

Ateliers R³



Session 6 - Git, pour soi et avec les autres

Quoi faire => Comment faire

...pour se simplifier la vie

....pour permettre aux autres de reproduire ses analyses

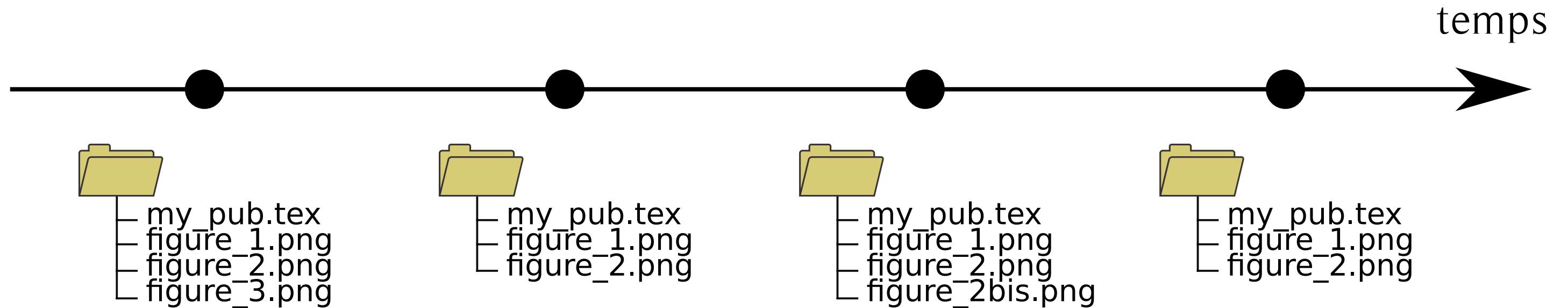
...pour être sûr(e) que ses analyses sont correctes

etc.



