

ANDROID COURSE

GIT FLOW

vcs

ДЛЯ ЧЕГО НУЖЕН VCS?

- ▶ Записывает все изменения файла и позже мы можем восстановить определенную версию.
- ▶ Что и где было изменено
- ▶ Кто менял
- ▶ Когда были изменения
- ▶ Описание

ВИДЫ VCS

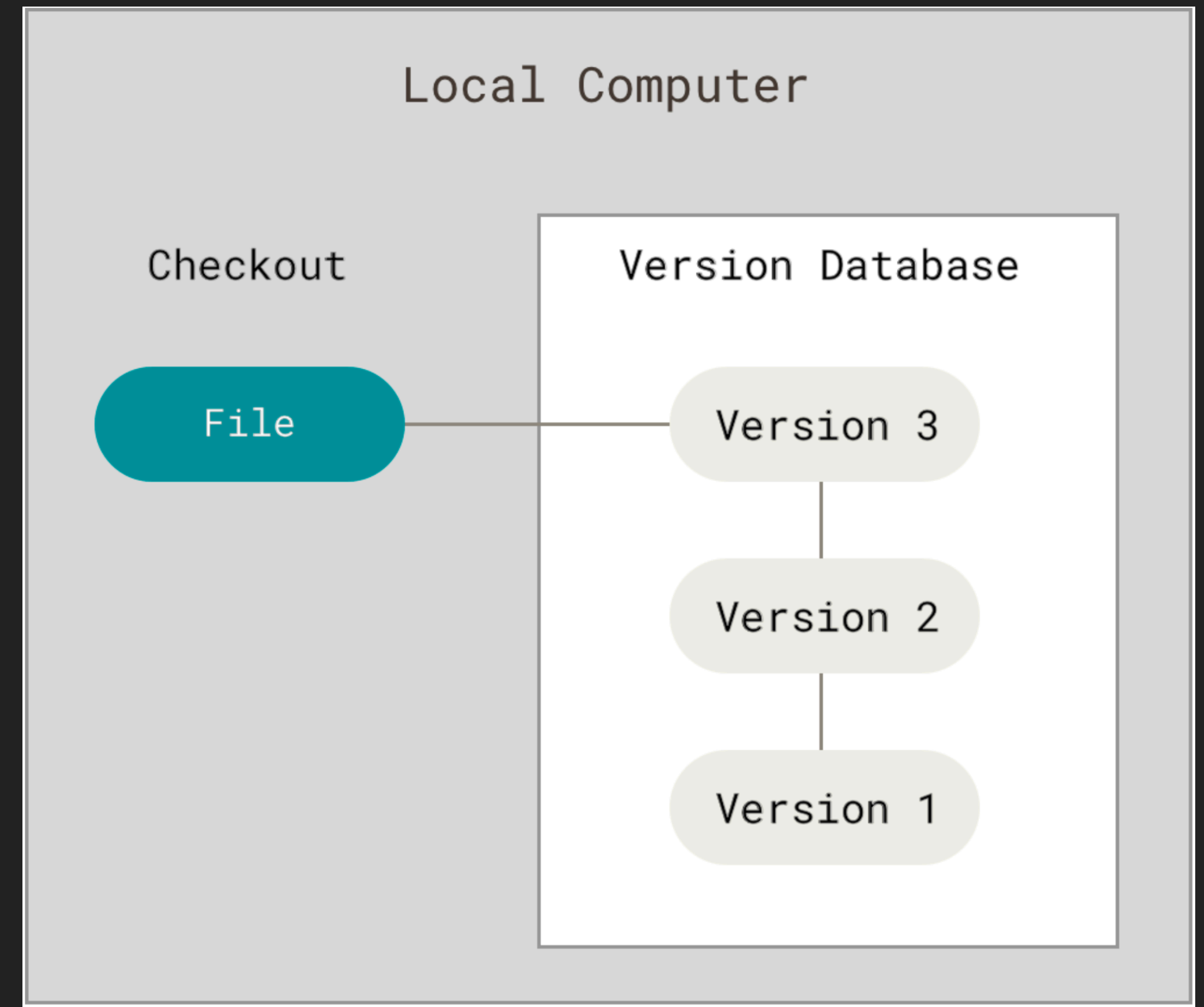
LOCAL VERSION CONTROL SYSTEMS

▶ Плюсы:

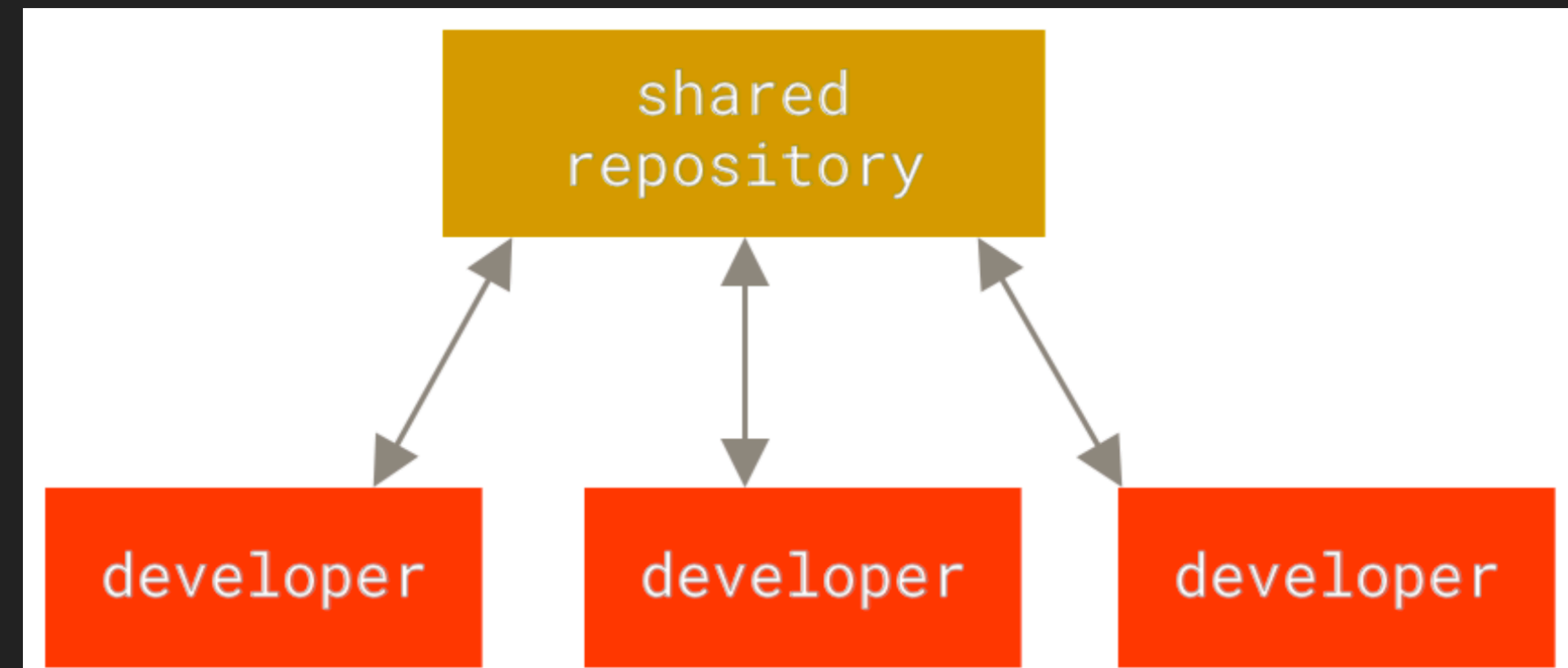
- ▶ 1. Все просто

▶ Минусы:

