



Félix-Antoine Bourbonnais

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

28 janvier 2005

Université Laval,
Québec

1 2 **Programmer avec CVS** ... ∞ →

Acte 1

Présentation et utilisation

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]

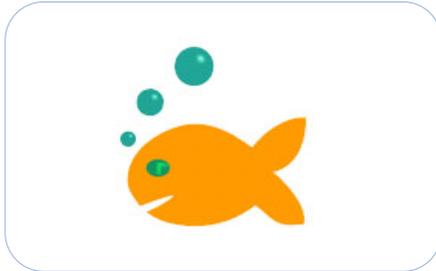
- ∂ Les équipes de travail deviennent de plus en plus grandes.
 - ∂ Difficile de fusionner et intégrer manuellement toutes les modifications.
 - ∂ Les modifications concurrentes sur une même partie sont fréquentes

- ∂ 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

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



Subversion

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

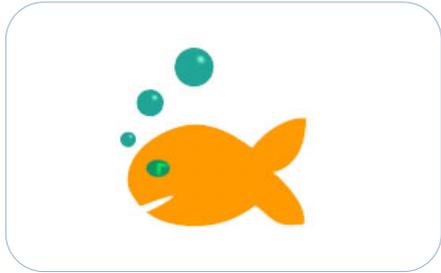
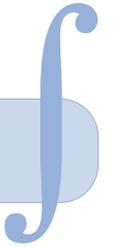
∂ Des développeurs (ou entreprises) de logiciels

∂ Des particuliers pour gérer leurs documents

(1) <http://stats.openoffice.org/spreadsheet/index.html>

Deuxième partie

Présentation de CVS



CVS

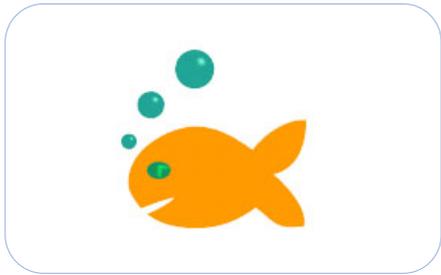
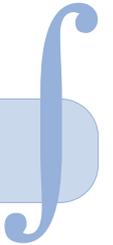
∂ CVS: Concurrent Versions System

∂ Permet la gestion des documents partagés et édités simultanément par plusieurs membres d'une même équipe.

« I created CVS to be able to cooperate with my students Erik Baalbergen and Maarten Waage on the ACK (Amsterdam Compiler Kit) C compiler. The three of us had vastly different schedules (one student was a steady 9-5 worker, the other was irregular, and I could work on the project only in the evenings). Their project ran from July 1984 to August 1985. CVS was initially called cmt, for the obvious reason that it allowed us to commit versions independently. » (1)

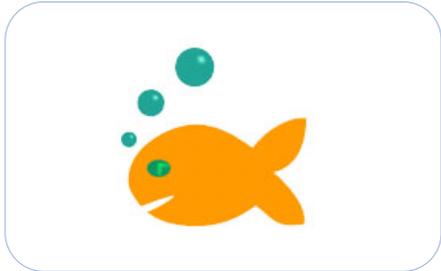
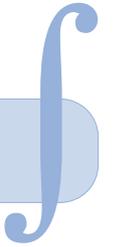
-- Dick Dune, <http://www.cs.vu.nl/~dick/ CVS.html>

(1) Source: Éric Martel, CVS Un ami de tous les jours..., Présenté à LinuQ 2004
<http://ericmartel.net:8080/documents/LinuQ/ CVSpres.pdf>



CVS

- ⌚ 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)
NB. L'algorithme est surtout efficace pour les fichiers textes
 - ⌚ 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.



CVS

- ∂ Fusion assistée
 - ∂ 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 conflits.
 - ∂ S'il y a conflit, CVS aide à repérer et à le résoudre
- ∂ Possibilité de marquer une étape dans le développement (Tag)
- ∂ Permet de scinder temporairement ou non un projet (Branch)

Troisième partie

Les concepts clefs

∂ Le dépôt [Repository] (syn: Référentiel)

∂ **Racine du CVS** (dossier le plus parent)

∂ Il s'agit de la base d'un CVS

∂ Espace de travail dans lequel **on place des projets**

∂ Conserve les configurations globales

∂ C'est avec le dépôt que l'on établit la connexion

∂ Méthodes de connexion:

∂ **pserver**: Par mot de passe (authentification et transfert en texte clair)

∂ **ssh**: **Par un « tunnel » SSH (sécuritaire!).** [:ext:utilisateur@serveur:depot]

