Subversion alias SVN

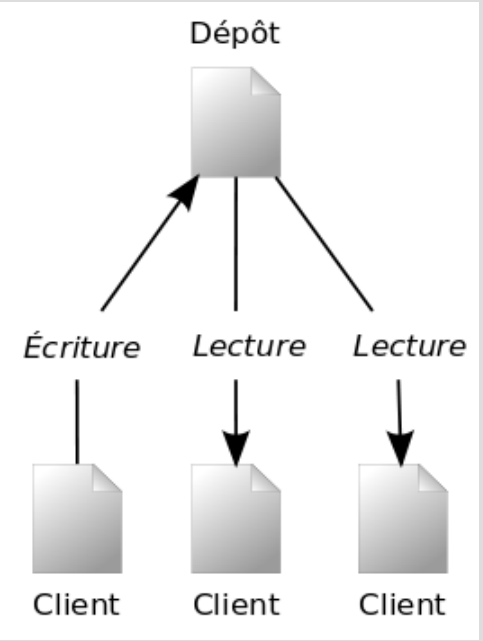
- Site officiel: http://subversion.apache.org/
- Documentation: http://svnbook.red-bean.com/

Pourquoi ?

- Garder l'historique de l'évolution d'un projet.
  - Retrouver les versions antérieures de fichiers
  - Suivre les modifications
  - Coordonner le travail d'une équipe
  - Faire une photographie (release, tag) du projet
- Pas de code (conf pour les admins) sans gestionnaire de version
  - Adieu les .old, .sav, .new, .bkp, .au_cas_ou

