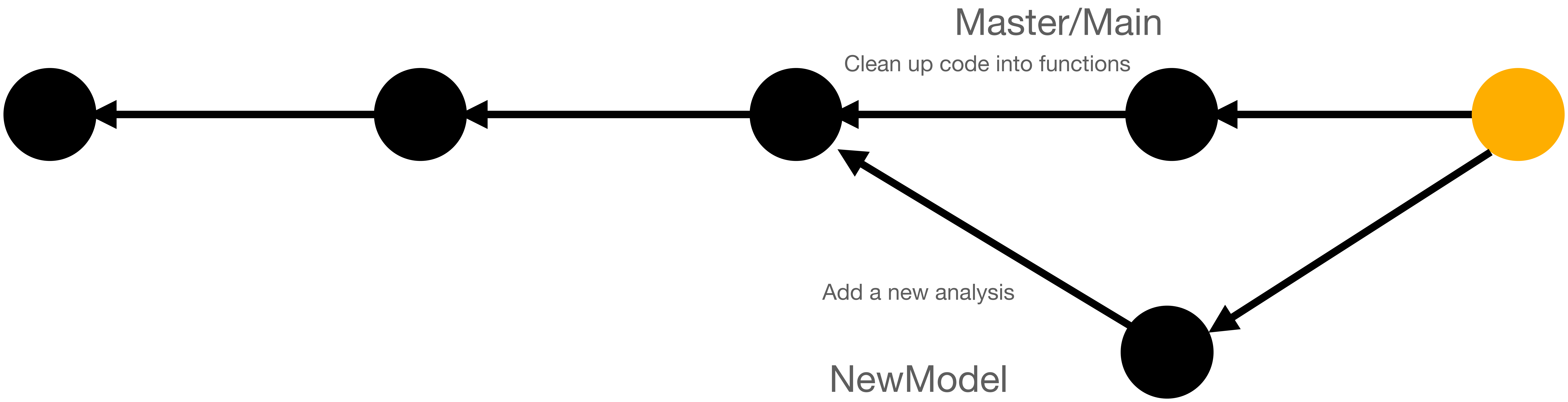
The github-bub

Sharing projects and working on shared projects

Michael Colaresi

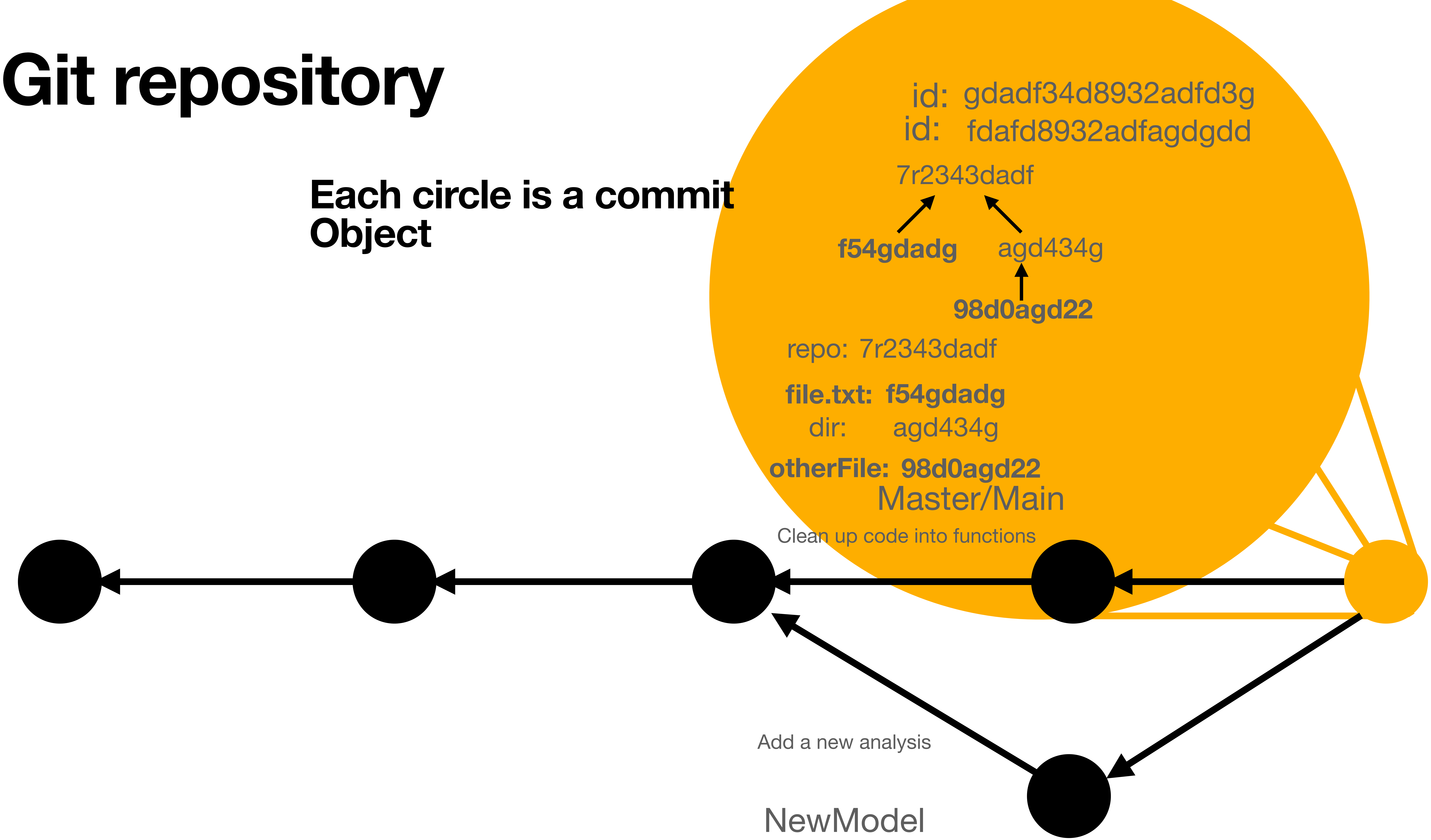
Git repository

- Holds snapshots of information
  - Organized into trees (directories/paths) and blobs (file contents)
  - SHA1 hash used to fingerprint information
  - A commit object is a snapshot of information across files within directories



Git repository

Each circle is a commit Object



id: gdadf34d8932adfd3g
id: fdafd8932adfagdgdd

7r2343dadf

f54gdadg agd434g

98d0agd22

repo: 7r2343dadf

file.txt: f54gdadg
dir: agd434g
otherFile: 98d0agd22

Master/Main

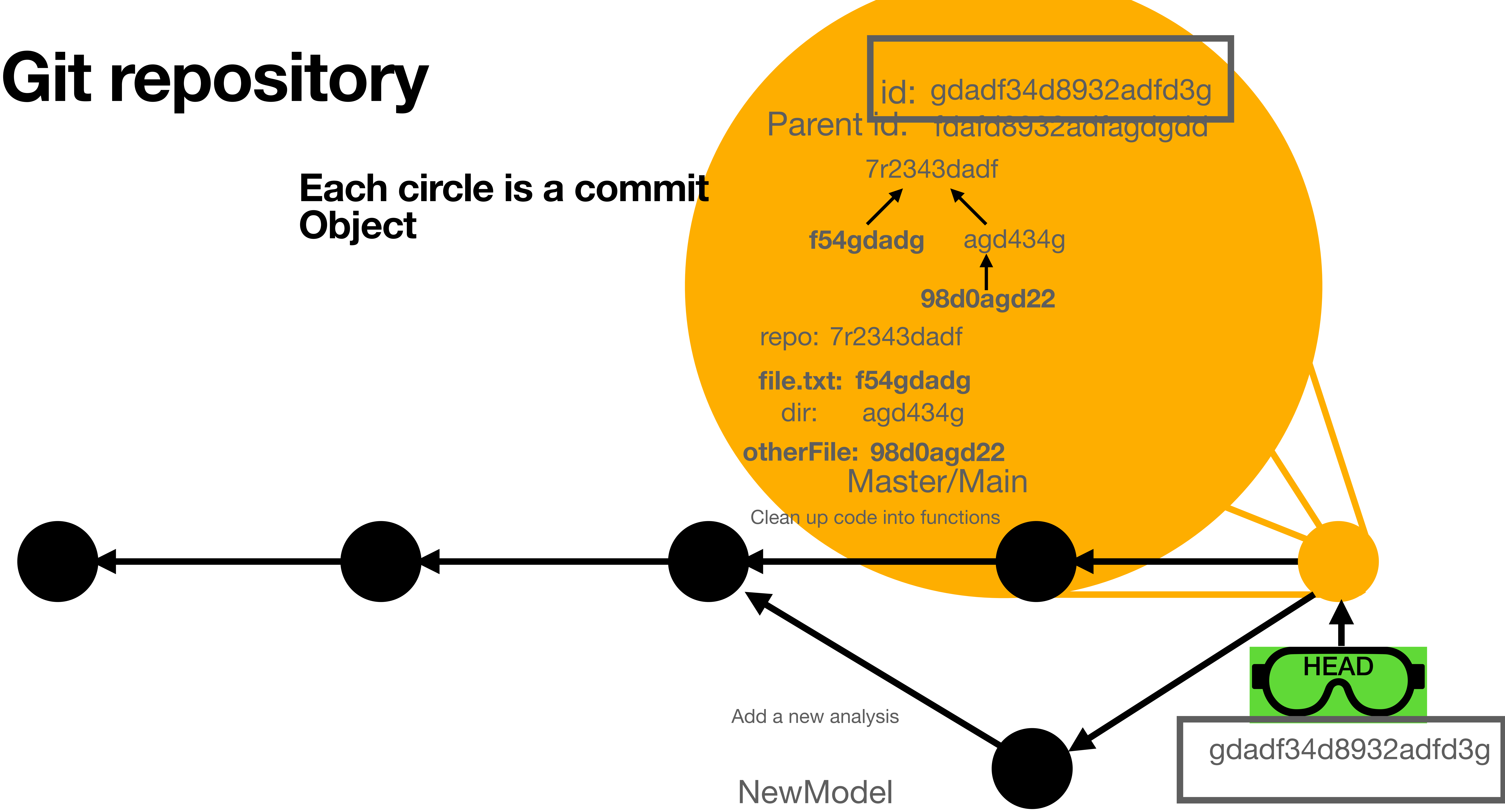
Clean up code into functions

Add a new analysis

NewModel

Git repository

Each circle is a commit Object



id: gdadf34d8932adfd3g

Parent id: fdafd8932adfagdgdd

7r2343dadf

f54gdadg agd434g

98d0agd22

repo: 7r2343dadf

file.txt: f54gdadg

dir: agd434g

otherFile: 98d0agd22

Master/Main

Clean up code into functions

HEAD

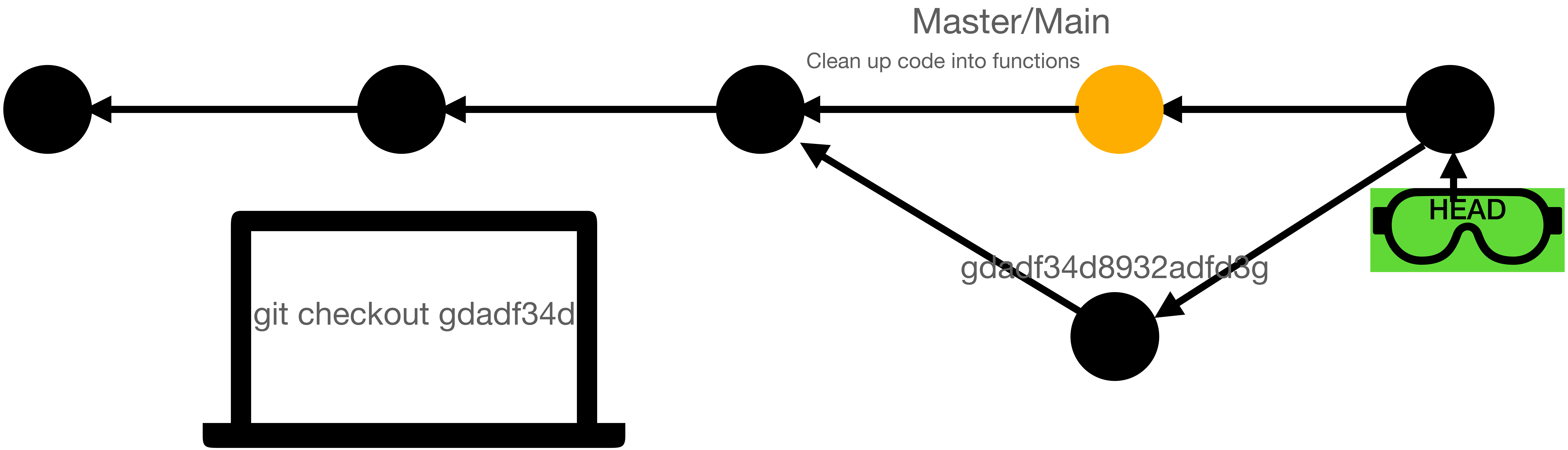
gdadf34d8932adfd3g

Add a new analysis

NewModel

Git repository

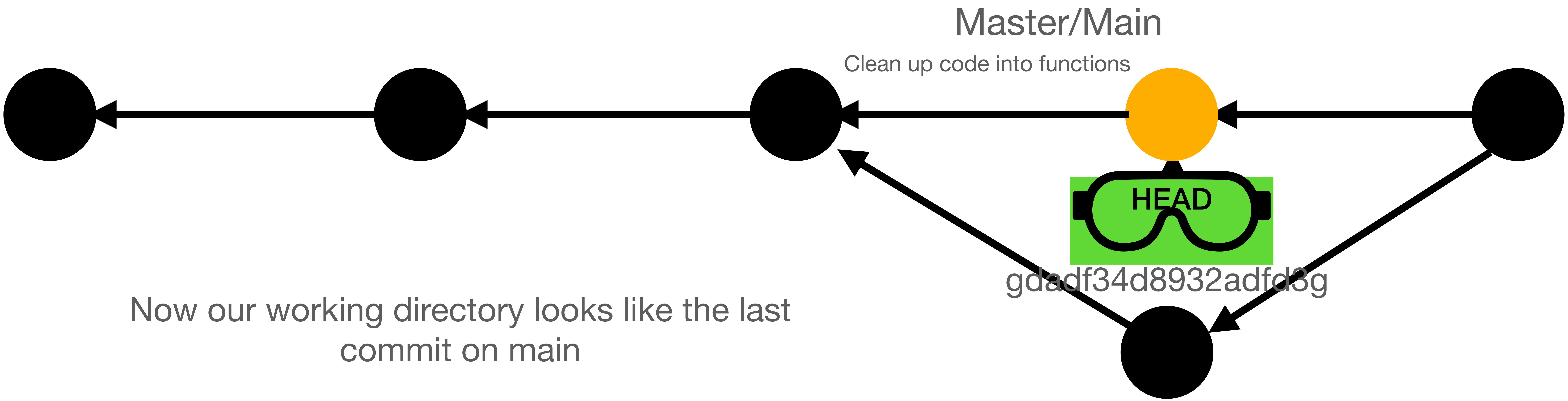
Remember we can move between commits!



```
git checkout gdadf34d
```

Git repository

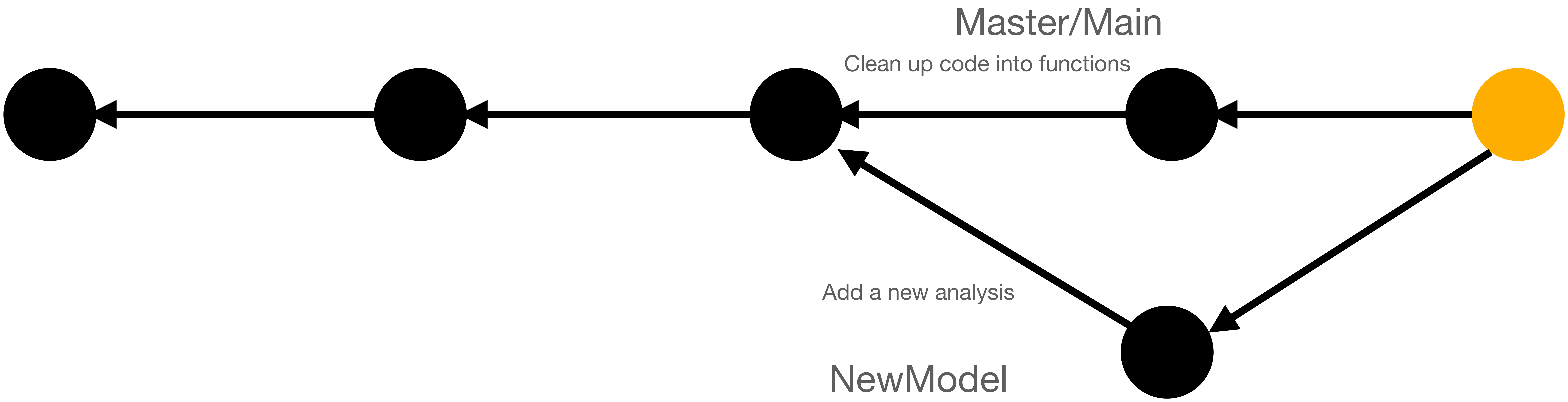
Each circle is a commit Object



Now our working directory looks like the last commit on main

Git is a DISTRIBUTED version control system

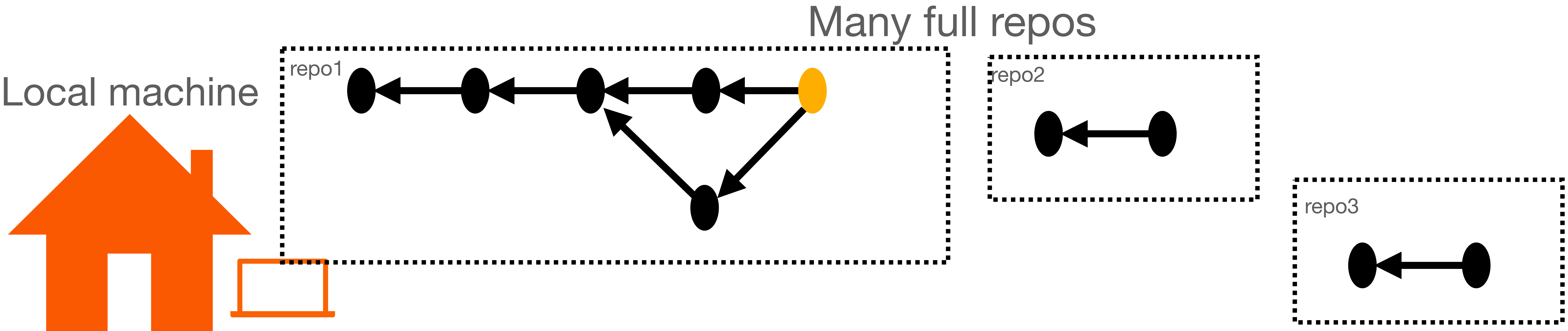
- Every local or remote repository is a FULL repository
  - The work is DISTRIBUTED around to many systems (lots of copies)
  - There is not one centralized repository but MANY!



