

Git Workflow Best practices



<http://git-scm.com>

Anar Manafov, GSI Darmstadt
ROOT Users Workshop 2013

History

patches and tarballs → BitKeeper → Git

Git was initially designed and developed by **Linus Torvalds**:
“I'm an egotistical bastard, and I name all my projects after myself.”

(Git - British English slang for a stupid or unpleasant person)

Design

Design

Torvalds had several design criteria:

Design

Torvalds had several design criteria:

- Take CVS as an example of what not to do.

Design

Torvalds had several design criteria:

- Take CVS as an example of what not to do.
- Support a distributed, BitKeeper-like workflow.

Design

Torvalds had several design criteria:

- Take CVS as an example of what not to do.
- Support a distributed, BitKeeper-like workflow.
- Very strong safeguards against corruption, either accidental or malicious.

Design

Torvalds had several design criteria:

- Take CVS as an example of what not to do.
- Support a distributed, BitKeeper-like workflow.
- Very strong safeguards against corruption, either accidental or malicious.
- Very high performance.

Free and Open Source



GPLv2 FREE SOFTWARE

Small and Fast

Let's see how common operations stack up against Subversion, a common centralized version control system that is similar to CVS or Perforce. *Smaller is faster.*



All of these times are in seconds.

Operation	Git	SVN	Ratio
Commit Files (A)	0.64	2.60	4x
Commit Images (B)	1.53	24.70	16x
Diff Current	0.25	1.09	4x
Diff Recent	0.25	3.99	16x
Diff Tags	1.17	83.57	71x
Log (50)	0.01	0.38	31x
Log (All)	0.52	169.20	325x
Log (File)	0.60	82.84	138x
Update	0.90	2.82	3x
Blame	1.91	3.04	1x

For testing, large AWS instances were set up in the same availability zone. Git and SVN were installed on both machines, the Ruby repository was copied to both Git and SVN servers, and common operations were performed on both.

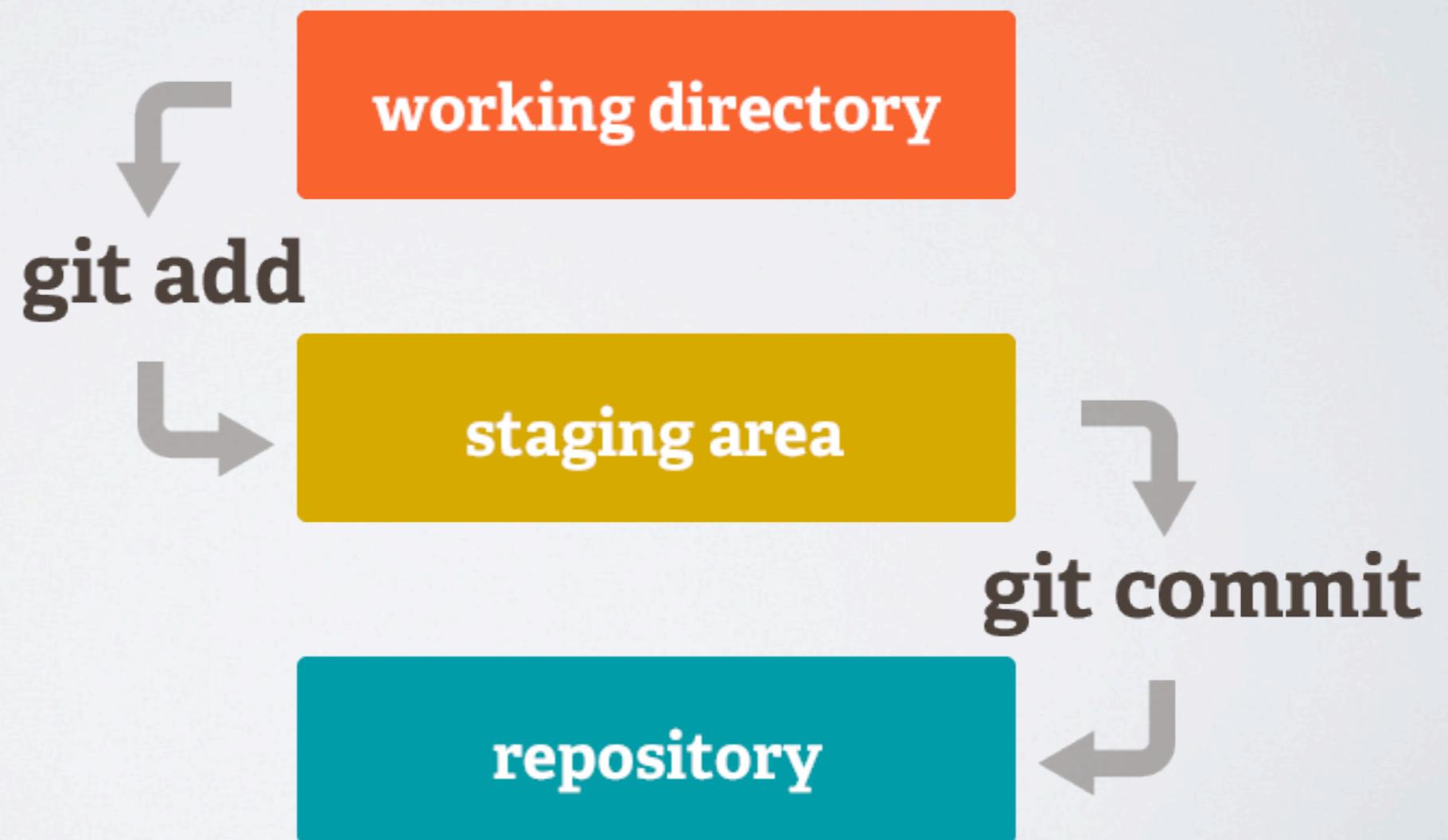
Data Assurance

Every file and commit is checksummed
and retrieved by its checksum when checked back out.

It's impossible to get anything out of Git other than the exact bits you
put in.

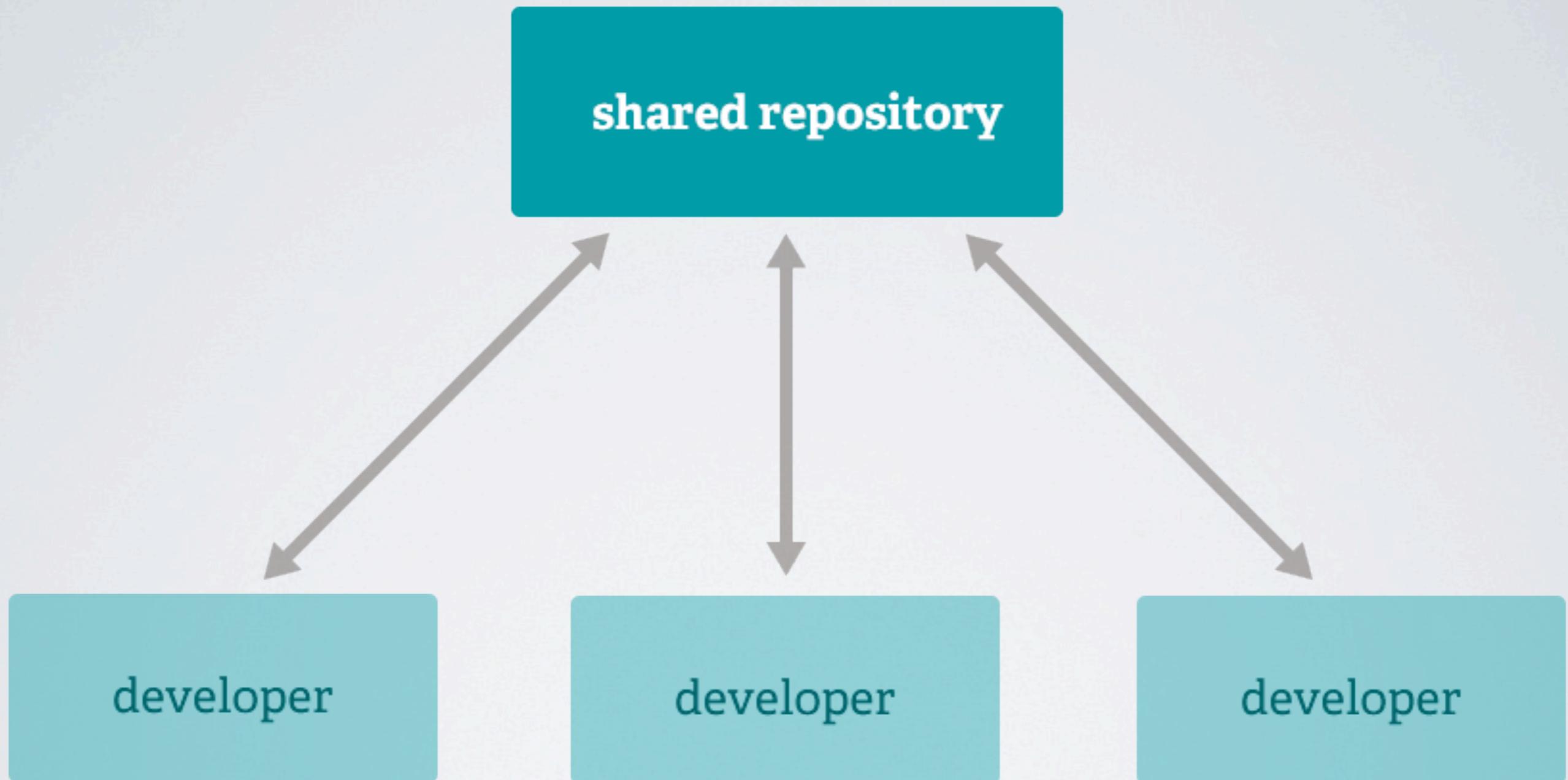
Staging Area

- Staging and Unstaging Files.
- Staging Patches.

