Distributed Version Control For Small Teams And Solo Developers

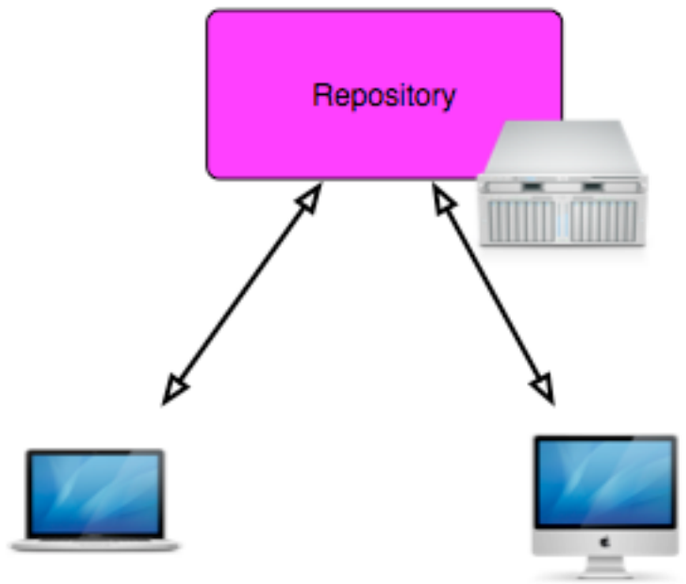
Abizer Nasir

Scope

Why you should consider using a DVCS, some common operations for local and distributed use and links to further information.

Not a recommendation for a particular system, nor a command reference.

Centralised Systems

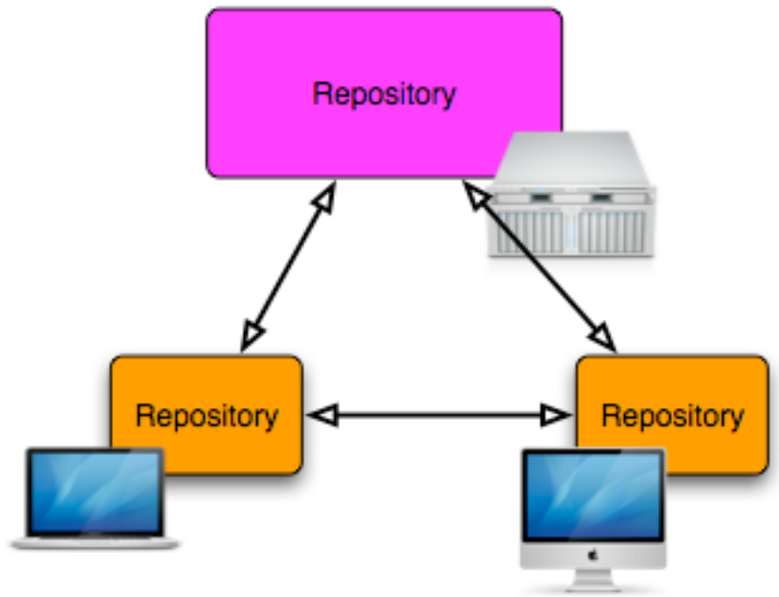


There is only one repository.

All changes are public.

Requires an active connection.

Distributed Systems

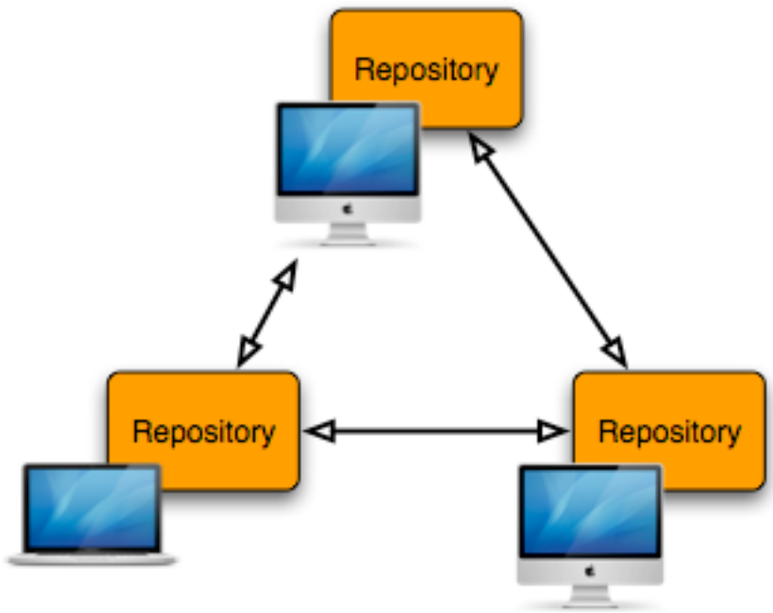


There can be more than one repository.

