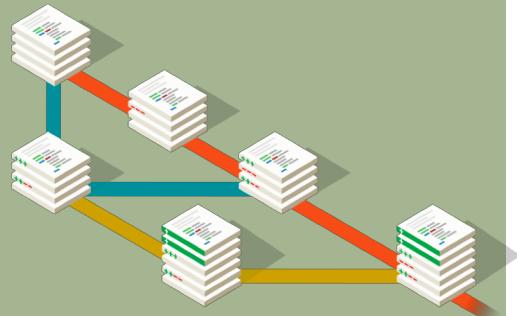


<https://git-scm.com/book/>

Meet the Git

MEETPY 26.02

GRIGORII V. SMIRNOV-PINCHUKOV



WHAT ISSUES ARE ADDRESSED

- How to efficiently save the previous (or stable) steps of the project
- How to spread the development of a project to the group
- How to distribute the project between people and computers
- How to improve the programming skills

TAKE AWAY

- Learn first with the command-line interface
- GUI is fine when you understand what is going on
- GUI for merges is fine (meld)

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: hw.sh

Untracked files:

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

out

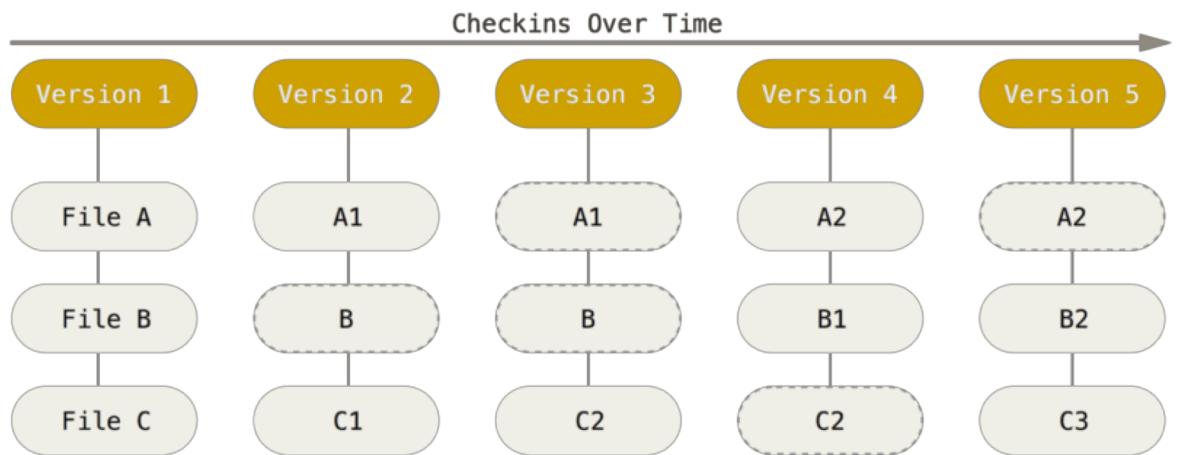
no changes added to commit (use "git add" and/or "git commit -a")



