

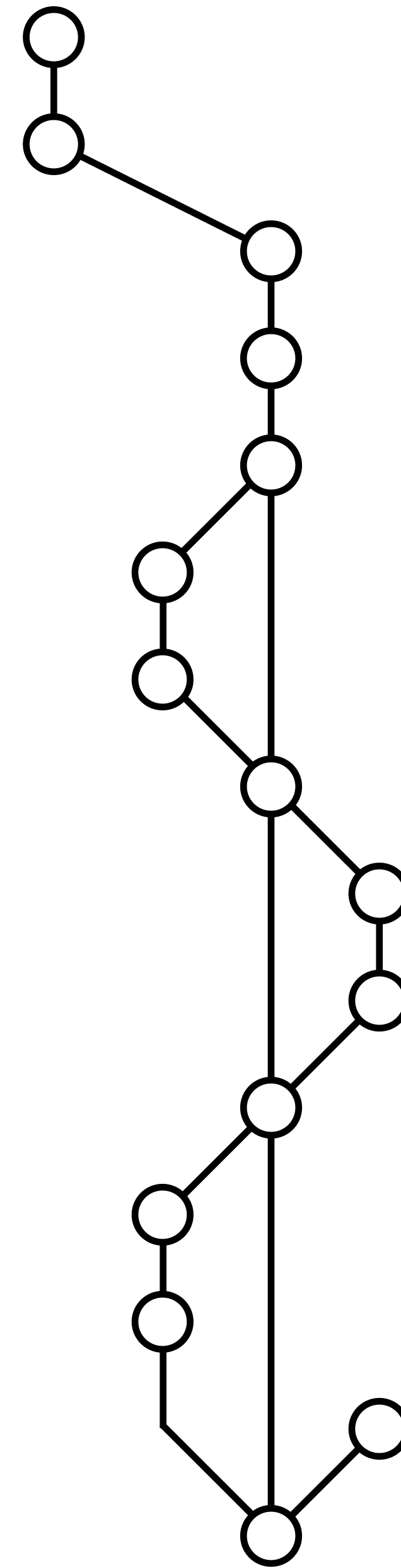
GitHub for the Rest of Us

Morten Rand-Hendriksen, Y Oslo, 2023

Cheat sheet for
(most) git
commands.

REPOSITORY

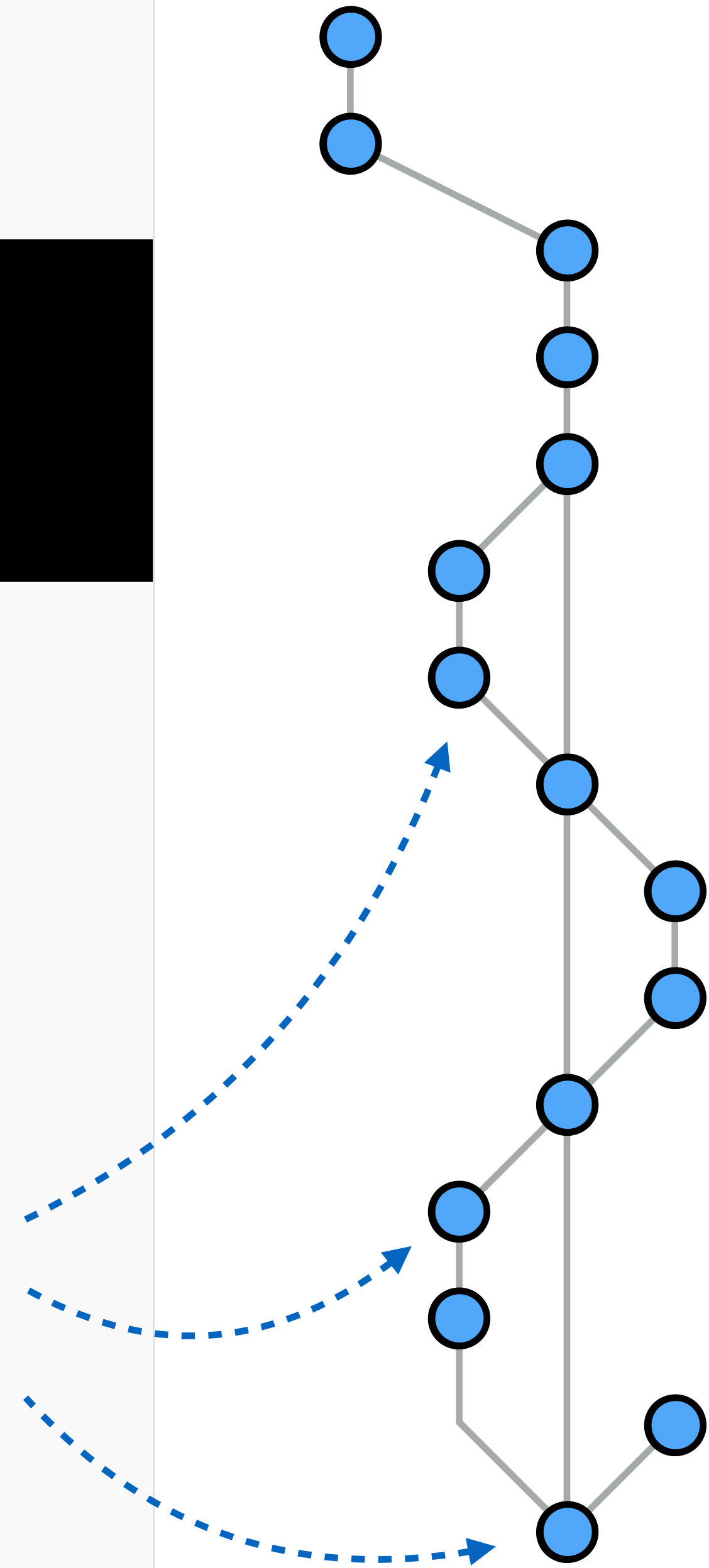
The universe in which your project exists.



```
git commit
```

