

Github

Sven Schippkus – SPIN Friday Seminar – 2022-01-21



<https://xkcd.com/1597/>

THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.



<https://xkcd.com/1597/>

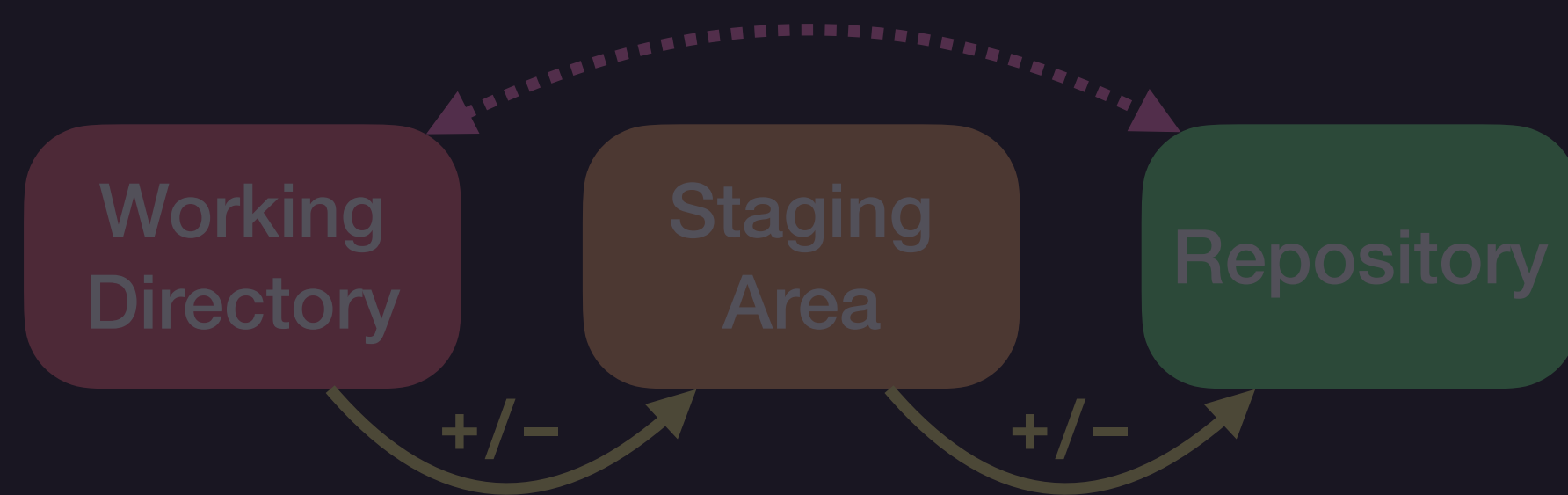
The problem git solves

Documents

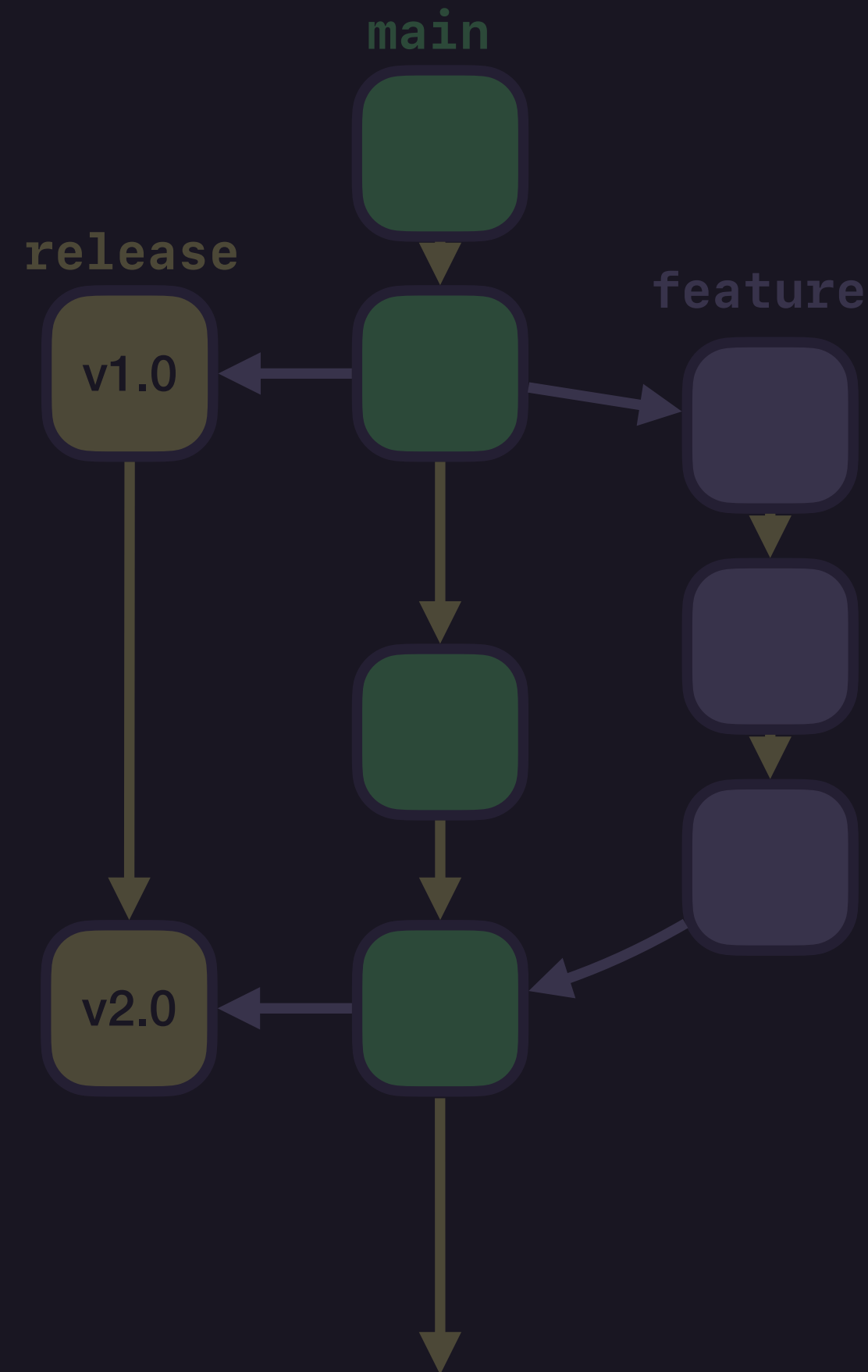
- 📄 Alland_DRAFT_2018_12_19_hh_01.doc
- 📄 Alland_DRAFT_1.docx
- 📄 Alland_DRAFT_2018_12_13_ZD.docx
- 📄 Alland_DRAFT_2018_12_13.docx
- 📄 Alland_DRAFT_2018_12_18.docx
- 📄 Alland_DRAFT_2018_12_19.docx
- 📄 Alland_DRAFT_2018_12_20_clean.docx
- 📄 Alland_DRAFT_2018_12_20-1.docx
- 📄 Alland_DRAFT_2018_12_20.docx

Overview

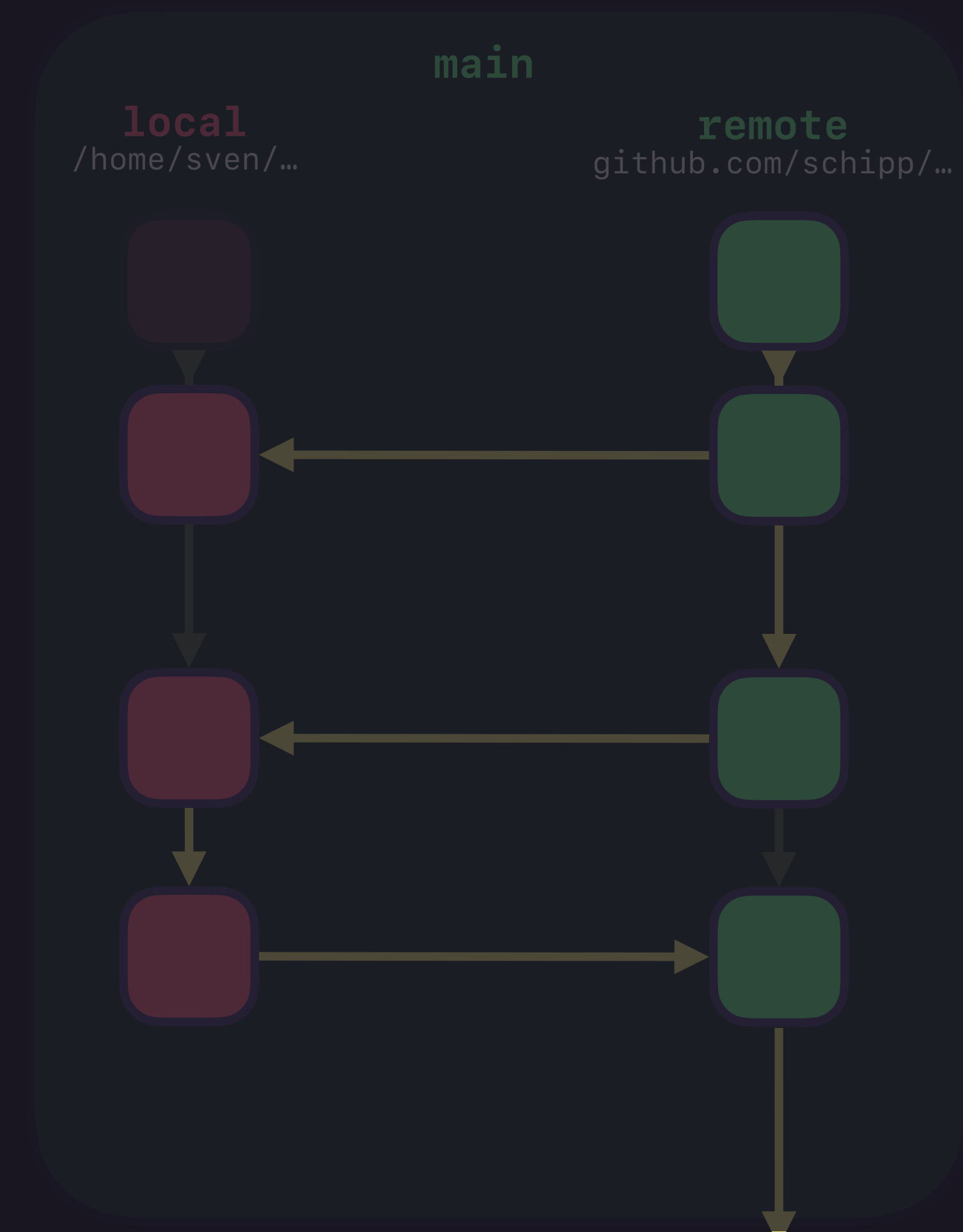
basics



branching



remote

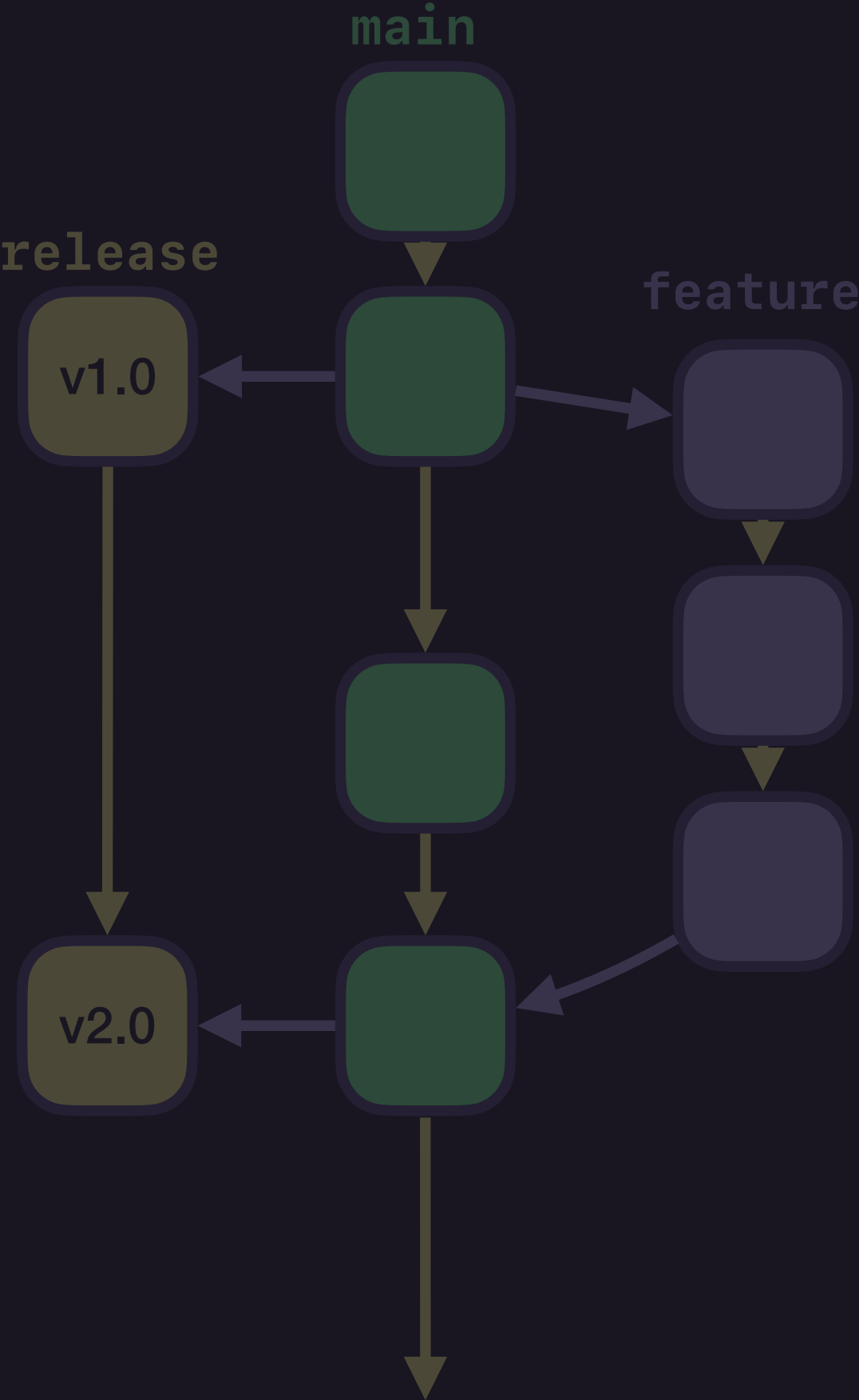


Overview

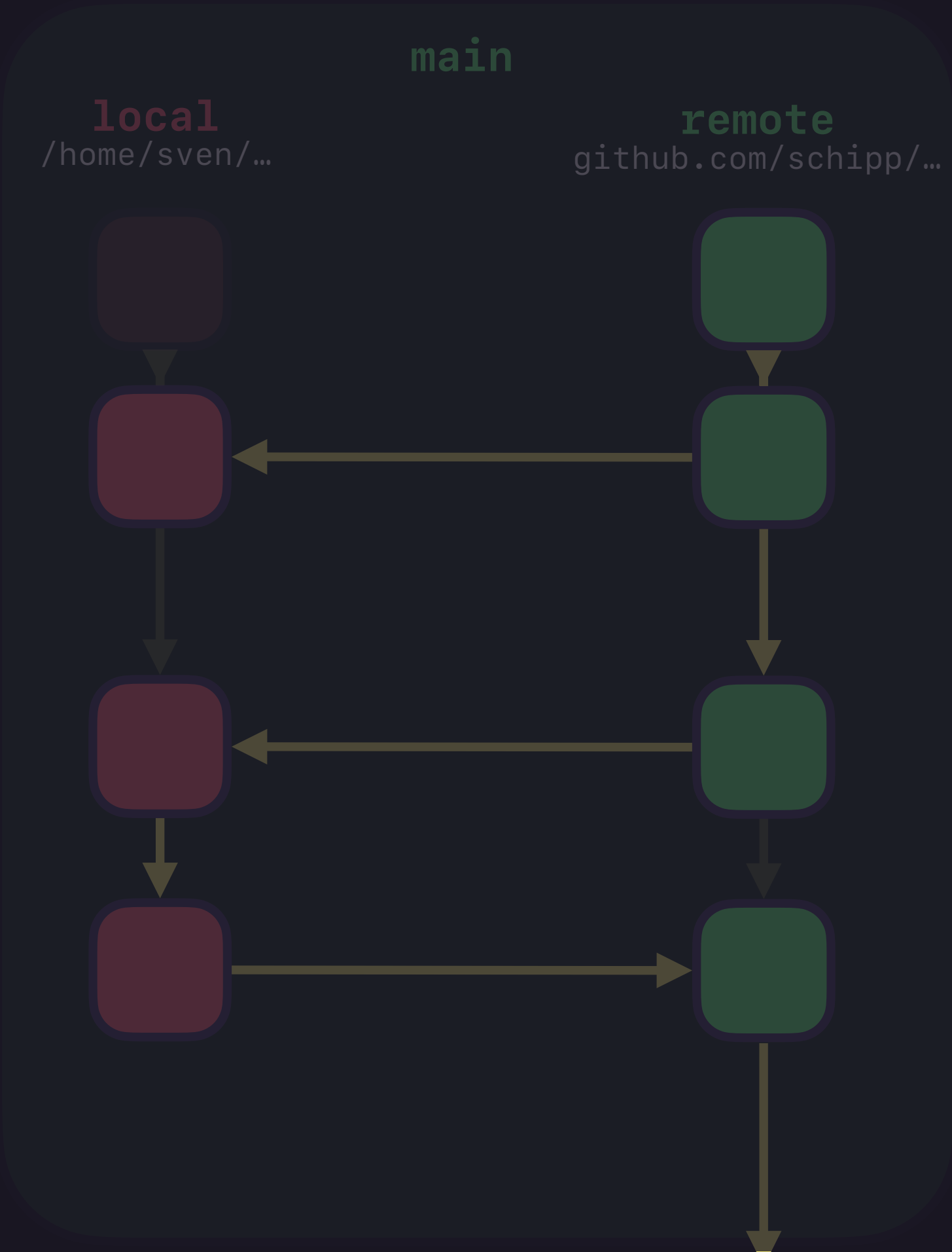
basics



branching



remote



Git

the basics

Working
Directory

```
~/Documents/output/workshops/2022/01_SPIN_git/demo ..... ☆ seis395 13:16:47  
> █
```

Git

the basics

Working
Directory

```
> ls -al
total 8
drwxr-xr-x@ 3 sven  staff   96 Jan 17 13:15 .
drwxr-xr-x@ 4 sven  staff  128 Jan 17 13:17 ..
-rw-r--r--  1 sven  staff   16 Jan 17 13:09 README.md

~/Documents/output/workshops/2022/01_SPIN_git/demo ..... ☆ seis395 13:17:08
> █
```


Git

the basics

Working
Directory

Repository

`git init` - initialise **repository** in **working directory**

```
> ls -al
total 8
drwxr-xr-x@ 3 sven  staff   96 Jan 17 13:15 .
drwxr-xr-x@ 4 sven  staff  128 Jan 17 13:17 ..
-rw-r--r--  1 sven  staff   16 Jan 17 13:09 README.md
> git init
Initialized empty Git repository in /Users/sven/Documents/output/workshops/2022/01_SPIN_git/demo/.git/

~/Doc/output/workshops/2022/01_SPIN_git/demo main ?1 ... ☆ seis395 13:17:32
> █
```

Git

the basics

Working
Directory

Repository

`git init` - initialise **repository** in **working directory**

```
> ls -al
total 8
drwxr-xr-x@ 3 sven  staff   96 Jan 17 13:15 .
drwxr-xr-x@ 4 sven  staff  128 Jan 17 13:17 ..
-rw-r--r--  1 sven  staff   16 Jan 17 13:09 README.md
> git init
Initialized empty Git repository in /Users/sven/Documents/output/workshops/2022/01_SPIN_git/demo/.git/
> ls -al
total 8
drwxr-xr-x@ 4 sven  staff  128 Jan 17 13:17 .
drwxr-xr-x@ 4 sven  staff  128 Jan 17 13:18 ..
drwxr-xr-x@ 9 sven  staff  288 Jan 17 13:17 .git
-rw-r--r--  1 sven  staff   16 Jan 17 13:09 README.md

~/Doc/output/workshops/2022/01_SPIN_git/demo main ?1 ... ☆ seis395 13:18:18
> █
```


Git

the basics

Working
Directory

Repository

`git init` - initialise **repository** in **working directory**


`git status` - check current status of git

```
> ls -al
total 8
drwxr-xr-x@ 3 sven  staff   96 Jan 17 13:15 .
drwxr-xr-x@ 4 sven  staff  128 Jan 17 13:17 ..
-rw-r--r--  1 sven  staff   16 Jan 17 13:09 README.md
> git init
Initialized empty Git repository in /Users/sven/Documents/output/workshops/2022/01_SPIN_git/demo/.git/
> ls -al
total 8
drwxr-xr-x@ 4 sven  staff  128 Jan 17 13:17 .
drwxr-xr-x@ 4 sven  staff  128 Jan 17 13:18 ..
drwxr-xr-x@ 9 sven  staff  288 Jan 17 13:17 .git
-rw-r--r--  1 sven  staff   16 Jan 17 13:09 README.md
> git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md

nothing added to commit but untracked files present (use "git add" to track)
~/Doc/output/workshops/2022/01_SPIN_git/demo main ?1 ... ☆ seis395 13:18:50
> 
```



Git

the basics

`git init` - initialise **repository** in **working directory**

`git status` - check current status of git

`git add` - move **changes** to **staging area**



```
> ls -al
total 8
drwxr-xr-x@ 3 sven  staff   96 Jan 17 13:15 .
drwxr-xr-x@ 4 sven  staff  128 Jan 17 13:17 ..
-rw-r--r--  1 sven  staff   16 Jan 17 13:09 README.md
> git init
Initialized empty Git repository in /Users/sven/Documents/output/workshops/2022/01_SPIN_git/demo/.git/
> ls -al
total 8
drwxr-xr-x@ 4 sven  staff  128 Jan 17 13:17 .
drwxr-xr-x@ 4 sven  staff  128 Jan 17 13:18 ..
drwxr-xr-x@ 9 sven  staff  288 Jan 17 13:17 .git
-rw-r--r--  1 sven  staff   16 Jan 17 13:09 README.md
> git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md

nothing added to commit but untracked files present (use "git add" to track)
> git add README.md

~/Doc/output/workshops/2022/01_SPIN_git/demo main +1 ... ☆ seis395 13:19:54
> █
```

A blue arrow points to the "+1" in the status bar of the terminal output.

