



03

Gitshop

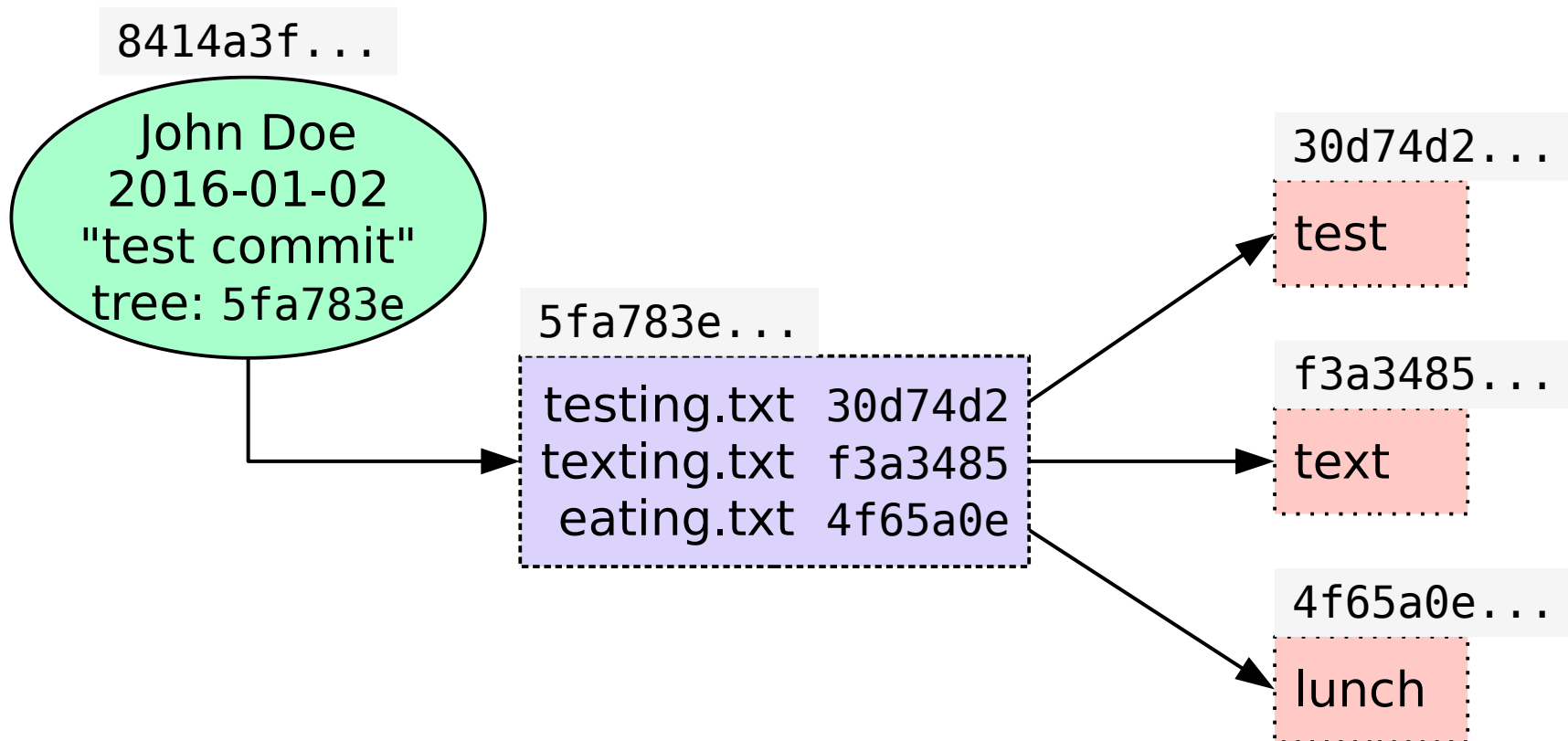
Git - bits and bobs

Jiří Jabůrek (jjaburek)
Red Hat 2016



Recap

Objects

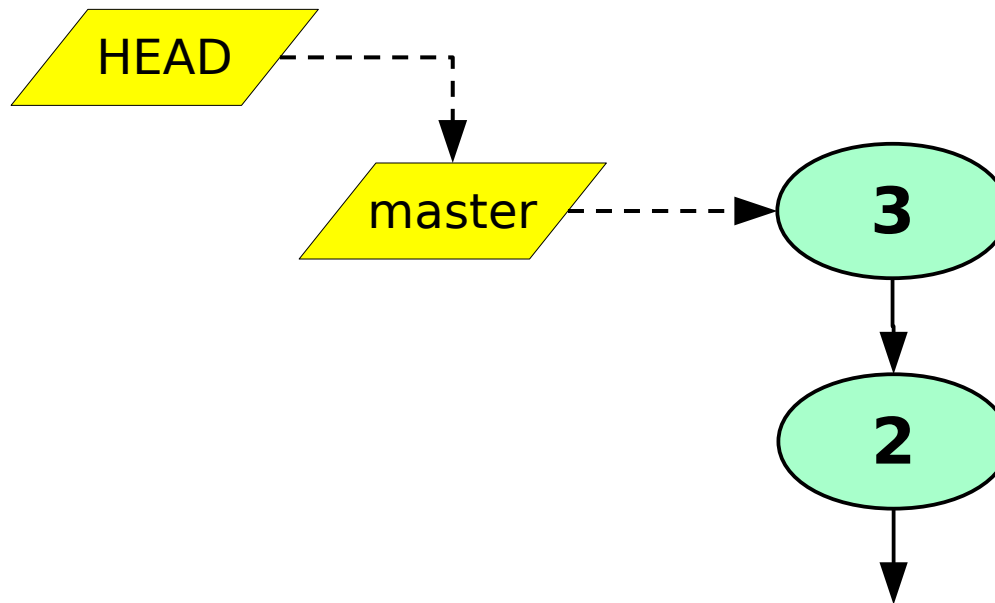


References

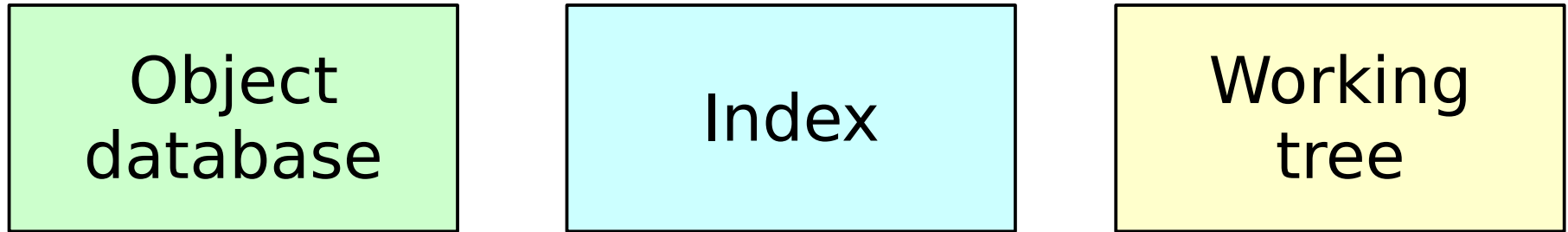
ref

■ Point to objects

- like hardlinks: 929c2ee5ca19b576b87f43f8cdc44eac447e8404
- like symlinks: ref: path/to/another/ref



The Index



- bare vs non-bare repository
- idx entry:
 - 4B: ctime (seconds) + 4B: ctime (nanoseconds)
 - 4B: mtime (seconds) + 4B: mtime (nanoseconds)
 - 4B: dev id
 - 4B: inode nr.
 - 4B: filemode
 - 4B: uid + 4b: gid
 - 4B: file size
 - ...



Missing bits

(no bobs)

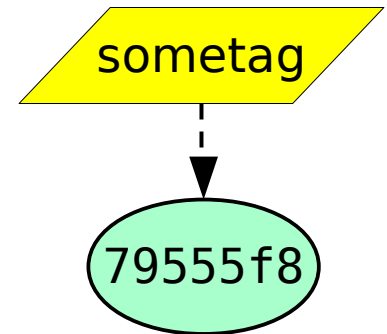
Objects - tags

■ Point to objects

□ "lightweight" - as refs

- refs/tags/sometag:

