How Should We Tackle the Automobile with Modern Massive Complexity ?


GRIFFIN TECHNOLOGY

Case 1 : 10 years old “Jog Dial”

PowerMate USB, Classic Aluminum

Case 1 : 10 years old “Jog Dial”

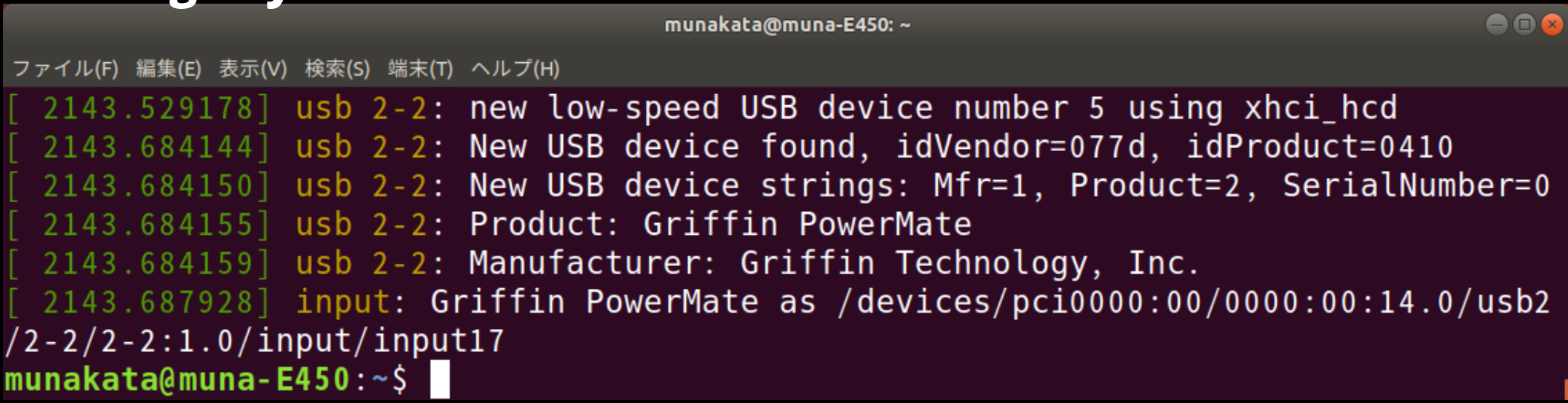
System Requirements:

- **Mac: Mac OS X 10.3.9 or greater, and a USB port**
- **PC: Windows XP, or Vista, USB port**

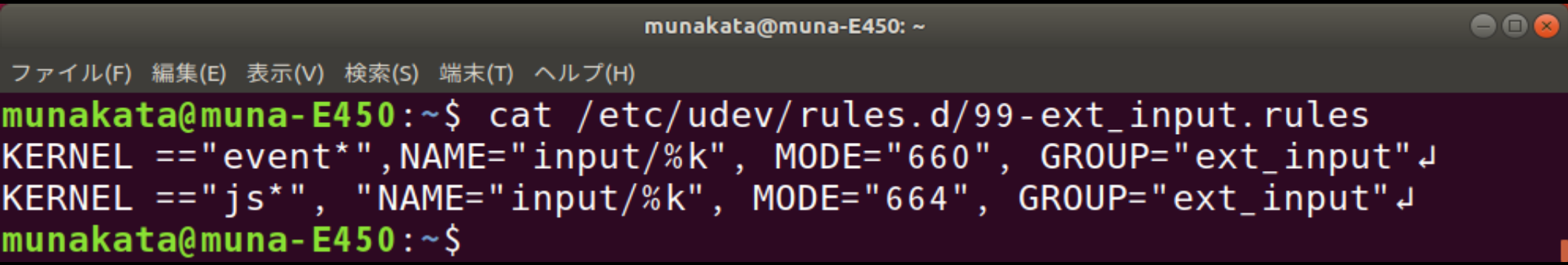
PowerMate USB, Classic Aluminum

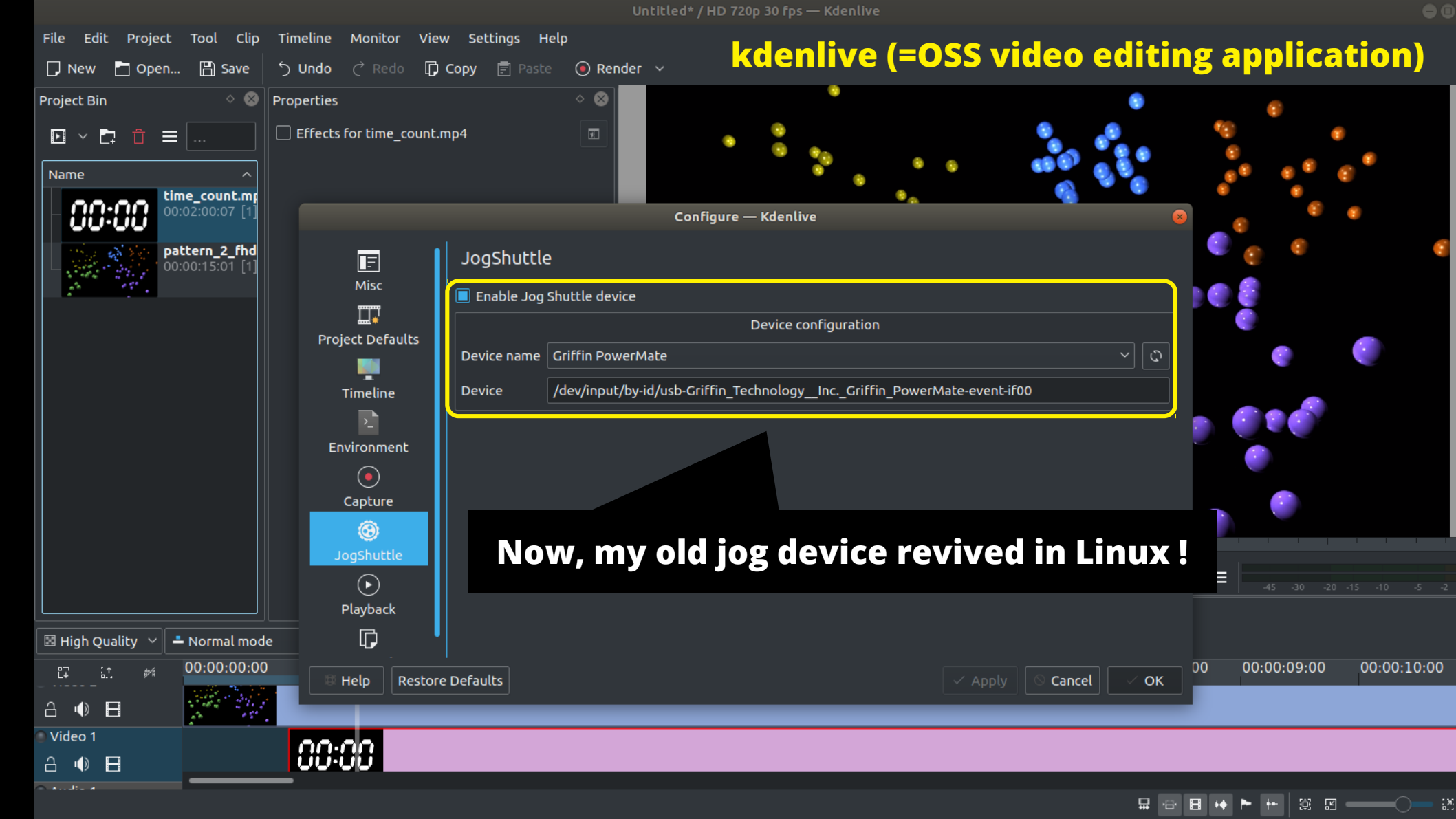
Linux kernel automatically detected new USB HID class device

dmesg says



No new driver coding needed,
just added a new udev rule to enable new input method




Untitled* / HD 720p 30 fps — Kdenlive
File Edit Project Tool Clip Timeline Monitor View Settings Help
New Open... Save Undo Redo Copy Paste Render
kdenlive (=OSS video editing application)
Project Bin
Properties
Effects for time_count.mp4
Name
time_count.mp
00:02:00:07 [1]
pattern_2_fhd
00:00:15:01 [1]
Configure — Kdenlive
Misc
Project Defaults
Timeline
Environment
Capture
JogShuttle
Playback
JogShuttle
Enable Jog Shuttle device
Device configuration
Device name Griffin PowerMate
Device /dev/input/by-id/usb-Griffin_Technology__Inc._Griffin_PowerMate-event-if00
Now, my old jog device revived in Linux !
Help Restore Defaults
Apply Cancel OK
High Quality Normal mode
00:00:00:00
00:00:09:00
00:00:10:00
Video 1

Why Jog device could successfully revive ?

- HW comply with USB standard spec
- Linux USB driver could detect it as a new HID device
- Linux udev mechanism worked to add new input method
- Application SW accept standard Linux input device

Case 2: 5 years old Linux device driver code

```
{
        u16 flags;

        if (!vring_use_dma_api(vq->vq.vdev))
                return;

        flags = virtio16_to_cpu(vq->vq.vdev, desc->flags);

        if (flags & VRING_DESC_F_INDIRECT) {
                dma_unmap_single(vring_dma_dev(vq),
                                 virtio64_to_cpu(vq->vq.vdev, desc->addr),
                                 virtio32_to_cpu(vq->vq.vdev, desc->len),
                                 (flags & VRING_DESC_F_WRITE) ?
                                 DMA_FROM_DEVICE : DMA_TO_DEVICE);
        } else {
                dma_unmap_page(vring_dma_dev(vq),
                               virtio64_to_cpu(vq->vq.vdev, desc->addr),
                               virtio32_to_cpu(vq->vq.vdev, desc->len),
                               (flags & VRING_DESC_F_WRITE) ?
                               DMA_FROM_DEVICE : DMA_TO_DEVICE);
        }
}
```

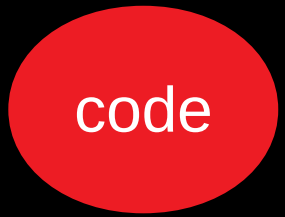
- runs on 3.10 kernel
- source code
- README
- built-in help only
- no separate document

- written by 3rd party company
- not mainlined

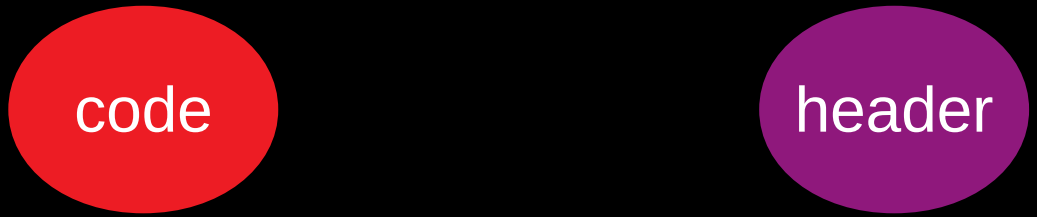
You may hit variety of troubles

- Can not compile, due to
  - tool chain (library, compiler) miss match
  - in-kernel driver API change
- Hard to apply security fix patch
- Hard to access
  - original author
  - code change history (intention, related fix)

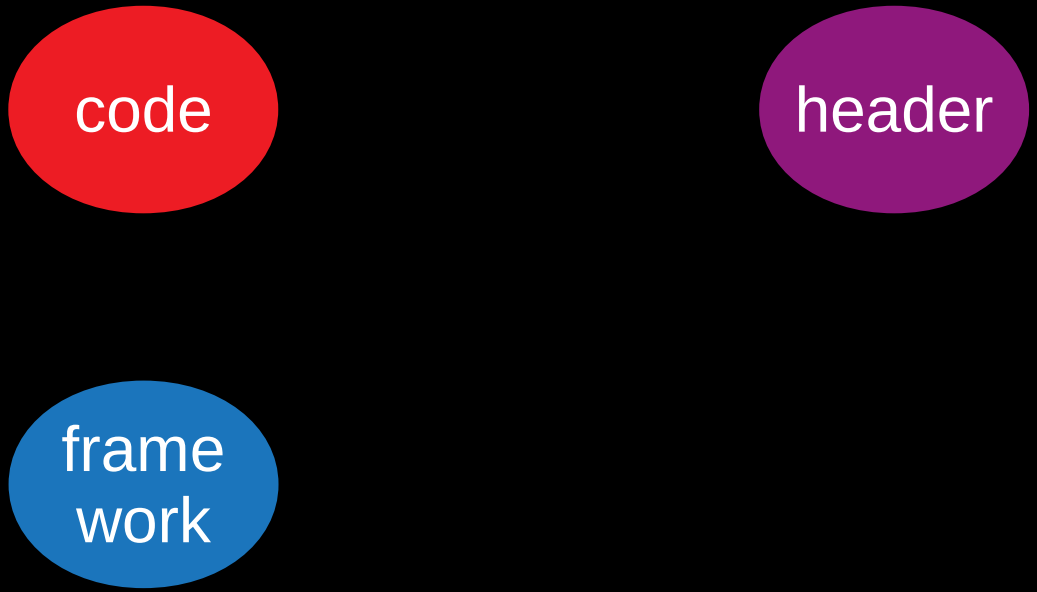
SW complexity seems bigger than HW, why?



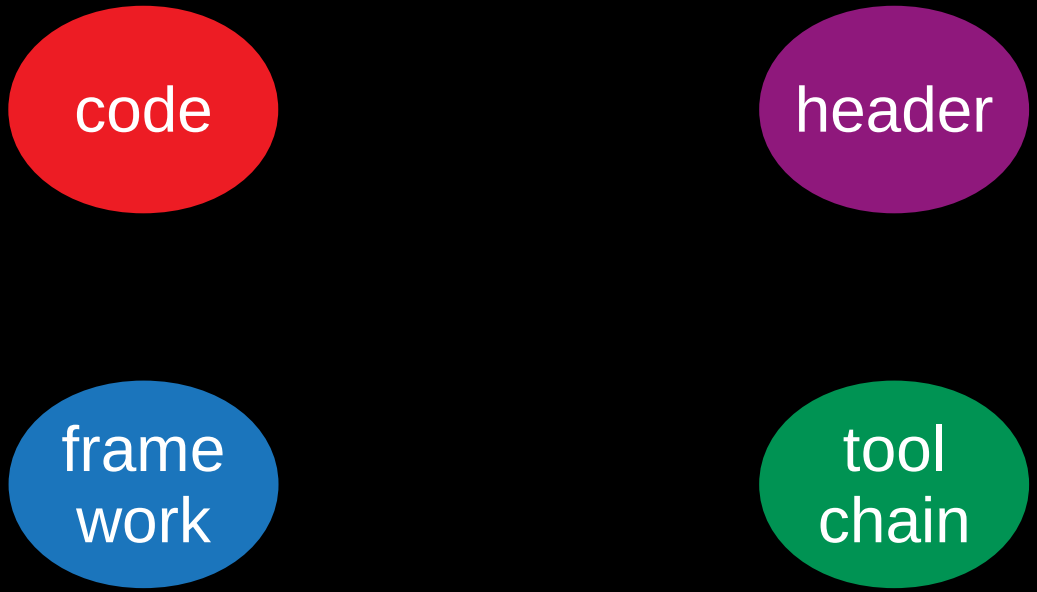
SW complexity seems bigger than HW, why?



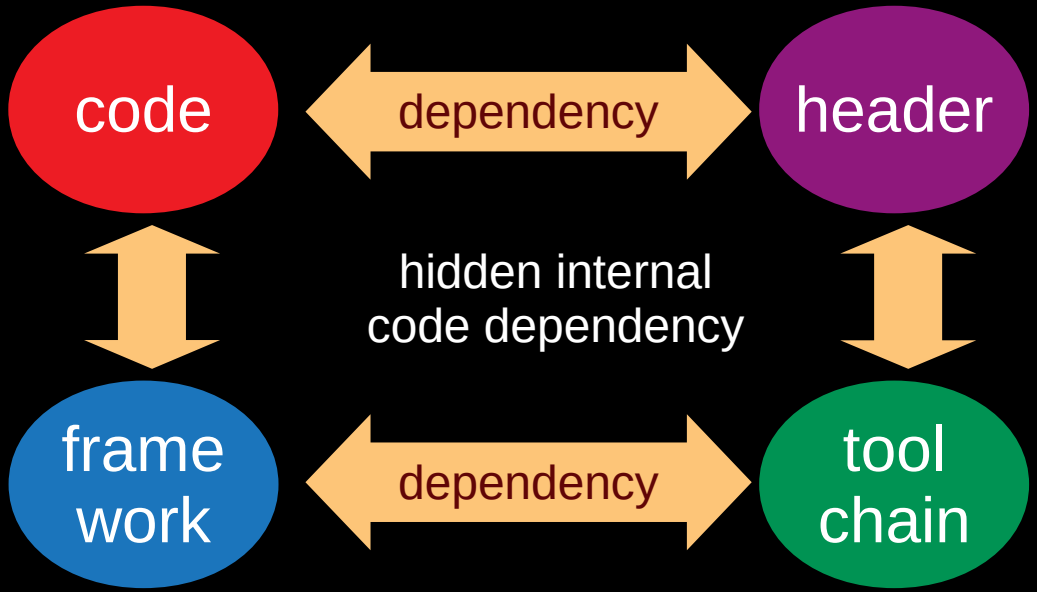
SW complexity seems bigger than HW, why?



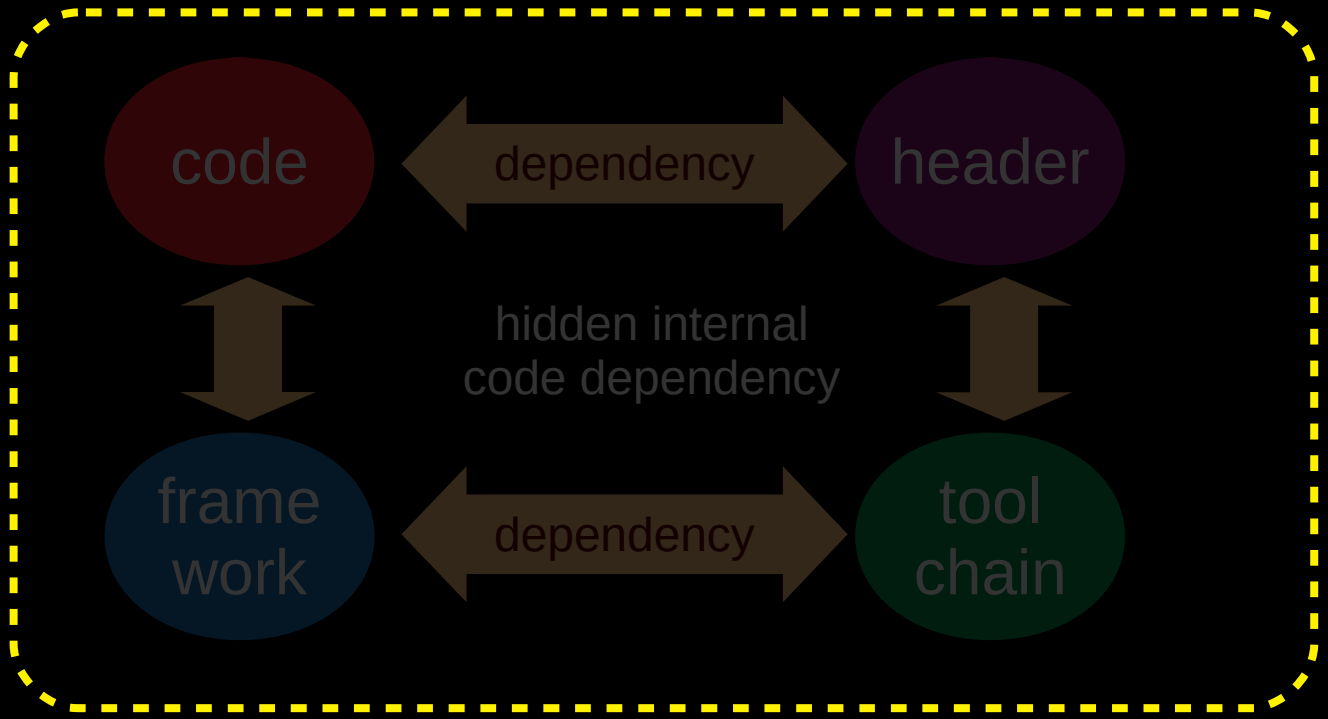
SW complexity seems bigger than HW, why?



SW complexity seems bigger than HW, why?

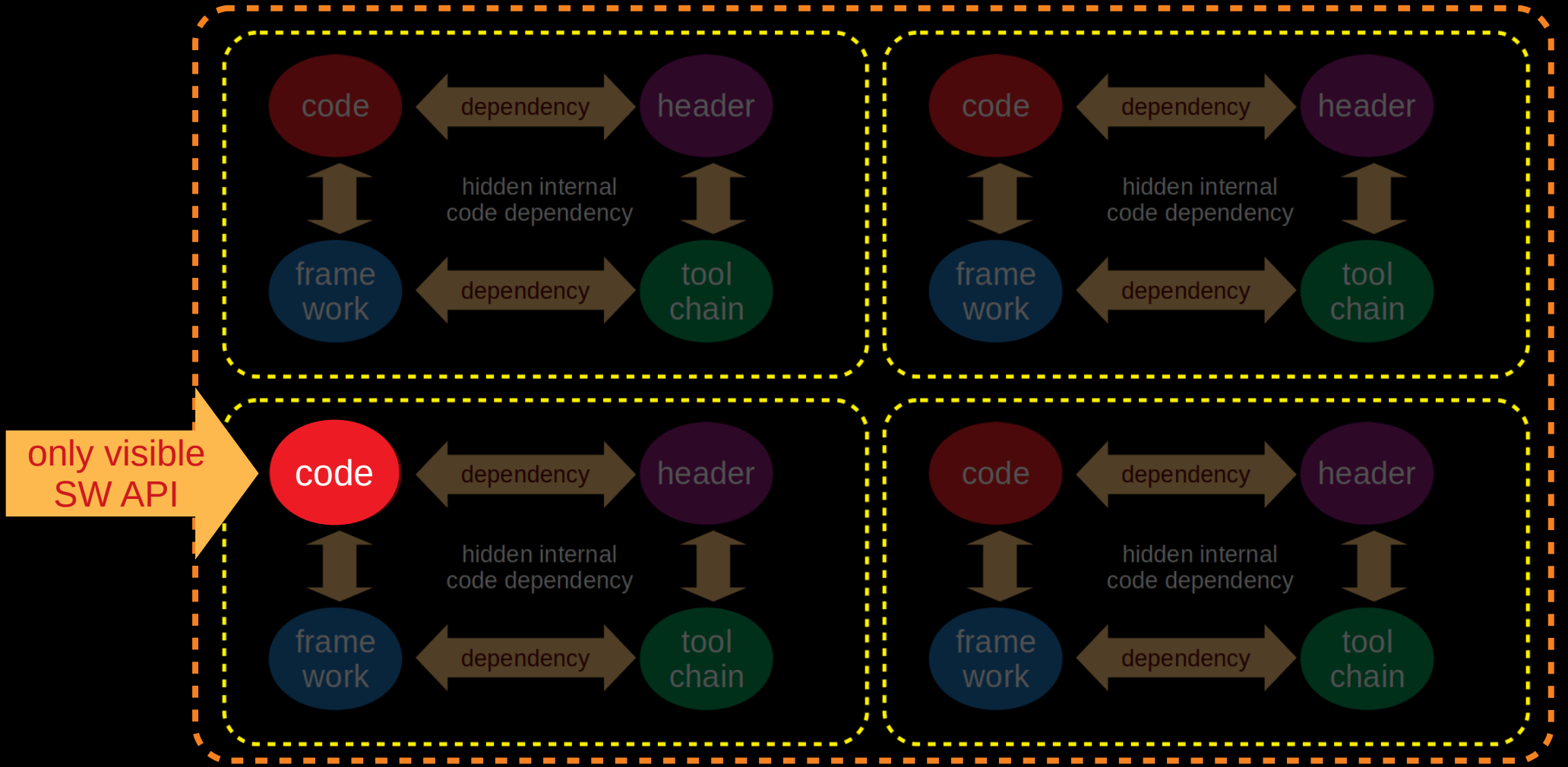


SW complexity seems bigger than HW, why?

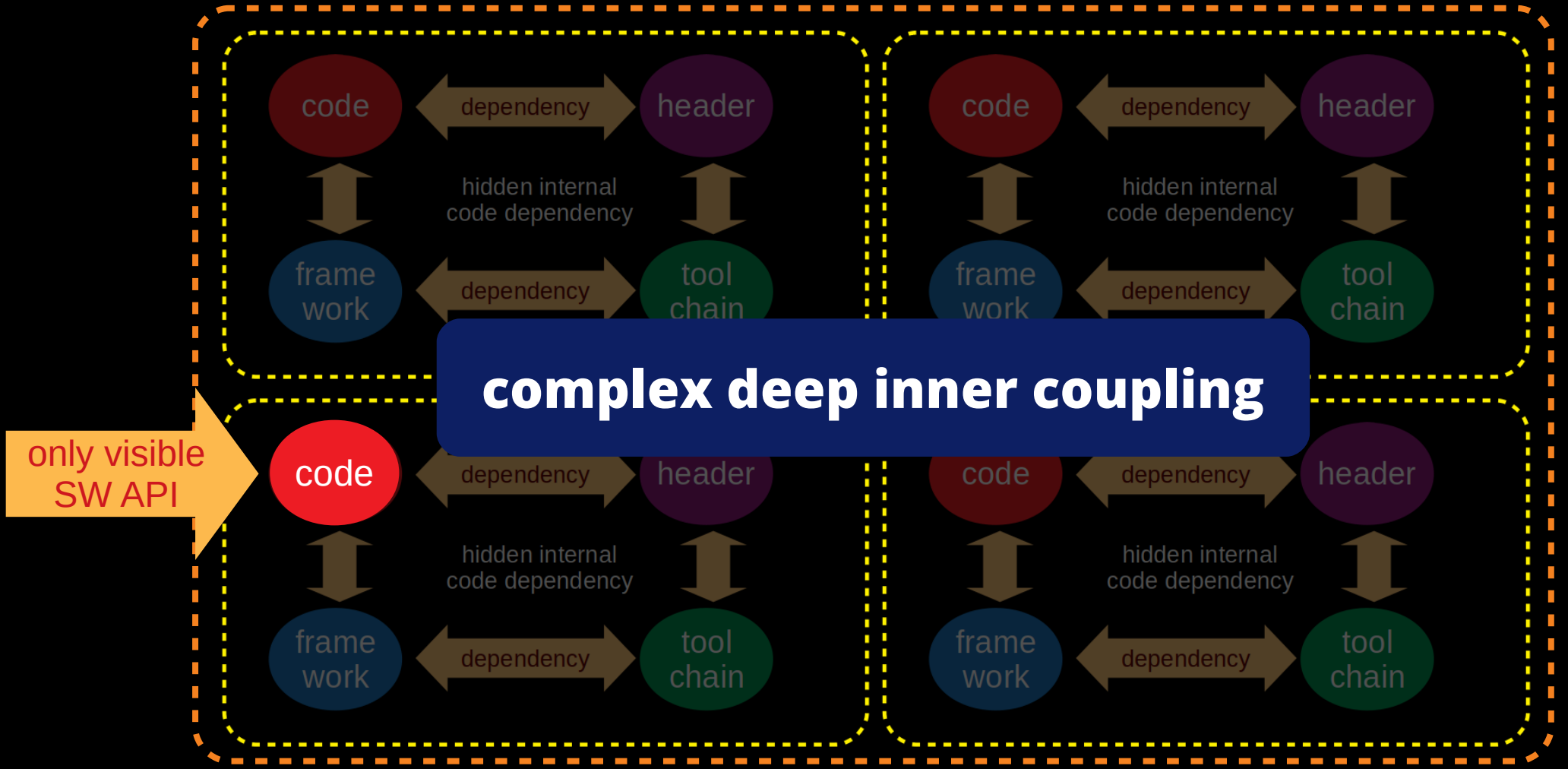


internal hidden coupling easily create the blockbox

Due to the low visibility, very hard to maintain SW

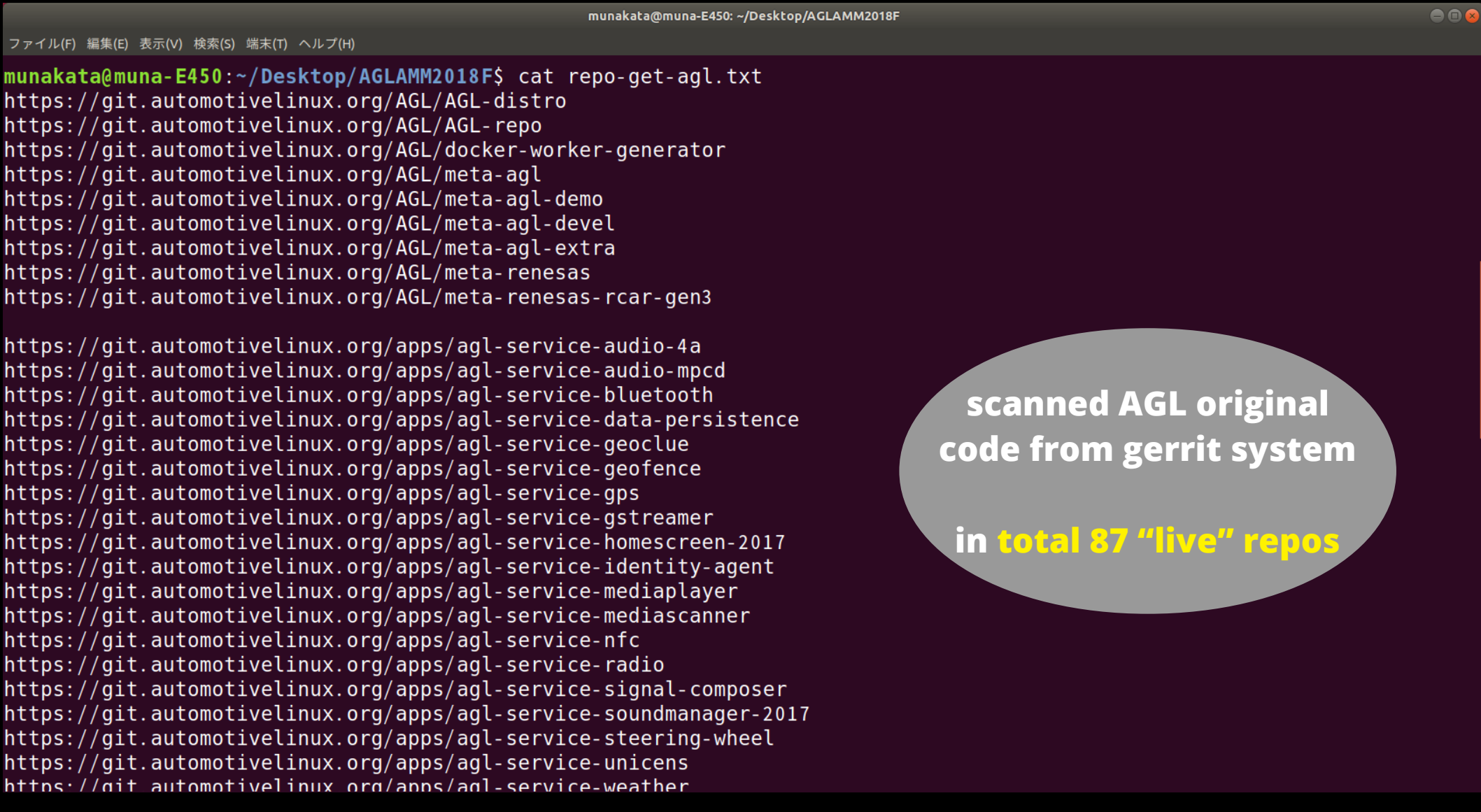


Due to the low visibility, very hard to maintain SW

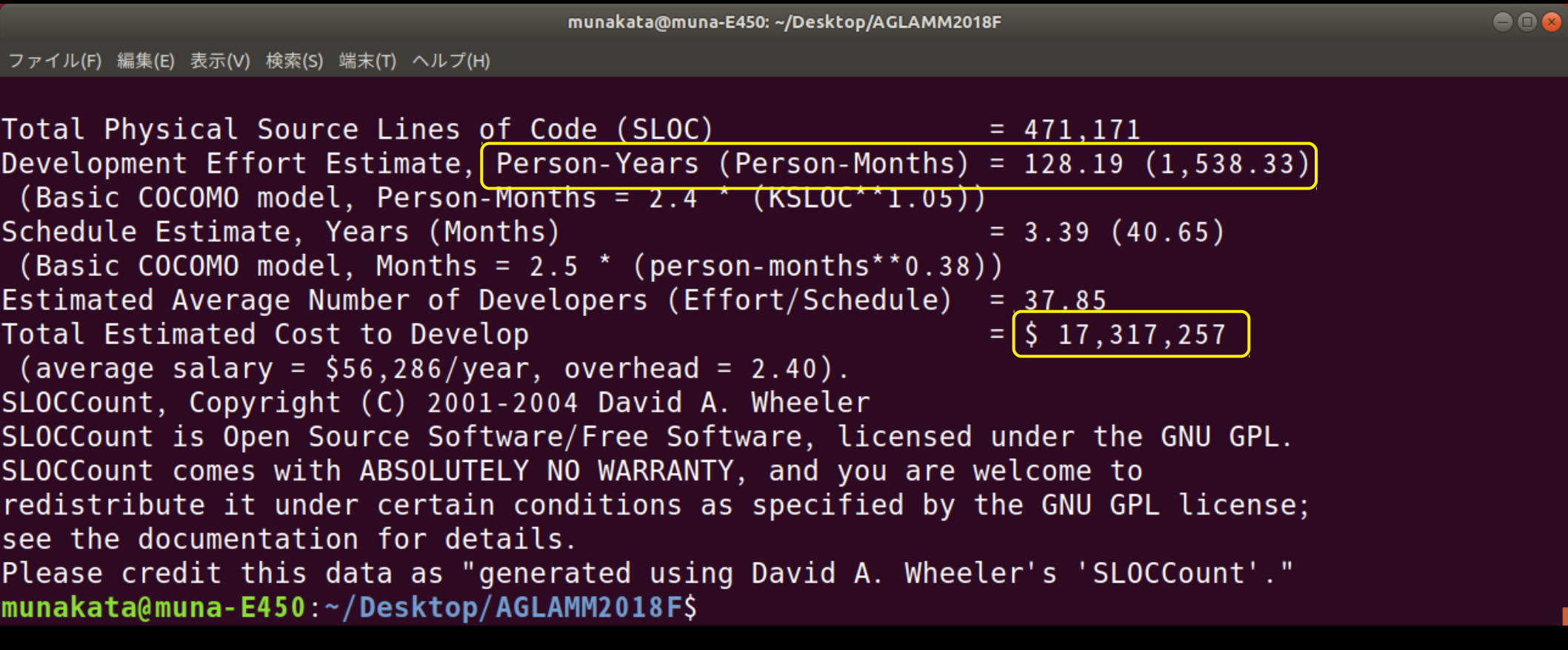


Let’s see the AGL code

Where is the code ?


munakata@muna-E450: ~/Desktop/AGLAMM2018F
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
munakata@muna-E450:~/Desktop/AGLAMM2018F$ cat repo-get-agl.txt
https://git.automotivelinux.org/AGL/AGL-distro
https://git.automotivelinux.org/AGL/AGL-repo
https://git.automotivelinux.org/AGL/docker-worker-generator
https://git.automotivelinux.org/AGL/meta-agl
https://git.automotivelinux.org/AGL/meta-agl-demo
https://git.automotivelinux.org/AGL/meta-agl-devel
https://git.automotivelinux.org/AGL/meta-agl-extra
https://git.automotivelinux.org/AGL/meta-renesas
https://git.automotivelinux.org/AGL/meta-renesas-rcar-gen3
https://git.automotivelinux.org/apps/agl-service-audio-4a
https://git.automotivelinux.org/apps/agl-service-audio-mpcd
https://git.automotivelinux.org/apps/agl-service-bluetooth
https://git.automotivelinux.org/apps/agl-service-data-persistence
https://git.automotivelinux.org/apps/agl-service-geoclue
https://git.automotivelinux.org/apps/agl-service-geofence
https://git.automotivelinux.org/apps/agl-service-gps
https://git.automotivelinux.org/apps/agl-service-gstreamer
https://git.automotivelinux.org/apps/agl-service-homescreen-2017
https://git.automotivelinux.org/apps/agl-service-identity-agent
https://git.automotivelinux.org/apps/agl-service-mediaplayer
https://git.automotivelinux.org/apps/agl-service-mediascanner
https://git.automotivelinux.org/apps/agl-service-nfc
https://git.automotivelinux.org/apps/agl-service-radio
https://git.automotivelinux.org/apps/agl-service-signal-composer
https://git.automotivelinux.org/apps/agl-service-soundmanager-2017
https://git.automotivelinux.org/apps/agl-service-steering-wheel
https://git.automotivelinux.org/apps/agl-service-unicens
scanned AGL original code from gerrit system
in total 87 “live” repos

Rough value projection of AGL developed code (by SLOCCount)



KPI of SW productivity

- **LOC (lines of code, steps) / man**
- **LOC / month**
- **MM (man month) / project**

Assumption

Every development (re)starts from the scratch

KPI of SW productivity is changing

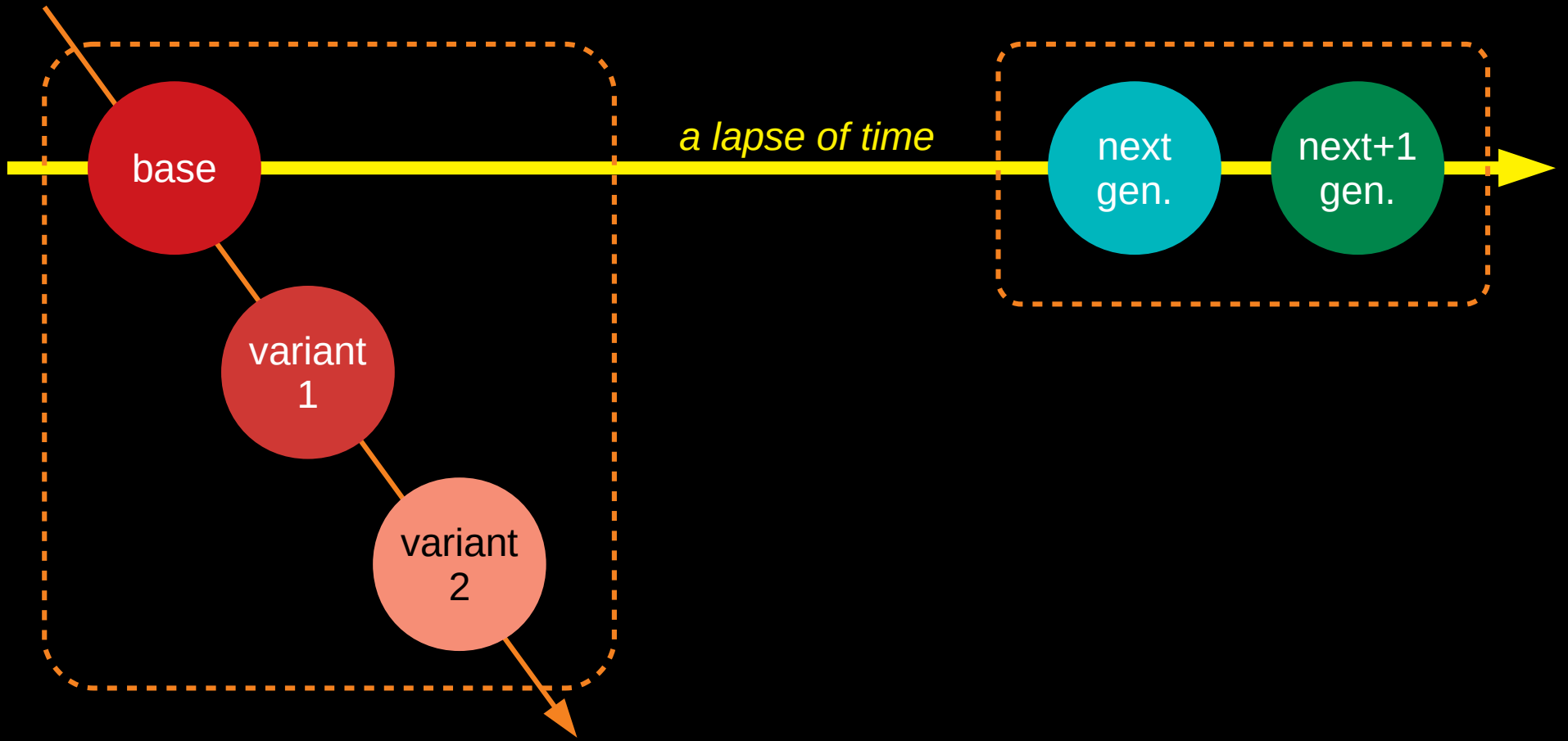
- **LOC (lines of code, steps) / man**
- **LOC / month**
- **MM (man month) / project**

Assumption

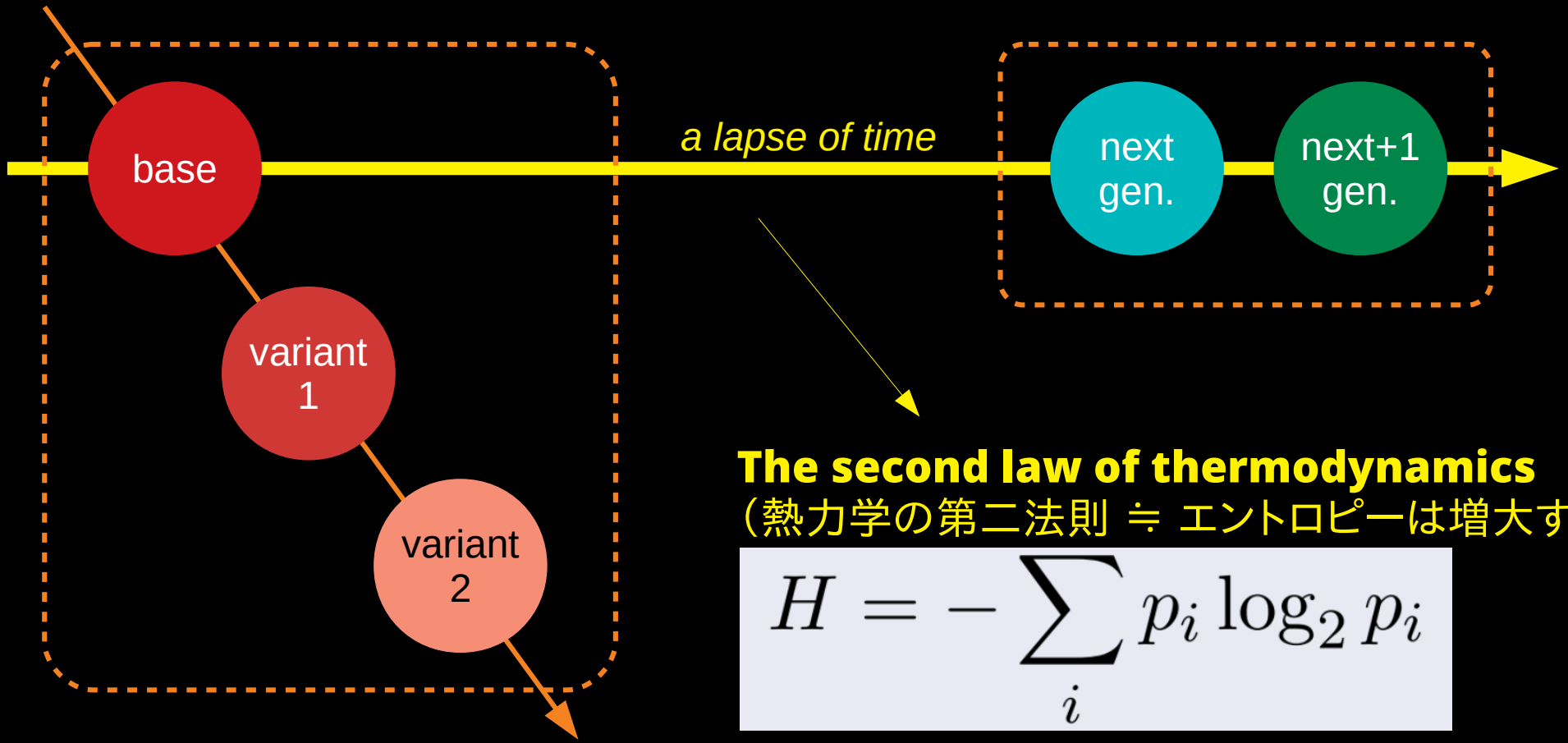
Every development (re)starts from the scratch

- **Code reuse ratio**
- **Code migration (update) cycle**
- **Bug fix application period)**

SW re-usability (vertical and horizontal)



SW re-usability (vertical and horizontal)



The second law of thermodynamics
（熱力学の第二法則 ≒ エントロピーは増大する）

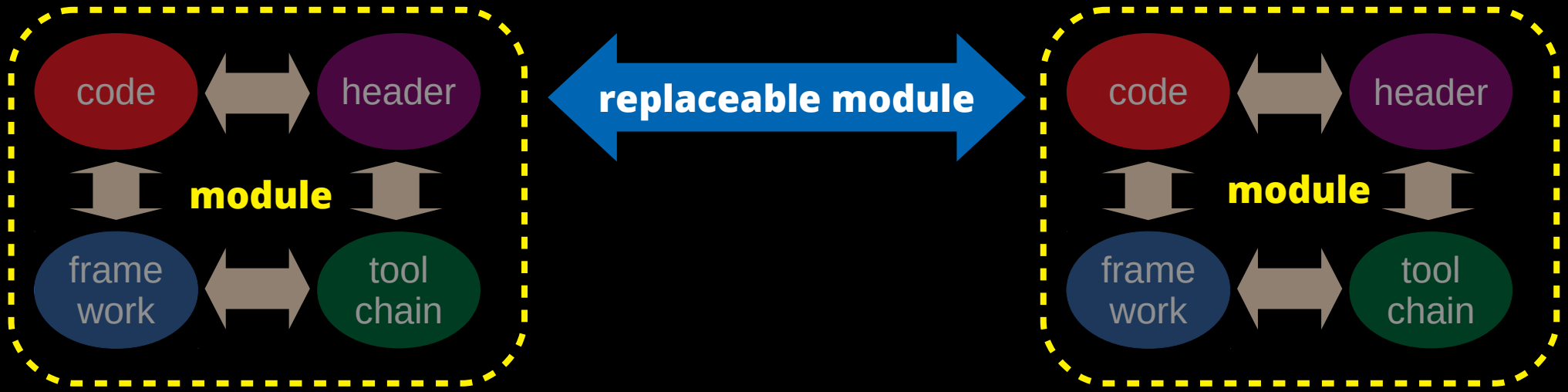
$$H = -\sum_i p_i \log_2 p_i$$

quote from http://web.tuat.ac.jp/~s-hotta/info/slide11.pdf

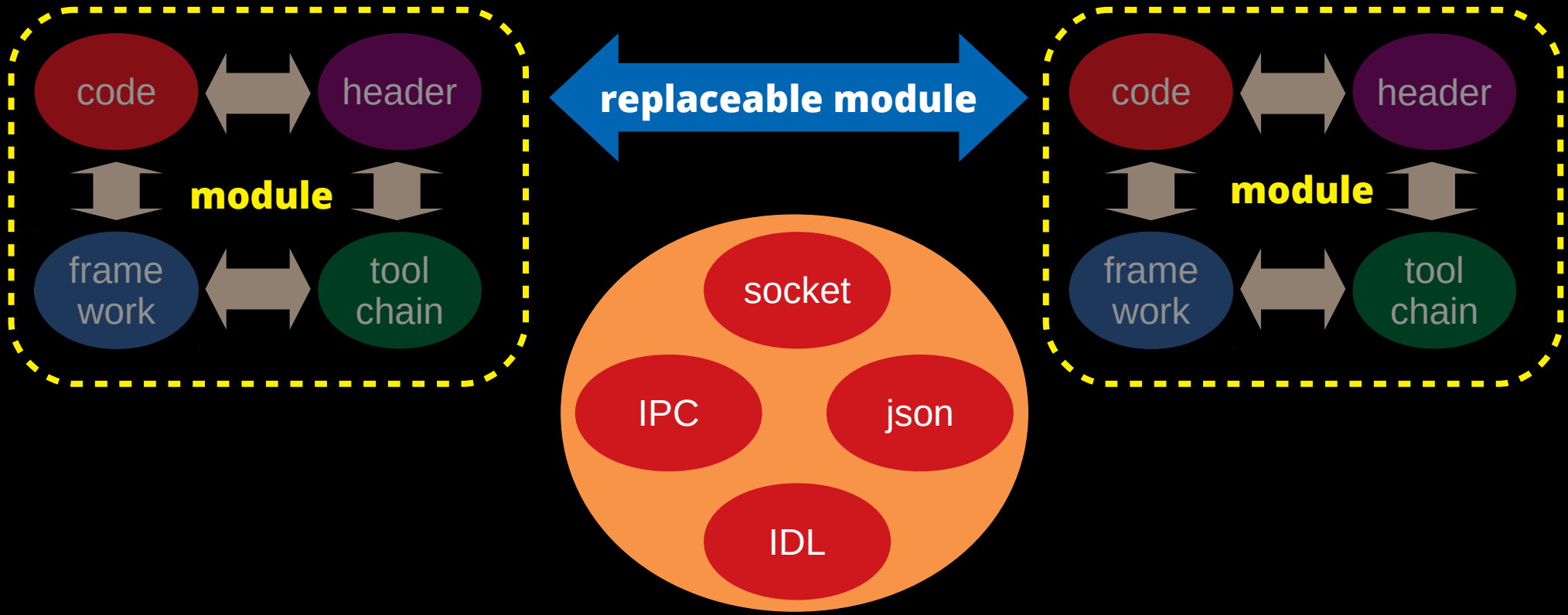
How should we manage SW?

- **Increase of hidden inner coupling**
  (eliminate state of black boxing)
- **Increase of entropy**
  (loss of information)

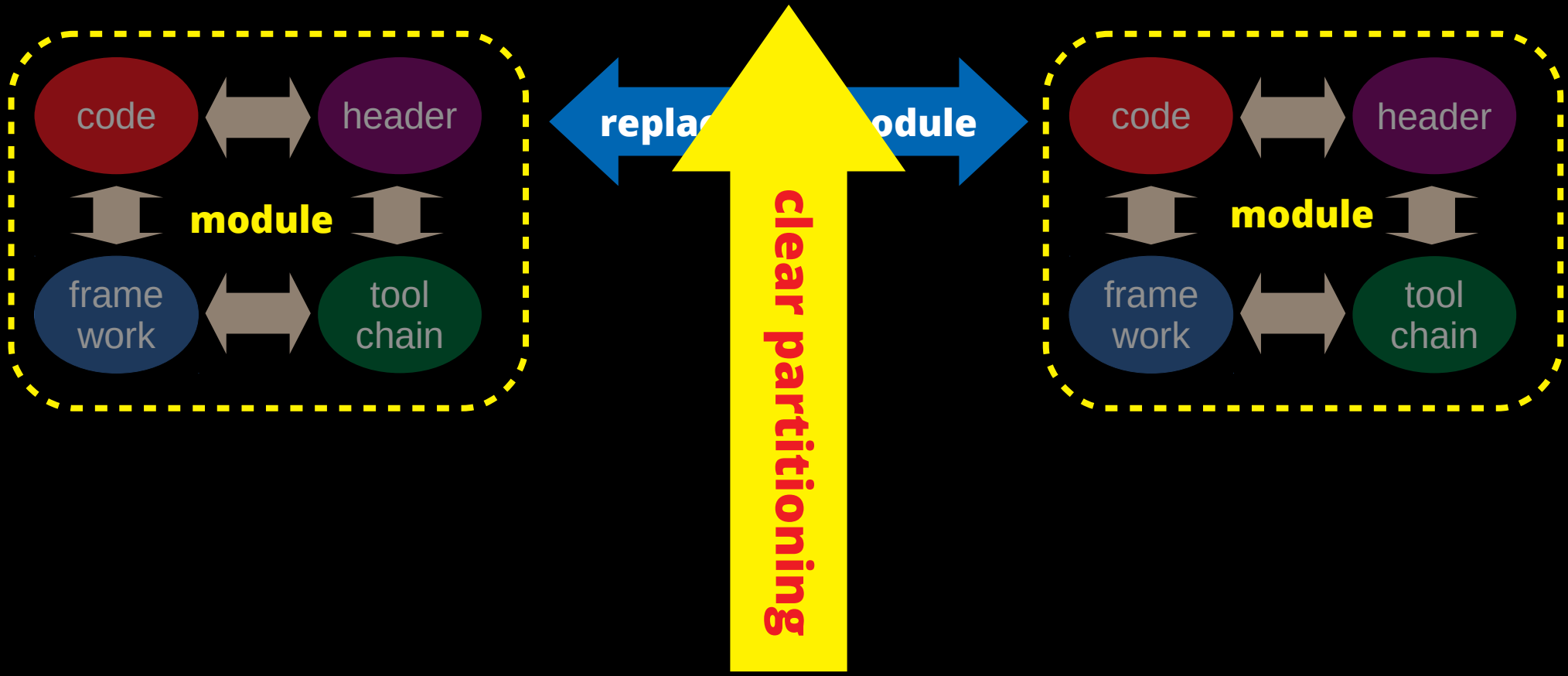
1. decoupling : unbind hidden internal dependency

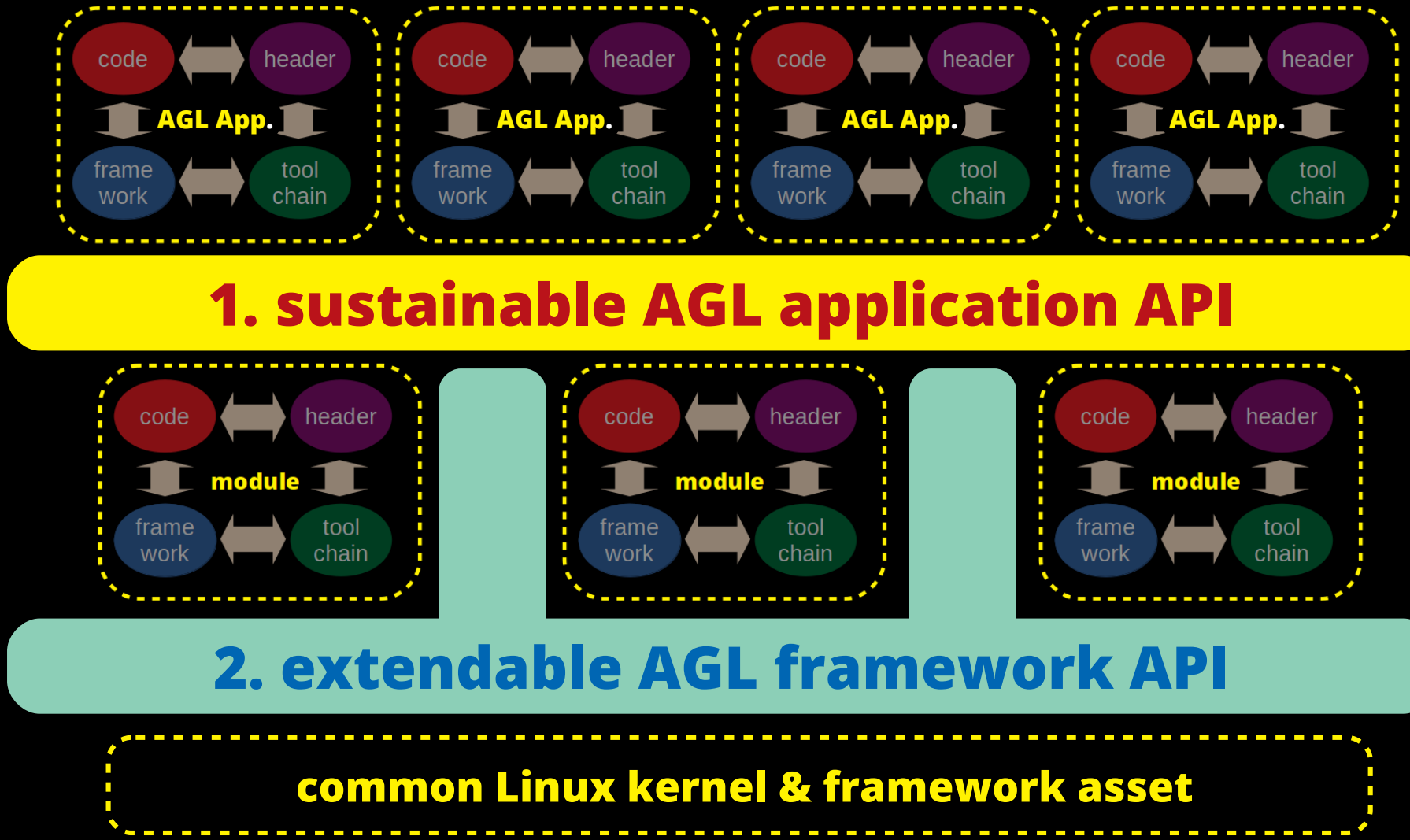


1. decoupling : unbind hidden internal dependency

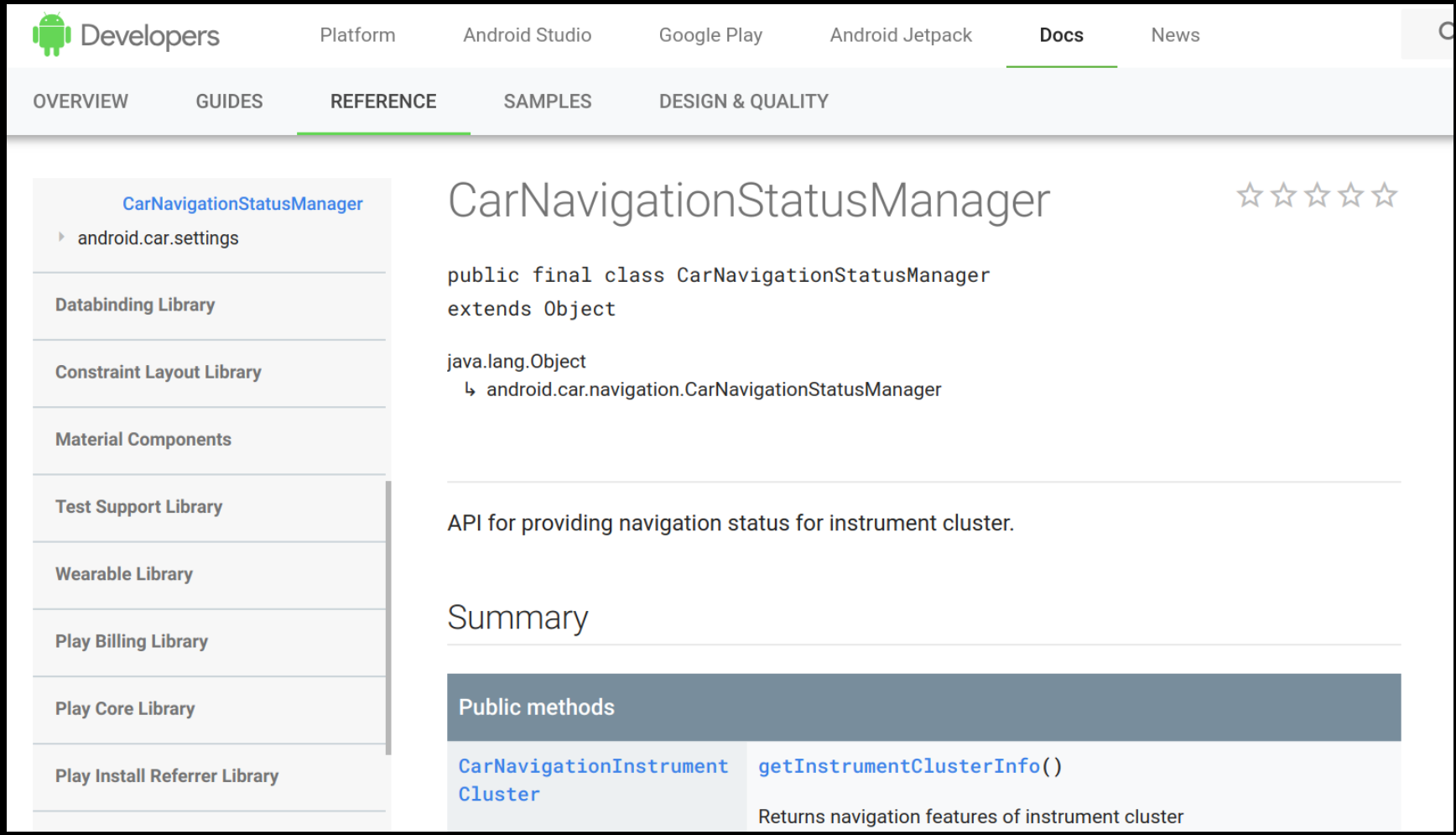


1. decoupling : unbind hidden internal dependency




code
header
AGL App.
frame work
tool chain
code
header
AGL App.
frame work
tool chain
code
header
AGL App.
frame work
tool chain
code
header
AGL App.
frame work
tool chain
1. sustainable AGL application API
code
header
module
frame work
tool chain
code
header
module
frame work
tool chain
code
header
module
frame work
tool chain
2. extendable AGL framework API
common Linux kernel & framework asset

AGL need to declare solid Application API, and ...



Developers | Platform | Android Studio | Google Play | Android Jetpack | Docs | News

OVERVIEW | GUIDES | REFERENCE | SAMPLES | DESIGN & QUALITY

CarNavigationStatusManager
android.car.settings
Databinding Library
Constraint Layout Library
Material Components
Test Support Library
Wearable Library
Play Billing Library
Play Core Library
Play Install Referrer Library

CarNavigationStatusManager

```
public final class CarNavigationStatusManager
extends Object
```

java.lang.Object
↳ android.car.navigation.CarNavigationStatusManager

API for providing navigation status for instrument cluster.

Summary

Public methods	
CarNavigationInstrumentCluster	getInstrumentClusterInfo() Returns navigation features of instrument cluster

And, of course we should learn from OSS development best practice

2. git/gerrit = change management (traceability)

3.10 master remotes/origin/master Merge tag 'armsoc-fixes-4.19' of git://git.kernel.org/pub,	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2018-10-13 00:41:27
Merge tag 'soc-fsl-fix-v4.19-2' of git://git.kernel.org/pub/scm/linux/kernel/git/leo/linux into fixes	Arnd Bergmann <arnd@arndb.de>	2018-10-05 00:37:17
soc: fsl: qman_portals: defer probe after qman's probe	Laurentiu Tudor <laurentiu.tudor@nxp.com>	2018-09-26 22:22:31
soc: fsl: qbman: add APIs to retrieve the probing status	Laurentiu Tudor <laurentiu.tudor@nxp.com>	2018-09-26 22:22:30
Merge tag 'imx-fixes-4.19-2' of git://git.kernel.org/pub/scm/linux/kernel/git/shawnguo/linux int	Arnd Bergmann <arnd@arndb.de>	2018-10-05 00:33:37
ARM: dts: imx53-qsb: disable 1.2GHz OPP	Sascha Hauer <s.hauer@pengutronix.de>	2018-09-12 15:23:01
afs: Fix afs_server struct leak	David Howells <dhowells@redhat.com>	2018-10-12 22:00:57
MAINTAINERS: use the correct location for dt-bindings includes for mux	Peter Rosin <peda@axentia.se>	2018-10-12 23:46:46
mux: adgs1408: use the correct MODULE_LICENSE	Peter Rosin <peda@axentia.se>	2018-10-12 23:46:40
afs: Fix cell proc list	David Howells <dhowells@redhat.com>	2018-10-12 06:45:49
Merge tag 'mmc-v4.19-rc7' of git://git.kernel.org/pub/scm/linux/kernel/git/ulfh/mmc	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2018-10-12 19:57:05
libertas: call into generic suspend code before turning off power	Daniel Mack <daniel@zonque.org>	2018-10-09 05:03:57
mmc: block: avoid multiblock reads for the last sector in SPI mode	Chris Boot <bootc@bootc.net>	2018-10-09 00:07:30
Merge tag 'gpio-v4.19-4' of git://git.kernel.org/pub/scm/linux/kernel/git/linusw/linux-gpio	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2018-10-12 19:56:25
gpio: Assign gpio_irq_chip::parents to non-stack pointer	Stephen Boyd <swboyd@chromium.org>	2018-10-09 01:32:13
Merge tag 'pinctrl-v4.19-5' of git://git.kernel.org/pub/scm/linux/kernel/git/linusw/linux-pinctrl	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2018-10-12 19:55:47
pinctrl: mcp23s08: fix irq and irqchip setup order	Marco Felsch <m.felsch@pengutronix.de>	2018-10-02 17:06:46
Merge tag 'mtd/fixes-for-4.19-rc8' of git://git.infradead.org/linux-mtd	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2018-10-12 19:54:26
lib/bch: fix possible stack overrun	Arnd Bergmann <arnd@arndb.de>	2018-10-11 20:06:17
Merge tag 'drm-fixes-2018-10-12-1' of git://anongit.freedesktop.org/drm/drm	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2018-10-12 19:53:48
Merge branch 'linux-4.19' of git://github.com/skeggsb/linux into drm-fixes	Dave Airlie <airlied@redhat.com>	2018-10-08 15:37:56
drm/nouveau/drm/nouveau: Grab runtime PM ref in nv50_mstc_detect()	Lyude Paul <lyude@redhat.com>	2018-09-15 05:44:03
Merge tag 'for-gkh' of git://git.kernel.org/pub/scm/linux/kernel/git/rdma/rdma	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2018-10-12 19:53:06
IB/mlx5: Unmap DMA addr from HCA before IOMMU	Valentine Fatiev <Valentinef@mellanox.com>	2018-10-10 15:56:25
Merge branch 'for-linus' of git://git.kernel.org/pub/scm/linux/kernel/git/dtor/input	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2018-10-12 19:35:02
Input: uinput - add a schedule point in uinput_inject_events()	Dmitry Torokhov <dmitry.torokhov@gmail.com>	2018-10-05 09:50:48
Input: evdev - add a schedule point in evdev_write()	Dmitry Torokhov <dmitry.torokhov@gmail.com>	2018-10-05 09:45:54

collection of small atomic piece with change logs

3. CICD :
Continuous Integration Continuous Delivery

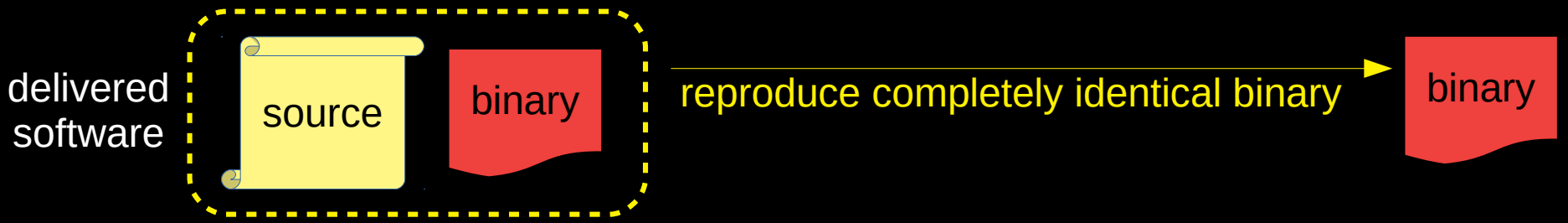
to avoid dirty code injection

If not fully pass the test, force shift to the next release cycle

https://reproducible-builds.org/

4. Reproducible builds

Reproducible builds are a set of software development practices that create an independently-verifiable path from source code to the binary code used by computers.



Conclusion

- **Creation of long-term sustainable SW asset is highly demanded by the industry But it is a big challenge against the SW entropy.**

- **“Decoupling” is the key to success.**
  (SW partitioning, git, CICD, reproducible build)