

GIT GUD

Agenda

1. Background
2. Internals
3. Tips & Tricks
4. Best Practices

Background

what is Git?

VCS

Version Control System

[...] a system that records
changes to a file or set of
files over time

– **Scott Chacon & Ben Straub** ProGit

Version Control Systems

- CVS
- Perforce
- SVN
- Mercurial
- Git

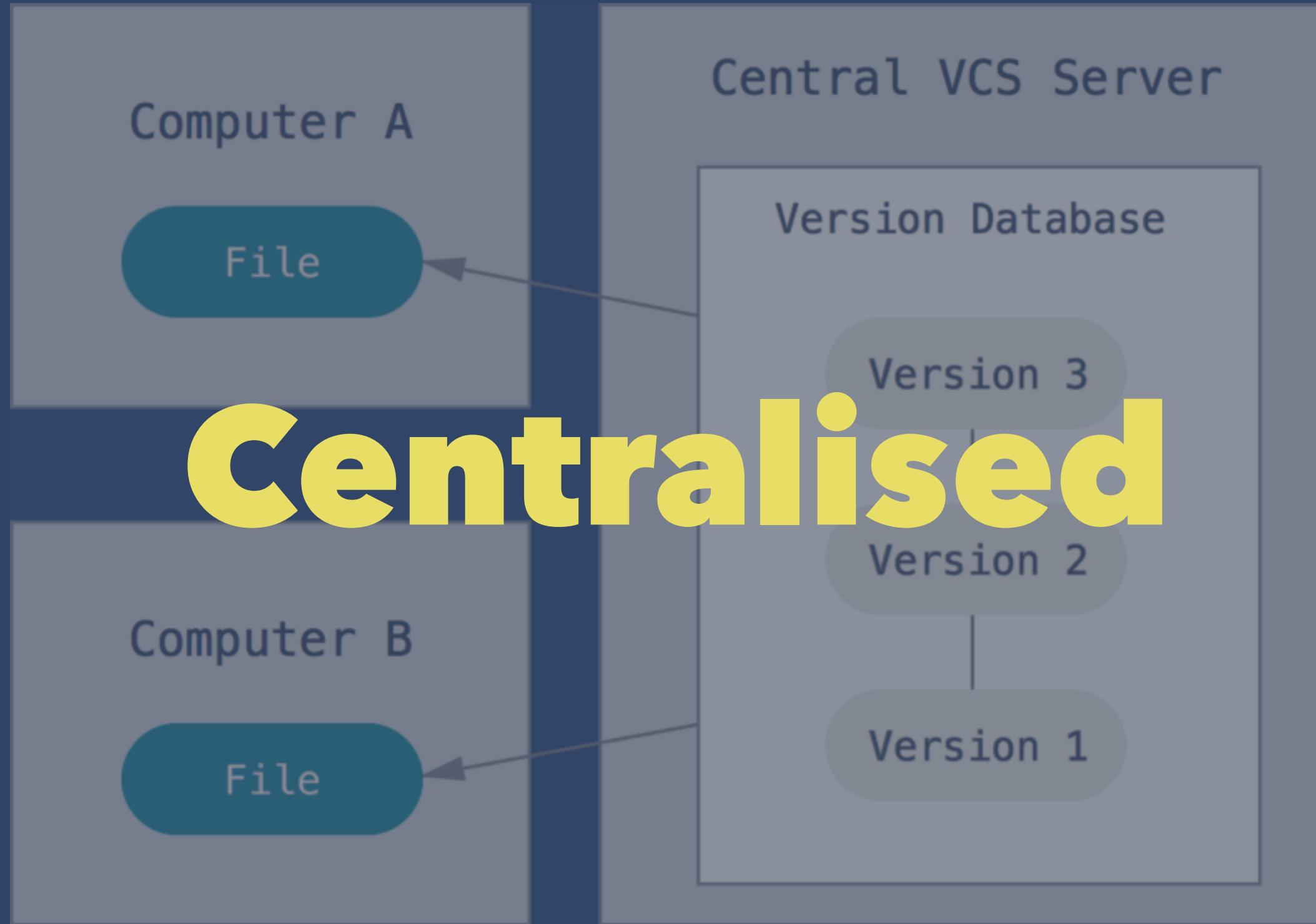
Open Source

GNU GPL2

