Asymmetric Multi-Processing (AMP) for Embedded - Where Are We?

Wolfram Sang, Renesas / Consultant

07.10.2026, ELC Europe

1. Introduction
2. AMP buildings
3. AMP Building Blocks
4. Building Blocks Health Status
5. Off-the-Shelf?
6. Conclusion

Introduction

Hands up!

Hands up!

- Who has AMP running on some board?

Hands up!

- Who has AMP running on some board?
- ... using only Free Software?

Hands up!

- Who has AMP running on some board?
- ... using only Free Software?
- ... fully upsteam?

Hands up!

- Who has AMP running on some board?
- ... using only Free Software?
- ... fully upsteam?
- *Who thinks that hwspinlocks can improve protection of a virtio-msg based remoteproc channel to ensure the validity of SCMI protocols (virtualized)?*

Pandora's box (or luckily I am not alone)

Pandora's box (or luckily I am not alone)

Wednesday

- 09:50 Geert Uytterhoeven (right here):
  Marrying Device Tree and SCMI
- 11:20 Iuliana Prodan (South Hall 2B):
  Ten Years of Zephyr: Replace Linux, or Work With It?
- 16:30 Marek Vasut (right here):
  Porting Hafnium SPM to a Modern ARM SoC
- 17:45 Dorinda Bassey, et al. (South Hall 1B, LPC):
  BoF: Virtio and Vhost-User Ecosystem for Production and Automotive Systems

Pandora's box (or luckily I am not alone) II

Thursday

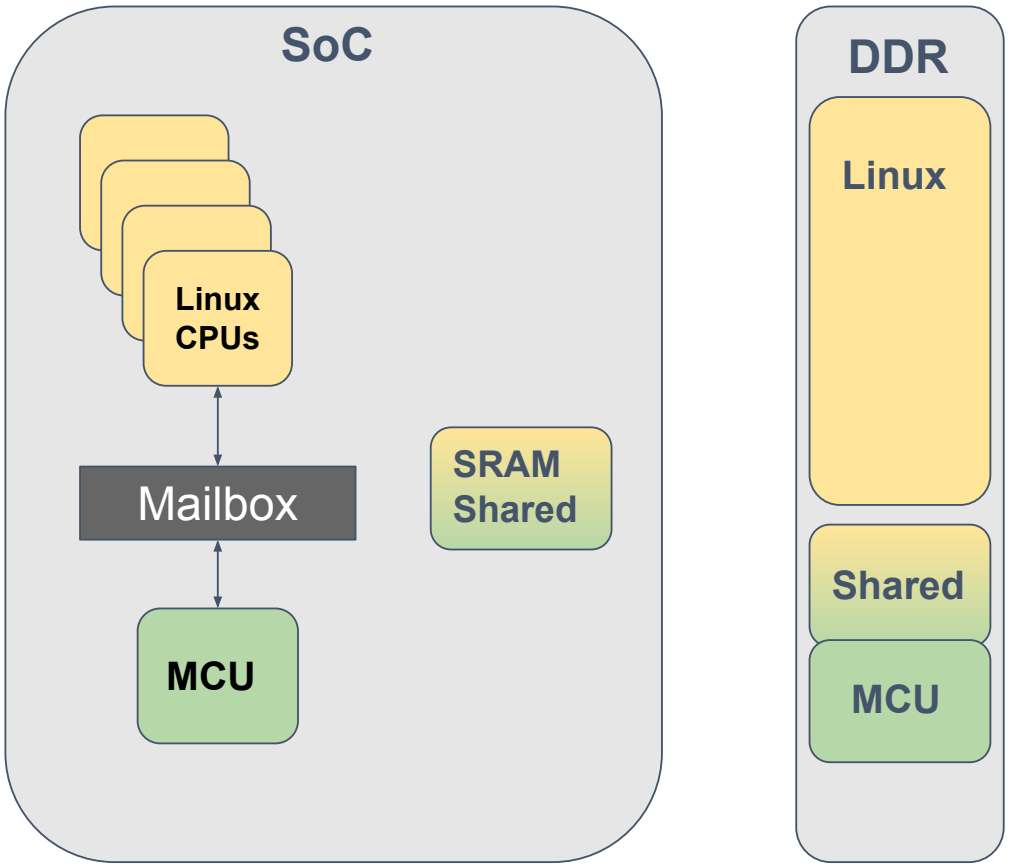
- 10:50 Harunobu Kurokawa (South Hall 2A):
  Zephyr and Xen in Automotive Mixed-OS Systems: Building Safety Readiness and Upstream

