Kontent.ai

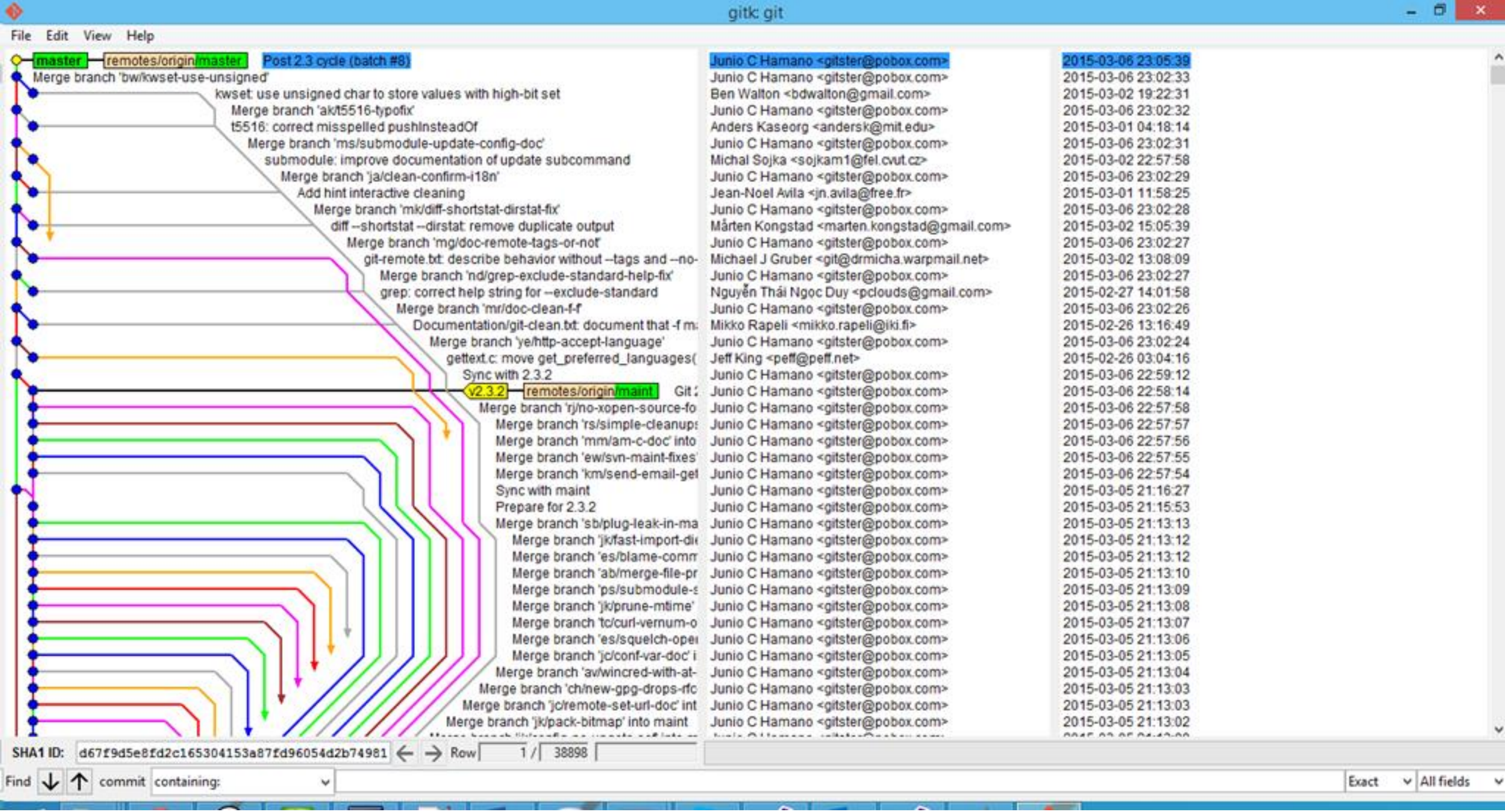
GIT

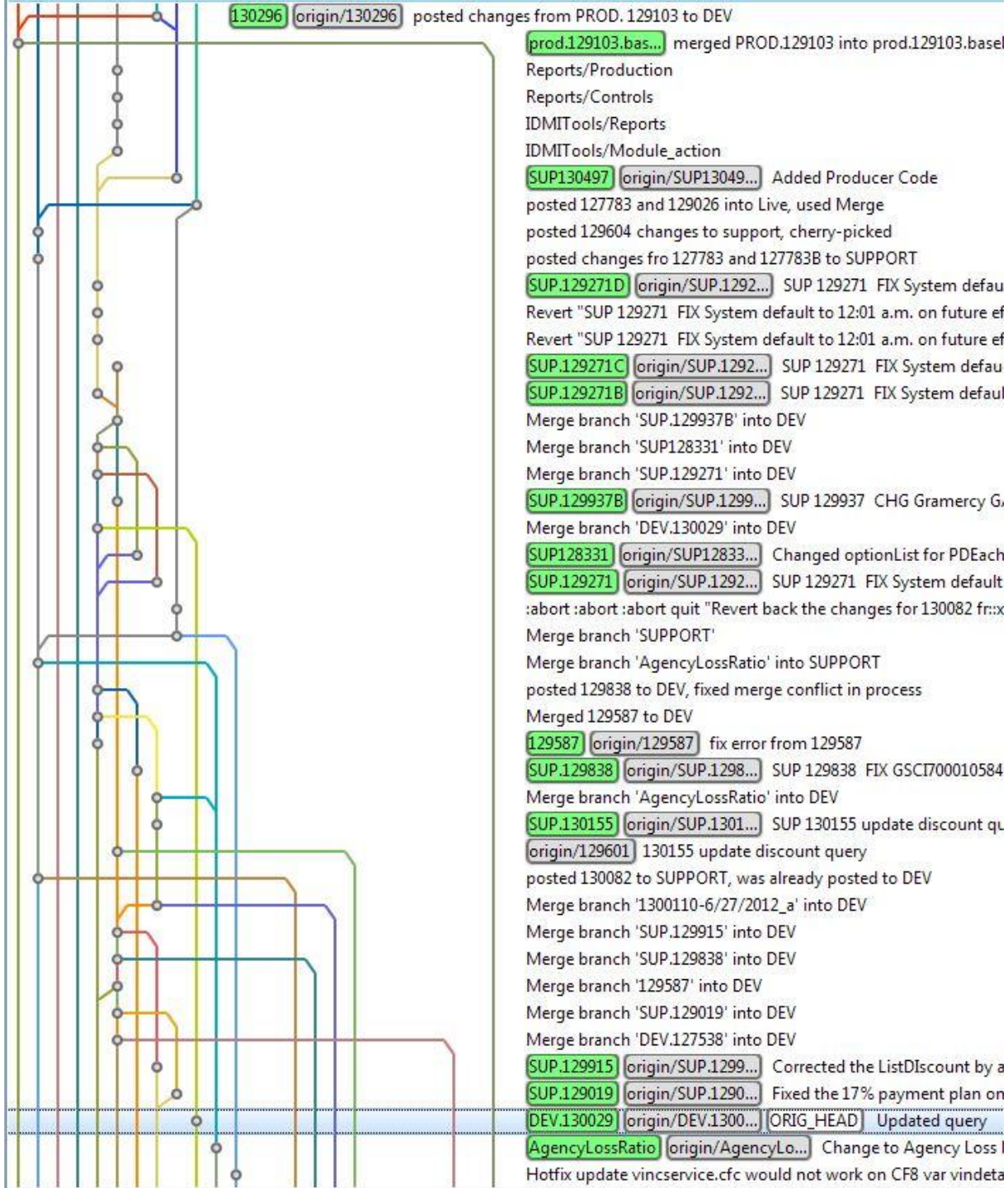
Patrik Medveď & Vilém Ťulák Jeniš

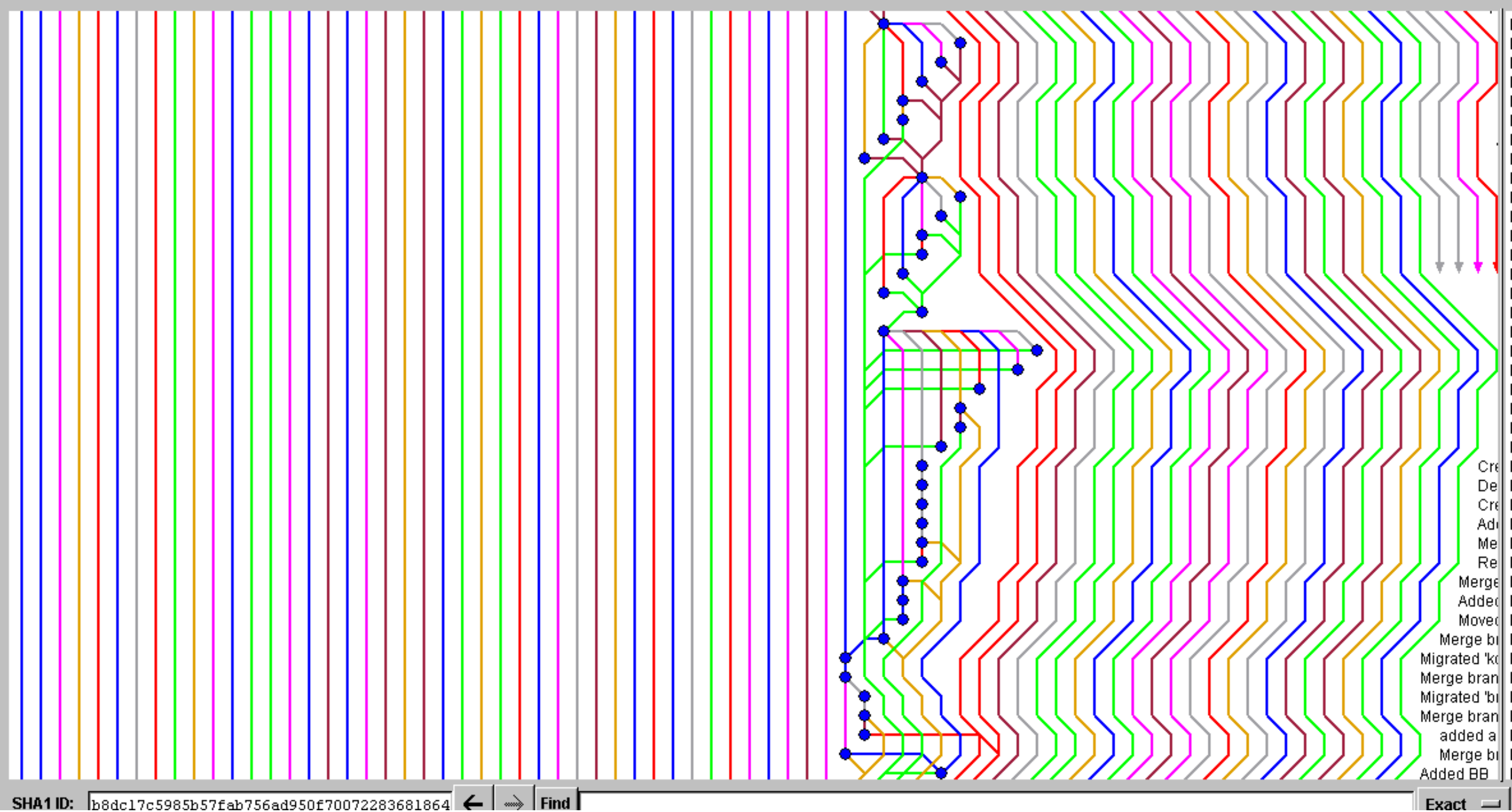
Kontent.ai

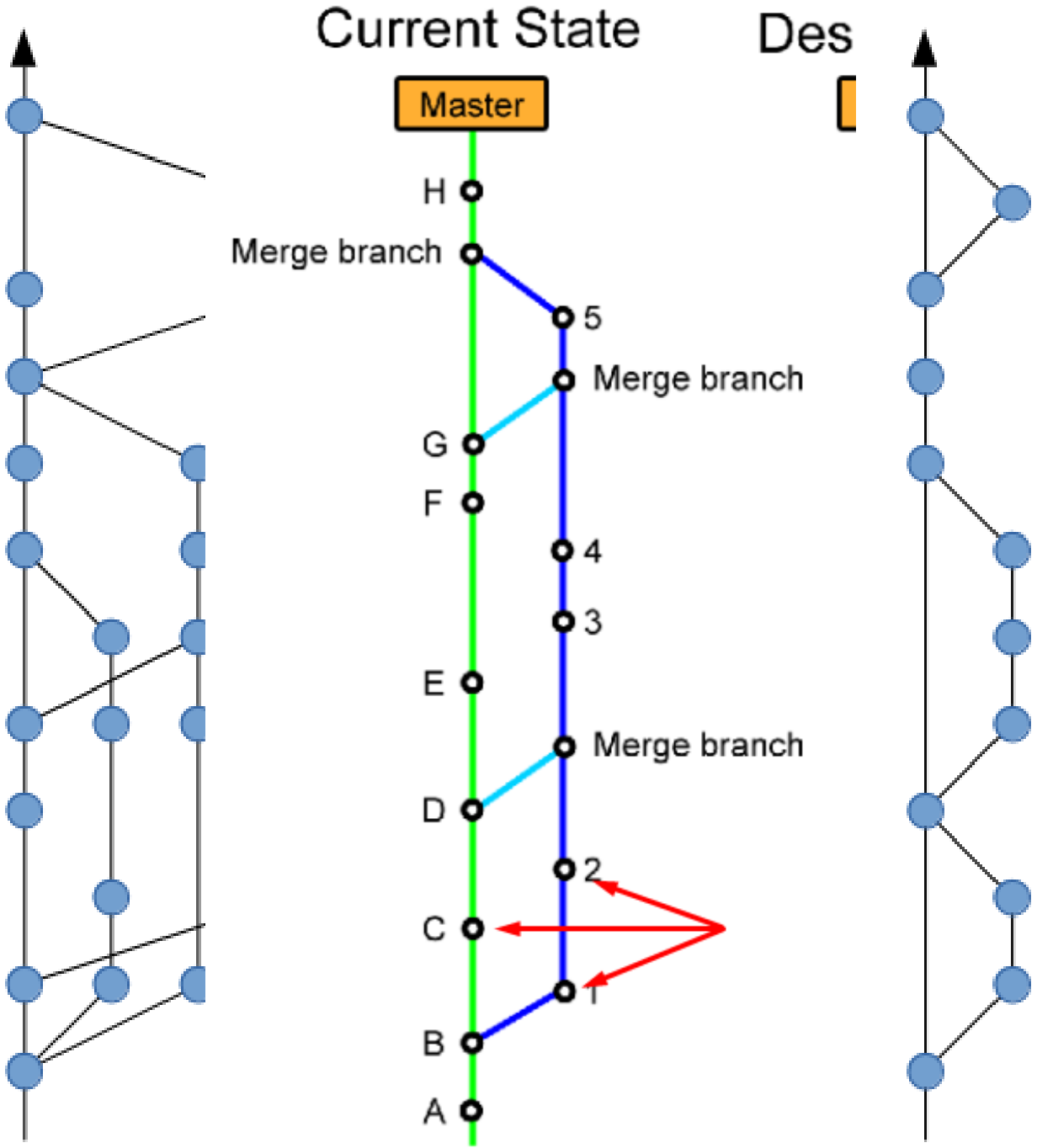
Motivation

Why even bother?


gitk: git
File Edit View Help
master remotes/origin/master Post 2.3 cycle (batch #8)
Merge branch 'bw/kwset-use-unsigned'
kwset: use unsigned char to store values with high-bit set
Merge branch 'ak/t5516-typofix'
t5516: correct misspelled pushInsteadOf
Merge branch 'ms/submodule-update-config-doc'
submodule: improve documentation of update subcommand
Merge branch 'ja/clean-confirm-i18n'
Add hint interactive cleaning
Merge branch 'mk/diff-shortstat-dirstat-fix'
diff --shortstat --dirstat: remove duplicate output
Merge branch 'mg/doc-remote-tags-or-not'
git-remote.txt: describe behavior without --tags and --no-
Merge branch 'nd/grep-exclude-standard-help-fix'
grep: correct help string for --exclude-standard
Merge branch 'mr/doc-clean-f-f'
Documentation/git-clean.txt: document that -f m
Merge branch 'ye/http-accept-language'
gettext.c: move get_preferred_languages(
Sync with 2.3.2
v2.3.2 remotes/origin/maint
Merge branch 'rj/no-xopen-source-fo
Merge branch 'rs/simple-cleanups
Merge branch 'mm/am-c-doc' into
Merge branch 'ew/svn-maint-fixes'
Merge branch 'km/send-email-get
Sync with maint
Prepare for 2.3.2
Merge branch 'sb/plug-leak-in-ma
Merge branch 'jk/fast-import-di
Merge branch 'es/blame-comm
Merge branch 'ab/merge-file-pr
Merge branch 'ps/submodule-s
Merge branch 'jk/prune-mtime'
Merge branch 'tc/curl-vernum-o
Merge branch 'es/squelch-ope
Merge branch 'jc/conf-var-doc' i
Merge branch 'av/wincred-with-at-
Merge branch 'ch/new-gpg-drops-rfc
Merge branch 'jc/remote-set-url-doc' int
Merge branch 'jk/pack-bitmap' into maint
Junio C Hamano <gitster@pobox.com> 2015-03-06 23:05:39
Junio C Hamano <gitster@pobox.com> 2015-03-06 23:02:33
Ben Walton <bdwalton@gmail.com> 2015-03-02 19:22:31
Junio C Hamano <gitster@pobox.com> 2015-03-06 23:02:32
Anders Kaseorg <andersk@mit.edu> 2015-03-01 04:18:14
Junio C Hamano <gitster@pobox.com> 2015-03-06 23:02:31
