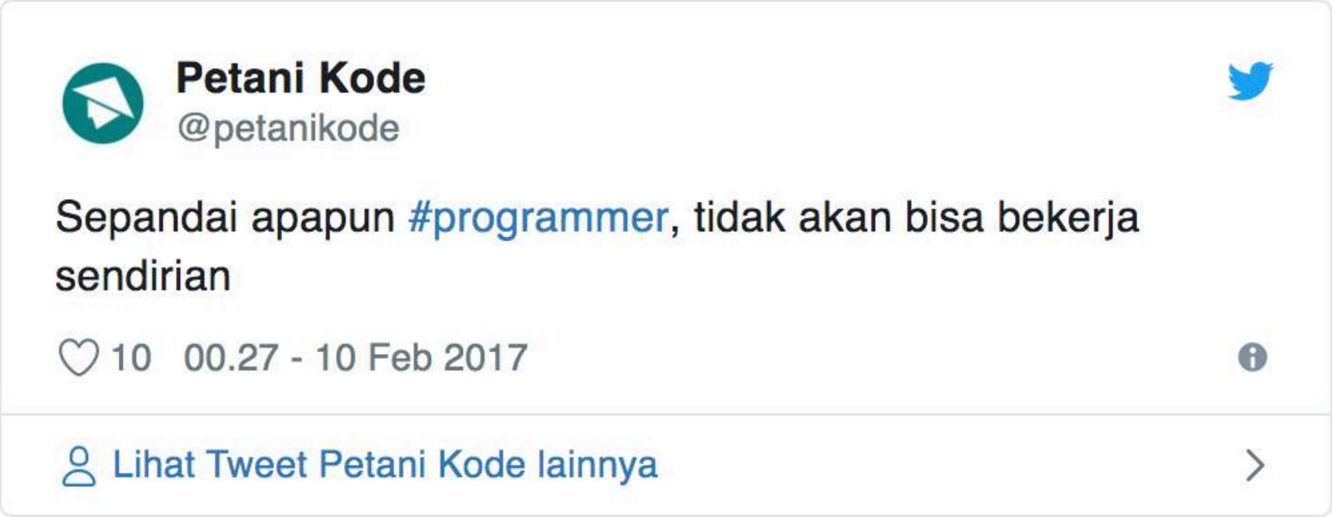
Pengenalan GIT



Petani Kode
@petanikode

Sepandai apapun #programmer, tidak akan bisa bekerja sendirian

♡ 10 00.27 - 10 Feb 2017

Lihat Tweet Petani Kode lainnya

Target

- Version Control System (VCS) and Git **beginners**
- Developers using Git commands **without knowing what's actually happening**

Goal

- Familiar with VCS
- **Basic understanding** of how Git operates
- Can **use basic Git commands** and know what they do to the filesystem

Content

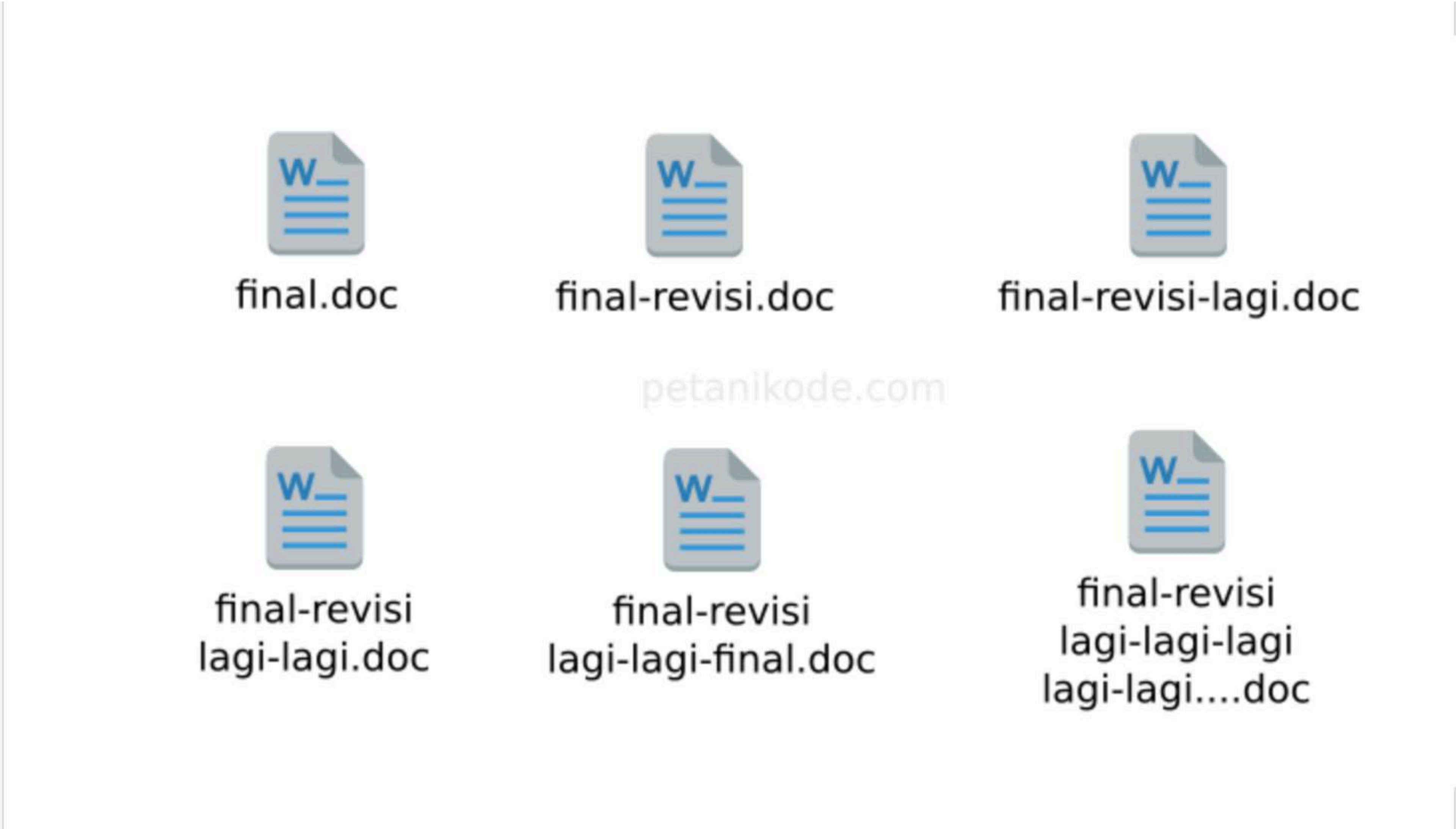
- Version Control (VC)
- About Git
- Git in practice

What’s VC about?

- It’s about file history

```
> touch foo.js      // Create a new file
…                   // Edit
> cp foo.js bar.js // Create a backup
…                   // Do something horrible
> cp bar.js foo.js // Revert to backup
```

