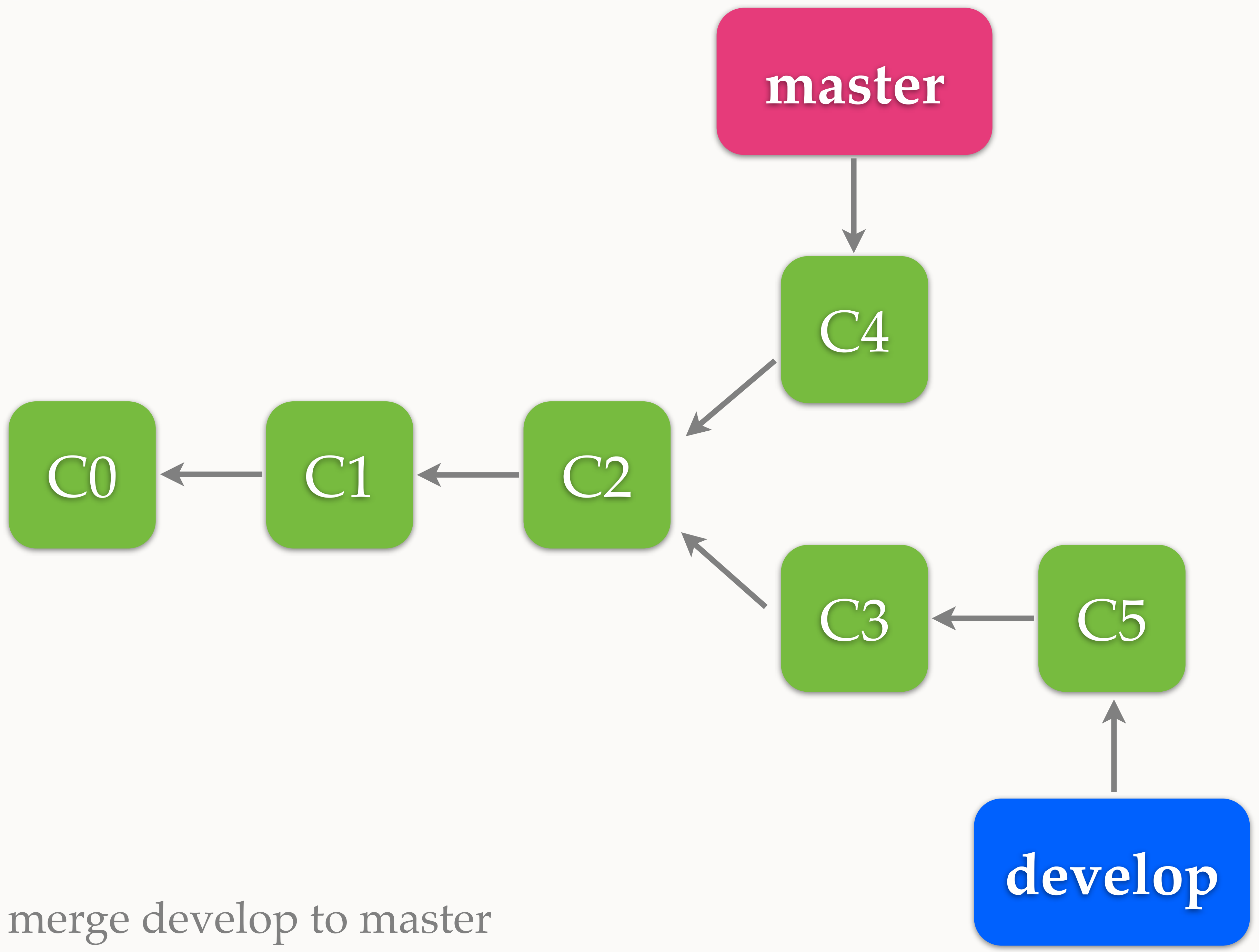
COMMITS



COMMITS

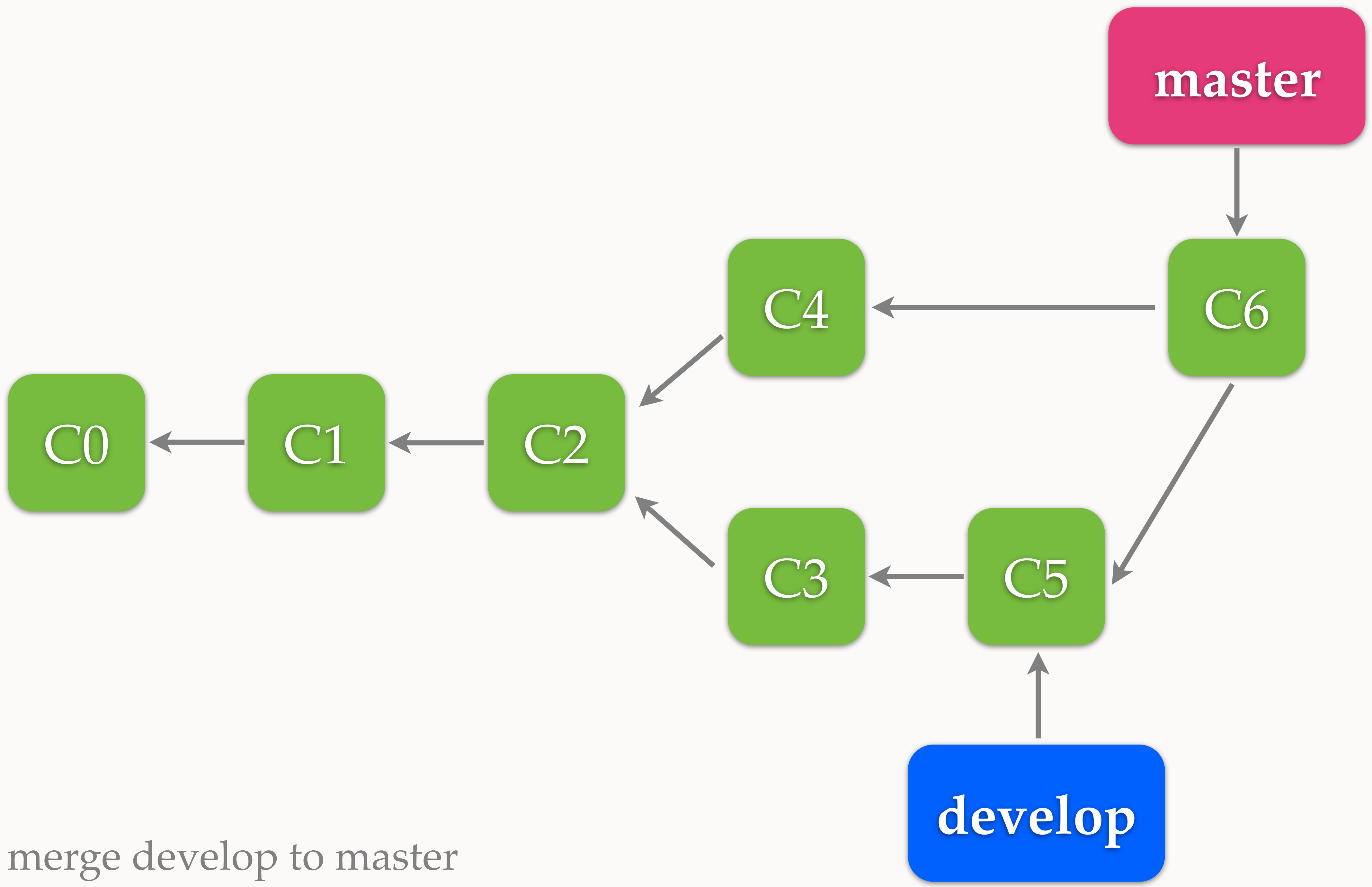


MERGE



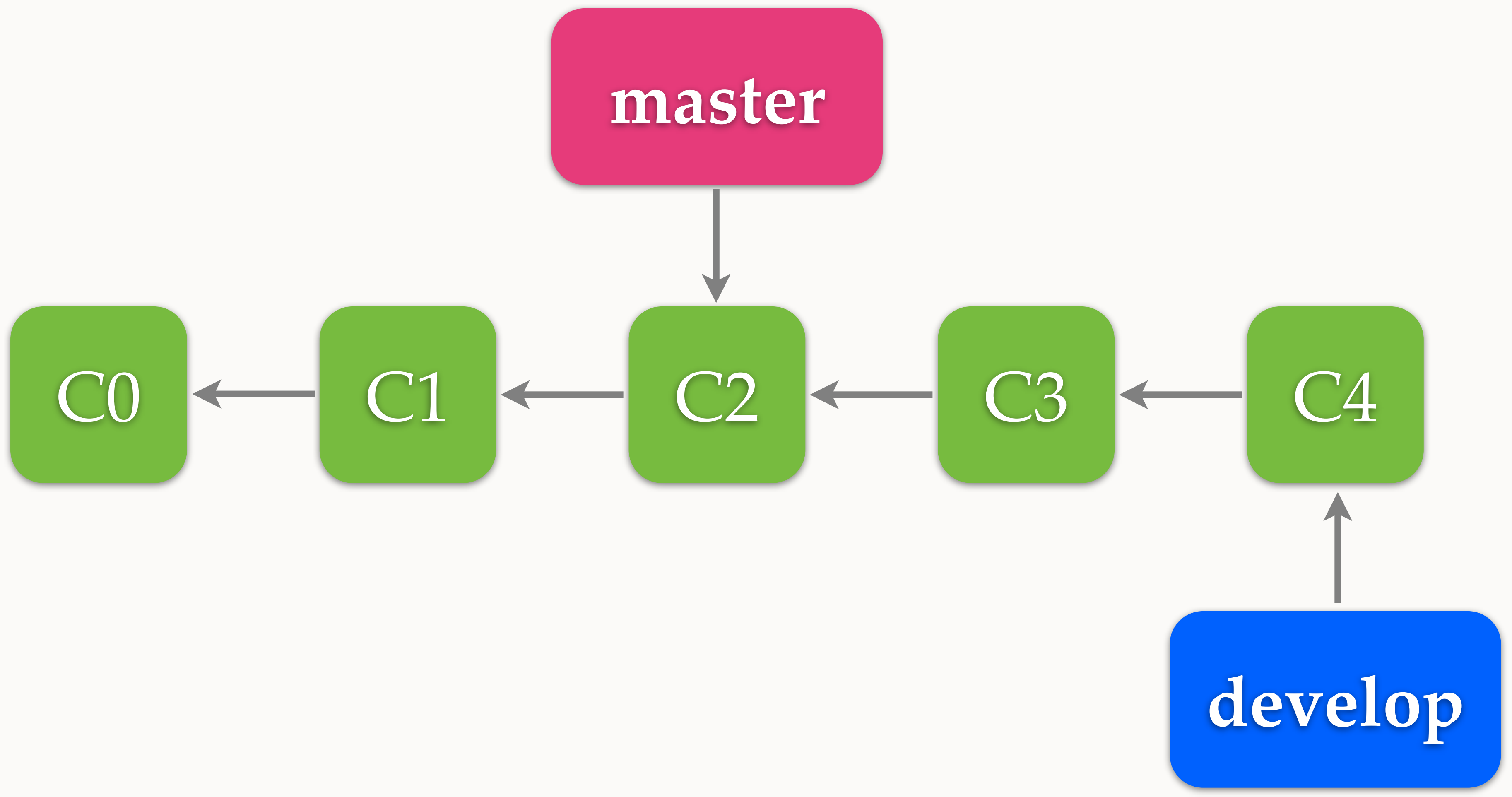
merge develop to master

MERGE



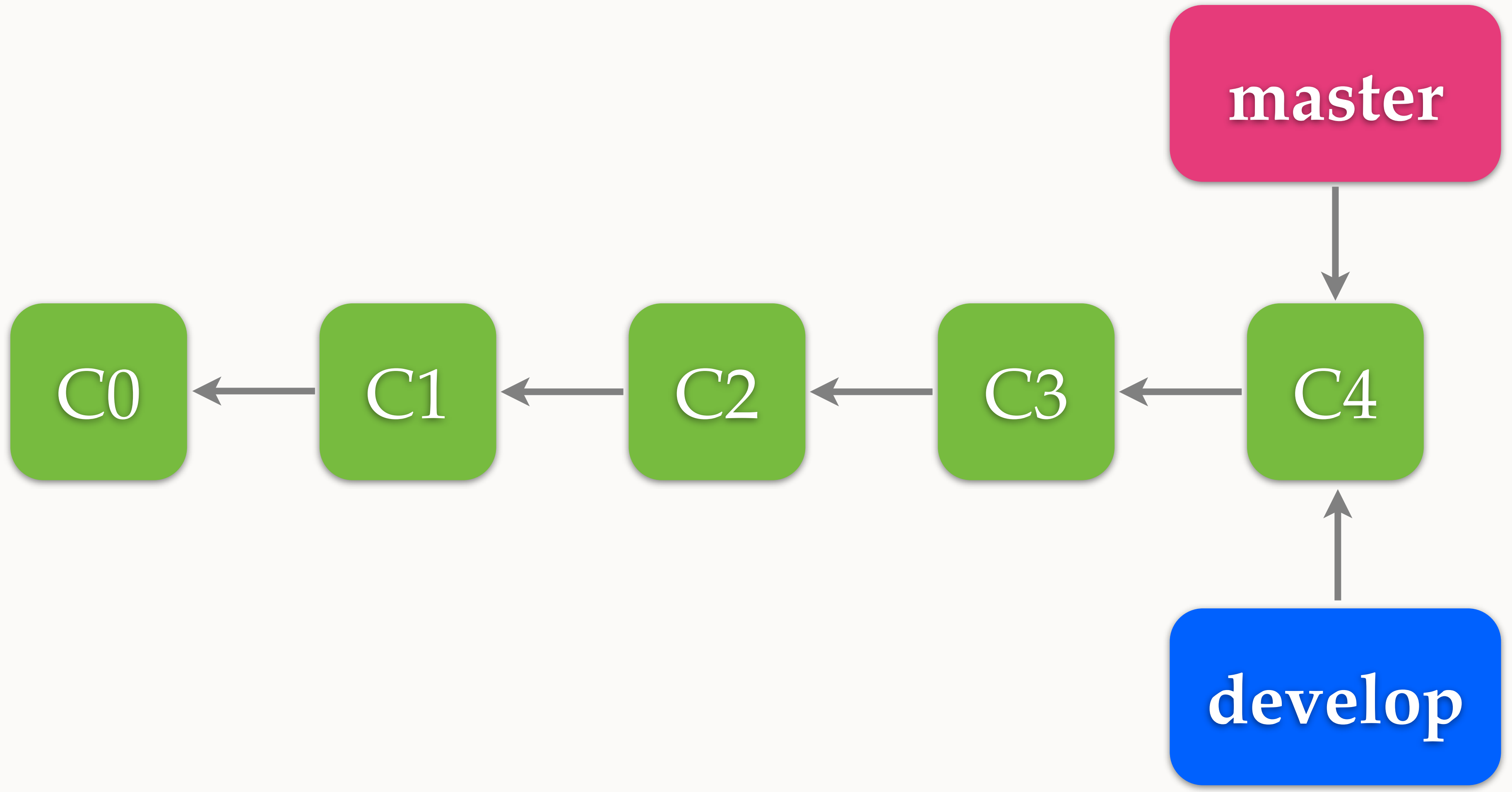
merge develop to master

FAST-FORWARD



