



**Félix-Antoine Bourbonnais**

$\int \int_D \frac{\partial \phi}{\partial x} \frac{\partial P}{\partial y} dx dy$   
fbourbonnais@rubico.info

12 janvier 2004

Groupe LinuQ,  
Québec



1

**Subversion: le CVS de demain!**



Subversion, le CVS de demain!

Première partie

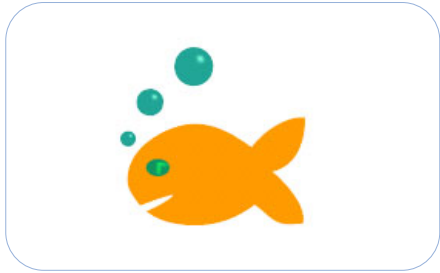
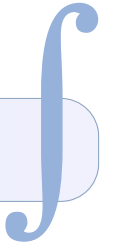
Le travail collaboratif

- δ Travail concurrent : Plusieurs personnes travaillent sur une même ressource (code ou document) dans un intervalle de temps commun.
  
- δ Travail collaboratif : Se dit de ce qui, dans un environnement informatisé ou en ligne, vise à favoriser la collaboration entre pairs, en permettant d'échanger et de partager des compétences pour mieux réussir un projet commun. [GDT]

- Les équipes de travail deviennent de plus en plus grandes.
  - Difficile de fusionner et intégrer manuellement toutes les modifications.
  - Les modifications concurrentes sur une même partie sont fréquentes
- Le personnel et les membres sont dispersés géographiquement
  - Impensable de prendre le téléphone pour parler à son collègue
  - Disparité des heures de travail

- La taille des projets peut être énorme
  - Difficile de gérer efficacement tous ces documents ou sources
  - Il est même parfois impossible qu'un seul être humain puisse connaître l'ensemble d'un projet (ex.: Mozilla)
- Les projets à source ouvert (OpenSource)
  - Regroupent de plus en plus de développeurs
  - Sont de plus en plus utilisés ==> augmentation de l'activité

- Permettre à un grand nombre de personnes de travailler :
  - Sur une même ressource (document ou bout de code)
  - En même temps (de manière concurrente)
- Avoir un système qui :
  - Fusionne automatiquement les travaux
  - Permet de revenir en arrière
  - Garde des traces des modifications
  - Permet à chacun de travailler sans se préoccuper des autres  
[==> indépendance des travaux]



CVS

[www.cvshome.org](http://www.cvshome.org)

- Concurrent Versions System
- Créé par Dick Dune en 1984-1986
- Très largement utilisé et répandu
- Gratuit / Source ouvert (OpenSource)



Subversion

[subversion.tigris.org](http://subversion.tigris.org)

- Remplacement à CVS (nouvelle génération)
- CVS + améliorations (principalement compatible)
- Gratuit / Source ouvert (OpenSource)



### Subversion / CVS

#### Saviez-vous que ?

- ∞ Le noyau Linux utilise BitKeeper plutôt que CVS ou SVN.
- ∞ BitKeeper n'est pas un projet libre.
- ∞ Ceci a créé de vives controverses lors de son adoption par Linus Torvalds.

∂ La majorité des projets à code ouvert (OpenSource)

∂ Mozilla

∂ KDE

∂ OpenOffice

∂ ...

*On parle ici de  
centaine de  
développeurs partout  
dans le monde !*

*OpenOffice.org a  
enregistré ~700 000  
« commit » dans  
son CVS entre  
octobre 2001 et  
janvier 2004 ! (1)*

