Azrieli College of Engineering Jerusalem
עזריאלי מכללה אקדמית להנדסה ירושלים

הנדסת תוכנה

6-7. בקרת תצורה + git / github

Pragmatic Programmer Tip :
Always Use Source Code Control
Source code control is a time machine for your work—you can go back.

השבוע

- בקרת תצורה \ קוד – Revision Control
- כלים: git / github
  – משימה אישית 3
- שיטות: מודל ענפים
- סקר ZFR
- פרויקט 6: תחילת סבבים – סבב MVP 1
- תרגיל – סקר ZFR
  (+ השלמת back-end)

בקרת גרסאות קוד - מקורות

- Sink, Version Control by Example
- Google Tech Talk: Linus Torvalds on git
- Spolsky, Hg Init: a Mercurial tutorial http://hginit.com
- Sink, Source Control HOWTO http://www.ericsink.com/scm/source_control.html
- Intro to Distributed Version Control http://betterexplained.com/articles/intro-to-distributed-version-control-illustrated/

VCS Links

- Eric Raymond on vcs
- The 10 commandments of good source control management, blog 2011
- FogBugz and Kiln video, 2011
  - DVCS University Slides (*)
- Spolsky, Hg Init: a Mercurial tutorial
- Azad, A Visual Guide to Version Control
  - Intro to Distributed Version Control (Illustrated)
- Eric Sink, Source Control HOWTO

Git Links (see wiki links)

- **http://git-scm.com/** (getting started)
- Set Up Git (Win), ssh issues http://help.github.com/win-set-up-git/
- Info: http://git-scm.com, gitref.org, progit.org (book, basic), Git In The Trenches
- Windows client list, O'Reilly Webcast: Git in One Hour
- git internals video

More Git / Github Links

- Git branching with git-flow, Heb. Video, 2013
- Videos: Git For Ages 4 And Up, Git Going(oredev'12), Think like a Git
- learn.github.com/, try.github.com/Gitflow
- saastv: Using Branches with Git
- no branches at flickr
- Insider Guide to GitHub (video)
- Articles: article (including .gitignore), post, difficulties,

איפה אנחנו בפרויקט (בקורס)?

- למה?
  בעיה (פלט: הצעת פרויקט\חזון\SOW)
- מי?
  צוות (Inception, אתחול\תכנון פרויקט)
- מה?
  דרישות (SRS)
- איך?
  תיכון (ארכיטקטורה) (SDS)
- מתי?
  תכנון וניהול – (ZFR)
- בניה
  (סבבי פיתוח)

בקרת תצורה (SCM)

- לפרויקט תוכנה תוצרים שונים:
- מסמכי דרישות ותיכון, קוד, executables, מדריכי שימוש, בדיקות,...
- פרויקט תוכנה משתמש בכלים שונים:
- מהדרים, עורכים, צד ג', שת"פ, (מ"ה), ...

בקרת תצורה

- מבחינת תהליך זה אלו נקראים CI – Configuration Items
- לכל אחד יכולים להיות גרסאות ועותקים שונים
- אנו צריכים יכולת לזהות, לעקוב ולאחסן אותם
- נתמקד בנושא של גרסאות

בקרת גרסאות – Version Control

- איך (האם?) אתם שומרים את תוצרי העבודה שלכם?
- האם אפשר לשפר?
- האם יש הבדל בין מפתח בודד לחברה גדולה?
- שמות שונים:
  - בקרת תצורה
  - Revision Control
  - Software Configuration Management
  - Source-Code/Version Control System

Joel Test (~2000 / stackoverflow)

1. Do you use source control?

וגם היום...

"You've just spent twenty minutes doing a presentation for your teammates on adopting source control. Yeah, they don't do source control at all. Yep, not at all—it's as if the last 20 years of computer science never happened. But better late than never, and frankly any source control is better than none, because disaster is one errant delete away."

- "Driving Technical Change: Why People on Your Team Don't Act on Good Ideas, and How to Convince Them They Should", chap. The Cynic

בקרת גרסאות – בשביל מה? יעדים

- איסוף כל הגרסאות ומעקב אחרי שינויים
  - חזרה לגרסה מסוימת, השוואה
  - מאפשר מחיקת קוד
- ניהול מספר גרסאות במקביל
- גיבוי והצלה
- שיתוף מספר מפתחים (מרוחקים) בו זמנית
  - טיפול בסתירות
- מאגר מעודכן של תוצרי הפרויקט
  - במיוחד עם Continuous Integration\daily build

בפרויקט תדרשו להדגים את בקרת התצורה שלכם

פעולות נדרשות

- בקרת שינויים
  - זיהוי ותיעוד (למשל מי משנה, הסיבה, זמן וכדו')
  - ניתוח והערכה (של שינוי)
  - אישור \ דחיה
  - אימות, מימש ושחרור
- בקרת גרסאות
  - מאגר
  - הכנסה והוצאה
  - ענפים ומיזוגים
  - תיוג

כלים: היסטוריה (השוואה)

Generation	Networking	Operations	Concurrency	Examples
1	None	One file at a time	Locks	RCS, SCCS
2	Centralized	Multi-file	Merge before commit	CVS, SourceSafe, Subversion, Team Foundation Server, IBM Rational ClearCase
3	Distributed	Changesets	Commit before merge	Bazaar, Git, Mercurial

40 Years of Version Control

SCCS & RCS (1970s)

CVS (1986)

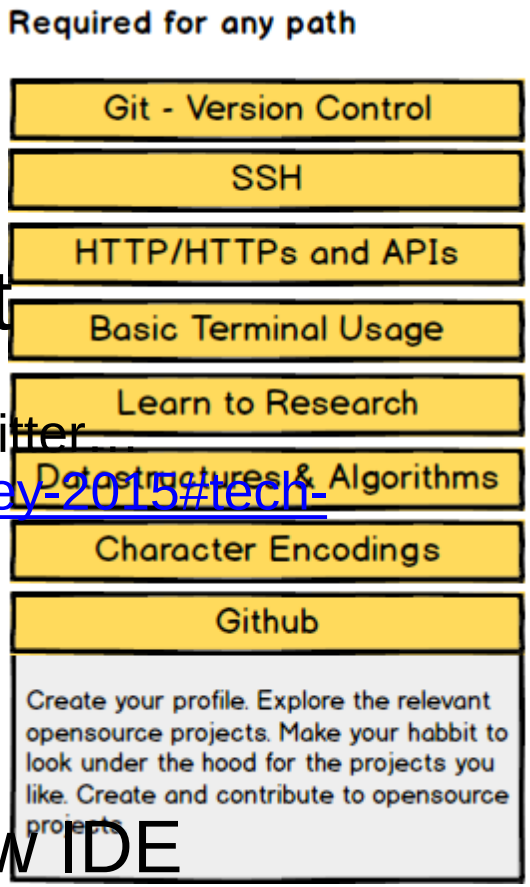
Subversion (2001)

Git (2005)

Image © TheSun.au

Git

- Successful open source project
  https://git.wiki.kernel.org/index.php/GitProjects
  https://github.com/google, microsoft, facebook, twitter…
  http://stackoverflow.com/research/developer-survey-2015#tech-sourcecontrol
- Problems / Issues:
  - Usability!
  - Mainly a scripted / toolset (by now IDE integration and GUIs)
  - Binary/big file (by now, e.g., git-lfs)
  - Enterprise (e.g. locking)



משל גיט

- Tom Preston-Werner http://tom.preston-werner.com/2009/05/19/the-git-parable.html
- Herland, http://www.infoq.com/presentations/git-details, slides: https://github.com/jherland/git_parable

The Git Parable

Johan Herland

johan@herland.net

The Git Parable

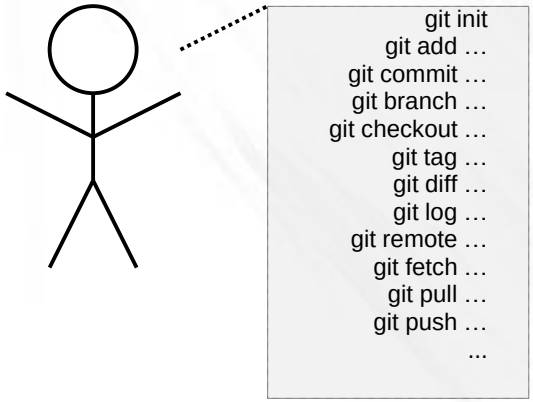
- Shamelessly stolen from Tom Preston-Werner
  http://tom.preston-werner.com/2009/05/19/the-git-parable.html
- I'm lazy...
- Also: Best introduction to Git I've found so far

The Git Parable

- Git - simple & powerful

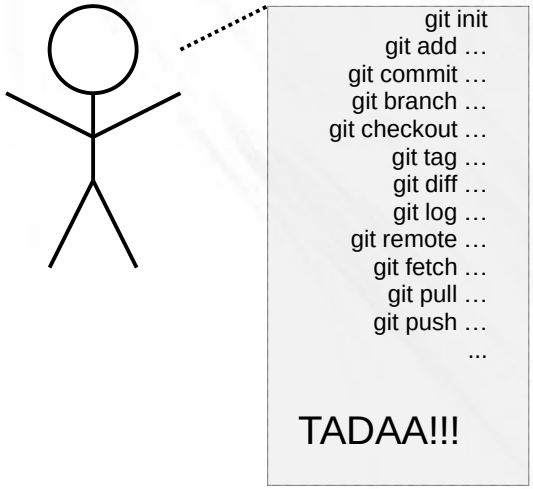
The Git Parable

- Git - simple & powerful



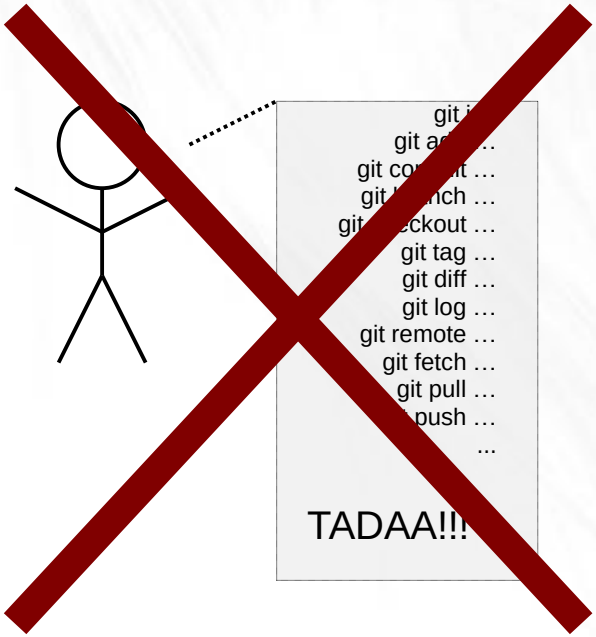
The Git Parable

- Git - simple & powerful



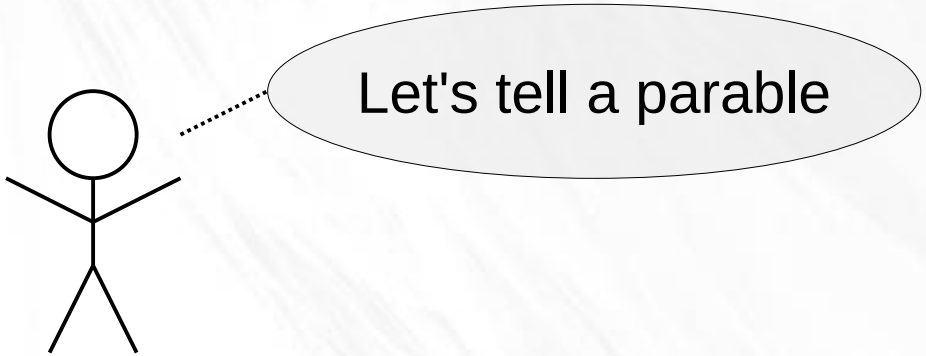
The Git Parable

- Git - simple & powerful



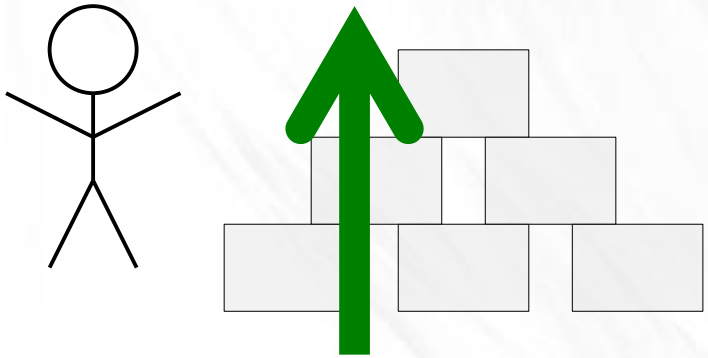
The Git Parable

- Git - simple & powerful



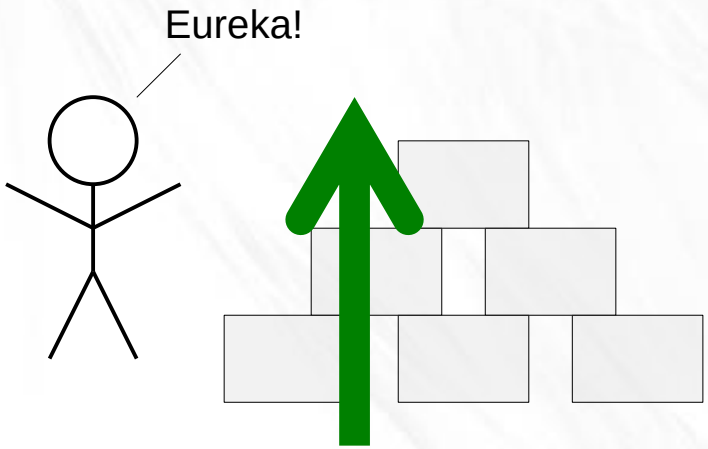
The Git Parable

- Git - simple & powerful



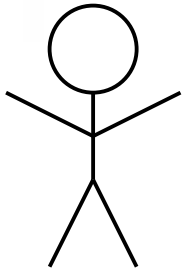
The Git Parable

- Git - simple & powerful



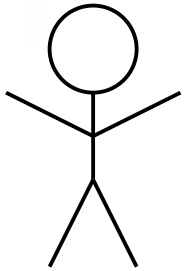
The Parable

- A simple computer
  - A text editor
  - A few filesystem commands



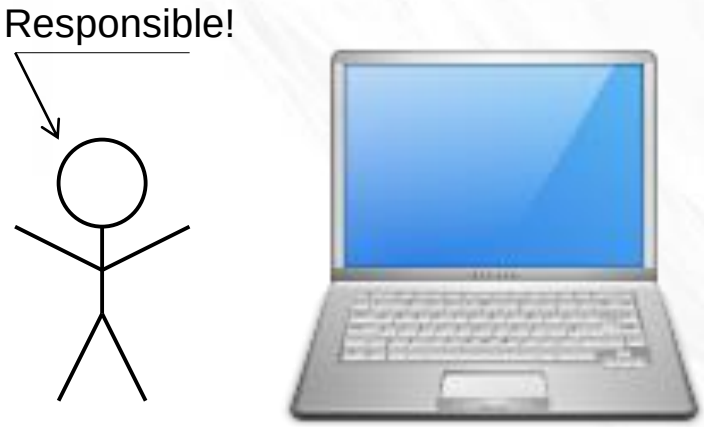
The Parable

- Write a large software program



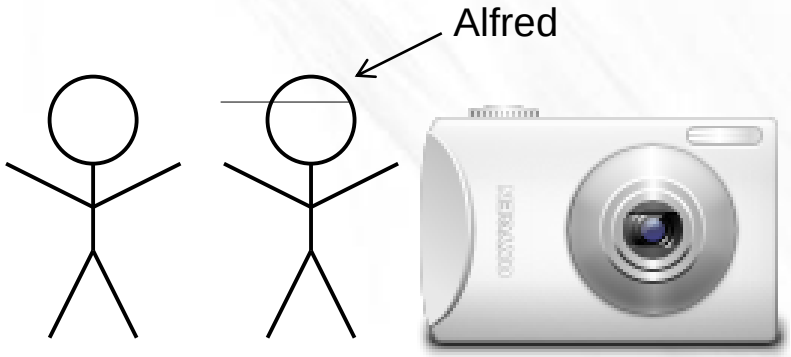
The Parable

- Write a large software program
- Invent some method to keep track of versions
  - retrieve code that you changed/deleted



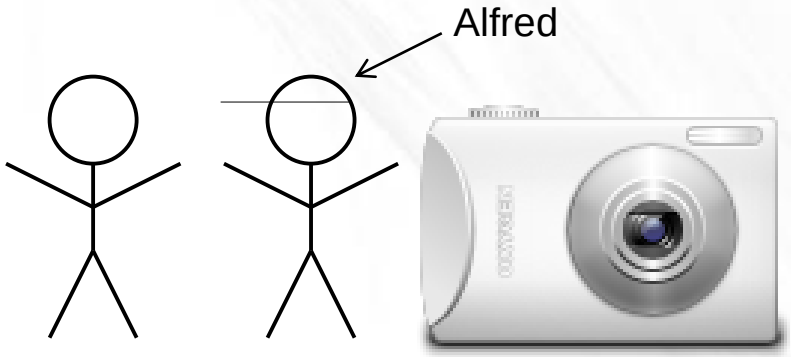
Snapshots

- Alfred, the photographer



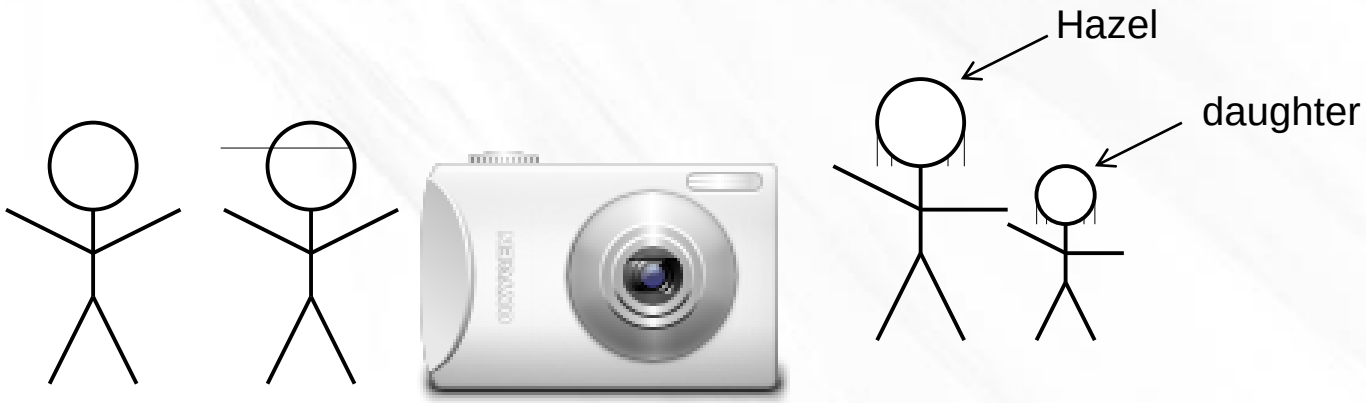
Snapshots

- Alfred, the photographer



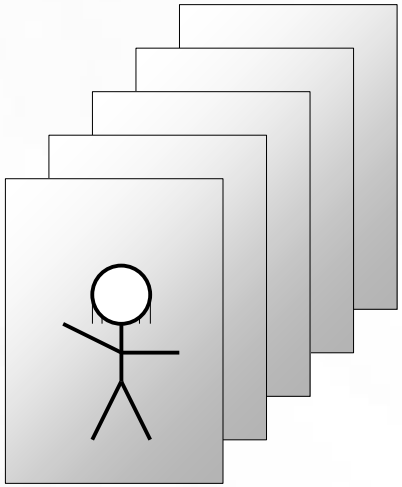
Snapshots

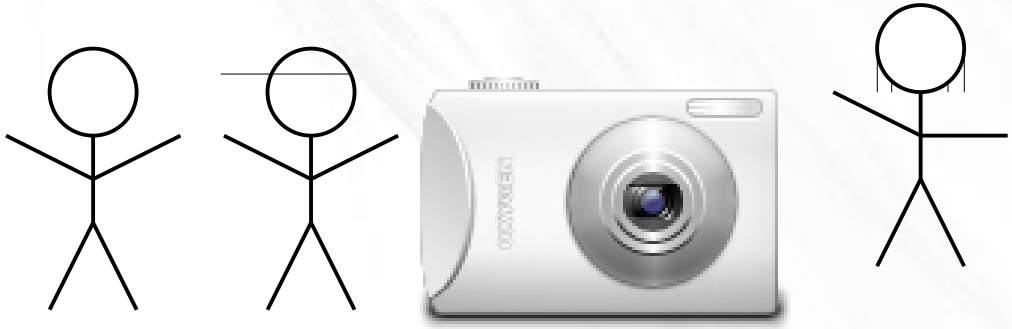
- Alfred, the photographer
- Hazel and her daughter



Snapshots

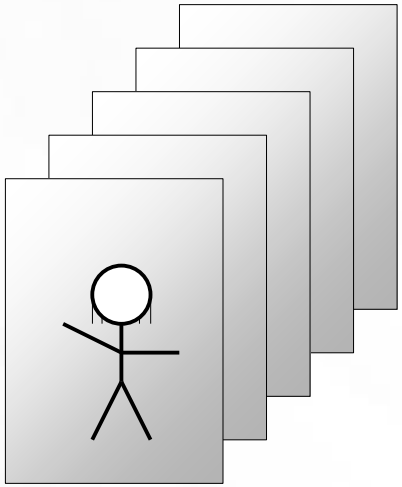
- Alfred, the photographer
- Hazel and her daughter
  - Remember what the daughter was like at each different stage

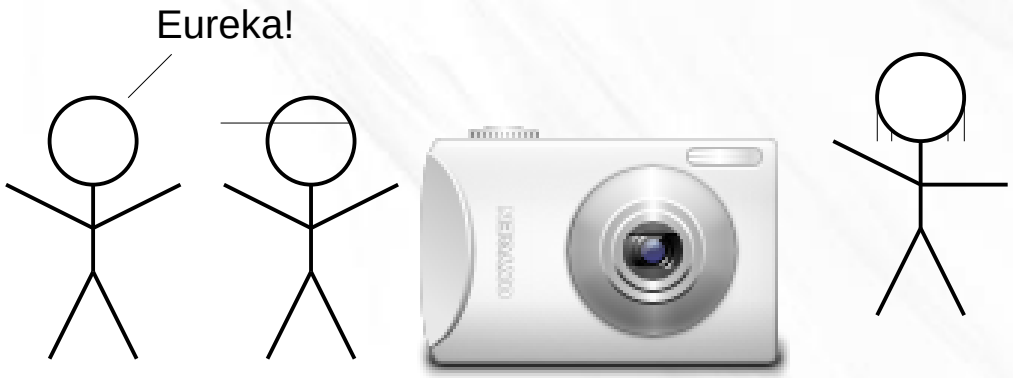




Snapshots

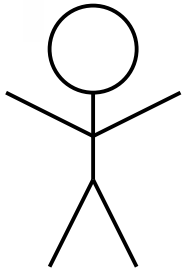
- Alfred, the photographer
- Hazel and her daughter
  - Remember what the daughter was like at each different stage



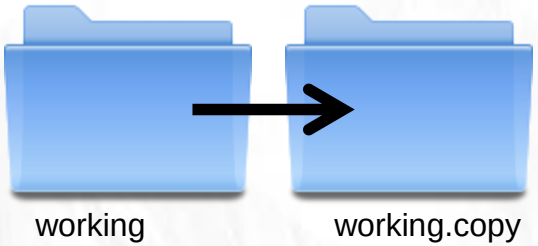


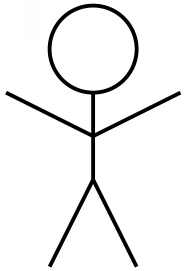
Snapshots

working

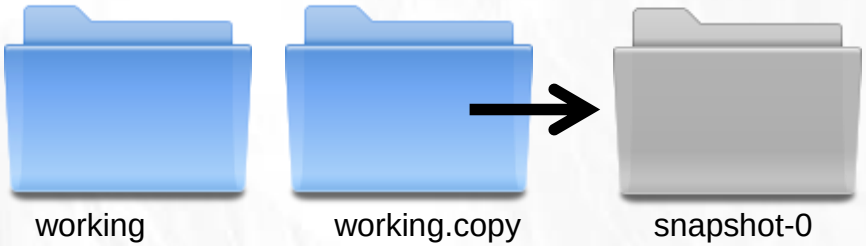


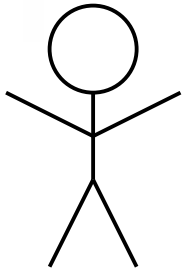
Snapshots



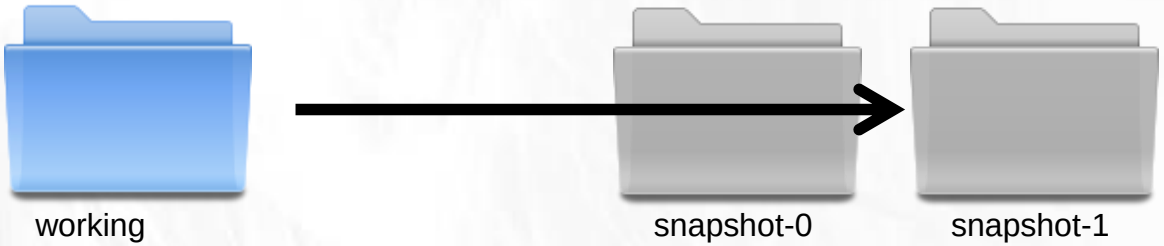


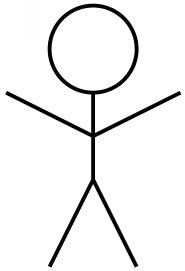
Snapshots





Snapshots

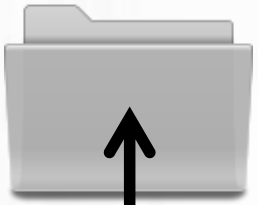




Snapshots

working

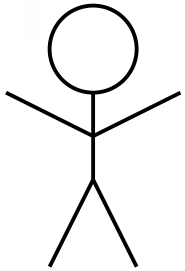
snapshot-0



snapshot-1

message

message



2009-05-20 12:34:56

Initial version

2009-05-21 23:45:01

Introduced a new foo, and reset the bar to xyzzy.

Branches

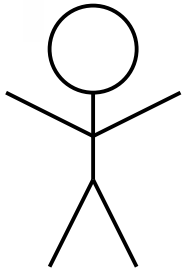
working

snapshot-0

snapshot-1

. . .

snapshot-99



Branches

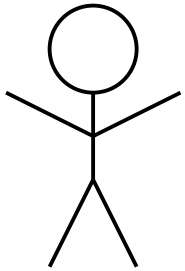
working

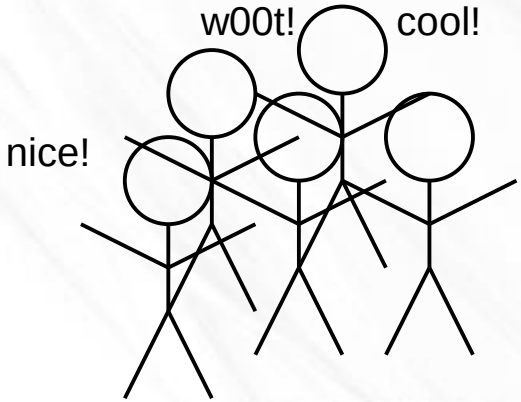
snapshot-0

snapshot-1

. . .

snapshot-99





Branches

working

snapshot-0

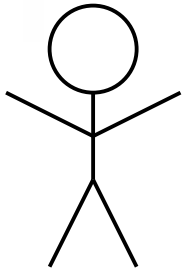
snapshot-1

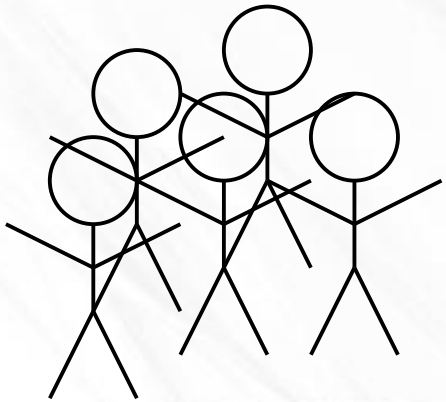
. . .

snapshot-99

. . .

snapshot-109





Branches

working

snapshot-0

snapshot-1

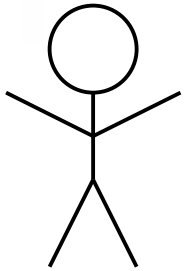
. . .

snapshot-99

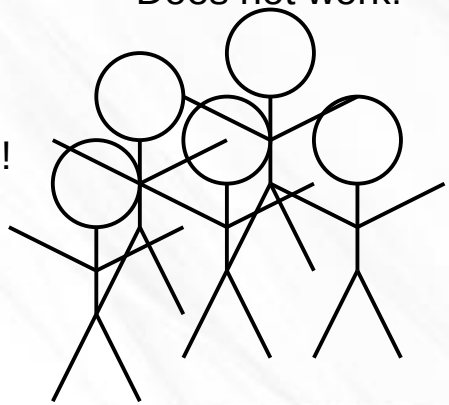
. . .

snapshot-109

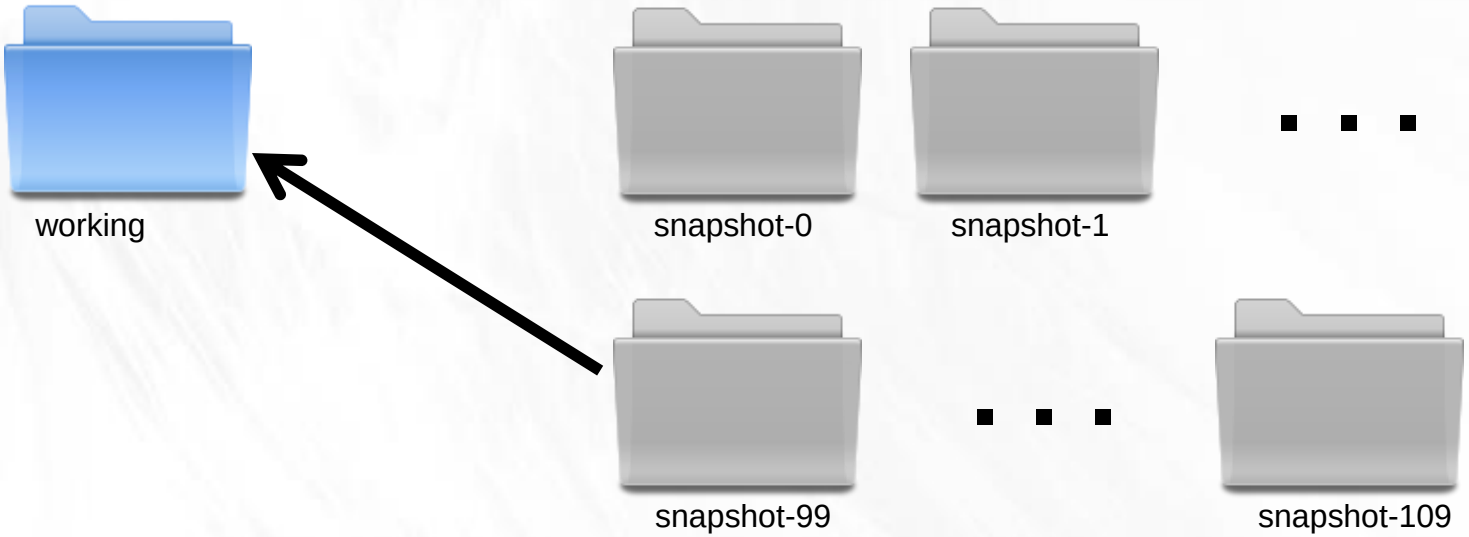
Does not work!

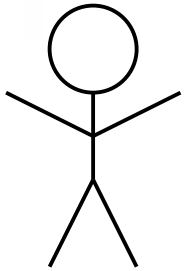


Boo!



Branches





Branches

working

snapshot-0

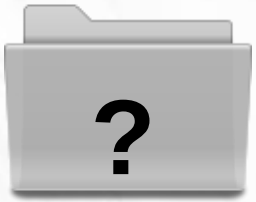
snapshot-1

. . .