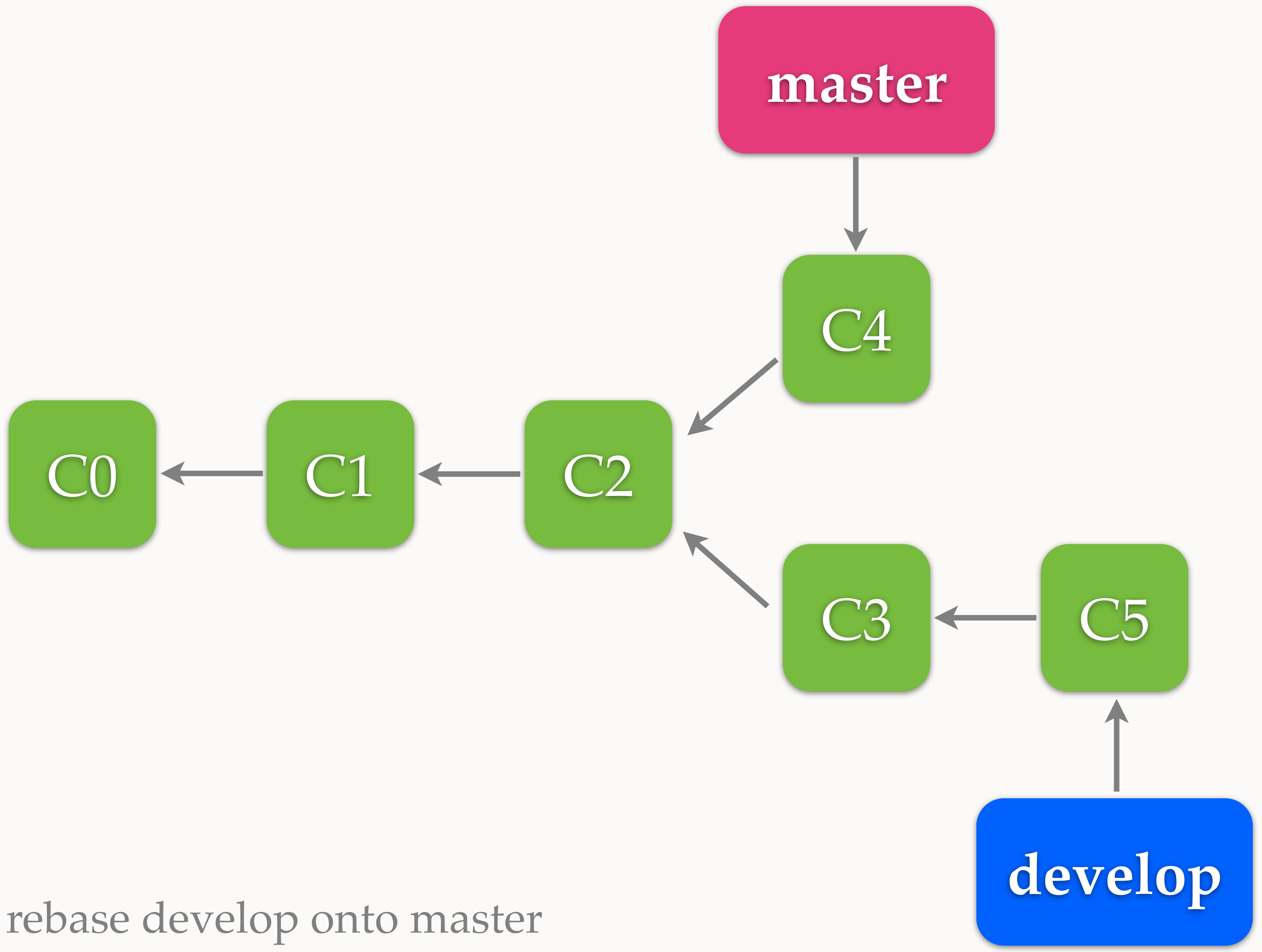
merge develop to master

FAST-FORWARD



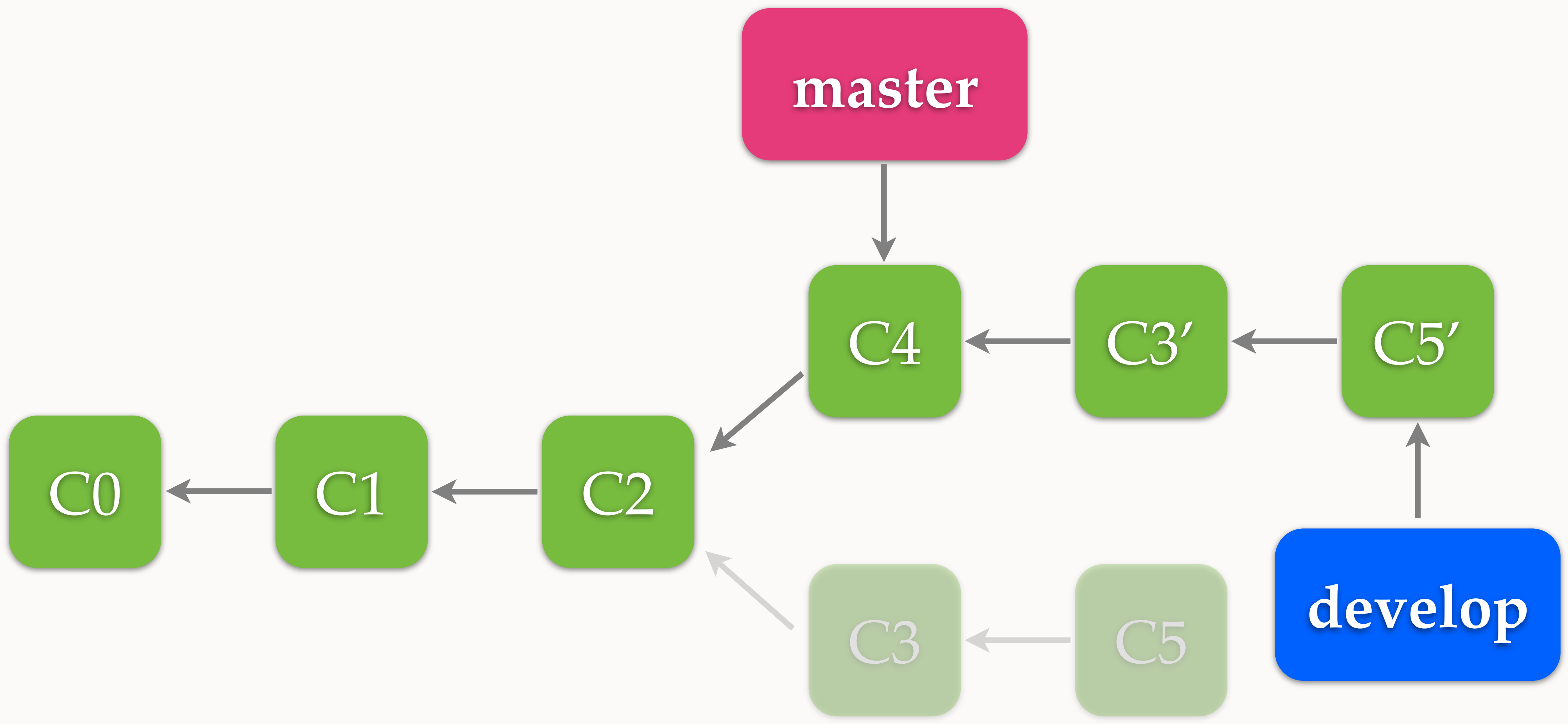
merge develop to master

REBASE



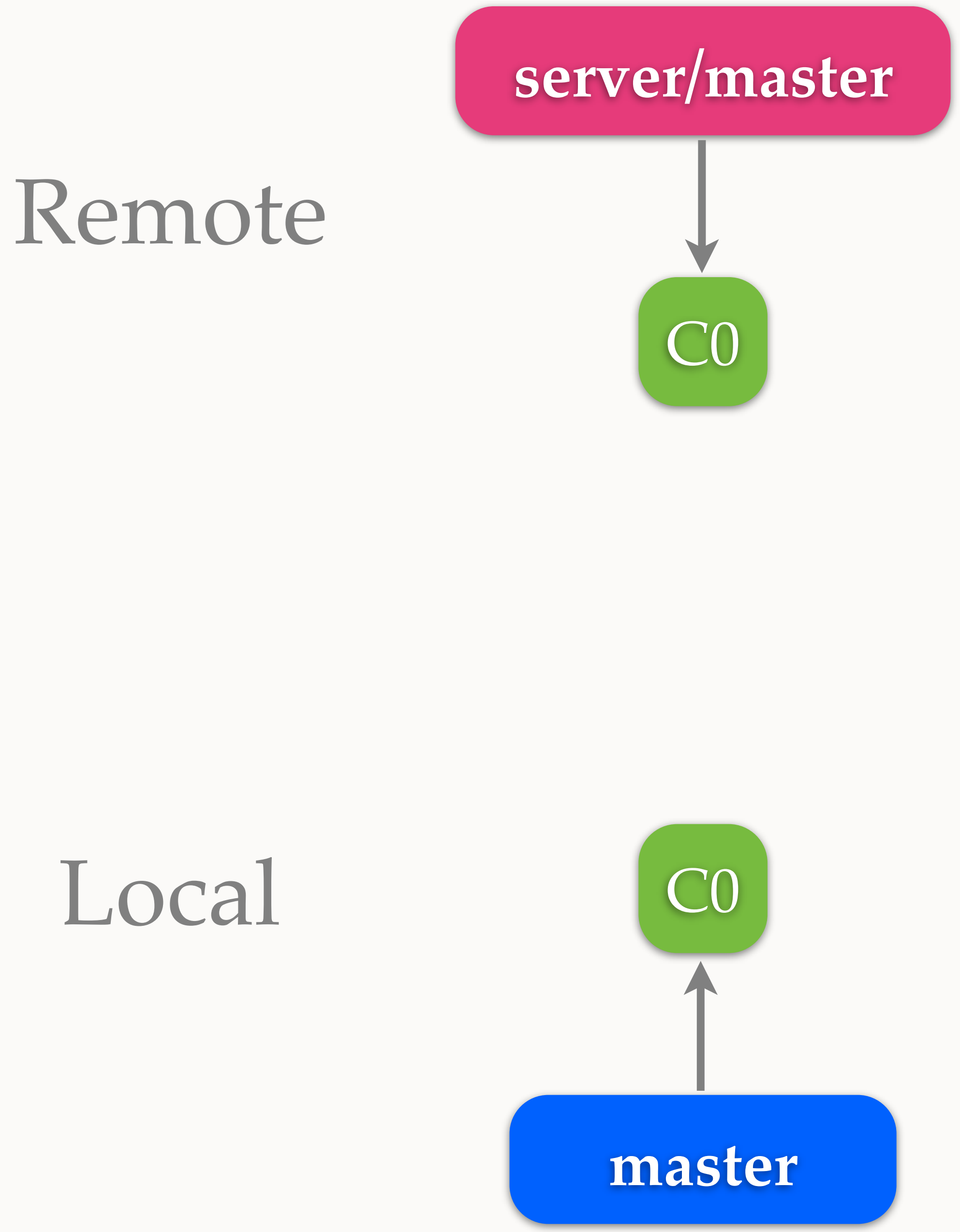
rebase develop onto master

REBASE

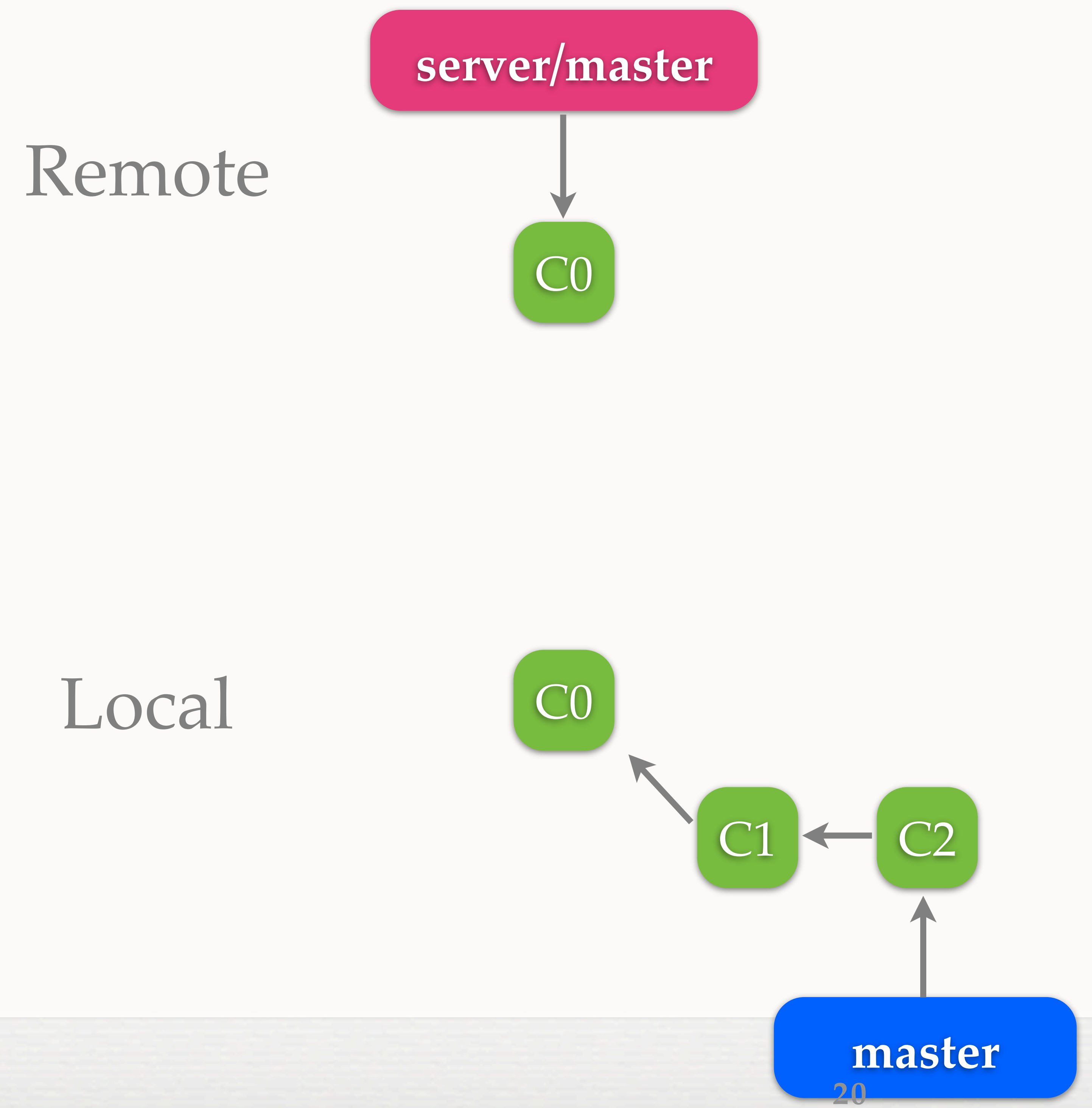


