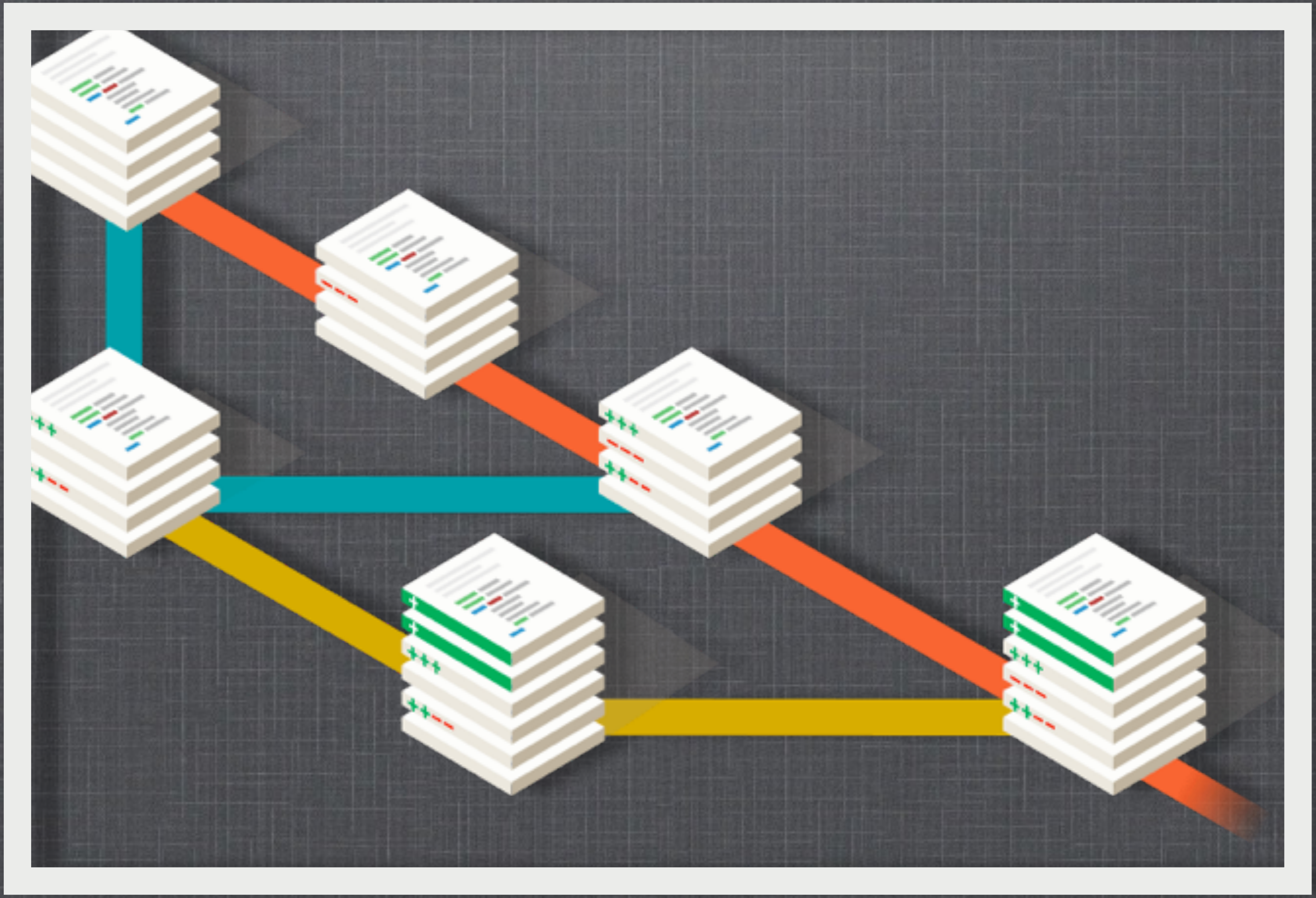
GIT

Ming-Hsien Tsai



SDM 2016

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

WHAT IS GIT

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

BEFORE VERSION CONTROL

version 1

BEFORE VERSION CONTROL

version 1

version 2

BEFORE VERSION CONTROL

version 1

version 3

version 2

BEFORE VERSION CONTROL

version 1

version 3

version 2

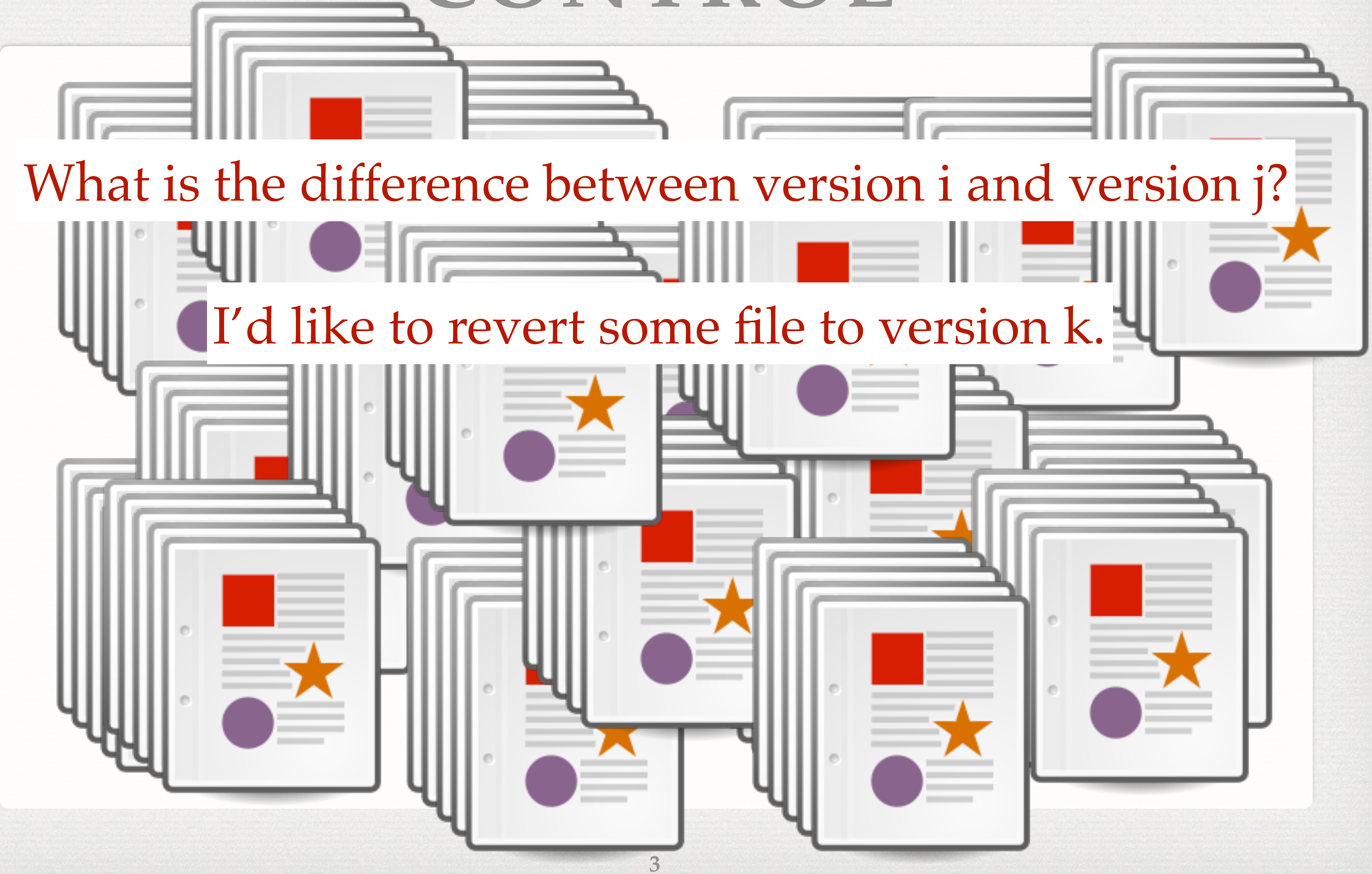
version 4

BEFORE VERSION CONTROL

BEFORE VERSION CONTROL

What is the difference between version i and version j?

BEFORE VERSION CONTROL



What is the difference between version i and version j?

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

BEFORE VERSION CONTROL

What is the difference between version i and version j?

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

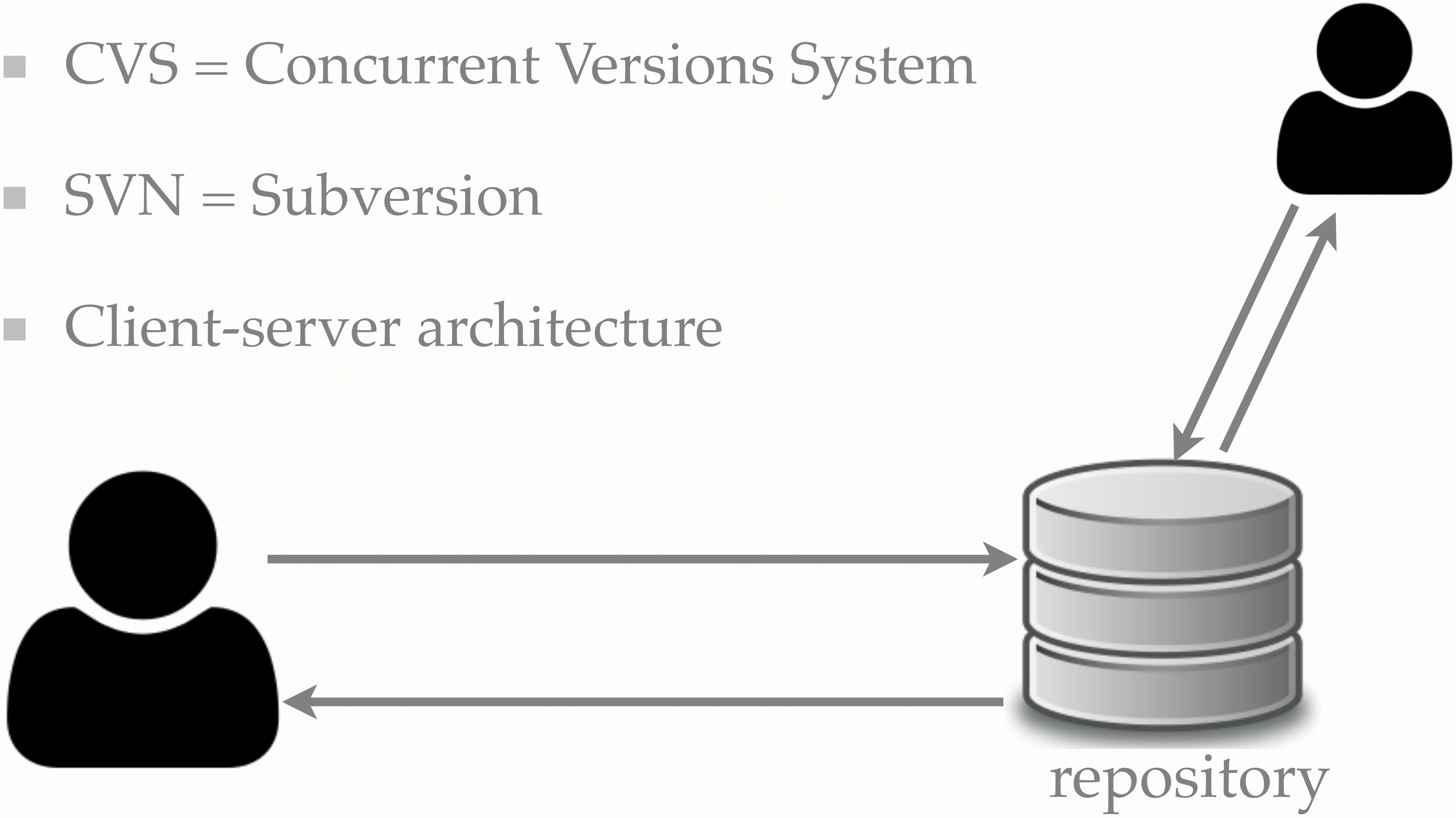
Need a better management.

SCCS (1972), RCS (1982)

- SCCS = Source Code Control System
- RCS = Revision Control System
- Control source code and other text files
- Local files only
- RCS: each version-controlled file has its own repository

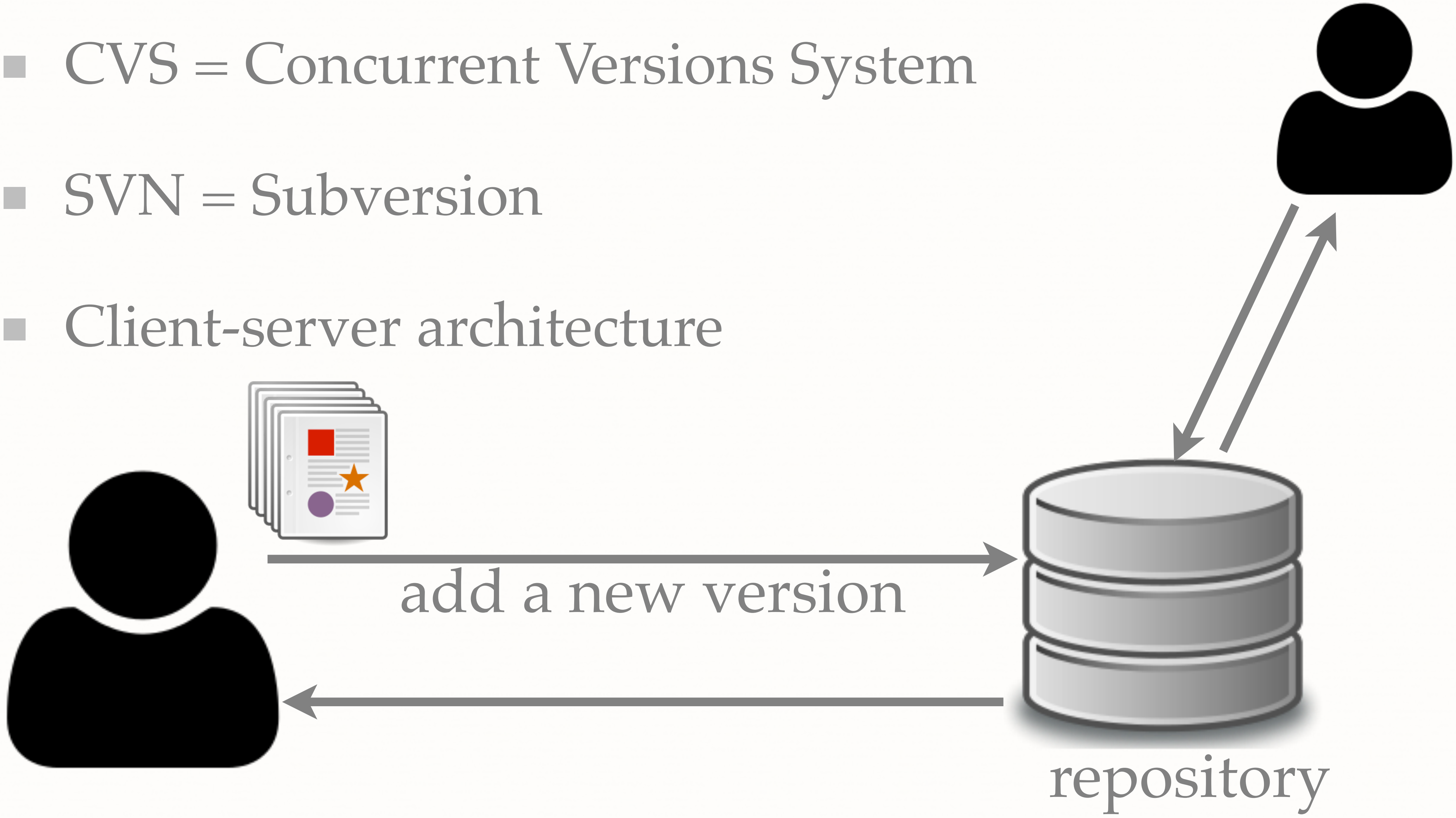
CVS (1986), SVN (2000)

- CVS = Concurrent Versions System
- SVN = Subversion
- Client-server architecture



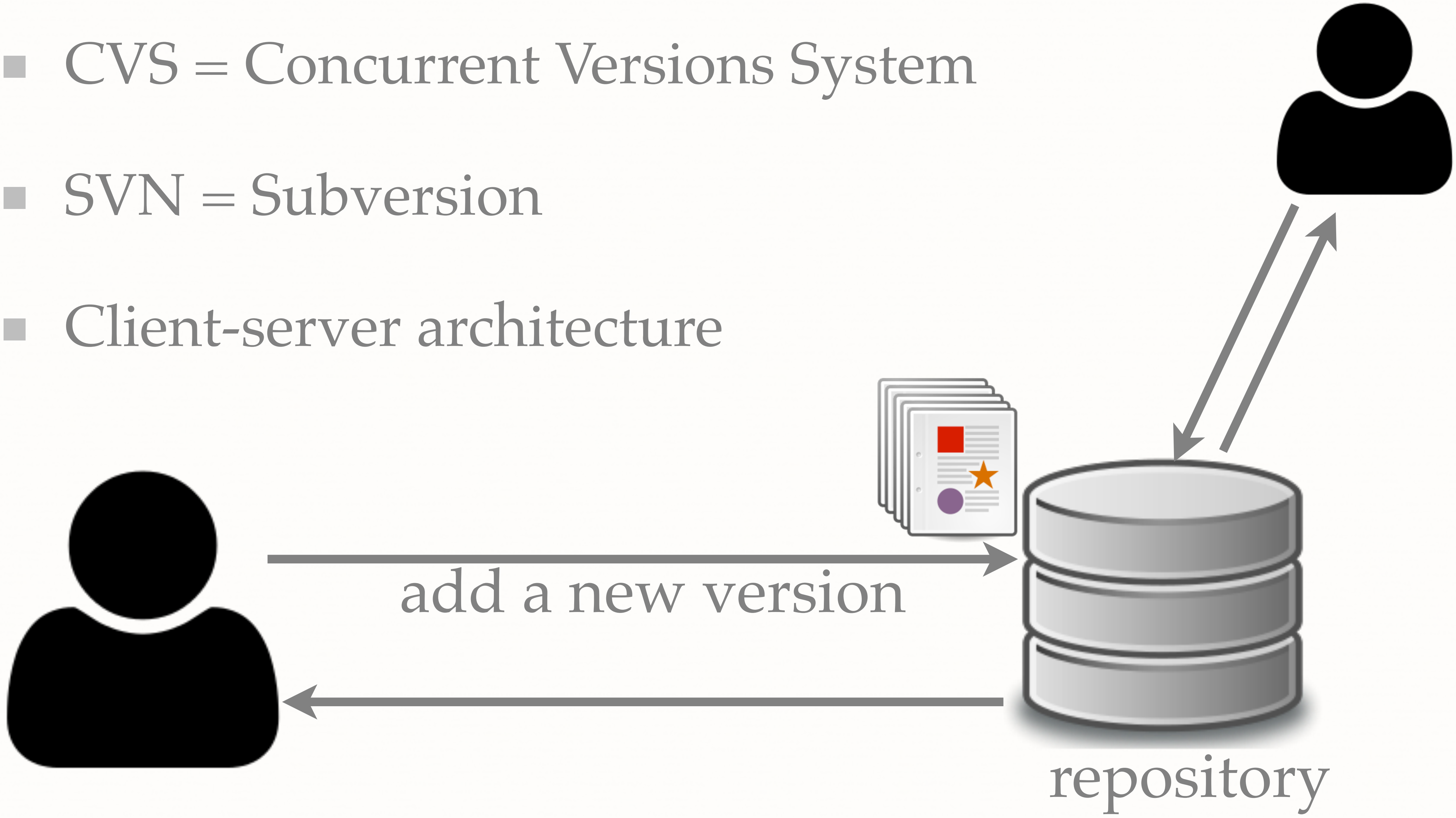
CVS (1986), SVN (2000)

- CVS = Concurrent Versions System
- SVN = Subversion
- Client-server architecture



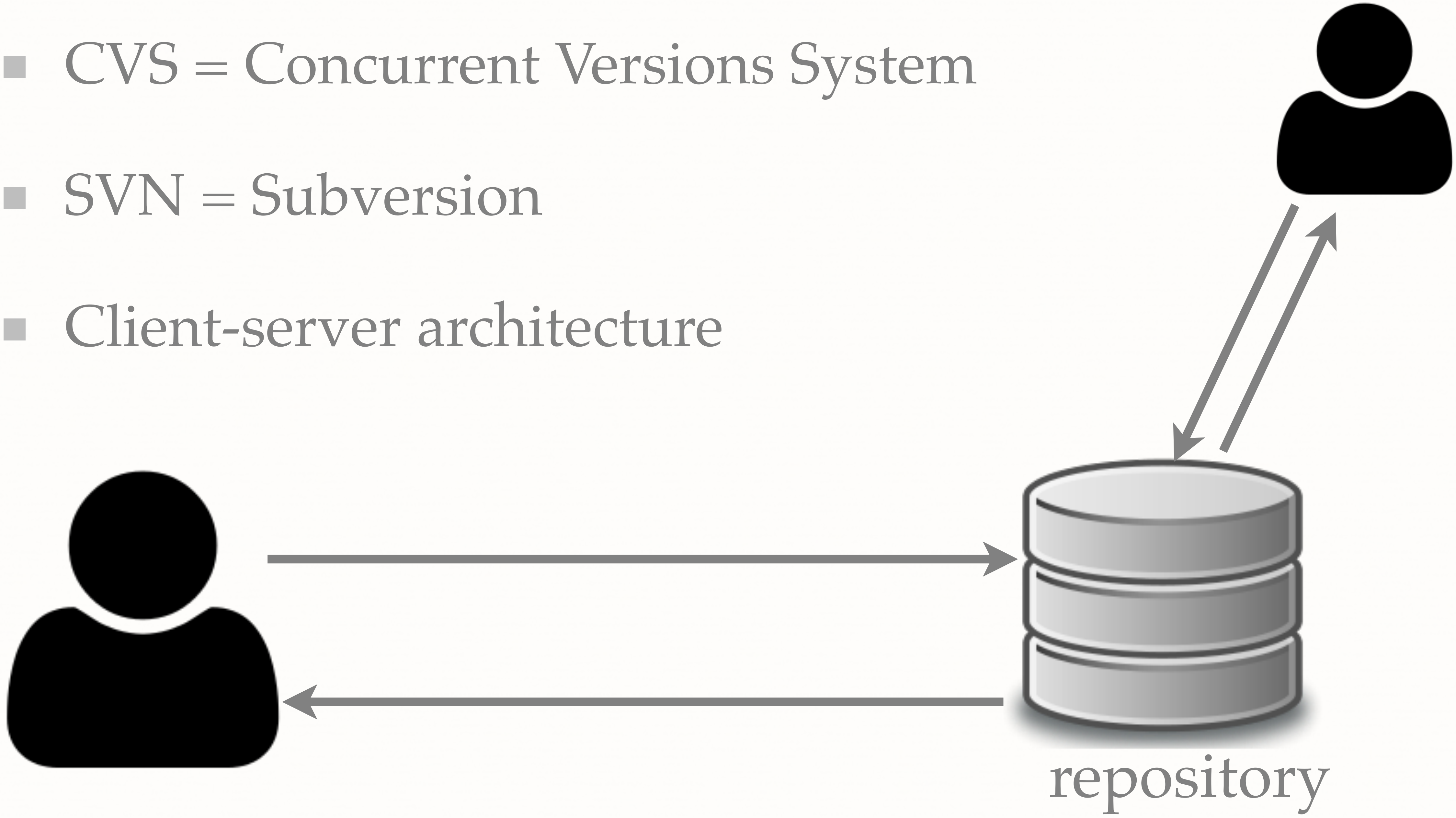
CVS (1986), SVN (2000)

