

Git

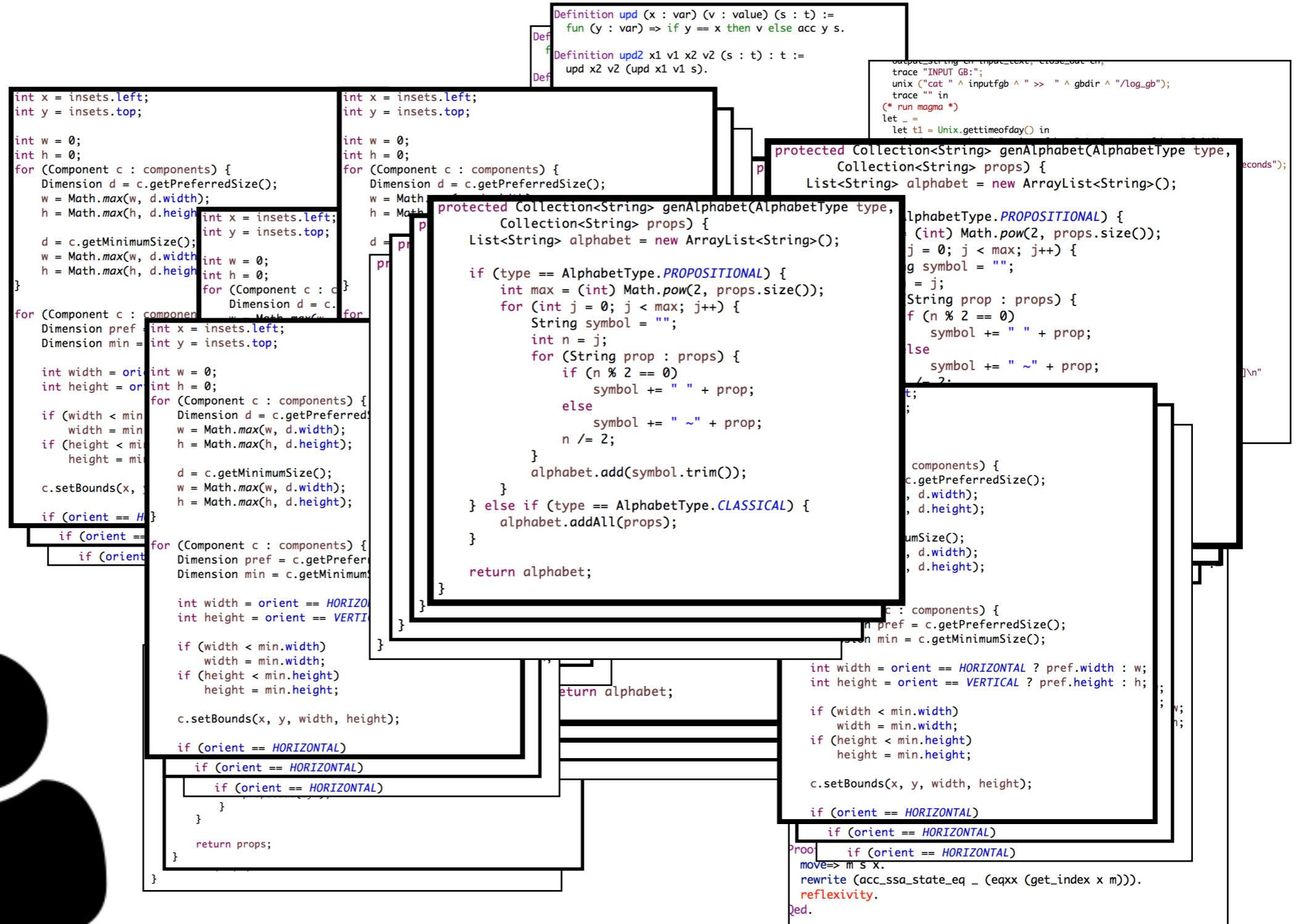
Ming-Hsien Tsai

National Institute of Cyber Security

Productivity Tools for Developers

- Editors
- Debuggers
- Build automation tools
- Integrated development environment
- Modeling tools
- Source code analyzers / testing tools
- Version control tools ← *git*

File Changes



File Changes

How to revert to a previous version?

backup



Project Backup

2017/08/05 **What has been done in a version?**

```
int w = 0;
int h = 0;
for (Component c : components) {
    Dimension d = c.getPreferredSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);
}

d = c.getPreferredSize();
w = Math.max(w, d.width);
h = Math.max(h, d.height);

int w = 0;
int h = 0;
for (Component c : components) {
    Dimension d = c.getPreferredSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);
}

protected Collection<String> genAlphabet(AlphabetType type,
    Collection<String> props) {
    List<String> alphabet = new ArrayList<String>();

    if (type == AlphabetType.PROPOSITIONAL) {
        int max = (int) Math.pow(2, props.size());
        for (int j = 0; j < max; j++) {
            String symbol = "";
            for (int i = 0; i < props.size(); i++) {
                if ((j >> i) > 0) {
                    symbol += props.get(i);
                }
            }
            alphabet.add(symbol.trim());
        }
    } else if (type == AlphabetType.CLASSICAL) {
        alphabet.addAll(props);
    }

    return alphabet;
}

let t1 = Unix.gettimeofday() in
seconds");

protected Collection<String> genAlphabet(AlphabetType type,
    Collection<String> props) {
    List<String> alphabet = new ArrayList<String>();

    if (type == AlphabetType.PROPOSITIONAL) {
        int max = (int) Math.pow(2, props.size());
        for (int j = 0; j < max; j++) {
            String symbol = "";
            for (int i = 0; i < props.size(); i++) {
                if ((j >> i) > 0) {
                    symbol += props.get(i);
                }
            }
            alphabet.add(symbol.trim());
        }
    } else if (type == AlphabetType.CLASSICAL) {
        alphabet.addAll(props);
    }

    return alphabet;
}

let t1 = Unix.gettimeofday() in
seconds");
```

How is a bug in version A fixed in version B?

```
if (height < min.height)
    height = min.height;

c.setBounds(x, y, width, height);

if (orient == HORIZONTAL)
    if (orient == HORIZONTAL)
        if (orient == HORIZONTAL)
            return props;
}

int width = orient == HORIZONTAL ? pref.width : w;
int height = orient == VERTICAL ? pref.height : h;

n /= 2;
}
alphabet.add(symbol.trim());
} else if (type == AlphabetType.CLASSICAL) {
    alphabet.addAll(props);
}

return alphabet;

components) {
    Dimension pref = c.getPreferredSize();
    Dimension min = c.getMinimumSize();

    int width = orient == HORIZONTAL ? pref.width : w;
    int height = orient == VERTICAL ? pref.height : h;

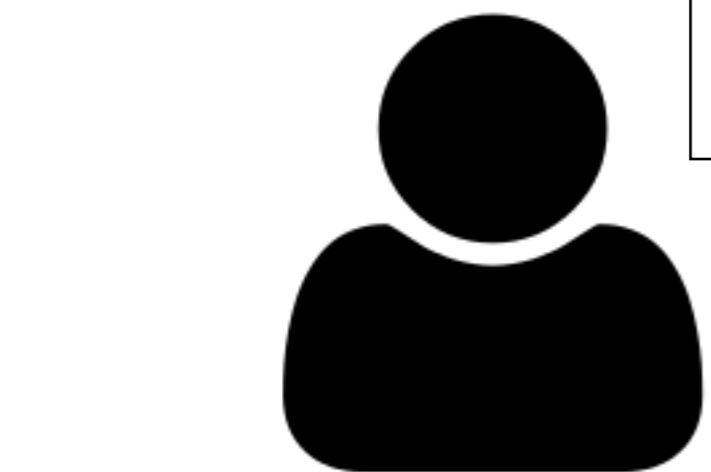
    if (height < min.height)
        height = min.height;

    c.setBounds(x, y, width, height);

    if (orient == HORIZONTAL)
        if (orient == HORIZONTAL)
            if (orient == HORIZONTAL)
                return props;
}

Proo
move=> m s x.
rewrite (acc_ssa_state_eq _ (eqxx (get_index x m))).
reflexivity.
Qed.
```

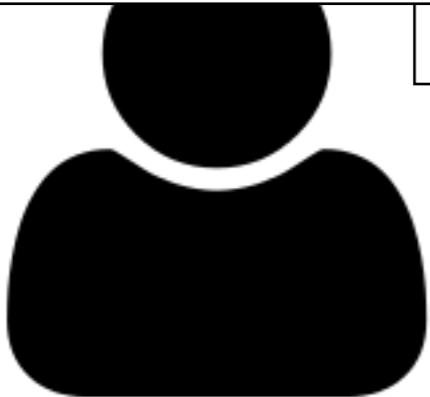
Merge two versions?



Project Backup

201 **What has been done in a version?**

Version Control Systems (VCS)





Merge two versions:

```
if (orient == HORIZONTAL)
  if (orient == HORIZONTAL)
    if (orient == HORIZONTAL)
      }
    }
  }
  return props;
}
```

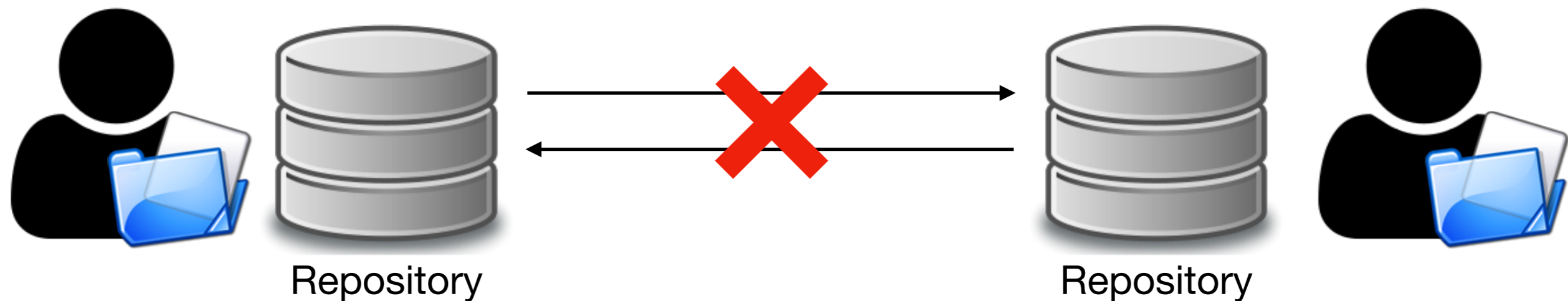
```
n.width)
n.width;
if (height < min.height)
  height = min.height;
c.setBounds(x, y, width, height);
if (orient == HORIZONTAL)
  if (orient == HORIZONTAL)
    if (orient == HORIZONTAL)
      Proo if (orient == HORIZONTAL)
      move=> m s x.
      rewrite (acc_ssa_state_eq _ (eqxx (get_index x m))).
      reflexivity.
Qed.
```

Version Control System

- The developer needs a system that can
 - store files in different versions with log messages,
 - compare files between different versions,
 - restore files from specific versions,
 - ...
- All versions are stored in a **repository**  
- A version is checked out to a **working area** for editing

Local Version Control

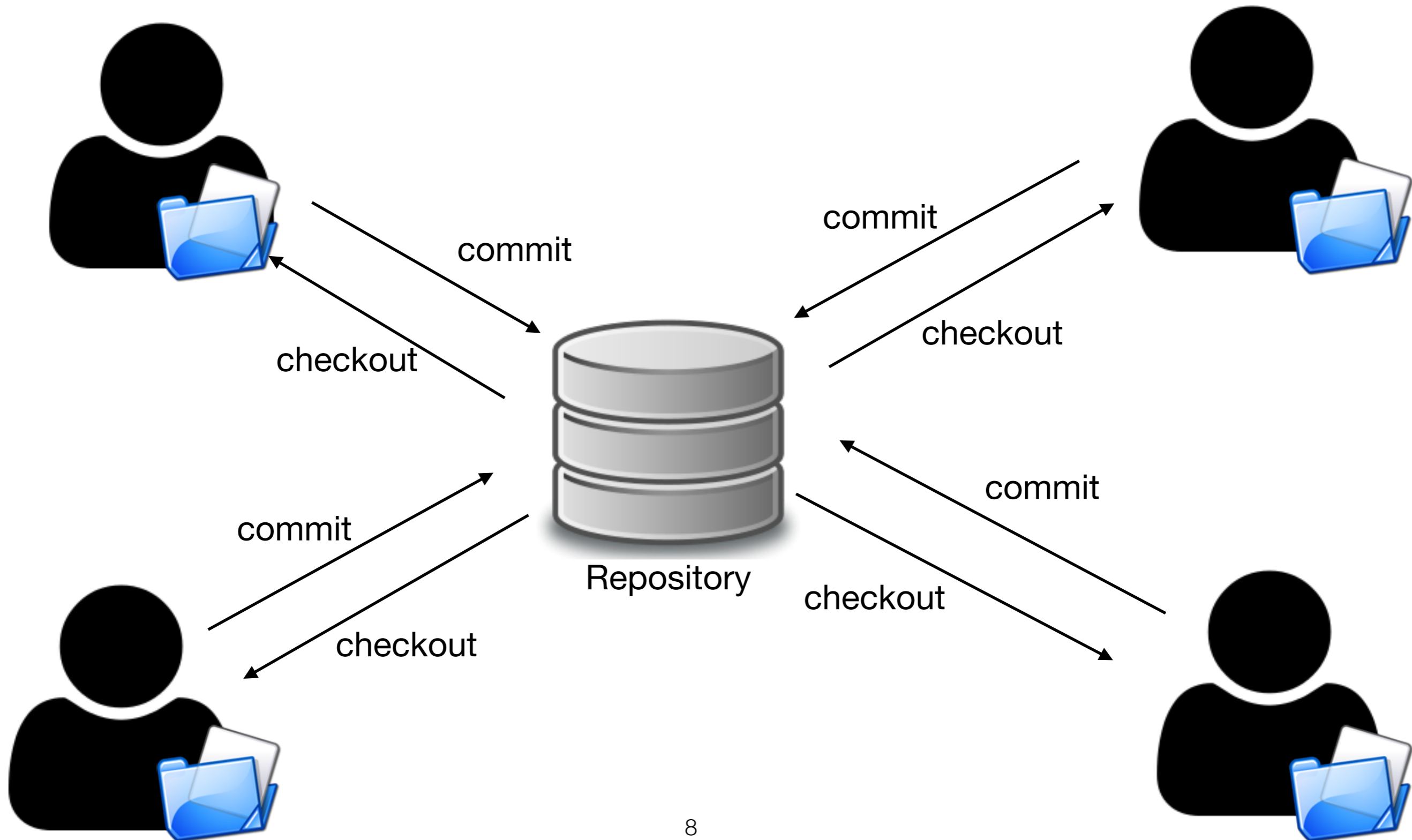
- SCCS (Source Code Control System, 1972)
- RCS (Revision Control System, 1982)



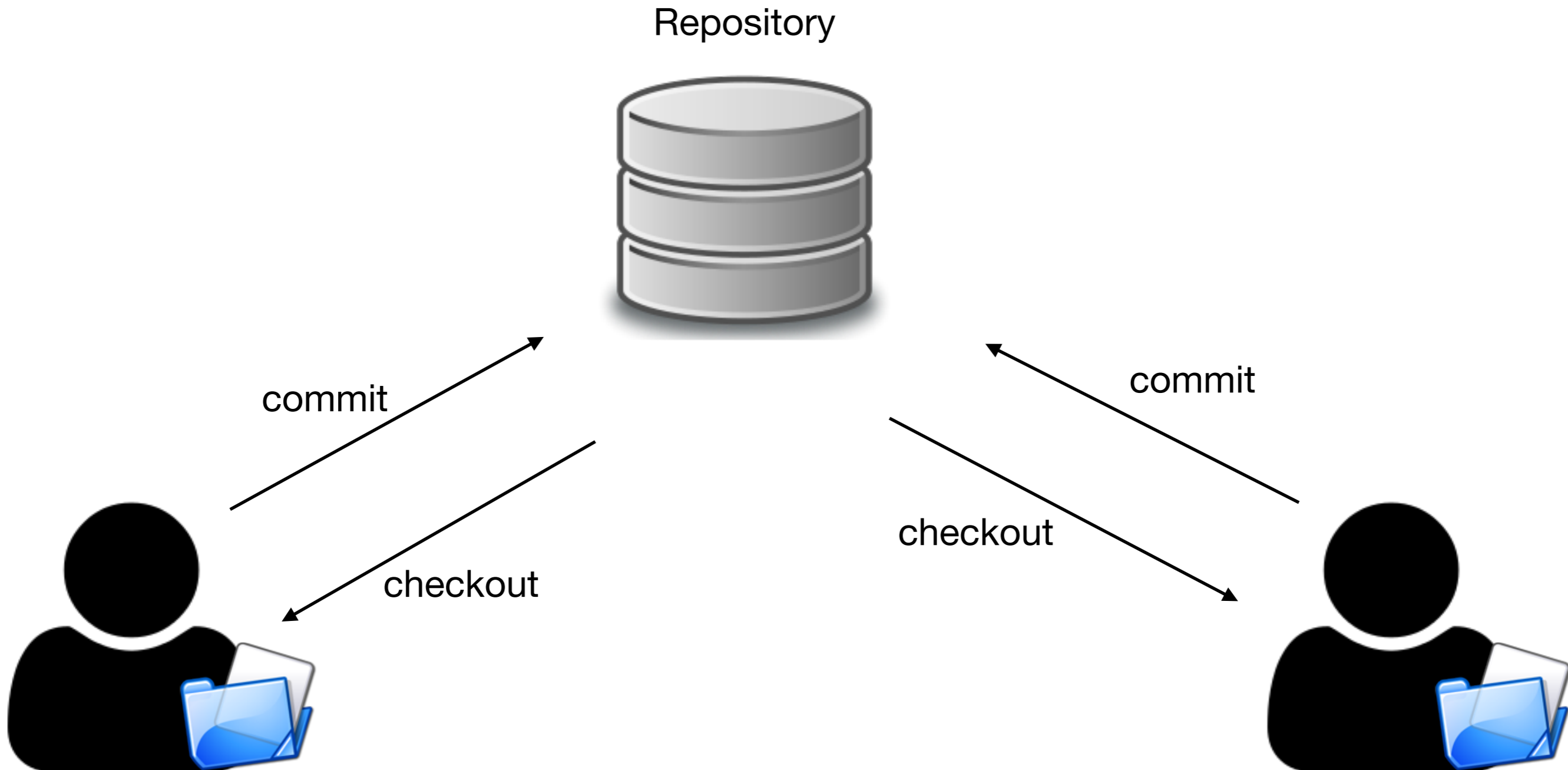
Centralized Version Control

- CVS (Concurrent Versions System, 1990)
- SVN (Subversion, 2000)
- ClearCase
- Visual SourceSafe