rebase develop onto master

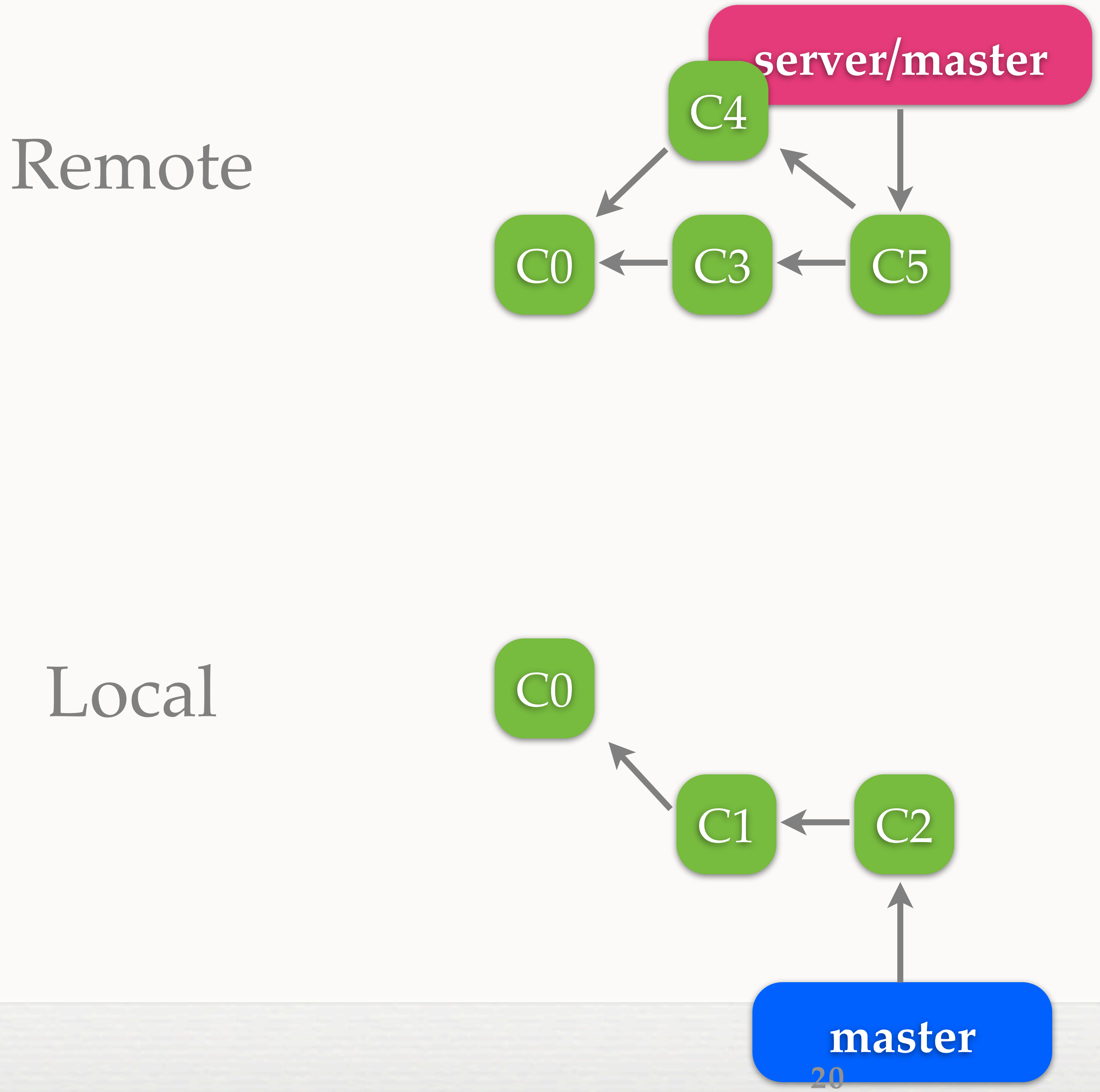
REBASE PUBLISHED COMMITS



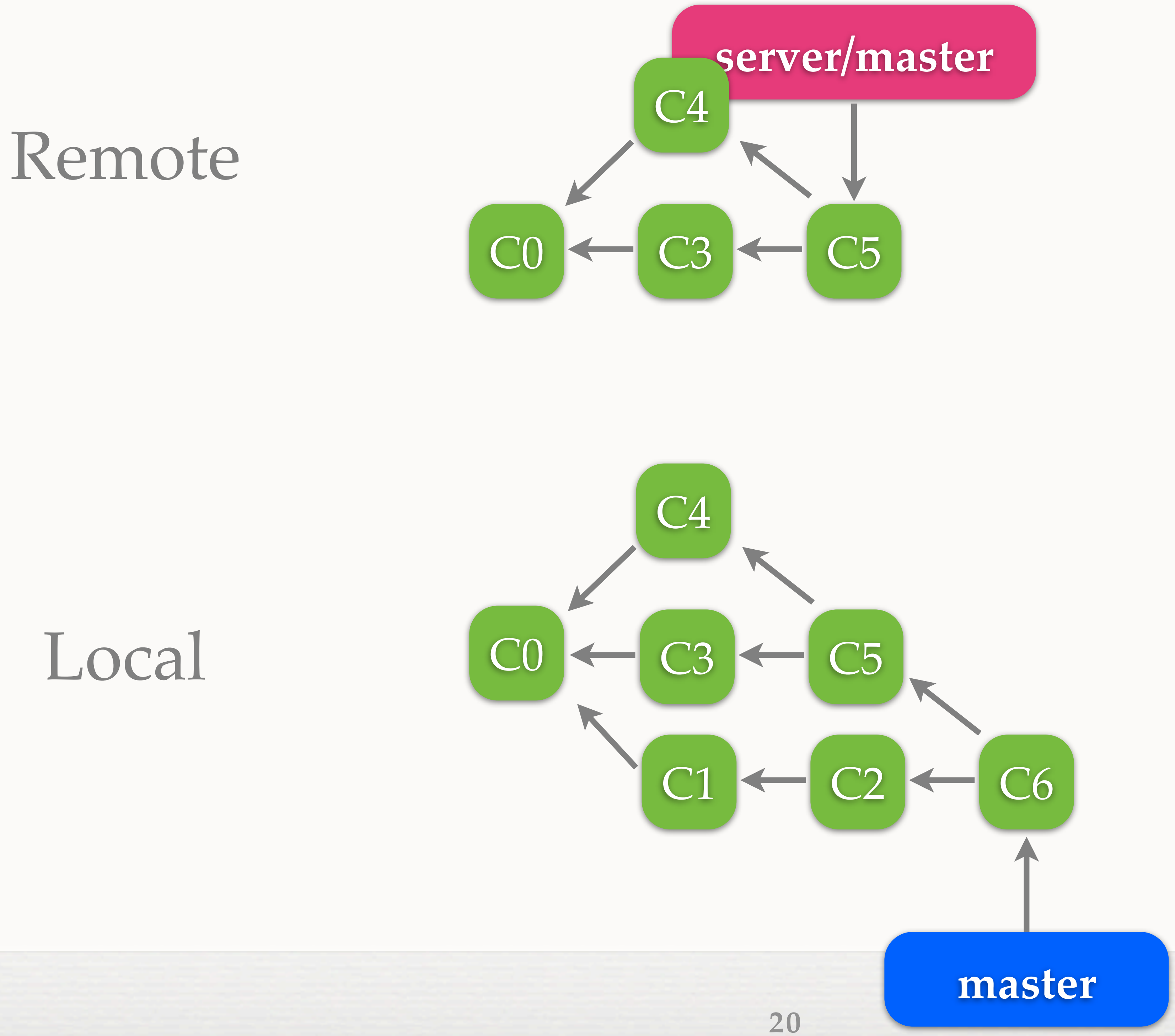
REBASE PUBLISHED COMMITS



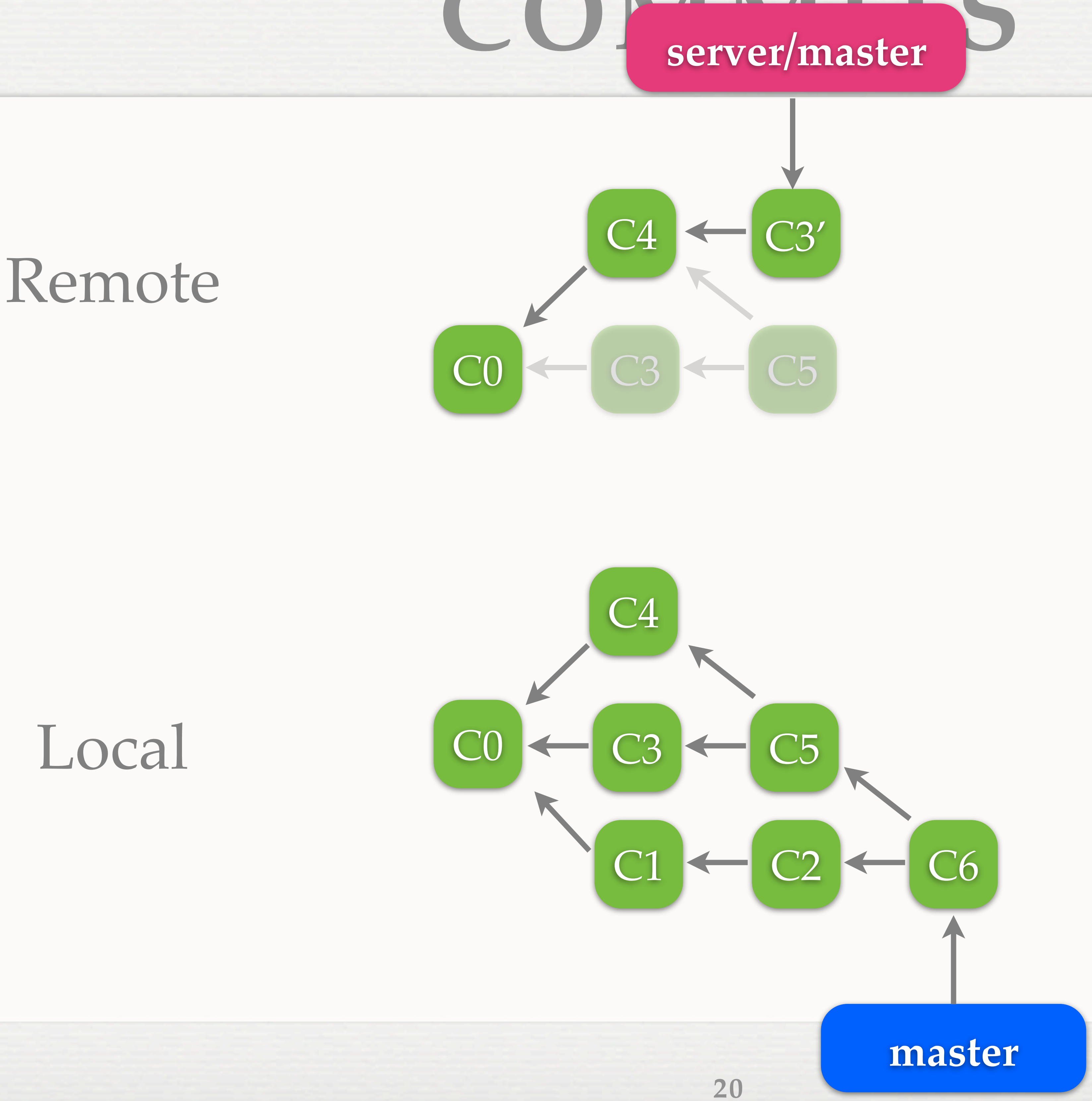
REBASE PUBLISHED COMMITS



