

Knowledge is Power: Understanding git

Steve Smith
@tarkasteve



follow us @gotochgo

Conference: May 24th-25th / Workshops: May 23rd & 26th

“ Git is fundamentally a content-addressable filesystem with a VCS user interface written on top of it ,”

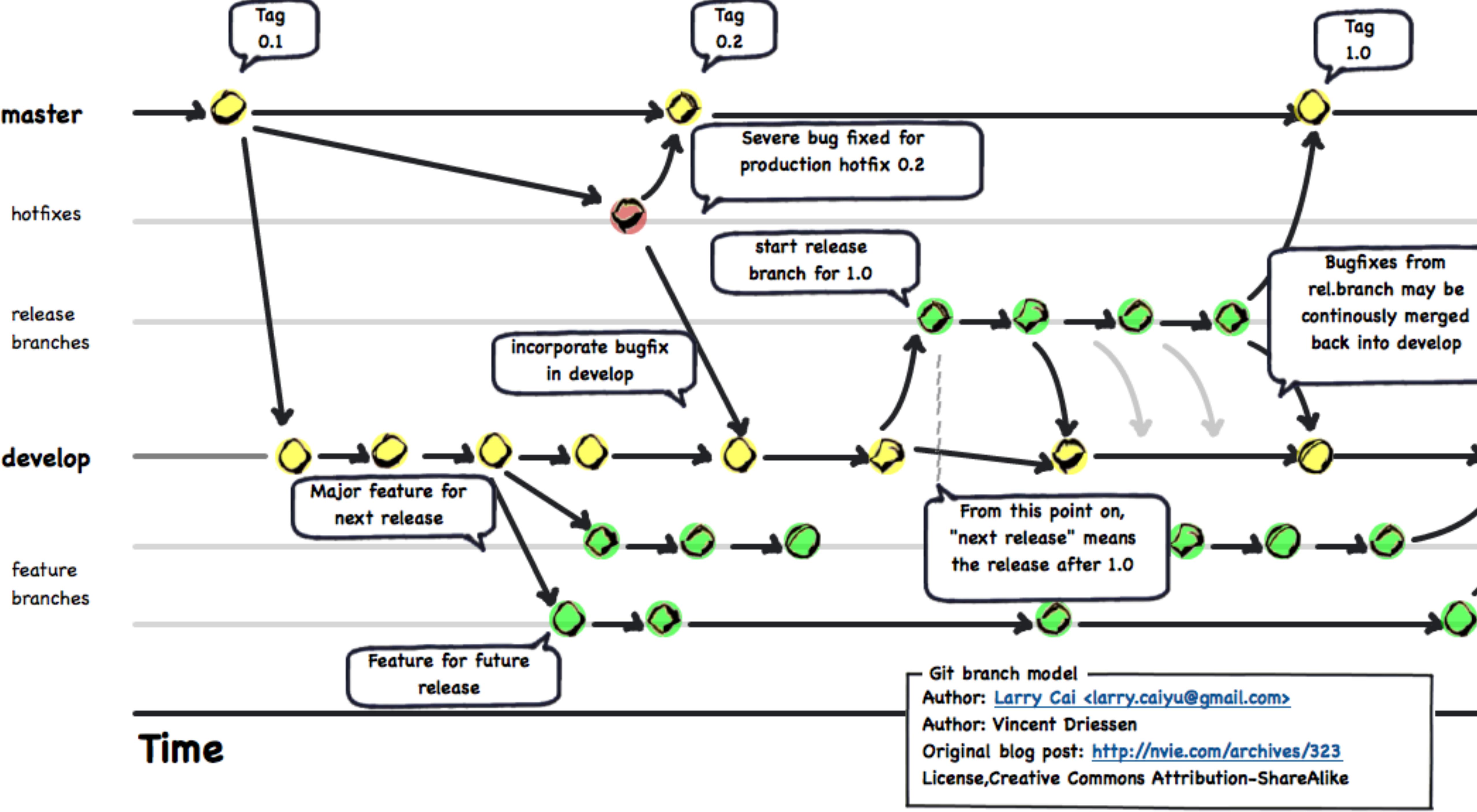
Pro Git Book, Section: Git Internals



“ It's easy, it's just a directed acyclic graph! ”
If I hear that one more time I may have to
punch something!

Emma Jane Hogbin Westby, Git-Merge 2015





THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.

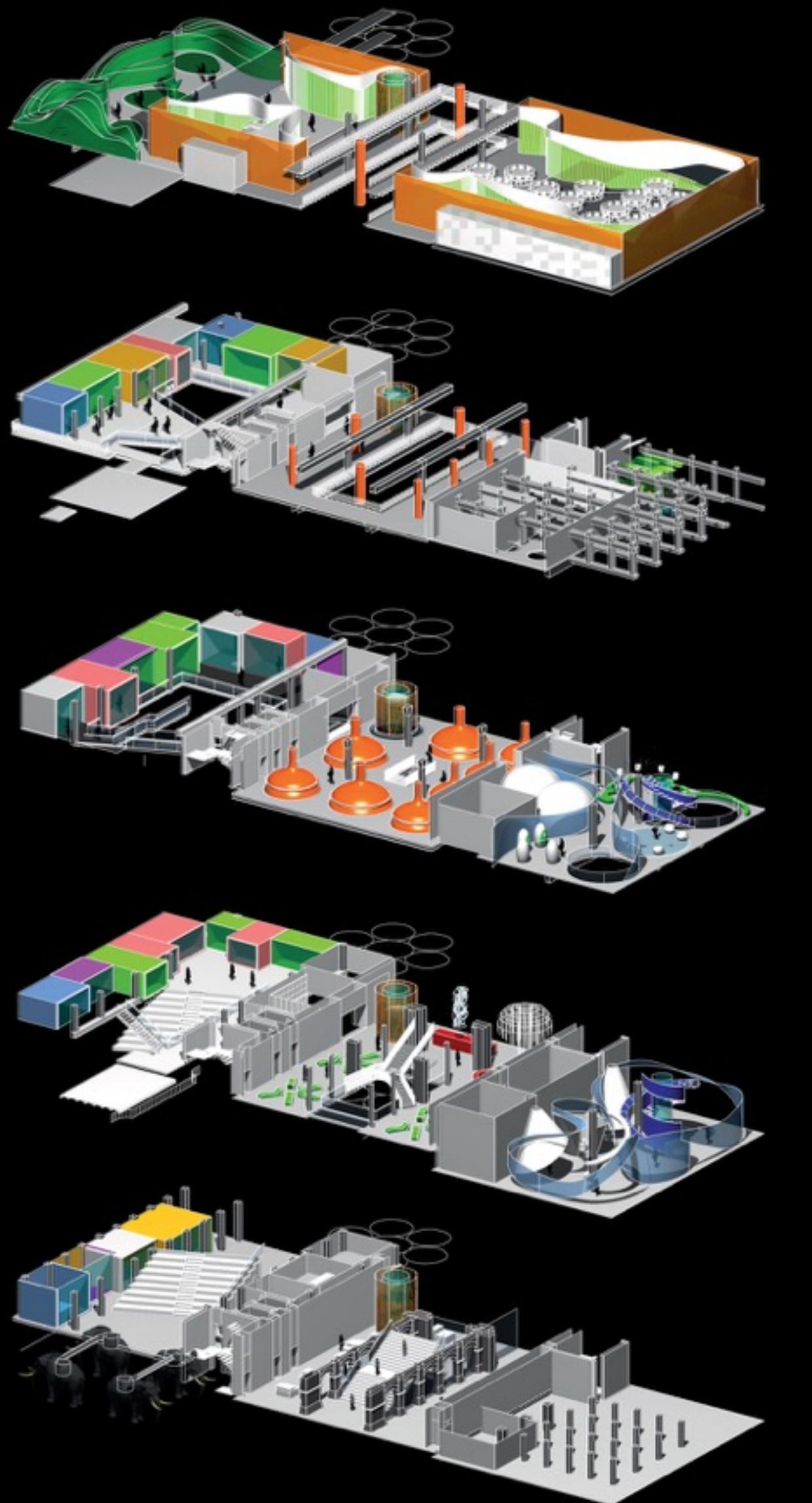


“ If that doesn't fix it, git.txt contains the phone number of a friend of mine who understands git. Just wait through a few minutes of 'It's really pretty simple, just think of branches as...' and eventually you'll learn the commands that will fix everything. ”