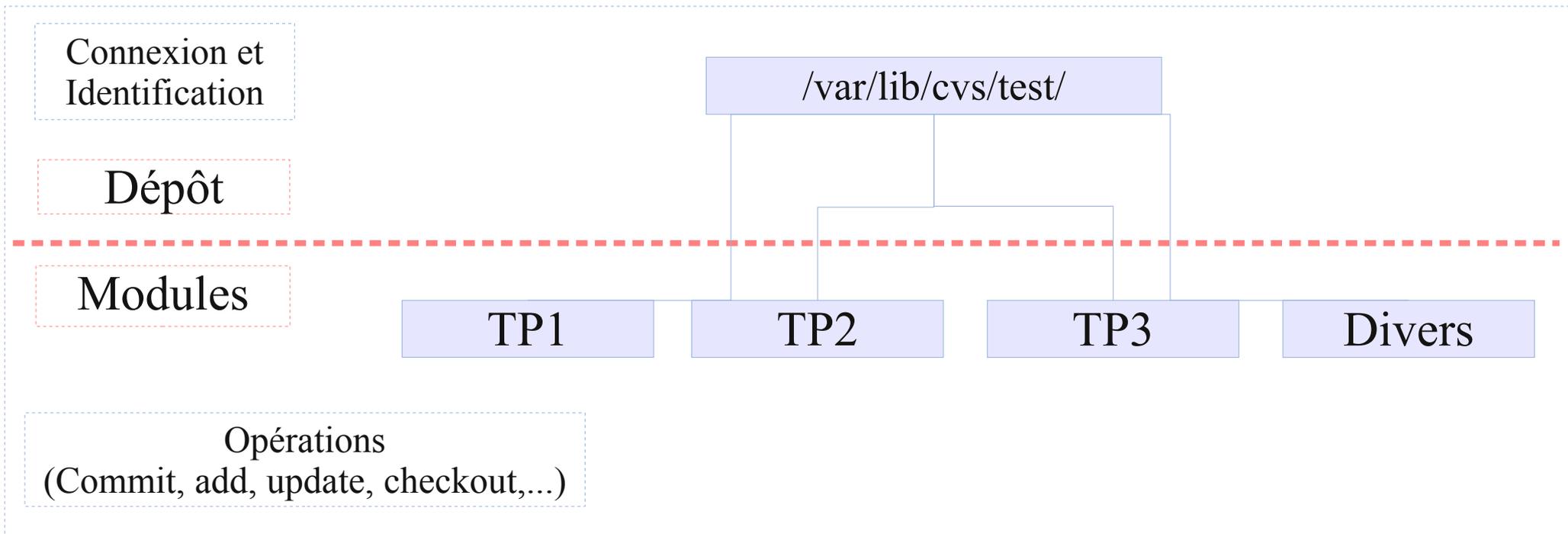
∂ **local**: Par le système de fichier local (peu pratique pour un travail collectif)

∂ Les modules

∂ On peut les comparer à des « projets »

∂ **Attention: il ne s'agit pas que de simples sous-répertoires!**

∂ La majorité des opérations (commandes) se feront sur les modules

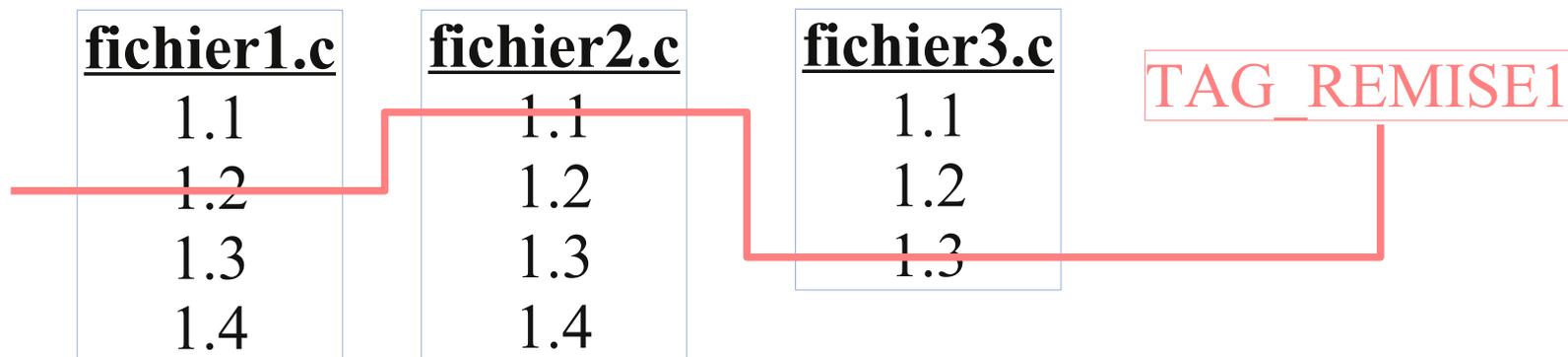


∂ Révision:

- ∂ « Version » d'un fichier
- ∂ Chaque fichier possède sa propre révision
- ∂ Par défaut, le dernier chiffre est incrémenté de 1 à chaque commit (envoi).
- ∂ Ex.: main.c (1.3) ; fonctions.c (1.5)
- ∂ **HEAD** : Tronc principal regroupant toutes les versions les plus récentes

