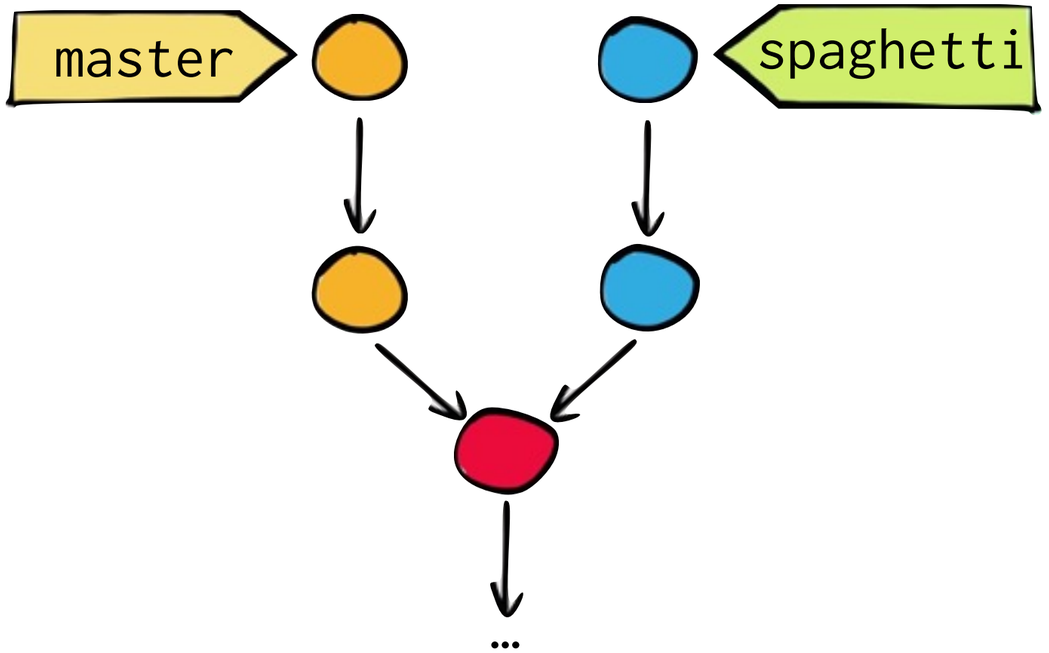
Rebasing Made Simple

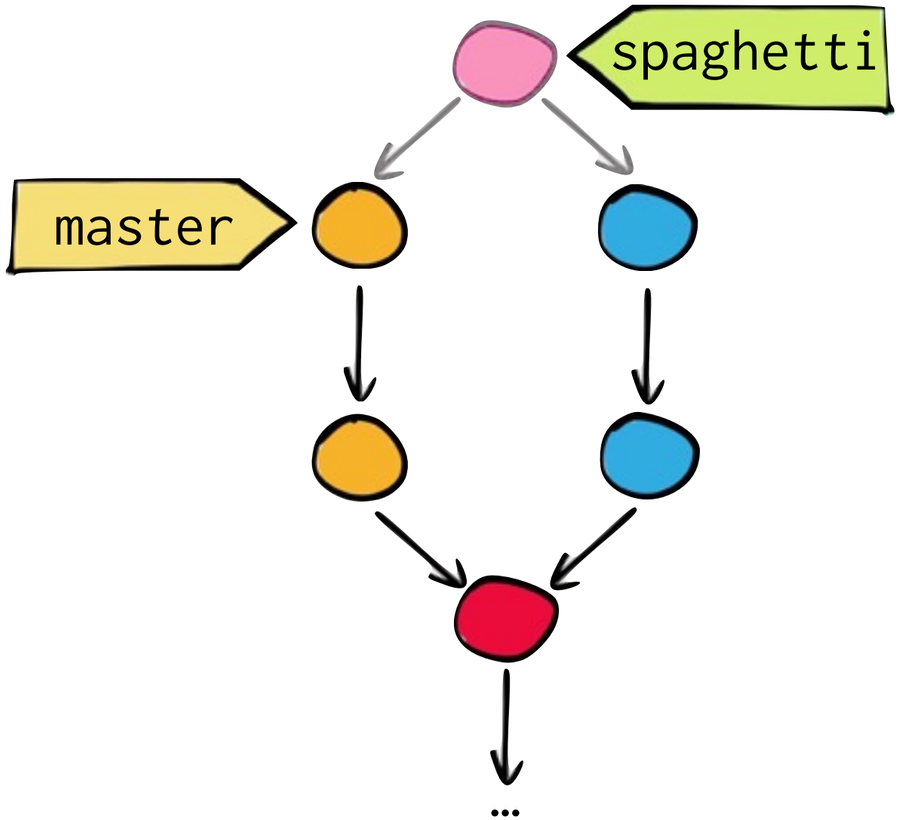
Paolo Perrotta

@nusco