Git

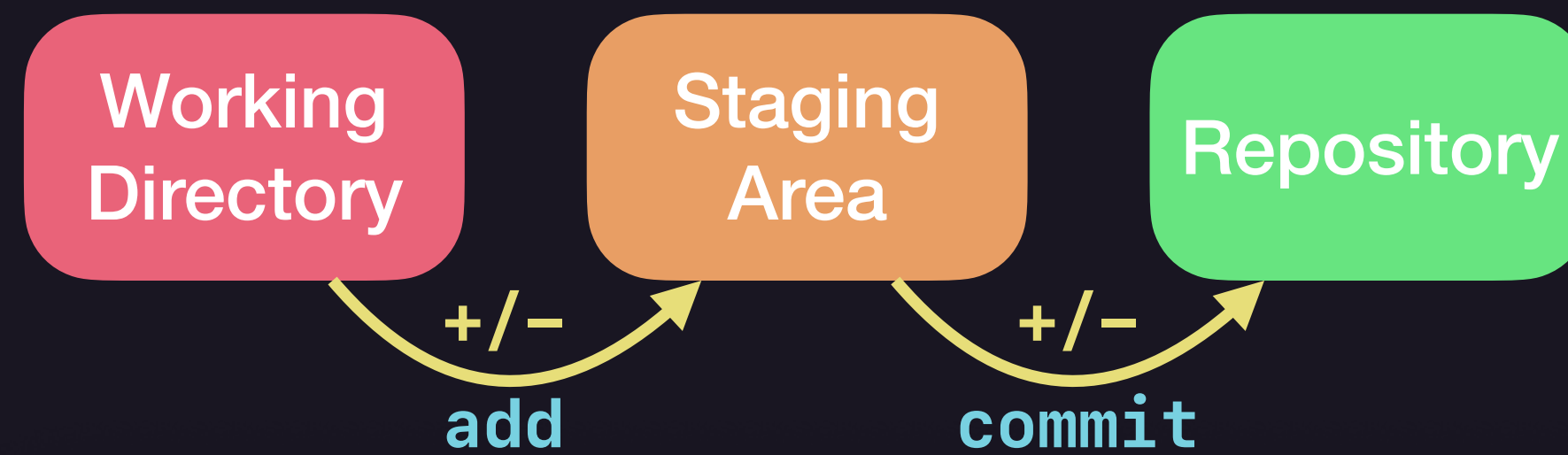
the basics

`git init` - initialise **repository** in **working directory**

`git status` - check current status of git

`git add` - move **changes** to **staging area**

`git commit` - move **changes** to **repository**



```
drwxr-xr-x@ 9 sven  staff  288 Jan 17 13:17 .git
-rw-r--r--  1 sven  staff   16 Jan 17 13:09 README.md
> git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md

nothing added to commit but untracked files present (use "git add" to track)
> git add README.md
> git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md

> git commit -m "first commit"
[main (root-commit) 16f03ef] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md

~/Doc/output/workshops/2022/01_SPIN_git/demo main ..... ☆ seis395 13:21:06
> █
```


Git

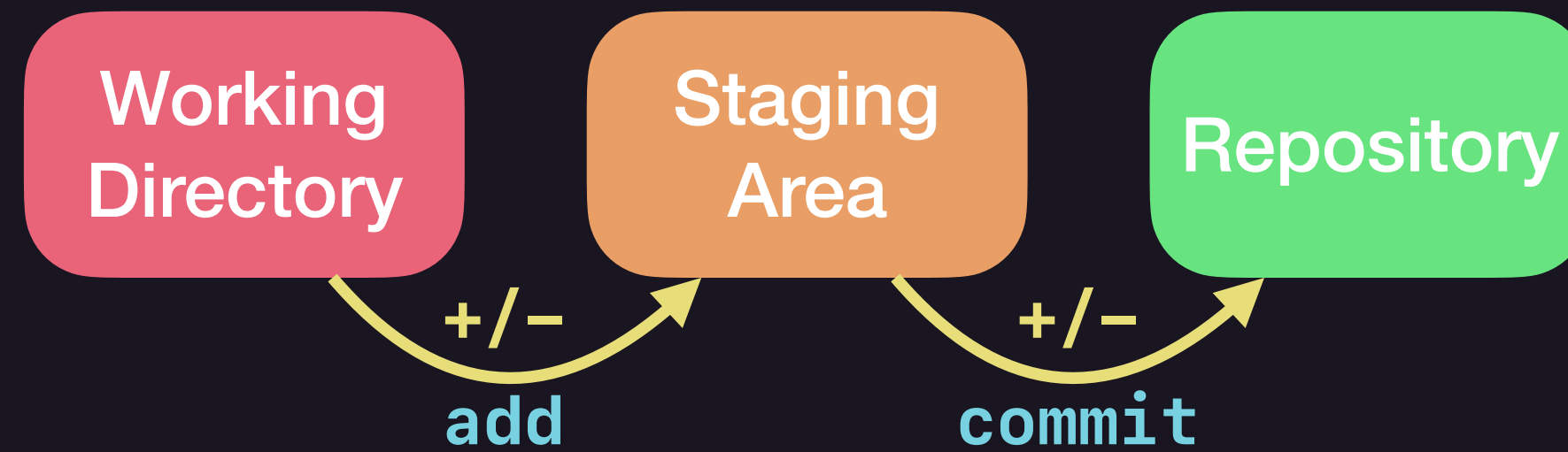
the basics

`git init` - initialise **repository** in **working directory**

`git status` - check current status of git

`git add` - move **changes** to **staging area**

`git commit` - move **changes** to **repository**



```
On branch main
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  README.md

nothing added to commit but untracked files present (use "git add" to track)
> git add README.md
> git status
On branch main
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   README.md

> git commit -m "first commit"
[main (root-commit) 16f03ef] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
> git status
On branch main
nothing to commit, working tree clean

~/Doc/output/workshops/2022/01_SPIN_git/demo main ..... ☆ seis395 13:21:23
> █
```

Git

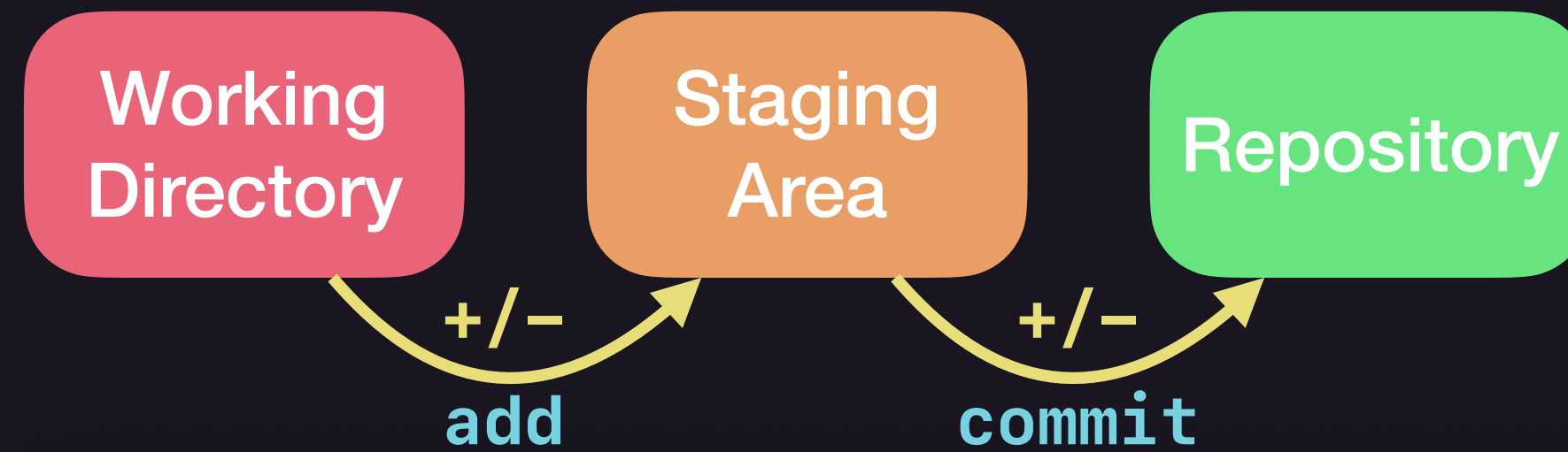
the basics

`git init` - initialise **repository** in **working directory**

`git status` - check current status of git

`git add` - move **changes** to **staging area**

`git commit` - move **changes** to **repository**



```
> git add README.md
> git status
On branch main


No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.md

> git commit -m "first commit"
[main (root-commit) 16f03ef] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
> git status
On branch main
nothing to commit, working tree clean
> echo "some_text" >> README.md
> git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

~/Doc/output/workshops/2022/01_SPIN_git/demo main !1 ... ☆ seis395 13:22:07
> █
```



Git

the basics

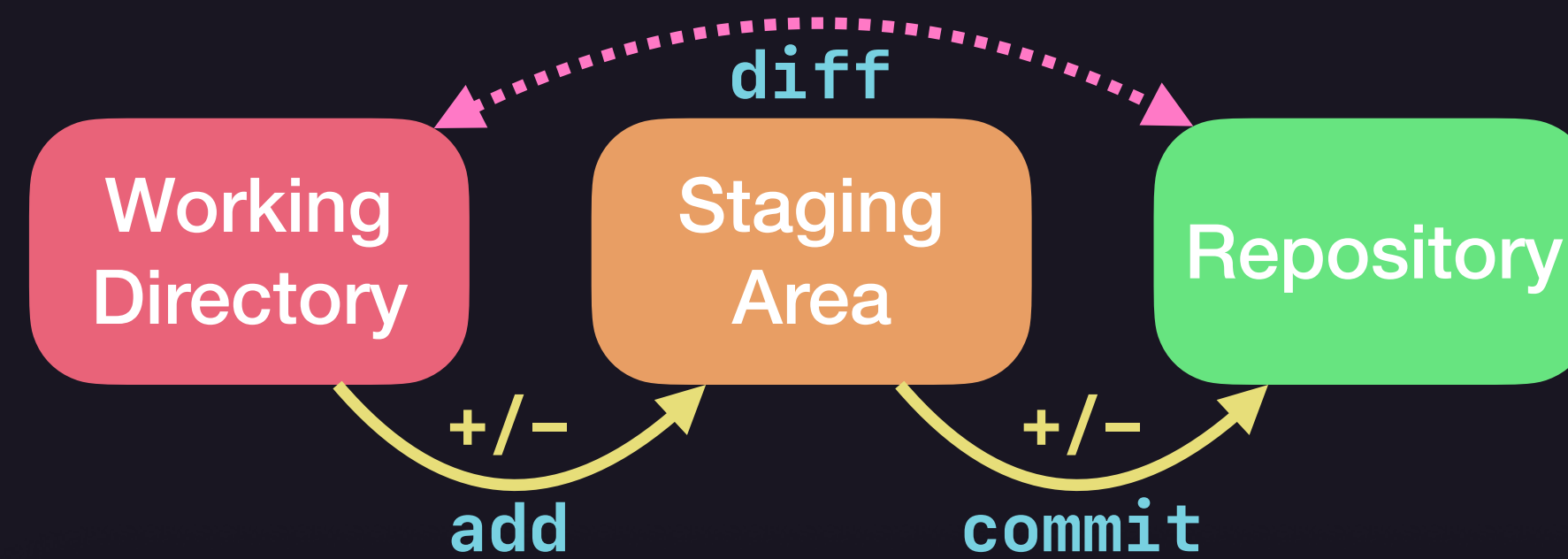
`git init` - initialise **repository** in **working directory**

`git status` - check current status of git

`git add` - move **changes** to **staging area**

`git commit` - move **changes** to **repository**

`git diff` - compare differences between **working directory** and **repository**



```
> git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.md

> git commit -m "first commit"
[main (root-commit) 16f03ef] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
> git status
On branch main
nothing to commit, working tree clean
> echo "some_text" >> README.md
> git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
> git diff

~/Doc/ou/workshops/2022/01_SPIN_git/demo main !1 ···· 4s ☆ seis395 13:22:47
> █
```

Git

the basics

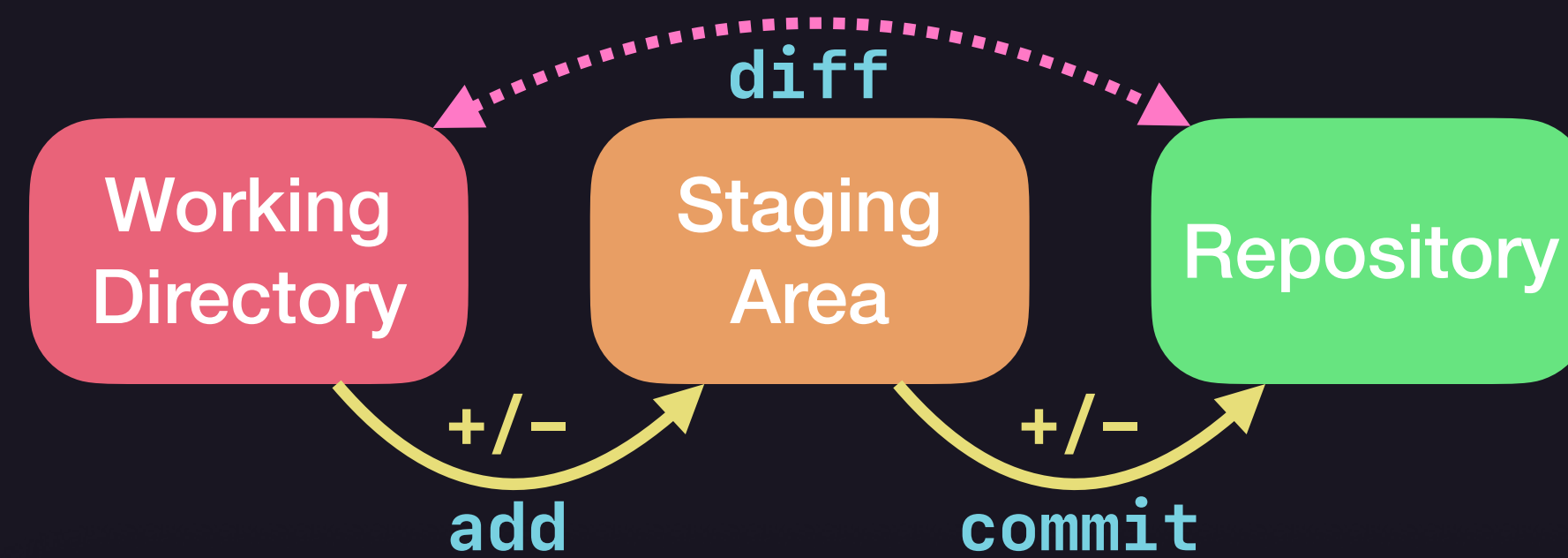
`git init` - initialise **repository** in **working directory**

`git status` - check current status of git

`git add` - move **changes** to **staging area**

`git commit` - move **changes** to **repository**

`git diff` - compare differences between **working directory** and **repository**



```
diff --git a/README.md b/README.md
index 6077931..5675c3c 100644
--- a/README.md
+++ b/README.md
@@ -1,2 @@
 # SPIN git demo
+some_text
(END)
```

Git

the basics

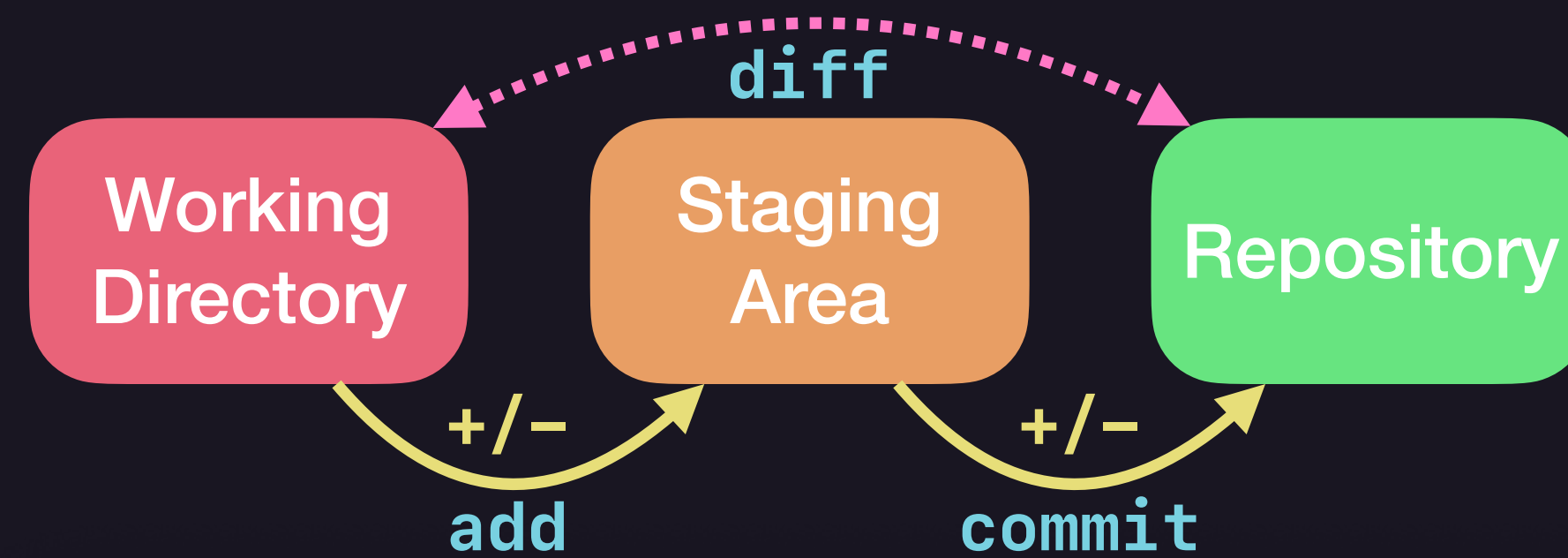
`git init` - initialise **repository** in **working directory**

`git status` - check current status of git

`git add` - move **changes** to **staging area**

`git commit` - move **changes** to **repository**

`git diff` - compare differences between **working directory** and **repository**



```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.md

> git commit -m "first commit"
[main (root-commit) 16f03ef] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
> git status
On branch main
nothing to commit, working tree clean
> echo "some_text" >> README.md
> git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
> git diff
> git diff
> git commit -a -m "update README"
[main 4293625] update README
 1 file changed, 1 insertion(+)

~/Doc/output/workshops/2022/01_SPIN_git/demo main ..... ☆ seis395 13:24:09
> █
```