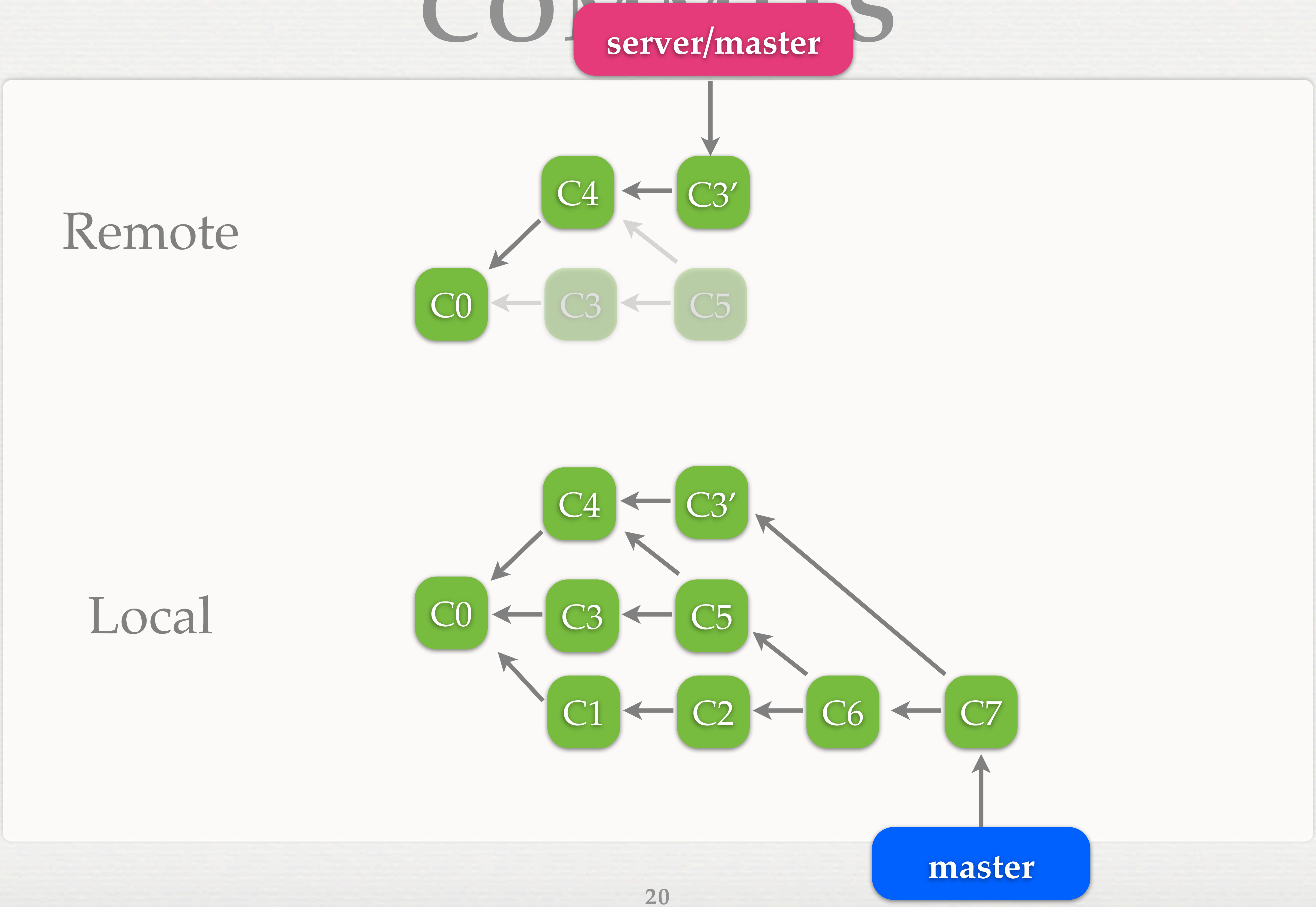
REBASE PUBLISHED COMMITS



REBASE PUBLISHED COMMITS



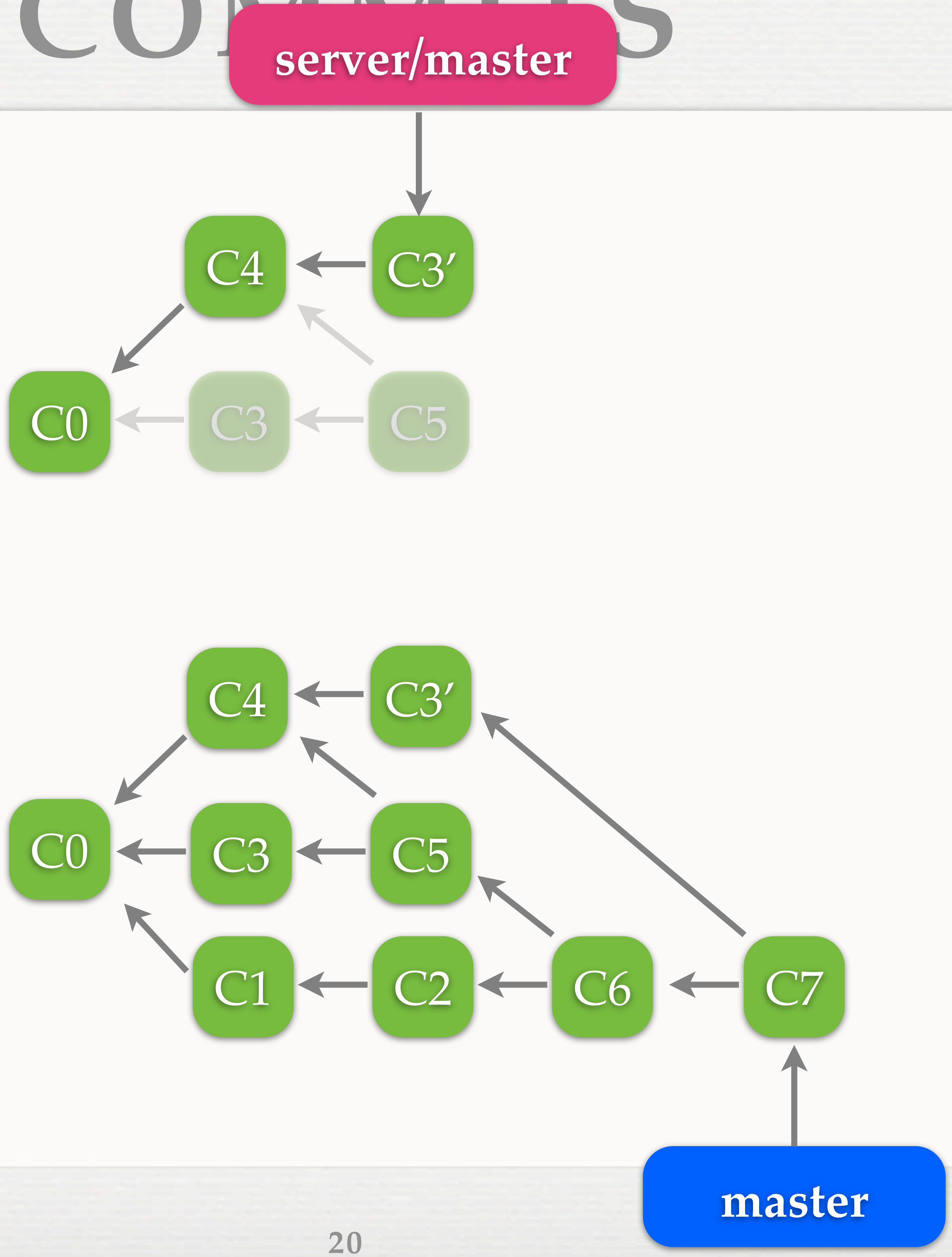
REBASE PUBLISHED COMMITS



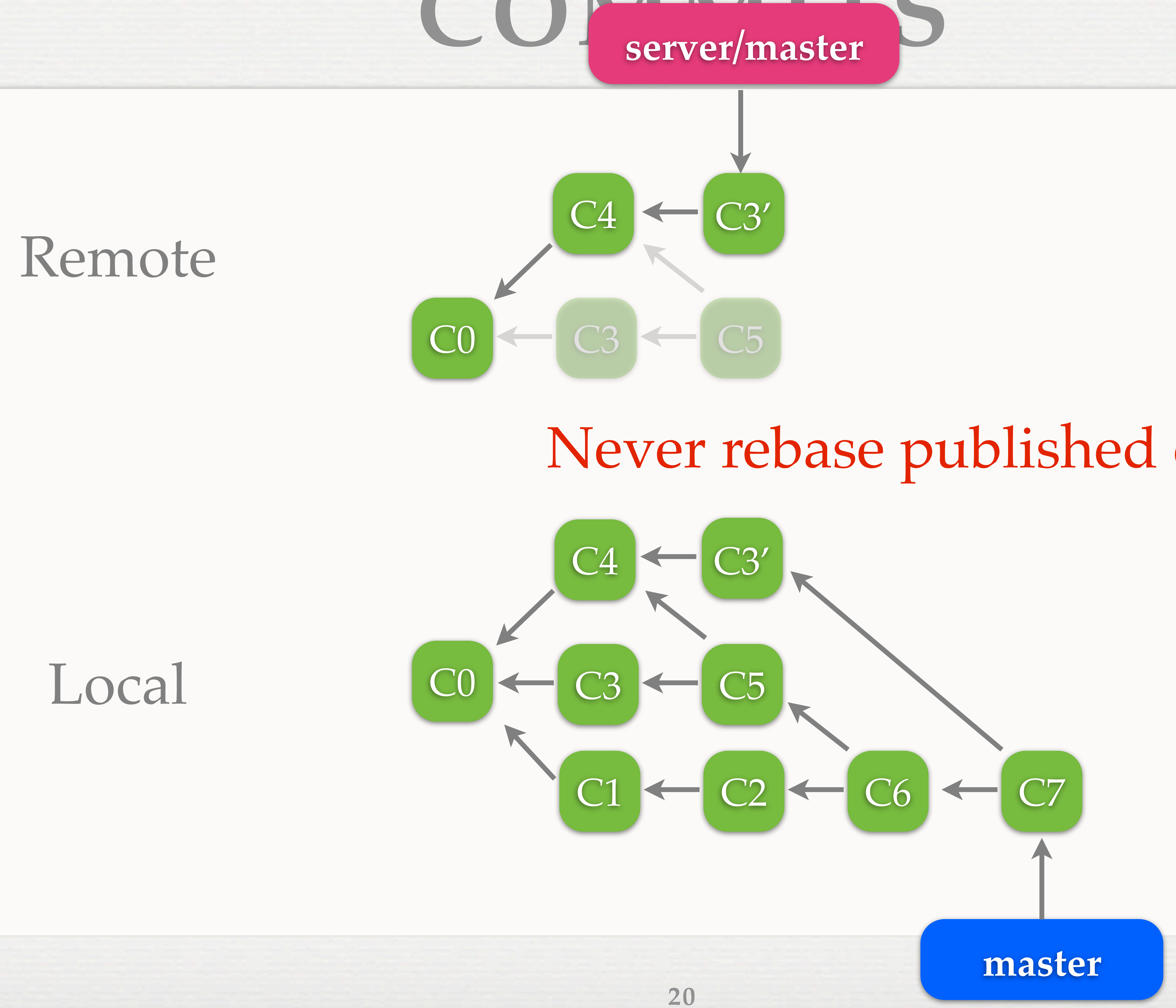
REBASE PUBLISHED COMMITS

Remote

Local

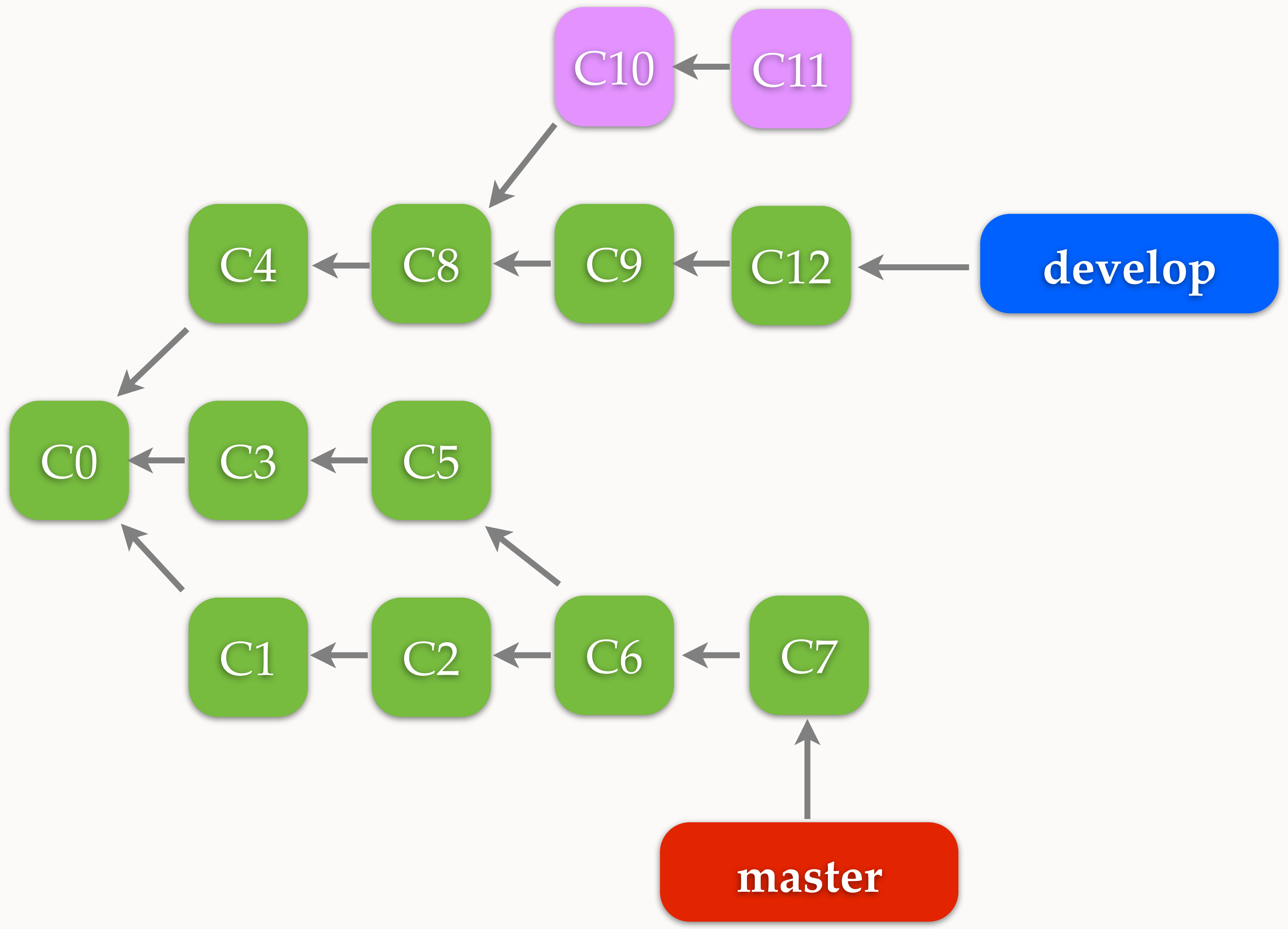