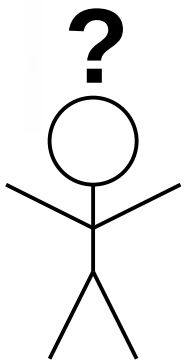
snapshot-99

. . .

snapshot-109

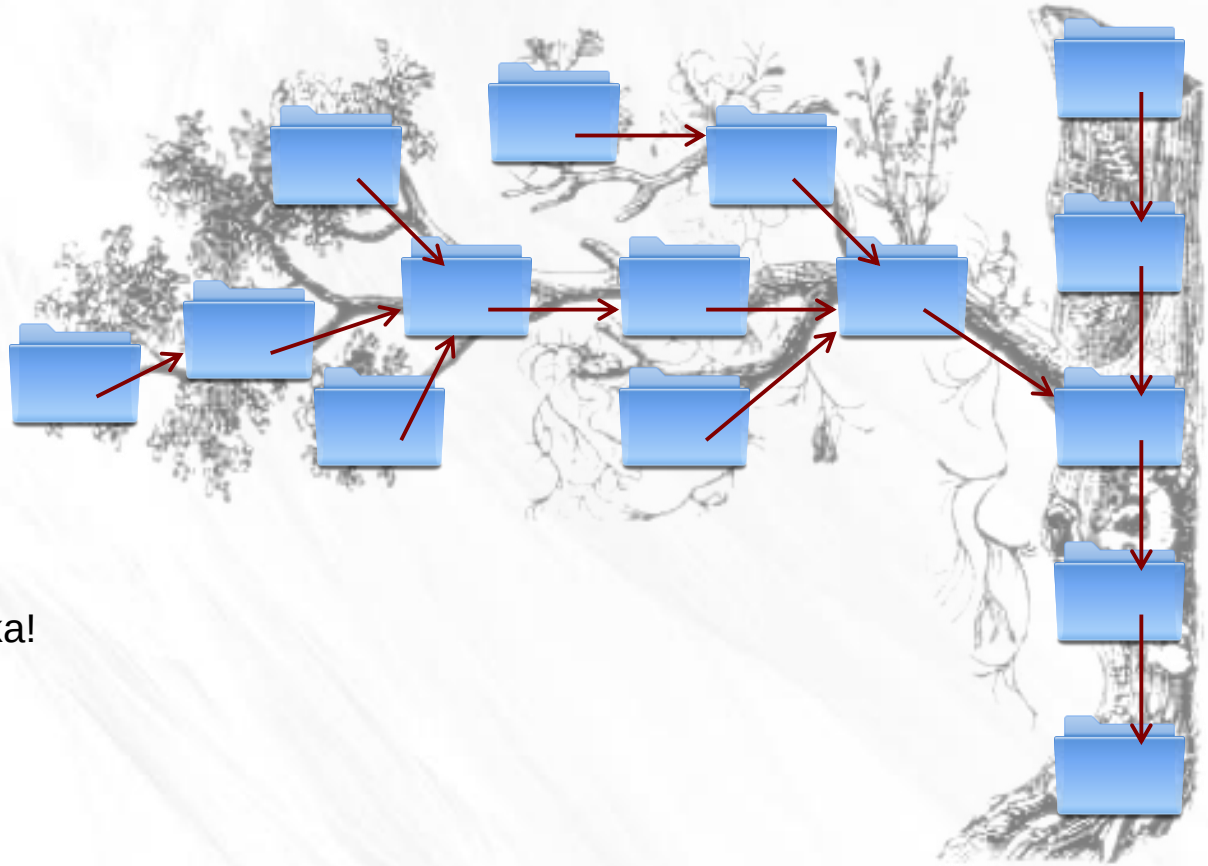


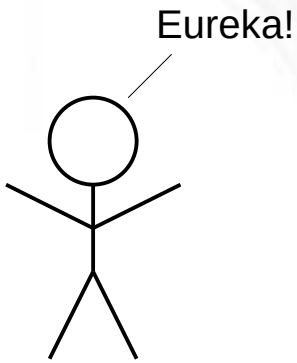
snapshot-110



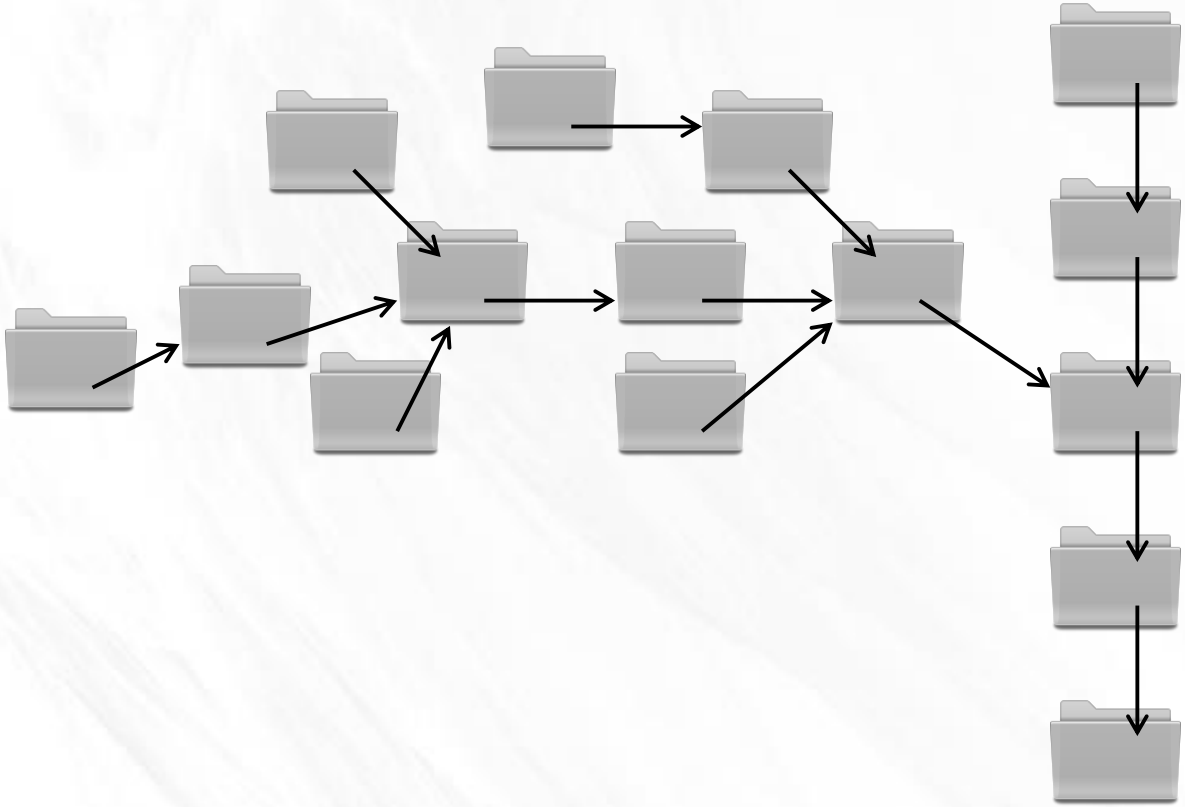
Branches

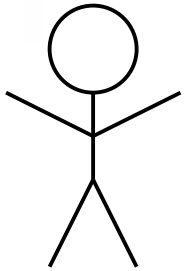
Branches



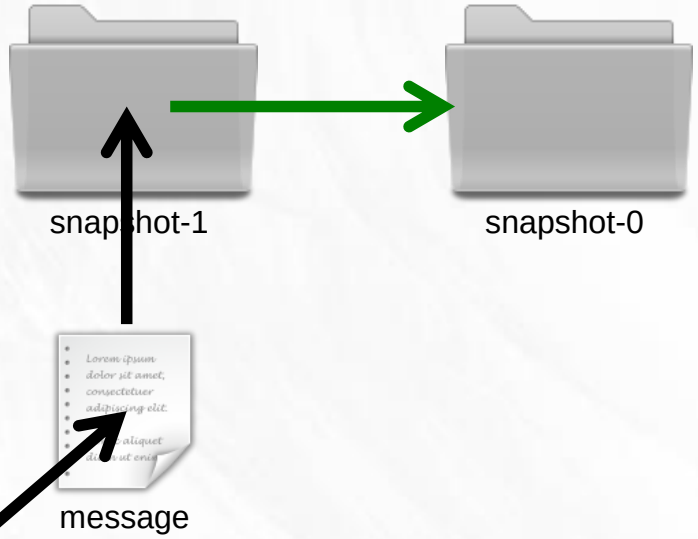


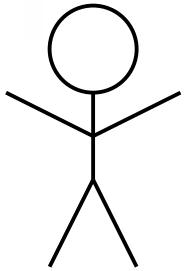
Branches





Branches



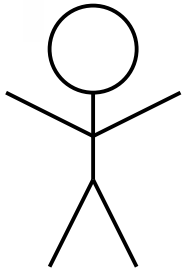


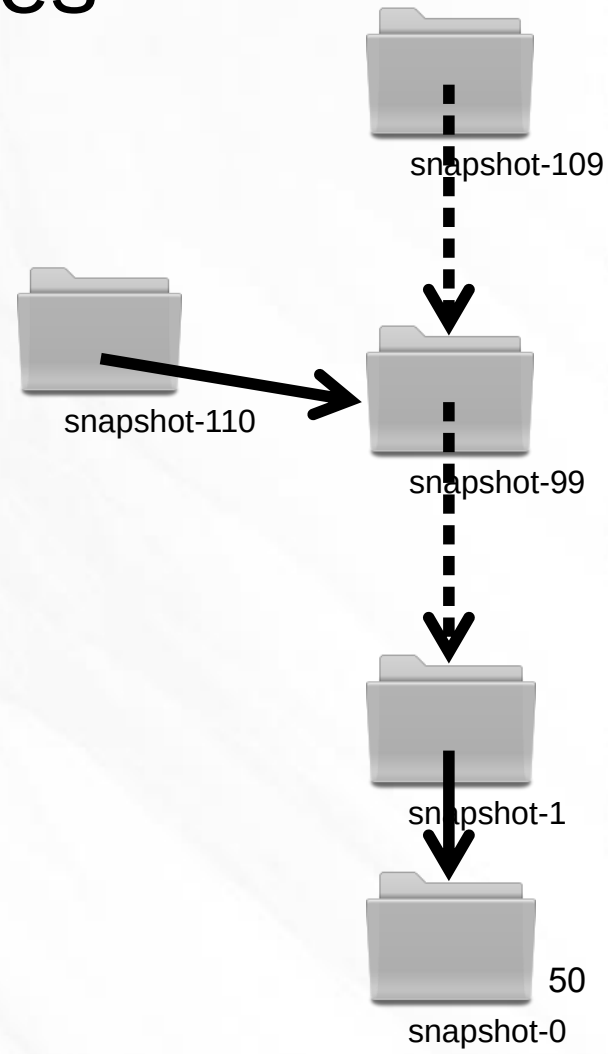
2009-05-21 23:45:01
parent: snapshot-0

Introduced a new foo, and reset the bar to xyzzy.

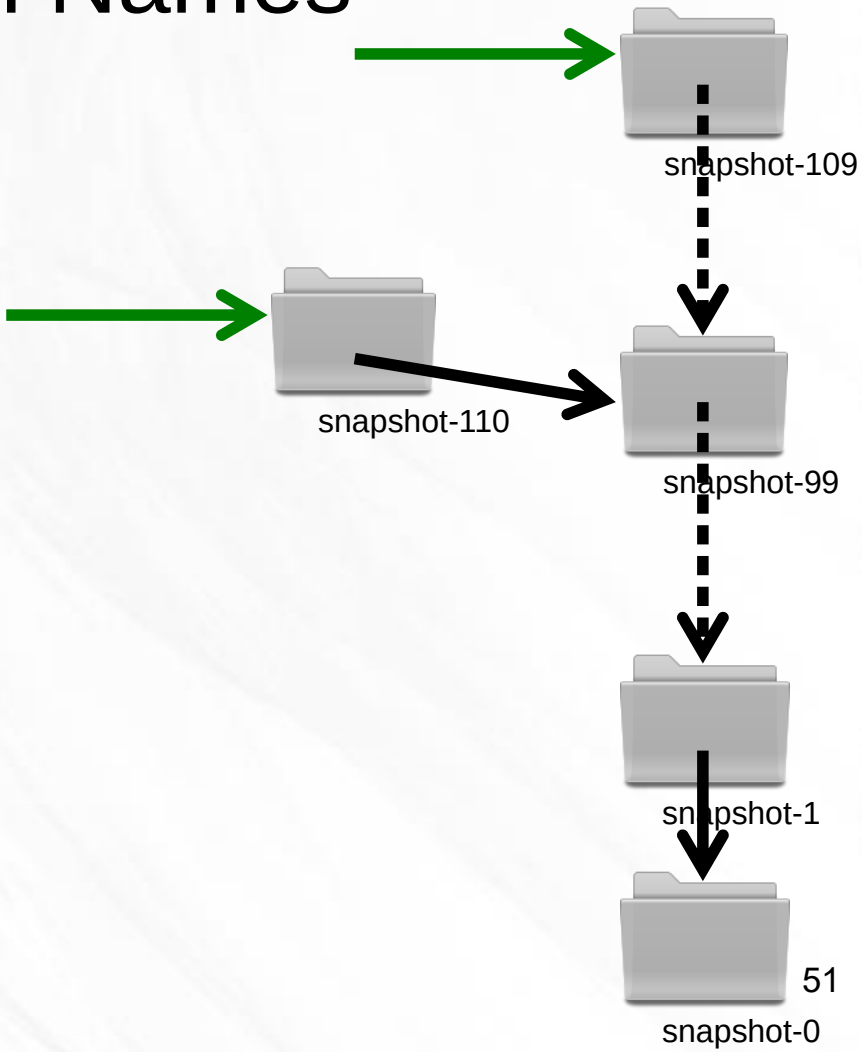
Branch Names

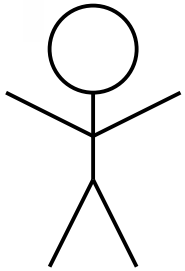
working



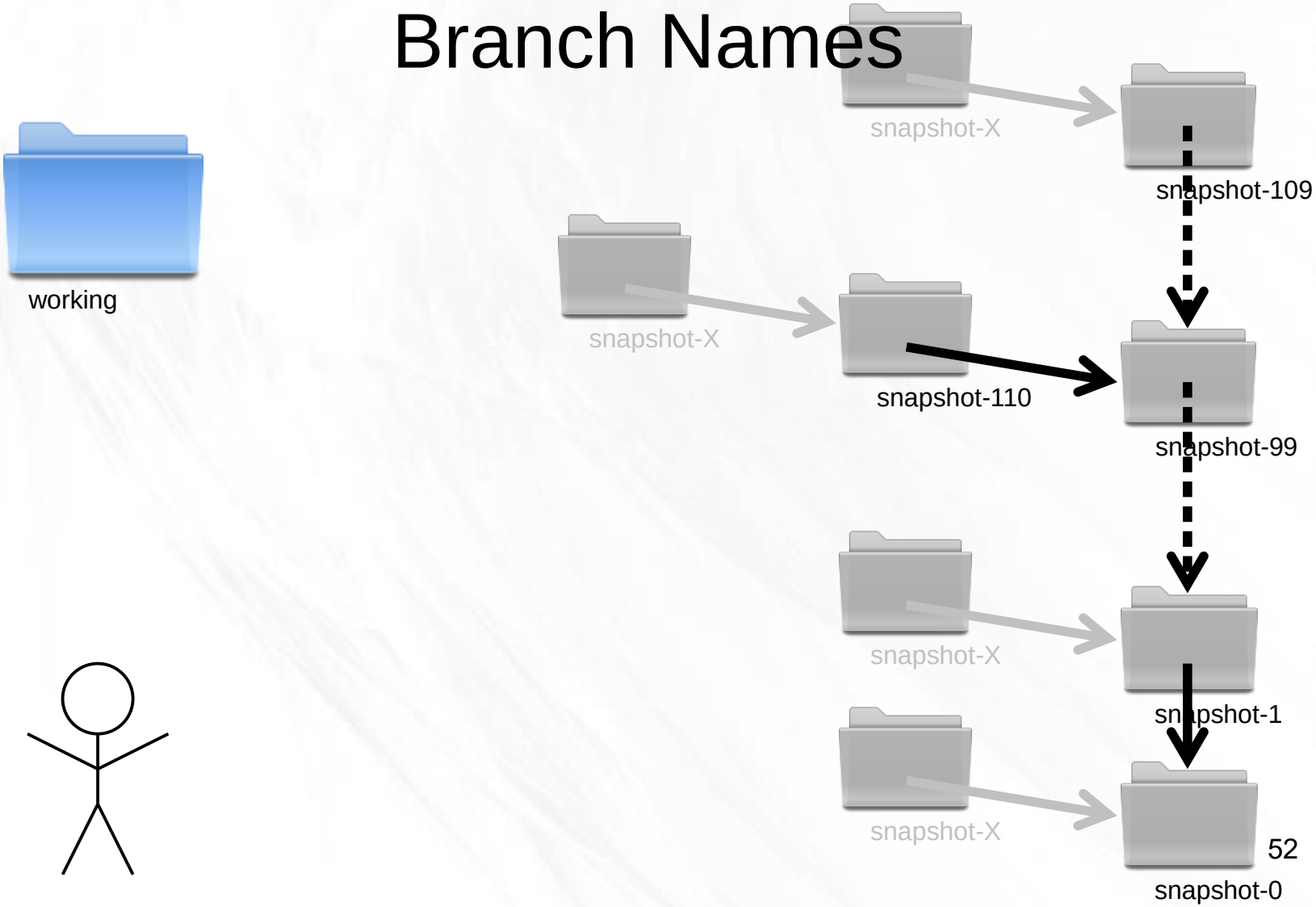


Branch Names

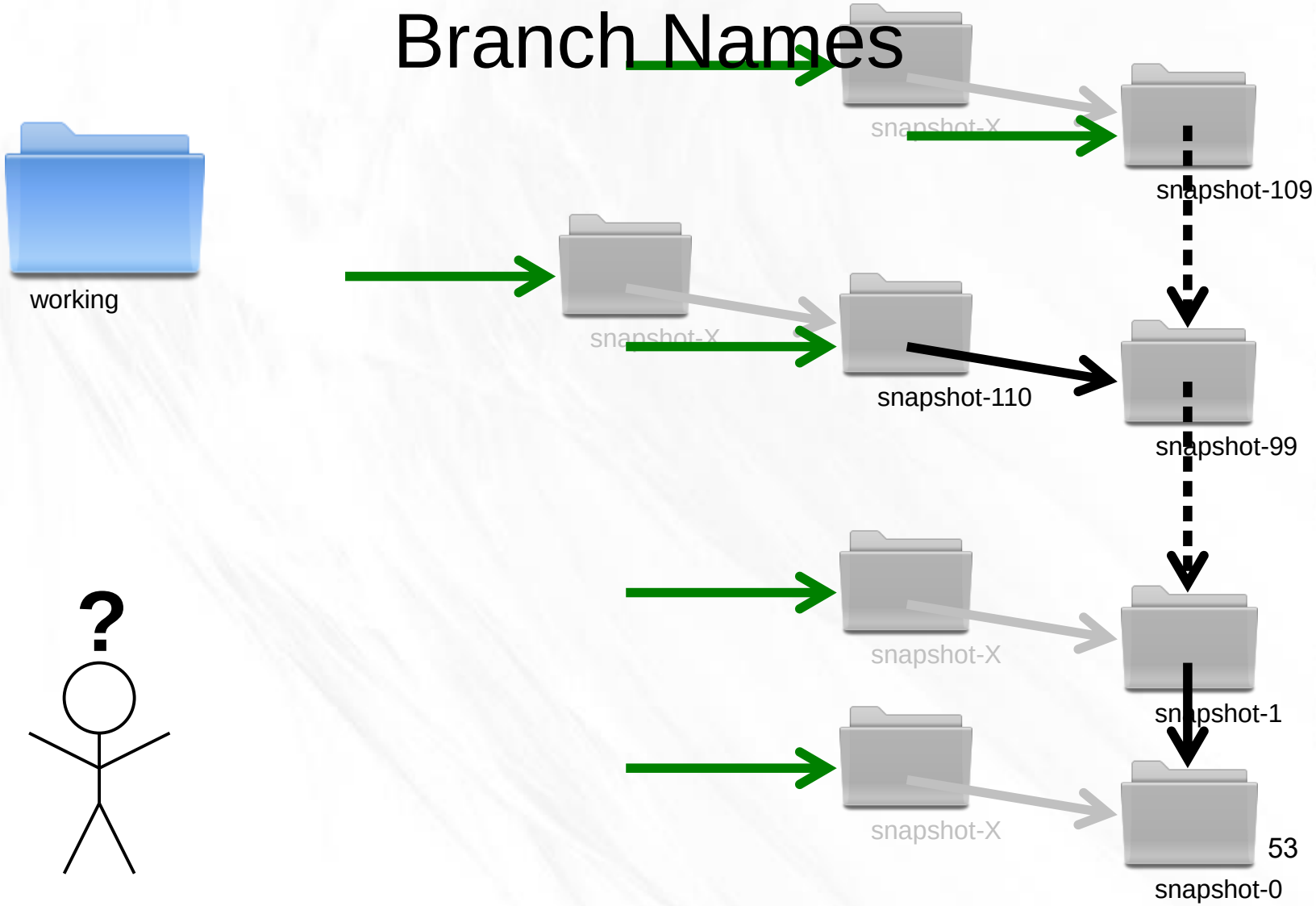




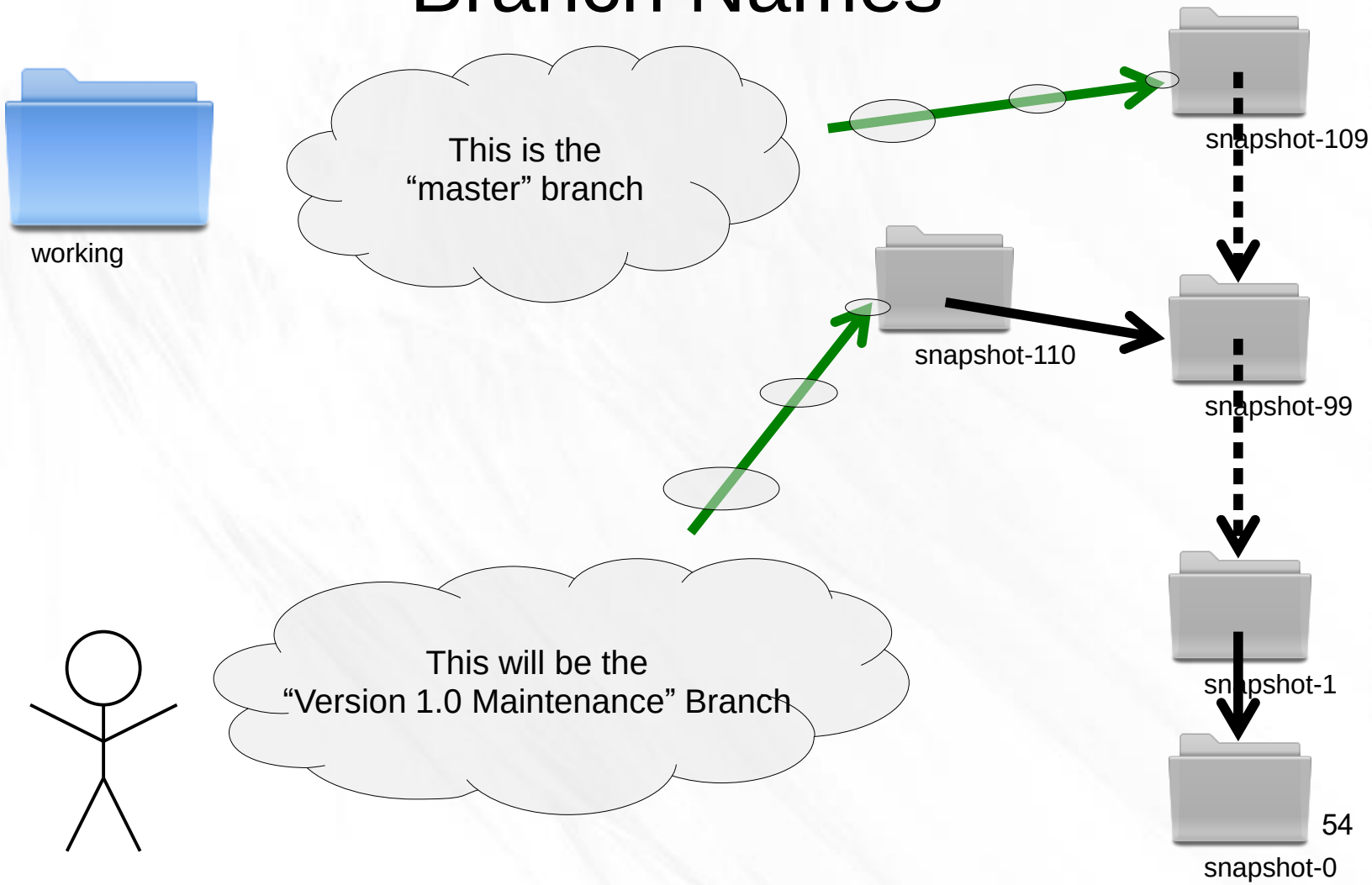
Branch Names



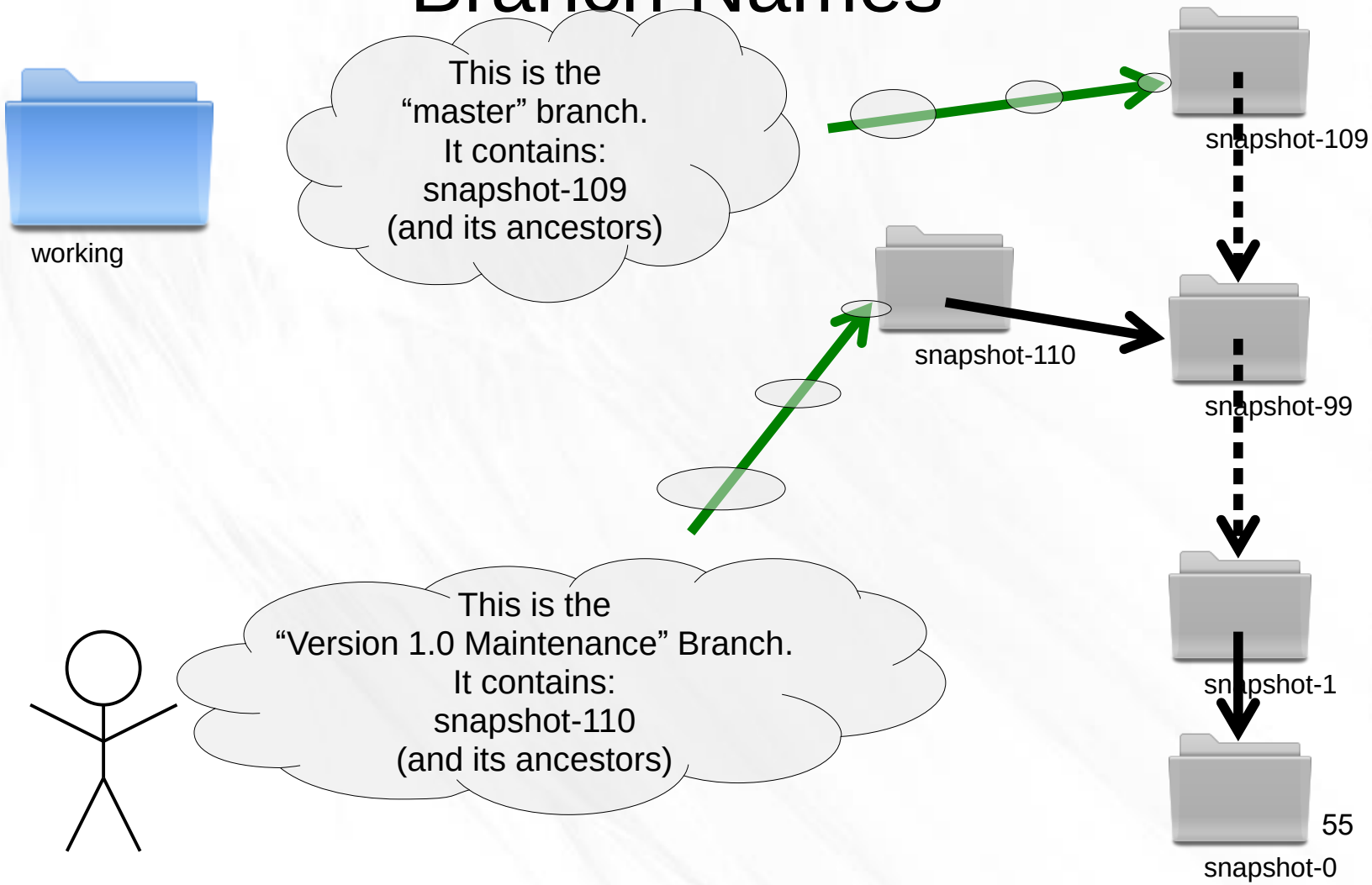
Branch Names



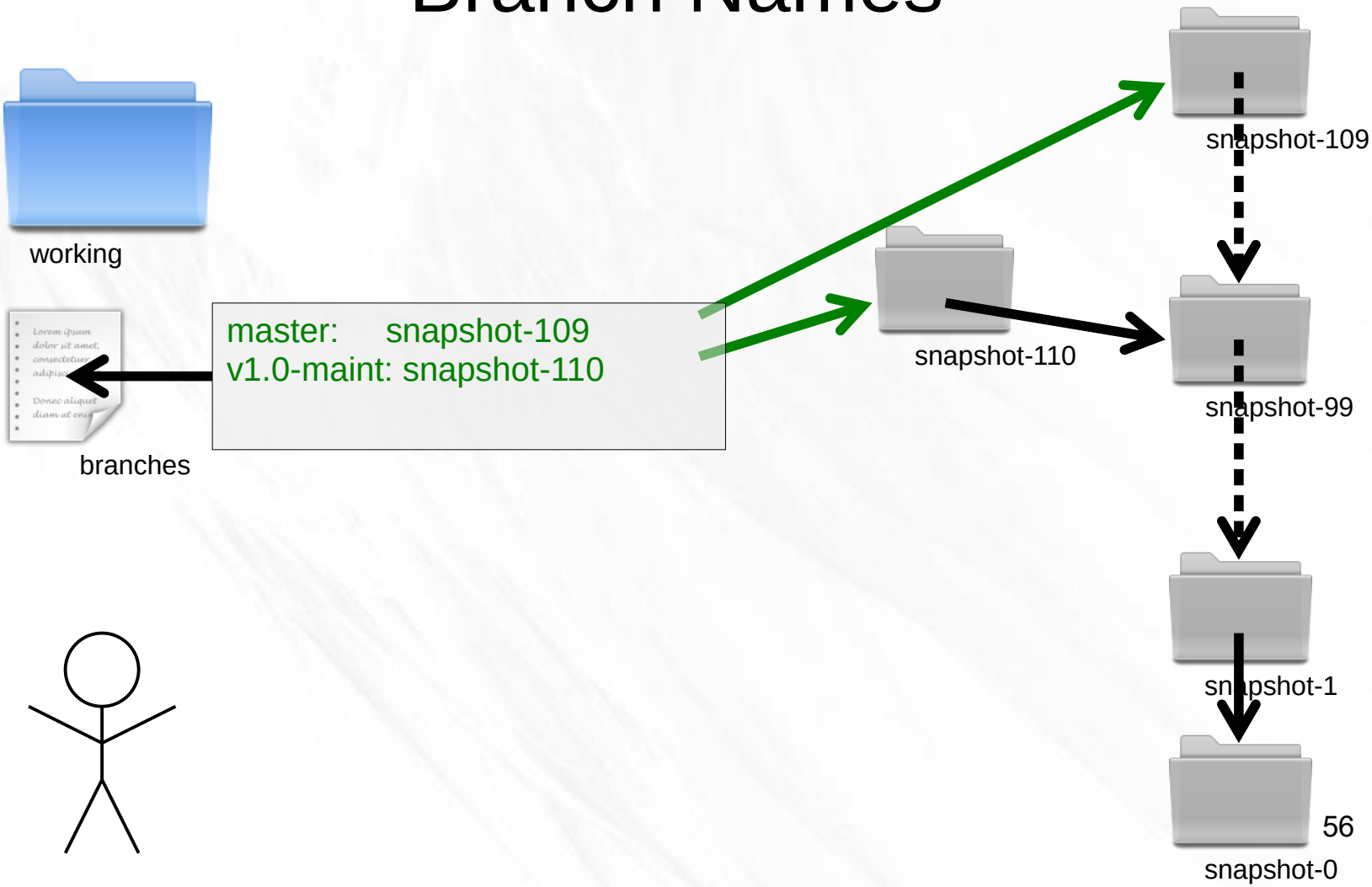
Branch Names



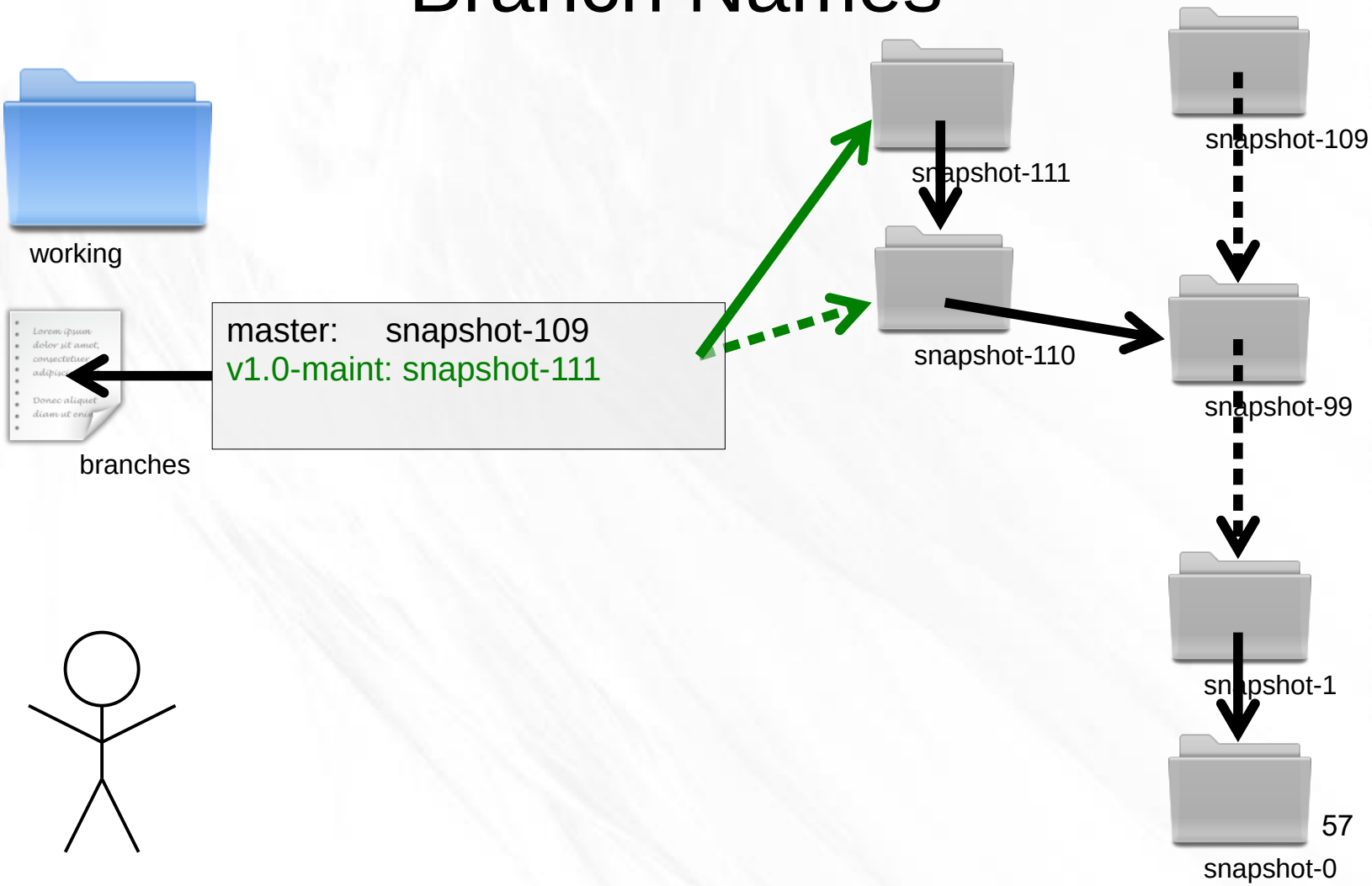
Branch Names



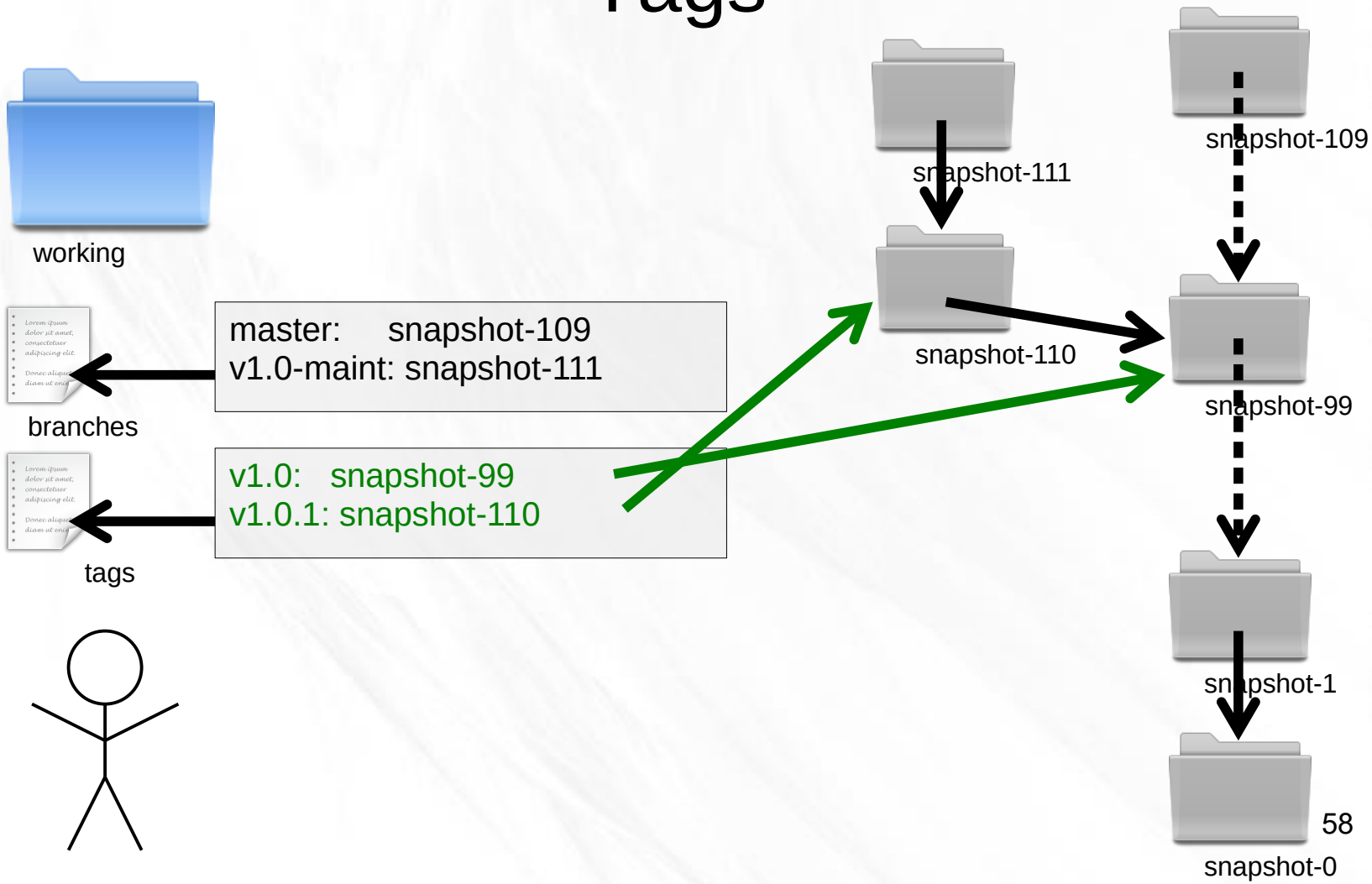
Branch Names



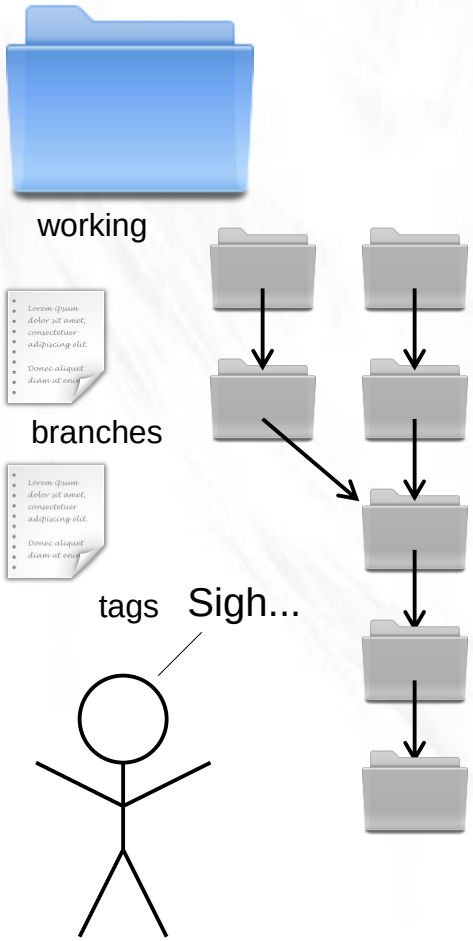
Branch Names



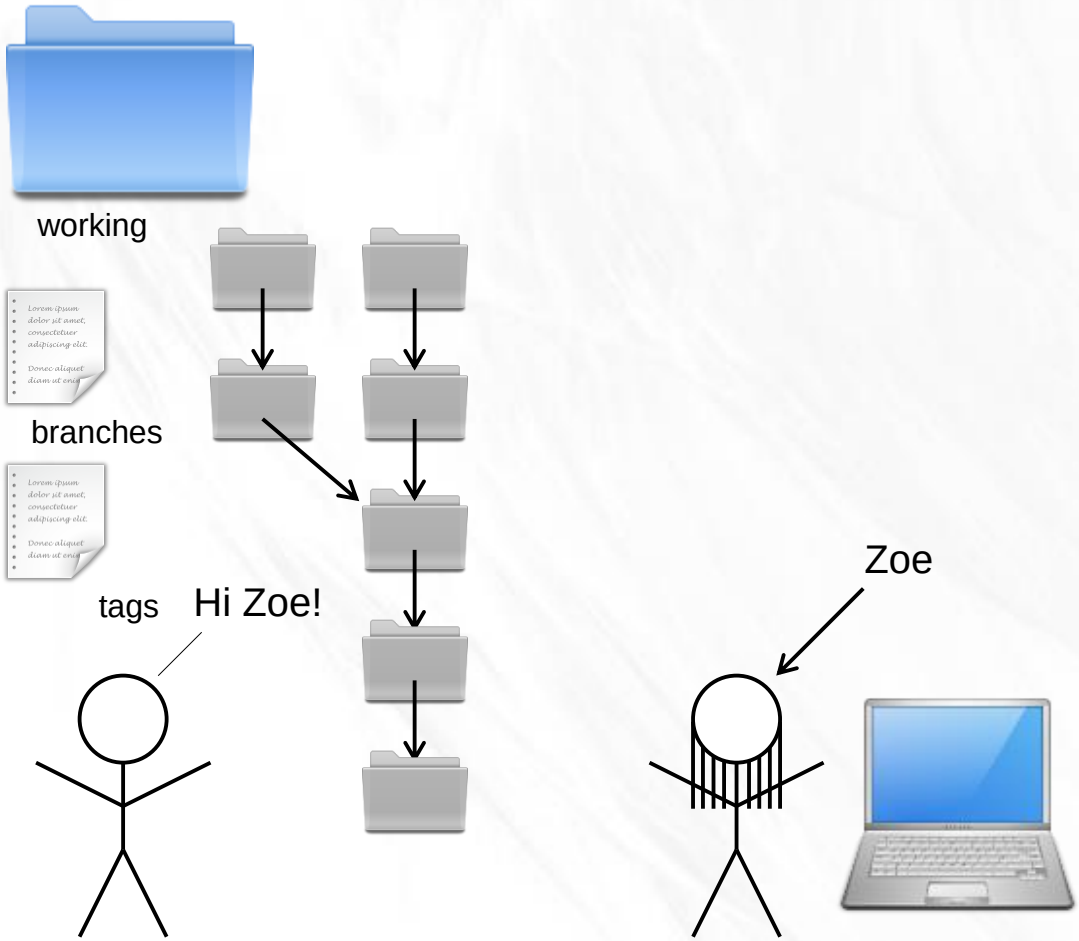
Tags


working
branches
tags
master: snapshot-109
v1.0-maint: snapshot-111
v1.0: snapshot-99
v1.0.1: snapshot-110
snapshot-111
snapshot-110
snapshot-109
snapshot-99
snapshot-1
snapshot-0