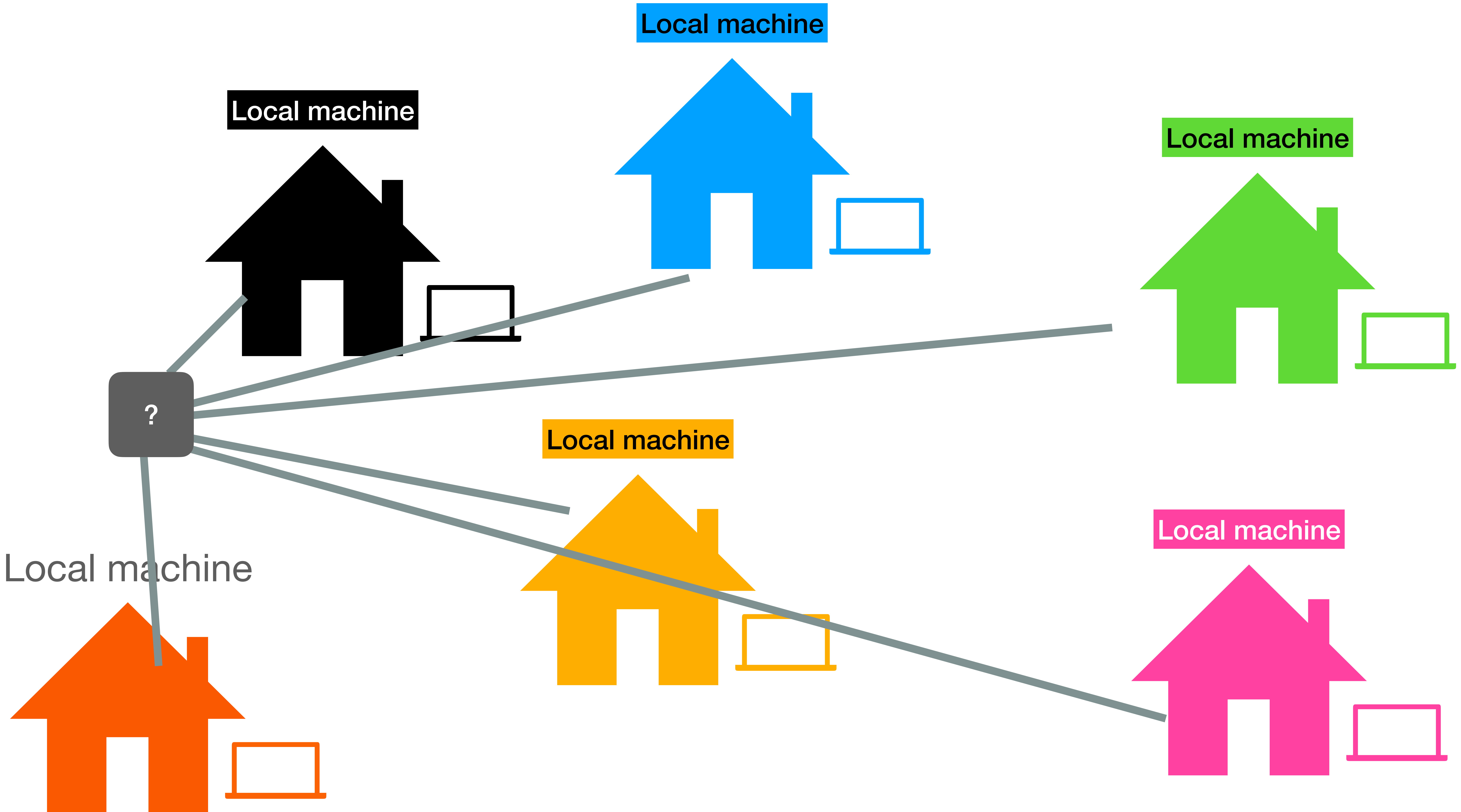
Your local machine can hold many repos

Local machine

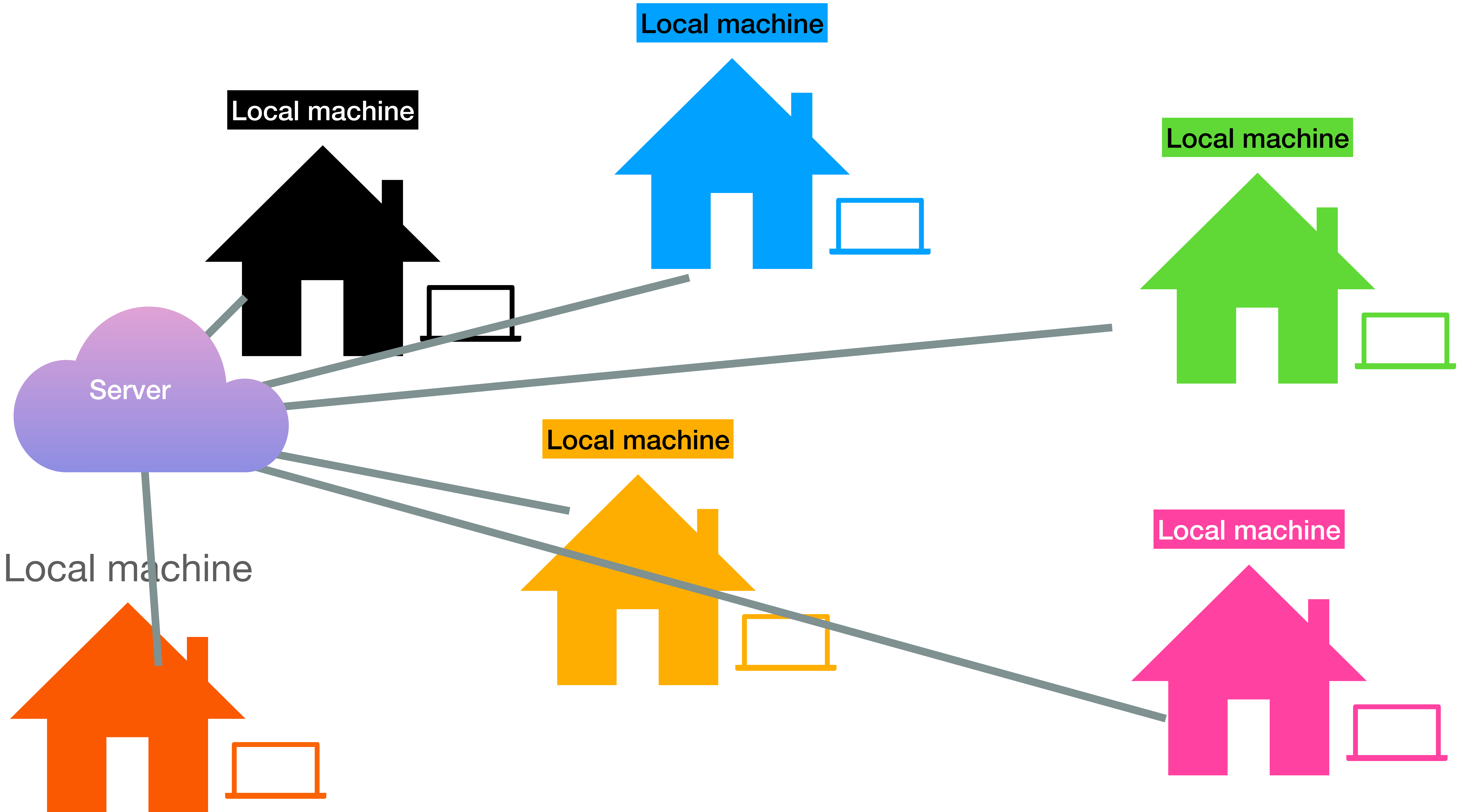
Your local machine can hold many repos



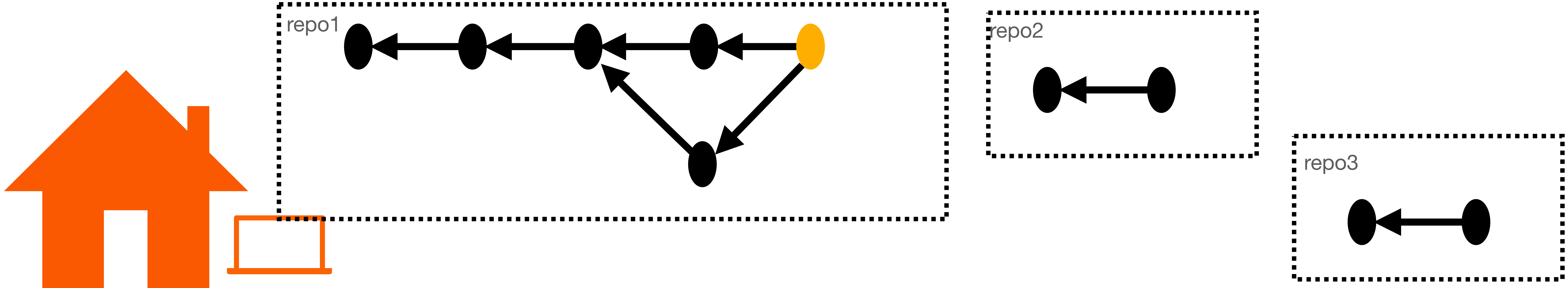
+ local machine can share code with others



Remote servers “serve” as connections

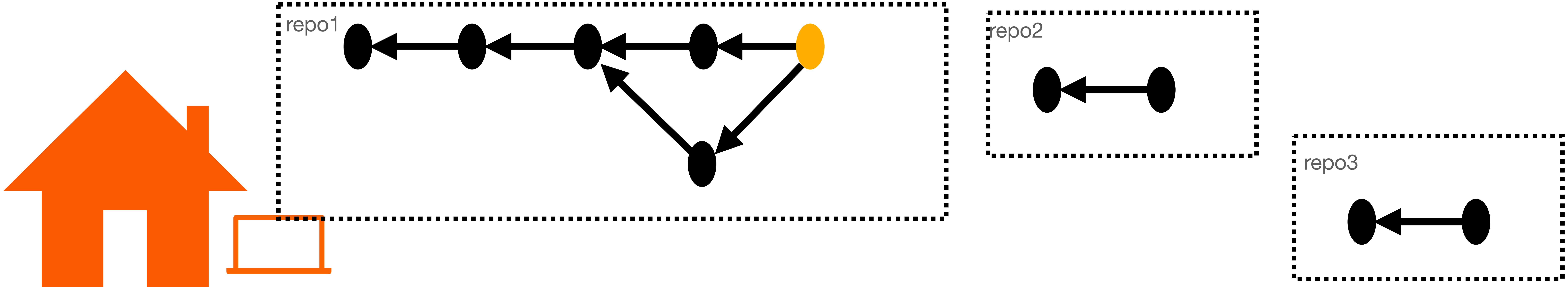


REMOTE repositories are on servers (in the cloud)
That can be accessed through https, ssh, or other protocols



Github

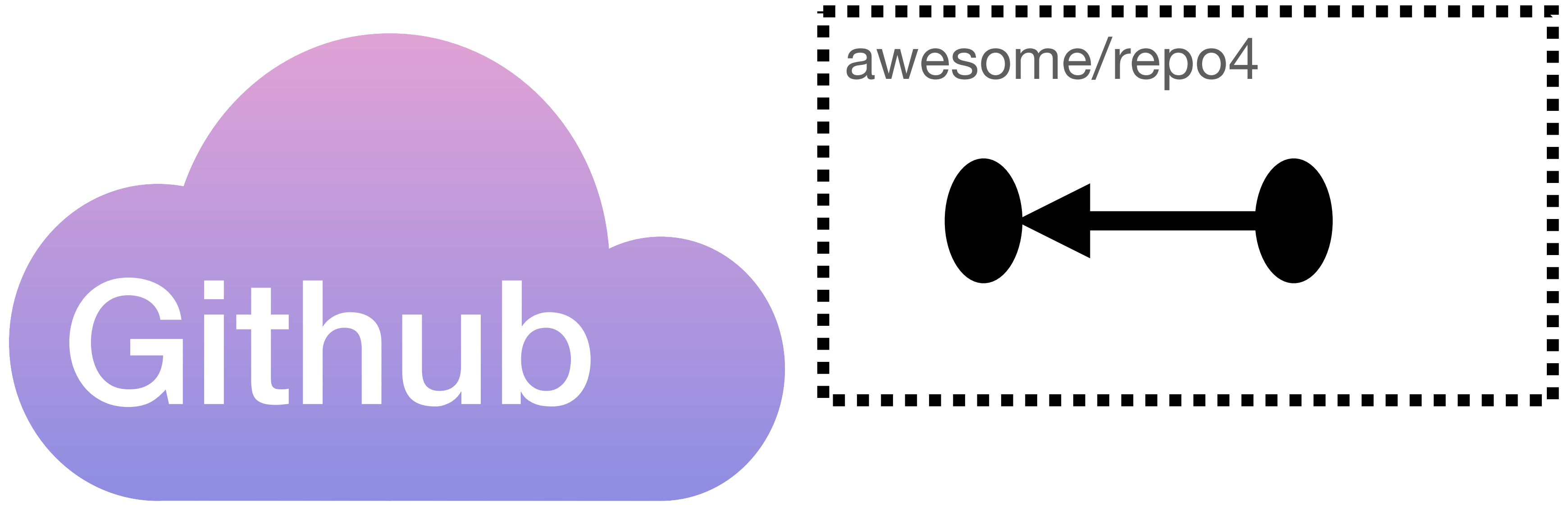
Like Github!



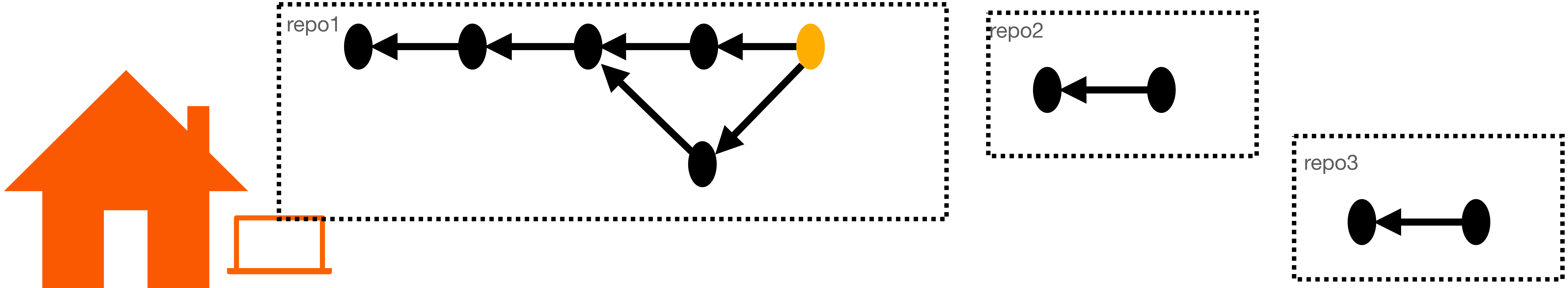
repo1

repo2

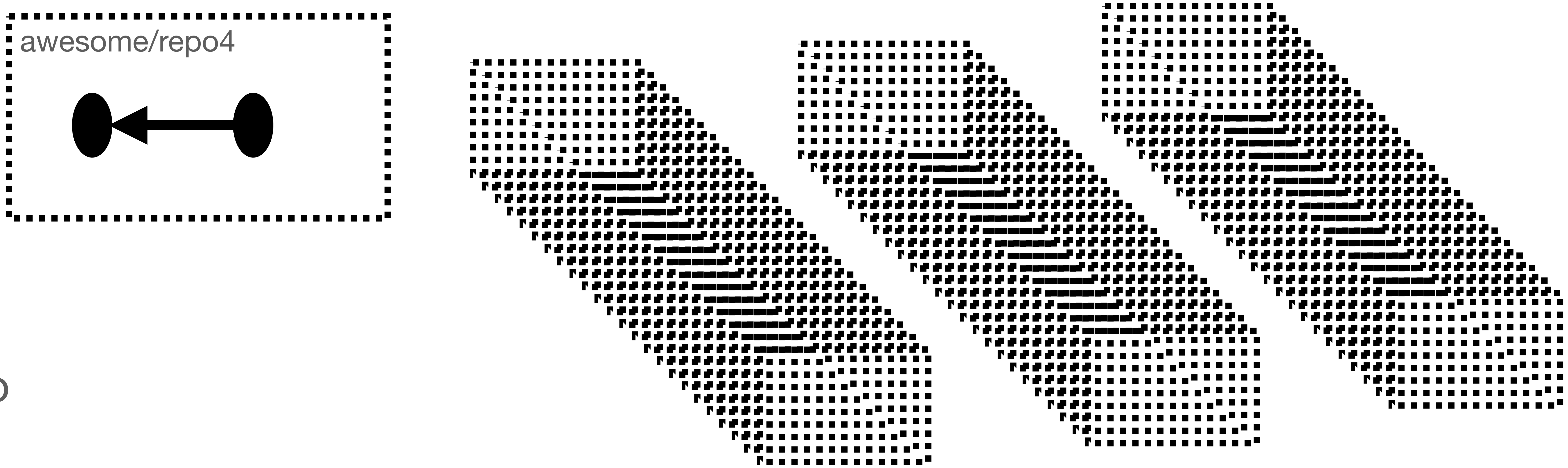
repo3

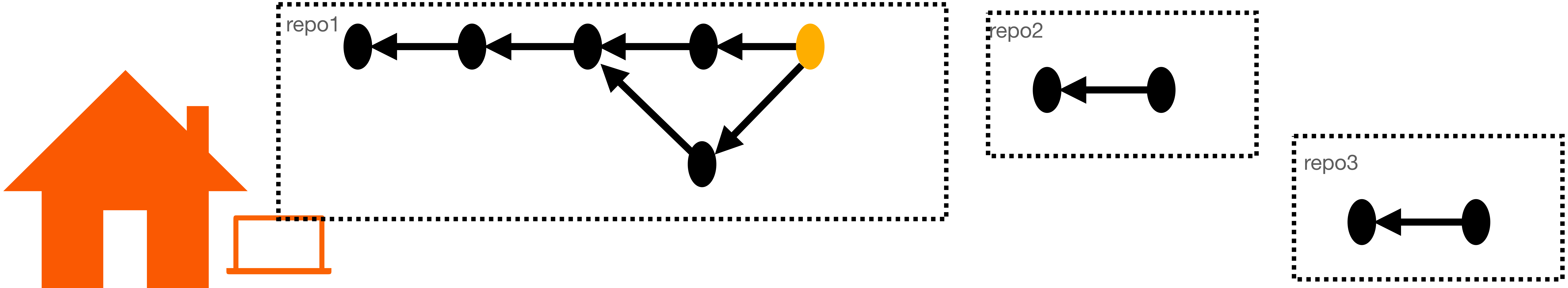


REMOTE repositories are on servers (in the cloud)
That can be accessed through https, ssh, or other protocols



Repositories
Can be on servers too
(in the cloud)



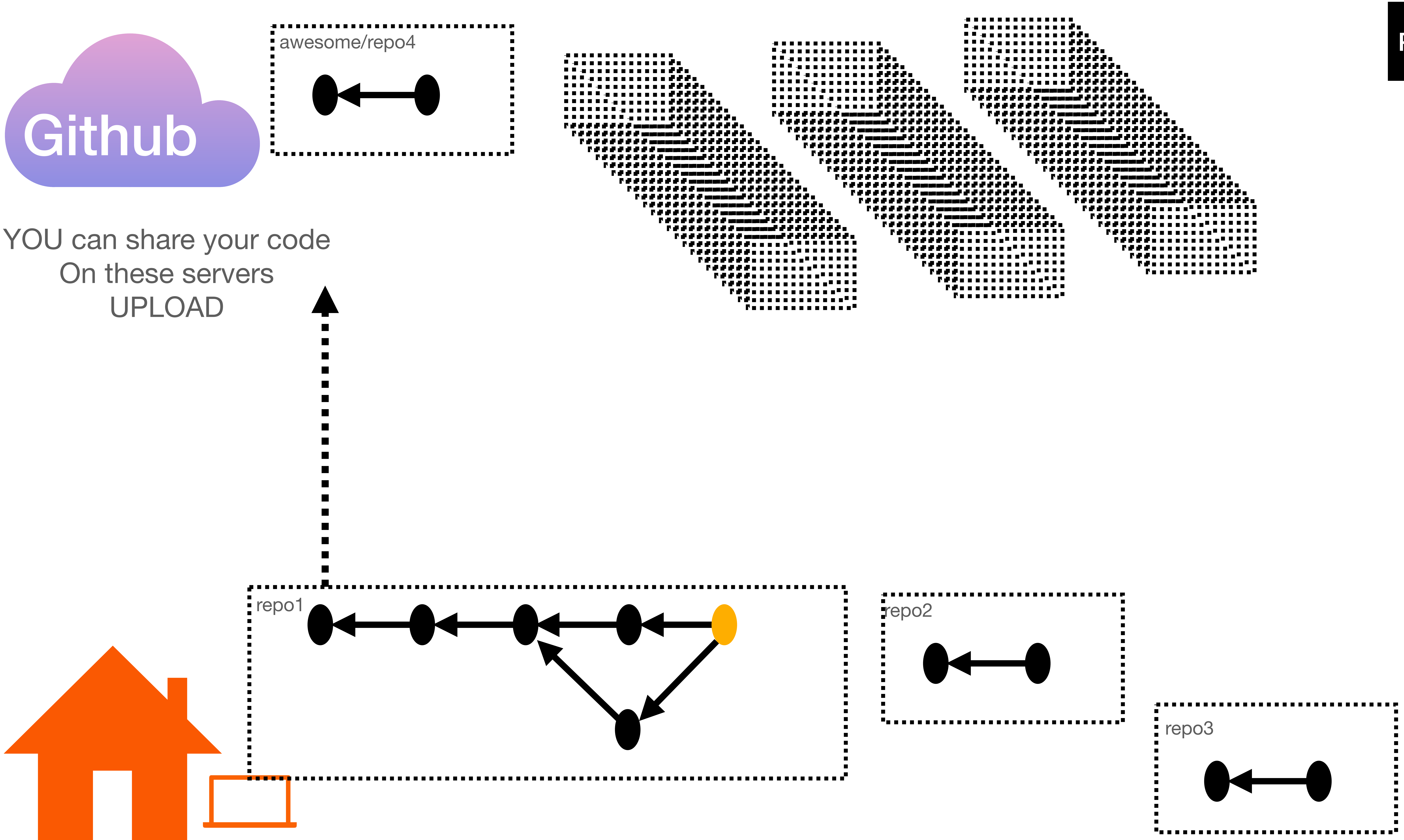


Github
awesome/repo4
bitbucket
Gitlab instances
repo1
repo2
repo3

Repositories
Can be on servers too
(in the cloud)

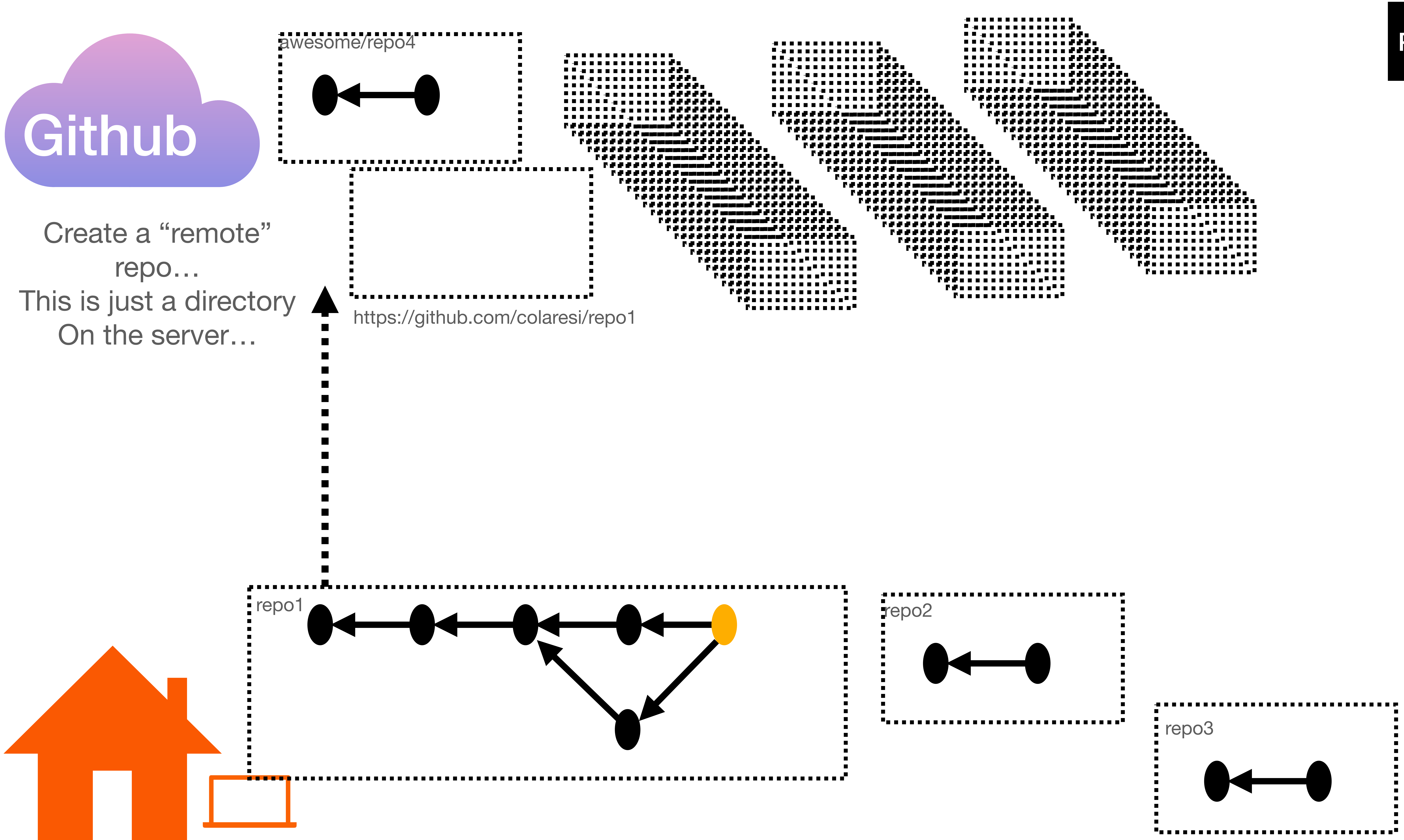
There are other popular remote servers too…