Michal Sojka <sojkam1@fel.cvut.cz> 2015-03-02 22:57:58
Junio C Hamano <gitster@pobox.com> 2015-03-06 23:02:29
Jean-Noel Avila <jn.avila@free.fr> 2015-03-01 11:58:25
Junio C Hamano <gitster@pobox.com> 2015-03-06 23:02:28
Mårten Kongstad <marten.kongstad@gmail.com> 2015-03-02 15:05:39
Junio C Hamano <gitster@pobox.com> 2015-03-06 23:02:27
Michael J Gruber <git@drmicha.warpmail.net> 2015-03-02 13:08:09
Junio C Hamano <gitster@pobox.com> 2015-03-06 23:02:27
Nguyễn Thái Ngọc Duy <pclouds@gmail.com> 2015-02-27 14:01:58
Junio C Hamano <gitster@pobox.com> 2015-03-06 23:02:26
Mikko Rapeli <mikko.rapeli@iki.fi> 2015-02-26 13:16:49
Junio C Hamano <gitster@pobox.com> 2015-03-06 23:02:24
Jeff King <peff@peff.net> 2015-02-26 03:04:16
Junio C Hamano <gitster@pobox.com> 2015-03-06 22:59:12
Junio C Hamano <gitster@pobox.com> 2015-03-06 22:58:14
Junio C Hamano <gitster@pobox.com> 2015-03-06 22:57:58
Junio C Hamano <gitster@pobox.com> 2015-03-06 22:57:57
Junio C Hamano <gitster@pobox.com> 2015-03-06 22:57:56
Junio C Hamano <gitster@pobox.com> 2015-03-06 22:57:55
Junio C Hamano <gitster@pobox.com> 2015-03-06 22:57:54
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:16:27
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:15:53
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:13:13
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:13:12
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:13:12
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:13:10
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:13:09
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:13:08
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:13:07
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:13:06
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:13:05
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:13:04
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:13:03
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:13:03
Junio C Hamano <gitster@pobox.com> 2015-03-05 21:13:02
SHA1 ID: d67f9d5e8fd2c165304153a87fd96054d2b74981
Row 1 / 38898
Find commit containing:
Exact
All fields


