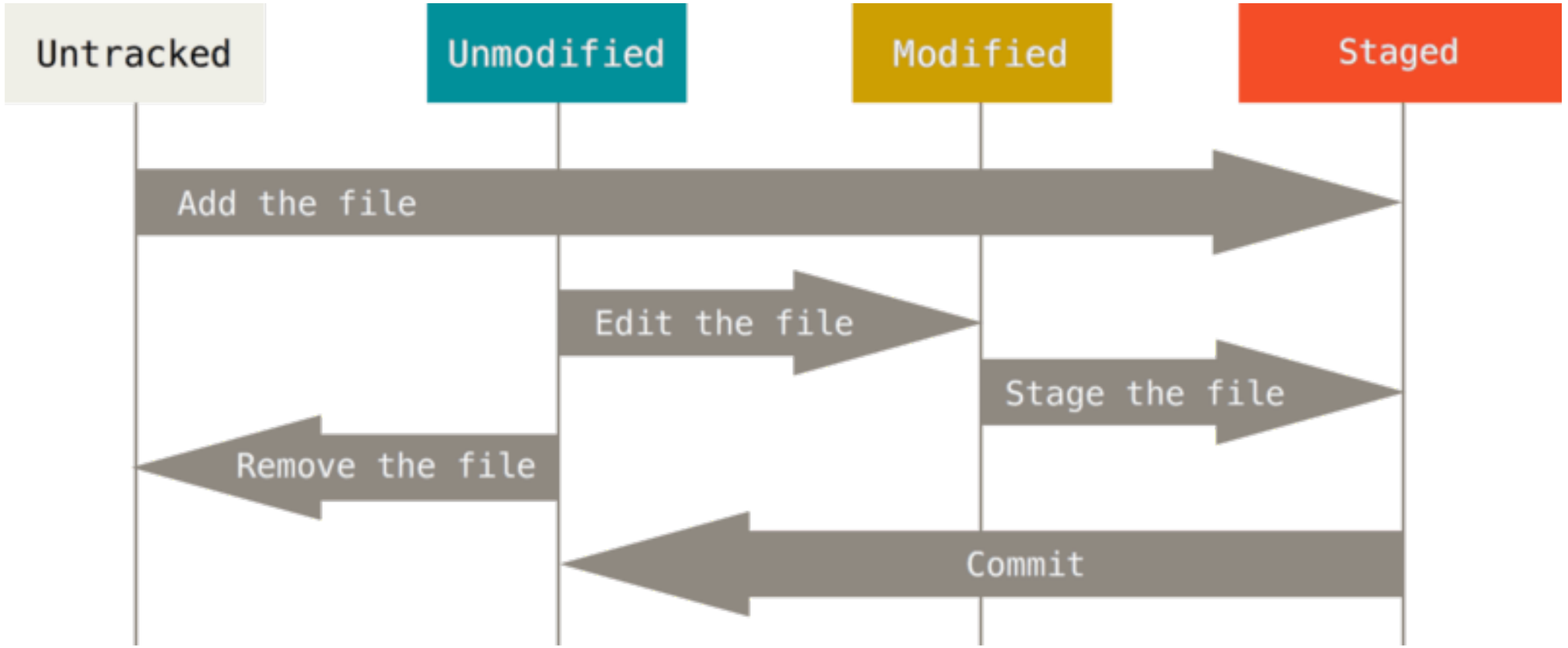
GENERAL ASSEMBLY

GIT II:

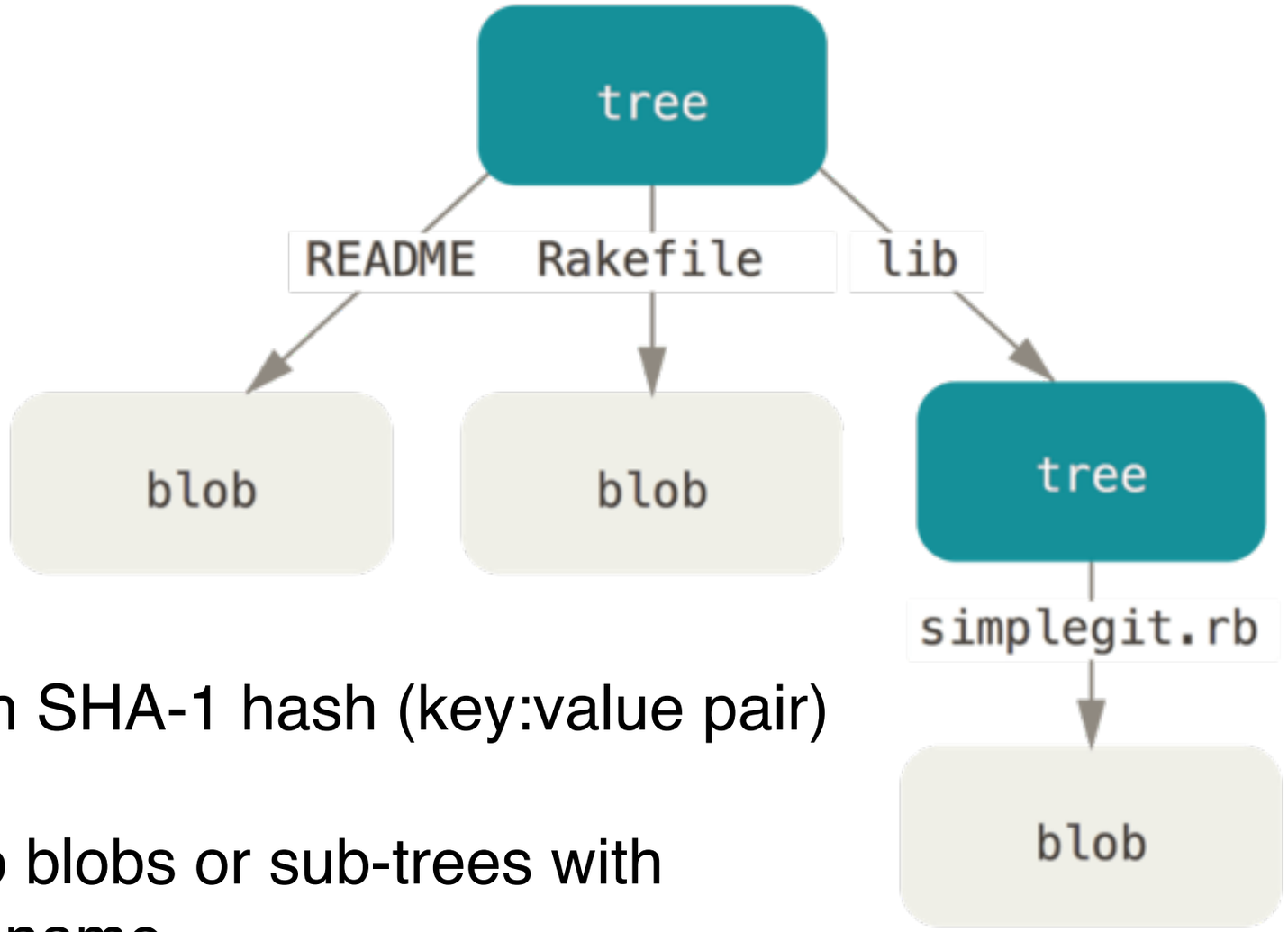
REVENGE OF GIT

- Git conceptual overview
- Our workflow
- Fetching + merging = pulling
- Where did my files go?
- Merge conflicts
- Further reference

