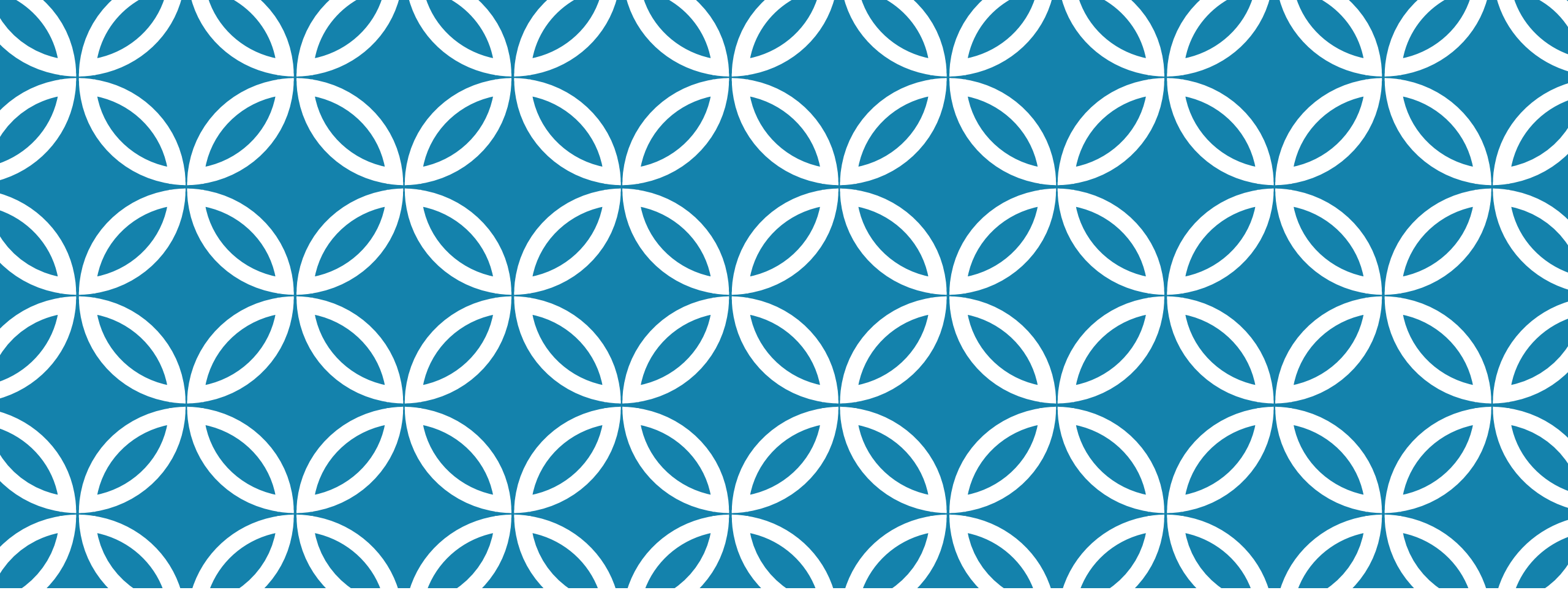
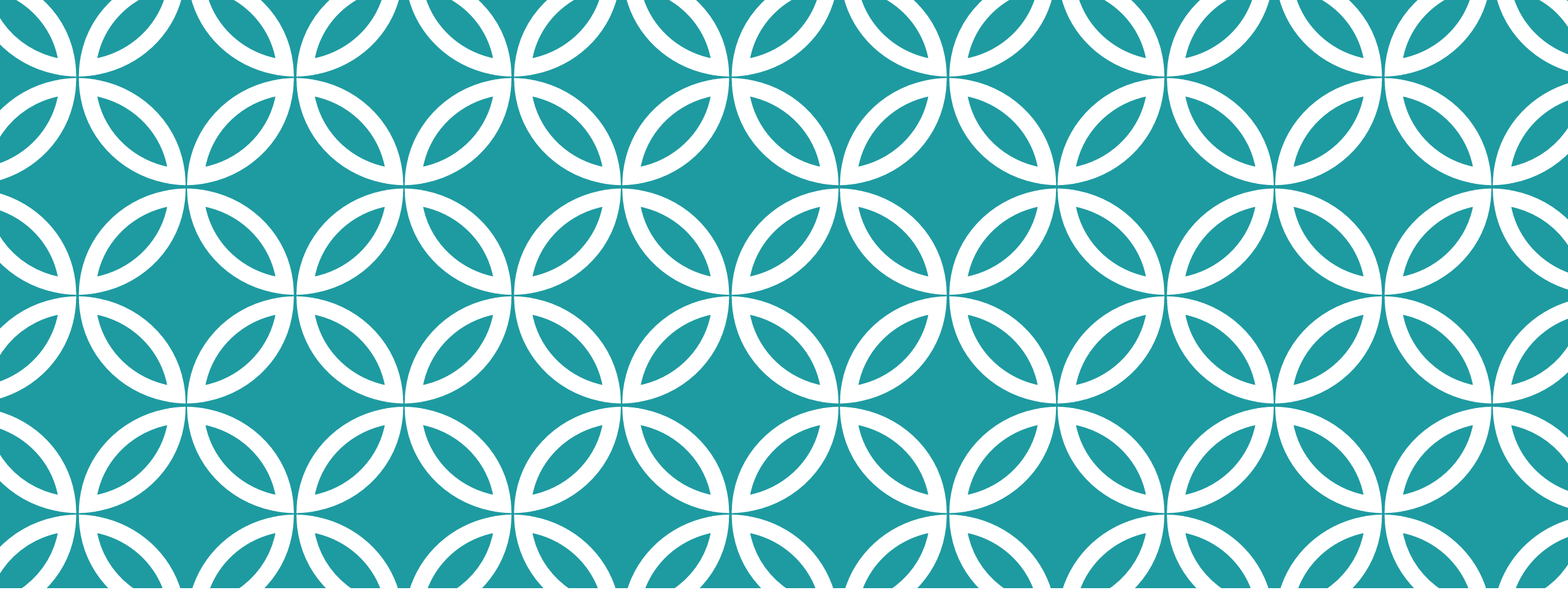