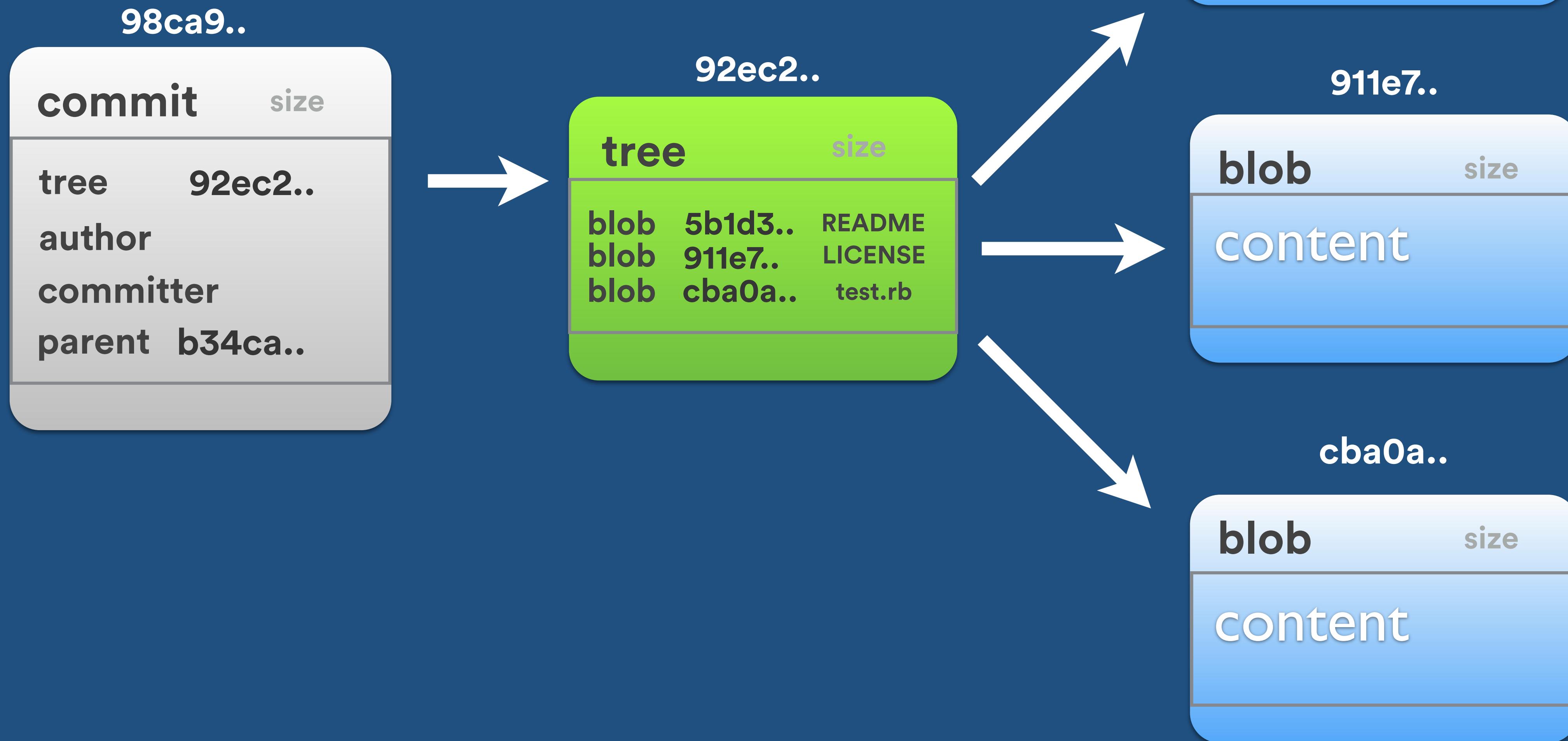
git internals







git data model



```
$> git init  
$> tree .git/objects  
.git/objects  
├── info  
└── pack
```

2 directories



```
$> touch some-file.txt  
$> git add some-file.txt
```



```
$> tree .git/objects
.git/objects
└── e6
    └── 9de29bb2d1d6434b8b29ae775ad8c2e48c5391
└── info
└── pack
3 directories, 1 file
```

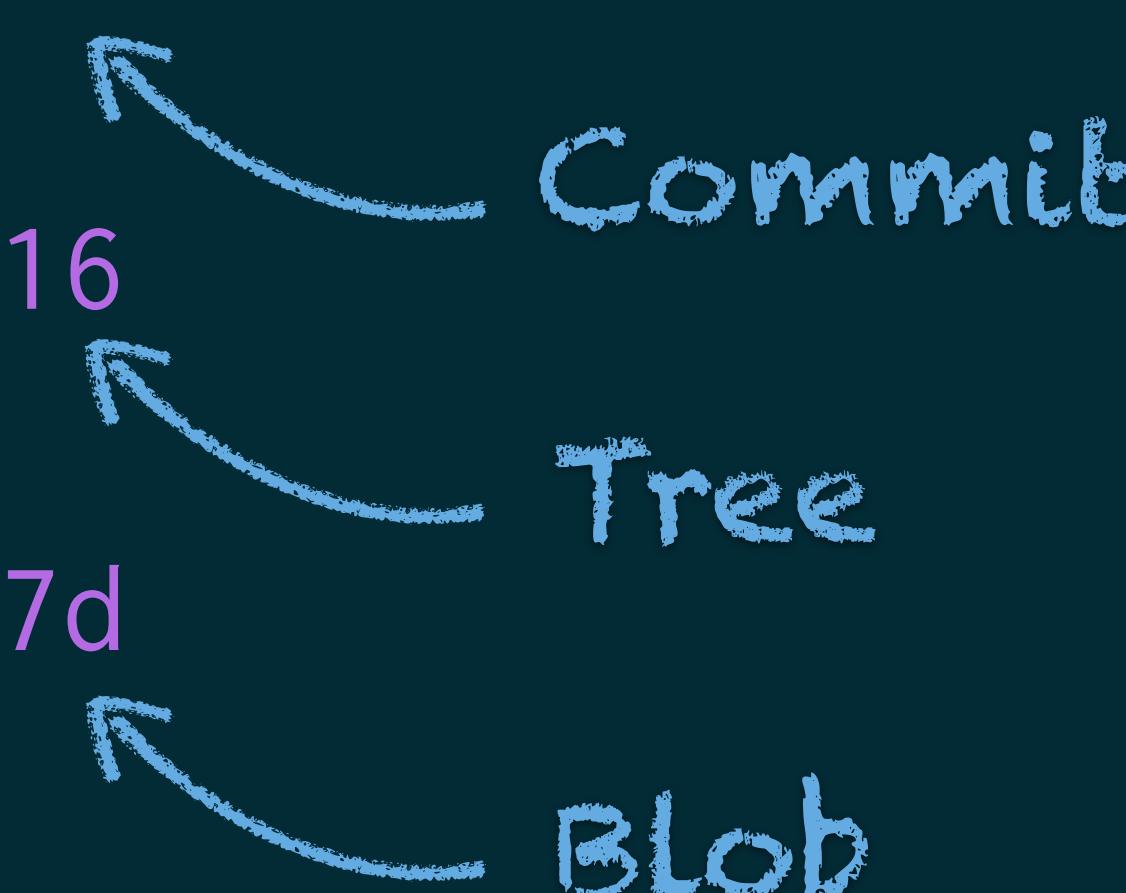
zLib compressed
SHA1



```
$> git commit -m "First commit"
```



```
$> tree .git/objects
.git/objects
└── 13
    └── 1e360ae1a0c08acd18182c6160af6a83e0d22f
        └── 31
            └── 995f2d03aa31ee97ee2e814c9f0b0ffd814316
                └── e4
                    └── 3a6ac59164adadac854d591001bbb10086f37d
                        └── info
                            └── pack
5 directories, 3 files
```



The diagram shows handwritten labels with arrows pointing to specific objects in the tree output:

- A blue arrow points from the word "Commit" to the root directory "13".
- A blue arrow points from the word "Tree" to the directory "31".
- A blue arrow points from the word "Blob" to the directory "e4".



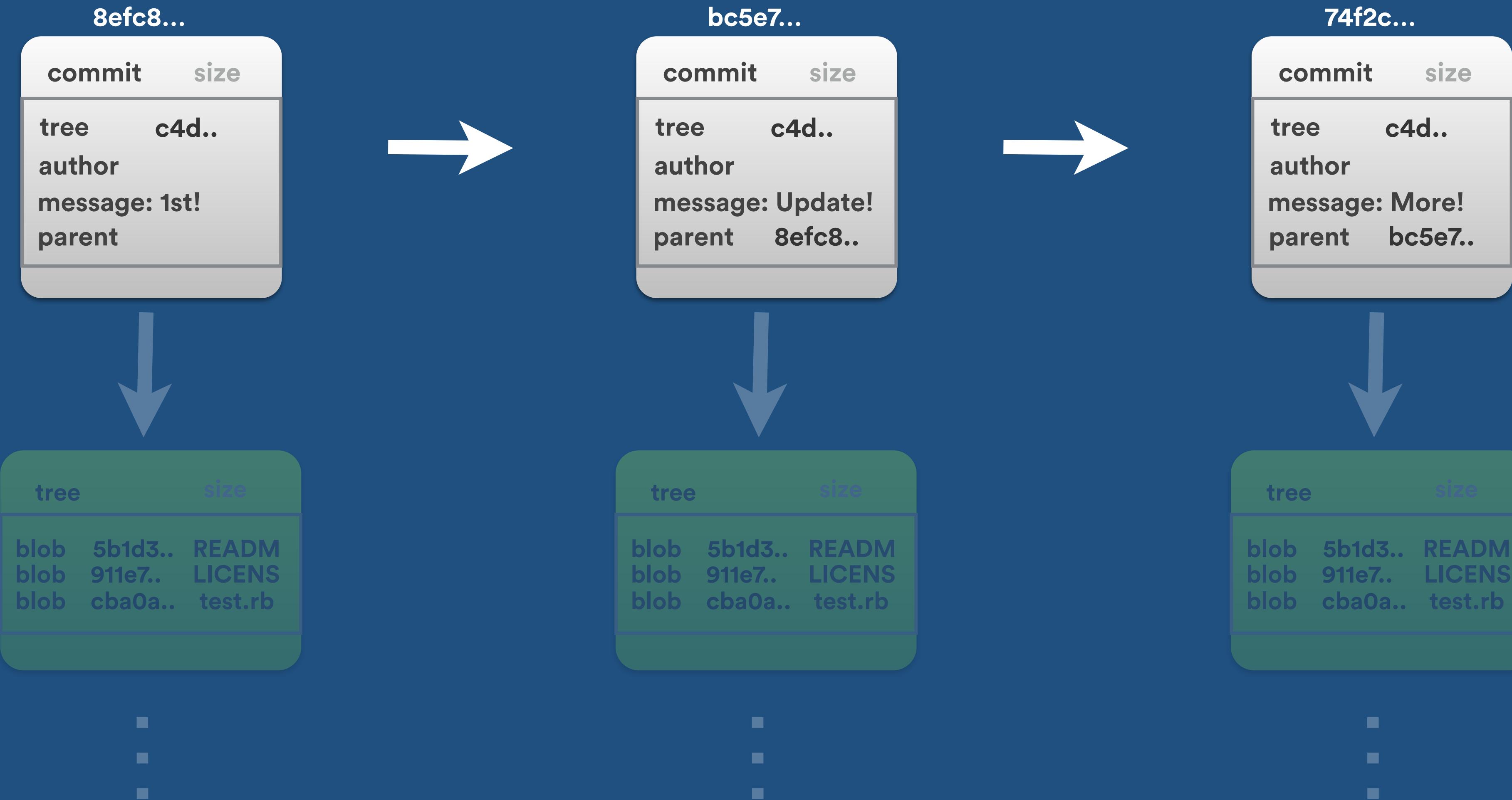


git data model





git data model



```
$> echo "/* Comment" >> some-file.txt  
$> git add some-file.txt
```



```
$> tree .git/objects  
.git/objects  
├── 13  
│   └── 1e360ae1a0c08acd18182c6160af6a83e0d22f  
├── 31  
│   └── 995f2d03aa31ee97ee2e814c9f0b0ffd814316  
├── c1  
│   └── 9e6823e34980033917b6427f3e245ce2102e6e  
└── e4  
    └── 3a6ac59164adadac854d591001bbb10086f37d
```

