

# About SJTUG

- ▶ SJTU \*nix User Group
- ▶ SJTU Joyful Techie User Group
- ▶ Homepage - <https://sjtug.org/>
- ▶ SJTUG Mirrors - <https://mirrors.sjtug.sjtu.edu.cn/>
- ▶ GitHub - <https://github.com/sjtug>





# Git Basic Tutorial

Zhou Fan (@Evensgn), SJTUG

i@evensgn.com

# What is Git

- ▶ Git is a free and open source distributed version control system
- ▶ Then what is GitHub/GitLab/BitBucket?



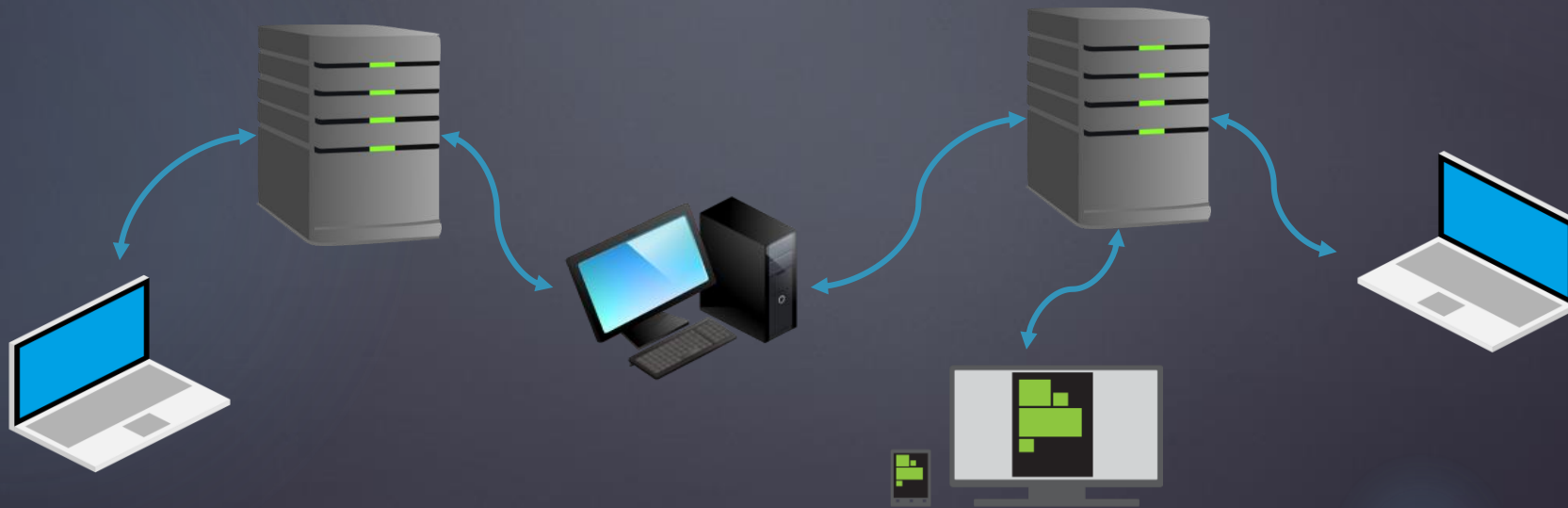
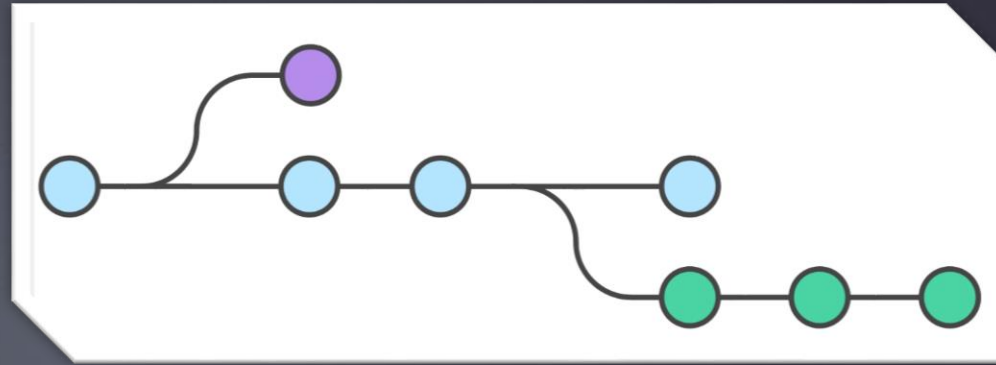
# History of Git

- ▶ Maintenance of Linux kernel project
  - ▶ 1991 – 2002 patches and archived files
  - ▶ 2002 – 2005 BitKeeper (was not open-source)
  - ▶ 2005
    - ▶ BitKeeper's free-of-charge status was revoked
    - ▶ Linus developed Git



# Why Git

- ▶ Great branching model
- ▶ Distributed
- ▶ Small & Fast



# Get Started!

- ▶ Suppose we need to develop a project, say a program to calculate  $A + B$
- ▶ Fix bugs / Add new features
- ▶ Collaborating with others
- ▶ We have Git installed
  - ▶ <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

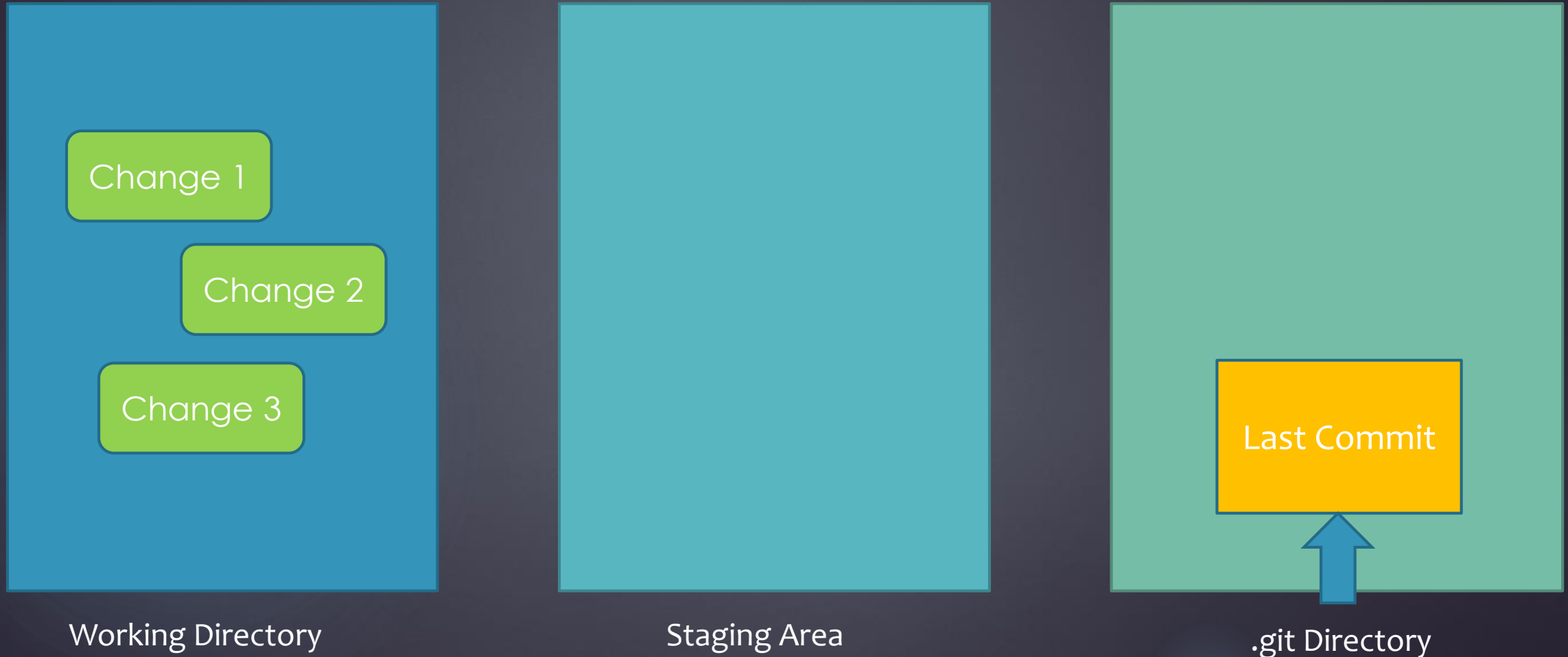
# Getting a Git Repository

- ▶ Turn a local directory into a git repository
  - ▶ git init
- ▶ Clone a remote git repository
  - ▶ git clone



