



IMT Mines Albi-Carmaux
École Mines-Télécom

INITIATION À GITHUB

IFIE3 - EAE/ETN

Valentin PÈRE

SOMMAIRE

1. POURQUOI UTILISER GITHUB ?
2. PREMIERS PAS
3. TRAVAIL EN BRANCHES



IMT Mines Albi-Carmaux
École Mines-Télécom

| POURQUOI UTILISER GITHUB ?



IMT Mines Albi-Carmaux
École Mines-Télécom



git

Système de contrôle de version distribué

- Développer à plusieurs sur le même répertoire.
- Revenir en arrière lorsque vous faites une erreur... Après des heures de programmation

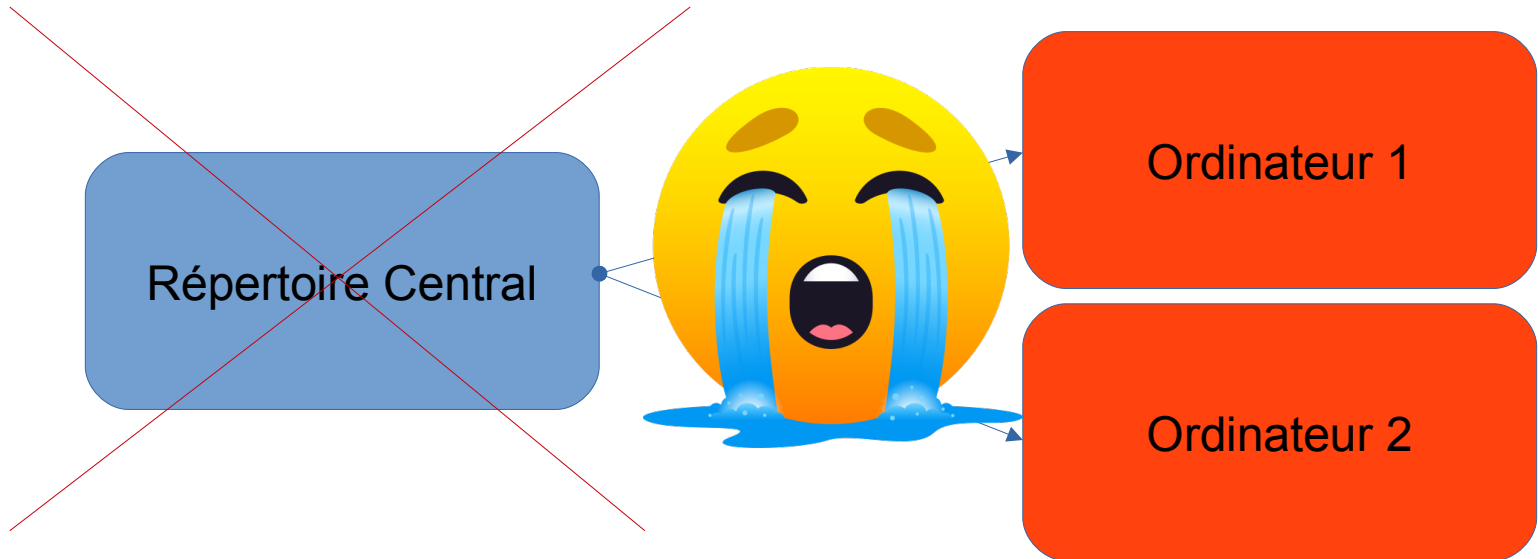
Avec Git, vous donnez des **versions** de votre code, que vous pouvez ré-utiliser en cas de besoin



git

Système de contrôle de version distribué

Cas centralisé :



