

מודלים לפיתוח מערכות תוכנה

Software Systems Modeling

קורס 12003

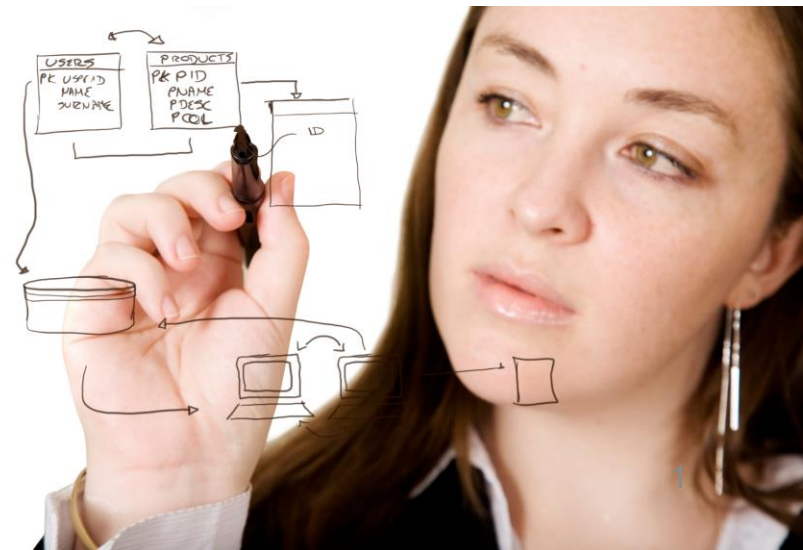
סמסטר ב' תשע"ה

9. מודל קונספטואלי לבקרת קוד מבוזרת

ד"ר ראובן יגל

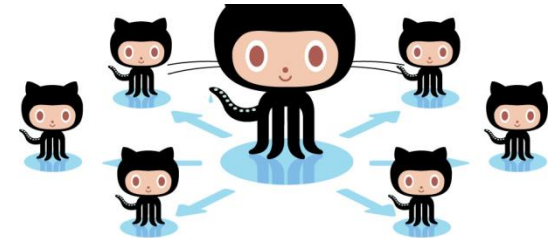
robi@post.jce.ac.il

modeling15b-yagel



הפעם

- מידול תהליכי פיתוח תוכנה
 - עבודת צוות ומחזור חיים
 - תיכון וארכיטקטורה -> High Level Models
 - בדיקות
 - ריבוע הבדיקות
 - תיעוד חי? (BDD)
 - מימוש : **גיט**
- המשך מצגות פרויקט
- סיכום הקורס והצגת הסקר



הנדסת תוכנה

6. 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 – Version Control
- כלים: git / github
- משימה אישית 3
- סקר ZFR
- פרויקט: תחילת סבבים – סבב 1 MVP
- תרגיל – סקר ZFR
(+ השלמת back-end)



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), ...

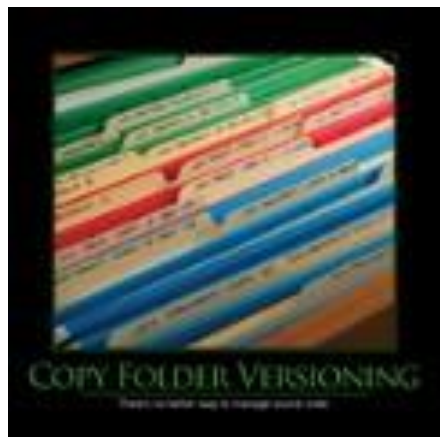
בקרת תצורה

- 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**

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 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](https://github.com/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
 - 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_pparable

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... .

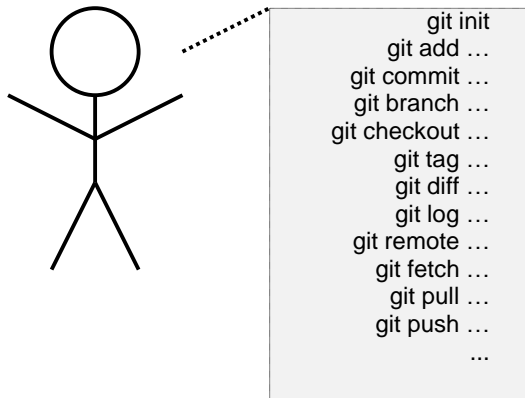
Also: Best introduction to Git I've found so far .

The Git Parable

Git - simple & powerful .

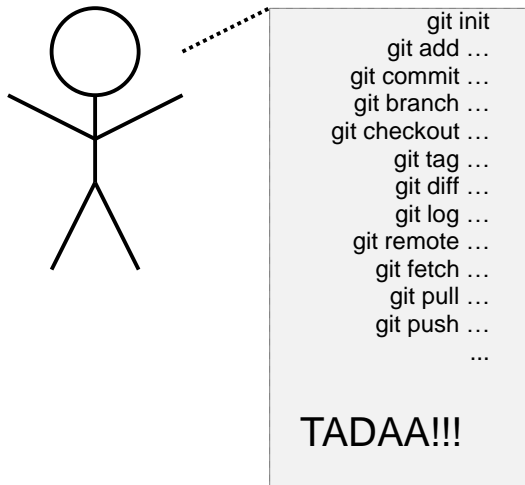
The Git Parable

Git - simple & powerful .



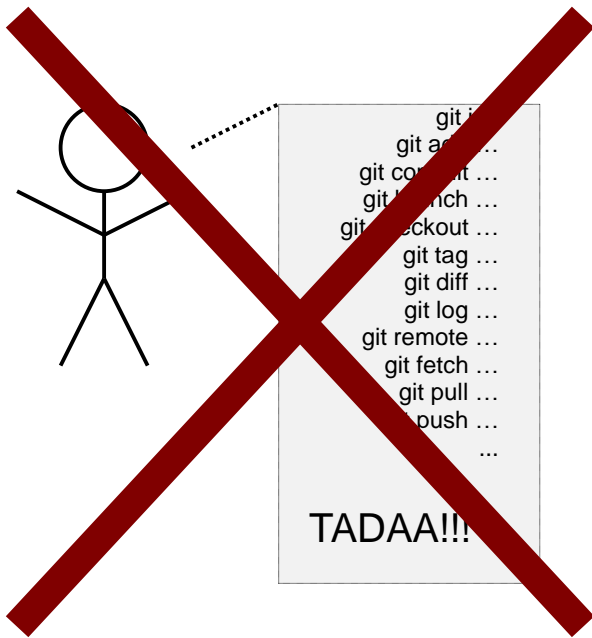
The Git Parable

Git - simple & powerful .



The Git Parable

Git - simple & powerful .



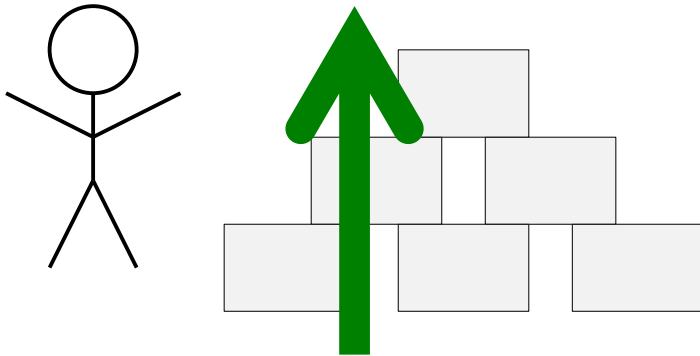
The Git Parable

Git - simple & powerful .



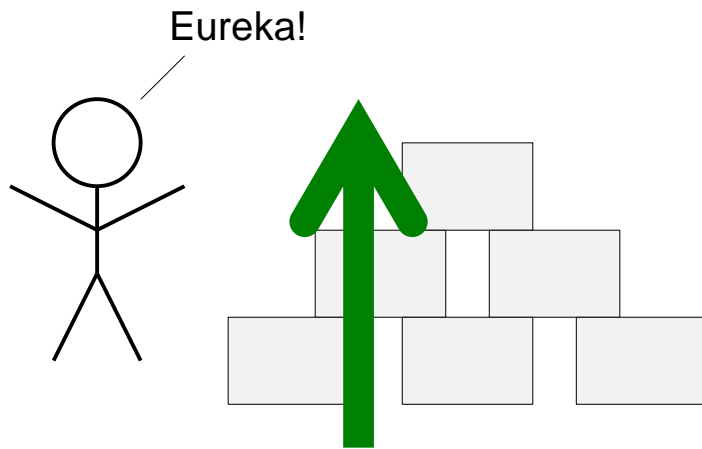
The Git Parable

Git - simple & powerful .



The Git Parable

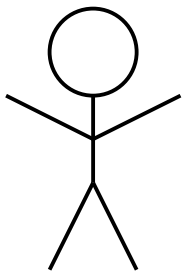
Git - simple & powerful .



The Parable

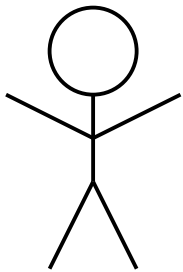
A simple computer .

- A text editor
- A few filesystem commands



The Parable

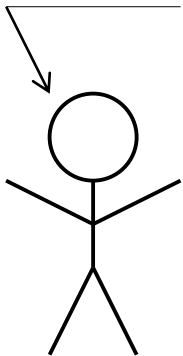
Write a large software program .



The Parable

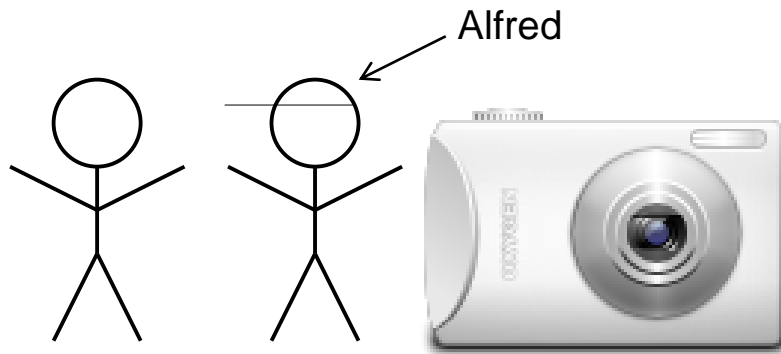
- Write a large software program .
- Invent some method to keep track of versions .
 - retrieve code that you changed/deleted

Responsible!



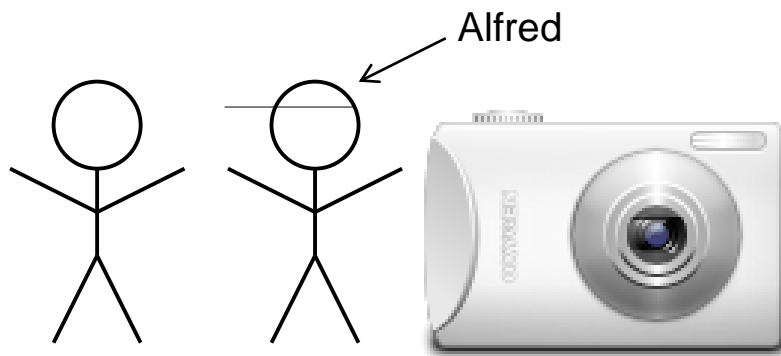
Snapshots

Alfred, the photographer .



Snapshots

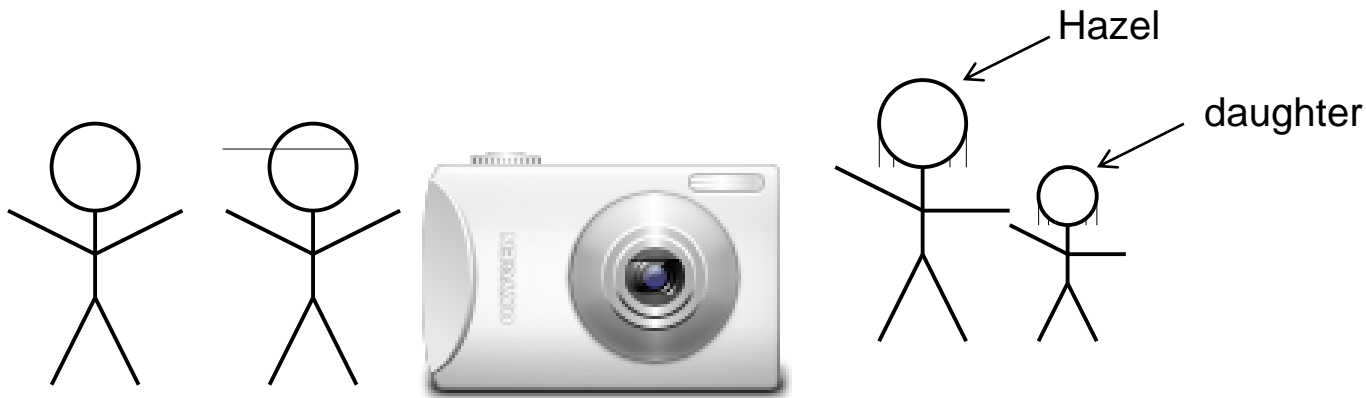
Alfred



Snapshots

Alfred, the photographer .

Hazel and her daughter .

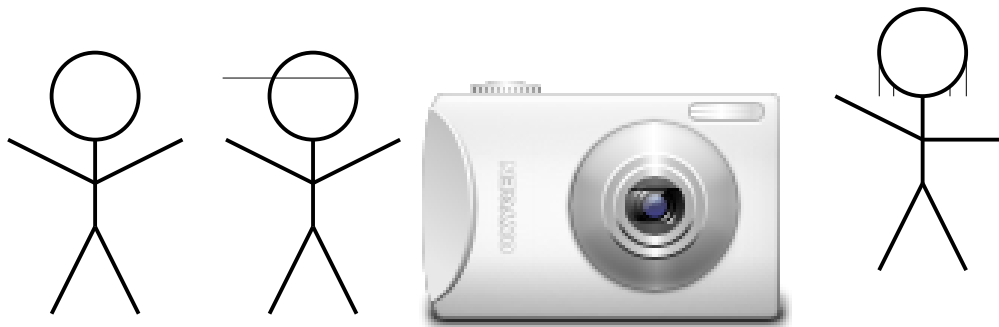


Snapshots

Alfred, the photographer .

Hazel and Hazel .

- Remember that the daughter is like at each different stage

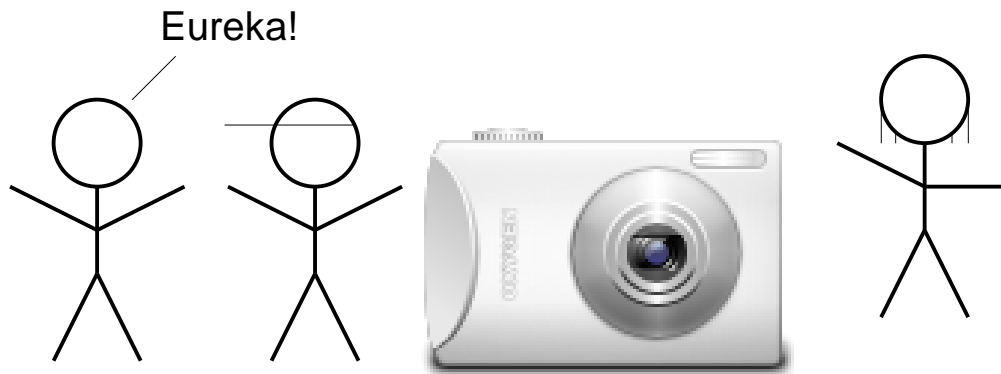


Snapshots

Alfred, the photographer .

Hazel and Hazel .

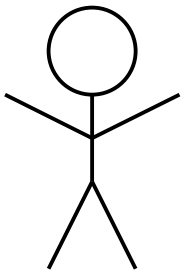
- Remember that the daughter is like at each different stage



Snapshots

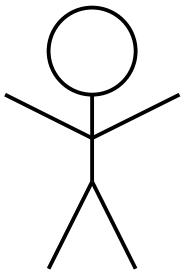
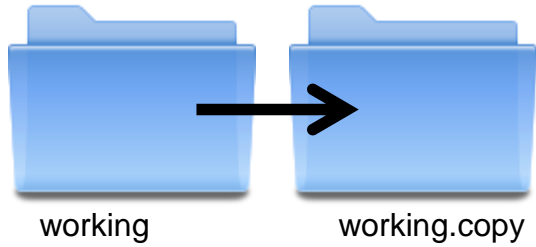


working

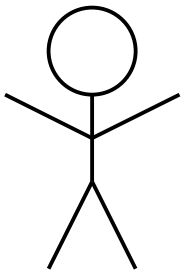


se15b-yagel

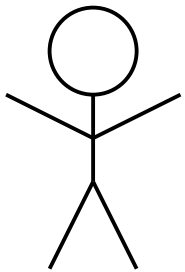
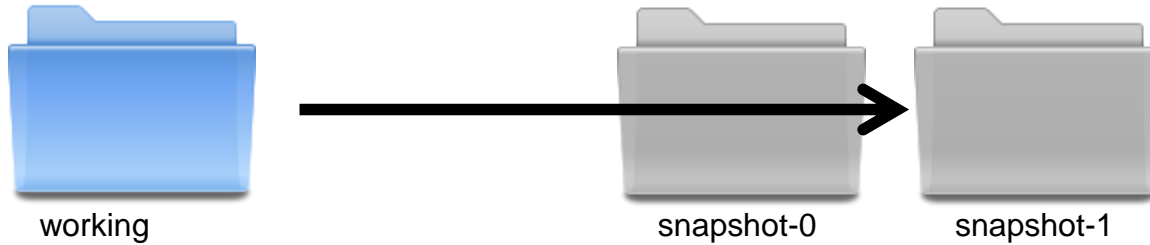
Snapshots



Snapshots



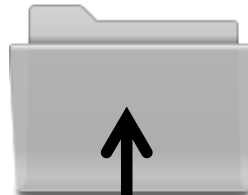
Snapshots



Snapshots



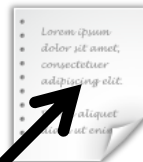
working



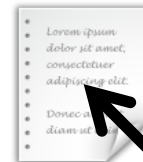
snapshot-0



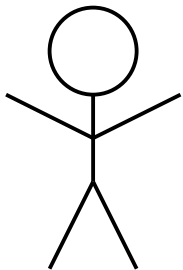
snapshot-1



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
xyzzzy.

Branches



working



snapshot-0

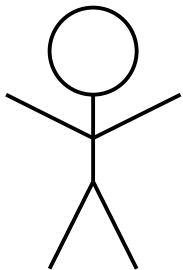


snapshot-1

...



snapshot-99



Branches



working



snapshot-0

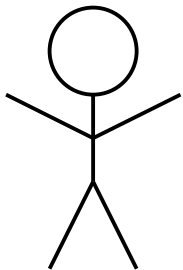


snapshot-1

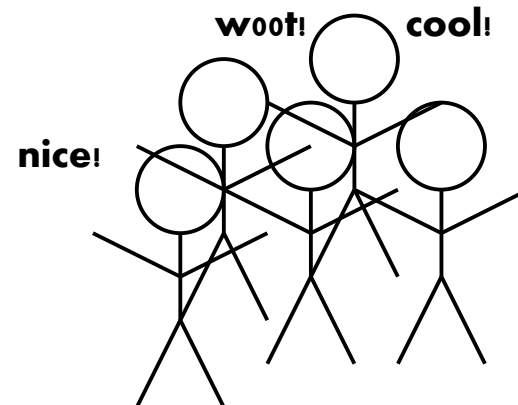
...



snapshot-99



se15b-yagel



Branches



working



snapshot-0



snapshot-1

...

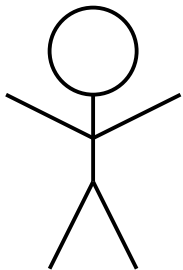


snapshot-99

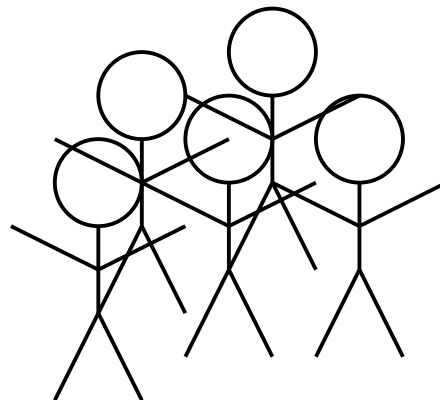
...



snapshot-109



se15b-yagel



Branches



working



snapshot-0



snapshot-1

...



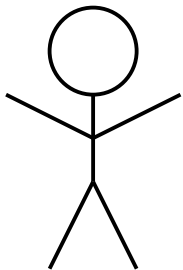
snapshot-99

...

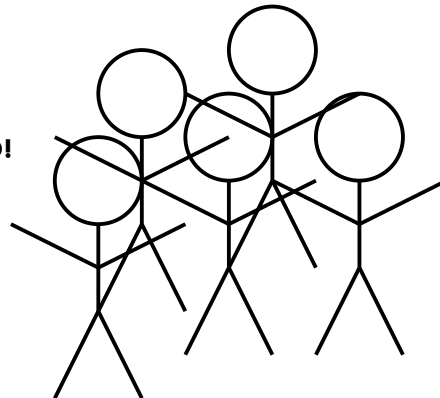


snapshot-109

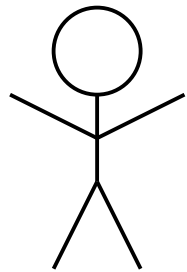
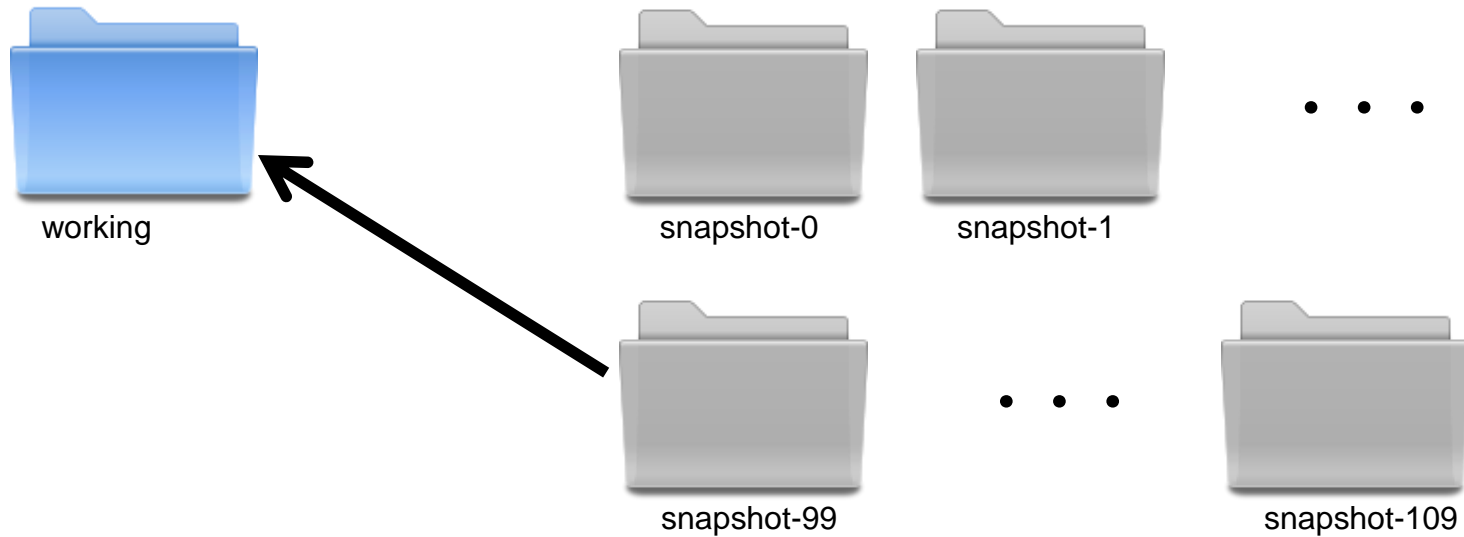
Does not work!



Boo!



Branches



Branches



working



snapshot-0



snapshot-1

...

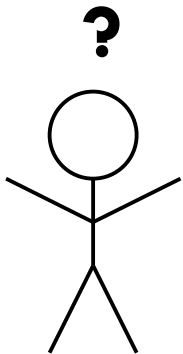


snapshot-99

...



snapshot-109

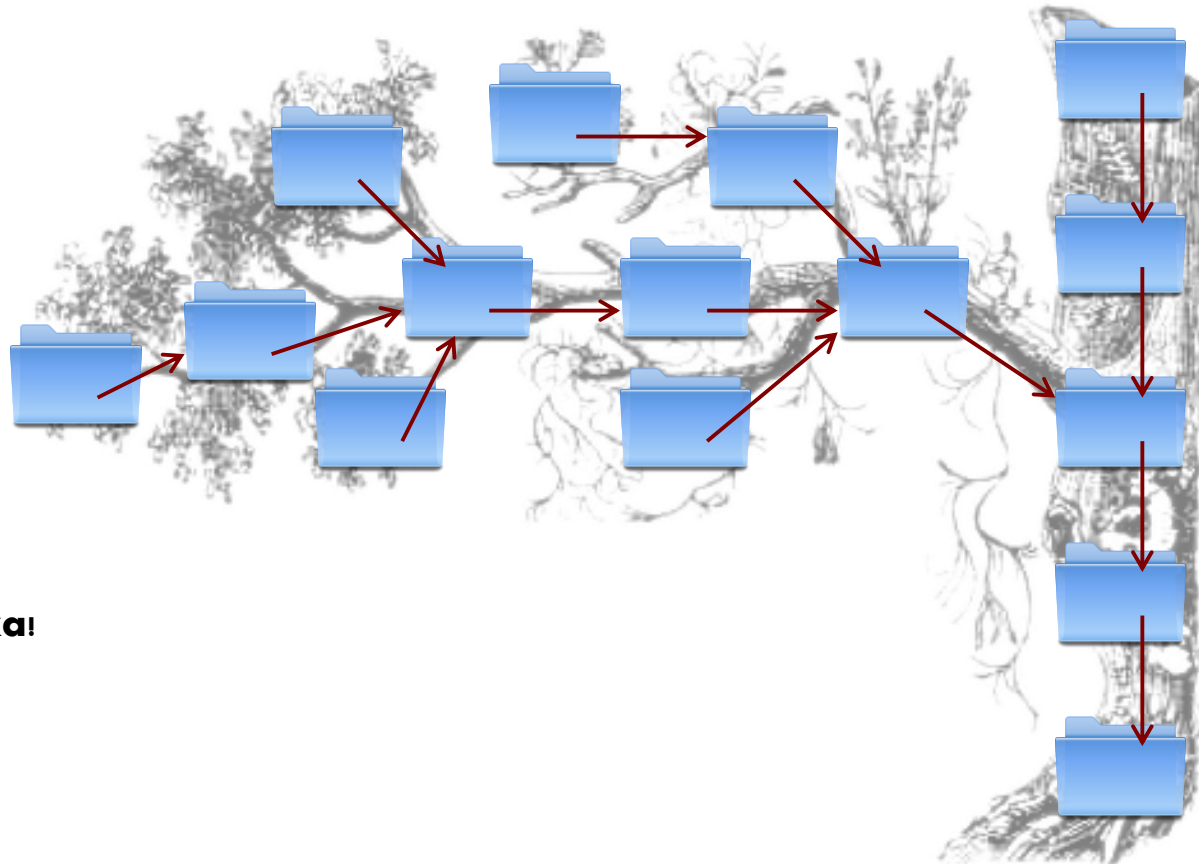


snapshot-110

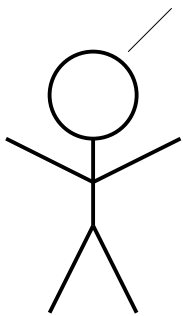
Branches



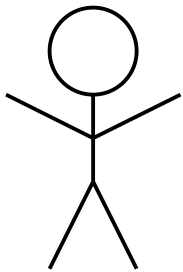
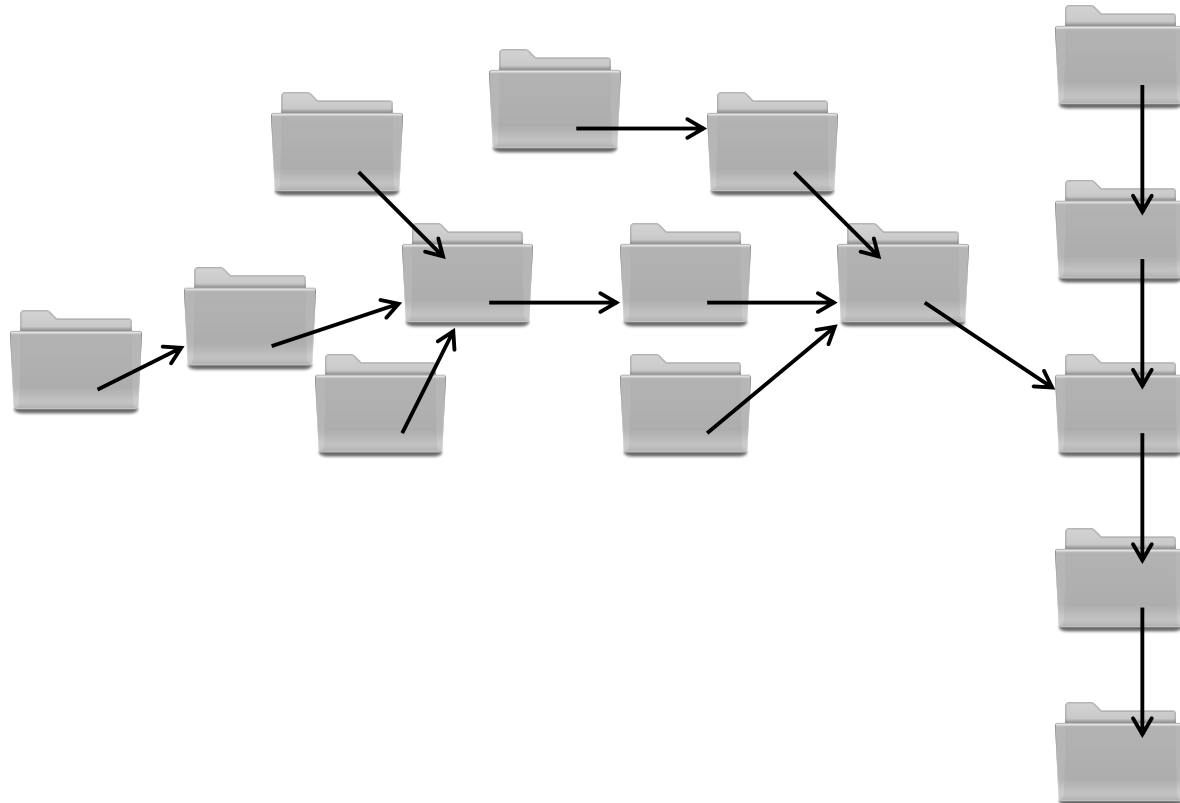
Branches



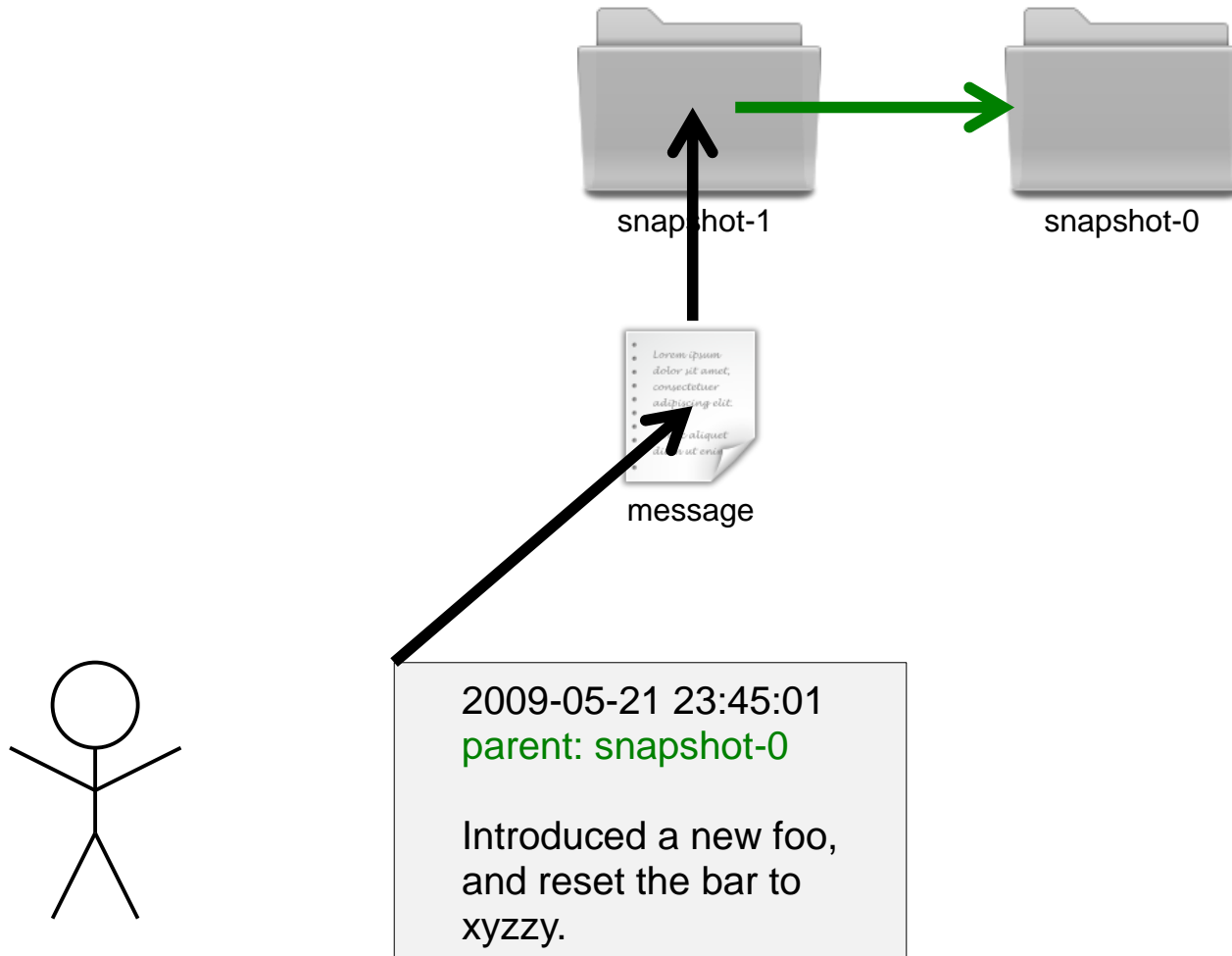
Eureka!