- CVS = Concurrent Versions System
- SVN = Subversion
- Client-server architecture



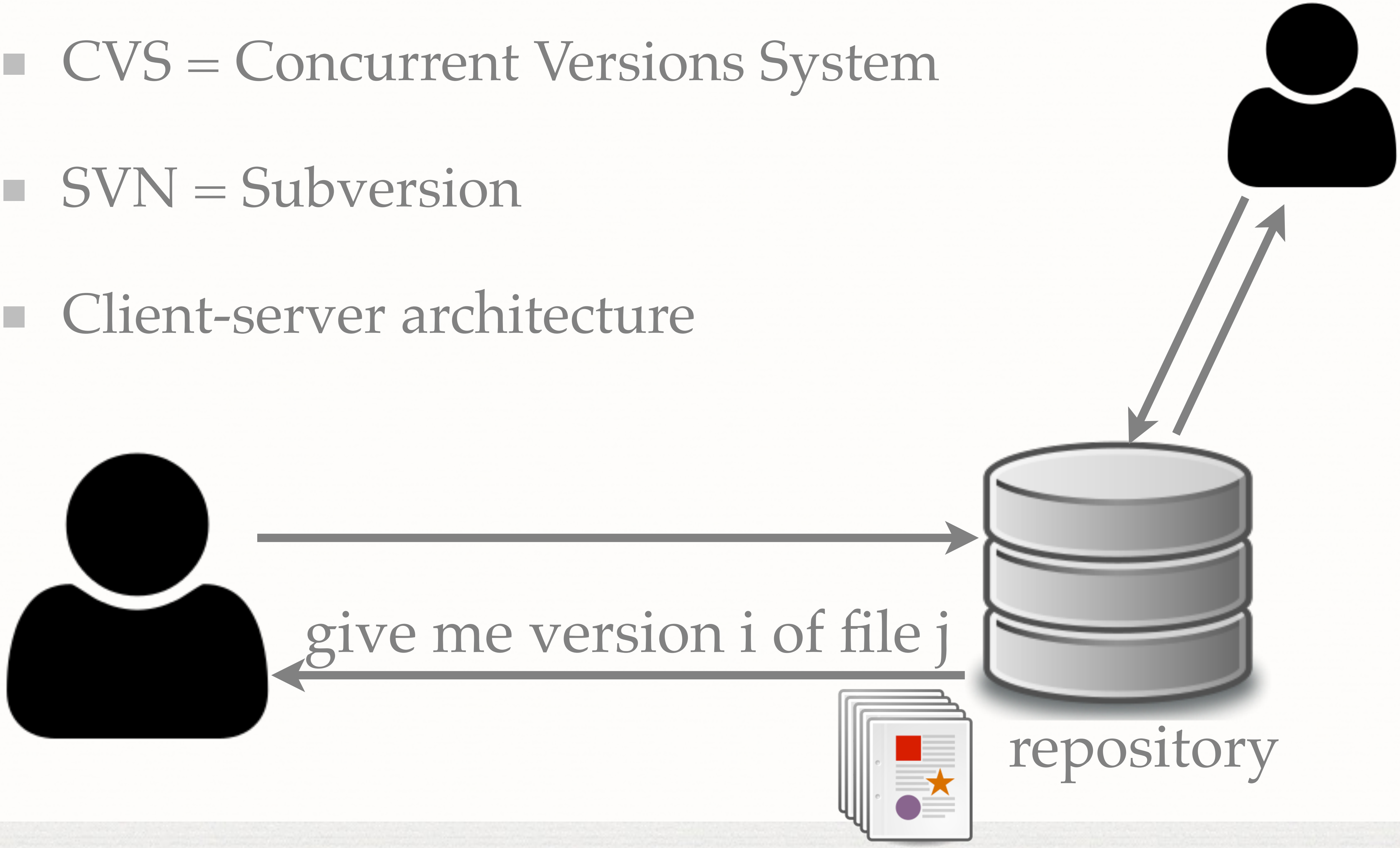
CVS (1986), SVN (2000)

- CVS = Concurrent Versions System
- SVN = Subversion
- Client-server architecture



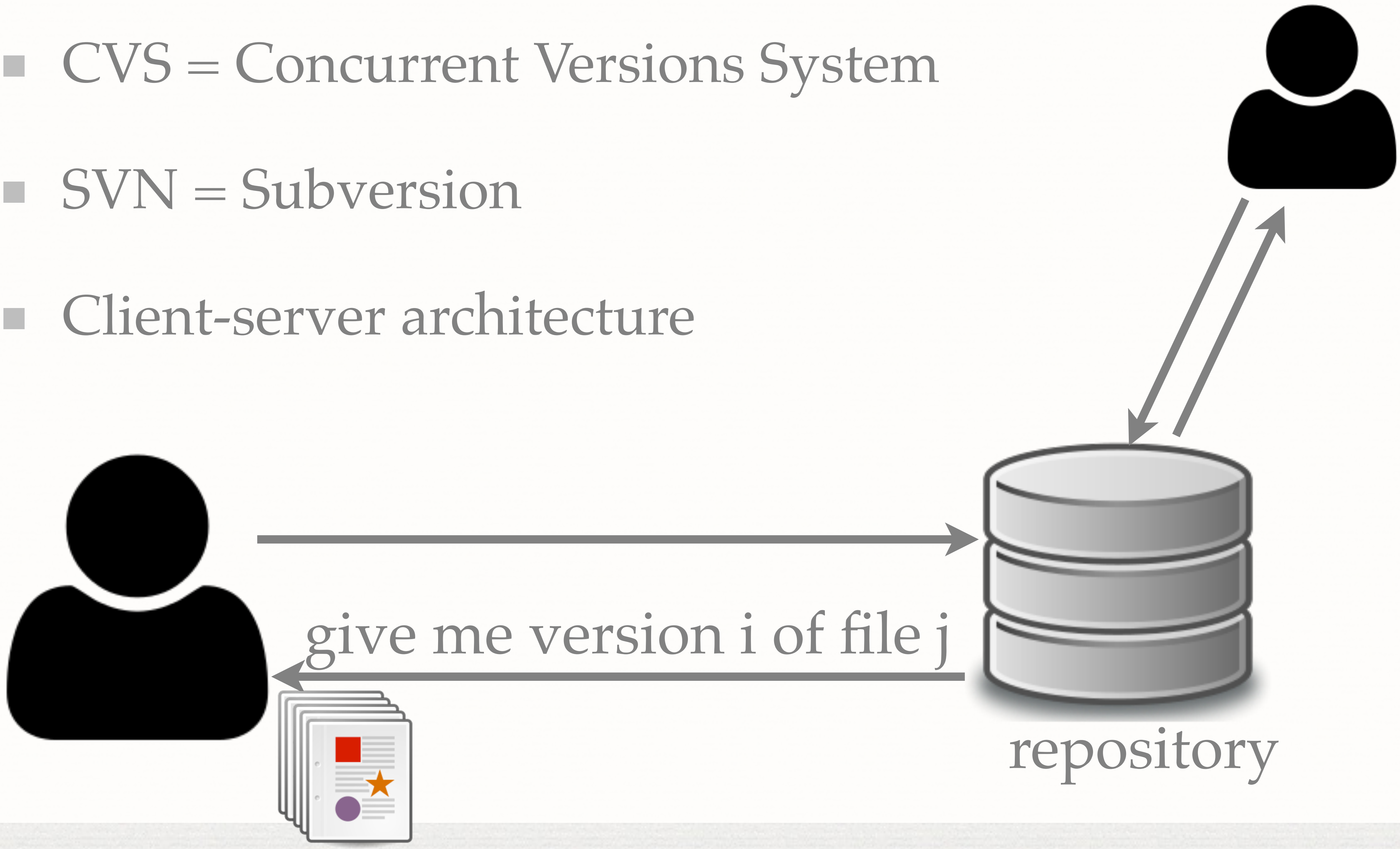
CVS (1986), SVN (2000)

- CVS = Concurrent Versions System
- SVN = Subversion
- Client-server architecture



CVS (1986), SVN (2000)

- CVS = Concurrent Versions System
- SVN = Subversion
- Client-server architecture

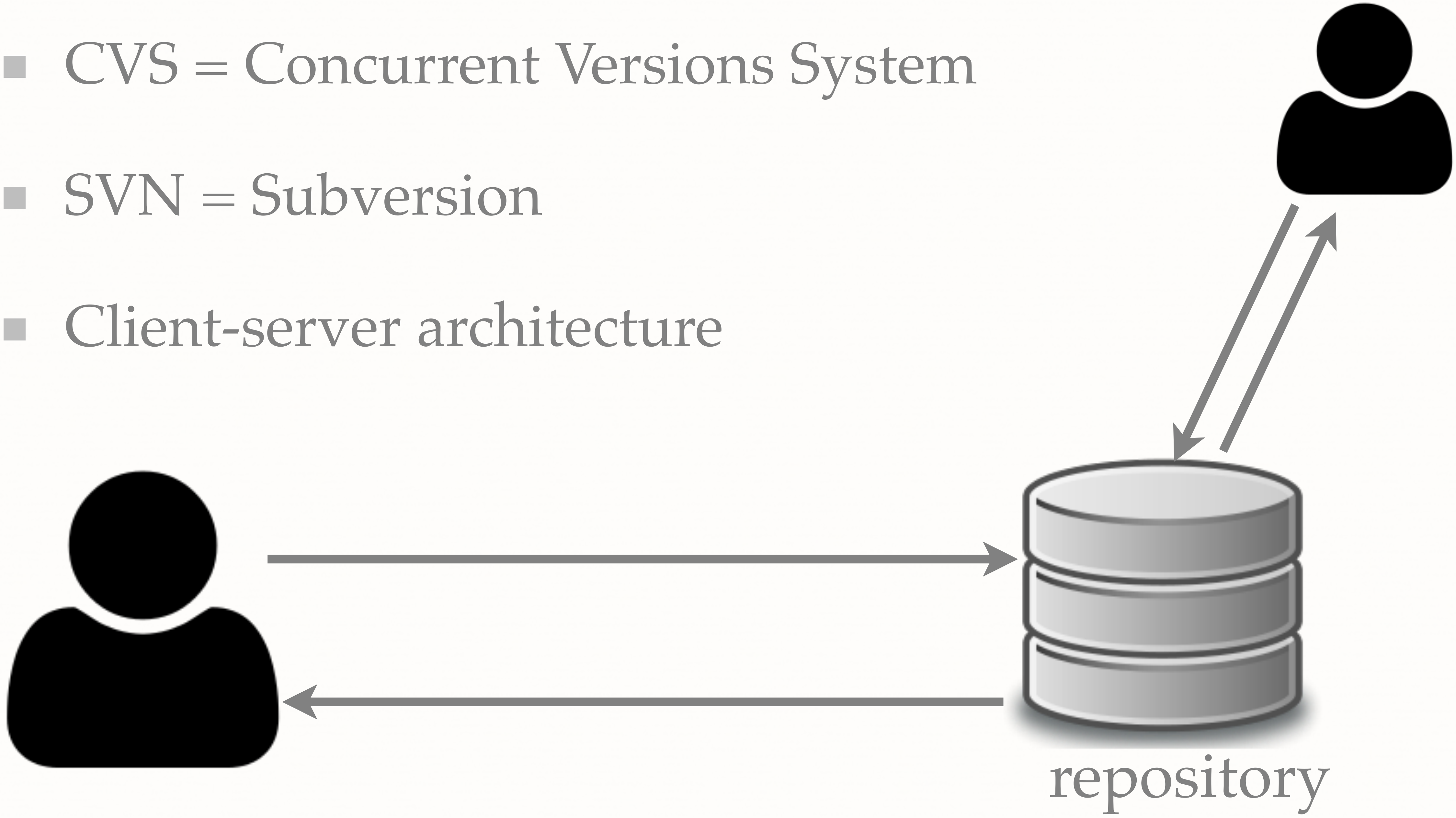


give me version i of file j

repository

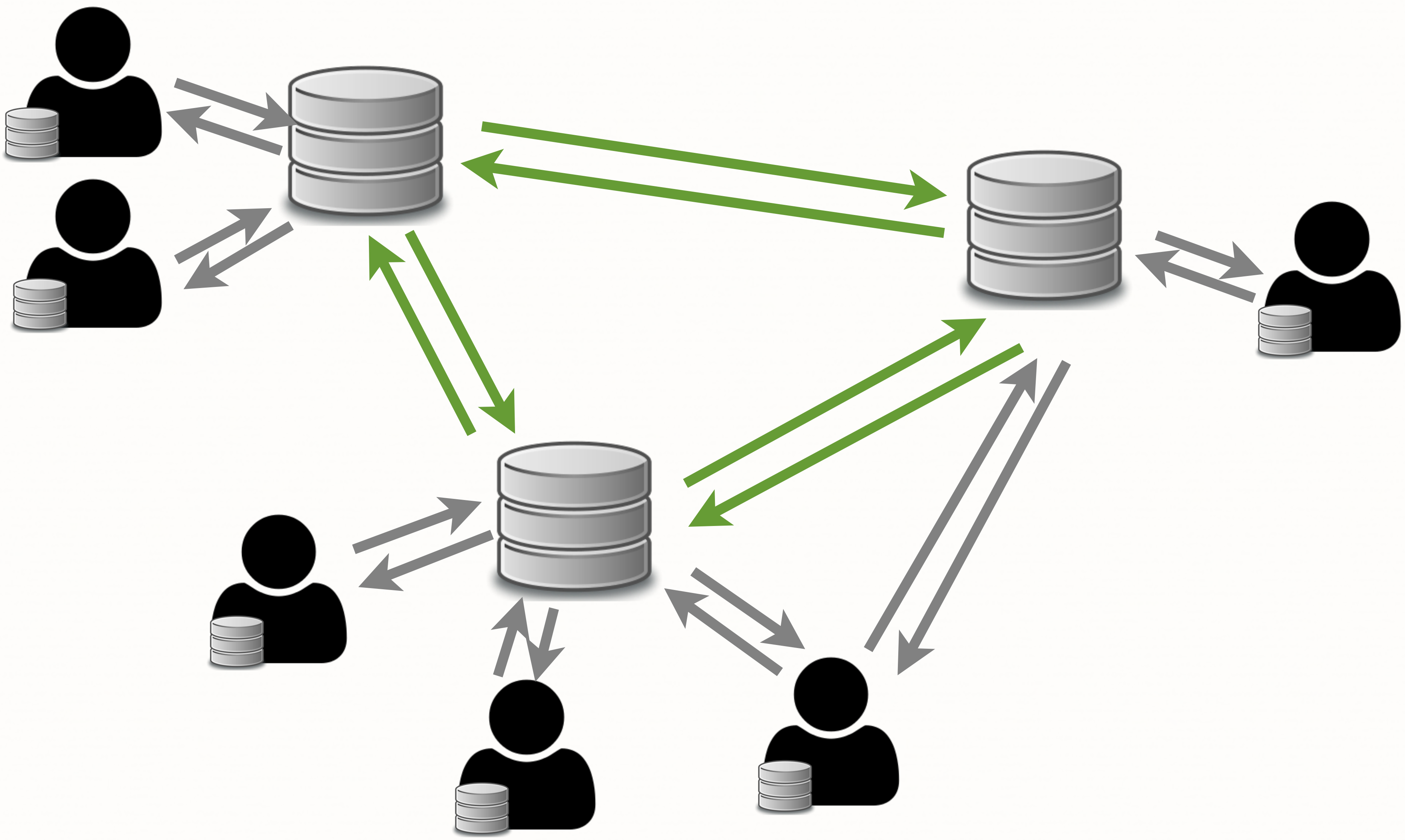
CVS (1986), SVN (2000)

- CVS = Concurrent Versions System
- SVN = Subversion
- Client-server architecture



GIT (2005)、MERCURIAL (2005)

- Distributed



GIT V.S. SVN

- https://git-scm.com/about
- https://git.wiki.kernel.org/index.php/GitSvnComparsion

PROJECTS USING GIT

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

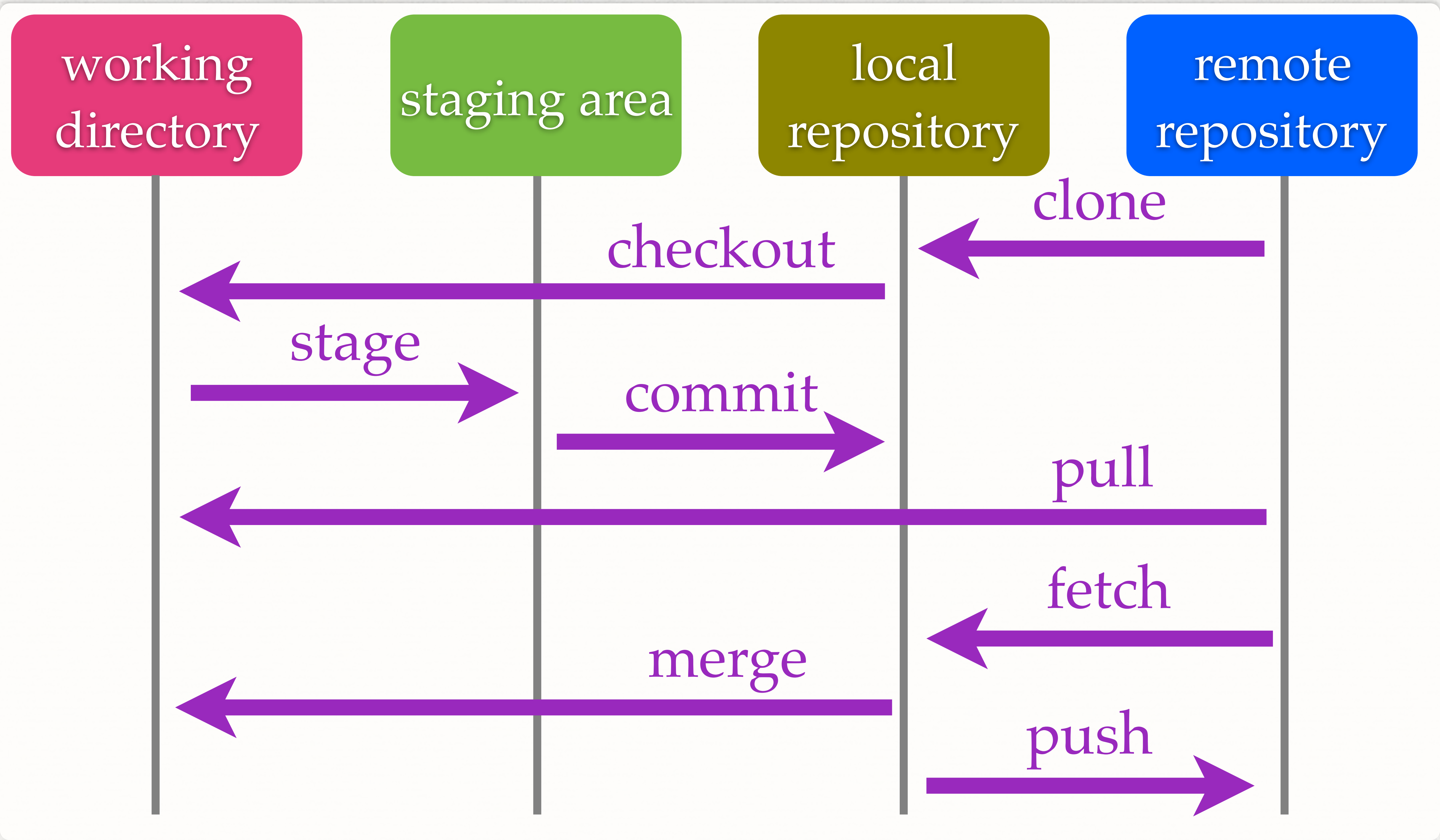
PROJECT HOSTING

- GitHub (http://github.com/):
  - Git
- Bitbucket (http://gitbucket.org/)
  - Git, Mercuial
- SourceForge (https://sourceforge.net/)
  - CVS, SVN, Bazaar, Git, Mercurial