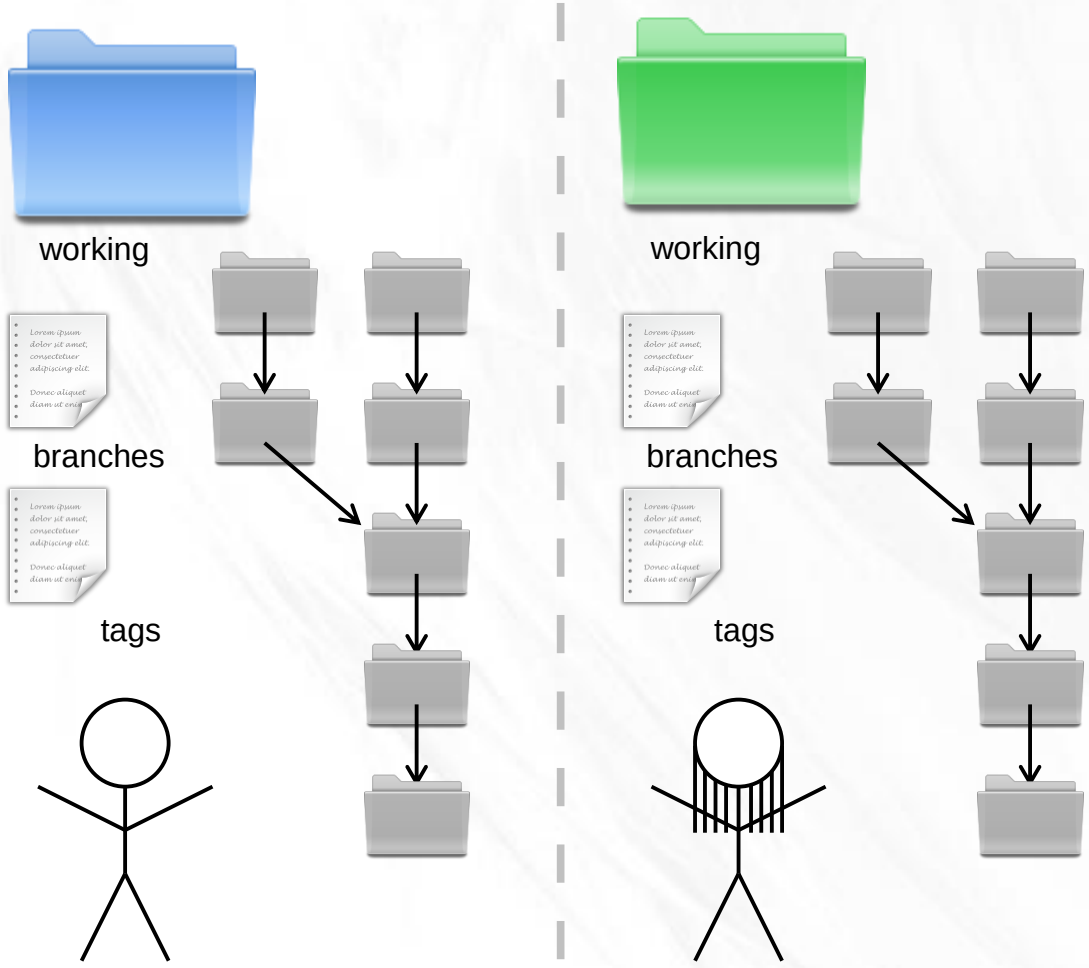
Distributed



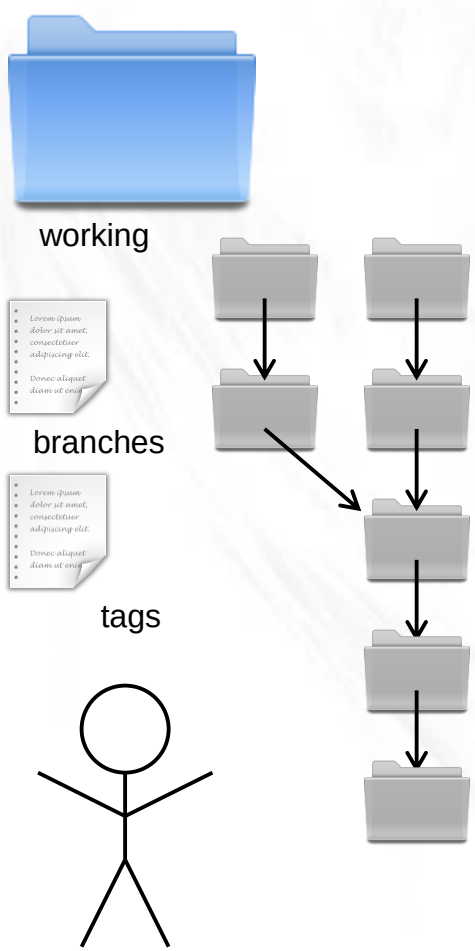
Distributed

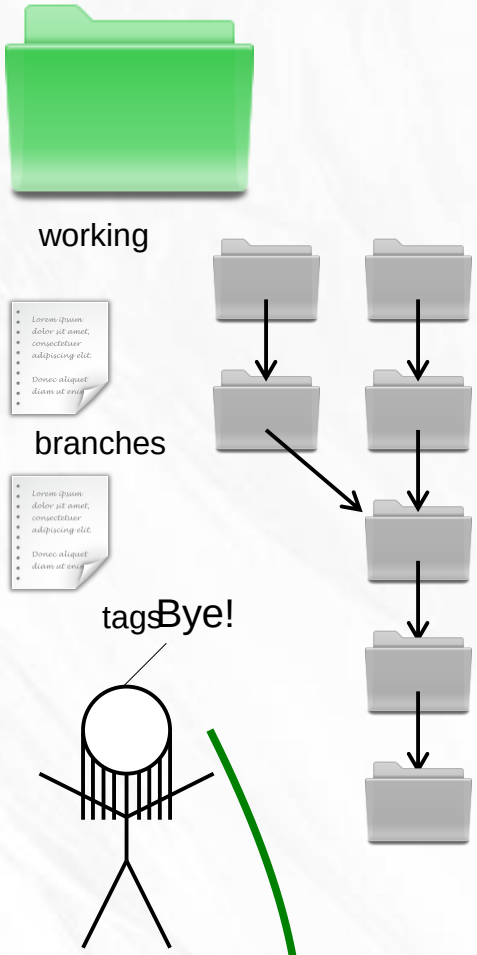


Distributed

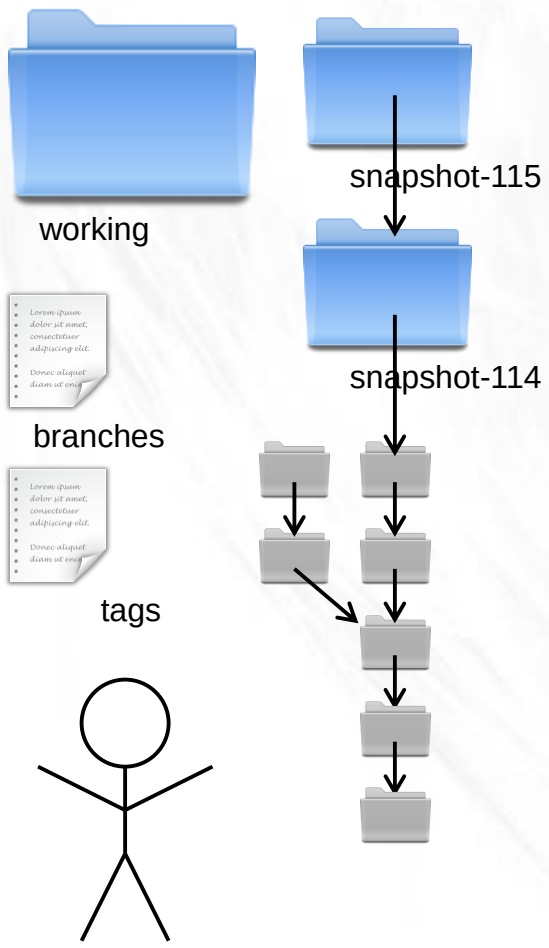


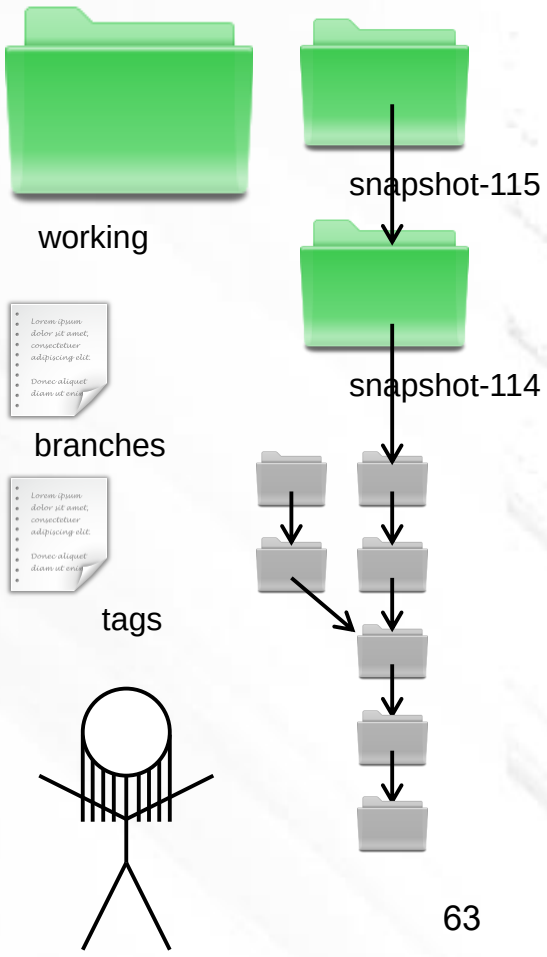
Distributed



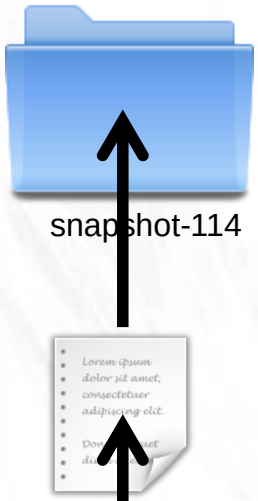


Distributed



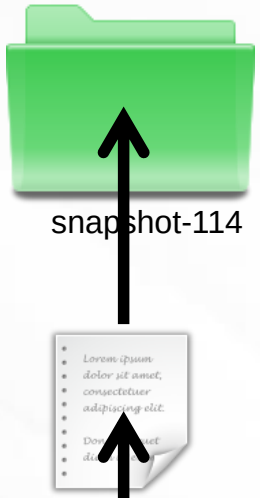


Distributed



2009-05-22 12:12:12
parent: snapshot-113
author: Me <me@me.me>

Blarfle, a cool new feature; extends the existing blorg.



2009-05-21 23:45:01
parent: snapshot-113
author: Zoe <zoe@z.oe>

Introduced a new foo, and reset the bar to xyzzy.

Distributed

8ba3441b6b89cad23387ee875f2ae55069291f4b

SHA1

message

2009-05-22 12:12:12
parent: snapshot-113
author: Me <me@me.me>

Blarfle, a cool new feature; extends the existing blorg.

db9ecb5b5a6294a8733503ab57577db96ff2249e

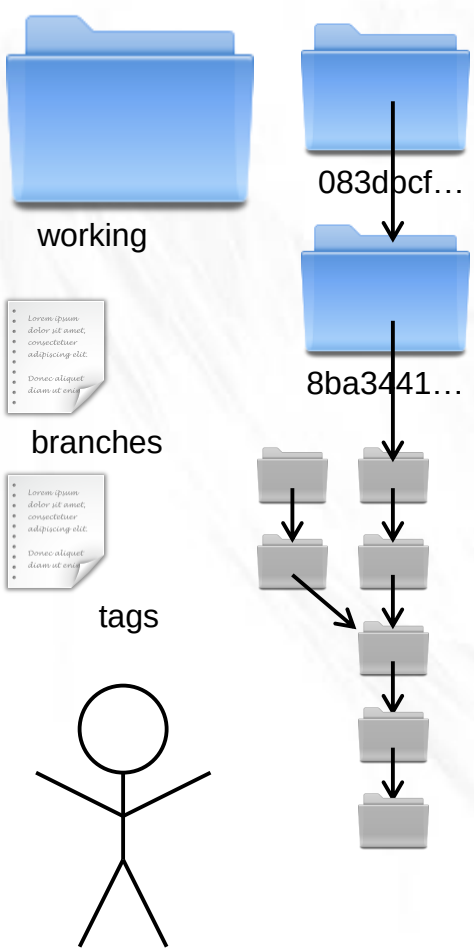
SHA1

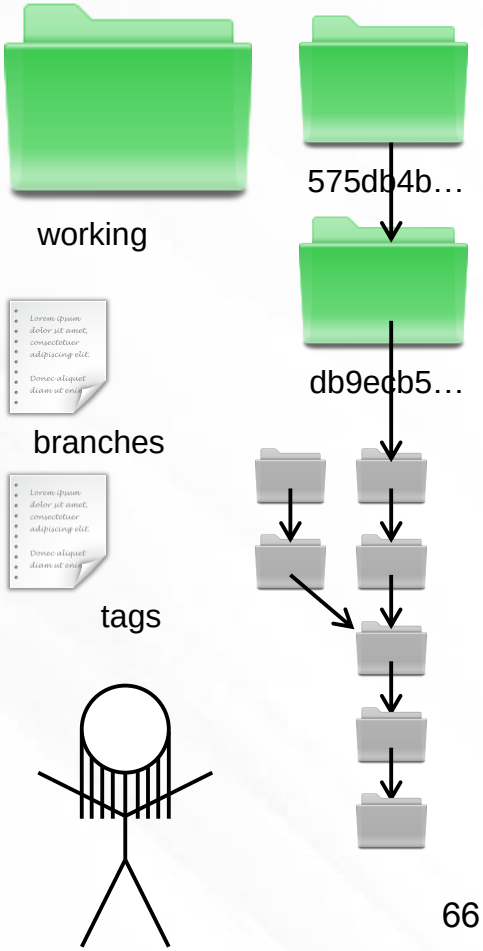
message

2009-05-21 23:45:01
parent: snapshot-113
author: Zoe <zoe@z.oe>

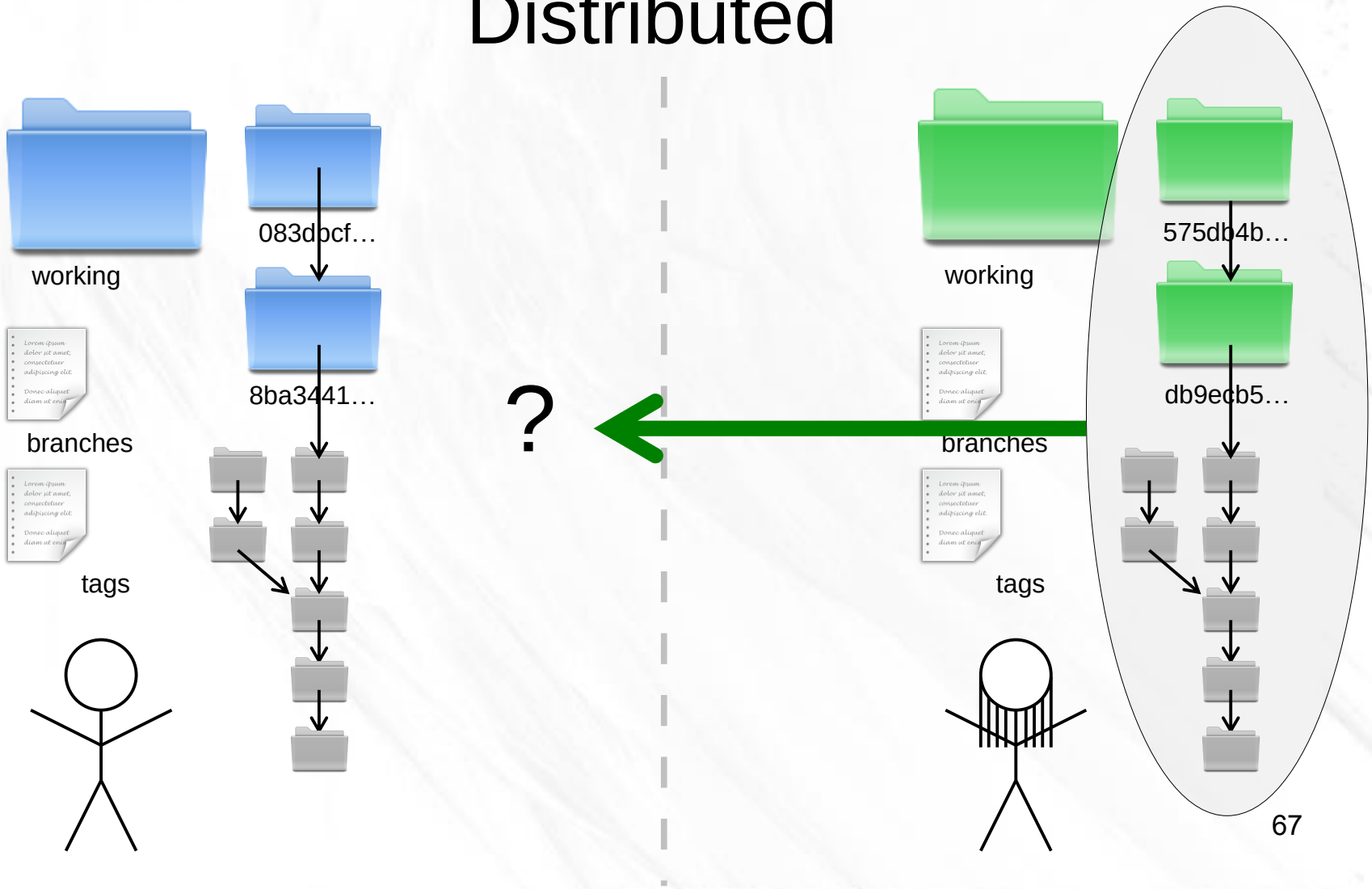
Introduced a new foo, and reset the bar to xyzzy.