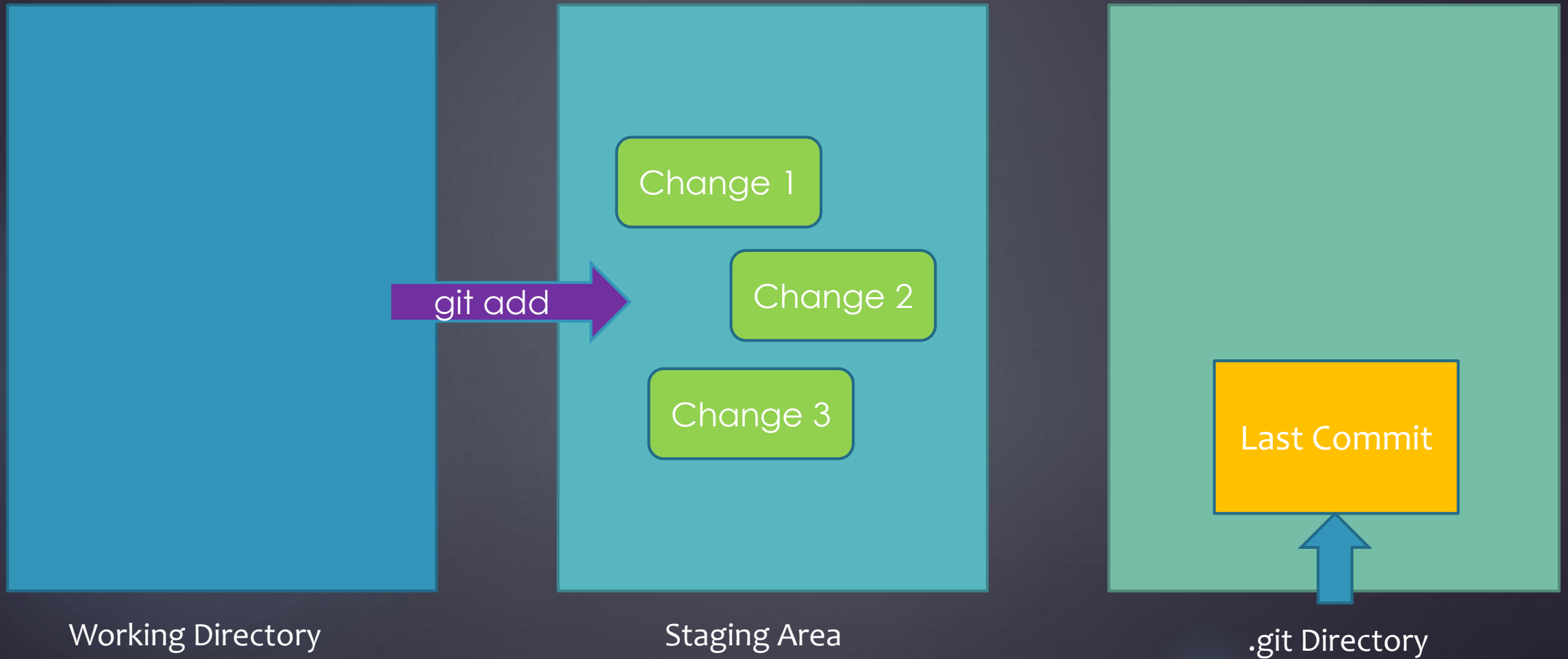
```
$ mkdir aplusb  
$ cd aplusb  
$ git init  
Initialized empty Git repository in /home/evensgn/Code/apusb/.git/
```

# Staging Area – Make Some Changes

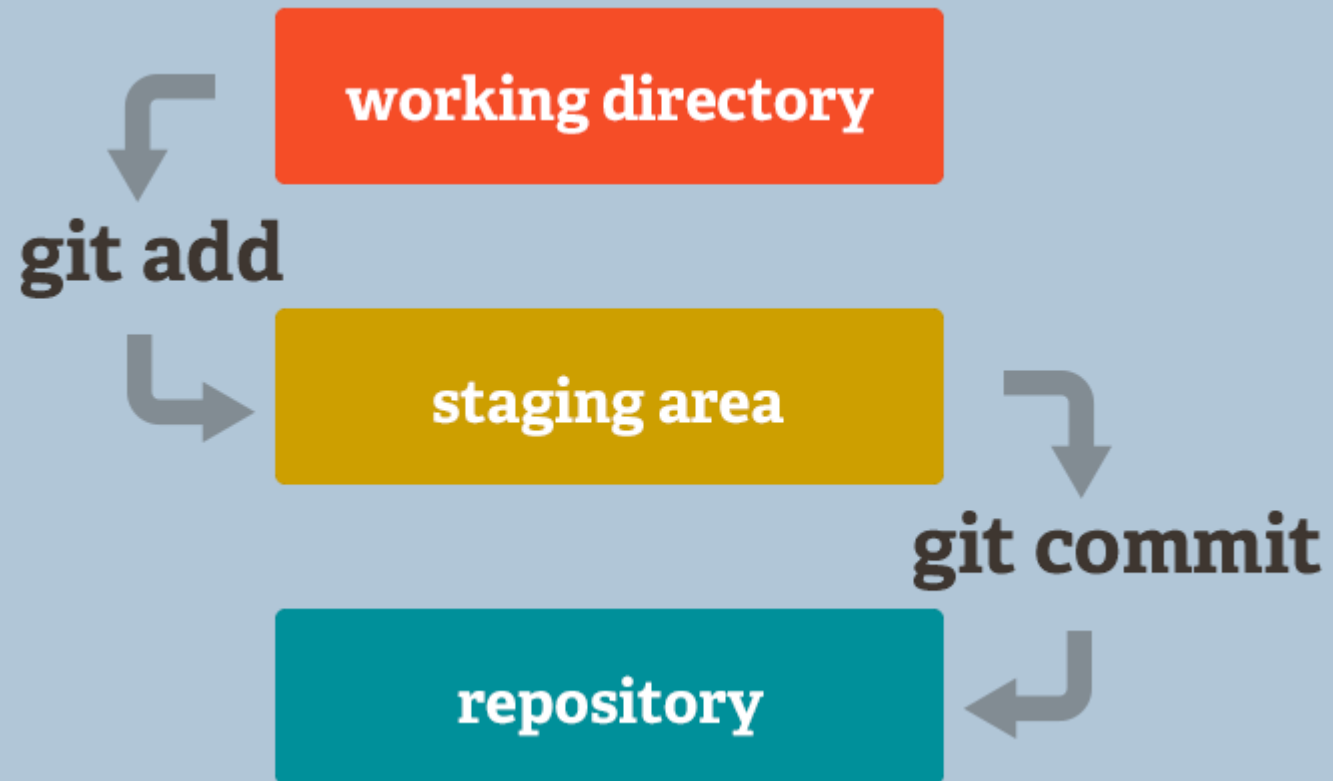




# Staging Area - Add



# Staging Area



# Staging changes

- ▶ `git status`
  - ▶ Check the status of the working directory
- ▶ `git add`
  - ▶ `git add <file>`
  - ▶ Stage change of <file> so that it would be included in the next commit

```
$ echo "Hello, world!" > readme.md
$ git status
On branch master

No commits yet

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

    readme.md

nothing added to commit but untracked files present (use "git add" to track)
$ git add readme.md
$ git status
On branch master


No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   readme.md
```

