

# Tracking FreeBSD Customizations with a Local Mercurial Branch

Giorgos Keramidas

[keramida@FreeBSD.org](mailto:keramida@FreeBSD.org)

# What is FreeBSD?

- Historical roots at the first BSD releases
- BSD UNIX, free software
  - Open and documented release engineering process
  - Fairly complete set of base system tools
- 2-clause BSD license
  - Attractive to commercial entities too
  - Most of non-BSD code in separate subdirs

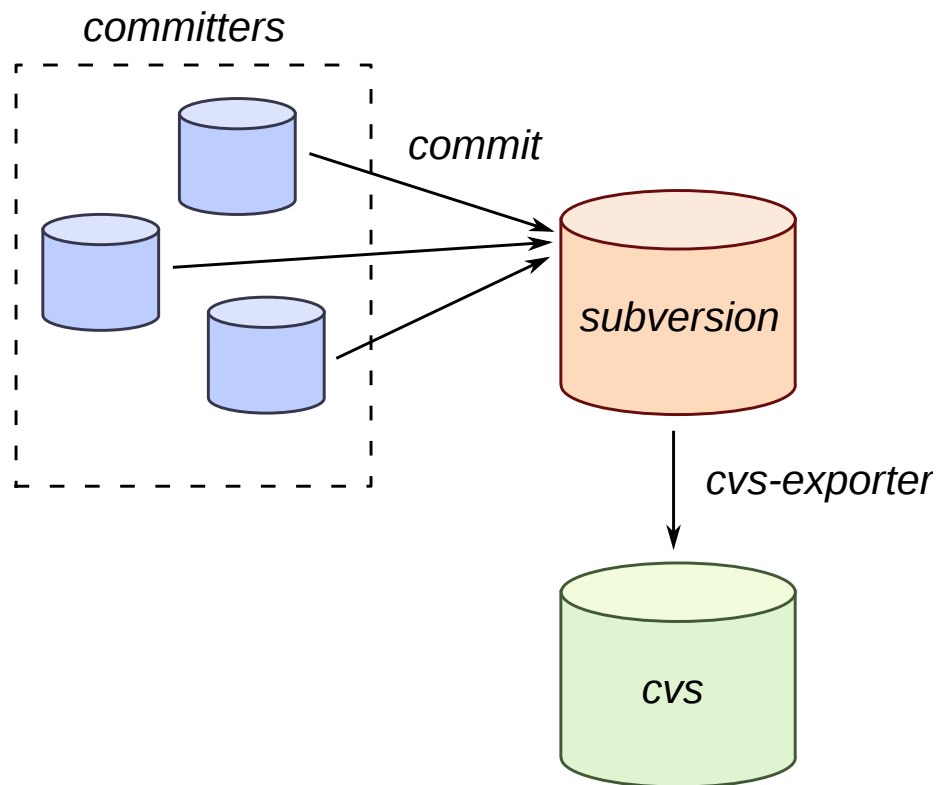
# Who Would Want to Track FreeBSD Sources

<i>Tracker</i>	<i>Project</i>
Developers	Short or long term projects
Academic institutions	Research projects or other long duration work
Commercial vendors	Embedding all or parts of BSD
Everyone else	Customization is in our nature

# Availability of the FreeBSD Sources

- CVS repository since 1993
- Subversion repository since 2008
- Single repository for all sources
  - 'One stop' place for a full UNIX-like system
  - Centralized, controlled updates of main source tree

# The FreeBSD Repository



# Common Source Tracking Scenarios

- Projects aiming to be included in BSD itself:
  - Develop a one-off patch for a bug fix, including review cycle
  - Fork BSD for long-term project (new features, research, other experimentation)
- Projects using BSD as a 'thirdparty' source tree:
  - Personal customizations
  - Team that embeds parts of BSD in other project(s)
  - Building custom BSD images