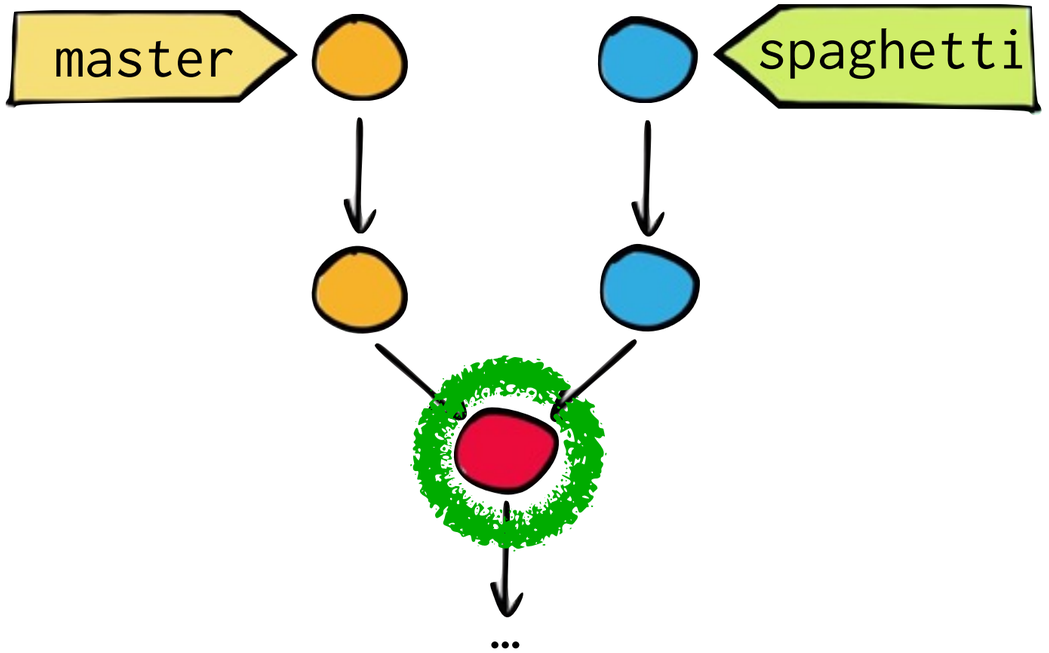
Two Branches



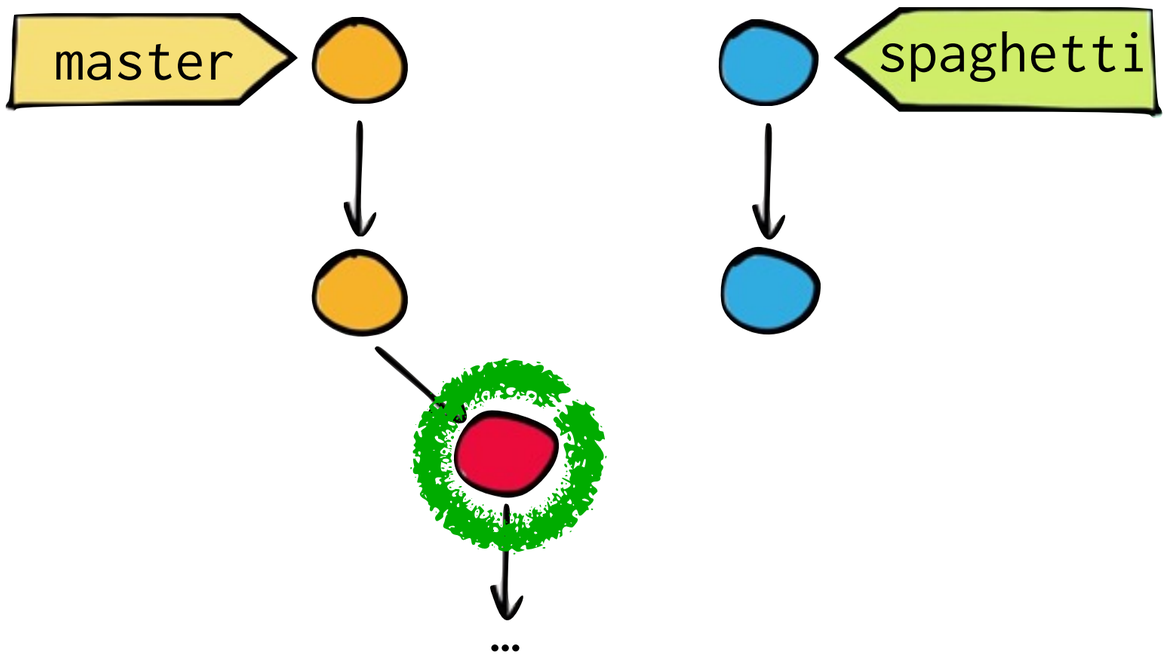
Merge



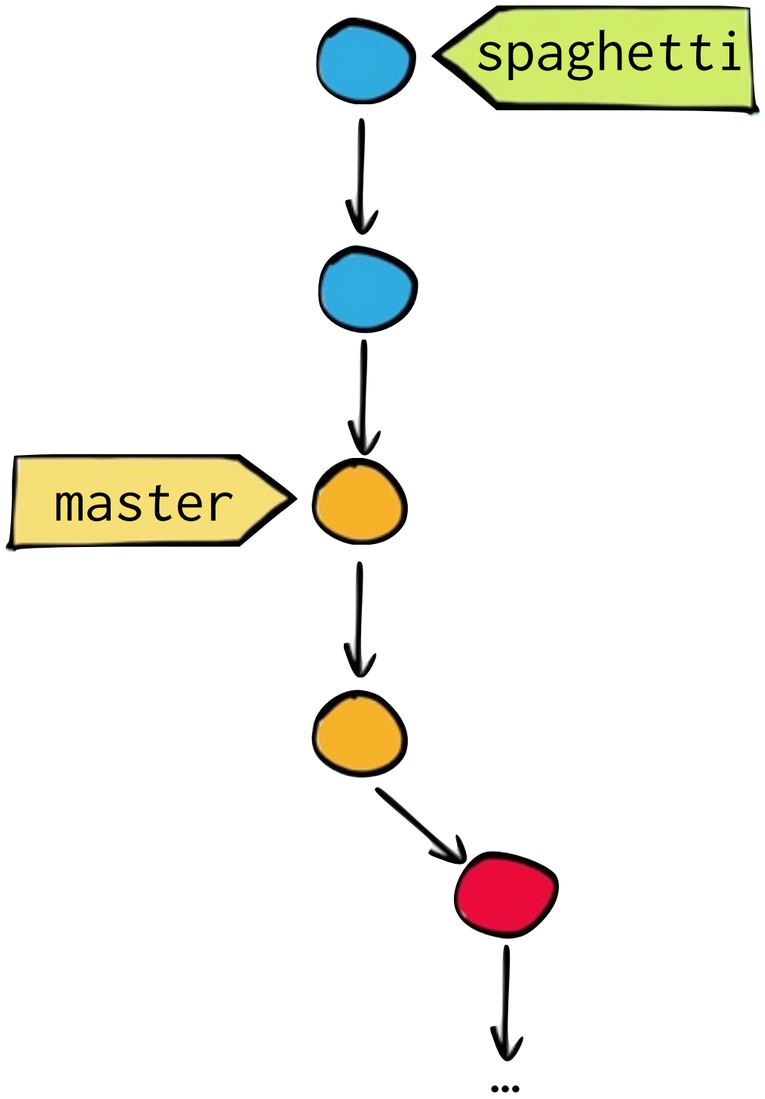