Entirely new BLOB

6 directories, 4 files

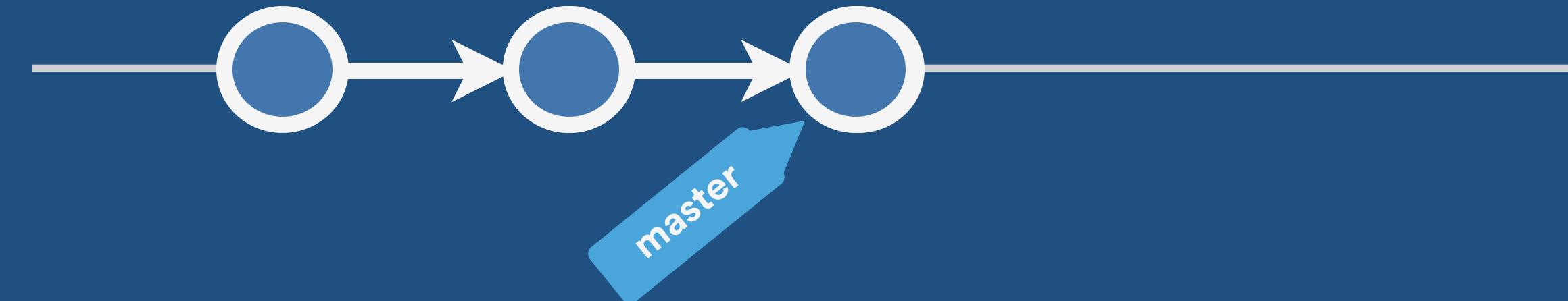




Refs and Branches
And Tags

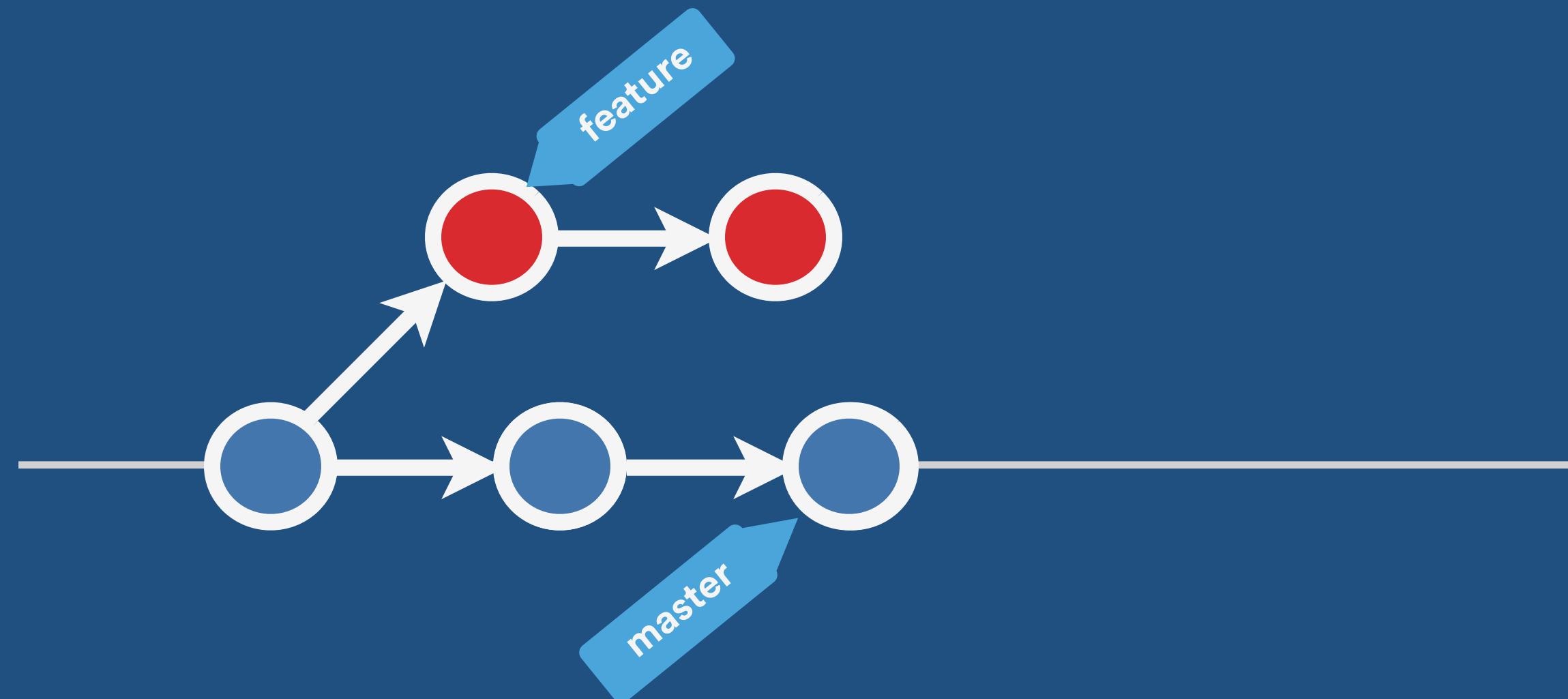
What is a ‘ref’?

A ref is just a pointer to an object



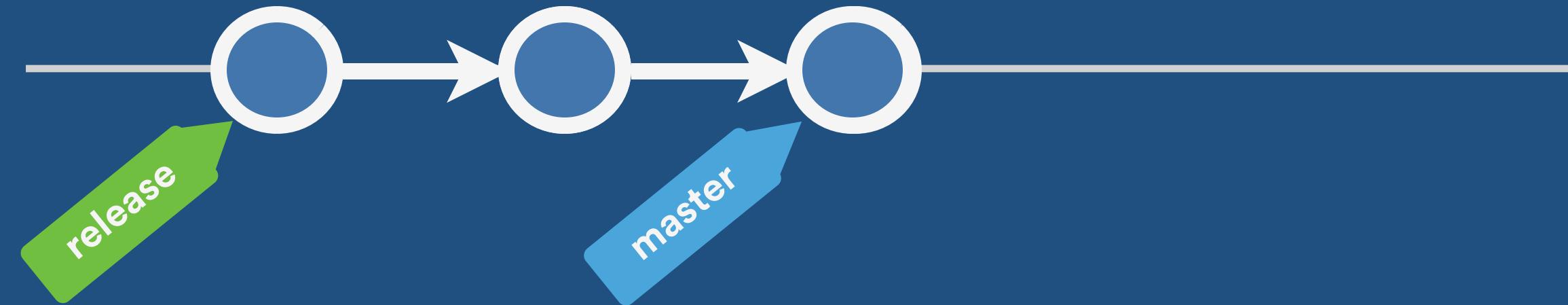
What is a ‘branch’?

A branch is divergence from a common point, and ref to a commit, the “HEAD”



What is a ‘tag’?

A tag is just special ref used a mark
a commit in the history



```
$> git tag a-tag -m"A tag"  
$> git branch a-branch
```

```
$> tree .git/refs/  
.git/refs/  
└── heads  
    ├── a-branch  
    └── master  
└── tags  
    └── a-tag
```

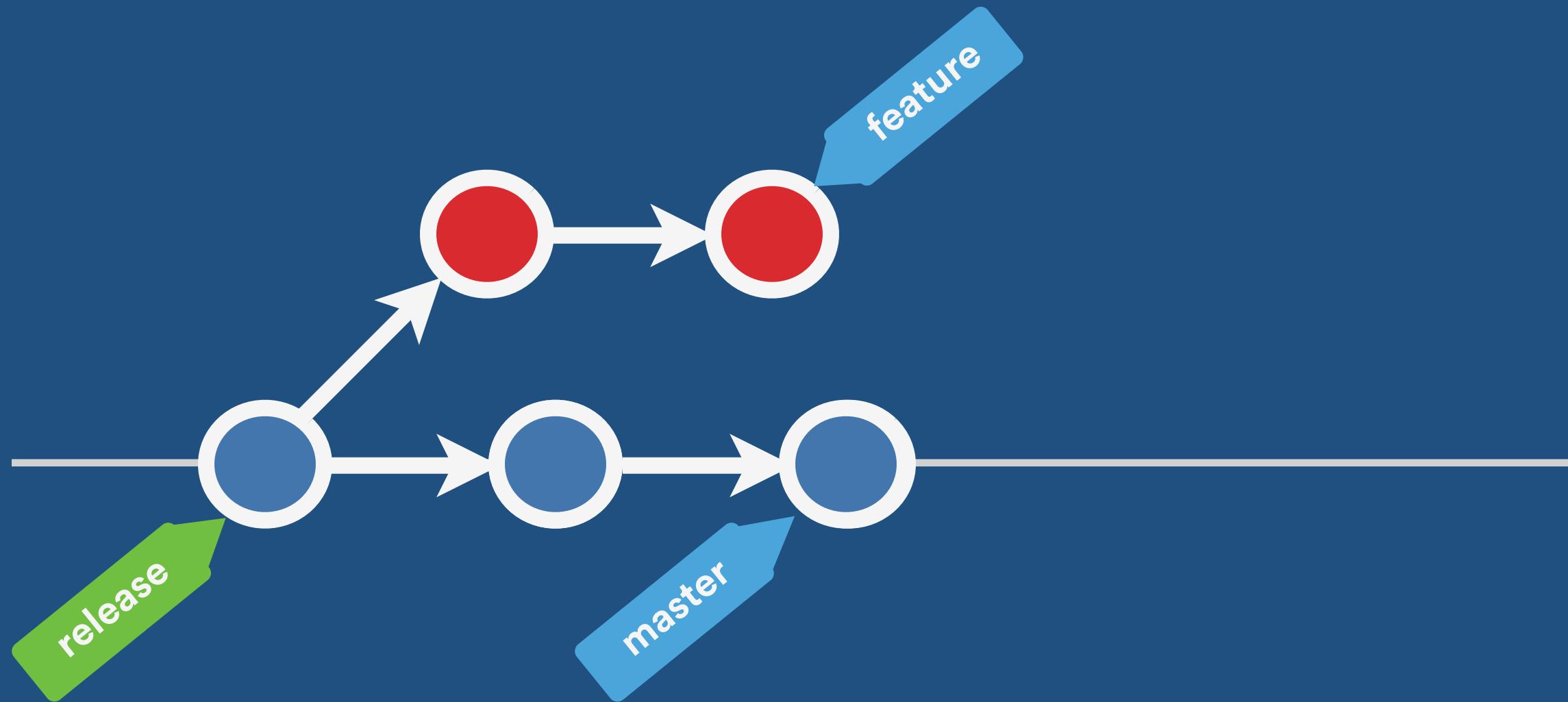
```
$> cat .git/refs/heads/a-branch  
c13e27cdfd15c5acdcd8b510eefed7be68c41c8e
```