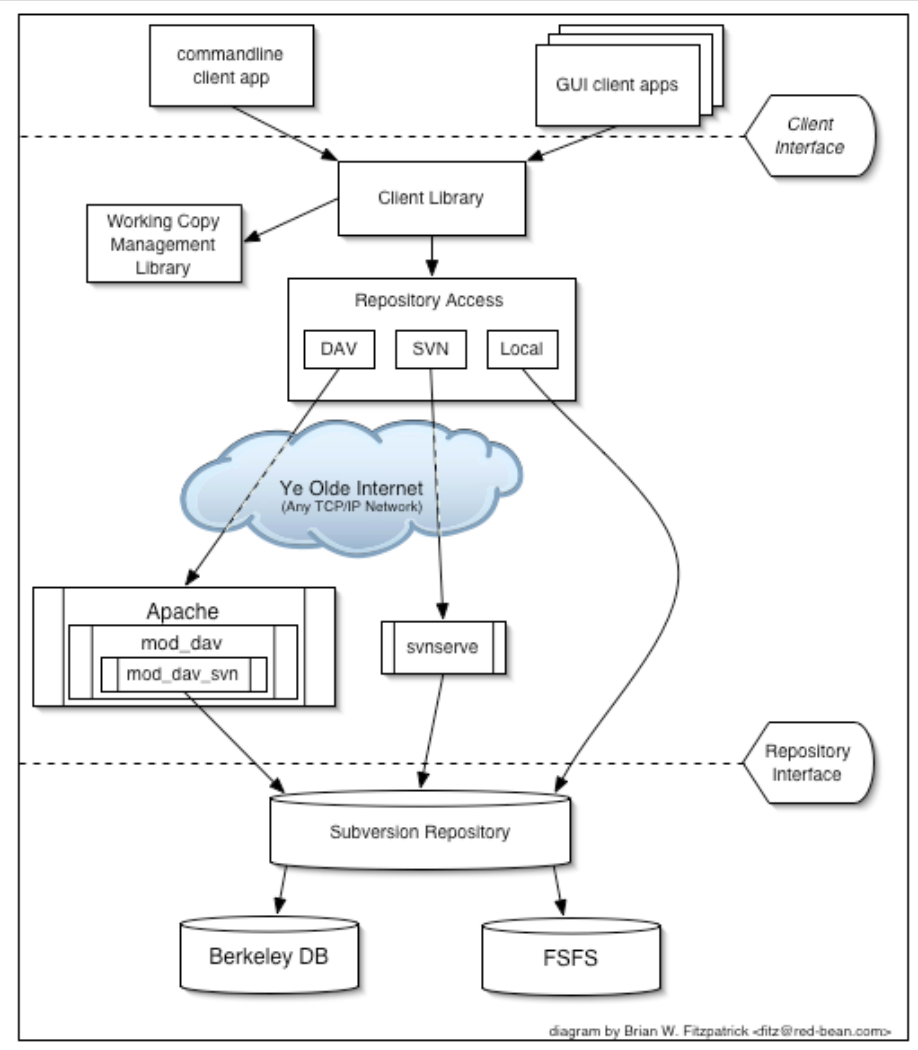
Concepts de base

- Dépôt (repository), plutôt coté serveur
- Copie de travail (working copy), coté client
- Architecture Client/Serveur



Architecture de SVN

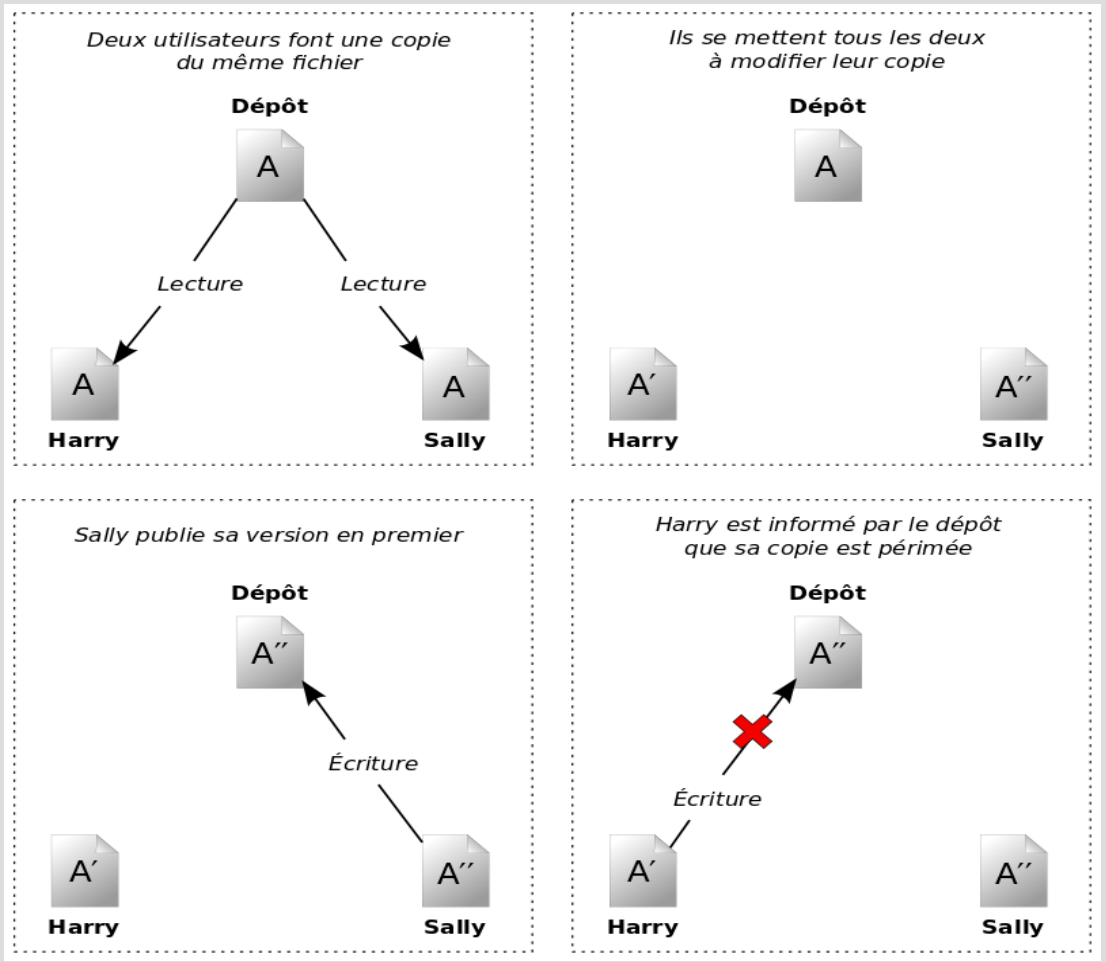
- DAV (http:)
- SVN (svn: ou svn+ssh:)
- Locale (file:)