Rebase



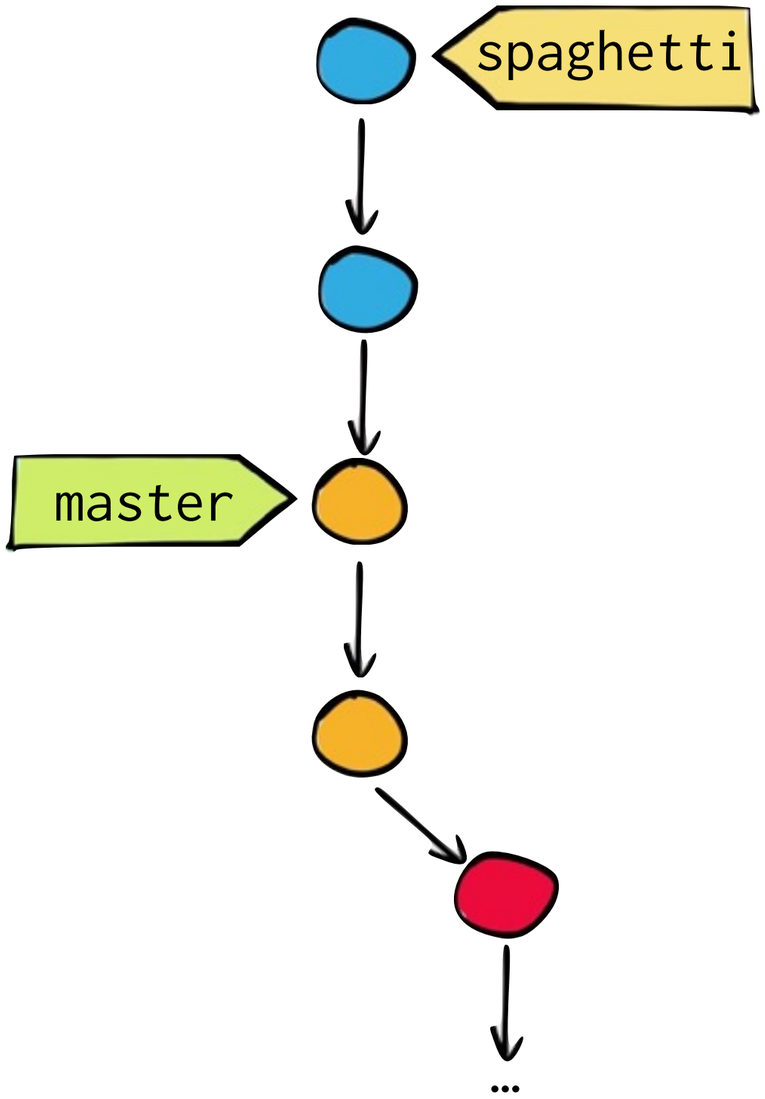
Rebase



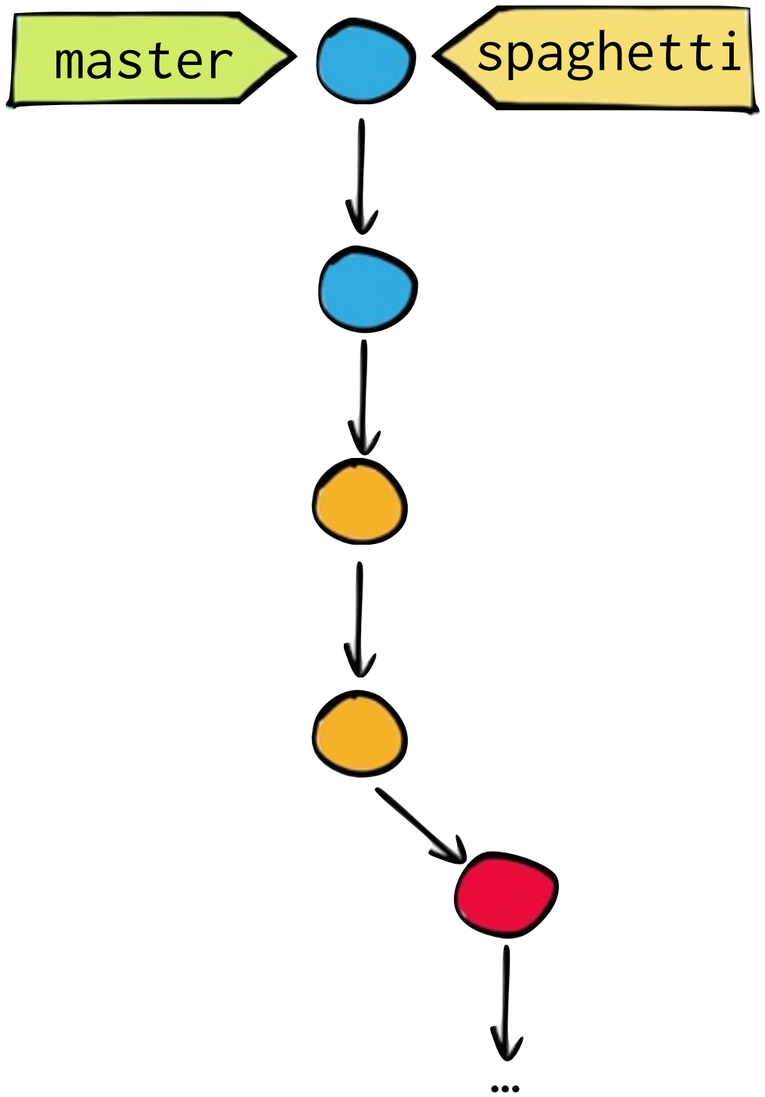
Rebase



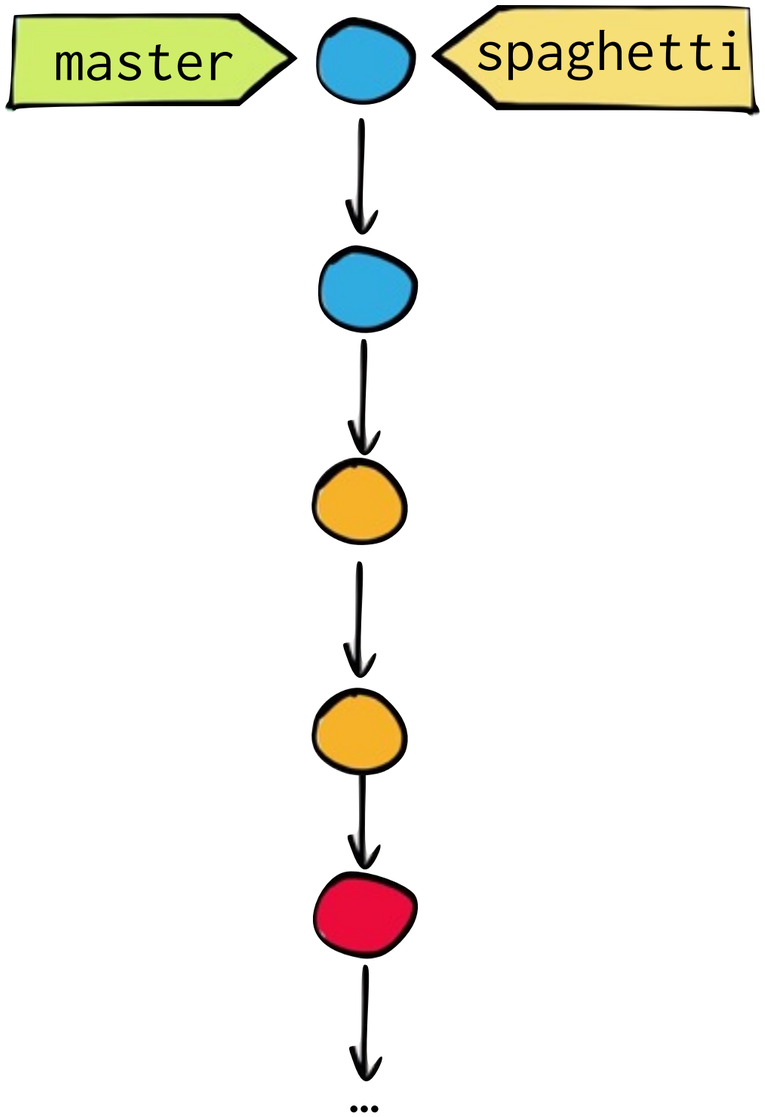