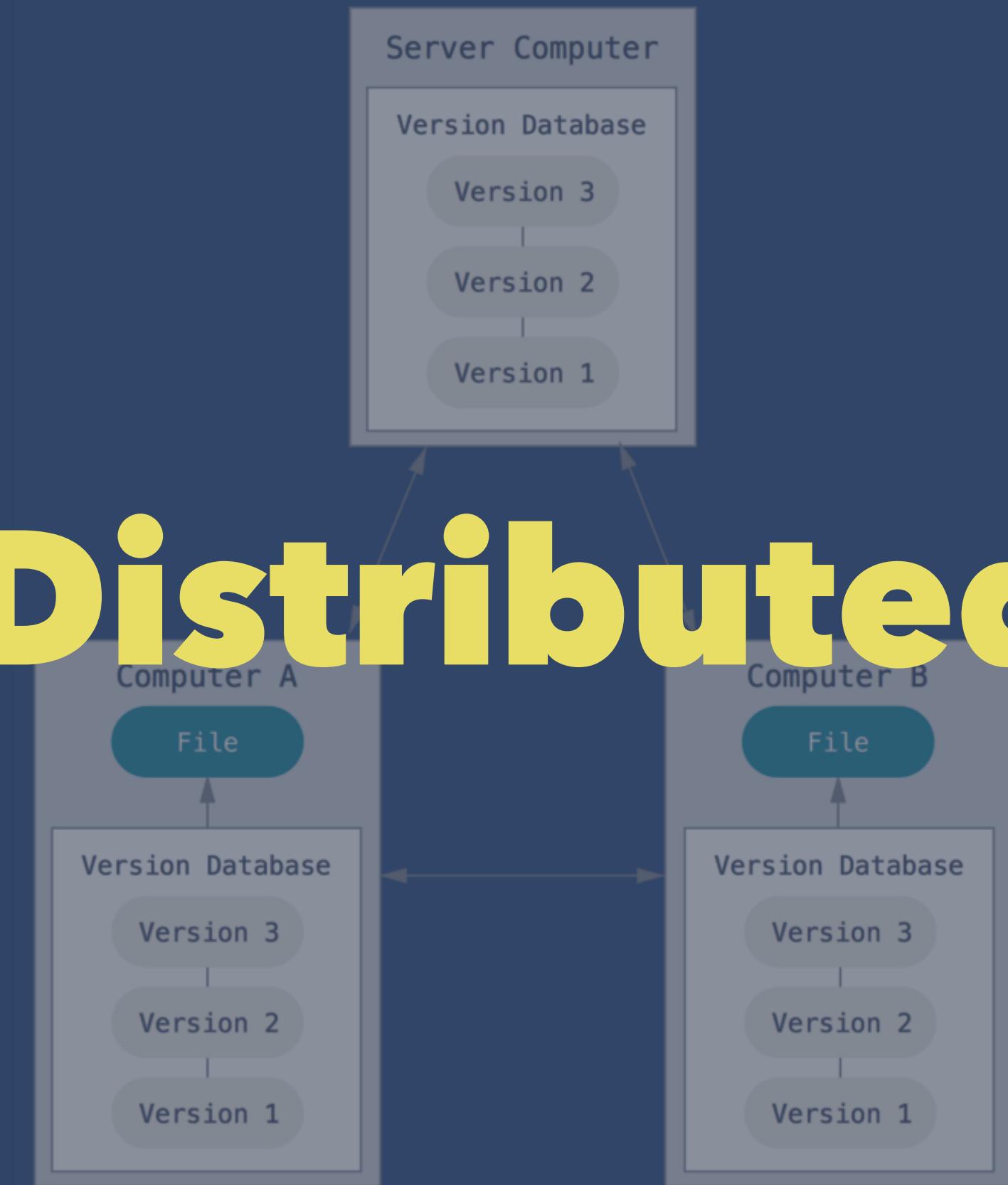
Distributed

Centralised
vs
Distributed

Centralised



Distributed





Internals

let's get our hands dirty

Expected Knowledge

- add
- reset
- commit
- merge

Working
Directory

Staging
Area

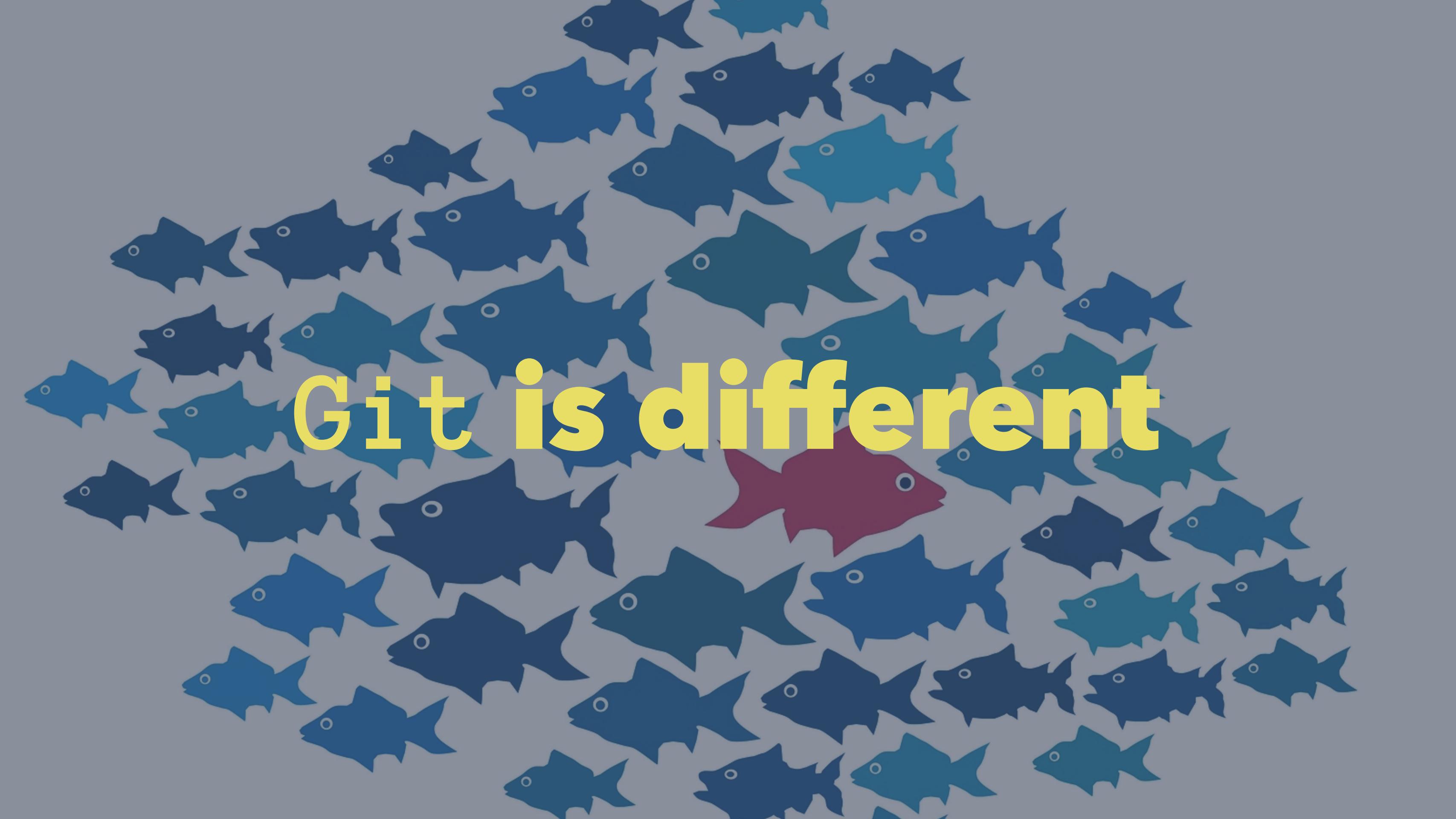
.git directory
(Repository)

Checkout the project

Stage Fixes

Commit

How does Git track changes?