- ▶ 1. Легко можно ошибиться и записать в другой файл
- ▶ 2. Нет возможности восстановления

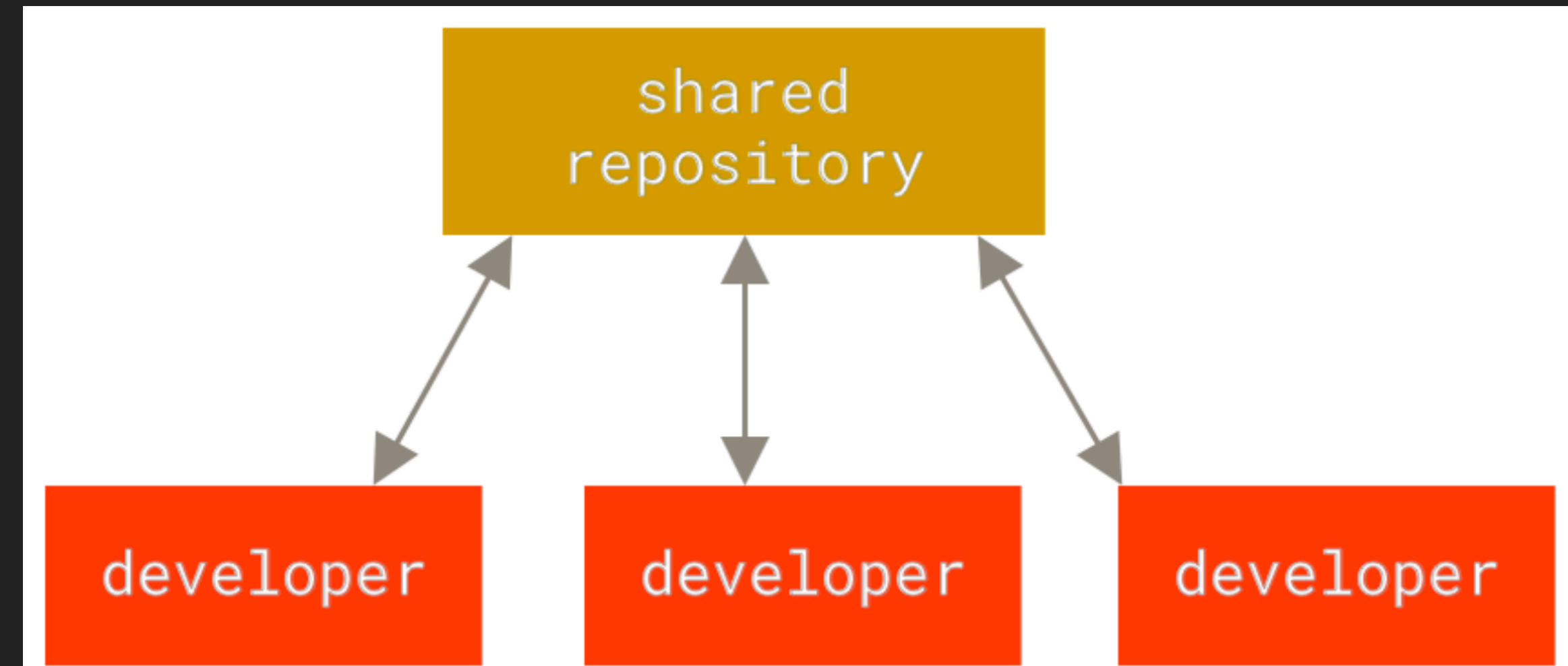


CENTRALIZED VERSION CONTROL SYSTEMS



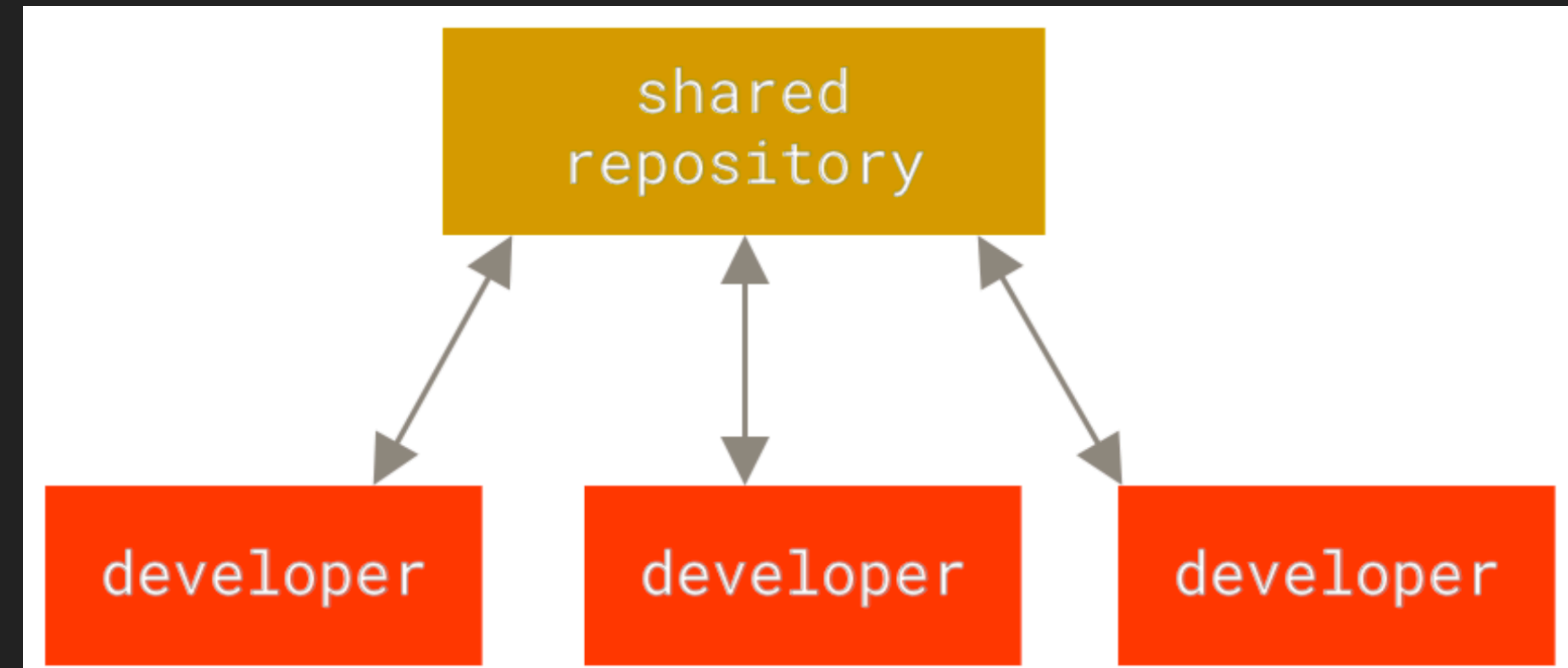