Before



final.doc

final-revisi.doc

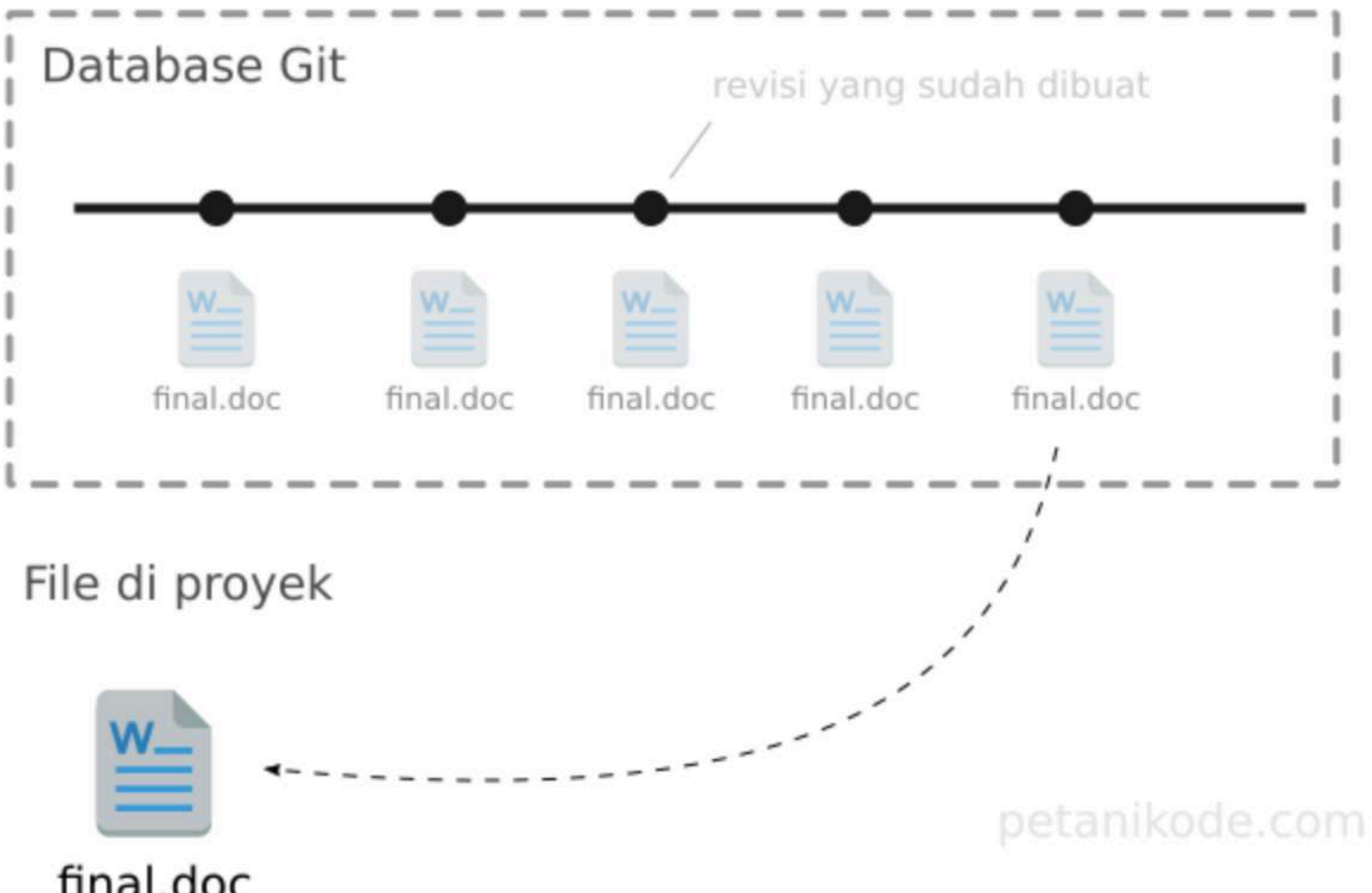
final-revisi-lagi.doc

final-revisi
lagi-lagi.doc

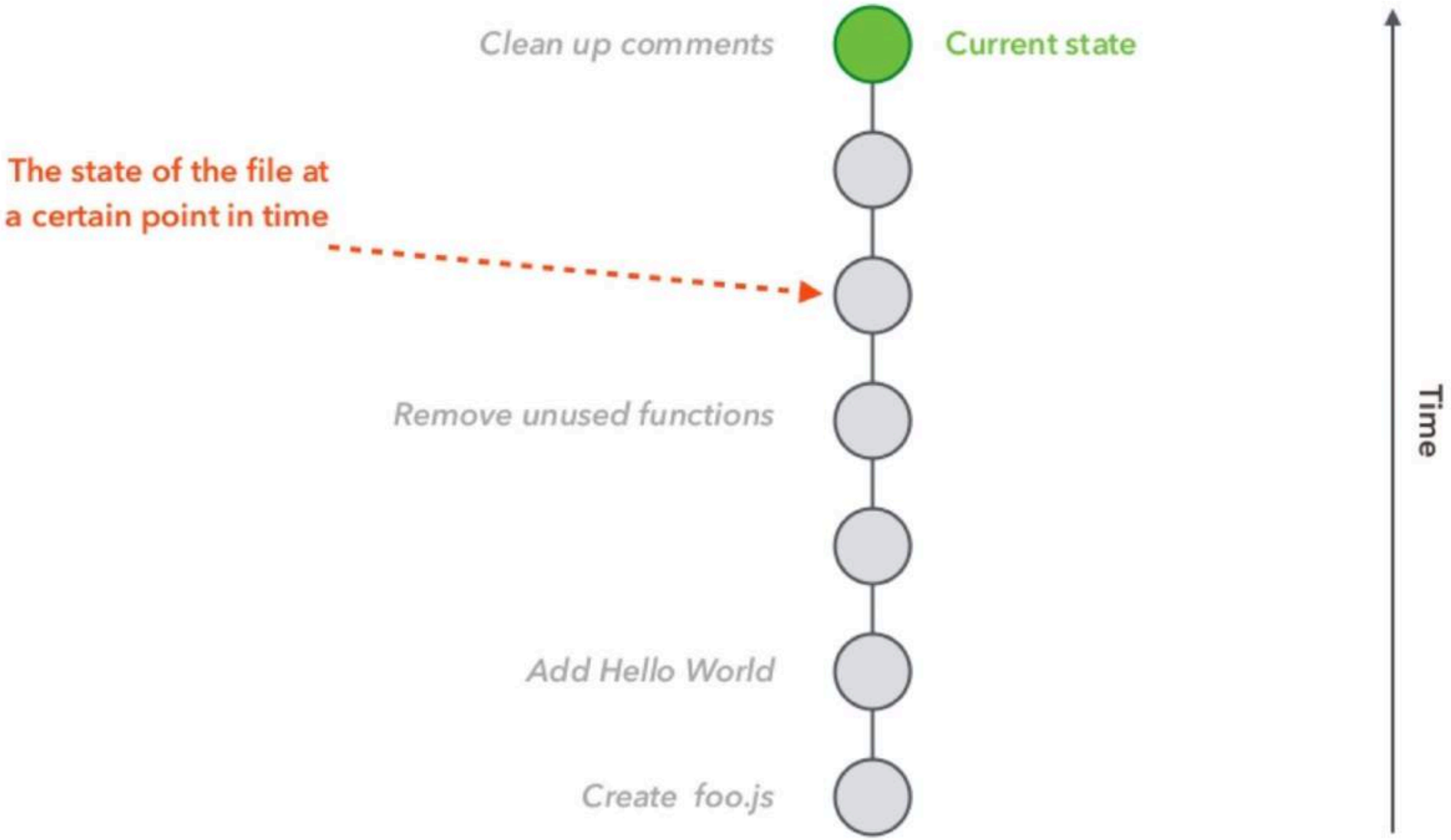
final-revisi
lagi-lagi-final.doc

final-revisi
lagi-lagi-lagi
lagi-lagi....doc

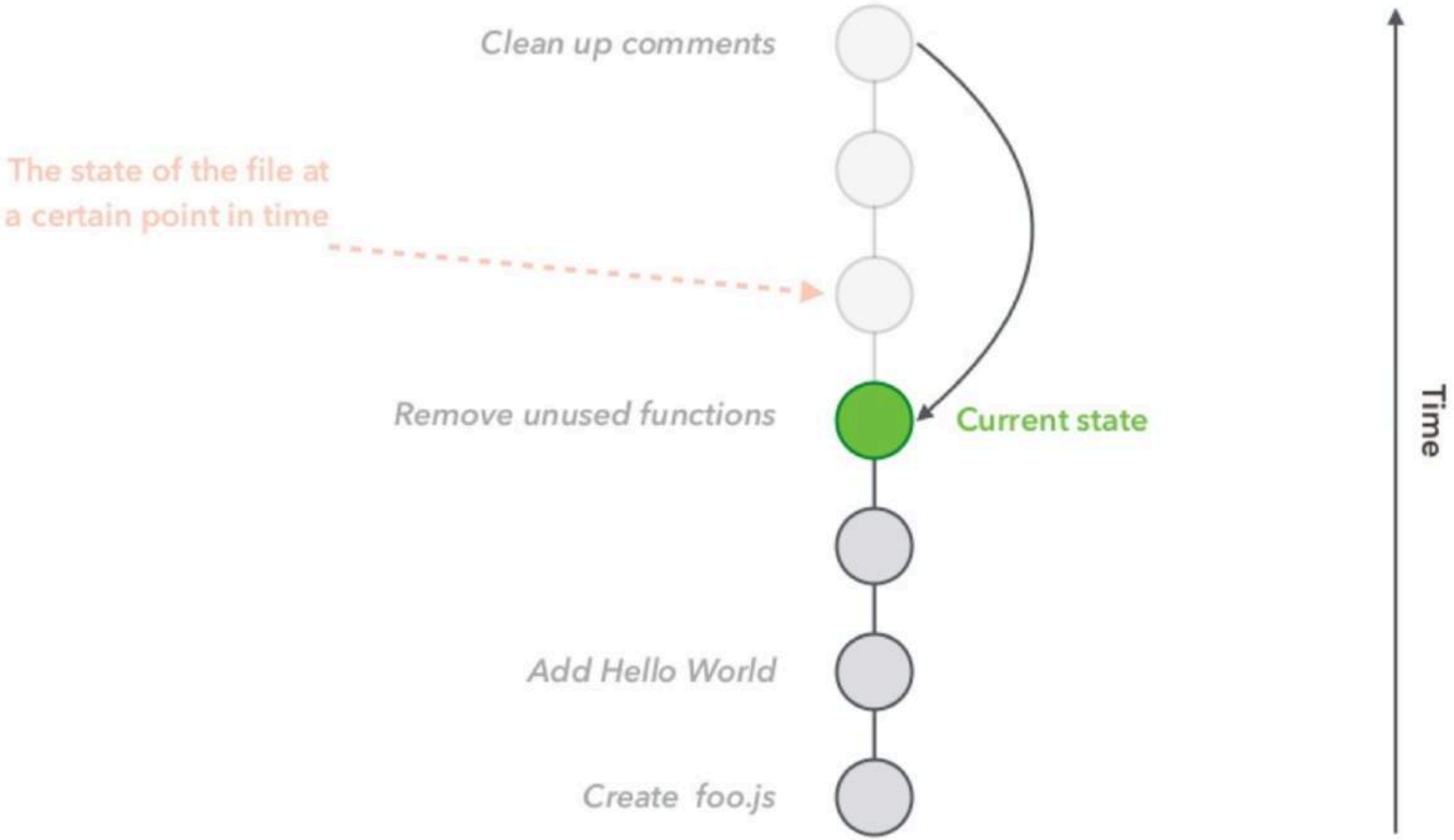
After


Database Git
revisi yang sudah dibuat
final.doc
final.doc
final.doc
final.doc
final.doc
File di proyek
final.doc
petanikode.com

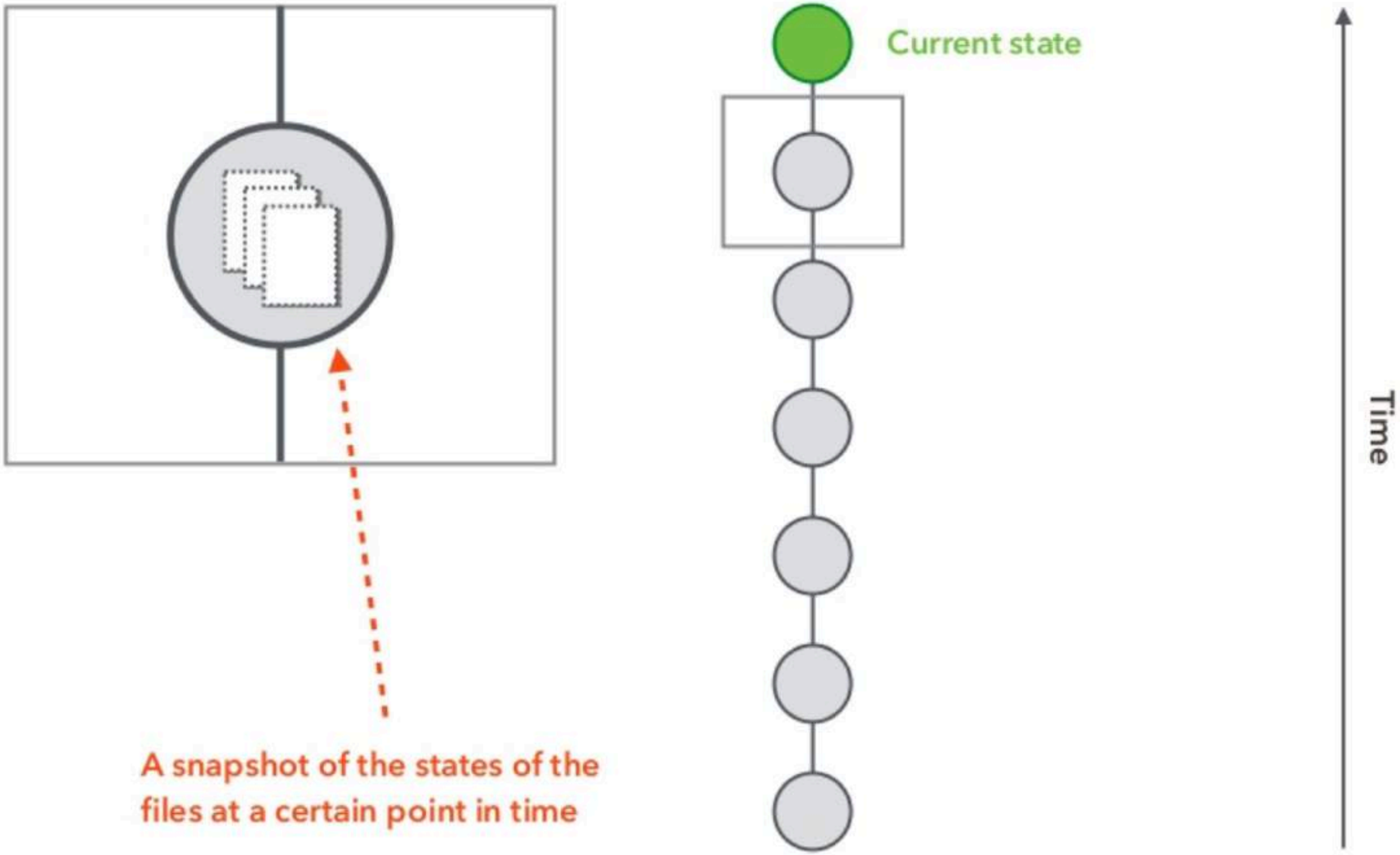
File


Clean up comments
Current state
The state of the file at a certain point in time
Remove unused functions
Add Hello World
Create foo.js
Time