# Commit changes

- ▶ `git commit`
  - ▶ Confirm changes in the staging area
  - ▶ Make a snapshot of the working directory (ignore changes not added)



```
$ git commit -m "Add readme"
[master (root-commit) a911362] Add readme
1 file changed, 1 insertion(+)
create mode 100644 readme.md
```

# Commit changes

- ▶ You may have to tell git who you are at the first time



```
$ git commit -m "Add readme"
```

```
*** Please tell me who you are.
```

```
Run
```

```
git config --global user.email "you@example.com"
```


```
git config --global user.name "Your Name"
```

```
to set your account's default identity.
```

```
Omit --global to set the identity only in this repository.
```

```
fatal: unable to auto-detect email address (got 'evensgn@ubuntu.(none)')
```

# git log



```
$ git log
commit ea99e6005929d1c7c78a65a6794fe32a8f55c8d2 (HEAD -> master)
Author: Evensgn <i@evensgn.com>
Date:   Tue Jul 3 15:30:10 2018 +0800

    Delete hello world

commit 13d0485ce1ee342e468f941fc62f4dc4921be634
Author: Evensgn <i@evensgn.com>
Date:   Tue Jul 3 15:29:44 2018 +0800

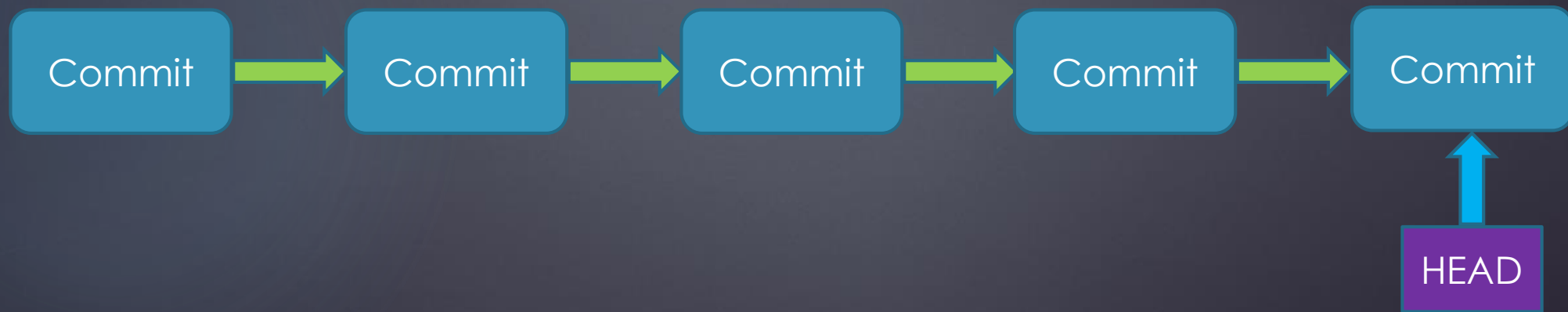
    Add SJTUG intro

commit a9113622642fa9ace5db08f8436ec96d4fccccd4
Author: Evensgn <i@evensgn.com>
Date:   Tue Jul 3 15:20:11 2018 +0800

    Add readme
```

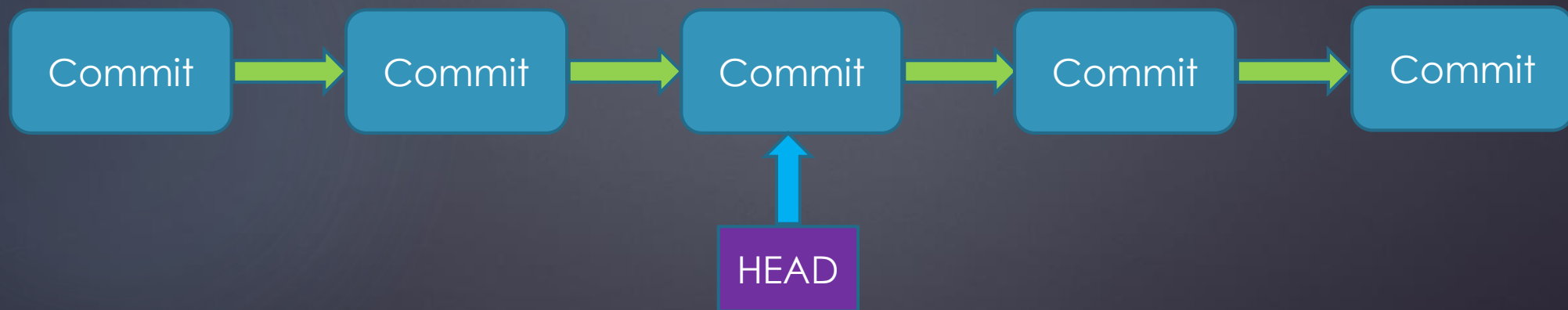
# git reset

- ▶ Reset current HEAD to the specified commit
  - ▶ `git reset --soft <commit>`: leaves all your changed files "Changes to be committed"
  - ▶ `git reset --hard <commit>` discard any changes to tracked files since <commit>



# git reset

- ▶ Reset current HEAD to the specified commit
  - ▶ `git reset --soft <commit>`: leaves all your changed files "Changes to be committed"
  - ▶ `git reset --hard <commit>` discard any changes to tracked files since <commit>






# Undoing things

- ▶ Unstaging a staged file

- ▶ `git reset HEAD <file>`



```
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   readme.md

$ git reset HEAD readme.md
Unstaged changes after reset:
M   readme.md
```

# Undoing things

- ▶ Unmodifying a modified file

- ▶ `git checkout -- <file>`




```
$ git status
On branch master
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:   readme.md

no changes added to commit (use "git add" and/or "git commit -a")
$ git checkout -- readme.md
```

# Remote repositories

- ▶ `git remote add`
  - ▶ Add a remote repository to be tracked



```
$ git remote
origin
$ git remote add github git@github.com:Evensgn/aplusb.git
$ git remote
github
origin
```


# git pull

- ▶ `git pull <repository> <branch>`
- ▶ Incorporates changes from a remote repository into the current branch

```
$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:Evensgn/aplusb
 * branch                master      -> FETCH_HEAD
    b18a6d6..5be180b      master      -> origin/master
Updating b18a6d6..5be180b
Fast-forward
 README.md | 2 ++
 1 file changed, 2 insertions(+)
```

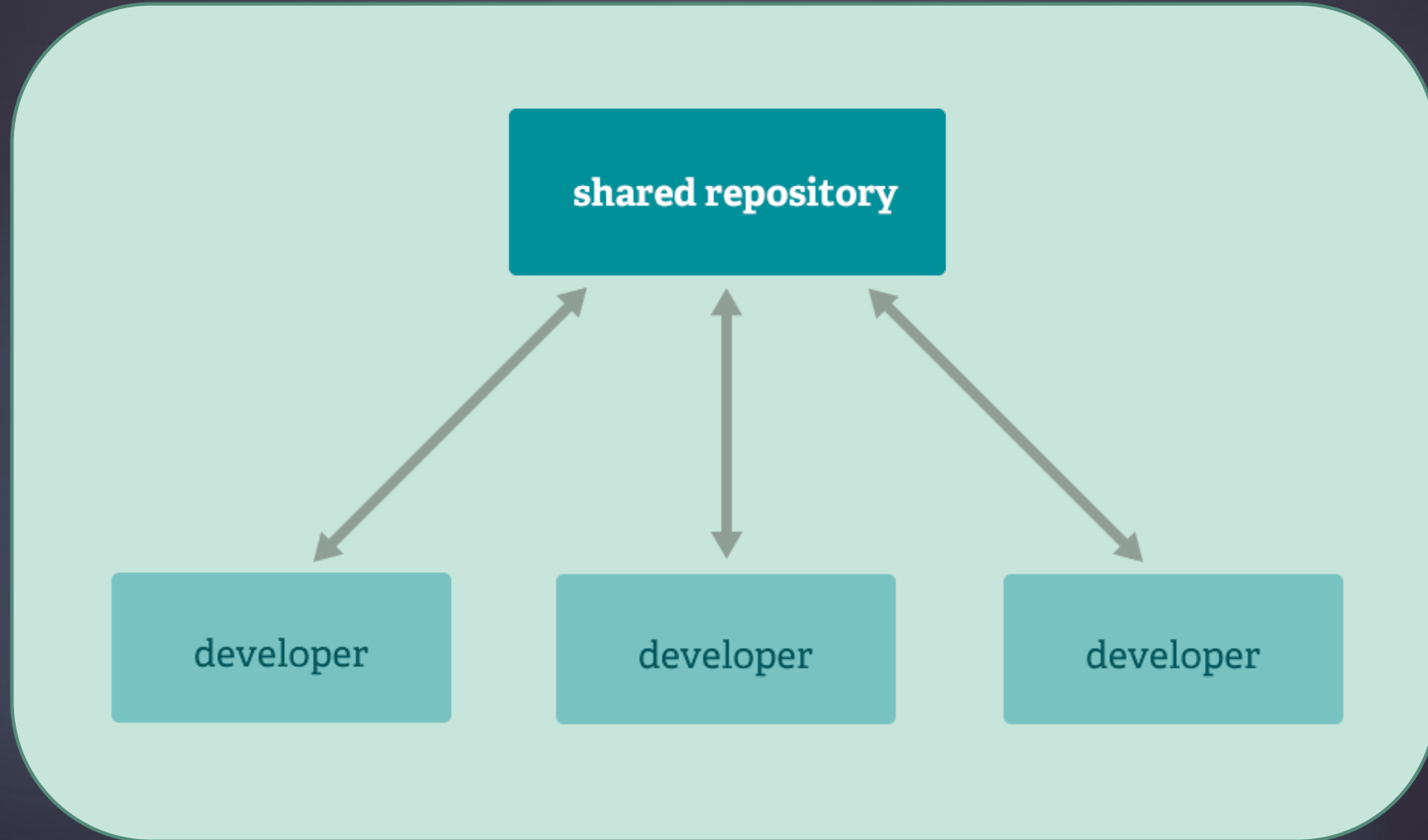
# git pull

- ▶ `git pull <repository> <branch>`
- ▶ Incorporates changes from a remote repository into the current branch



```
      A---B---C origin/master
      /           \
D---E---F---G---H master
```

# Collaborate with others




# git push

- ▶ `git push <repository> <branch>`
- ▶ Push local changes the remote repository

```
$ git push origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 274 bytes | 274.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Evensgn/aplusb.git
    5be180b..f78a356  master -> master
```

# Oops! Merge conflicts



```
$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:Evensgn/aplusb
 * branch                master      -> FETCH_HEAD
   f78a356..6da28b5      master      -> origin/master
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```



# Resolving merge conflicts



```
# aplusb
```

```
A program to compute a + b.
```

```
This repository is for SJTUG 2018 summer course.
```

```
<<<<<< HEAD
```

```
Hello, world! Today is July 3.
```


```
=====
```

```
Hello, World! Today is July 4.
```

```
>>>>>> 6da28b553a09b2c1da7182c39044ecab10742ccd
```

# Branching

- ▶ `git branch <new branch>`
  - ▶ Create a new branch
- ▶ `git checkout <branch>`
  - ▶ Switch to <branch>
- ▶ `git branch -d <branch>`
  - ▶ Delete <branch>



```
$ git branch
* master
$ git branch dev
$ git branch
dev
* master
$ git checkout dev
Switched to branch 'dev'
```

# git stash

- ▶ Stash the changes in a dirty working directory away


- ▶ git stash
- ▶ git stash list
- ▶ git stash apply
- ▶ git stash clear



```
$ git checkout master
error: Your local changes to the following files would be overwritten by checkout:
    README.md
Please commit your changes or stash them before you switch branches.
Aborting
$ git stash
Saved working directory and index state WIP on dev: 3655382 Add something
$ git status
On branch dev
nothing to commit, working tree clean
```

# git merge

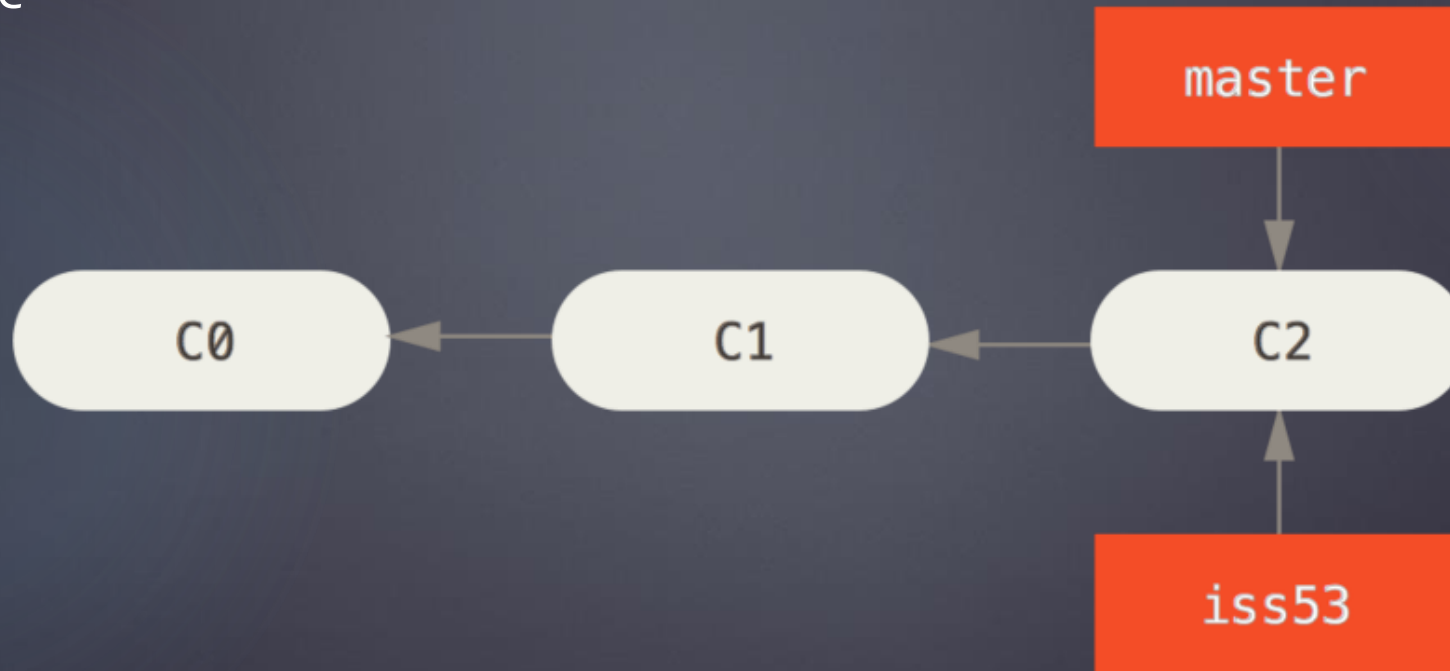
- ▶ `git merge <branch>`
  - ▶ Merge <branch> into the current branch