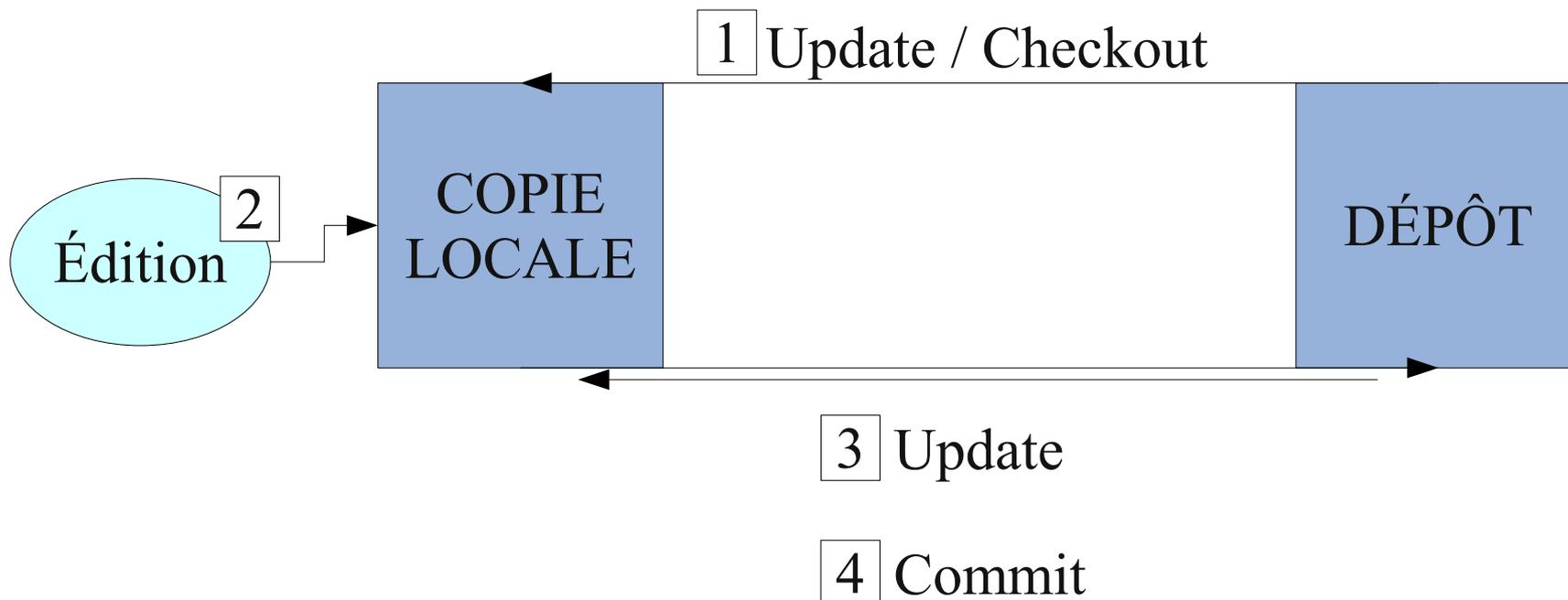
∂ Étiquette [Tag]

- ∂ Permet de marquer et fixer l'état de tous les fichiers d'un module à un instant par une étiquette.



- ▶ On ne modifie jamais les fichiers directement dans le dépôt (serveur).
- ▶ 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 >



∂ .cvsignore

- ∂ Permet de préciser les fichiers à ignorer

∂ Les mots clefs

- ∂ Dans un fichier texte!
- ∂ CVS remplace automatiquement ces mots par diverses informations
- ∂ Exemples:
 - ∂ \$Id\$: Fichier, Révision, Date (UTC), Auteur
 - ∂ \$Author\$: L'auteur
 - ∂ \$Date\$: La date (UTC)
 - ∂ ...

- ∂ Les principales commandes (il existe des alias de ces commandes) :
 - ∃ import : Création d'un nouveau module dans le dépôt
 - ∃ 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
 - ∃ tag: Fixer l'état des documents à un instant donné
 - ∃ ... man cvs : votre ami! (*sous Un*x*)
 - ∃ ... Google: un autre ami!

Quatrième partie

Utilisation de CVS



Ms. Windows®



Clients:



Ligne de commande



TortoiseCVS [www.tortoise cvs.org]



WinCVS [www.wincvs.org]



Diff:



WinMerge [winmerge.sf.net]



CSDiff [www.componentsoftware.com/products/csdiff/]



...



Mac OS X®



Ligne de commande



Xcode (intégration) [developer.apple.com/tools/macosxtools.html]



CLV



MacCVSClient [www.heilancoo.net/MacCVSClient/]



...



