

Introduction aux systèmes de gestion de version

1ère rencontre DevelopR6
28 mai 2009

F. Langrognat



DEVELOP
DR6

PLAN

- 1 Objectifs d'un Système de Gestion de Version (SGV)
- 2 Comment partager des fichiers ?
- 3 Petit tour d'horizon des SGV
- 4 Zoom sur Subversion

PLAN

1 Objectifs d'un Système de Gestion de Version (SGV)

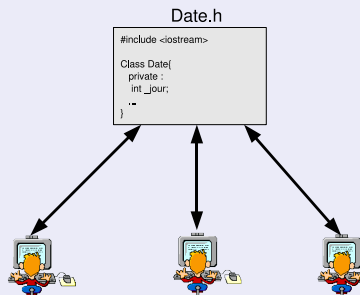
2 Comment partager des fichiers ?

3 Petit tour d'horizon des SGV

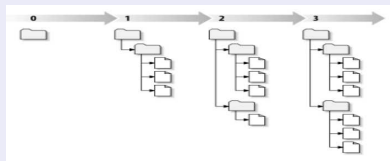
4 Zoom sur Subversion

Objectifs d'un Système de Gestion de Version

Travailler à plusieurs



Conserver l'historique



- Pouvoir revenir en arrière
- Qui a modifié pour la dernière fois ce fichier ?
- Quelles sont les différences entre 2 versions de ce fichier ?
- Quelle est la version du 15 mars 2007 ?

Objectifs d'un Système de Gestion de Version (suite)

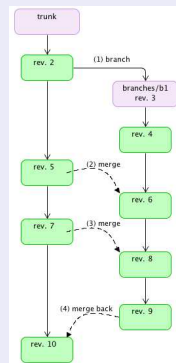
Et aussi ...

- **La gestion des branches**

Objectif : mener en parallèle plusieurs versions
(stable, testing, ...)

- **L'utilisation de tags**

Objectif : donner un nom explicite à une version pour
pouvoir y accéder facilement

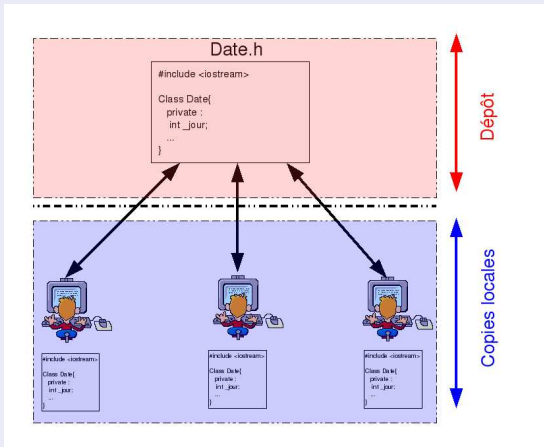


PLAN

- 1 Objectifs d'un Système de Gestion de Version (SGV)
- 2 Comment partager des fichiers ?
- 3 Petit tour d'horizon des SGV
- 4 Zoom sur Subversion

Principe de base

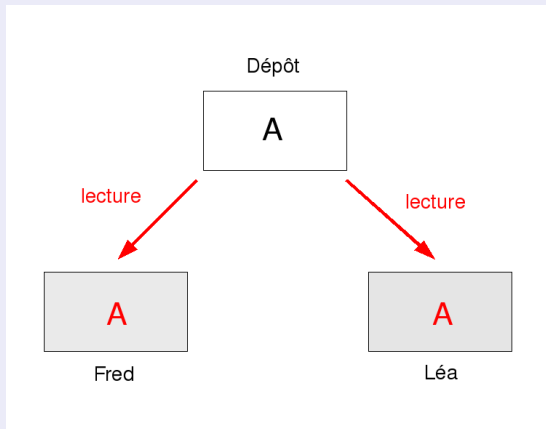
Notion de *dépôt* et *copie locale*



Les accès (écriture/lecture) se font via le **système de gestion de version**

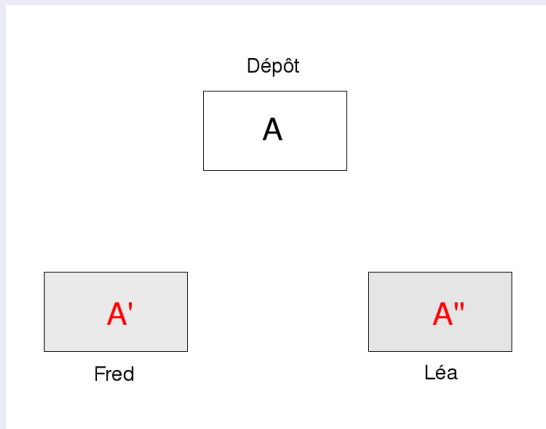
Le problème ...