∂ Samba utilise Subversion

∂ Des développeurs (ou entreprises) de logiciels

∂ Des particuliers pour gérer leurs documents



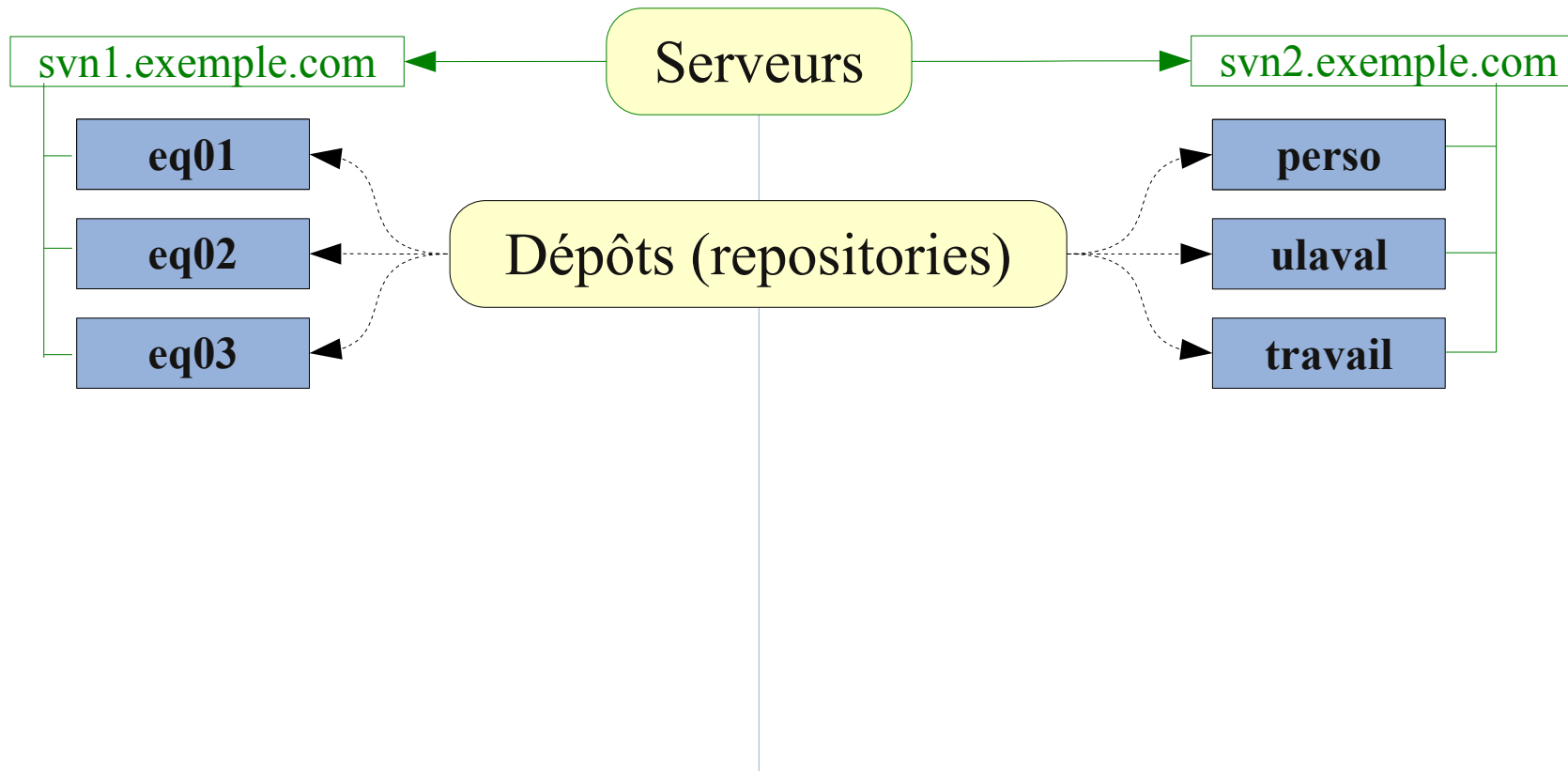
Subversion, le CVS de demain!