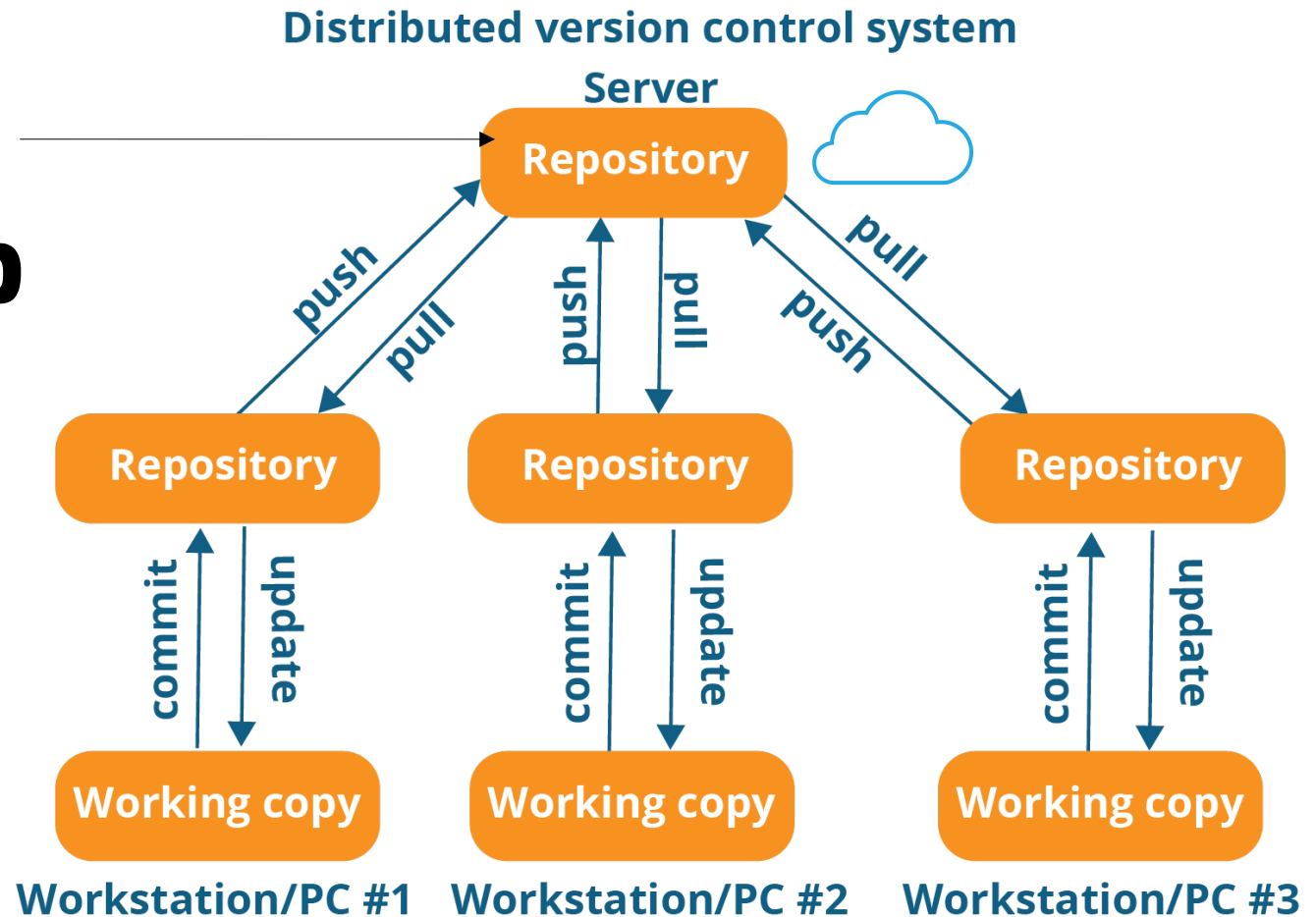
I : POURQUOI GITHUB ?

1.1 Qu'est ce que Git et GitHub ?

6



Si l'ordinateur ou le serveur central crash, pas de soucis !



I : POURQUOI GITHUB ?

7

1.1 Qu'est ce que Git et GitHub ?



Ensemble **d'outils** pour gérer des répertoires contenant du code.

Gratuit, openSource



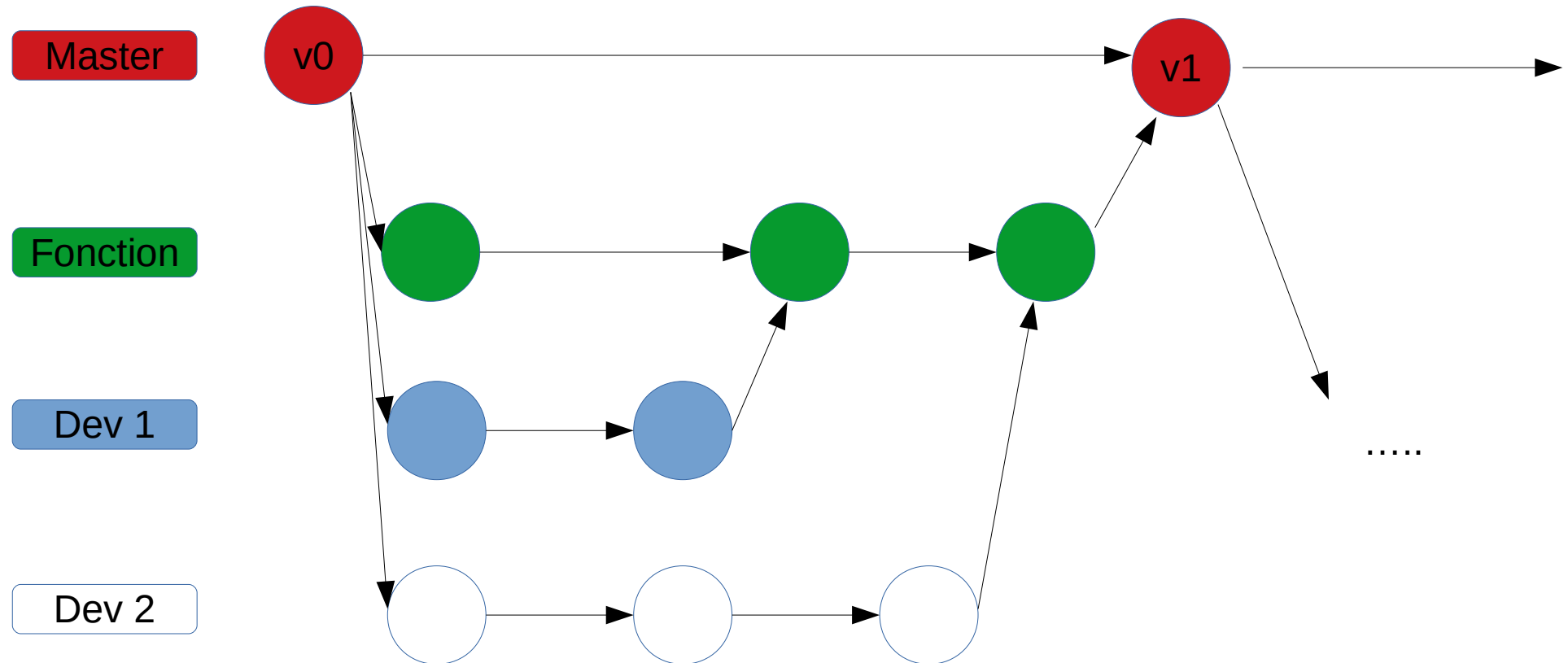
Services d'hôte de répertoires Git.

