

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 6: תחילת סבבים – סבב 1 MVP
- 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.ericssink.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](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 (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_pparable

The Git Parable

Johan Herland

johan@herland.net

The Git Parable

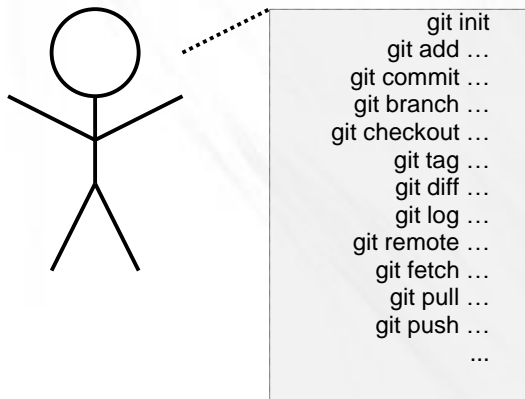
- Shamelessly stolen from Tom Preston-Werner
<http://tom.preston-werner.com/2009/05/19/the-git-parable.html>
- I'm lazy...
- Also: Best introduction to Git I've found so far

The Git Parable

- Git - simple & powerful

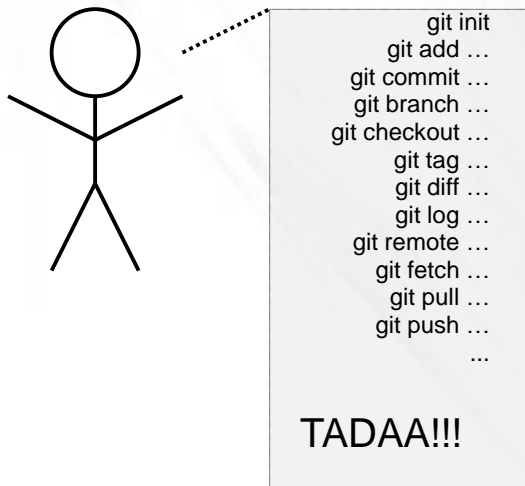
The Git Parable

- Git - simple & powerful



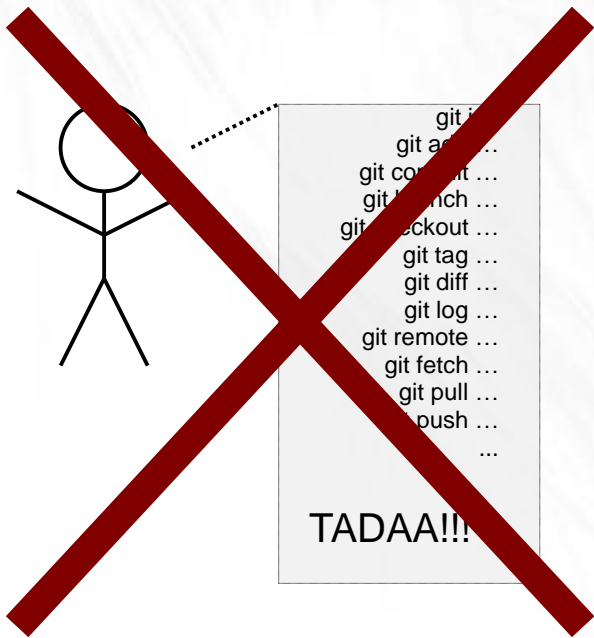
The Git Parable

- Git - simple & powerful



The Git Parable

- Git - simple & powerful



se17a-yagel

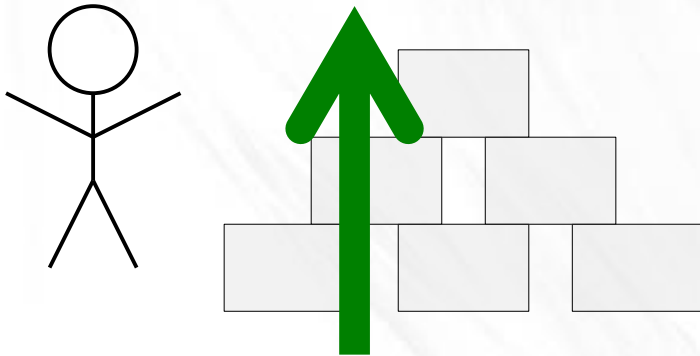
The Git Parable

- Git - simple & powerful



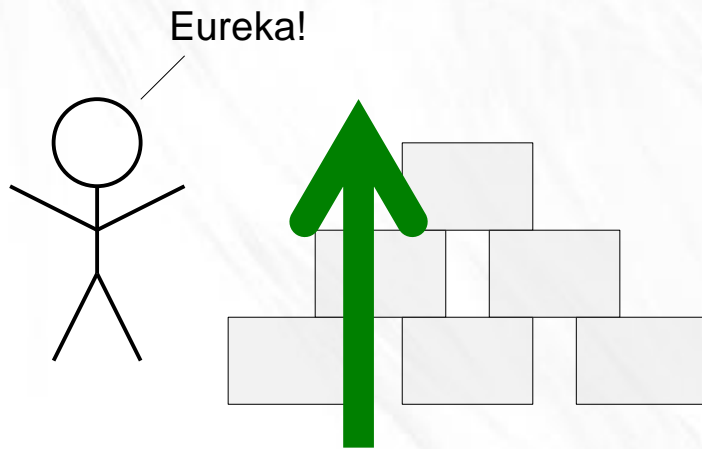
The Git Parable

- Git - simple & powerful



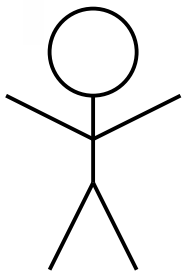
The Git Parable

- Git - simple & powerful



The Parable

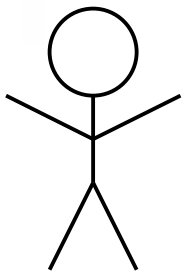
- A simple computer
 - A text editor
 - A few filesystem commands



se17a-yagel

The Parable

- Write a large software program

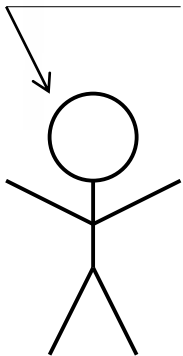


se17a-yagel

The Parable

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

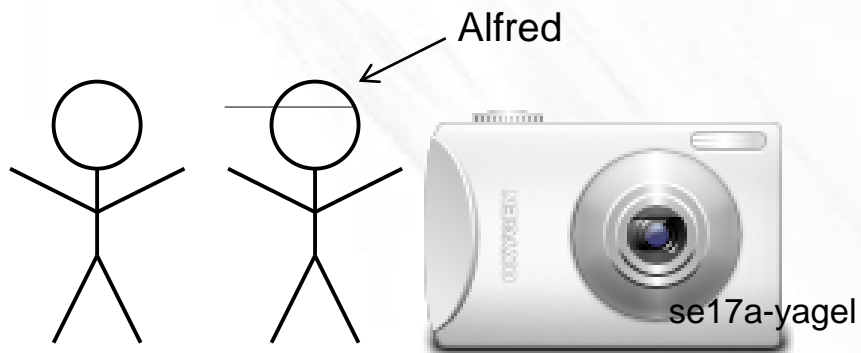
Responsible!



se17a-yagel

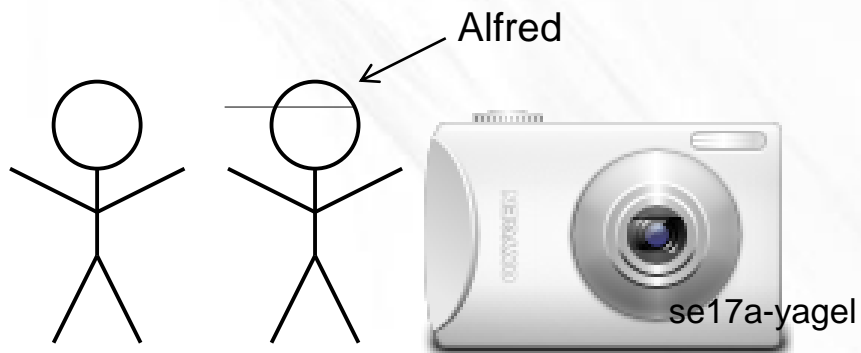
Snapshots

- Alfred, the photographer



Snapshots

- Alfred, the photographer



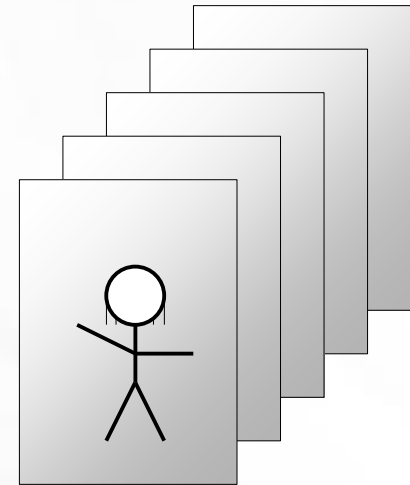
Snapshots

- Alfred, the photographer
- Hazel and her daughter



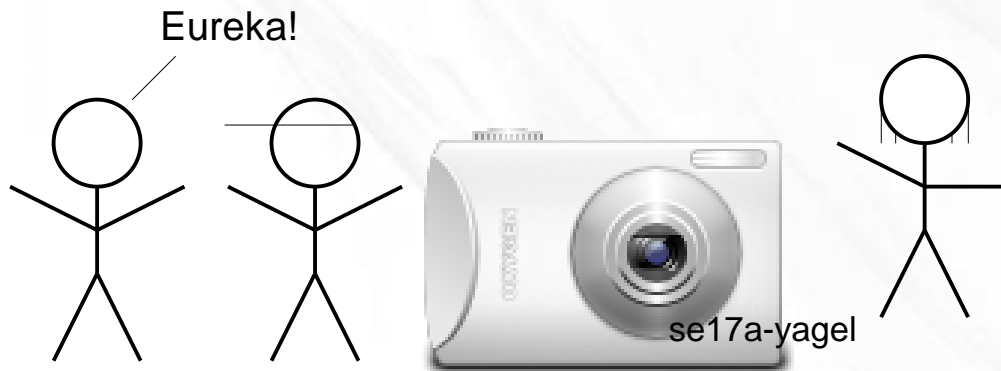
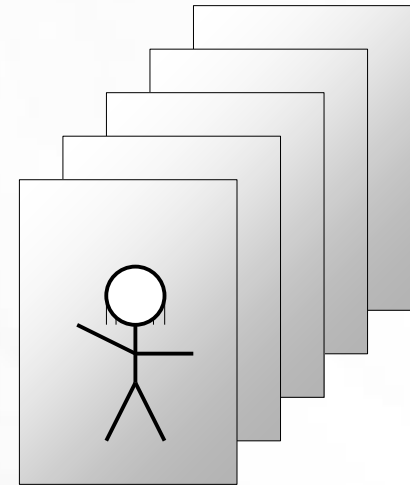
Snapshots

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



Snapshots

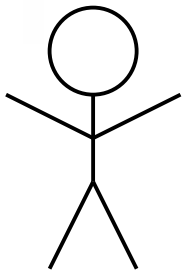
- Alfred, the photographer
- Hazel and her daughter
 - Remember what the daughter was like at each different stage



Snapshots



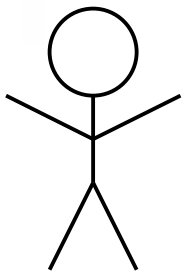
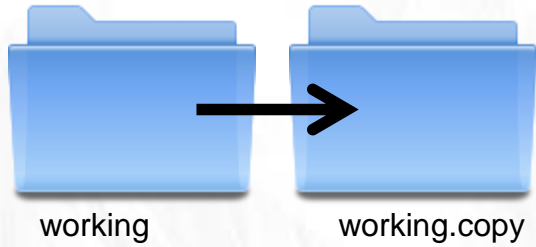
working



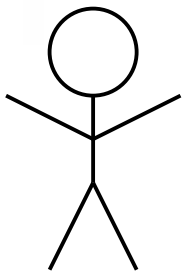
se17a-yagel

35

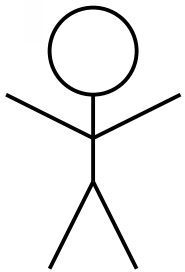
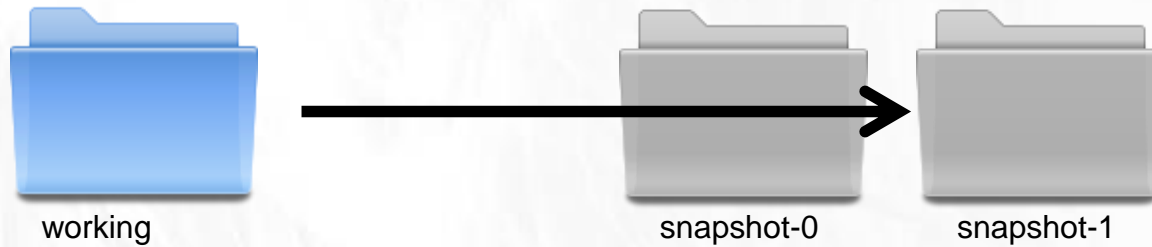
Snapshots



Snapshots



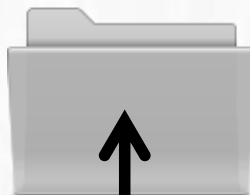
Snapshots



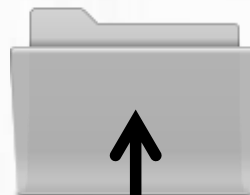
Snapshots



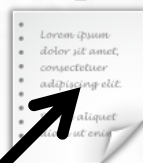
working



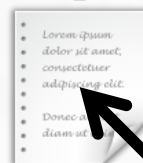
snapshot-0



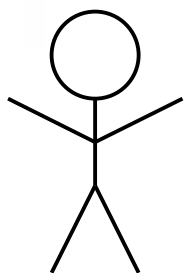
snapshot-1



message



message



2009-05-20 12:34:56

Initial version

se17a-yagel

2009-05-21 23:45:01

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

Branches



working



snapshot-0

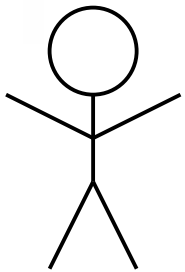


snapshot-1

...



snapshot-99



se17a-yagel

40

Branches



working



snapshot-0

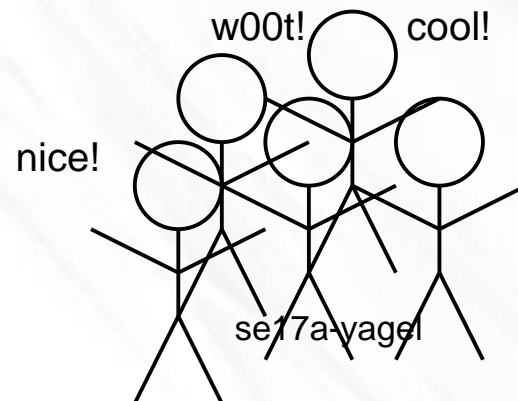
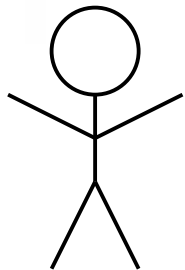


snapshot-1

...



snapshot-99



Branches



working



snapshot-0



snapshot-1

...

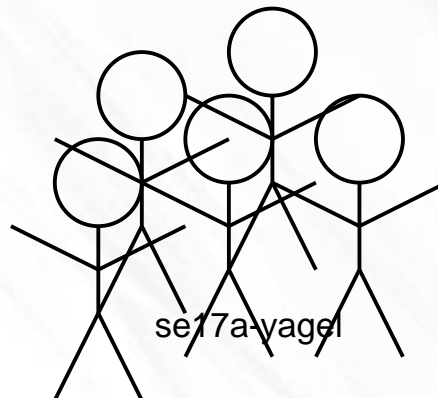
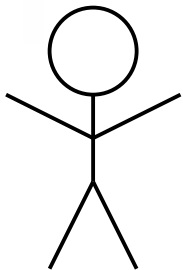


snapshot-99

...



snapshot-109



Branches



working



snapshot-0



snapshot-1

...



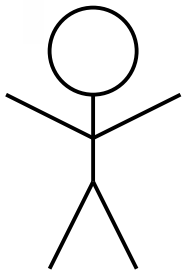
snapshot-99

...

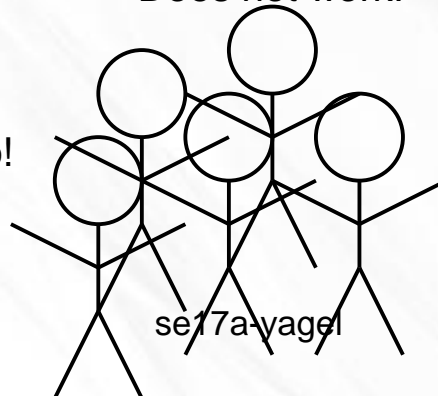


snapshot-109

Does not work!

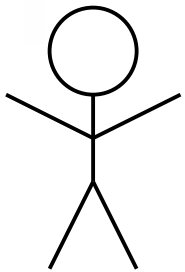
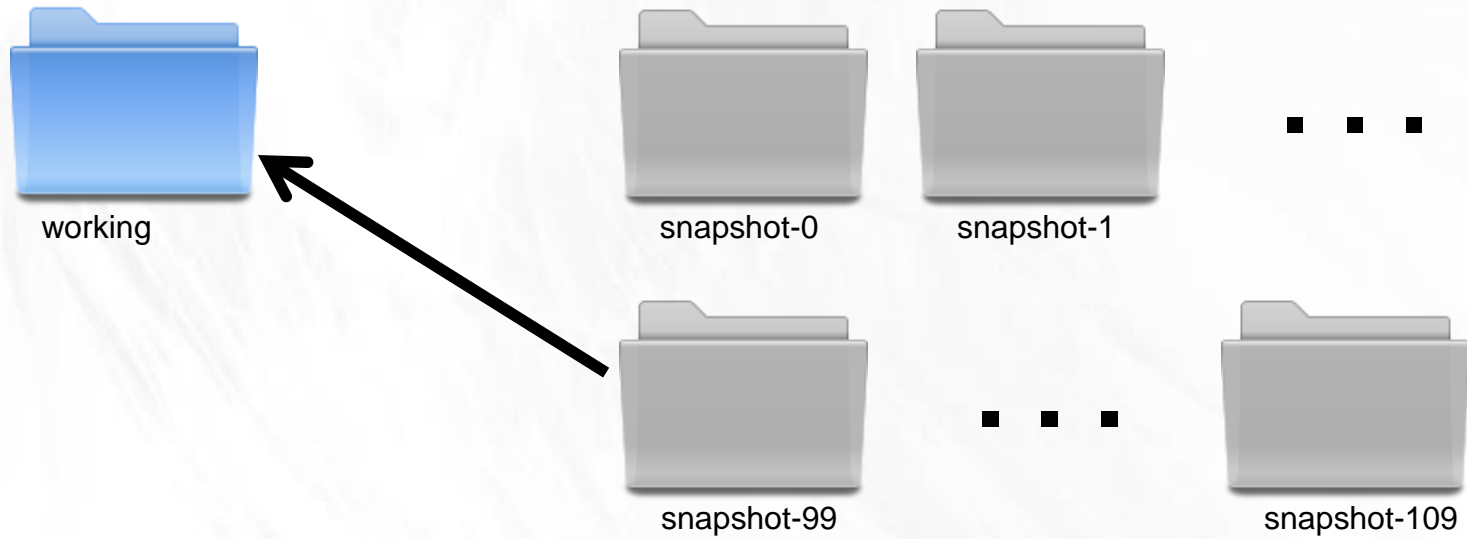


Boo!



se17a-yager

Branches



Branches



working



snapshot-0



snapshot-1

...



snapshot-99

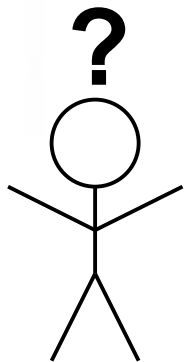
...



snapshot-109

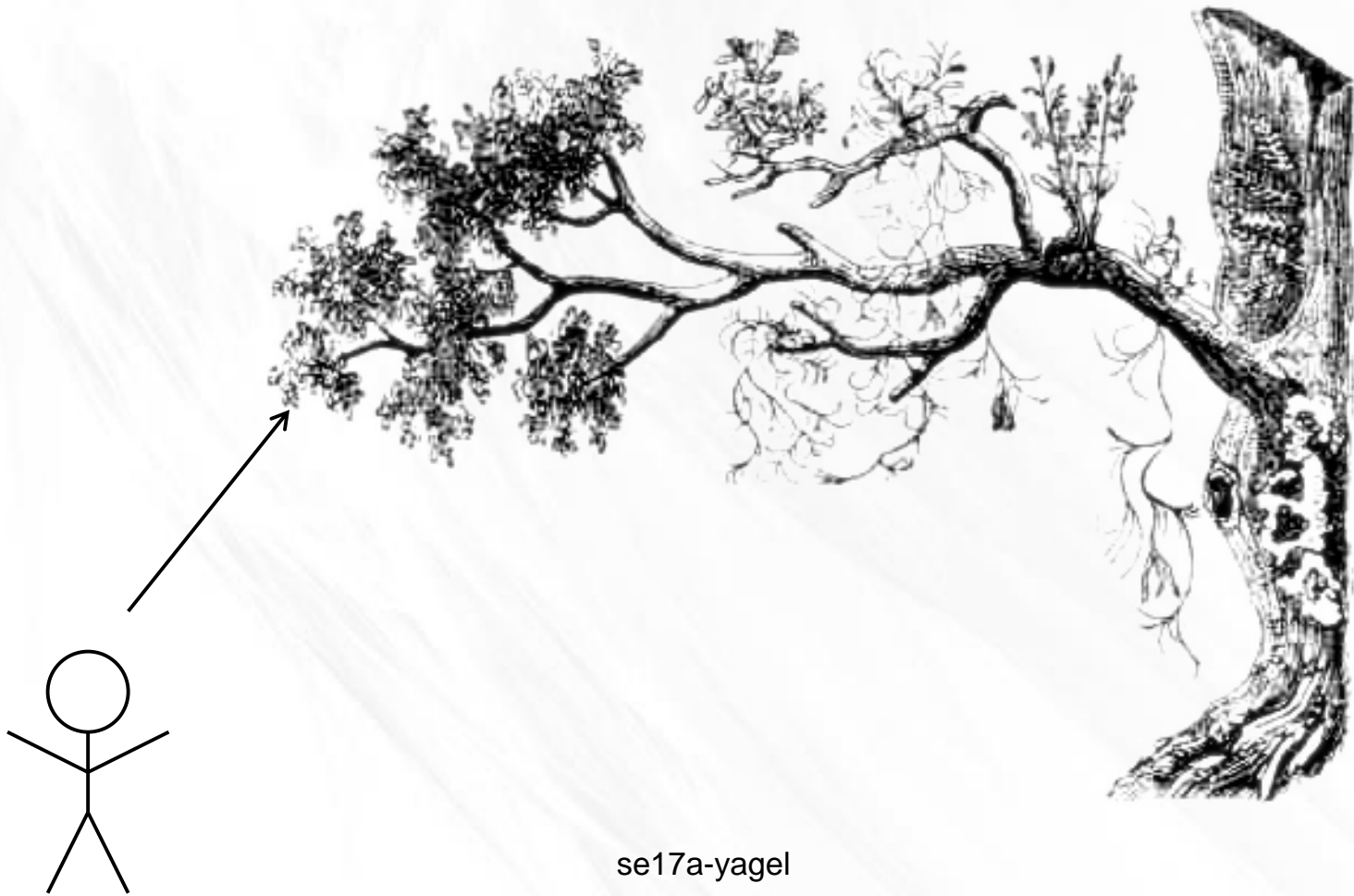


snapshot-110



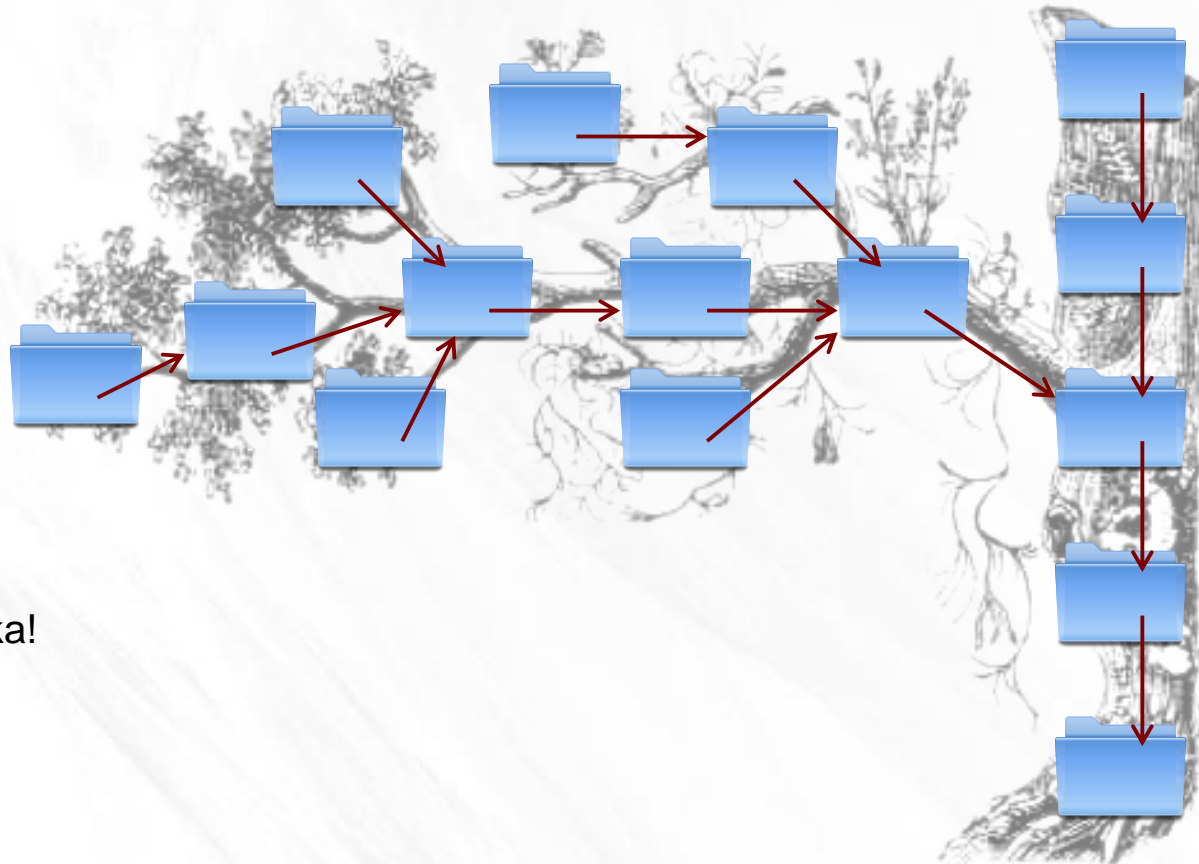
se17a-yagel

Branches

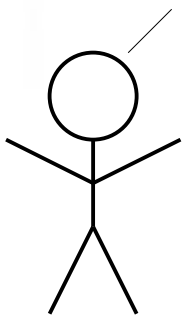


se17a-yagel

Branches

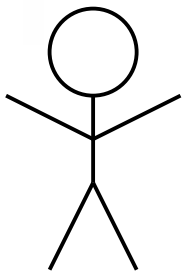
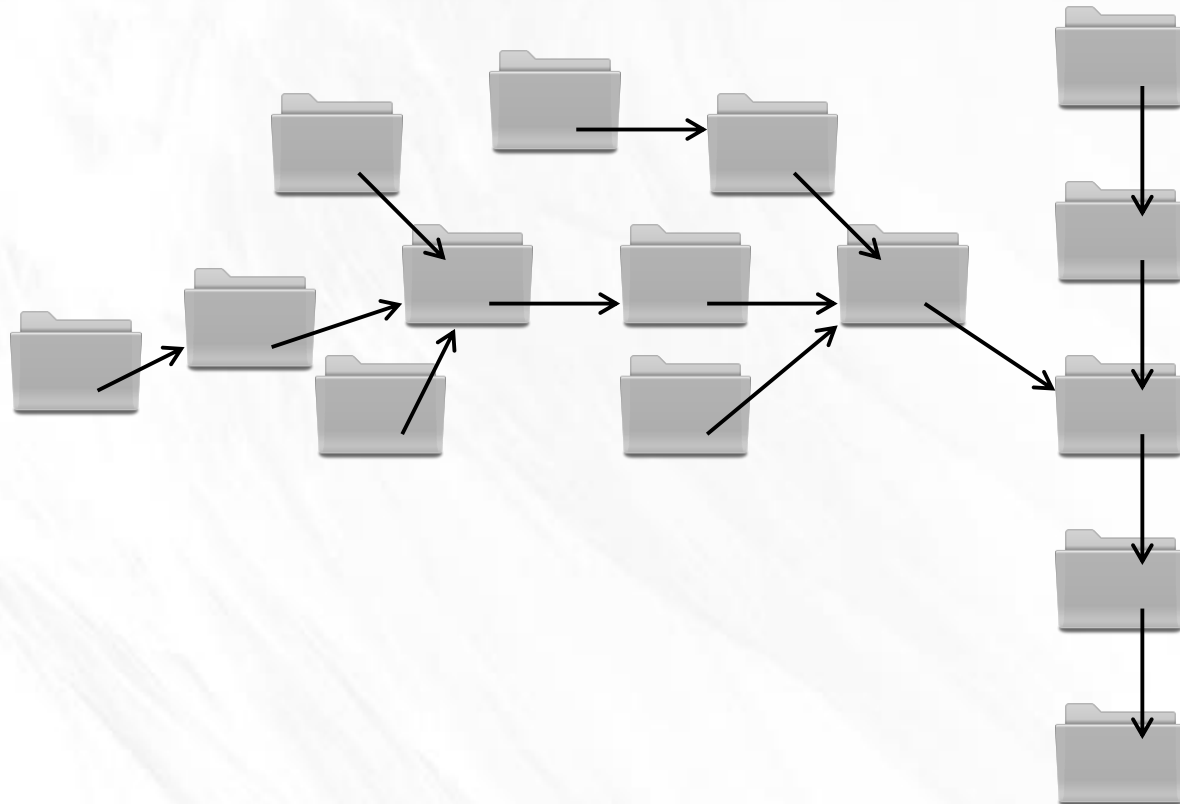


Eureka!