```
$ git checkout master
Switched to branch 'master'
$ git merge iss53
Merge made by the 'recursive' strategy.
index.html |      1 +
1 file changed, 1 insertion(+)
```

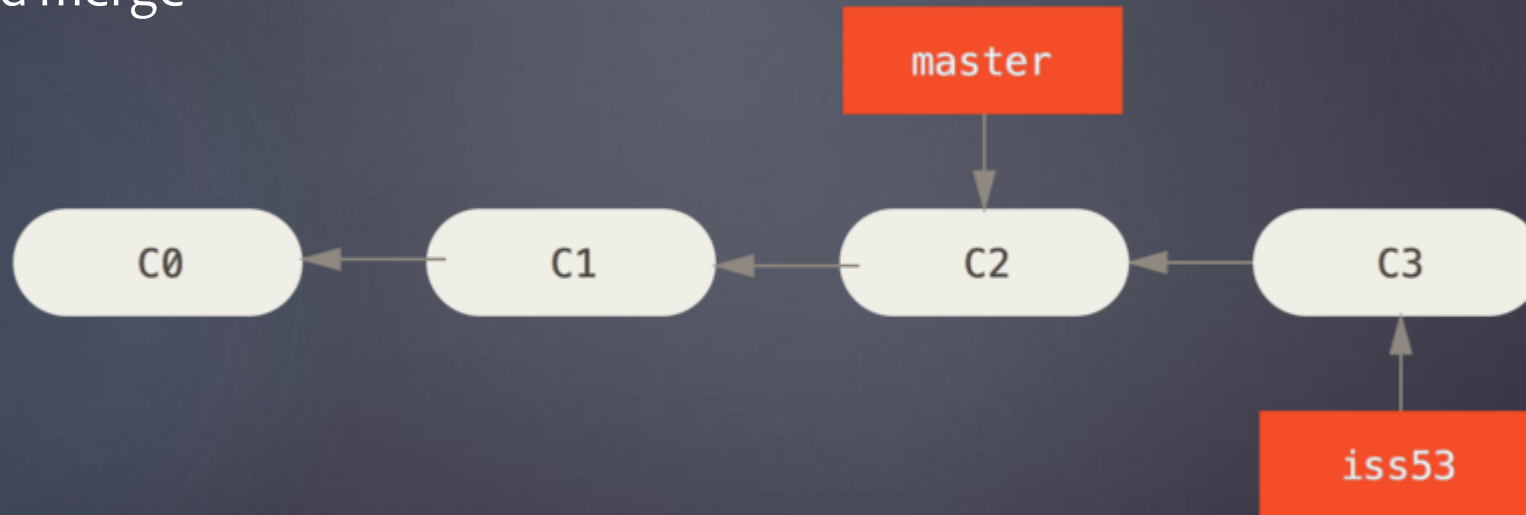
# git merge

- ▶ `git merge <branch>`
  - ▶ Merge <branch> into the current branch
- ▶ A simple case



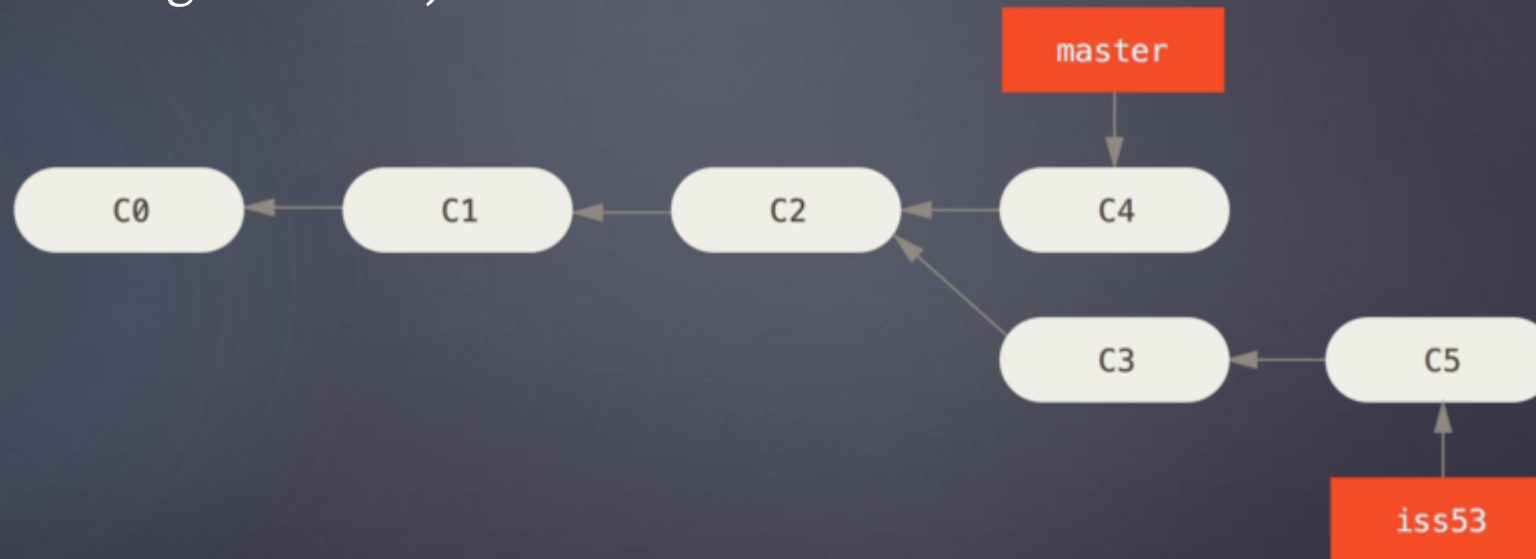
# git merge

- ▶ `git merge <branch>`
  - ▶ Merge <branch> into the current branch
- ▶ A simple case
- ▶ fast-forward merge



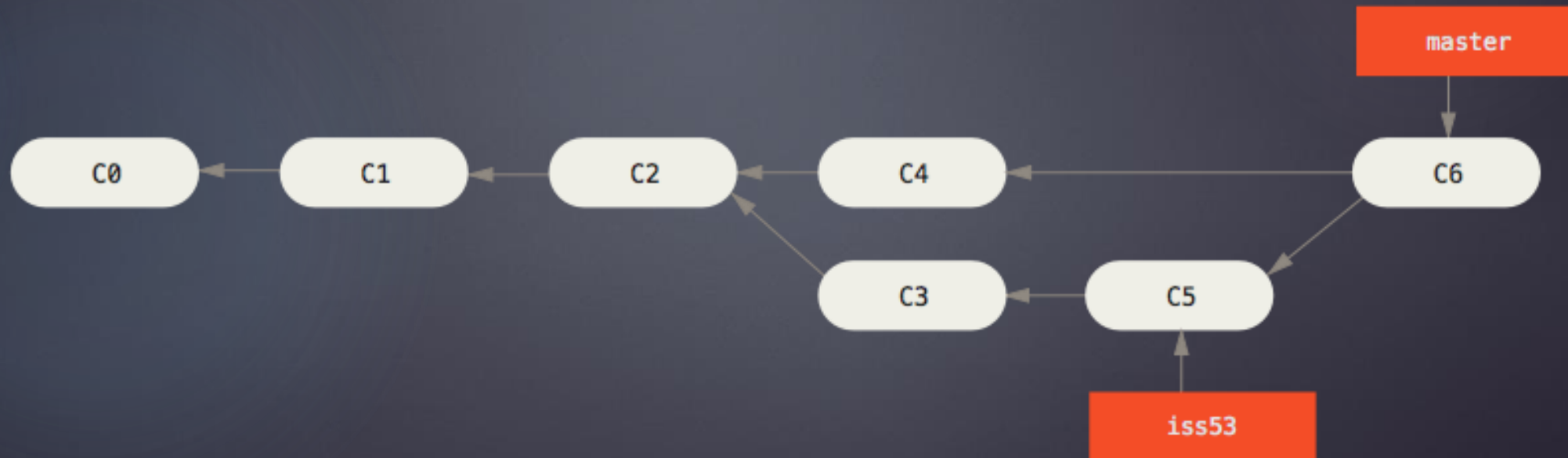
# git merge

- ▶ `git merge <branch>`
  - ▶ Merge <branch> into the current branch
- ▶ A not-that-simple case
- ▶ (There may be merge conflicts)



# git merge

- ▶ `git merge <branch>`
  - ▶ Merge <branch> into the current branch
- ▶ A not-that-simple case



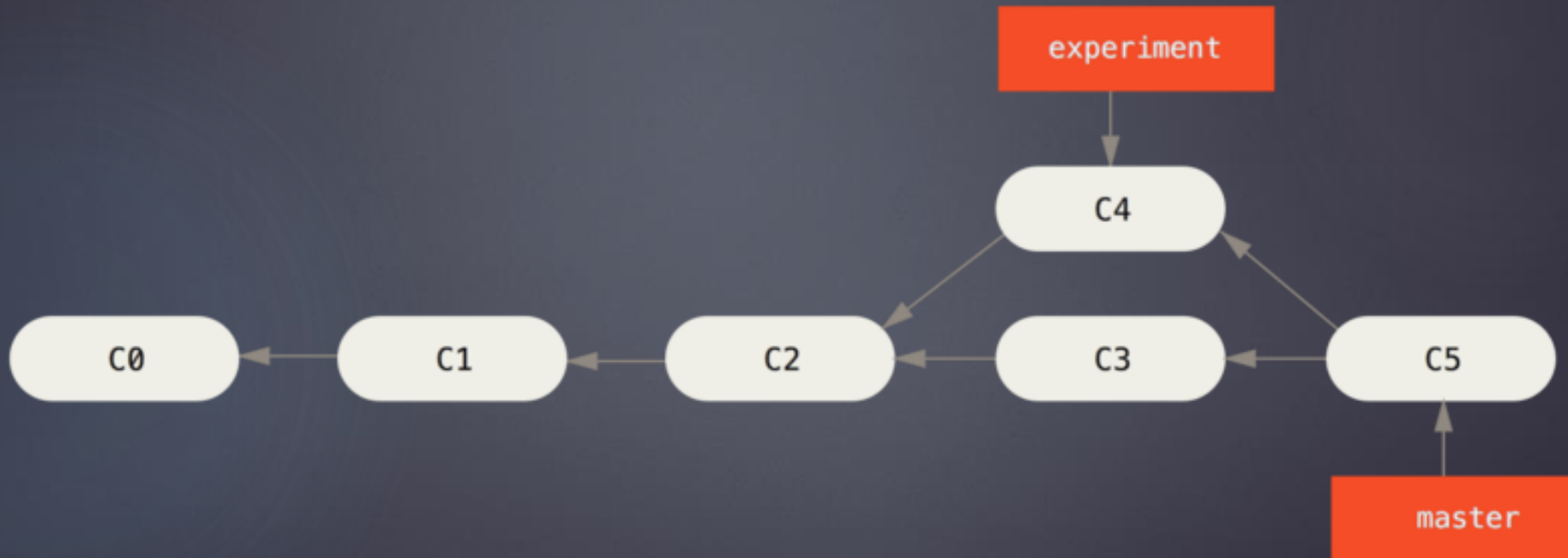


# Git workflow



# git rebase

- Review: branch merging

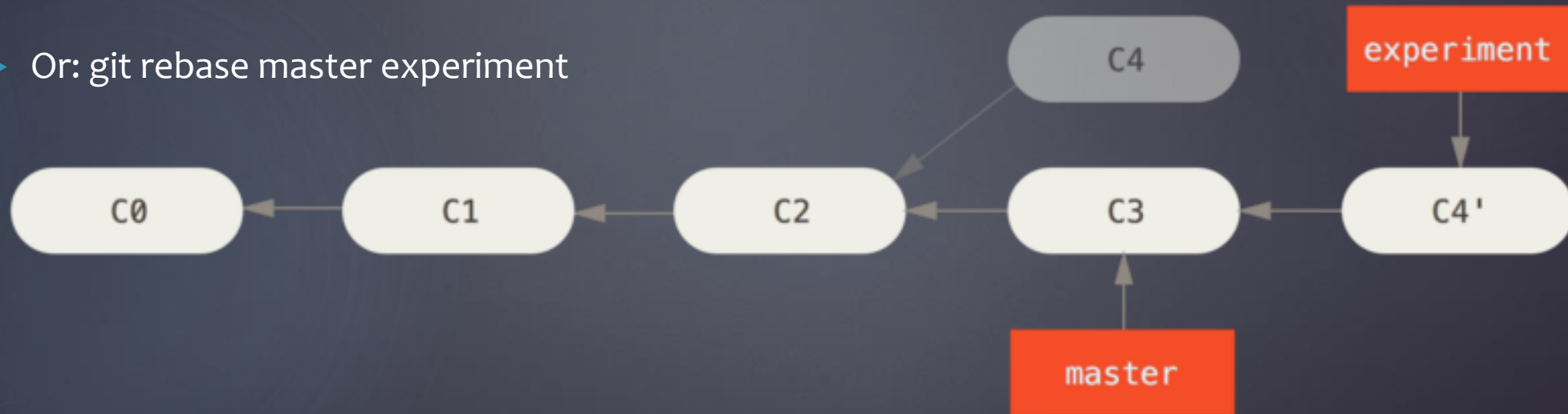


# git rebase

▶ Another way to do that: replay changes of experiment to master

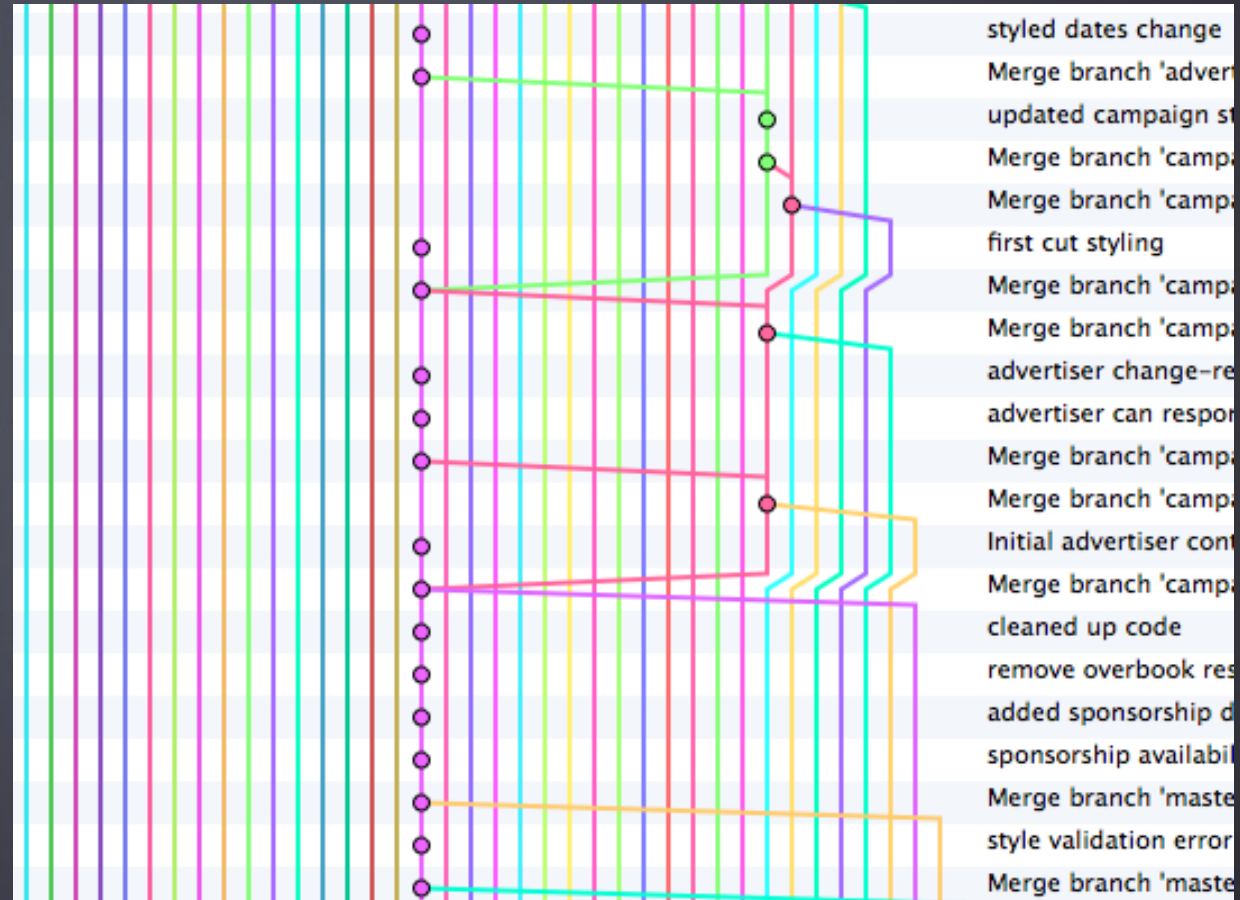
- ▶ git checkout experiment