Ils sont capables d'utiliser tous les codes sources et fonctionnalités de Git.

I : POURQUOI GITHUB ?

1.1 Qu'est ce que Git et GitHub ?

8



C'est parti pour tester tout ça !

- Ouvrez votre navigateur et tapez « download Git », cliquez sur le premier lien et téléchargez Git selon votre système d'exploitation.
- Suivez l'installation.
- Créez un dossier, puis un fichier texte (.txt) et tapez la phrase de votre choix avant de sauvegarder (Ctrl+S).
- Si vous êtes sous Windows, faites un click droit dans le dossier et cliquez sur « Git Bash here ».

1.2 Configuration de git

Entrez les lignes de commandes suivantes avec votre nom et votre adresse mail

```
(base) vpere@I6488:~/Documents/Cours/Git/test_git$ git config --global user.name "Valentin"
(base) vpere@I6488:~/Documents/Cours/Git/test_git$ git config --global user.email "valentin.pere@mines-albi.fr"
```

Dîtes à Git qu'il doit associer sa structure au dossier en question :

```
(base) vpere@I6488:~/Documents/Cours/Git/test_git$ git init
Dépôt Git vide initialisé dans /home/vpere/Documents/Cours/Git/test_git/.git/
```

Des fichiers masqués sont apparus, votre répertoire a bien une structure git.