Problème



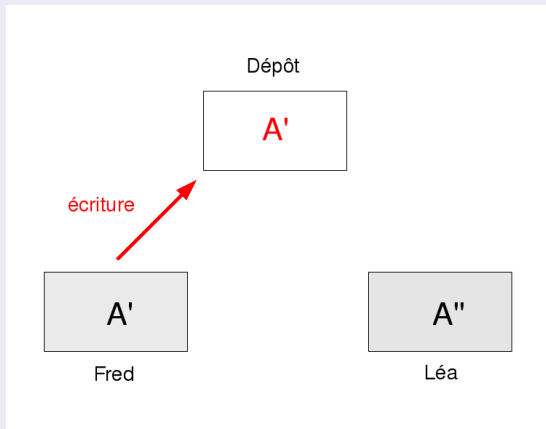
Fred et Léa accèdent au **même fichier** et le copient chez eux

Problème (suite)



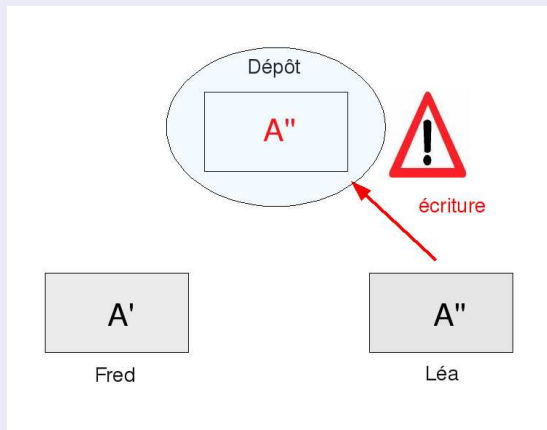
Fred et Léa font **chacun des modifications**

Problème (suite)



Fred écrit sur le dépôt

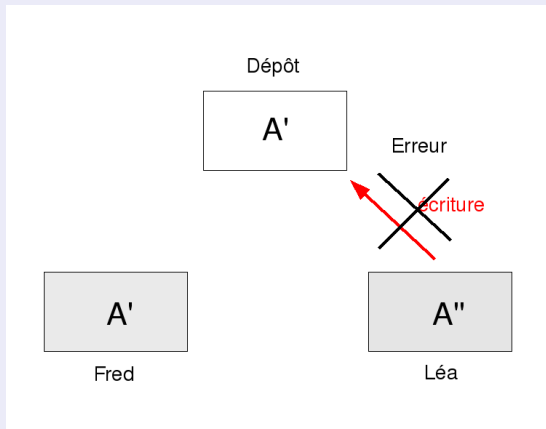
Problème (suite)



Léa écrit sur le dépôt en **écrasant la version de Fred**

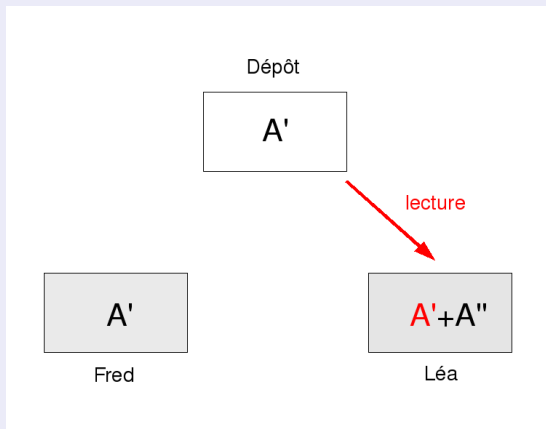
La solution ...

Solution



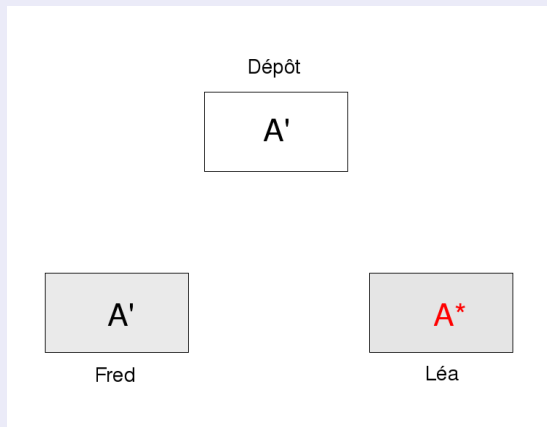
Léa **ne peut pas écrire** sur le dépôt car sa version **n'est pas à jour**

Solution (suite)