CENTRALIZED VERSION CONTROL SYSTEMS

- ▶ Взаимодействия разработчиков (CVS, Subversion, Perforce)



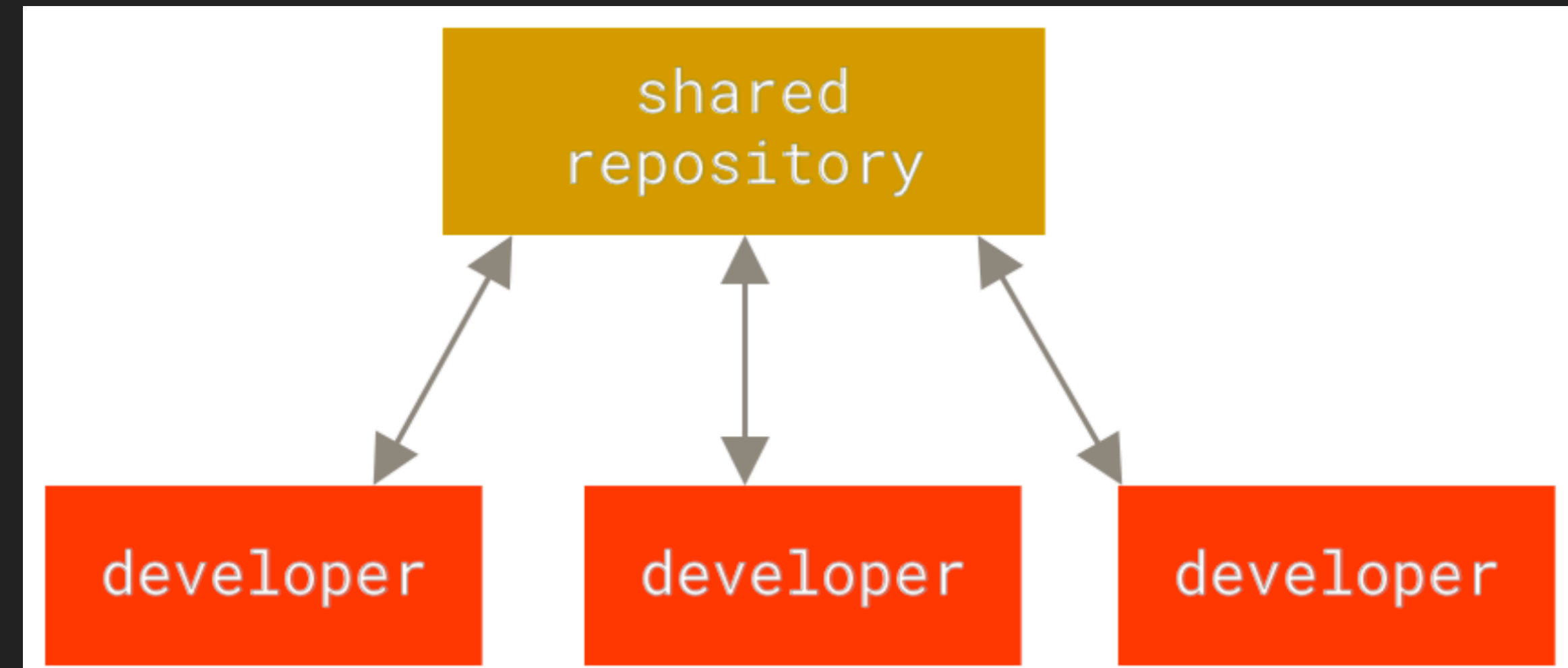
CENTRALIZED VERSION CONTROL SYSTEMS

- ▶ Взаимодействия разработчиков (CVS, Subversion, Perforce)
- ▶ Минус:

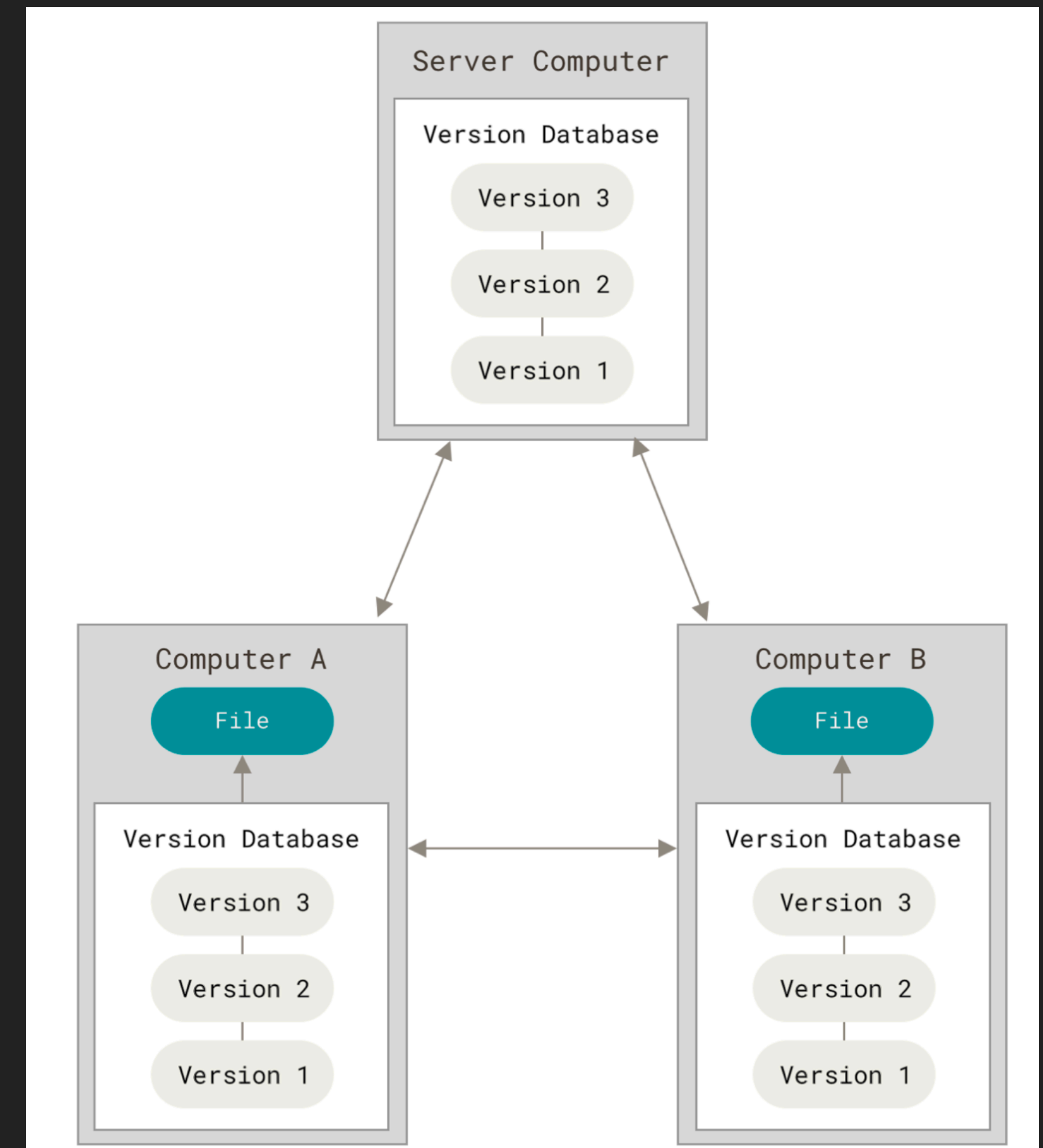


CENTRALIZED VERSION CONTROL SYSTEMS

- ▶ Взаимодействия разработчиков (CVS, Subversion, Perforce)
- ▶ Минус: зависимость от сервера

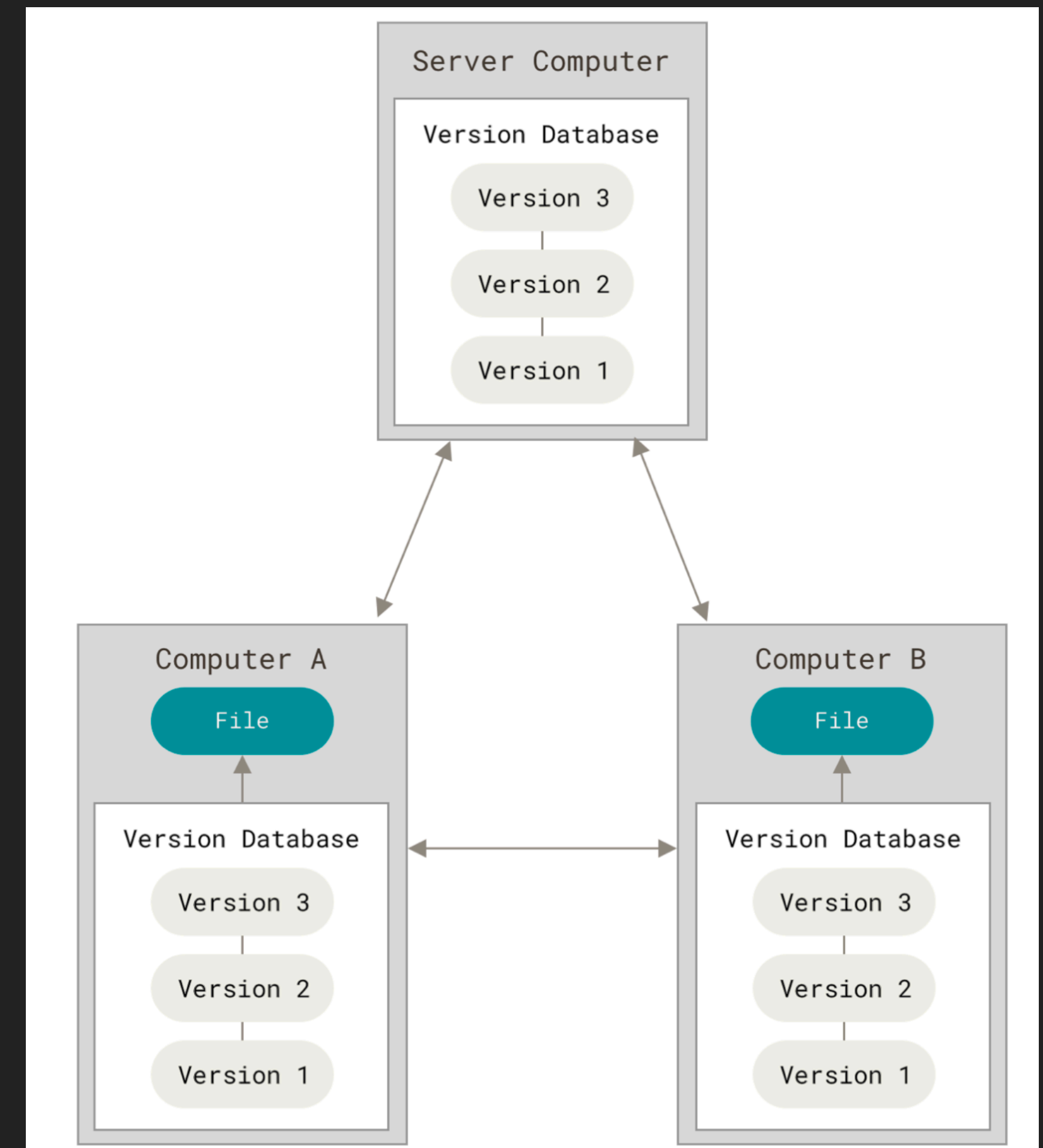


DISTRIBUTED VERSION CONTROL SYSTEMS



DISTRIBUTED VERSION CONTROL SYSTEMS

- ▶ Git, Mercurial, Bazaar or Darcs



ПОЧЕМУ GIT?

GIT

WHY GIT?

WHY GIT?

- ▶ Популярный

WHY GIT?