Distributed

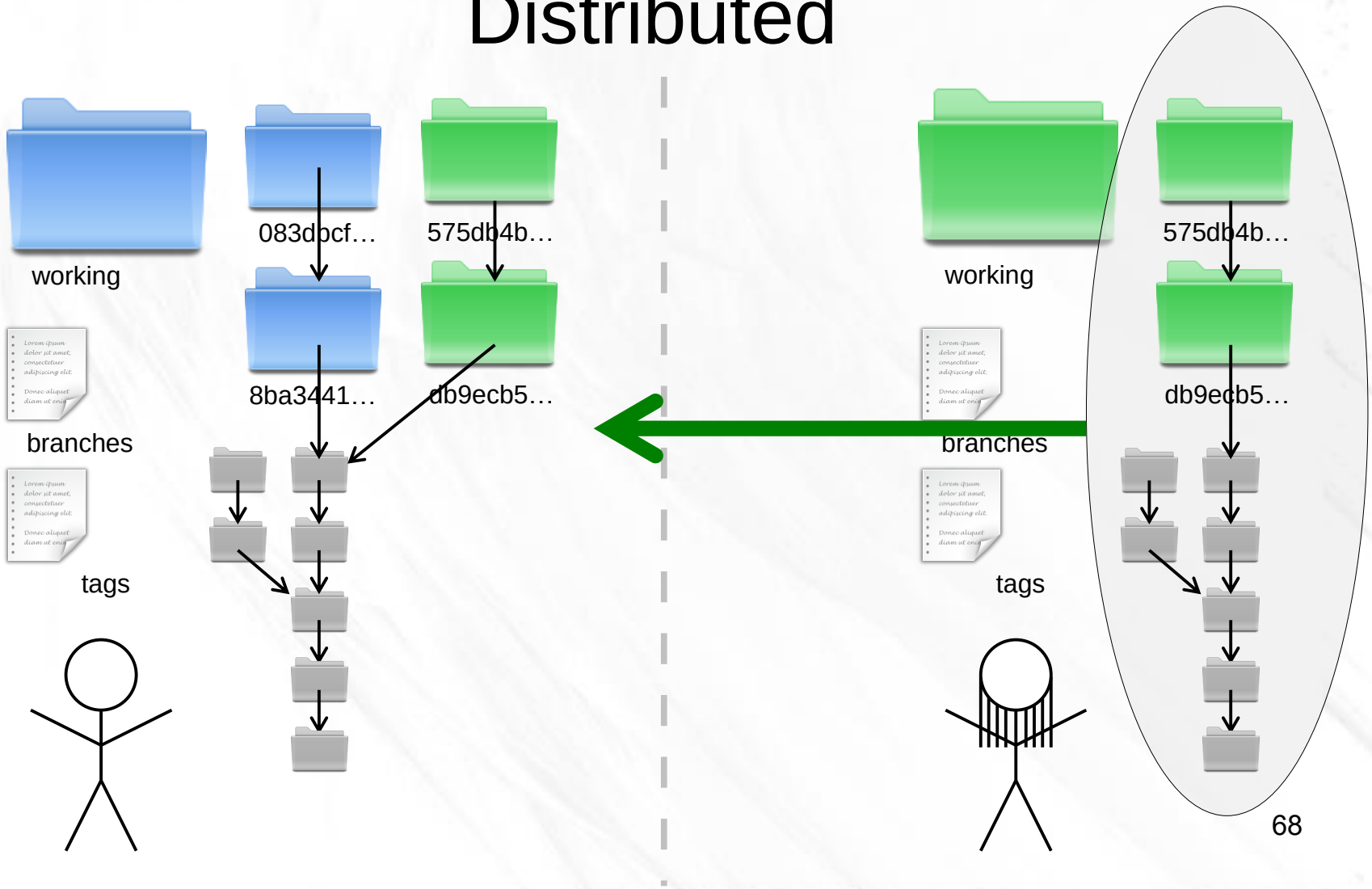




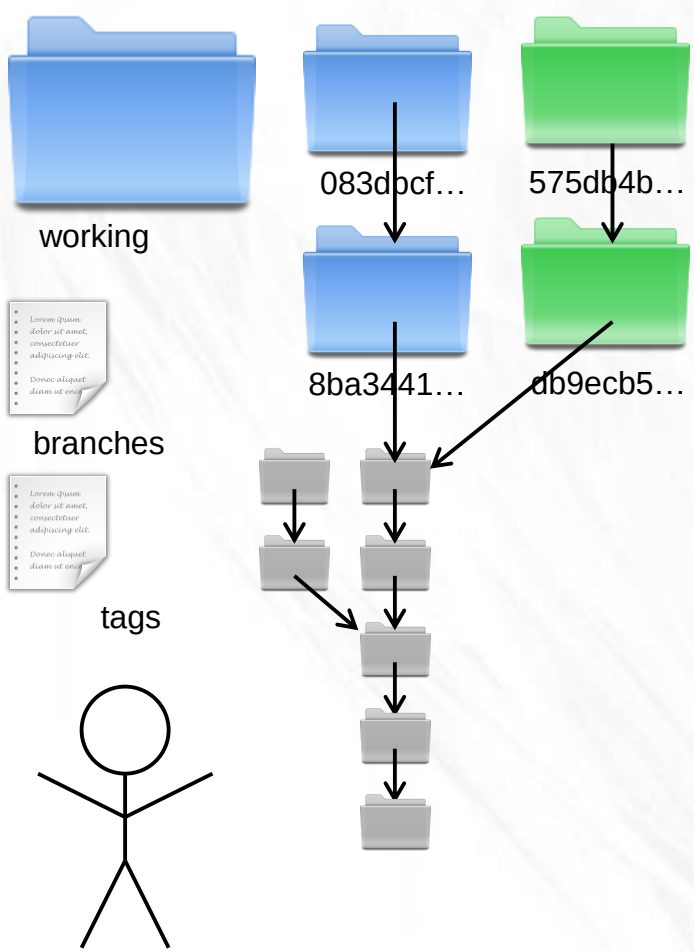
Distributed

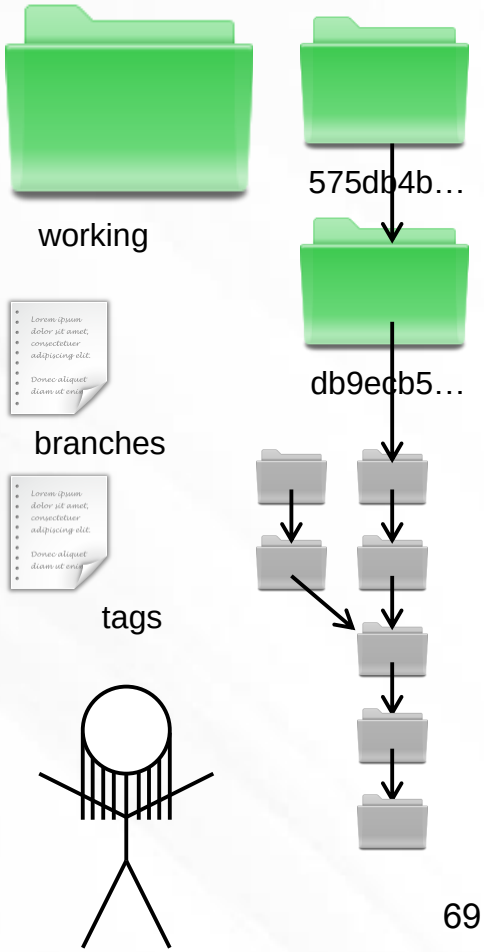


Distributed

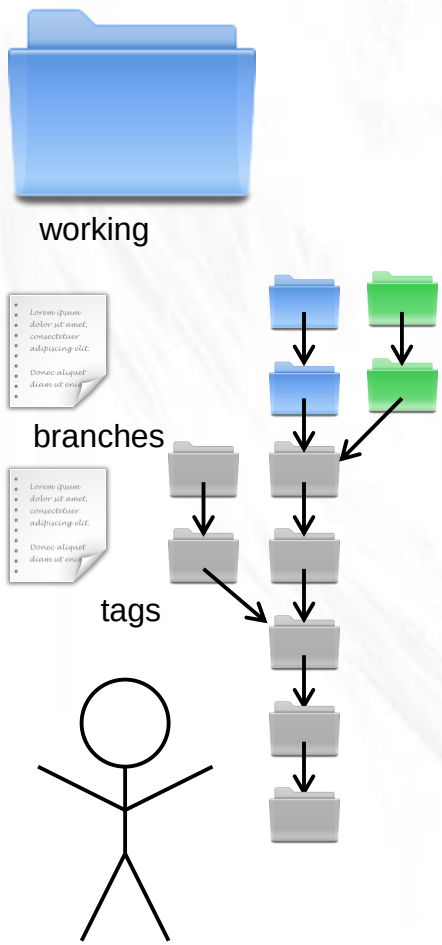


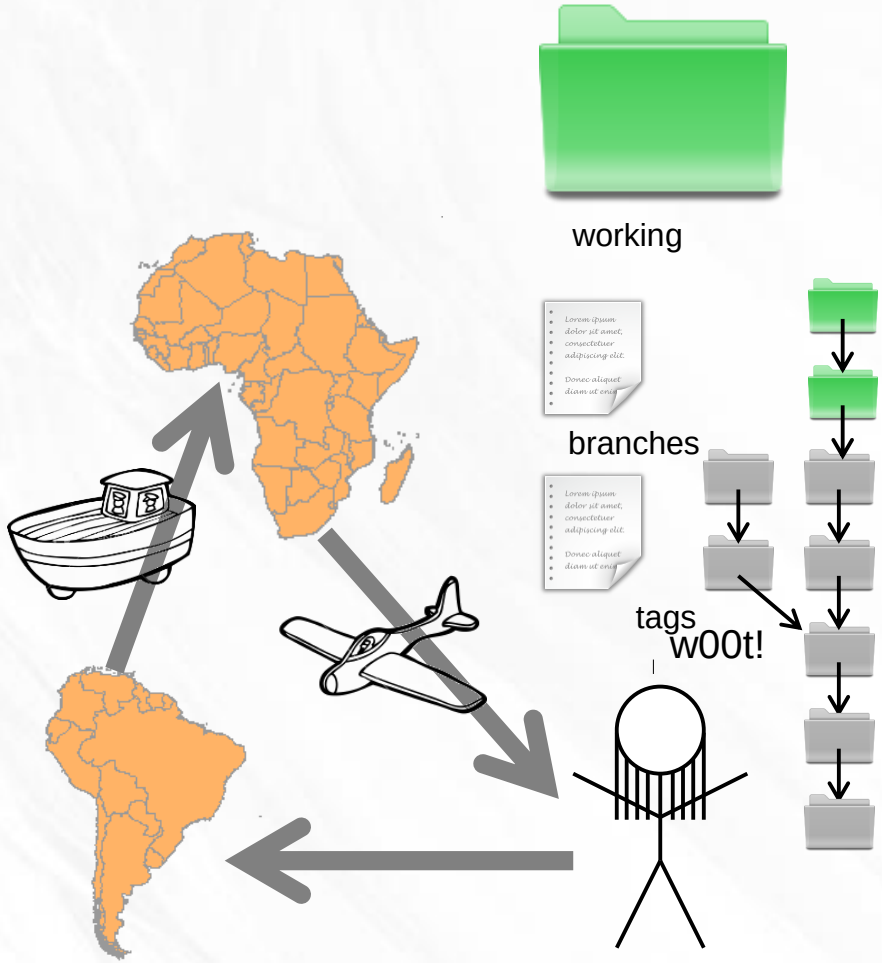
Distributed



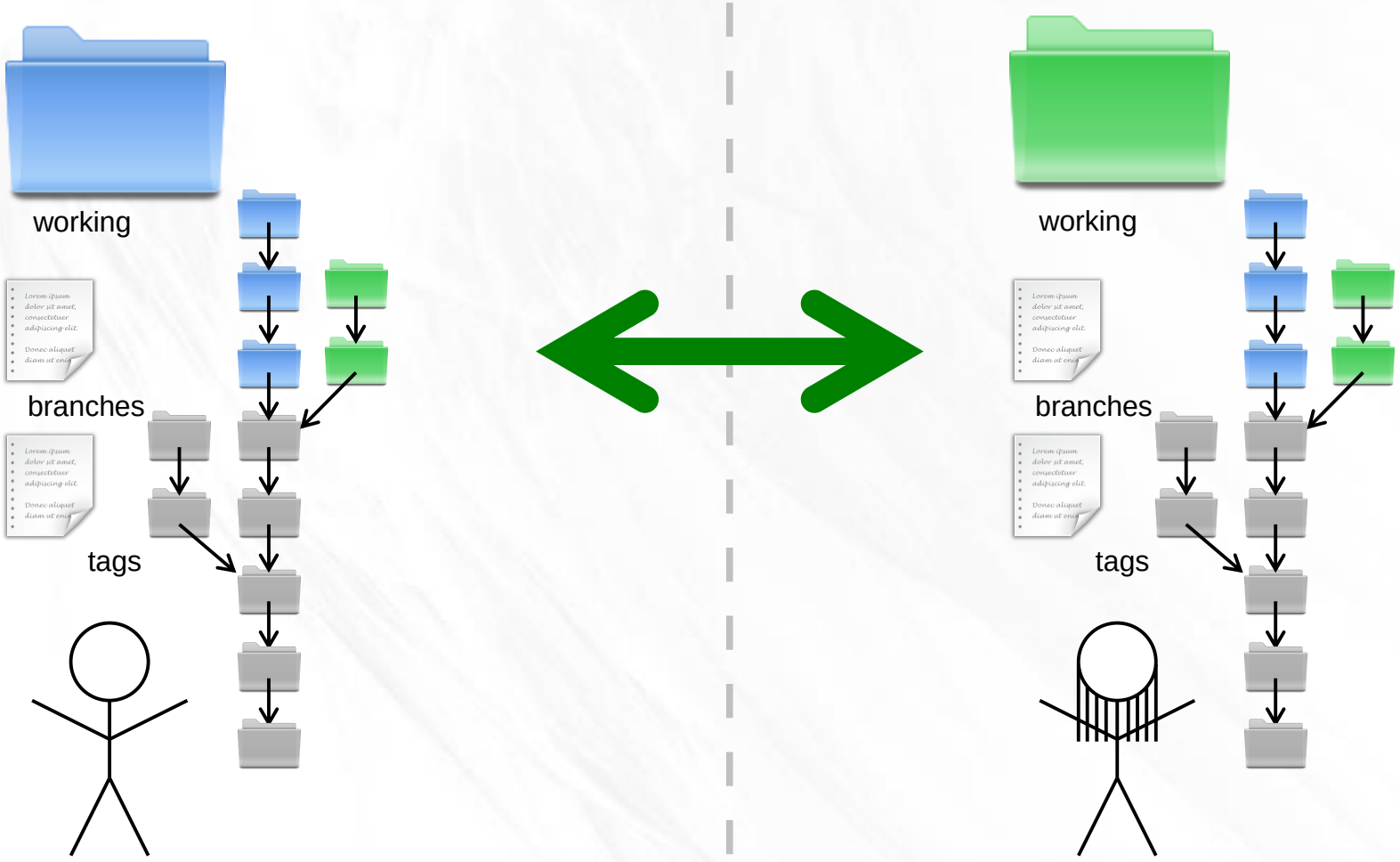


Offline

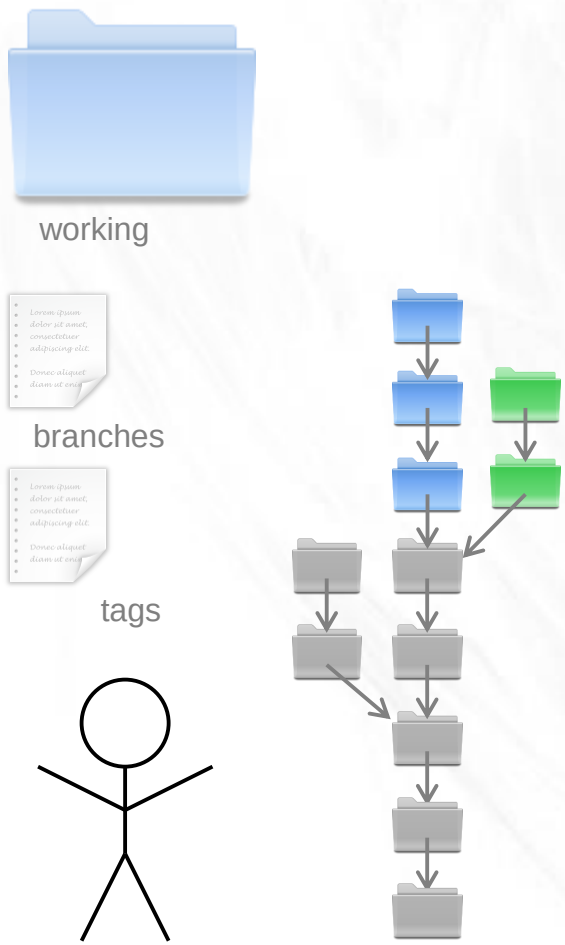


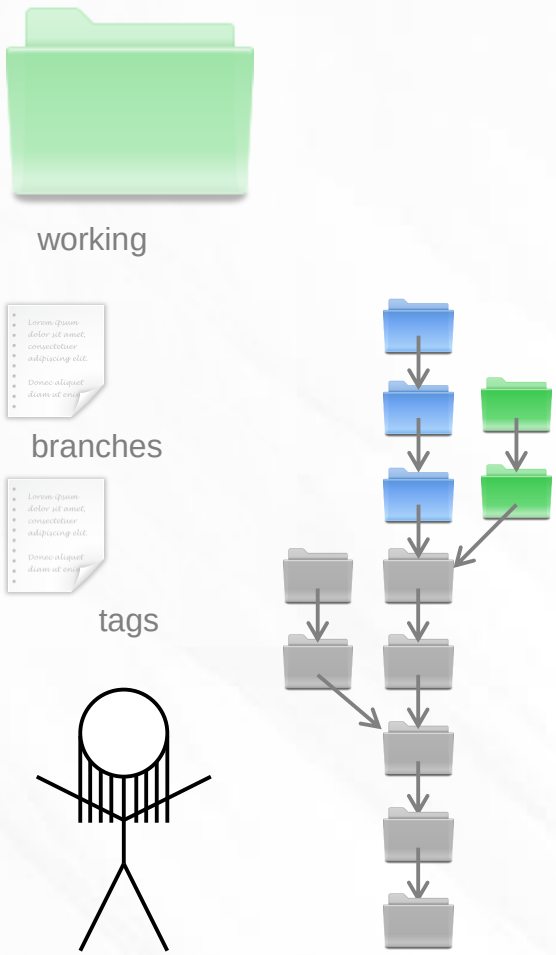


Offline

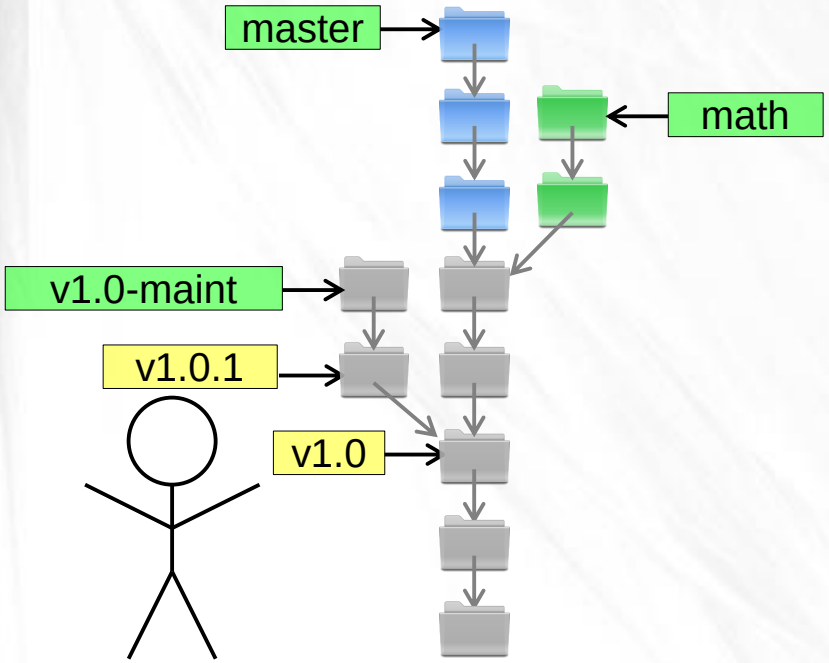


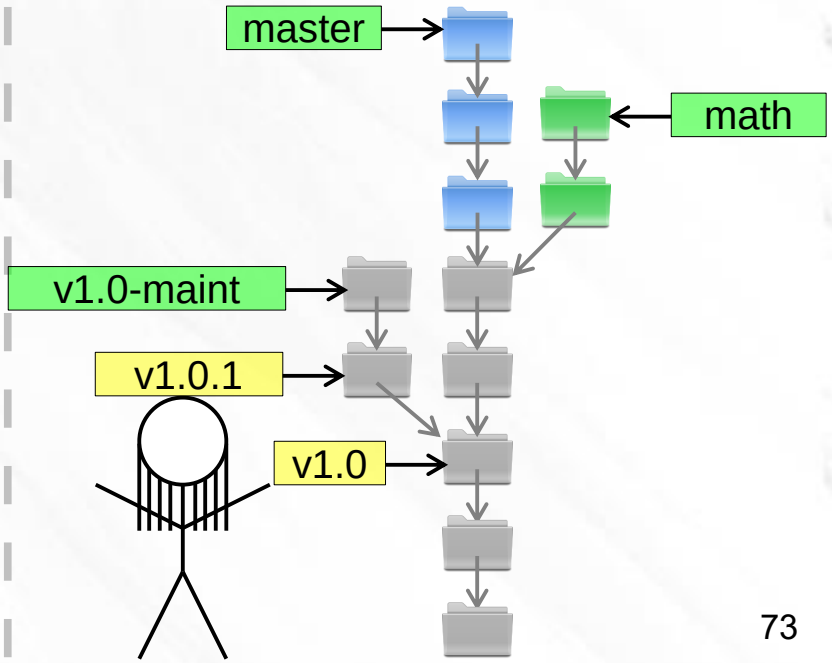
(simpler drawings)



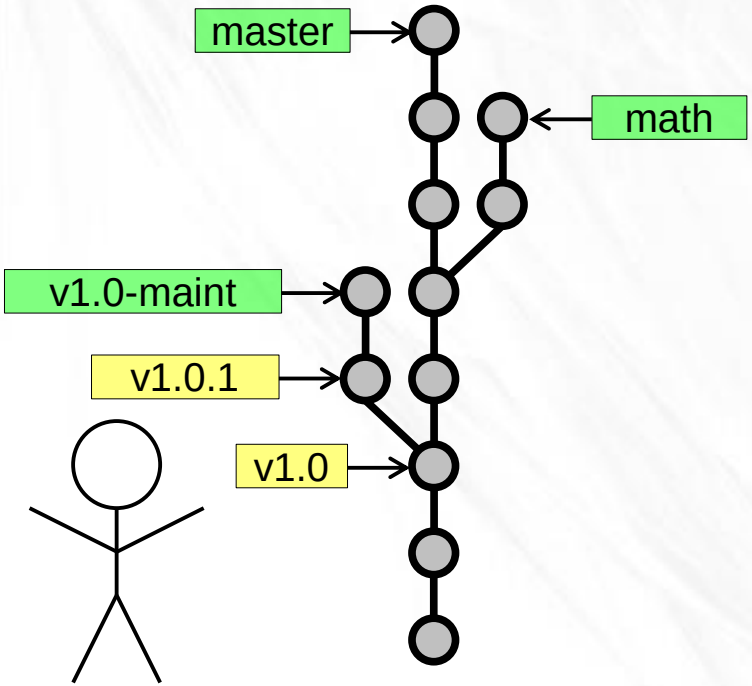


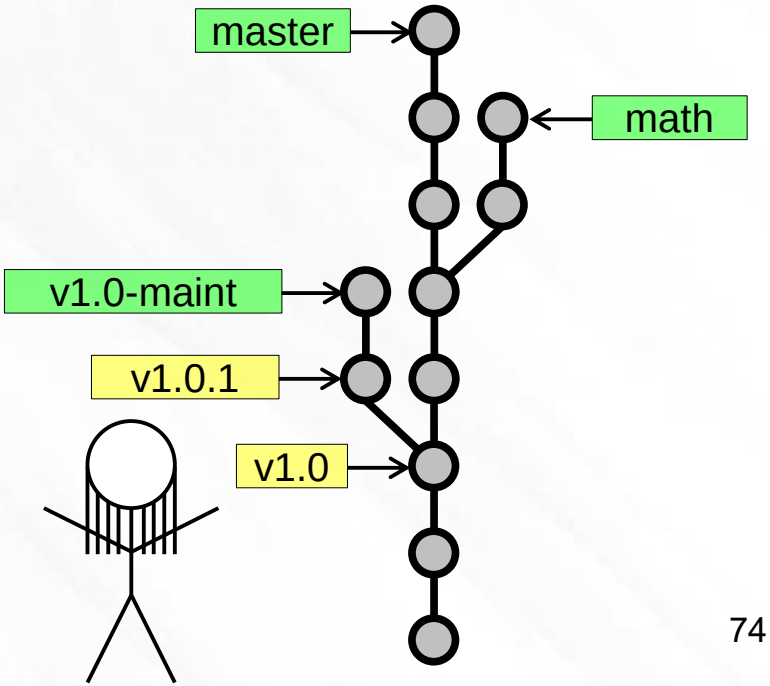
(simpler drawings)



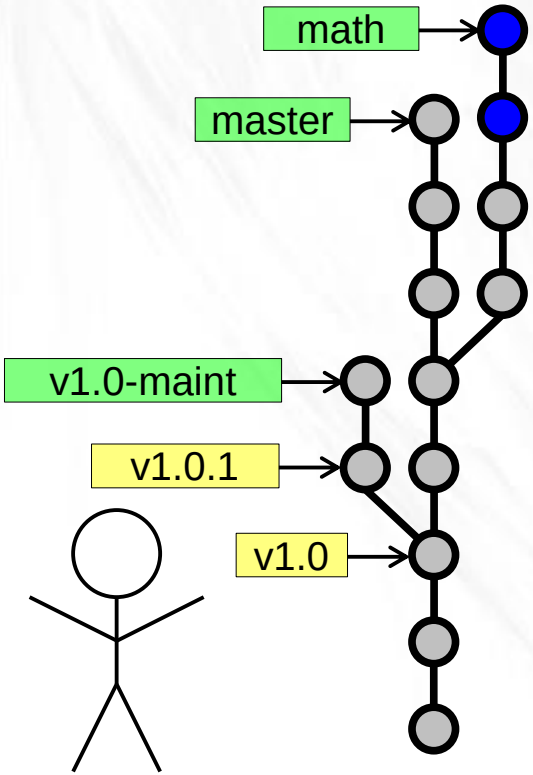


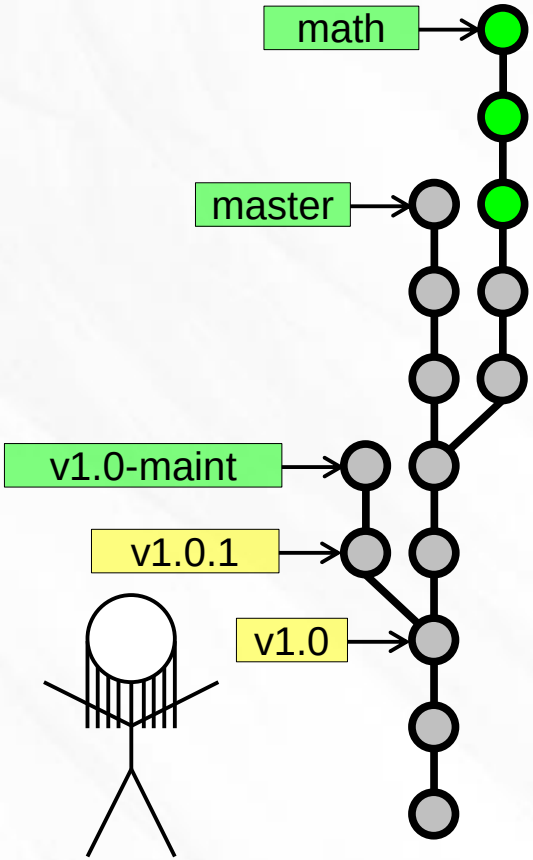
(simpler drawings)