INITIALIZE A REPOSITORY

- git init --bare --shared REPOSITORY_NAME
- git clone REMOTE

WORKING WITH GIT



working directory

staging area

local repository

remote repository

clone

checkout

stage

commit

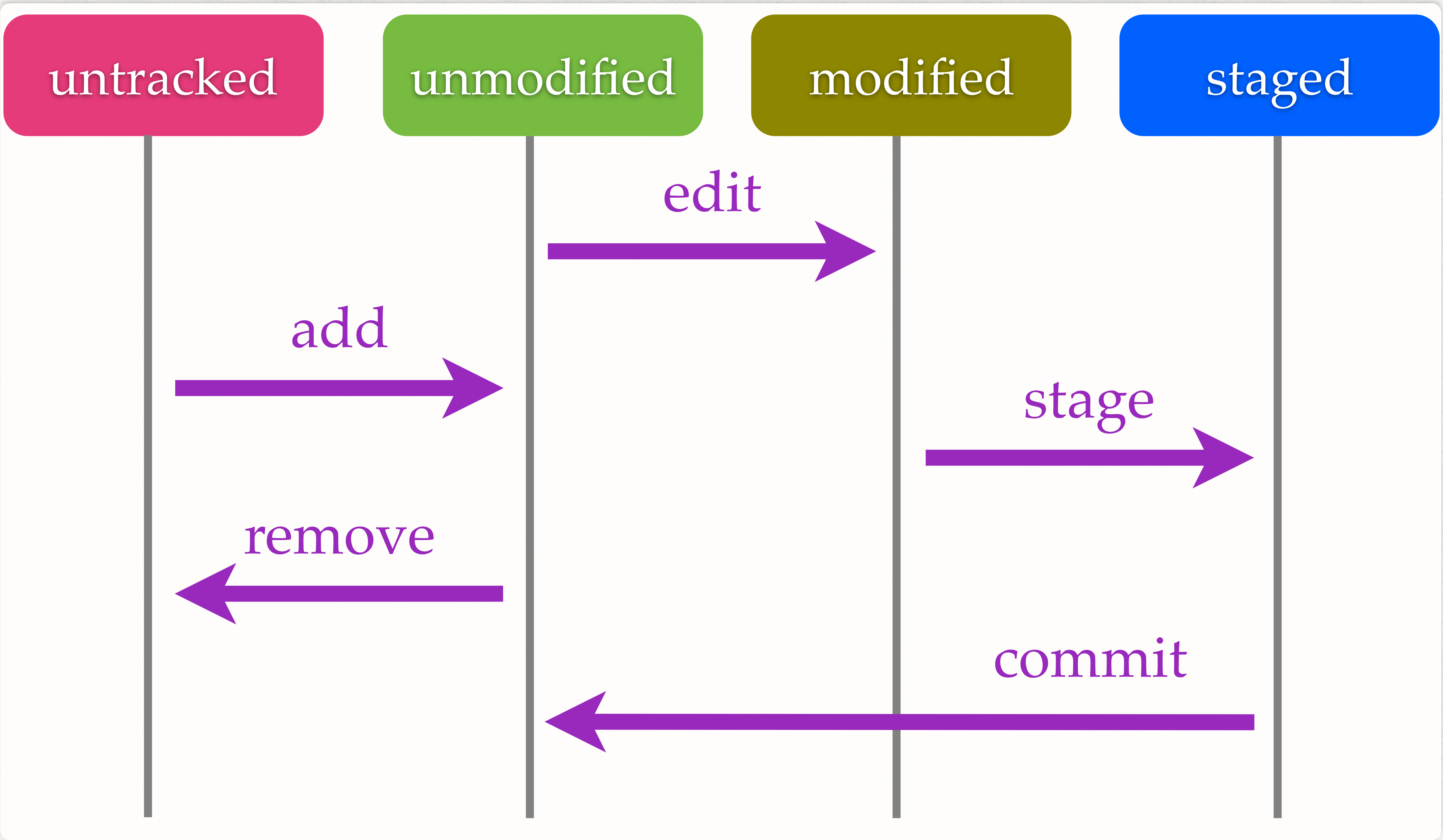
pull

fetch

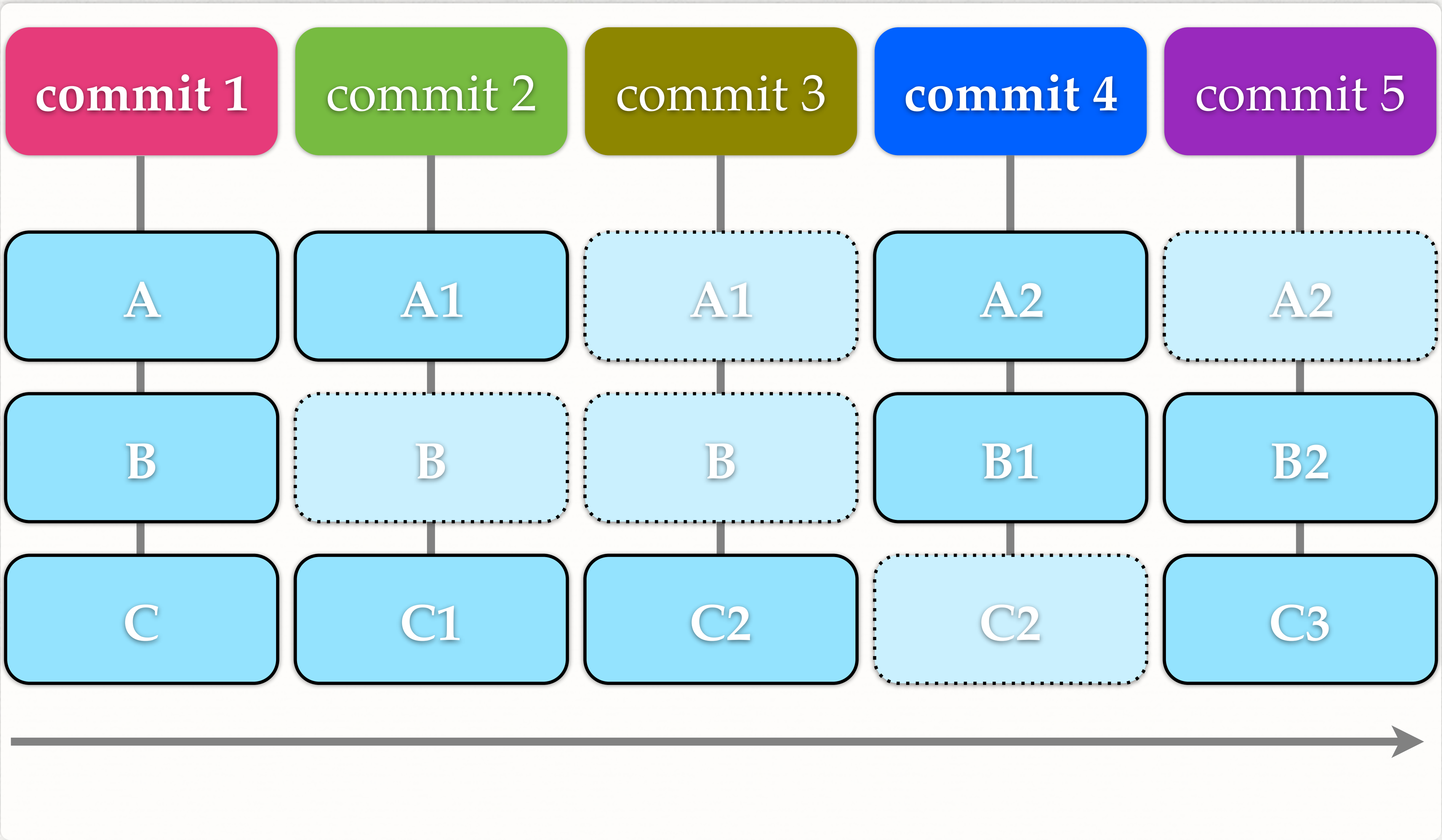
merge

push

FILE STATUS LIFECYCLE

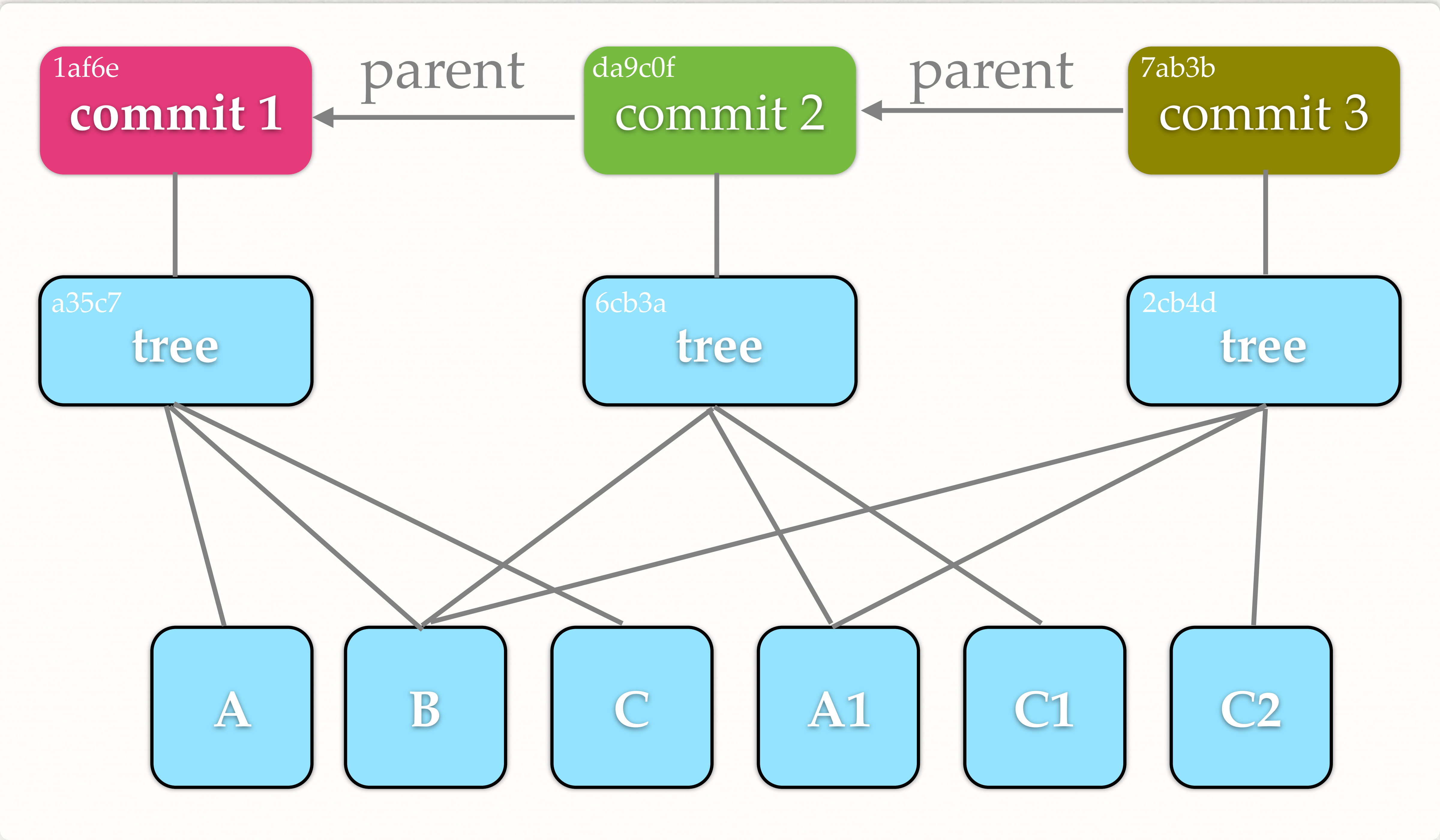


SNAPSHOTS

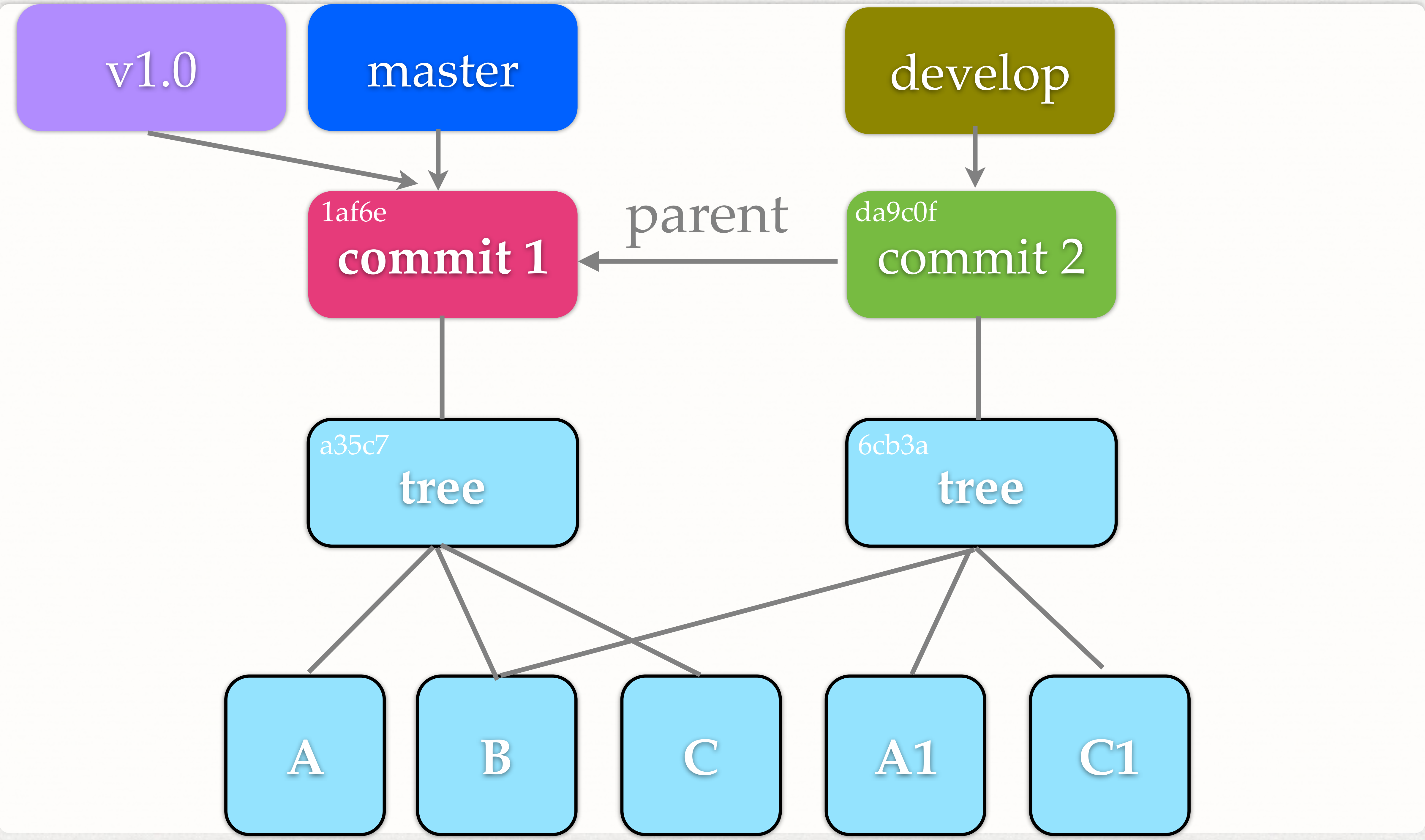


commit 1	commit 2	commit 3	commit 4	commit 5
A	A1	A1	A2	A2
B	B	B	B1	B2
C	C1	C2	C2	C3