79555f8777798c58bbd12994d884575c479cb00f7



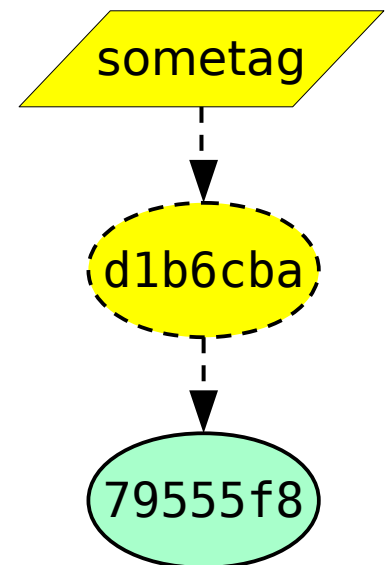
□ "announced" - as objects

- refs/tags/sometag:

d1b6cbab93aeb521c49ff3c87abd522dc8a7376f

- inside tag object:

79555f8777798c58bbd12994d884575c479cb00f7



Objects - annotated tag

tag

- Annotates referenced object
 - can hold any text - even GPG signatures

```
d1b6cbab93aeb521c49ff3c87abd522dc8a7376f
```

```
tag 152\0object 79555f877798c58bbd12994d884575c479cb00f7\n
type commit\n
tag mysupertag\n
tagger Jiri Jaburek <jjaburek@redhat.com> 1466524138 +0200\n
\n
some tag message\n
```


Packed refs

ref

■ Lines, not files

□ files:

```
.git/refs/heads/master  
.git/refs/tags/sometag  
.git/refs/remotes/origin/master
```

- each with ie. `79555f877798c58bbd12994d884575c479cb00f7`
 - or `ref: refs/heads/somebranch`

□ lines (in `.git/packed-refs`):

```
3d5b6dc78d3746bcd3b85b8bc7fad34fa07b3a72 refs/heads/master  
79555f877798c58bbd12994d884575c479cb00f7 refs/tags/sometag  
2a97406ba86b9450bedd5b723615ed8343c2a75e refs/remotes/origin/master  
d1b6cbab93aeb521c49ff3c87abd522dc8a7376f refs/heads/mysupertag  
^79555f877798c58bbd12994d884575c479cb00f7
```

- symbolic refs are ignored



Some bobs

(no bits)

Bobs

```
HEAD
branches/
config
description
hooks/
index
info/
info/attributes
info/exclude
info/grafs
info/refs
info/sparse-checkout
logs/
objects/
objects/info/
objects/info/alternates
packed-refs
refs/
refs/notes/
refs/replace/
shallow
```

Bobs

```
HEAD
branches/
config
description
hooks/
index
info/
info/attributes
info/exclude
info/grafts
info/refs
info/sparse-checkout
logs/
objects/
objects/info/
objects/info/alternates
packed-refs
refs/
refs/notes/
refs/replace/
shallow
```

Reflogs

```
HEAD
branches/
config
description
hooks/
index
info/
info/attributes
info/exclude
info/grafts
info/refs
info/sparse-checkout
logs/
objects/
objects/info/
objects/info/alternates
packed-refs
refs/
refs/notes/
refs/replace/
shallow
```

Reflogs

■ Changes of refs, in time

□ files:

```
.git/logs/HEAD  
.git/logs/refs/<anything>
```

- each with entries defined in `git-update-ref(1)`:

```
oldsha1 SP newsha1 SP committer LF  
oldsha1 SP newsha1 SP committer TAB message LF
```

- committer is `Name <email> timestamp +timezone`

```
53256a8.. bb4ec52.. John <jdoe@mail> 1468334259 +0200 \  
    pull --rebase origin testbr: my commit  
bb4ec52.. bb4ec52.. John <jdoe@mail> 1468334259 +0200 \  
    rebase finished: returning to refs/heads/curbr  
bb4ec52.. 9aa5181.. John <jdoe@mail> 1468340224 +0200 \  
    commit: add more documentation  
9aa5181.. eaaa520.. John <jdoe@mail> 1470678582 +0200 \  
    checkout: moving from curbr to origin/testbr
```

Reflogs

- Changes of refs, in time
 - porcelain: `git-reflog(1)`

```
$ git reflog -4
f52620c HEAD@{0}: checkout: moving from curbr to origin/testbr
9aa5181 HEAD@{1}: commit: add more documentation
bb4ec52 HEAD@{2}: rebase finished: returning to refs/heads/curbr
bb4ec52 HEAD@{3}: pull --rebase origin testbr: my commit

$ git checkout HEAD@{2}

$ git checkout HEAD@{10.minutes.ago}

$ git log -10 some-feature@{2.months.ago}

$ git reset --hard master@{8.hours.ago}
```