Git

the basics

`git init` - initialise **repository** in **working directory**

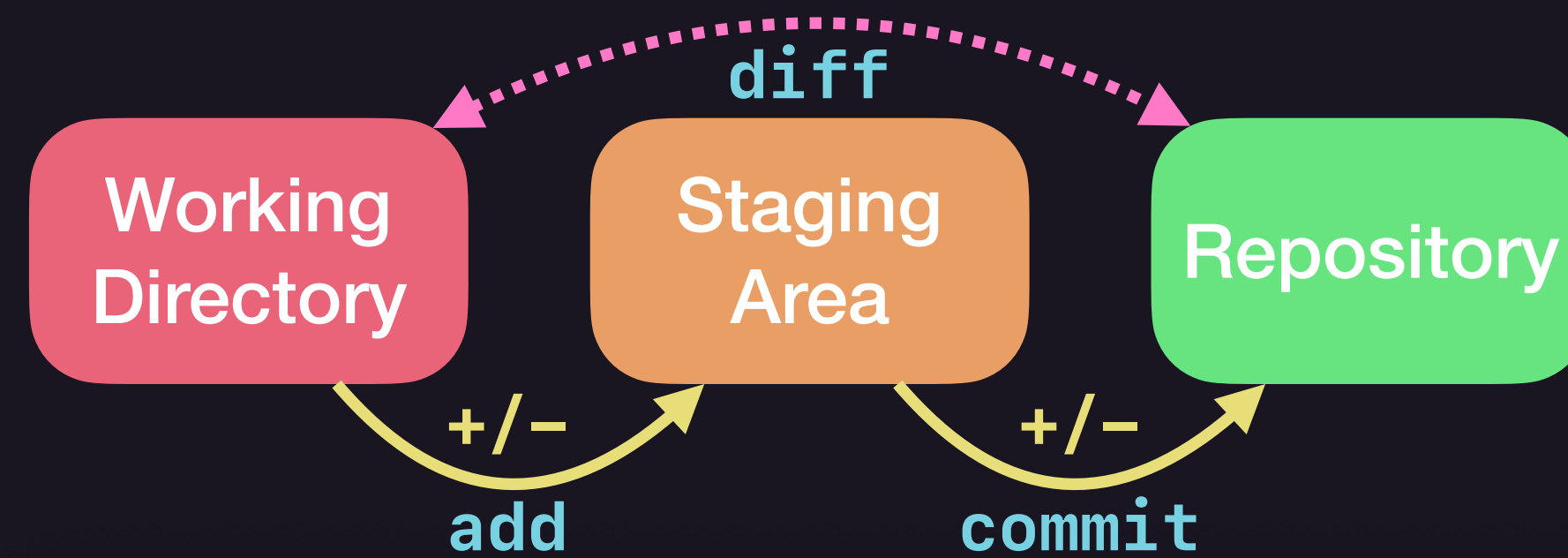
`git status` - check current status of git

`git add` - move **changes** to **staging area**

`git commit` - move **changes** to **repository**

`git diff` - compare differences between **working directory** and **repository**

The **staging area** exists to allow commit of changes on a granular level



```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.md

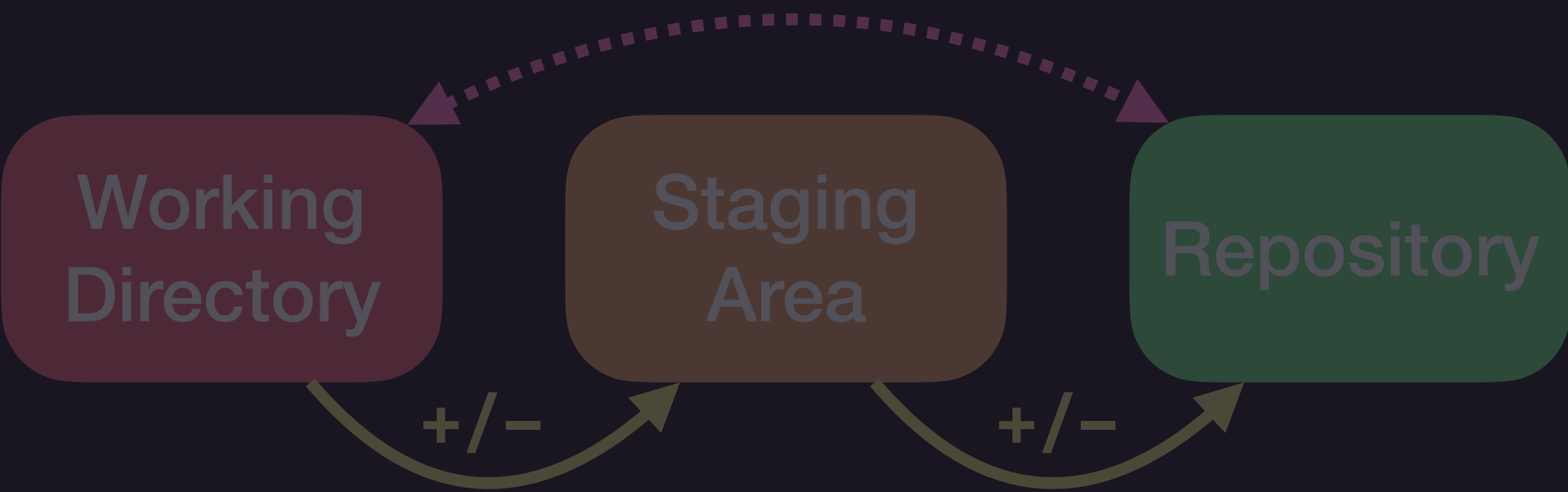
> git commit -m "first commit"
[main (root-commit) 16f03ef] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
> git status
On branch main
nothing to commit, working tree clean
> echo "some_text" >> README.md
> git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
> git diff
> git diff
> git commit -a -m "update README"
[main 4293625] update README
 1 file changed, 1 insertion(+)

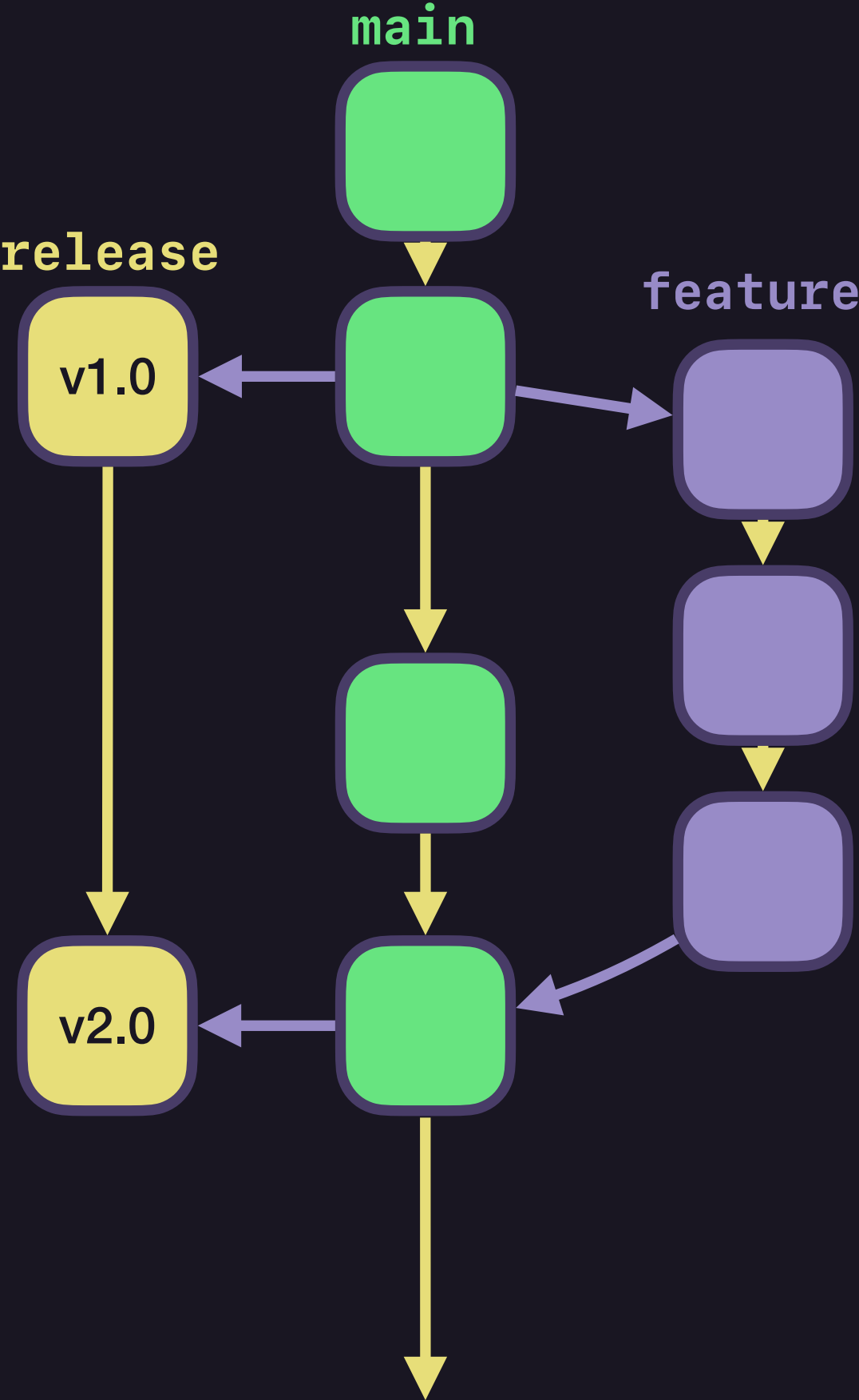
~/Doc/output/workshops/2022/01_SPIN_git/demo main ..... ☆ seis395 13:24:09
> █
```