What is a ‘reset’?

Manipulates the branch ‘head’

```
$> git reset --hard feature^          # '^' means 'parent'
```



Reflog keeps a history

```
$> git reflog
0c35628 HEAD@{1}: reset: moving to HEAD^
6cc6637 HEAD@{2}: commit: Add B
0c35628 HEAD@{3}: merge: Merge made by the 'recursive' strategy.
e0c0d65 HEAD@{4}: cherry-pick: A
80bb854 HEAD@{5}: checkout: moving from alpha to master
5044136 HEAD@{6}: commit: A
80bb854 HEAD@{7}: checkout: moving from master to alpha
80bb854 HEAD@{8}: commit (initial): 1
```

(Only 90 days by default though!)



A man in a dark trench coat and fedora hat stands in a doorway, looking towards the camera with a serious expression. He is positioned in the center-left of the frame, with a bright light source behind him creating a strong silhouette effect. The background shows a hallway with wooden doors.

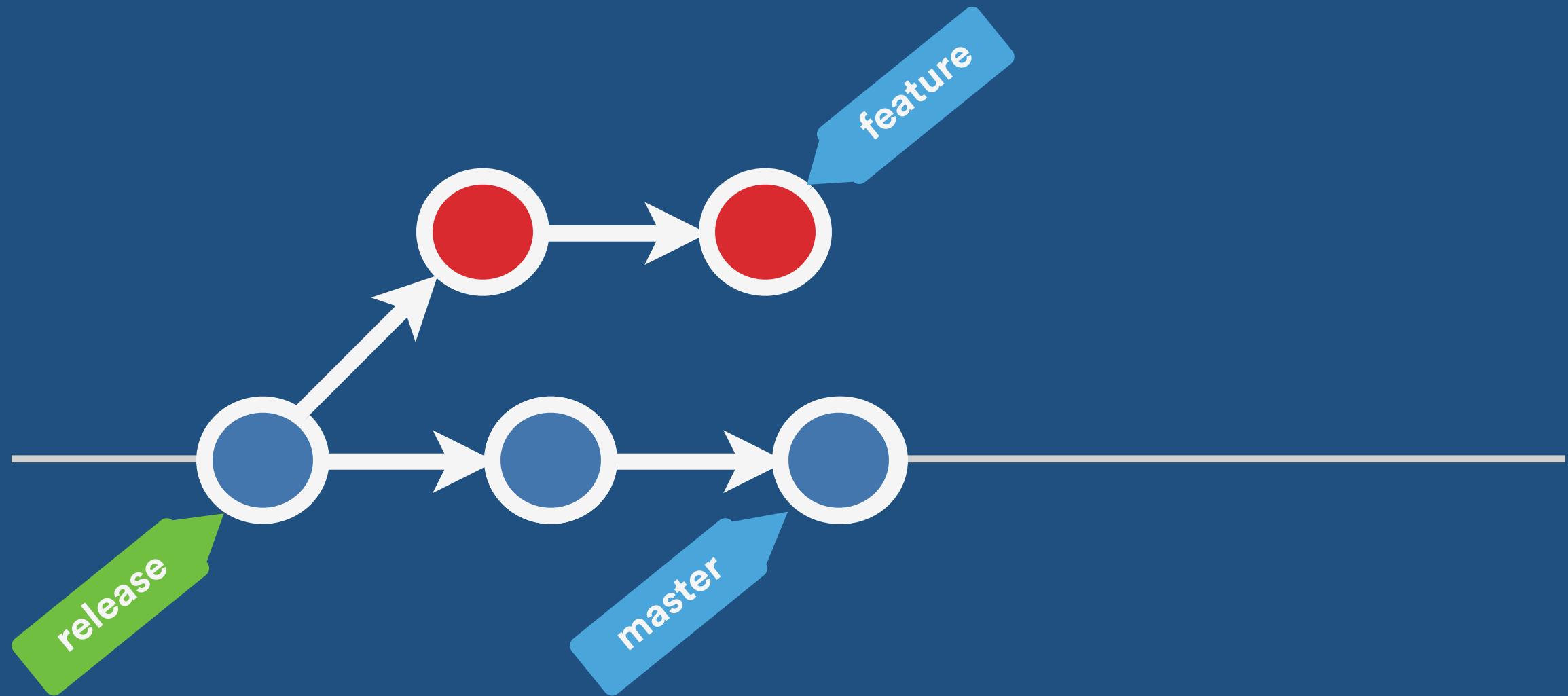
Keeping things
clean

What is a ‘gc’?

Orphaned objects are eligible for removal

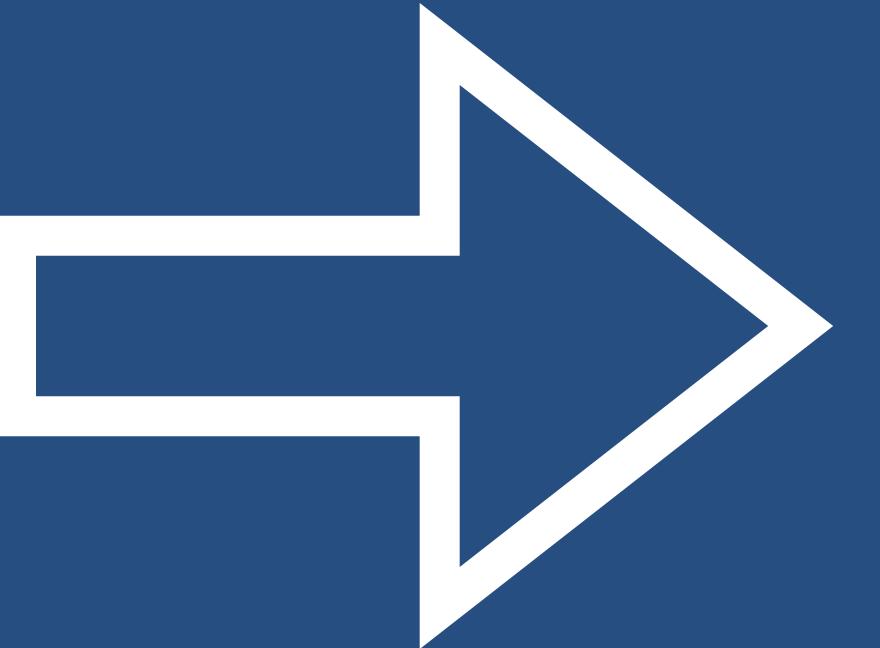
```
$> git reset feature^  
$> git gc --prune=all
```

‘^’ means ‘parent’

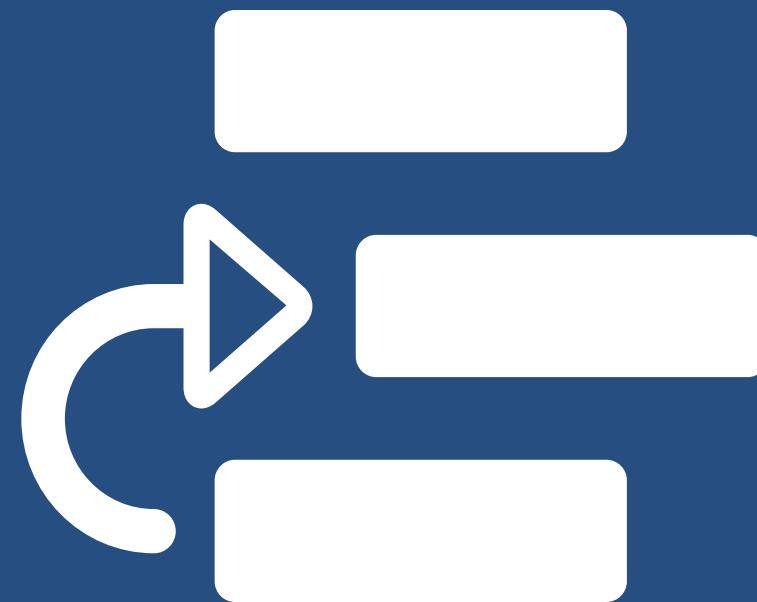


GC also packs objects

Loose Objects



Packfile



1. zLib compressed
2. Delta encoded



```
$> tree .git/objects
.git/objects
└── info
    └── packs
└── pack
    ├── pack-7475314b451a882d77b1535d215def8bad0f4306.idx
    └── pack-7475314b451a882d77b1535d215def8bad0f4306.pack
```