Push



YOU can share your code
On these servers
UPLOAD

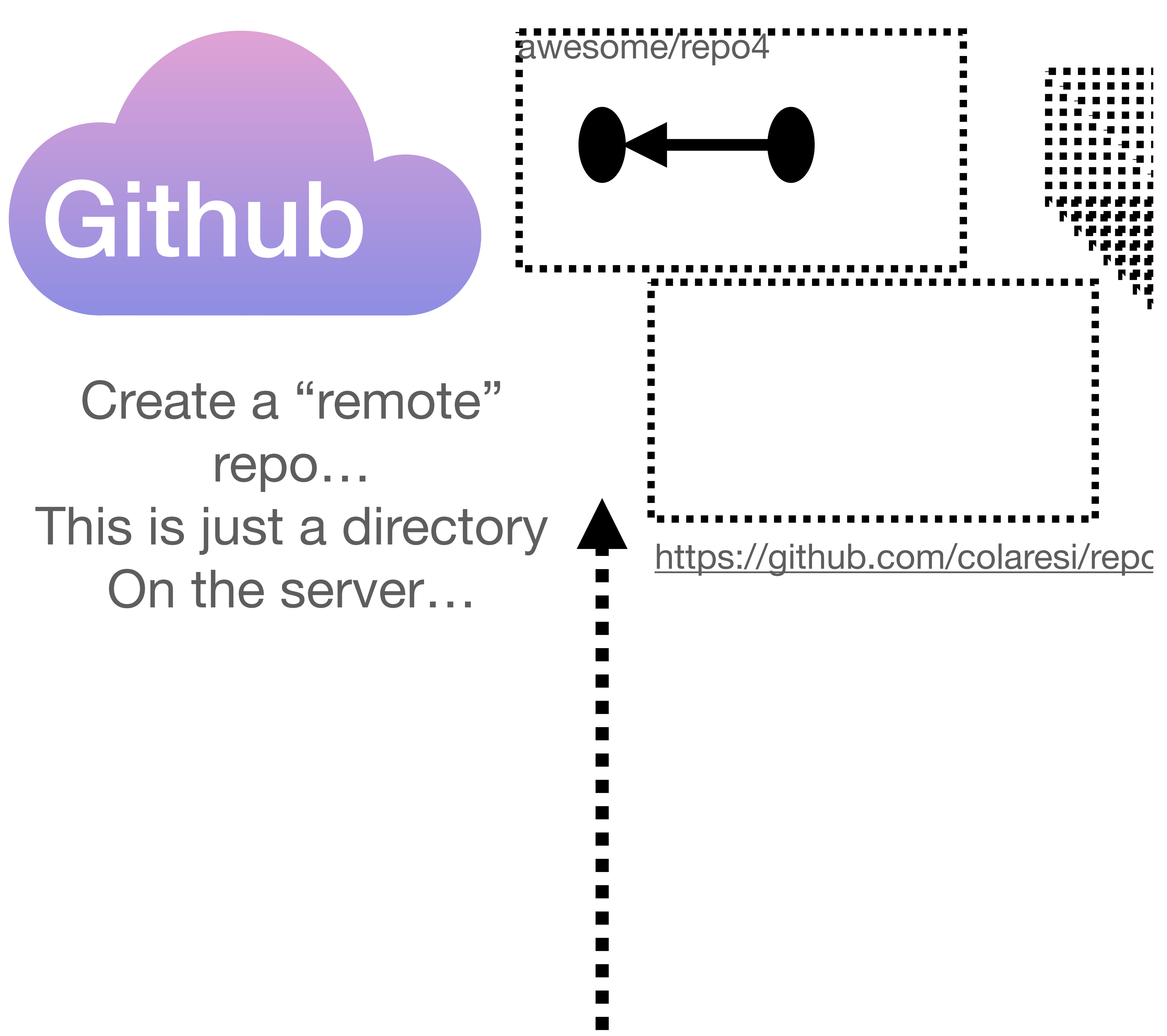
Push



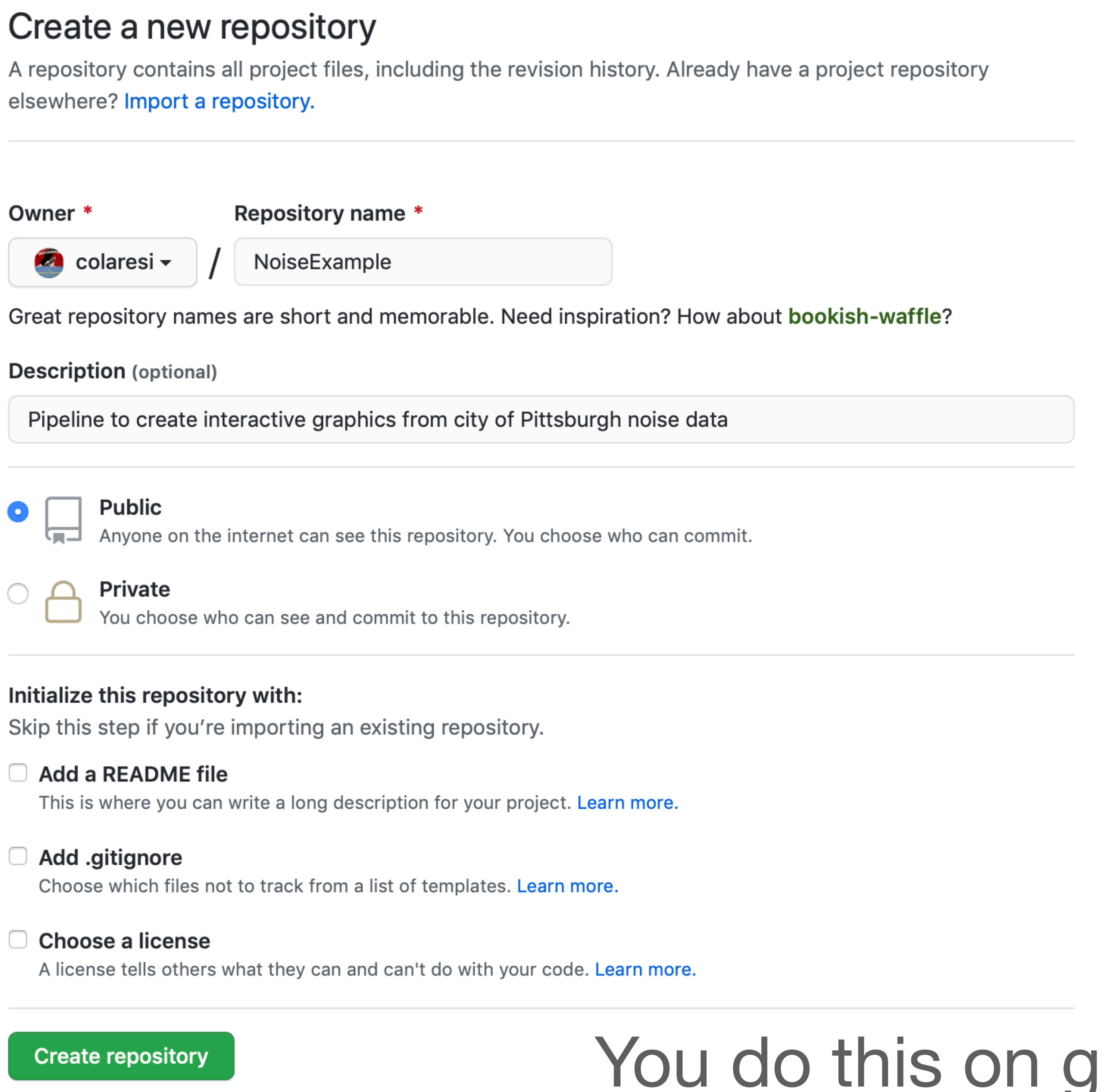
Create a “remote” repo…
This is just a directory
On the server…

https://github.com/colaresi/repo1

Push



Create a “remote” repo…
This is just a directory
On the server…



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Repository name *

colaresi ▾ / NoiseExample

Great repository names are short and memorable. Need inspiration? How about bookish-waffle?

Description (optional)

Pipeline to create interactive graphics from city of Pittsburgh noise data

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

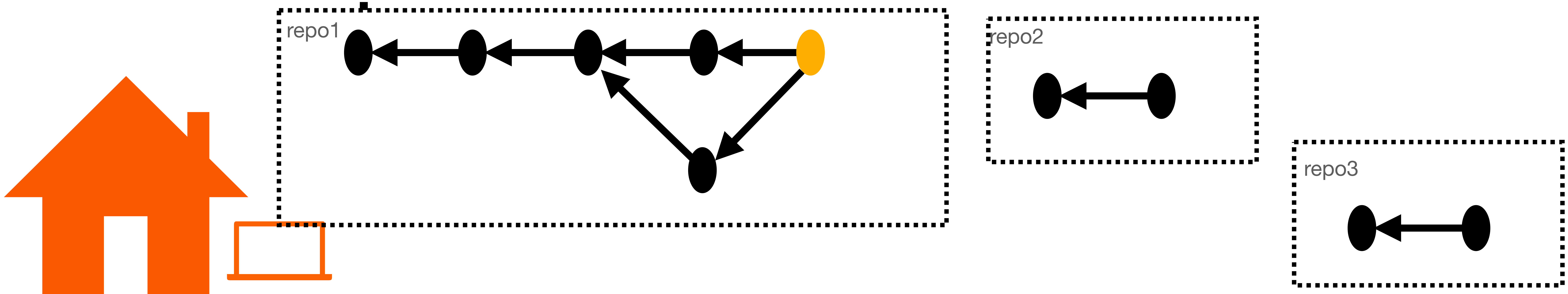
Initialize this repository with:
Skip this step if you’re importing an existing repository.

□ Add a README file
This is where you can write a long description for your project. Learn more.

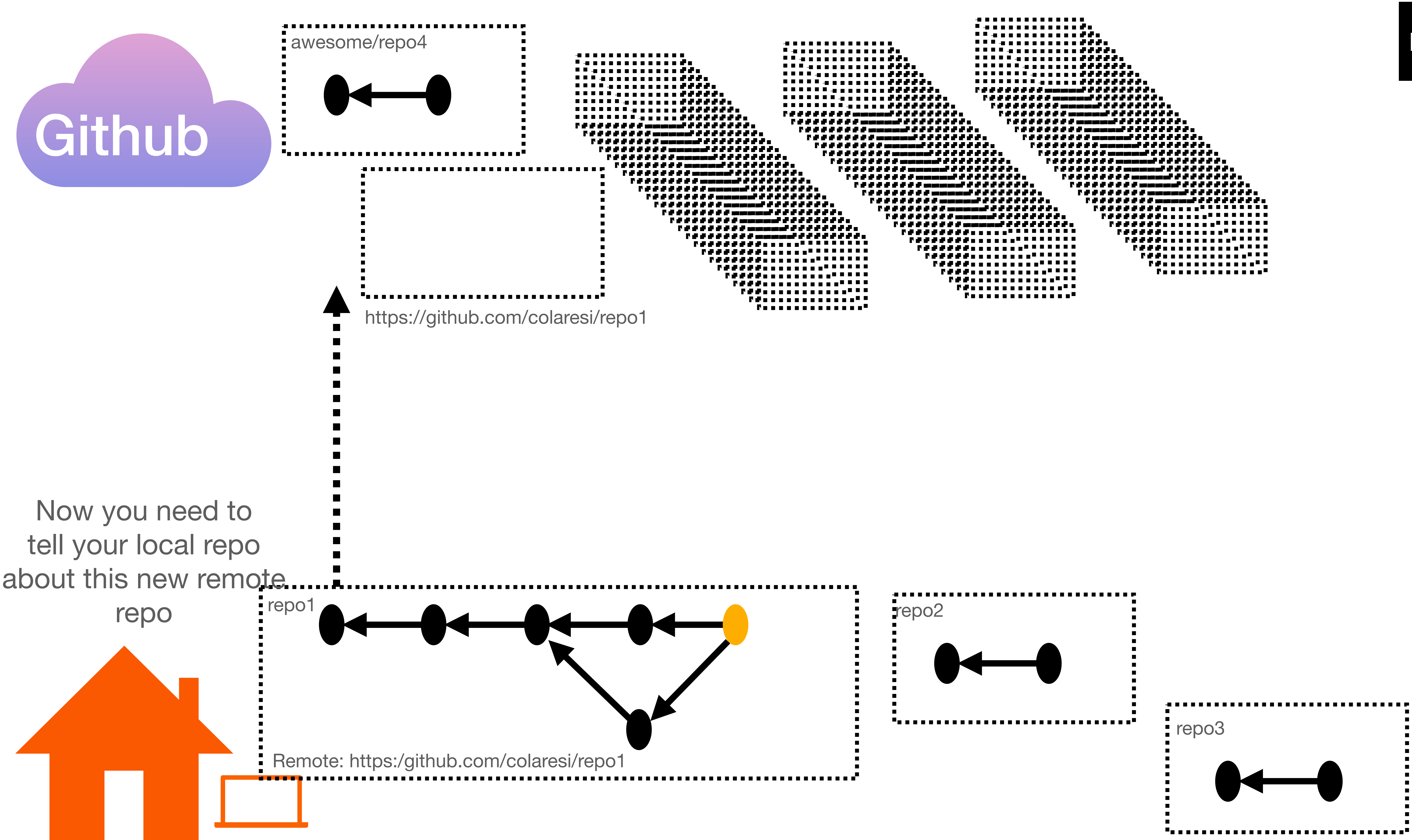
□ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

□ Choose a license
A license tells others what they can and can't do with your code. Learn more.

You do this on github
For example, and it is looks like this

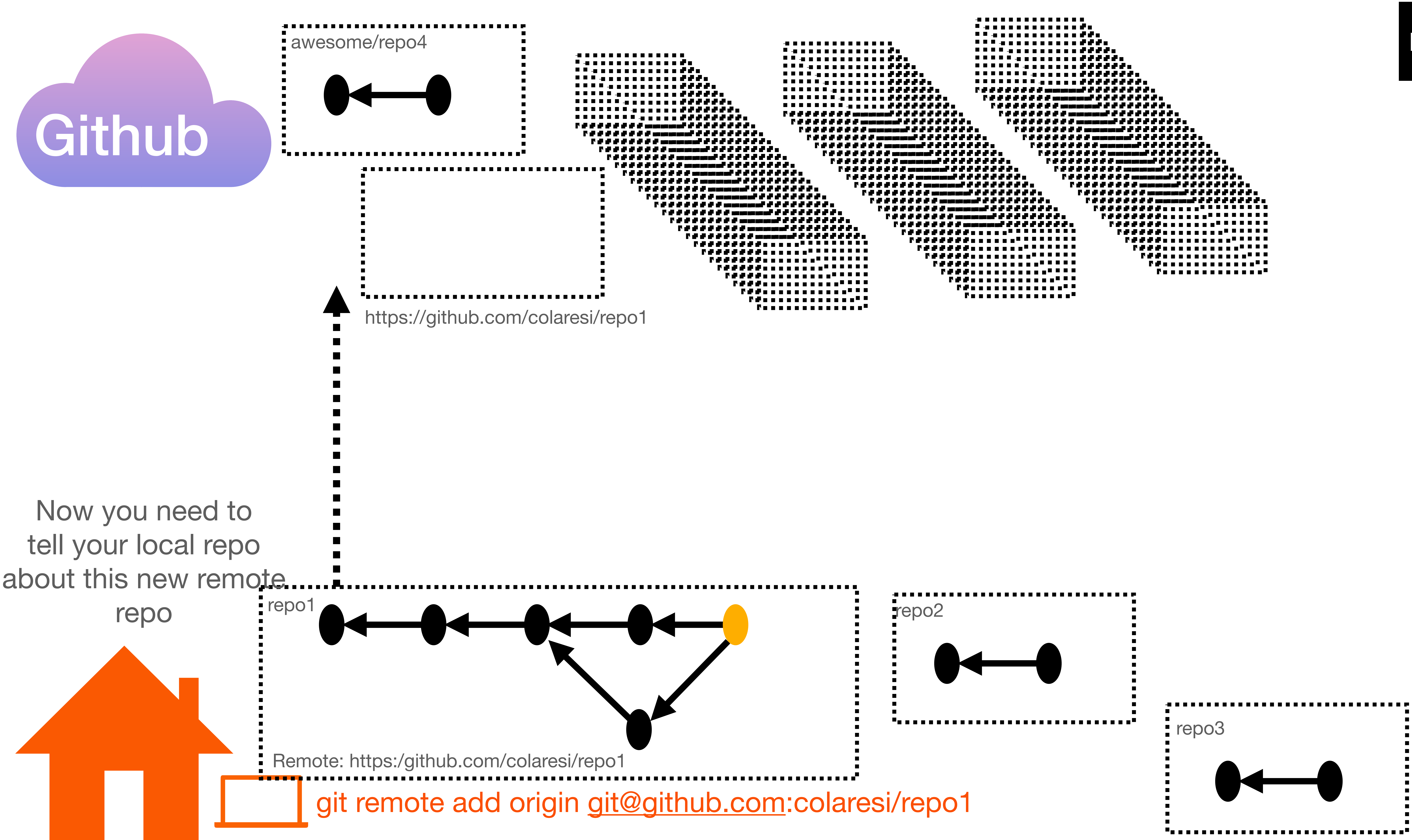


Push


Github
awesome/repo4
https://github.com/colaresi/repo1
repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo3

Now you need to tell your local repo about this new remote repo

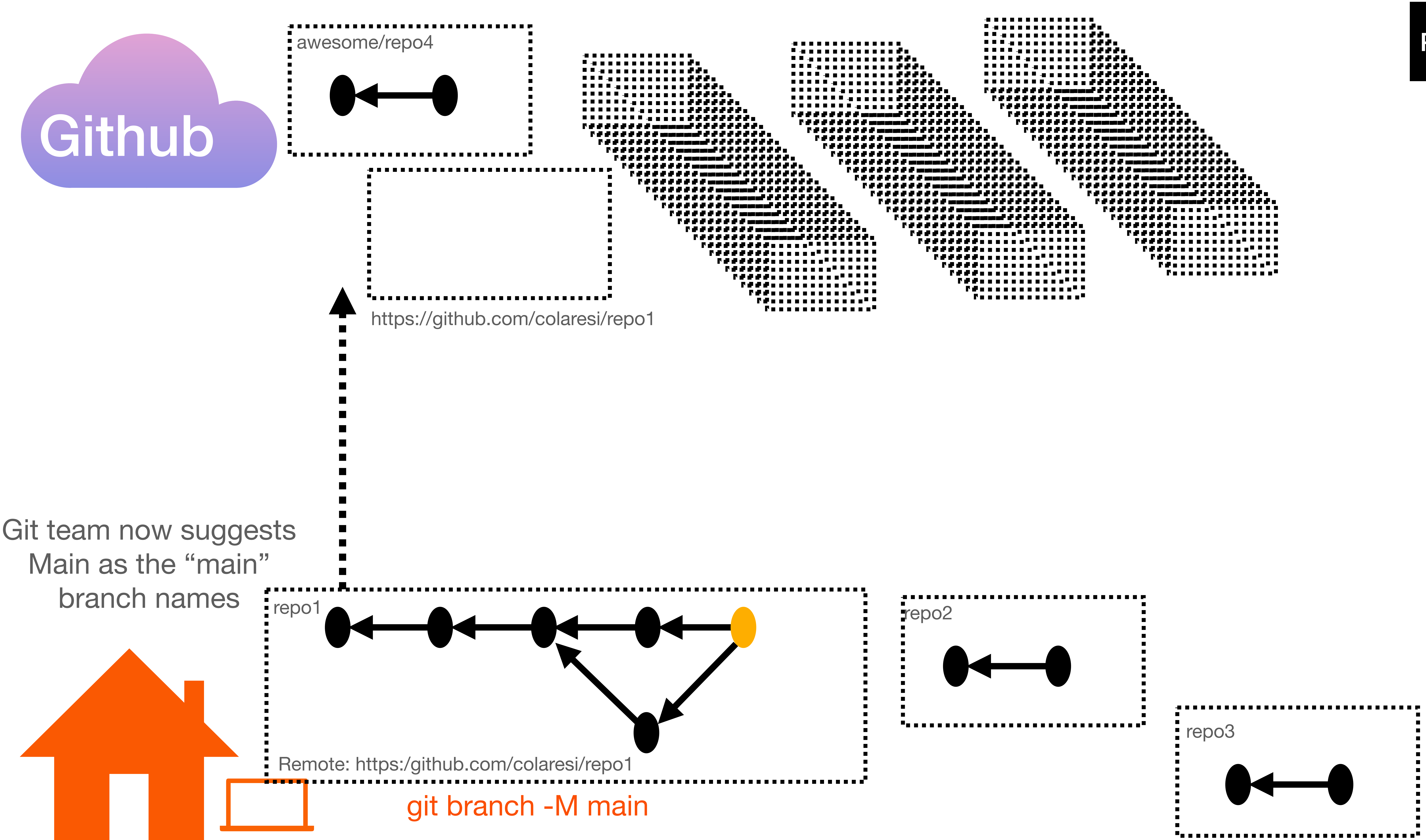
Push


Github
awesome/repo4
https://github.com/colaresi/repo1
repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo3

Now you need to tell your local repo about this new remote repo

`git remote add origin git@github.com:colaresi/repo1`

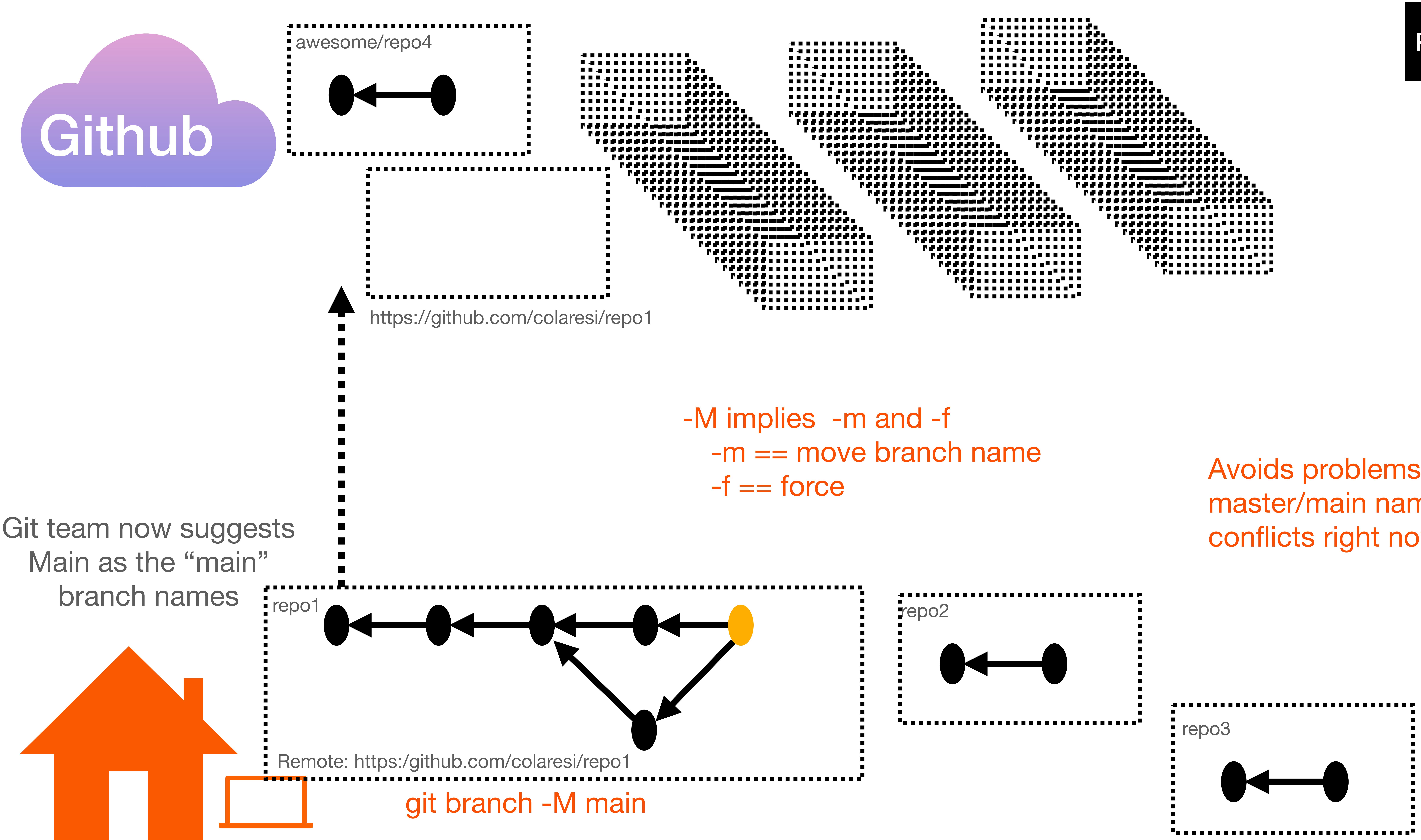
Push


Github
awesome/repo4
https://github.com/colaresi/repo1
repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo3