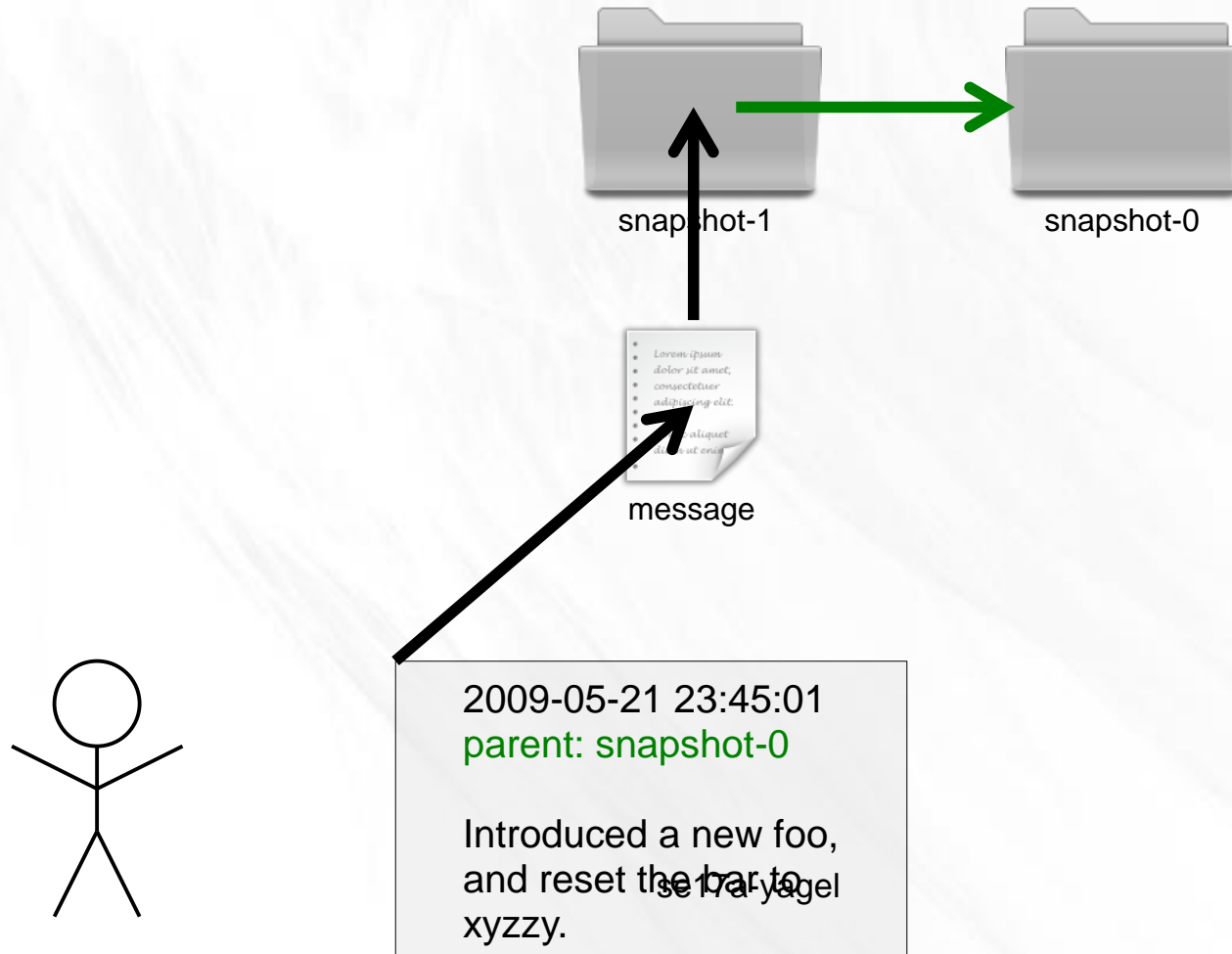


se17a-yagel

Branches



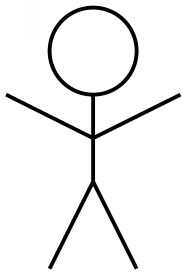
Branches



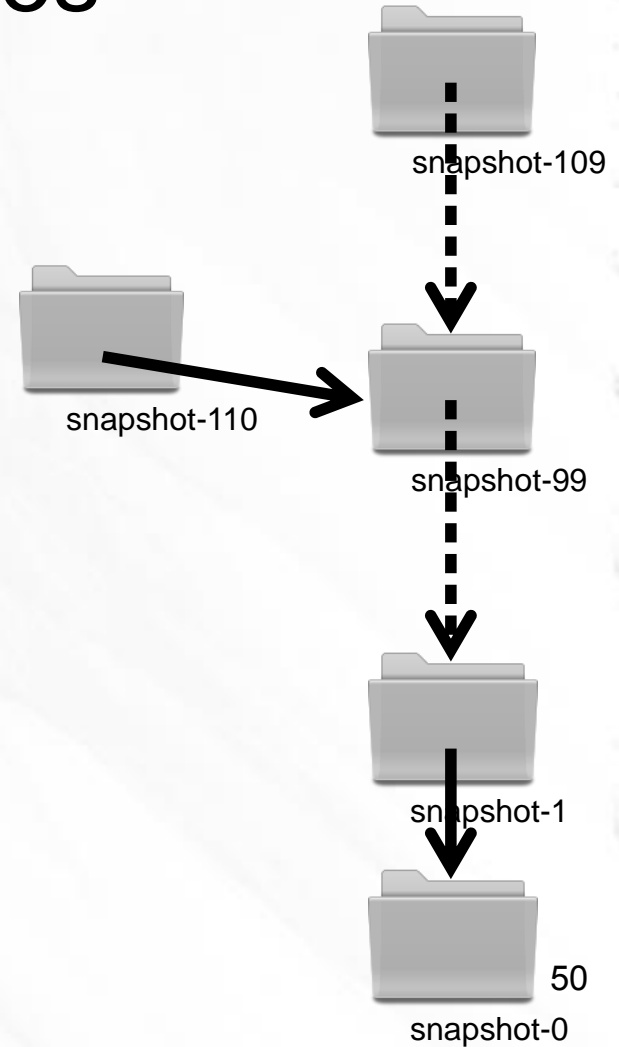
Branch Names



working



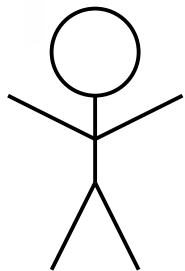
se17a-yagel



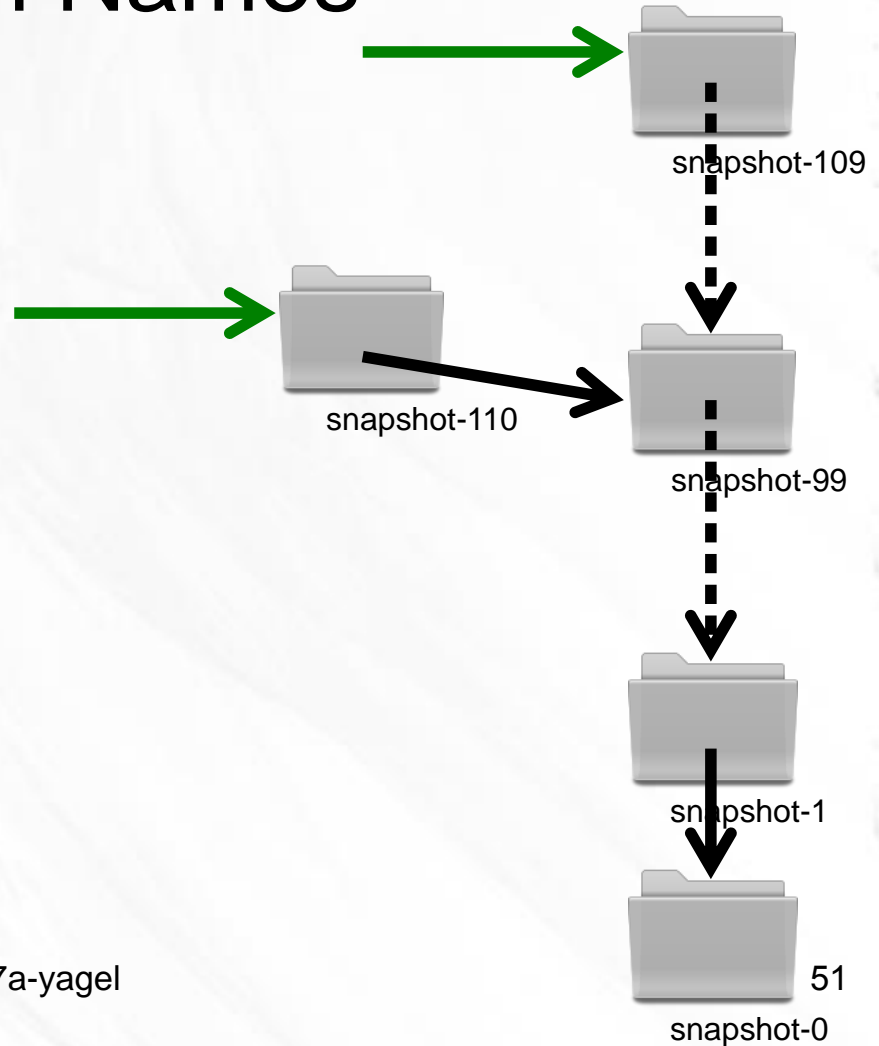
Branch Names



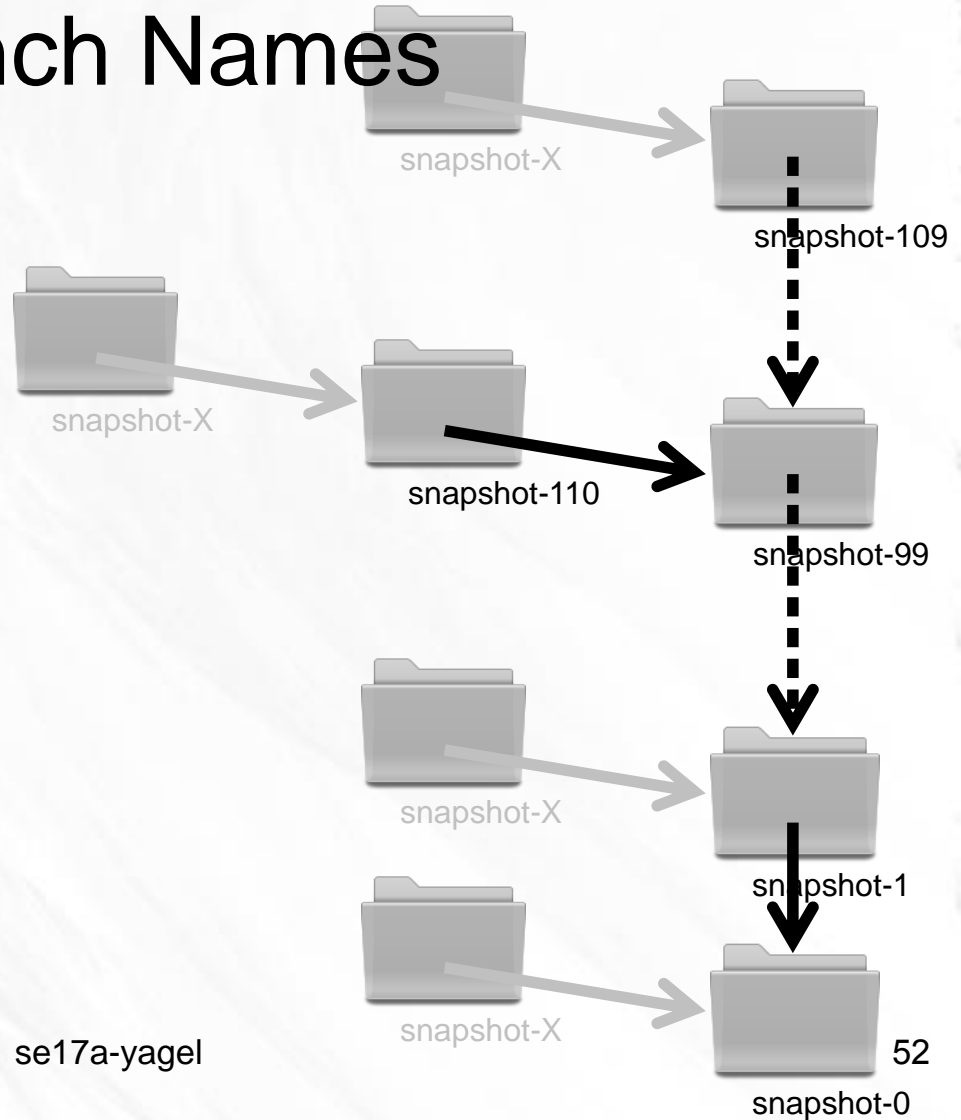
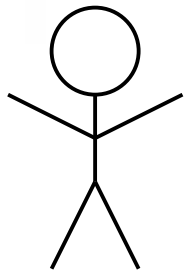
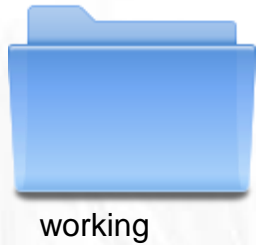
working



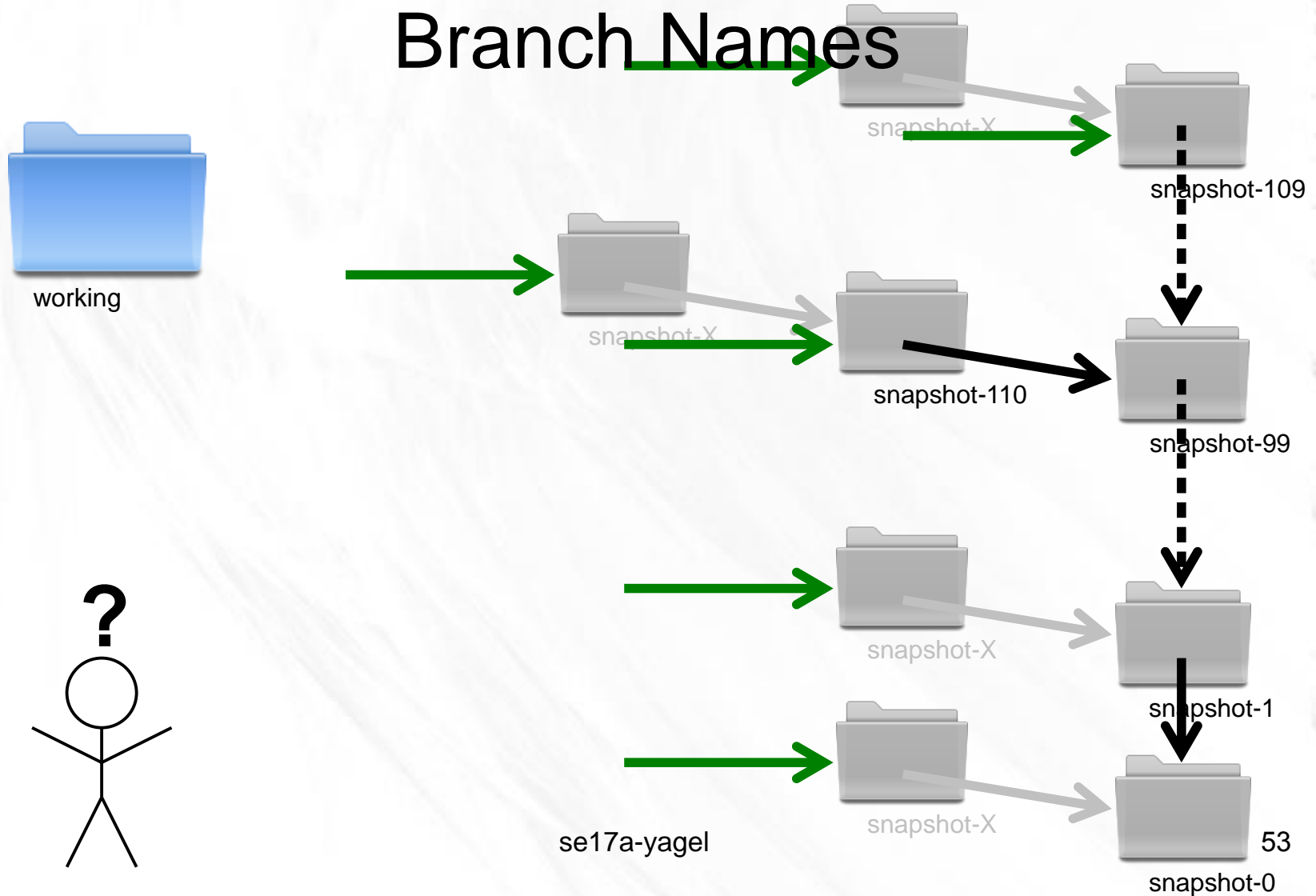
se17a-yagel



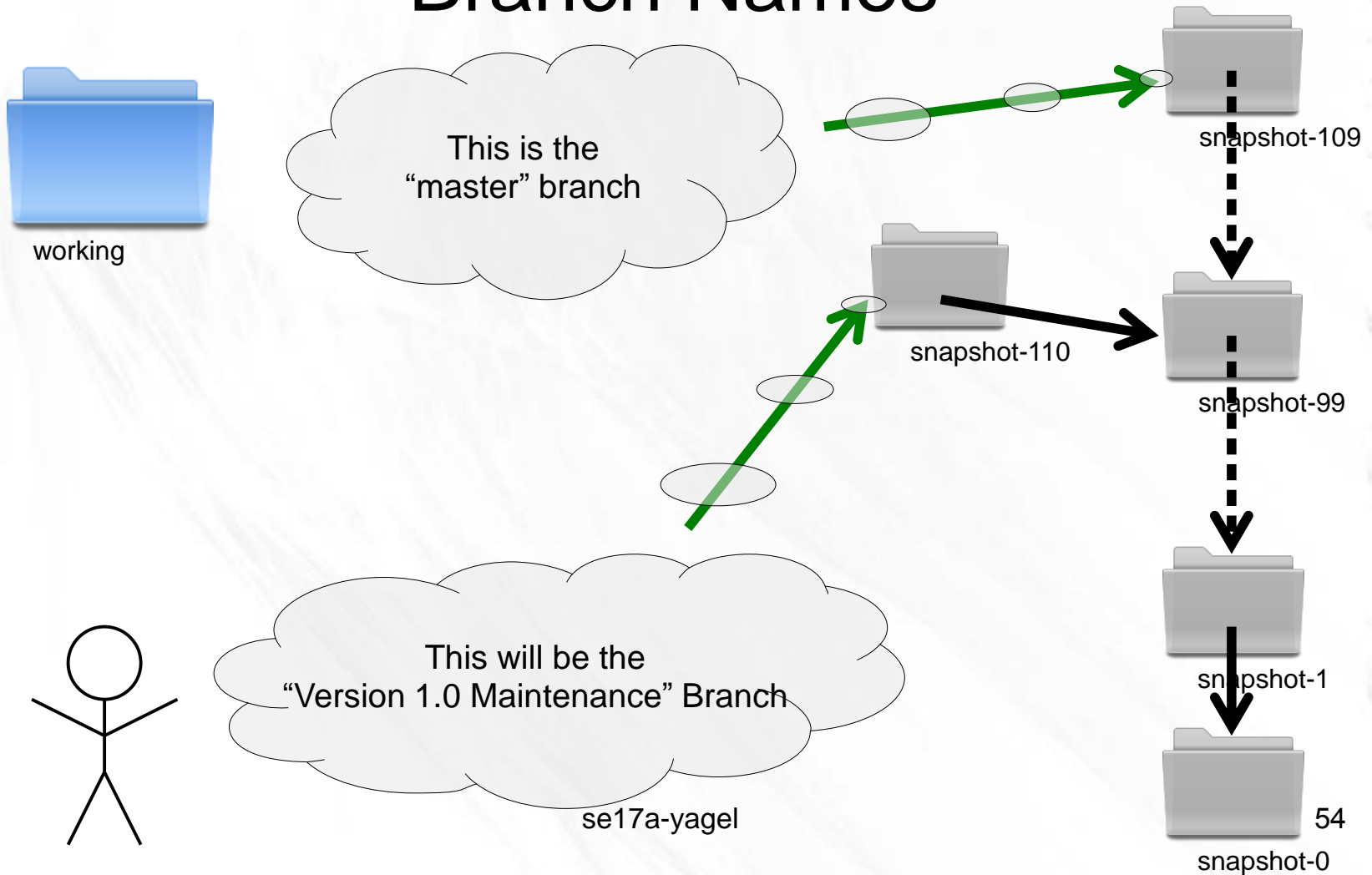
Branch Names



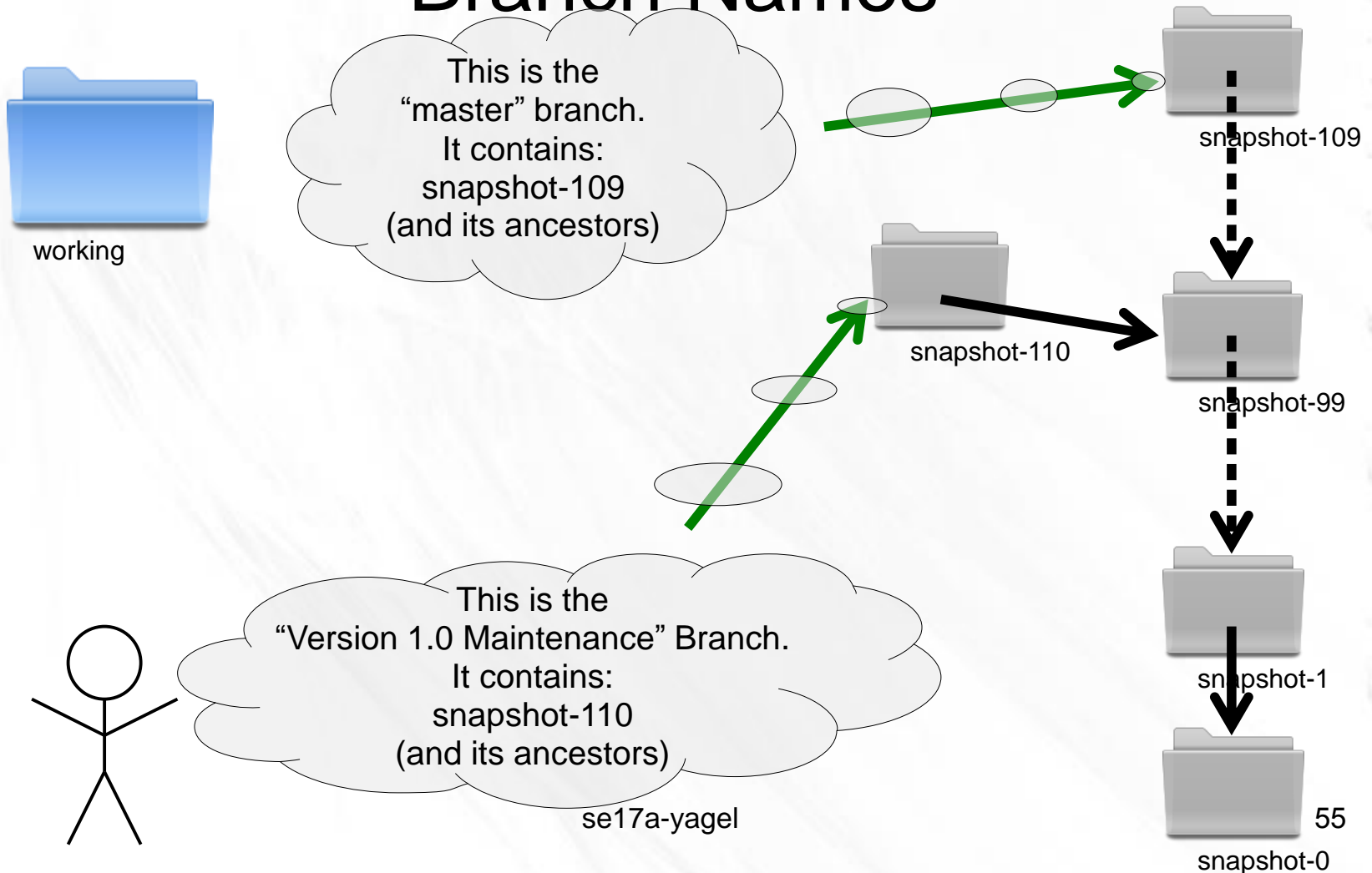
Branch Names



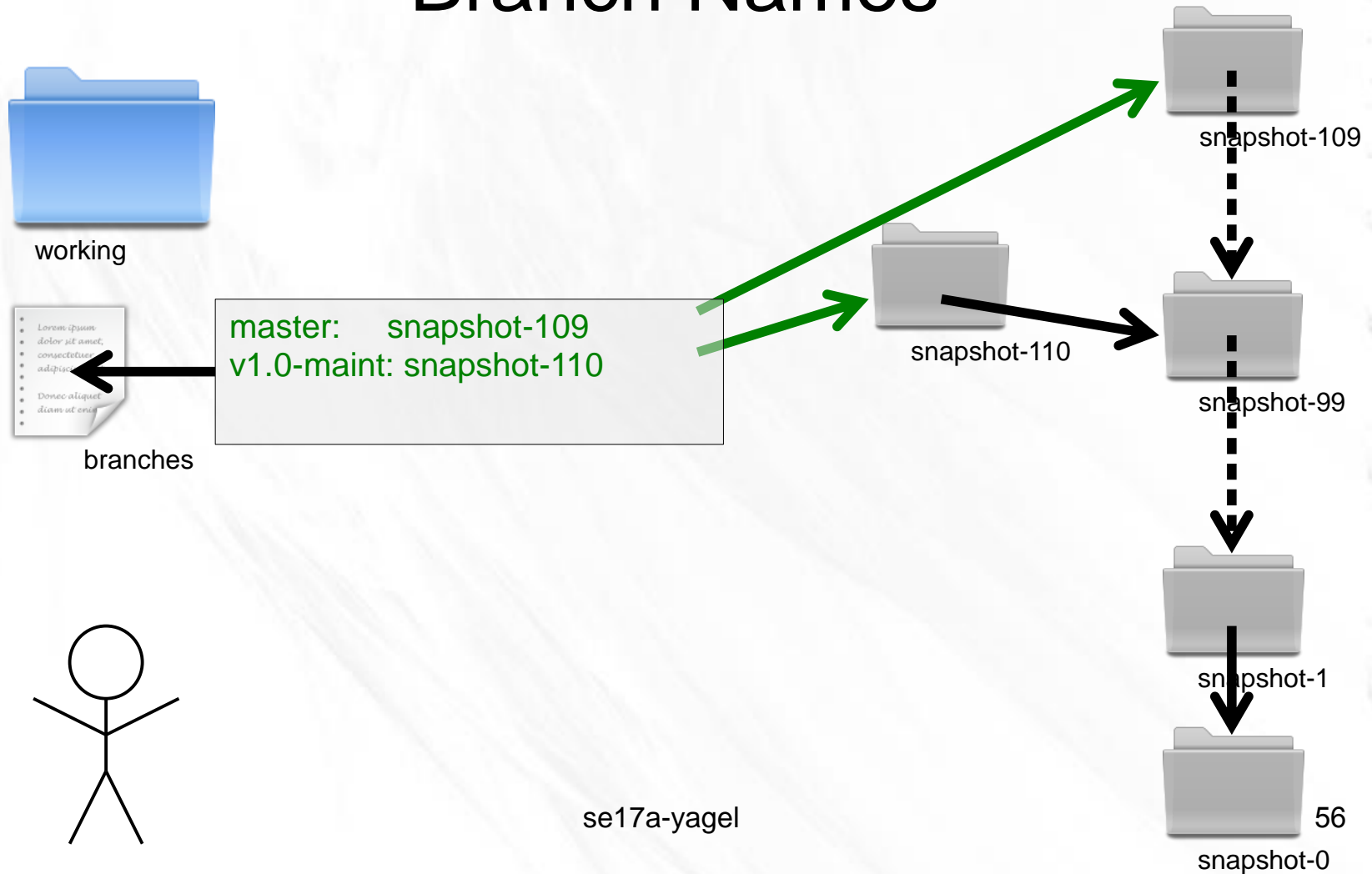
Branch Names



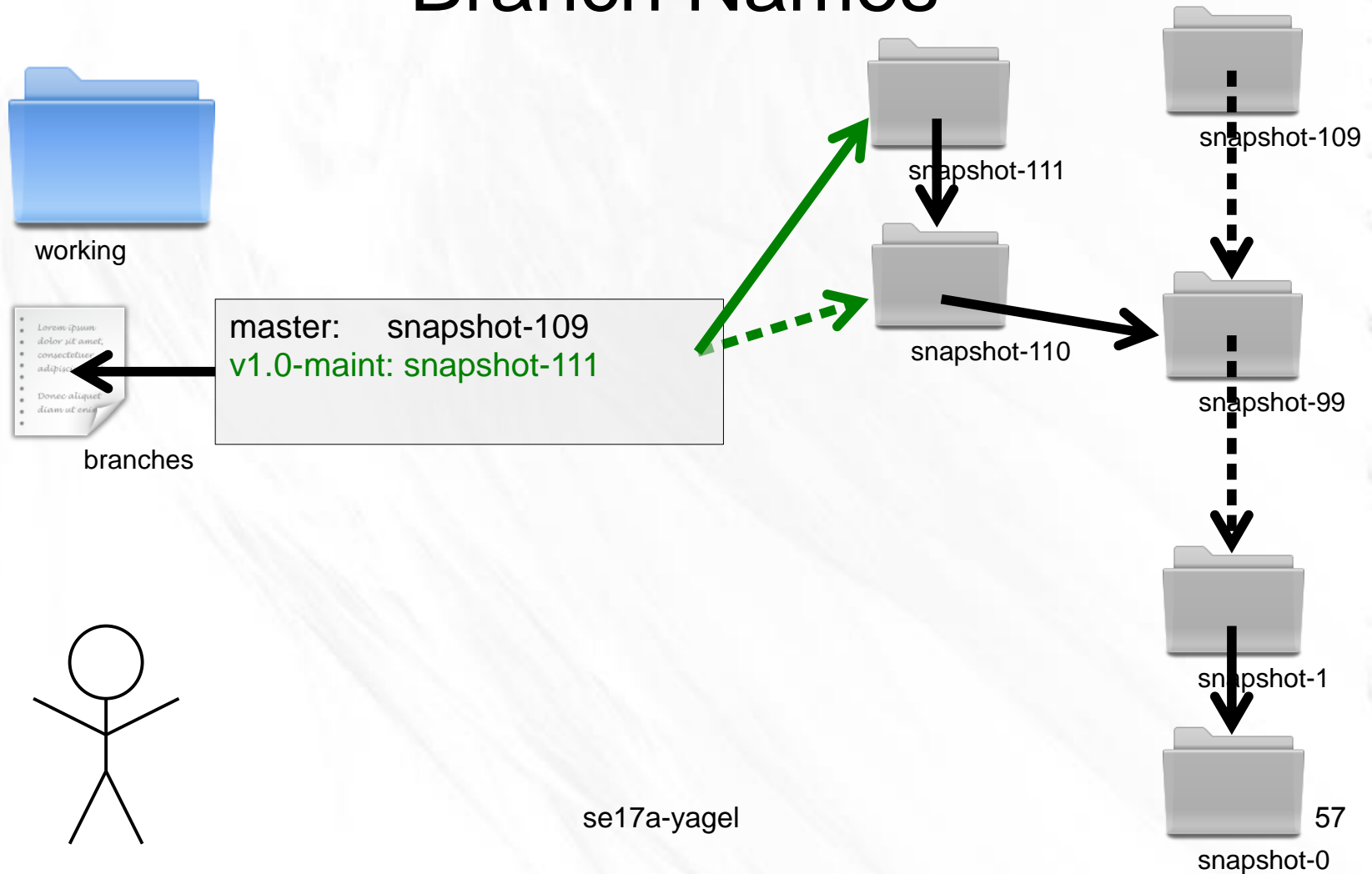
Branch Names



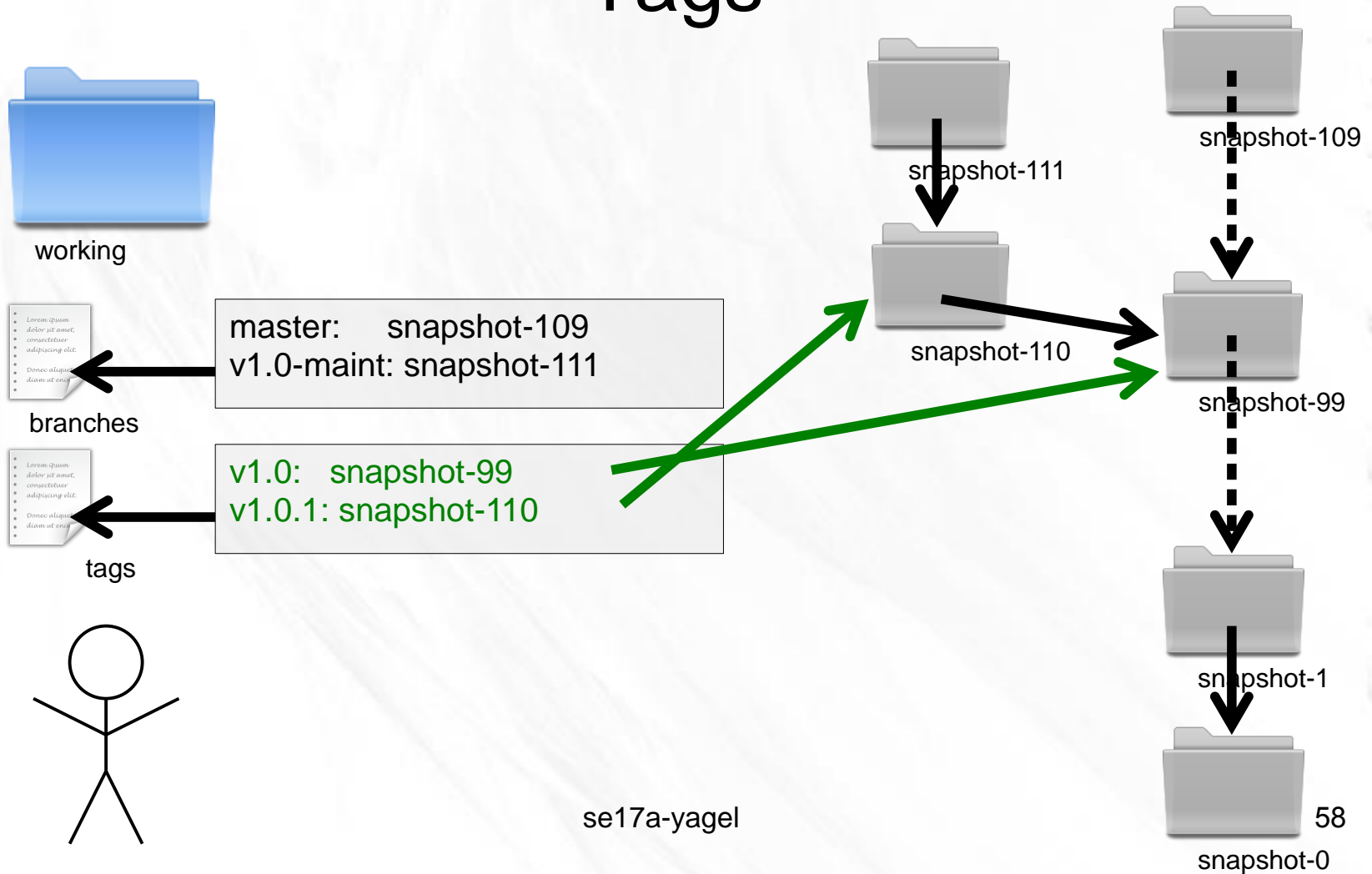
Branch Names



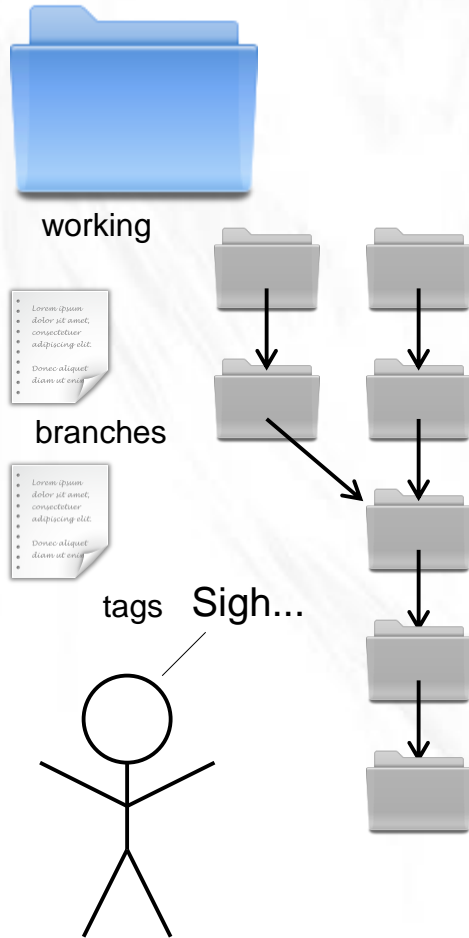
Branch Names



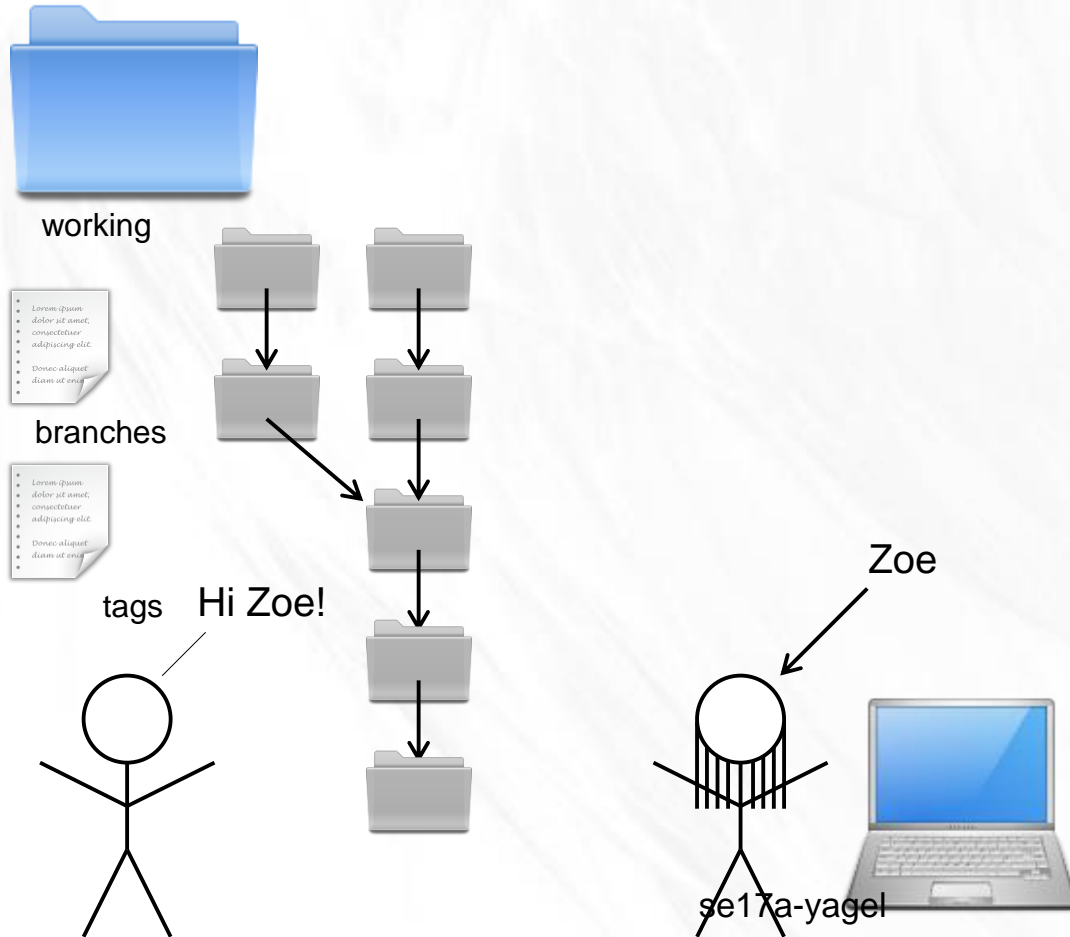
Tags



Distributed



Distributed



Distributed



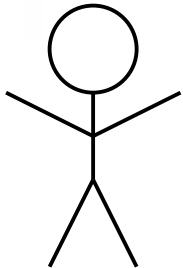
working



branches



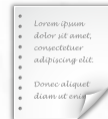
tags



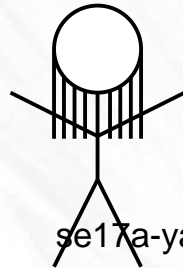
working



branches



tags



se17a-yagel

Distributed



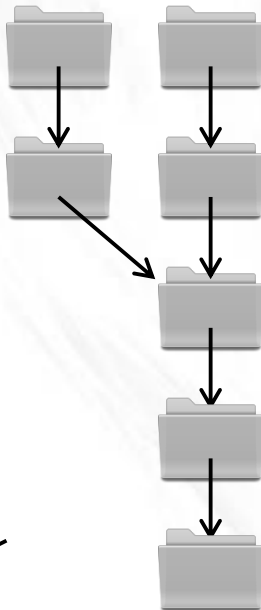
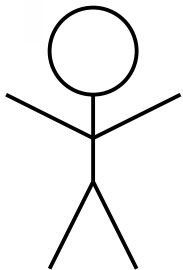
working



branches



tags



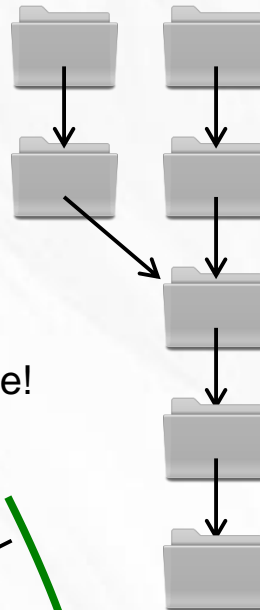
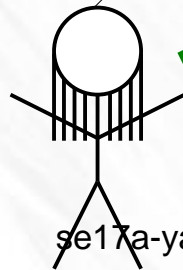
working



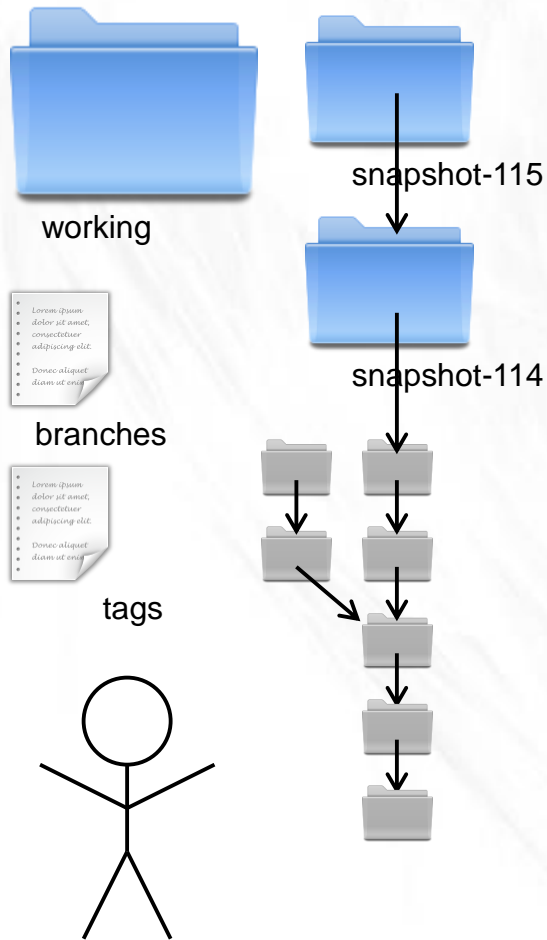
branches



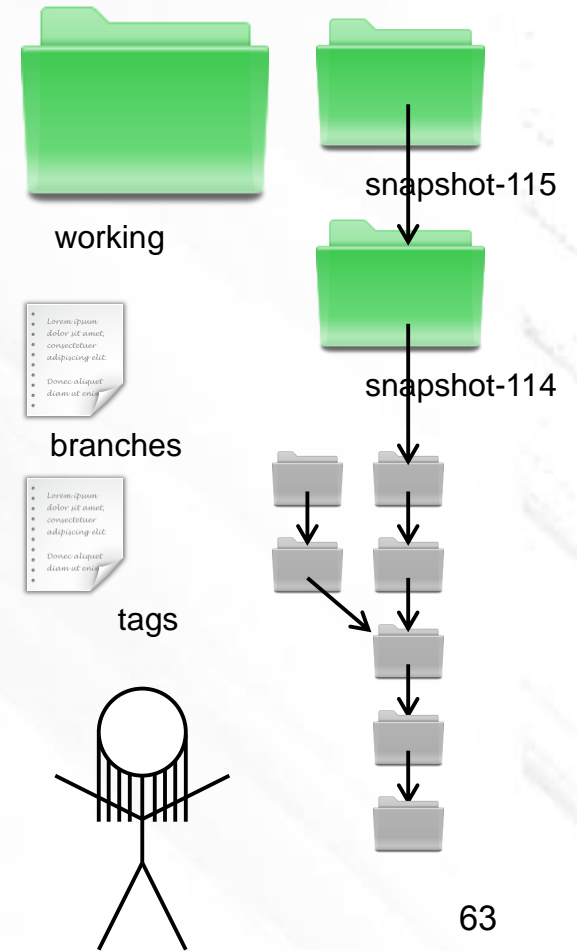
tags



Distributed



se17a-yagel

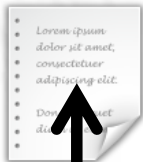


63

Distributed



snapshot-114



message

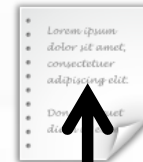
2009-05-22 12:12:12
parent: snapshot-113
author: Me <me@me.me>

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

se17a-yagel



snapshot-114

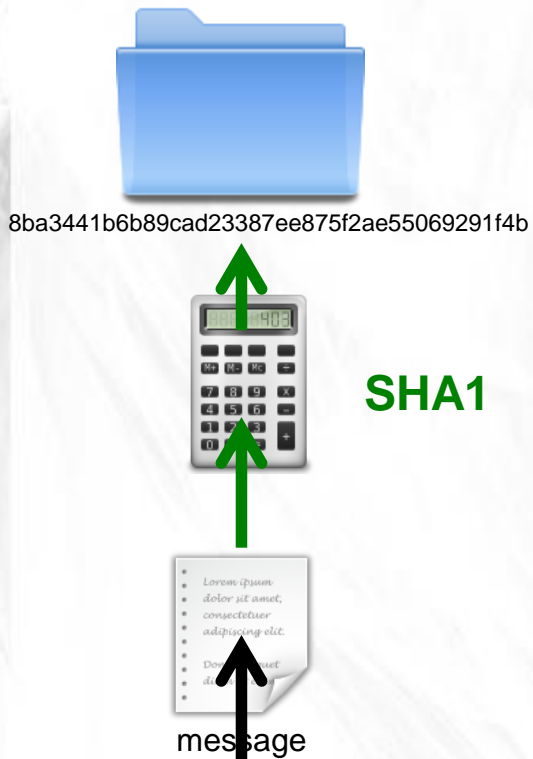


message

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

Introduced a new foo,
and reset the bar to 64
xyzyy.

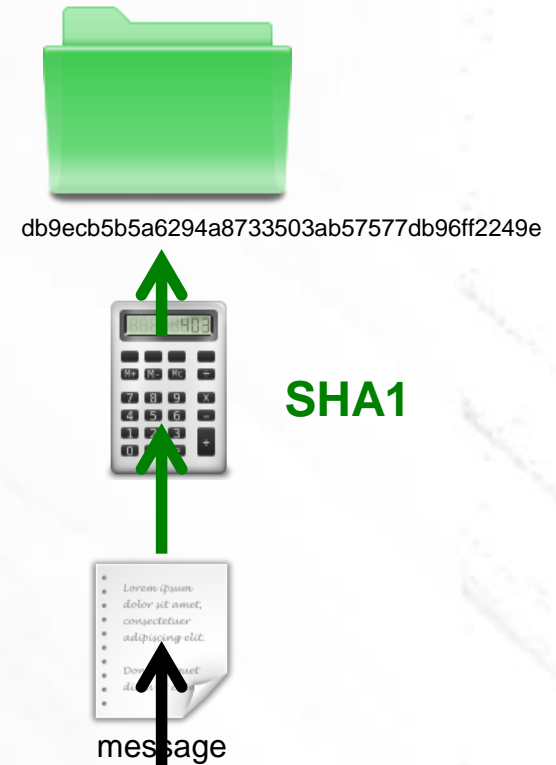
Distributed



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

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

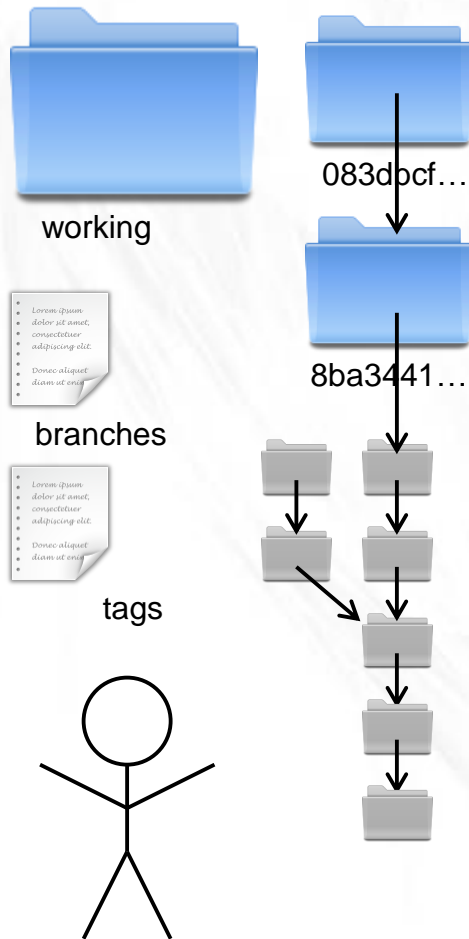
se17a-yagel



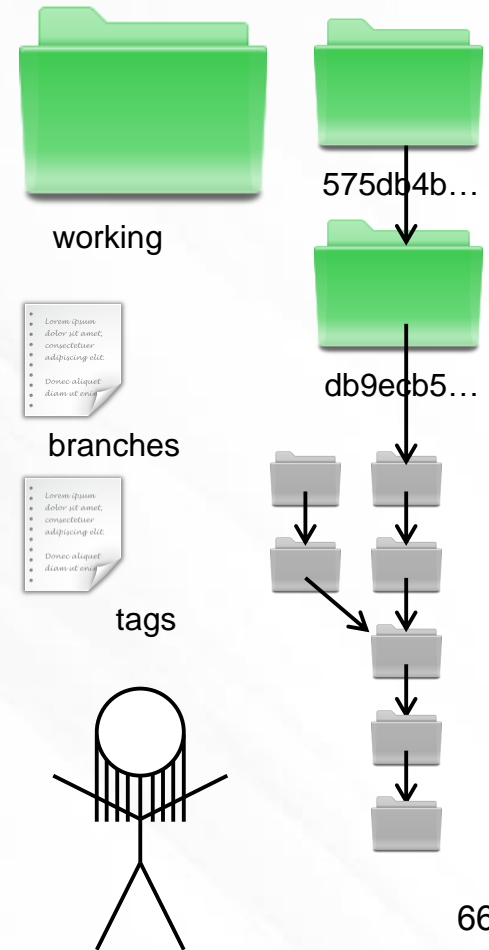
2009-05-21 23:45:01
parent: snapshot-113
author: Zoe <zoe@z.oe>

Introduced a new foo, 65
and reset the bar to
xyzyz.

Distributed

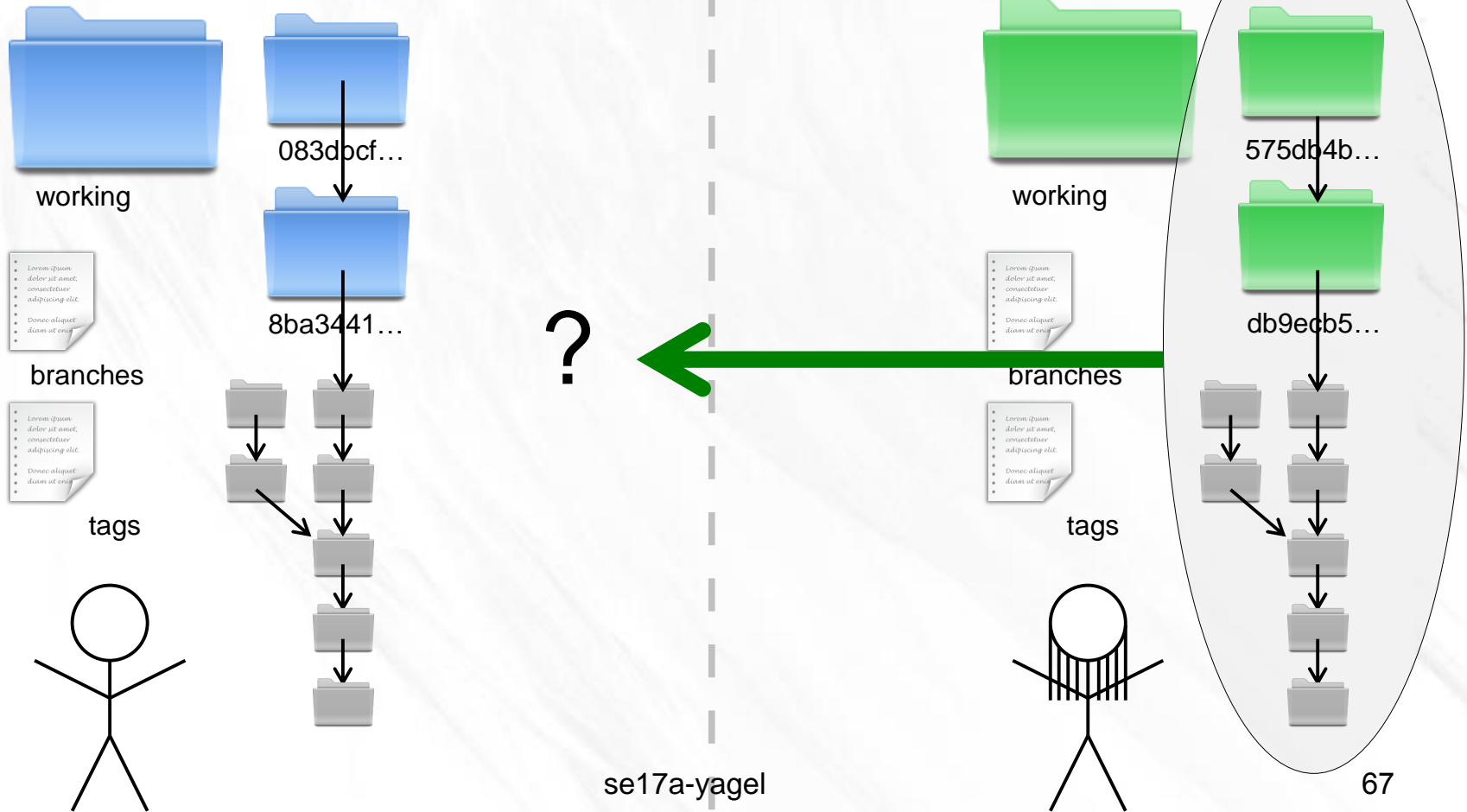


se17a-yagel

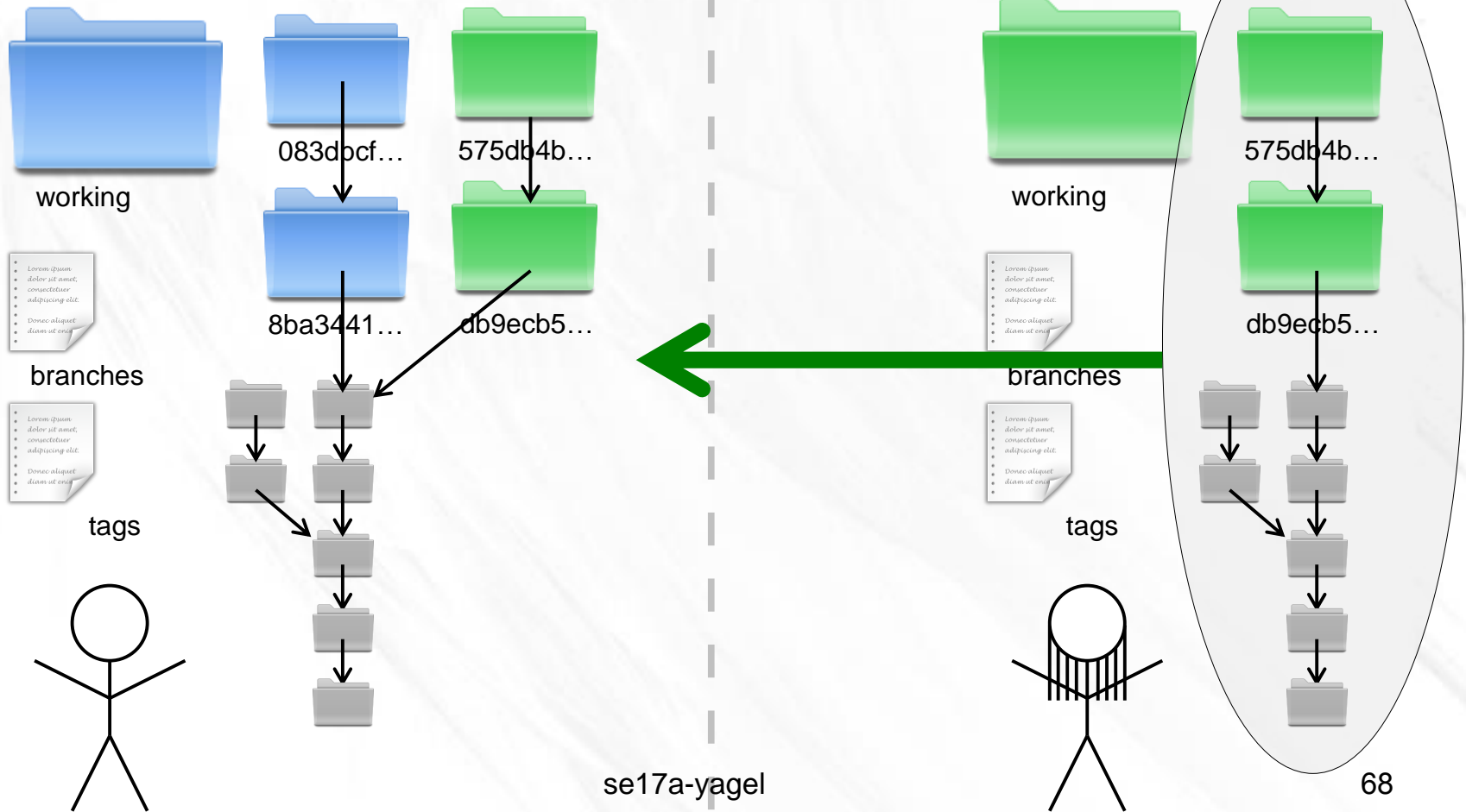


66

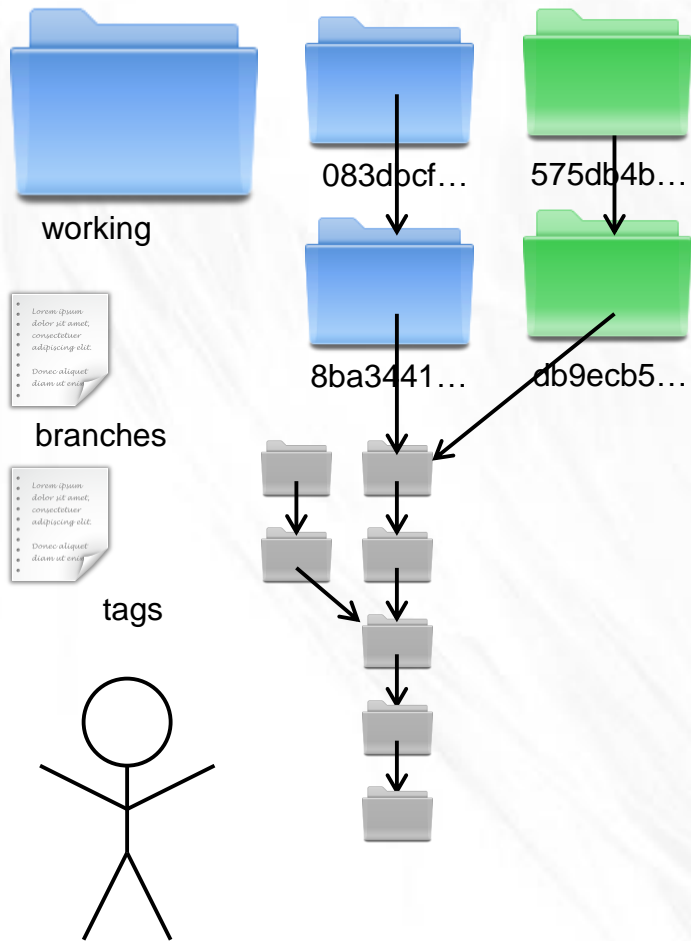
Distributed



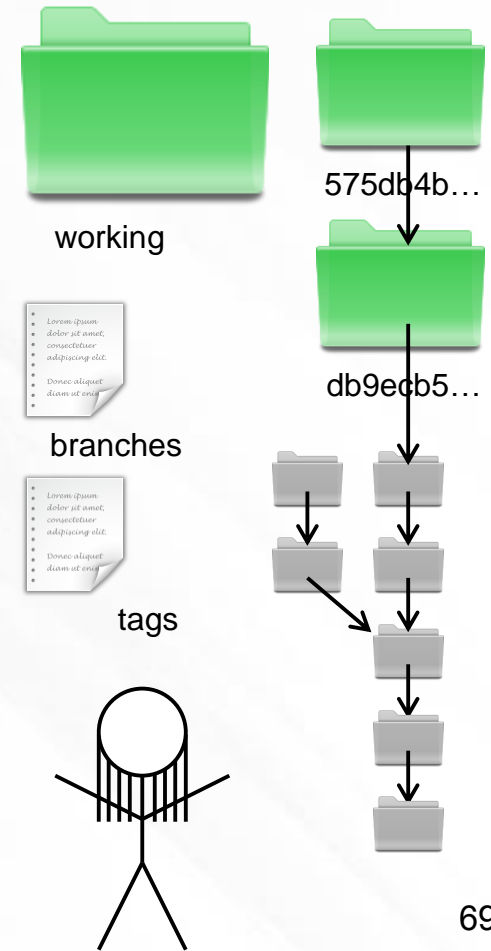
Distributed



Distributed

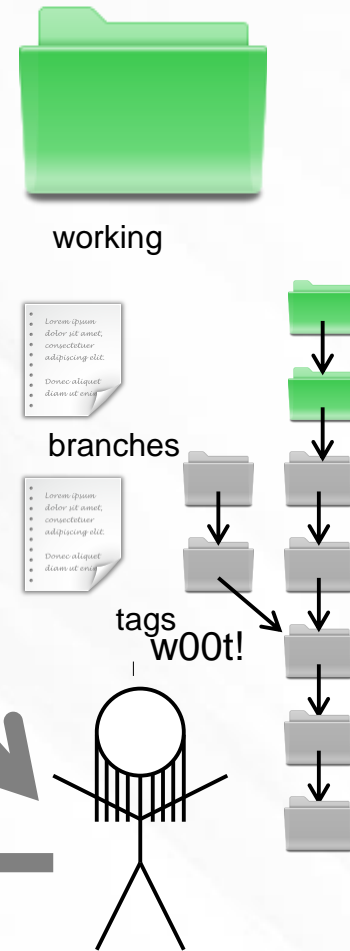
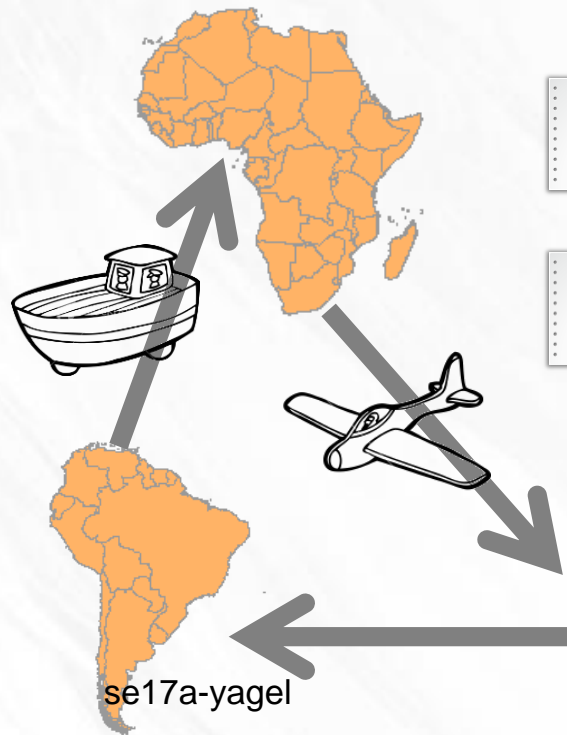
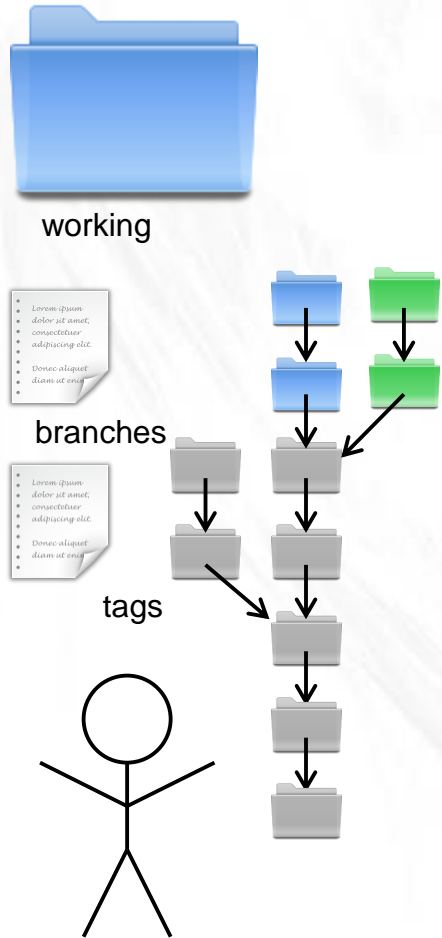


se17a-yagel

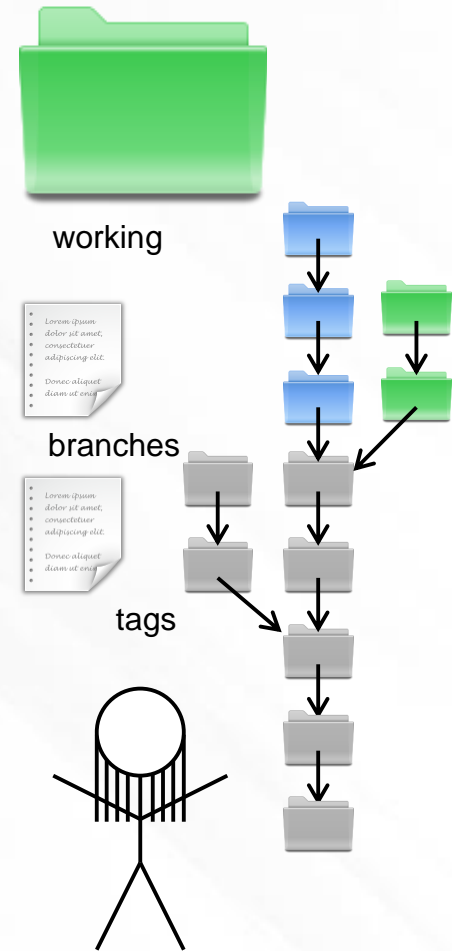
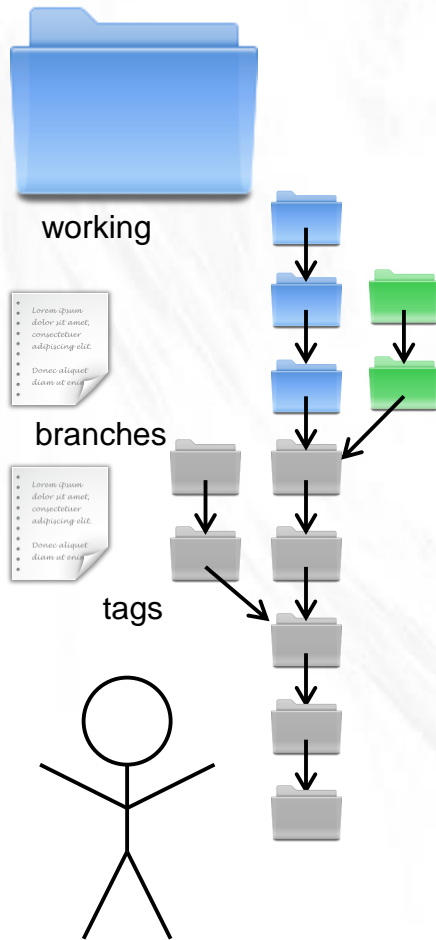


69

Offline



Offline



se17a-yagel

(simpler drawings)

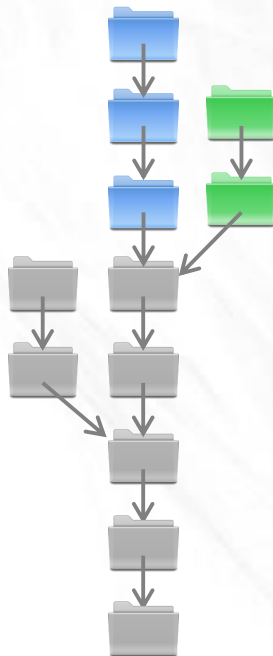
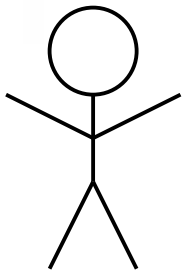
working



branches



tags



se17a-yagel

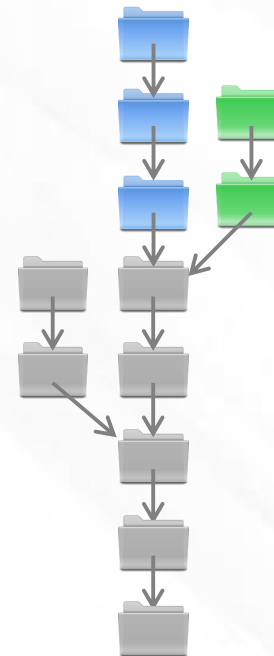
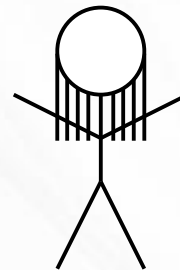
working



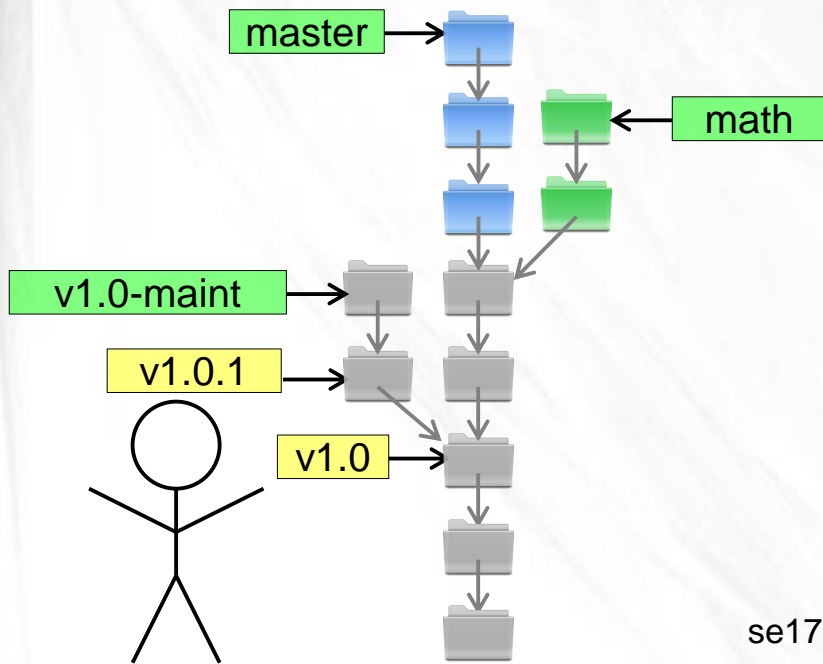
branches



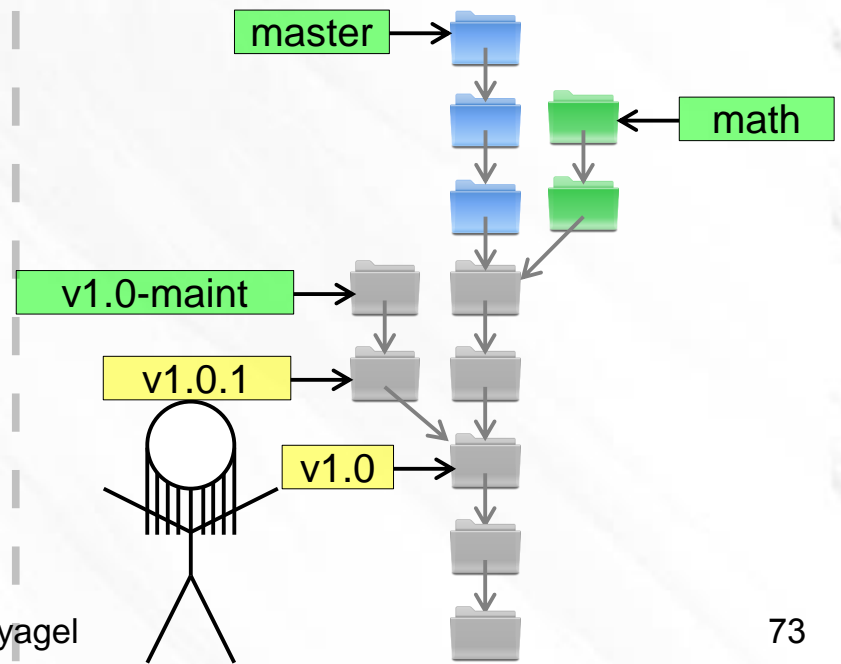
tags



(simpler drawings)

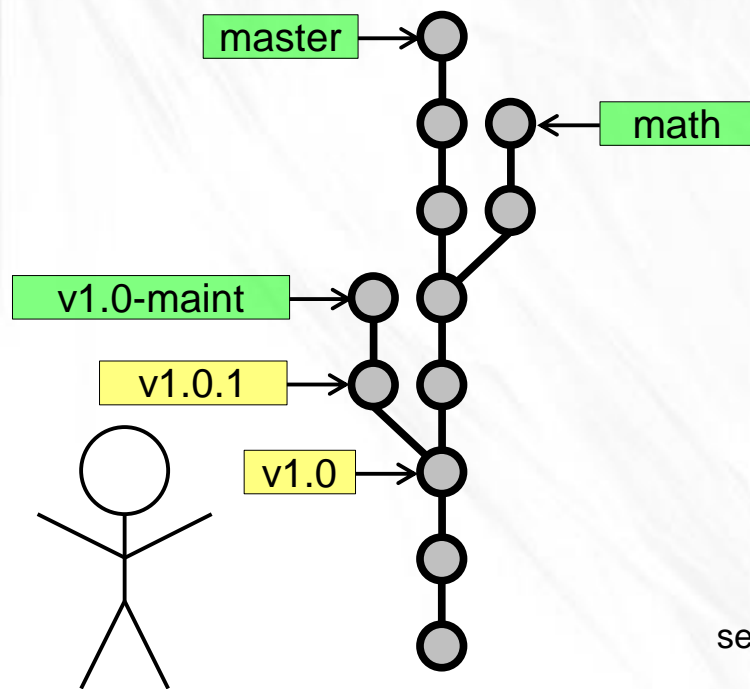


se17a-yagel

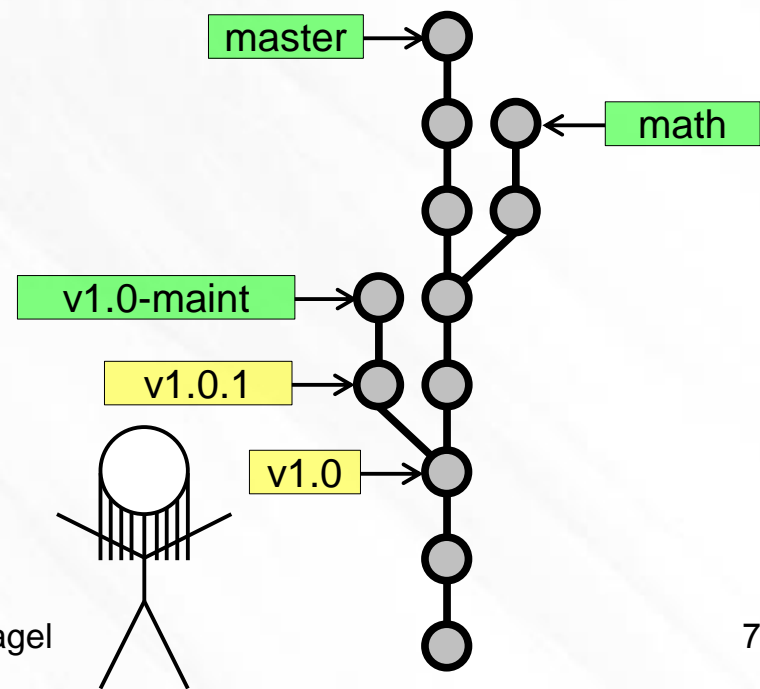


73

(simpler drawings)

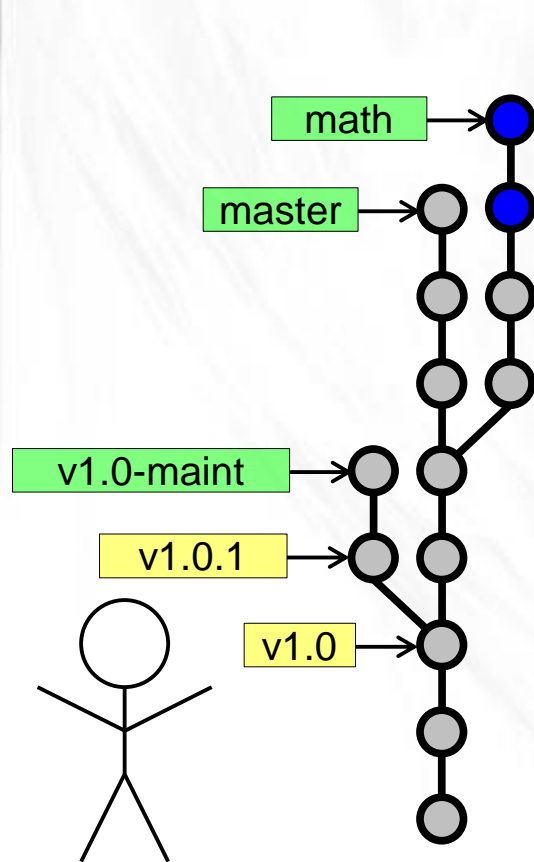


se17a-yagel

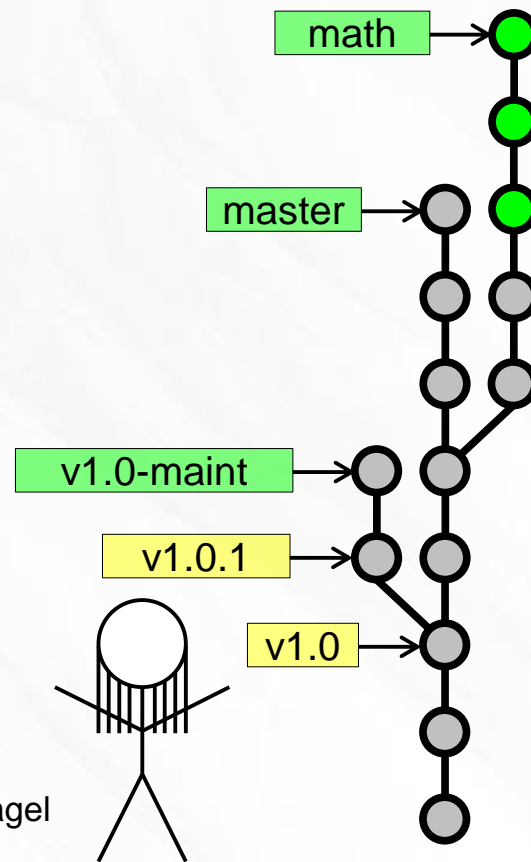


74

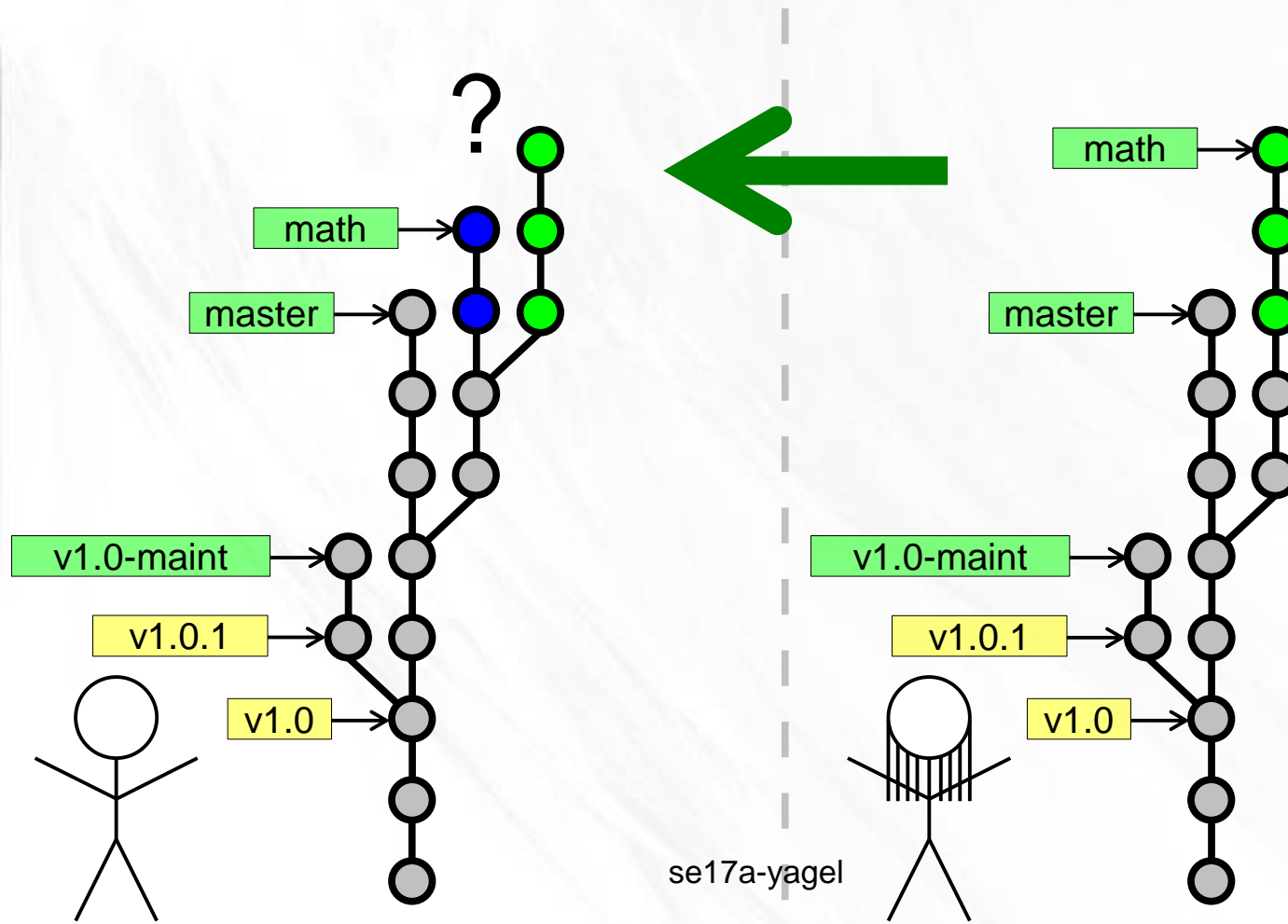
Merges



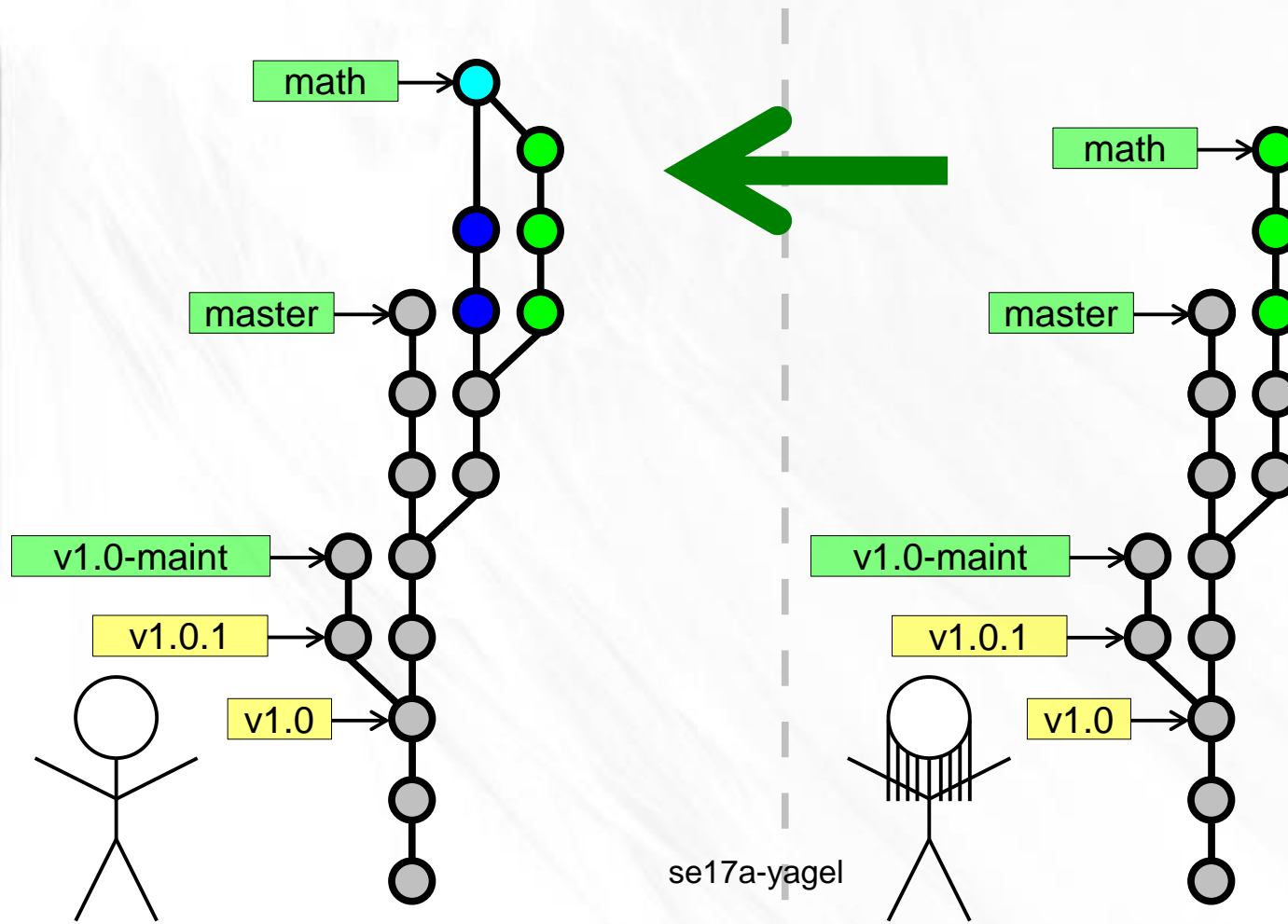
se17a-yagel



Merges

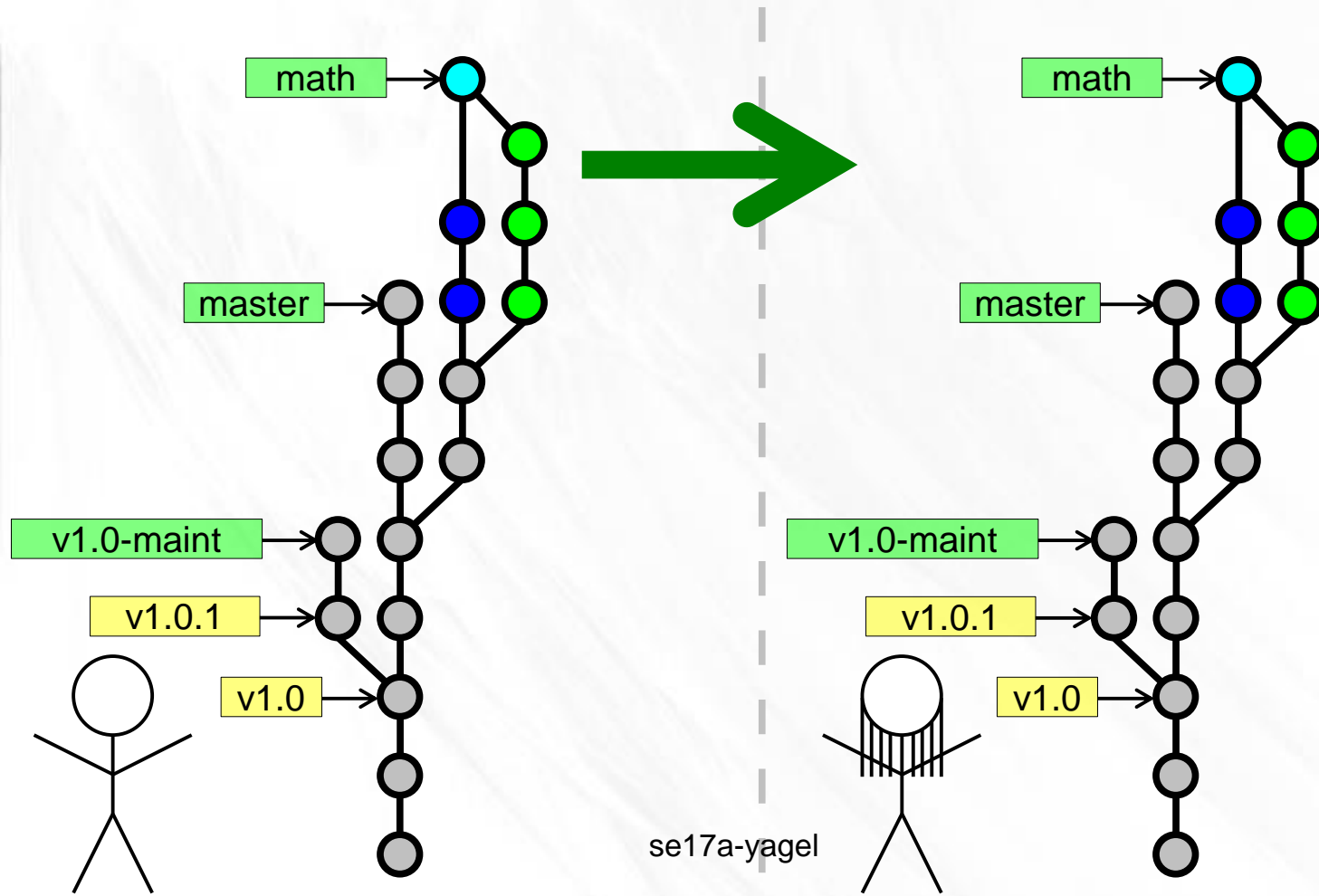


Merges

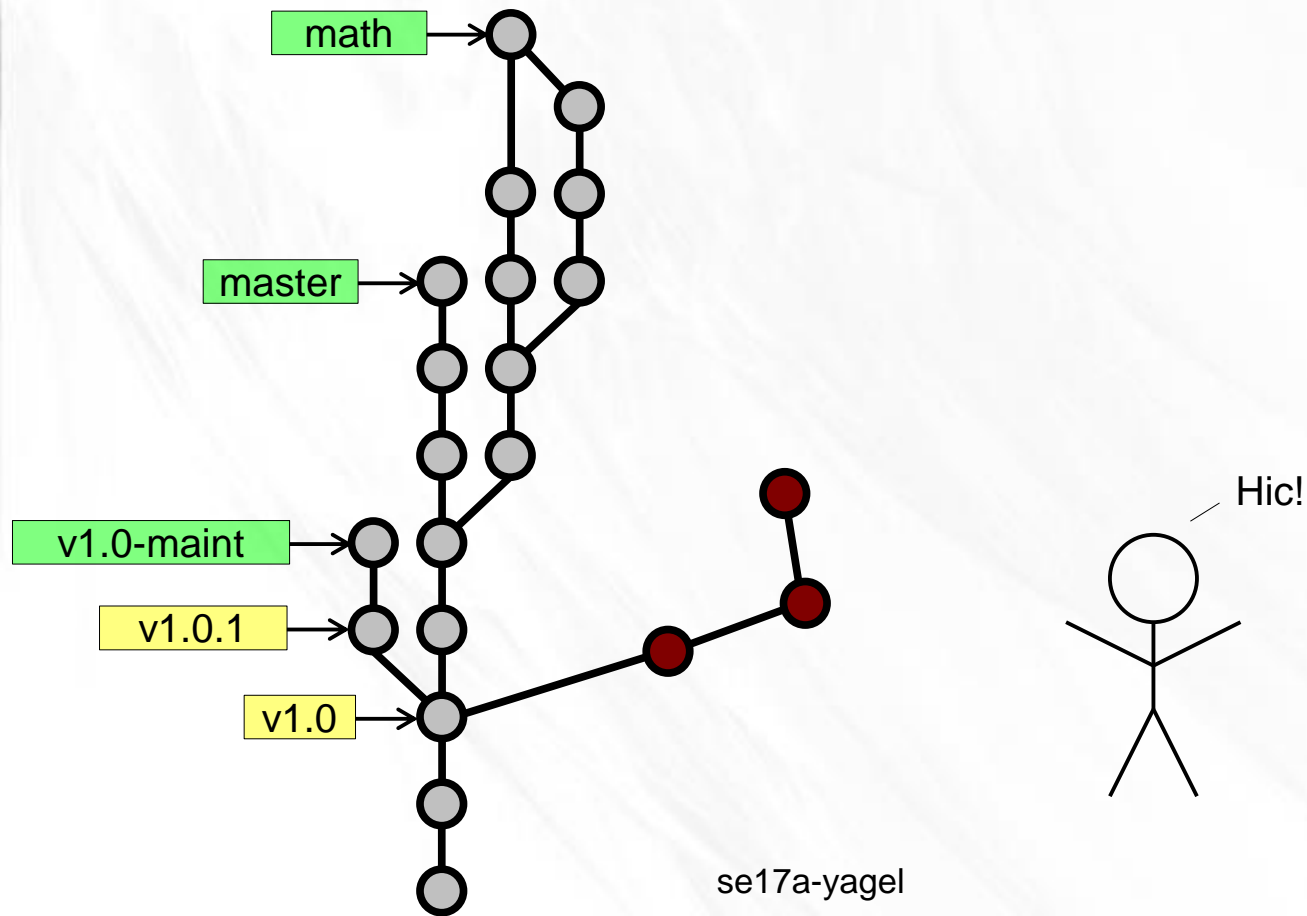


se17a-yagel

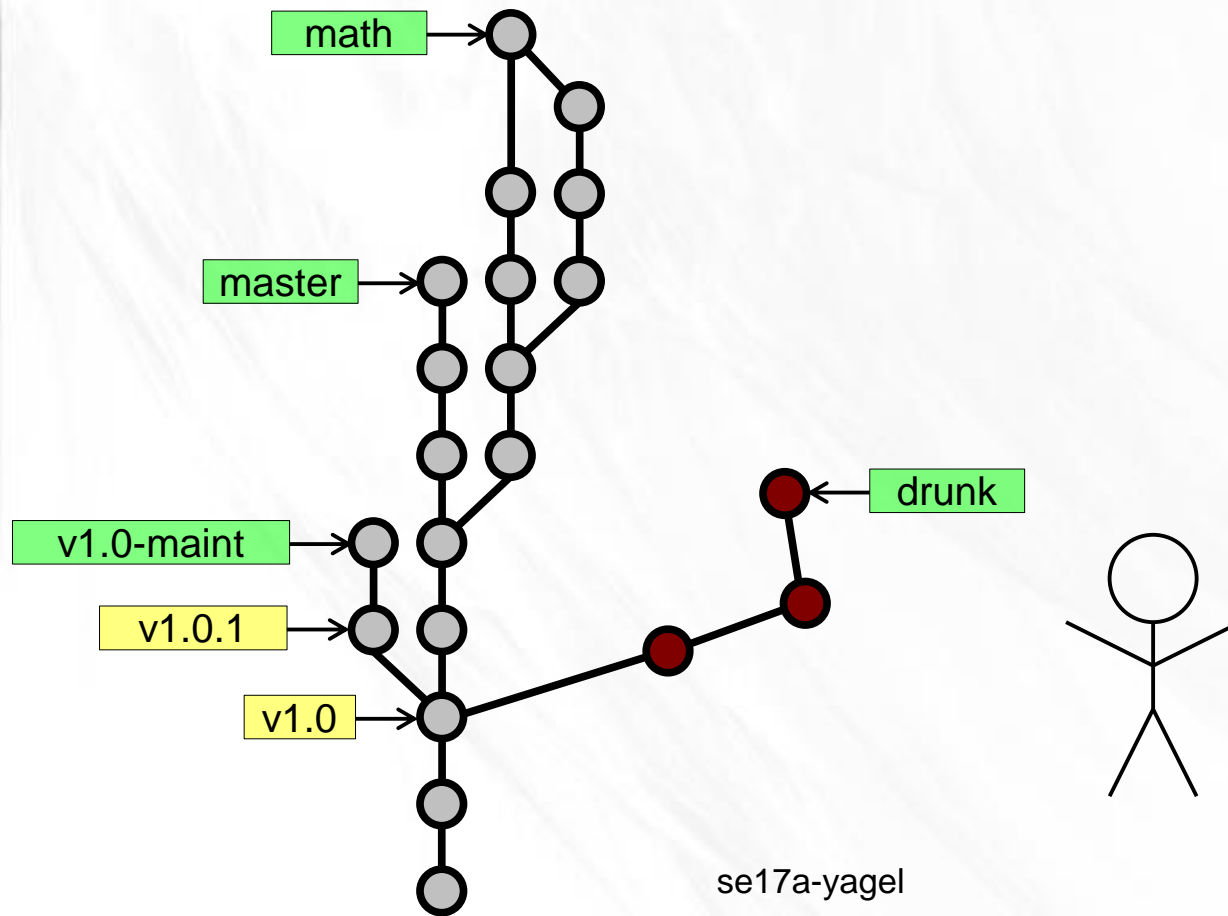
Merges



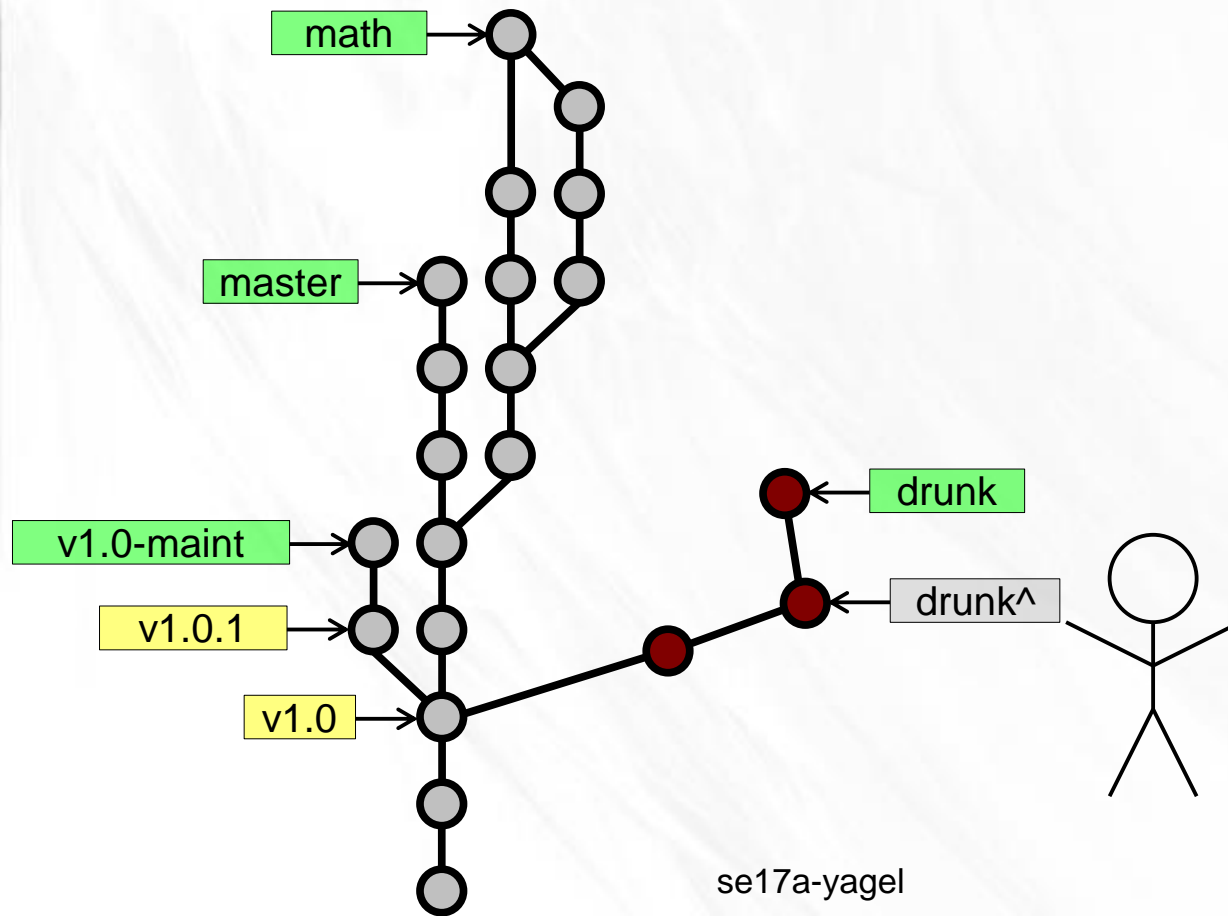
Rewriting History



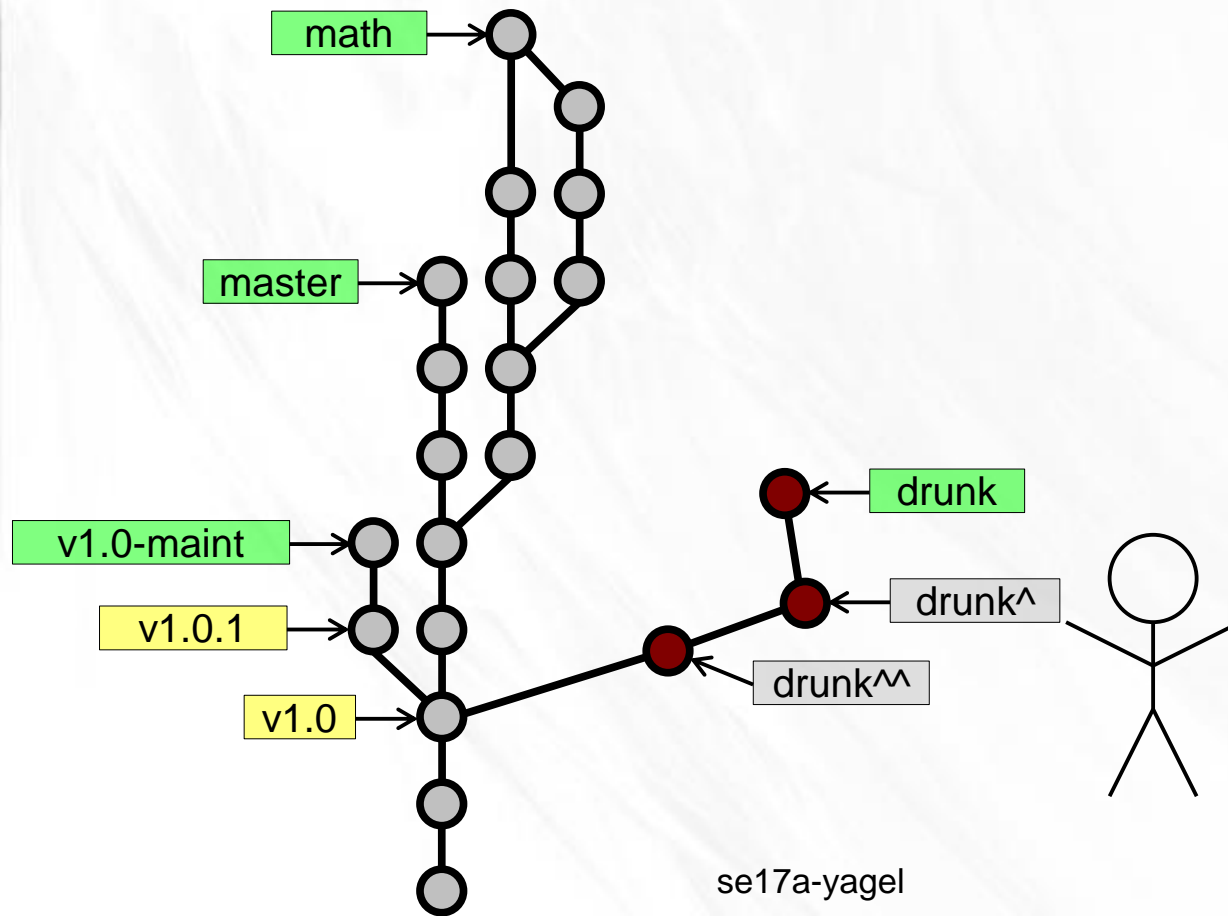
Rewriting History



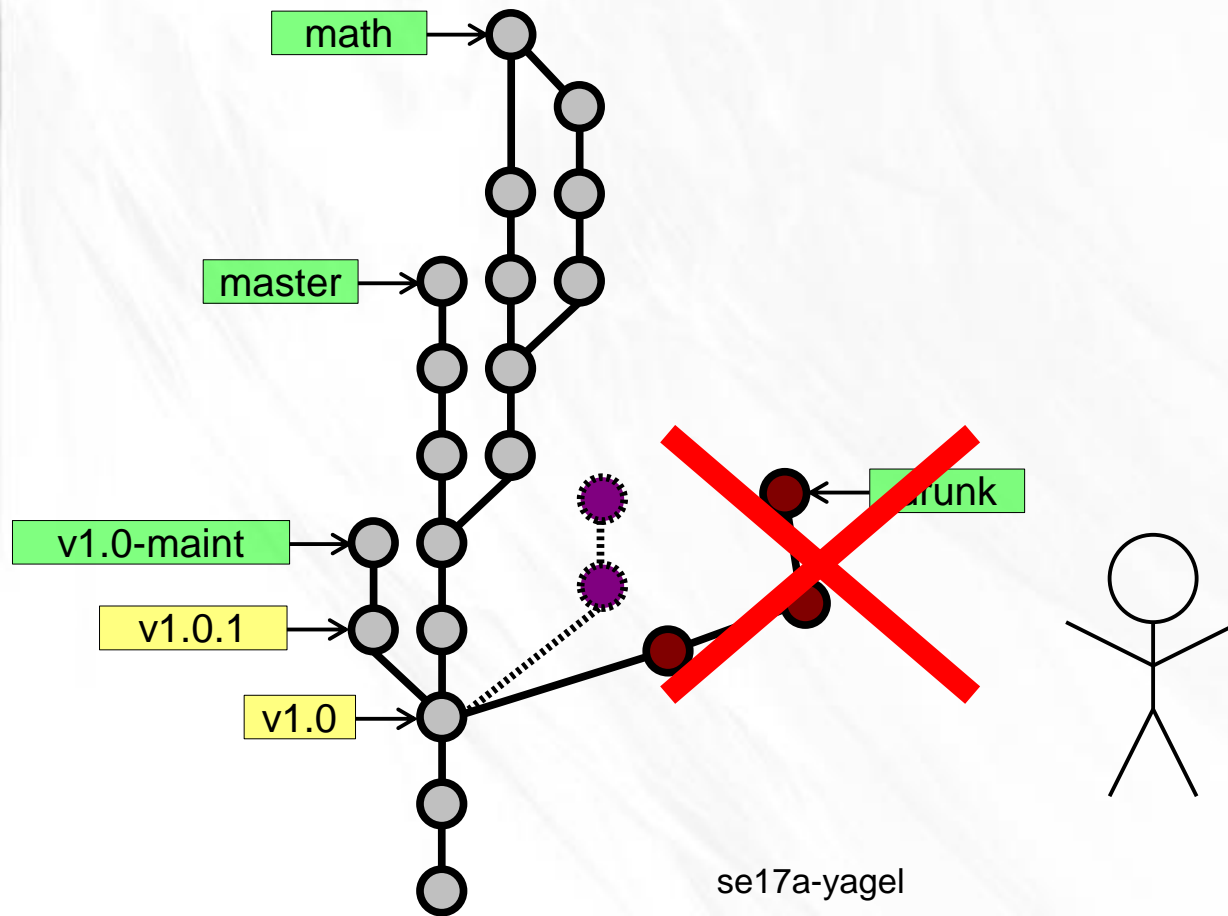
Rewriting History



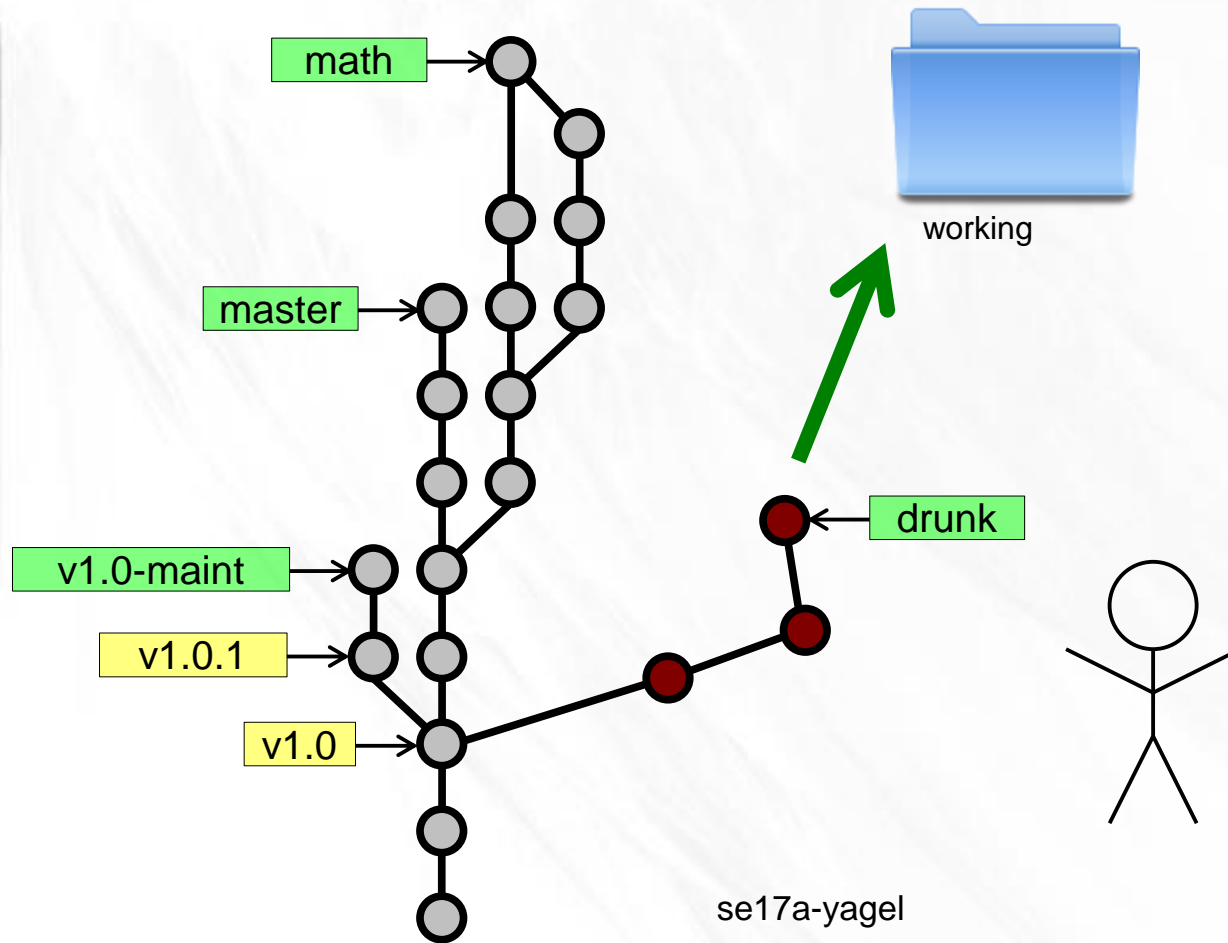
Rewriting History



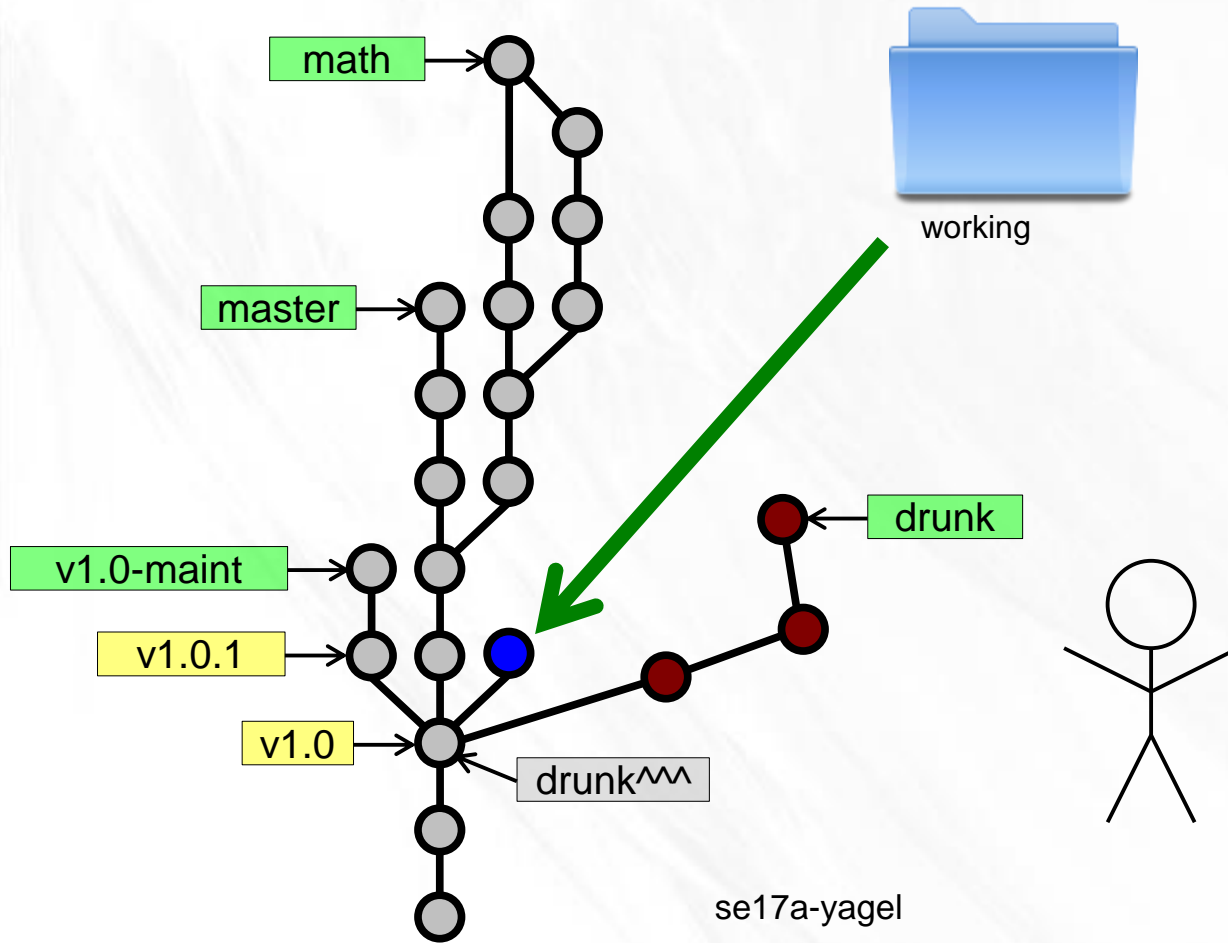
Rewriting History



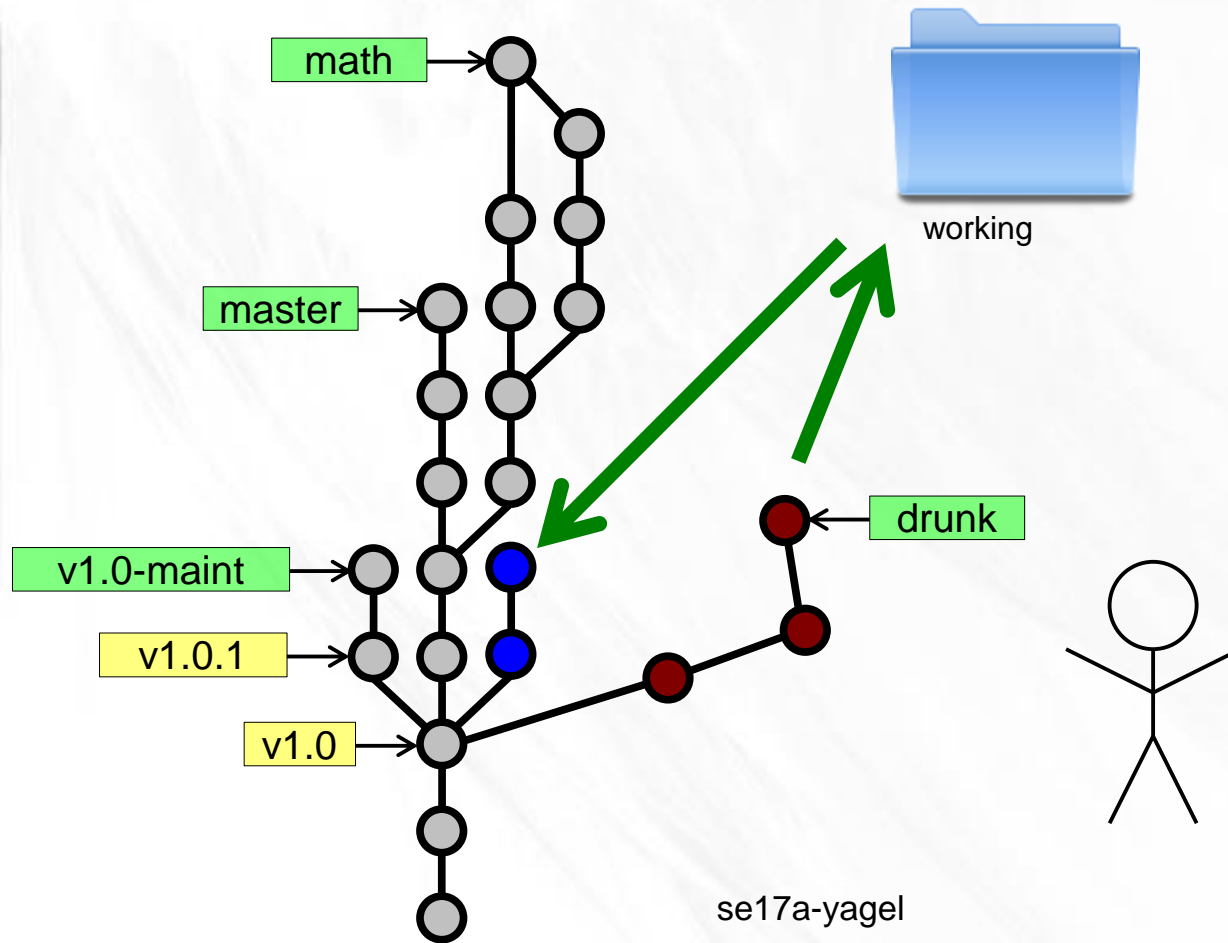
Rewriting History



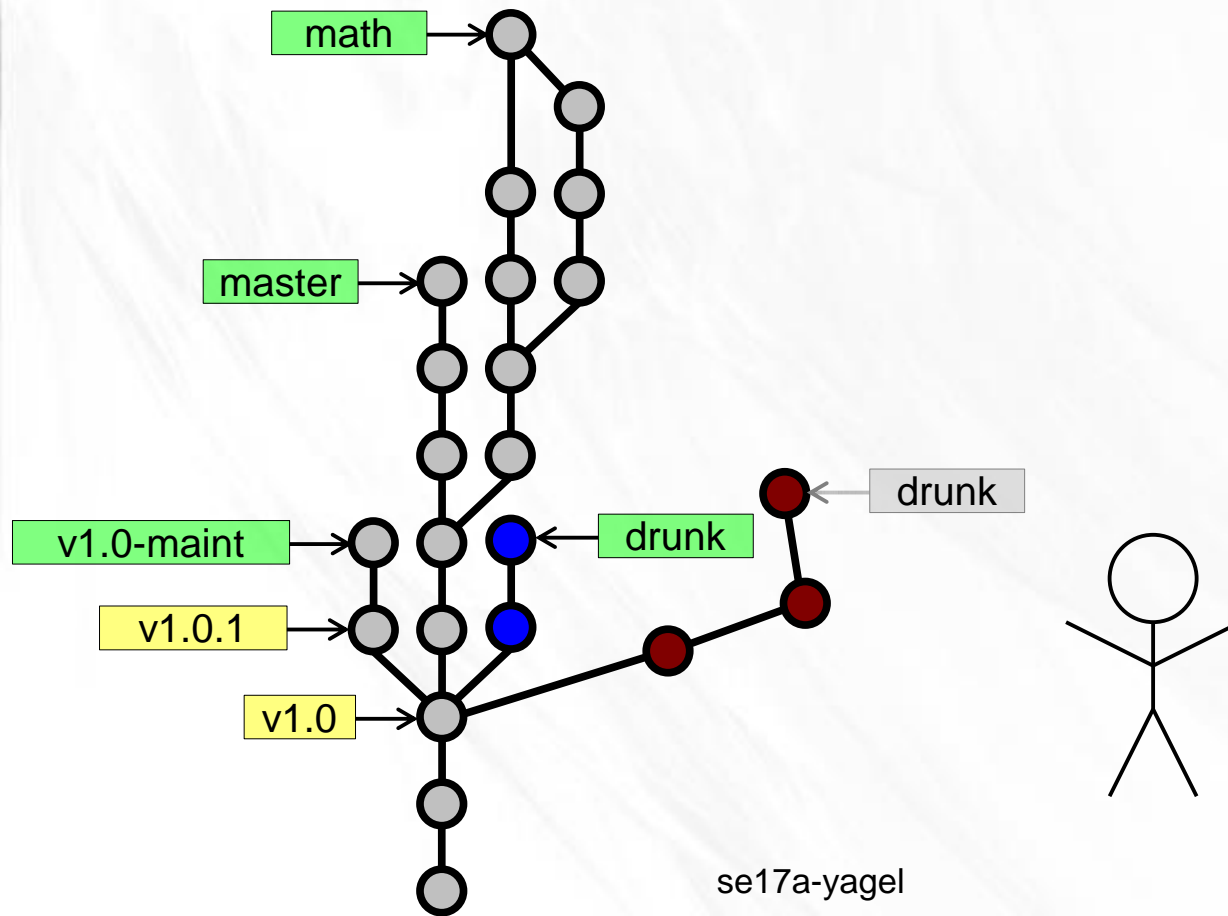
Rewriting History



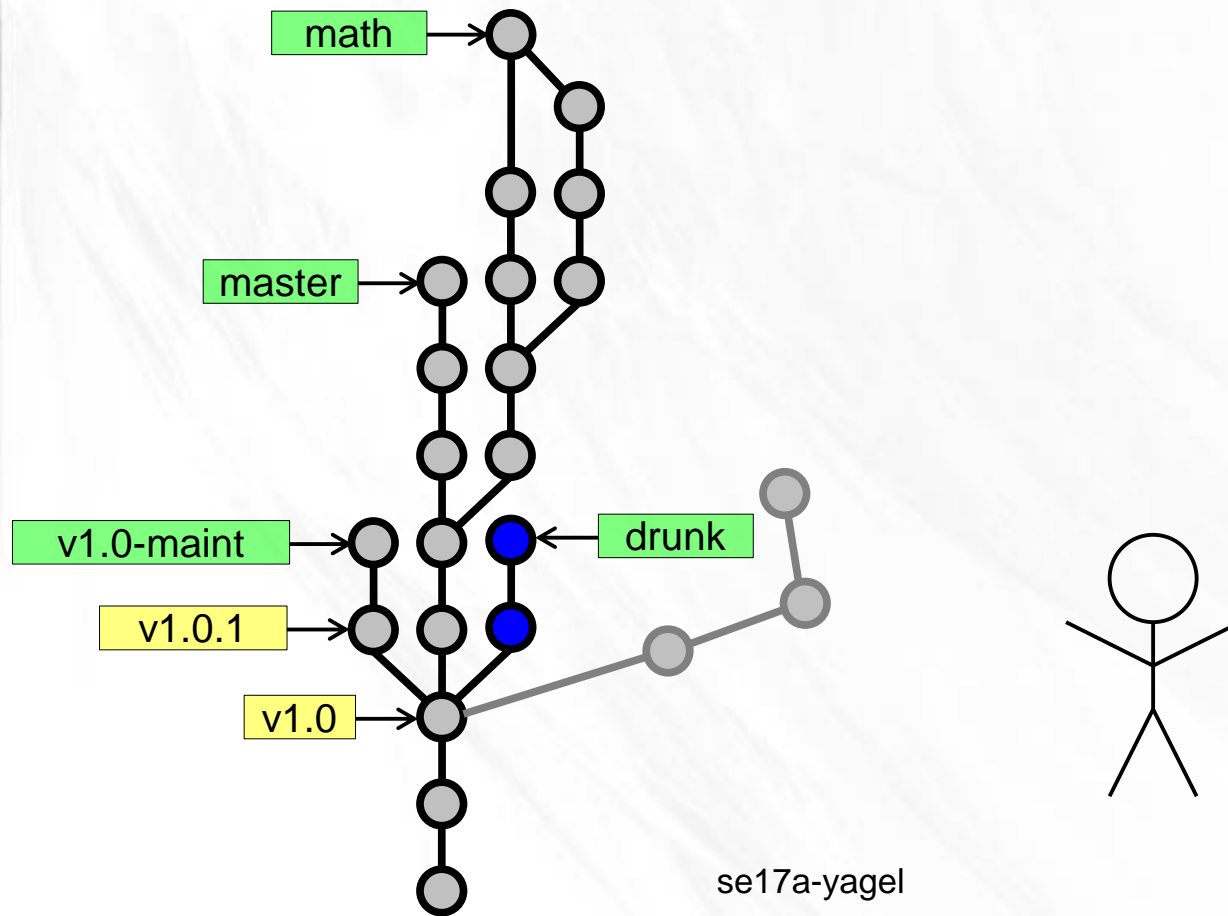
Rewriting History



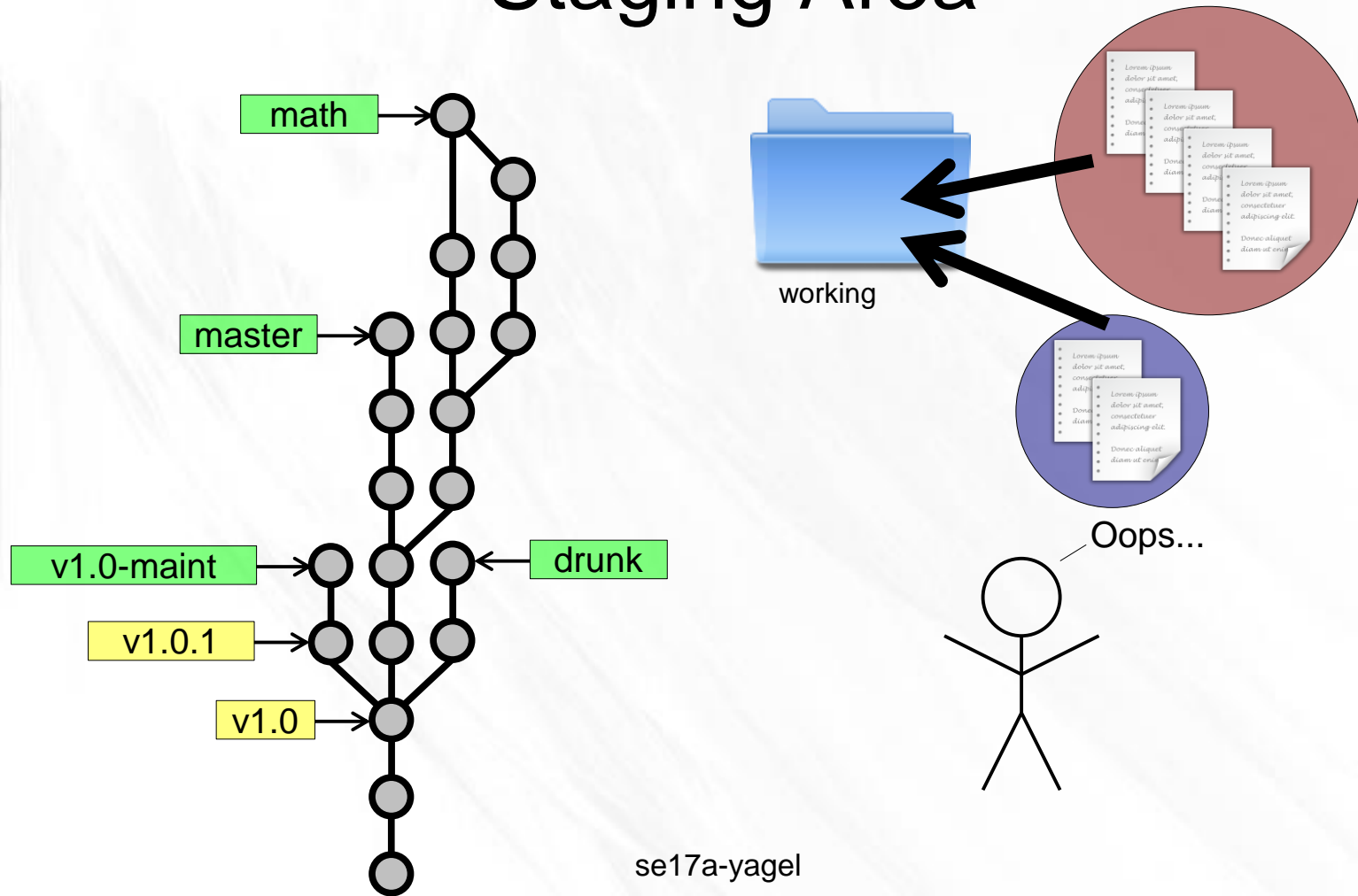
Rewriting History



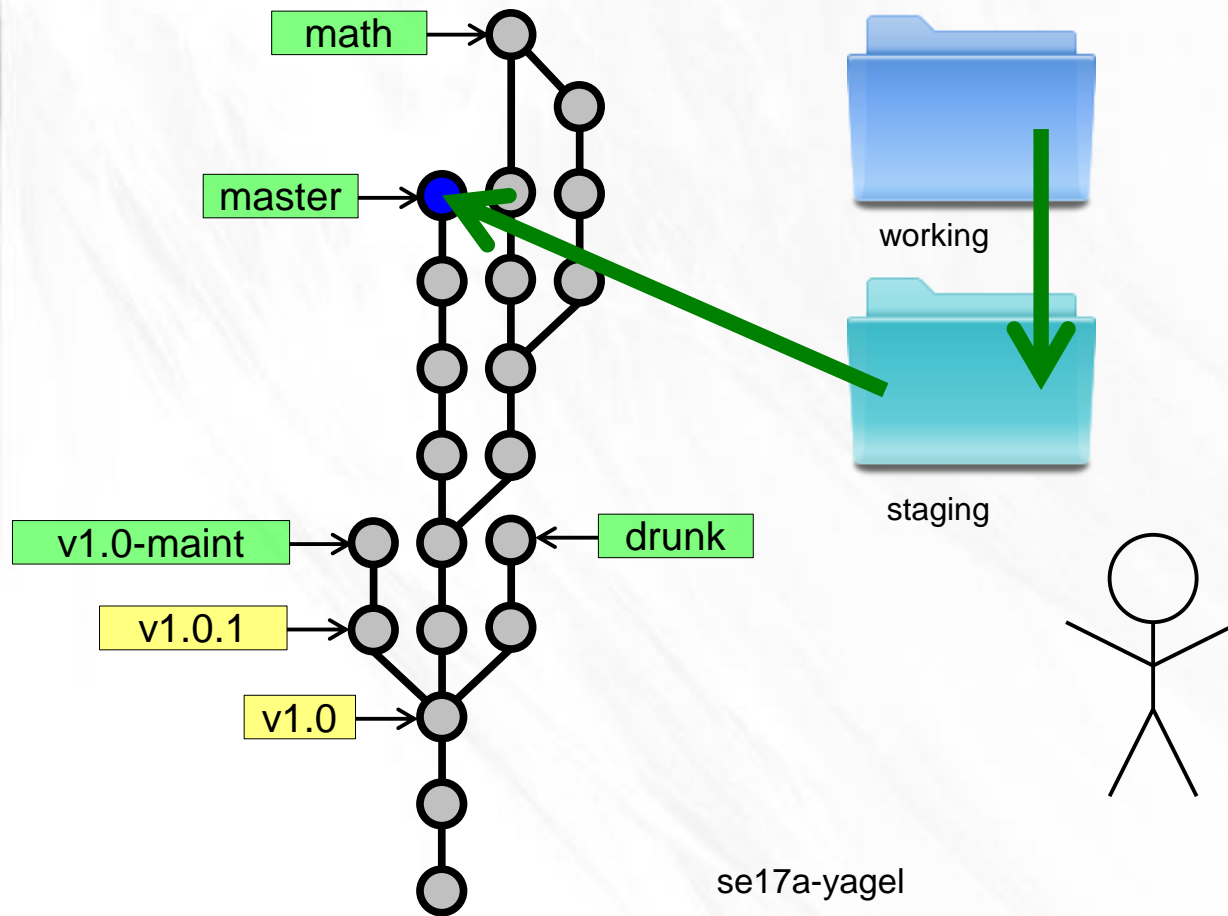
Rewriting History



Staging Area

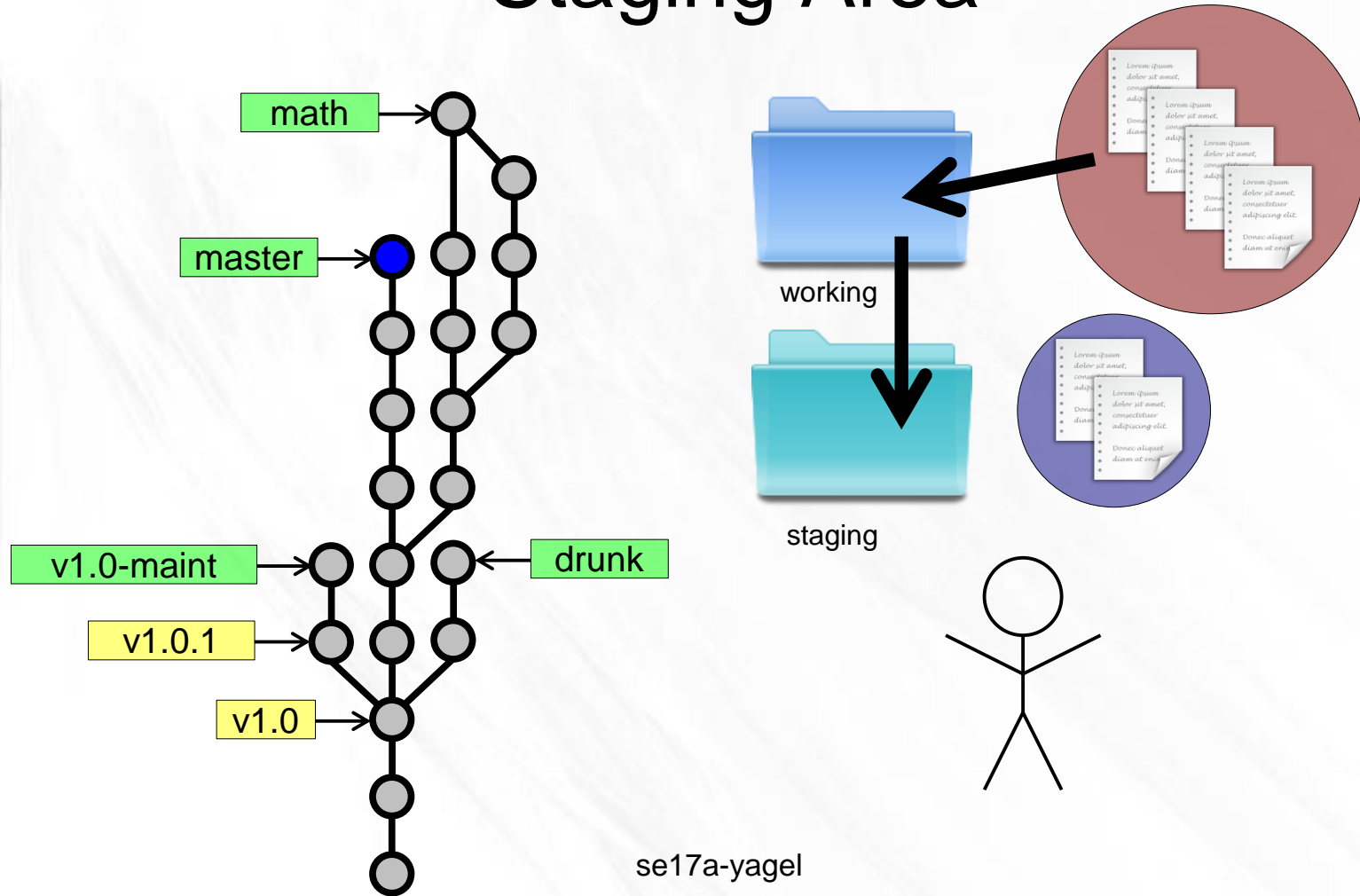


Staging Area

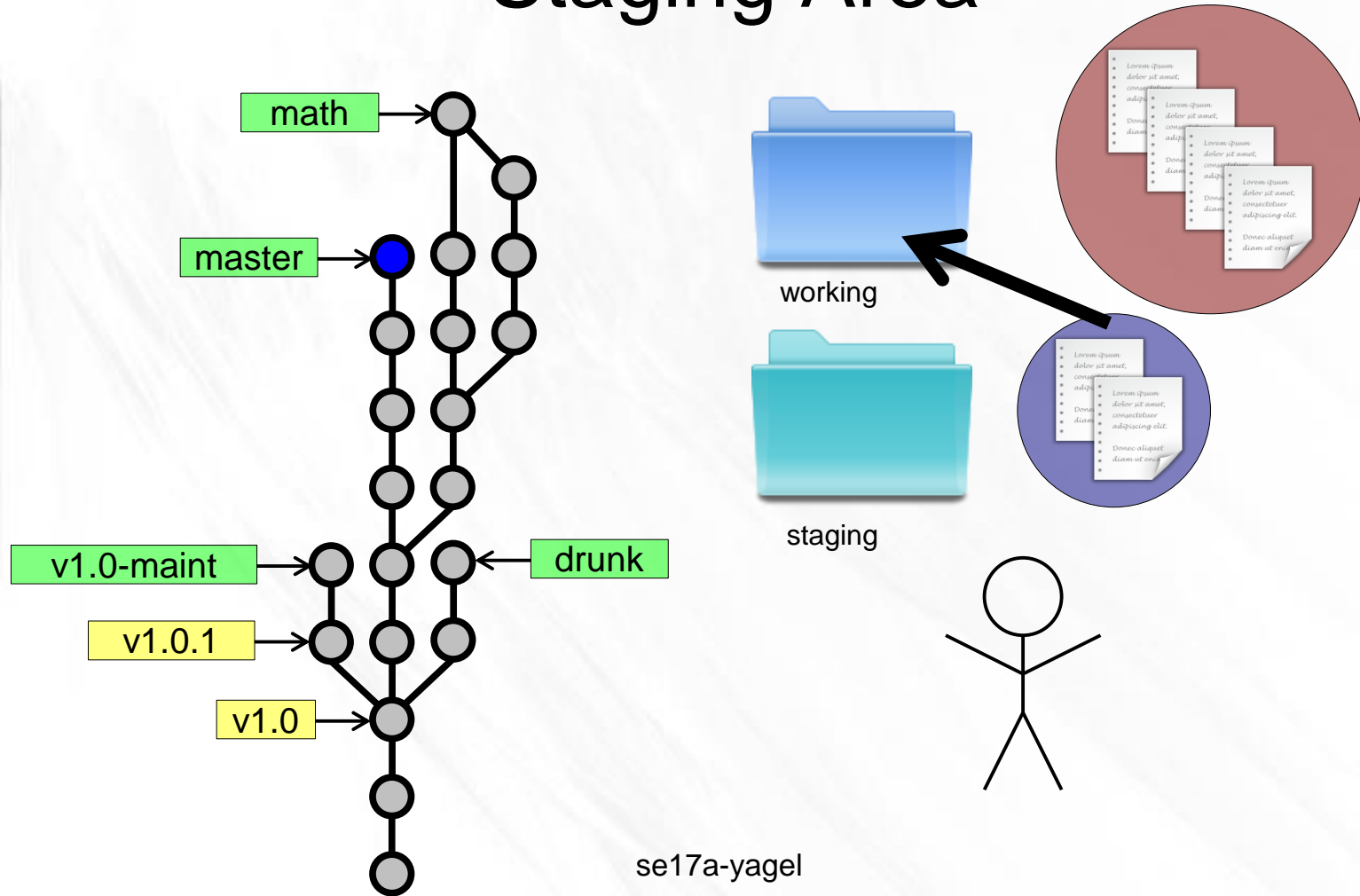


se17a-yagel

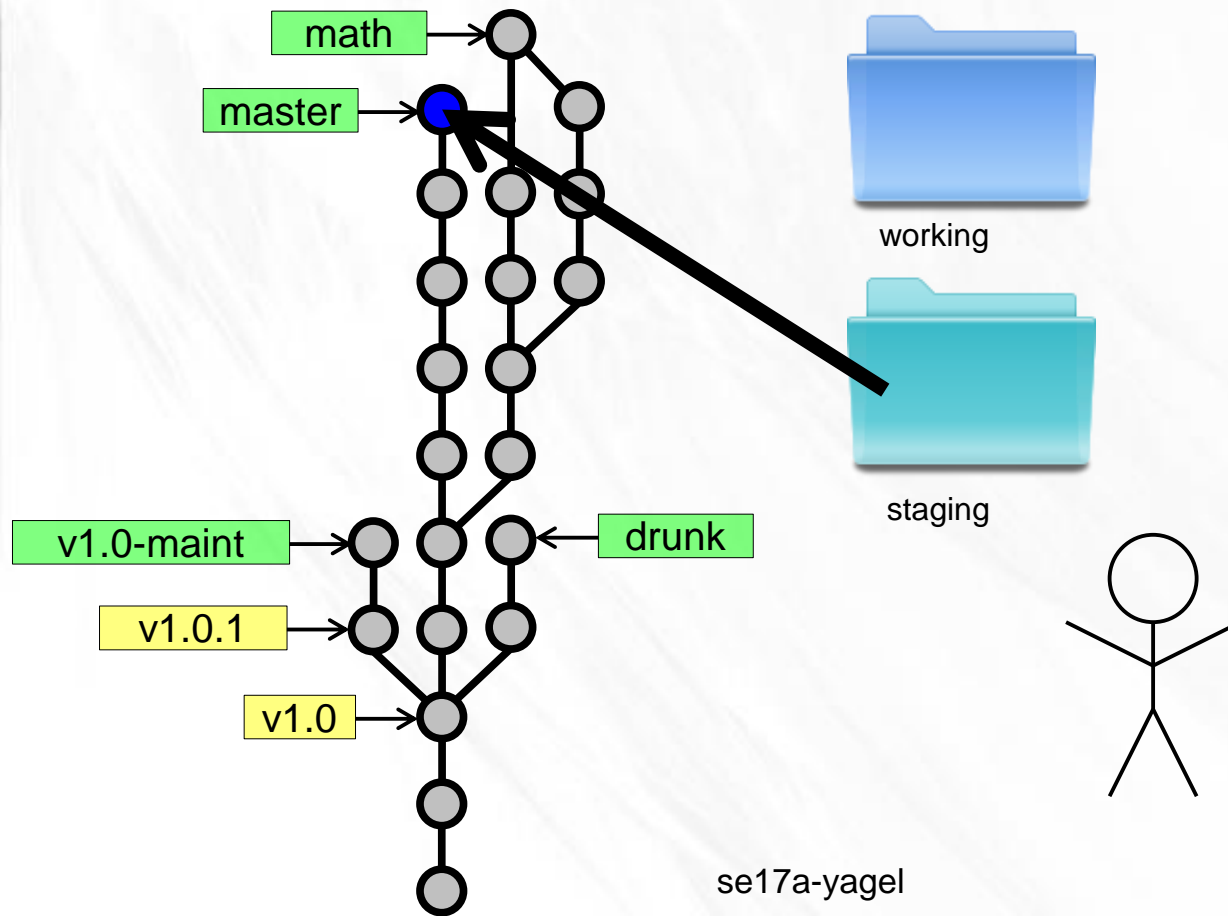
Staging Area



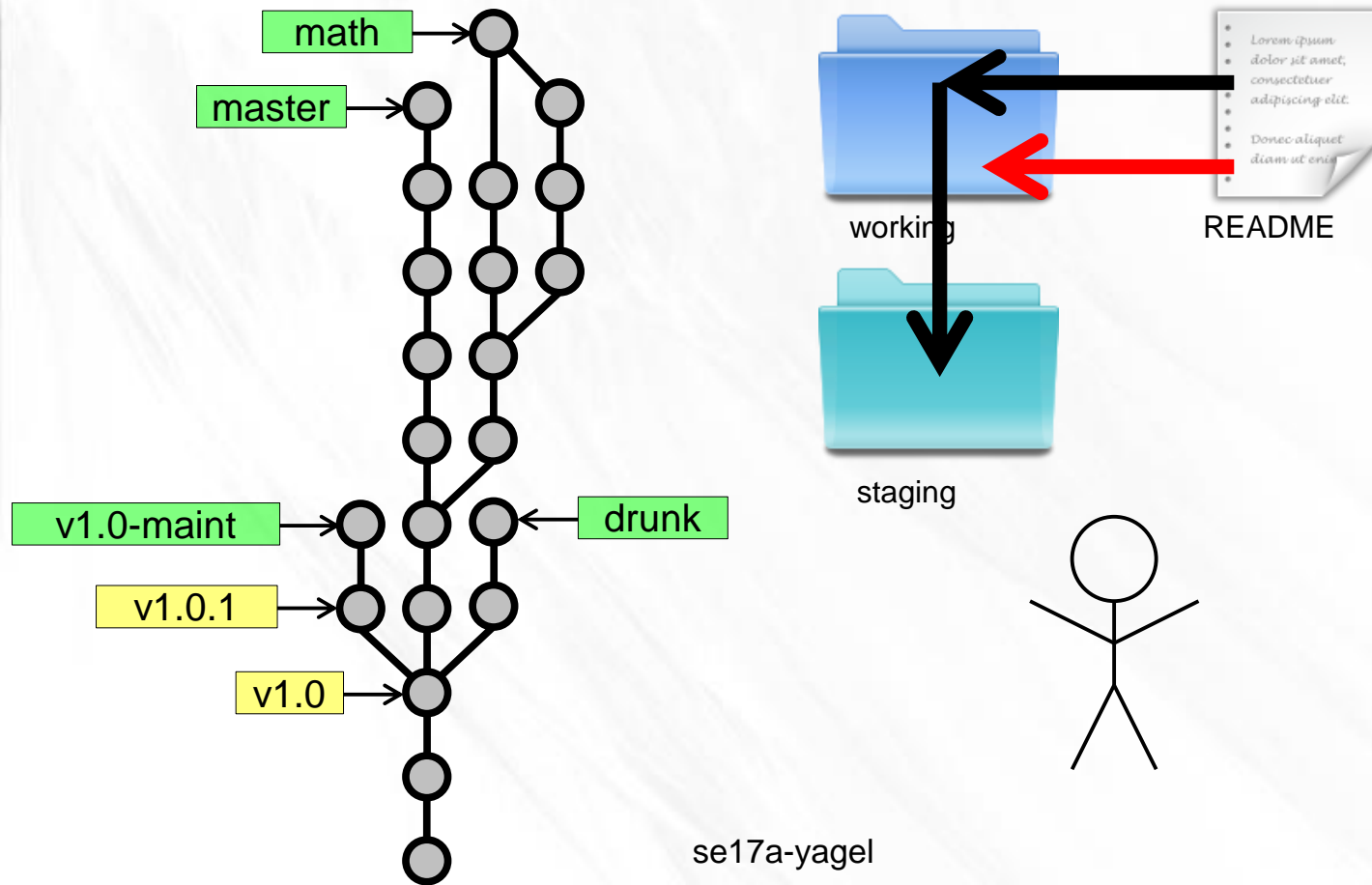
Staging Area



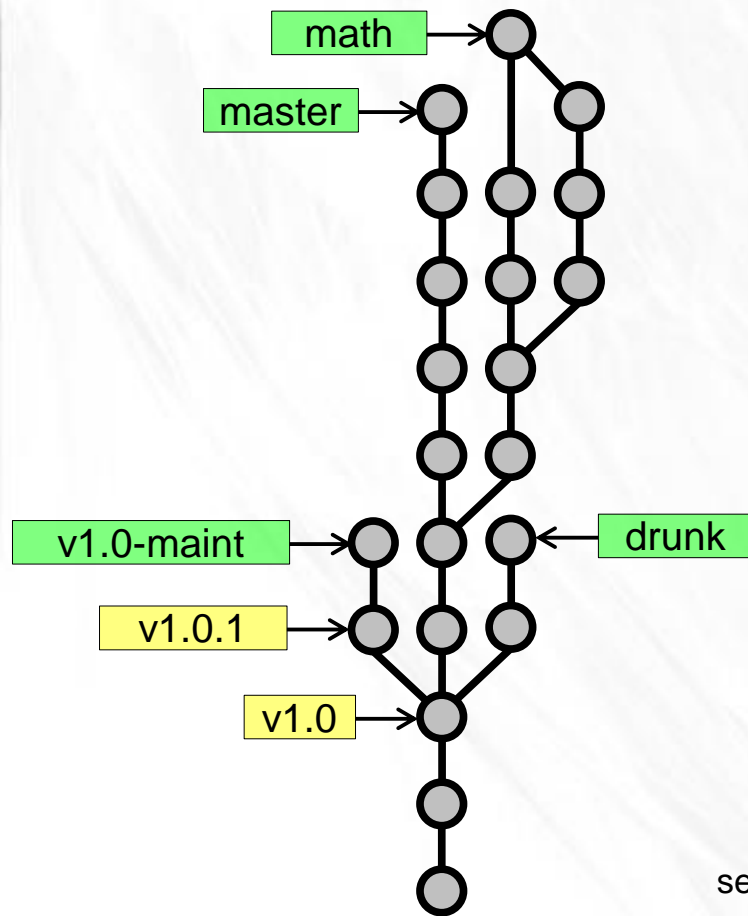
Staging Area



Staging Area



Diffs



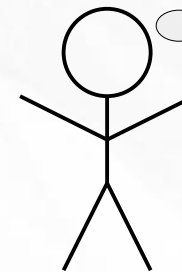
working



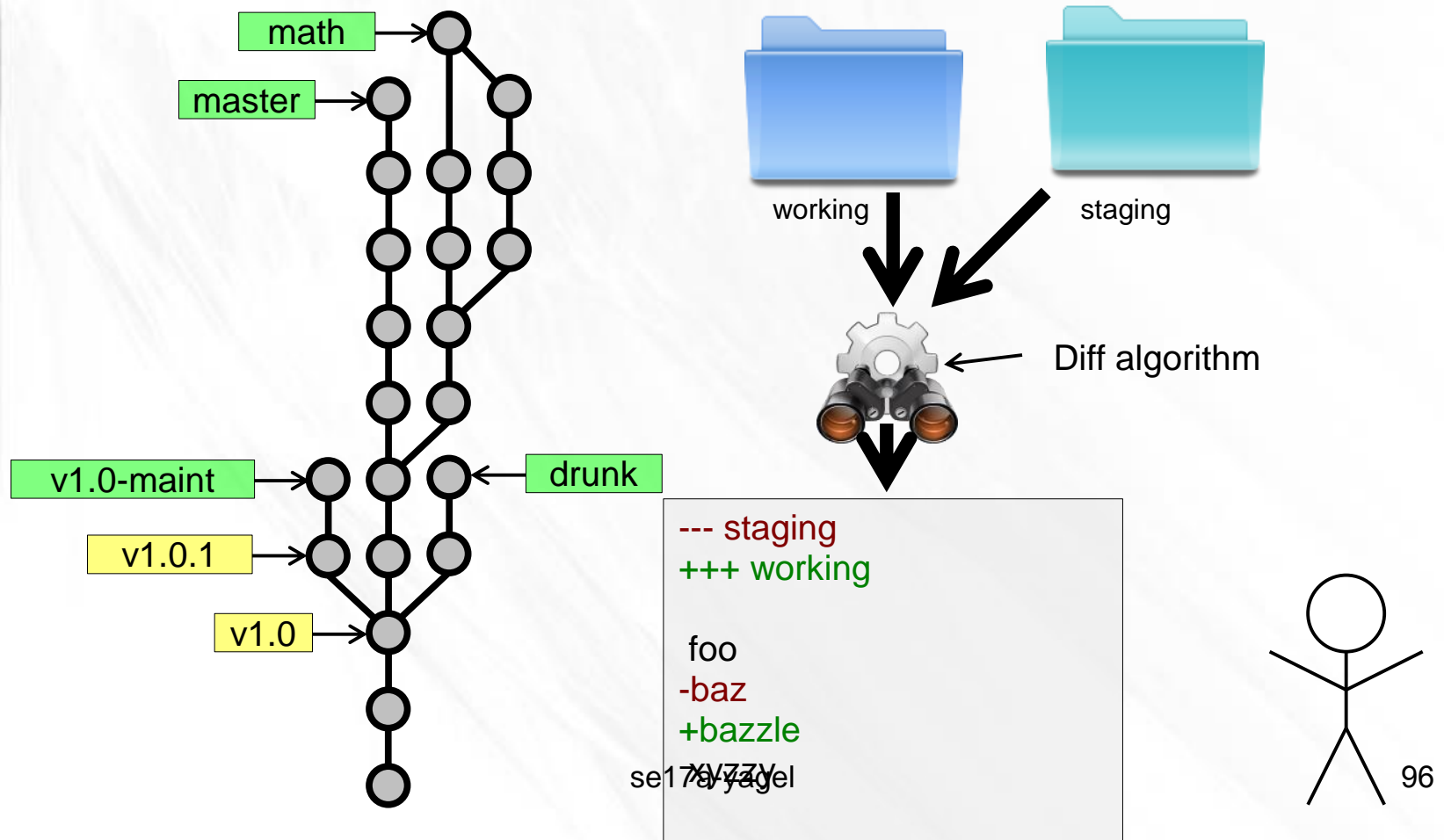
staging

What are the changes?

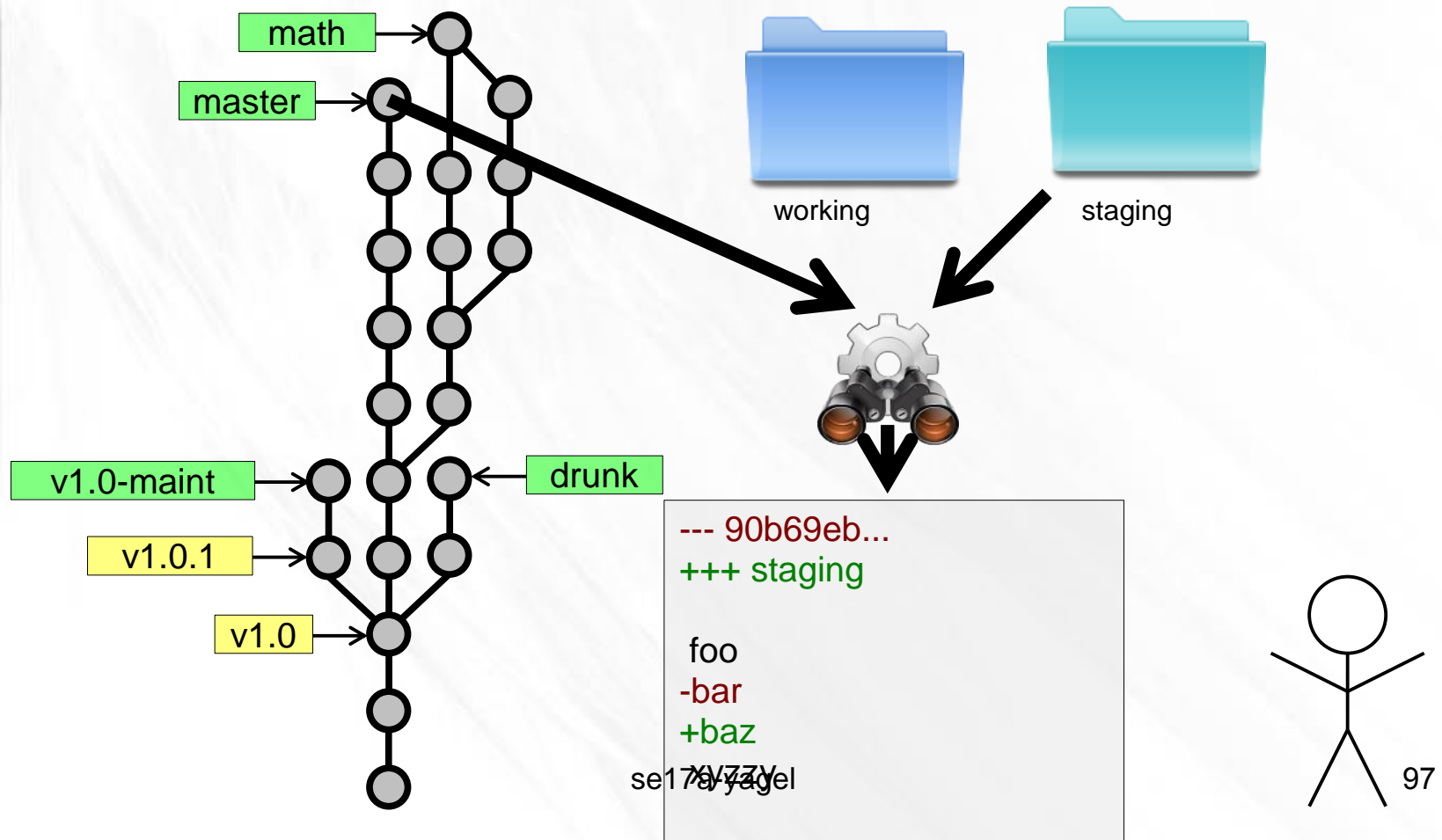
- working vs. staging
- working vs. snapshot X
- staging vs. snapshot X
- snapshot X vs. snapshot Y



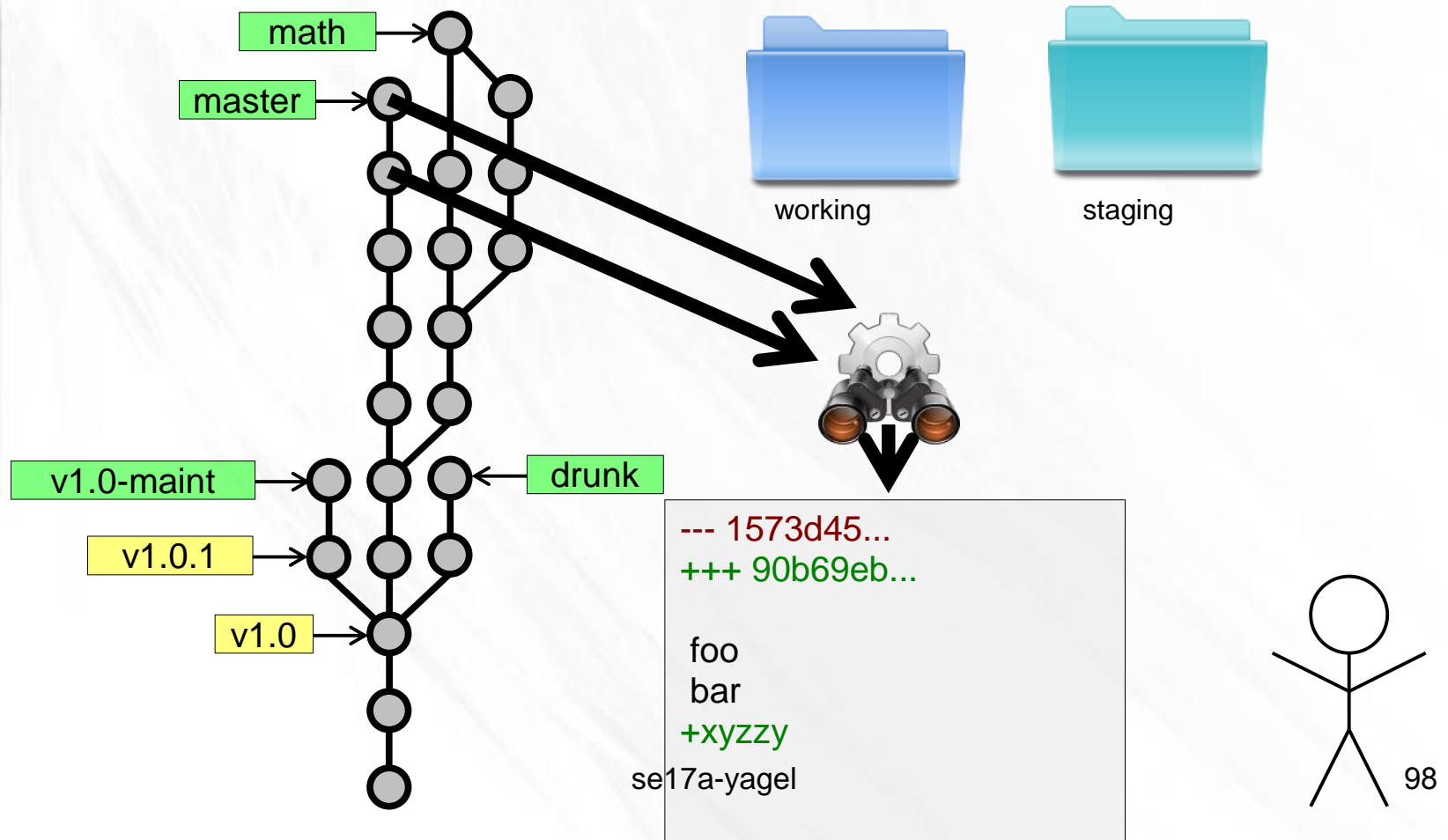
Diffs



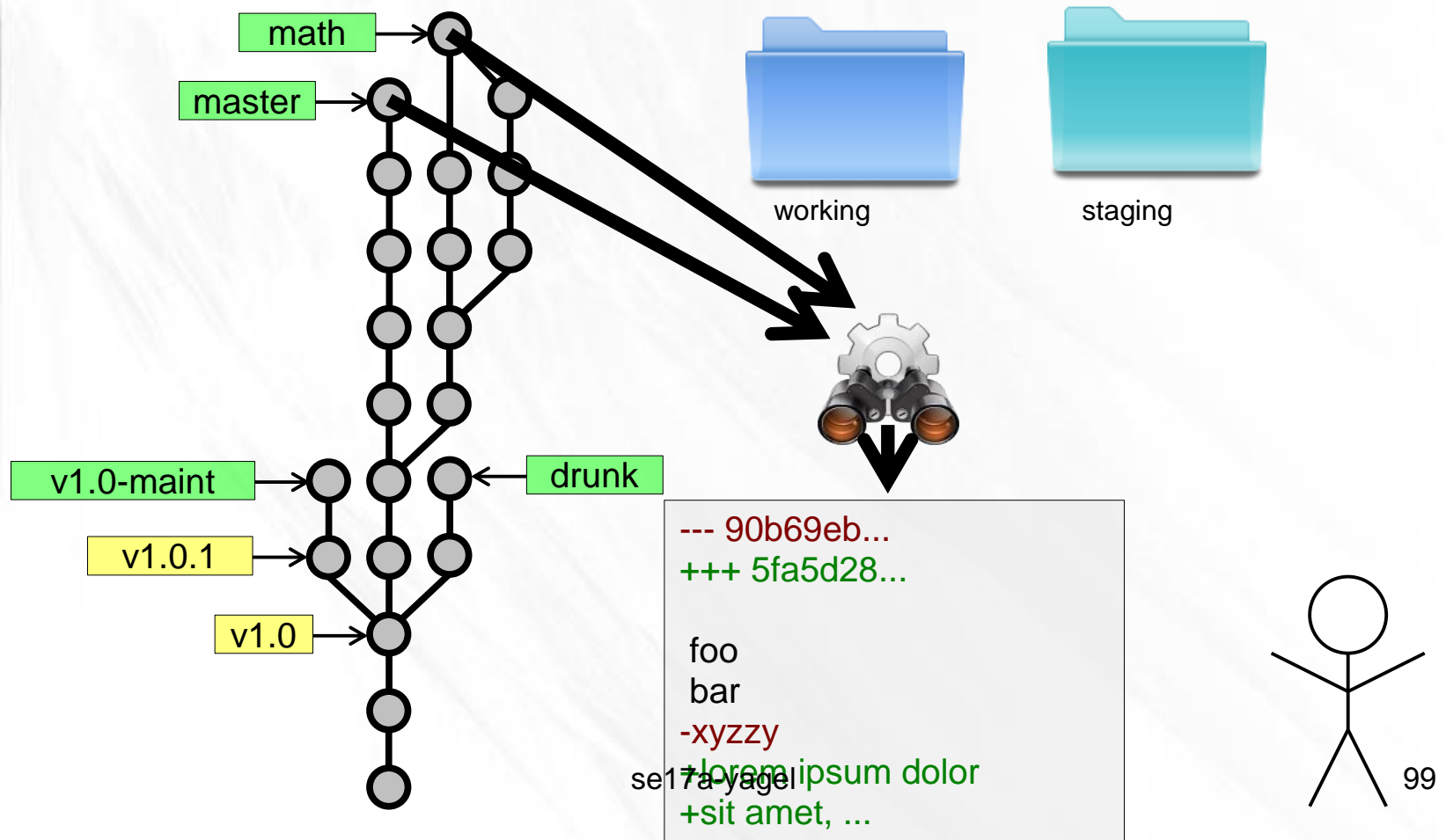
Diffs



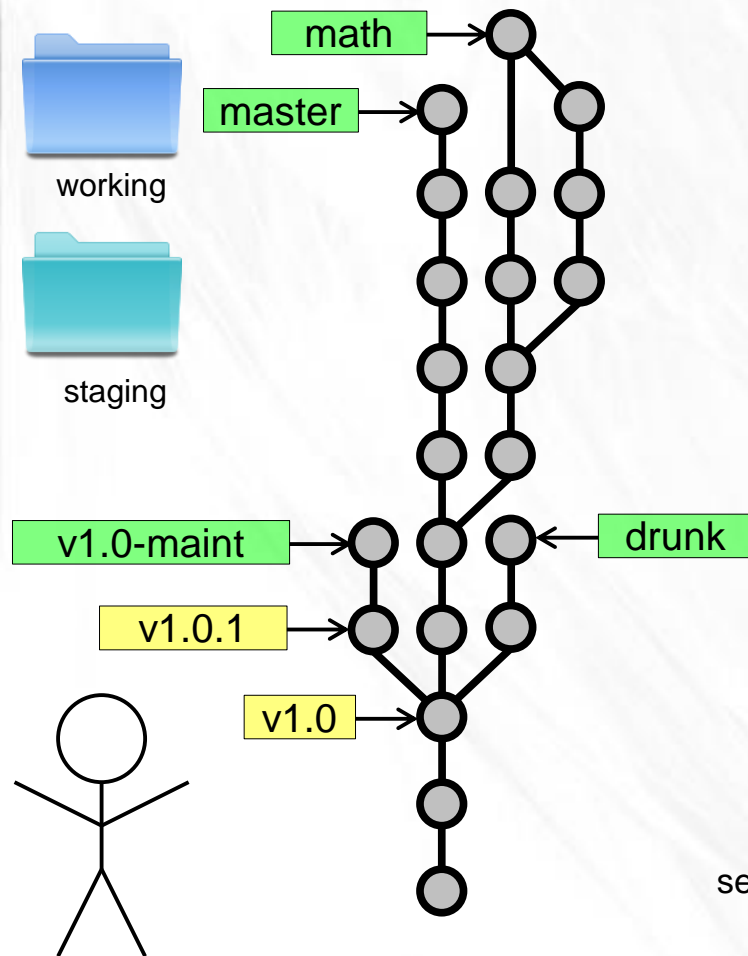
Diffs



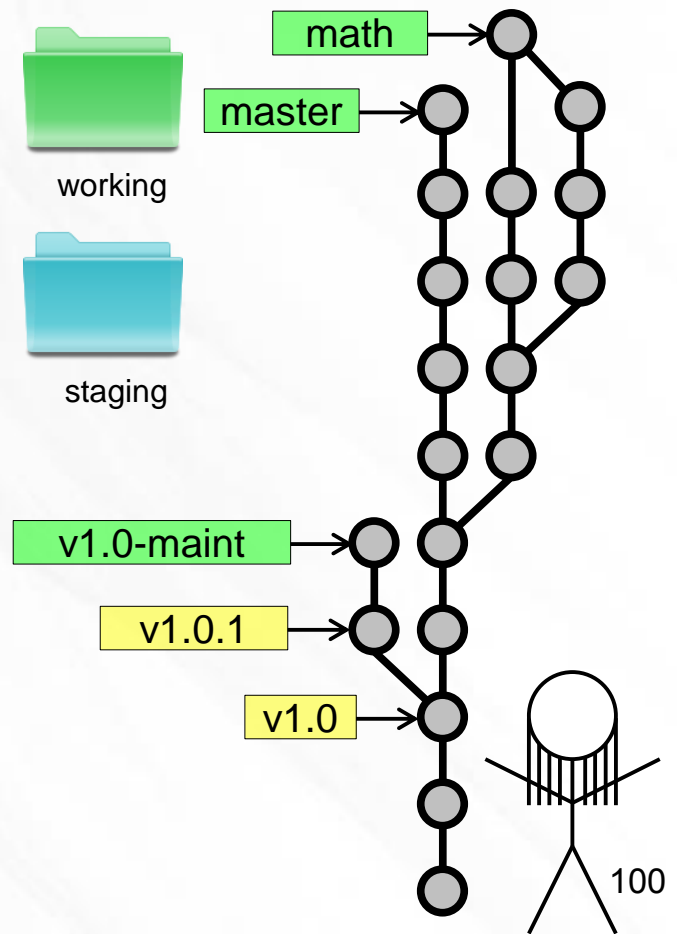
Diffs



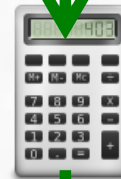
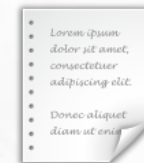
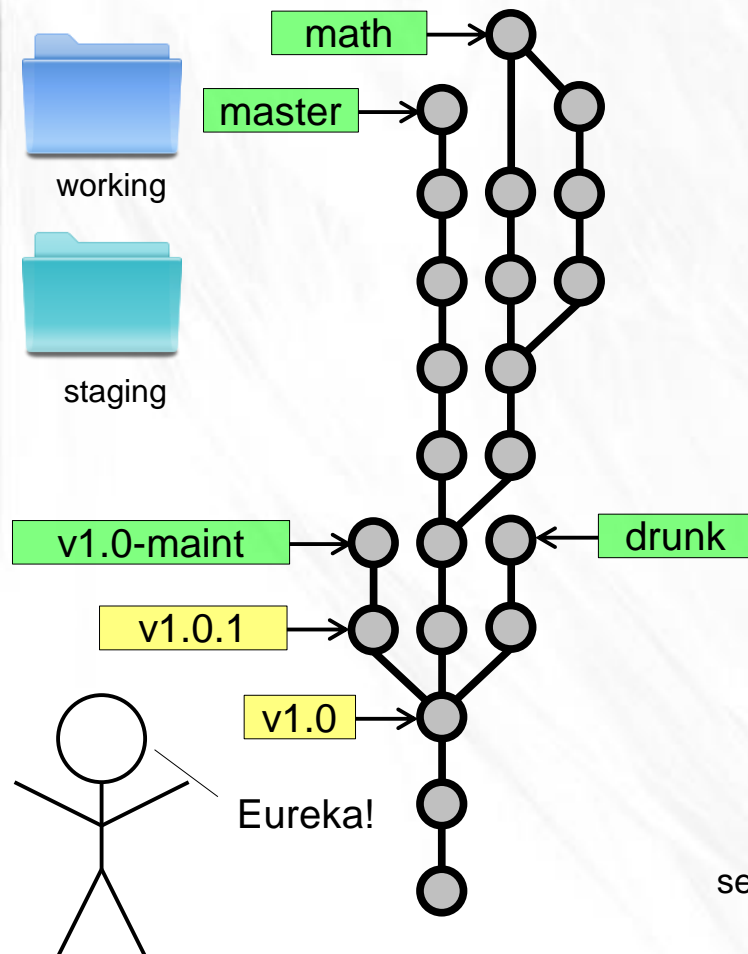
Eliminating Duplication



se17a-yagel



Eliminating Duplication



SHA1

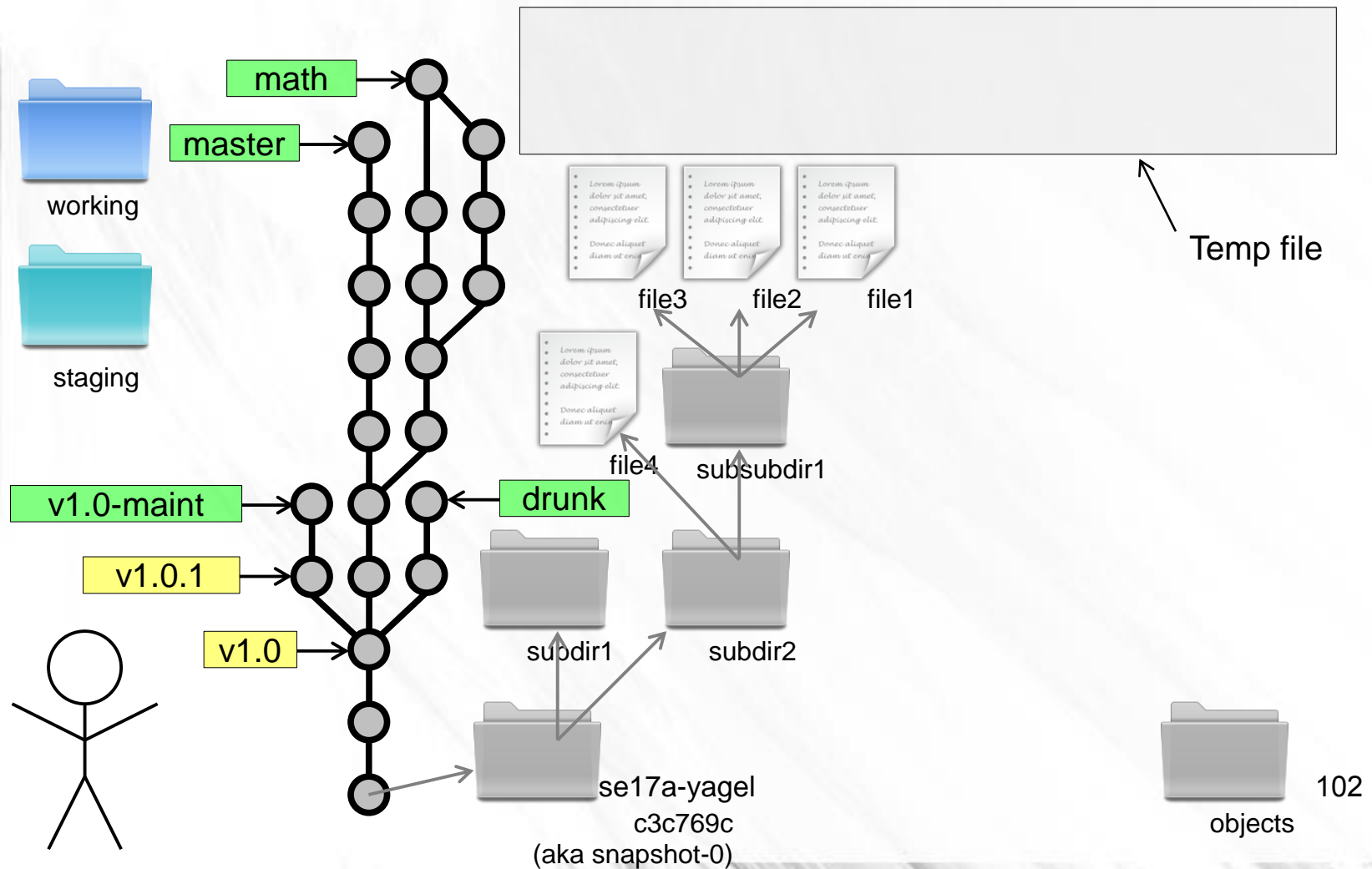
2804133755c3ed396d162028c7b30a1cbcfecded



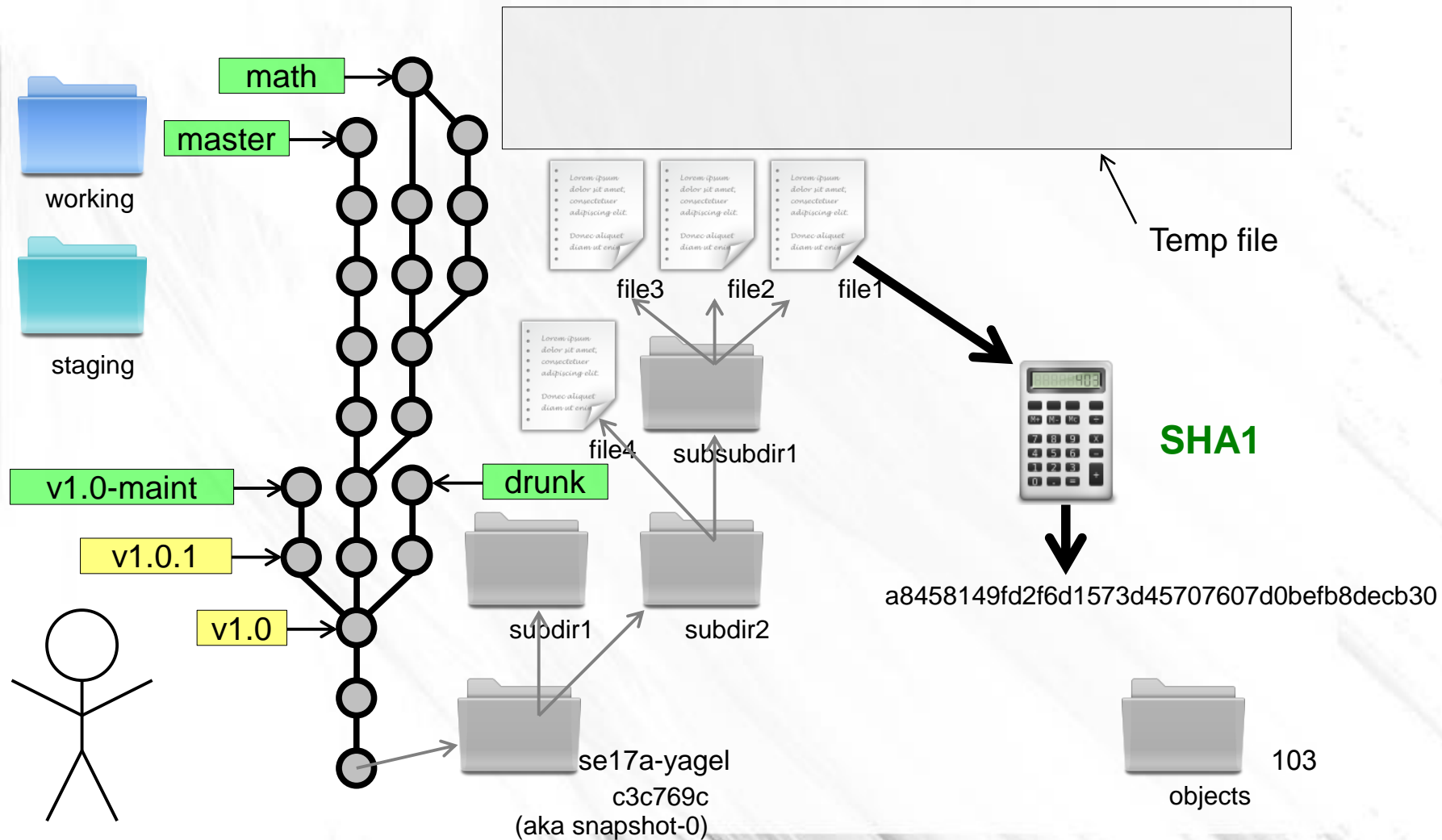
101

objects

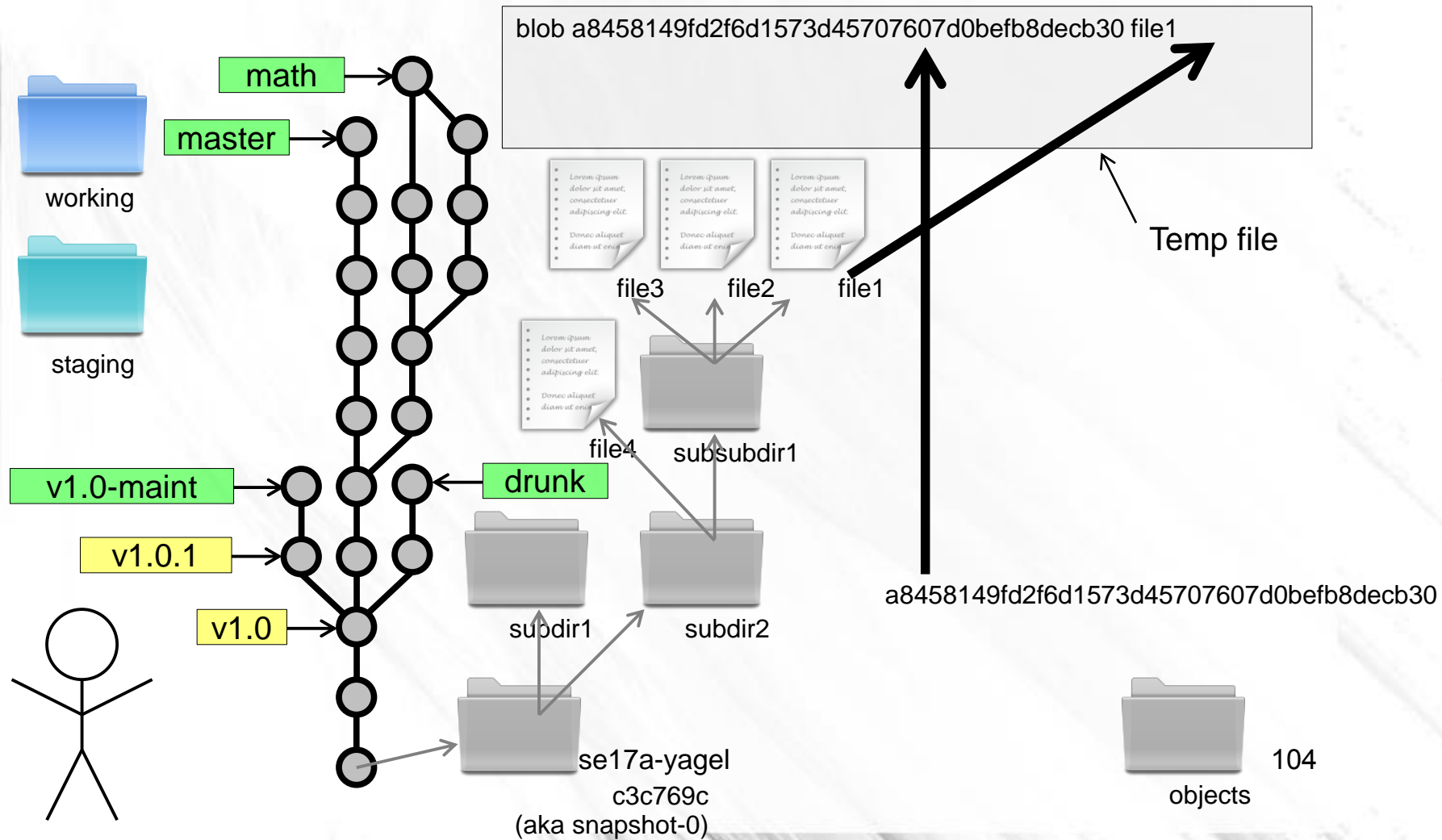
Eliminating Duplication



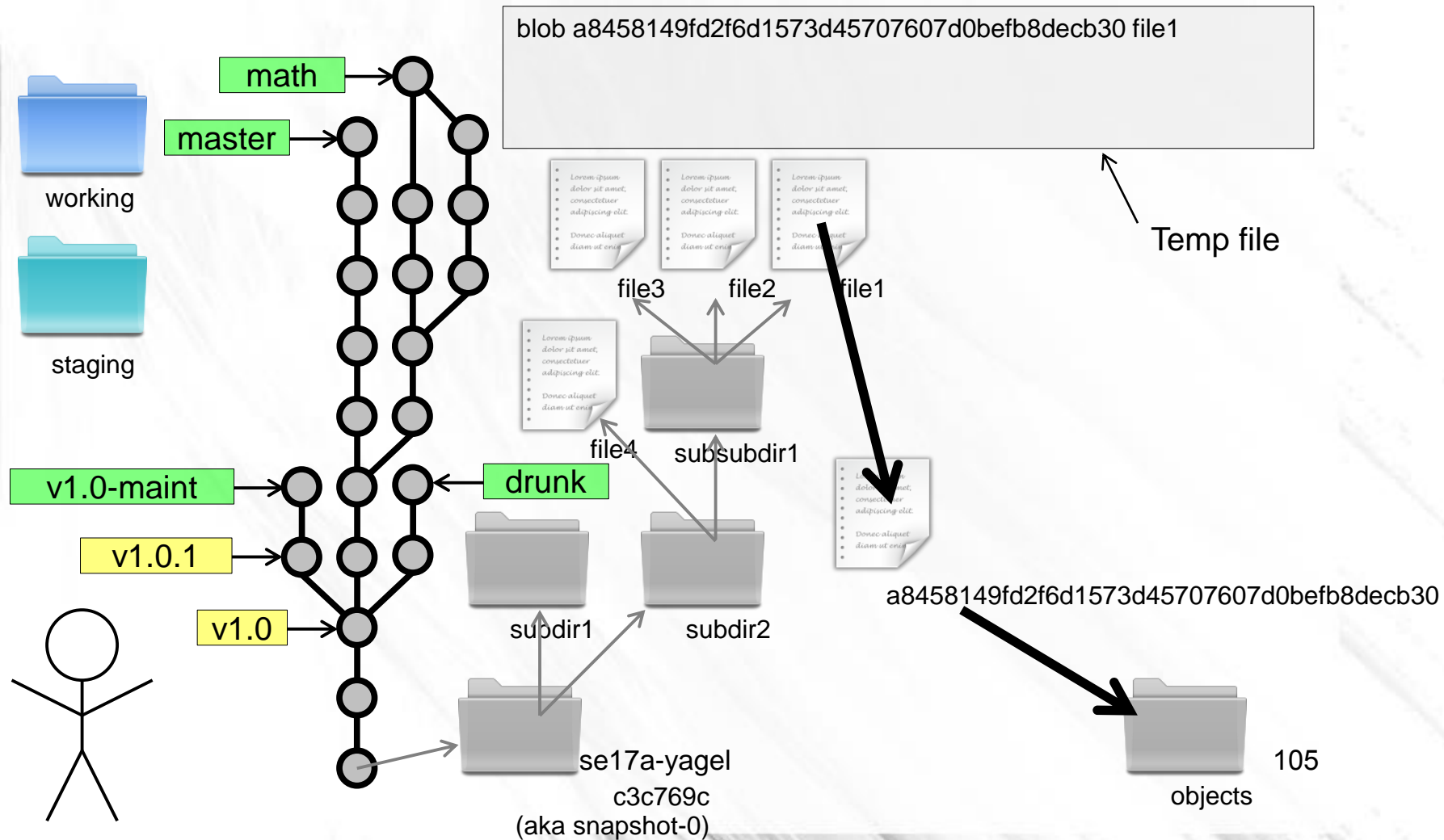
Eliminating Duplication



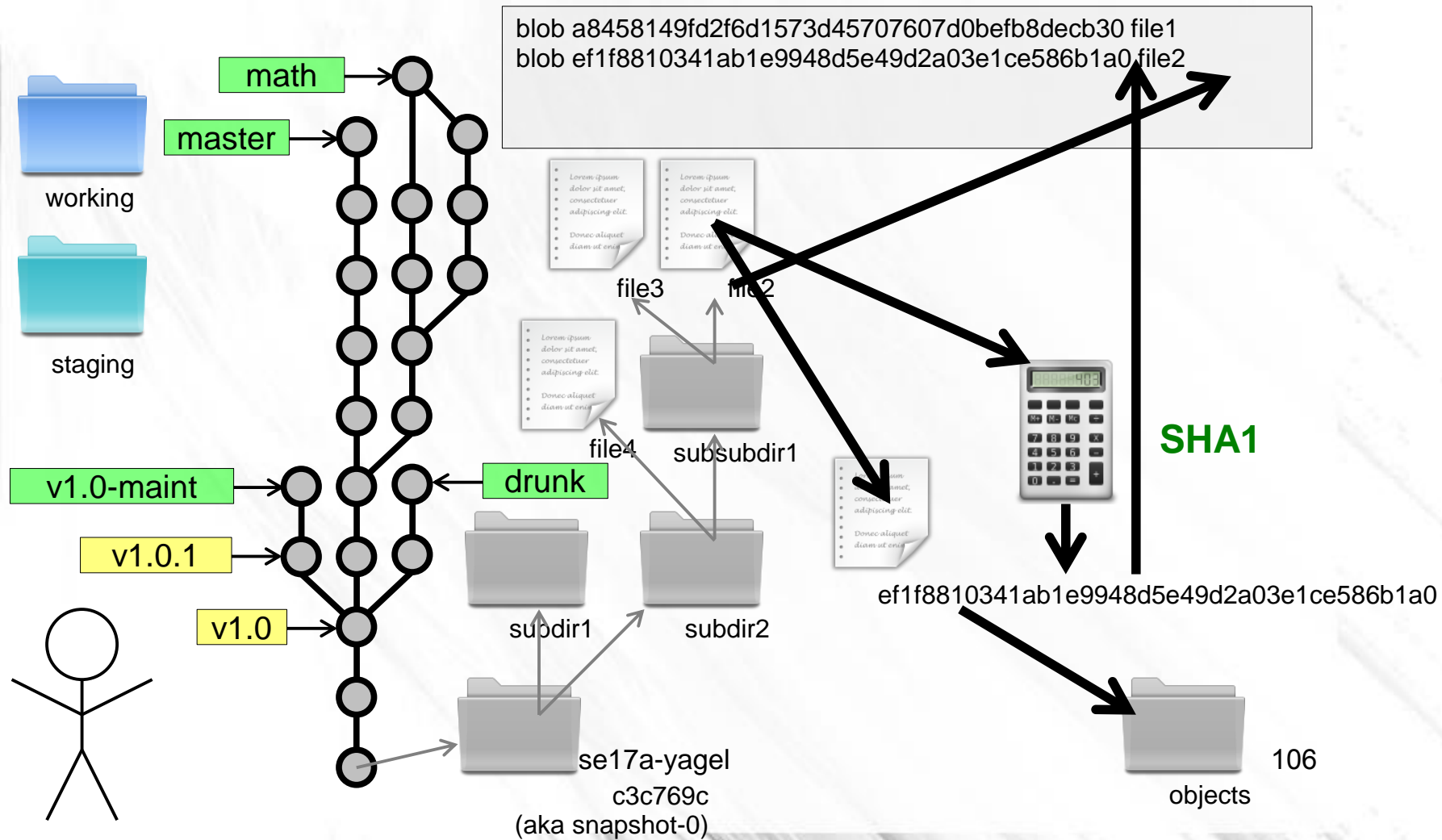
Eliminating Duplication



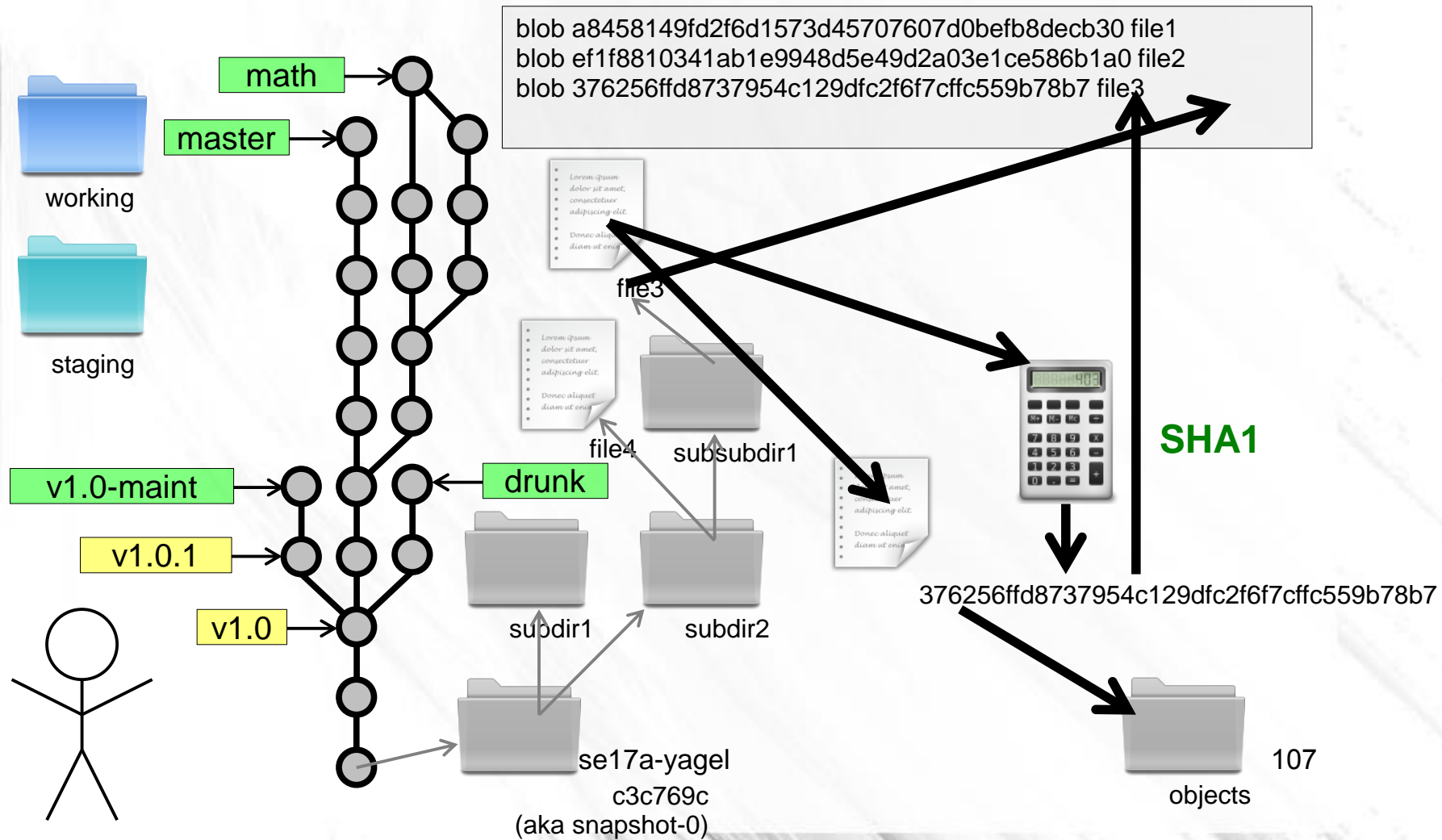
Eliminating Duplication



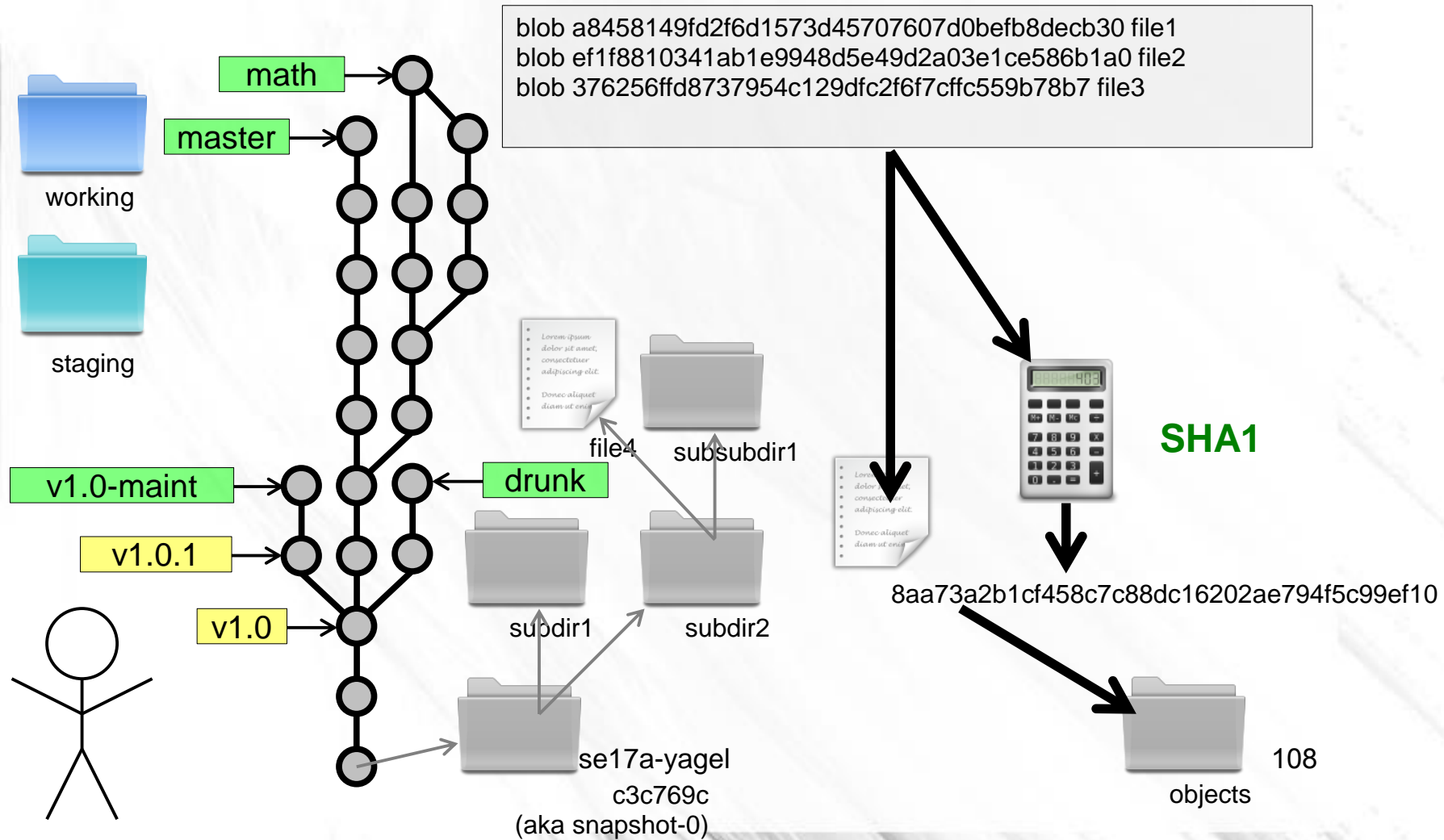
Eliminating Duplication



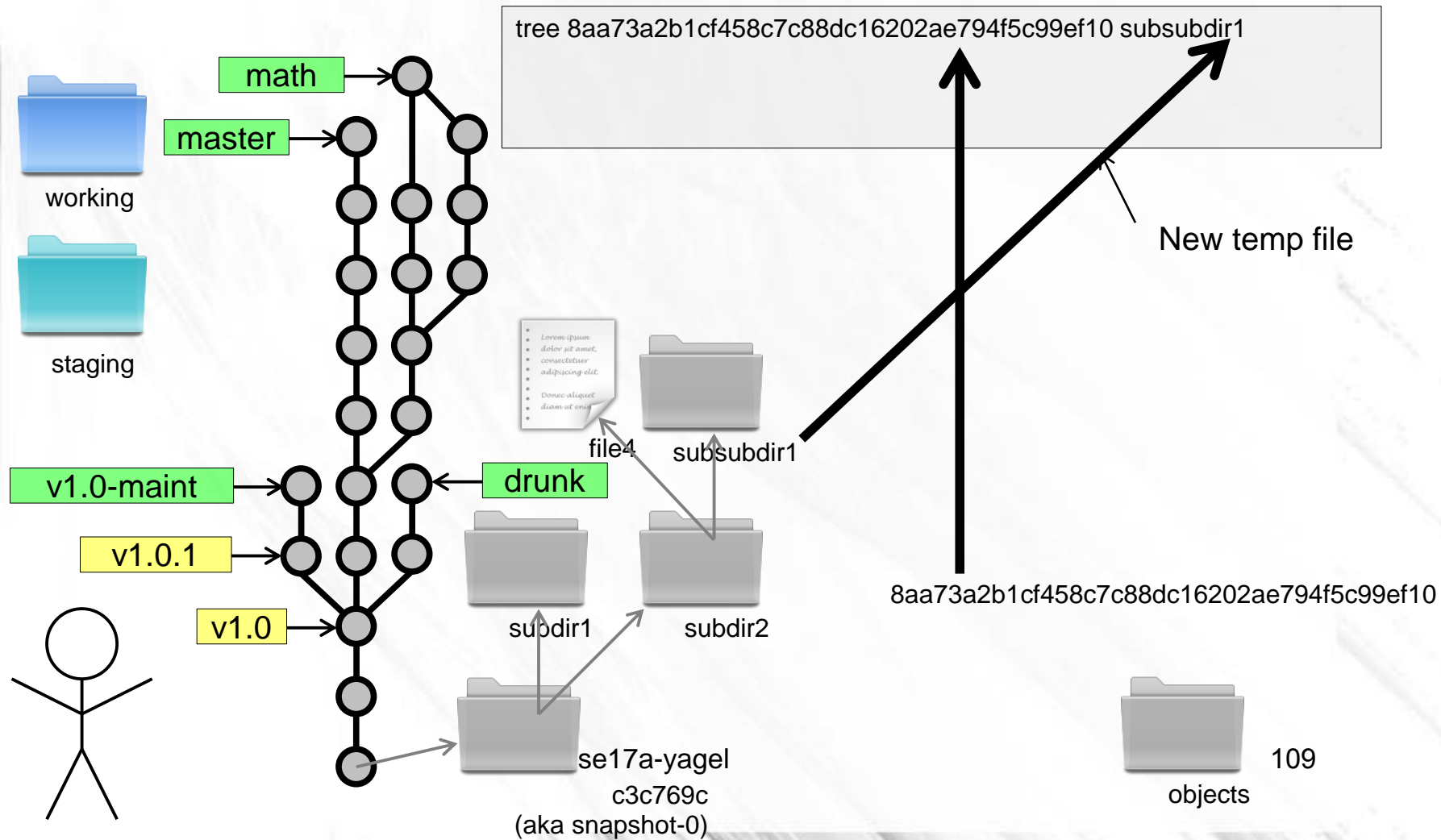
Eliminating Duplication



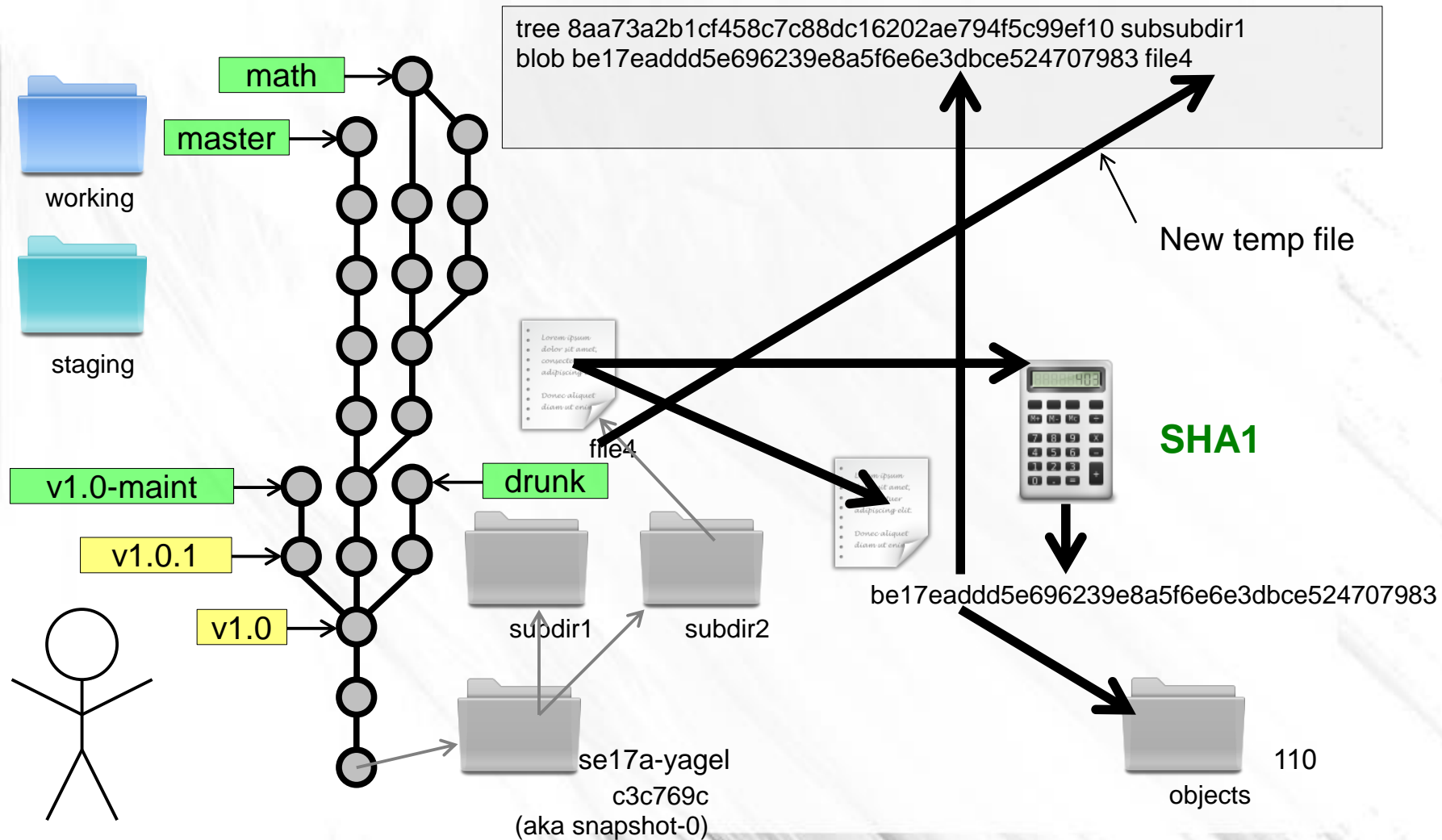
Eliminating Duplication



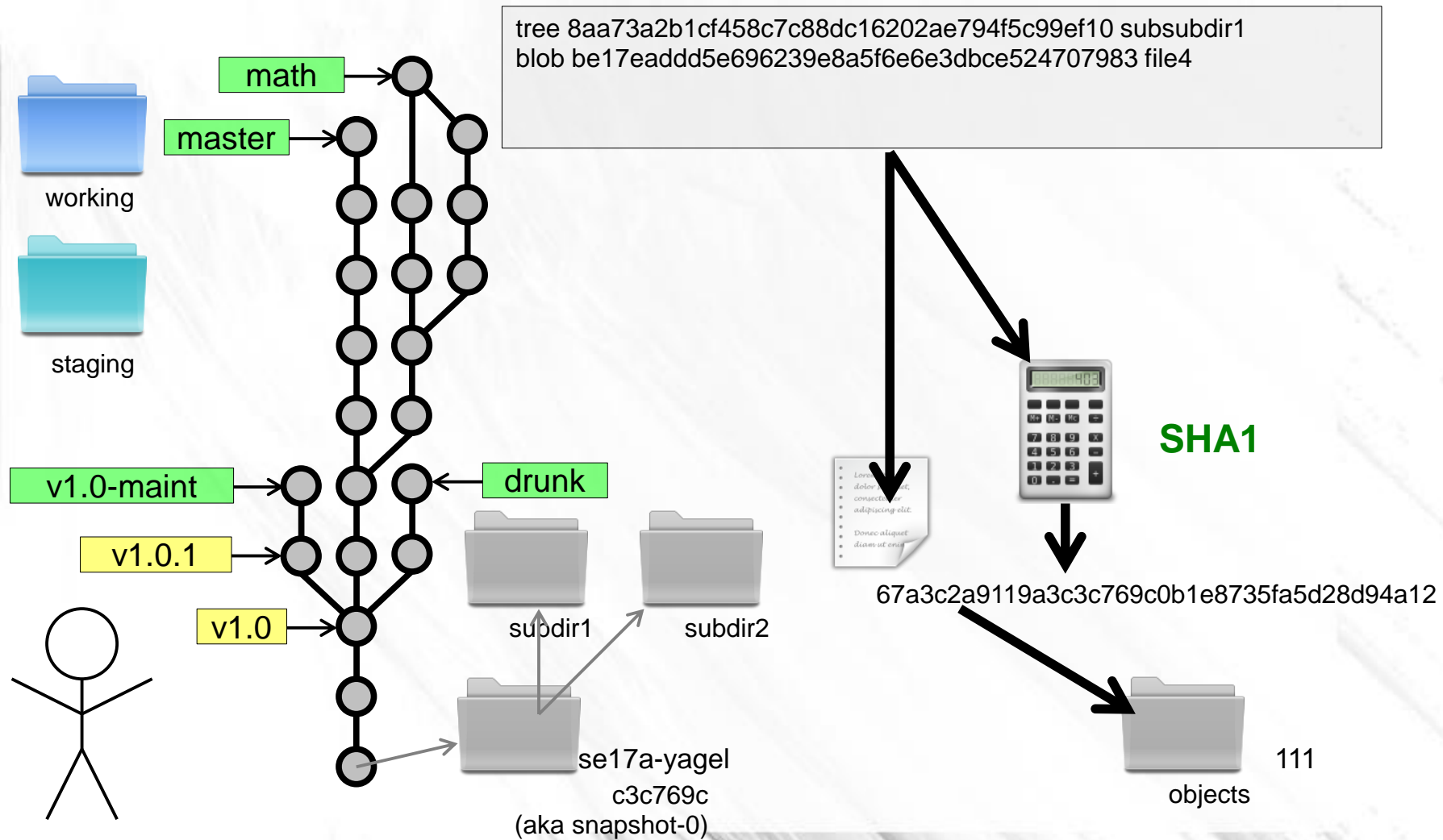
Eliminating Duplication



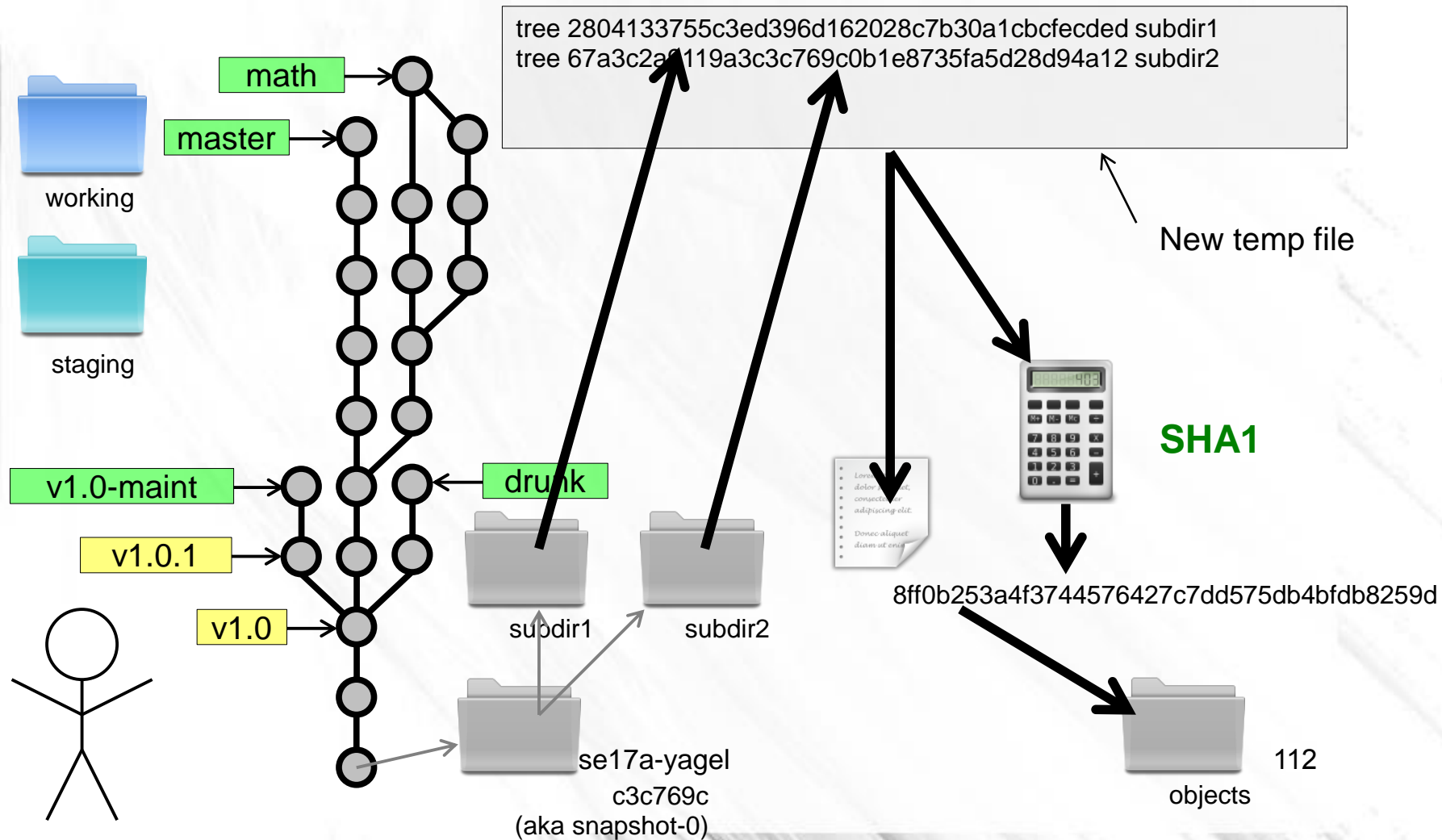
Eliminating Duplication



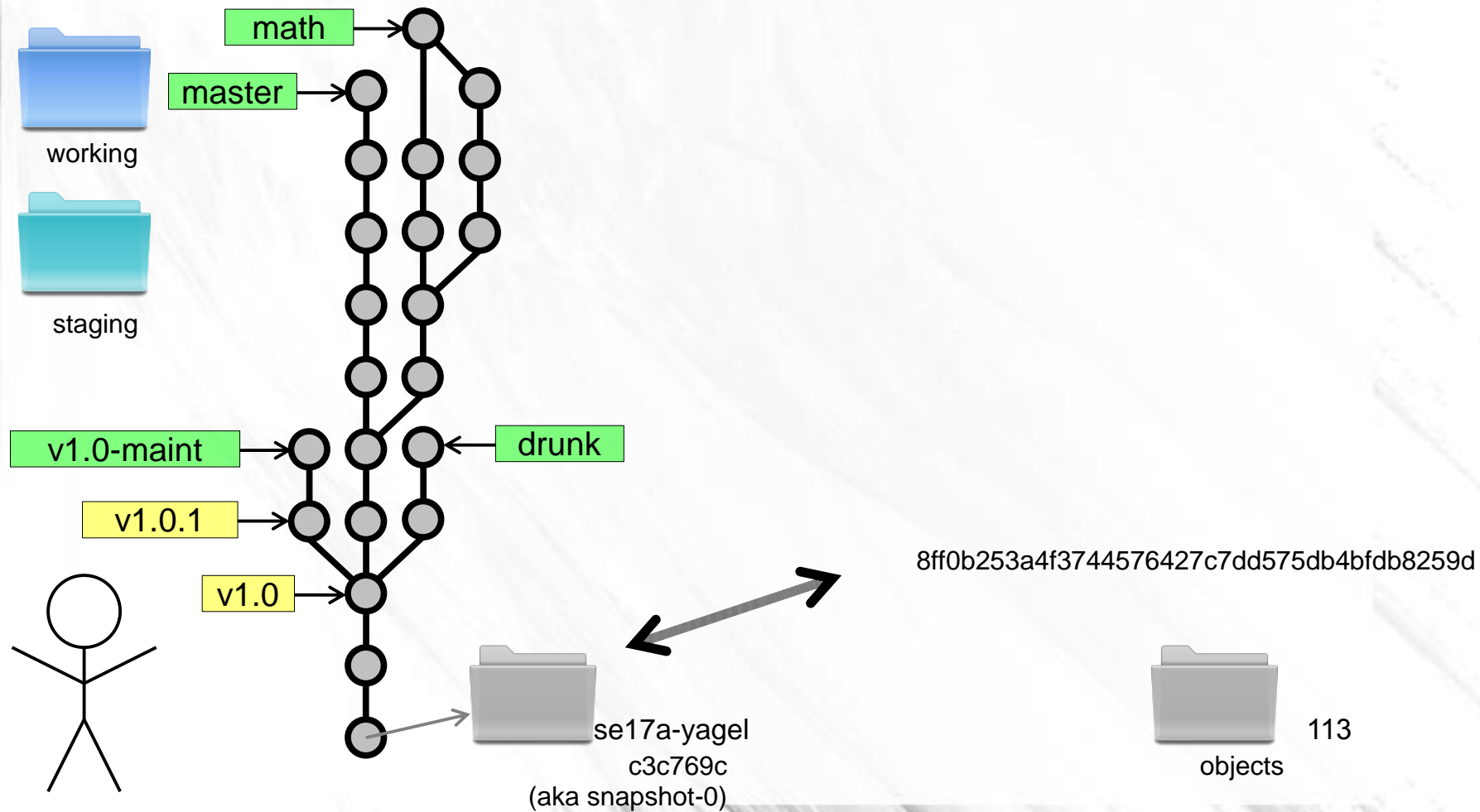
Eliminating Duplication



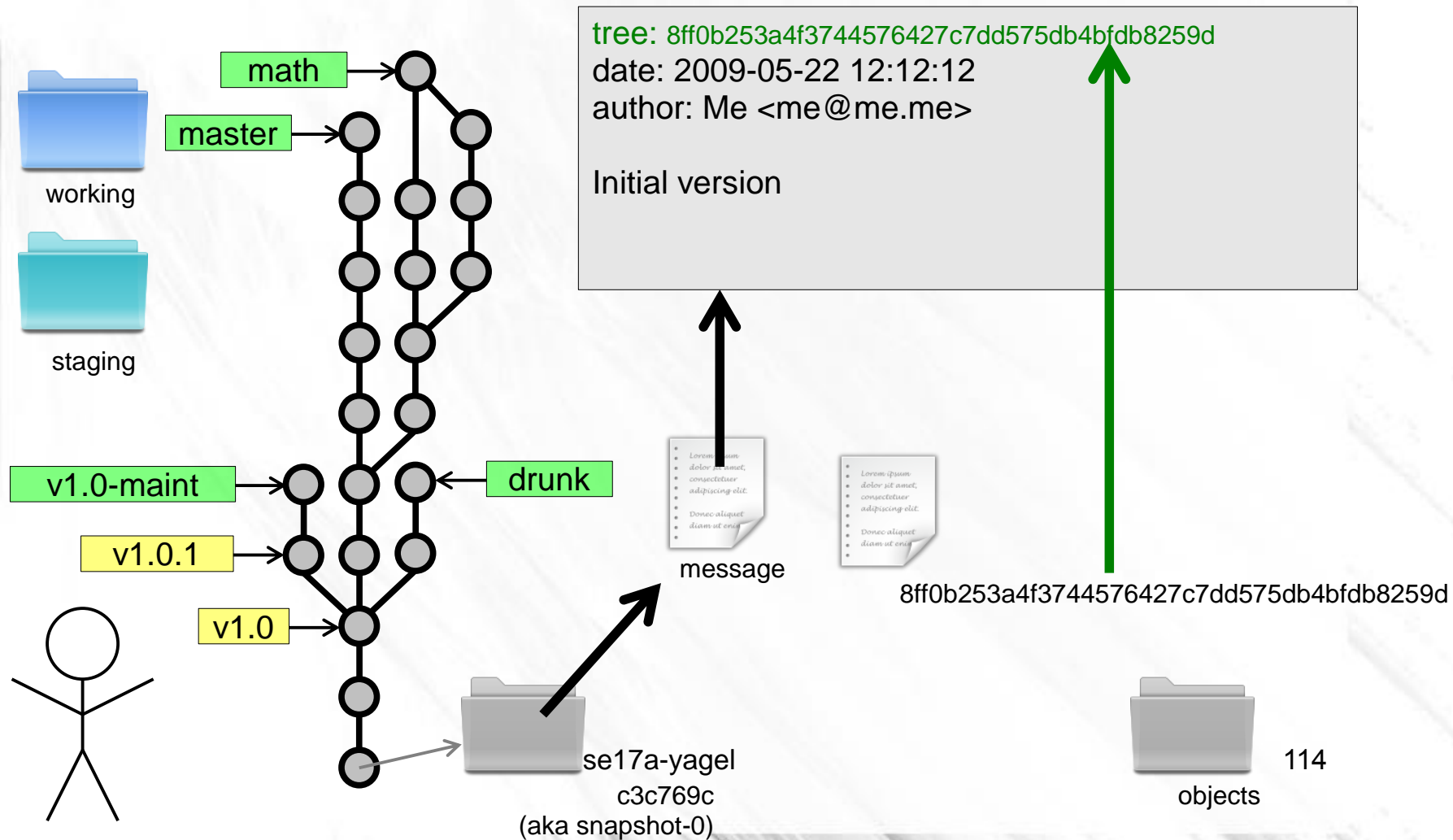
Eliminating Duplication



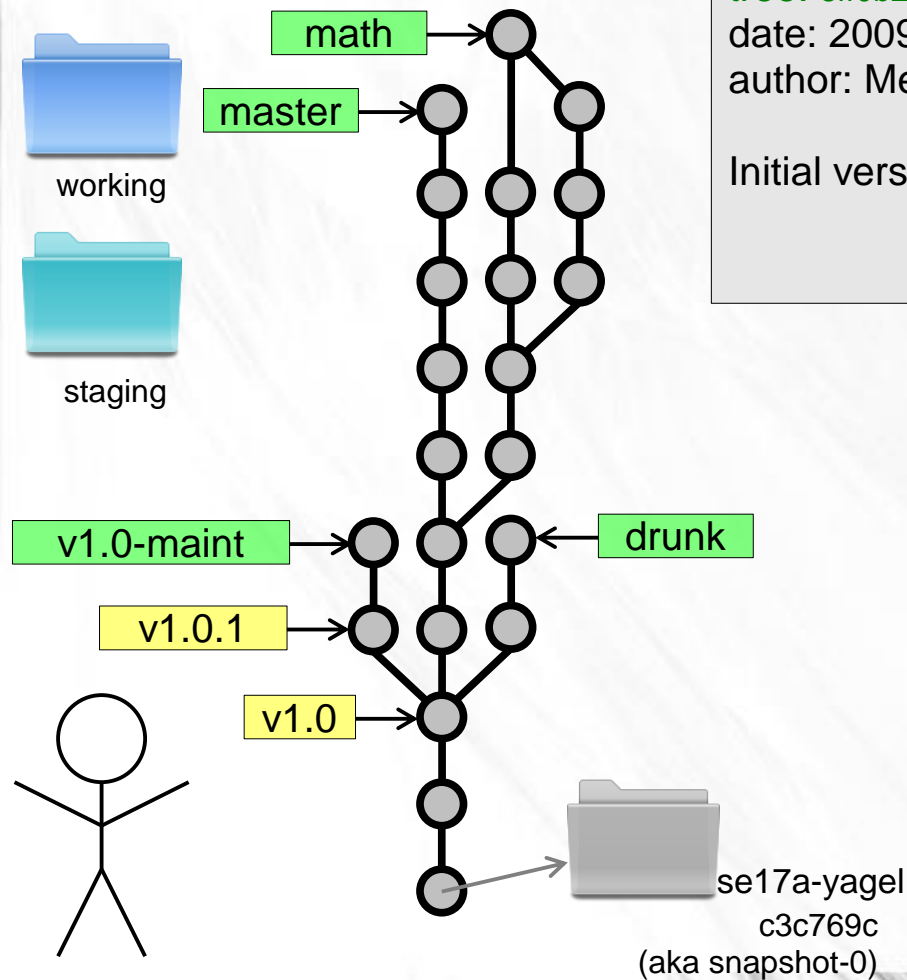
Eliminating Duplication



Eliminating Duplication



Eliminating Duplication

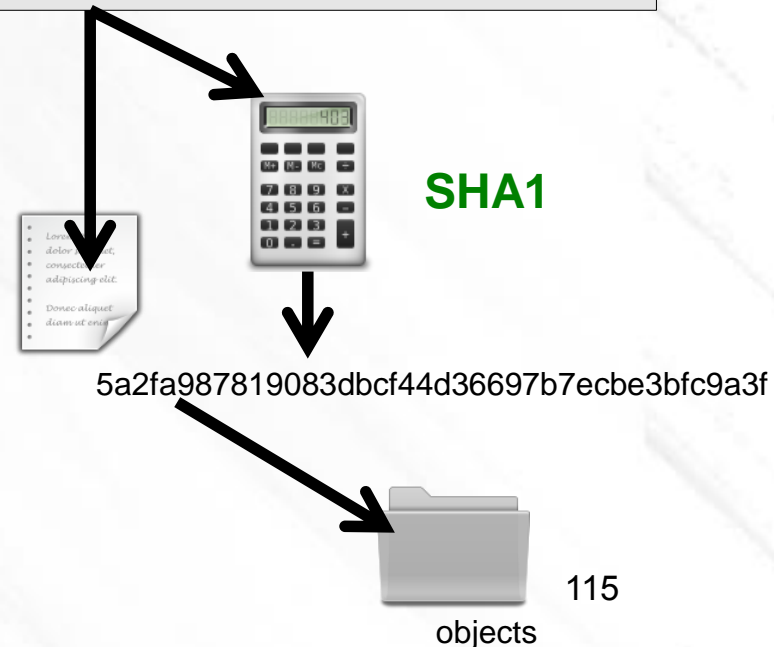


tree: 8ff0b253a4f3744576427c7dd575db4bfdb8259d

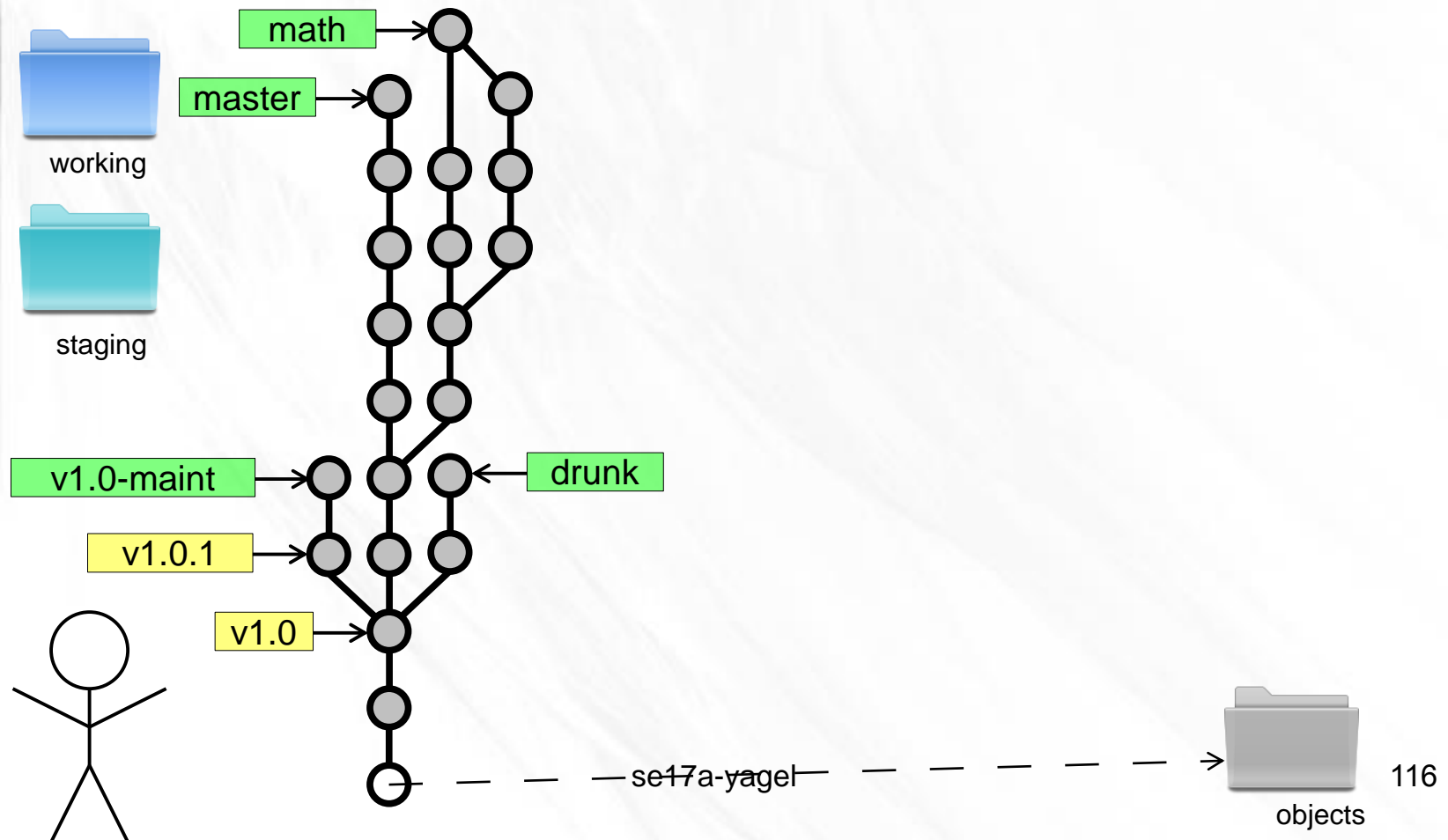
date: 2009-05-22 12:12:12

author: Me <me@me.me>

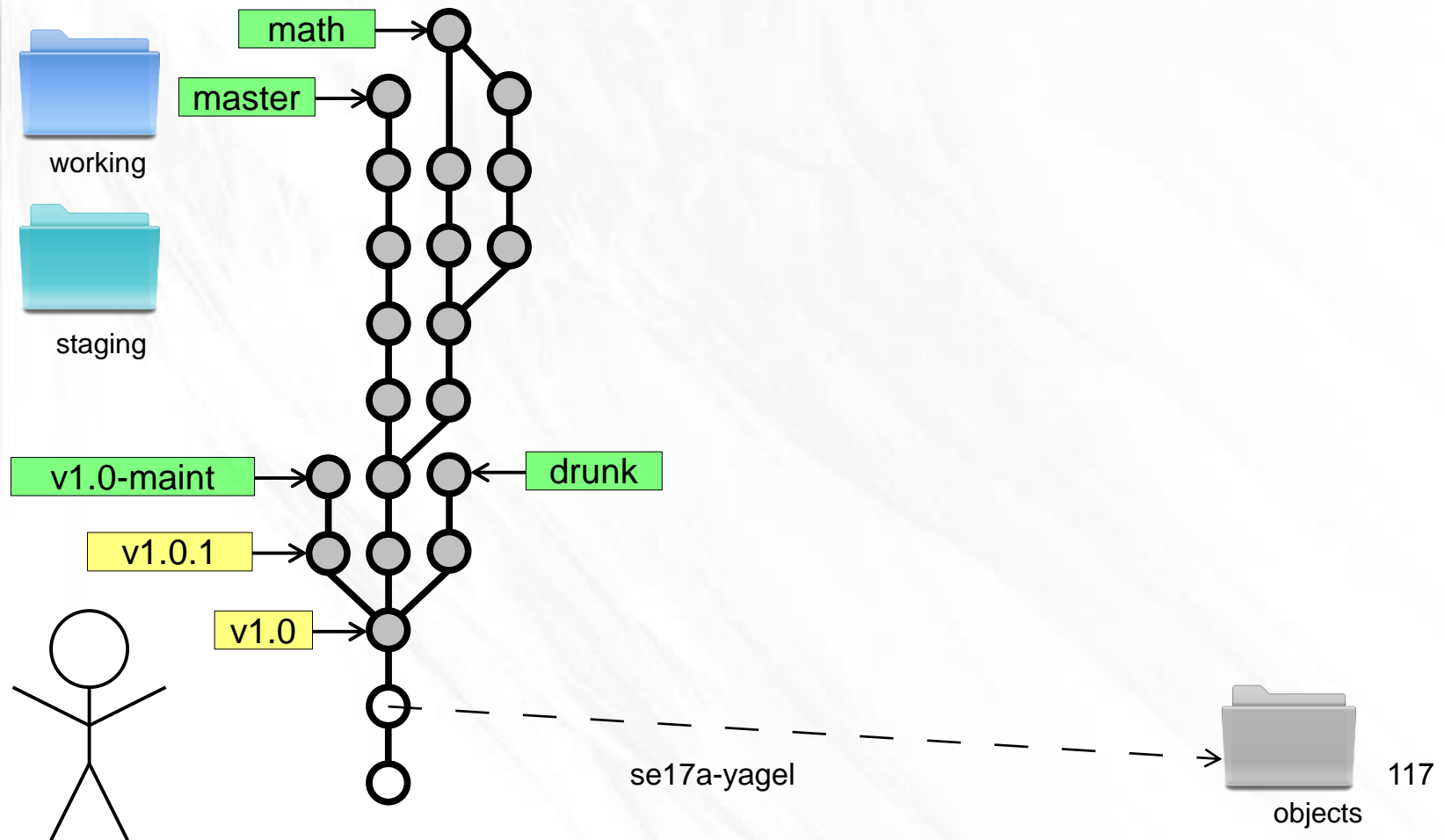
Initial version



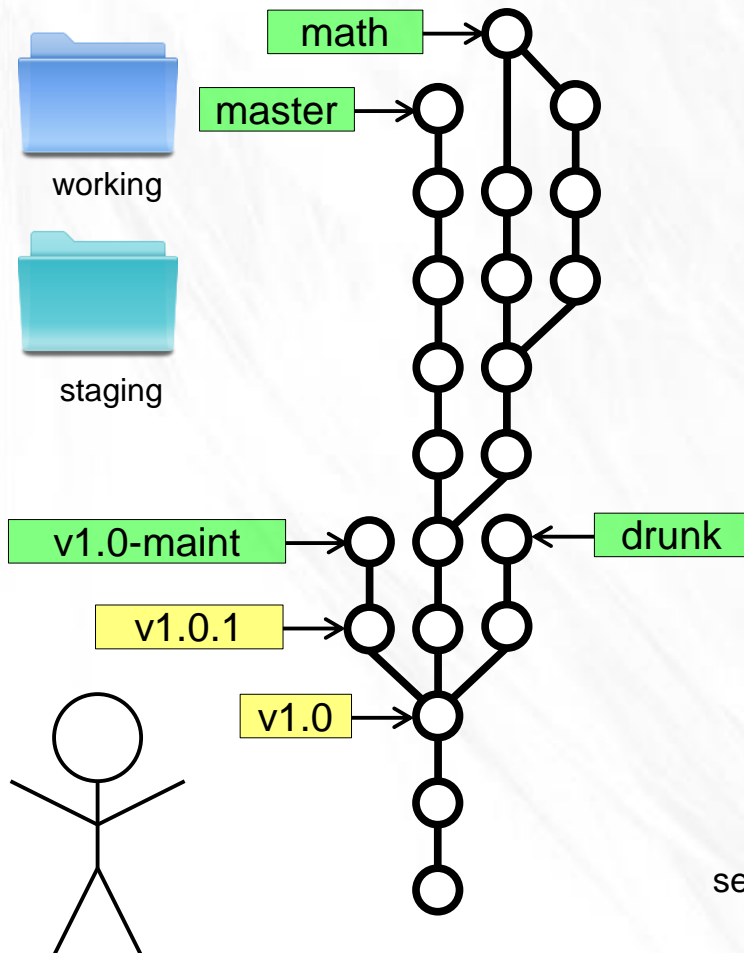
Eliminating Duplication



Eliminating Duplication



Eliminating Duplication



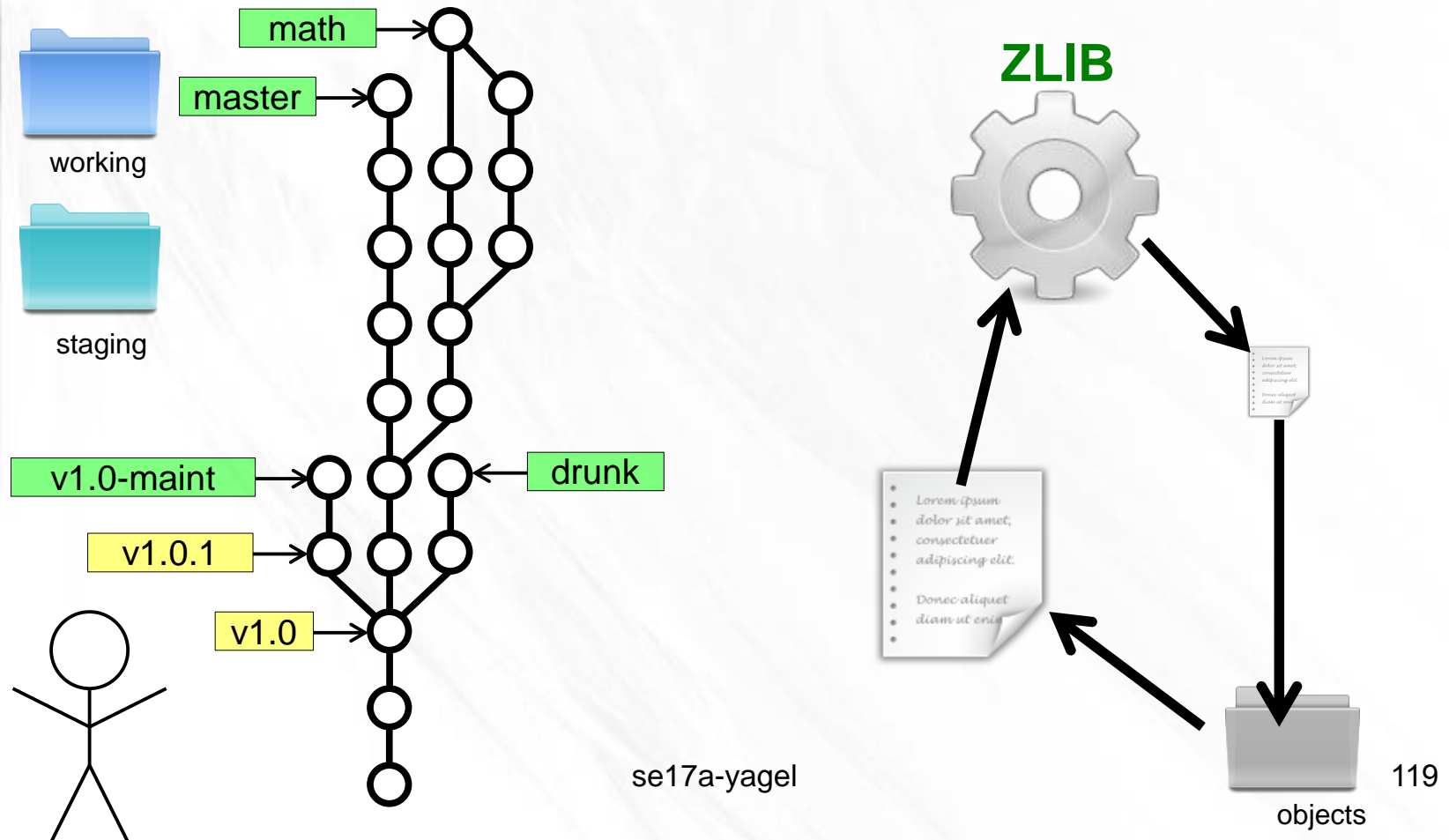
se17a-yagel



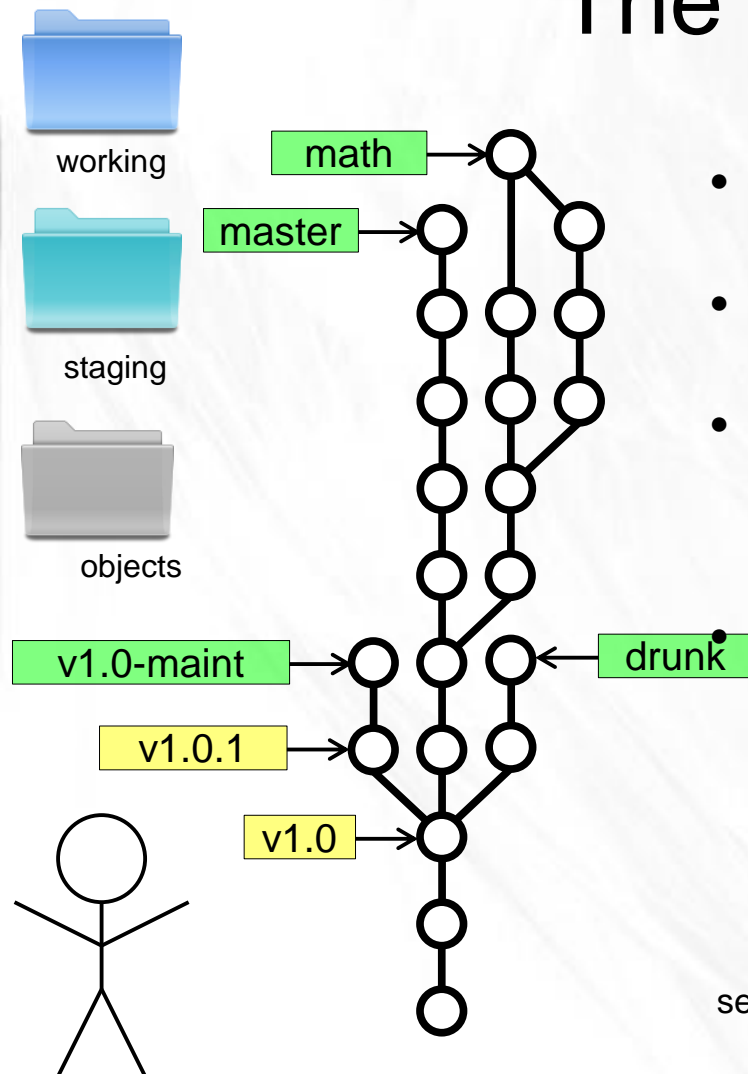
objects

118

Compressing Blobs



The True Git

