

Tak.



**Tak.
Po prostu.**



Let's move the Java world!
GeeCON

lunar logic
polska



@ktosopl



github.com/ktoso



blog.project13.pl



**Masz pytanie?
Nie czekaj do końca prezentacji!**



Poproszę...
„Pytanie do publiczności“

Ważne słowa

(-: WAŻNE :-)

System kontroli wersji
Scentralizowany czy Zdecentralizowany
(CVCS vs DVCS)

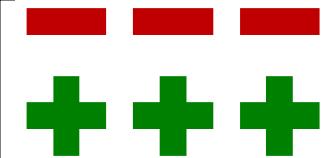
Zespół
Współpraca

Zmiany
Diff („różnice“)

Integracja zmian

Słowniczek:

Commit
Snapshot
Konflikt
Branch



git-ninjitsu



Config

Init

Clone

Add / Stage

Reset

Status

Commit

Log

Gitk

Git gui



Stash

Branch

Checkout

Merge

Stash

Diff

Fetch

Pull

Push

Daemon

Rebase

Add --patch

Grep

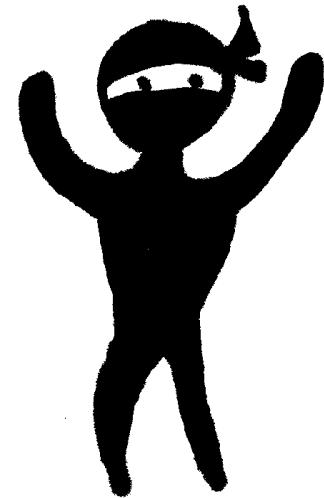
Bisect

Blame

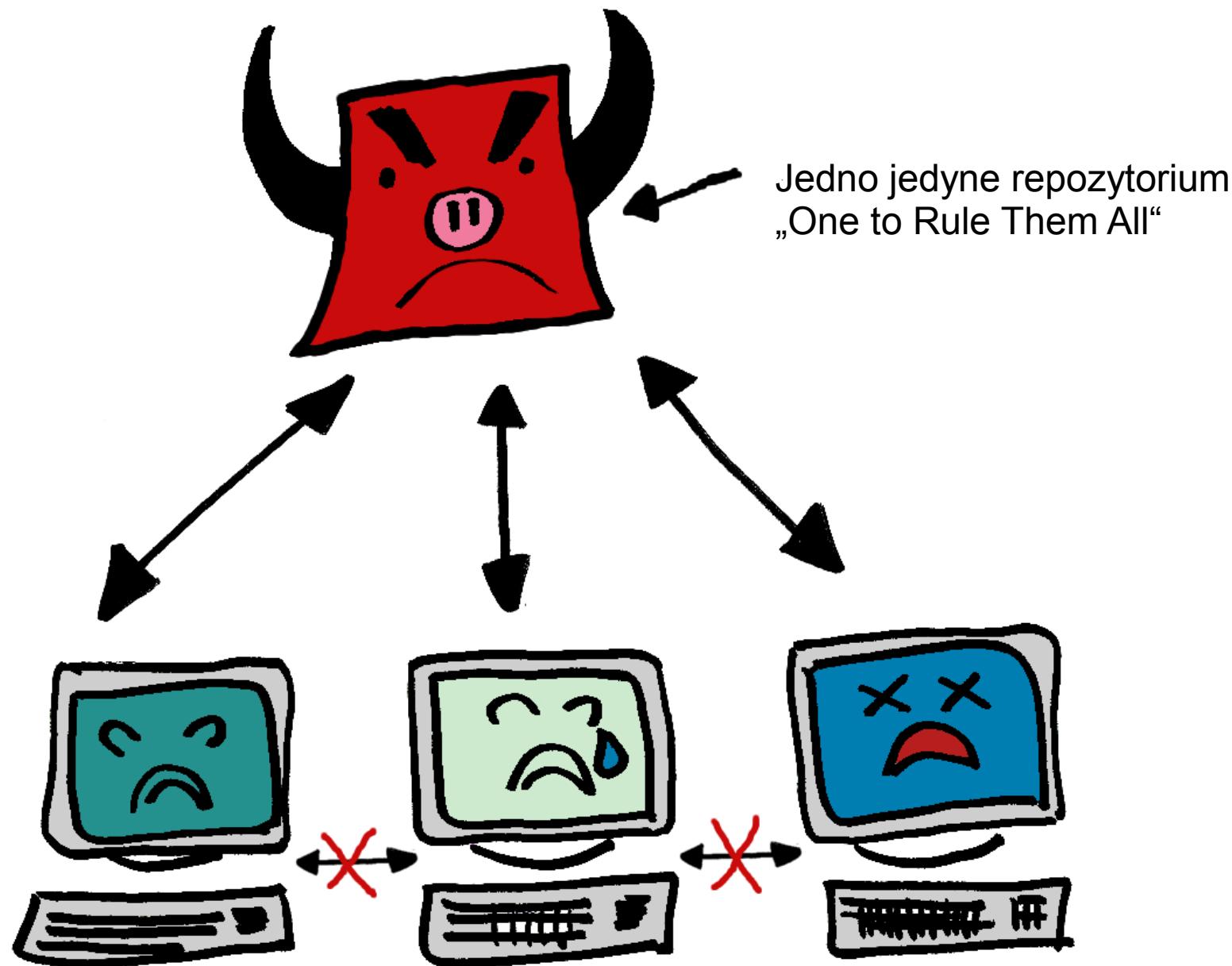
Gc

Format-patch

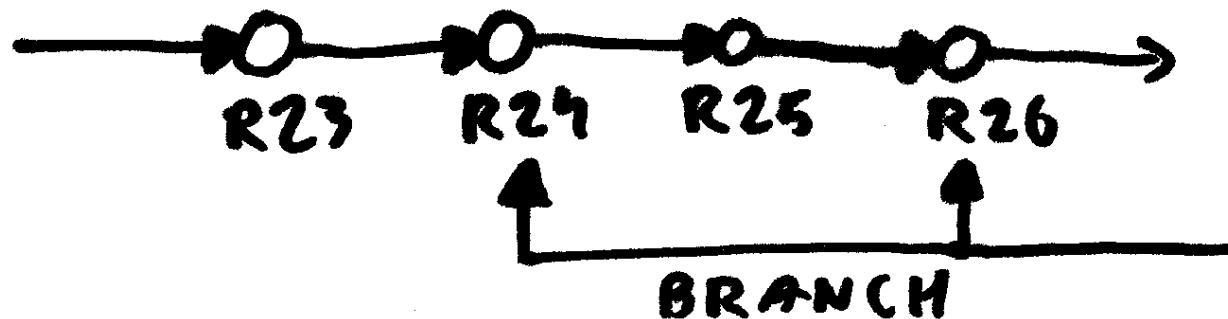
Svn



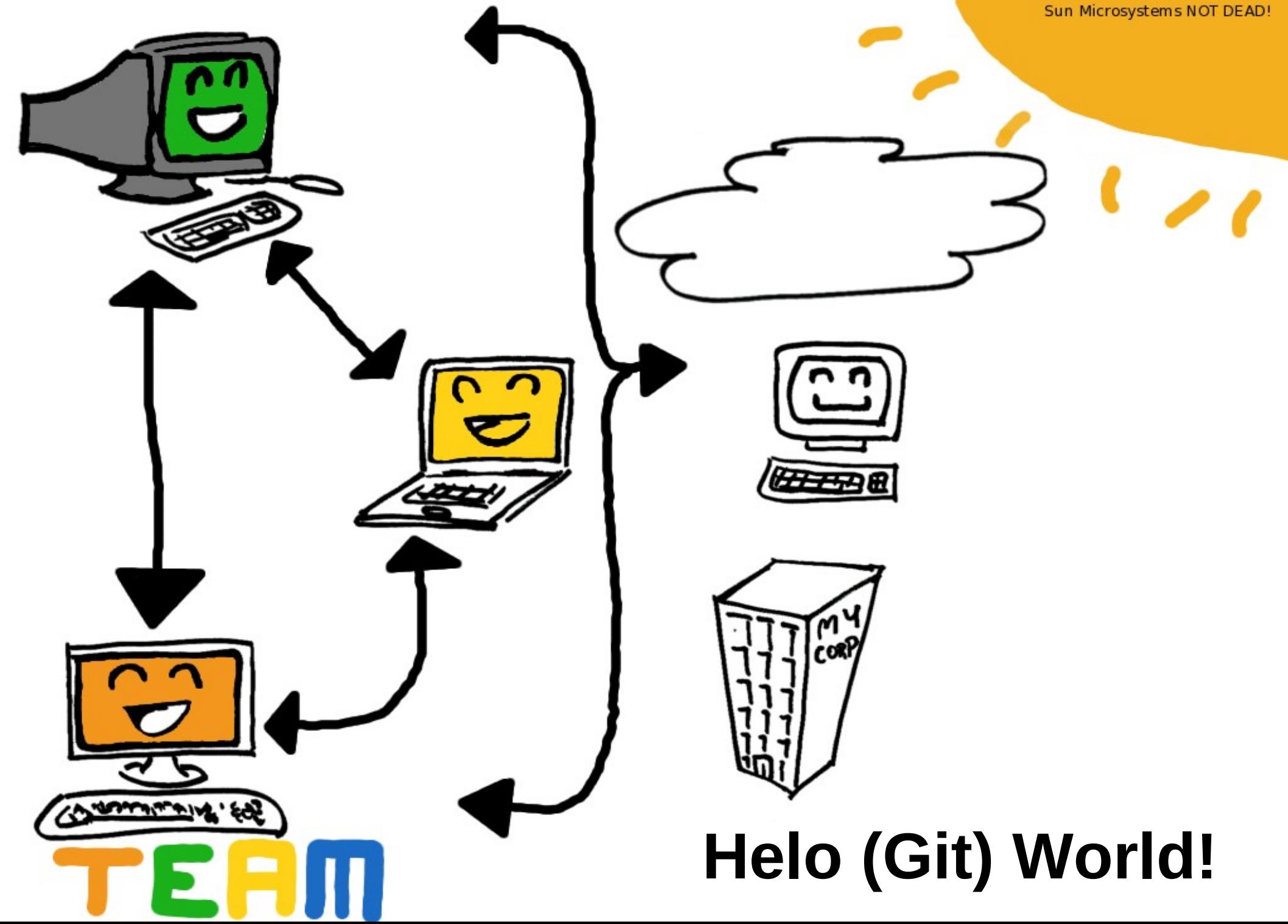
Centralized SCM World



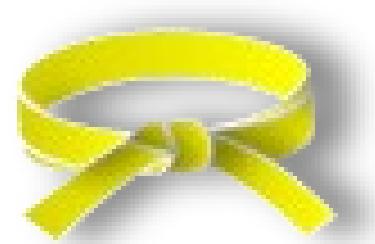
Svn UDAJE branche i tagi



/tags/...
/branches/...
/trunk/...



Hello (Git) World!

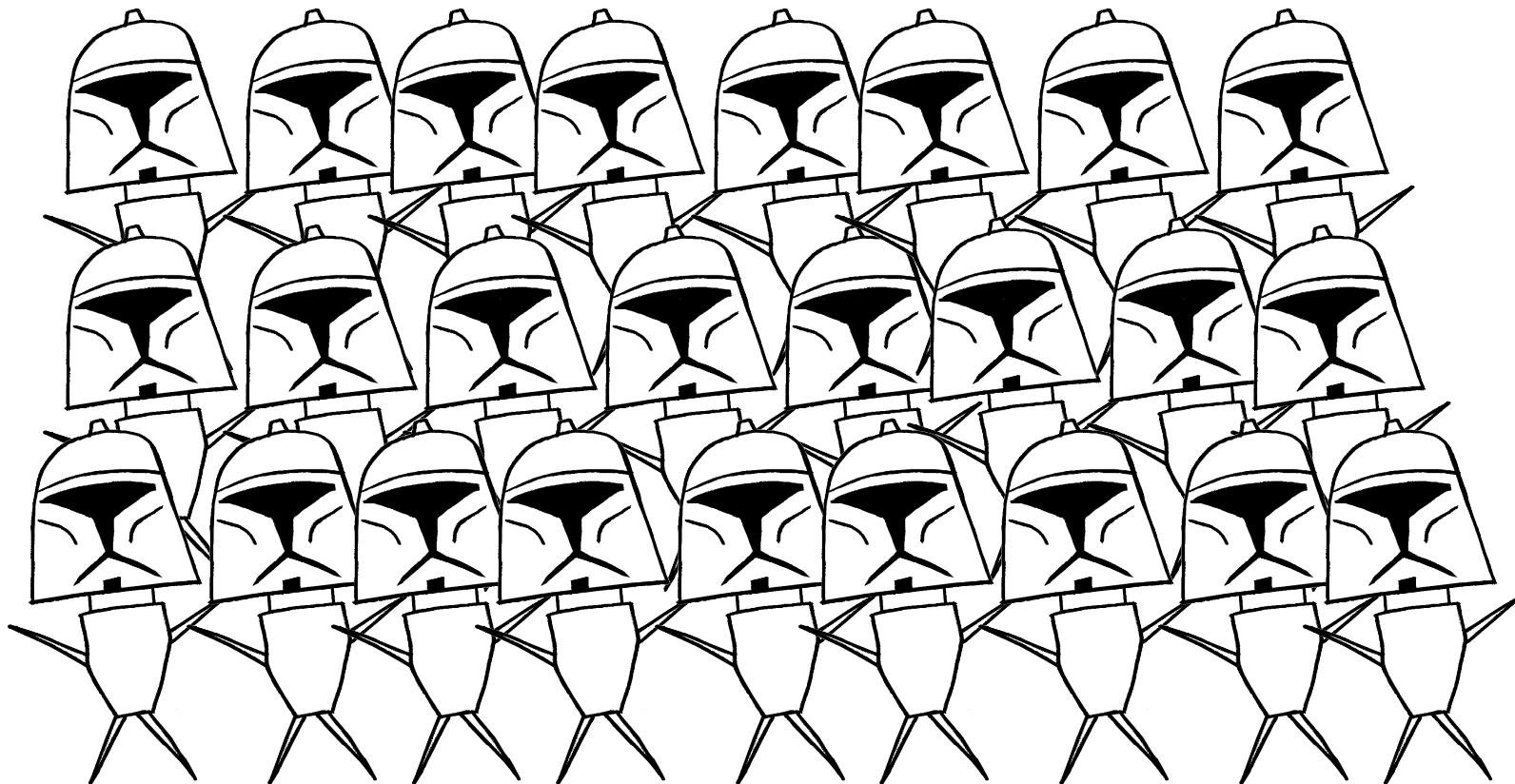


Bardzo skomplikowany setup ;-)

Git init .

(Przyjrzymy się strukturze plików git za moment „na żywo“)

SCM Wars: Attack of the clones



Git clone file:///home/ktoso/coding/intellij

Git clone ssh://ktoso@dynamics/repository

Git clone http://git.project13.pl/pub/repository

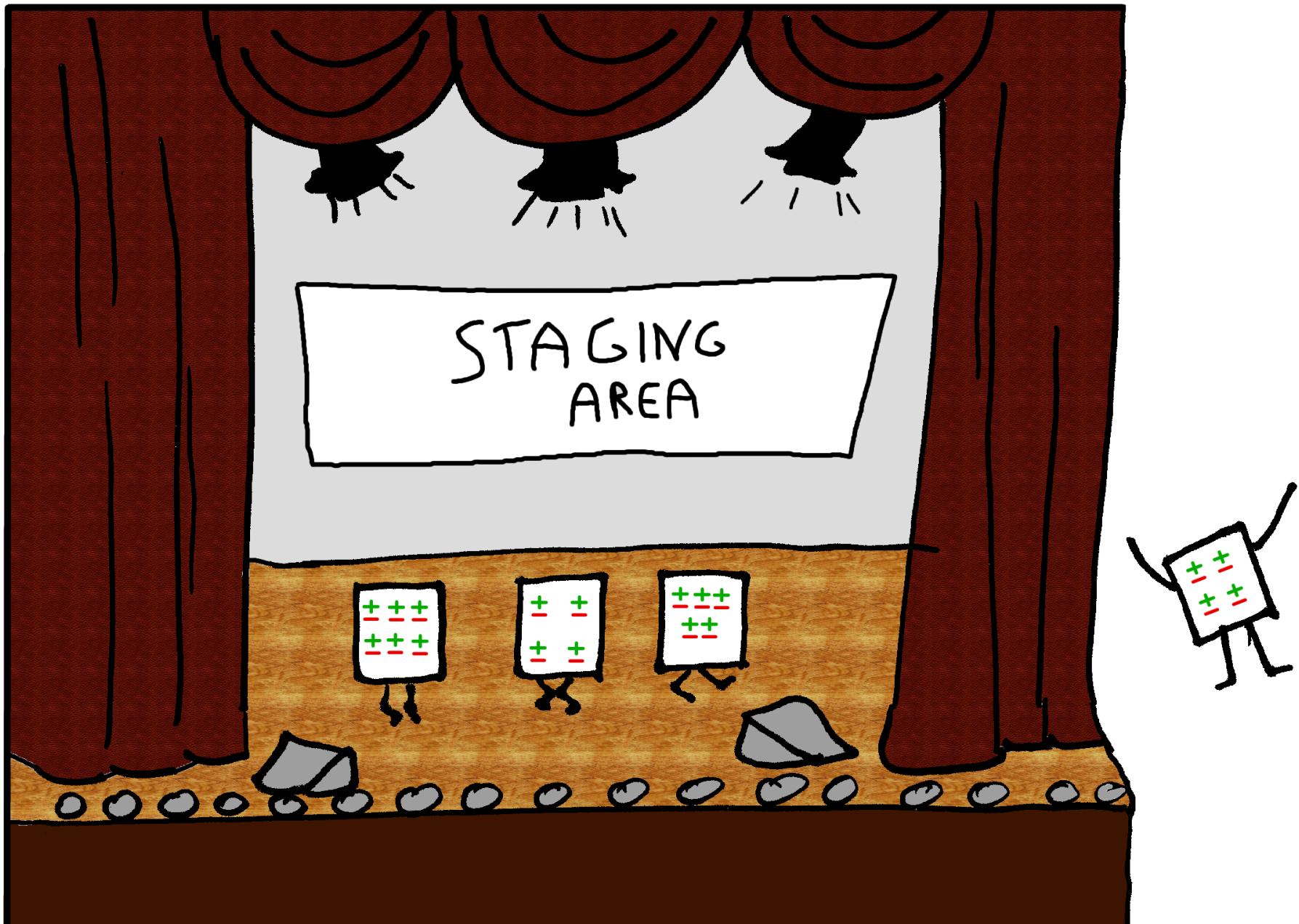
Git clone https://git.project13.pl/pub/repository

Git clone git://git.project13.pl/git/repository.git

./git/index AKA „Staging Area“



./git/index AKA „Staging Area“



./git/index AKA „Staging Area“



indeks ponieważ istnieje plik .git/index
staging area ponieważ idealnie obrazuje jego działanie

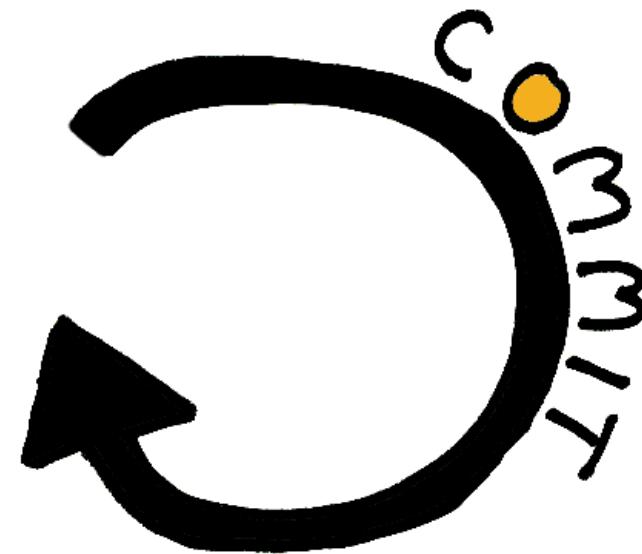
Commit jest LOKALNY



Commit jest LOKALNY



Commit thyself, Lancelot!



