

Sistemas de Control de Versiones

CVS Subversion Arch Aegis BitKeeper
Monotone Vesta Superversion
CVK

Introducción a los
Sistemas de Control de
Versiones



► Introducción: Ponentes



Manuel Resinas de Reyna

Investigador miembro de TDG (The Distributed Group) y estudiante de doctorado



Manuel J. Recena Soto

Alumno de la Escuela Técnica Superior de Ingeniería Informática (ETSII)



Sistemas de Control de Versiones
Subversion Arch Aegis BitKeeper
Monotone Vesta Superversion
SVK

► Introducción: Objetivos del seminario

- Proporcionar una visión general de la problemática que que existe al trabajar de forma individual y/o colaborativa con código fuente.
- Los sistemas de control de versiones como solución.
- Mencionar los distintos sistemas de control de versiones de software abierto.
- ¿Qué están usando grandes comunidades de desarrollo de software?
- ¿Sistemas de Control de Versiones sólo para desarrollos colectivos?

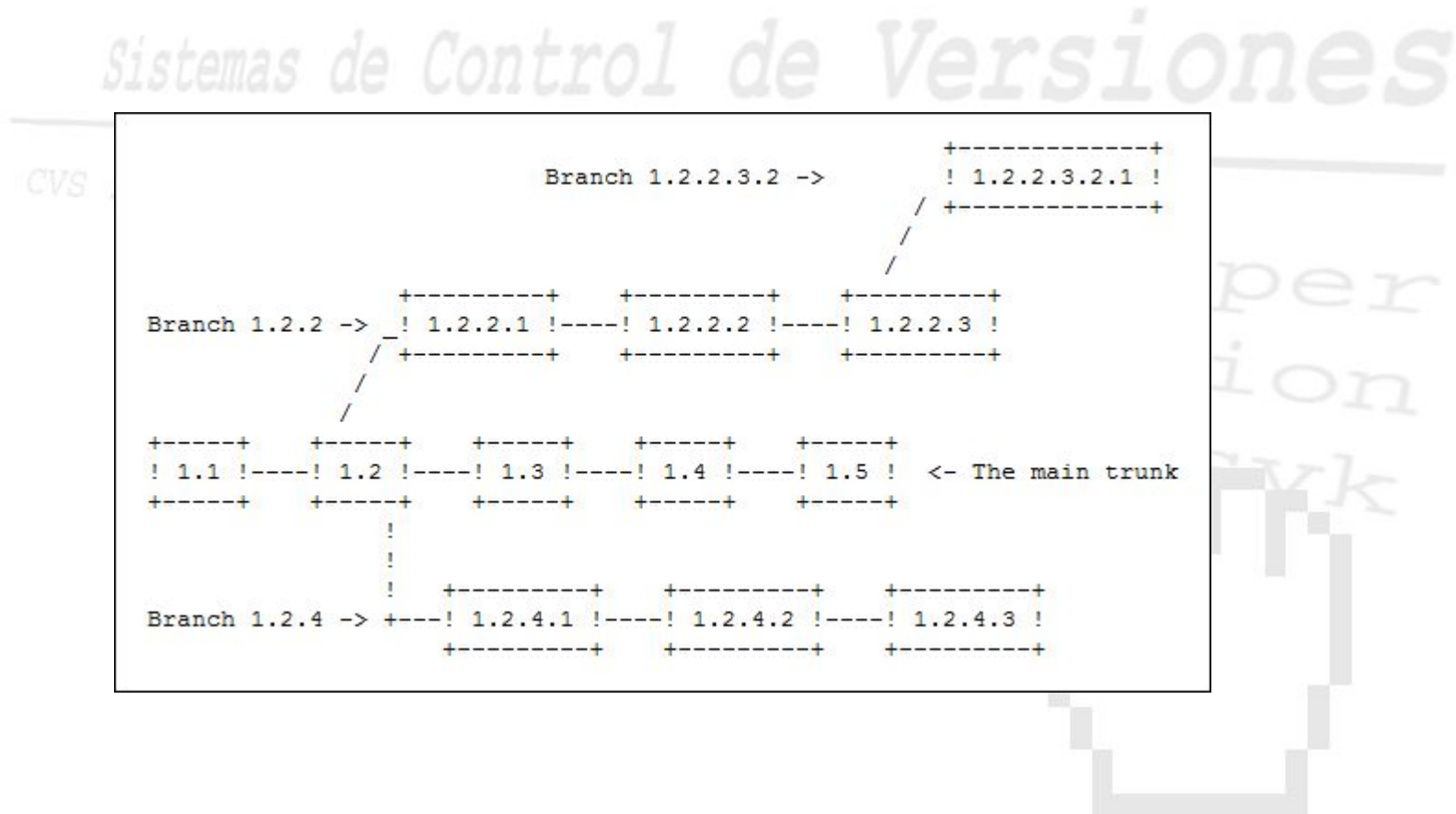
► Desarrollando de forma individual o colectiva

- Hagamos un breve resumen de las diferencias más significativas entre un desarrollo individual y uno colectivo.
- Ramas (branches), etiquetas (tags) y rama principal (trunk).
- Cuándo debemos definir una rama y por qué.
- No debemos olvidar que se está hablando de un desarrollo de software. Código fuente principalmente.



► Desarrollando de forma individual o colectiva

- Línea de tiempo asociada a la vida de un código fuente



Esquemas obtenidos de <http://svnbook.red-bean.com> y <http://www.cvshome.org> respectivamente

► Desarrollando de forma individual o colectiva

- Diferencias:
 - Un desarrollo colectivo implica la aparición de conflictos
- Semejanzas:
 - Etiquetado
 - Creación de ramas (estables, inestables, demo, etc)
 - Comparación entre estados de un código en el tiempo
 - Recuperación de estados pasados
 - Sistema de respaldo en caso de pérdida (backup)

► ¿Qué es un sistema de control de versiones?