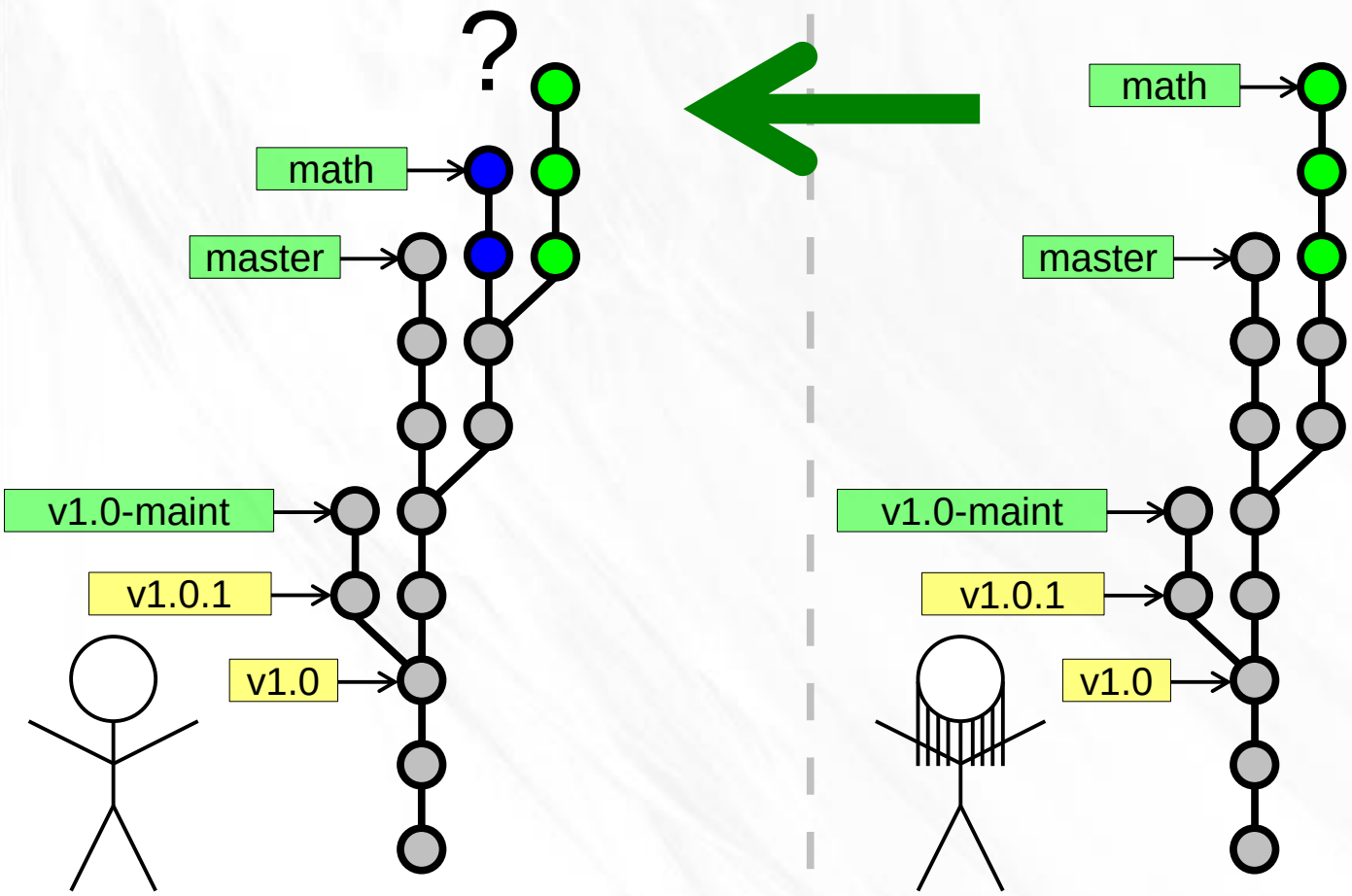


Merges

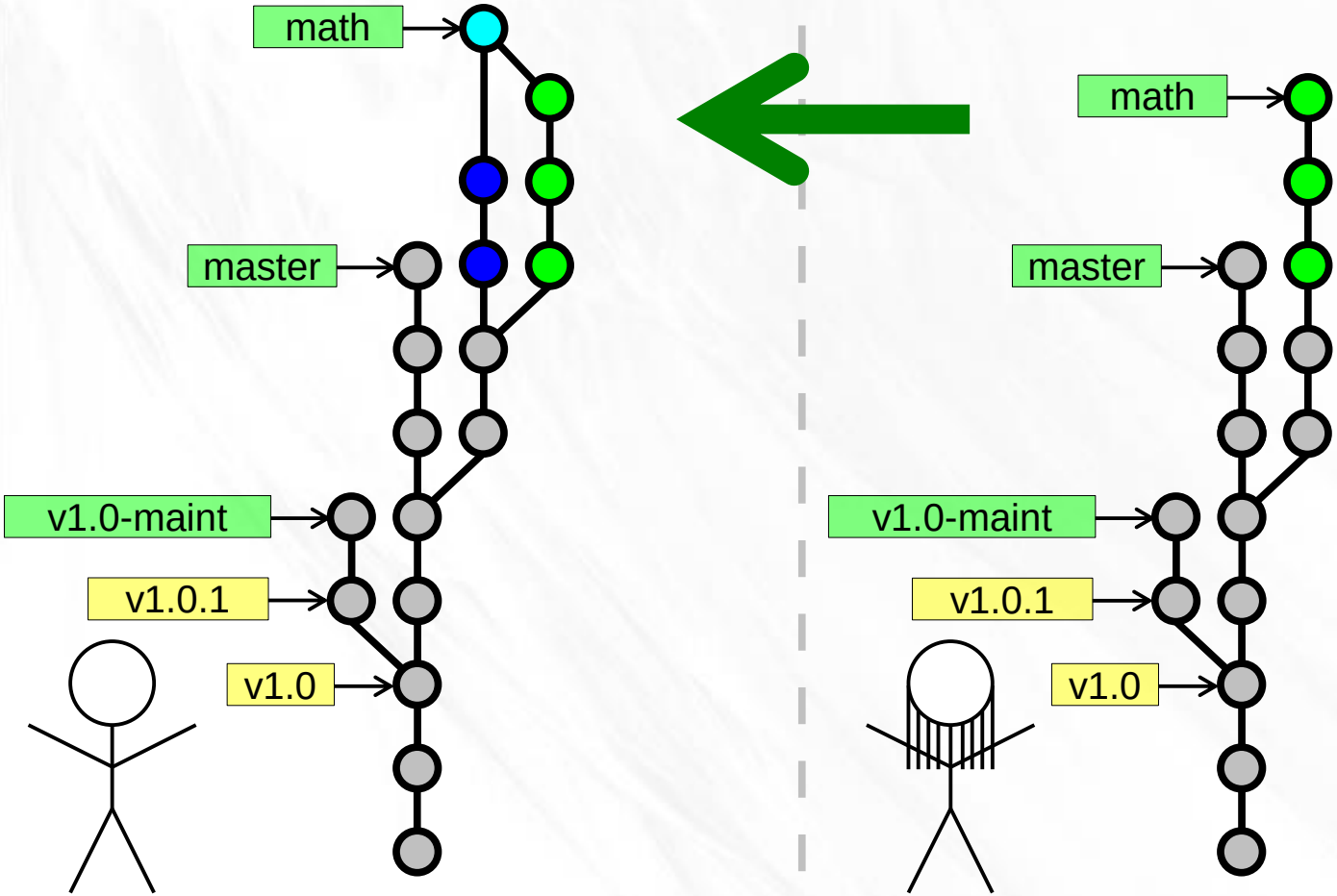

math
master
v1.0-maint
v1.0.1
v1.0


math
master
v1.0-maint
v1.0.1
v1.0

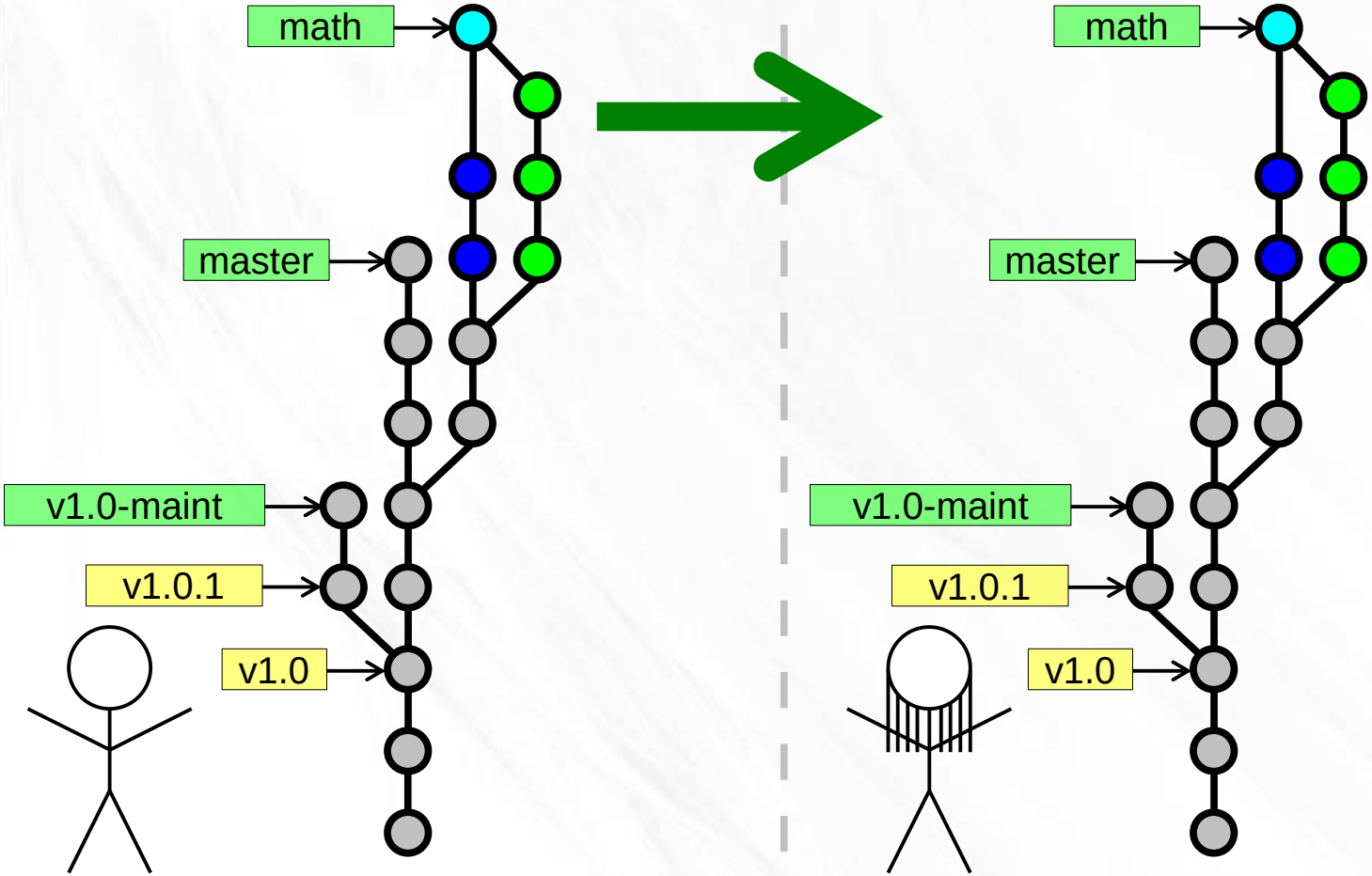
Merges


?
math
master
v1.0-maint
v1.0.1
v1.0
math
master
v1.0-maint
v1.0.1
v1.0

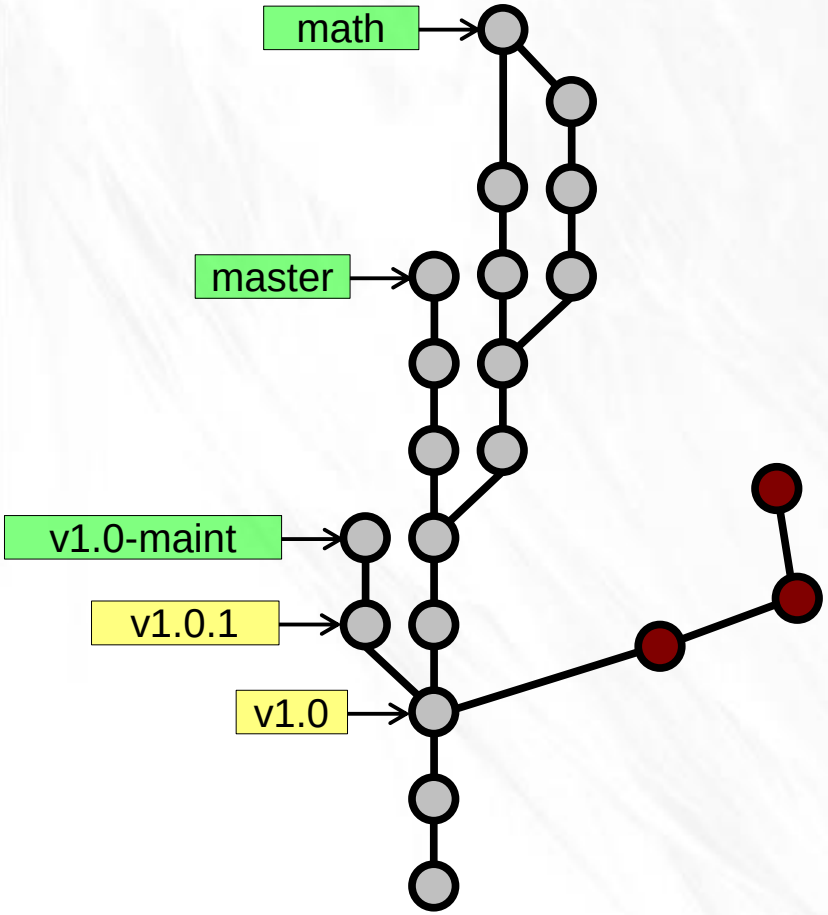
Merges

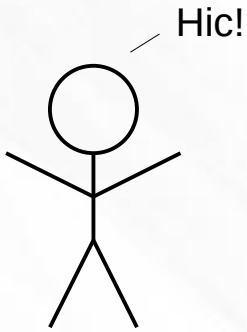

math
master
v1.0-maint
v1.0.1
v1.0
math
master
v1.0-maint
v1.0.1
v1.0

Merges

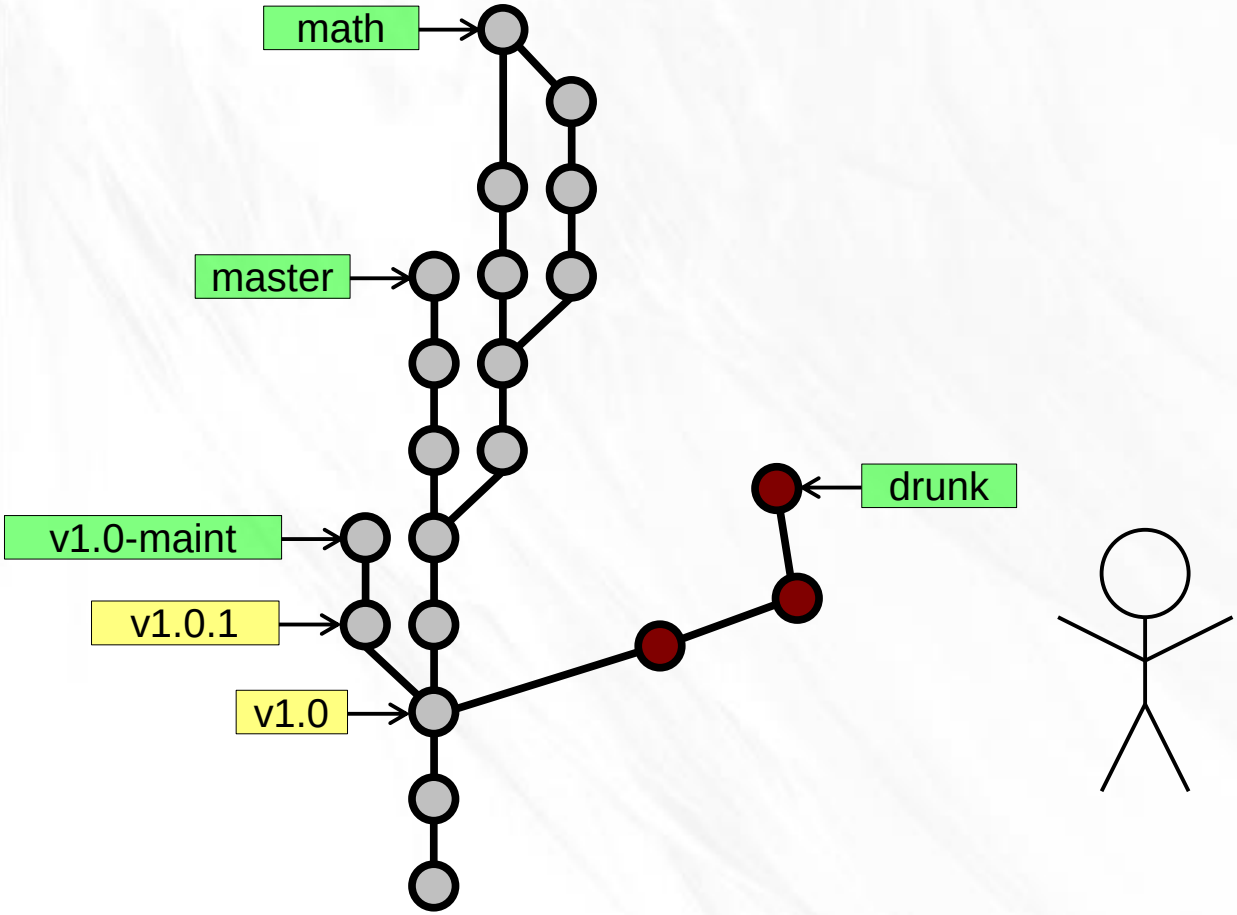

math
master
v1.0-maint
v1.0.1
v1.0
math
master
v1.0-maint
v1.0.1
v1.0

Rewriting History

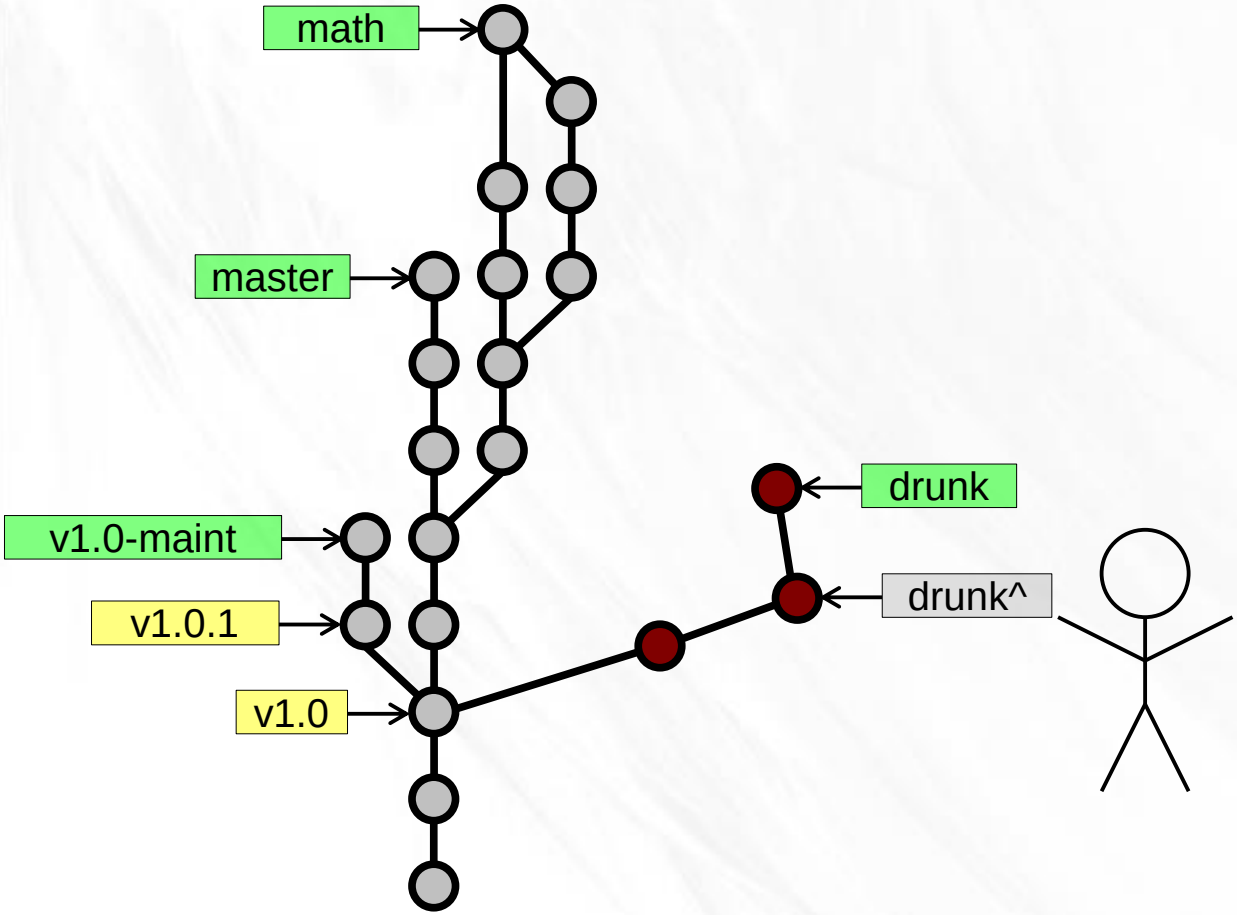




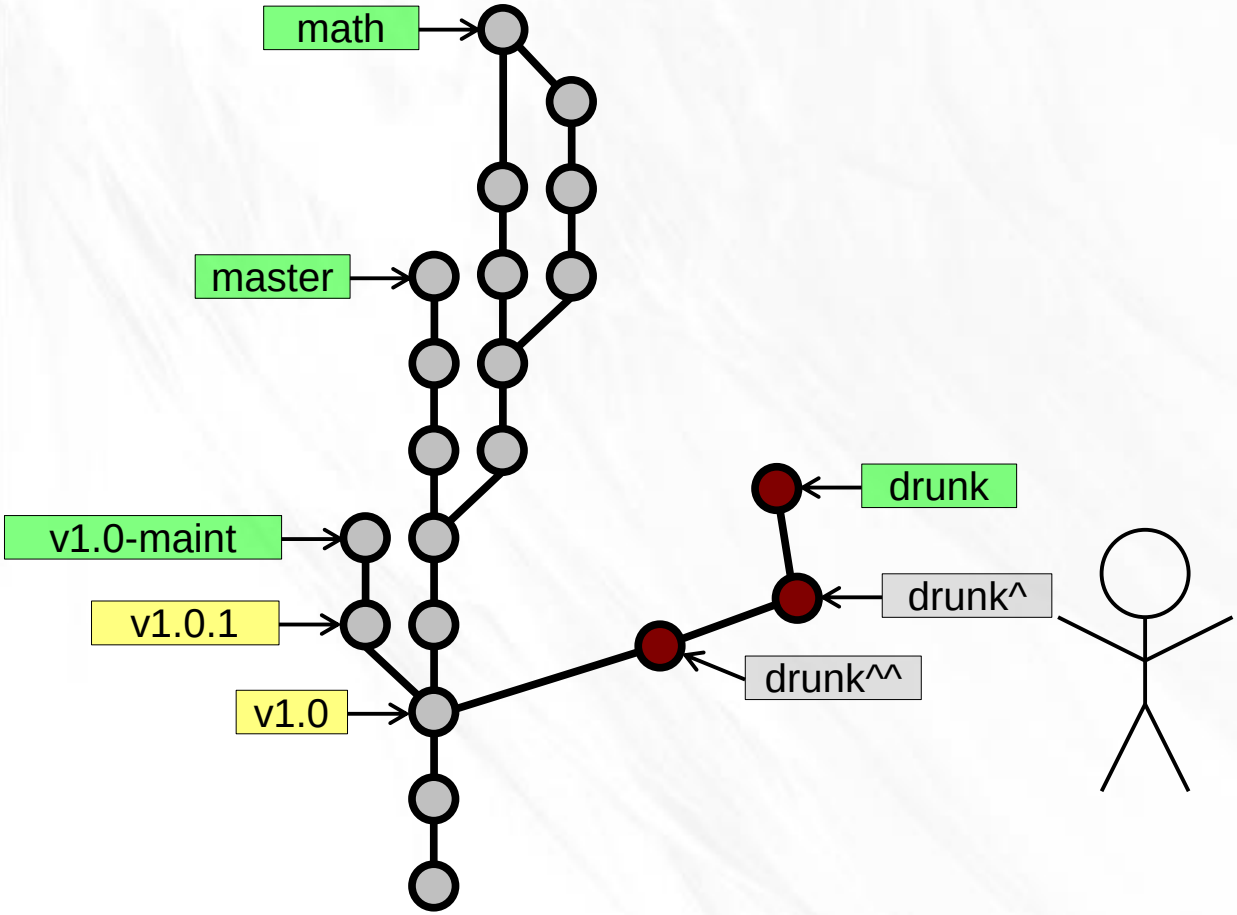
Rewriting History



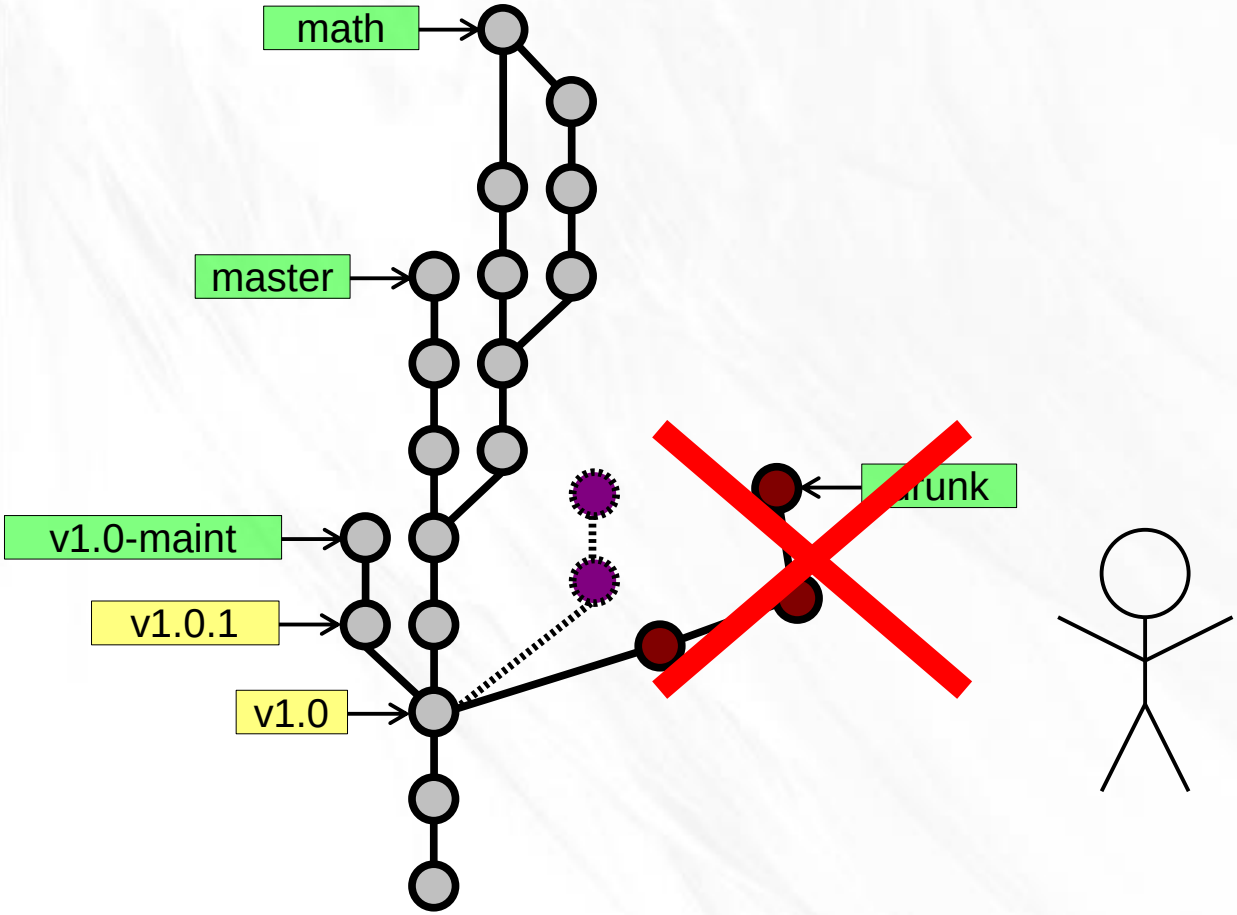
Rewriting History



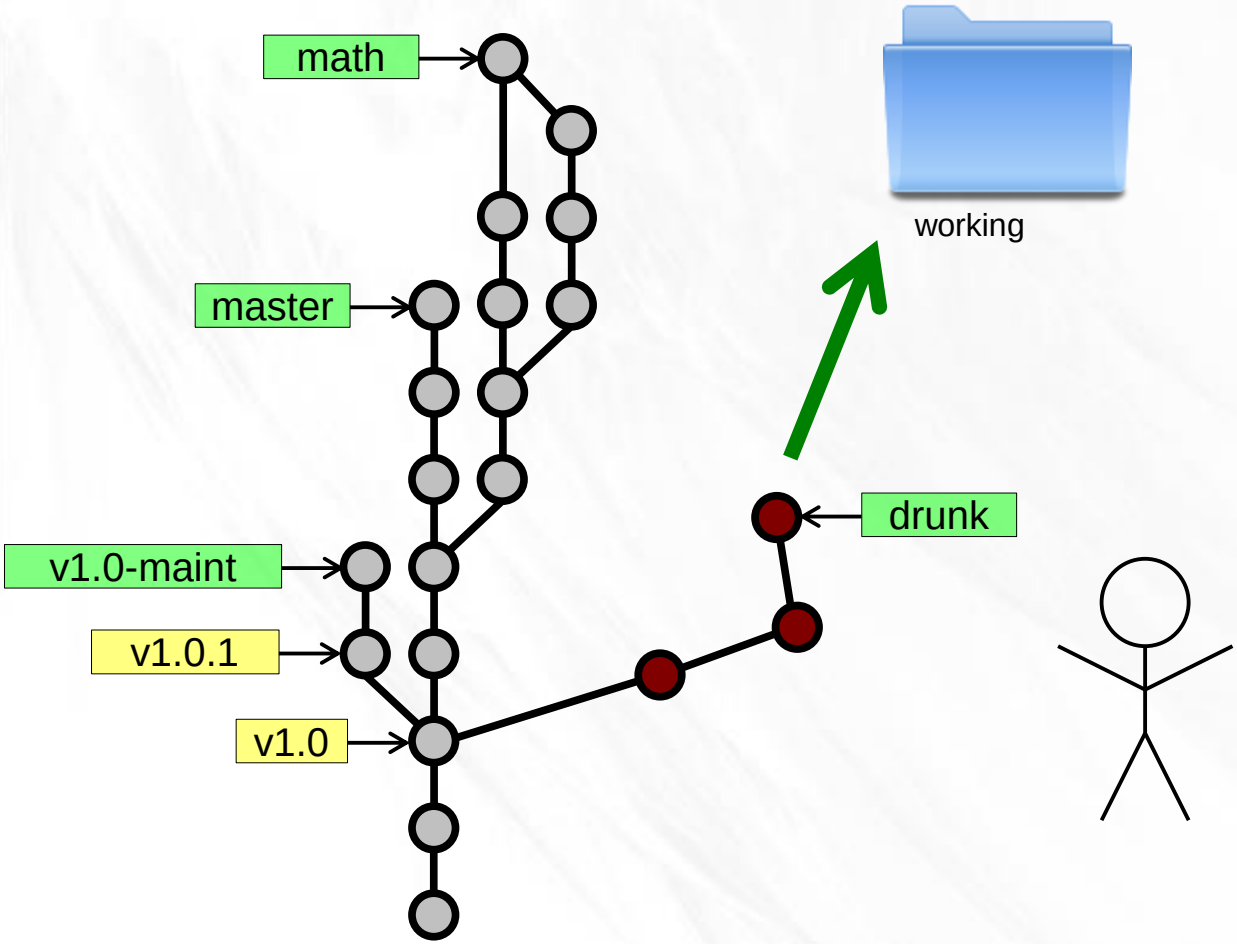
Rewriting History



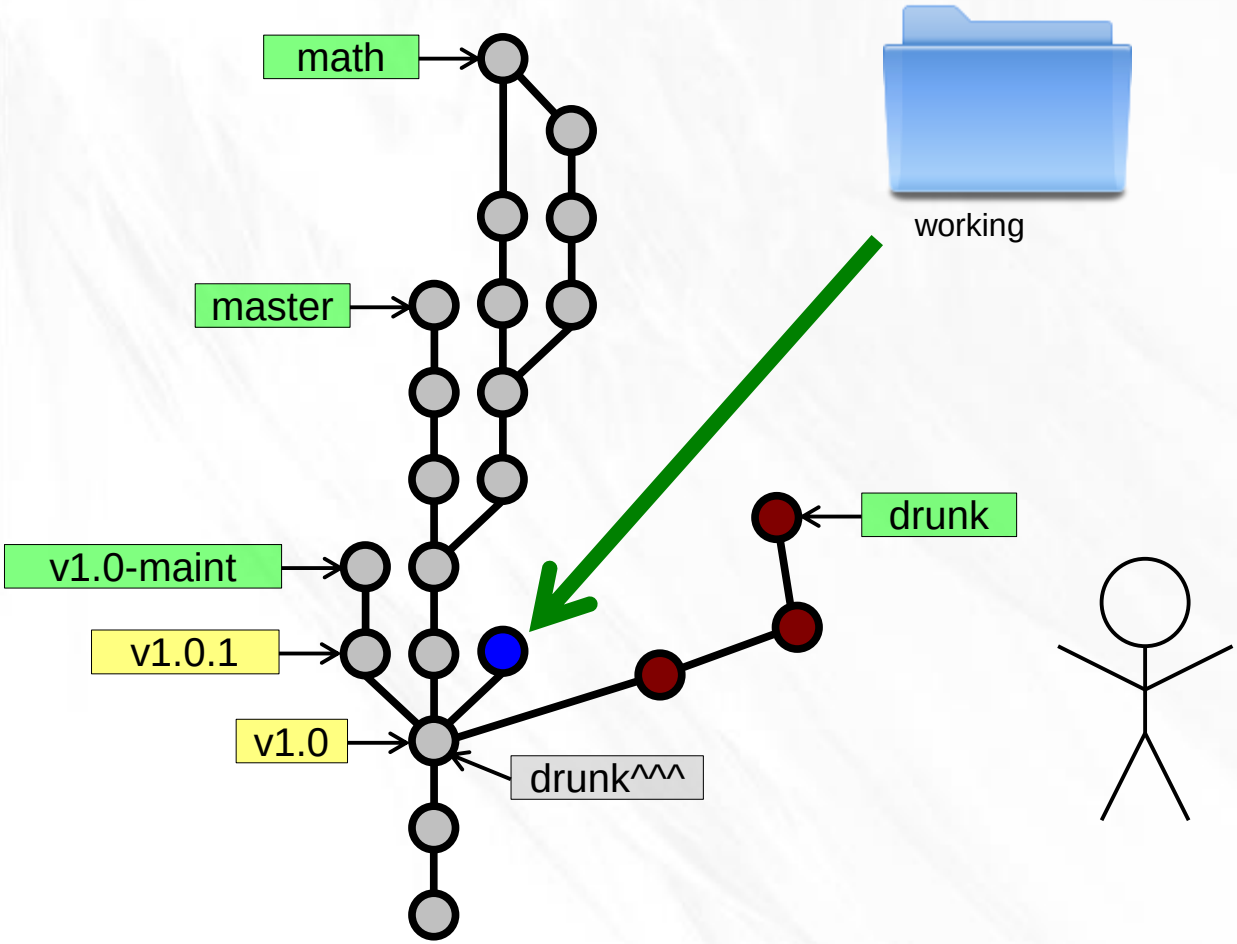
Rewriting History



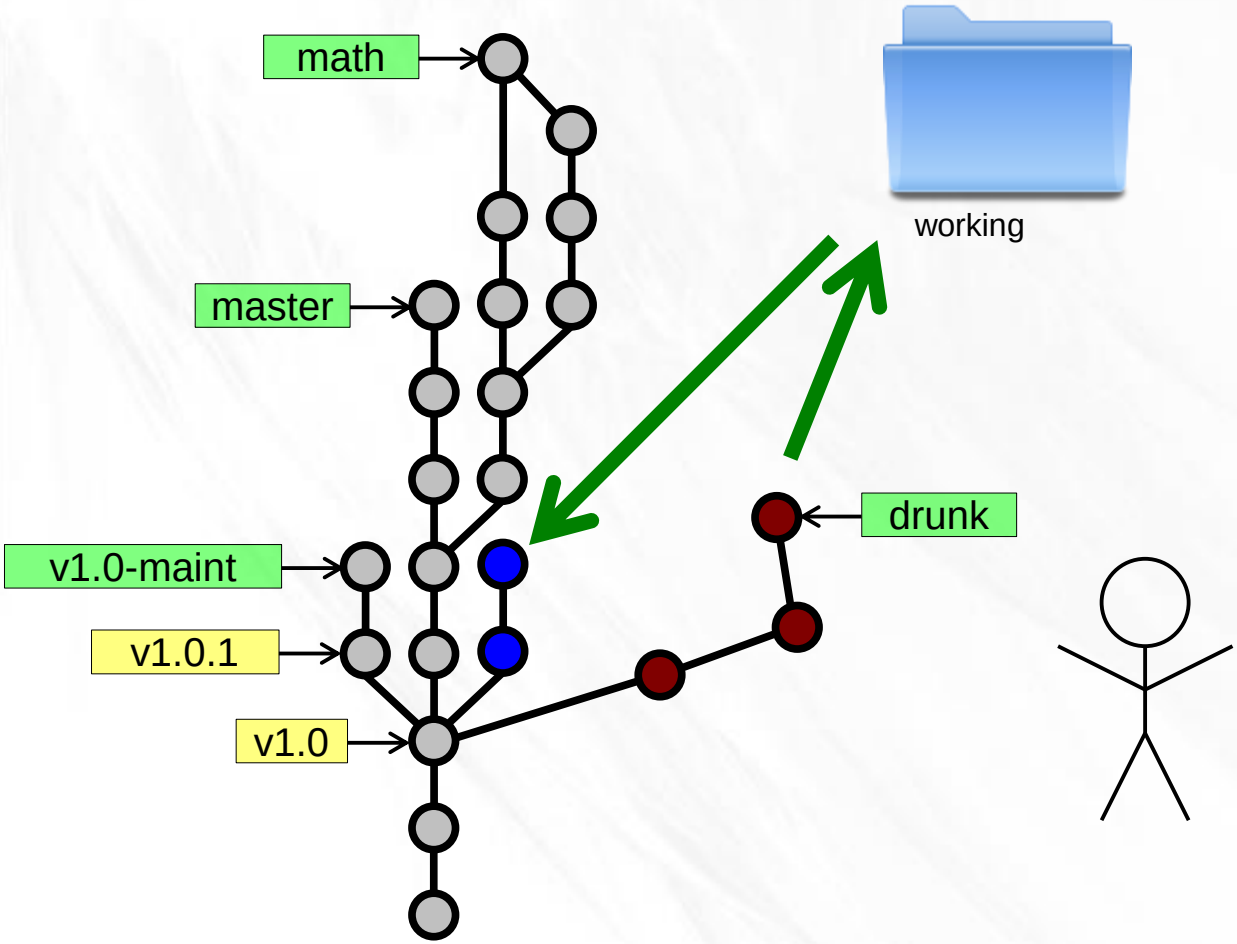
Rewriting History



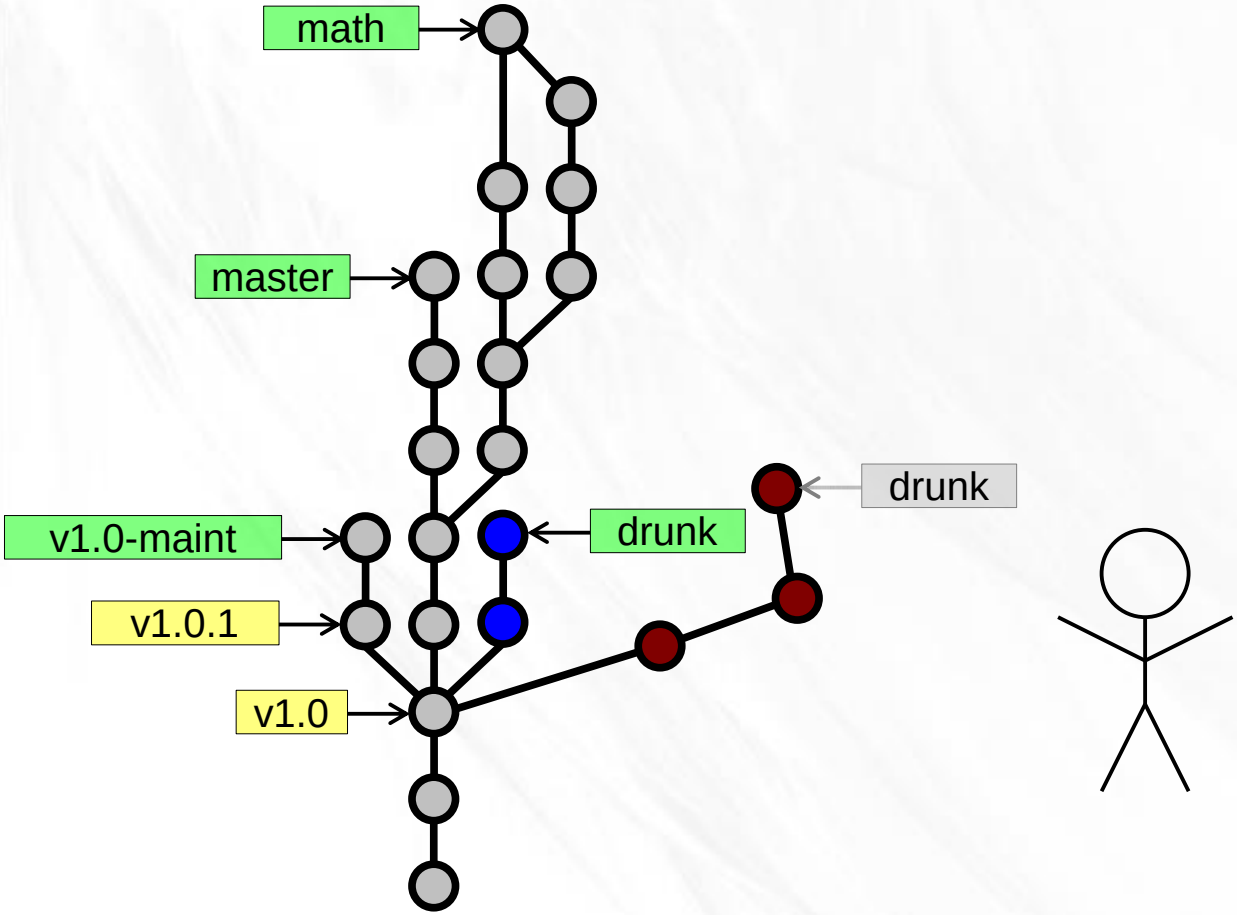
Rewriting History



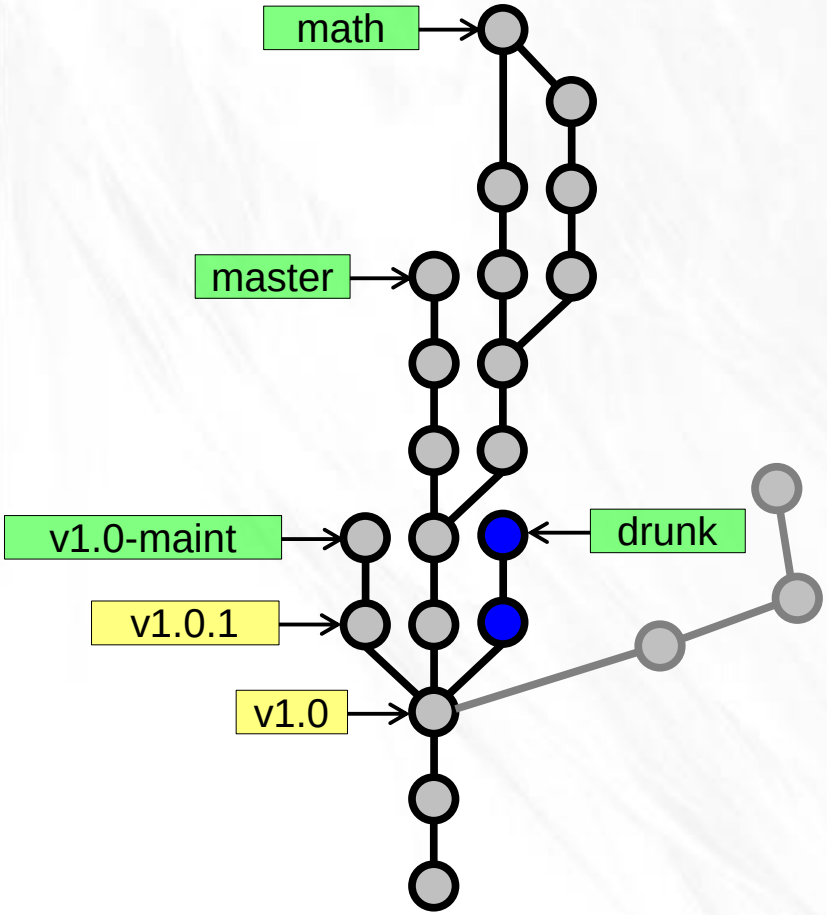
Rewriting History

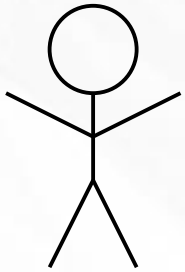


Rewriting History

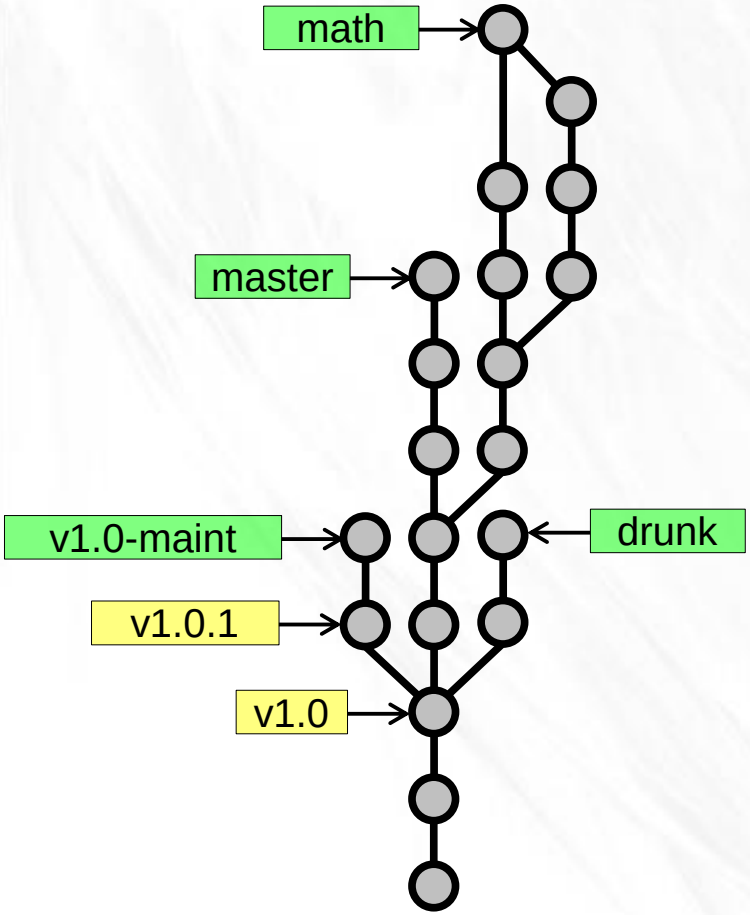


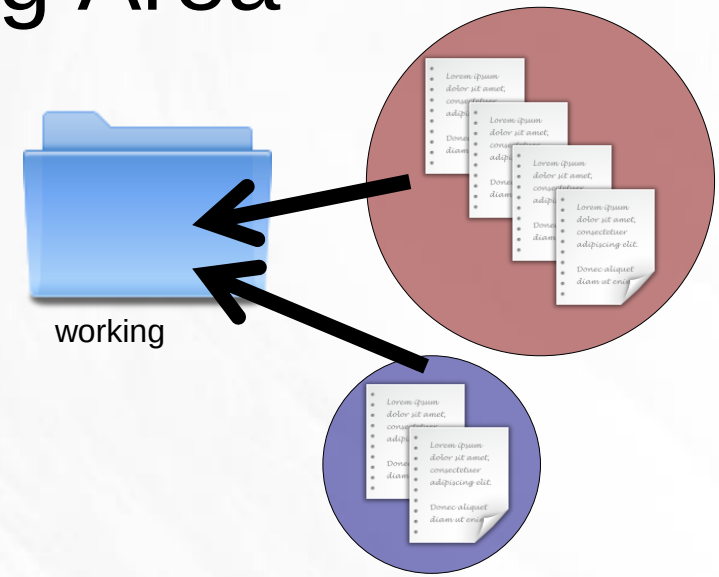
Rewriting History

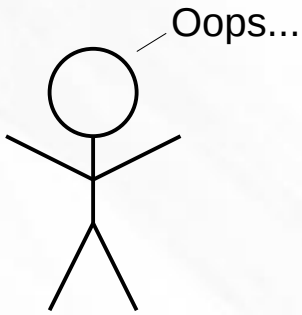




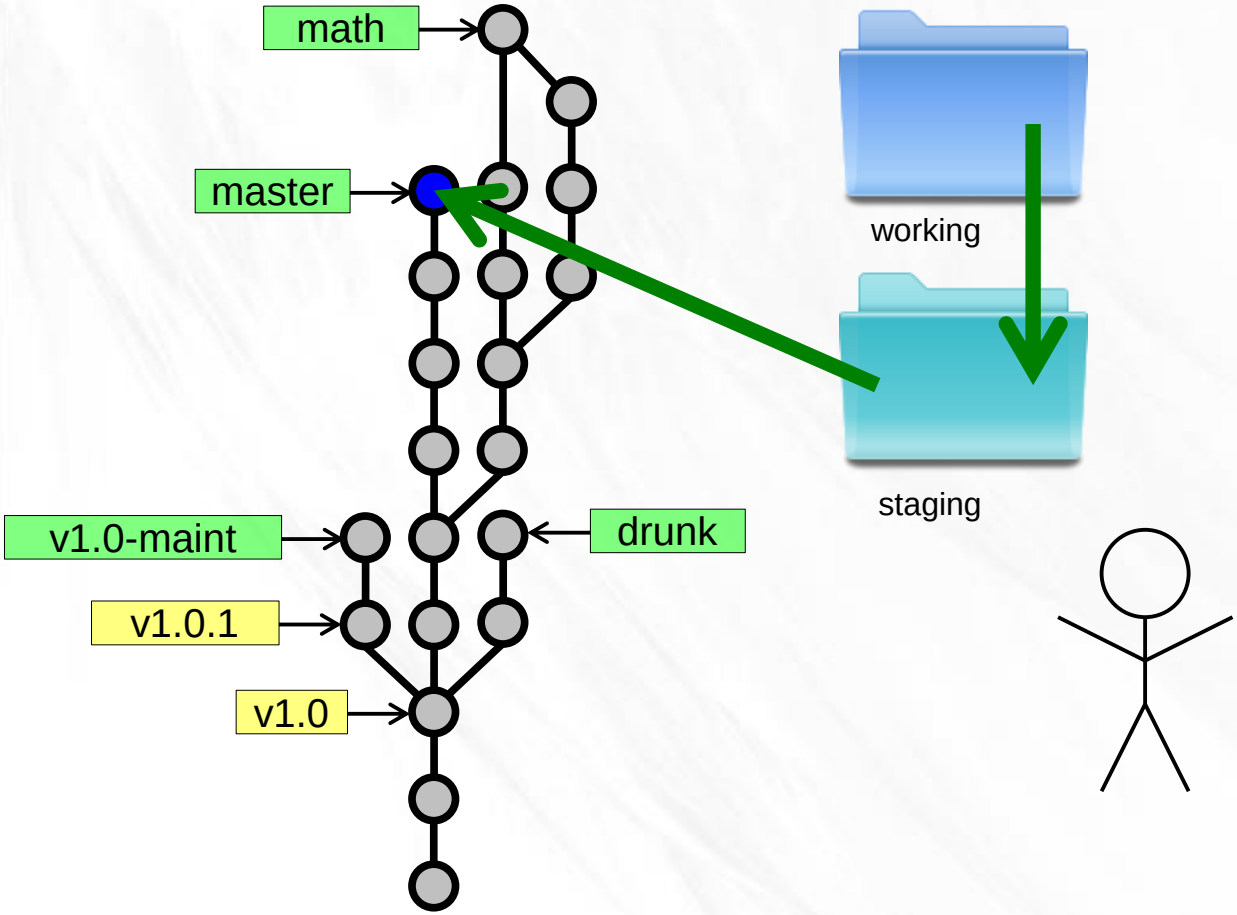
Staging Area



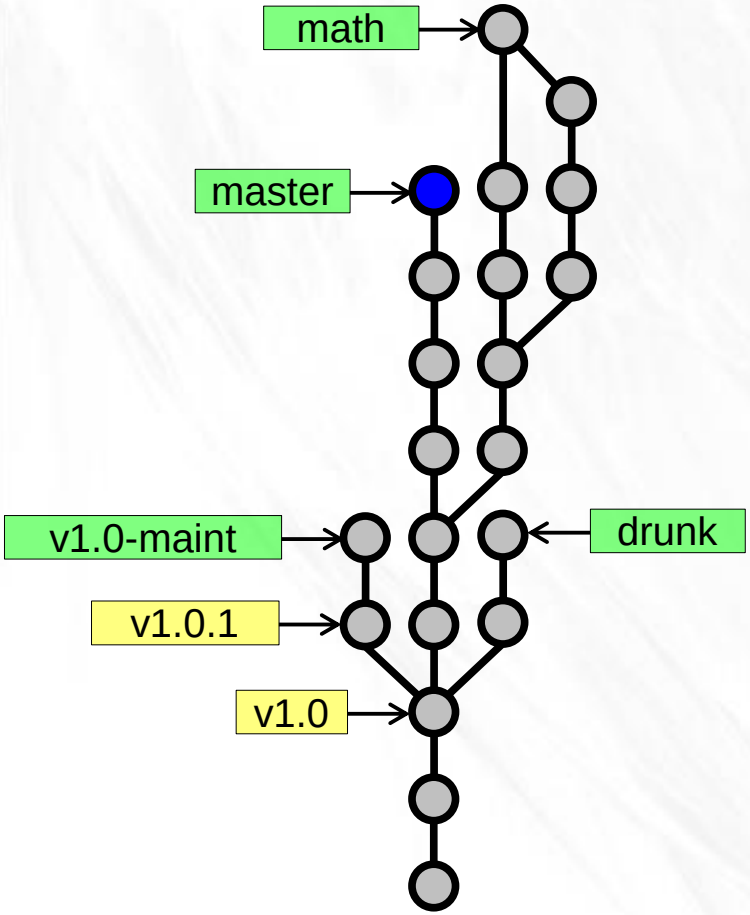


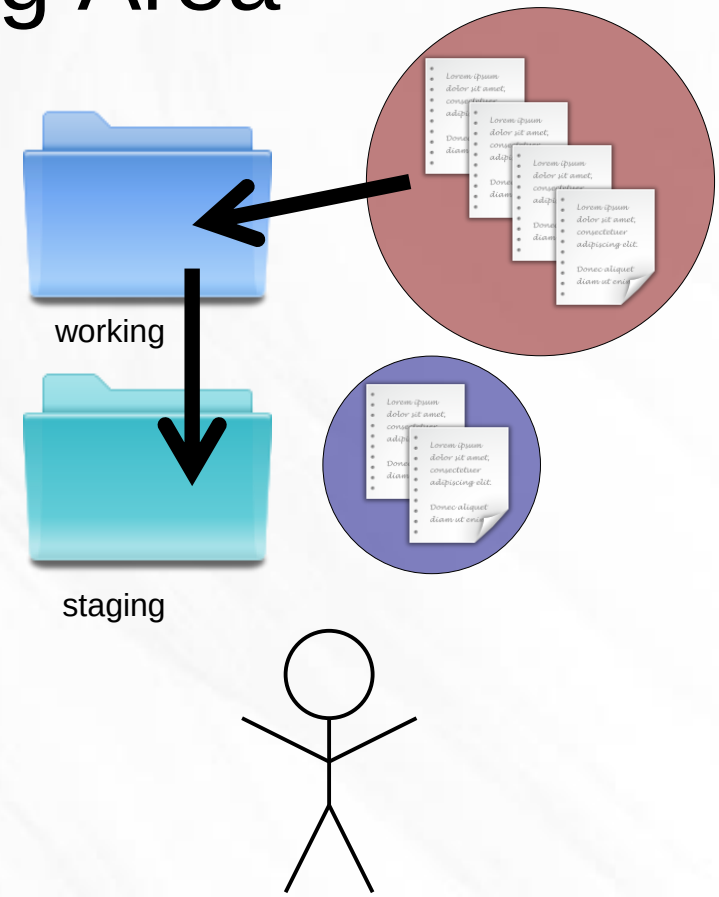


Staging Area