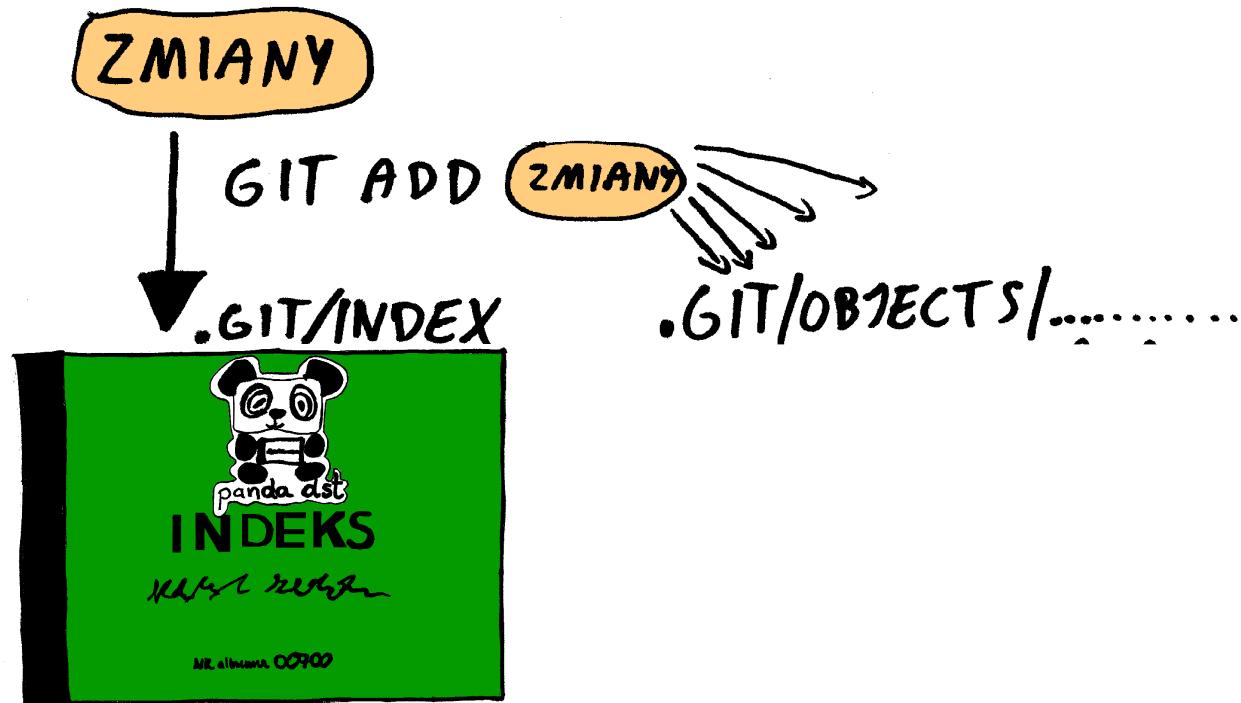
Local (internal details) workflow

ZMIANY

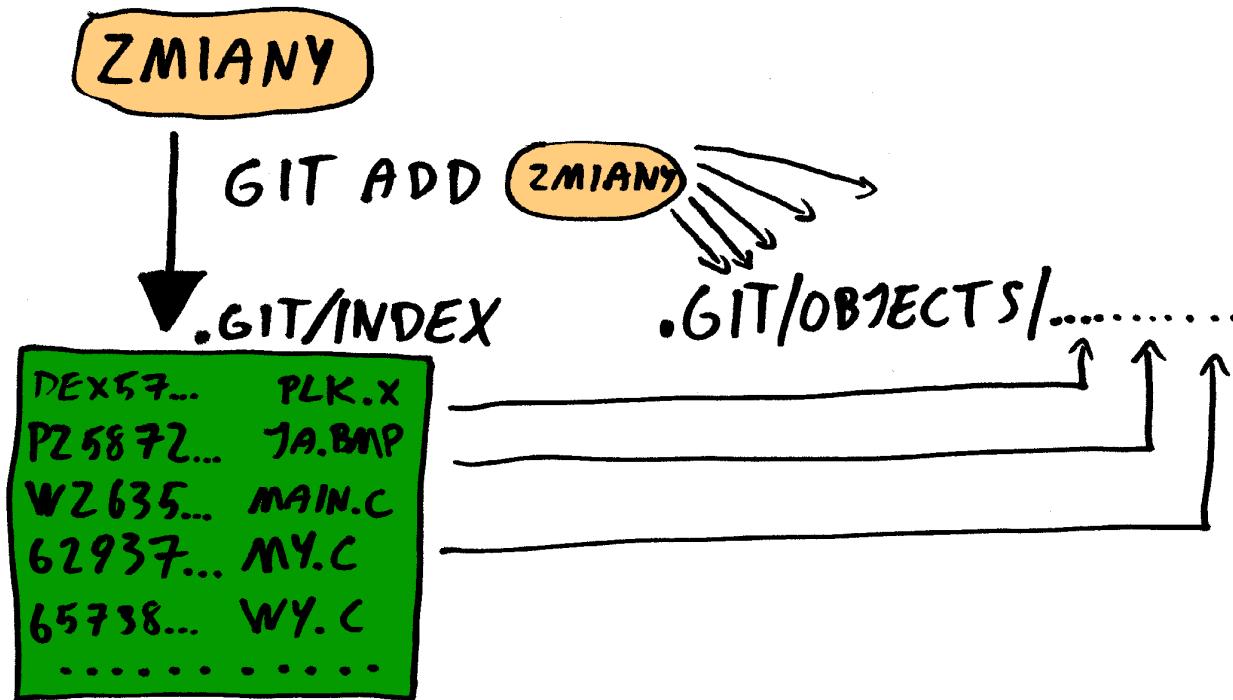
Local (internal details) workflow



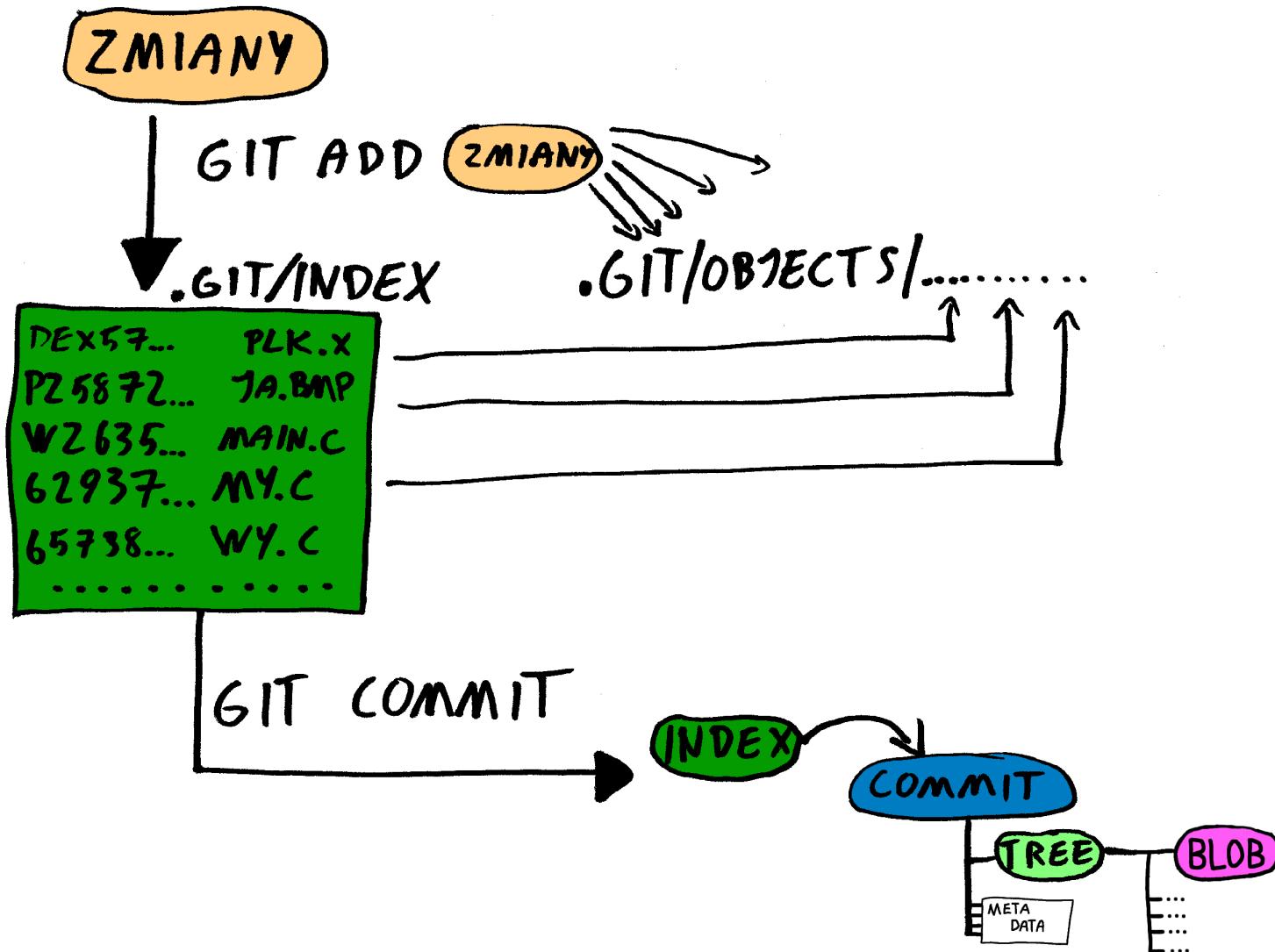
Local (internal details) workflow



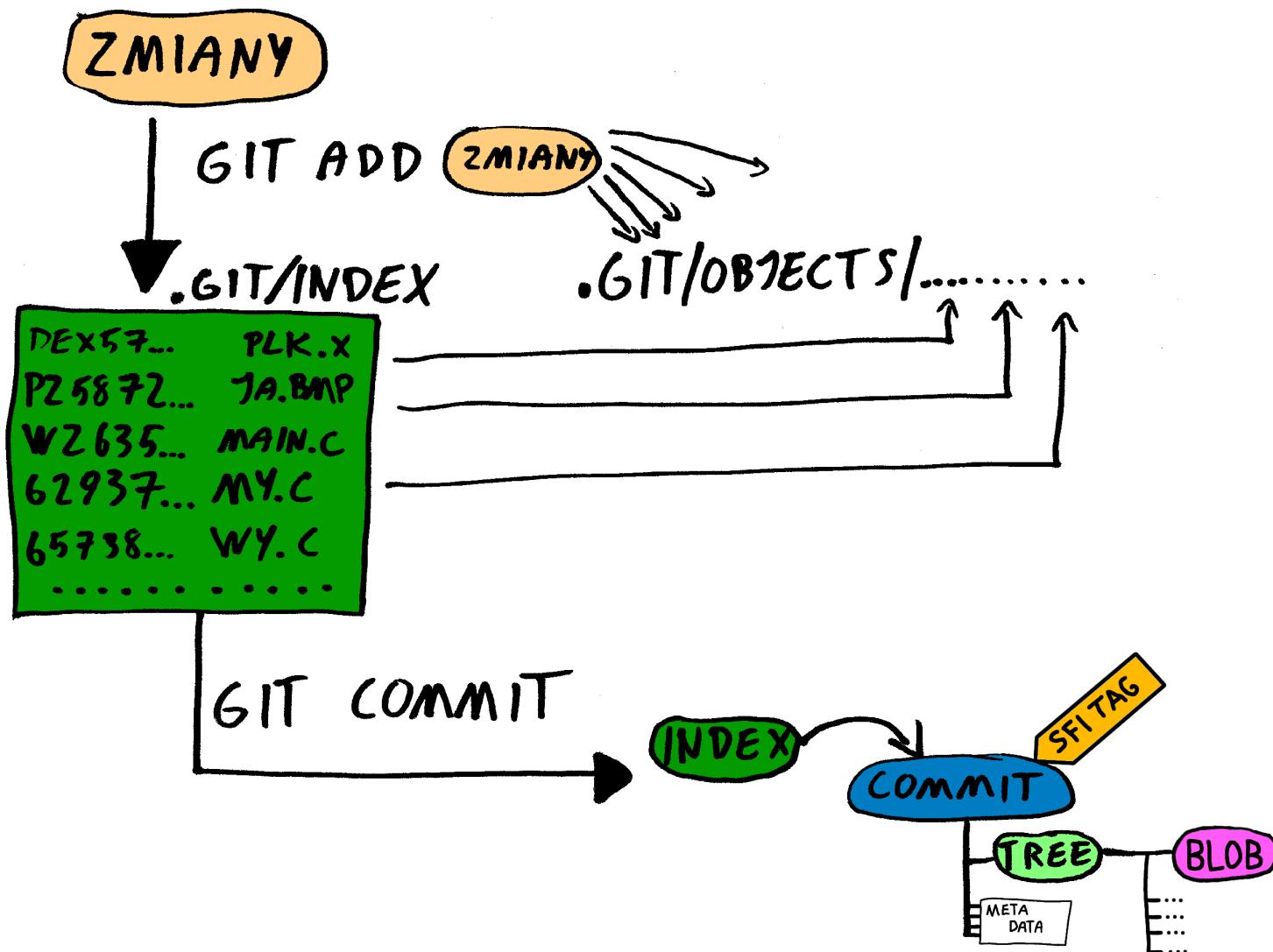
Local (internal details) workflow



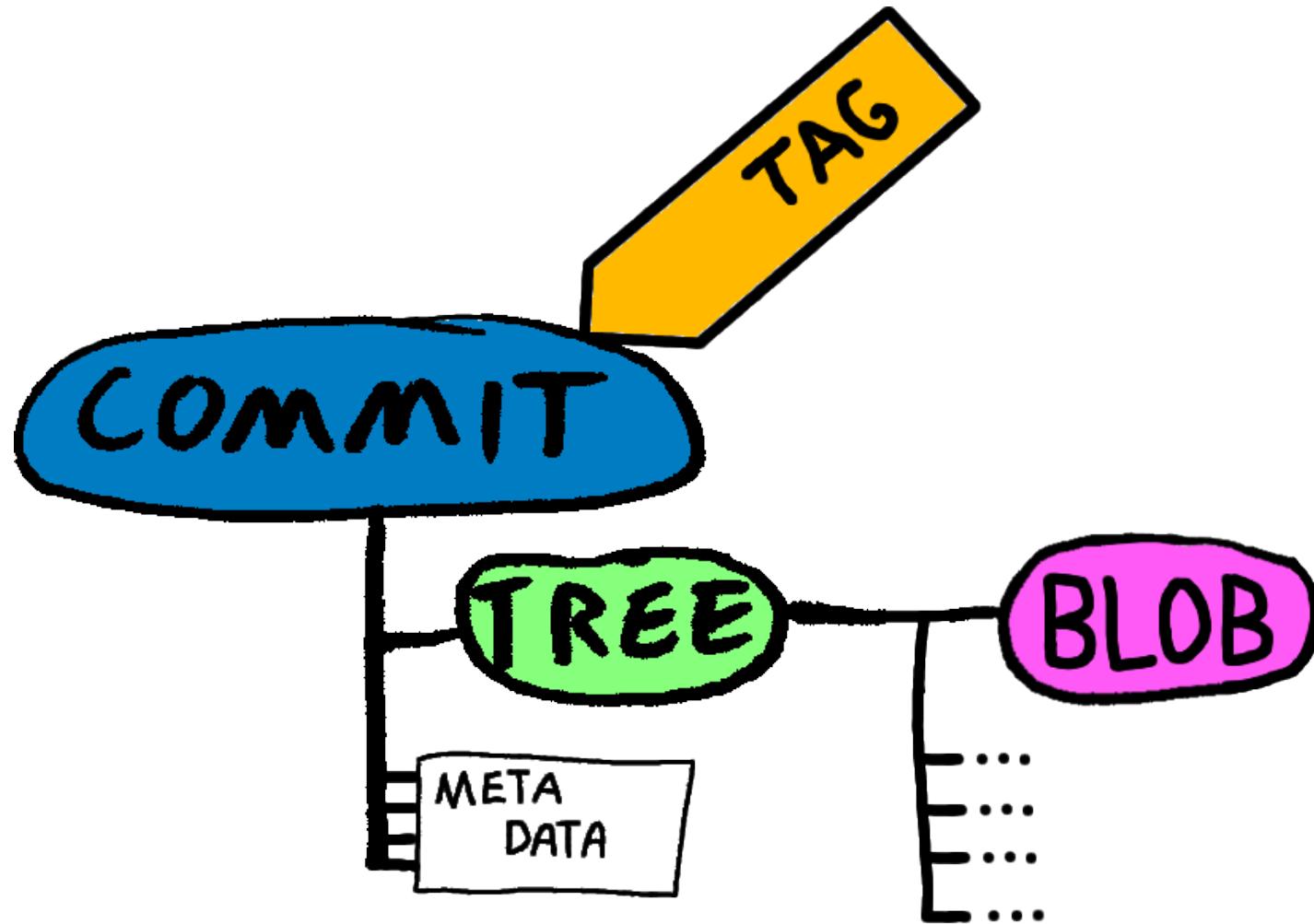
Local (internal details) workflow



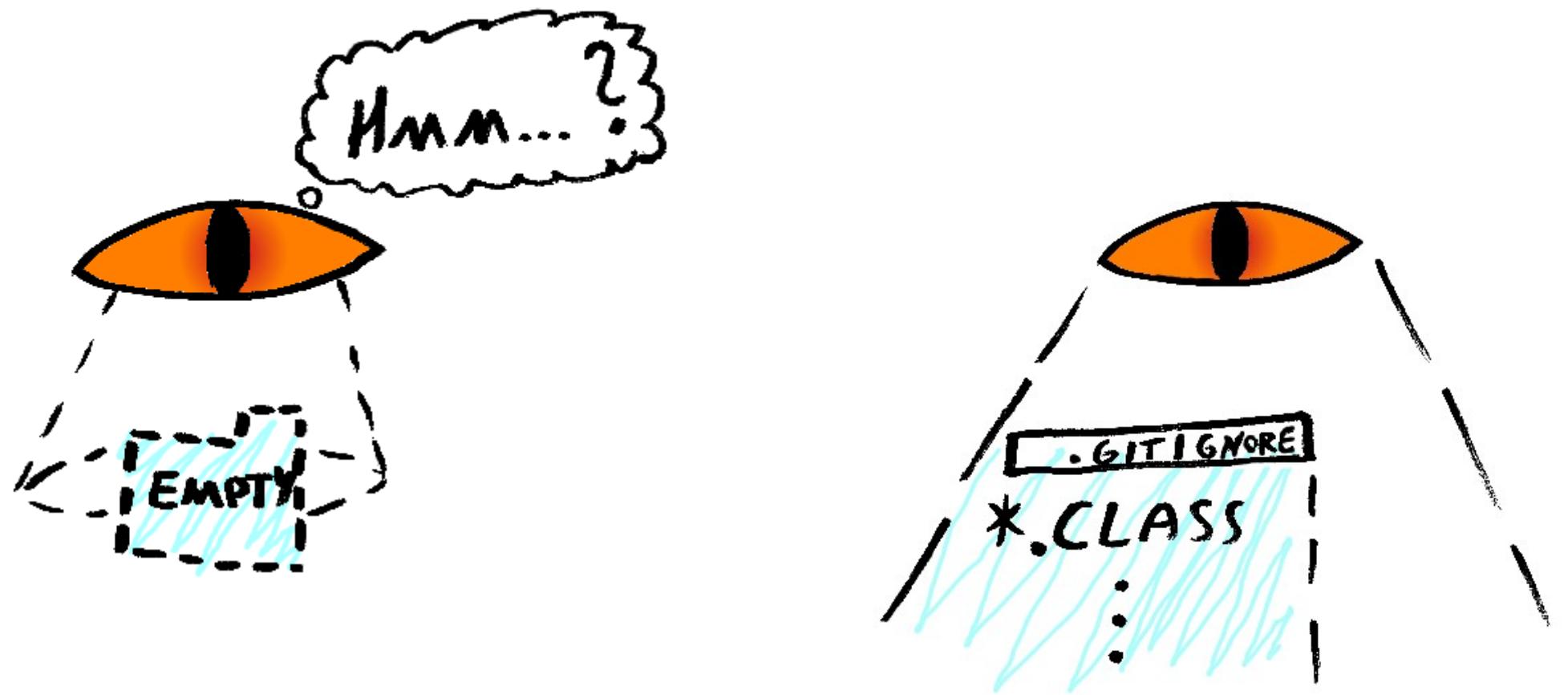
Local (internal details) workflow



ALL Git Objects on 1 slide!



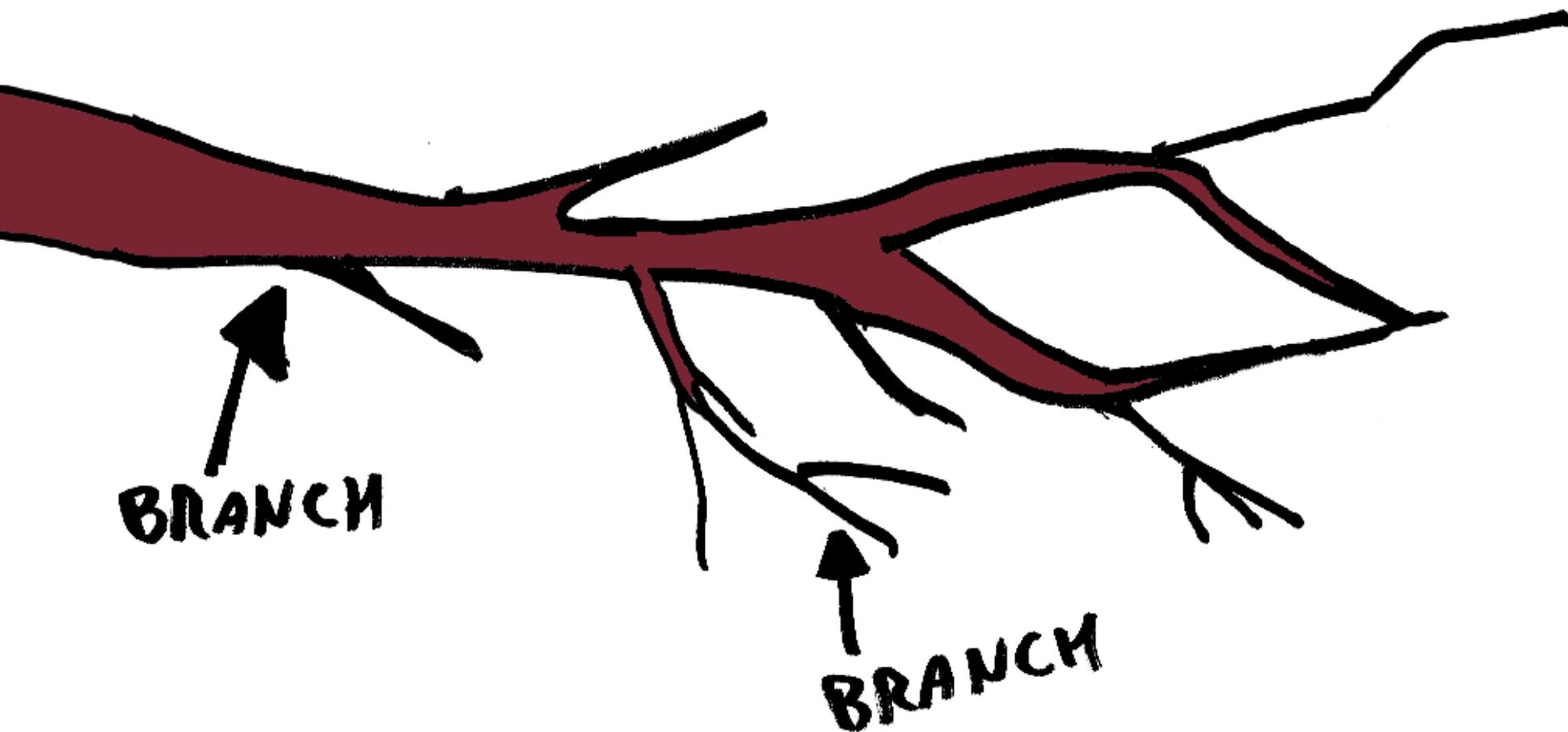
Nie ma treści = nie ma nic.