Overview

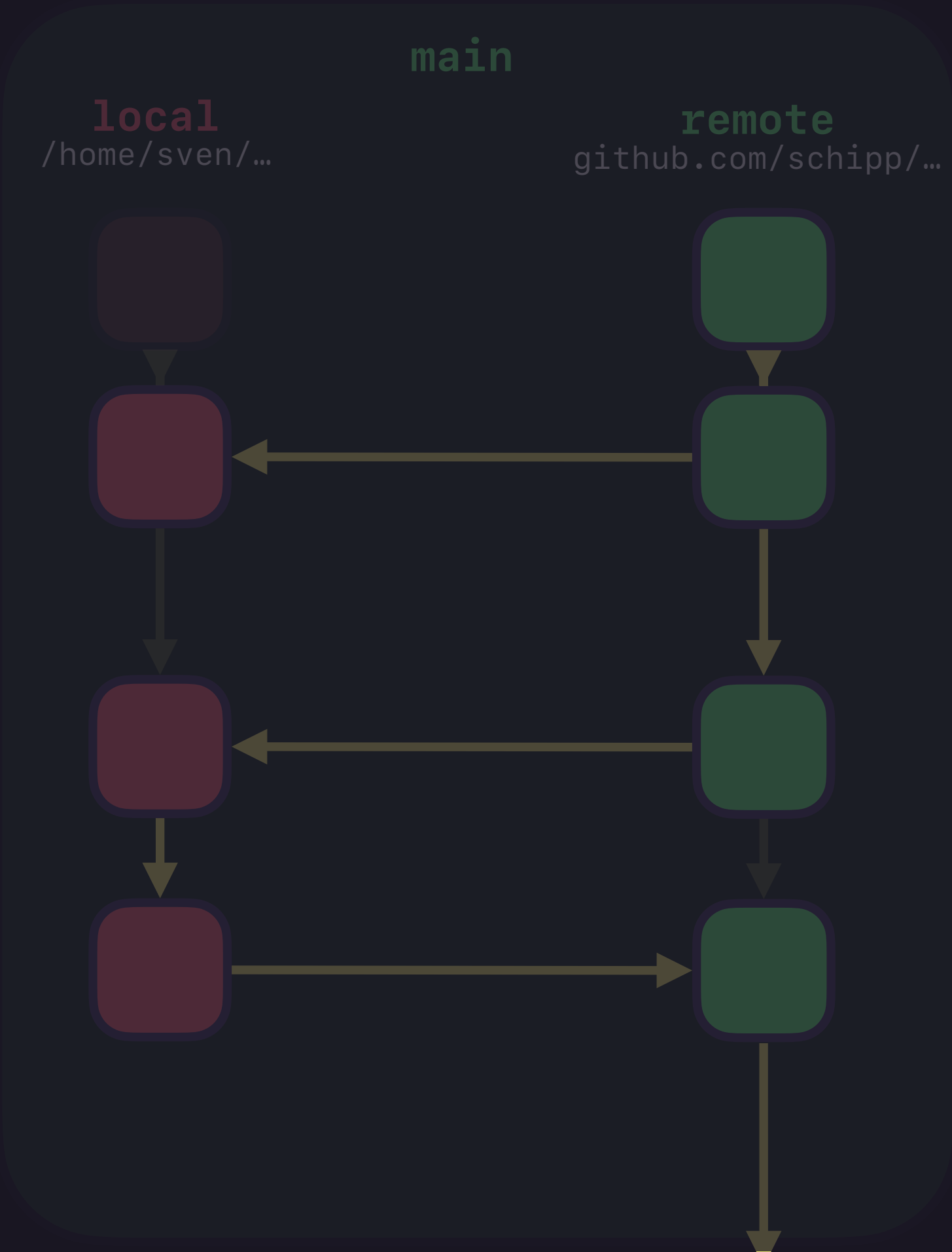
basics



branching

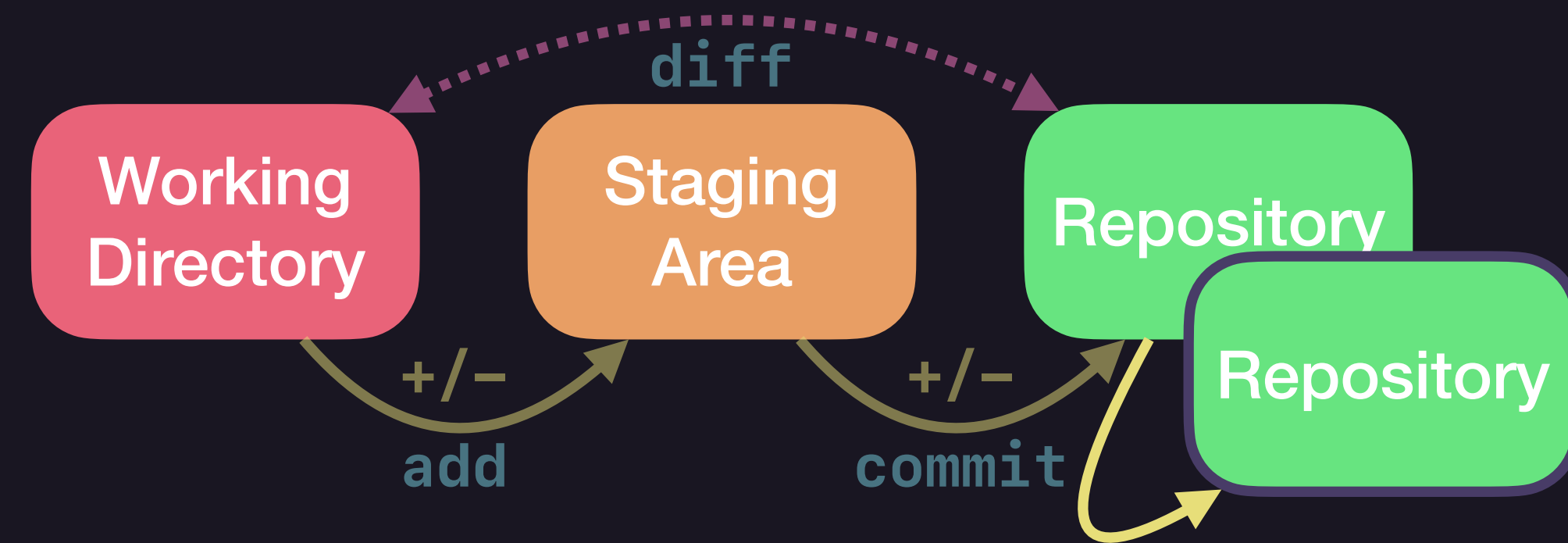


remote



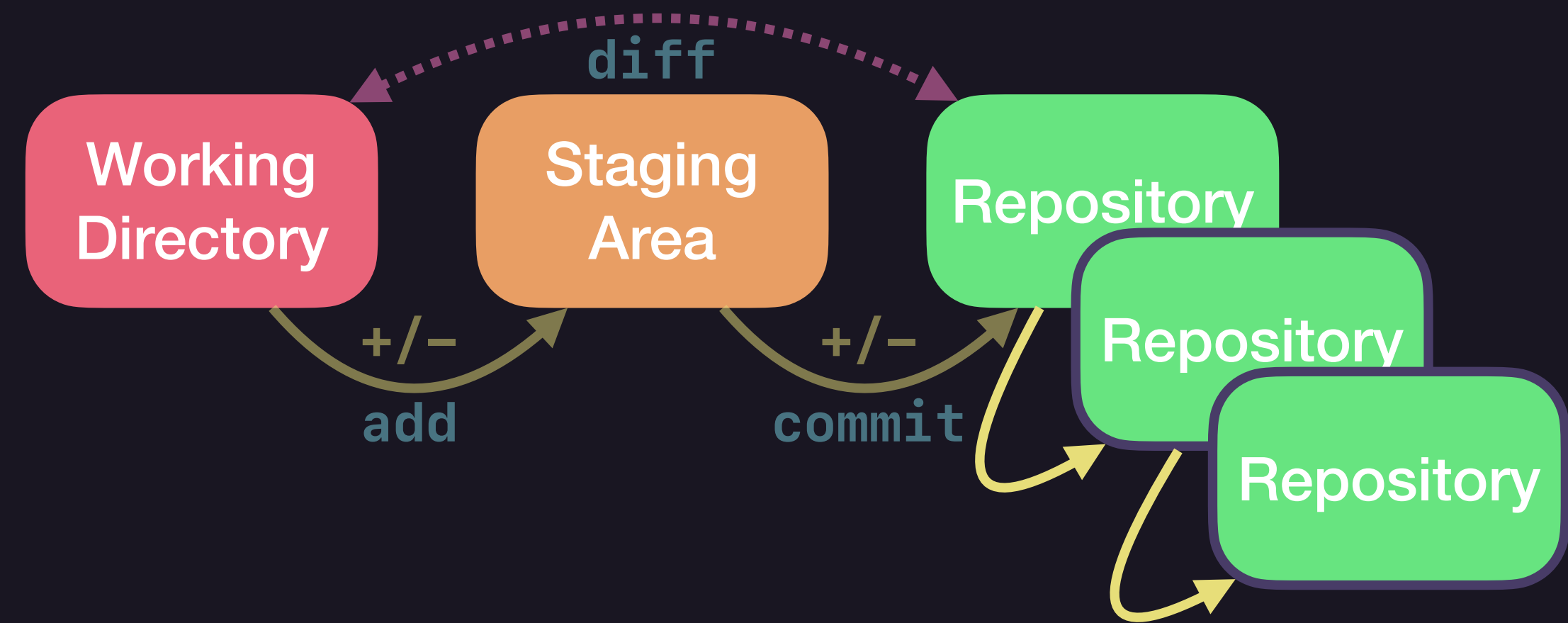
Git

branches



Git

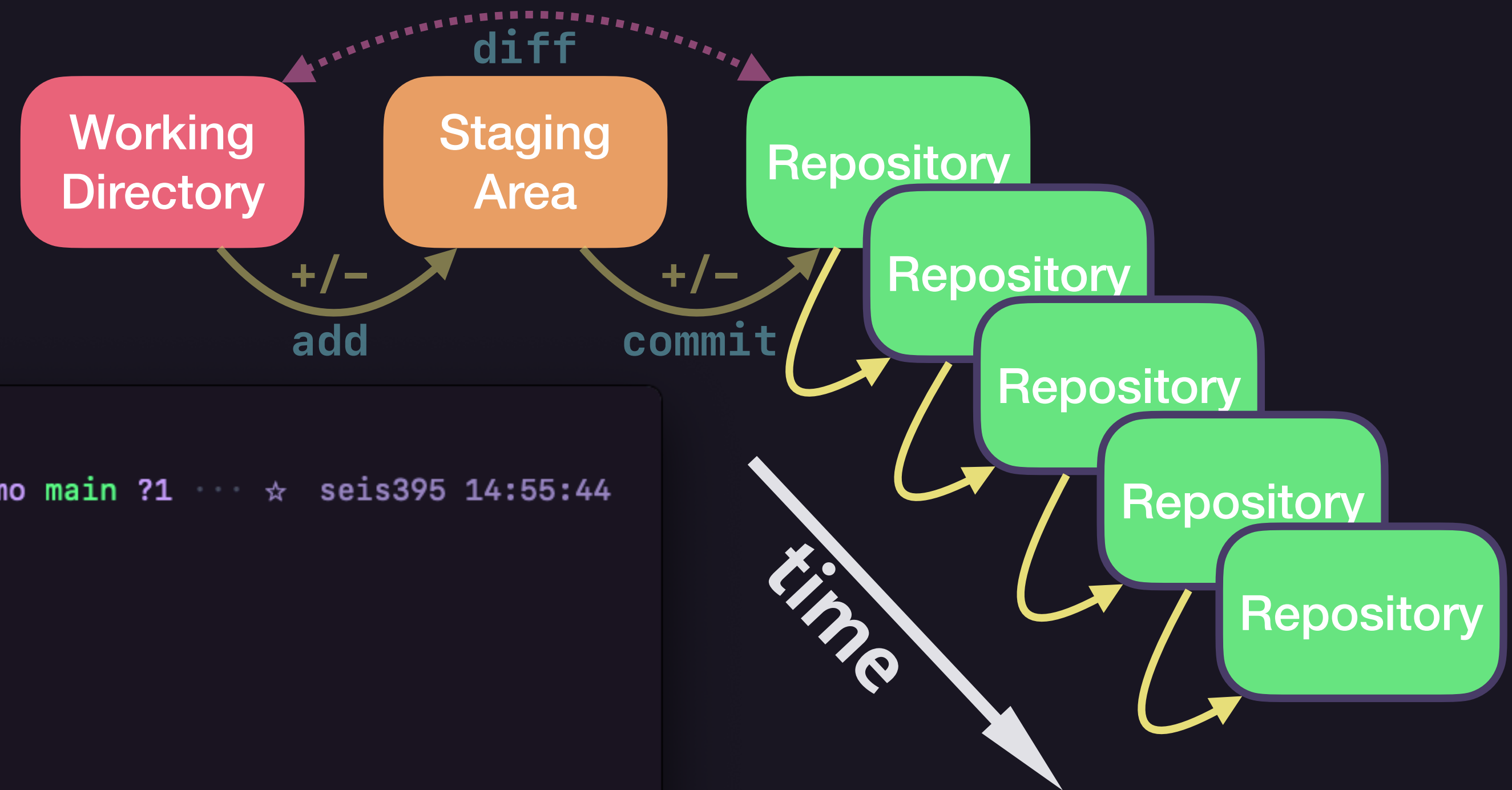
branches



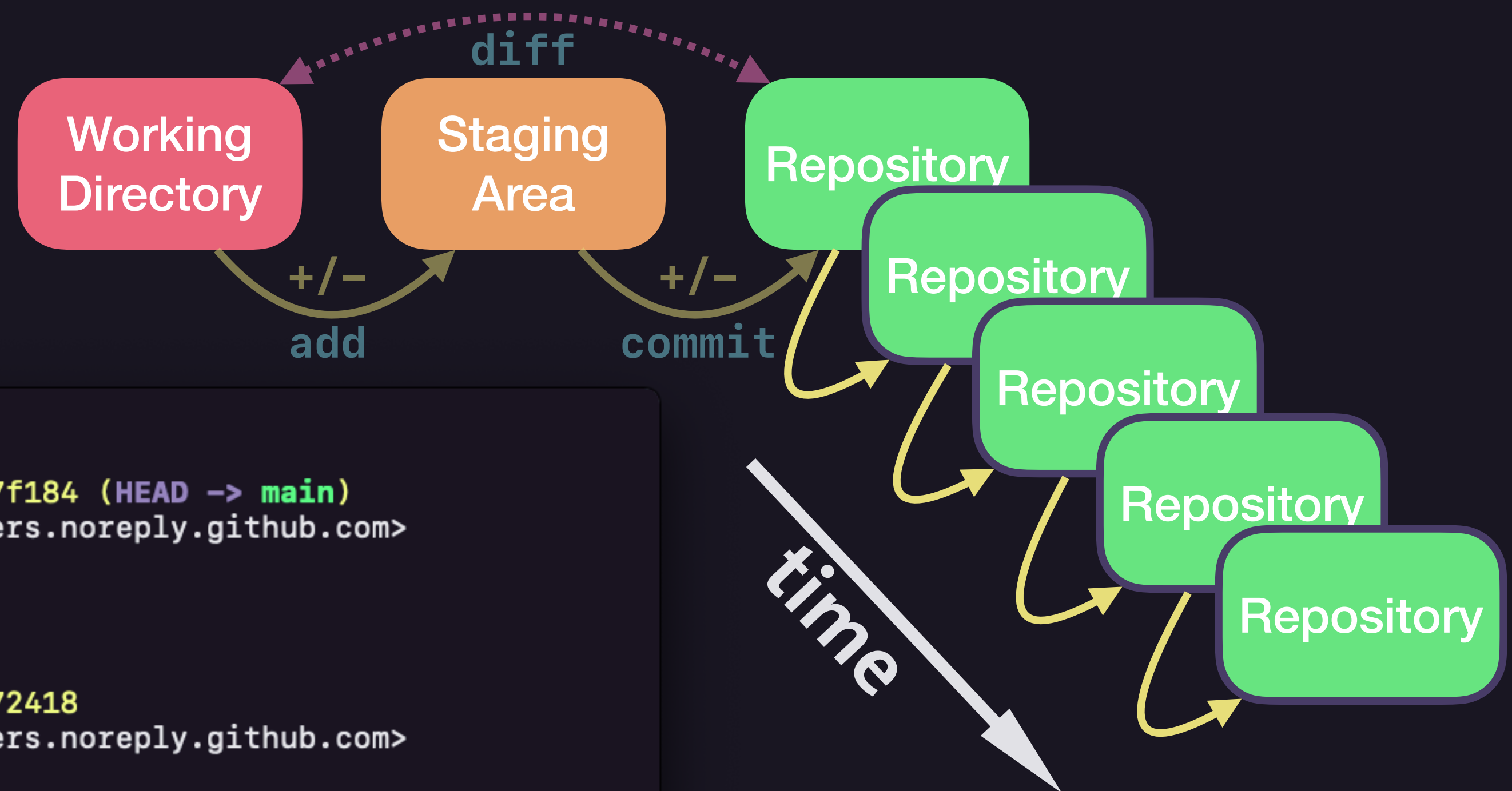
Git

branches

```
> git log  
~/Doc/output/workshops/2022/01_SPIN_git/demo main ?1 ... ☆ seis395 14:55:44  
> █
```



Git branches



```
commit e89e971015092bf09aca1ec80aea4106b8d7f184 (HEAD -> main)
Author: Sven Schippkus <26145768+schipp@users.noreply.github.com>
Date:   Mon Jan 17 13:51:13 2022 +0100

    add newfile1

commit 4293625c08ab3fb0d4d68789005cda092d072418
Author: Sven Schippkus <26145768+schipp@users.noreply.github.com>
Date:   Mon Jan 17 13:24:09 2022 +0100

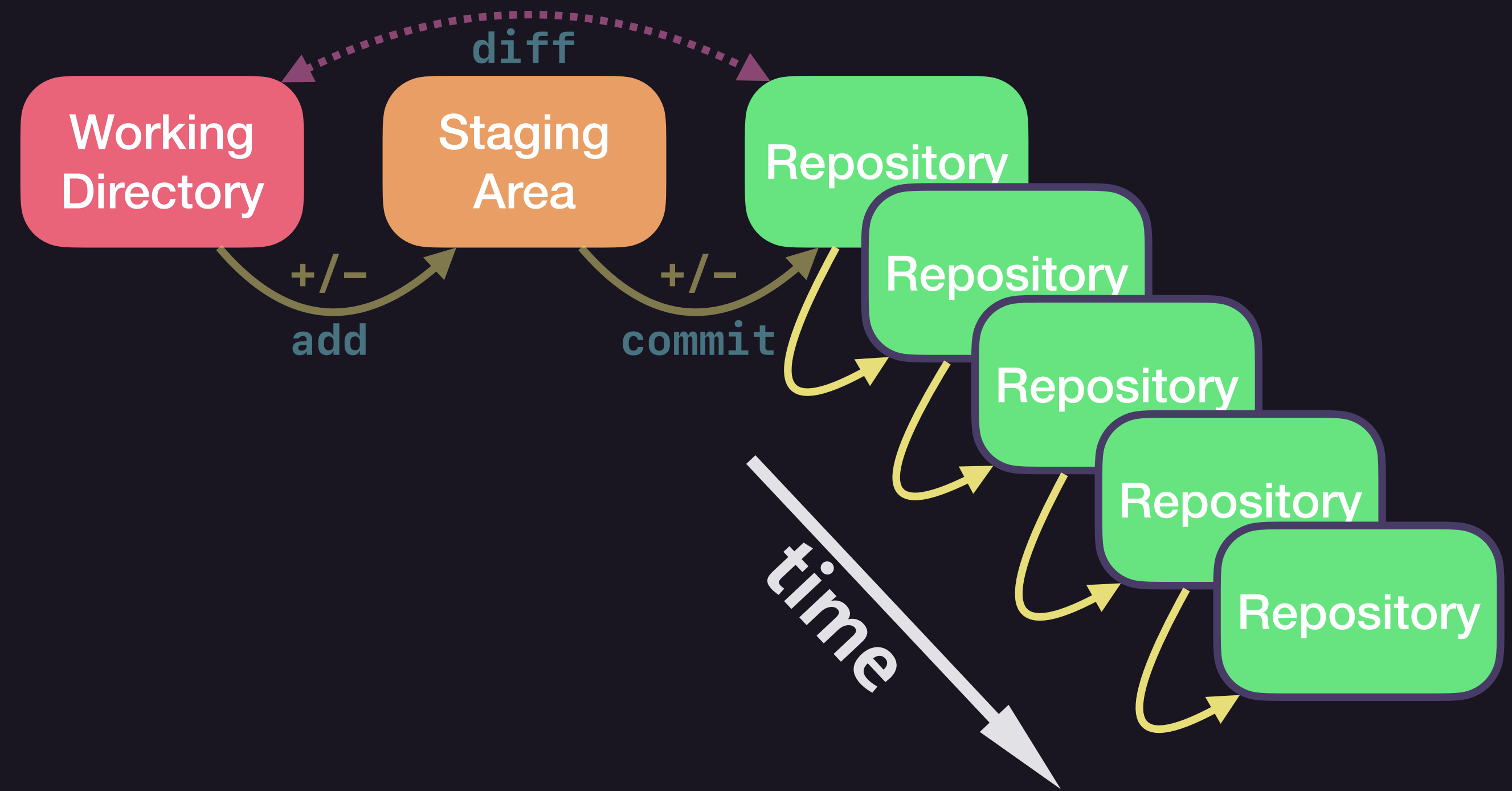
    update README

commit 16f03efaf3dd64ceb66551926a202e2406a3555f
Author: Sven Schippkus <26145768+schipp@users.noreply.github.com>
Date:   Mon Jan 17 13:21:06 2022 +0100

    first commit
(END)
```