Deuxième partie

Le fonctionnement

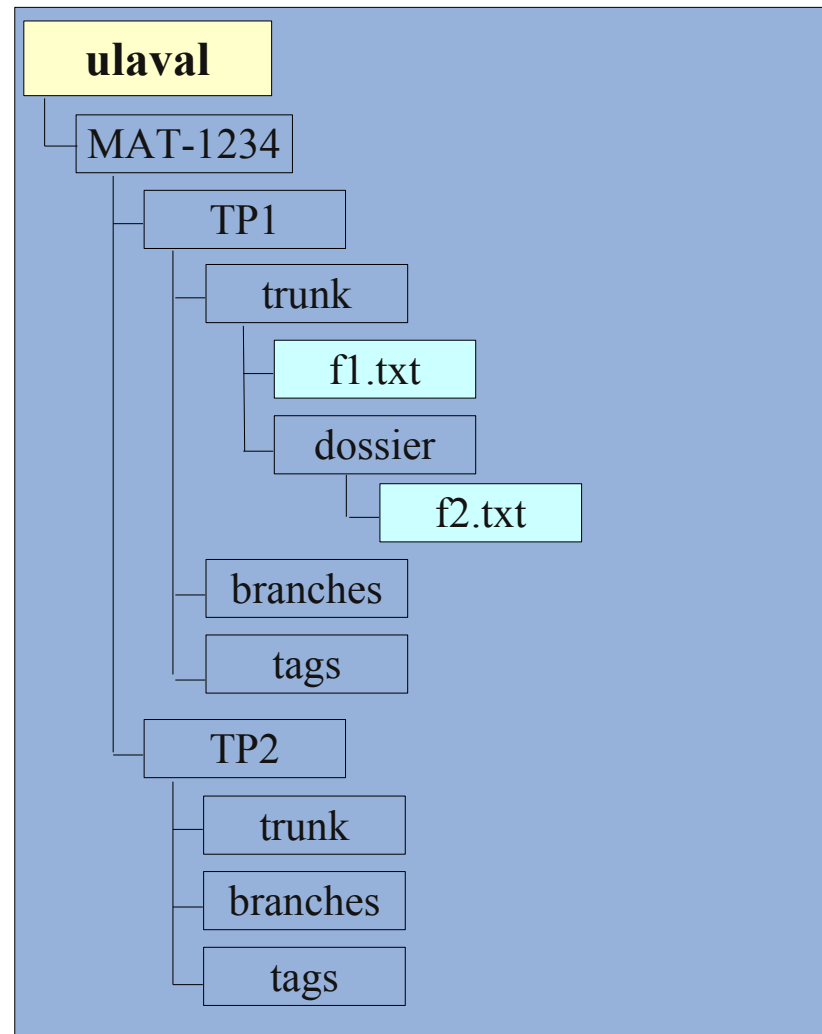
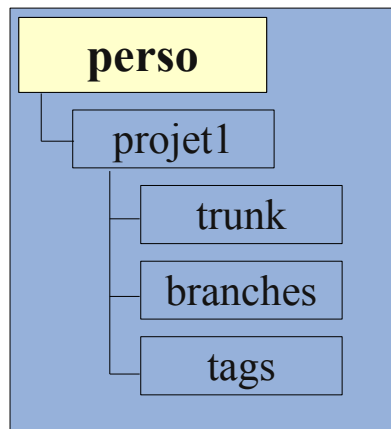
∂ Le dépôt (repository) :

- ∂ Racine du Subversion (dossier parent)
- ∂ Espace de travail dans lequel on place des projets ou sous-projets.
- ∂ Structure qui contient des fichiers, dossiers, et informations.