REBASE PUBLISHED COMMITS

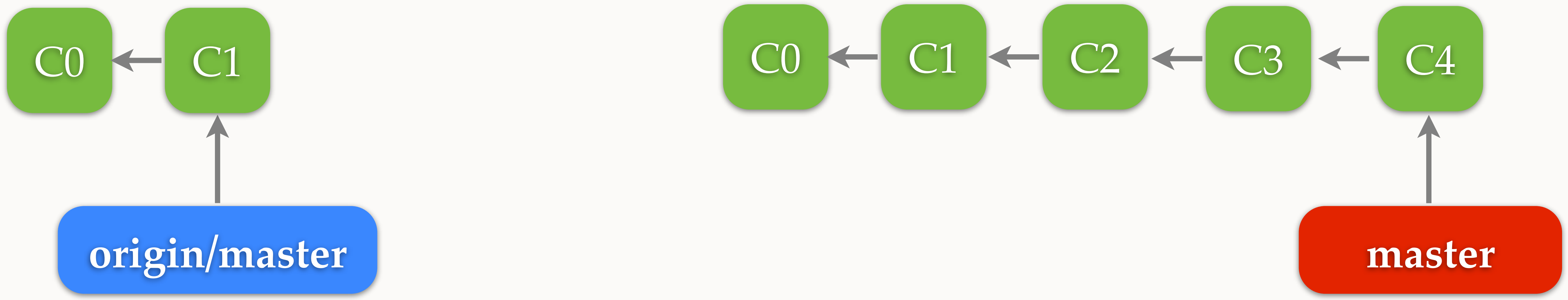


Never rebase published commits

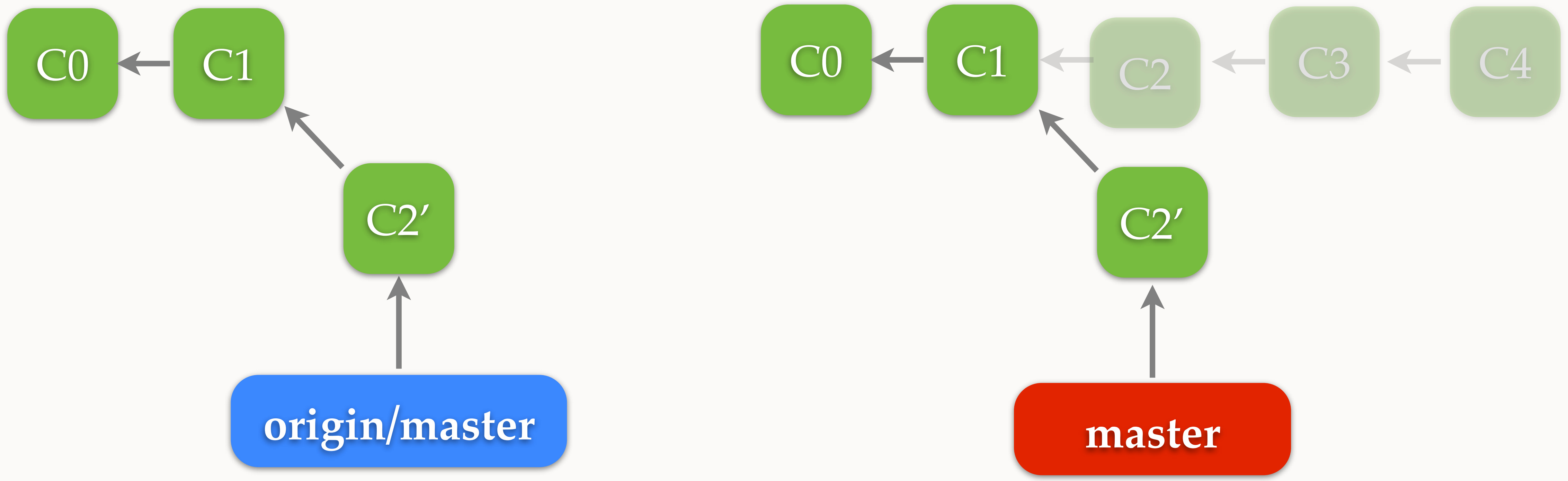
DANGLING COMMITS



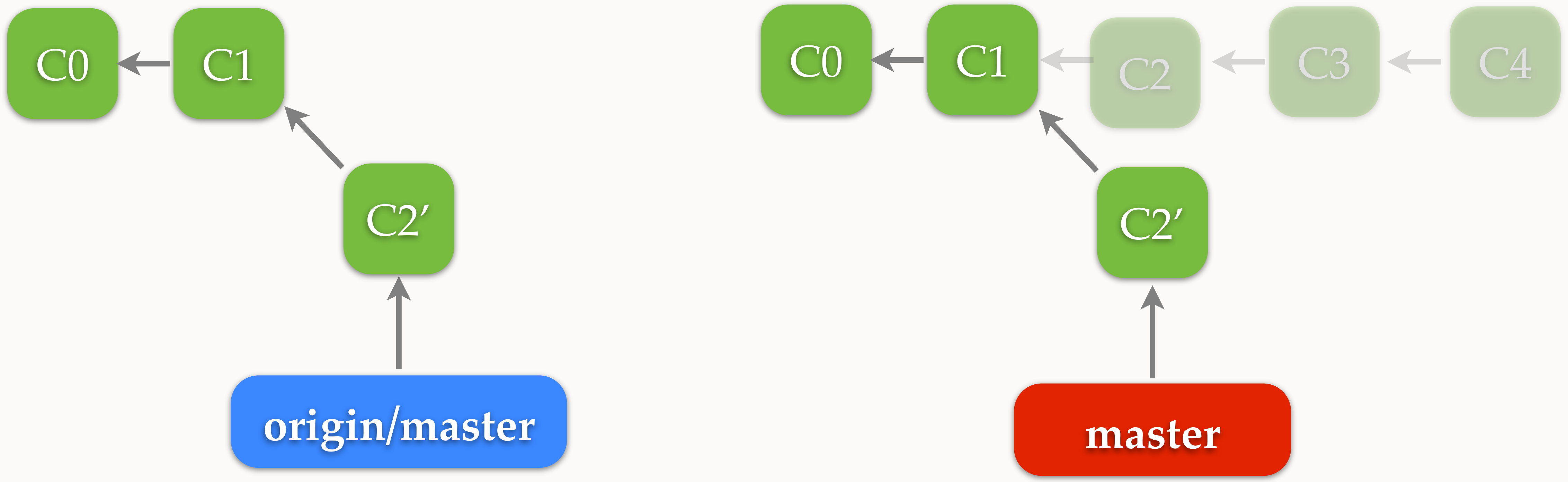
SQUASH



SQUASH

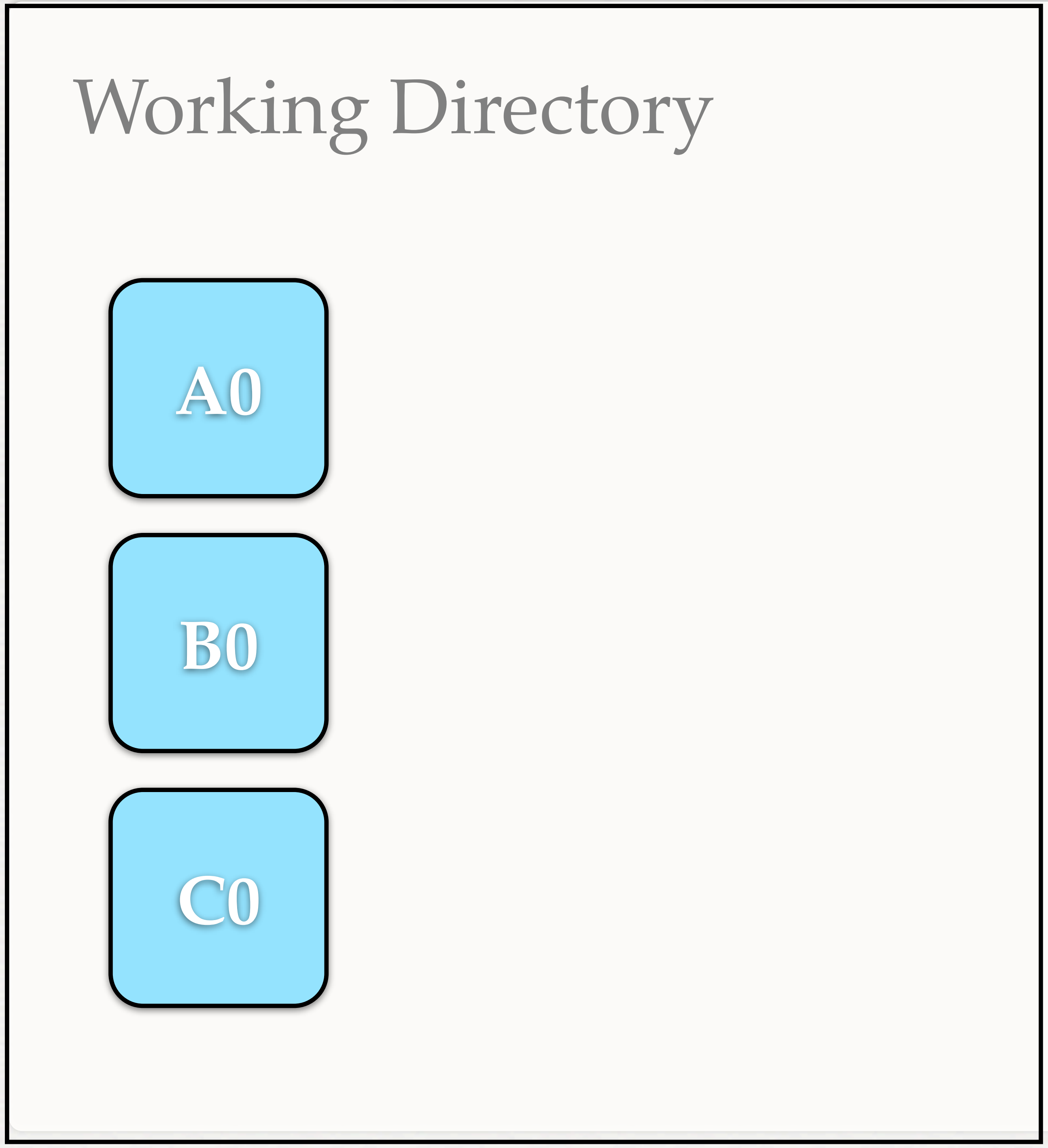