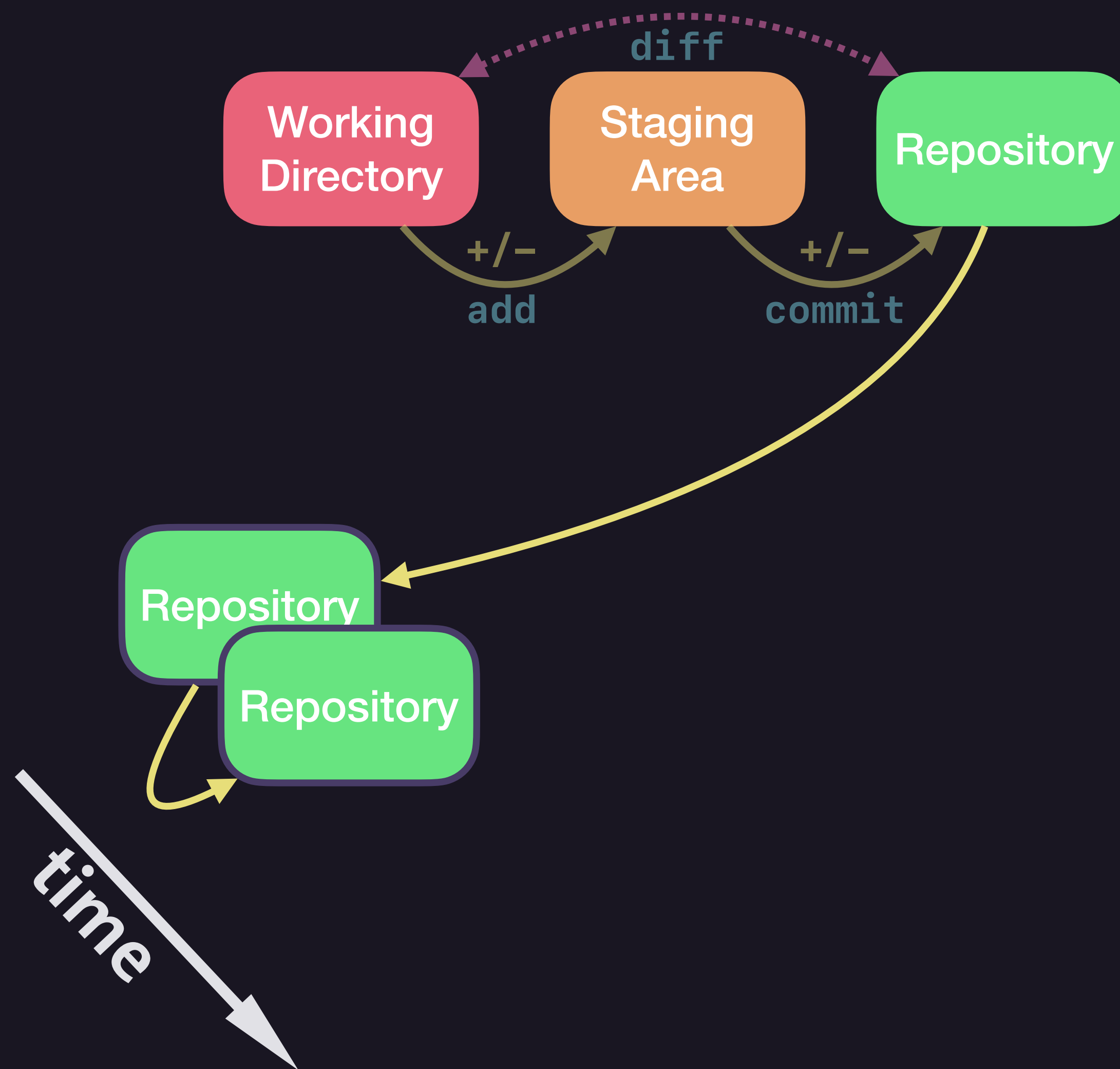
Git branches

`git log` - view history of **commits**



Git branches

`git log` - view history of **commits**

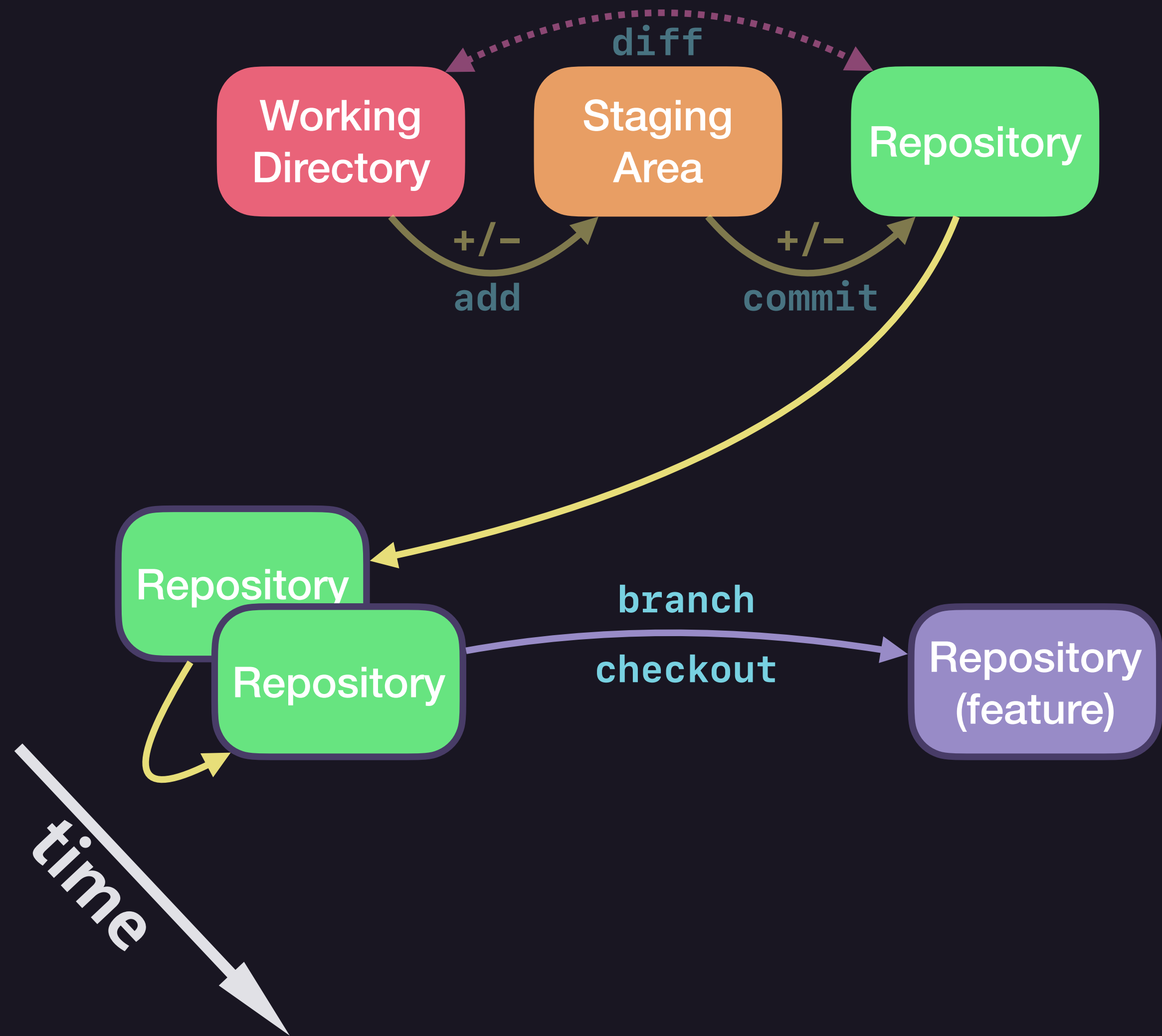


Git branches

`git log` - view history of **commits**

`git branch` - manage **branches**

`git checkout` - switch **branches**

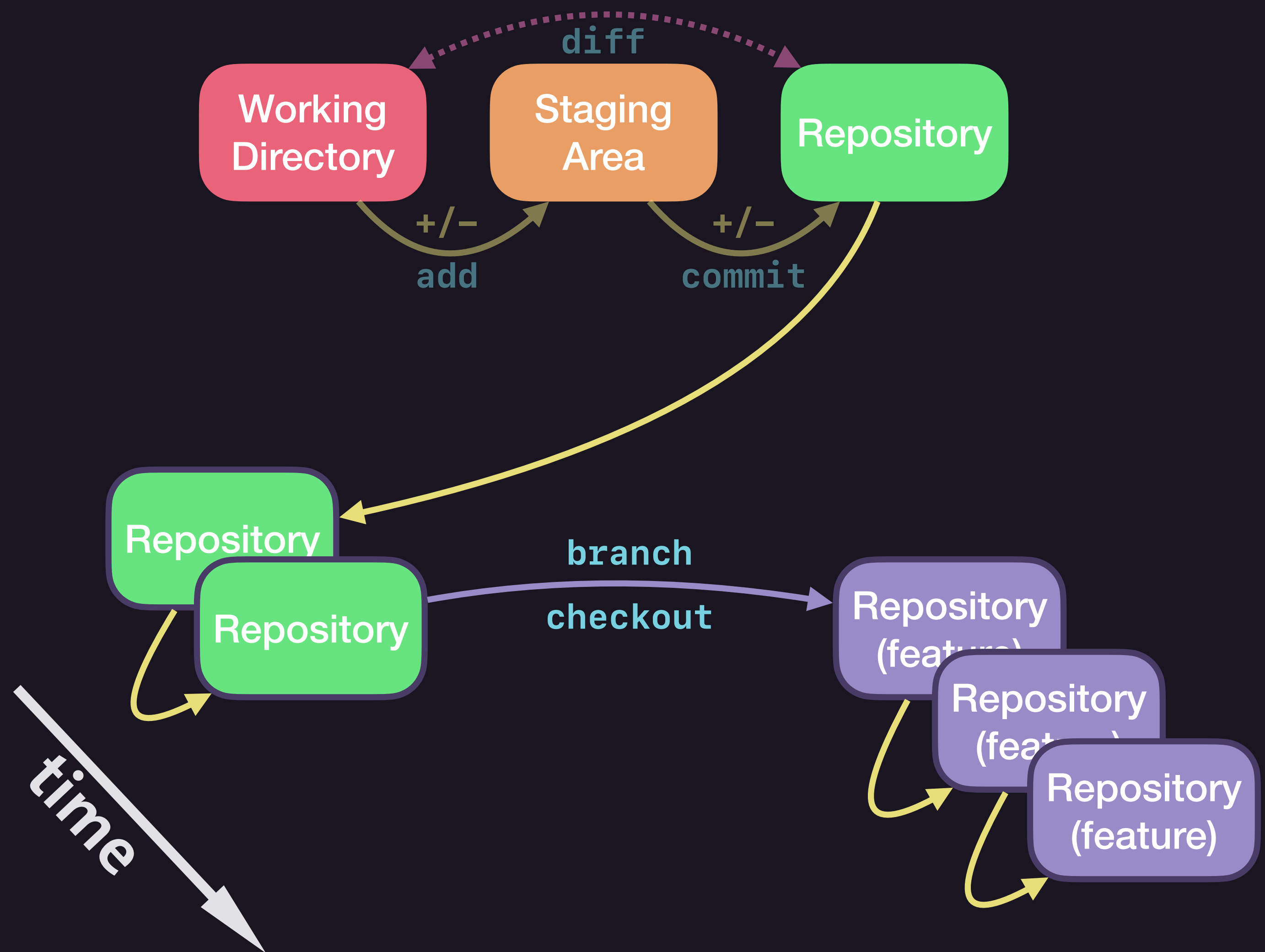


Git branches

git log - view history of **commits**

git branch - manage **branches**

git checkout - switch **branches**

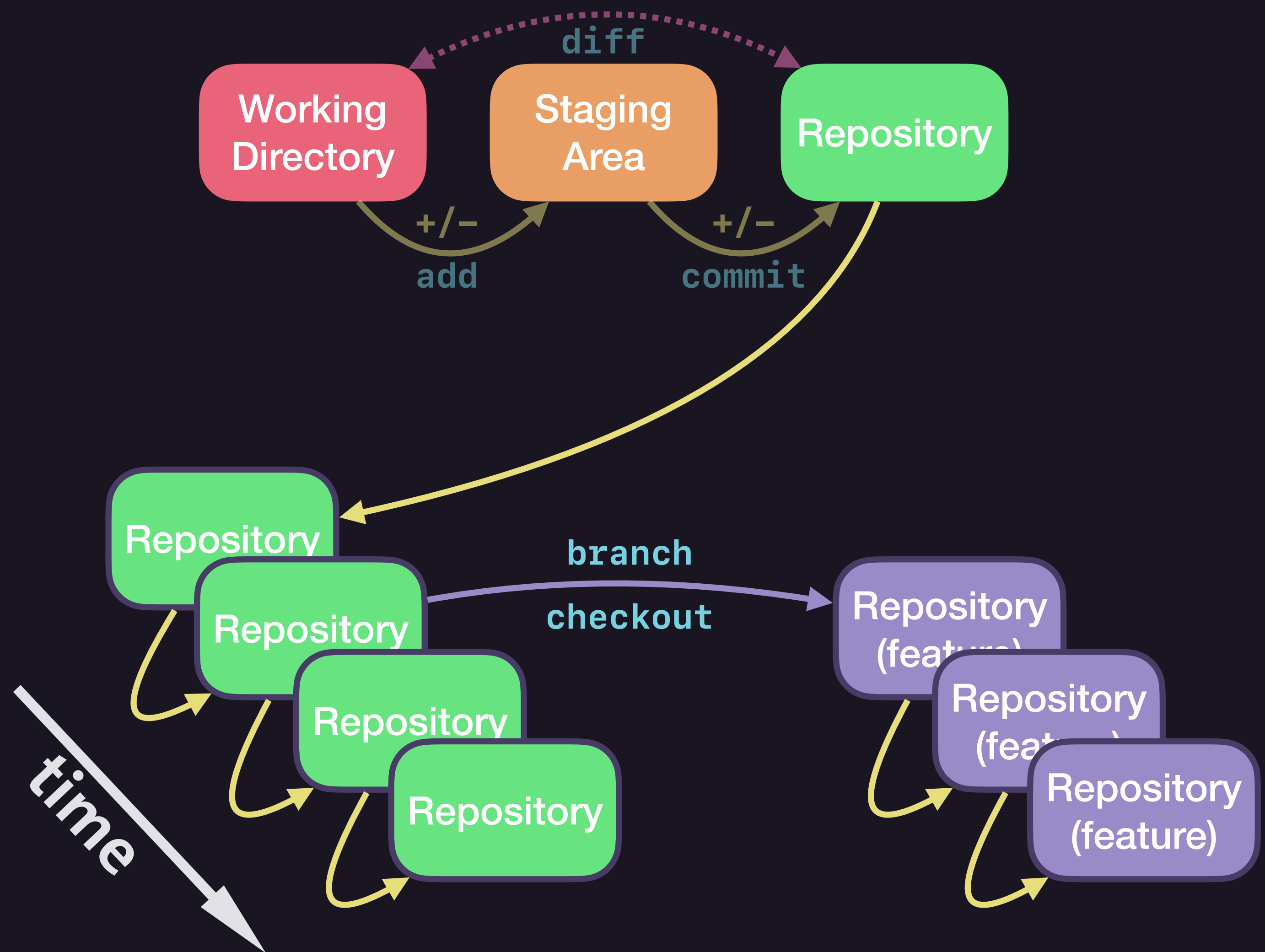


Git branches

git log - view history of **commits**

git branch - manage **branches**

git checkout - switch **branches**



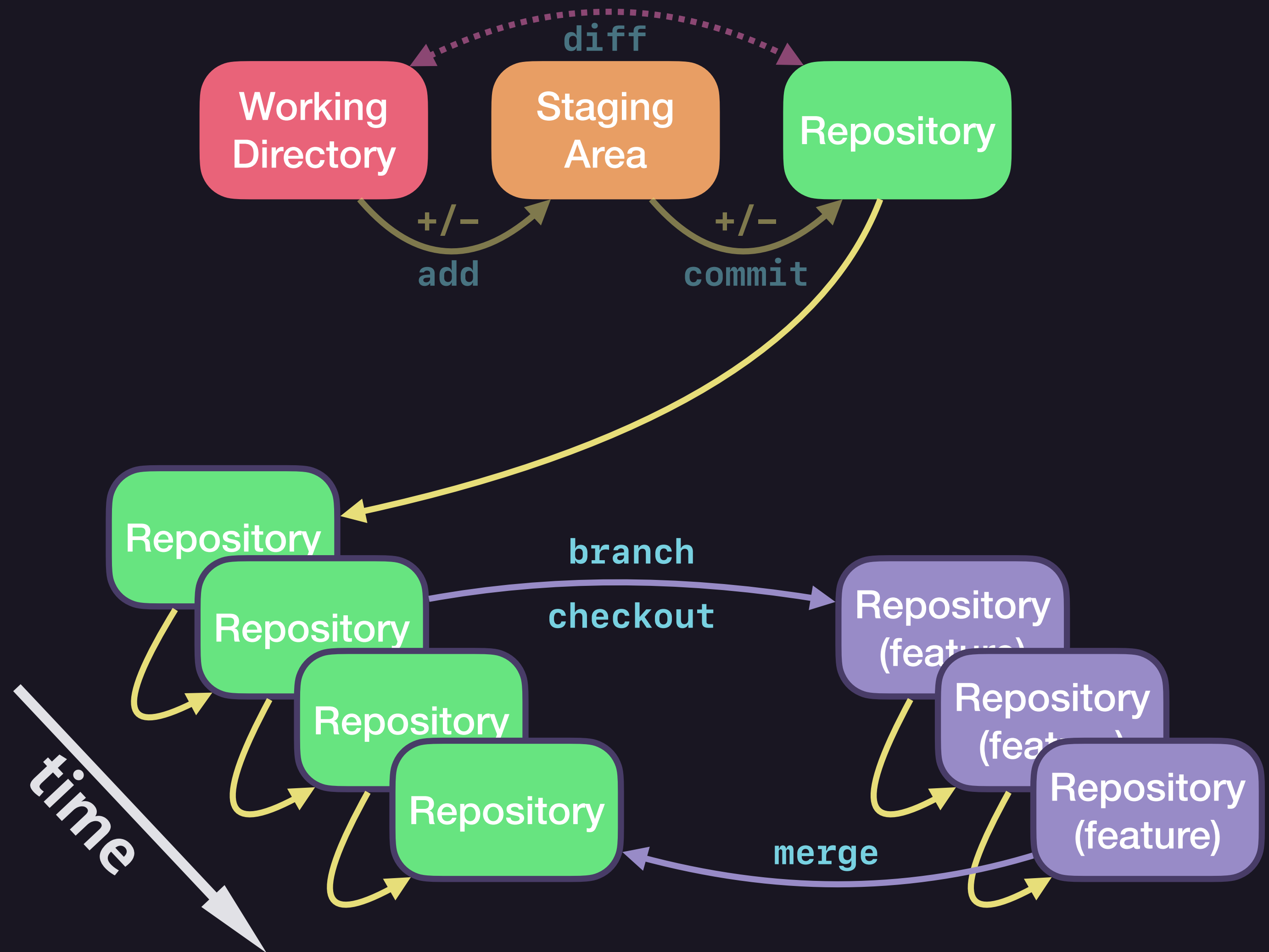
Git branches

git **log** - view history of **commits**

git **branch** - manage **branches**

git **checkout** - switch **branches**

git **merge** - merge changes
from one **branch** to another **branch**