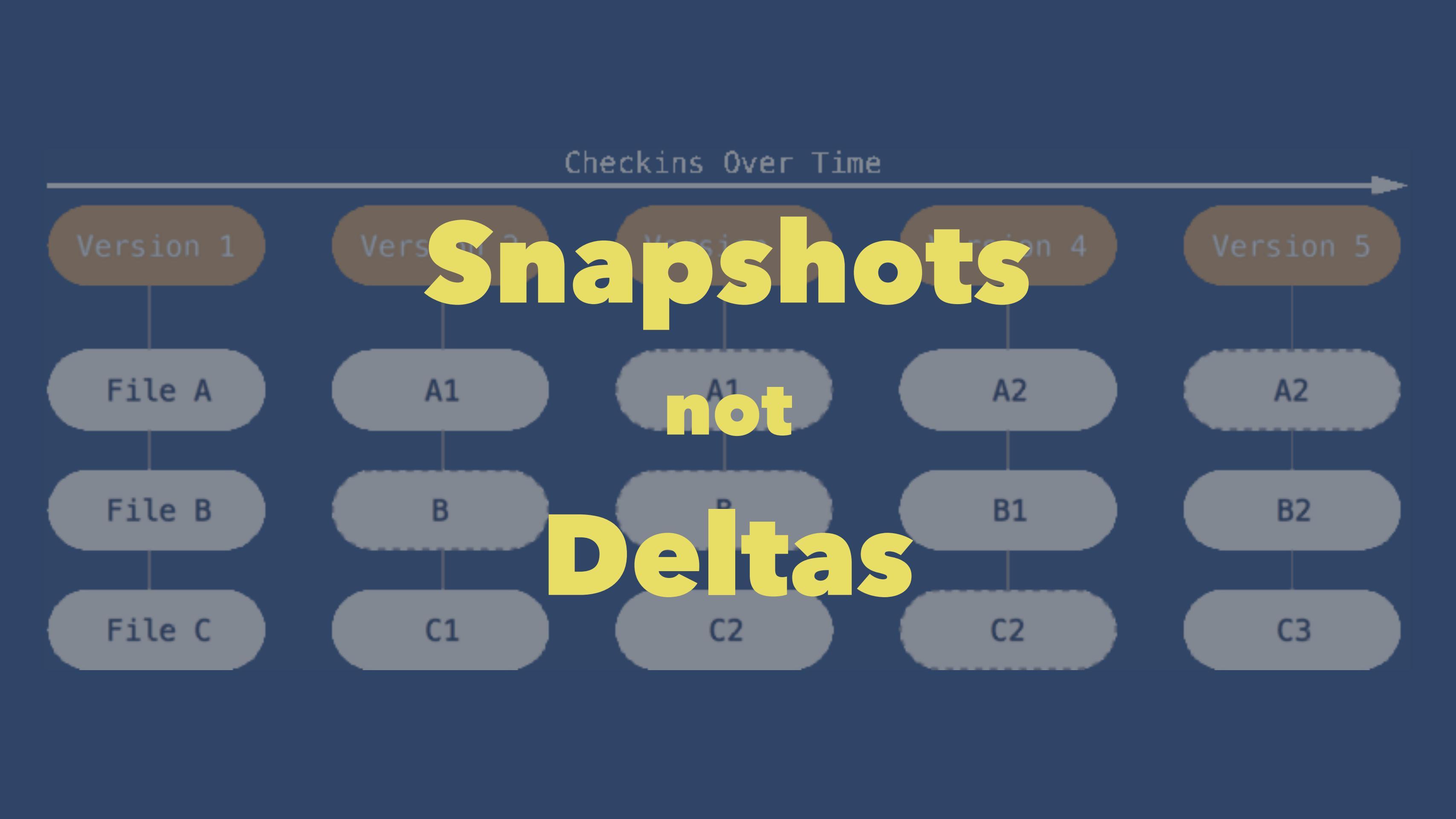


Git is different

*The major difference [...] is
the way Git thinks about
its data.*

– **Scott Chacon & Ben Straub** ProGit

Checkins Over Time



Snapshots not Deltas

What is a
Commit

98ca9

commit size
tree 92ec2
author Scott
committer Scott

The initial commit of my project

92ec2

tree size
blob 5b1d3 README
blob 911e7 LICENSE
blob cba0a test.rb



SHA1

Try it out!