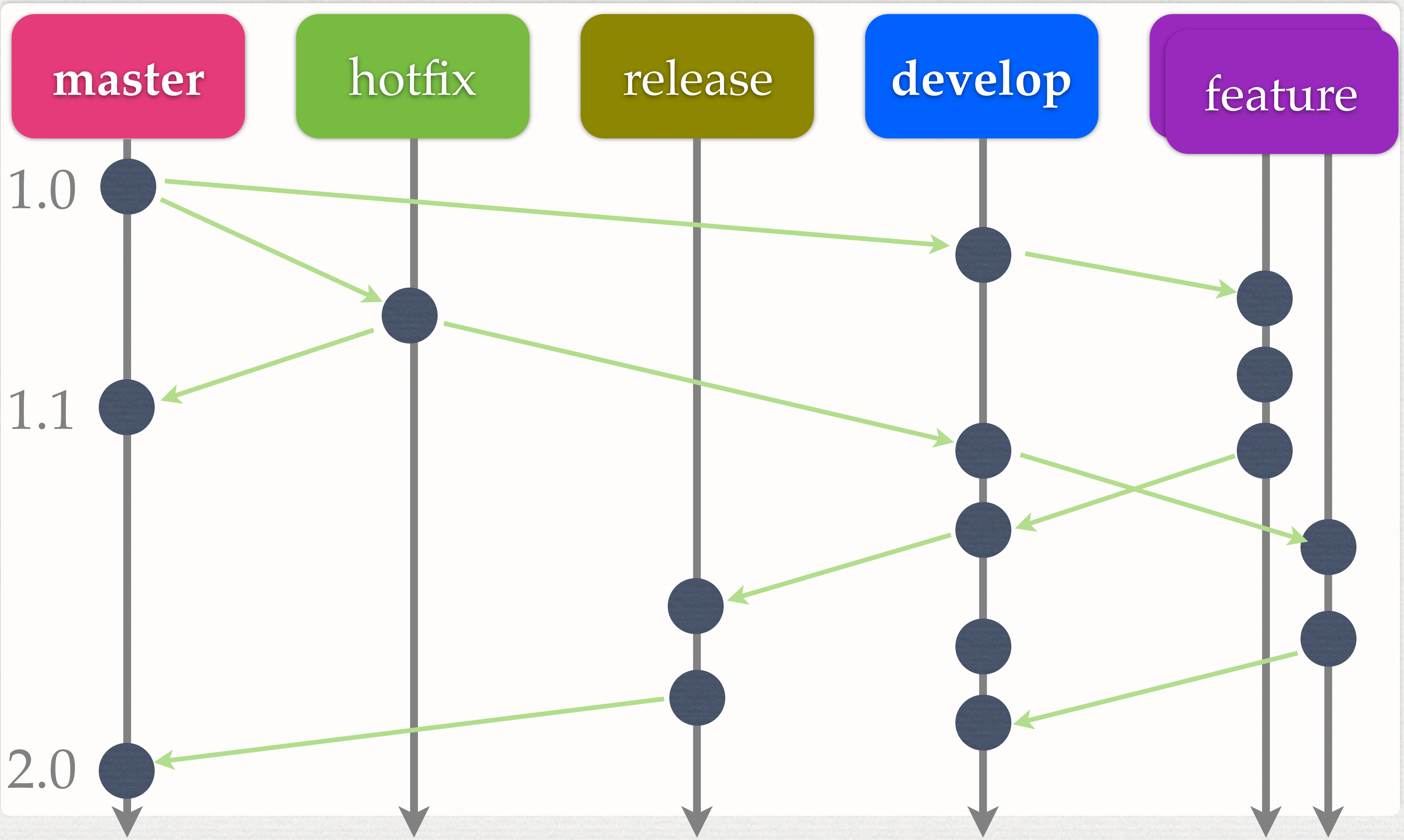
DATA MODEL


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

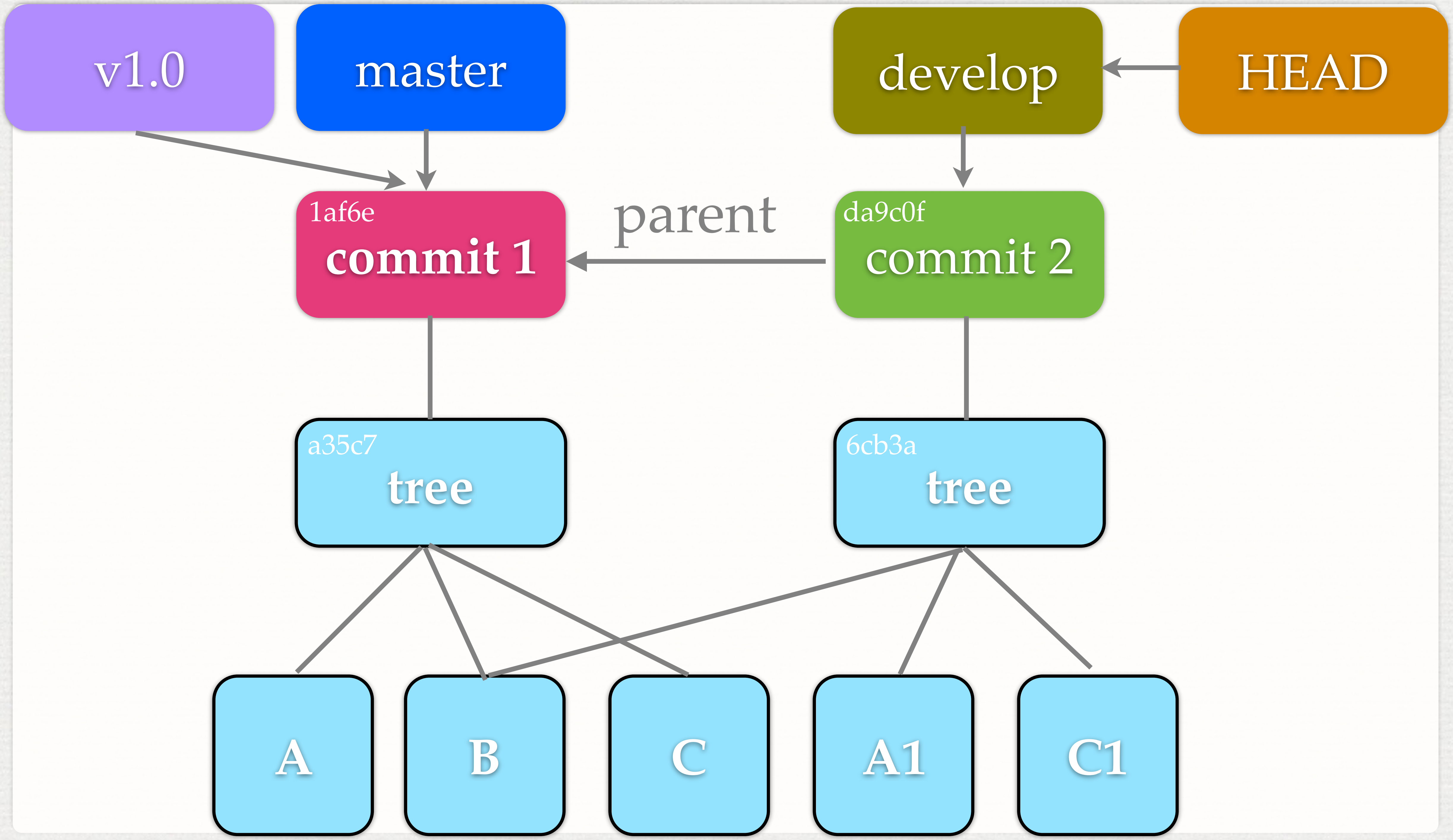


BRANCHING MODEL

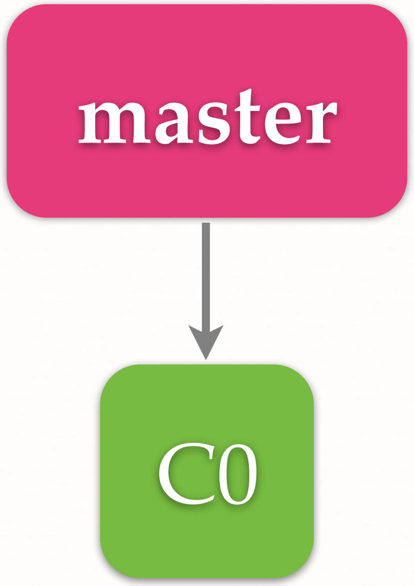


http://nvie.com/posts/a-successful-git-branching-model/

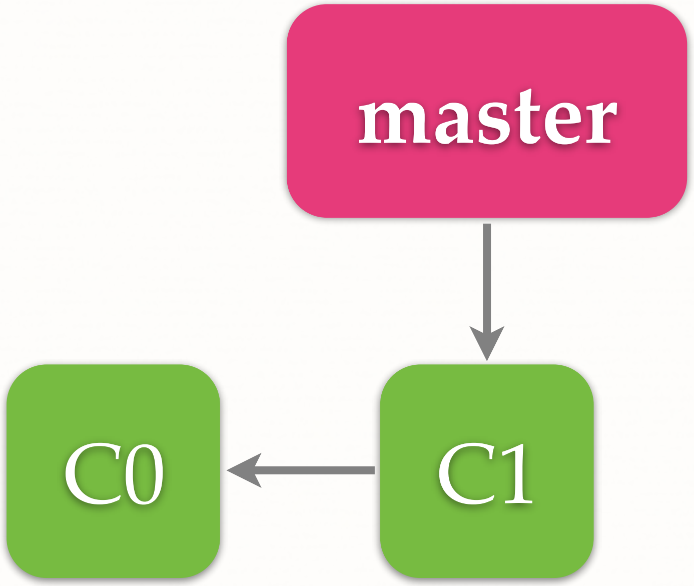
HEAD


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

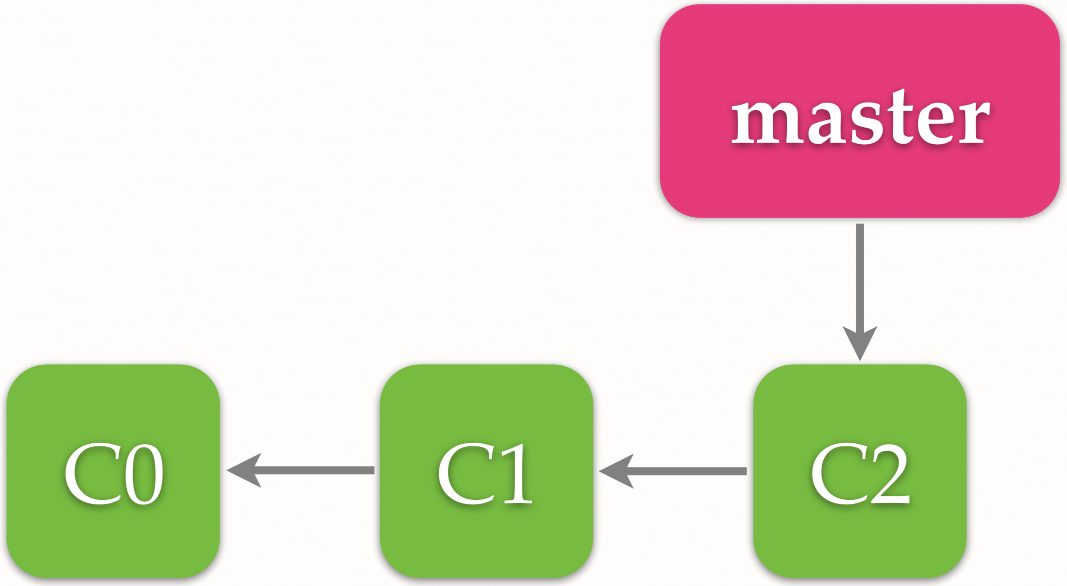
COMMITS



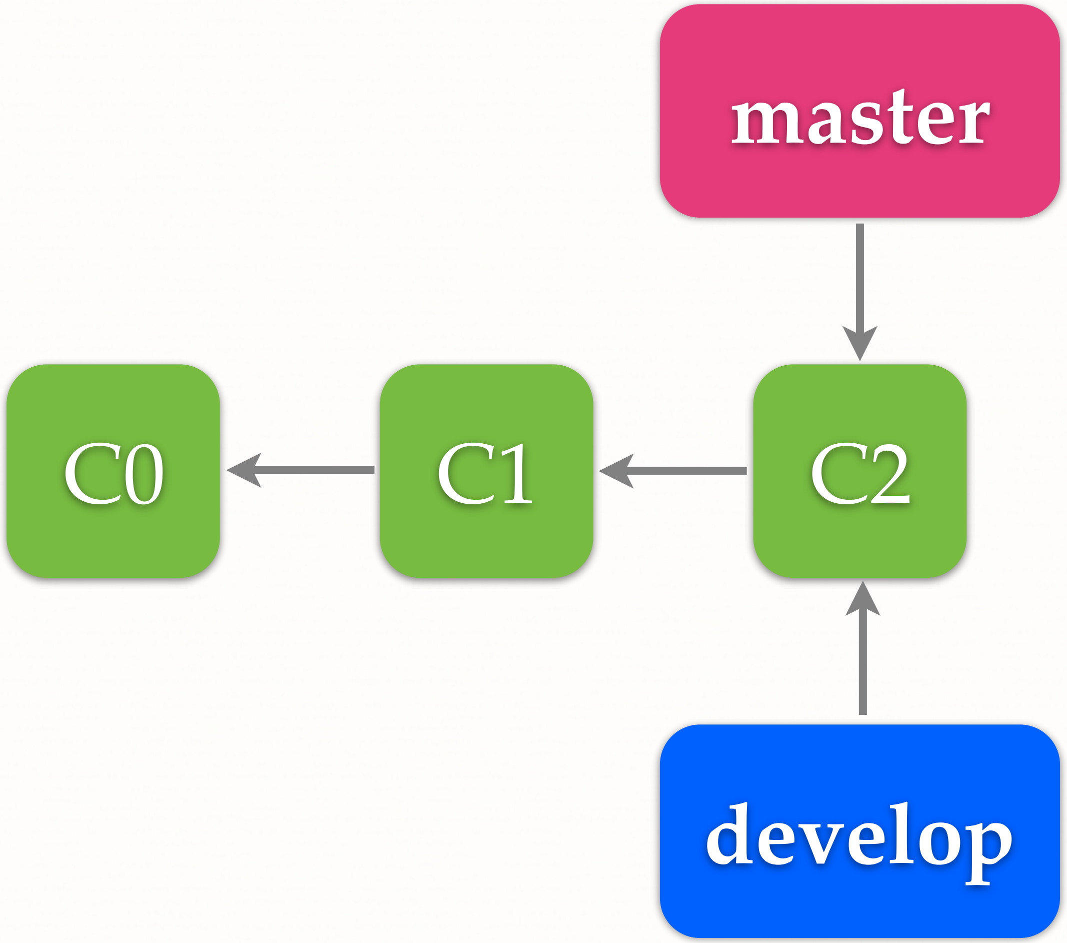
COMMITS



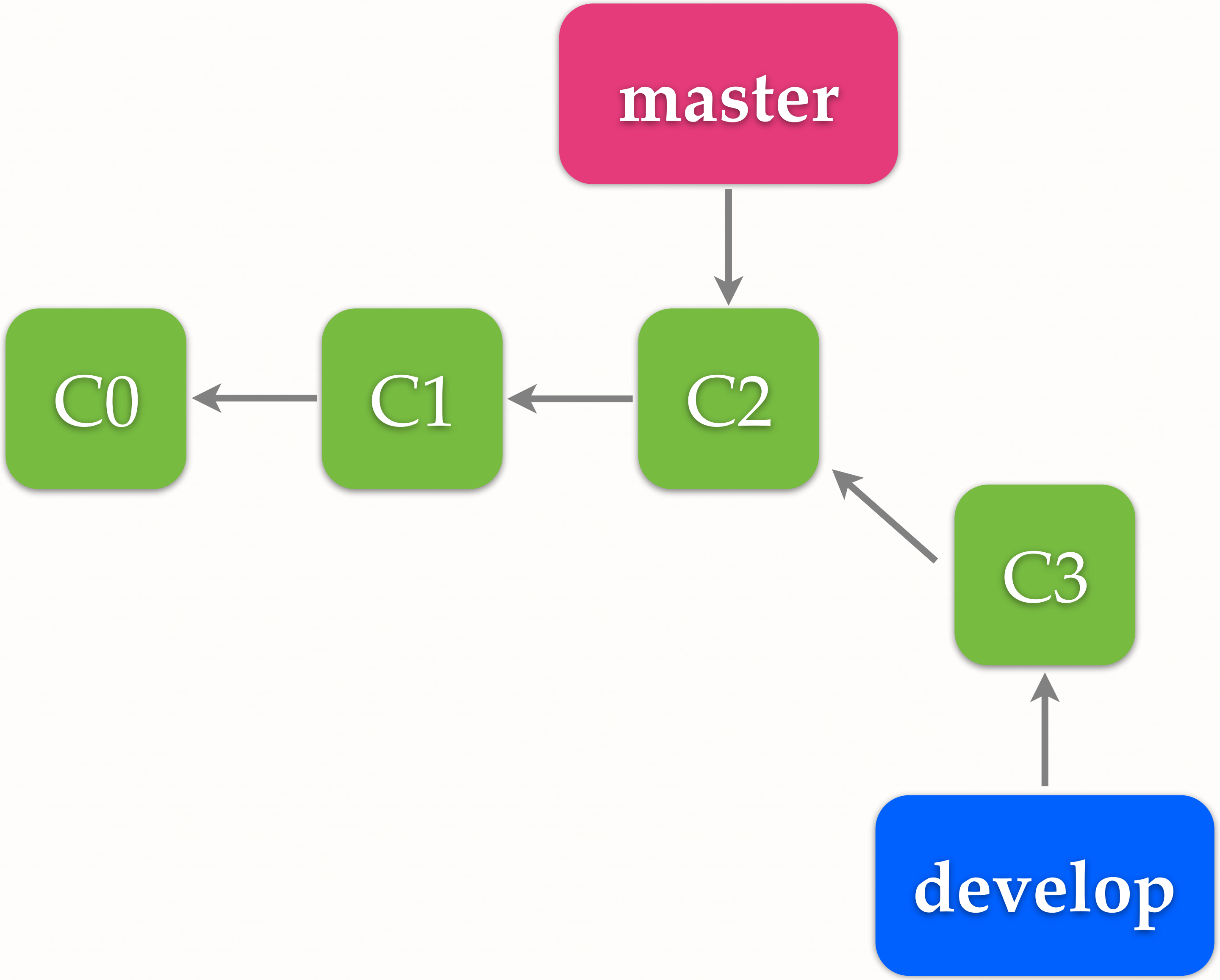
COMMITS



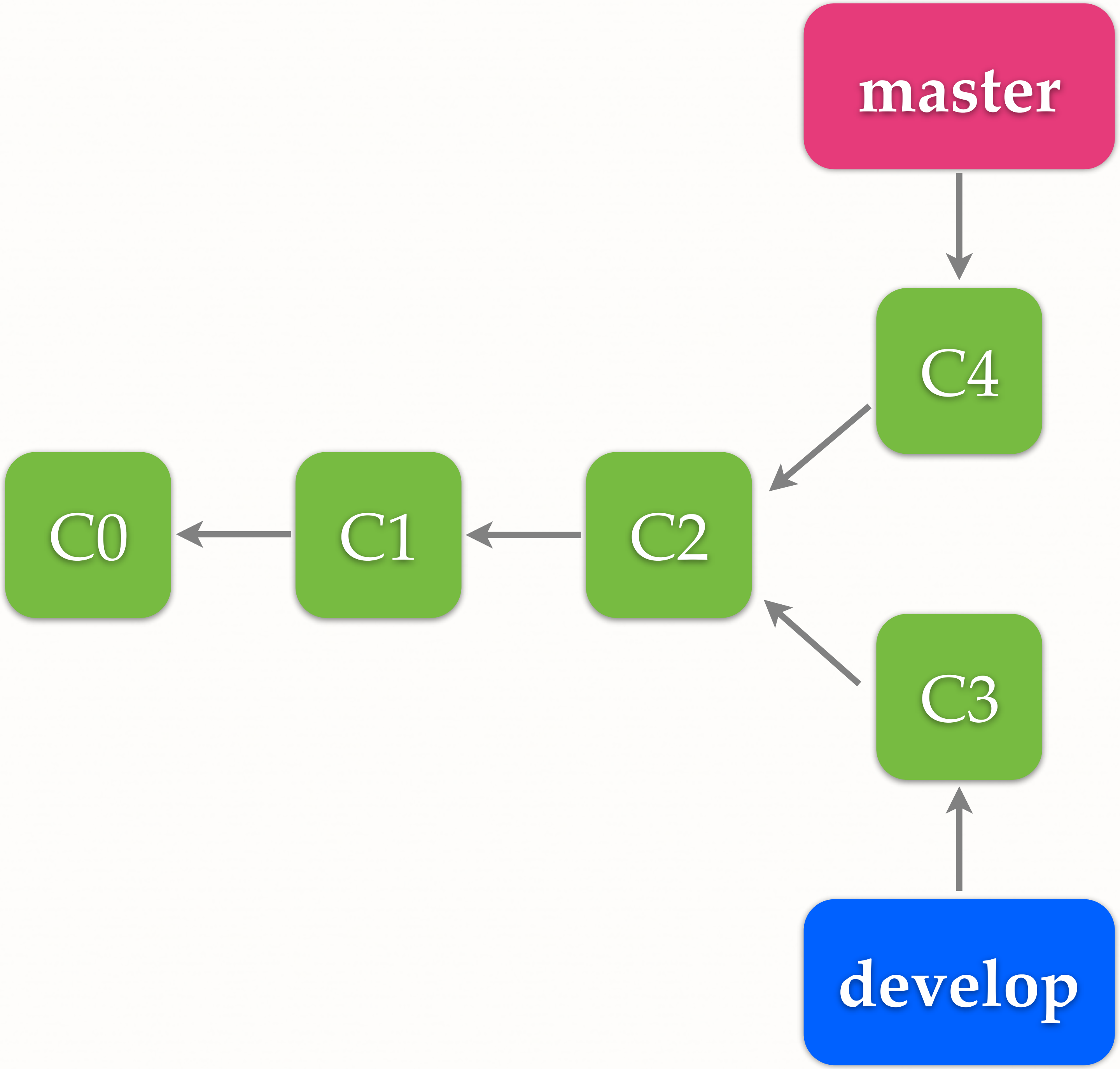