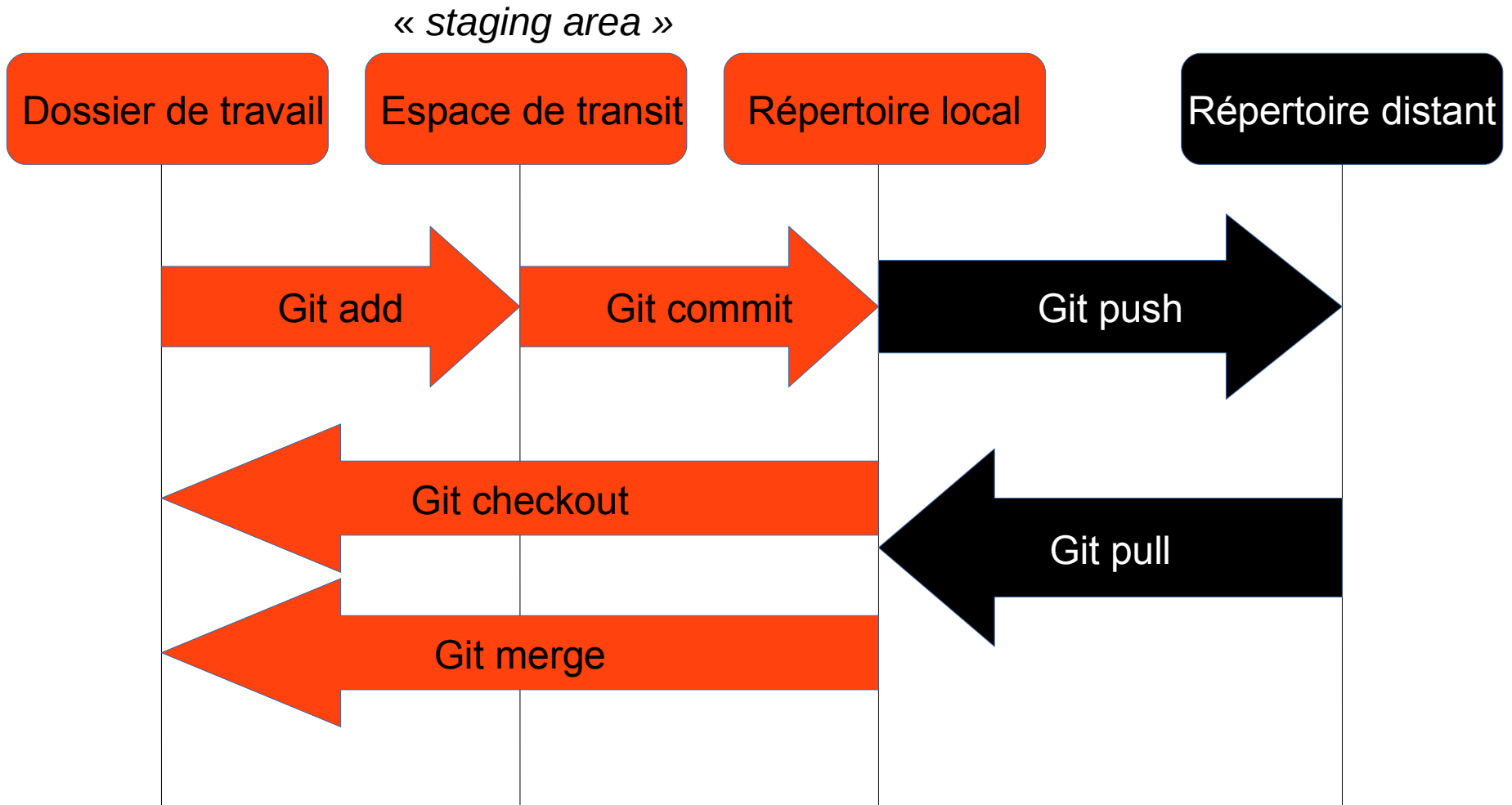


Git sera en mesure de suivre (ou « traquer ») l'évolution des fichiers dans ce dossier

II PREMIERS PAS



IMT Mines Albi-Carmaux
École Mines-Télécom



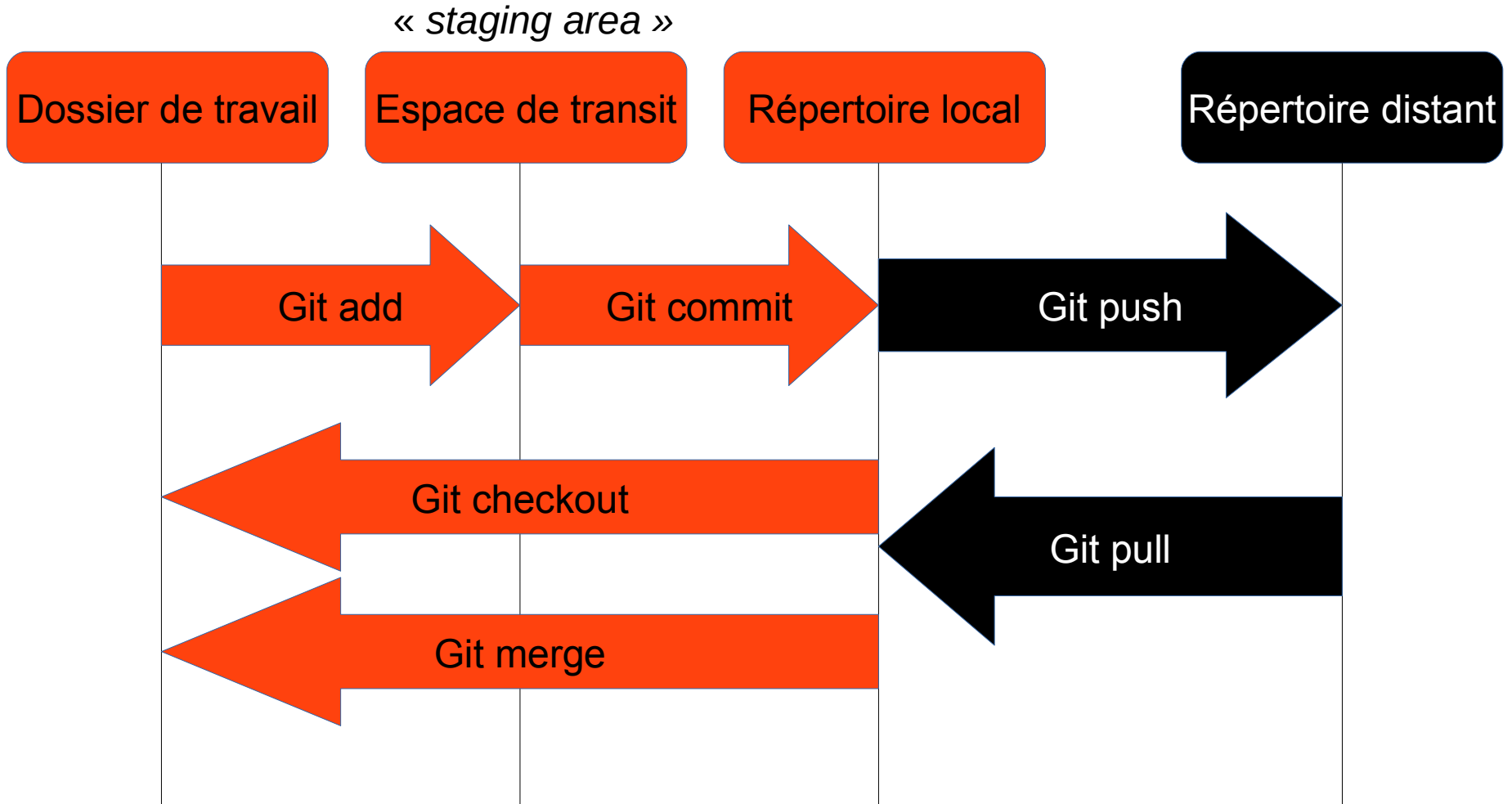
L'espace de **transit** est un espace sur lequel on positionne les fichiers modifiés avant de les mettre à jour dans le répertoire git local (puis distant).

