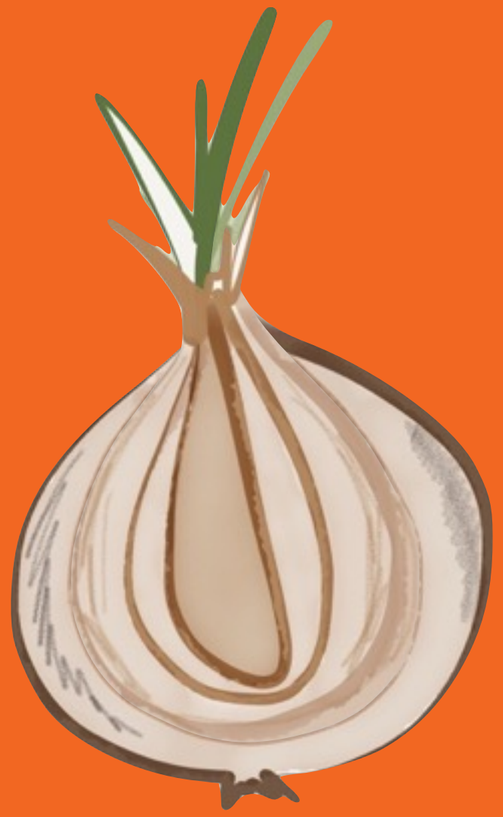
Distributed Version Control

Paolo Perrotta

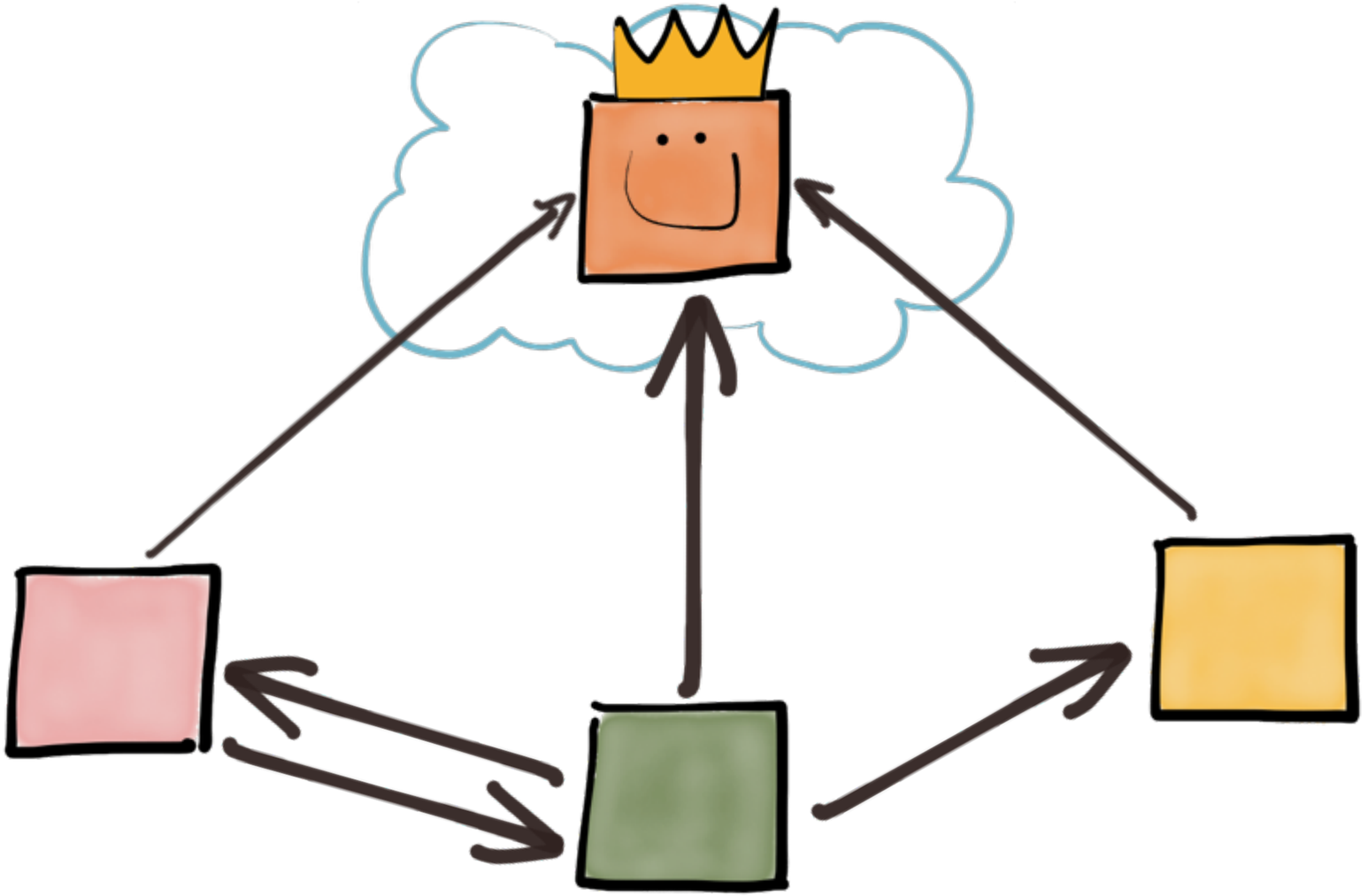
@nusco

Git Is..

…a Distributed Revision Control System



Multiple Repos

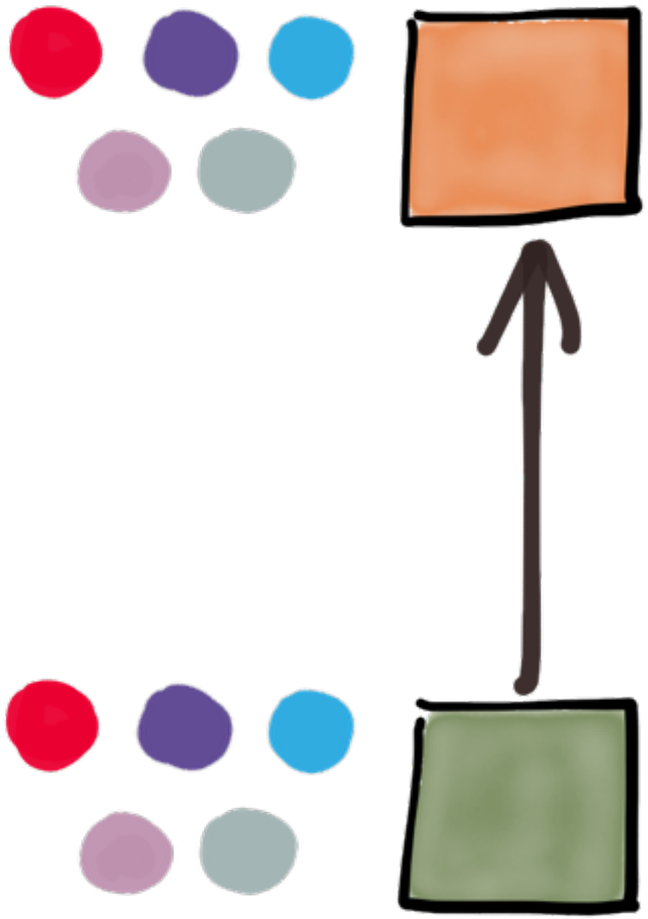


Like a local branch, a remote branch
is just a reference to a commit.

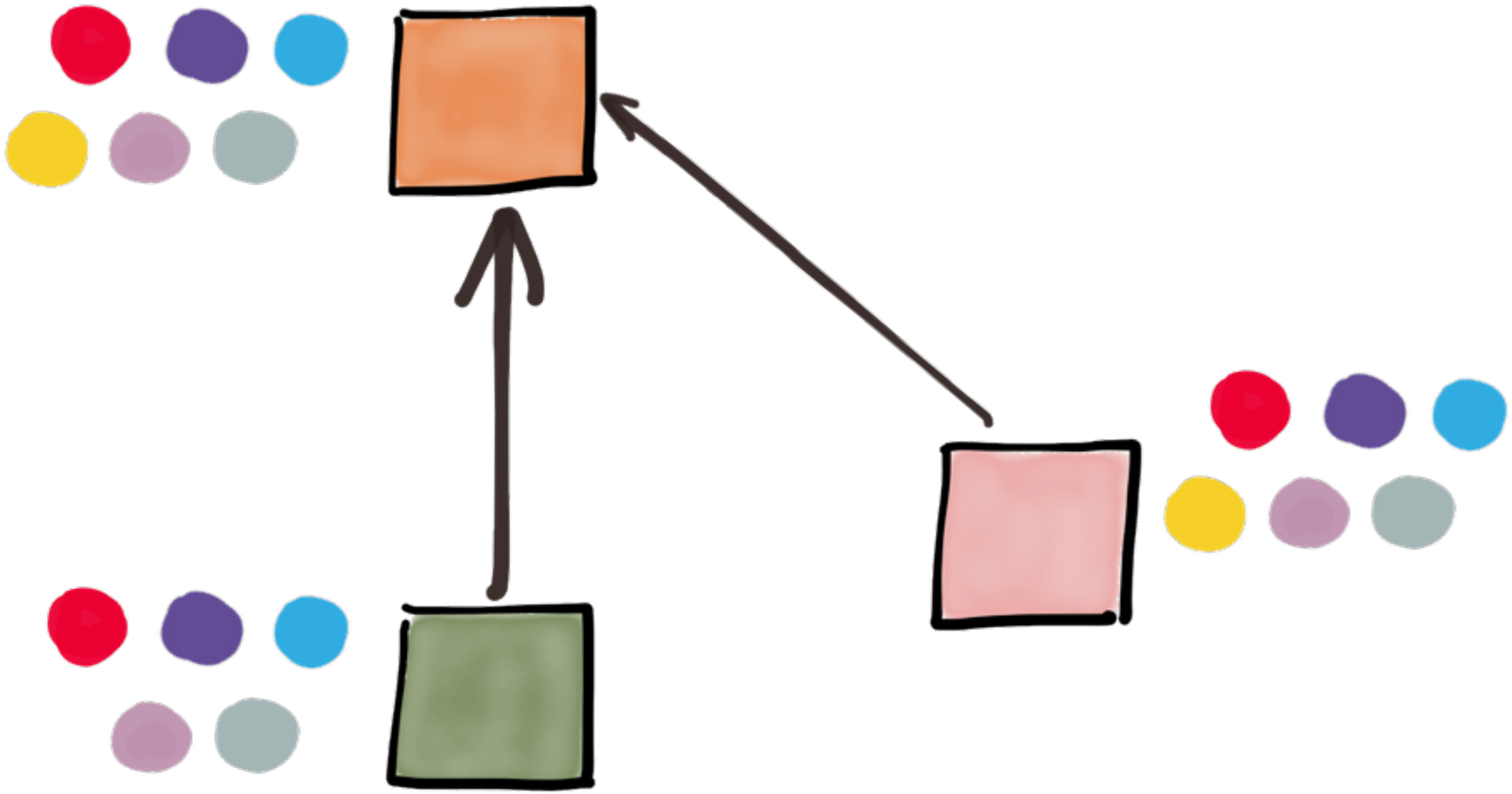
“Apple Pie”

