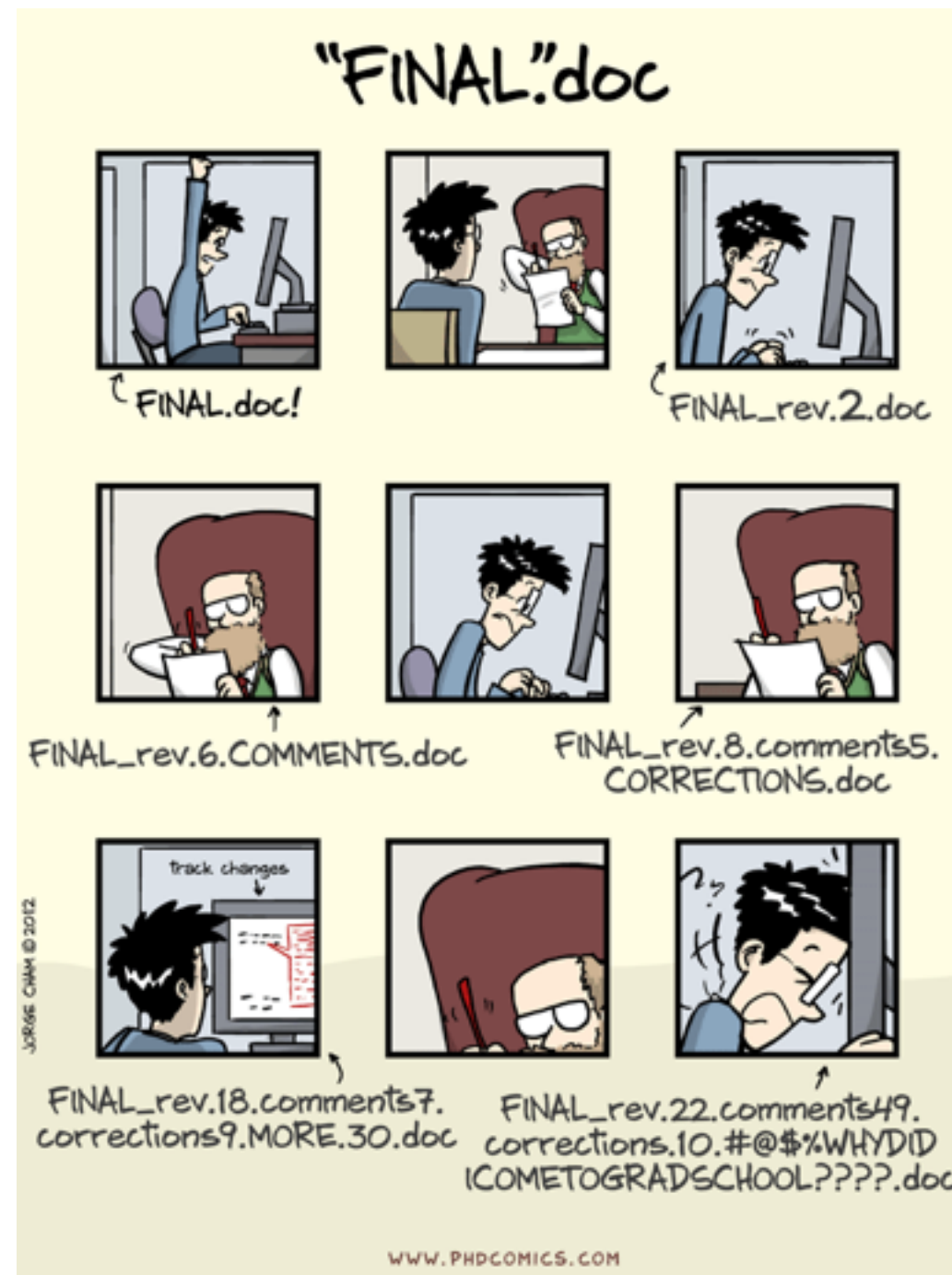


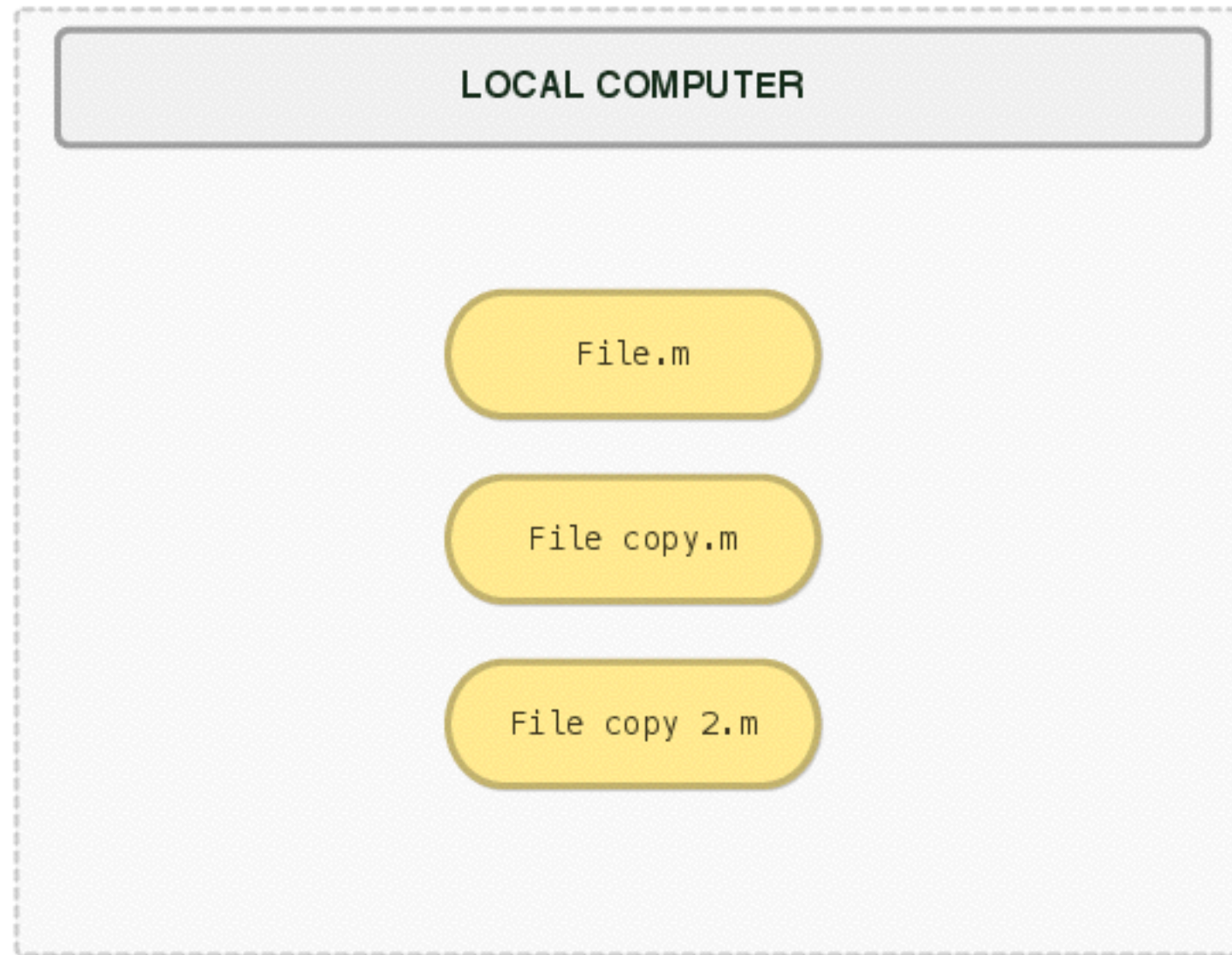
Nicola Cavallini

Basic Git  
(adapted from Luca Heltai,  
adapted from Patrick Hogan)

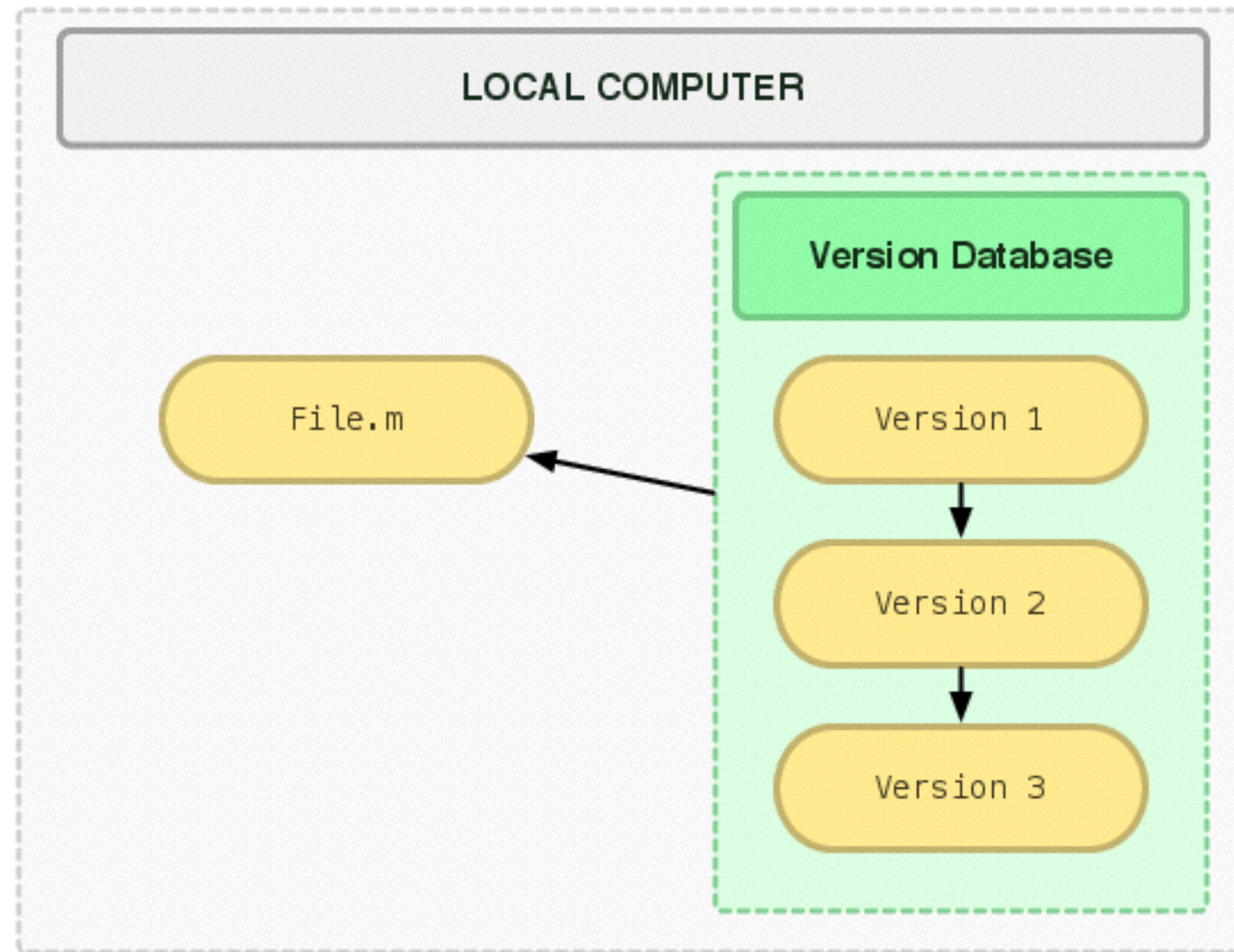
# Why Version Control?



# Local Filesystem

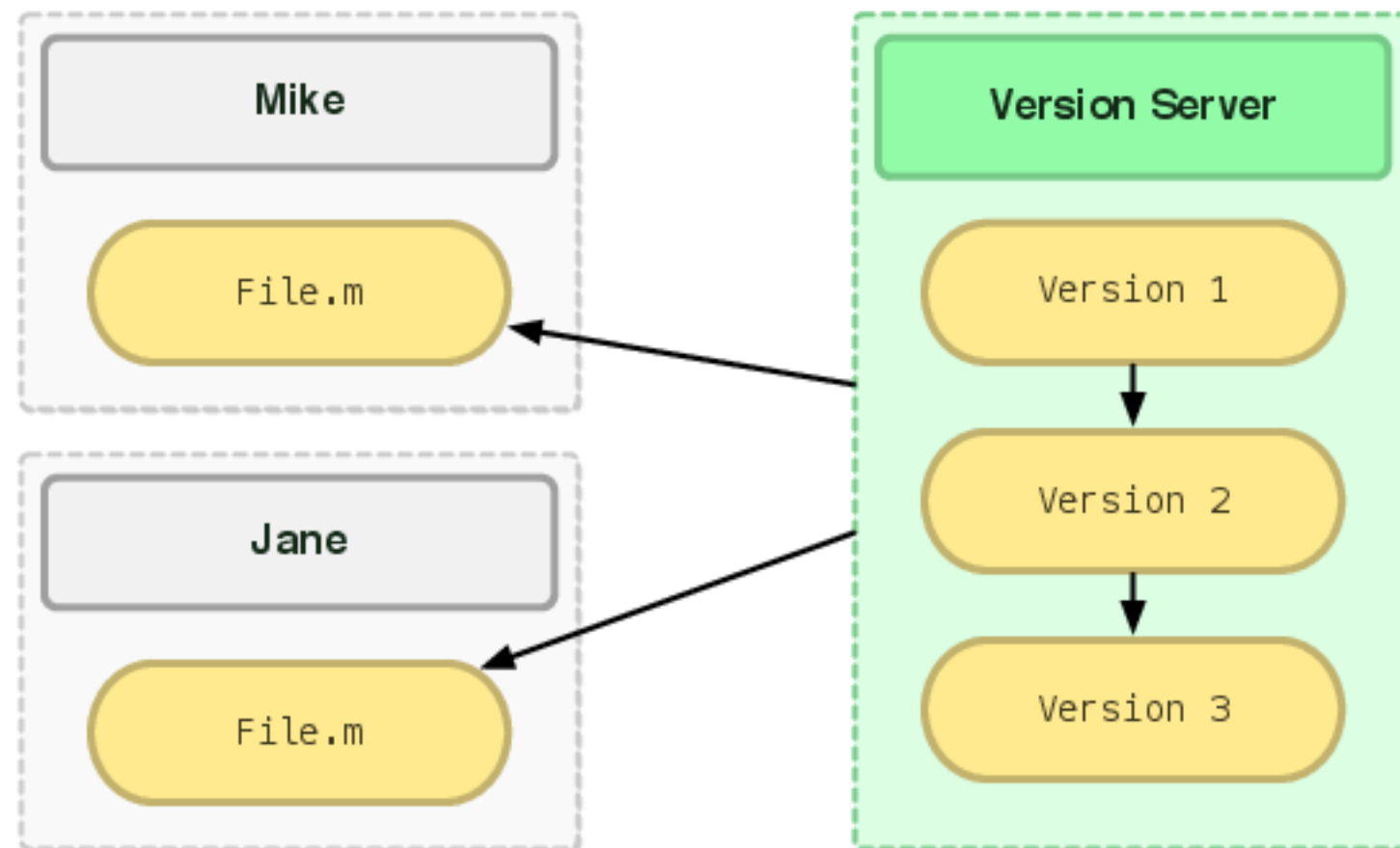


# LOCAL Version Control System



RCS: one image for each file or directory,  
each one with its own history.

