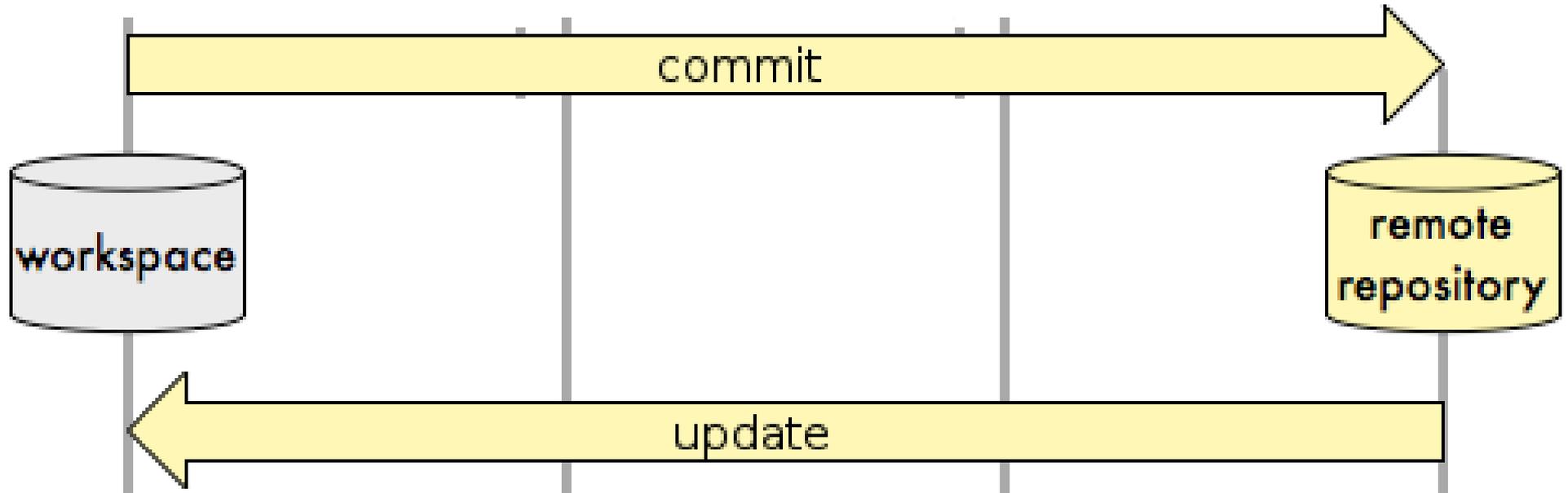


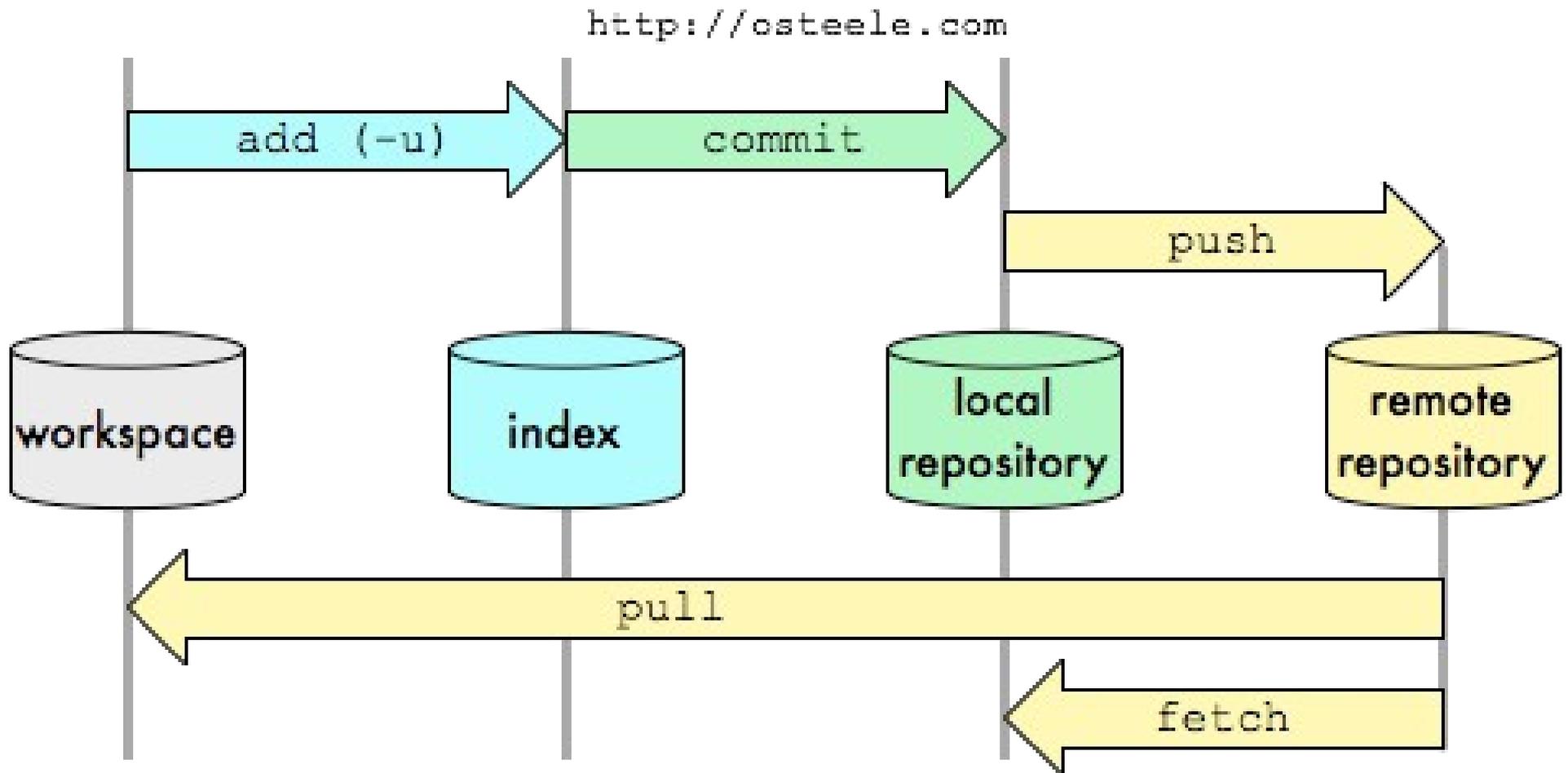
Bref aperçu de Git pour un utilisateur de SVN

Pascal Rigaux
pascal.rigaux@univ-paris1.fr

SVN workflow



Git workflow



Index ou « Aire d'assemblage »

- Permet une sélection fine des modifications à commiter :

```
git add -p
```

- Spécificité de git
 - non dispo en SVN/Mercurial

Dépôt local

- répertoire `.git`
 - à la racine (ne pollue pas tous les répertoires)
 - contient tous les commits
 - contient la configuration du projet
- à part `push/pull` tout se fait en `local` :
 - manipulation des commits et des branches
 - consultation de l'historique (`log`, `blame`)
- dépôt techniquement équivalent au `remote repository`

Branches

Subject	Author
 esup-4.0.3 configuration de test pour univ-paris1.fr	Pascal Rigaux
 origin/HEAD origin/esup-4.0.3 local.properties: valeurs par défaut plus adaptées aux universités	Pascal Rigaux
 portal.properties: use fr_FR by default	Pascal Rigaux
 cas logout: redirect to portal in guest mode instead of staying on cas logout screen	Pascal Rigaux
 create conf variable <code>\${environment.build.cas.context}</code>	Pascal Rigaux
 use mysql instead of hsqldb	Pascal Rigaux
 uportal-4.0.3 [maven-release-plugin] prepare release uportal-4.0.3	Eric Dalquist
 Update license headers for release	Eric Dalquist
 Update NOTICE for release	Eric Dalquist
 Fix JAXB Spec version, fix io xsd version specification	Eric Dalquist
 Simplify JAXB Plugin configuration	Eric Dalquist
 Merge remote-tracking branch 'upstream/master'	Eric Dalquist
 Update dependencies for release	Eric Dalquist
 Merge branch 'UP-3308'	Jen Bourey

Branches

- Les commits Git :
 - sont nommés par leur SHA (pas de numéros de révision)
 - sont traités comme des branches
- Facile de créer des branches
 - notamment des branches temporaires
- Facile de travailler entre branches
 - s'échanger des commits
 - mettre à jour une branche par rapport à une autre branche
 - merge conserve l'historique des 2 branches
- L'équivalent de trunk est master

Divers

- **Configuration** (ne pas oublier de le faire!)

```
git config --global user.name "Your Name"
```

```
git config --global user.email xxx@univ-zzz.fr
```

- `git config --global color.ui auto`

- **Convention pour les messages de commit :**

- ligne 1 : résumé des modifications

- ligne 2 : rien

- lignes suivantes : le détail des modifications

- `git push` utilise `ssh` pour l'authentification

Commandes utiles

<code>git add -p</code>	ajouter finement des modifications à commiter
<code>git status</code>	voir l'état des fichiers modifiés et des fichiers non ajoutés dans git
<code>git log</code>	voir les commits
<code>git reset HEAD^</code>	annuler le dernier commit (sans perdre les modifications de ce commit)
<code>git commit --amend</code>	modifier le dernier commit (notamment le message)
<code>git diff</code>	voir les modifications non commitées
<code>git checkout <filename></code>	revenir à la dernière version commitée d'un fichier
<code>git grep</code>	chercher dans les fichiers

Liens utiles

- Si vous connaissez les commandes svn :
<https://git.wiki.kernel.org/articles/g/i/t/GitSvnCrashC>
- Pour débiter en git sur sourcesup :
<https://services.renater.fr/sourcesup/git/index>
- Sinon, énormément de documentation disponible sur <http://git-scm.com/documentation> , notamment <http://progit.org/book/fr/>