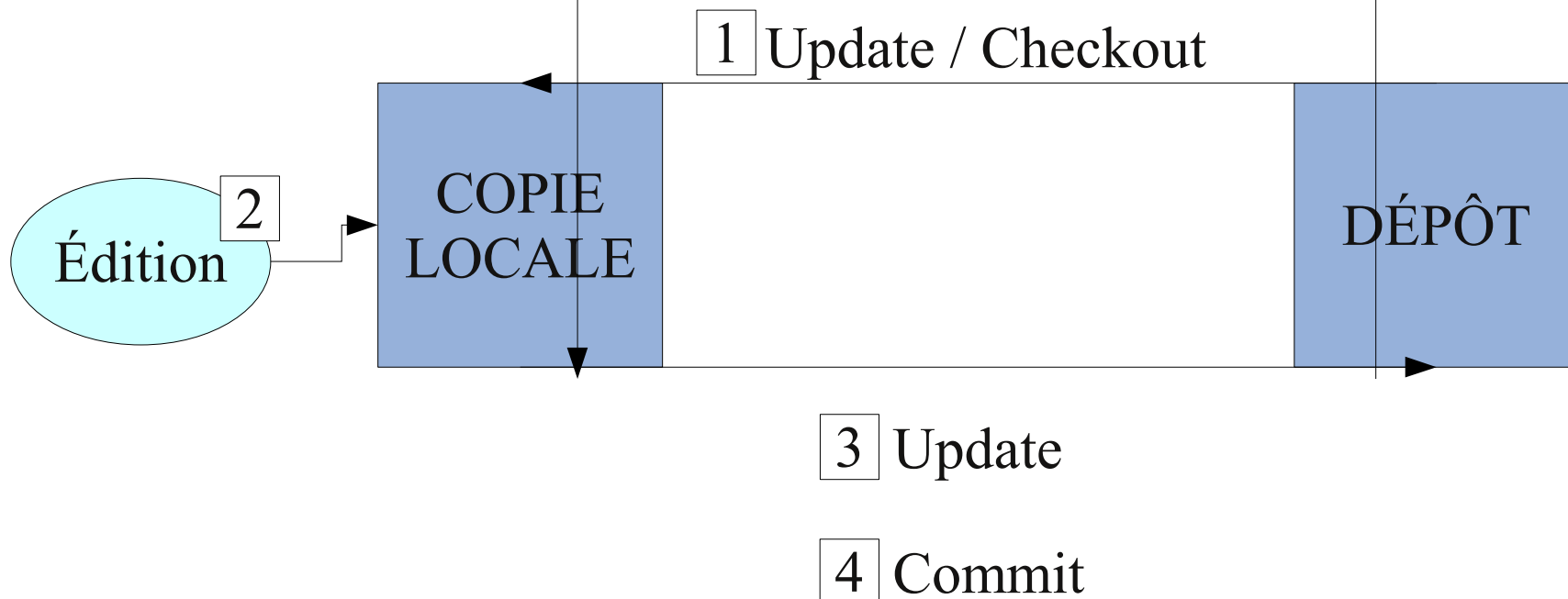
- ∂ Aucune hiérarchie n'est obligatoire pour Subversion (contrairement à CVS)
- ∂ Il y a cependant des formules recommandées:

svn2.exemples.com



► Opérations à effectuer pour utiliser un dépôt existant:

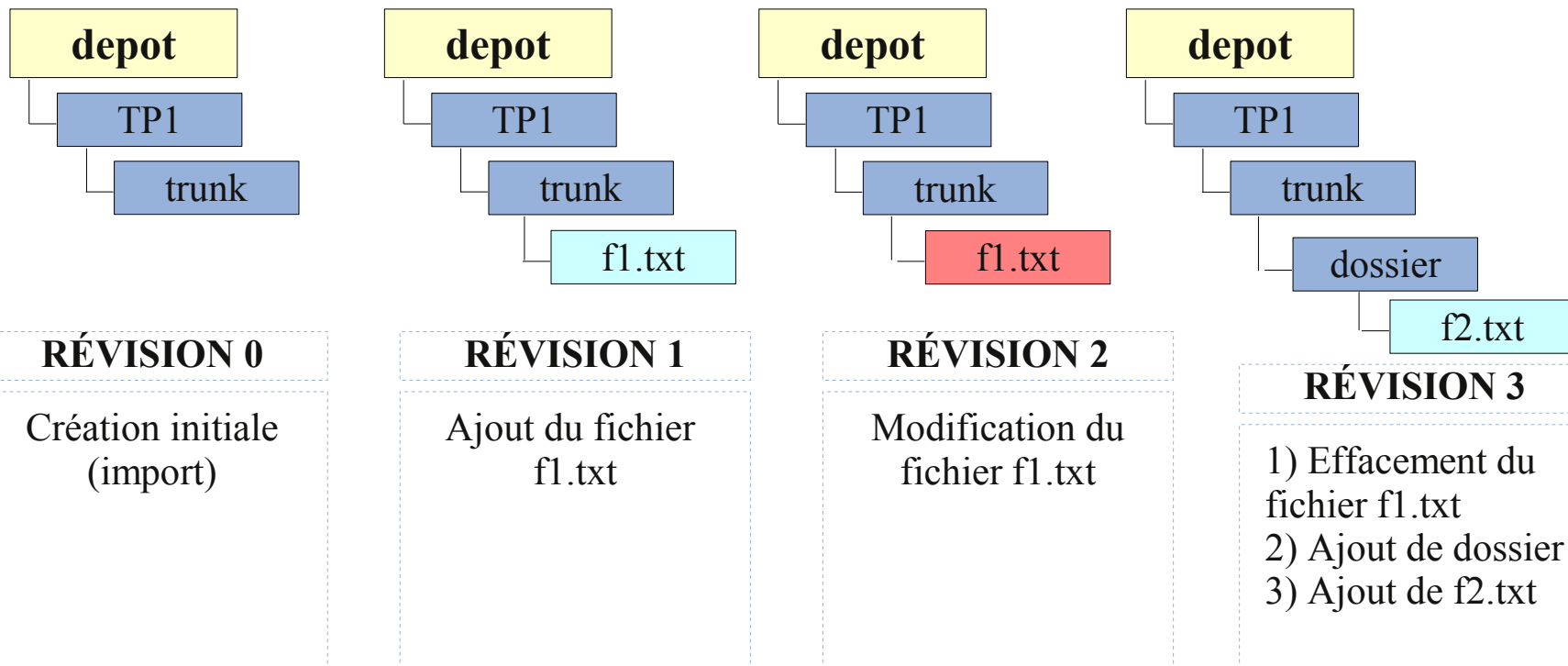
- Σ 1- On télécharge une copie du dépôt [copie locale] 〈 Checkout/Update 〉
- Σ 2- On travaille (lecture et modifications) dans **la copie locale**
- Σ 3- On se synchronise avec le dépôt (s'il y a lieu) 〈 Update 〉
- Σ 4- On envoie les modifications (effectuées sur la copie locale) vers le dépôt Commit 〉



- ∂ Les principales commandes (il existe des alias de ces commandes) :
  - ∋ import : a) Création d'un nouveau module dans le dépôt [CVS]  
b) Envoi (commit) récursif avec création des parents [SVN]
  - ∋ add : Ajouter un élément dans le dépôt
  - ∋ remove: Effacer un fichier du dépôt (les anciennes versions restent!)
  - ∋ checkout: Créer la copie locale (première fois)
  - ∋ update: Mettre à jour la copie locale à partir du dépôt
  - ∋ commit: Mettre à jour le dépôt avec les modifications locales (envoi)
  - ∋ status: État des fichiers dans la copie locale
  - ∋ log: Affiche les messages de chacun des envois (commit) affectant l'élément
  - ∋ diff: Affiche les différences entre deux révisions (ou chemins pour SVN)
  - ∋ move, copy: Déplacer, renommer ou copier un élément [SVN]
  - ∋ revert: Annuler les modifications locales [SVN]

∂ Révisions:

- ∂ Version du dépôt. Comme une image de l'état du dépôt à un moment précis.
- ∂ Propre à tout le dépôt (contrairement à CVS)
- ∂ Une nouvelle révision est créée après chaque commit accepté
- ∂ Versions spéciales: HEAD, BASE, PREV, COMMITTED



Subversion, le CVS de demain!

