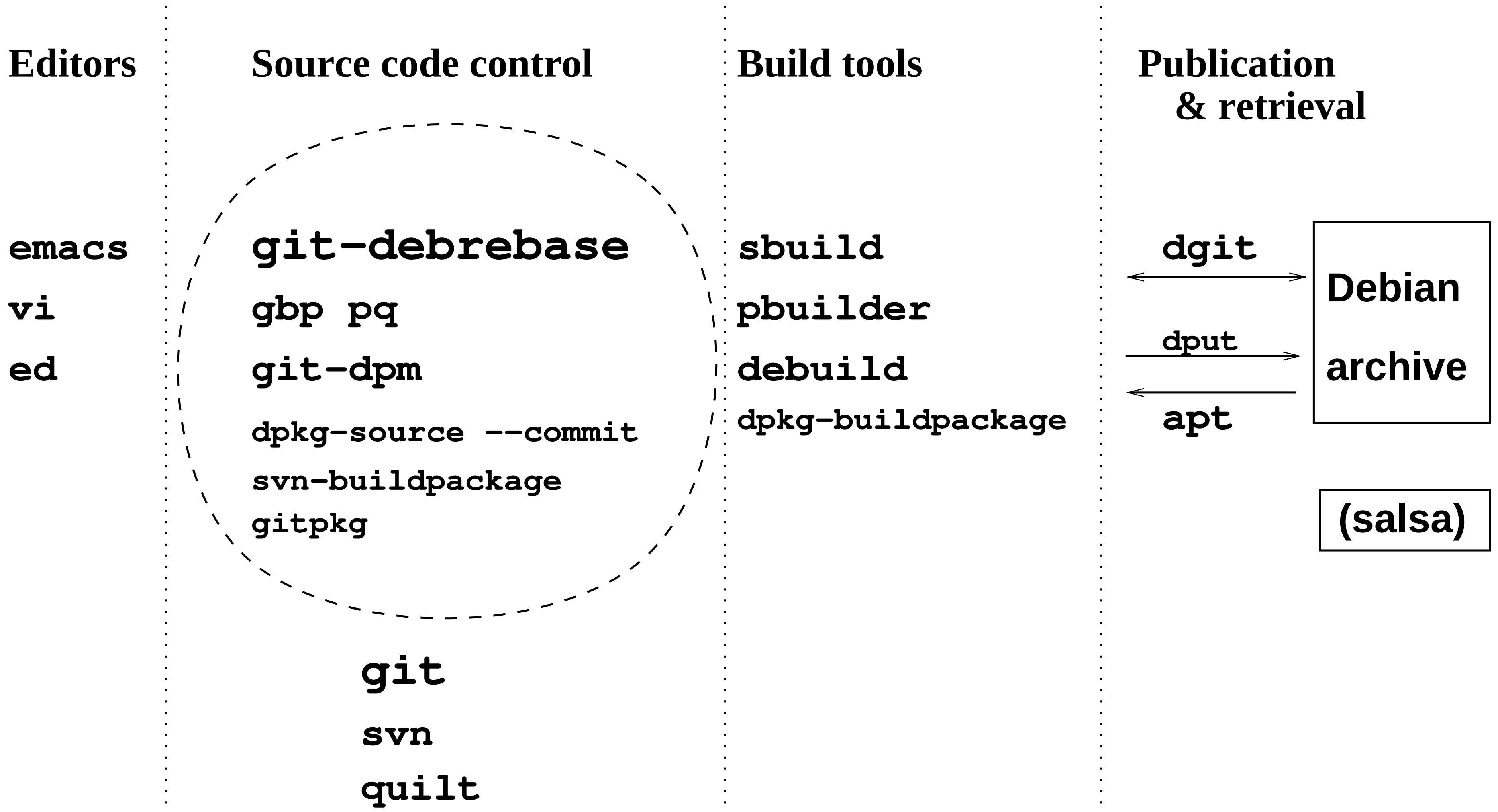
git−debrebase − context



Editors

emacs

vi

ed

Source code control

git−debrebase

gbp pq

git−dpm

dpkg−source --commit

svn−buildpackage

gitpkg

git

svn

quilt

Build tools

sbuild

pbuilder

debuild

dpkg−buildpackage

Publication & retrieval

dgit

dput

apt

Debian archive

(salsa)

Ian Jackson & Sean Whitton
Debconf 2018 Hsinchu

git−debrebase − features (1/3)

Standard `git−rebase` workflow
Delta queue editing is immediate and easy

... unlike any other tool

May always `git−cherry−pick` from upstream
May always `git−commit −m fixup!`
May make "mixed" commits

... unlike any other delta queue workflow

No in−tree metadata

... unlike `git−dpm`

git−debrebase − features (2/3)

No need to switch branches

Tree is always `dpkg−buildpackage`’able

... unlike `gpb pq` and `git−dpm`

Working tree is never dirtied

- **by patch application** **... unlike `gbp pq`**
- **by metadata conflicts**
- **by patch export** **... unlike `gbp pq`**

No reading diffs of diffs **... unlike `gbp pq`**

git−debrebase − features (3/3)

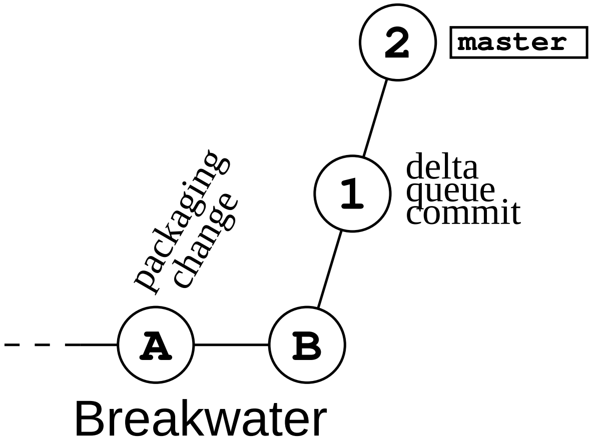
`git−blame` (and `git−log -- file`, `git−log −G`) work
Tree makes sense to upstreams & users

... like `dgit−maint−merge(7)` and `git−dpm`

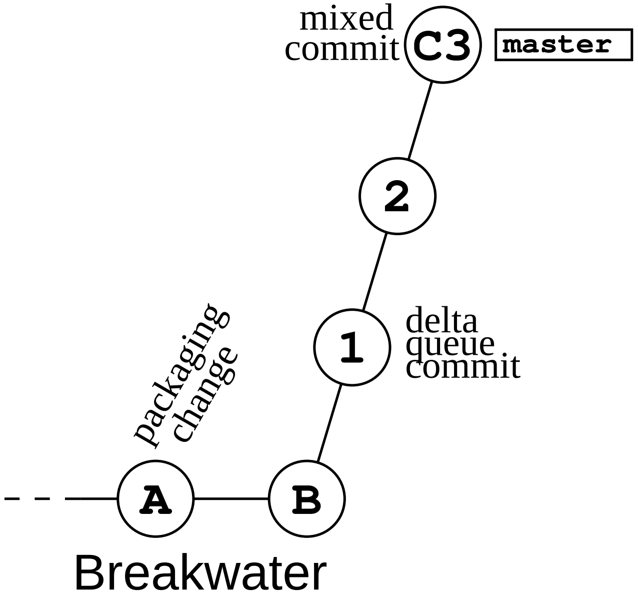
No need to ever use `quilt(1)`
Can mostly ignore `3.0 (quilt) dsc`
`3.0 (quilt) dsc` output is always perfect