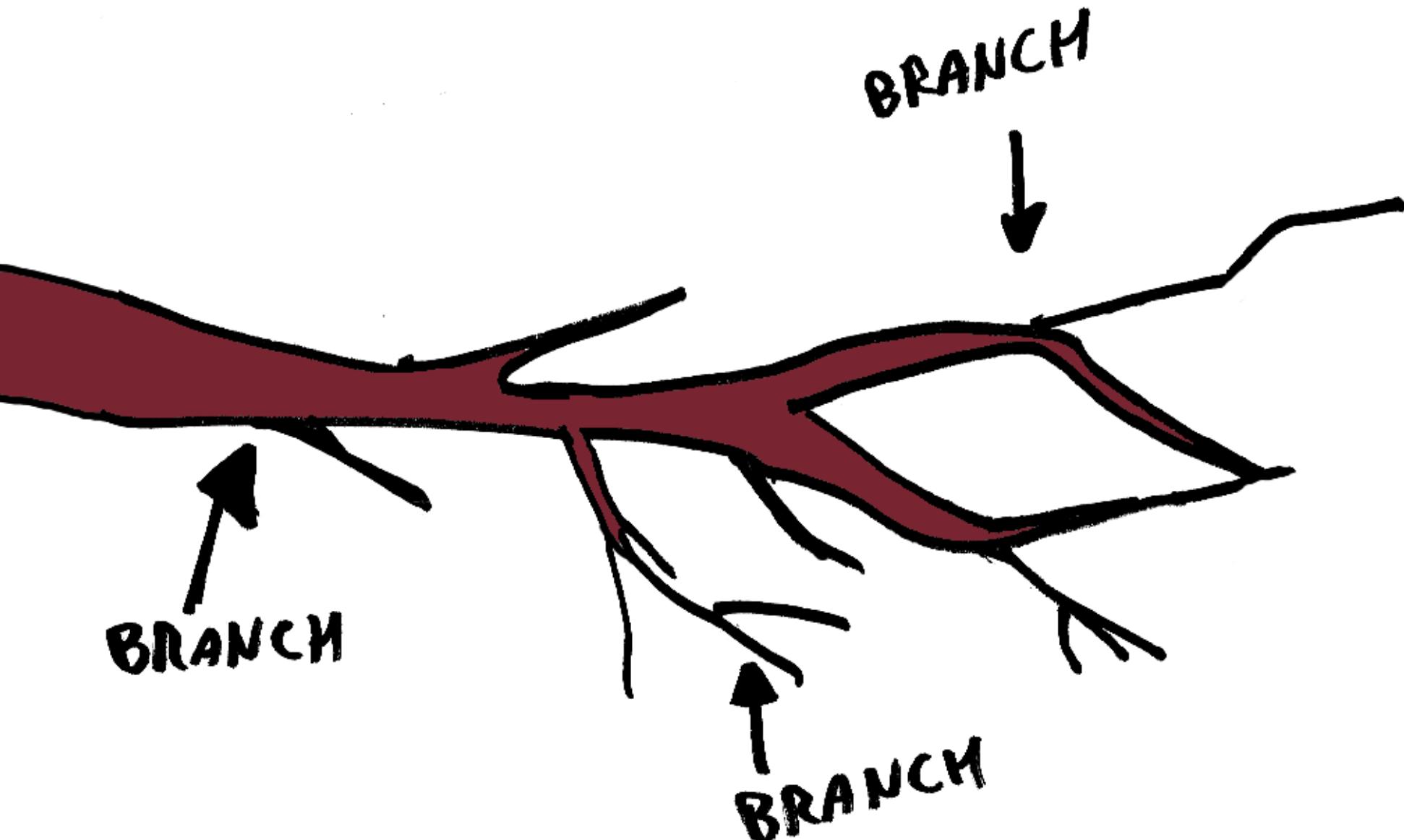
BRANCH + MERGE



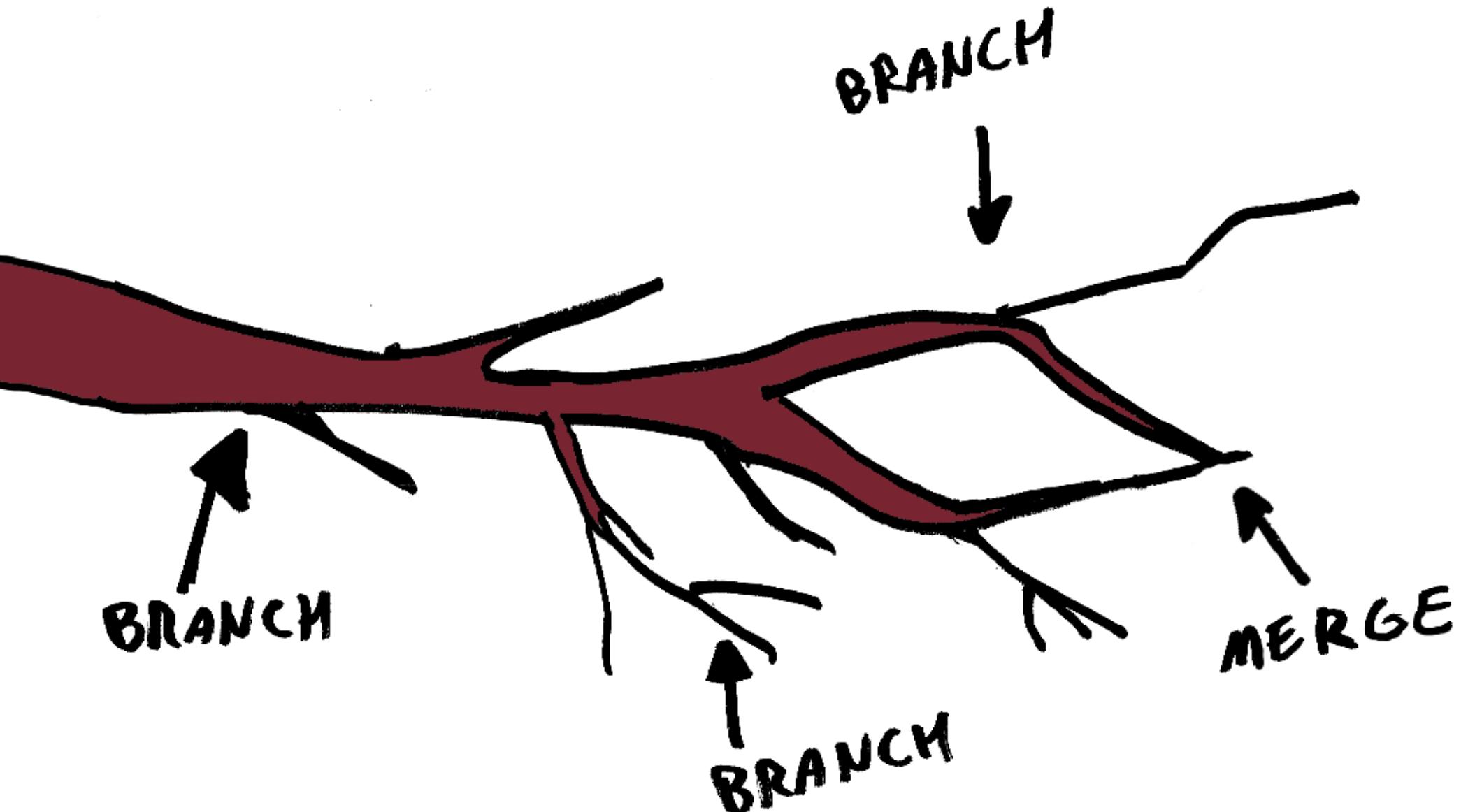
BRANCH + MERGE



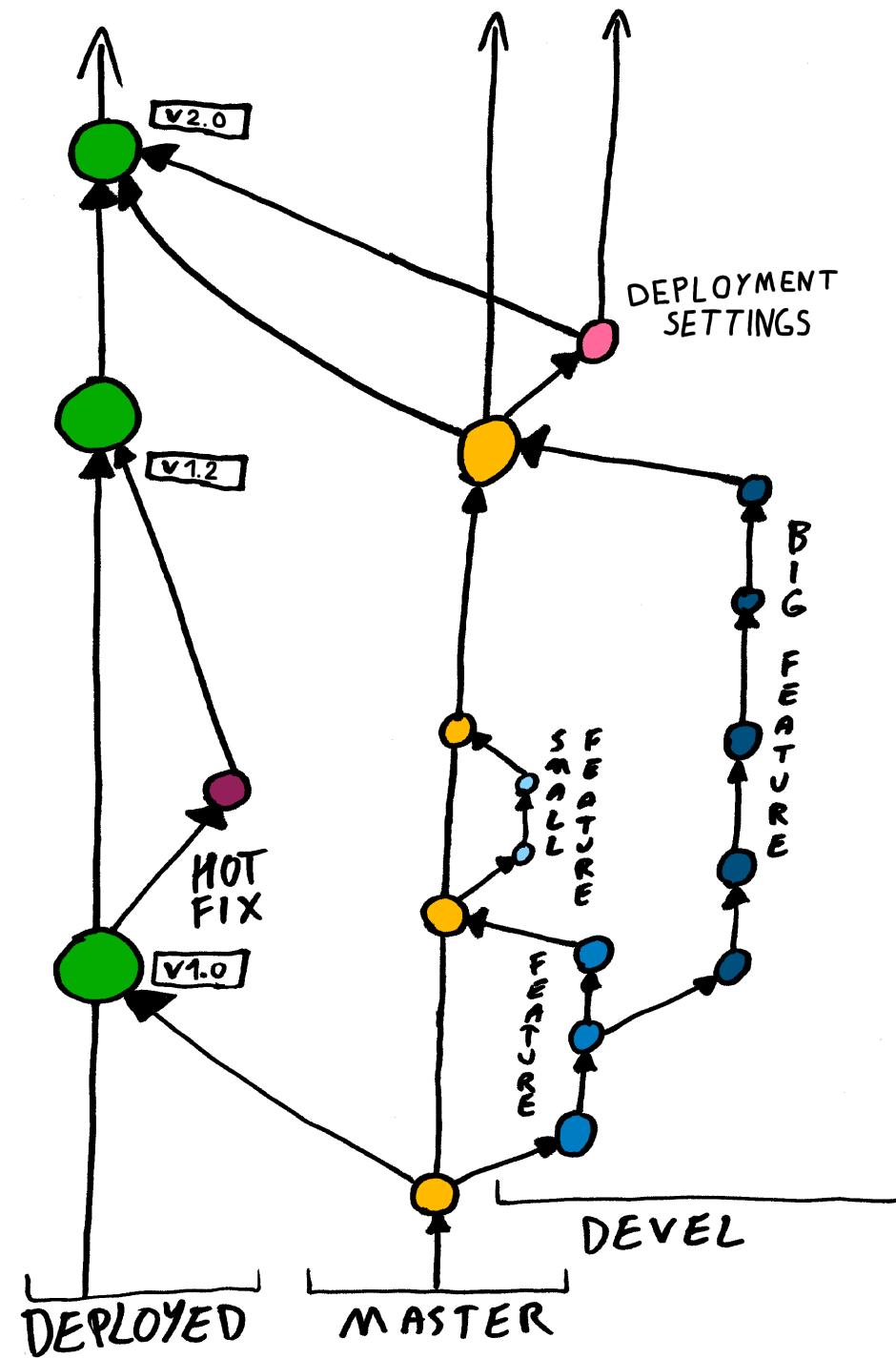
BRANCH + MERGE



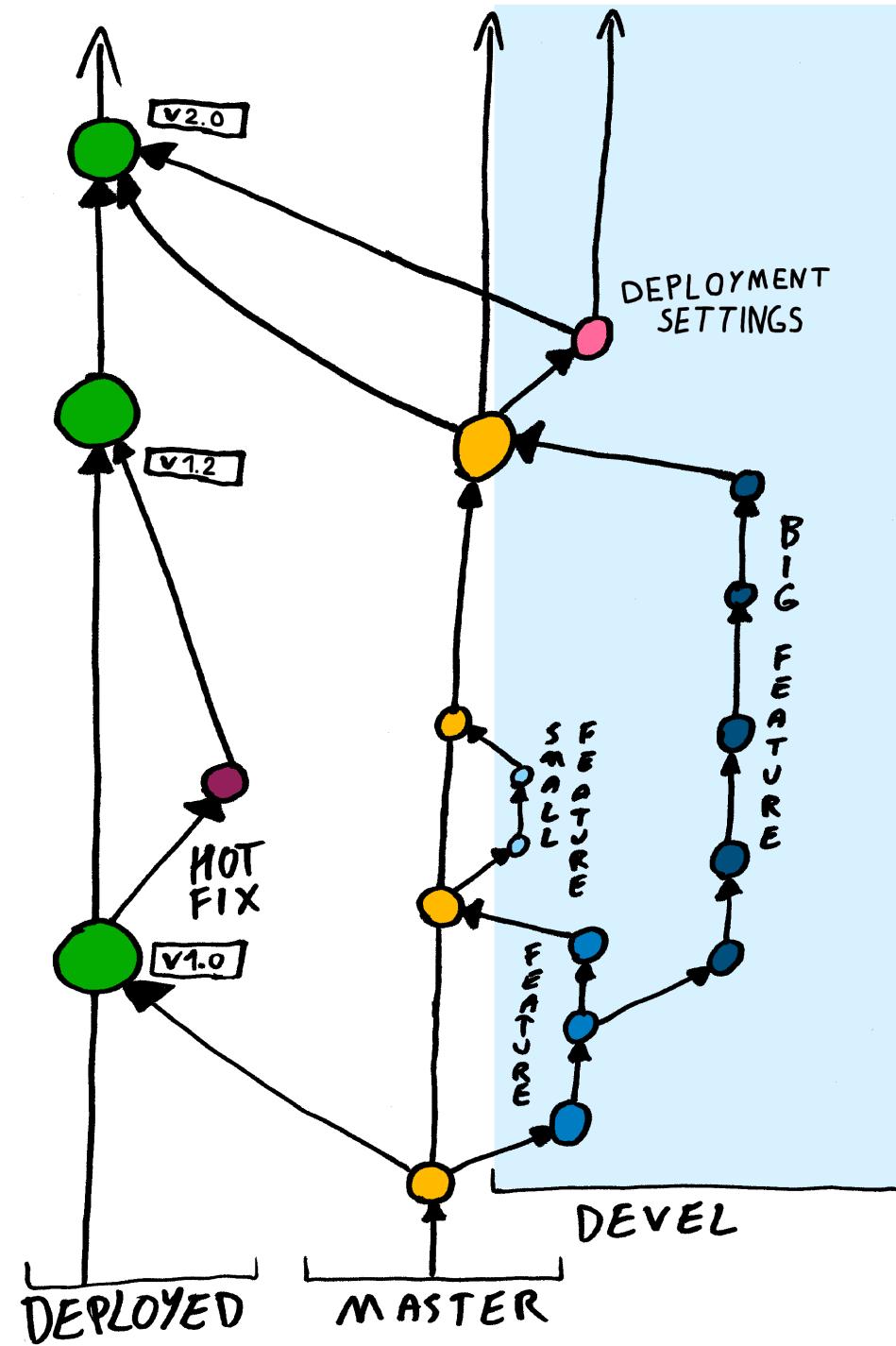
BRANCH + MERGE



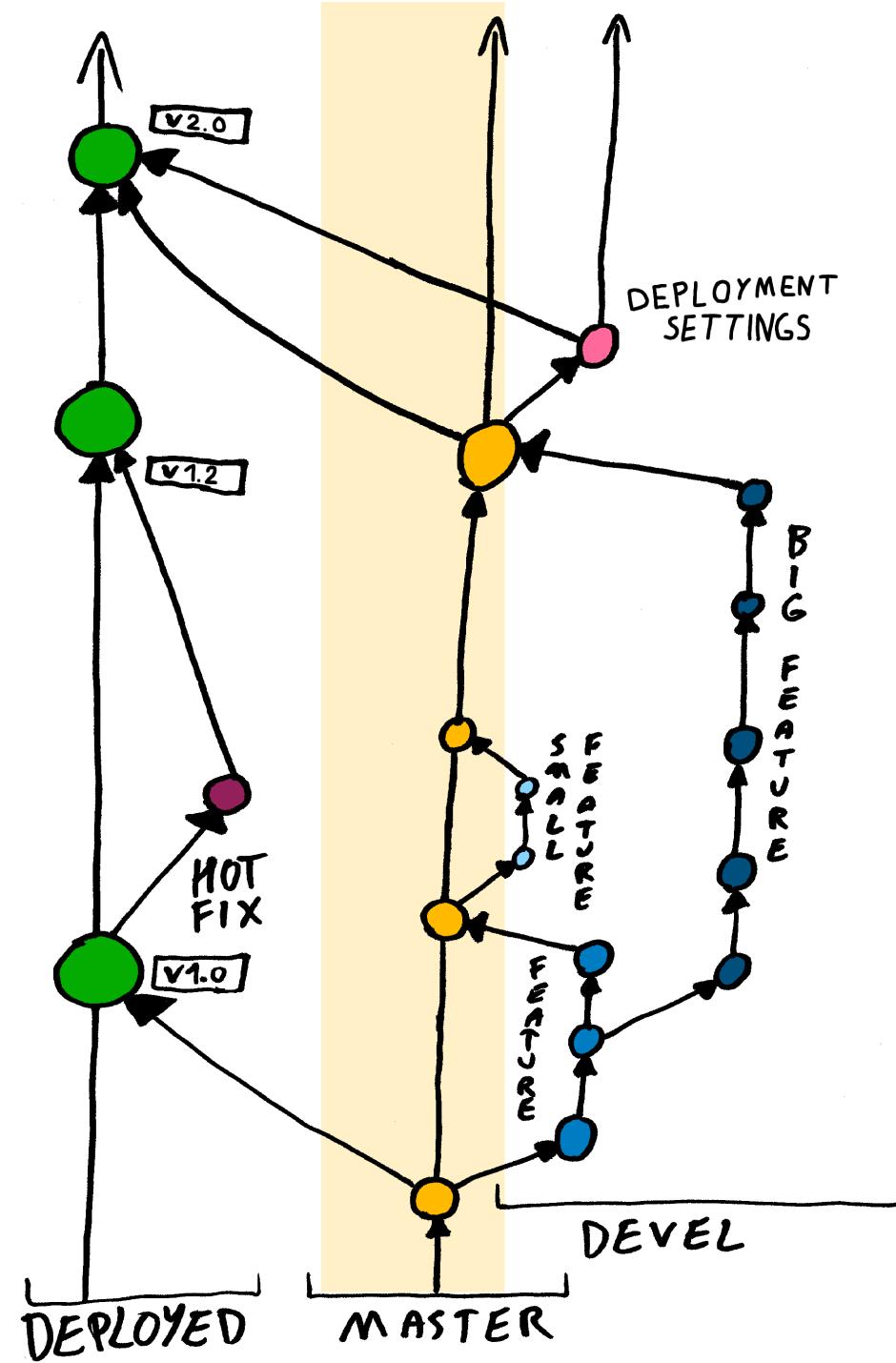
Przykład „Real Life“



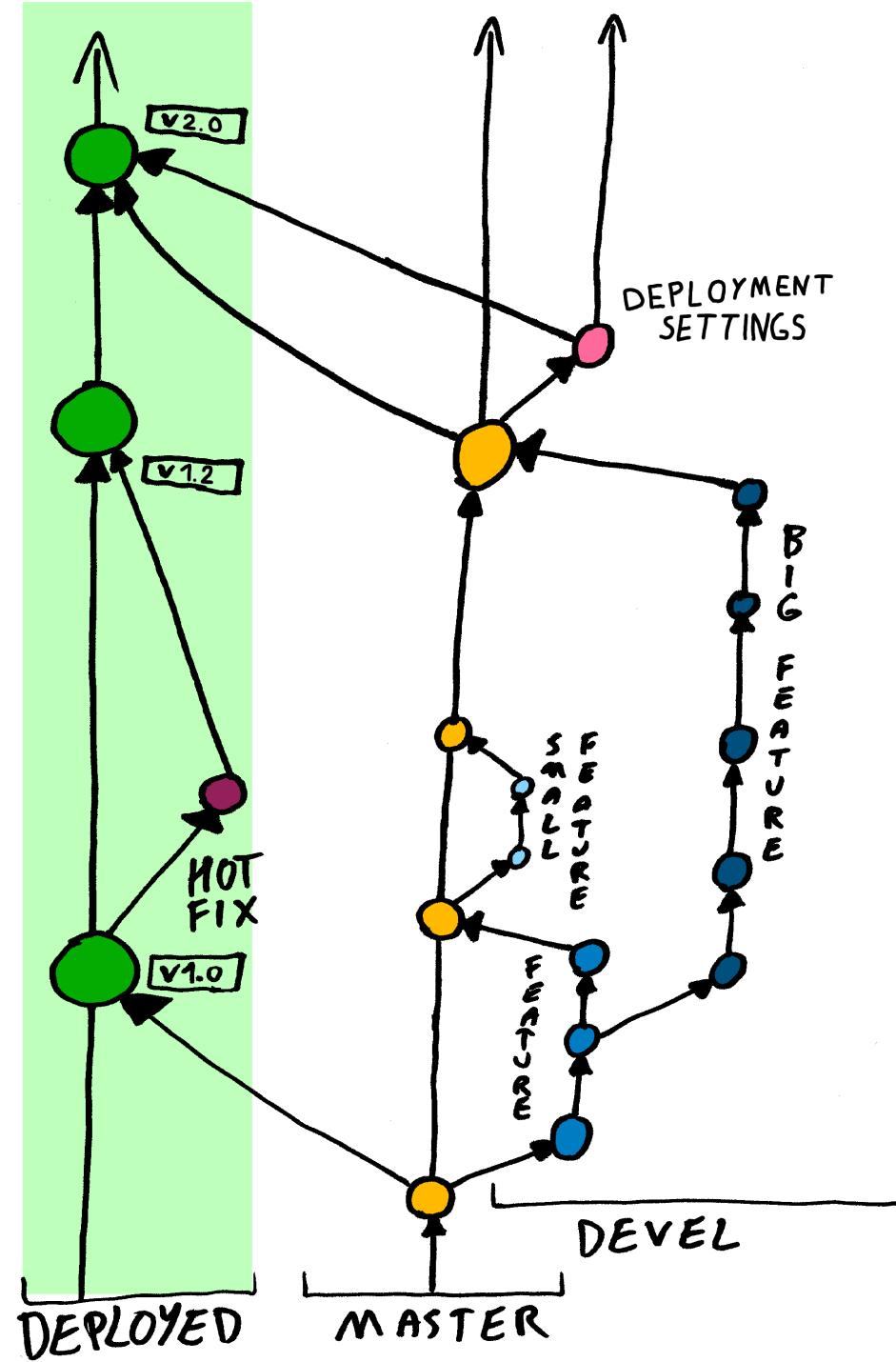
Przykład „Real Life“



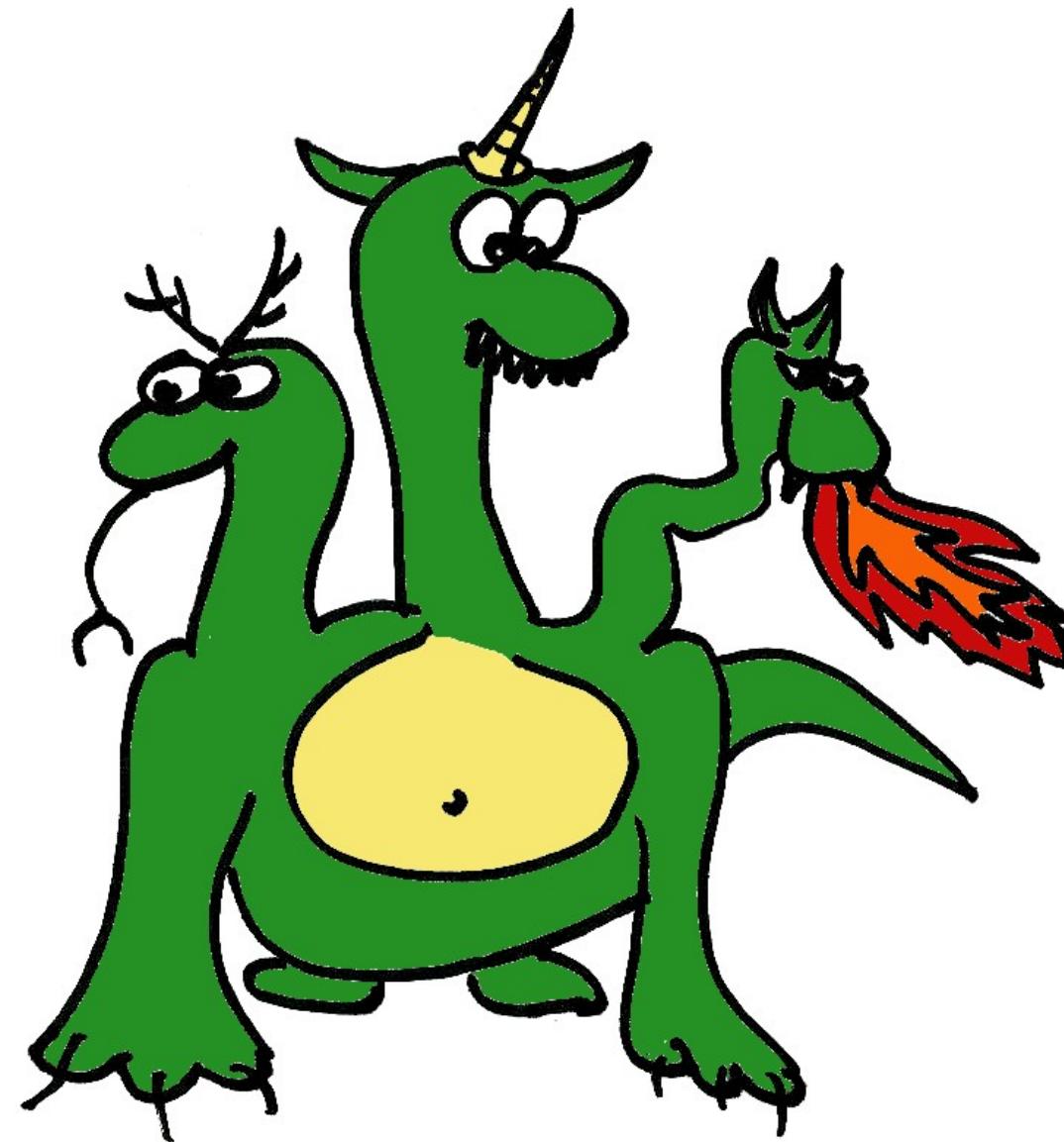