COMMITS



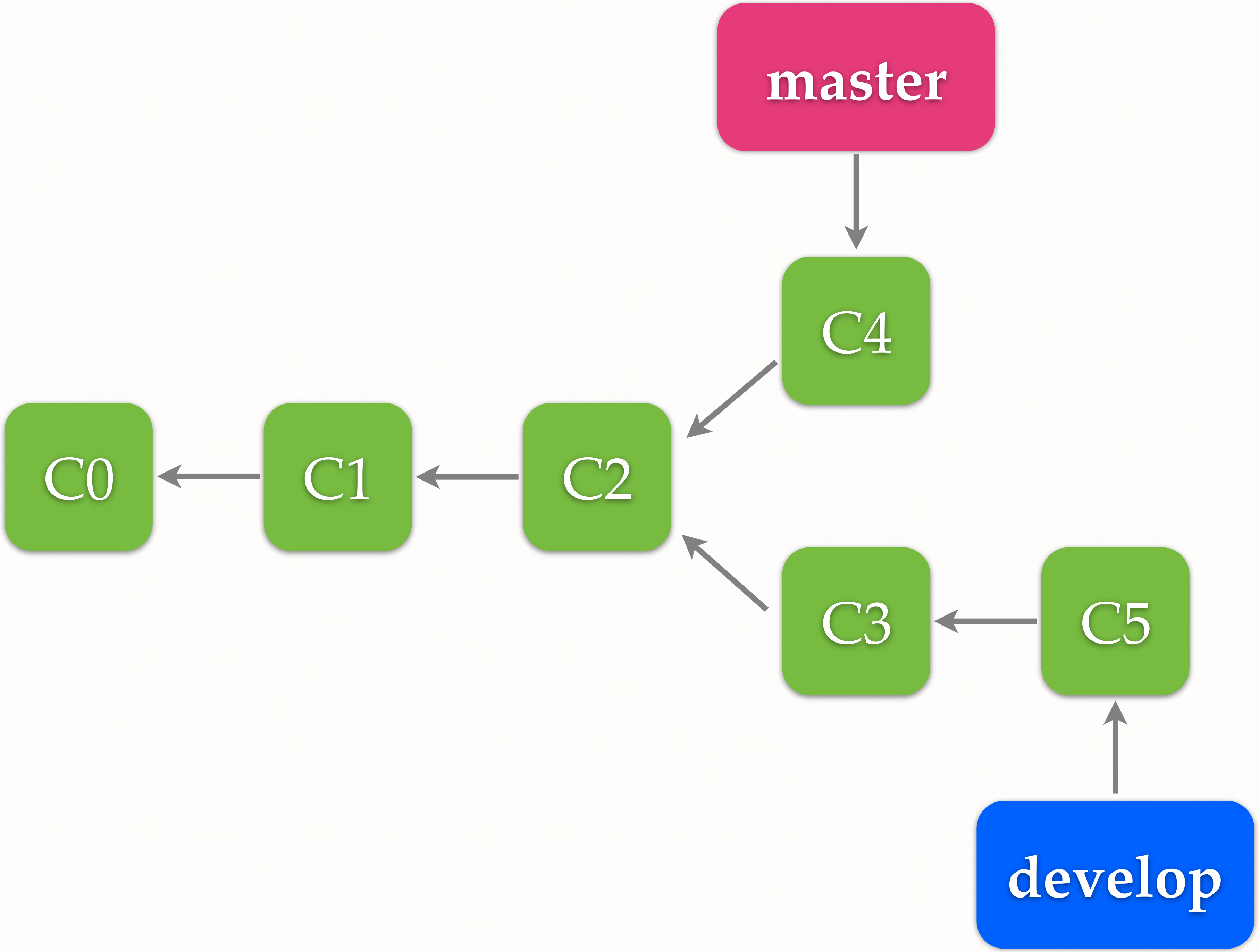
COMMITS



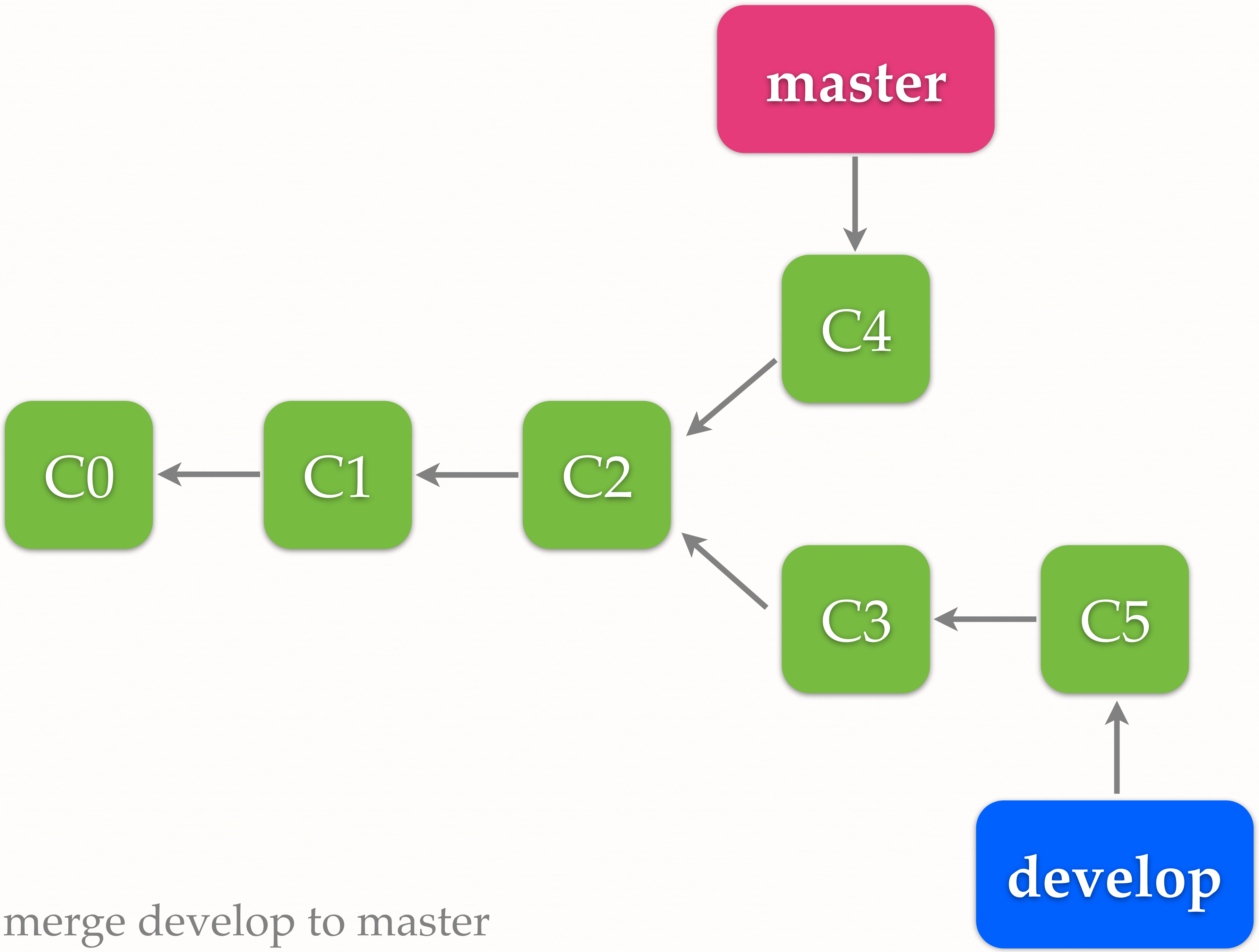
COMMITS



COMMITS

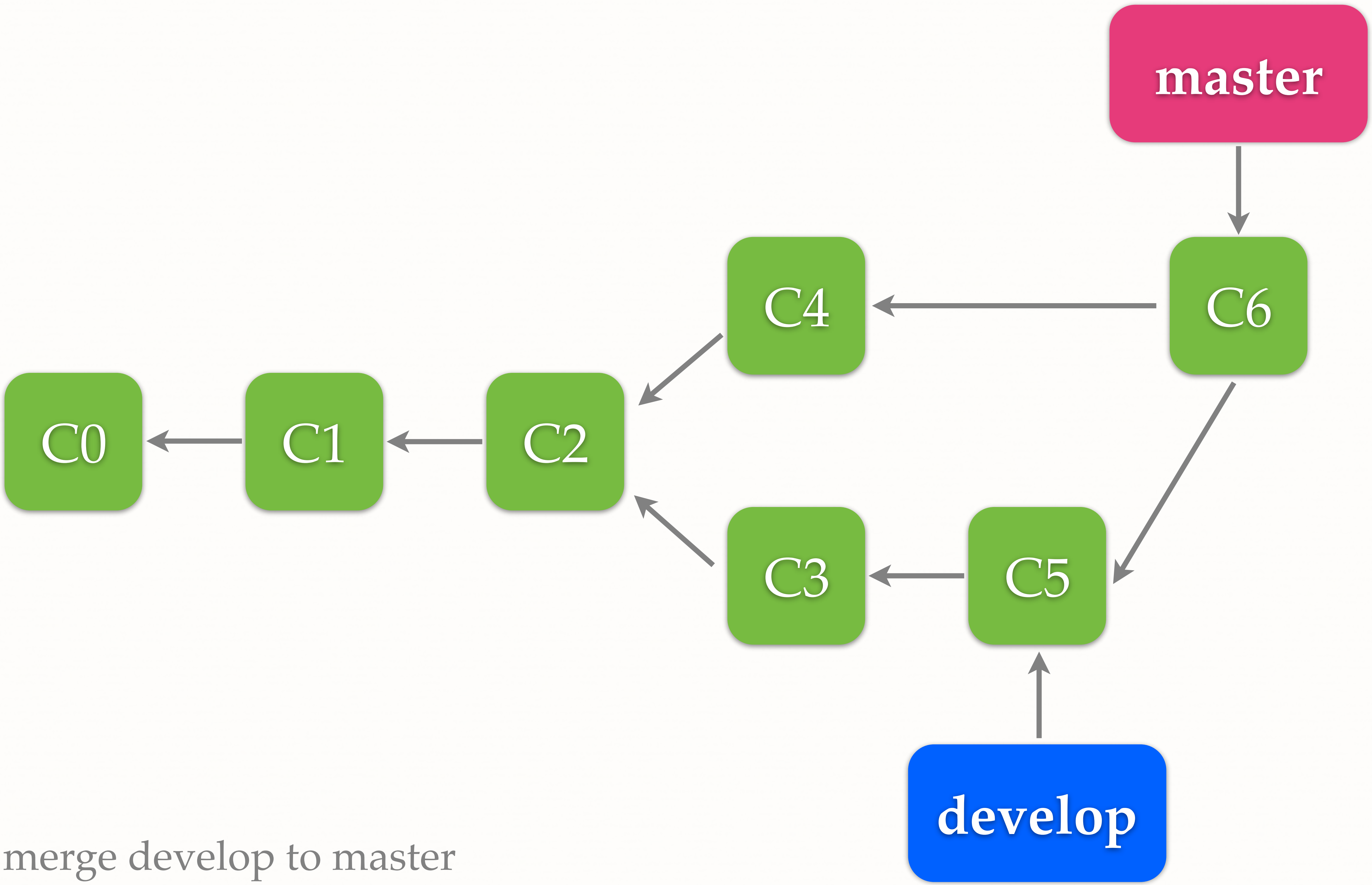


MERGE



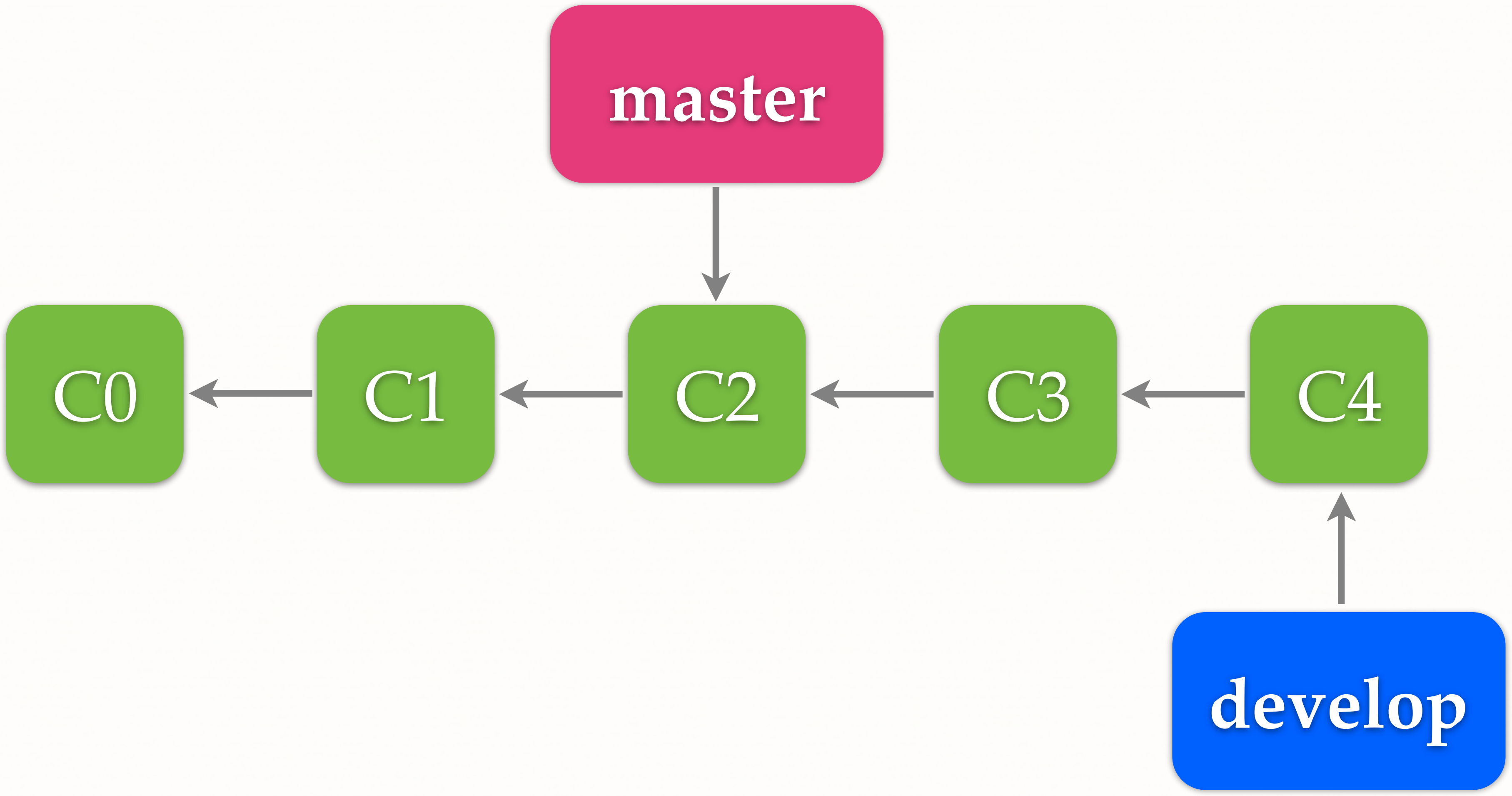
merge develop to master

MERGE



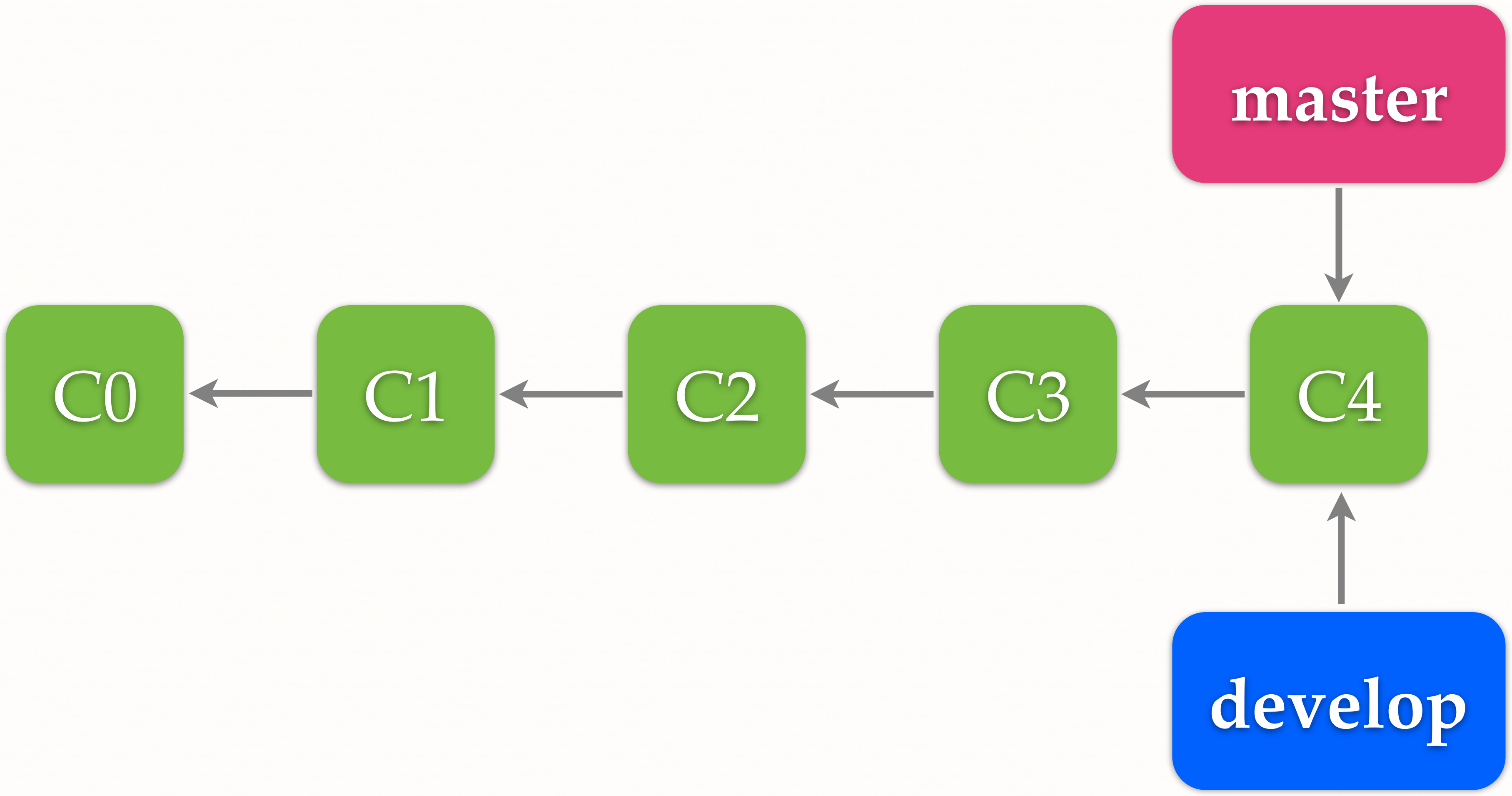
merge develop to master

FAST-FORWARD



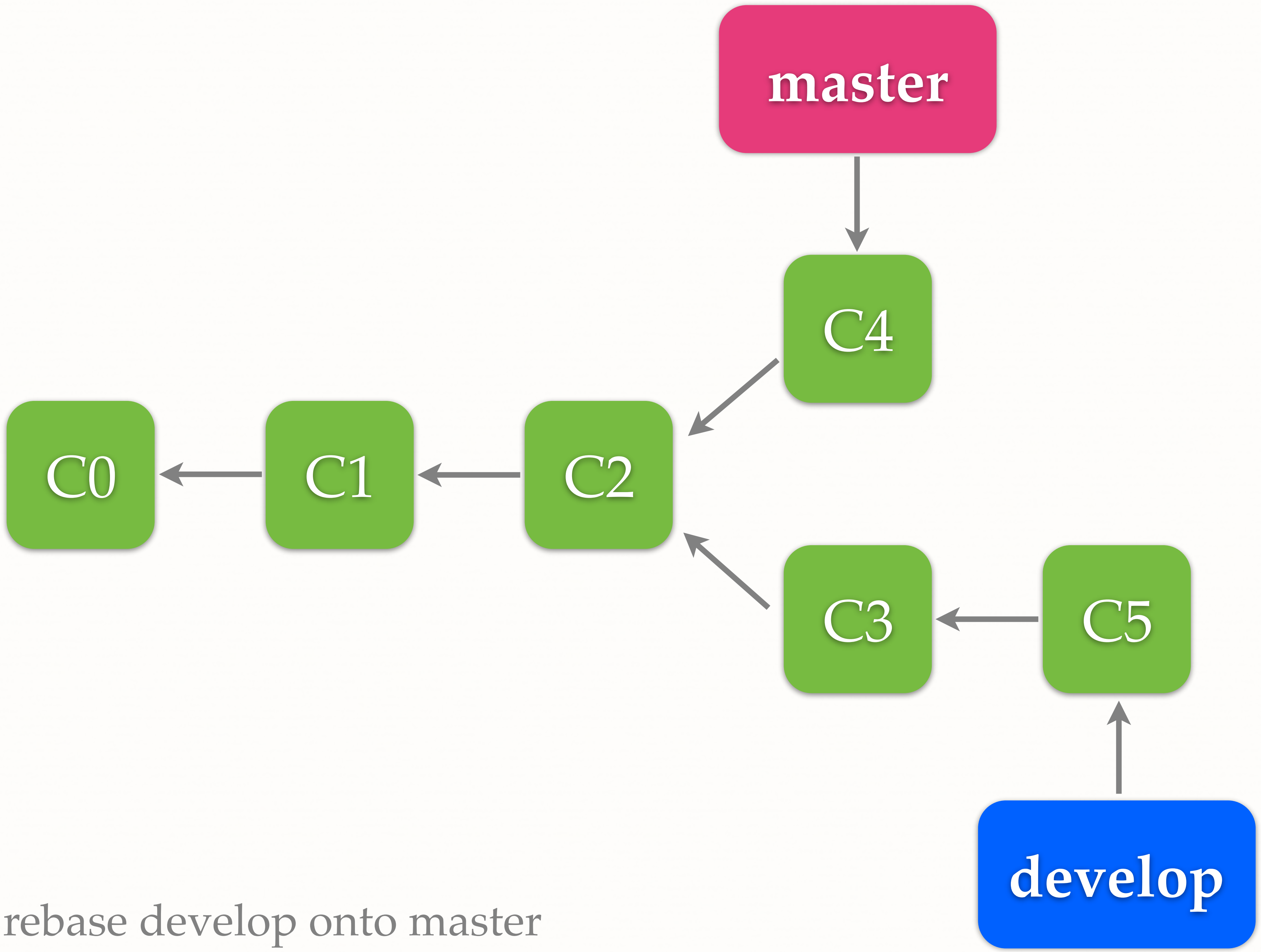
merge develop to master

FAST-FORWARD



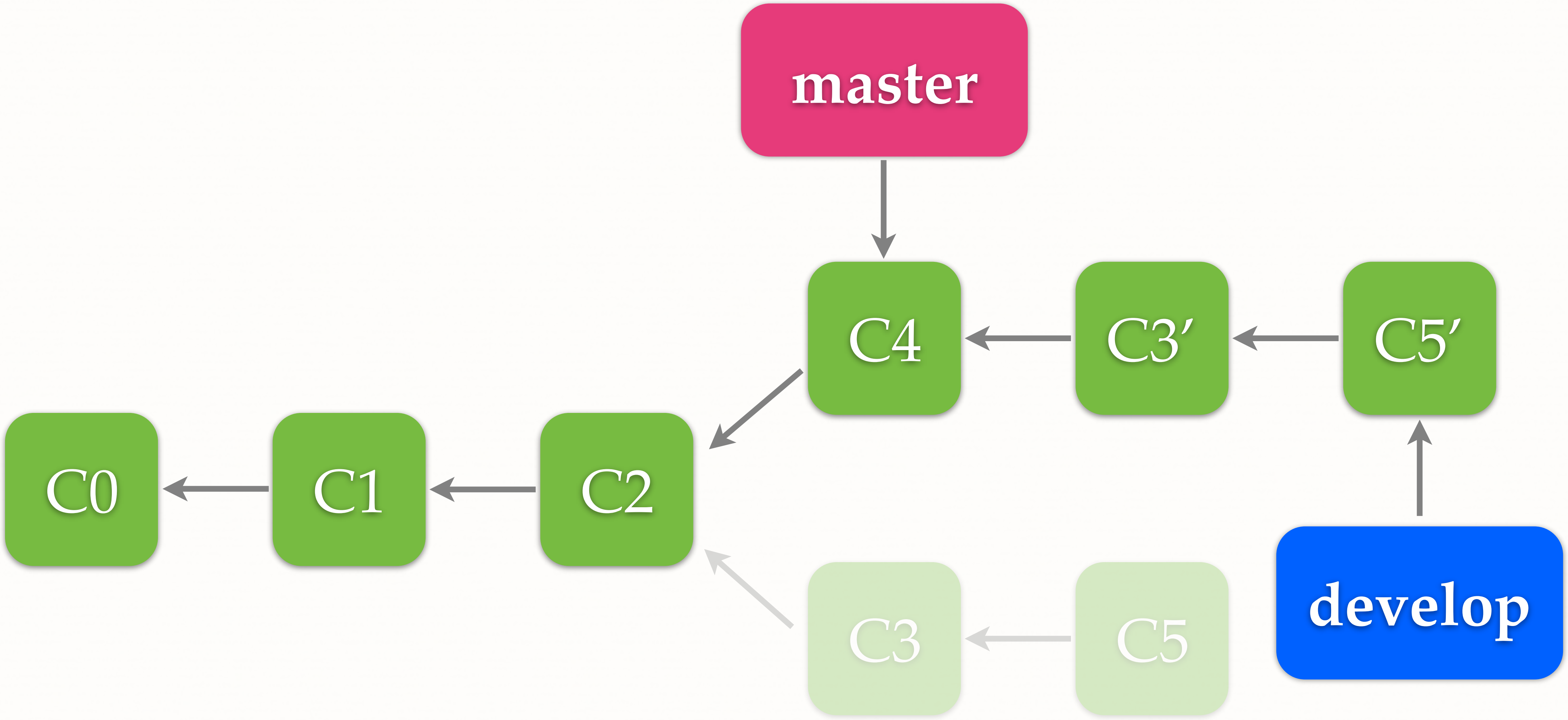
merge develop to master

REBASE



rebase develop onto master

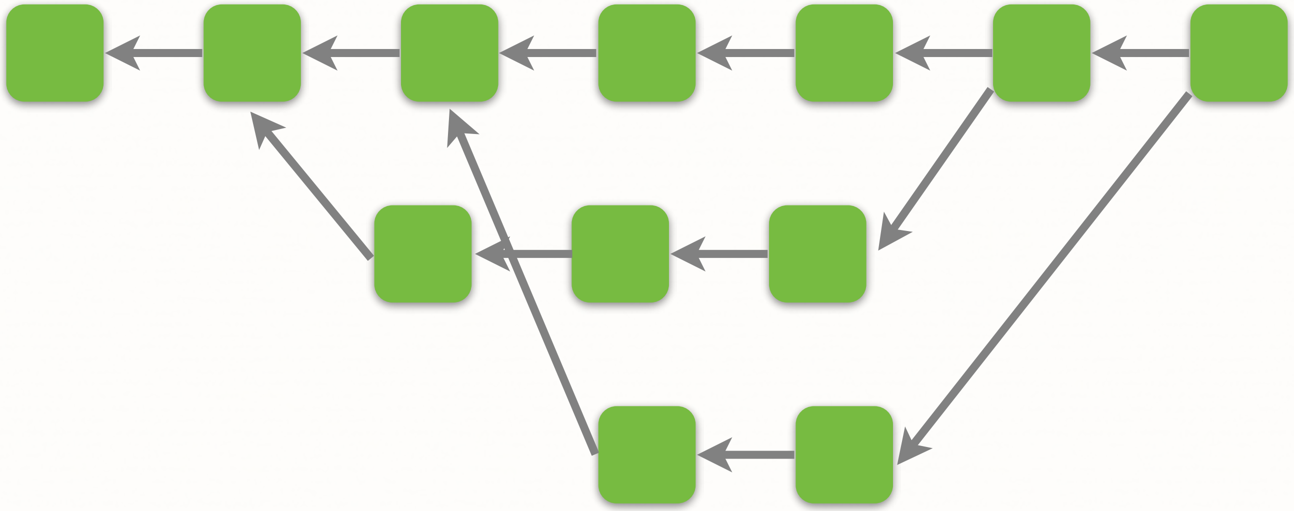