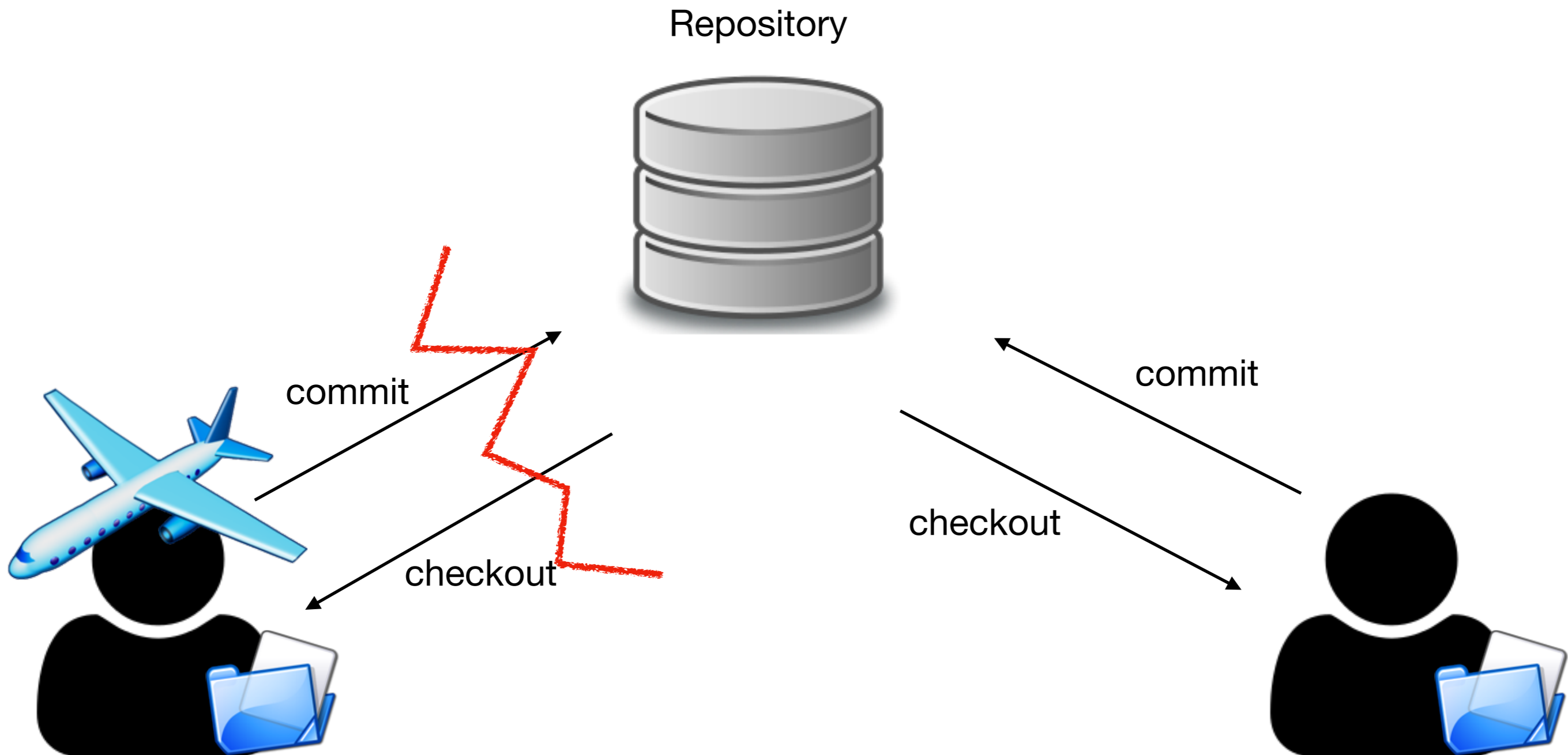
Multiple Developers



Problems



Problems

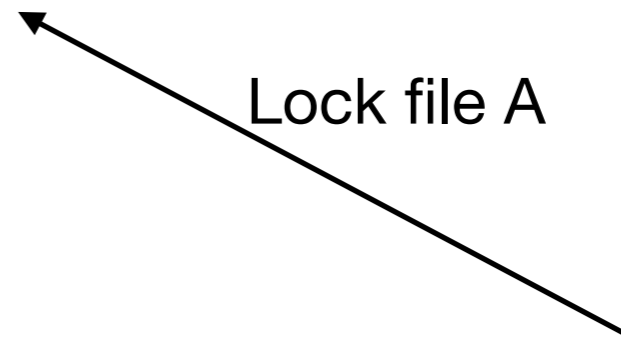


Problems

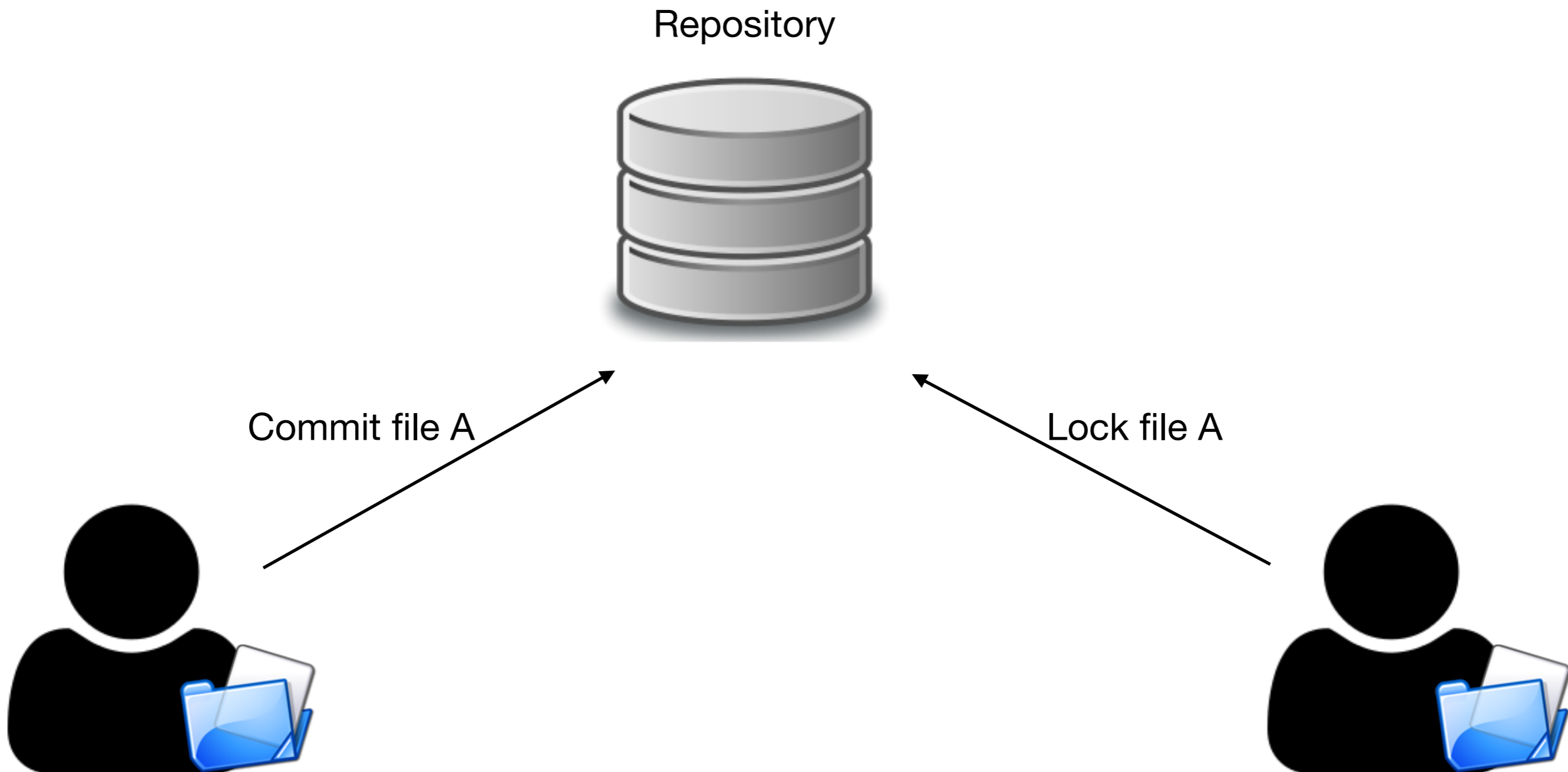
Repository



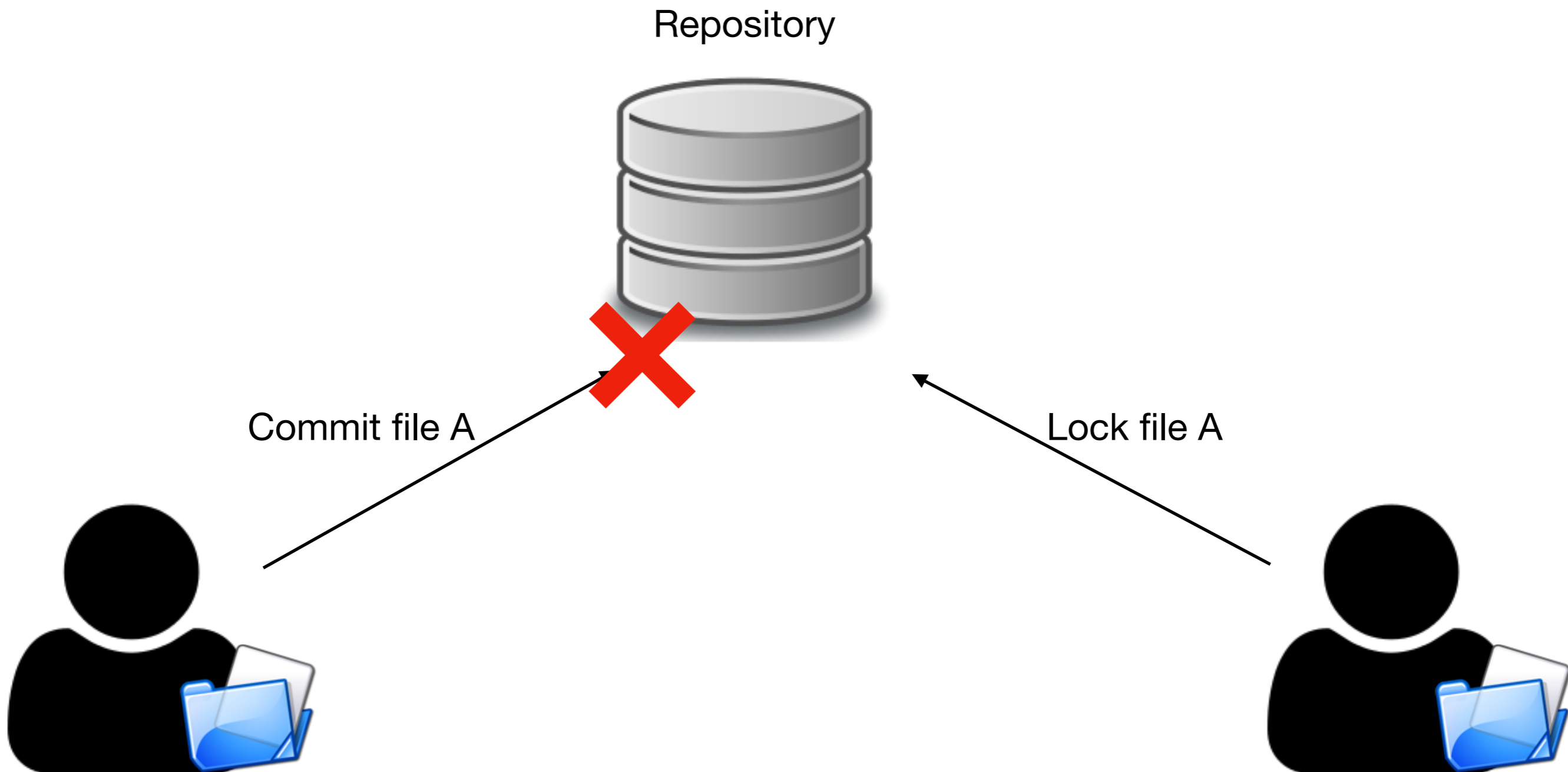
Lock file A



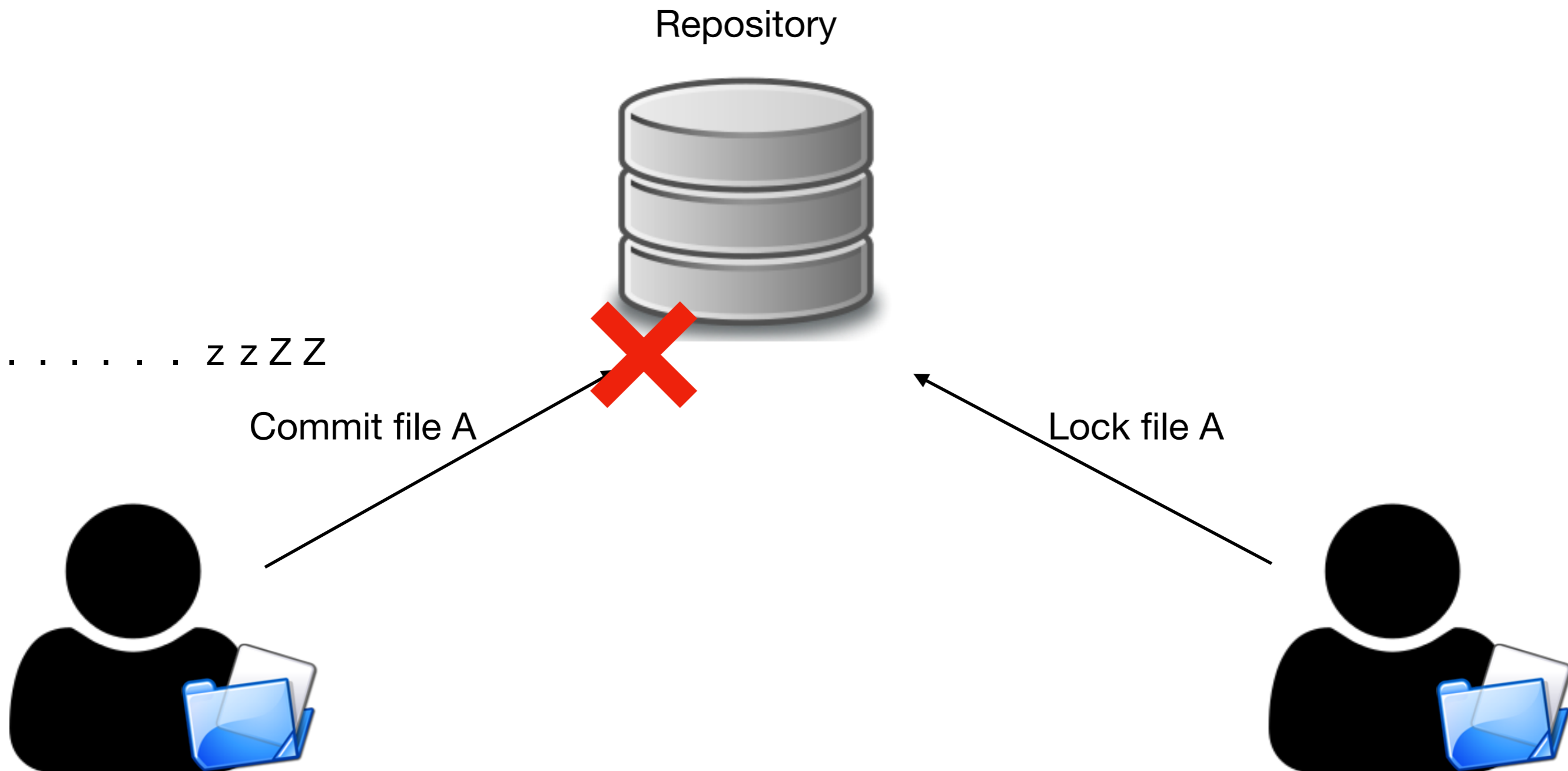
Problems



Problems



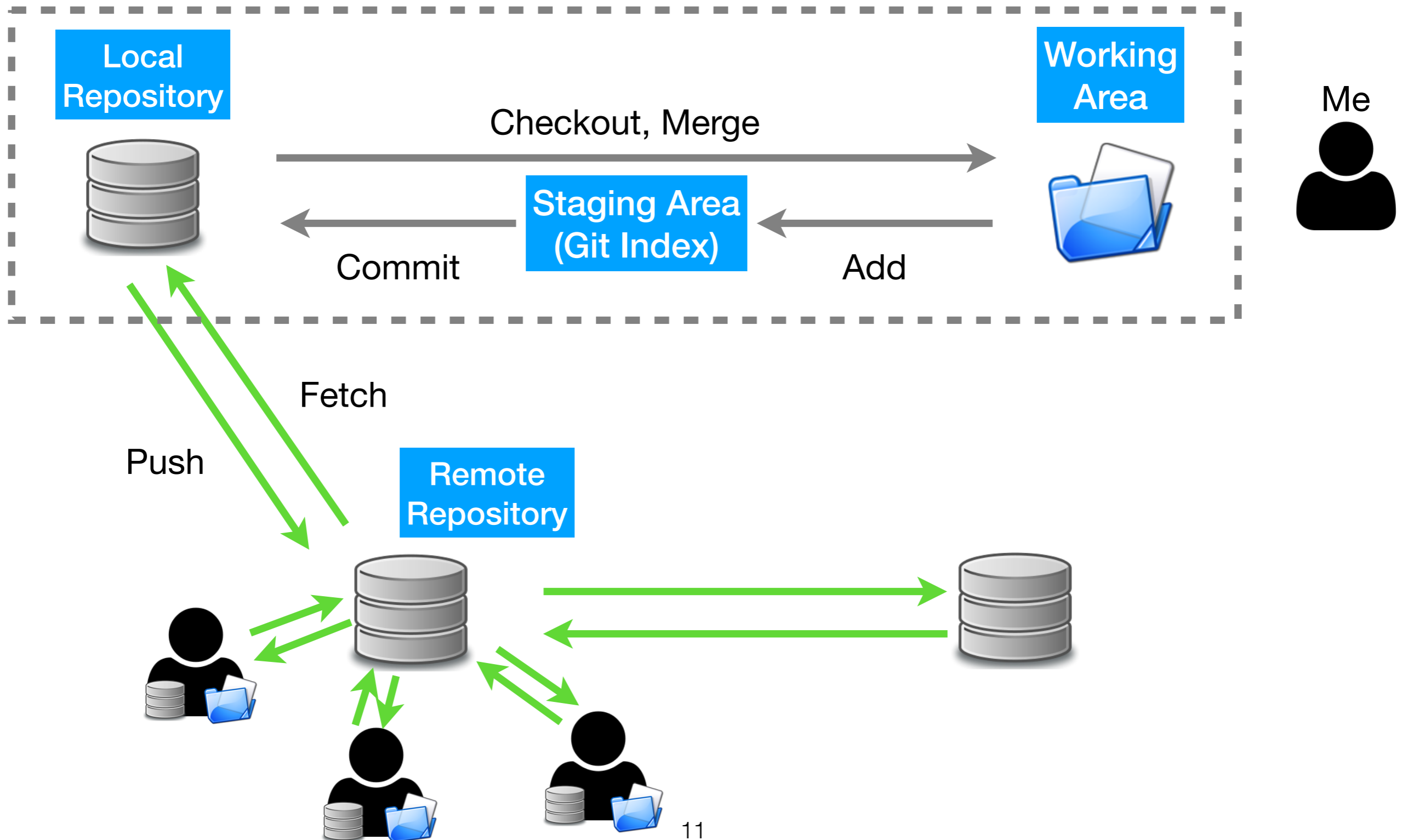
Problems



Distributed Version Control

- BitKeeper
- Git (2005)
- Mercurial (2005)

Distributed Approach (Git)



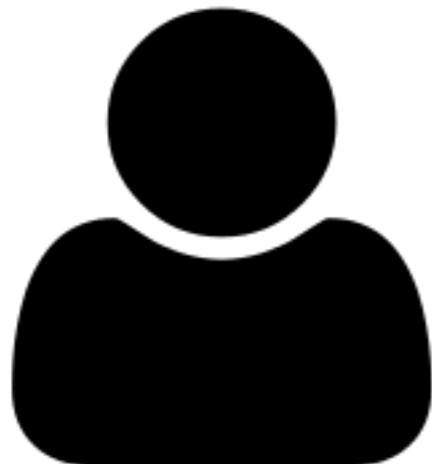
Create a Repository

```
DHCP-22126 project $ git init
Initialized empty Git repository in /Users/mht208/project/.git/
DHCP-22126 project $ ls
DHCP-22126 project $ ls -a
.  ..  .git
DHCP-22126 project $
```

repository

working area (working tree)

Files in the “project” folder can be added to the repository



Bare Repository

```
~/project
```

```
▶ git init --bare --shared
```

```
Initialized empty shared Git repository in /Users/mht208/project/
```

```
~/project master ✓
```

```
▶ ls -a
```

.	HEAD	description	info	refs
..	config	hooks	objects	

```
~/project master ✓
```

```
▶ █
```

no working area in a bare repository

Clone



Clone

Supported protocols: ssh, git, http, and https

```
DHCP-22126 demo $ ls
DHCP-22126 demo $ git clone https://github.com/atom/atom.git
Cloning into 'atom'...
remote: Counting objects: 179291, done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 179291 (delta 13), reused 13 (delta 10), pack-reused 179265
Receiving objects: 100% (179291/179291), 285.63 MiB | 4.65 MiB/s, done.
Resolving deltas: 100% (127760/127760), done.
DHCP-22126 demo $ ls
atom
DHCP-22126 demo $ █
```



Add and Commit

```
DHCP-22126 project $ ls -a
.          ..          .git      README.md
DHCP-22126 project $ █
```



```
/** Convert a DSL state to an SSA state. */
definiton ssa_state (m : vmap) (s : State.t) : bv64SSA.State.t :=
  fun m =>
    if (sidx v) == get_index (svor v) m
    then State.acc (svor v) s
    else State.acc (svor v) State.empty,
    ...

def ssa_state_equiv :
  forall m s, state_equiv m s (ssa_state m s).
  proof.
  move=> m s x.
  rewrite (acc_ssa_state_eq _ (get_index x m)).
  reflexivity.
end.
```

README.md



Add and Commit

```
DHCP-22126 project $ ls -a
.          ..          .git      README.md
DHCP-22126 project $ git add README.md
DHCP-22126 project $ git commit
```



```
/* Convert a DSL state to an SSA state. */
definition ssa_state (m : mmap) (s : State.t) : bv64SSA.State.t :=
  fun m =>
    if (sidx v) == get_index (svor v) m
    then State.acc (svor v) s
    else State.acc (svor v) State.empty,
    ...

lemma ssa_state_equiv :
  forall m s, state_equiv m s (ssa_state m s).
  proof.
  move=> m s x.
  rewrite (acc_ssa_state_eq _ (get_index x m)).
  reflexivity.
end.
```

README.md



Add and Commit

```
2. git
GNU nano 2.8.5 File: /Users/mht208/project/.git/COMMIT_EDITMSG Modified
Add README.md.
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch
#
# Initial commit
#
# Changes to be committed:
#   new file:   README.md
#
[ Read 10 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify    ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```



README.md

```
Convert a DSL state to an SSA state. *)
definition ssa_state (m : vmmap) (s : State.t) : bv64SSA.State.t :=
  fun m =>
    if (sidx v) == get_index (svor v) m
    then State.acc (svor v) s
    else State.acc (svor v) state.empty.
...
def ssa_state_equiv :
  forall m s, state_equiv m s (ssa_state m s).
proof.
move m s x.
rewrite (acc_ssa_state_eq _ (eqxx (get_index x m))).
reflexivity.
end.
```



Add and Commit

```
2. git
DHCP-22126 project $ git add README.md
DHCP-22126 project $ git commit
[master (root-commit) d05fe5d] Add README.md.
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
DHCP-22126 project $ █
```

```
[ Read 10 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```



README.md

```
/* Convert a DSL state to an SSA state. */
definition ssa_state (m : vmap) (s : State.t) : bv64SSA.State.t :=
  fun m =>
  if (sidx v) == get_index (svor v) m
  then State.acc (svor v) s
  else State.acc (svor v) state.empty,
  ...

def ssa_state_equiv :
  forall m s, state_equiv m s (ssa_state m s).
  proof.
  move m s x.
  rewrite (acc_ssa_state_eq _ (get_index x m)).
  reflexivity.
end.
```

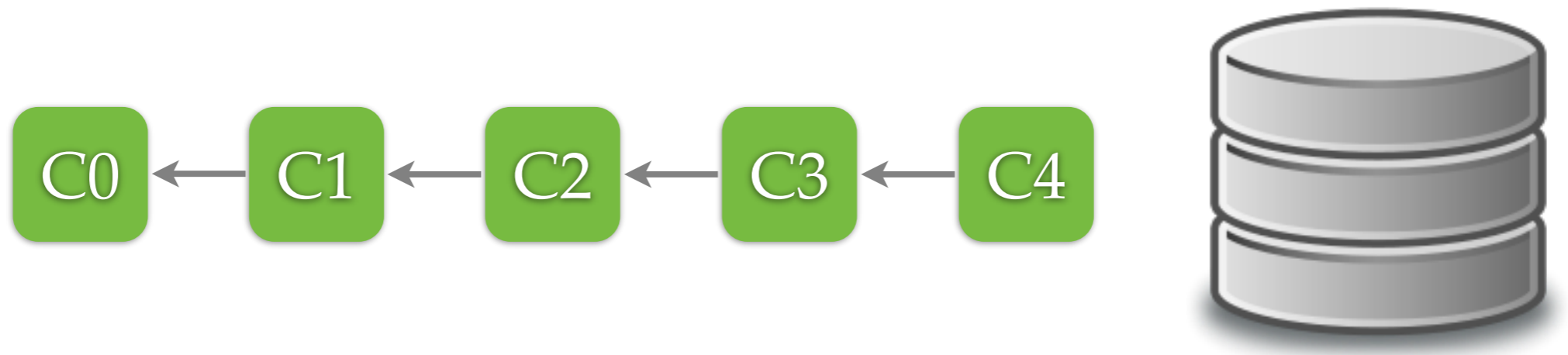


Add VS Commit

- git add
 - Add files to the staging area
 - The files may be new or modified
- git commit
 - Add files in the staging area to the repository as a new version