Git team now suggests Main as the “main” branch names

git branch -M main

Push


Github
awesome/repo4
https://github.com/colaresi/repo1
repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo3

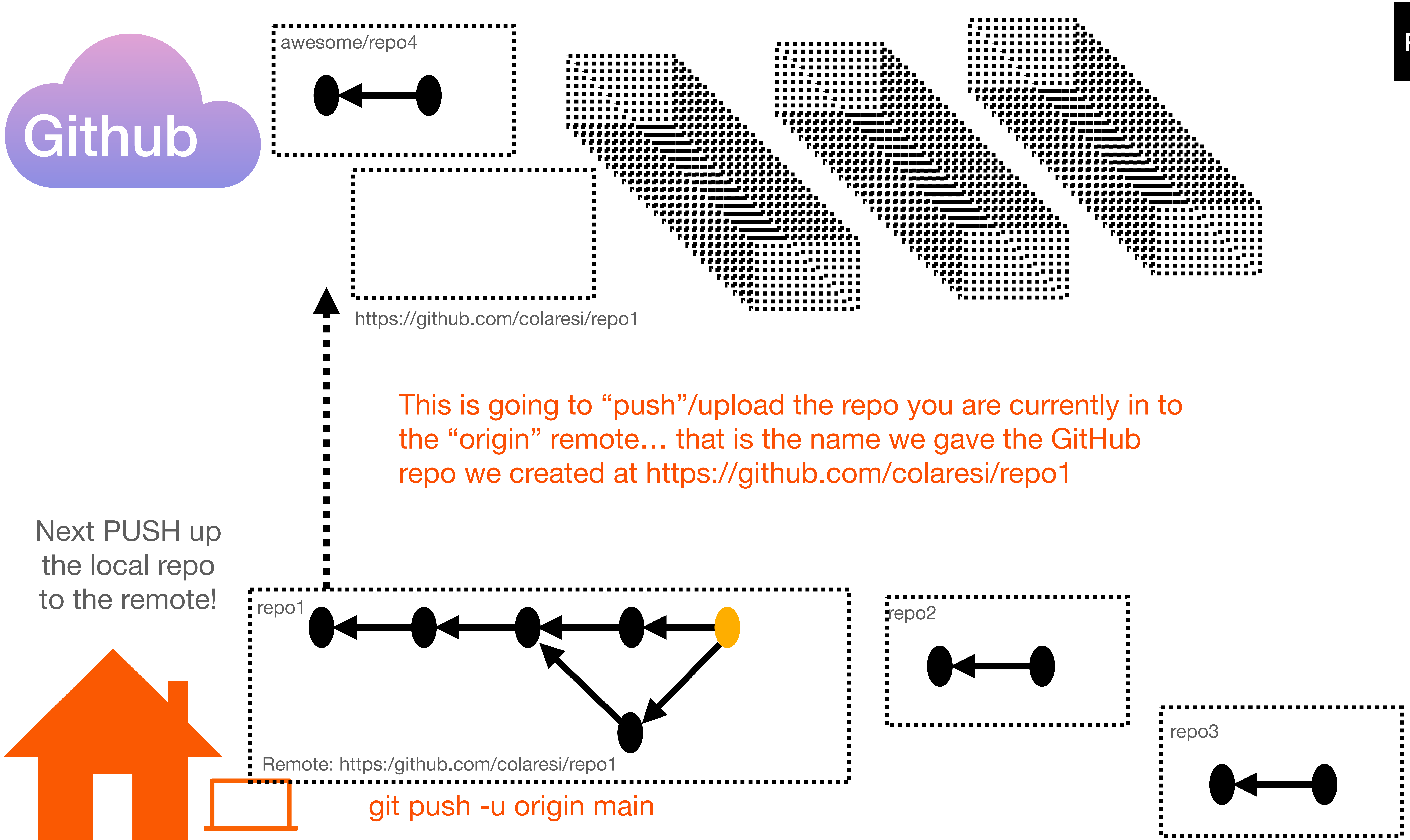
Git team now suggests Main as the "main" branch names

-M implies -m and -f
-m == move branch name
-f == force

Avoids problems with master/main naming conflicts right now

git branch -M main

Push

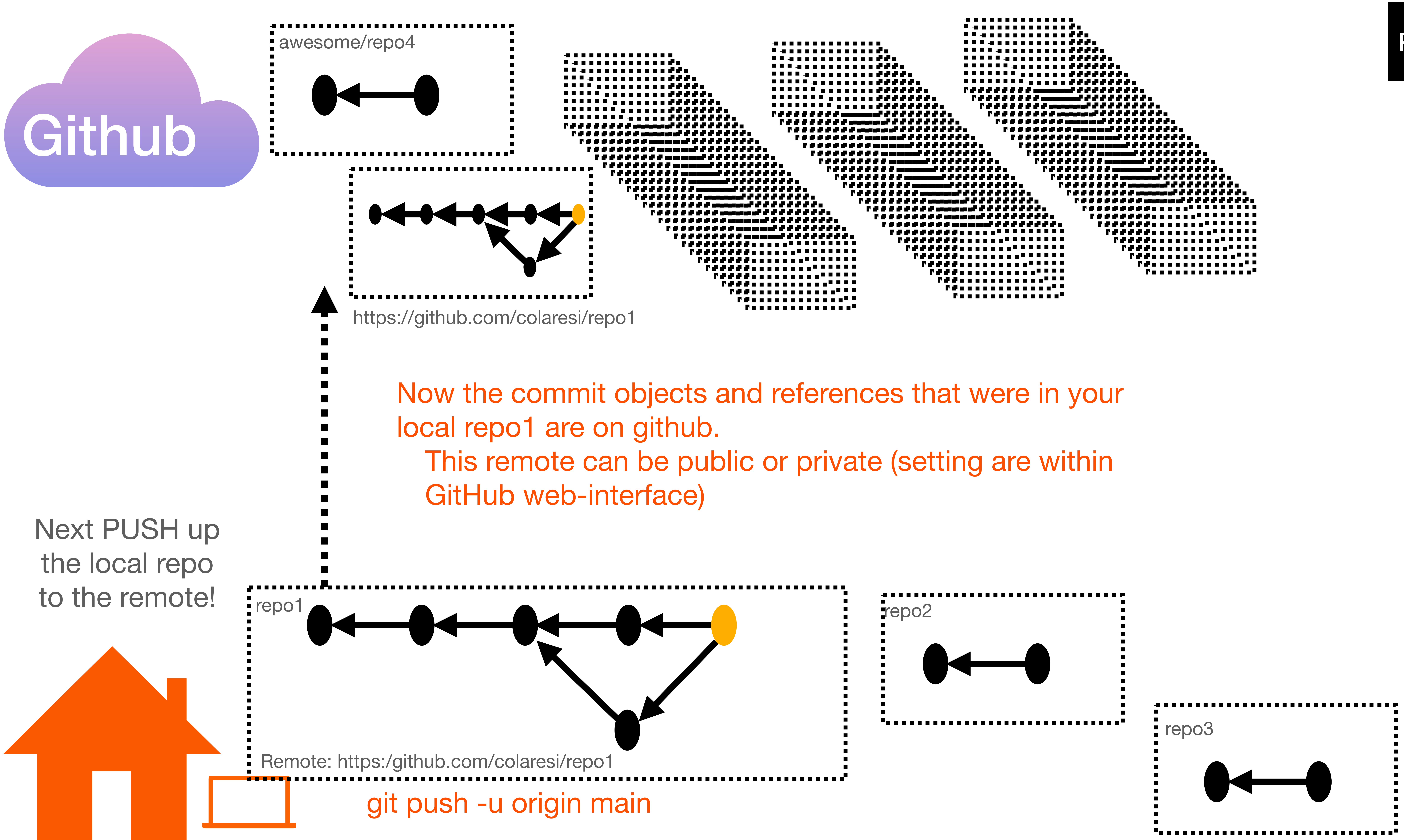

Github
awesome/repo4
https://github.com/colaresi/repo1
repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo3

This is going to "push"/upload the repo you are currently in to the "origin" remote… that is the name we gave the GitHub repo we created at https://github.com/colaresi/repo1

Next PUSH up the local repo to the remote!

```
git push -u origin main
```

Push


Github
awesome/repo4
https://github.com/colaresi/repo1
repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo3

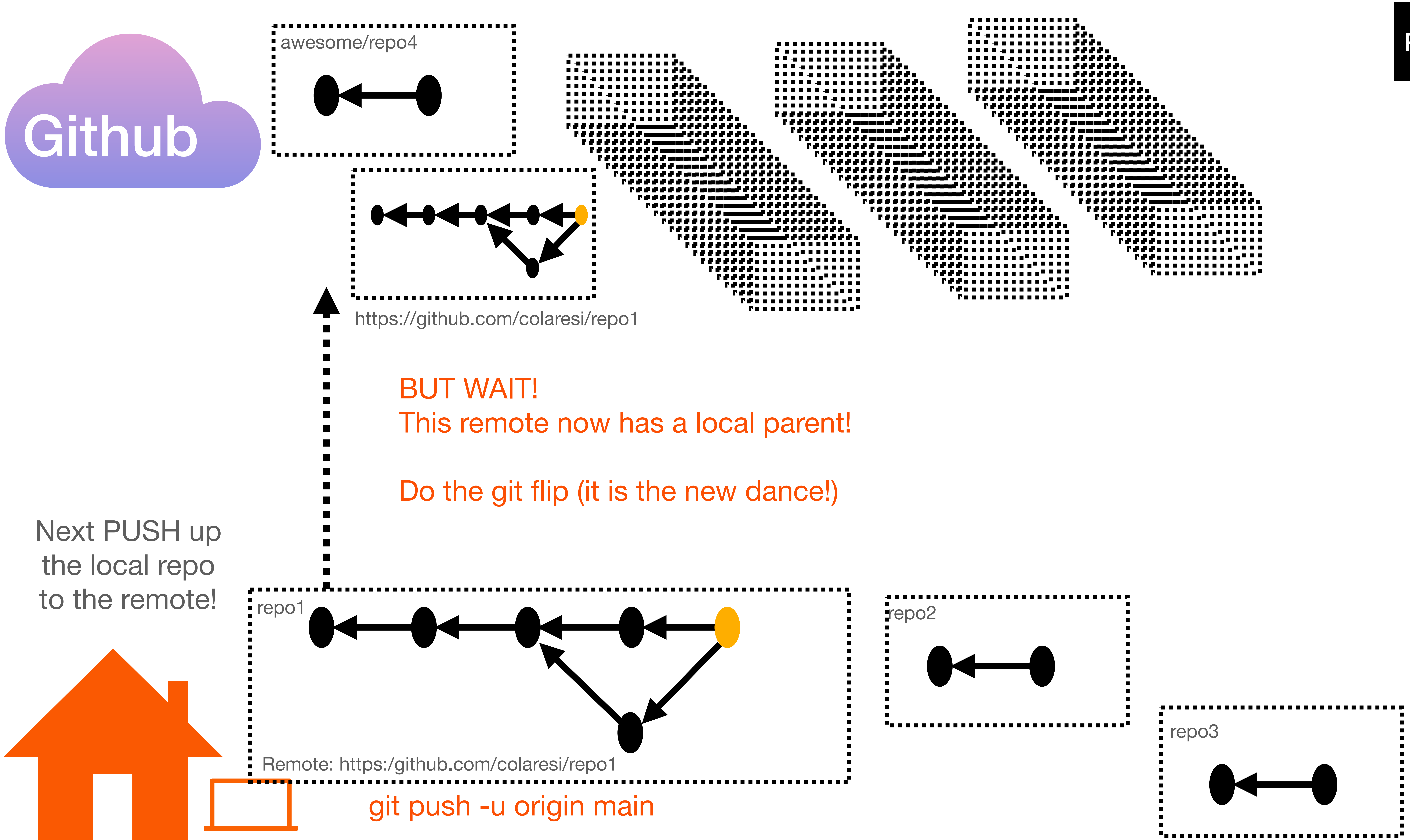
Now the commit objects and references that were in your local repo1 are on github.

This remote can be public or private (setting are within GitHub web-interface)

Next PUSH up the local repo to the remote!

git push -u origin main

Push


Github
awesome/repo4
https://github.com/colaresi/repo1
repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo3

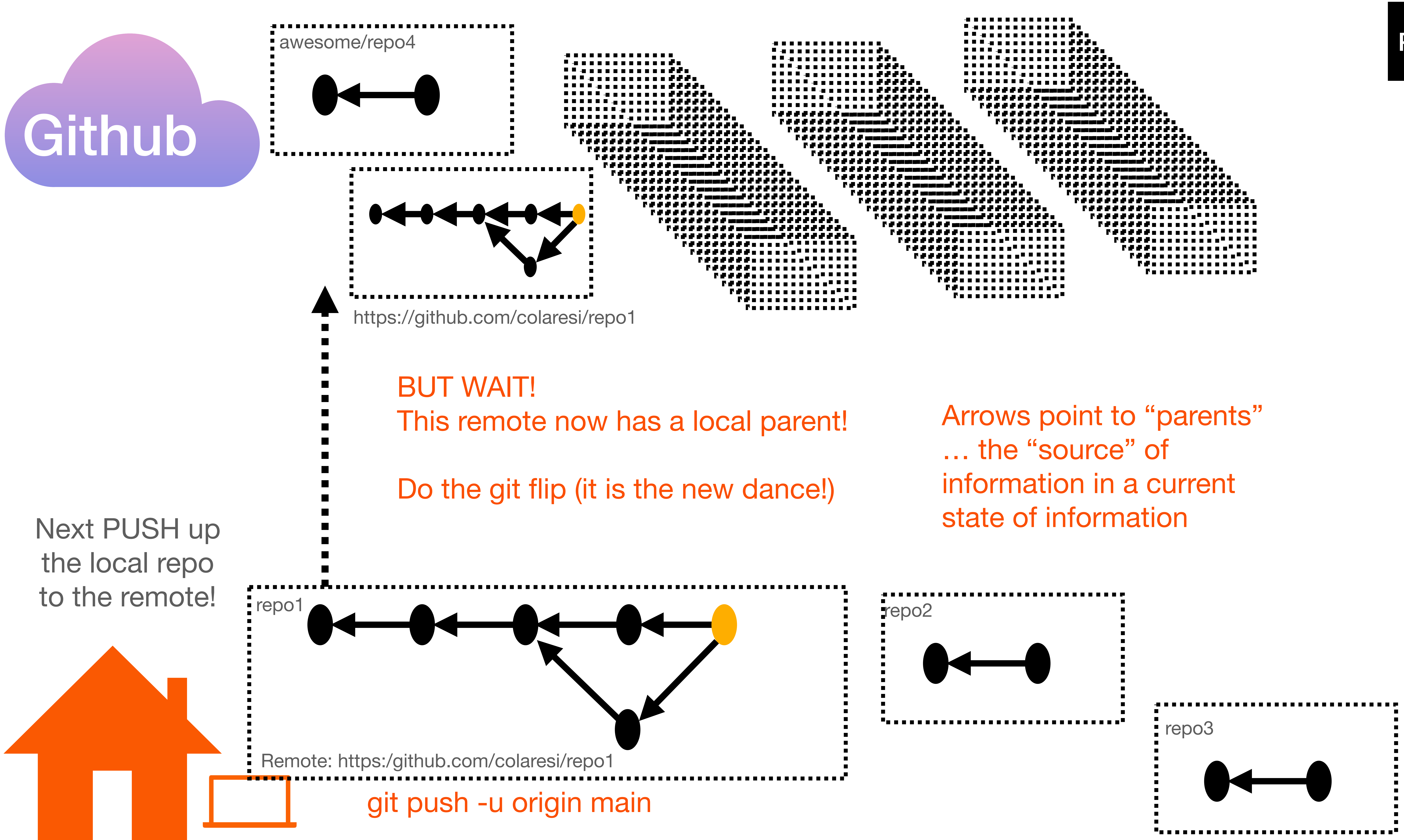
BUT WAIT!
This remote now has a local parent!

Do the git flip (it is the new dance!)

Next PUSH up
the local repo
to the remote!

git push -u origin main

Push


Github
awesome/repo4
https://github.com/colaresi/repo1
repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo3

Next PUSH up the local repo to the remote!

BUT WAIT!
This remote now has a local parent!

Do the git flip (it is the new dance!)

Arrows point to “parents” … the “source” of information in a current state of information

```
git push -u origin main
```

Push

Github
awesome/repo4
https://github.com/colaresi/repo1
repo1
Remote: https:/github.com/colaresi/repo1
repo2
3

Next PUSH up the local repo to the remote!

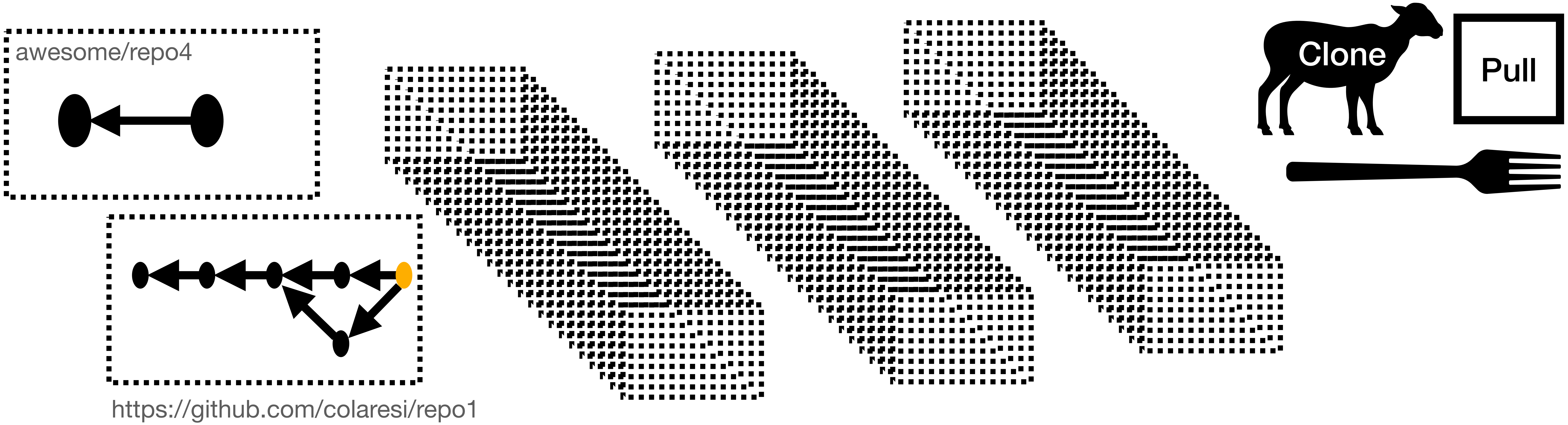
Arrows point to “parents” … the “source” of information in a cur state of informati

```
git push -u origin main
```

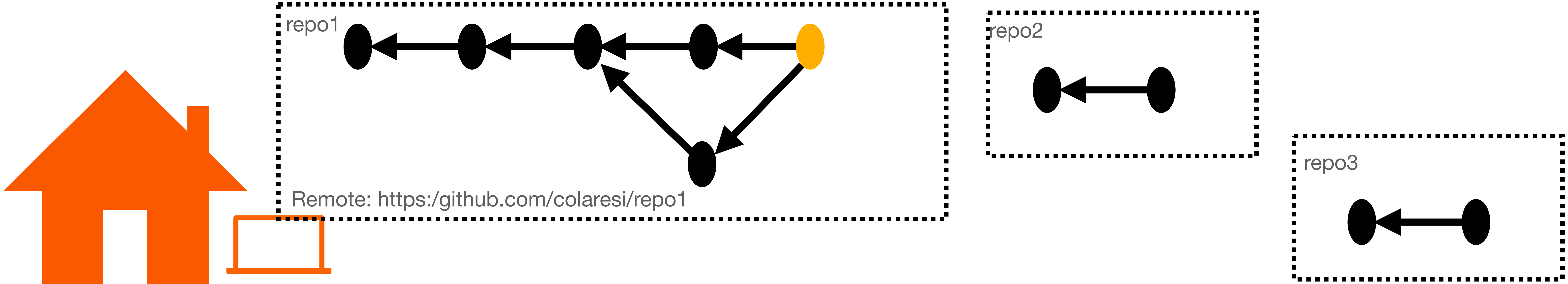
-u means —set-upstream Allows you to “pull" down content later…

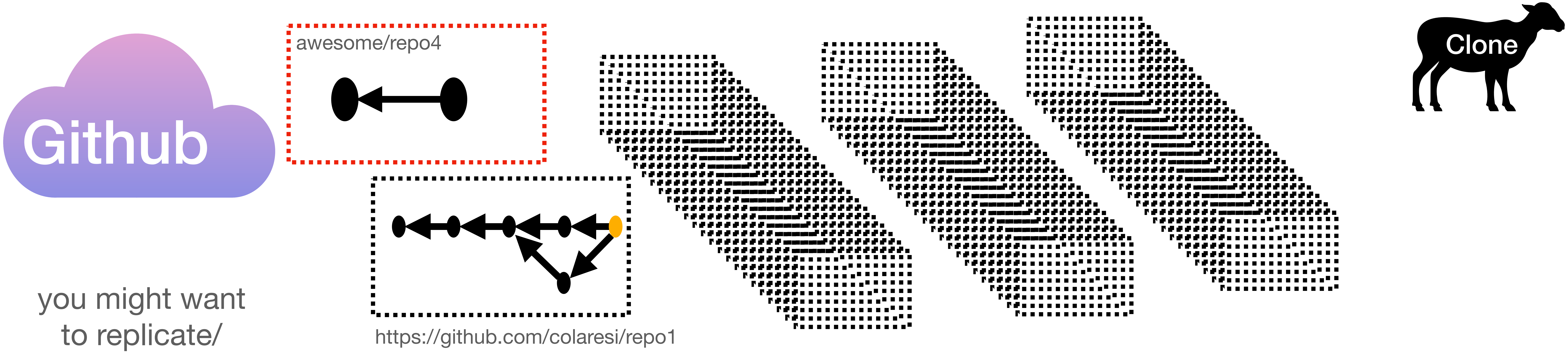
Github

If the repo is public, now others can “clone”, or “fork” your repo and you or they can “pull” in new commits