# Who Can Customize FreeBSD?

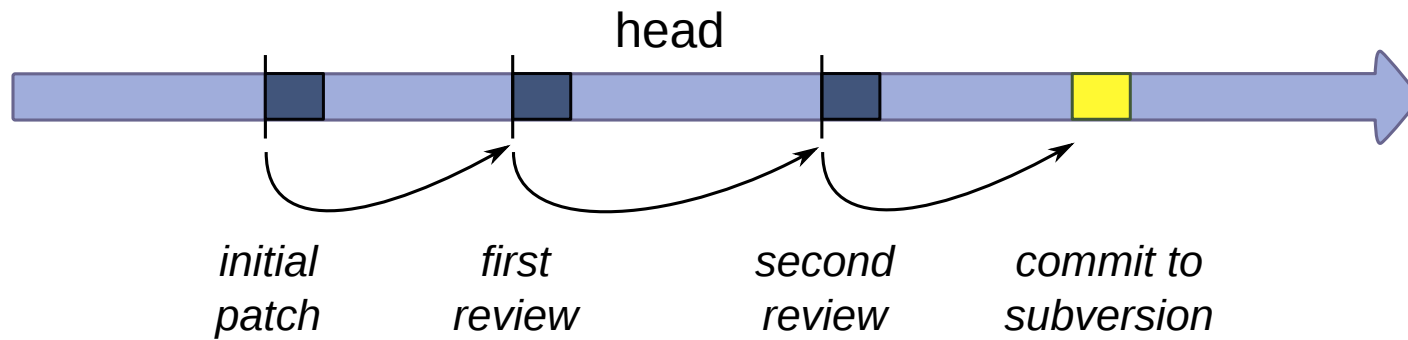
- Everyone! We mean that... everyone!
  - Internet Service Providers
  - Hosting companies, UNIX as a Service Providers
  - Universities & research institutions
  - Companies looking for 'reference implementations'
  - Developers themselves, for personal reasons

# Source Tracking Requirements

- A simple patch being merged or rebased on top of a branch
- A series of patches customizing many bits of a BSD branch
- Repeated merges of more complex updates and full features