Léa **met à jour** : elle récupère la version du dépôt **sans perdre ses modifications**

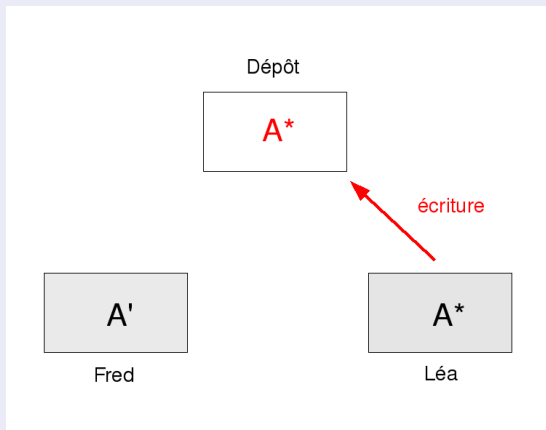
Solution (suite)



Léa **fusionne** la version du dépôt (A') avec sa version (A'')

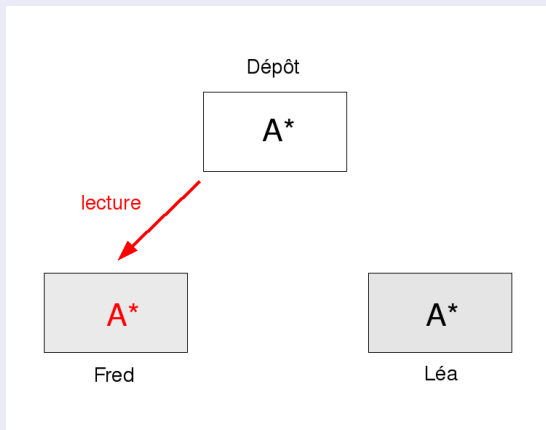
$A', A'' \rightarrow A^*$

Solution (suite)



Léa **peut écrire** sur le dépôt

Solution (suite)



Fred récupère la nouvelle version

Système de Gestion de Version

Système de Gestion de Version

Un SGV gère le mécanisme de **lecture-fusion-écriture**

- Les demandes de lecture, écriture se font via le SGV
- La fusion automatique est possible si
 - ▶ il s'agit d'un fichier texte
 - ▶ les modifications sont assez éloignées les unes de autres
- Le SGV conserve l'historique
- Et aussi : gestion des branches, tags, ...

PLAN

1 Objectifs d'un Système de Gestion de Version (SGV)

2 Comment partager des fichiers ?

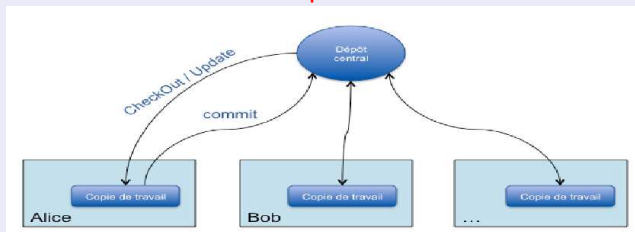
3 Petit tour d'horizon des SGV

4 Zoom sur Subversion

2 grandes catégories de SGV

1. Les systèmes centralisés

Un seul dépôt centralisé



Des qualités ...

- Technologie éprouvée
- Largement disponible (IDE, Forges)
- Portabilité
- Sécurité

et des défauts !

- Echange entre les dépôts impossible
- Travail hors connexion impossible
- Temps de mise à jour long pour de gros projet
- Et si le serveur tombe en panne ?

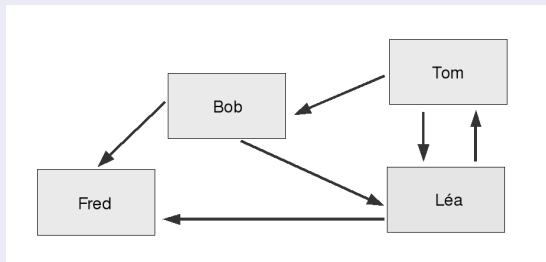
2 grandes catégories de SGV (suite)

2. Les systèmes décentralisés

Objectifs : pallier les problèmes des systèmes centralisés

- Pouvoir utiliser ce système *hors connexion*
- Ne pas être dépendant d'un dépôt centralisé (panne, temps, ...)
- Pouvoir échanger ses fichiers avec une partie des développeurs
- ...