Notes

```
HEAD
branches/
config
description
hooks/
index
info/
info/attributes
info/exclude
info/grafts
info/refs
info/sparse-checkout
logs/
objects/
objects/info/
objects/info/alternates
packed-refs
refs/
refs/notes/
refs/replace/
shallow
```


Notes

■ Sticky labels for commits

- don't touch the original commit, are separate
- not cloned/fetched by default
- files:

```
.git/refs/notes/commits      (default)  
.git/refs/notes/<anything>
```

- kind of like branches, with commits, trees, blobs
- hashes of labeled commits stored as blob names

Notes

```
$ git notes add -m 'test notes over here' HEAD

$ git log -1
commit f1b1875775154fab4f0270965876d2991b419ec7
Author: Jiri Jaburek <jjaburek@redhat.com>
Date:   Wed Sep 21 13:48:39 2016 +0200

    test commit

Notes:
    test notes over here

$ git diff refs/notes/commits{^,}
...
new file mode 100644
index 0000000..39934fc
--- /dev/null
+++ b/f1b1875775154fab4f0270965876d2991b419ec7
@@ -0,0 +1 @@
+test notes over here
```

Replace

```
HEAD
branches/
config
description
hooks/
index
info/
info/attributes
info/exclude
info/grafts
info/refs
info/sparse-checkout
logs/
objects/
objects/info/
objects/info/alternates
packed-refs
refs/
refs/notes/
refs/replace/
shallow
```

Replace

- Sneaky injection of commits

- simple "swap" of one commit for another
 - parent traversal uses parents of the new commit
- not cloned/fetched by default
- files:

```
.git/refs/replace/<hash>
```

- file name represents commit to replace
- file contents are ASCII commit hash of replacement

```
$ cat .git/refs/replace/4fb4eaf*  
cd279e3695b9ba2de5d8d5184303a503ac47692c
```

Replace

```
$ git log --oneline -5
7b648f2 fourth commit
d767b9e third commit
4fb4eaf second commit
ca9a048 first commit
```

```
$ git log --oneline -3 cd279e3
cd279e3 Fix unicode handling in the rt plugin
591fa83 Make tags available for tests as well
61cfef2 Simplify idonethis smoke test as data are changing
```

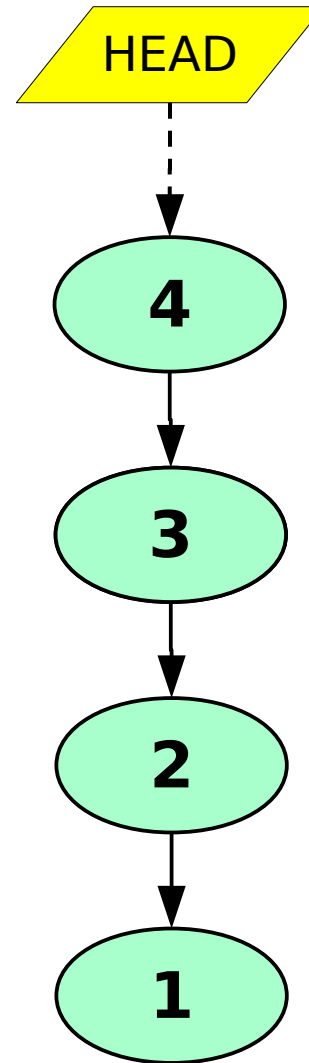
```
$ git replace d767b9e cd279e3
```

```
$ git log --oneline -5
7b648f2 fourth commit
d767b9e Fix unicode handling in the rt plugin
591fa83 Make tags available for tests as well
61cfef2 Simplify idonethis smoke test as data are changing
888336b Create vim tags using the 'make tags' target
```