23991897e13e47ed0adb91a0082c31c82fe0cbe5

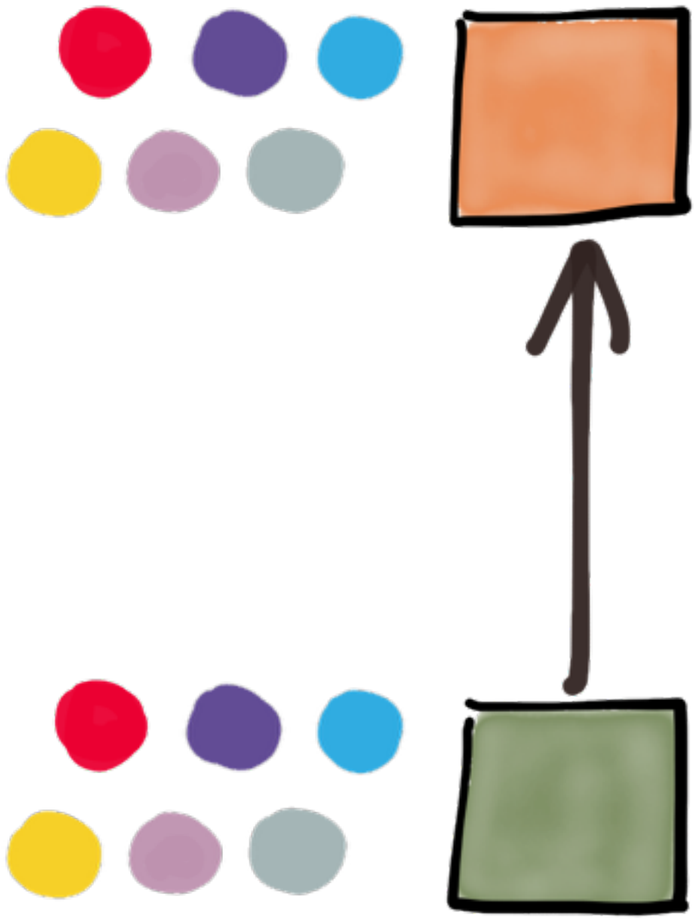
Synchronizing Repos



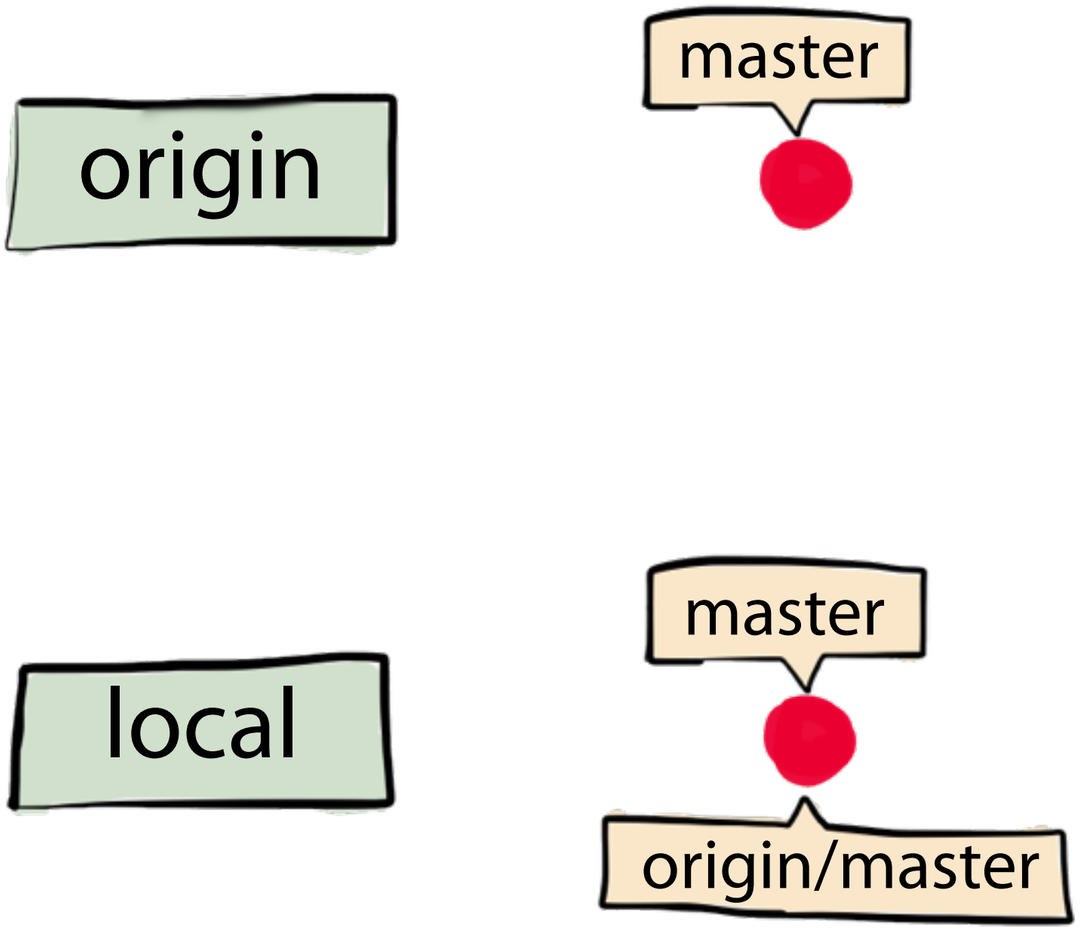
Synchronizing Repos



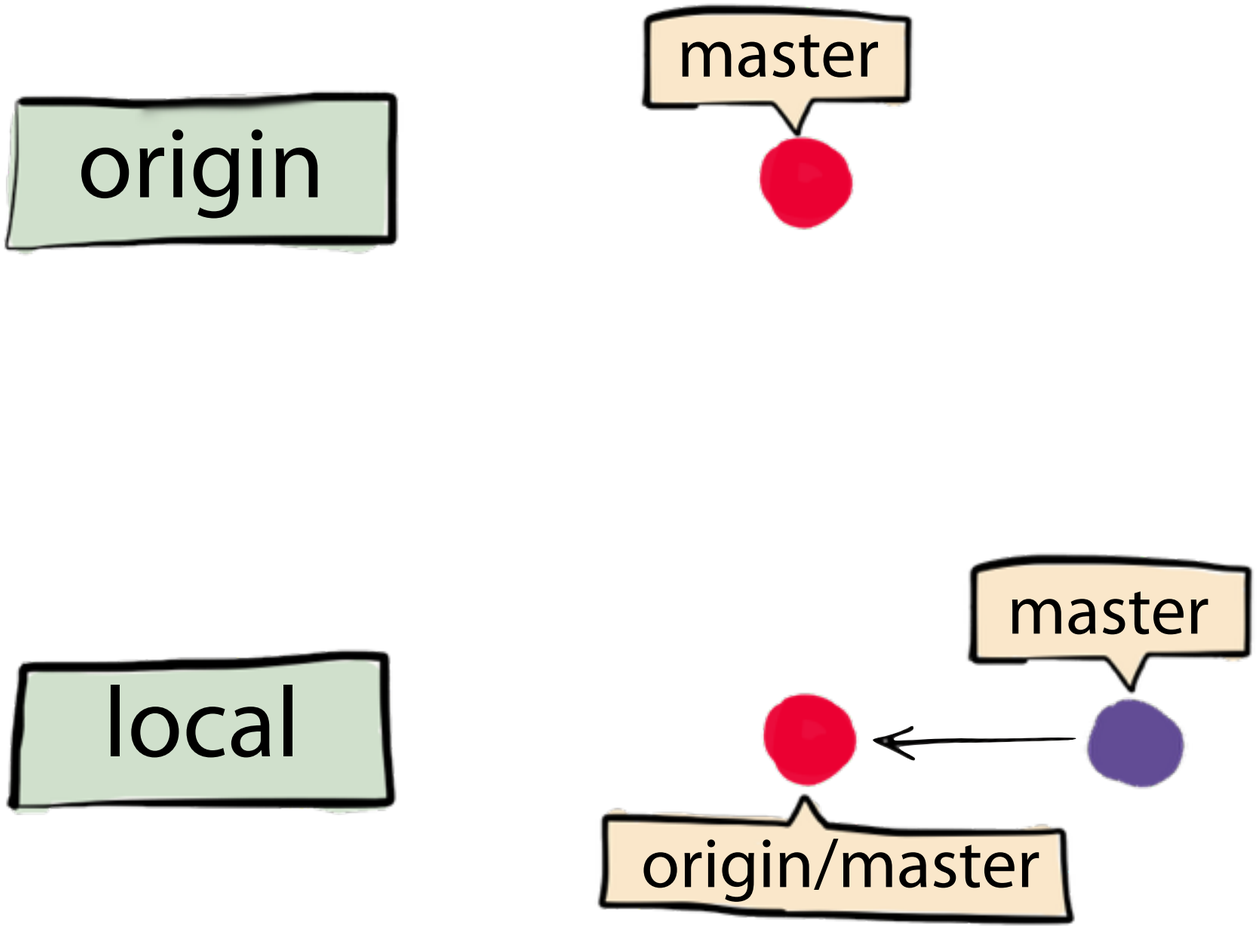
Synchronizing Repos



Synchronizing with a Remote

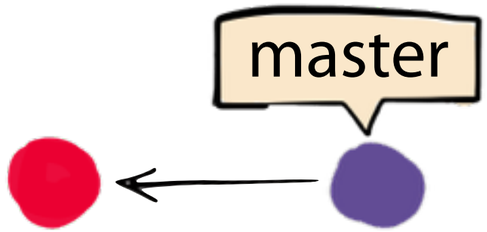


Adding a Commit

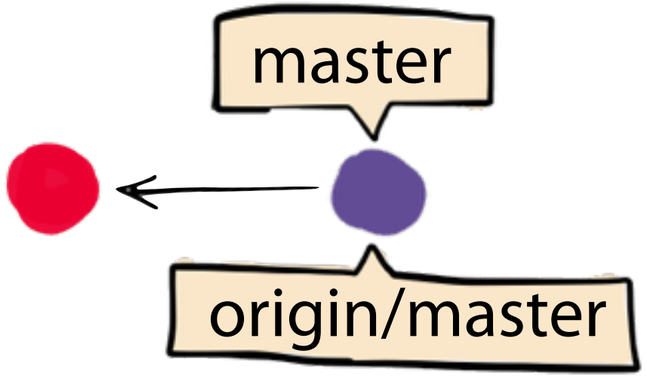


git push

origin

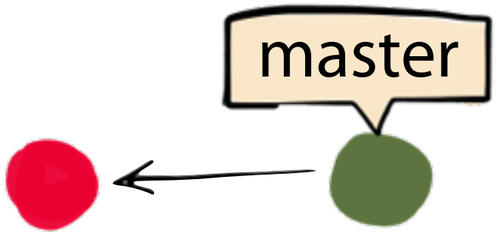


local

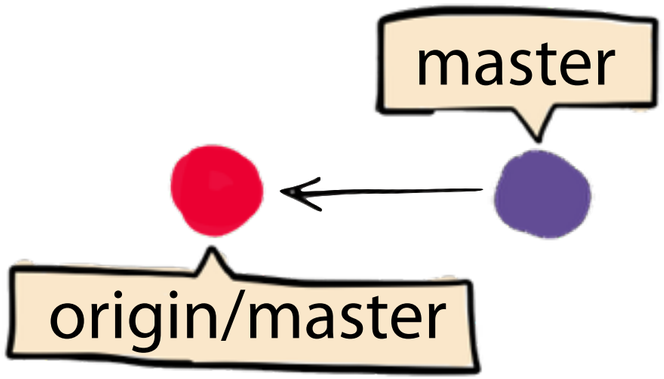


Conflict!