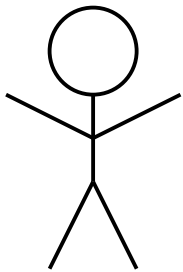
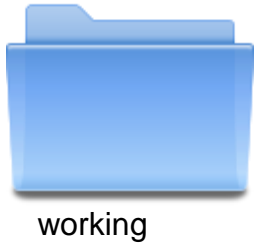
Branches



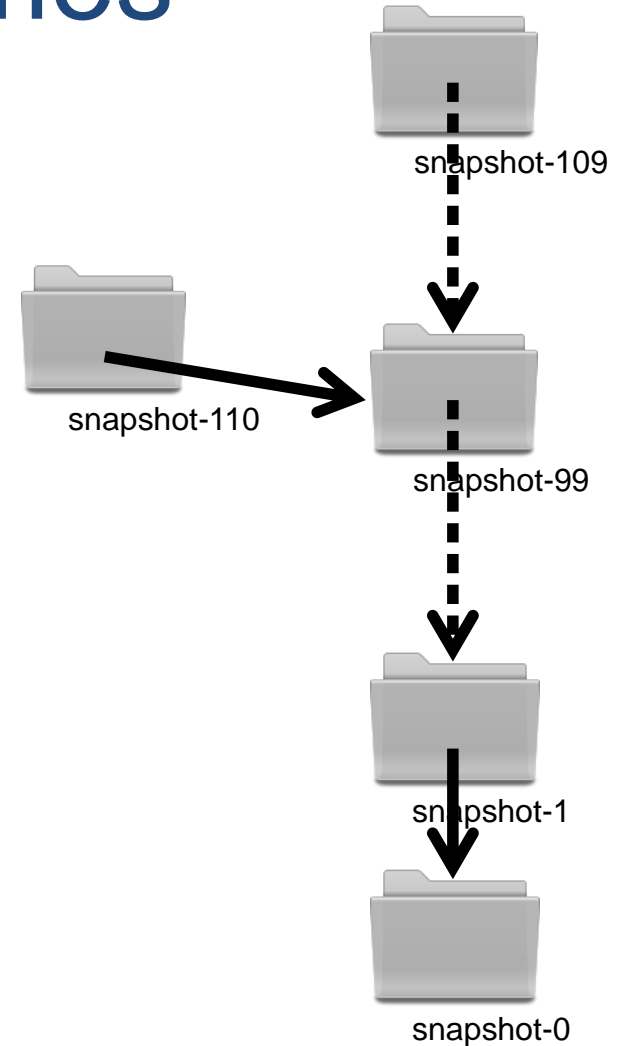
Branches



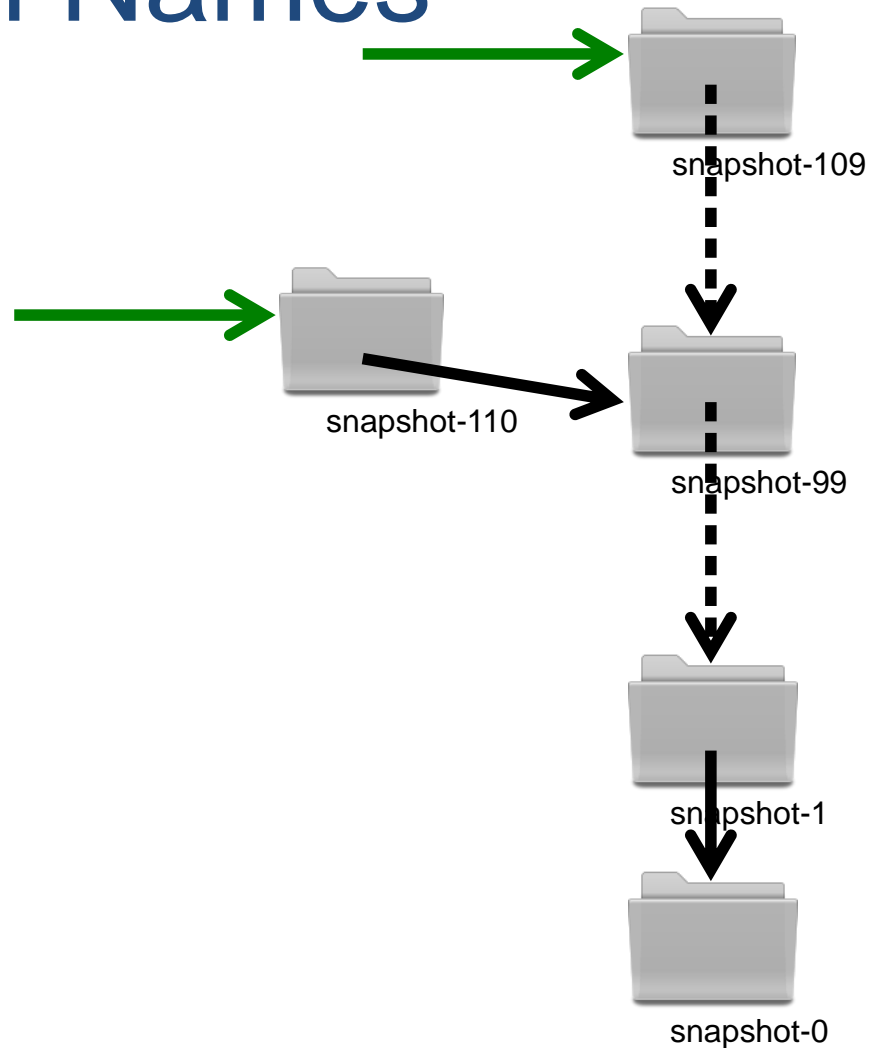
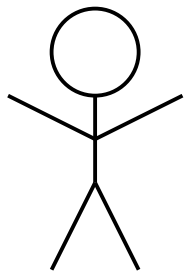
Branch Names



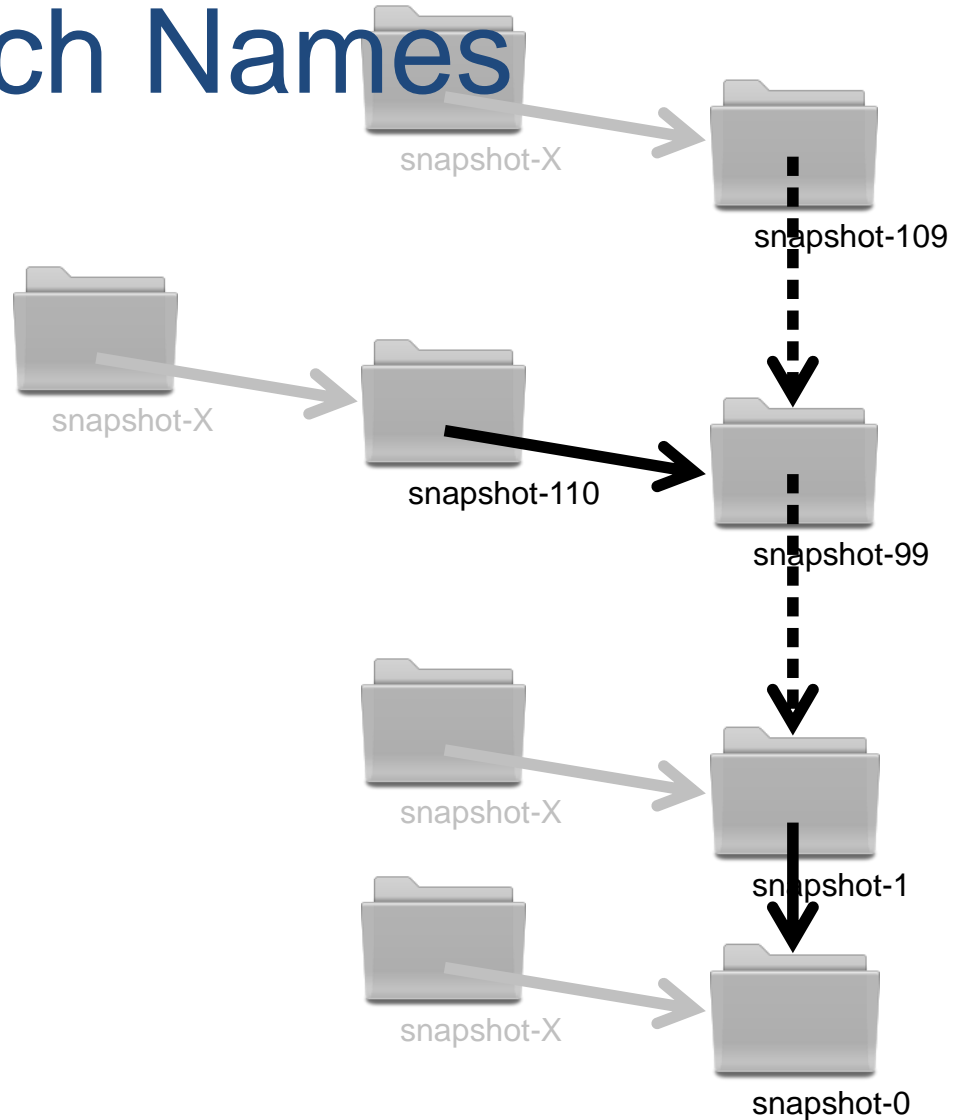
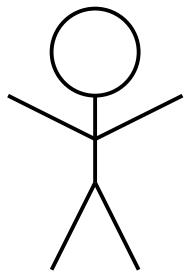
se15b-yagel



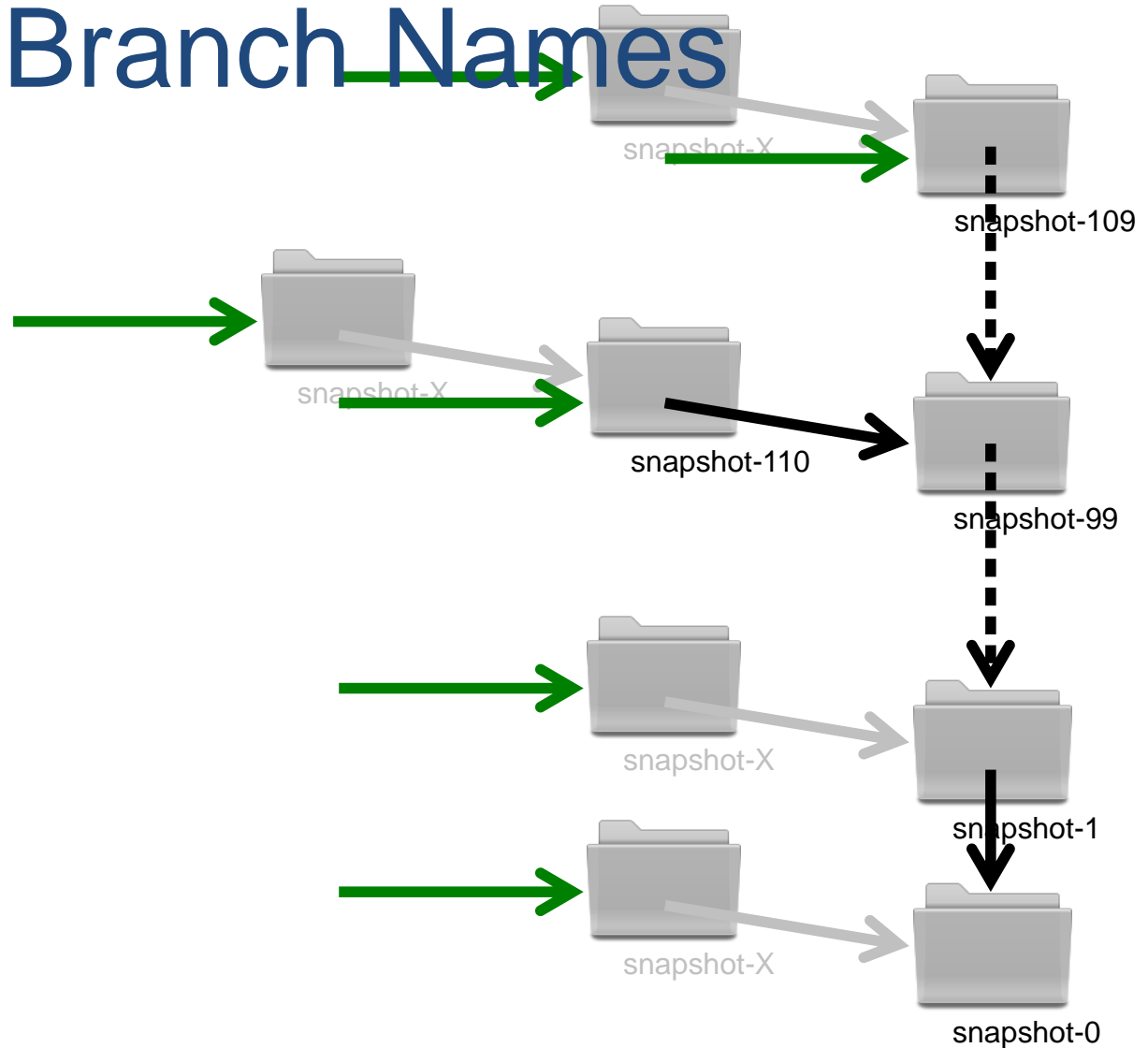
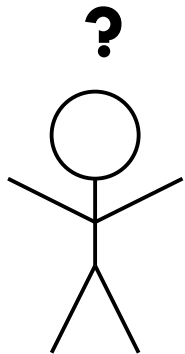
Branch Names



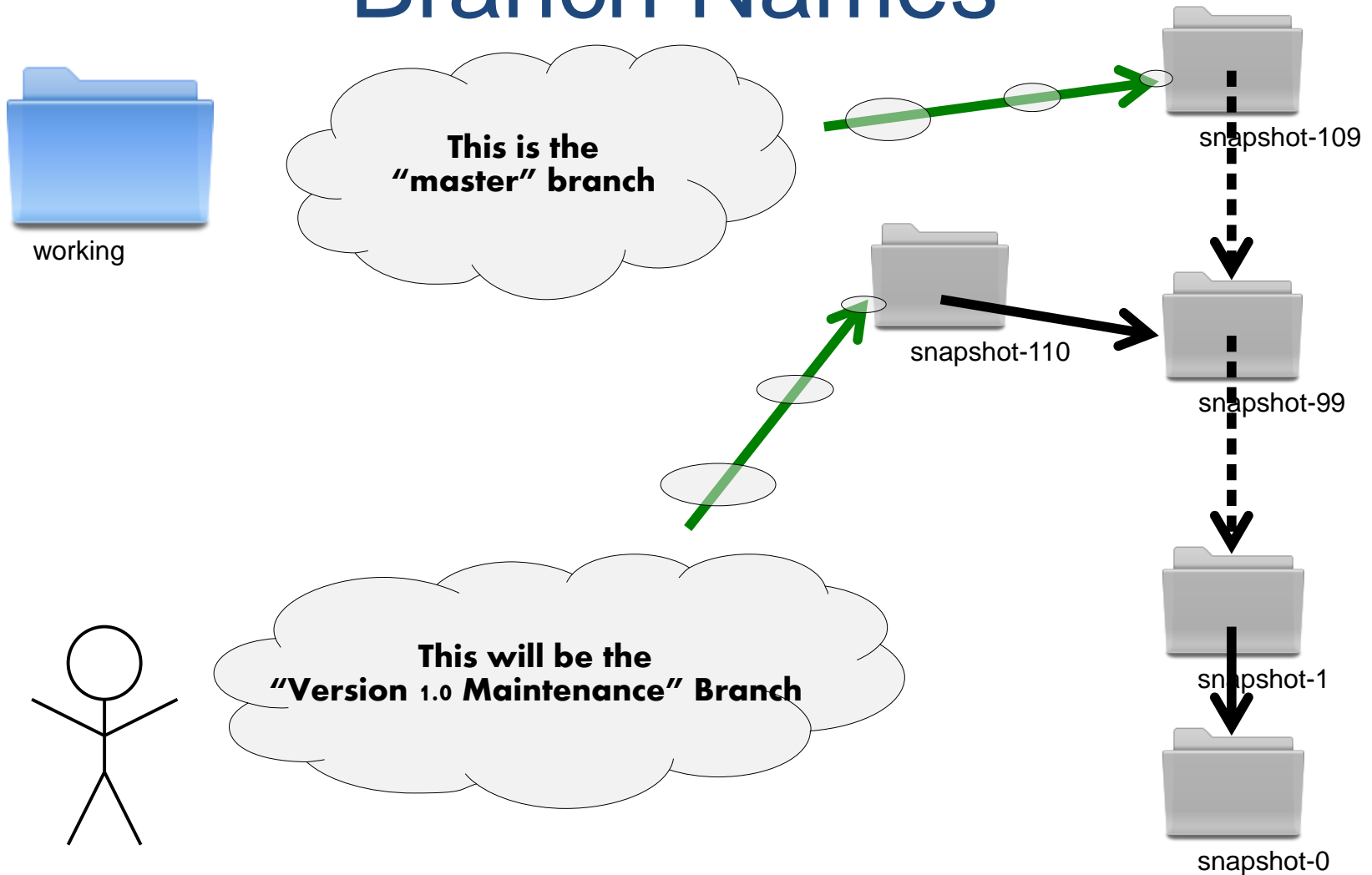
Branch Names



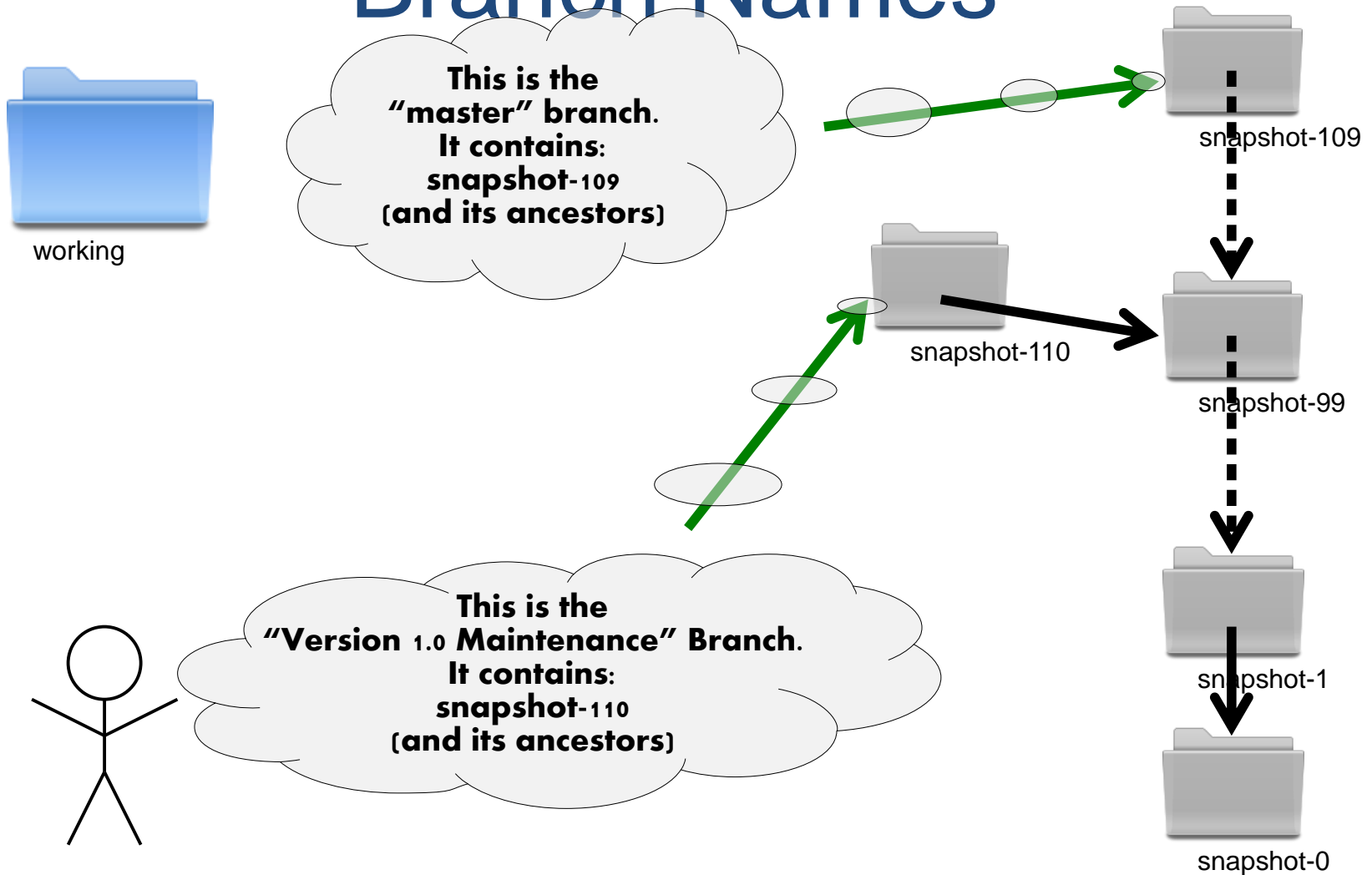
Branch Names



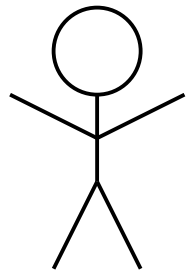
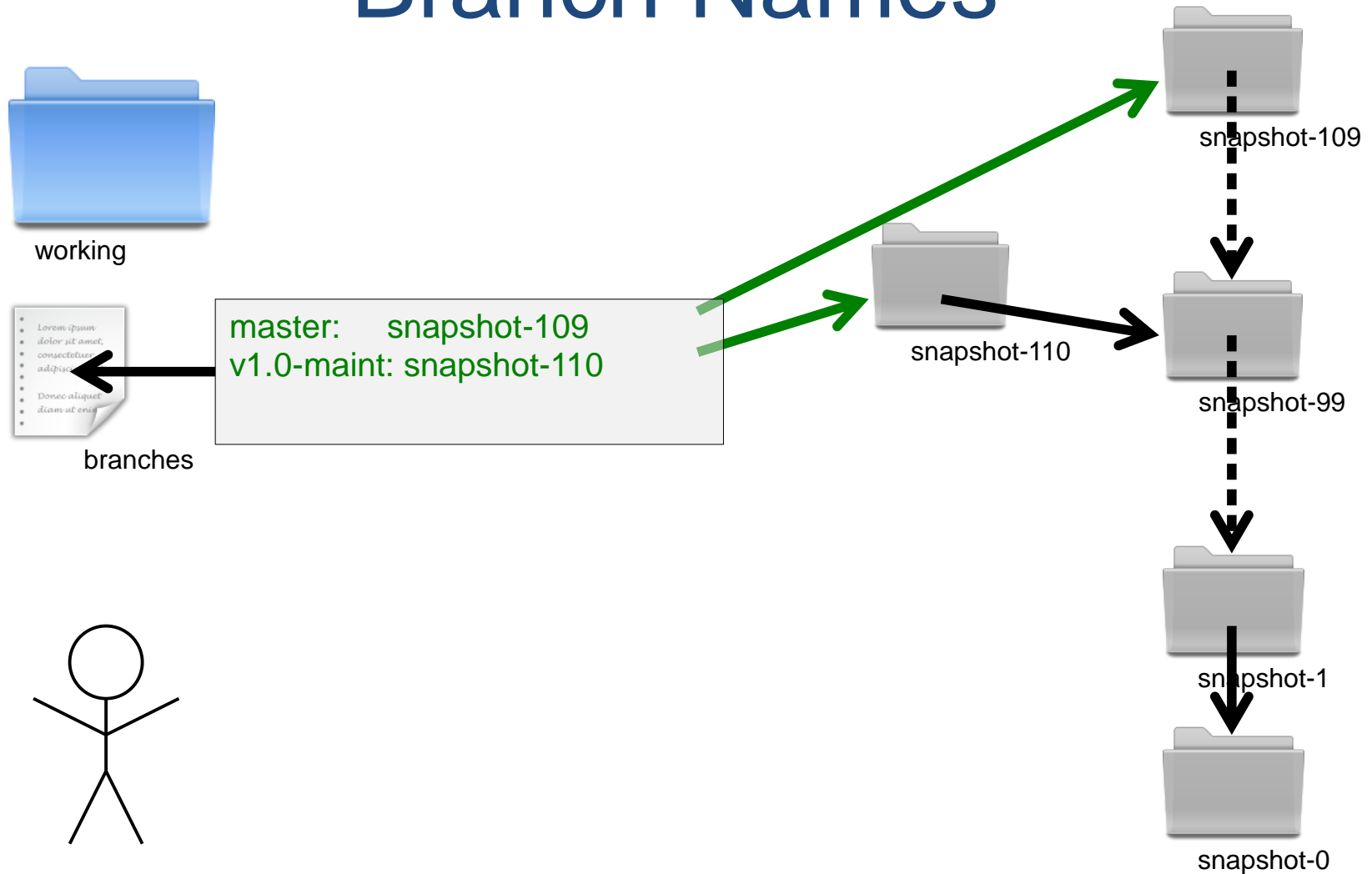
Branch Names