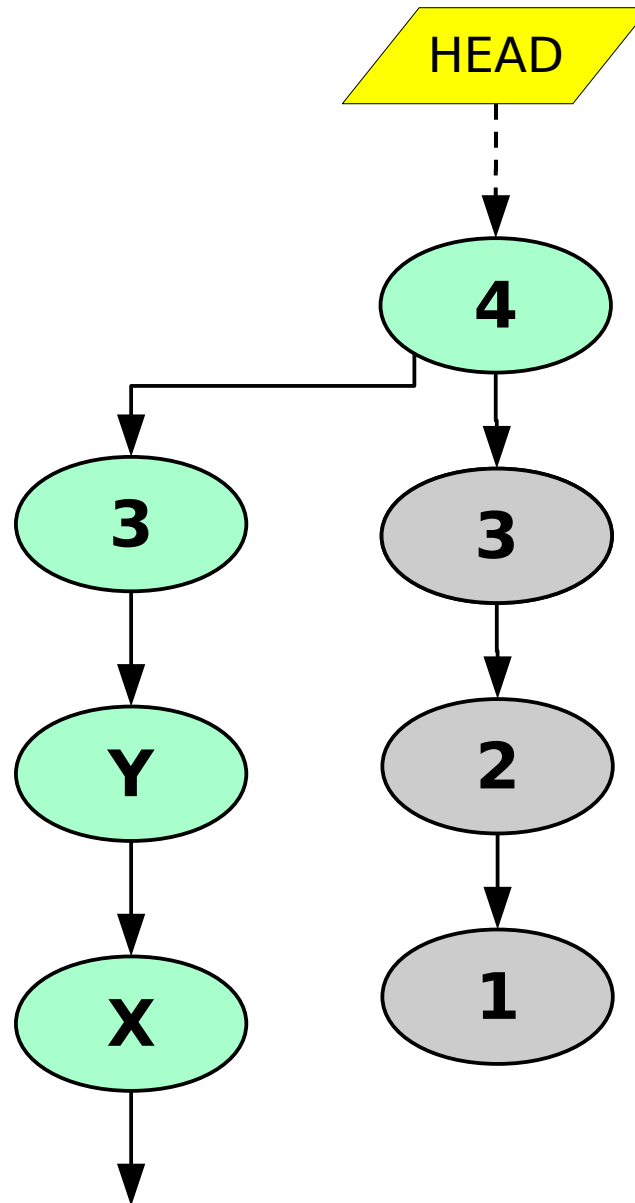
```
$ git log --format='format:%h %p%n' d767b9e -1
d767b9e 591fa83
```

```
$ rm -f .git/refs/replace/d767b9e*
$ git log --format='format:%h %p%n' d767b9e -1
d767b9e 4fb4eaf
```