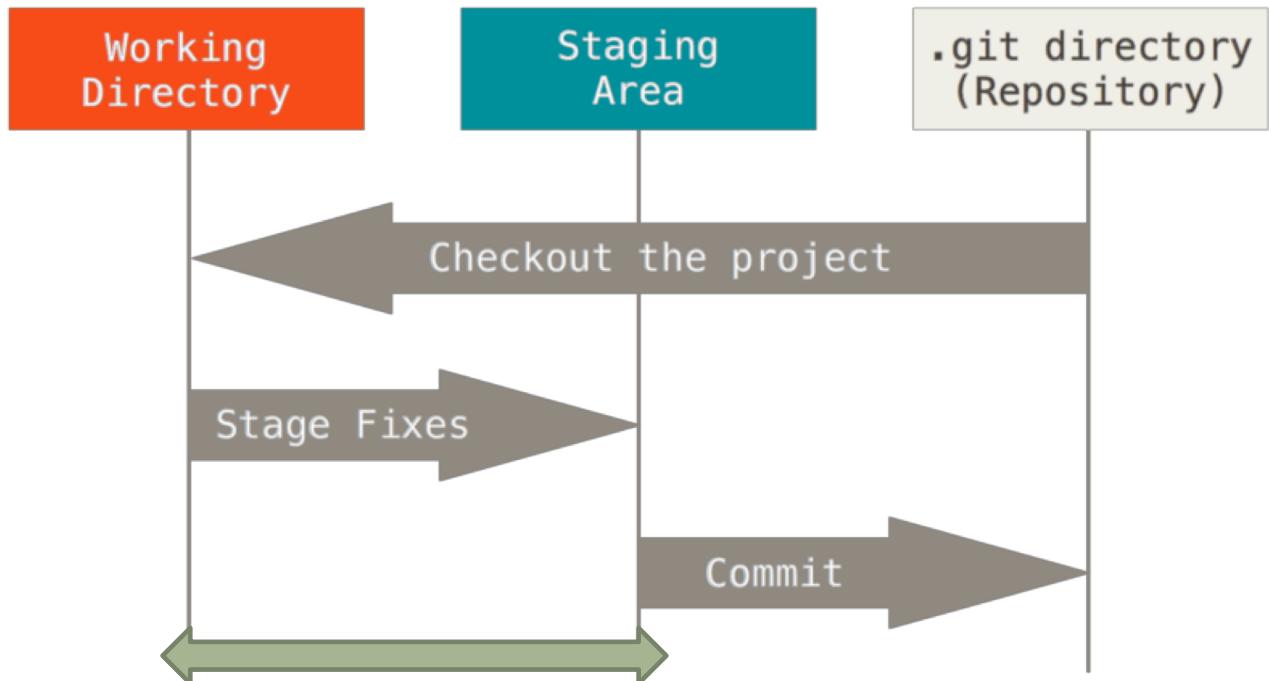
GIT WORKFLOW – LOCAL

COMMITS AND BRANCHES

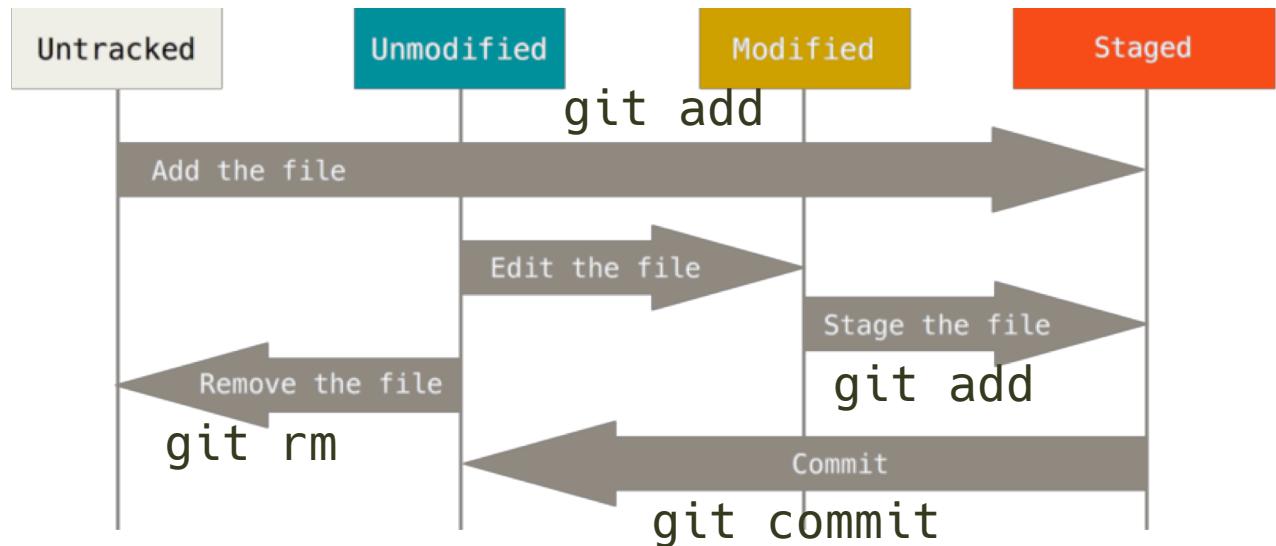
SAVING THE STEPS