Create a snapshot, or save point, to which you can always return.

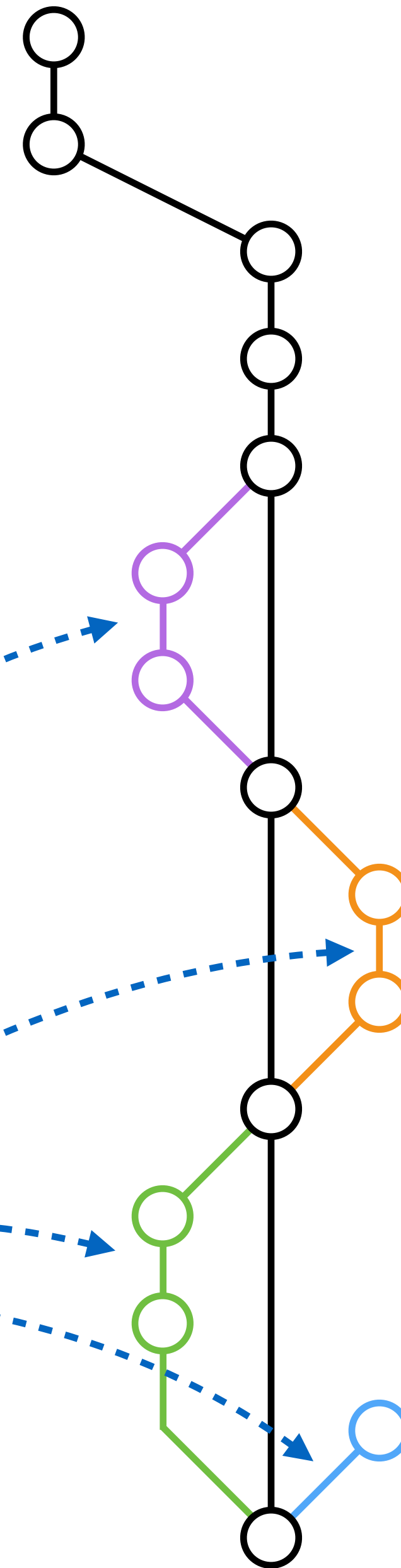
Commits



git branch

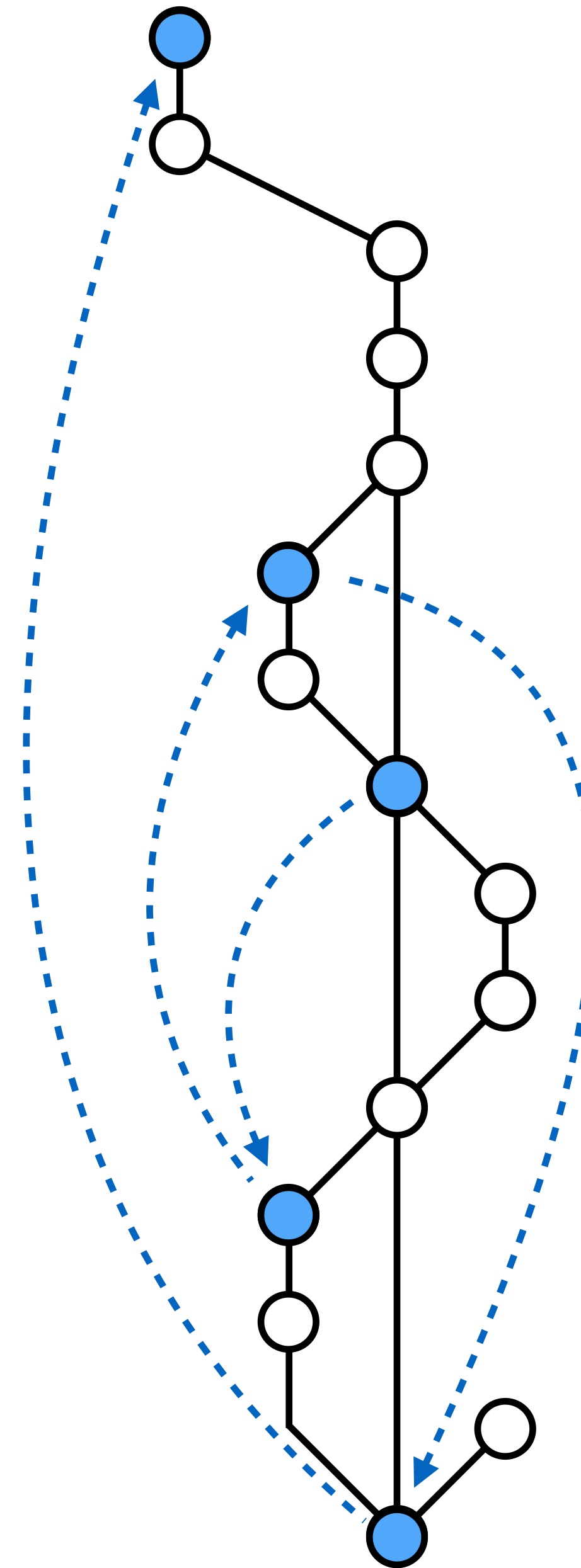
An alternate timeline within
the current universe.

