The Subversion Project: Building a Better CVS

Paper by: Ben Collins-Sussman
Presented by: Brice Dobry
Goals of Subversion

• Design a functional replacement for CVS
• Fix the flaw’s in CVS’s design
• Maintain similar feel to make transition easy for existing CVS users
Nice Features of SVN

- Copy/rename files while maintaining history
- Atomic commit (all or nothing)
- Hash-table associated with files for storing meta-data
- Much cleaner code than CVS
- Faster network access
Subversion’s Design
Subversion Filesystem

- Sits on top of Berkely DB
- C-API interacts with DB and simulates a filesystem
- Files can be read from and written to just like any other filesystem only all history is saved
Subversion Filesystem

- With each revision, a new tree is created and labeled with a global revision number
- Files that changed are rewritten
- Older versions are stored as differences against the latest version
- New tree points to previous revision copies of any unchanged files
Network Layer

- Apache/WebDAV
- svnserv
- Local disk
Network Layer

- Apache/WebDAV
  - “Why reinvent the wheel?”
  - Take advantage of all of Apache’s already proven features
  - WebDAV allows reads, writes and versioning of files over the web
Network Layer

- svnservice
  - Lightweight server
  - Uses custom protocol over TCP/IP
  - Can be tunnelled through ssh
Network Layer

<table>
<thead>
<tr>
<th>Feature</th>
<th>Apache + mod_dav_svn</th>
<th>svnservice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication options</td>
<td>HTTP(G) basic auth, X.509 certificates, LDAP, NTLM, or any other mechanism available to Apache httpd</td>
<td>CRAM-MD5 or SSH</td>
</tr>
<tr>
<td>User account options</td>
<td>private 'users' file</td>
<td>private 'users' file, or existing system (SSH) accounts</td>
</tr>
<tr>
<td>Authorization options</td>
<td>blanket read/write access, or per-directory access control</td>
<td>blanket read/write access</td>
</tr>
<tr>
<td>Encryption</td>
<td>via optional SSL</td>
<td>via optional SSH tunnel</td>
</tr>
<tr>
<td>Interoperability</td>
<td>partially usable by other WebDAV clients</td>
<td>not interoperable</td>
</tr>
<tr>
<td>Web viewing</td>
<td>limited built-in support, or via 3rd-party tools such as ViewCVS</td>
<td>via 3rd-party tools such as ViewCVS</td>
</tr>
<tr>
<td>Speed</td>
<td>somewhat slower</td>
<td>somewhat faster</td>
</tr>
<tr>
<td>Initial setup</td>
<td>somewhat complex</td>
<td>fairly simple</td>
</tr>
</tbody>
</table>

Taken from http://svnbook.red-bean.com
Network Layer

- Local Disk
  - Access to the repository can also be done by simply accessing the local disk
Repository Access Layer

- Hides the underlying layers
- Client does not have to care whether the repository is on the local disk, served by Apache or by svnserv
- Major advantage over CVS
Client Libraries

- Working copy library stores administrative information in SVN subdirectory
  - Current state of the working copy
  - Location of file properties (metadata)
  - Local copies of unchanged versions of each file
  - Authentication data
Client Libraries

- Subversion client library
  - Middleman between working copy library and the repository access layer
Client Applications

- Standard command line client
- GUI's such as kdesvn, svnX, etc.
- IDE's such as Xcode, Subclipse, etc.
Evaluating Subversion

• Author says, “If in three years Subversion is widely presumed to be the standard SCM system in the Open Source community, then the project will have succeeded.” (2001)

• Currently, it is becoming more and more widely used but still has not surpassed CVS

• Probably the most popular alternative to CVS
Evaluating Subversion

- Many major projects use Subversion (Apache, KDE, Gnome, GCC, Python,...)
- Google Code uses Subversion exclusively
- SourceForge provides Subversion as an option for its hosted projects
Sources

- “The Subversion Project: Building a Better CVS” by Ben Collins-Sussman
- “Subversion (software)” - Wikipedia