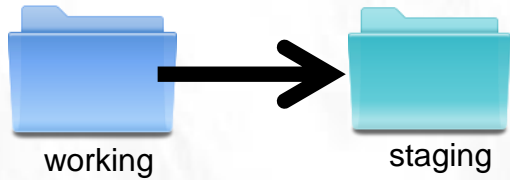


- TADAA!
- This is pretty much Git
- Nicer command line tools for all these operations
- Many, many other tools

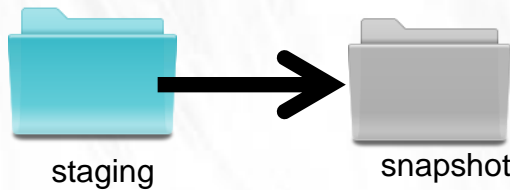
Commands: Getting Started

- 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`



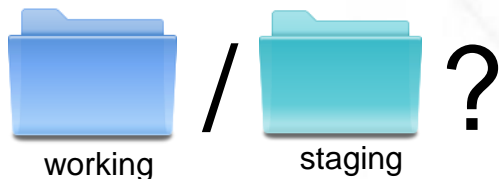
- `git commit`

}

```
git commit -a
```

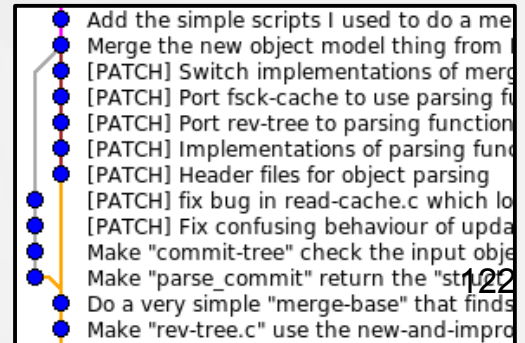


- `git log`



se17a-yagel

- `git status`



gitk

Commands: Diffing



vs.



- `git diff`



vs.



- `git diff --staged`



vs.



- `git diff HEAD`



vs.



se17a-yagel

- `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>`

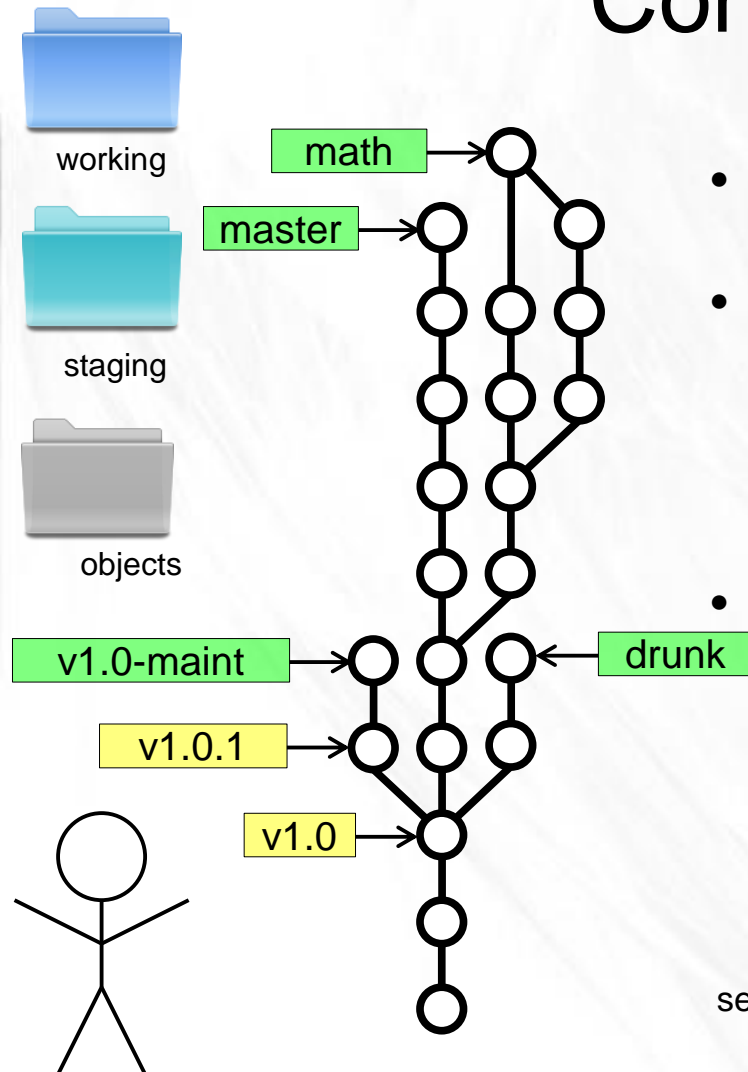
- `git fetch <name>`

}

`git 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>



Git Clients (Windows)

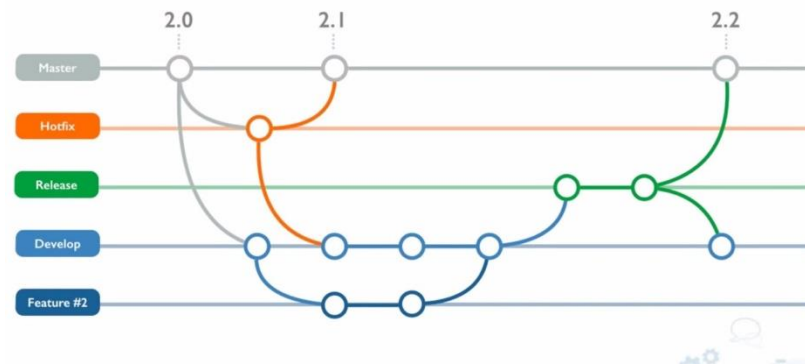
- CLI Shell: Git Bash
- Windows Explorer Shell
- Github Desktop (+powershell)
- Bitbucket SourceTree
- IDE Integration
 - Visual Studio (Code / VS2013+ native)
 - Eclipse Egit
 - IntelliJ, Webstorm embedded
 - Brackets plugin

הדגמה \ שב 3- טיפים

- <http://gitimmersion.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 אישית מס' 3 – ליווי גם בהרצאה
- בדיקות, פיתוח מונחה מפרטים
- שעה שלישית – מעבדה (משימה אישית 4, להביא מחשבים)
- בהמשך
- ? בקרת גרסאות || – תרחישים נוספים עם git
- עקרונות תיכון מונחה עצמים ועוד

לסיכום

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