2 directories, 3 files



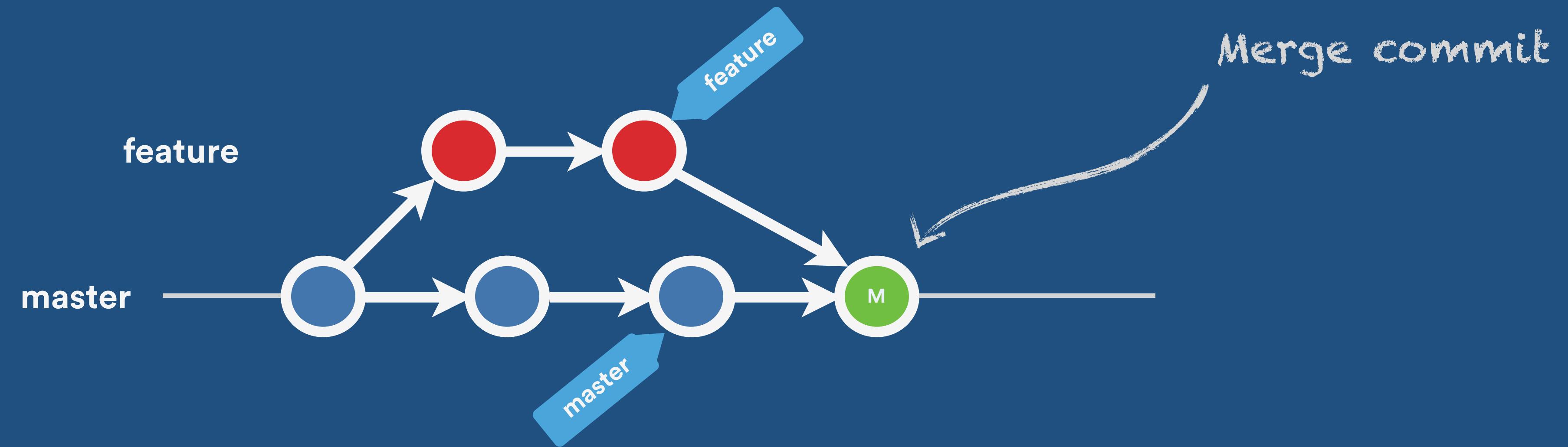


git merge



What is a merge?

merges keep the context of
the feature's commits



Anatomy of a merge

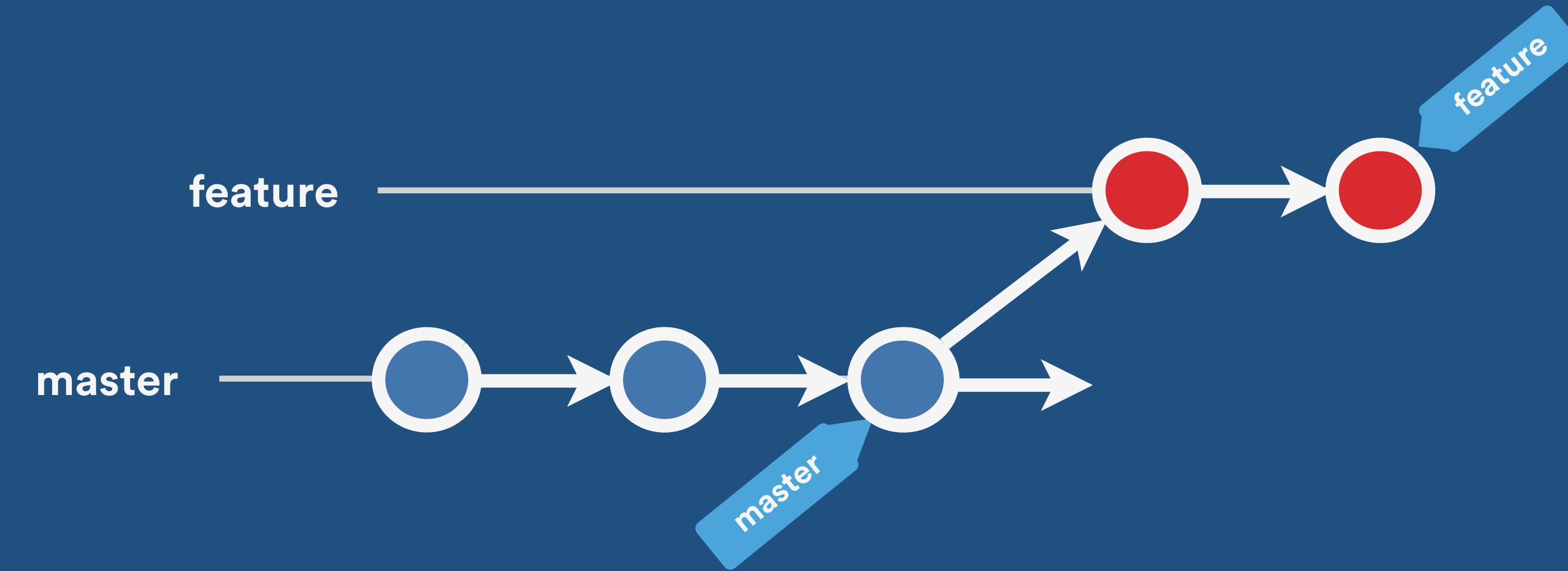
```
$> git cat-file 3680d8c8fd182f97cb0e75045e2fed5c7b7613ed  
tree f362c42032aff677c1a09c3f070454df5b411239  
parent 49a906f5722ad446a131778cea52e3fda331b706  
parent bd1174cd0f30fe9be9efdd41dcd56256340f230e  
author Marcus Bertrand <mbertrand@atlassian.com> 1409002123 -0700  
committer Marcus Bertrand <mbertrand@atlassian.com> 1409002123 -0700
```

Merge branch 'foo/mybranch'



What is a fast-forward merge?

It will just shift the HEAD tag



What are ‘merge strategies’?

git has breadth of choice on
how to merge changes!

resolve
ours

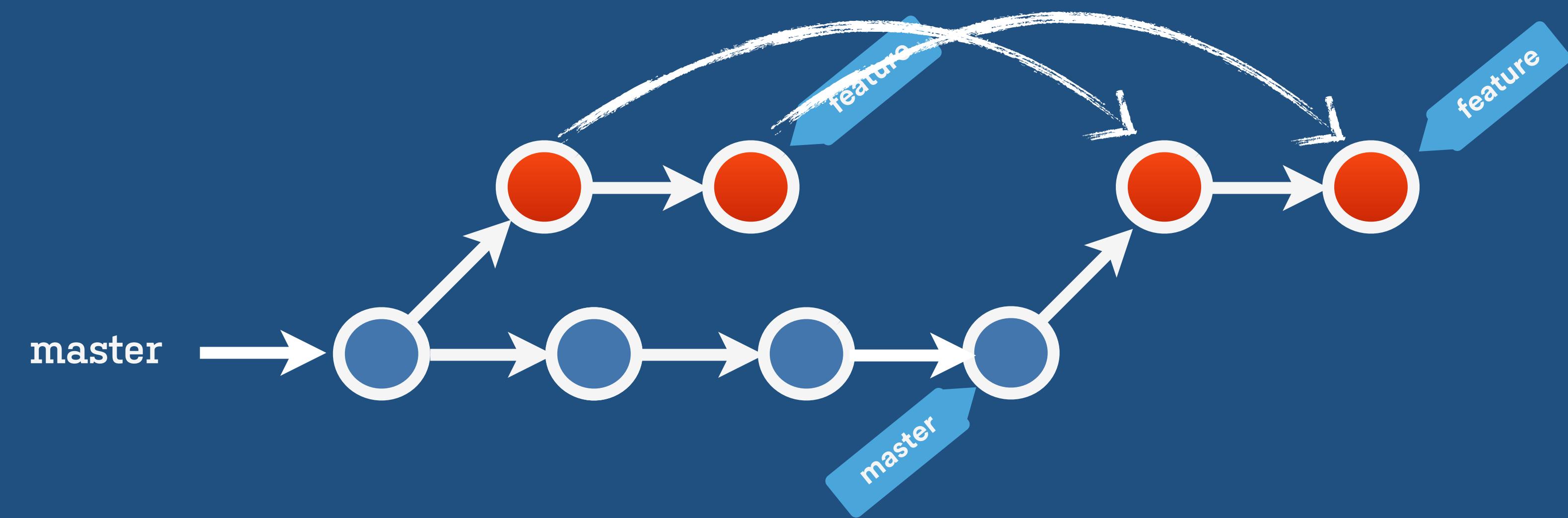
recursive
subtree

octopus
custom?