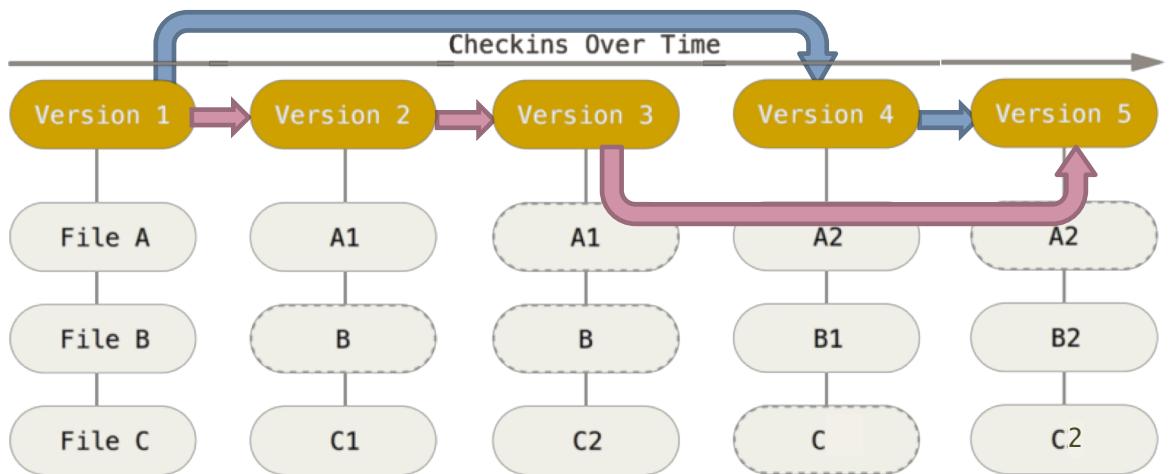
GIT WORKFLOW



GIT WORKFLOW



MERGE



TAKE AWAY

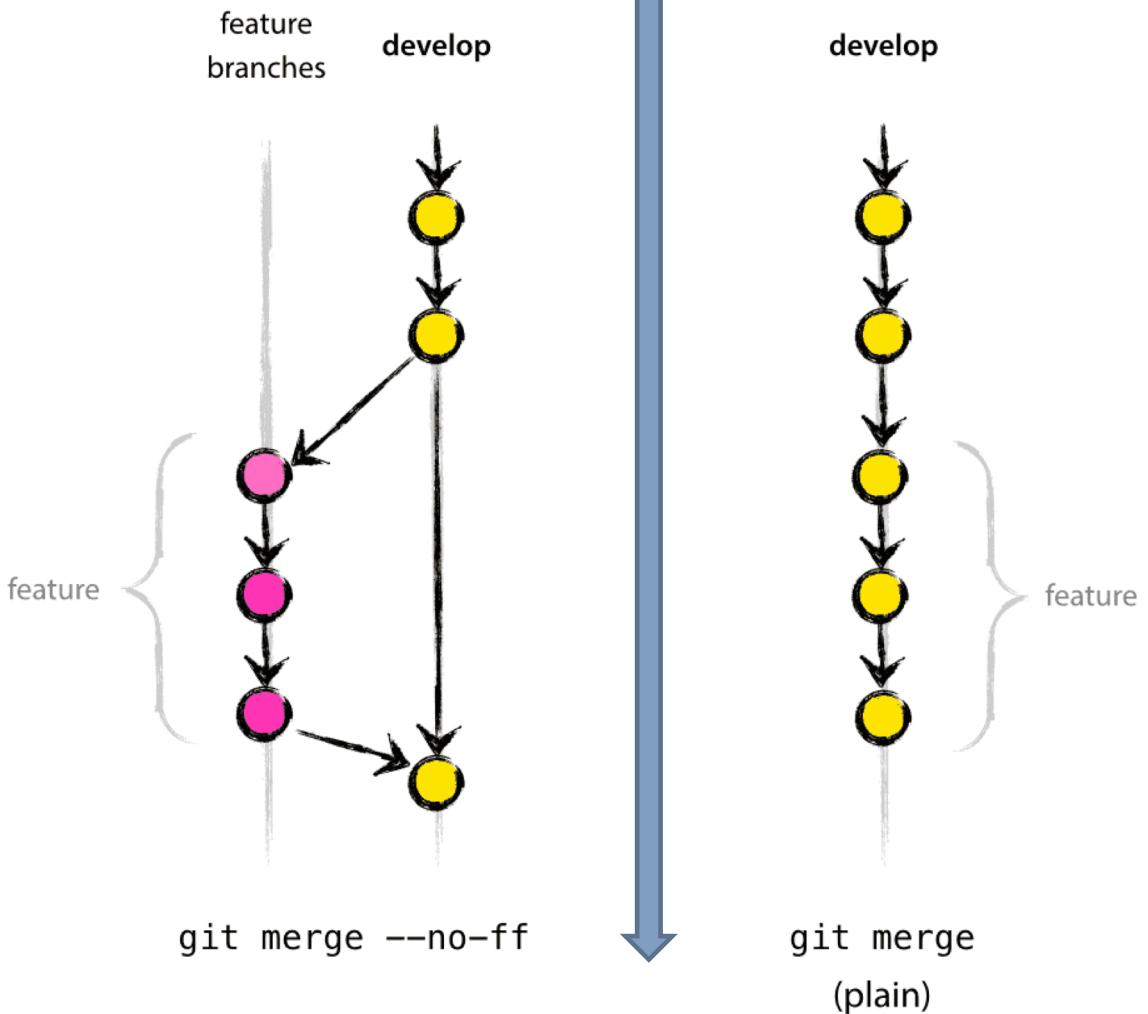
- Branch is not like a tree branch. Branch is a link to commit.
- Branch moves forward to a new commit if HEAD is attached to branch.
- Branch is almost free.
- Meld is a good tool for merge conflicts.