Directly compatible with `dgit`
No command needed to upload other than `dgit push−source`

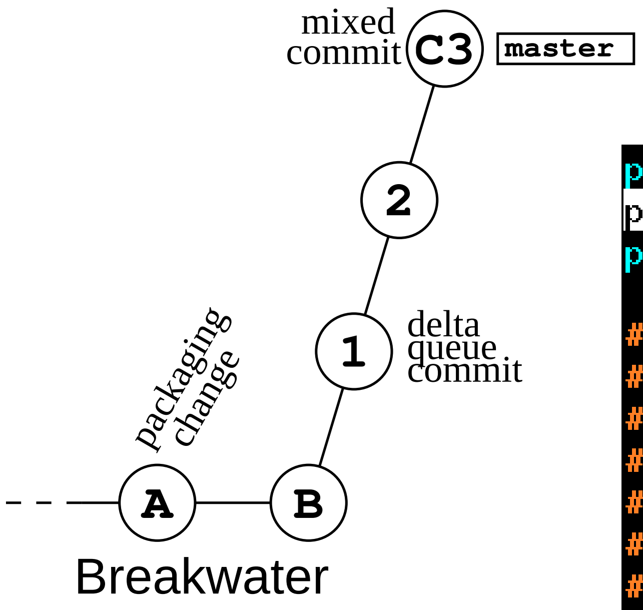
git−debrebase − commit history structure

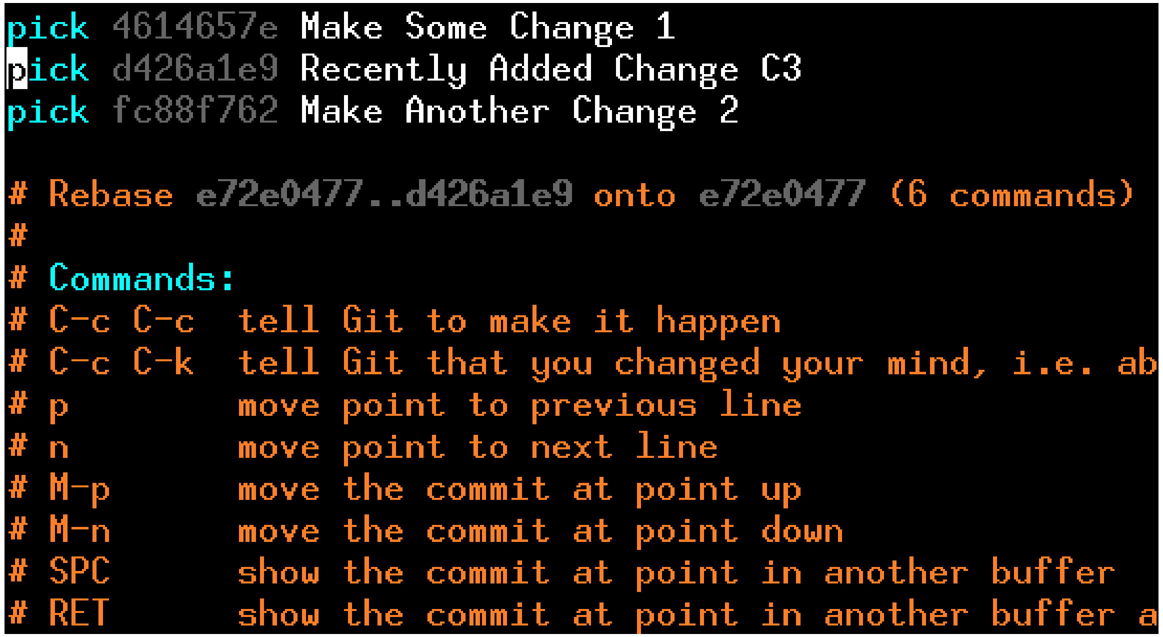


git−debrebase − commit history structure



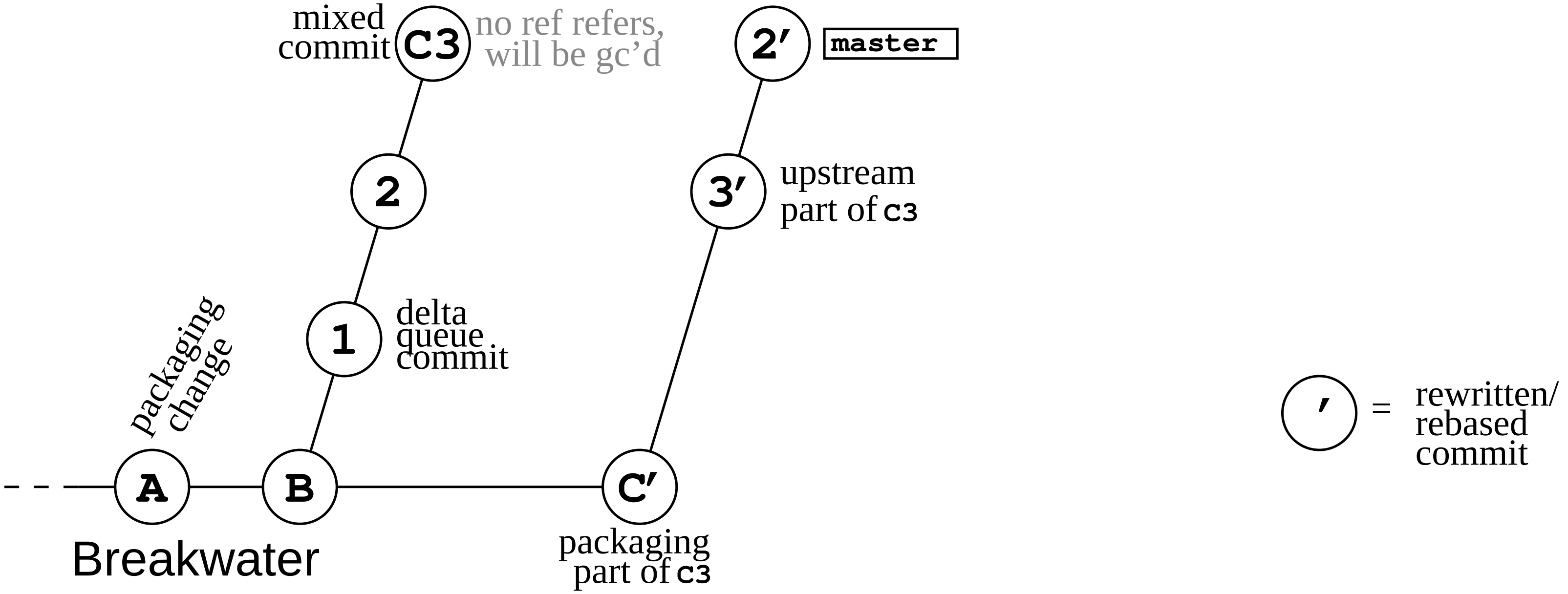
git−debrebase − commit history structure