Chaque développeur possède son propre dépôt



Les systèmes décentralisés

Les avantages d'un système centralisé (en local)

Chaque développeur a son propre dépôt et sa copie de travail

Il peut donc utiliser un SGV décentralisé pour (par exemple)
conserver l'historique ou gérer des branches en local

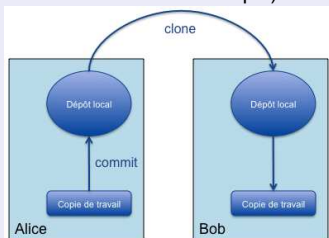


Les systèmes décentralisés (suite)

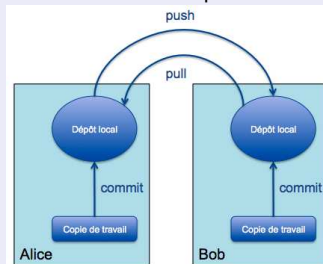
Travail entre dépôts

Les dépôts locaux peuvent communiquer

Clone d'un dépôt vers un autre (en conservant l'historique)



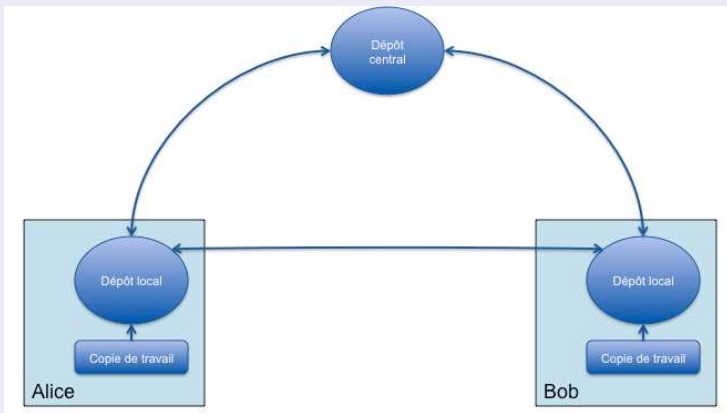
Ecriture/Lecture d'un dépôt vers un autre



Les systèmes décentralisés (suite)

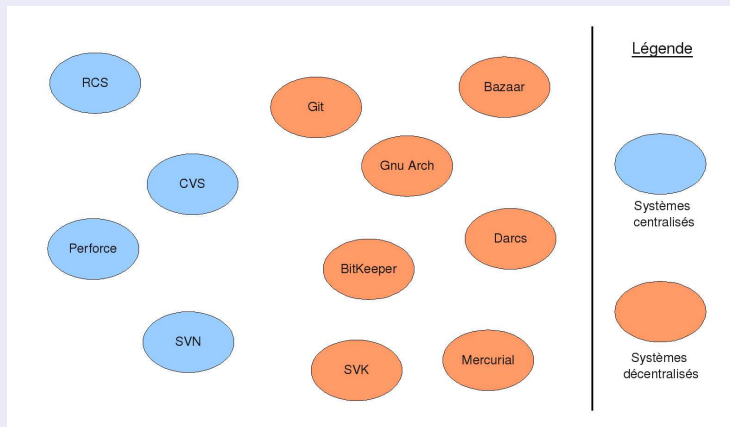
Avec un dépôt central ?

Si le nombre d'utilisateurs est grand, il peut être utile de mettre en place un **dépôt central** pour stocker la version la plus à jour du système



Quel SGV choisir ?

Cartographie (incomplète) des SGV



Quel SGV choisir ?

Vaste choix

- Technologie en pleine évolution
- De nouveaux systèmes apparaissent régulièrement

Elements à prendre en compte

- Pérérité : systèmes *leaders* vs. systèmes *émergents*
- Intégré dans des *IDE*
- Proposé par des *Forges*
- Interfaces graphiques
- Portabilité (multi OS)
- Sécurité
- Documentations abondantes
- Outils connexes (ex : cvs2svn)

PLAN

- 1 Objectifs d'un Système de Gestion de Version (SGV)
- 2 Comment partager des fichiers ?
- 3 Petit tour d'horizon des SGV
- 4 Zoom sur Subversion**

L'exemple de Subversion (SVN)

Un SGV très répandu

- Documentations très riches, forums actifs
- Interfaces graphiques
 - ▶ Linux : rapidsvn, kdesvn, esvn, Qsvn, ...
 - ▶ Windows : intégré à l'explorateur via le plugin TortoiseSVN
- Proposé dans les Forges et intégré dans certains IDE (Eclipse, Kdevelop)

Le *successeur* de CVS

Reprend le modèle de CVS en comblant certains manques :

- Renommage et déplacement de fichiers sans perte de l'historique
- Gestion des répertoires
- commits atomiques
- Gestion des *metadonnées* (ex : permissions)
- Possibilité de migrer de CVS vers SVN sans perte de l'historique (cvs2svn)
- Protocoles réseaux sécurisés (HTTPS)

FIN

Merci de votre attention