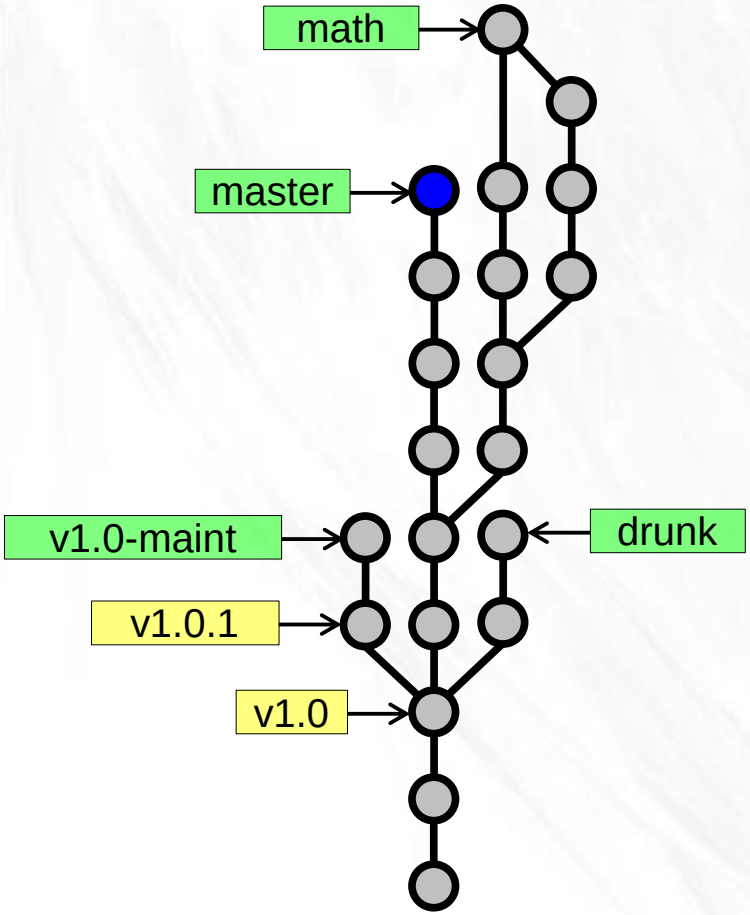


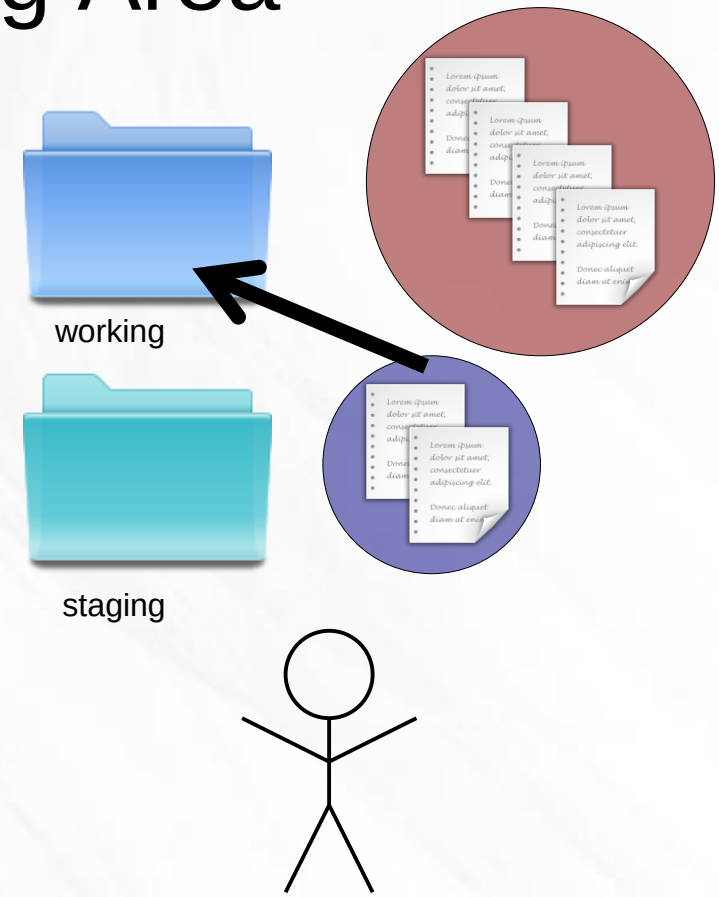
Staging Area



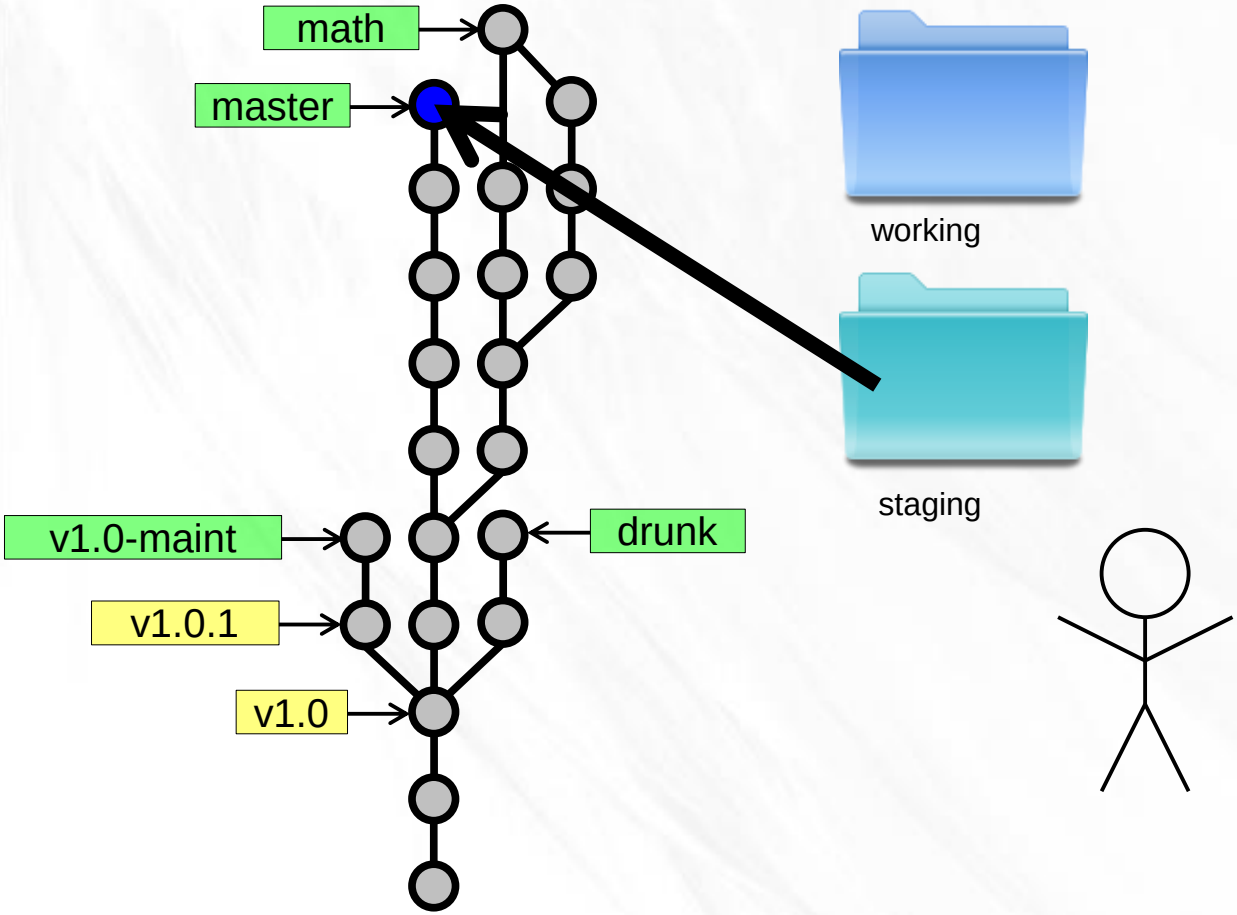


Staging Area

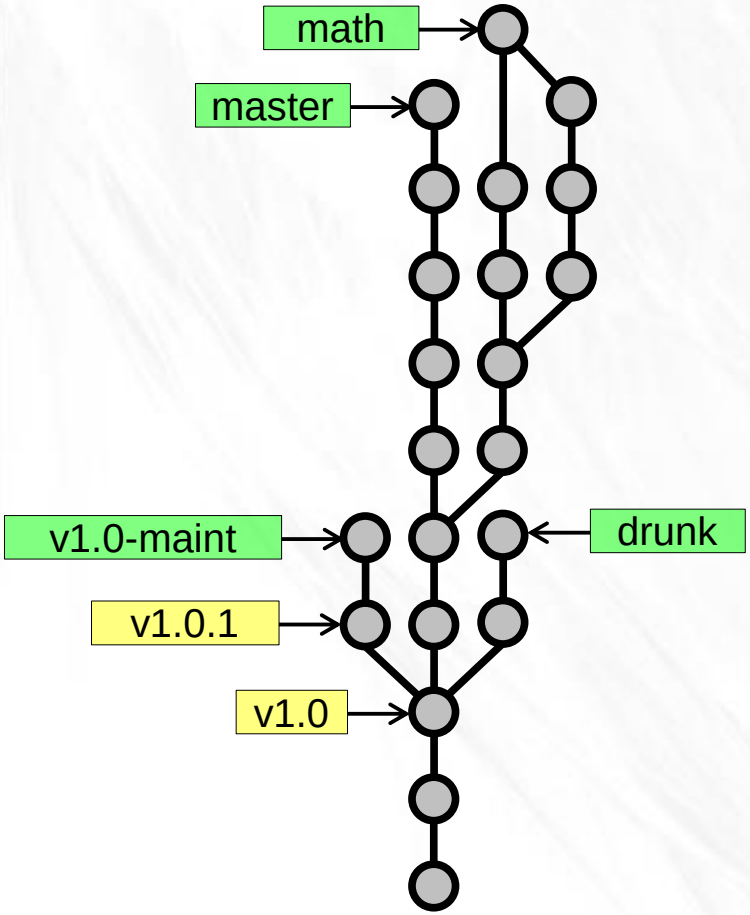


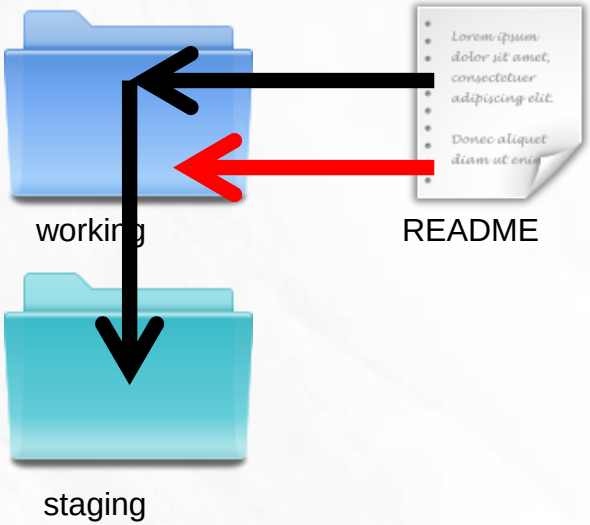


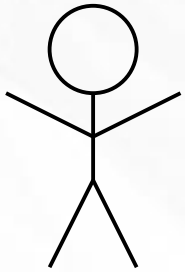
Staging Area



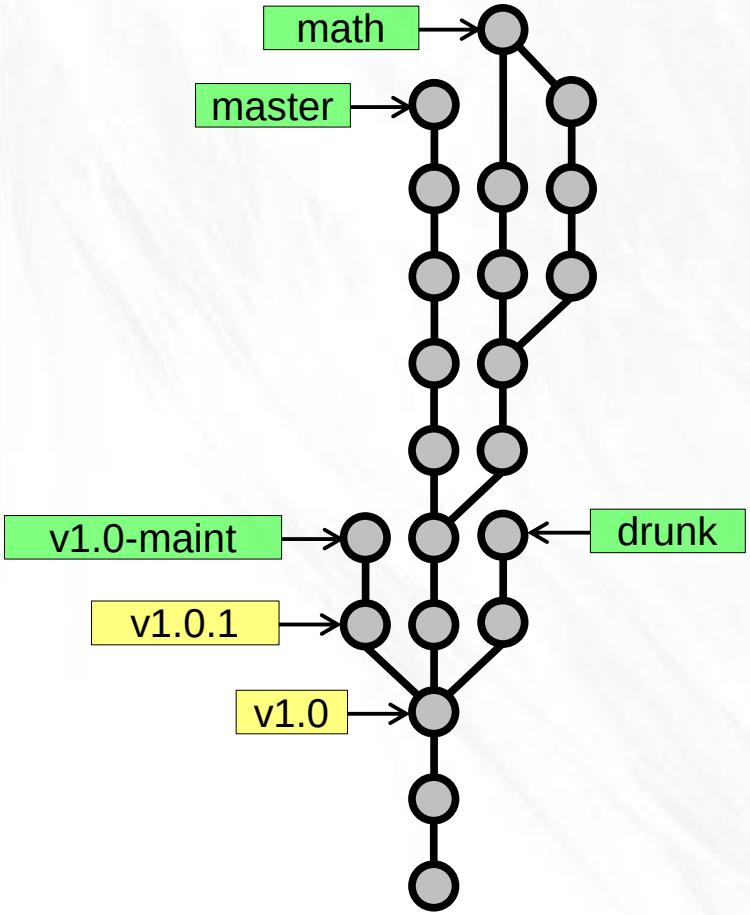
Staging Area

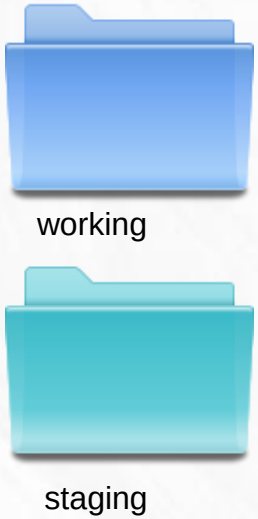




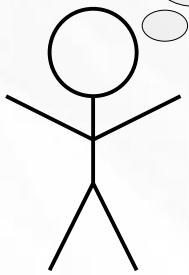


Diffs

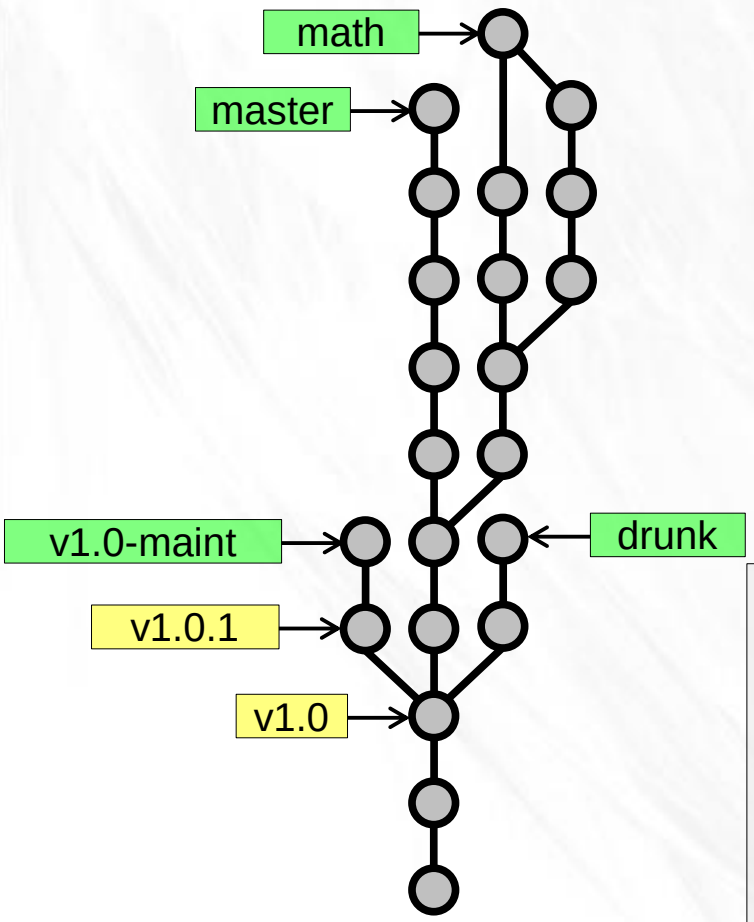


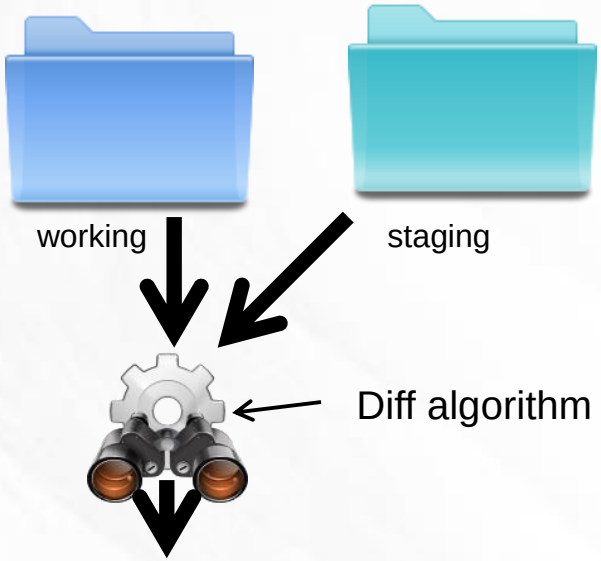


What are the changes?
- working vs. staging
- working vs. snapshot X
- staging vs. snapshot X
- snapshot X vs. snapshot Y



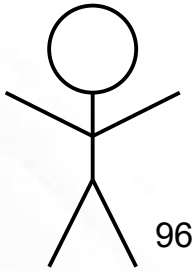
Diffs



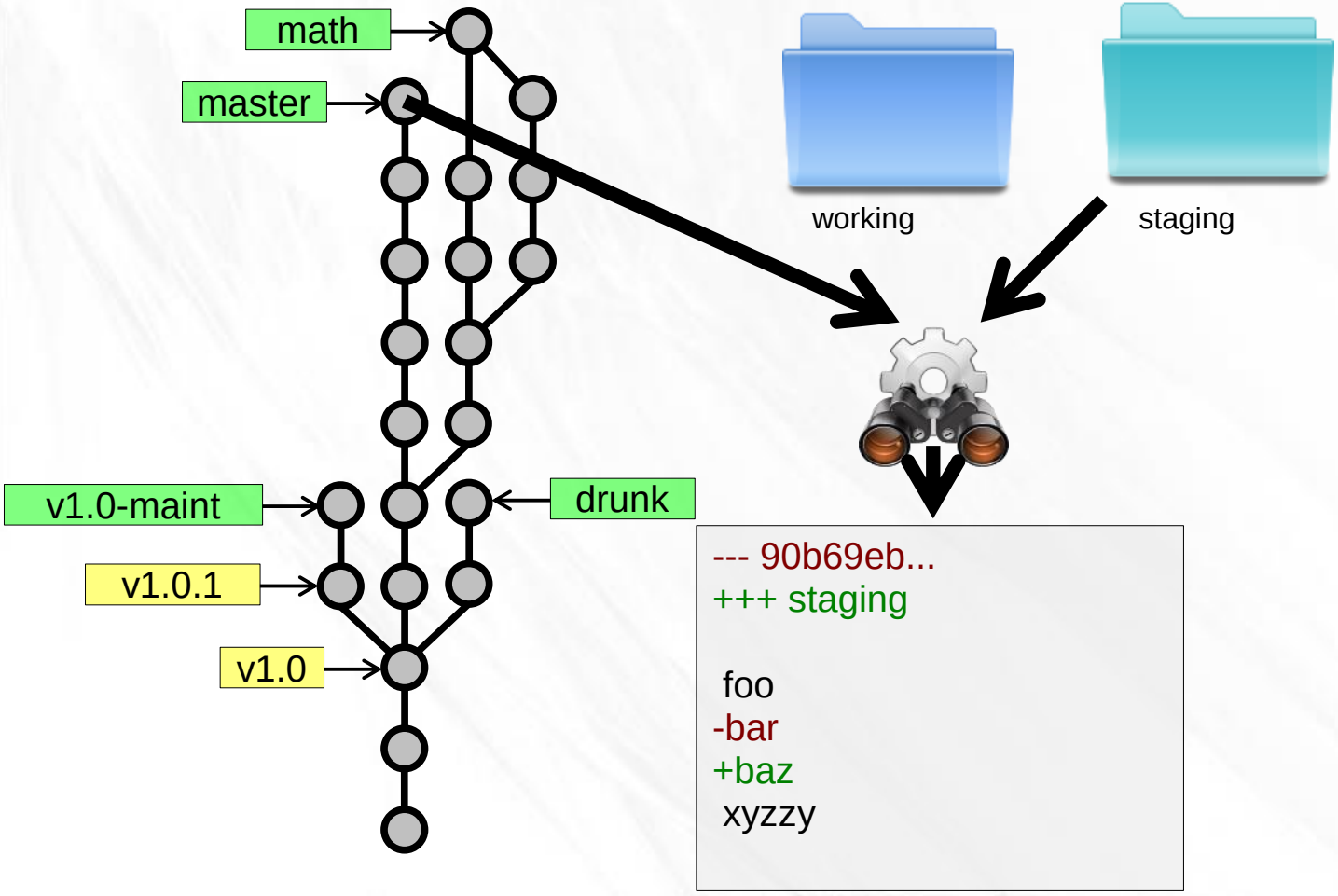


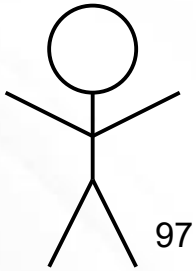

```
--- staging
+++ working

 foo
-baz
+bazzle
 xyzzy
```

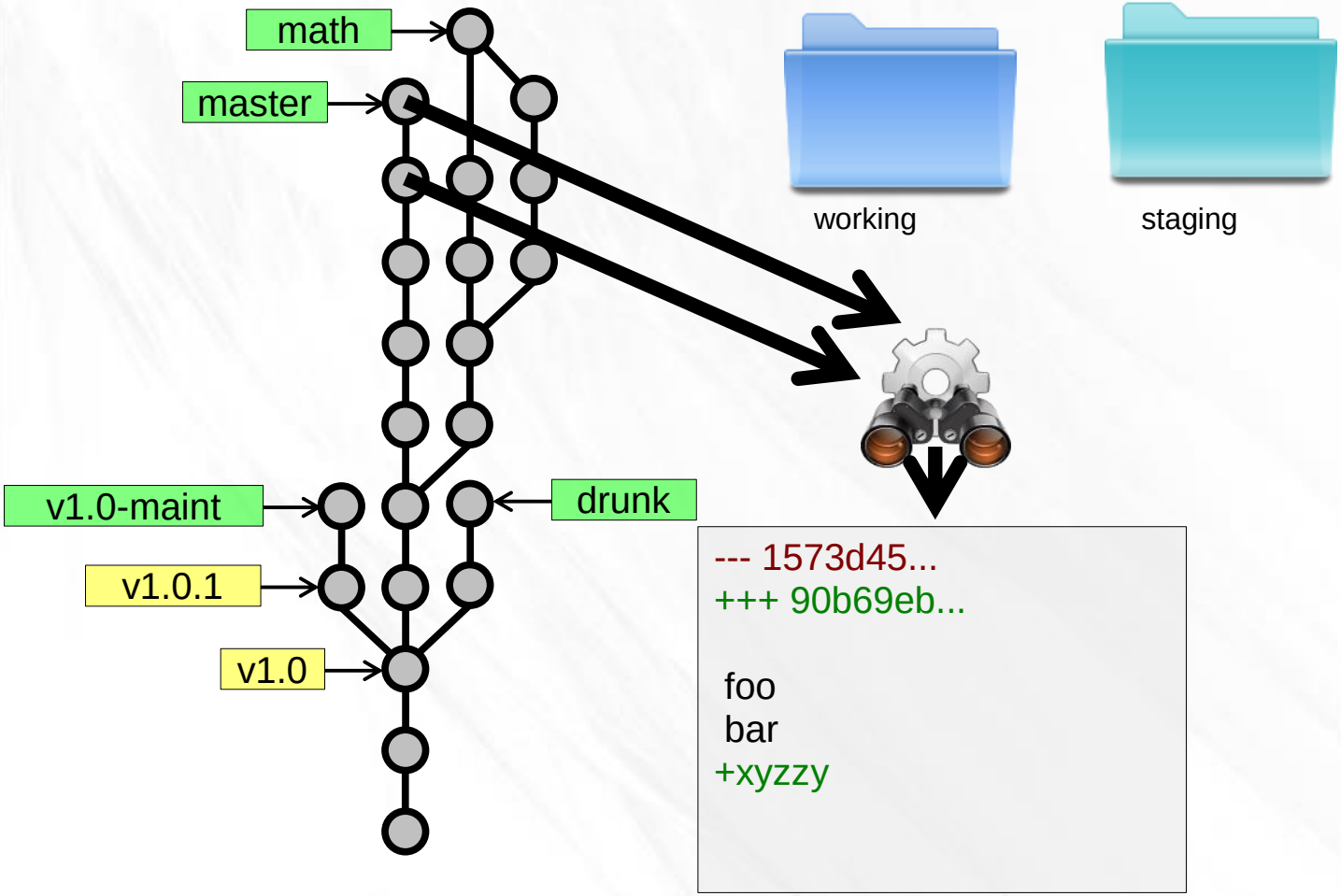


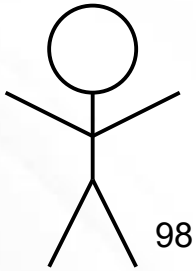
Diffs


math
master
v1.0-maint
drunk
v1.0.1
v1.0
working
staging
--- 90b69eb...
+++ staging
foo
-bar
+baz
xyzzy

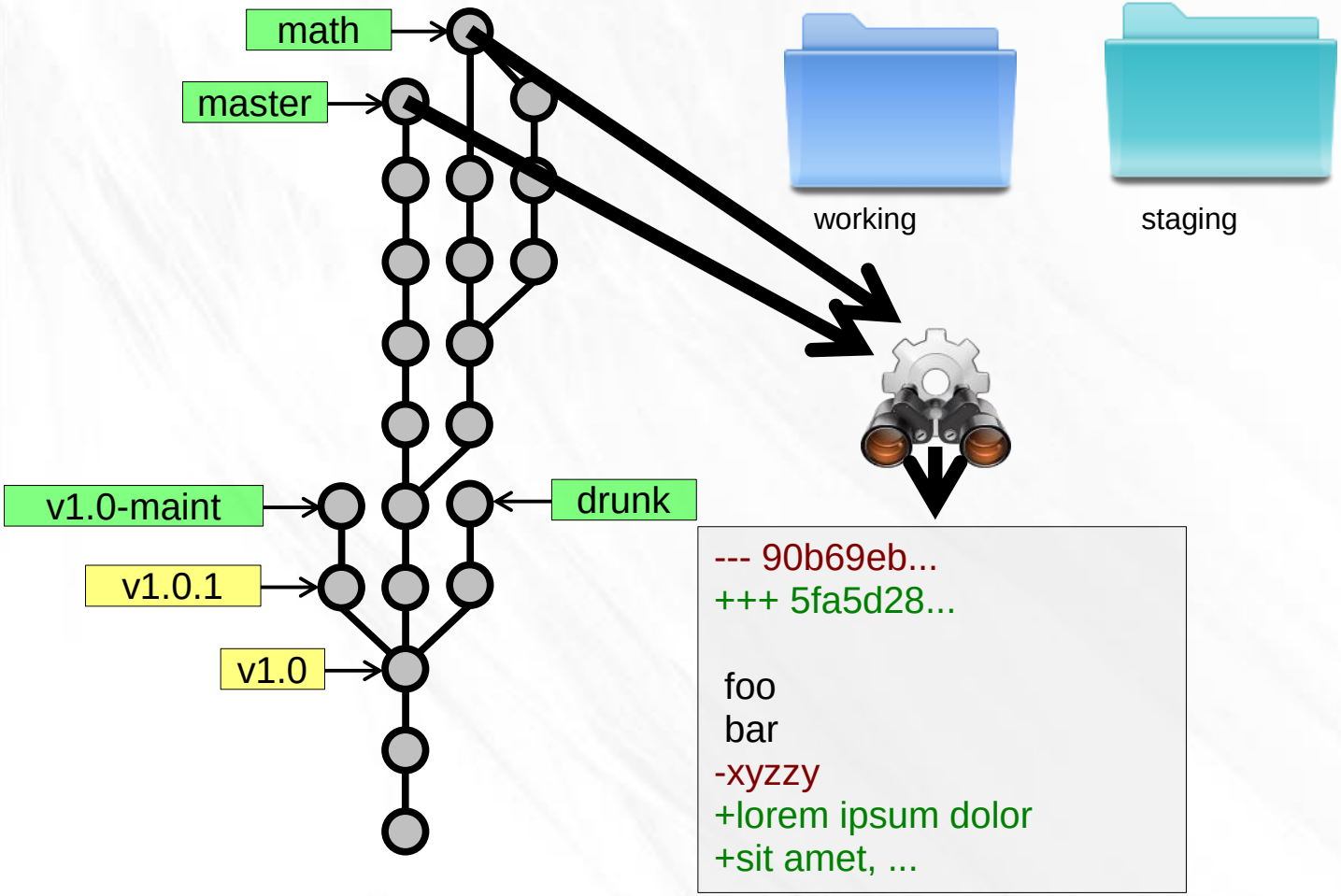


Diffs

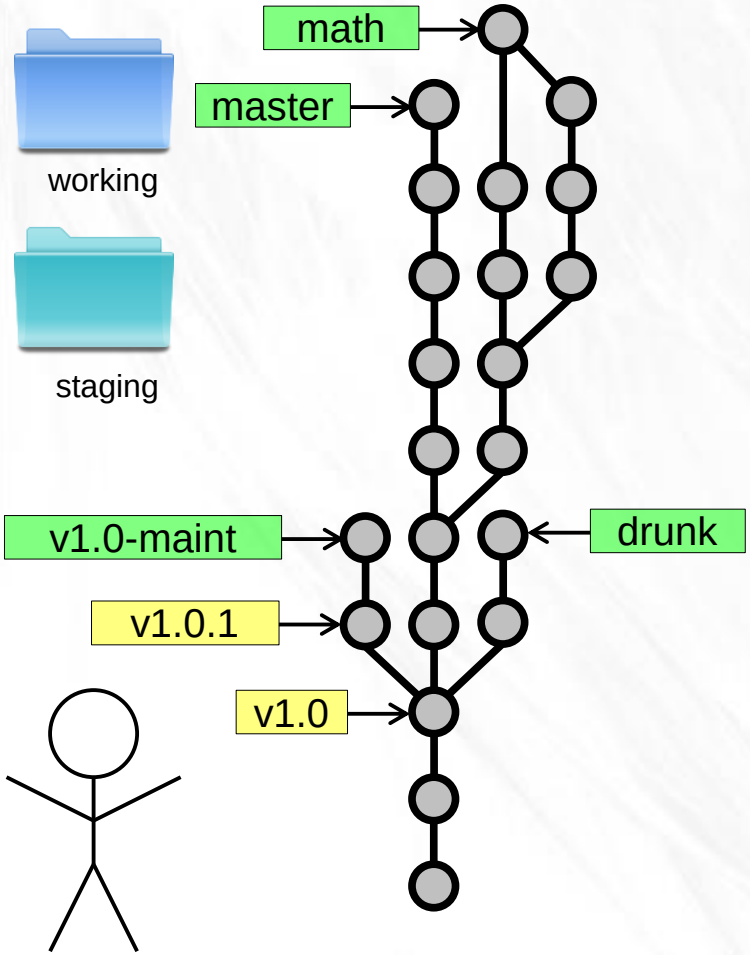

math
master
v1.0-maint
drunk
v1.0.1
v1.0
working
staging
--- 1573d45...
+++ 90b69eb...
foo
bar
+xyzzy

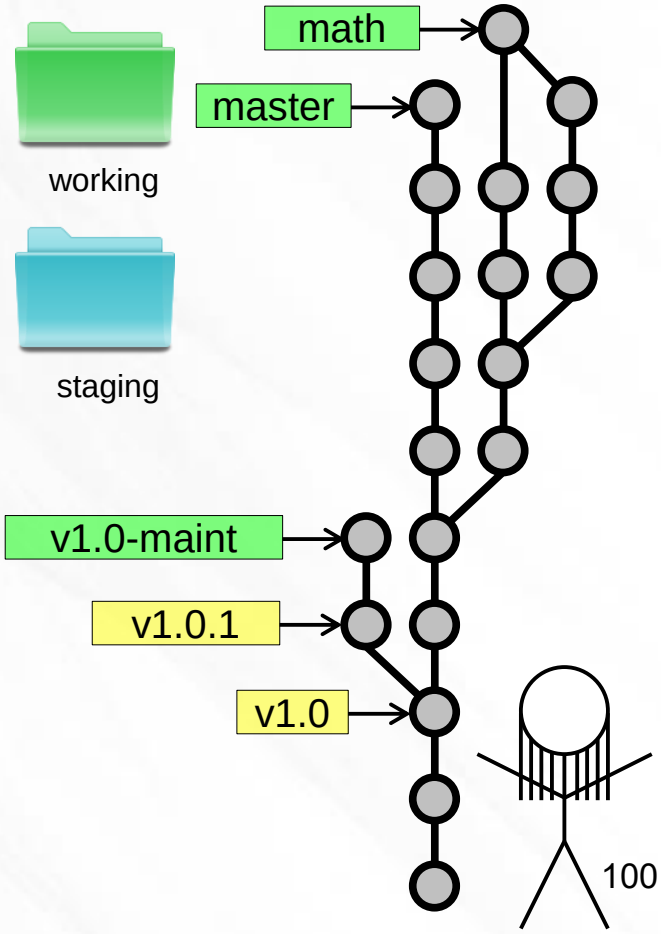


Diffs

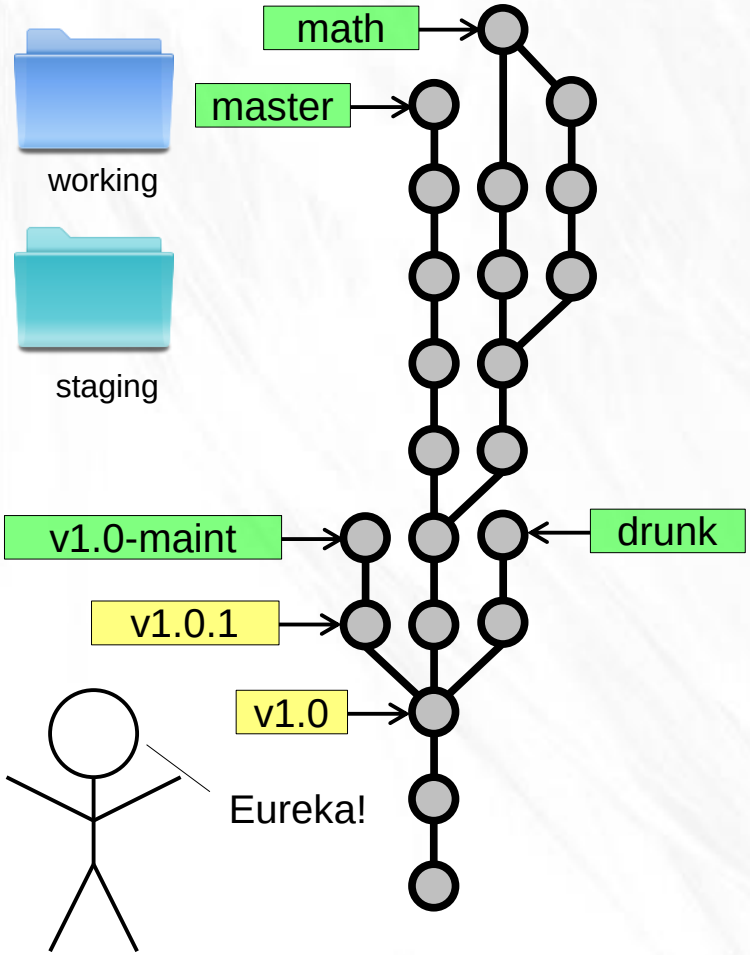

math
master
v1.0-maint
v1.0.1
v1.0
drunk
working
staging
--- 90b69eb...
+++ 5fa5d28...
foo
bar
-xyzzy
+lorem ipsum dolor
+sit amet, ...

