Git and github

version control and collaboration

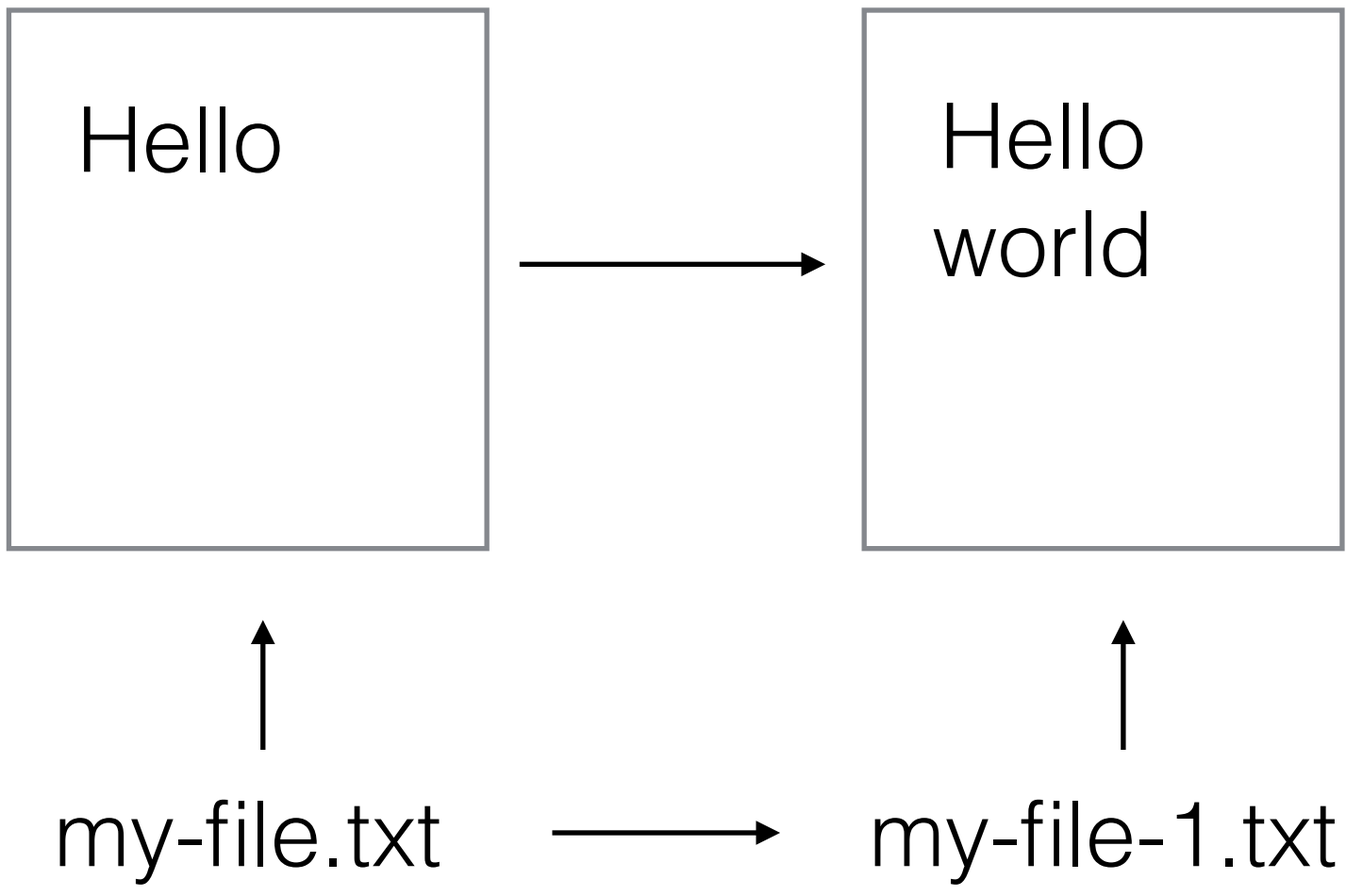
version control

The management of changes to documents.

