XSTRA-DEV

Le 17/06/2016

Contrôle de source

Subversion vs Git

Sylvain Théry

thery@unistra.fr

ICube

But

- Exposer et expliquer les concepts
- Avantages & inconvénients
- Pas de liste de commandes compliquées

➔ Vous donner l'envie d'utiliser un VCS

➔ Vous donner l'envie d'utiliser **git**

Qu'est-ce qu'un VCS

- Permet de stocker un ensemble de fichiers (source) en conservant la chronologie des modifications.
- Facilite le travail collaboratif sur un projet commun.
- Peut être employé sur tout type de fichiers textes, pas spécifique à un langage.

Terminologie

- Version Control System
- Centralisé / Distribué
- Dépôt (repository)
- Copie locale
- Commit
- Révision

Pourquoi (solo)

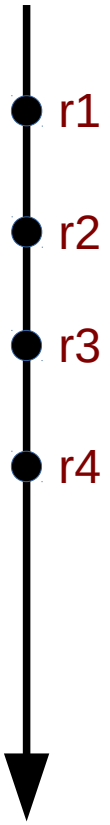
- Sauvegarde des fichiers sources
- Garder l'historique des modifications
- Synchroniser son travail sur plusieurs machines
- Structurer son développement

Pourquoi (équipe)

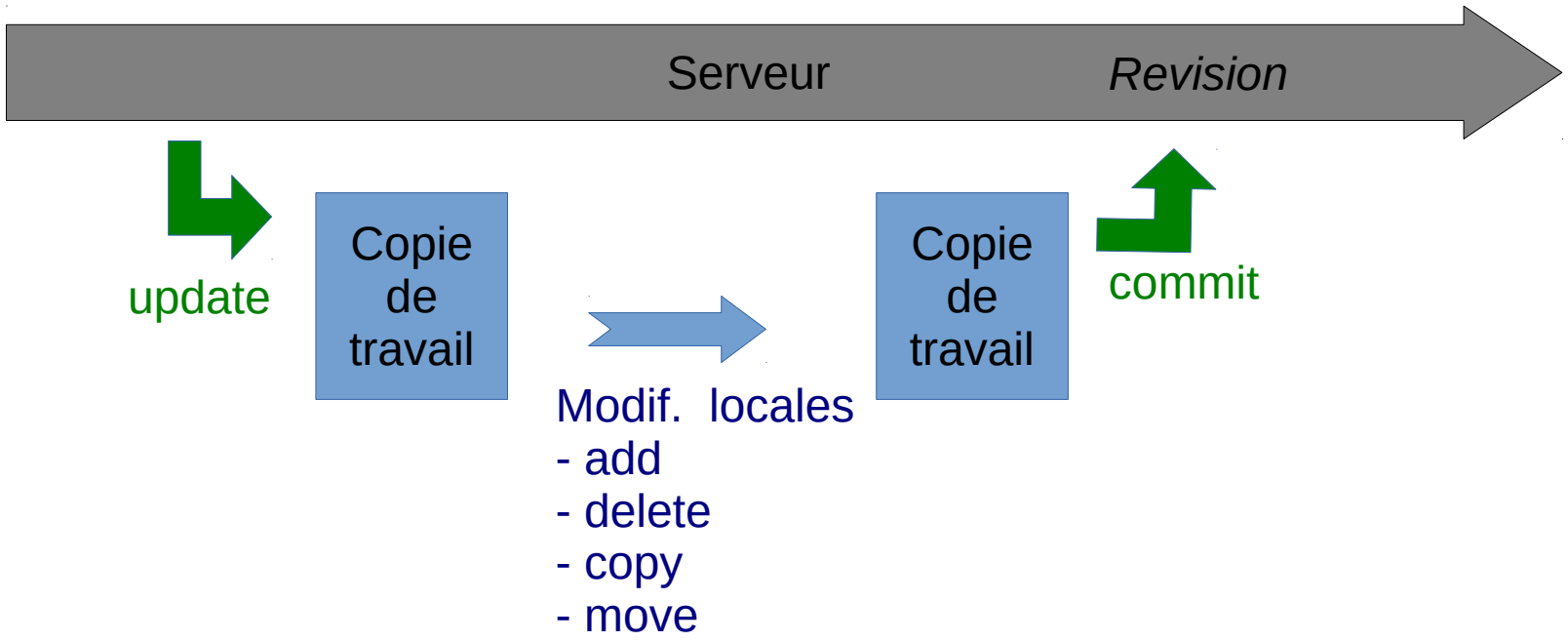
- Partager avec d'autres développeurs
- Fusion *facile* des développements
- Eviter / gérer les conflits
- Structurer le travail d'équipe
- Premier pas vers l'intégration continue

