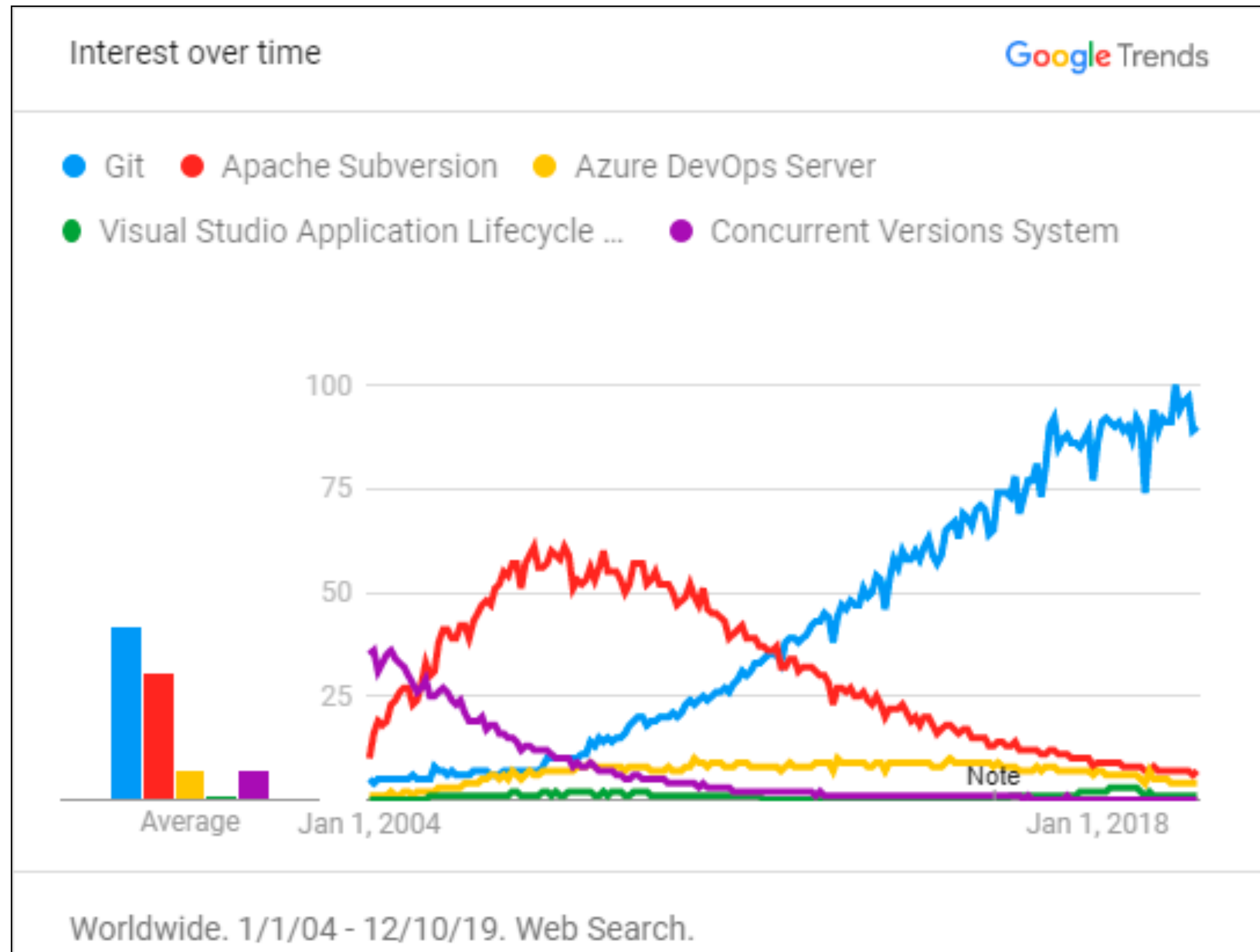


Git is a great design!

in which we will become git experts *and* better software architects in one talk

Aur Saraf, aur@sarafconsulting.com



git ate the world

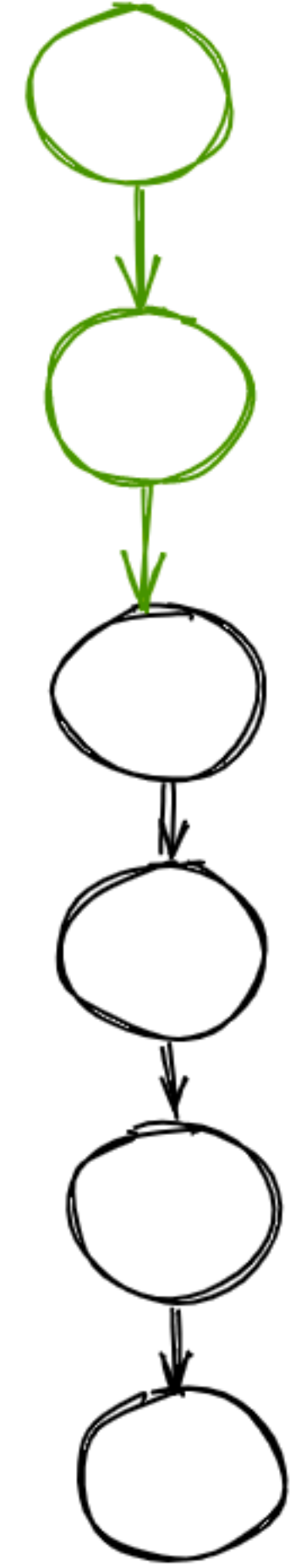
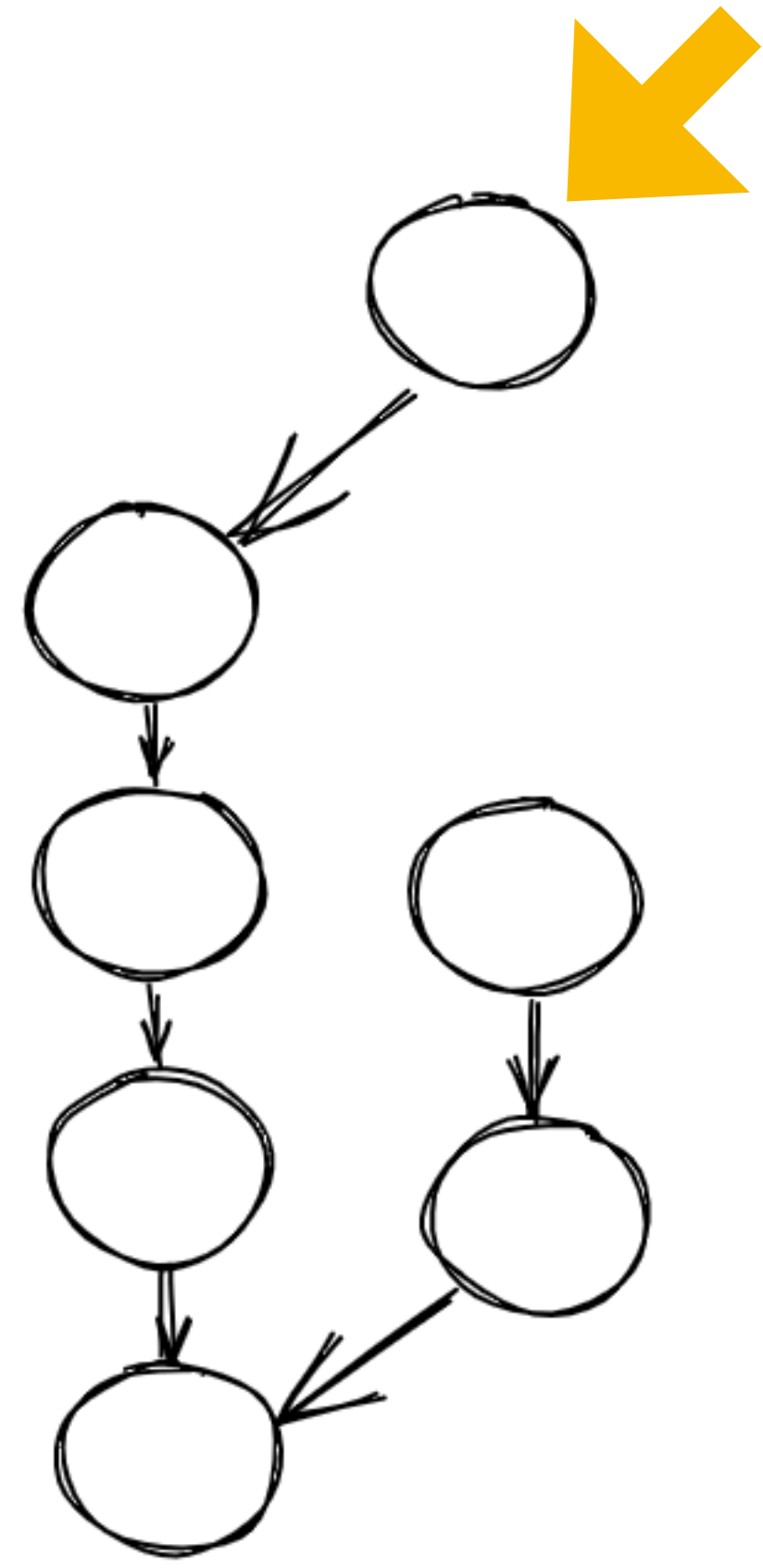
what is git