Not

- my-file.txt
- my-file-1.txt
- my-file-2.txt
- my-file-2-new.txt
- my-file-2-final.txt
- my-file-2-final-2016-03-15.txt
- my-file-2-final-2016-03-15-v1.txt

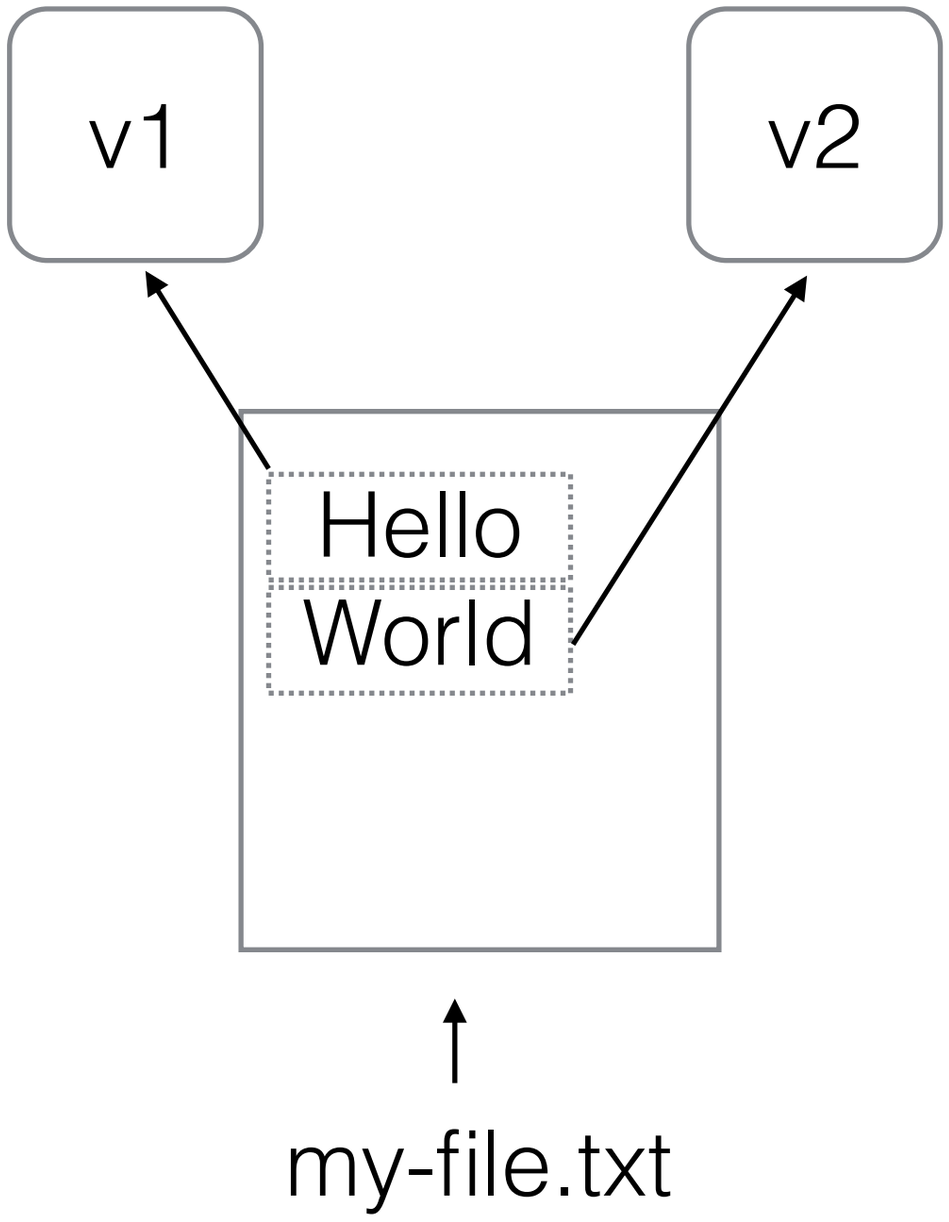