Not all changes need to be public.

Works offline.

Distributed Systems



There can be more than one repository.

Not all changes need to be public.

Works offline.

No repository needs to be canonical

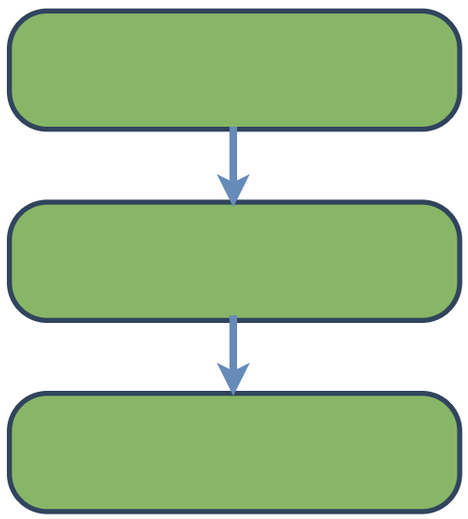
Repositories don’t need to be exact copies.

Version Control

- Store snapshots of a project.
- See the differences between these snapshots.
- Maintain multiple lines of development.
- Roll back changes to a previous snapshot.
- Work with other developers.
- Distributed - deal with other repositories.

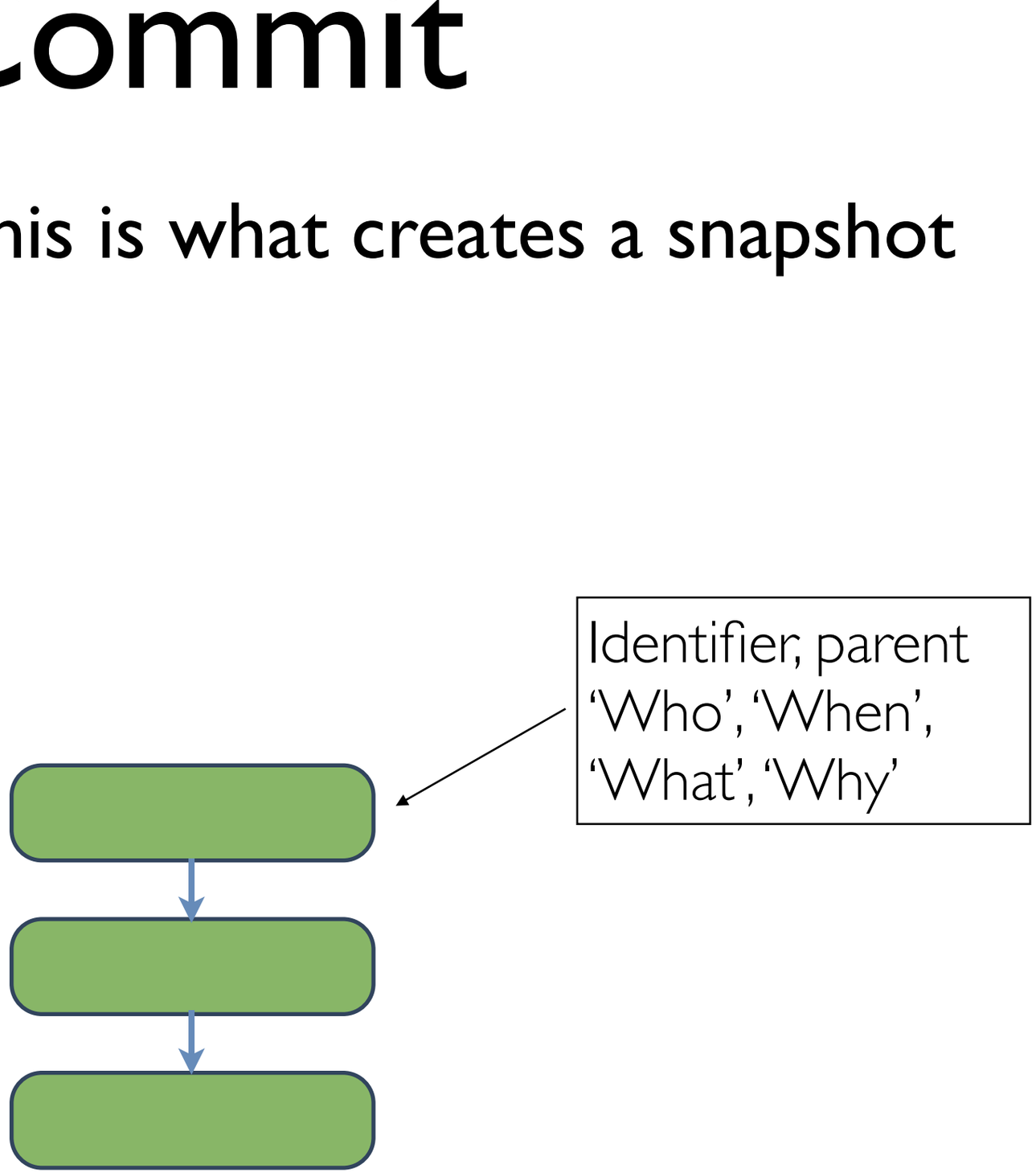
Commit

This is what creates a snapshot



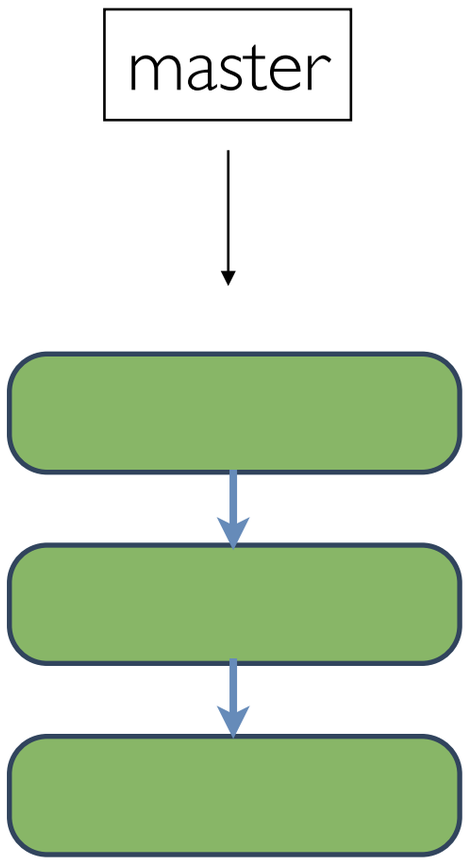
Commit

This is what creates a snapshot