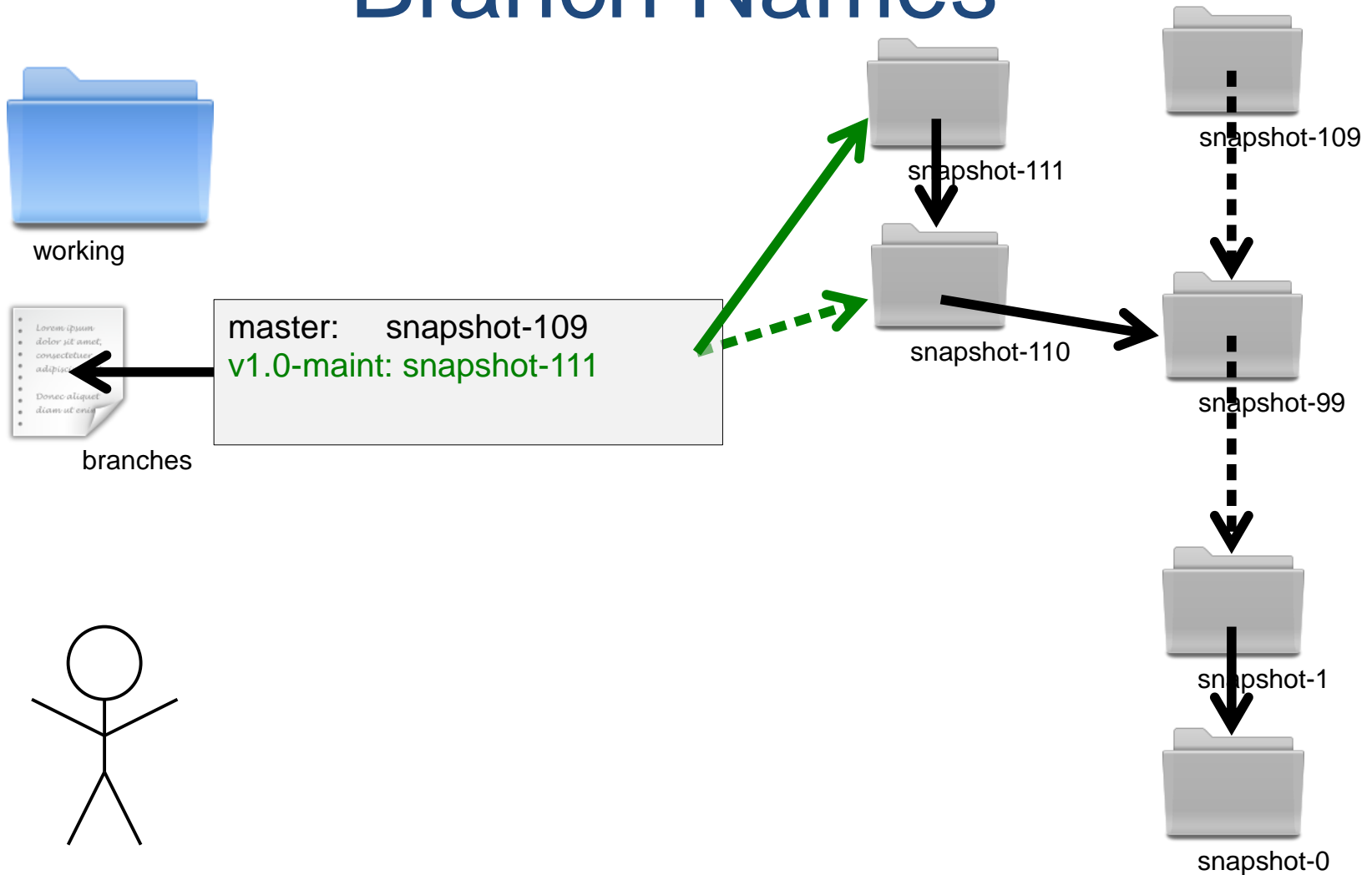
Branch Names



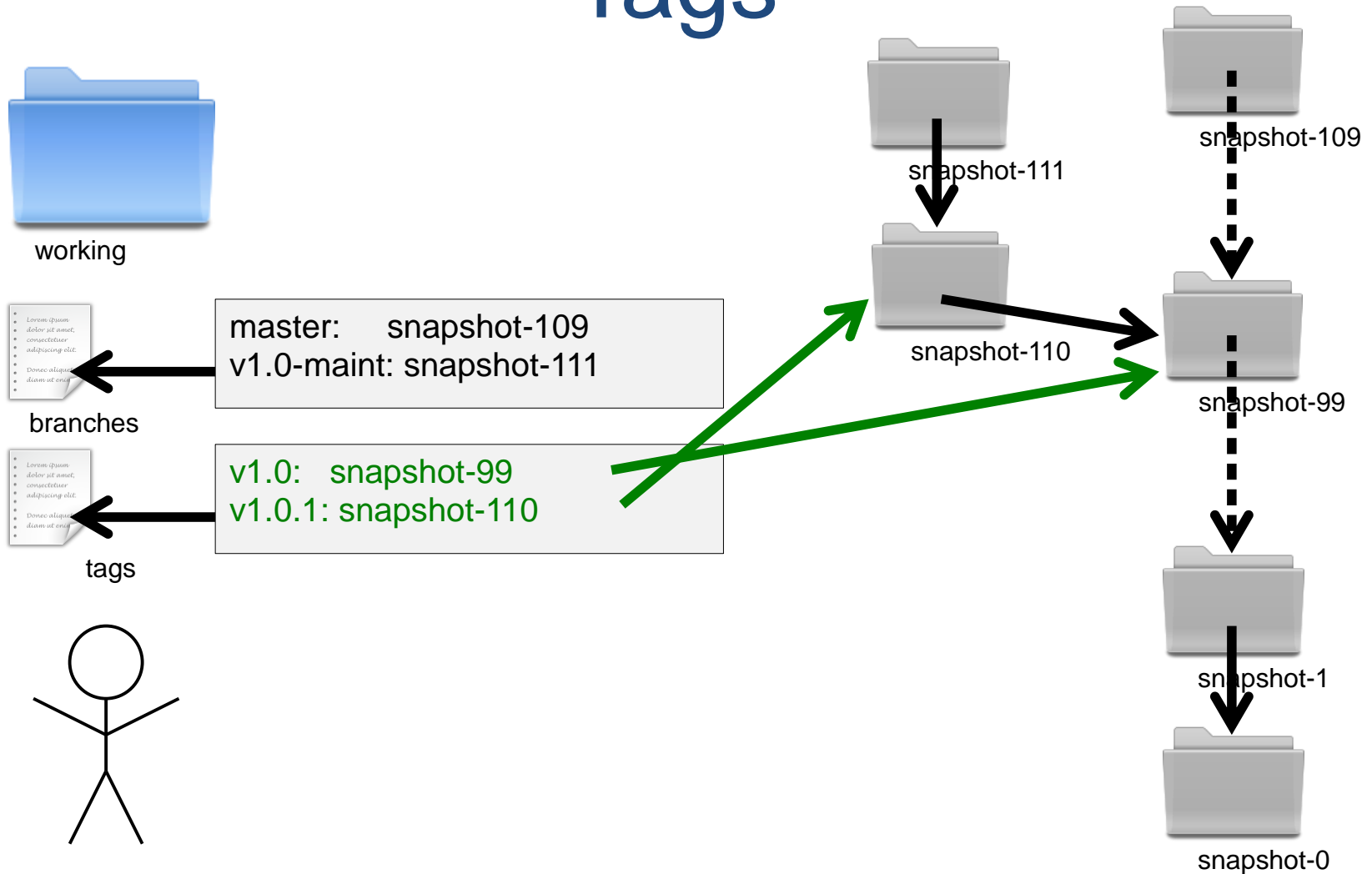
Branch Names



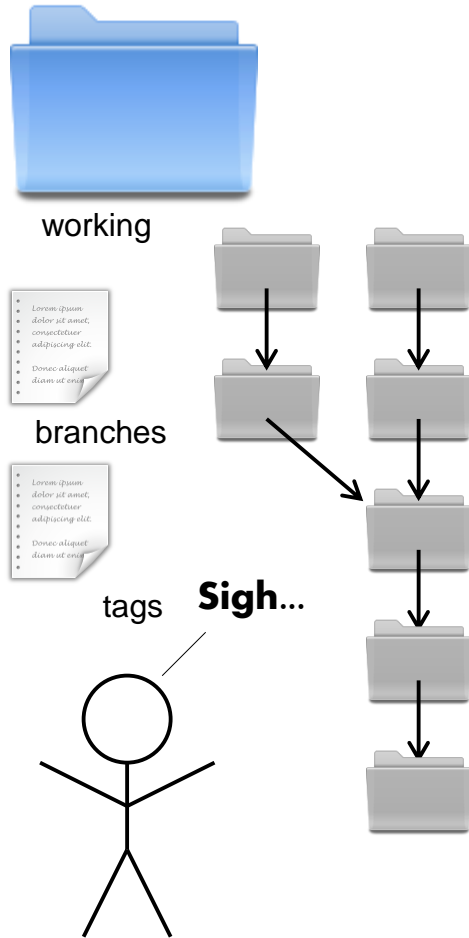
Branch Names



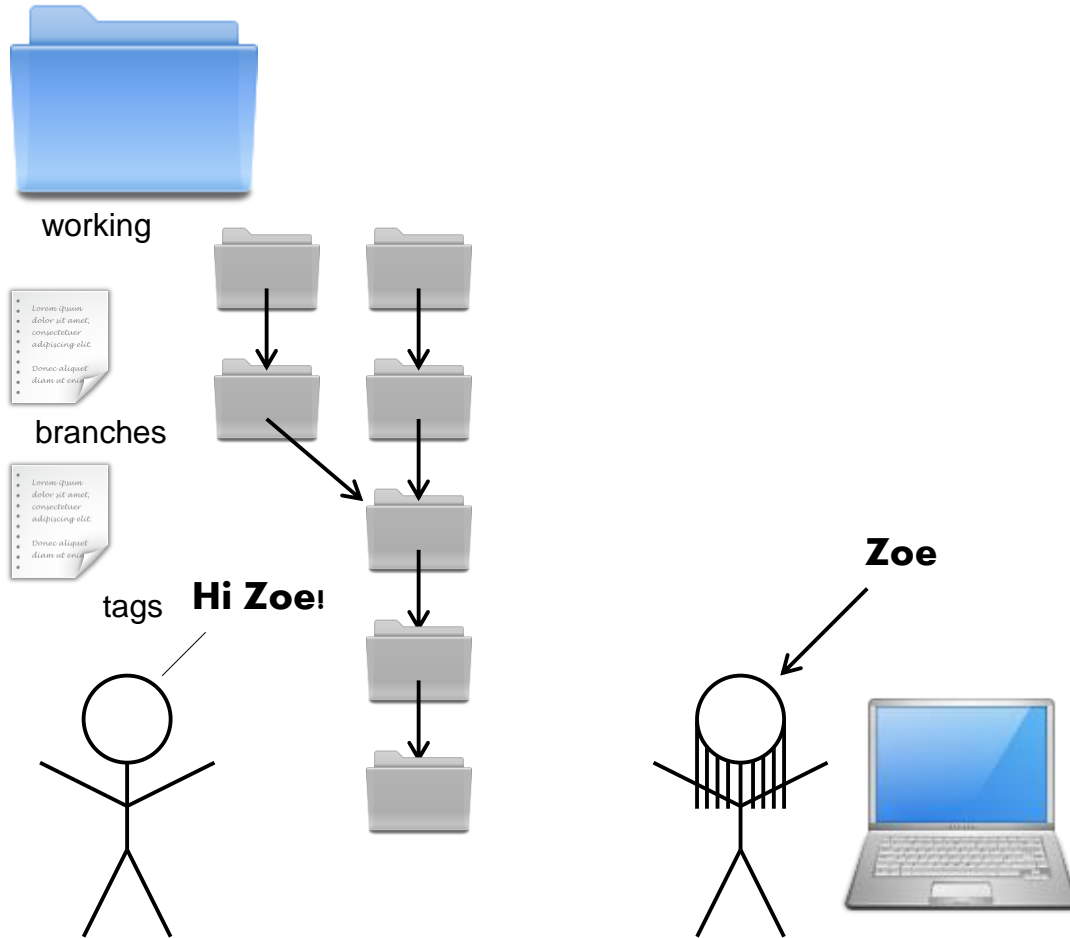
Tags



Distributed



Distributed



Distributed



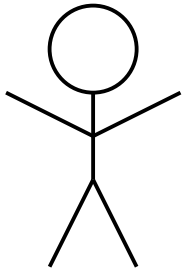
working



branches



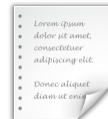
tags



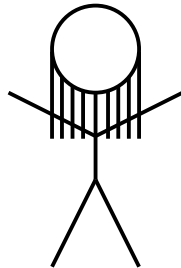
working



branches



tags



Distributed



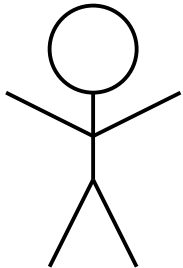
working



branches



tags



se15b-yagel



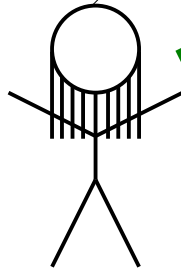
working



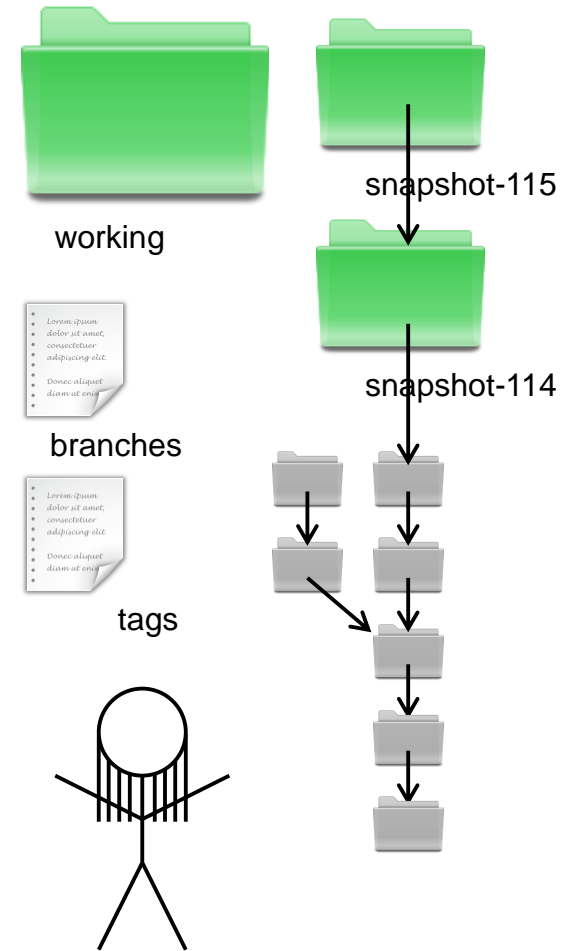
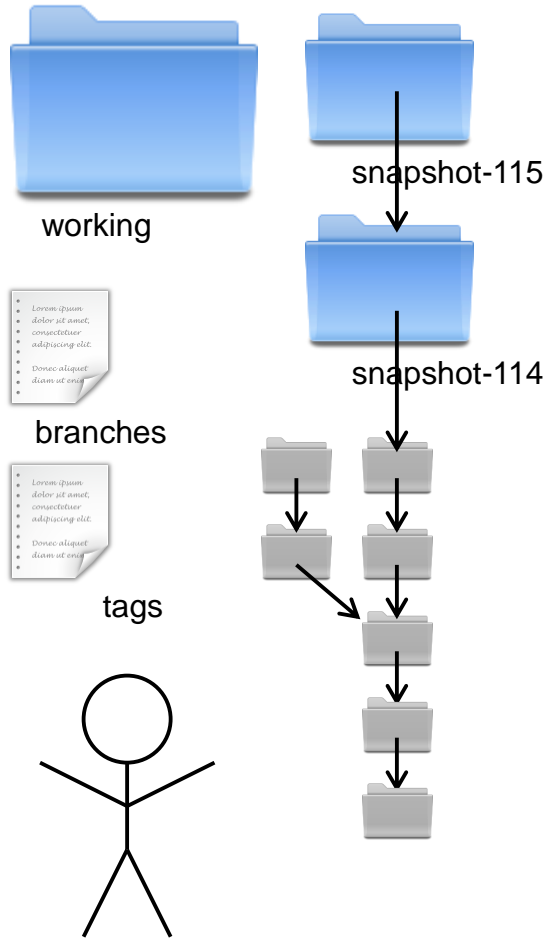
branches



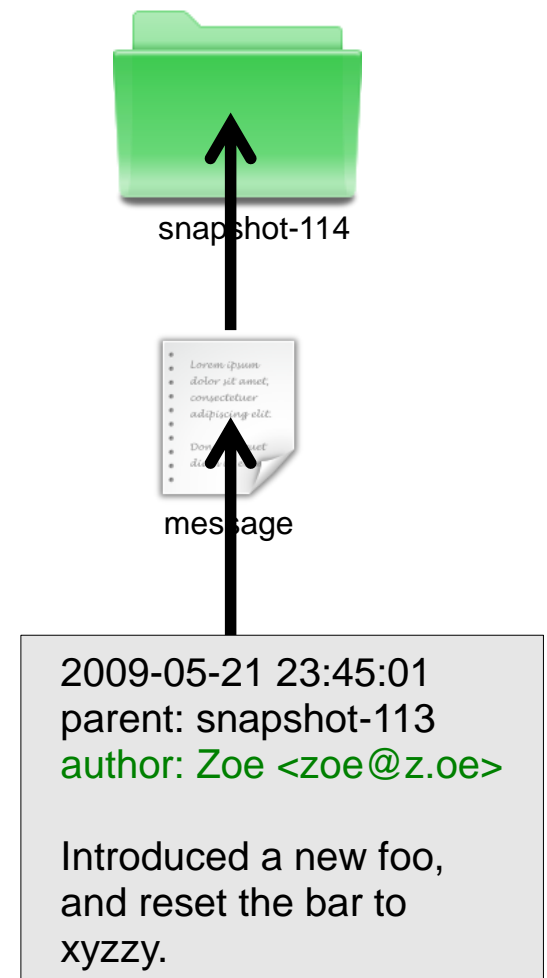
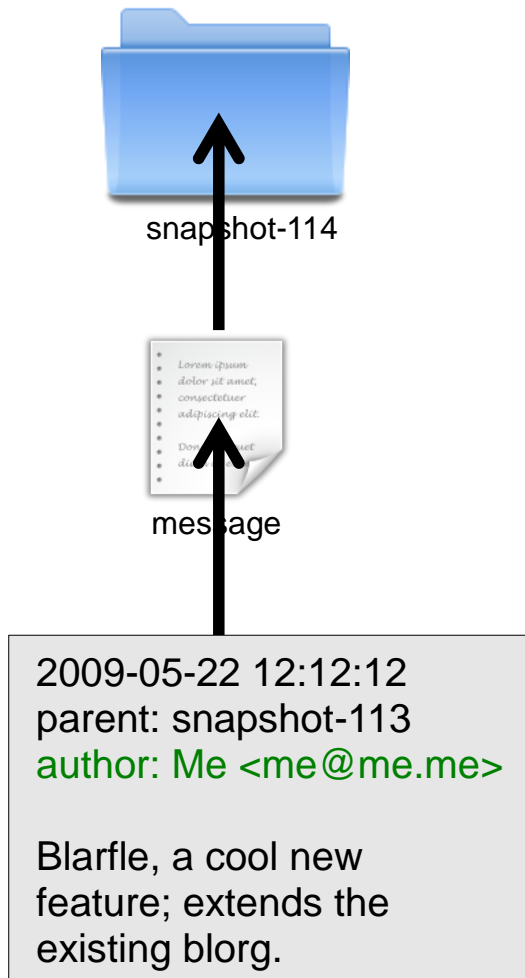
tags **Bye!**



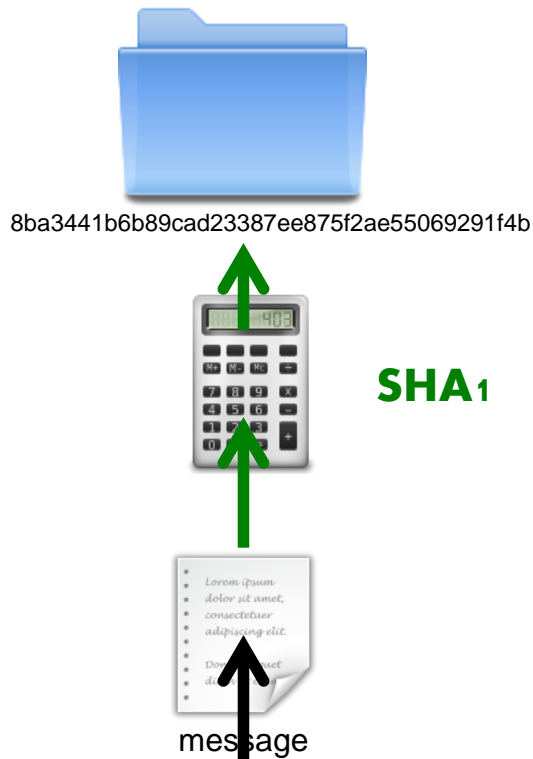
Distributed



Distributed



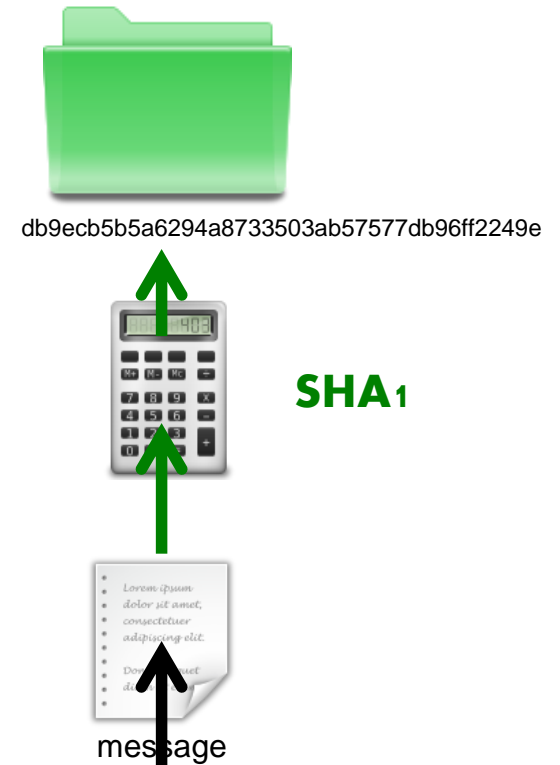
Distributed



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

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

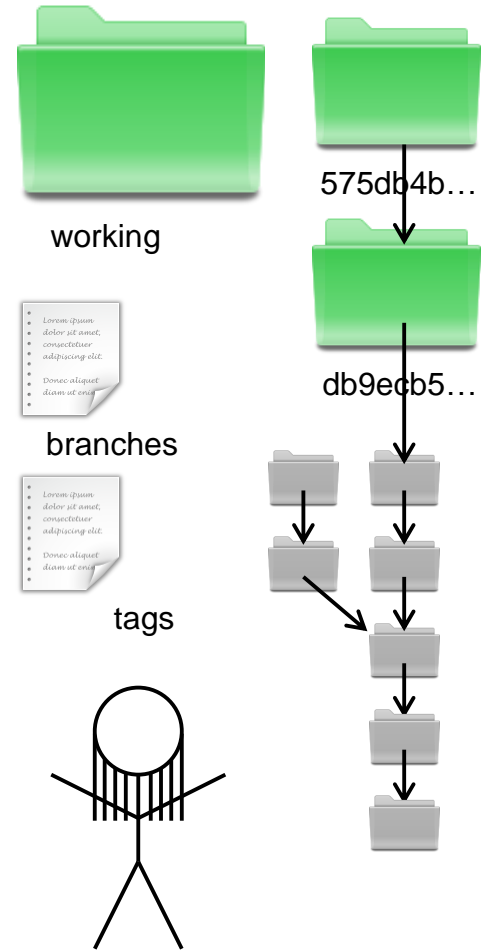
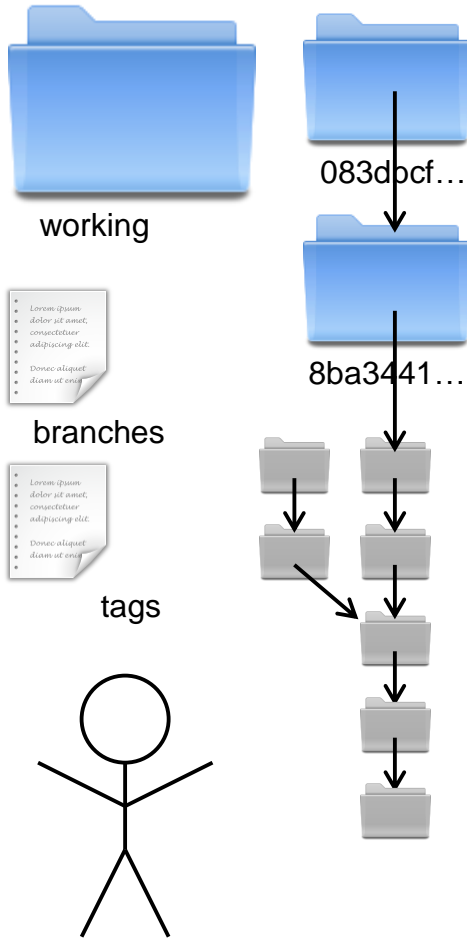
se15b-yagel



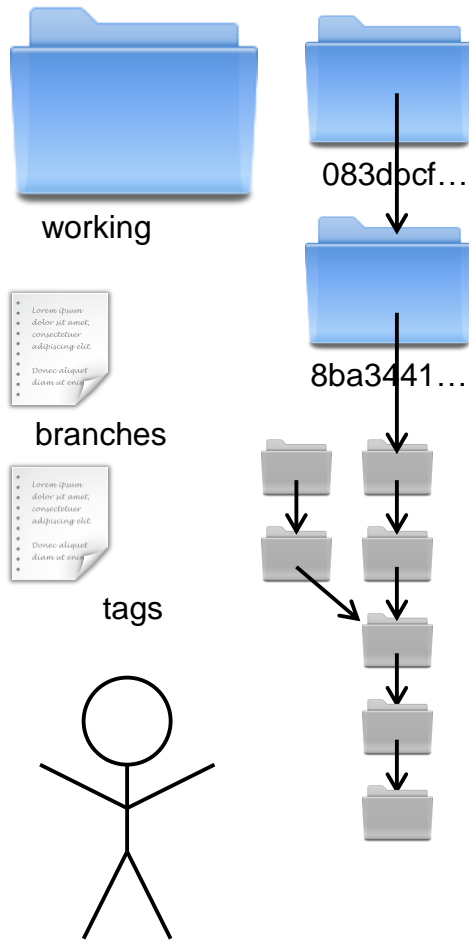
2009-05-21 23:45:01
parent: snapshot-113
author: Zoe <zoe@z.oe>

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

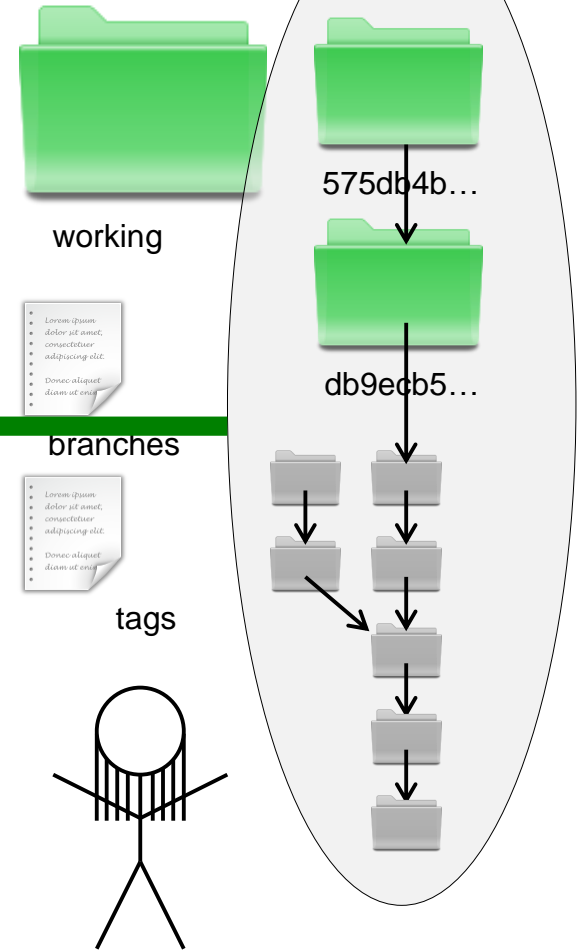
Distributed



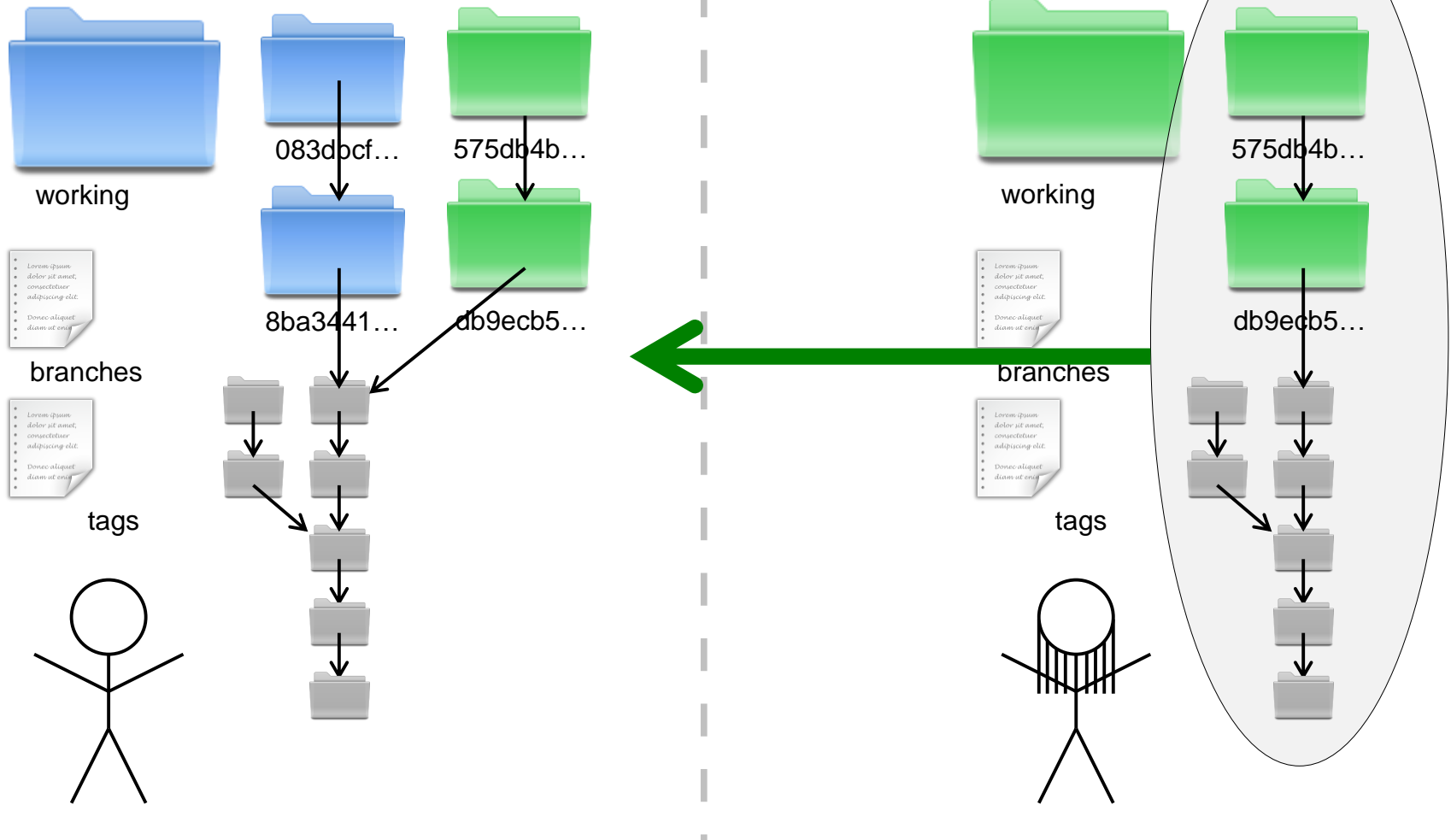
Distributed



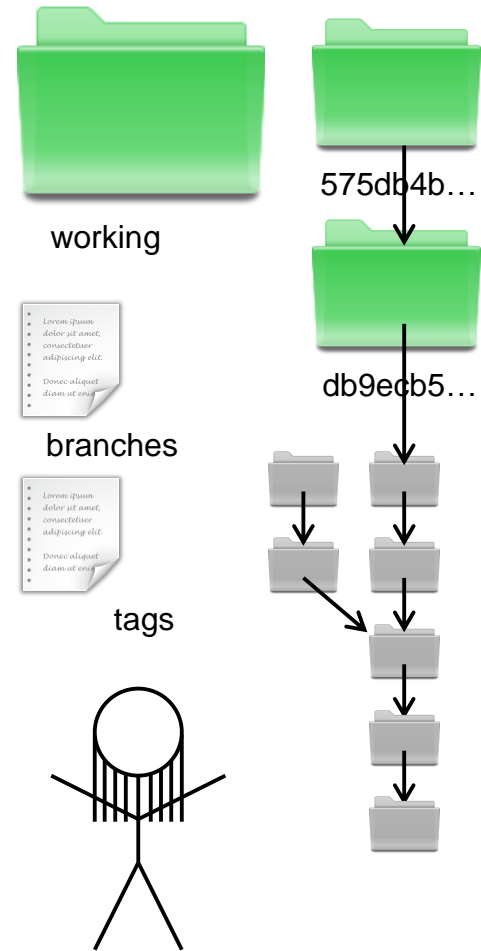
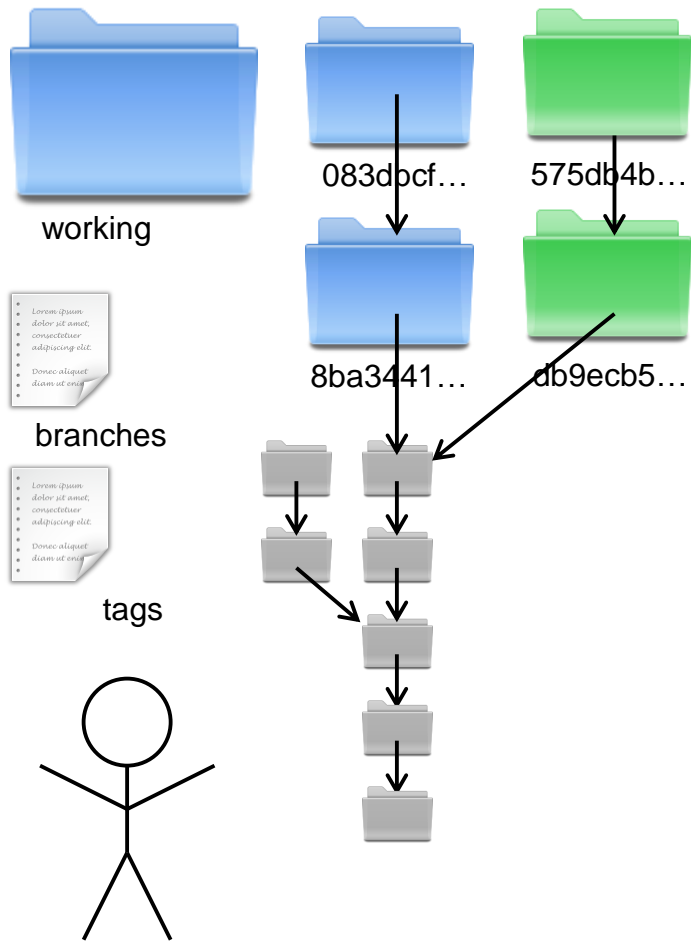
?



Distributed



Distributed



Offline



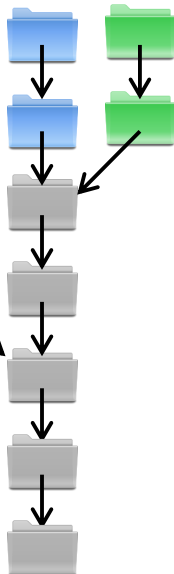
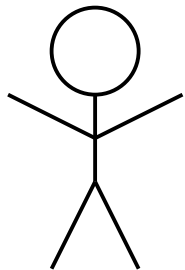
working



branches



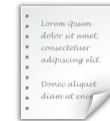
tags



working

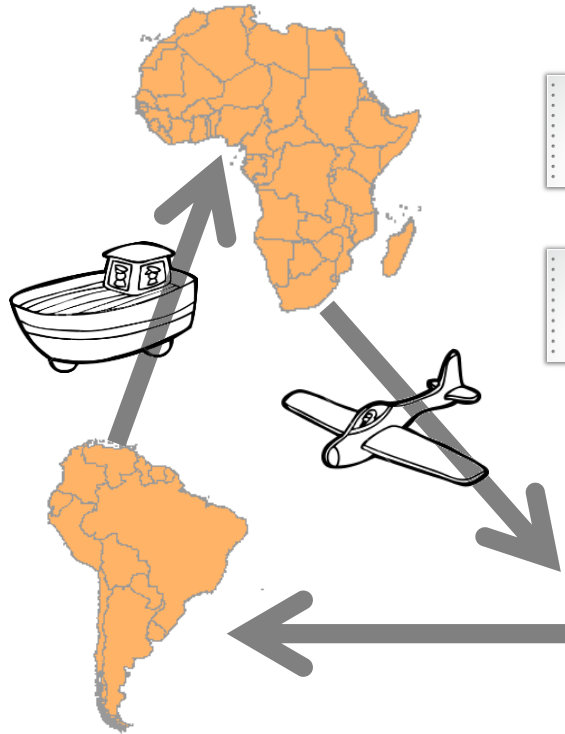
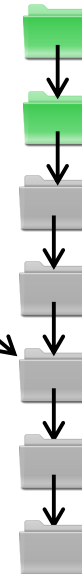
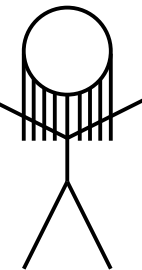


branches

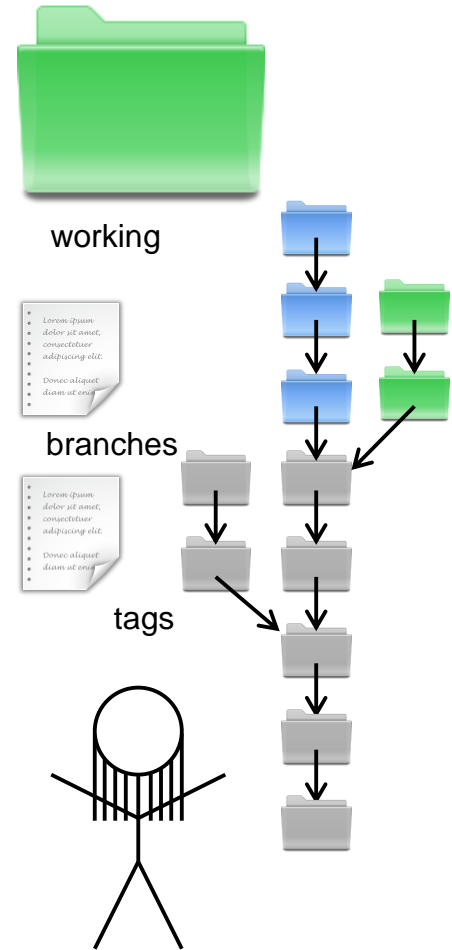
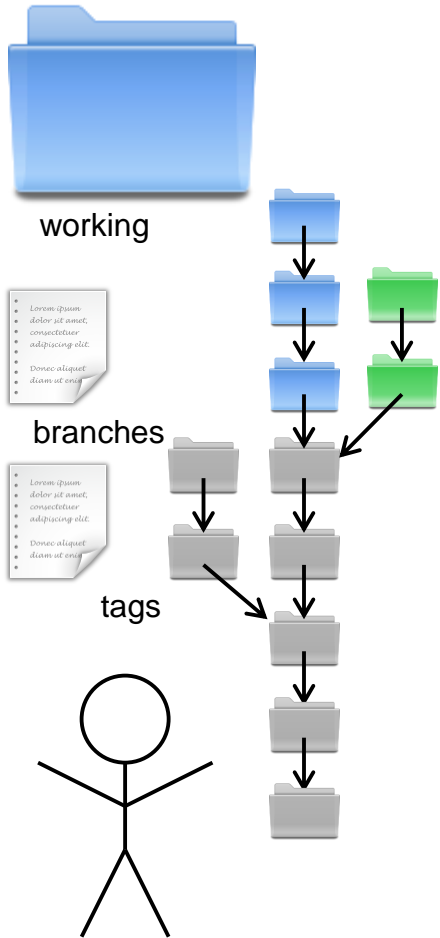


tags

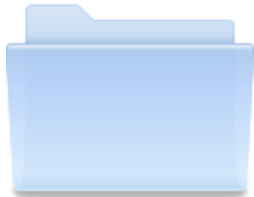
woot!