Friday

- 14:50 Hiroshi Tokita (South Hall 2B):
  Zephyr-Based VIRTIO Backend on Xen: From Tiny Driver Domains To Real Hardware Integration

AMP buildings

OpenAMP - basic



SCMI

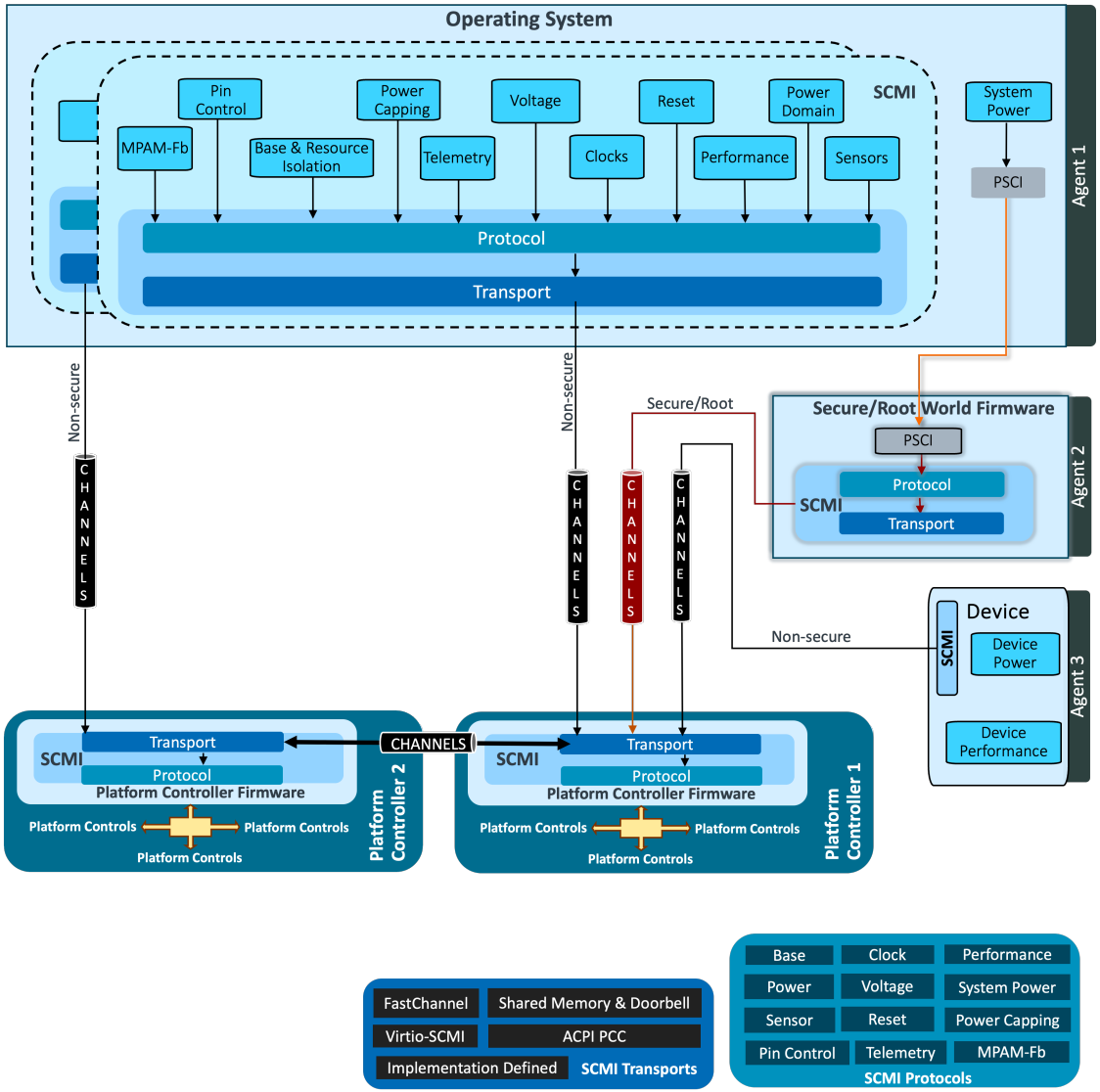
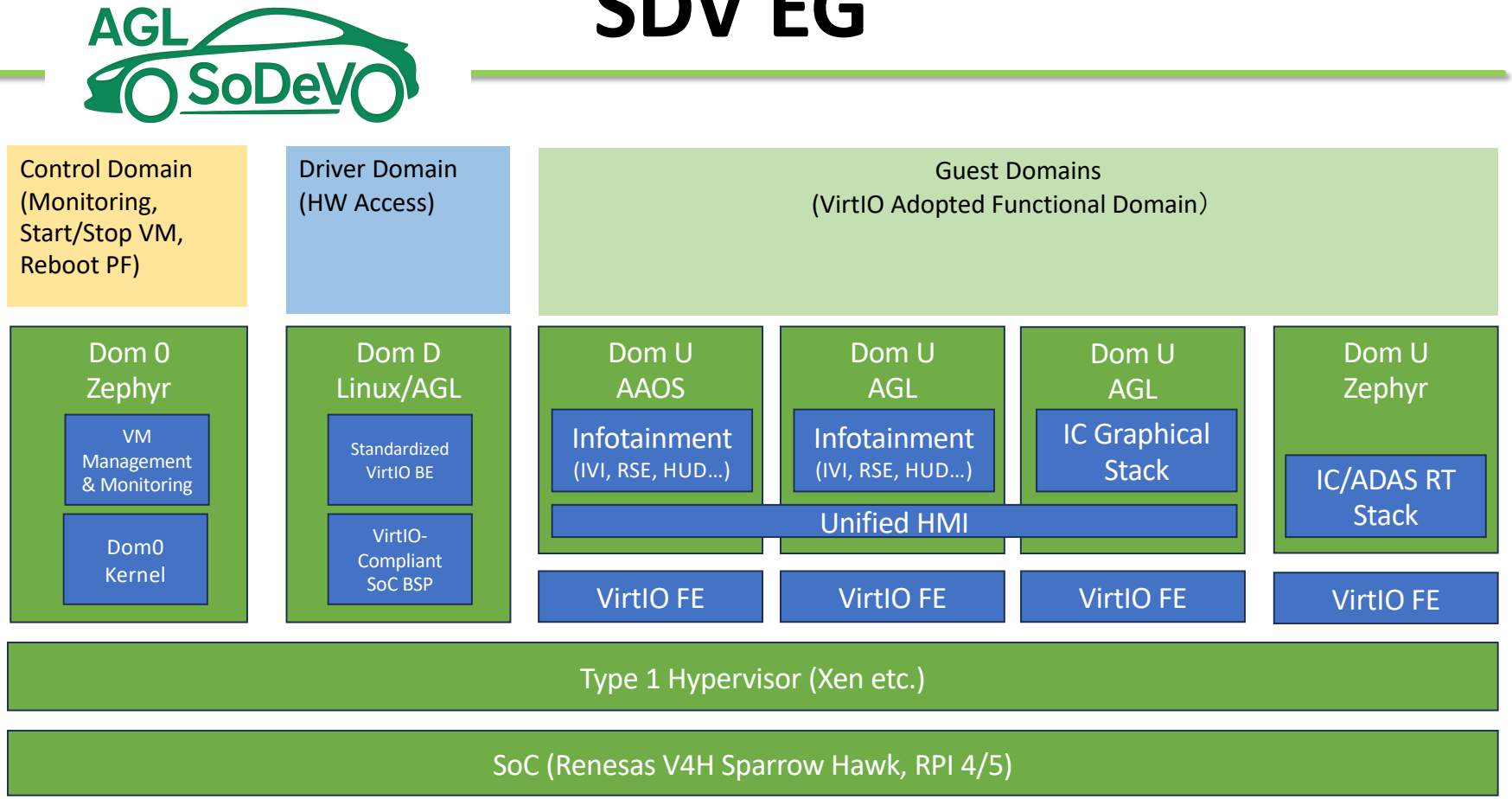


