Downstream development

hidenorly

Recap : Upstream and Downstream

Upstream:
e.g.
kernel.org
android.com

Downstream:
e.g.
SoC vendor maintained kernel
SoC vendor maintained Android

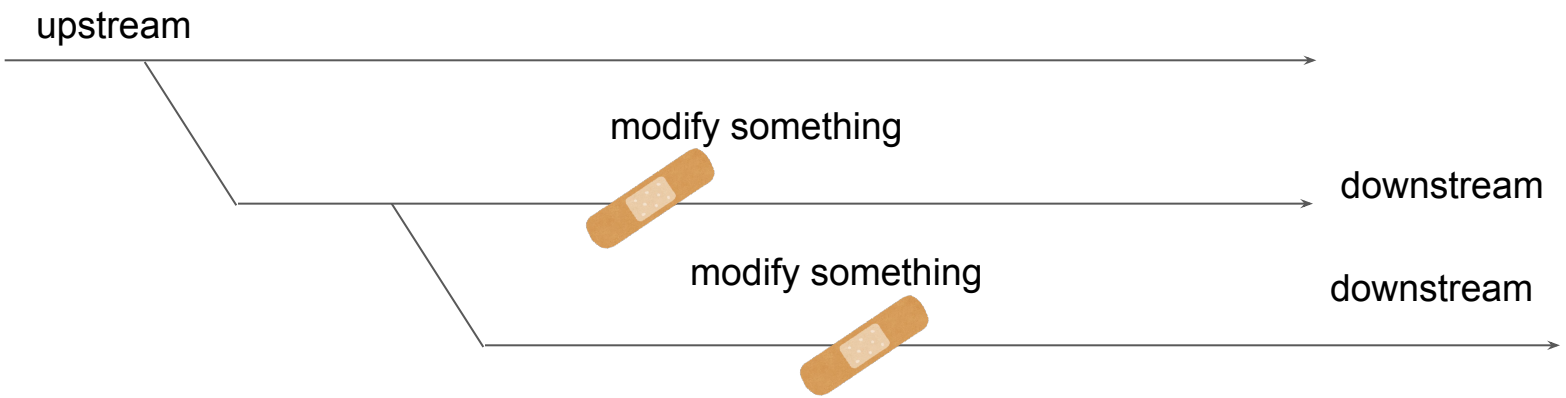
What’s happened on downstream?

Upstream:
e.g.
kernel.org
android.com

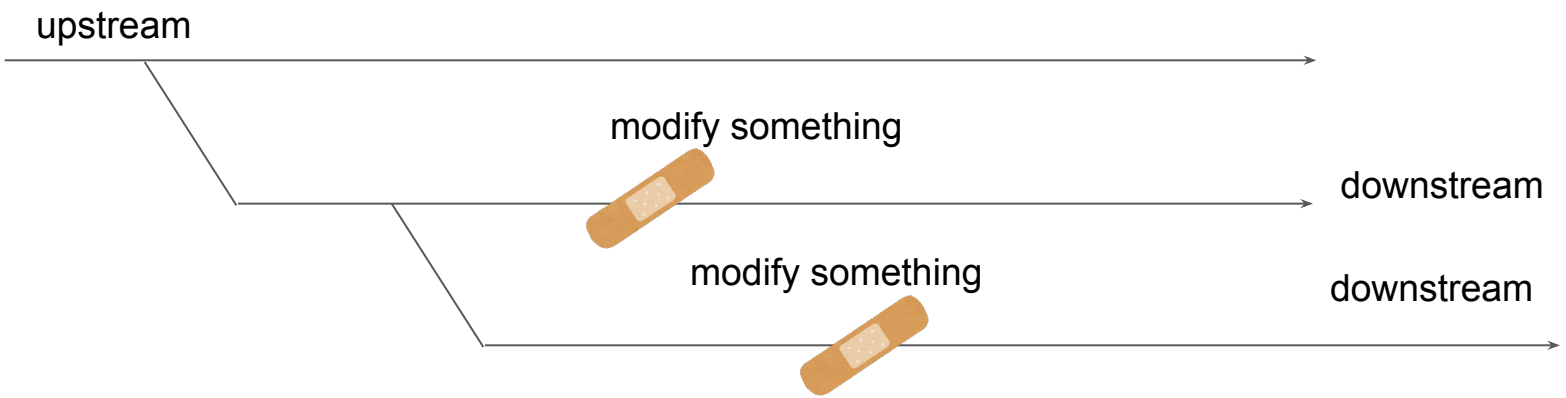
Downstream:
e.g.
SoC vendor maintained kernel
SoC vendor maintained Android