Branches



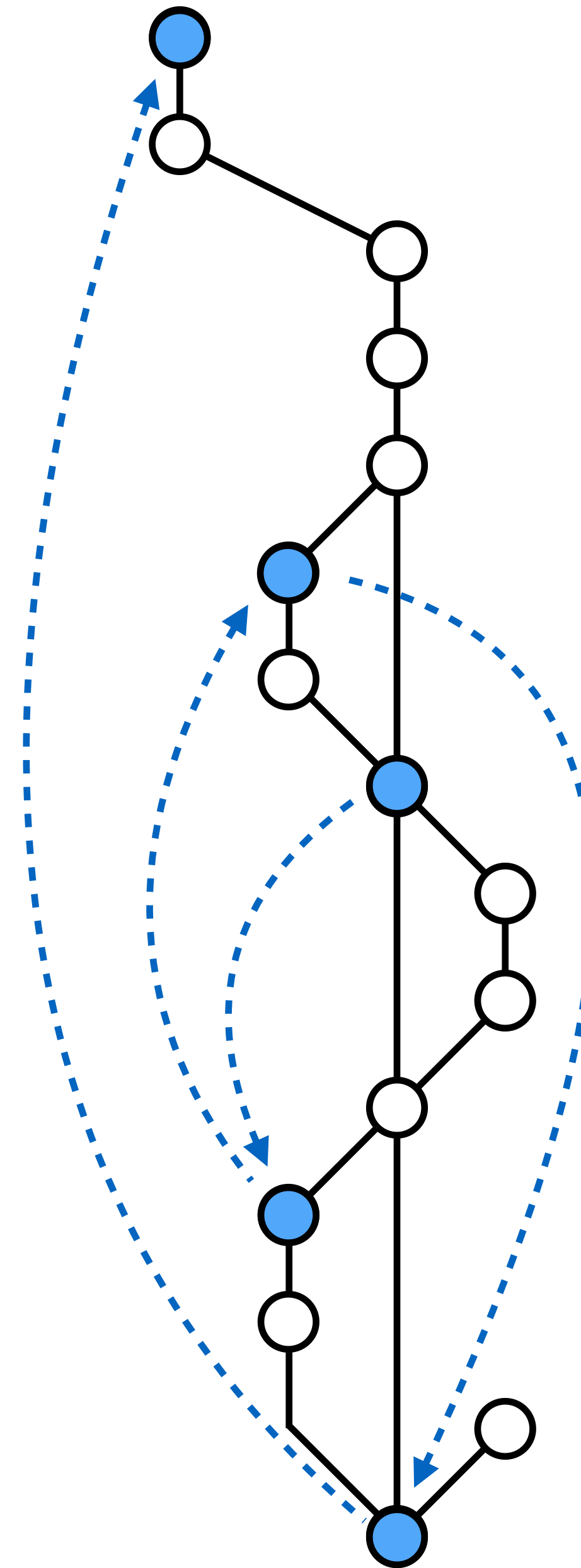
git checkout

Your personal time machine.
Checkout allows you to jump
to any commit on any branch,
in your current universe or
any other universe.