Offline



(simpler drawings)



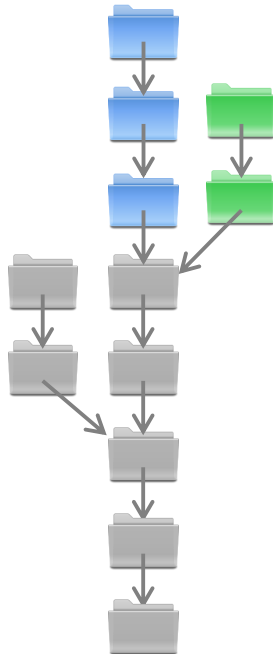
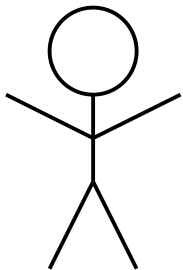
working



branches



tags



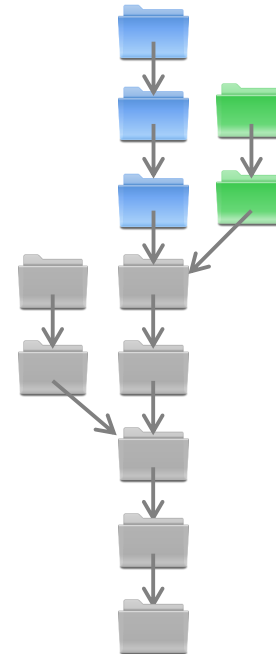
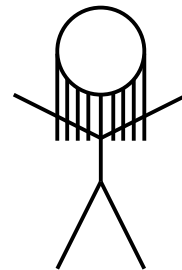
working



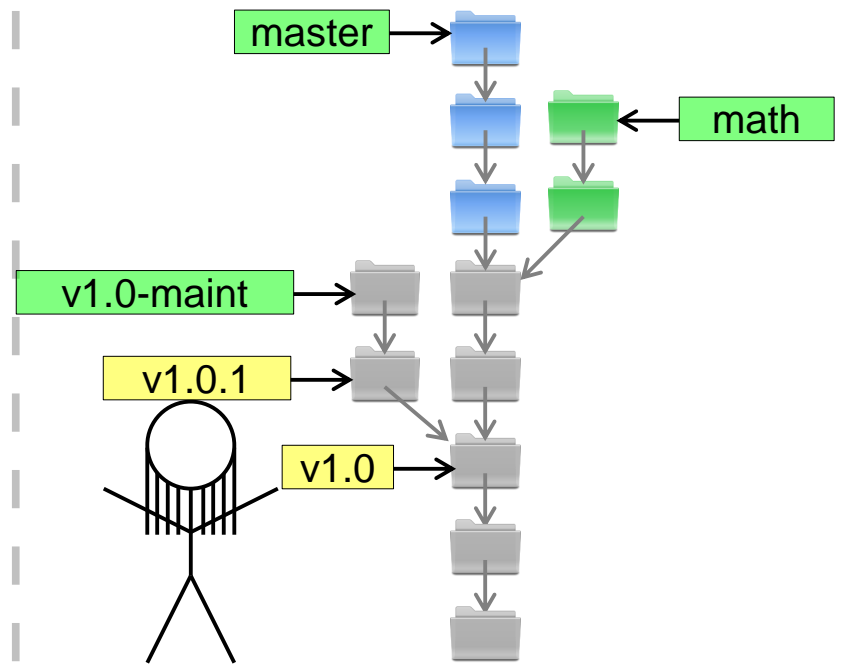
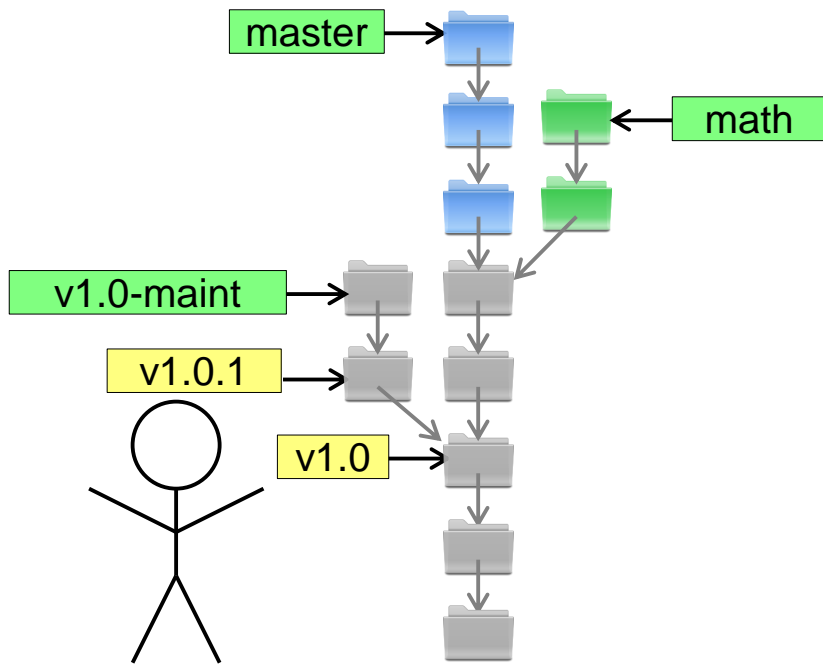
branches



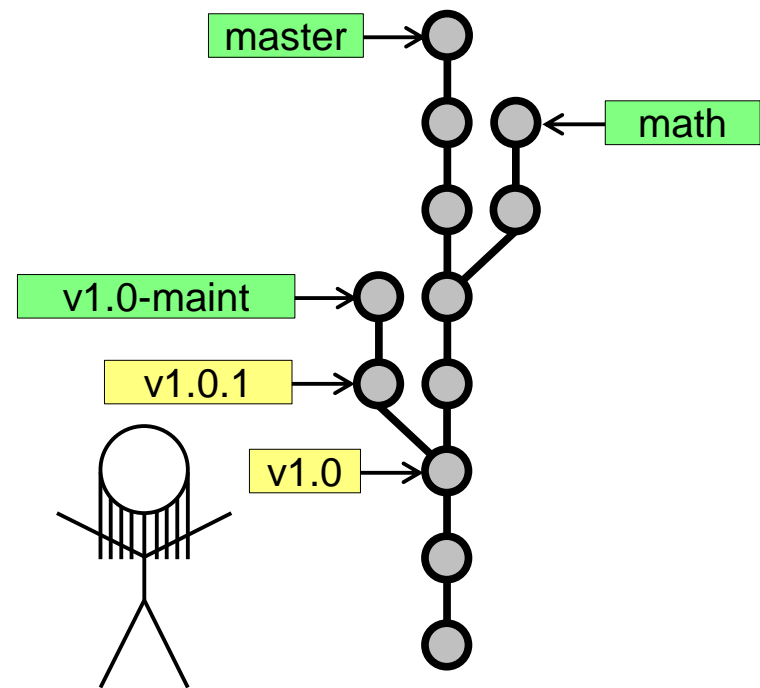
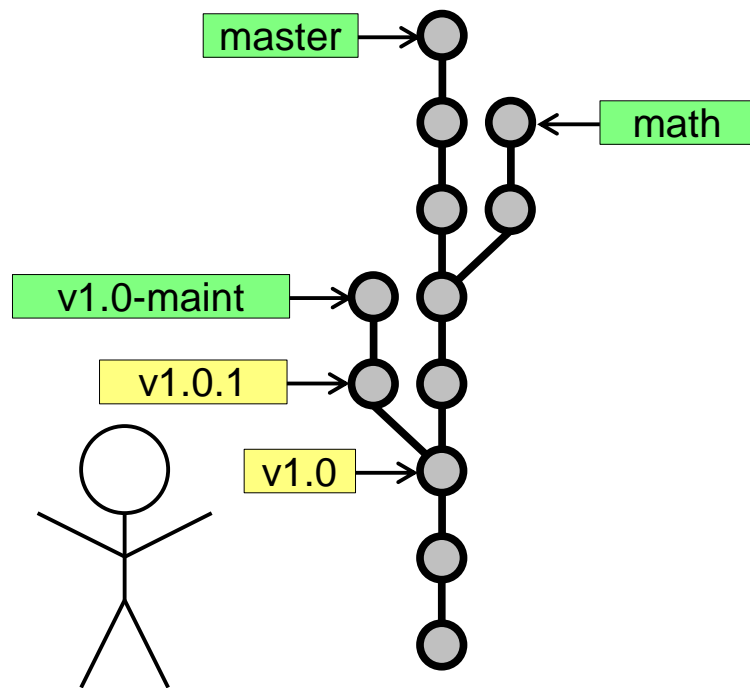
tags



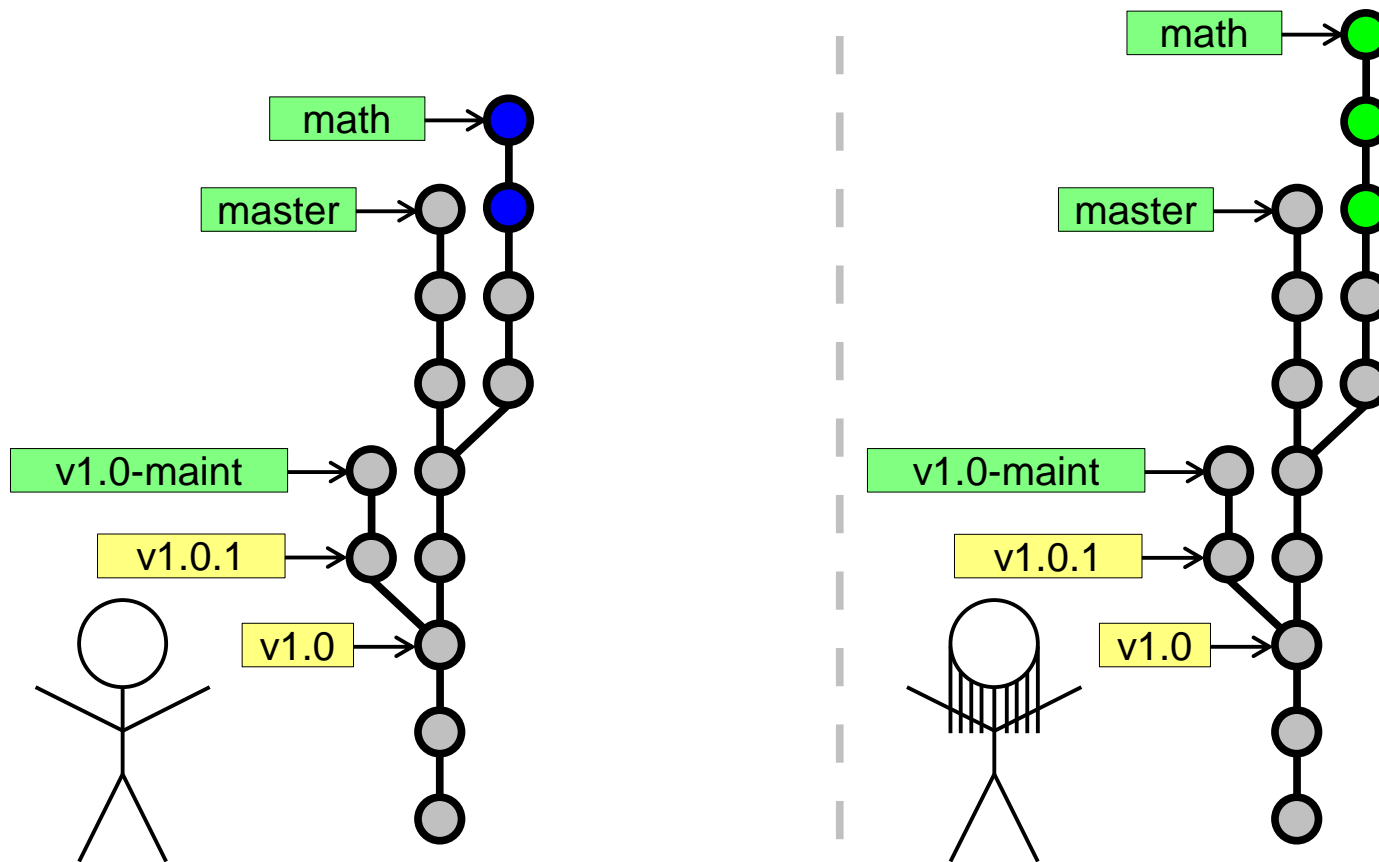
(simpler drawings)



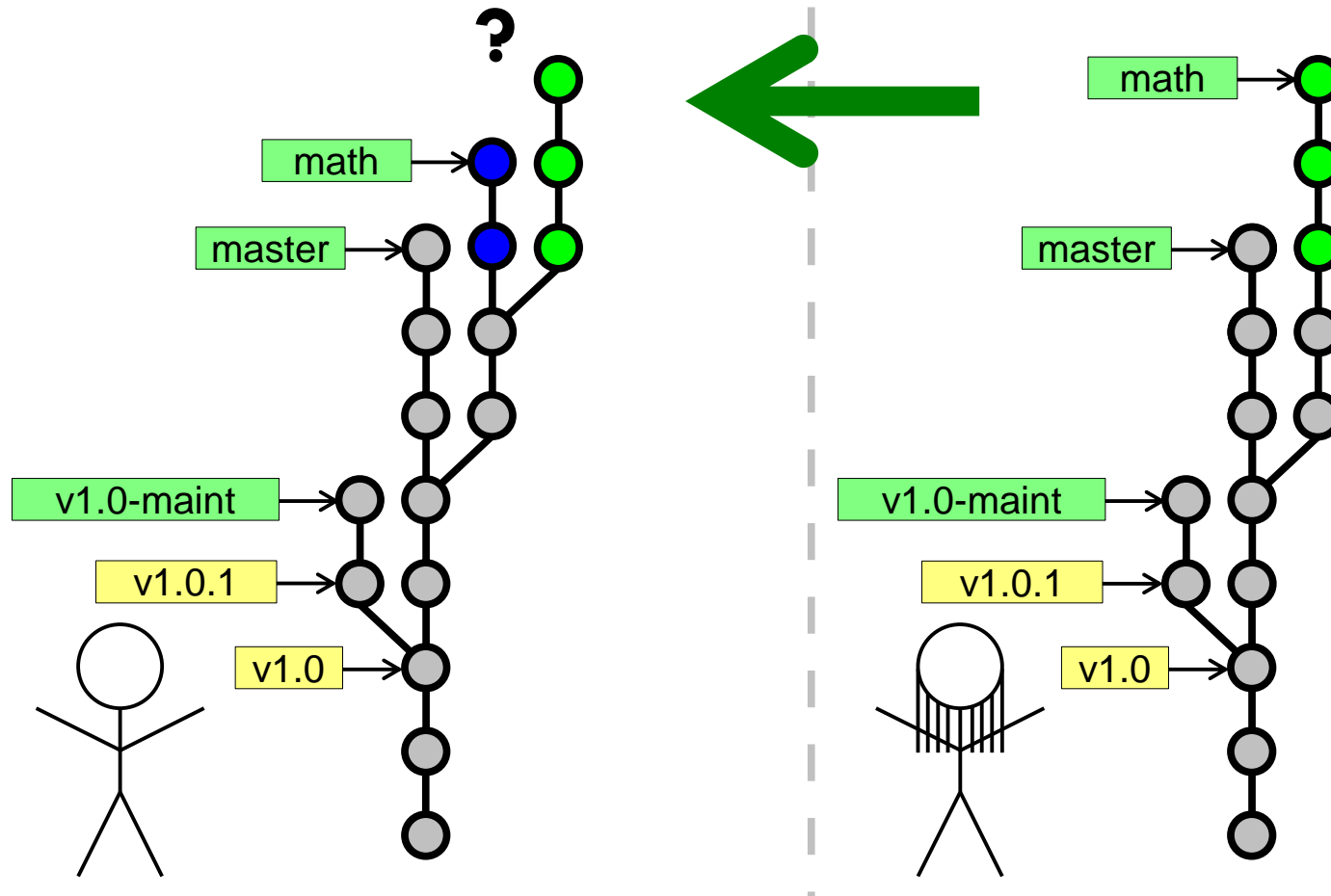
(simpler drawings)