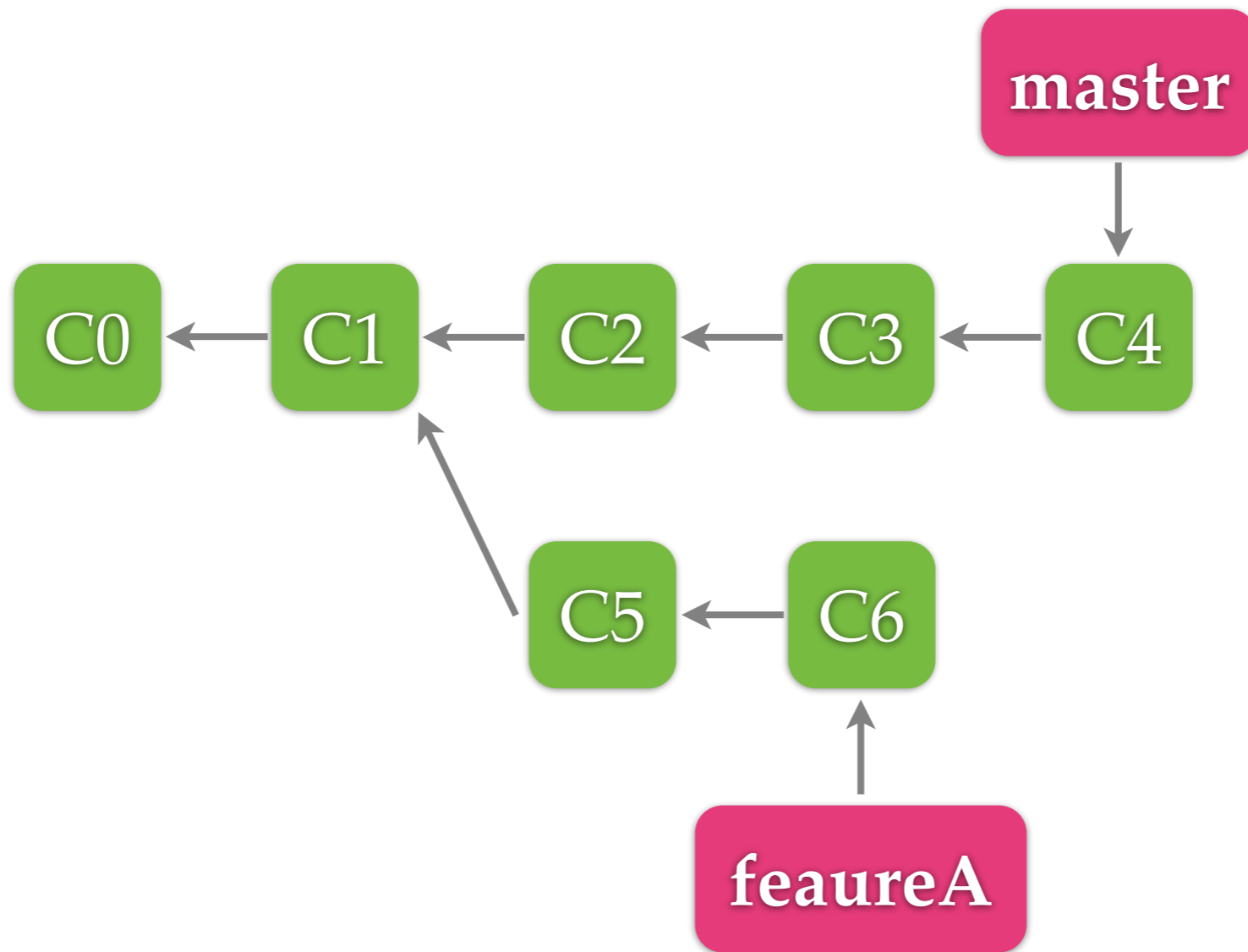


Add More Commits



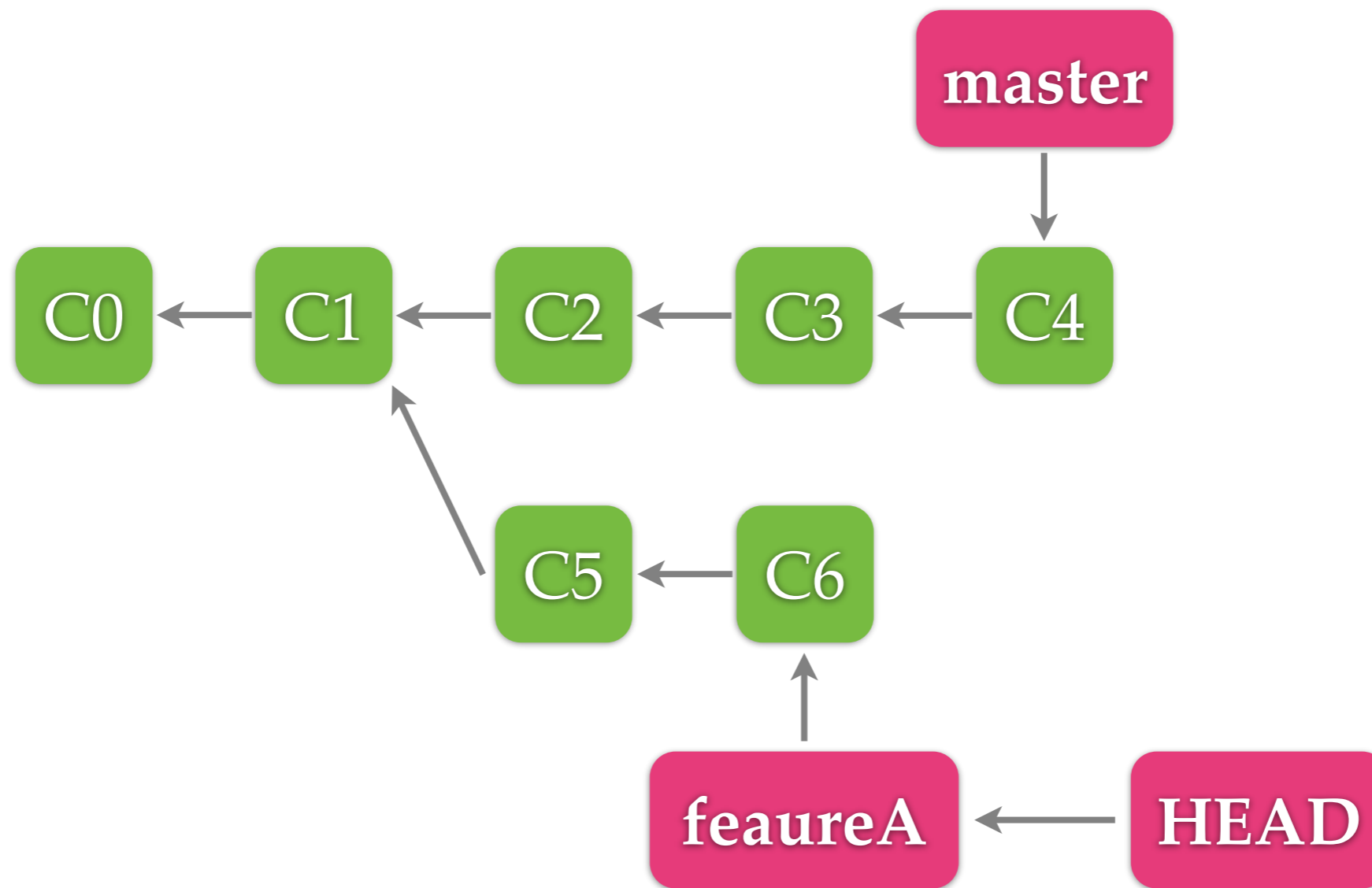
A commit has pointers back to its parents

Branches



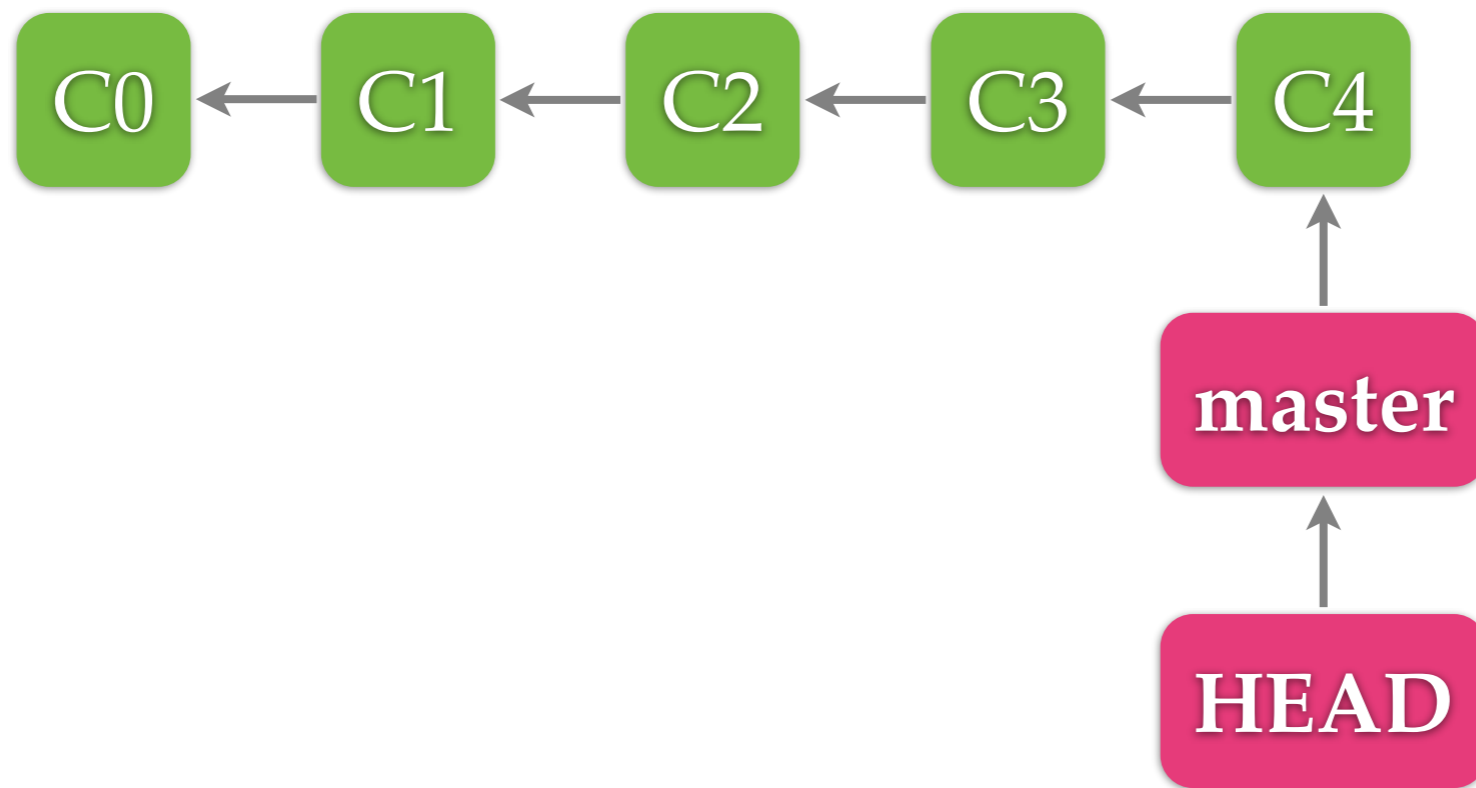
A branch is used to track a line of development
master (main) is the default branch

Head



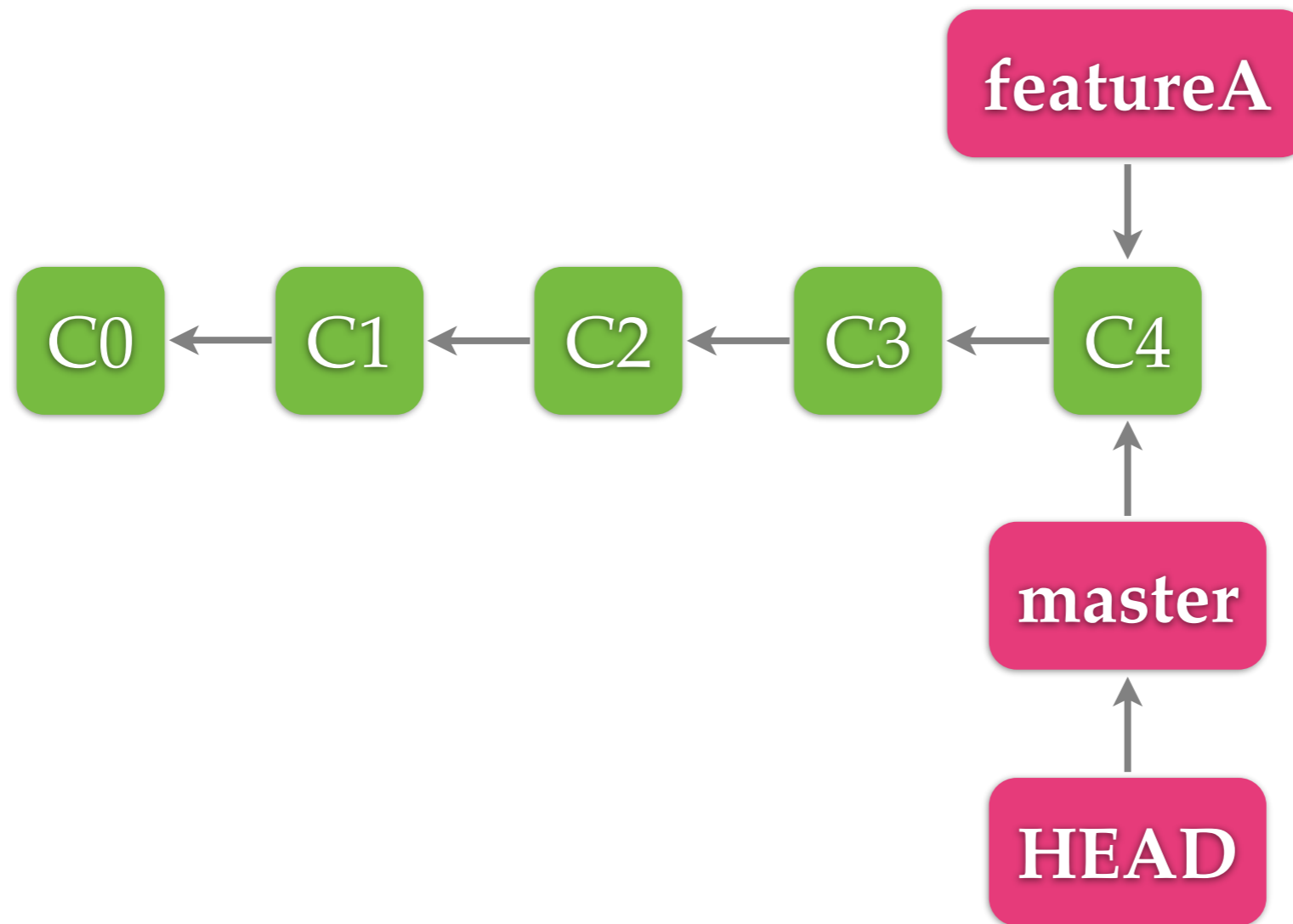
HEAD points to the branch or the commit of the working area