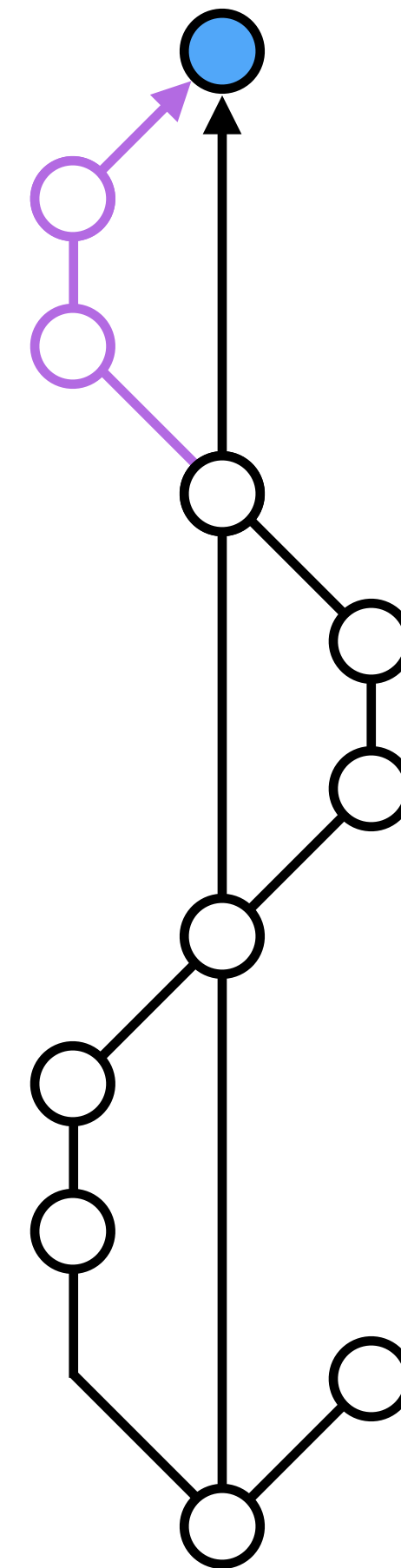
git switch

Same as git checkout,
introduced to create clearer
separation of concerns.



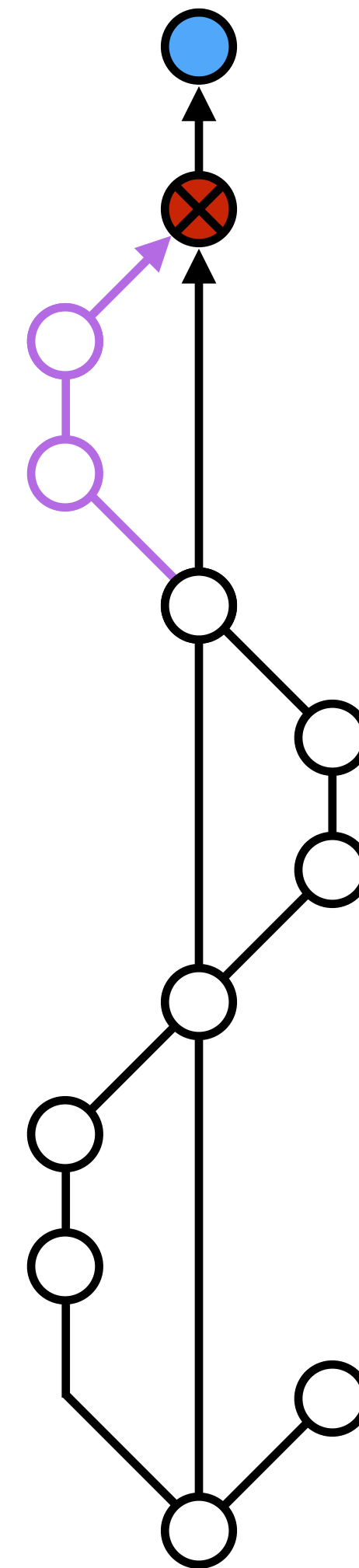
git merge

Merge the contents of one timeline into another.



