Git for Developers

Agenda

- Introduction
- Anatomy of a Repository
- Basic Snapshotting
- Going Distributed
- Branching and Merging
- Status and Diff
- Rewriting History
- Workflow

INTRODUCTION

Git

"I'm an egotistical bastard, and I name all my projects after myself. First 'Linux', now 'git'."

Linus Torvalds

The Commands

- init
- add
- commit
- checkout
- branch
- merge
- tag
- diff
- log
- status
- reset
- rm
- clean
- fetch
- pull
- push
- remote
- clone
- rebase
- cherry-pick

Goals of a Version Control System

There are **three** basic goals of a version control system (VCS):

1. We want people to be able to work **simultaneously**, not serially.
2. When people are working **at the same time**, we want their changes to not **conflict** with each other.
3. We want to **archive** every version of everything that has ever existed — ever.

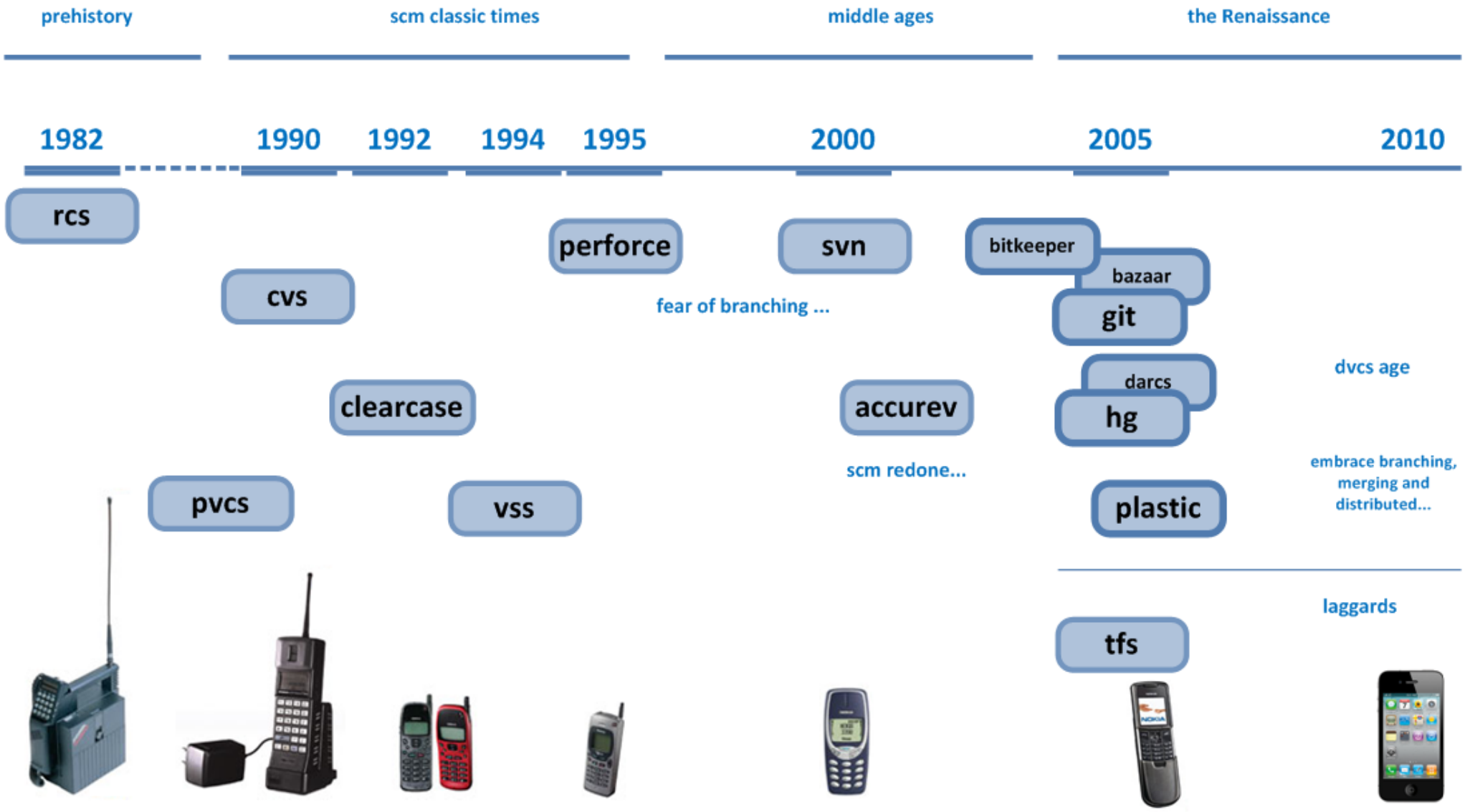
http://www.ericsink.com/vcbe/html/intro.html

A History of Version Control

Generation	Networking	Operations	Concurrency	Examples
First	None	One file at a time	Locks	RCS, SCCS
Second	Centralized	Multi-file	Merge before commit	CVS, SourceSafe, Subversion, Team Foundation Server
Third	Distributed	Changesets	Commit before merge	Bazaar, Git, Mercurial

http://www.ericsink.com/vcbe/html/intro.html

Timeline



http://codicesoftware.blogspot.com/2010/11/version-control-timeline.html

Some of the design goals

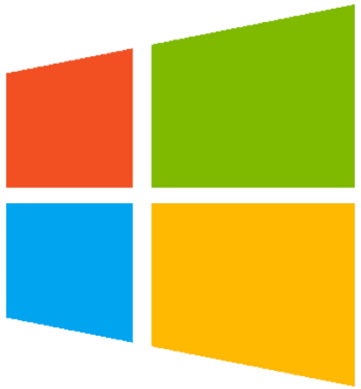
- Speed
- Simple design
- Strong support for non-linear development (thousands of parallel branches)
- Fully distributed
- Able to handle large projects like the Linux kernel efficiently (speed and data size)

http://git-scm.com/book/en/Getting-Started-A-Short-History-of-Git

Git Reference

- http://git-scm.com/docs
- http://gitref.org/index.html
- http://ndpsoftware.com/git-cheatsheet.html
- http://www.git-tower.com/files/cheatsheet/Git_Cheat_Sheet_grey.pdf

Installation



git

GitHub Windows

chocolatey

beta

```
$ apt-get install git
```

User Configuration

- git config --global
  - user.name
  - user.email

Git in the Corporate

- Port 9418 blocked by firewall?
  - git config --global url."**https://**".insteadOf **git://**

  or:

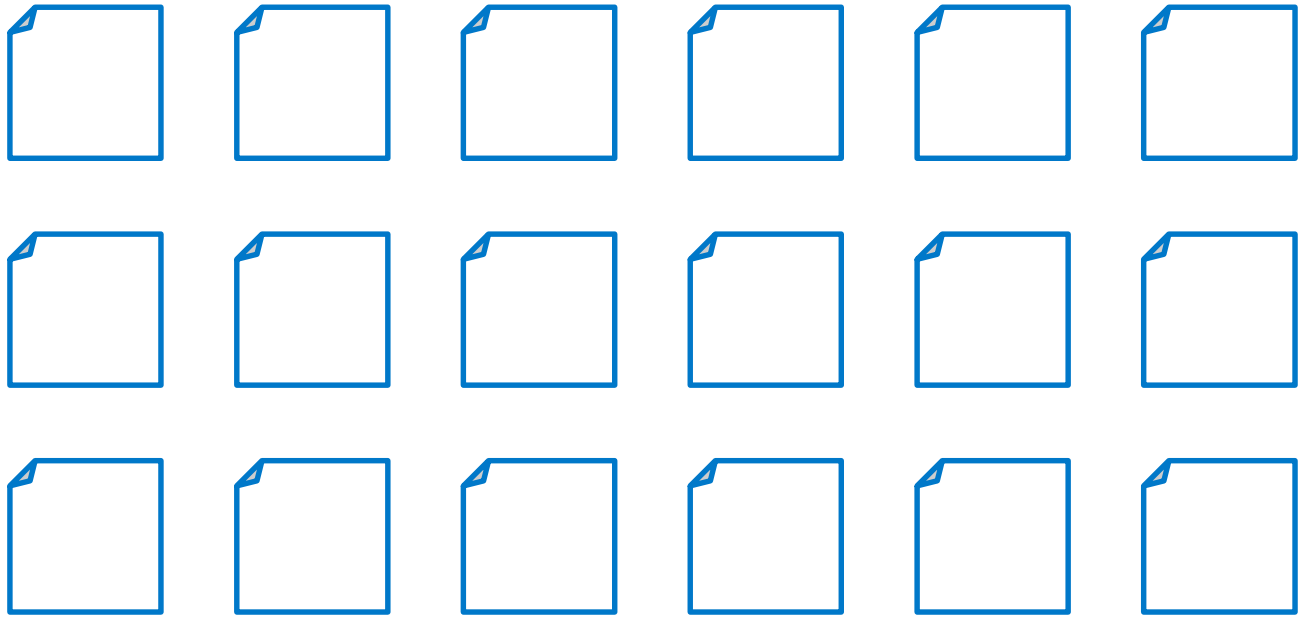
  - git config --global url."**https://github**".insteadOf **git://github**
- Relevant for submodules
  - External repository cloned into subdirectory

http://stackoverflow.com/a/10729634/129269

Anatomy of a Repository

Repository

objects



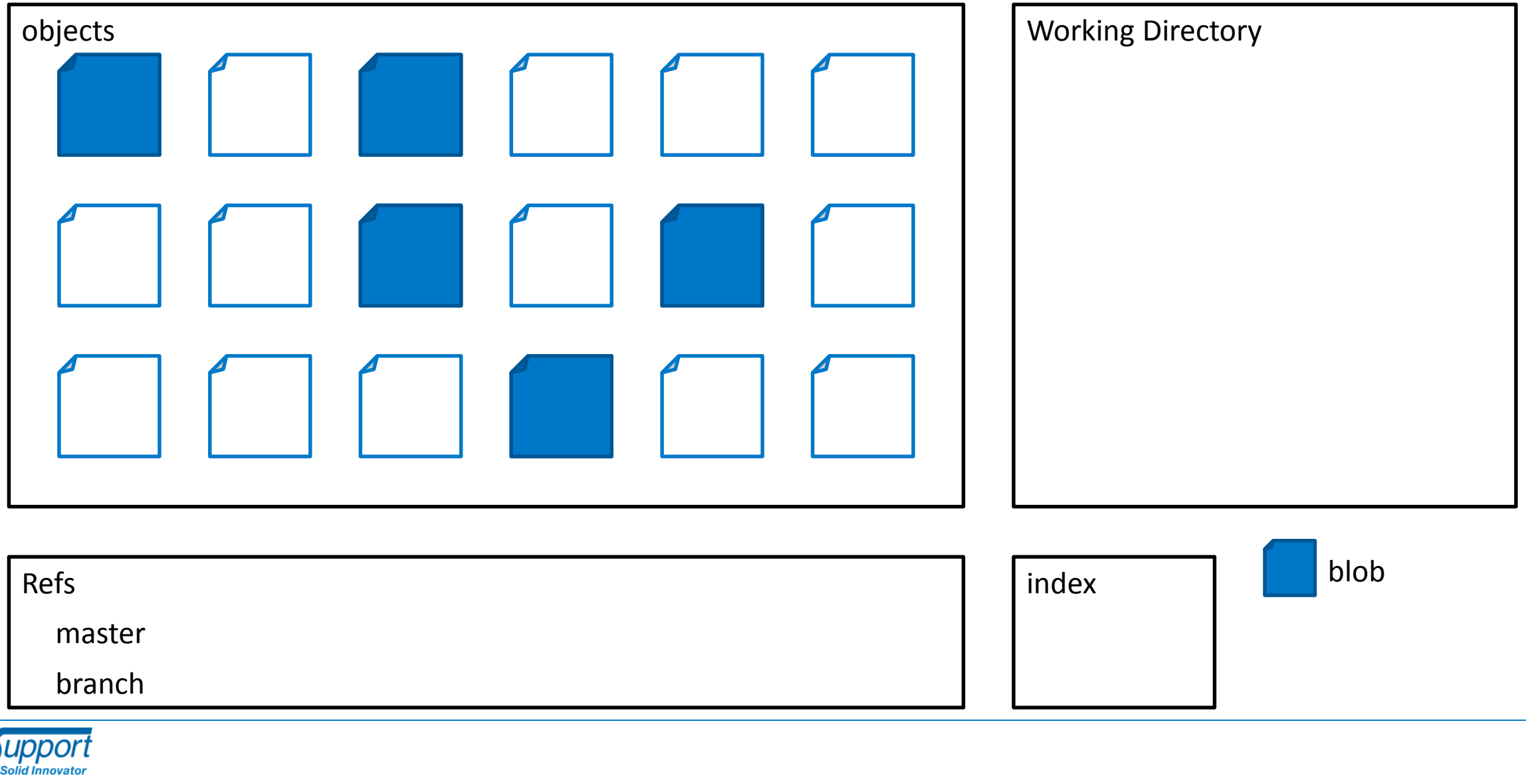
Working Directory

Refs

- master
- branch

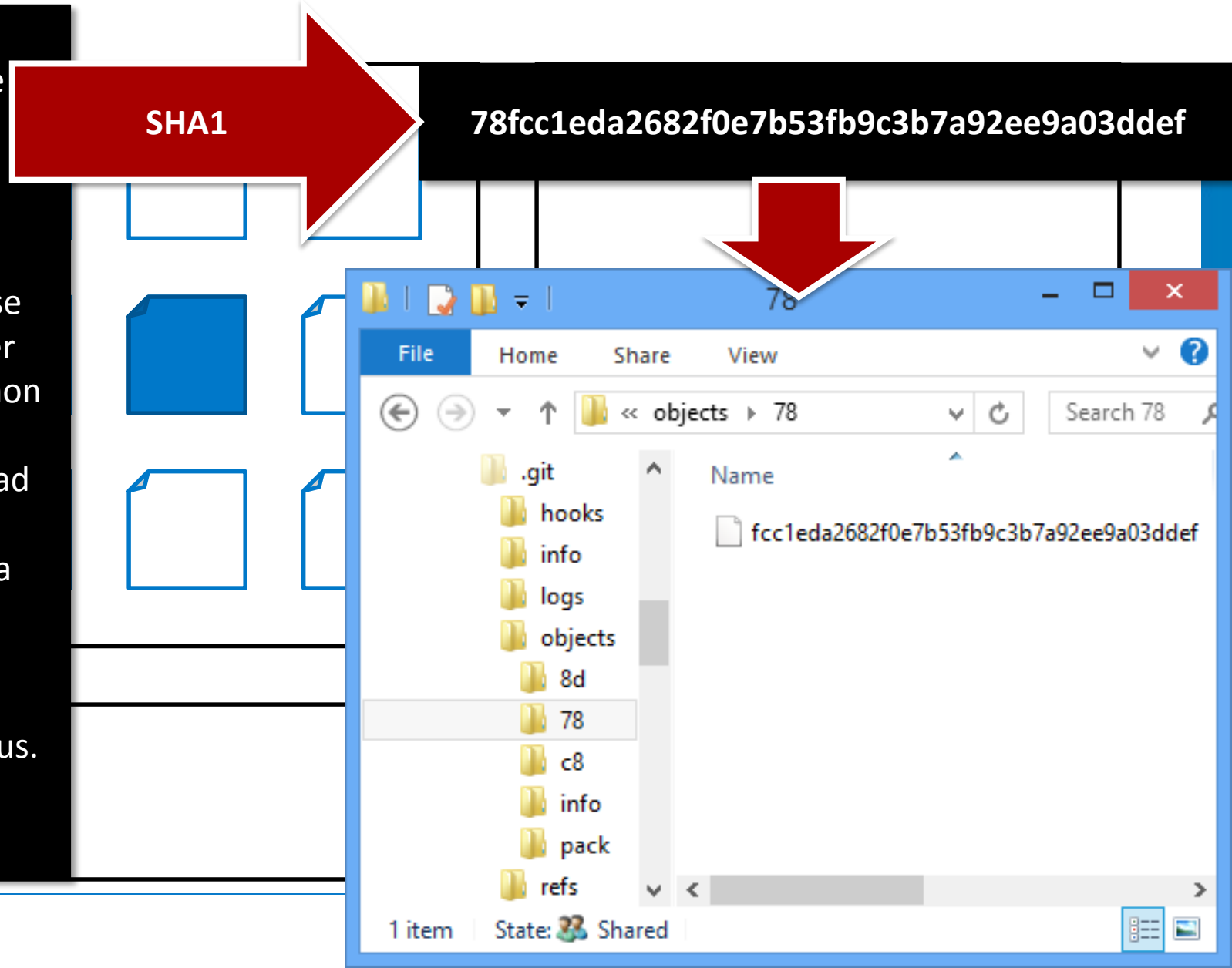
index

Blobs

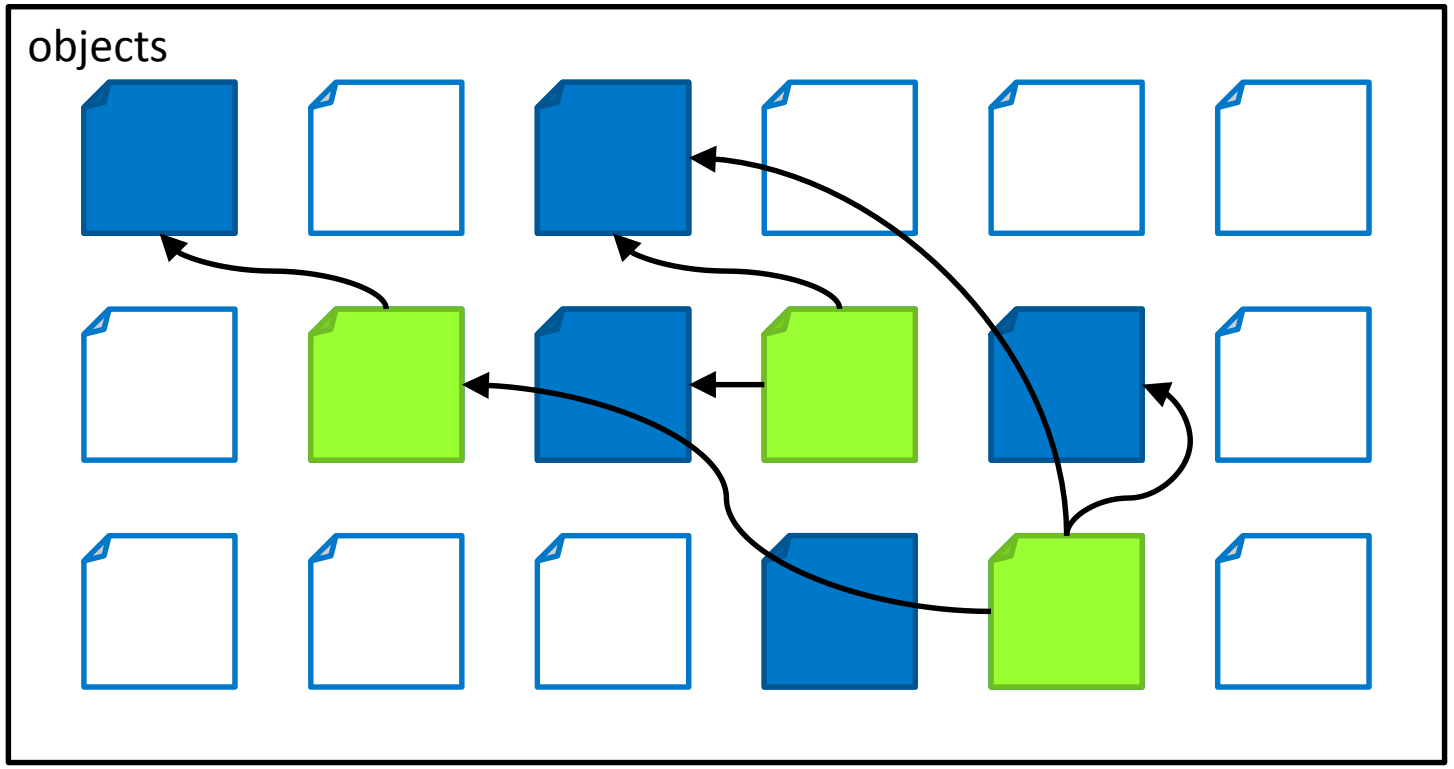


Blobs

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Qui autem esse poteris, nisi te amor ipse ceperit? Atque haec ita iustitiae propria sunt, ut sint virtutum reliquarum communia. Habent enim et bene longam et satis litigiosam disputationem. Duo Reges: constructio interrete. Huic mori optimum esse propter desperationem sapientiae, illi propter spem vivere. Quam ob rem tandem, inquit, non satisfacit? Quod non faceret, si in voluptate summum bonum poneret. Expectoque quid ad id, quod quaerebam, respondeas.
Sed fortuna fortis; De illis, cum volemus. Quia dolori non voluptas contraria est, sed doloris privatio. Quod si ita se habeat, non possit beatam praestare vitam sapientia. Quia nec honesto quic quam honestius nec turpi turpius. Sint ista Graecorum;