Rebase

It's a way to replay commits,
one by one, on top of a branch



Getting out of trouble

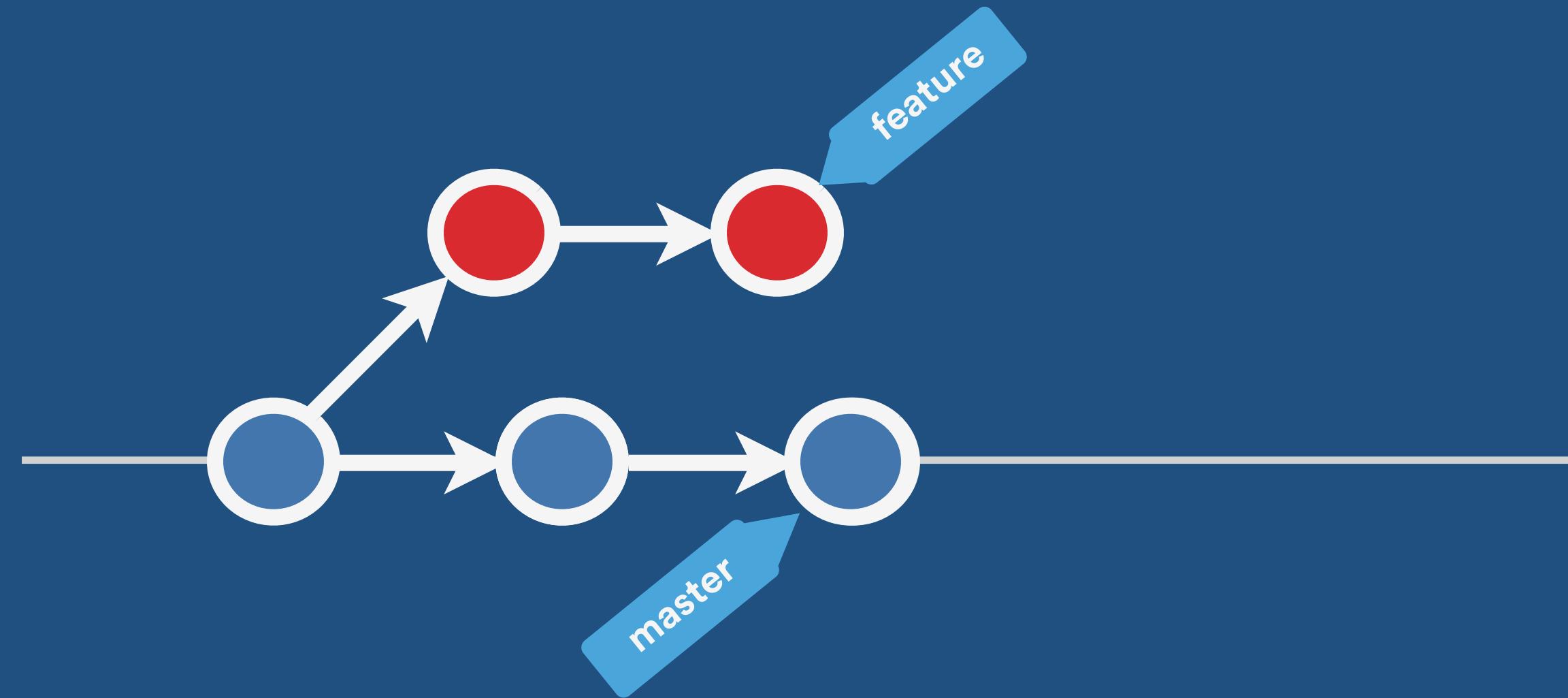
A blurry, high-motion photograph of a person performing a backflip or roll over a rocky cliff edge. The person is wearing a dark t-shirt and shorts, and appears to be wearing a helmet. The background is a steep, rocky cliff face with sparse vegetation.

Professional stunt. Do not attempt.

reset —hard ohshit

—hard removes all staged and working changes!

```
$> git reset --hard feature^ # '^' means 'parent'
```



Reflog!

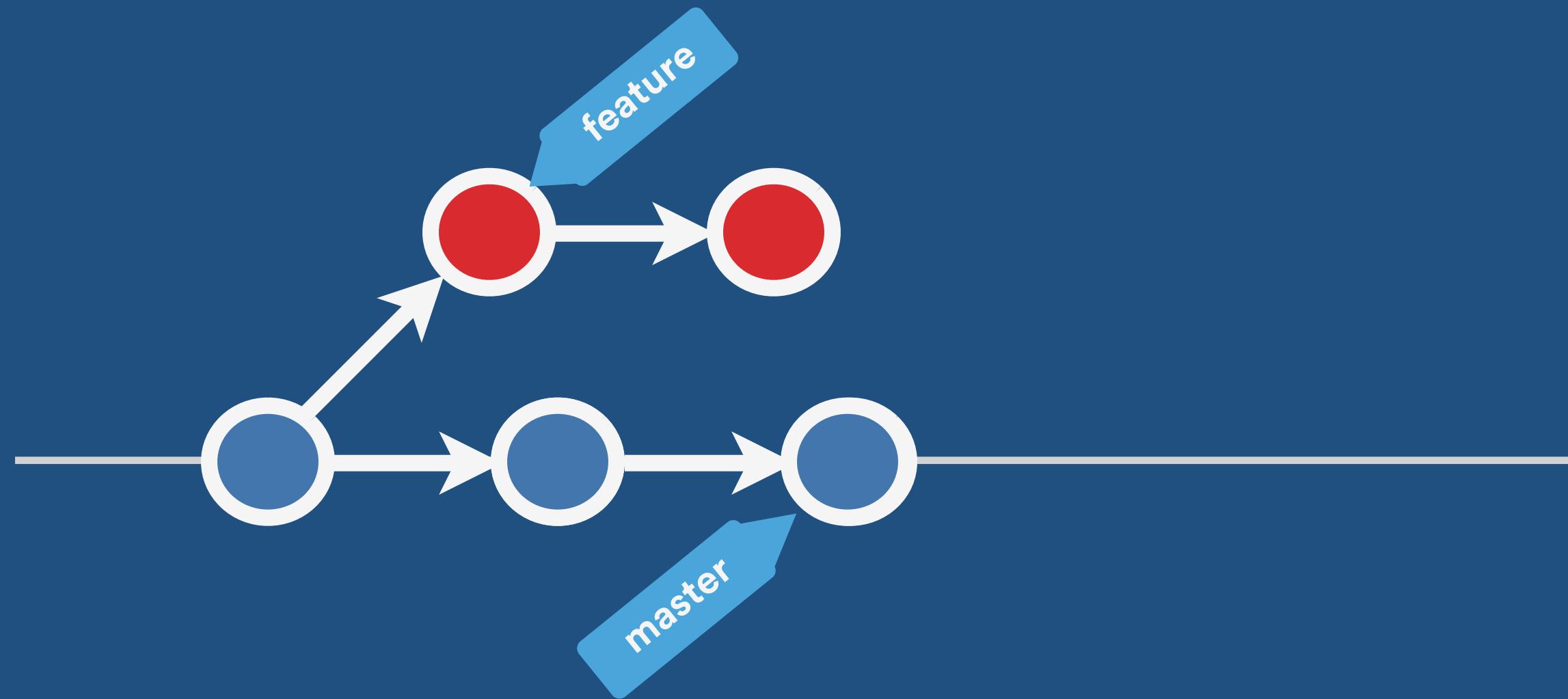
```
$> git reflog
0c35628 HEAD@{1}: reset: moving to HEAD^
6cc6637 HEAD@{2}: commit: Add B
0c35628 HEAD@{3}: merge: Merge made by the 'recursive' strategy.
e0c0d65 HEAD@{4}: cherry-pick: A
80bb854 HEAD@{5}: checkout: moving from alpha to master
5044136 HEAD@{6}: commit: A
80bb854 HEAD@{7}: checkout: moving from master to alpha
80bb854 HEAD@{8}: commit (initial): 1
```



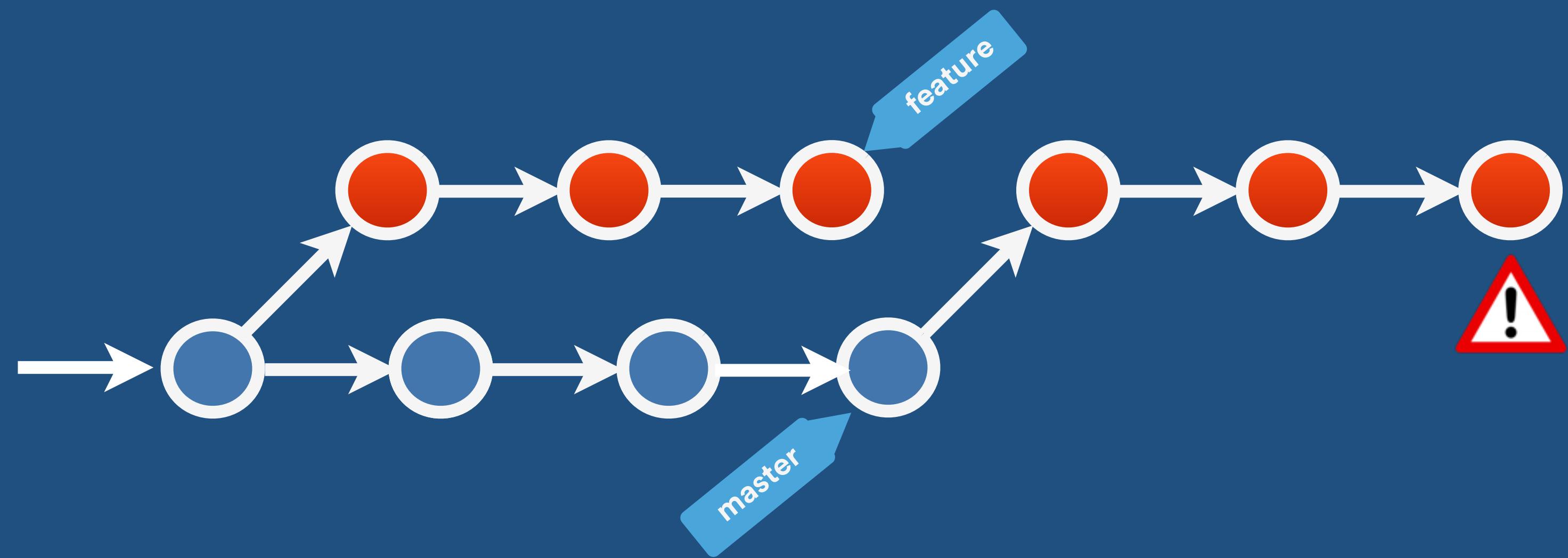
Reflog+Reset = Redo

Reset back to our commit!

```
$> git reset --hard 6cc6637
```



Rebase Broke The Build!