Not



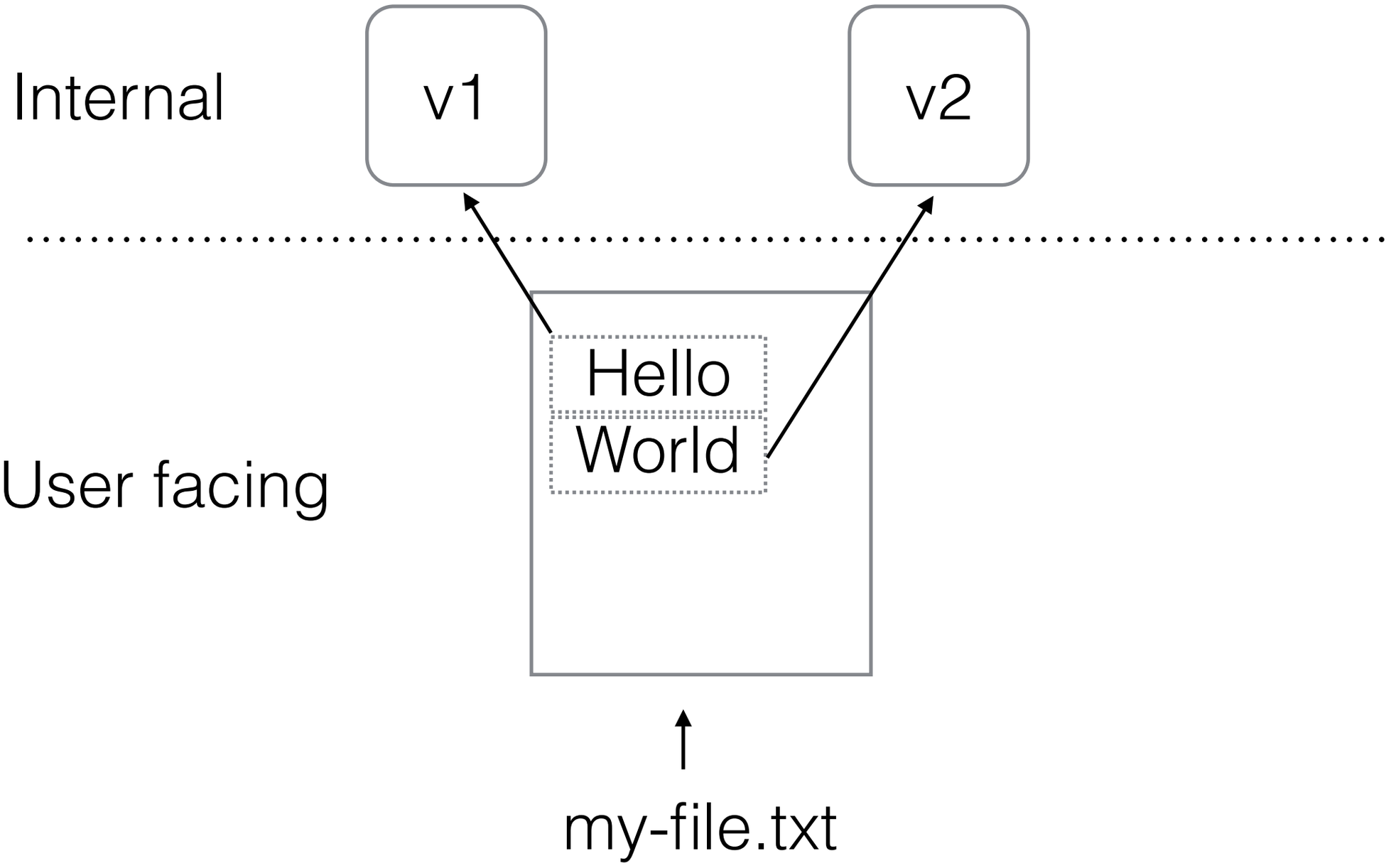
Rather

my-file.txt

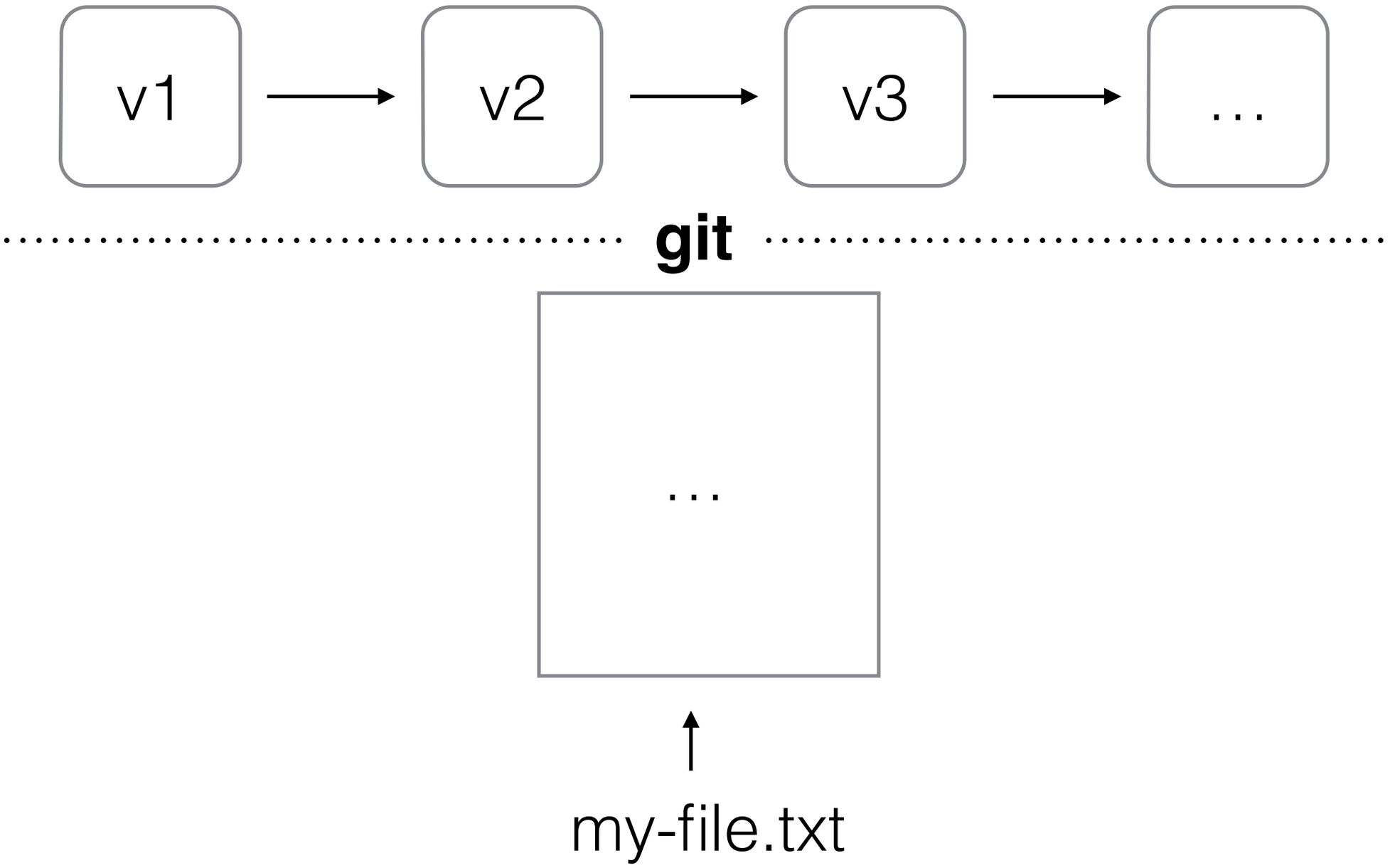
Rather



Rather



Rather



git

- https://git-scm.com/
- initially designed and developed to manage source code for the Linux operating system in 2005
- git is a program, or in more contemporary terms an app
- it manages directories of files - these are called repositories or repos