CONTRÔLE DE VERSIONS

Jérémy PERROUAULT



GIT

Introduction à GIT

INTRODUCTION

Système de contrôle de versions (ou gestionnaire de sources) **décentralisé**

- Chaque personne possède sa propre copie du dépôt sur son poste

Autorise plusieurs personnes à travailler sur des documents communs

- Chaque personne a une copie locale

Synchronise les différentes versions des documents

Permet un rollback des fichiers sur des versions précédentes

Permet de suivre les modifications au cours du temps

FONCTIONNEMENT

Créer un dépôt distant (git init)

Créer une copie de ce dépôt en local (git clone)

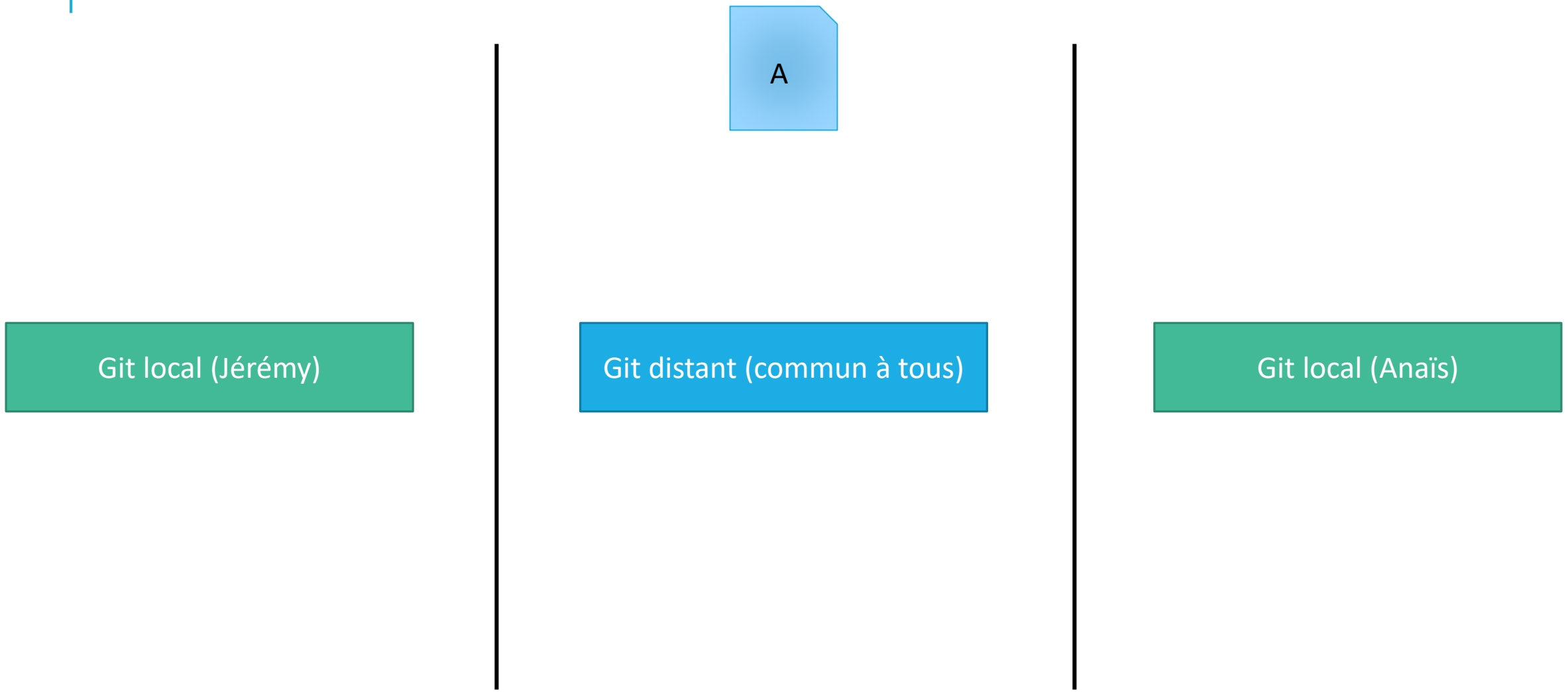
Faire un commit des changements en local (git commit)

Pousser les modifications sur le dépôt distant (git push)