- ▶ git rebase master

▶ Or: git rebase master experiment



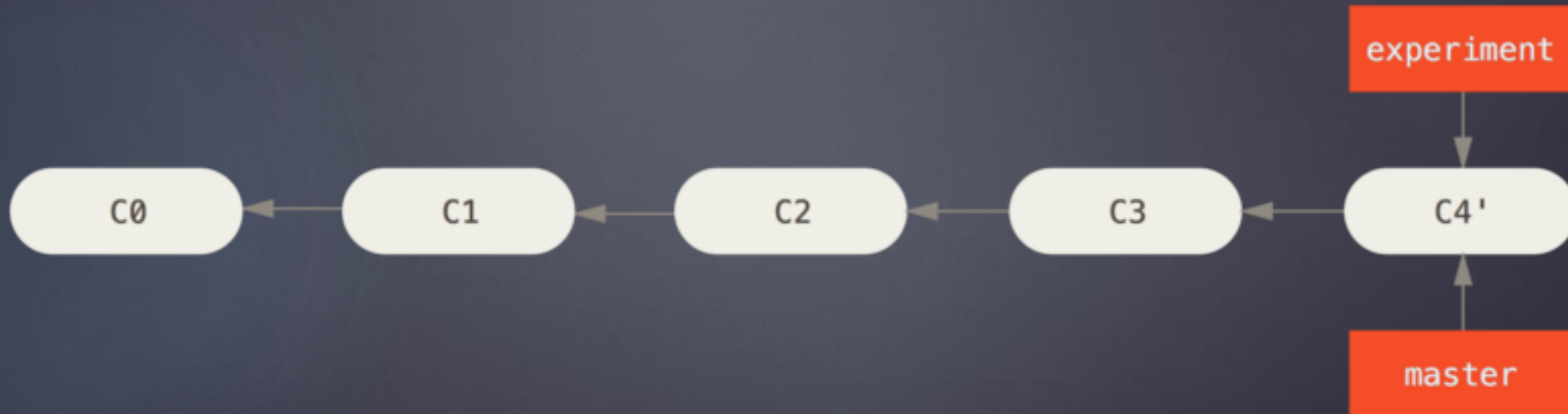
# git rebase

- ▶ With merging ...



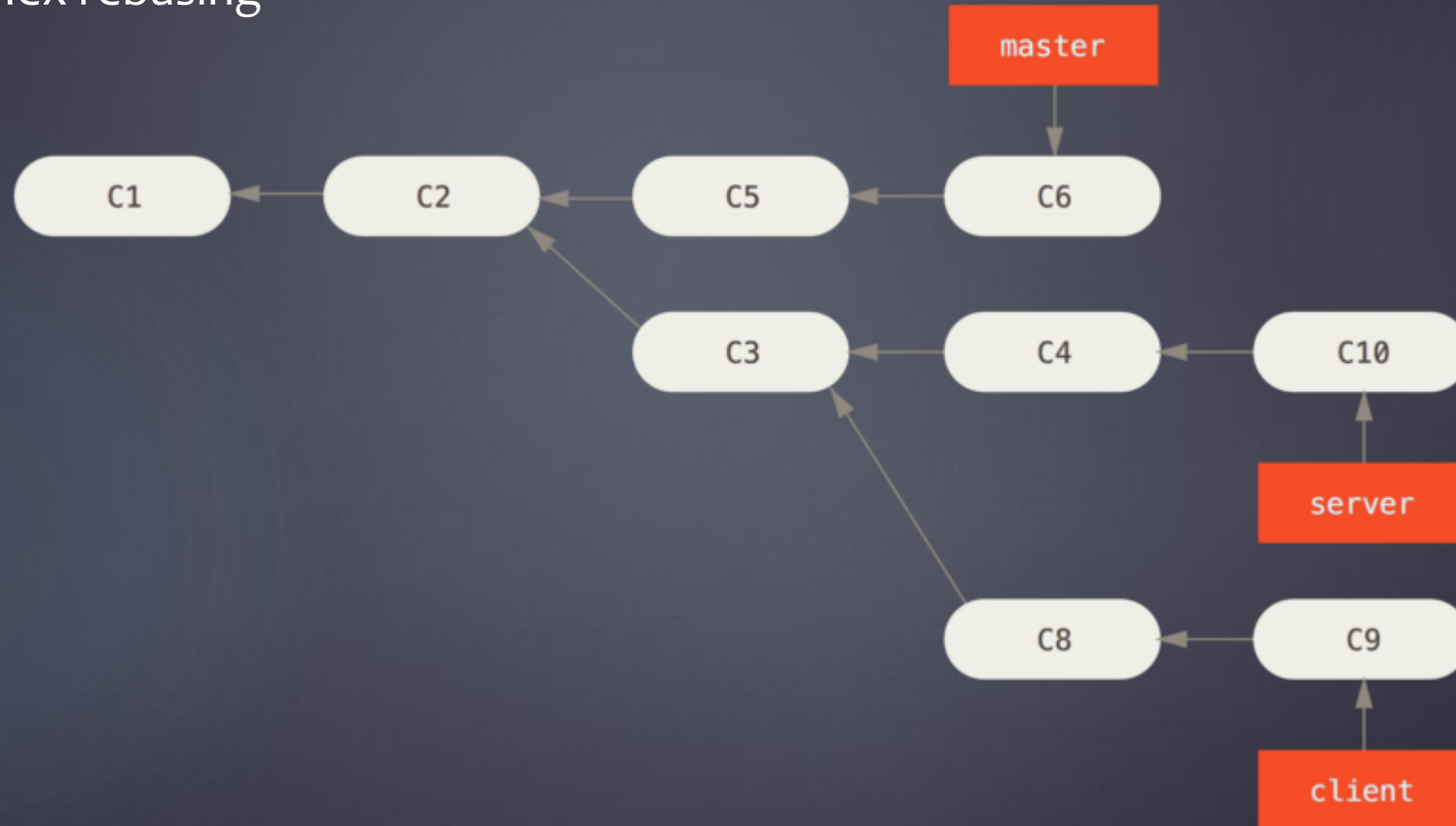
# git rebase

- ▶ Fast-forward merge
  - ▶ git checkout master
  - ▶ git merge experiment



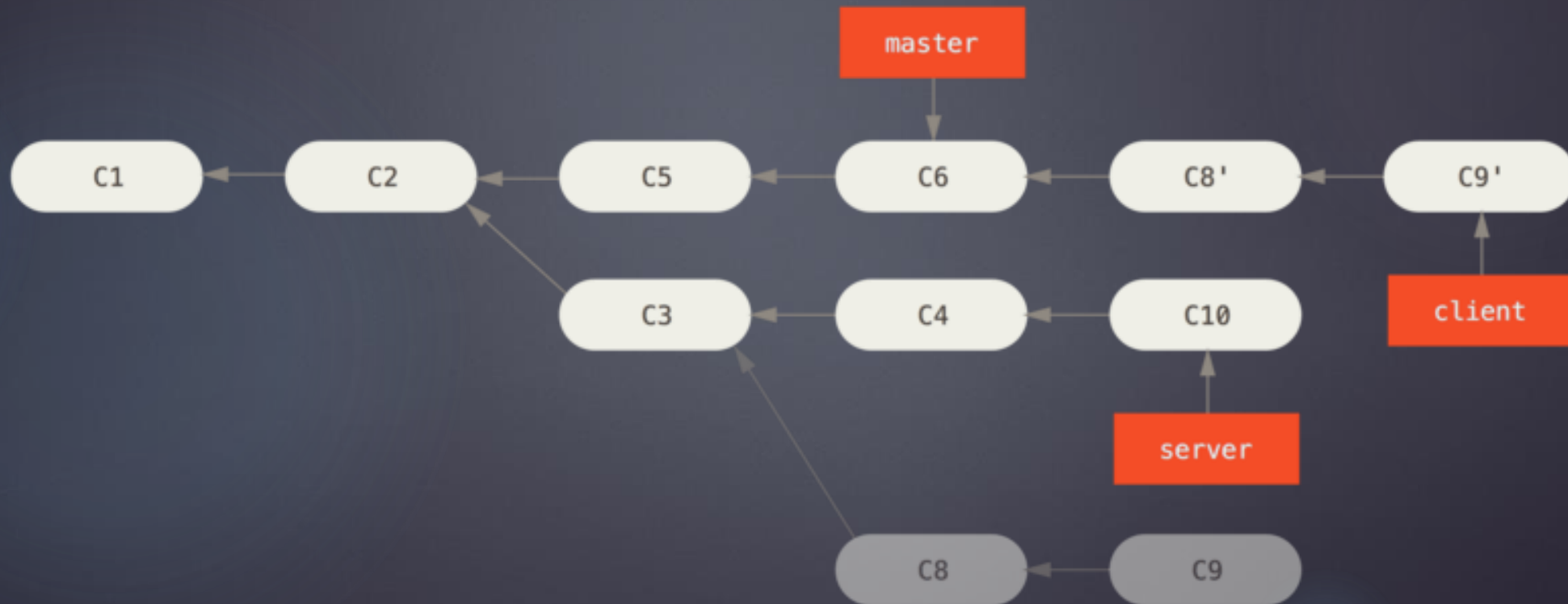
# git rebase

## ► More complex rebasing



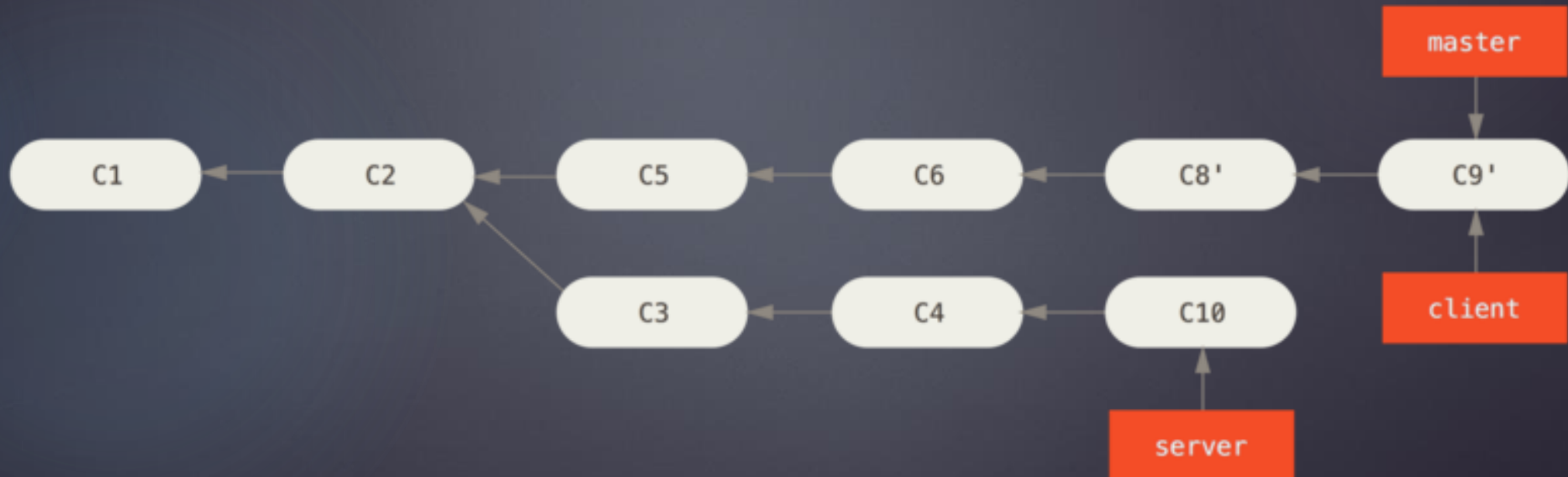
# git rebase

- ▶ More complex rebasing
  - ▶ git rebase -onto master server client



# git rebase

- ▶ More complex rebasing
  - ▶ git checkout master
  - ▶ git merge client (Fast-forward merge)





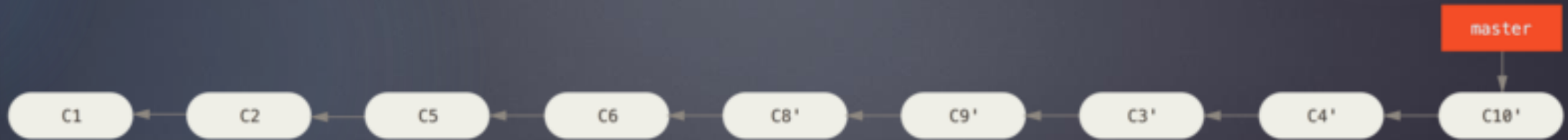
# git rebase

- ▶ More complex rebasing
  - ▶ git rebase master server



# git rebase

- ▶ More complex rebasing
  - ▶ git checkout master
  - ▶ git merge server (Fast-forward merge)



# git rebase

- ▶ A way to clean up your story
  - ▶ rebase local changes you've made before you share them
  - ▶ then you push them

# Reference

- ▶ Git Reference - <https://git-scm.com/docs>
- ▶ Pro Git Book - <https://git-scm.com/book/en/v2>





# Thanks