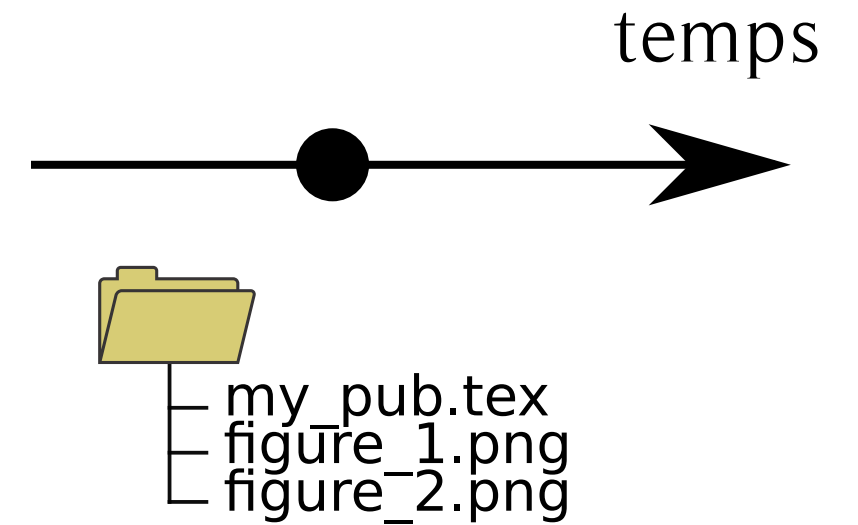
+ bonnes pratiques

Qu'est-ce que git ? Un système de *versionnage* des fichiers

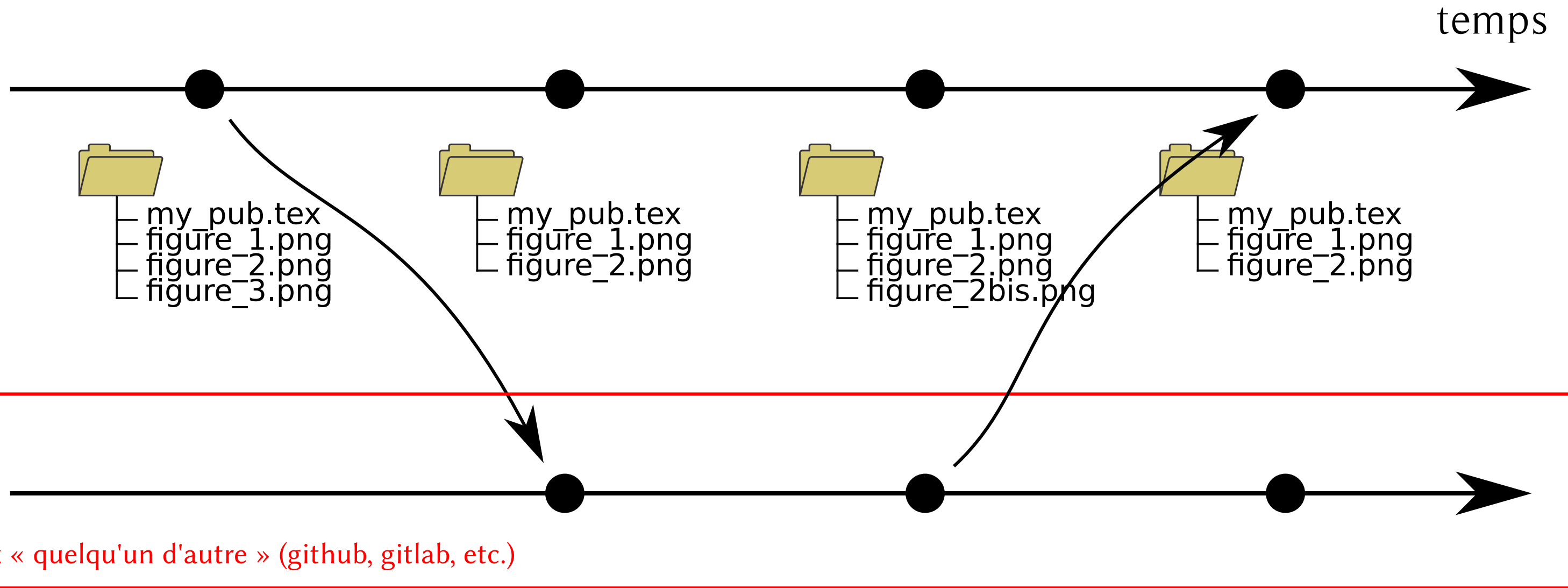


Qu'est-ce que git ? Un système de *versionnage* des fichiers

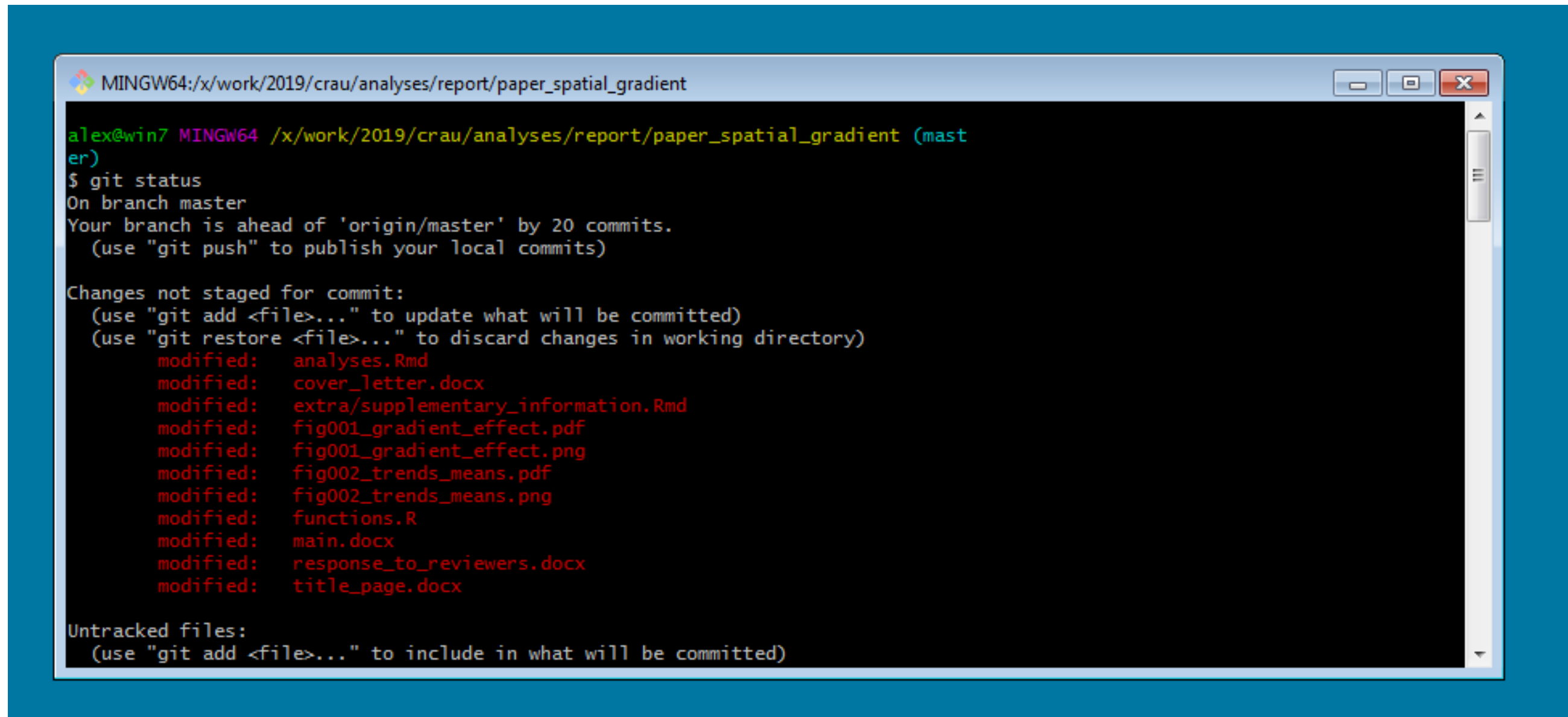
(sans versionnage = pas d'historique)



Qu'est-ce que git ? Un système de *collaboration*



Qu'est-ce que git ? Un outil en ligne de commande (windows compris)