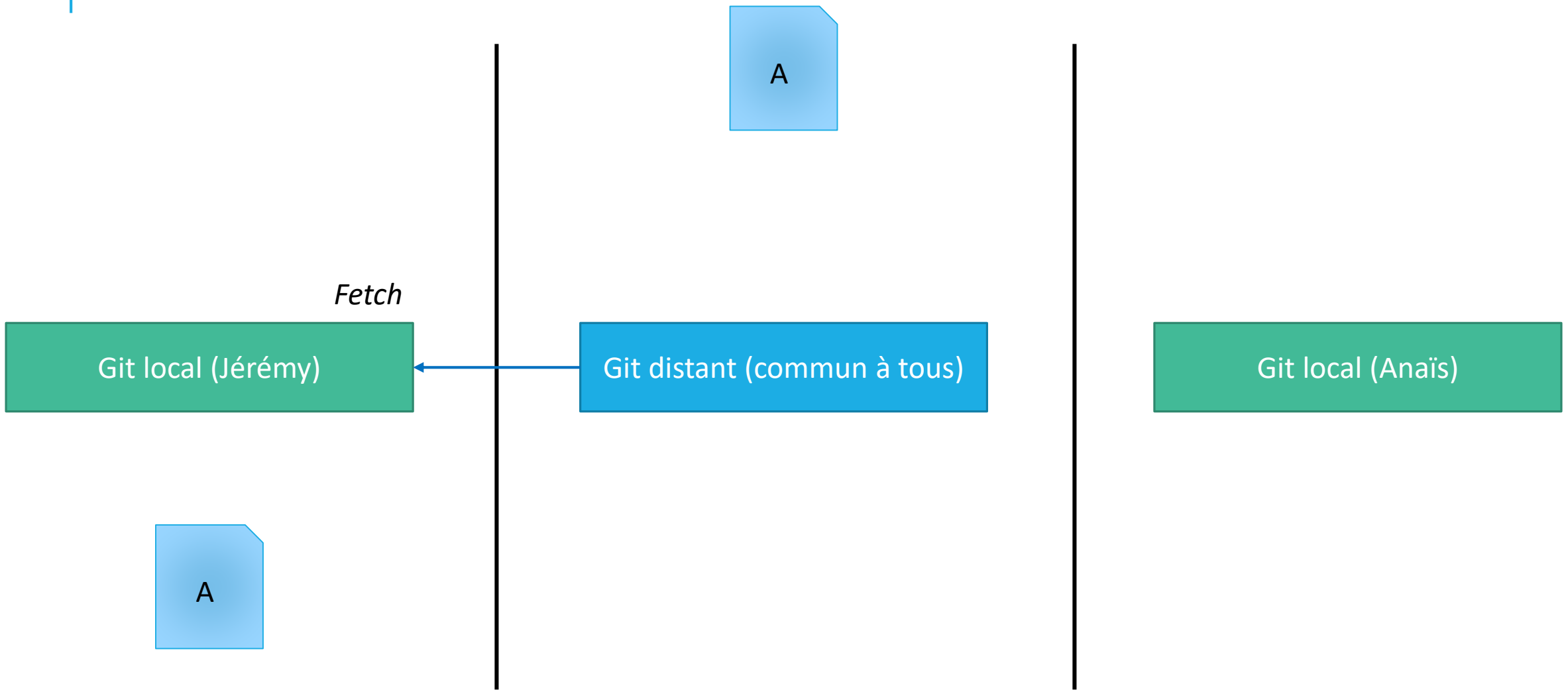
Récupérer les modifications des autres (git fetch)

Eventuellement rassembler des morceaux (git merge)