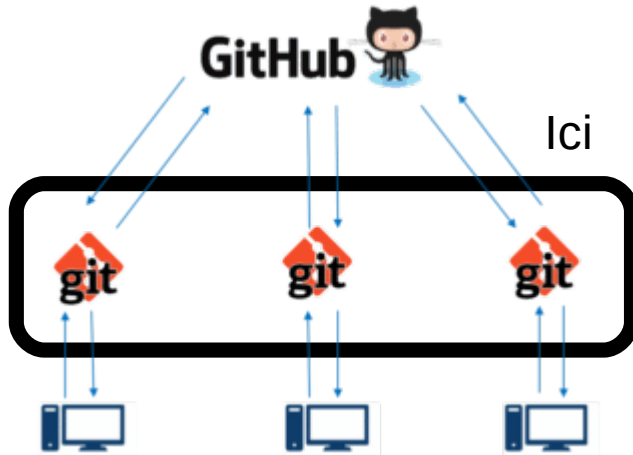
<git add> permet d'ajouter un ou plusieurs fichiers dans l'espace de transit :

- <git add [Nom du/des Fichier(s)]> ajoute les fichiers spécifiés dans l'espace de transit.
- <git add .> ajoute tous les fichiers du dossier de travail dans l'espace de transit.

<git rm --cached [Nom du/des fichiers]> sort les fichiers concernés de l'espace de transit.

Si ça ne fonctionne pas, essayer <git reset [Nom du fichier]>.