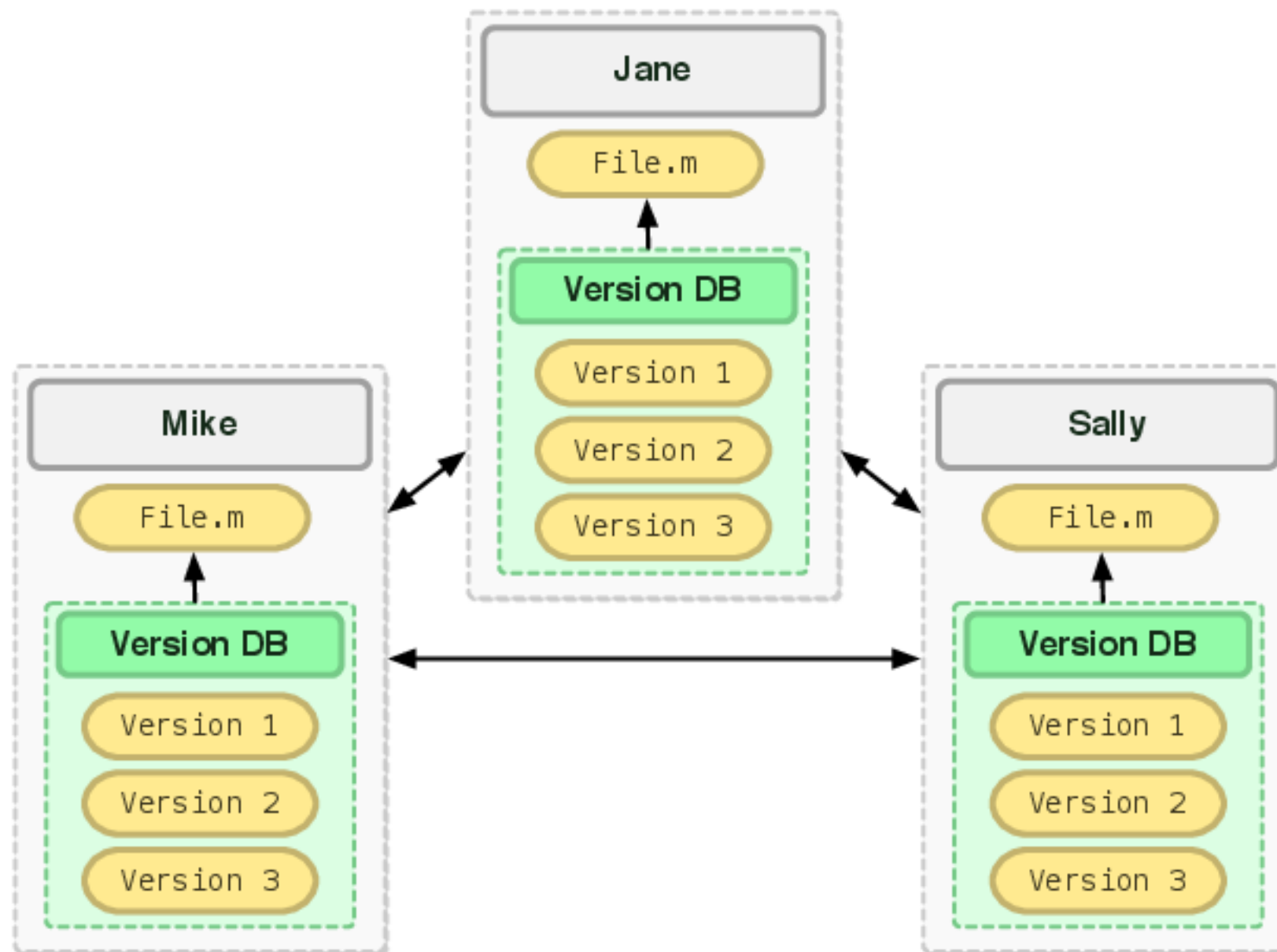
# CENTRALIZED Version Control System



CVS: centralized version of RCS,  
with tags that address single version for each file or dir.



