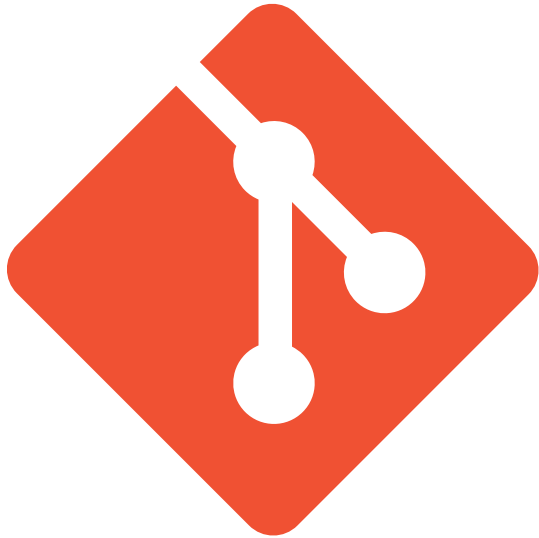
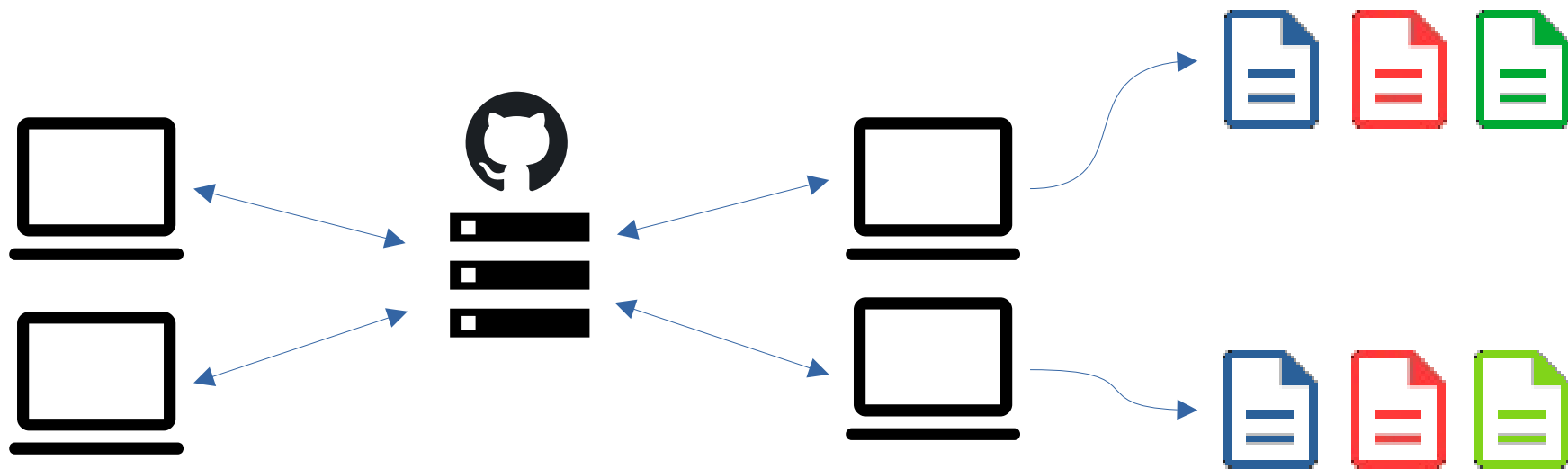