Eliminating Duplication





Eliminating Duplication

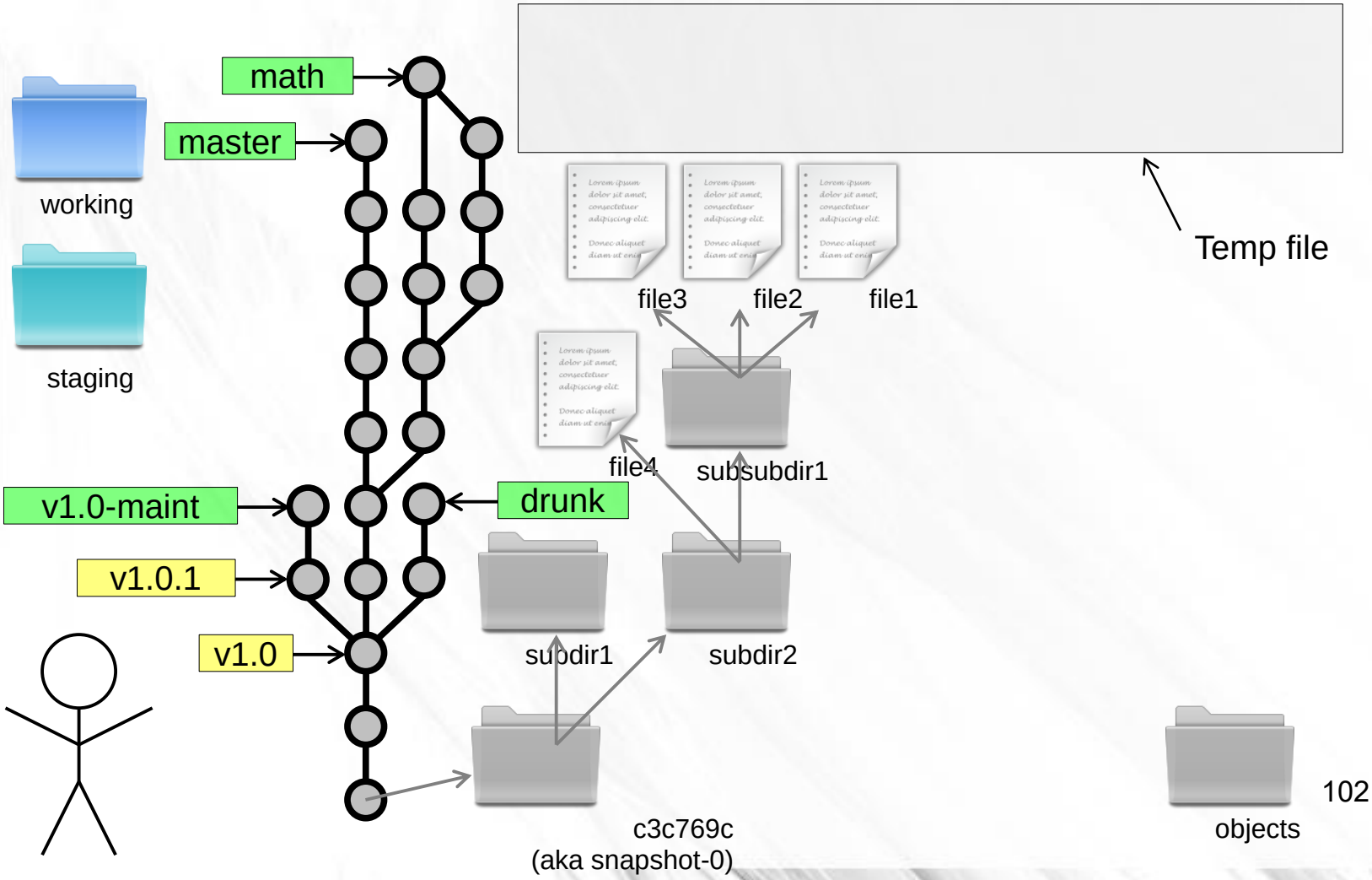


SHA1

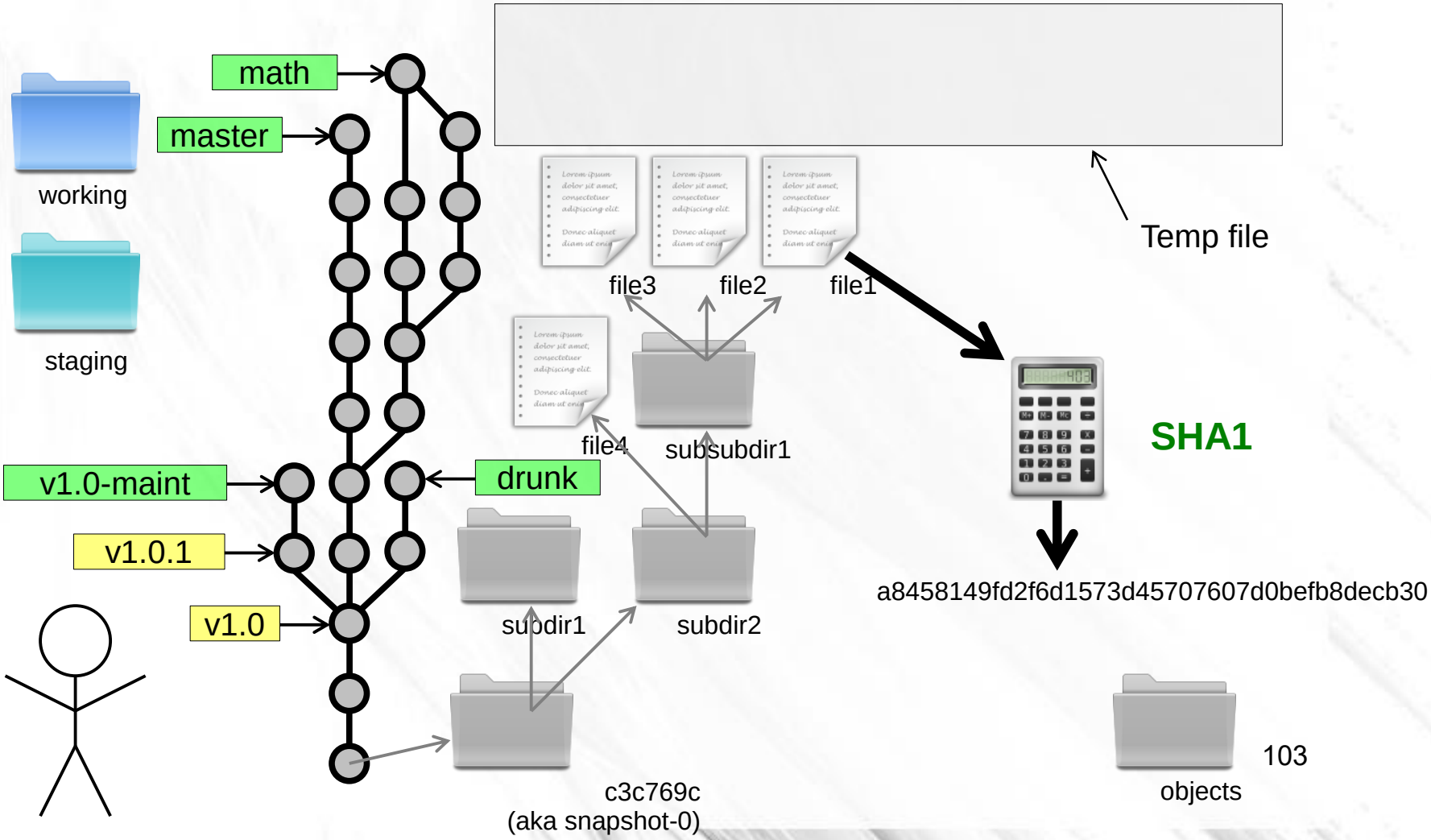
2804133755c3ed396d162028c7b30a1cbcfecded