130296 origin/130296 posted changes from PROD. 129103 to DEV
prod.129103.bas... merged PROD.129103 into prod.129103.basel
Reports/Production
Reports/Controls
IDMITools/Reports
IDMITools/Module_action
SUP130497 origin/SUP13049... Added Producer Code
posted 127783 and 129026 into Live, used Merge
posted 129604 changes to support, cherry-picked
posted changes fro 127783 and 127783B to SUPPORT
SUP.129271D origin/SUP.1292... SUP 129271 FIX System defau
Revert "SUP 129271 FIX System default to 12:01 a.m. on future ef
Revert "SUP 129271 FIX System default to 12:01 a.m. on future ef
SUP.129271C origin/SUP.1292... SUP 129271 FIX System defau
SUP.129271B origin/SUP.1292... SUP 129271 FIX System defaul
Merge branch 'SUP.129937B' into DEV
Merge branch 'SUP128331' into DEV
Merge branch 'SUP.129271' into DEV
SUP.129937B origin/SUP.1299... SUP 129937 CHG Gramercy G
Merge branch 'DEV.130029' into DEV
SUP128331 origin/SUP12833... Changed optionList for PDEach
SUP.129271 origin/SUP.1292... SUP 129271 FIX System default
:abort :abort :abort quit "Revert back the changes for 130082 fr::x
Merge branch 'SUPPORT'
Merge branch 'AgencyLossRatio' into SUPPORT
posted 129838 to DEV, fixed merge conflict in process
Merged 129587 to DEV
129587 origin/129587 fix error from 129587
SUP.129838 origin/SUP.1298... SUP 129838 FIX GSCI700010584
Merge branch 'AgencyLossRatio' into DEV
SUP.130155 origin/SUP.1301... SUP 130155 update discount qu
origin/129601 130155 update discount query
posted 130082 to SUPPORT, was already posted to DEV
Merge branch '1300110-6/27/2012_a' into DEV
Merge branch 'SUP.129915' into DEV
Merge branch 'SUP.129838' into DEV
Merge branch '129587' into DEV
Merge branch 'SUP.129019' into DEV
Merge branch 'DEV.127538' into DEV
SUP.129915 origin/SUP.1299... Corrected the ListDIscount by a
SUP.129019 origin/SUP.1290... Fixed the 17% payment plan on
DEV.130029 origin/DEV.1300... ORIG_HEAD Updated query
AgencyLossRatio origin/AgencyLo... Change to Agency Loss
Hotfix update vincservice.cfc would not work on CF8 var vindeta


Cre
De
Cre
Ad
Me
Re
Merge
Added
Moved
Merge bu
Migrated 'ko
Merge bran
Migrated 'bu
Merge bran
added a
Merge bu
Added BB
SHA1 ID:
b8dc17c5985b57fab756ad950f70072283681864
Find
Exact


Current State
Des
Master
H
Merge branch
5
Merge branch
G
F
4
3
E
Merge branch
D
2
C
1
B
A

Agenda

- Terminology
- Commit culture
- Immutable nature of GIT
- Rebase and merge
- Savepoints
- Interactive rebase