Przykład „Real Life“



Przykład „Real Life“

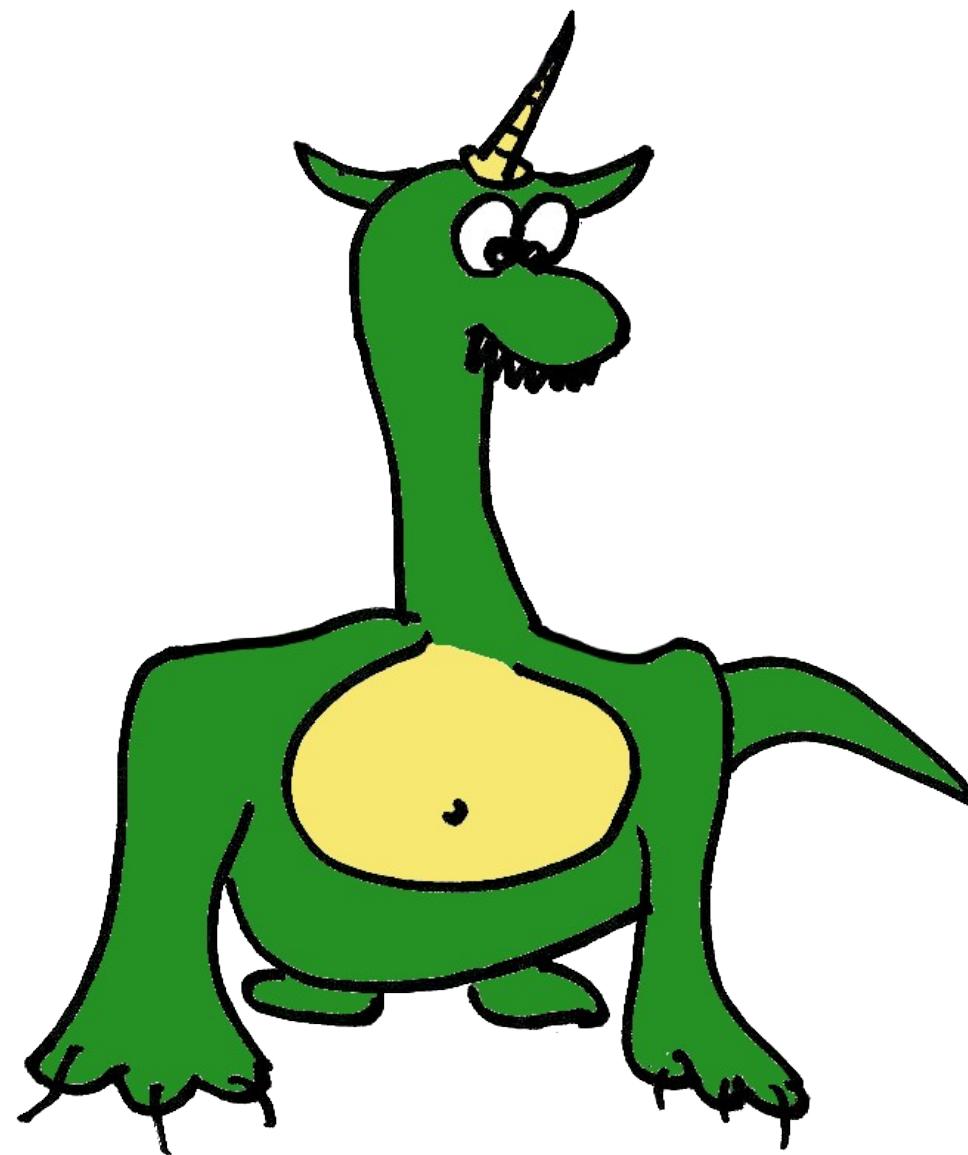


So many heads...?



There's just one .git/HEAD

```
$ Git branch  
* master  
  fire  
  snake
```

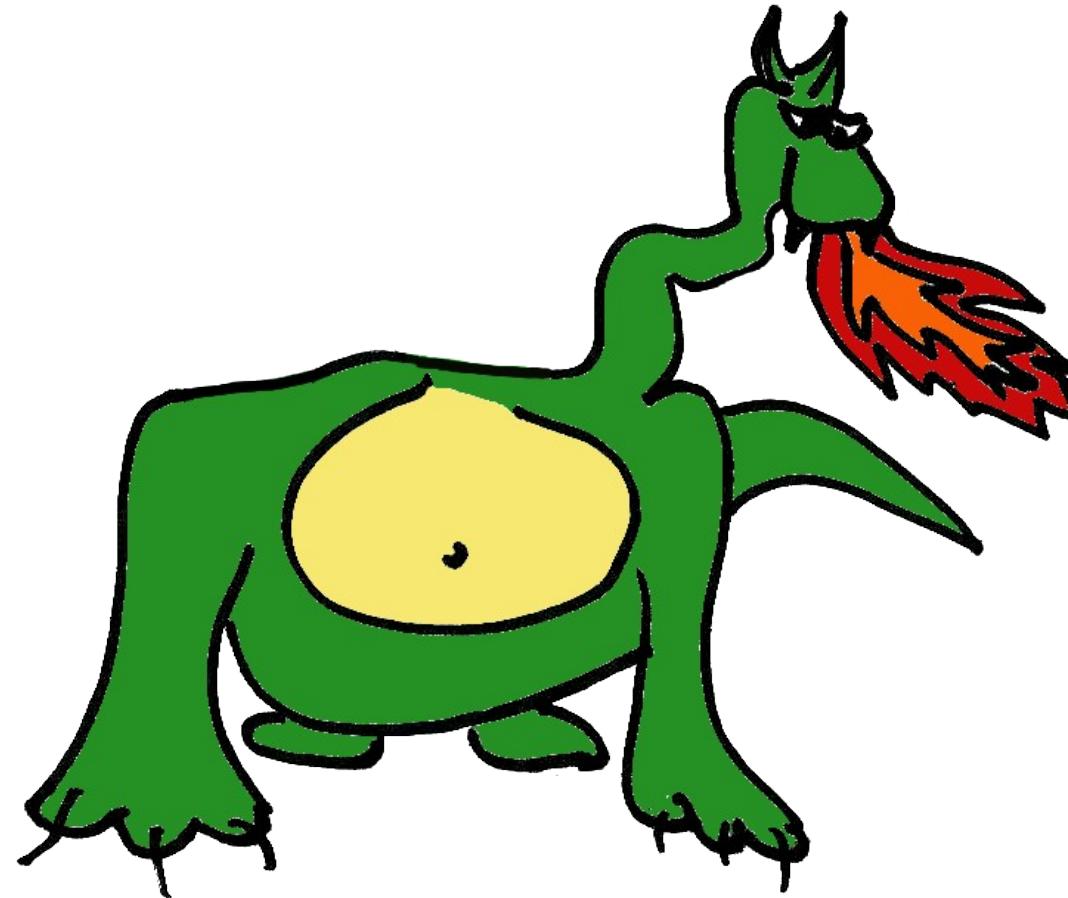


There's just one .git/HEAD

Git checkout fire

There's just one .git/HEAD

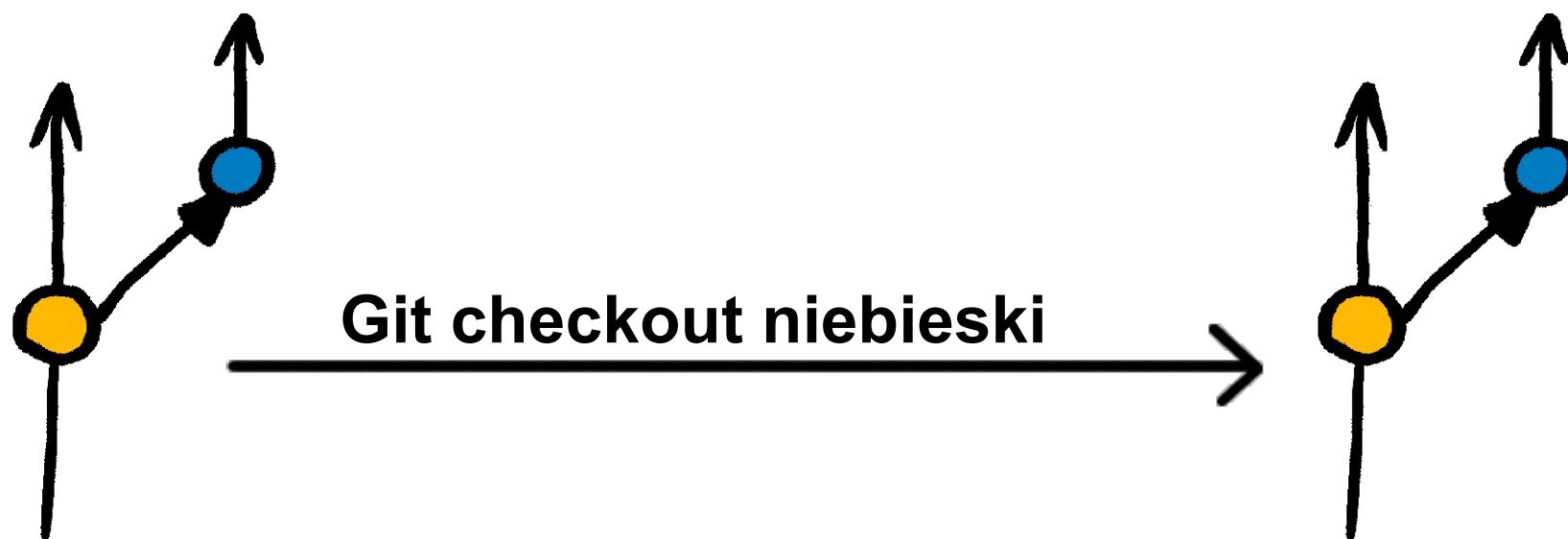
```
$ Git branch  
  master  
* fire  
  snake
```