GIT CONCEPTUAL OVERVIEW



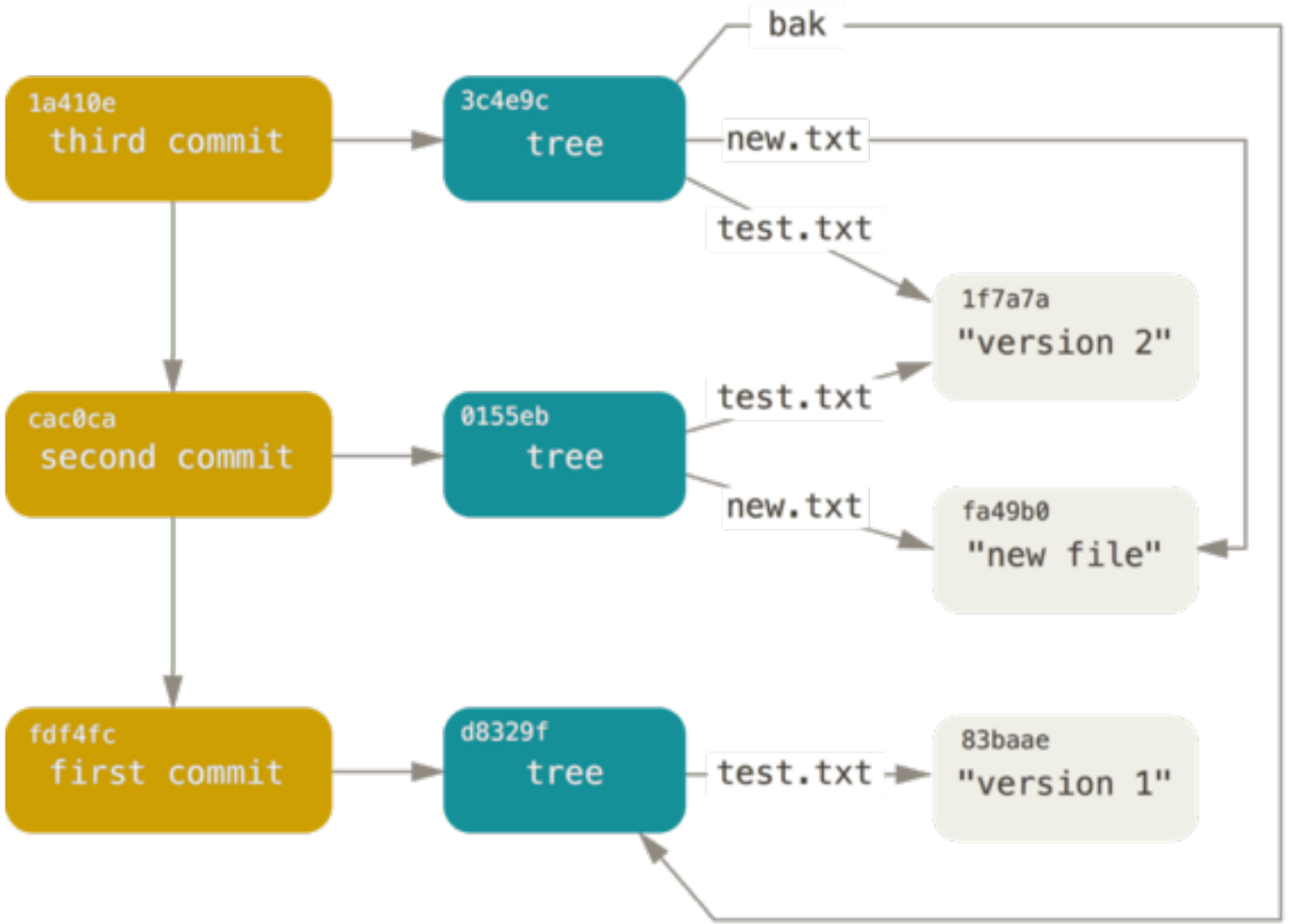
Source: https://git-scm.com/book/en/v2/



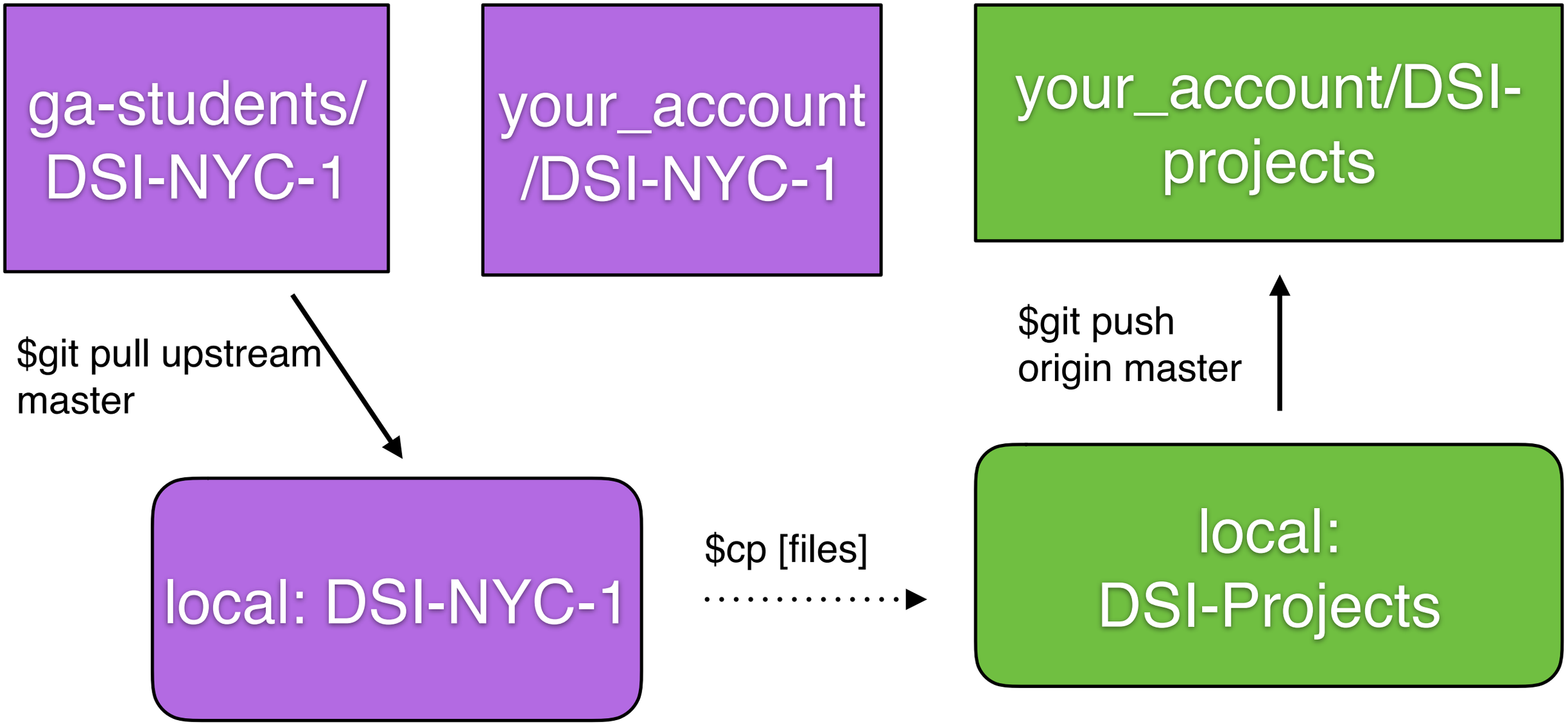
blob - file content named with SHA-1 hash (key:value pair)

tree - ‘directories’, pointing to blobs or sub-trees with associated data including filename