Trees



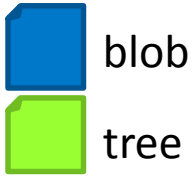
Working Directory

Refs

master

branch

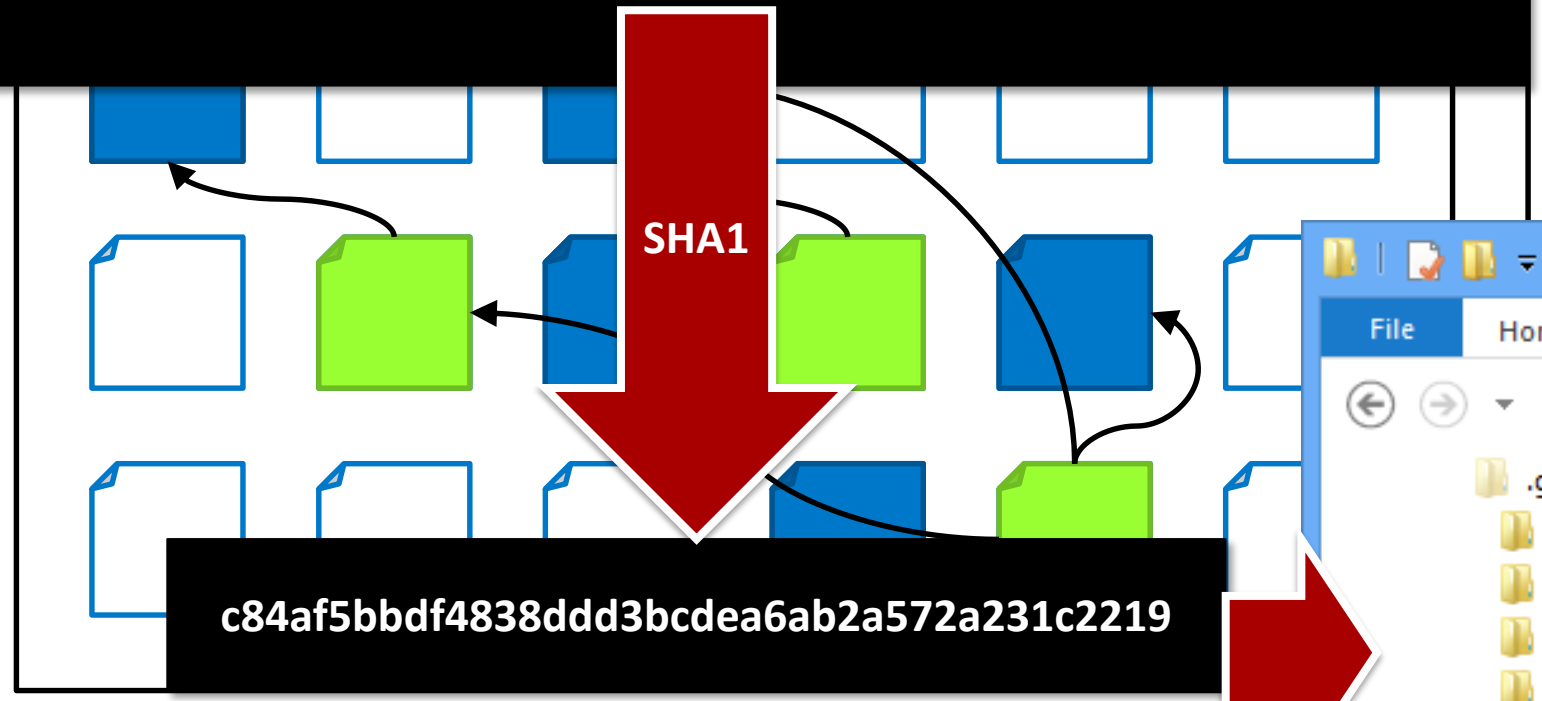
index

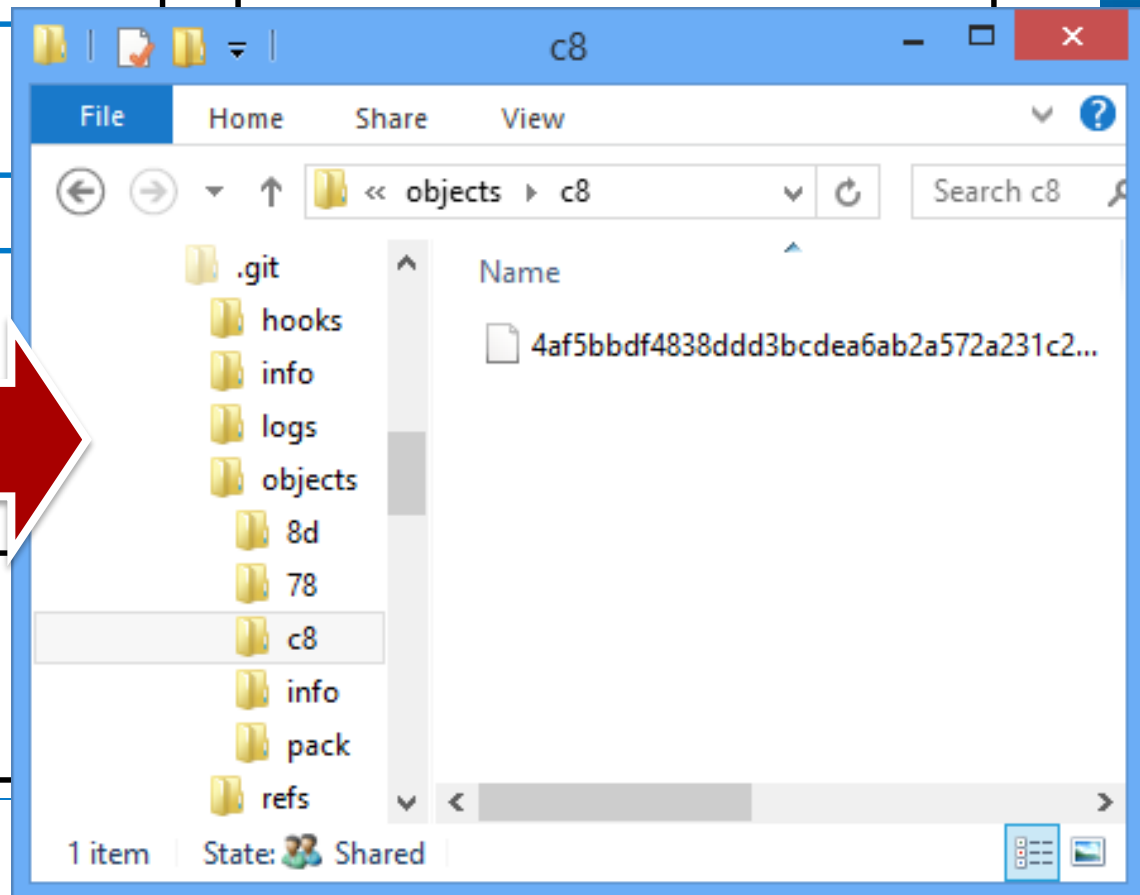


Trees

100644 Lorum Ipsum.txt 78fcc1eda2682f0e7b53fb9c3b7a92ee9a03ddef

Working Directory



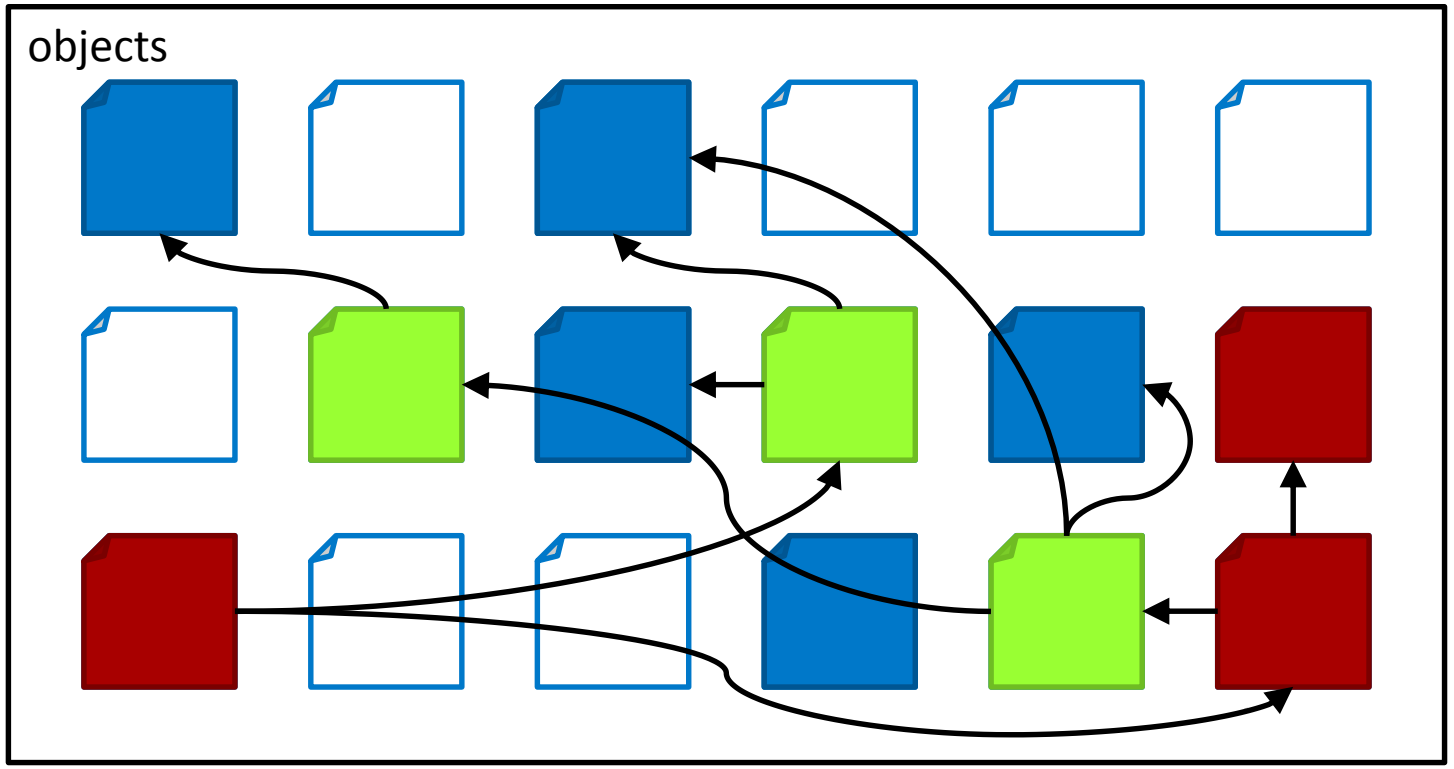


Refs

master

branch

Commits



Working Directory

Refs

master

branch

index

blob

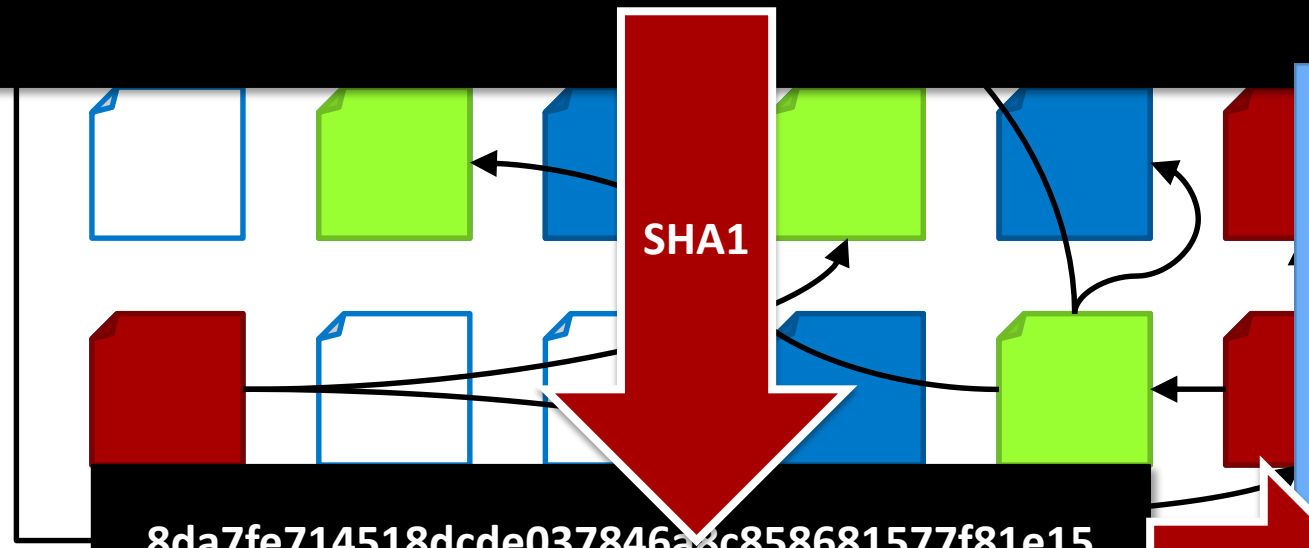
tree

commit

Commits

```
tree c84af5bbdf4838ddd3bcdea6ab2a572a231c2219
author Manuel Riezebosch <mriezebosch@gmail.com>  1378203723 +0200
committer Manuel Riezebosch <mriezebosch@gmail.com>  1378203723 +0200

my first commit
```