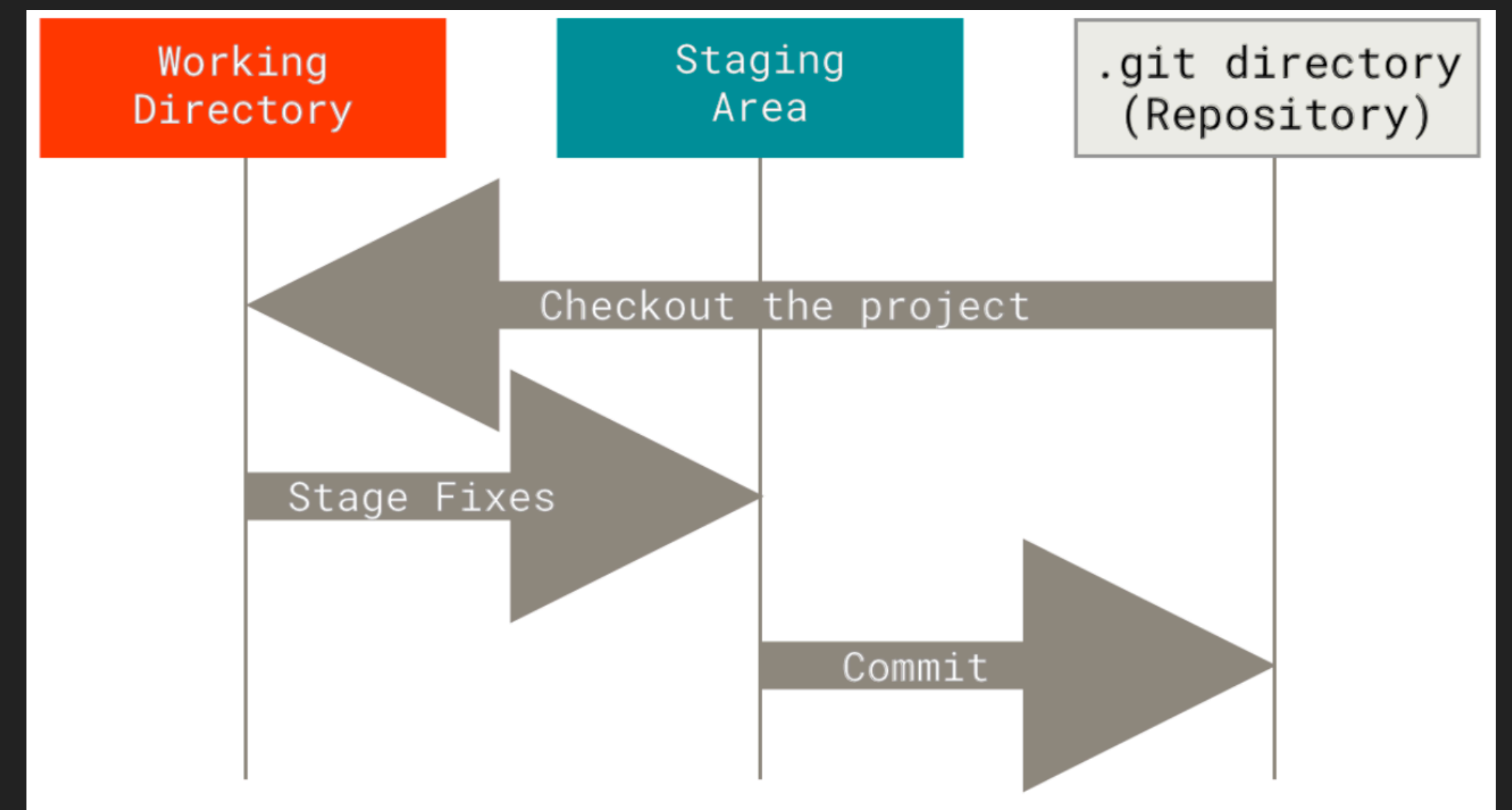
- ▶ Популярный
- ▶ Параллельная разработка

WHY GIT?