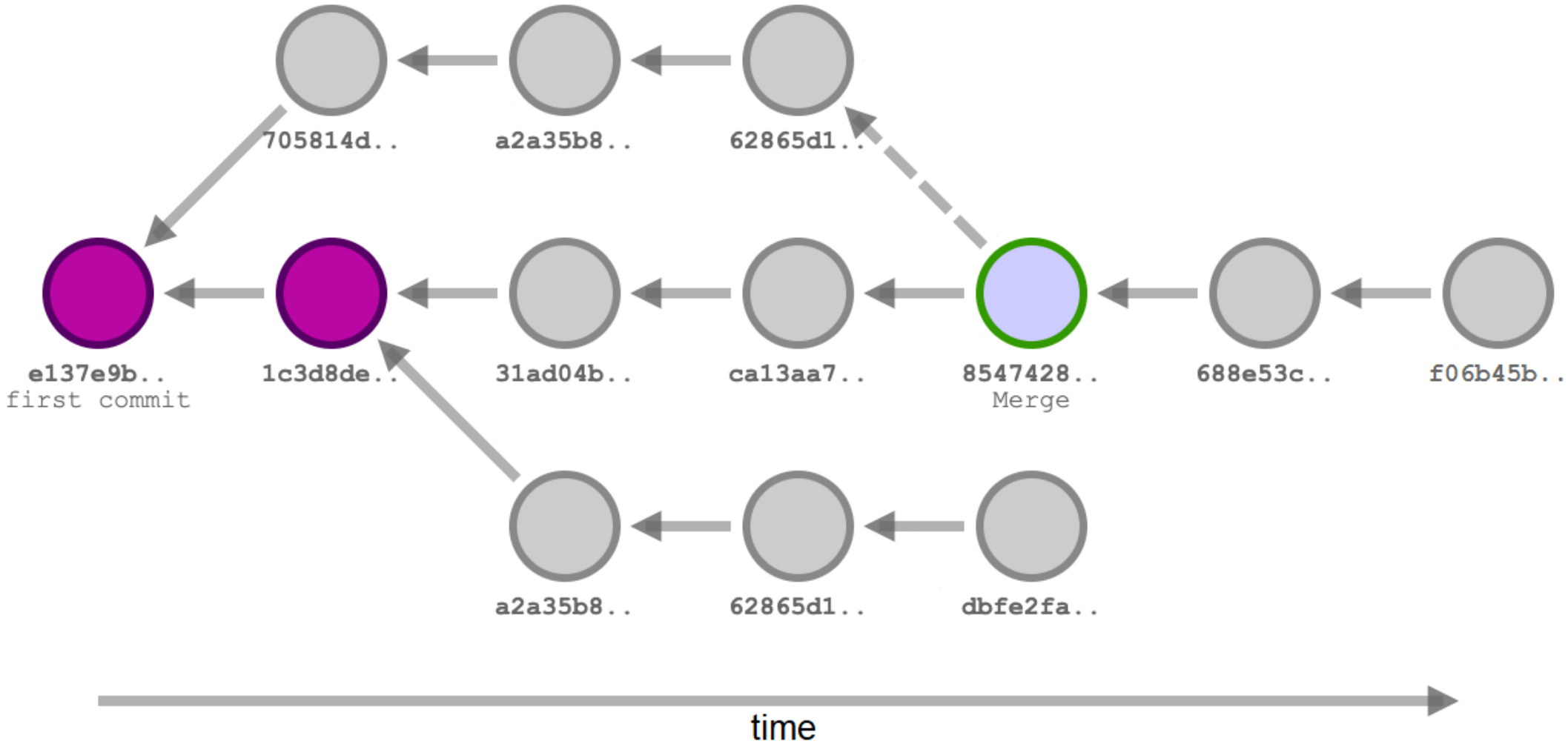
Kontent.ai

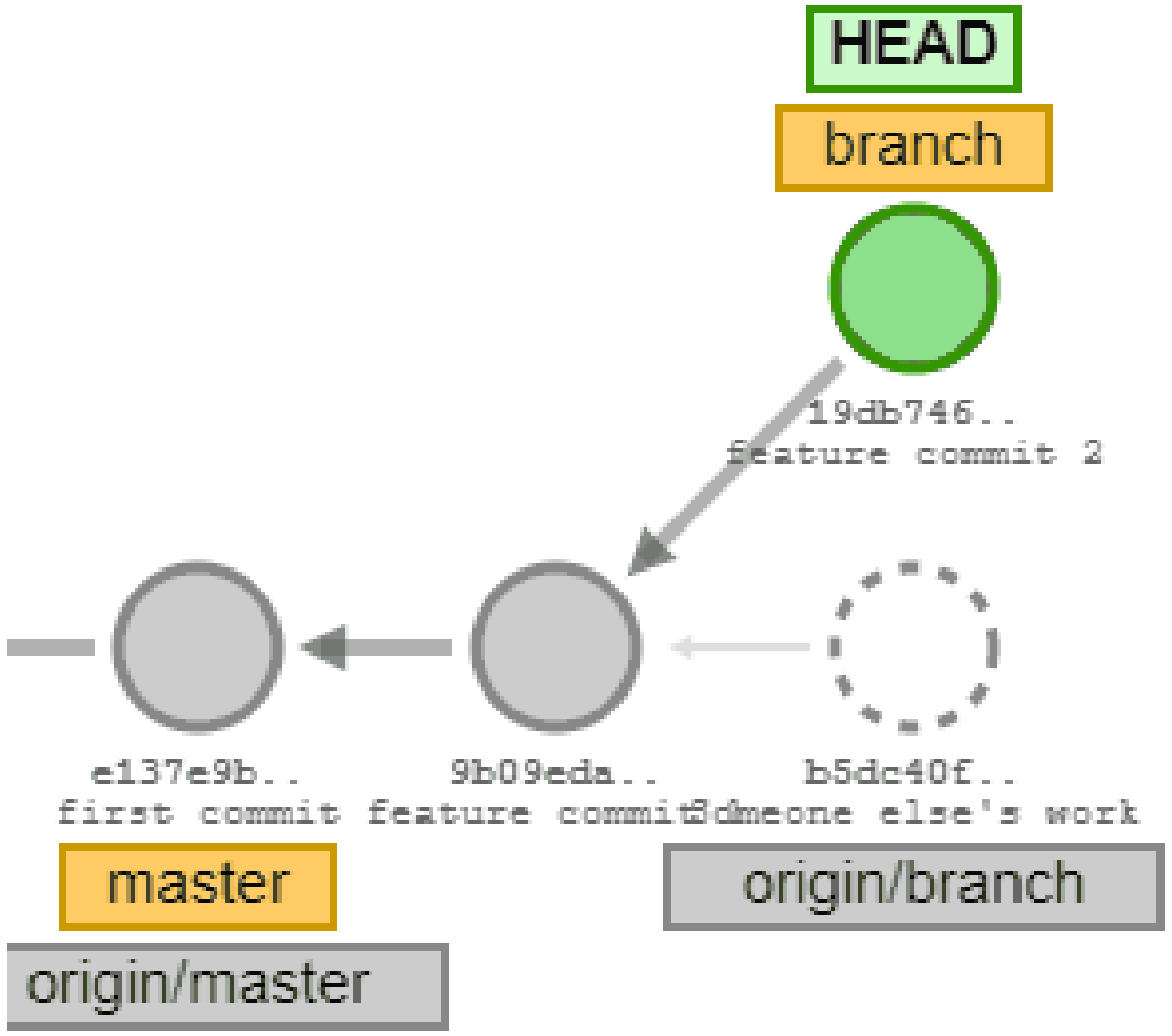
Terminology

First things first

Terminology

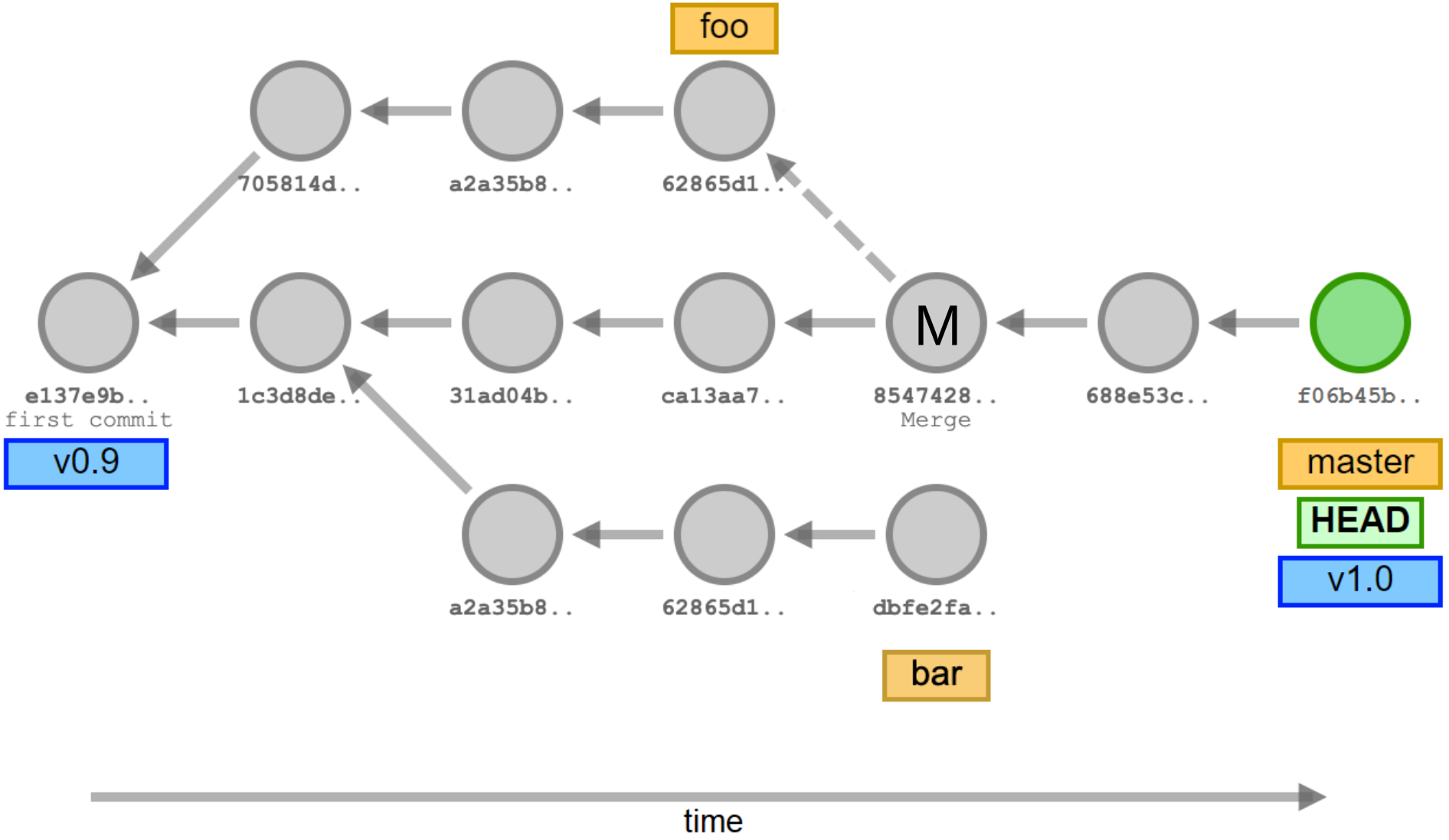
- Commit
  - State of files in time
  - Also, a set of changes
  - Node in the history graph
  - Commits are immutable


705814d..
a2a35b8..
62865d1..
e137e9b..
first commit
1c3d8de..
31ad04b..
ca13aa7..
8547428..
Merge
688e53c..
f06b45b..
a2a35b8..
62865d1..
dbfe2fa..
time


HEAD
branch
19db746..
feature commit 2
e137e9b..
first commit
9b09eda..
feature commit
b5dc40f..
someone else's work
master
origin/branch
origin/master

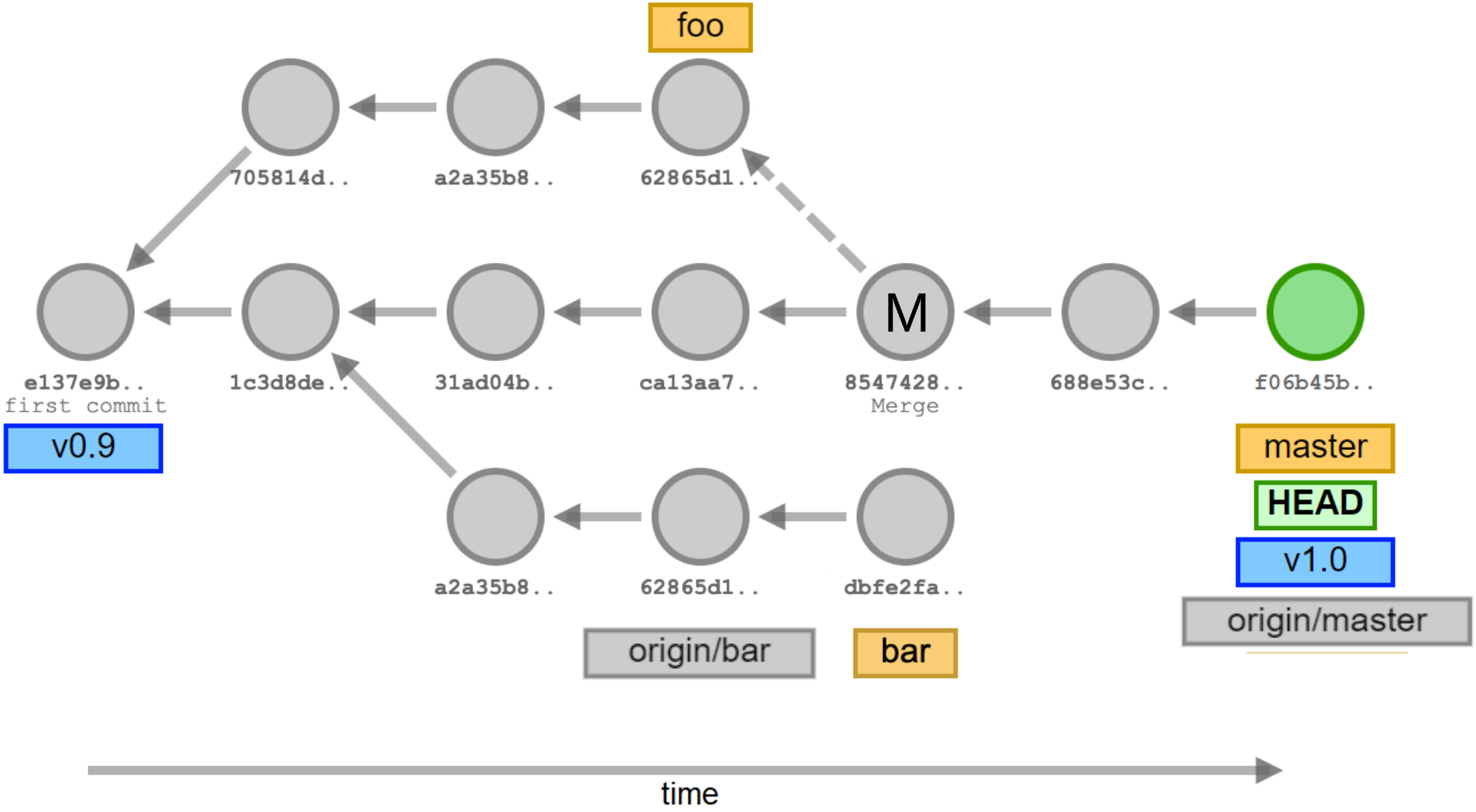
Terminology

- References
  - Named pointers to commits
  - **Branch** – named pointer
  - **Tag** – immutable named pointer
  - **HEAD** – current commit


foo
705814d..
a2a35b8..
62865d1..
e137e9b..
first commit
v0.9
1c3d8de..
31ad04b..
ca13aa7..
M
8547428..
Merge
688e53c..
f06b45b..
master
HEAD
v1.0
a2a35b8..
62865d1..
dbfe2fa..
bar
time