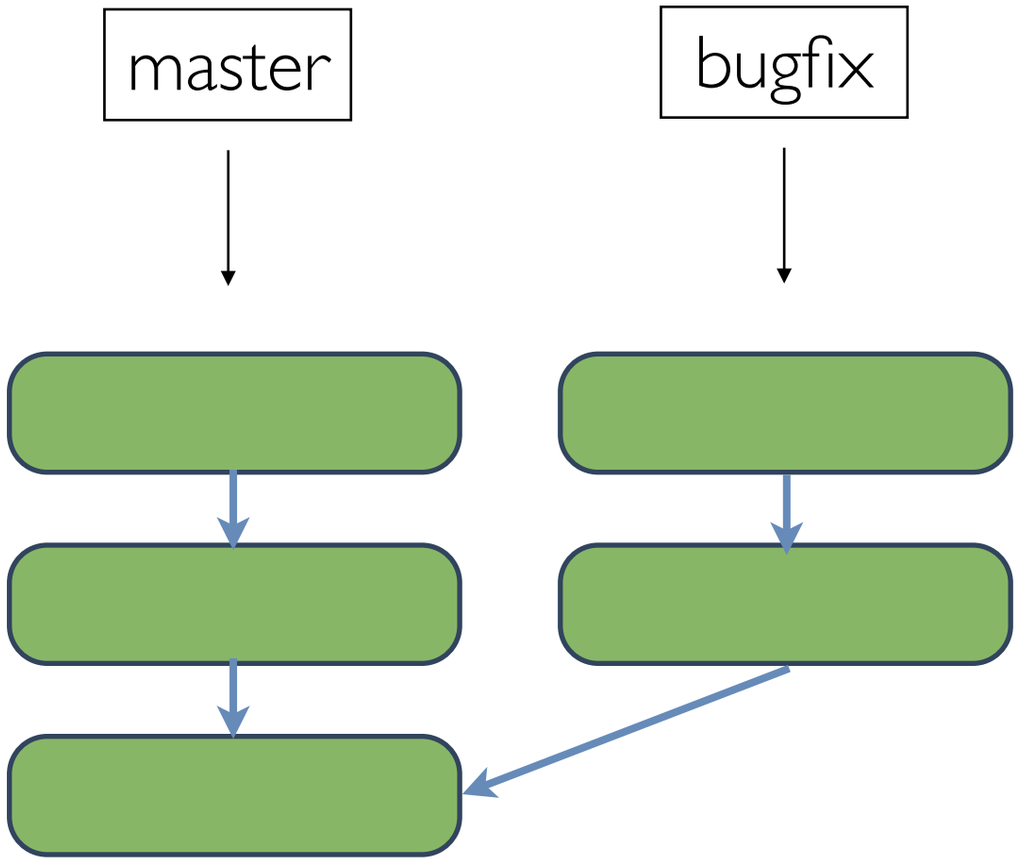
Branches

Multiple lines of history



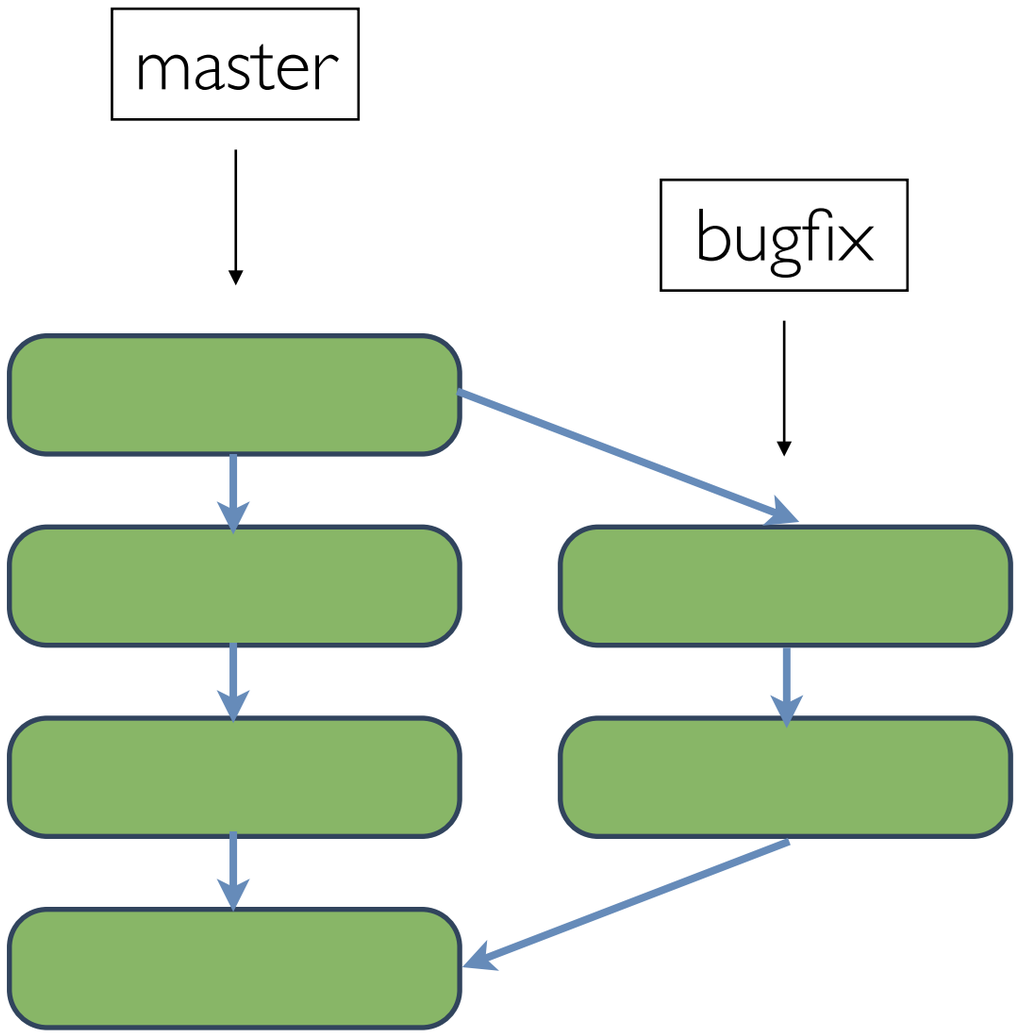
Branches

Multiple lines of history



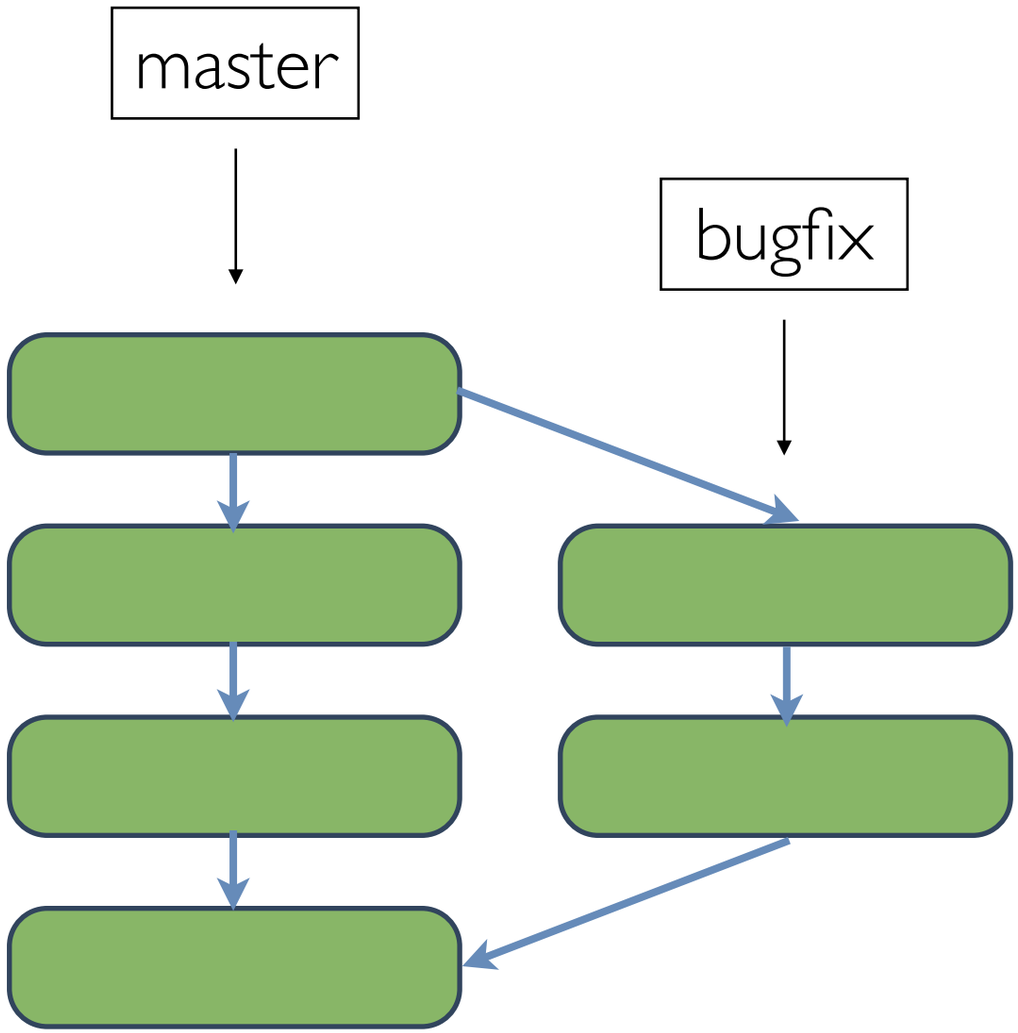
Merging

Pulling changes back together



Merging

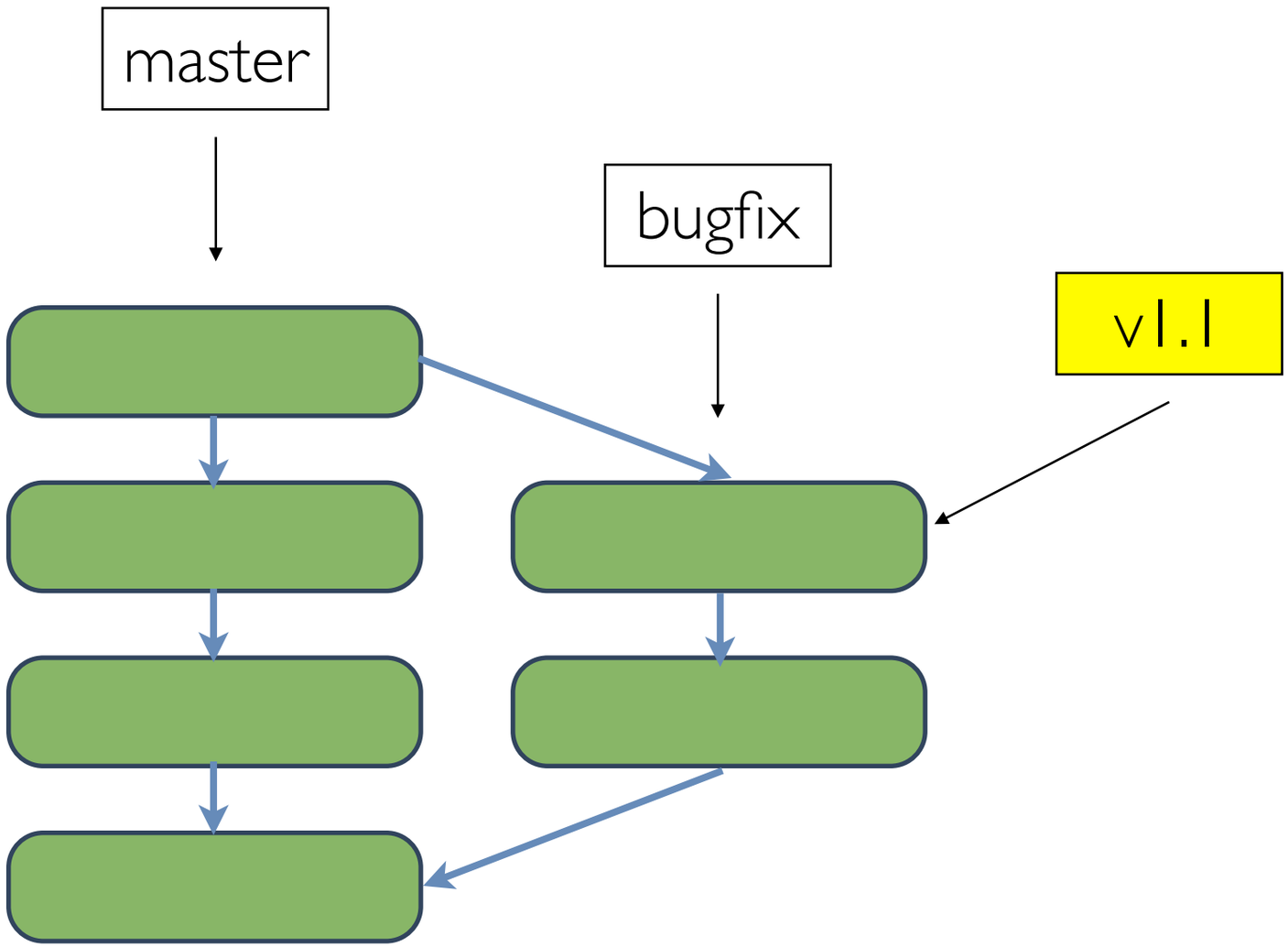
Pulling changes back together



The branch remains until you get rid of it

