

# Gestion de versions

## Introduction à Subversion

Loïc Gouarin

<sup>1</sup>Laboratoire de mathématiques d'Orsay

12-14 décembre 2011

# Plan

- 1 Présentation générale
- 2 Modèle client-serveur
- 3 Modèle distribué
- 4 Subversion
- 5 Références

## Plan

- 1 Présentation générale
- 2 Modèle client-serveur
- 3 Modèle distribué
- 4 Subversion
- 5 Références

## Qu'est-ce qu'un gestionnaire de versions ?

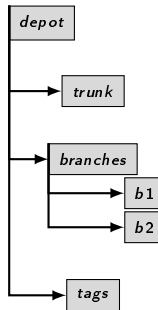
- C'est un logiciel.
- Il permet de conserver l'intégralité des versions d'un fichier ou d'un répertoire.
- Il permet de connaître toute l'historique des modifications.
- Il est généralement constitué
  - d'un dépôt (local, distant) : contenant toutes les versions,
  - de copies de travail : contenant les modifications d'un utilisateur qui seront ensuite incluses dans le dépôt.

# Pourquoi se servir d'un gestionnaire de versions ?

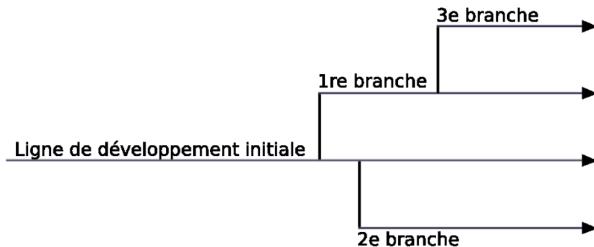
- conserver l'historique
- possibilité de revenir en arrière
- travail collaboratif

## Structuration générale

- Dépôt local ou distant répertoriant l'ensemble des modifications
- Arborescence



## Structuration générale



## Structuration générale

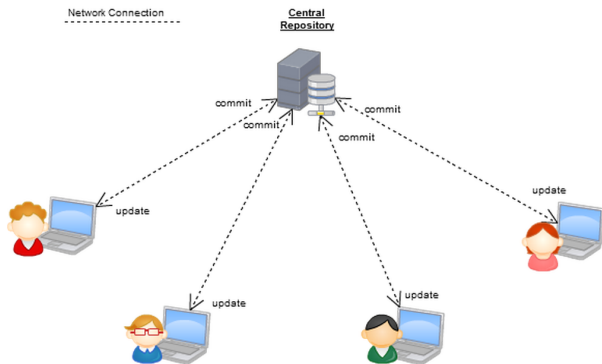
- Les branches servent à
  - corriger un problème sur une ancienne version,
  - développer 2 idées en parallèle,
  - gérer sa propre version du logiciel,
  - fusionner après une divergence.
- Les tags
  - sont des marques symboliques sur une révision,
  - permettent de définir les versions du projet,
  - permettent de nommer des branches.



## Plan

- 1 Présentation générale
- 2 Modèle client-serveur**
- 3 Modèle distribué
- 4 Subversion
- 5 Références

## Qu'est-ce que c'est ?



## Logiciels

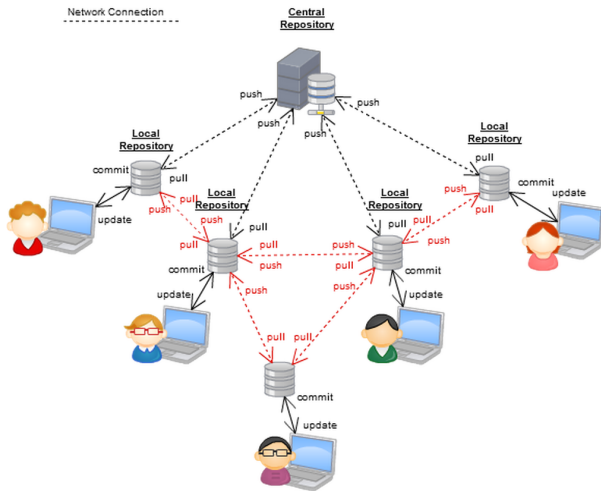
- CVS
- Subversion

## Plan

- 1 Présentation générale
- 2 Modèle client-serveur
- 3 Modèle distribué**
- 4 Subversion
- 5 Références

# Modèle distribué

## Qu'est-ce que c'est ?



## Logiciels

- Git
- Mercurial
- Bazaar

## Plan

- 1 Présentation générale
- 2 Modèle client-serveur
- 3 Modèle distribué
- 4 Subversion**
- 5 Références

- conçu pour remplacer CVS
- projet initié en 2000 par CollabNet
- le 14 février 2010, SVN est devenu officiellement un logiciel de la Fondation Apache





## Accès à un dépôt

- `file://` : accès direct au dépôt sur un disque local,
- `http://` : accès via le protocole WebDAV à un serveur Apache configuré pour Subversion,
- `https://` : identique à `http://`, mais avec chiffrement SSL,
- `svn://` : accès via un protocole personnalisé à un serveur svnserv,
- `svn+ssh://` : identique à `svn://`, mais à travers un tunnel SSH.

## Premières commandes

### Initialisation du dépôt

L'initialisation se fait sur la machine accueillant le dépôt via la commande `svnadmin`

```
terminal$ svnadmin create /chemin/du/dépôt
```

### Importation initiale

```
terminal$ svn import monprojet type_accès://chemin/du/dépôt  
-m "import initial"
```

## Premières commandes

### Extraction initiale

```
terminal$ svn checkout type_accès://chemin/du/dépôt
```

### Obtenir de l'aide

```
terminal$ svn help  
terminal$ svn help command
```

## Commandes utilisées pendant le cycle du développement

- mettre à jour une copie de travail  
`svn update`
- effectuer des changements dans l'arborescence  
`svn add`, `svn delete`, `svn copy`, `svn move`, `svn mkdir`
- examiner les changements effectués  
`svn status`, `svn diff`
- annuler des changements  
`svn revert`
- résoudre des conflits  
`svn update`, `svn resolve`, `svn resolved`
- propager des changements  
`svn commit`
- fusionner des branches  
`svn merge`

## Plan

- 1 Présentation générale
- 2 Modèle client-serveur
- 3 Modèle distribué
- 4 Subversion
- 5 Références**

- site de Subversion
- site de Git
- Systèmes de gestion de code source par Matthieu Herrb
- Introduction à Subversion
- Gestion de versions avec Subversion