Downstream:
e.g.
manufacturer maintained kernel
manufacturer maintained Android

Recap : Upstream and Downstream as SCM

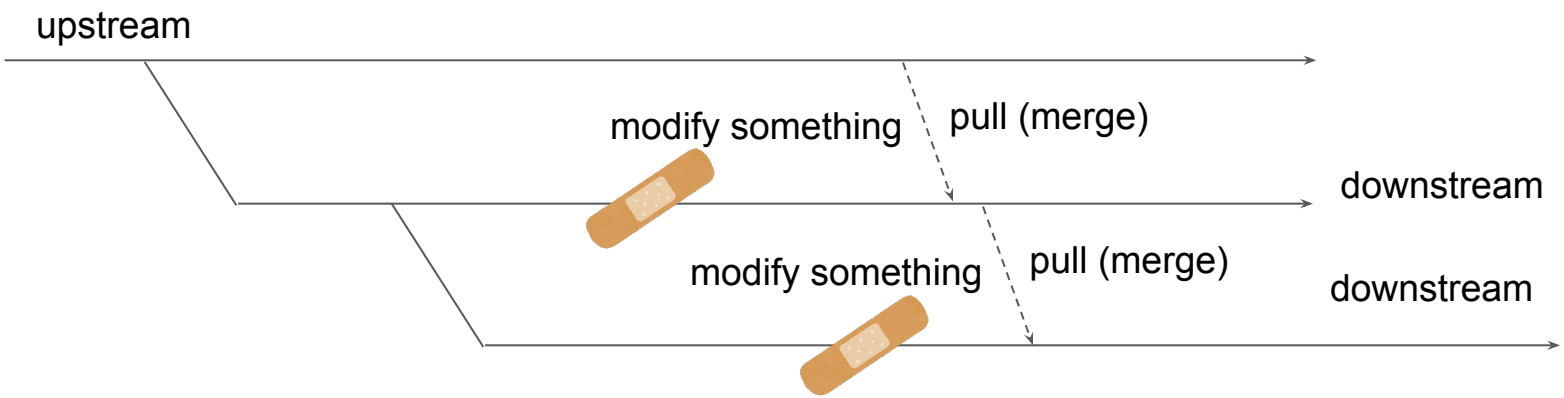


Recap : Upstream and Downstream as SCM



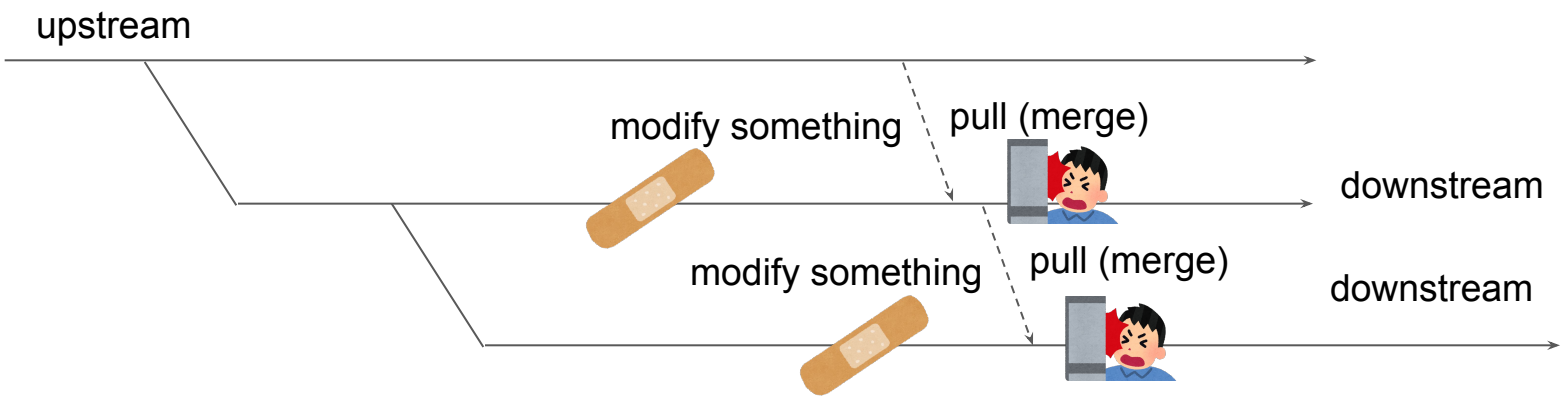
If this is oneshot activity, it’s no problem!
but…