SQUASH

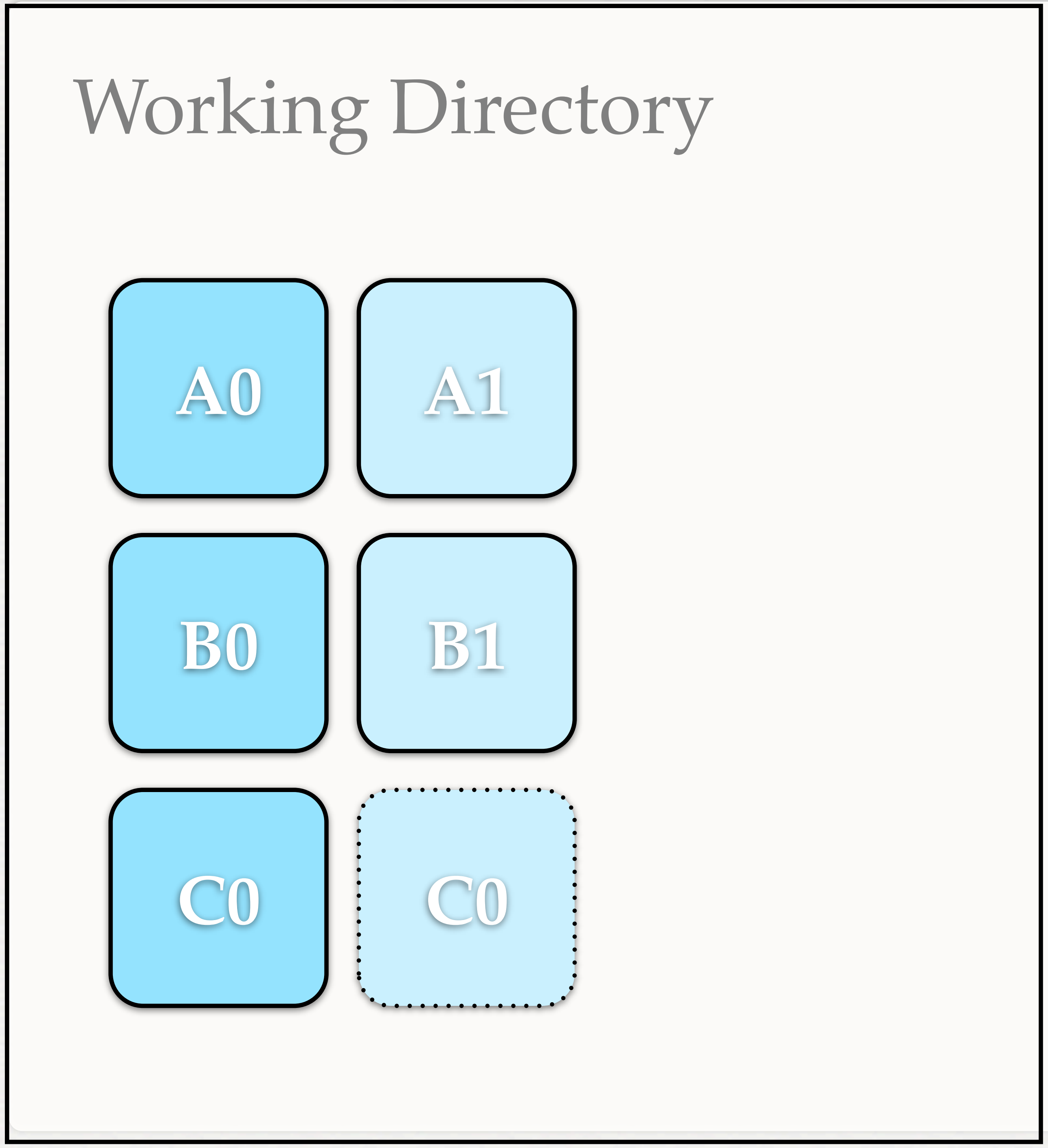


Don’t squash published commits

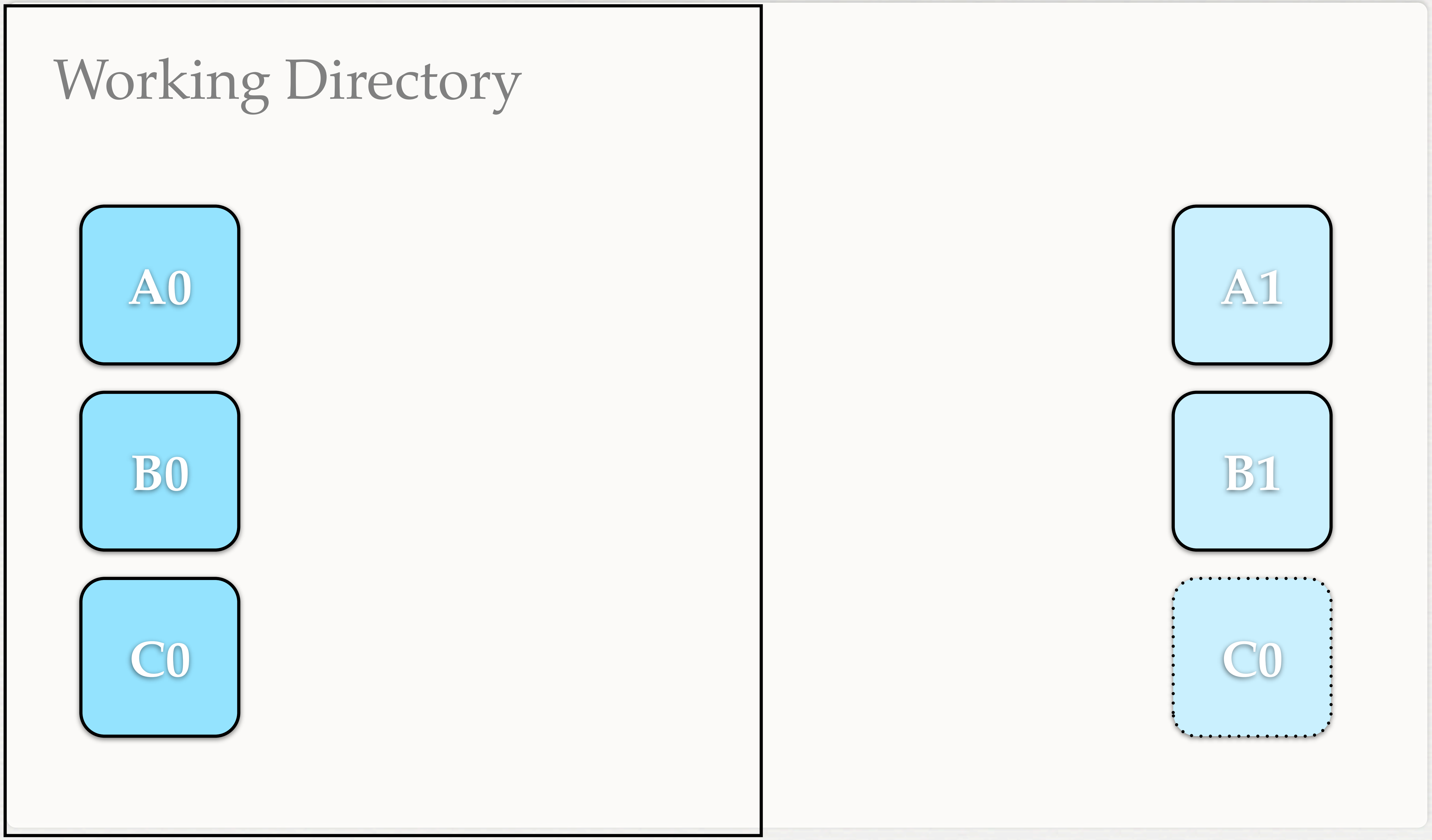
STASH



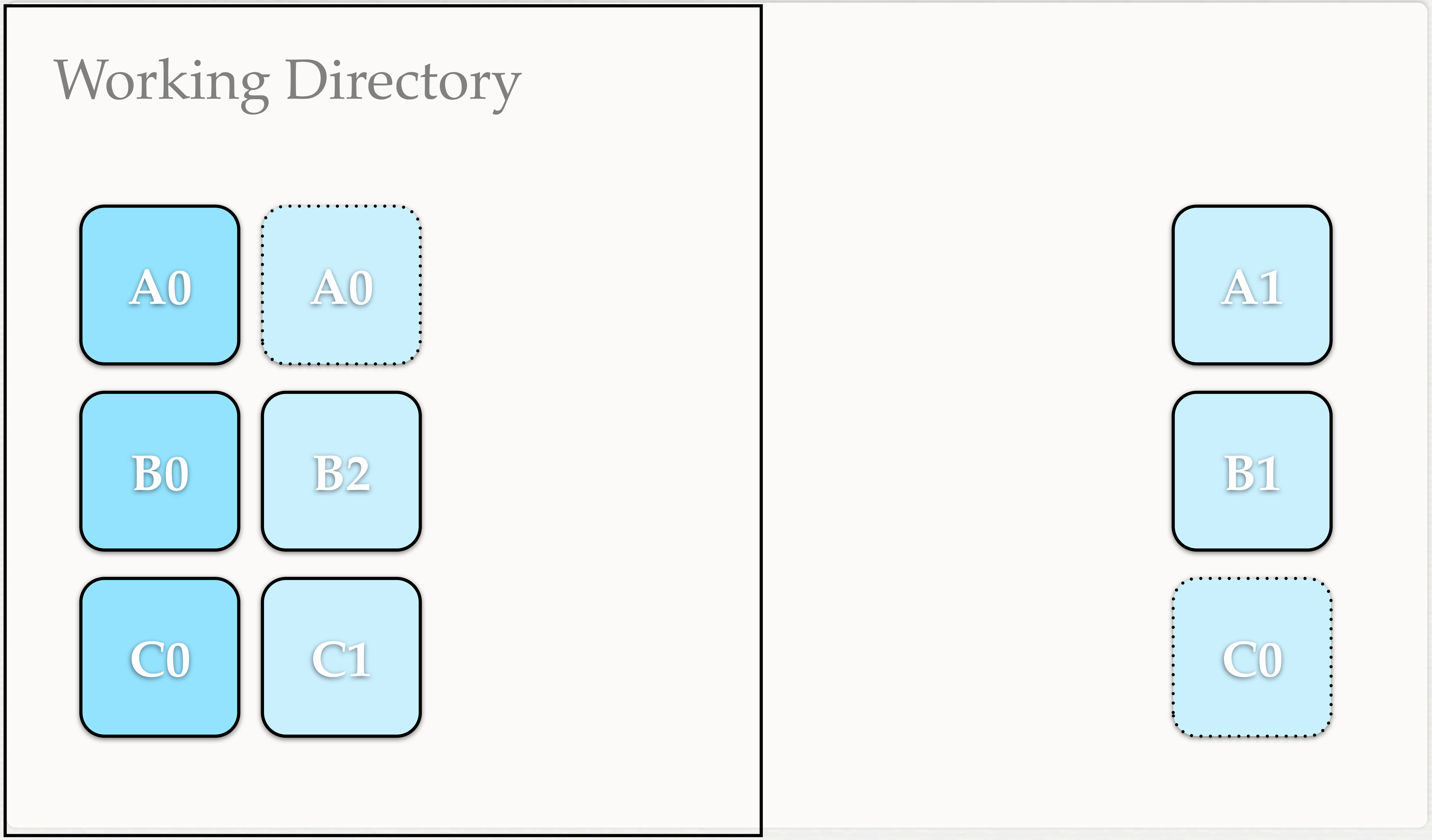
STASH



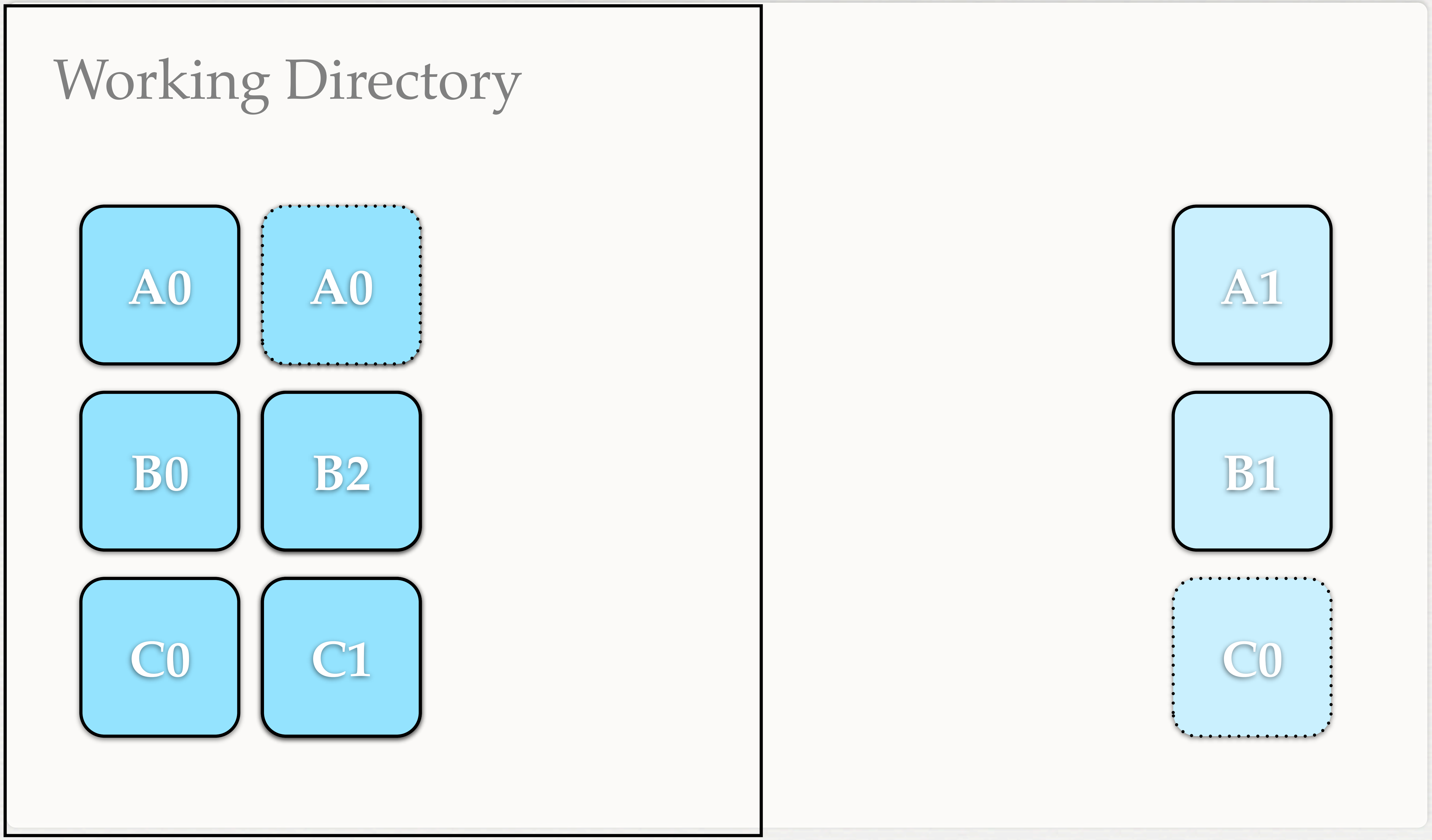
STASH



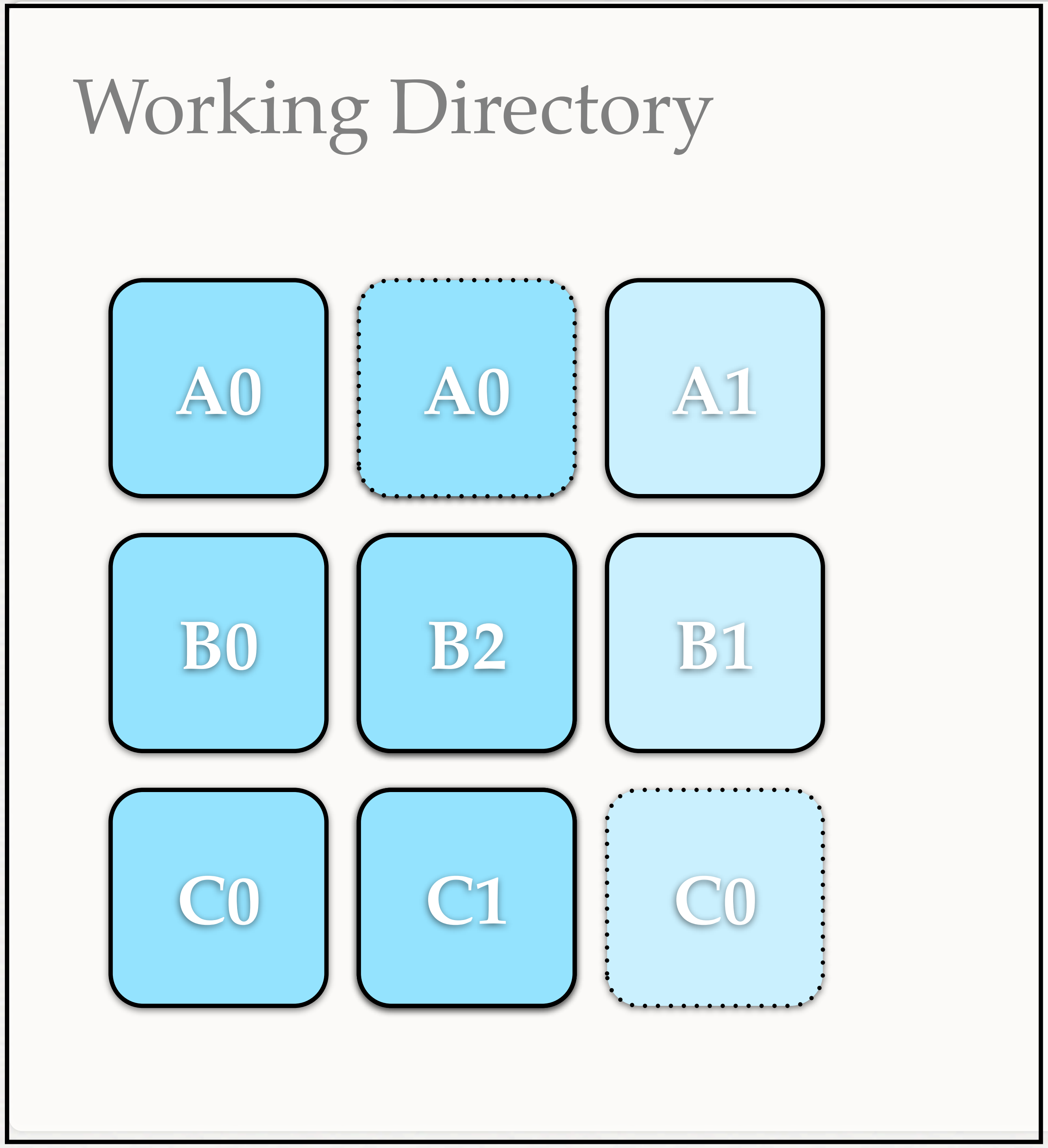