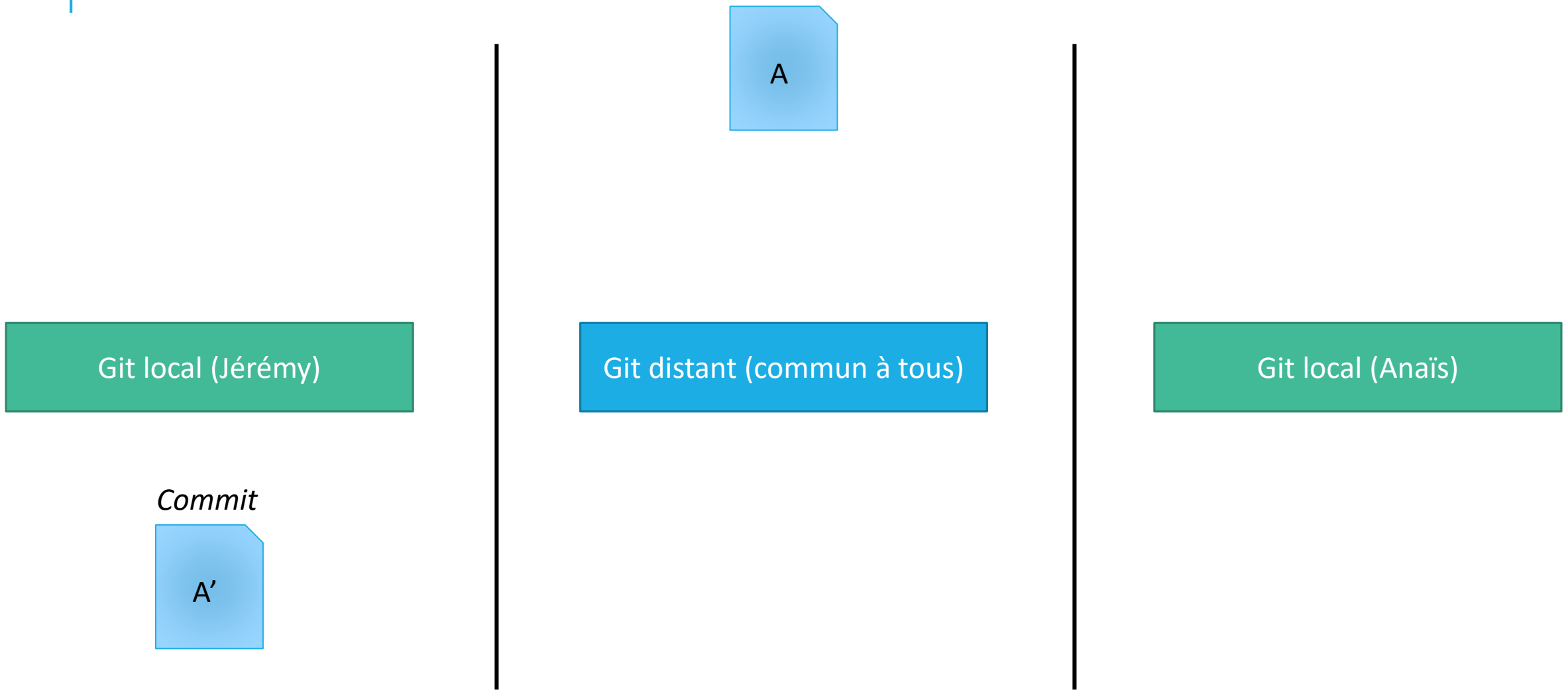
FONCTIONNEMENT



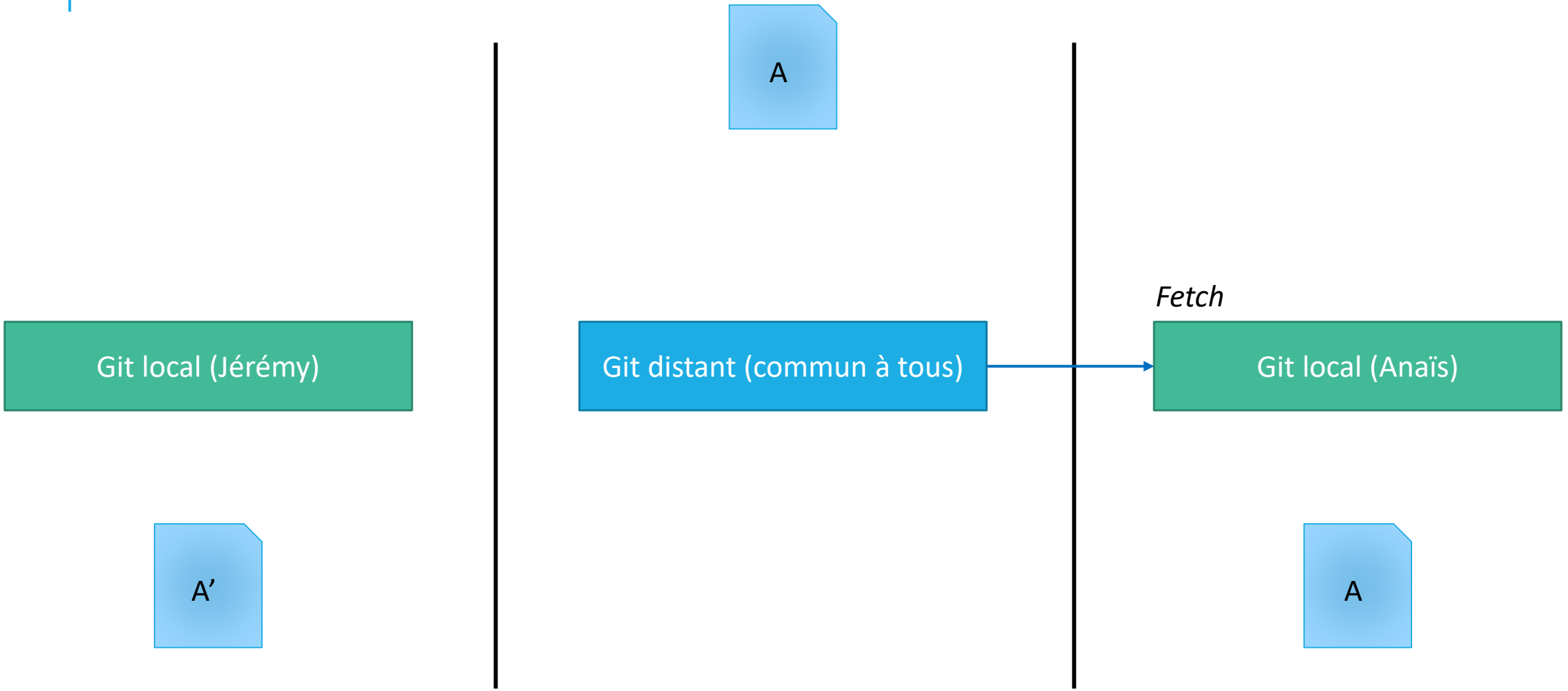
FONCTIONNEMENT



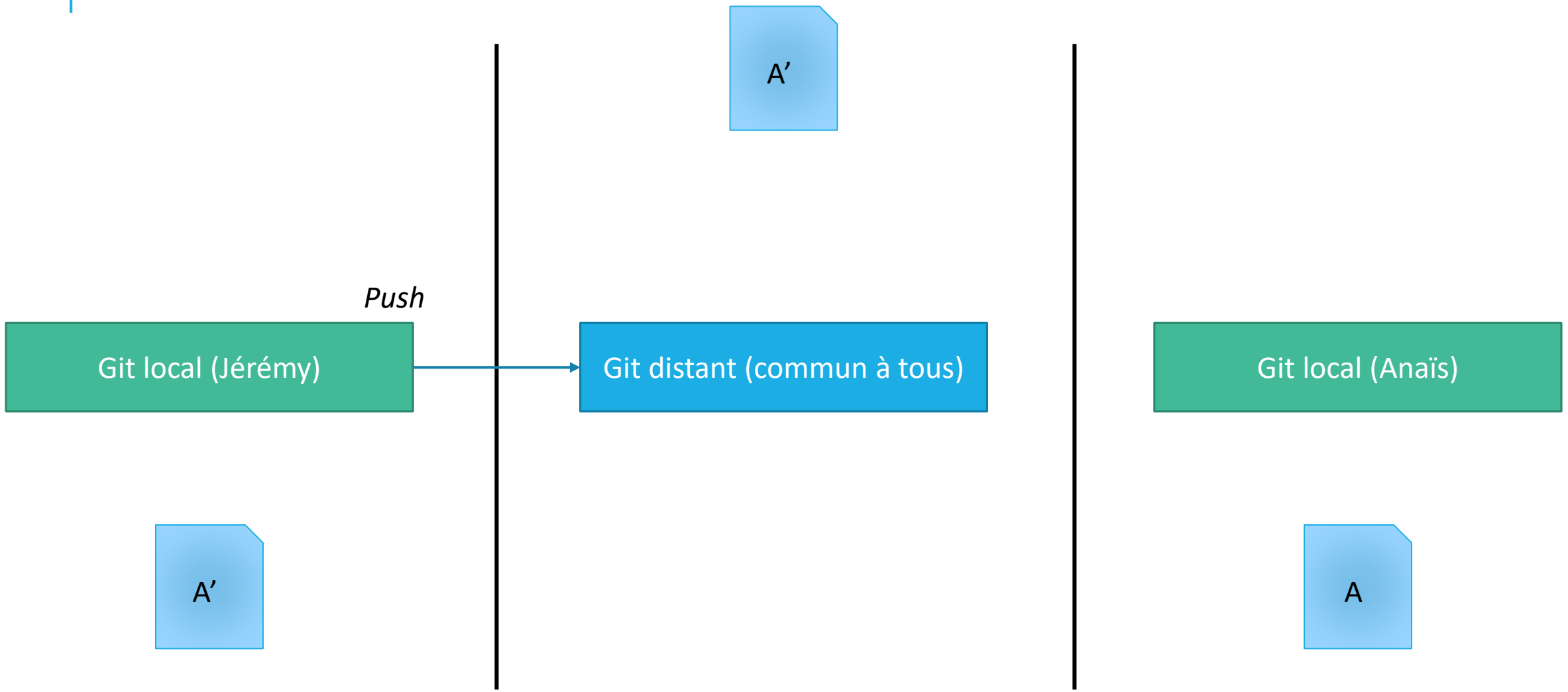
FONCTIONNEMENT



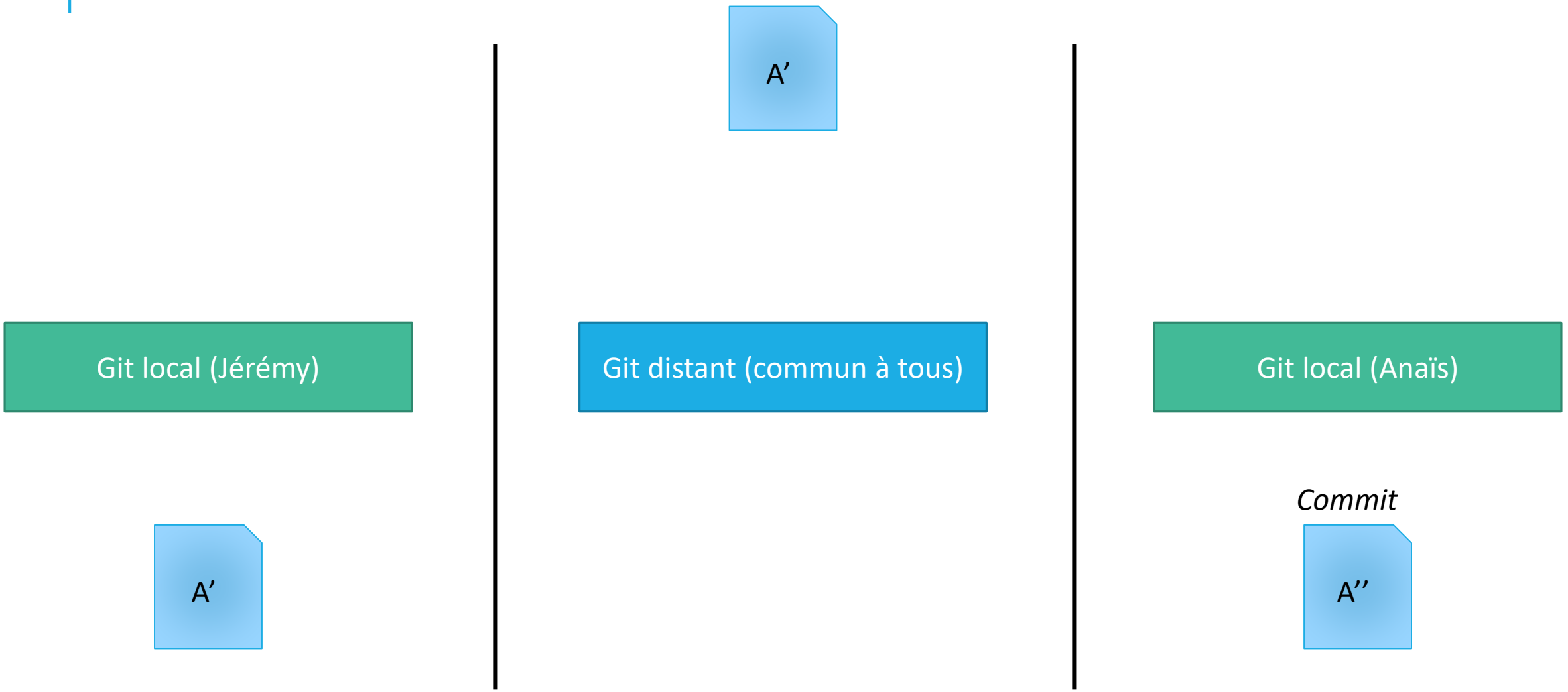
FONCTIONNEMENT



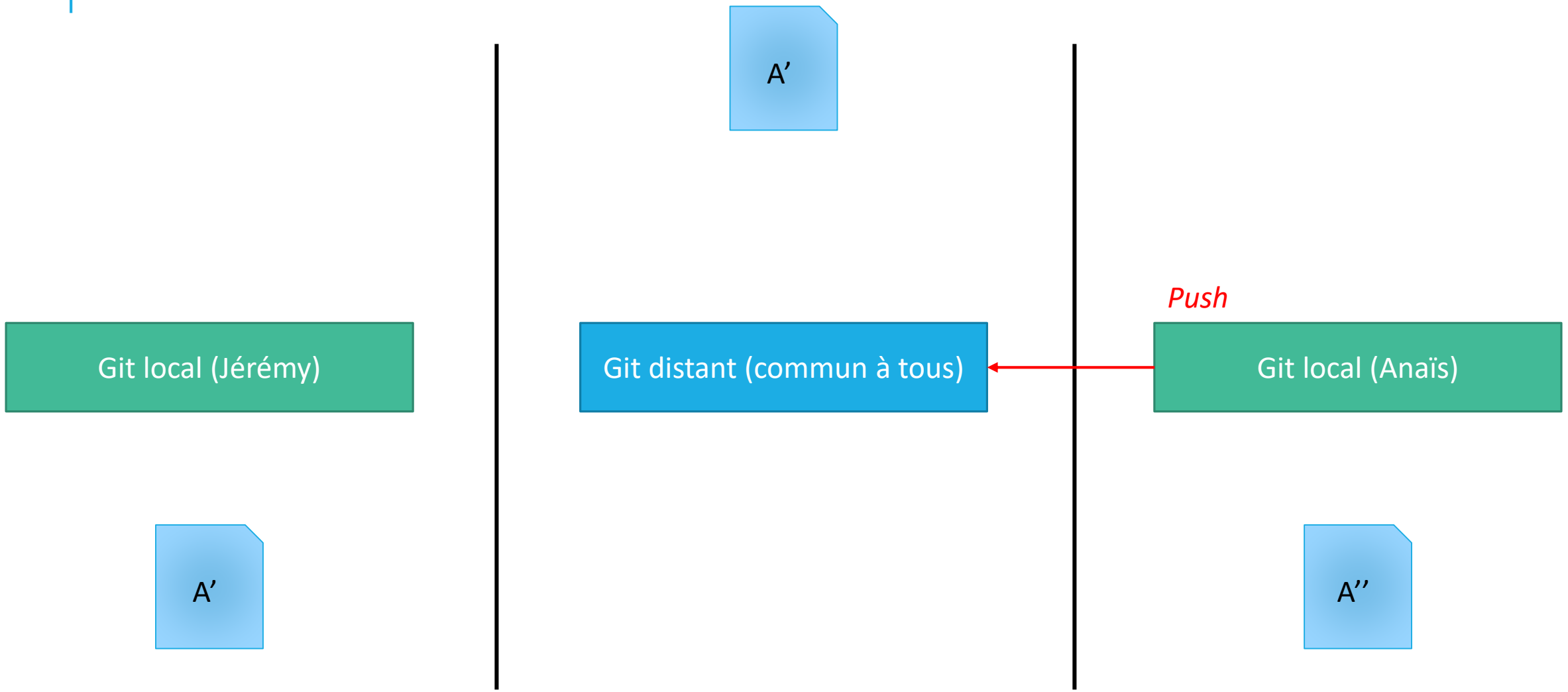
FONCTIONNEMENT



FONCTIONNEMENT



FONCTIONNEMENT



FONCTIONNEMENT

Dans le dernier cas, ça coince

- Il est impossible de pusher sur une branche qui n'est pas à jour

Il est nécessaire de mettre à jour sa branche

- Si des conflits sont détectés, il faudra les résoudre
  - Il est impossible de faire un commit si les conflits ne sont pas résolus

LE REMISAGE

Imaginons que dans le dernier cas, le commit n'ait pas été fait

- Mais que Anaïs cherche à mettre à jour sa branche (à récupérer les modifications)

D'une manière générale, le travail n'est pas terminé mais il faut soit :

- Changer de branche
- Faire un merge
- Récupérer le travail d'un partenaire
- Mettre à jour la branche

Périlleux de faire ces actions s'il y a du travail en cours !