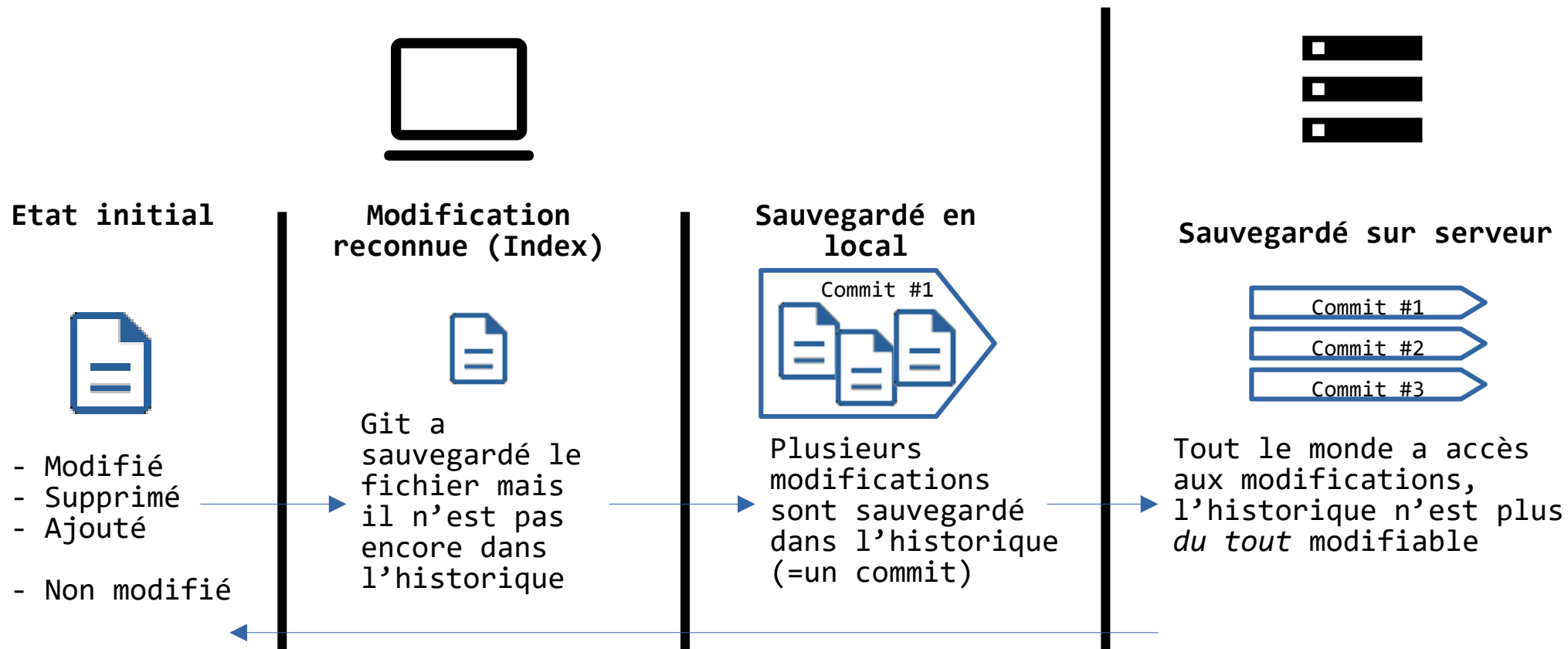
Git



Git - Comment ça marche



Git - Fichiers vs Commit



Git – Commandes

git clone <url>

clonner un repository depuis github/gitlab...

git status

à utiliser tout le temps

git add <fichier1> <fichier2>...

enregistrer la modification d'un fichier

git add .

enregistrer toutes les modifications

git commit -m "<nom du commit>"

créer un commit avec toutes les modifications enregistrées

git pull

fusionner les commits du serveur avec vos fichiers

git push

envoyer vos commits vers le serveur

Kit de survie pour la shell

pwd

affiche votre position dans vos fichiers

ls

liste les fichiers/dossiers là où vous êtes

cd <dossier>

se déplacer dans un dossier

- . le dossier courant
- .. le dossier parent
- ~ votre dossier utilisateur (/users/Albin)

1. Clonner le repository à TODO
2. Ajouter un fichier à votre nom
3. Commit et push
4. Pull et signez le fichier d'un autre
5. Commit et push

Attention aux merge conflicts !

Git – Extra



.gitignore

Permet d'exclure des fichiers de la synchronisation

à faire absolument pour les fichiers de configuration d'IDE !



Les branches

Pour ne pas se marcher dessus, chacun travaille sur sa branche et la branche principale ne contient *que* le code fonctionnel

Il peut aussi y avoir des branches de tests/développement alternatif etc



Les stashes

Permet de stocker des modifications temporairement, pour les déplacer d'une branche à une autre ou éviter un merge par exemple



Les merge requests

Fonction de github (pas de git)

Permet de discuter de l'intégration de deux branches avant de les fusionner

git branch

Liste les branches

git branch <nom>

Créer une branche

git checkout <nom>

Créer de se déplacer vers une branche

git merge <nom>

Fusionne la branche <nom> et la branche sur laquelle vous êtes

git stash

Créer un stash (à partir de tout ce qui n'est pas commit)

git stash pop

Retire et applique un stash