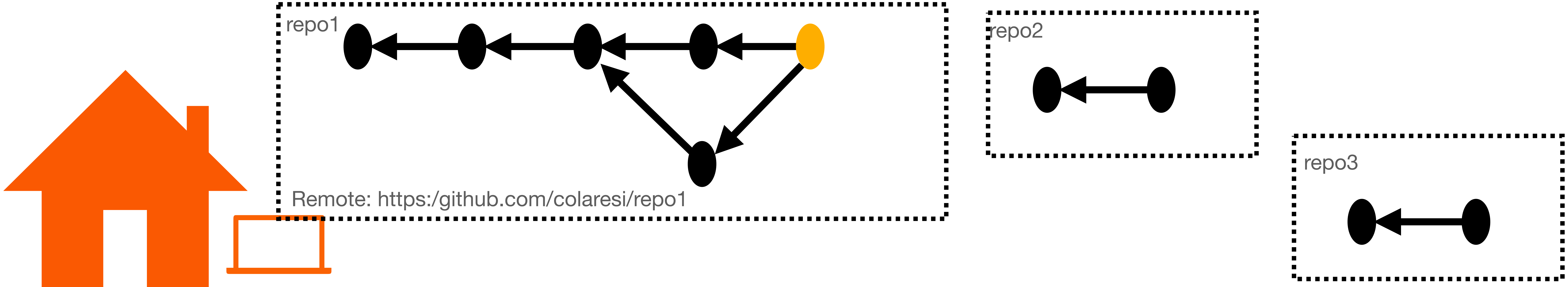


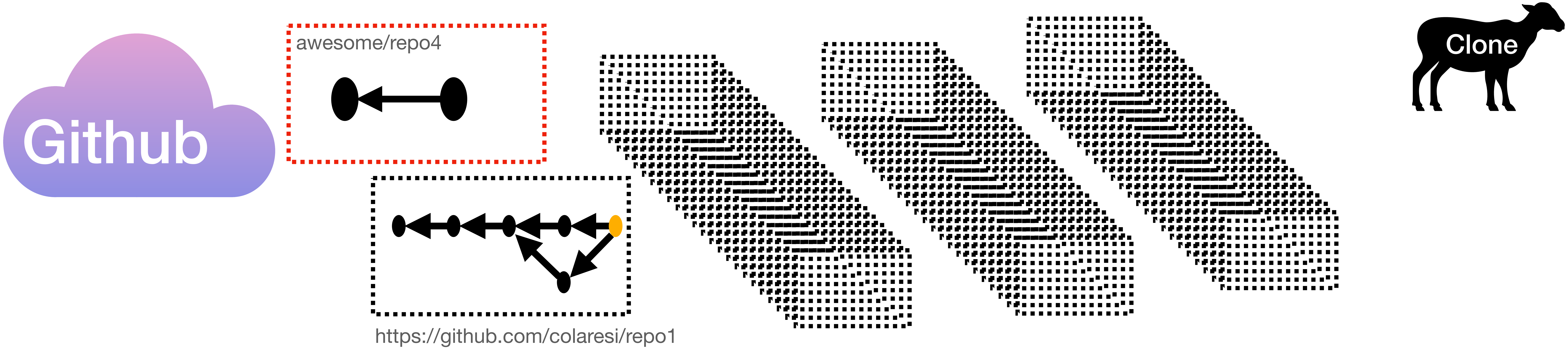
DOWNLOADS



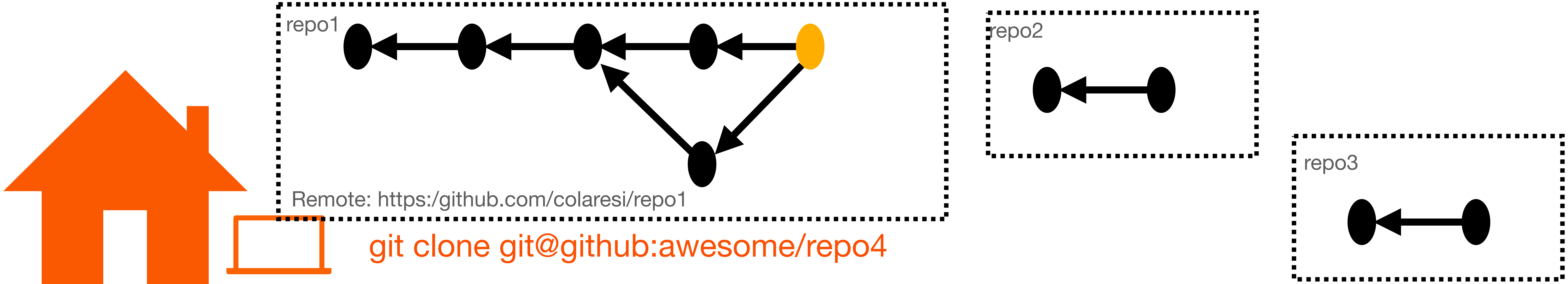


you might want to replicate/ clone work that is in awesome/ repo4

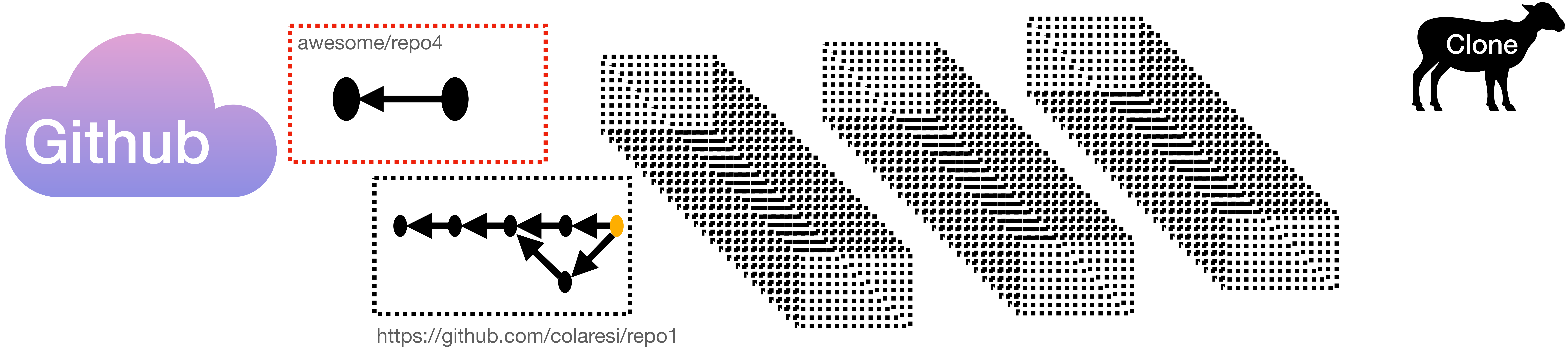




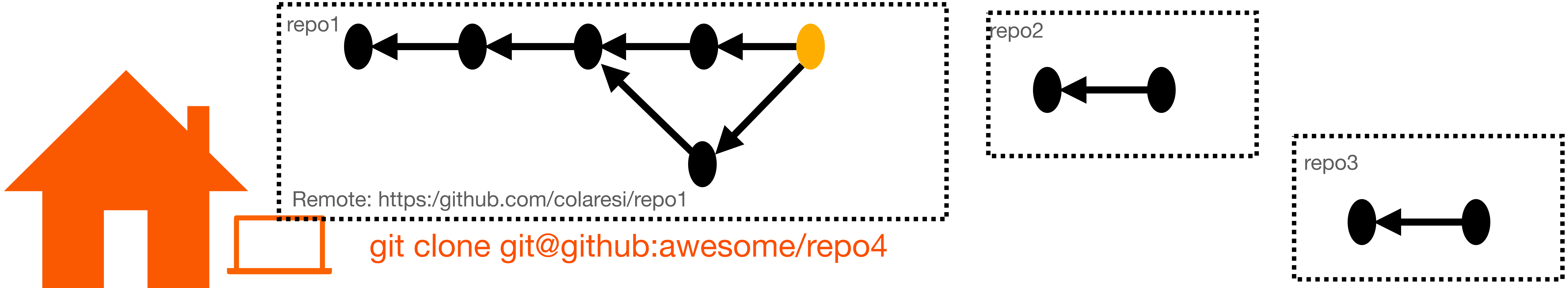
git clone REPO_ADDRESS



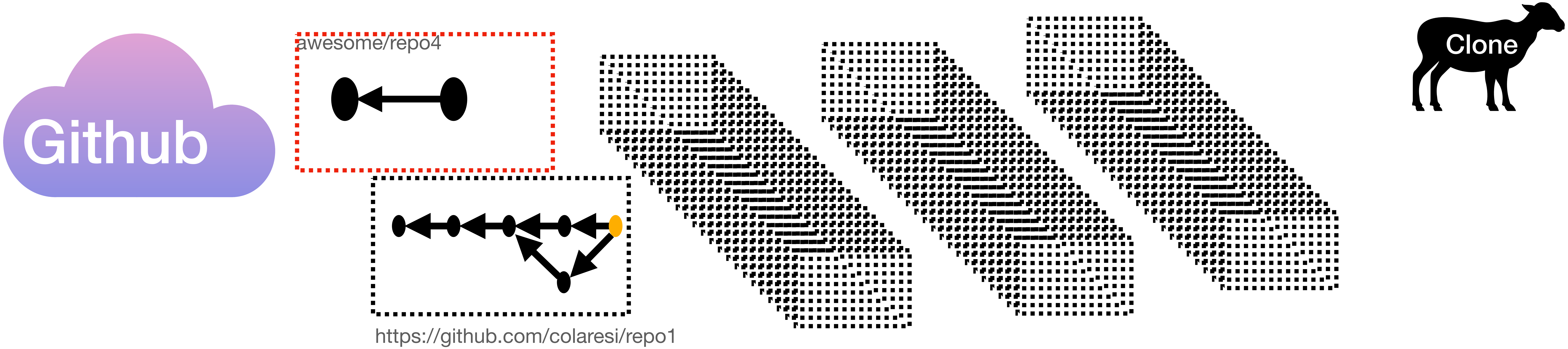
git clone git@github:awesome/repo4



git clone is going to do a number of things … in sequence for you

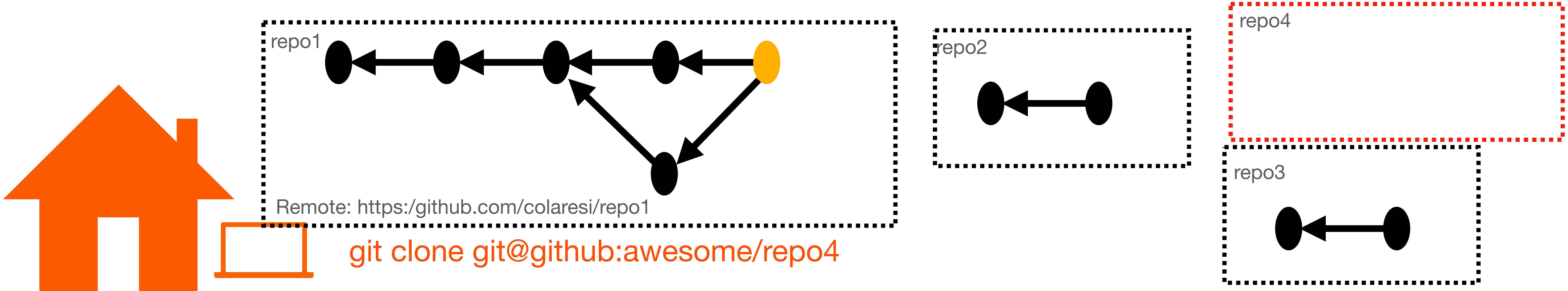


git clone git@github:awesome/repo4

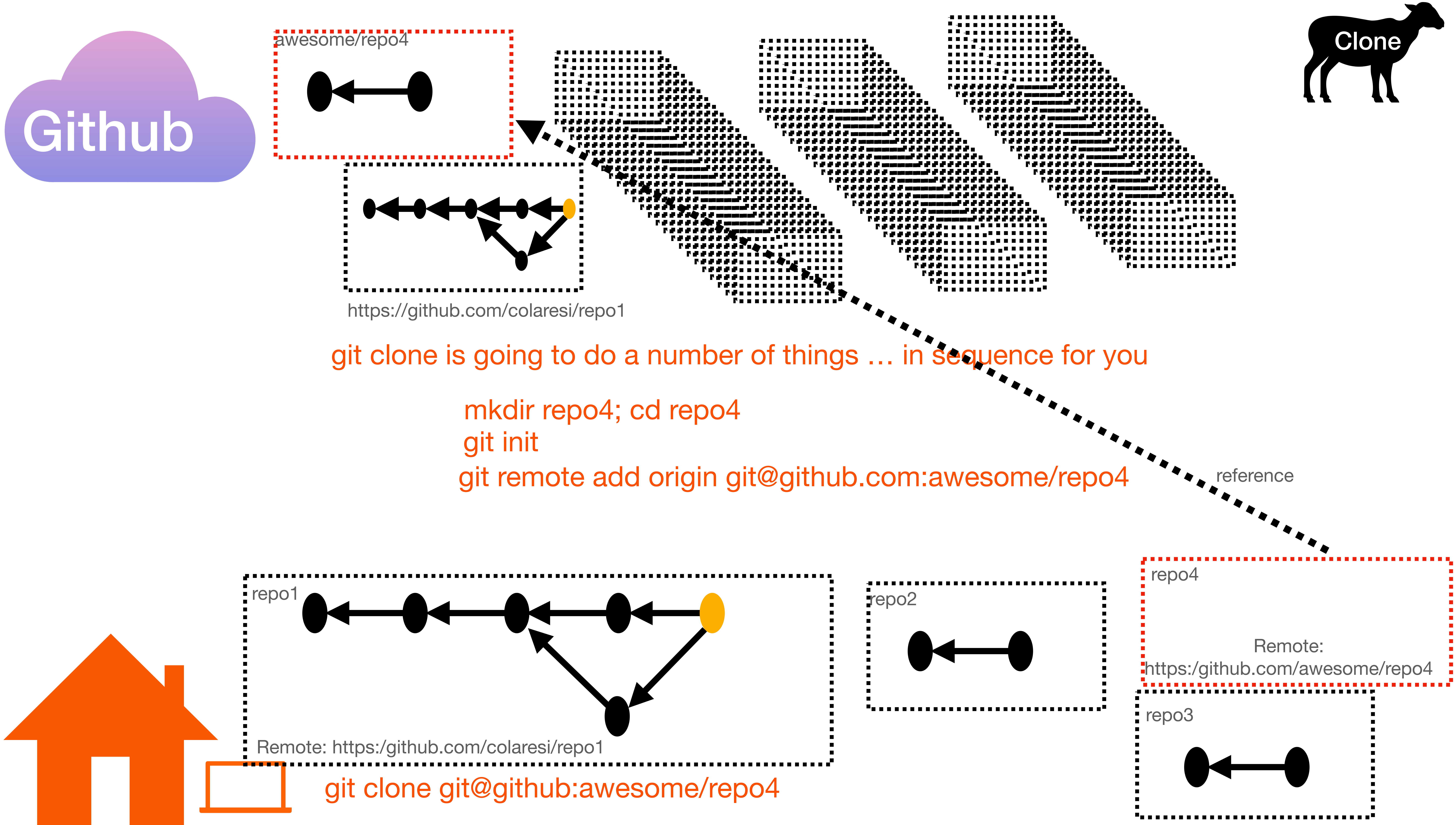


git clone is going to do a number of things … in sequence for you

```
mkdir repo4; cd repo4
git init
```



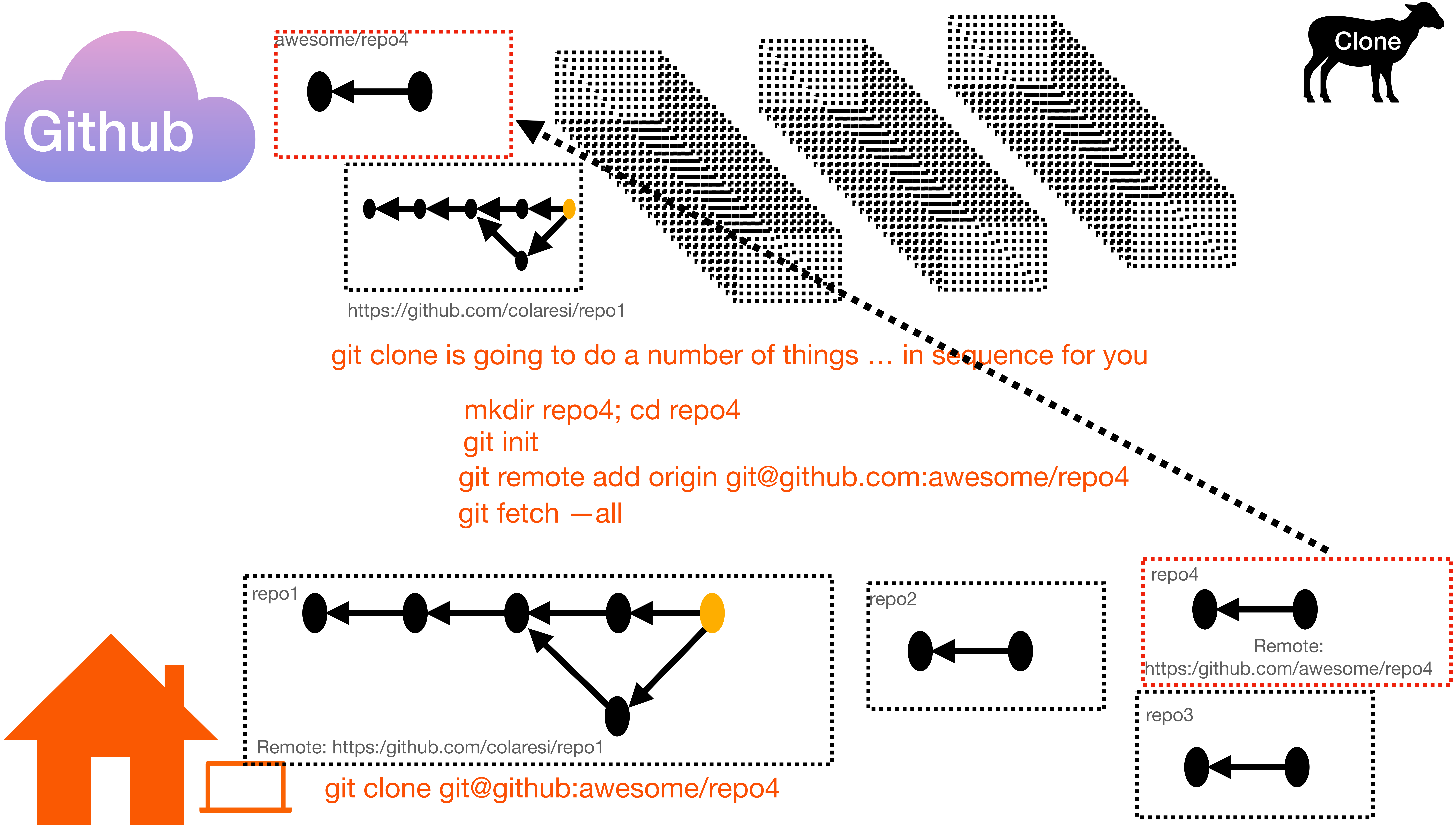
```
git clone git@github:awesome/repo4
```


Github
awesome/repo4
https://github.com/colaresi/repo1
Clone
reference
repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo4
Remote:
https:/github.com/awesome/repo4
repo3

git clone is going to do a number of things … in sequence for you

```
mkdir repo4; cd repo4
git init
git remote add origin git@github.com:awesome/repo4
```

```
git clone git@github:awesome/repo4
```


Github
awesome/repo4
https://github.com/colaresi/repo1
Clone
repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo4
Remote:
https:/github.com/awesome/repo4
repo3

git clone is going to do a number of things … in sequence for you

```
mkdir repo4; cd repo4
git init
git remote add origin git@github.com:awesome/repo4
git fetch —all
```

```
git clone git@github:awesome/repo4
```

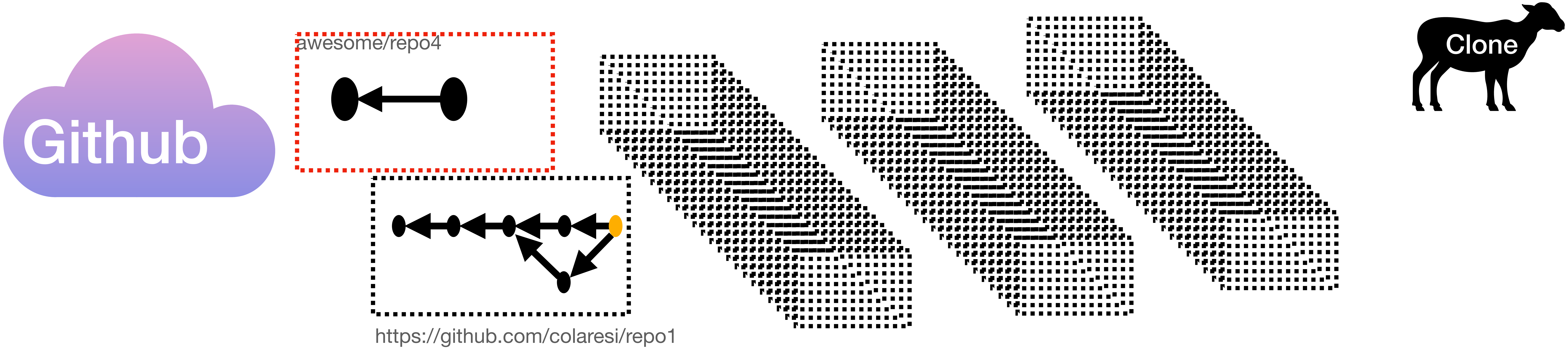
Github
awesome/repo4
https://github.com/colaresi/repo1
Clone
HEAD
repo4
Remote:
https:/github.com/awesome/repo4
repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo3

git clone is going to do a number of things … in sequence for you

```
mkdir repo4; cd repo4
git init
git remote add origin git@github.com:awesome/repo4
git fetch —all
git checkout —track origin/main
```