- Sistema que sirva para mantener distintas versiones de un fichero, normalmente código fuente, documentación o ficheros de configuración
- Ejemplos:
 - Guardar distintas copias de los ficheros nombrándolos adecuadamente
 - Hacer *scripts* para automatizar las copias
 - Usar un *software* específico para realizar el control de versiones

► Usando un sistema de control de versiones

- Sistema de control de versiones mantiene un repositorio con los ficheros
- Forma habitual de trabajo:
 - Mantener una copia en local y modificarla. Después actualizarla en el repositorio.
 - Ventaja: No necesita acceso continuo al repositorio, puedes asegurarte de que lo actualizado “está bien”.
 - Con algunos sistemas de control de versiones es posible trabajar directamente contra el repositorio.
 - Ventaja: Más transparente
 - Inconveniente: bloqueo de ficheros

► El problema de compartir ficheros

- El problema que queremos evitar es el siguiente:

Sistemas de Control
CVS Subversion Arch Aeg
Monotone Vesta

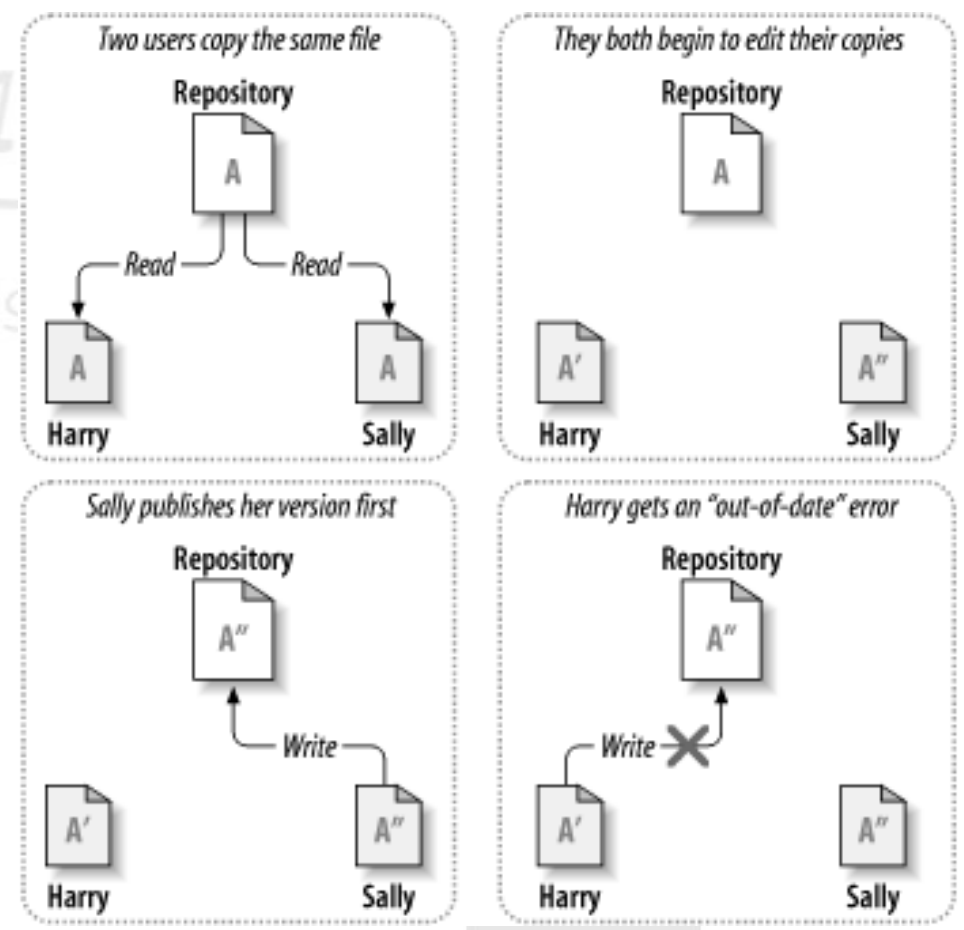


Imagen obtenida de <http://svnbook.red-bean.com/en/1.1/ch02s02.html> (Subversion book)

► Solución Bloquear-Modificar-Desbloquear

- Una primera solución al problema es la siguiente:

Sistemas de Control
CVS Subversion Arch Aeg
Monotone Vesta

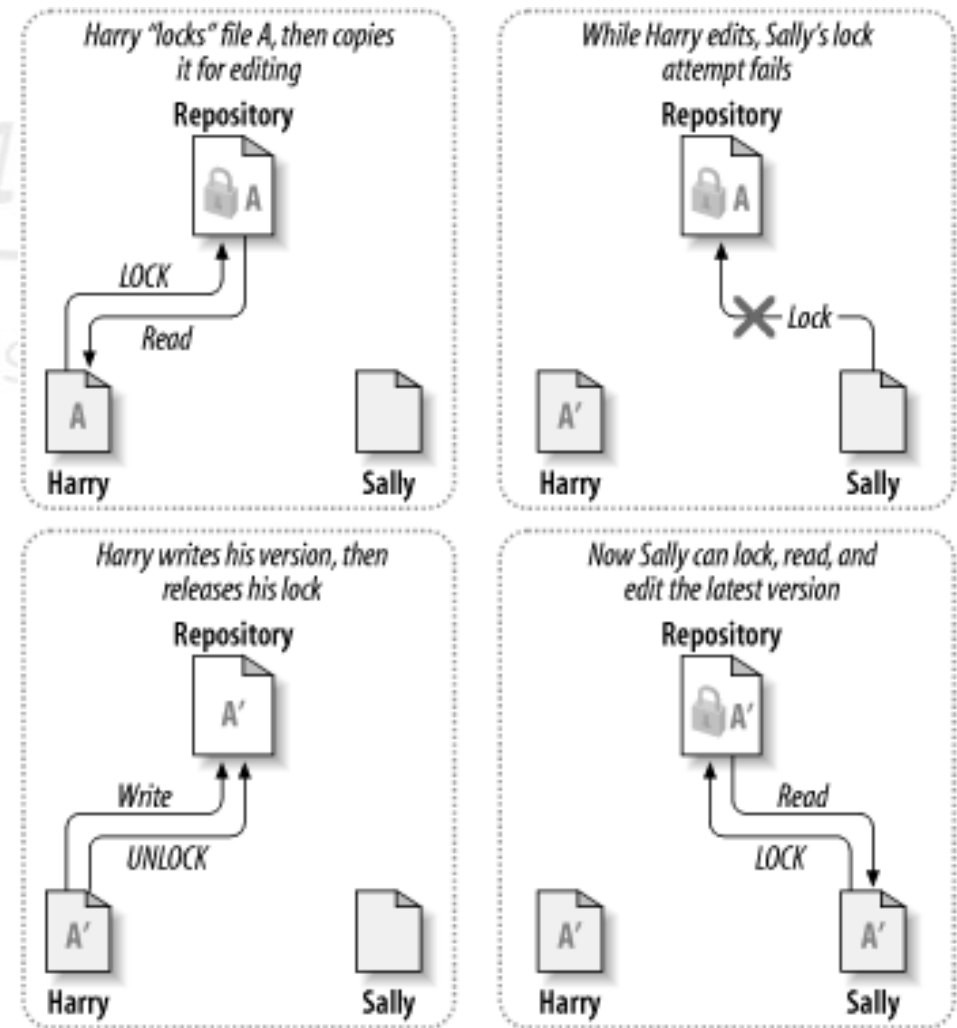


Imagen obtenida de <http://svnbook.red-bean.com/en/1.1/ch02s02.html> (Subversion book)

► Solución Copiar-Modificar-Unir

- La segunda solución (la más habitual) es:

Sistemas de Control
CVS Subversion Arch Aeg
Monotone Vesta

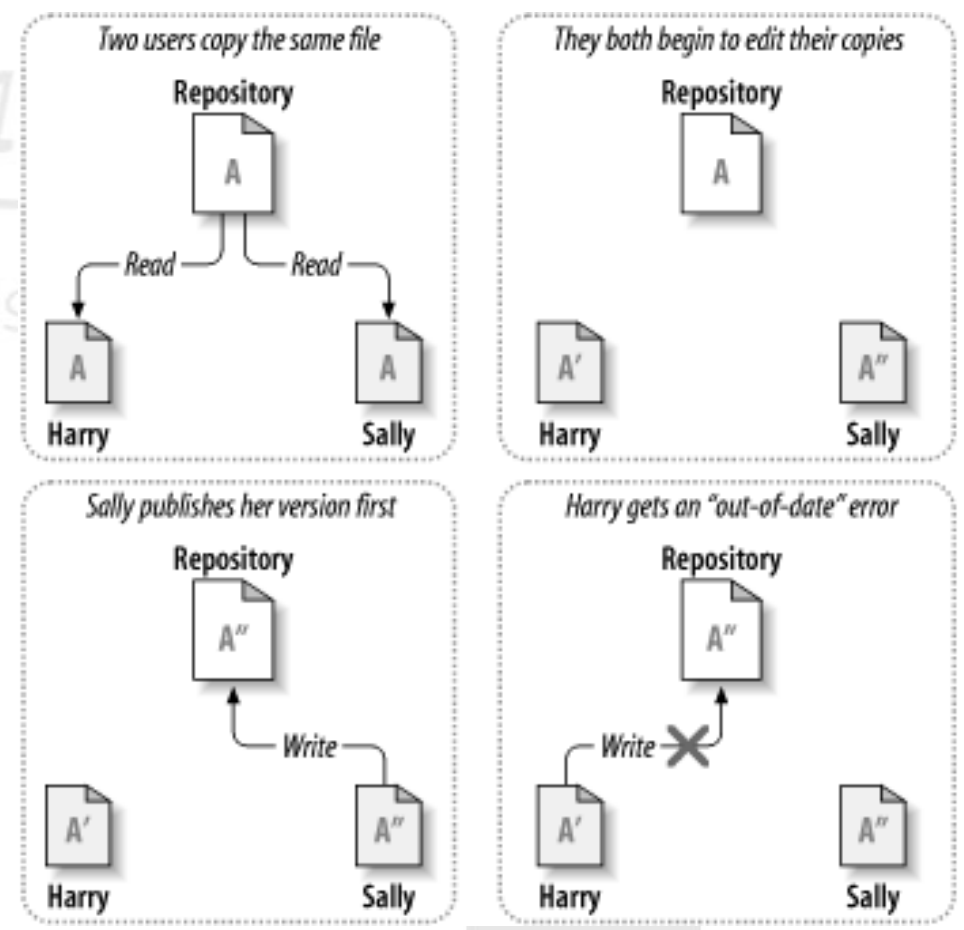


Imagen obtenida de <http://svnbook.red-bean.com/en/1.1/ch02s02.html> (Subversion book)