Clone workshop repository

```
git clone  
git@github.gcxi.de:swolf/git-workshop.git
```

```
$ cd git-workshop
```

```
$ echo 'Some random text' > my_file
```

```
$ git hash-object -w my_file  
1a76b8a41993e2c667f5b191fb57abdab2102a8b
```

```
$ git cat-file -t 1a76b8a41993e2c667f5b191fb57abdab2102a8b  
blob
```

```
$ git cat-file -p 1a76b8a41993e2c667f5b191fb57abdab2102a8b  
Some random text
```

→ git log --all --decorate --graph --oneline

- * f097585 (**HEAD → waldo-came-and-left, origin/waldo-came-and-left**) Add some more distracting stuff
- * 0356fc5 Add some distracting stuff
- * 8a28c0f Waldo left the building
- * eeaf062 Add company for waldo
- * c85f3cf Add some more distracting stuff
- * d175476 Add waldo's name to the file
- | * f20c965 (**origin/master, master**) Merge branch 'develop' into master

How history is made

How history is made

git history at least



Honour thy parents

Navigate through history

Λ

and

~ < N >

wat



```
$ git checkout ancestry
```

```
...
```

```
$ git log --oneline -1 HEAD  
822eca1 Add the number 5 to numbers
```

```
$ git log --oneline -1 HEAD^  
d4bd446 Add the number 4 to numbers
```

```
$ git log --oneline -1 HEAD^^  
1b94d52 Add the number 3 to numbers
```

```
$ git log --oneline -1 HEAD~2  
1b94d52 Add the number 3 to numbers
```



What is a
Branch

Let's take a look

```
$ ls -l .git/refs/heads
total 32
-rw-r--r-- 1 swolf staff 41 Jan 16 09:06 ancestry
-rw-r--r-- 1 swolf staff 41 Jan 16 07:00 dev
-rw-r--r-- 1 swolf staff 41 Jan 16 07:00 master
-rw-r--r-- 1 swolf staff 41 Jan 16 07:02 waldo-came-and-left
```

Huh, master is a file ...

I wonder what's in there ...

```
$ cat .git/refs/heads/master  
f20c96538c6dca6fb37e631388814fe941afc2ae
```

It's just a hash!

```
$ cat .git/refs/heads/master | git cat-file --batch  
f20c96538c6dca6fb37e631388814fe941afc2ae commit 540  
tree 9ab6d12a4587fb4a02de1b5e745277c6c832cf5a  
parent c9866a994539e01938ceec6c75ada013ce456e9b  
parent 12e86b3370f931c423022a8c8d7d7fa7413e0bc4  
author Sascha Wolf <sascha.wolf@grandcentrix.net> 1516082444 +0100  
committer Sascha Wolf <sascha.wolf@grandcentrix.net> 1516082444 +0100  
gpgsig -----BEGIN PGP SIGNATURE-----
```

```
iHUEABEIA...Mj...07ff76Vkk26j80vFo8w6AAUCW12VDAAKCRCj80vFo8w6  
AOqtAPoCPrGCN1QZKKGm1qZu43n824v2wviWbUujd9CBwuUURQD/dn2EJ2zHi+zQ  
qBKm1cfsi5vwTiYj3104Ub0Z1rJbW1M=  
=vi7g  
-----END PGP SIGNATURE-----
```

Merge branch 'dev'

What about HEAD?

```
$ cat .git/HEAD  
ref: refs/heads/master
```

```
$ git checkout --detach master  
...
```

```
$ cat .git/HEAD  
f20c96538c6dca6fb37e631388814fe941afc2ae
```



**What is a
Merge**

What does the log tell us?

```
$ git log --decorate --graph --oneline
*   f20c965 (origin/master, master) Merge branch 'dev'
| \
| * 12e86b3 (origin/dev, dev) Add bar
| * 4043e69 Add some content to foo
| /
* c9866a9 Add foo
```

Remember the commit object?

```
$ cat .git/refs/heads/master | git cat-file --batch  
f20c96538c6dca6fb37e631388814fe941afc2ae commit 540  
tree 9ab6d12a4587fb4a02de1b5e745277c6c832cf5a  
parent c9866a994539e01938ceec6c75ada013ce456e9b  
parent 12e86b3370f931c423022a8c8d7d7fa7413e0bc4  
author Sascha Wolf <sascha.wolf@grandcentrix.net> 1516082444 +0100  
committer Sascha Wolf <sascha.wolf@grandcentrix.net> 1516082444 +0100  
gpgsig -----BEGIN PGP SIGNATURE-----
```

```
iHUEABEIA...Mj...07ff76Vkk26j80vFo8w6AAUCW12VDAAKCRCj80vFo8w6  
AOqtAPoCPrGCN1QZKKGm1qZu43n824v2wviWbUujd9CBwuUURQD/dn2EJ2zHi+zQ  
qBKm1cfsi5vwTiYj3104Ub0Z1rJbW1M=  
=vi7g  
-----END PGP SIGNATURE-----
```

Merge branch 'dev'

Tips & Tricks

We're going to look at

- bisect
- --patch mode
- stash
- reflog
- rebase
- filter-branch

One thing before we start ...