Switch to main branch that is tracking/paired to the remote main branch

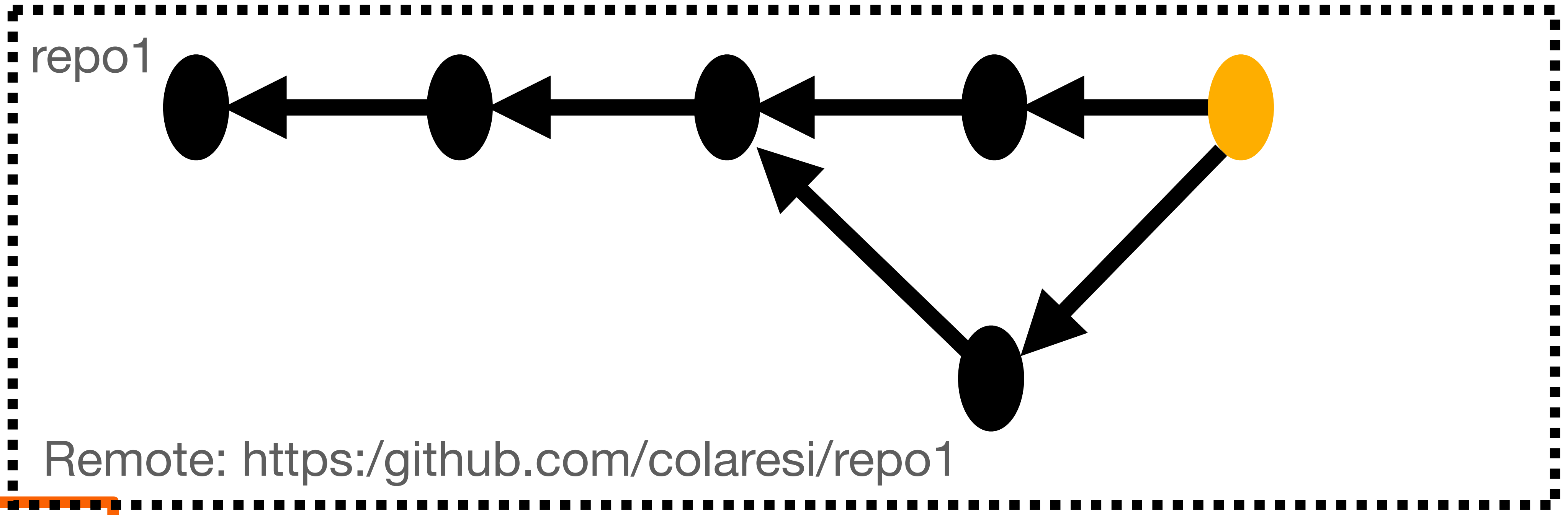
```
git clone git@github:awesome/repo4
```

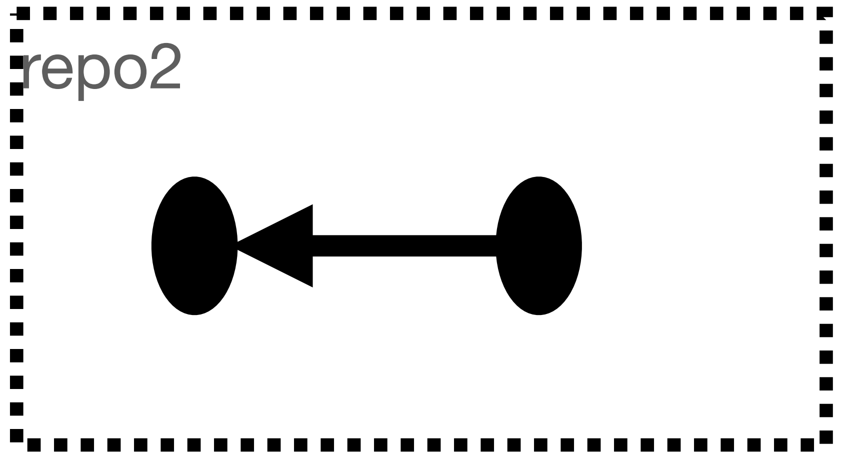


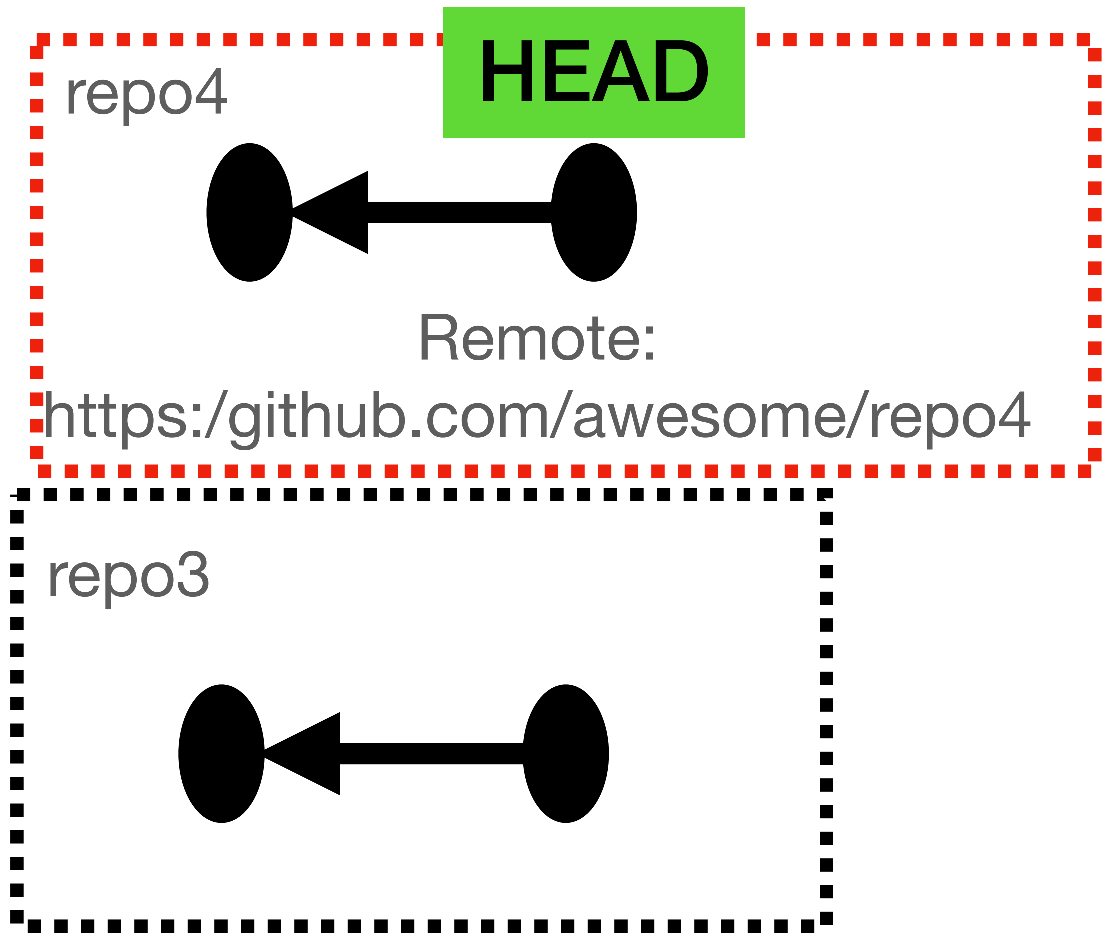
git clone is going to do a number of things … in sequence for you

```
mkdir repo4; cd repo4
git init
git remote add origin git@github.com:awesome/repo4
git fetch —all
git checkout —track origin/main
```

FINALLy:
cd ..

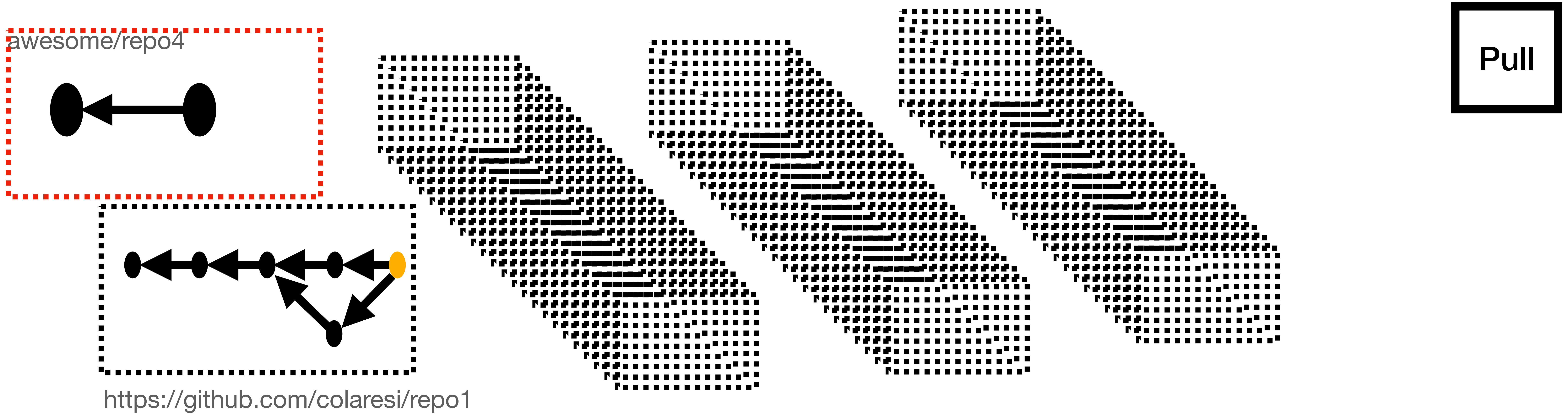




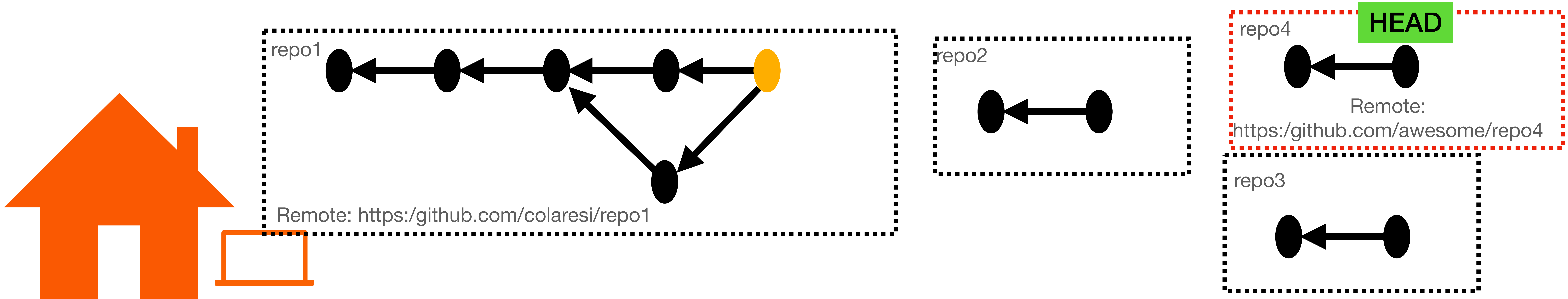


git clone git@github:awesome/repo4

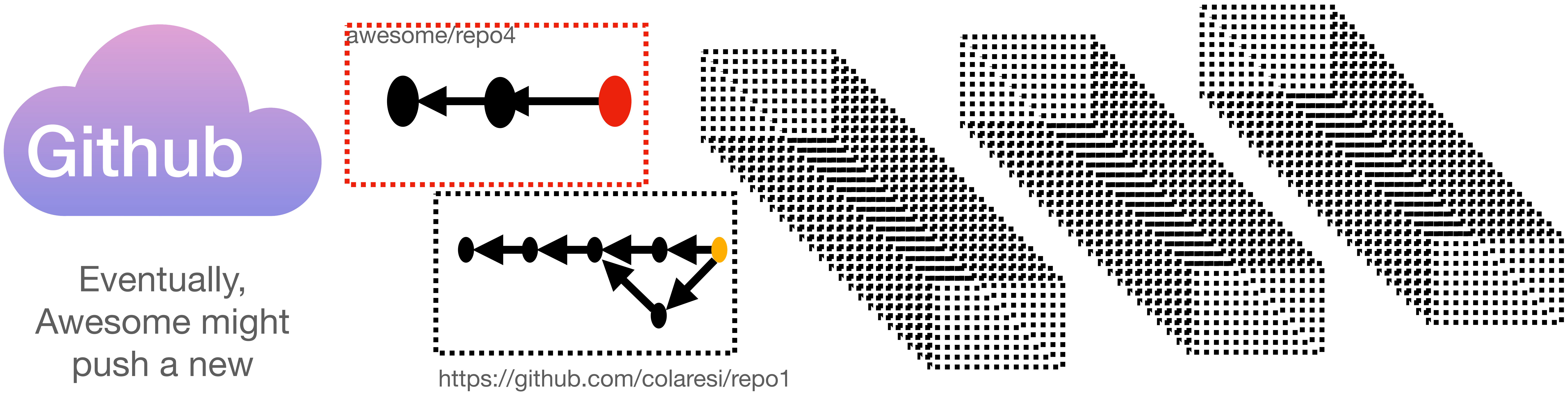
Github



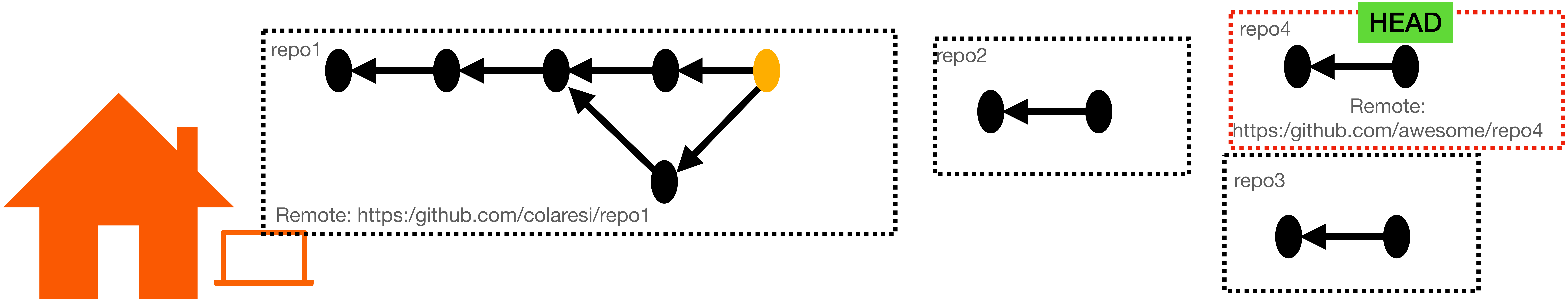
Git clone is just the start… you can “pull” new commits into your repos


repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo4
HEAD
Remote:
https:/github.com/awesome/repo4
repo3

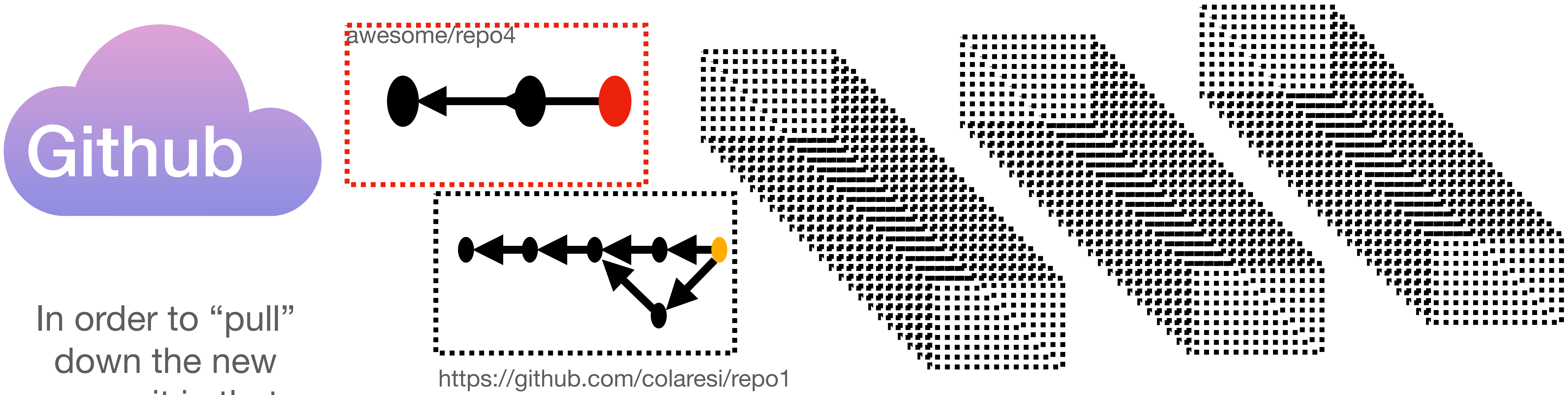
Pull



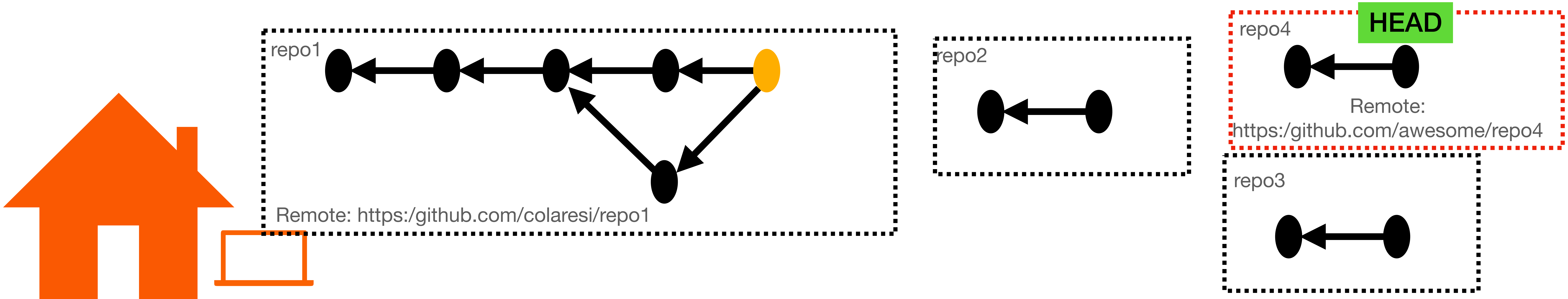
Eventually, Awesome might push a new commit to awesome/repo4


repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo4
HEAD
Remote:
https:/github.com/awesome/repo4
repo3

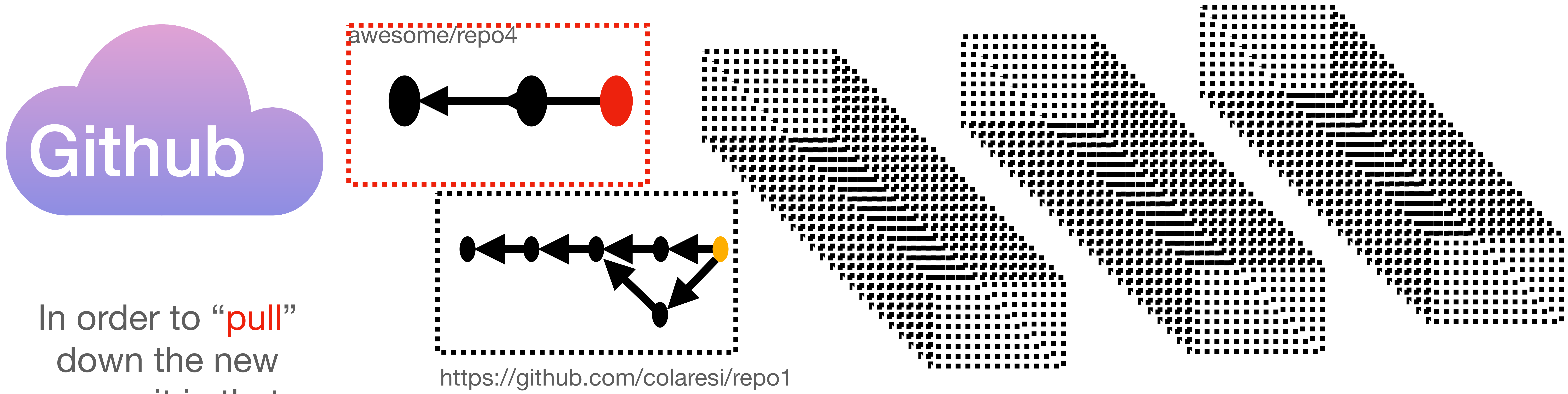
Pull



In order to “pull” down the new commit in that repo.. you actually are doing 2 things!


repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo4
HEAD
Remote:
https:/github.com/awesome/repo4
repo3

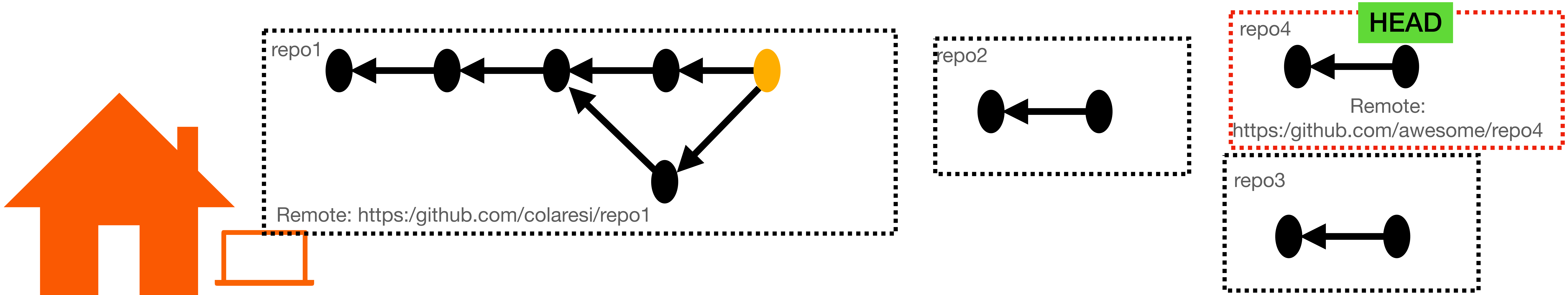
Pull



In order to "pull" down the new commit in that repo.. you actually are doing 2 things!