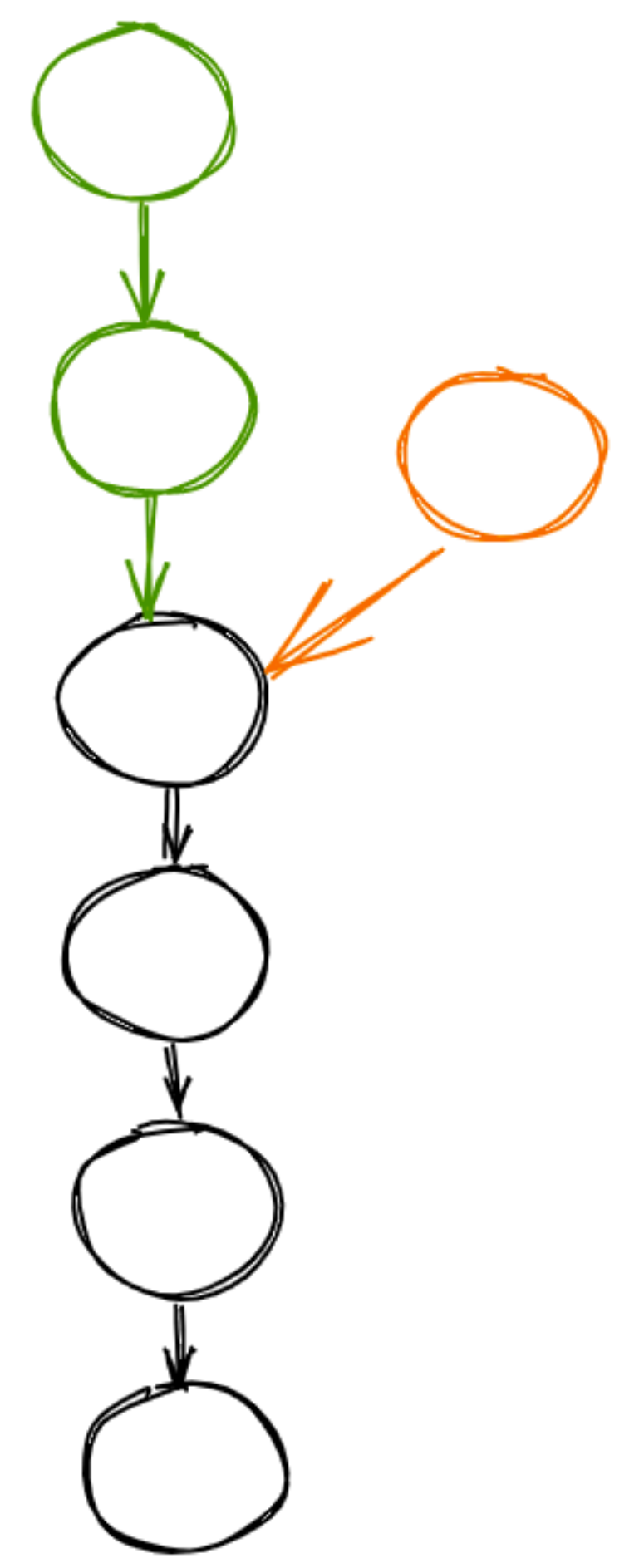
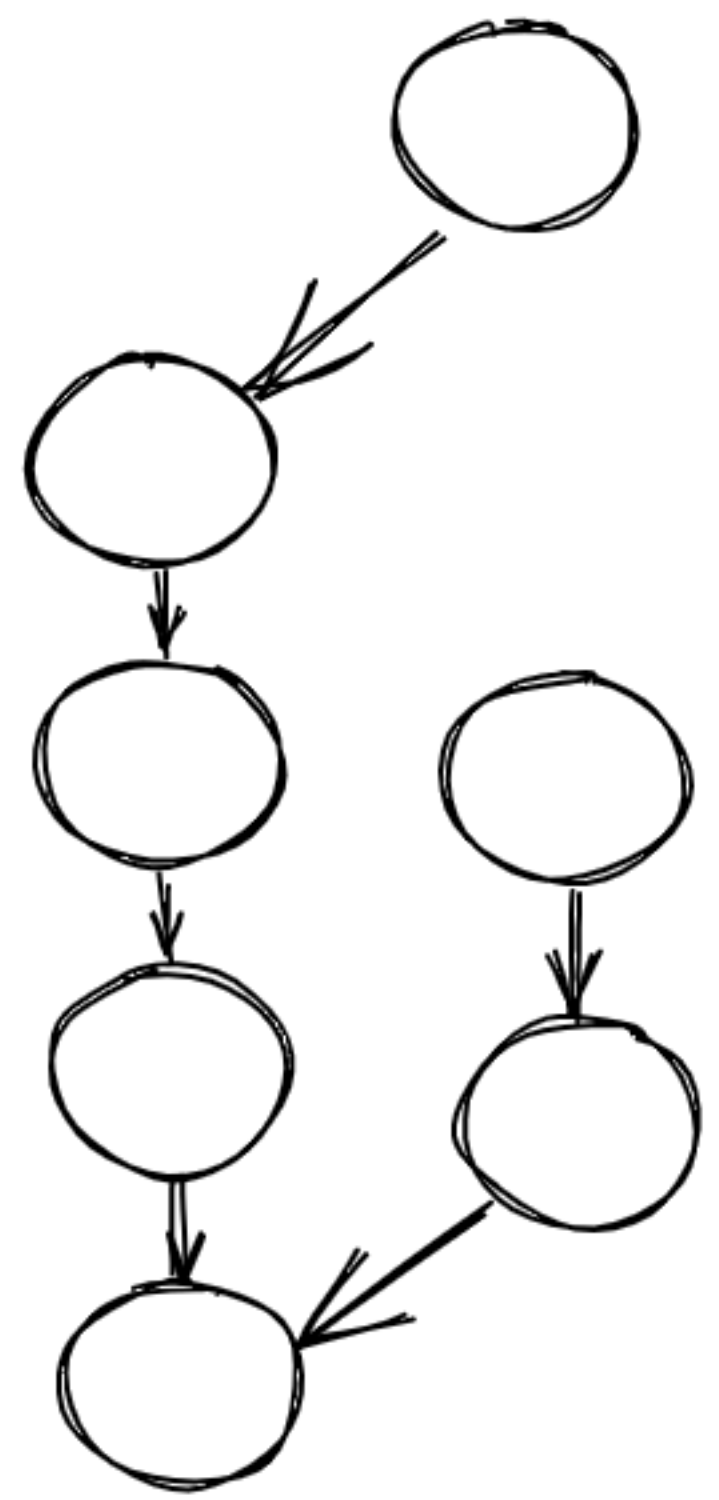
a git repository is a distributed, versioned store of directory trees... with a version control system implemented over it