File


Clean up comments
The state of the file at a certain point in time
Remove unused functions
Current state
Add Hello World
Create foo.js
Time

File(s)



A snapshot of the states of the files at a certain point in time

Git

Git

Git (git) is a VCS, a program that is run from the CLI

```
> git <command>
```

First Git

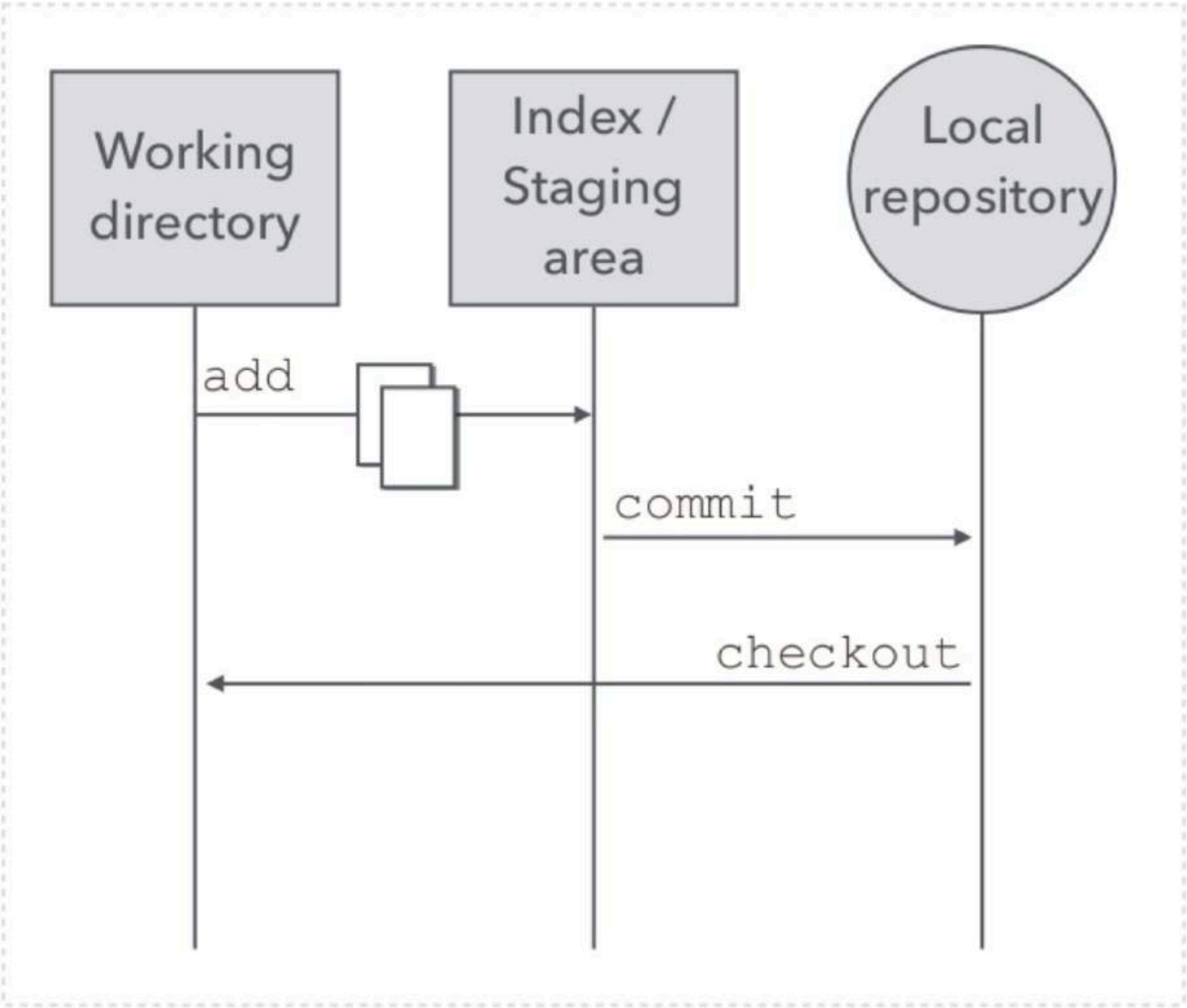
Start a new project

```
> git init
```

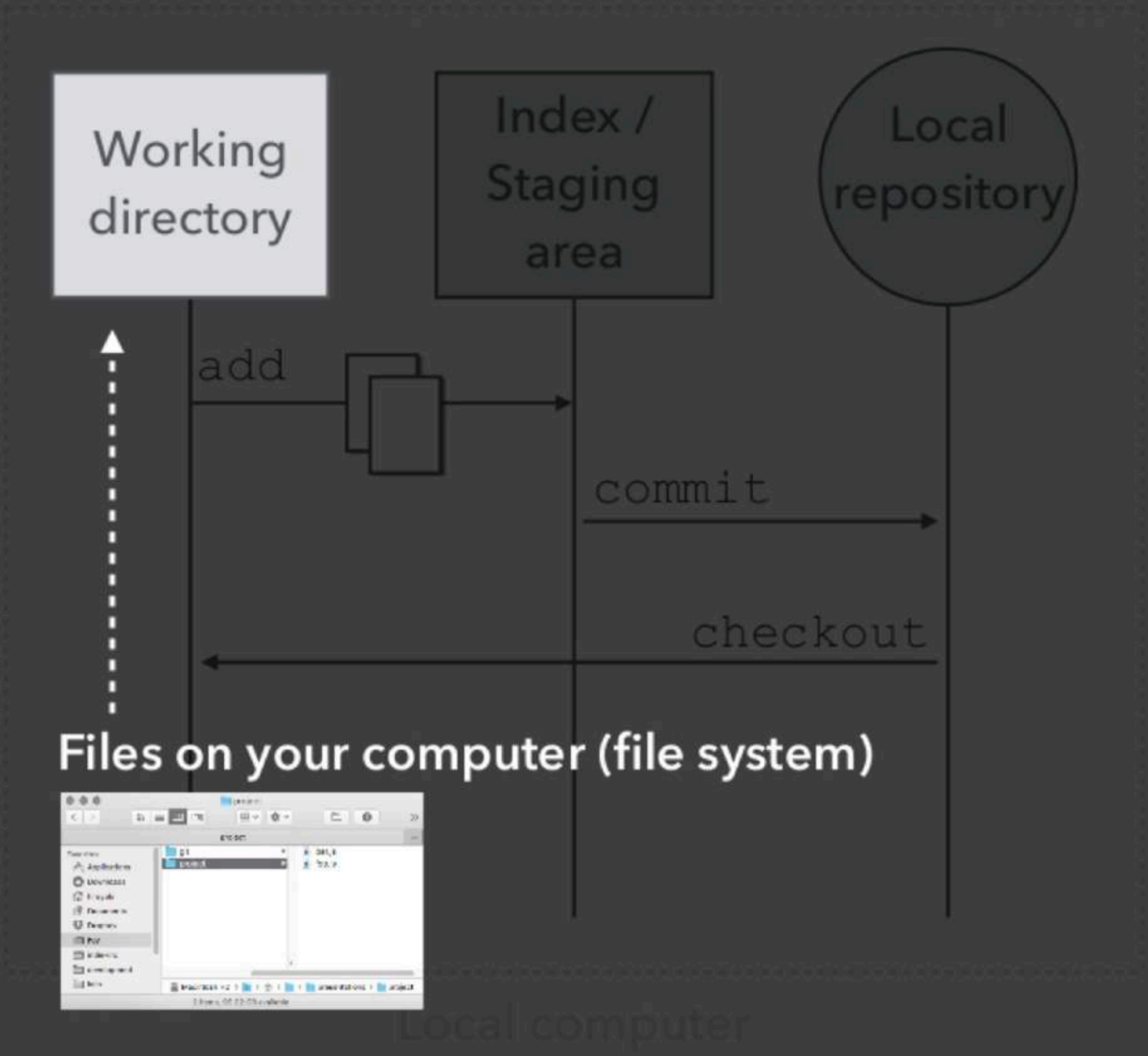
Download an existing project

```
> git clone <url>
```

