A deep dive into the Git internals

Christoph Häfner
@chr1shaefn3r

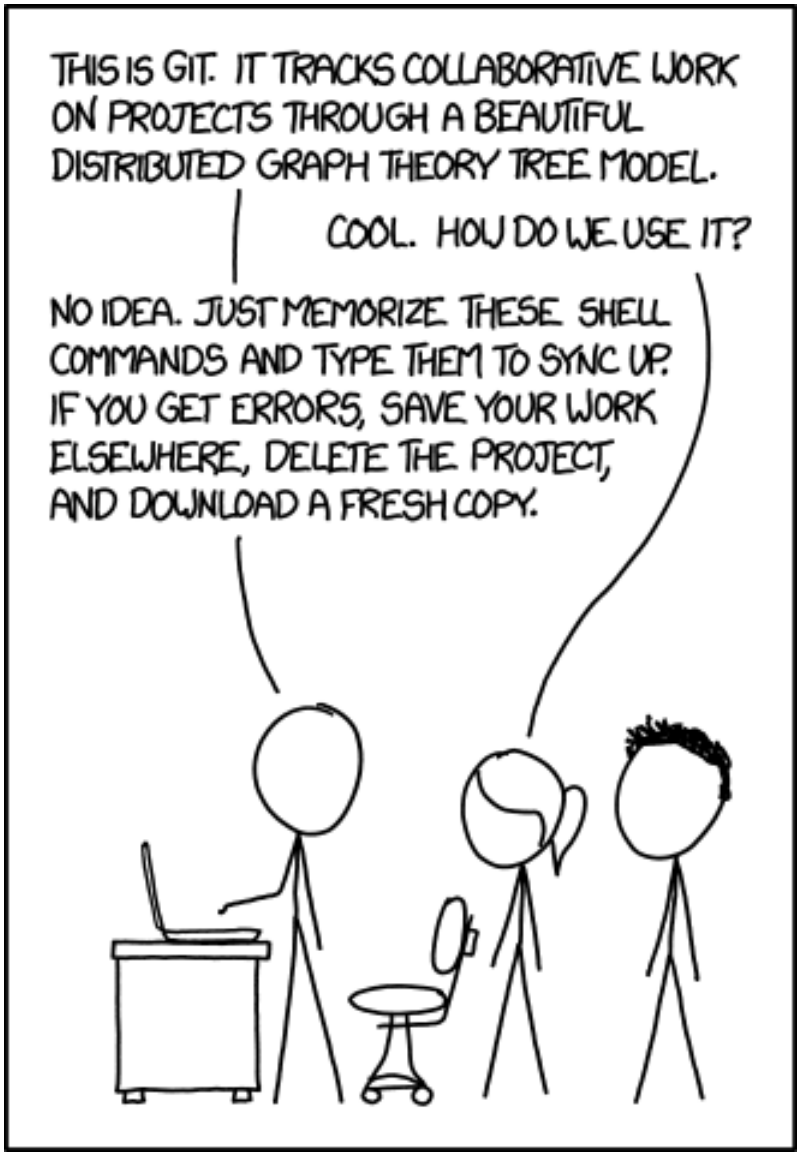
Quick Topic Clarification

We are going to talk about:

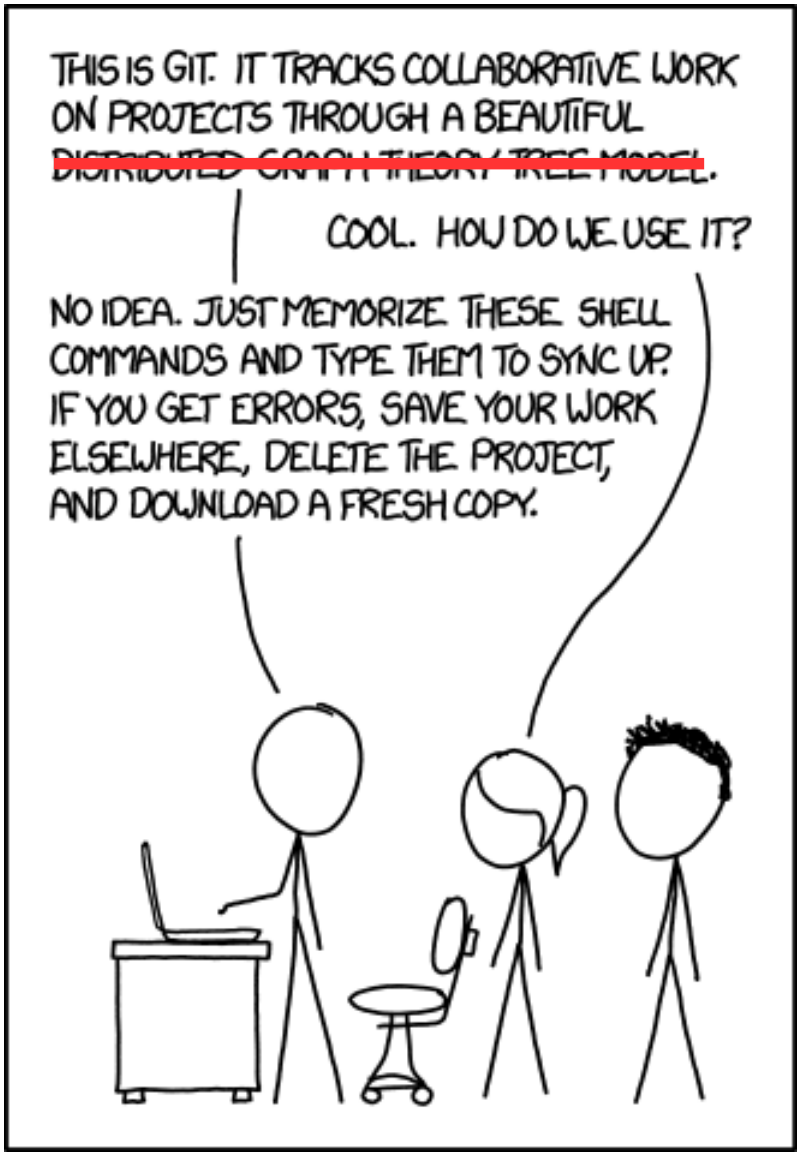
- Git Plumbing
- .git/objects/ (the object database)
- .git/refs/

We are NOT going to talk about:

- ~~Basics~~
- ~~Merge strategies~~
- ~~git:// Protokoll~~



https://xkcd.com/1597/

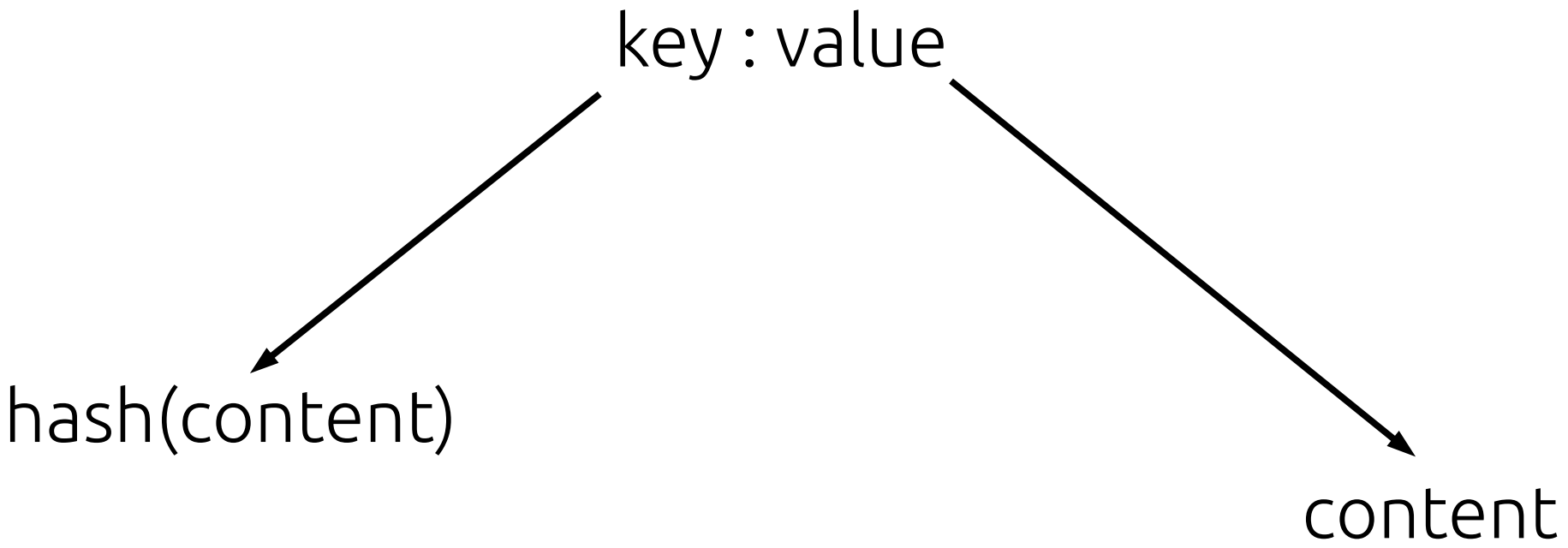

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL ~~DISTRIBUTED GRAPH THEORY TREE MODEL~~.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Plumbing vs. Porcelaine

hash-object
cat-file
ls-tree
mktree
commit-tree
fsck
gc
index-pack
ls-files
mktag
pack-objects
repack
verify-pack