origin

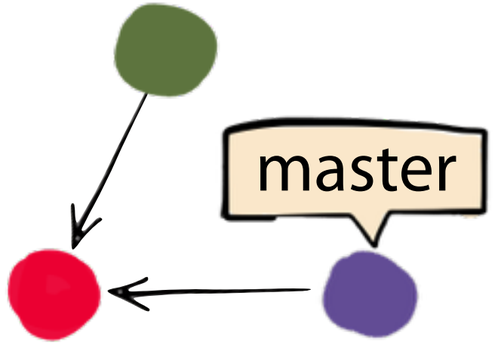


local

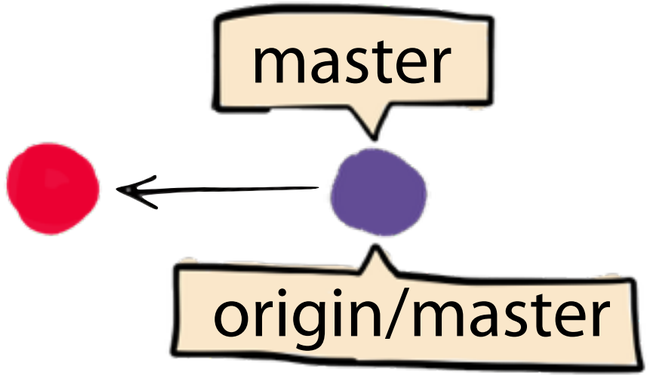


git push -f

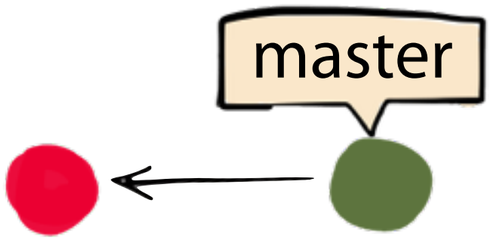
origin



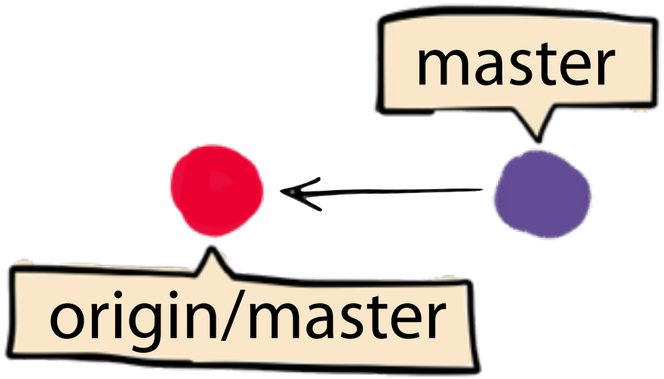
local



origin

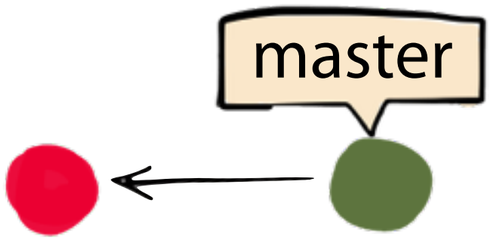


local

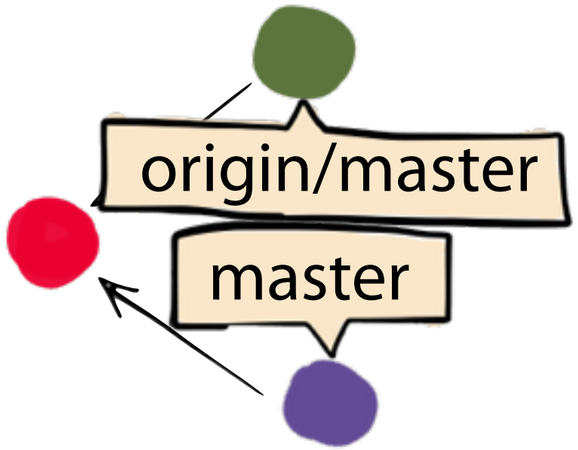


git fetch

origin

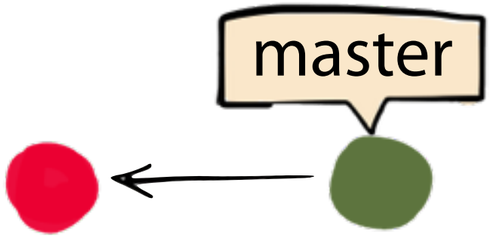


local

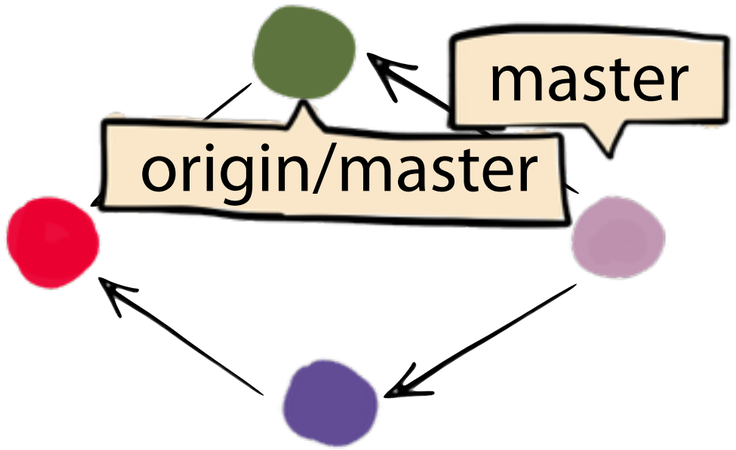


git merge origin/master