Checkout

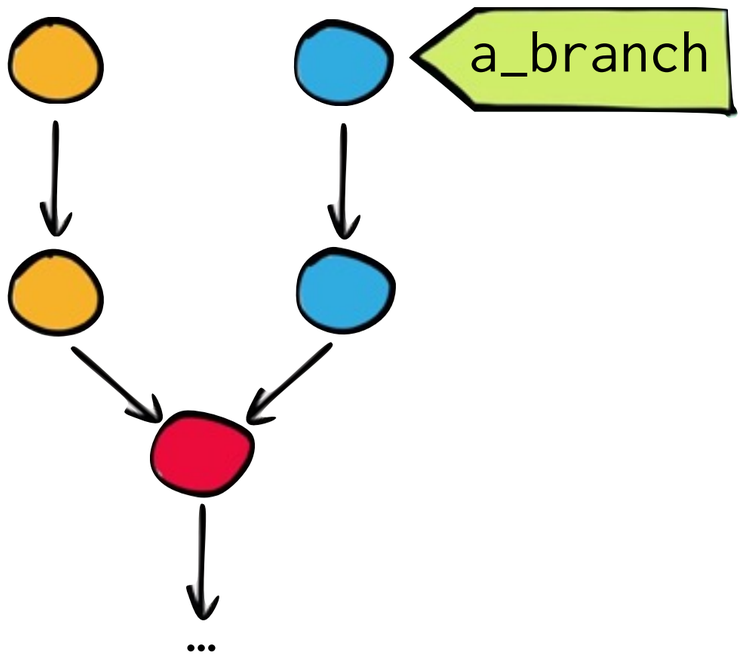


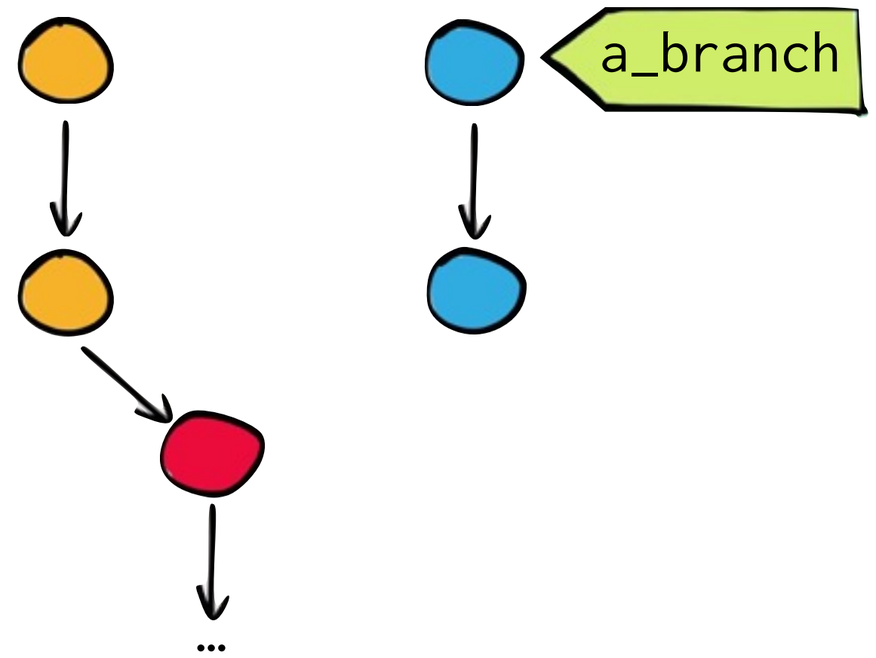
Rebase (with Fast-forward)