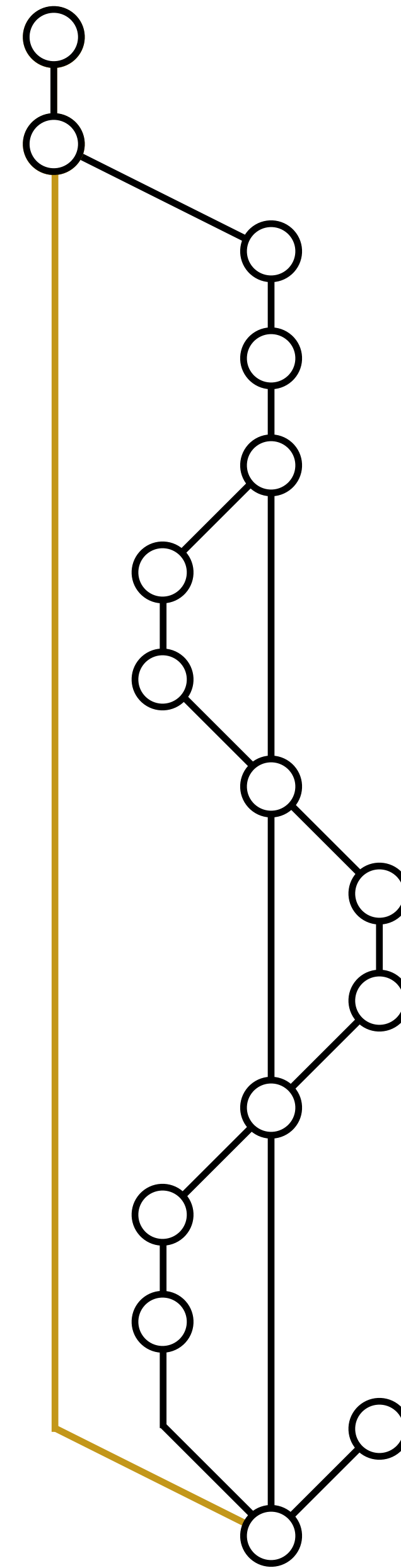
CONFLICT

Changes have taken place in both timelines after they were split apart. You must decide what change to keep and what change to discard.



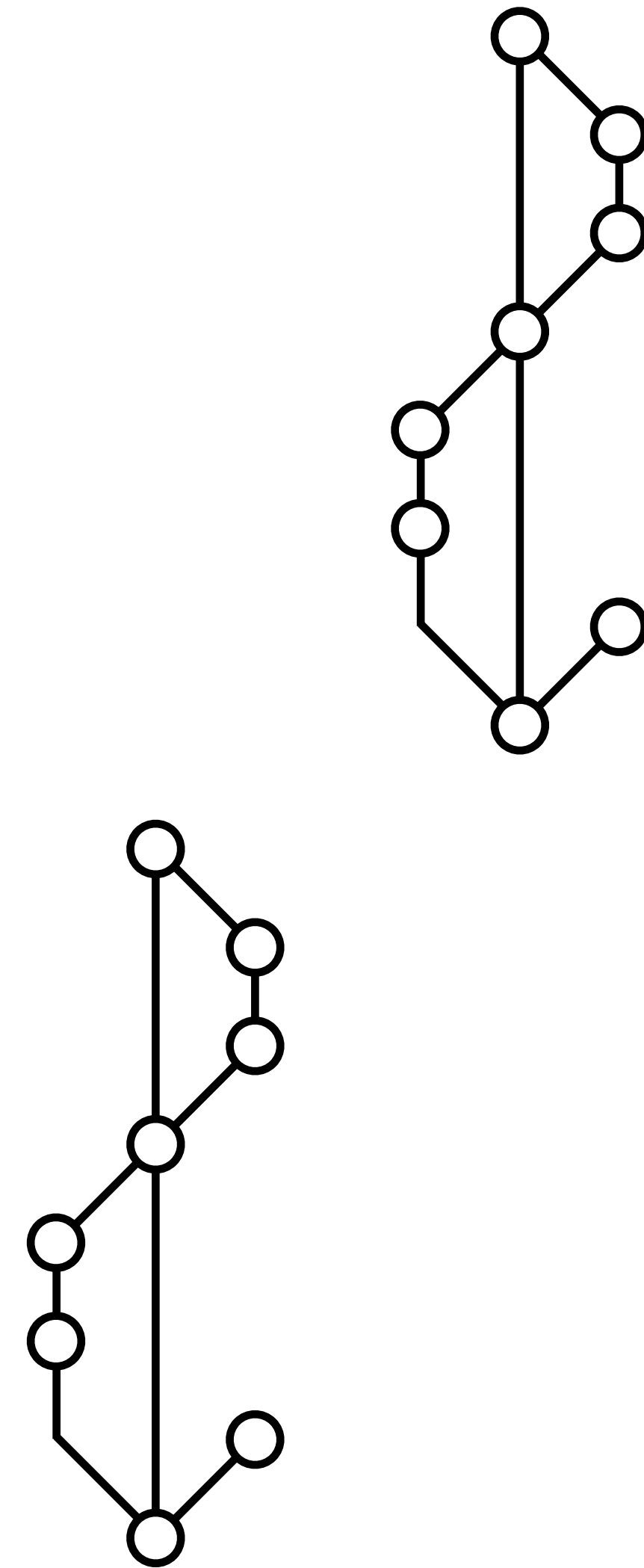
git rebase

Rewrite history by tacking the events of one timeline onto another as if they were always there.