data

metadata

hash
author
date

git

README.md

worker.py

docs.html

commit 1

data

metadata

hash
author
date

git

README.md

worker.py

docs.html

commit 1

commit 2

data

metadata

hash
author
date

data

metadata

hash
author
date,
relation

git

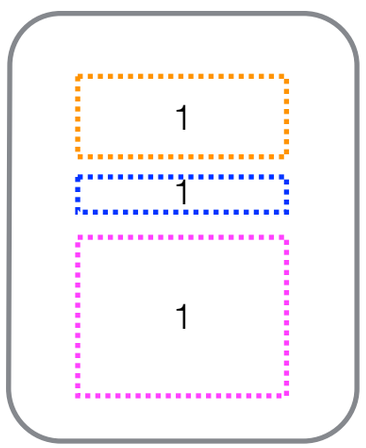
README.md

worker.py

docs.html

commit 1

data

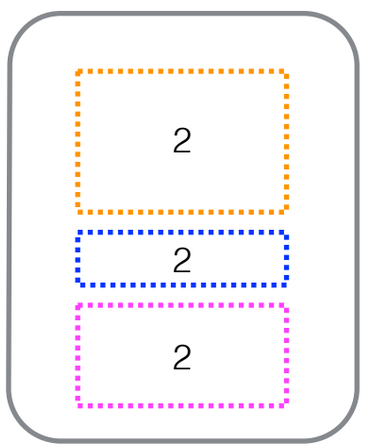


metadata

hash
author
date

commit 2

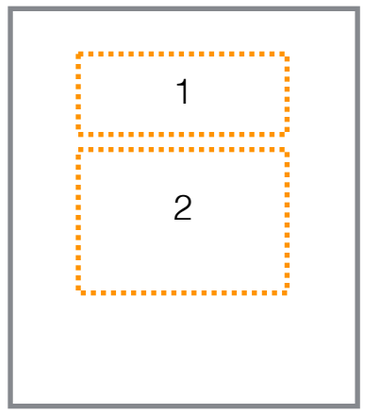
data



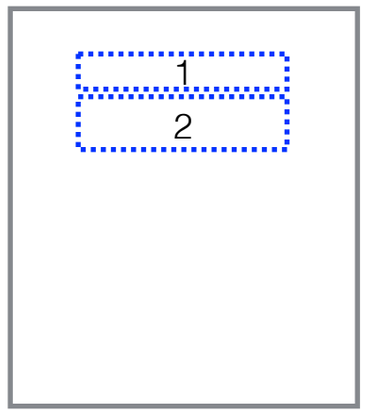
metadata

hash
author
date,
relation

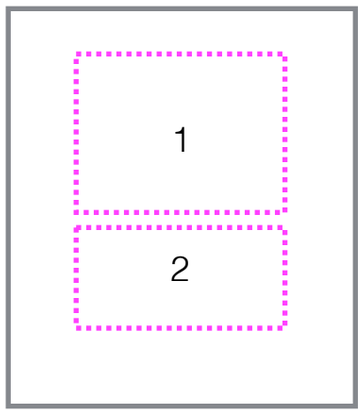
git



README.md



worker.py

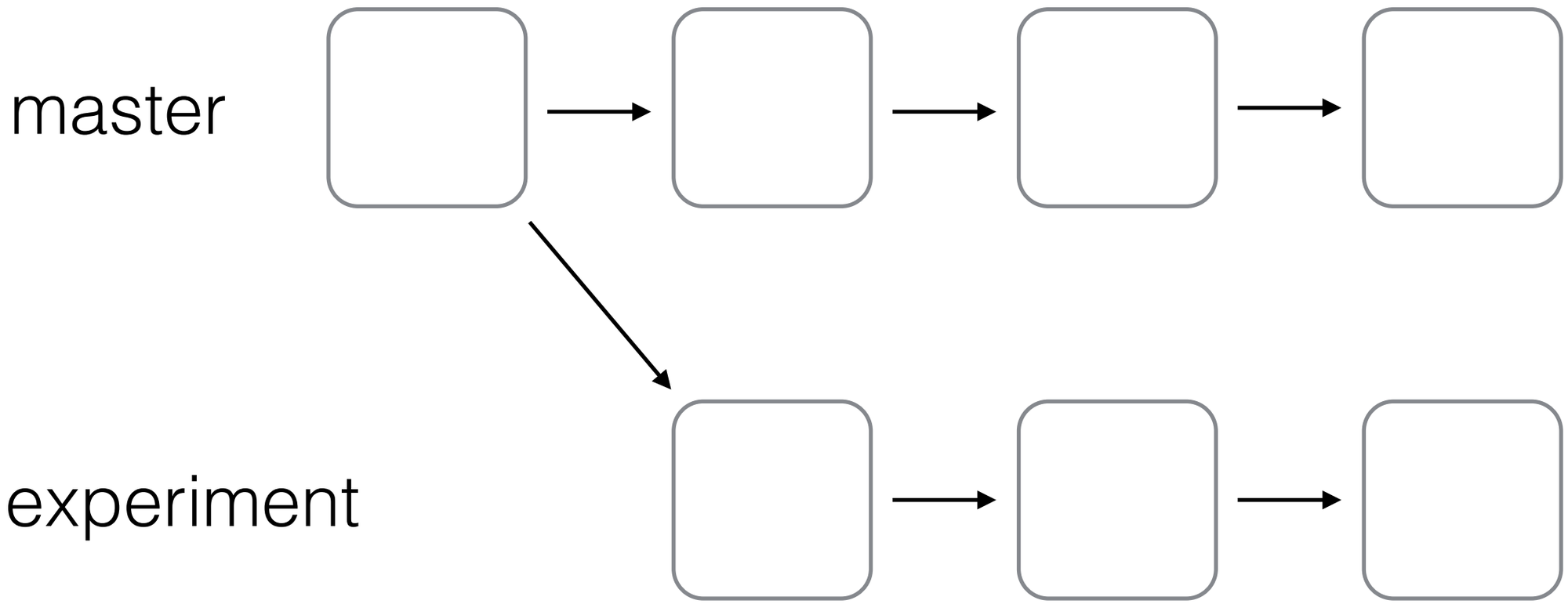


docs.html

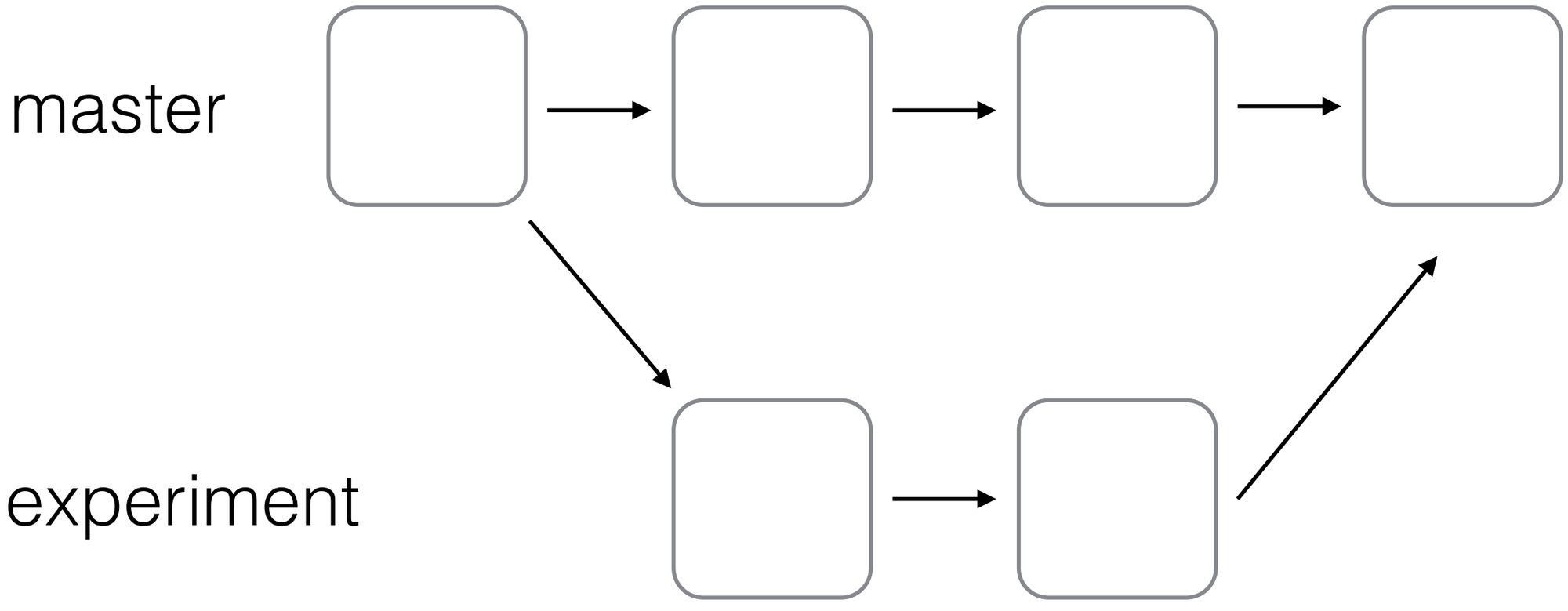
git

- so changes to a repo are recorded as commits
- commits contain data and metadata, like author, date+time, relation to other commits
- commits always happen on a branch
- the default branch is called ‘master’ - it is just a name

branches



merge branches



collaboration?

- share repos with each other
- not through email or dropbox :)
- set up a remote on a server
- collaborators sync their copies with the remote one, agree on authoritative changes when conflicts occur

remotes

- a popular option: https://github.com
- another option: https://about.gitlab.com/
- there are many more

the commit

- each commit also has a commit message
- describe what you have done - important project metadata
- a good commit message introduces the reader to the nature of the changes that were applied
- normally it has a short heading and an extended description if relevant
- all commit messages are collected in a commit log

github

1. signup
2. create a repo
3. create a file, add changes, commit it
4. add more changes, commit it
5. view the difference, comment on them
6. read the commit log