# DISTRIBUTED Version Control System



# **Everything is Local**

## **(Almost)**

# No Network Required

**Create Repo**

**Status**

**Commit**

**Revisions**

**Merge**

**Diff**

**Branch**

**History**

**Rebase**

**Bisect**

**Tag**

**Local Sync**



# Advantages

**Everything is Fast**

**Everything is Transparent**

**Every Clone is a Backup**

**You Can Work Offline**

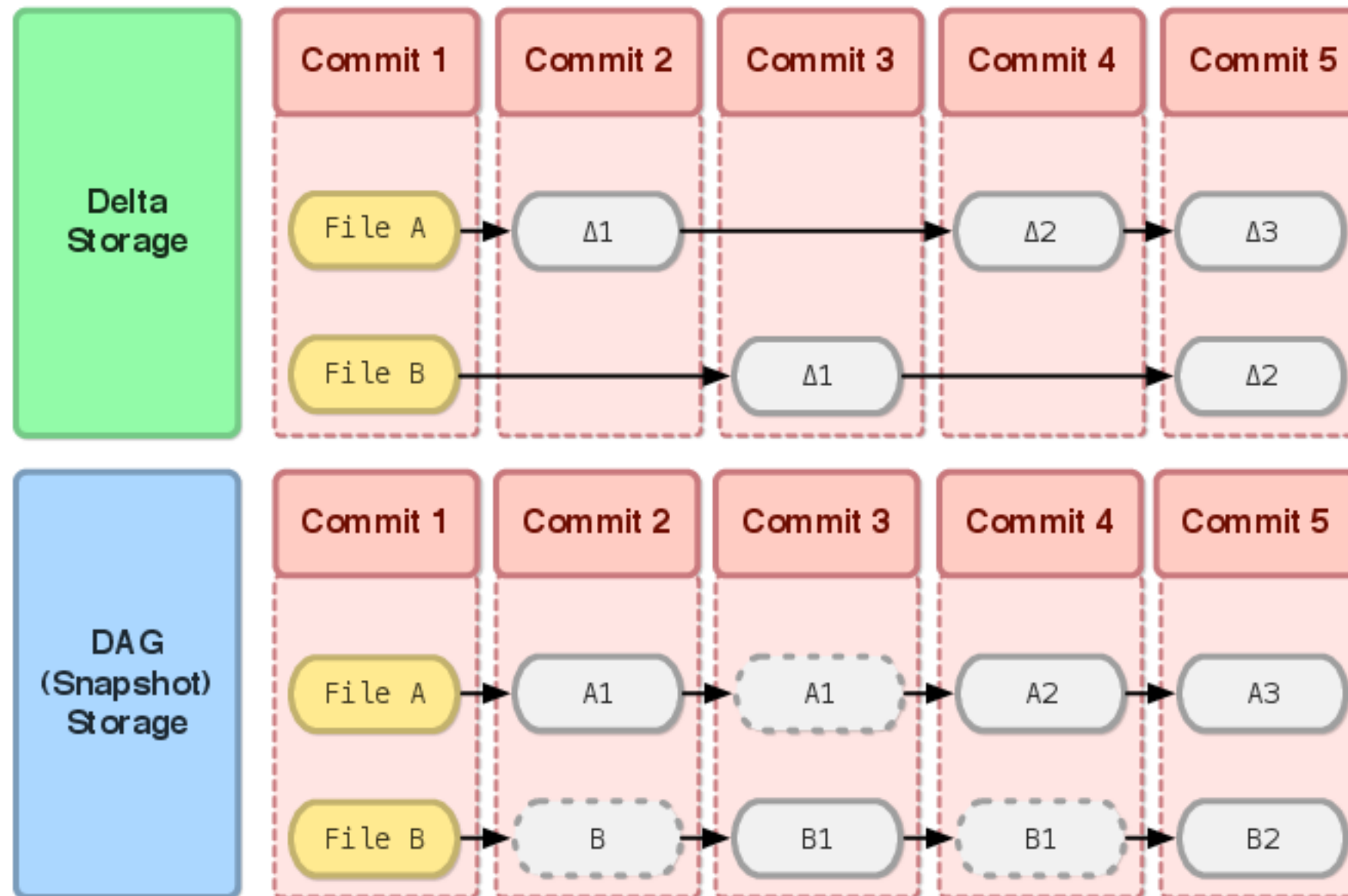
# Storage

# Delta Storage

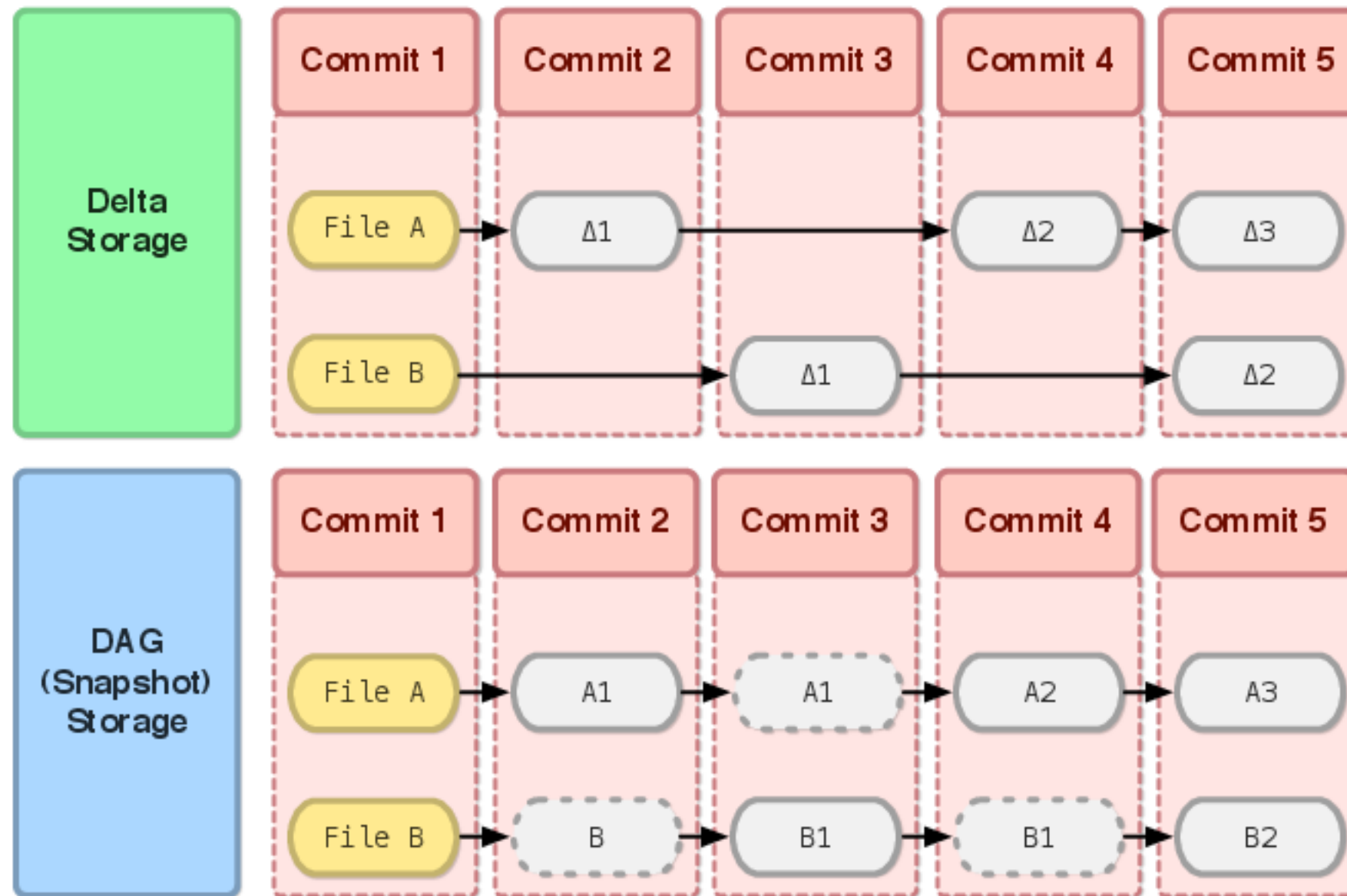
# **Snapshot Storage**

**(a.k.a. Direct Acyclic Graph)**

# Delta vs. Snapshot



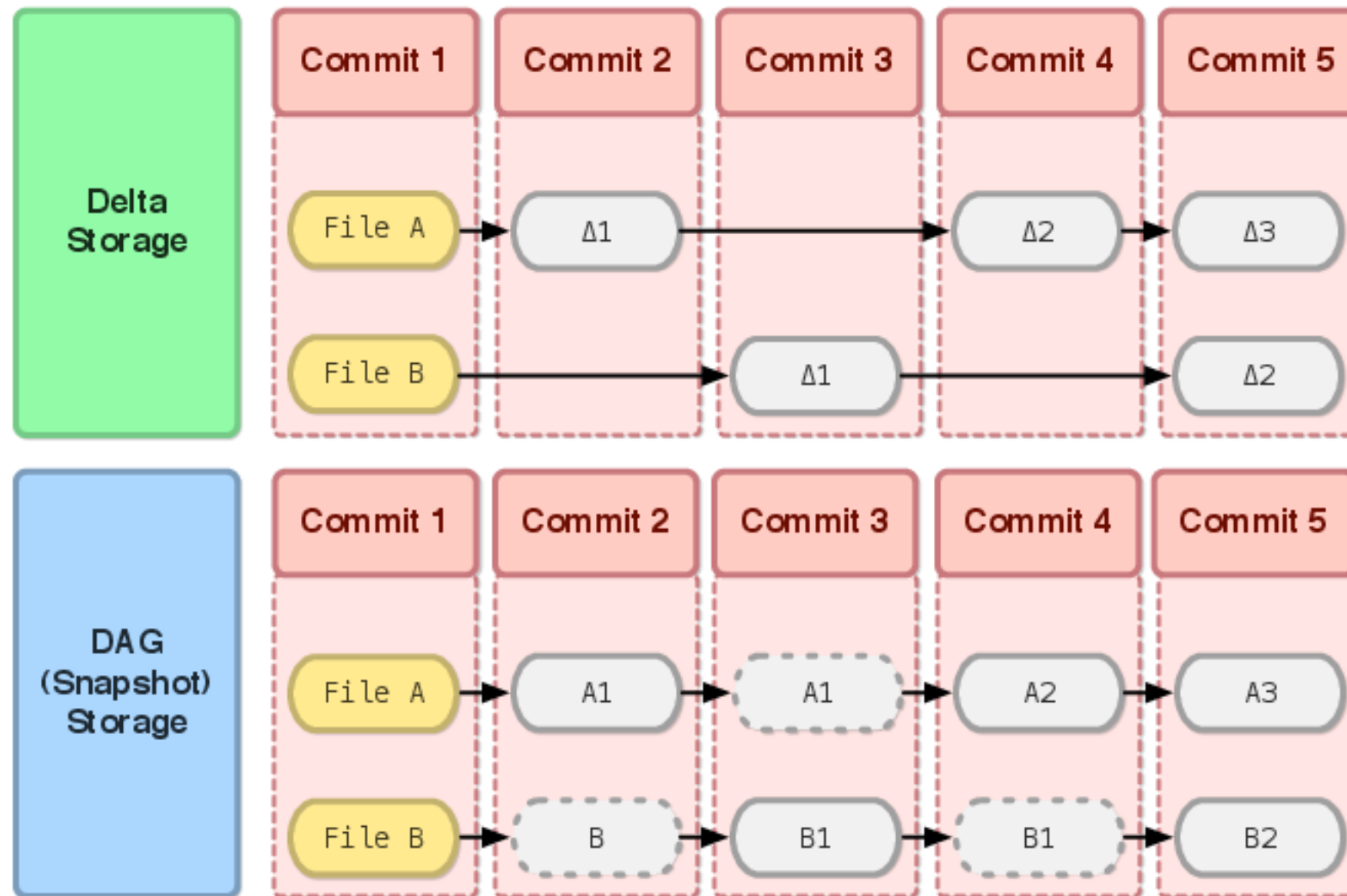
# Delta vs. Snapshot



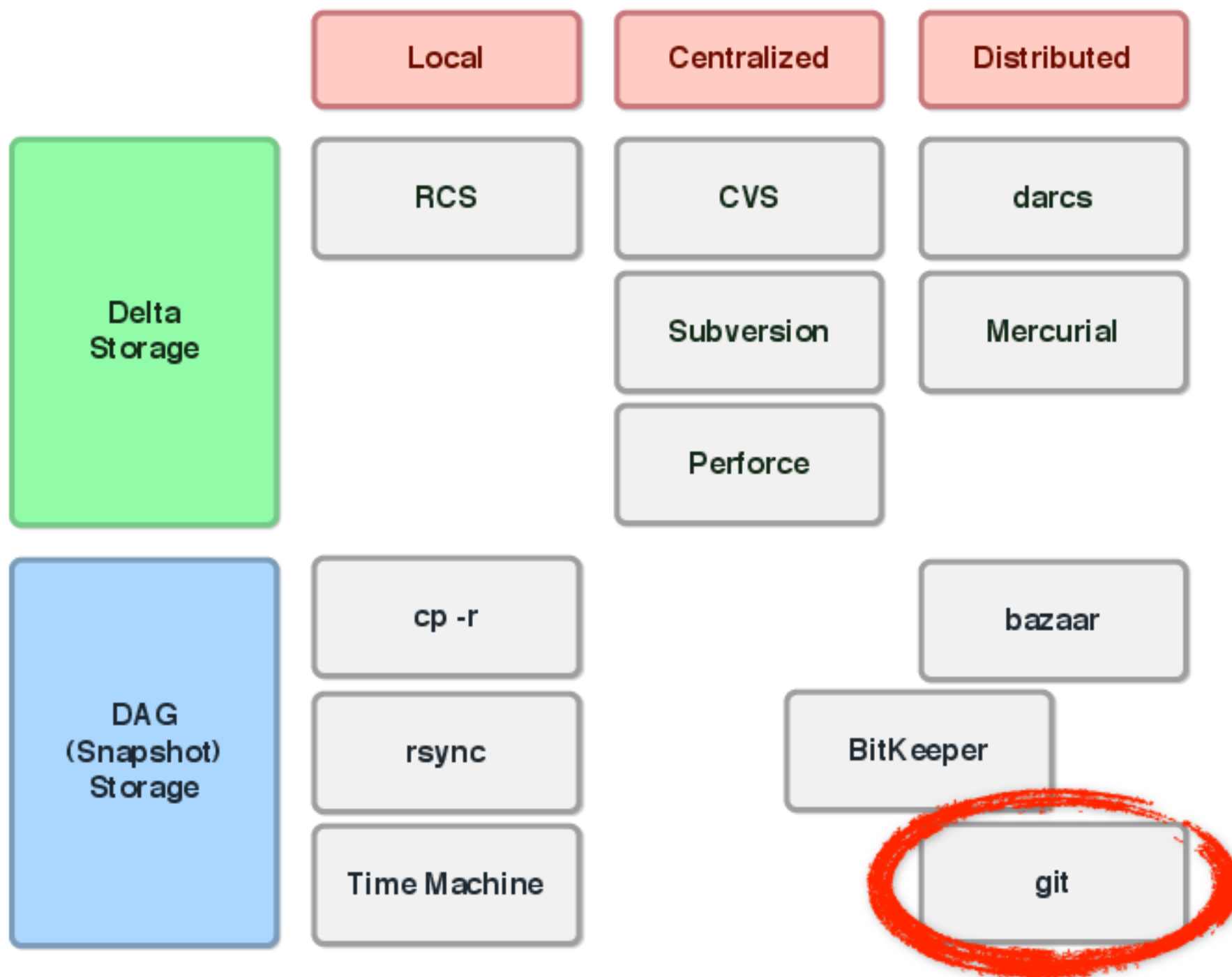
DELTA STORAGE, commit 3 and 4 are:  
**useless chunks of data.**



# Delta vs. Snapshot



**SNAPSHOTS STORAGE, commit 3 and 4 are:  
independent repository versions.**



# About Git

# **Git in a Nutshell**

**Free and Open Source**

**Distributed Version Control System**

**Designed to Handle Large Projects**

**Fast and Efficient**

**Excellent Branching and Merging**

# Projects Using Git

**Git**

**Linux**

**Perl**

**Eclipse**

**Qt**

**Rails**

**Android**

**PostgreSQL**

**KDE**

**Gnome**

# Under The Hood



# Git Directory

```
$ ls -lA
-rw-r--r--@ 1 pbhogan staff 21508 Jul  3 15:21 .DS_Store
drwxr-xr-x 14 pbhogan staff  476 Jul  3 14:46 .git
-rw-r--r--@ 1 pbhogan staff  115 Aug 11 2010 .gitignore
-rw-r--r--@ 1 pbhogan staff  439 Dec 27 2010 Info.plist
drwxr-xr-x 17 pbhogan staff  578 Feb  6 10:54 Resources
drwxr-xr-x  7 pbhogan staff  238 Jul 18 2010 Source
...
```

# Git Directory

```
$ tree .git
.git
├── HEAD
├── config
├── description
├── hooks
│   ├── post-commit.sample
│   └── ...
├── info
│   └── exclude
├── objects
│   ├── info
│   └── pack
└── refs
    ├── heads
    └── tags
```

**.git only in root of  
Working Directory  
(unlike Subversion)**

# **Git Directory**

**Configuration File**

**Hooks**

**Object Database**

**References**

**Index**

# Git Directory

~~Configuration File~~

~~Hooks~~

**Object Database**

**References**

**Index**

# Object Database



# Object Database

**≈ NSDictionary**  
**(Hash Table / Key-Value Store)**

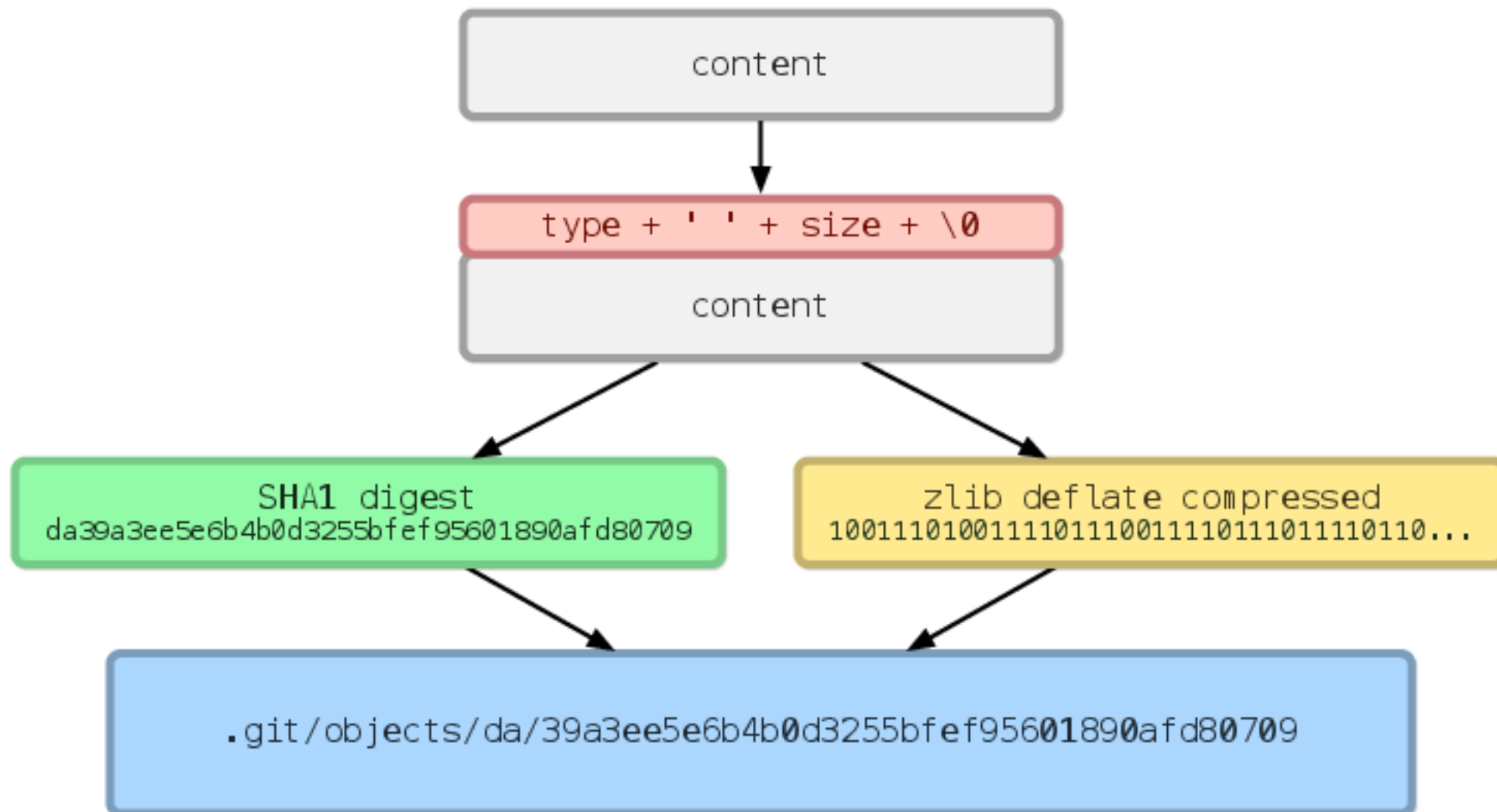
NSDictionary class represents an unordered collection of objects;  
each of this objects is associated to a key.

# Object Database

**What matters is the content of our repo.**

content

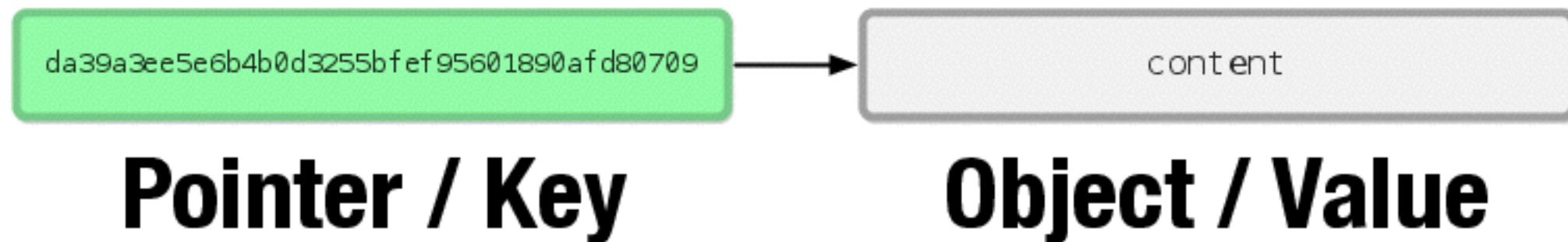
# Object Database



**"loose format"**

# Object Database

≈ NSDictionary



Secure Hash Algorithm (SHA): two equal SHA1 for the same content, do not exist.

# Object Database

da39a3ee5e6b4b0d3255bfe95601890afd80709

da39a3ee5e6b4b0d3255...

da39a3ee5e6...

da39a...

If the first  $n$  digits are unique, than these  $n$  digits are equivalent to the whole 16.