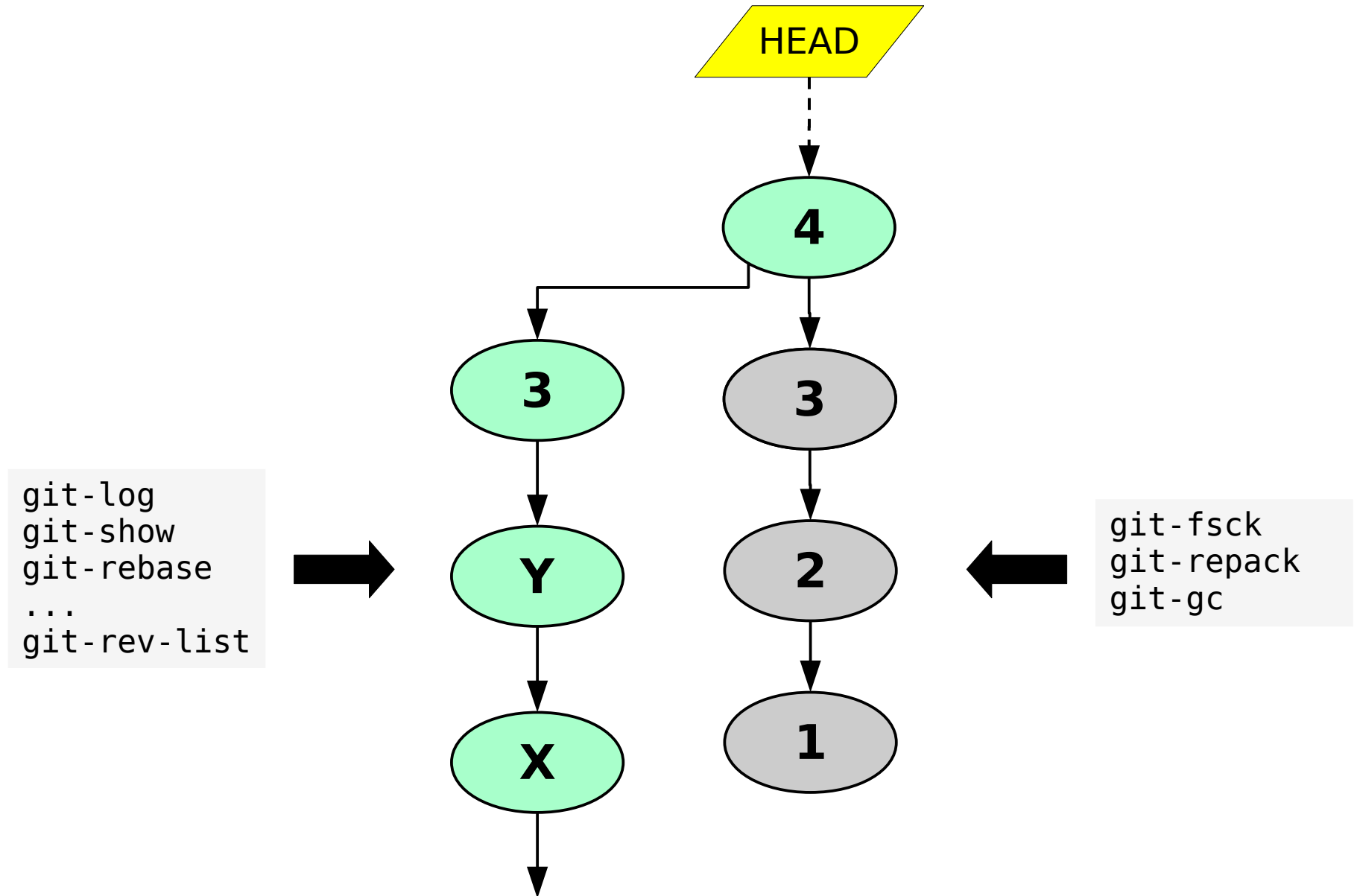
Replace



Replace



Replace



Replace

■ Grafts

- predecessor of `git-replace`
- simpler: it only fakes parents
- files:

`.git/info/grafts`

- one line per commit, format:

```
<commit> [parent1] [parent2] ...  
...
```

- commit without parents is a valid syntax

- hidden commits are ignored by `git-fsck` !!!
 - `git-repack` / `git-gc` will catch them



Extra bits

(without bobs)

Ignore

```
HEAD
branches/
config
description
hooks/
index
info/
info/attributes
info/exclude
info/grafts
info/refs
info/sparse-checkout
logs/
objects/
objects/info/
objects/info/alternates
packed-refs
refs/
refs/notes/
refs/replace/
shallow
```

Ignore

- Prevent adding untracked files
 - (a.k.a. prevent adding new files to index)
 - does nothing to tracked files
 - files:

```
.git/info/exclude    (local, not pushed)
.gitignore           (tracked file)
```

- contain match patterns, see `gitignore(5)`:

```
# custom git pathspecs
generated.config
builds/

# shell globs, see fnmatch(3)
*.exe
documentation/**/*.html
```

Attributes

```
HEAD
branches/
config
description
hooks/
index
info/
info/attributes
info/exclude
info/grafts
info/refs
info/sparse-checkout
logs/
objects/
objects/info/
objects/info/alternates
packed-refs
refs/
refs/notes/
refs/replace/
shallow
```

Attributes

- Define special handling of files

- (how files are displayed/diff-ed/added/.. by git)
- files:

```
.git/info/attributes    (local, not pushed)
.gitattributes          (tracked file)
```

- contain patterns like `gitignore(5)`, with attribute specs:

```
# checkout windows-related docs with CRLF always
windows-docs/      eol=crlf
*.vcproj           eol=crlf

# treat postscript as binary (in diffs, etc.)
*.ps               -text

# resolve $Id$ in Makefiles as commit hash on checkout
Makefile           ident
```

- see `gitattributes(5)`



Clone bobs

(git-clone features)

Alternates

```
HEAD
branches/
config
description
hooks/
index
info/
info/attributes
info/exclude
info/grafts
info/refs
info/sparse-checkout
logs/
objects/
objects/info/
objects/info/alternates
packed-refs
refs/
refs/notes/
refs/replace/
shallow
```