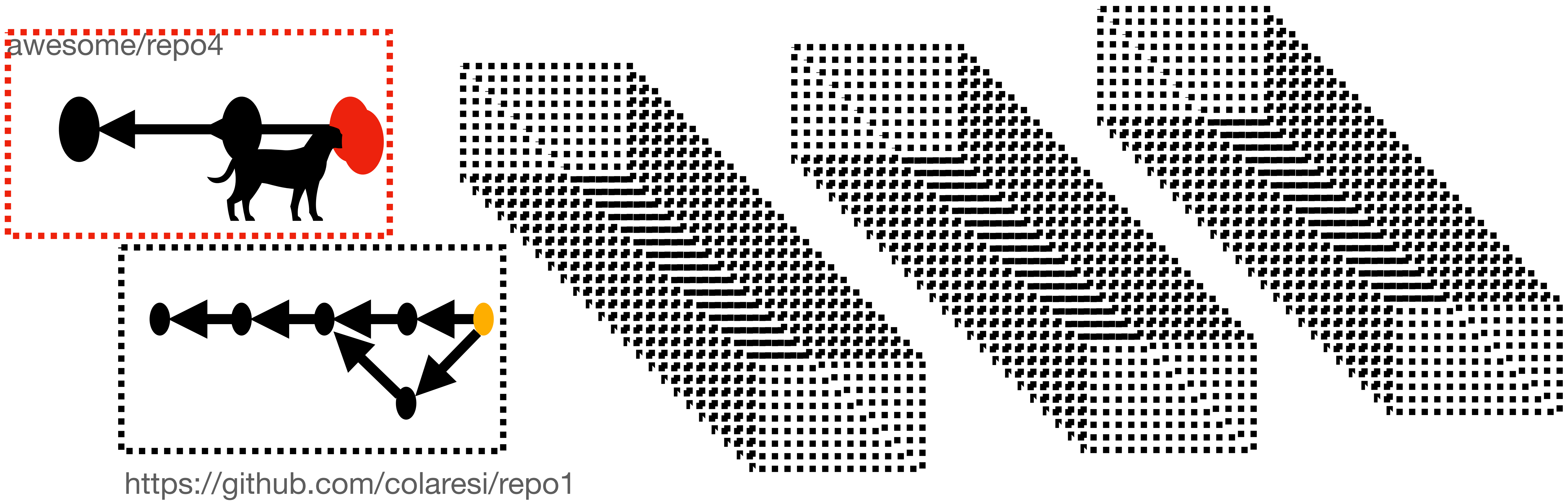
git fetch

git merge


repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo4
HEAD
Remote:
https:/github.com/awesome/repo4
repo3

Pull

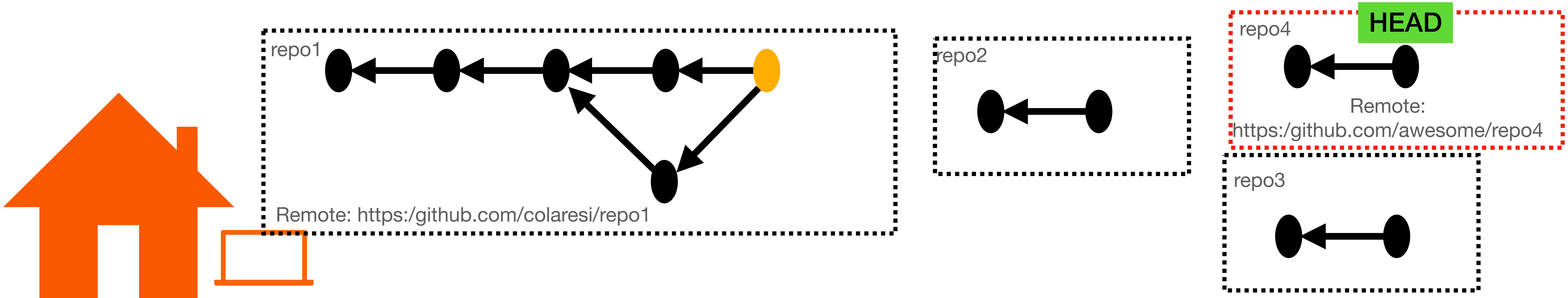
Github



You want to "fetch" that commit from the server,

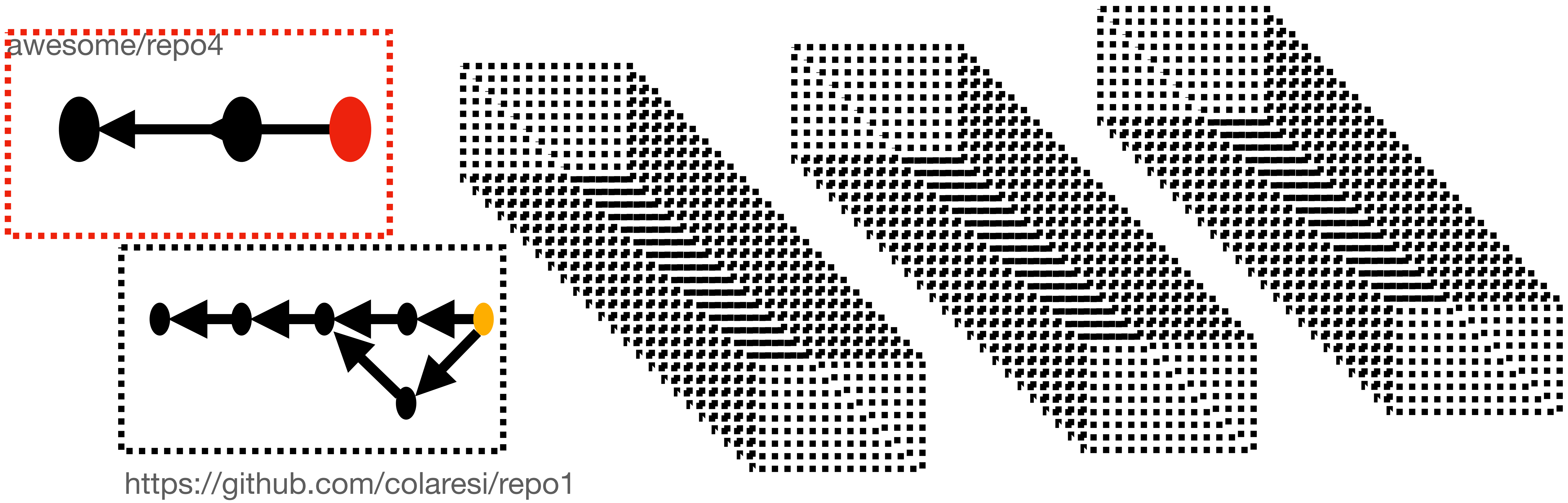
git fetch

git merge


repo1
Remote: https:/github.com/colaresi/repo1
repo2
HEAD
repo4
Remote:
https:/github.com/awesome/repo4
repo3

Pull

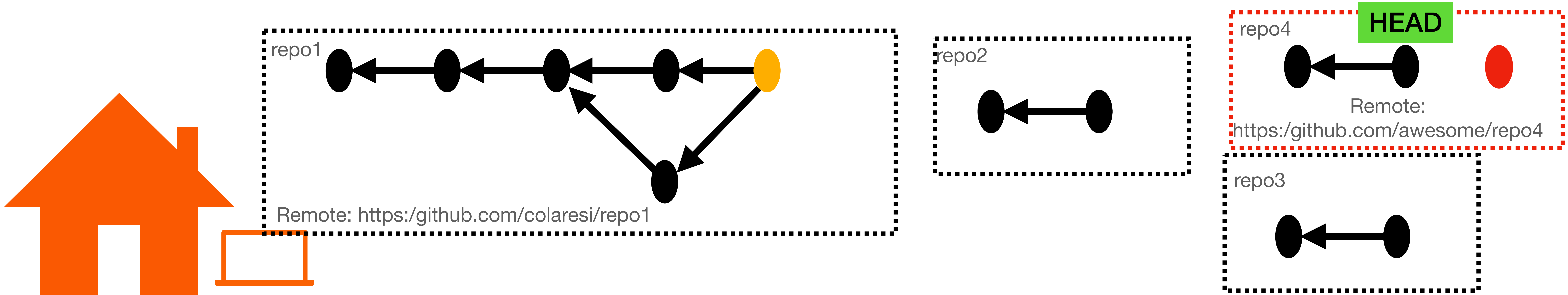
Github



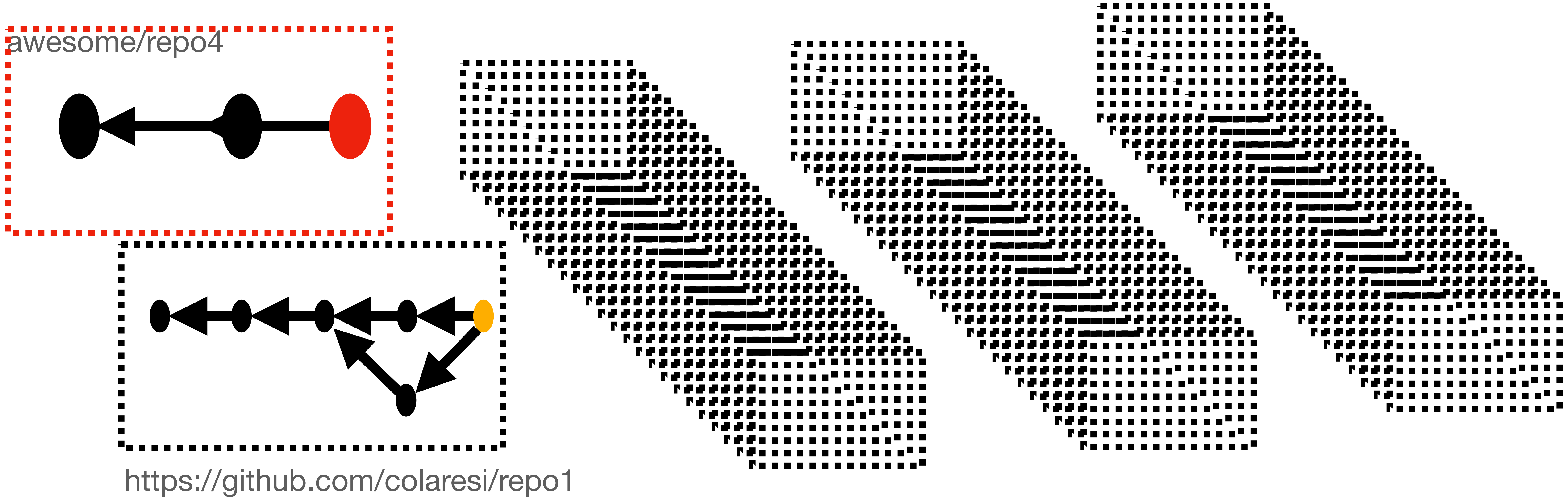
You want to “fetch” that commit from the server, and “merge” it into your local repo

git fetch

git merge


repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo4
HEAD
Remote:
https:/github.com/awesome/repo4
repo3

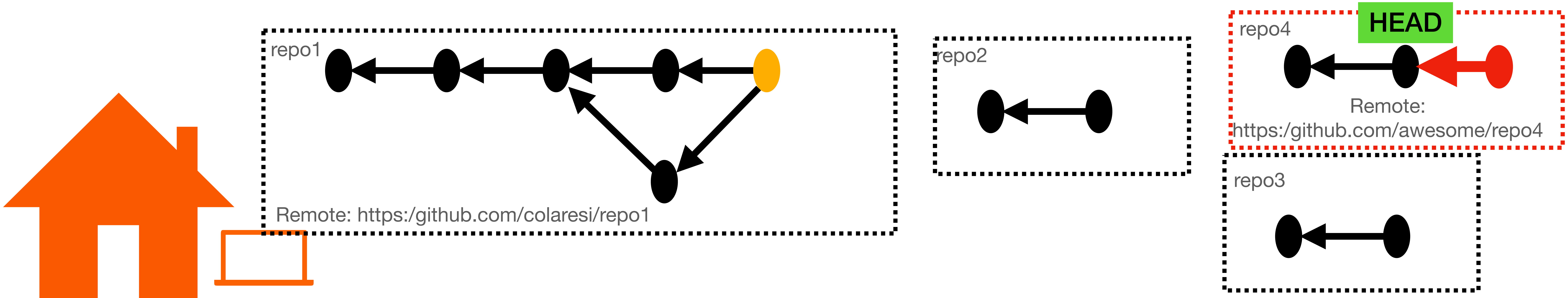
Pull



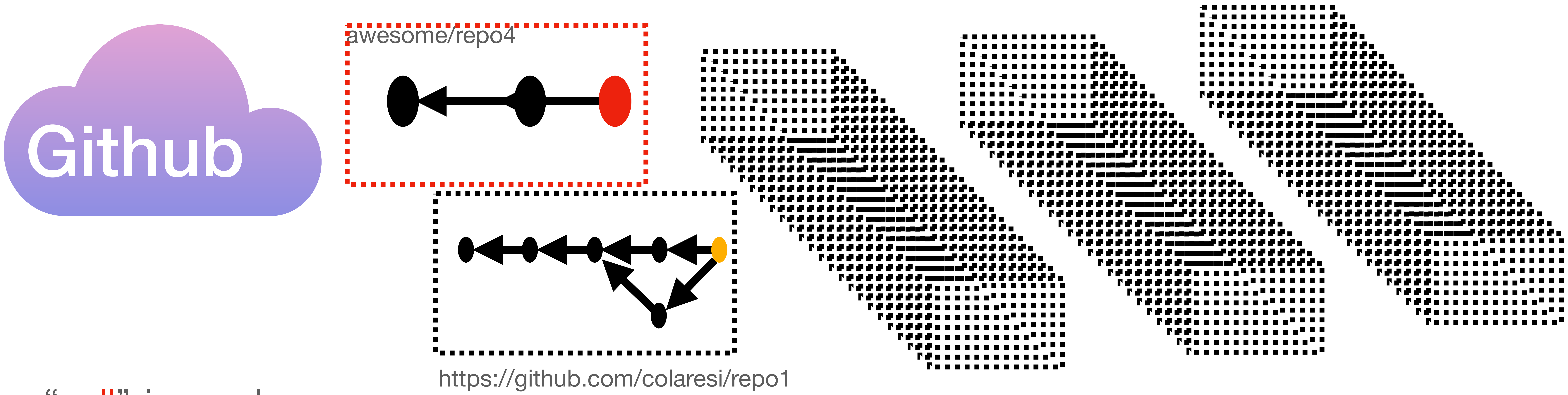
You want to “fetch” that commit from the server, and “merge” it into your local repo

git fetch

git merge


repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo4
HEAD
Remote:
https:/github.com/awesome/repo4
repo3

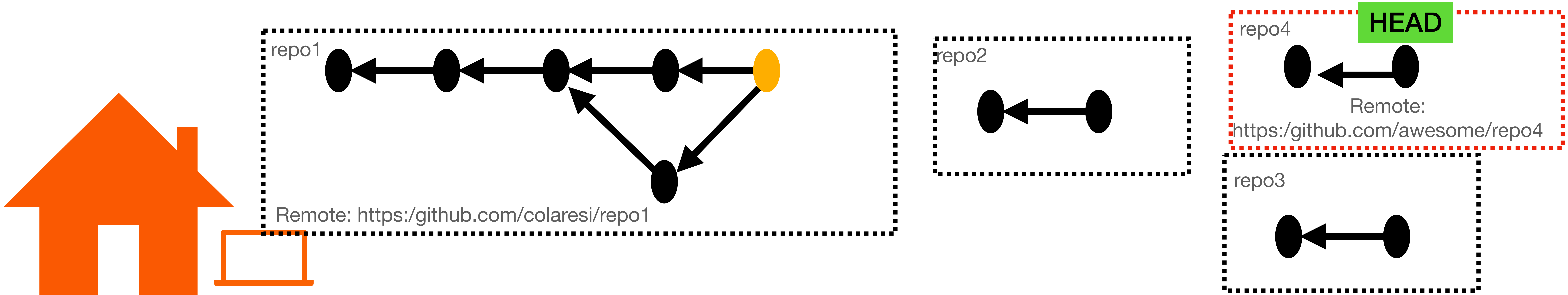
Pull



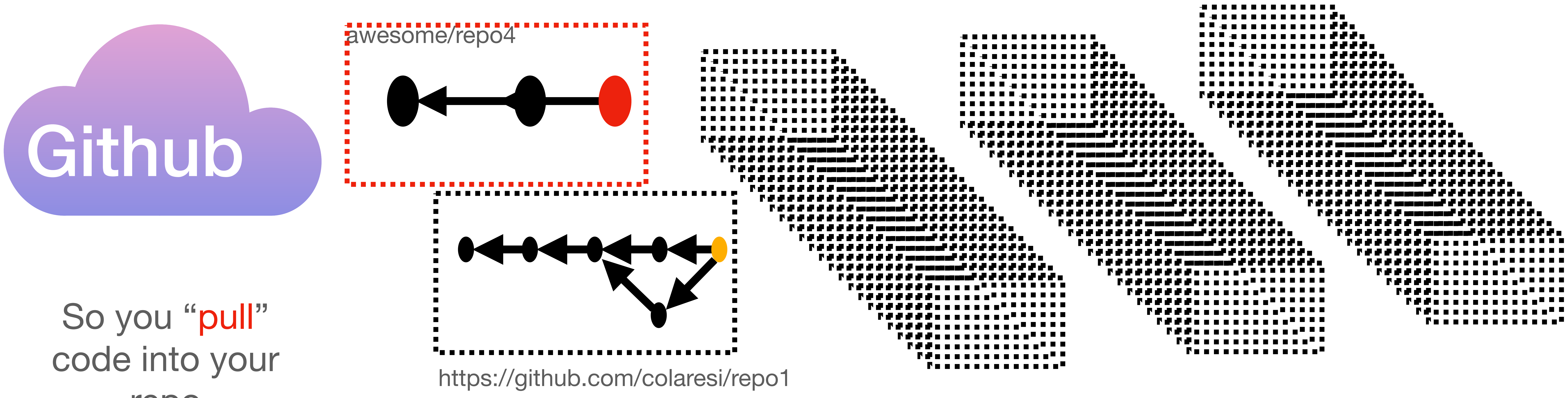
“pull”-ing code
IS
Fetching and merging

git fetch
git merge origin/master] = git pull

Does both


repo1
Remote: https:/github.com/colaresi/repo1
repo2
HEAD
repo4
Remote:
https:/github.com/awesome/repo4
repo3

Pull

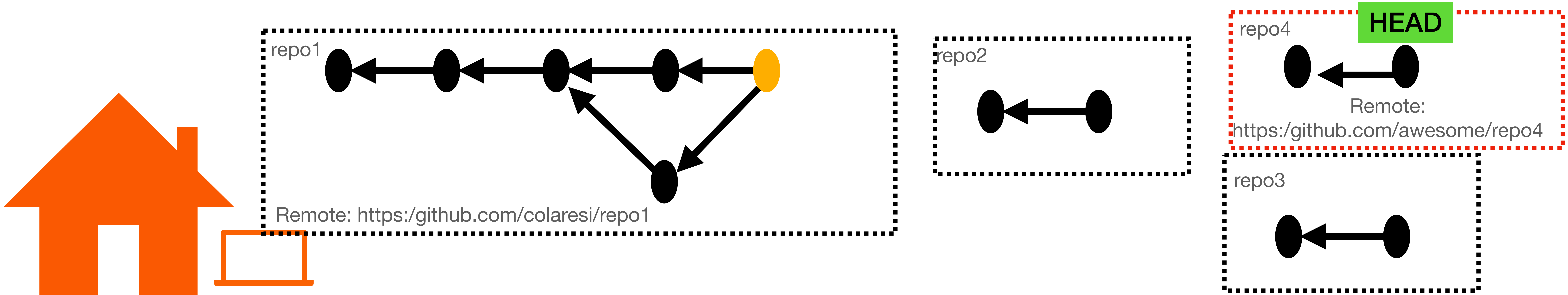


So you “pull” code into your repo

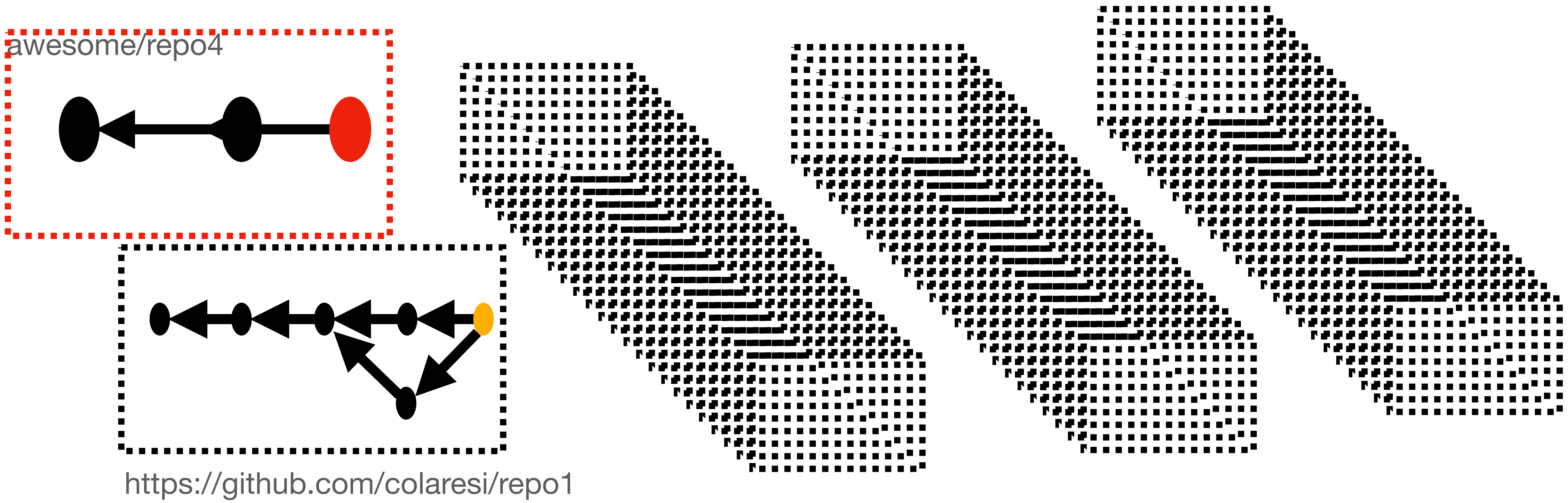
You pull done code a lot when working as part of teams

git fetch
git merge origin/master
= git pull

Does both


repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo4
HEAD
Remote:
https:/github.com/awesome/repo4
repo3

Pull

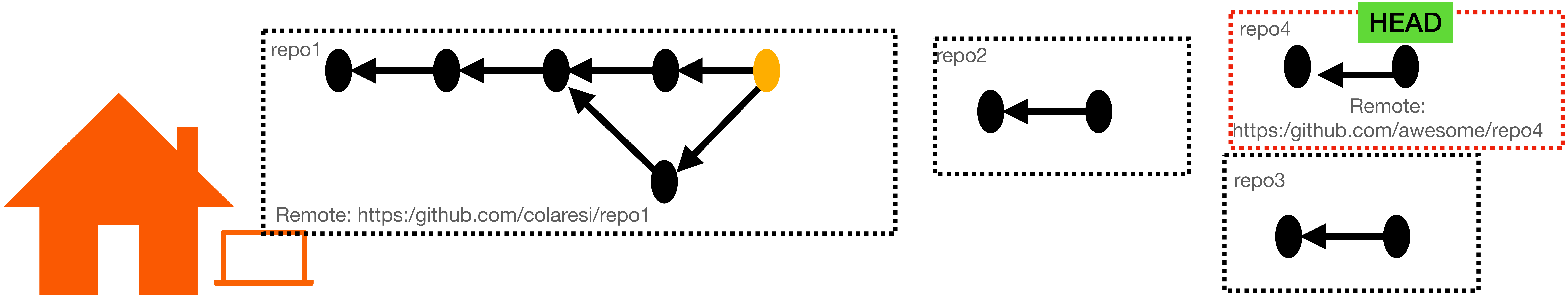


So you "pull" code into your repo

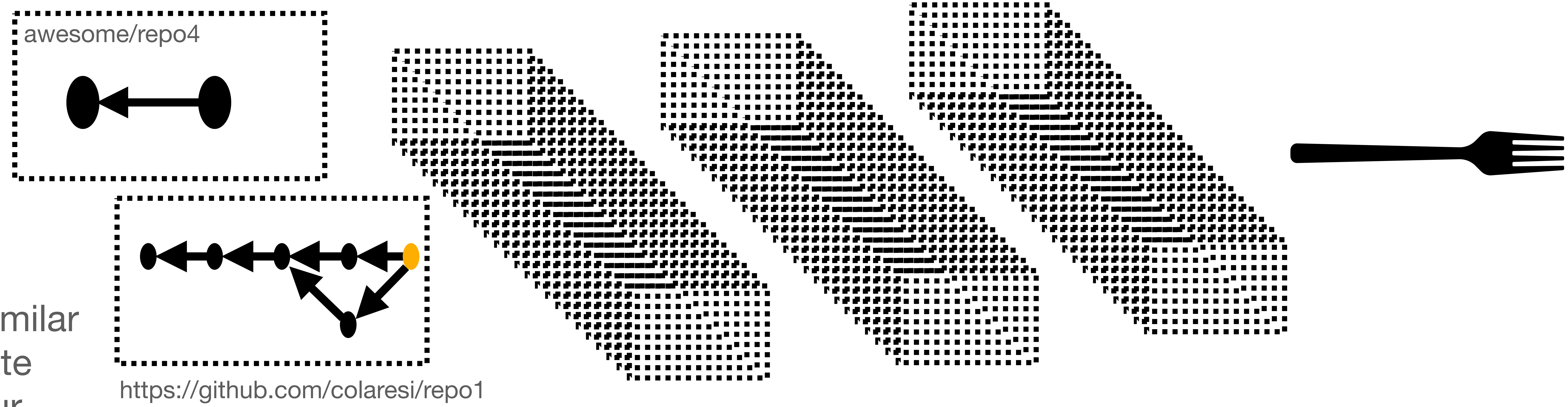
You pull done code a lot when working as part of teams

git fetch
git merge origin/master] = git pull

Does both ←

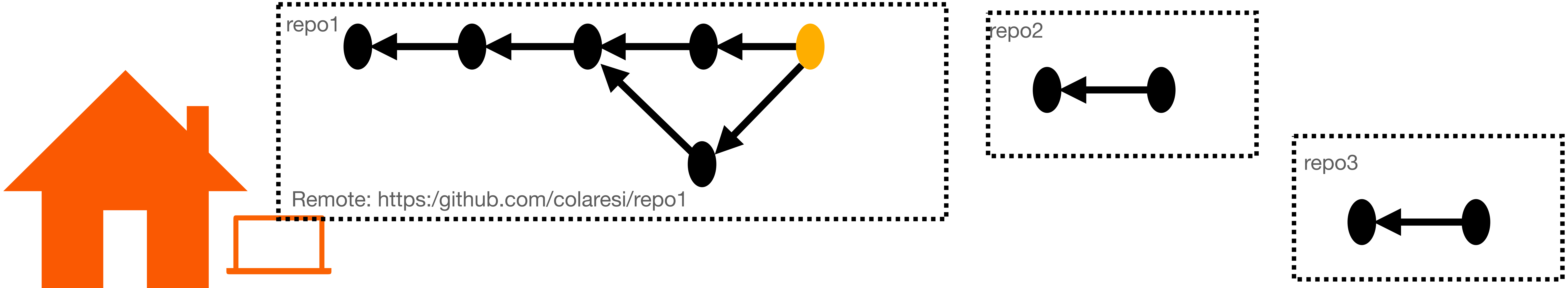

repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo4
HEAD
Remote:
https:/github.com/awesome/repo4
repo3