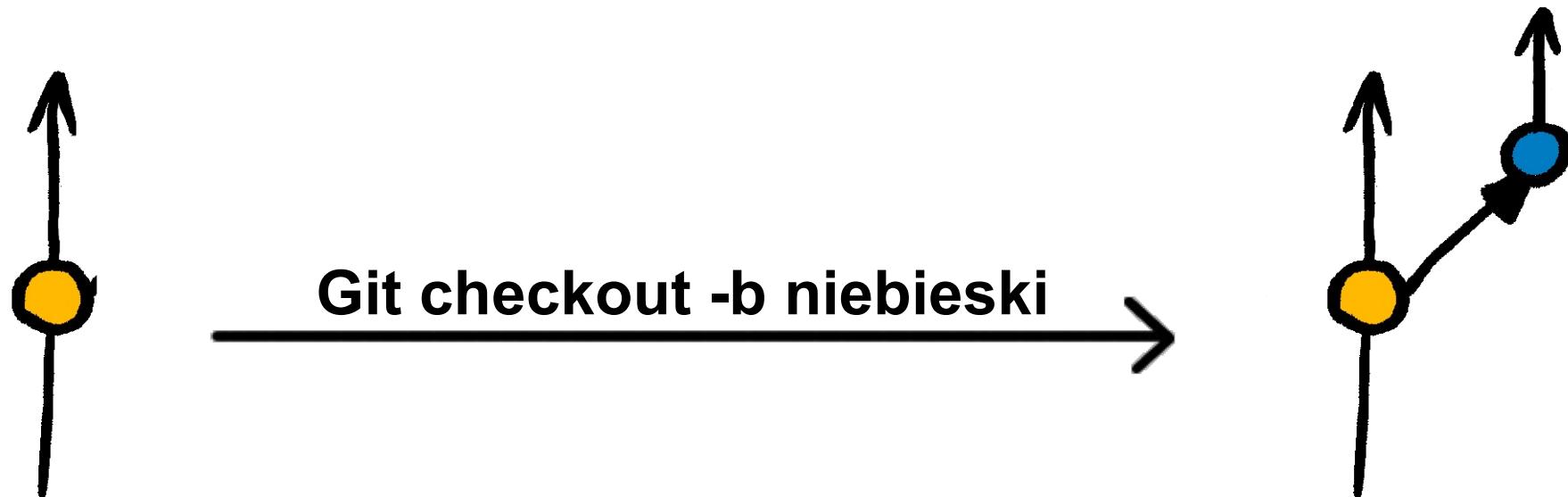
Nowy branch = trywialne



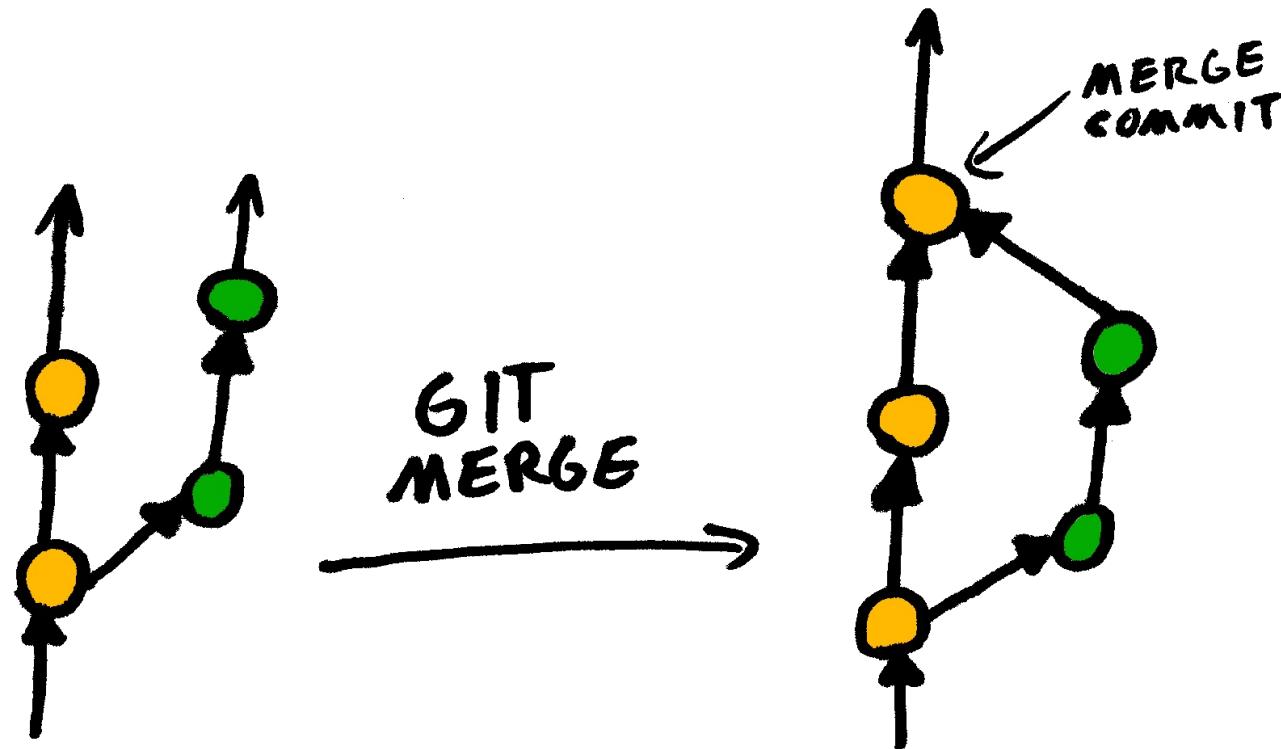
Nowy branch = trywialne



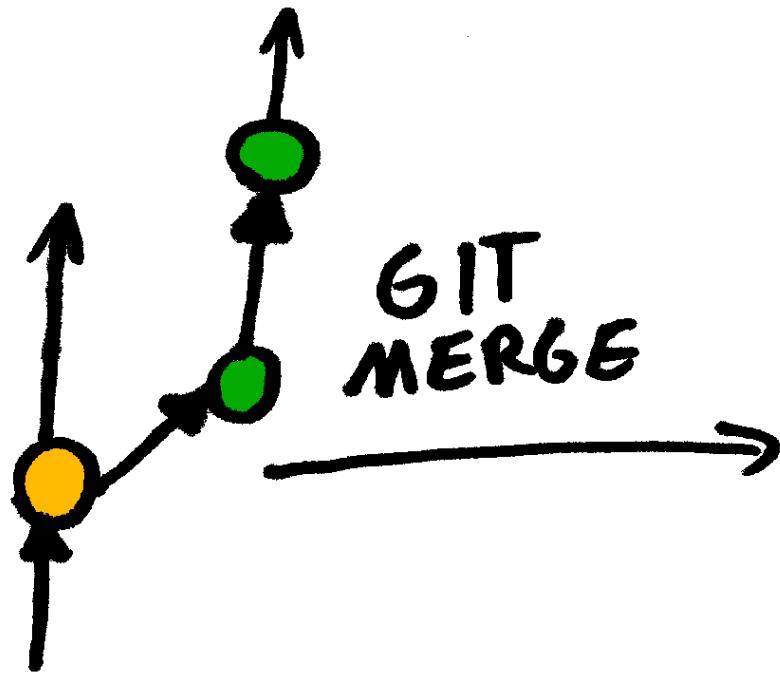
Nowy branch, trywialniej



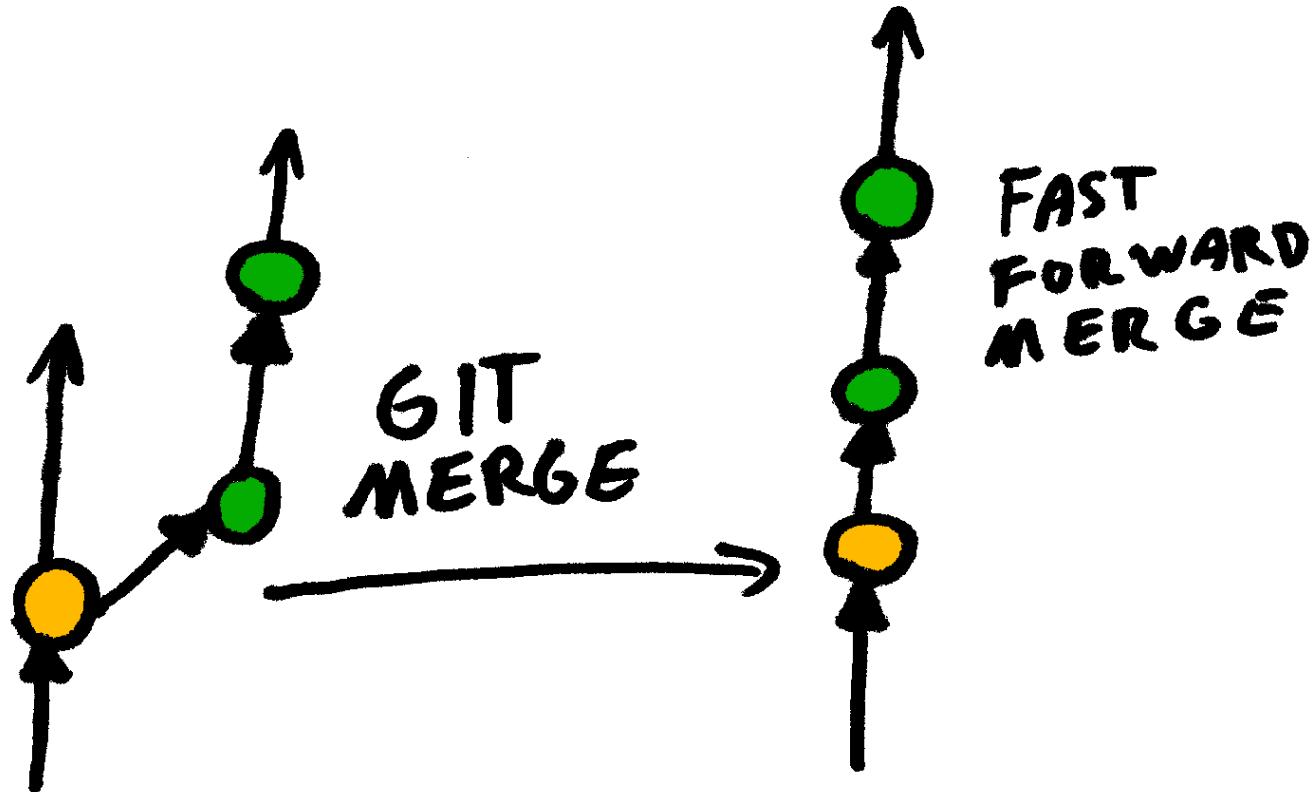
Git merge zielony



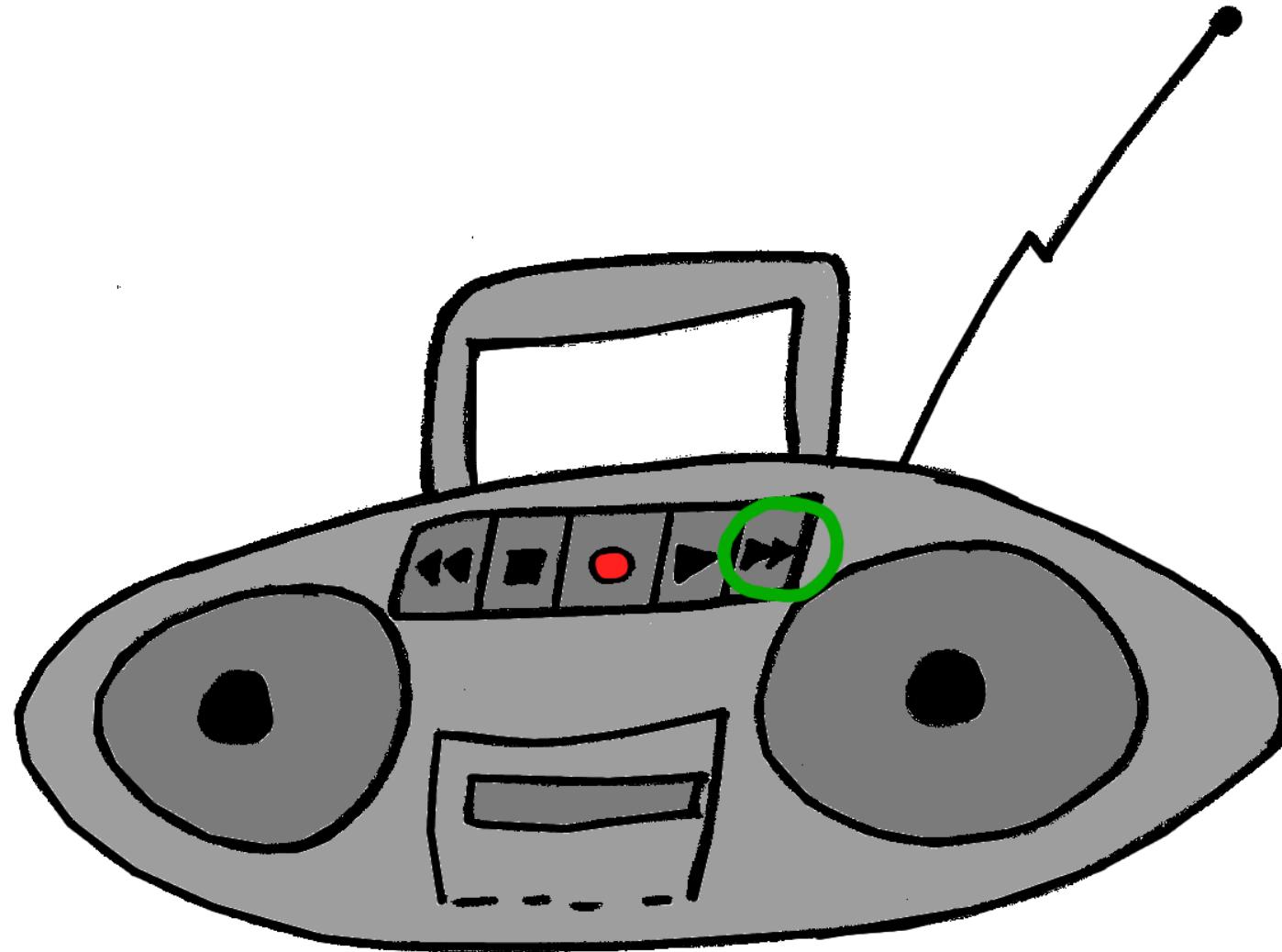
Git merge zielony



Git merge zielony



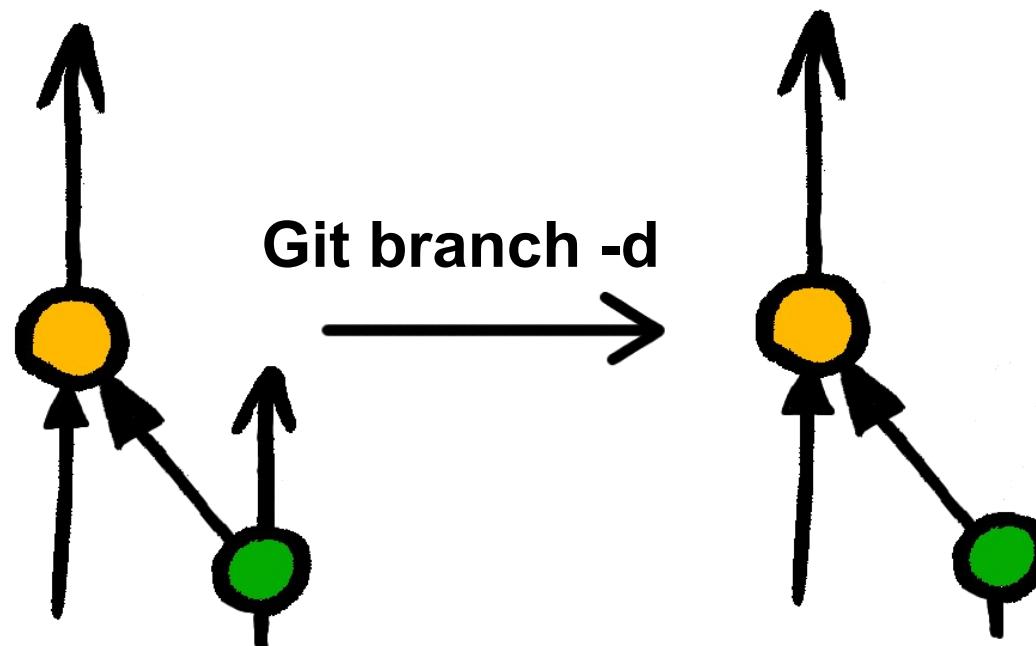
Fast Forward merge strategy



Git merge zielony --no-ff

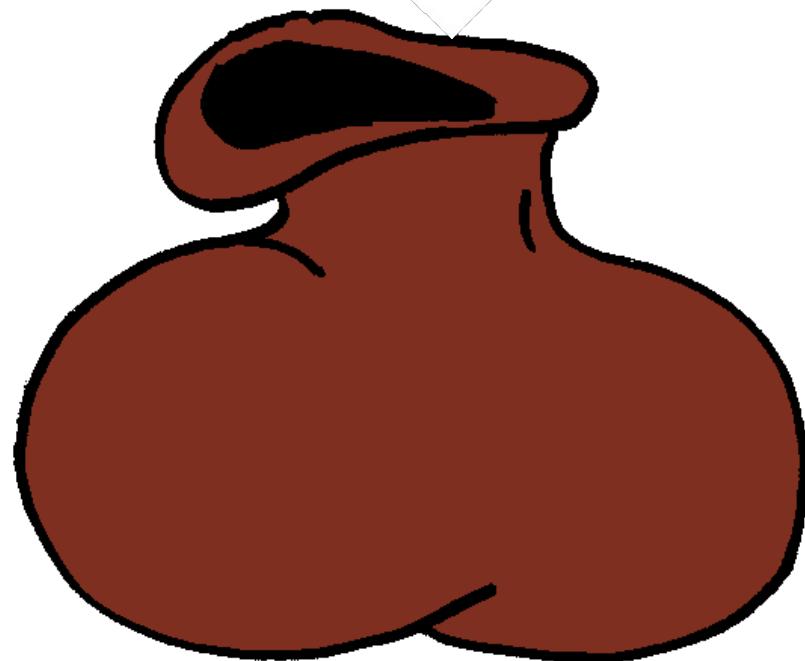
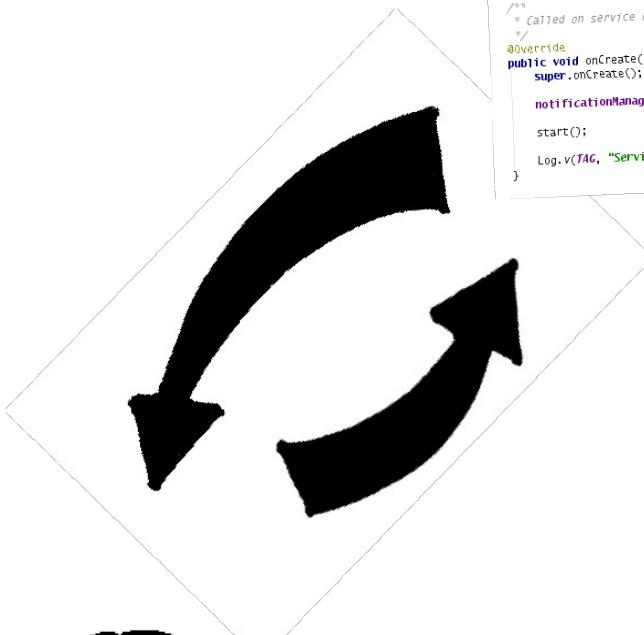


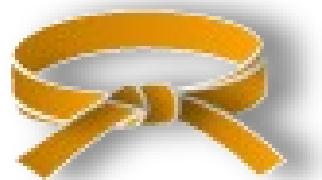
Git branch -d zielony



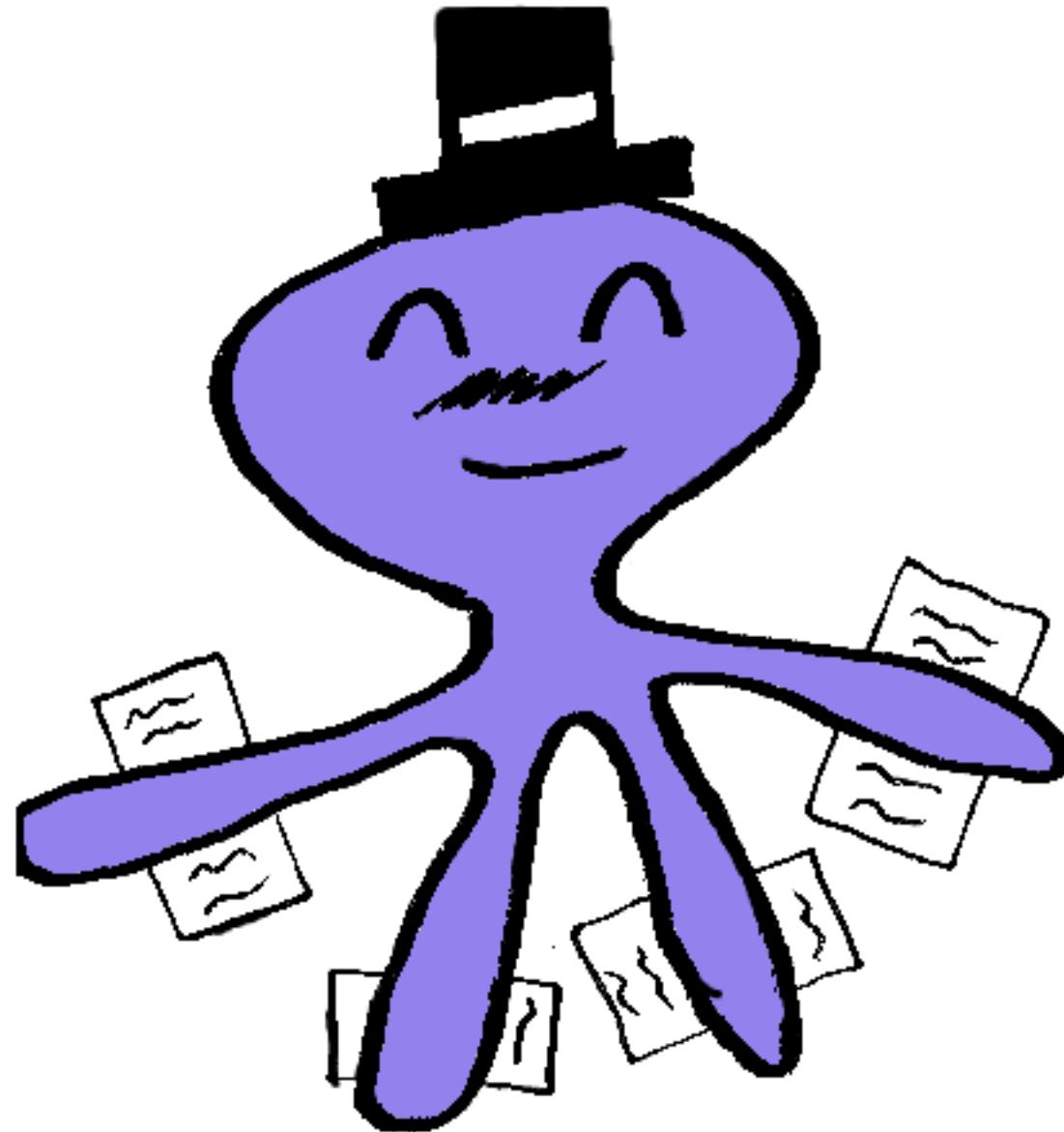
STASH

```
// logger tag  
private static final String TAG = WorkTimeNotifierService.class.getSimpleName();  
  
// time checking interval  
private static final long INTERVAL = 10 * Constants.MINUTE;  
  
private Timer timer = new Timer();  
public static final String INTENT_HEY_STOP_WORKING = "HEY_STOP_WORKING";  
  
NotificationManager notificationManager;  
  
/**  
 * Called on service creation, will start the timer  
 */  
@Override  
public void onCreate() {  
    super.onCreate();  
  
    notificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);  
  
    start();  
  
    Log.v(TAG, "Service started");  
}
```



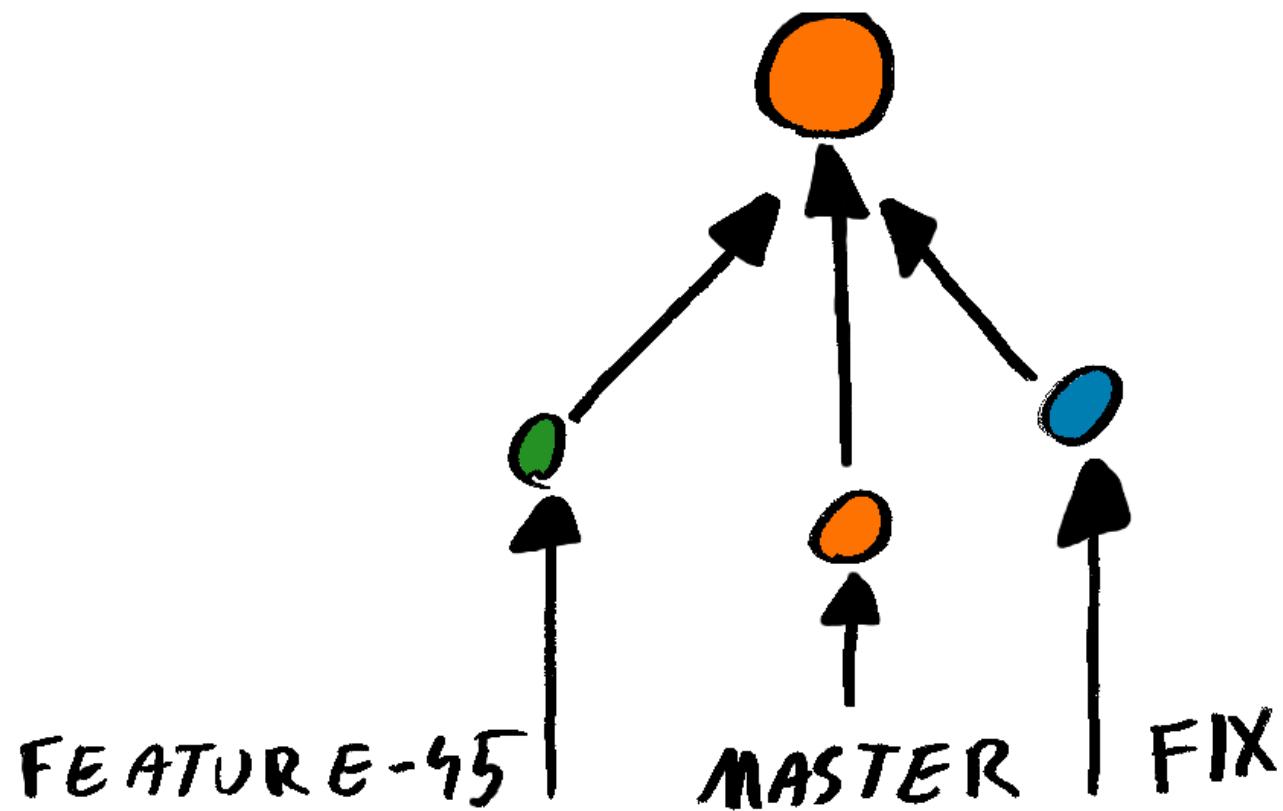


OCTOPUS

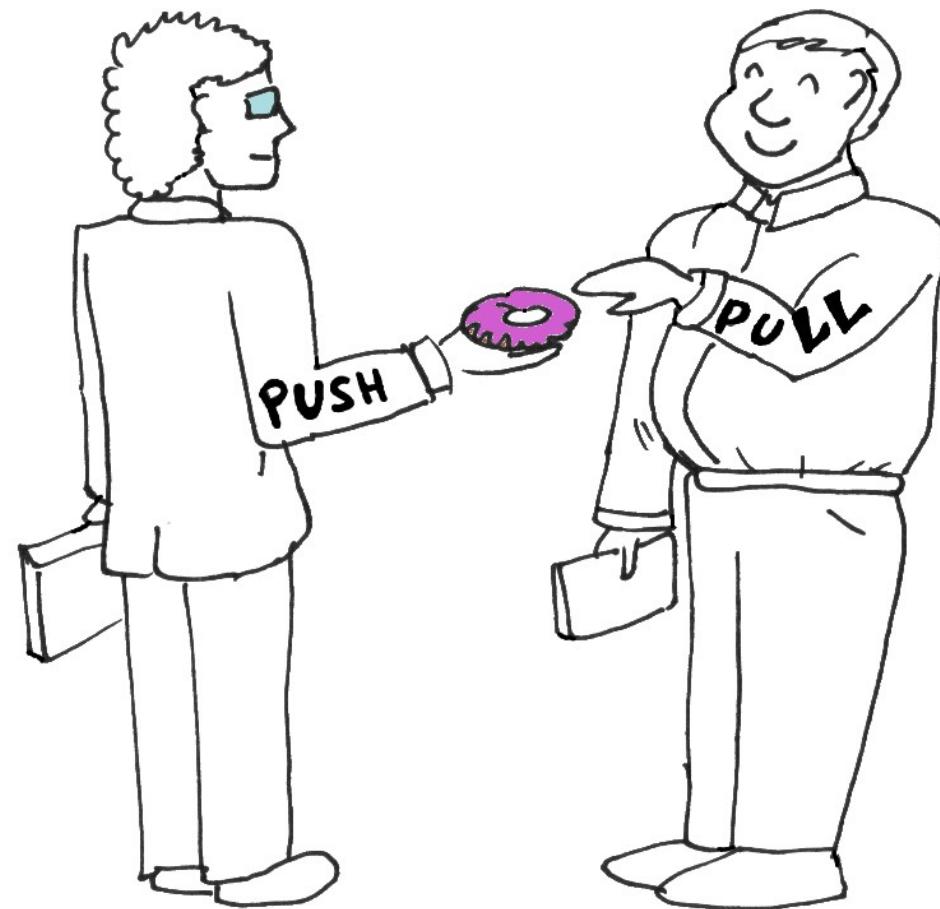


OCTOPUS (Merge Engine)

**Git checkout master
Git merge feature-45 fix**

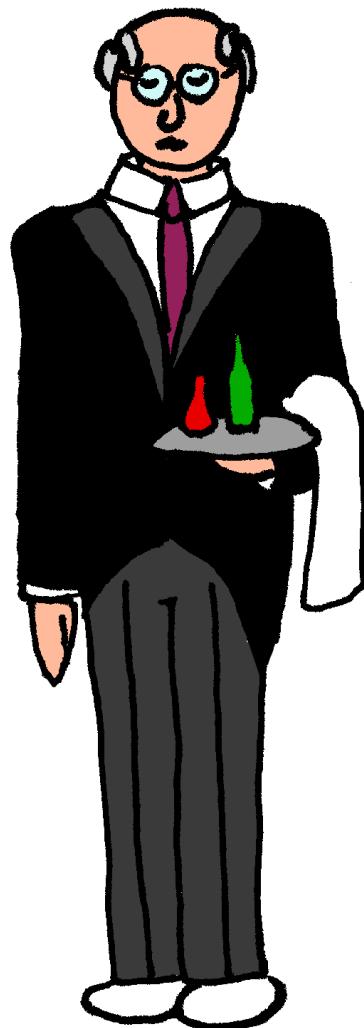


Shall I push or shall I pull..?