GNU/Linux

 Clients:

 **Ligne de commande**

 **Cervisia (Avec KDE)** [www.kde.org/apps/cervisia/]

 **LinCVS** [www.lincvs.org]

 Diff:

 Diff (Souvent inclus avec GNU/Linux)

 **Kompare (Avec KDE)**

 ...



Multiplateforme

★ gCVS

★ jCVS

★ tkCVS

★ **Eclipse (Intégration)**

★ ...

- ∂ Les majuscules/minuscules sont importantes (serveur unix)
 - ∂ Un fichier a.txt != A.txt sous Un*x
- ∂ Toujours entrer des messages détaillés à chaque commit
- ∂ Ajout de fichiers **textes** vs fichiers **binaires** (ajouter -kb)
- ∂ Soyez très disciplinés
 - ∂ Respectez et définissez des conventions entre les membres
 - ∂ Respectez l'ordre des opérations pour éviter des problèmes
- ∂ Éviter d'avoir à renommer des fichiers ou des dossiers
- ∂ Éviter les caractères spéciaux et accentués dans les noms de fichiers/dossiers
- ∂ N'attendez pas trop avant de faire un commit (une idée = un commit)
- ∂ Ne jamais nommer un dossier CVS

∂ Quelques conseils tirés du KDE Commit Policy:

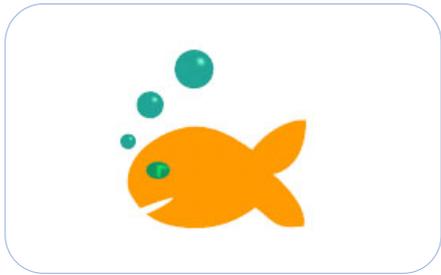
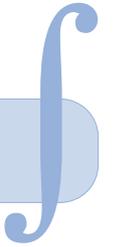
1. Think twice before committing.
2. **Never commit code that doesn't compile.**
3. Test your changes before committing.
4. Double check what you commit.
[**Do a "cvs update" and a "cvs diff" before committing**]
5. **Always add descriptive log messages.**
9. When you plan to make changes which affect a lot of different code in CVS, announce them on the relevant mailing list in advance.
11. Take responsibility for your commits.
[If your commit breaks something or has side effects on other code, take the responsibility to fix or help fix the problems.]
12. Don't commit code you don't understand.
16. **Avoid losing CVS history by renaming or moving files.**
18. **Don't add generated files to the repository.**

Source: <http://developer.kde.org/policies/commitpolicy.html>

- ∂ Vous travaillez **TOUJOURS** dans votre copie locale
- ∂ Quand vous ne savez plus quoi faire, demandez-vous ce que vous avez et ce que le dépôt a.
- ∂ Toutes les actions et opérations effectuées sur la copie locale seront propagées (sur le dépôt) uniquement après un commit.
 - ∂ Ceci inclut les opérations add et remove.
 - ∂ Le tag est une opération sur le module (pas sur la copie locale)
- ∂ Les révisions sont propres à chaque fichier
 - ∂ Pour marquer l'ensemble des fichiers il faut faire un tag

Cinquième partie

Références et liens



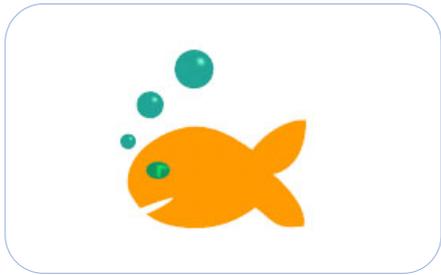
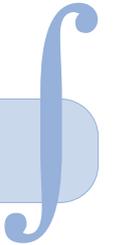
CVS

Tutoriel CVS

Vincent ADRIX,

*Pour le département d'informatique et de génie logiciel
de la faculté des sciences et de génie, Université Laval*

<http://www.ift.ulaval.ca/ift-10541/TPs/CVS/TutorielCVS.pdf>



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>



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>

L <http://www.cvshome.org/>

L <http://www.tortoise cvs.org/>

L <http://www.lincvs.org/>

L <http://www.wincvs.org/>

L <http://winmerge.sf.net/>

L <http://www.componentsoftware.com/products/csdiff/>

L <http://www.kde.org/apps/cervisia/>

L <http://www.gentoo.org/doc/fr/cvs-tutorial.xml>

L <http://www.eclipse.org/>

L <http://www.ift.ulaval.ca/ift-10541/TPs/CVS/TutorielCVS.pdf>

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

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

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

L <http://subclipse.tigris.org/>

L <http://rapidsvn.tigris.org/>

L <http://subcommander.tigris.org/>

L <http://tortoisesvn.tigris.org/>

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

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

Acte 2

Étude de cas

∂ Serveur:

∂ Serveur: dolly.ift.ulaval.ca

∂ Dépôt: /var/lib/cvs/10541aXX/

∂ Modules disponibles: - Aucun (pour l'instant) -

∂ Utilisateurs (équipe 0) :

∂ Jean

∂ Paul

∂ Connexion (pour Jean): [SSH]