Aliase

```
git config --global alias.<your-alias> <command>
```

In Action

```
$ git config --global alias.l 'log --decorate --graph --oneline'  
  
$ git l -3  
*   f20c965 (HEAD -> master, origin/master) Merge branch 'dev'  
| \  
| * 12e86b3 (origin/dev, dev) Add bar  
| * 4043e69 Add some content to foo  
| /
```

A close-up photograph of a shotgun lying horizontally. The shotgun has a dark, possibly black or dark blue, finish on its barrels and receiver. The wood on the stock and forestock is a rich, dark brown. The barrels are slightly curved downwards. A dark, semi-transparent rectangular overlay covers the upper half of the image, containing the text.

git bisect

a smart bug hunters weapon

git bisect

Search for the first commit which introduces an issue

```
$ git bisect start  
$ git bisect bad <known bad commit>  
$ git bisect good <known good commit>
```

Shortcut

```
$ git bisect <known bad commit> <known good commit>
```

Automate it!

```
git bisect run <command>
```

Let's try it out!

```
git bisect start waldo-came-and-left d175
```

Stop bisecting

git bisect reset



--patch
add | checkout | reset

Let's try it out!

```
$ git checkout patch-practice
```

```
...
```

```
$ git apply patch-practice.diff
```

```
...
```

```
$ git add --patch # Try to add only the above line!
```



Level up your --patch

edit mode

A wooden Easter basket is nestled in a patch of vibrant green grass. The basket is overflowing with hand-painted Easter eggs in various colors like red, blue, yellow, and white, some featuring floral or animal designs. A small white rabbit figurine is peeking out from behind the basket. The background is a soft-focus view of more green grass.

git stash
quicksave for git

git stash

- push create a stash from current changes
- pop apply and delete a stash
- drop delete a stash

Use `stash@{<n>}` to reference older stashes

git reflog
a history for your HEAD

Try it out!

git reflog

Branches have reflogs too!