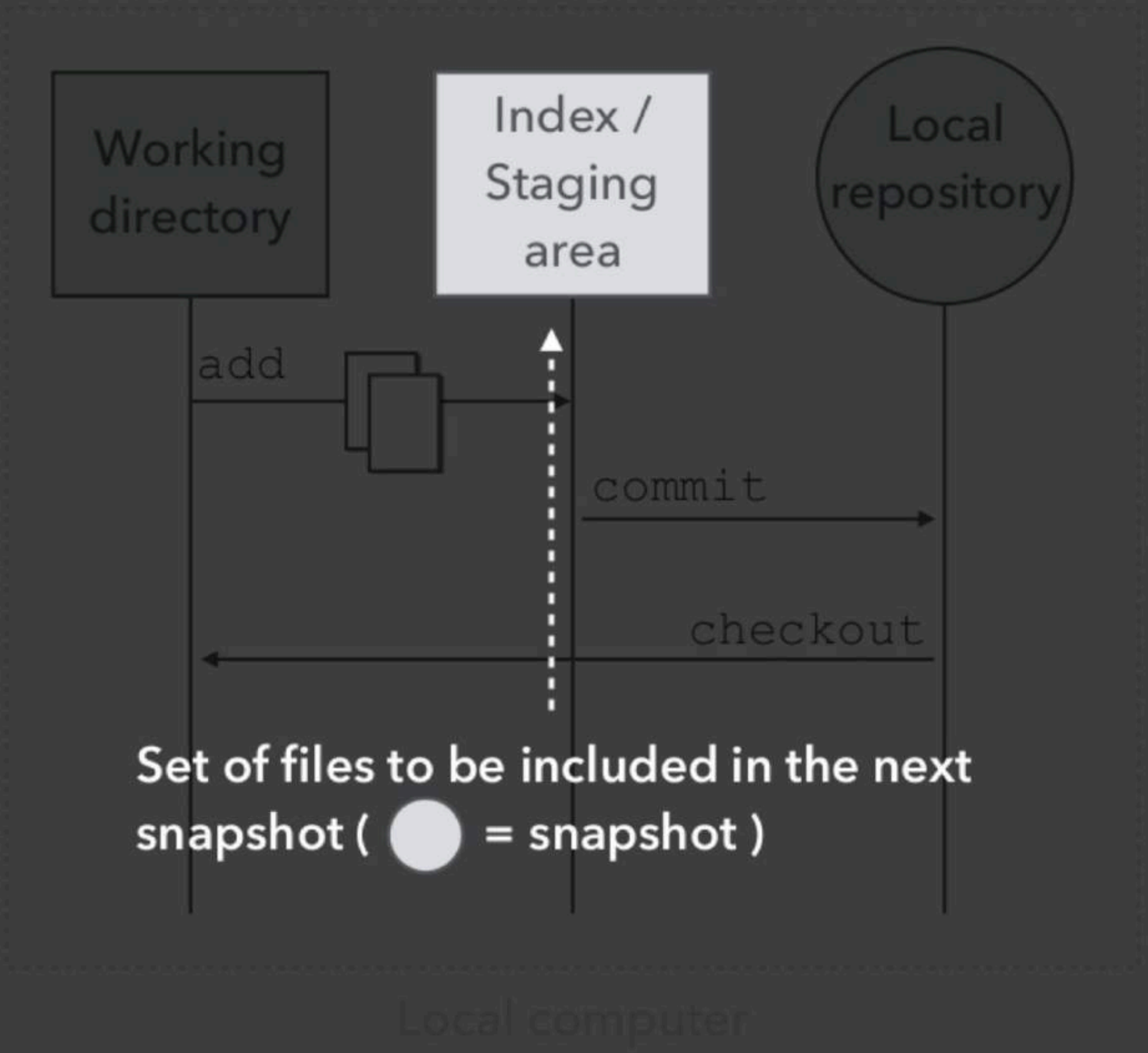
Overview



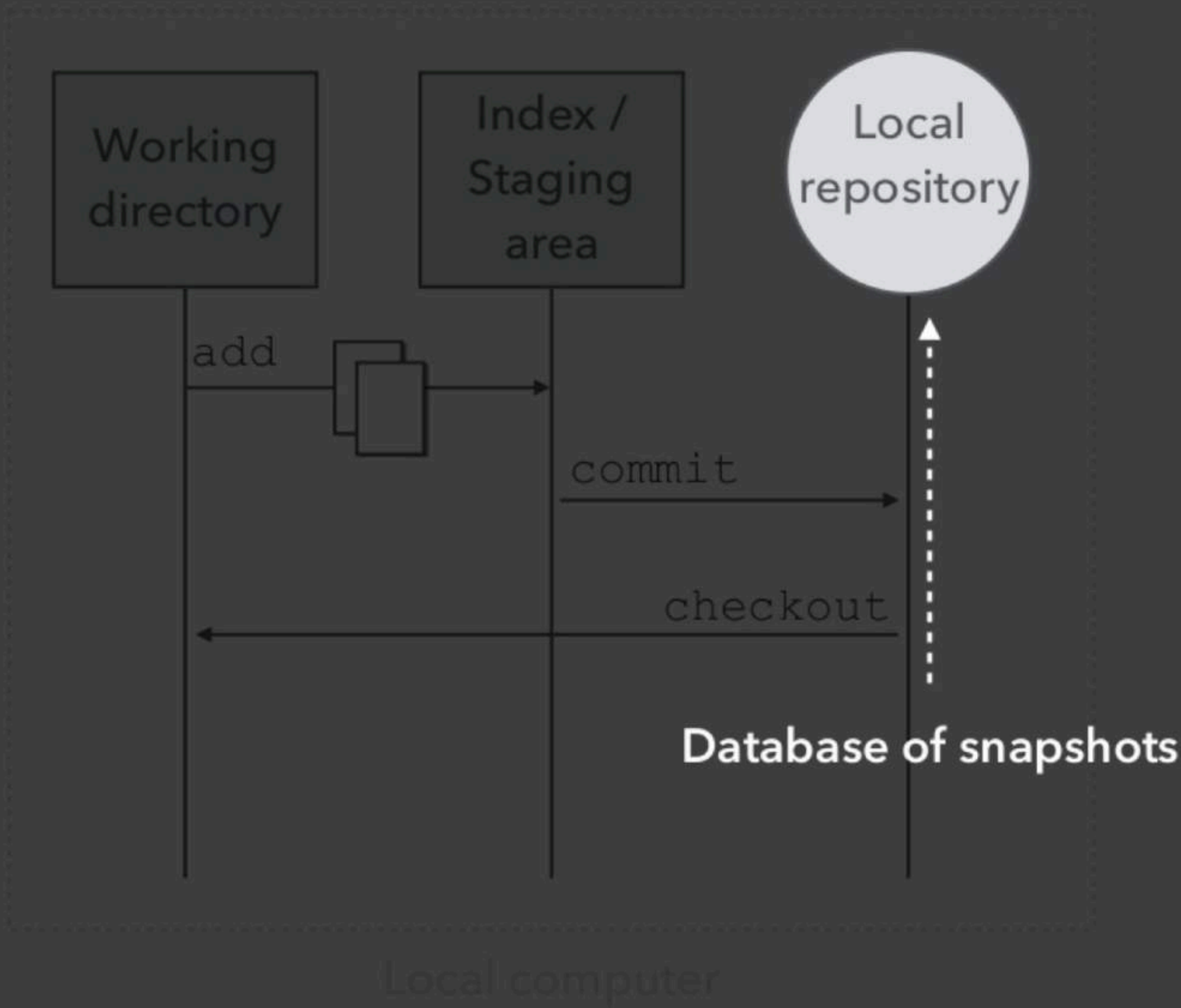
Local computer



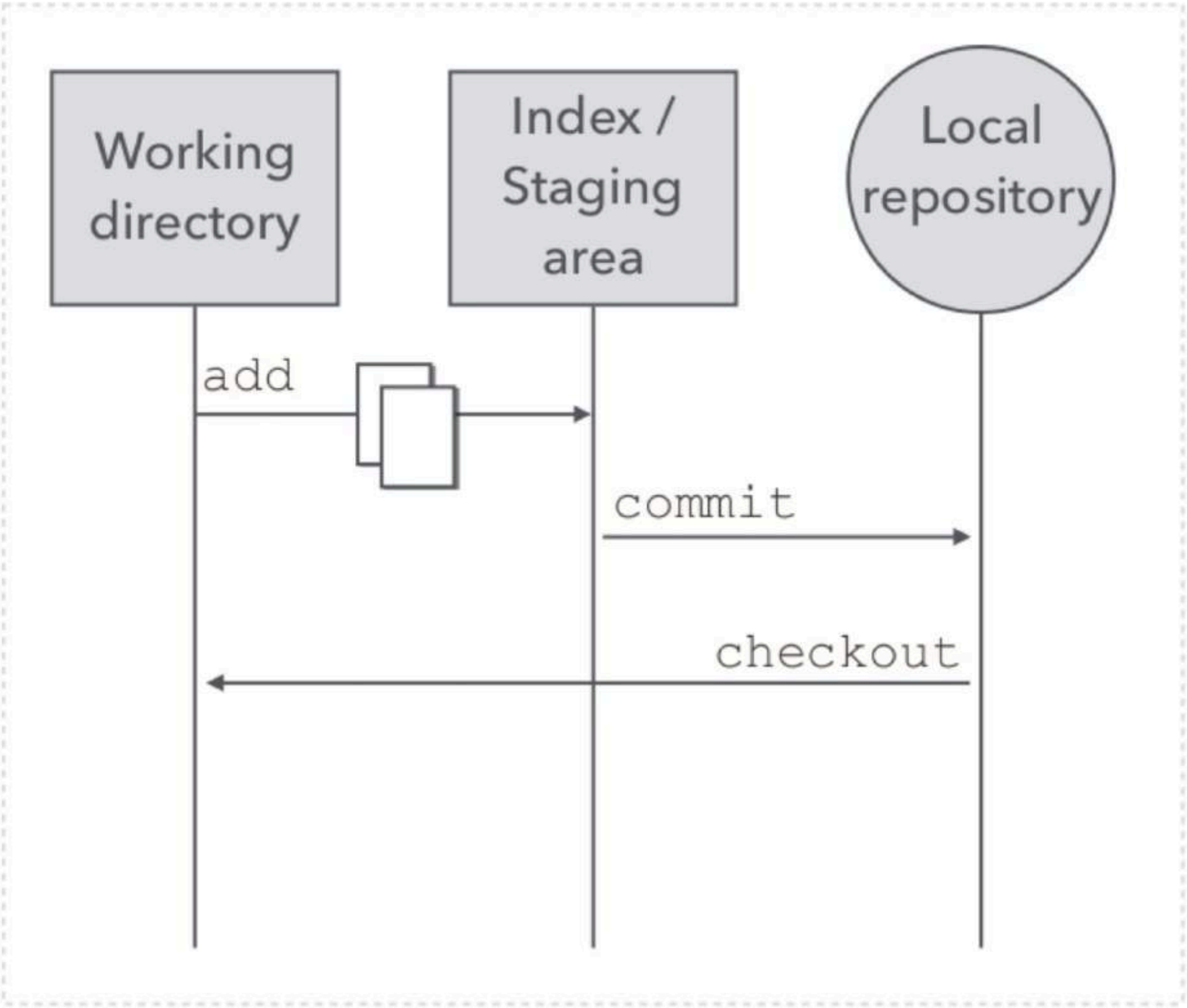
Files on your computer (file system)



Set of files to be included in the next snapshot (● = snapshot)



Database of snapshots

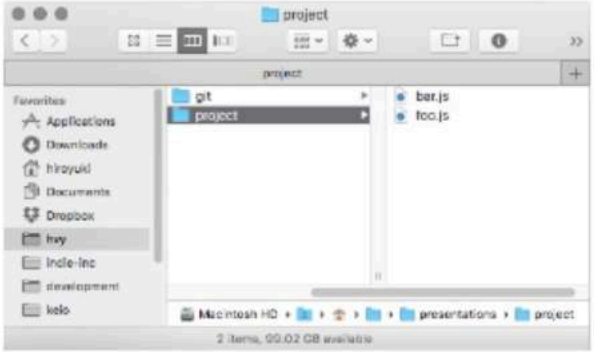


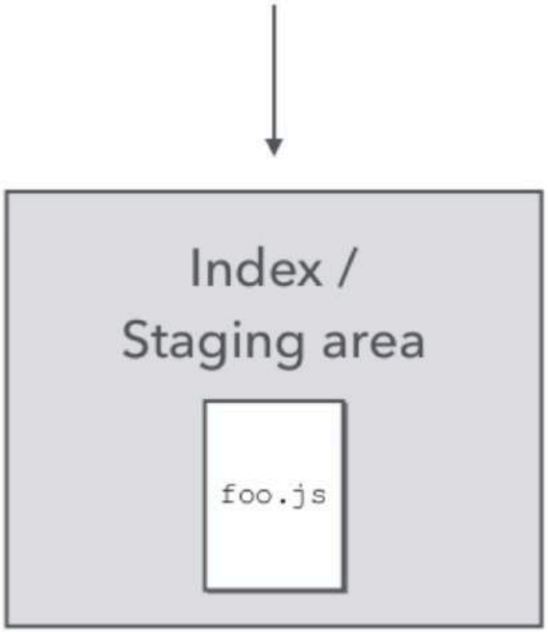
Local computer

Add

Tell Git that a file is to be **included in the next snapshot** by adding it to the index / staging area

```
> git add foo.js
```





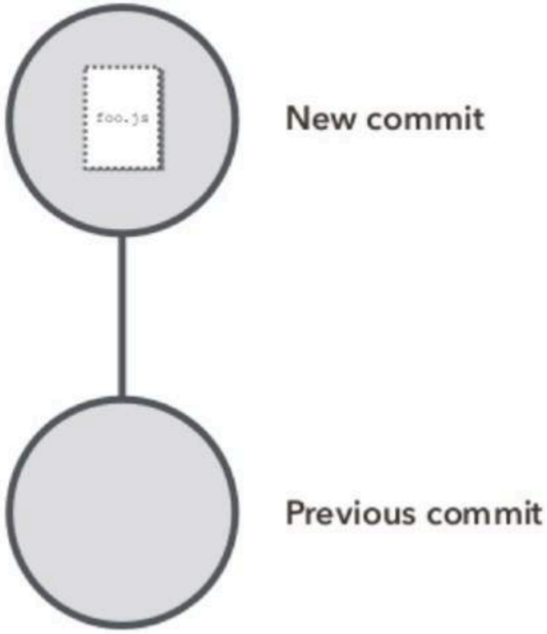
Index /
Staging area

Commit

Take a new snapshot of the state of the files in the staging area (append it to the previous commit)

Store a snapshot in the local repository

```
> git commit
```

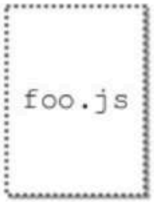


References

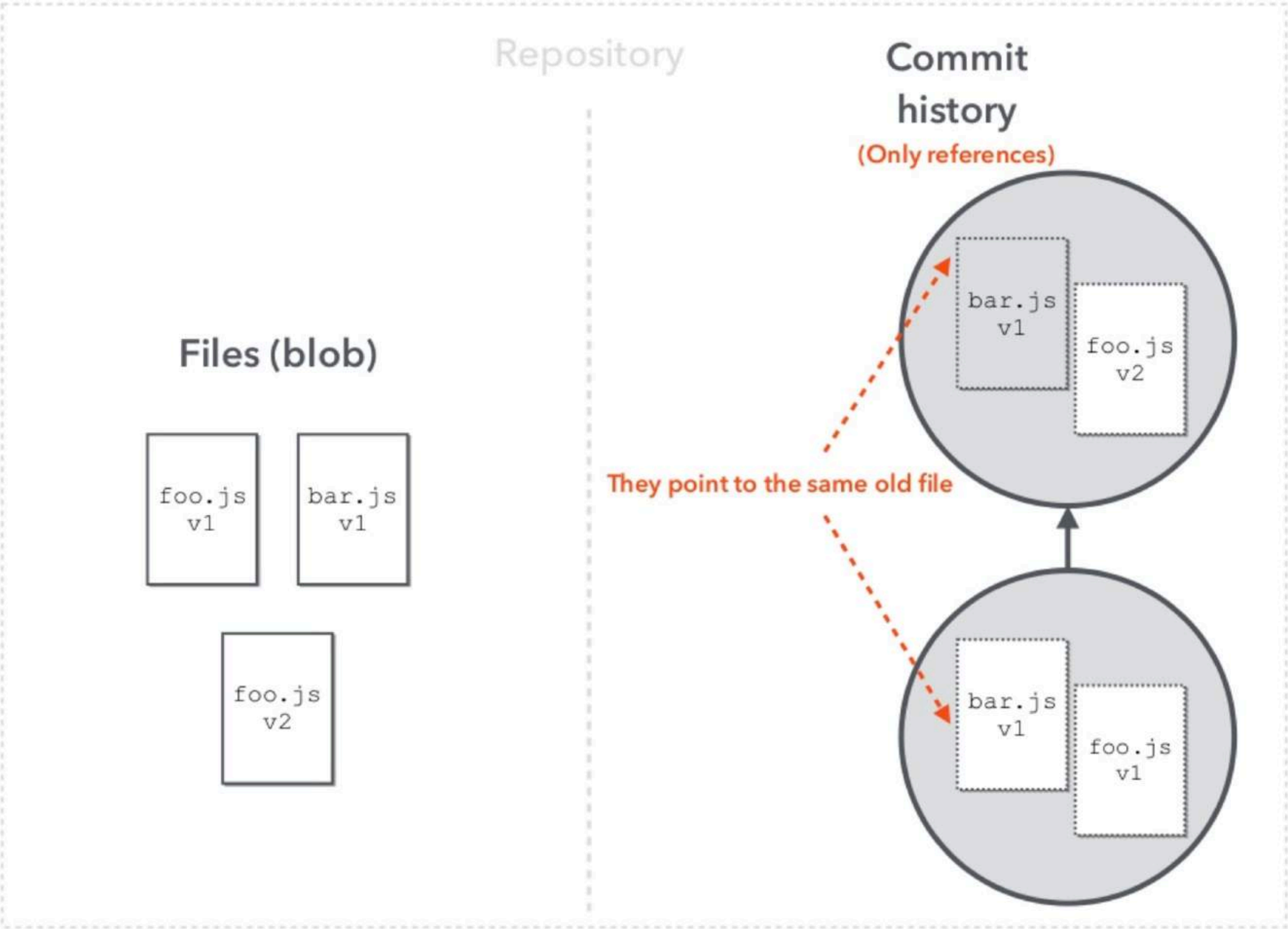
Git only creates new files when a file is modified

Files are not stored in a commit, but rather **references** to files

File `foo.js`



Reference to `foo.js`



Repository

Files (blob)

Commit history

(Only references)

They point to the same old file

Inside a Commit

Commit Reference (SHA-1 checksum)

Author

Date

Message

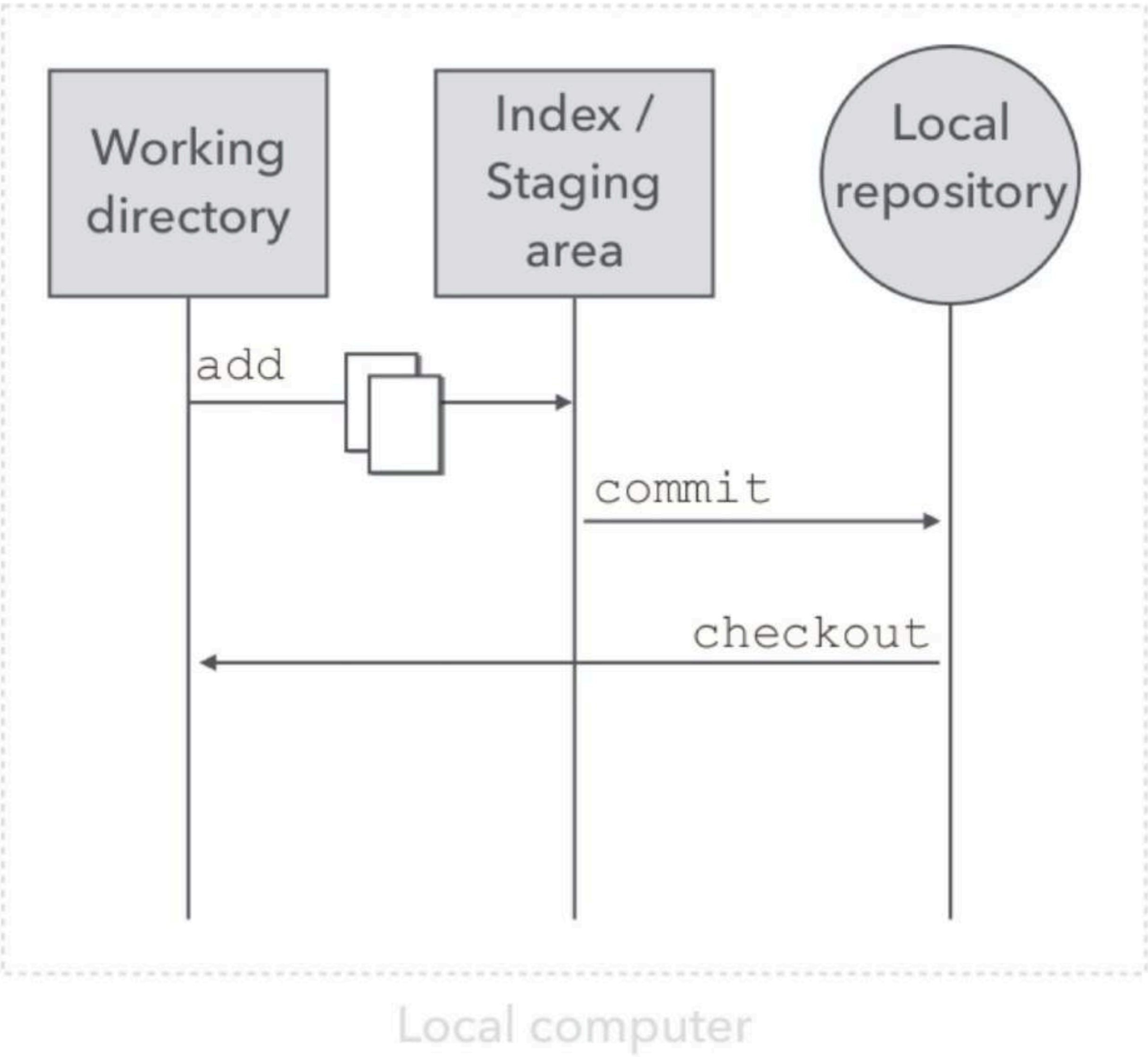
Reference to parent commit(s)

Tree

Inside a Commit

Commit Reference (SHA-1 checksum)
Author
Date
Message
Reference to parent commit(s)
Tree

Includes the references to
all the files in the snapshot

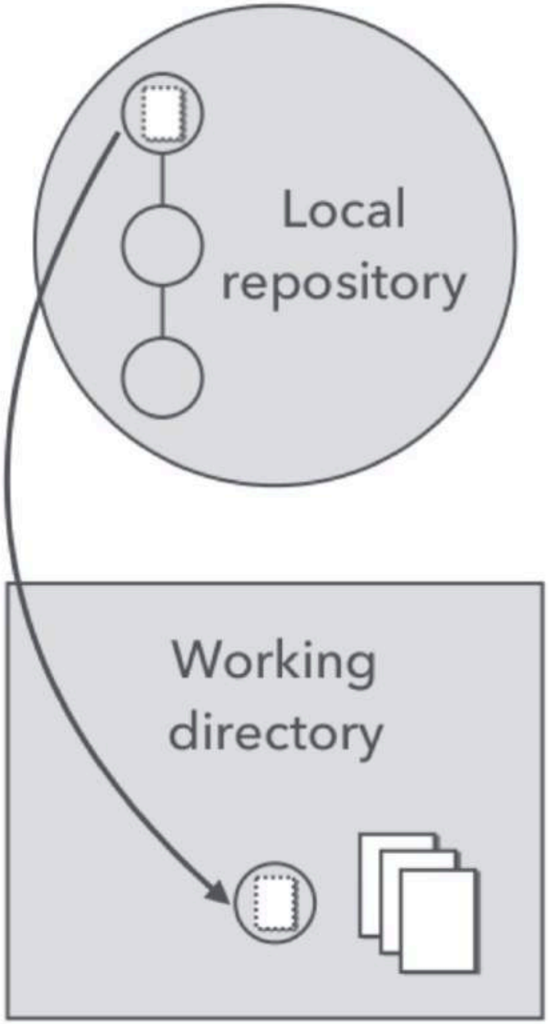

Working directory
Index / Staging area
Local repository
add
commit
checkout
Local computer

Checkout

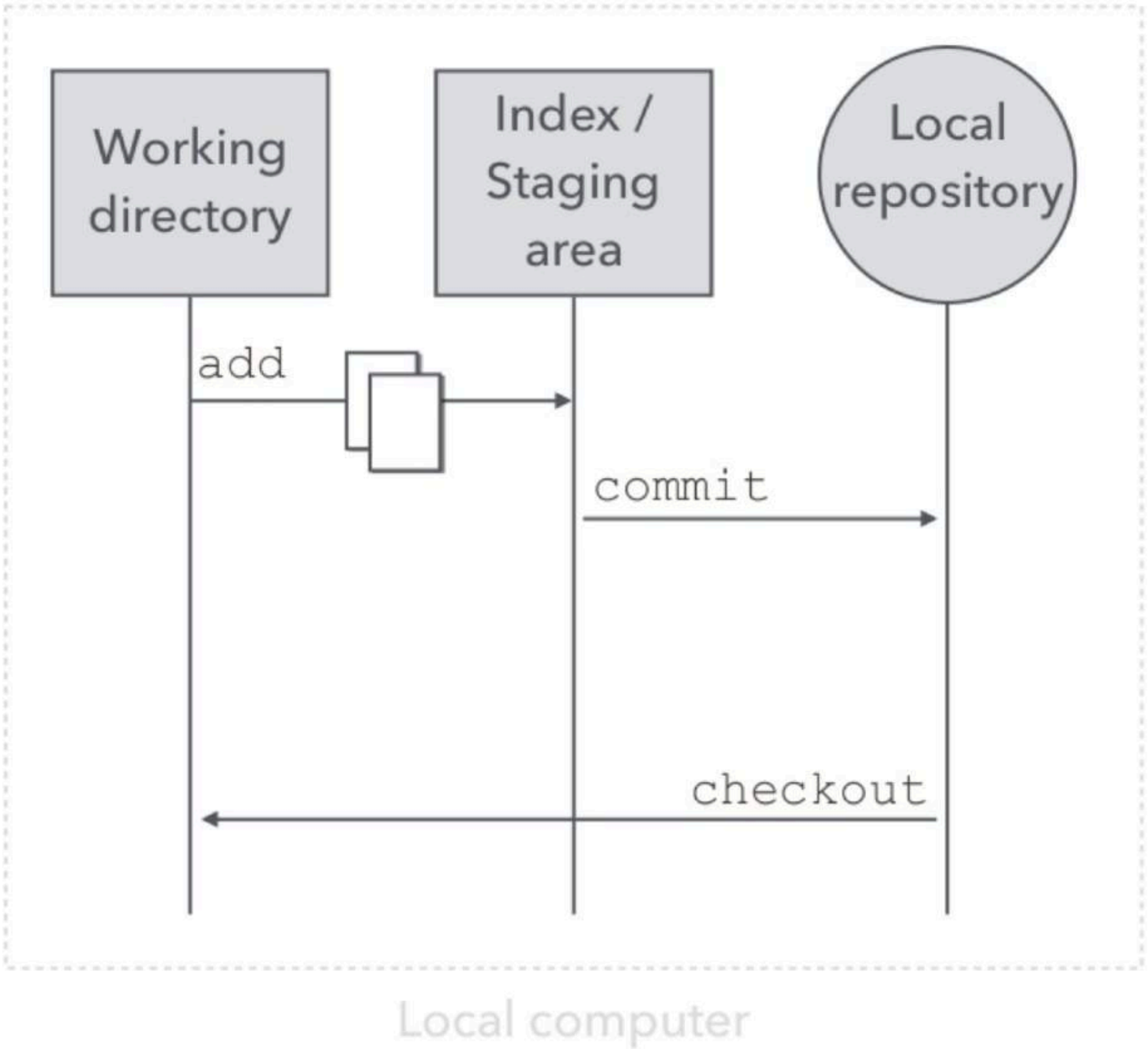
Get the data **from the repository into the working directory**

Again, not the files but the references to the files from a commit

```
> git checkout <commit>
```



Collaboration



Working directory

Index / Staging area

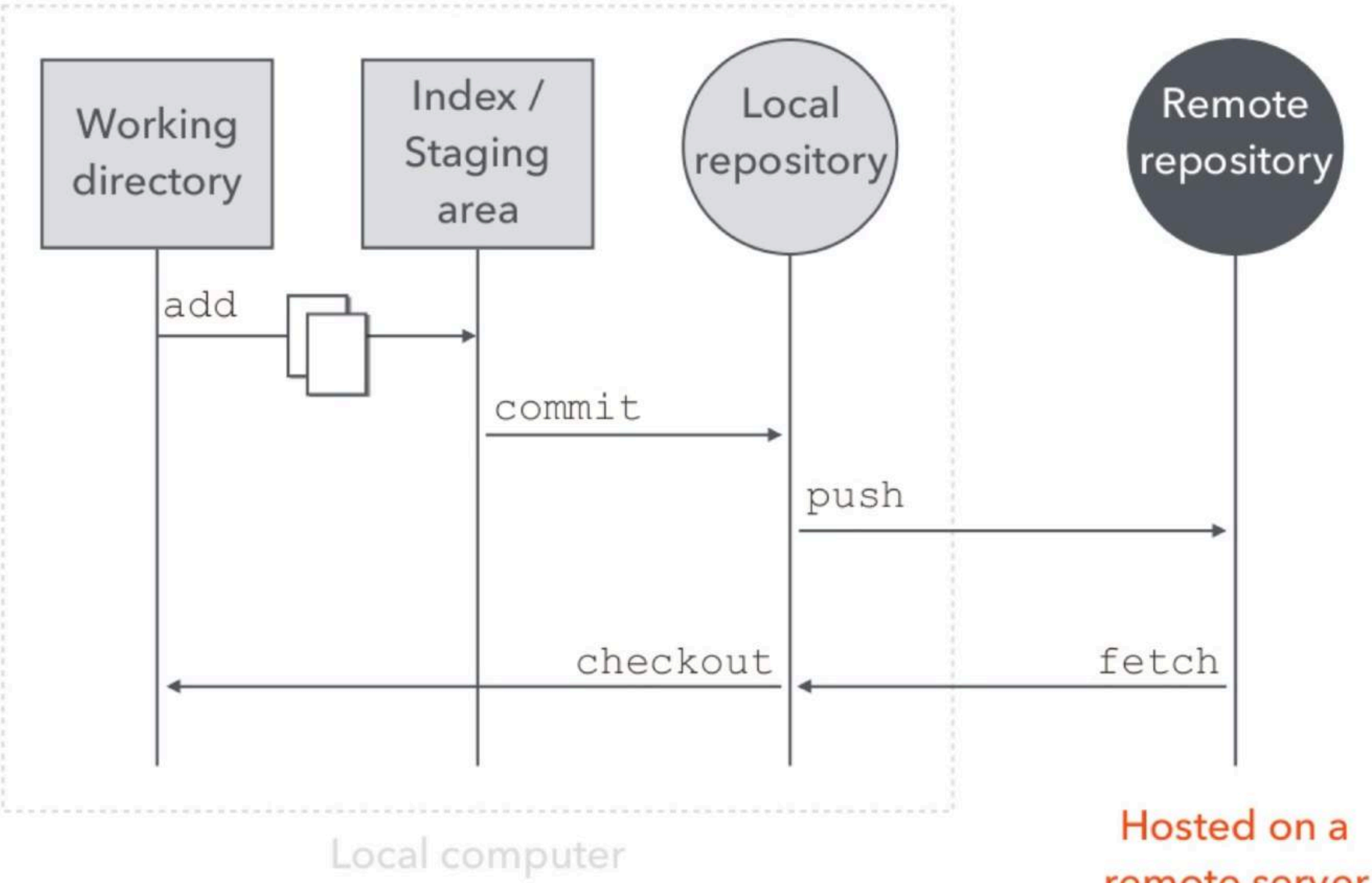
Local repository

add

commit

checkout

Local computer



Hosted on a remote server

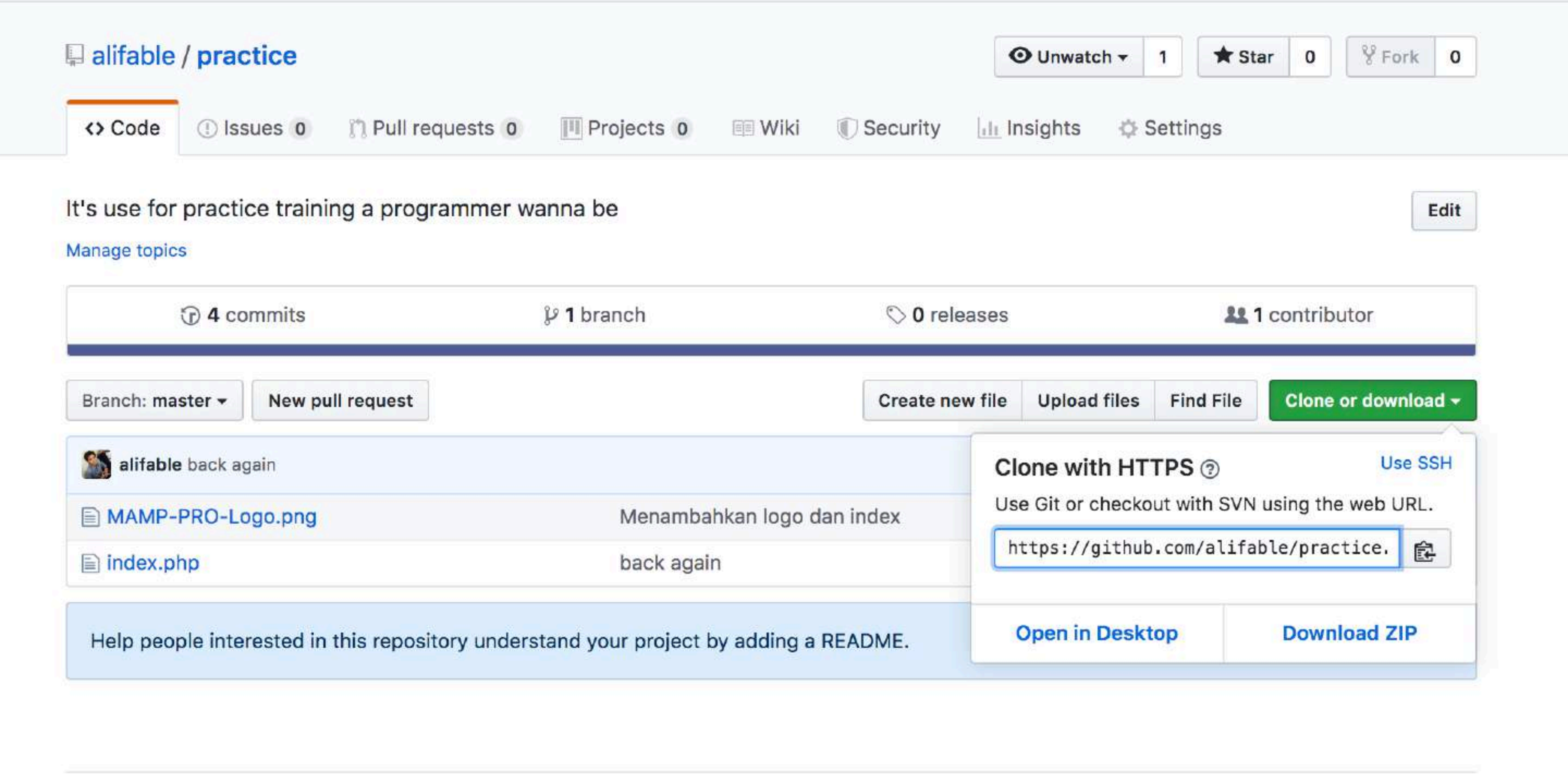
Remote Repository

A copy of the local repository, **hosted** on a network

Other people can join a project, by **cloning** its remote repository

Note: GitHub is a service that provides this remote repository hosting

https://github.com/alifable/practice



alifable / practice

It's use for practice training a programmer wanna be

4 commits · 1 branch · 0 releases · 1 contributor

Branch: master

alifable back again

MAMP-PRO-Logo.png	Menambahkan logo dan index
index.php	back again

Help people interested in this repository understand your project by adding a README.

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

```
https://github.com/alifable/practice.
```

Demo

1. Create a new **local repository**

2. Create a file and make a new commit to the local repository

3. Associate the local repository with a **remote repository**

4. **Push** it to the remote repository

Demo Cont.

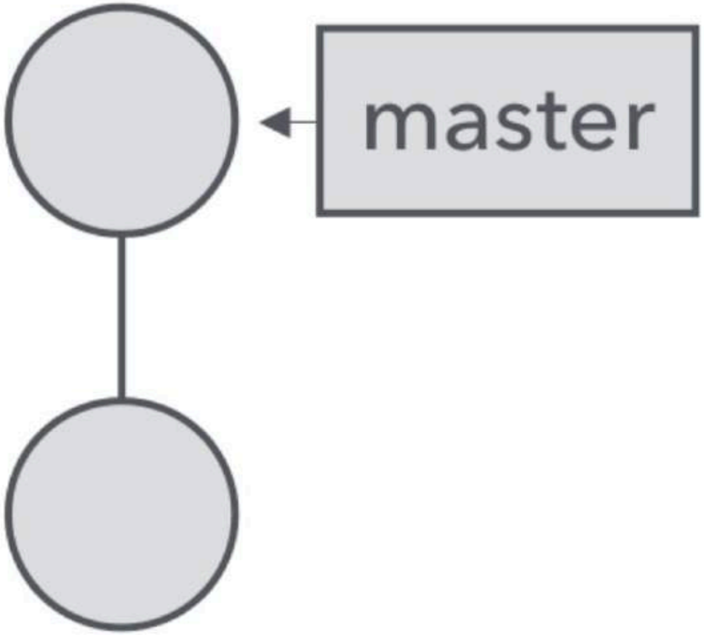
1. **Clone** an existing repository
2. Create a new **branch**
3. Make a commit to the new branch
4. Switch between branches (checkout and change HEAD), see the changes in the local filesystem
5. **Push** the new branch to the remote repository

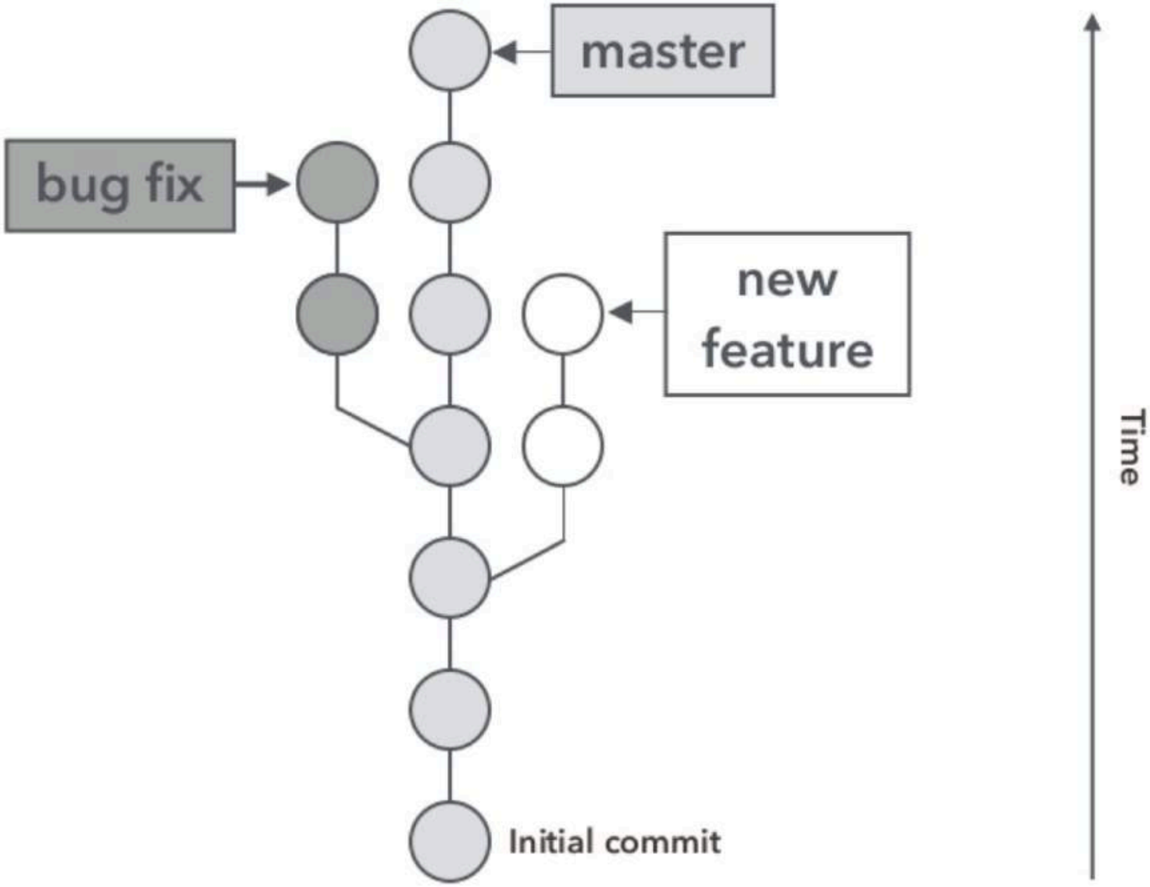
Branch

Branch

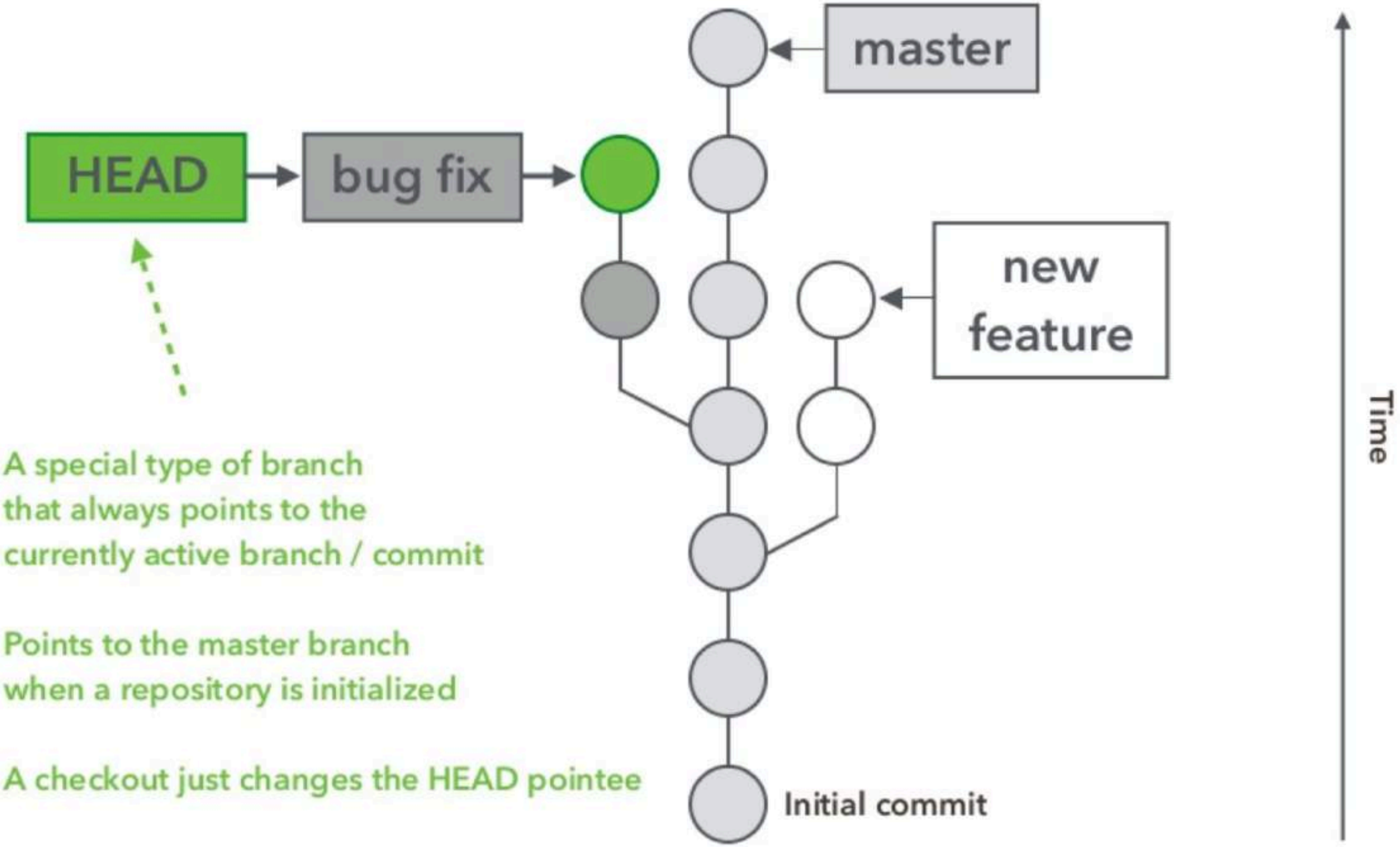
A branch is a **pointer** to a commit

master, is the name of the default branch





Example with 3 branches

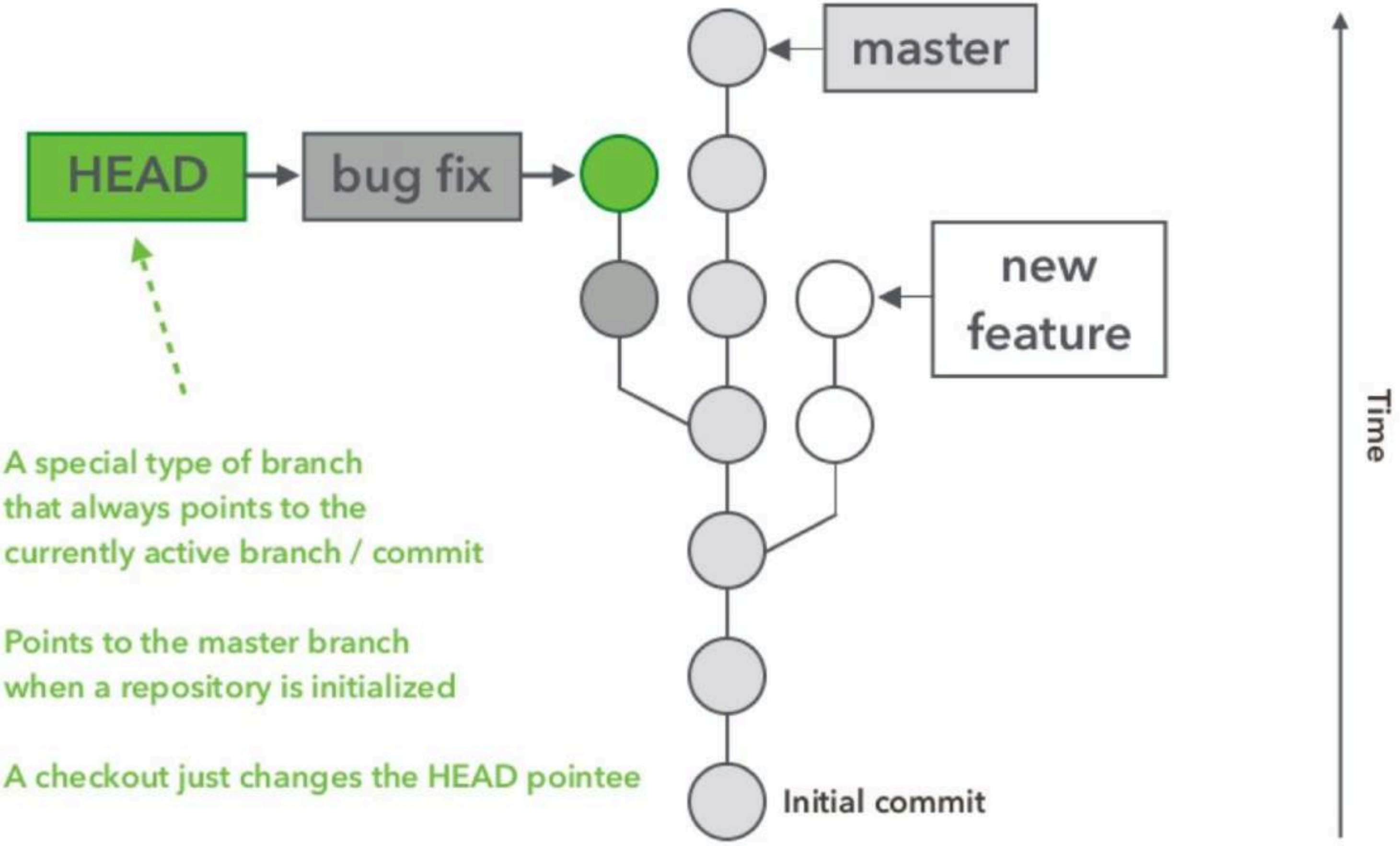


A special type of branch that always points to the currently active branch / commit

Points to the master branch when a repository is initialized

A checkout just changes the HEAD pointee

Example with 3 branches



A special type of branch
that always points to the
currently active branch / commit

Points to the master branch
when a repository is initialized

A checkout just changes the HEAD pointee

Example with 3 branches

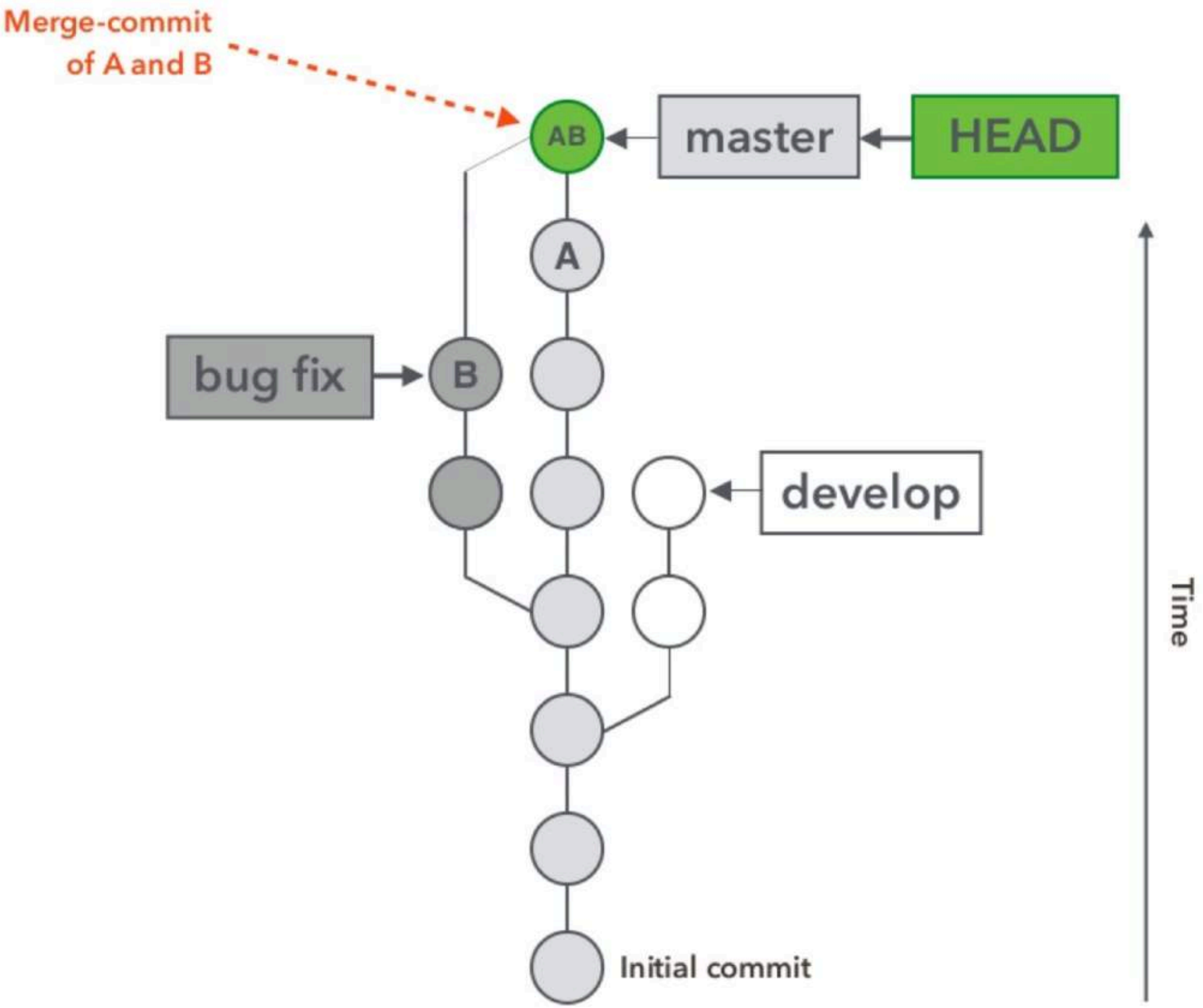
Merge

Merge two commits together

Can be used to merge branches

May cause **conflicts** between commits that need to be resolved manually

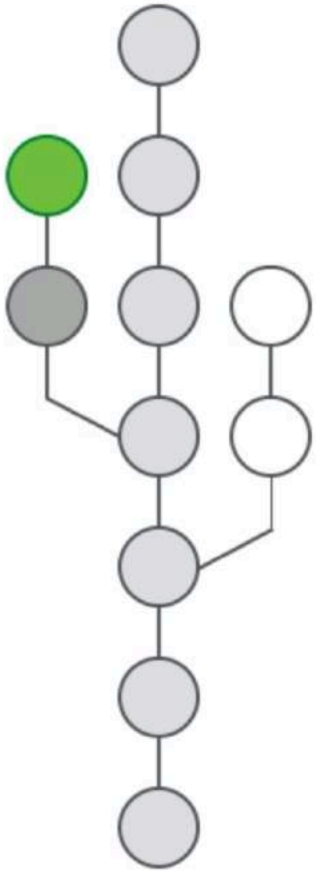
```
// Merge <commit> into HEAD pointee

> git merge <commit>
```


Merge-commit
of A and B
AB
master
HEAD
A
bug fix
B
develop
Time
Initial commit

Summary

Version Control

Series of commits



Three components

Working directory

Index / Staging area

Repository, local and remote

Commit ○

Snapshot of file references (and other metadata)

Branch / HEAD

Just a pointer to a commit

More Git Features

Reset, --soft, --mixed, --hard

Revert

Fast-Forward / No Fast-Forward merge

Rebase

Cherrypick

Cheatsheet 💯 *

```
> git init

> git status

> git branch [-a] [-v]

> git remote add <name> <url>

> git fetch

> git add <file>

> git commit -m “<message>”

> git push <repository> <branch>

> git pull <repository> <branch>

> git checkout [-b] <branch> -- <file>

> git log [--graph]

> git merge

> git --help

> git revert <commit>

> git reset --hard HEAD
```

* Actually not even close to 100