Centralized VCS (svn)

- Dépôt hébergé par un serveur (http/ssh)
- Simplicité d'utilisation
- Historique linéaire
- Nécessite le réseau
- Branches/Etiquettes complexes (copies)
- *rcs / cvs / **svn***



Flux C-VCS typique

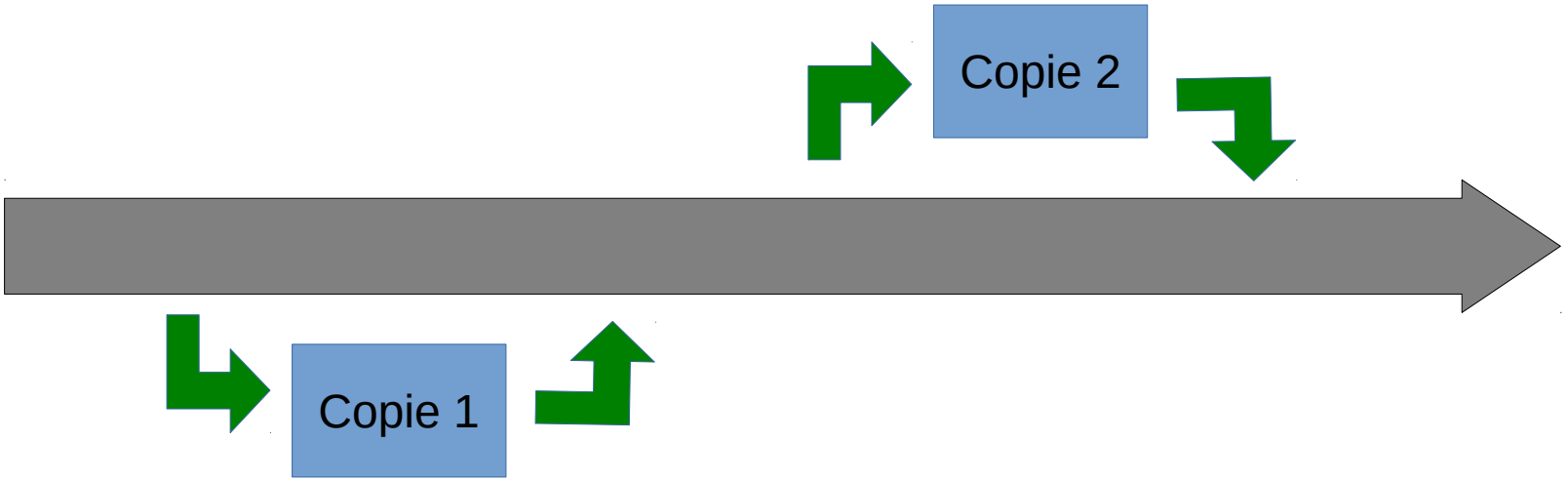


Il faut faire des commits régulièrement sinon l'historique perd son utilité !

Mais pas trop pour que ça reste pertinent !