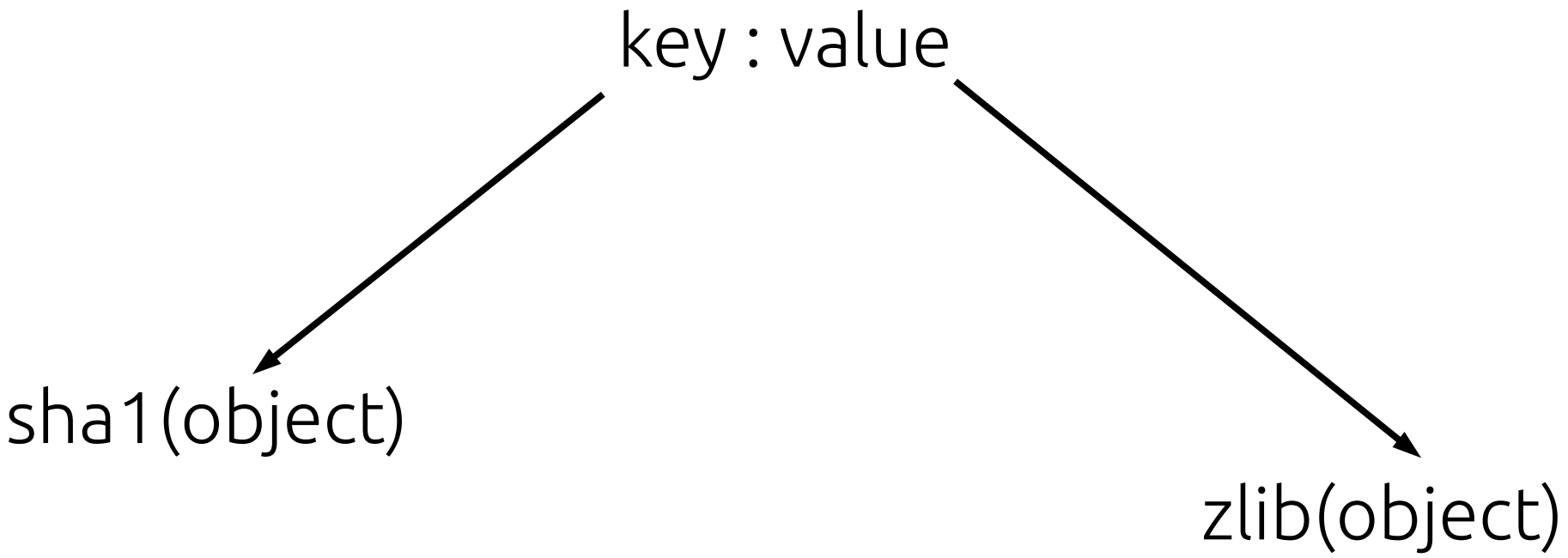
commit
push
pull
merge
branch
checkout
clone
bisect
tag
diff

Core Principle



aka. Content-adressable storage

Git Object



Format:

<type>␣<size>\0<value>

Four Types of Git Objects

blob

filecontent

tree

<mode>␣<filename>\0bin(<sha1>)

commit

tree␣<sha1>\n
parent␣<sha1>\n{0,2}
author␣<name>␣<timestamp+zeitzone>\n
committer␣<name>␣<timestamp+zeitzone>\n\n
<commit message>

tag

object␣<sha1>\n
type␣<type>\n
tag␣<name>
tagger␣<name>␣<timestamp+zeitzone>\n\n
<tag message>

Wo werden in Git eigentlich die Diffs gespeichert?!?!

Quellen

https://www.youtube.com/watch?v=4FUAaPyhOm0

https://git-scm.com/book/en/v2/Git-Internals-Plumbing-and-Porcelain

https://www.npmjs.com/package/gitviz

Ende

Christoph Häfner
SAP SE
@chr1shaefn3r
christoph.haefner@mailbox.org