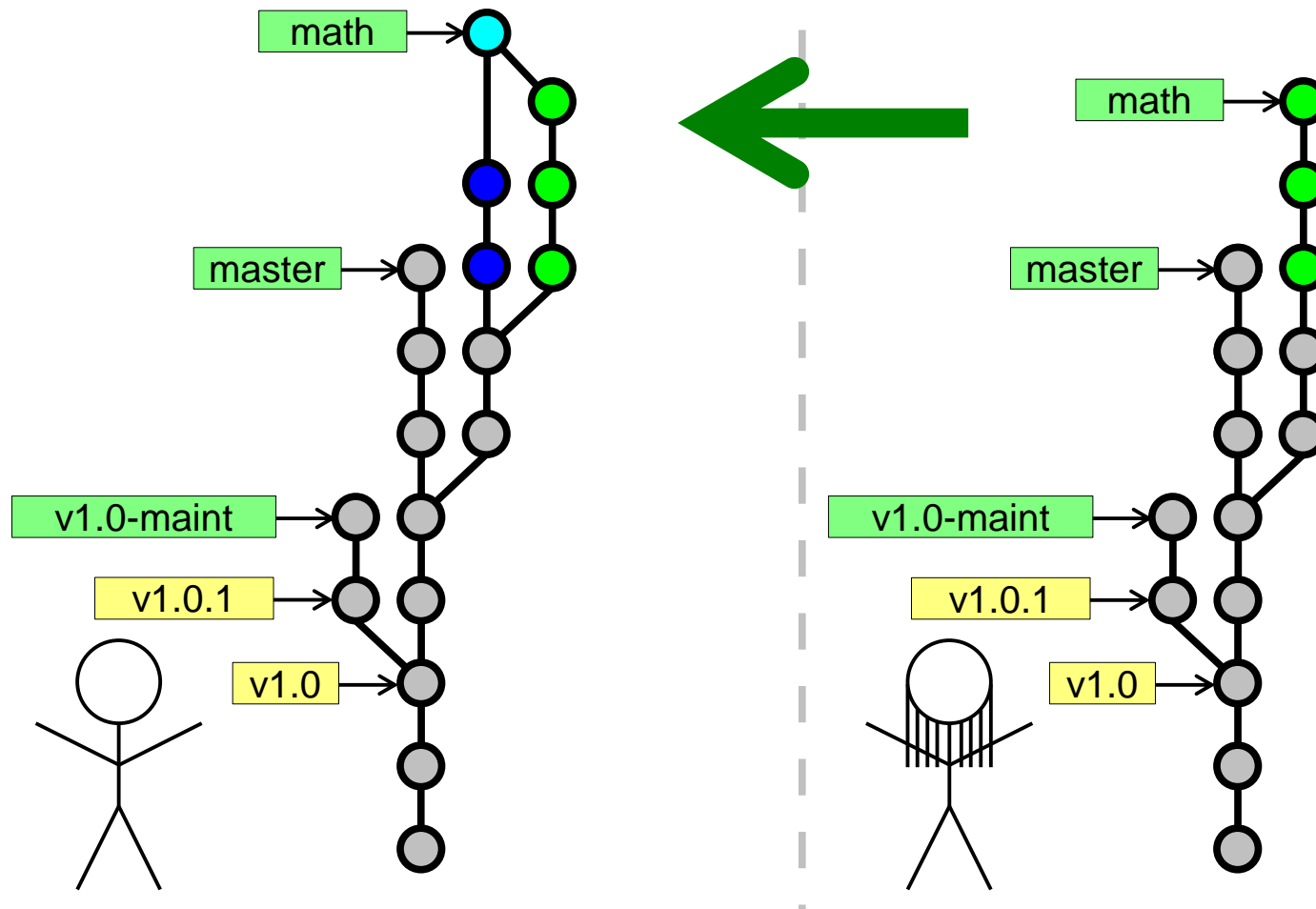
Merges



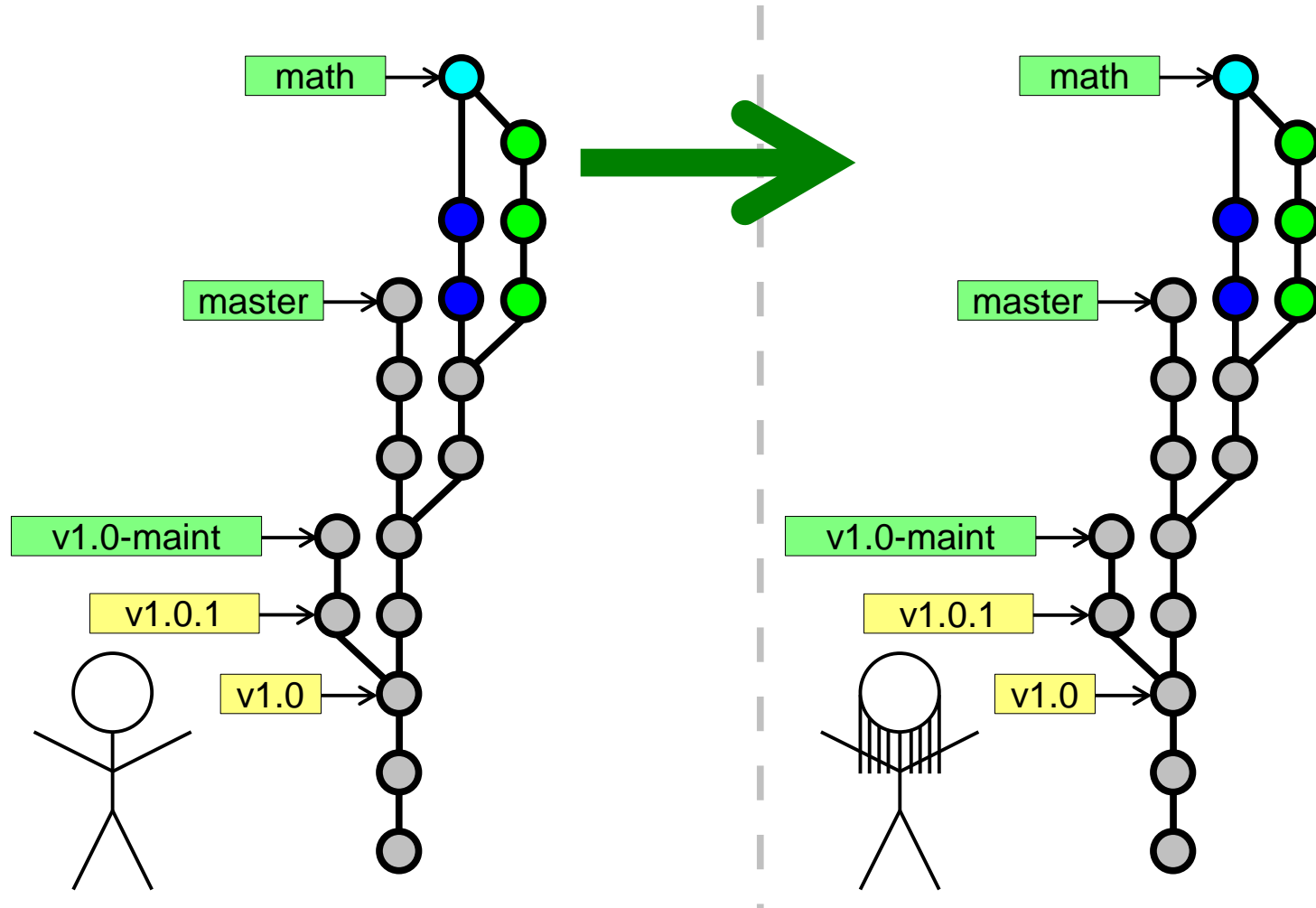
Merges



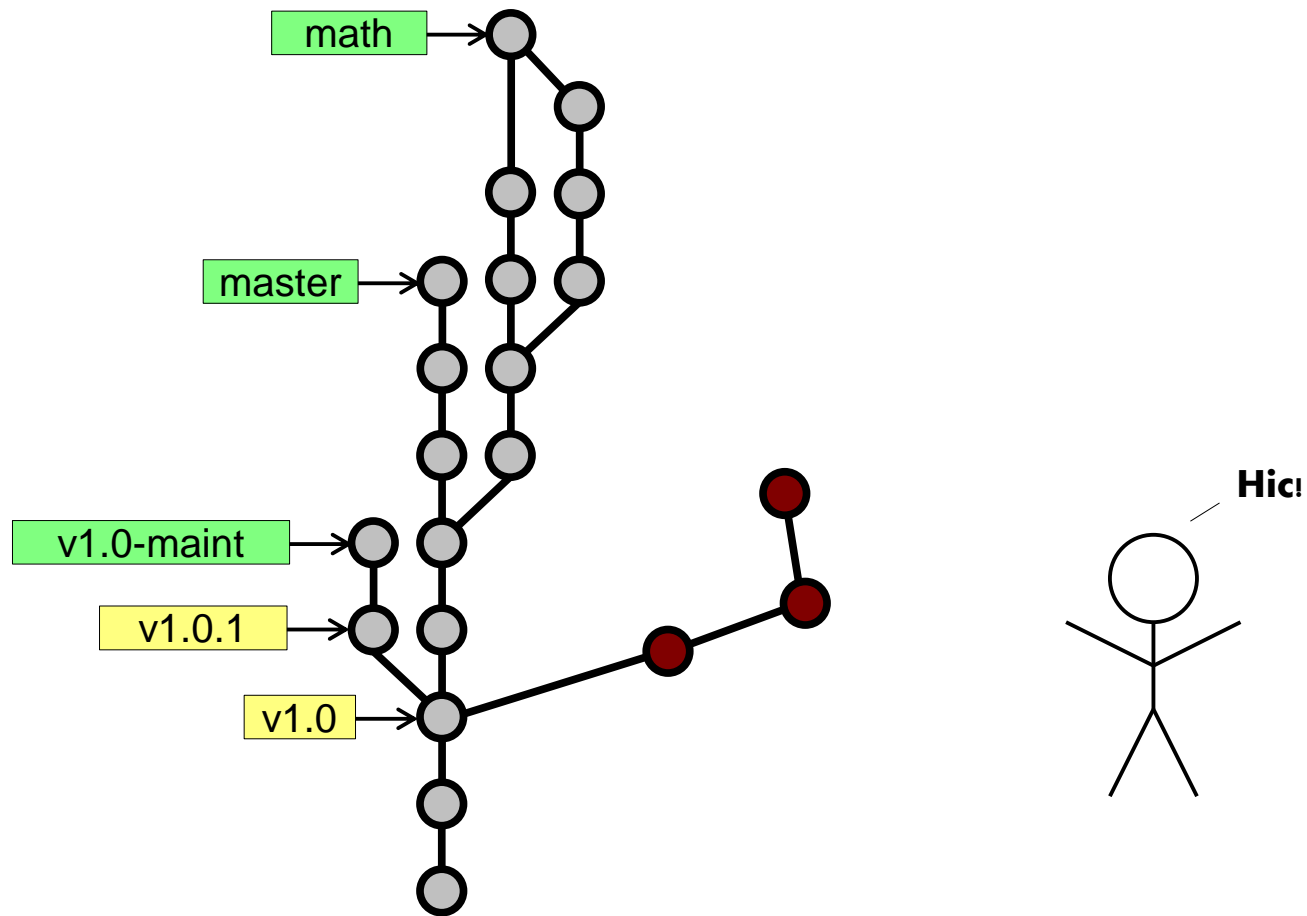
Merges



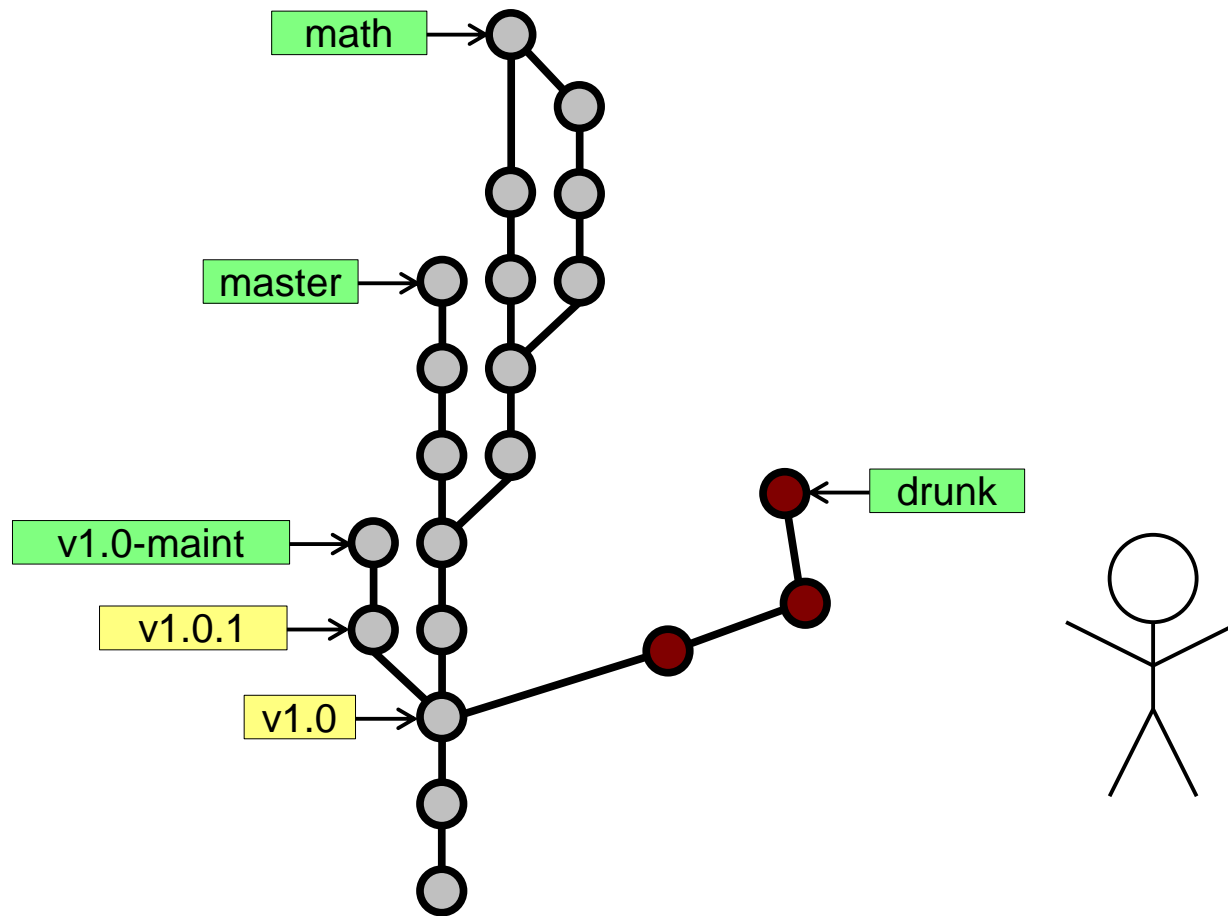
Merges



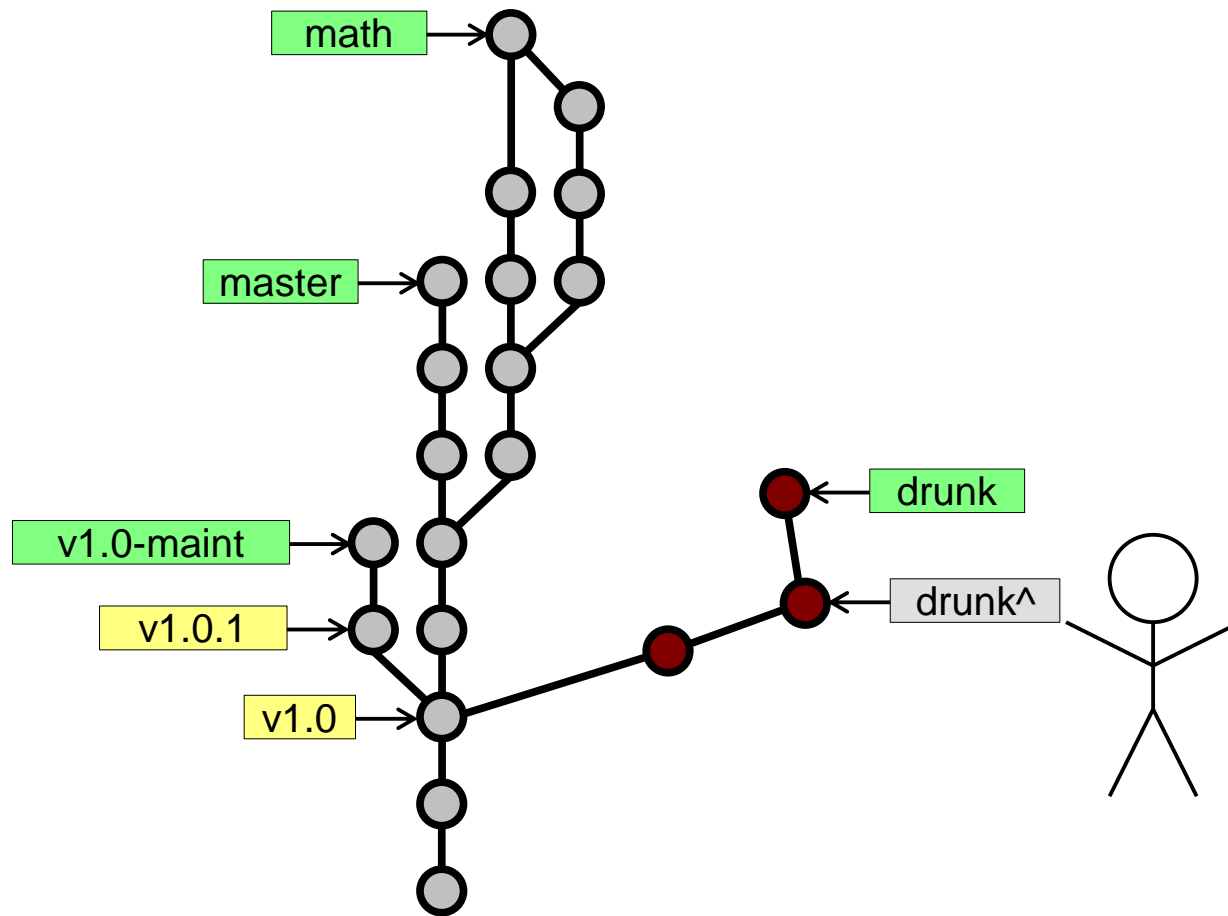
Rewriting History



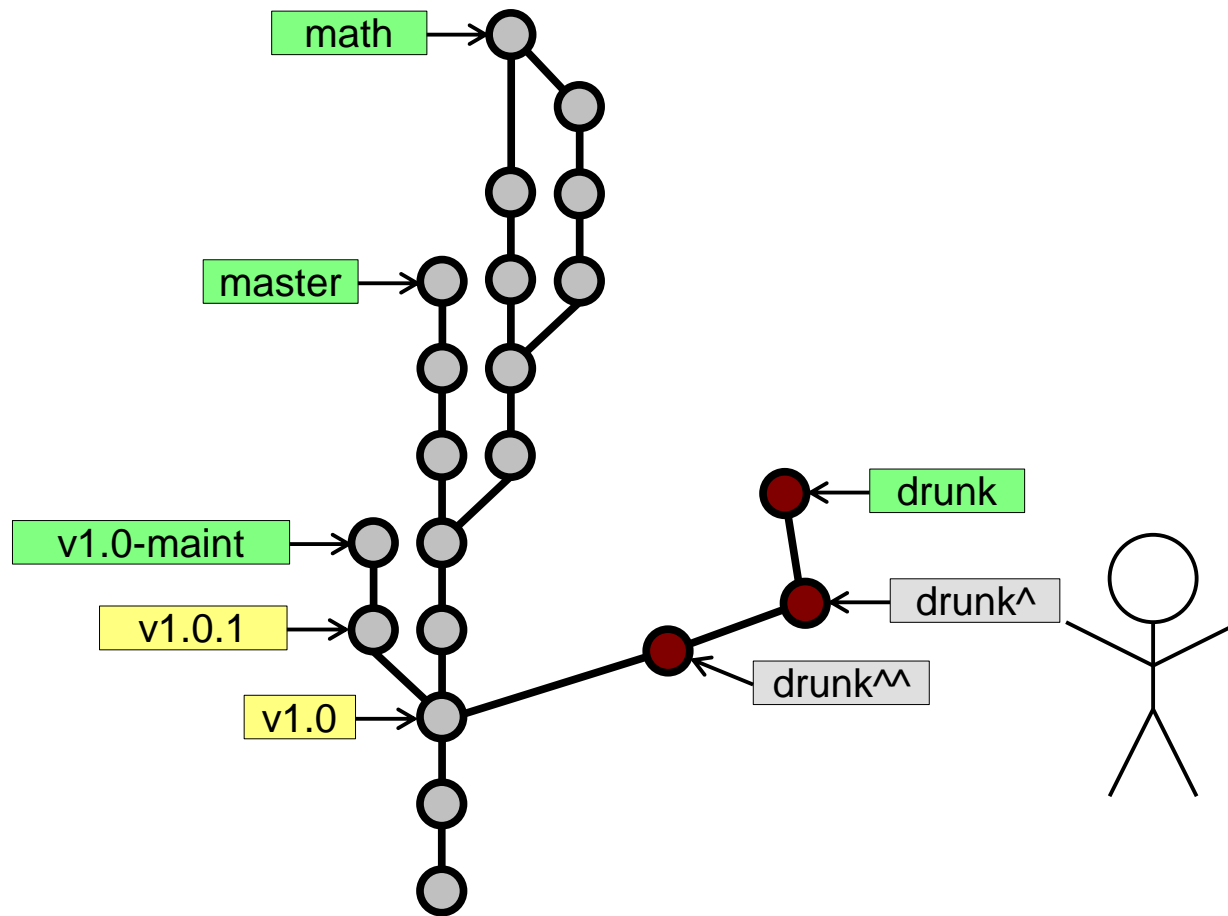
Rewriting History



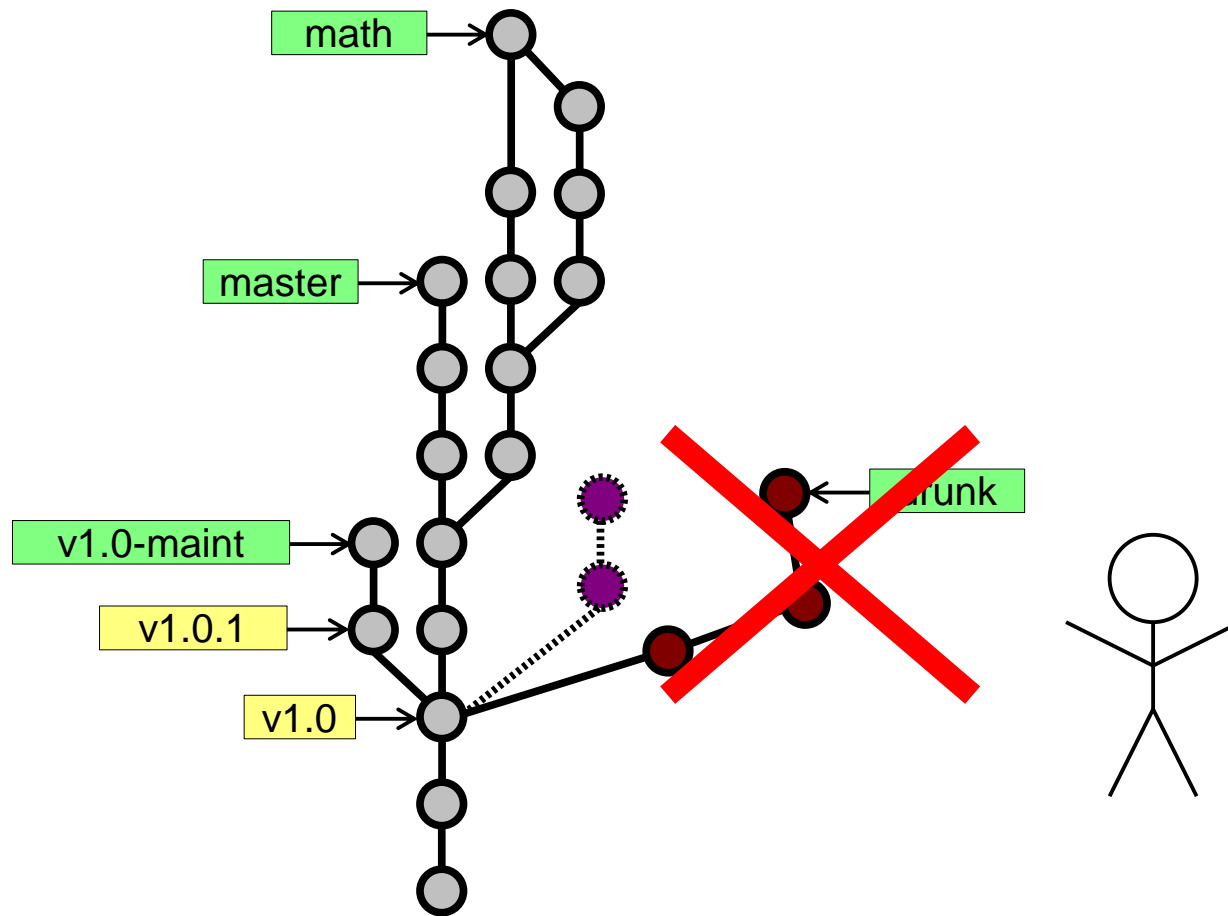
Rewriting History



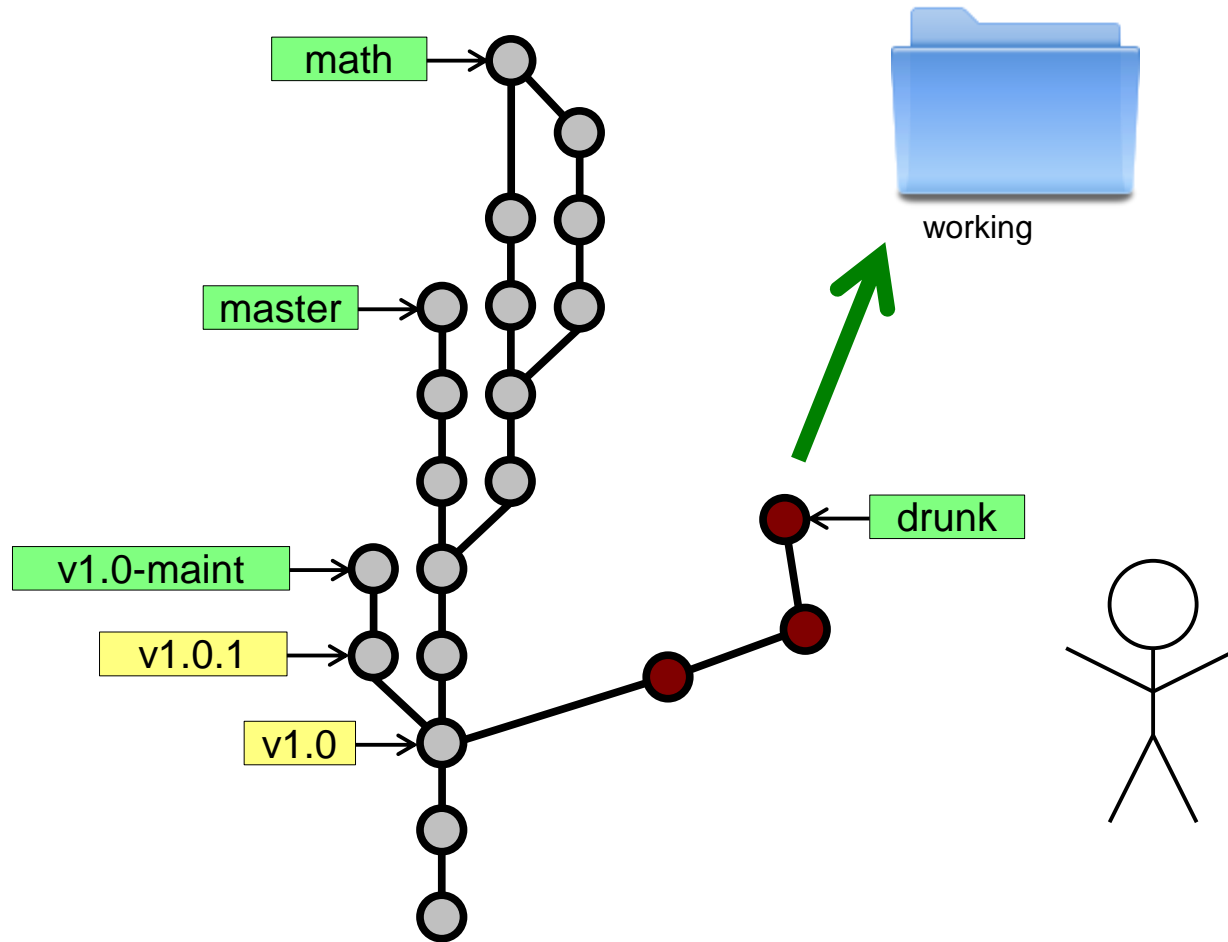
Rewriting History



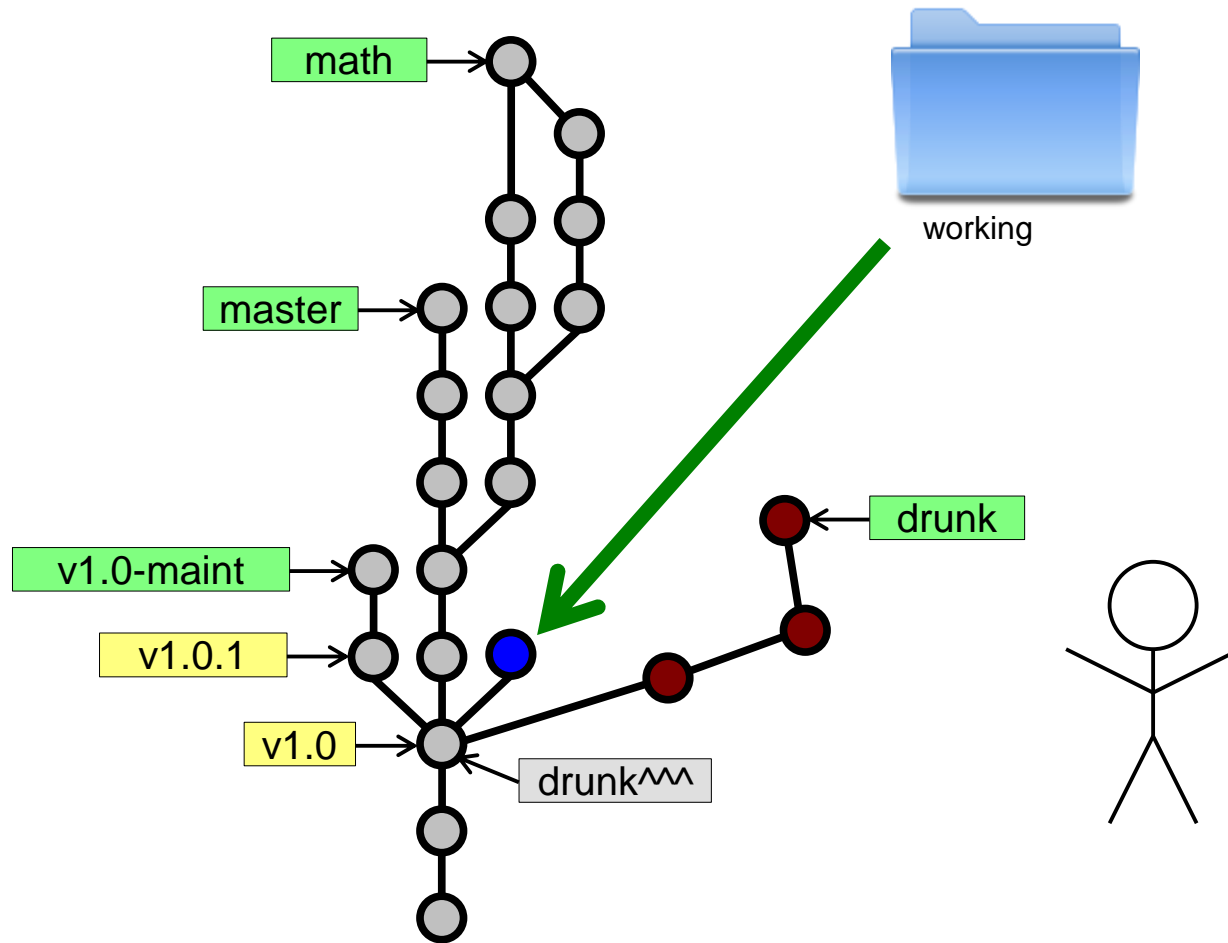
Rewriting History



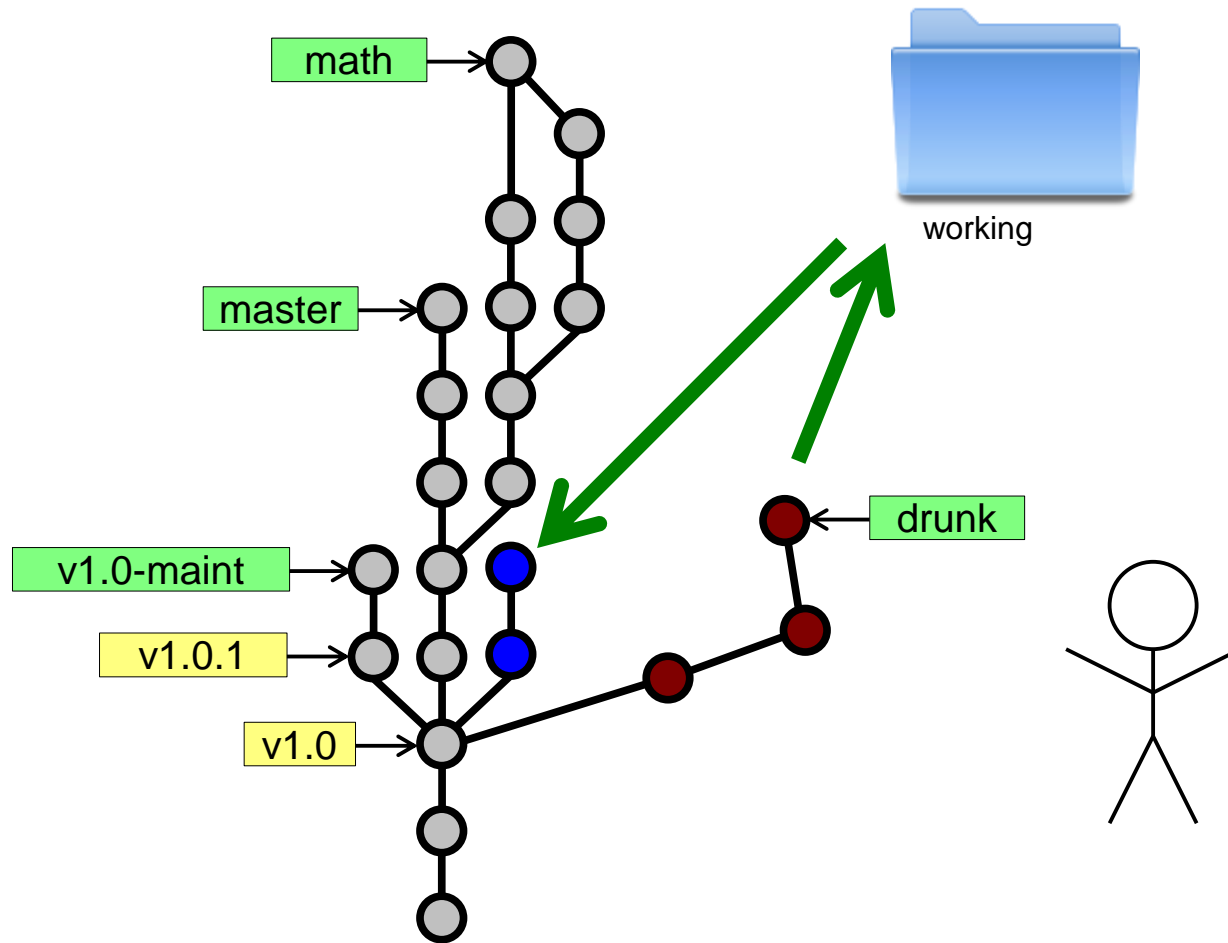
Rewriting History



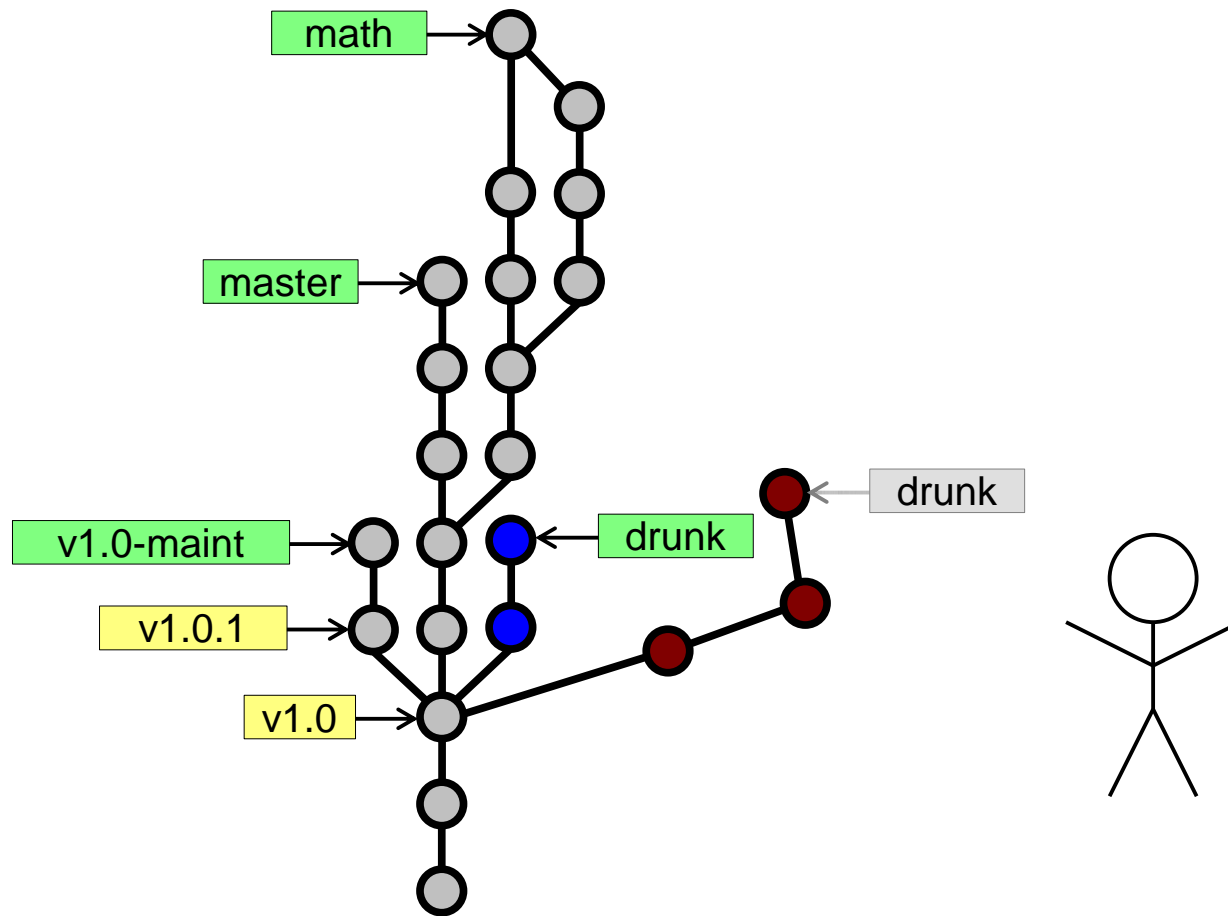
Rewriting History



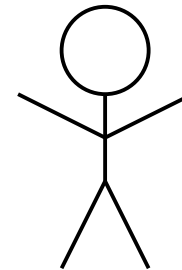
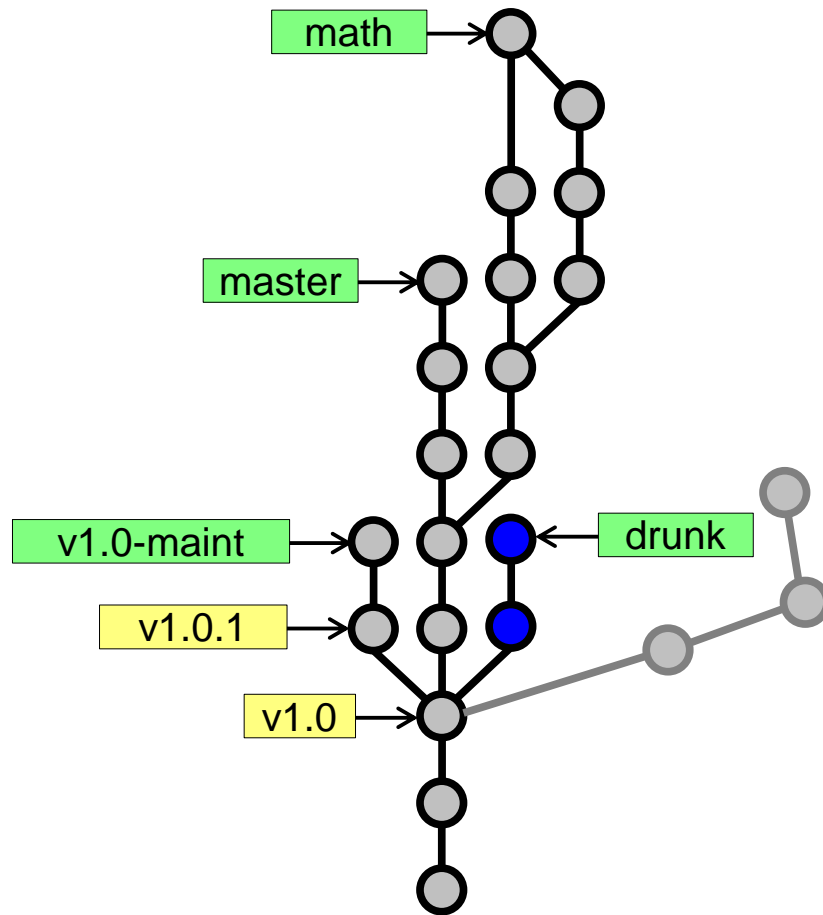
Rewriting History



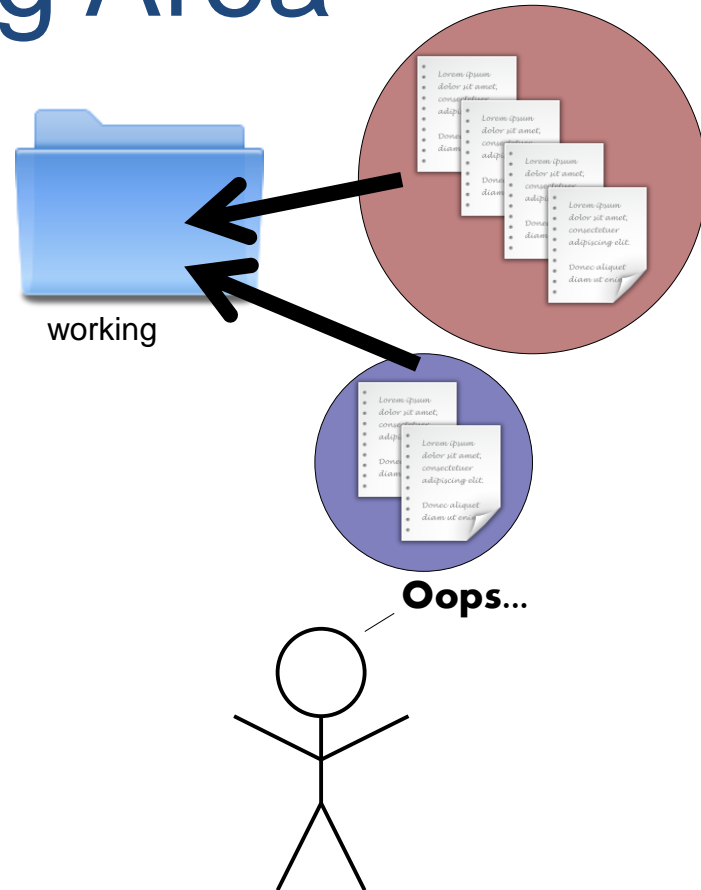
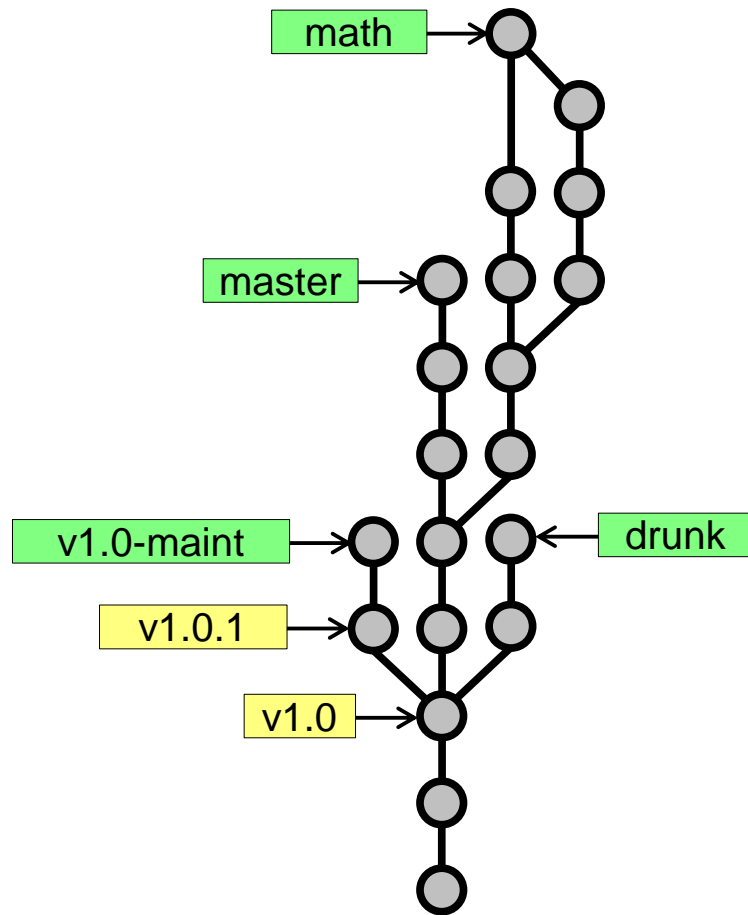
Rewriting History



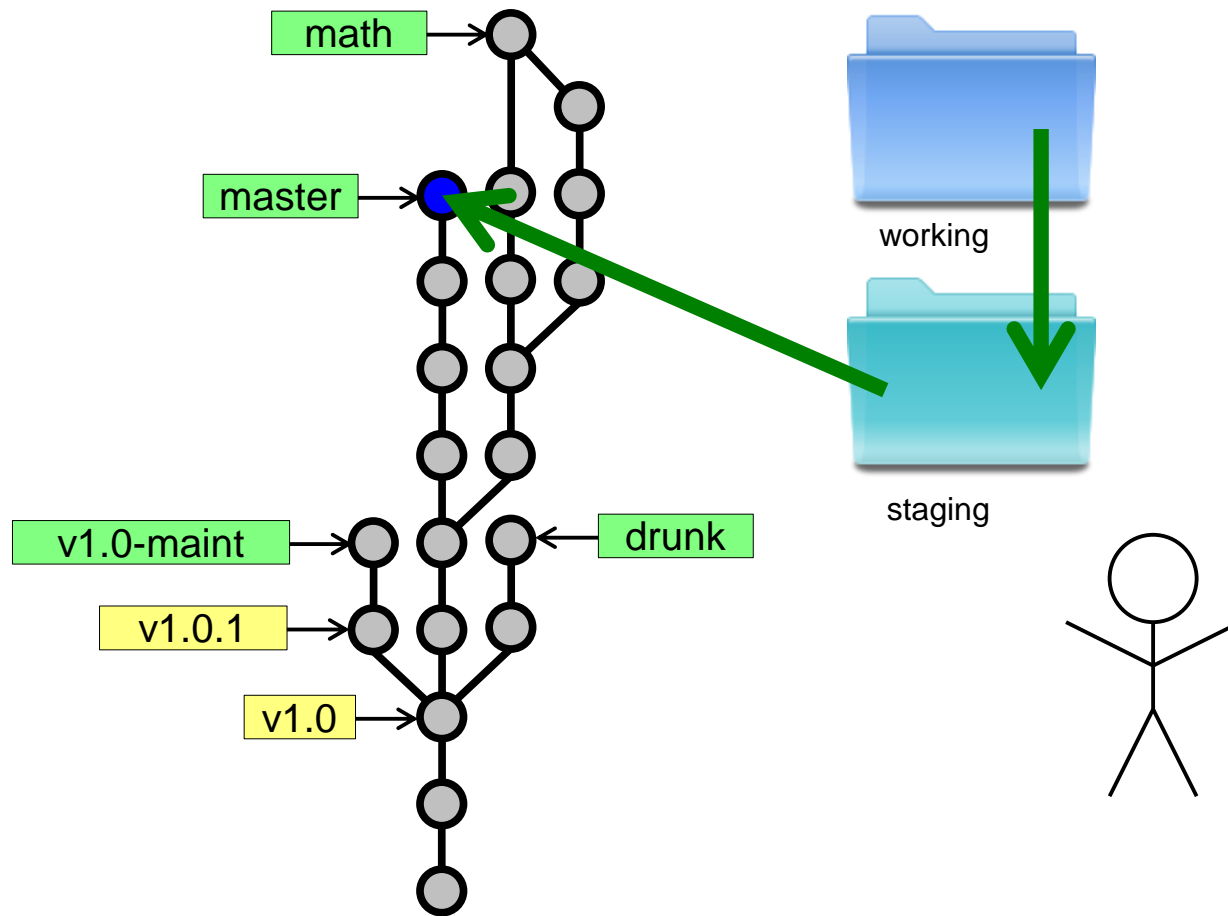
Rewriting History



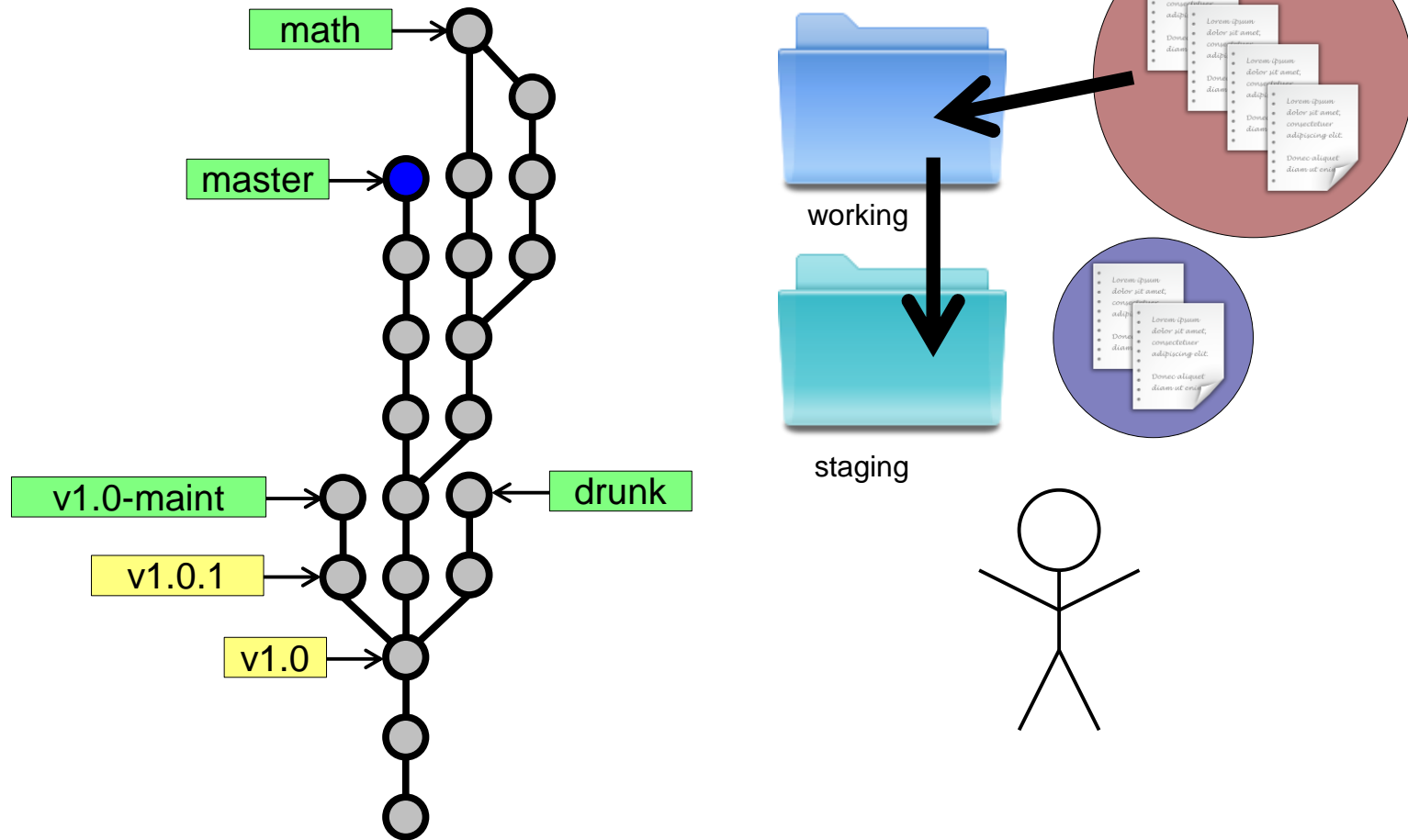
Staging Area



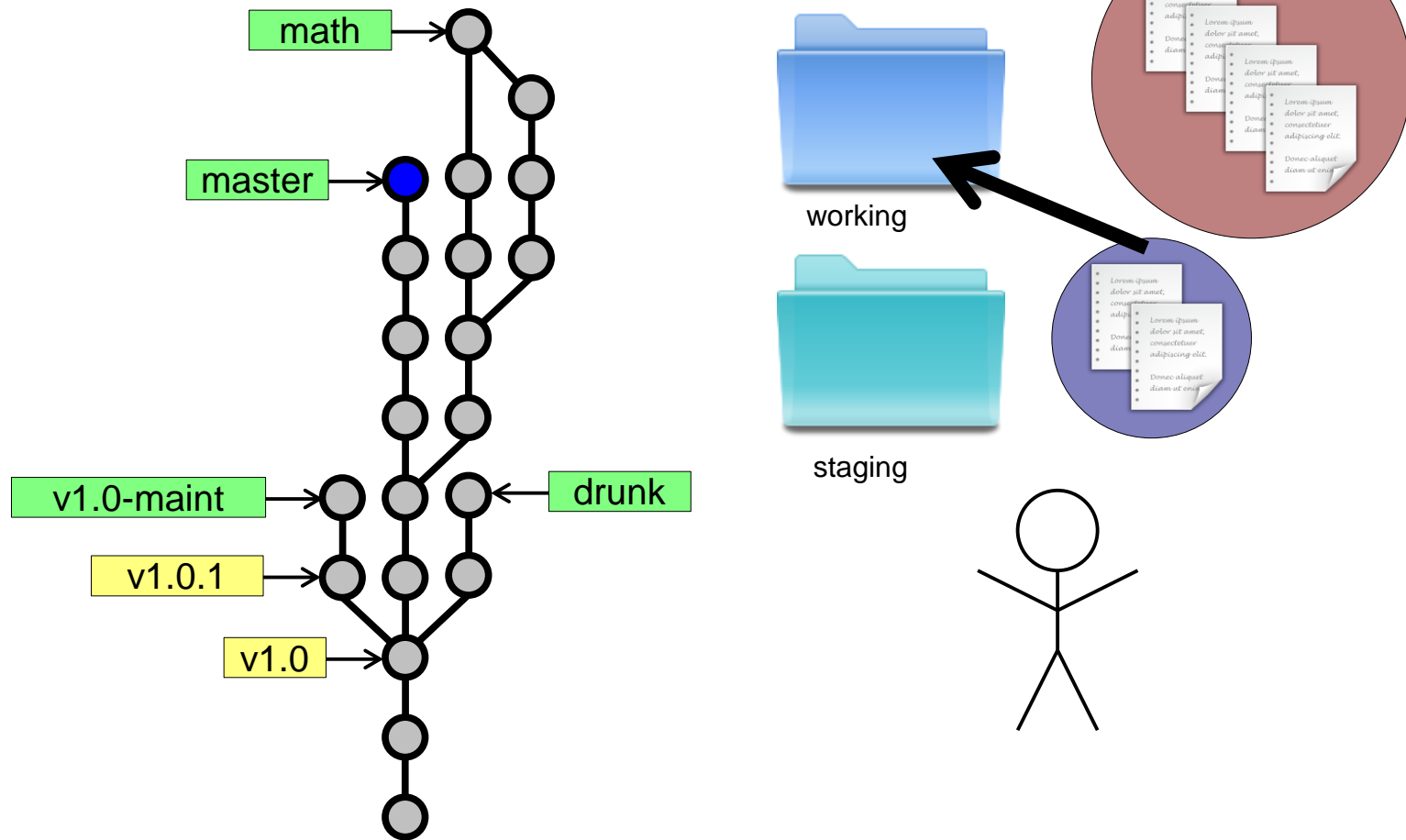
Staging Area



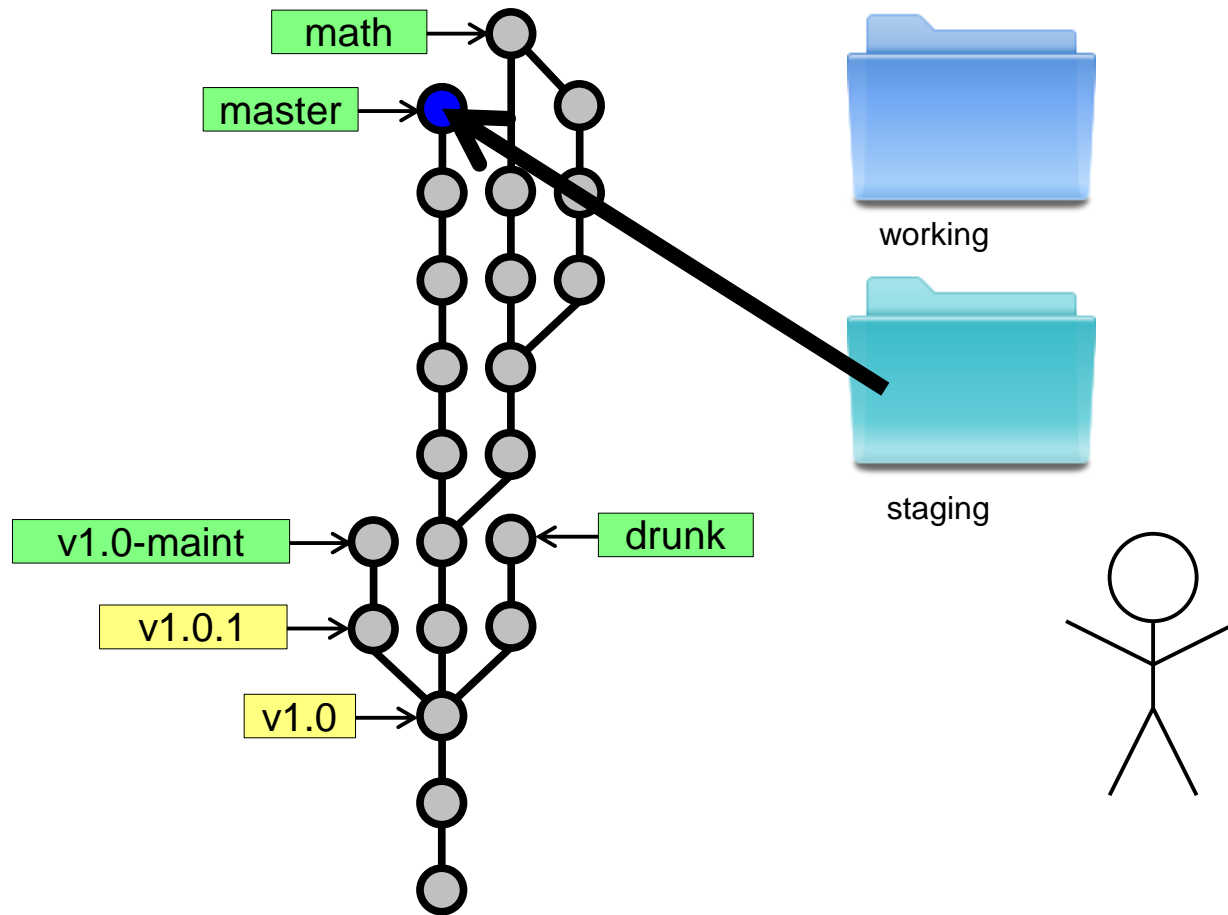
Staging Area



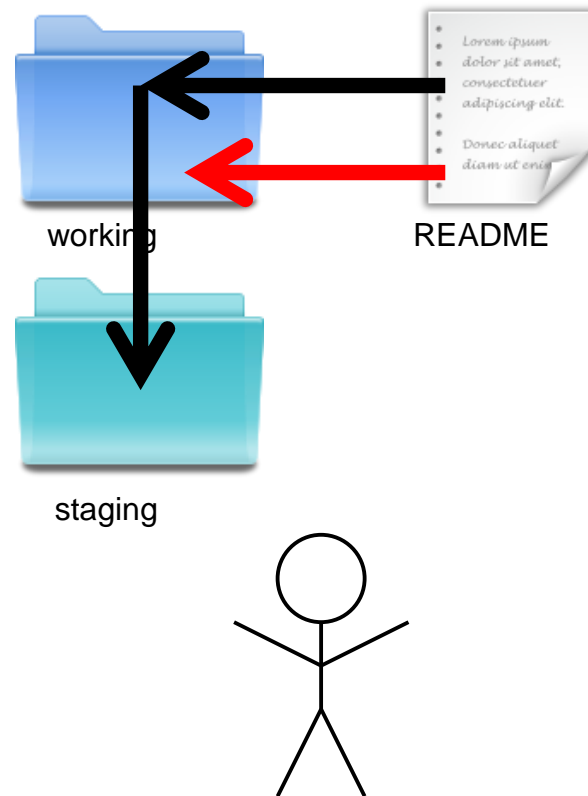
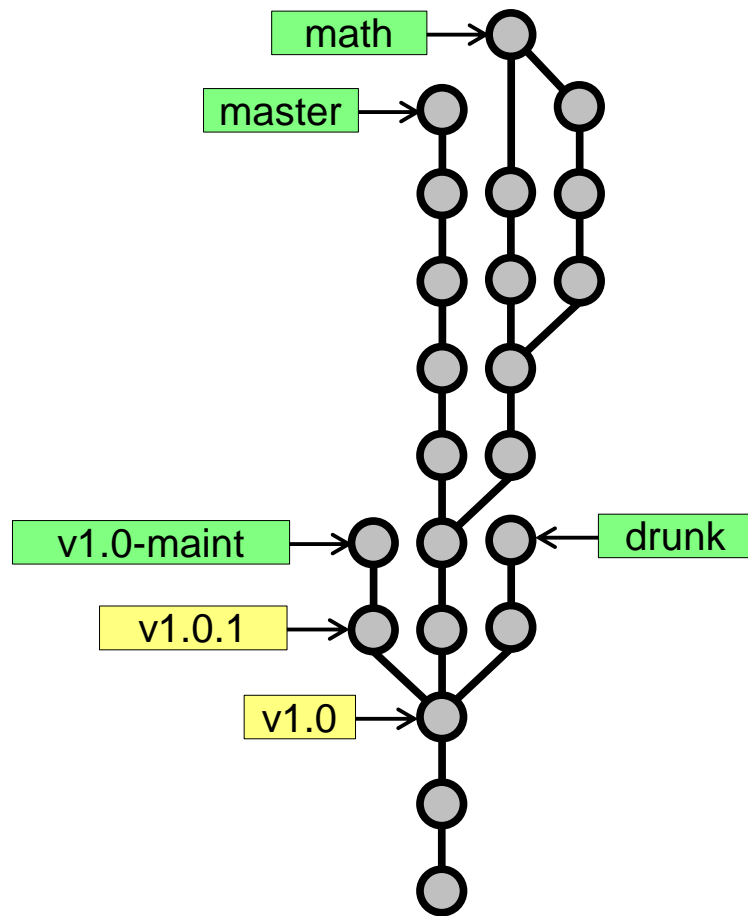
Staging Area