STASH



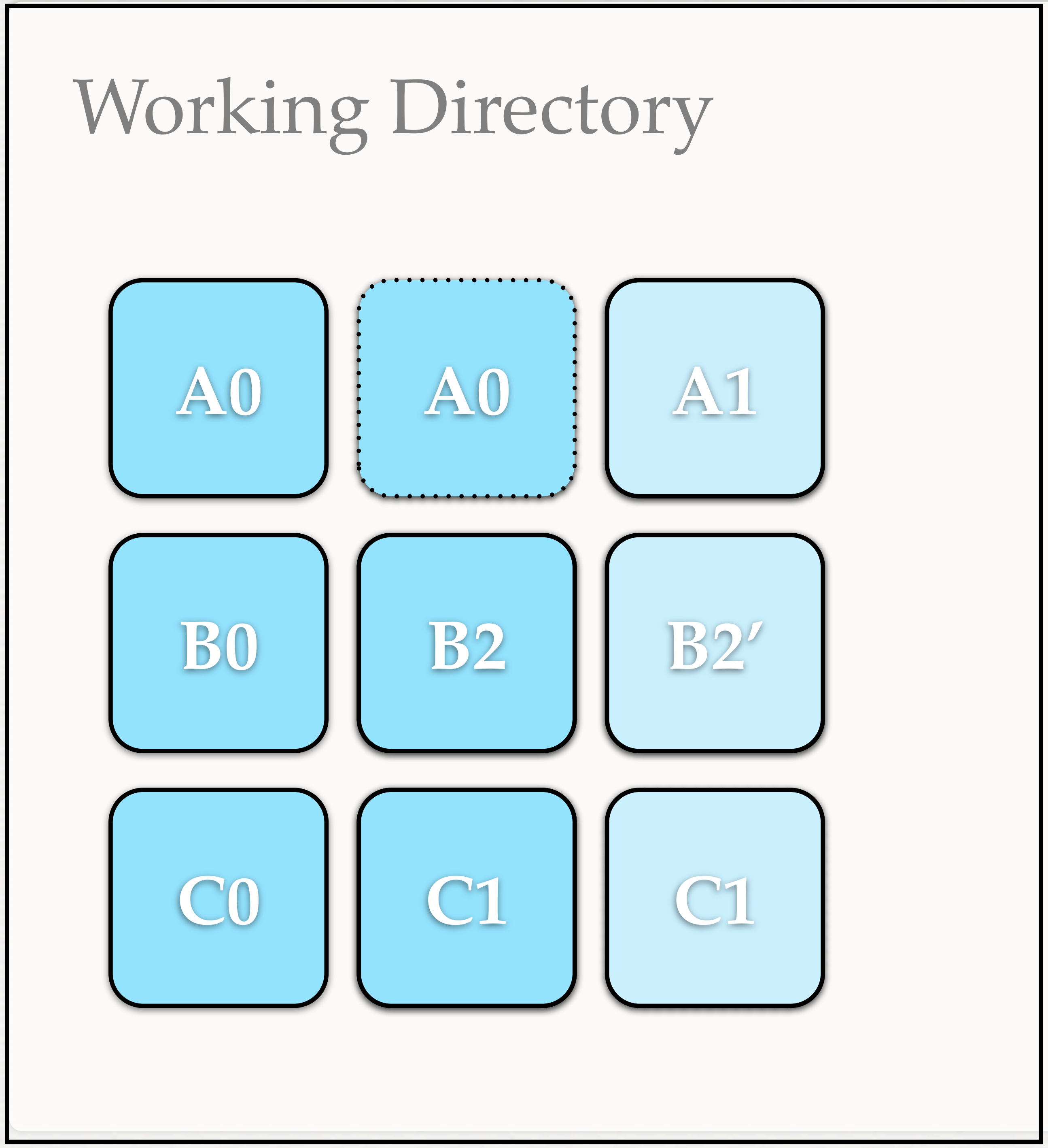
STASH



STASH



STASH



BEFORE USING GIT

- $ git config user.name “YOUR NAME”
- $ git config user.email “YOUR EMAIL”
- $ git config http.sslVerify false
  - for our server with a self-signed certificate

DEMO

- git add
- git branch
- git checkout
- git clone
- git commit
- git diff
- git fetch
- git init
- git log
- git merge
- git pull
- git push
- git rebase
- git remote
- git stash
- git status

REFERENCES

- http://git-scm.com/book
- http://git-scm.com/docs