► Solución Copiar-Modificar-Unir

- Continuamos con la solución

Sistemas de Control
CVS Subversion Arch Aeg
Monotone Vesta

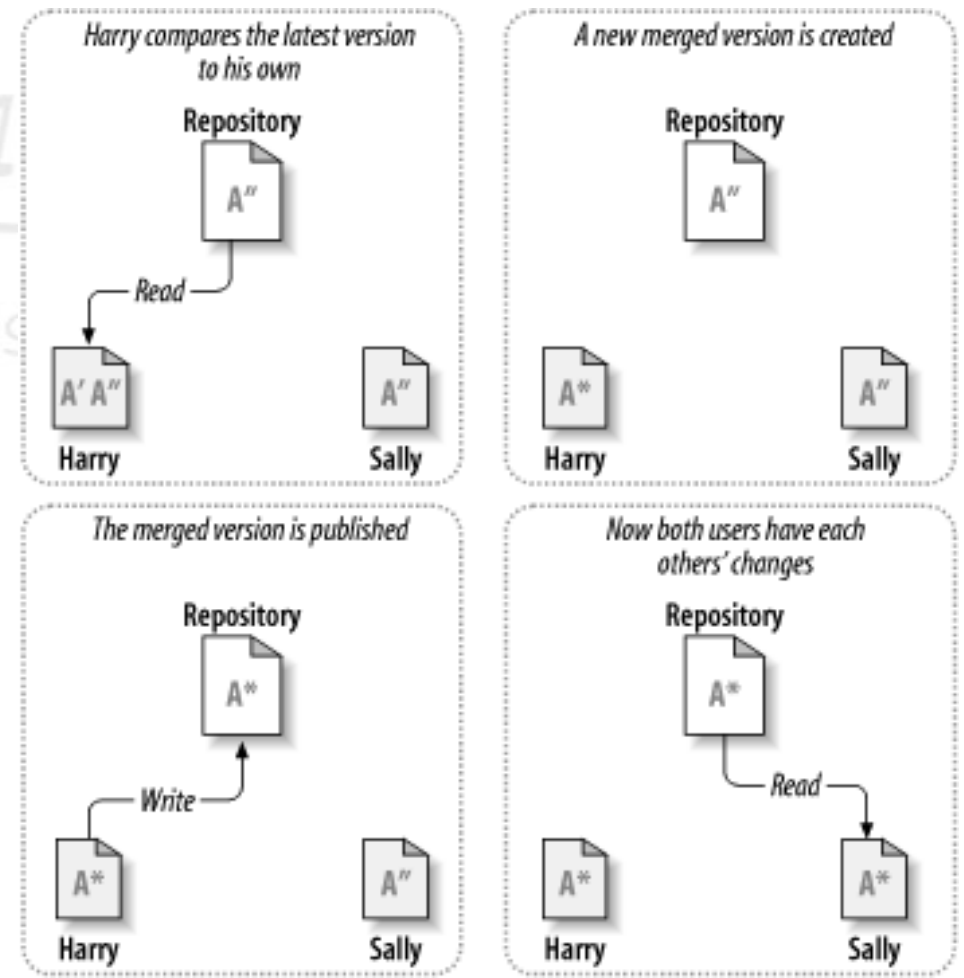


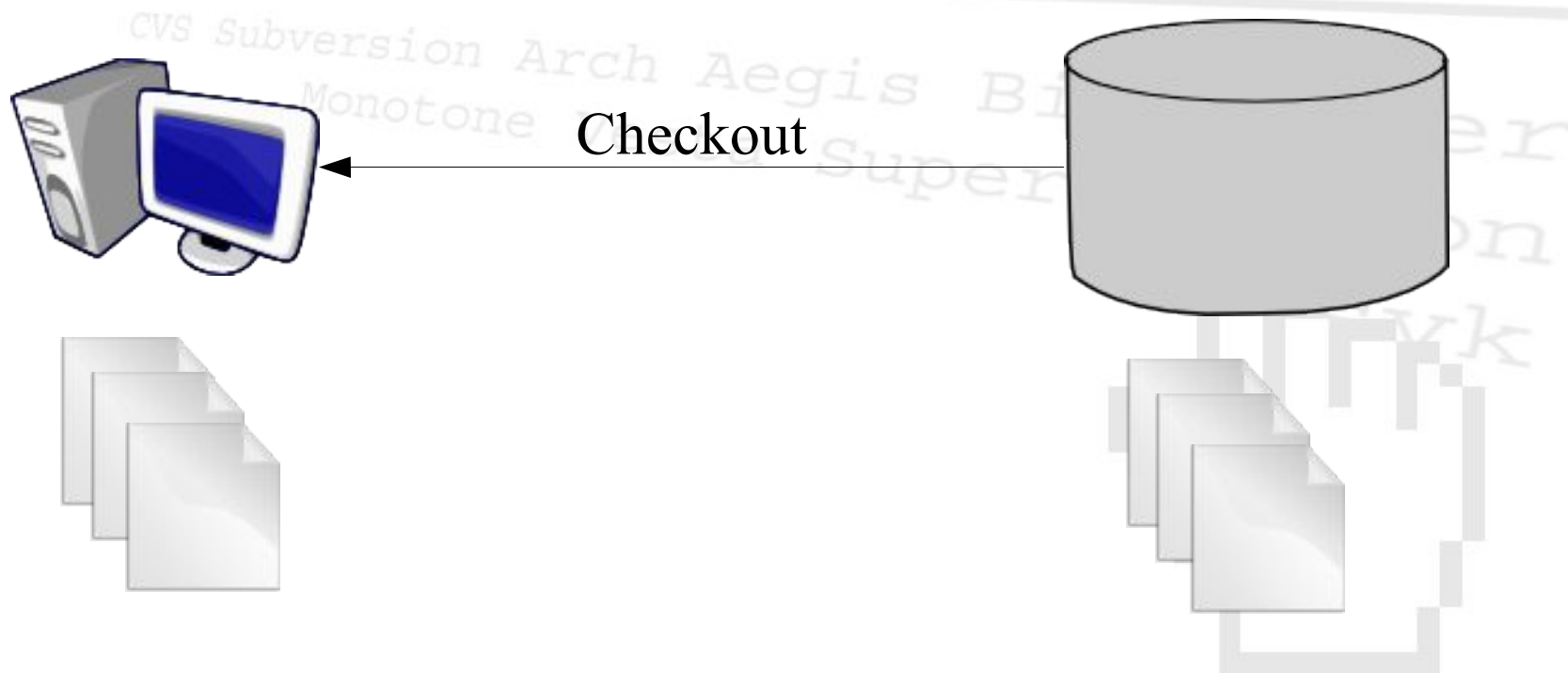
Imagen obtenida de <http://svnbook.red-bean.com/en/1.1/ch02s02.html> (Subversion book)

► Uso de un sistema de control de versiones

- Procedimiento de uso habitual de un sistema de control de versiones:
 - Descarga de ficheros inicial (*Checkout*)
 - Ciclo de trabajo habitual:
 - Modificación de los ficheros
 - Actualización de ficheros en local (*Update*)
 - Resolución de conflictos (si los hay)
 - Actualización de ficheros en repositorio (*Commit*).