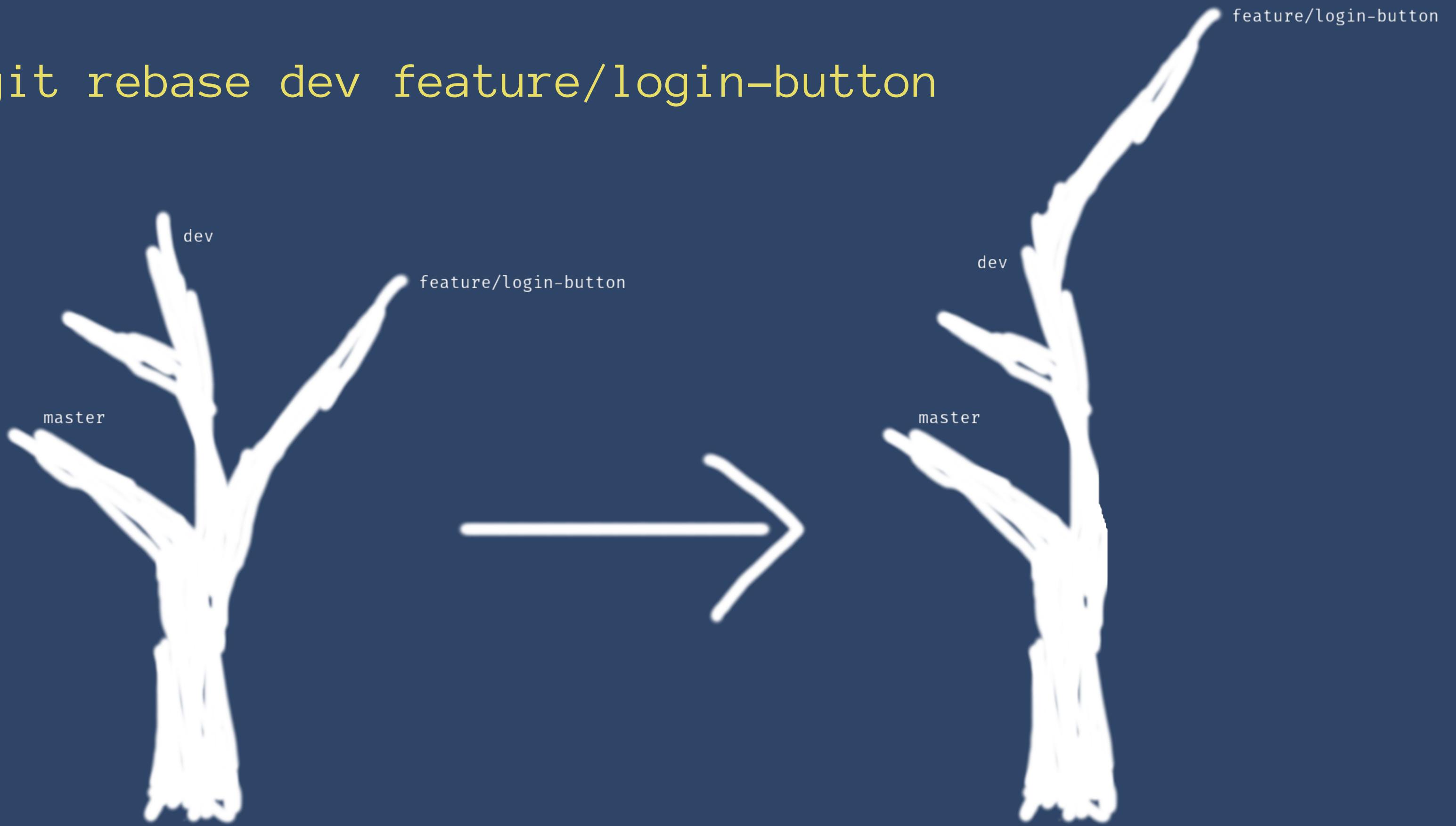
```
git reflog <some branch>
```

git rebase

one command to rule them all

feature/login-button

git rebase dev feature/login-button



Let's try it out

```
$ git checkout rebase-practice
```

```