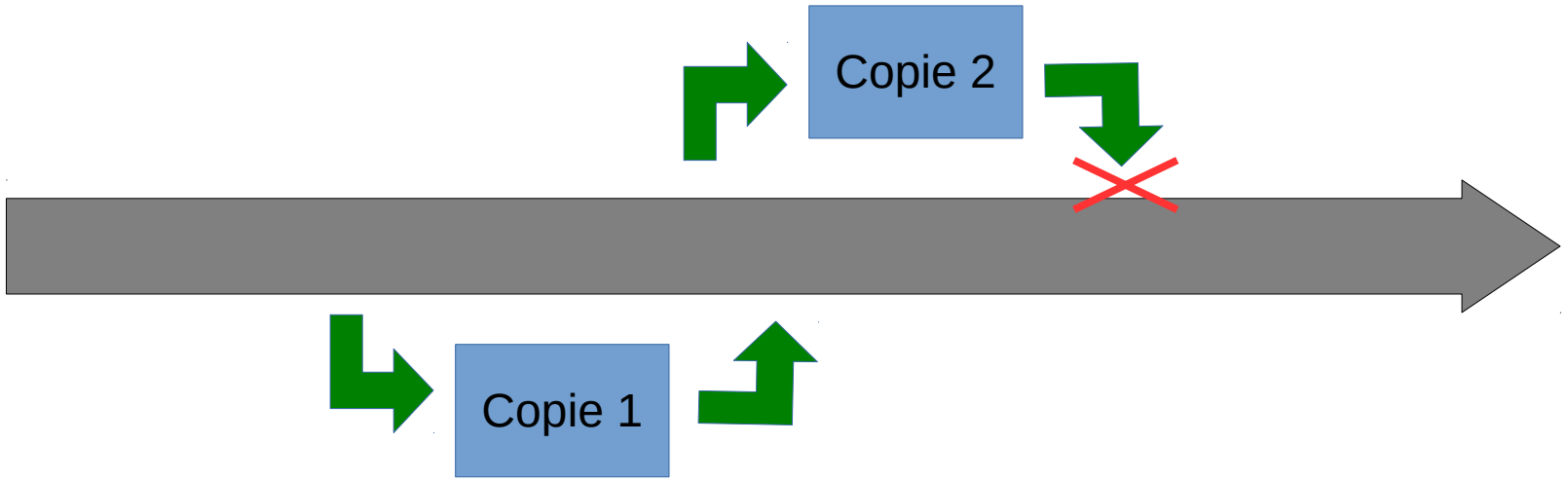
Flux C-VCS typique

Plusieurs copies locales : plusieurs utilisateurs, plusieurs machines



Flux C-VCS typique

Attention à la syncro !

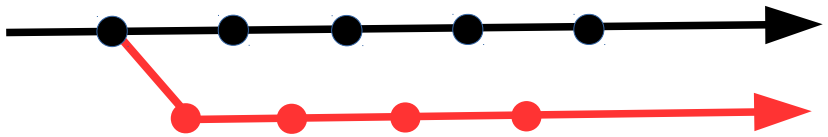


Il faut mettre à jour avant d'envoyer ses modifications
La résolution des conflits se fait à l'update pas au commit

Branches / Etiquettes

Branche :

- Isole le développement d'une fonctionnalité
- Isole le travail d'un développeur

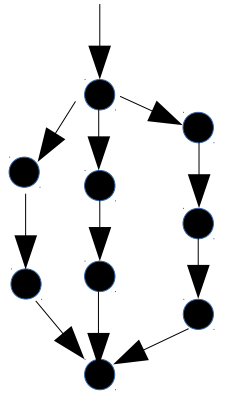


Etiquette (Tag) :

- Désigne une version spécifique (1.3)
- Associe un id plus simple que la révision