Recap : Upstream and Downstream as SCM



Usual activity: integrate the upstream changes into the downstream

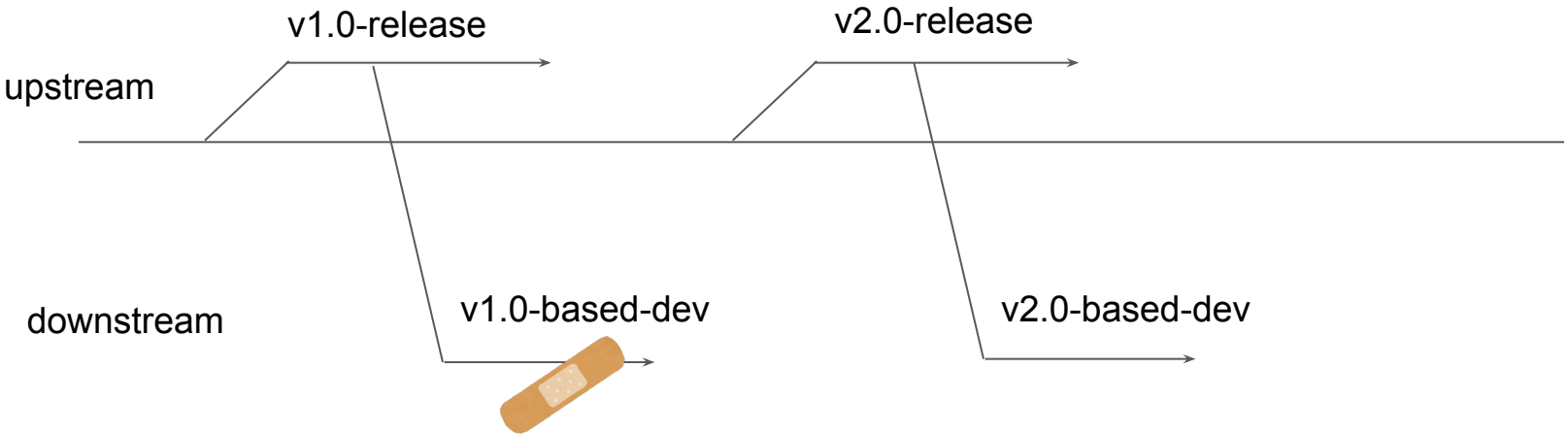
Recap : Upstream and Downstream as SCM



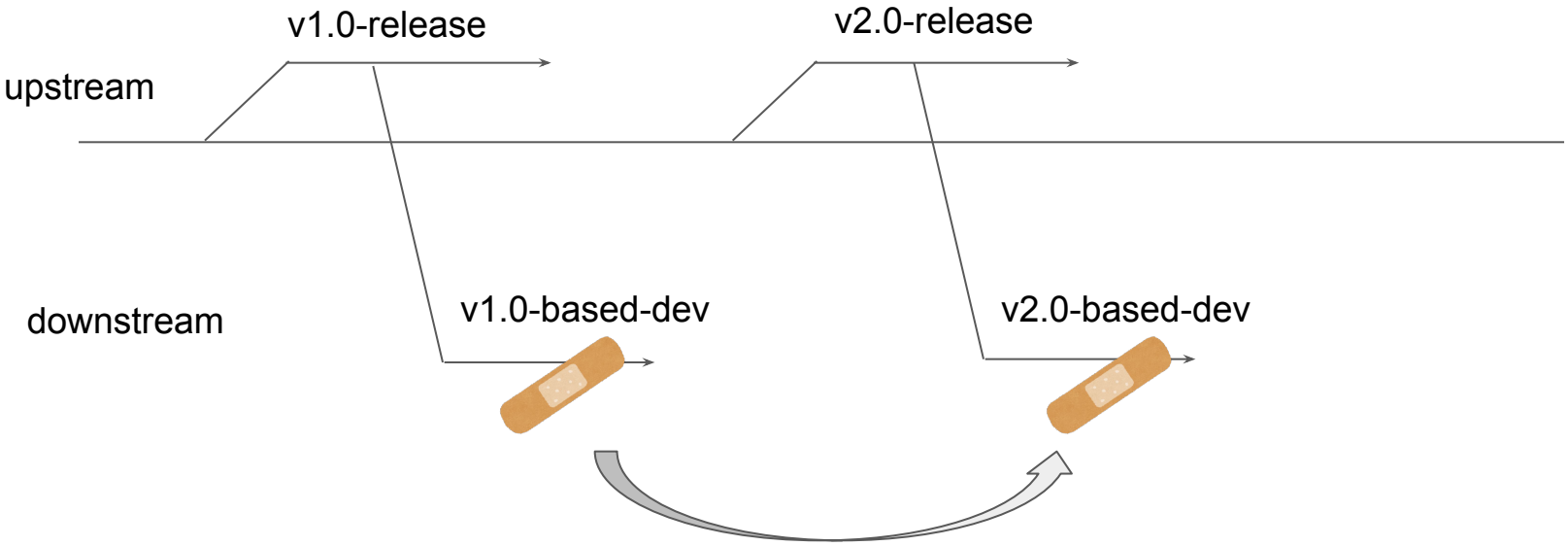
Usual activity: integrate the upstream changes into the downstream
→ may encounter merge conflict!!!