REBASE



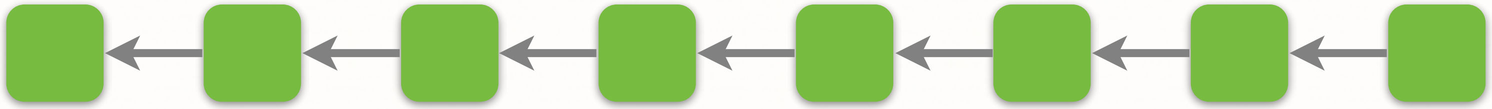
rebase develop onto master

REBASE V.S. MERGE

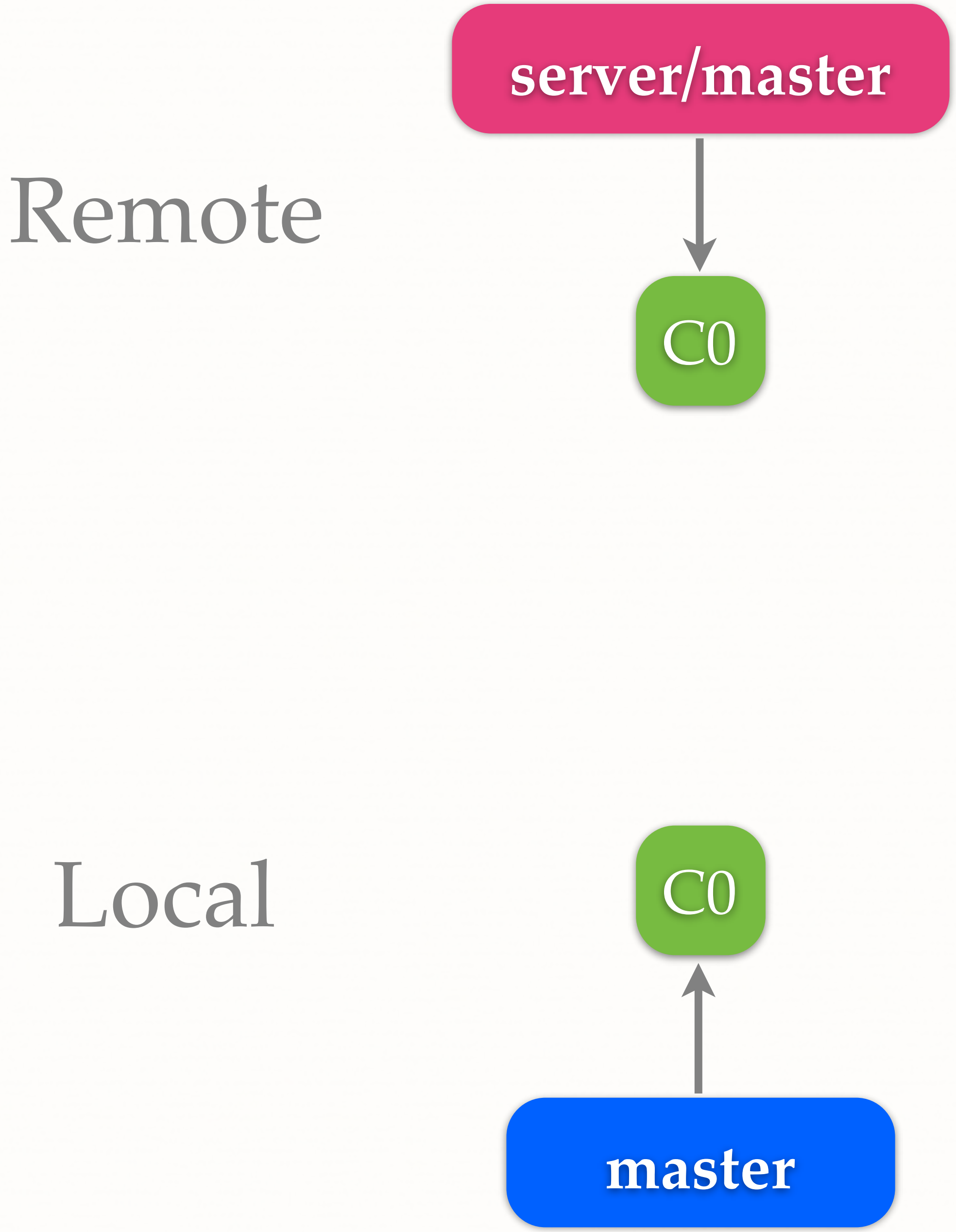
Merge



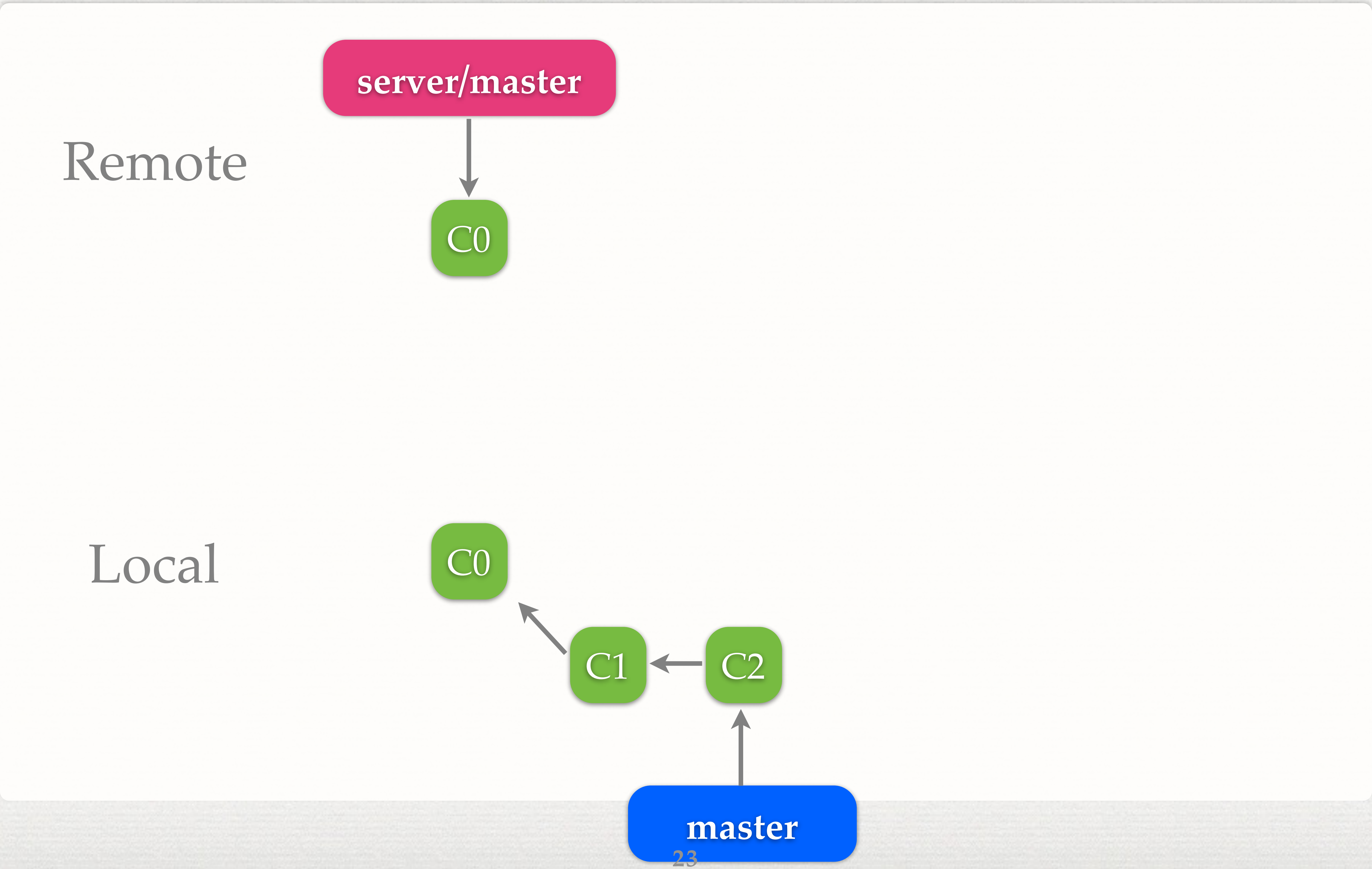
Rebase



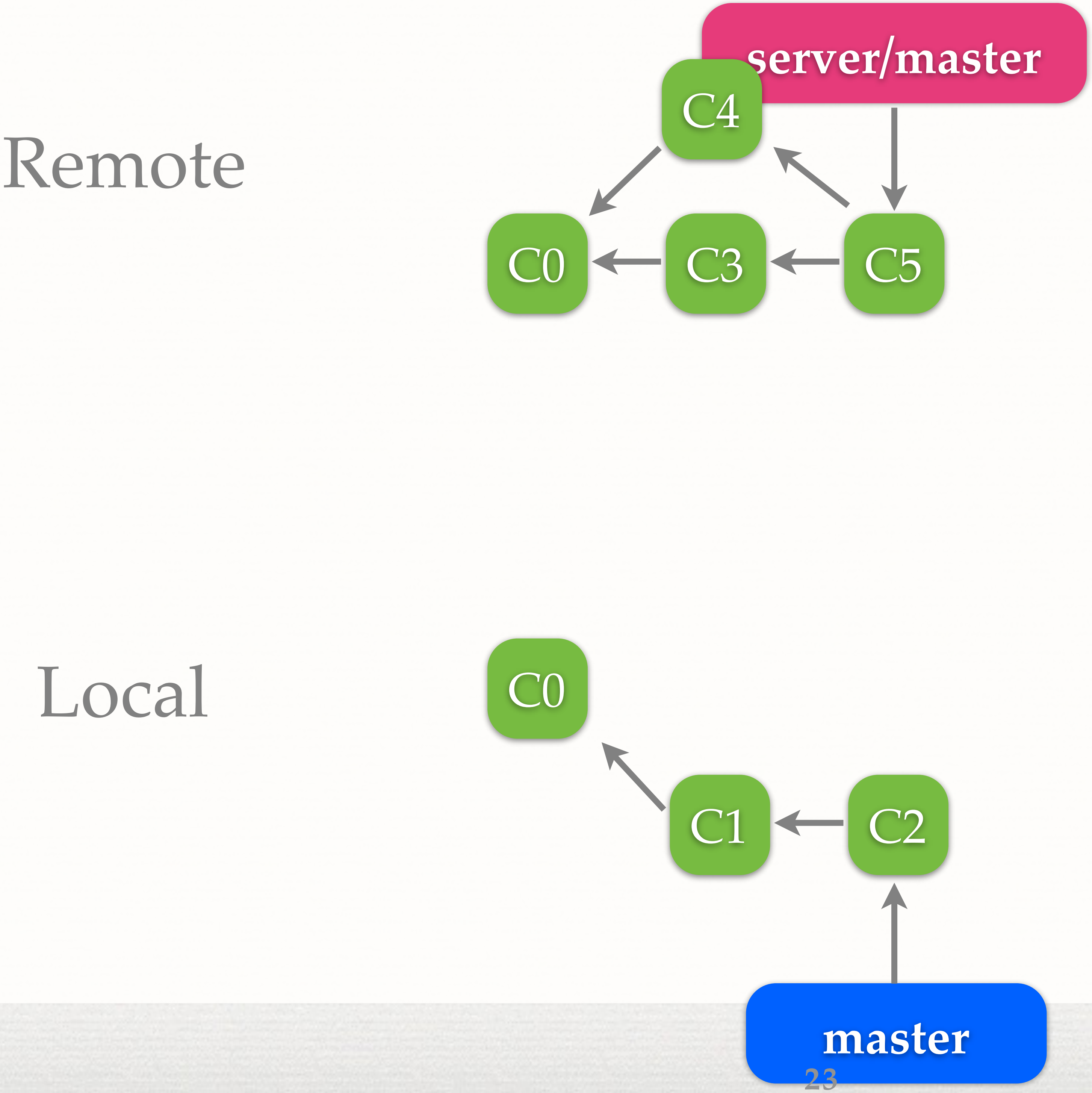
REBASE PUBLISHED COMMITS



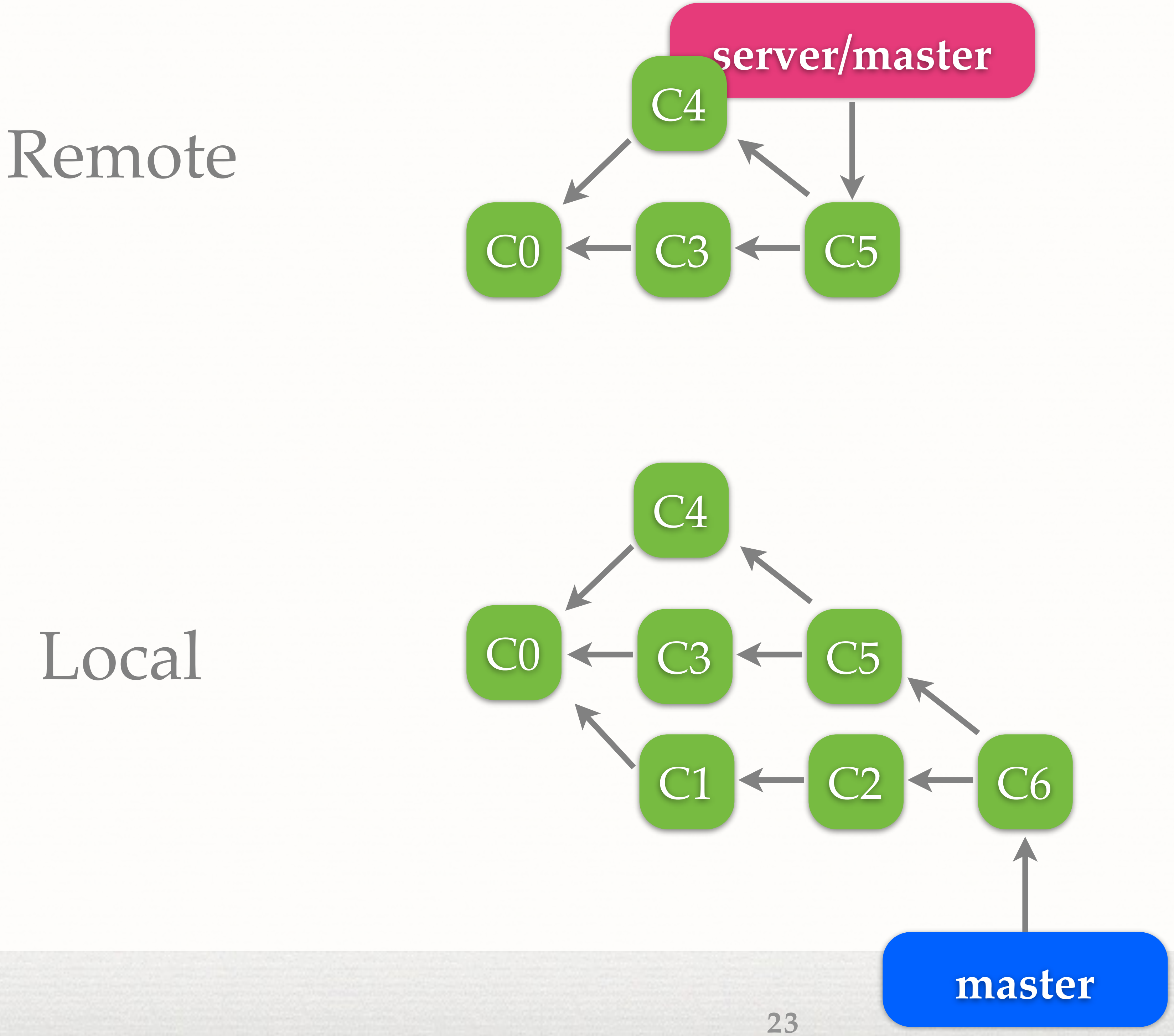
REBASE PUBLISHED COMMITS



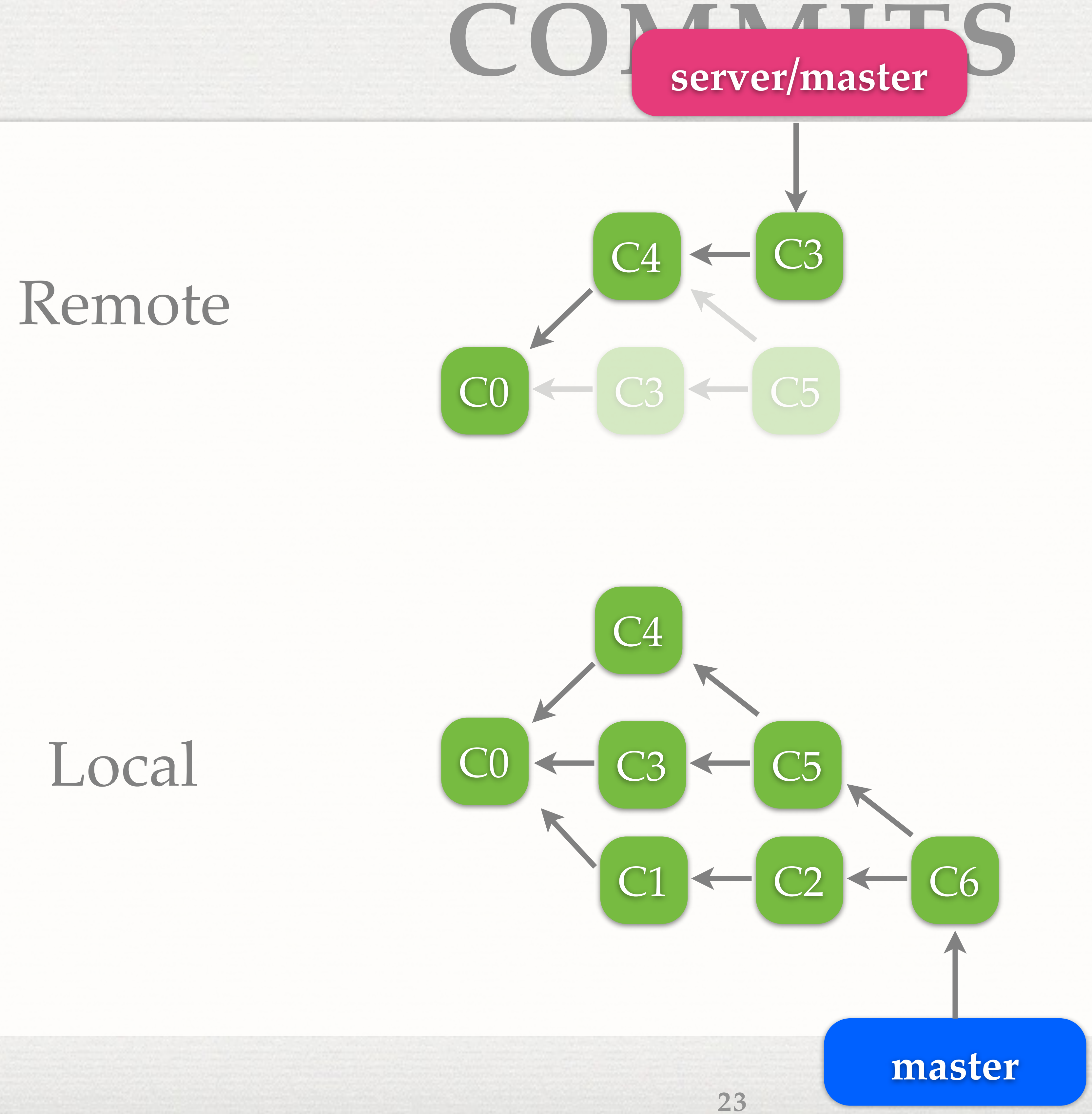
REBASE PUBLISHED COMMITS



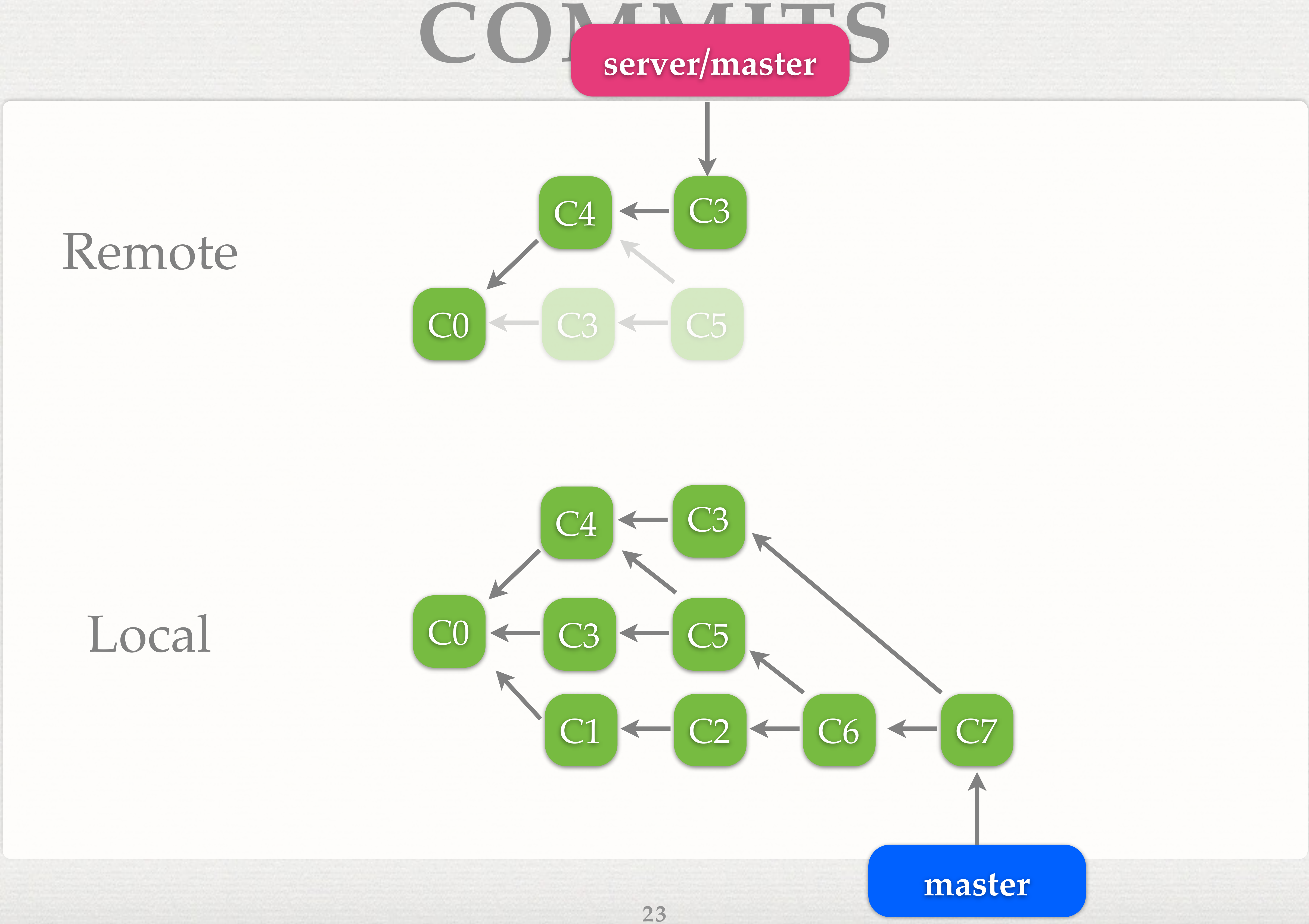
REBASE PUBLISHED COMMITS



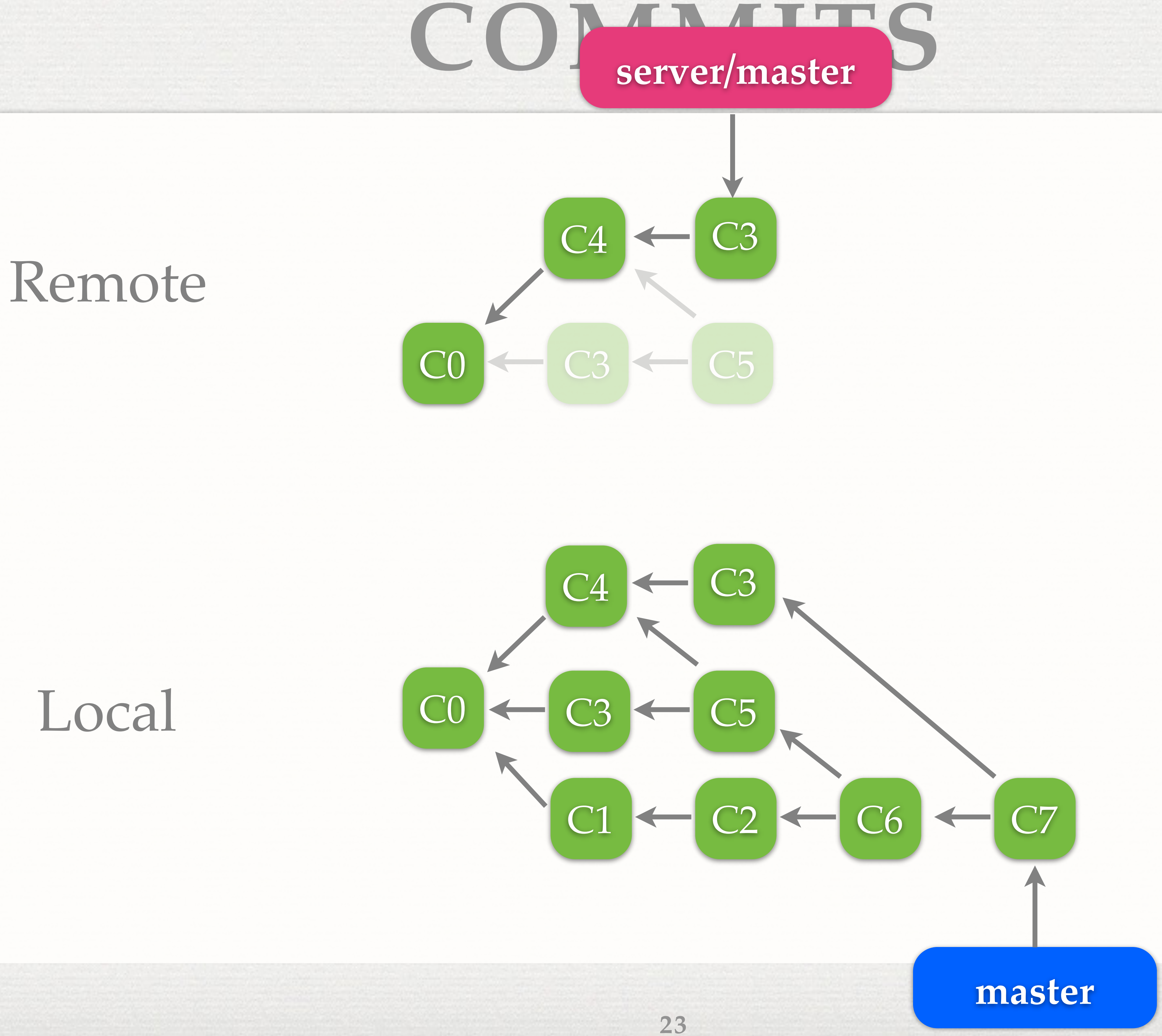
REBASE PUBLISHED COMMITS



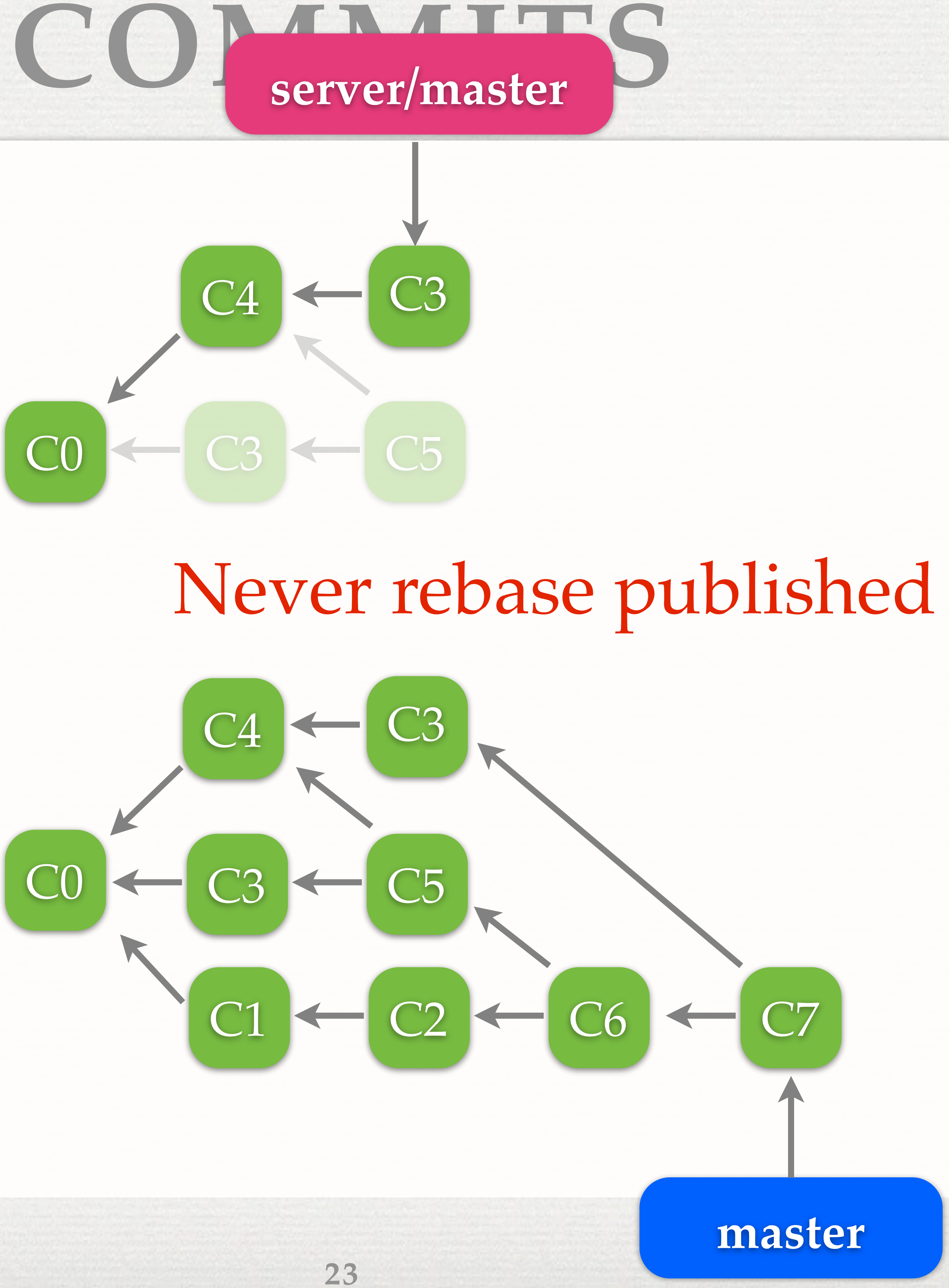
REBASE PUBLISHED COMMITS



REBASE PUBLISHED COMMITS



REBASE PUBLISHED COMMITS

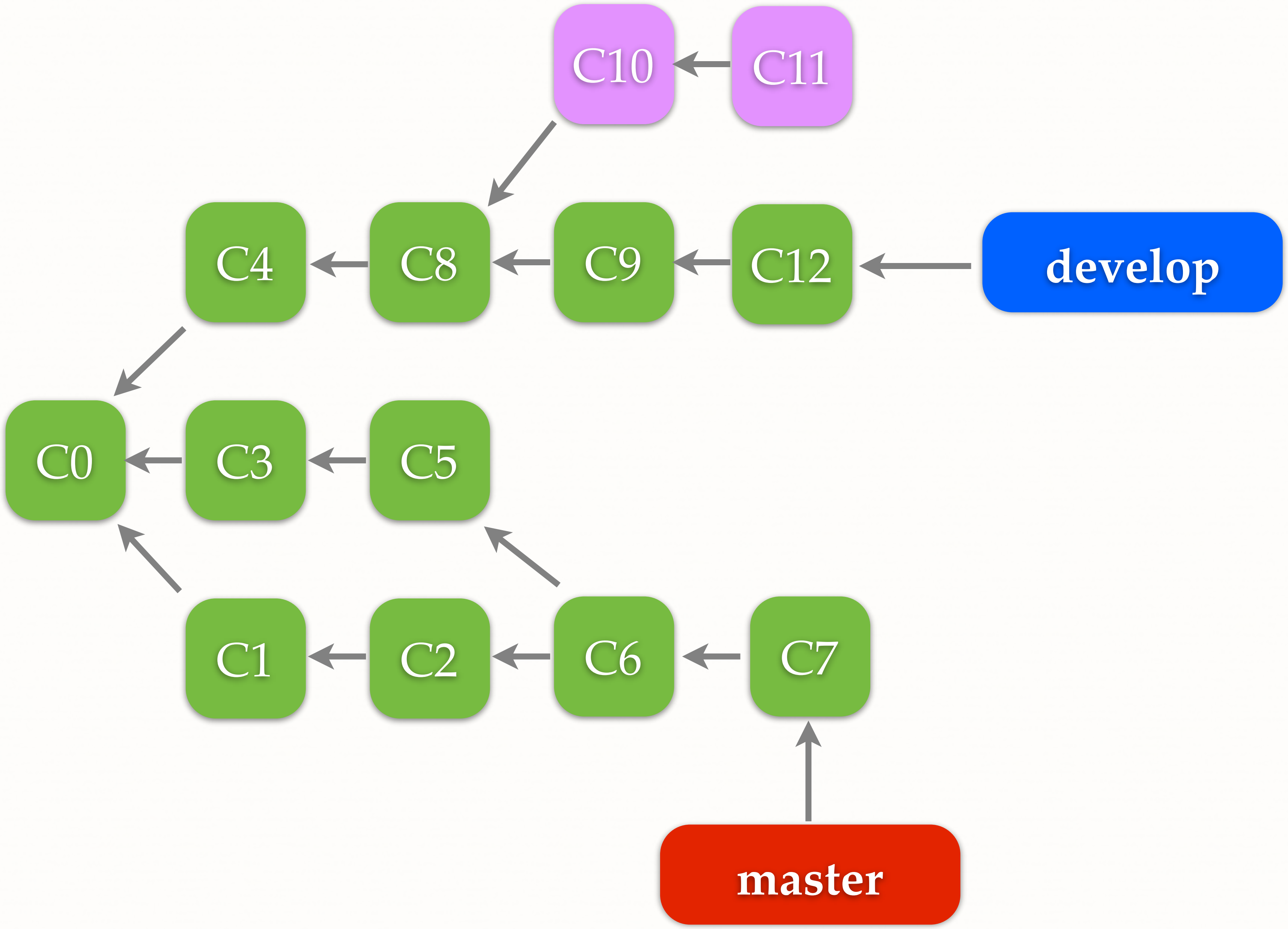


Remote

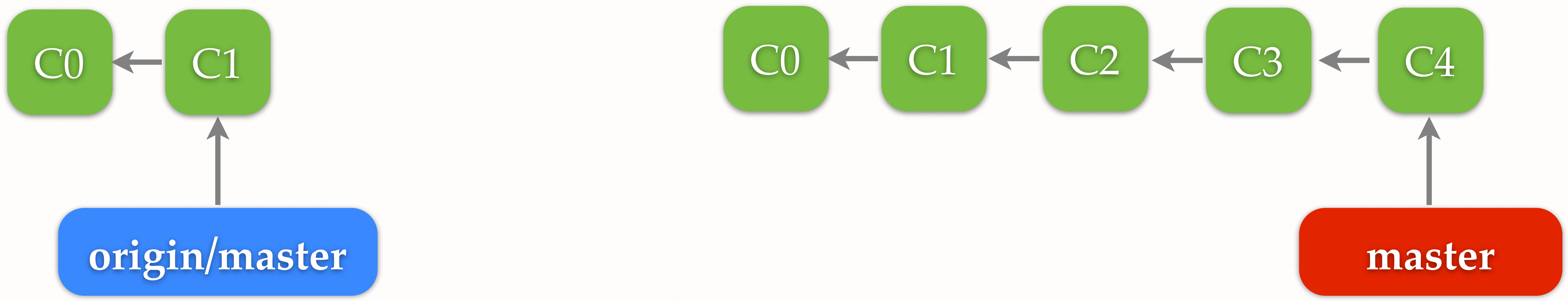
Local

Never rebase published commits

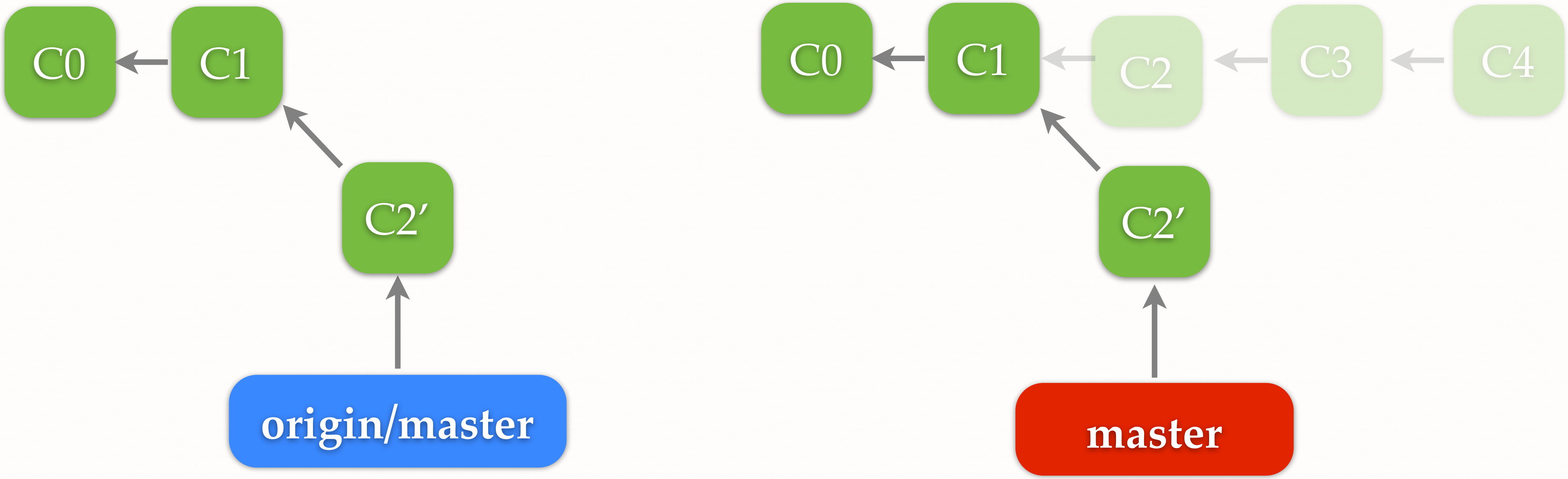
DANGLING COMMITS



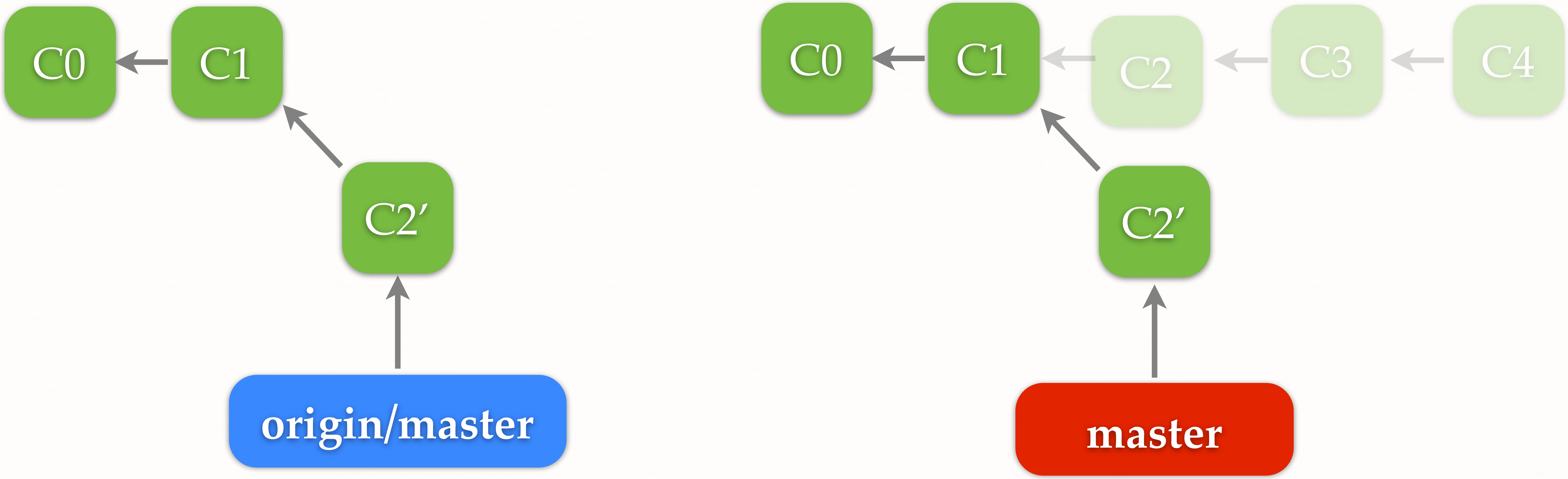
SQUASH



SQUASH

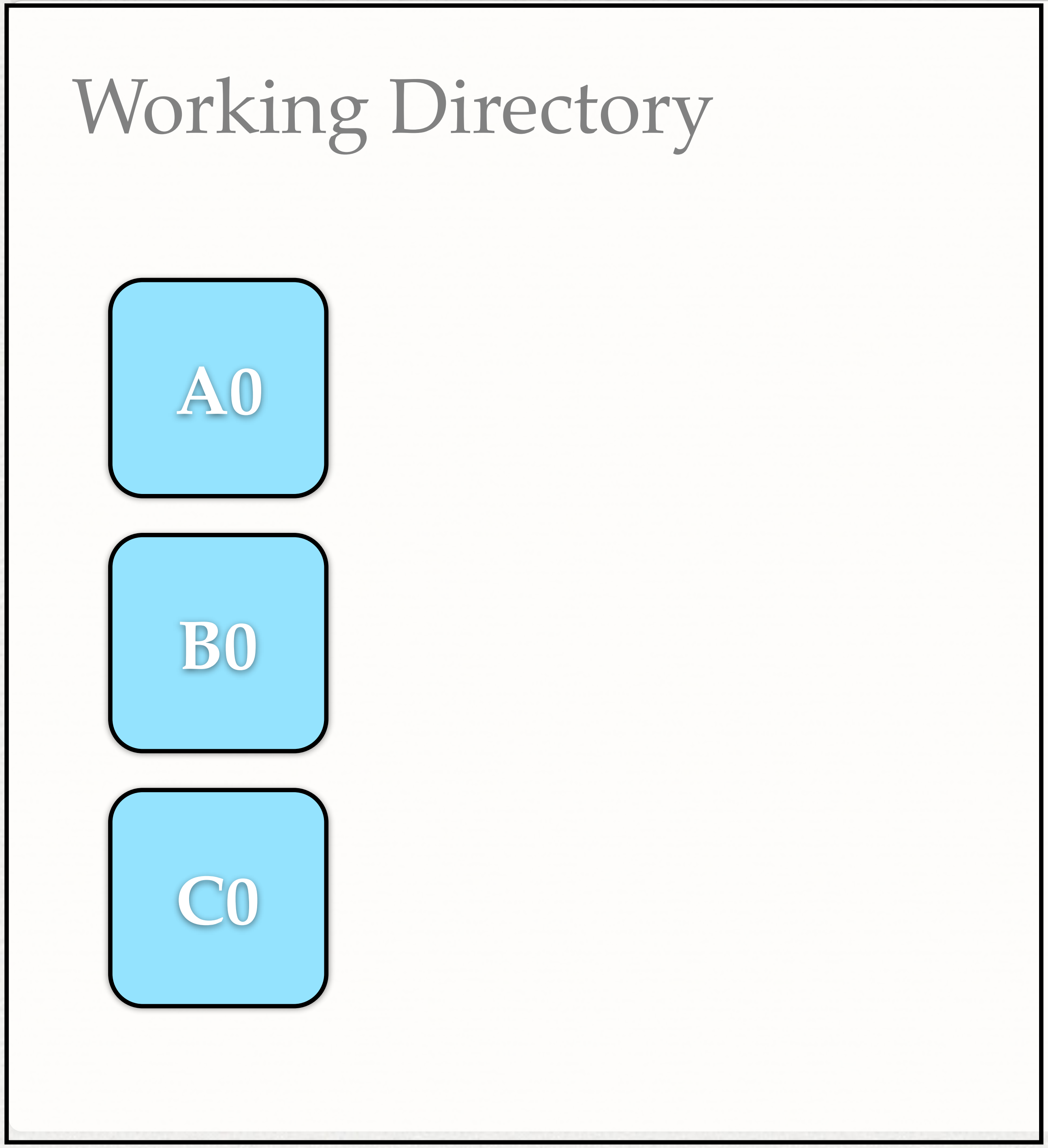


SQUASH



Don’t squash published commits

STASH



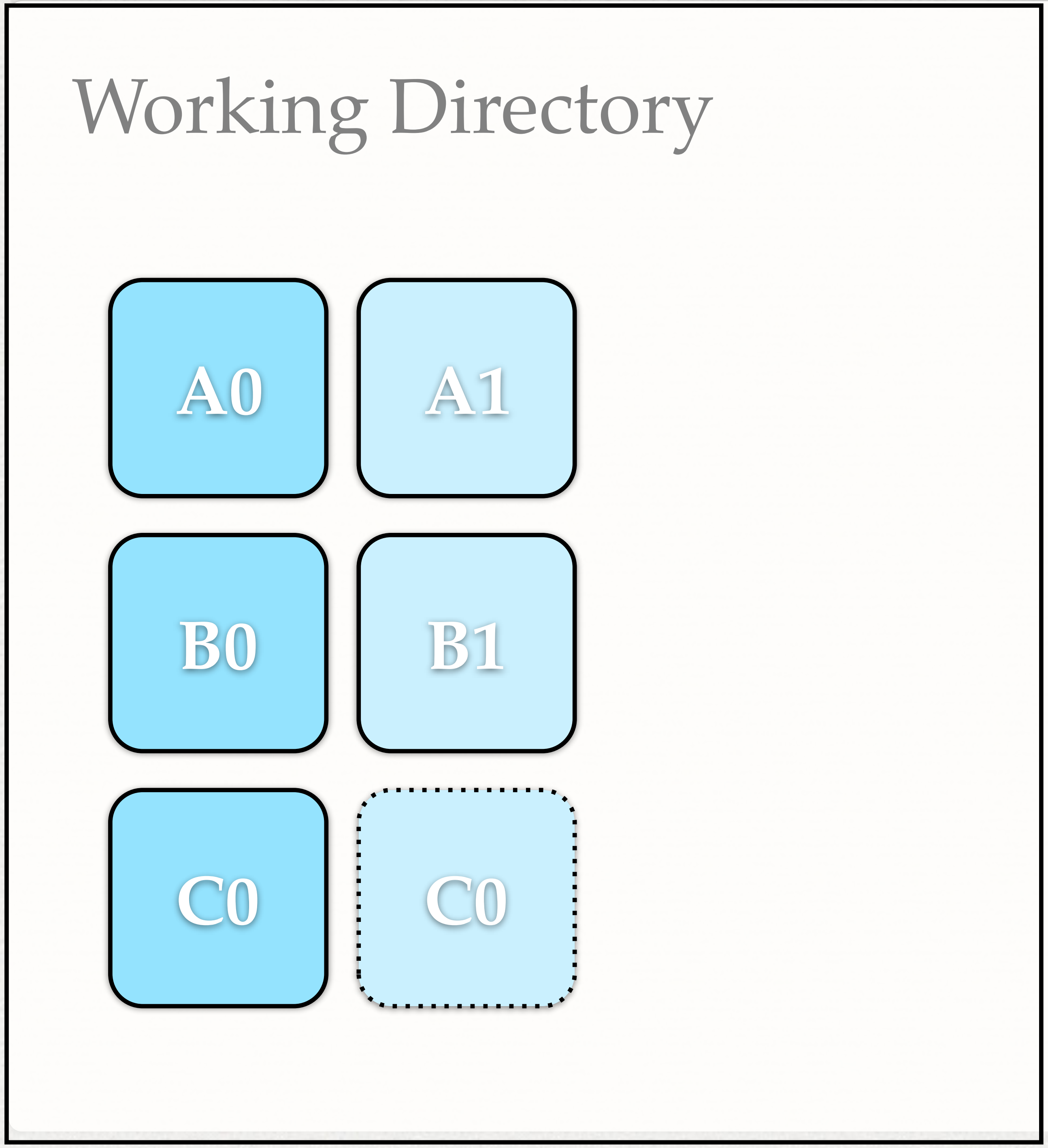
Working Directory

A0

B0

C0

STASH



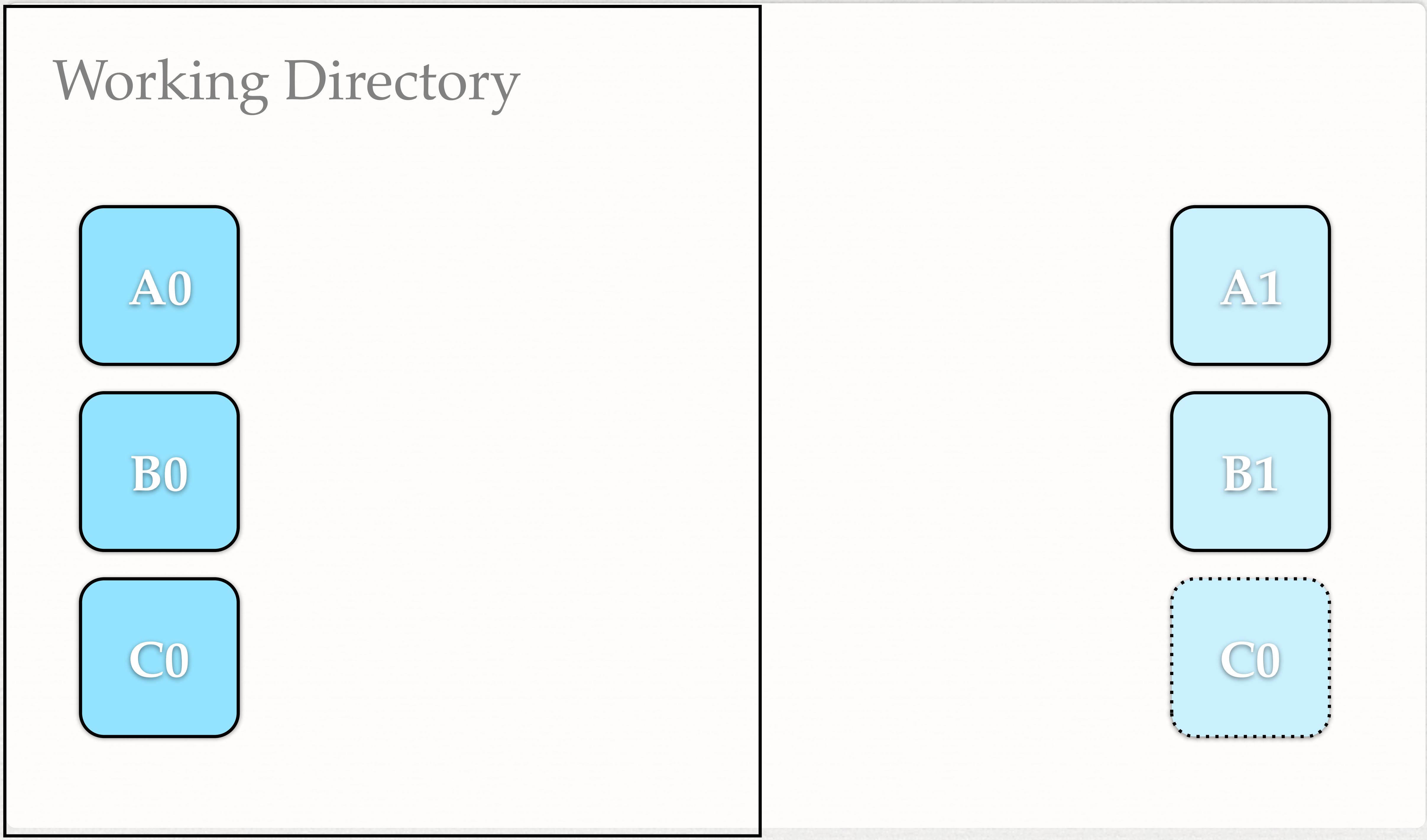
Working Directory

A0 A1

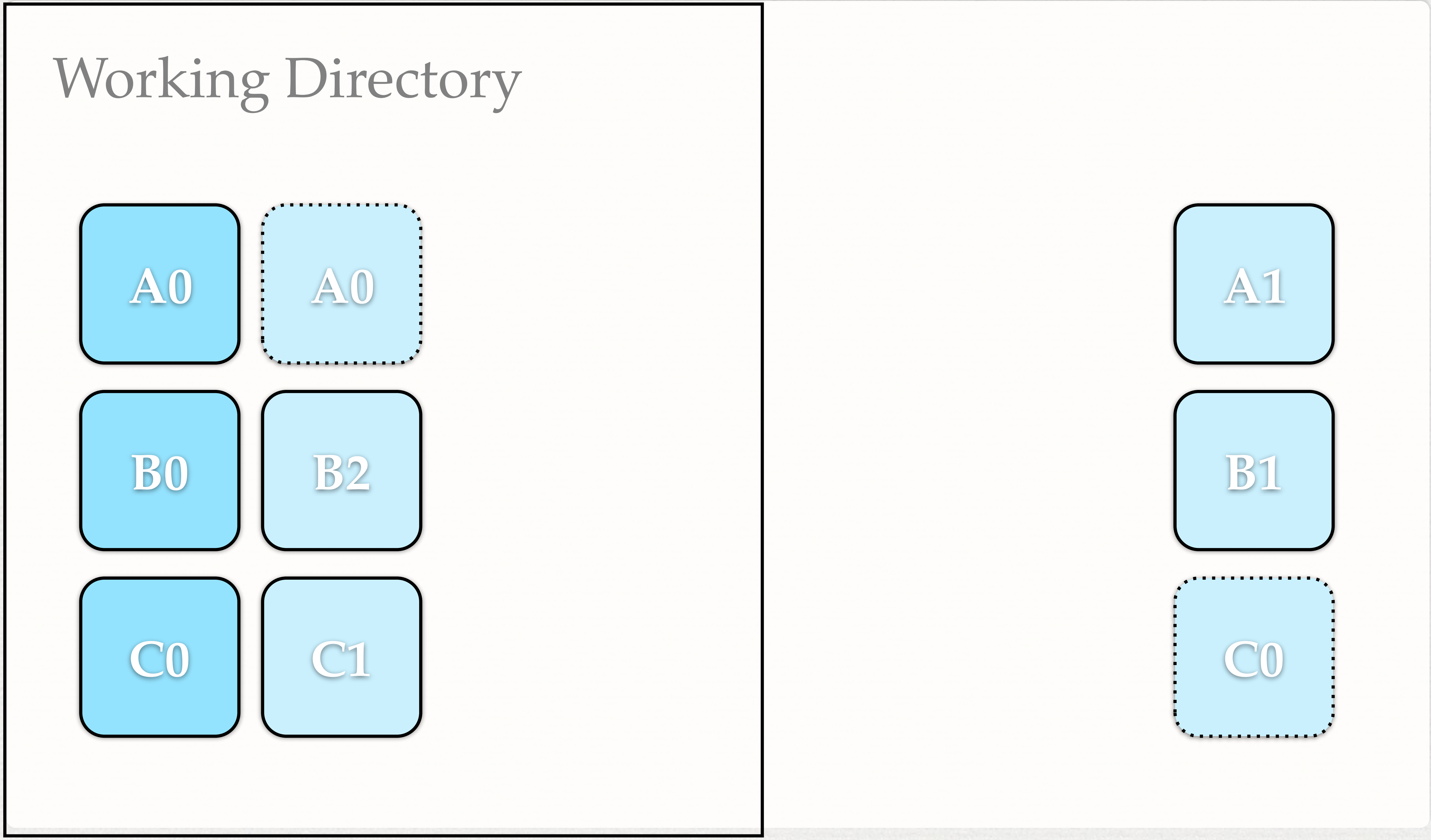
B0 B1

C0 C0

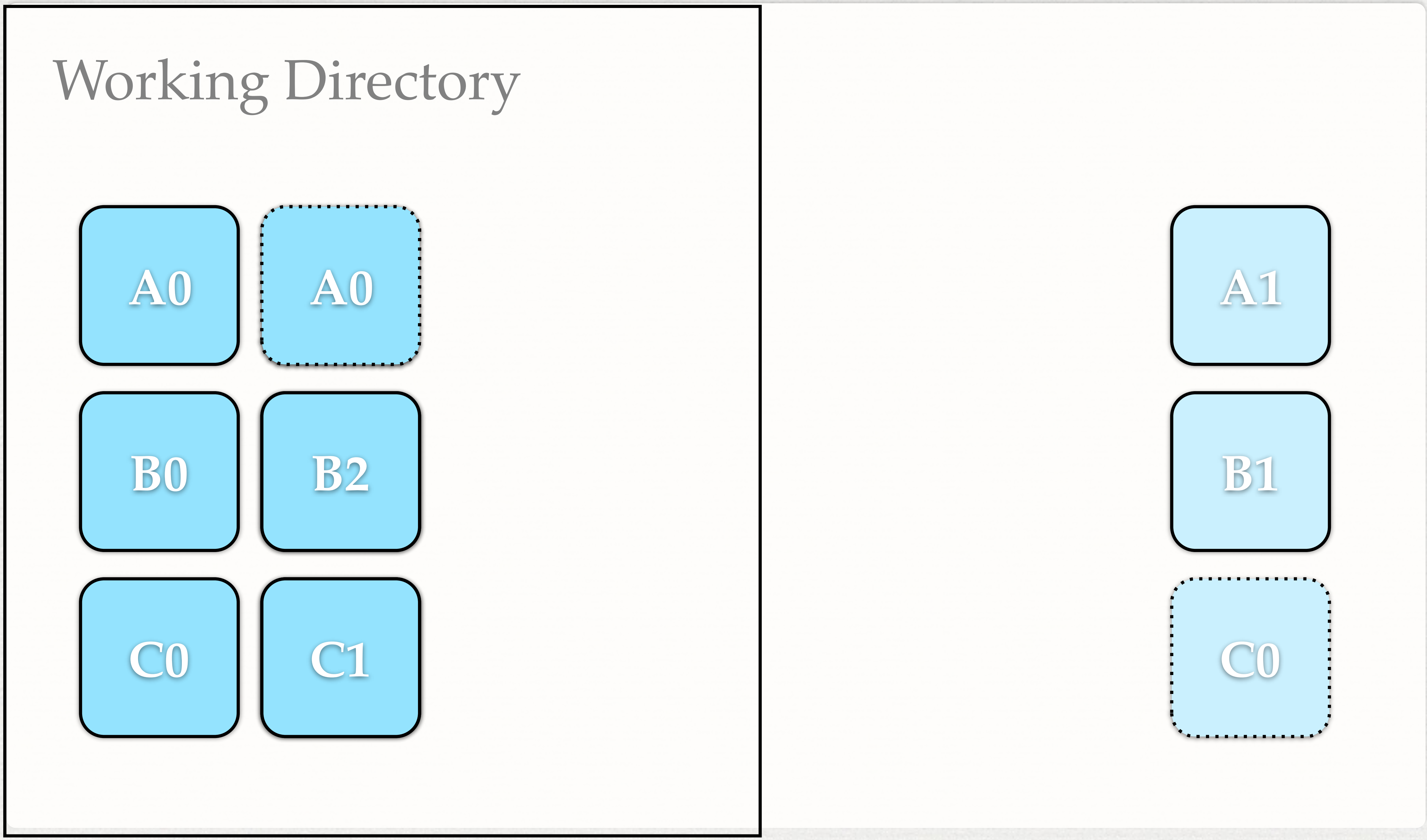
STASH



STASH



STASH



Working Directory

A0 A0

B0 B2

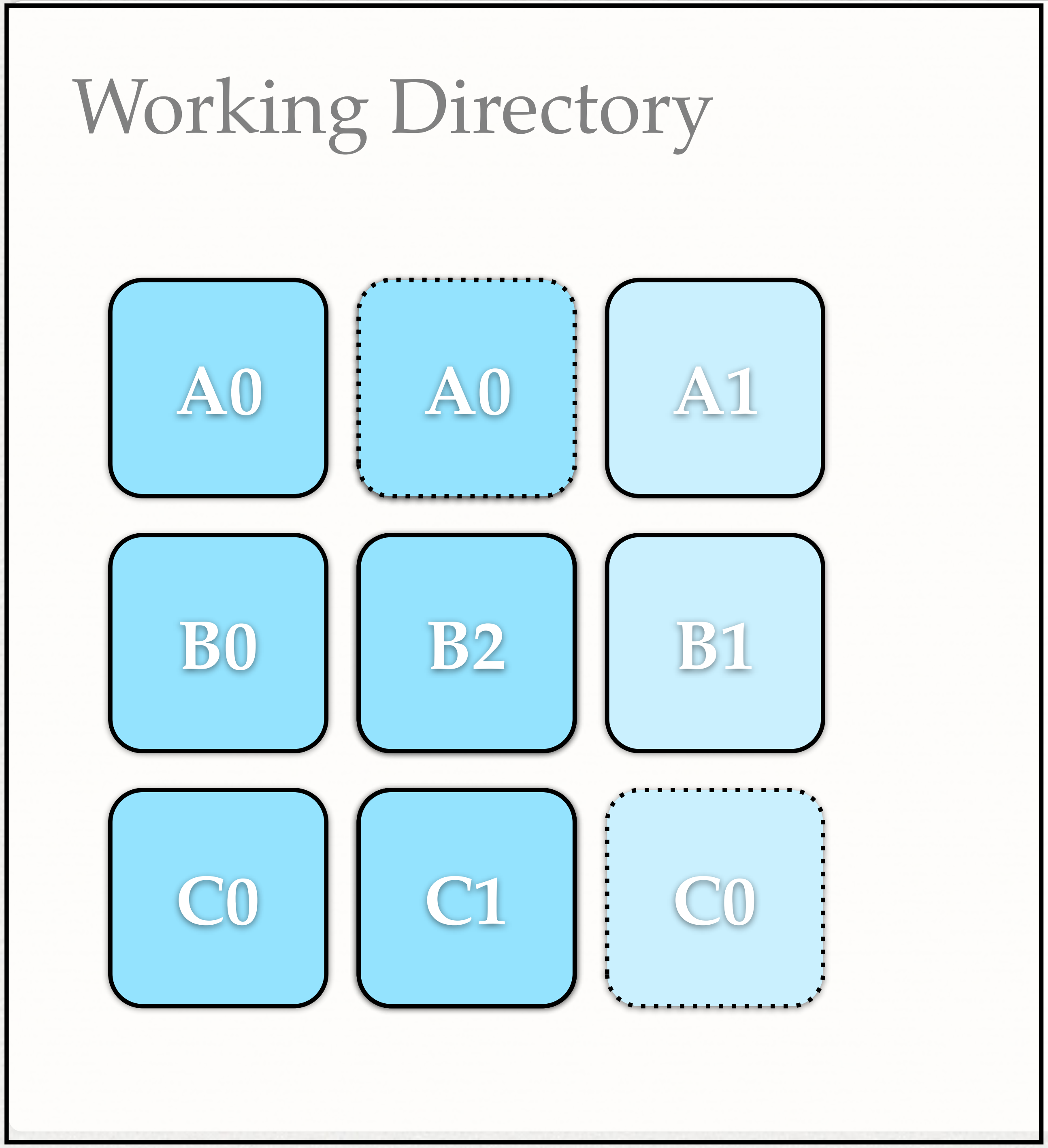
C0 C1

A1

B1

C0

STASH



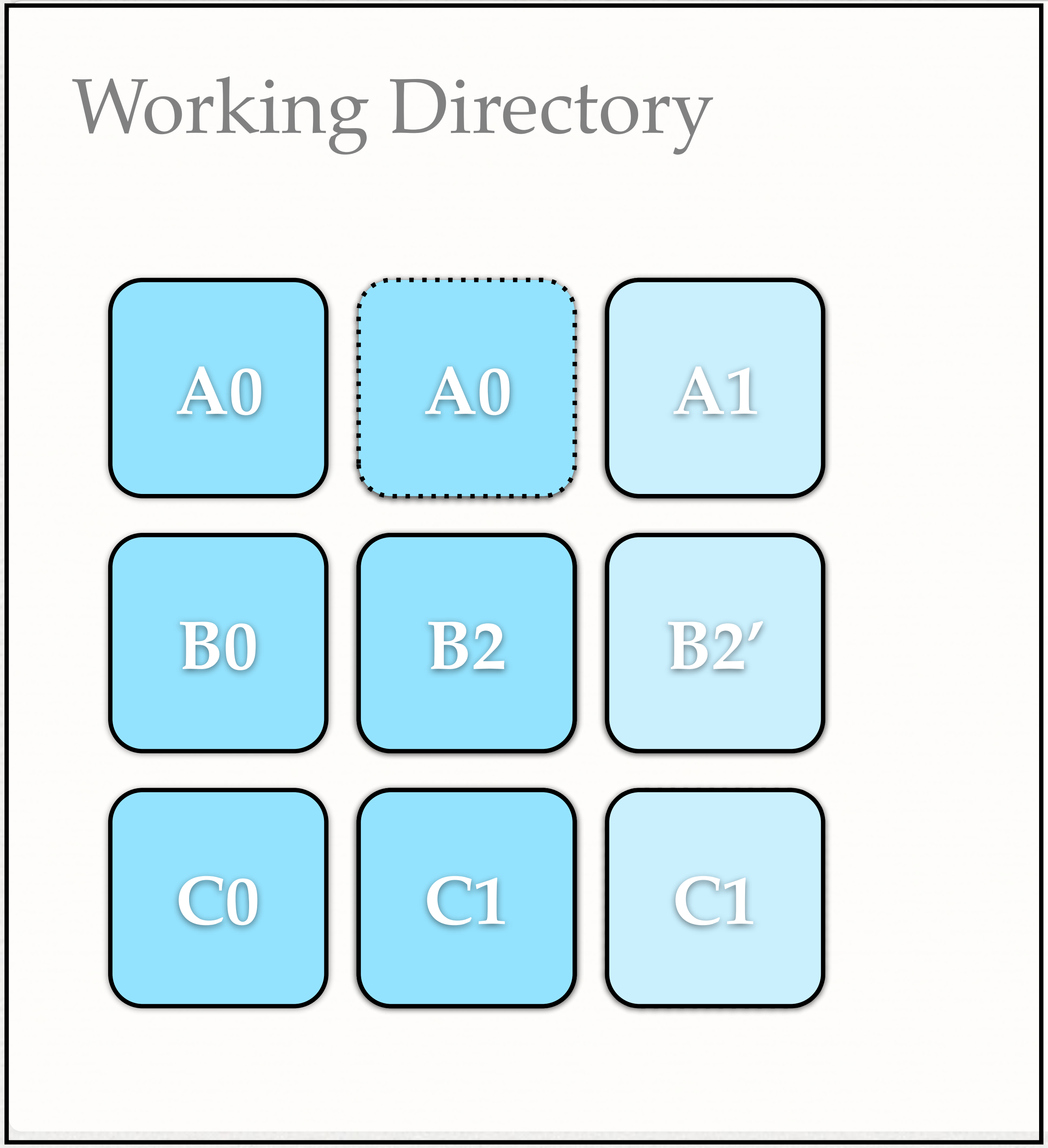
Working Directory

A0 A0 A1

B0 B2 B1

C0 C1 C0

STASH



Working Directory

A0 A0 A1

B0 B2 B2′

C0 C1 C1

BEFORE USING GIT

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

FREQUENTLY USED COMMANDS

- 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