...
```

```
$ git rebase dev  
???
```



git rebase --interactive

Let's fix this

We don't want to remove foo!

```
$ git rebase --abort
```

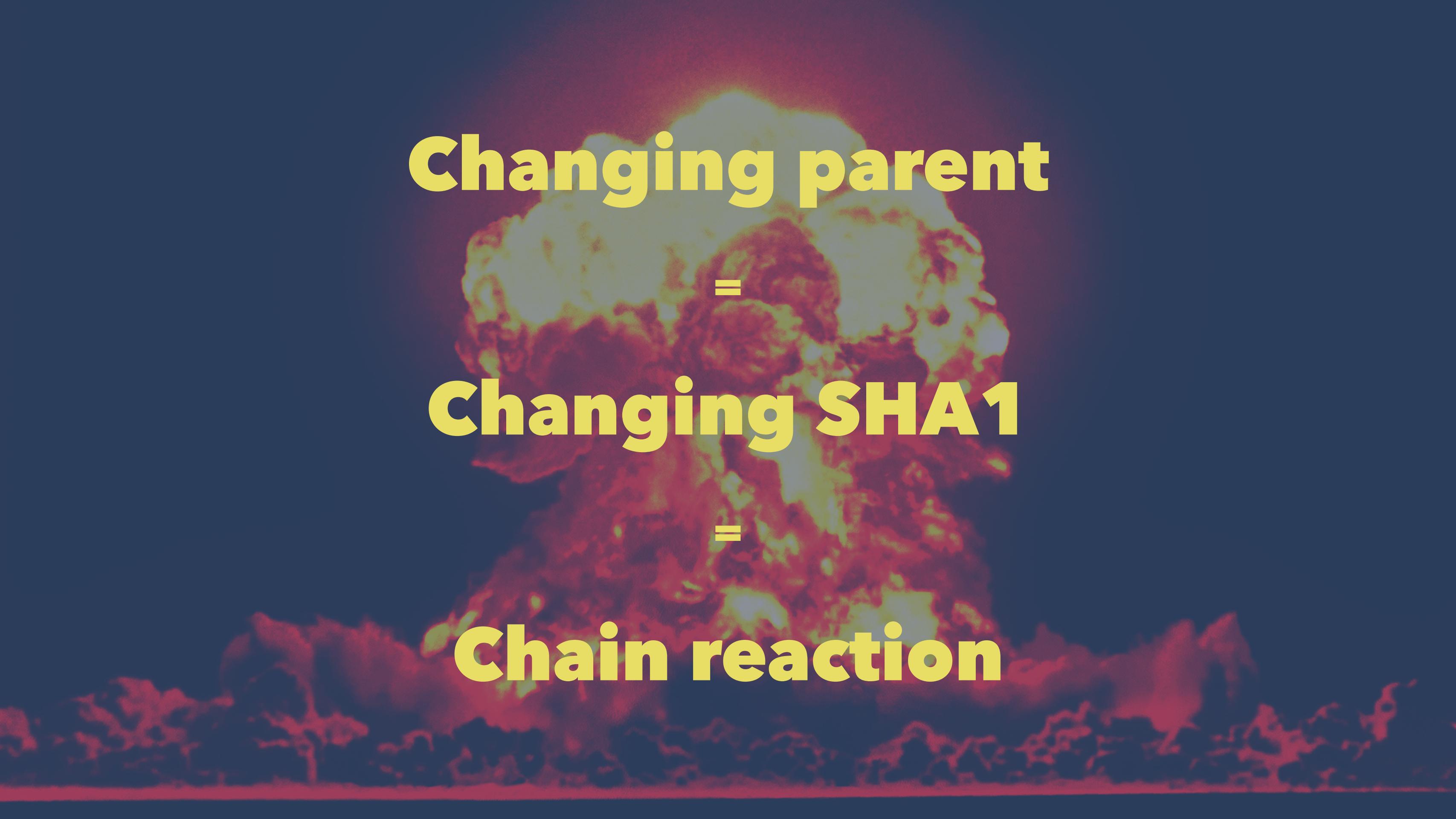
...

```
$ git rebase --interactive dev  
???
```