Terminology

- Origin & Upstream
  - Origin
  - Upstream
  - Remote
- Local & remote branches
  - Local branch
  - Remote branch
  - Remote-tracking branch


foo
705814d..
a2a35b8..
62865d1..
e137e9b..
first commit
v0.9
1c3d8de..
31ad04b..
ca13aa7..
M
8547428..
Merge
688e53c..
f06b45b..
master
HEAD
v1.0
origin/master
a2a35b8..
62865d1..
dbfe2fa..
origin/bar
bar
time

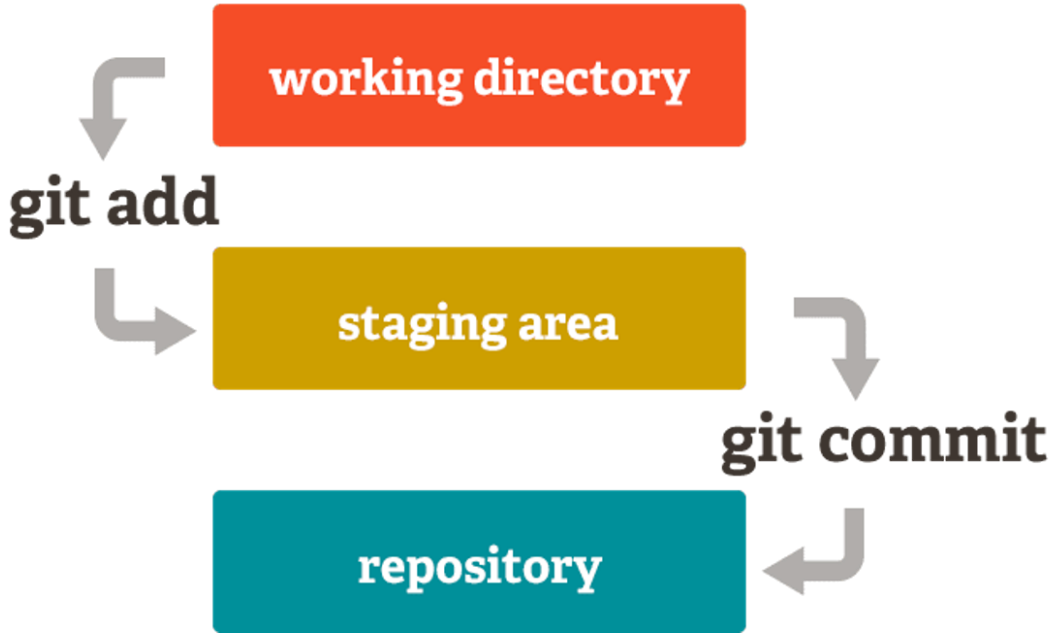
Terminology

- Working directory
  - Directory/file state
  - In sync with file system

Terminology

- Index/staging area
  - (Uncommitted) changes known to git
  - Staging area between the file system and the commit history
  - Snapshot of a future commit

Terminology



Kontent.ai

Commit culture

XXXXXXXXXX?????XXXXXXXXXXXXXX???XX

Make commits **atomic**!

- Contains one change (and its implications)
- Can be compiled and works after changes

- Easier reviews
- Easier rollback
- Easier bug identification

Commit message

Source: XKCD | xkcd.com

Kontent.ai

Commit message

- 2 parts
  - Title (~50 chars)
  - Description

```
git commit -m <title> -m <description>
```

(Or use the editor...)

Commit message

- 2 parts
  - Title (~50 chars)
  - Description

```
git commit -m <title> -m <description>
```

(Or use the editor...)

- Bad:
  - Fix review, Minor changes, change const X to 5, wtf, add tests, ...
- Good:
  - “write commits as if you were giving the computer an instruction on what changes to make”
  - Fix header to be consistent across all screens

Kontent.ai

Conventional Commits 1.0.0

Summary

The Conventional Commits specification is a lightweight convention on top of commit messages. It provides an easy set of rules for creating an explicit commit history; which makes it easier to write automated tools on top of. This convention dovetails with SemVer, by describing the features, fixes, and breaking changes made in commit messages.

The commit message should be structured as follows:

```
<type>[optional scope]: <description>

[optional body]

[optional footer(s)]
```

git add --patch

```
>git add --patch
diff --git a/.svgrrc.js b/.svgrrc.js
index 9b82a86334..f6947df385 100644
--- a/.svgrrc.js
+++ b/.svgrrc.js
@@ -13,9 +13,17 @@ module.exports = {
   ref: true,
   replaceAttrValues: { '#000': 'currentColor' },
   svgoConfig: {
-    plugins: {
-      convertPathData: false, // messes up some of the icons
-    }
+    plugins: [
+    {
+      name: 'preset-default',
+      params: {
+        overrides: {
+          convertPathData: false, // messes up some of the icons,
+          removeViewBox: false,
+        },
+      },
+    },
+  ],
   },
   template: iconTemplate,
   titleProp: false,
(1/1) Stage this hunk [y,n,q,a,d,e,?]? ?
```

Kontent.ai

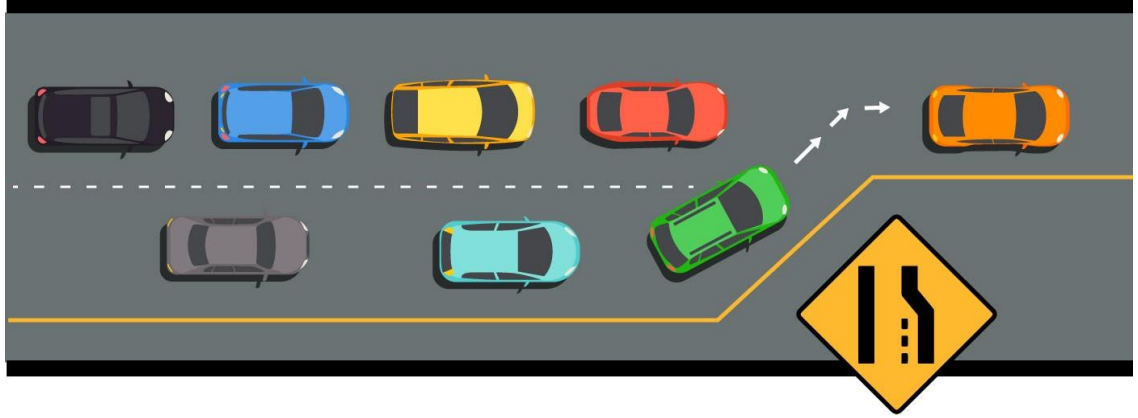
```
y - stage this hunk
n - do not stage this hunk
q - quit; do not stage this hunk or any of the remaining ones
a - stage this hunk and all later hunks in the file
d - do not stage this hunk or any of the later hunks in the file
e - manually edit the current hunk
? - print help
@@ -13,9 +13,17 @@ module.exports = {
   ref: true,
   replaceAttrValues: { '#000': 'currentColor' },
   svgoConfig: {
-    plugins: {
-      convertPathData: false, // messes up some of the icons
-    }
+    plugins: [
+    {
+      name: 'preset-default',
+      params: {
+        overrides: {
+          convertPathData: false, // messes up some of the icons,
+          removeViewBox: false,
+        },
+      },
+    },
+  ],
   },
   template: iconTemplate,
   titleProp: false,
(1/1) Stage this hunk [y,n,q,a,d,e,?]?
```

Kontent.ai

Merge and rebase

Not versus...

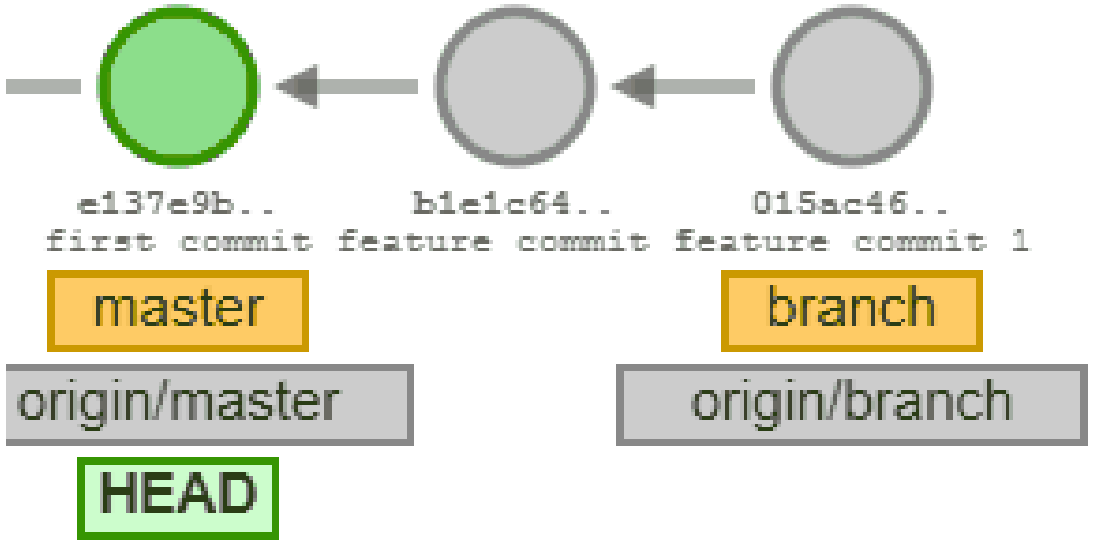
merge



- Reconciles two branches
  - Doesn’t have to be branches, but let’s stick to what we mostly do.
  - Has several ~~strategies~~ approaches:
    - --ff-only
    - --no-ff
    - --ff
  - Strategies are these: https://git-scm.com/docs/merge-strategies