Git

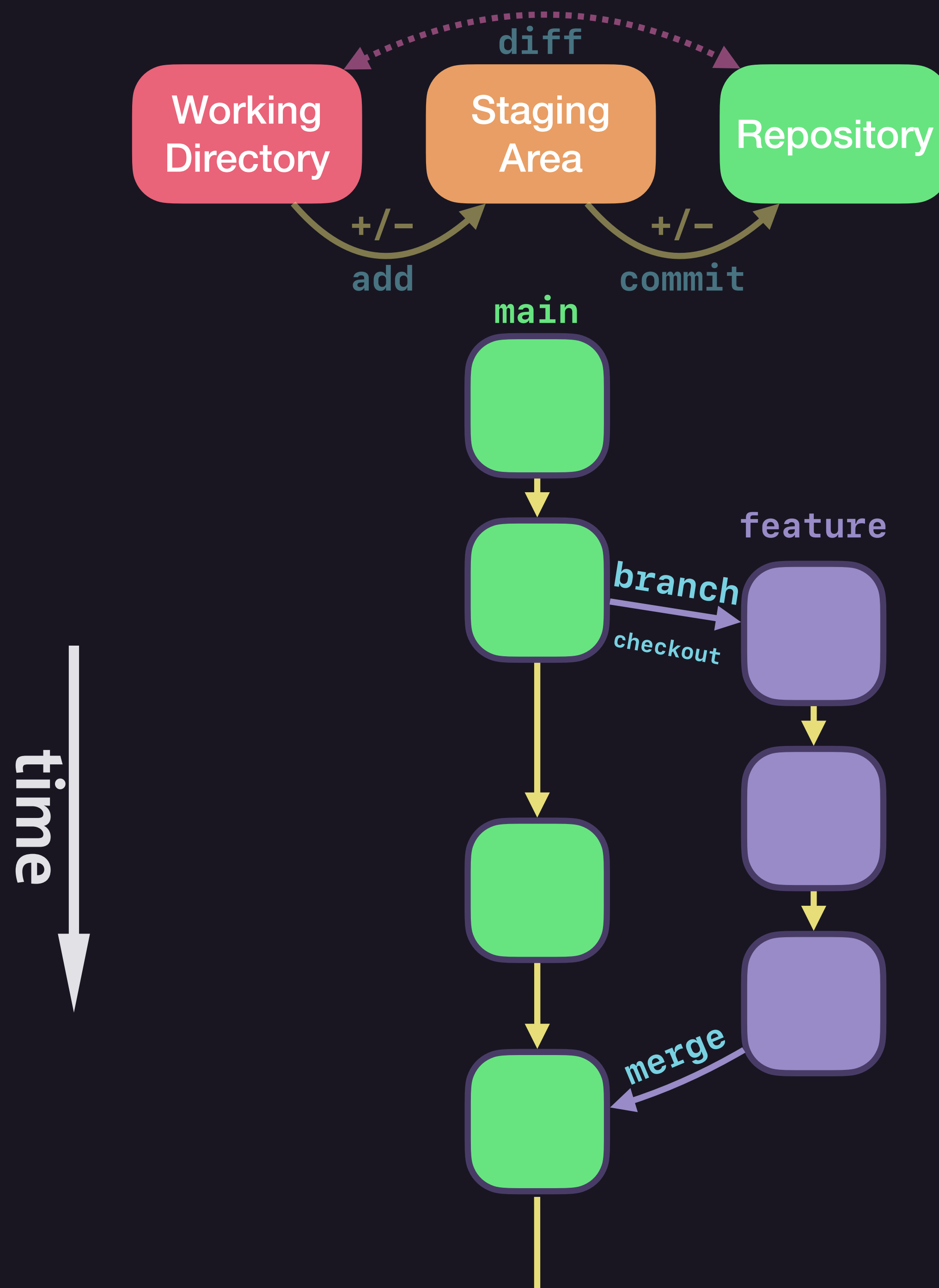
branches

`git log` - view history of **commits**

`git branch` - manage **branches**

`git checkout` - switch **branches**

`git merge` - merge changes
from one **branch** to another **branch**



Git

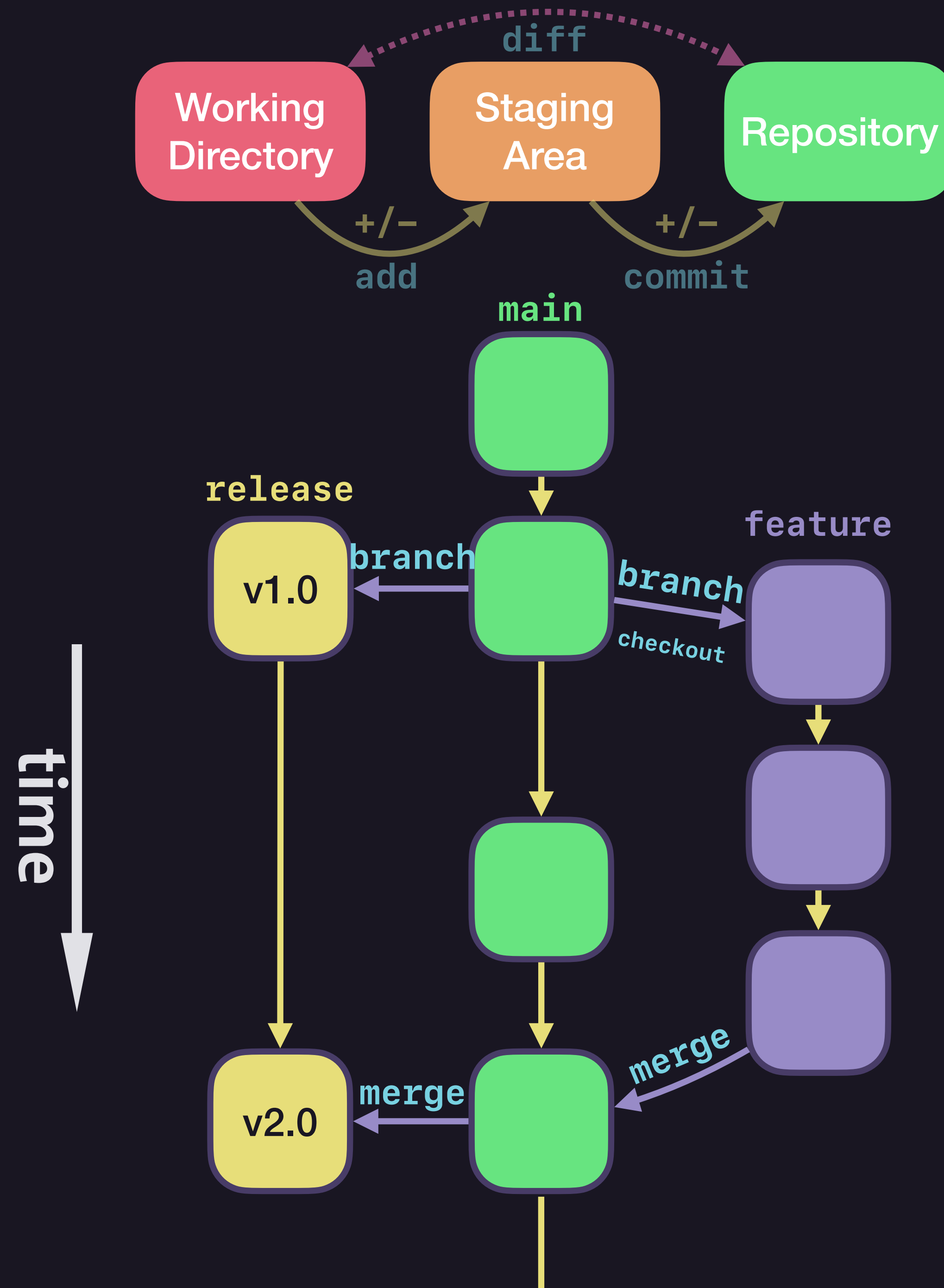
branches

`git log` - view history of **commits**

`git branch` - manage **branches**

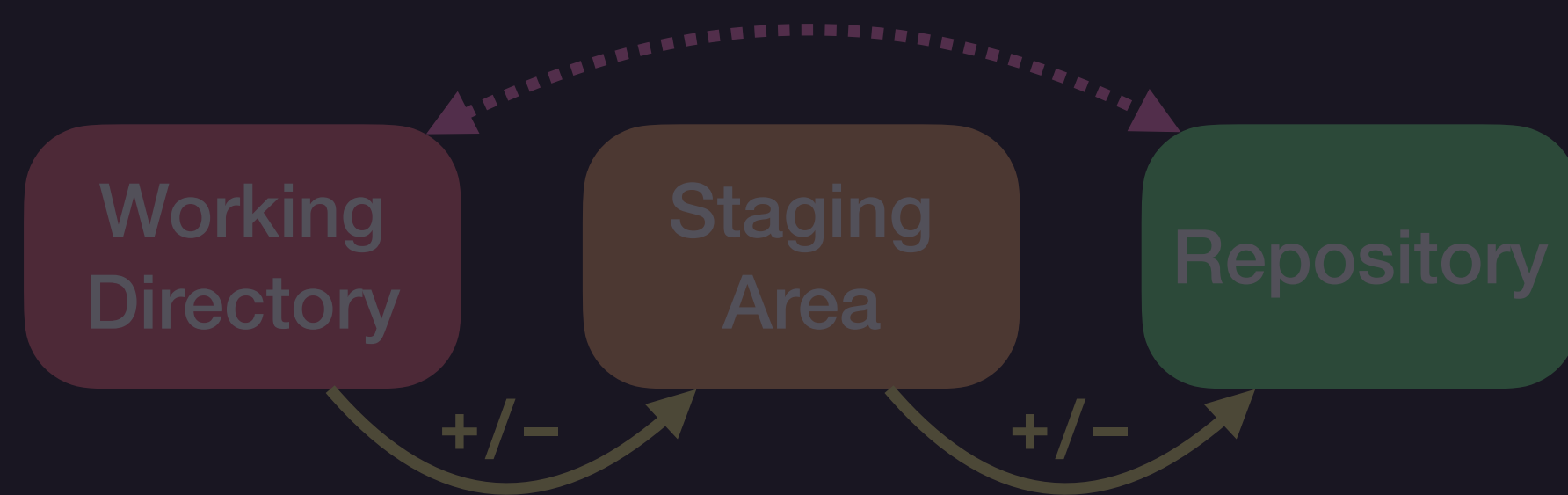
`git checkout` - switch **branches**

`git merge` - merge changes
from one **branch** to another **branch**

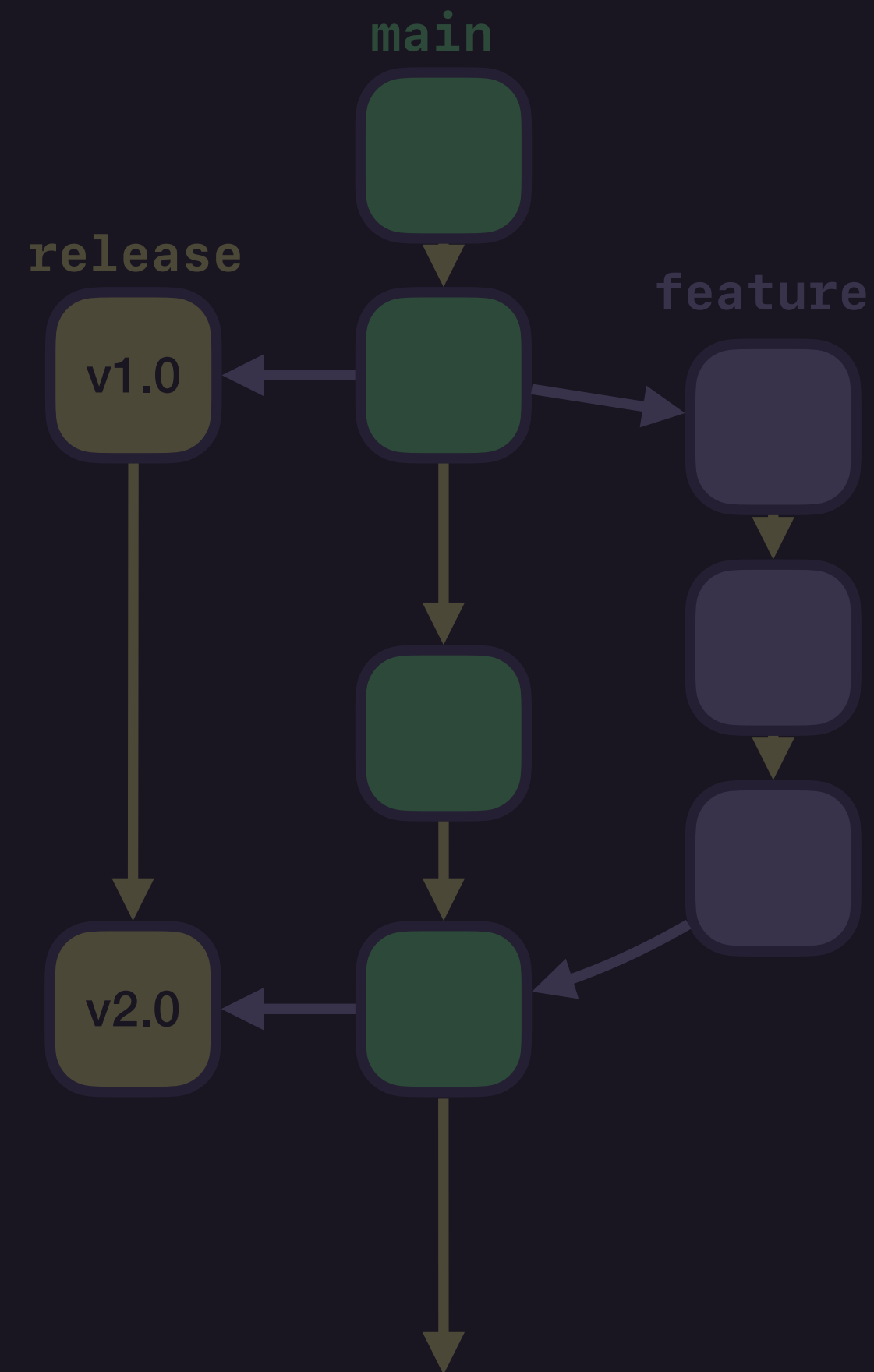


Overview

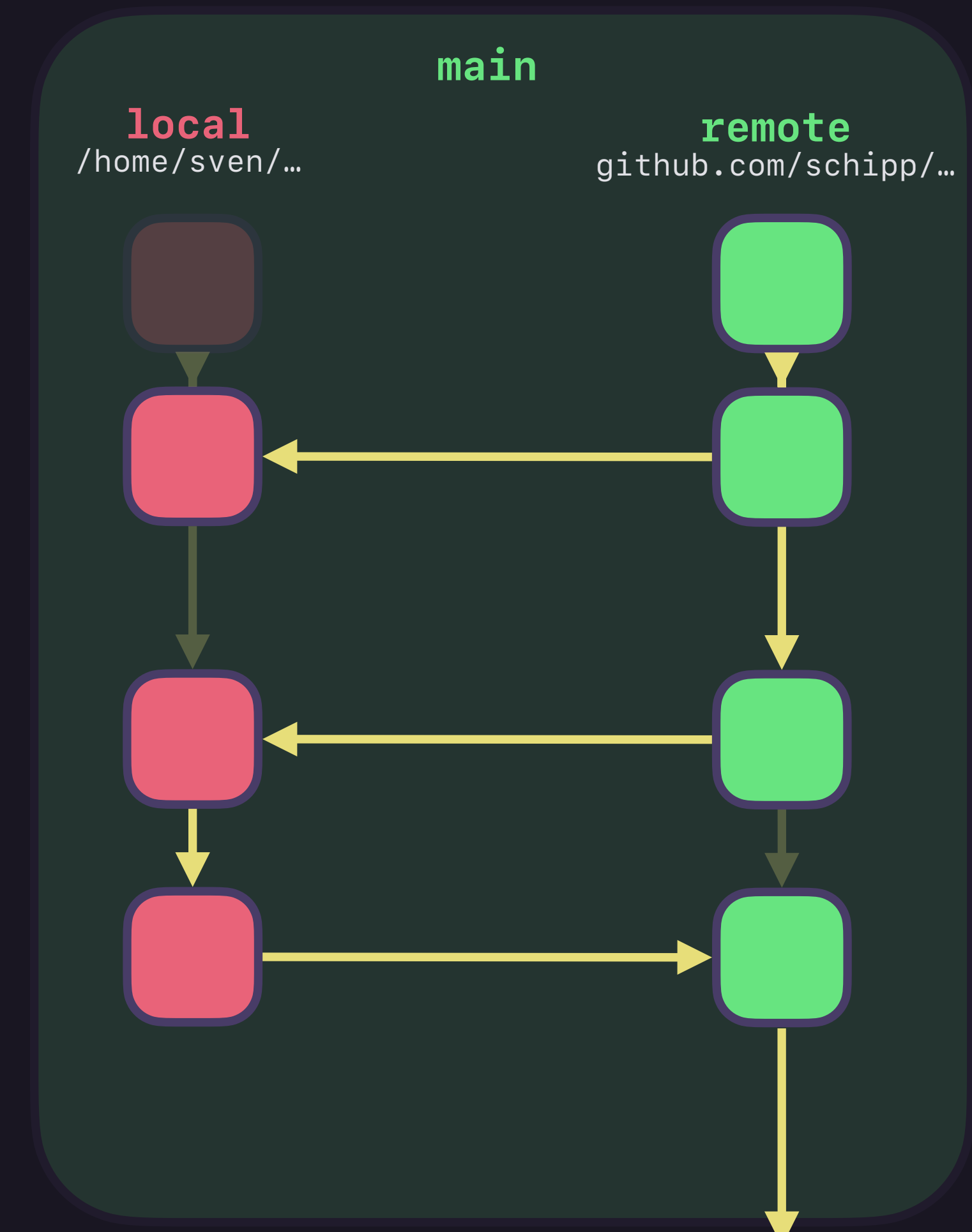
basics



branching

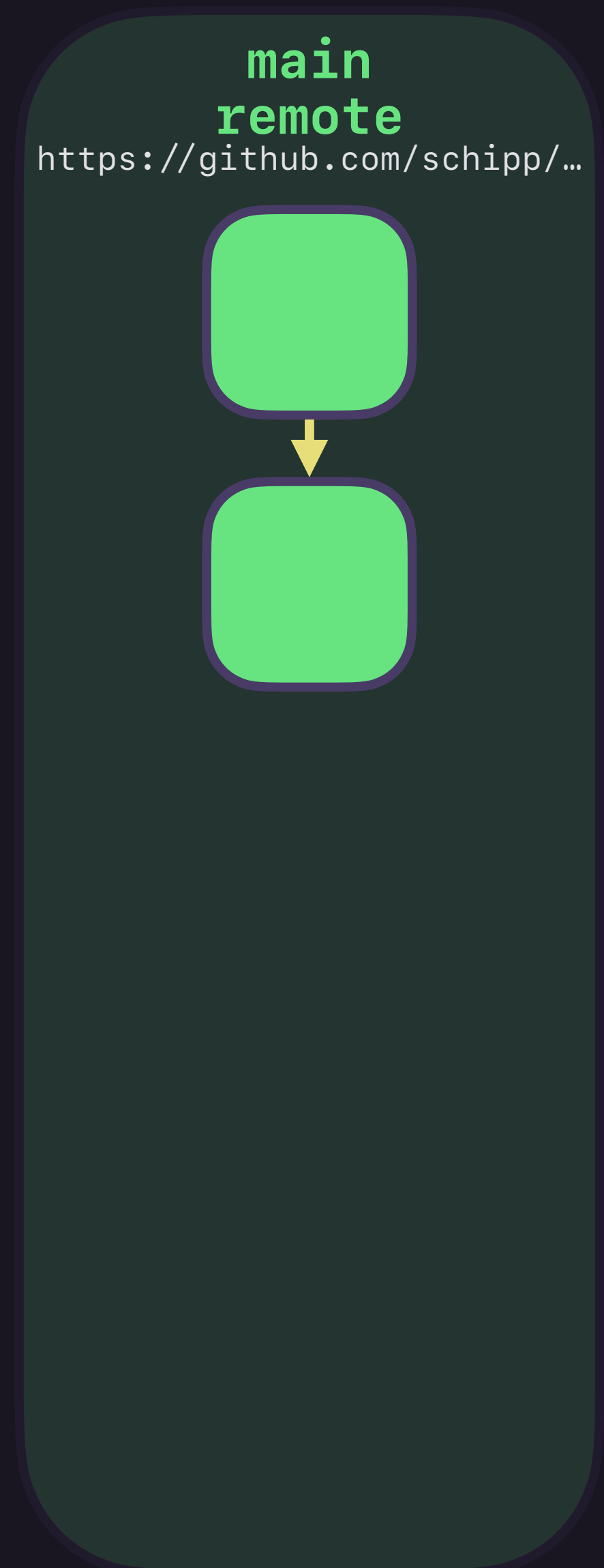


remote



Git

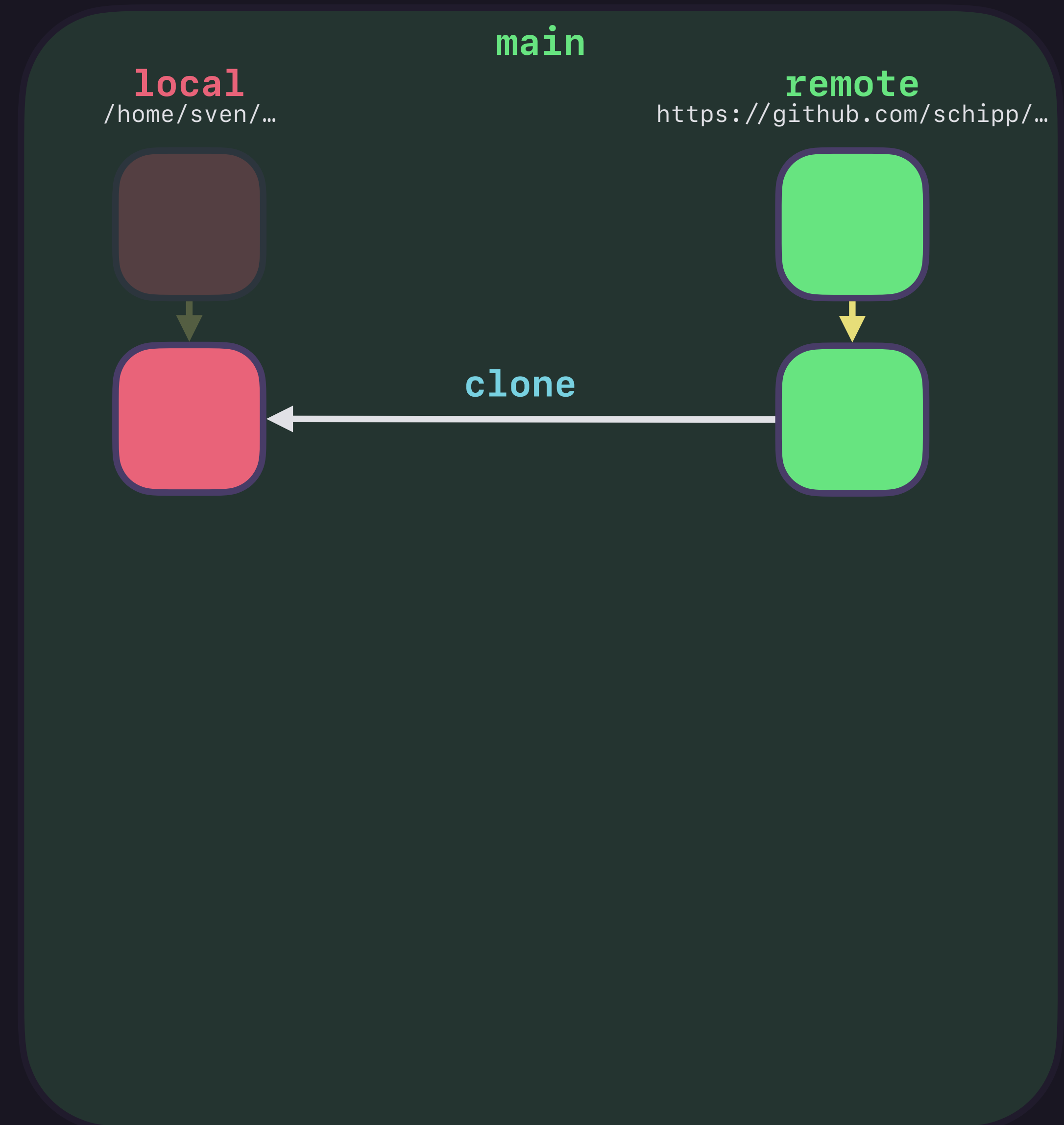
remote repositories



Git

remote repositories

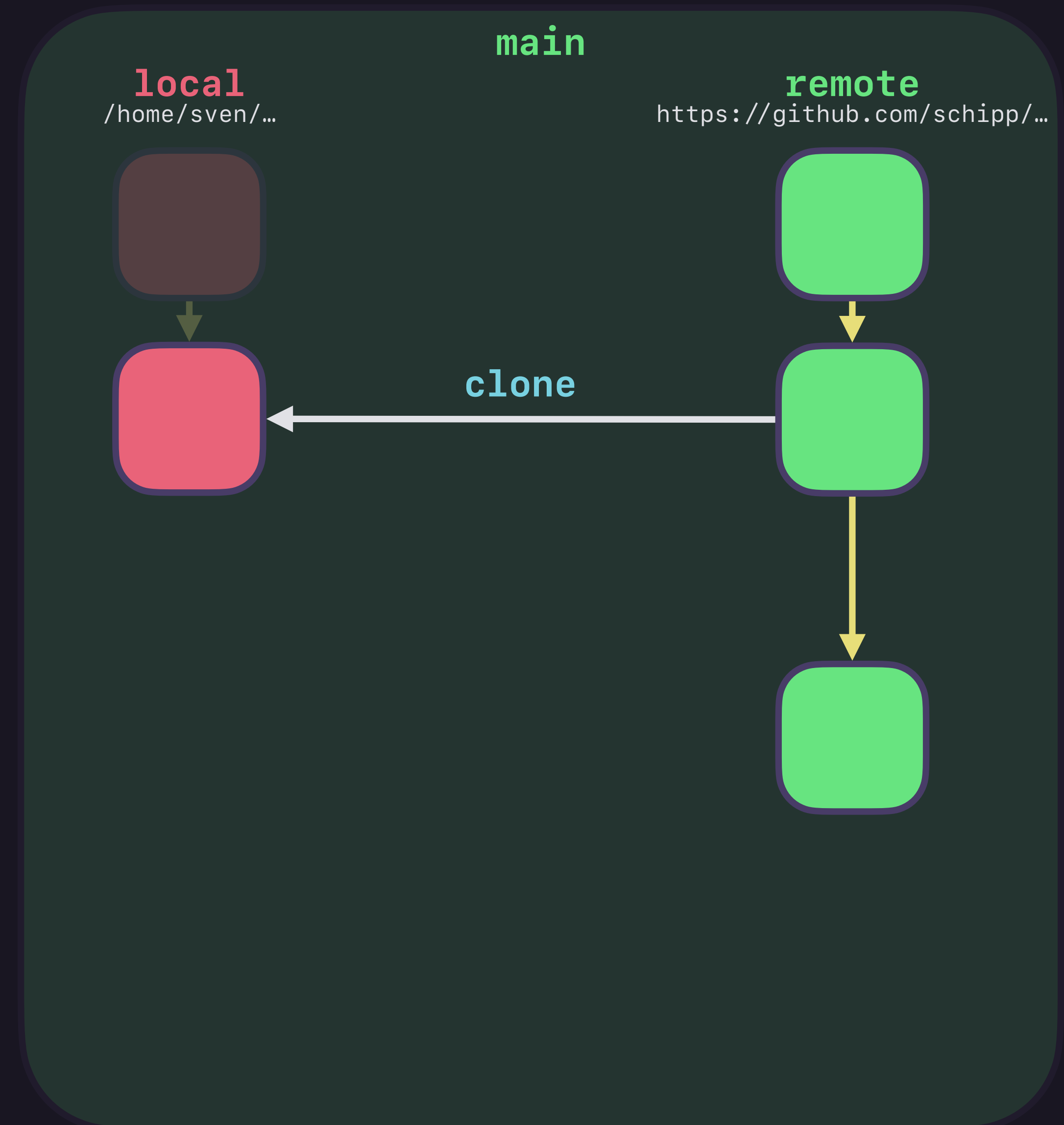
`git clone` - **local** copy of **remote** repository