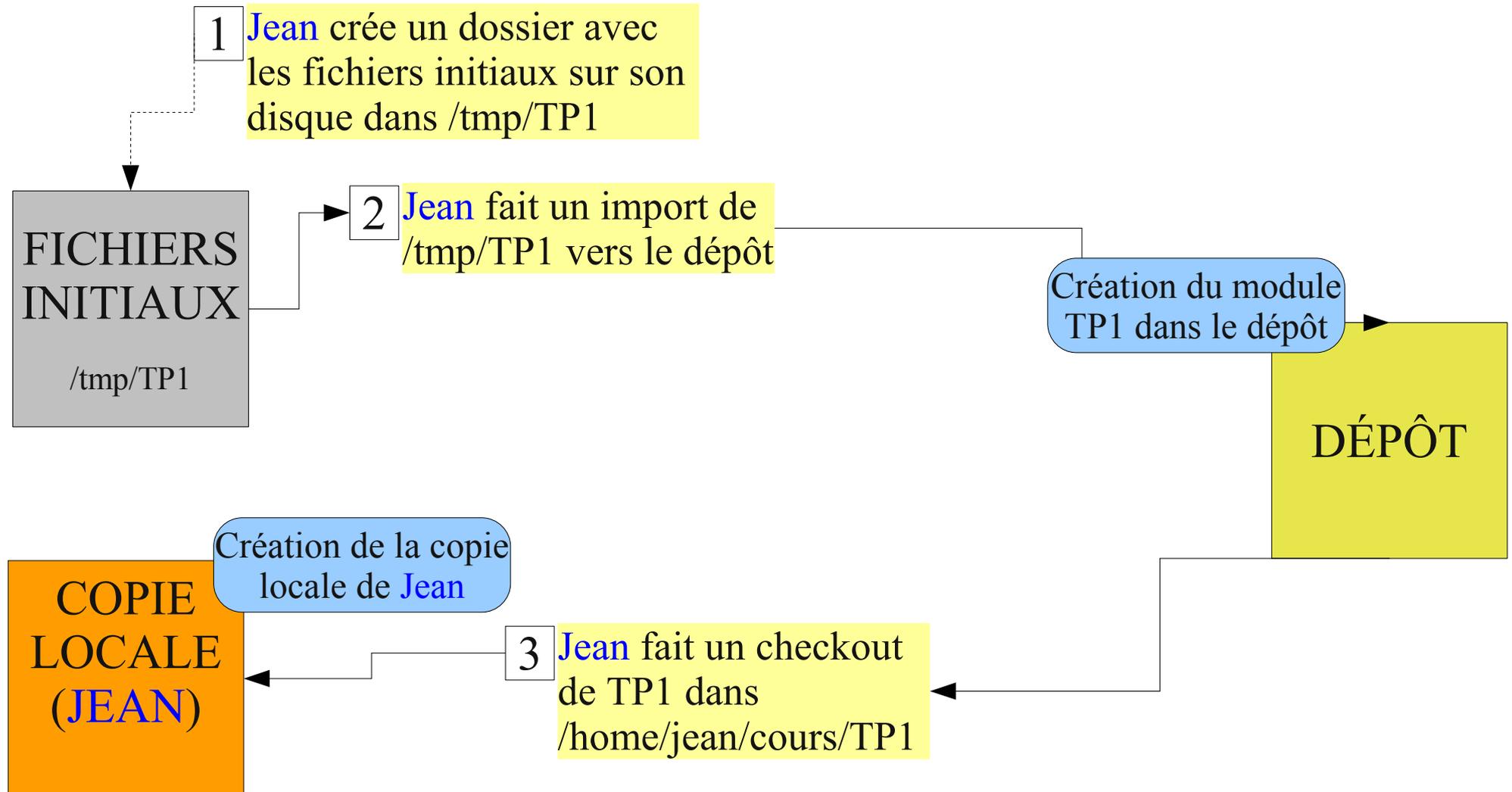
∂ :ext:jean@dolly.ift.ulaval.ca:/var/lib/cvs/10541aXX/

∂ Connexion (pour Paul): [SSH]

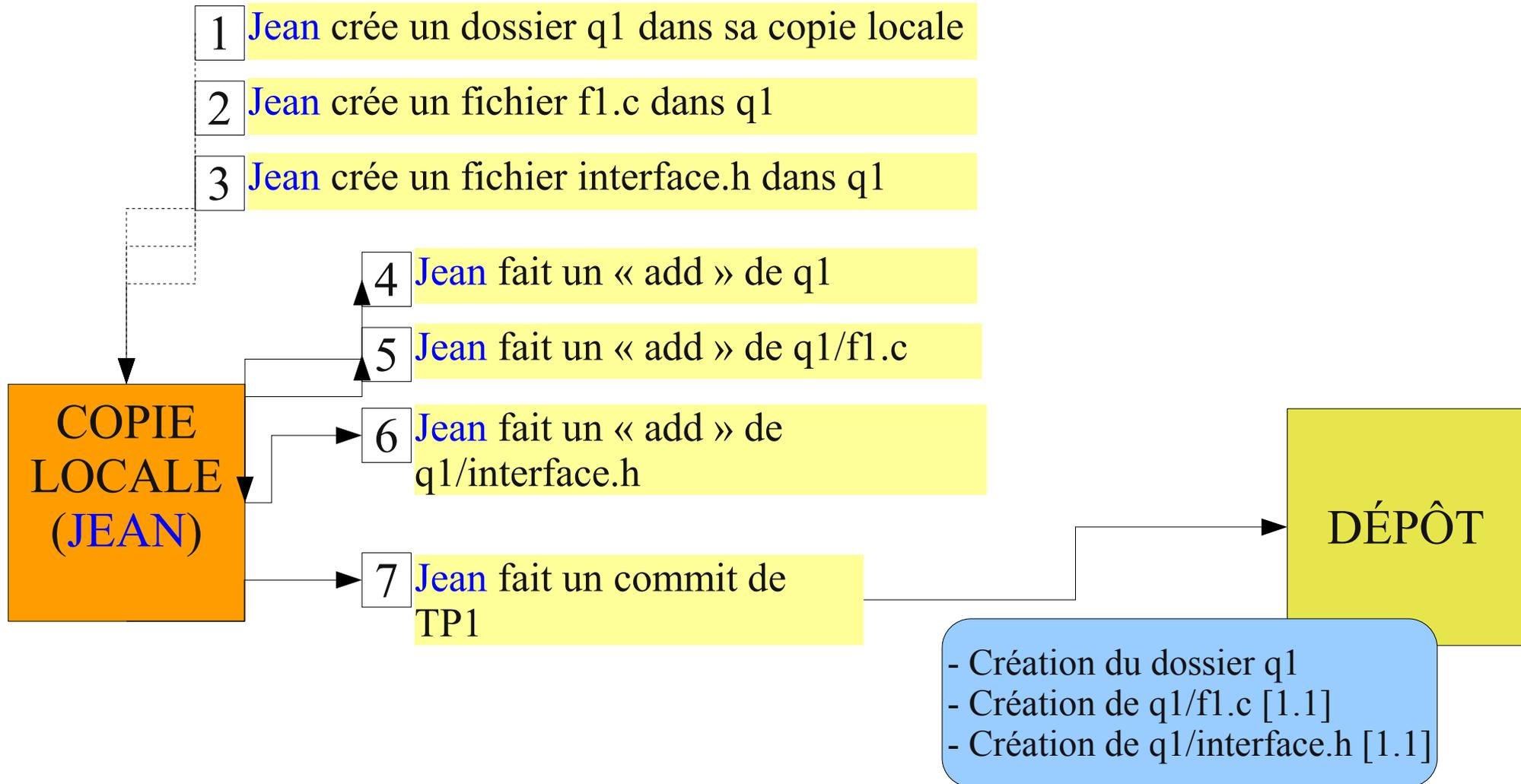
∂ :ext:paul@dolly.ift.ulaval.ca:/var/lib/cvs/10541aXX/