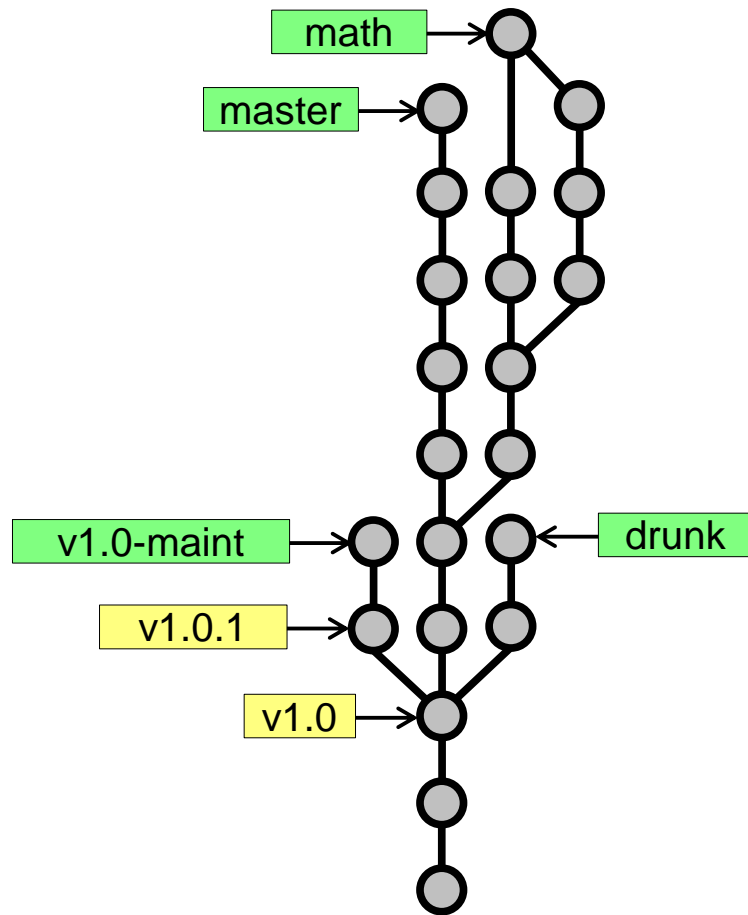
Staging Area



Staging Area



Diffs



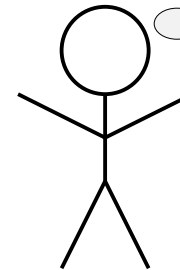
working



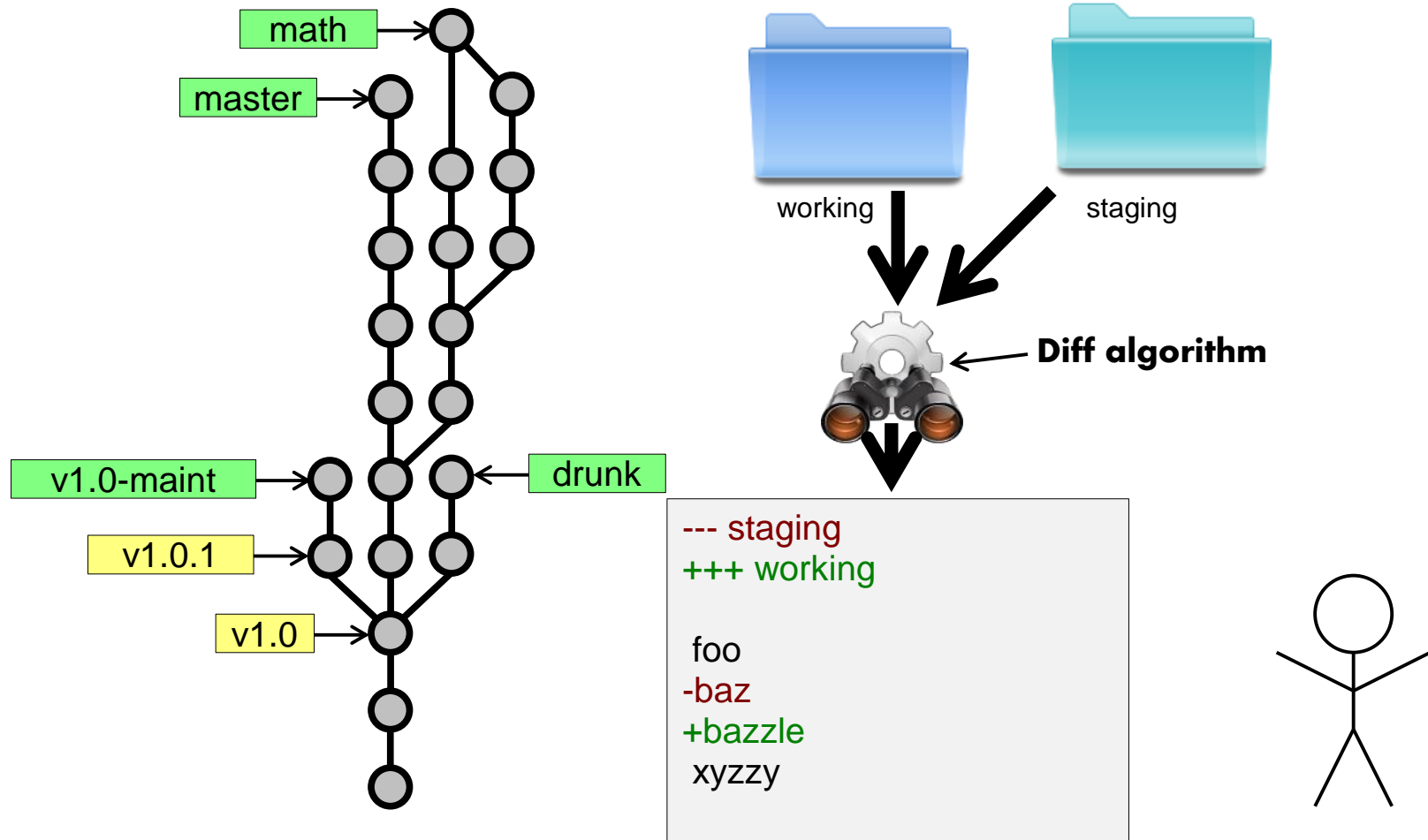
staging

What are the changes?

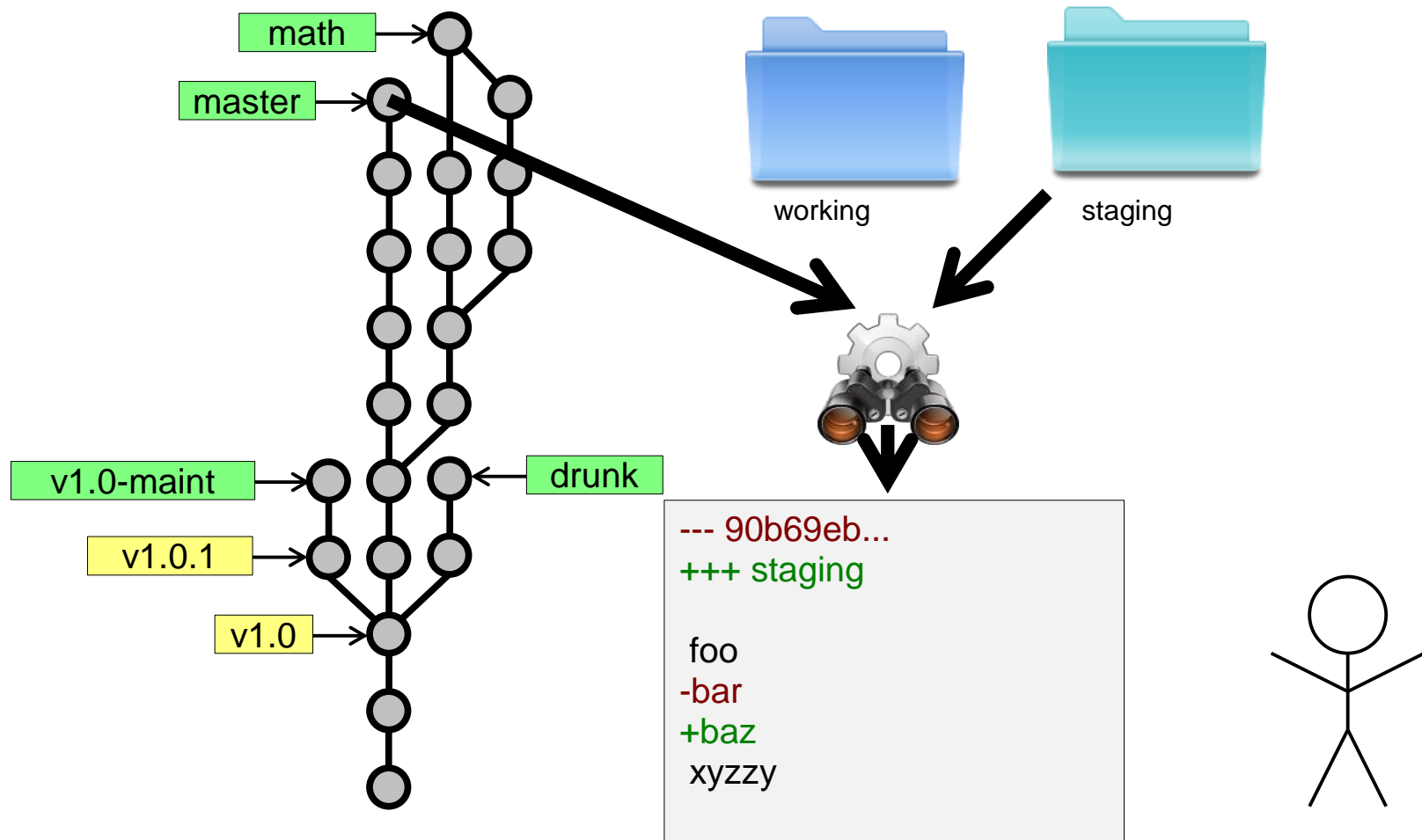
- working vs. staging
- working vs. snapshot X
- staging vs. snapshot X
- snapshot X vs. snapshot Y



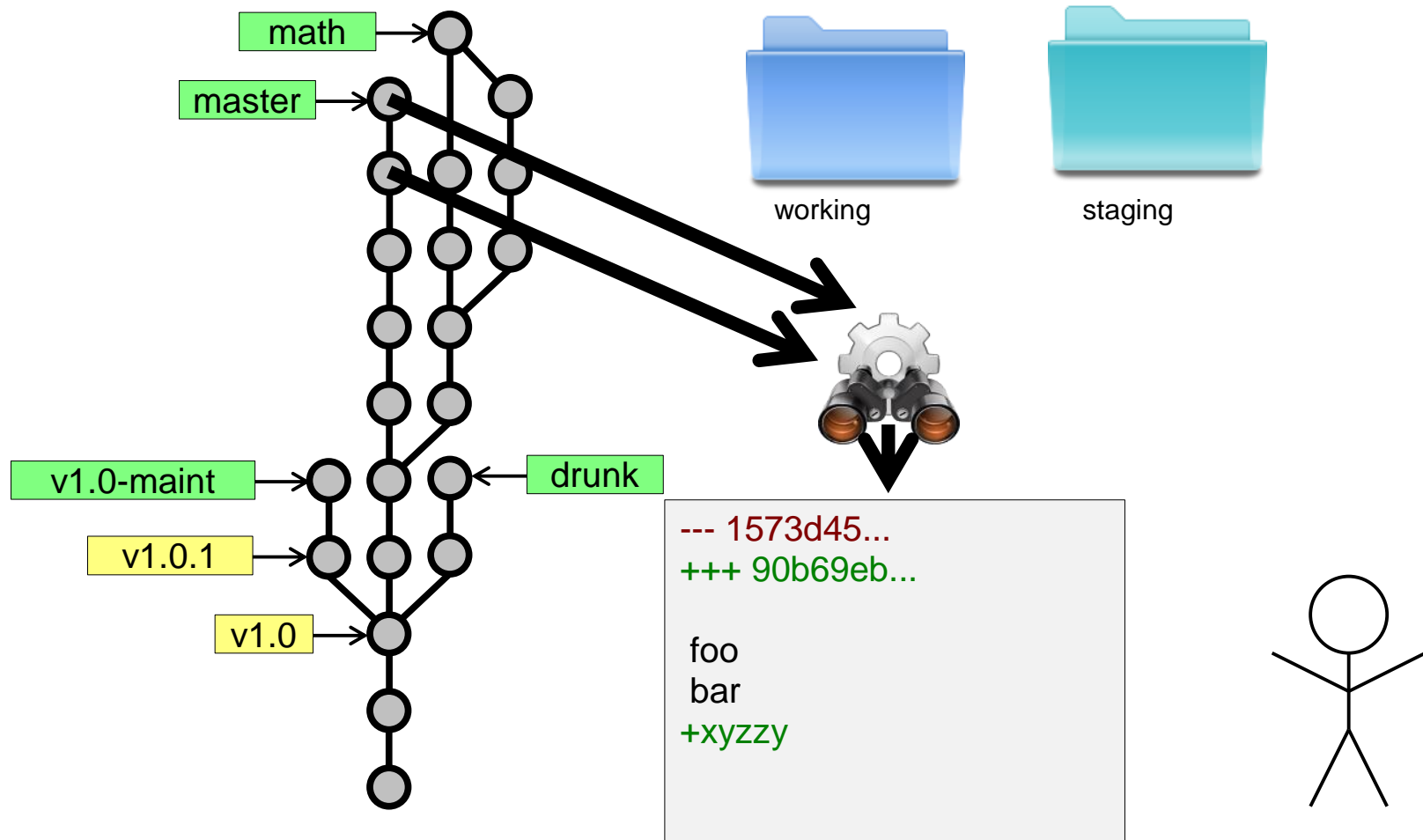
Diffs



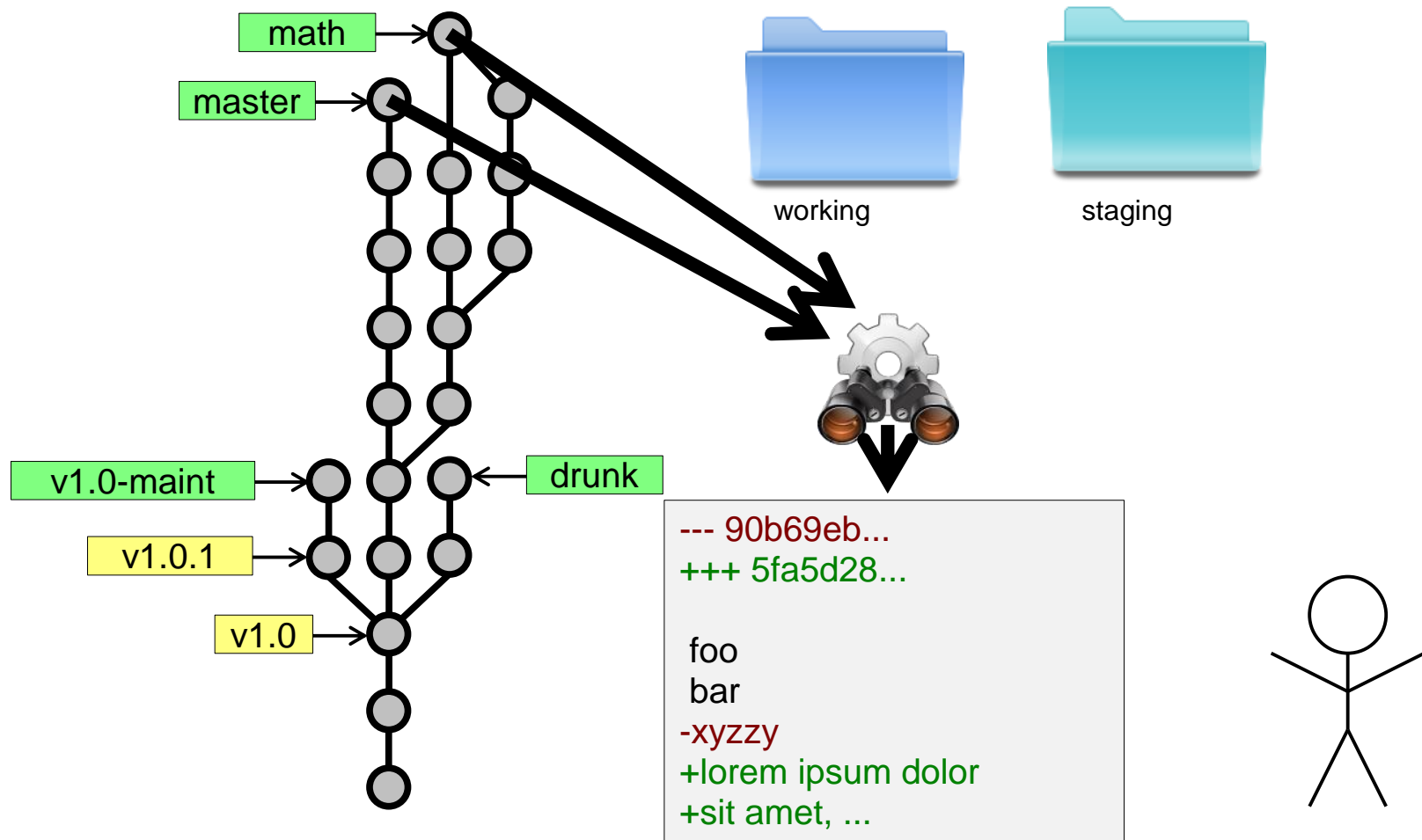
Diffs



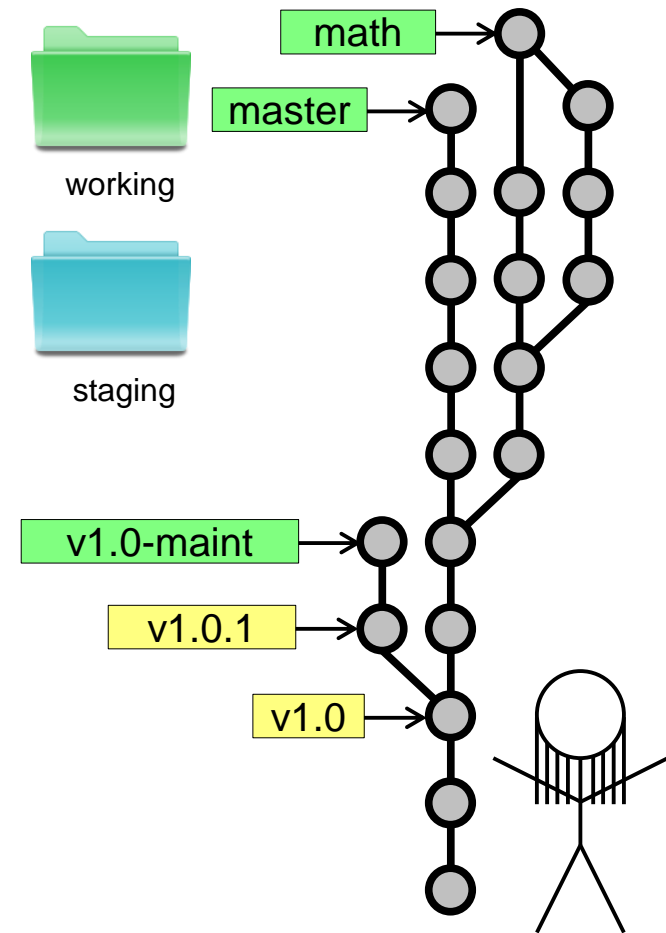
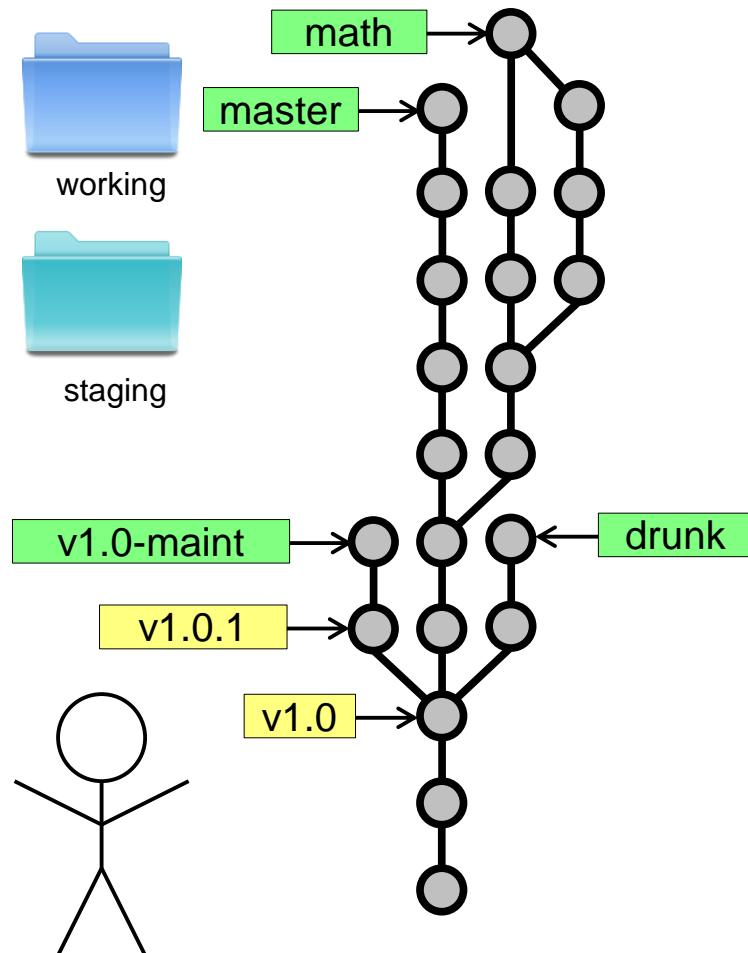
Diffs



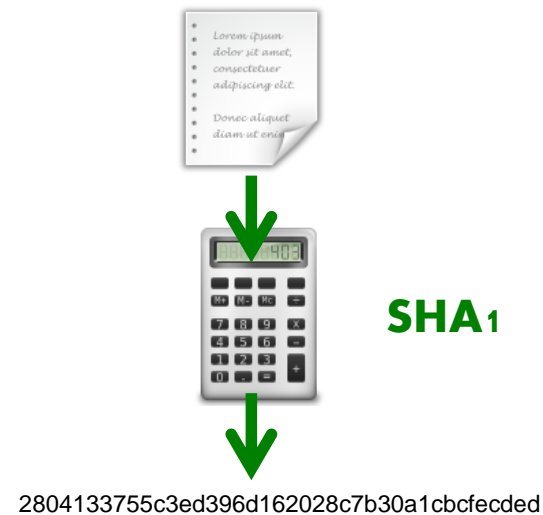
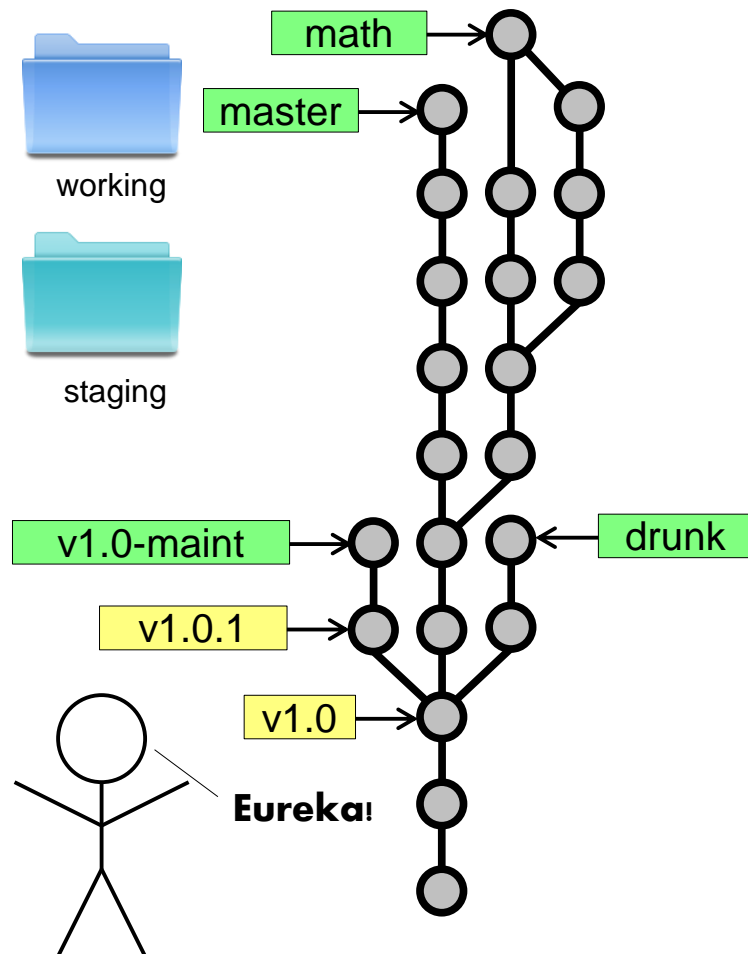
Diffs