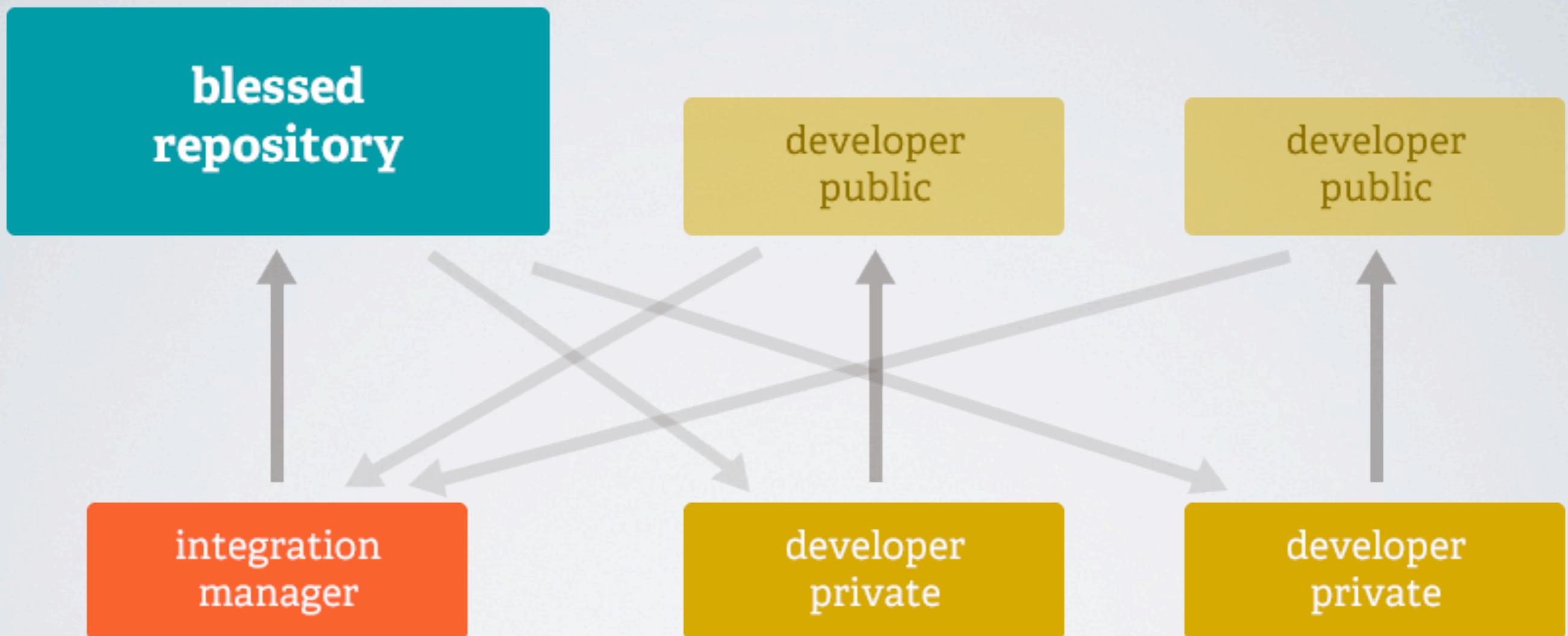


Distributed

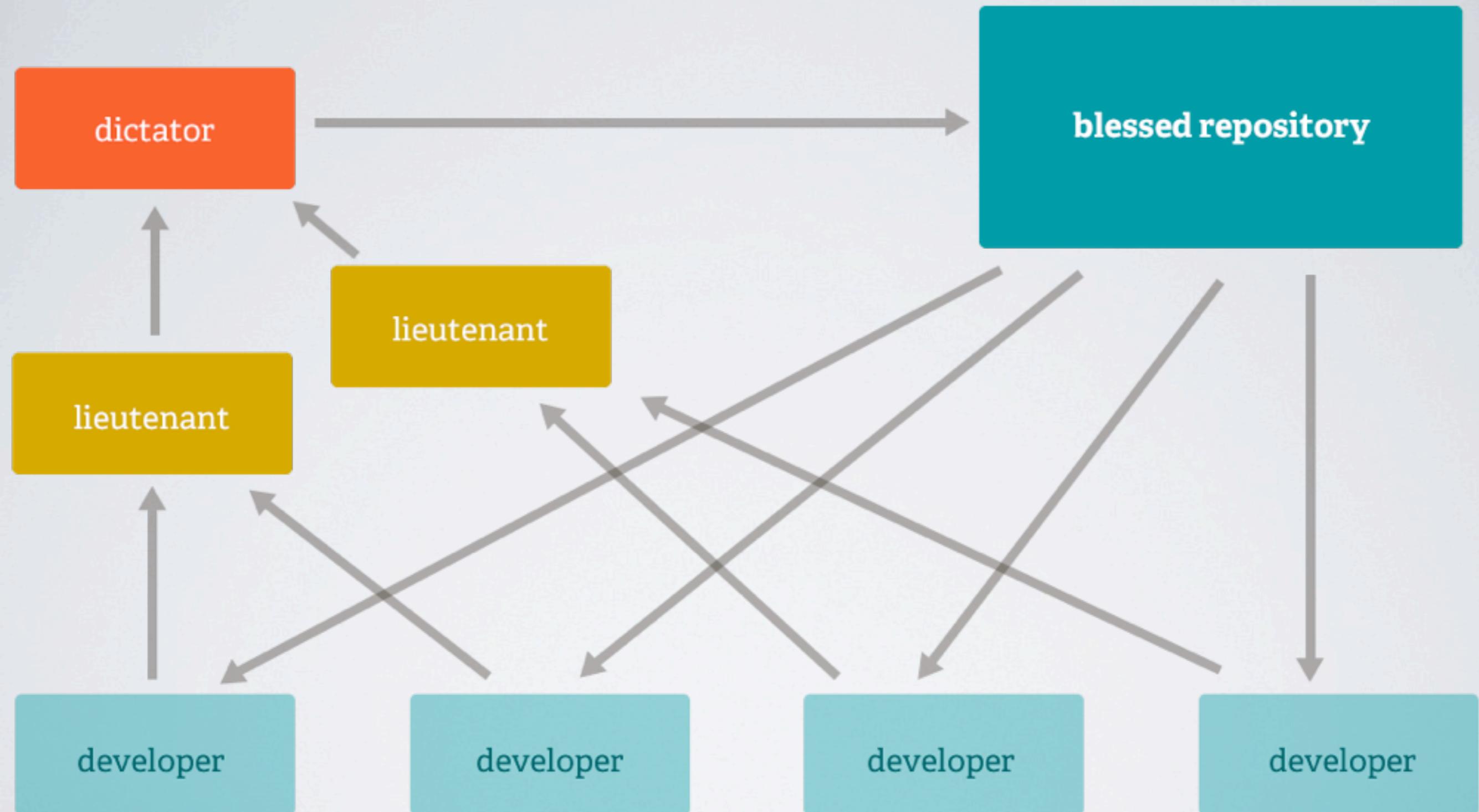
Subversion-Style Workflow



Integration Manager Workflow

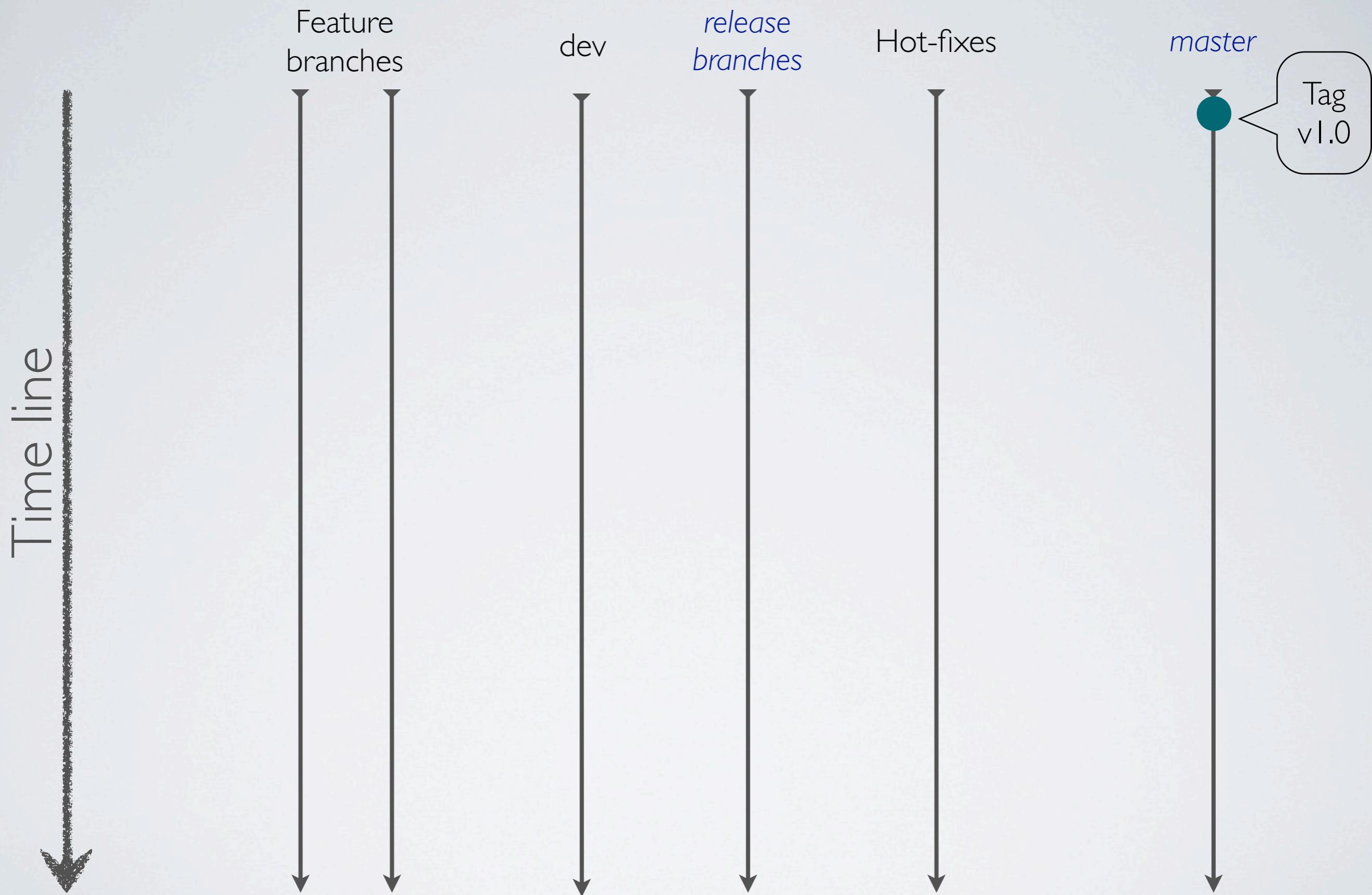


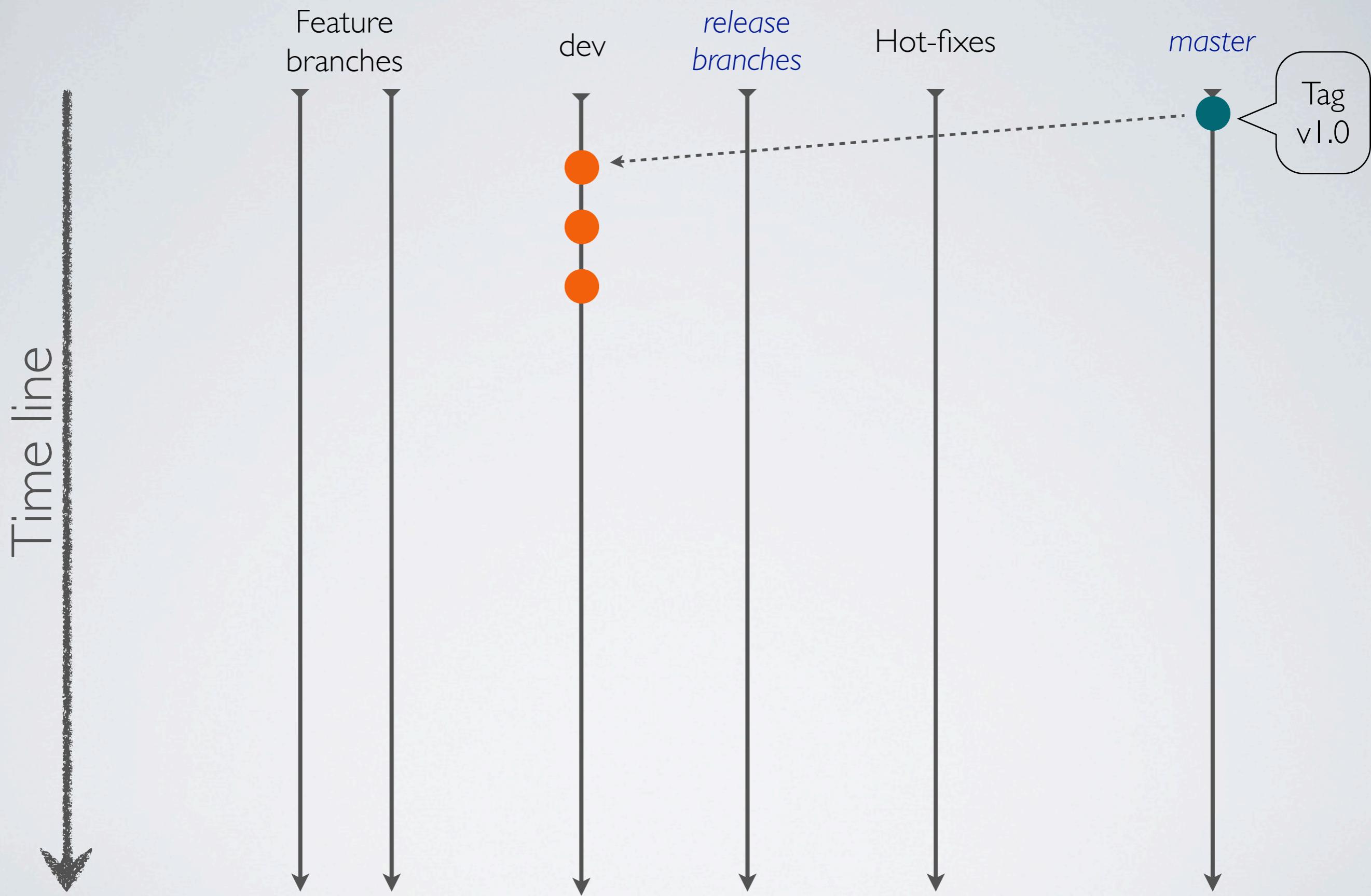
Dictator and Lieutenants Workflow