Git

remote repositories

`git clone` - **local** copy of **remote** repository

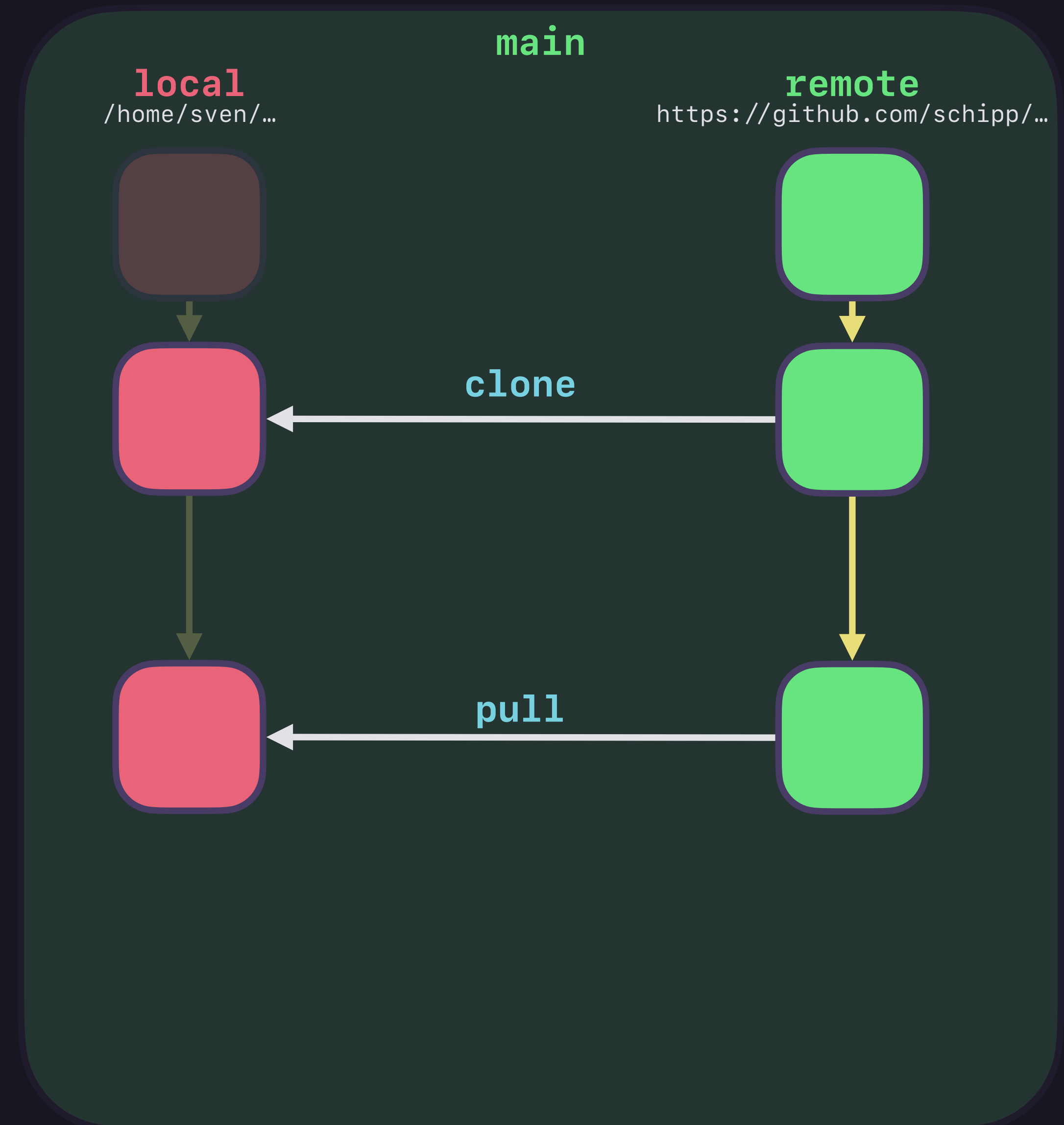


Git

remote repositories

`git clone` - **local** copy of **remote** repository

`git pull` - fetch & integrate **changes** from **remote**

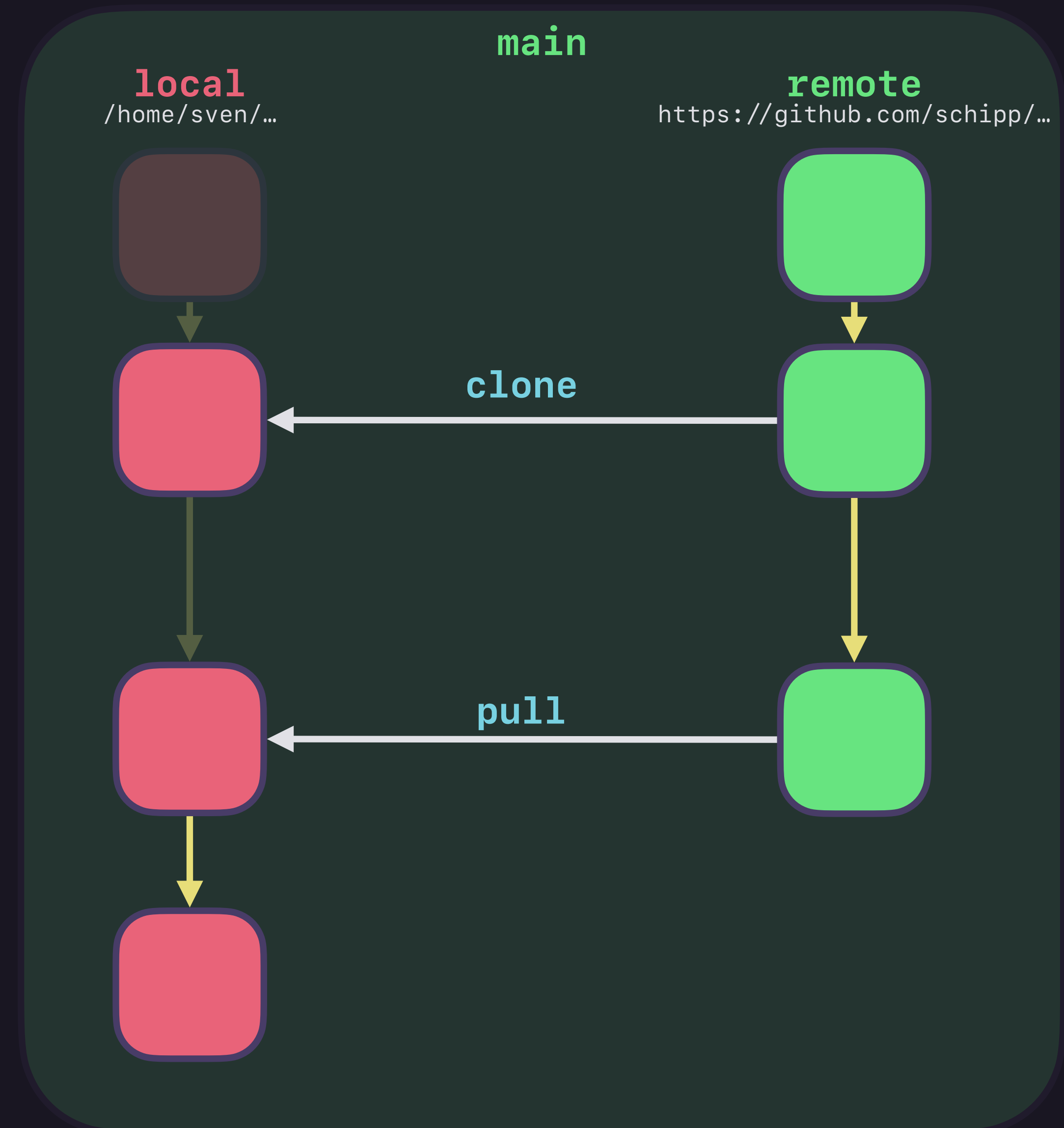


Git

remote repositories

`git clone` - **local** copy of **remote** repository

`git pull` - fetch & integrate **changes** from **remote**



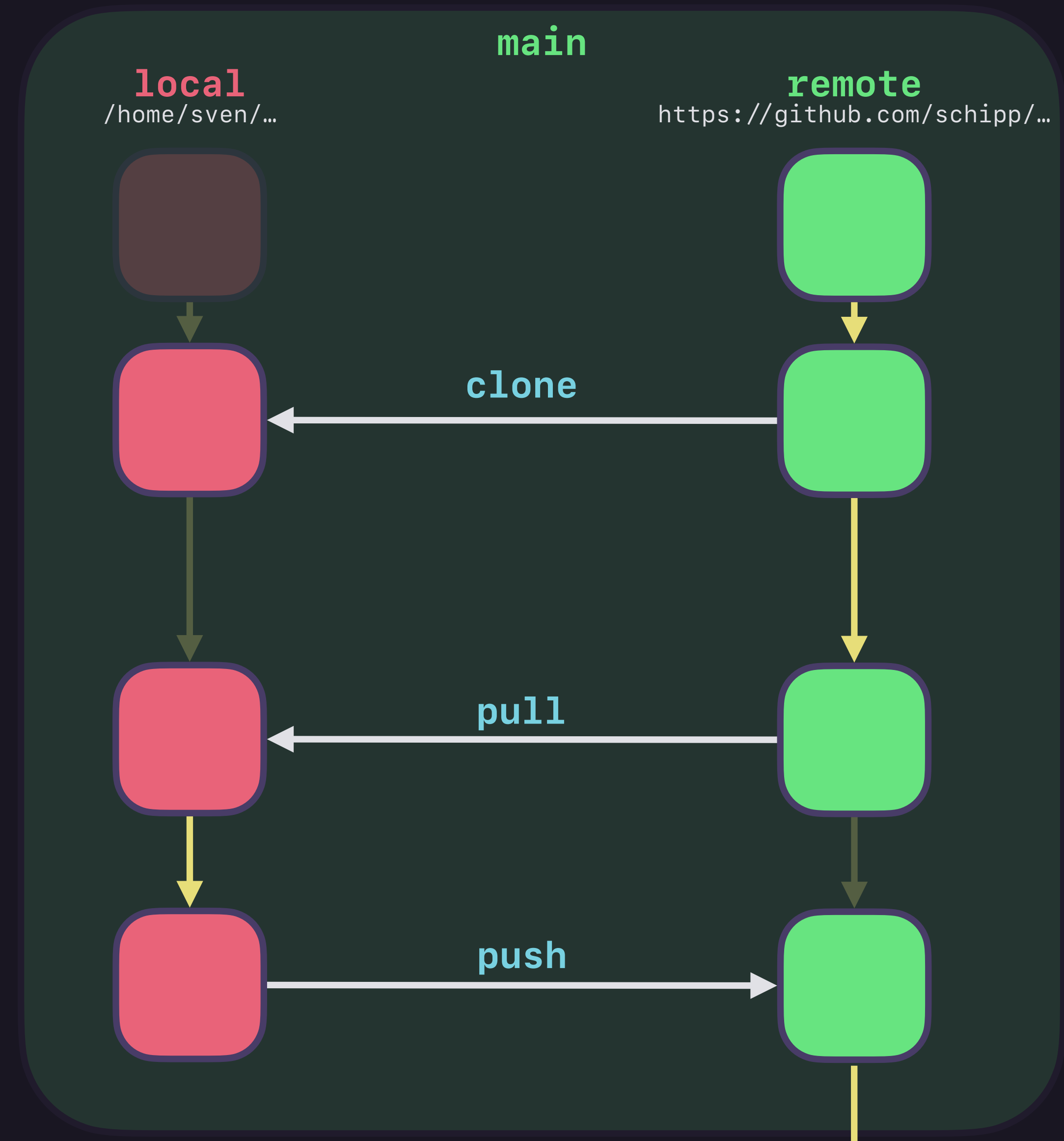
Git

remote repositories

`git clone` - **local** copy of **remote** repository

`git pull` - fetch & integrate **changes** from **remote**

`git push` - integrate **changes** from **local** at **remote**



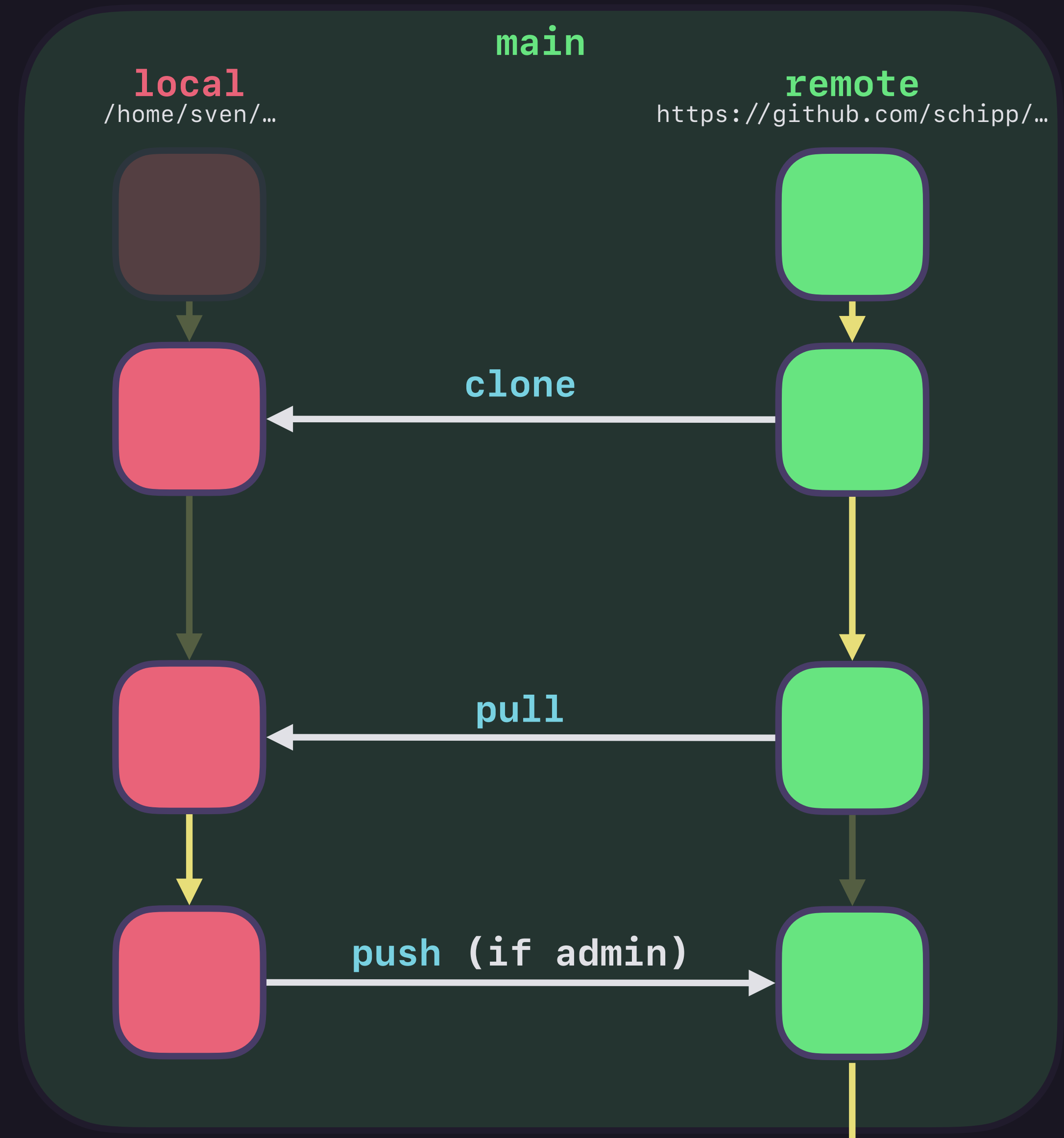
Git

remote repositories

`git clone` - **local** copy of **remote** repository

`git pull` - fetch & integrate **changes** from **remote**

`git push` - integrate **changes** from **local** at **remote**



Git

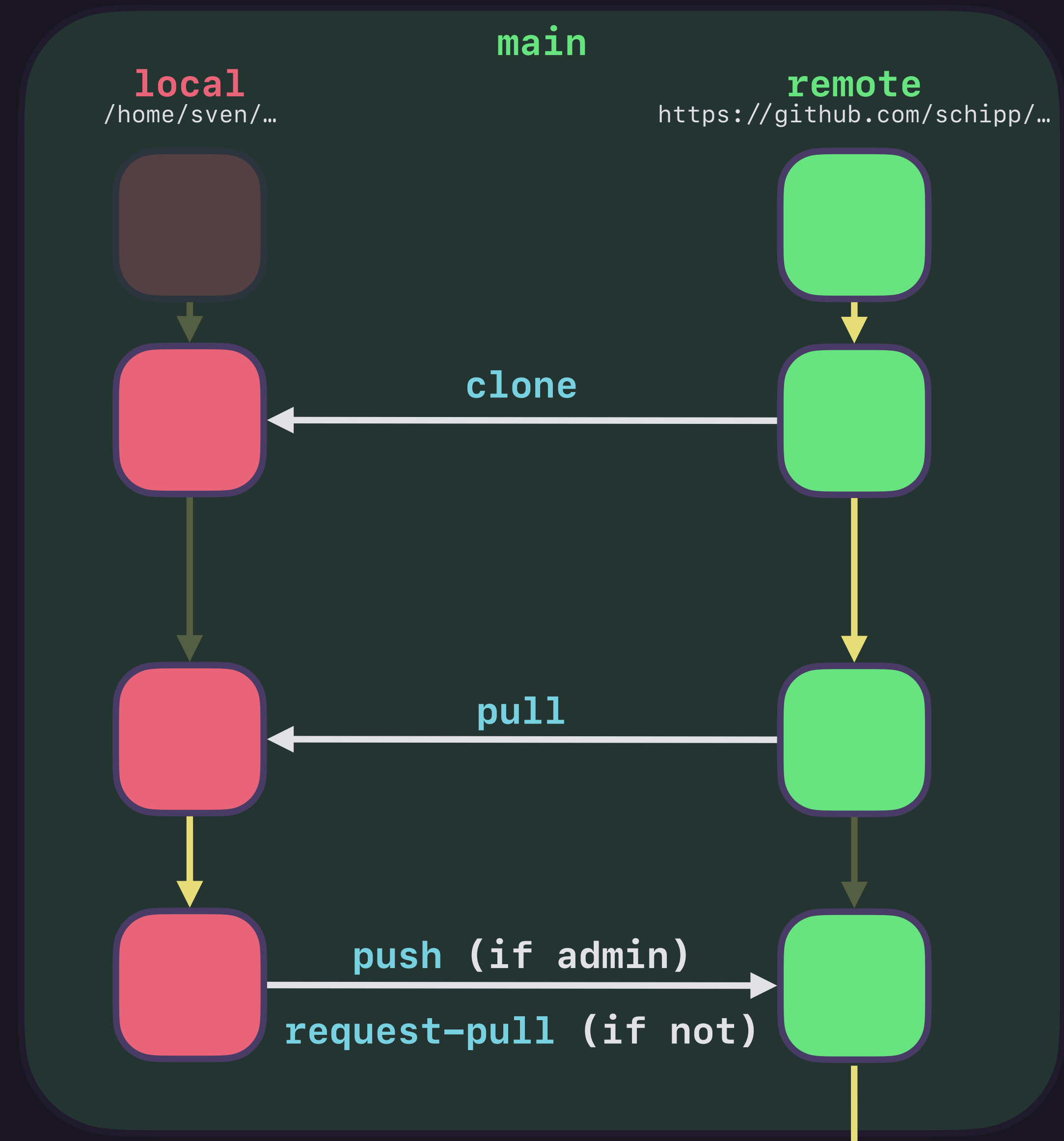
remote repositories

`git clone` - **local** copy of **remote** repository

`git pull` - fetch & integrate **changes** from **remote**

`git push` - integrate **changes** from **local** at **remote**

`git request-pull` - ask **remote** repository to pull **changes** from **local**



Git

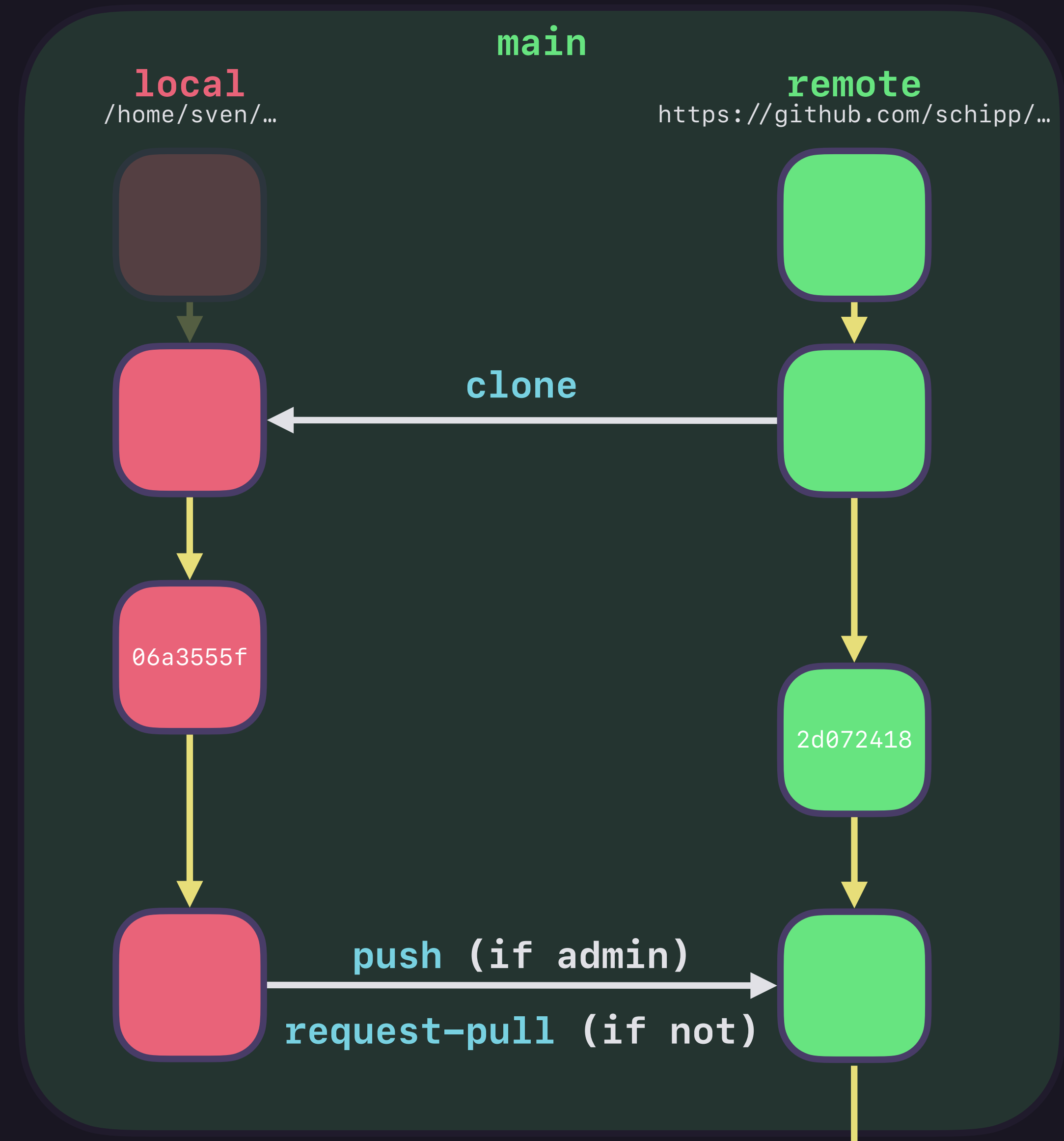
remote repositories

`git clone` - **local** copy of **remote** repository

`git pull` - fetch & integrate **changes** from **remote**

`git push` - integrate **changes** from **local** at **remote**

`git request-pull` - ask **remote** repository to pull **changes** from **local**