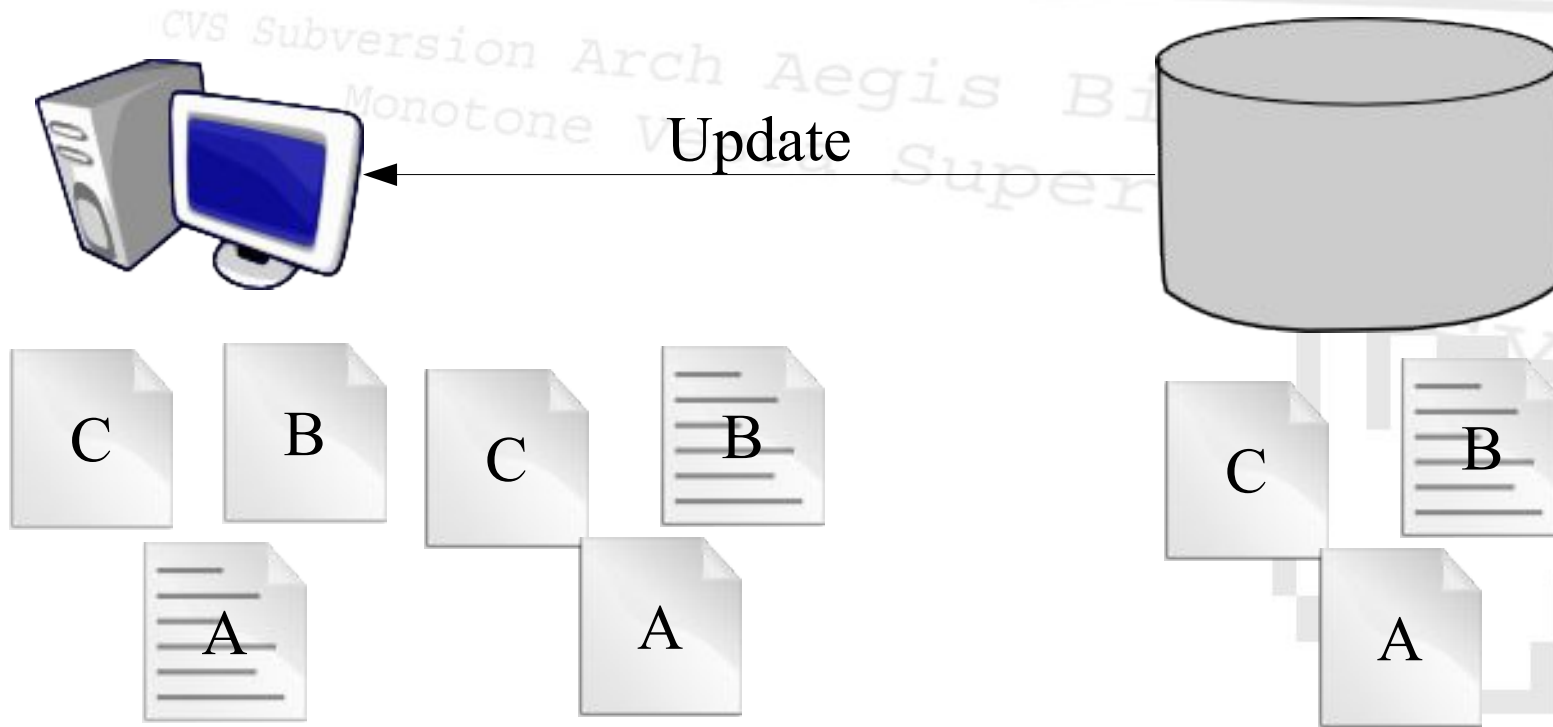
► Descargando los ficheros (*checkout*)

- El primer paso es bajarse los ficheros del repositorio
- El *checkout* sólo se hace la primera vez que se usan esos ficheros



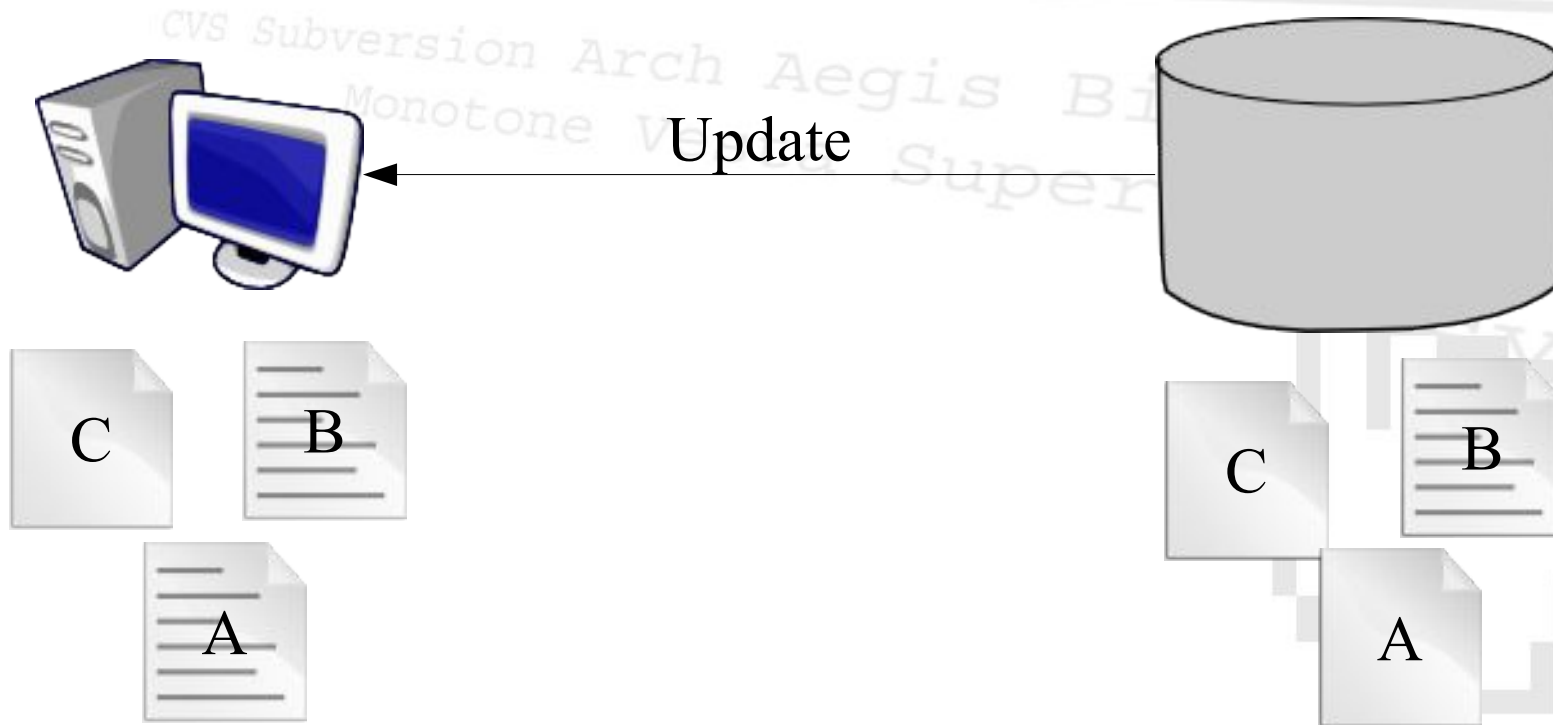
► Actualizando los ficheros (*update*)