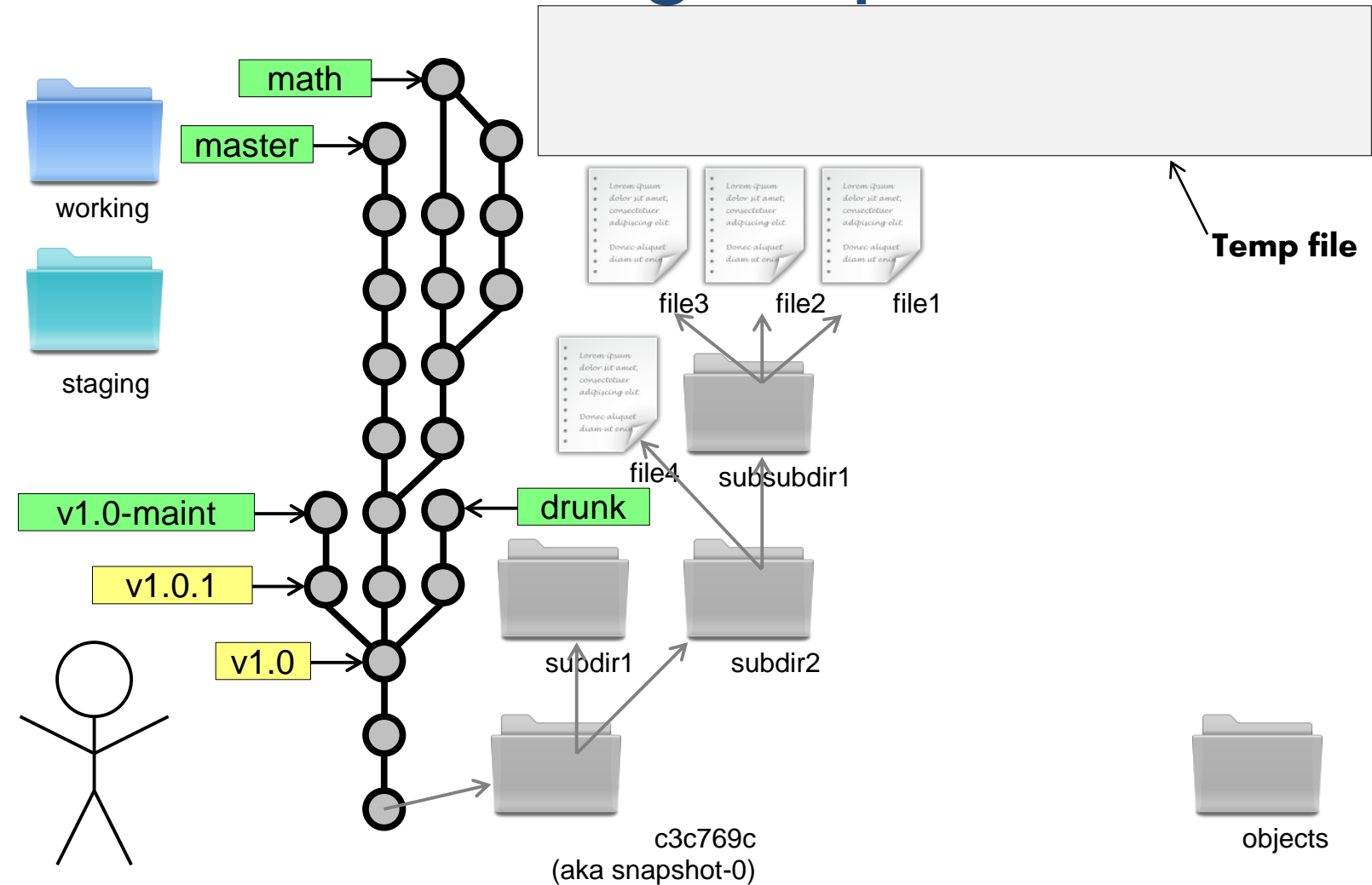
Eliminating Duplication



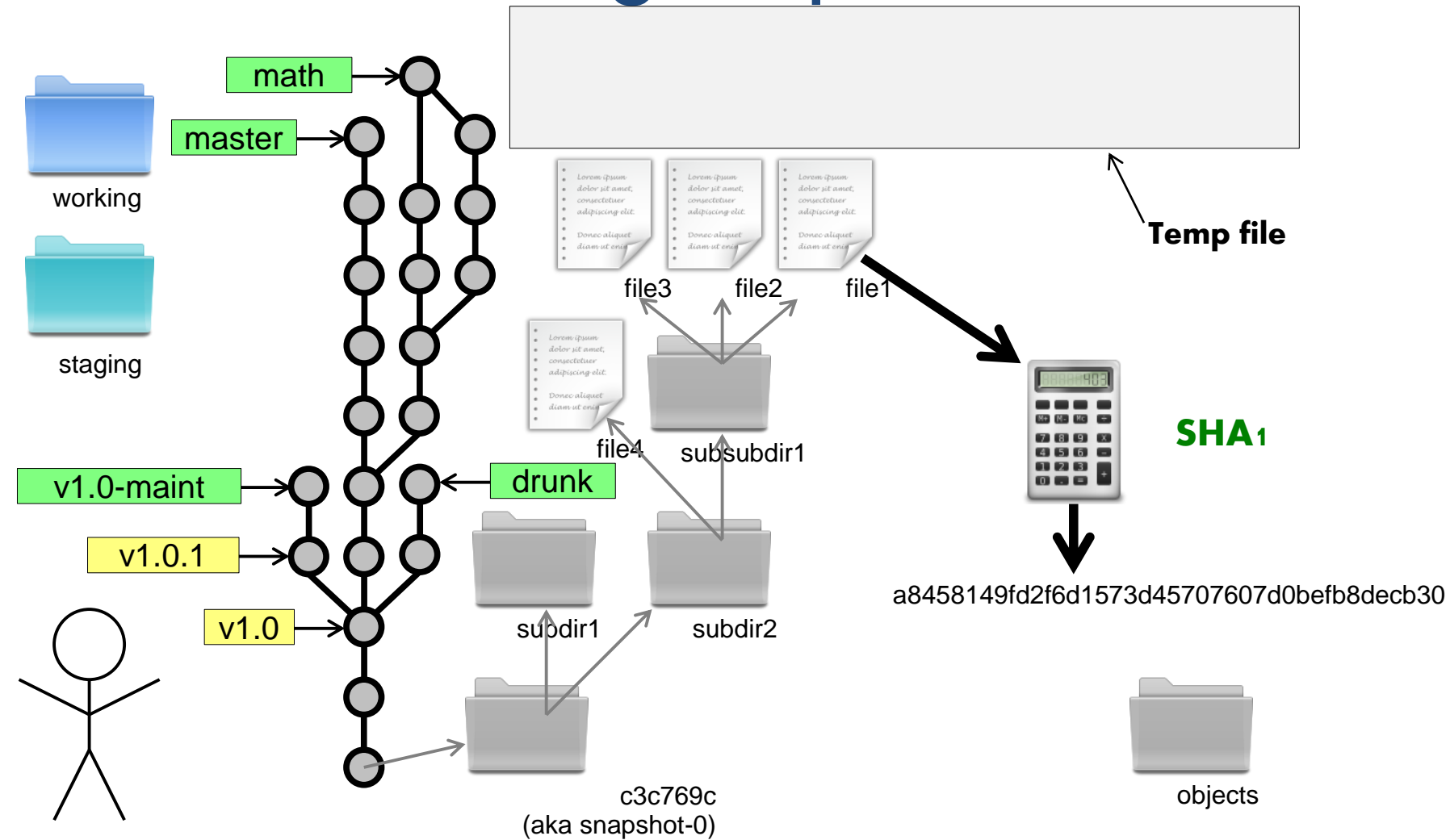
Eliminating Duplication



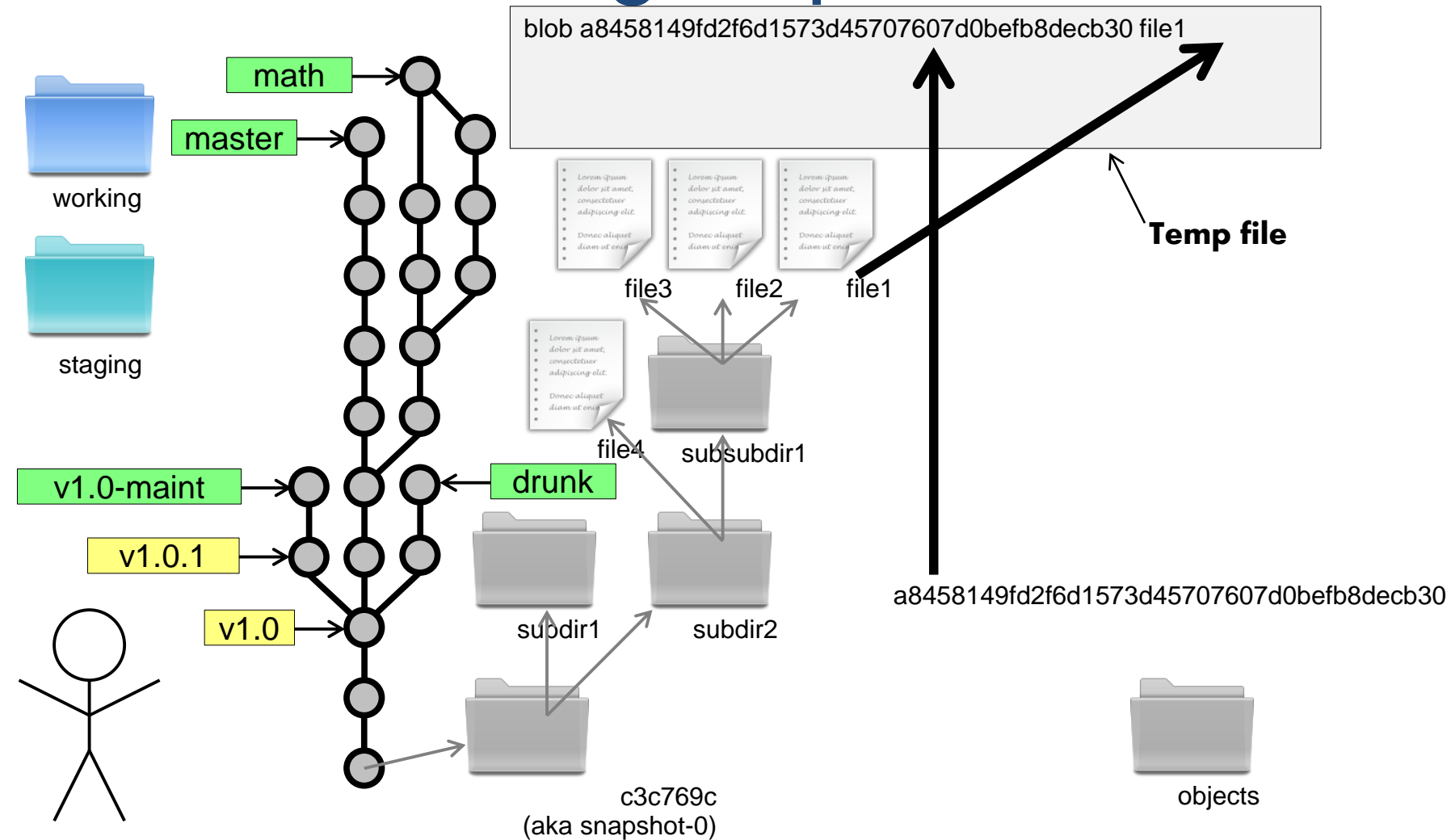
Eliminating Duplication



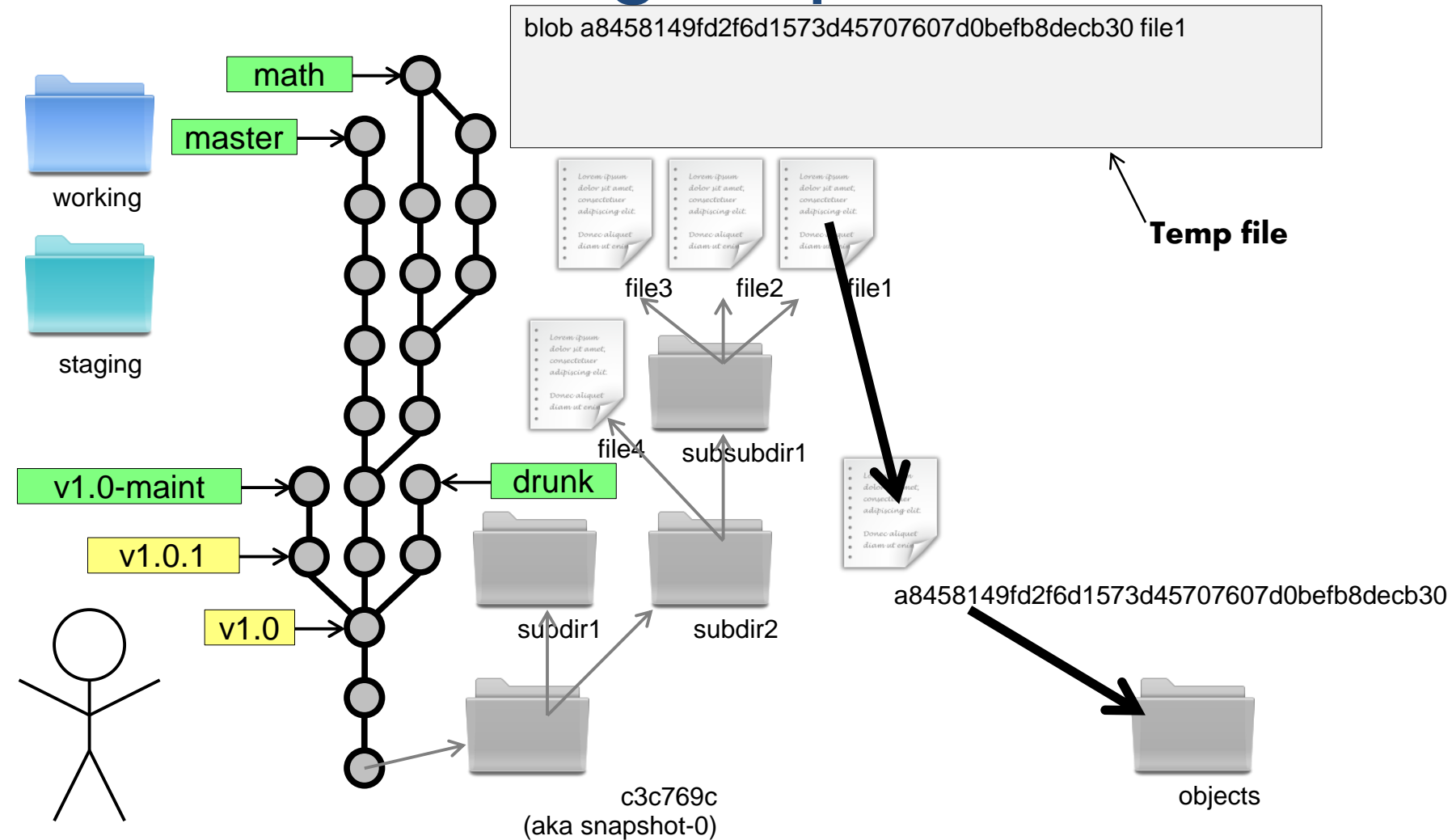
Eliminating Duplication



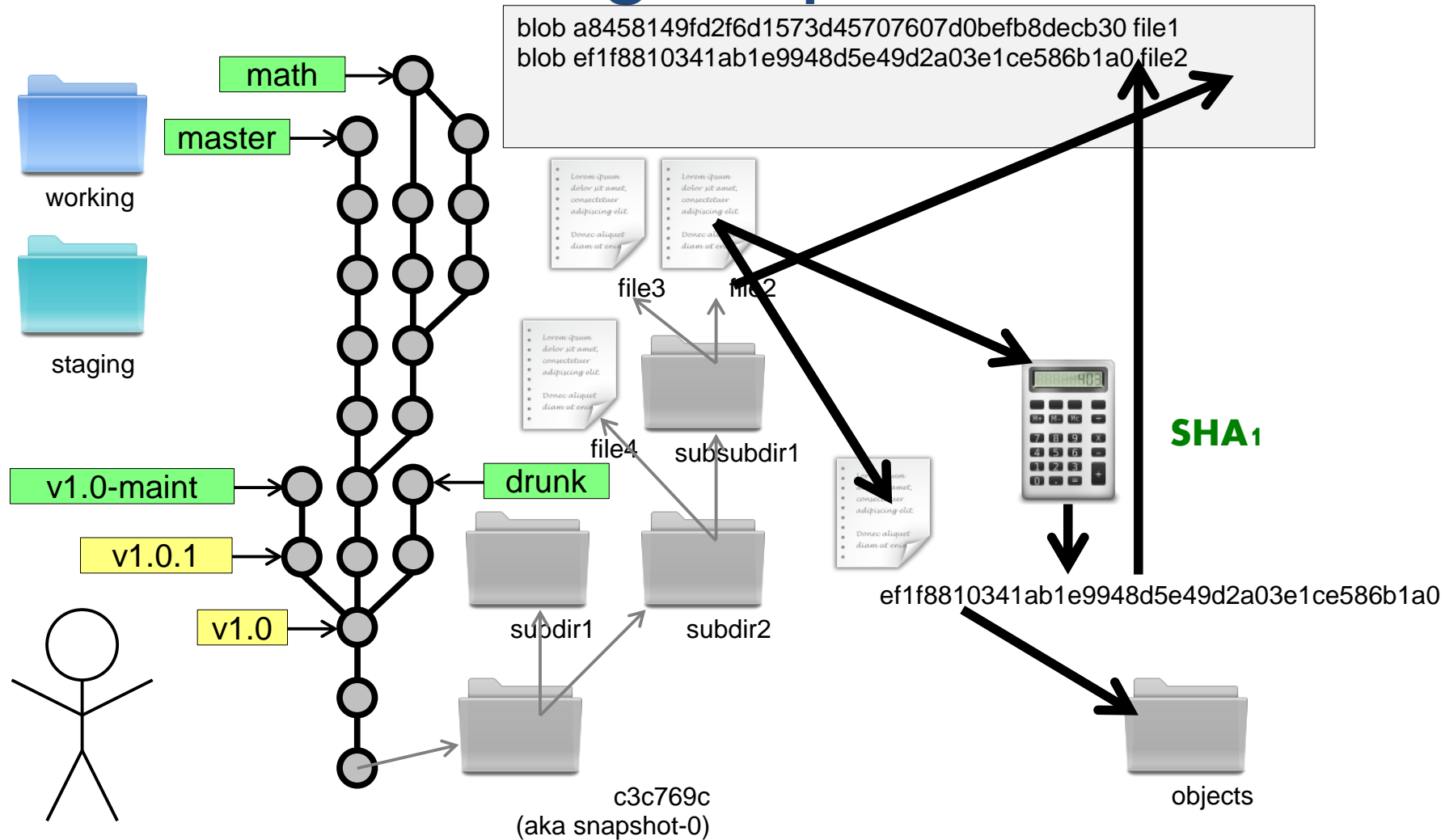
Eliminating Duplication



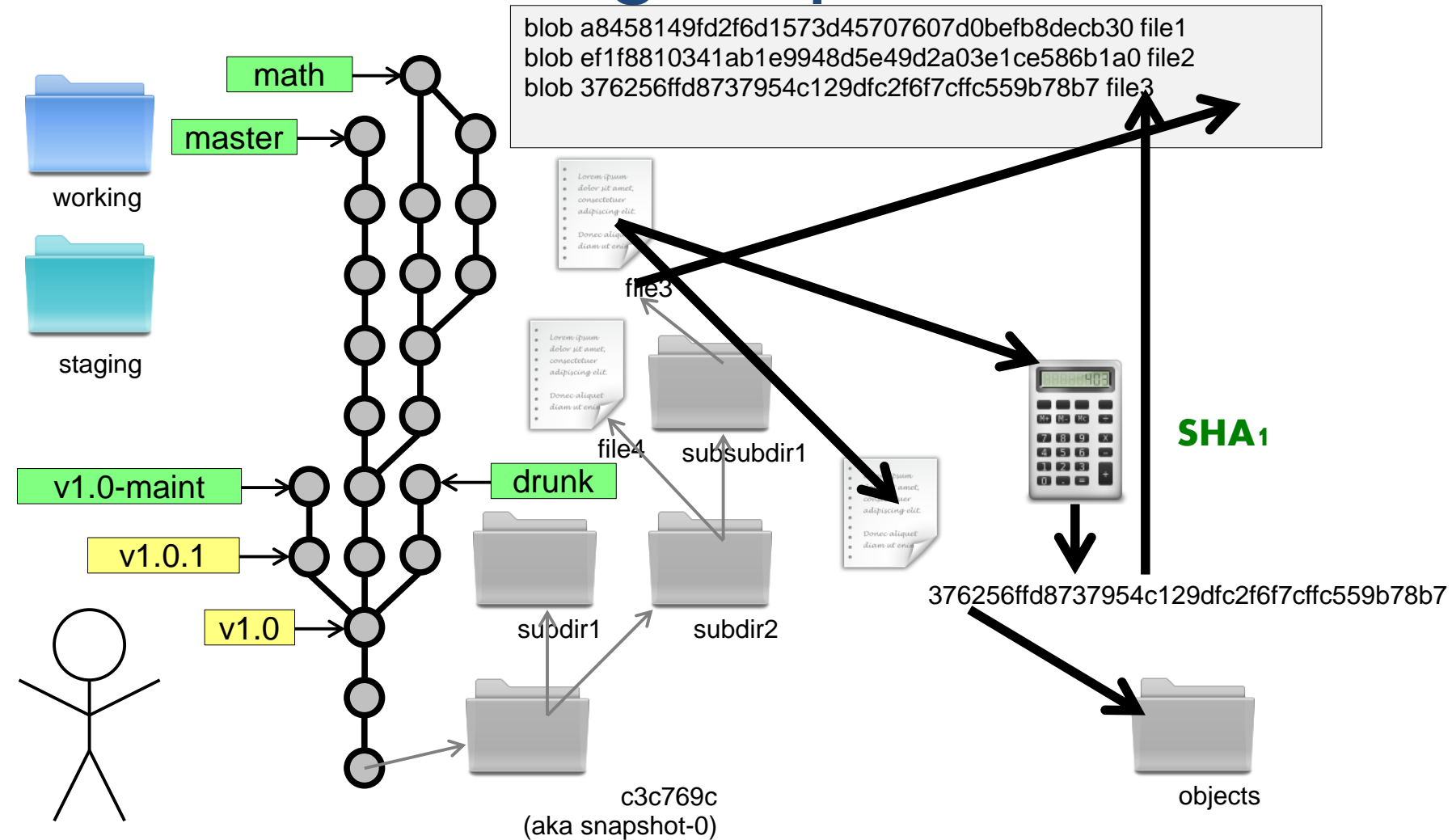
Eliminating Duplication



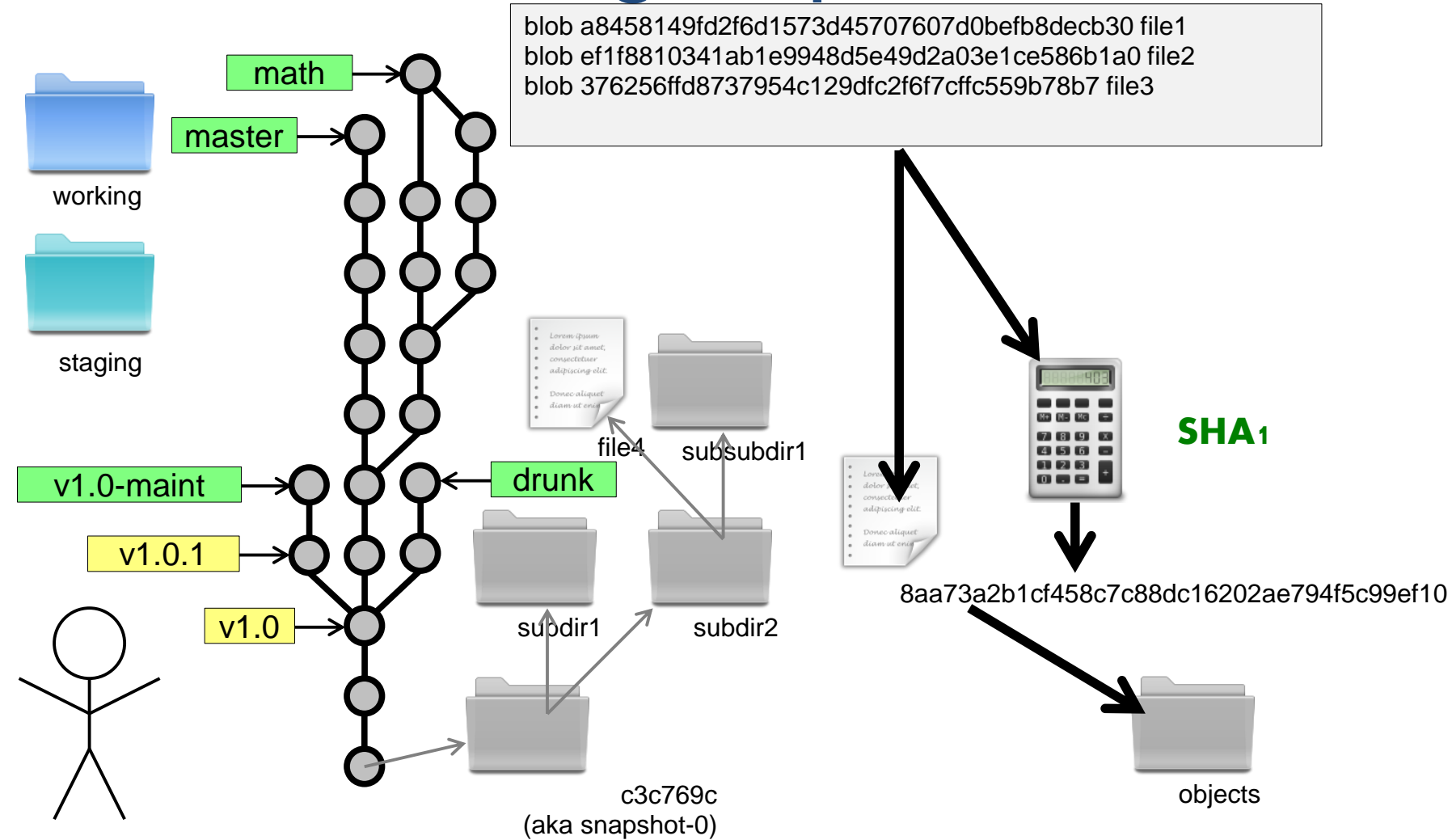
Eliminating Duplication



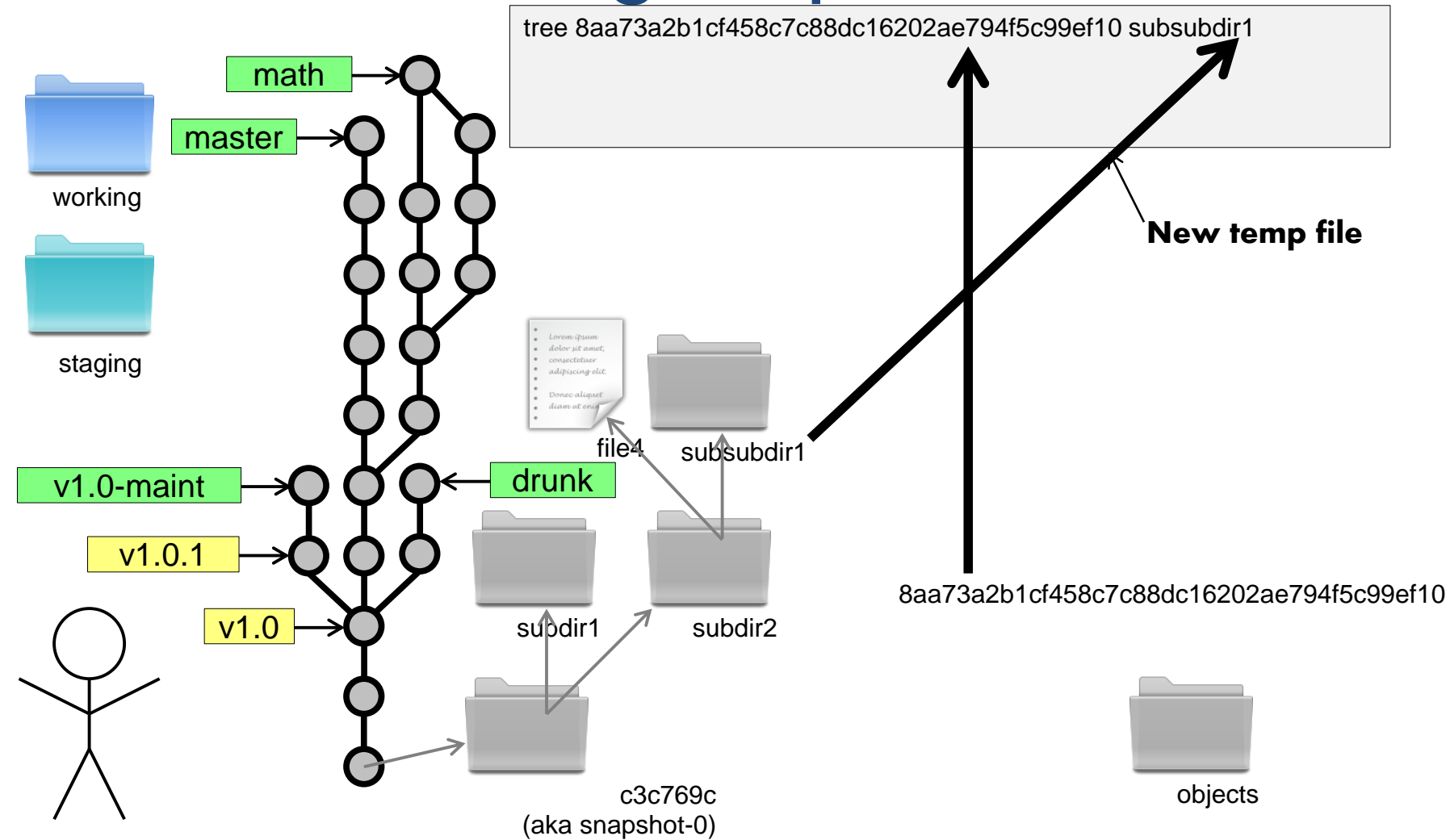
Eliminating Duplication



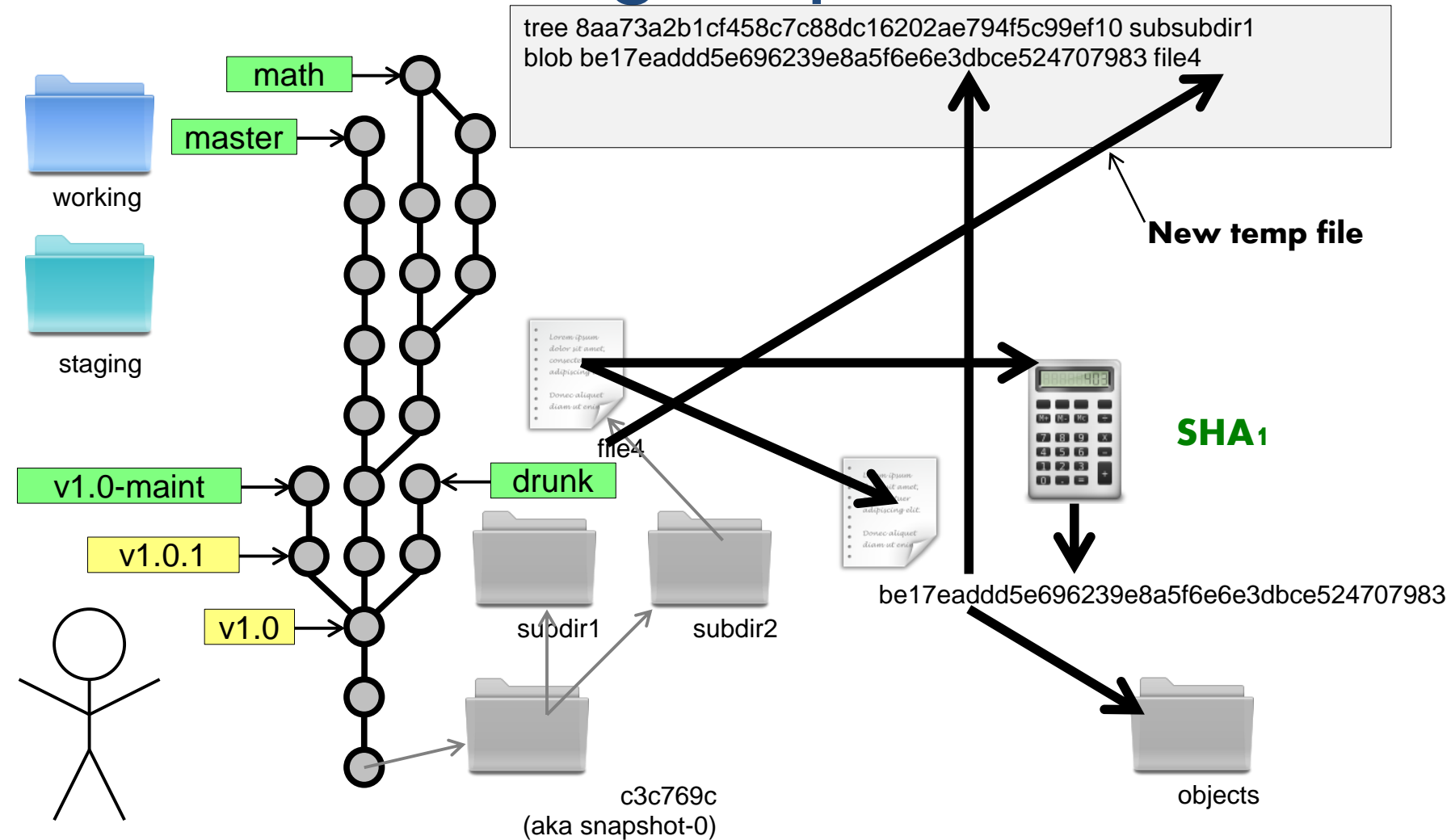
Eliminating Duplication



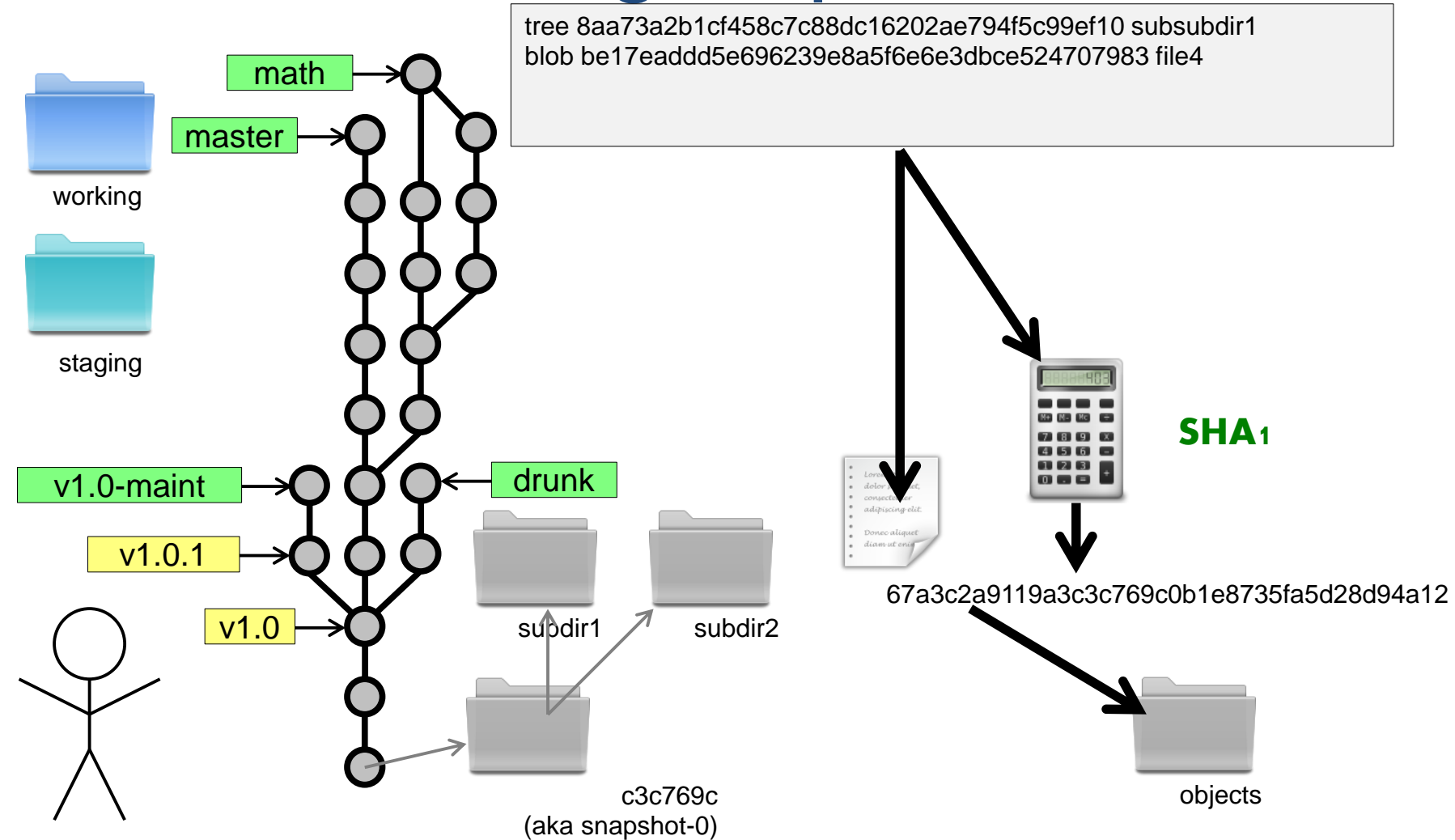
Eliminating Duplication



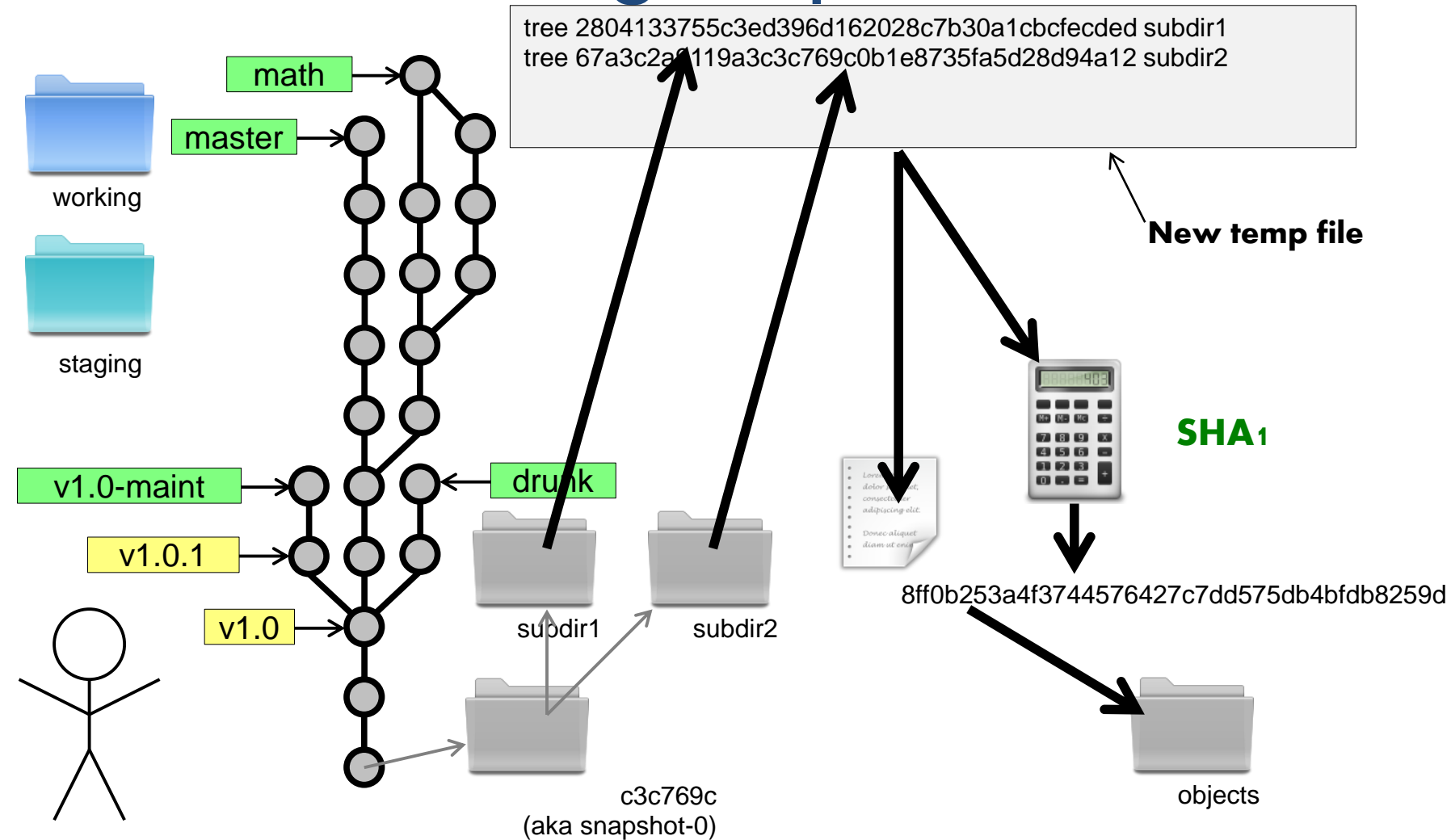
Eliminating Duplication



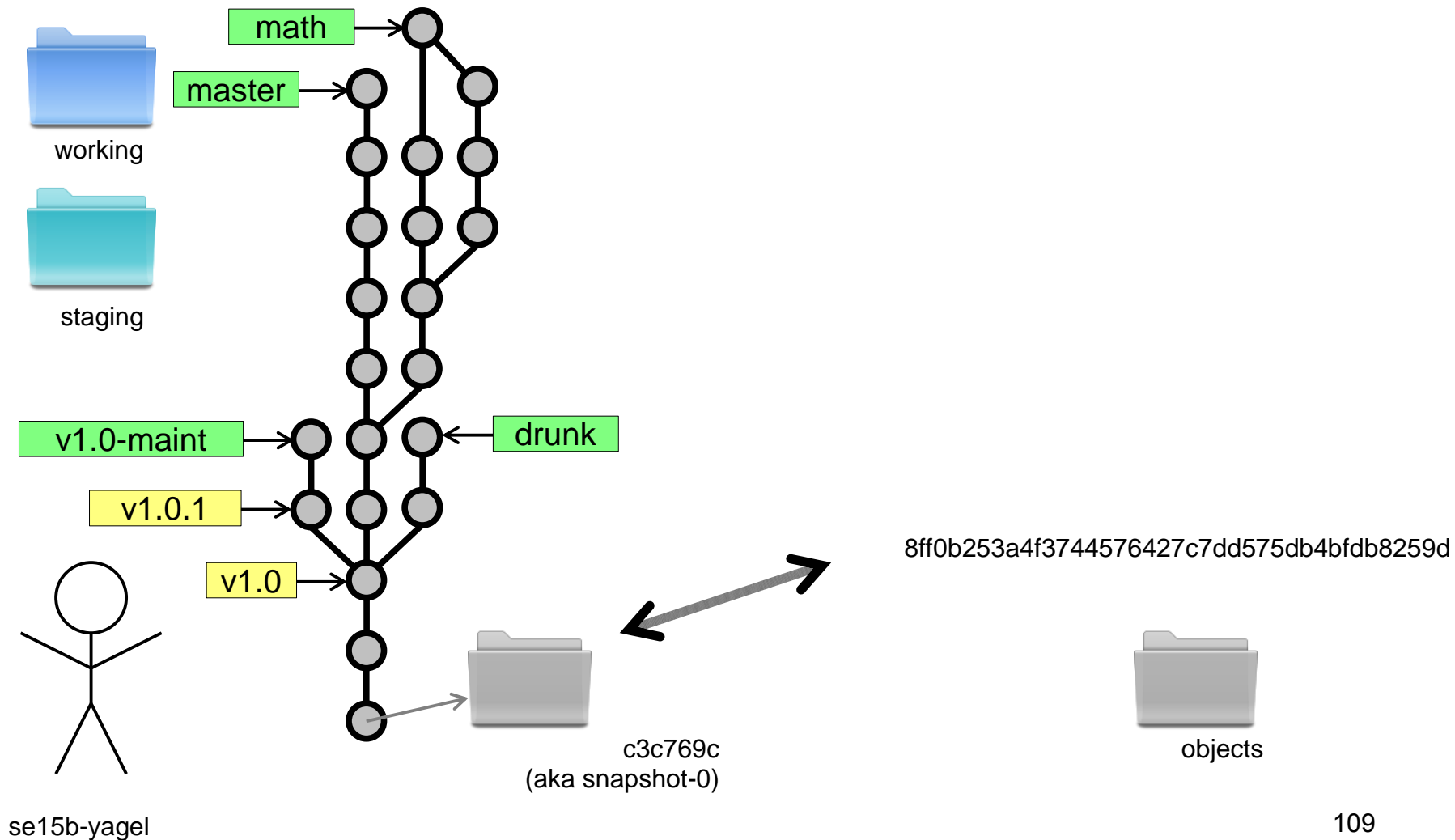
Eliminating Duplication



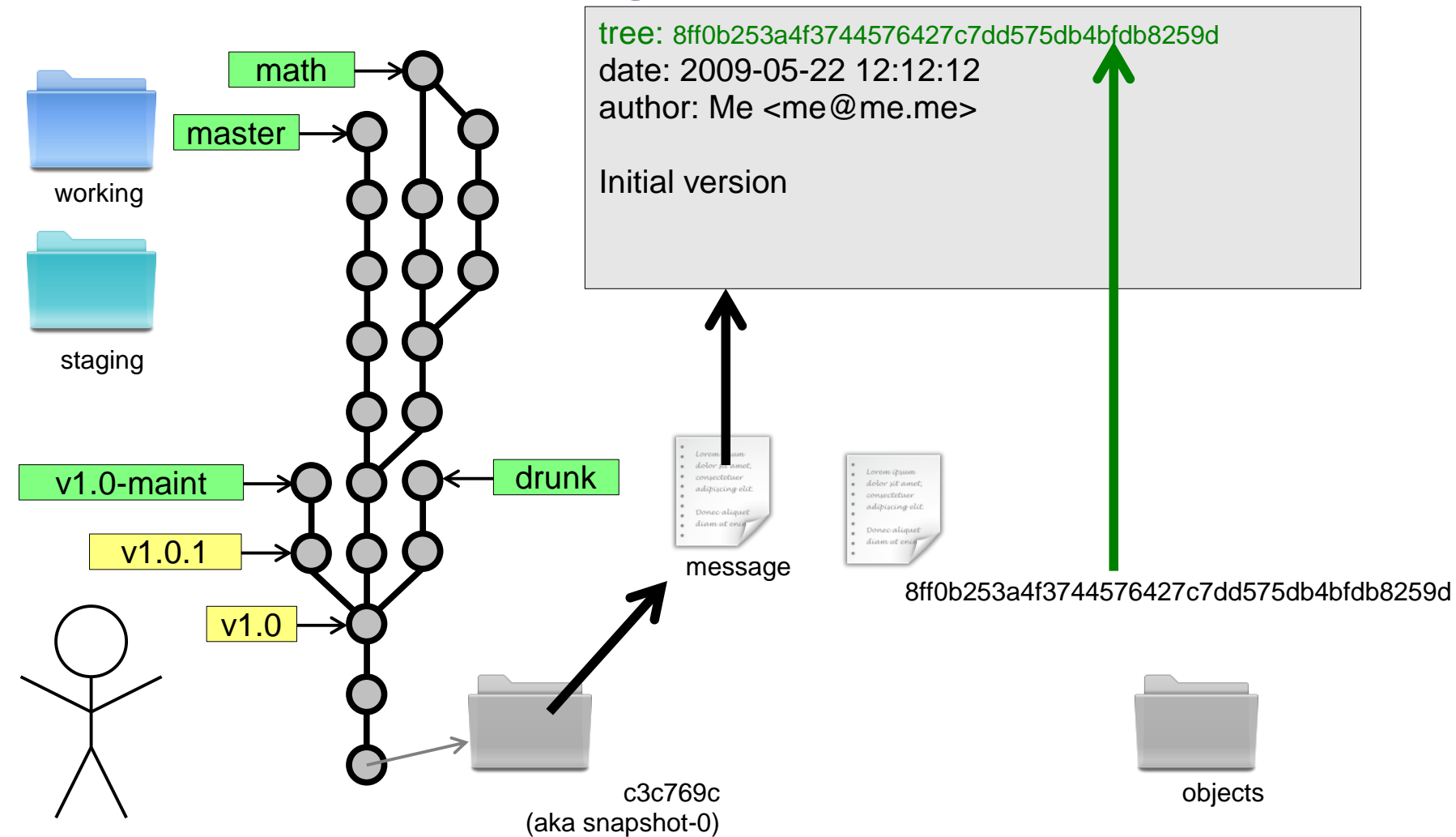
Eliminating Duplication



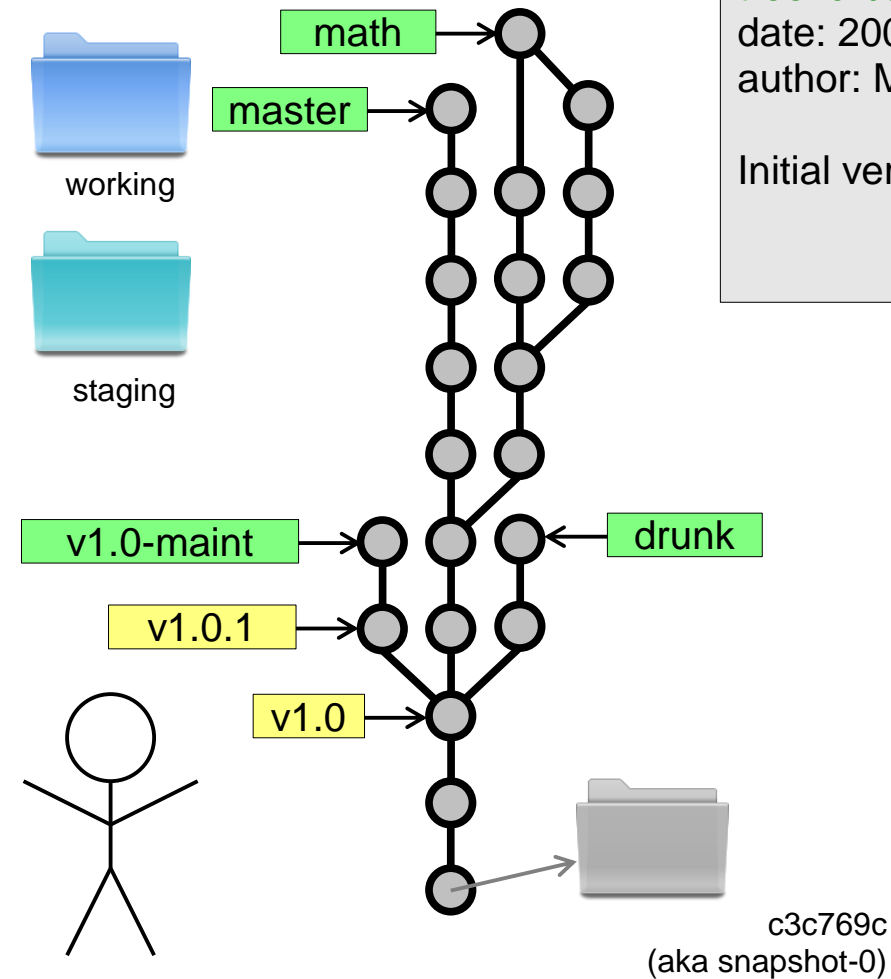
Eliminating Duplication



Eliminating Duplication



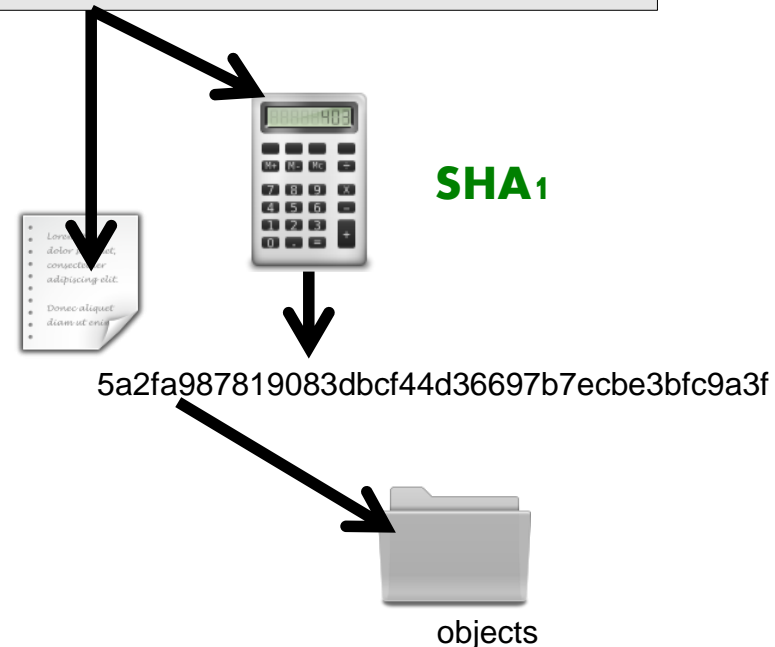
Eliminating Duplication



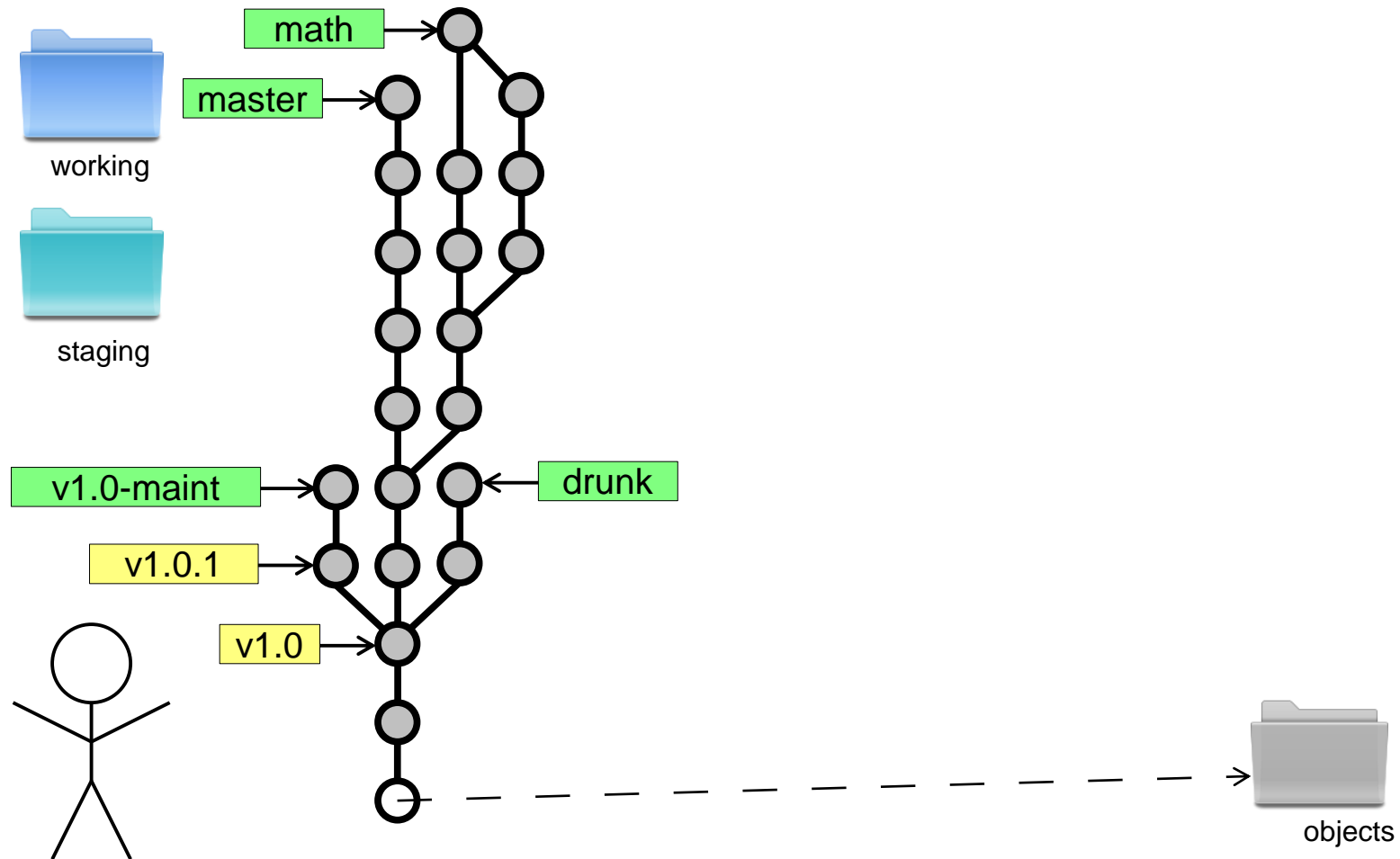
se15b-yagel

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

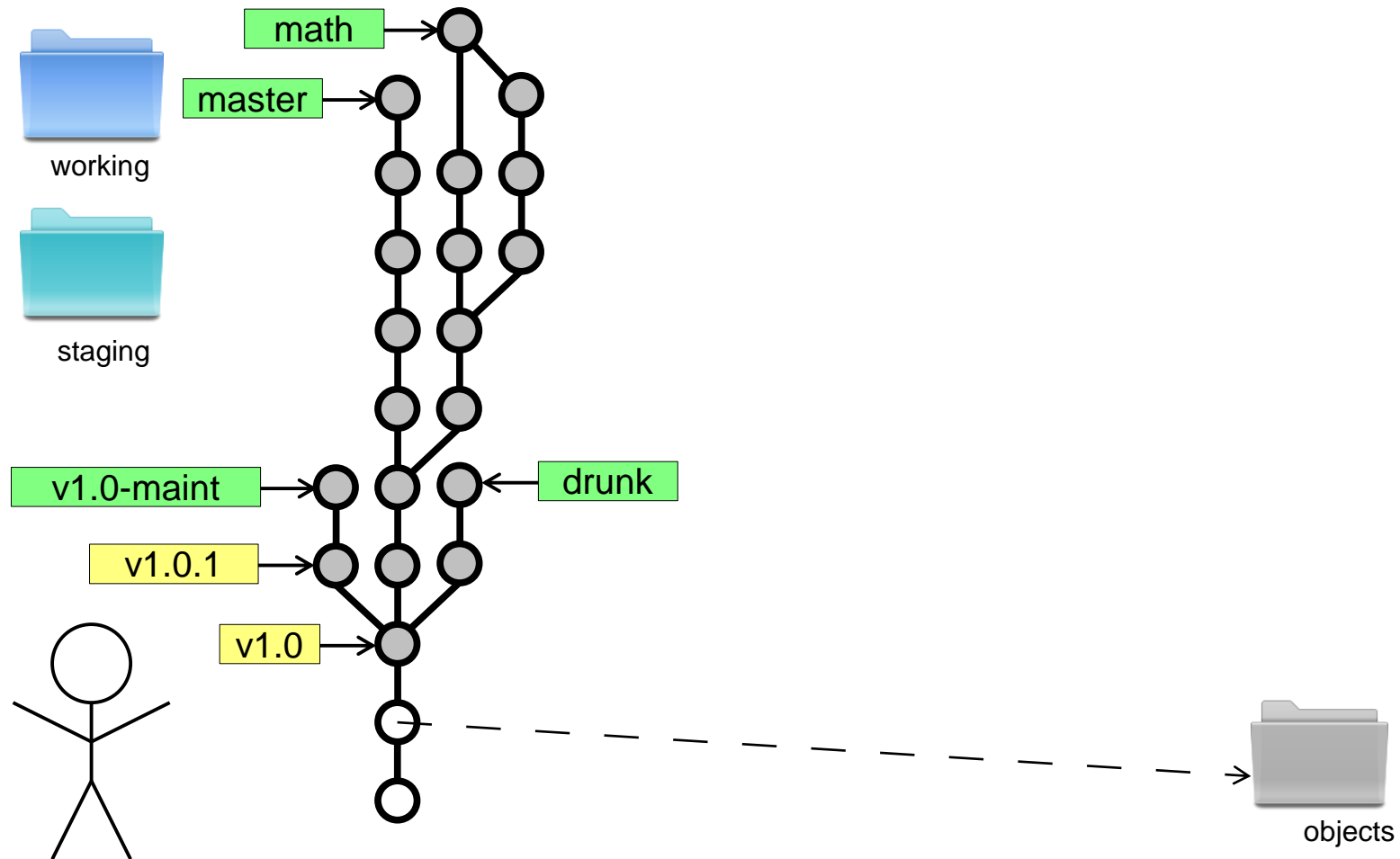
Initial version
```



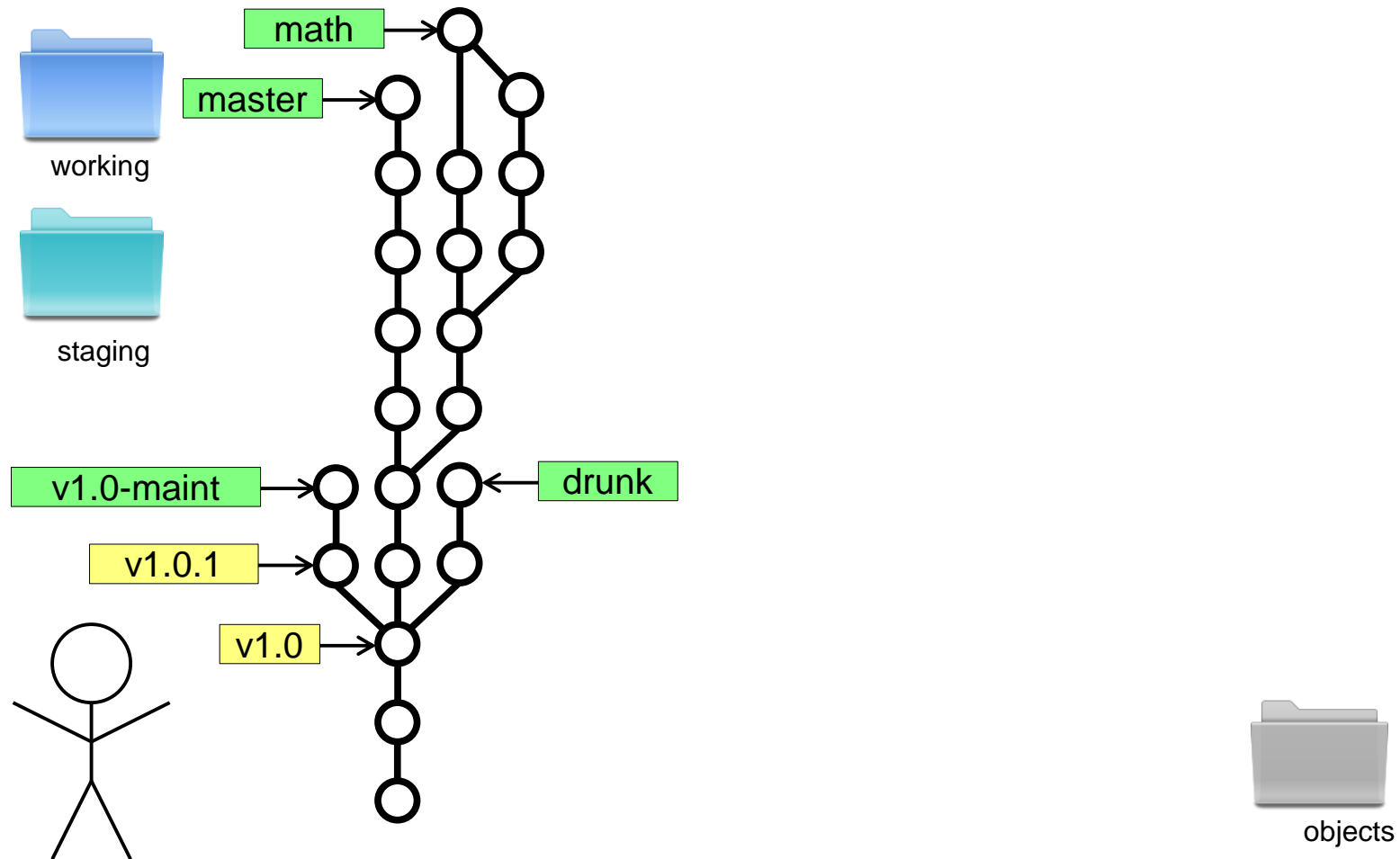
Eliminating Duplication



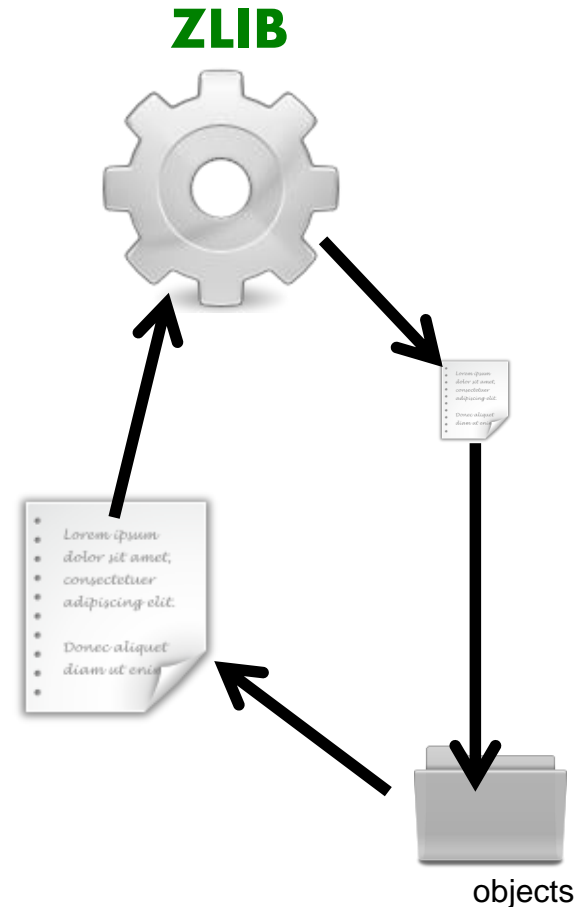