# Troisième partie

# Subversion (SVN)

## Qu'est-ce que Subversion ?



### Subversion

- ∂ Permet de partager des documents entre plusieurs personnes
  - ∂ Plus besoin d'envoyer les modifications par courriel!
  - ∂ Dépôt unique pour tous les documents et toutes les versions
- ∂ Conserve une copie de toutes les versions du document
  - ∂ Les différences sont conservées par rustines (patches)
  - ∂ Détection automatique des binaires Amélioration!
  - ∂ Possibilité de revenir en arrière tel qu'était le document ou le projet à une certaine date ou à une certaine révision précise.
  - ∂ Permet de consulter les révisions antérieures.



## Qu'est-ce que Subversion ?



### Subversion

- ∂ Fusion automatisée
  - ∂ Tous les membres d'une équipes peuvent travailler sur le projet en même temps et sur les mêmes fichiers.
  - ∂ La fusion est automatique si les modifications ne sont pas en conflit.
  - ∂ S'il y a conflit, Subversion aide à repérer et à résoudre ce dernier
  - ∂ Basé sur le « mime-type » du document **Amélioration!**
- ∂ Permet de renommer/déplacer les fichiers **Amélioration!**
- ∂ Les répertoires sont aussi versionnés **Amélioration!**
- ∂ Possibilité de marquer une étape dans le développement (Tag)
- ∂ Permet de scinder temporairement ou non un projet (Branch)



1 Révision par fichiers

1 Révision commune à tout le dépôt

2 Versionnage des fichiers

2 Versionnage des fichiers, dossiers et métadonnées

3 Système de branches et d'étiquettes (tags)

3 Les branches et étiquettes sont de simples répertoires

4 Impossible d'associer des propriétés aux fichiers.

4 Possibilité d'ajouter des informations (métadonnées) aux fichiers et répertoires (ex.: `svn::ignore`, `svn::mime-type`,...)

5 

5 Gestion différente des conflits (`.mine`, `.rOLD`, `.rNEW`)

6 Opérations sur des modules dans un dépôt

6 Abolition du concept des modules (ils sont de simples dossiers)

7 Hiérarchie obligatoire (modules/projets)

7 Hiérarchie libre

8 Impossible de renommer/déplacer (sans perdre le versionnage)

8 Possibilité de renommer et déplacer des fichiers et/ou des dossiers



9 Authentification généralement basée sur pserver (non-securisé) ou SSH.

9 Authentification par svnserve (svn+ssh) ou directement par les mécanismes offerts par Apache 2

10 Les binaires doivent être précisés manuellement

10 Auto-détection des binaires (ou plutôt des non-text/\*)

11 Pas d'algorithme spécifique pour les fichiers binaires (considérés comme texte).

11 Algorithme pour faire des différentiels de fichiers binaires.



Subversion

- ∂ 1- DAV avec Apache 2 pour la communication client/serveur
  - ☆ L'authentification est gérée par Apache
  - ☆ La sécurité est assurée par SSL via Apache
  - ☆ Toutes les configurations d'apache sont possibles (<Directory>, <Location>, ...)
  - ☆ Possibilité de visualiser la dernière révision via un navigateur web. [lecture seule, rev. HEAD uniquement]
- ∂ 2- Ou un serveur indépendant (svnserve) + SSH
- ∂ Une base de données Berkeley DB ou FSFS
- ∂ On accède au dépôt via une url
  - `https://jean@svn.exemples.com/svn/projet1/trunk`
- ∂ Possibilité d'accéder au dépôt via une interface webdav (KDE,...)  
[lecture seule, rev. HEAD uniquement]