commit - points to top-level tree; includes commit message and authorship

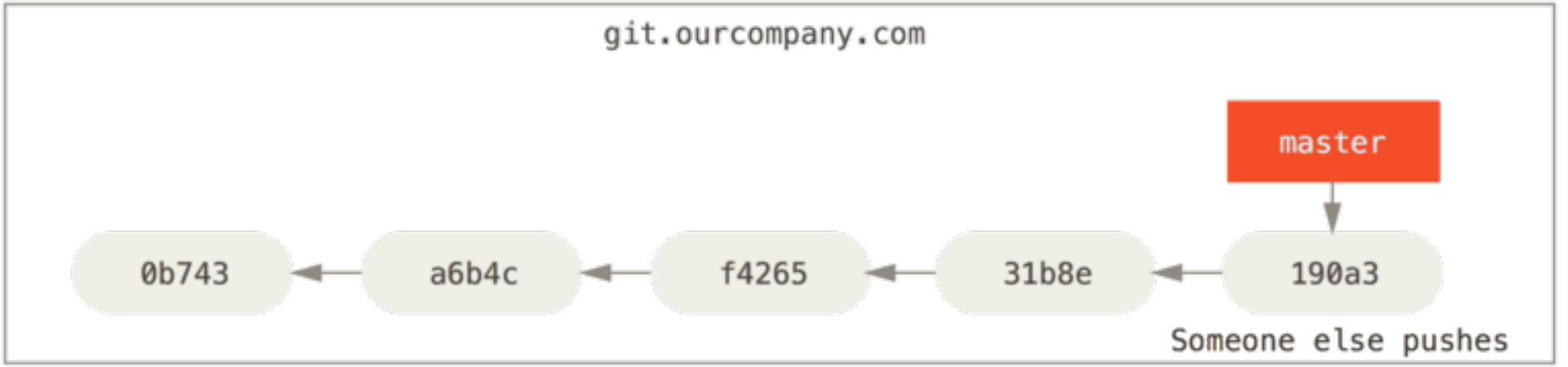


OUR WORKFLOW

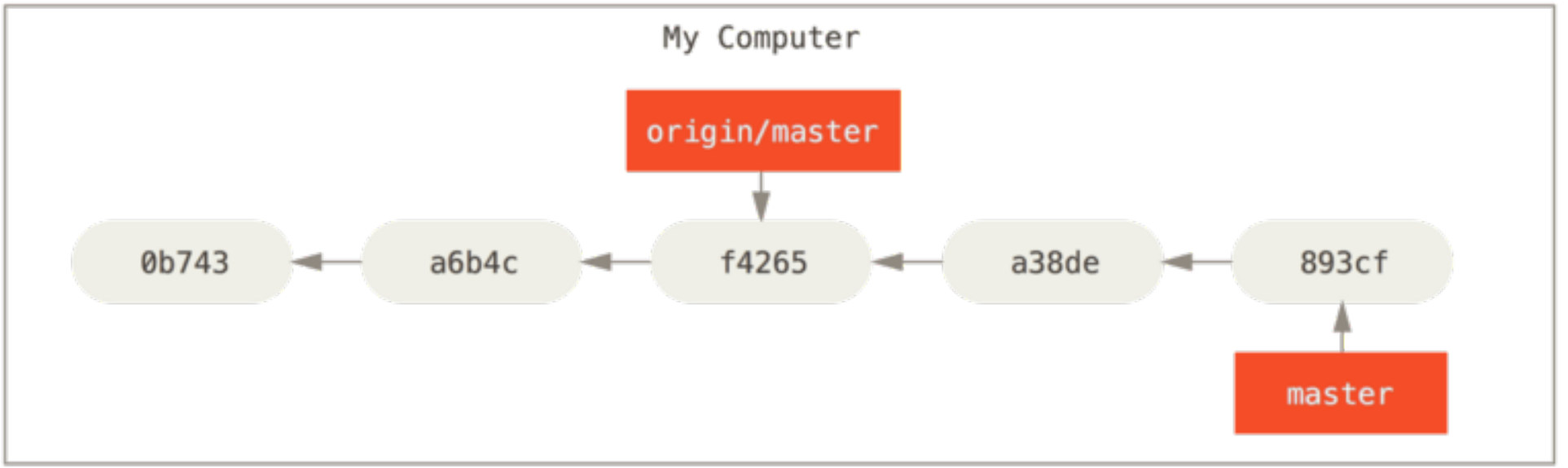


FETCHING + MERGING = PULLING

branch - pointer to a commit (and, by extension, its ancestor commits)