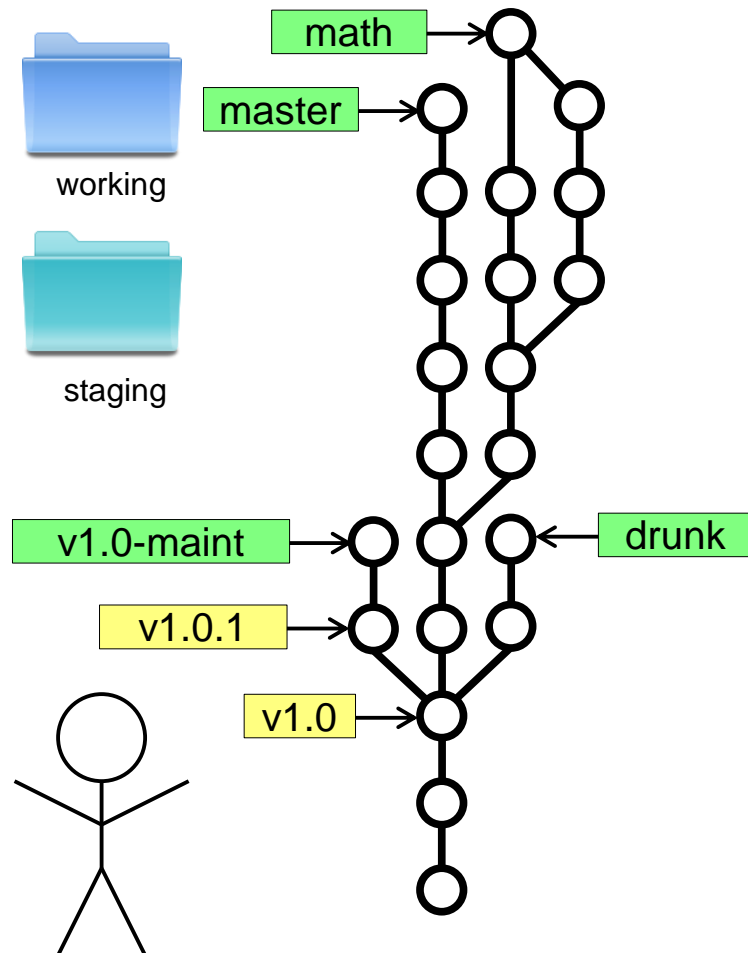
Eliminating Duplication



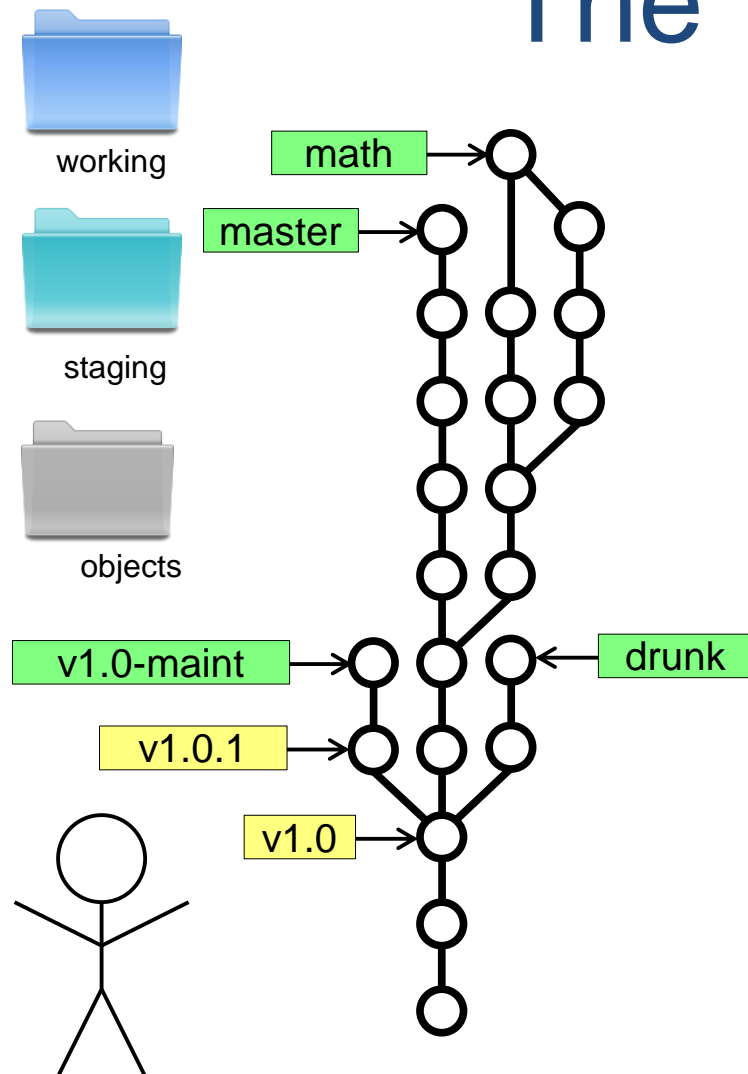
Eliminating Duplication



Compressing Blobs



The True Git



TADAAA! .

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](#)”

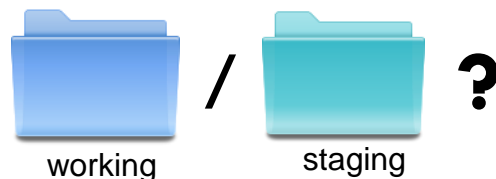
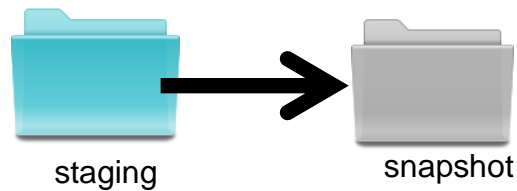
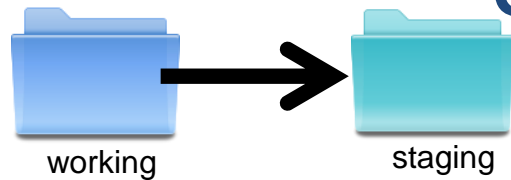
Get help: .

- 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`

gith

Commands: Diffing



working

vs.



staging

`git diff .`



staging

vs.



snapshot

`git diff --staged .`



working

vs.



snapshot

`git diff HEAD .`



snapshot

vs.



snapshot

`git diff <from> <to> .`

Commands: Branches & Tags

git branch .

git branch <branch> .

git checkout -b ...

git checkout <branch> .

git tag -l .

git tag <tag> .

Commands: Fetching & Merging

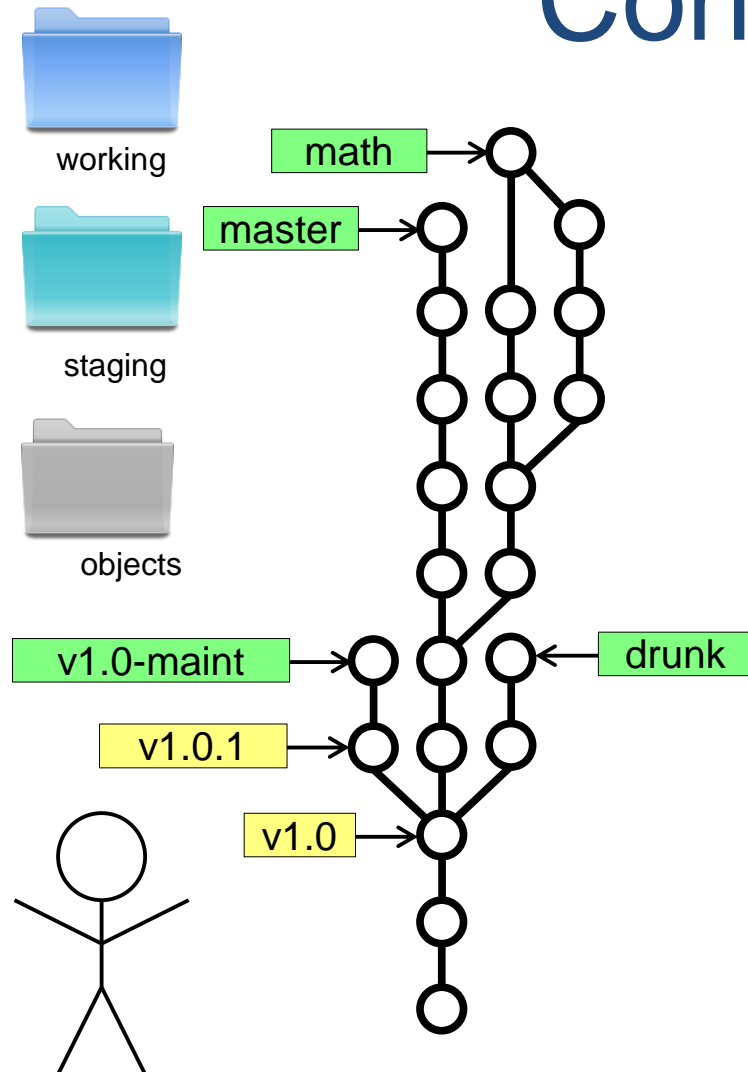
`git remote add <name> <URL> .`

`git fetch <name> .`

`git pull`

`git merge <name>/<branch> .`

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> •

Questions?

Thanks for your attention! .

These slides are available at: .
https://github.com/jherland/git_parable

Reach me at <johan@herland.net> .



לסיכום

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

סיכום הקורס

- נושאים שעברנו
- סקר ודיון
- מה הלאה