# Object Database

## Garbage Collection



# Object Database

## Similar Objects

`.git/objects/da/39a3...0709`

`.git/objects/e9/d71f...8f98`

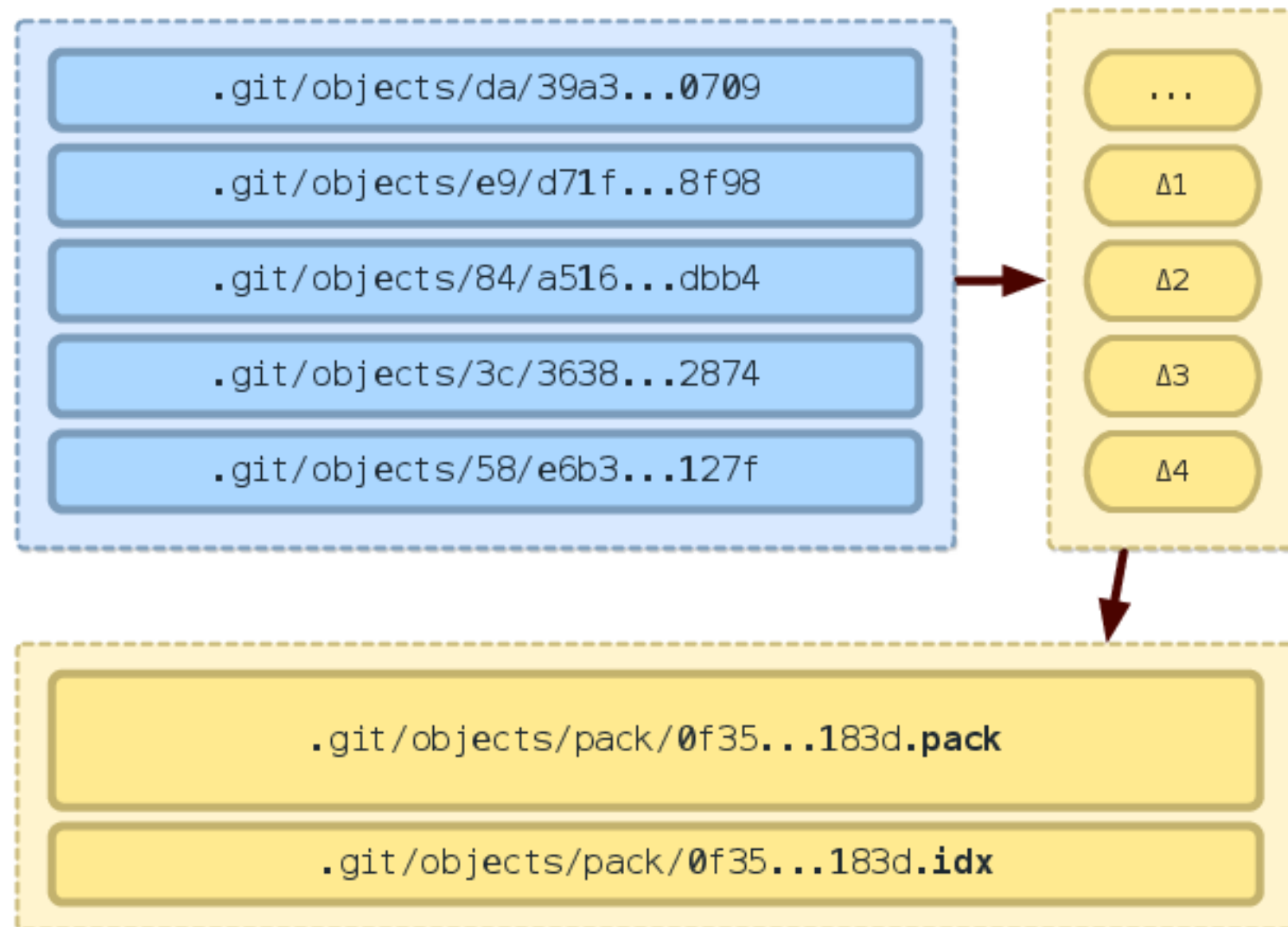
`.git/objects/84/a516...dbb4`

`.git/objects/3c/3638...2874`

`.git/objects/58/e6b3...127f`

**git gc**

# Object Database



**"packed format"**

# Object Database

**Four Object Types**

# Object Database

blob

tree

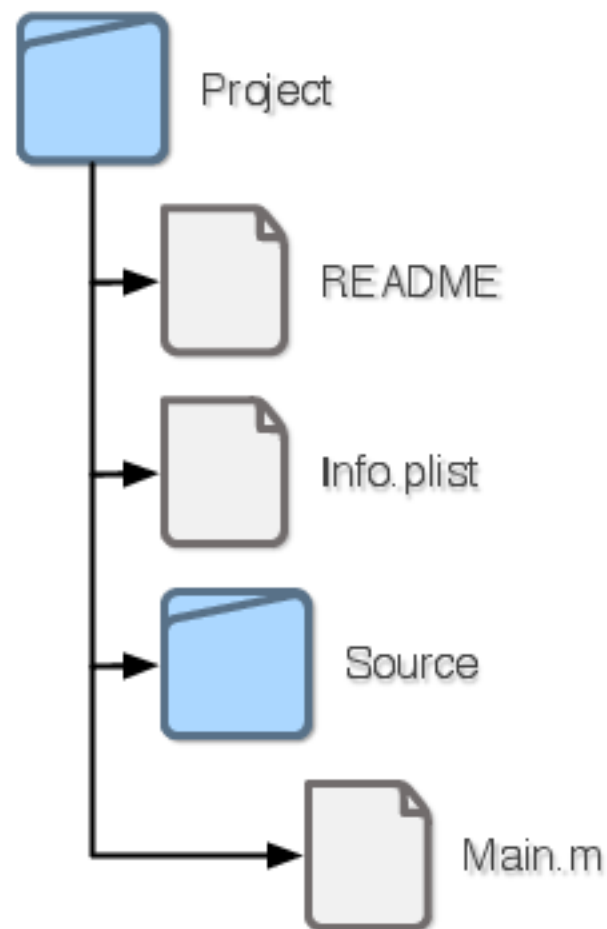
commit

tag

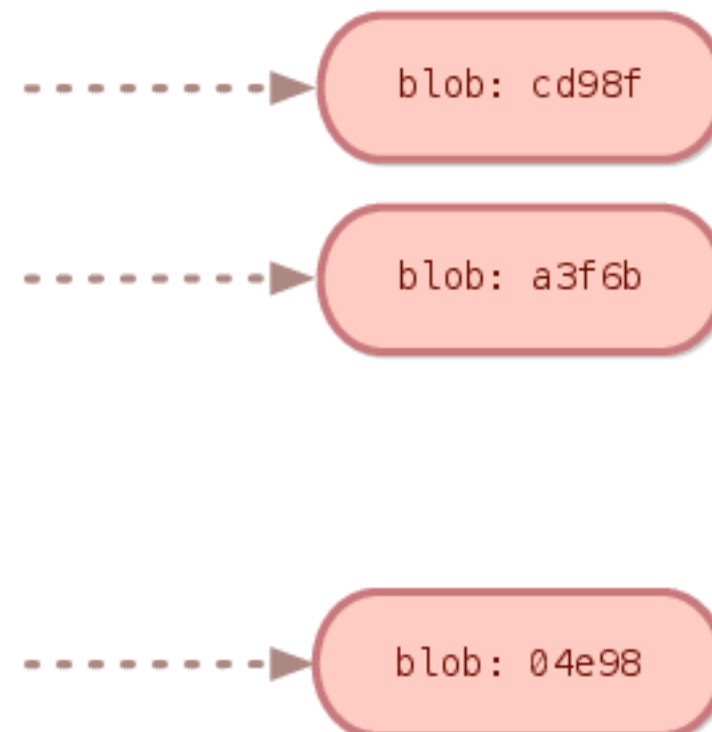
# Object Database

blob

## Working Directory



## Git Directory



# Object Database

blob

blob 109\0

```
#import <Cocoa/Cocoa.h>

int main(int argc, const char *argv[])
{
    return NSApplicationMain(argc, argv);
}
```

# Object Database

blob

tree

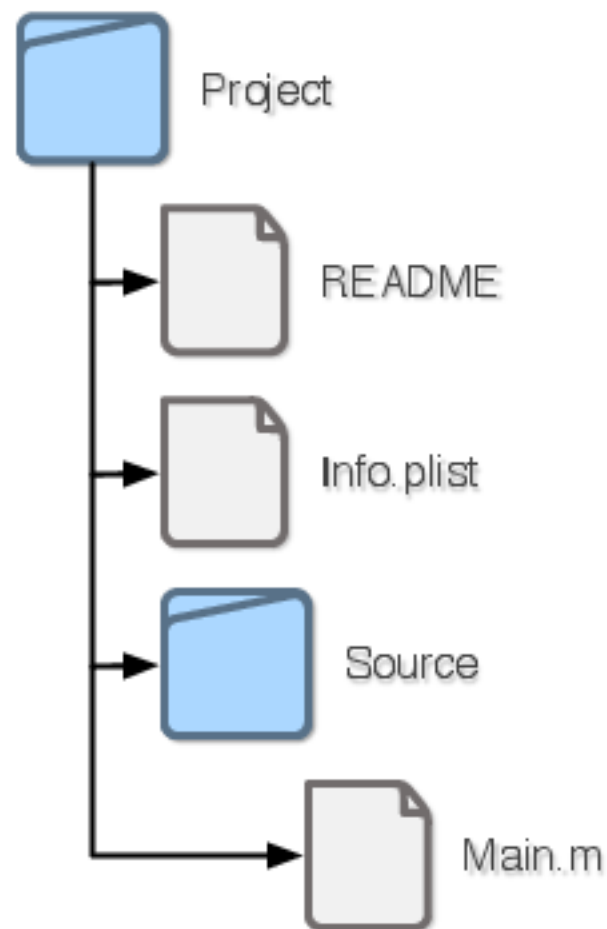
commit

tag

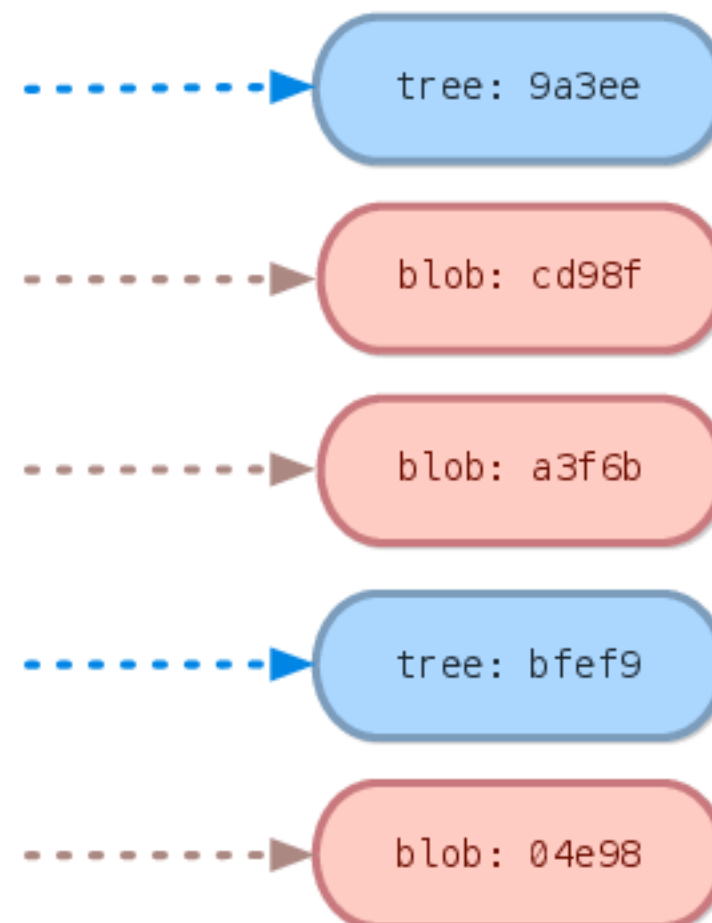
# Object Database

tree

## Working Directory



## Git Directory





# Object Database

tree

```
tree 84\0
```

```
100644 blob cd98f    README
100644 blob a3f6b    Info.plist
040000 tree bfef9    Source
```

# Object Database

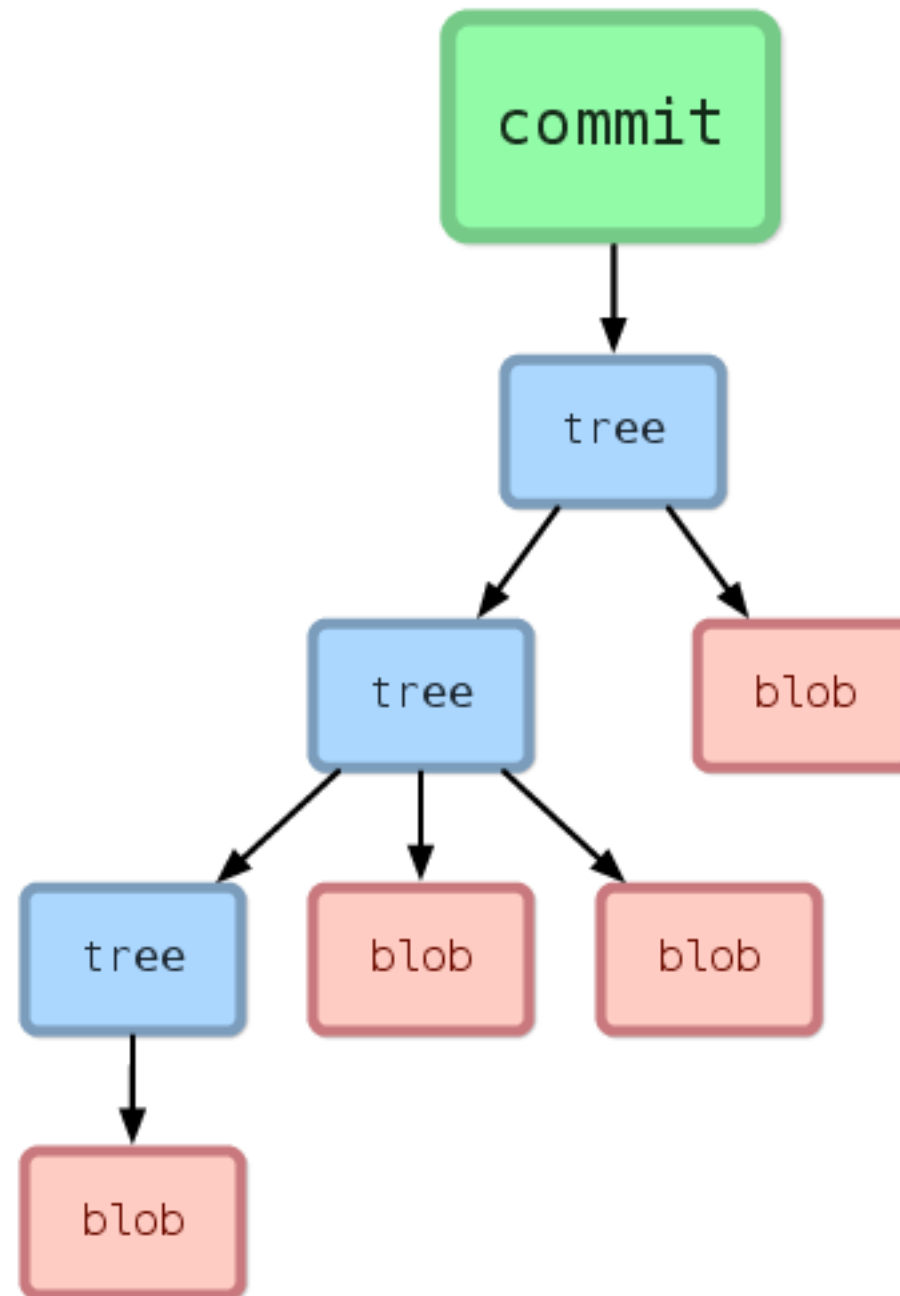
blob

tree

commit

tag

# Object Database



# Object Database

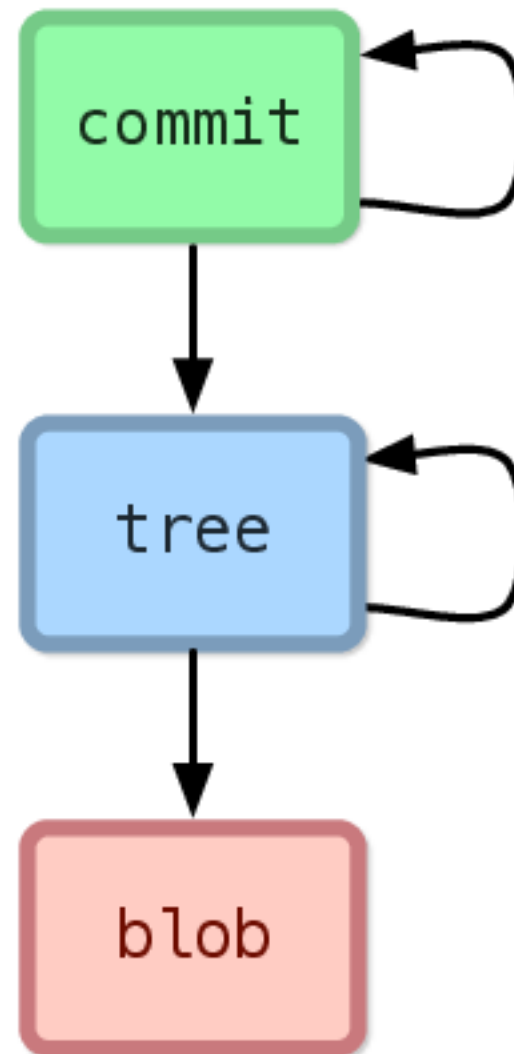
commit

```
commit 155\0
```

```
tree 9a3ee
parent fb39e
author Patrick Hogan <pbhogan@gmail.com> 1311810904
committer Patrick Hogan <pbhogan@gmail.com> 1311810904
```

```
Fixed a typo in README.
```