origin

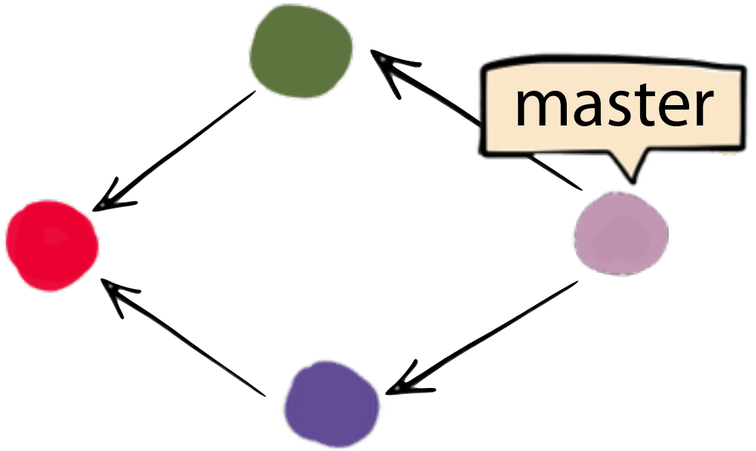


local

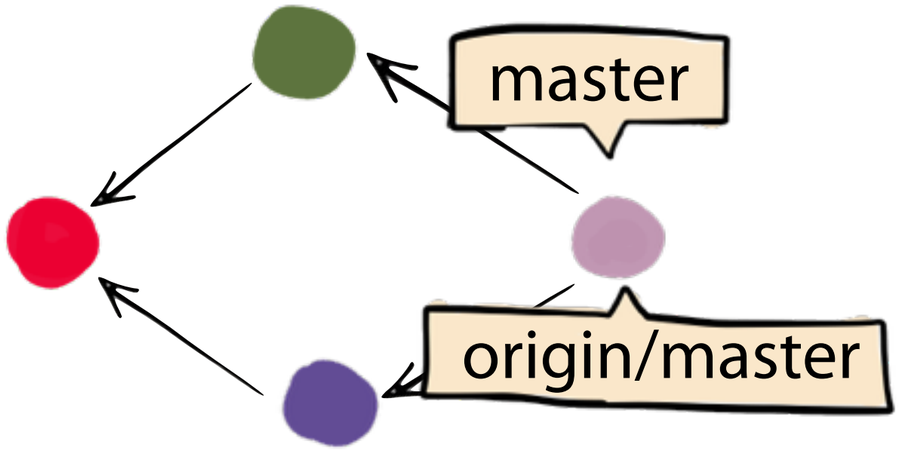


git push

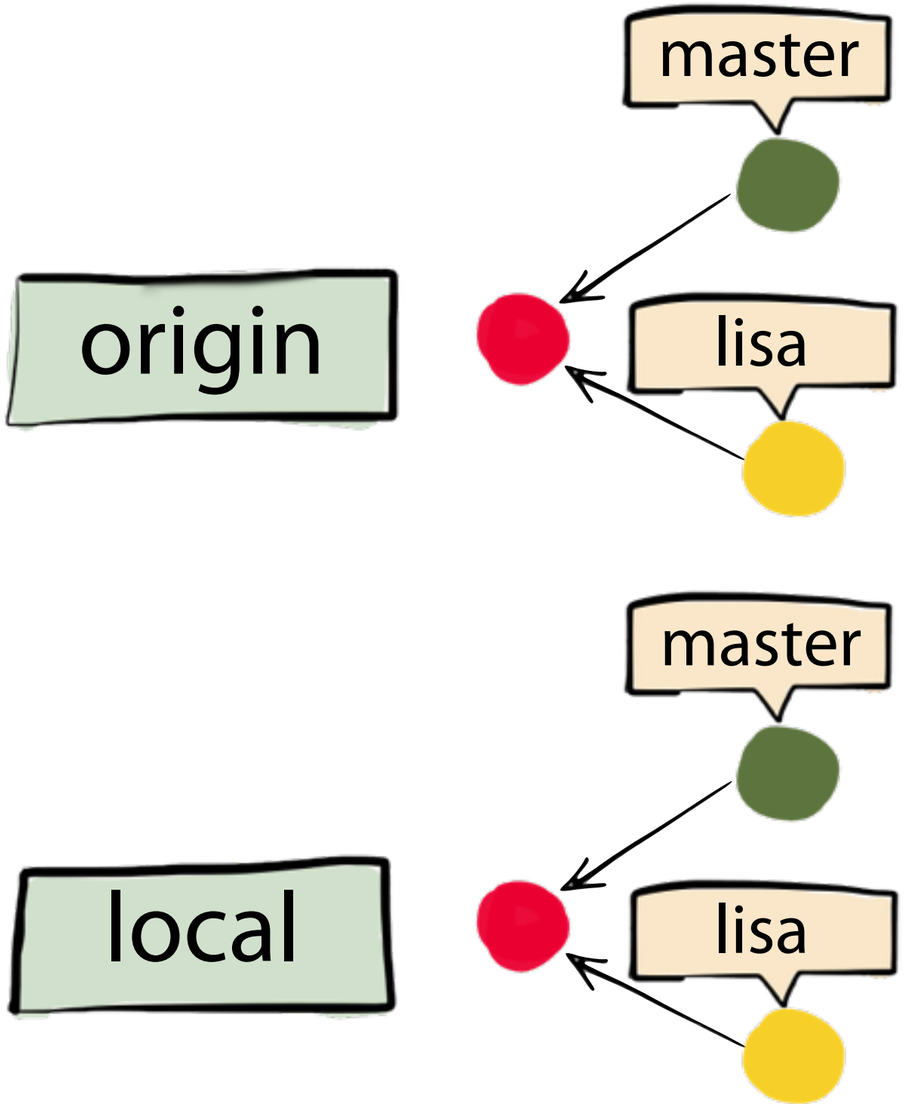
origin



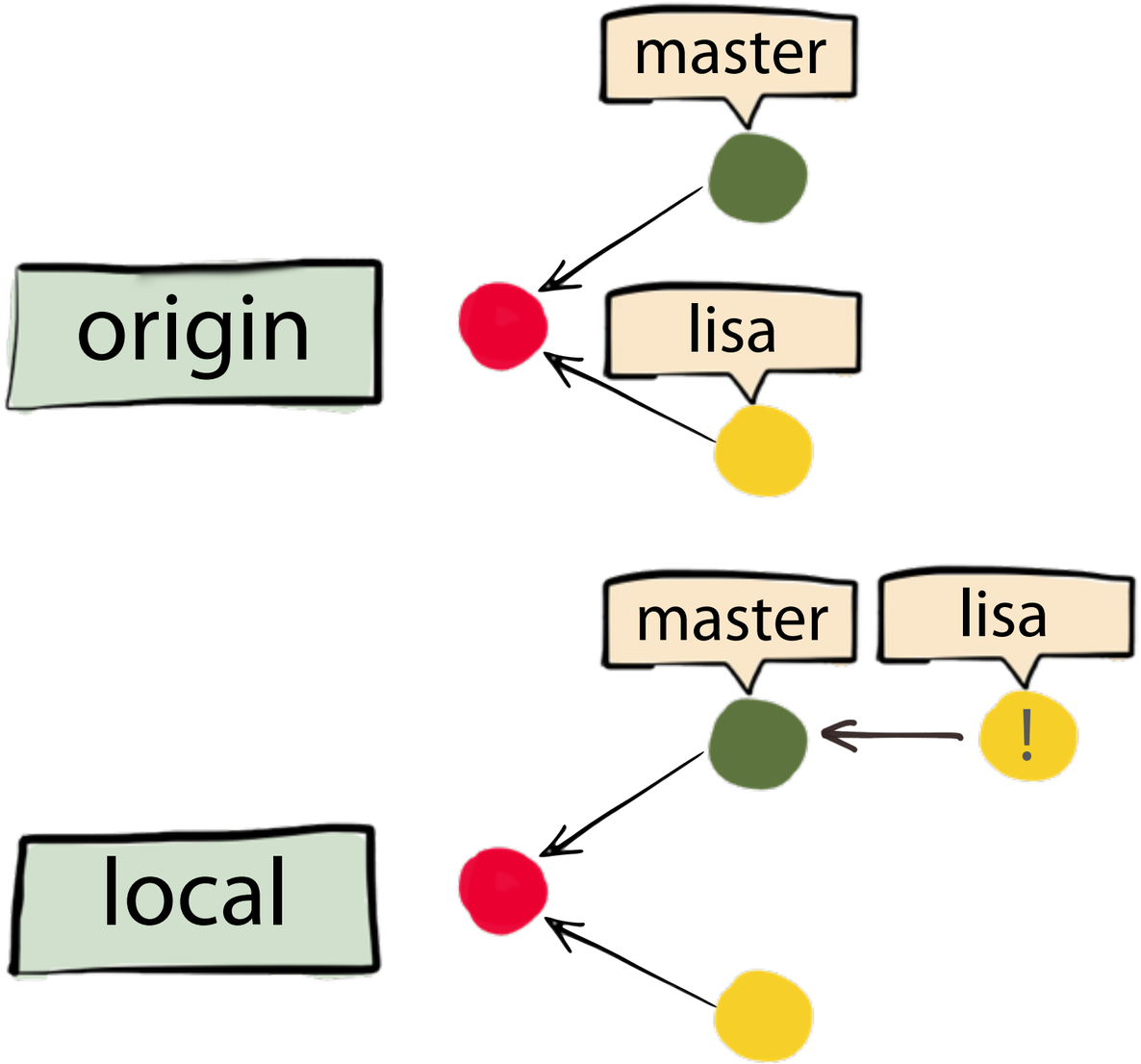
local



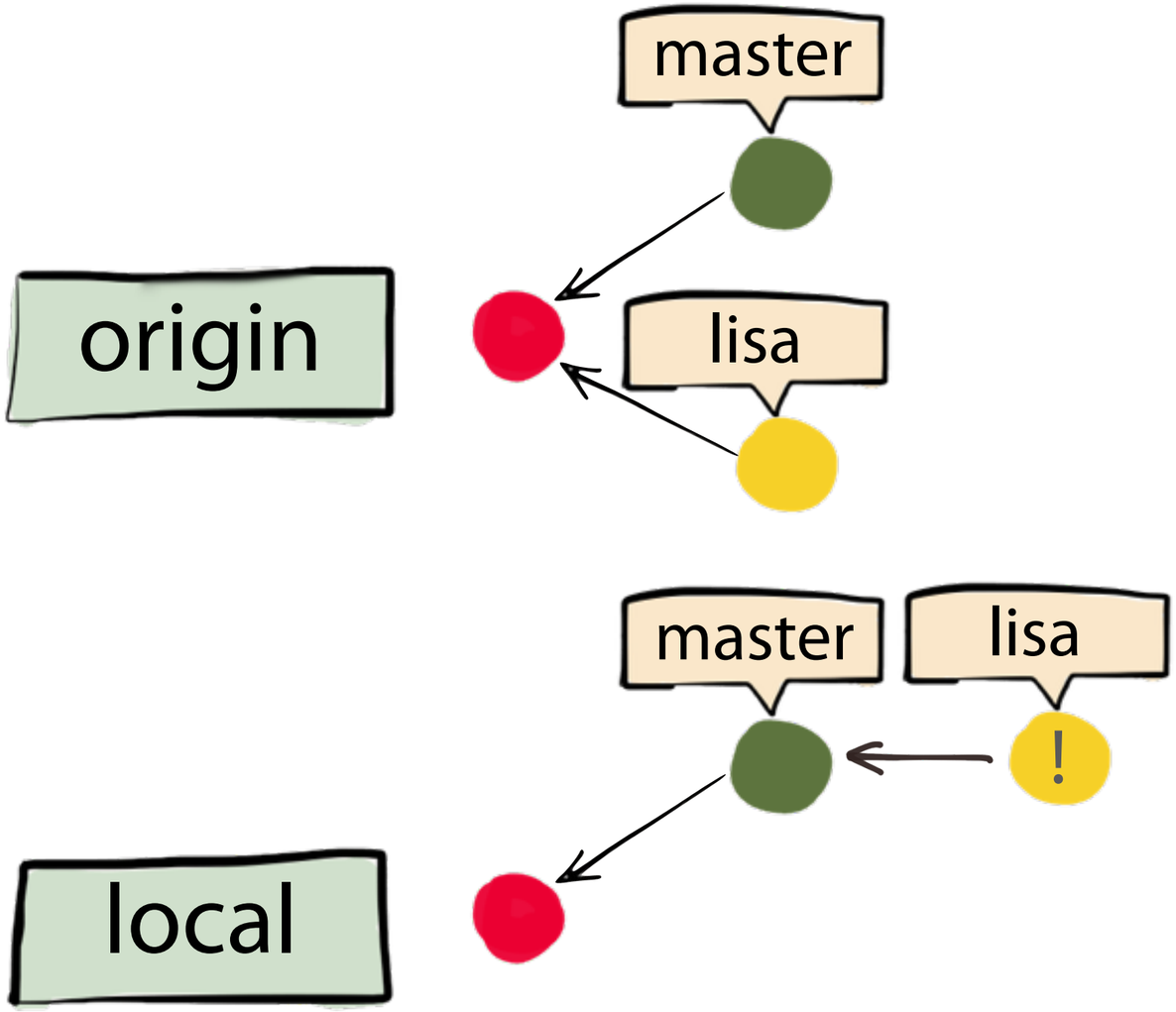
You “pull”, then you “push”.


master
origin
lisa
master
local
lisa

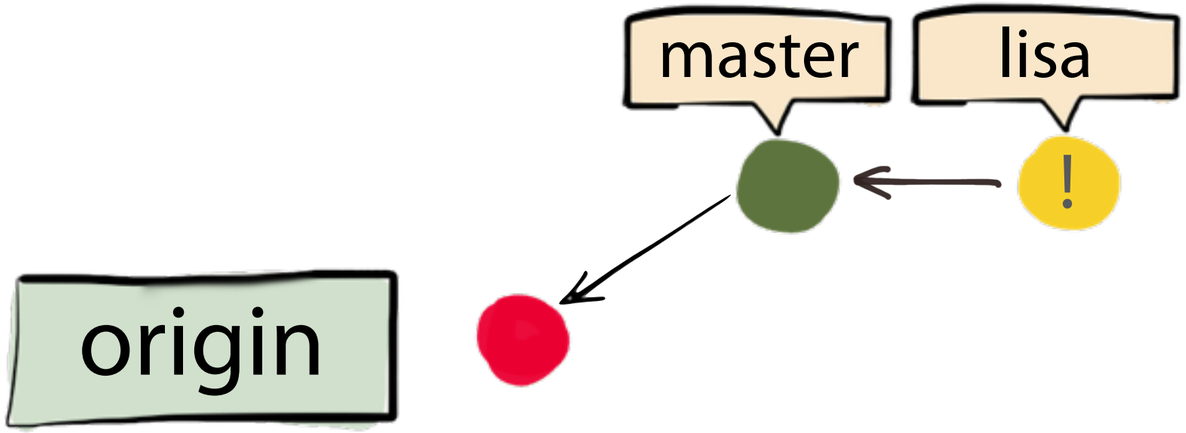
git rebase master

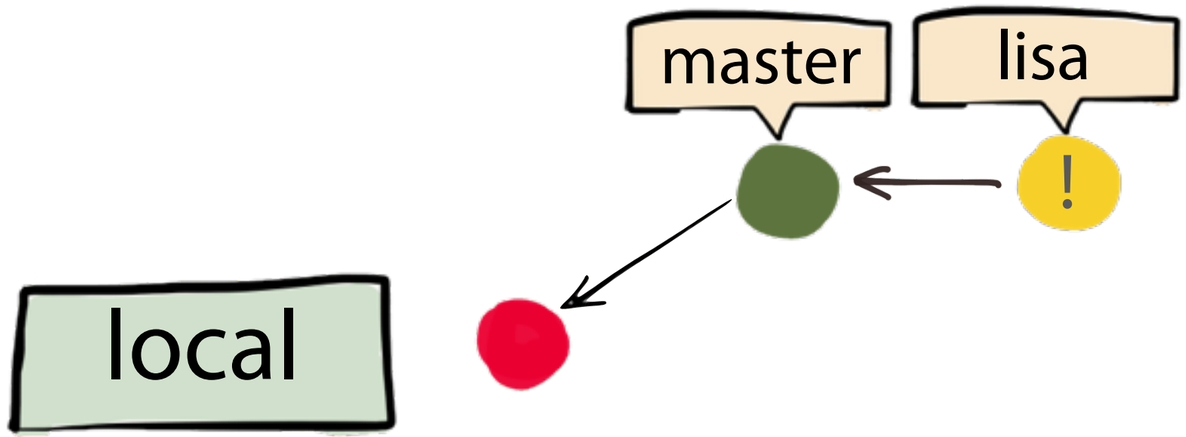


Conflict!