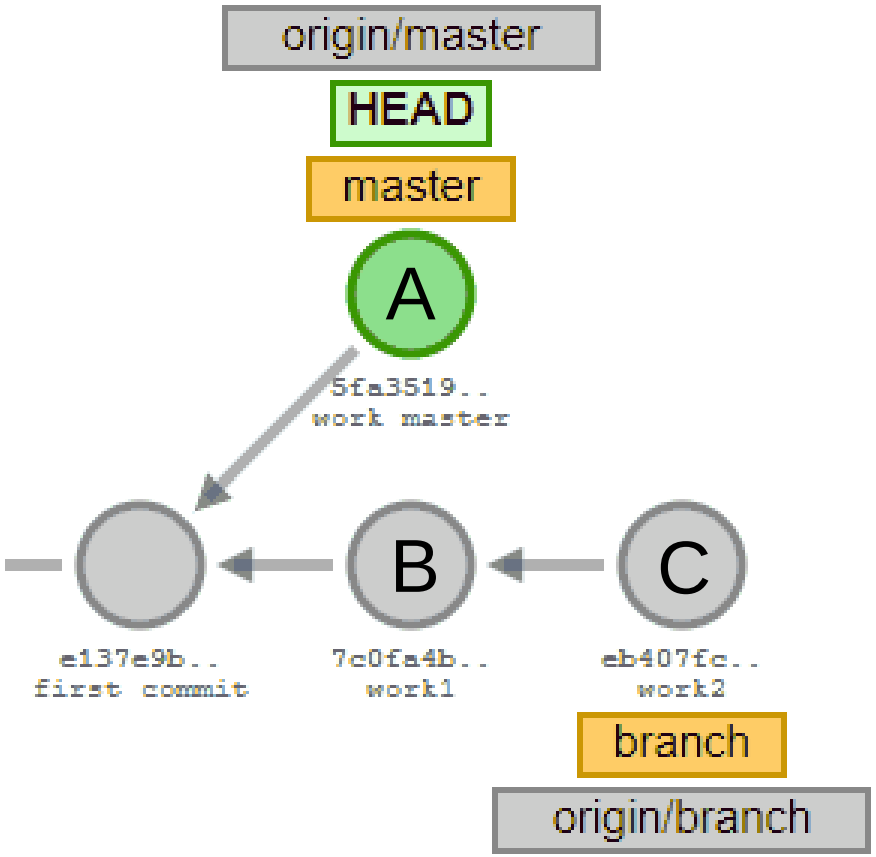
merge

- Fast forward



merge

- Fast forward not possible
- `git merge branch`

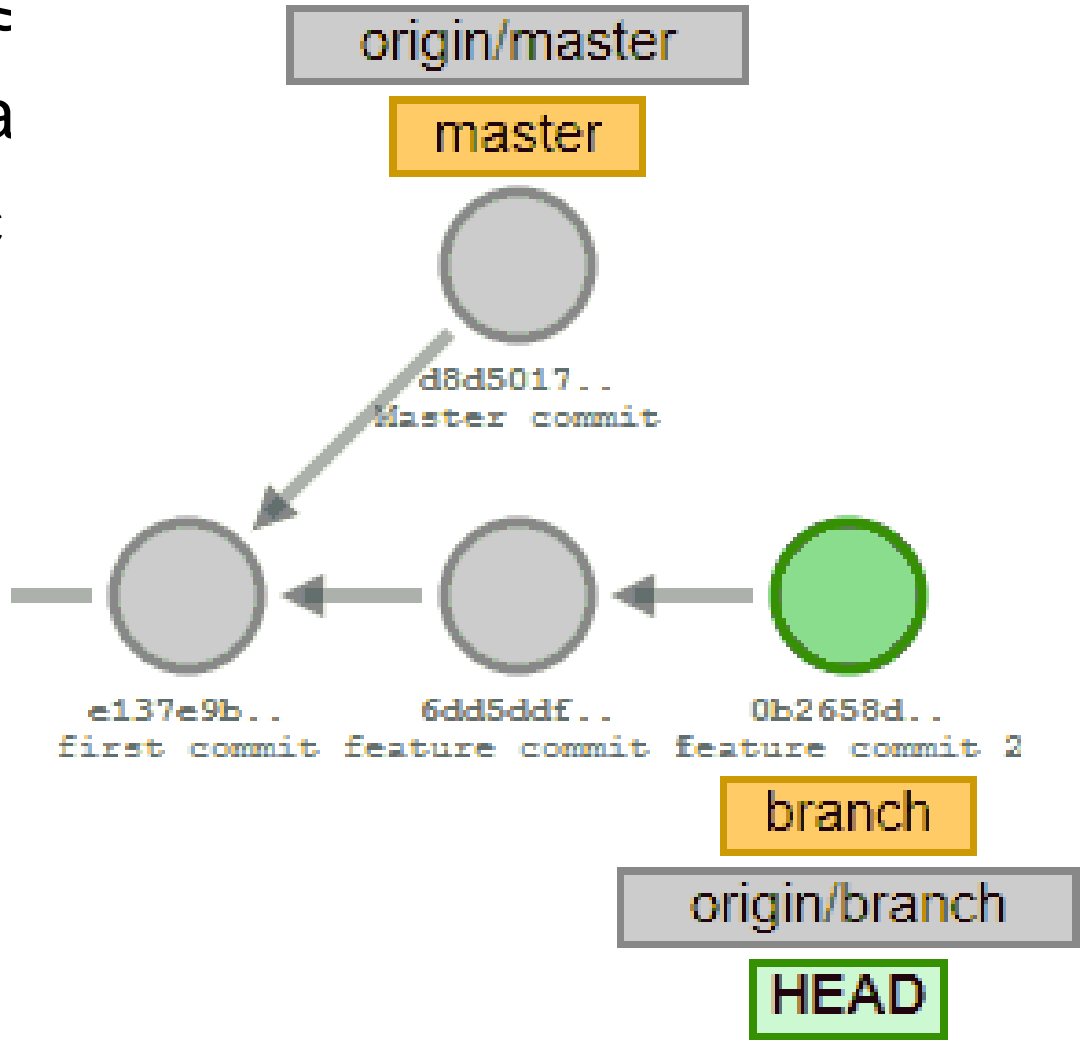


rebase

- Not a substitute for merge!
- A way to always be able to fast forward.
- Assisted changeset re-application.

rebase

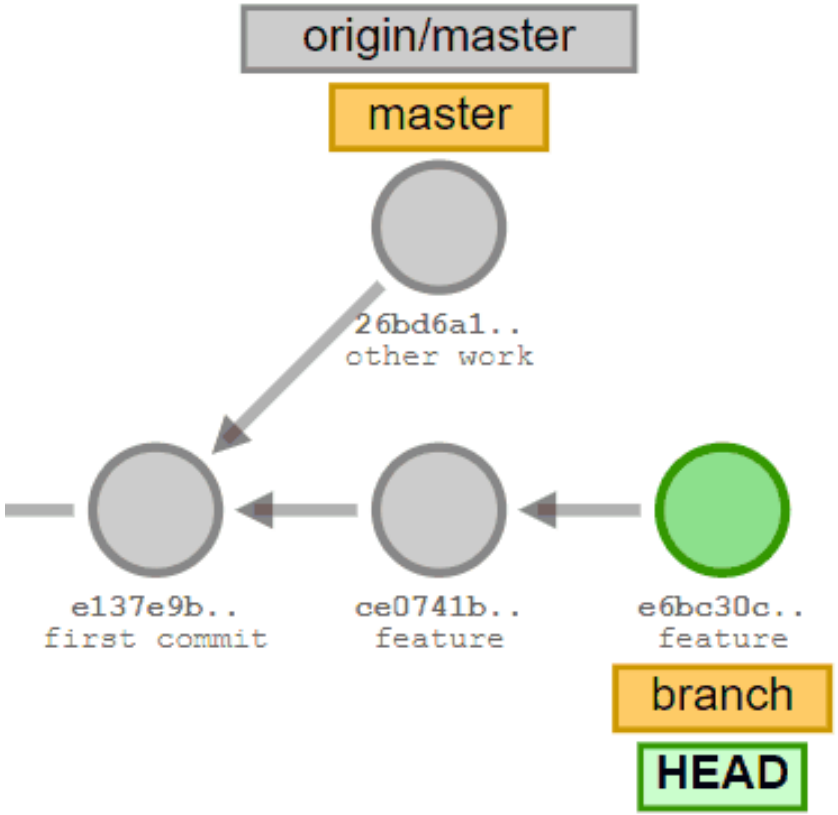
- Not a subs
- A way to a
- Assisted c



Kontent.ai

rebase & merge (fast-forward)