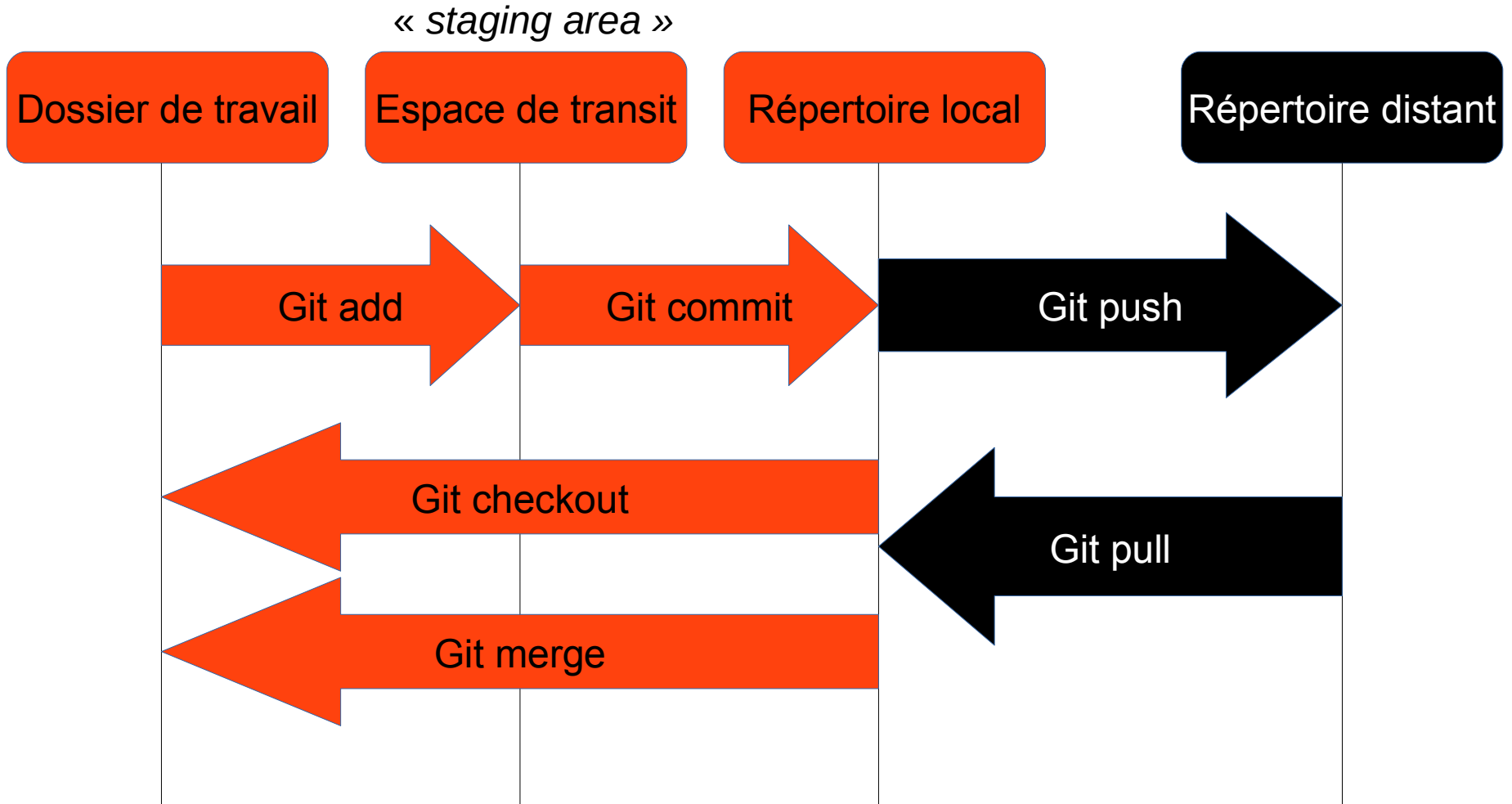




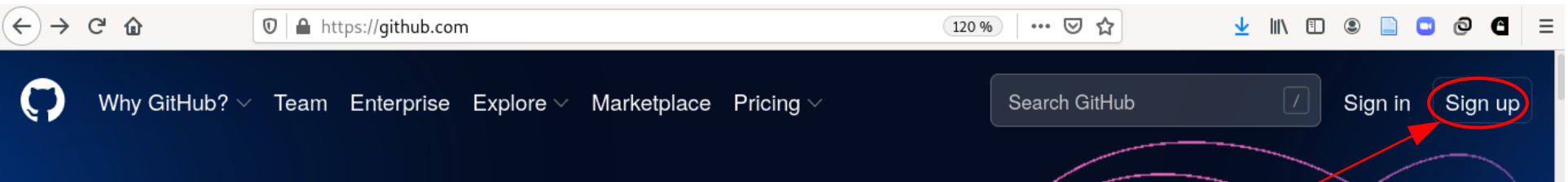
<git commit -m « [Votre message] »> met à jour le répertoire local Git selon les modifications des fichiers de l'espace latent.

<git status> permet de consulter les différences entre les fichiers dans l'espace de travail, l'espace de transit et le répertoire Git local. Il affiche les fichiers modifiés par rapport au répertoire local Git qui sont et ne sont pas dans l'espace de transit.

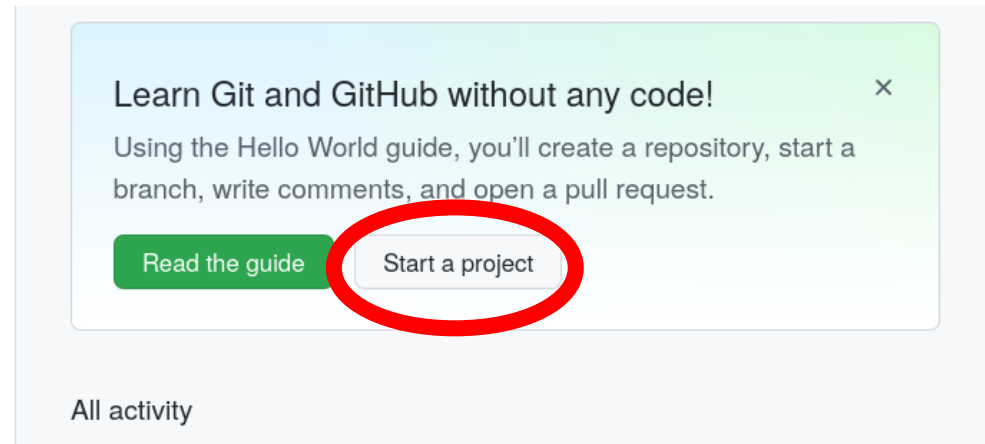
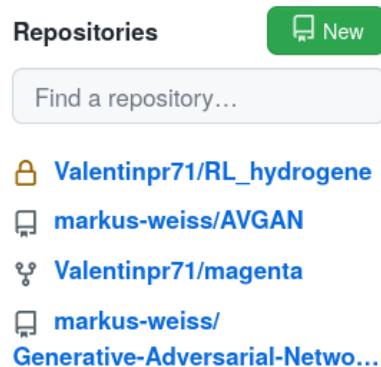
Conseil : Installer notepad++ pour voir des couleurs dans votre invité de commande Windows.



Inscrivez-vous sur GitHub



Puis créez un projet :



Nommez votre répertoire puis créez-le !

Enfin, copier le code proposé dans la section ci-dessous (depuis votre page de projet) et coller le dans votre terminal

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

```
git remote add origin https://github.com/Valentinpr71/Test_GitHub.git  
git branch -M main  
git push -u origin main
```