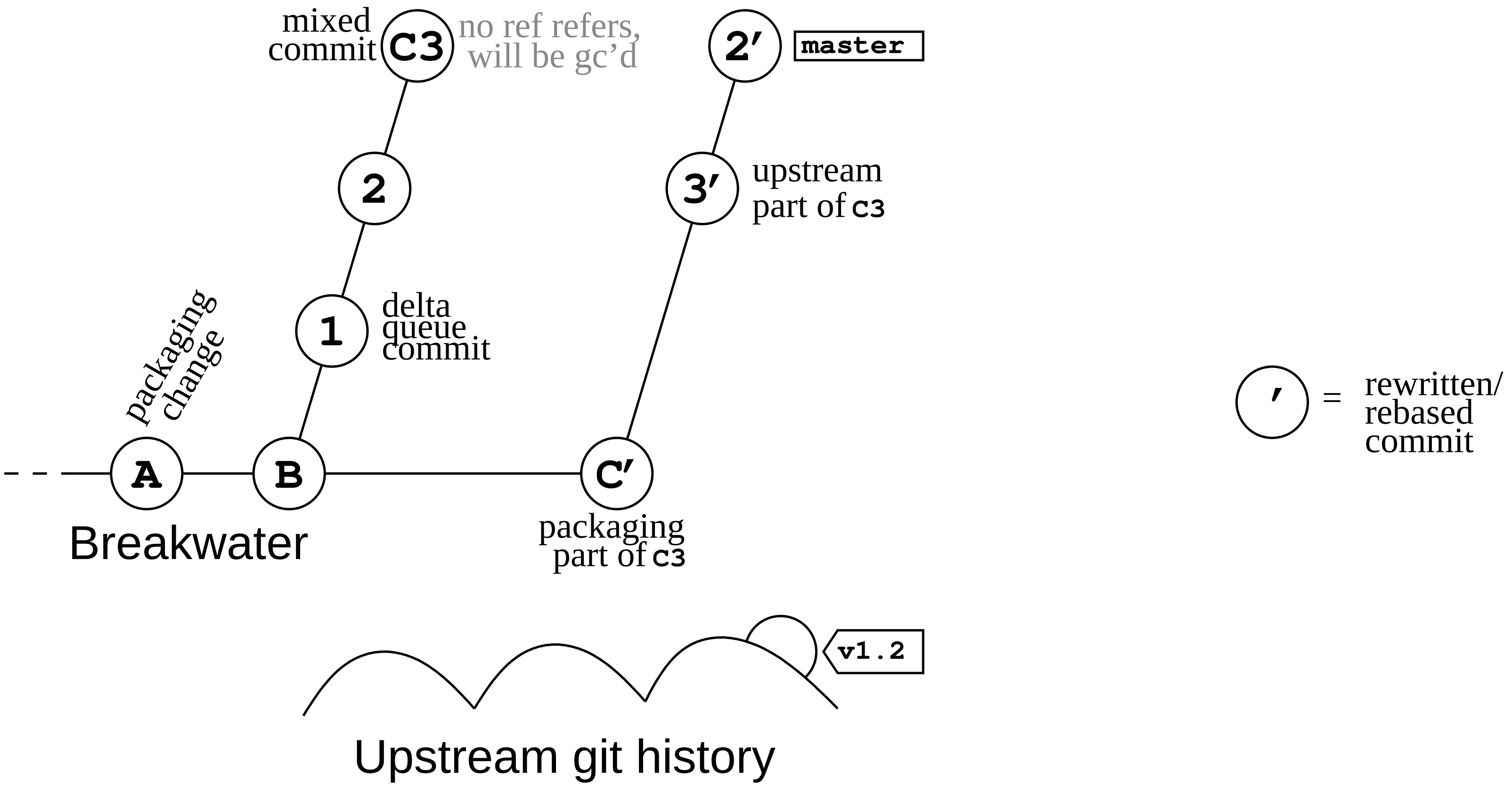


```
pick 4614657e Make Some Change 1
pick d426a1e9 Recently Added Change C3
pick fc88f762 Make Another Change 2

# Rebase e72e0477..d426a1e9 onto e72e0477 (6 commands)
#
# Commands:
# C-c C-c  tell Git to make it happen
# C-c C-k  tell Git that you changed your mind, i.e. ab
# p        move point to previous line
# n        move point to next line
# M-p      move the commit at point up
# M-n      move the commit at point down
# SPC      show the commit at point in another buffer
# RET      show the commit at point in another buffer a
```

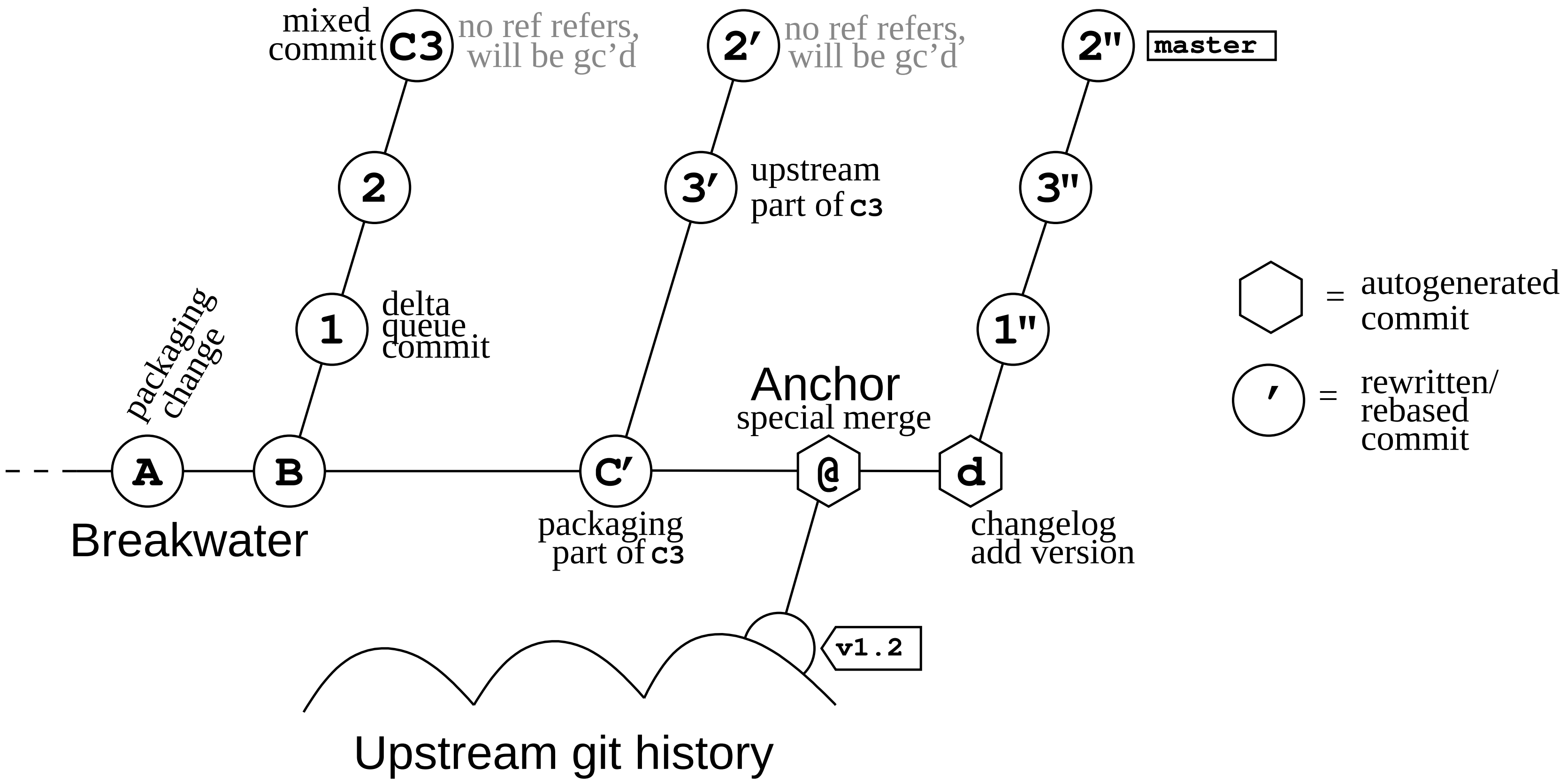
git−debrebase − commit history structure   after git−debrebase