Create Branches



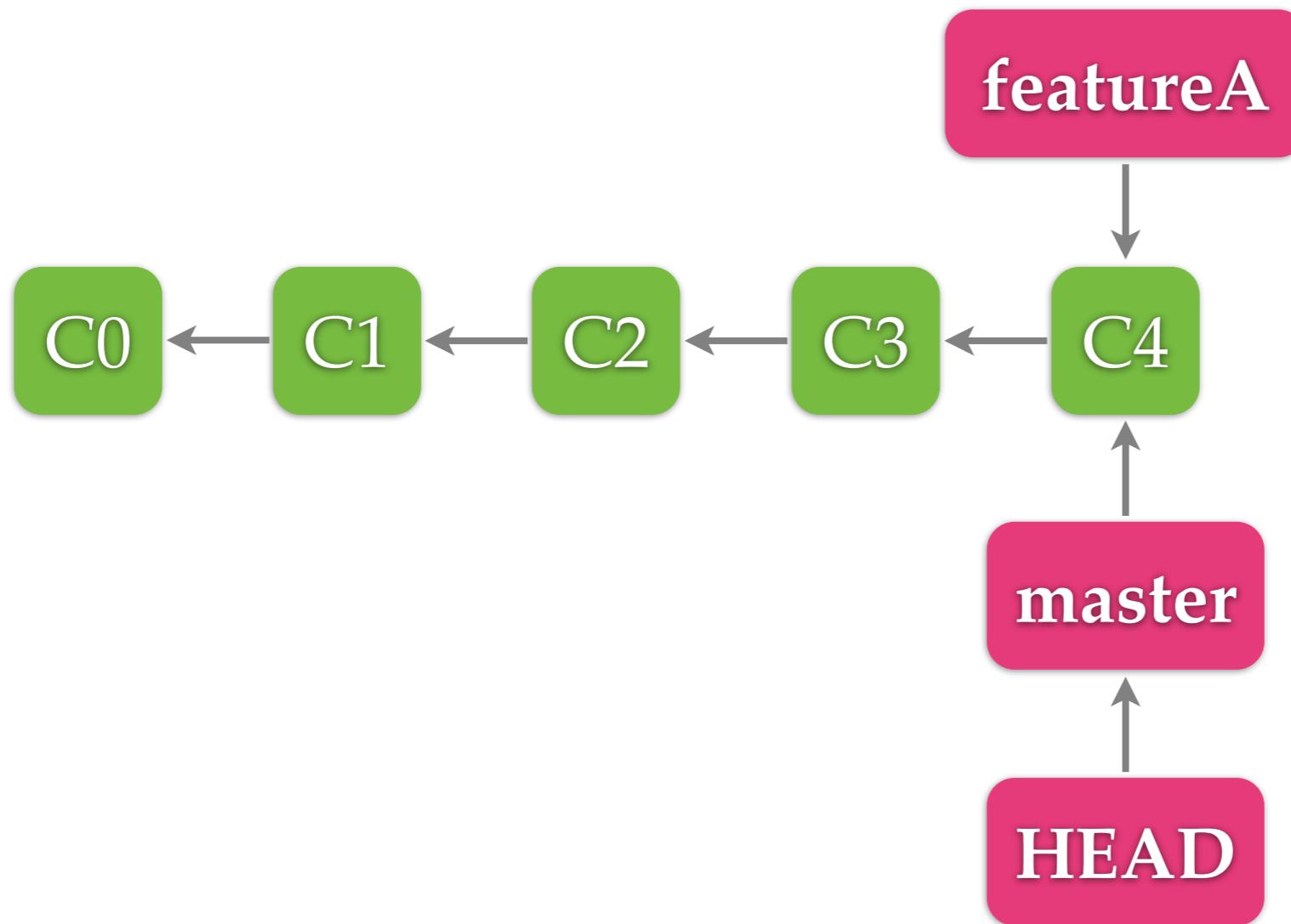
\$ git branch featureA

Create Branches



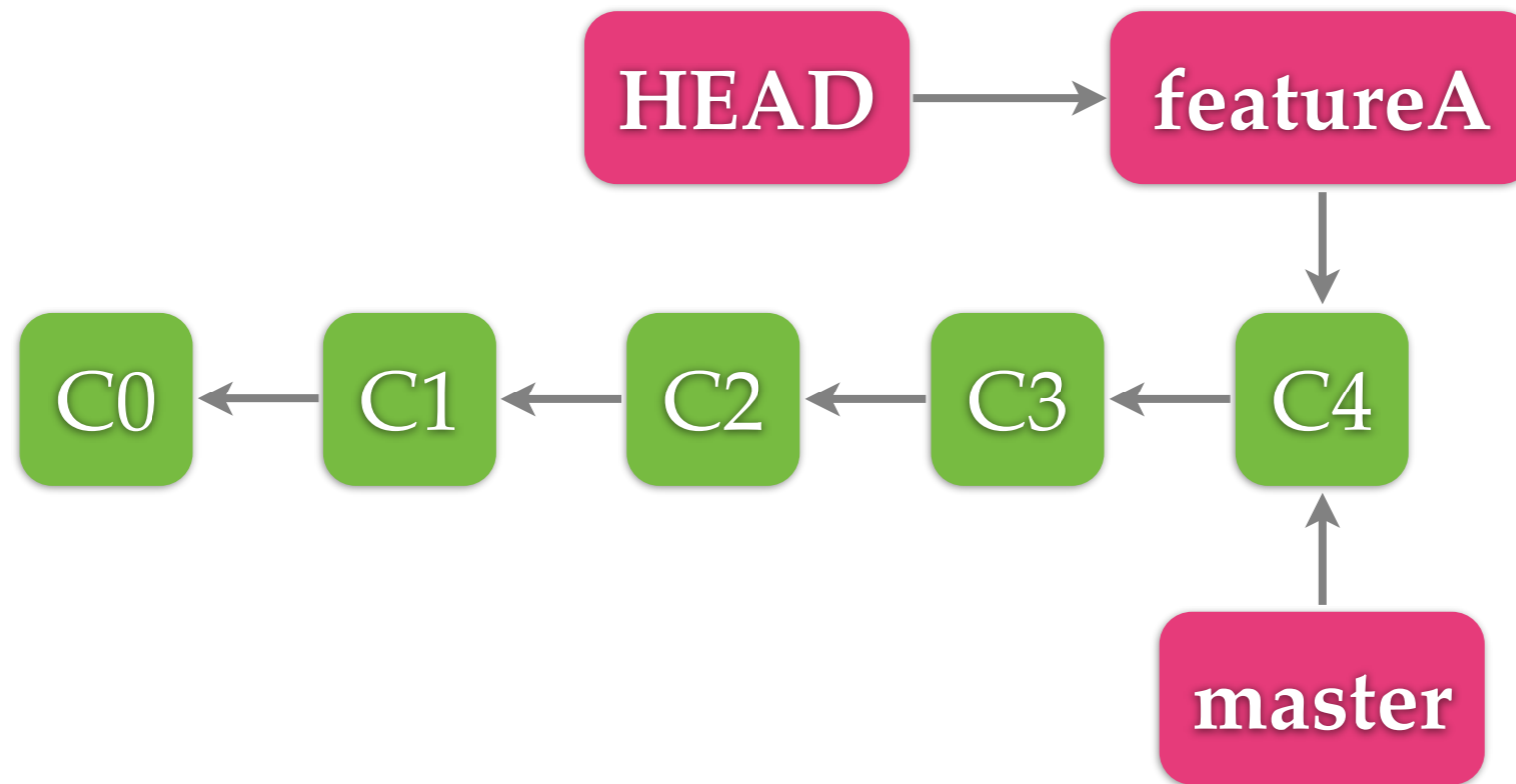
➔ \$ git branch featureA

Switch Branches



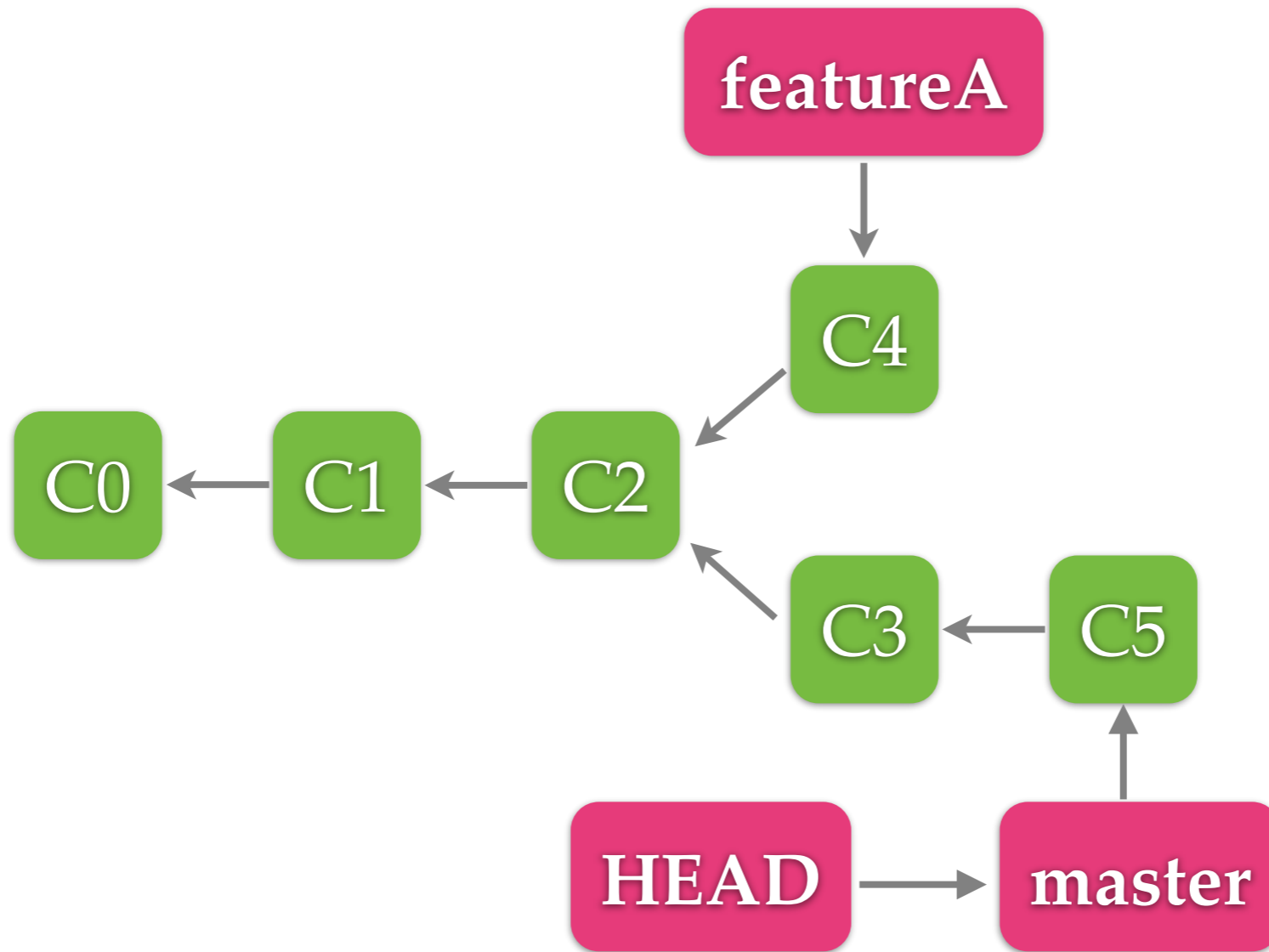
\$ git checkout featureA

Switch Branches



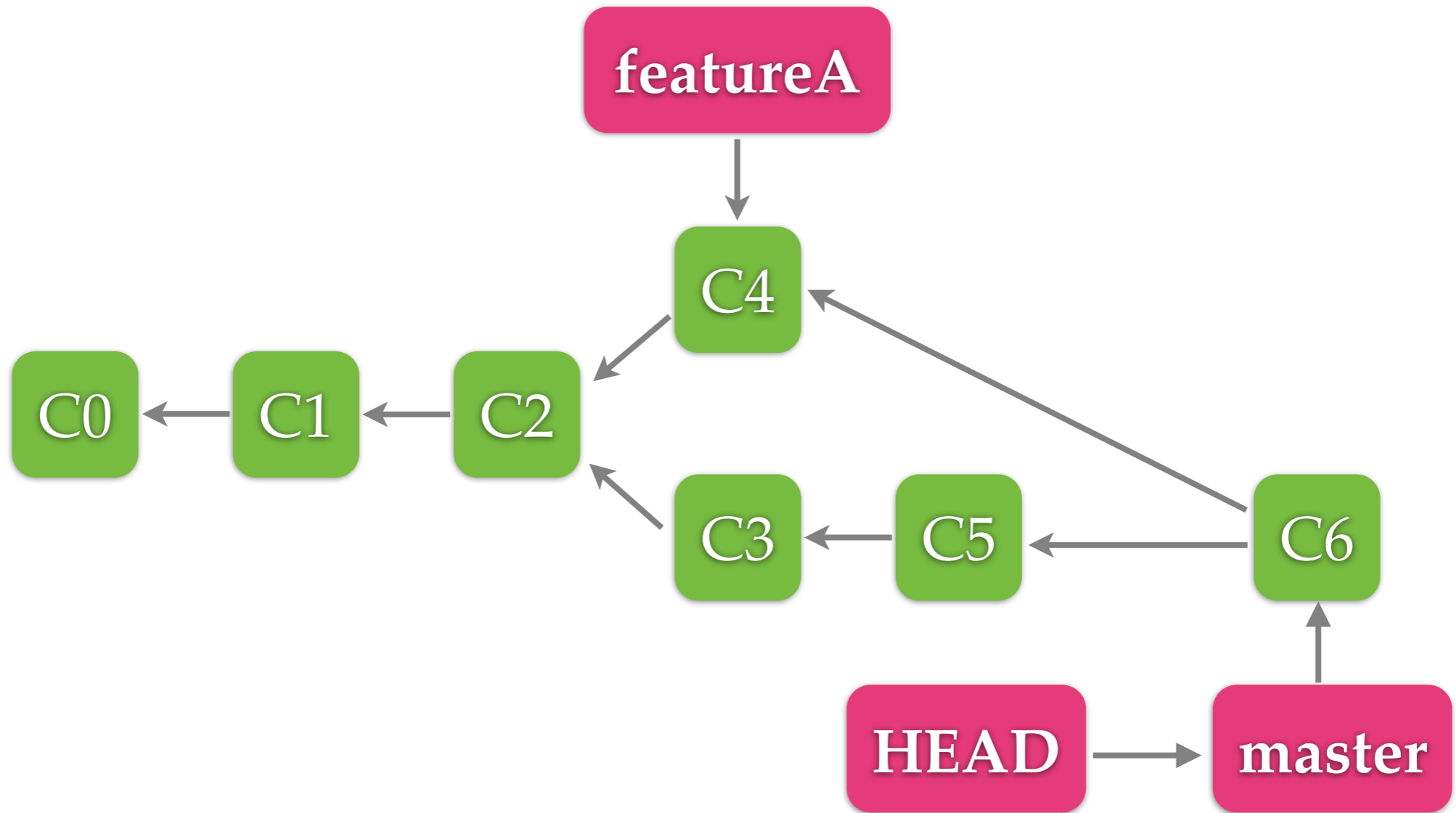
➔ \$ git checkout featureA

Merge



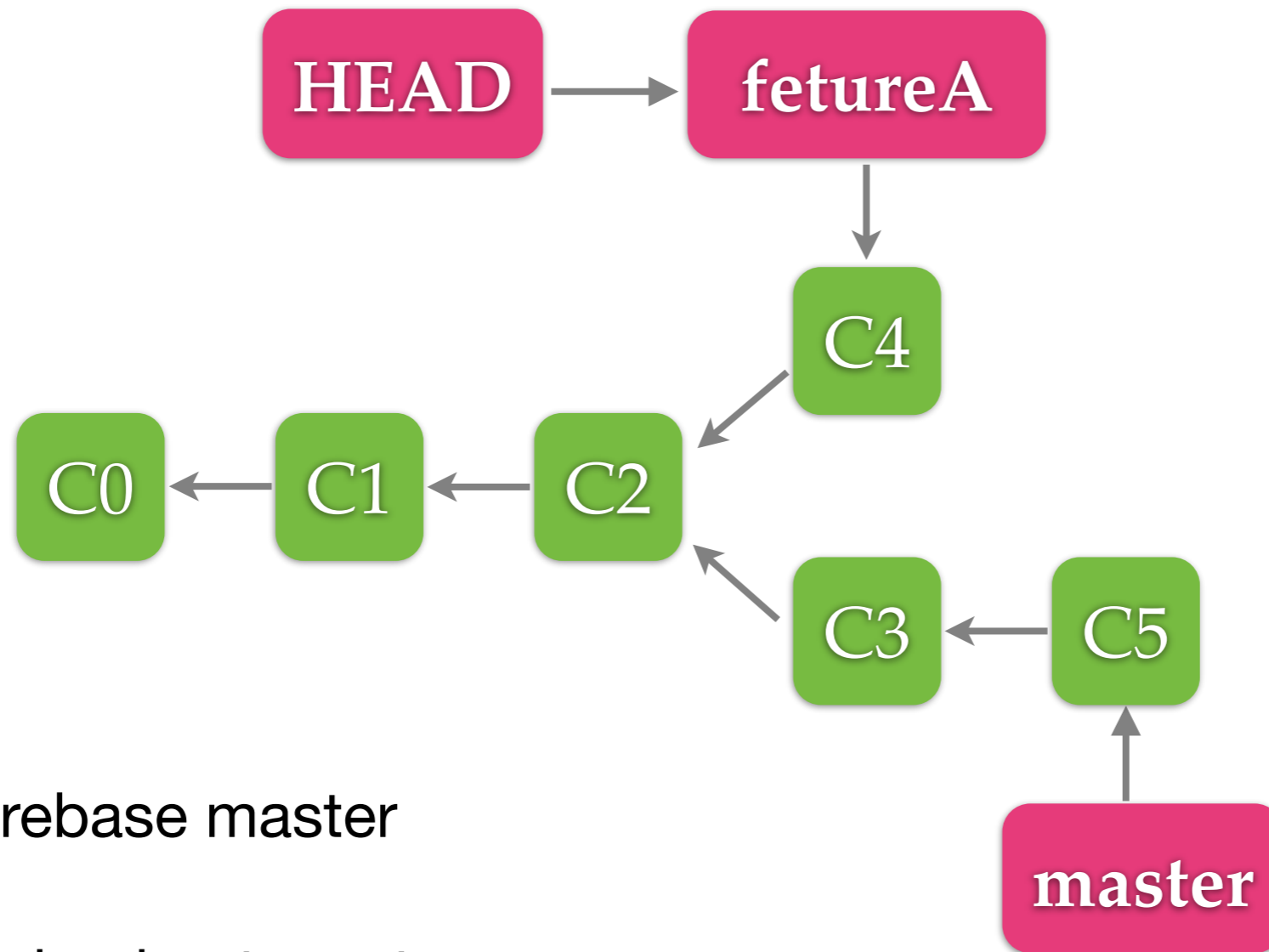
```
$ git merge featureA
```

Merge



➔ `$ git merge featureA`

Rebase

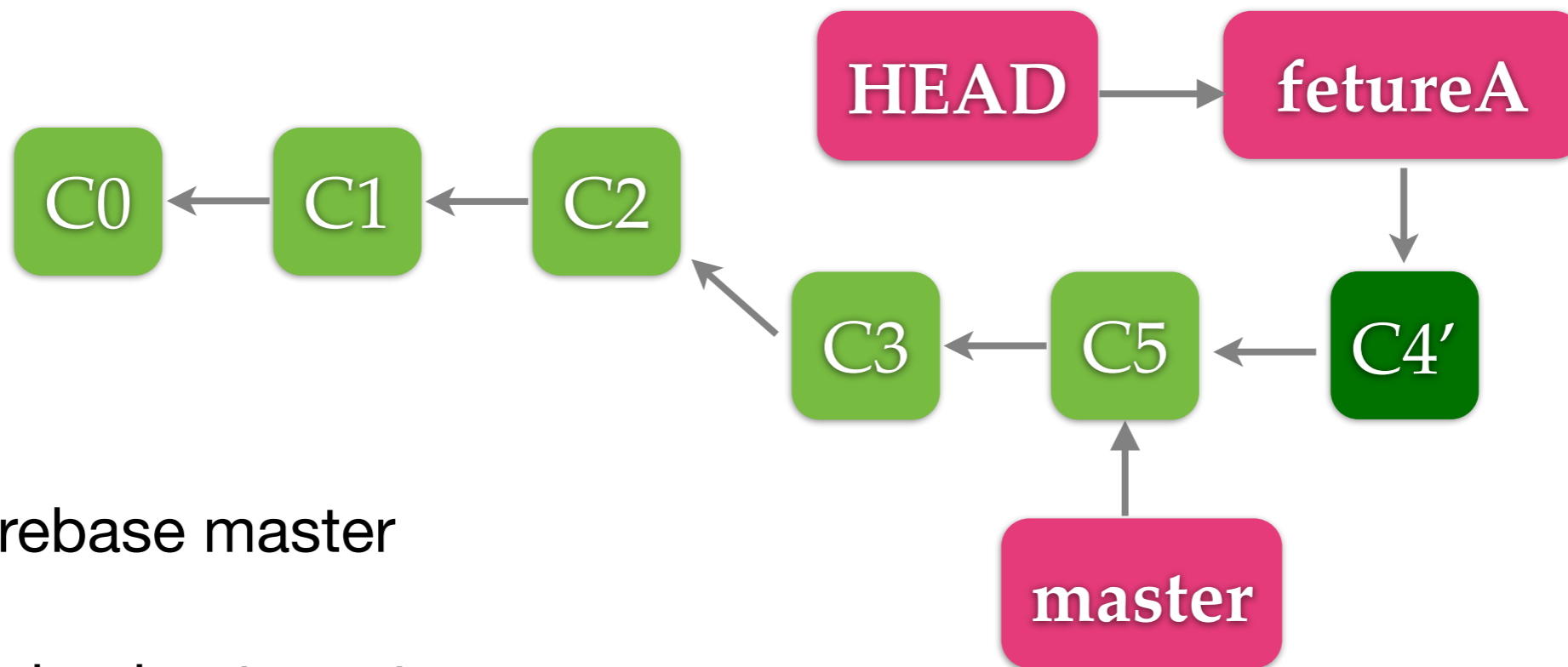


\$ git rebase master

\$ git checkout master

\$ git rebase featureA

Rebase

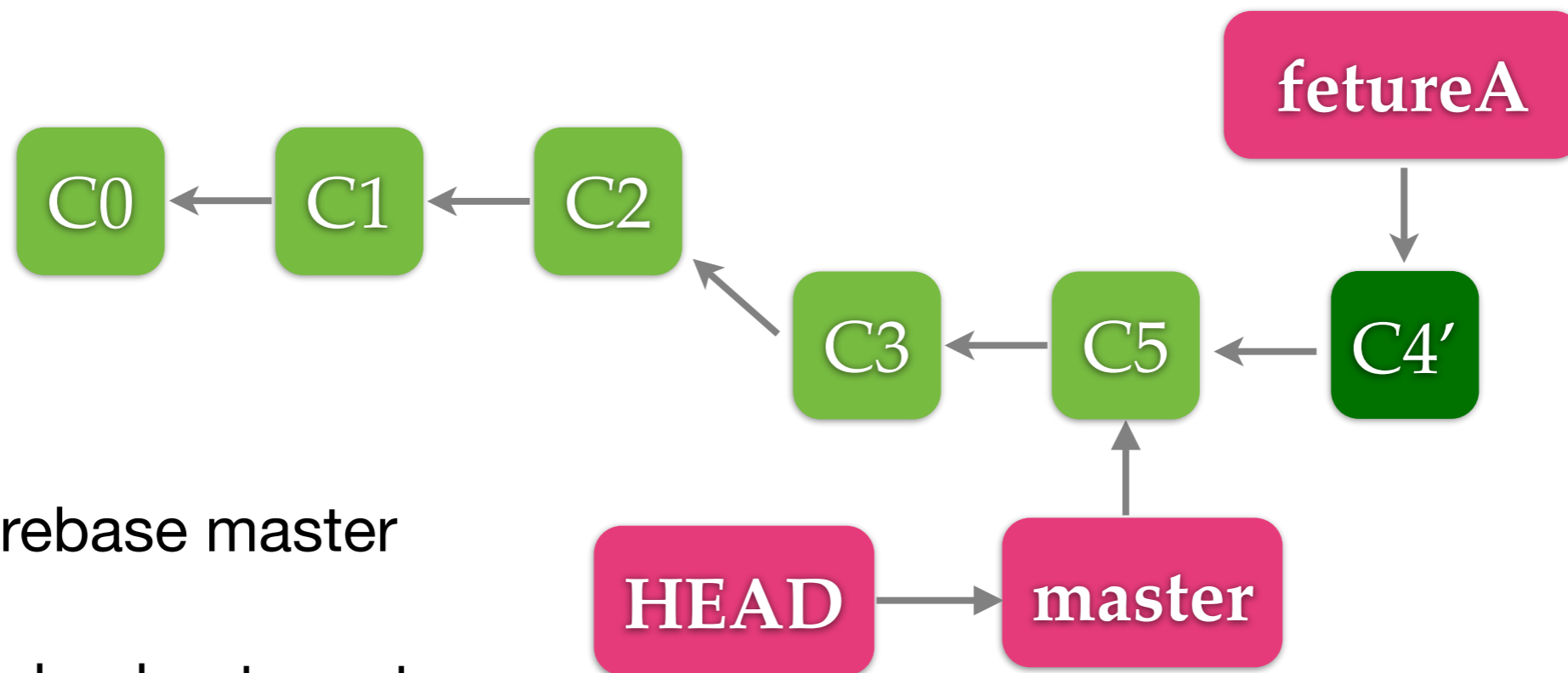


➔ \$ git rebase master

\$ git checkout master

\$ git rebase featureA

Rebase

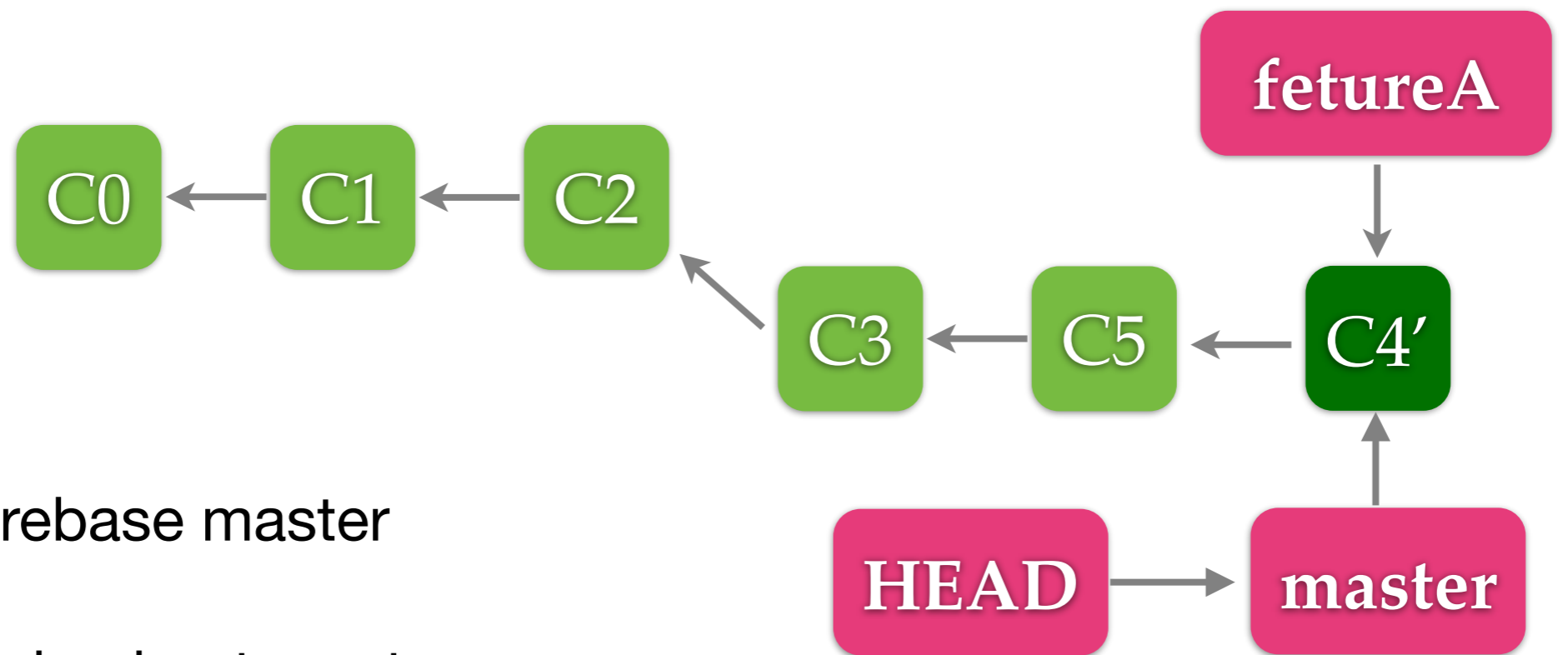


\$ git rebase master

➔ \$ git checkout master

\$ git rebase featureA

Rebase



\$ git rebase master

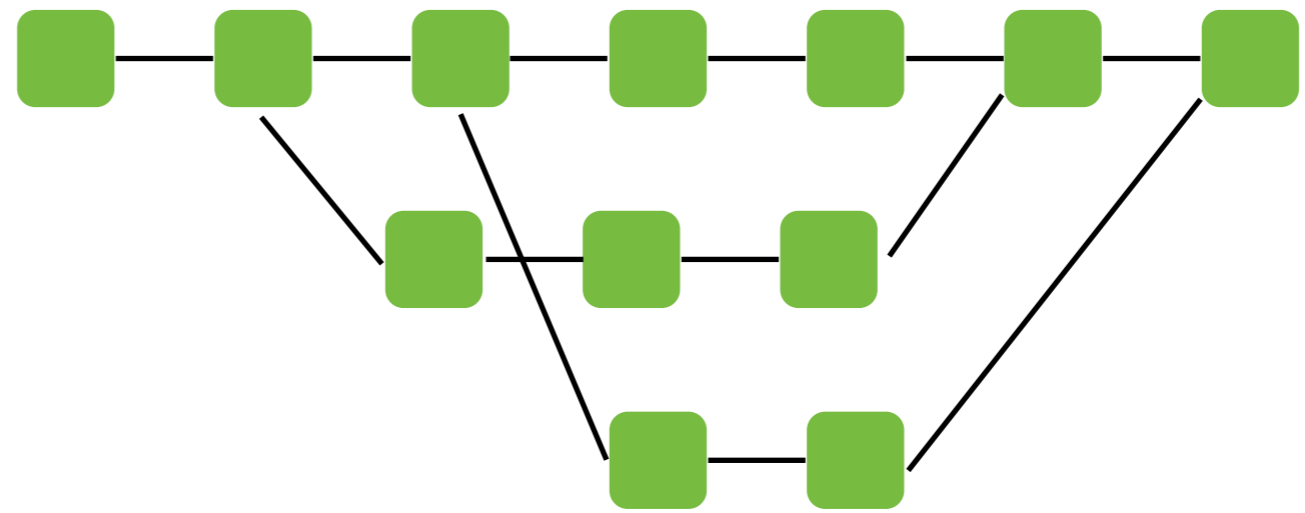
\$ git checkout master

➔ \$ git rebase featureA

Fast Forward

Merge VS Rebase

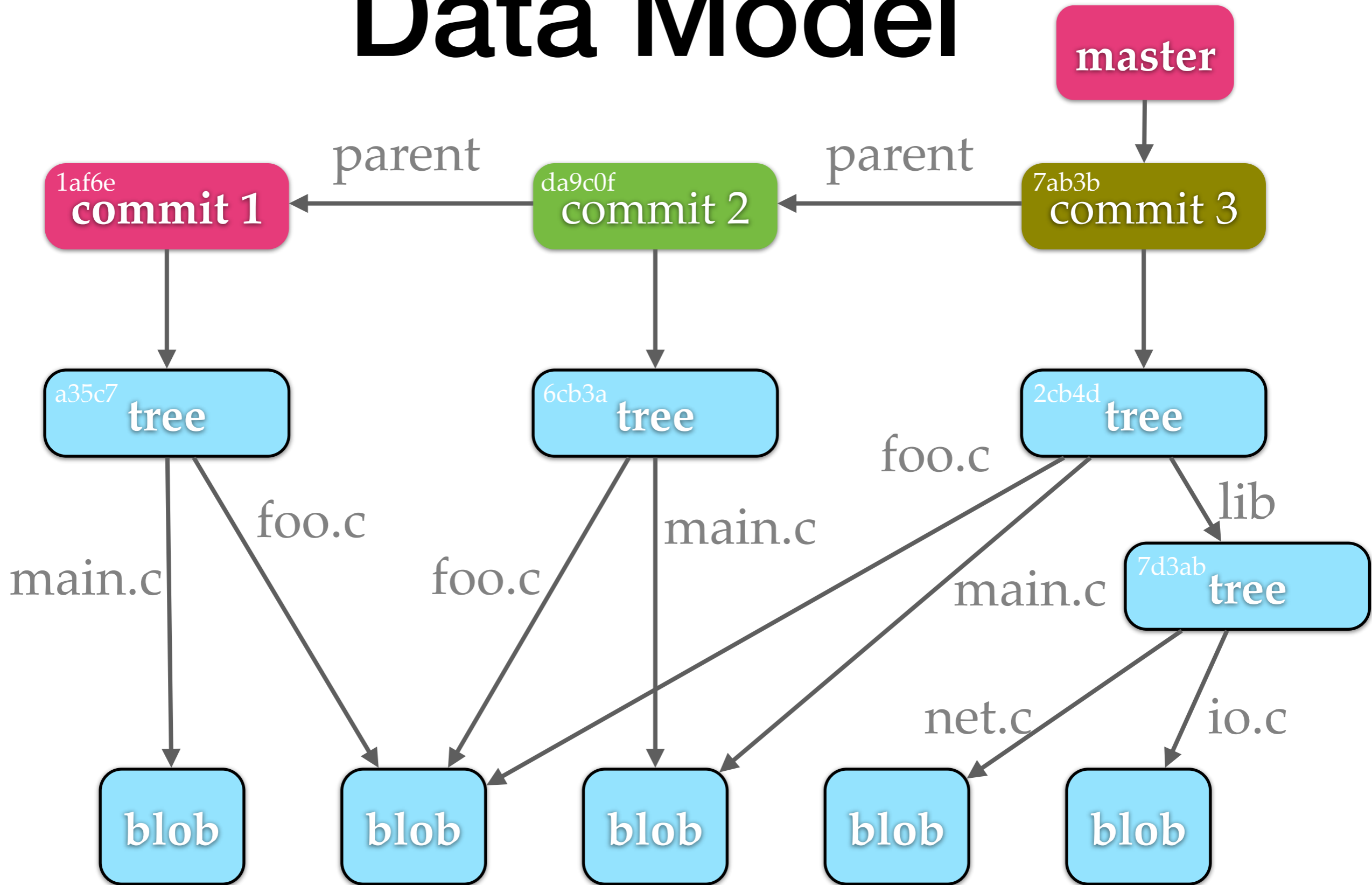
Merge



Rebase



Data Model



Data compressed by ZLIB

SHA1 hash of data and header as file name

Git Objects

- Git objects:
 - Commit
 - Tree
 - Blob

```
mht208@pssslvpn159 ~/.git/test (main)
$ ls
README

mht208@pssslvpn159 ~/.git/test (main)
$ git hash-object -w README
557db03de997c86a4a028e1ebd3a1ceb225be238

mht208@pssslvpn159 ~/.git/test (main)
$ ls .git/objects/55
7db03de997c86a4a028e1ebd3a1ceb225be238

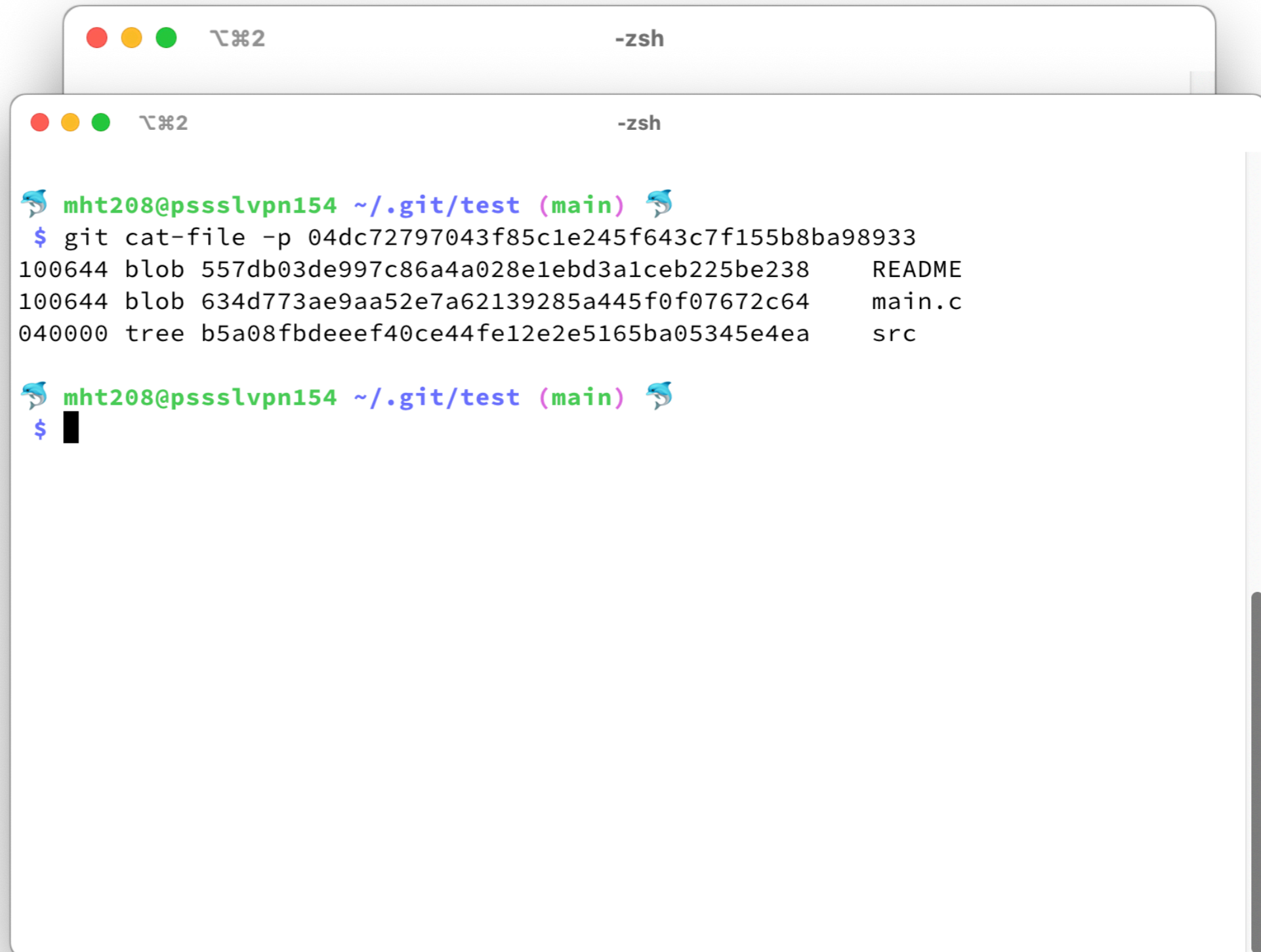
mht208@pssslvpn159 ~/.git/test (main)
$ git cat-file -t 557db
blob

mht208@pssslvpn159 ~/.git/test (main)
$ git cat-file -p 557db
Hello World

mht208@pssslvpn159 ~/.git/test (main)
$
```

Git Objects

- Git objects:
 - Commit
 - Tree
 - Blob



```
mht208@pssslvpn154 ~/.git/test (main)
$ git cat-file -p 04dc72797043f85c1e245f643c7f155b8ba98933
100644 blob 557db03de997c86a4a028e1ebd3a1ceb225be238    README
100644 blob 634d773ae9aa52e7a62139285a445f0f07672c64    main.c
040000 tree b5a08fbdeef40ce44fe12e2e5165ba05345e4ea    src
mht208@pssslvpn154 ~/.git/test (main)
$
```

History

this is the ID of the version

```
DHCP-22126 project $ git log
commit d05fe5d5bb03bd42abb662e45284020b1c6d85f9
Author: Ming-Hsien Tsai <mhtsai208@gmail.com>
Date: Tue Sep 12 14:07:39 2017 +0800
```

Add README.md.

```
DHCP-22126 project $ █
```


History

```
4. git

commit 0119516da8f2d96da0695ed9c1a0045616fca9ca
Author: Ming-Hsien Tsai <mhtsai208@gmail.com>
Date: Tue Jun 27 16:58:41 2017 +0800

    Add bvSubC, bvSbb, and bvSbbC.

commit f5313f6f9174e781e544198fd7ccbb0be4e502f9
Author: Bow-Yaw <bowyaw@bitbucket.org>
Date: Thu Jun 22 13:31:10 2017 +0800

    add succ_instr and succ_program

commit 886c57392c65723e2741dbe61f79b25e68aa04a1
Author: Bow-Yaw <bowyaw@bitbucket.org>
Date: Fri Jun 16 08:57:55 2017 +0800

    a script to identify addition chains

commit 0e25d1e7d7ca1c45ce9270c8681056f649865705
Author: Ming-Hsien Tsai <mhtsai208@gmail.com>
Date: Tue Jun 13 11:54:35 2017 +0800

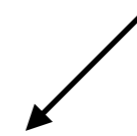
    Fix a bug where ule in Coq becomes ult in OCaml.

commit c9d360dc5609bb9bca06c3dfd0018a59ef59ca08
Author: Bow-Yaw <bowyaw@bitbucket.org>
Date: Tue Jun 13 11:10:07 2017 +0800

    return exit code from verification

commit 7228396b0dc5e7c71722716cbf52a2b28988bf9e
Merge: c1009c5 372c29a
Author: Bow-Yaw Wang <bywang@iis.sinica.edu.tw>
:
```

is is the ID of the version



```
5284020b1c6d85f9
208@gmail.com>
7 +0800
```

History

```
4. git
commit 0119516da8f2d96da0695ed9c1a0045616fca9ca
Author: Ming-Hsien Tsai <mhtsai208@gmail.com>
Date: Tue Jun 27 16:58:41 2017 +0800

    Add bvSubC, bvSbb, and bvSbbC.

commit f5313f6f9174e781e544198fd7ccbb0be4e502f9
Author: Bow-Yaw <bowyaw@bitbucket.org>
Date: Thu Jun 22 13:31:10 2017 +0800

    add succ_instr and succ_program

commit 886c57392c65723e2741dbe61f79b25e68aa04a1
Author: Bow-Yaw <bowyaw@bitbucket.org>
Date: Fri Jun 16 08:57:55 2017 +0800

    a script to identify addition chains

commit 0e25d1e7d7ca1c45ce9270c8681056f649865705
Author: Ming-Hsien Tsai <mhtsai208@gmail.com>
Date: Tue Jun 13 11:54:35 2017 +0800

    Fix a bug where ule in Coq becomes ult in OCaml.

commit c9d360dc5609bb9bca06c3dfd0018a59ef59ca08
Author: Bow-Yaw <bowyaw@bitbucket.org>
Date: Tue Jun 13 11:10:07 2017 +0800

    return exit code from verification

commit 7228396b0dc5e7c71722716cbf52a2b28988bf9e
Merge: c1009c5 372c29a
Author: Bow-Yaw Wang <bywang@iis.sinica.edu.tw>
:
```

```
4. less
* f6fbbf4 (HEAD -> nondet) Add value type as an argument of store functors.
* 3cdda6c Move conversion between VS and SSAVS to DSL2SSACommons.
* 5be9e28 Add bvDSLRel.
* a479a2a (origin/nondet) Simplify some proofs.
* 4ea01b5 Improve dp_mem and dp_subset.
* a6b4a99 Add SsrFSet to avoid redundant declarations of variable sets.
* 4136533 Remove useless code in bvDSL. Simplify some proofs in bvDSL.
* 73cbe0b Add Equal to Store.
* a73c784 Merge branch 'master' into nondet
|\
| * 1b7952e Add a buggy implementation of fe25519_mul with radix 2^64 representa
tion.
| * 34b2c45 Add simple decision procedures for mem and subset.
| * 0119516 (subtraction) Add bvSubC, bvSbb, and bvSbbC.
* | f5313f6 add succ_instr and succ_program
|/
* 886c573 (origin/ccs17) a script to identify addition chains
* 0e25d1e Fix a bug where ule in Coq becomes ult in OCaml.
* c9d360d return exit code from verification
* 7228396 Merge branch 'master' of fm1.iis.sinica.edu.tw:/home/git/certified_q
hasm_vcg
|\
| * 372c29a Remove unused admitted lemmas in bvSSA2QFBV.
| * 4de5f10 Proved admitted lemmas in bvSSA2zSSA.
| * ca71d71 Add more lemmas.
* | c1009c5 refer to modified gbarith
|/
* c4e55f6 add to_bvdsl.py and rename to_dsl.py
* a5e23c4 allow configurations in comments
* b78c849 Fix a typo.
* 5fd989f Fix a problem in rewrite_zpower_nat.
* 5092d34 Update specifications.
* e7159cd Use Z for equality assertions.
* af5e257 Fix a bug in rewrite_assign2_with.
: 27
```

File Difference

```
DHCP-22126 project $ git diff
diff --git a/README.md b/README.md
index 52cb6cd..eb99592 100644
--- a/README.md
+++ b/README.md
@@ -1,3 @@
+==== About ====
+
This is readme.
DHCP-22126 project $ █
```

File Restore (Checkout)

```
DHCP-22126 project $ cat README.md  
===== About =====
```

This is readme.

```
DHCP-22126 project $ git checkout README.md
```

```
DHCP-22126 project $ cat README.md
```

This is readme.

```
DHCP-22126 project $ █
```

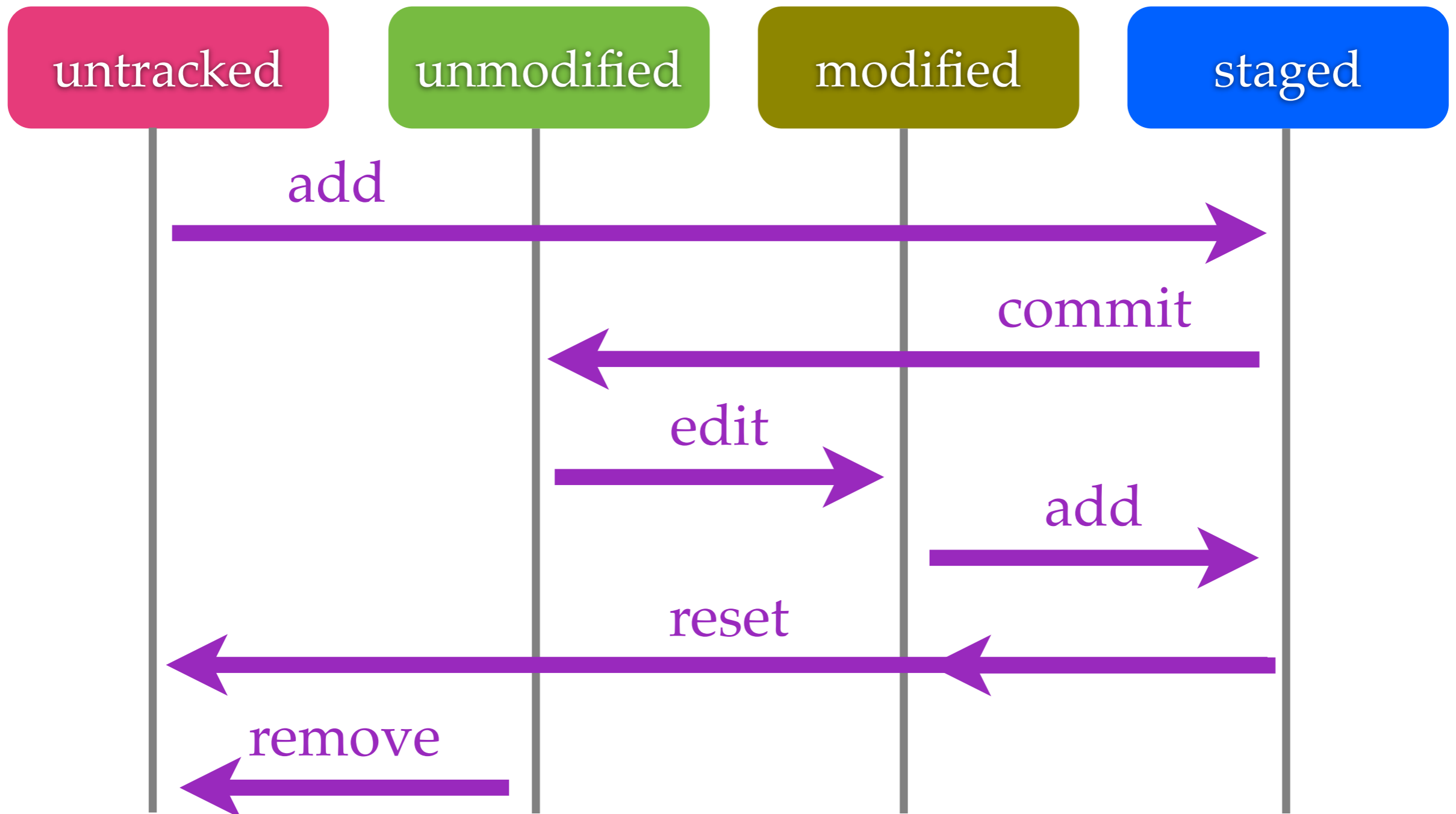


README.md

```
** Convert a DSL state to an SSA state. **  
definition ssa_state (m : vmmap) (s : State.t) : bv64SSA.State.t :=  
  fun v =>  
    if (sidx v) == get_index (svor v) m  
    then State.occ (svor v) s  
    else State.occ (svor v) State.empty.  
  
lemma ssa_state_equiv :  
  forall! m s, state_equiv m s (ssa_state m s).  
  proof.  
    intros m s s'.  
    rewrite (occ_ssa_state_eq _ (exact (get_index x m))).  
    reflexivity.  
  qed.
```



File Status



File Status

```
DHCP-22126 project1 $ git status
```

```
On branch master
```

```
Your branch is ahead of 'origin/master' by 3 commits.
```

```
(use "git push" to publish your local commits)
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
    modified:   B.c
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
    modified:   A.c
```

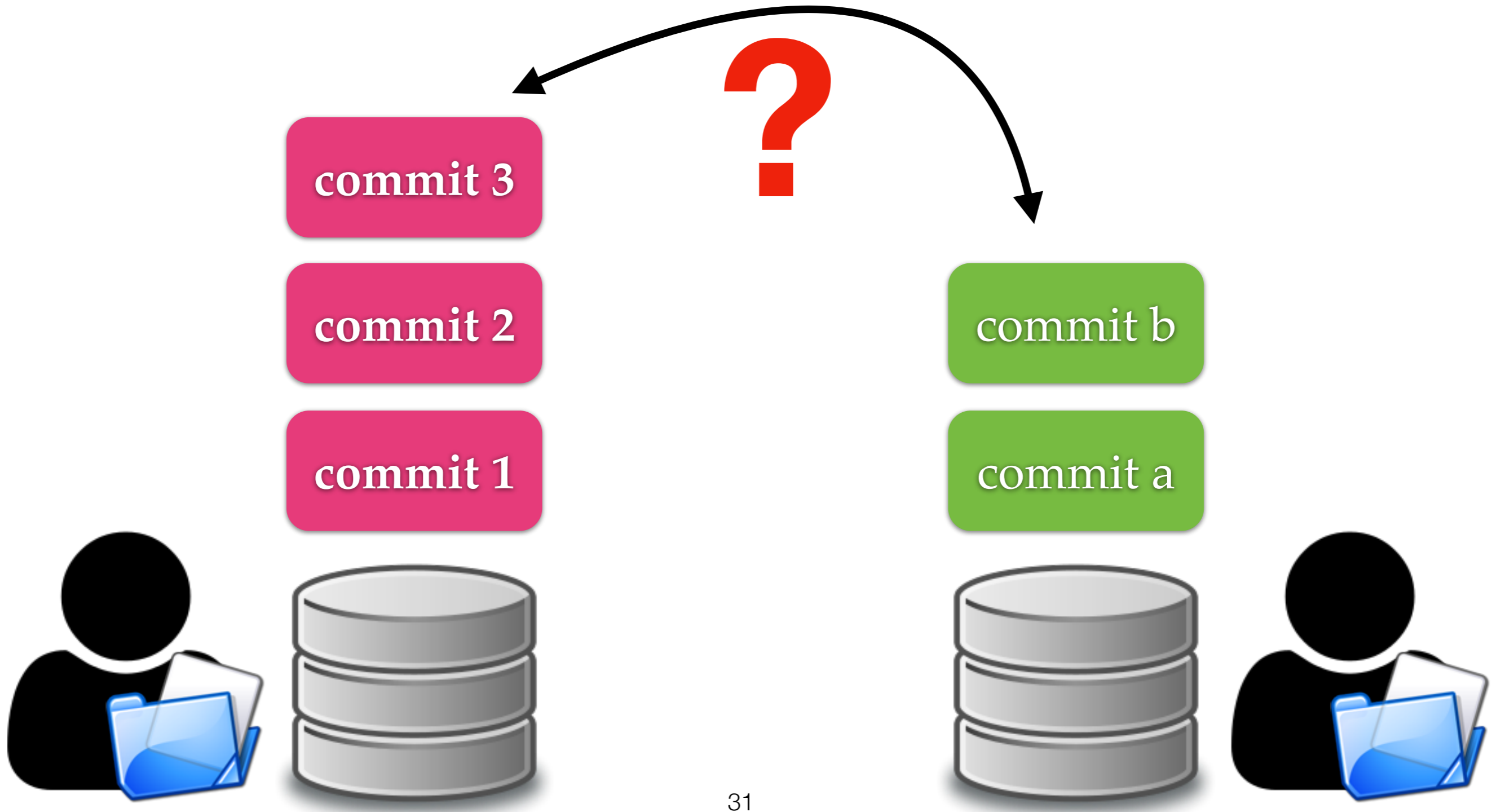
```
    modified:   README.md
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

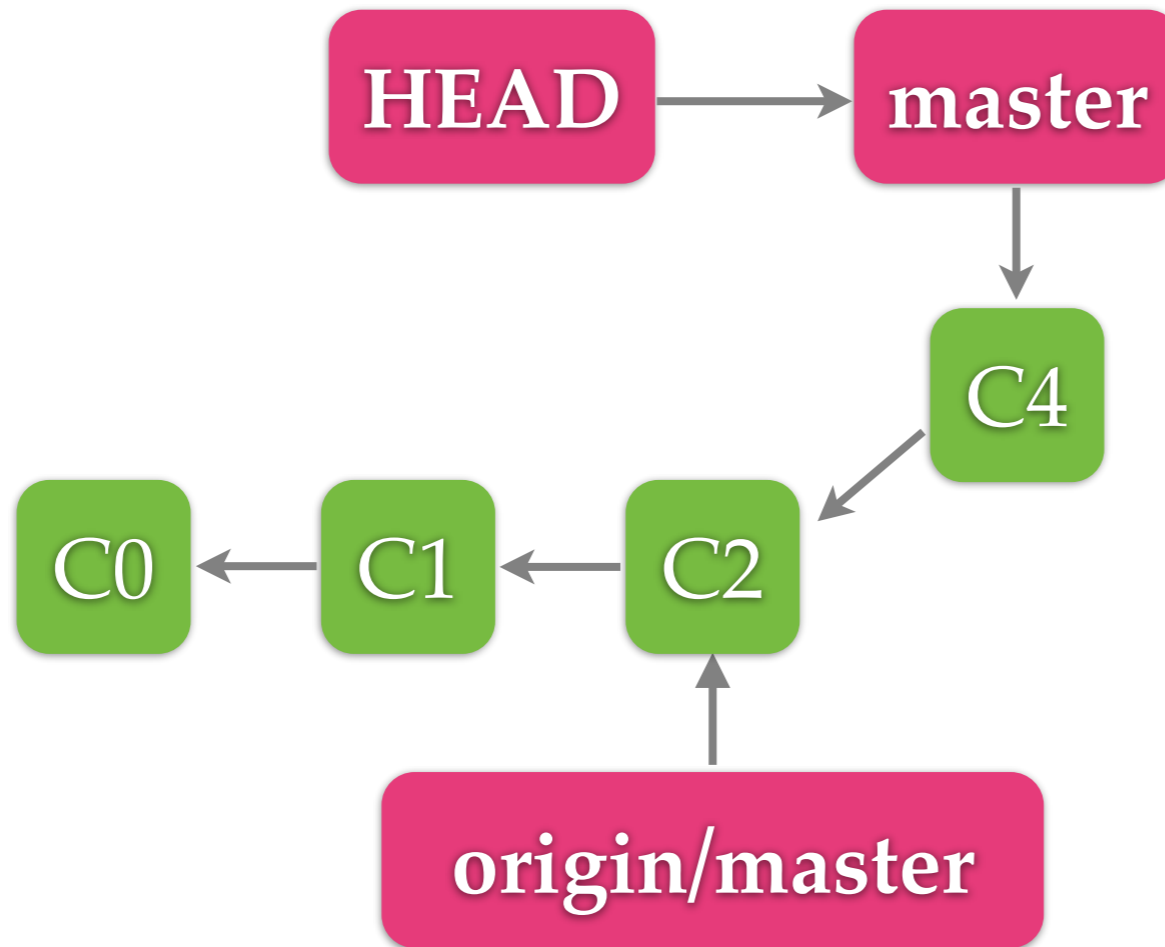
```
    D.java
```

Synchronization



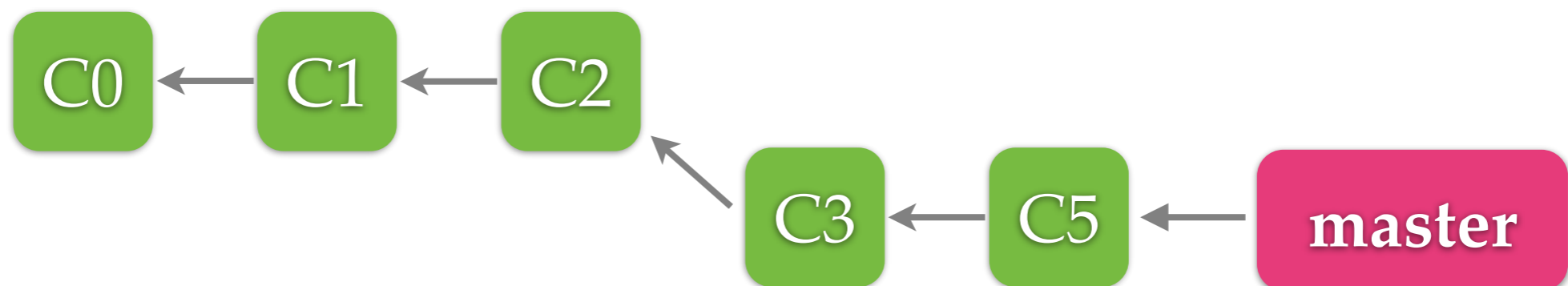
Fetch

Local



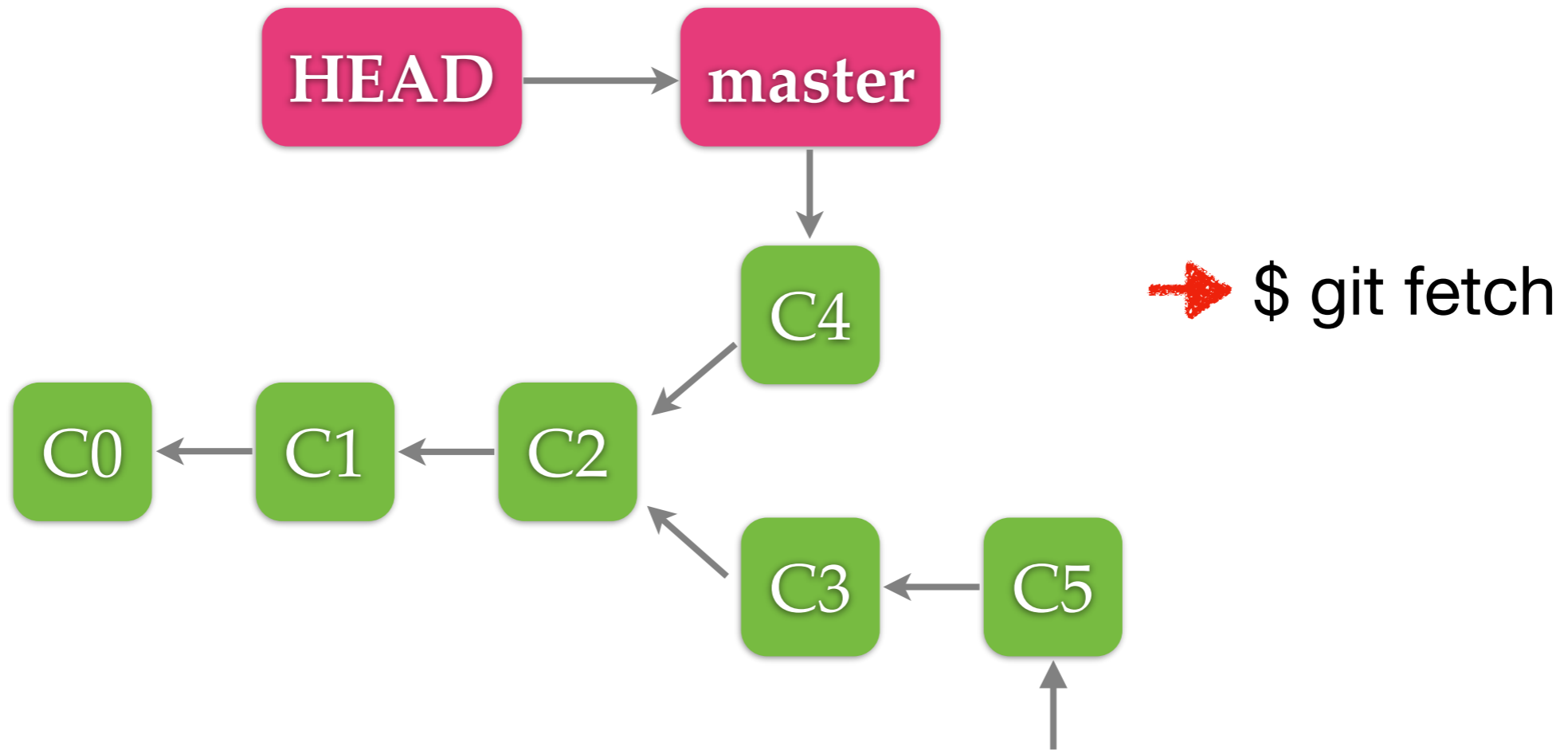
\$ git fetch

Remote (origin)

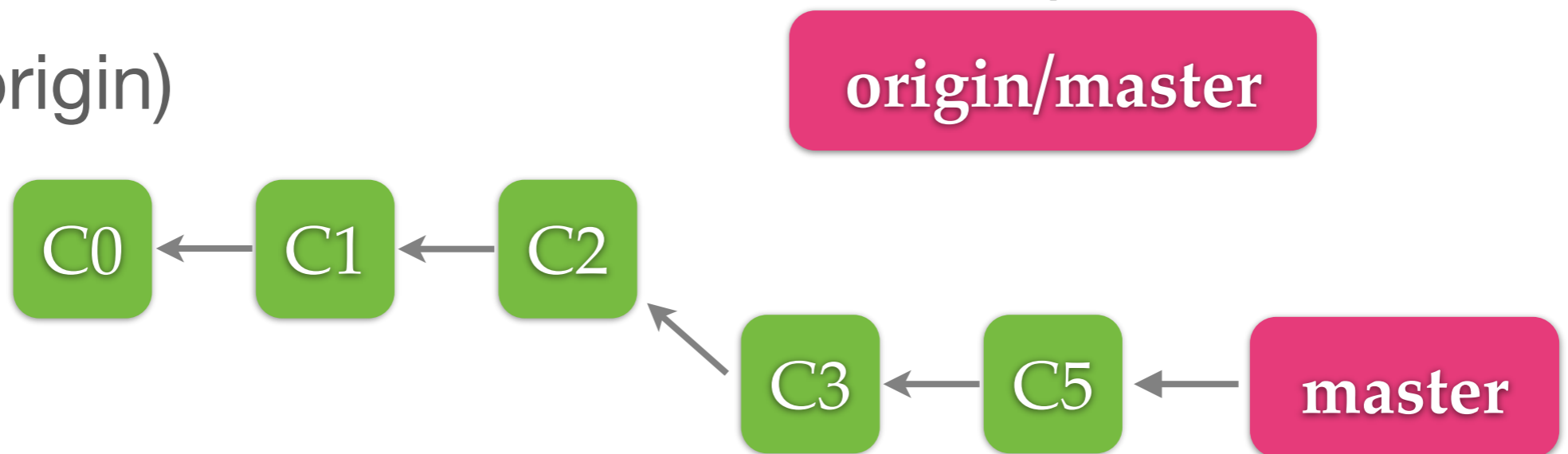


Fetch

Local

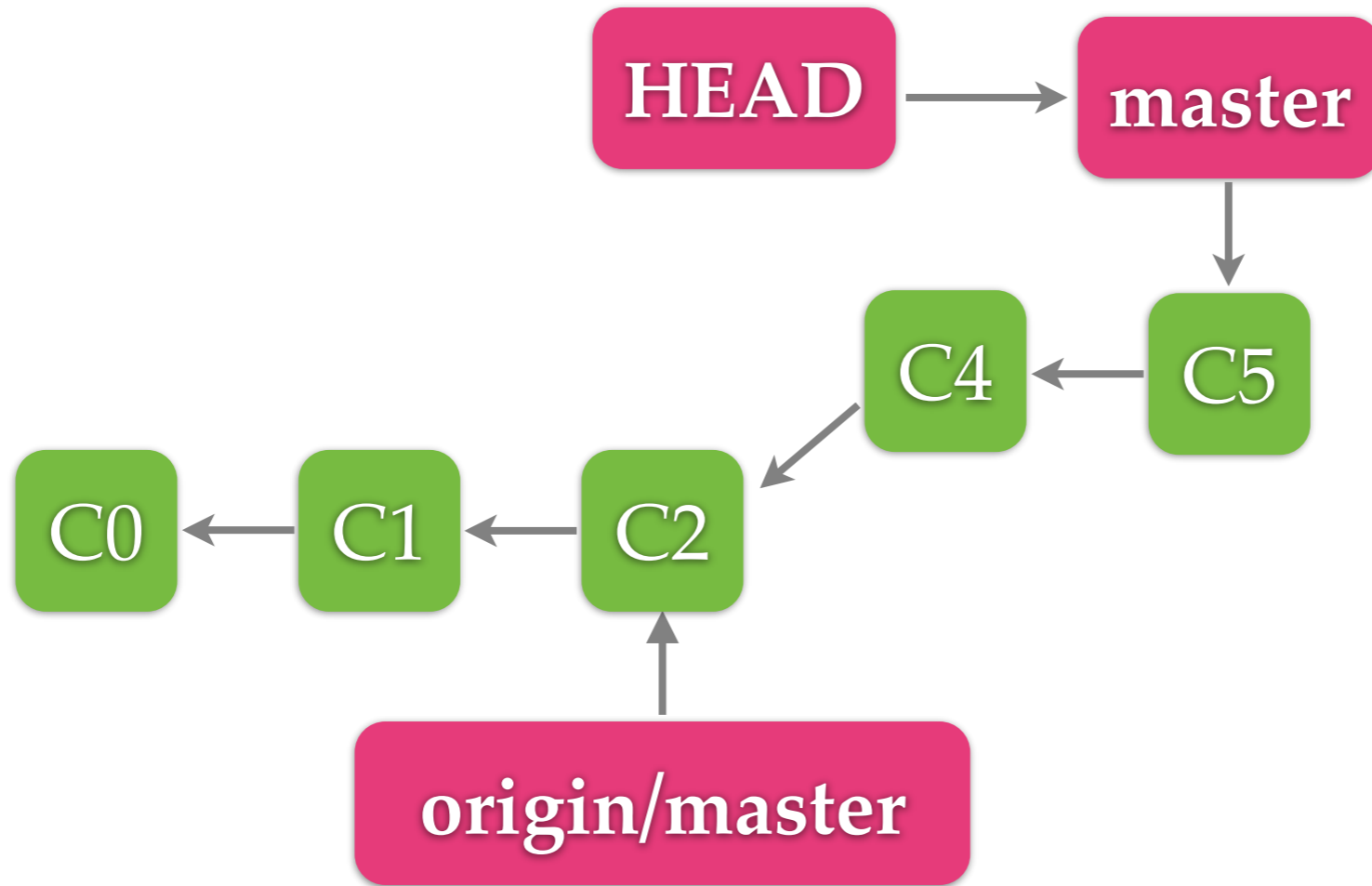


Remote (origin)



Push

Local



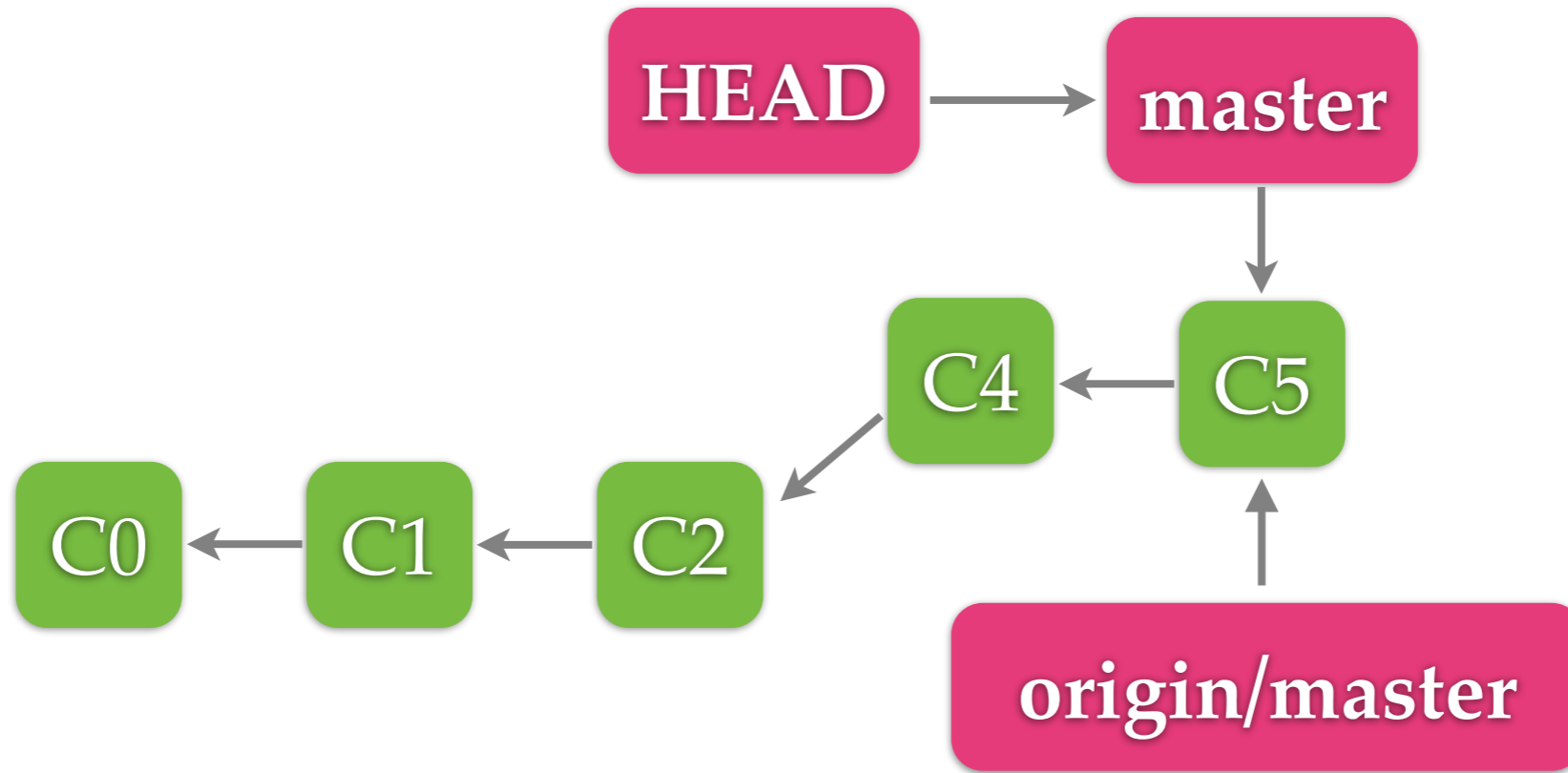
Remote (origin)



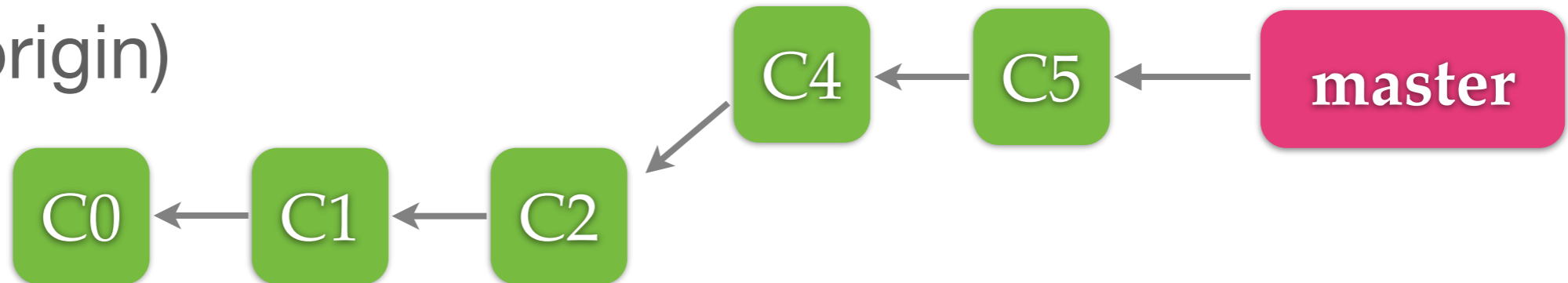
\$ git push origin master

Push

Local



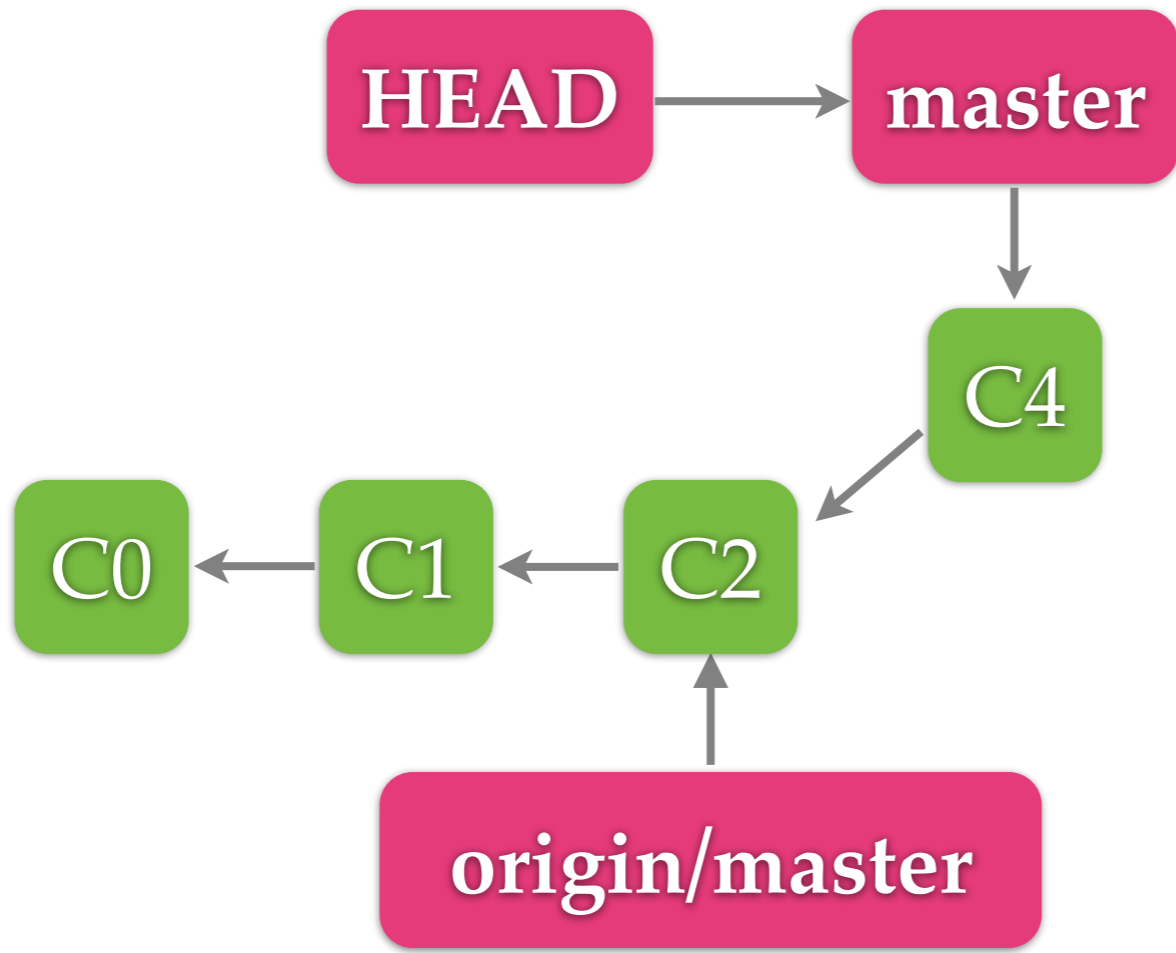
Remote (origin)



➔ \$ git push origin master

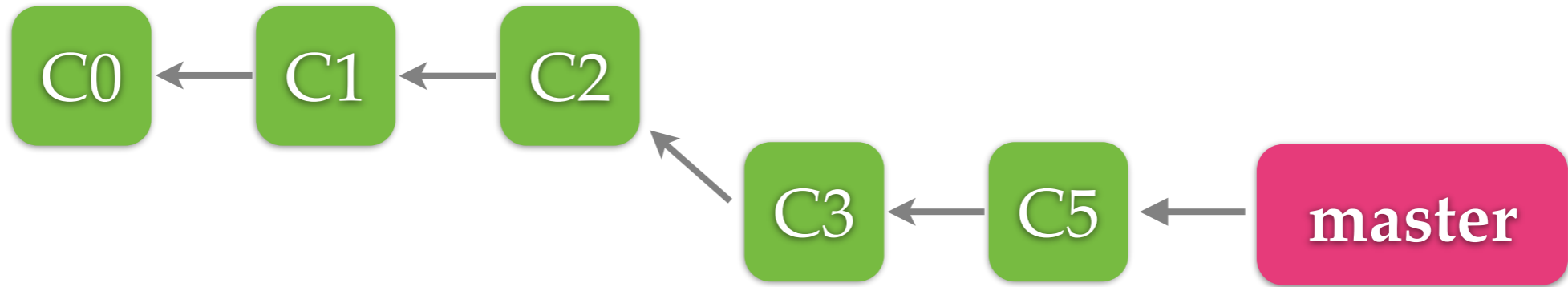
Pull

Local



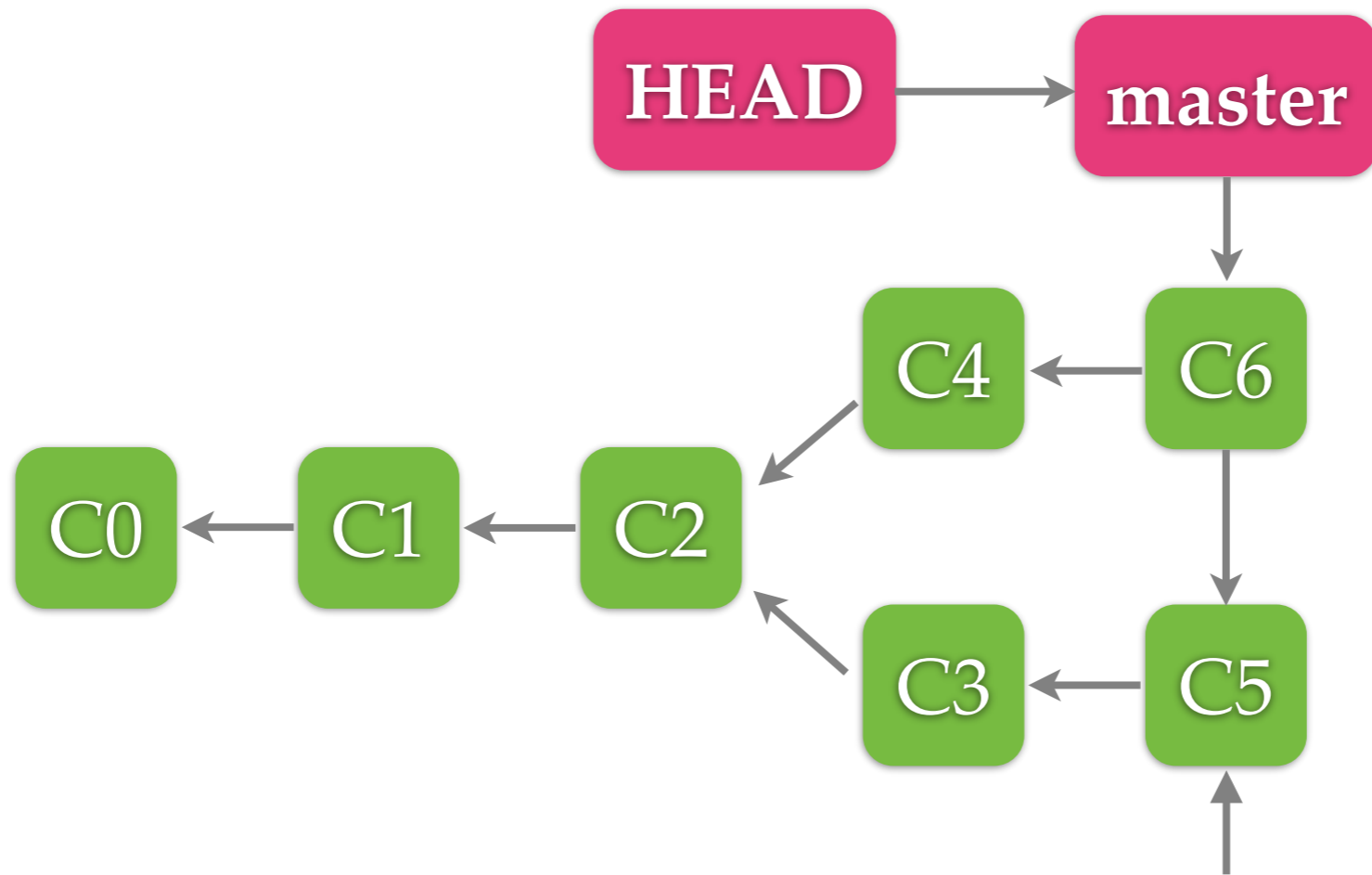
\$ git pull
= fetch + merge

Remote (origin)



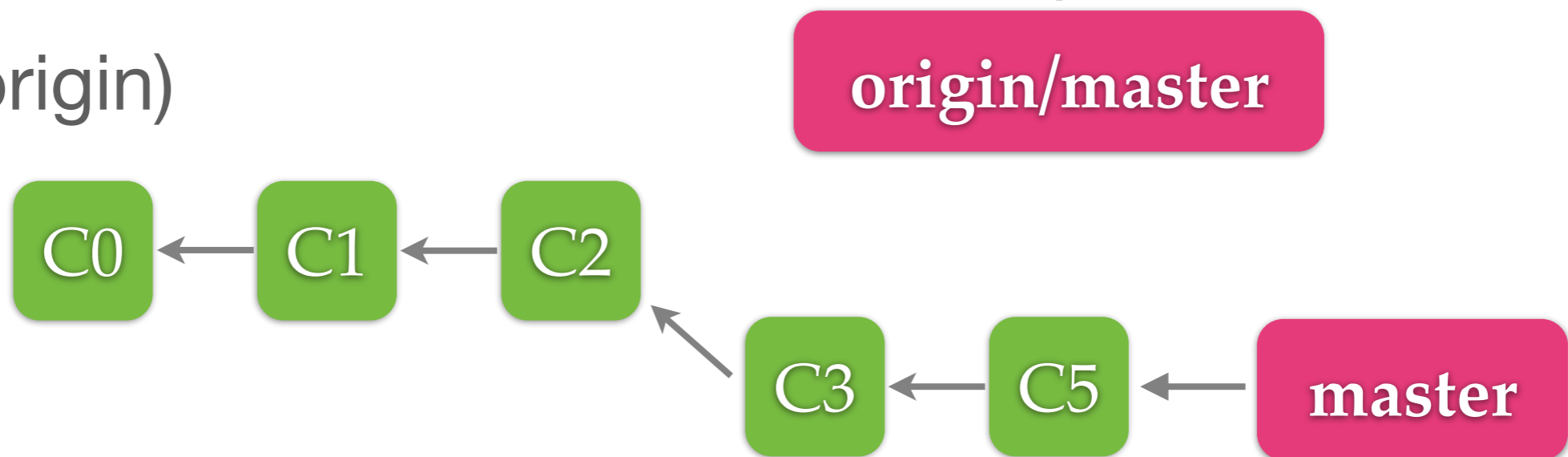
Pull

Local



➔ \$ git pull
= fetch + merge

Remote (origin)



Resolve Conflicts

```
DHCP-22126 project1 $ git fetch
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From /Users/mht208/demo/server
   3691b0d..57d4f6e  master    -> origin/master
DHCP-22126 project1 $ git merge
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
DHCP-22126 project1 $ █
```

Resolve Conflicts

1. Edit files with conflicts

```
█<<<<<<< HEAD
==== About Project A ====
=====
==== About Me ====
>>>>>>> refs/remotes/origin/master
```

This is readme.

Resolve Conflicts

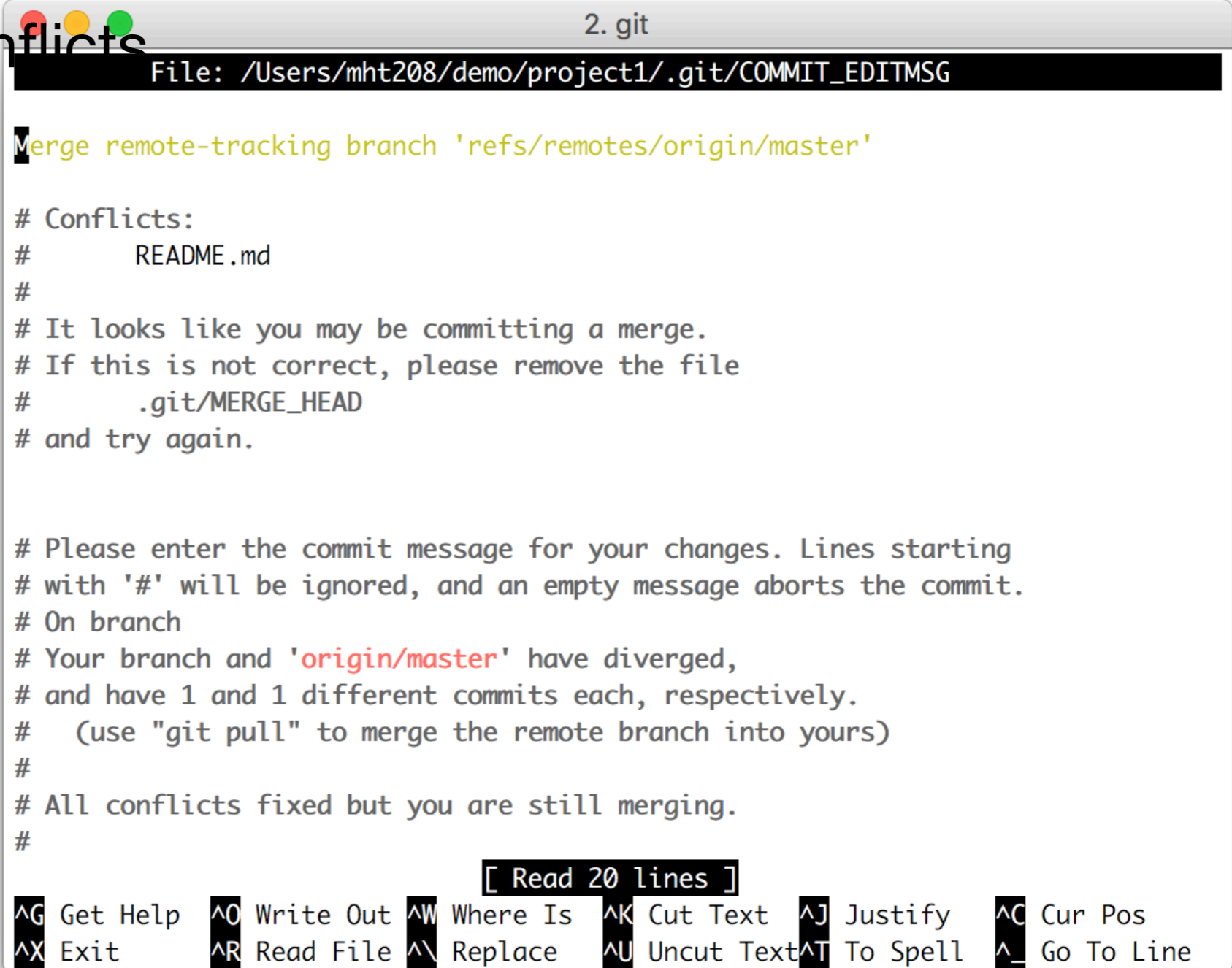
1. Edit files with conflicts
2. git add

```
█<<<<<<< HEAD
===== About Project A =====
=====
===== About Me =====
>>>>>>> refs/remotes/origin/master
```

This is readme.

Resolve Conflicts

1. Edit files with conflicts
2. git add
3. git commit



```
2. git
File: /Users/mht208/demo/project1/.git/COMMIT_EDITMSG

Merge remote-tracking branch 'refs/remotes/origin/master'

# Conflicts:
#   README.md
#
# It looks like you may be committing a merge.
# If this is not correct, please remove the file
#   .git/MERGE_HEAD
# and try again.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch
# Your branch and 'origin/master' have diverged,
# and have 1 and 1 different commits each, respectively.
# (use "git pull" to merge the remote branch into yours)
#
# All conflicts fixed but you are still merging.
#

[ Read 20 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify    ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

Resolve Conflicts

1. Edit files with conflicts
2. git add
3. git commit

```
2. git
File: /Users/mht208/demo/project1/.git/COMMIT_EDITMSG
Merge remote-tracking branch 'refs/remotes/origin/master'

# Conflicts:
#   README.md
#
```

```
DHCP-22126 project1 $ git add README.md
```

```
DHCP-22126 project1 $ git commit
```

```
[master 3e5947f] Merge remote-tracking branch 'refs/remotes/origin/master'
```

```
DHCP-22126 project1 $
```

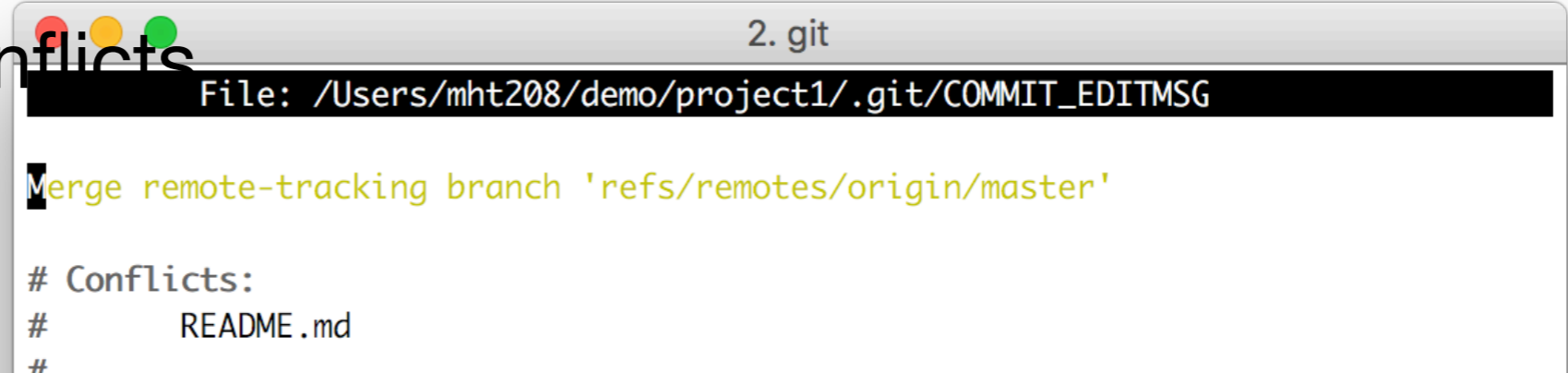
```
# with '#' will be ignored, and an empty message aborts the commit.
# On branch
# Your branch and 'origin/master' have diverged,
# and have 1 and 1 different commits each, respectively.
# (use "git pull" to merge the remote branch into yours)
#
# All conflicts fixed but you are still merging.
#
```

[Read 20 lines]

```
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

Resolve Conflicts

1. Edit files with conflicts
2. git add
3. git commit



```
2. git
File: /Users/mht208/demo/project1/.git/COMMIT_EDITMSG
Merge remote-tracking branch 'refs/remotes/origin/master'
# Conflicts:
# README.md
#
```

```
DHCP-22126 project1 $ git add README.md
```

```
DHCP-22126 project1 $ git commit
```

```
[master 3e5947f] Merge remote-tracking branch 'refs/remotes/origin/master'
```

```
DHCP-22126 project1 $
```

```
# with '#' will be ignored, and an empty message aborts the commit.
# On branch
```

```
* 3e5947f (HEAD -> master) Merge remote-tracking branch 'refs/remotes/origin/master'
```

```
| \
```

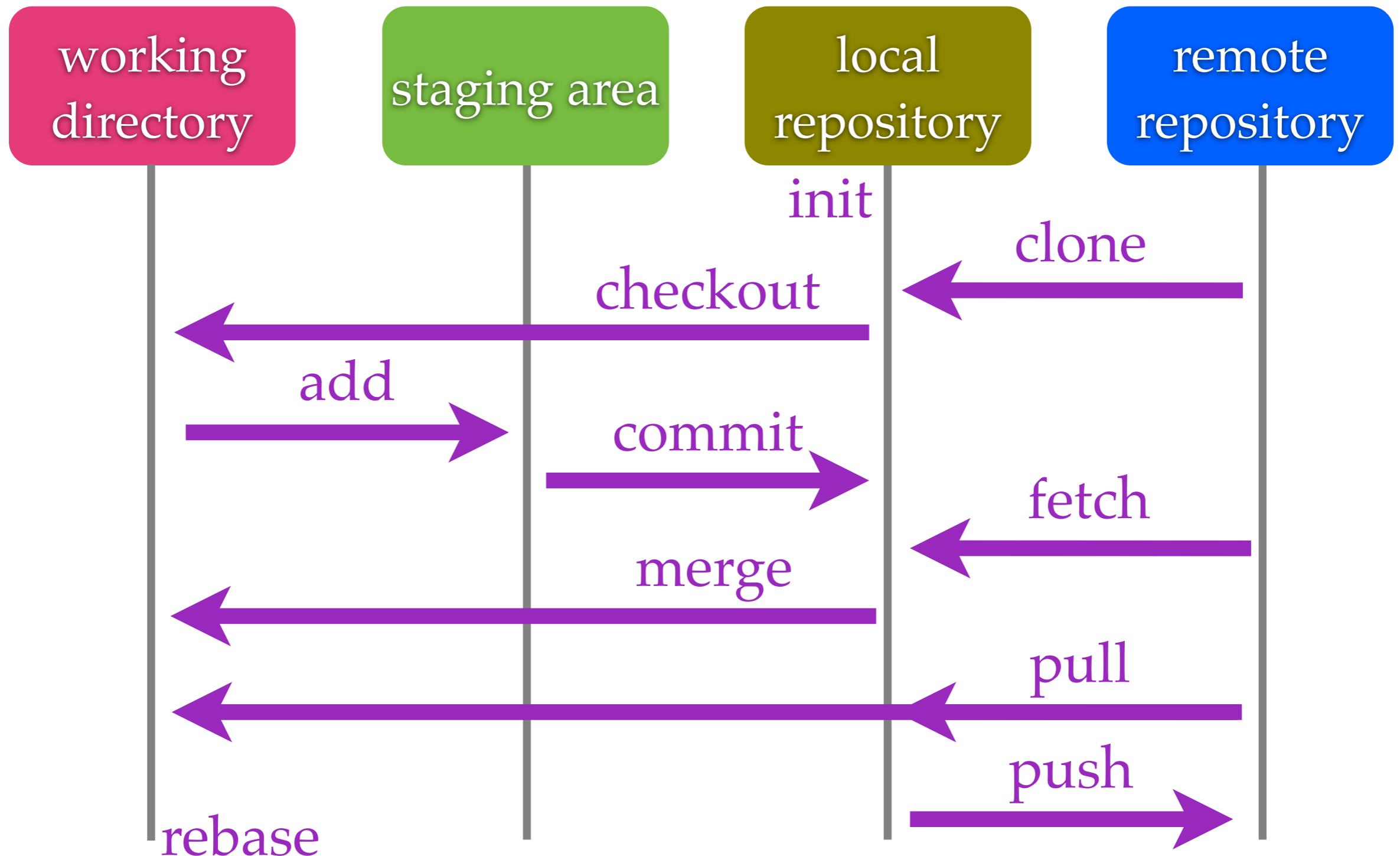
```
| * 57d4f6e (origin/master) Modify README.md.
```

```
* | 22804d8 Modify README.md.
```

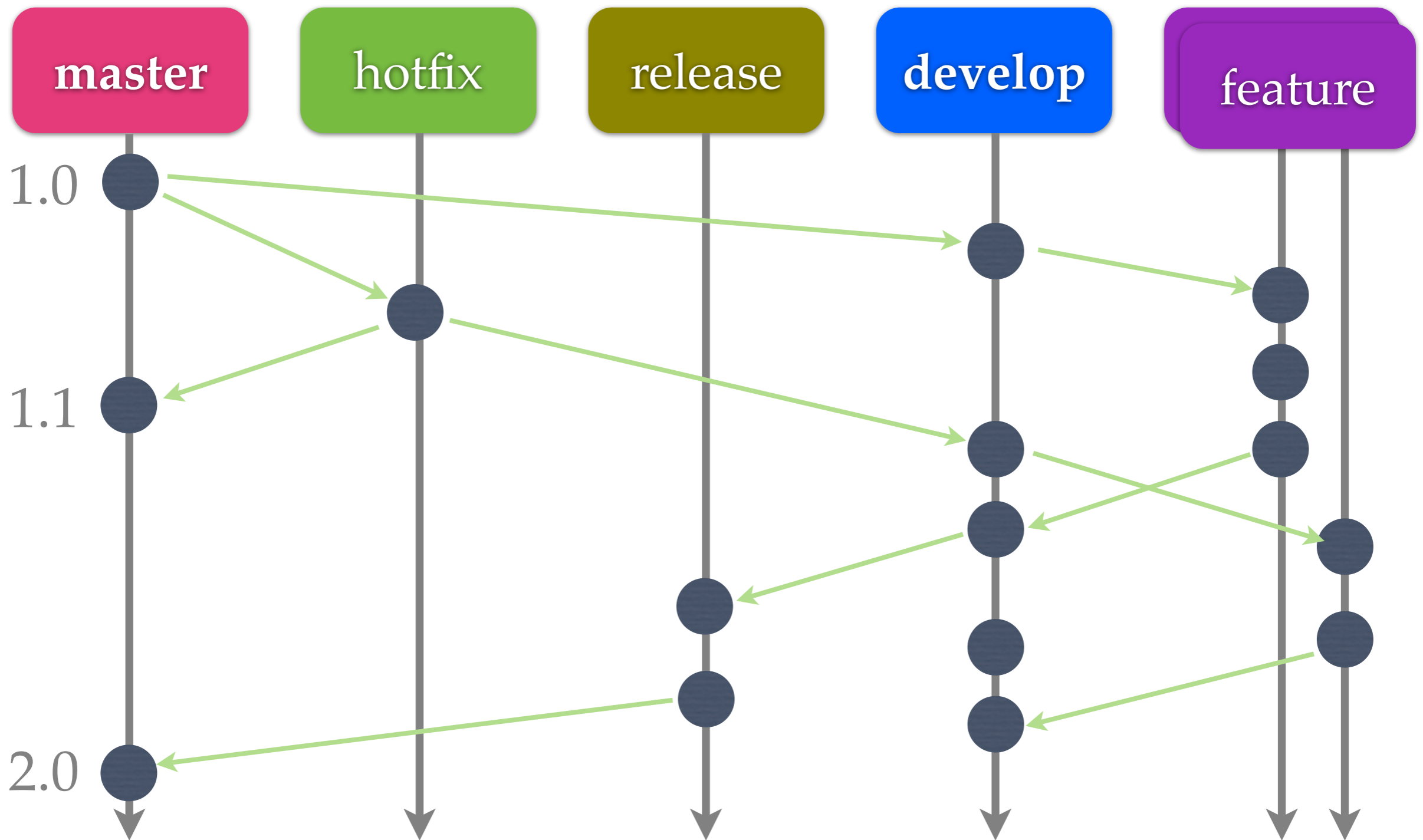
```
| /
```

```
* 3691b0d Add README.md.
```

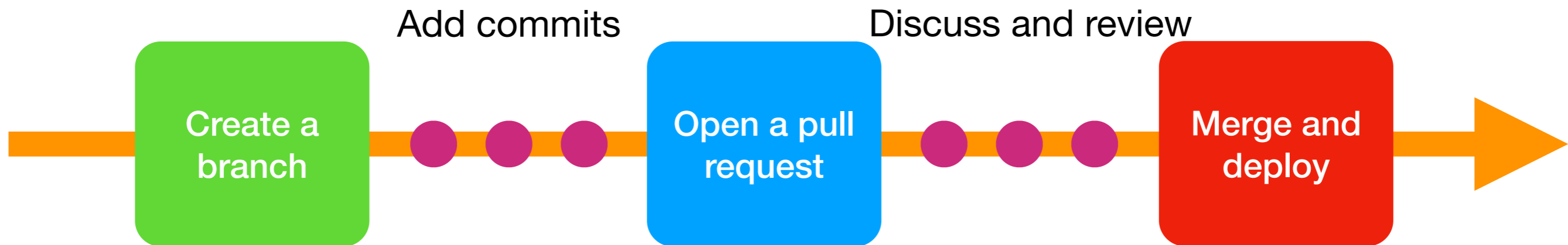
Summary



Git Flow



GitHub Flow



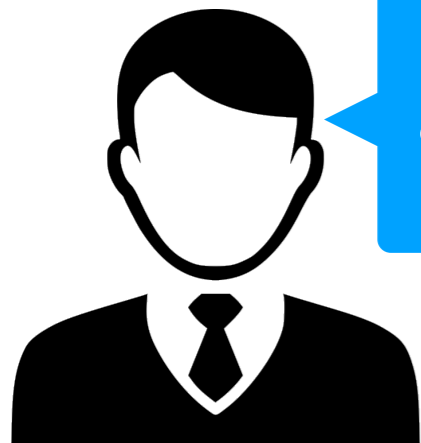
Stash



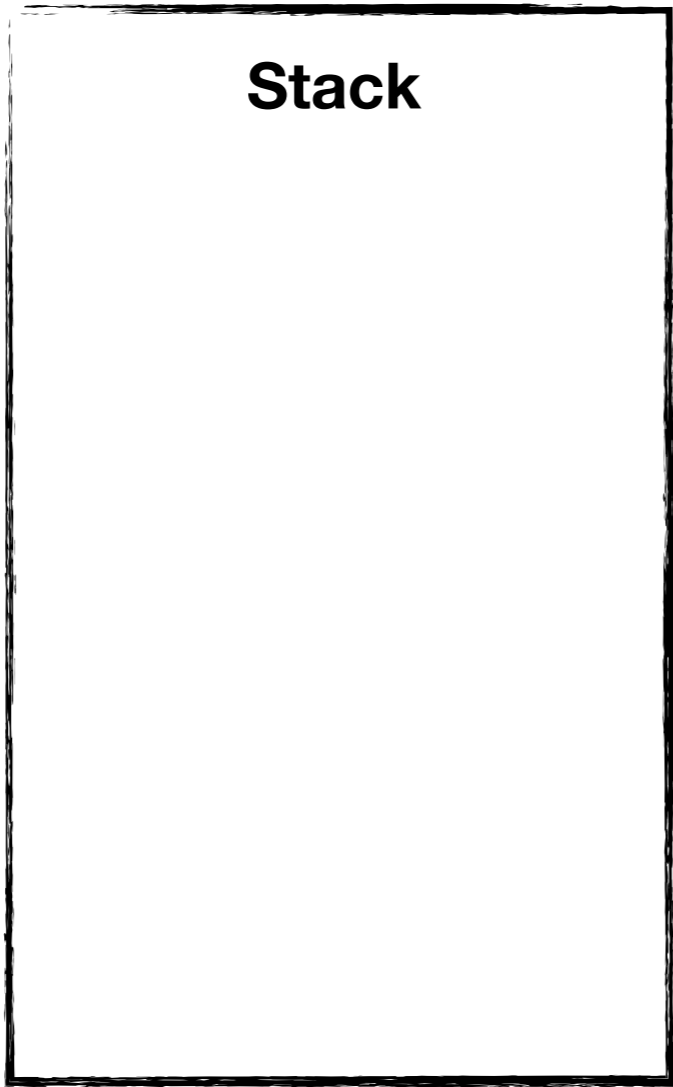
Stash



Stash



add feature A



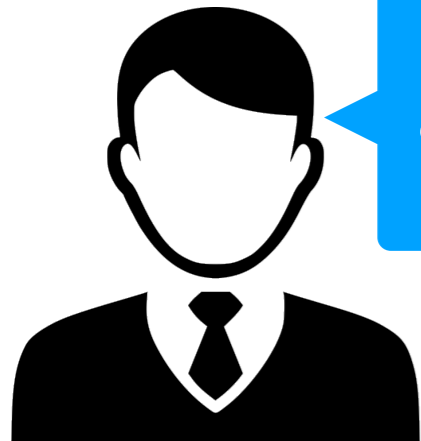
unfinished

commit 2

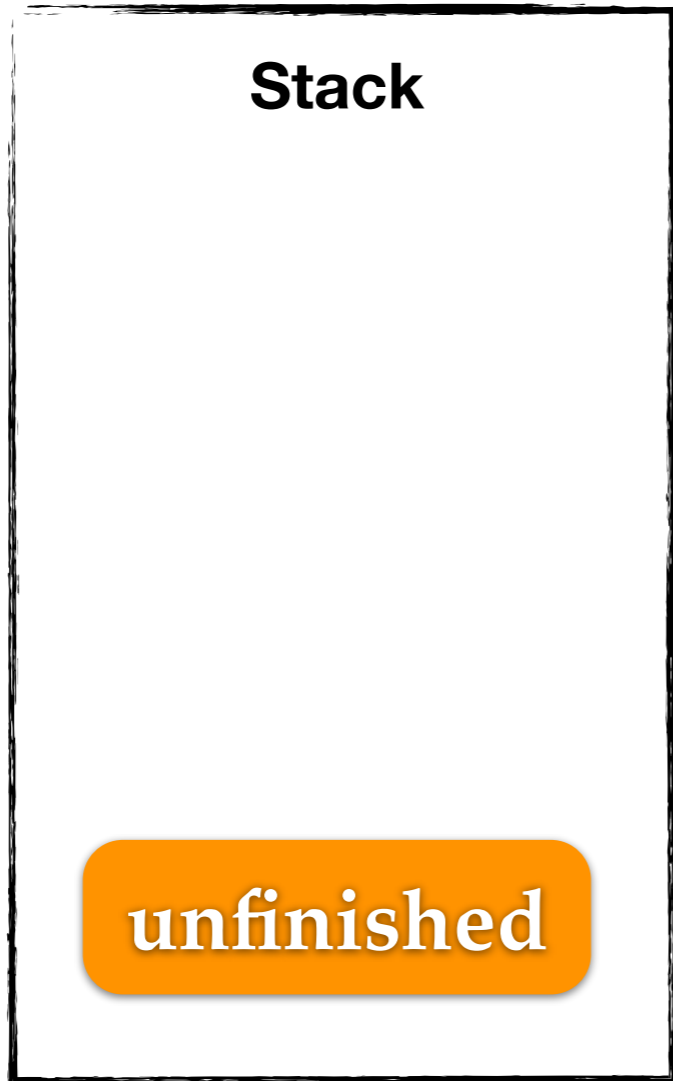
commit 1



Stash



add feature A



feature A

commit 2

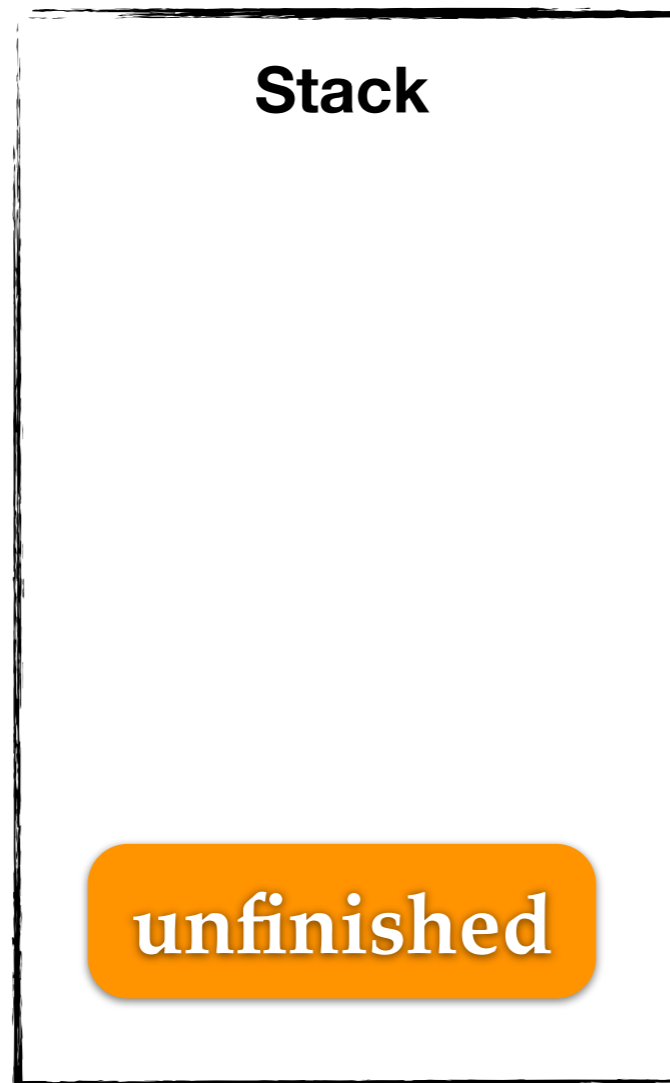
commit 1



Stash



fix a bug, right now!



feature A

An orange rounded rectangle containing the text "feature A".

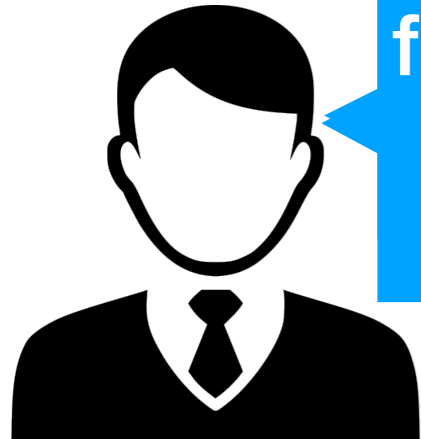
commit 2

A pink rounded rectangle containing the text "commit 2".

commit 1

A pink rounded rectangle containing the text "commit 1".

Stash



fix a bug, right now!



bug fix

commit 2

commit 1



Stash



fix a bug, right now!



bug fix

commit 2

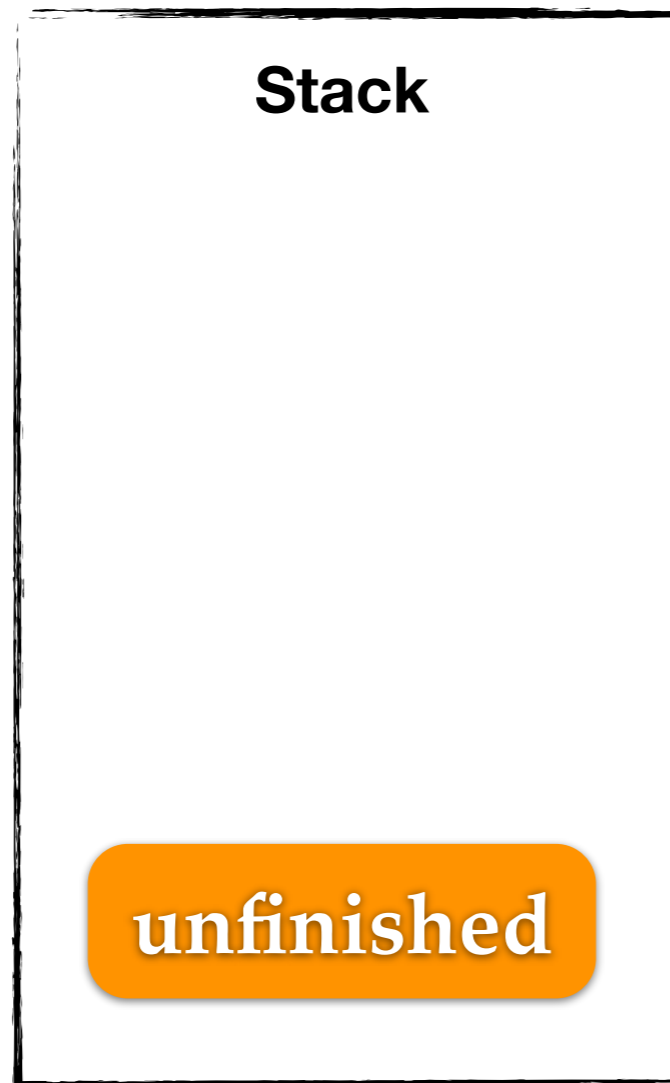
commit 1



Stash



fix a bug, right now!



feature A'



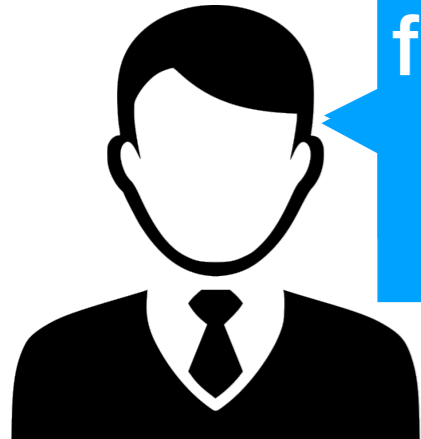
bug fix

commit 2

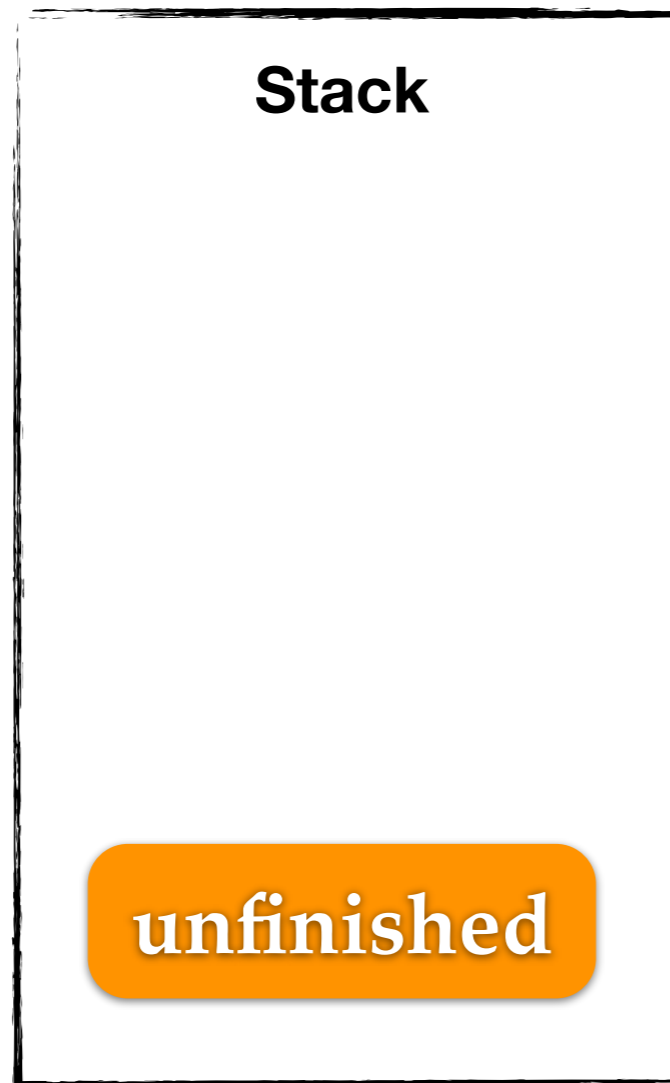
commit 1



Stash



fix a bug, right now!



feature A'

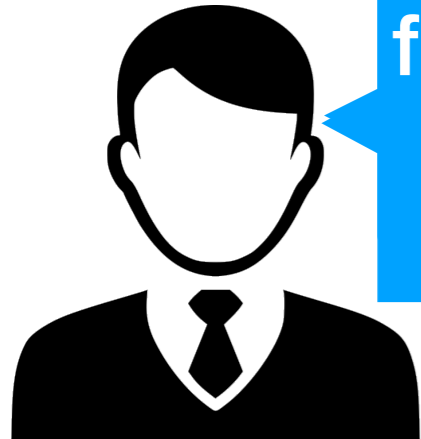
bug fix

commit 2

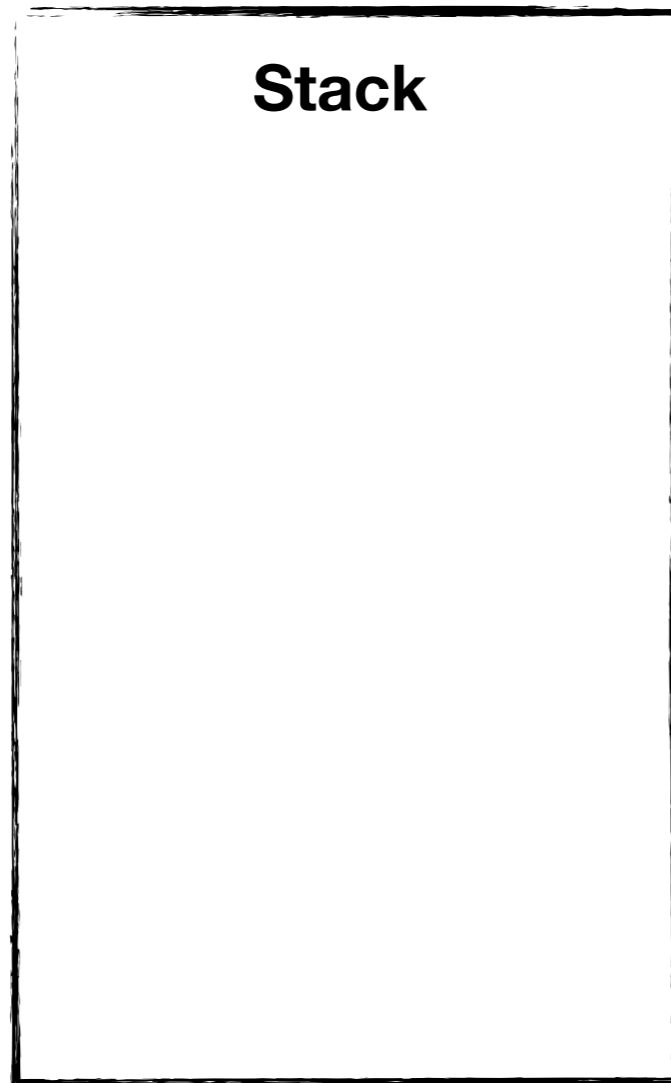
commit 1



Stash



fix a bug, right now!



feature A'

bug fix

commit 2

unfinished'

commit 1



Squash

```
DHCP-22126 squash $ git glog
* 5718716 (HEAD -> master) Improve performance.
* e5abbde Fix a bug in function B.
* be689c1 Add test cases for functions A and B.
* 4960ef6 Fix a bug in function A.
* 697a699 Add function B.
* d7bce04 Add function A.
* aaa944b Add main.c.
DHCP-22126 squash $ █
```



glog is an alias of “log --graph --decorate --pretty=oneline --abbrev-commit”

Squash

```
DHCP-22126 squash $ git glog
* 5718716 (HEAD -> master) Improve performance.
* e5abbde Fix a bug in function B.
* be689c1 Add test cases for functions A and B.
* 4960ef6 Fix a bug in function A.
* 697a699 Add function B.
* d7bce04 Add function A.
* aaa944b Add main.c.
DHCP-22126 squash $ █
```



```
DHCP-22126 squash $ git rebase -i HEAD~6
HEAD: the version of the working tree
```

Squash

```
File: /Users/mht208/demo/squash/.git/rebase-merge/git-rebase-todo
```

```
pick d7bce04 Add function A.  
pick 697a699 Add function B.  
pick 4960ef6 Fix a bug in function A.  
pick be689c1 Add test cases for functions A and B.  
pick e5abbde Fix a bug in function B.  
pick 5718716 Improve performance.
```

```
# Rebase aaa944b..5718716 onto aaa944b (6 commands)  
#  
# Commands:  
# p, pick = use commit  
# r, reword = use commit, but edit the commit message  
# e, edit = use commit, but stop for amending  
# s, squash = use commit, but meld into previous commit  
# f, fixup = like "squash", but discard this commit's log message  
# x, exec = run command (the rest of the line) using shell  
# d, drop = remove commit  
#  
# These lines can be re-ordered; they are executed from top to bottom.  
#
```

Squash

File: /Users/mht208/demo/squash/.git/rebase-merge/git-rebase-todo Modified

```
pick d7bce04 Add function A.
squash 697a699 Add function B.
squash 4960ef6 Fix a bug in function A.
squash be689c1 Add test cases for functions A and B.
squash e5abbde Fix a bug in function B.
squash 5718716 Improve performance.
```

```
# Rebase aaa944b..5718716 onto aaa944b (6 commands)
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
# d, drop = remove commit
#
# These lines can be re-ordered; they are executed from top to bottom.
#
```

Squash

```
GNU nano 2.8.5 File: /Users/mht208/demo/squash/.git/COMMIT_EDITMSG
```

```
# This is a combination of 6 commits.
```

```
# This is the 1st commit message:
```

```
Add function A.
```

```
# This is the commit message #2:
```

```
Add function B.
```

```
# This is the commit message #3:
```

```
Fix a bug in function A.
```

```
# This is the commit message #4:
```

```
Add test cases for functions A and B.
```

```
# This is the commit message #5:
```

```
Fix a bug in function B.
```

Squash

```
GNU nano 2.8.5 File: /Users/mht208/demo/squash/.git/COMMIT_EDITMSG Modified
```

Add feature A. █

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Wed Sep 13 09:36:38 2017 +0800
#
# interactive rebase in progress; onto aaa944b
# Last commands done (6 commands done):
#   squash e5abbde Fix a bug in function B.
#   squash 5718716 Improve performance.
# No commands remaining.
```

Squash

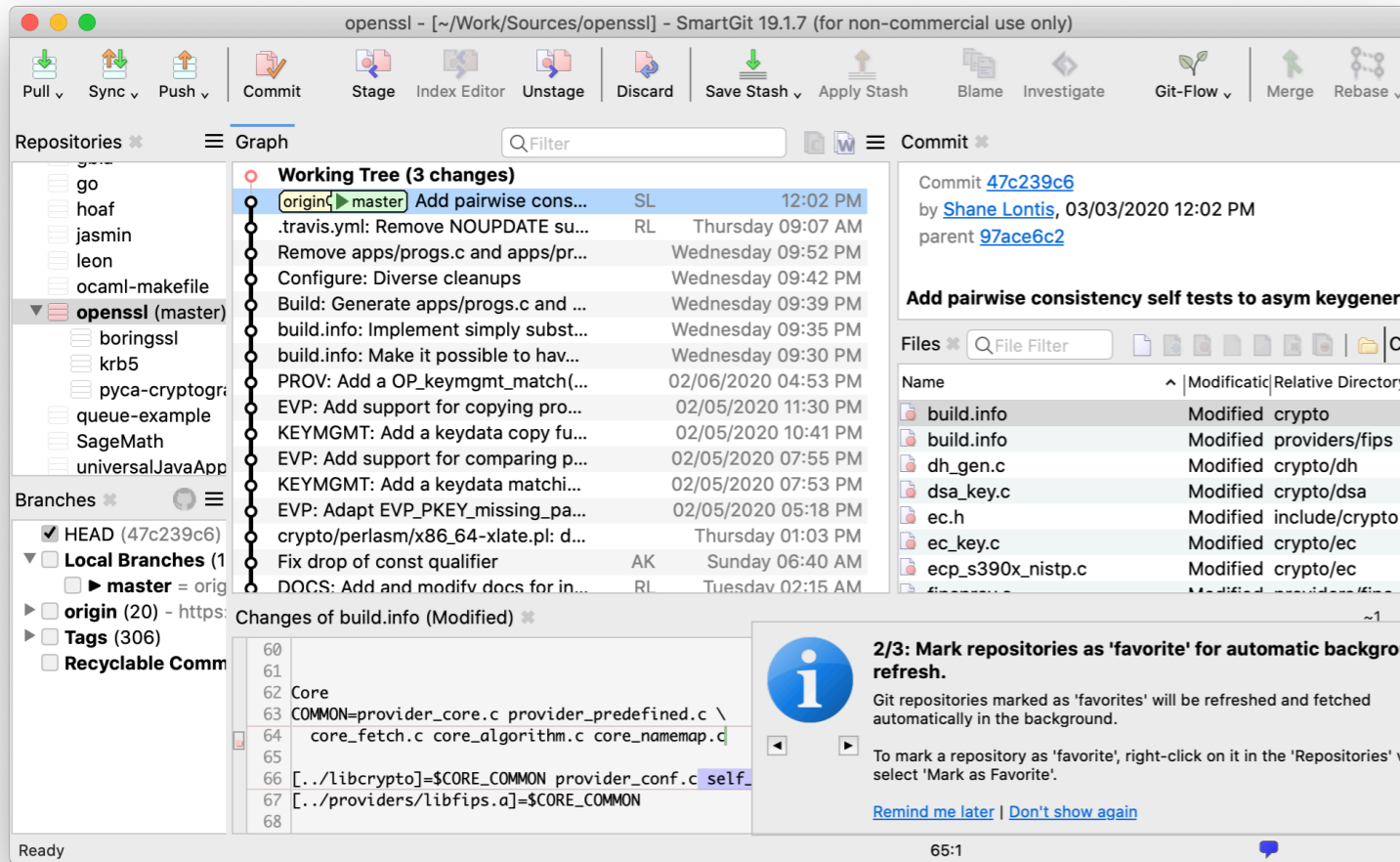
```
DHCP-22126 squash $ git rebase -i HEAD~6
[detached HEAD 4baf53e] Add feature A.
Date: Wed Sep 13 09:36:38 2017 +0800
1 file changed, 12 insertions(+)
Successfully rebased and updated refs/heads/master.
DHCP-22126 squash $ git glog
* 4baf53e (HEAD -> master) Add feature A.
* aaa944b Add main.c.
DHCP-22126 squash $ █
```



Git Hooks

- Git offers the following hooks:
 - Client-side: pre-commit, pre-push, post-merge, ...
 - Server-side: pre-receive, update, post-receive
- Can be used to notify a continuous integration server, update a ticket-tracking system, etc.

GUI Clients

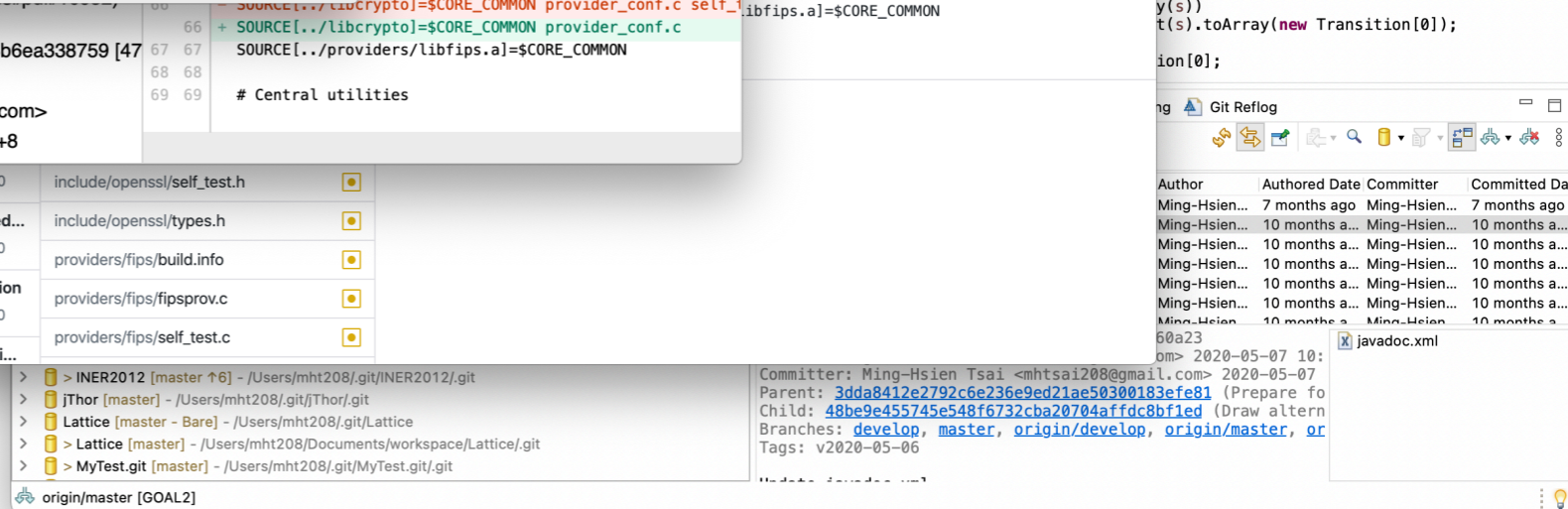


GitHub Desktop

Eclipse EGit

SmartGit

SourceTree



References

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