### ∂ Contrôle des accès :

#### ∂ Général (blanket):

[Apache]

```
<Location /svn>
  DAV svn
  SVNParentPath /usr/local/svn
  AuthType Basic
  AuthName "Subversion repository"
  AuthUserFile /path/to/users/file
  <LimitExcept GET PROPFIND OPTIONS REPORT>
    Require valid-user
  </LimitExcept>
</Location>
```

#### ∂ Par répertoire:

[Apache + fichier d'accès]

```
<Location /repos>
  DAV svn
  SVNParentPath /usr/local/svn
  AuthzSVNAccessFile /path/to/access/file
  Require valid-user
  AuthType Basic
  AuthName "Subversion repository"
  AuthUserFile /path/to/users/file
</Location>
```

```
[groups]
calc-developers = harry, sally, joe
paint-developers = frank, sally, jane
everyone = harry, sally, joe, frank, sally, jane

[calc:/branches/calc/bug-142]
harry = rw
sally = r

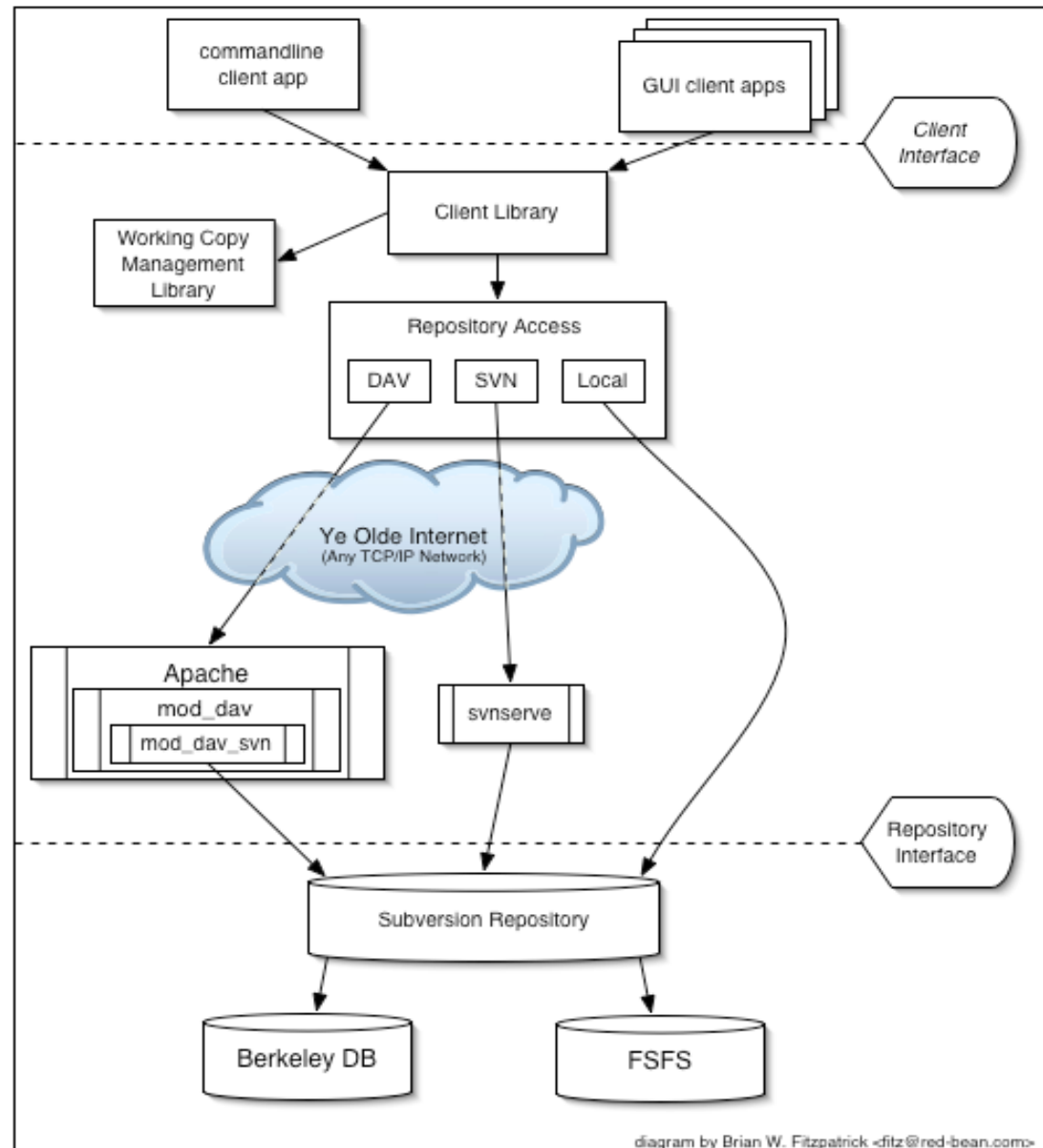
# give sally write access only to the 'testing' subdir
[calc:/branches/calc/bug-142/testing]
sally = rw
```