objects

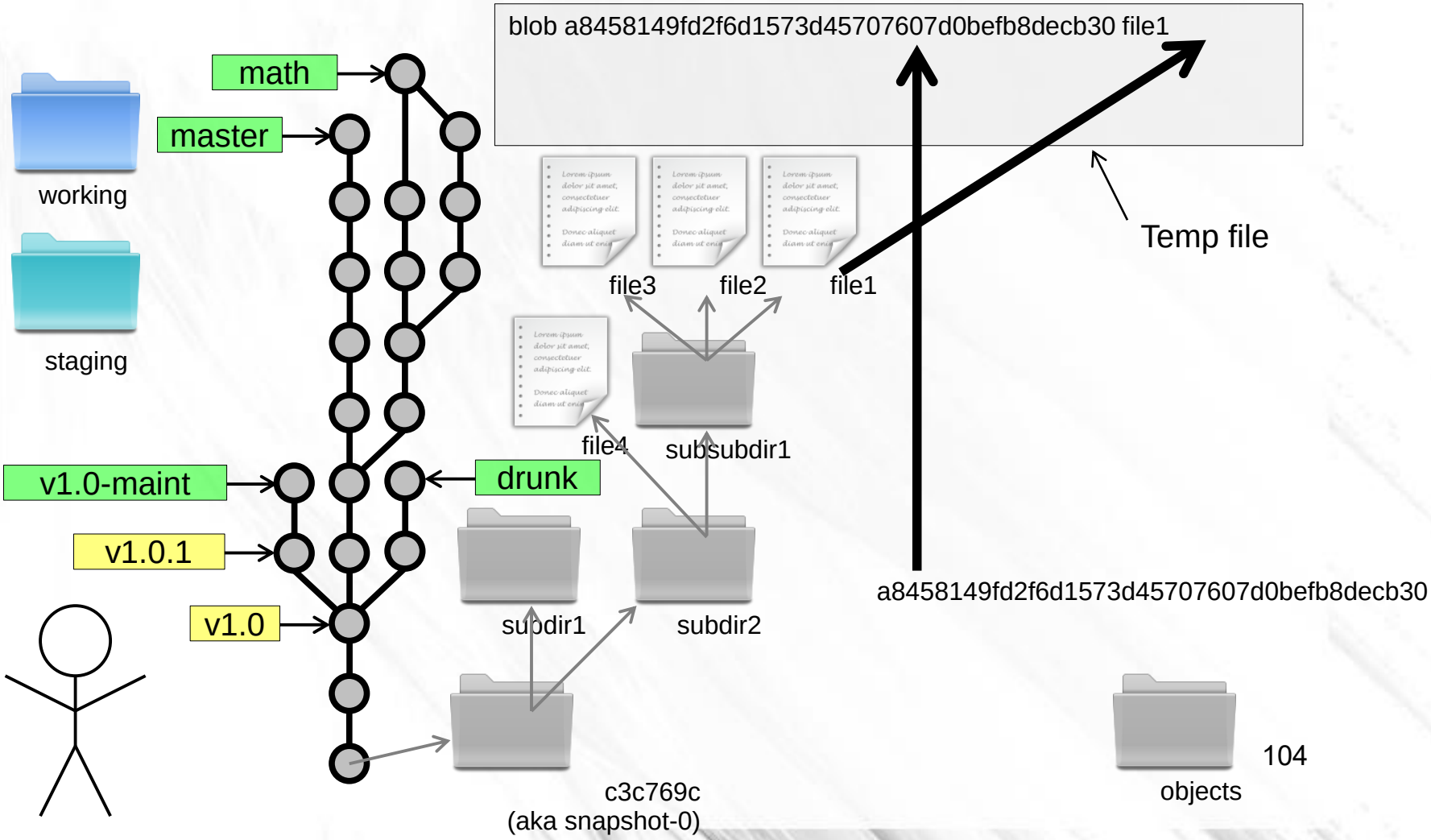
Eliminating Duplication



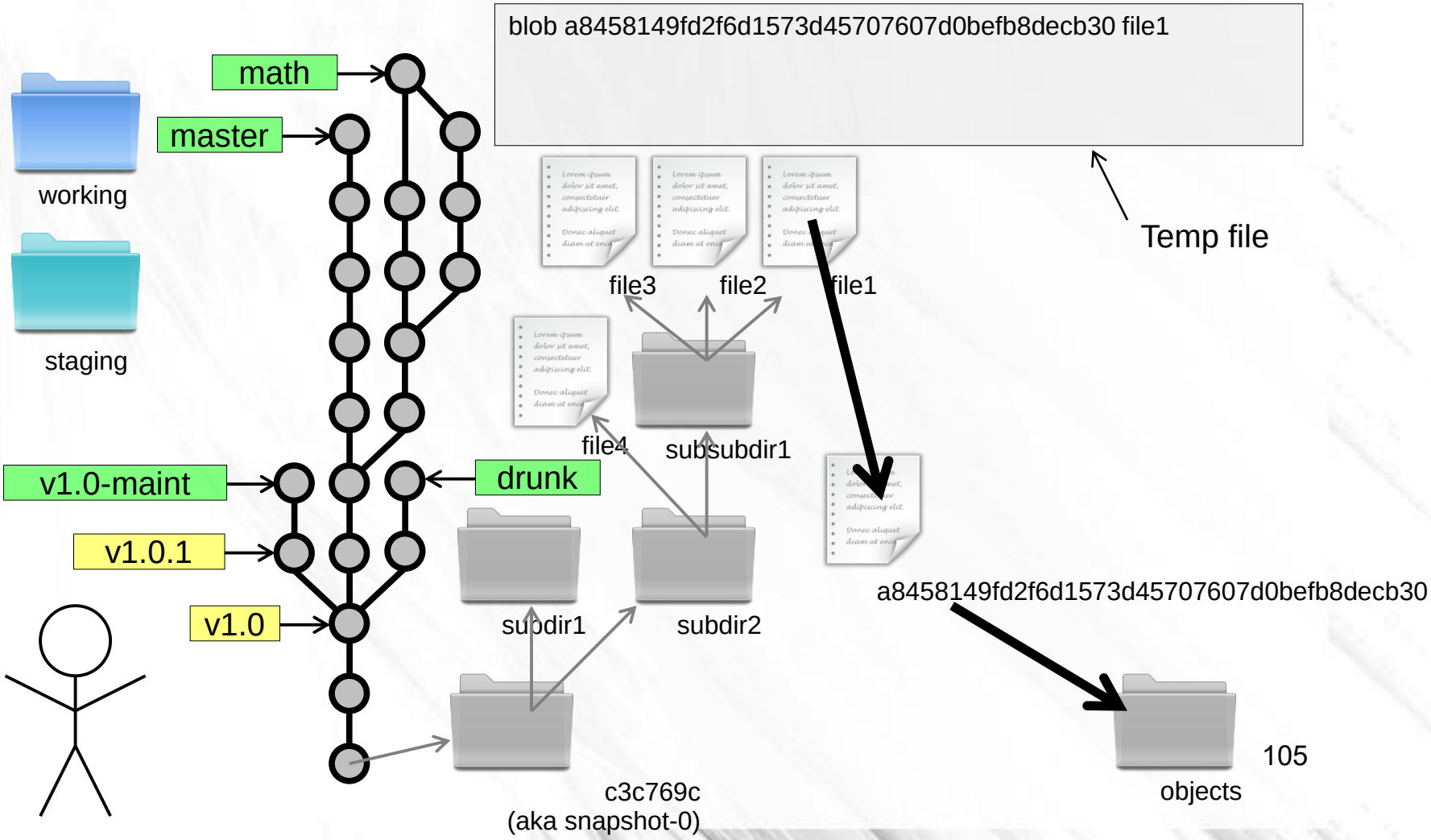
Eliminating Duplication



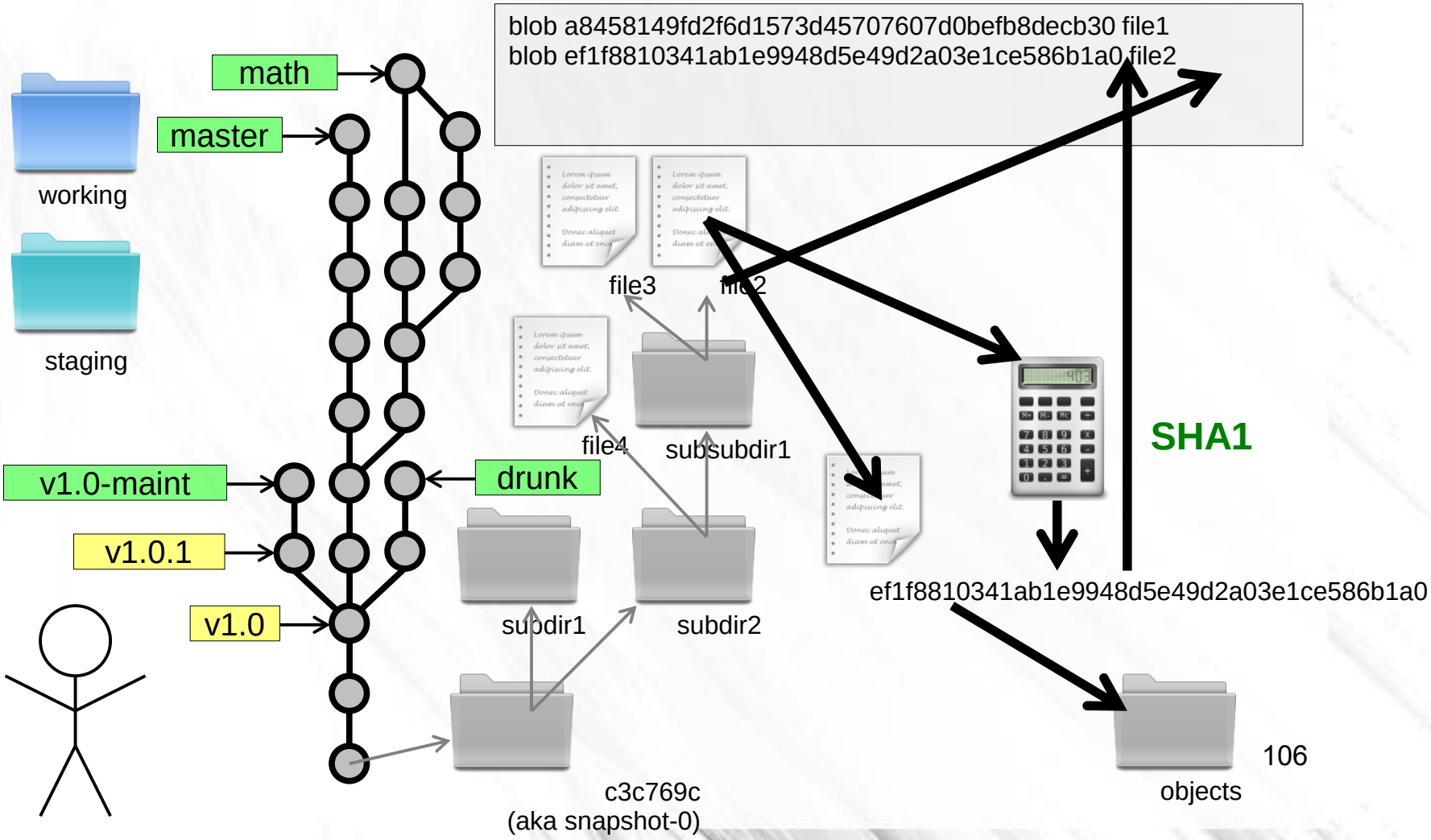
Eliminating Duplication



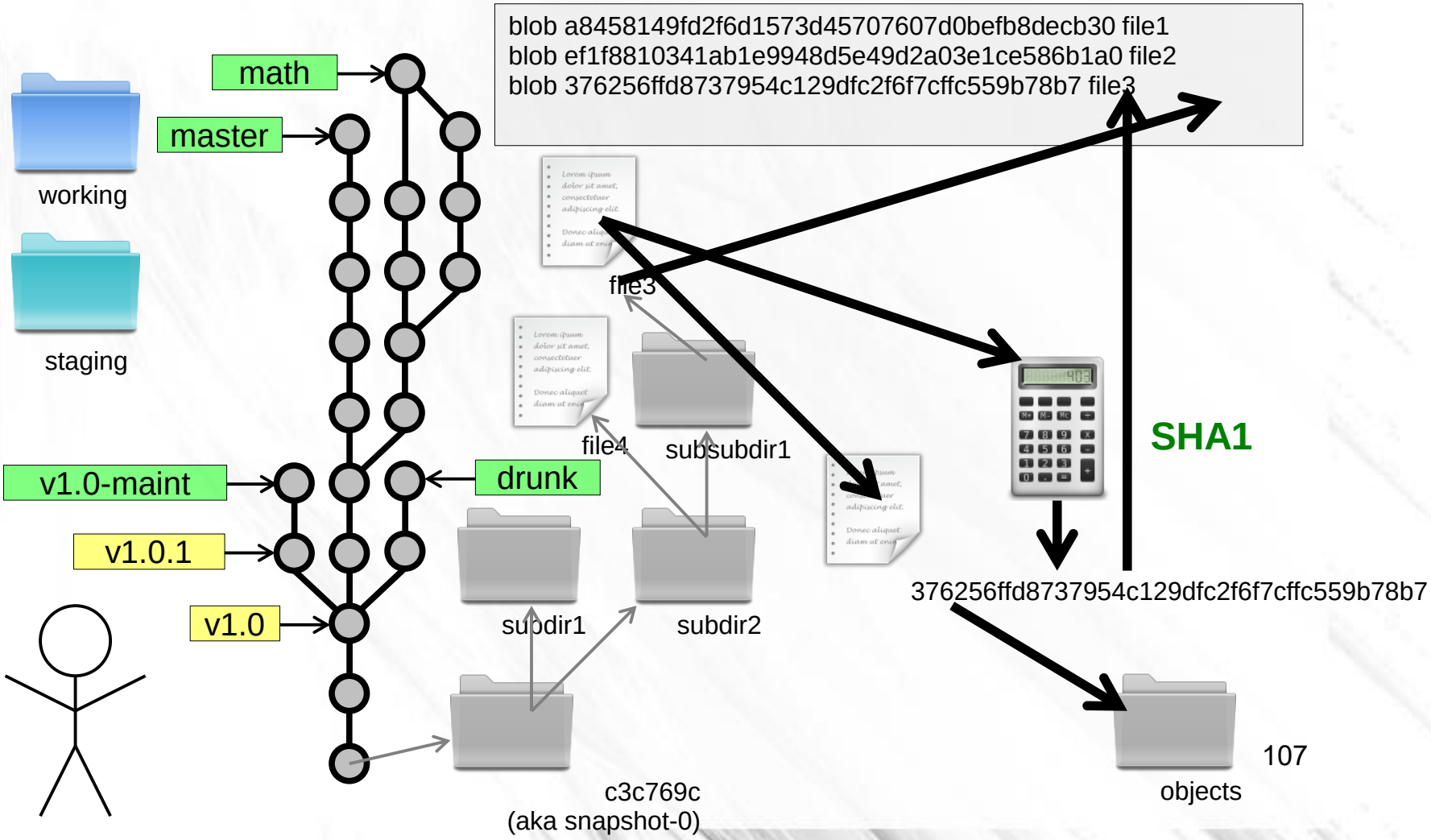
Eliminating Duplication



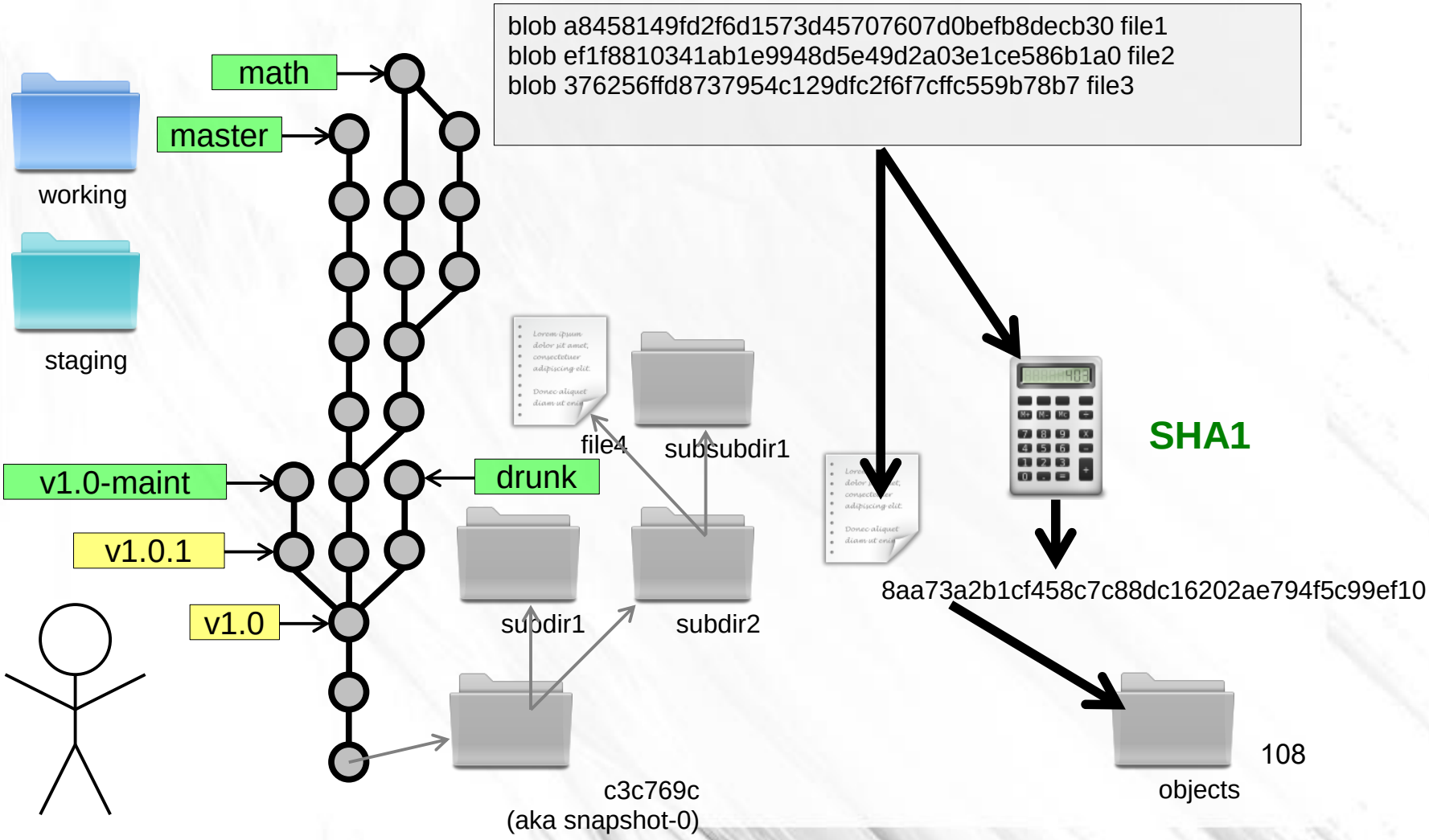
Eliminating Duplication



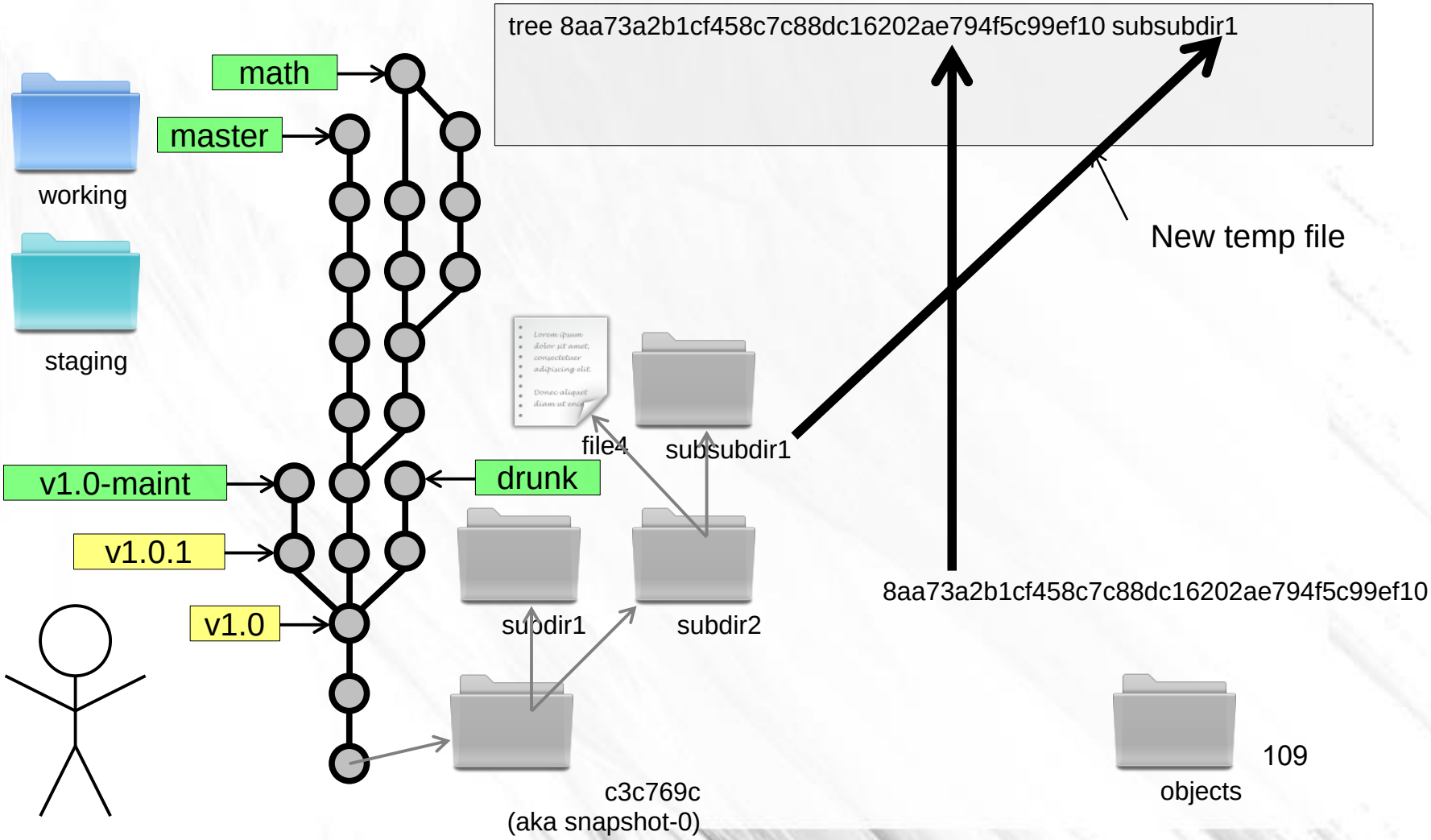
Eliminating Duplication



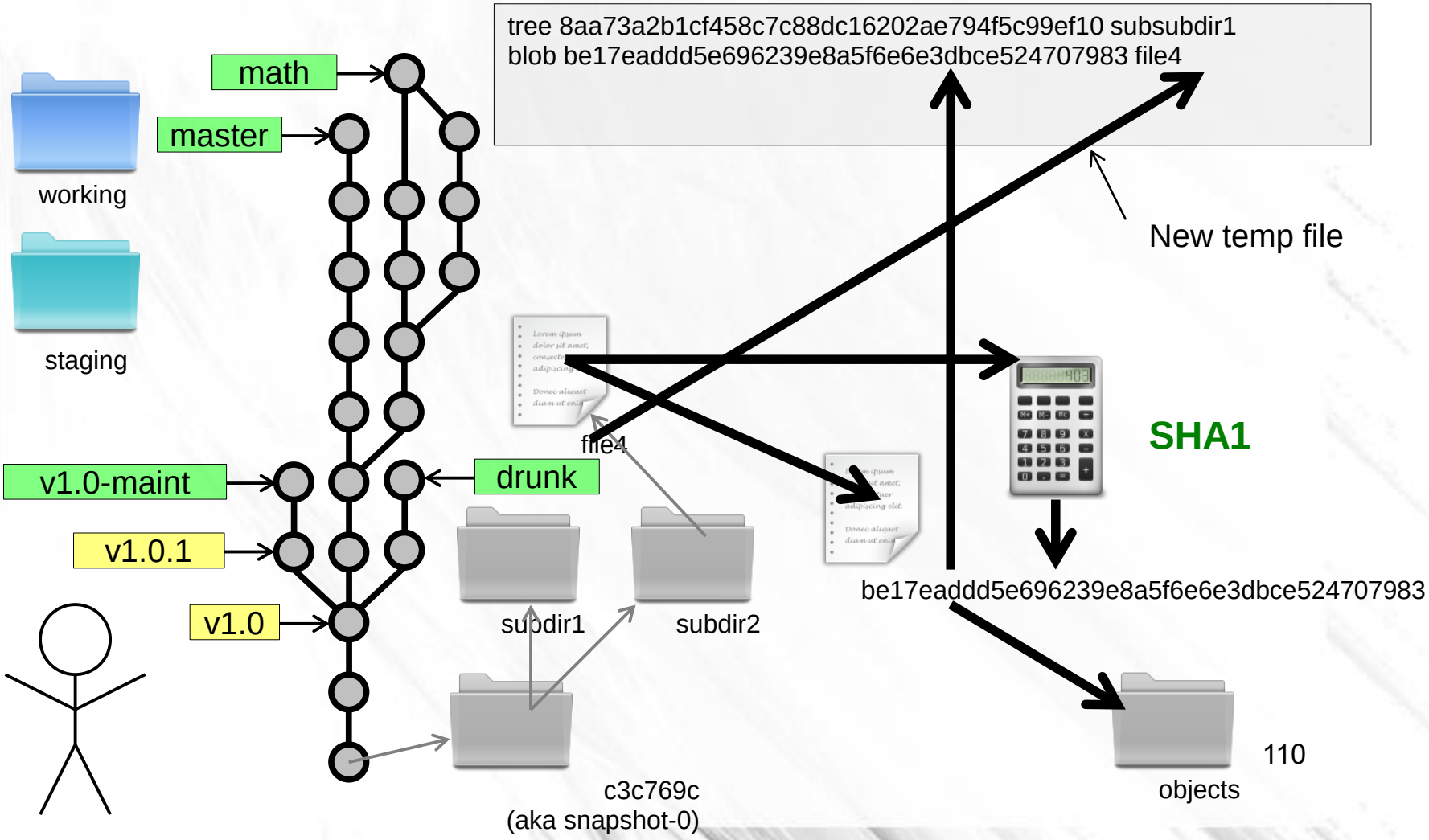
Eliminating Duplication



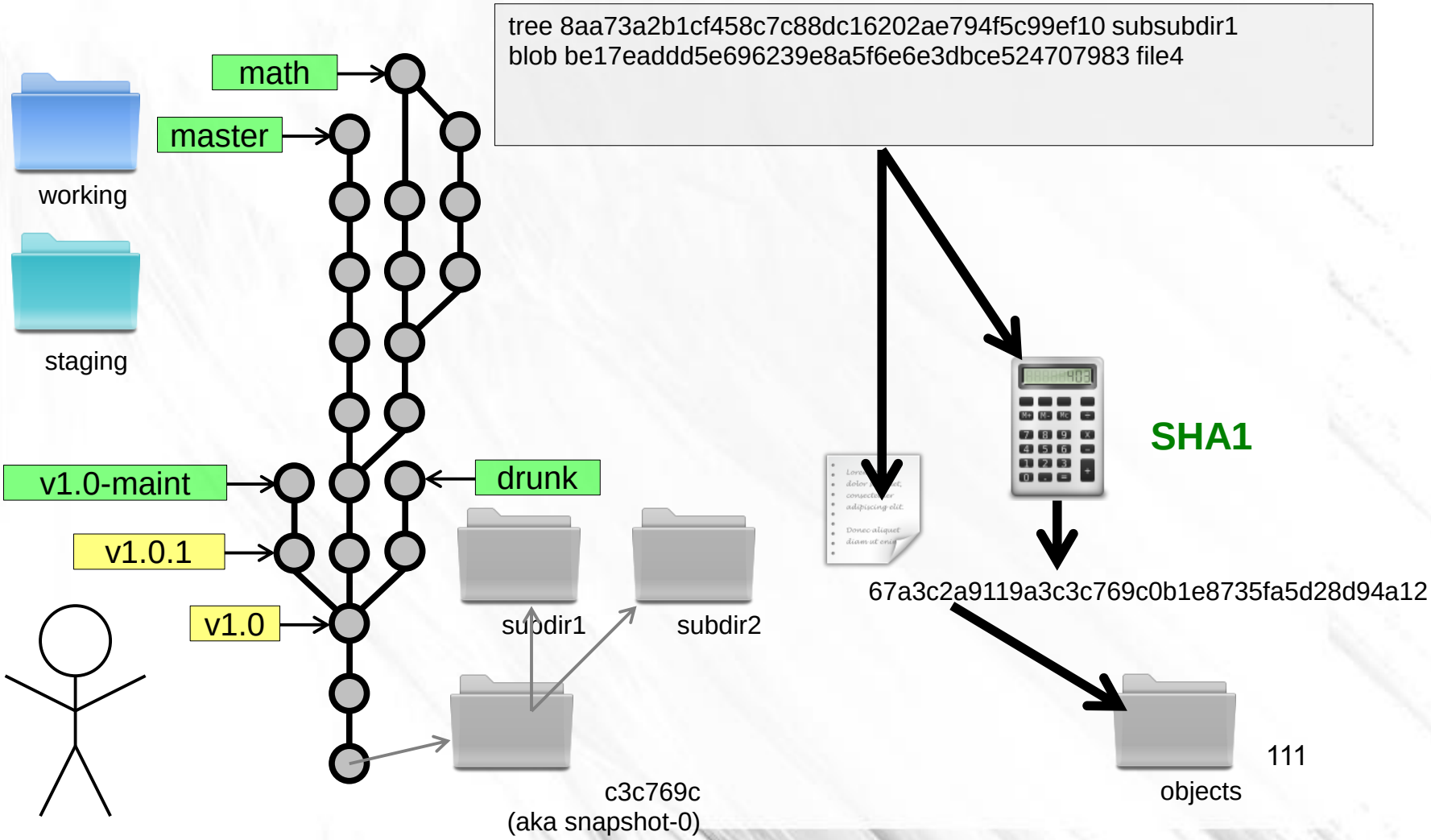
Eliminating Duplication



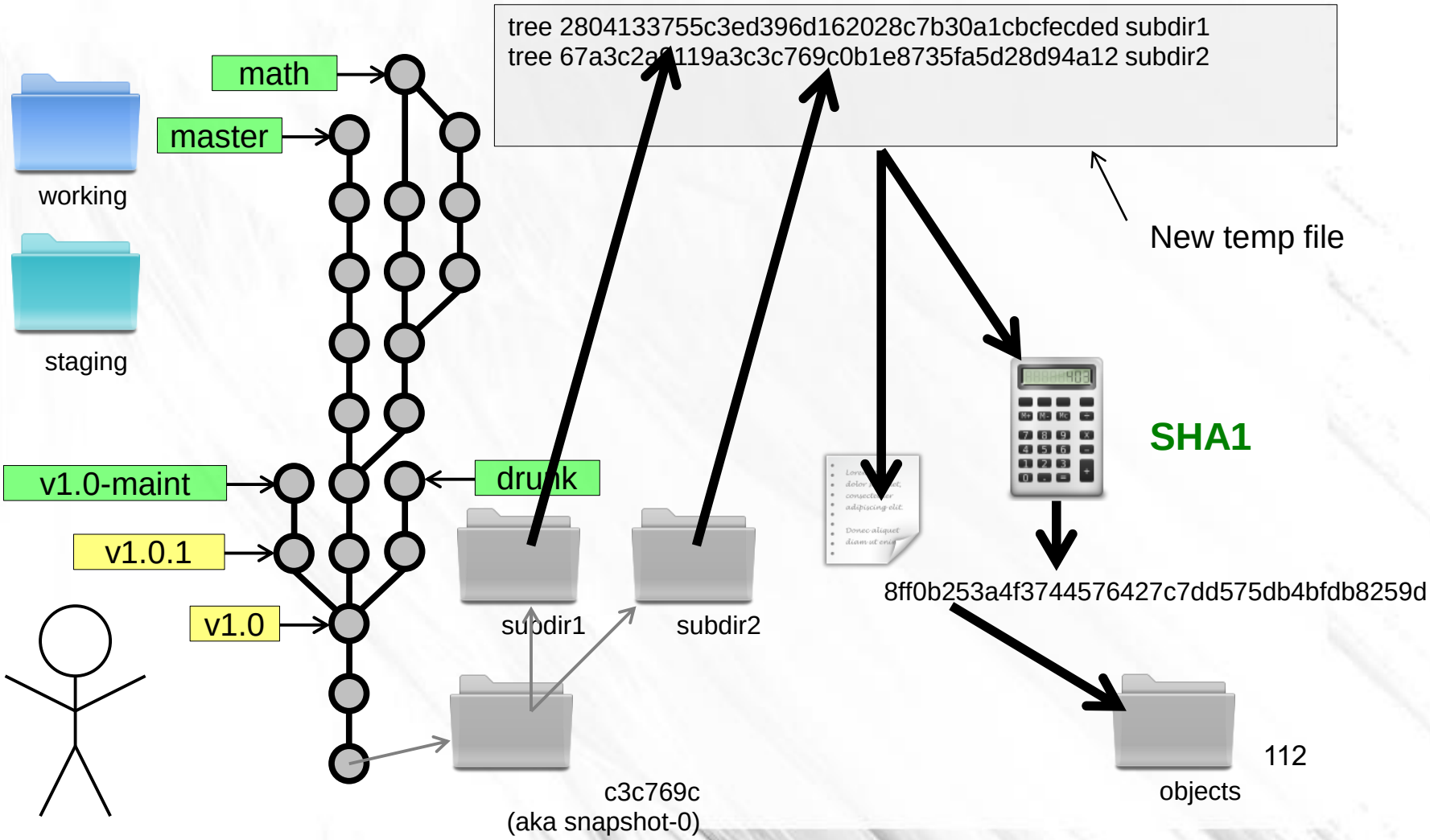
Eliminating Duplication



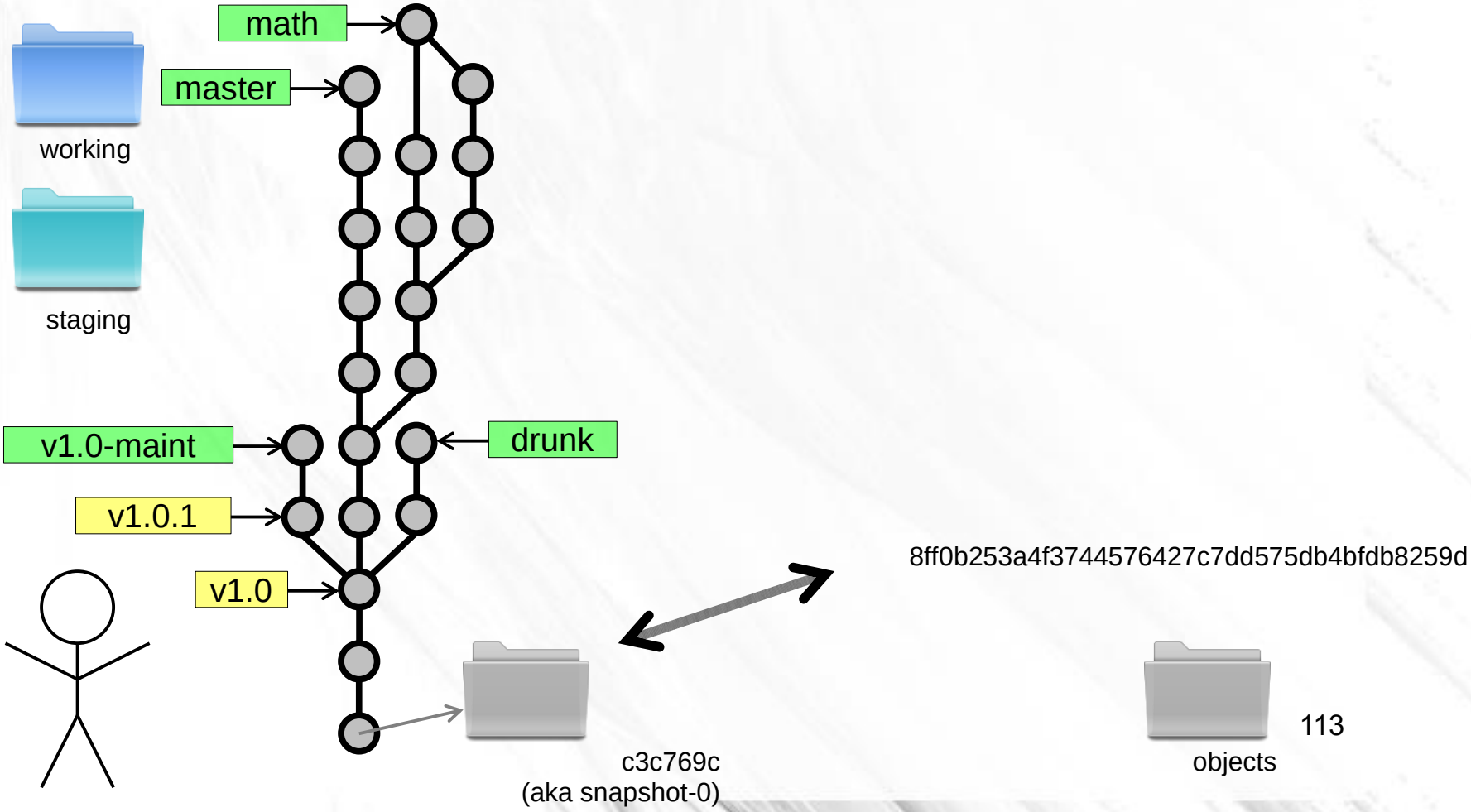
Eliminating Duplication



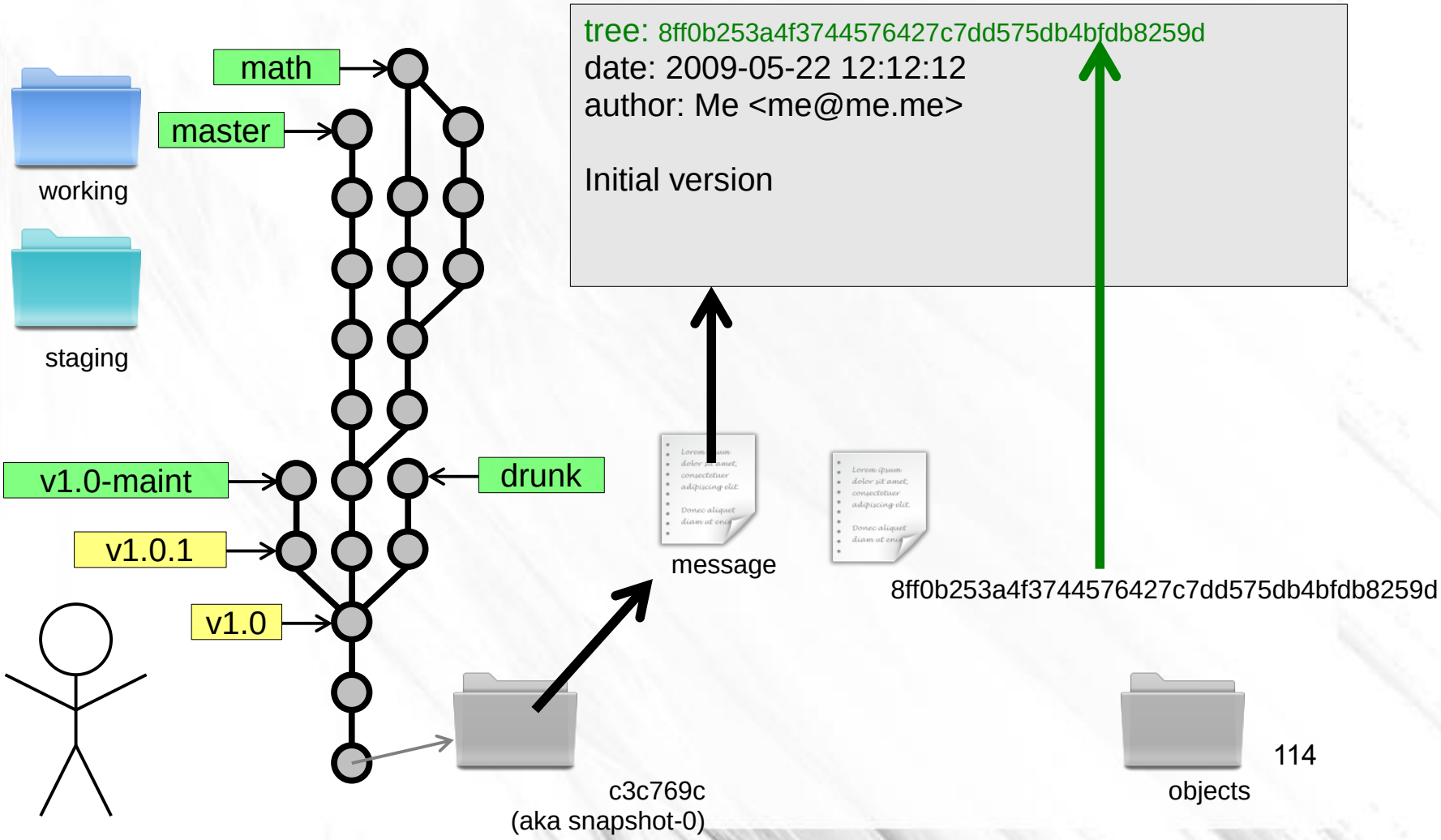
Eliminating Duplication



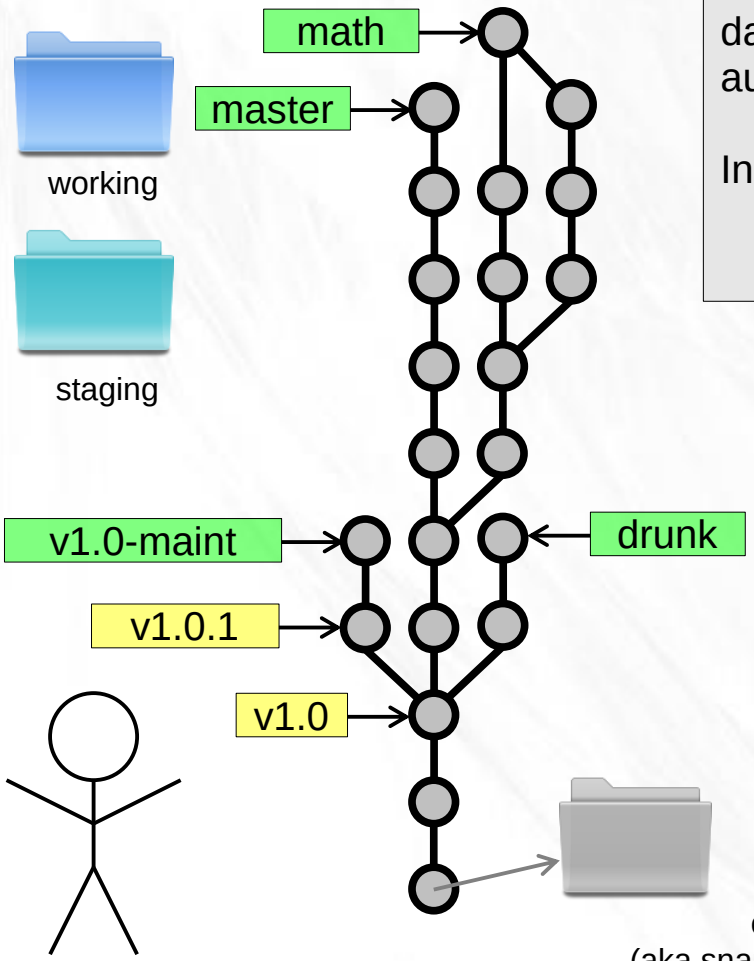
Eliminating Duplication



Eliminating Duplication

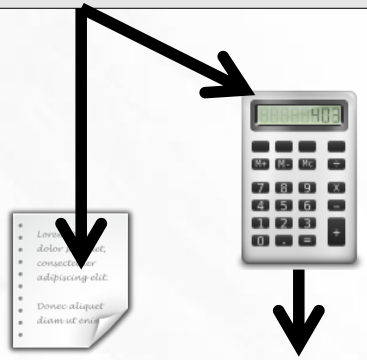


Eliminating Duplication



tree: 8ff0b253a4f3744576427c7dd575db4bfdb8259d
date: 2009-05-22 12:12:12
author: Me <me@me.me>

Initial version



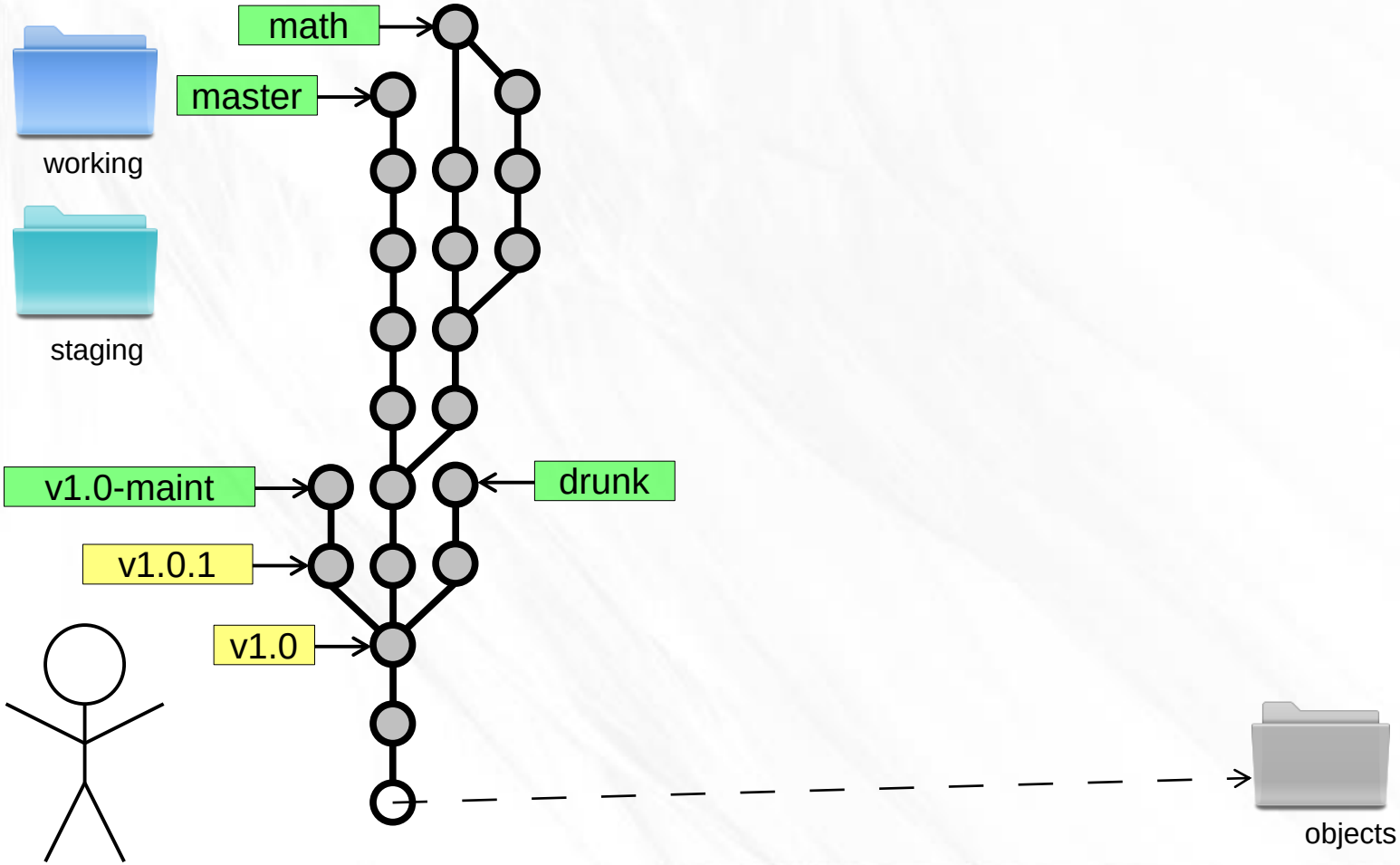
SHA1

5a2fa987819083dbcf44d36697b7ecbe3bfc9a3f

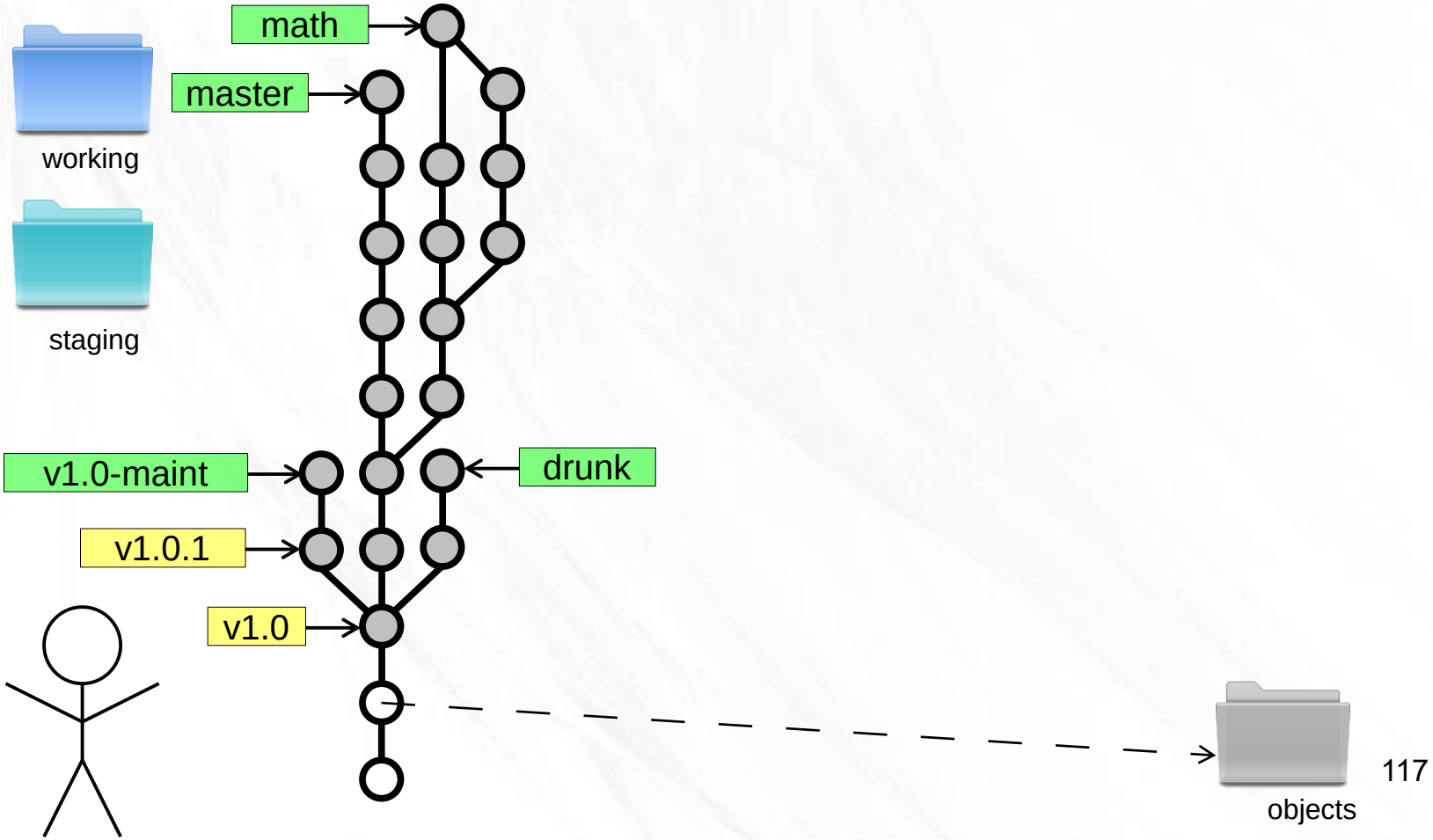
c3c769c
(aka snapshot-0)

objects

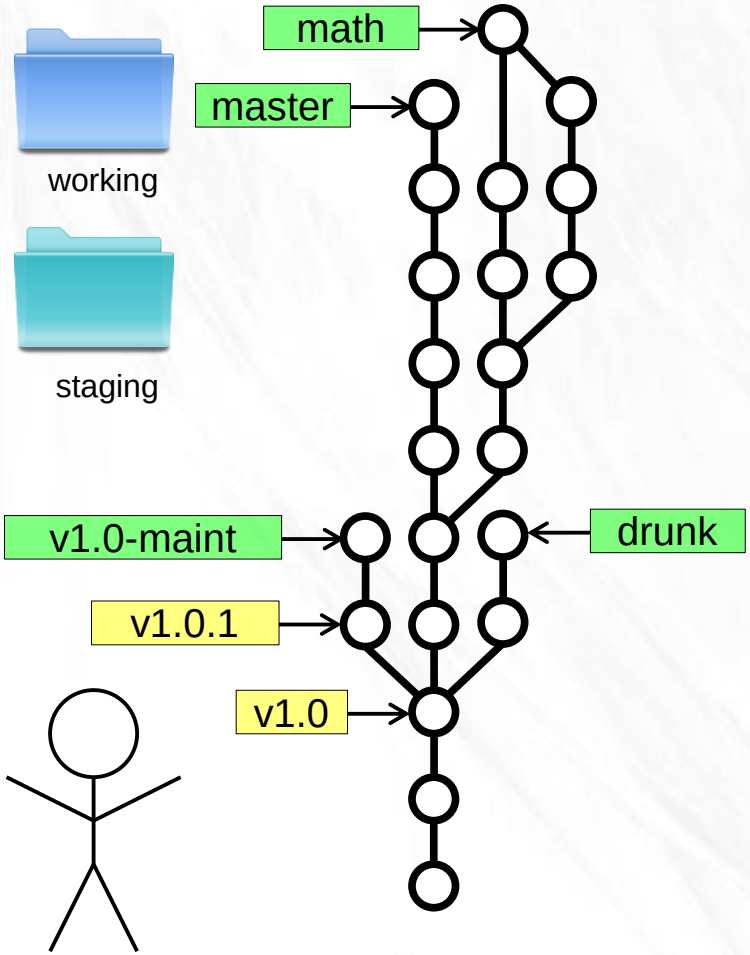
Eliminating Duplication



Eliminating Duplication

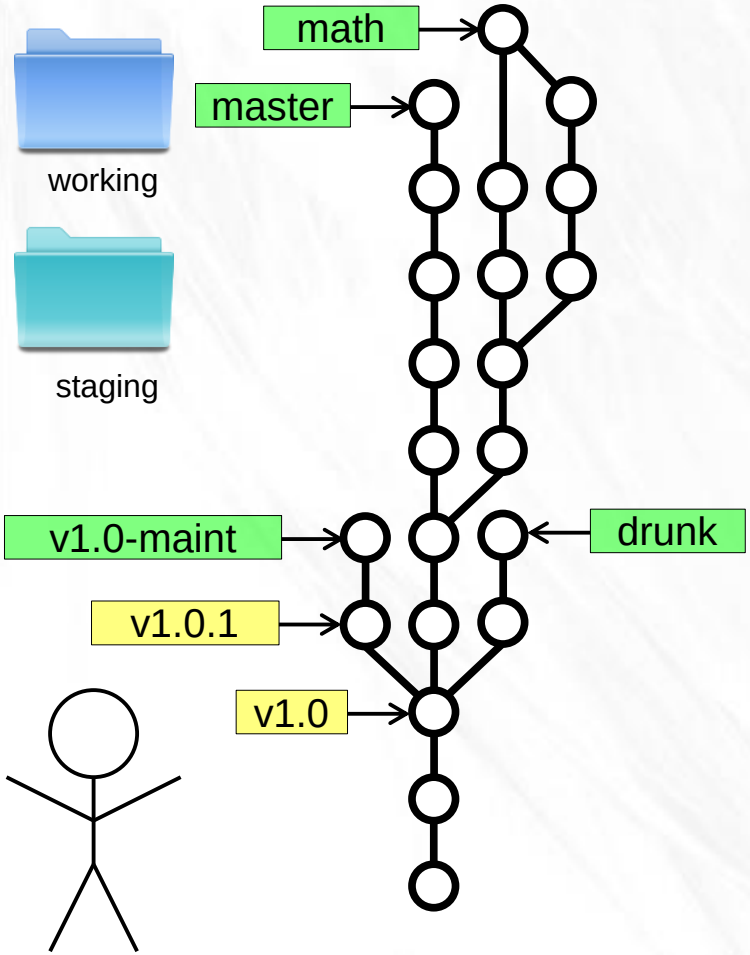


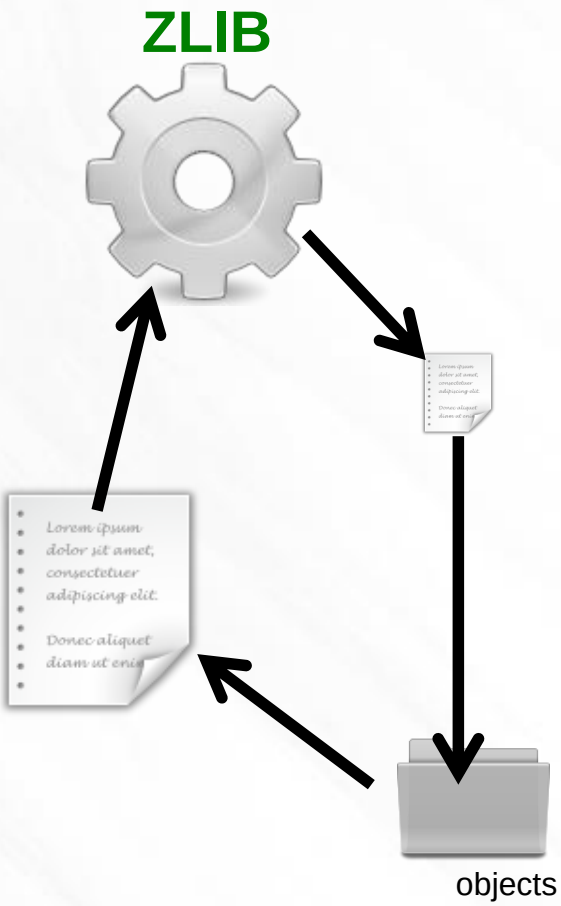
Eliminating Duplication



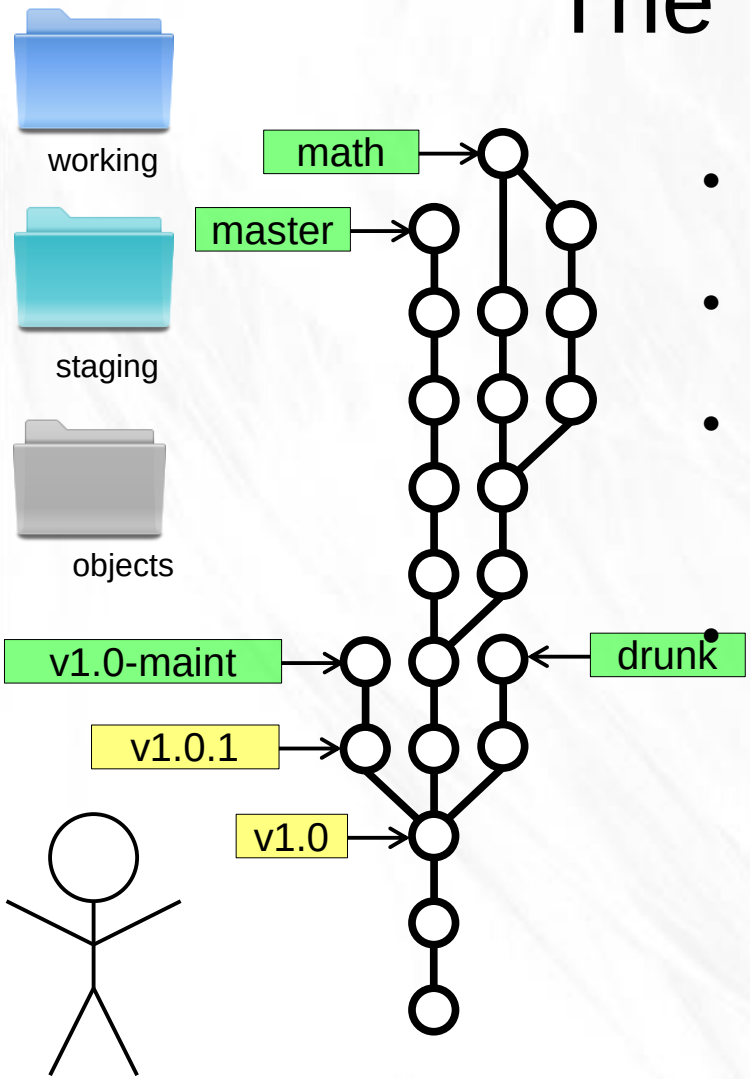

objects

Compressing Blobs





The True Git

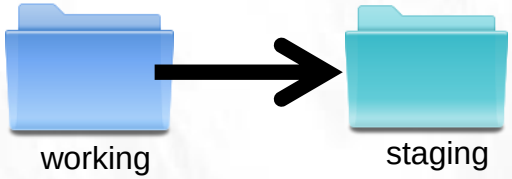


- TADAA!
- This is pretty much Git
- Nicer command line tools for all these operations
- Many, many other tools

Commands: Getting Started

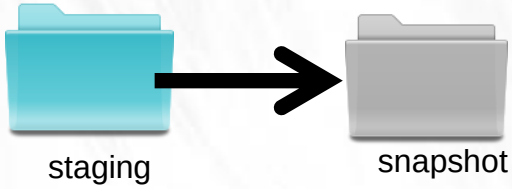
- First, tell Git who you are:
  - git config --global user.name “My Name”
  - git config --global user.email “my@email.address”
- Get help:
  - git <command> -h
  - git help <command>
- Start a new Git repository:
  - git init

Commands: Making snapshots



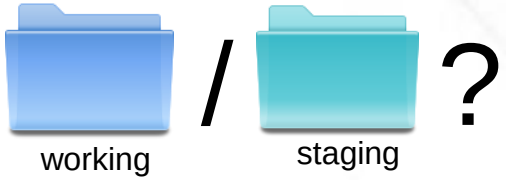
- git add

git commit -a

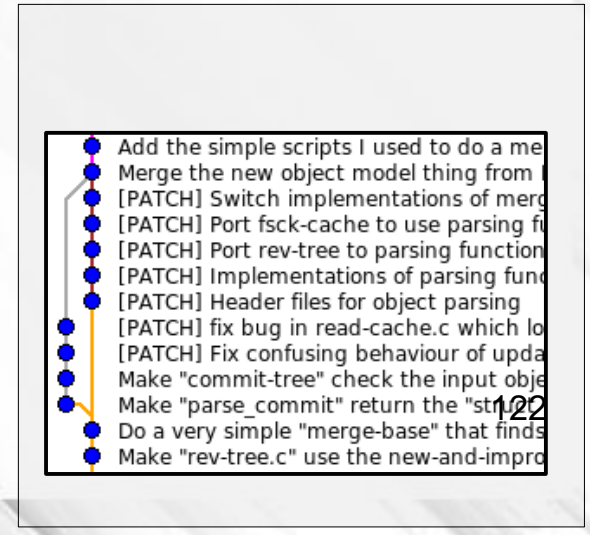


- git commit

- git log

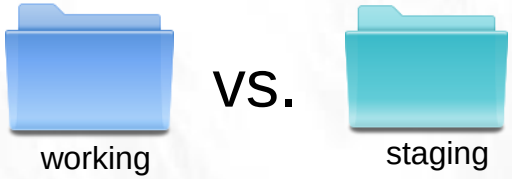


- git status

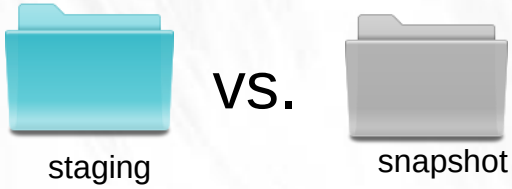


gitk

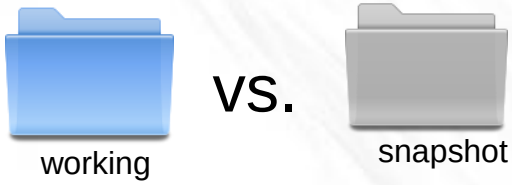
Commands: Diffing



- git diff



- git diff --staged



- git diff HEAD

- git diff <from> <to>

Commands: Branches & Tags

- git branch
- git branch <branch>
- git checkout <branch>
- git tag -l
- git tag <tag>

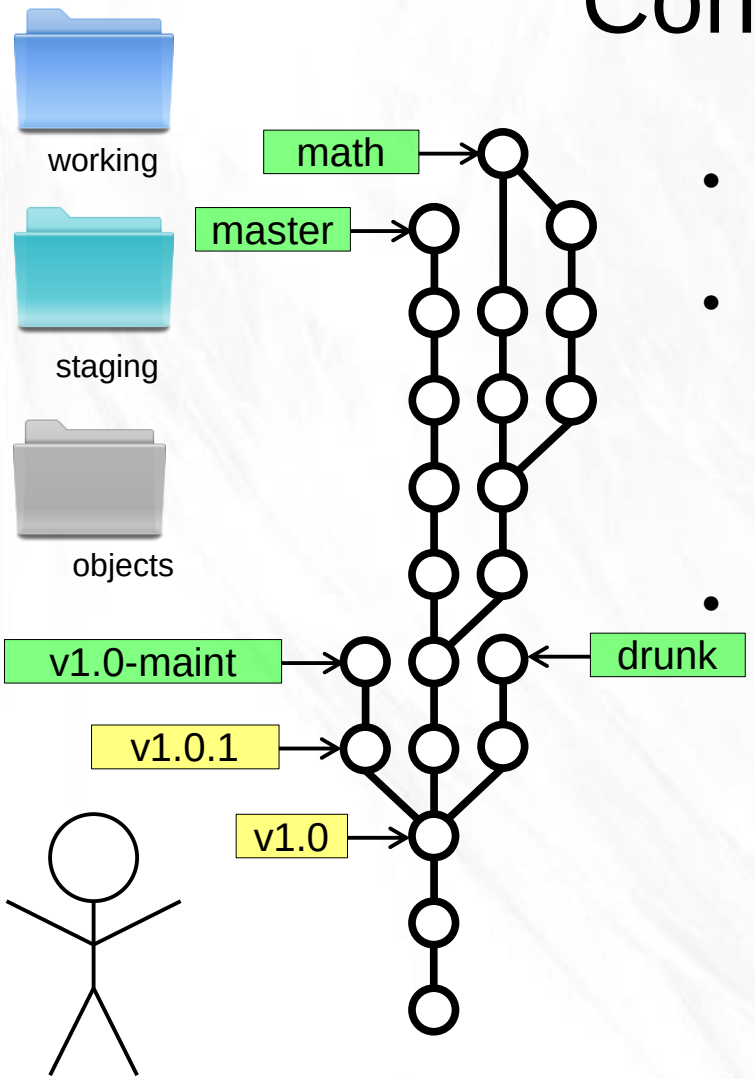
git checkout -b ...

Commands: Fetching & Merging

- git remote add <name> <URL>
- git fetch <name>
- git merge <name>/<branch>

git pull

Conclusion



- Keep this parable in mind
- Git is simple and powerful
- One more thing:

  git reflog

Where to go next?

- Git homepage: http://git-scm.com
- Pro Git: http://git-scm.com/book
- Git Reference: http://gitref.org
- GitHub: http://github.com
- Gitorious: http://gitorious.org
- Gitlab

Questions?

- Thanks for your attention!
- These slides are available at:
  https://github.com/jherland/git_parable
- Reach me at <johan@herland.net>

Git Clients (Windows)

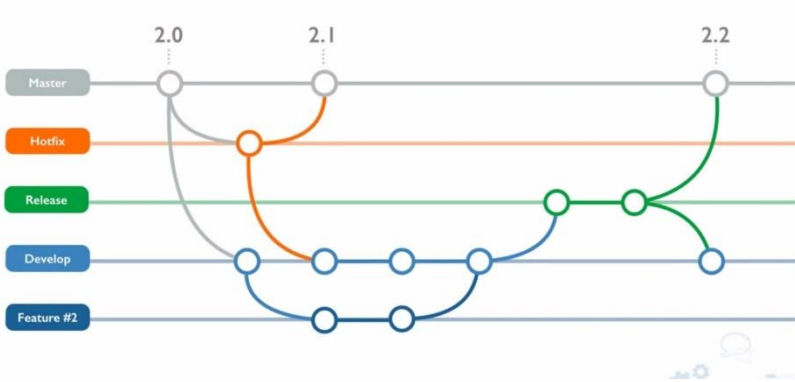
- CLI Shell: Git Bash
- Git Extentions: CLI +Windows Explorer Shell
- Github Desktop (+powershell)
- Bitbucket SourceTree
- IDE Integration
  - Visual Studio (Code / VS2013+ native)
  - Eclipse Egit
  - IntelliJ, Webstorm embedded
  - Brackets plugin
  - VS Code

הדגמה \ טיפים

- http://try.github.com
- http://learn.github.com/ (education.. - free student account)
- http://help.github.com/create-a-repo/
  Local user settings:
  git config user.name <user>
  git config user.email user@example.com
- git pull
- git add: add / stage
- git commit –a == add+commit

Git Workflows - שיטות

- Centralized Workflow
- Feature Branch Workflow
- Gitflow Workflow
- Forking Workflow (OSS)



- Github Flow
- Pull request – בפרויקט?
- דוגמא: תהליך העבודה בפרויקט Nuget:
  Contributing a Bug Fix or Feature
- קישורים נוספים בויקי

בפעם הבאה

- פרויקט: פיתוח בסבבים
- משימה <u>אישית</u> מס' 3 – ליווי גם בהרצאה
- בקרת גרסאות II – תרחישים נוספים עם git - Workflows
- שעה שלישית – מעבדה (משימה אישית 3, להביא מחשבים)
  - הרצאת העשרה בקורס -דפנה גולד-מלכיאור - ארגז כלים למדענים- חידוד מסרים והעברת פרזנטציות
- בהמשך
  - בדיקות, פיתוח מונחה מפרטים
  - עקרונות תיכון מונחה עצמים ועוד

לסיכום

- תהליך: בקרת תצורה וגרסאות
- כלים: git / github
- שיטות: למשל git flow