Alternates

- Borrow objects from another repo
 - read-only, any unique objects remain local
 - completely separate from `git-remote` functionality
 - files:
 - `.git/objects/info/alternates`
 - contains absolute path to another objects/ dir
 - can be kept long-term, with `git-repack --local`
 - objects may disappear in parent due to gc!
 - (if not referenced by a ref)

Alternates

```
$ git clone git://some-url-with/repo.git mainrepo
Cloning into 'mainrepo'...
remote: Counting objects: 23195, done.
remote: Compressing objects: 100% (6057/6057), done.
remote: Total 23195 (delta 16244), reused 22873 (delta 16025)
Receiving objects: 100% (23195/23195), 15.55 MiB | 0 bytes/s, done.
Resolving deltas: 100% (16244/16244), done.
Checking connectivity... done.

$ git clone --reference mainrepo git://some-url-with/repo.git testrepo
Cloning into 'testrepo'...
Checking connectivity... done.

$ du -sh mainrepo/.git/ testrepo/.git/
17M      mainrepo/.git/
224K     testrepo/.git/

$ cat testrepo/.git/objects/info/alternates
/home/jjaburek/mainrepo/.git/objects
```

Shallow clone

```
HEAD
branches/
config
description
hooks/
index
info/
info/attributes
info/exclude
info/grafts
info/refs
info/sparse-checkout
logs/
objects/
objects/info/
objects/info/alternates
packed-refs
refs/
refs/notes/
refs/replace/
shallow
```

Shallow clone

■ Getting partial history

- exactly like grafts - faking zero parents
- files:

`.git/shallow`

– same syntax as `.git/info/grafts`

- is recognized by `git-repack / git-gc`
 - grafted commit is hashed, parents not checked

```
commit 229\0tree 292778a18c54dea66754260b9a2d9ff8aef73b46\n
parent a95274e7931ab0f5108ab292140620750a0de8ff\n
author Jiri Jaburek <jjaburek@redhat.com> 1456166293 +0100\n
committer Jiri Jaburek <jjaburek@redhat.com> 1456166293 +0100\n
\n
some message\n
```

Shallow clone

```
$ git clone --depth 3 git://some-url-with/repo.git mainrepo
Cloning into 'mainrepo'...
remote: Counting objects: 735, done.
remote: Compressing objects: 100% (514/514), done.
remote: Total 735 (delta 360), reused 371 (delta 214)
Receiving objects: 100% (735/735), 962.05 KiB | 0 bytes/s, done.
Resolving deltas: 100% (360/360), done.
Checking connectivity... done.
```

```
$ cd mainrepo/
```

```
$ git log --oneline
5330b97 lsns: improve formatting and wording of the man page
28a2df8 libblkid: fix buffers mapping on 32bit archs
e03f29b line: keep stdin unbuffered
```

```
$ cat .git/shallow
e03f29b6941fcf71b03039b0bf9e8fec66c81b58
```

```
<commit> [parent1] [parent2] ...
...
```



Summary

Summary

- Annotated tags
 - refs pointing to tag objects pointing to commits
- Packed refs
 - lines in a file instead of files in directories
- Reflogs
 - logged versioned history for ref updates
- Notes
 - sticky notes for commits (or other objects)
- Replace / grafts
 - faking commit history (parents)
- Ignore (gitignore) and file attributes
 - not staging unwanted files, fixing broken diffs
- Alternates and shallow clone
 - saving disk space, living on the edge of history



Questions?

The End

Thanks for listening



Workshop >>

Workshop

- 1) **Create a repo, make several commits adding some files**
 - use `git commit --amend` to correct a commit
- 2) **Make any new *.txt files ignored** (man gitignore)
 - test with: `touch blah.txt ; git add . ; git status`
- 3) **Make anything *.pdf appear binary** (man gitattributes)
 - `echo text > blah.pdf ; git add . ; git diff --cached`
- 4) **Clone your repo into another one, sharing objects**
 - `git clone -s repo1 repo2 ; ls -R repo2/.git/objects`
 - `git log` in repo2 shows valid commits
- 5) **In repo1 , list reflog entries and hard-reset to the middle**
 - `git reflog , git reset --hard HEAD@{number}`
- 6) **Look for unpacked refs in repo1, pack them**
 - `ls -R .git/refs ; git pack-refs --all`
 - `cat .git/packed-refs`