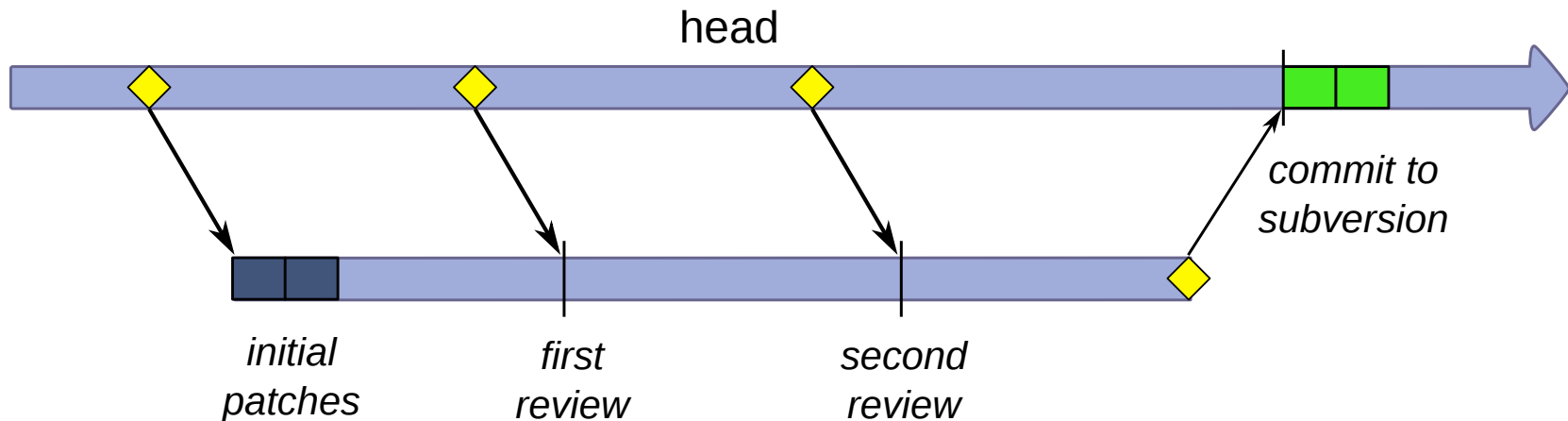


# Developing a Single Patch



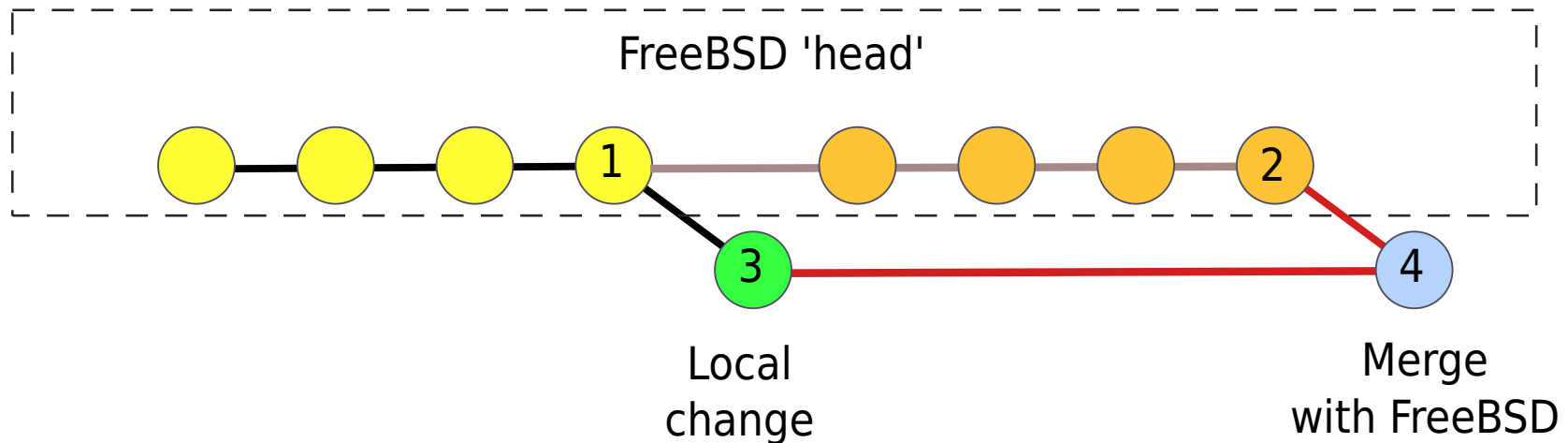
- There can be more than one review cycle
- The patch has to be “forward merged” until accepted
- In the meantime the *head* branch moves along
- Relatively easy, even without Mercurial

# A More Realistic Picture of Repeated Merges



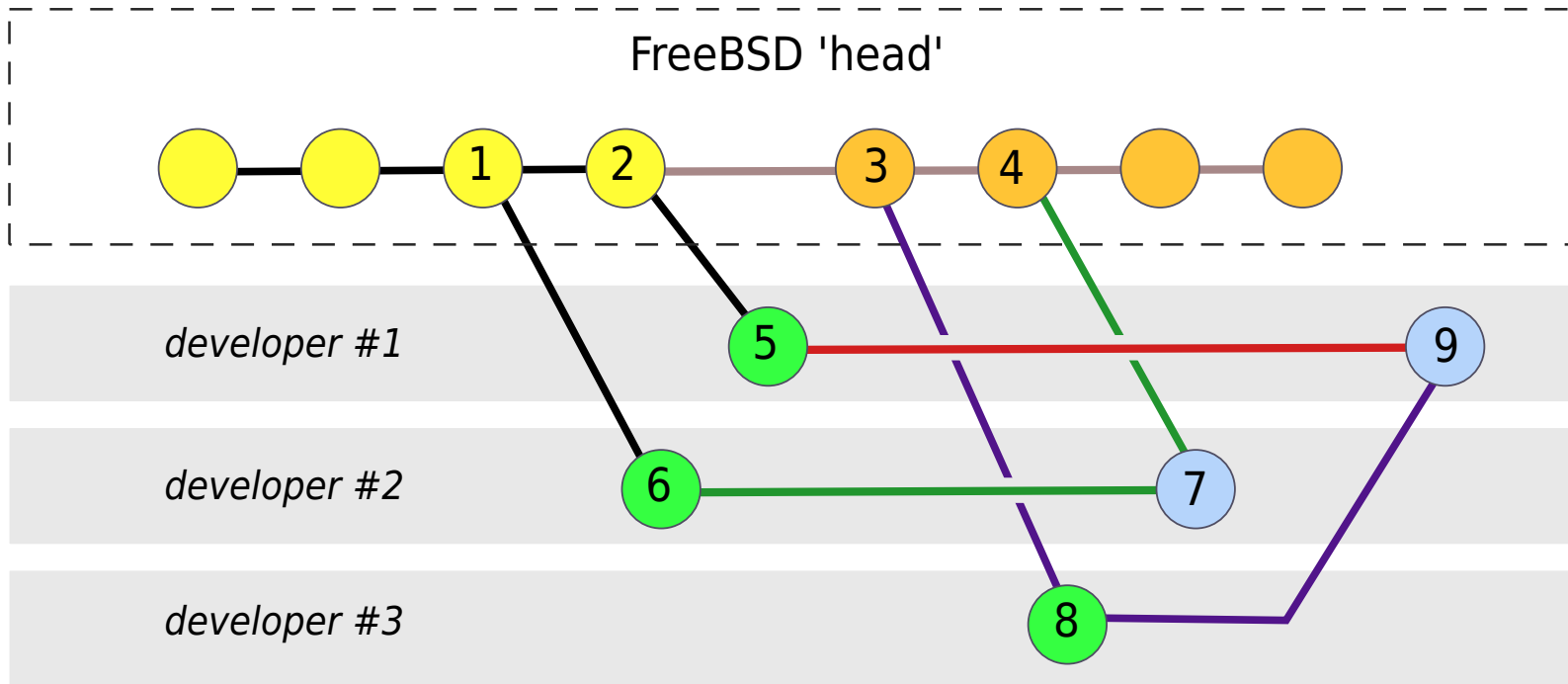
- Even a single local change effectively 'forks' FreeBSD
- Creating a local 'branch' with its own local patchset
- It has its own, separate merge history with the FreeBSD 'vendor' code it derives from
- When the patches are approved by FreeBSD committers, their final versions hit subversion as normal commits