Avoid rebasing pushed commits!

rebase rewrites history

with great power comes great responsibility



Changing parent

=

Changing SHA1

=

Chain reaction



MOAR POWER!

over history

git filter-branch

REWRITE



ALL THE HISTORY!

git filter-branch

Allows to rewrite

```
--env-filter <command>
--index-filter <command>
--msg-filter <command>
--parent-filter <command>
--subdirectory-filter <directory>
--tag-name-filter <command>
--tree-filter <command>
```

Honourable mentions

- `cherry-pick`: "copy" one+ commits onto the current branch
- `clean`: remove untracked files and folders
- `commit --amend`: include added changes in last commit
- `grep`: search pattern in tracked files
- `worktree`: checkout a reference into another worktree



Best Practices²

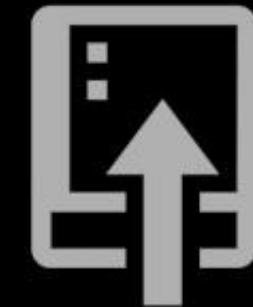
² <http://sethrobertson.github.io/GitBestPractices>



IN CASE OF FIRE



Git Commit



Git Push



Git Out

**Commit often
Perfect later
Publish once**

Use shortlived branches

or at least merge upstream work regularly

**Do not rewrite
published history!**



ORBO

Commit Messages

Commit Messages

<Topic|File>: <Short description> (50 characters sort limit)

<More detailed description> (bullet points are fine)

- problem (what is the issue with the current implementation)
- chosen approach with reasoning
- maybe rejected approaches

Use imperativ

Add foo instead of Added foo

	COMMENT	DATE
O	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
O	ENABLED CONFIG FILE PARSING	9 HOURS AGO
O	MISC BUGFIXES	5 HOURS AGO
O	CODE ADDITIONS/EDITS	4 HOURS AGO
O	MORE CODE	4 HOURS AGO
O	HERE HAVE CODE	4 HOURS AGO
O	AAAAAAA	3 HOURS AGO
O	ADKFJSLKDFJSDFKJ	3 HOURS AGO
O	MY HANDS ARE TYPING WORDS	2 HOURS AGO
O	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

A photograph of a diverse group of people, likely students, sitting in rows in a classroom. They are all looking towards the right side of the frame. The background is filled with green foliage from trees outside the window.

Thank you