# Object Database



# Object Database

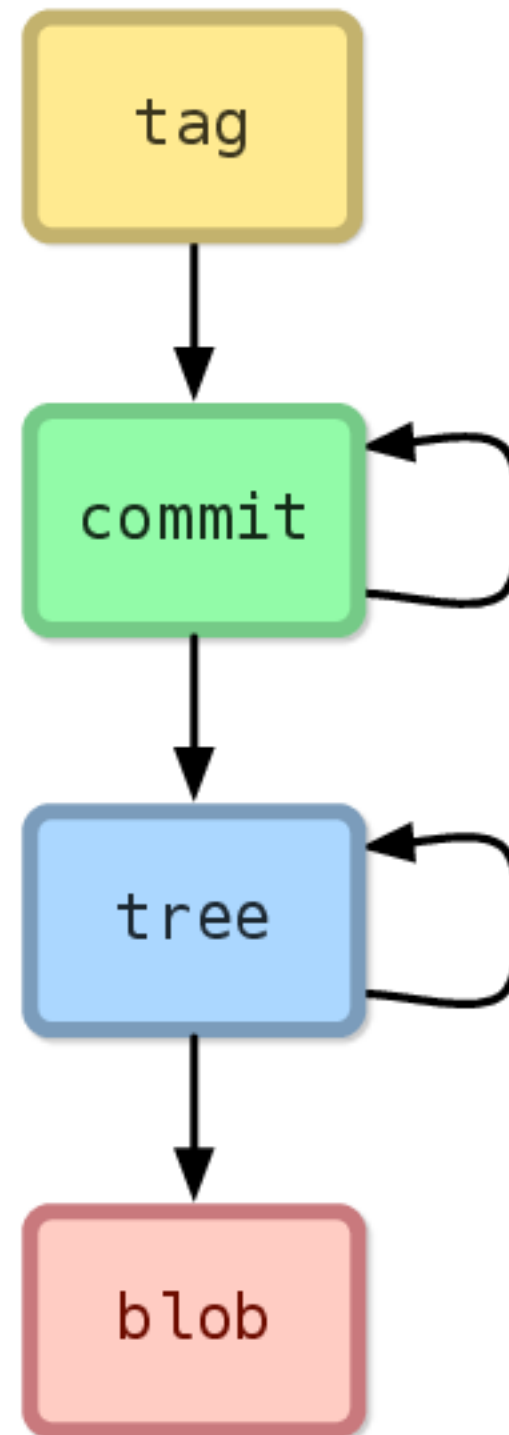
blob

tree

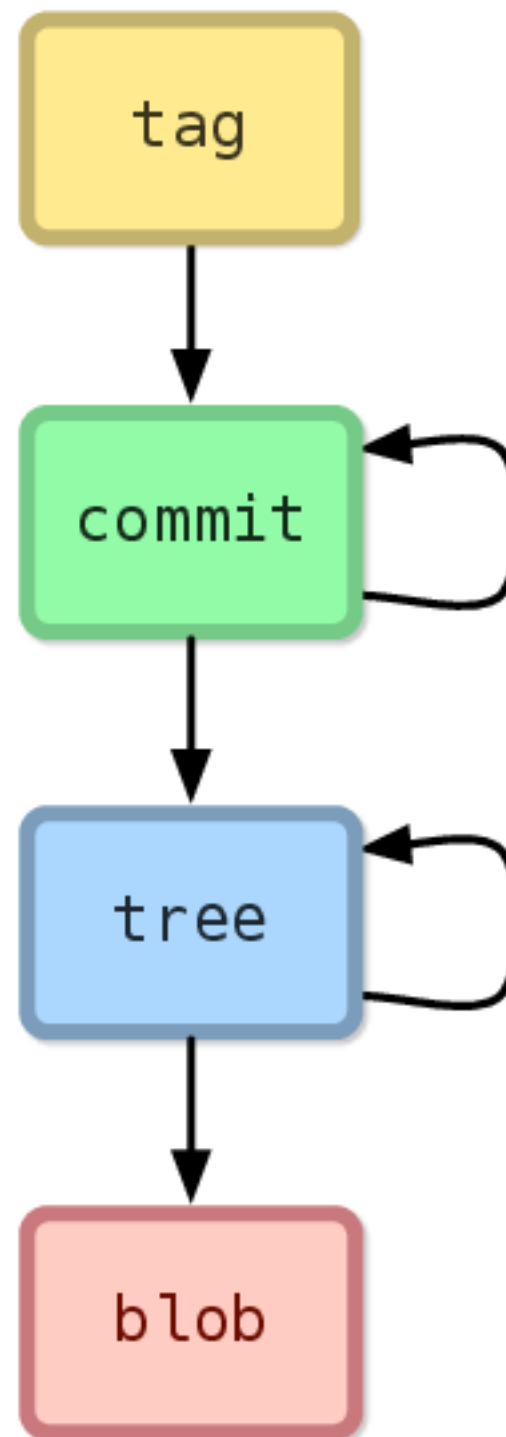
commit

tag

# Object Database



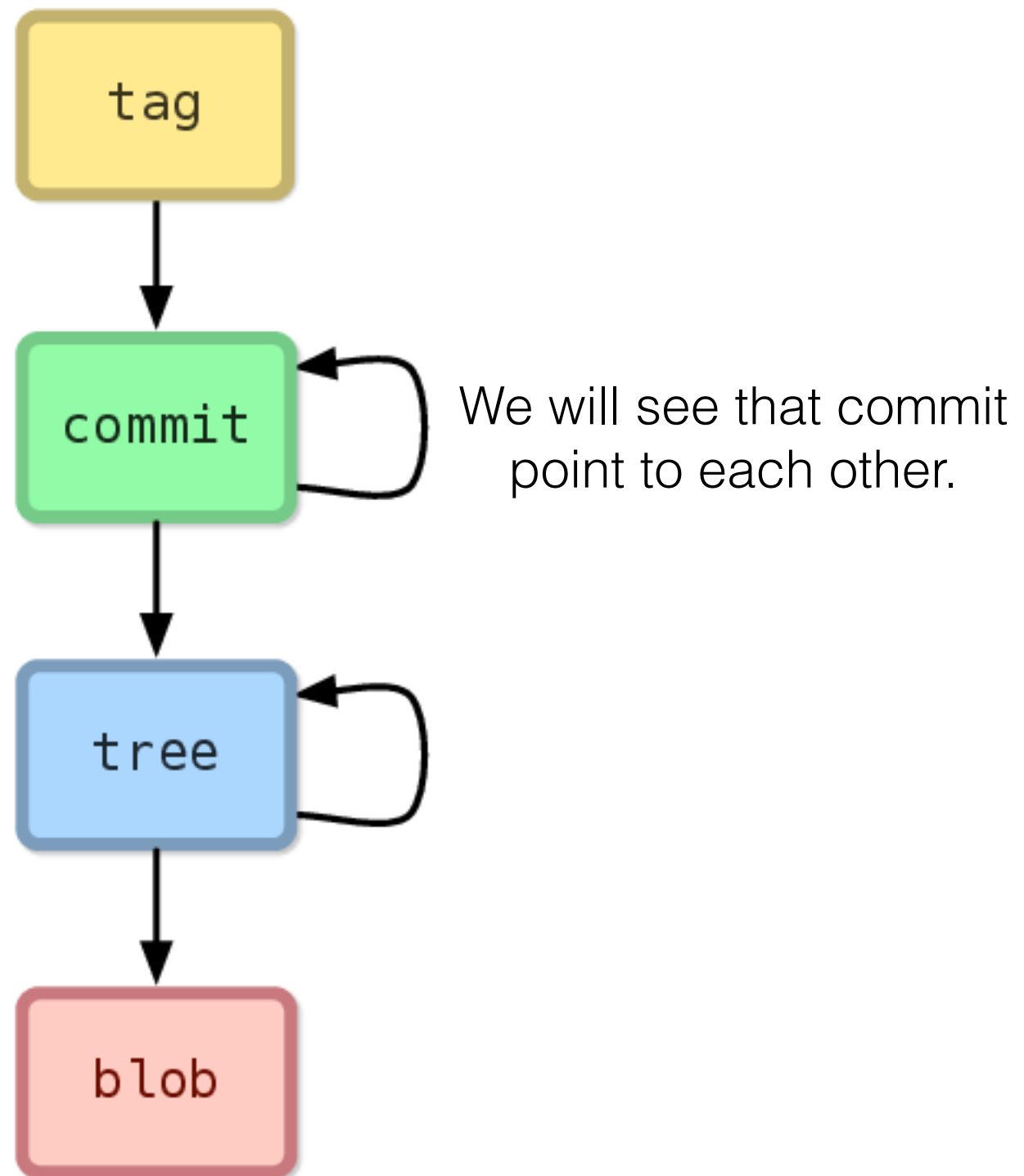
# Object Database



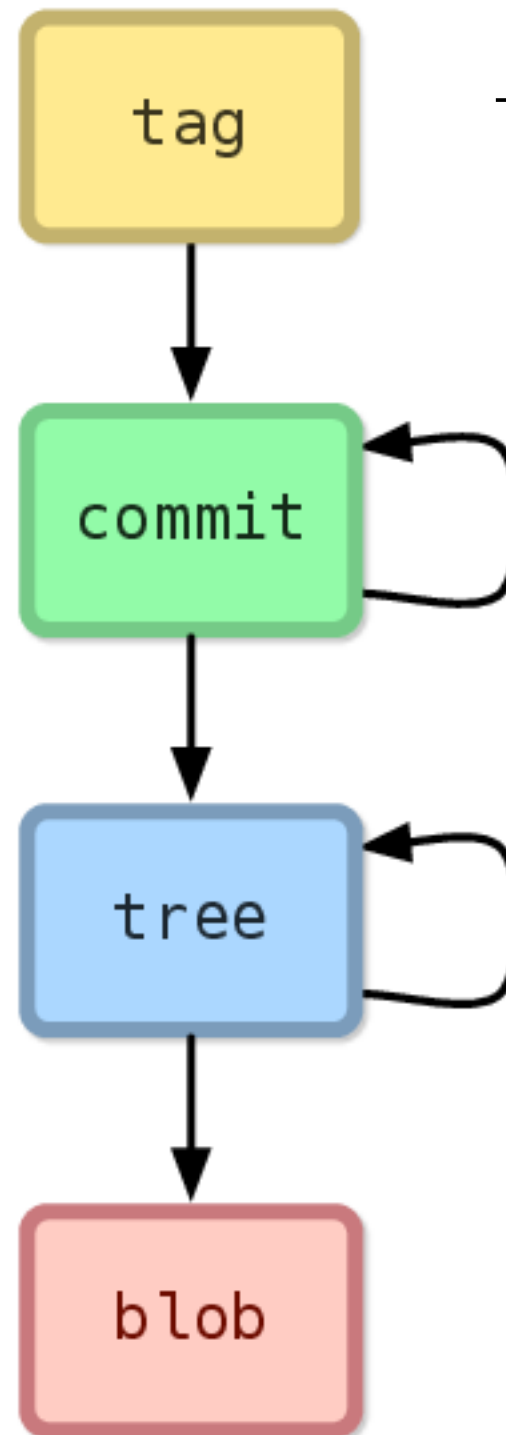
We saw that tree point to each other.



# Object Database



# Object Database



Tag is a pointer to a commit.

# Object Database

tag

```
tag 121\0
```

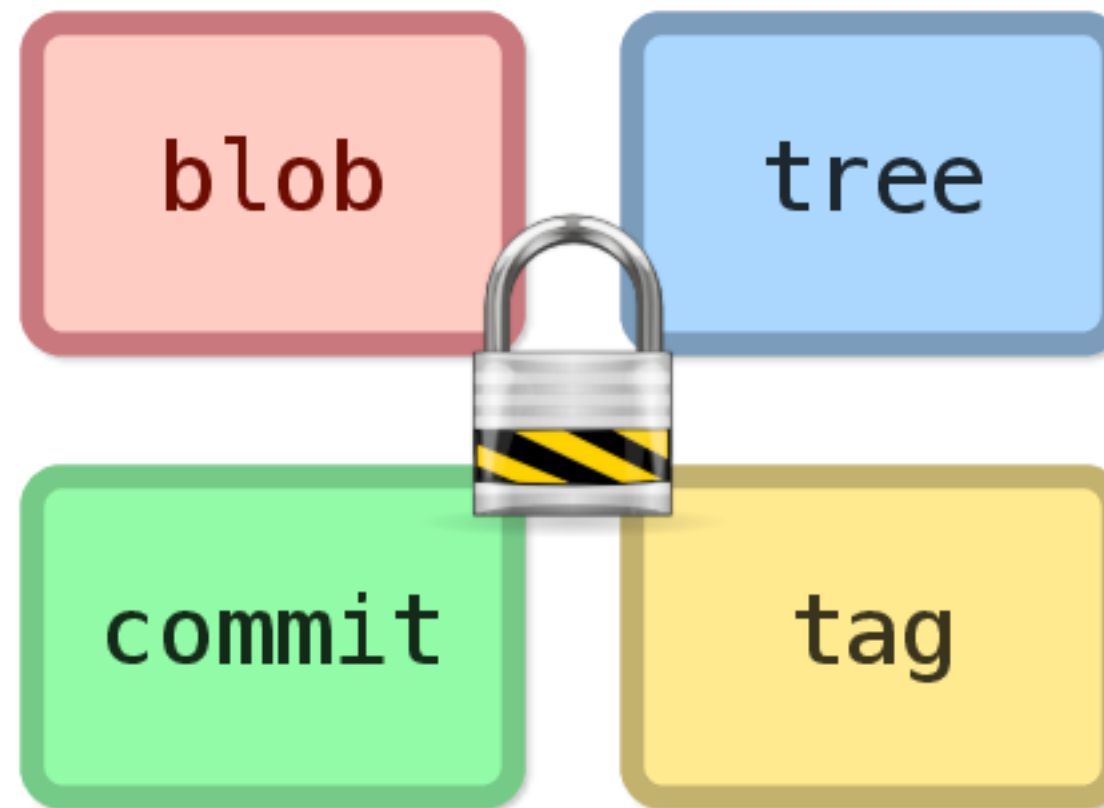
```
object e4d23e
type commit
tag v1.2.0
tagger Patrick Hogan <pbhogan@gmail.com> 1311810904

Version 1.2 release -- FINALLY!
```