# This is Exactly How a Modern DVCS Treats History



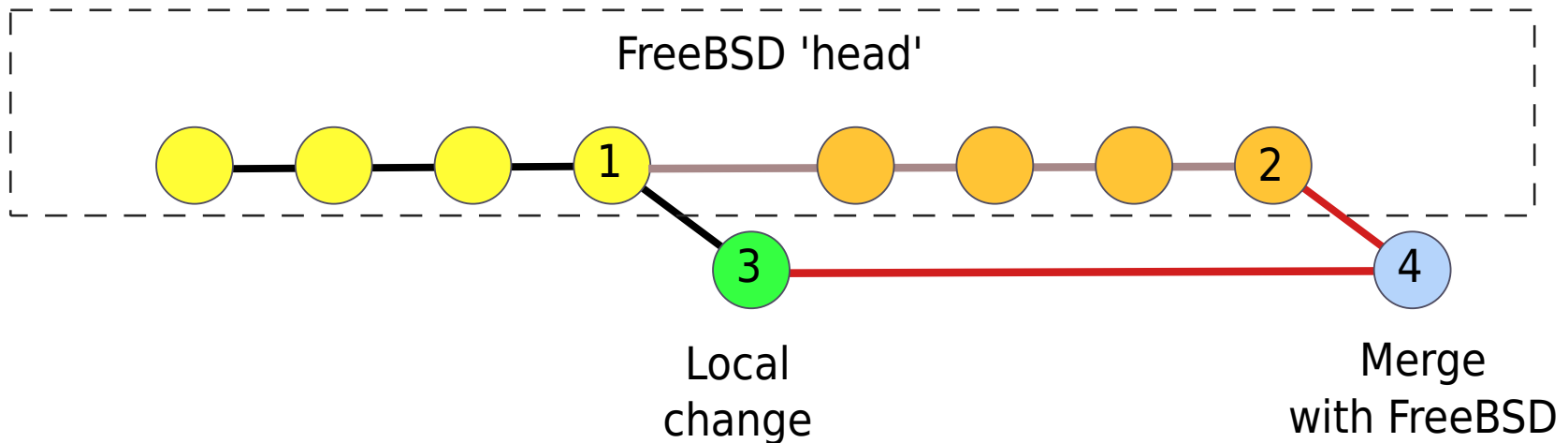
- The FreeBSD 'head' commit line may be snapshots or the svn commits themselves
- Local patch **r3** committed on top of FreeBSD version **r1**
- New versions up to **r2** merged later to create local revision **r4**
- Merge history clearly visible in the Mercurial repository
- Diffs between arbitrary revisions (**r2...r4**) are *a lot* faster than svn

# A Multi-Person Team Making FreeBSD Changes



- FreeBSD history is essentially a 'vendor' branch (1, 2, 3, 4, ...)
- Each developer works separately on their changes (5, 6, 8)
- Developers #1 and #3 merge with each other (9)
- No need to go through *the* repository to work with each other