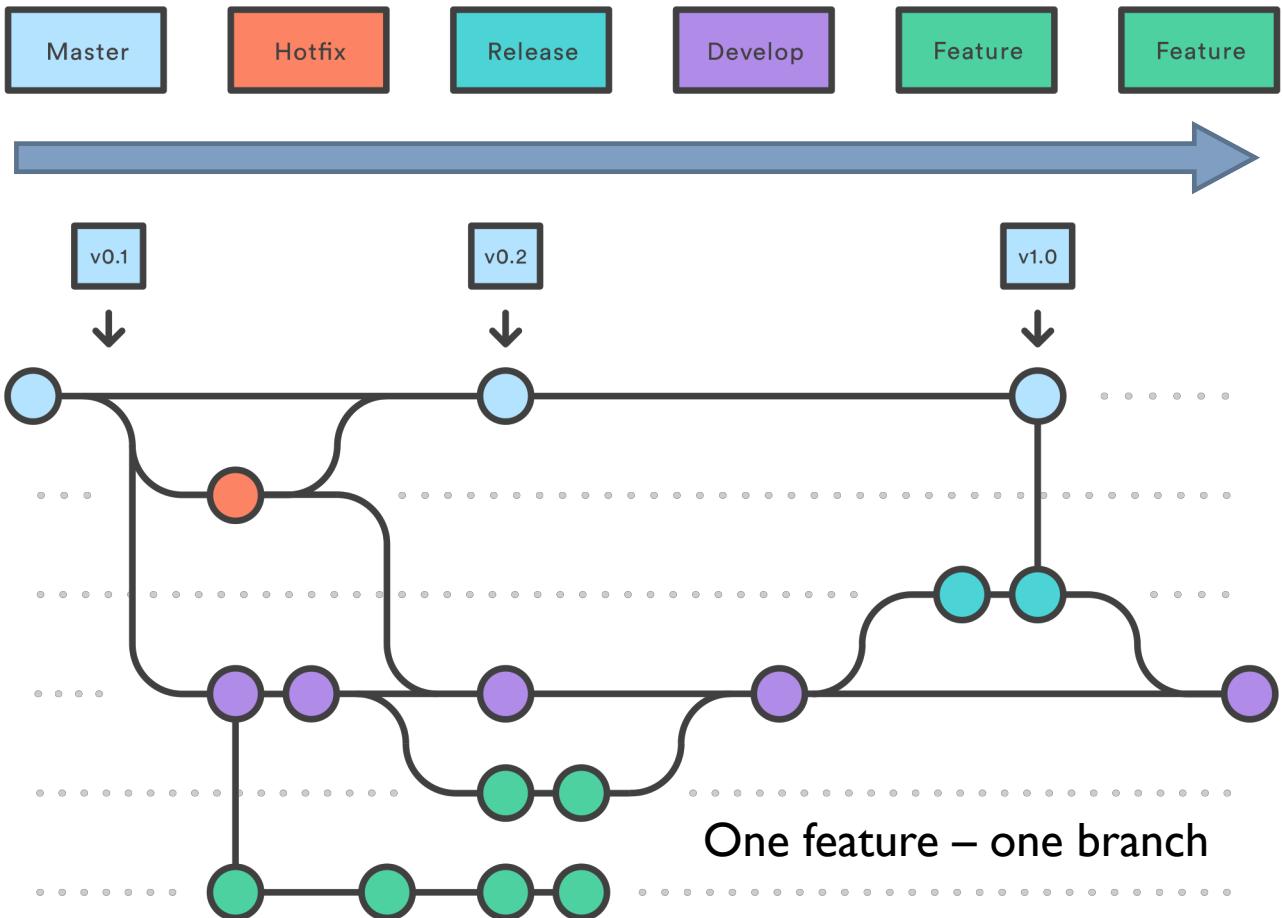
GIT WORKFLOW – LOCAL 2

CONTROLLING THE GRAPH

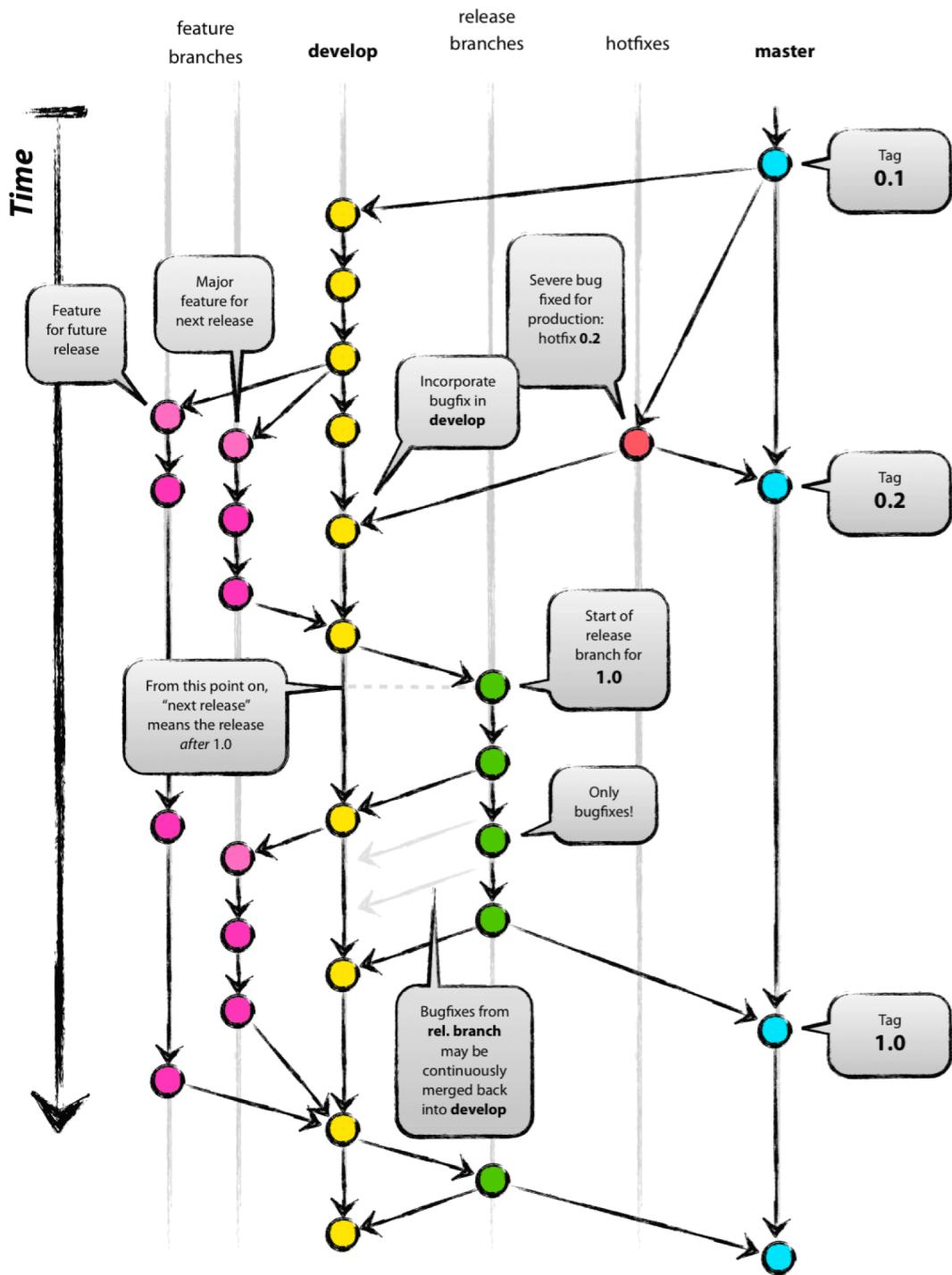
FAST-FORWARD MERGE



GITFLOW BY ATLASSIAN



<https://nvie.com/posts/a-successful-git-branching-model>



.gitignore

- .gitignore contains filenames to be ignored by `$ git add`
- For output and compiled files, tons of templates are available at
<https://github.com/github/gitignore>



REMOTES

GITHUB AND OTHERS



GIT != GITHUB

- Github – one of many, the most popular.
- GitLab, BitBucket, Overleaf...
- Git can be local-only
- Remotes can be on the same host