- ▶ Популярный
- ▶ Параллельная разработка
- ▶ Snapshots, Not Differences

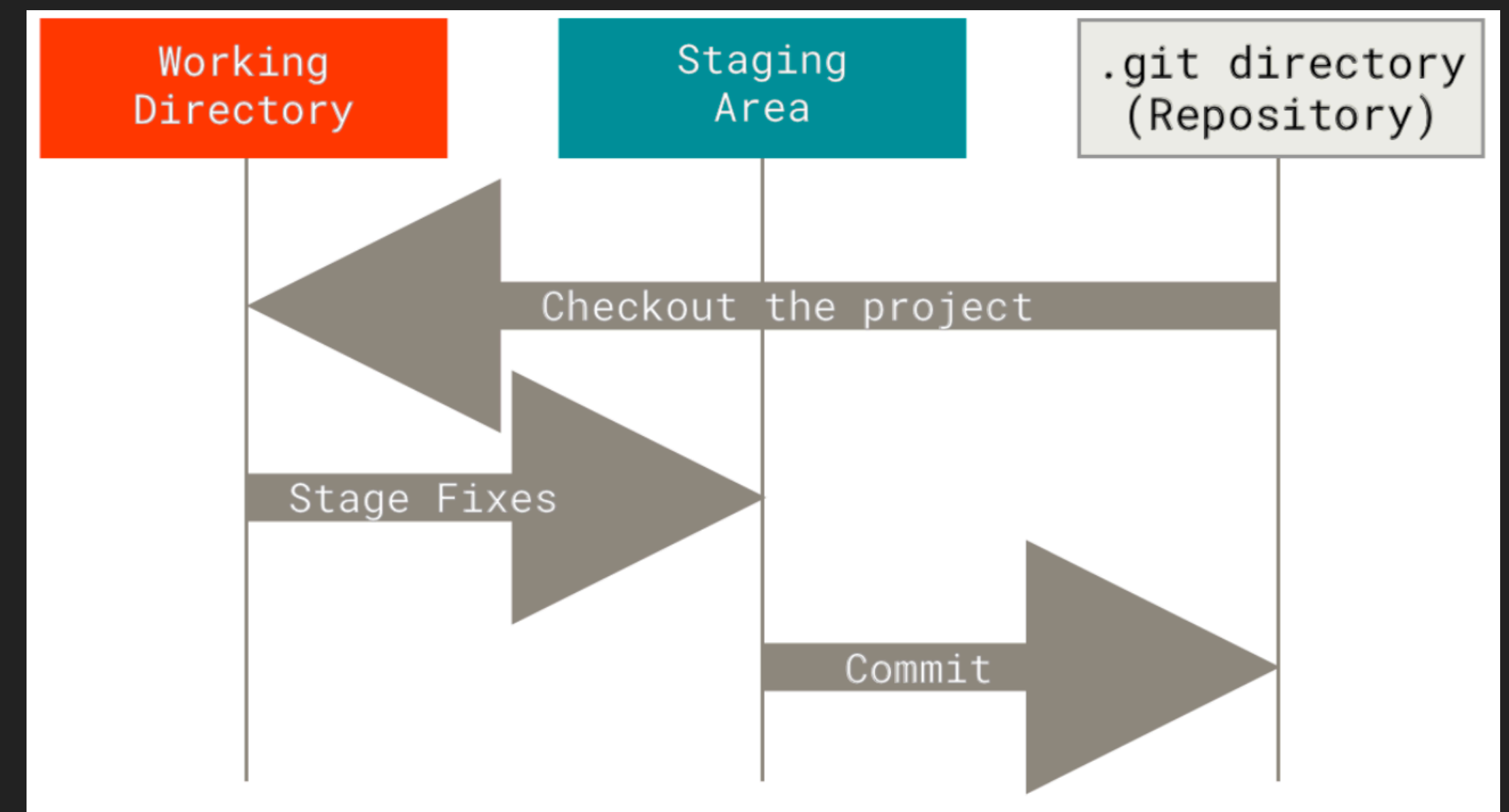
STATES

STATES



STATES

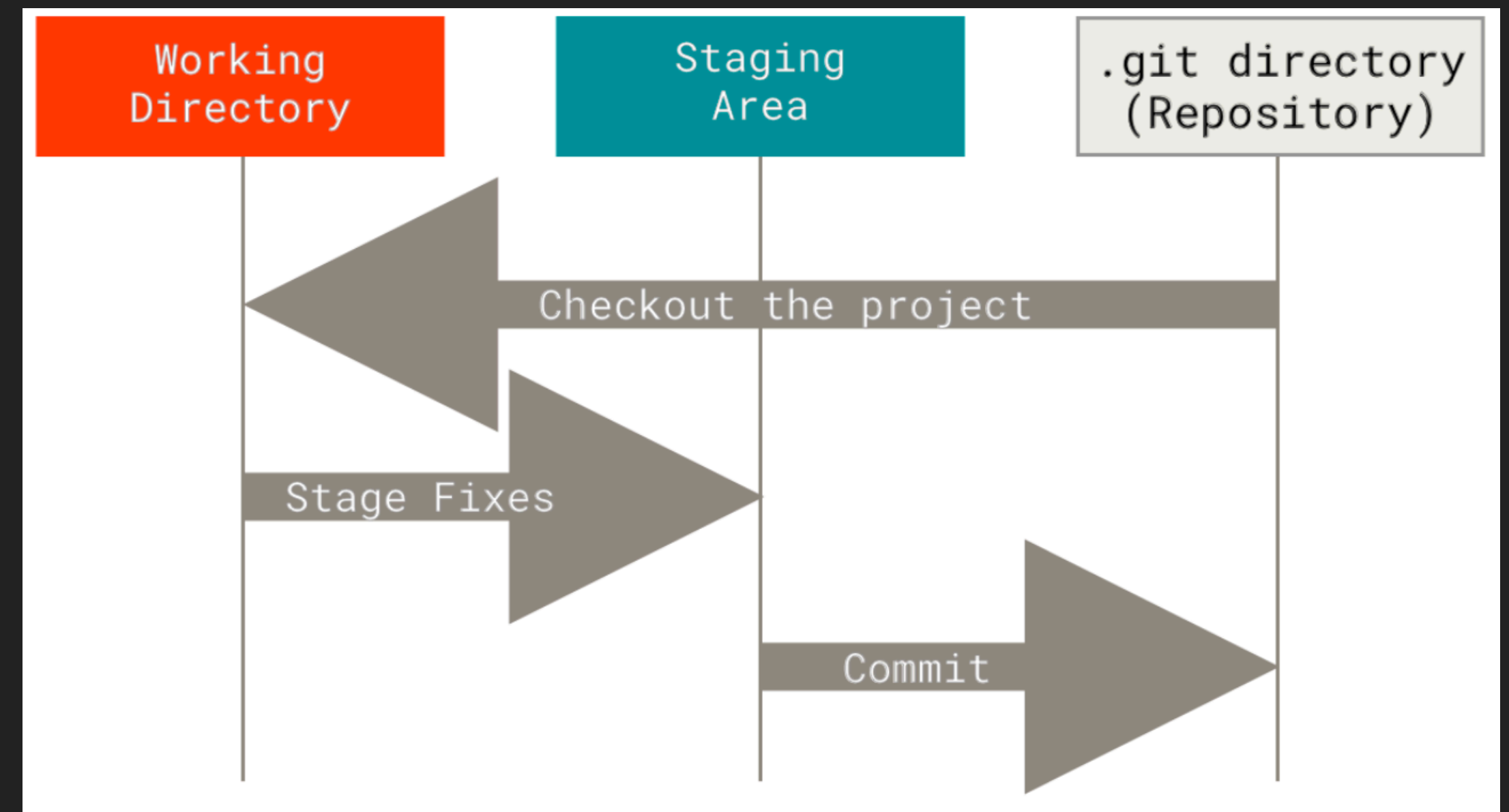
► Modified



STATES

► Modified

► Staged

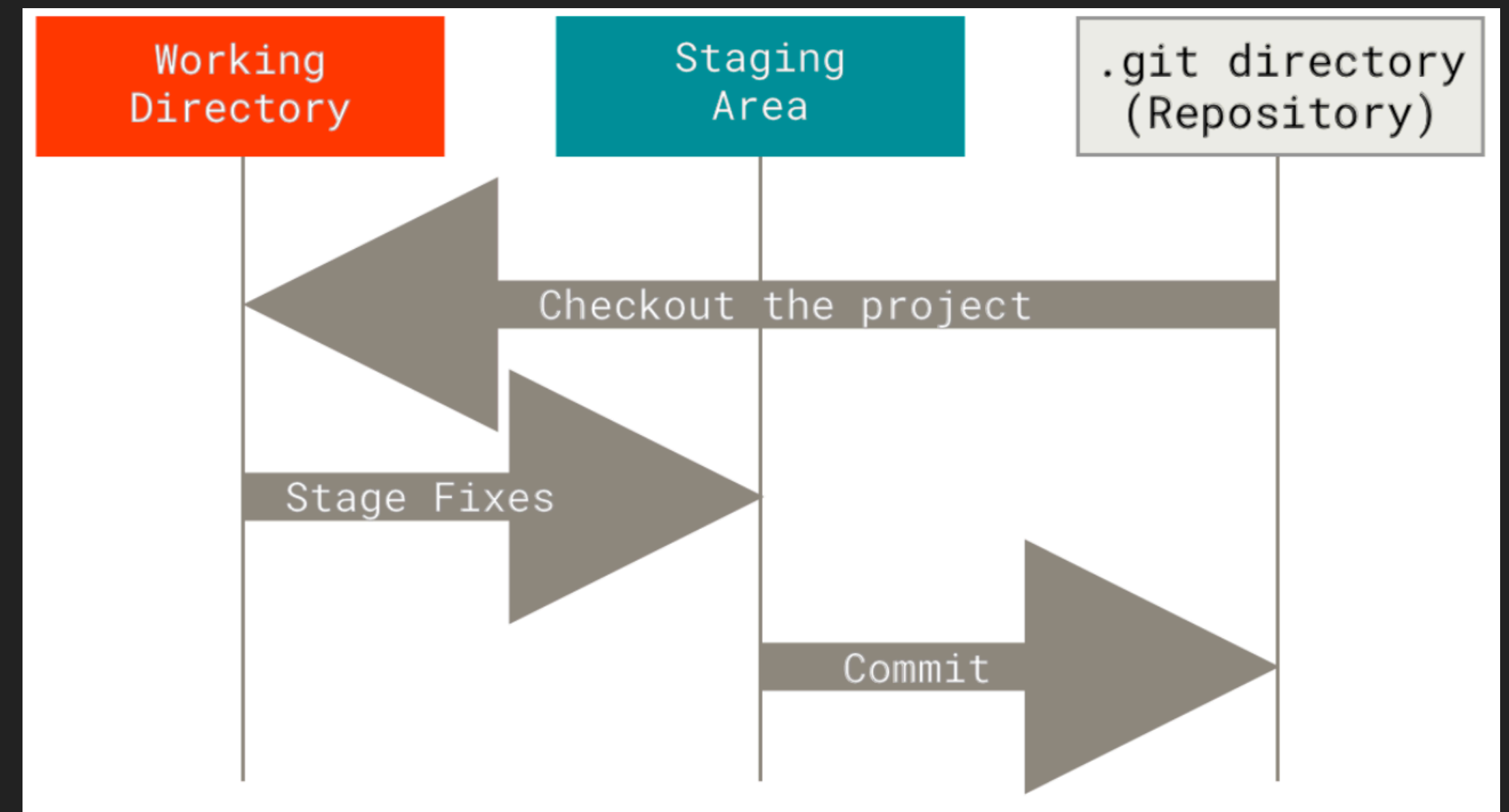


STATES

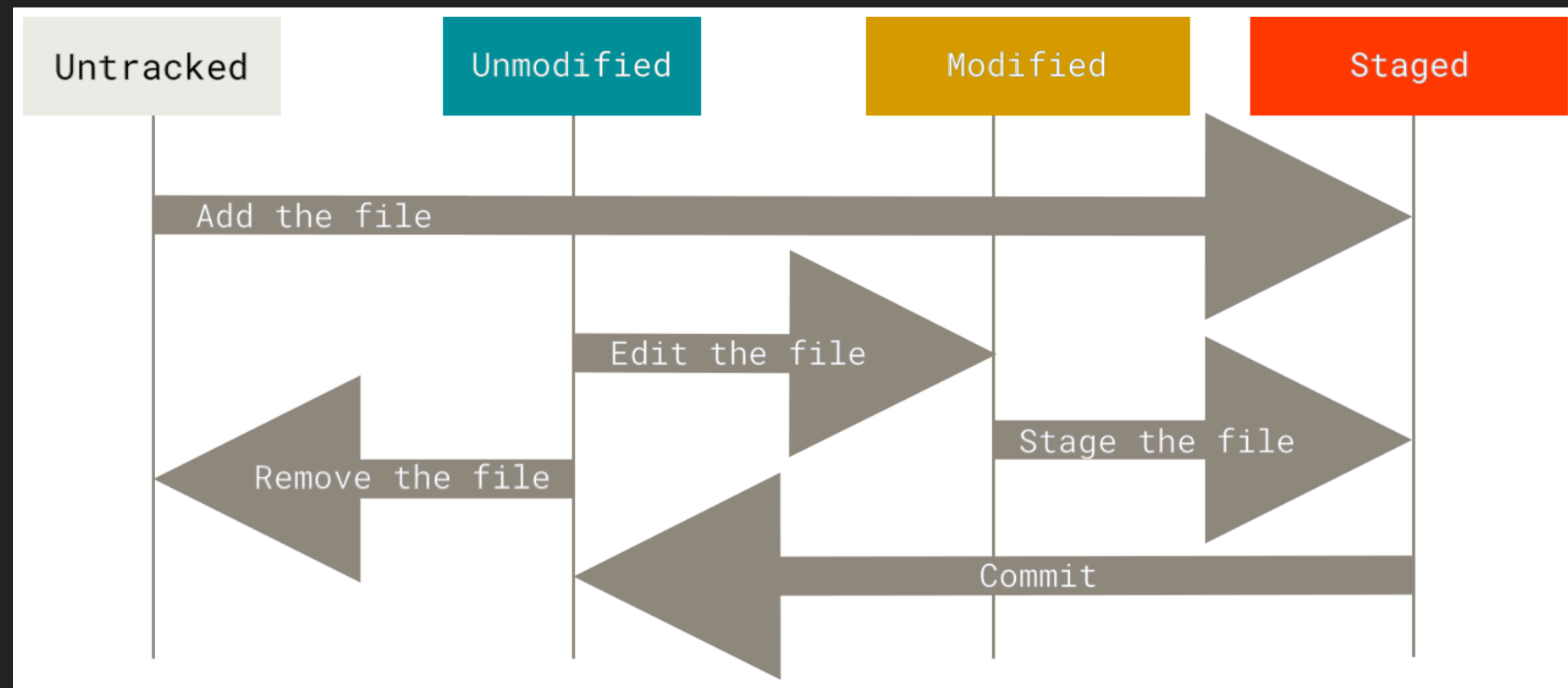
► Modified

► Staged

► Committed



STATES



КОМАНДЫ GIT

КОНФИГУРАЦИЯ

- ▶ `git config --global user.name "John Doe"`
- ▶ `git config --global user.email johndoe@example.com`

КОМАНДЫ GIT

- ▶ `git init`
- ▶ `git add .`
- ▶ `git add LICENSE`
- ▶ `git add /app/main.java`
- ▶ `git commit -m 'Initial project version'`

- ▶ `git clone https://github.com/libgit2/libgit2`
- ▶ `git status`
- ▶ `git diff`
- ▶ `git diff --staged`
- ▶ `git diff --cached`

КОМАНДЫ GIT

- ▶ `git log`
- ▶ `git commit --amend`
- ▶ `git reset HEAD <file>`
- ▶ `git checkout -- <file>`