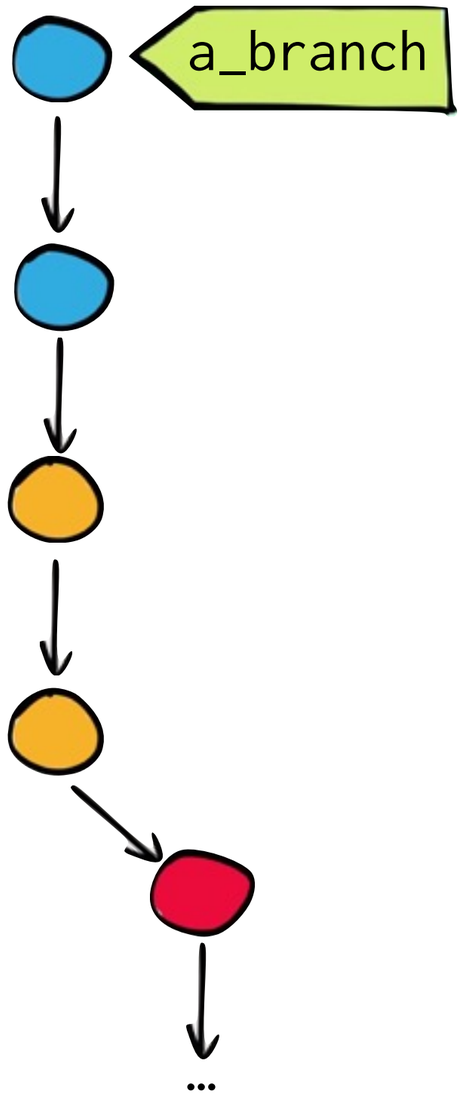


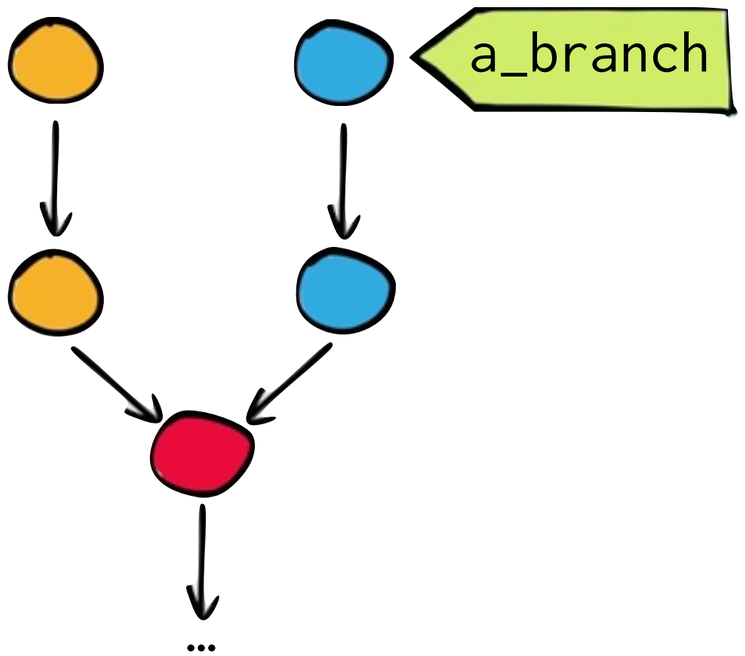
After a Rebase

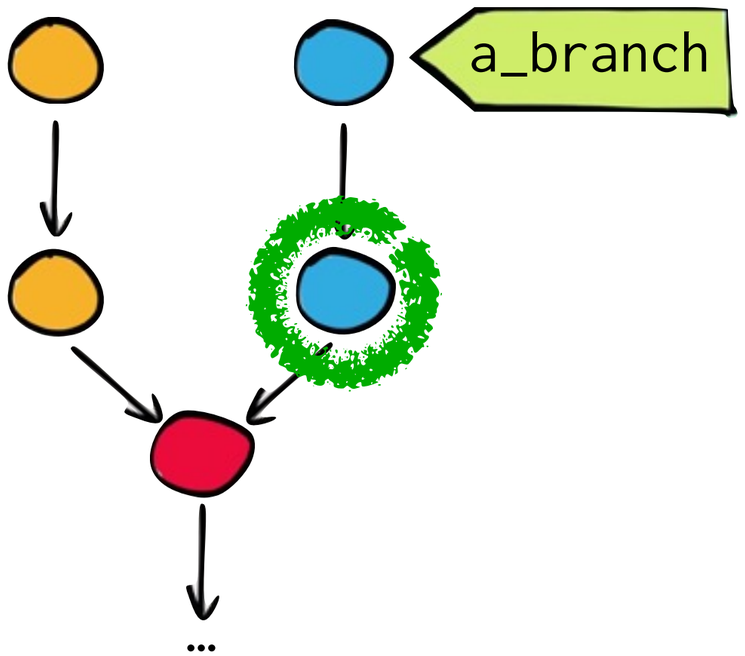


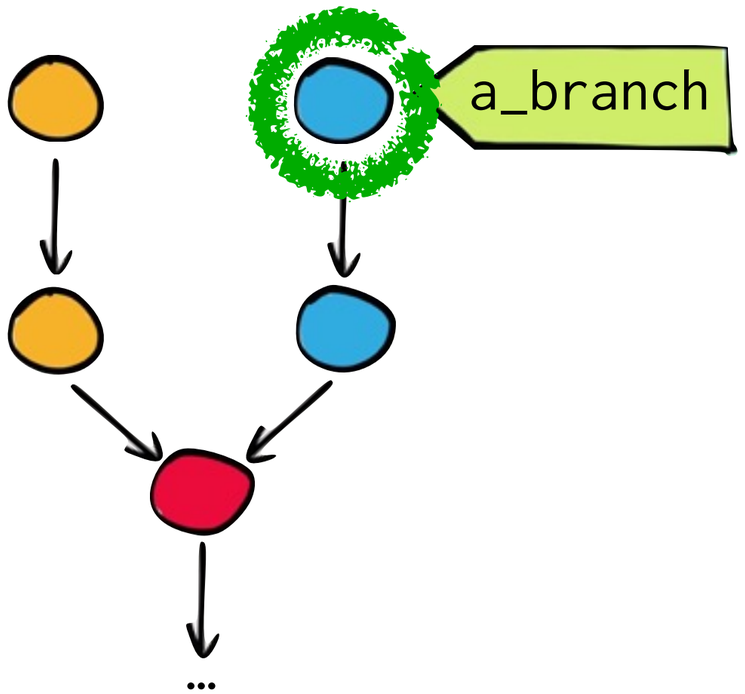

a_branch
...