how git works

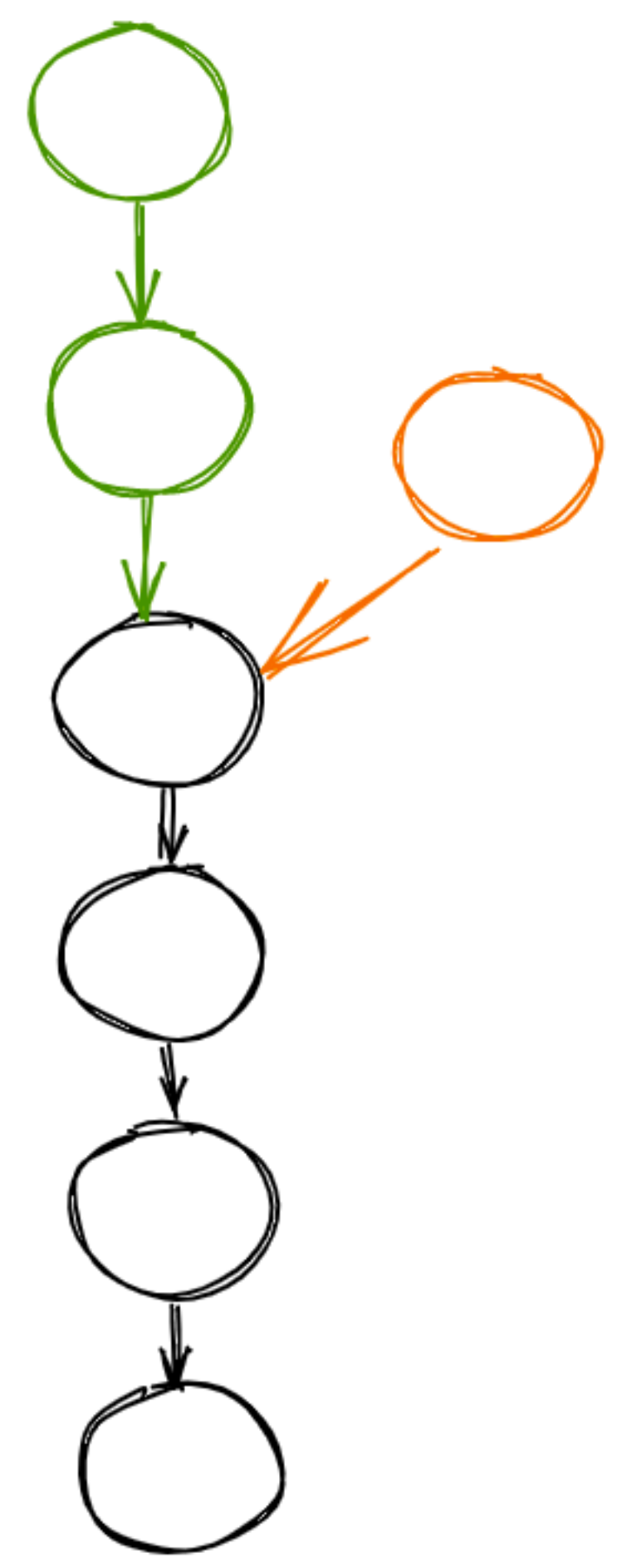
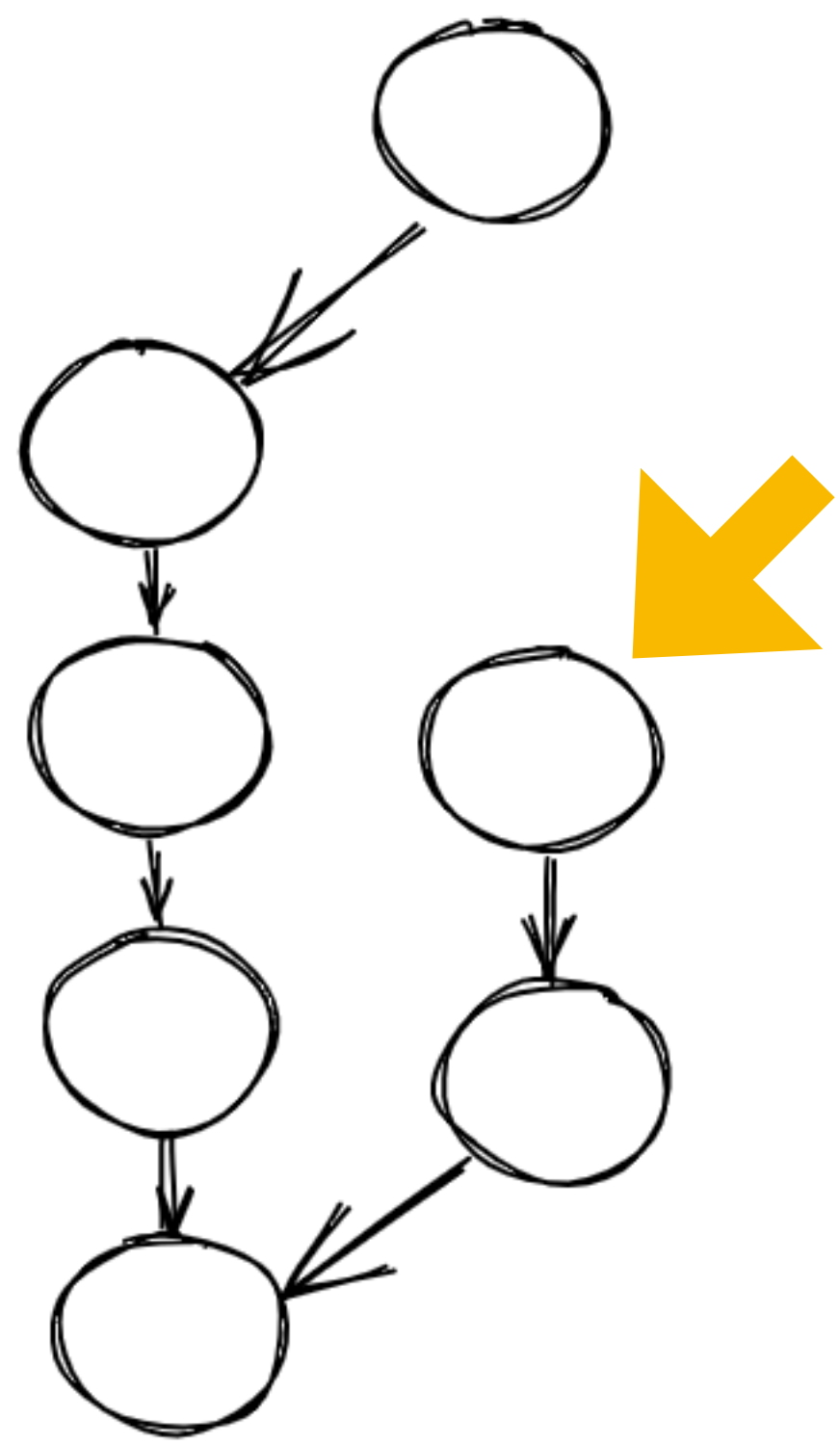
**a repository is a key-value object store
where the key is the object's
cryptographic hash**