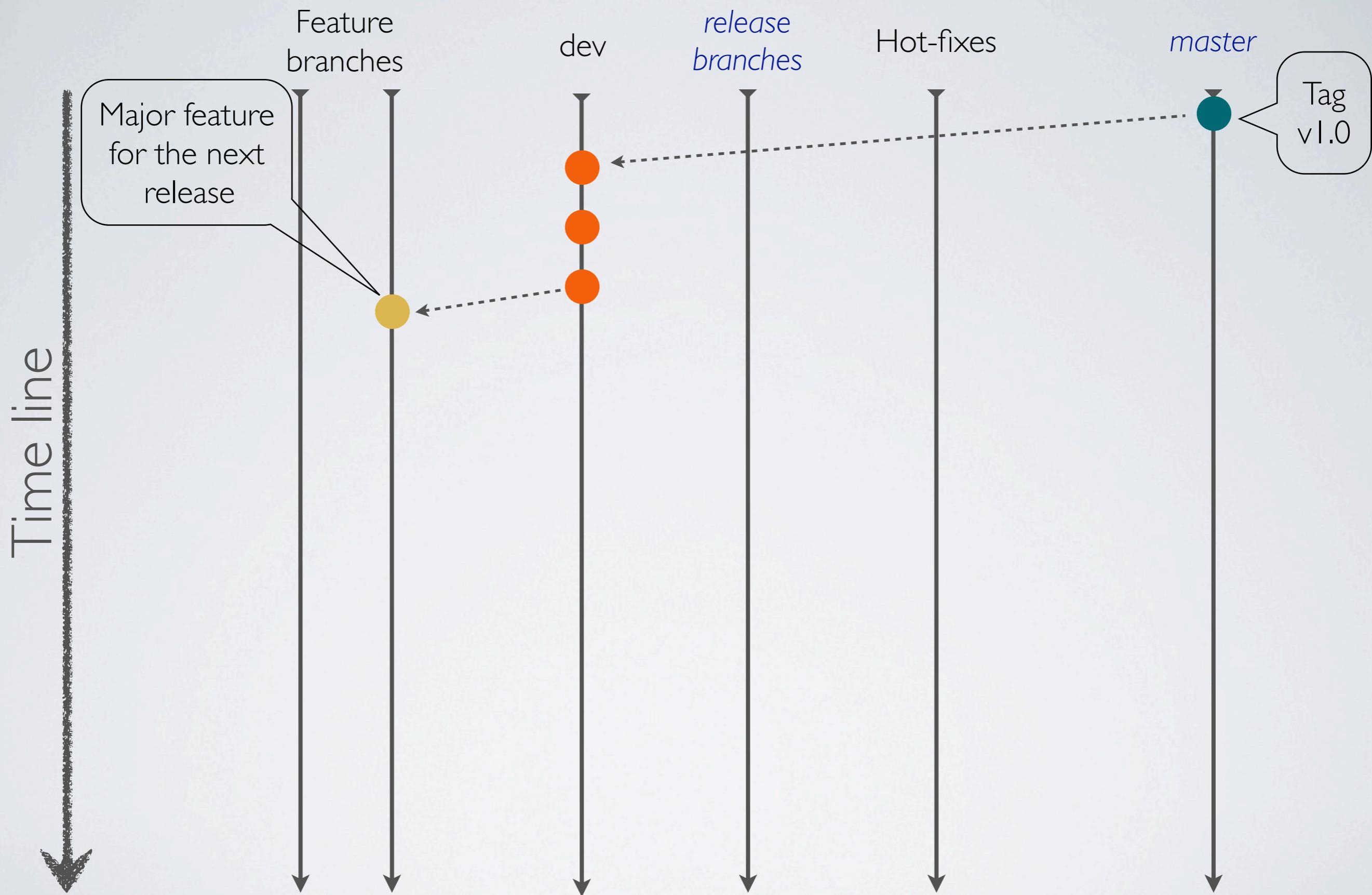


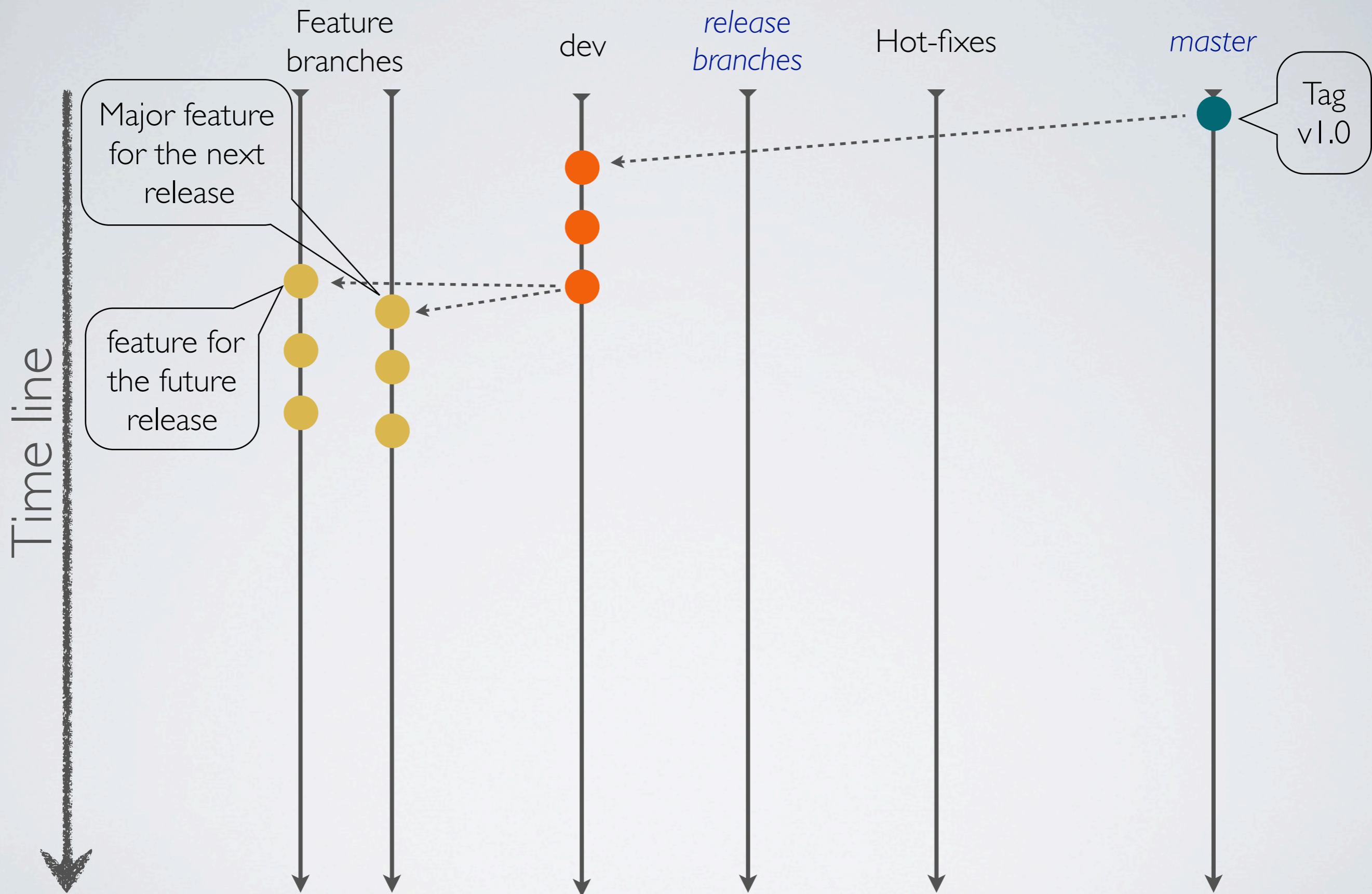
Branching and Merging

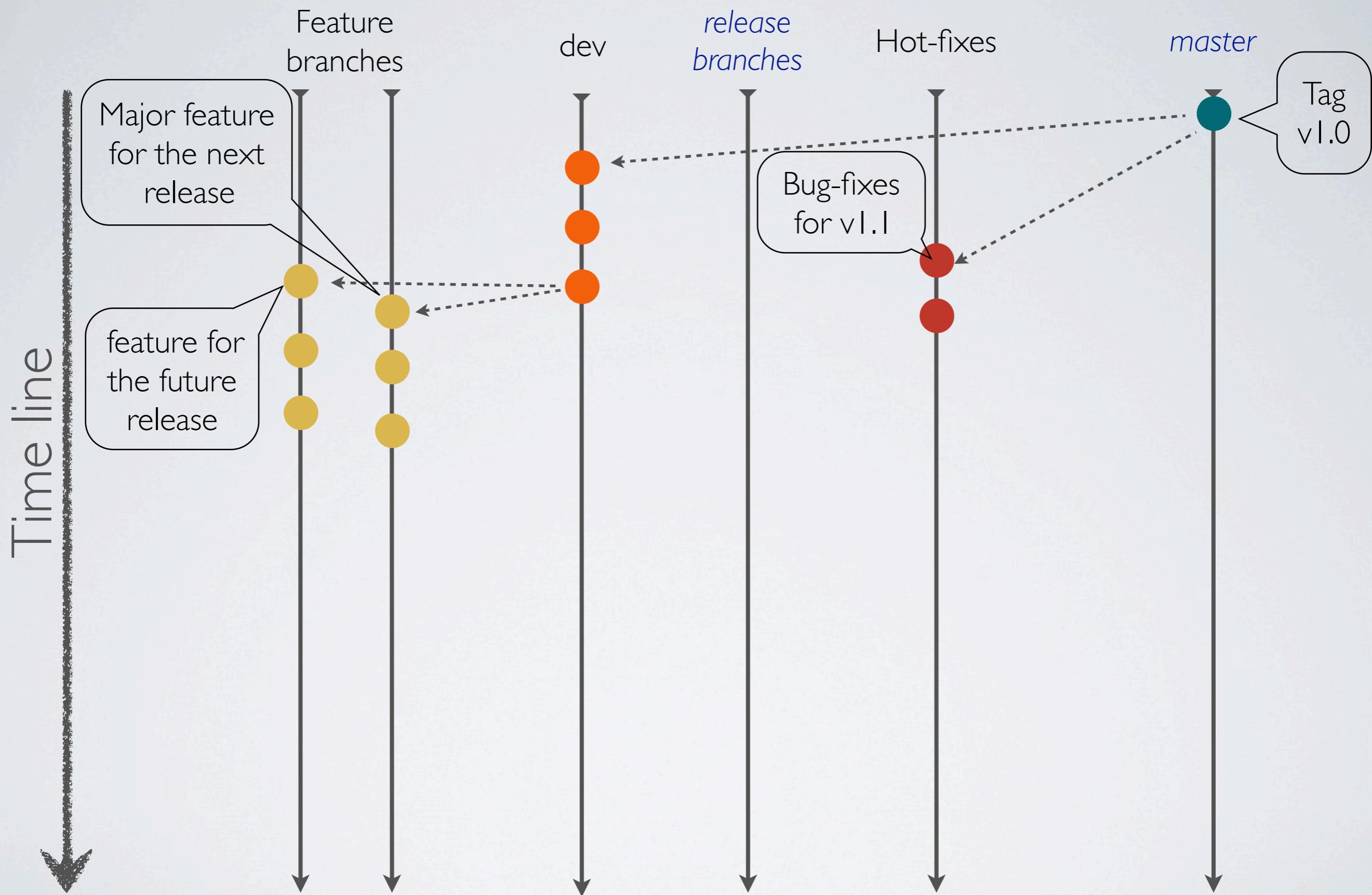
- Frictionless Context Switching.
- Role-Based Codelines.
- Feature Based Workflow.
- Disposable Experimentation.