**.git/objects/20/c71174453dc760692cd1461275bf0cf feb772f**

**.git/refs/tags/v1.2.0**

# Object Database



**Immutable!**

# **Never Removes Data**

## **(Almost)**

# **"Rewriting History"** **Writes Alternate History**

# **Git Directory**

**Configuration File**

**Hooks**

**Object Database**

**References**

**Index**

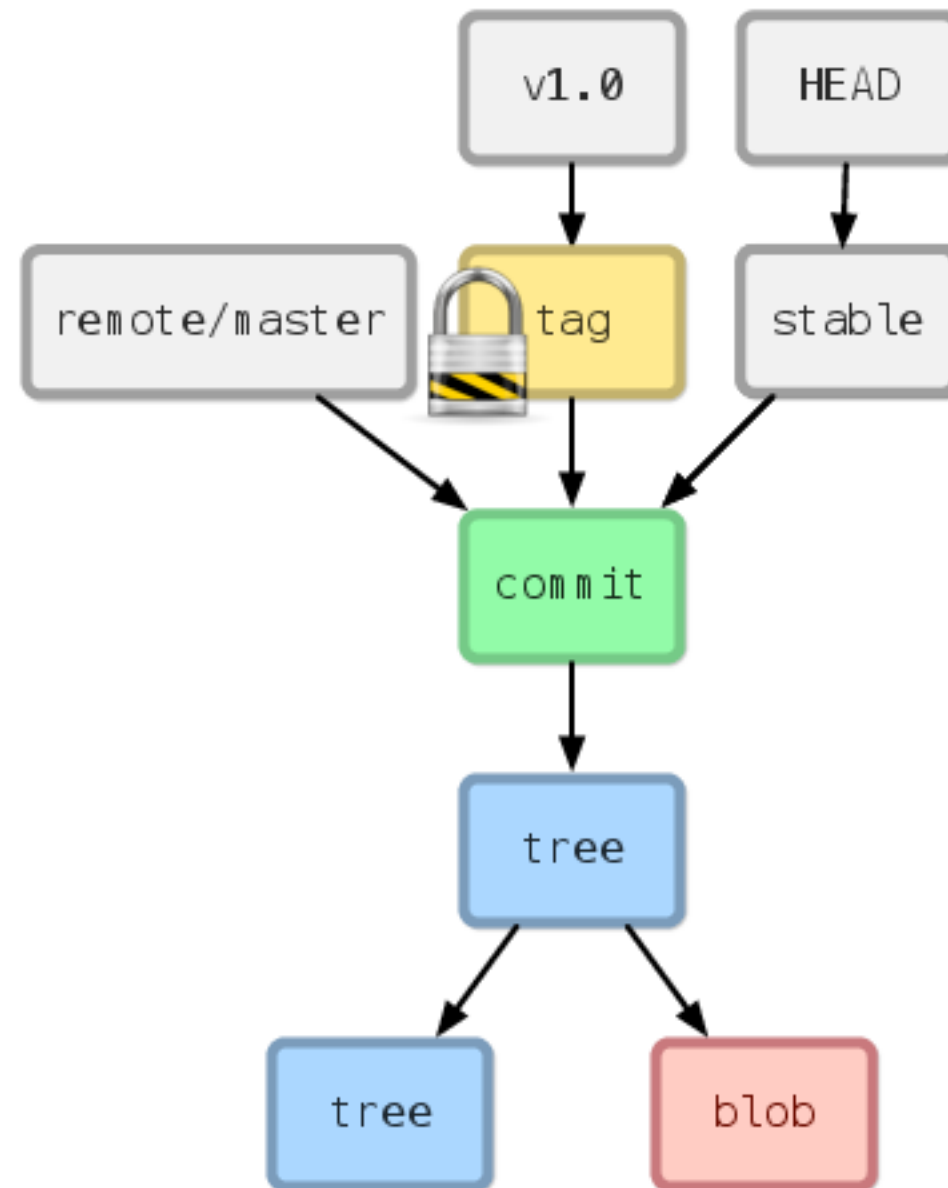
# References



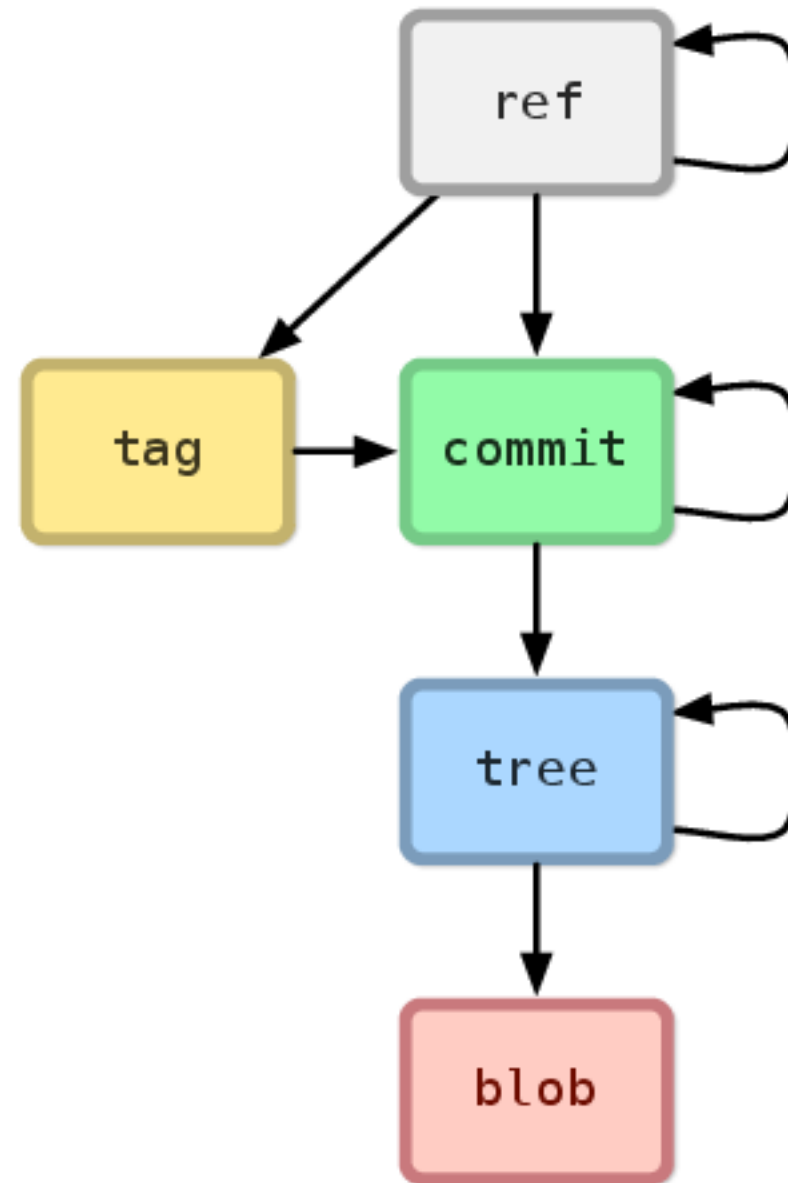
# References

**Lightweight, Movable  
Pointers to Commits**  