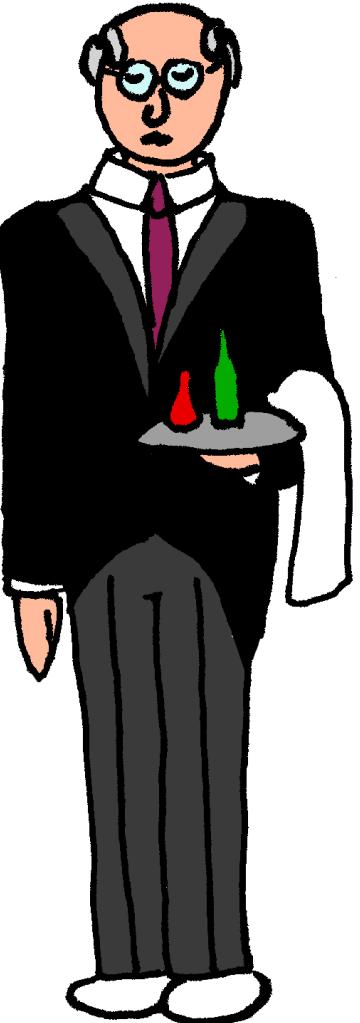
Introducing: the butler (server)

Git serve



Introducing: the butler (server)

Git serve

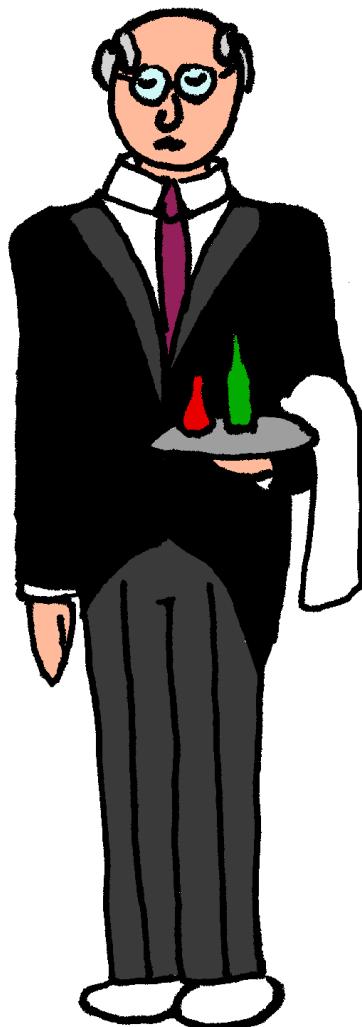


```
git daemon \  
  --reuseaddr \  
  --verbose \  
  --base-path=. \  
  --export-all \  
  ./.git
```

```
Git config alias.serve "...."
```

Introducing: the butler (server)

Git serve

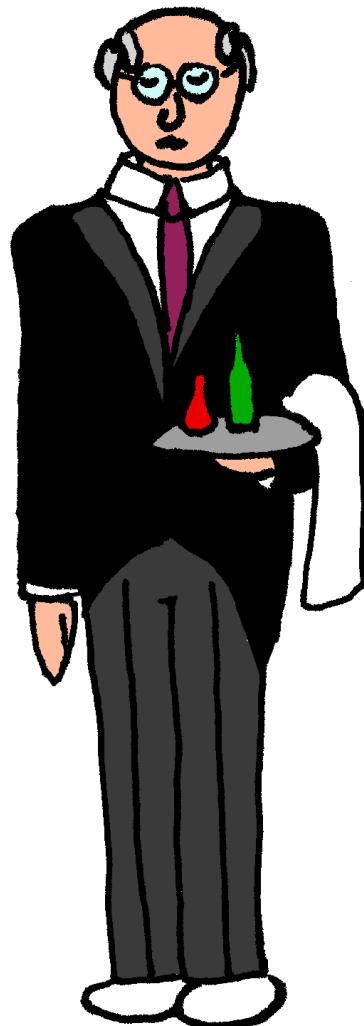


Git clone git://butler/repo.git

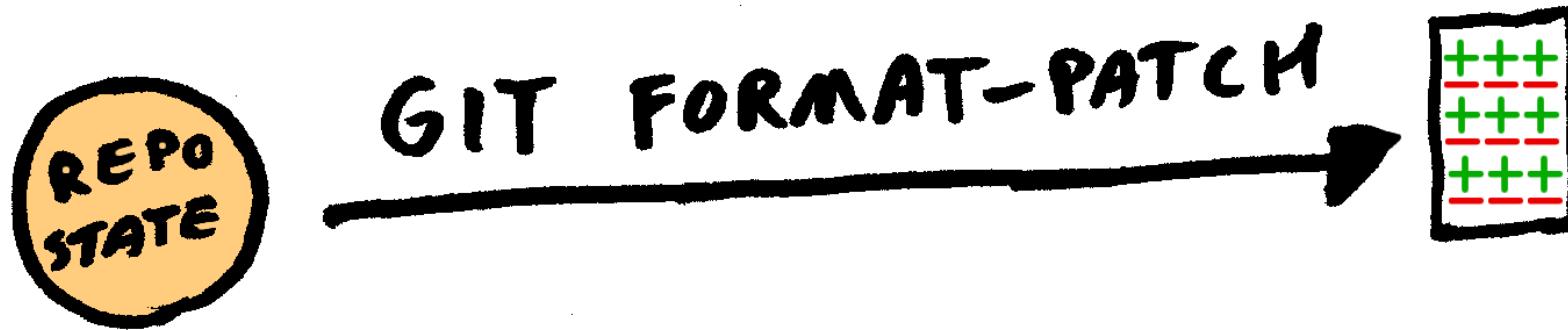


Introducing: the butler (server)