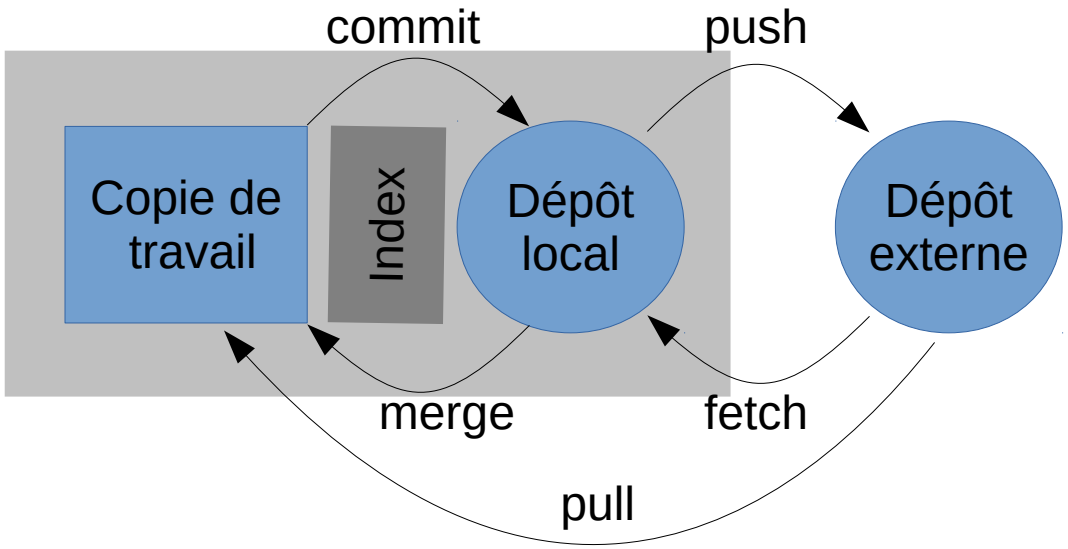
D-CVS (git)

- Plusieurs dépôts possibles pour un projet
- L'historique est un graphe
- Travail local temporaire hors-ligne