(and other things)

# References

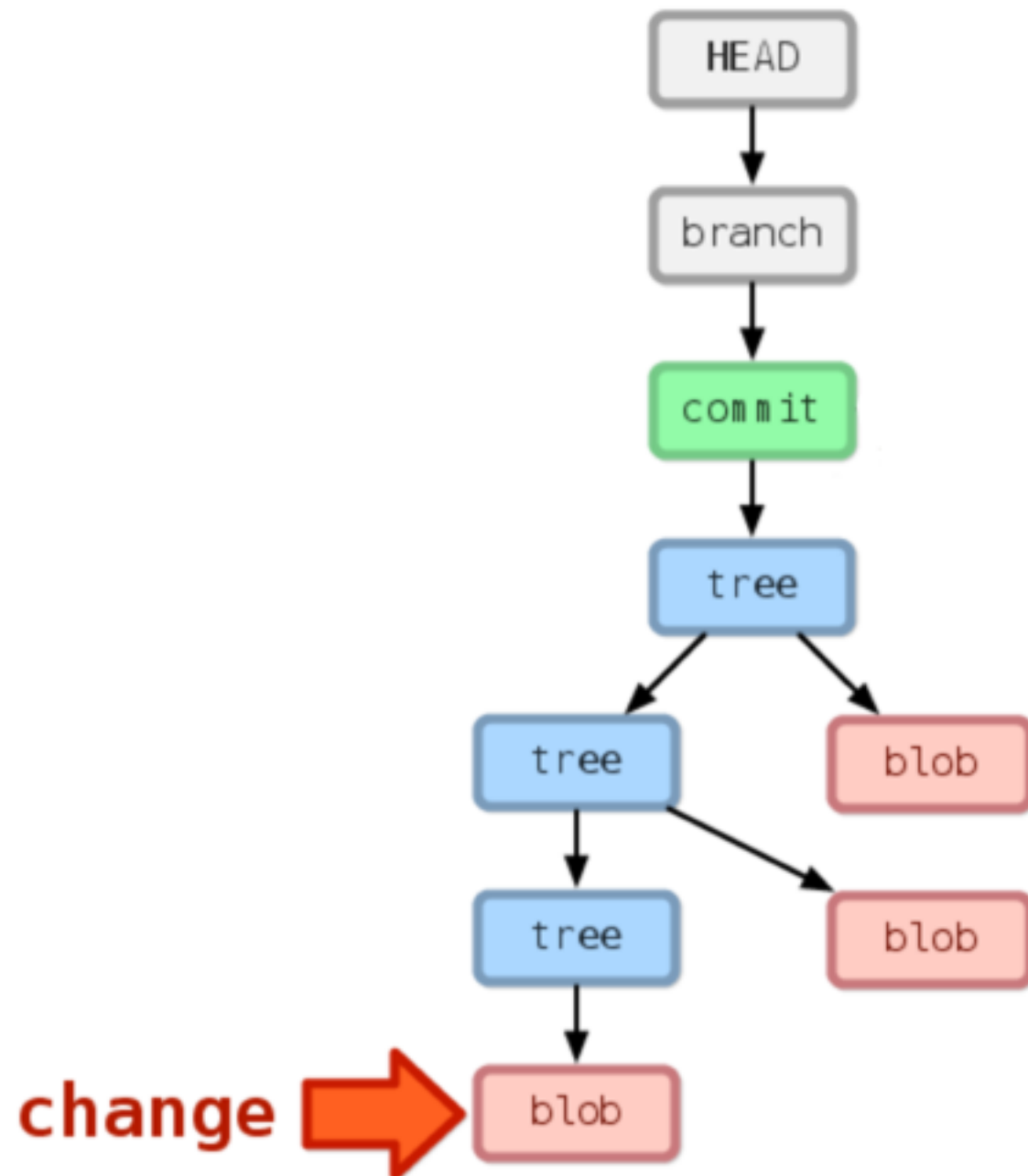


# References

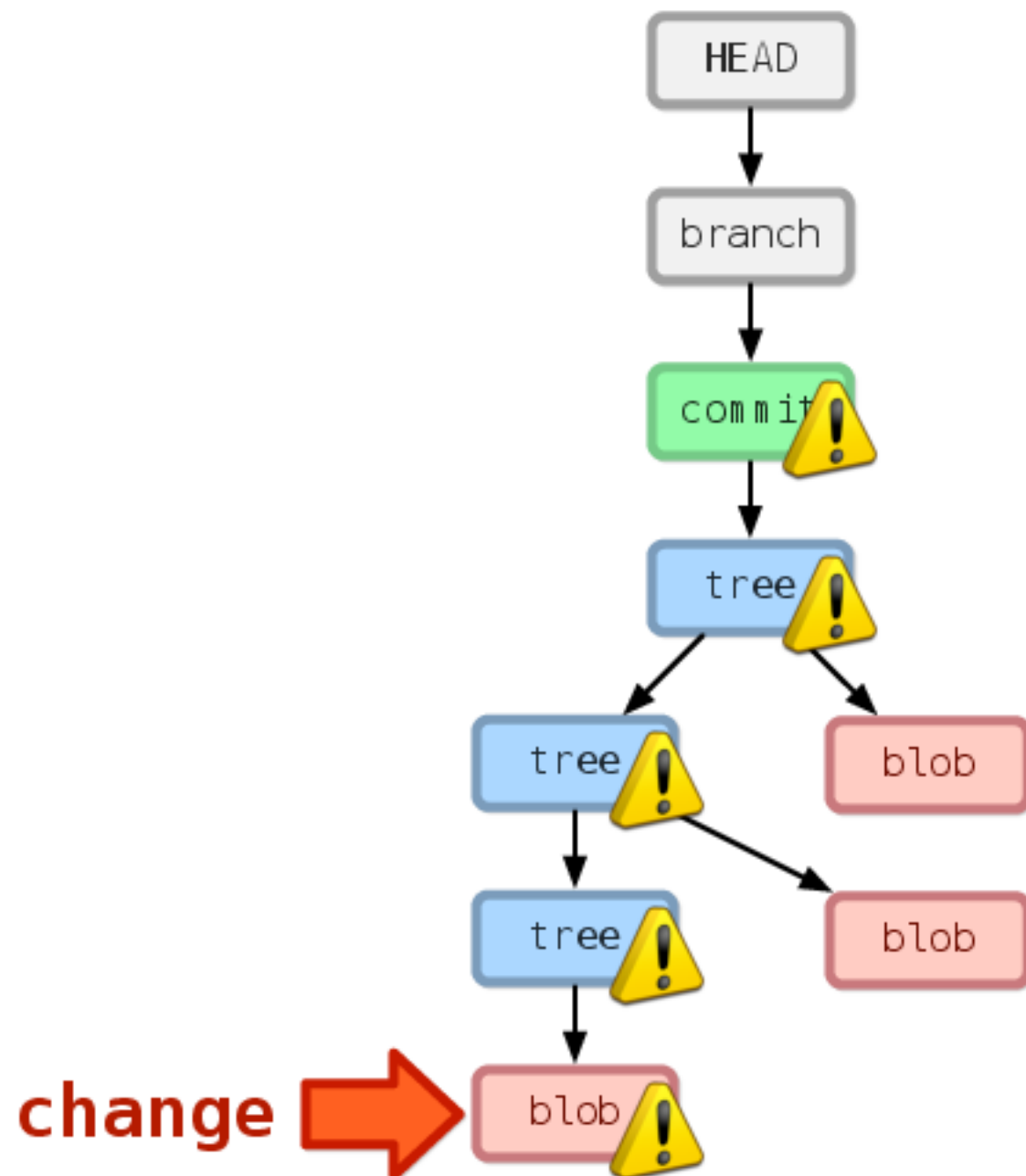


# Scenario

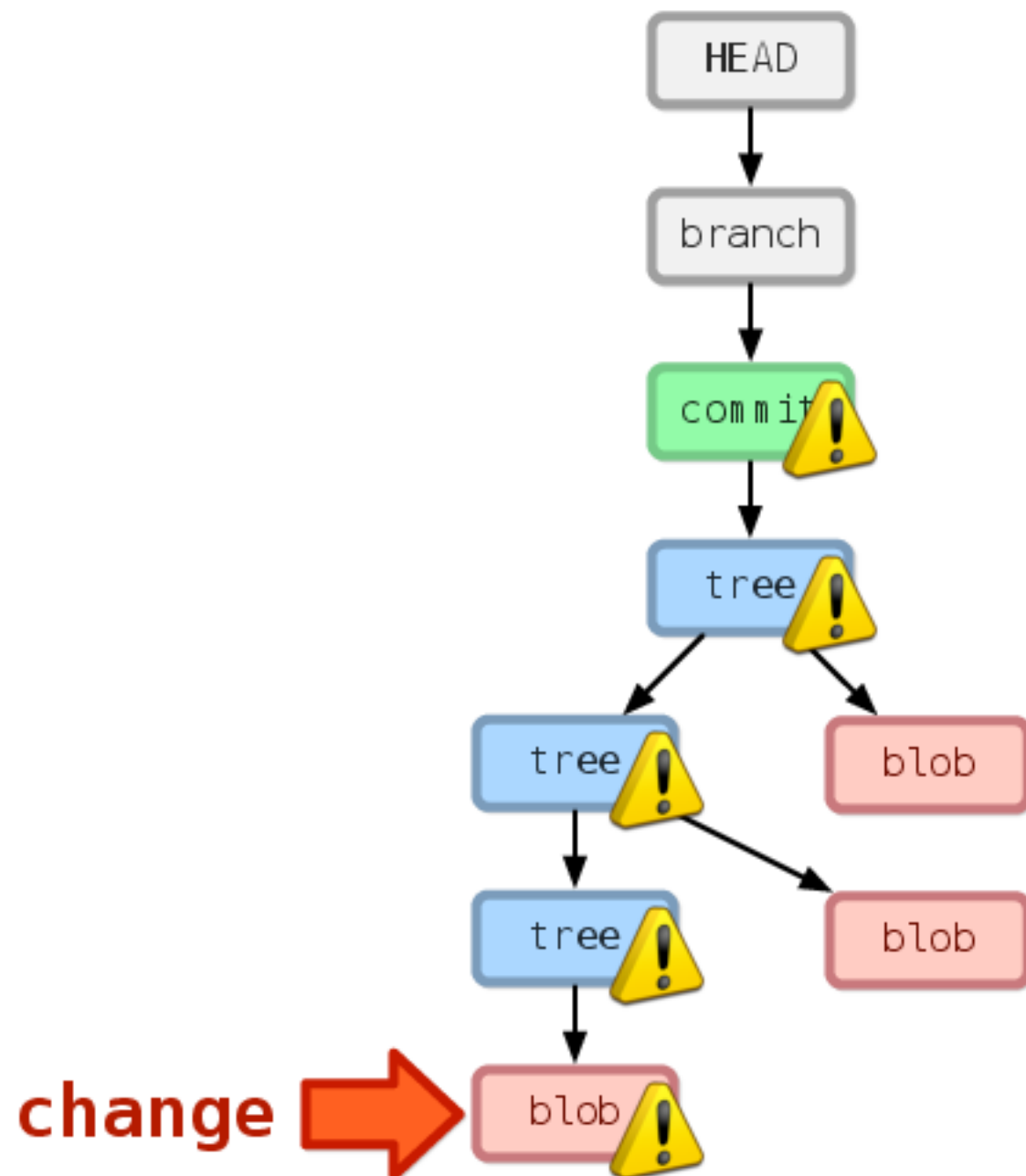
# Scenario



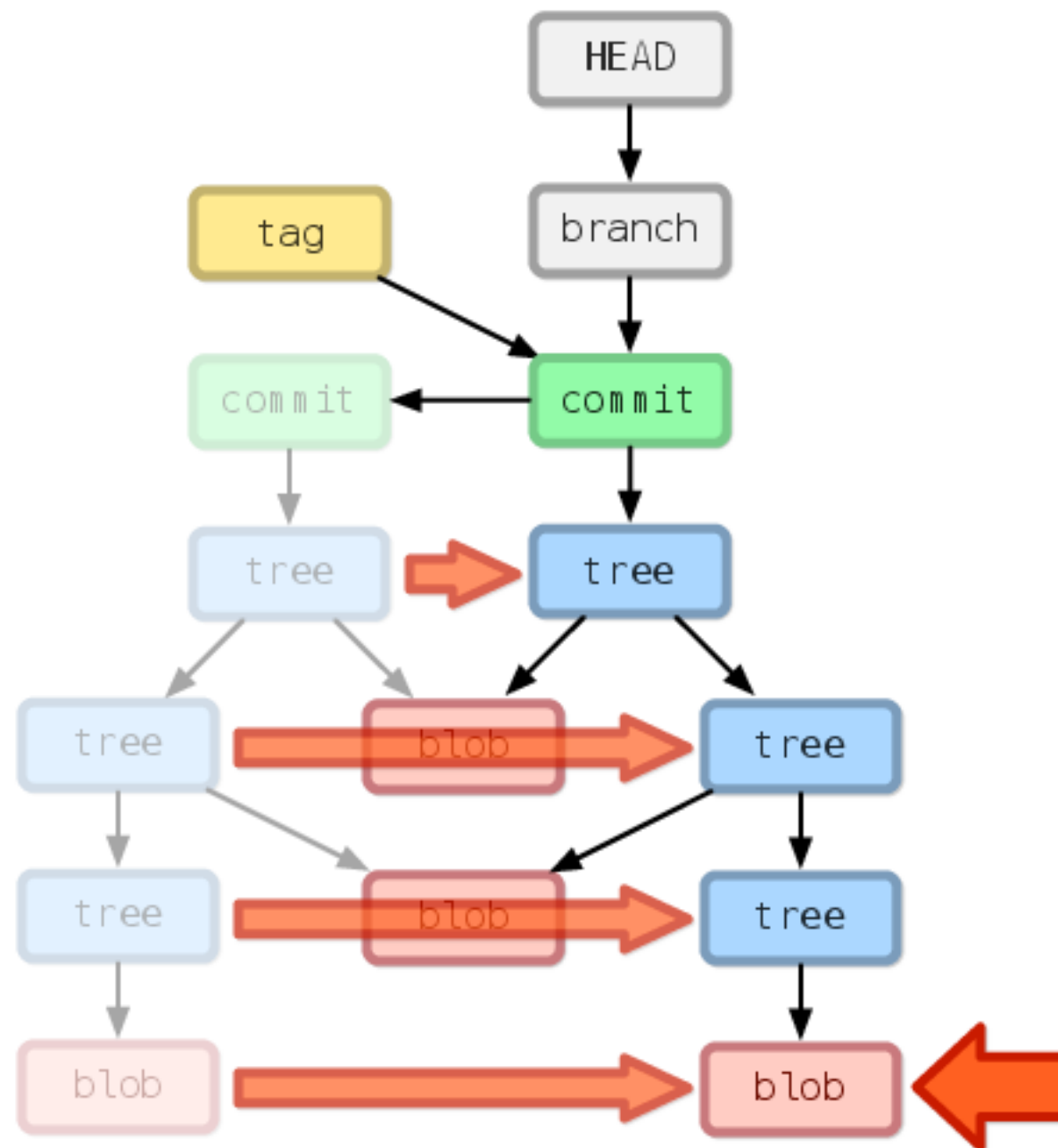
# Scenario



# Scenario



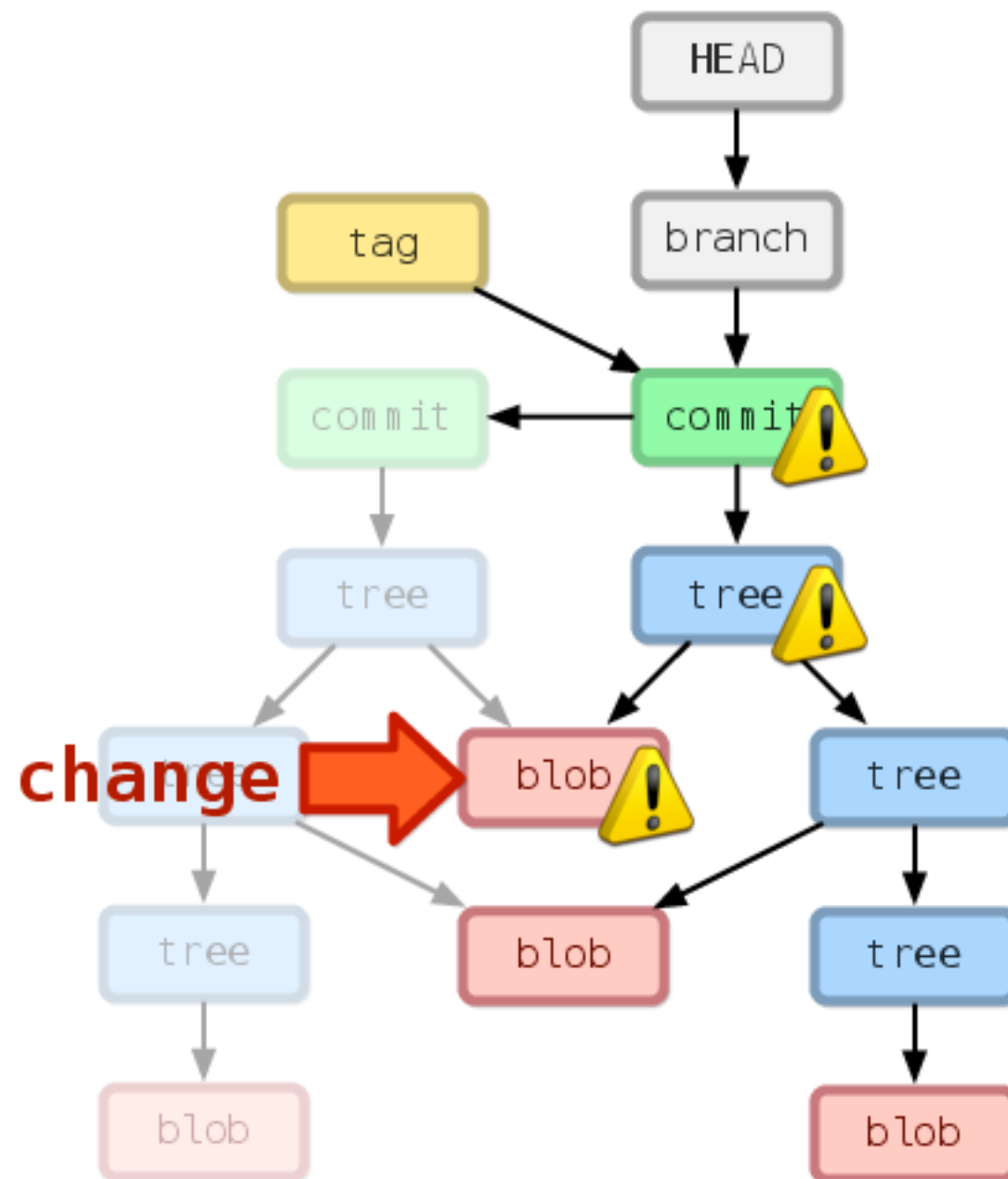
# Scenario



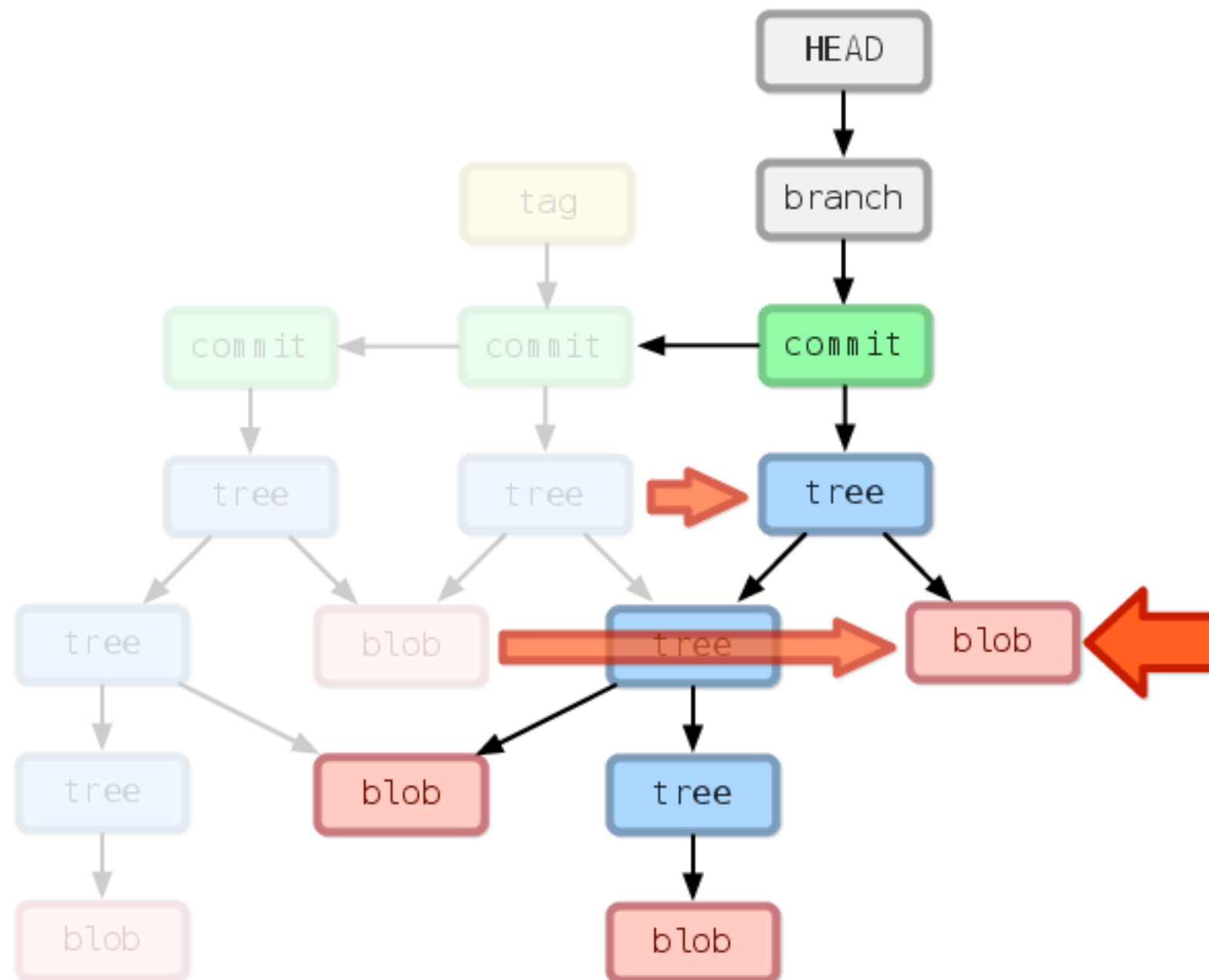
**new objects**



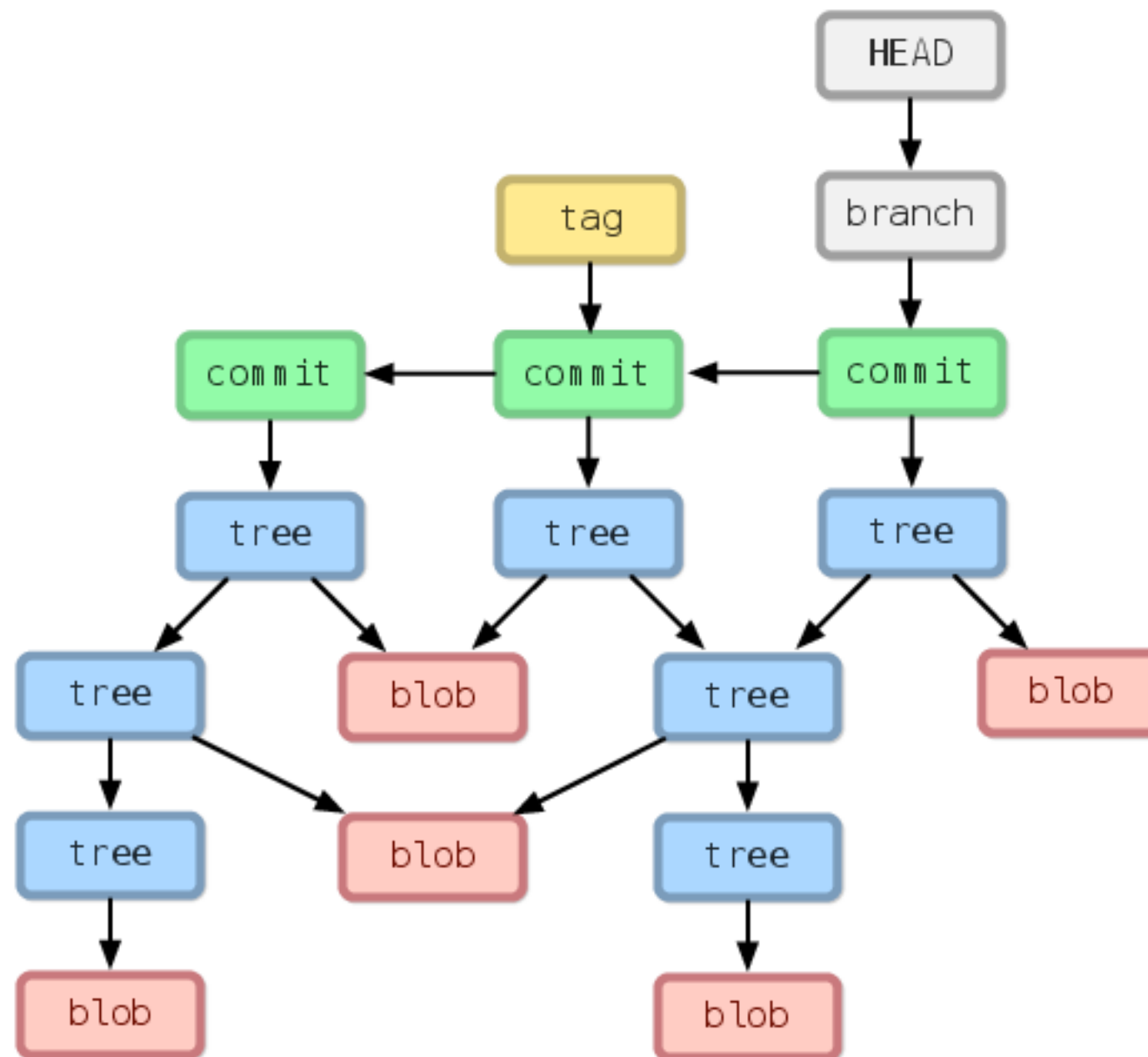
# Scenario



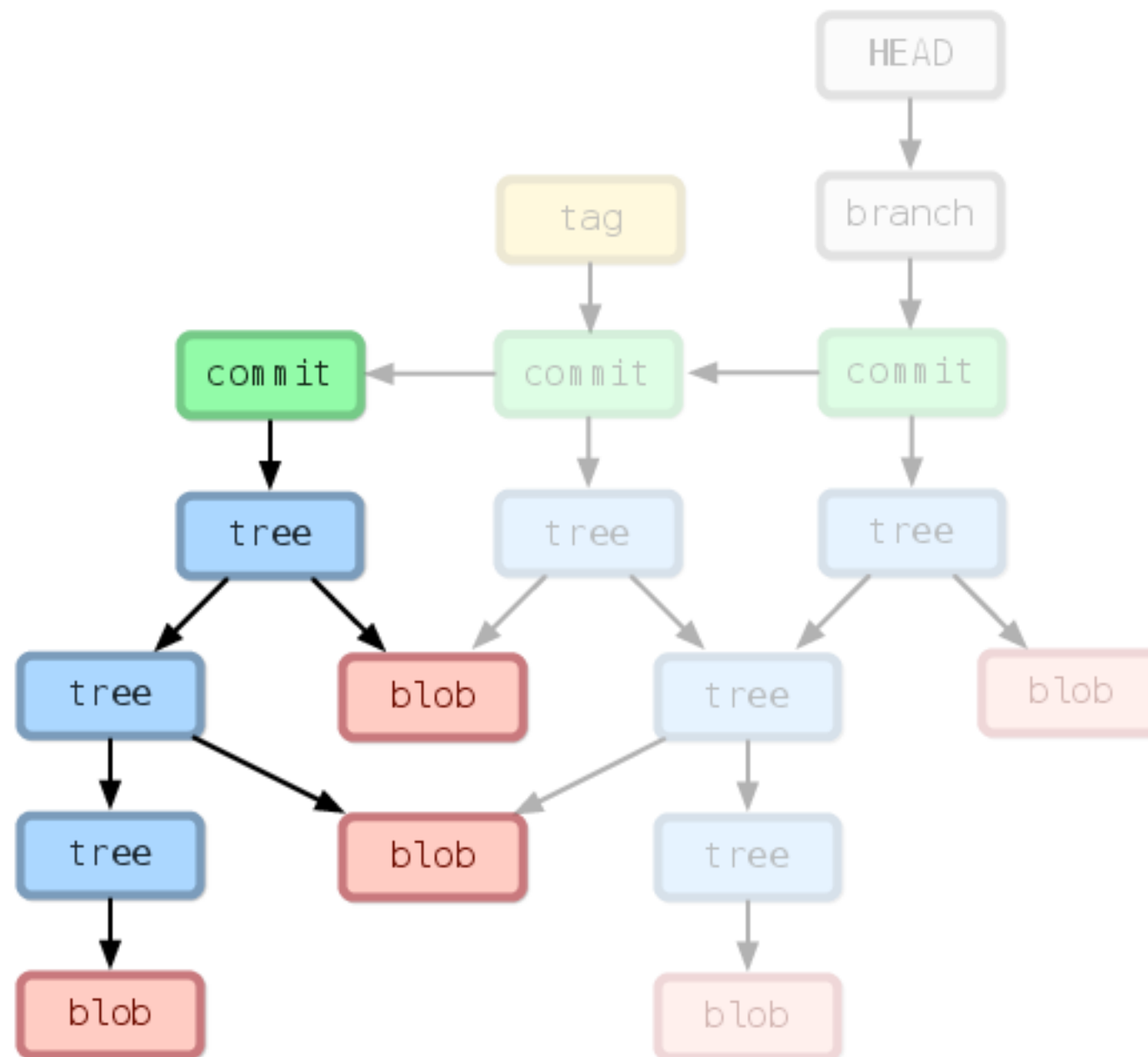