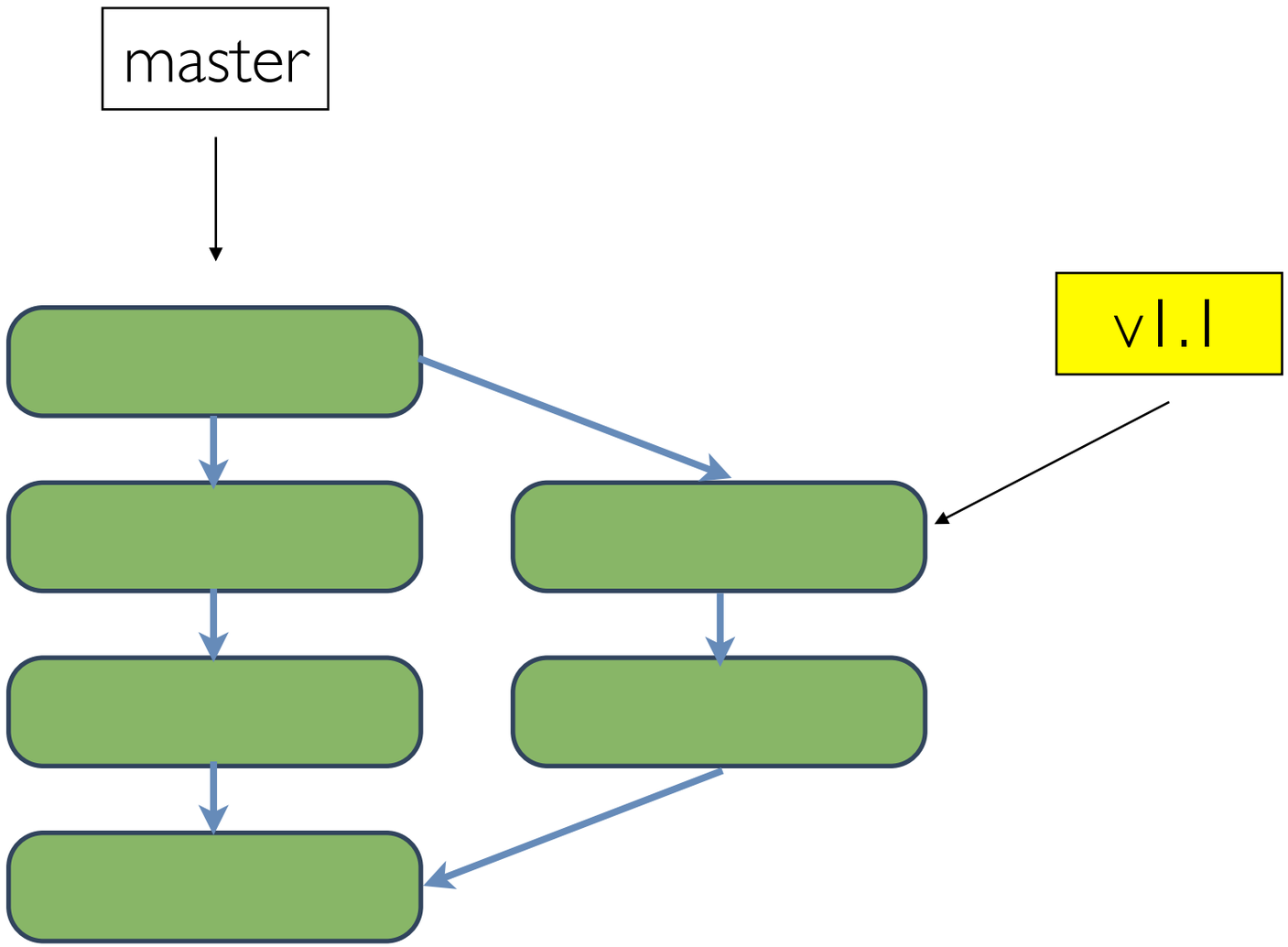
Tags

A simple way to refer to a change



Tags

A simple way to refer to a change



Pushing

- Propagating changes to other repositories.
- Don’t have to worry about making all changes public.
- Don’t change history once a set of changes is public.
- There are online services that take away the pain of managing a public repository.

Pulling

- Adding the changes in public repositories to your local one.
- Merge can be automatic or manual.
- Conflicts are bound to happen, but relatively easy to deal with.

Disadvantages

- Repositories can and do get out of sync.
- Merge conflicts can be difficult to deal with.
- There is a lot to learn.
- Graphical interfaces are still maturing.

Git

- Written in C
- Fast, powerful, but can be confusing.
- GitHub!

Mercurial

- Written in Python
- Fast
- Fewer commands to learn, but can be just as functional with add-ons
- BitBucket

Pick a system that works with your own workflow. Learn it well, and get on with solving problems

Use Git

@abizern

365git.tumblr.com