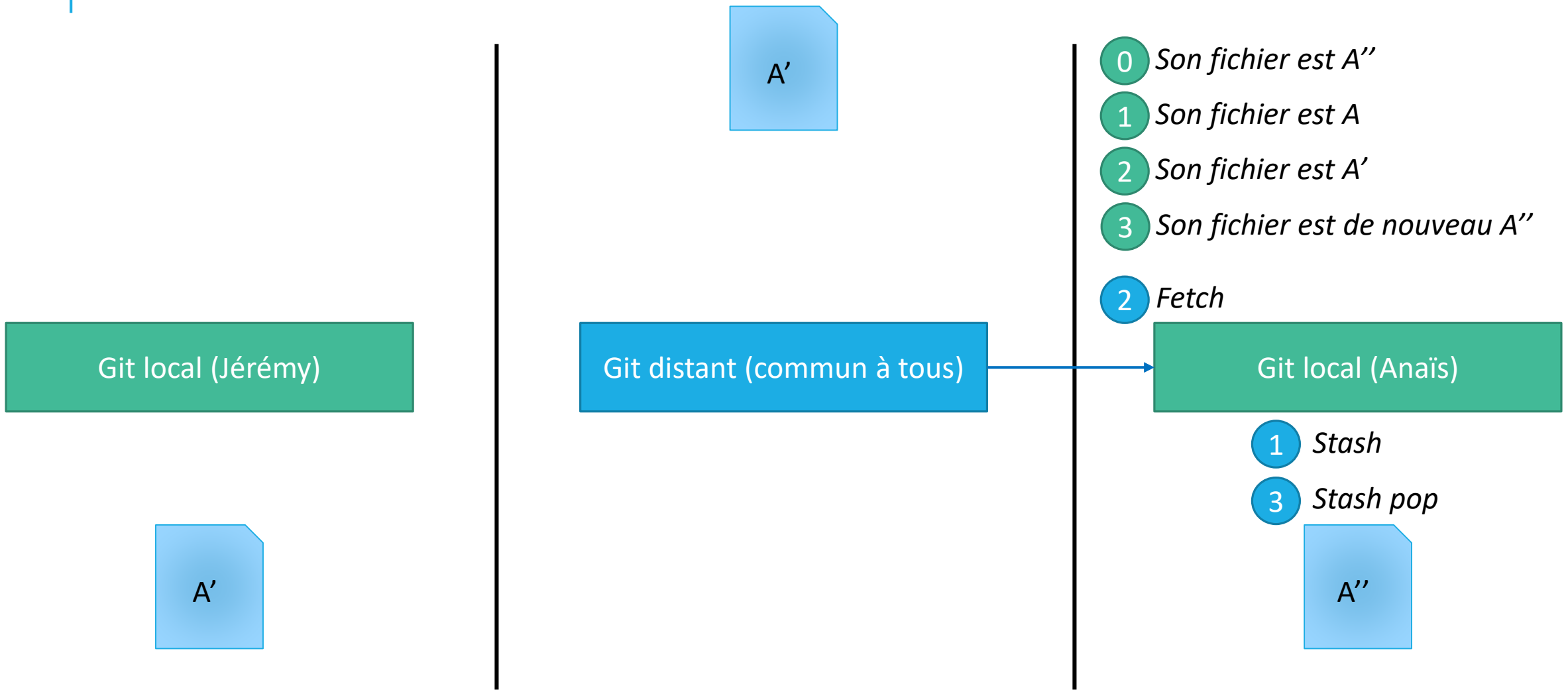
Il faut donc faire

- Un commit des changements
- Supprimer les changements
- Ou remettre les changements à plus tard : c'est le remise, avec la commande : « git stash »

Une fois que la manipulation est faite, il faut ré-appliquer les changements « stashés »

- Pour appliquer les changements : « git stash apply »
- Pour supprimer le stash : « git stash drop »
- Pour faire ces deux actions en une : « git stash pop »

LE REMISAGE


A’
0 Son fichier est A’’
1 Son fichier est A
2 Son fichier est A’
3 Son fichier est de nouveau A’’
2 Fetch
Git local (Jérémy)
Git distant (commun à tous)
Git local (Anaïs)
1 Stash
3 Stash pop
A’
A’’

EXERCICE

Créer un compte gratuit sur Github

Créer un premier projet (git init)

Télécharger et installer GitKraken

Lier GitKraken et Github (avec une clé SSH !)

Cloner le projet de Github (git clone)

Tester de créer un fichier et de le placer sur le dépôt local

- Commit (avec commentaires !)
- Push

FONCTIONNEMENT – COMMANDES

Action	Description
init	Initialiser un nouveau dépôt (ou nouveau projet)
clone	Récupérer un dépôt existant, et en créer une copie locale
fetch	Récupérer les documents du git distant
merge	Assembler des parties
pull	Les commandes **fetch** et **merge** sont exécutées
commit	Versionne dans le git local
push	Envoie les commits locaux vers le git distant
revert	Revenir à une ancienne version
diff	Différence entre deux versions
branche	Créer une nouvelle branche
checkout	Naviguer de branche en branche
ignore	Ignorer des fichiers ou des répertoires lors des synchronisations (fichier .gitignore)

FONCTIONNEMENT – REVISIONS

Un numéro de version du dépôt

- HEAD – La plus récente
- BASE – La version locale

FONCTIONNEMENT – .GITIGNORE

Fichier texte qui s'appelle « .gitignore » (avec le point devant)

Fichier qui permet d'ignorer des fichiers et/ou des répertoires

- A placer généralement à la racine du projet
- Peut également être dans des sous-répertoires
  - (sont ignorés les fichiers et répertoires à partir de l'emplacement de .gitignore)