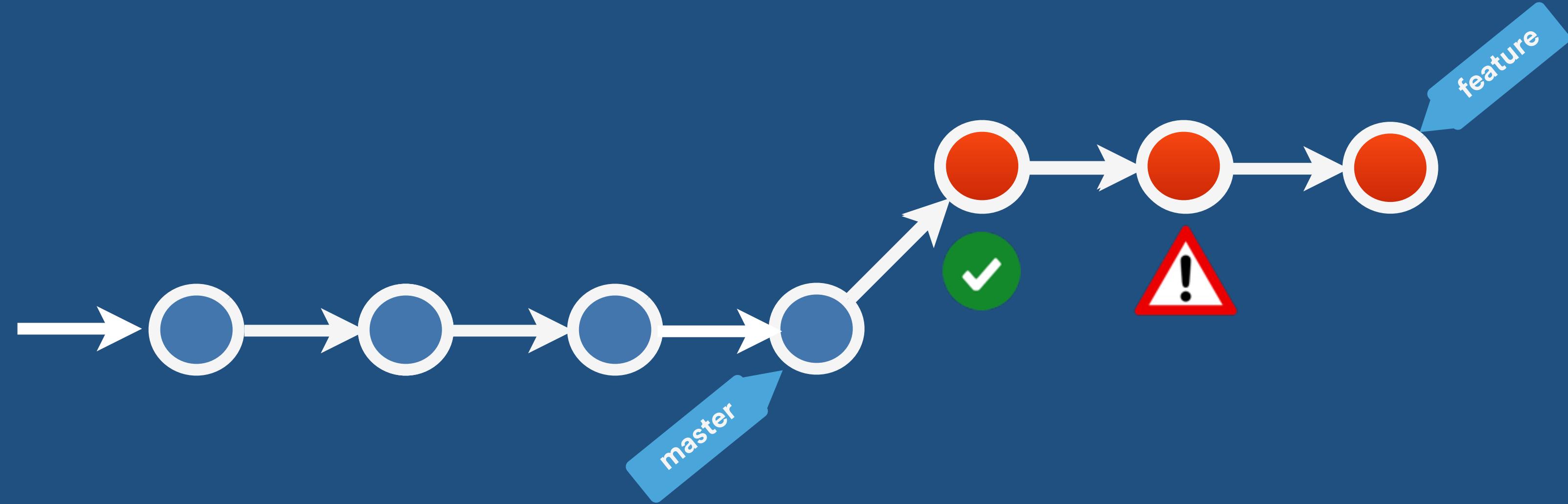


Fix with rewind/replay

Reflug to reset.

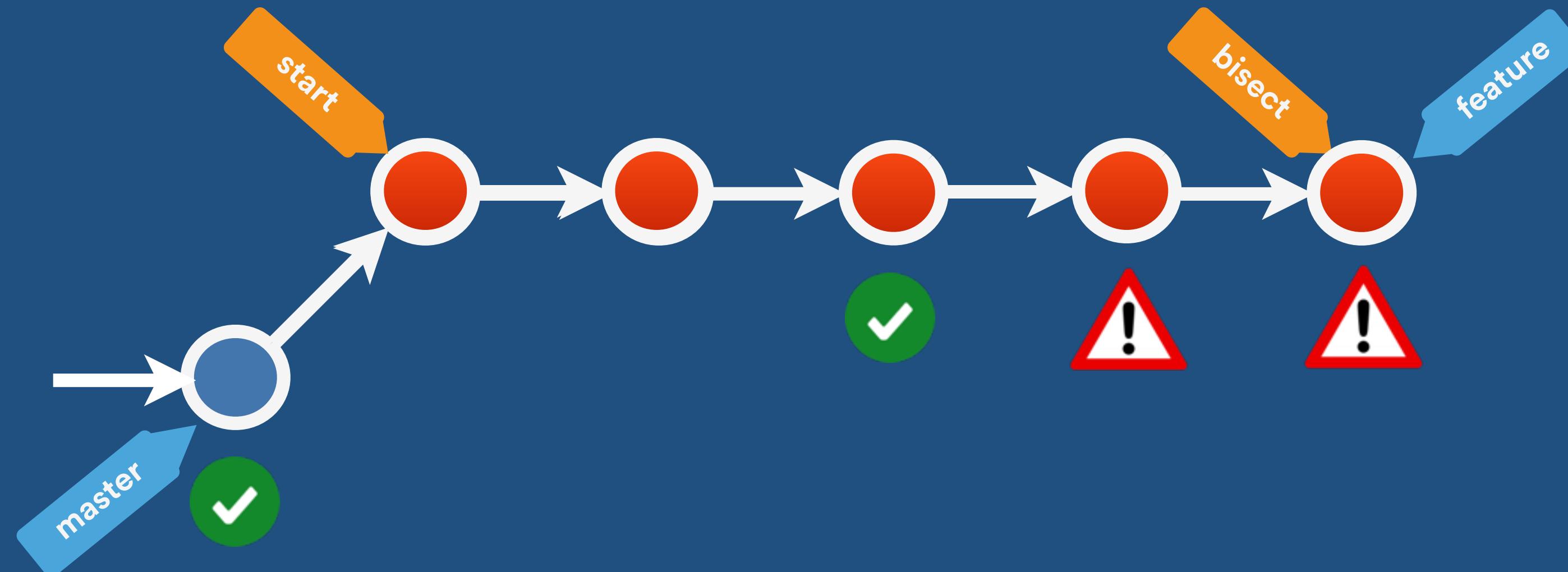
rebase --exec to test each step.

```
$> git rebase master --exec "make test"
```



Fix in place: bisect

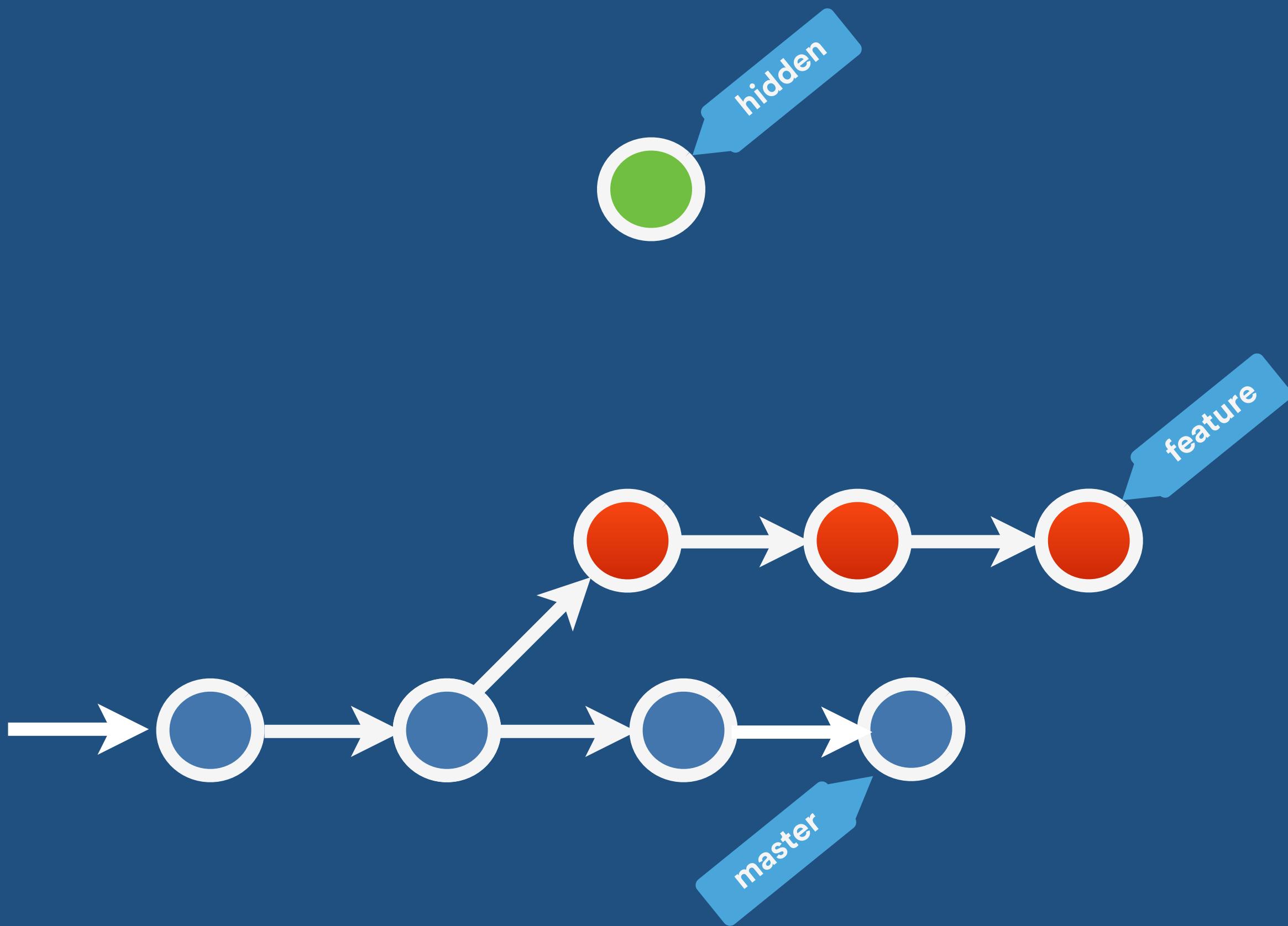
```
$> git bisect start  
$> git bisect good master  
$> git bisect run make test
```





Powers of invisibility

Hiding files in Git



Hiding files in Git

```
$> git hash-object -w hidden.txt  
929cb9f0f8bde8e8b9dc4a69cf3459d2326191c8  
$> git tag hidden 929cb9f0f8bde8e8b9dc4a69cf3459d2326191c8
```



Retrieving

```
$> git cat-file -p hidden
```

Recipe for Poitín

- * Boil 5 gallons of water and pour it over a mix of ten pounds of rolled oats that has had a pound of 6 row barley ground and mixed in.
- * Allow this to sit until it is cool enough to add yeast, then add a dry ale yeast.
- * Ferment until dry.
- * Double distill in a potstill
- * Don't age, drink it white.



Actual use...

```
$ ~/git> git cat-file -p junio-gpg-pub
```

```
-----BEGIN PGP SIGNATURE-----
```

```
Version: GnuPG v1
```

```
iQIcBAABAgAGBQJUbV7AAoJELC16IaWr+bLvjYP+wSOG8rp1y77ExHDJQj7HBLm  
9PVQIb70tkiBxAGUpVNNbaoJQBuMAgFdT4Baj8CIo2jdzDeeqbjtcdy/XsHZ3IMl  
7VRm9cI8Veh5nPc9U3z6XYTMmDhxxSsnCONQECYYFEJ5QGhnwkpOUTDYPRmnghWu  
4JSDyWX0RLNXVPZkoP/zwVbv74Exx1Gz24p6fPlc4iETrKNHzzSM07CPK1AmXRHT  
Ozh/NXfiE30ZSFDUNKkzHQJHtyFEPpVw/aUIyObvOcSYy6ZxQ+56U9h24D1mW8kY  
w8BPdhzQ5d9nB/zpX83/cqsv6IXsK62ls8Qj51MfNM902D0pghVBInjR2xekWx+b  
A05VMaRiC+hgHvyzAM2nXuDfTUssd9ryYIx/p1fw0gcKZ41S3HzHTiXYqRqd404I  
1kFlWL11no7od+T9T2nap5wfdL1iQpMxxZFLsqH67LzAvHxg/NjtIMXDxUfo++tV  
qBR153FxcJX8A0cQo3zY7WotbmgS+8JPrOn3XvjXN0lz9HAxR2+orxhEC3pTVfeC  
/aEZzs+ub9pV6ZQzVdfHzRZbtN4Zw2ycW+EoXvlUCwNy72efaBRpKjQ14zJHKfD  
LAwljdrenm8X6RjBKd/vPyTPSHx1AtSoXqseVw8Ly+qqoXnhsh2N0Xe20zx3D0lk  
PtscHvD+diDMRLbLgK/k  
=Sxgg  
-----END PGP SIGNATURE-----
```

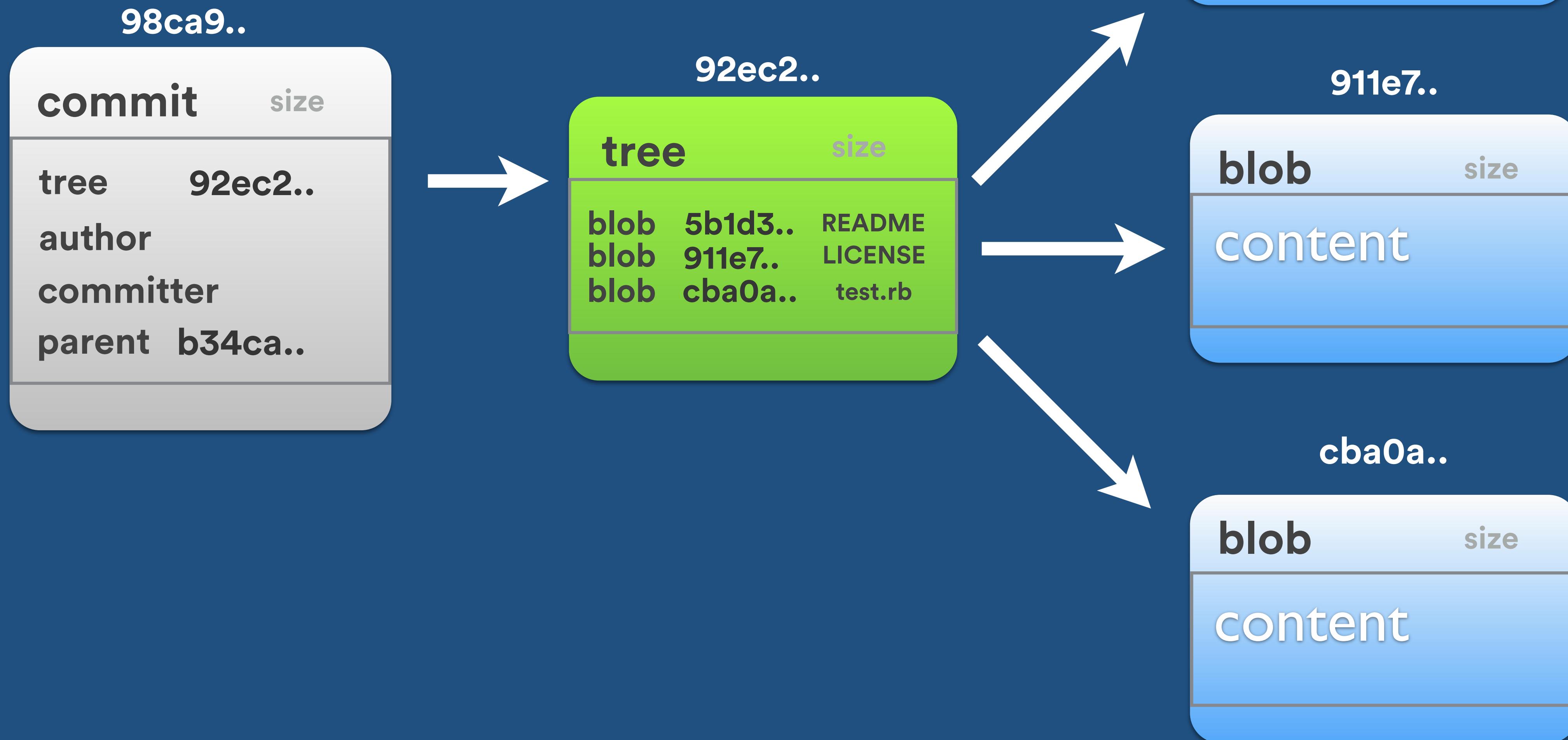




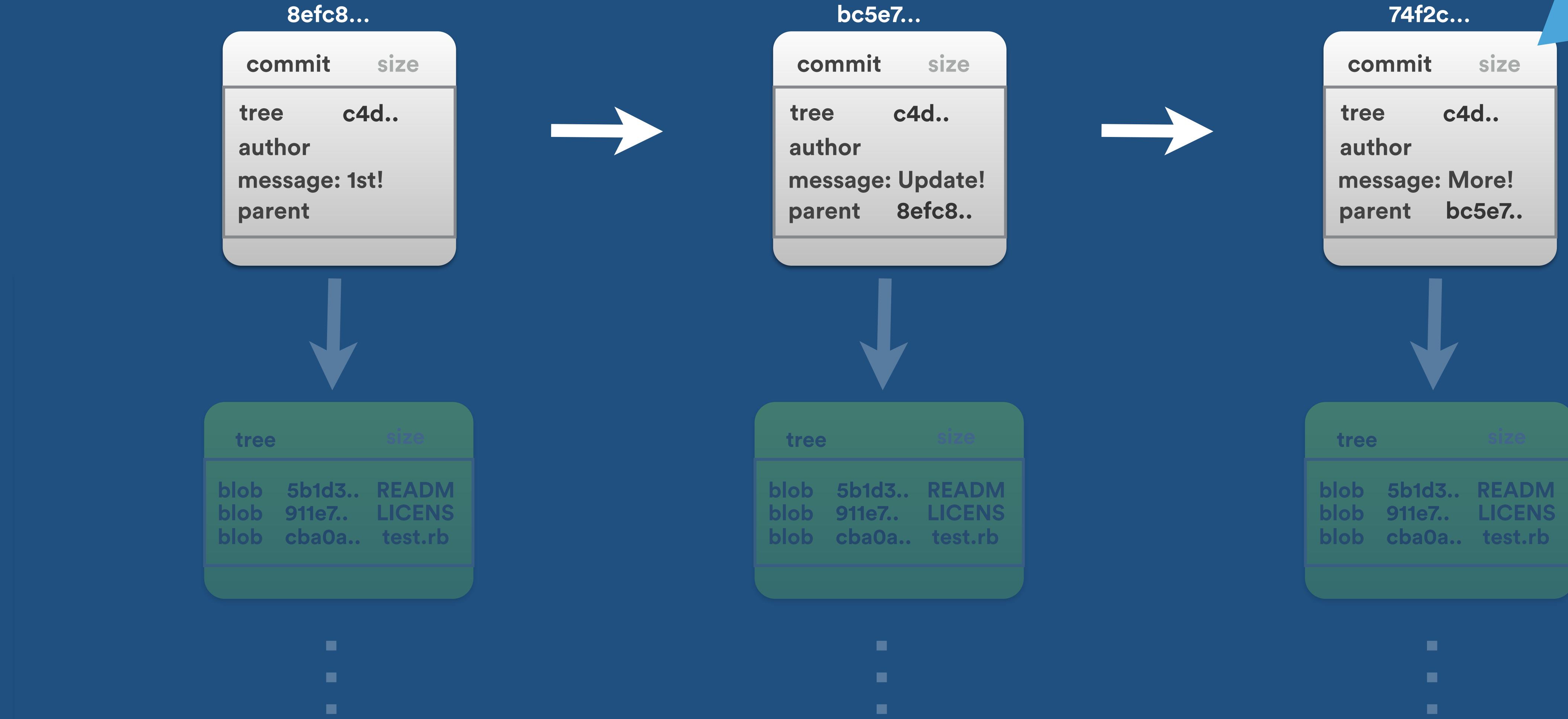
Out of Time



git data model



git data model



A woman with dark hair, wearing a dark jacket over a green top, stands in a hallway. She is pointing her right index finger directly at the camera with a serious expression. In the background, there are doors and a large window showing a night view of trees and lights.

So now we're samrt!



Thank you!



STEVE SMITH • DEVOPS ADVOCATE • ATLASSIAN • @TARKASTEVE



Please

**Remember to
rate this session**

Thank you!



follow us @gotochgo