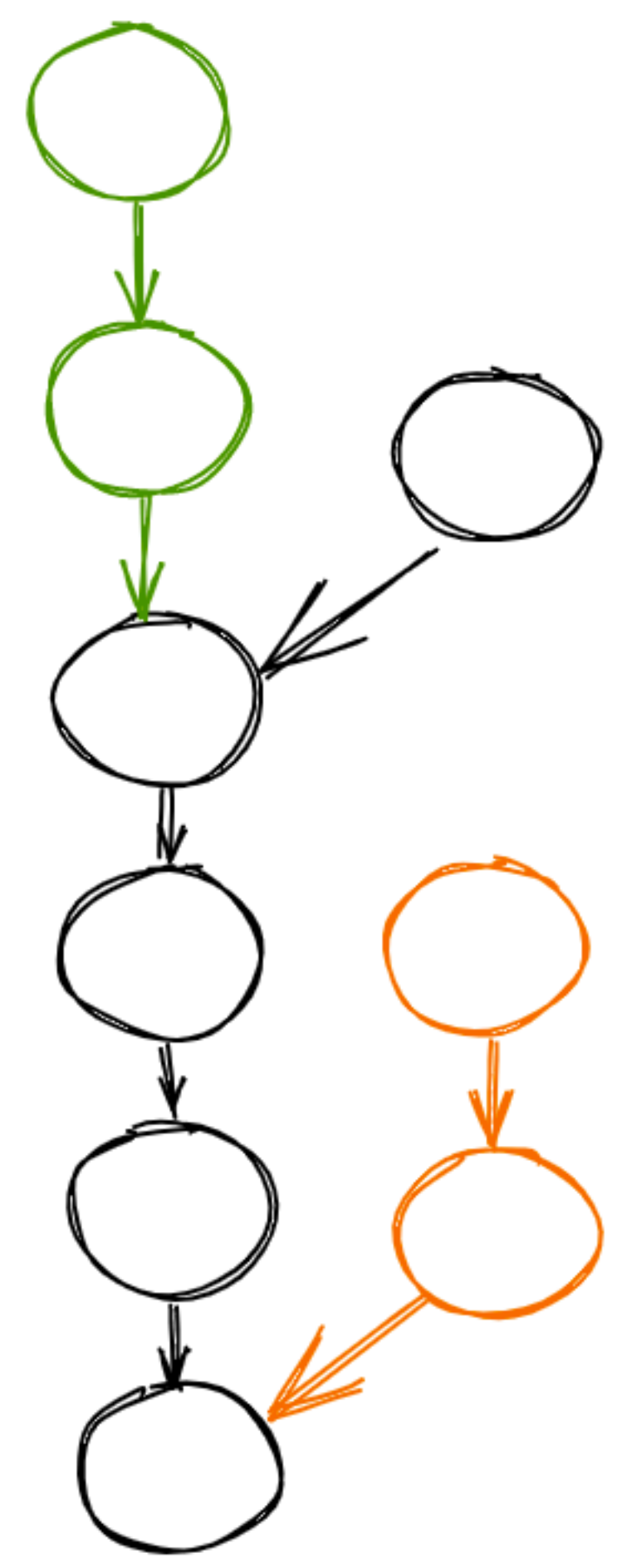
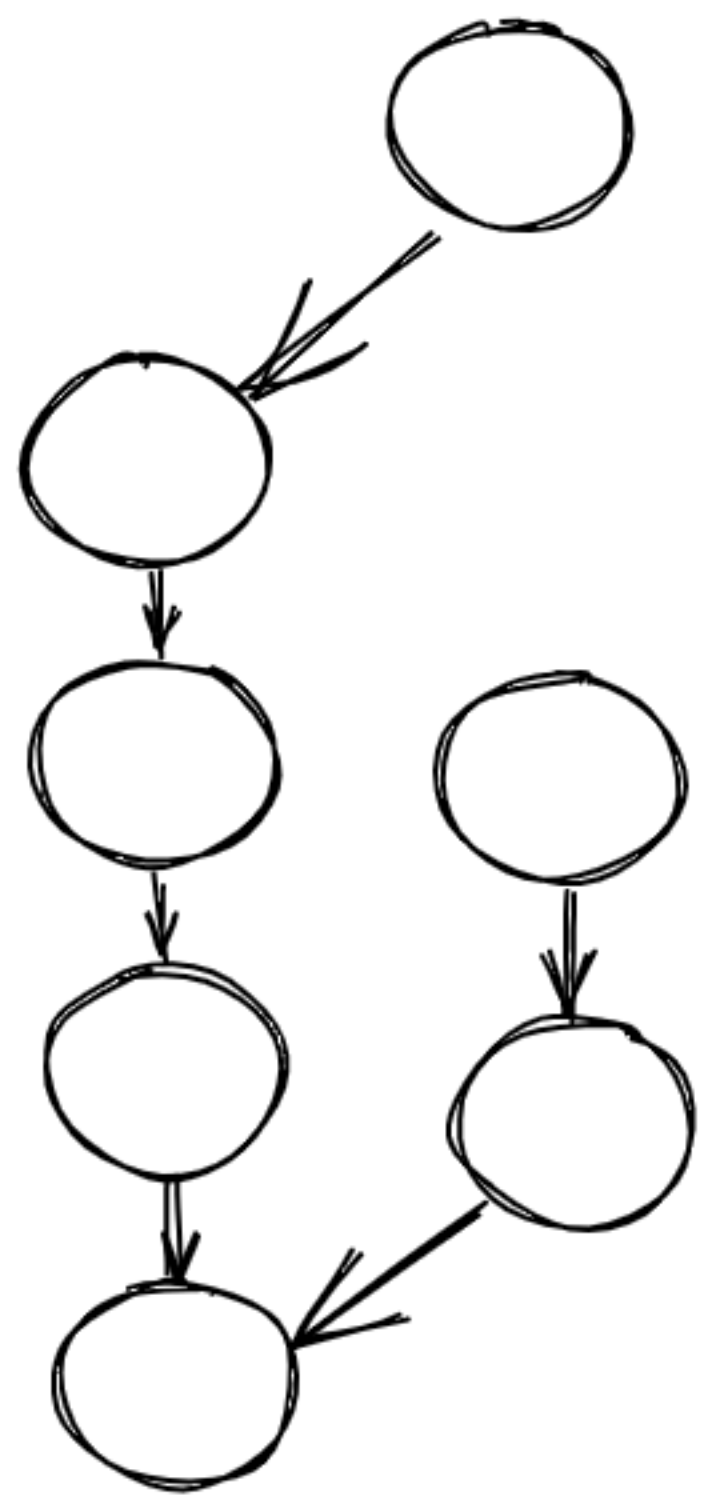
**this is trivial to synchronize between
instances**





objects point to other objects by their keys, which means an object's hash specifies the entire tree