git−debrebase − commit history structure



git−debrebase − commit history structure   after new−upstream



mixed commit

C3

no ref refers, will be gc'd

2

1

delta queue commit

packaging change

A

B

Breakwater

2′

no ref refers, will be gc'd

3′

upstream part of c3

C′

packaging part of c3

Anchor

special merge

@

d

changelog add version

1"

3"

2"

master

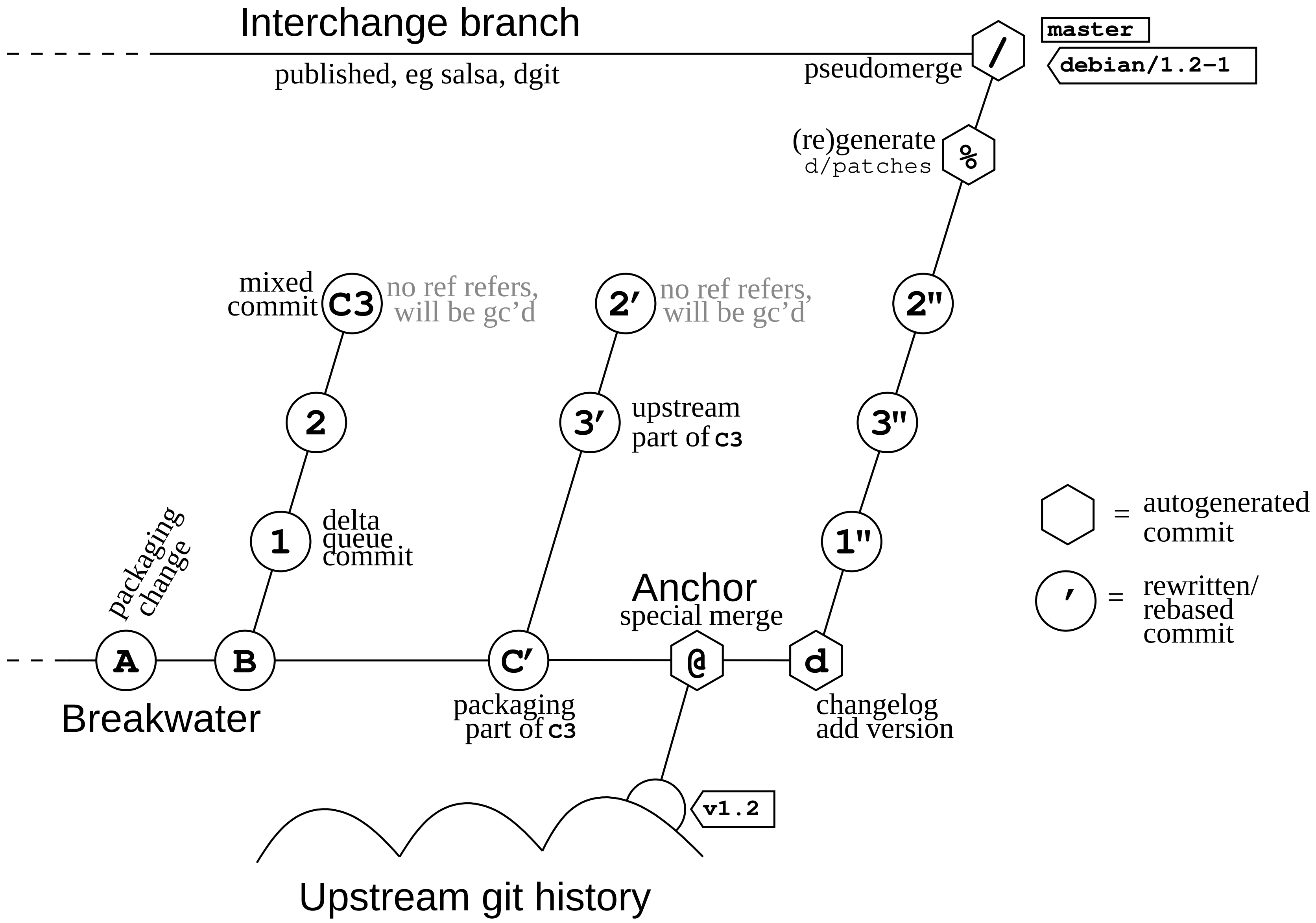
v1.2

Upstream git history

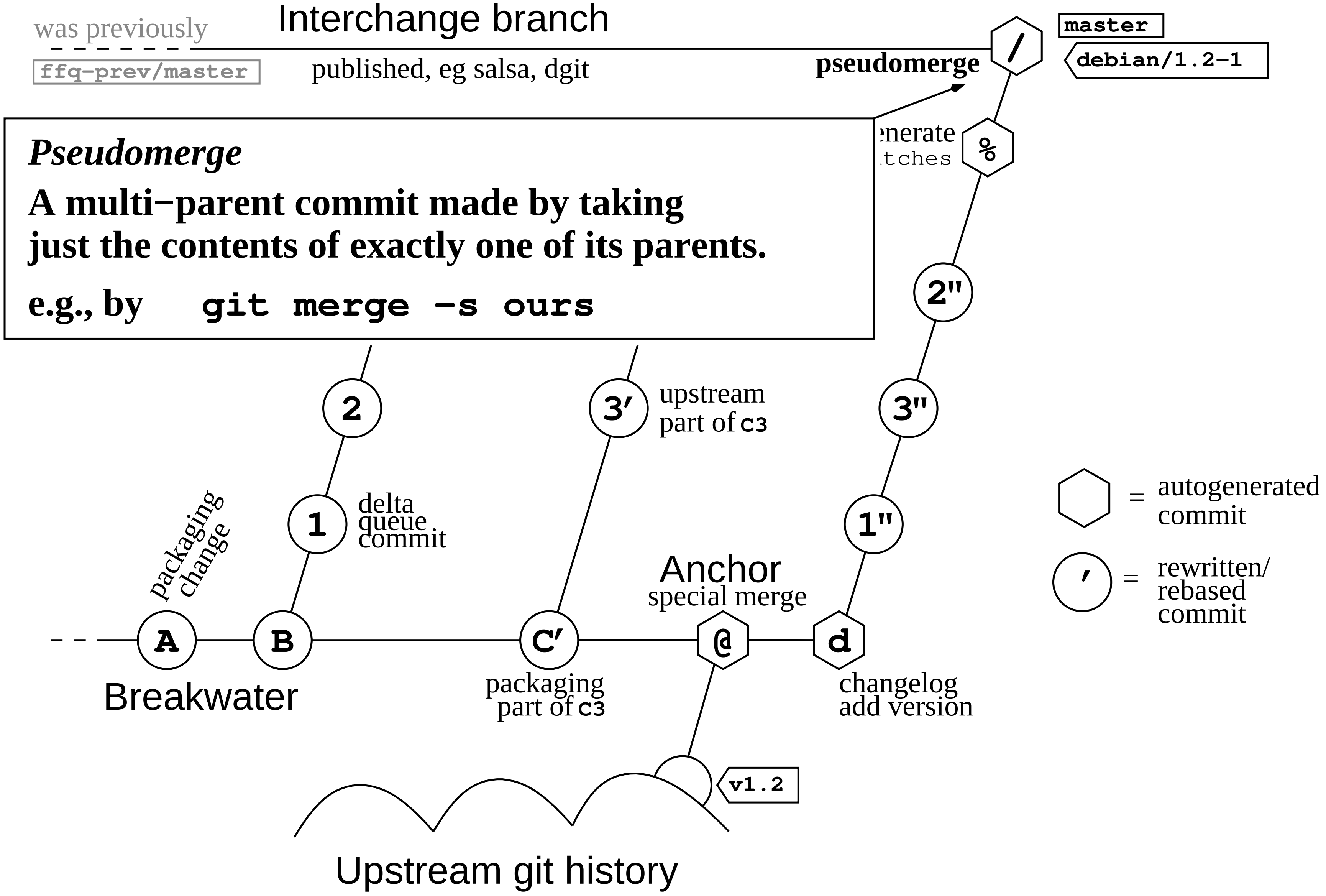
⬡ = autogenerated commit

′ = rewritten/ rebased commit

git−debrebase − commit history structure after upload