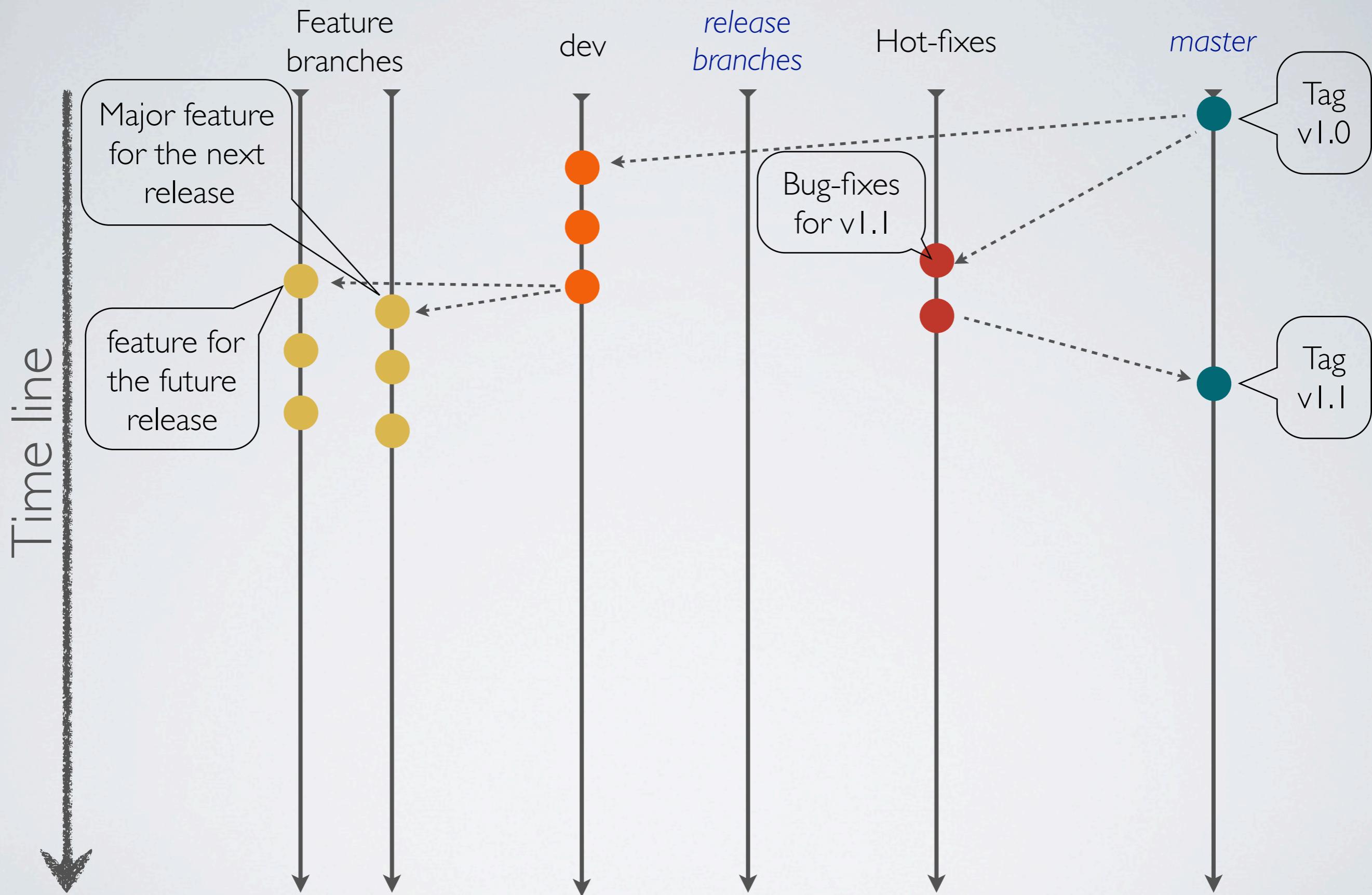


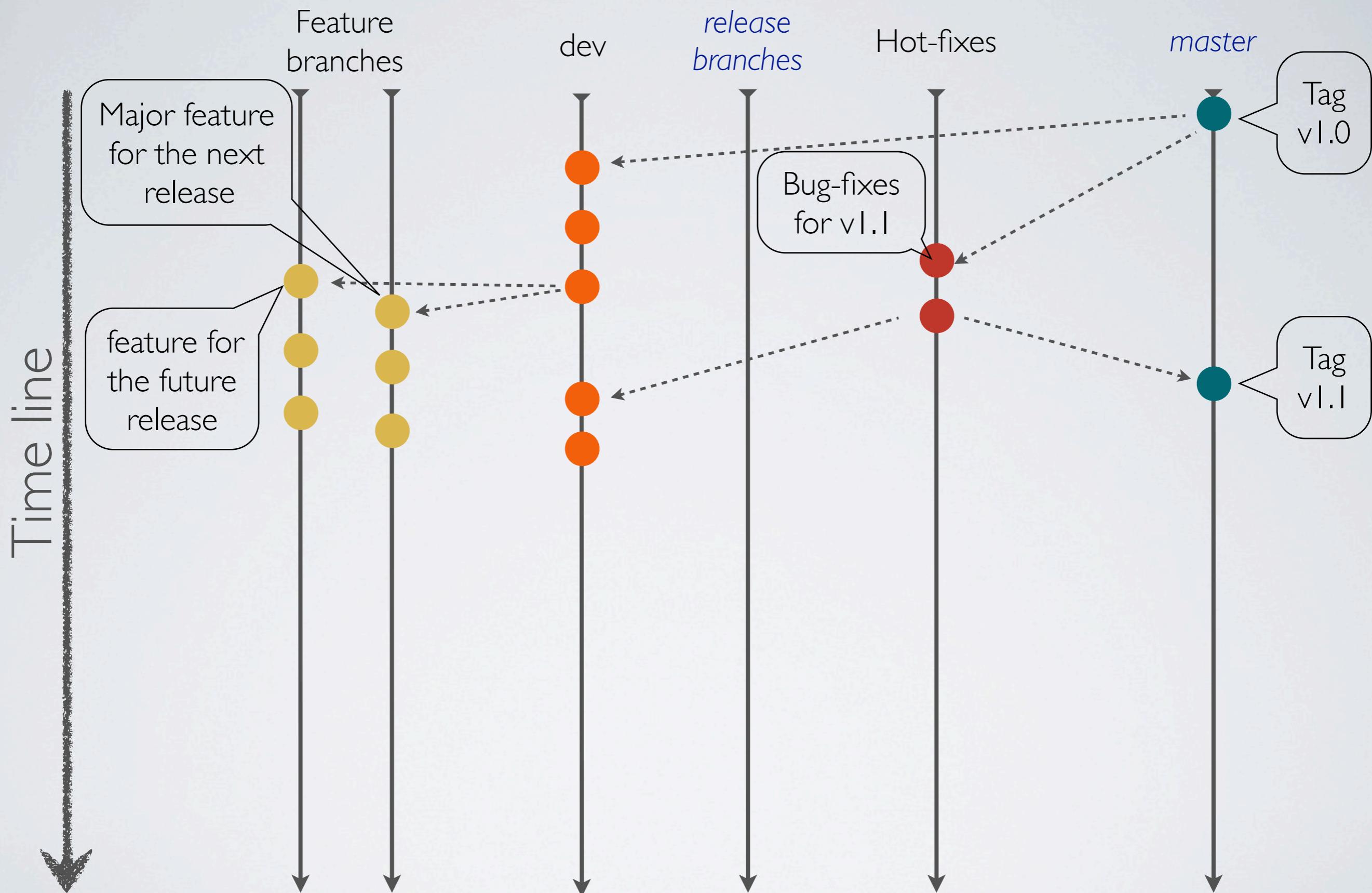


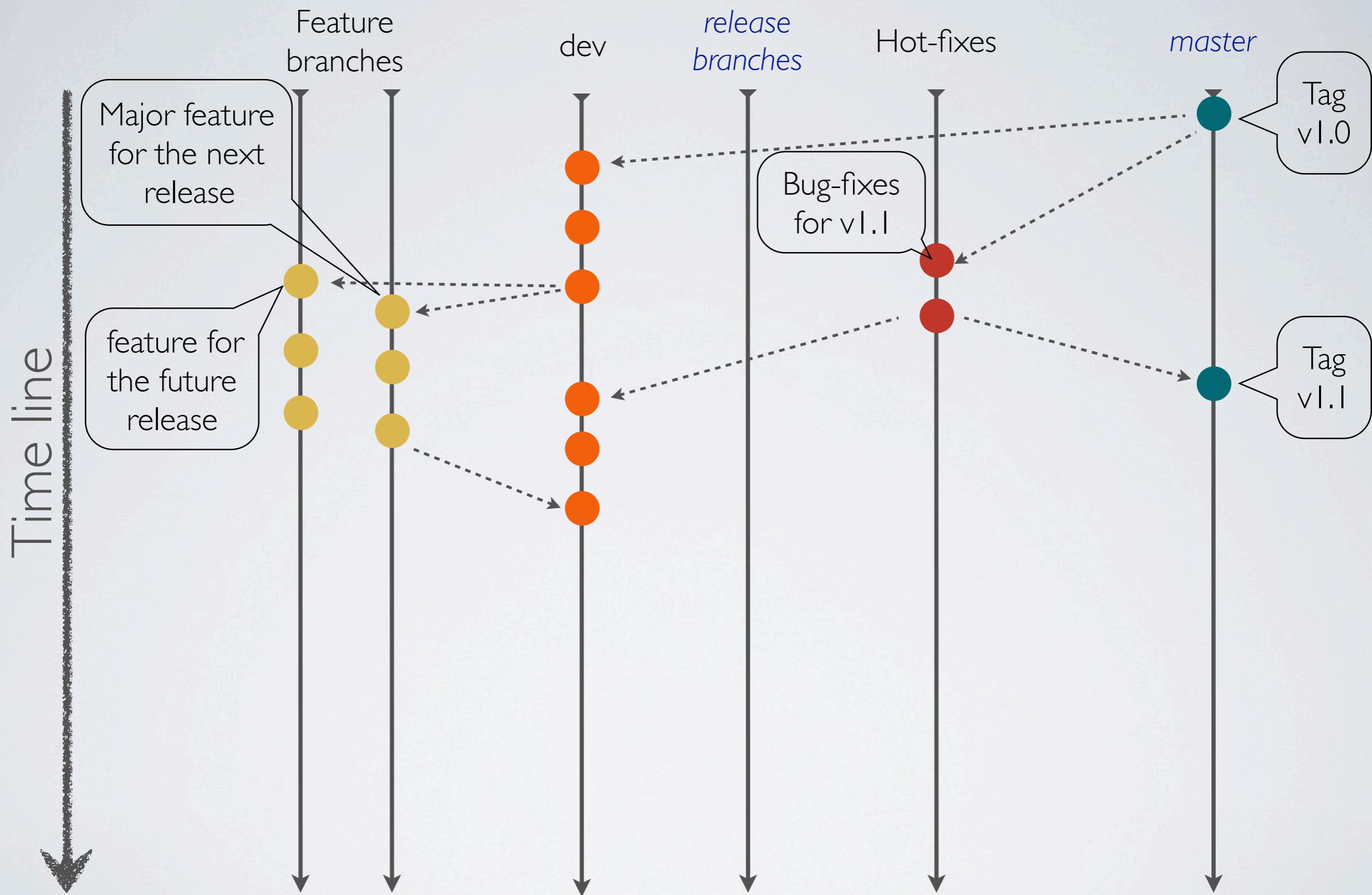


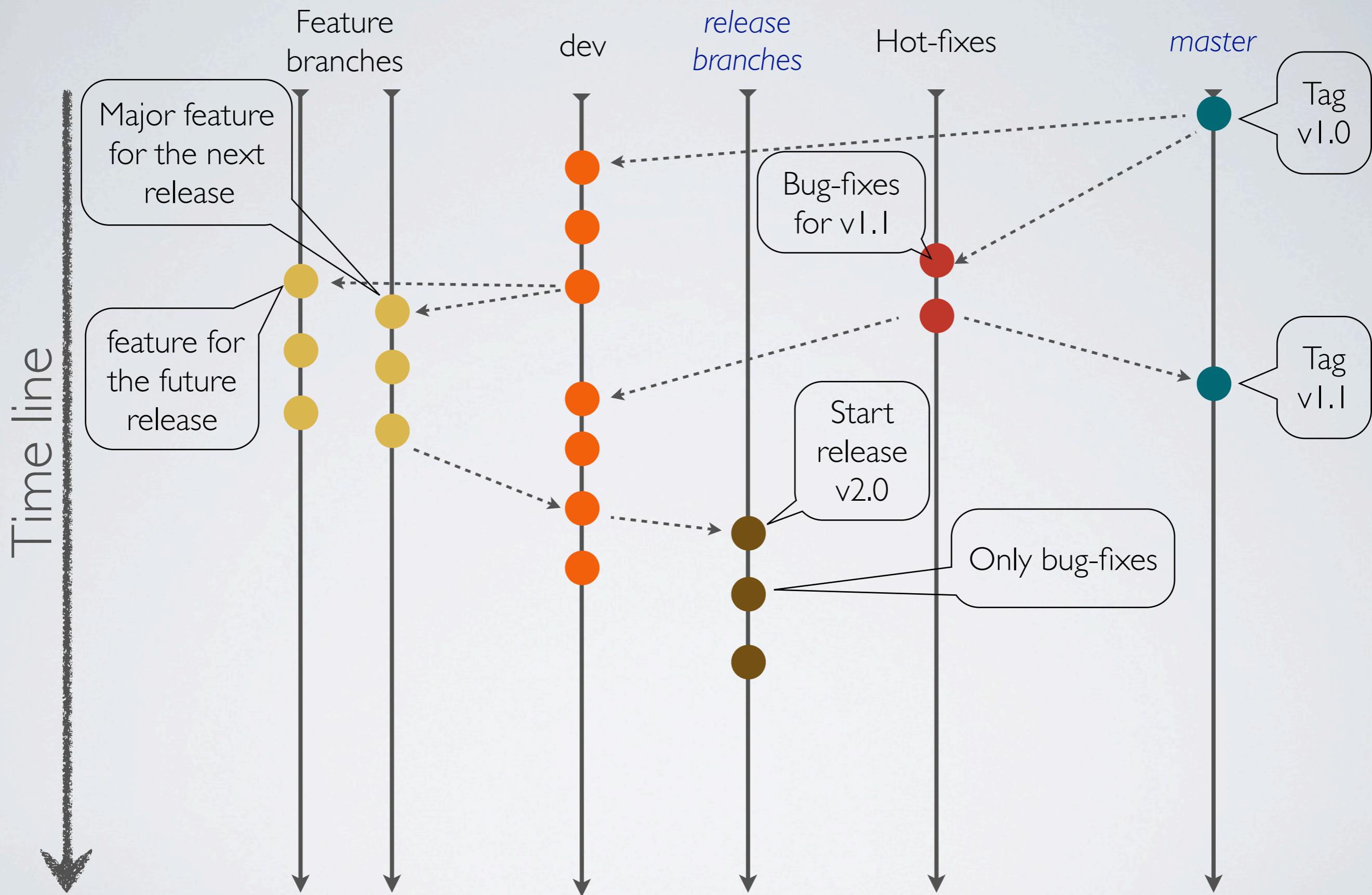


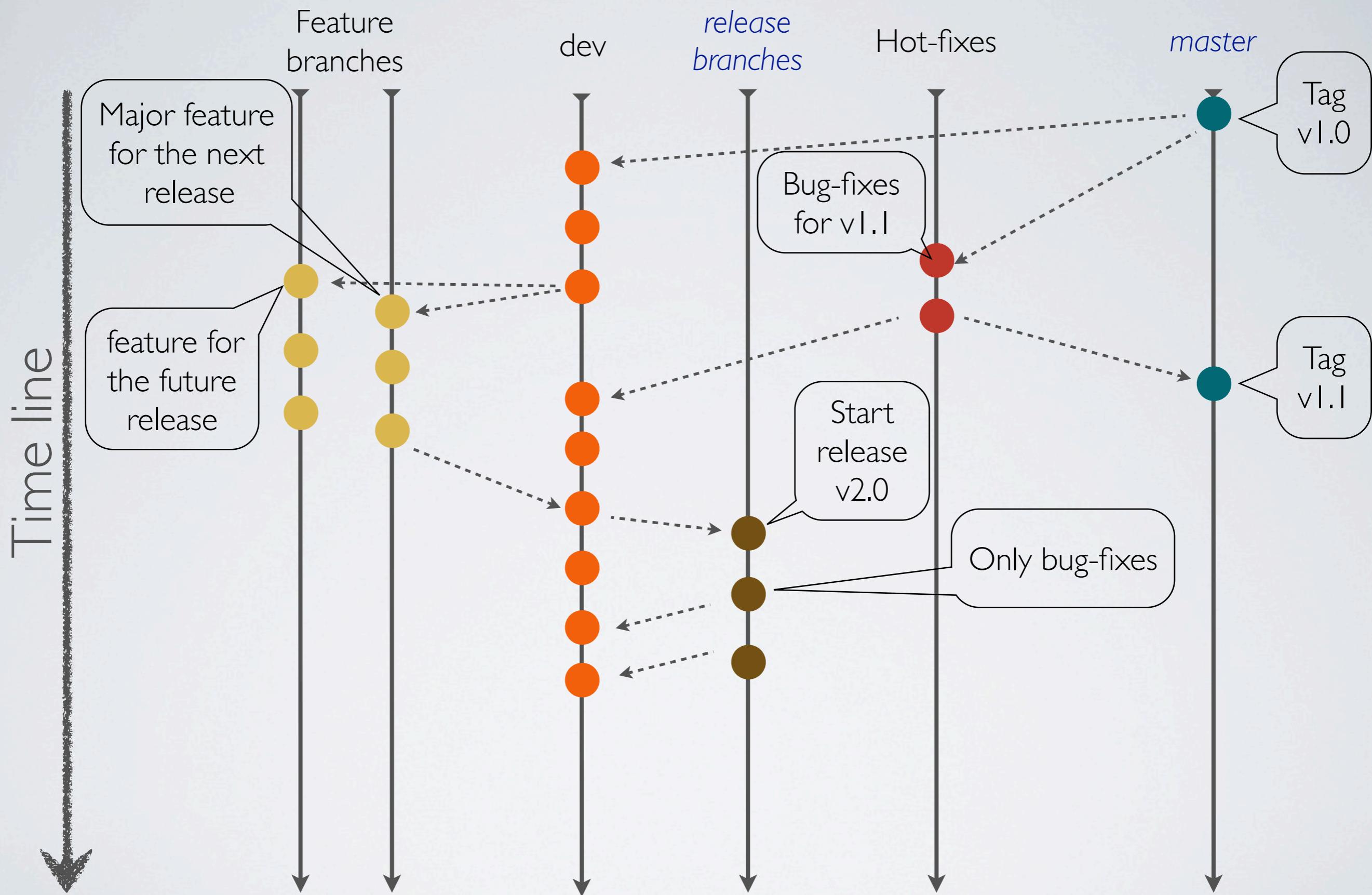


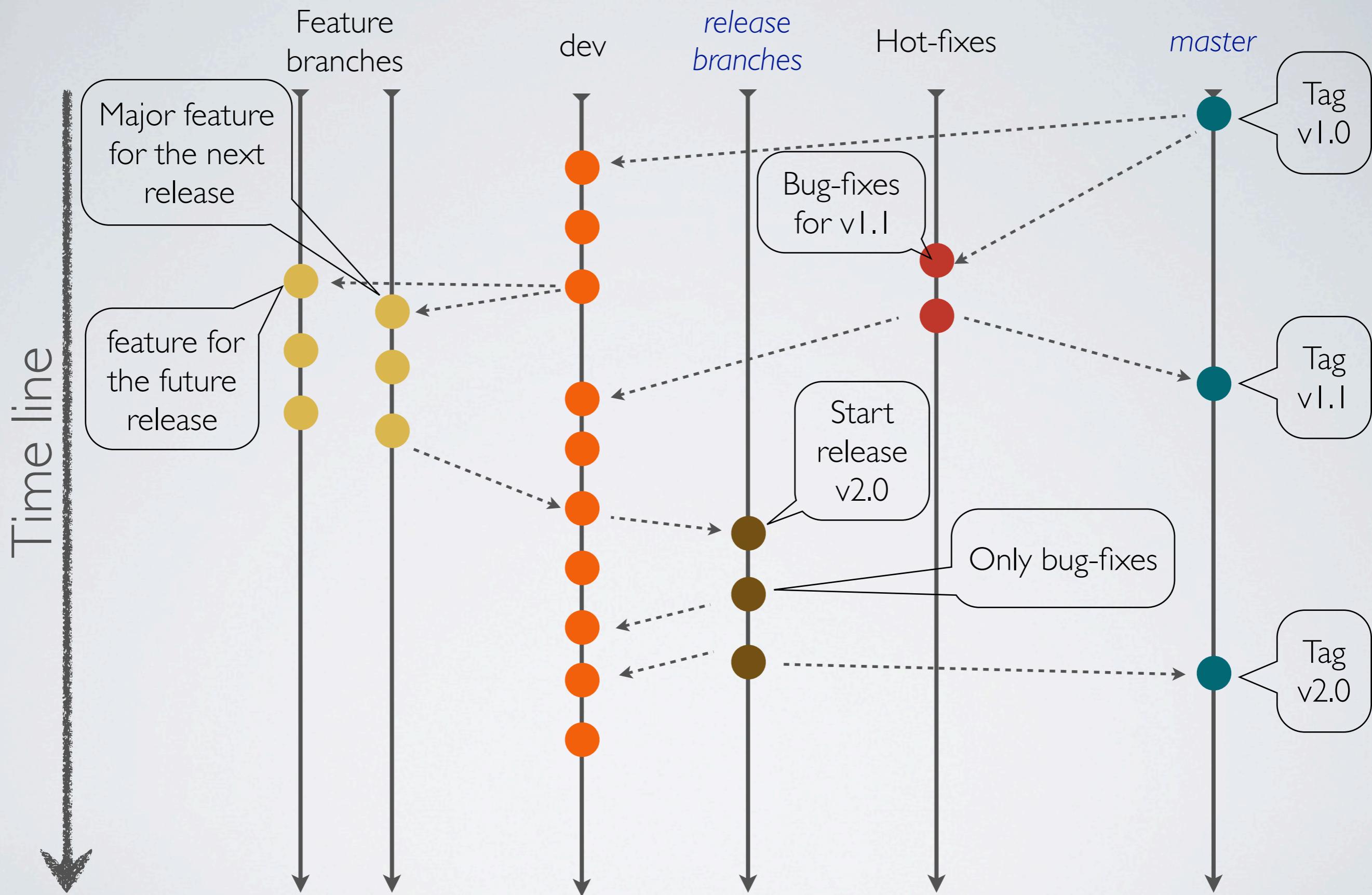


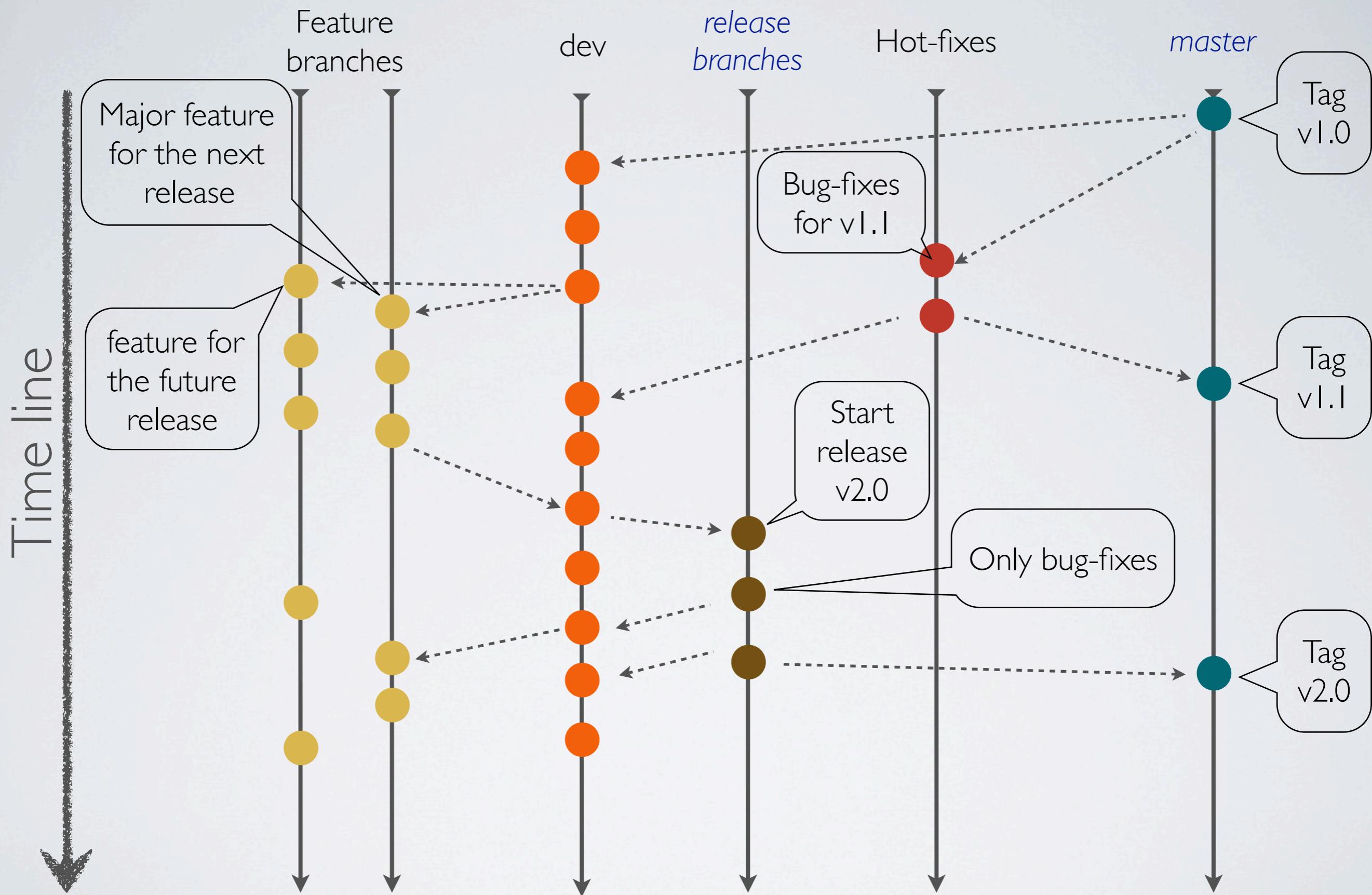


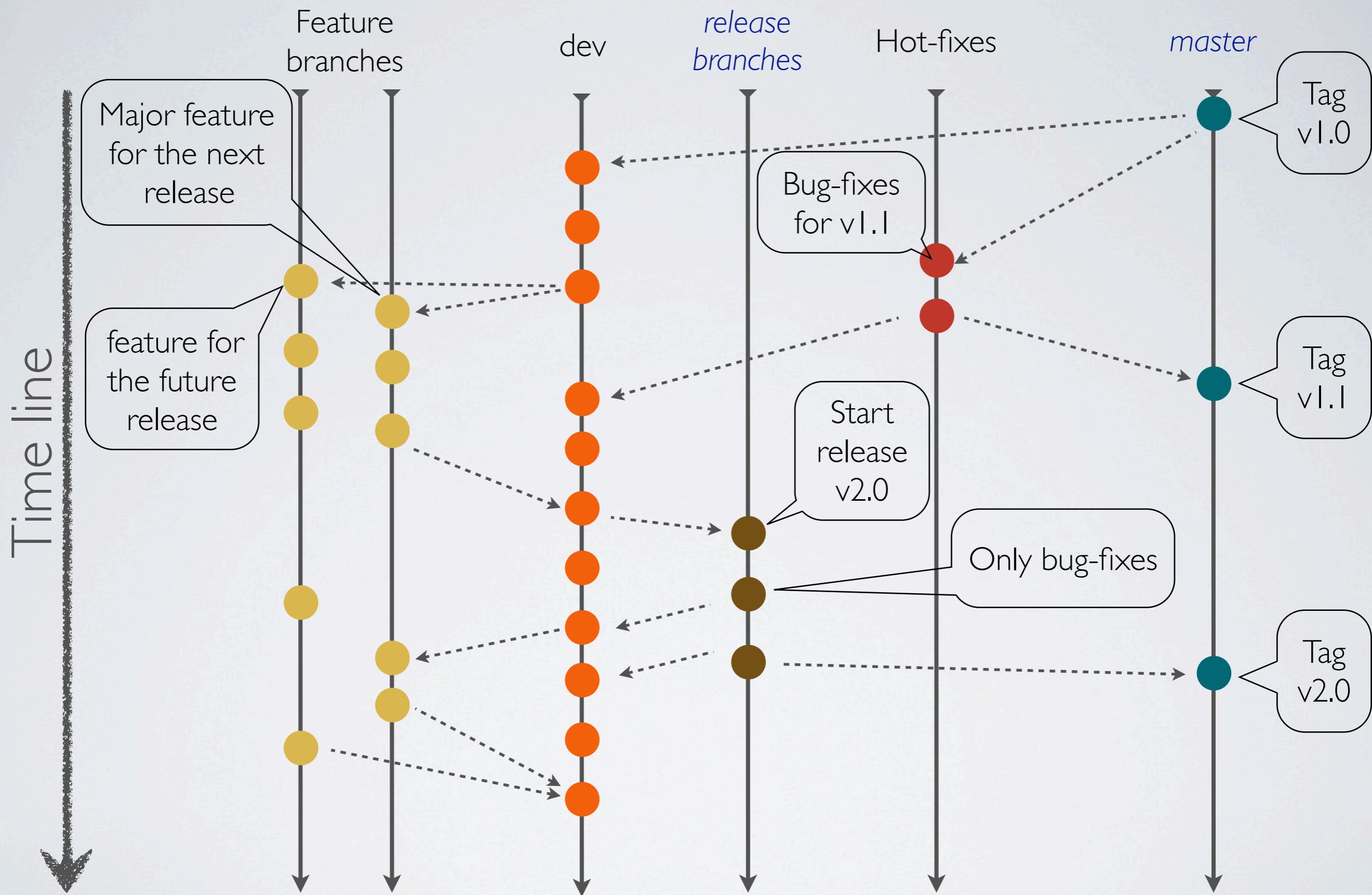


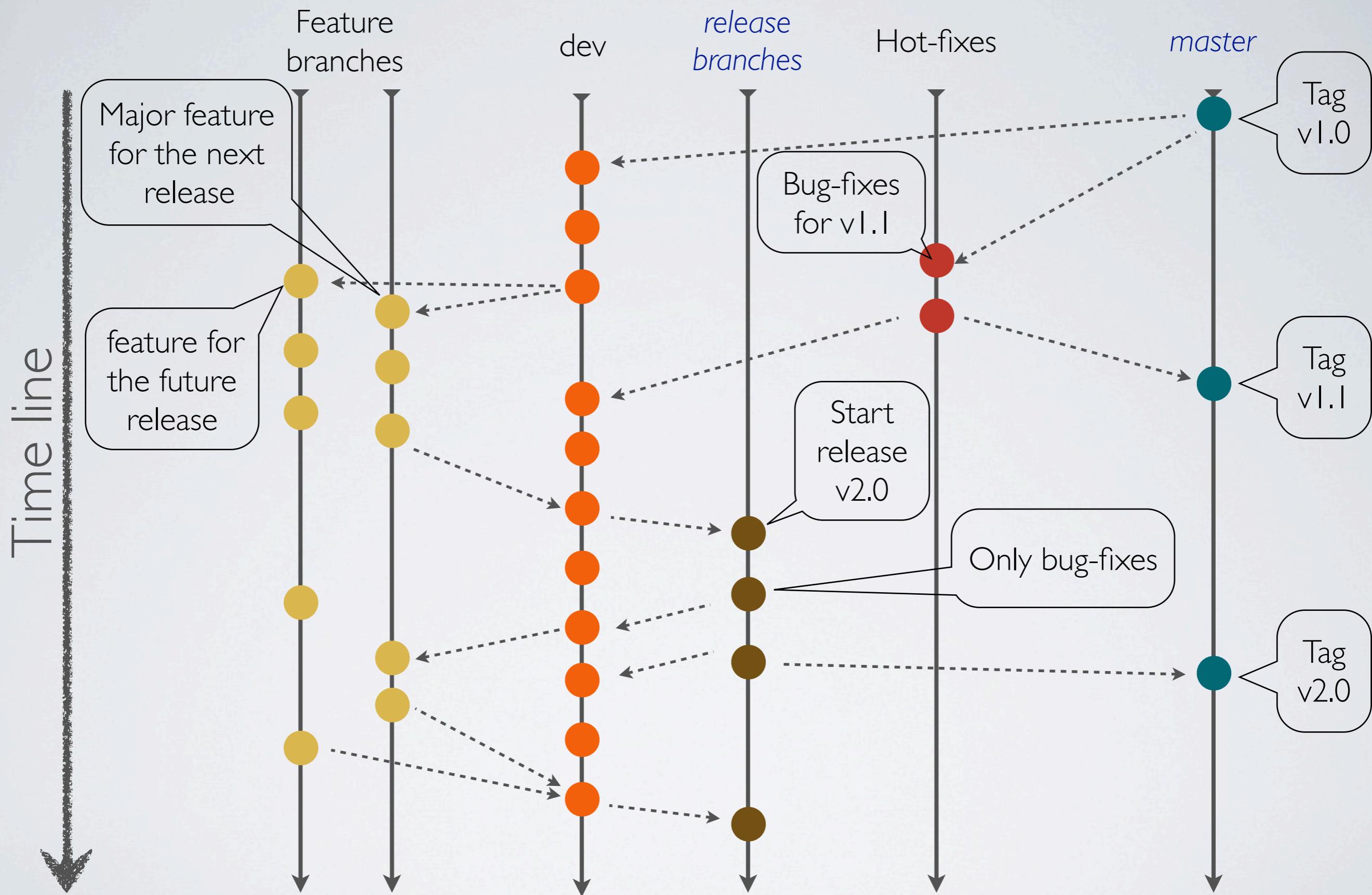