- Modifica los ficheros en local
- Sincronizar los ficheros con los del repositorio



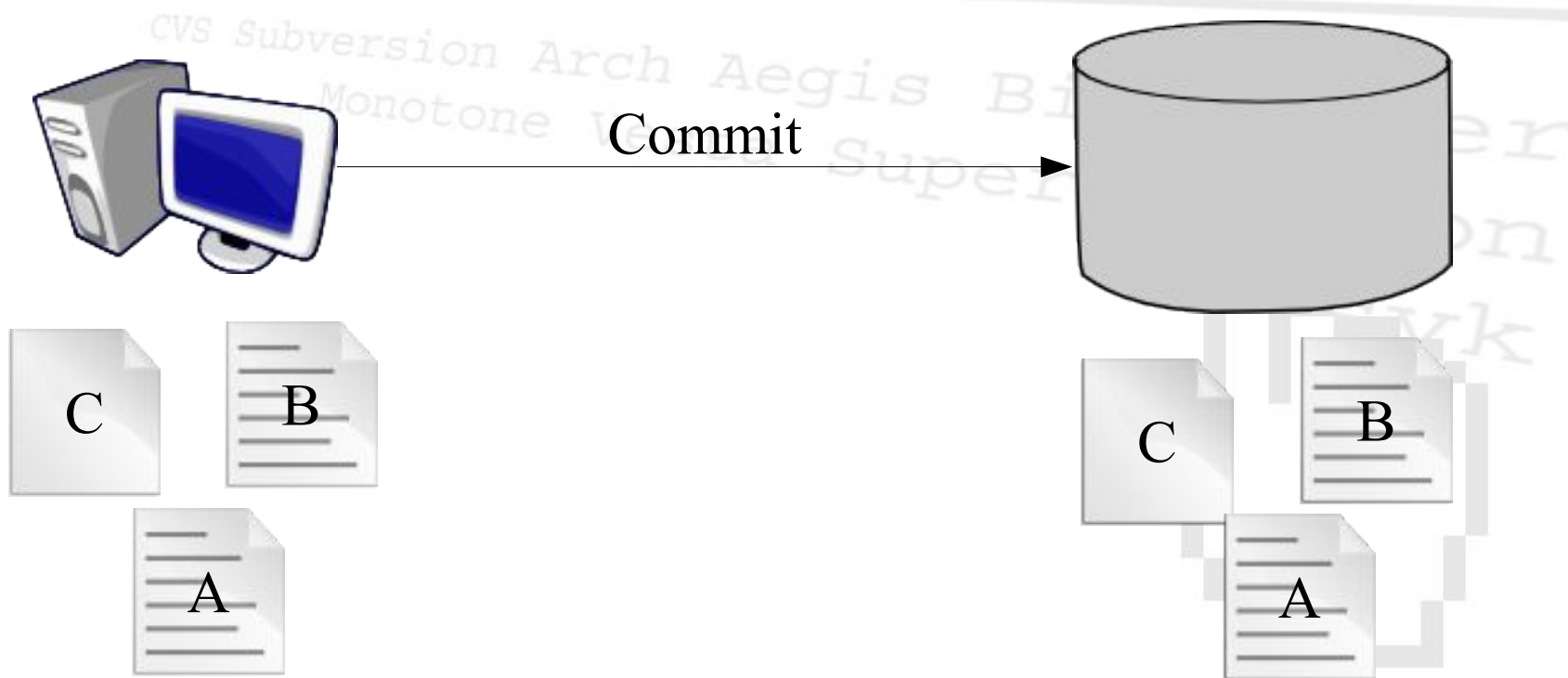
► Actualizando los ficheros (*update*)

- Modifica los ficheros en local
- Sincronizar los ficheros con los del repositorio



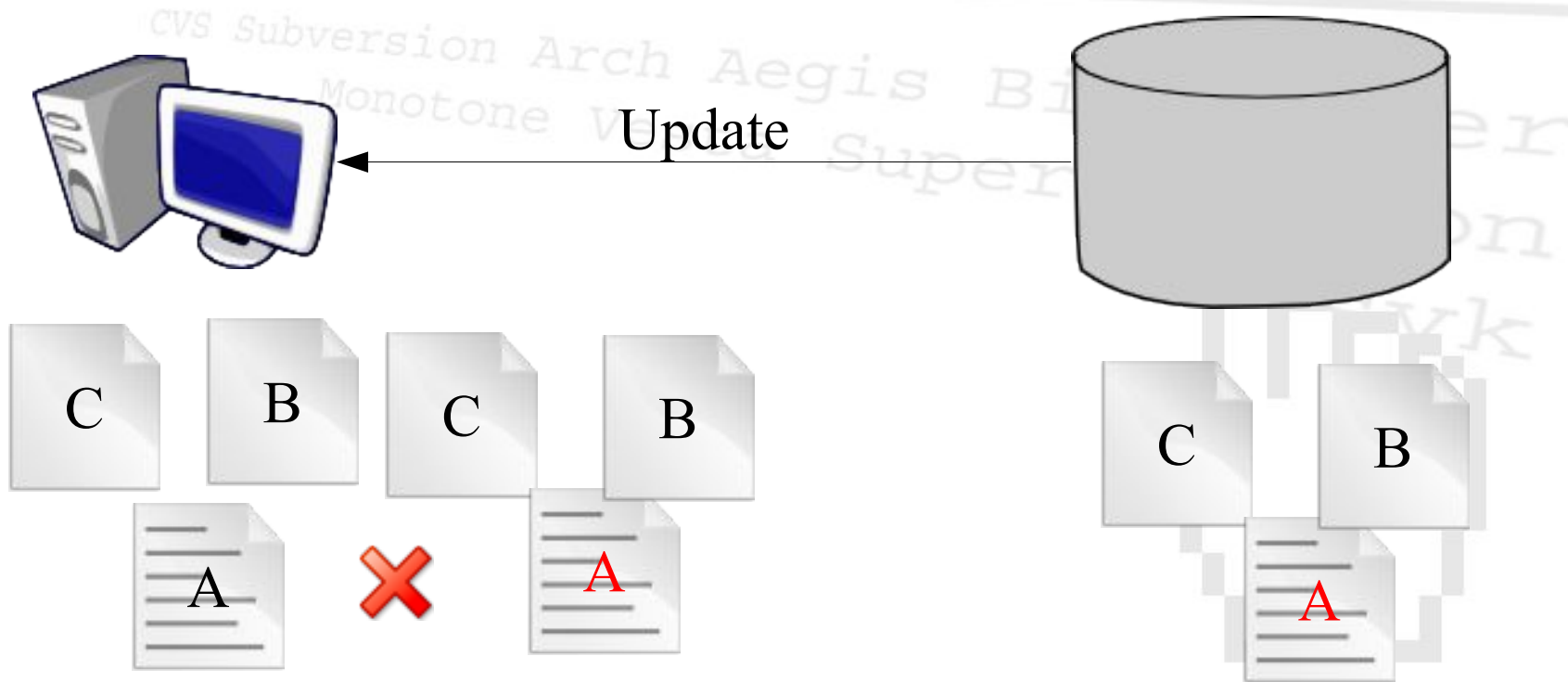
► Actualizando el repositorio (*commit*)