git−debrebase − commit history structure after upload



Interchange branch

was previously

ffq−prev/master

published, eg salsa, dgit

pseudomerge

master

debian/1.2−1

/

:nerate

.tches

%

2"

3"

1"

2

1

delta queue commit

3′

upstream part of c3

packaging change

A

B

C′

packaging part of c3

Anchor special merge

@

d

changelog add version

Breakwater

v1.2

Upstream git history

= autogenerated commit

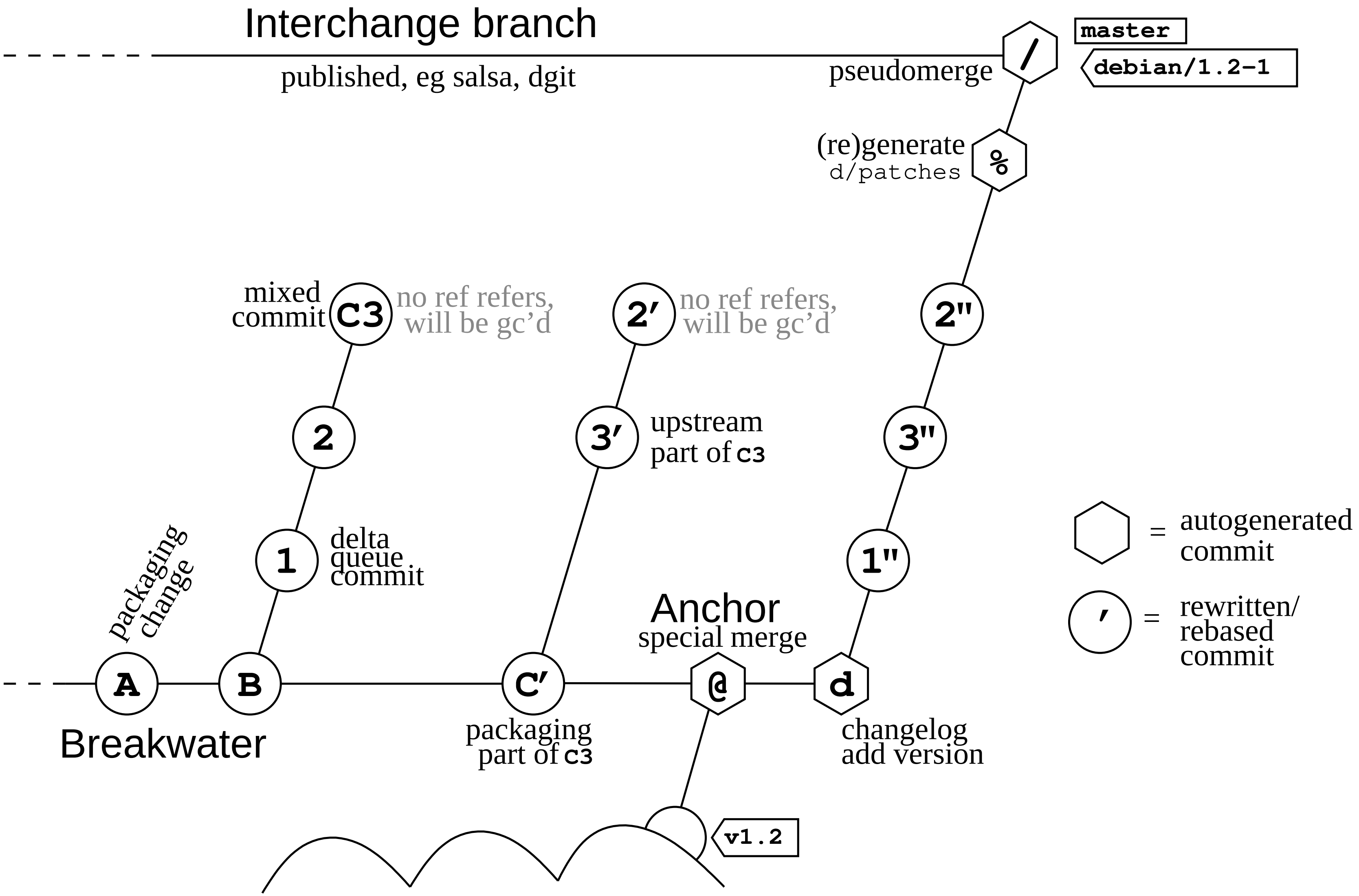
′ = rewritten/ rebased commit

Pseudomerge

A multi−parent commit made by taking just the contents of exactly one of its parents.