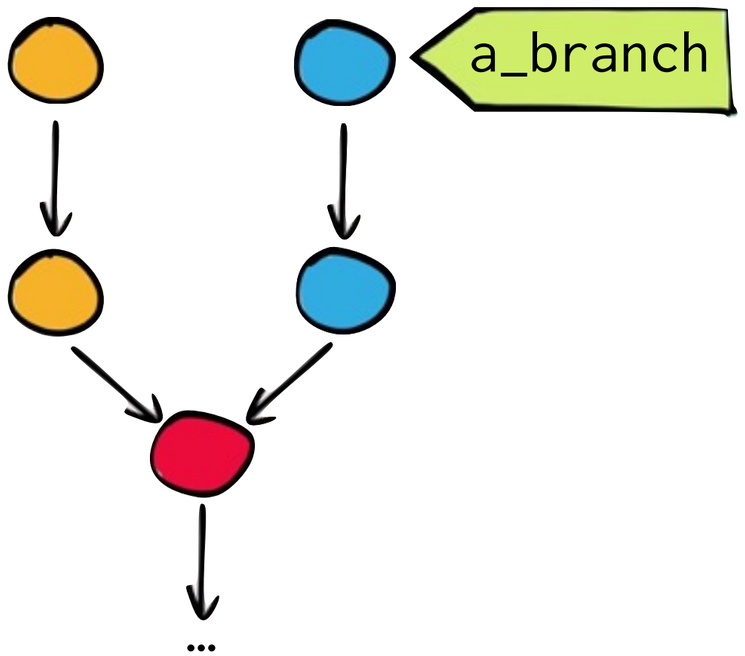

a_branch
...

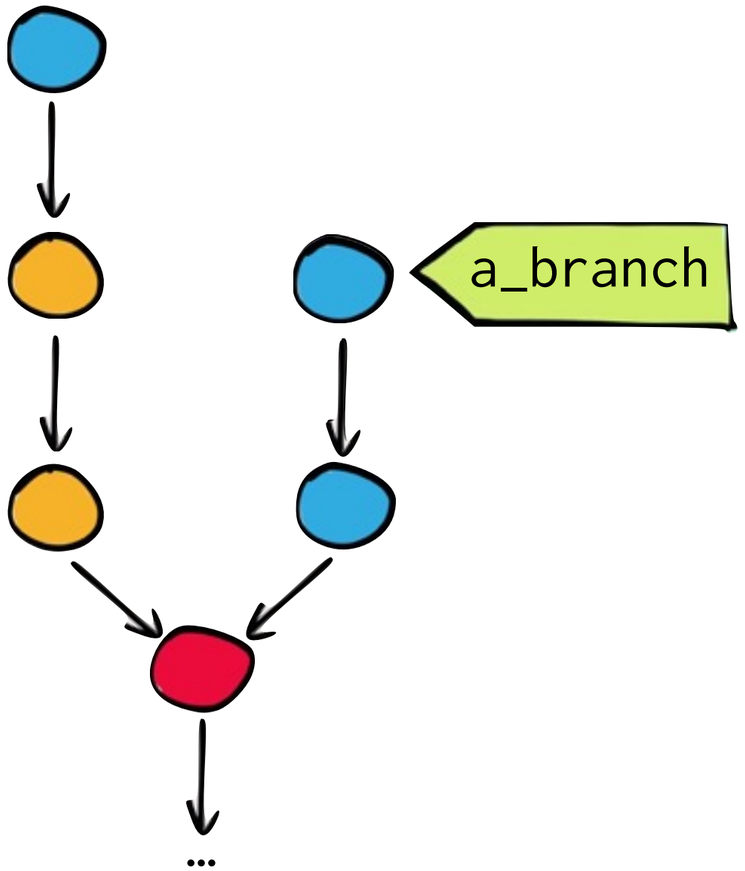

a_branch
...

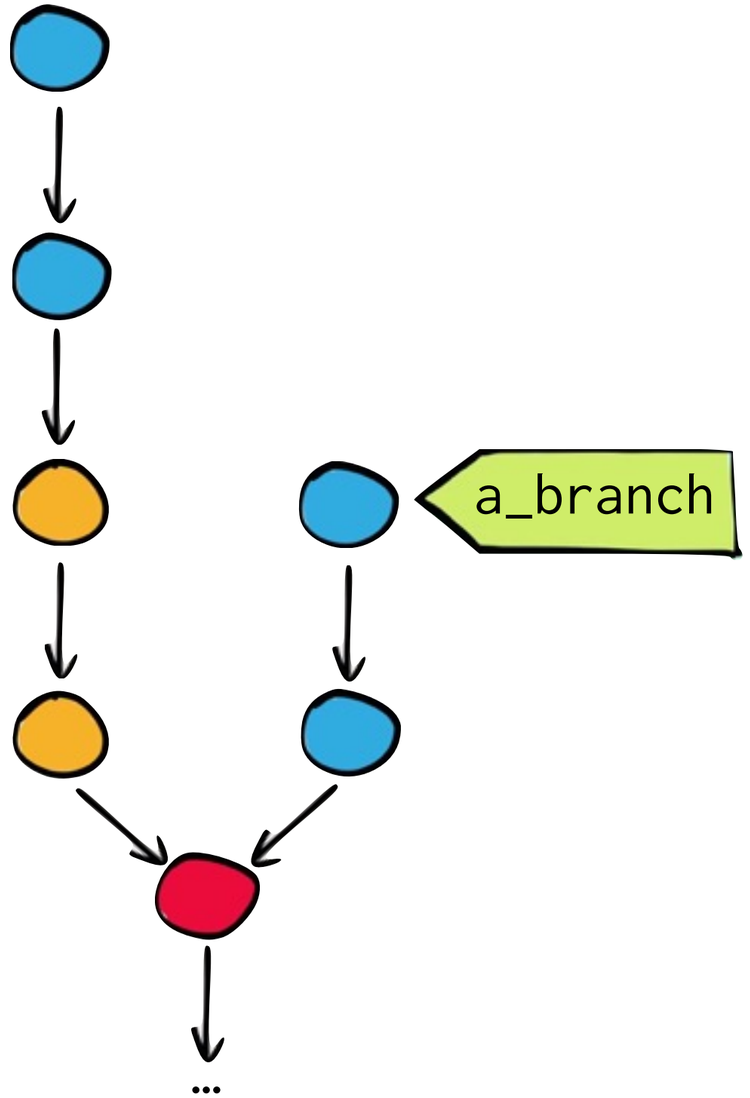

a_branch
...