- Modifica los ficheros en el repositorio
- El sistema de control de versiones comprueba que las versiones que se suben estén actualizadas



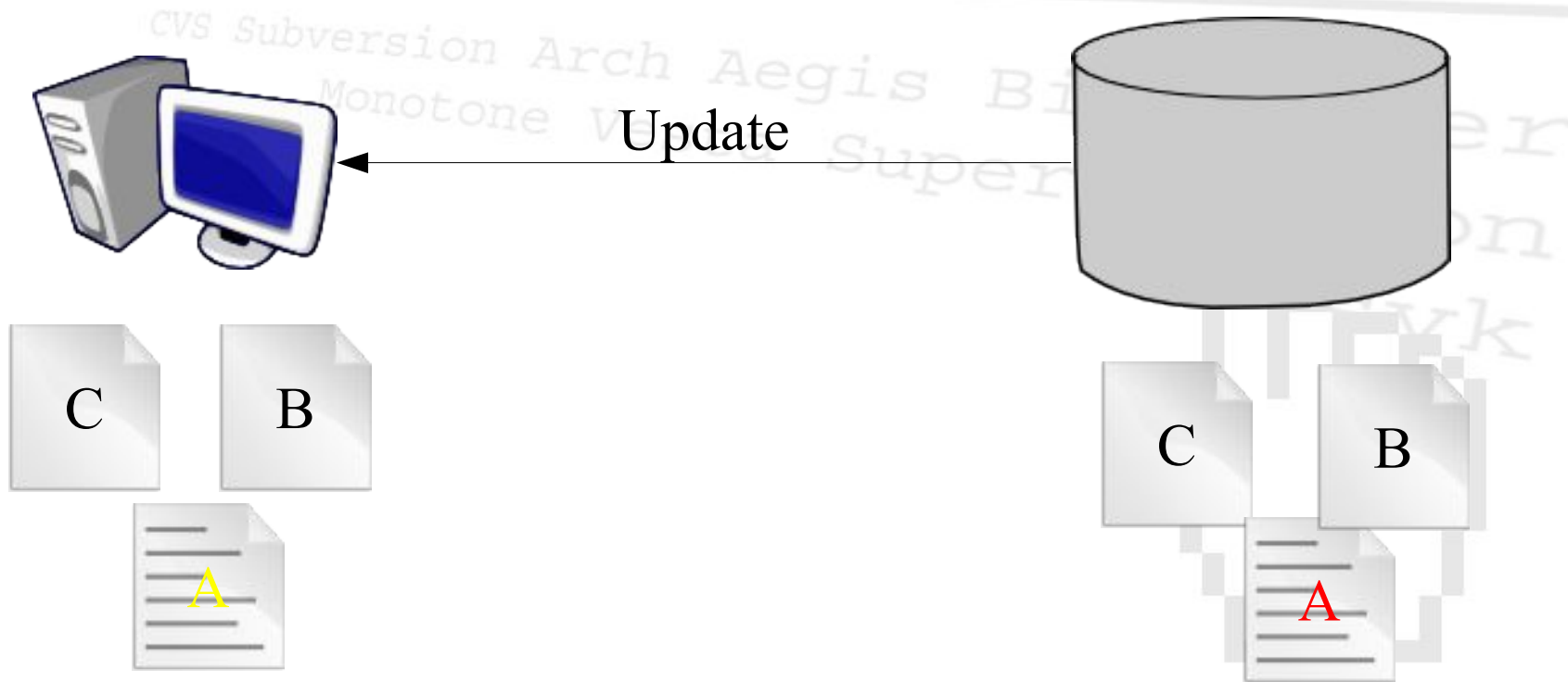
► Actualizando los ficheros (*update*)

- Modifica los ficheros en local
- Sincronizar los ficheros con los del repositorio



► Actualizando los ficheros (*update*)

- Modifica los ficheros en local
- Sincronizar los ficheros con los del repositorio



► Distintas alternativas

- Hay muchas formas de implementar los conceptos antes mencionados
- Nos centraremos en las siguientes alternativas:
 - CVS
 - Subversion
 - Arch
 - Monotone
 - Bitkeeper



► Operaciones con el repositorio

▪ *Commits* atómicos y conjuntos de cambios

	<i>Commits</i> atómicos	Soporte conjuntos de cambios
- CVS	✗	✗
- Subversion	✓	✓
- Arch	✓	✓
- Monotone	✓	✓
- Bitkeeper	✓	✓



► Mover, renombrar y copiar ficheros

- ¿Se pueden copiar, mover y renombrar ficheros?