- Branches et tags plus simples à utiliser
- *baazar / mercurial / **git***



Git

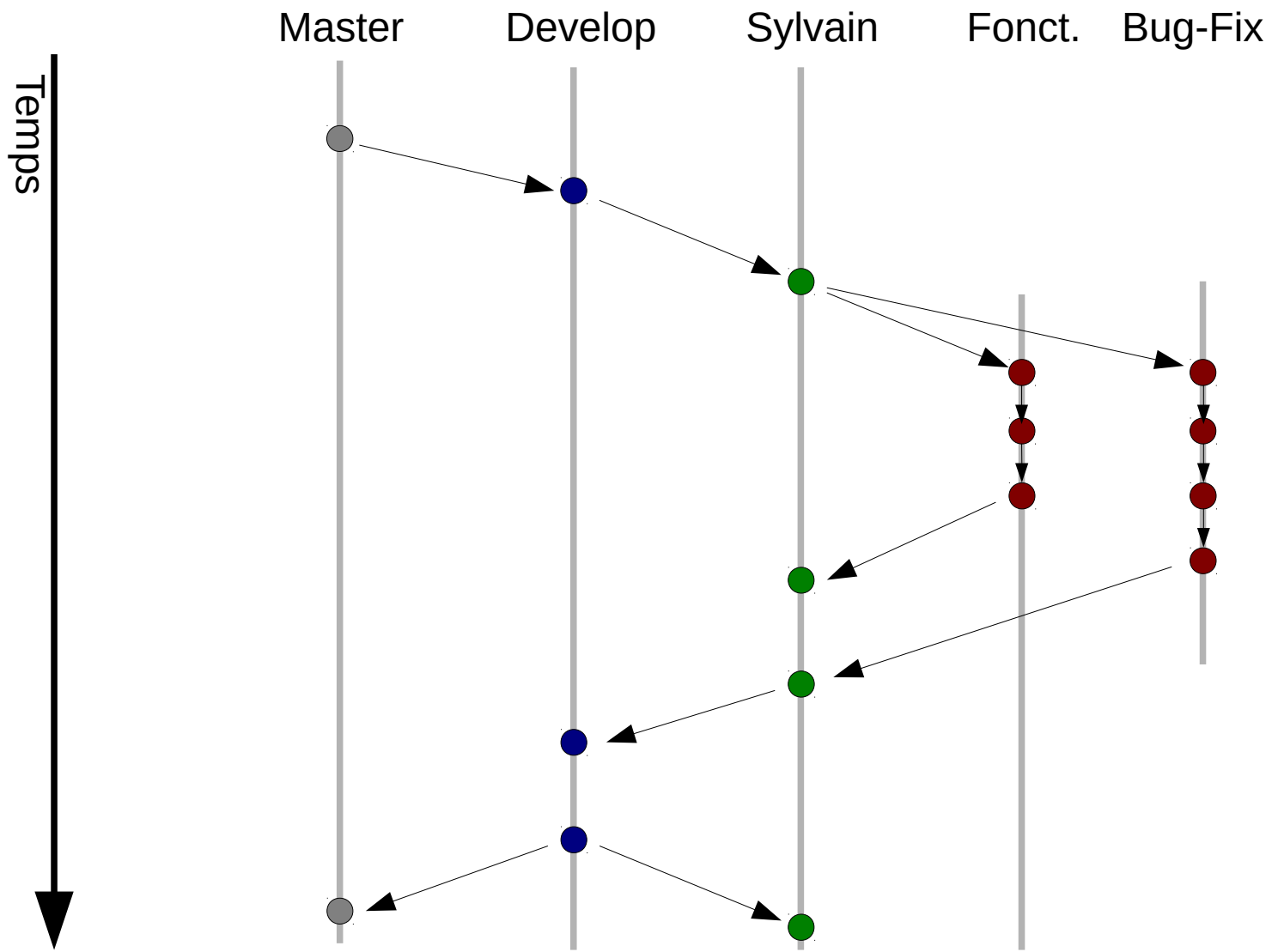
- Dépôts externes (distants)
- Dépôt local
- Copie de travail