# Scenario



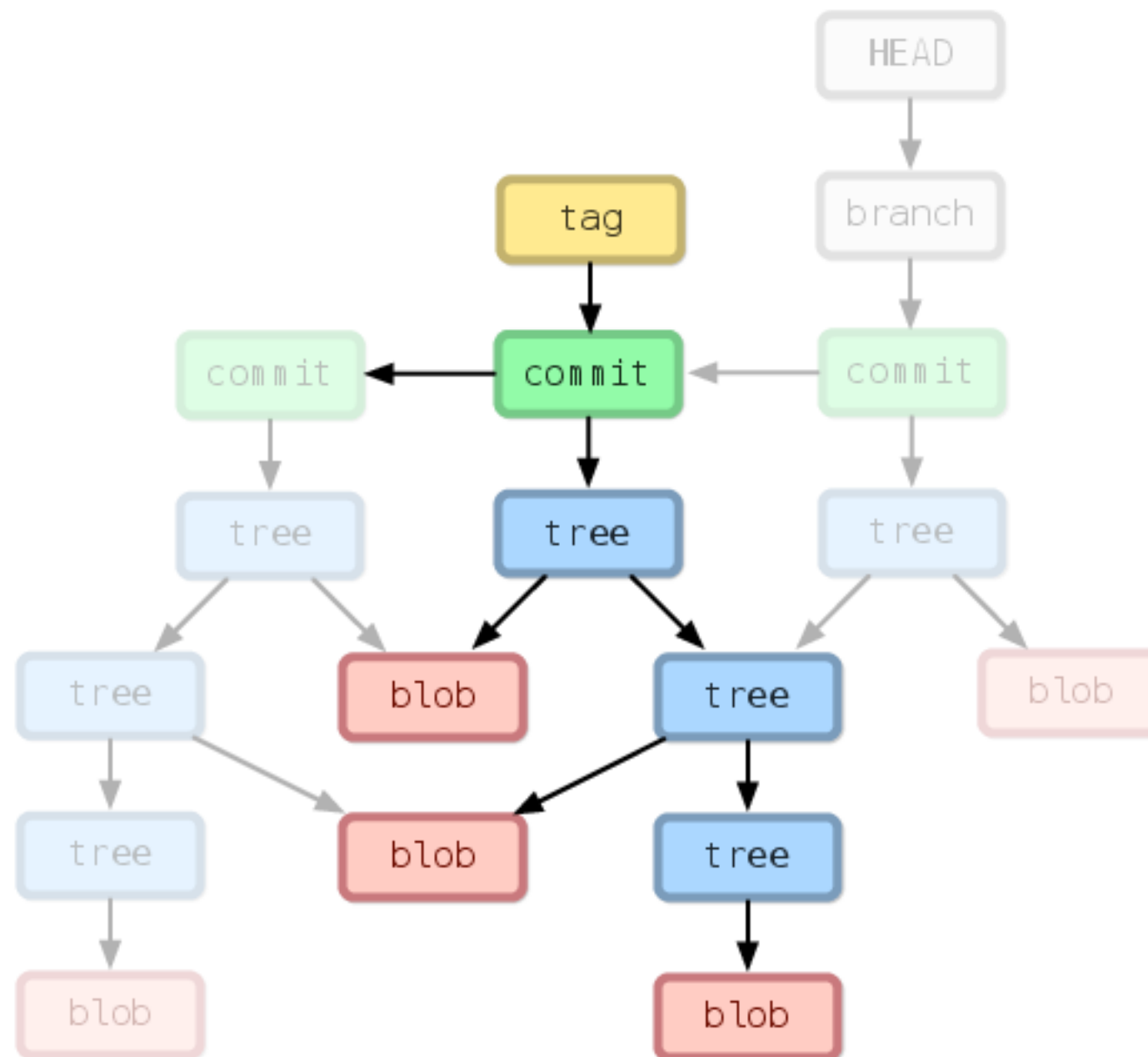
# Scenario



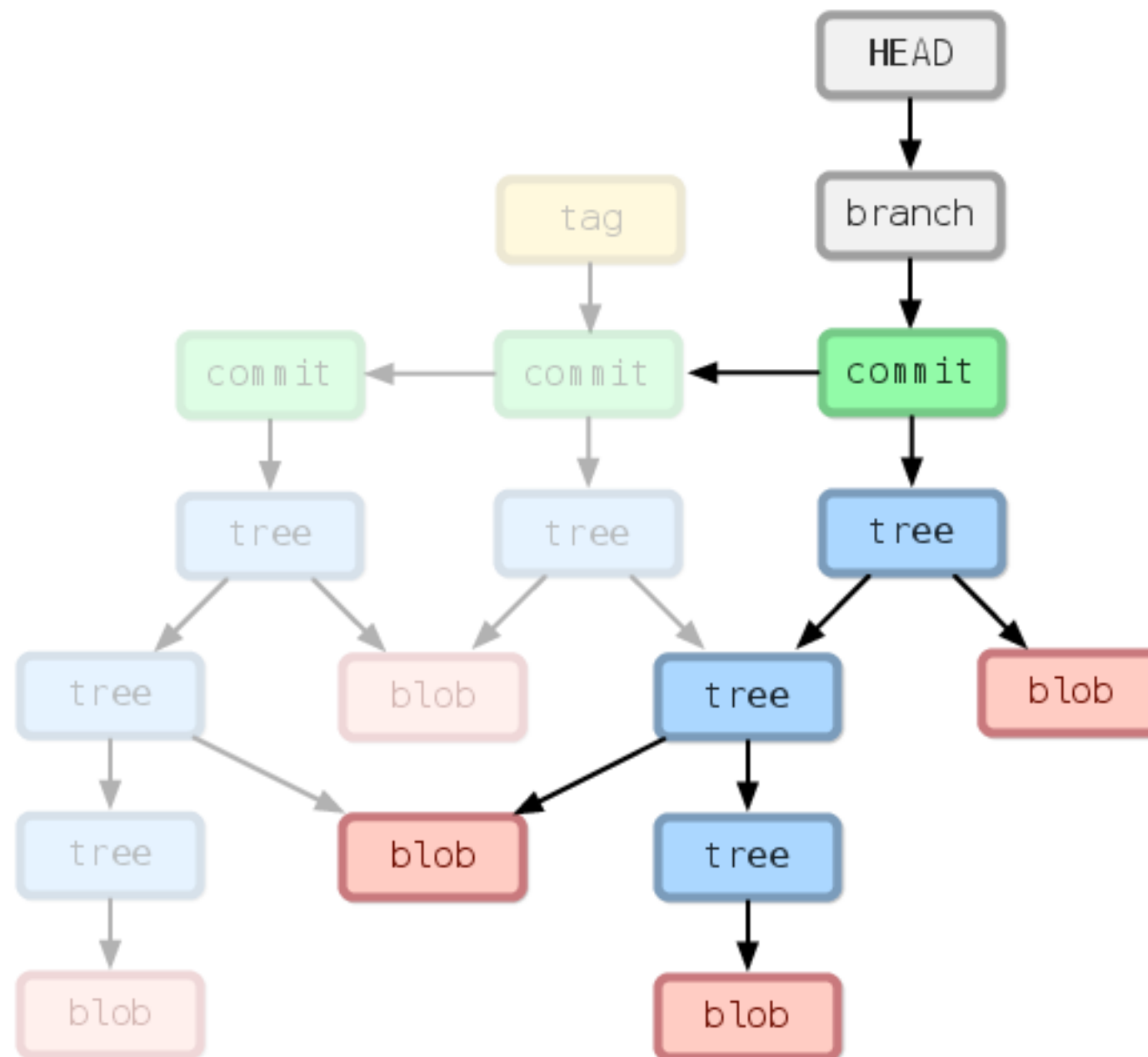
# Scenario



# Scenario



# Scenario



# **Git Directory**

**Configuration File**

**Hooks**

**Object Database**

**References**

**Index**

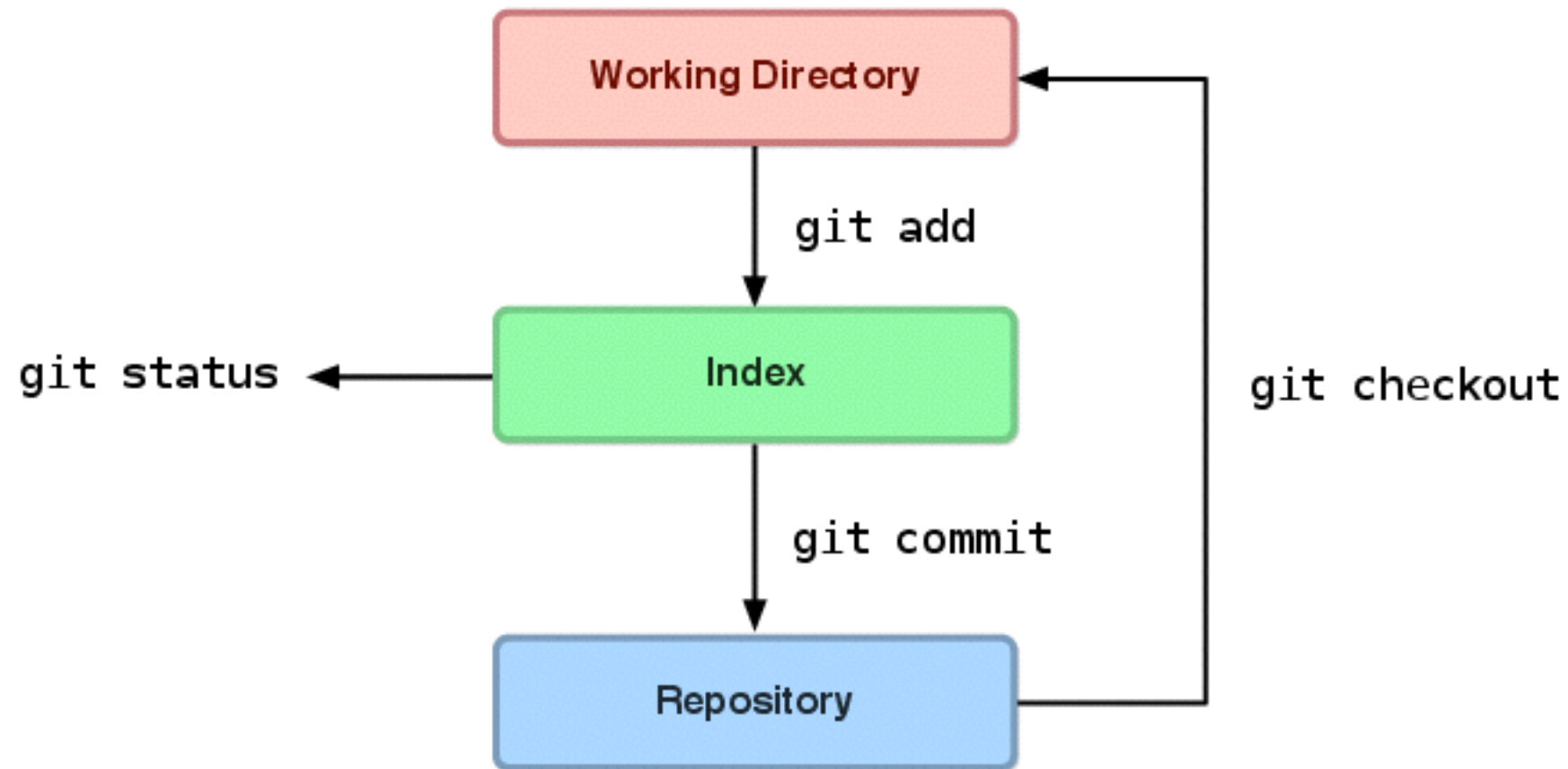
# Index



# Index

**== Staging Area**

# Index



# **Index FTW**

**No Need To Commit All At Once**

**Pick (Stage) Logical Units to Commit**

**Helps You Review Your Changes**

**Lets You Write Your History Cleanly**