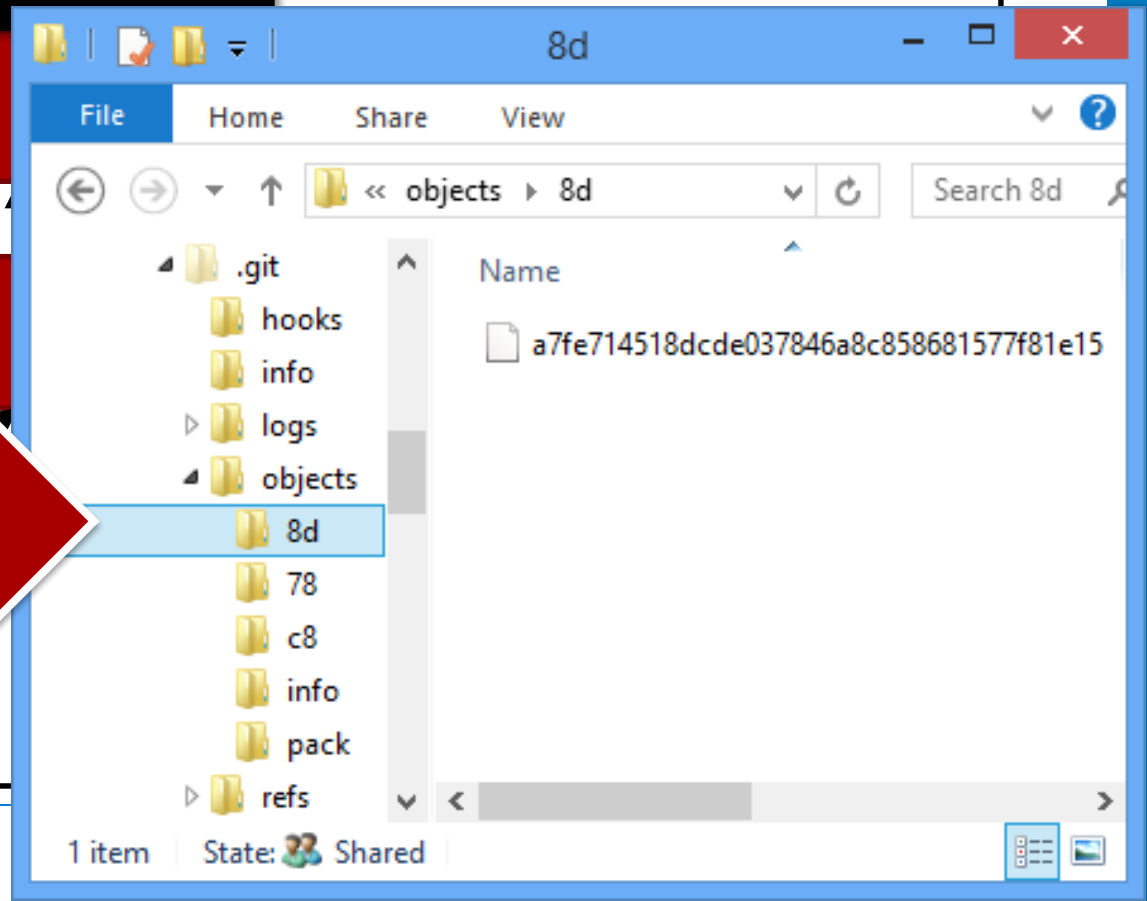


Working Directory

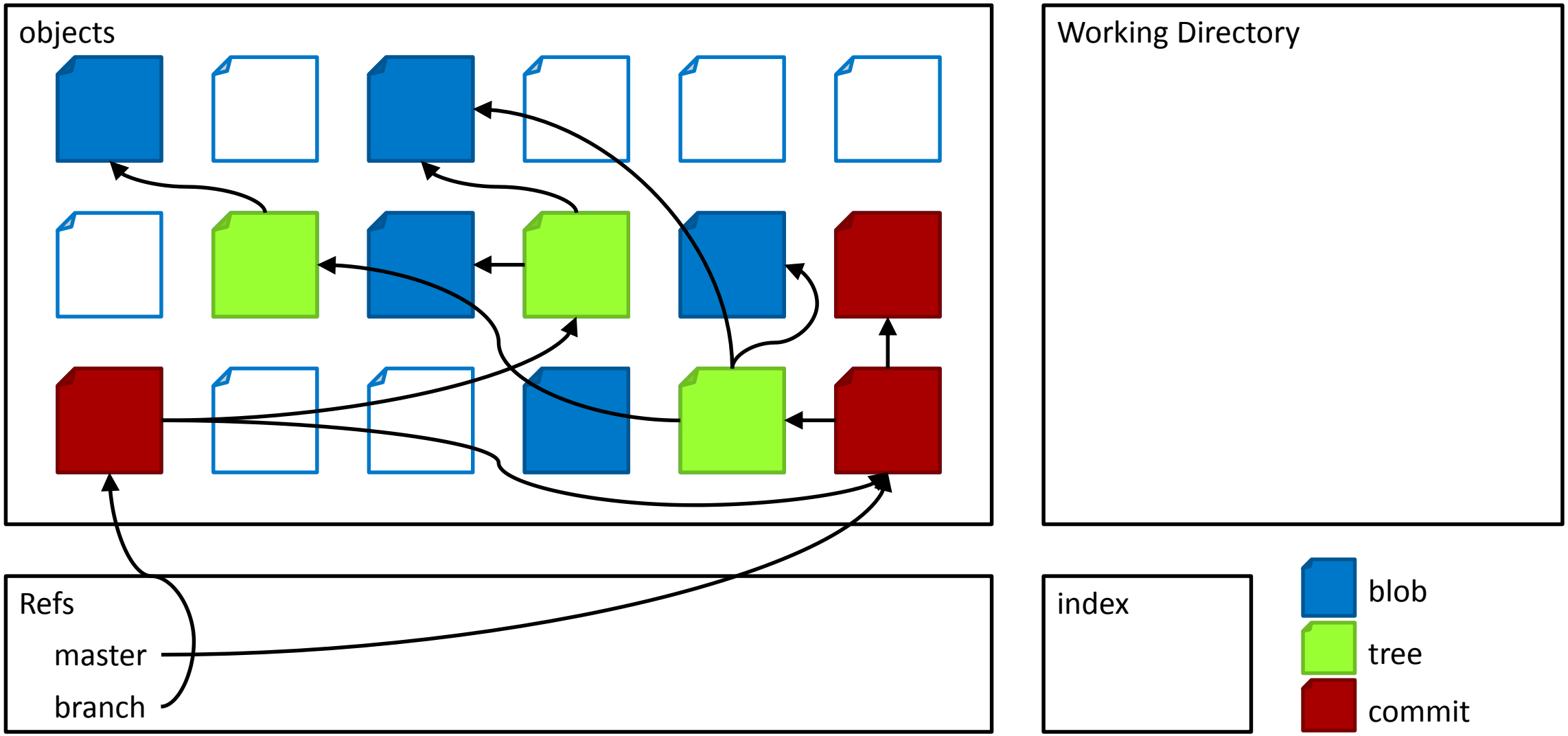
Refs

master

branch

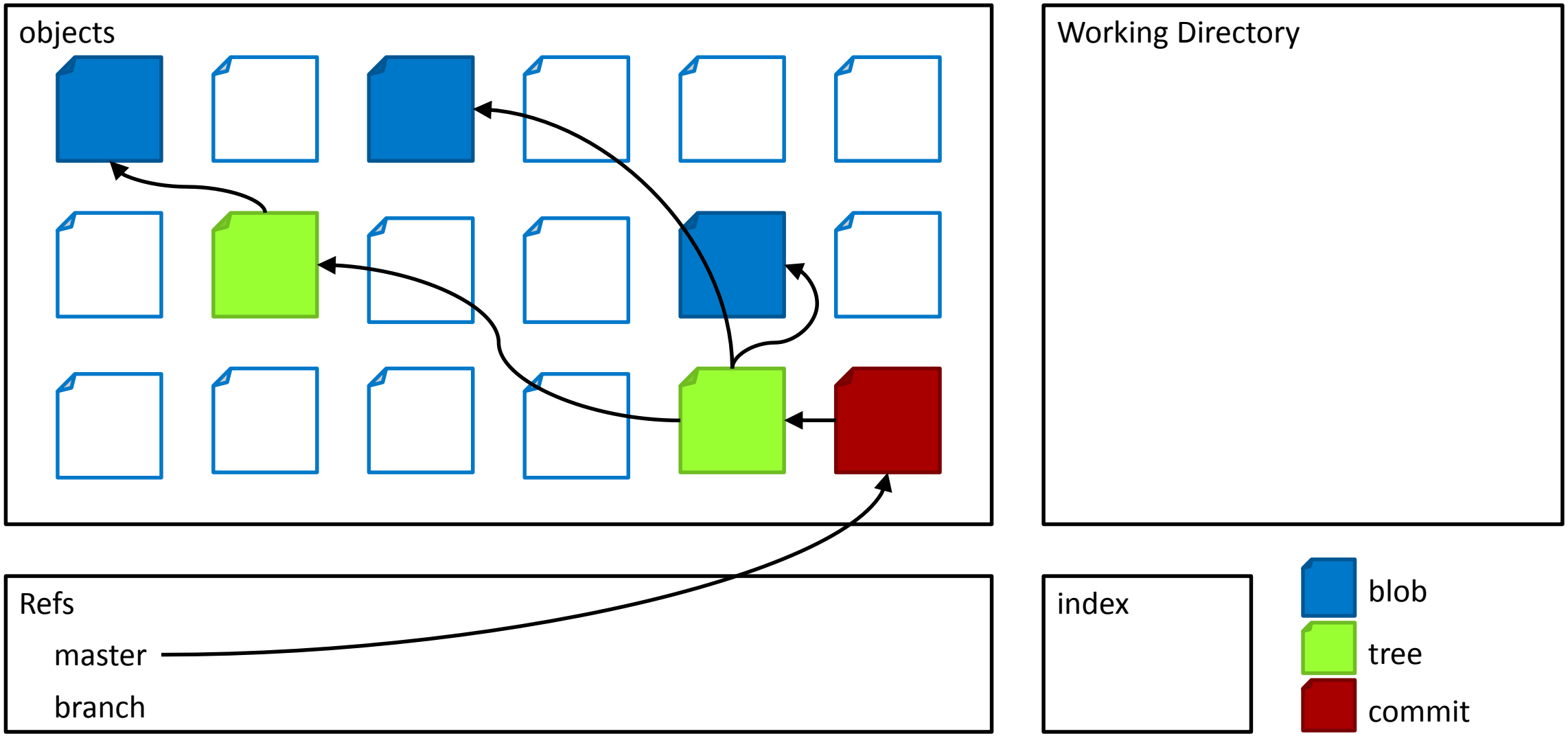


Branches

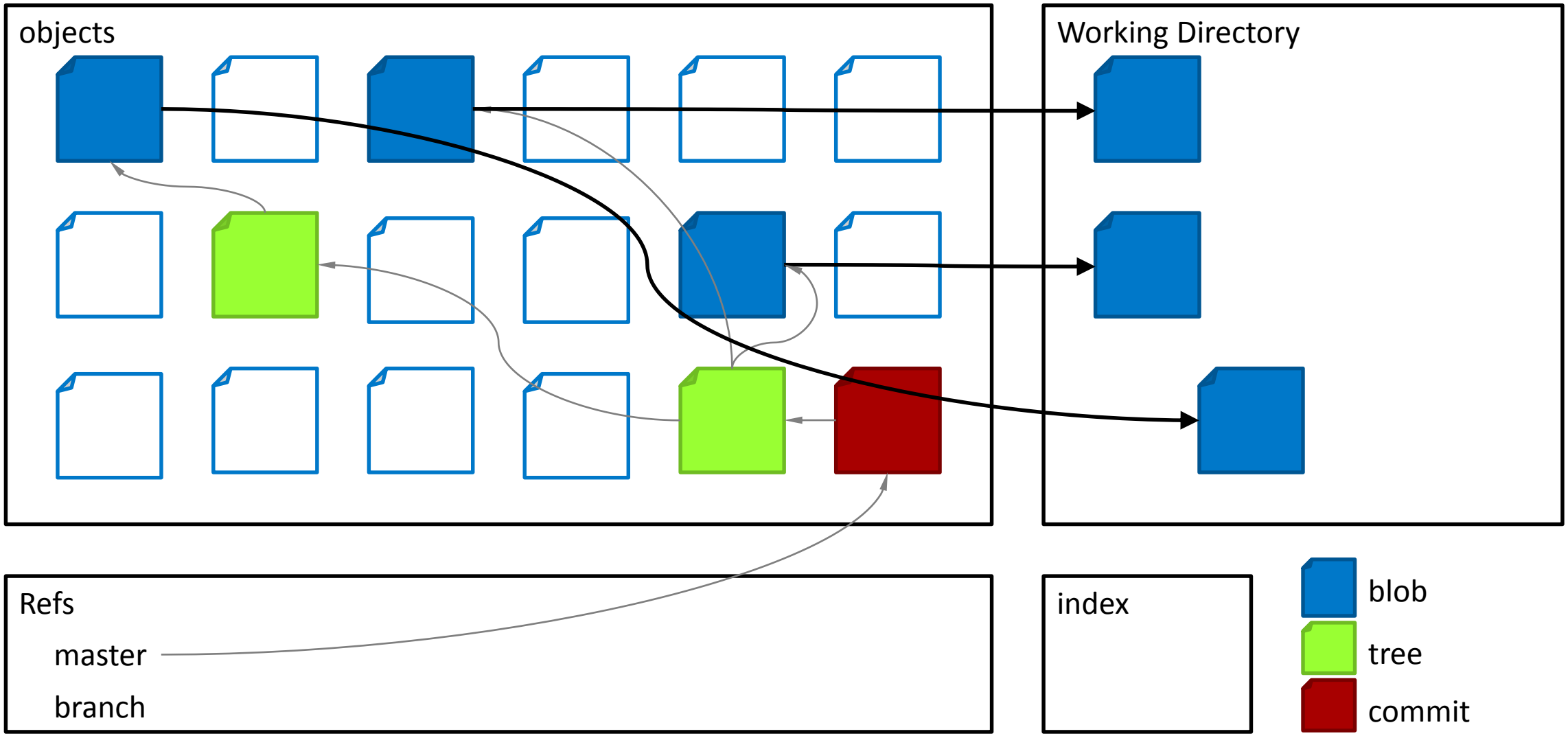

objects
Working Directory
Refs
master
branch
index
blob
tree
commit

Basic Snapshotting

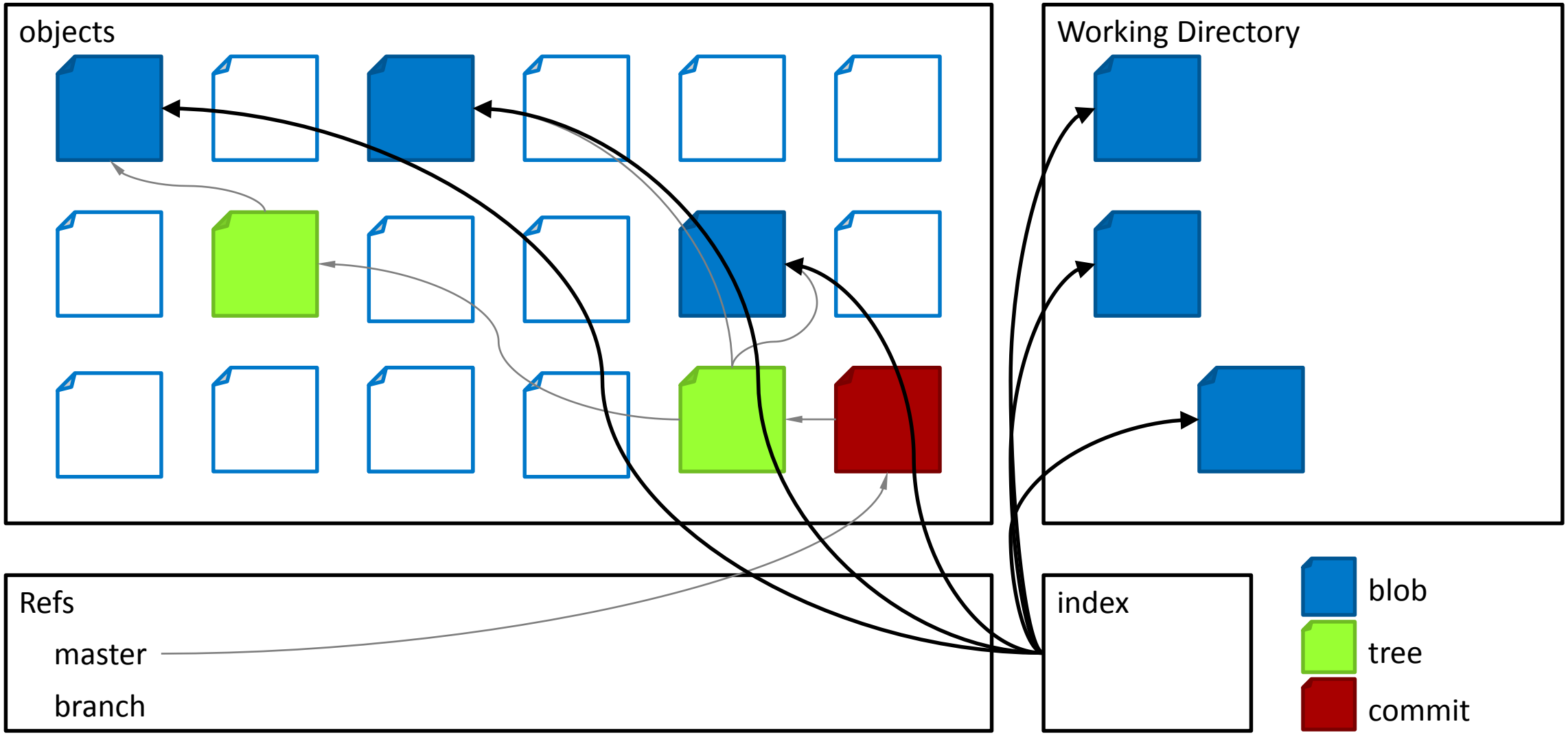
checkout – copy files



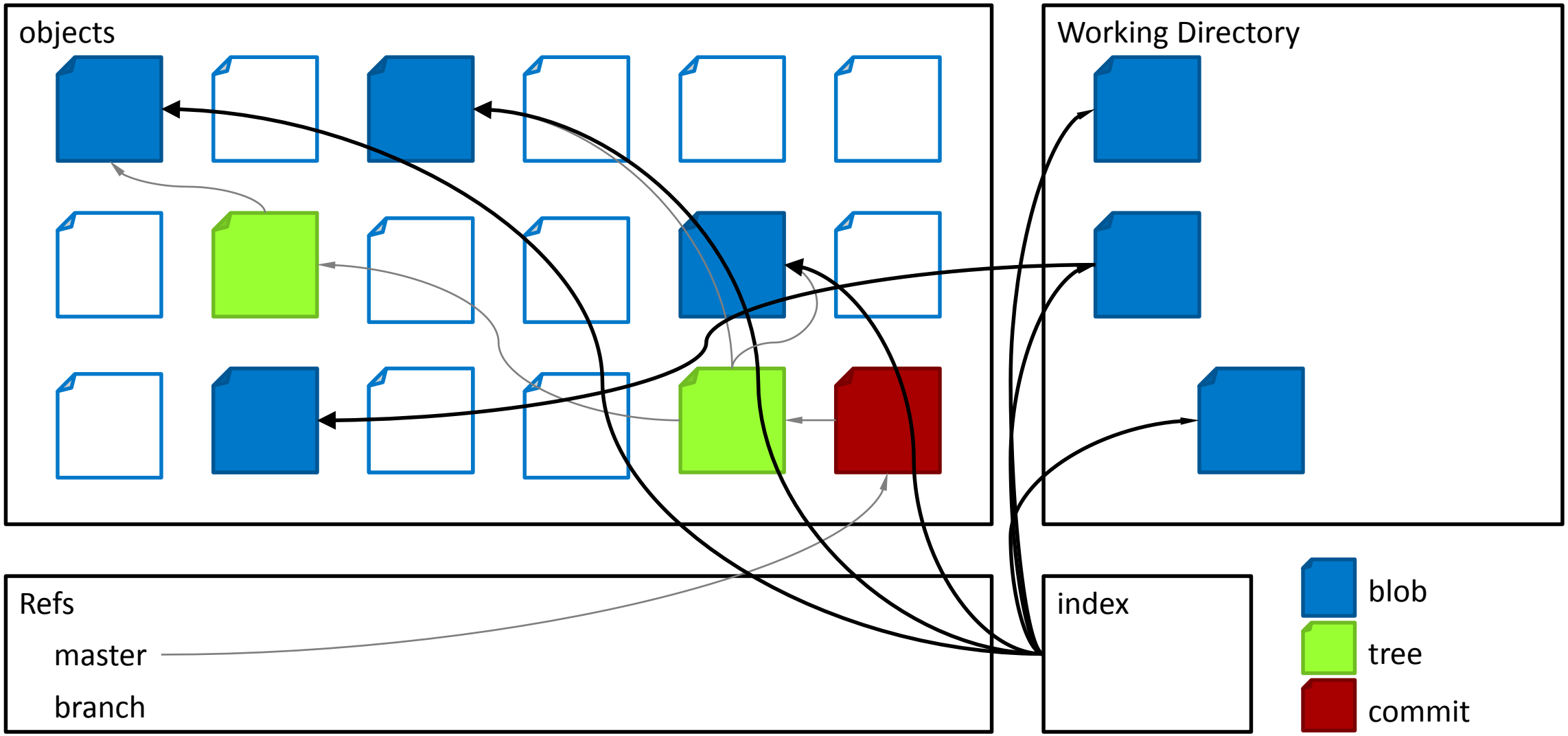
checkout – copy files



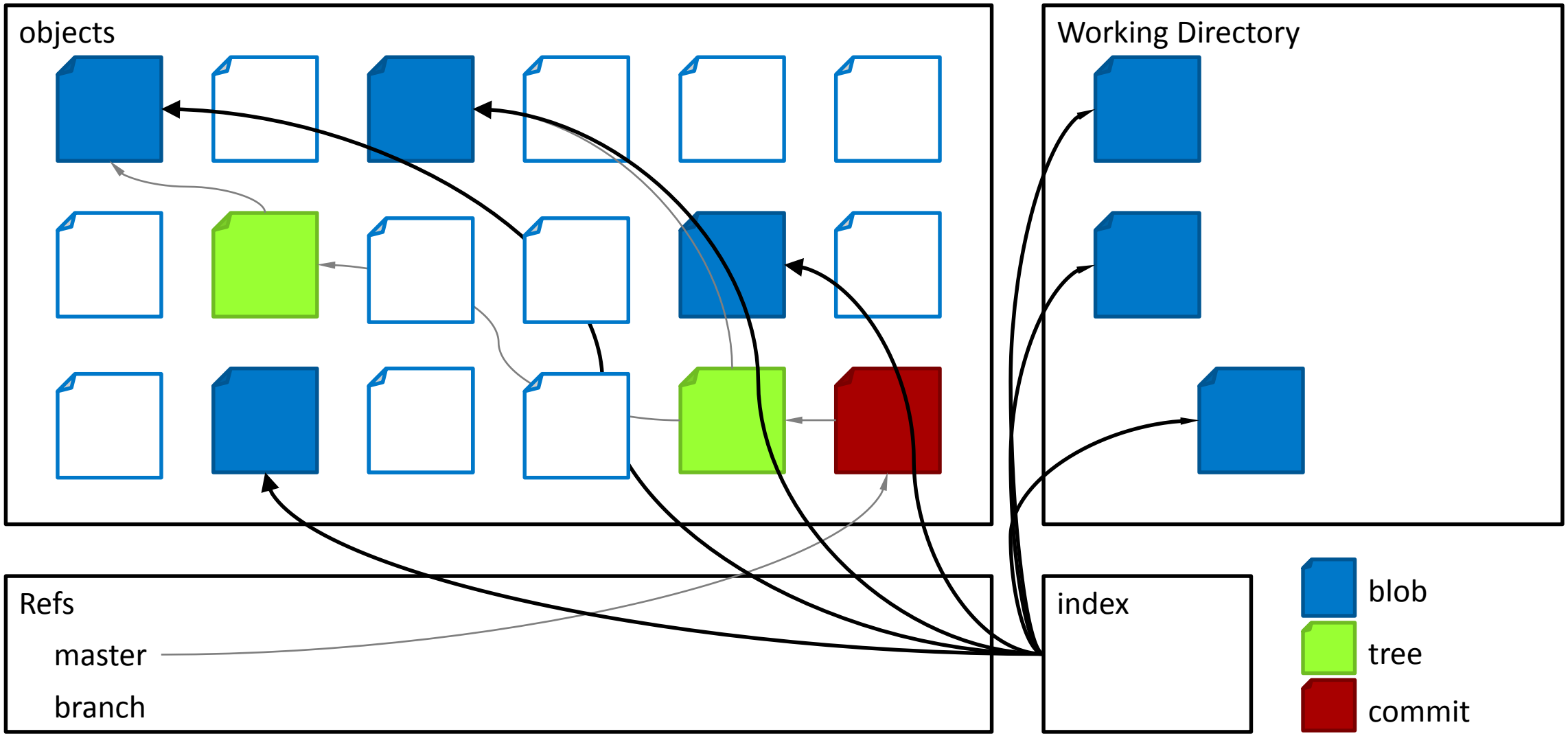
checkout – update index



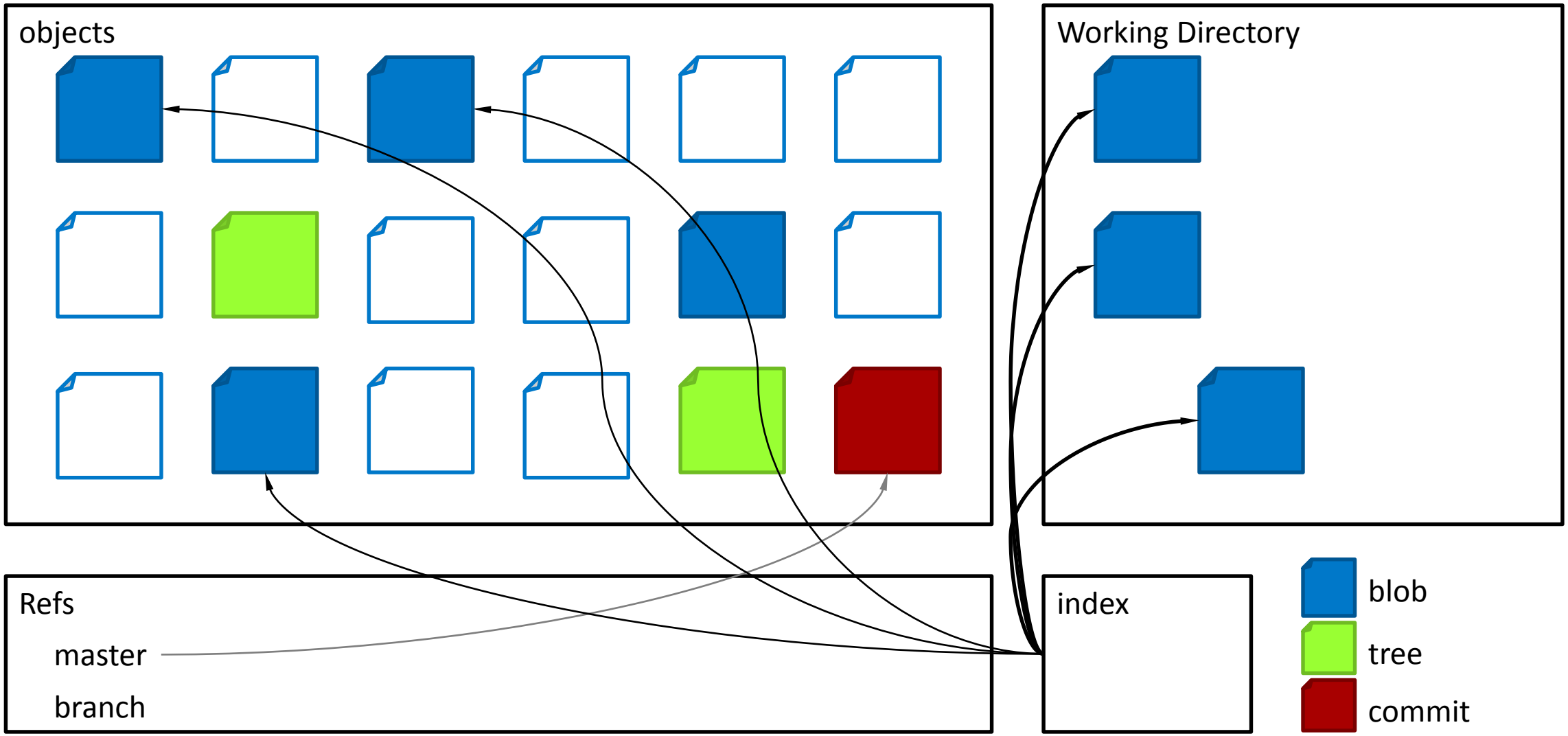
add – copy new/modified file(s)