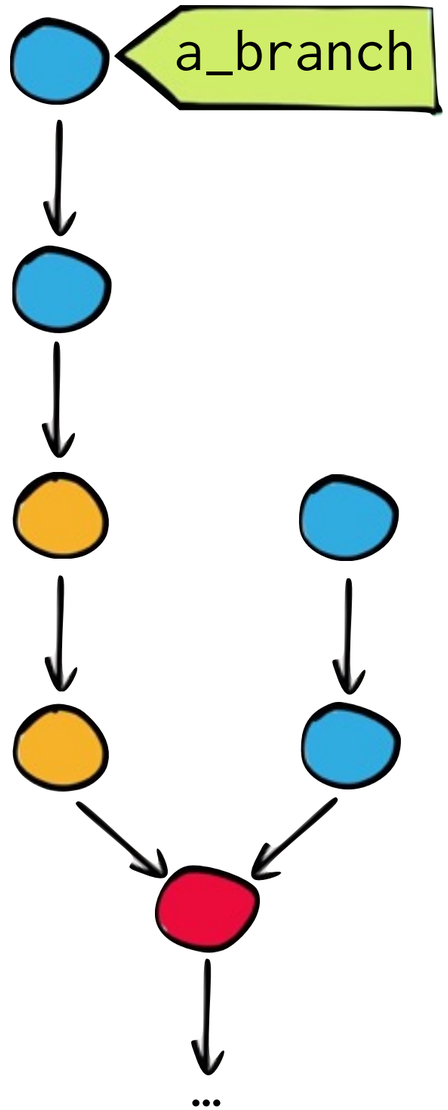

a_branch
...

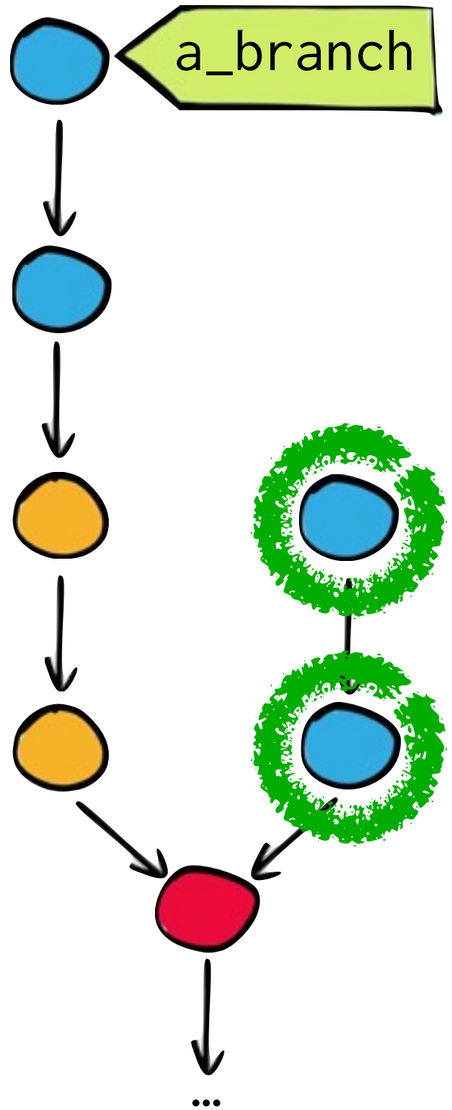

a_branch
...

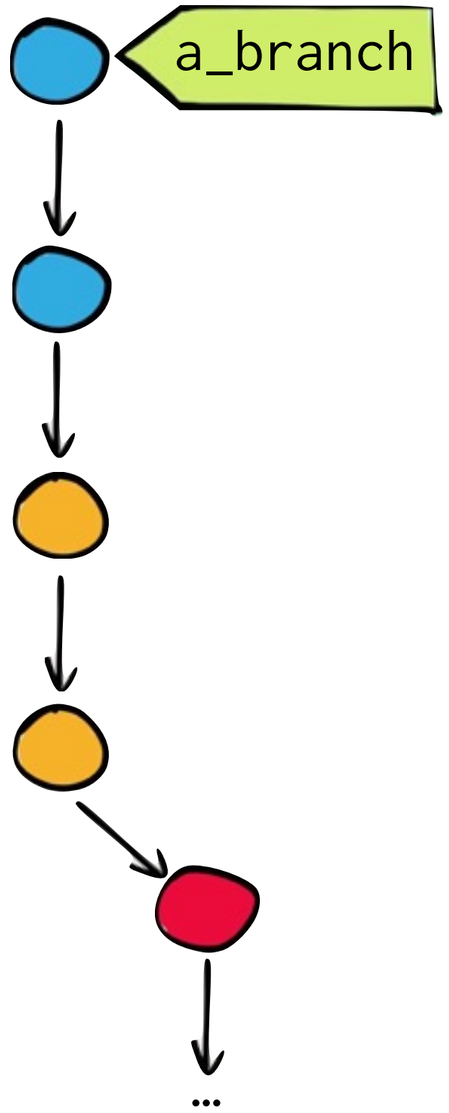

a_branch
...


a_branch
...


a_branch
...

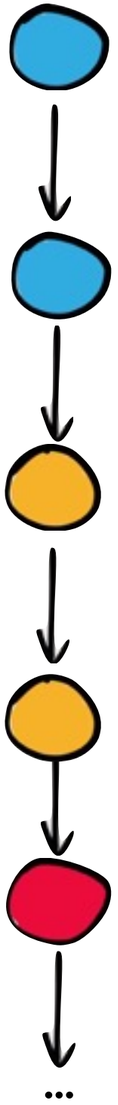

a_branch
...


a_branch
...