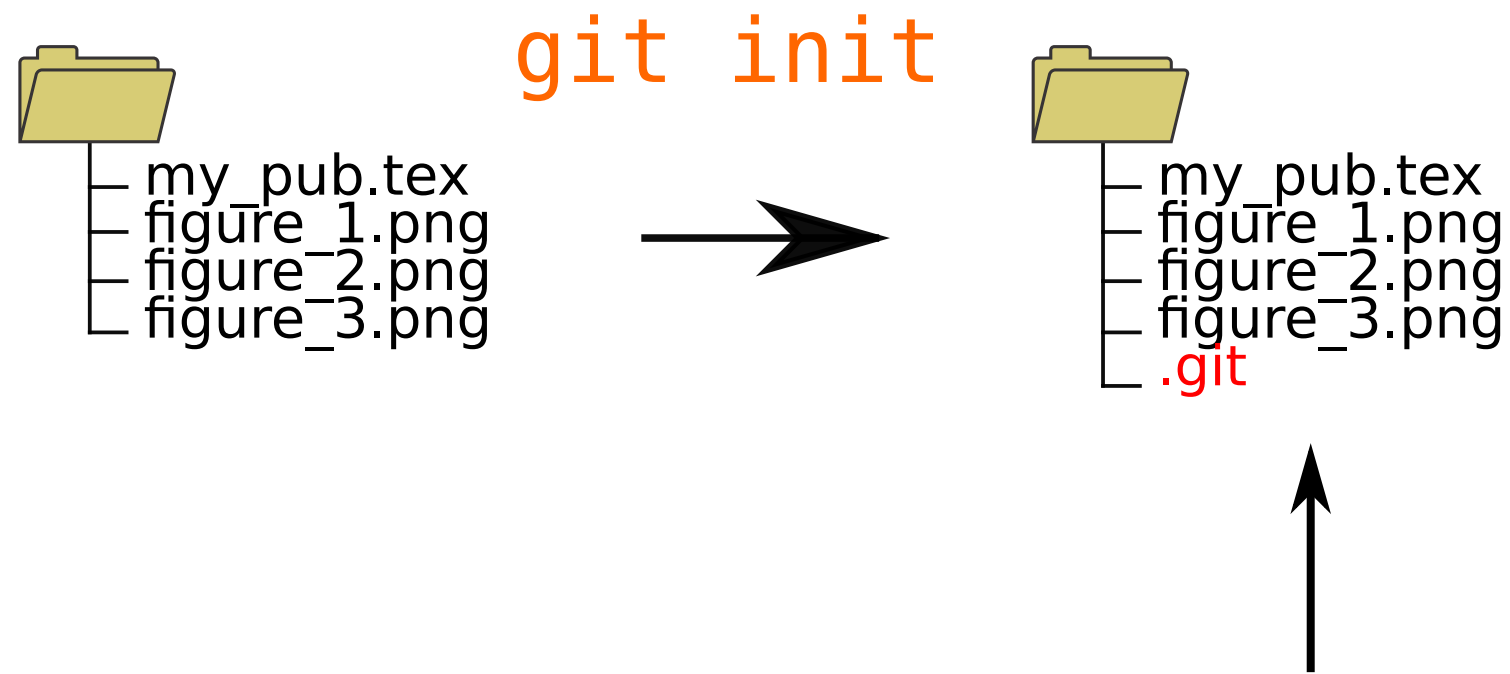


```
MINGW64:/x/work/2019/crau/analyses/report/paper_spatial_gradient
alex@win7 MINGW64 /x/work/2019/crau/analyses/report/paper_spatial_gradient (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 20 commits.
(use "git push" to publish your local commits)

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:   analyses.Rmd
    modified:   cover_letter.docx
    modified:   extra/supplementary_information.Rmd
    modified:   fig001_gradient_effect.pdf
    modified:   fig001_gradient_effect.png
    modified:   fig002_trends_means.pdf
    modified:   fig002_trends_means.png
    modified:   functions.R
    modified:   main.docx
    modified:   response_to_reviewers.docx
    modified:   title_page.docx