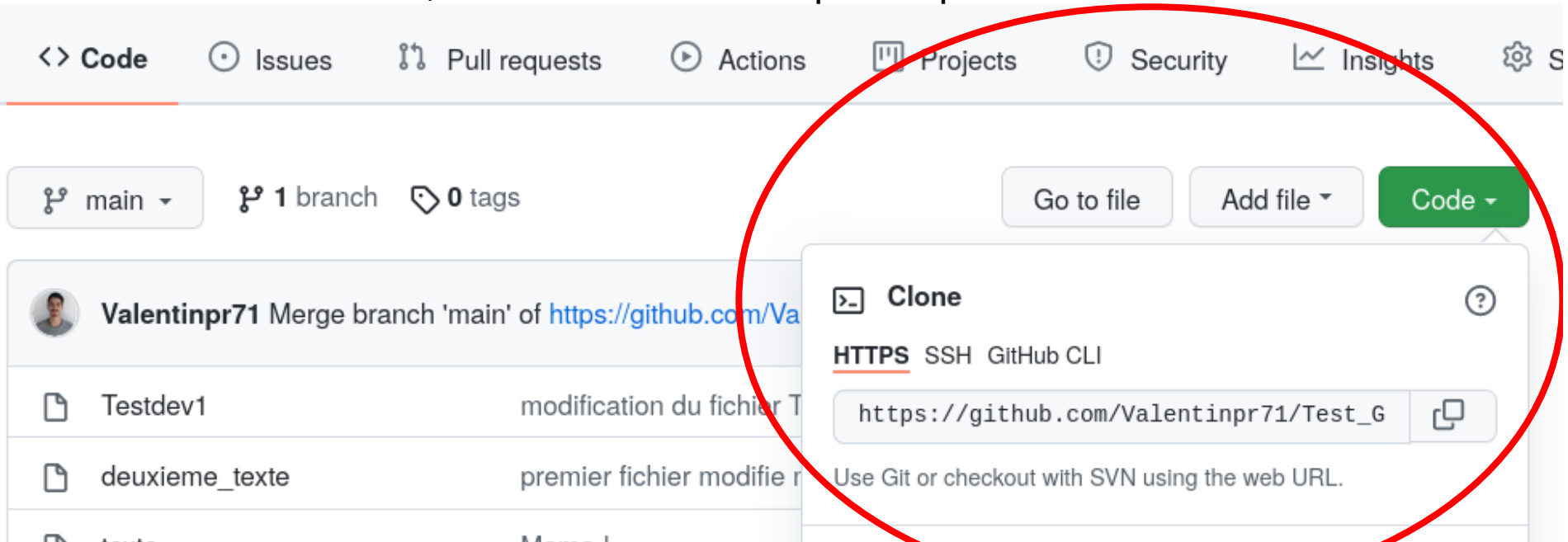
Puis identifiez-vous avec votre log-in GitHub

Une fois fait, tous vos *commits* seront visibles sur le cloud GitHub !

La commande `<git push>` permet d'envoyer vos *commits* de branche sur GitHub

Créez un nouveau dossier en dehors de votre dossier lié à Git.

Retournez sur GitHub, cliquez sur « Code » puis copiez le lien HTTPS :



Collez la ligne copiée dans votre terminal de commande.

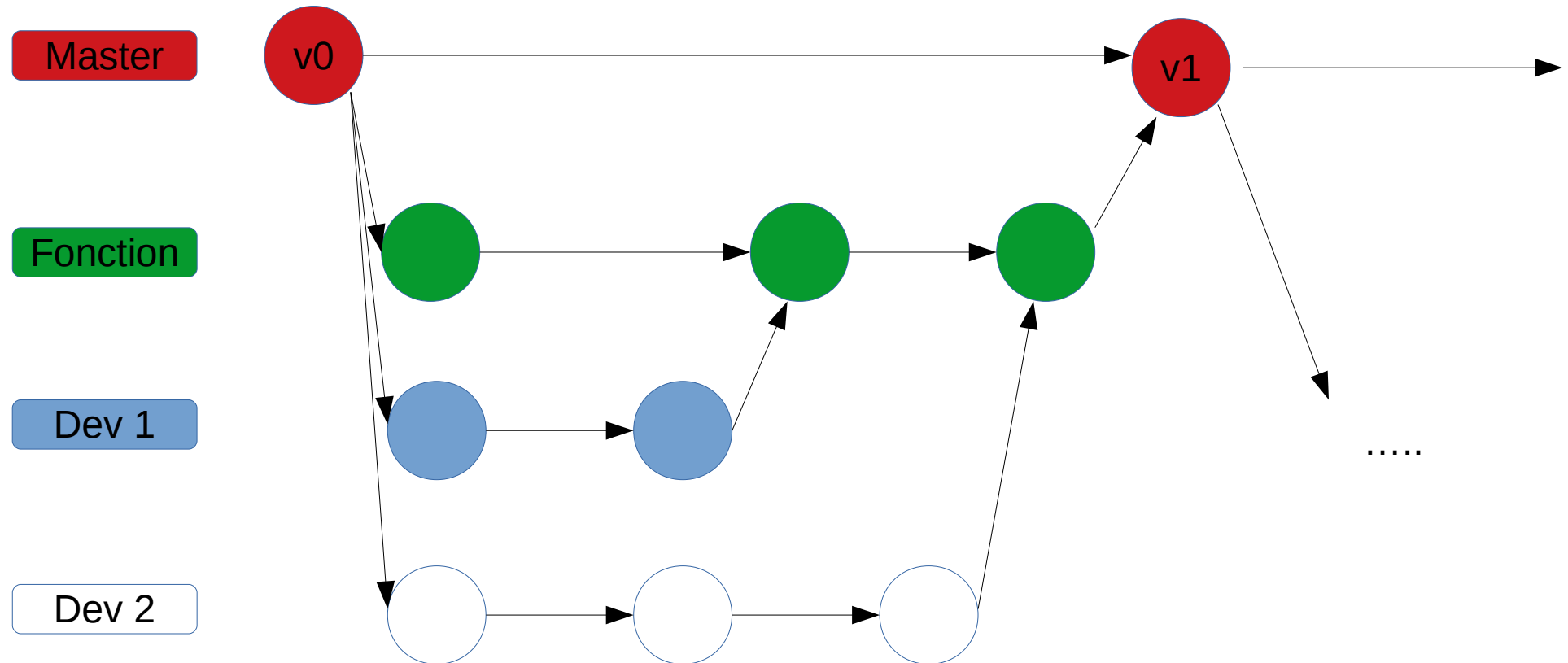
Votre dossier est désormais la copie conforme de votre projet GitHub, pratique pour le partager !