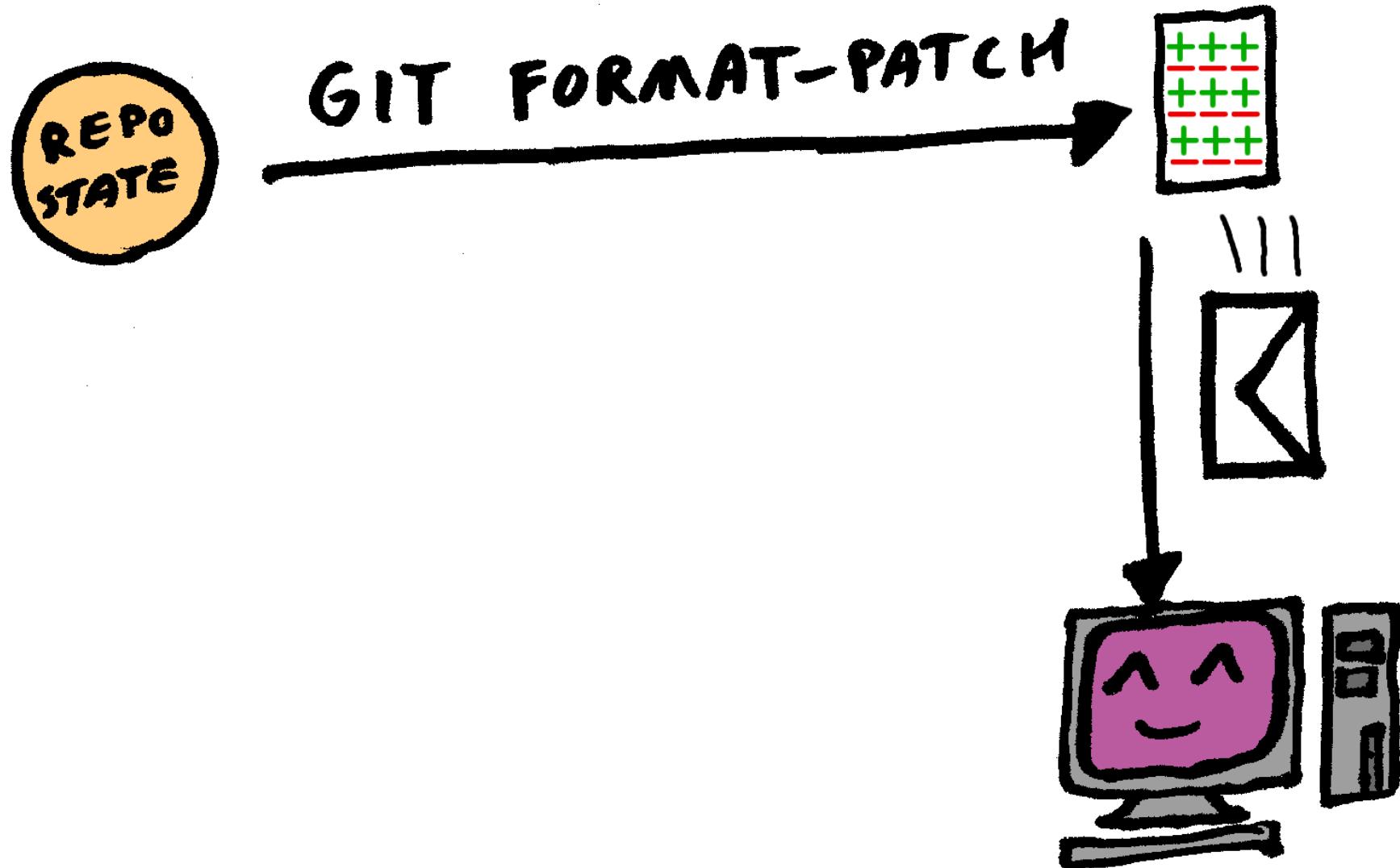
Git serve



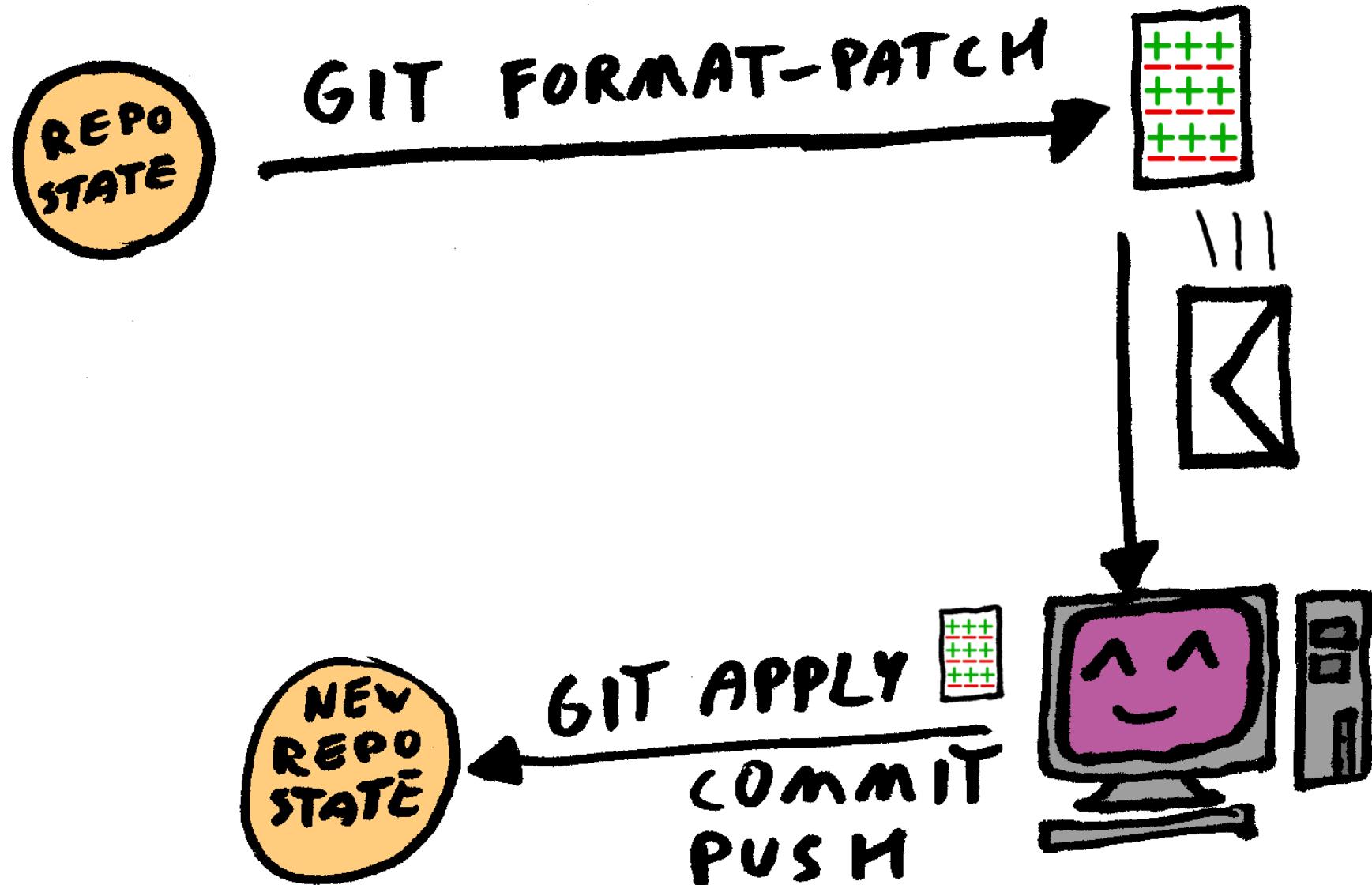
The good old way - Patches



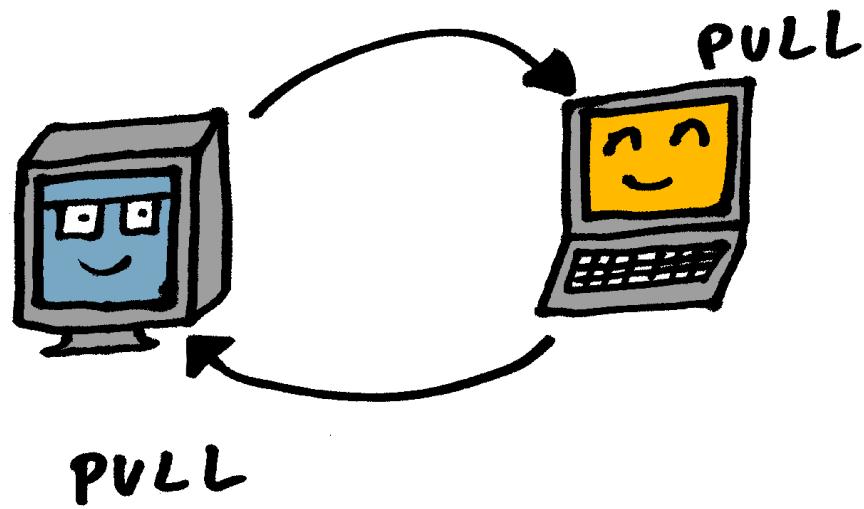
The good old way - Patches



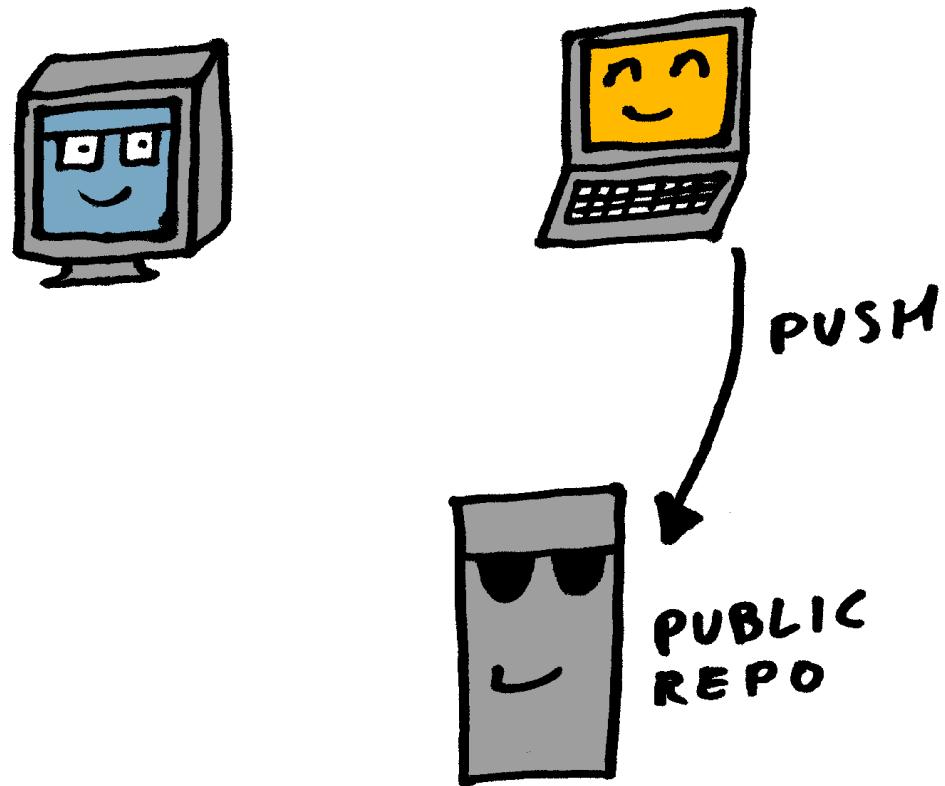
The good old way - Patches



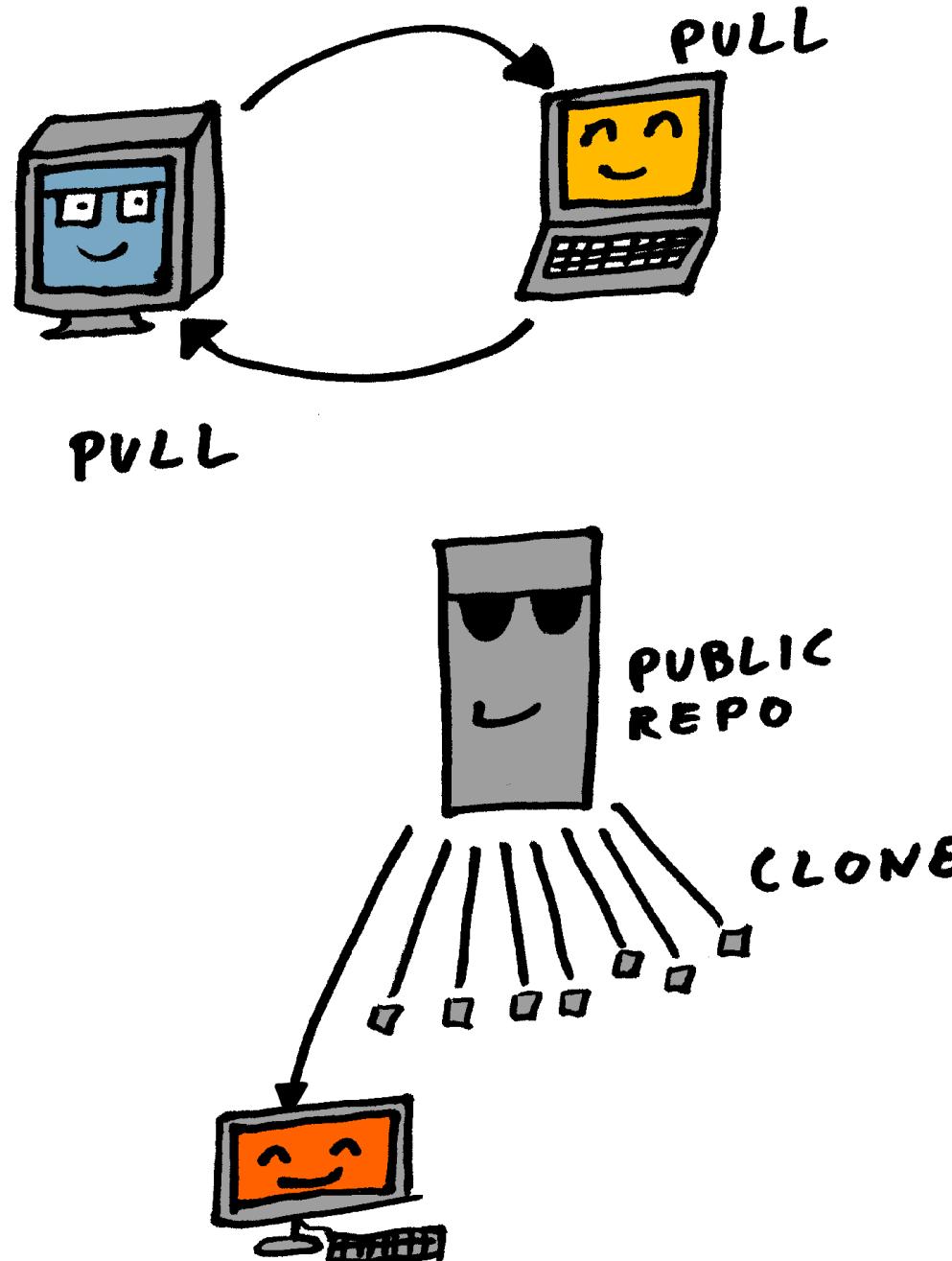
Typical Open Sou| Flow



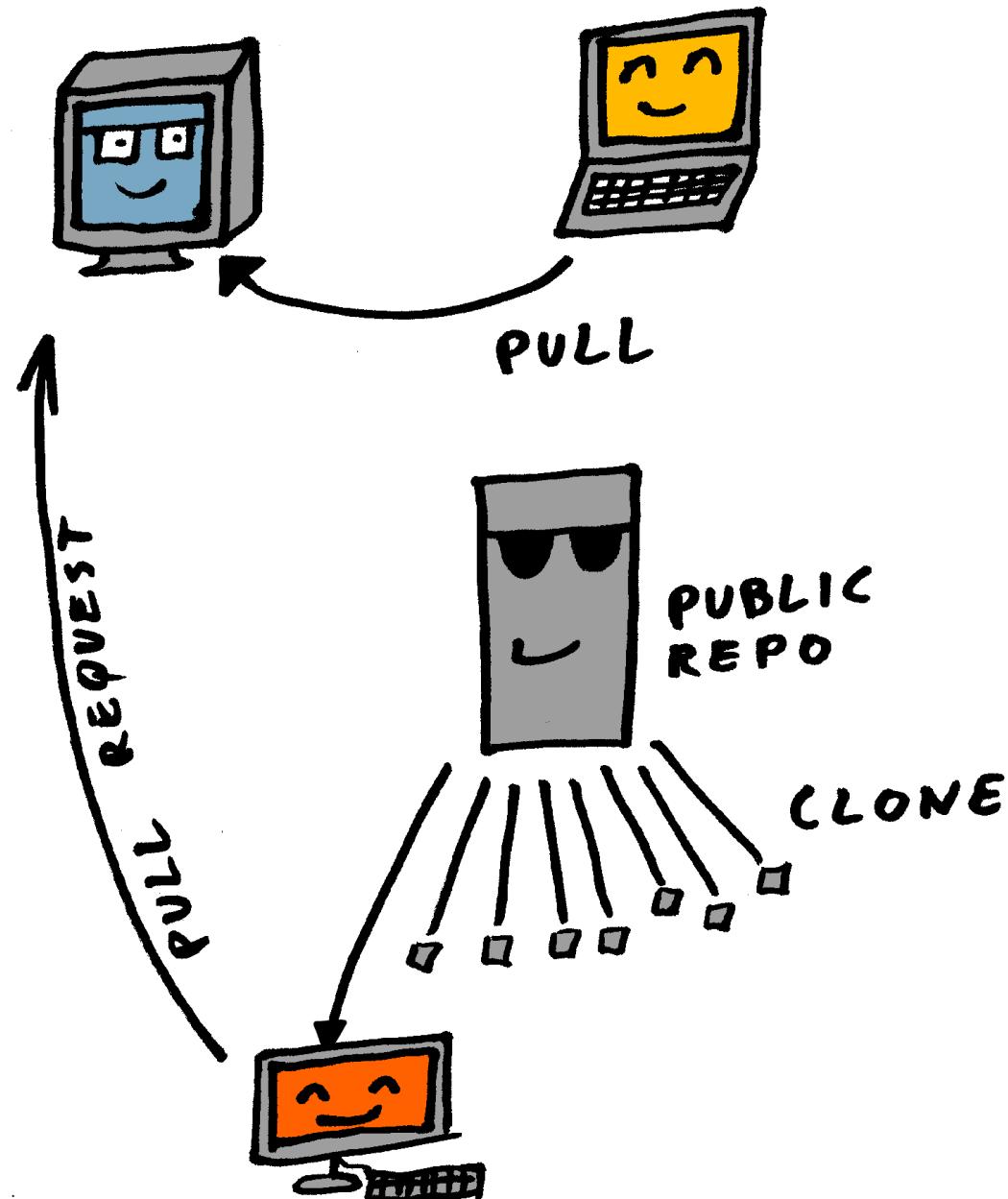
Typical Open Sou| Flow



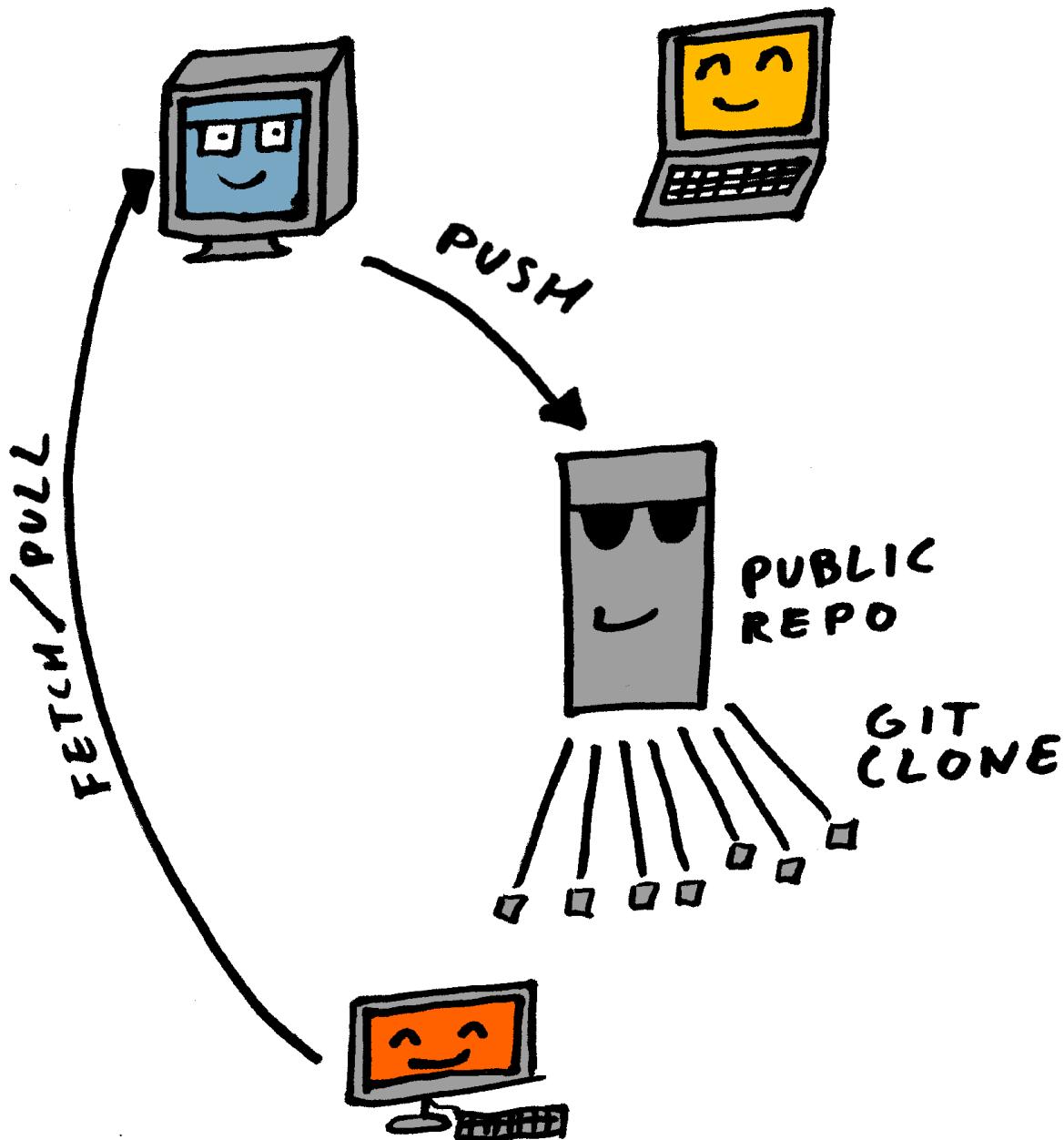
Typical Open Sou| Flow



Typical Open Sou| Flow



Typical Open Sou| Flow



THE MANLY MAN COMMAND

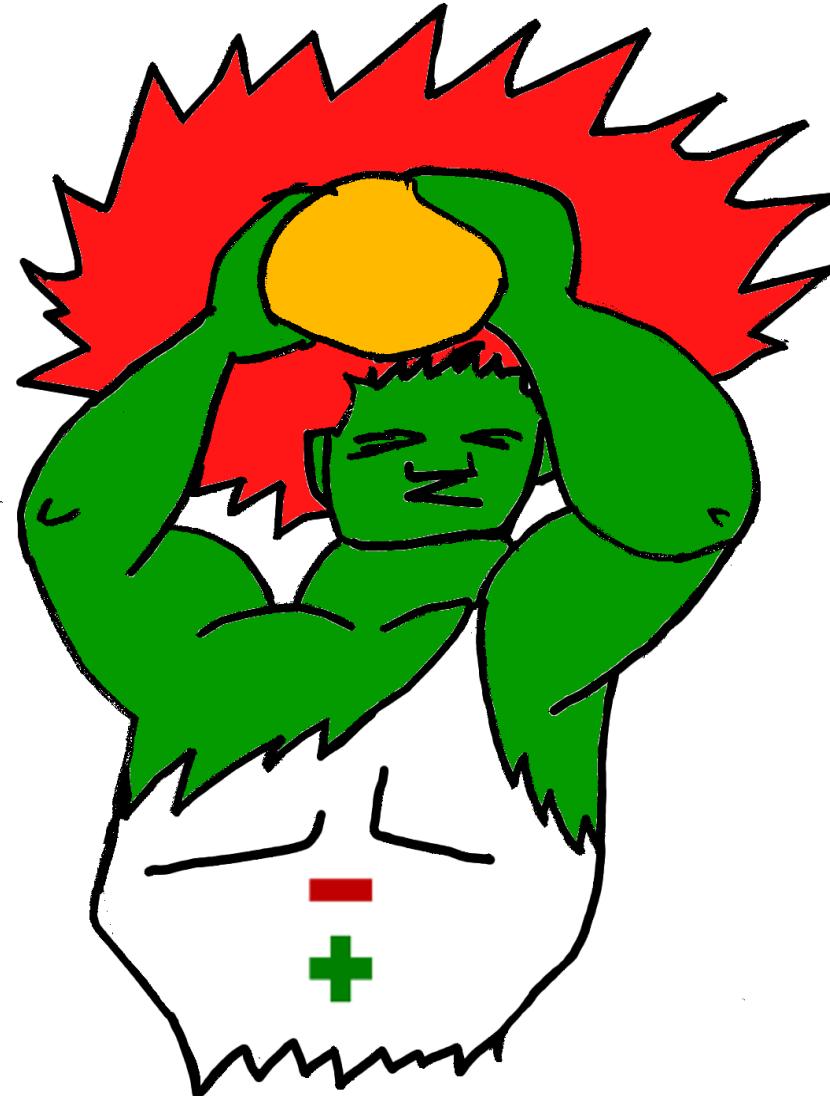
git rebase

Git rebase --interactive HEAD^²

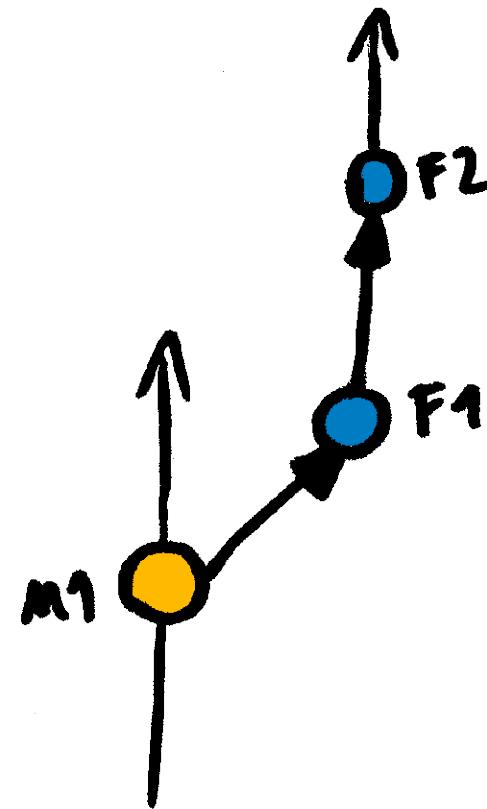


Git rebase --interactive HEAD^²

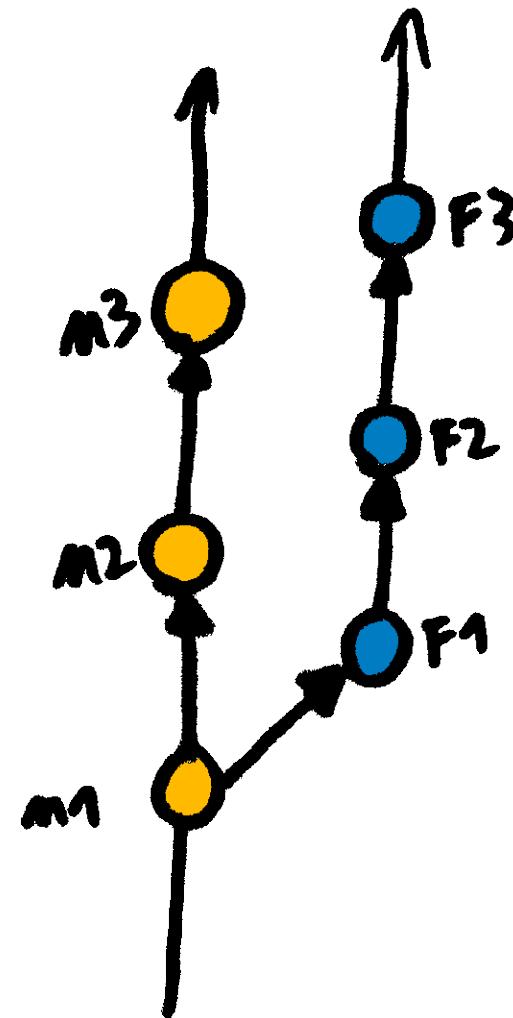
HULK SQUASH !!