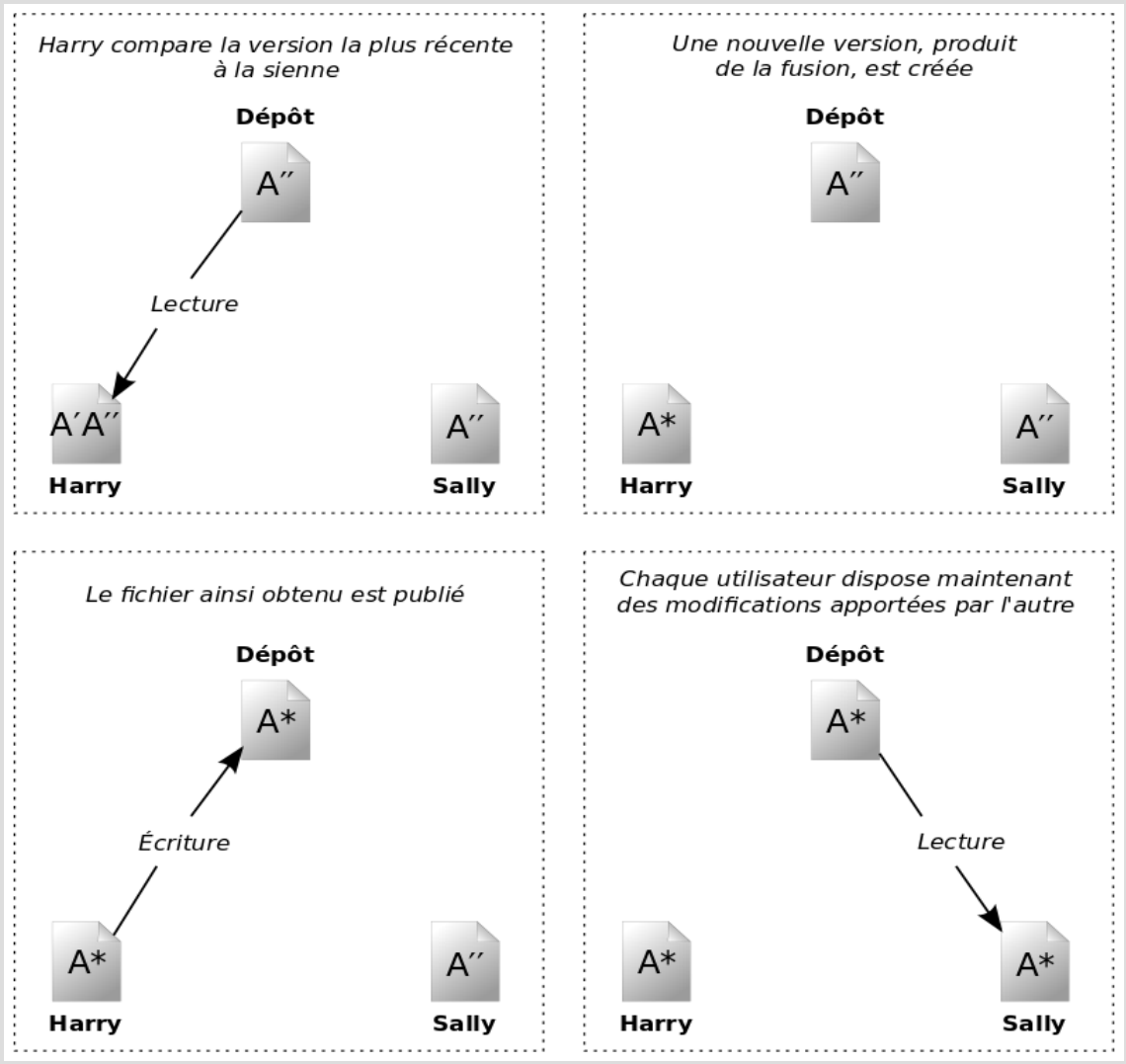
Modèle de gestion de versions

- Tous les logiciels de gestion de versions doivent résoudre le même problème fondamental : comment le logiciel va-t-il permettre aux utilisateurs de partager l'information, tout en les empêchant de se marcher mutuellement sur les pieds par accident ?