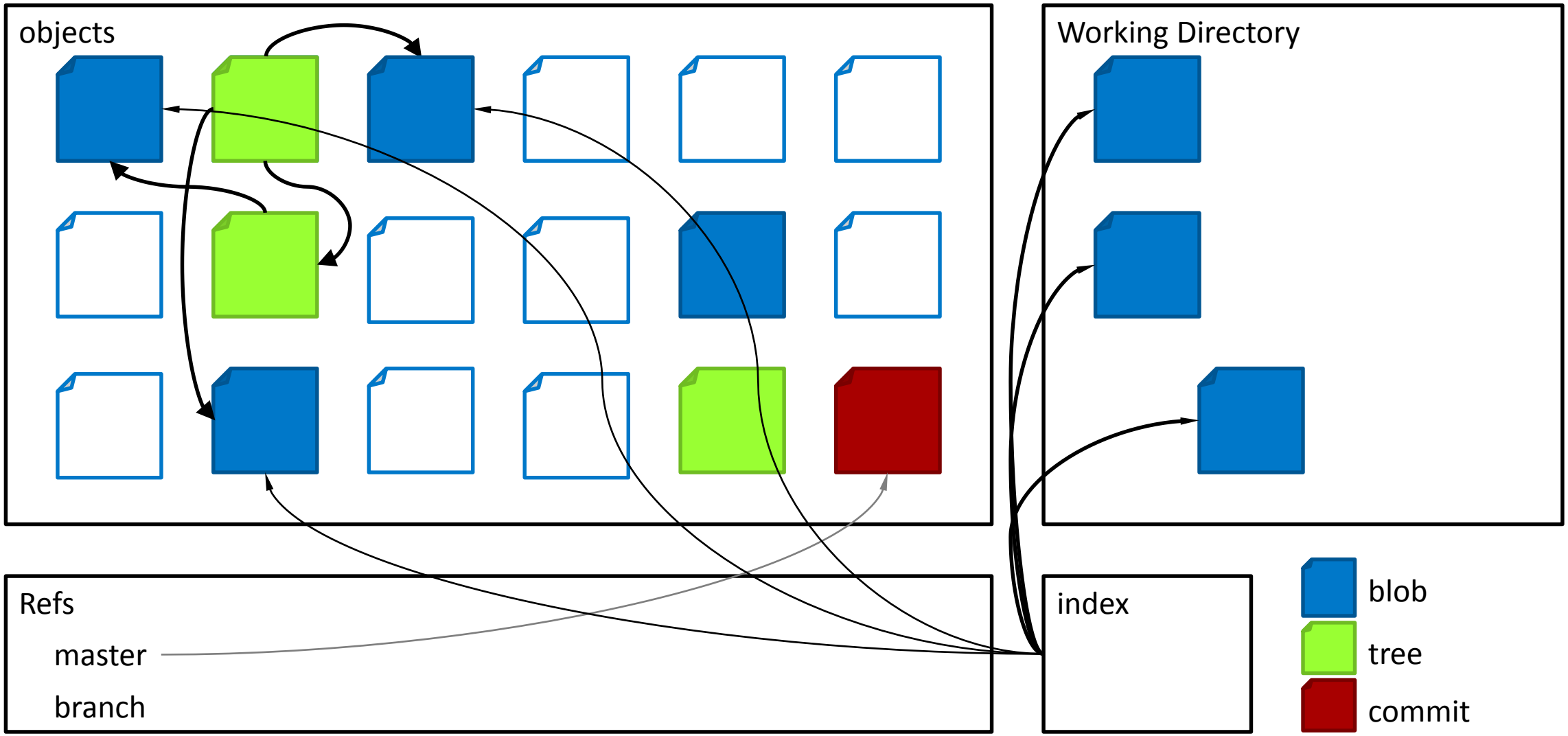
add – update index



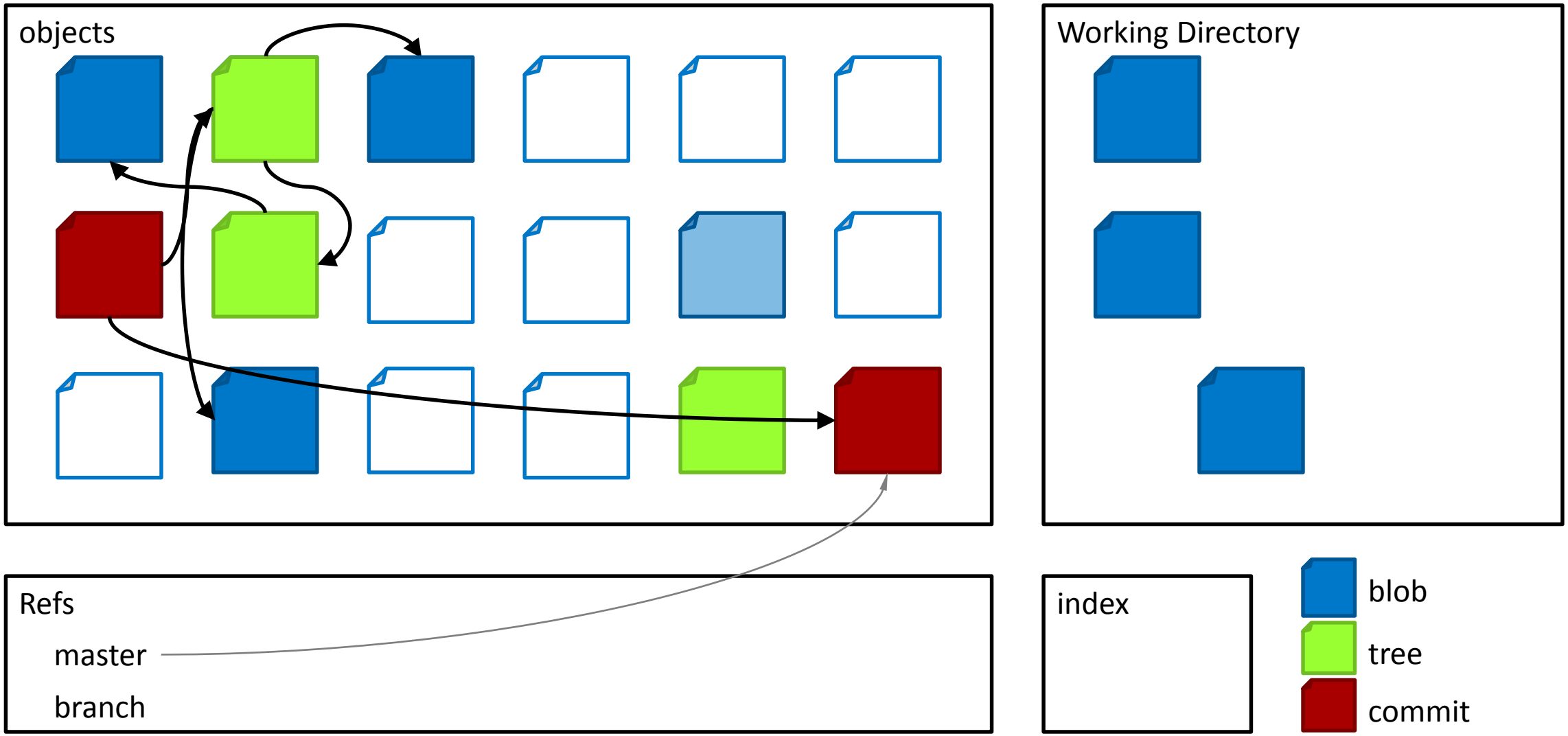
commit


objects
Working Directory
Refs
master
branch
index
blob
tree
commit

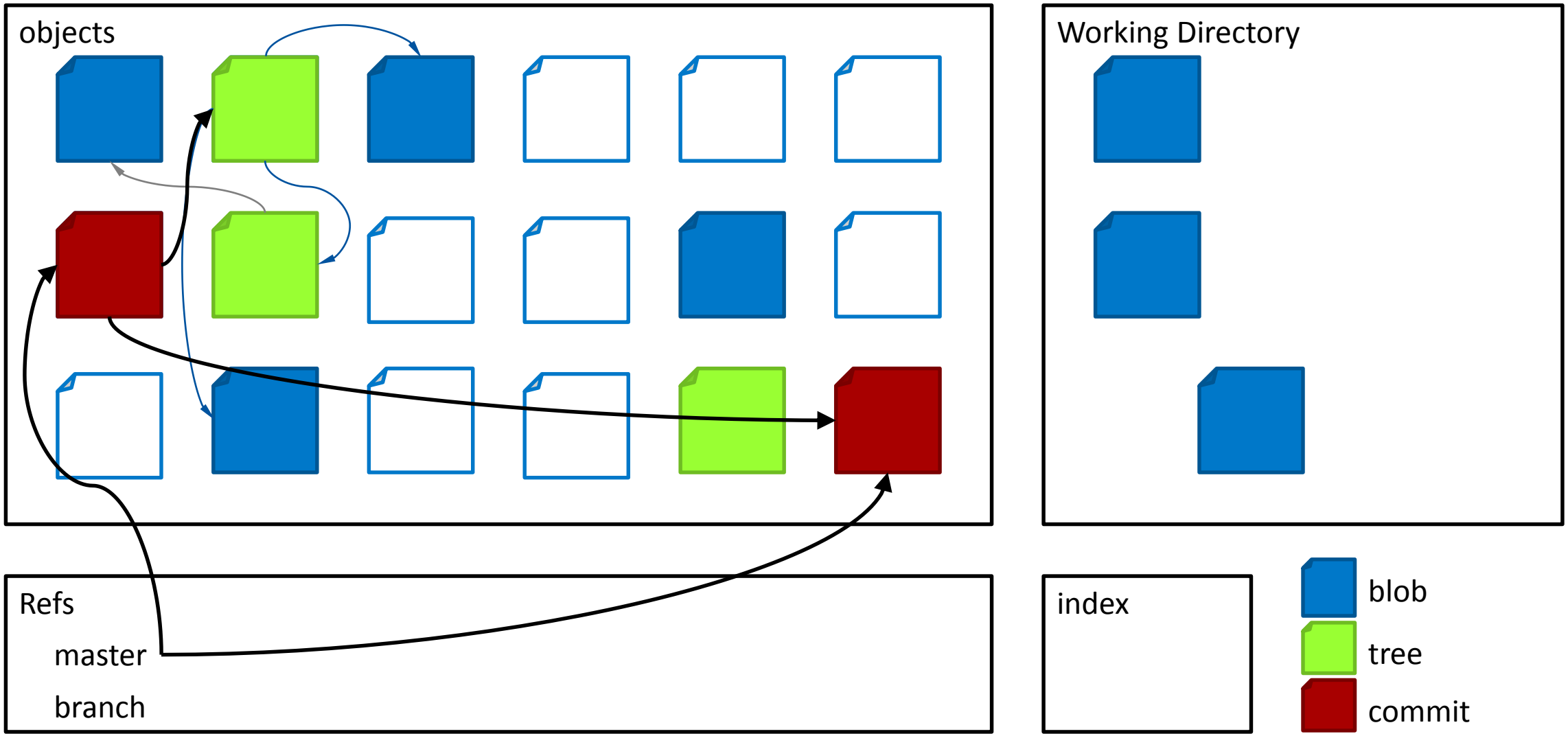
commit – create tree



commit – create commit



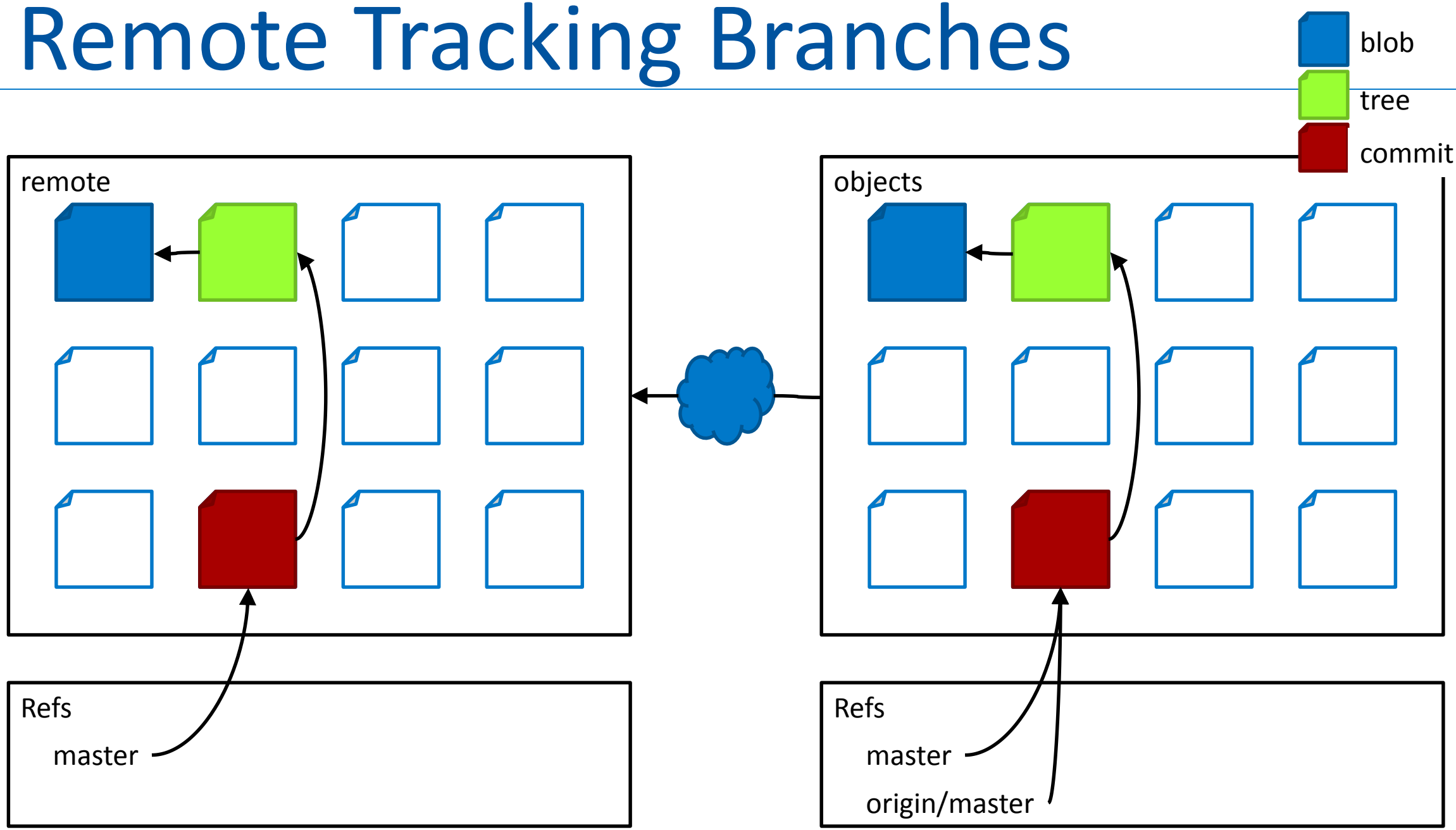
commit – update current branch



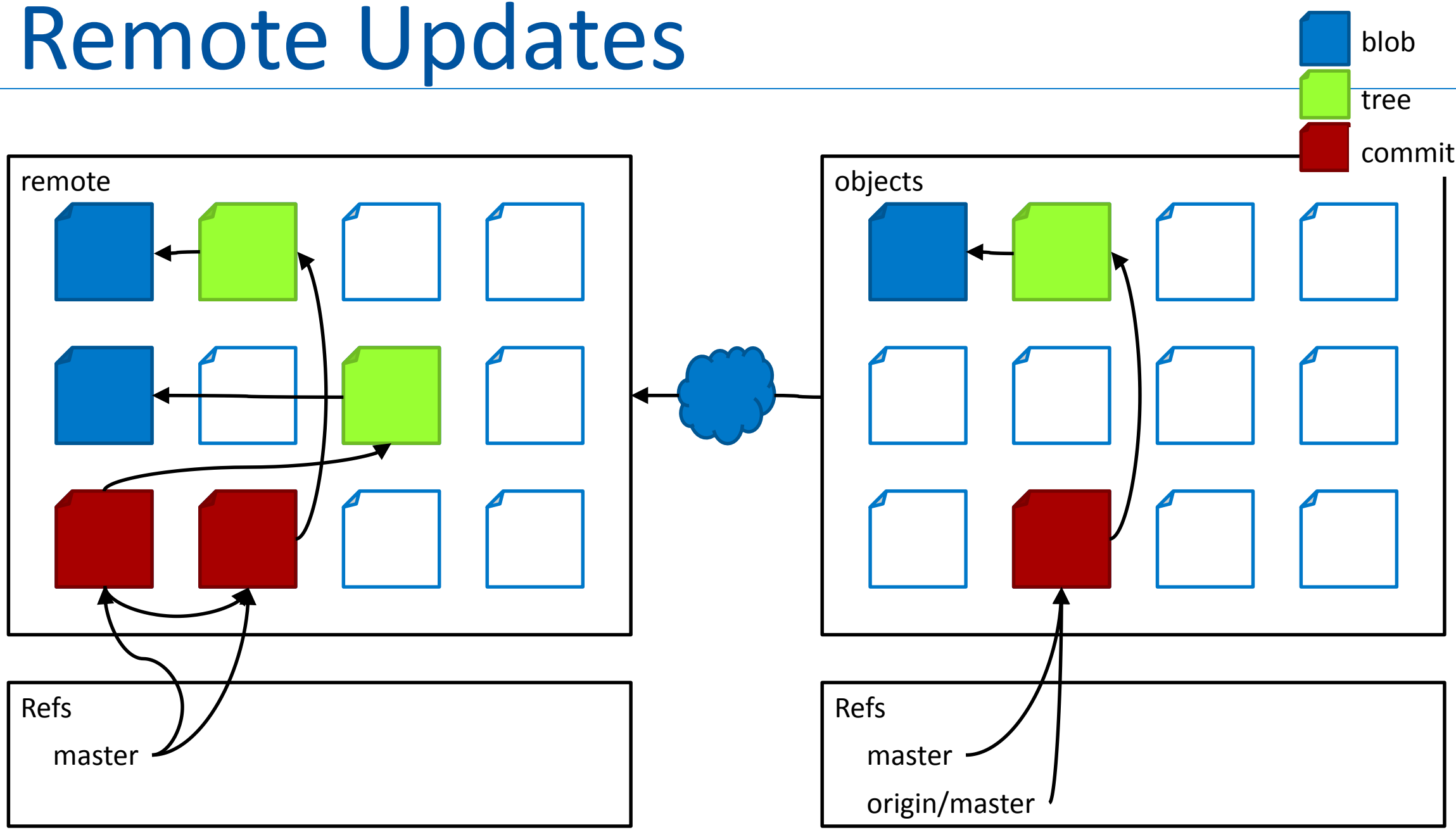
Going Distributed

Remote Tracking Branches

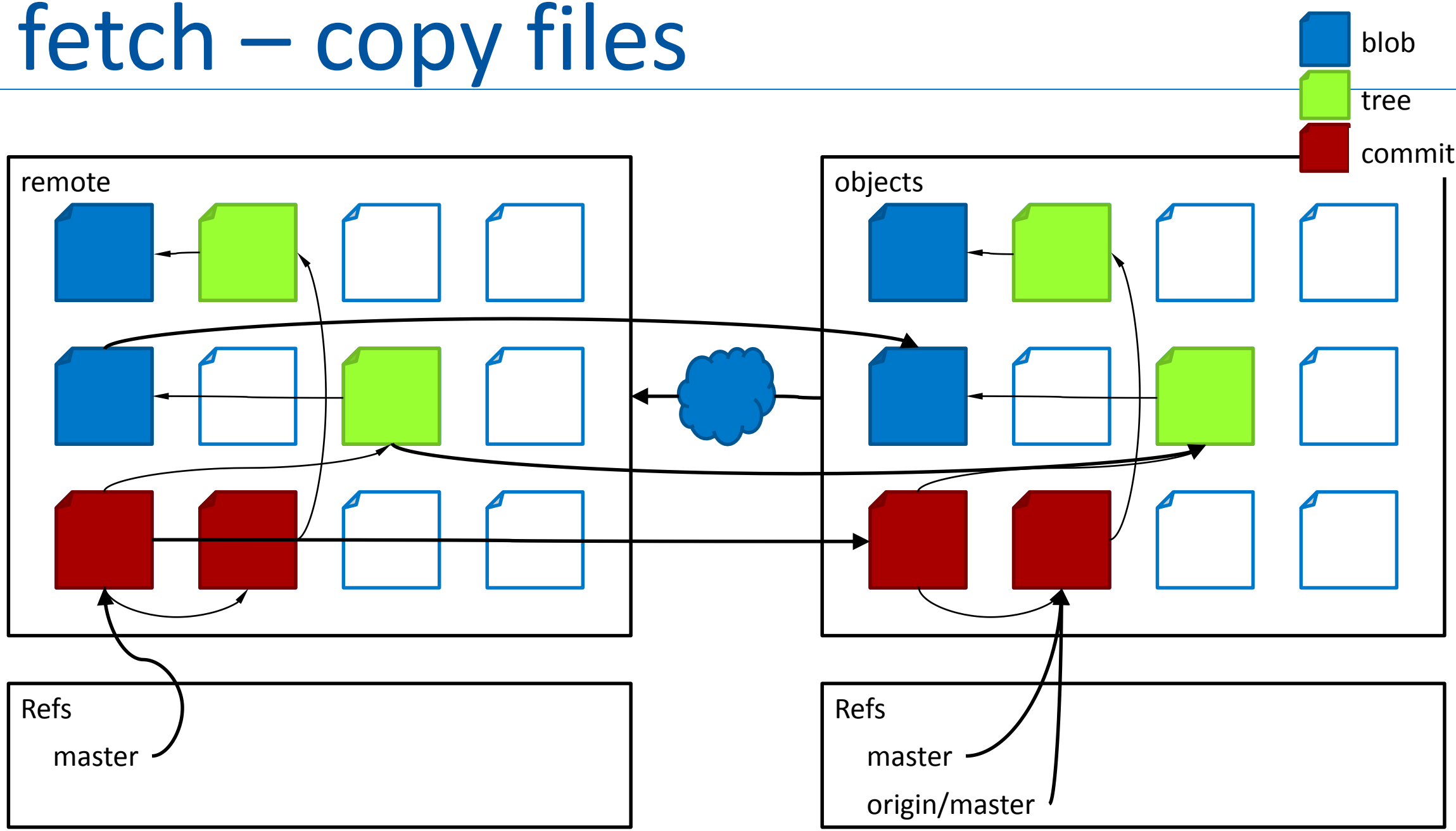
Remote Tracking Branches



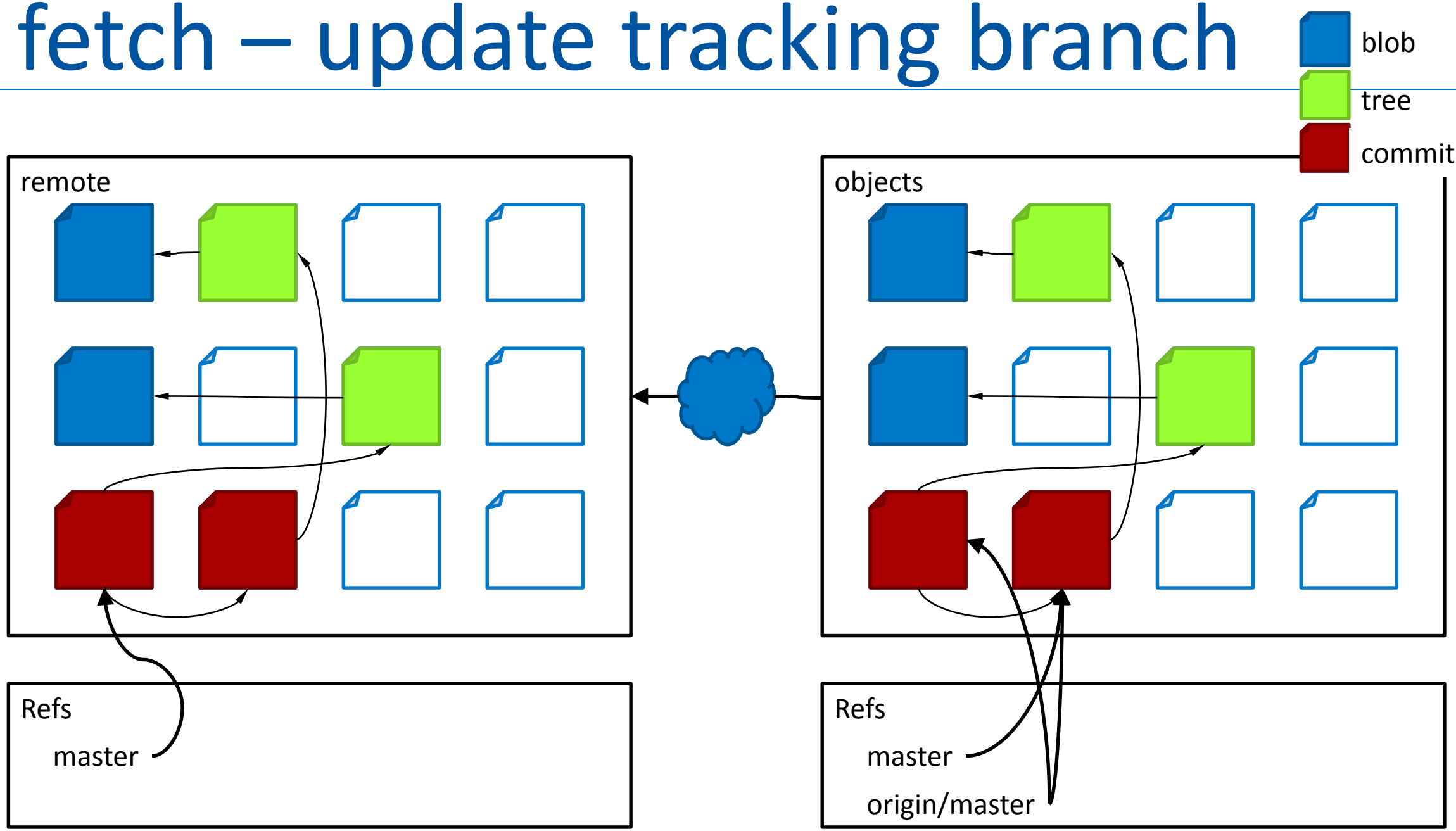
Remote Updates



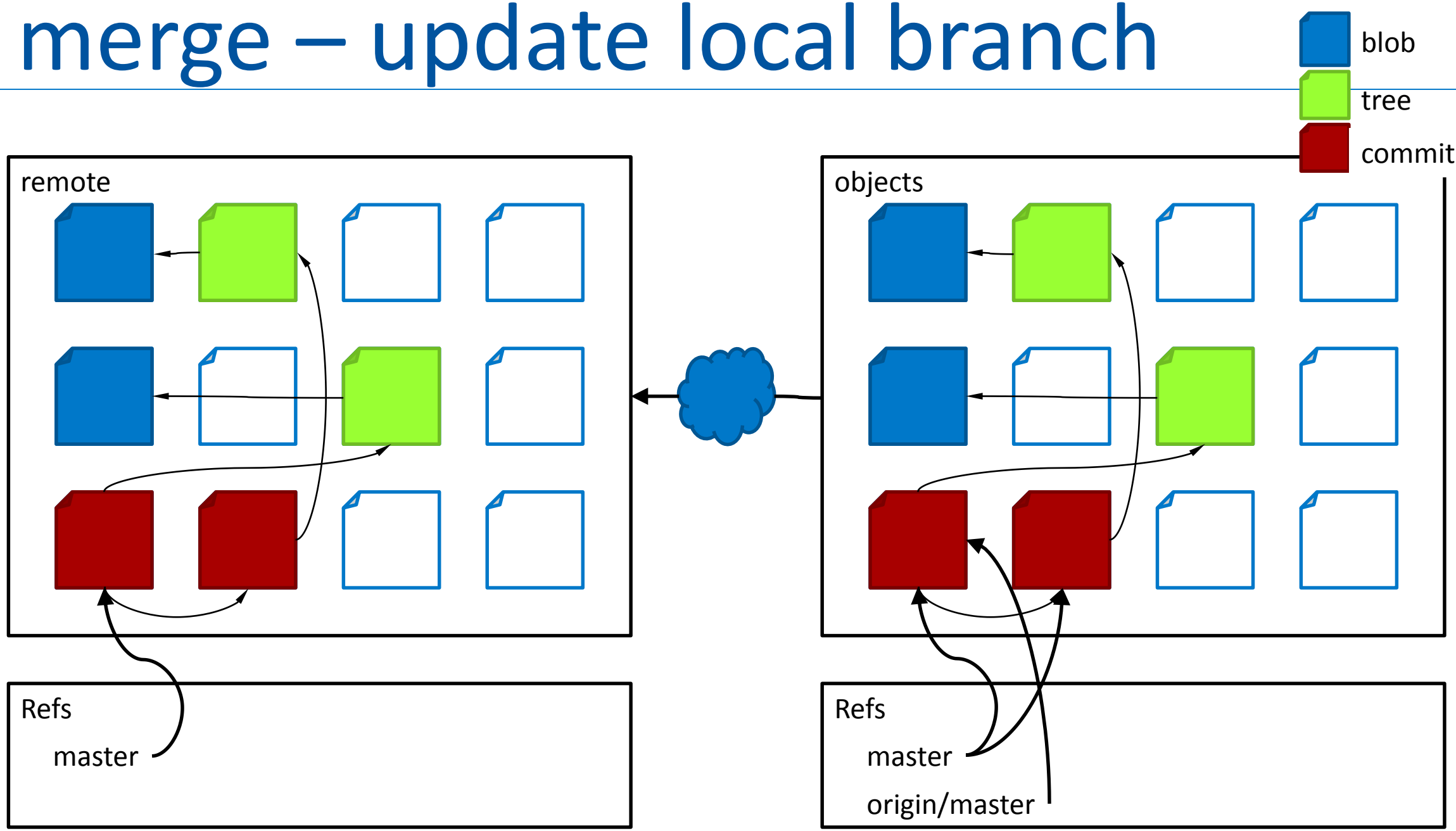
fetch – copy files



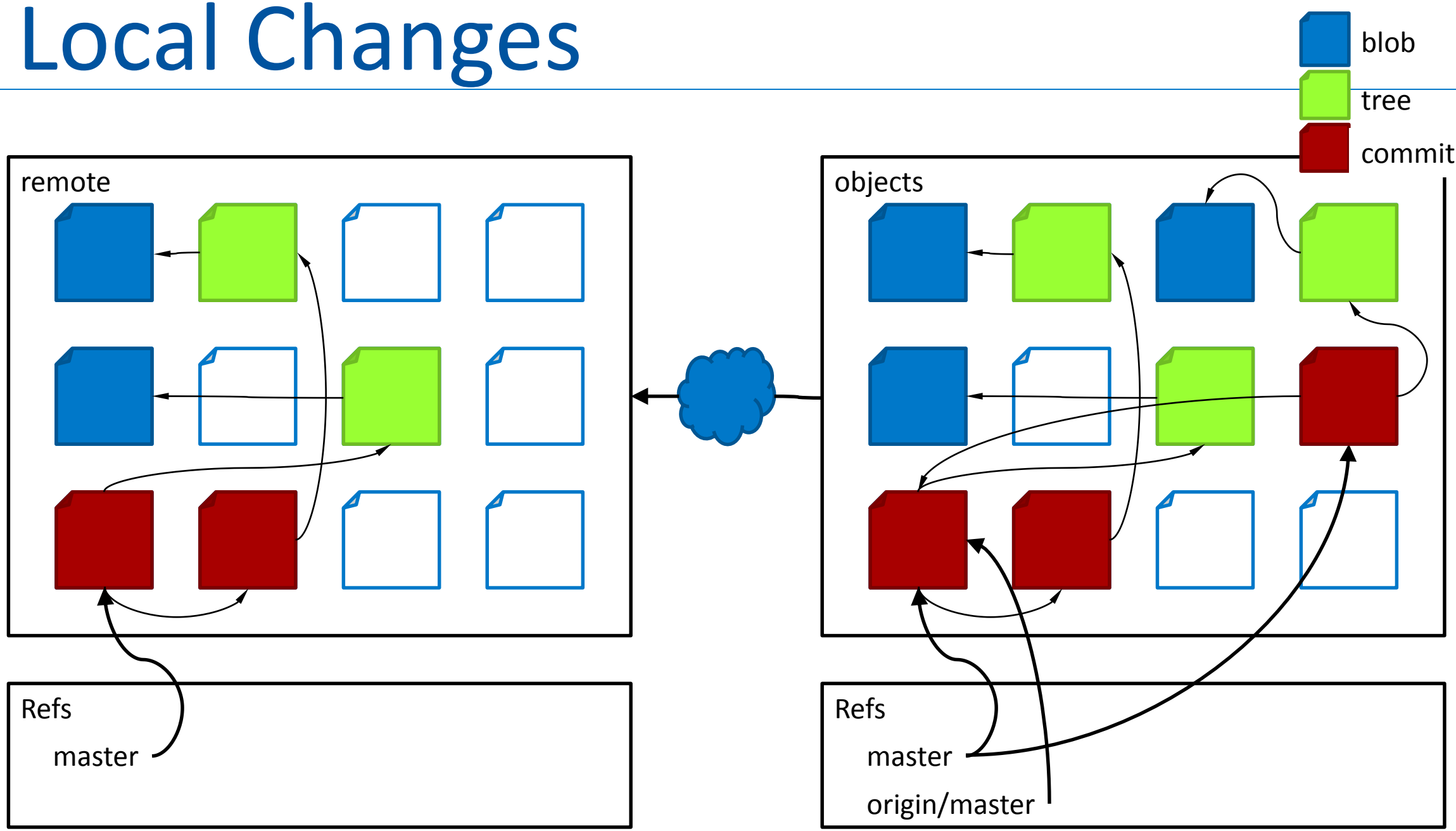
fetch – update tracking branch



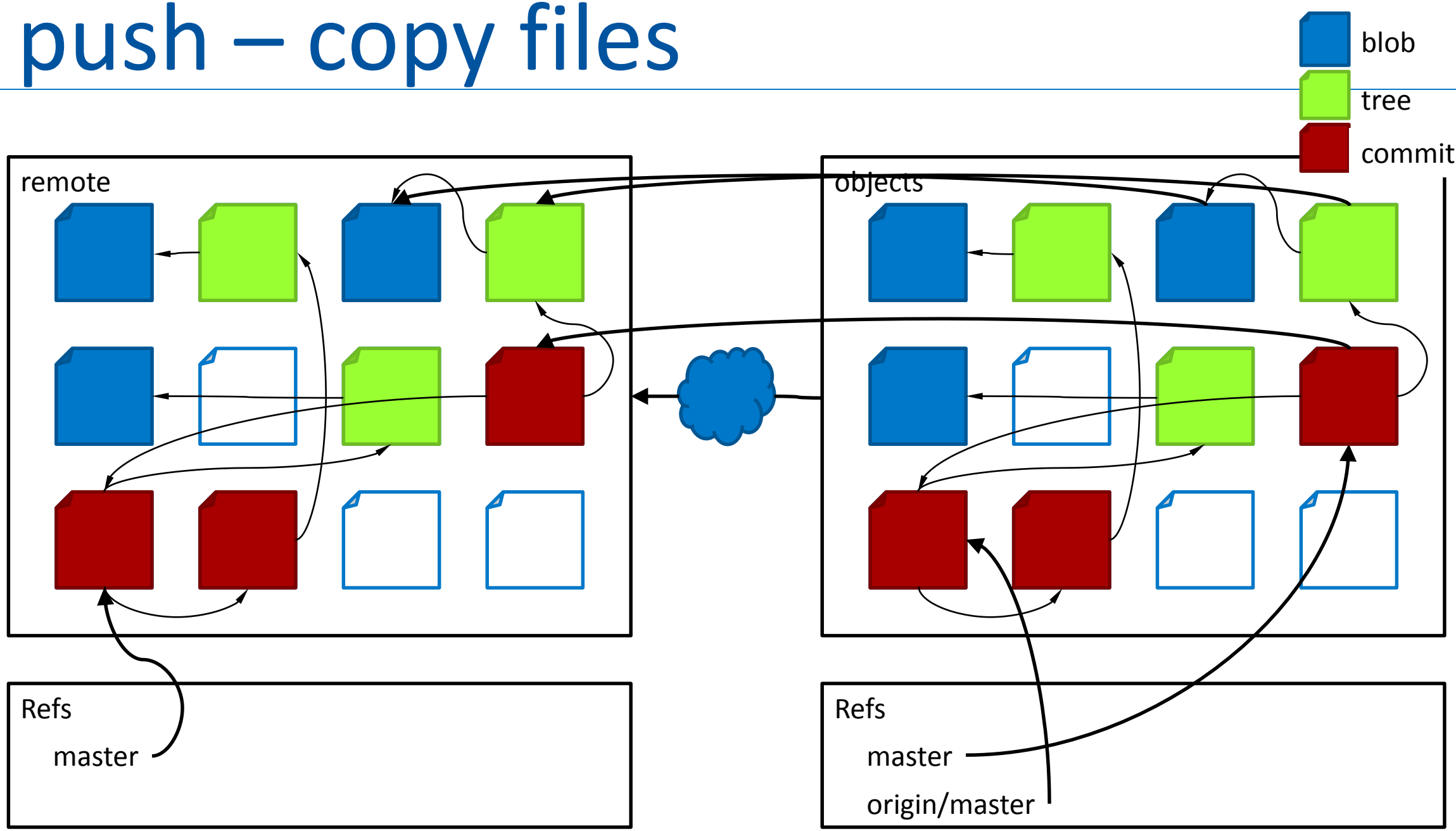
merge – update local branch



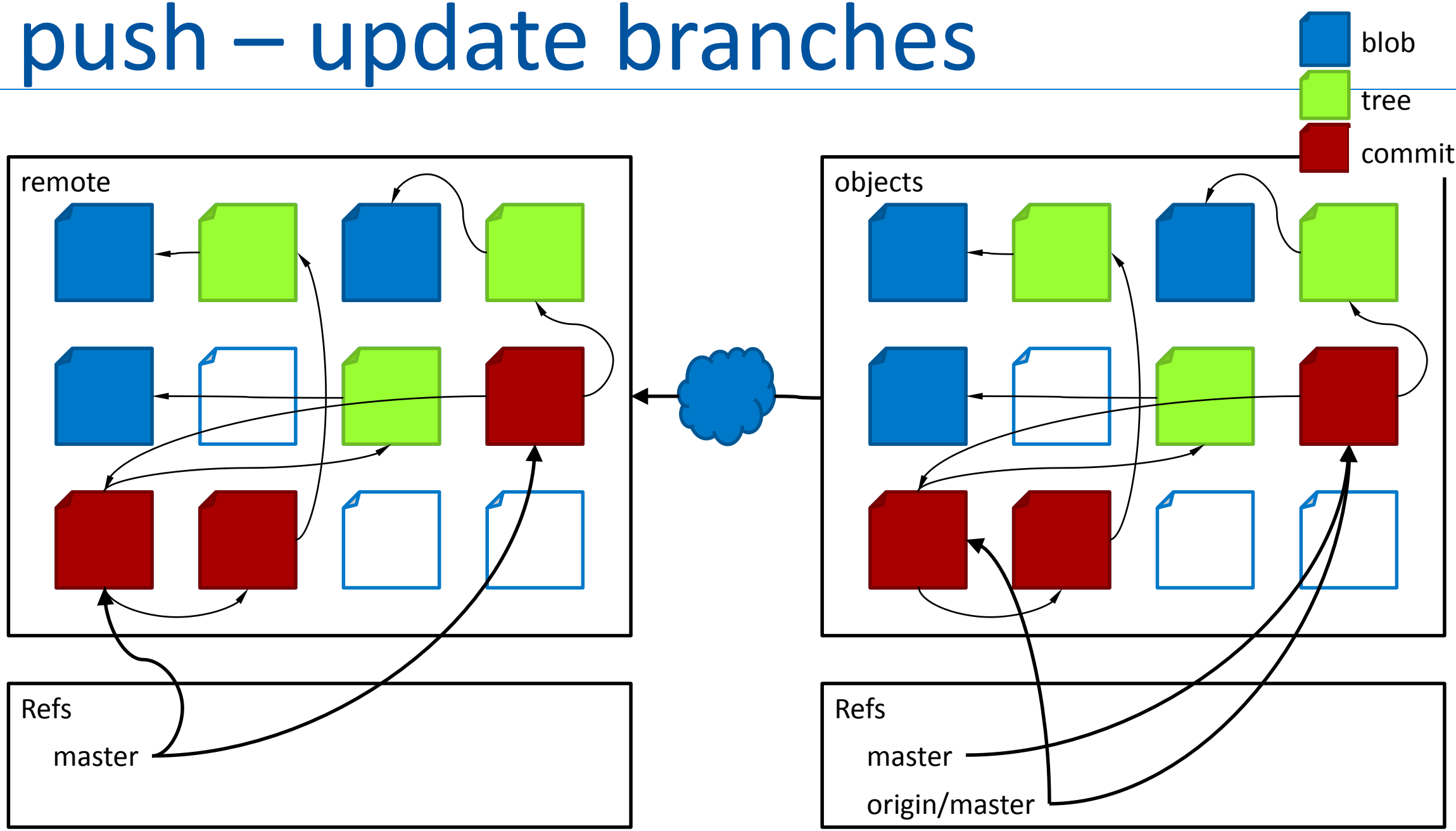
Local Changes



push – copy files

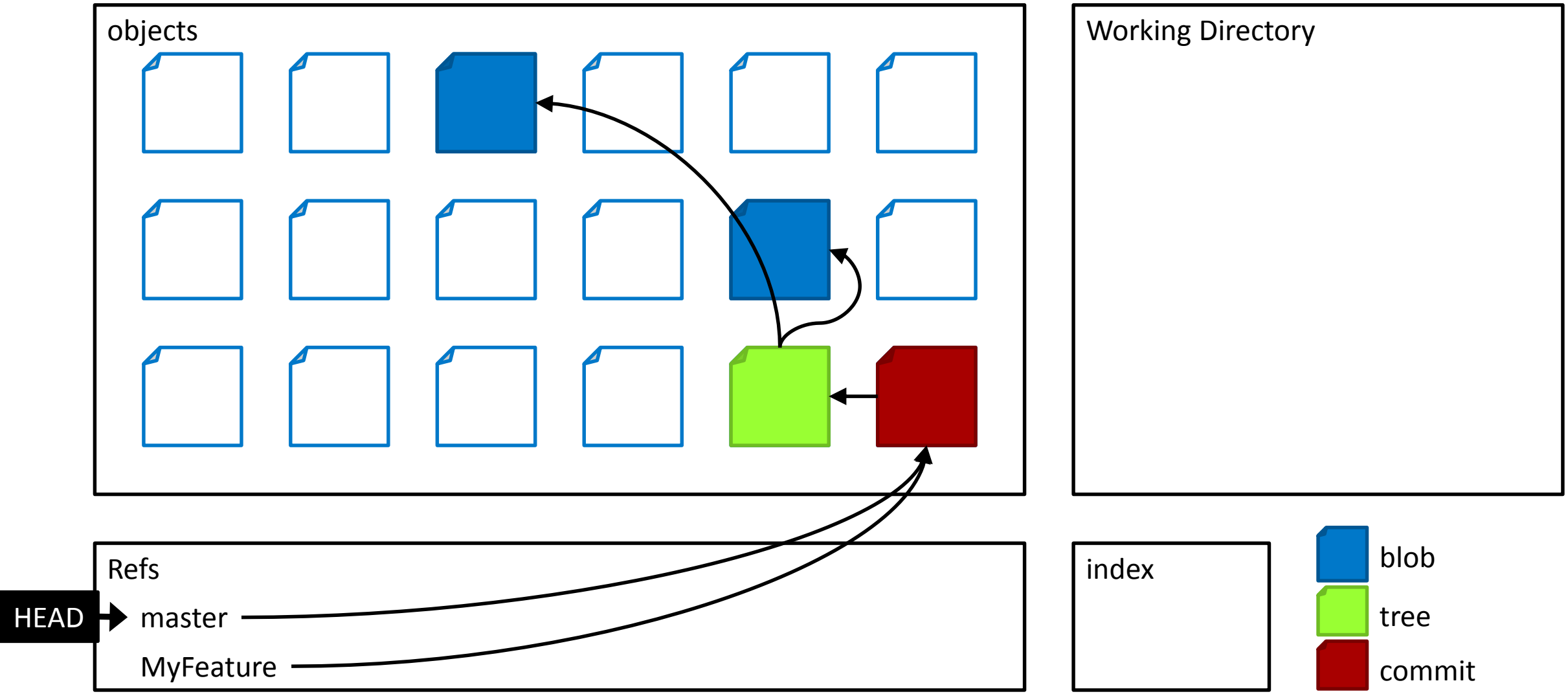


push – update branches

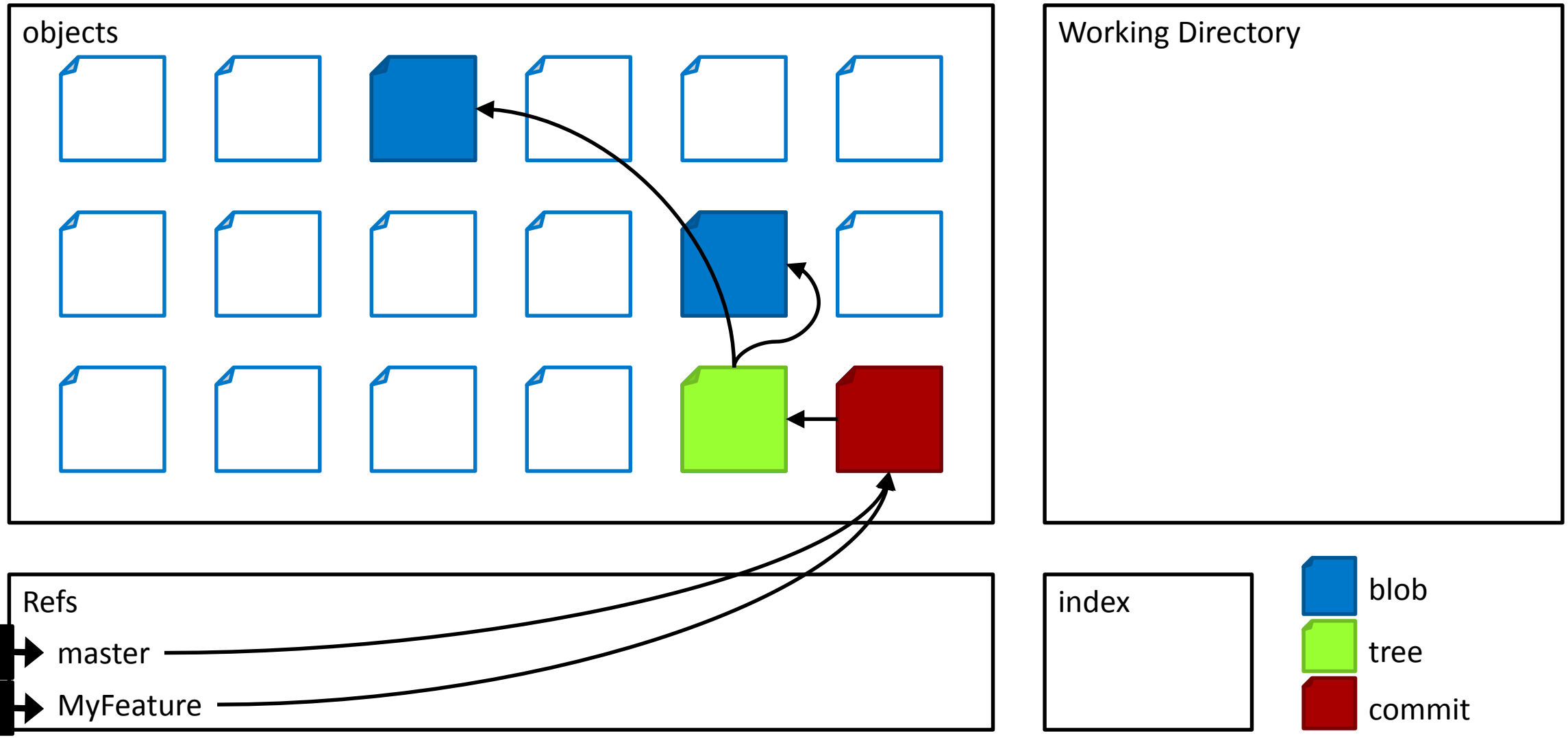


Branching & Merging

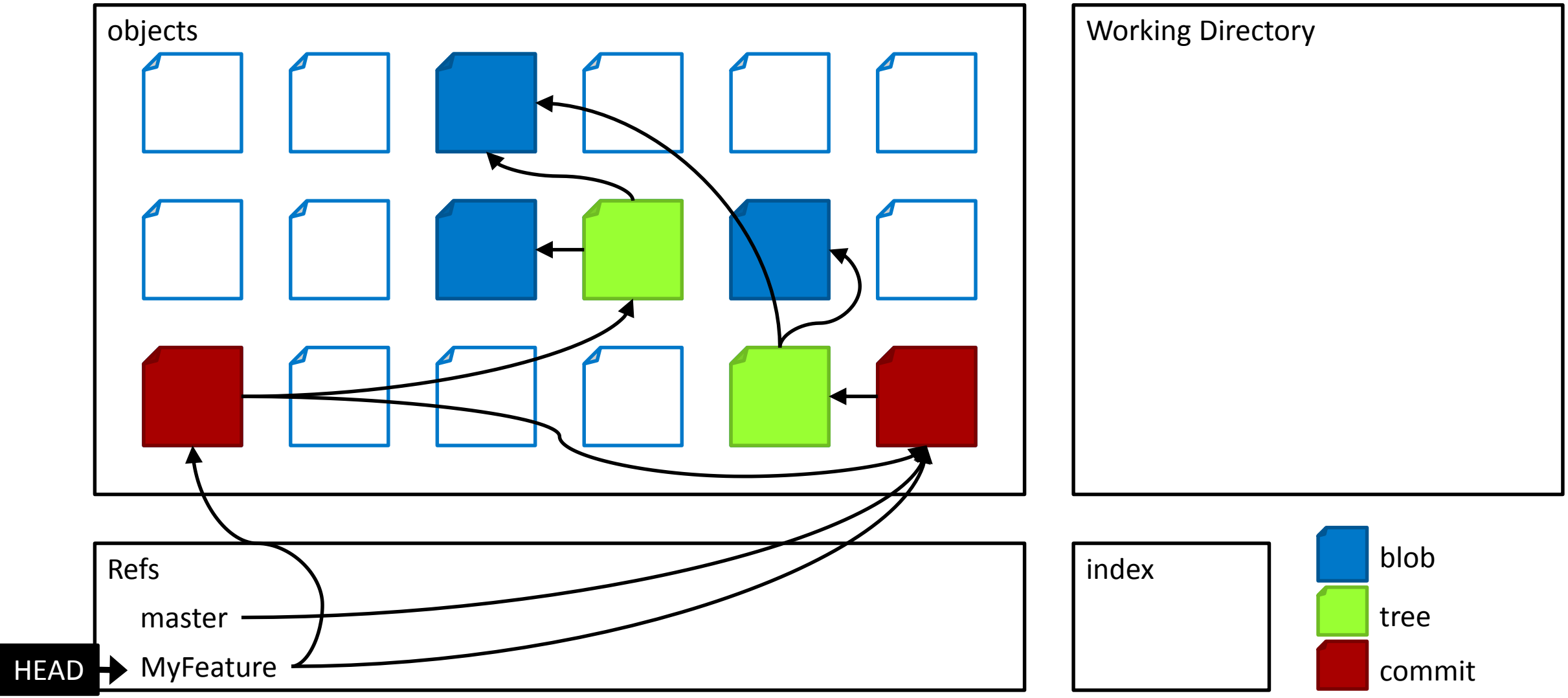
git branch MyFeature



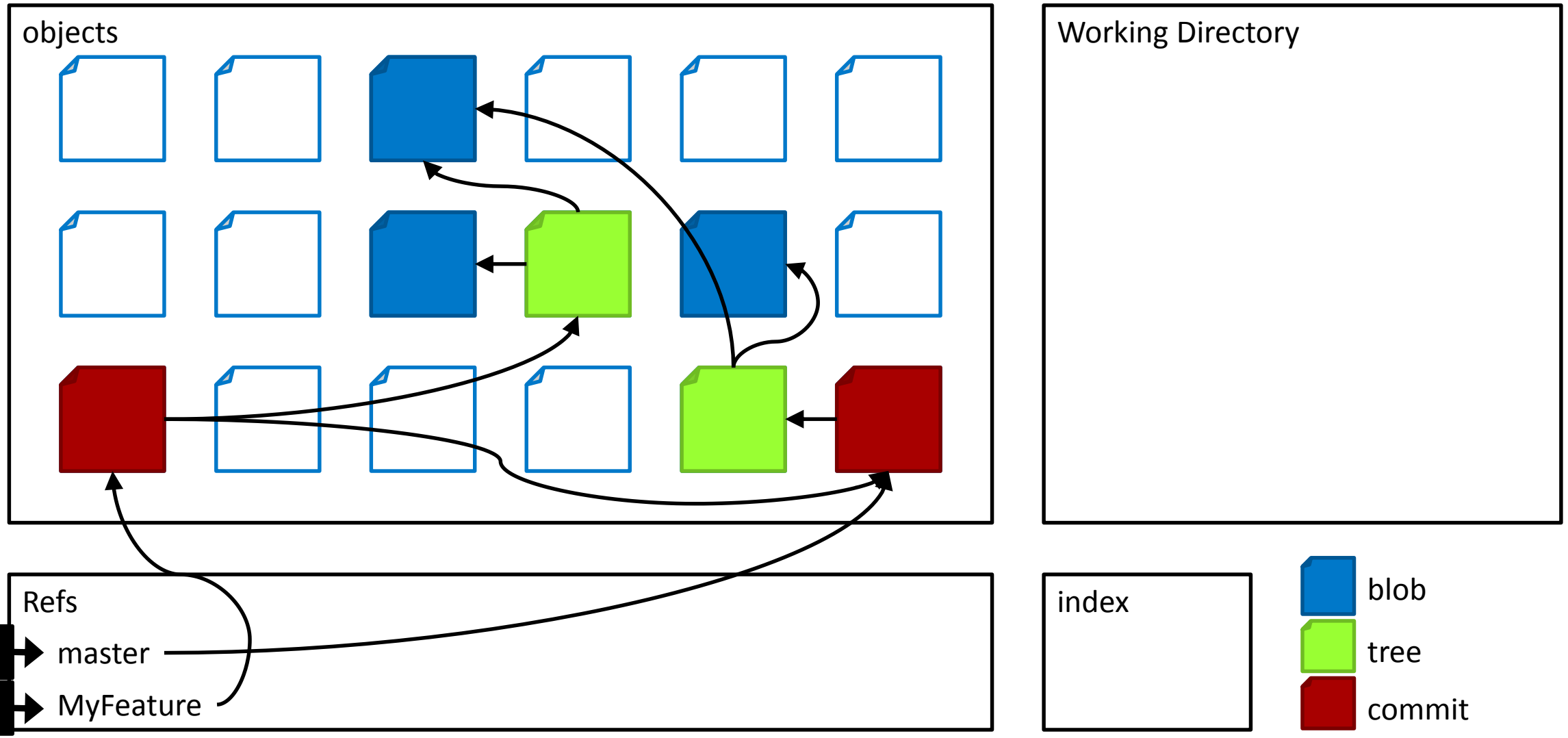
git checkout MyFeature



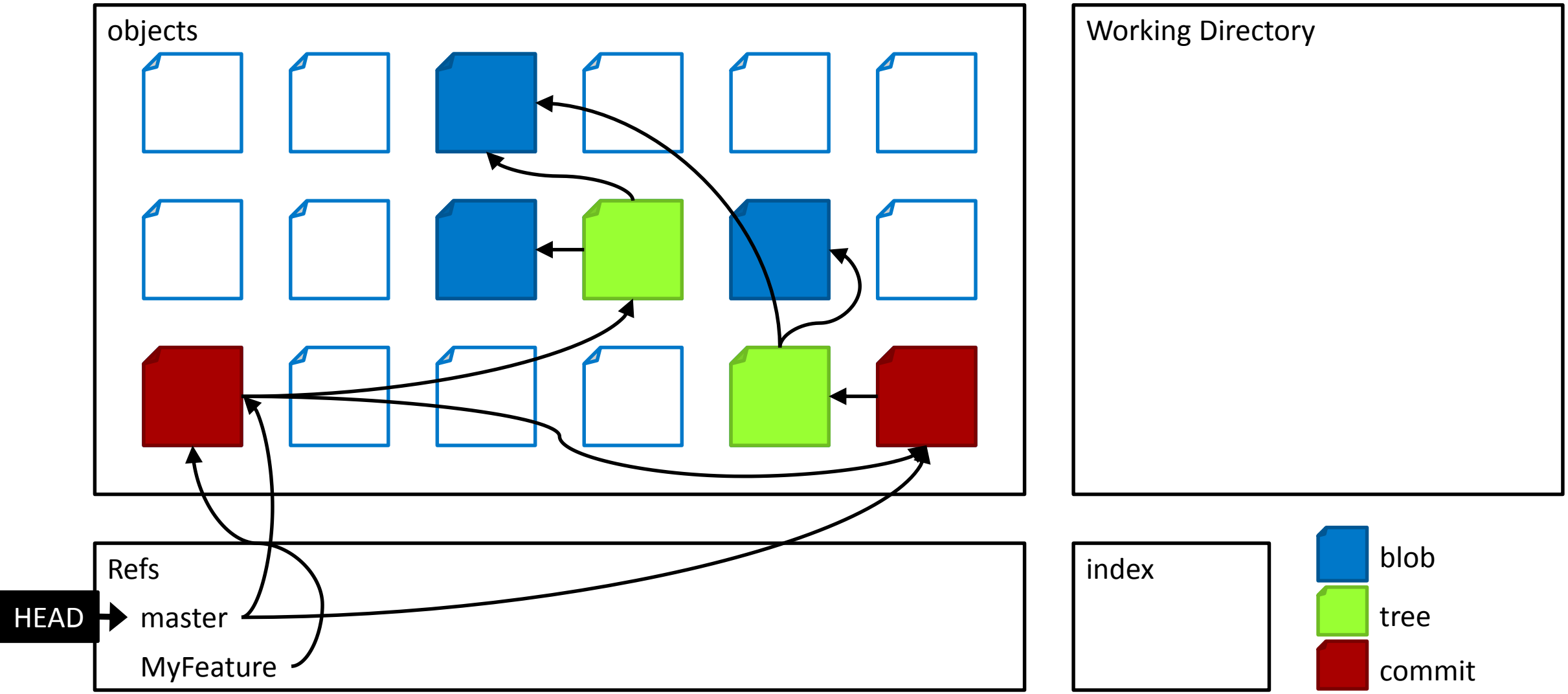
git commit -am ’demo’



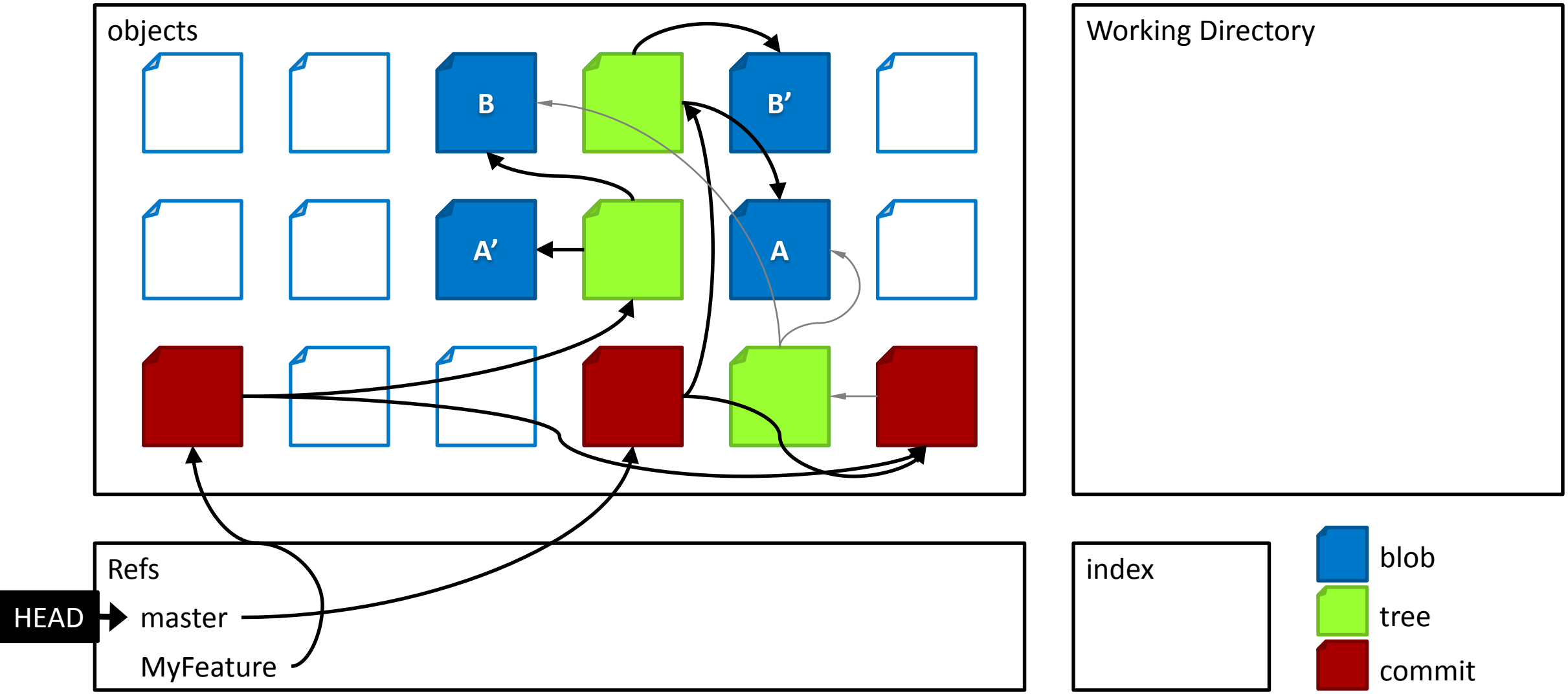
git checkout master



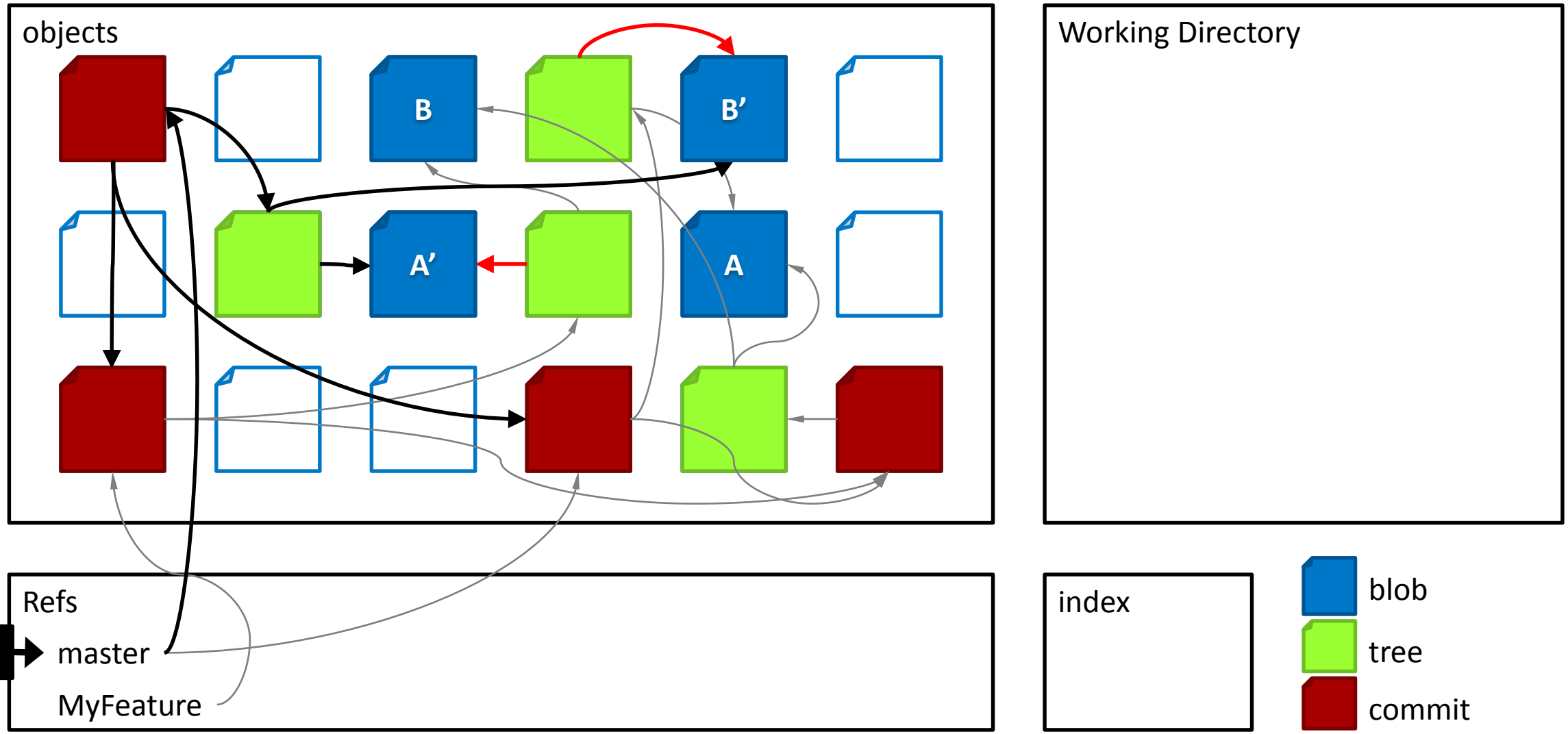
git merge MyFeature – ffwd merge



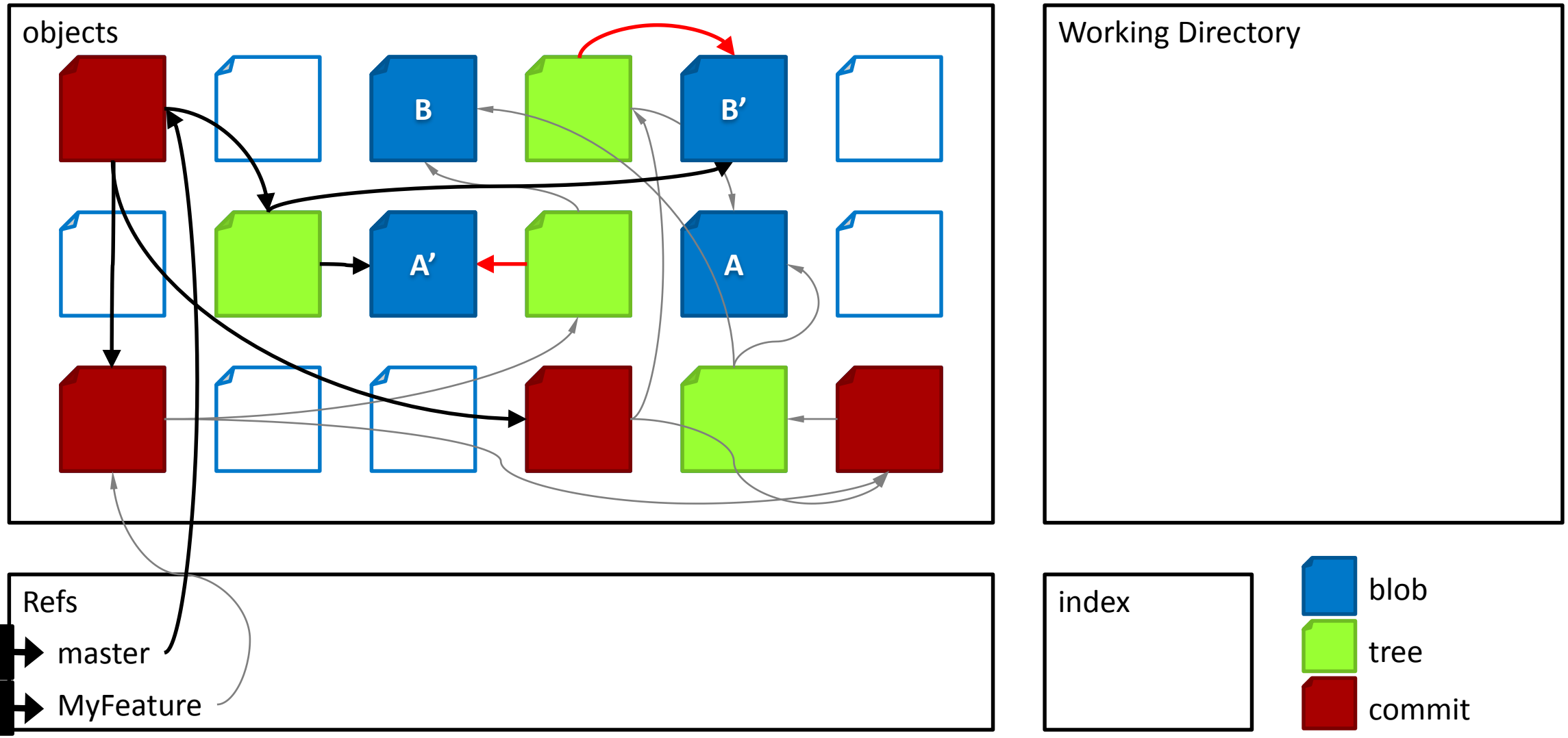
git merge MyFeature – non-ffwd



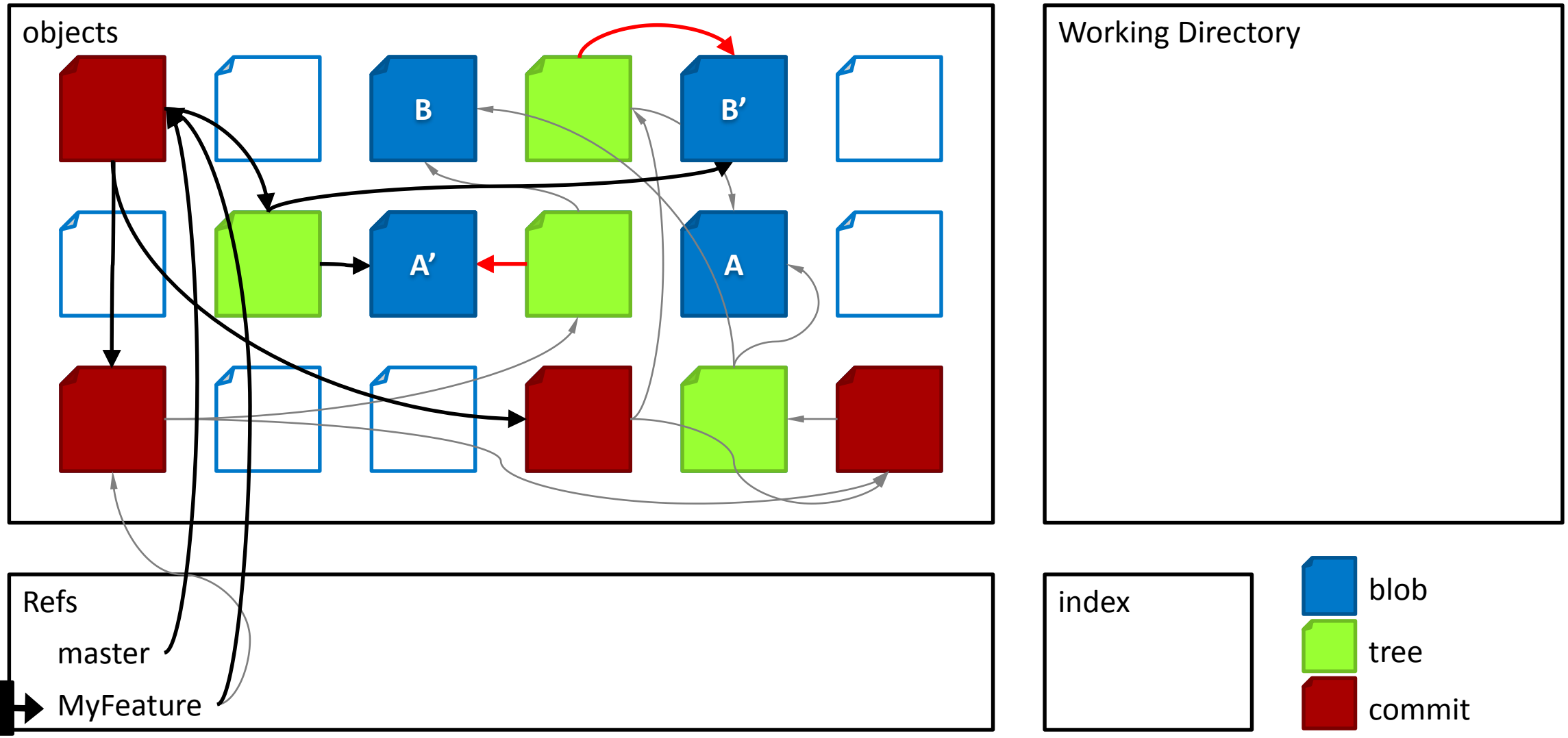
git merge MyFeature – non-ffwd



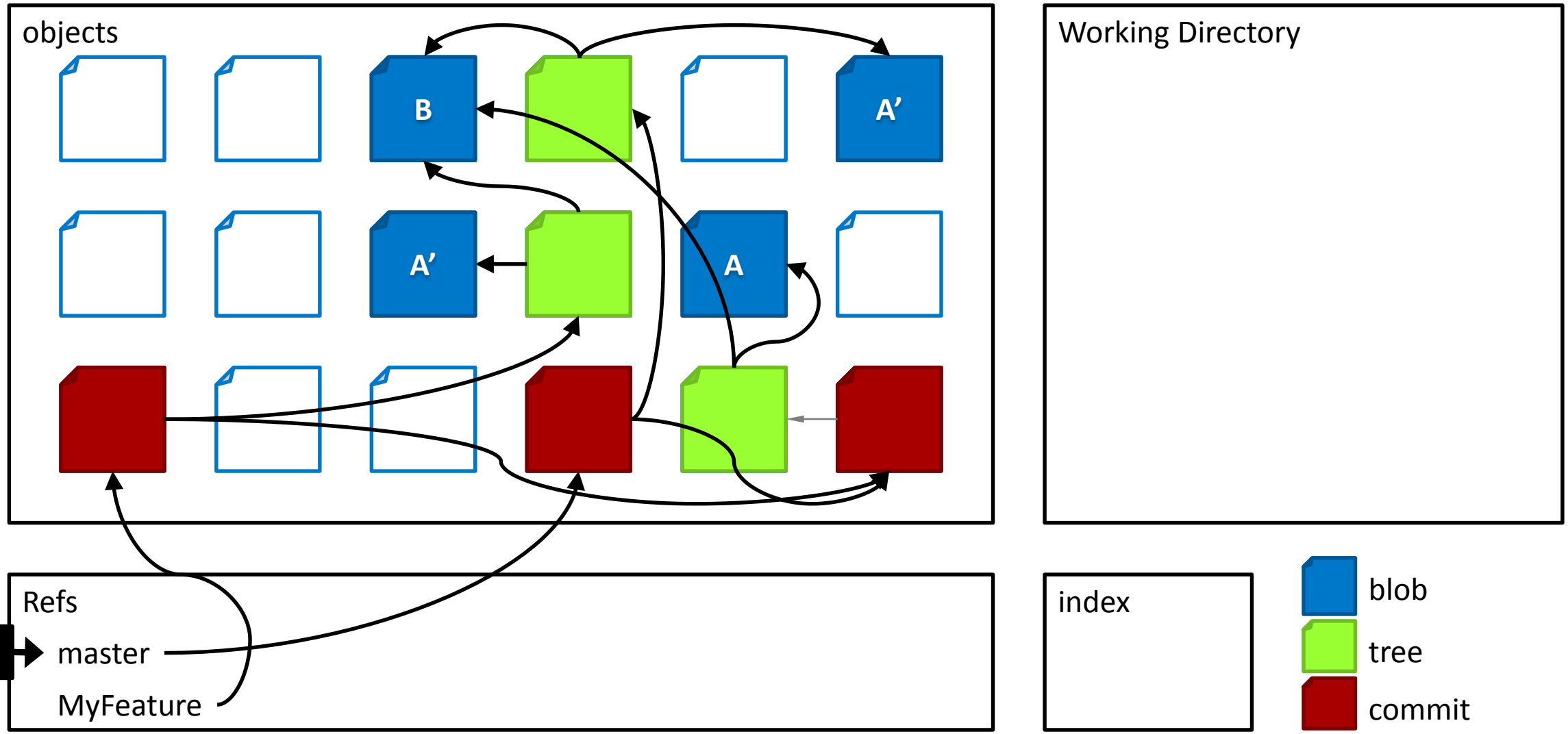
git checkout MyFeature



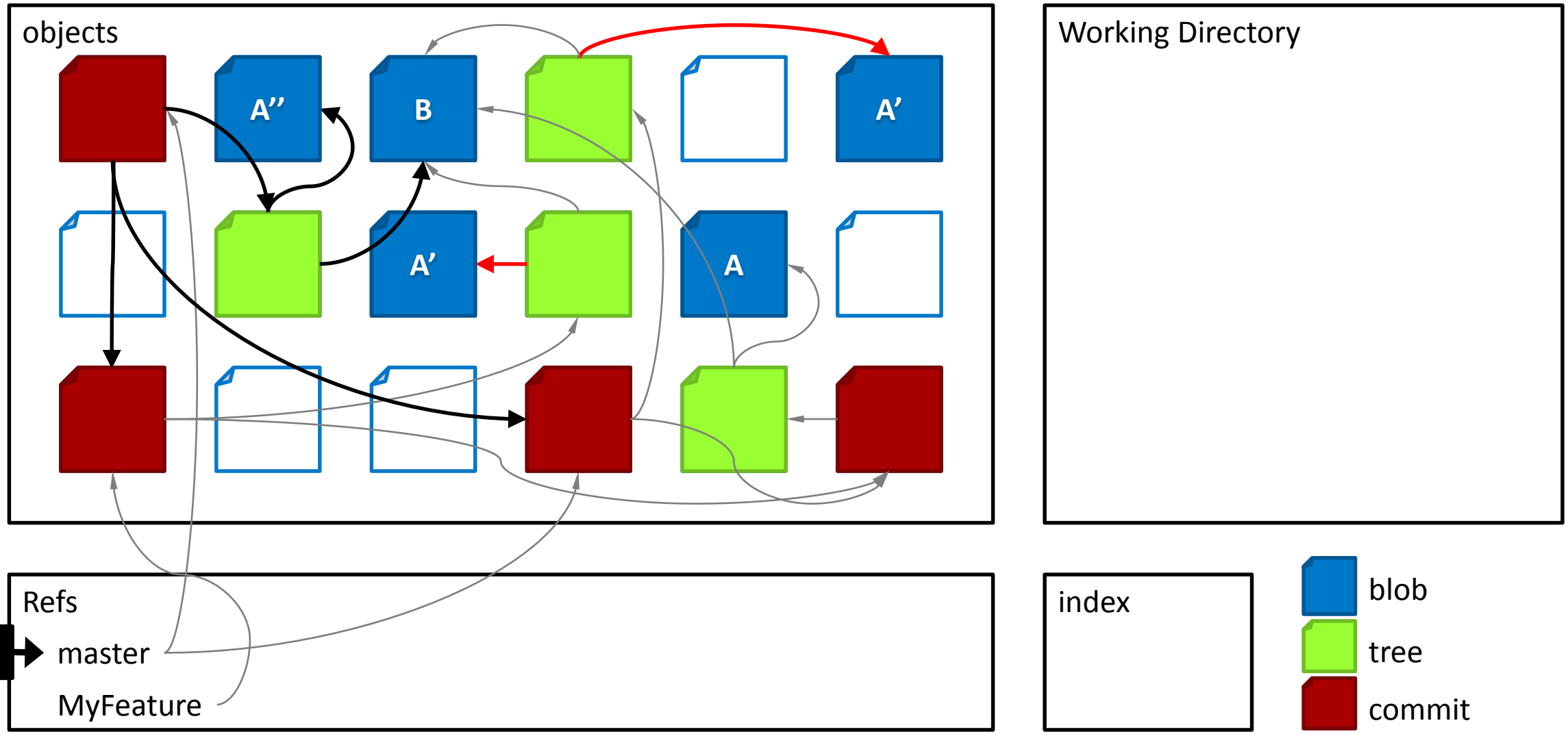
git merge master



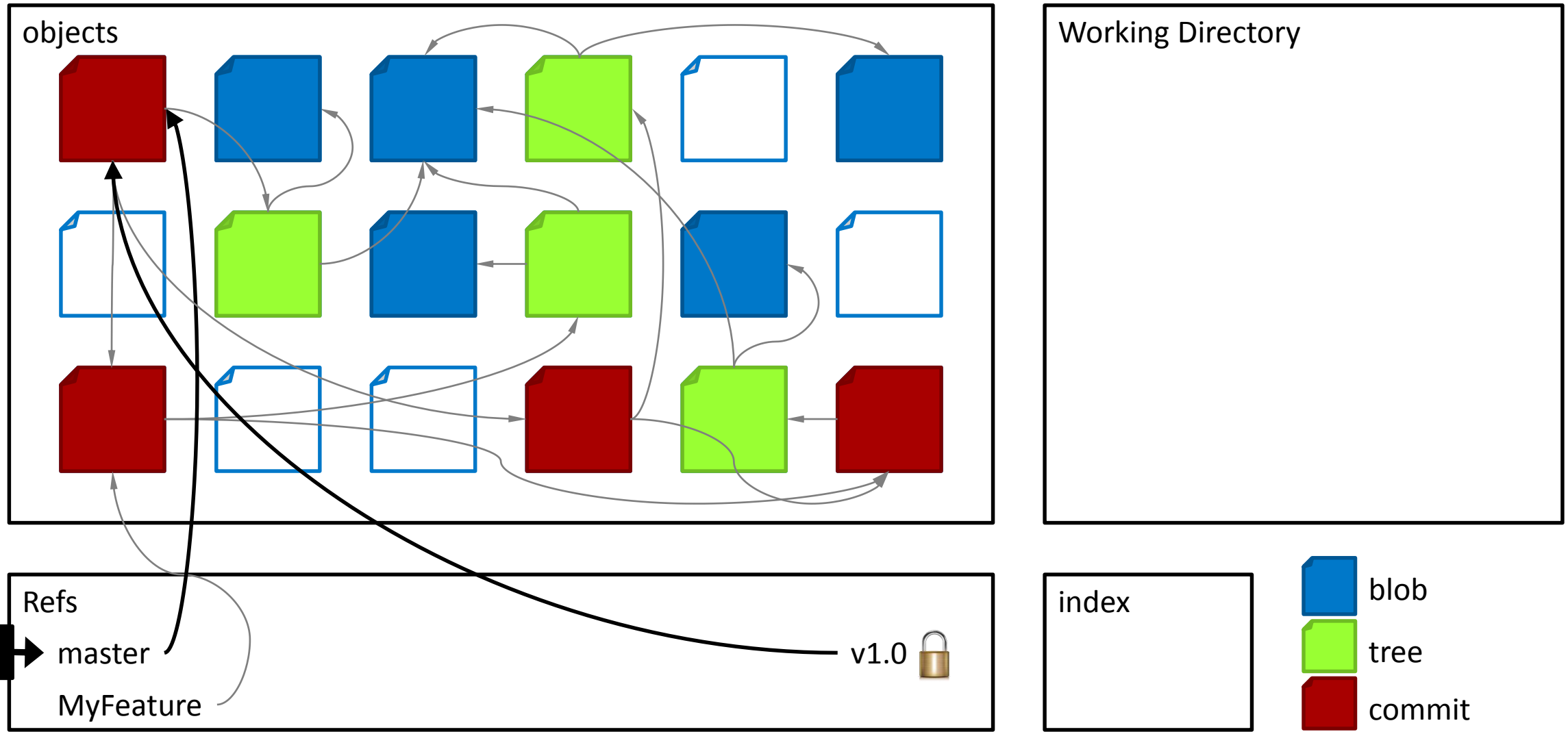
git merge MyFeature – conflict



git merge MyFeature – conflict

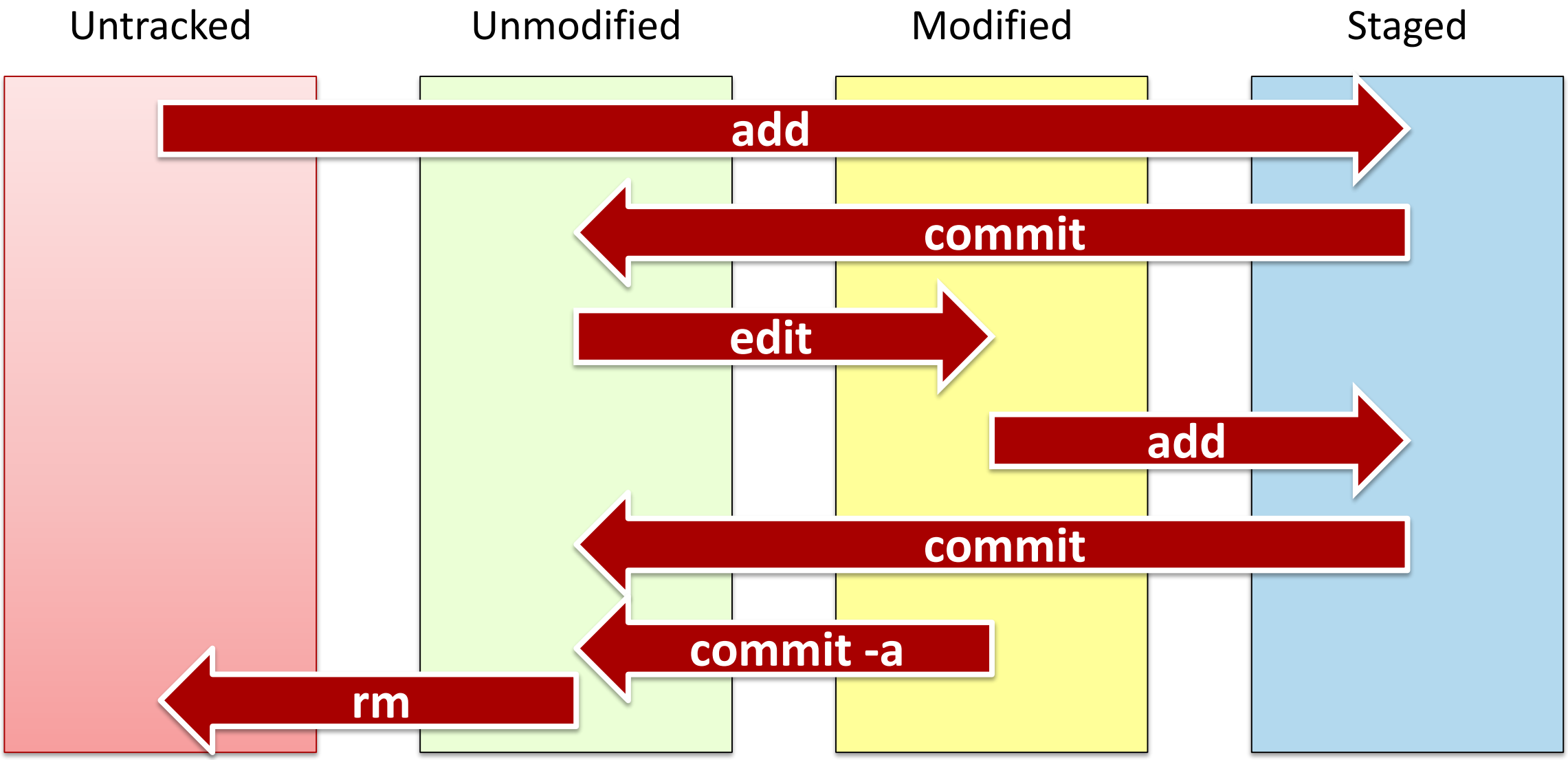


git tag v1.0


objects
Working Directory
Refs
HEAD
master
MyFeature
v1.0
index
blob
tree
commit

Status & Diff

States of Tracking



http://git-scm.com/book/en/Git-Basics-Recording-Changes-to-the-Repository

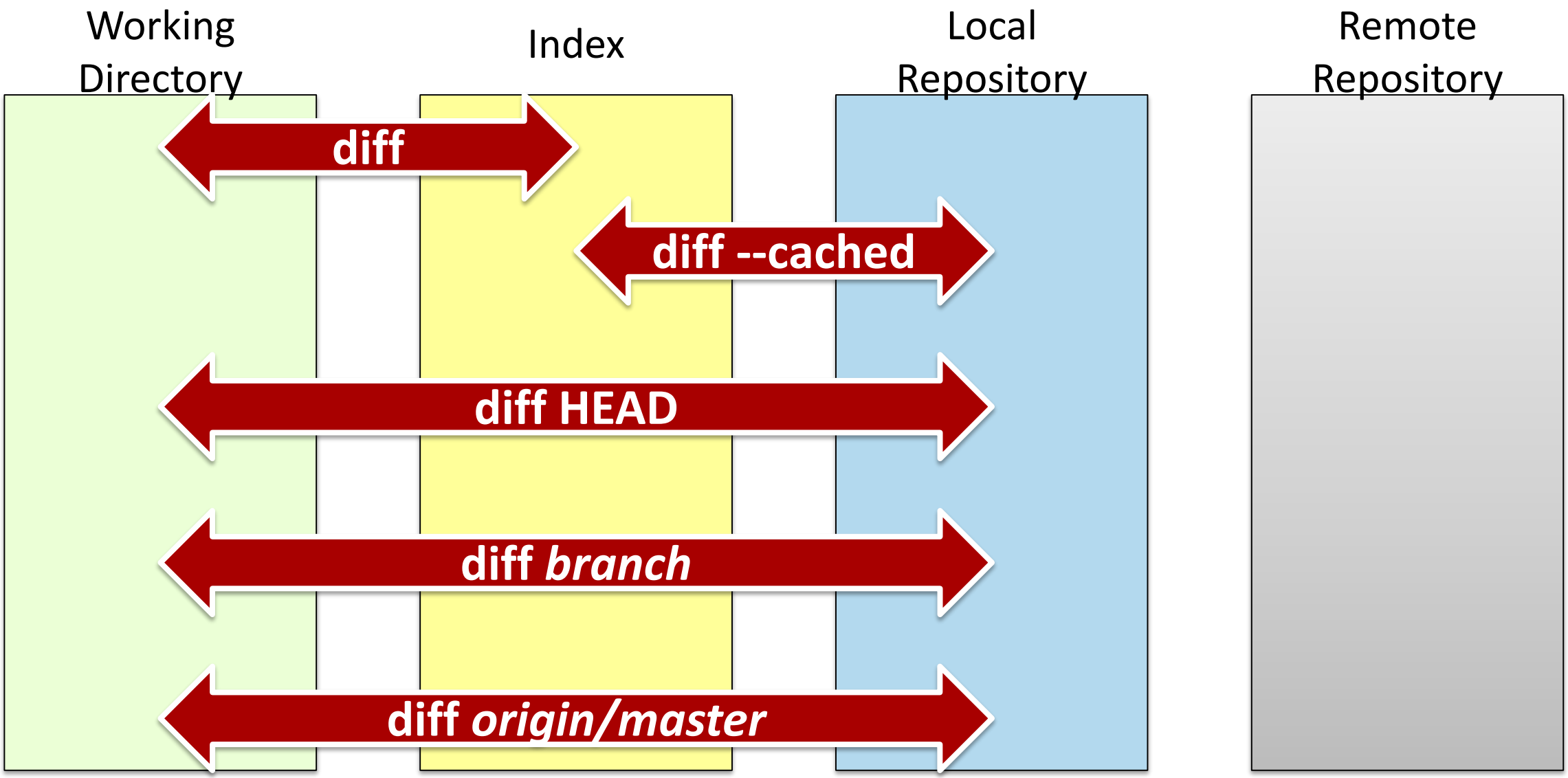
git status

- Show the status of the working tree
  1. Differences between **staging area** and **current branch**
  2. Differences between **working directory** and **staging area**
  3. Files that are **untracked** in the working directory

git diff

- git diff [*SomeFile*]
  - Shows differences between **working copy** and **staging area**
- git diff --cached [*SomeFile*]
  - Shows differences between **repository** and **staging area**
- git diff HEAD [*SomeFile*]
  - Shows differences between **working copy** and **repository**

git diff


Working Directory
Index
Local Repository
Remote Repository
diff
diff --cached
diff HEAD
diff branch
diff origin/master

git log

- Shows history of current branch
  - Use --oneline for more readable output
  - Use --decorate to output the ref names
  - Use --graph for a text based graph

git reset

- git reset <*filename*>
  - Resets **index** making changes in working directory **unstaged**
- git reset --hard
  - Resets **index** to current ref and working directory

git checkout

- git checkout <filename>
  - Discards changes in **working directory** (resets to **index**)

git rm

- git rm <filename>
  - Removes file from index **and** from the working directory
- git rm --cached <filename>
  - Removes file from **index only** (making it **untracked**)

git clean

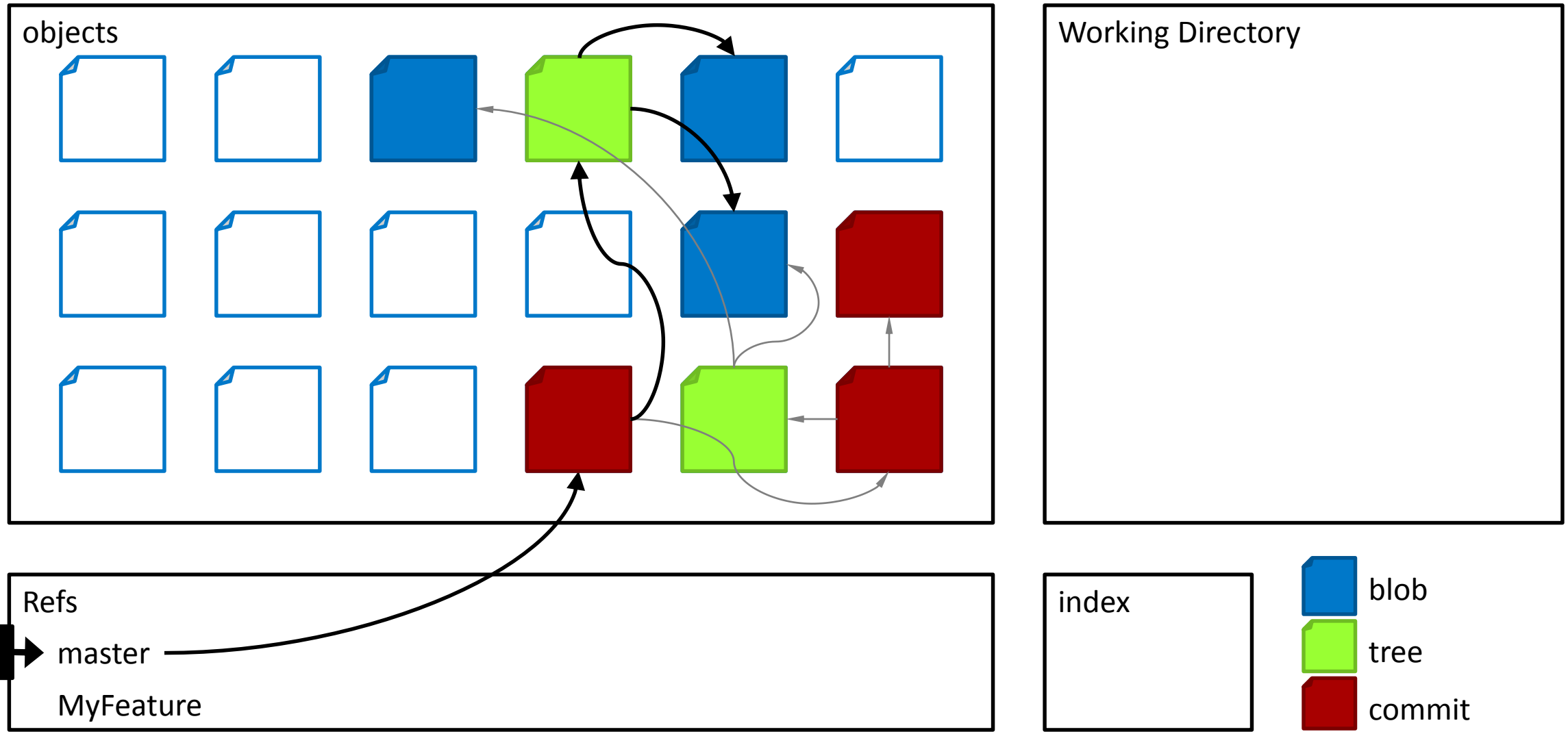
- git clean -f [-d -x]
  - Cleans the working tree by recursively removing files that are not under version control, starting from the current directory.
  - Normally, only files unknown to git are removed, but if the *-x* option is specified, ignored files are also removed.

Rewriting History

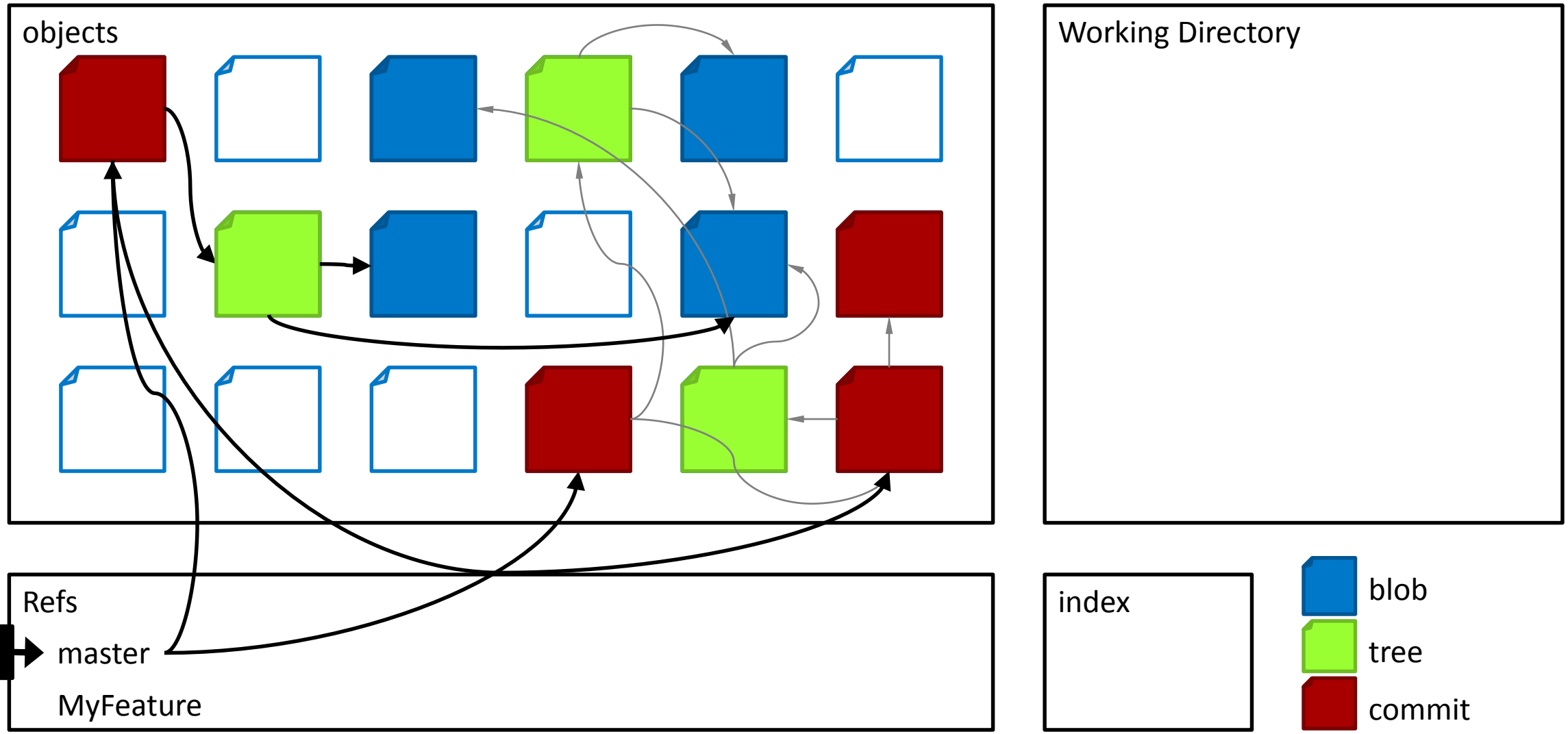
Rewriting History

Use it **only** to rewrite your **own** and **local** history!

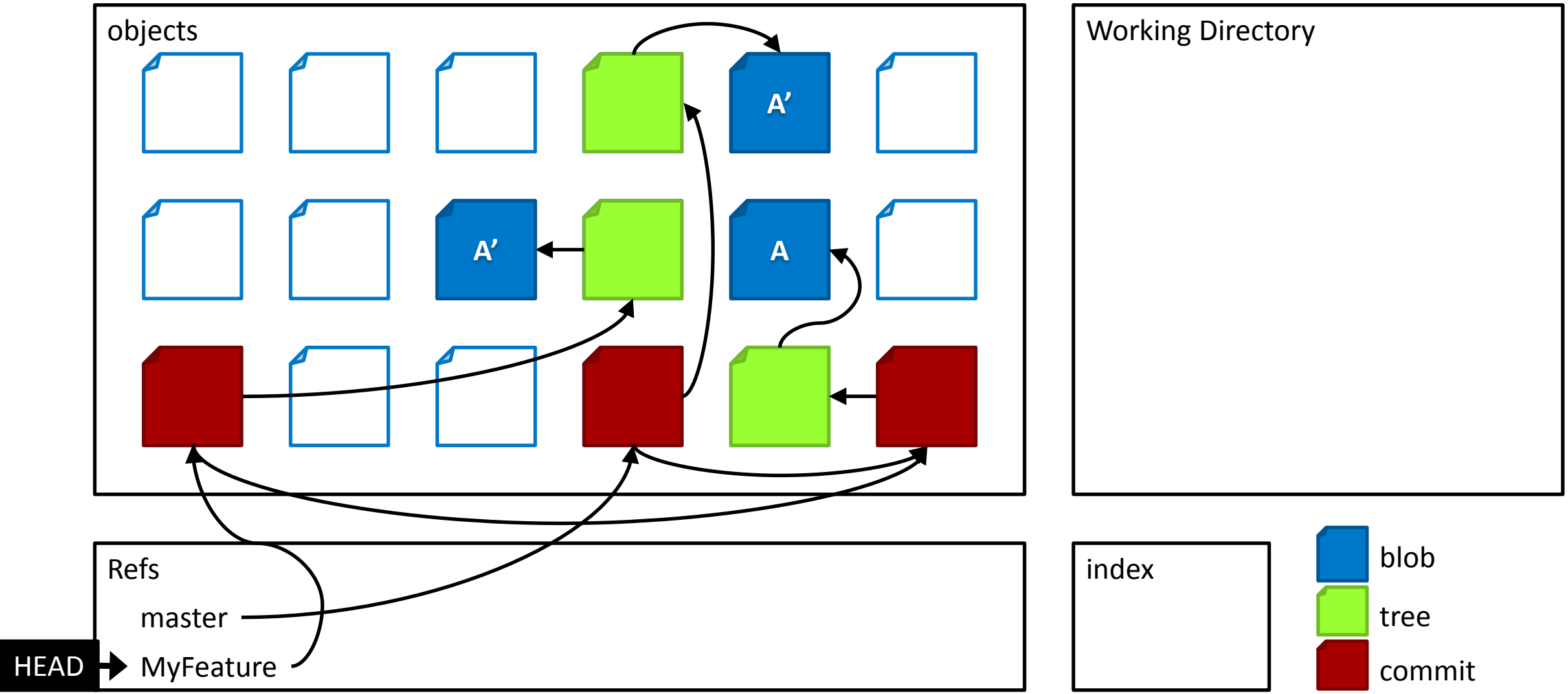
git commit --amend



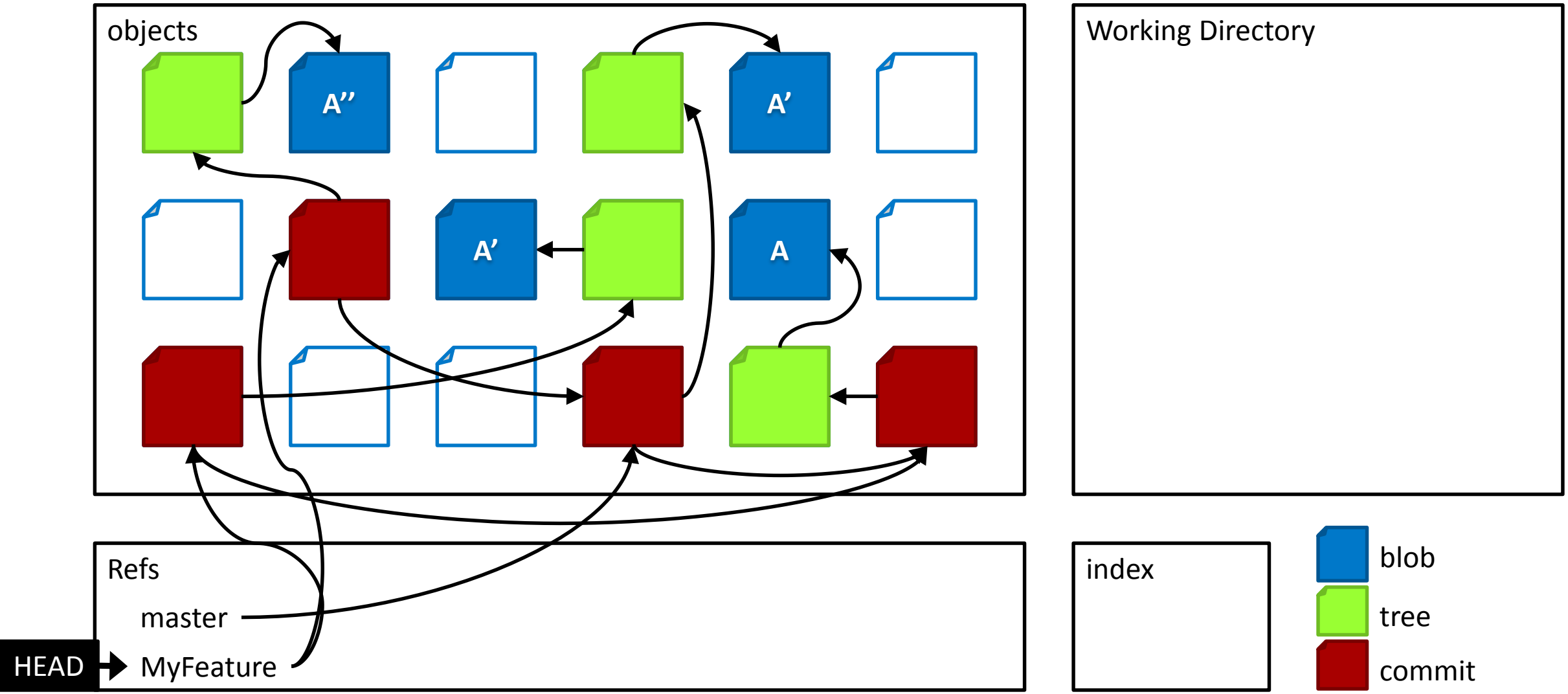
git commit --amend



git rebase master



git rebase master



Rewriting History

Use it **only** to rewrite your **own** and **local** history!

Rebase Interactive

git rebase -i <after-this-commit>

- Re-order commits
- Edit commits
- Squash commits
- Drop commits

git cherry-pick

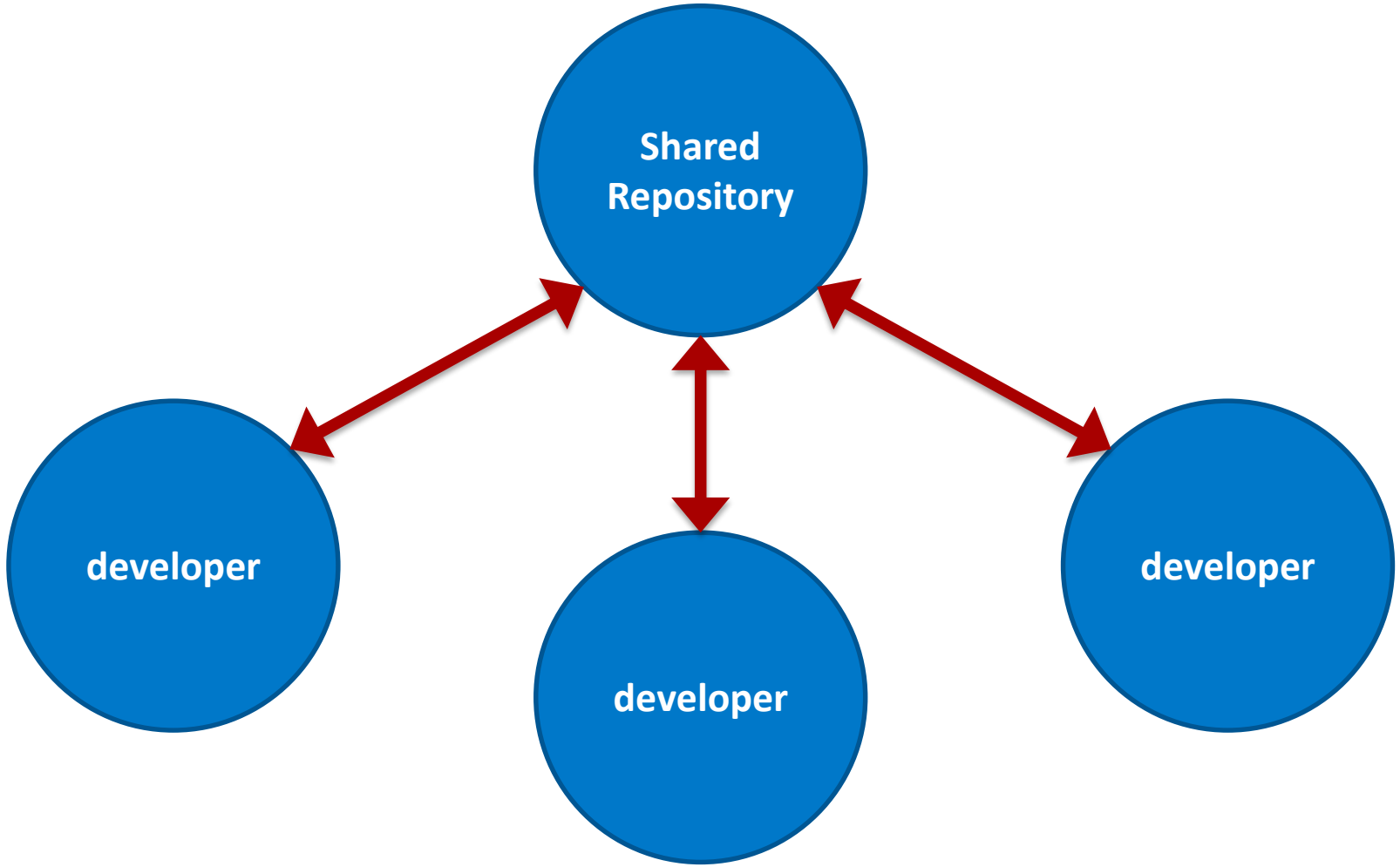
- Given one or more existing commits, apply the **change** each one introduces, recording a **new commit** for each. This requires your working tree to be clean (no modifications from the HEAD commit).

Rewriting History

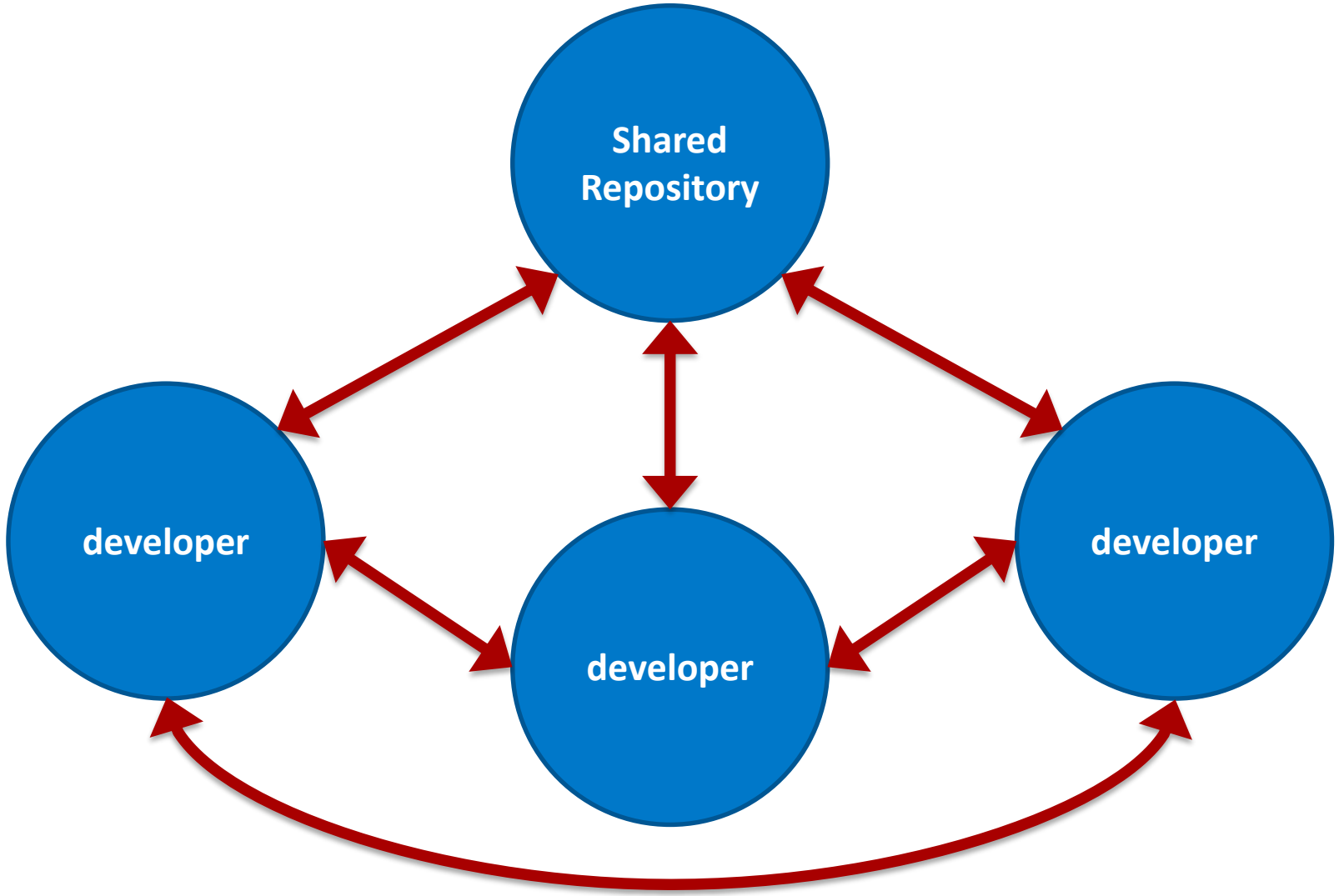
Use it **only** to rewrite your **own** and **local** history!

WORKFLOW

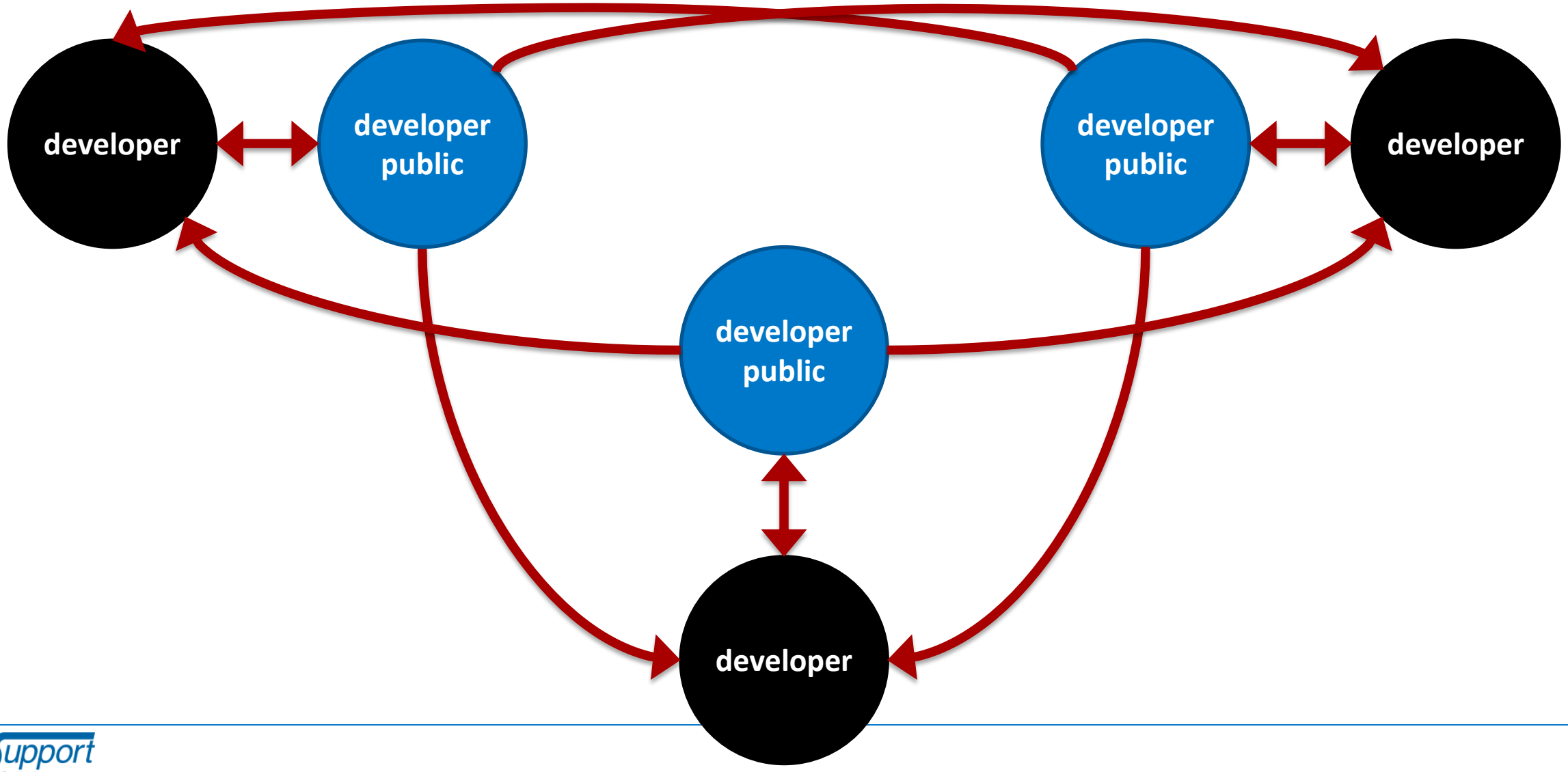
Centralized Workflow



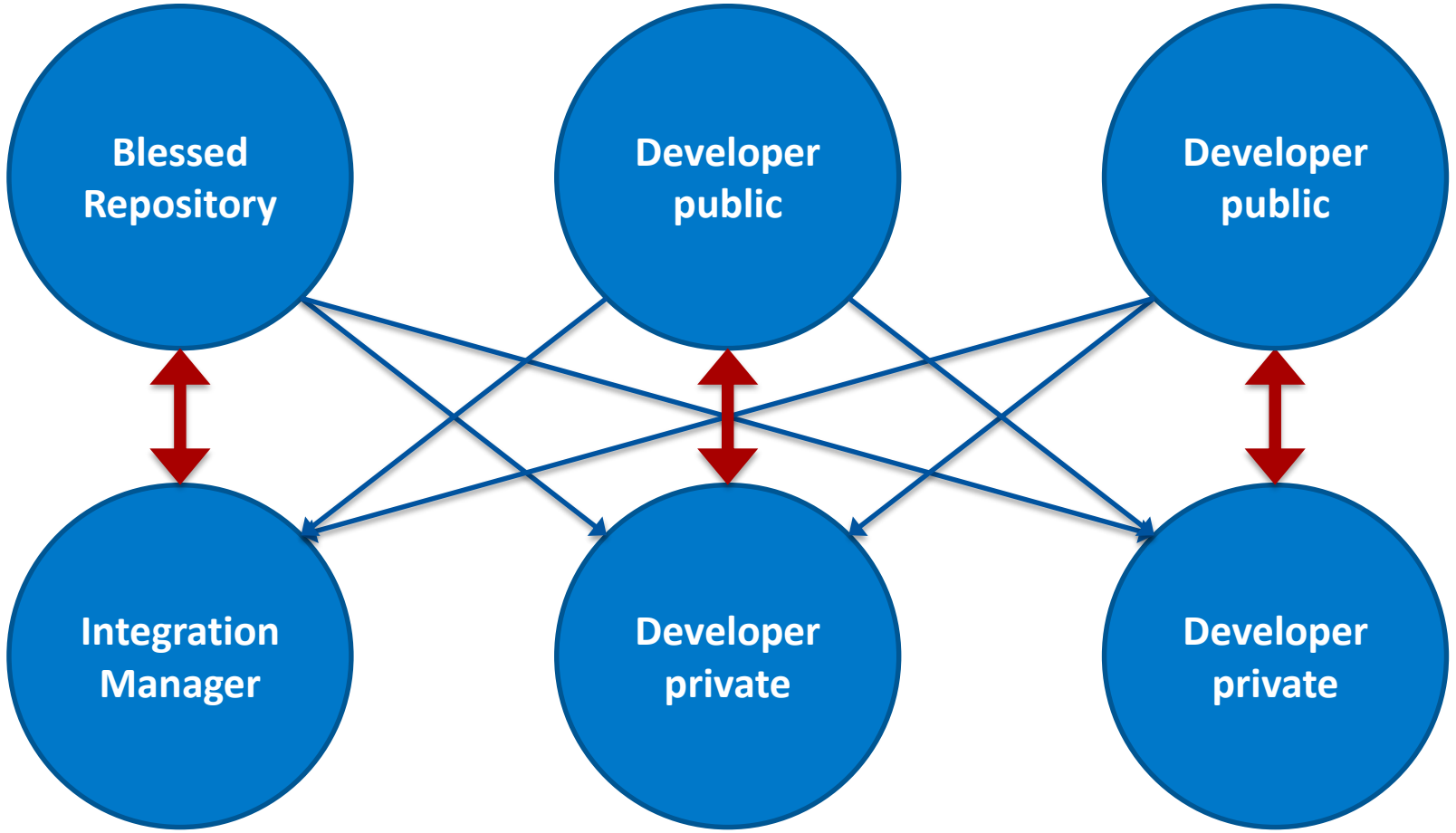
Centralized Workflow



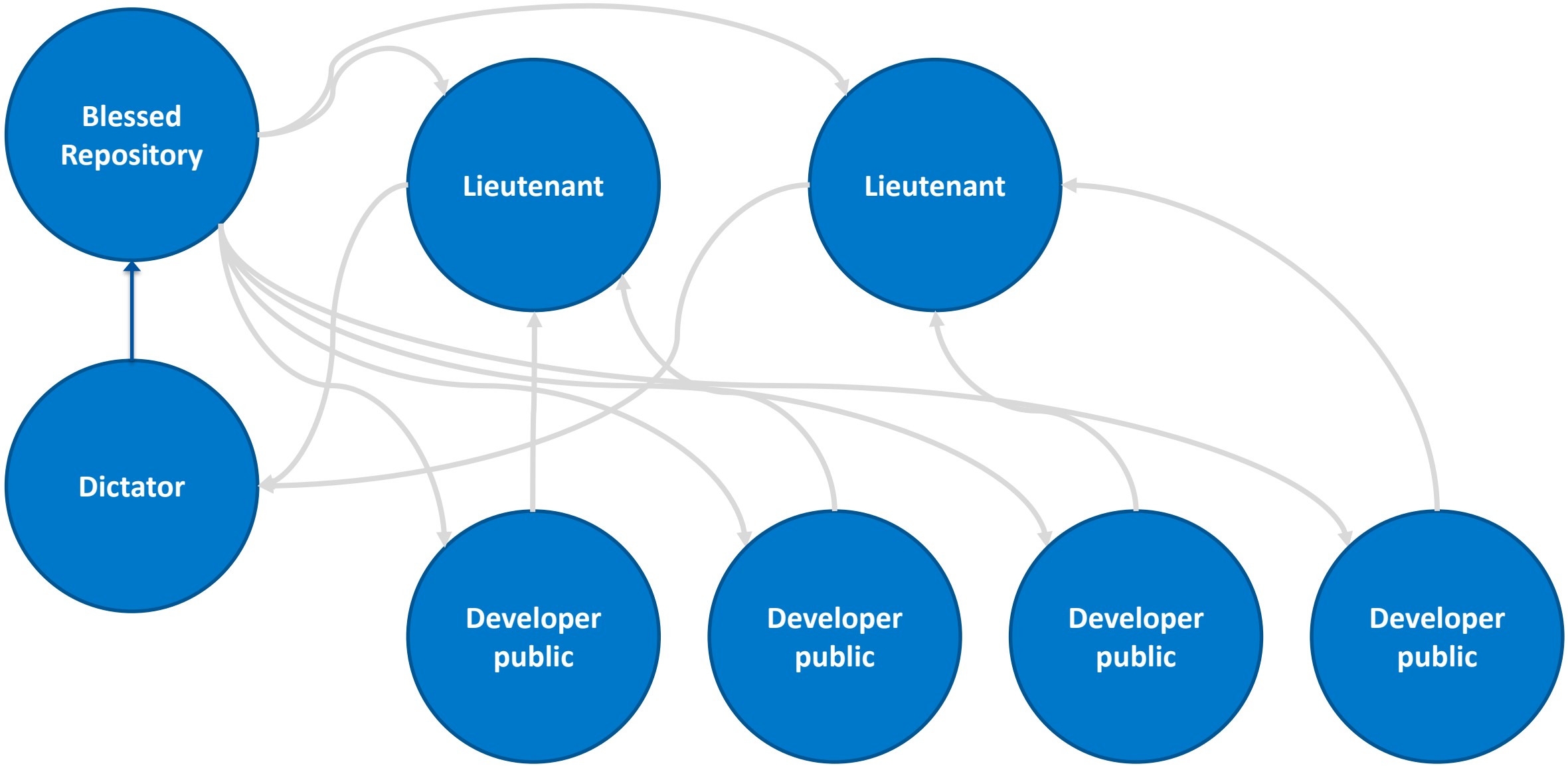
Distributed Workflow



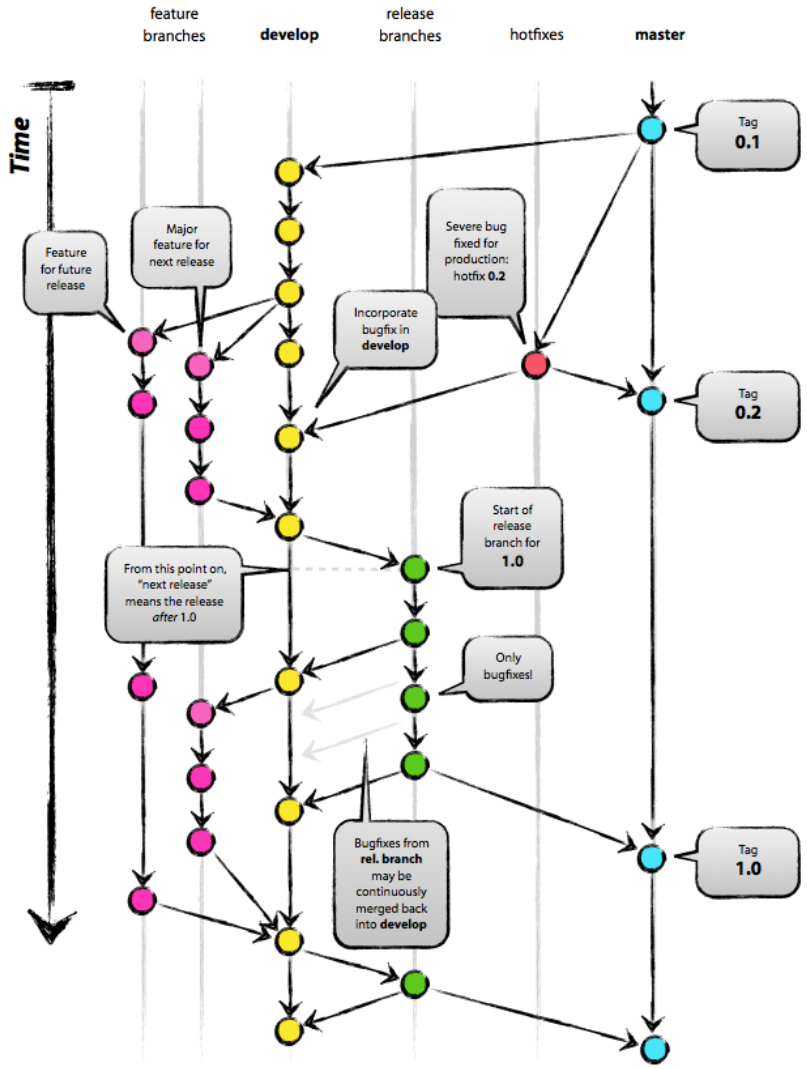
Integration Manager



Dictator and Lieutenants



GitFlow



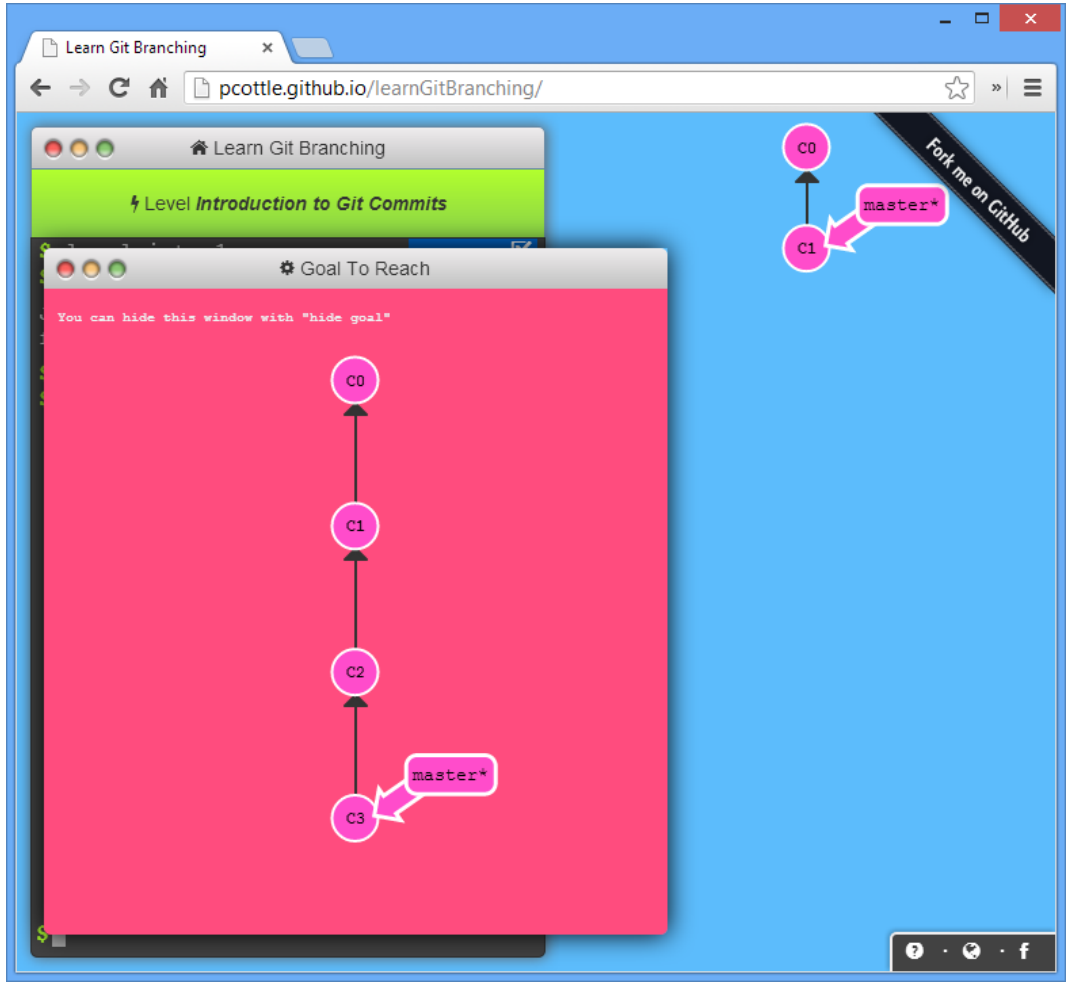
Original blog post:
http://nvie.com/posts/a-successful-git-branching-model/

Excellent description:
http://danielkummer.github.io/git-flow-cheatsheet/

BONUS

Exercise

- http://pcottle.github.io/learnGitBranching/



Must Watch

- Linus Torvalds on Git
  - http://www.youtube.com/watch?v=4XpnKHJAok8

- Introduction to Git by Scott Chacon
  - http://www.youtube.com/watch?v=ZDR433b0HJY

Also available on www.yovisto.com

Must Read

- The Thing About Git
  - http://tomayko.com/writings/the-thing-about-git

- Git from the bottom up
  - http://ftp.newartisans.com/pub/git.from.bottom.up.pdf