# Subversion, le CVS de demain!

## Détails techniques...



Subversion





Windows



Ligne de commande



**TortoiseSVN** [[www.tortoisesvn.org](http://www.tortoisesvn.org)]



Subclipse (pour Eclipse) [[subclipse.trigris.org](http://subclipse.trigris.org)]



...



GNU/Linux



**Ligne de commande**



eSVN [[esvn.umputun.com](http://esvn.umputun.com)]



**RapidSVN** [[rapidsvn.tigris.org](http://rapidsvn.tigris.org)]



Subclipse (pour Eclipse) [[subclipse.trigris.org](http://subclipse.trigris.org)]



Support dans KDE prévu pour 3.4 [[www.kde.org](http://www.kde.org)]



...



Mac OS X



**Ligne de commande**



svnX (alpha)



Xcode (intégration)

Subversion, le CVS de demain!

Conclusion

Références & liens





Subversion

*Version Control with Subversion,*  
Revision 12641, 2004

Ben Collins-Sussman Brian  
W. Fitzpatrick  
C. Michael Pilato

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



### Subversion

---

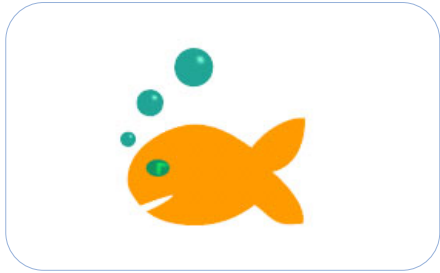
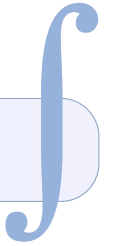
∂ *Subversion (Site officiel)*

<http://subversion.tigris.org/>

∂ *Subversion at the haus*

Zohar Melamed

<http://docs.codehaus.org/display/HAUS/How+to+Organize+a+Subversion+Repository>



CVS

## **Open Source Development with CVS, 3rd Edition**

Karl Fogel et Moshe Bar

(Le chapitre 2 peut être suffisant pour débiter)

<http://cvsbook.red-bean.com/>

## **CVS, Un ami de tous les jours**

Éric Martel, *Présenté à LinuQ, Janvier 2004*

<http://ericmartel.net:8080/documents/LinuQ/CVSpres.pdf>

- L* <http://subversion.tigris.org/>
- L* <http://www.gentoo.org/doc/fr/cvs-tutorial.xml>
- L* <http://subclipse.tigris.org/>
- L* <http://rapidsvn.tigris.org/>
- L* <http://subcommander.tigris.org/>
- L* <http://tortoisesvn.tigris.org/>
- L* <http://tortoisesvn.tigris.org/>
- L* <http://www.cvshome.org/>
- L* <http://www.tortoise cvs.org/>
- L* <http://www.lincvs.org/>
- L* <http://www.kde.org/apps/cervisia/>

## Subversion, le CVS de demain!

*Cette présentation est disponible en ligne:*

<http://www.rubico.info/docs/subversion/>