git rebase master



Kontent.ai

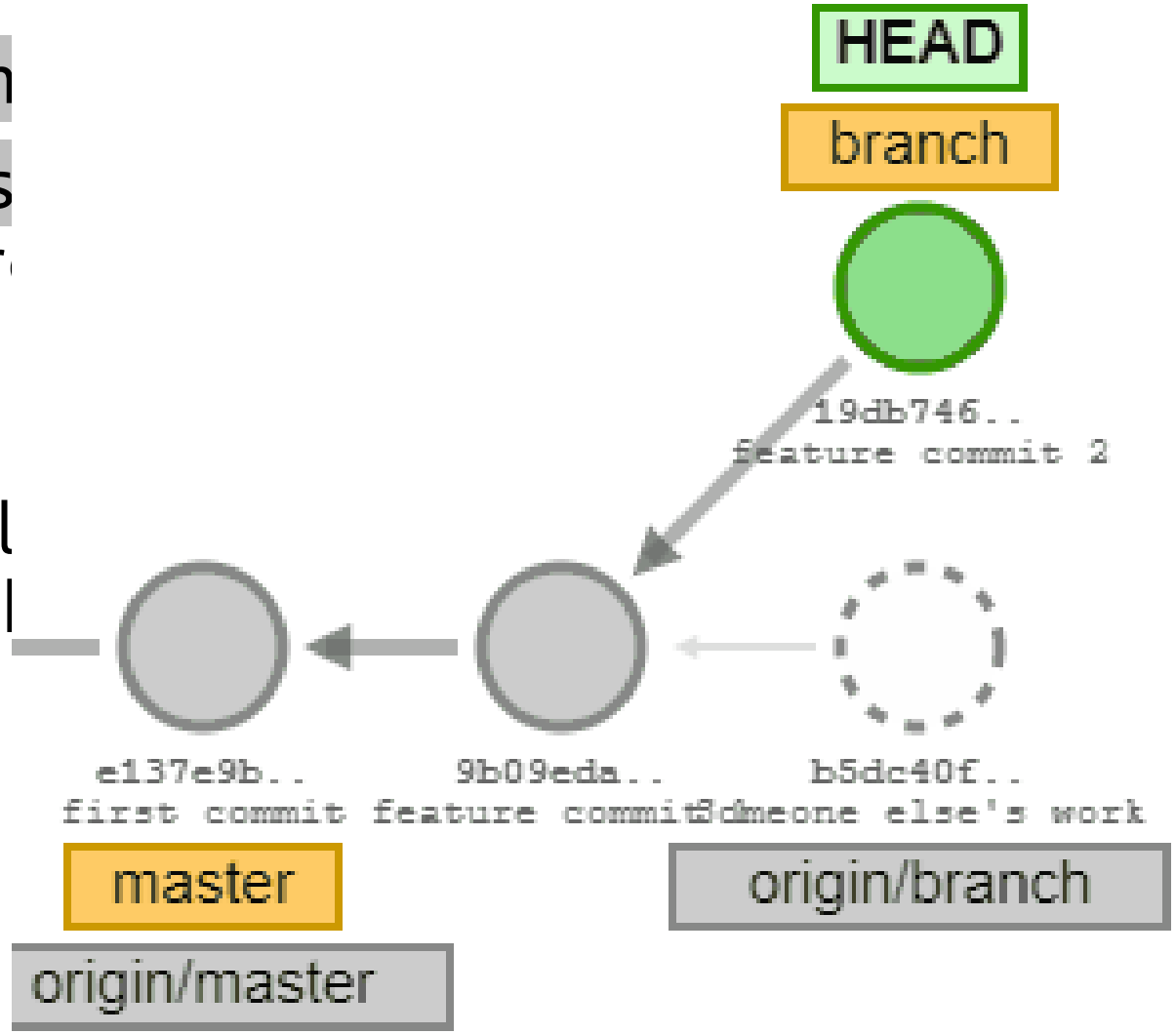
pull --rebase

- It `fetches`
- It `rebases` your local changes onto the most recent commit in the remote branch.

- Basically just:
  git fetch && git rebase "origin/$(git branch --show-current)"

pull --rebase

- It fetch
- It rebas
  in the r
- Basicall
  git fetcl



commit

current)"

Kontent.ai

Kontent.ai

Savepoints

Or when things get complicated


```shell
$ cd ..
$ cp -r work backup_work
$ cd work
```

Savepoints

Or when things get complicated

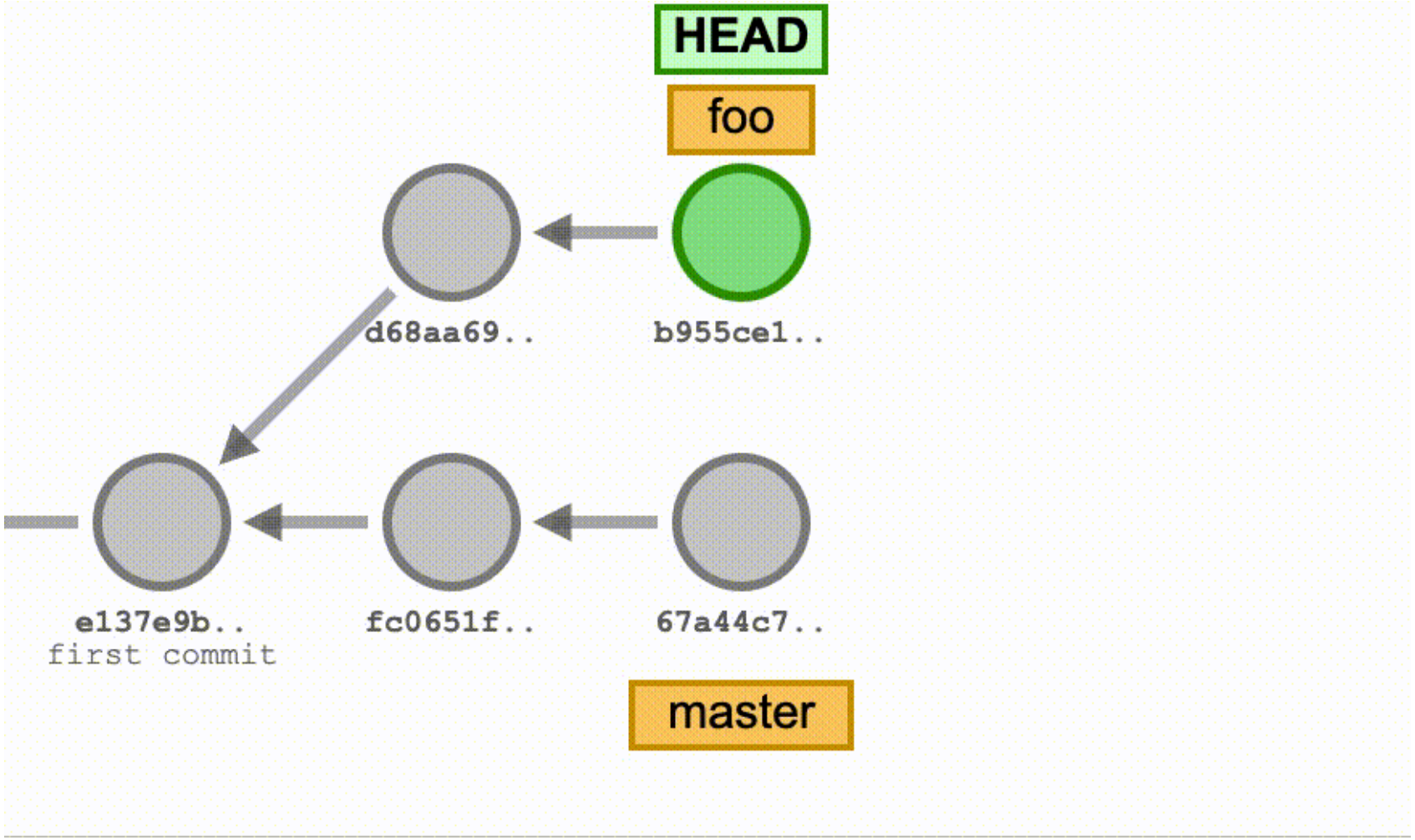
savepoints

1. Repository is a gigantic graph of nodes
   - Git periodically traverses the commits and garbage-collects unreachable ones
2. Operations on commits are immutable
3. References make commits reachable

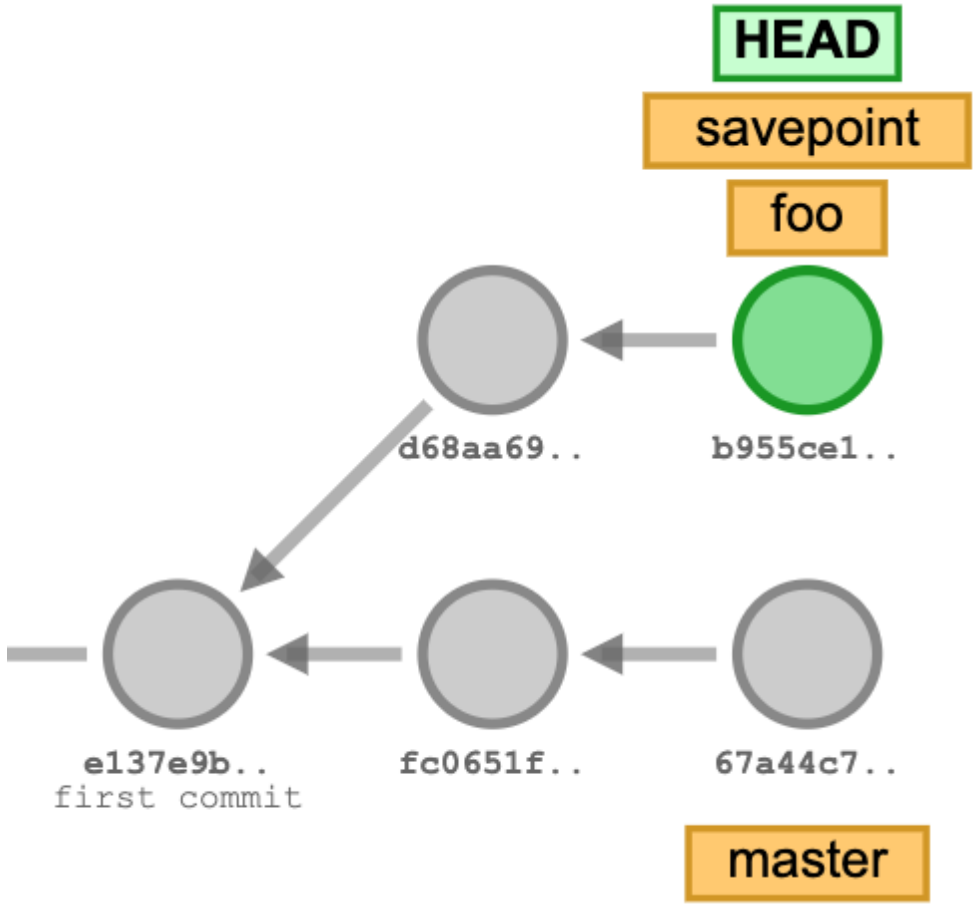
„Creating a branch before you try a merge or a rebase is like saving your game before you battle the boss“