Git

remote repositories

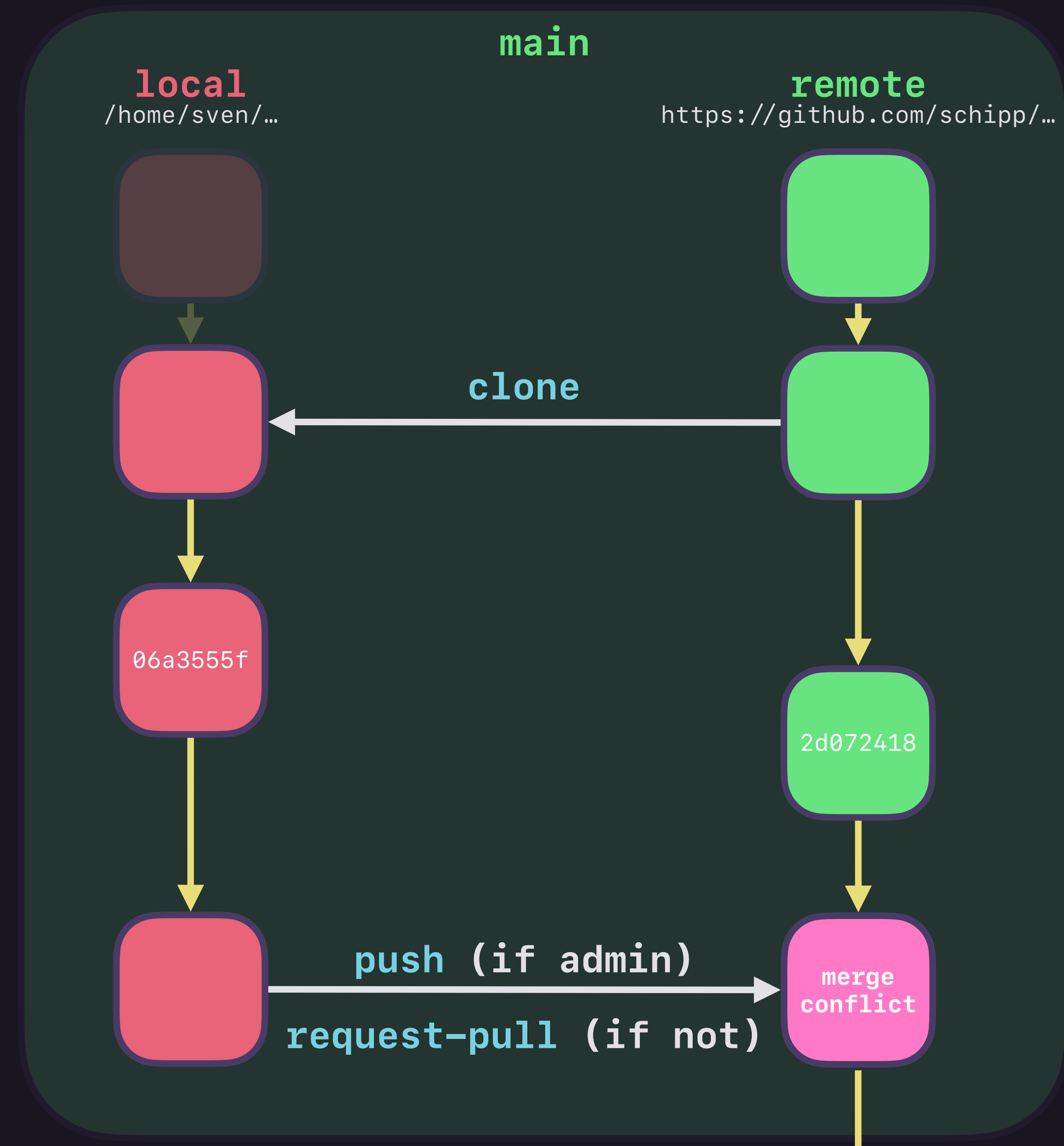
`git clone` - **local** copy of **remote** repository

`git pull` - fetch & integrate **changes** from **remote**

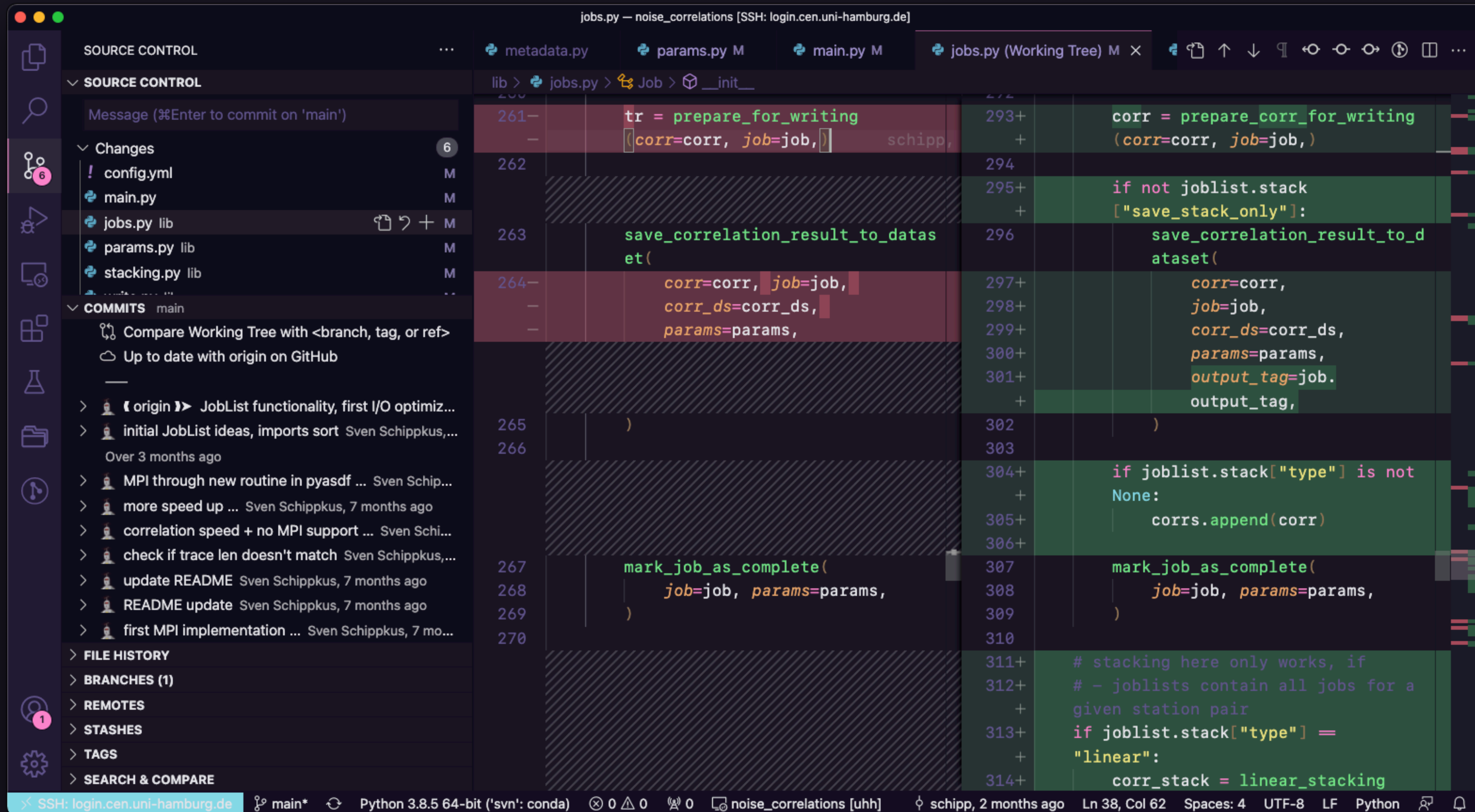
`git push` - integrate **changes** from **local** at **remote**

`git request-pull` - ask **remote** repository to pull **changes** from **local**

A **merge conflict** can occur when changes have been made to the same file in different **branches** or repository locations (**local**, **remote**).



Git in practise (example vscode)



jobs.py — noise_correlations [SSH: login.cen.uni-hamburg.de]

lib > jobs.py > Job > __init__

261- tr = prepare_for_writing
(corr=corr, job=job,) schipp,

262

263 save_correlation_result_to_data
et(

264- corr=corr, job=job,
corr_ds=corr_ds,
params=params,

265)

266

267 mark_job_as_complete(
job=job, params=params,
268)
269
270

293+ corr = prepare_corr_for_writing
(corr=corr, job=job,)

294

295+ if not joblist.stack
["save_stack_only"]:
296 save_correlation_result_to_d
ataset(
297+ corr=corr,
298+ job=job,
299+ corr_ds=corr_ds,
300+ params=params,
301+ output_tag=job.
output_tag,
302)
303

304+ if joblist.stack["type"] is not
None:
305+ corrs.append(corr)
306+

307 mark_job_as_complete(
308 job=job, params=params,
309)
310

311+ # stacking here only works, if
312+ # - joblists contain all jobs for a
given station pair
313+ if joblist.stack["type"] =
"linear":
314+ corr_stack = linear_stacking

SSH: login.cen.uni-hamburg.de main* Python 3.8.5 64-bit ('svn': conda) 0 0 0 noise_correlations [uhh] schipp, 2 months ago Ln 38, Col 62 Spaces: 4 UTF-8 LF Python

github.com

Search or jump to...

Pull requests

Issues

Marketplace

Explore

+

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

schipp

/

spin_demonstration

Great repository names are short and memorable. Need inspiration? How about [automatic-memory?](#)

Description (optional)

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.](#)

Creating repository...

© 2022 GitHub, Inc.

Terms

Privacy

Security

Status

Docs

Contact GitHub

Pricing

API

Training

Blog

About

github.com

Search or jump to...

Pull requestsIssuesMarketplaceExplore

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

schipp

/

spin_demonstration

✓

Great repository names are short and memorable. Need inspiration? How about [automatic-memory?](#)

Description (optional)

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.](#)

Creating repository...

© 2022 GitHub, Inc.

TermsPrivacySecurityStatusDocsContact GitHubPricingAPITrainingBlogAbout

github.com

Search or jump to...

Pull requestsIssuesMarketplaceExplore

schipp / spin_demonstration

Private

Unwatch 1

Fork 0

Star 0

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Quick setup — if you've done this kind of thing before

Set up in Desktop

or

HTTPS

SSH

https://github.com/schipp/spin_demonstration.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# spin_demonstration" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/schipp/spin_demonstration.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/schipp/spin_demonstration.git
git branch -M main
git push -u origin main
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

ProTip!

Use the URL for this page when adding GitHub as a remote.

Github, GitLab, ...

GitLab

Menu

Search GitLab

Create blank project

Create a blank project to house your files, plan your work, and collaborate on code, among other things.

New project

Create blank project

Project name

spin_demonstration

Project URL

https://gitlab.com/schipp/

Project slug

spin_demonstration

Want to house several dependent projects under the same namespace?

Create a group.

Project description (optional)

Description format

Visibility Level

Private

Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.

Public

The project can be accessed without any authentication.

Project Configuration

Initialize repository with a README

Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Create project

Cancel

GitLab

Menu

Search GitLab

The repository for this project is empty

You can get started by cloning the repository or start adding files to it with one of the following options.

Clone

Upload File

New file

Add README

Add LICENSE

Add CHANGELOG

Add CONTRIBUTING

Set up CI/CD

Configure Integrations

Command line instructions

You can also upload existing files from your computer using the instructions below.

Git global setup

```
git config --global user.name "Sven"
git config --global user.email "sven.schippkus@gmail.com"
```

Create a new repository

```
git clone https://gitlab.com/schipp/spin_demonstration.git
cd spin_demonstration
git switch -c main
touch README.md
git add README.md
git commit -m "add README"
git push -u origin main
```

Push an existing folder

```
cd existing_folder
git init --initial-branch=main
git remote add origin https://gitlab.com/schipp/spin_demonstration.git
git add .
git commit -m "Initial commit"
git push -u origin main
```

Push an existing Git repository

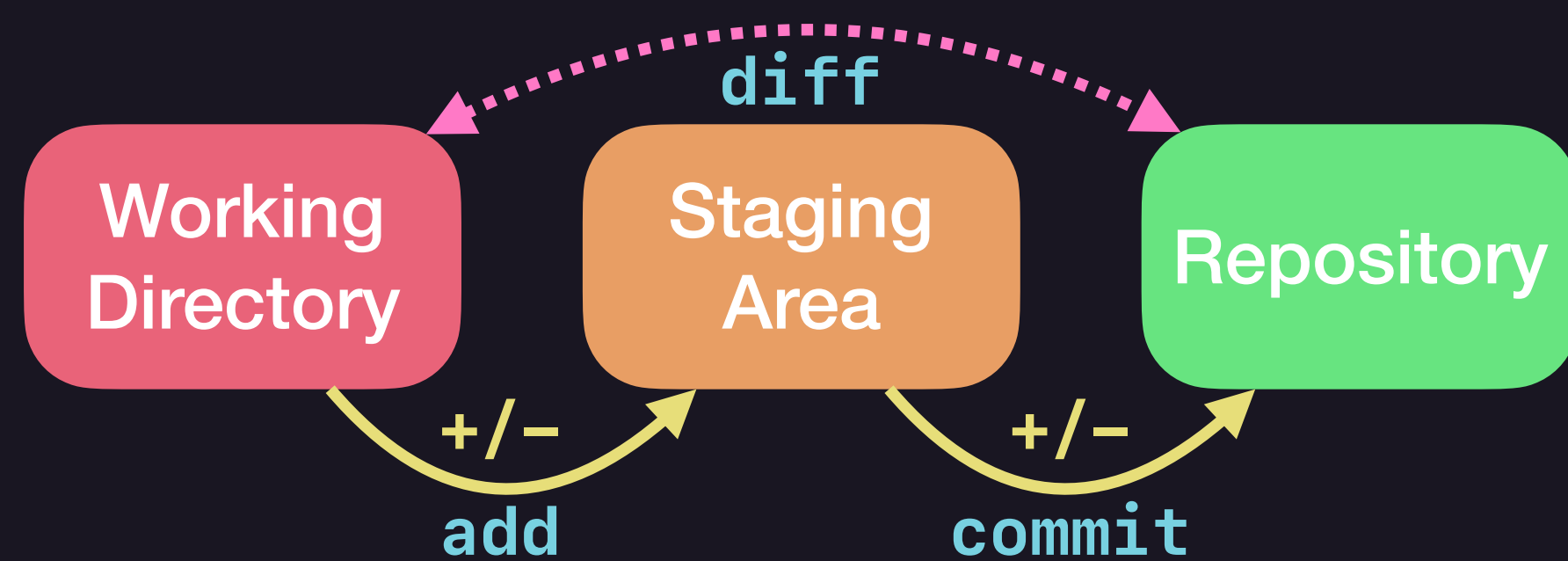
```
cd existing_repo
git remote rename origin old-origin
git remote add origin https://gitlab.com/schipp/spin_demonstration.git
git push -u origin --all
git push -u origin --tags
```


Summary

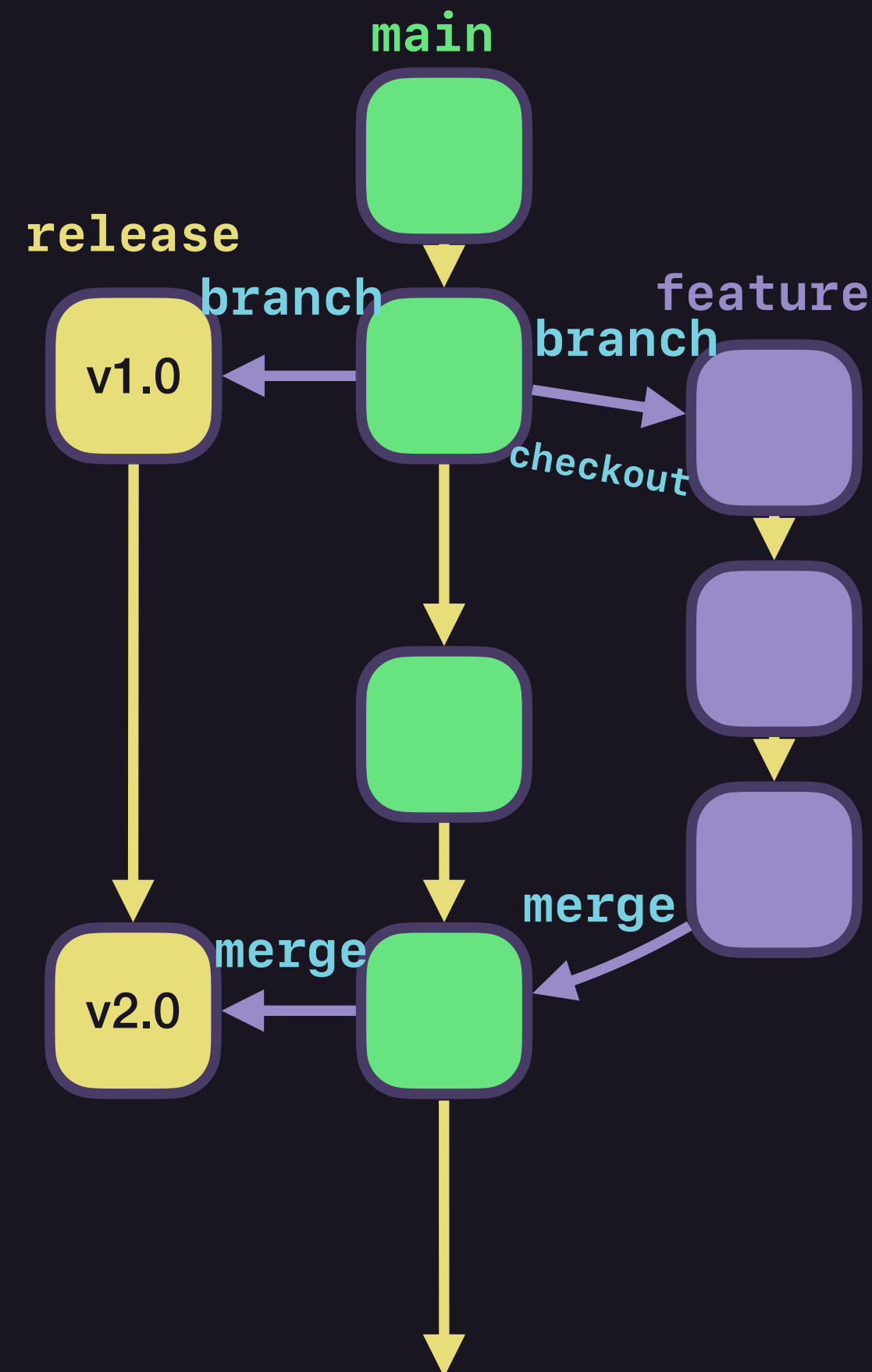
slides on our github.com/spin-itn/git_intro

video on the SPIN-ITN youtube channel: youtu.be/watch?v=HIWJZKoafWw

basics



branching



remote