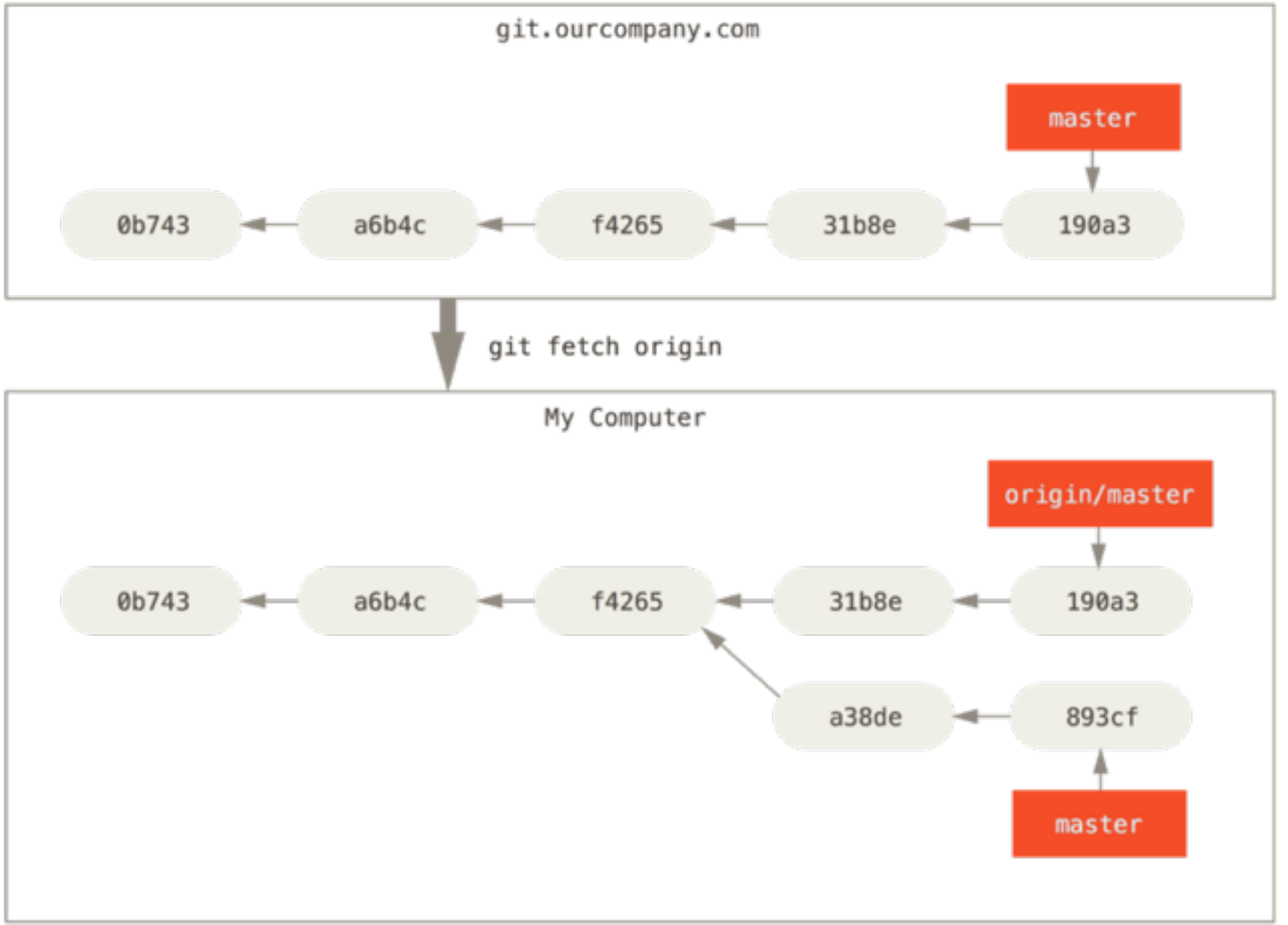


Branches can diverge in their project history.