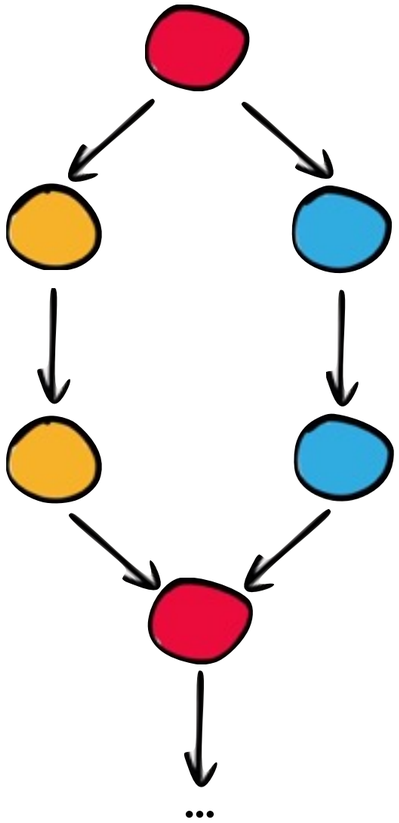

a_branch
...

Git garbage-collects
unreachable objects.

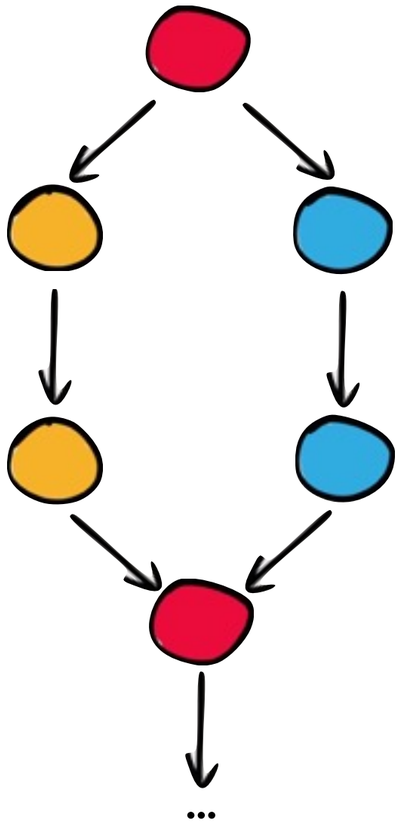
Rebase



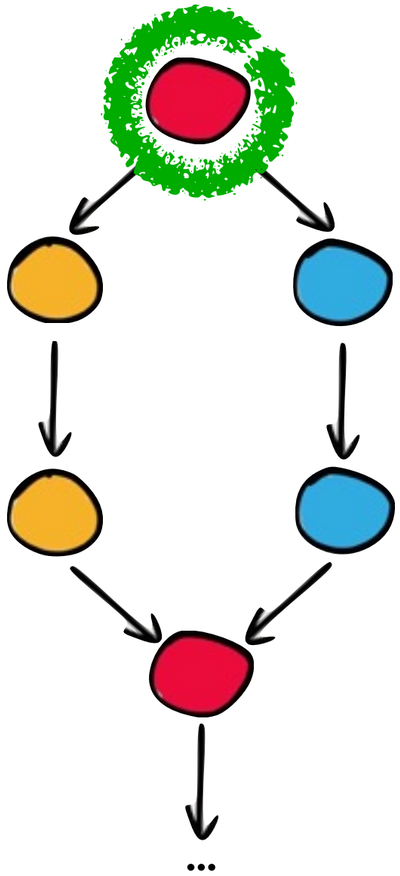
Merge



Merge

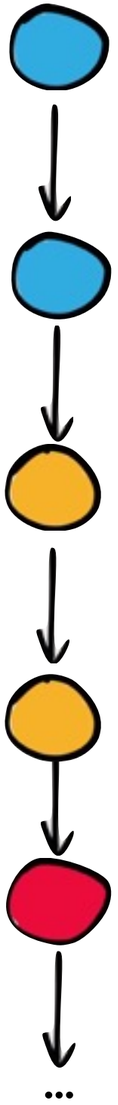


Merge

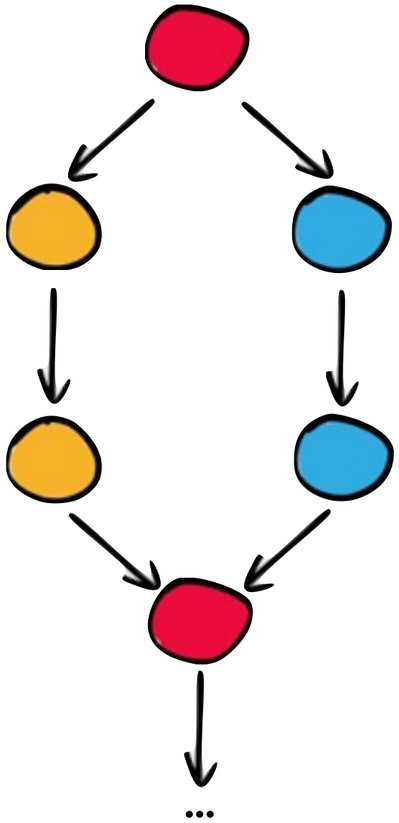


Merges preserve history.

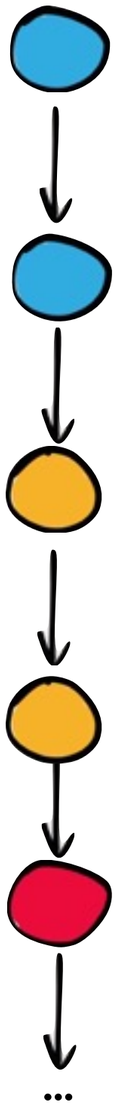
Rebase



Merge



Rebase



Rebases refactor history.

When in doubt, just merge.

Git has two kinds of tags.

A tag is like a
branch that doesn’t move.

Git Is..

…a Stupid Content Tracker

Git Is..

…a Revision Control System