# Extracting the Local Changes as a Patch



```
% cd /work/freebsd/mybranch
```

```
% hg shortlog --limit 3
```

```
87c12864cdc2 2010-02-02 23:25 keramida: Merge from head
5b786e3982e0 2010-02-02 20:38 qingli: Some of the existing ppp and...
e847dd4495d8 2010-02-02 19:44 gavin: Bump .Dd, forgotten in r203393
```

```
% time hg diff -r 5b786e3982e0:tip | diffstat -p1 | tail -1
 77 files changed, 18433 insertions(+), 44 deletions(-)
   3.106 real        1.769 user        0.726 sys
```

# Local BSD History

- Snapshot based (e.g. tarball drops every now and then)
  - Pros: very easy to setup, simple cron job, relatively easy with svn
  - Cons: full freebsd commit history not visible / commit date sorting not enough to get atomic changesets / merging large code 'drops' not as easy / no file-rename tracking at all or guessing with "hg addremove -similarity"
- Distinct Subversion changesets
  - Pros: fast incremental conversion support by hg itself, multiple plugins & converters available (convert extension, cvs2hg, hgsvn, etc.)
  - Cons: slightly more involved process, full history takes more space, not easy to get expanded \$FreeBSD\$ tags

## Keeping a Local Subversion Mirror

- Initial 'seed' tarball (available on request)
- svnsync for incremental updates:  
% `svnsync sync file:///repos/freebsd/base`
- Mirror size = 4.4 GB for src mirror
- Daily svnsync often takes less than 1 minute
- May not be strictly necessary for importing small bits of BSD
- Converting parts of src/ over the web also possible

# Full Local FreeBSD History

- Almost as easy as *one* command:

```
% hg convert file:///svnrepo/path/head \  
/work/freebsd/head
```

- History of 'head' since 2008-01-01 around 253 MB
- Full history of 'head' since 1993 around 450 MB
- Still smaller by about 50% than a checkout of a single svn revision of the full 'head' branch
- But is really *all* history useful for every day work?
  - Probably not for short-lived patches
  - Almost certainly not for head-only work



# Useful Conversion Tricks for Subversion

- Author names to emails
- Converting only one branch
- Converting only *after* a certain changeset (partial history)
- Converting only parts of head/ (partial branch)

## Author Names to Committers Emails

- Map svn usernames to emails through the 'authormap' option
- Most of the names available in doc/ tree as part of 'doc/en\_US.ISO8859-1/share/sgml/authors.ent'
- Sample authormap:  

```
philip=Philip Paeps <philip@FreeBSD.org>  
keramida=Giorgos Keramidas <keramida@FreeBSD.org>  
brooks=Brooks Davis <brooks@FreeBSD.org>  
erwin=Erwin Lansing <erwin@FreeBSD.org>
```
- Pros: Local usernames will not be confused with 'upstream' FreeBSD committers.
- Cons: Some old names may be lost forever. A bit of archaeological research may be necessary to recover all FreeBSD committer names.

## Converting Only One Branch

Empty branches/tags names → only final subdirectory converted to a Mercurial repository:

```
% hg convert \  
    --config convert.svn.trunk='head' \  
    --config convert.svn.branches='' \  
    --config convert.svn.tags='' \  
file:///svnrepo/freebsd/base \  
/work/freebsd/head
```

## Converting Only After A Specific Revision

The 'convert' extension accepts a 'convert.svn.startrev' option, e.g. 'head' branch, only history since 2008-01-01:

```
% hg convert \  
    --config convert.svn.startrev='175021' \  
    --config convert.svn.trunk='head' \  
    --config convert.svn.branches='' \  
    --config convert.svn.tags='' \  
file:///svnrepo/freebsd/base \  
/work/freebsd/head
```

## Selectively Converting Only Parts of head/

A 'filemap' can be used to include or exclude specific bits of the full FreeBSD tree (e.g. to completely strip out all GPL code).

```
% hg convert --filemap /tmp/my.map [options] args
```

where /tmp/my.map includes (or excludes) only particular bits of the full tree:

```
include lib/libc  
rename lib/libc .
```

# Conclusions

- Full changesets are rather easy to get out of Subversion
- Local Mercurial clones have nice merge tracking that helps *a lot* with repeated merges
- With the *convert* extension you choose what you will pull out of Subversion, where to pull it, and how often
- You don't need direct commit access to the main Subversion repository to branch your own copy & hack merrily away
- We've been experimenting with Mercurial in the FreeBSD team and we can help you if you have questions about using it for local work