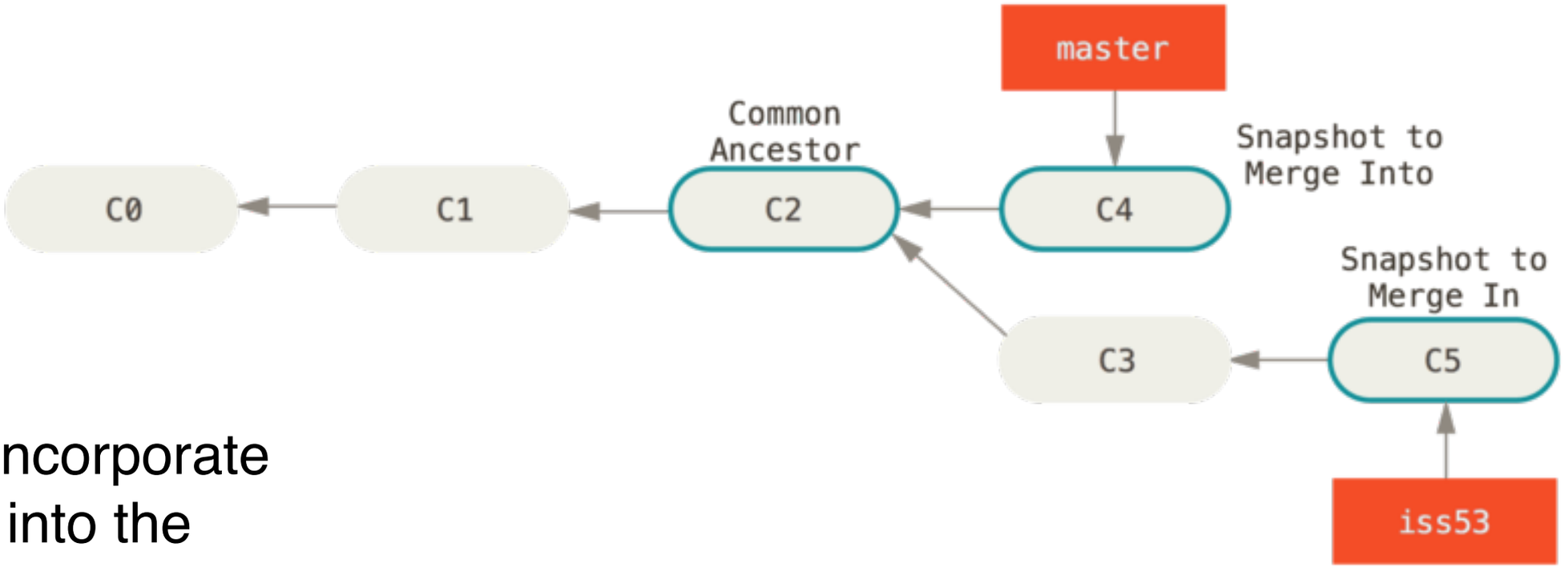
FETCHING + MERGING = PULLING

fetch - download or update the branch(es) from remote repository



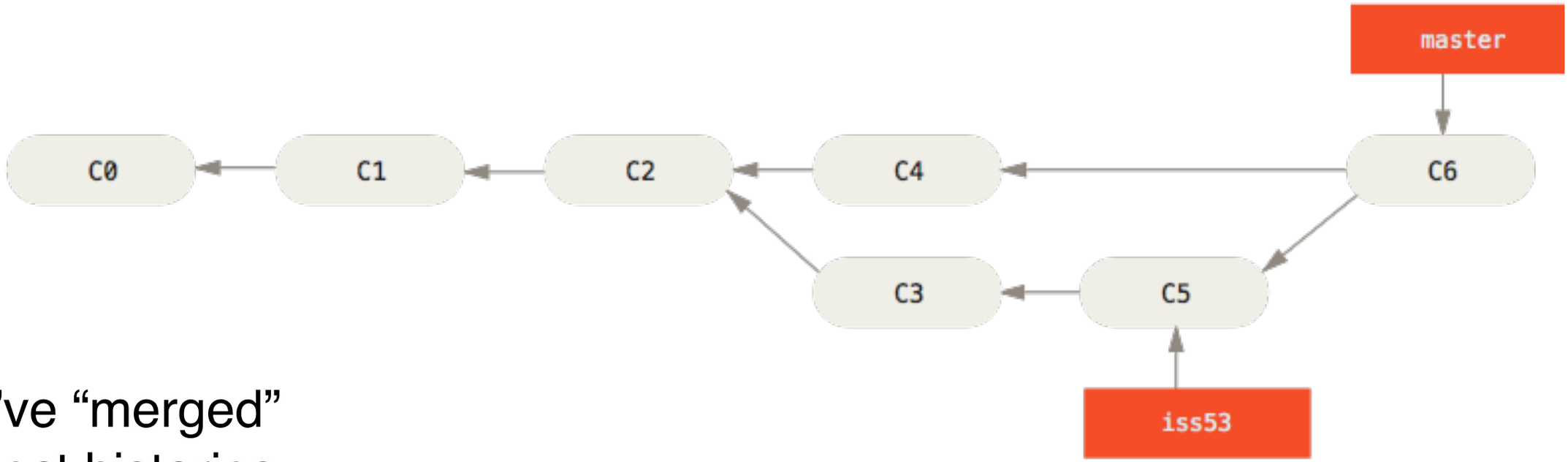
FETCHING + MERGING = PULLING

NB: new example



merge - incorporate changes into the current branch

FETCHING + MERGING = PULLING



We’ve “merged”
project histories
(commits), NOT files.

WHERE DID MY FILES GO? 11

$git pull upstream master - this updates your project history.

If part of that history is “Add Week 3, Day 2 lessons” (and that commit’s associated files), and you don’t have any history related to that commit, then there is no conflict: it is merged.

WHERE DID MY FILES GO? 12

If part of your history is “Delete Week 3, Day 1 starter code” (and that commit’s associated file deletions), and the upstream branch doesn’t have any subsequent history related to that commit, then there is no conflict: it is merged. (I.e. you have a merged version which INCLUDES those committed deletions.)

(Source: http://stackoverflow.com/questions/9591407/unstage-a-deleted-file-in-git)

“# this restores the file status in the index
git reset -- <file>
then check out a copy from the index
git checkout -- <file>
To undo 'git add ', the first line above suffices, assuming you haven't committed yet.”

WHERE DID MY FILES GO? 13

(Source: https://www.quora.com/How-can-I-recover-a-file-I-deleted-in-my-local-repo-from-the-remote-repo-in-Git)

“If the deletion has not been committed, the command below will restore the deleted file in the working tree.

$ git checkout -- <file>

You can get a list of all the deleted files in the working tree using the command below.

$ git ls-files --deleted

If the deletion has been committed, find the commit where it happened, then recover the file from this commit.

$ git rev-list -n 1 HEAD -- <file>

$ git checkout <commit>^ -- <file>”

UH OH.

Try “reset” on some of your commits. Or seek guidance online:

http://gitready.com/beginner/2009/01/25/branching-and-merging.html

We’ll address merging in some detail in Week 9.

FURTHER REFERENCE

- pro git: https://git-scm.com/book/en/v2
- git tower: https://www.git-tower.com/learn/git/ebook/
  - cheatsheet: https://www.git-tower.com/blog/git-cheat-sheet/