Github

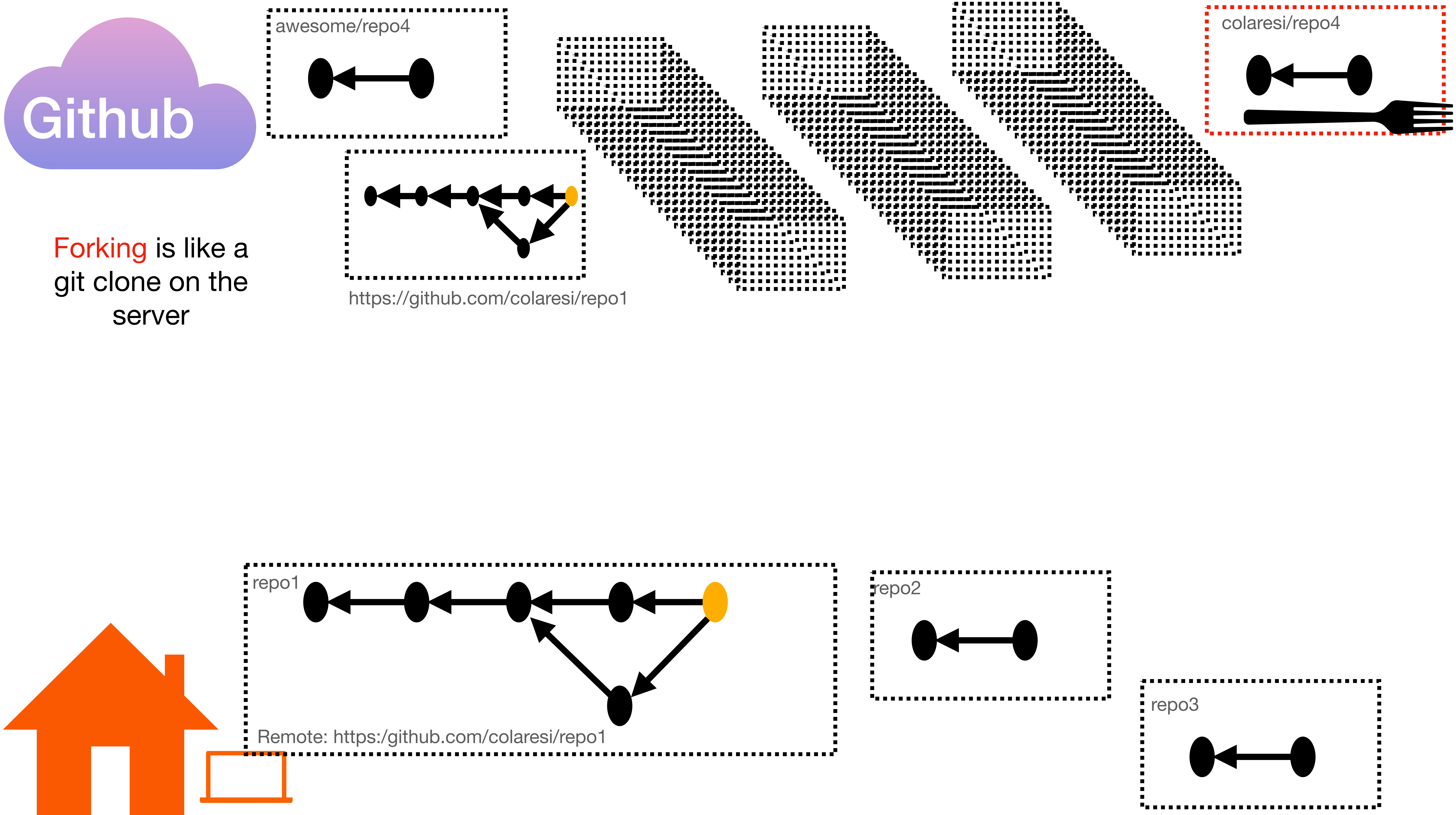


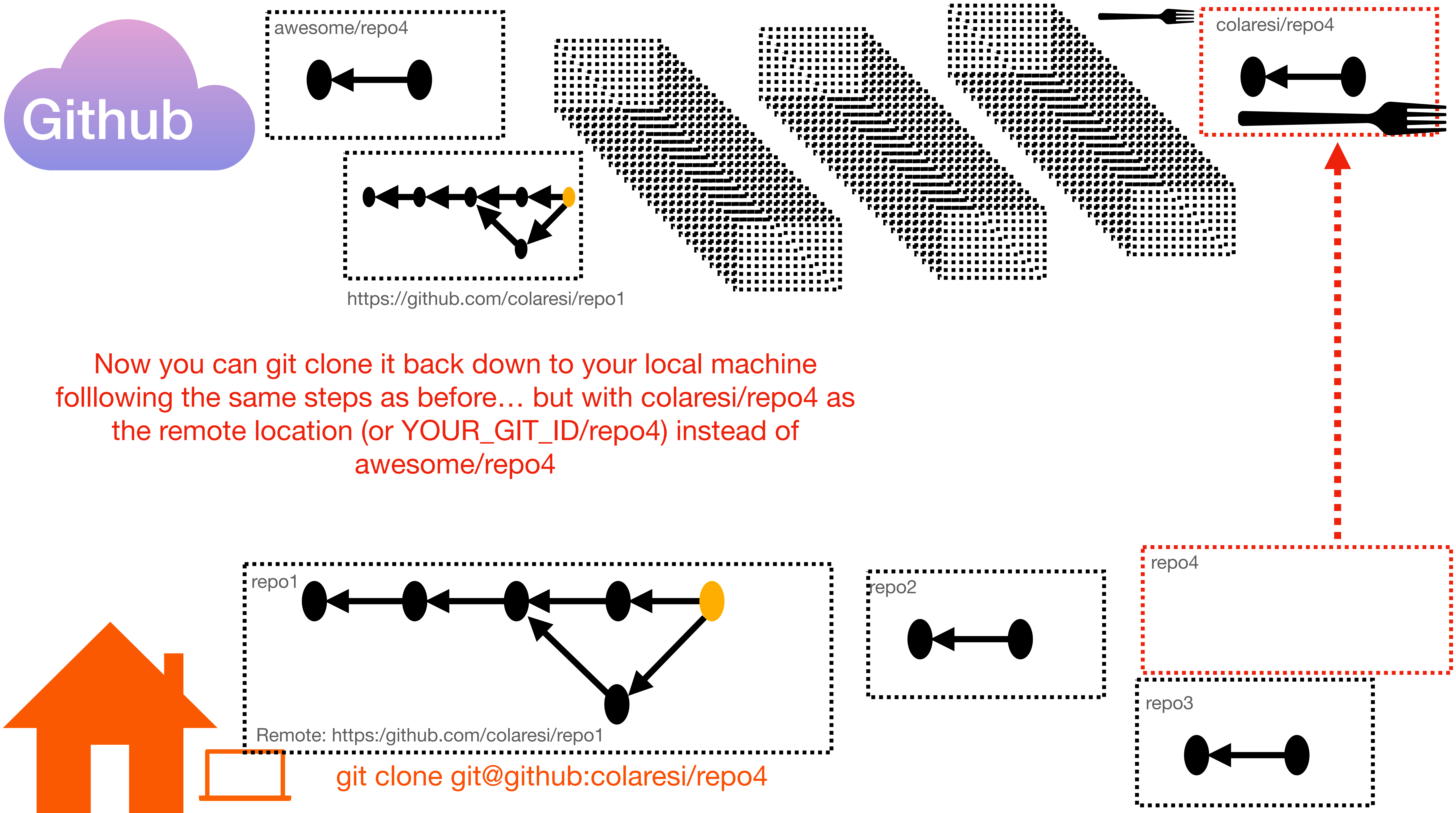
Forking a repo is similar
Except you create
The repo in your
Account on the server first

You do this on github



Forking is like a git clone on the server


Github
awesome/repo4
https://github.com/colaresi/repo1
colaresi/repo4
repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo3


Github
awesome/repo4
colaresi/repo4
https://github.com/colaresi/repo1
repo1
Remote: https:/github.com/colaresi/repo1
repo2
repo4
repo3

Now you can git clone it back down to your local machine folllowing the same steps as before… but with colaresi/repo4 as the remote location (or YOUR_GIT_ID/repo4) instead of awesome/repo4

```
git clone git@github:colaresi/repo4
```

Github

Github has other useful tools for sharing code and working as a team:

Issues (we all have issues)

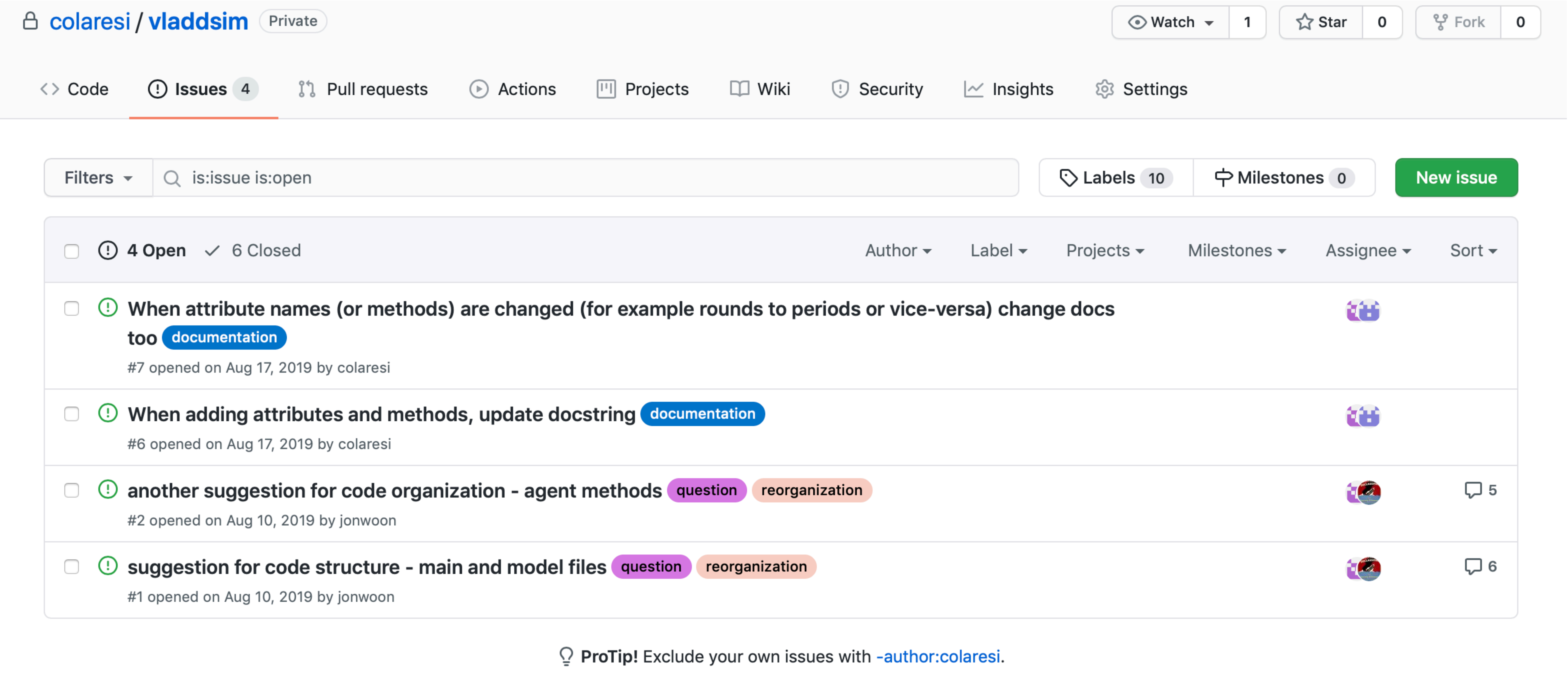
Pull requests

… like Actions, …

Github

Issues (we all have issues)

Github is a useful platform for tracking “issues”.
This means keeping notes on what is not working, who is assigned to fix it, and ideas for future enhancements



colaresi / vladdsim Private

is:issue is:open

4 Open ✓ 6 Closed

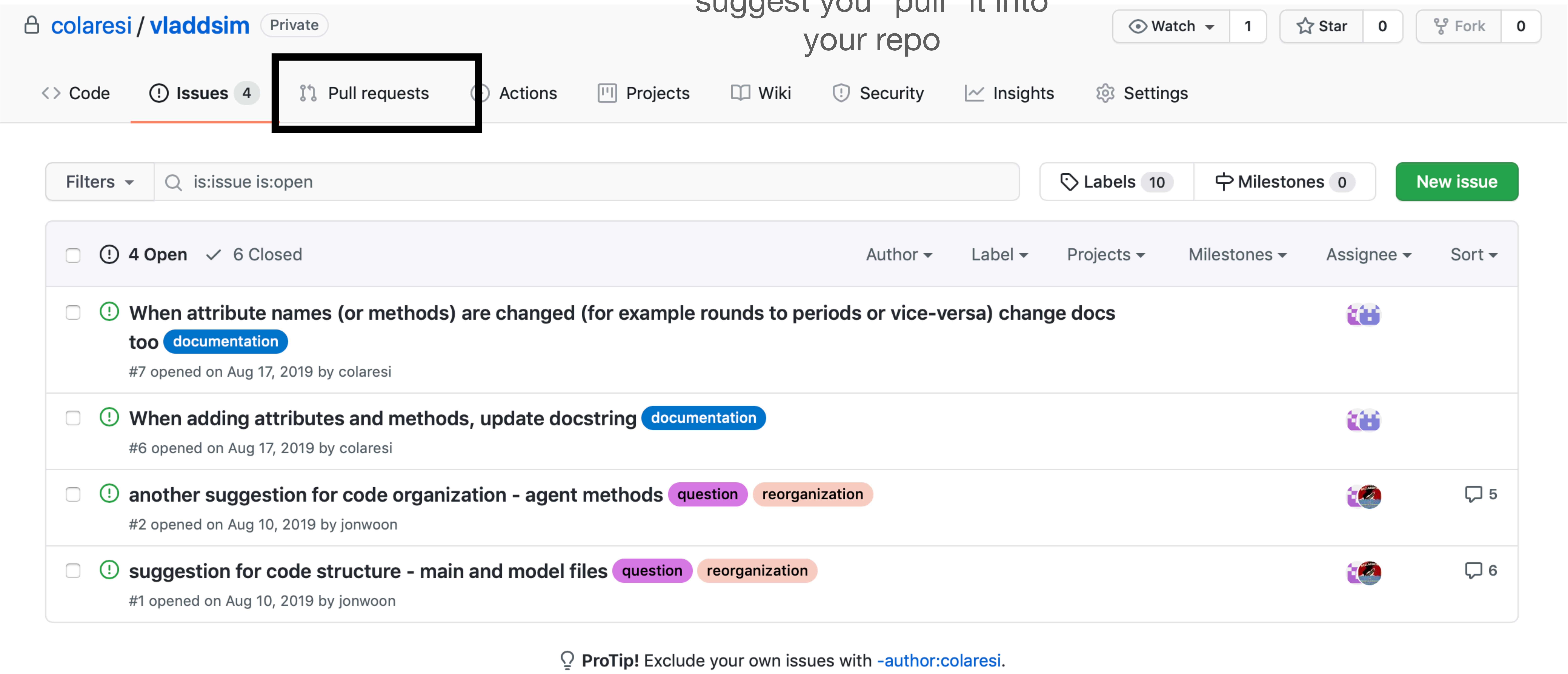
- □ When attribute names (or methods) are changed (for example rounds to periods or vice-versa) change docs too documentation
  #7 opened on Aug 17, 2019 by colaresi
- □ When adding attributes and methods, update docstring documentation
  #6 opened on Aug 17, 2019 by colaresi
- □ another suggestion for code organization - agent methods question reorganization 5
  #2 opened on Aug 10, 2019 by jonwoon
- □ suggestion for code structure - main and model files question reorganization 6
  #1 opened on Aug 10, 2019 by jonwoon

ProTip! Exclude your own issues with -author:colaresi.

Github

Pull requests

Other people write code and suggest you “pull” it into your repo



colaresi / vladdsim Private

Watch 1 | Star 0 | Fork 0

is:issue is:open

4 Open ✓ 6 Closed

- When attribute names (or methods) are changed (for example rounds to periods or vice-versa) change docs too `documentation`
  #7 opened on Aug 17, 2019 by colaresi
- When adding attributes and methods, update docstring `documentation`
  #6 opened on Aug 17, 2019 by colaresi
- another suggestion for code organization - agent methods `question` `reorganization` 5
  #2 opened on Aug 10, 2019 by jonwoon
- suggestion for code structure - main and model files `question` `reorganization` 6
  #1 opened on Aug 10, 2019 by jonwoon

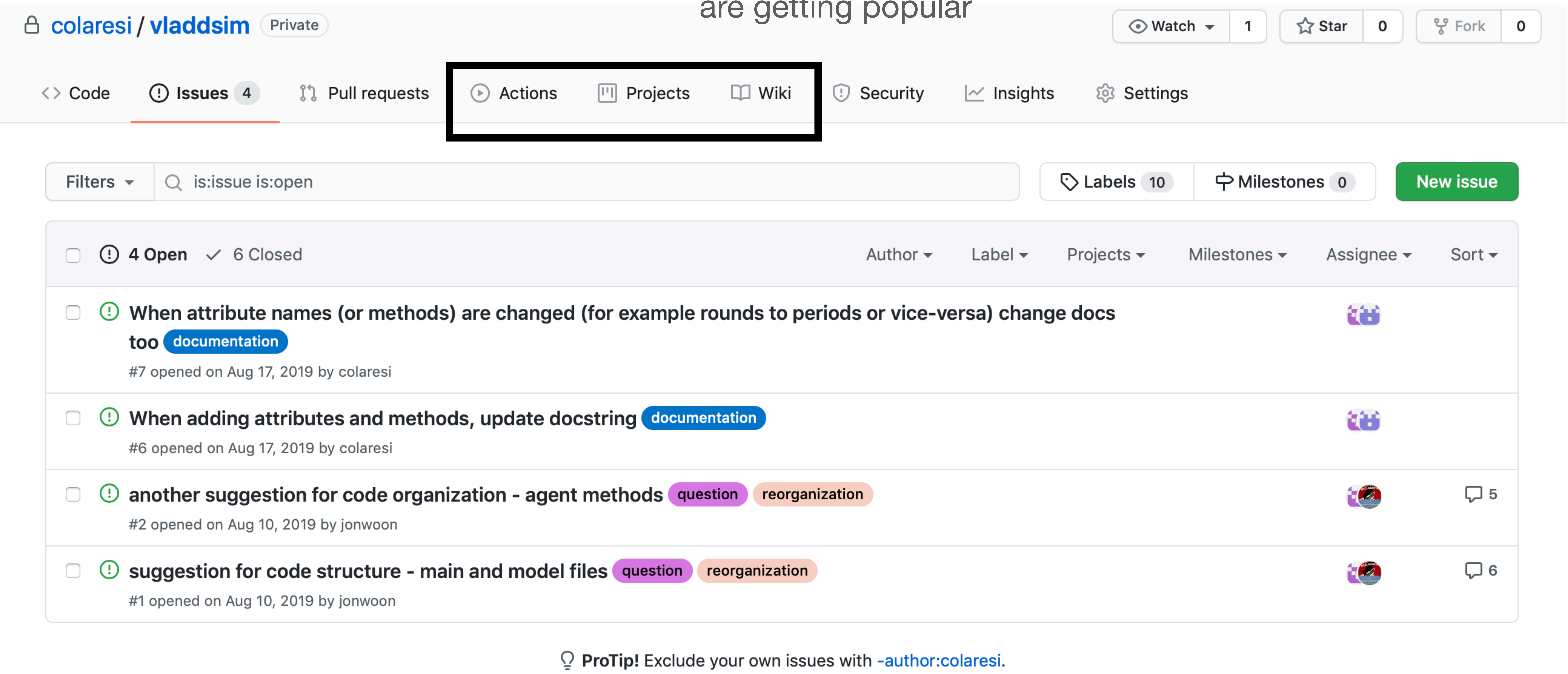
ProTip! Exclude your own issues with -author:colaresi.

Github

Pull requests

Actions: Run some code automatically

I use these less, but actions are getting popular



colaresi / vladdsim Private

is:issue is:open

4 Open ✓ 6 Closed

When attribute names (or methods) are changed (for example rounds to periods or vice-versa) change docs too documentation
#7 opened on Aug 17, 2019 by colaresi

When adding attributes and methods, update docstring documentation
#6 opened on Aug 17, 2019 by colaresi

another suggestion for code organization - agent methods question reorganization 5
#2 opened on Aug 10, 2019 by jonwoon

suggestion for code structure - main and model files question reorganization 6
#1 opened on Aug 10, 2019 by jonwoon

ProTip! Exclude your own issues with -author:colaresi.