Background : Usual situation on Upstream and Downstream as SCM

Background : Usual situation on Upstream and Downstream as SCM



Background : Usual situation on Upstream and Downstream as SCM



patch bring up

e.g. Android-S to Android-T

What’s solution for the situation?

(1) mainlining (contribute the changes to the upstream)

What’s solution for the situation?

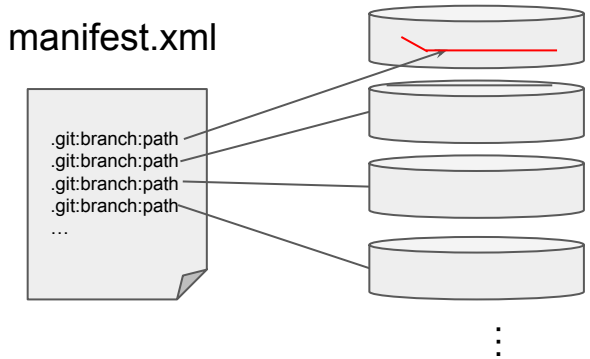
(1) mainlining (contribute the changes to the upstream)
(2) using tool <- today’s main topic

Recap: repo : Multiple git(s) utility

repo can handle multiple gits which are described in the manifest.xml file.

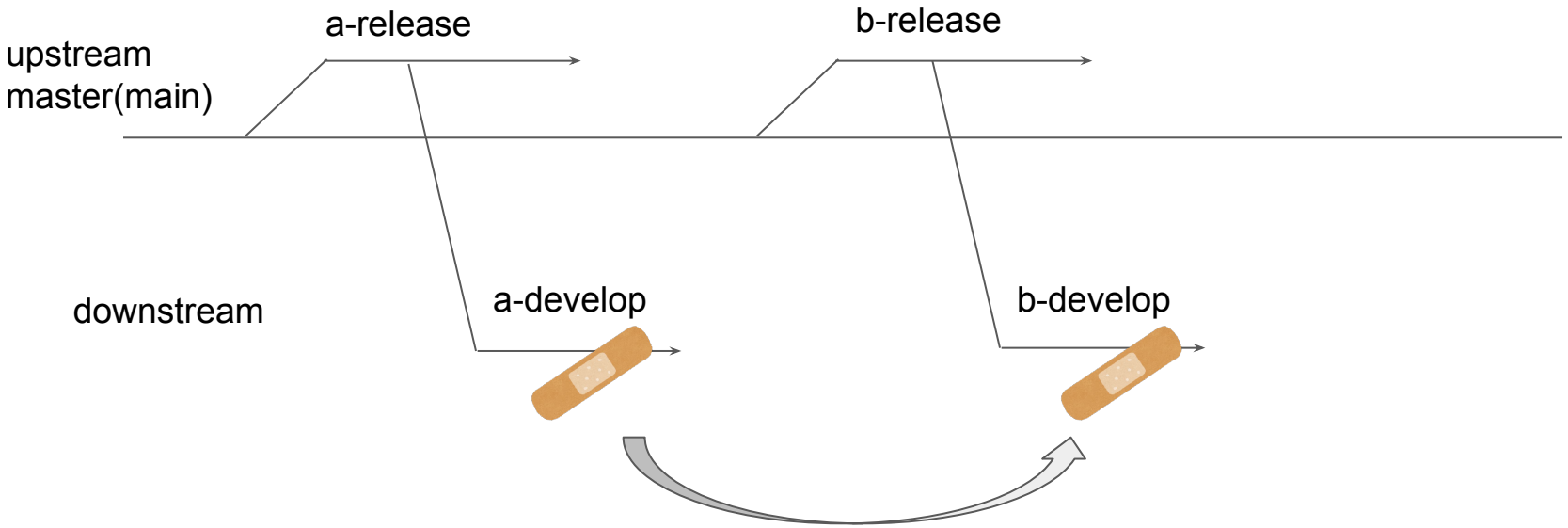
can handle Linux kernel.git, Android FW git(s), HAL git(s) and the Application git(s), etc.