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

Les bases essentielles



un dépôt (*repository*) = un dossier versionné

Votre premier commit



Votre premier commit



`git add my_pub.tex figure_1.png figure_2.png figure_3.png` (add specific files)

Votre premier commit

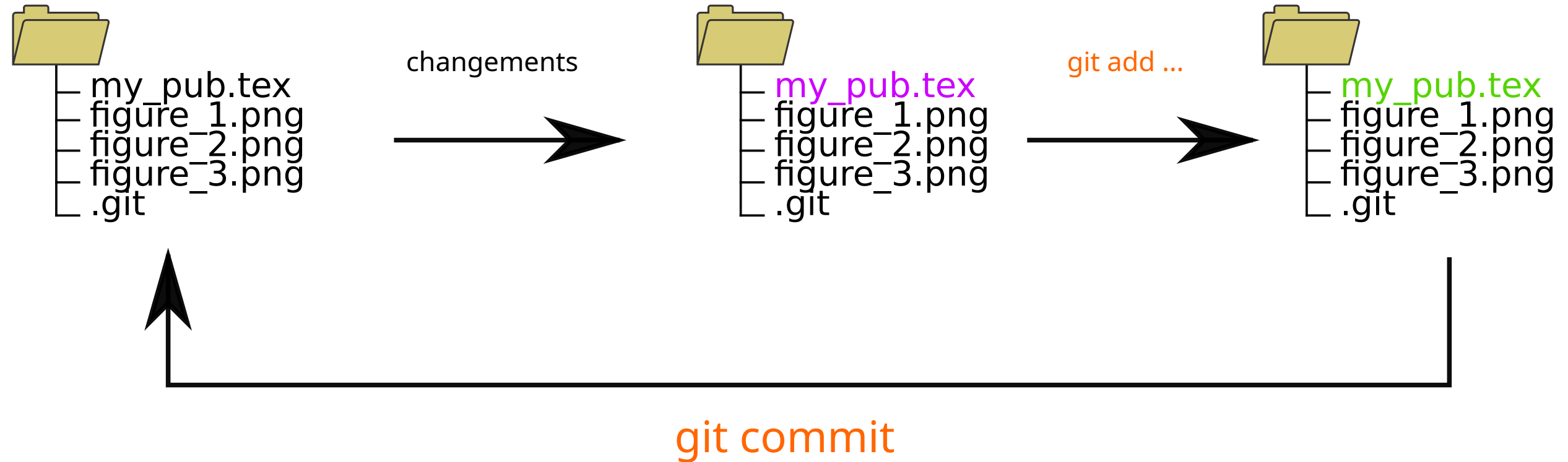


`git add my_pub.tex figure_1.png figure_2.png figure_3.png` (add specific files)

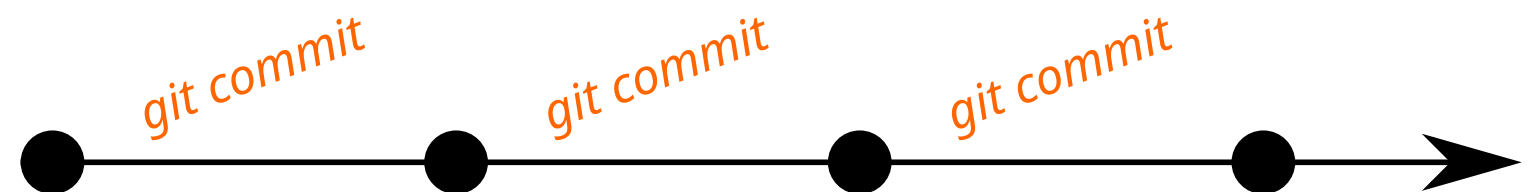
(then `git commit` !)

Un commit = un ensemble de changements faits sur les fichiers du dépôt

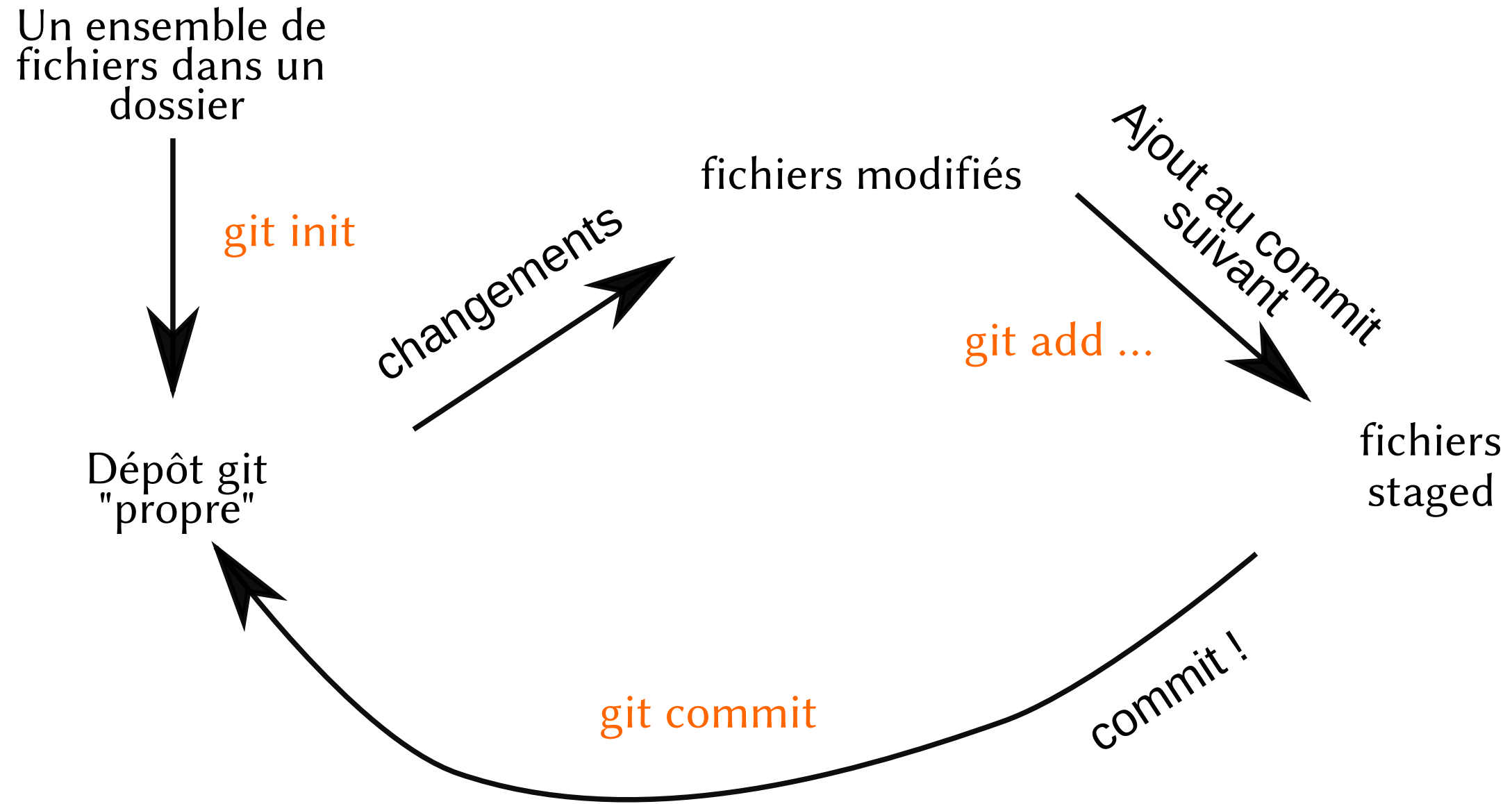
Votre deuxième commit



On enregistre tout l'historique de nos changements !

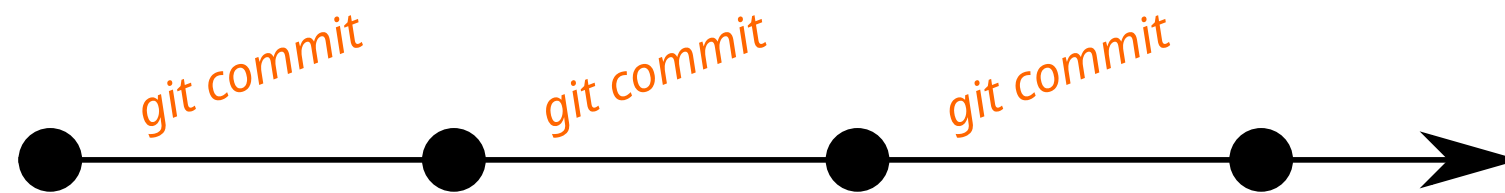


Le reste des commits de votre vie



A tout moment: **git status**

En pratique...