e.g., by `git merge -s ours`

git−debrebase − commit history structure after upload



Interchange branch

published, eg salsa, dgit

pseudomerge

master

debian/1.2−1

(re)generate
d/patches

mixed
commit

no ref refers,
will be gc'd

no ref refers,
will be gc'd

upstream
part of c3

delta
queue
commit

packaging
change

Anchor
special merge

packaging
part of c3

changelog
add version

Breakwater

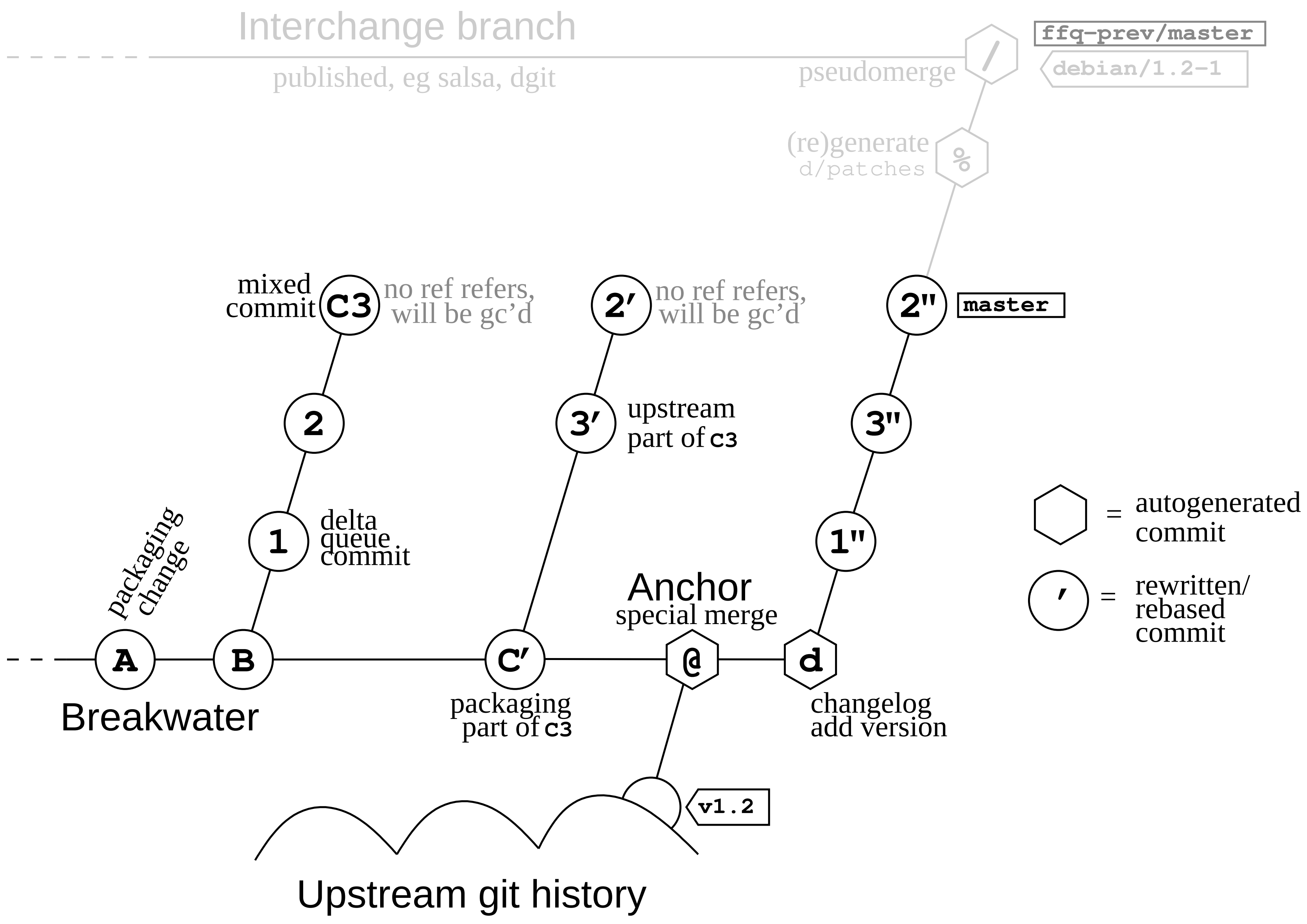
v1.2

= autogenerated
commit

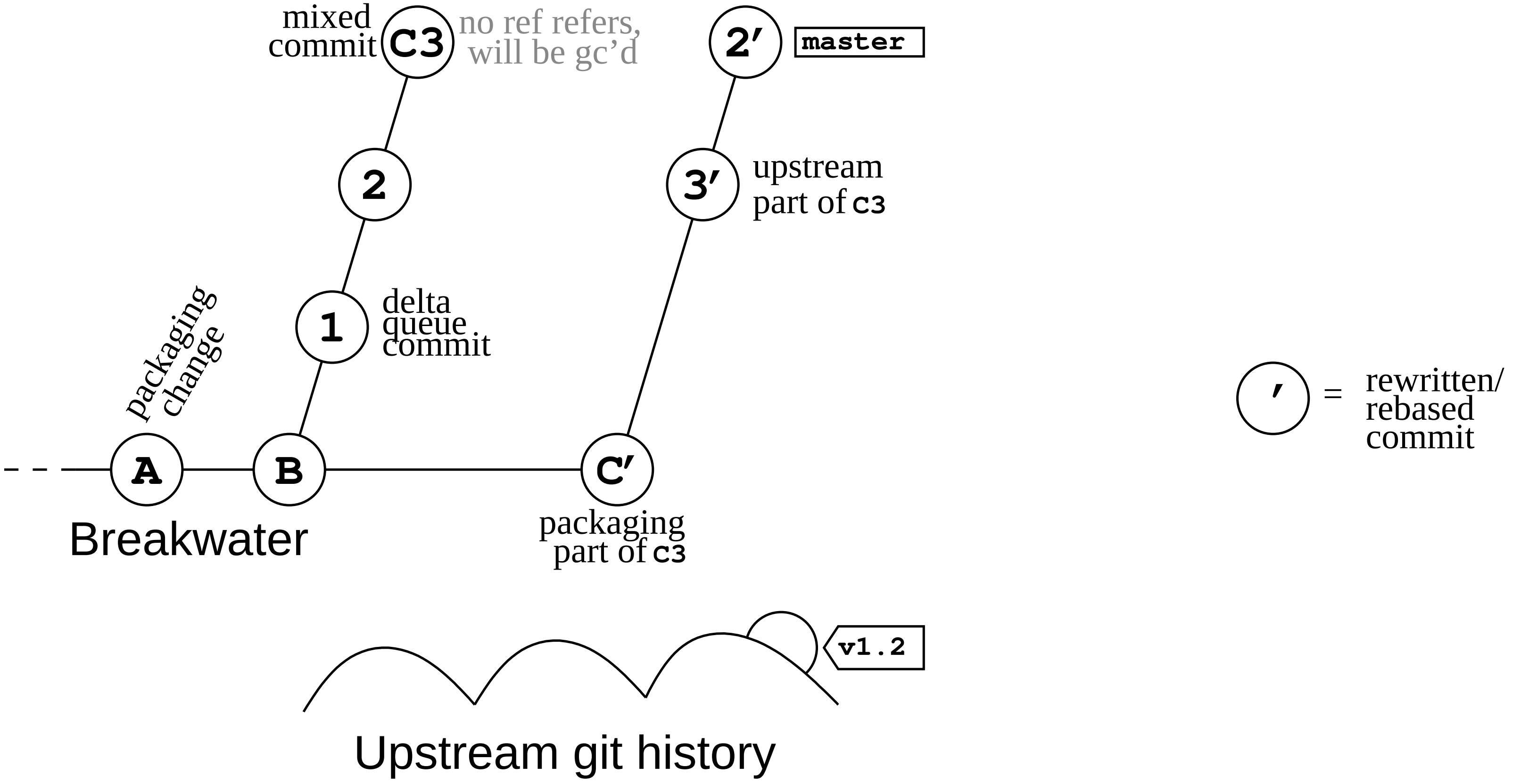
= rewritten/
rebased
commit

Upstream git history

git−debrebase − commit history structure rebasing again

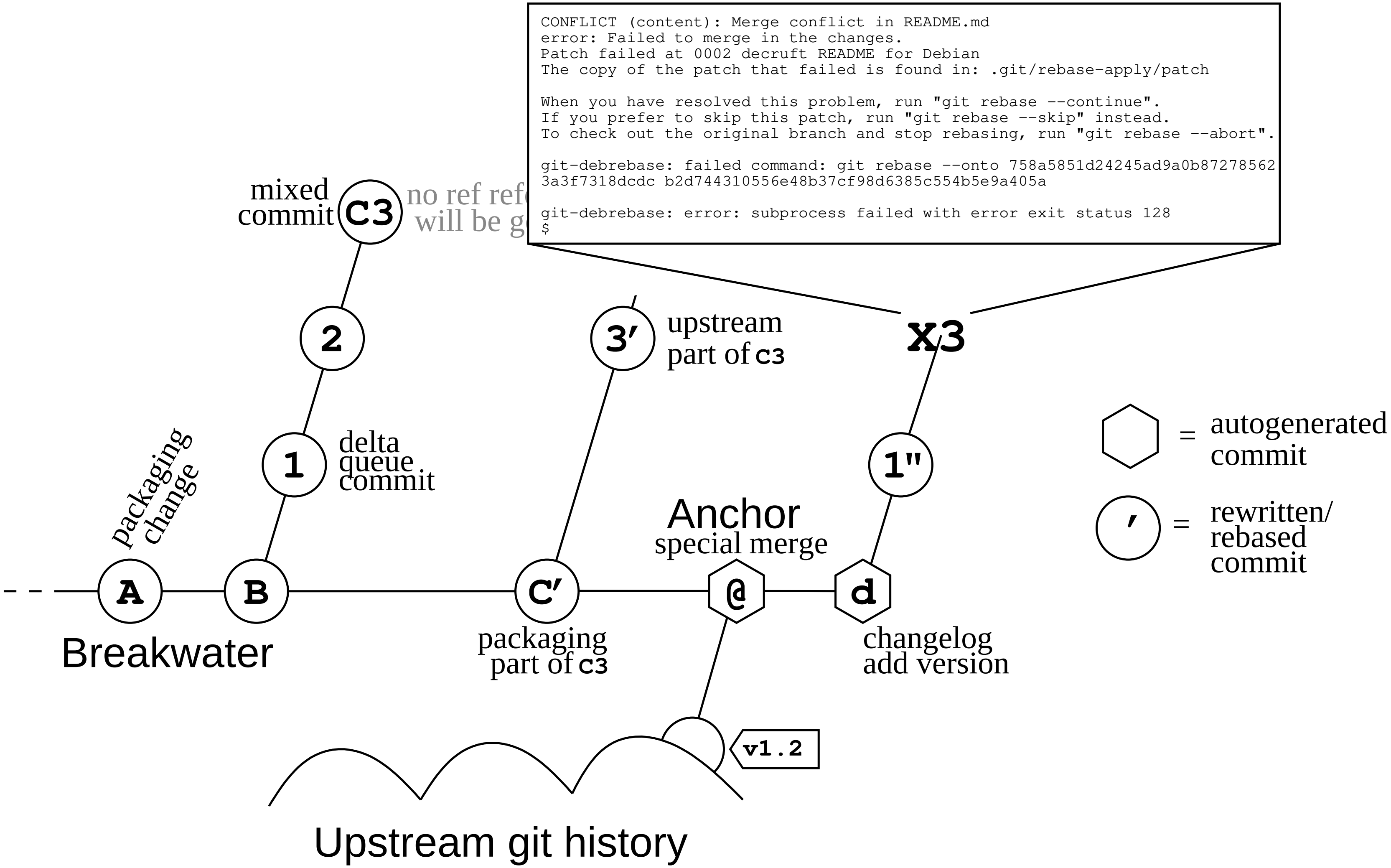


git−debrebase − commit history structure



git−debrebase − commit history structure

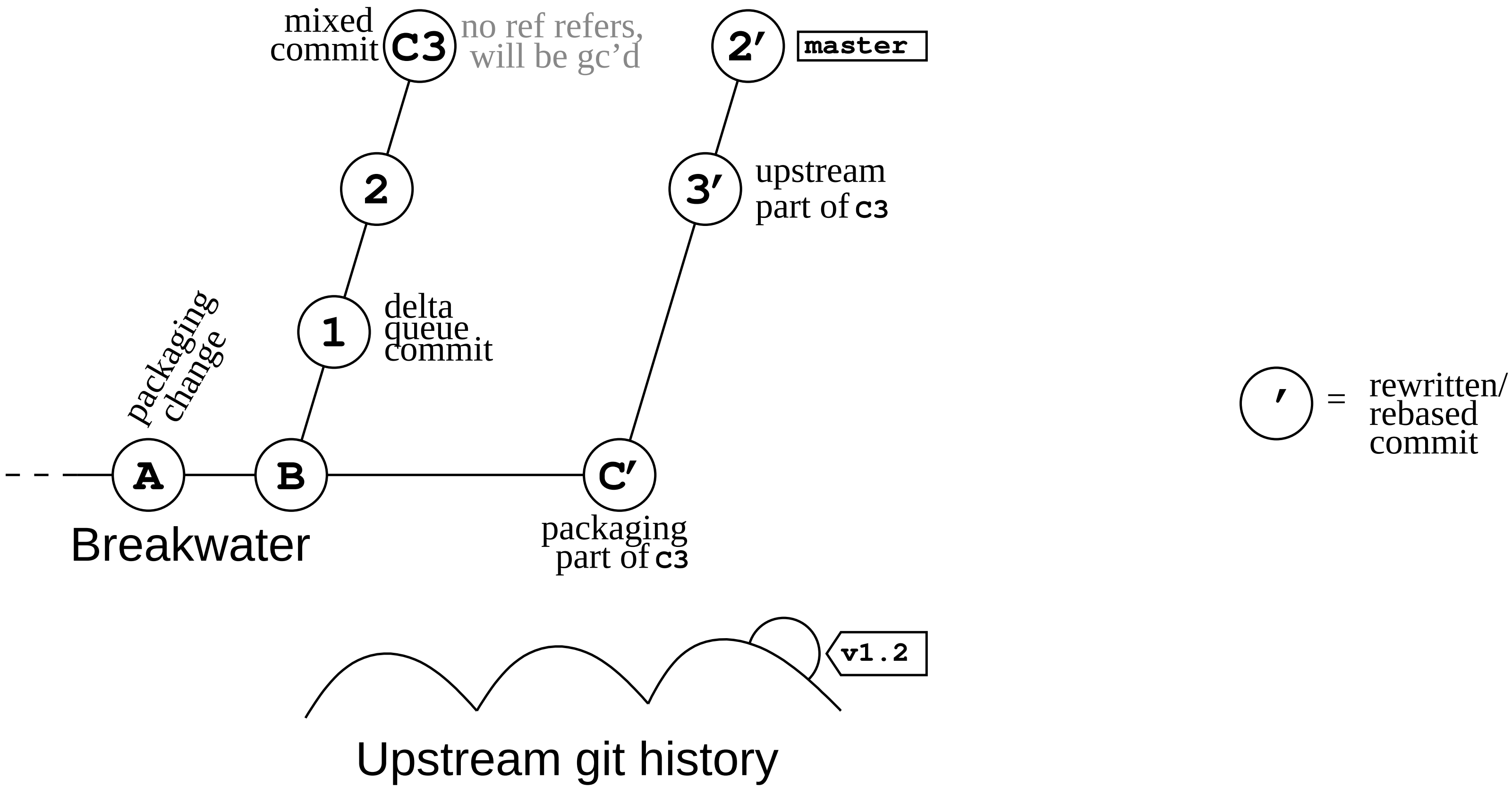
failed rebase for new−upstream



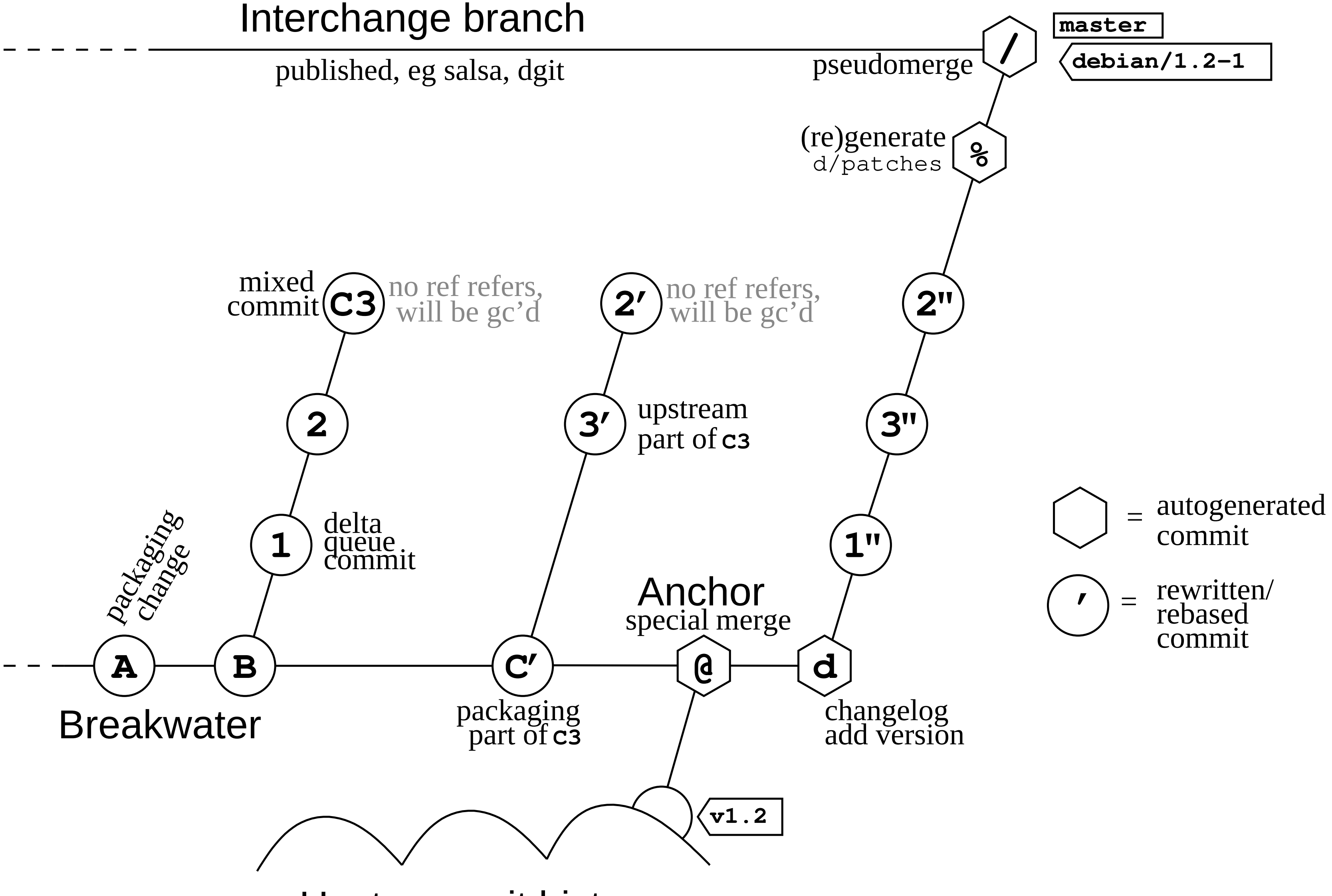
```
CONFLICT (content): Merge conflict in README.md
error: Failed to merge in the changes.
Patch failed at 0002 decruft README for Debian
The copy of the patch that failed is found in: .git/rebase-apply/patch

When you have resolved this problem, run "git rebase --continue".
If you prefer to skip this patch, run "git rebase --skip" instead.
To check out the original branch and stop rebasing, run "git rebase --abort".

git-debrebase: failed command: git rebase --onto 758a5851d24245ad9a0b87278562
3a3f7318dcdc b2d744310556e48b37cf98d6385c554b5e9a405a

git-debrebase: error: subprocess failed with error exit status 128
$
```

git−debrebase − commit history structure

aborted rebase for new−upstream



git−debrebase − commit history structure after upload



Interchange branch

published, eg salsa, dgit

pseudomerge

(re)generate
d/patches

mixed
commit

no ref refers,
will be gc'd

no ref refers,
will be gc'd

upstream
part of c3

delta
queue
commit

packaging
change

Anchor
special merge

Breakwater

packaging
part of c3

changelog
add version

= autogenerated
commit

= rewritten/
rebased
commit

Upstream git history

git−debrebase − references

`git−debrebase.deb` in testing and stretch−backports

`dgit−maint−debrebase(7)` workflow tutorial in `dgit.deb`

`https://manpages.debian.org/unstable/dgit−maint−debrebase`

Debconf 18 git−debrebase and dgit workshop
drop−in session (bring your laptop or just your questions)
Tuesday 10:00−11:45 room X

`git−debrebase(5)`
`git−debrebase(1)`

Formal reference manuals

```
git debrebase convert−from−gbp
```