REMOTES

Local Computer

Checkout

File

Version Database

Version 3

Version 2

Version 1

REMOTES

Computer A

File

Central VCS Server

Version Database

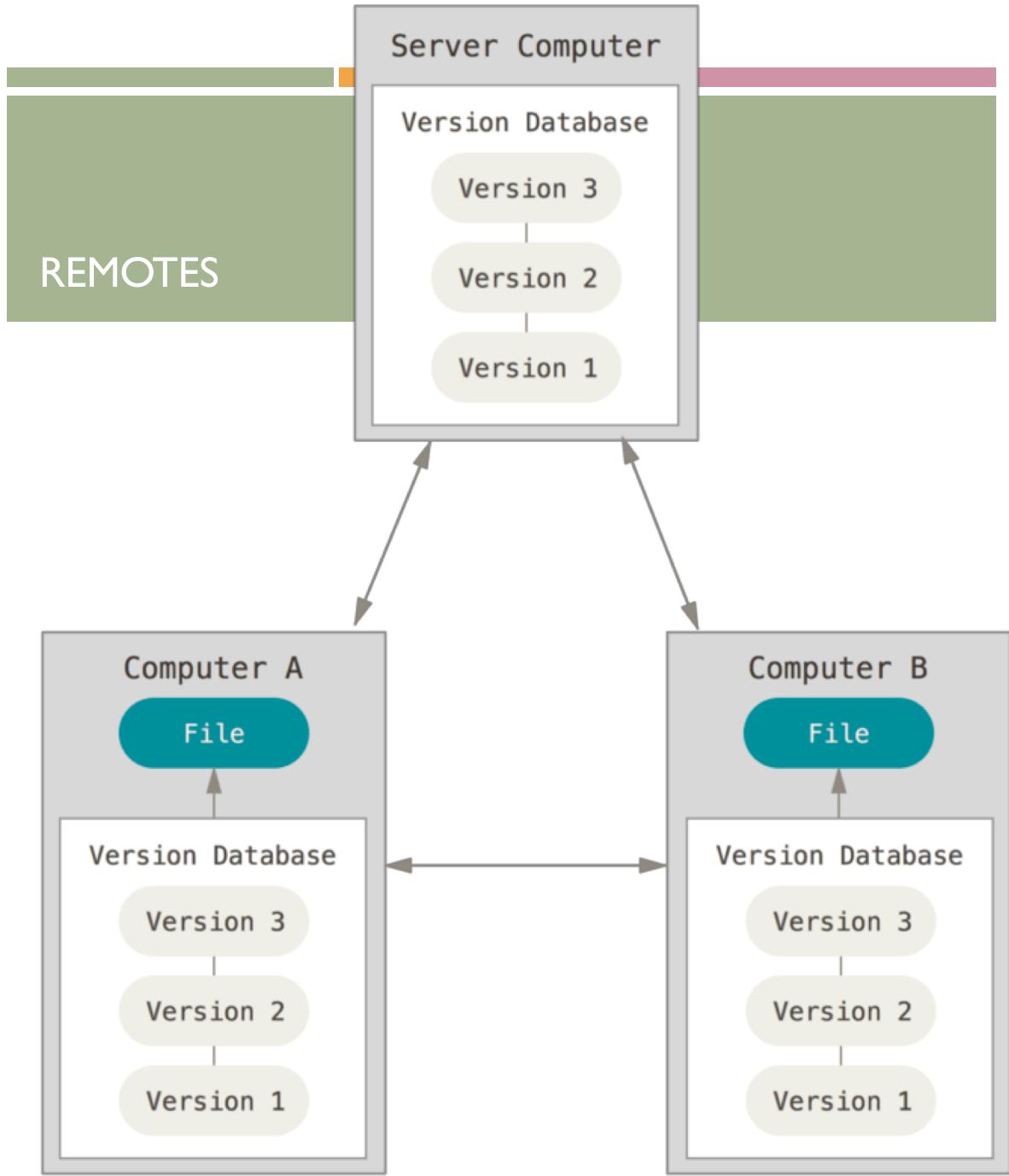
Version 3

Version 2

Version 1

Computer B

File



REMOTES

Clone from remote

- Create repository on your favourite cloud.
- Clone it (the link will be provided)

```
$ git clone  
https://gitlab.com/username/reponame.git
```
- The remote will be called "origin" by default

Create locally and push to remote

- Create repository locally
- ```
$ git init
```
- Add remote  

```
$ git remote add <remote>
git@gitlab.com:username/reponame.git
```
- ```
$ git push --set-upstream  
<remote> <branch>
```

Remotes can be accessed with SSH or HTTPS or FILE protocols. SSH requires SHA setup, but not passwords anymore.

REMOTES

- `$ git fetch # update repository – start with it!`
- `$ git pull # fetch and then merge into the current HEAD`
- `$ git push # merge HEAD into remote tracking branch`

- That's it



- Don't hesitate to ask more questions – now and always!
- Want the same pretty shell? Google oh-my-zsh

CONFIGURING GIT

- \$ git config --global user.name "John Doe"
username is default
- \$ git config --global user.email johndoe@example.com
username@hostname is default
- \$ git config --global core.editor emacs
\$EDITOR (vim) is default

- For SSH (no passwords) follow the cloud provider instructions