Étape 1: Création du module

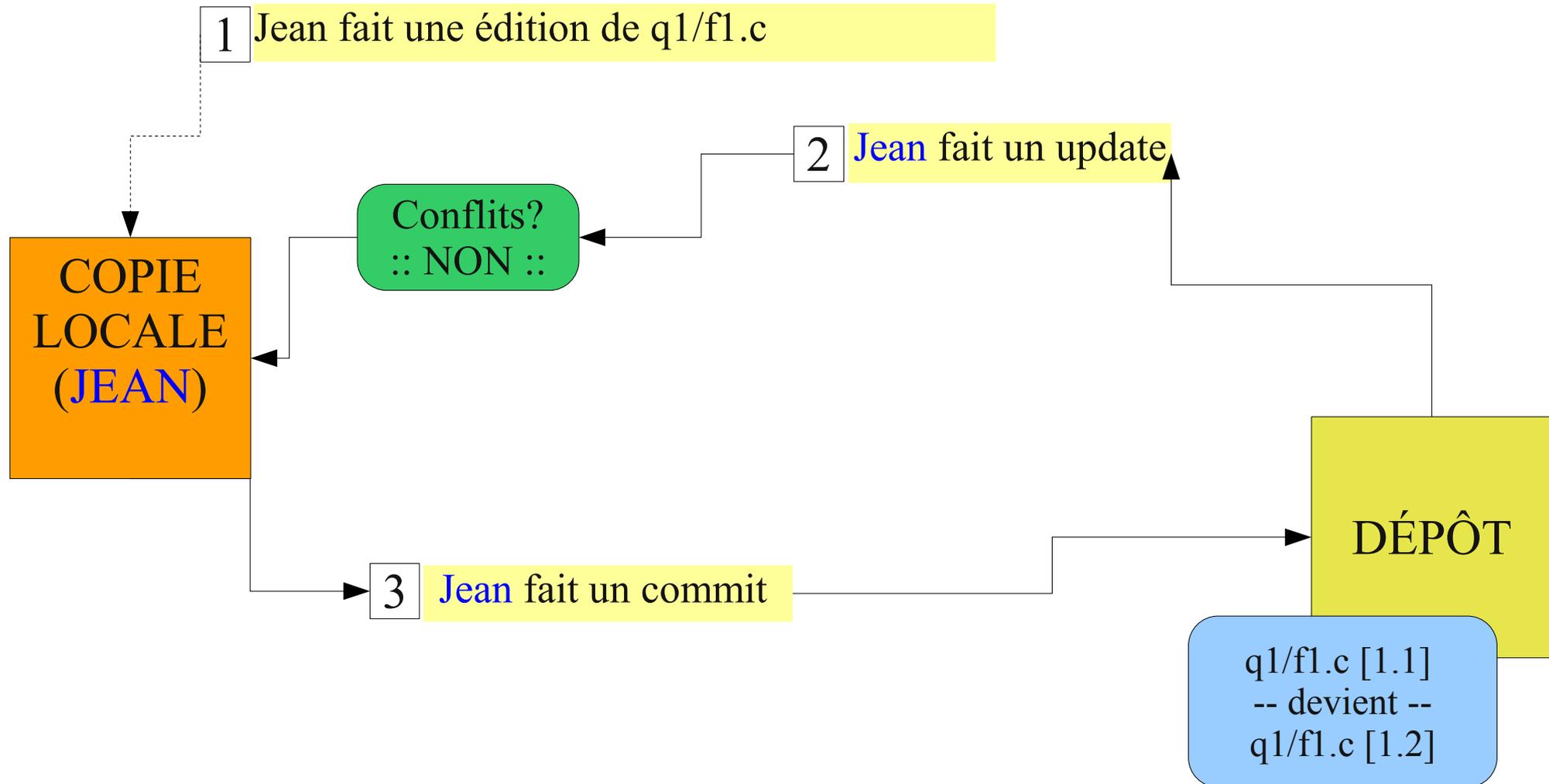
Jean va créer le premier module qui portera le nom TP1.



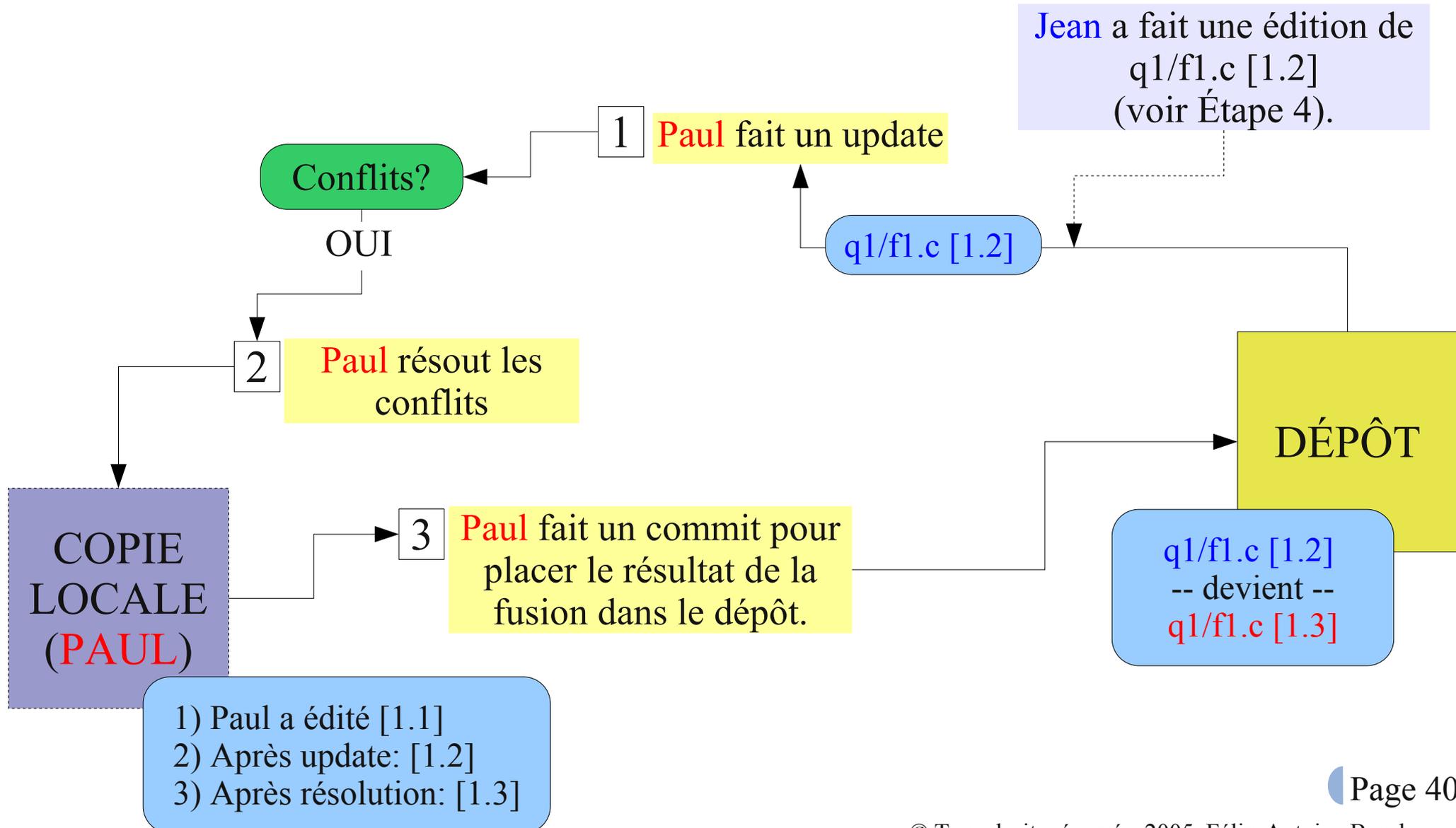
Jean va créer de nouveaux fichiers et répertoires