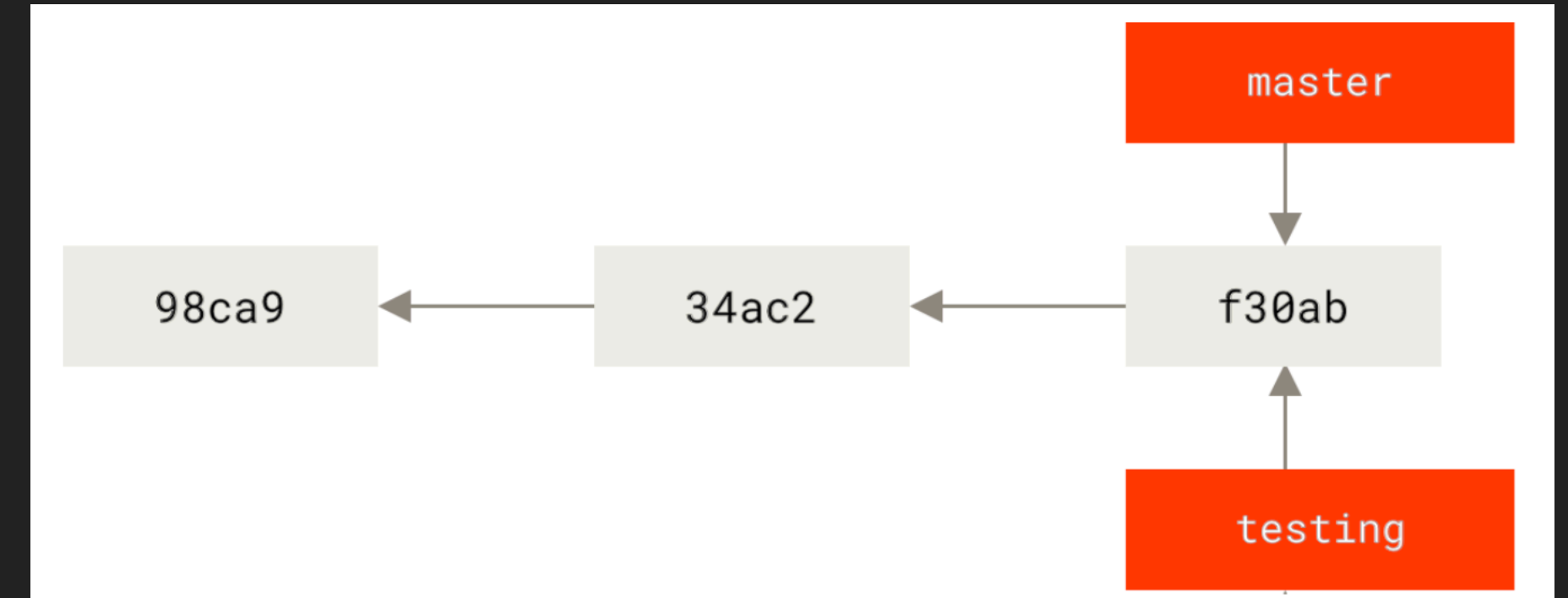
КОМАНДЫ GIT

- ▶ `git remote add <shortname> <url>`
- ▶ `git remote -v`
- ▶ `git fetch <remote>`
- ▶ `git pull`
- ▶ `git push origin master`

КОМАНДЫ GIT

- ▶ `git tag`
- ▶ `git tag -a v1.4 -m "my version 1.4"`

- ▶ git branch testing
- ▶ git checkout testing



BRANCH

BRANCH

- ▶ Pro git , 62 p.

GIT FLOW

Master

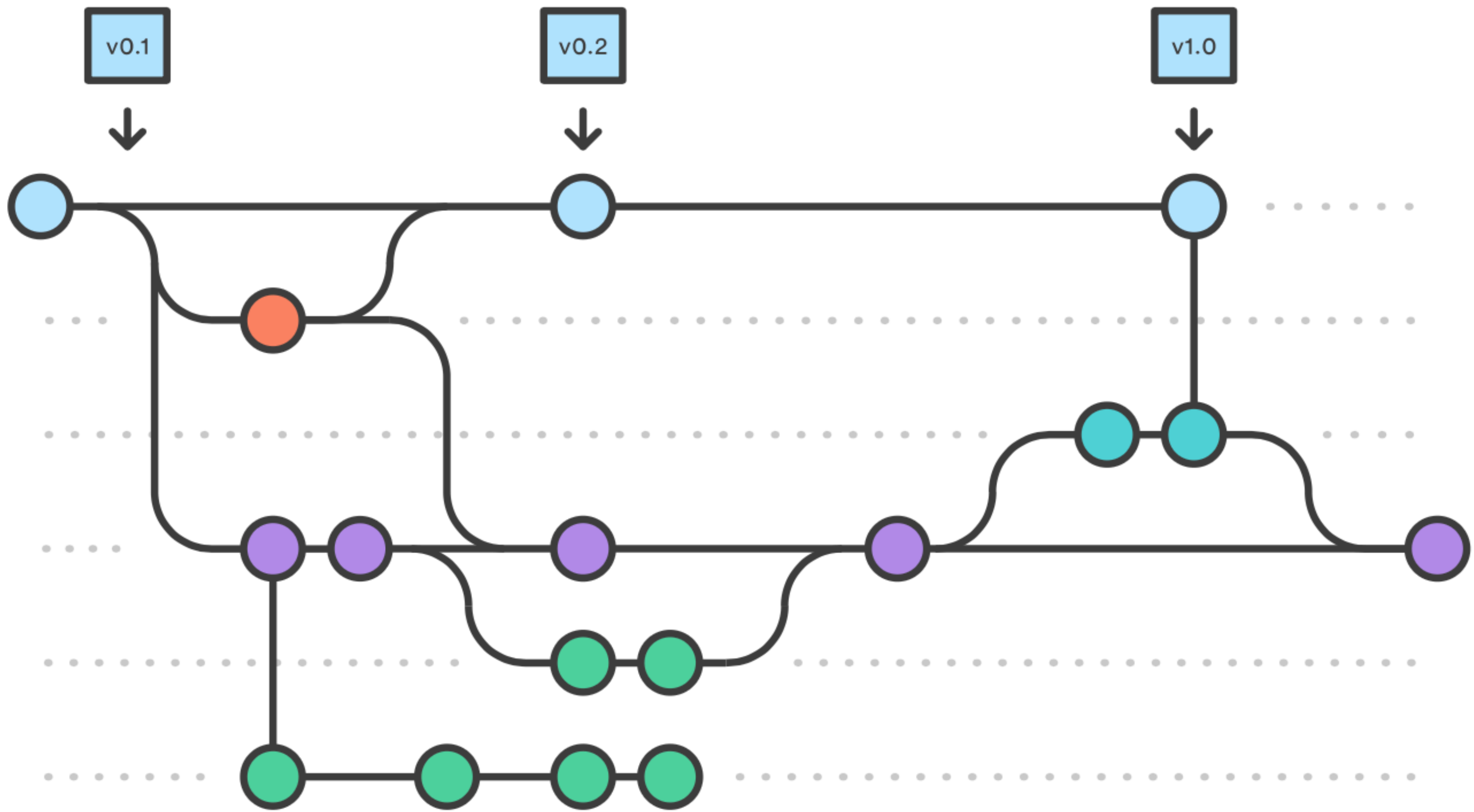
Hotfix

Release

Develop

Feature

Feature



GITHUB FLOW