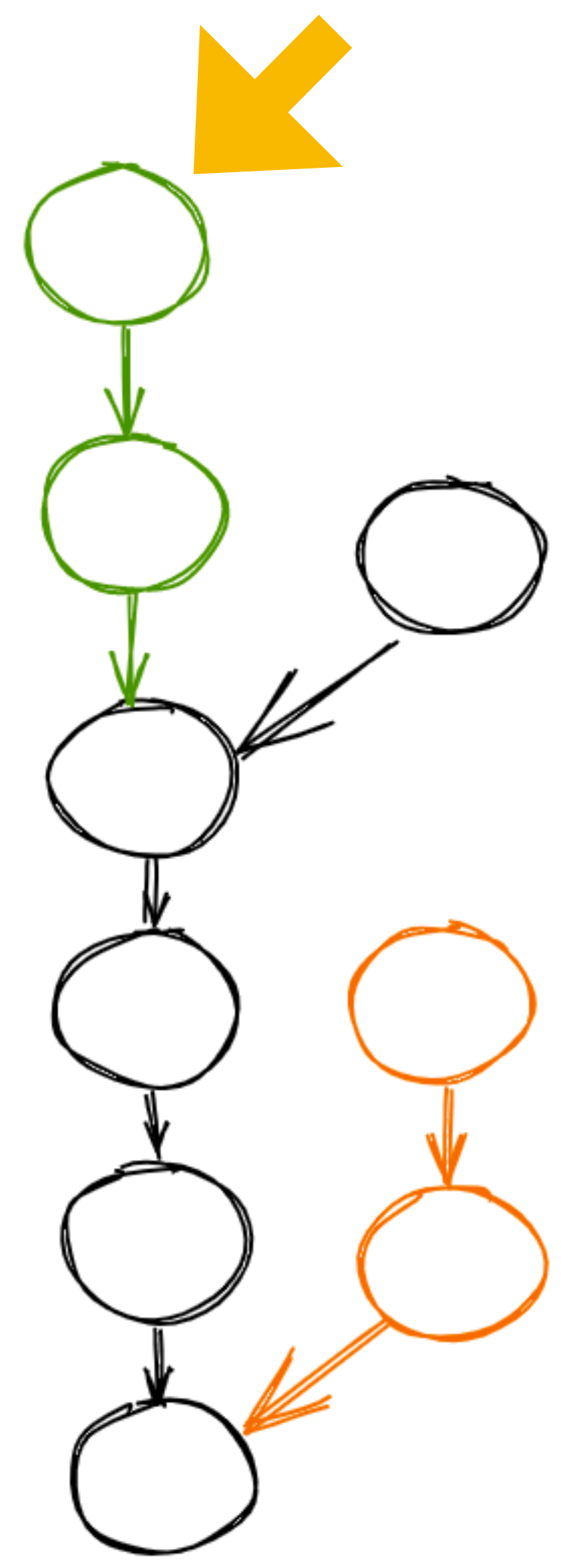
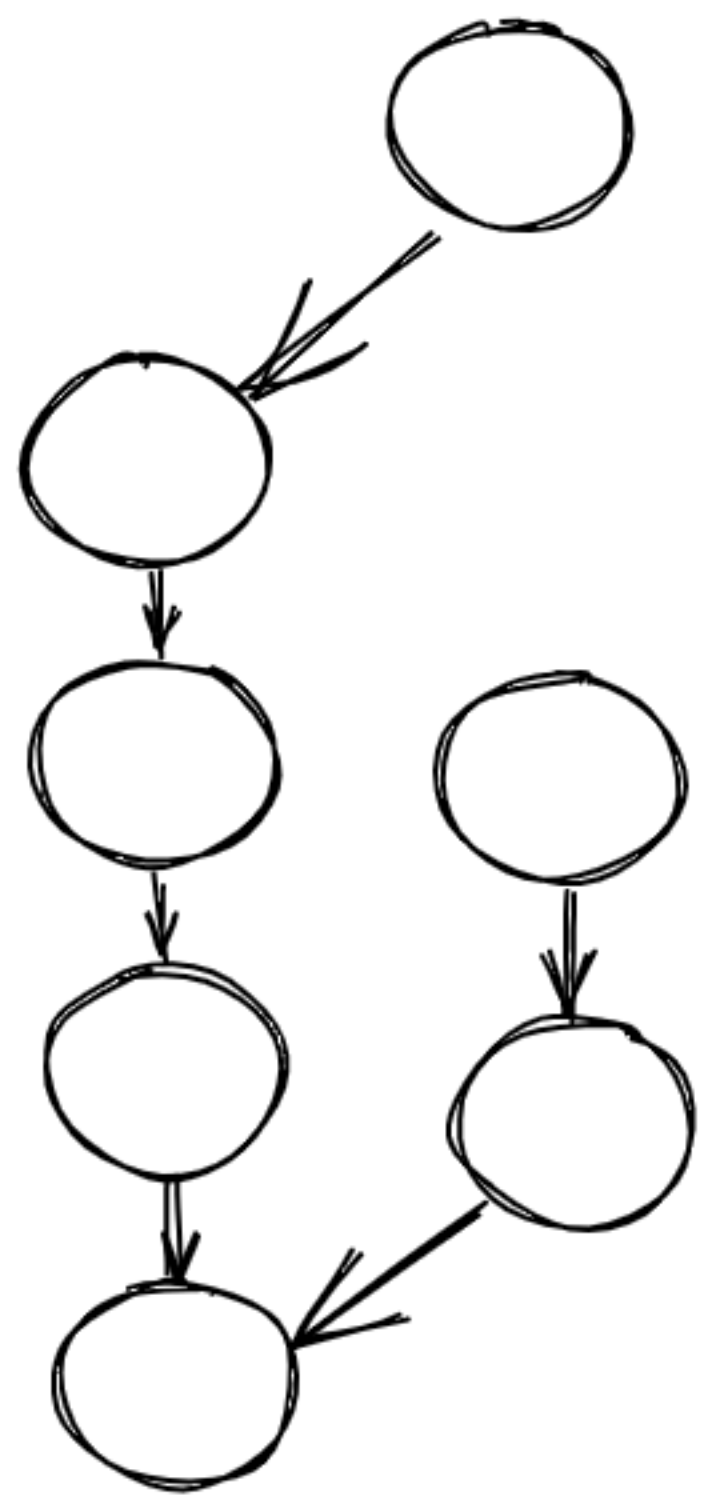


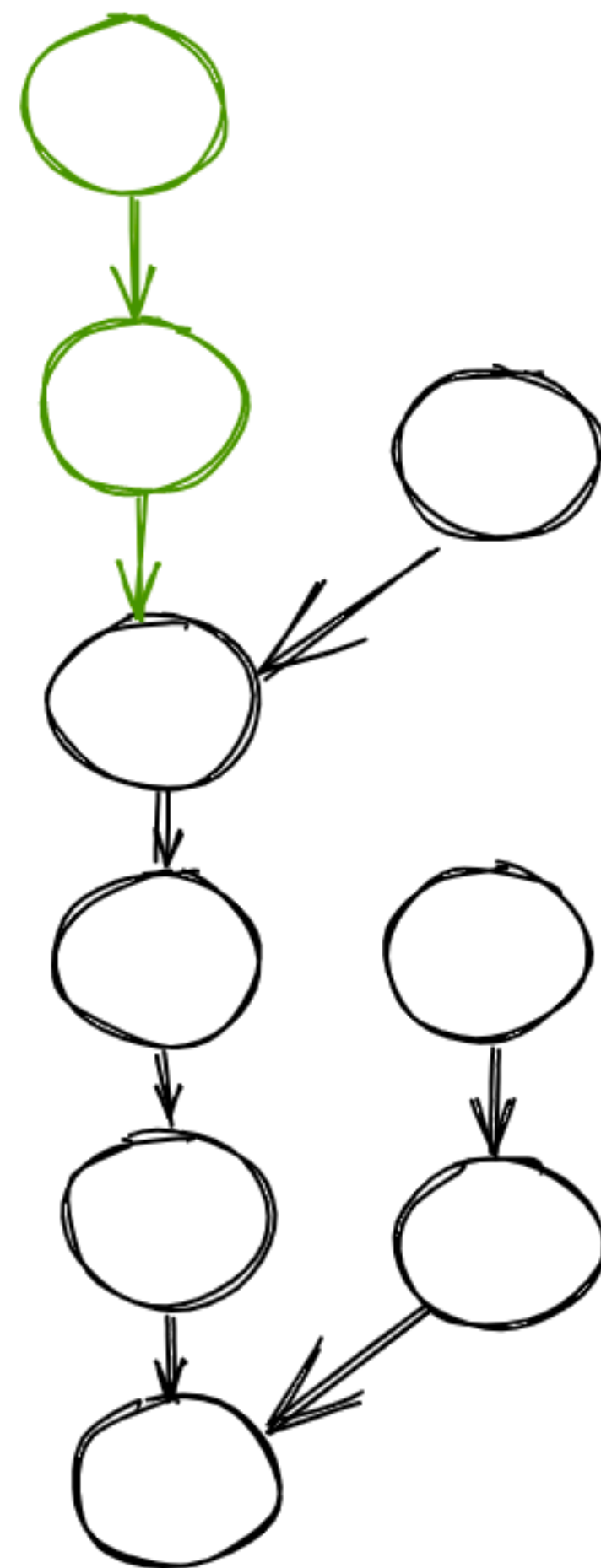
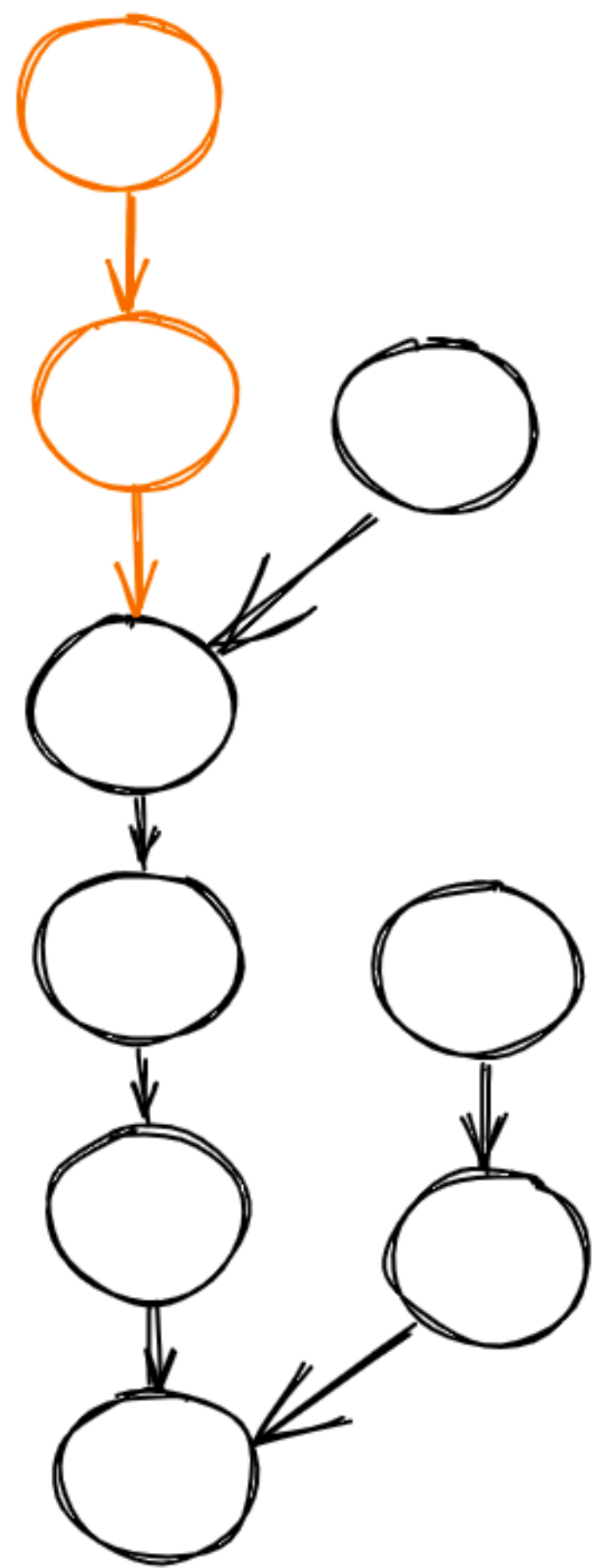
which is the server and which is the client?

there is no "server" or "client"

there is just... "repository"







note on hash collisions:

note on hash collisions:

**cryptographic hash
collisions never happen**

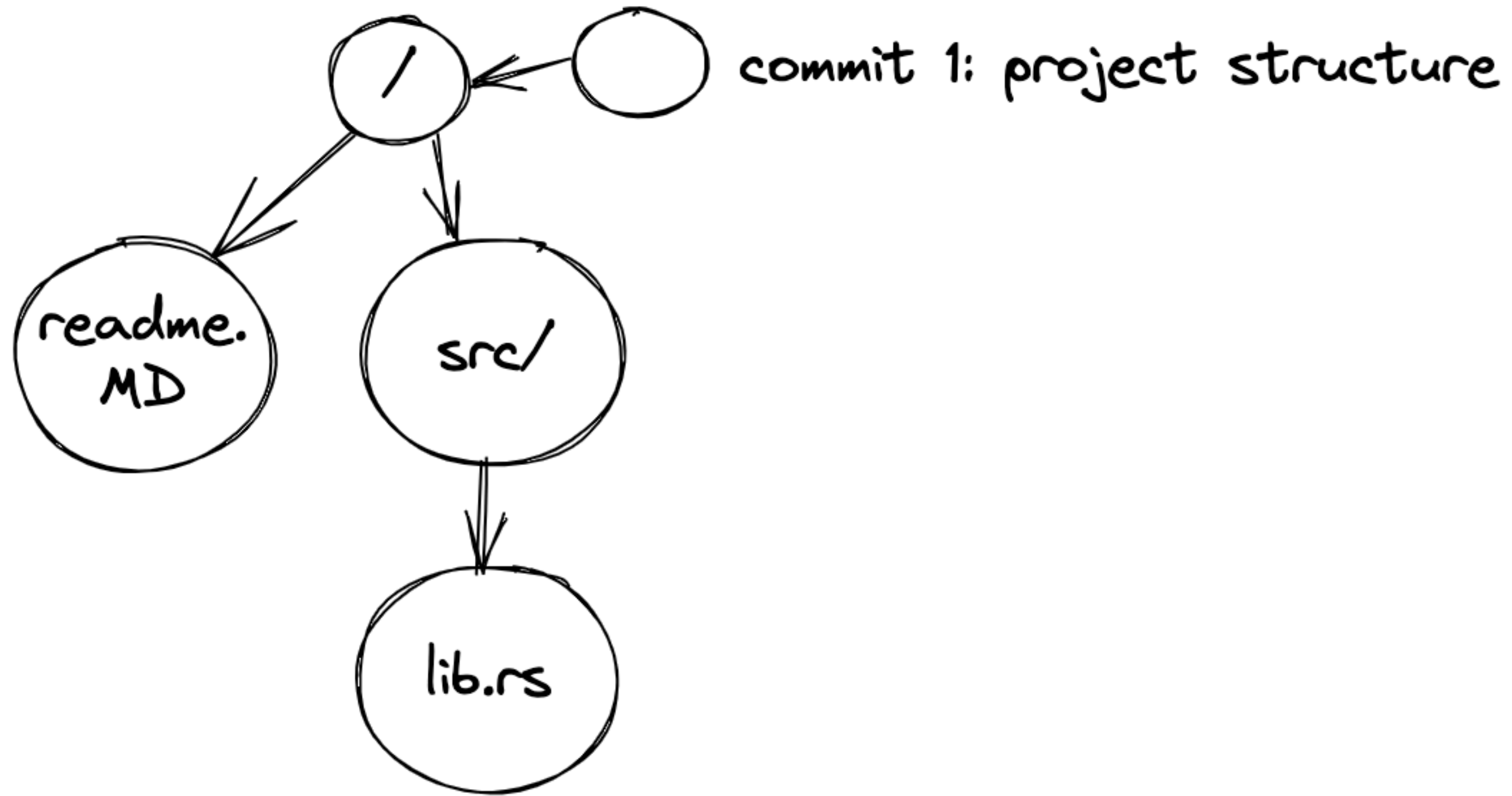
note on hash collisions:

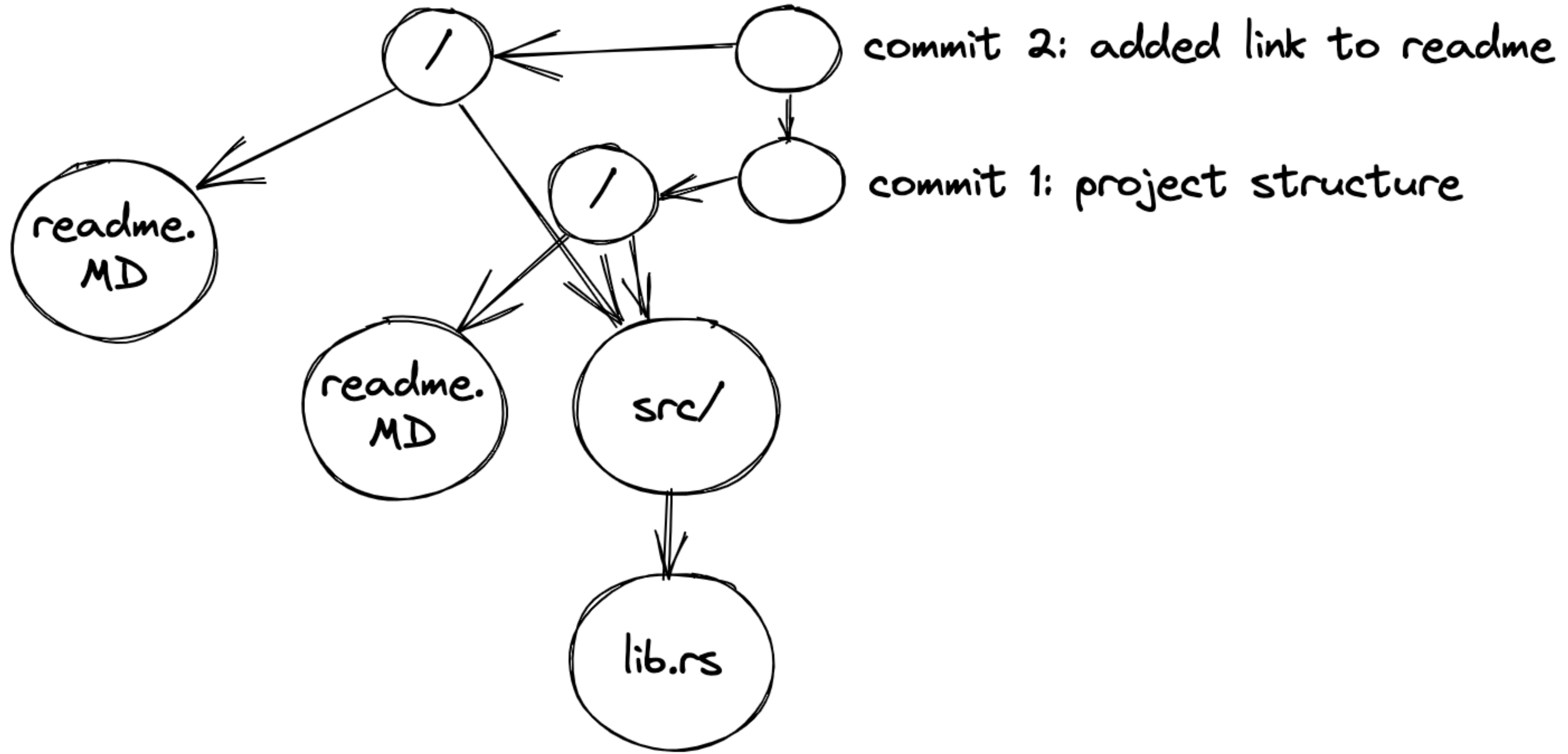
**cryptographic hash
collisions never happen**

thanks for attending my TED talk

version control with git

a version is called a **commit** and usually has
one or more **parents**





a branch is a commit label that moves,
a tag is a commit label that doesn't

```
$ ls .git
COMMIT_EDITMSG HEAD          ORIG_HEAD          config            description
hooks/          index                info/             logs/            objects/
packed-refs     refs/
$ cat .git/HEAD
ref: refs/heads/aur
$ ls .git/refs/
heads  remotes tags
$ ls .git/refs/heads
aur  main
$ cat .git/refs/heads/aur
b9172b443f61719d3fde1a70b5191a7f5e82d057
```

```
git commit -m "added link to  
readme" README.md
```

**(1) creates a new commit pointing to
the current HEAD (2) write hash of new
commit to branch file**

objects are stored in `objects/` by their hash

```
$ cat .git/refs/heads/aur
b9172b443f61719d3fde1a70b5191a7f5e82d057
$ ls .git/objects/
00/ 04/ 09/ 13/ 19/ 21/ 28/ 2b/ 3b/ 3d/ 4c/ 51/ 54/ 57/ 5b/
...
info/ pack/
$ ls .git/objects/b9
172b443f61719d3fde1a70b5191a7f5e82d057
$ git cat-file -p b9172b443f61719d3fde1a70b5191a7f5e82d057
tree 7faf7f4956a3265a4446f2e87c845630ac0d14eb
parent d62f04ec1ebe4e02e775f24c2434fb04bce26a76
author Aur Saraf <aur@sarafconsulting.com> 1662287945 +0300
committer Aur Saraf <sonoflilit@gmail.com> 1662287945 +0300
```

maxpool2d

objects are stored in objects/ by their hash

```
$ git cat-file -p b9172b443f61719d3fde1a70b5191a7f5e82d057  
tree 7faf7f4956a3265a4446f2e87c845630ac0d14eb
```

```
...
```

```
$ git cat-file -p 7faf7f4956a3265a4446f2e87c845630ac0d14eb  
100644 blob bcc6aeb5fd0885e4e0fe981e49890c2f106250cd      .gitignore  
100644 blob 20cb8b8e1d53e333696acc79fe045ae7e9b0aa74    .markdownlint.json  
040000 tree dbb3afc8121bd6a314b3aa1e3711877630e91729    .vscode  
100644 blob cca8123bc494fd36389a6b48a267b138b3db18d7    README.md
```

```
...
```

```
$ git cat-file -p cca8123bc494fd36389a6b48a267b138b3db18d7  
# MLAB August 2022
```

```
## Current Status and List of Significant Changes
```

```
Last updated Aug 5 2022
```

```
...
```

git is divided to "plumbing" and "porcelain"

"porcelain" refers to user-friendly
commands that expose a version control UI
over the distributed dirtree store,
historically **simple shell scripts** that invoke
plumbing

how to use git

the "index" or "staging" is like a "shopping
cart" of changes, you gradually add
changes and then commit them into a...
commit

git add, git reset add/remove from **index**

git commit create commit from **index**

git status see index

git log see commit list/tree

git diff compare versions of objects

git merge make a commit based on HEAD and another ref
(might need to resolve conflicts, default merge algo is stupid)

git rebase new commits by applying commits' diffs on top of
ref (might need to resolve conflicts after each commits)

git rebase -i HEAD^^^ edit last 3 commits

git fetch synchronize from remote

git pull fetch then merge

git push synchronize to remote

git branch create/remove/edit commit a branch points to

git switch move to different branch (with **-C:** to new branch)

git has many ways to name objects:

hash

label, tag, HEAD

ref[^] - first parent of ref

ref^{^^^} - first great-grandparent

**ref@{2} - what ref used to point to 2
changes ago**

ref:path/to/obj - file object

messed things up real bad?

git reflog show all hashes labels/tags *used to point to*

```
e1b23ba (HEAD -> master) HEAD@{0}: commit: src -> _ke/src, build with setuptools_rust
...
b776923 HEAD@{7}: commit: README: minor rearrangement
1c1c6fc HEAD@{8}: rebase (finish): returning to refs/heads/master
1c1c6fc HEAD@{9}: rebase (pick): README: cheat sheet, move stuff around, remove bad examples
b2d242b HEAD@{10}: rebase (pick): better demo gif
e4ad7f3 HEAD@{11}: rebase (pick): deleted some stale files
07d71a3 HEAD@{12}: rebase (pick): README: architecture, contributing, improve example
0d35d74 HEAD@{13}: rebase (start): checkout refs/remotes/origin/master
740d77b HEAD@{14}: commit: README: cheat sheet, move stuff around, remove bad examples
```

advanced things you should know exist:

reset --soft, reset --hard

precommit hooks

pgp signatures

submodules

filter-branch

large file storage

git is a great design

Good design chooses the right data structures to represent the problem

Great design chooses the right data structure to represent the problem

Who Am I



aur@sarafconsulting.com

Who Am I



Aur Saraf

December 30, 2020 · 🌐



הדוג'ו היום: הרצאת אורח של [Or Peles](#) על מה קורה כשאתם מריצים שאילתה על מסד נתונים.

דוג'ו התכנות של אאור

April 18 · 🌐

👤 השבוע בדוג'ו design review. חבר בדוג'ו יציג לנו את האפיון שלו לשכתוב של מוצר מסחרי שהוא עובד עליו, ואני ואתם ניתן לו פידבק.

רביעי 20:00-21:30, קישור יישלח כמ... See more

דוג'ו התכנות של אאור

June 28 · 🌐

👤 השבוע בדוג'ו, סקירה של פייתון למי שבא משפה אחרת. כל הרוחב של דברים שכדאי לדעת כשניגשים לכתוב פרויקט גדול בפייתון: מי נגד מי, איך זה עובד, כל

הפיצ'רים בשימוש יו... See more

bonus: my git aliases

```
alias s='git status'
alias d='git diff'
alias dc='git diff --cached'
alias c='git commit -m'
alias ca='git commit -am'
alias a='git add -p'
alias u='git checkout -p'
alias l="git log --graph --oneline --decorate --all --
pretty=format: '%Cgreen%ad%Creset %C(auto)%h%d %s %C(bold
blue)%aN %Cred%G?%GS%Creset' --date=format-local: '%Y-%m-
%d %H:%M'"
alias lx="git log --graph --oneline --decorate --
pretty=format: '%Cgreen%ad%Creset %C(auto)%h%d %s %C(bold
blue)%aN %Cred%G?%GT%Creset' --date=format-local: '%Y-%m-
%d %H:%M'"
```

bonus: my git aliases

```
* 2022-09-04 10:23 e1b23ba (HEAD -> master) src -> _ke/src, build with setuptools_rust
backend default Aur Saraf N
* 2022-09-02 08:50 e5ac45c (origin/master) fix html display Aur Saraf N
* 2022-09-02 08:48 30472e3 railroad diagrams Aur Saraf N
* 2022-09-02 08:36 e1272a2 fight the vertical scroll Aur Saraf N
* 2022-09-02 08:27 3300c31 make links more readable, fix typo Aur Saraf N
* 2022-09-02 08:25 7591e70 README: raw download links Aur Saraf N
* 2022-09-02 08:22 9bb22db README: repeat cheatsheet links Aur Saraf N
* 2022-09-02 08:20 b776923 README: minor rearrangement Aur Saraf N
* 2022-09-02 08:15 1c1c6fc README: cheat sheet, move stuff around, remove bad examples
* 2022-09-02 07:57 b2d242b better demo gif Aur Saraf N
* 2022-09-02 07:55 e4ad7f3 deleted some stale files Aur Saraf N
* 2022-09-02 02:47 07d71a3 README: architecture, contributing, improve example Aur Sara
* 2022-09-02 01:02 0d35d74 Merge pull request #11 from InbarRose/pytest_parameterize
ly@github.com>
|\
| * 2022-01-03 16:15 214a495 refactor test_short_names to use parametrize Inbar Rose N
* | 2022-09-02 01:00 c34327f Merge pull request #9 from InbarRose/master Aur Saraf UGit
_
```