	Mover/renombrar	Copiar
- CVS	✗	✗
- Subversion	✓	✓
- Arch	✓	✗
- Monotone	✓	✓
- Bitkeeper	✓	✓



► Administración de repositorios remotos

■ Administración de repositorios remotos

Clonar a local un repositorio remoto

Propagar cambios de un repositorio a otro

	Clonar a local un repositorio remoto	Propagar cambios de un repositorio a otro
- CVS	✗	✗
- Subversion	✓ *	✓ *
- Arch	✓	✗
- Monotone	✓	✓
- Bitkeeper	✓	✓

* Requiere de una utilidad adicional

► Acceso al repositorio

■ Acceso al repositorio

	Permisos acceso	Forma de acceso remoto
- CVS	Limitado	Servidor propio/Tunel SSH
- Subversion	✓	WebDAV/Propio/Tunel SSH
- Arch	✓ (sistema archivos)	FTP/HTTP/SMB/NFS/SFTP
- Monotone	✓	Propio (netsync)
- Bitkeeper		HTTP + Protocolo propio para comunicar repositorios

► Subversión como evolución de CVS

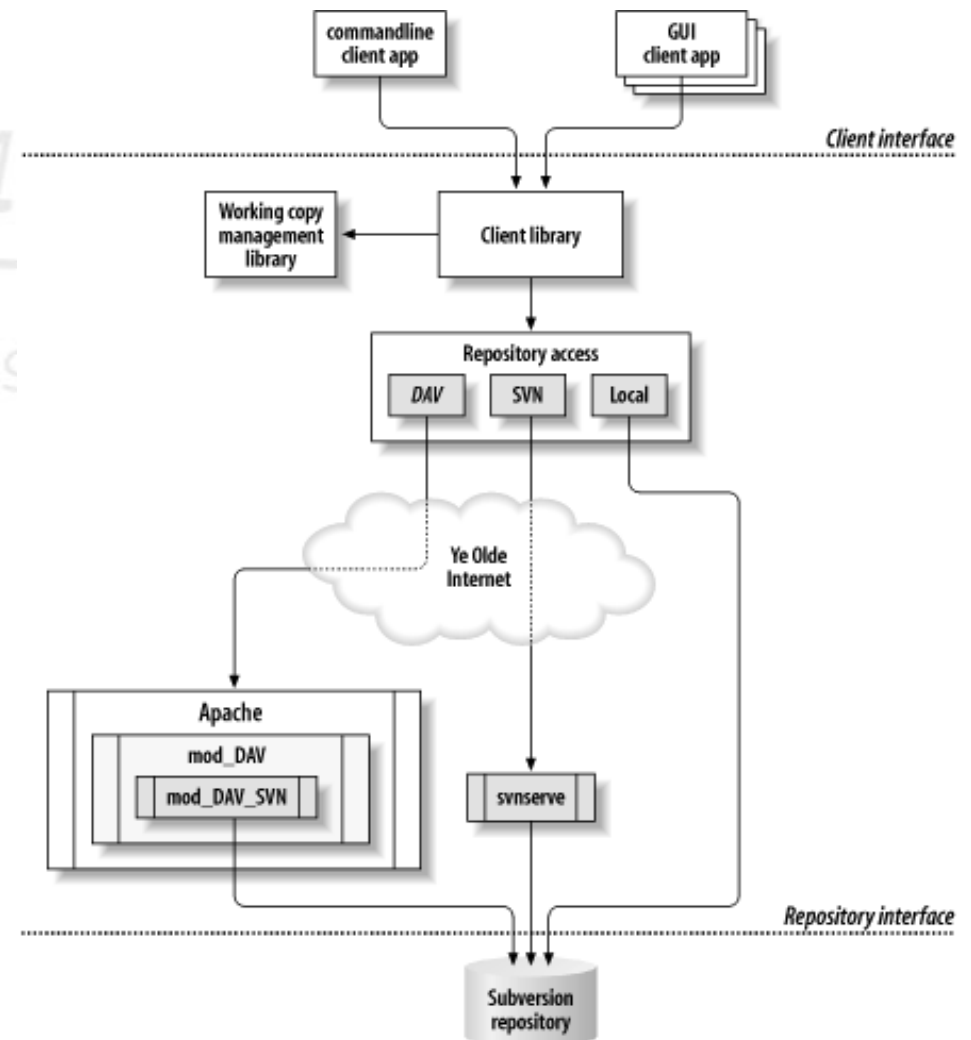
- Renombrar, copiar y mover ficheros y directorios sin pérdida del histórico.
- Commits atómicos.
- Implementación de tres tipos de acceso:
 - Local
 - Stand-alone
 - Apache + webDAV
- Mejorado el sistema de permisos.
- Reducción del riesgo de vulnerabilidades por sus distintos RA (repository access).
- Integración con Project Software Manager (Por ejemplo: trac).
- Grandes grupos de desarrollo como Apache Group o KDE están migrando sus sistemas a SVN.

► Clientes e integración con IDEs

- Para Concurrent Version Control (CVS):
 - WinCVS: <http://www.wincvs.org>
 - Eclipse (integrado)
 - Kdevelop (integrado)
 - TortoiseCVS: <http://www.tortoisecvs.org>
 - Cervisia: <http://www.kde.org/apps/cervisia>
- Para Subversión:
 - TortoiseSVN: <http://tortoisesvn.tigris.org>
 - Plugin para Eclipse (subclipse)

► Ejemplo práctico: Descripción del sistema

- Servidor Subversión configurado con webDAV como método de acceso al repositorio (RA).
- Nombre del repositorio: “seminario”.
- Existe un usuario “test” con acceso de lectura y escritura al repositorio “seminario”.



► Referencias

- Subversión (Project home): <http://subversion.tigris.org>
- Concurrent Version System: <http://subversion.tigris.org>
- Arch: <http://gnuarch.org>
- Version Control with Subversion: <http://svnbook.red-bean.com>
- BetterSCMInitiative: <http://better-scm.berlios.de>





► ¿Alguna pregunta?

Sistemas de Control de Versiones

*CVS Subversion Arch Aegis BitKeeper
Monotone Vesta Superversion
SVK*

¿Alguna pregunta?





► Agradecimientos



MUCHAS GRACIAS