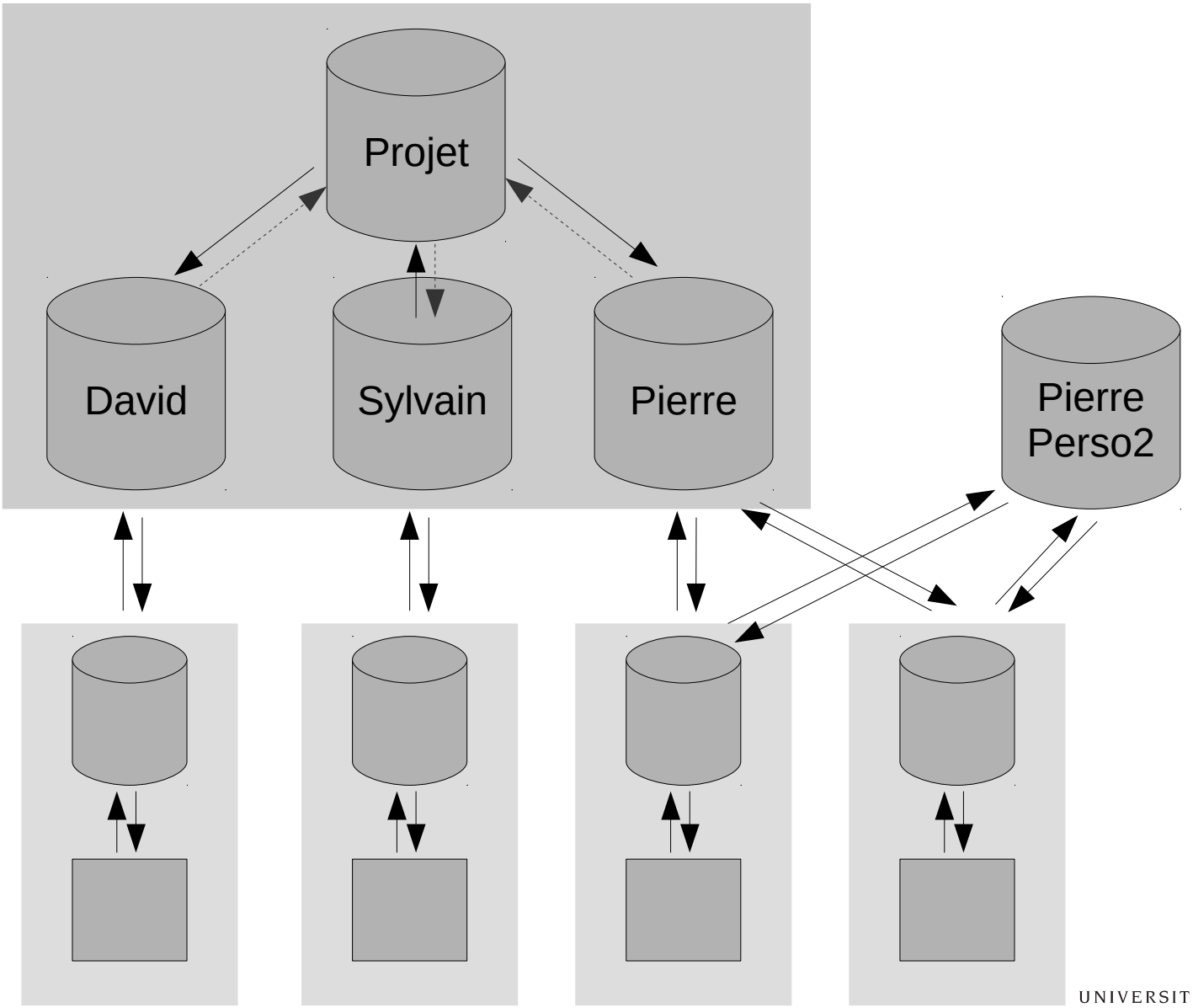
Organisation d'un projet

- Git permet de faire n'importe quoi !
- Il faut choisir une stratégie, une organisation
- Taille de l'équipe / Taille du projet → choix

Organisation d'un projet git

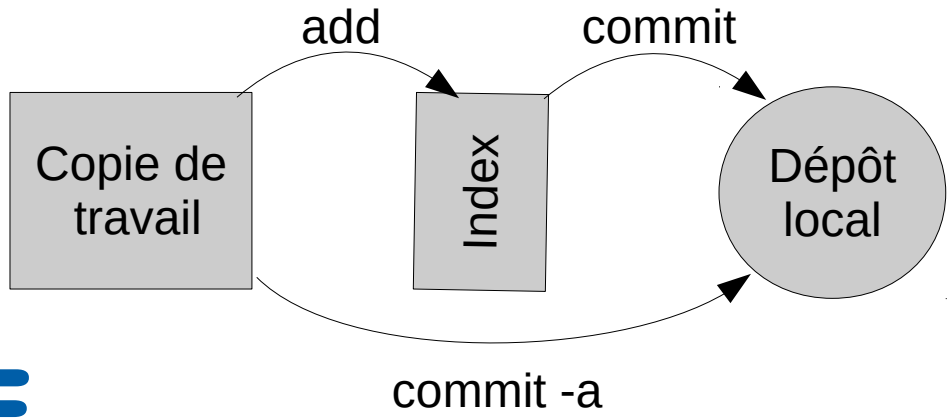


Organisation d'un projet git



Git Staging Area

- Zone de transit (index) des modifications
- On peut y ajouter des modifications :
  - de fichiers (*git add toto.cpp*)
  - de parties de fichier (*git add -p*)
- Pour les envoyer ensuite sur le dépôt (*git commit*)
- On peut la rendre transparente (*git commit -a*)



Git Historique

- On a parfois un historique brouillon !
- git permet la manipulation de l'historique
- à faire sur le dépôt local (avant le push)
  - ➔ Suppression de commits
  - ➔ Fusion de commits
  - ➔ Réordonnancement
- cherry-picking / rebase