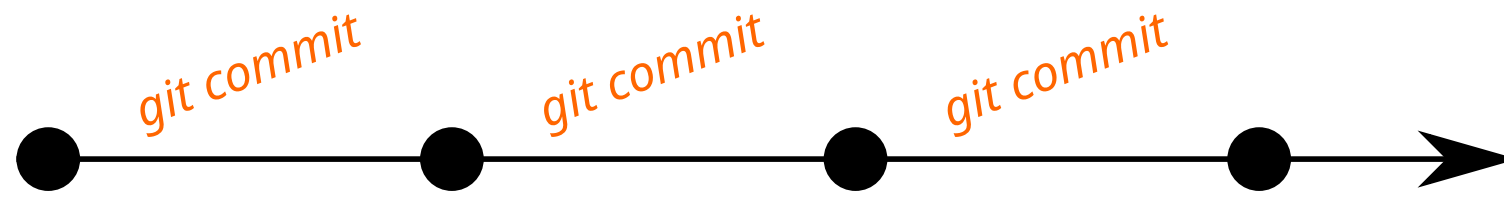


On a tout l'historique des changements

*... en cas de problème
... pour s'y retrouver*

Git avec les autres

Dans votre dépôt



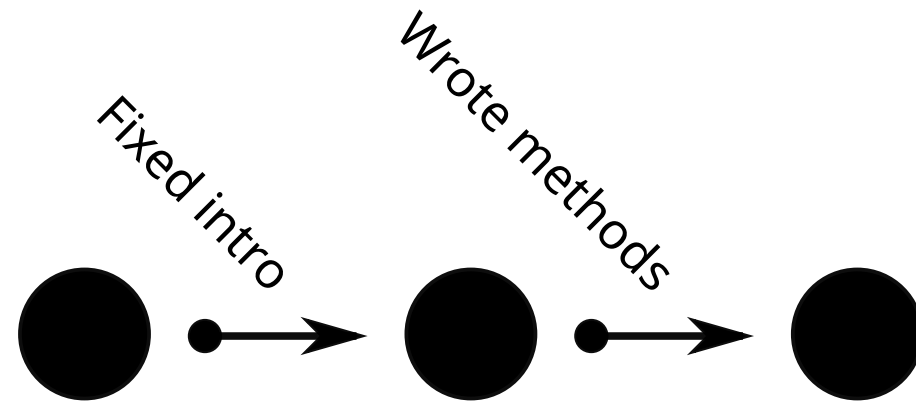
git pull

git push

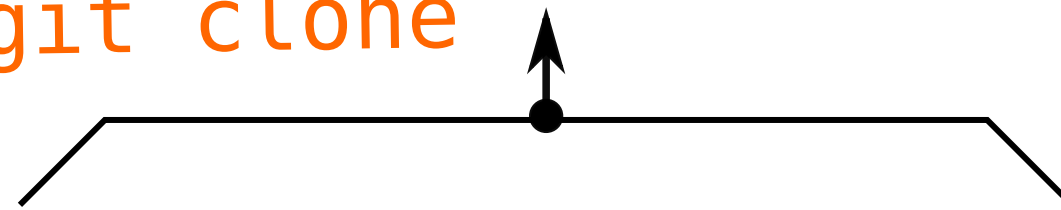
Un autre dépôt



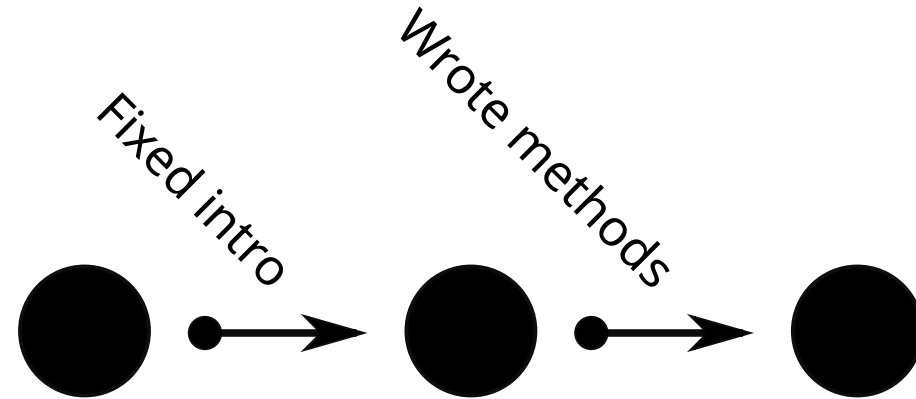
Dans votre dépôt



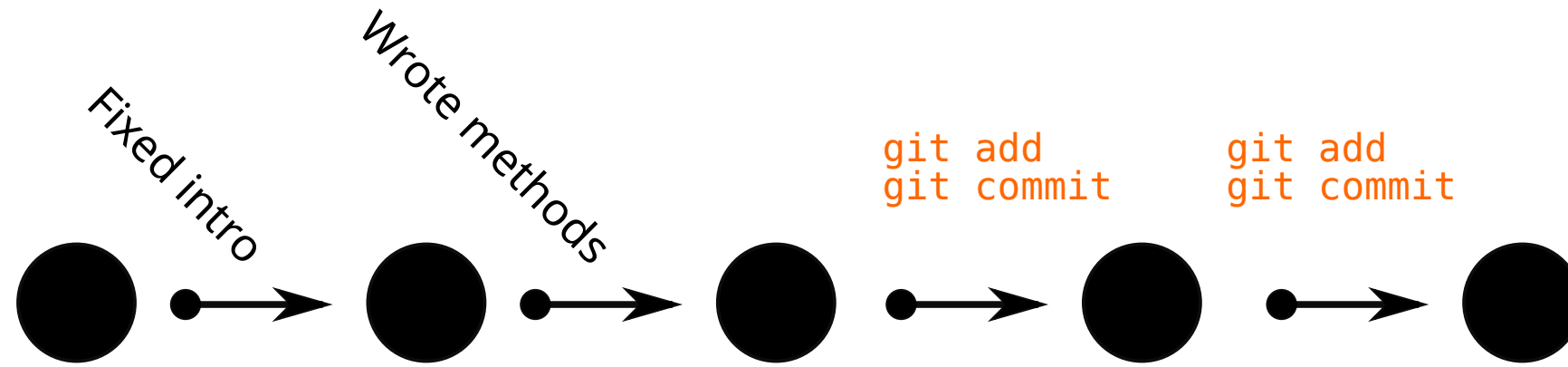
git clone



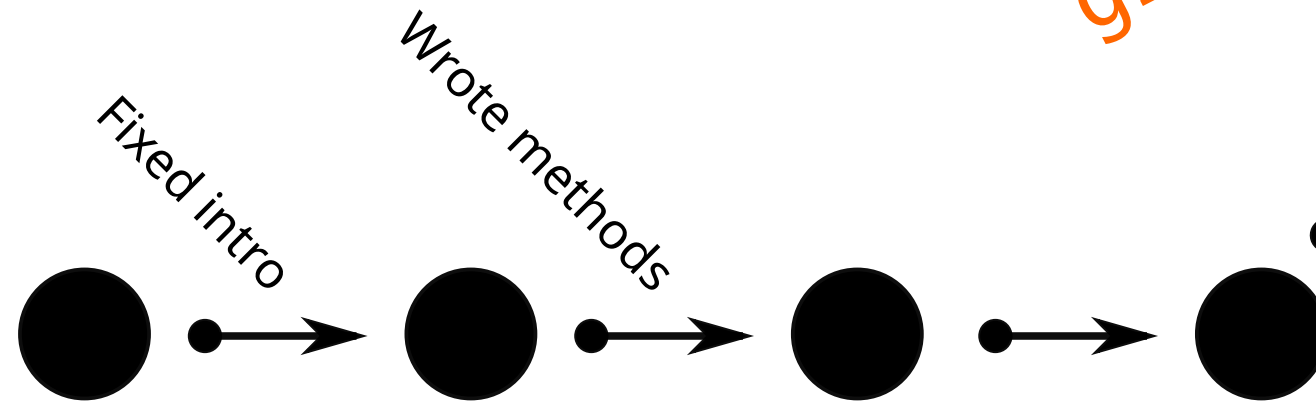
Un autre dépôt



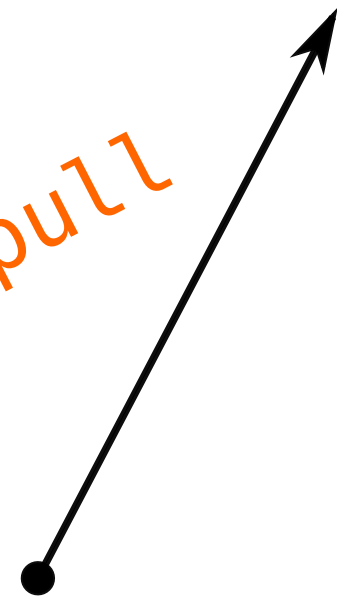
Dans votre dépôt



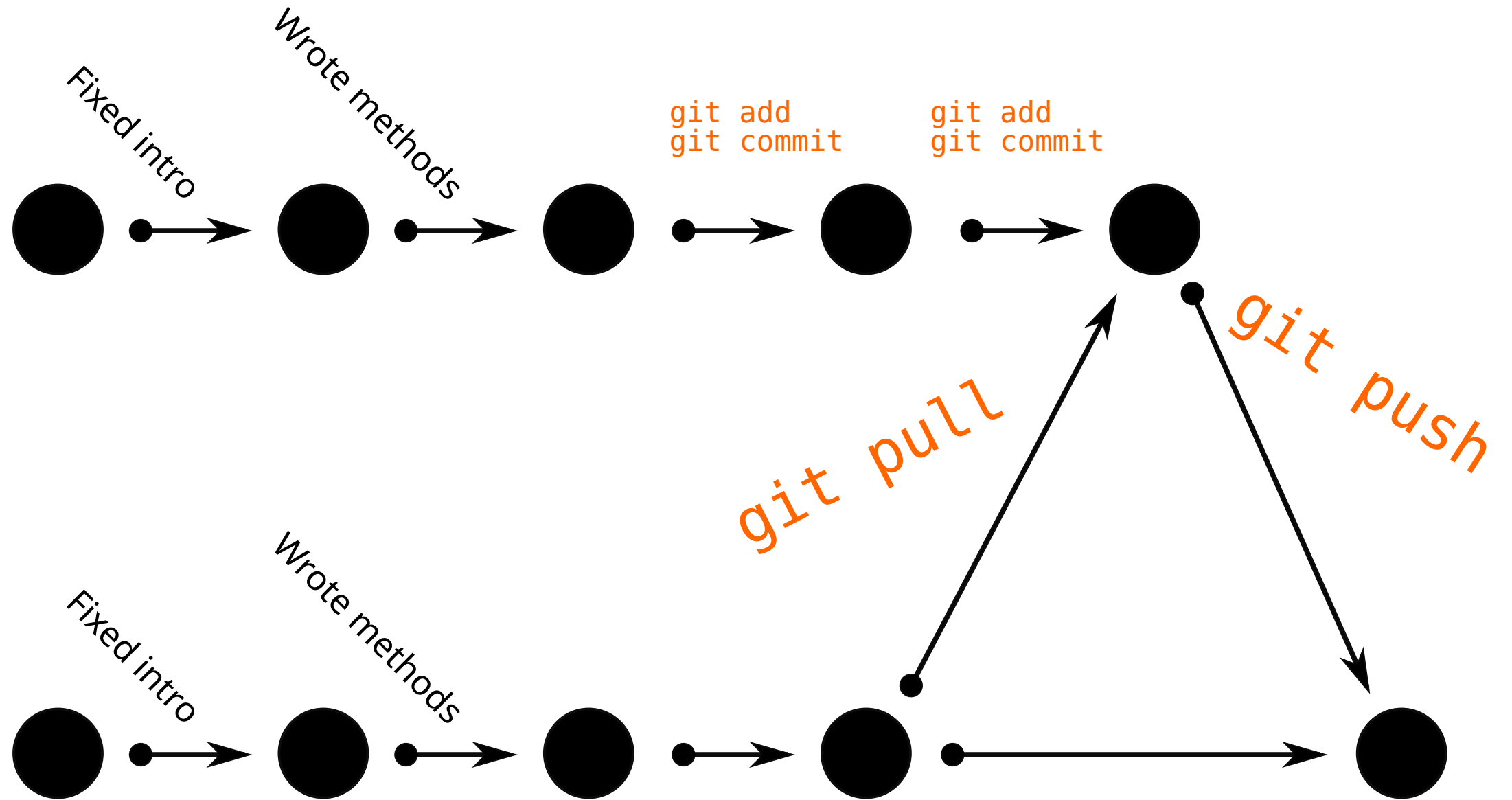
Un autre dépôt



git pull



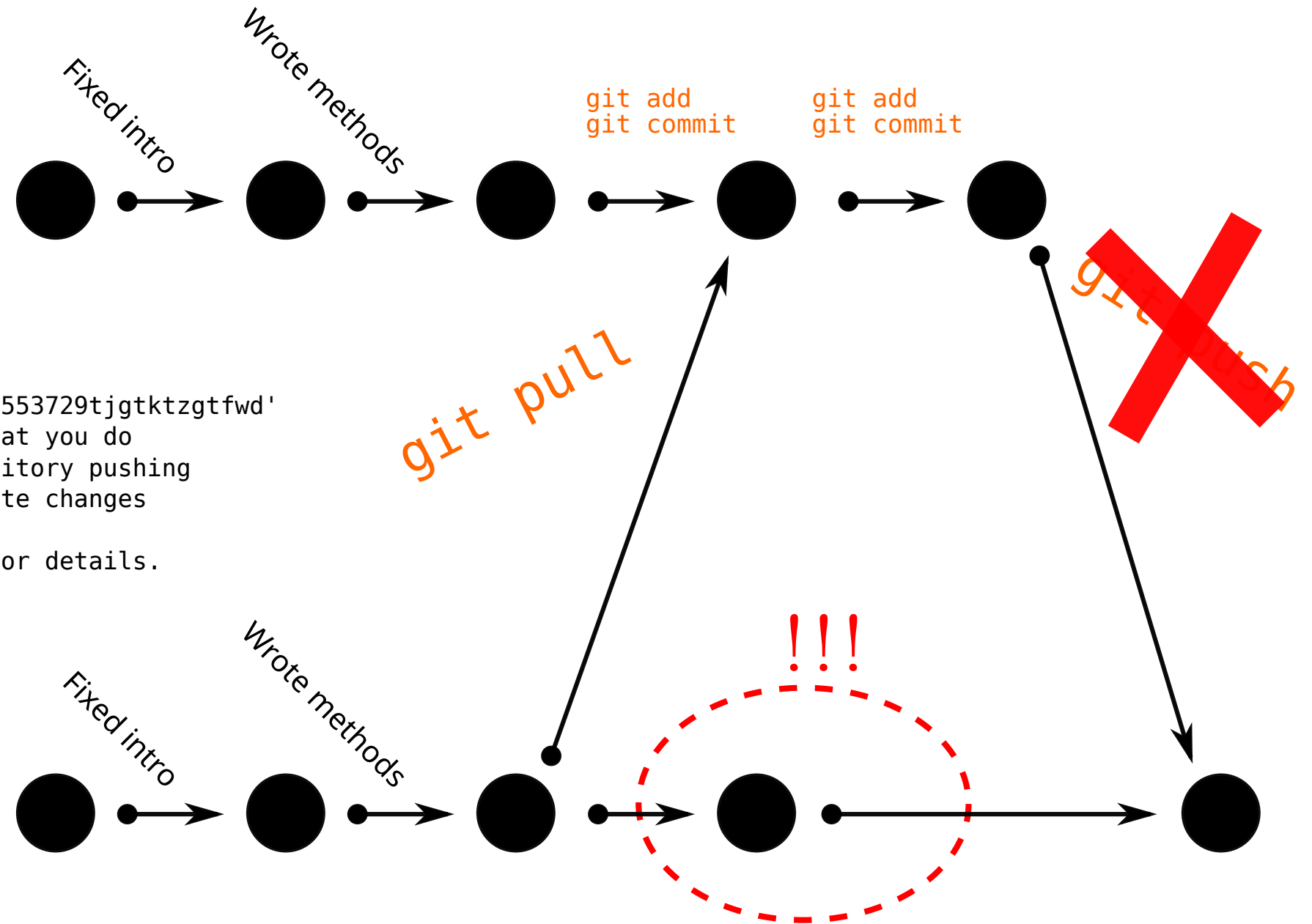
Dans votre dépôt



Un autre dépôt

Problèmes... changements non-présents dans votre dépôt

Dans votre dépôt

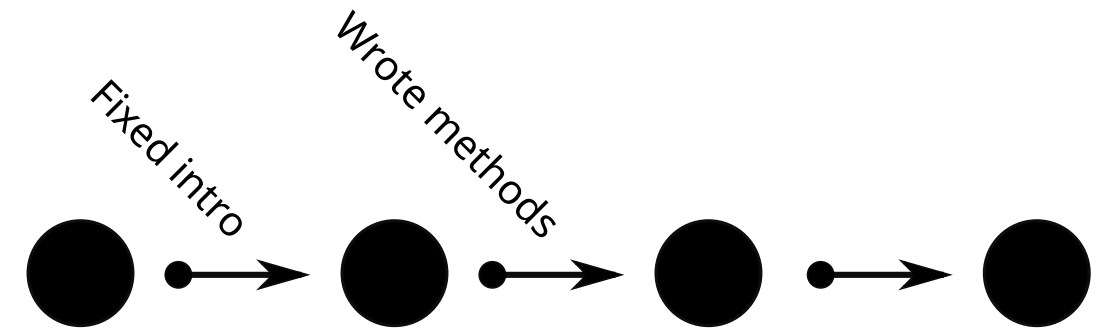


```
> git push
To https://git.overleaf.com/15553729tjgkktzgtfwd
 ! [rejected]      master -> master (fetch first)
error: failed to push some refs to 'https://git.overleaf.com/15553729tjgkktzgtfwd'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Un autre dépôt

Problèmes... conflits lors d'un `git pull` (merge)

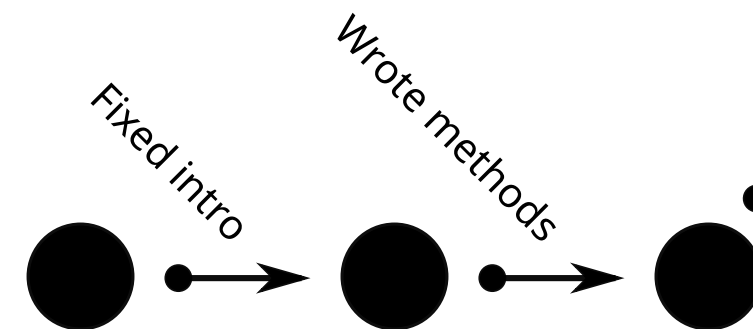
Dans votre dépôt



```
From https://git.overleaf.com/15553729tjgkktzgtfwd
 caefe81..599757b  master    -> origin/master
Auto-merging main.tex
CONFLICT (content): Merge conflict in main.tex
Automatic merge failed; fix conflicts and then commit the result.
```

~~git pull~~

Un autre dépôt



Problèmes... conflits lors d'un `git pull` (merge)

Votre version

```
1 #  
2 # Script d'analyse de données  
3 #  
4  
5 library(palmerpenguins)  
6  
7 ggplot(penguins) +  
8   geom_point(aes(x = body_mass_g, y = flipper_length_mm, color = species)) +  
9   labs(x = "Body mass (grams)",  
10        y = "Flipper length (mm)")  
11
```

Pas fusionnable automatiquement !

La version d'ailleurs

```
1 #  
2 # Script d'analyse de données  
3 #  
4  
5 library(palmerpenguins)  
6  
7 ggplot(penguins) +  
8   geom_point(aes(x = body_mass_g, y = flipper_length_mm, color = species)) +  
9   labs(x = "Body mass (g)",  
10        y = "Flipper length (mm)")  
11  
12 stats <- penguins %>%  
13   group_by(species) %>%  
14   summarise(body_mass_mean = mean(body_mass_g))  
15
```

Fusionnable automatiquement

Problèmes... conflits lors d'un `git pull` (merge)

Après le *git pull*

```
1 #
2 # Script d'analyse de données
3 #
4
5 library(palmerpenguins)
6
7 ggplot(penguins) +
8   geom_point(aes(x = body_mass_g, y = flipper_length_mm, color = species)) +
9   <<<<<< HEAD
10  labs(x = "Body mass (grams)",
11       <=====  
12  labs(x = "Body mass (g)",
13  >>>>>> 599757bbe3dd97df3ee210c8da56322cc531f119
14  y = "Flipper length (mm)")
15
```

Votre version

"Leur" version

Problèmes... conflits lors d'un `git pull` (merge)

Après le *git pull*, on résoud les conflits

```
1 #
2 # Script d'analyse de données
3 #
4
5 library(palmerpenguins)
6
7 ggplot(penguins) +
8   geom_point(aes(x = body_mass_g, y = flipper_length_mm, color = species)) +
9   <<<<<<< HEAD
10  labs(x = "Body mass (grams)",
11       <=====  
12  labs(x = "Body mass (g)",
13  >>>>>> 599757bbe3dd97df3ee210c8da56322cc531f119
14  y = "Flipper length (mm)")
15
```

Votre version

Problèmes... conflits lors d'un `git pull` (merge)

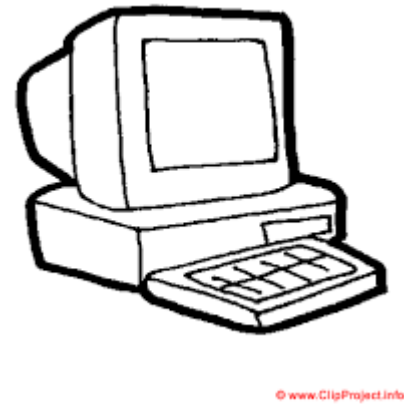
Après le *git pull*, on résoud les conflits, puis *git commit* !

```
1 #
2 # Script d'analyse de données
3 #
4
5 library(palmerpenguins)
6
7 ggplot(penguins) +
8   geom_point(aes(x = body_mass_g, y = flipper_length_mm, color = species)) +
9   <<<<<<< HEAD
10  labs(x = "Body mass (grams)",
11       <=====  
12  labs(x = "Body mass (g)",
13  >>>>>> 599757bbe3dd97df3ee210c8da56322cc531f119
14  | | | | y = "Flipper length (mm)")
15
```

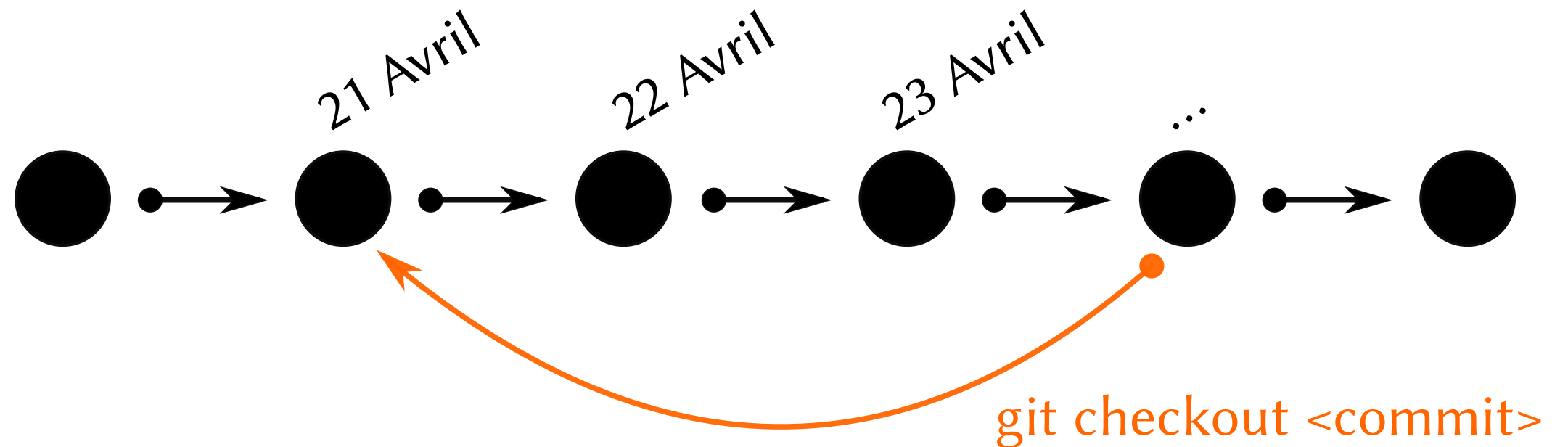
Votre version

En pratique...

Quelques exemples d'organisation



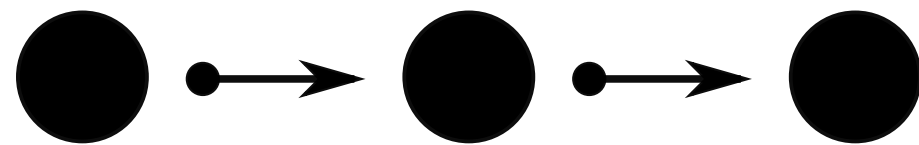
Dans un seul dossier, sauvegarde de l'historique



Quelques exemples d'organisation

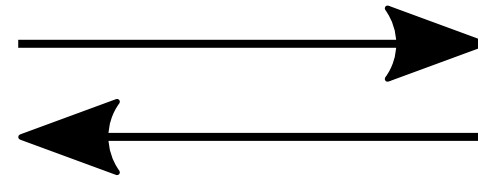


© www.ClgProject.info



Dossier local

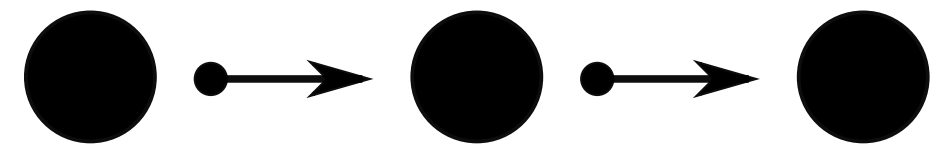
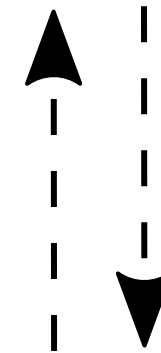
git push



git pull

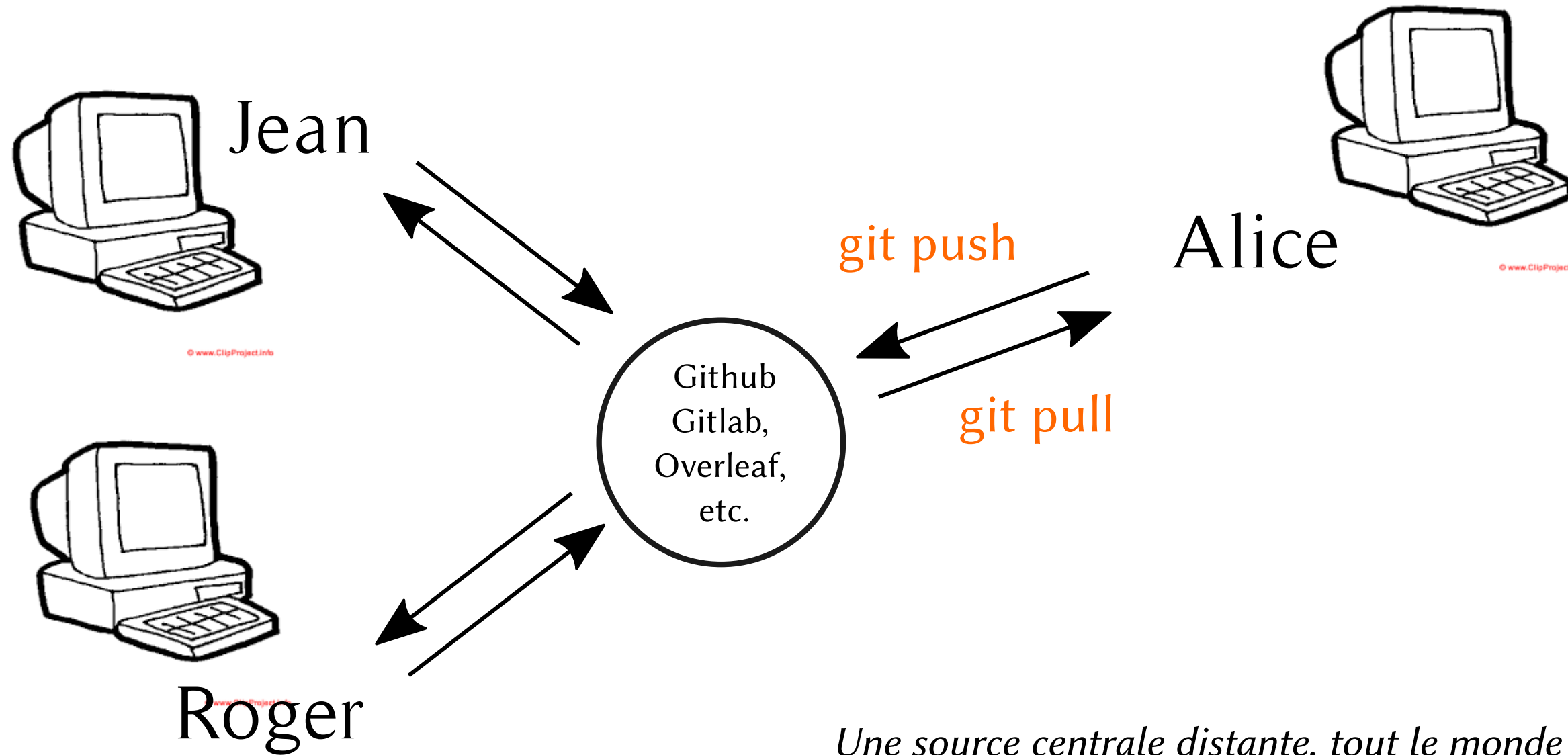


Les autres



*Dossier partagé
sur le même ordi*

Quelques exemples d'organisation



Une source centrale distante, tout le monde dialogue avec elle

Interfaces graphiques

git-gui (inclus sous windows)

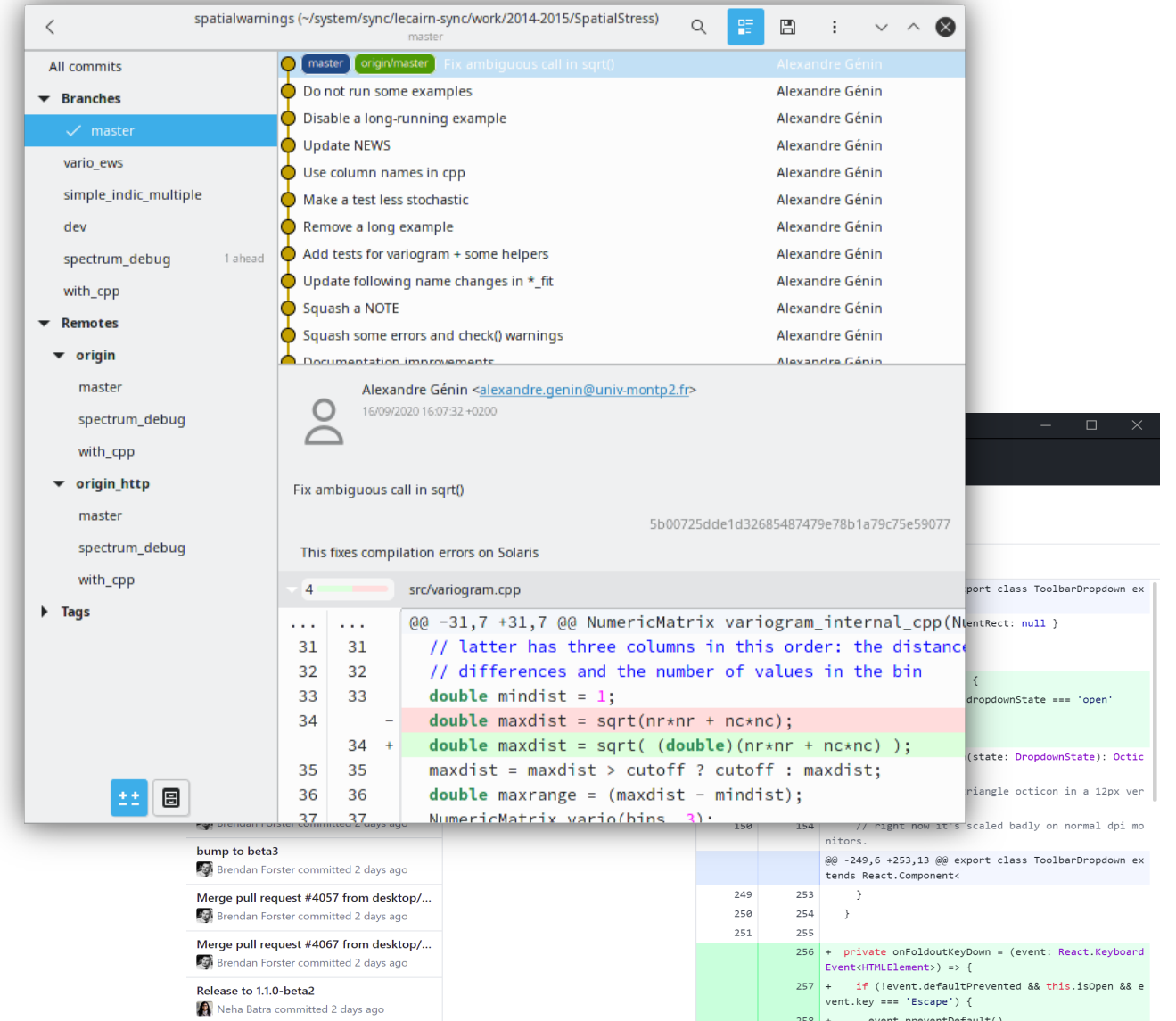
gitg (linux)

Rstudio (partout)

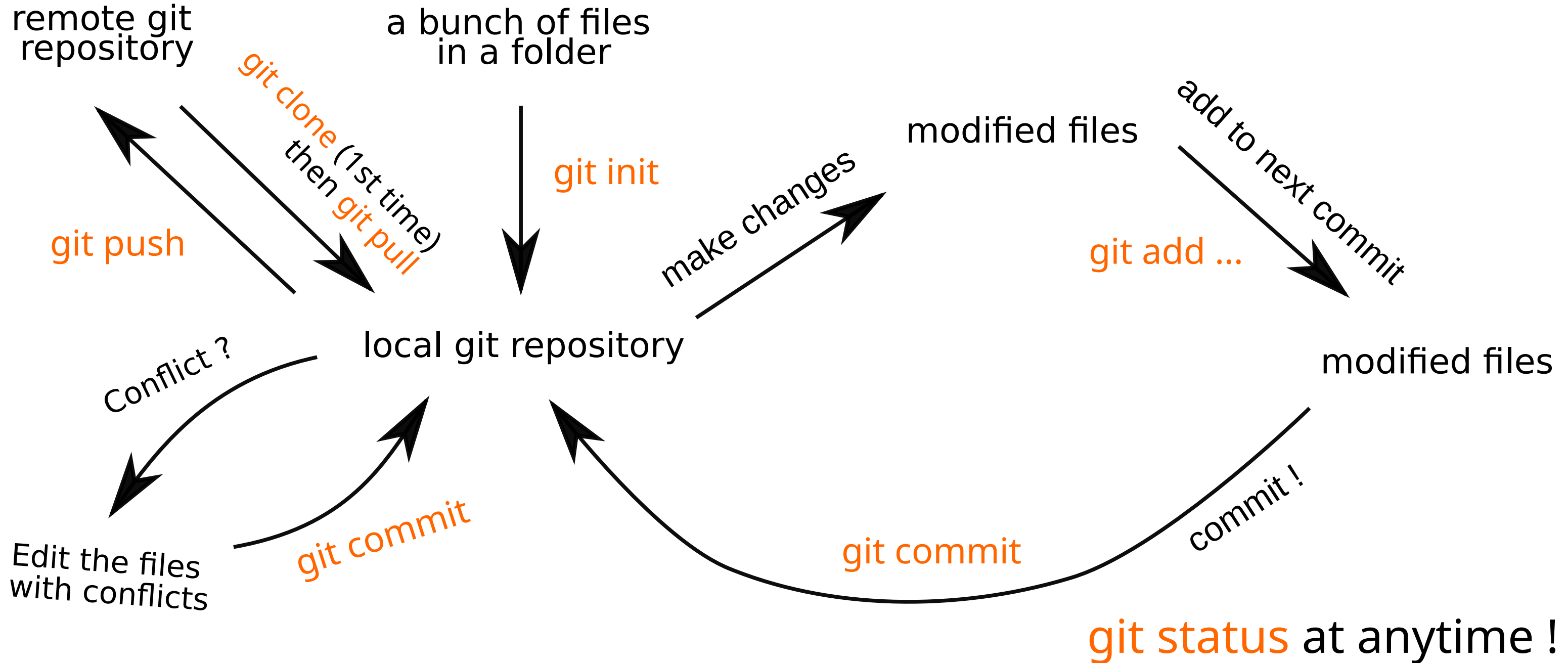
etc, etc.

Et plein d'autres...

<https://git-scm.com/downloads/guis>



En un graphique !



Recap:

- Les grands concepts de git (dépôt, commit, etc.)
- Utiliser git pour soi-même (commit loop)
- Collaborer avec git (git pull/push, conflits)

Atelier: jeudi 2 décembre, 14h !

double atelier !

Tous les exercices et infos sur <https://rrr.mbb.cnrs.fr>