Usually repo is used on Android project.



repo-gap-enumerator

https://github.com/hidenorly/RubyGitUtil



This tool can enumerate commits which are developed on the source repo's git (e.g. a-develop) and not contained in the destination repo's git (e.g. b-develop)

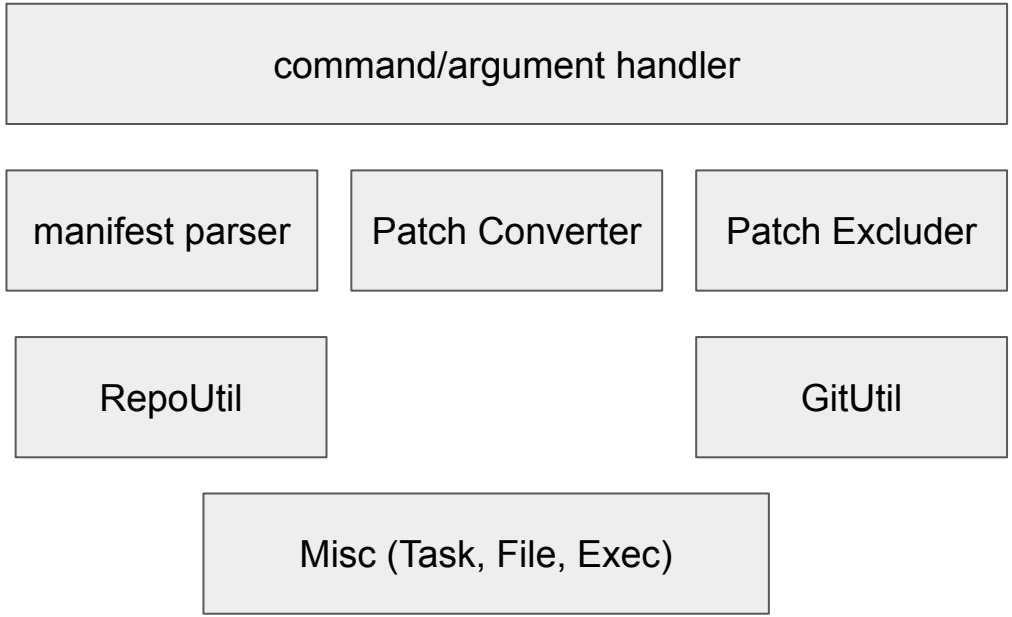
Remarkable points of the tool

- Enumerated patches are applicable with git am or git apply
- manifest git filter (e.g. can exclude prebuilts) to be faster execution
- Optional Git option supported (e.g. –author=)
- Robust missing commit check
  - Same commit id : supported
  - Different commit id : supported
    - Same change-id
    - Commit title
    - Time range search with actual commit body (actual diff level check)
  - mainlined commits are excluded by this tool
- Then actual missing commits are enumerated.

Expected users of the tool

- repo user
- Android developer for OS version up
- Developer of the successor project

mechanism of the tool



[tool]

1. Extract target git(s) from manifest & git filter option
2. Create patch with git option
   a. git format-patch -1 sha1
3. Exclude contained patches in the target git(s)
   a. remove the .patch if the .patch is included in the git

[result]

- necessary patches to bring up on target repo gits
  - git am can apply the patch. (internally using applier)

robust exclusion

- sha1 is included or not
- –grep with the Change-id in the patch and check the enumerated commit is same or not
- commit title search
- time range search

- robust-check whether commit is same or not
  - ignore commit message and comment lines //, #
  - actual +/- lines are same or not (for stripped string)

Call for action

- Use this tool and feedback to me if you have similar situation!!!
  - If you have situation to bring up patches to new project
  - If you're interested in the gap of repo/branch(s)
  - If you'd like to extract contribution candidates although git doesn't have enough git history to upstream.