git fork

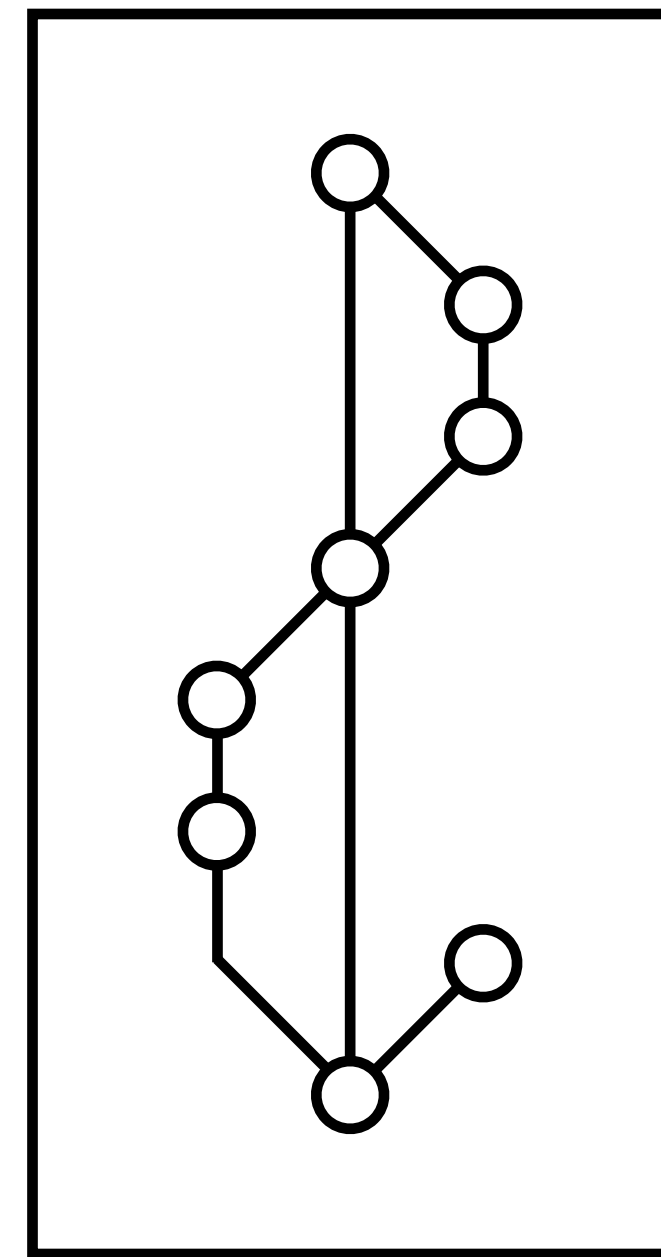
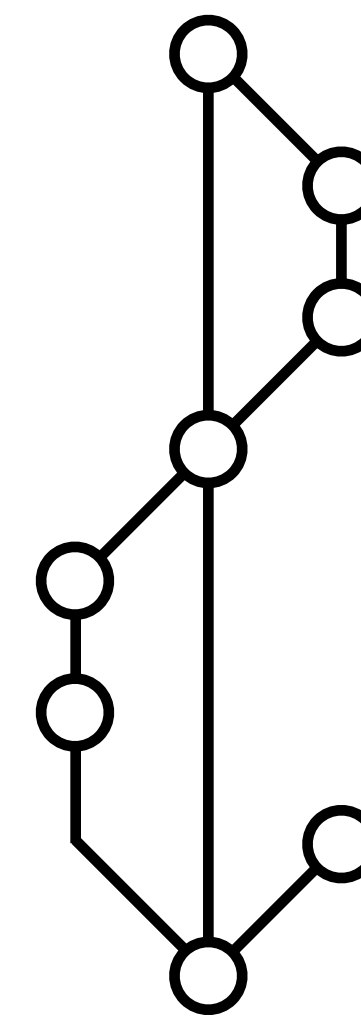
An alternate universe that starts as an exact copy of the one you forked.



`git clone`

Make a duplicate copy of an external universe and place it in the current one.