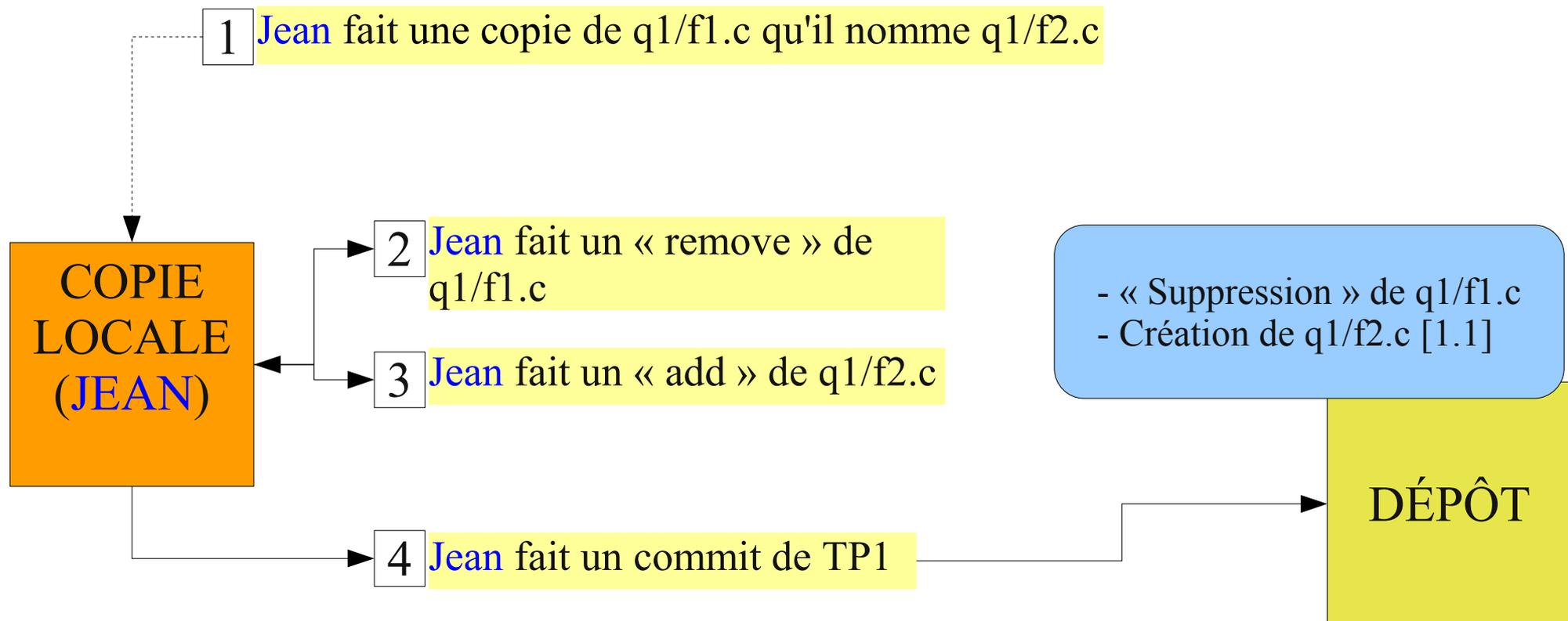
Jean veut modifier le fichier f1.c



Paul veut mettre sa copie locale à jour après avoir édité, lui aussi, q1/f1.c



Jean veut renommer le fichier f1.c pour f2.c



Cette présentation est disponible en ligne:

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

Acte 3

Démonstrations