SVN: modèle copier-modifier-fusionner



SVN: modèle copier-modifier-fusionner



Création du dépôt

- Commande côté Serveur:

  cd mon_path; svnadmin create mon_proj

- Attention au choix du type de dépôt ('fsfs' ou 'bdb').

cas particulier: Import d'un projet existant

- Ajouter une arborescence existante au dépôt
  - Via SVN+SSH

    svn import mon_arbo
    svn+ssh://login@mon_serveur/path/mon_proj -m
    'Depot initial de mon arbo'
  - Via HTTPS

    svn import mon_arbo
    https://login@mon_serveur/path/mon_proj -m
    'Depot initial de mon arbo'

Initialisation de la copie de travail

- svn checkout https://login@mon_serveur/path/mon_proj
- cd mon_proj
- Organisation conseillée

  svn mkdir trunk tags branches
- svn commit -m 'création de la structure'

Cycle de travail

- Je mets à jour (svn update)
- Je modifie mes fichiers
- Examiner les changements apportés (svn status, svn diff, svn revert)
- Je valide ma modification: Pensez aux autres !!
  - Compilation et tests OK
  - Service fonctionnel (Web, Mail, ...)
  - Attention aux données sensibles
- Je livre sur le dépôt (svn commit)

Commandes

- De base
  - checkout
  - update
  - add
  - mkdir
  - move
  - delete
  - commit
- Particulières
  - import
  - status
  - list
  - resolved
  - revert
  - log
  - copy
  - merge

Conflits

- Tout conflit doit être résolu pour pouvoir commiter
- Comportement SVN en cas de fichier texte
  - Intégration des conflits au fichier courant

    Fragments de différentes versions délimitées par <<< et >>>
  - Génération de fichiers supplémentaires (suffixe .mine, .r##)
  - Résolution à la main puis svn resolved

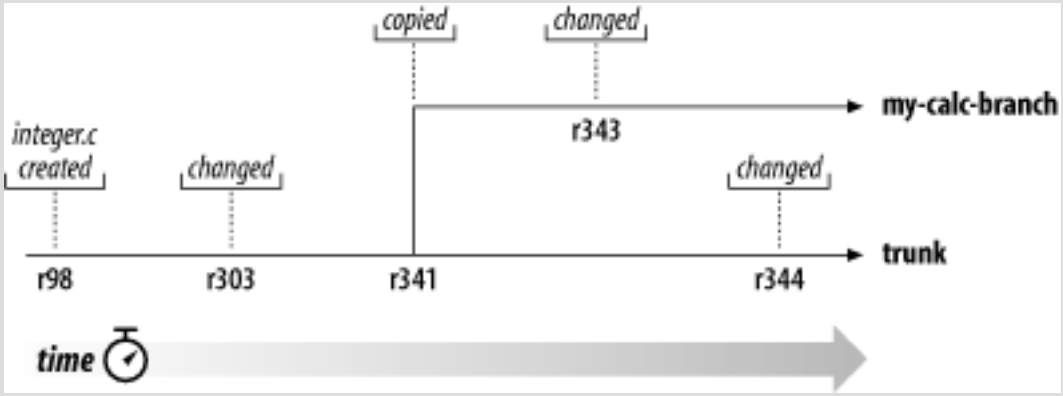
Révisions

- A chaque commit, le numéro de révision est incrémenté.
- Le numéro de révision est global au dépôt
- Quand on veut nommer une révision particulière (release, ...)

```
svn copy trunk tags/version_X_Y_Z
```

Branches

- Elle permet de faire des développements en parallèle à la ligne principale.



- Création:

  svn copy trunk branches/ma_branche

- Fusion dans trunk:

  svn merge -r 341:343 ../branches/ma_branche

Un autre gestionnaire: Git

- Faire des commits sans avoir de connexion au serveur
- Gestion décentralisée