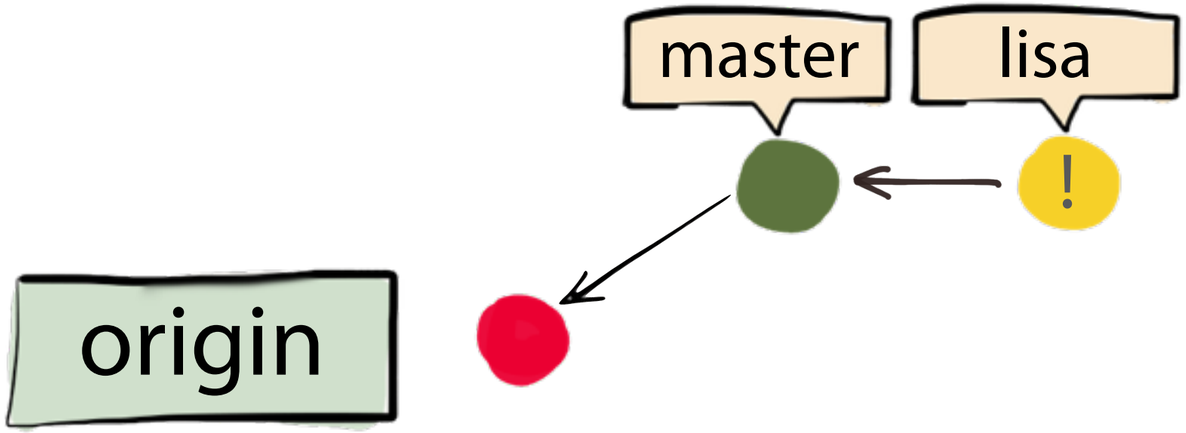


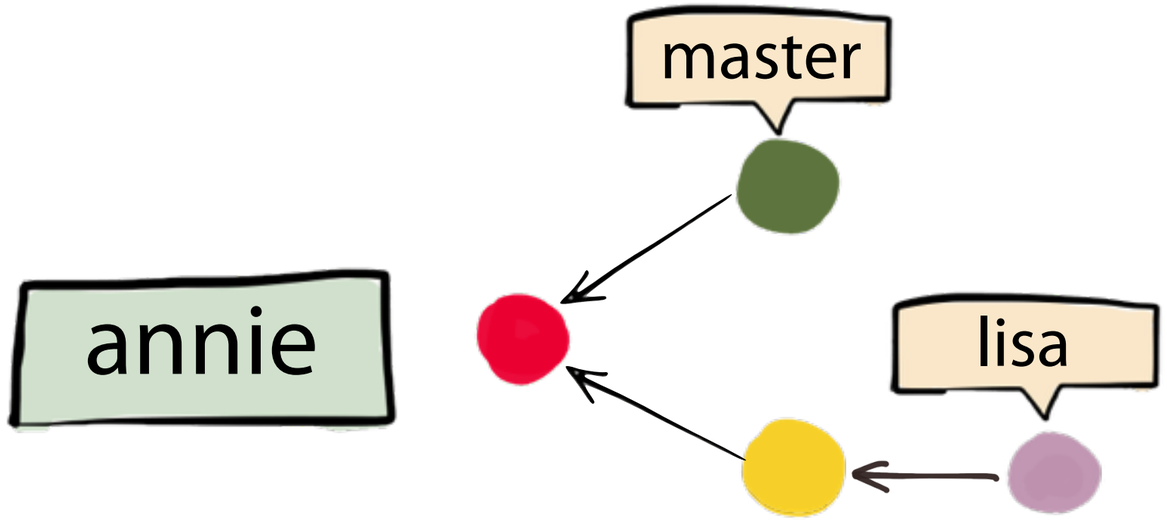
Solve the Conflict





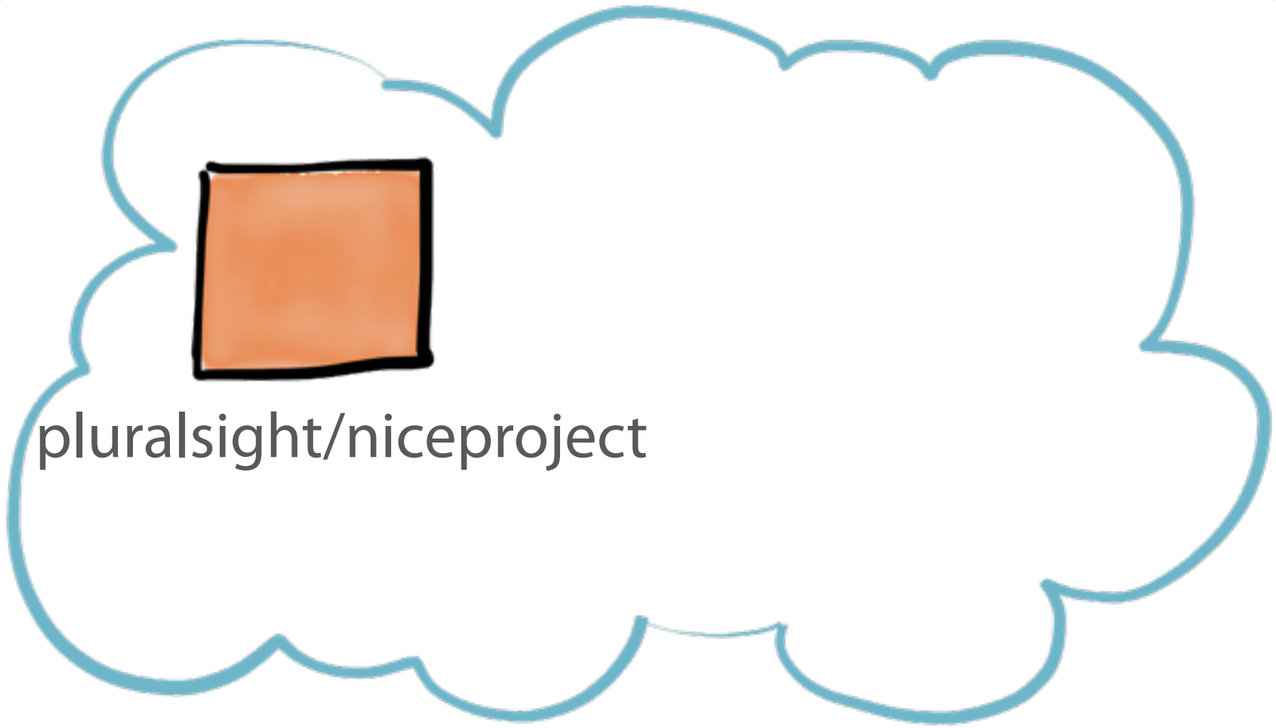
Another User



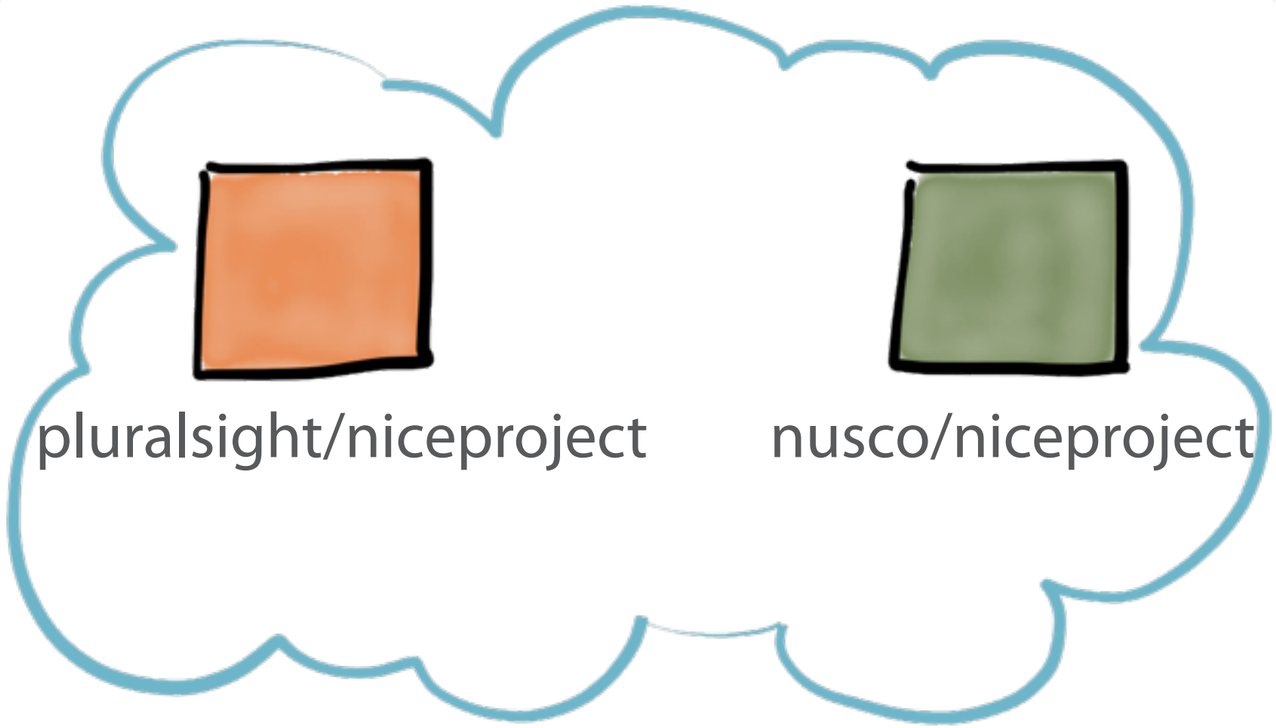


Never rebase shared commits.