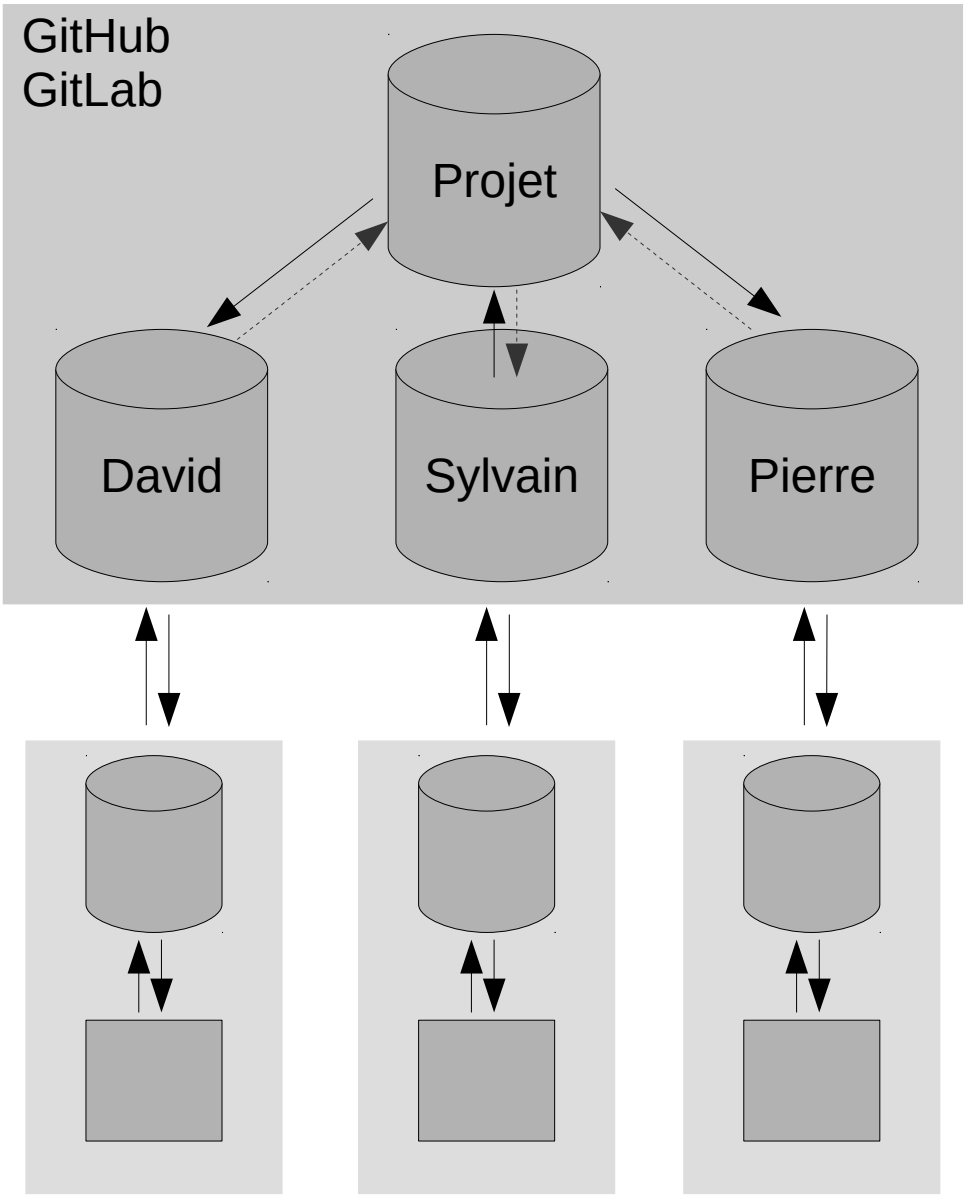
Git Défauts

- Permet de faire beaucoup (trop) de choses
- Grande liberté (trop)
- Ligne de commande inévitable
- Une syntaxe parfois barbare :
    - git checkout -- toto.cpp
- Les id de commit sont des hash SHA-1
- Mais les messages sont très pertinents

GitHub / GitLab

- Hébergement / Interface Web du dépôt distant
- Un administrateur pour le dépôt central
- Chaque utilisateur a son dépôt perso (fork)
- Système de pull-request (merge-request)
  - L'utilisateur demande l'intégration des modifications de sa branche
  - L'admin accepte (ou refuse) après discussions / modifications

GitHub / GitLab



CONCLUSION

- Utilisez un VCS
- Investissez-vous et utilisez git
- Choisissez une organisation et respectez la
- Faîtes un maximum de branches
- Pour le libre, utilisez GitHub, c'est gratuit
- Pour les projets privés utilisez GitLab

Références

http://svnbook.red-bean.com/

https://git-scm.com/

https://www.atlassian.com/git/

http://nvie.com/posts/a-successful-git-branching-model/

http://rogerdudler.github.io/git-guide/index.fr.html

https://marklodato.github.io/visual-git-guide/index-fr.html

CONCLUSION

à mettre dans tous les bureaux / couloirs: