



Free Webinar: Subversion in 2010 and Beyond

Audio: Use your microphone and speakers (VoIP) or call in using your telephone.

Dial: 312-878-0211
Access Code: 167-683-393
Audio PIN: 99



Hyrum Wright
President, Subversion Corporation
WANdisco



Phillip Martin
Lead Developer,
Subversion Wrkg Copy
WANdisco



Julian Foad
Lead Developer,
Obliterate
WANdisco



Dave Brown
Lead Developer, SVN J
WANdisco



Jim Campigli
VP Product Marketing
WANdisco

Which Subversion protocol are you using?

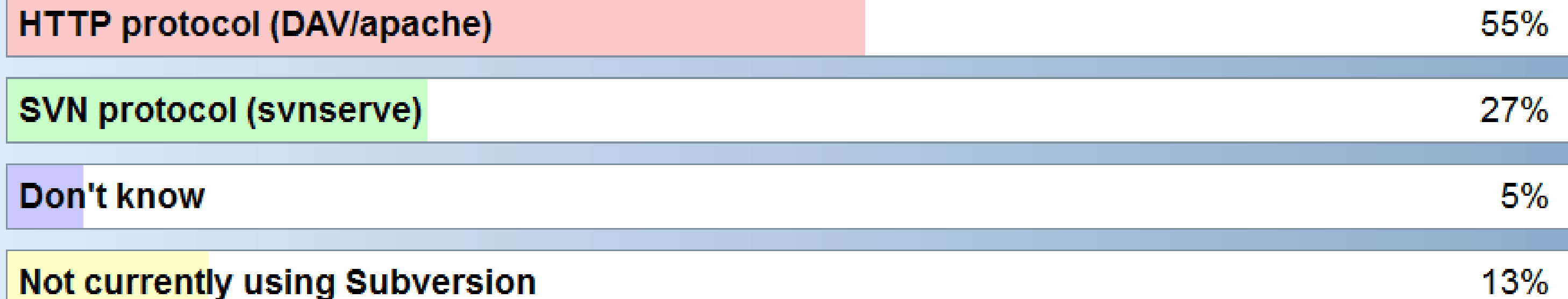
Please select one:

- HTTP protocol (DAV/apache)
- SVN protocol (svnserve)
- Don't know
- Not currently using Subversion

Submit

Which Subversion protocol are you using?

Poll Results (single answer required):



Subversion 1.7 & HTTPv2

Hyrum K. Wright

Director of Open Source, WANdisco
President, Subversion Corporation



Introduction

- HTTPv2
- Purpose
- Benefits
- Compatibility
- Subversion 1.7
- Features
- Release Process
- How you can help

- History
 - Subversion initially standardized on Apache and WebDAV/DeltaV
 - DeltaV turned out to be complex and inefficient
 - DeltaV was not widely deployed outside Subversion
 - The result is a large performance penalty
- Increase performance of Subversion's WebDAV layer
 - Remove **PROPFIND** discovery turnarounds

HTTPv2 Benefits

- Reduced number of roundtrips
 - Useful on high-latency links (such as intercontinental links)
- Easier for future developers to extend
- Better caching with web proxies

HTTPv2 Compatibility

- All 1.x servers and 1.y clients will continue to be compatible
- Older clients with new servers, and vice versa, will continue to use the older slower protocol

Subversion 1.7

- Features
 - Working Copy - Next Generation (WC-NG)
 - HTTPv2
 - svn patch
 - preliminary obliterate support
 - plenty of bugfixes and microfeatures
- Community involvement for beta program
 - Better real world testing
- First major release as part of the Apache Software Foundation

Subversion 1.7 Timeline

- No firm date, though we expect it to be during the first half of this year
 - Unique open source development strategy dictates “it’s done when it’s done”
 - You can help by donating developer time and resources
- Community-driven release process
 - Beta program
 - Release candidates
 - General release
- Caveat
 - Any forward-looking feature prognostications and release statements are purely speculative, and may change at any time.

Which client do you use most often?

Please select one:

- Subversion command line**
- Subclipse**
- Tortoise SVN**
- SmartSVN**
- SVNKit**

Submit

QUICKPOLL

Which client do you use most often?

Poll Results (single answer required):

Subversion command line	26%
Subclipse	16%
Tortoise SVN	50%
SmartSVN	8%
SVNKit	1%

Subversion WC-NG

The Next Generation Working Copy

Philip Martin

Lead Developer

Topics

- What is libsvn_wc?
- Old design and problems
- New design
- Benefits
- Current status

What is libsvn_wc?

- API and library to handle working copies
- Used by
 - Subversion command line client
 - Other clients: TortoiseSVN, Subclipse
 - Language bindings
- Convoluted Code

Old Design

- Meta-data stored in each directory
- One lock file per directory
- Deleted directories remain in the working copy
- Entries cached in memory
- Log files
- Short-lived database views

New Design

- Centralised meta-data
- SQLite database
- Workqueues instead of log files
- Properties in the database
- SHA1 indexed text-base storage
- Backward compatible

Benefits

- Faster locking
- Able to remove deleted directories
- API more friendly to GUIs
- Less reliance on a case-sensitive filesystem
- Shared text-bases

Current Status

- One SQLite database per directory
- Most in-memory entry caching removed
- Workqueues instead of log files
- Properties not yet in SQLite database
- Text-base not yet in SHA1 index
- Regression tests

Obliterating History in Subversion

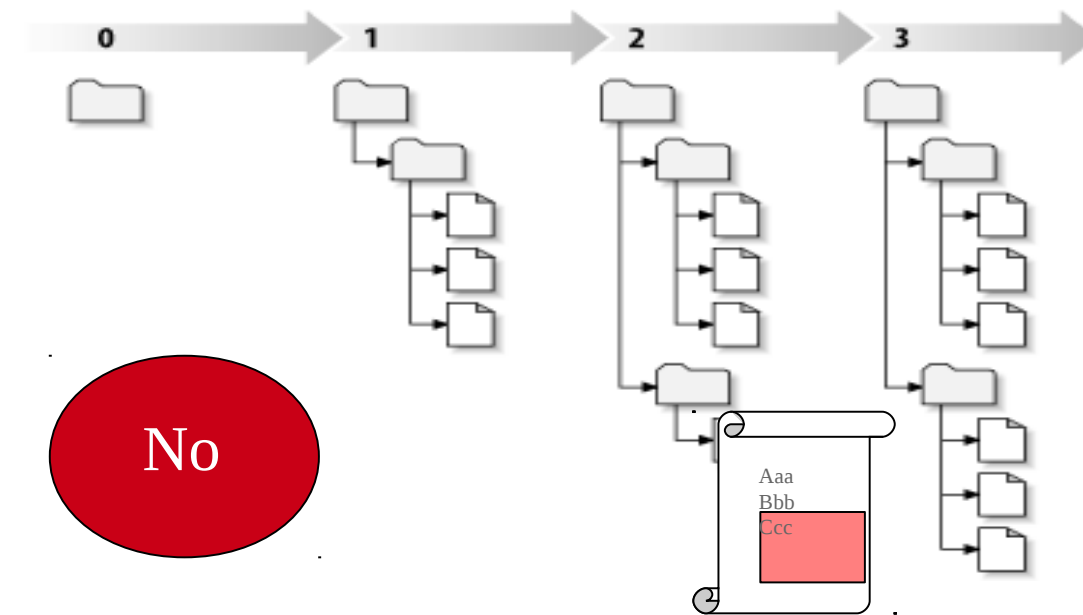
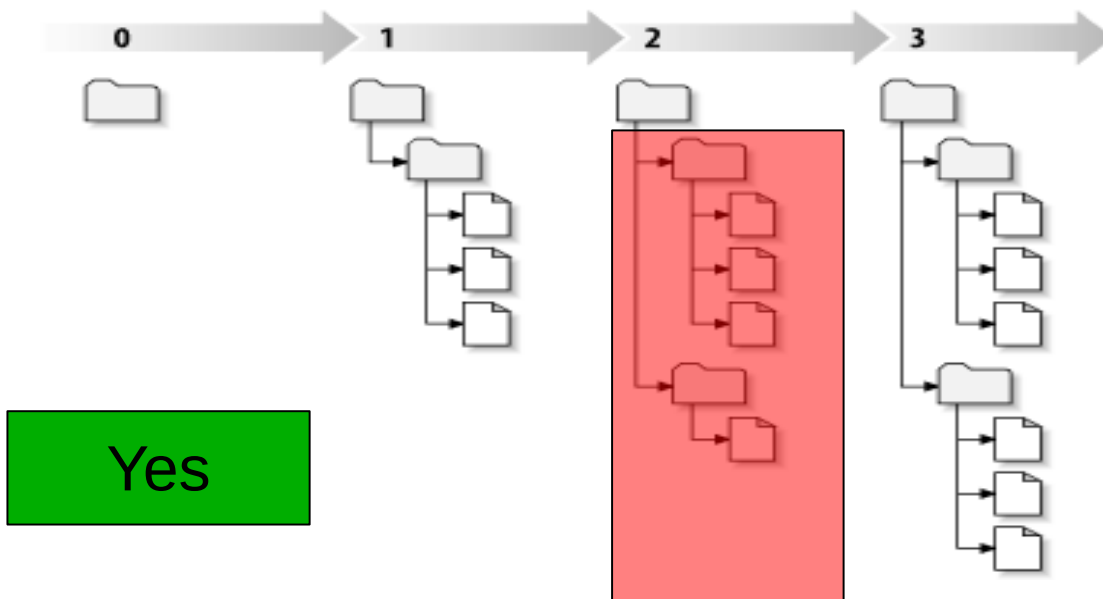
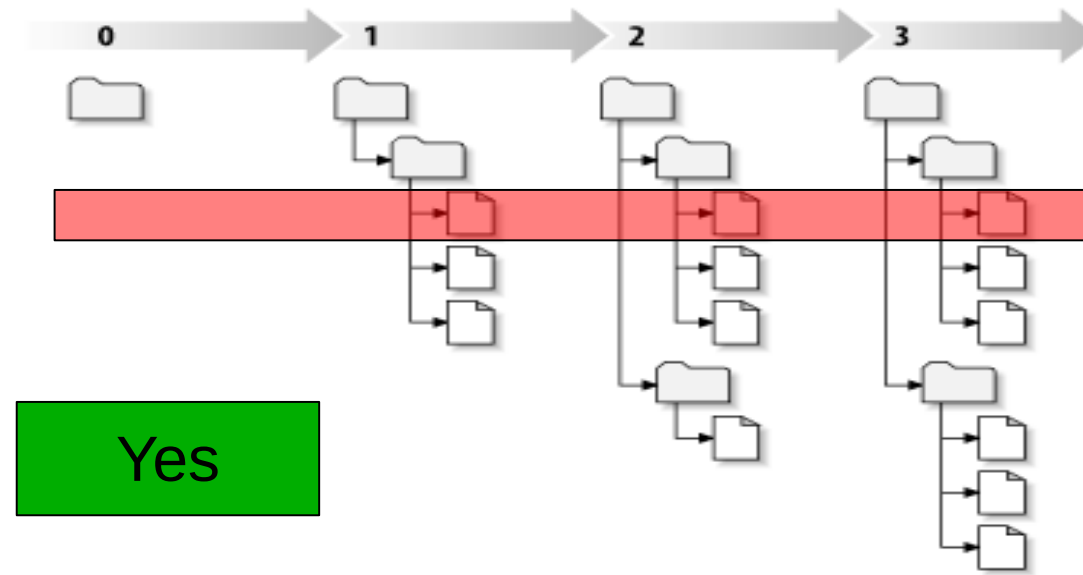
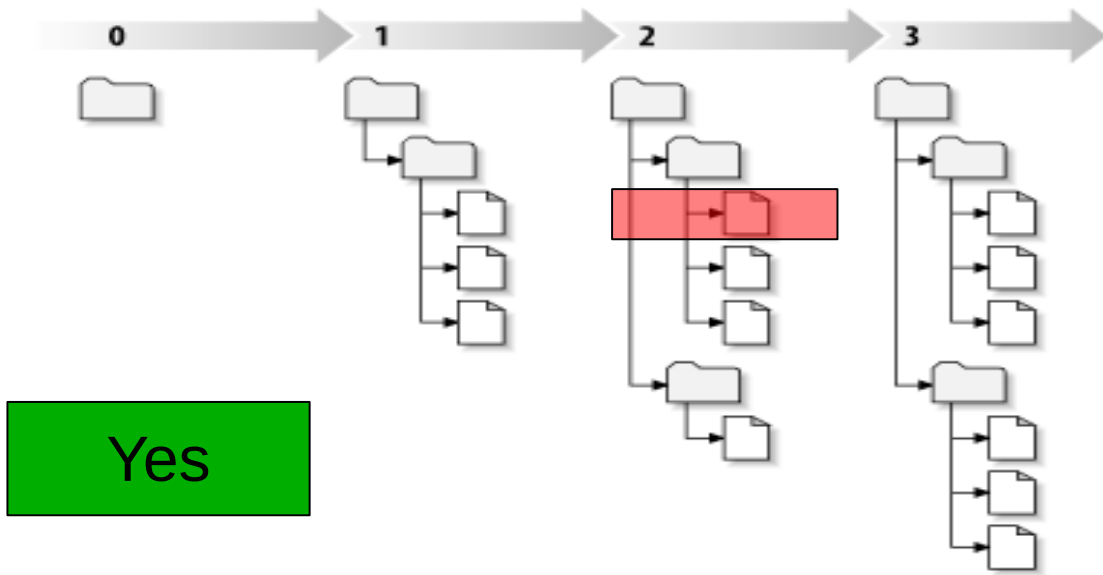
Julian Foad

Senior Subversion Committer

Obliterate – what for?

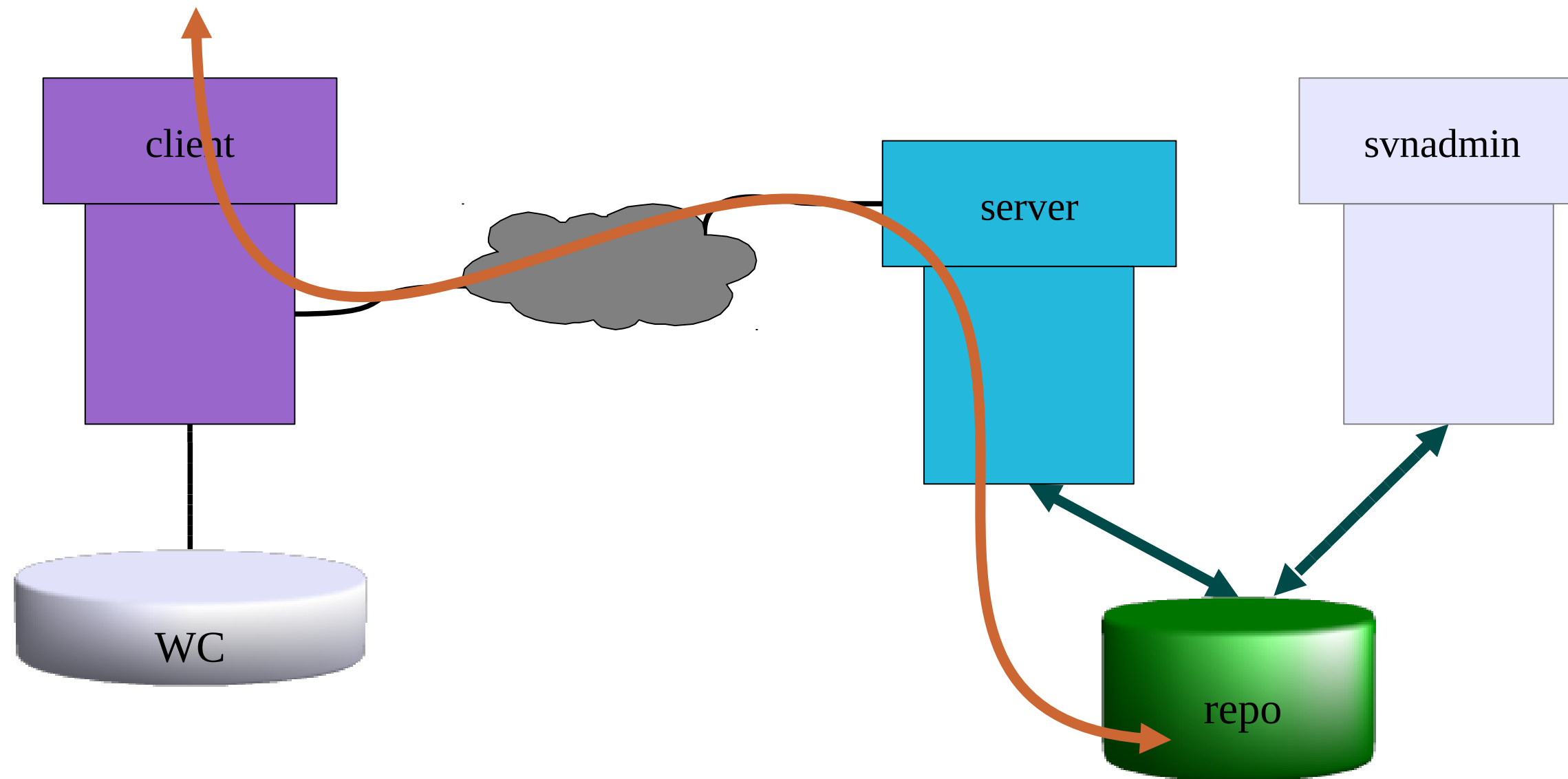
- ▶ User scenarios:
 - Hide some private data that I committed by mistake
 - Remove a huge unwanted file that I committed by mistake
 - one file in one revision
- ▶ Admin scenarios:
 - Remove an obsolete project from the repository to recover server disk space
 - many files in many revisions

Granularity



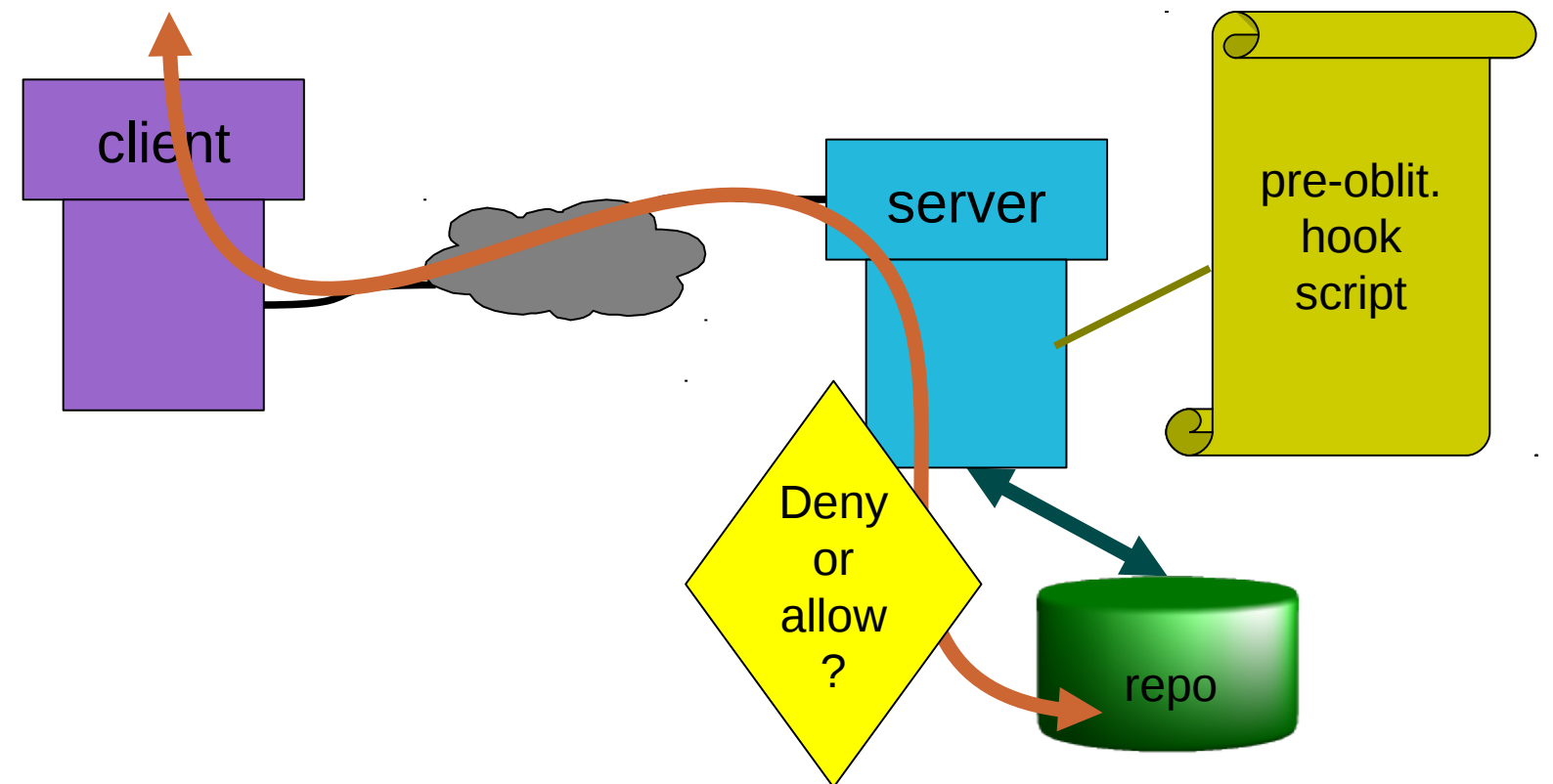
Server or Client

```
svn obliterate ...
```



Authorization

- ▶ Pre-obliterate hook
 - Control by user names
 - Control by paths
 - Obliteration is completely **DISABLED** until hook is installed
- ▶ Inputs
 - User name
 - Obliteration Set (PATH@REV ...)
- ▶ Result
 - Allow or Deny (with optional error message)



Audit Trail

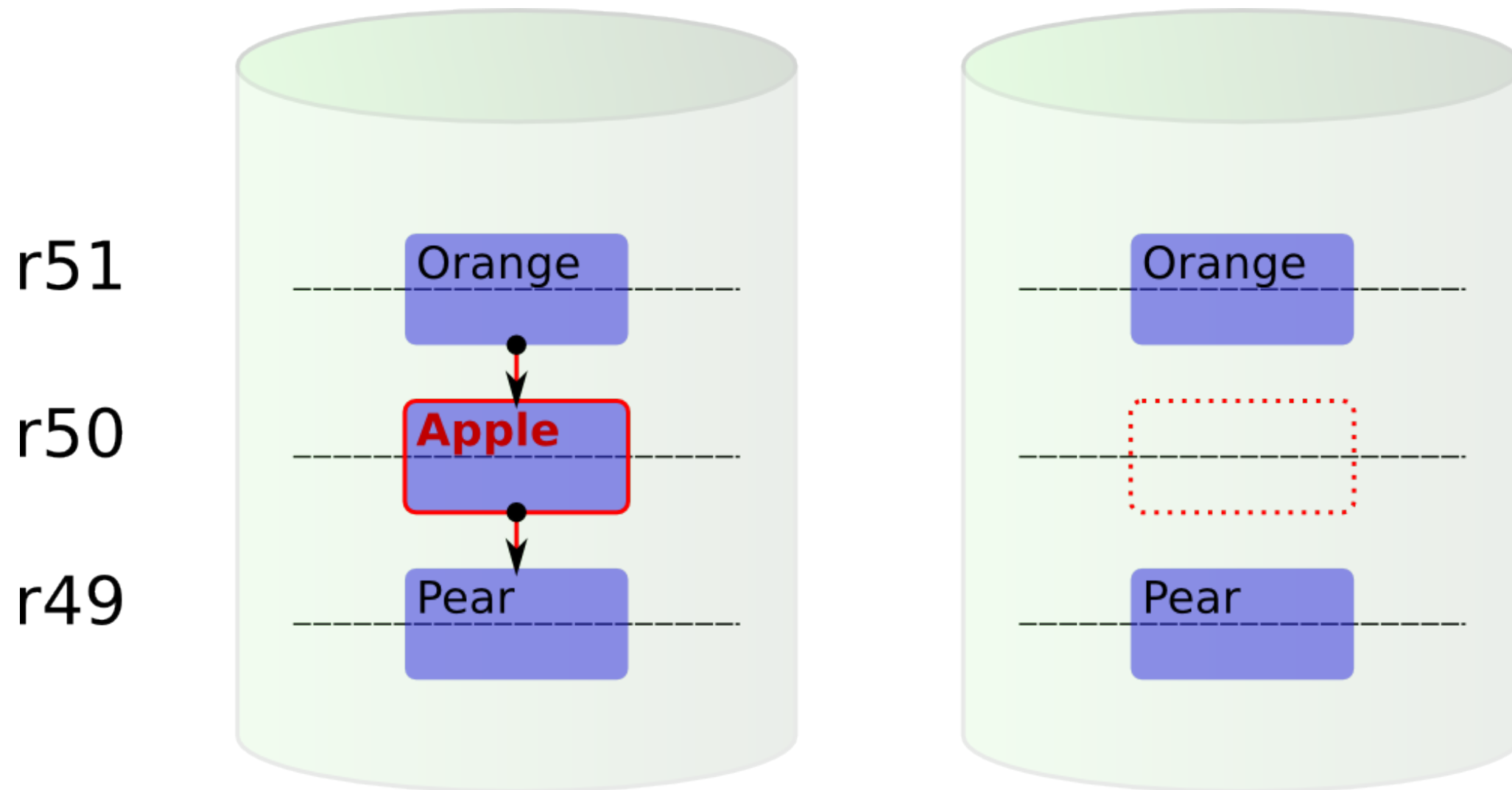
- ▶ Obliterate alters history, it does not create more history
- ▶ Obliterate makes existing history-altering tasks easier and quicker
- ▶ There will be an audit trail
 - who, when, what
- ▶ This info will be stored
 - On the server – in a dedicated log file
 - (Maybe) In a user-visible place – such as a rev-prop
- ▶ In a user-visible place, if paths are sensitive, it can just say “within directory D” or “within revision R”

Challenges

- ▶ FSFS / BDB
 - FSFS needs a successors table
- ▶ Large span of revisions – efficiency
- ▶ Working copy behaviour
- ▶ Local (file://) vs. remote (svn://, http://)
- ▶ Adjusting the future history (next slide)

Adjusting the future history

- ▶ Obliterate file 'F' @ r50:



Milestones

- ▶ M1: simplest possible case – youngest revision of a file
- ▶ M2: any node in any revision
- ▶ M3: FSFS; large range of revisions
- ▶ Unplanned: BDB, GUI

More detail:

<http://svn.apache.org/repos/asf/subversion/trunk/notes/obliterate/plan-milestones.html>

or

http://subversion.wandisco.com/wiki/index.php/Svn_Obliterate_Project_Plan

QUICKPOLL

In your repository, would you expect "svn obliterate" to be used:

Please select one:

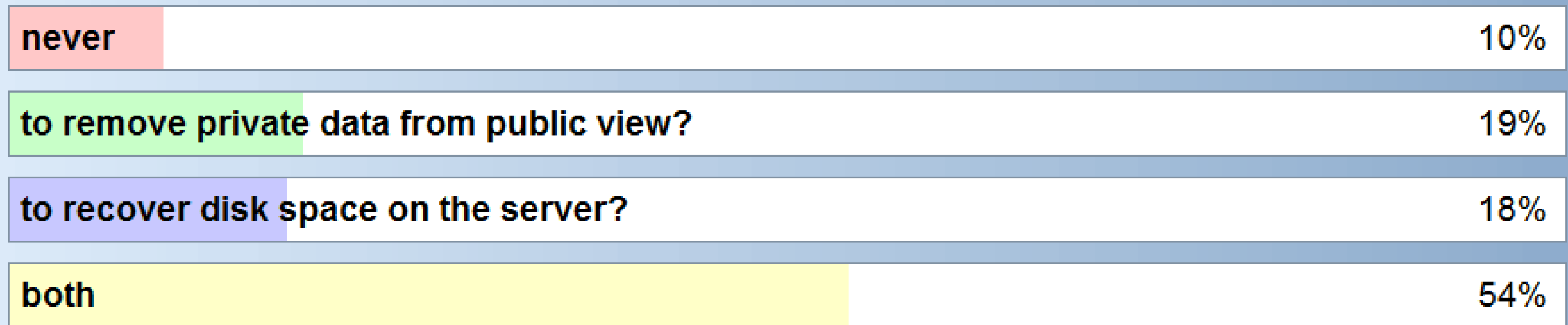
- never**
- to remove private data from public view?**
- to recover disk space on the server?**
- both**

Submit

QUICKPOLL

In your repository, would you expect "svn obliterate" to be used:

Poll Results (single answer required):



SubversionJ

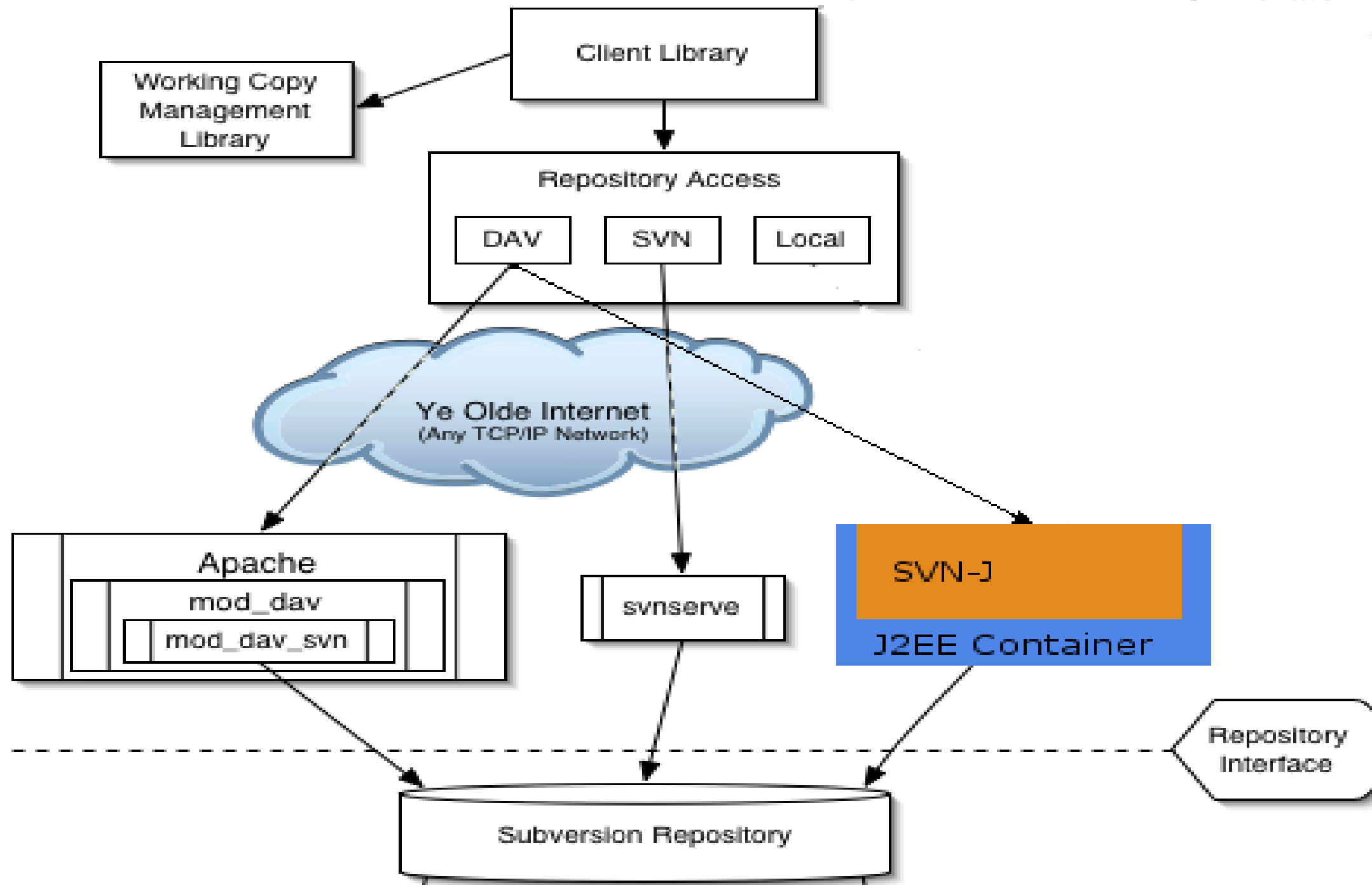
Server-side Java for Subversion

Dave Brown

SubversionJ: Server-side Subversion bindings in Java

- Deploy as J2EE-standard web application
 - Tomcat, Jetty, Weblogic, etc.
 - Instead of mod_dav_svn in Apache httpd
- Bridge to C API's via JNI
 - libsvn_repos
 - libsvn_fs
 - Java Delta Editor (svn_delta_editor_t)
 - ...others as needed?
- Re-use!

SubversionJ: == 1000 words...



SubversionJ: what's in it for you?

- Easy drop-in to existing java deployments
- Analogous to existing client bindings
 - Interesting client-based apps in java, python, ruby
 - SubversionJ enables the same on the server
 - “Bug-for-bug” compatible with the rest of Subversion
- Server extensions for java shops
 - Authorization
 - Replication !
 - Storage (?)

SubversionJ: Goals

- Build great products
 - Java on the server helps us
 - More Subversion adoption helps us
- Give users deployment option beyond Apache / httpd
 - Flexibility of Apache installation is a blessing / curse
 - Get us out of the Apache / httpd business

SubversionJ: Project Philosophy

- Stay in sync with core Subversion project
 - **Always** maintain compatibility
 - Rely only on supported Subversion API's
- Pragmatic
 - Features & API's driven by use cases
 - Minimal wheel re-invention

SubversionJ: Open Source

- Why Open Source?
 - Open more of Subversion to java developers
 - Give back to the community
- Apache 2.0 / BSD style license
 - Same as the rest of the Subversion project
 - Friendly for commercial redistribution

SubversionJ: Current Status

- Pre-alpha code
 - Read only: checkout / update working
 - Java delta editor working
 - Checkin operations are work-in-progress
- SVN 1.7 (HTTP-v2) protocol first; backward compat. Later
 - Automated unit, functional tests

SubversionJ: Availability

- Availability – soon!
- Public repository going live imminently
 - Watch <http://subversion.wandisco.com/> for announcement
- Get involved!
 - Feedback, participation welcome
- dave.brown@wandisco.com

Subversion Services
<http://wandisco.com/svn/services>

Jim Campigli
COO & VP, Product Marketing



Agenda

- Subversion Consulting Services
- Subversion Training Services
- Subversion Support Services



- Migration and Deployment Planning
 - Assess what data needs to be migrated
 - Map existing development processes to Subversion.
 - Map legacy version control system's functionality to Subversion.
- Go-Live Support
 - Remote on-call with access to your Subversion servers
 - On-site support
- Implementation Health Check
 - Full audit of your Subversion implementation



- Classes are designed to provide background and hands-on experience needed to become productive immediately.
- Developer and Administrator training available in the following formats:
 - Web-based training.
 - Public Class at one of our training centers
 - Private Class at the customer's location
 - Custom training
- Classes can be taken in conjunction with consulting services.
- Train the trainer for large organizations with dedicated training staff.

- Designed to eliminate the risk of deploying an open source solution such as Subversion.
- Our core developers ensure support tickets are resolved and fixes delivered on a timely basis.
- Subversion enhancement requests can also be submitted.
- Platinum level support includes a “Bug Buddy” assigned to your specific Subversion issues.

Technical Support	Silver	Gold	Platinum
Support Hours	8x5 Choose Pacific Time or GMT	12x5 Choose Pacific Time or GMT	24x7
Online Case Tracking	24x7	24x7	24x7
Phone Support			
E-mail Support			
Response time SLA - Top Priority [P1]	3 hrs	2 hrs	1 hr



- ▶ For more information on Subversion or to request a copy of slides used in today's presentation, please contact:
 - Email: sales@wandisco.com
 - Web: www.wandisco.com
 - Phone: 866 846 0404 (U.S.)
 - Phone: +44 (0)114 3039985 (Europe)
- Join the Subversion Community today!
 - <http://subversion.wandisco.com/>
- Learn more about Subversion services and training
 - <http://wandisco.com/svn/services>