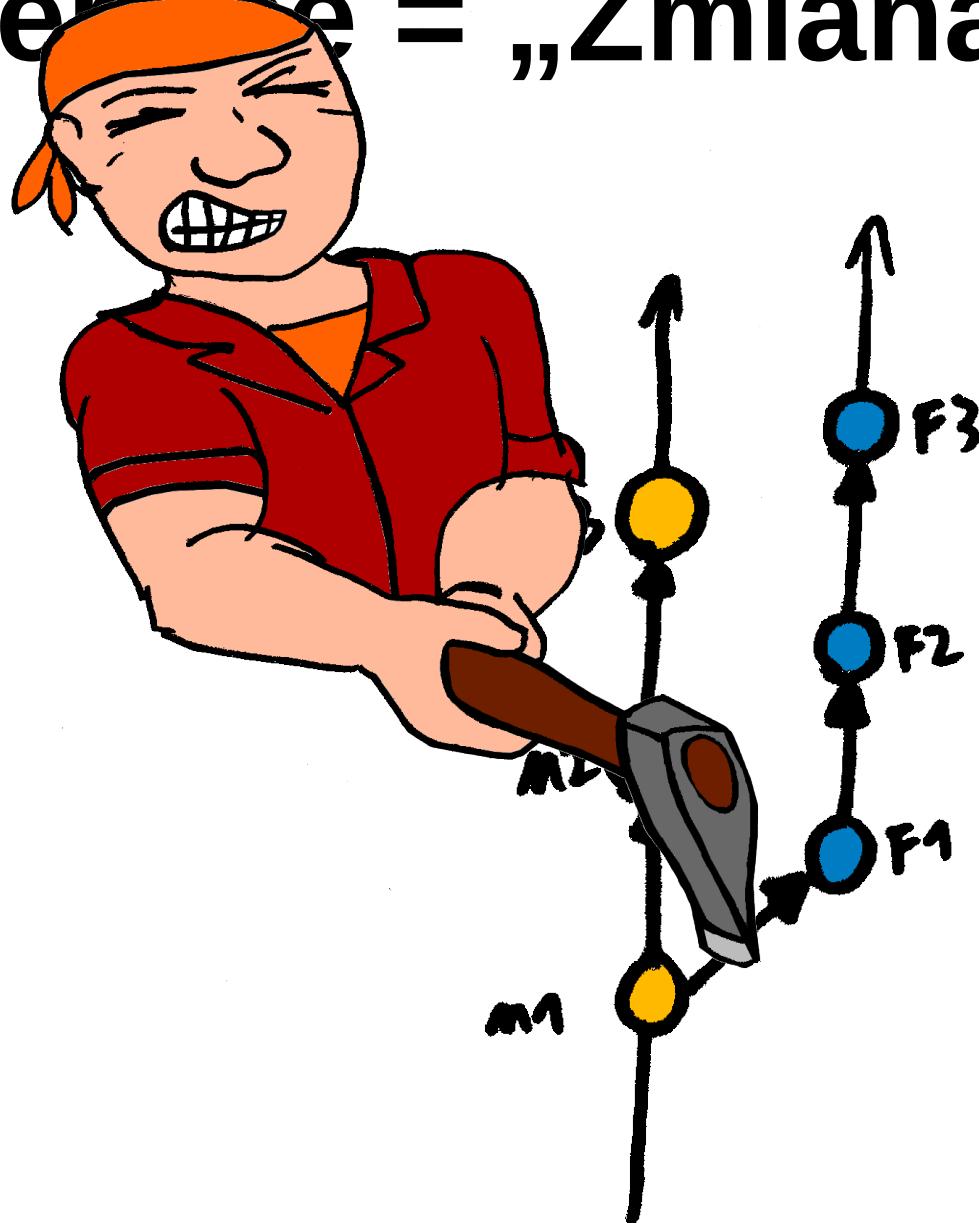
Rebase = „Zmiana bazy“



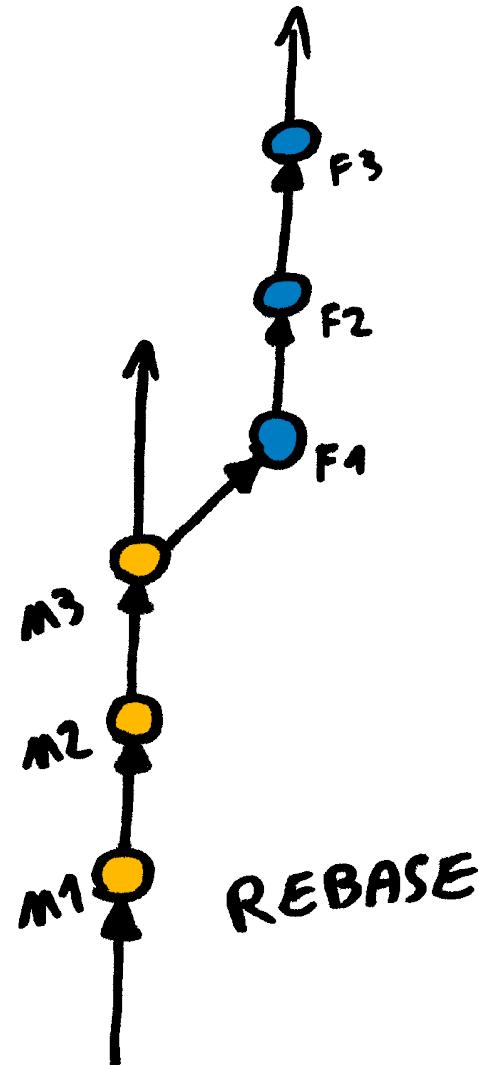
Rebase = „Zmiana bazy“



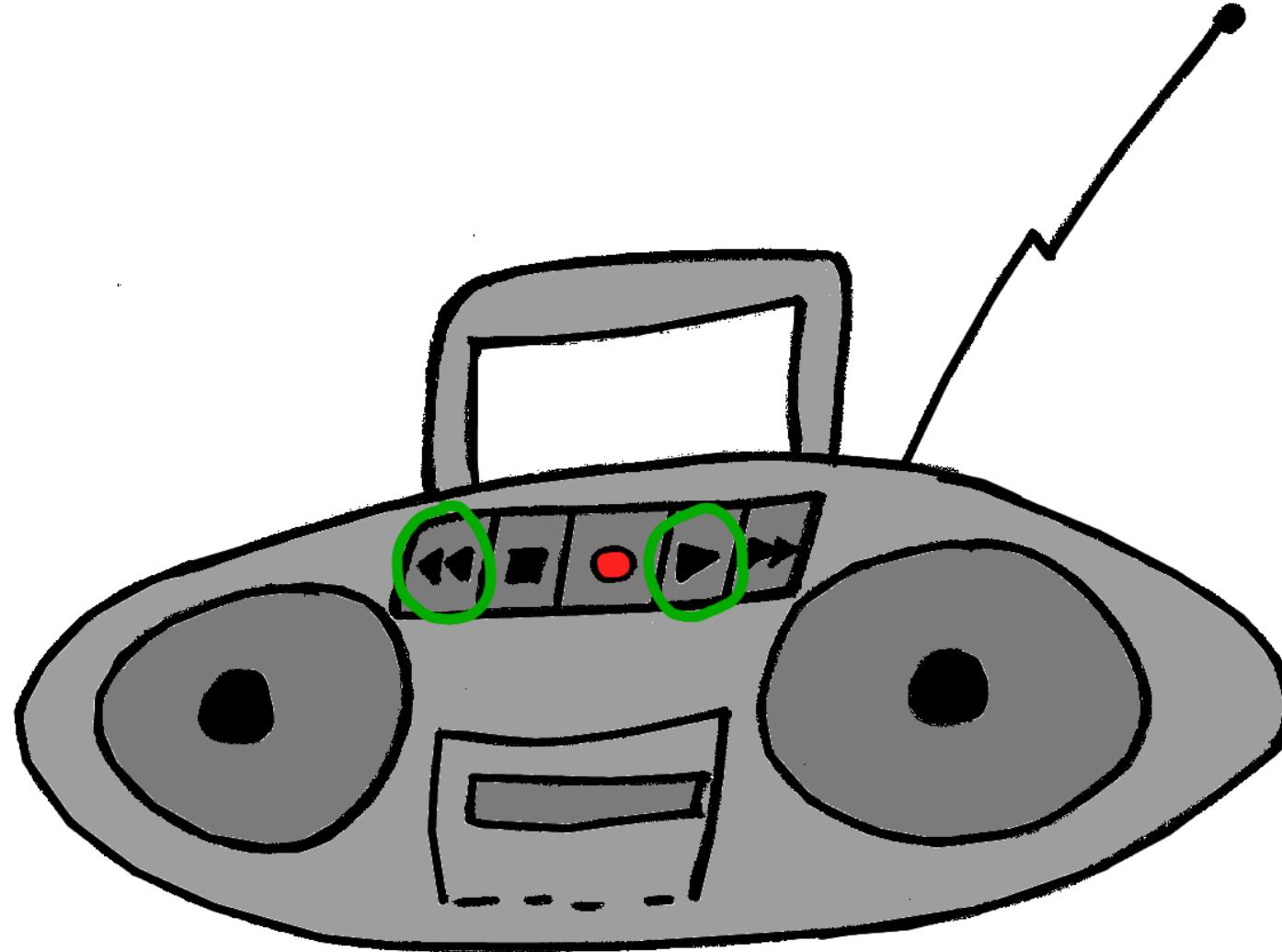
Rekurencje = „Zmiana bazy“



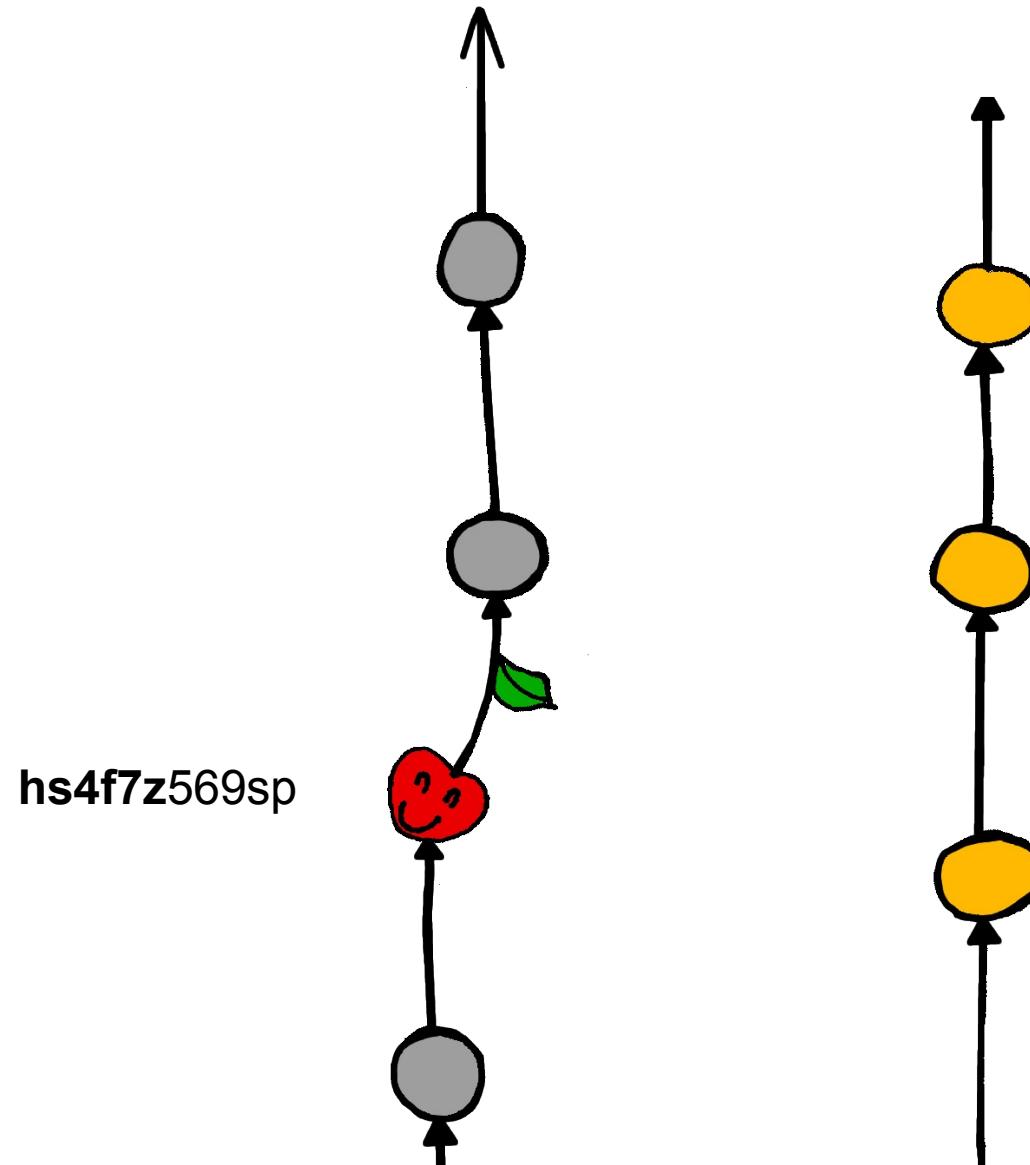
Rebase = „Zmiana bazy“



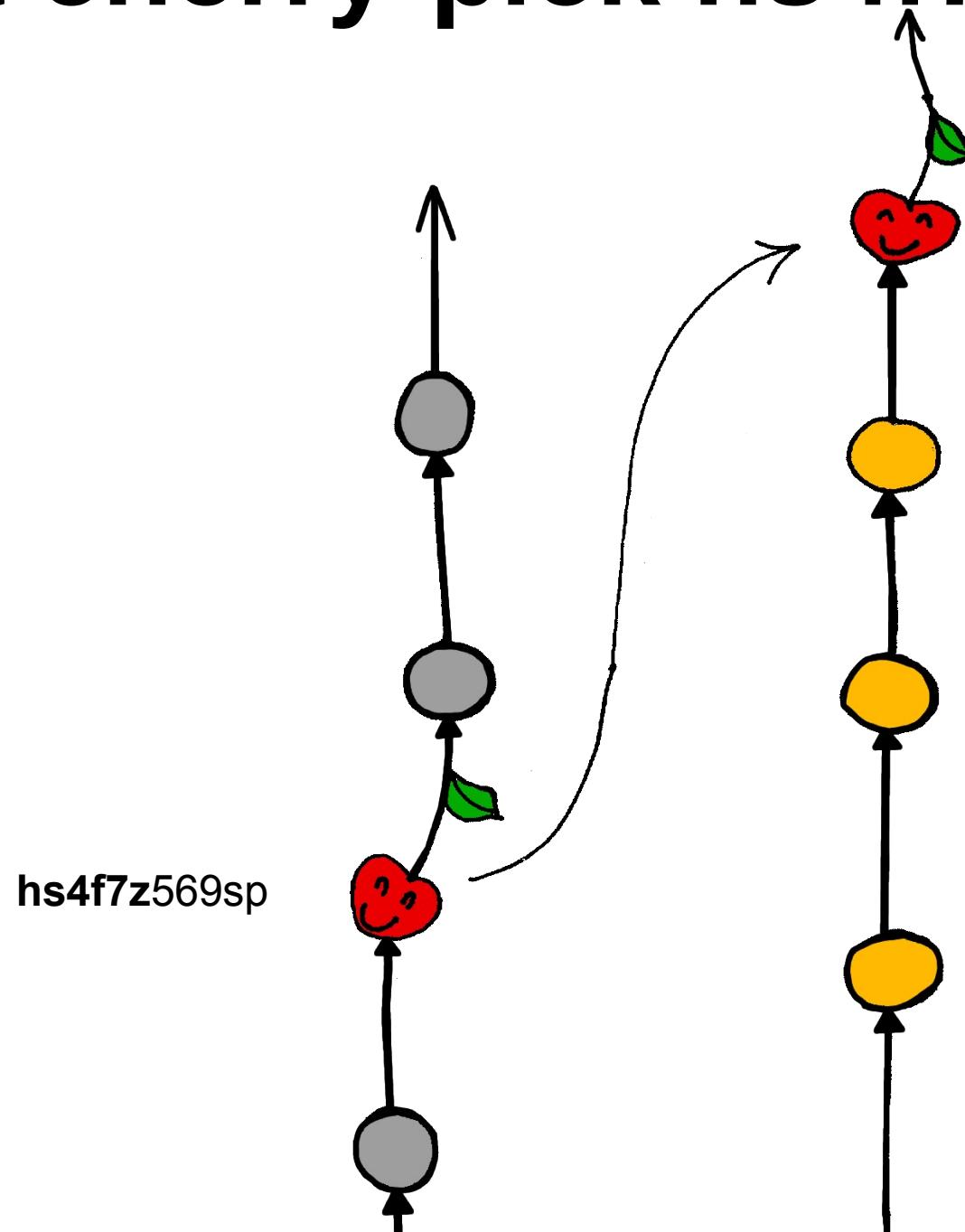
Rebase = Rewind + Replay



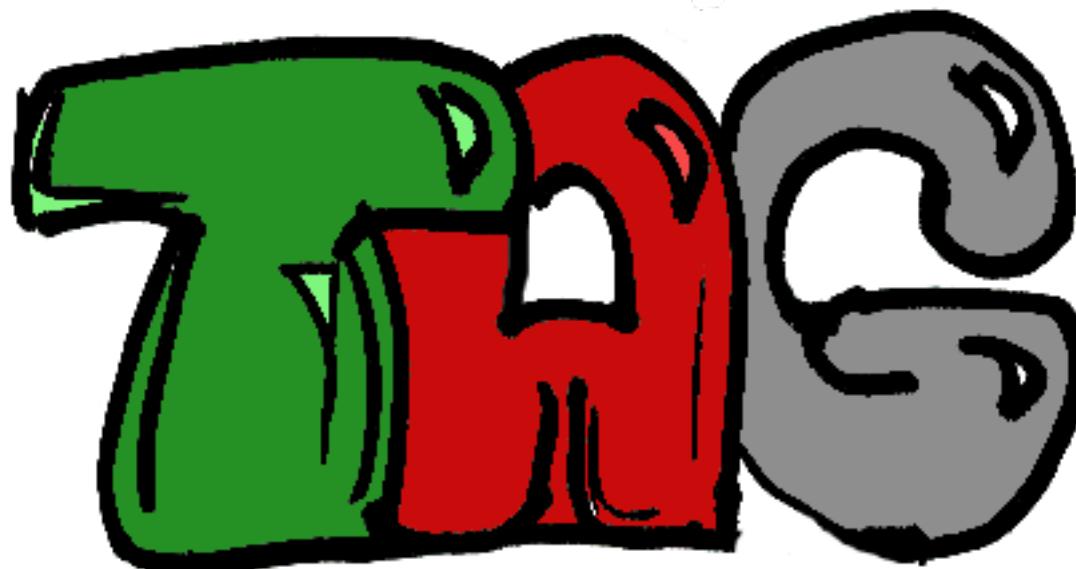
Git cherry-pick hs4f7z



Git cherry-pick hs4f7z

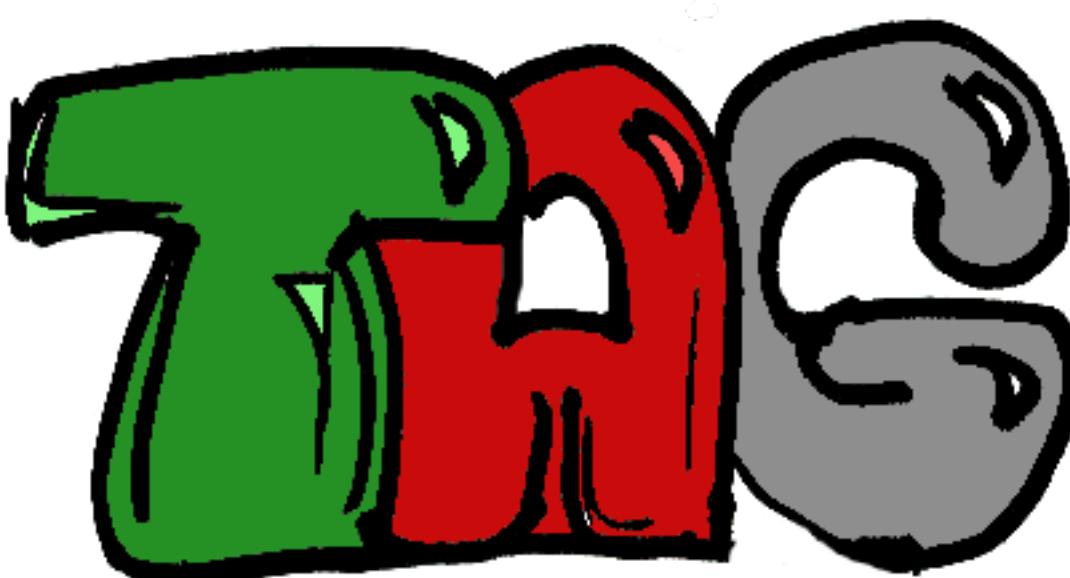


TAG



Git tag release-1.0

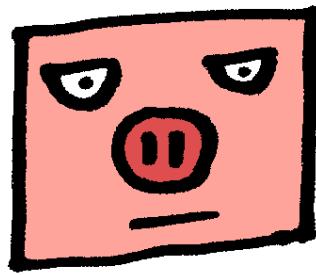
TAG

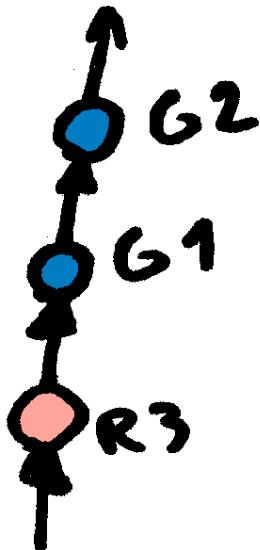


Git tag release-1.0
=

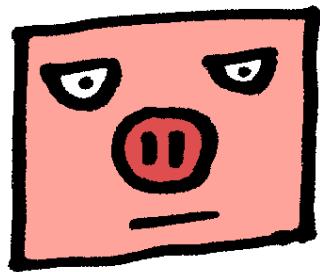
```
svn copy http://svn.example.com/repos/calc/trunk \
          http://svn.example.com/repos/calc/tags/release-1.0 \
          -m "Tagging the 1.0 release of the 'calc' project."
```

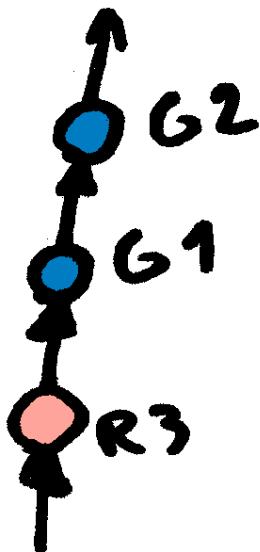
Git svn ...



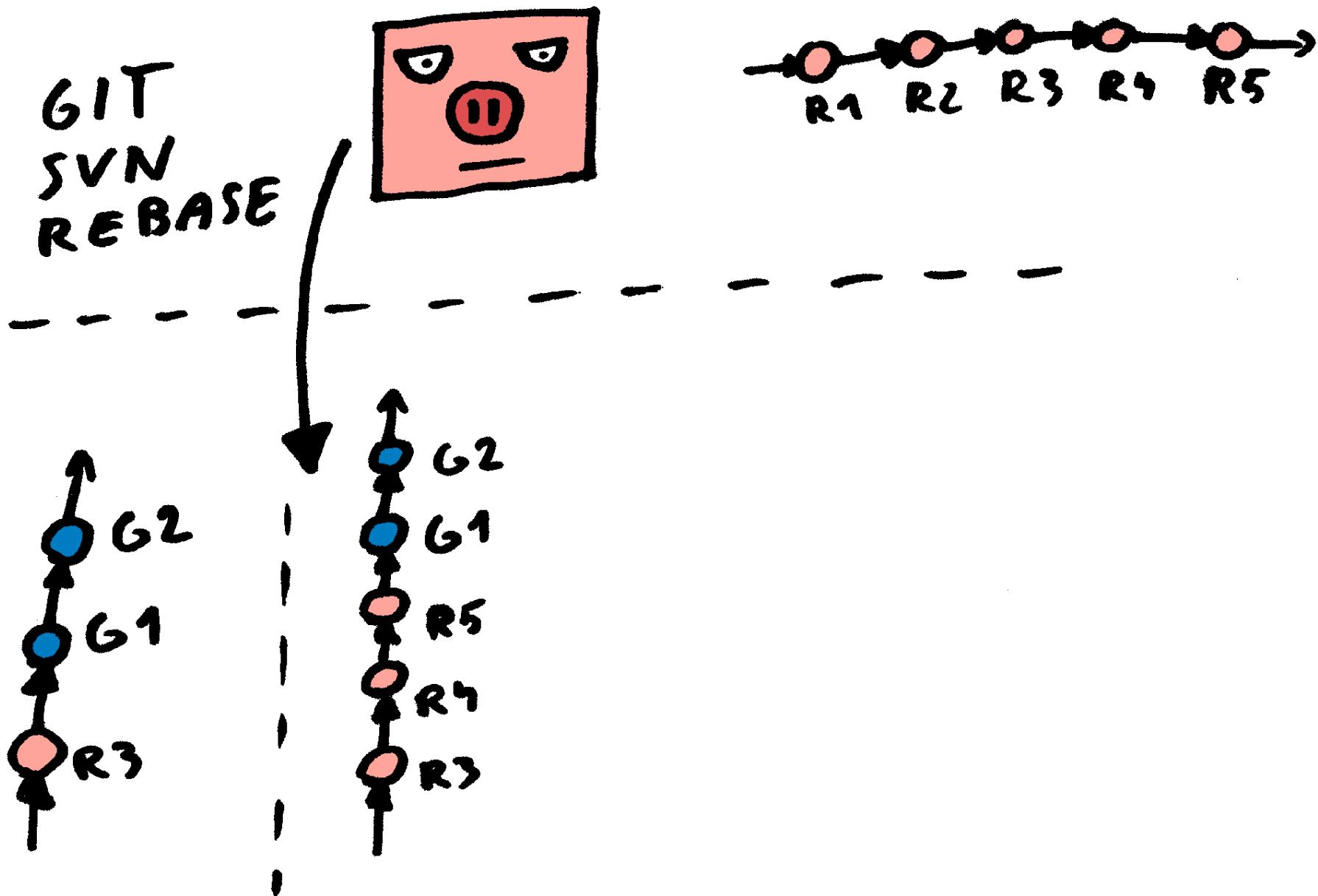


Git svn ...

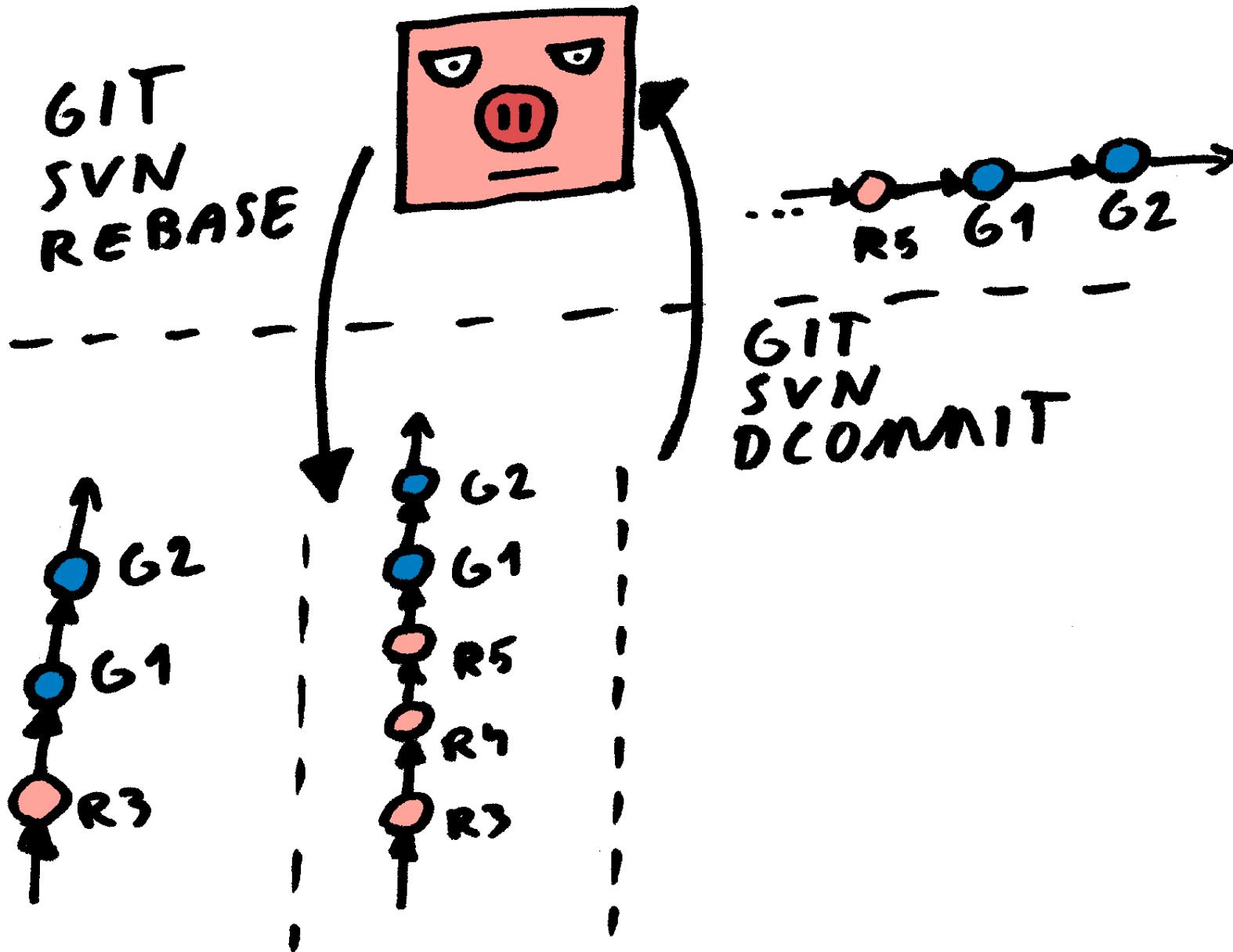




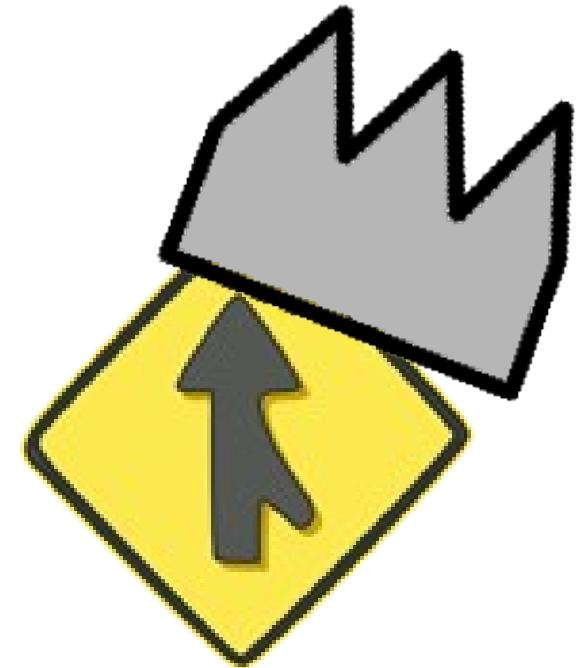
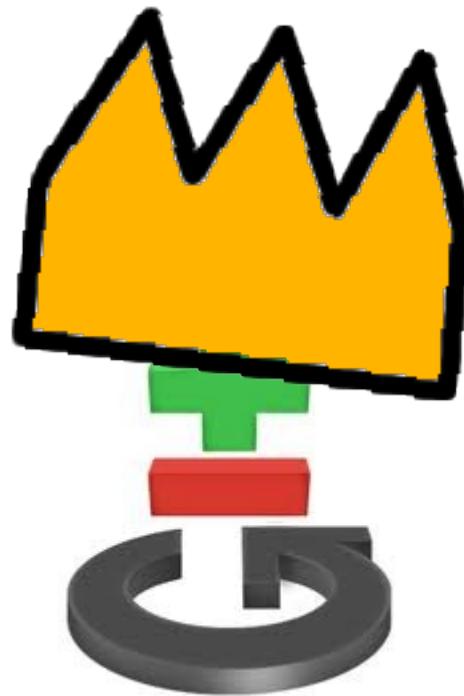
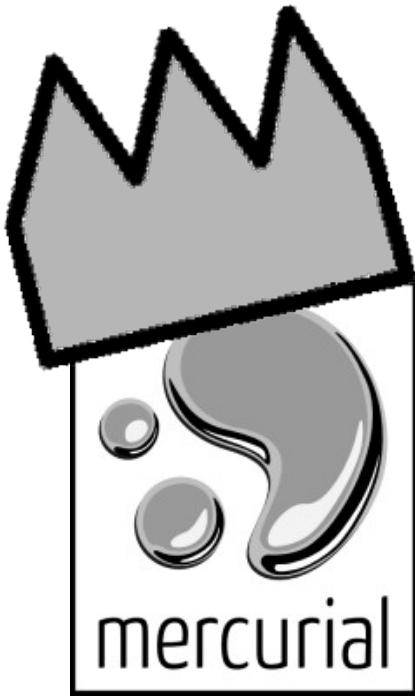
Git svn ...



Git svn ...

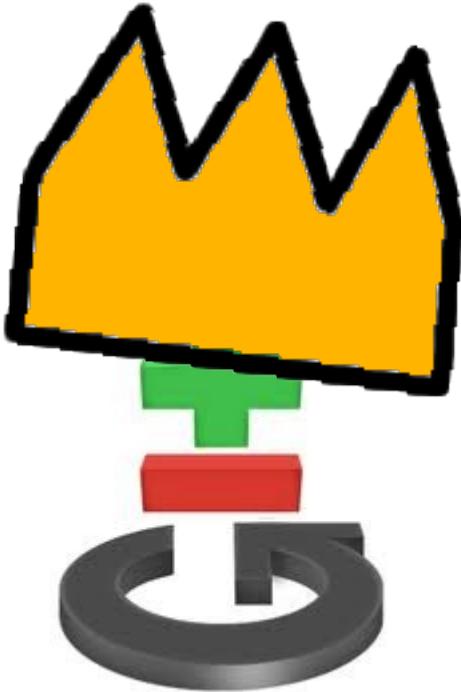


Poznaliśmy również podstawy:



Bazaar

Final bunch of links:



Best Resources:

- Git-scm.com
- Git.wiki.kernel.org
- Progit.org (Creative Commons Book)
- Gitref.org
- ...blogs!

Public OpenSource (or \$\$\$ private) hosting:

- Github.com
- Sourceforge.net
- Assembla.com
- Gitorious.org
- Gitenterprise.com

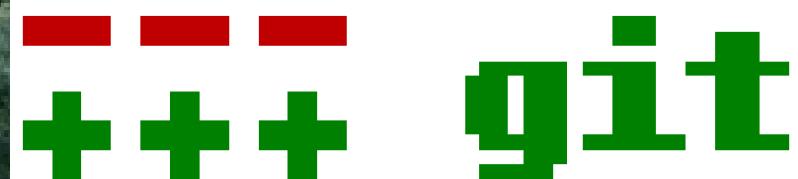
Private Hosting:

- Google -> gitosis
- Google -> gitolite

Other:

- Google -> git refcard
- Google -> successful git workflow

カピバラさん
ありがとう
(DZIĘKUJĘ)
ZA UWAGĘ!



Pytanie Bonusowe...



```
git init .
touch me
mkdir folder
git stage .
git commit -am 'Initial commit'
tree .git/objects/
fortune > anything.txt
git add .
git commit -am \
'How many git objects exist now ?'
```