III

TRAVAIL EN BRANCHES



IMT Mines Albi-Carmaux
École Mines-Télécom

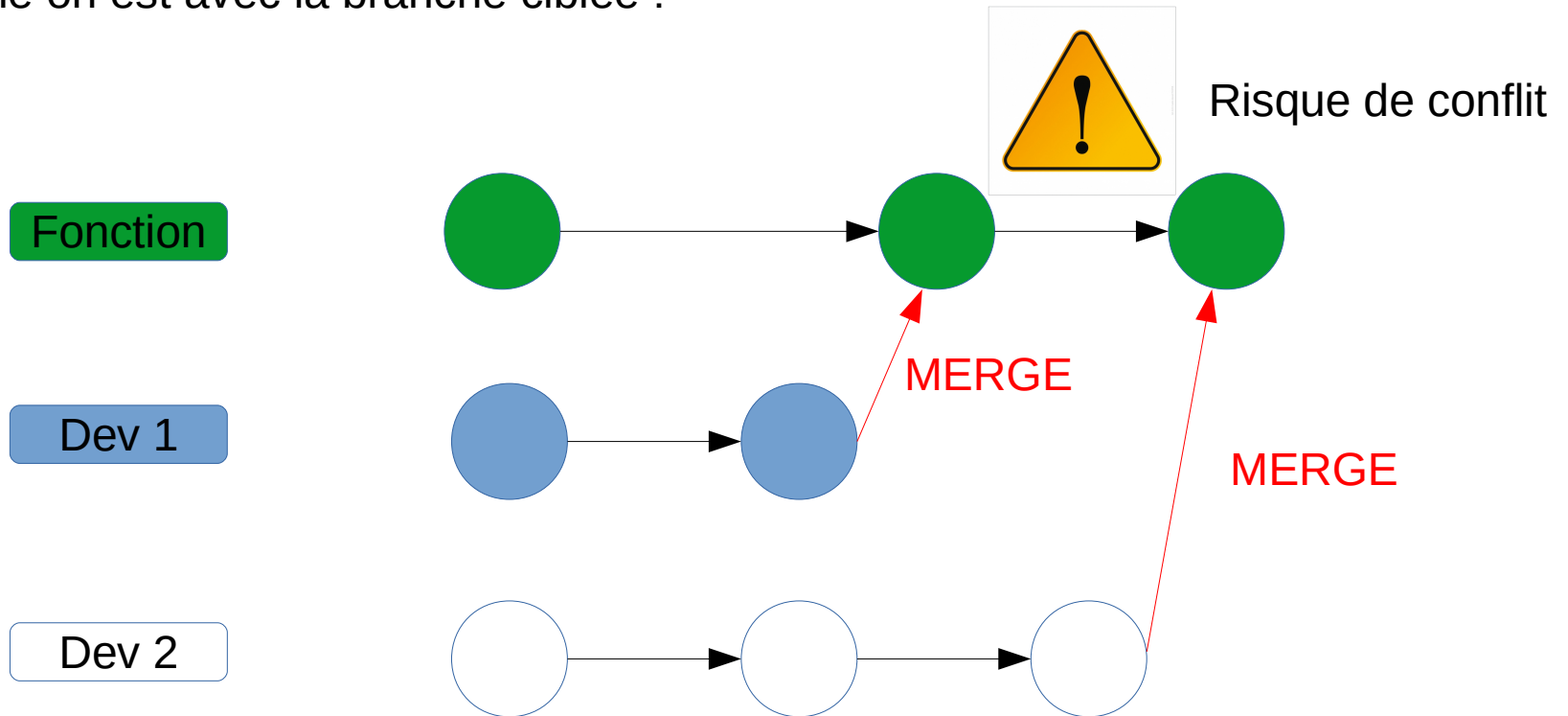


3.1 Création/Changement de branches

- Pour créer une branche : <git checkout -b [nom de la branche]>
- Pour consulter la liste des branches et voir où vous vous trouvez : <git branch>
- Pour se rendre dans une autre branche : <git checkout [nom de la branche]>
- Pour supprimer une branche : <git branch -d [nom de la branche]>



Faire un `<git merge [nom de la branche]>`, c'est fusionner la branche sur laquelle on est avec la branche ciblée :



Pour gérer les conflits éventuels :

<git diff [nom de la première branche] [nom de la deuxième branche]>

Cela devrait montrer les différences des deux branches dans leurs fichiers.

Plus qu'à modifier les fichiers, une fois fait :

<git add [nom du fichier modifié]> puis <git commit -m « ... »>

Et c'est tout bon, le merge devrait passer.

À vous de tester !

La commande <git push origin [nom de la branche distante]> va permettre de cloner la branche sur laquelle vous êtes sur la branche distante.

Attention !!!!

Pour ne pas écraser le travail d'un collègue, assurez vous d'avoir effectué un <git pull origin [nom de la branche distante]> avant de PUSH.

Ainsi, avec PULL vous vous mettrez à jour sur la branche distante et serez avertis en cas de conflit.

MERCI !



IMT Mines Albi-Carmaux
École Mines-Télécom