Kontent.ai

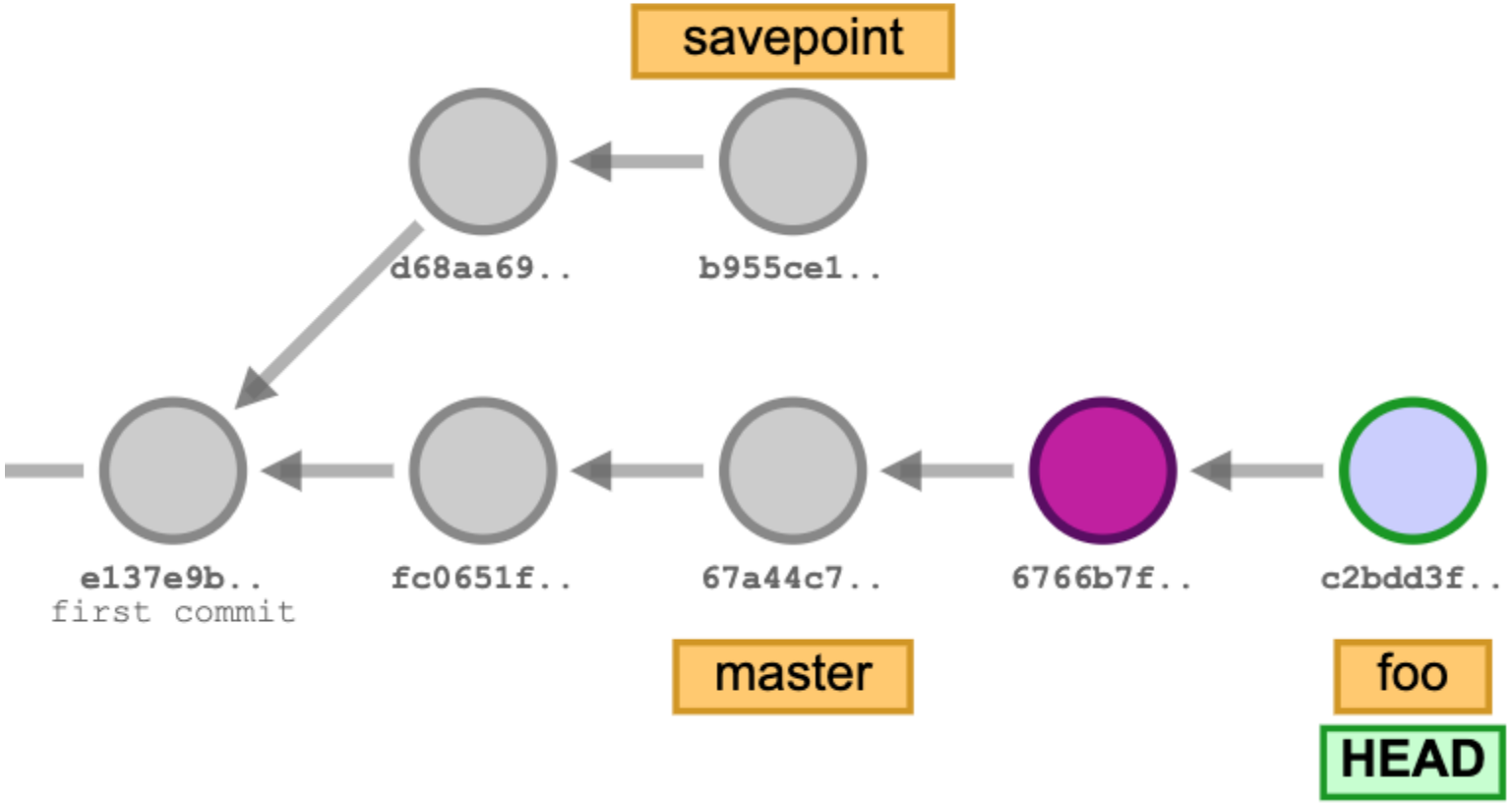
savepoints



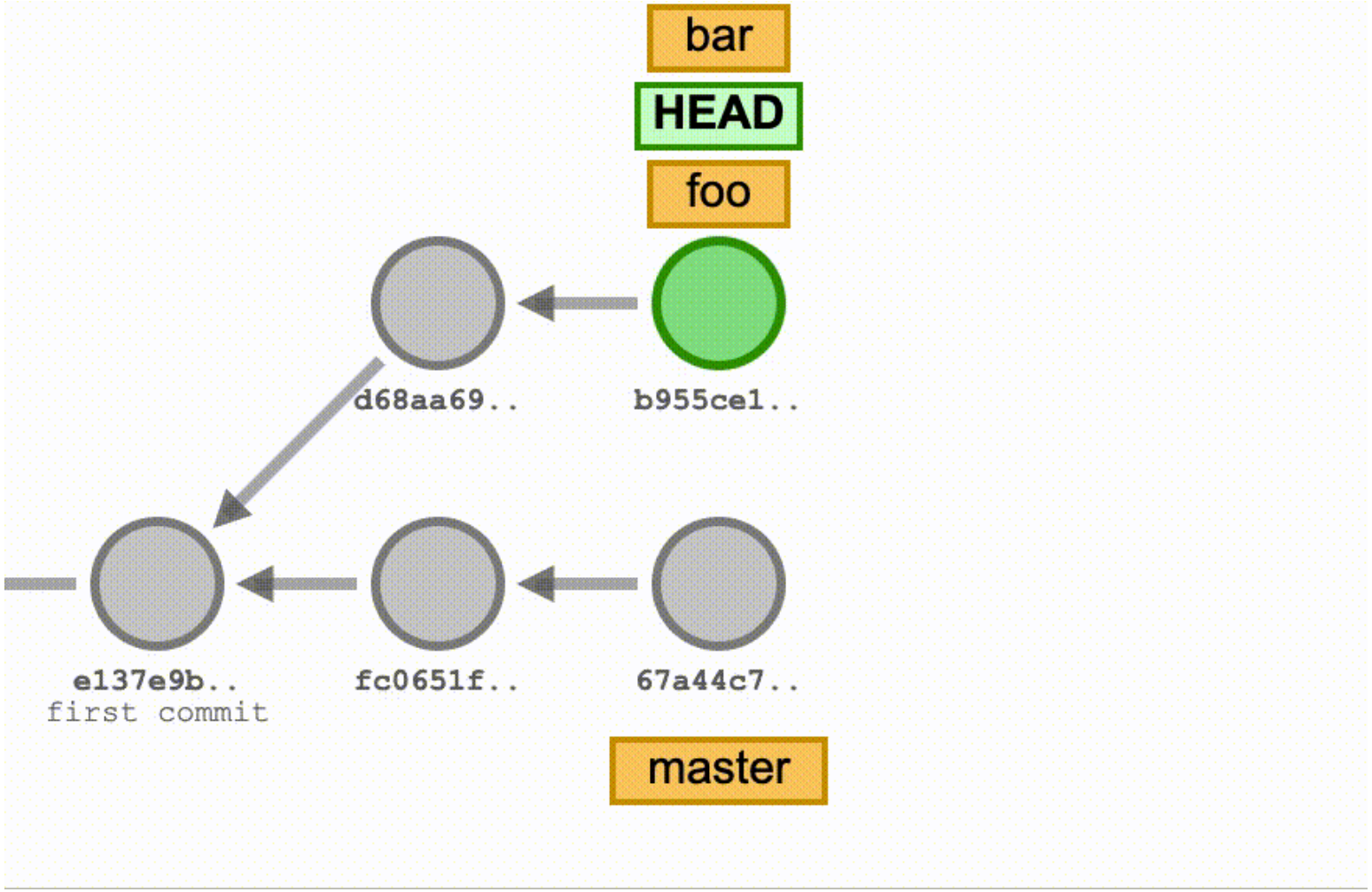
savepoints



savepoints



savepoints



savepoints

Savepoint pattern

- Create a new (*savepoint*) branch
- Do the merge
- Check the graph
- **Ok**: Delete the *savepoint*
- **Nok**: Move the branch pointer back to the *savepoint*

Scout pattern

- Create a new (*scout*) branch and **switch to it**
- Do the merge
- Check the graph
- **Ok**: Move previous branch forward where the *scout* branch is
- **Nok**: Delete the *scout* branch

Kontent.ai

Interactive rebase

Interactive rebase

- **why?** to clean up history (typically on feature (private) branch)
- **what?** edit, reorder, squash, split, delete commits, etc.
- **how?** using a text editor choose an action for each commit


GIT SQUASH?
THAT JUST SOUNDS LIKE FIXUP WITH EXTRA STEPS
imgflip.com

Kontent.ai

And that's it...

Let's recap

Takeaways

- Rebase and merge
  - Don't be afraid of conflict resolution, you're just re-applying the same changes. Same intentions – your intentions.
- Visualize your repository before you run any command
  - Remember A DOG `git log --all --decorate --oneline --graph`.
- References make commits reachable
  - As long as there is a pointer to a commit, the state can be safely changed back to this state.