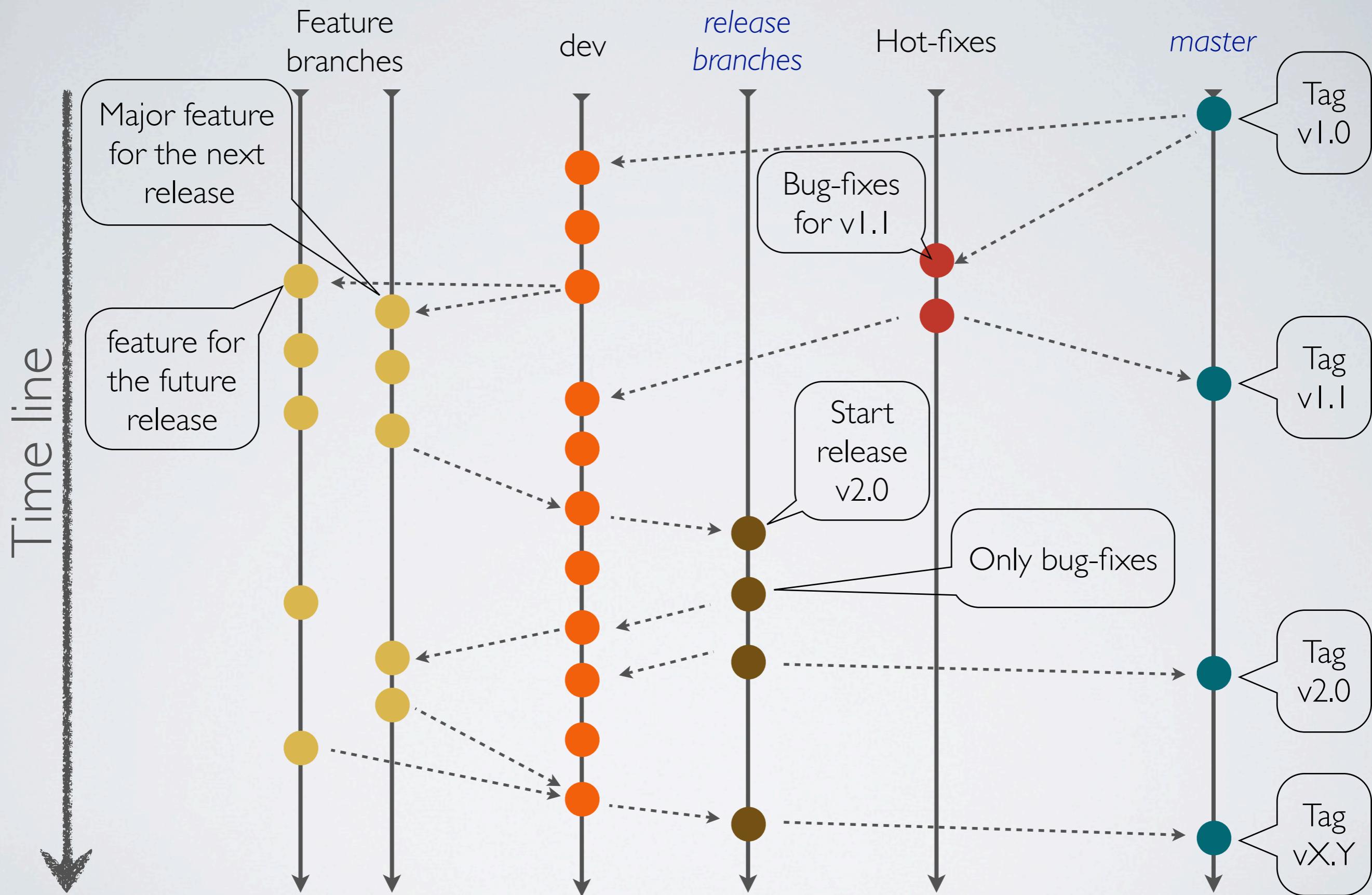


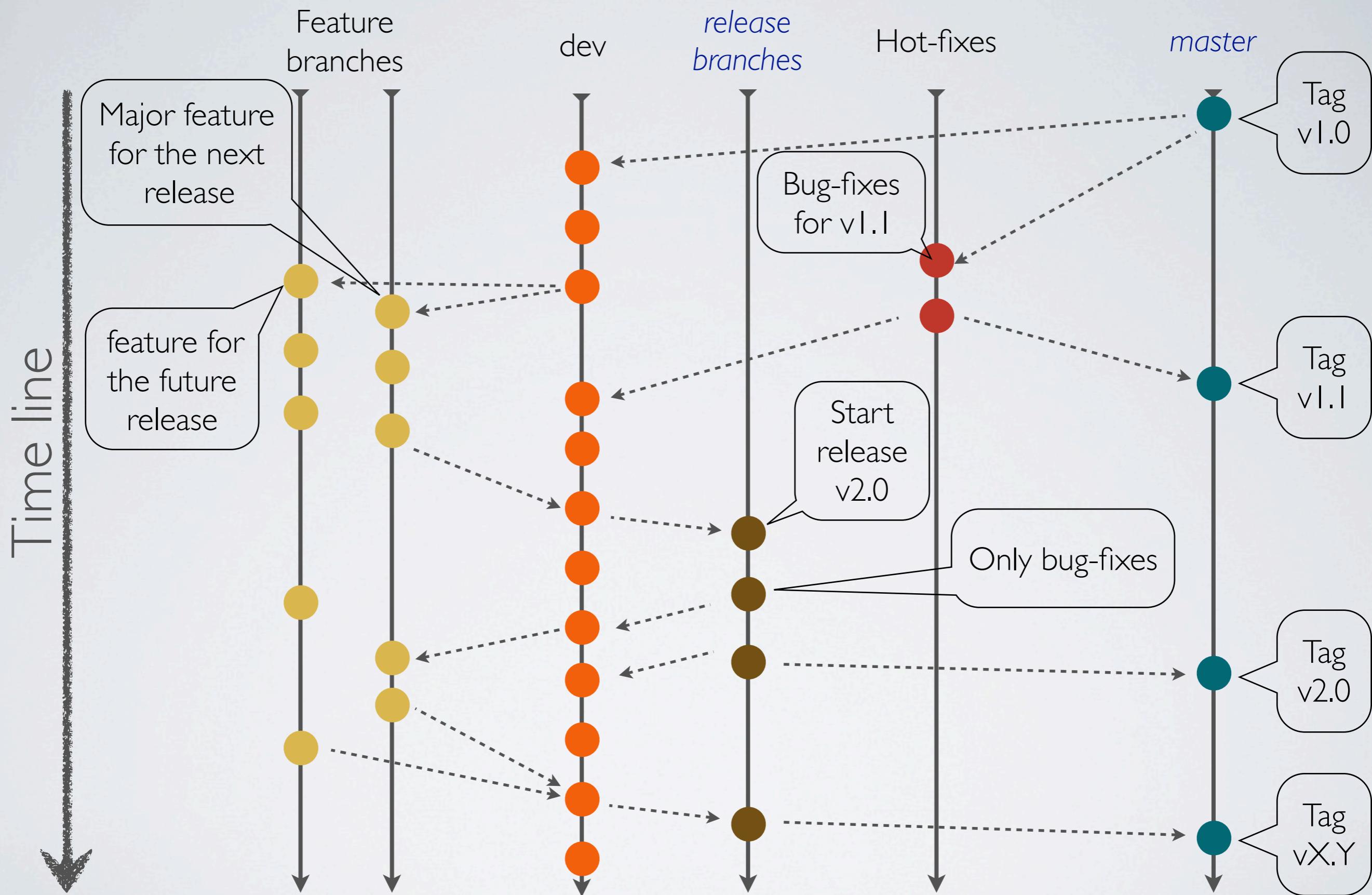








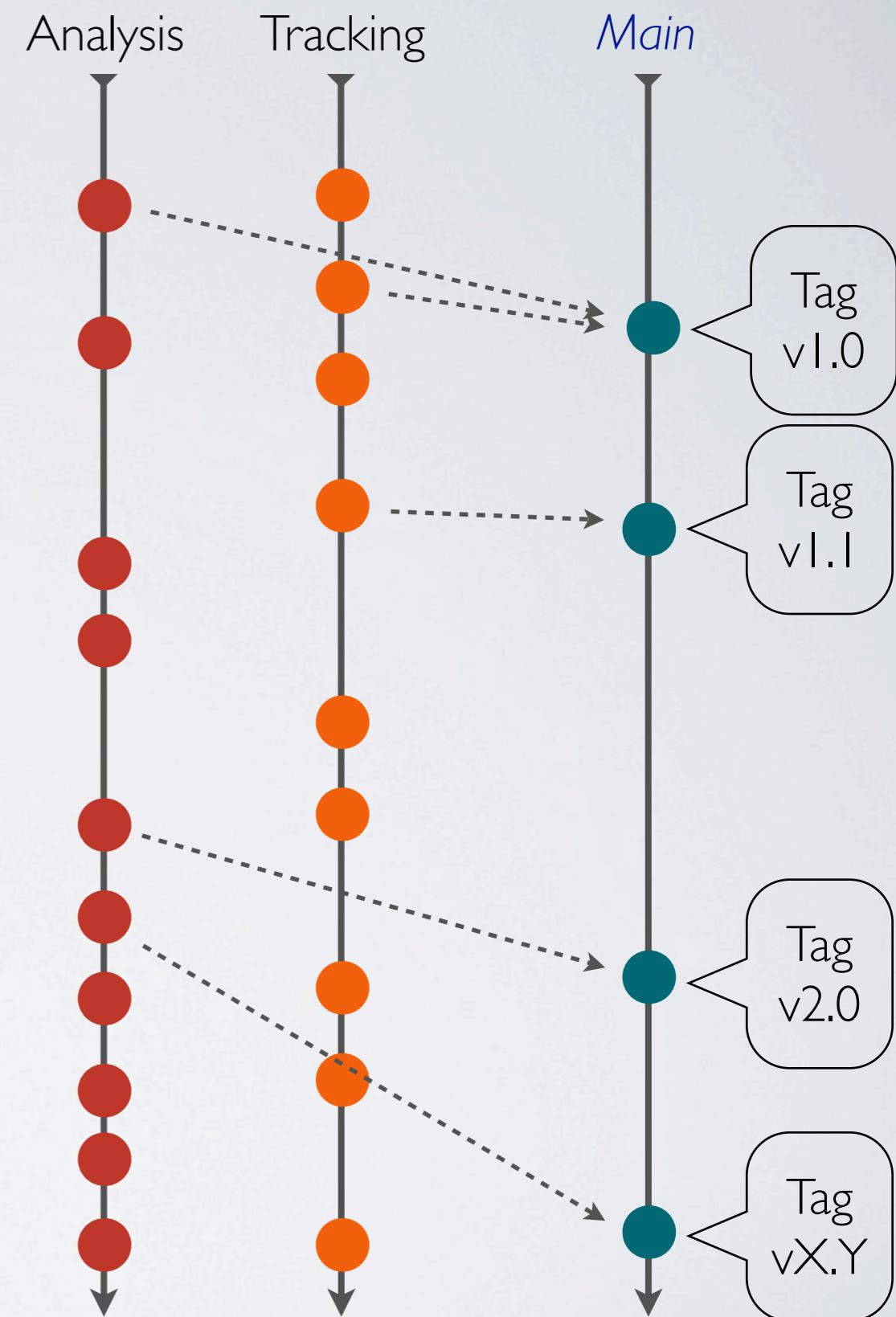
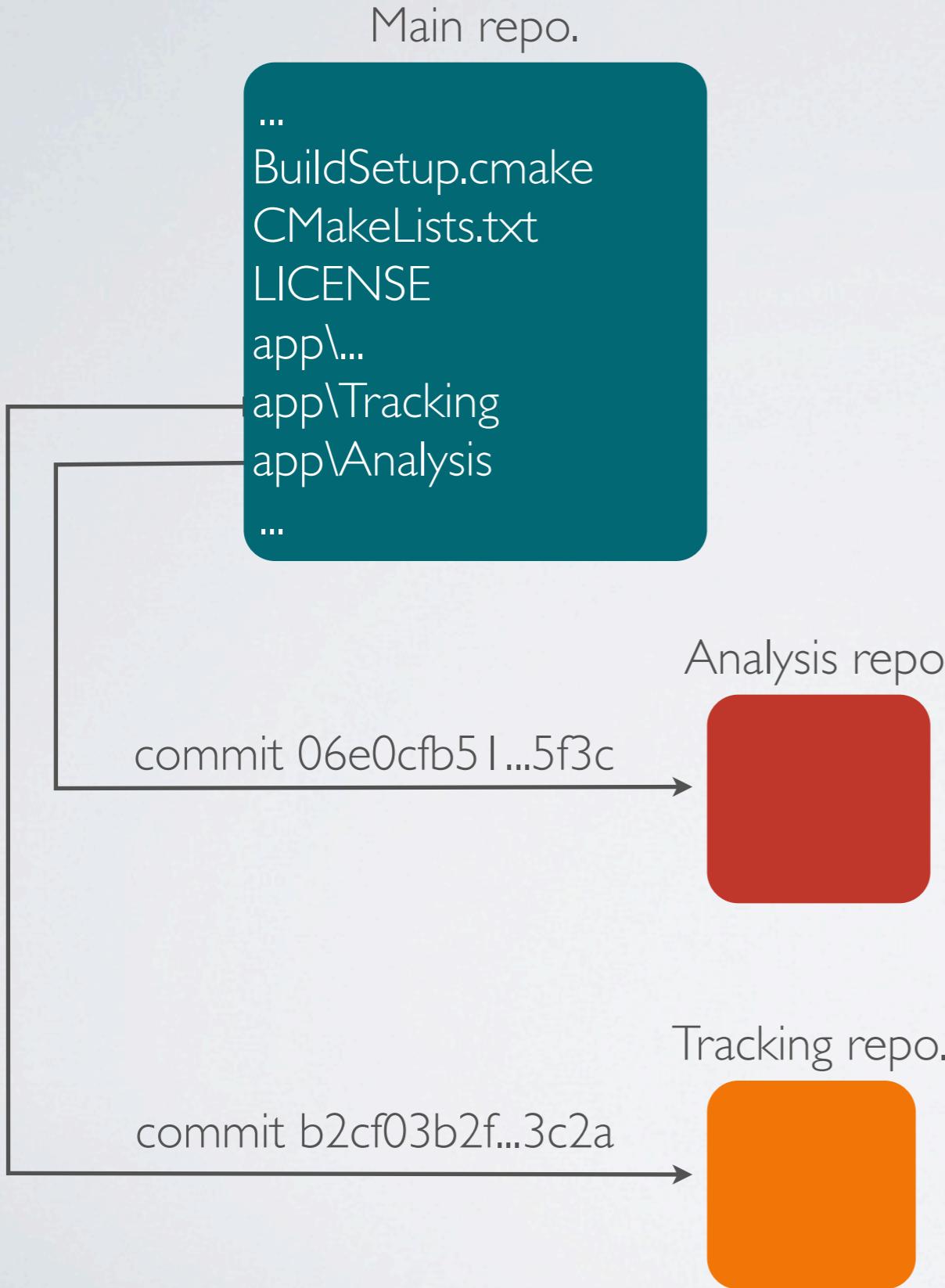




Git Submodules



Git Submodules



more Git features

more Git features

- Cut off a file or files from rep. (inc. history).

more Git features

- Cut off a file or files from rep. (inc. history).
- Merge / Divide repos while preserving history.

more Git features

- Cut off a file or files from rep. (inc. history).
- Merge / Divide repos while preserving history.
- Cherry Pick.

more Git features

- Cut off a file or files from rep. (inc. history).
- Merge / Divide repos while preserving history.
- Cherry Pick.
- Playing with commits (changing, delete, reordering, squashing, splitting).

more Git features

- Cut off a file or files from rep. (inc. history).
- Merge / Divide repos while preserving history.
- Cherry Pick.
- Playing with commits (changing, delete, reordering, squashing, splitting).
- You have always backups for free.

more Git features

- Cut off a file or files from rep. (inc. history).
- Merge / Divide repos while preserving history.
- Cherry Pick.
- Playing with commits (changing, delete, reordering, squashing, splitting).
- You have always backups for free.
- ...

Git commands

Git commands

Got 15 minutes and want to learn Git?

Git commands

Got 15 minutes and want to learn Git?



Learn Git in your browser for free with
<http://try.github.com/>