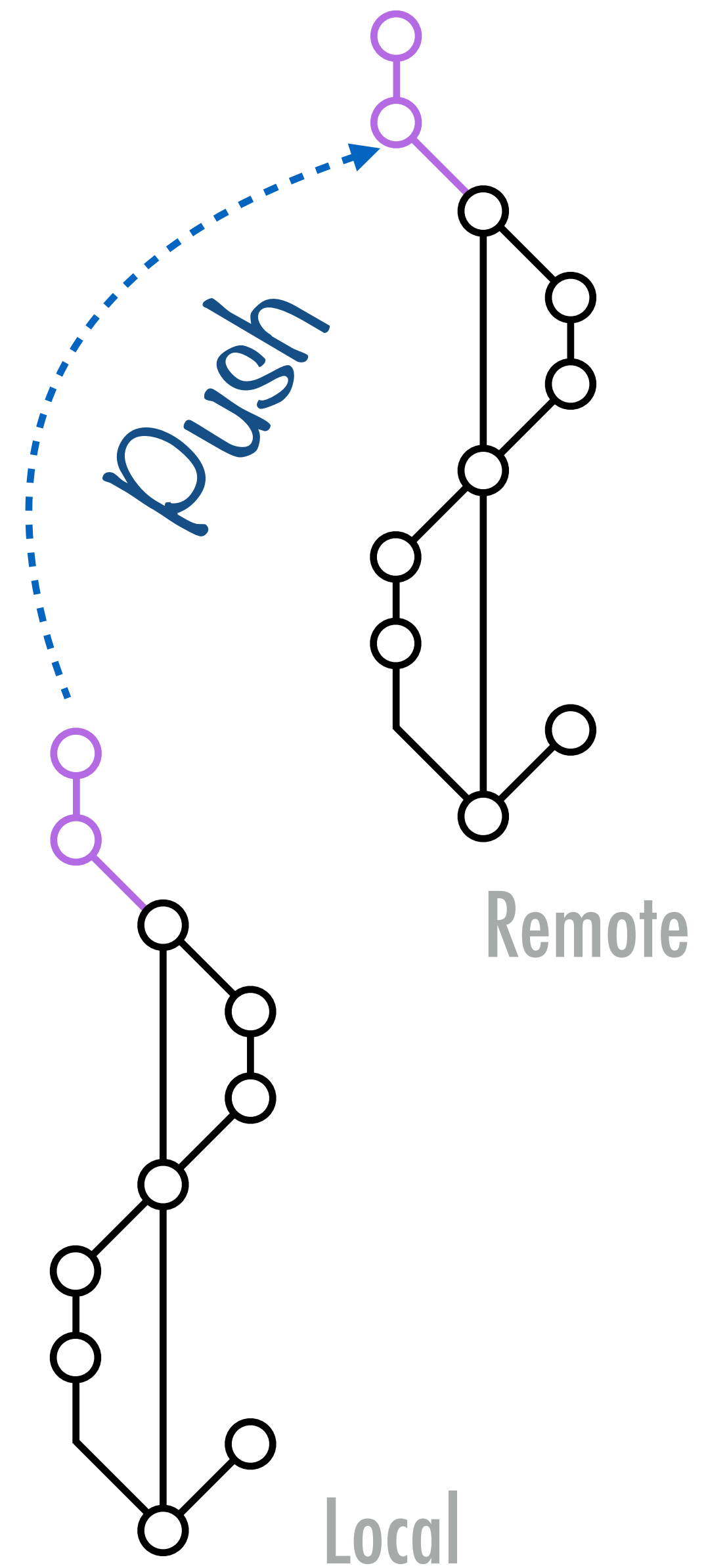
Empty repo



Clone

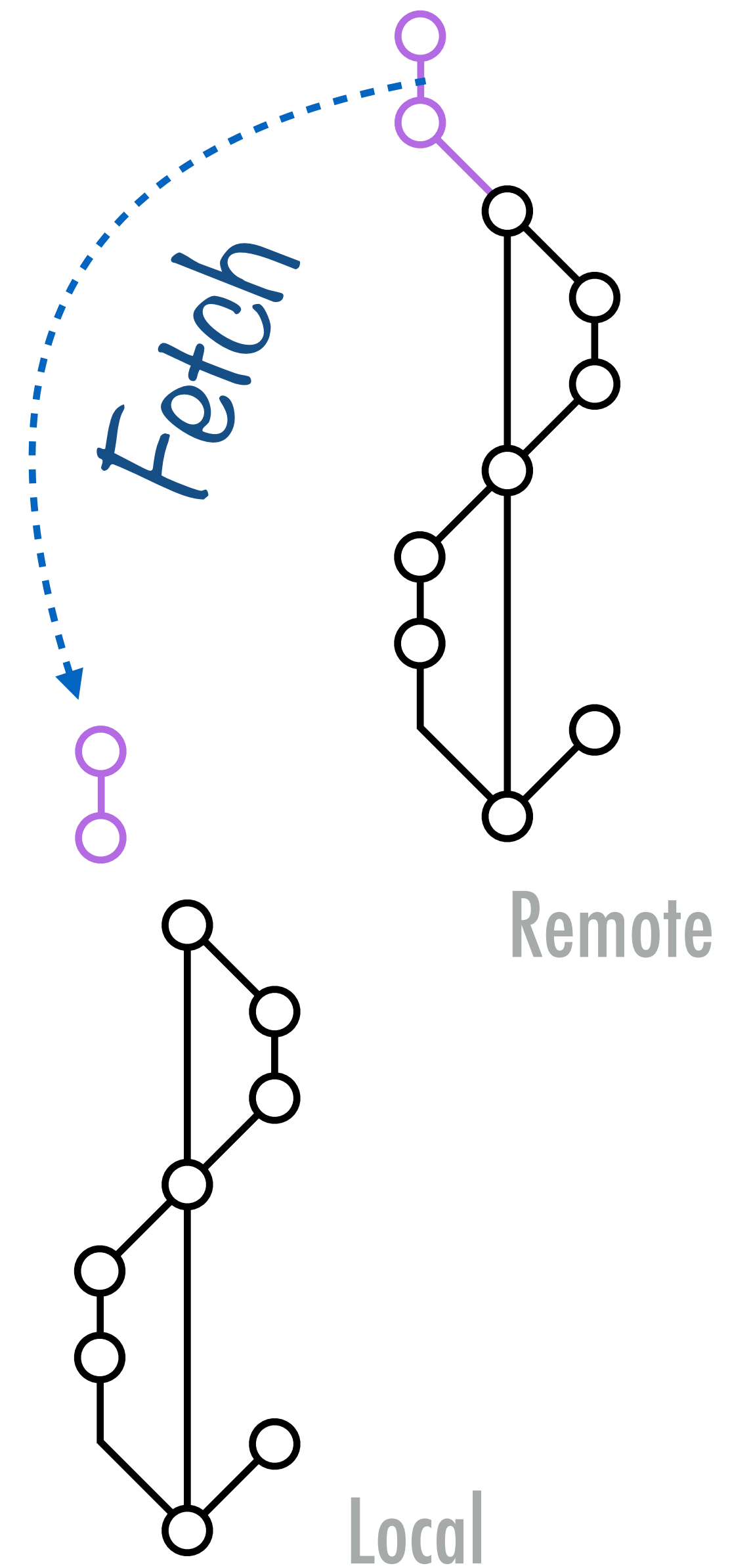
`git push`

Physically push your
commits to another
repository
(typically GitHub)



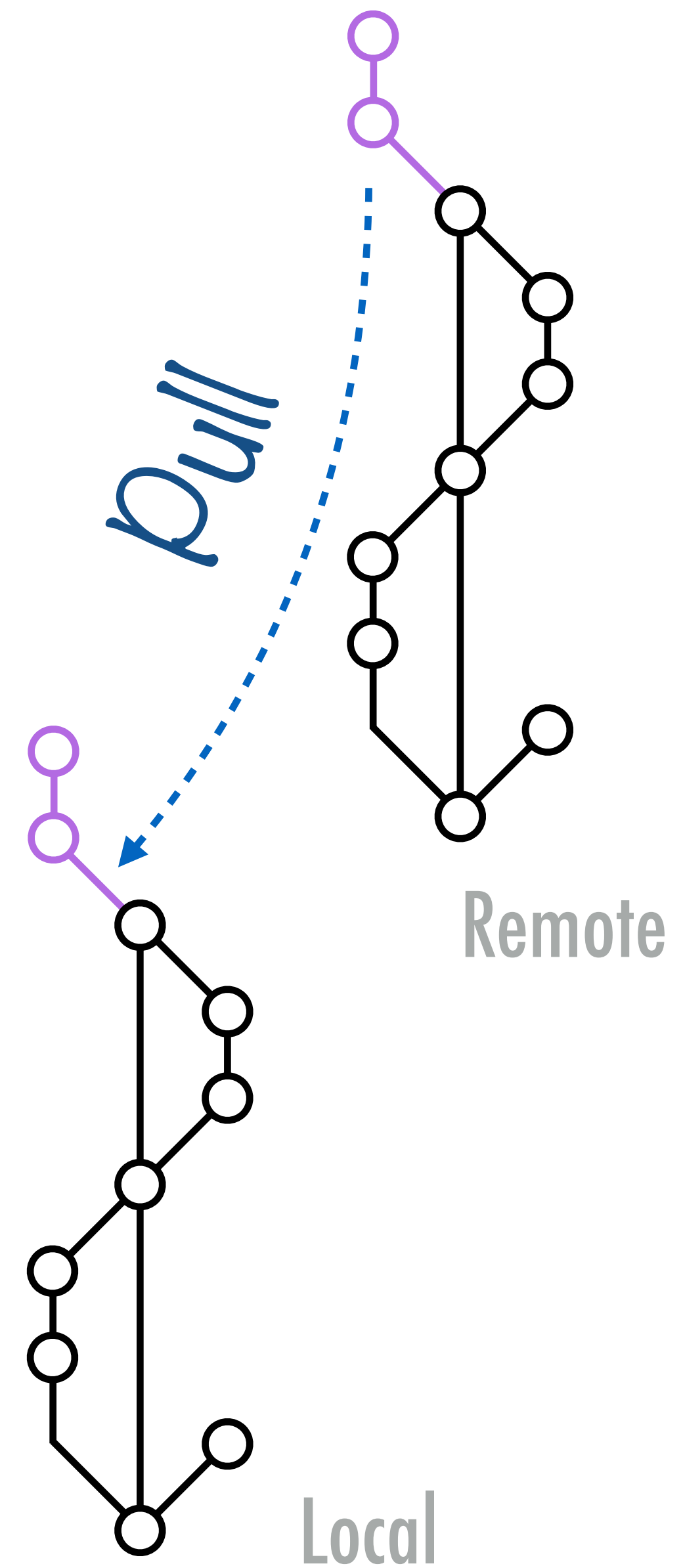
`git fetch`

Grab the current version of
an external timeline and
open it.



`git pull`

Grab the current version of an external timeline and merge it with the current timeline.



DETACHED HEAD

Before using the time machine, changes were made without being committed, thereby detaching them. Create a new timeline to restore reality.