FONCTIONNEMENT – BRANCHES

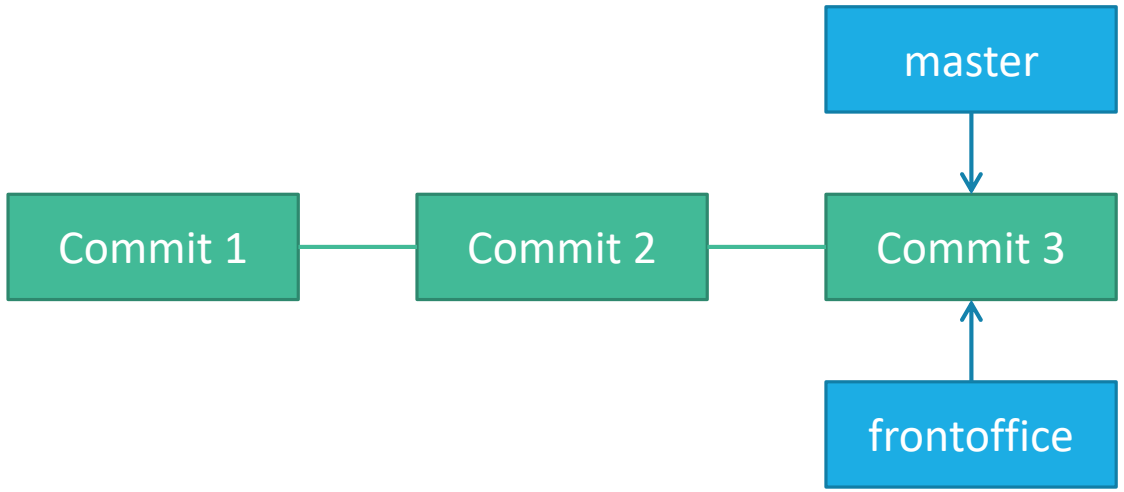
Pointeur sur un commit

Permet d'isoler des modifications de la branche principale, **master**

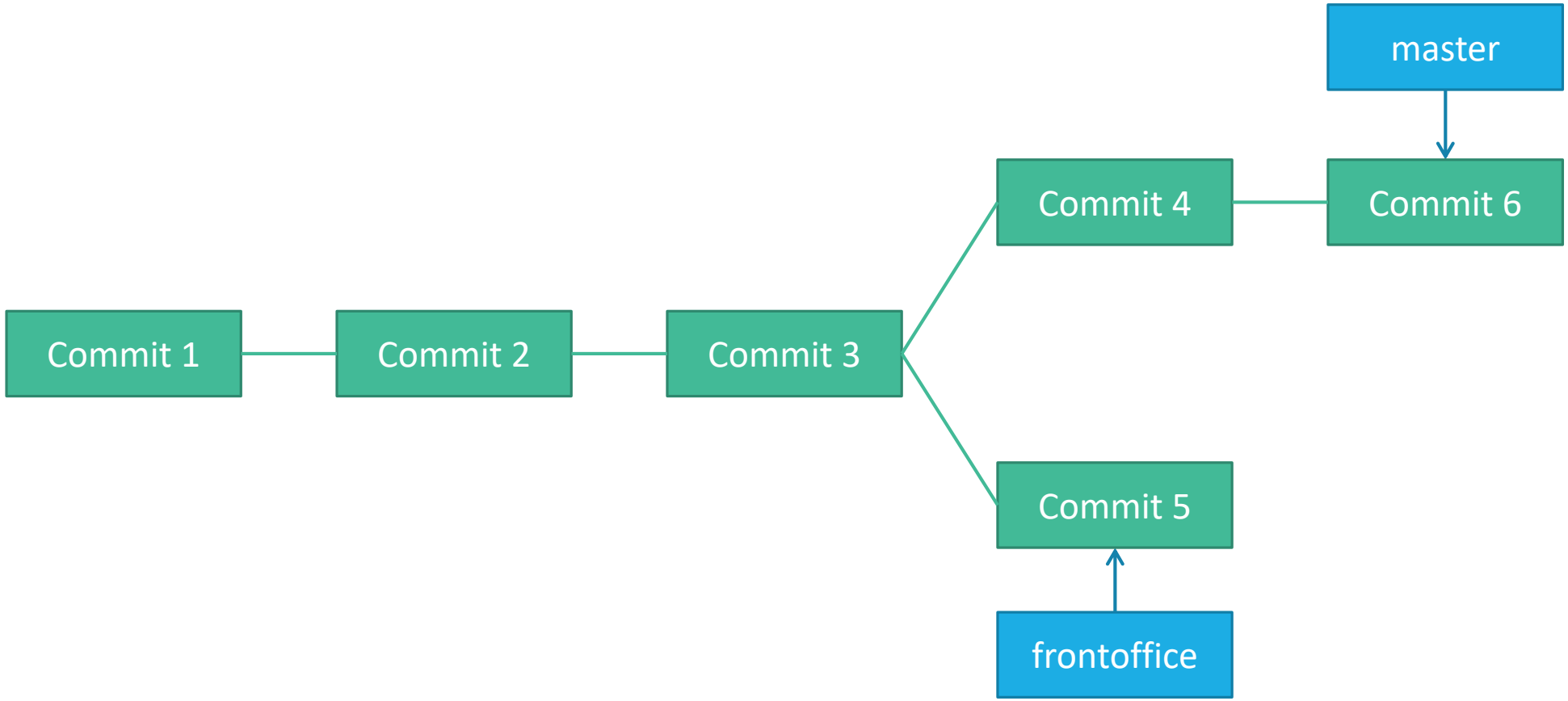
- Corriger un bug
- Ajouter une fonctionnalité bien particulière
  - Exemple : équipe Back-Office, équipe Front-Office
- …

FONCTIONNEMENT – BRANCHES

FONCTIONNEMENT – BRANCHES

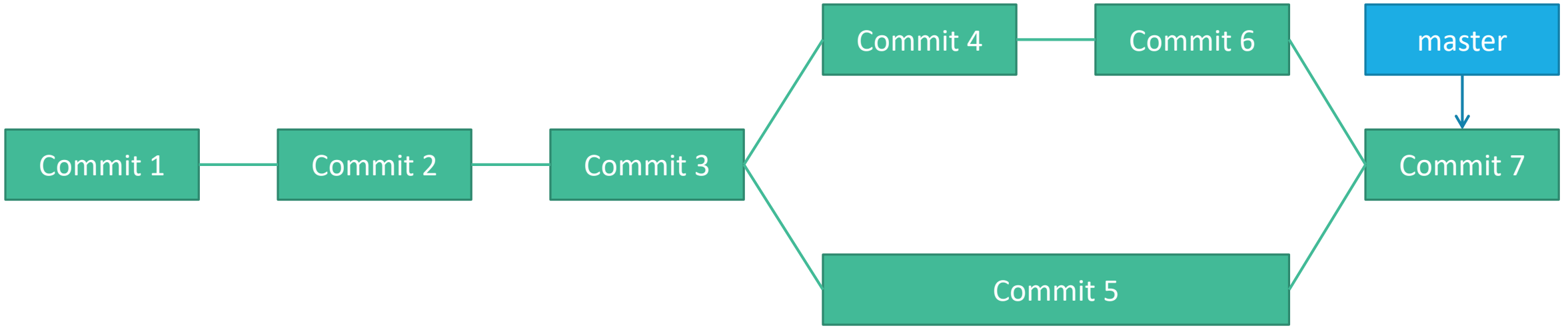


FONCTIONNEMENT – BRANCHES



FONCTIONNEMENT – BRANCHES

Pour fusionner deux branches entre-elles, on applique un « merge »



EXERCICE

Choisir un maître

- Il devra créer un nouveau dépôt

Les autres devront travailler sur ce dépôt