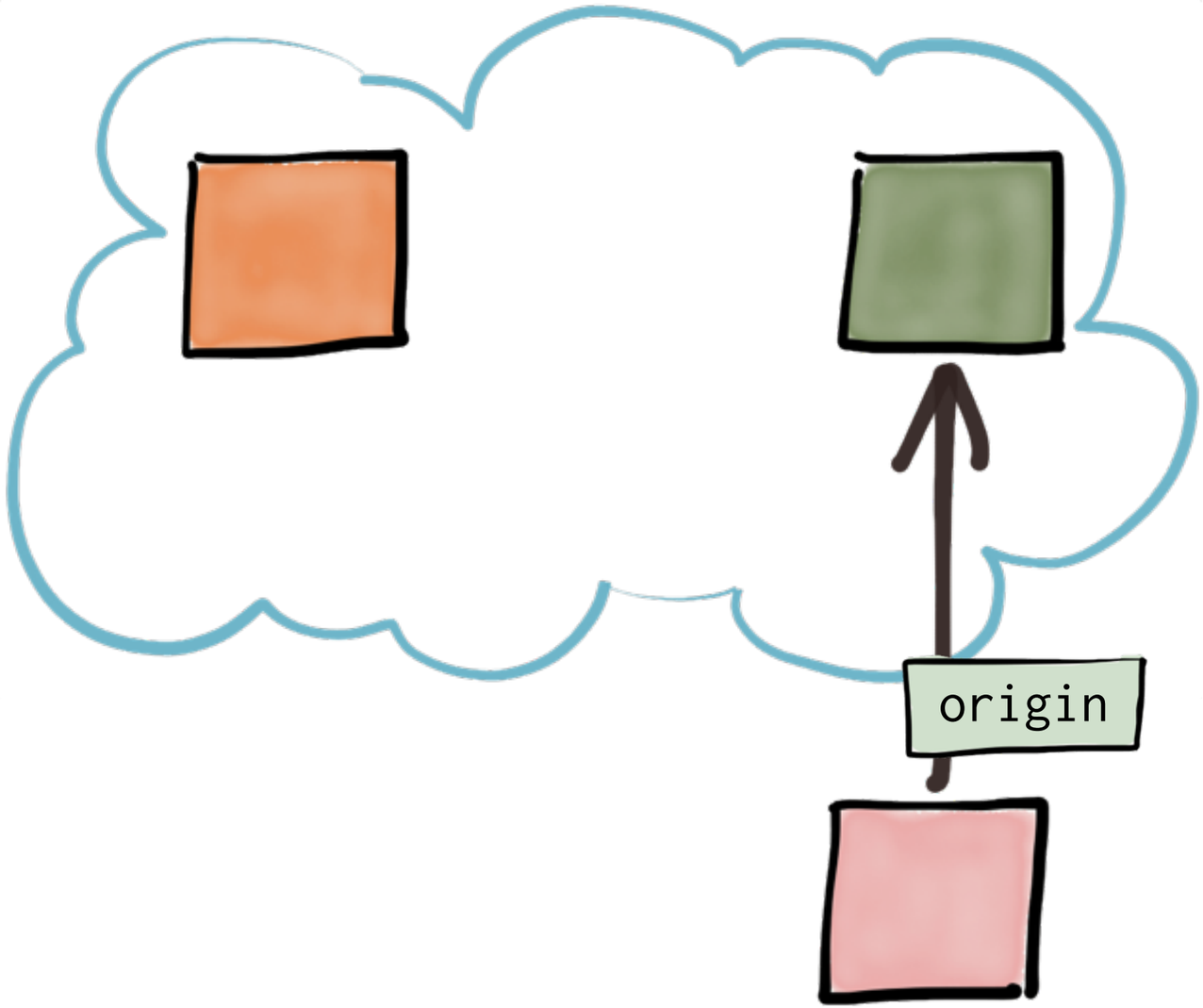
GitHub Features


pluralsight/niceproject

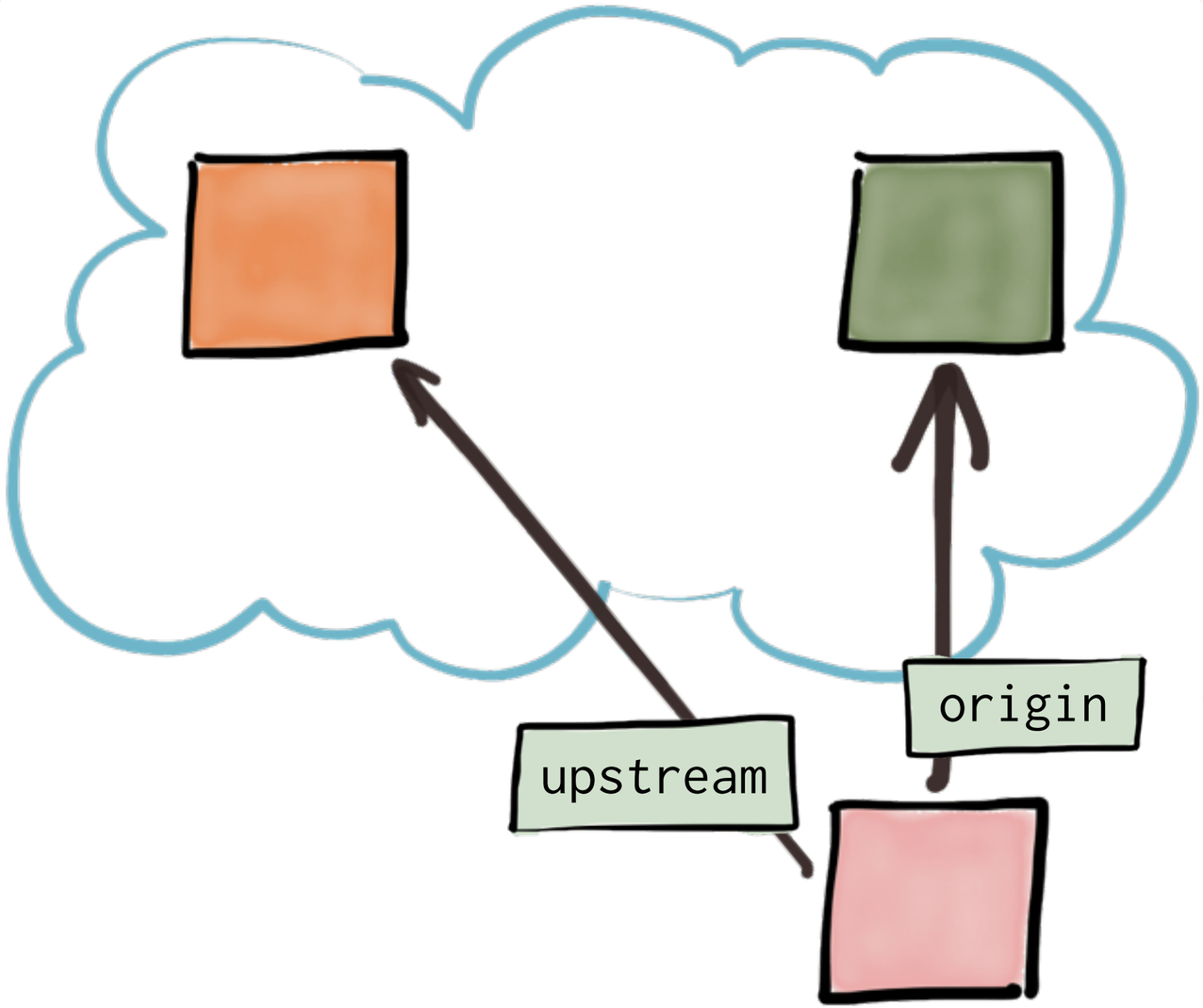
Fork



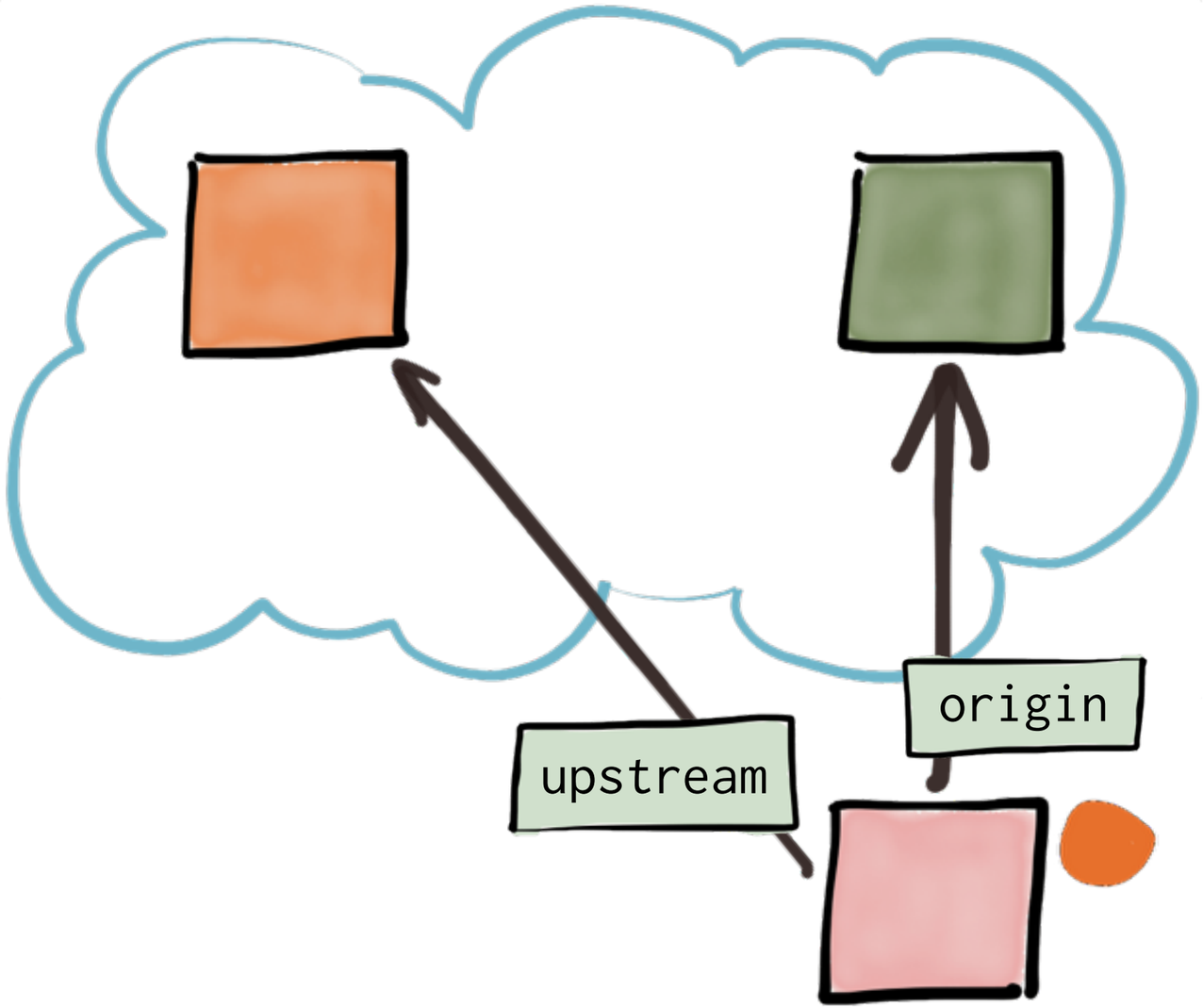
Clone



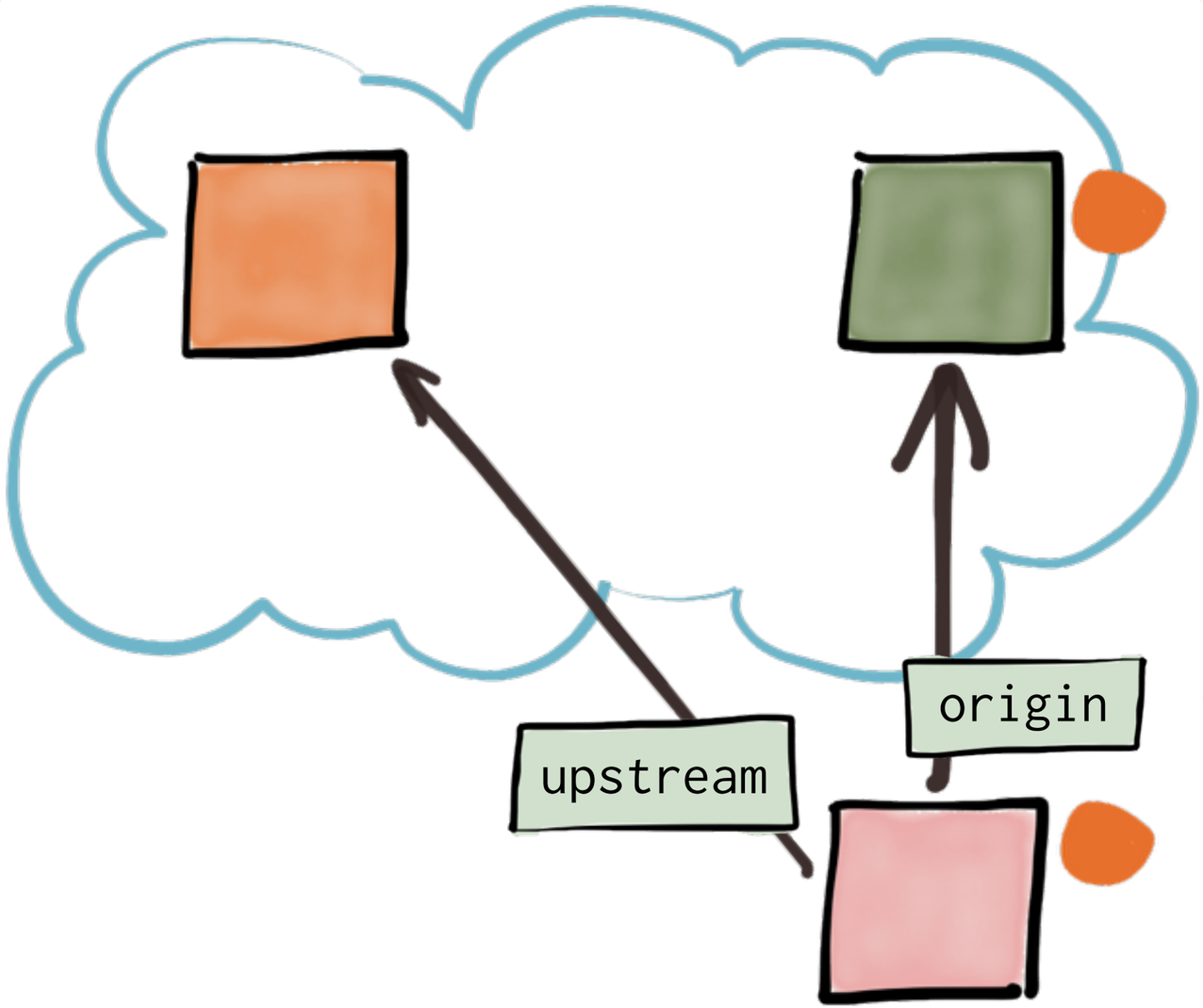
Two Remotes

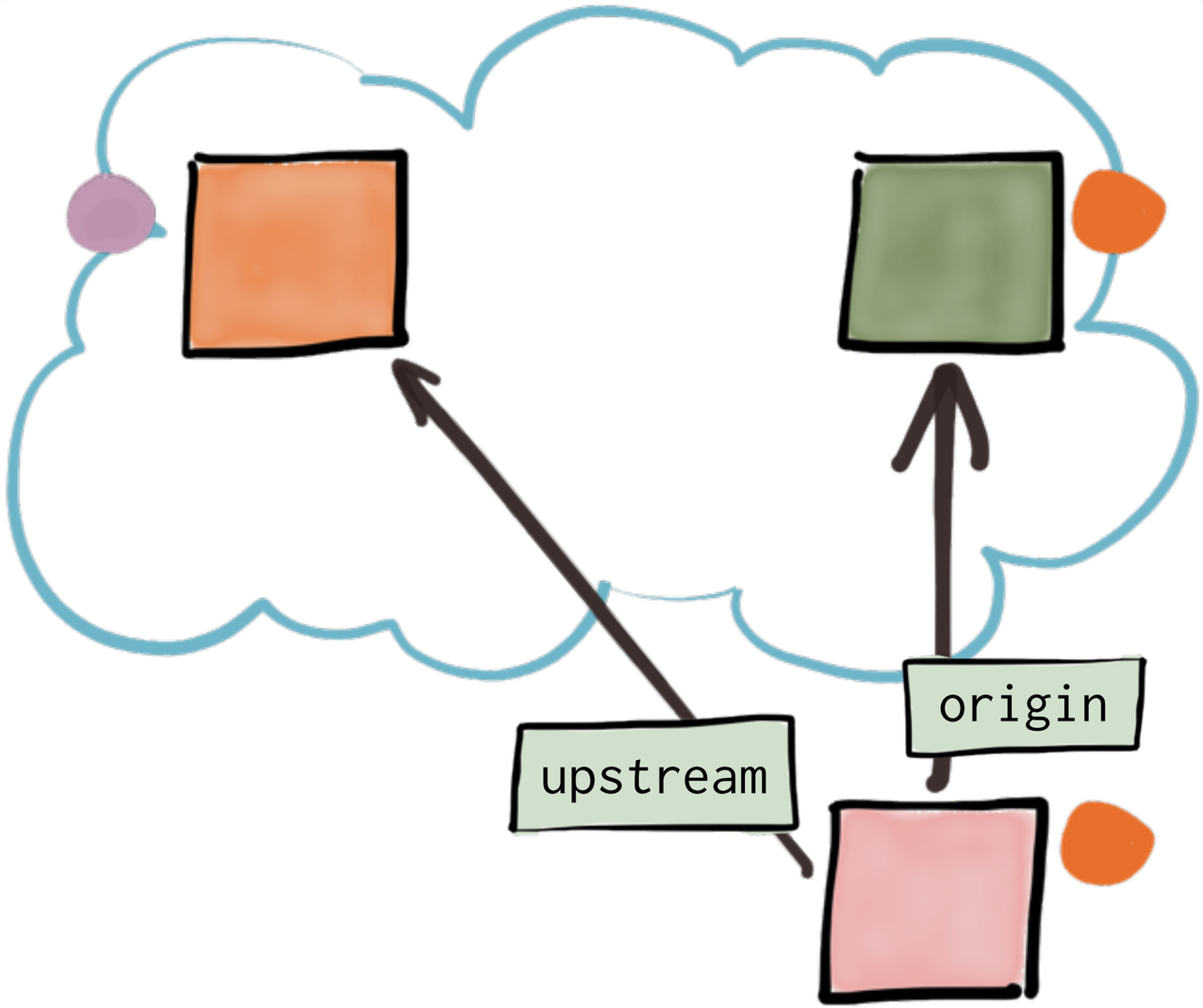


Commit

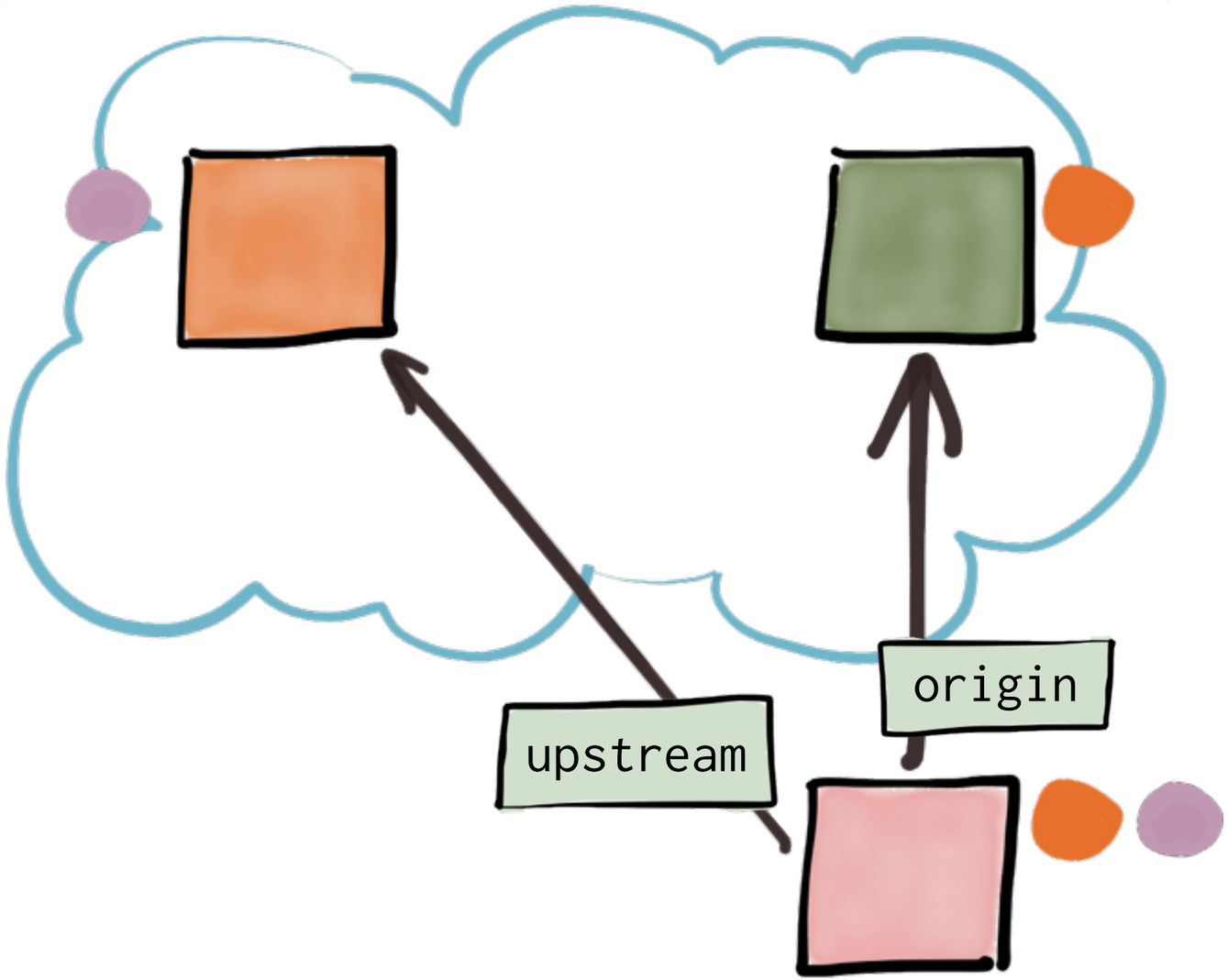


Push to Origin

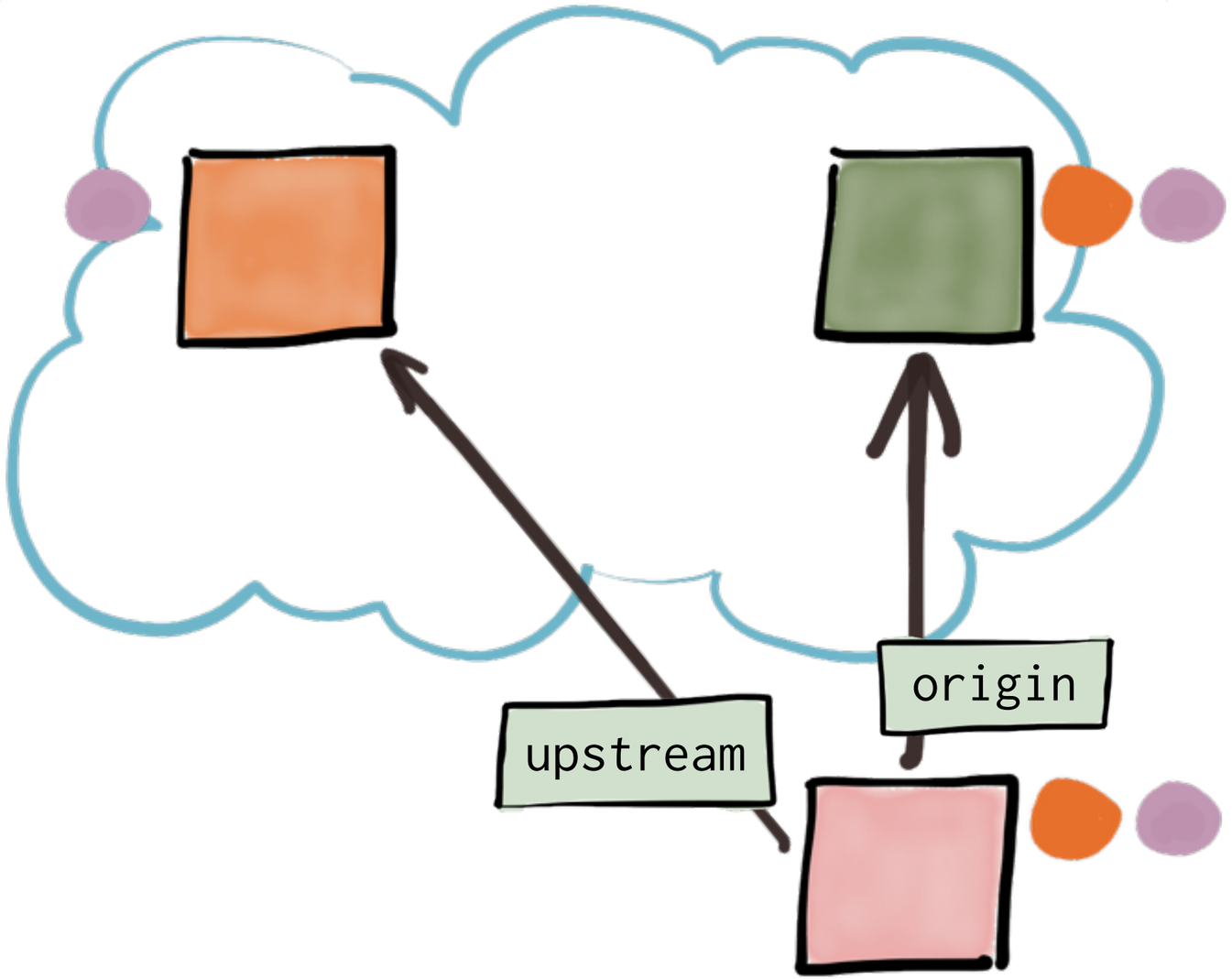



upstream
origin

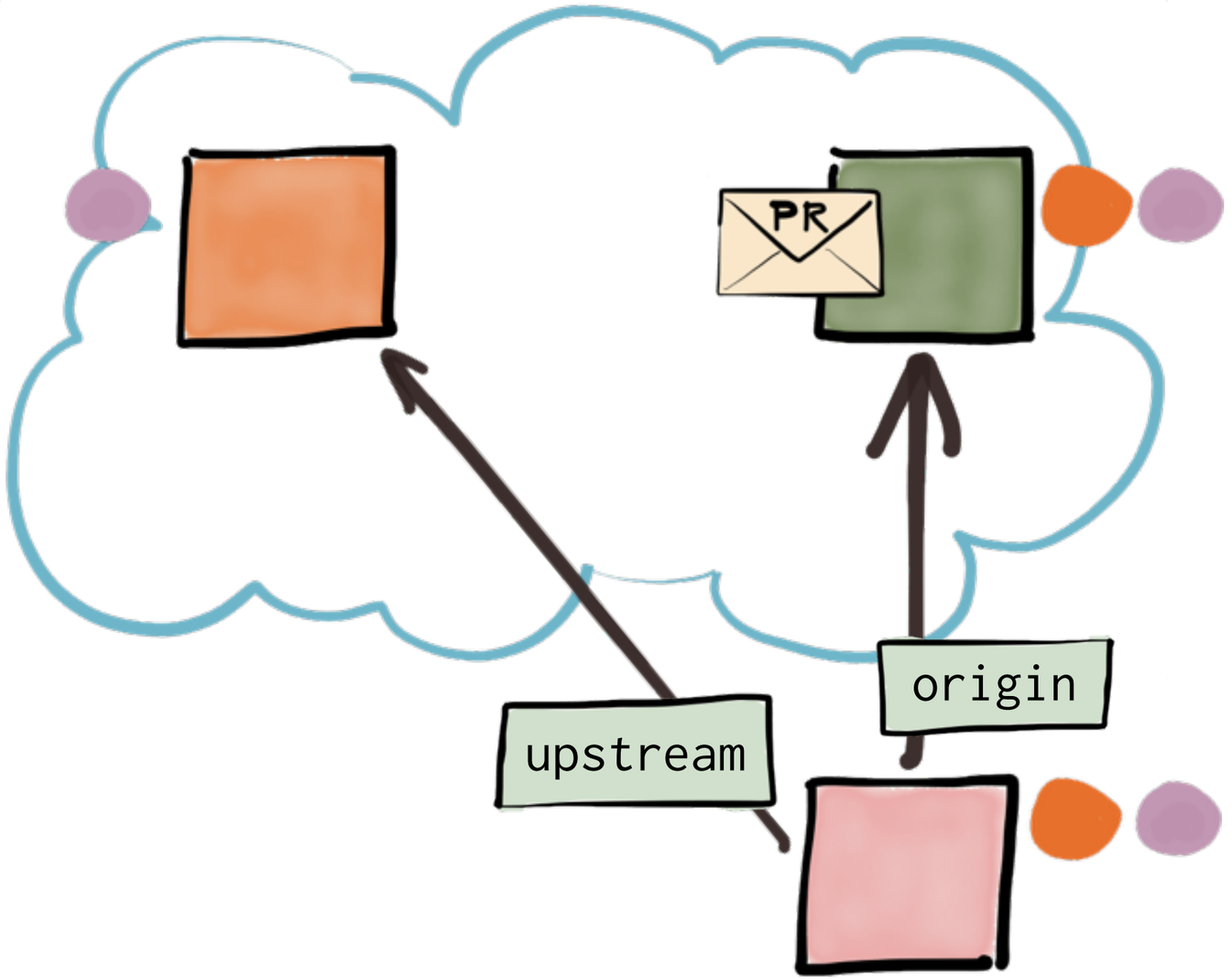
Pull from Upstream



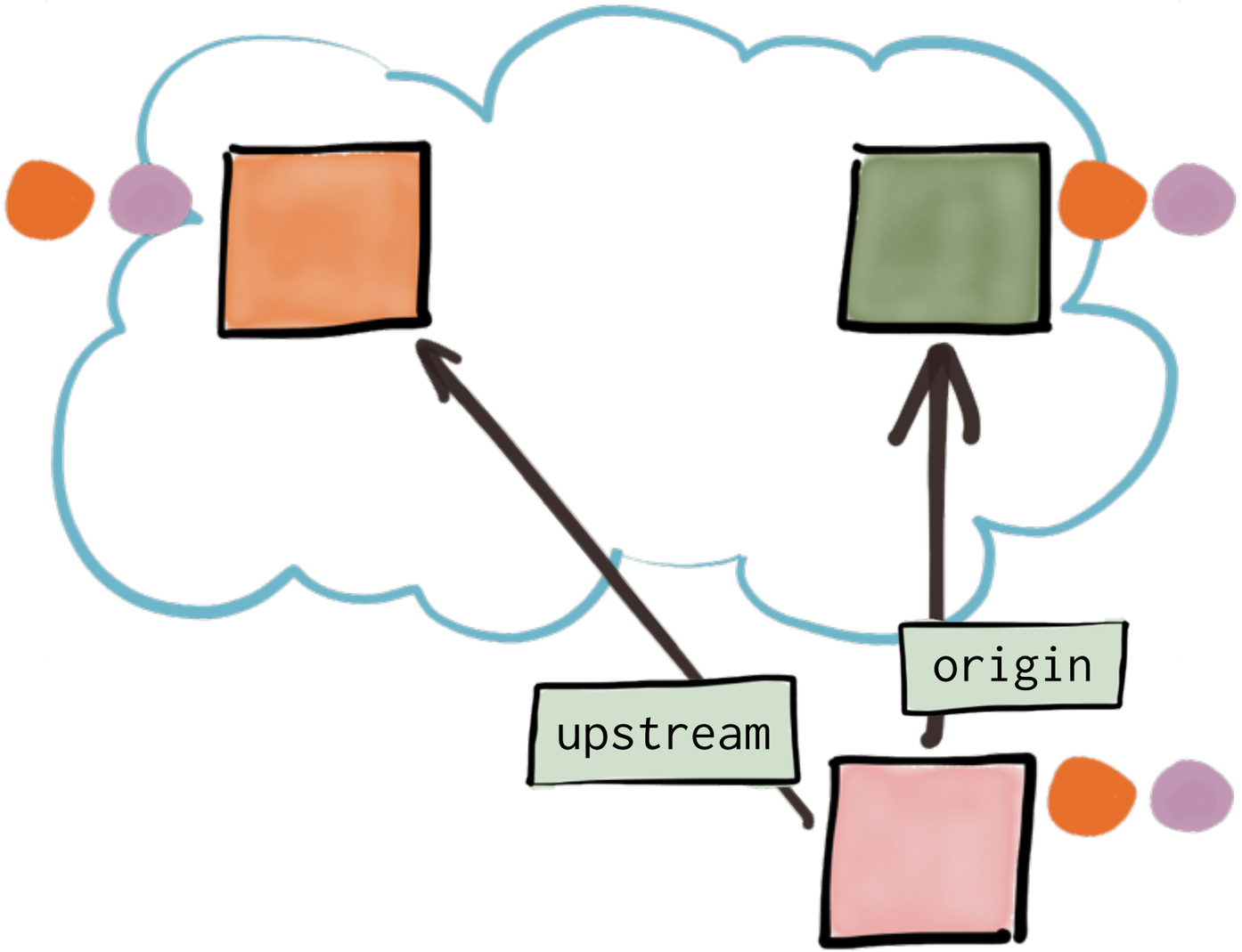
Push to Origin



Pull Requests



Pull Requests

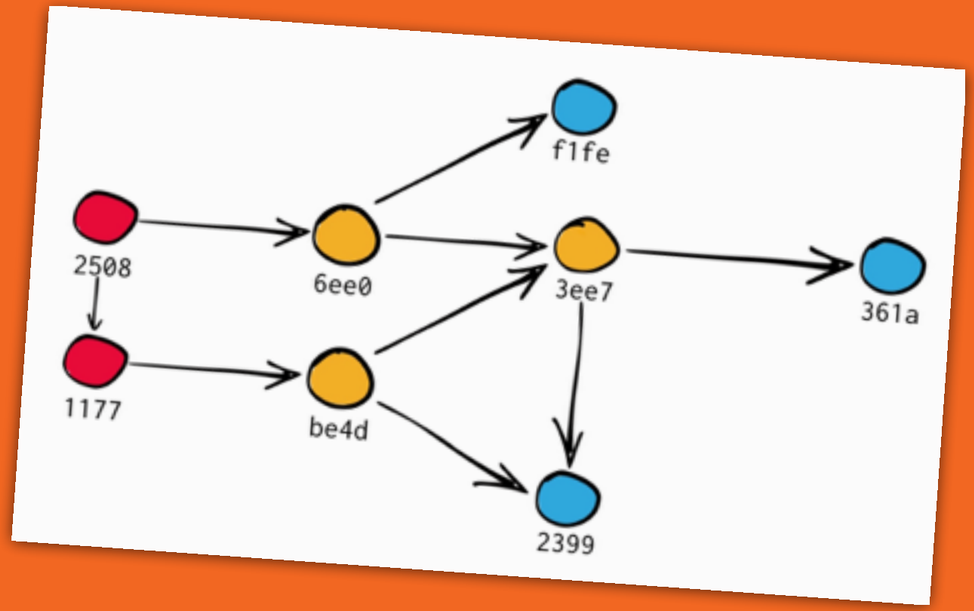


Git Is..

…a Persistent Map

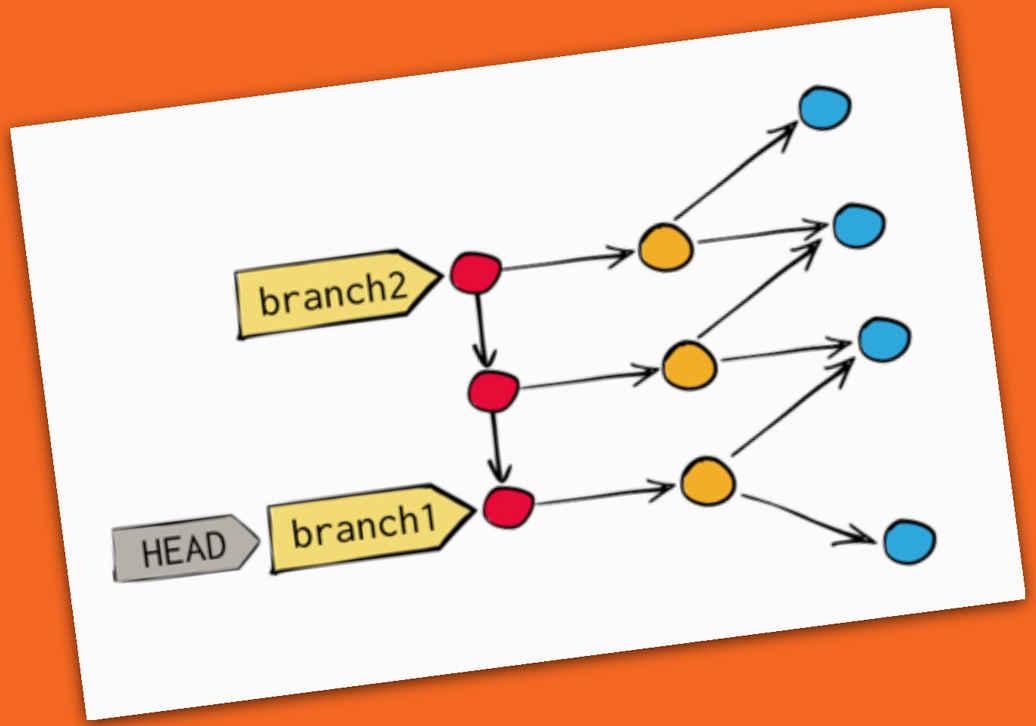
Git Is..

…a Stupid Content Tracker



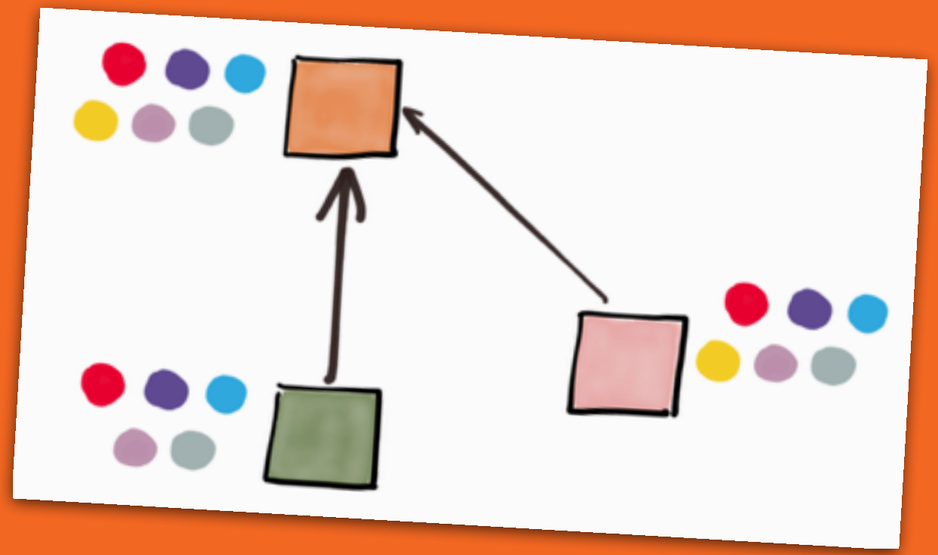
Git Is..

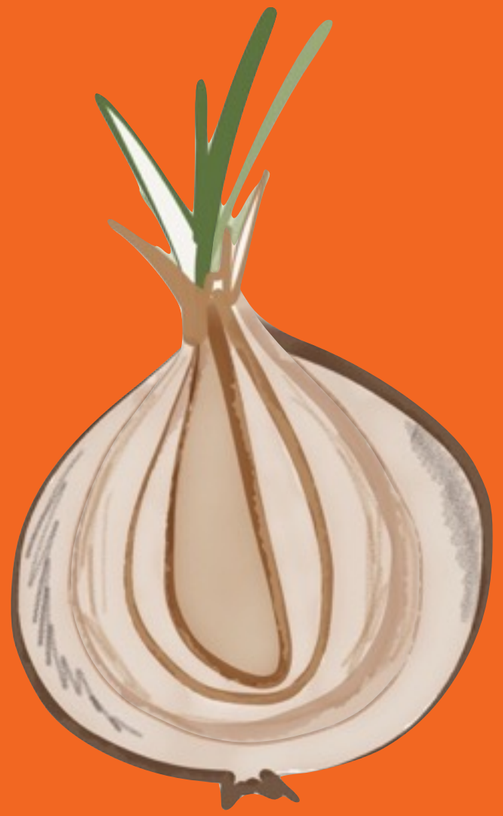
…a Revision Control System



Git Is..

…a Distributed Revision Control System






Thank
you!