Figure 1: SCMI Overview

More fun with SCMI

Stay in this room
for Geert's talk

SDV EG


Control Domain
(Monitoring,
Start/Stop VM,
Reboot PF)
Driver Domain
(HW Access)
Guest Domains
(VirtIO Adopted Functional Domain)
Dom 0
Zephyr
VM
Management
& Monitoring
Dom0
Kernel
Dom D
Linux/AGL
Standardized
VirtIO BE
VirtIO-
Compliant
SoC BSP
Dom U
AAOS
Infotainment
(IVI, RSE, HUD…)
Dom U
AGL
Infotainment
(IVI, RSE, HUD…)
Dom U
AGL
IC Graphical
Stack
Dom U
Zephyr
IC/ADAS RT
Stack
Unified HMI
VirtIO FE
VirtIO FE
VirtIO FE
VirtIO FE
Type 1 Hypervisor (Xen etc.)
SoC (Renesas V4H Sparrow Hawk, RPI 4/5)

AMP Building Blocks

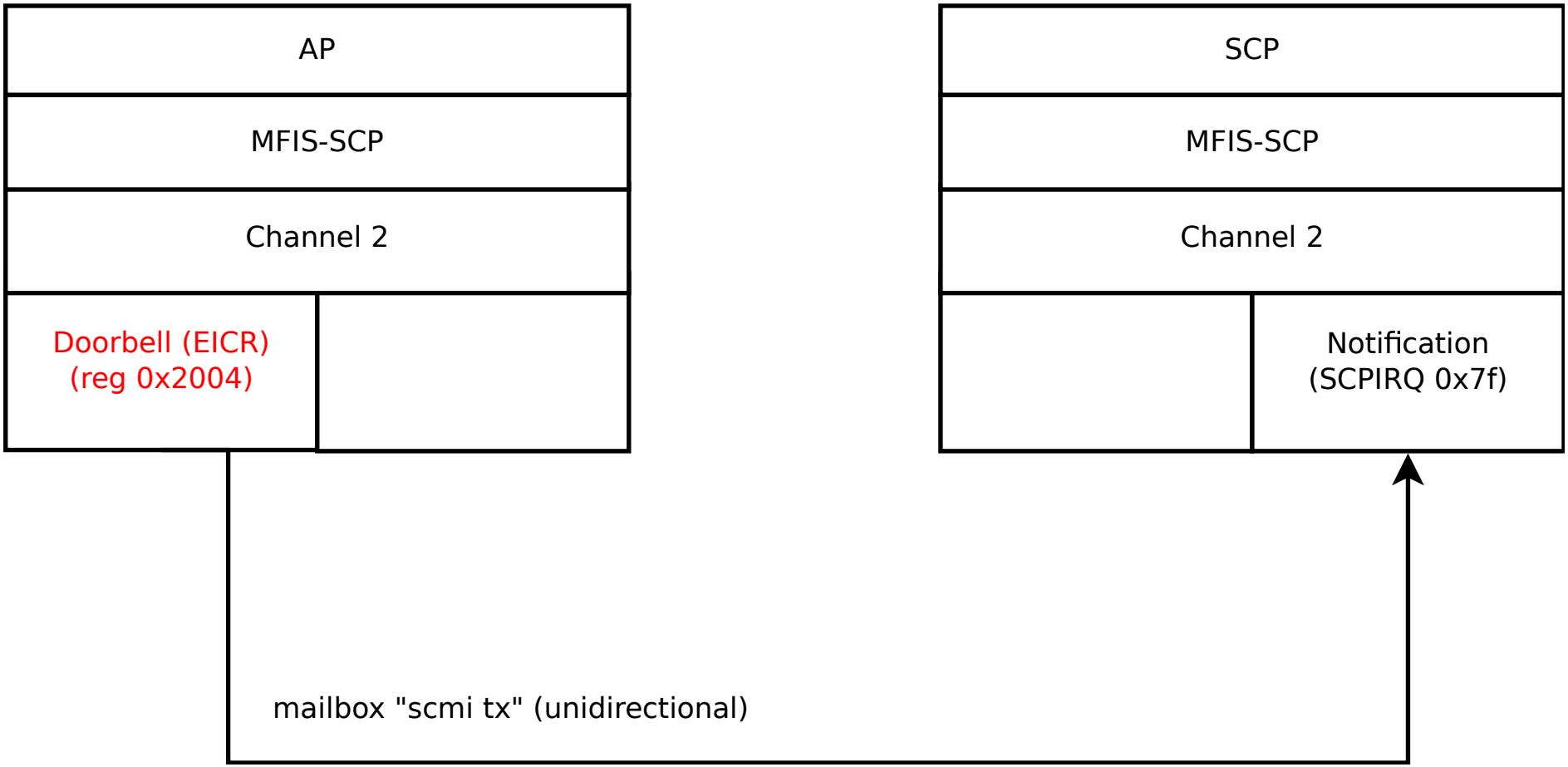
hardware spinlocks

- take / release lock
- all components need to respect it
- implemented by Ohad Ben-Cohen in 2011
- live in `drivers/hwspinlock`
- kind of last resort

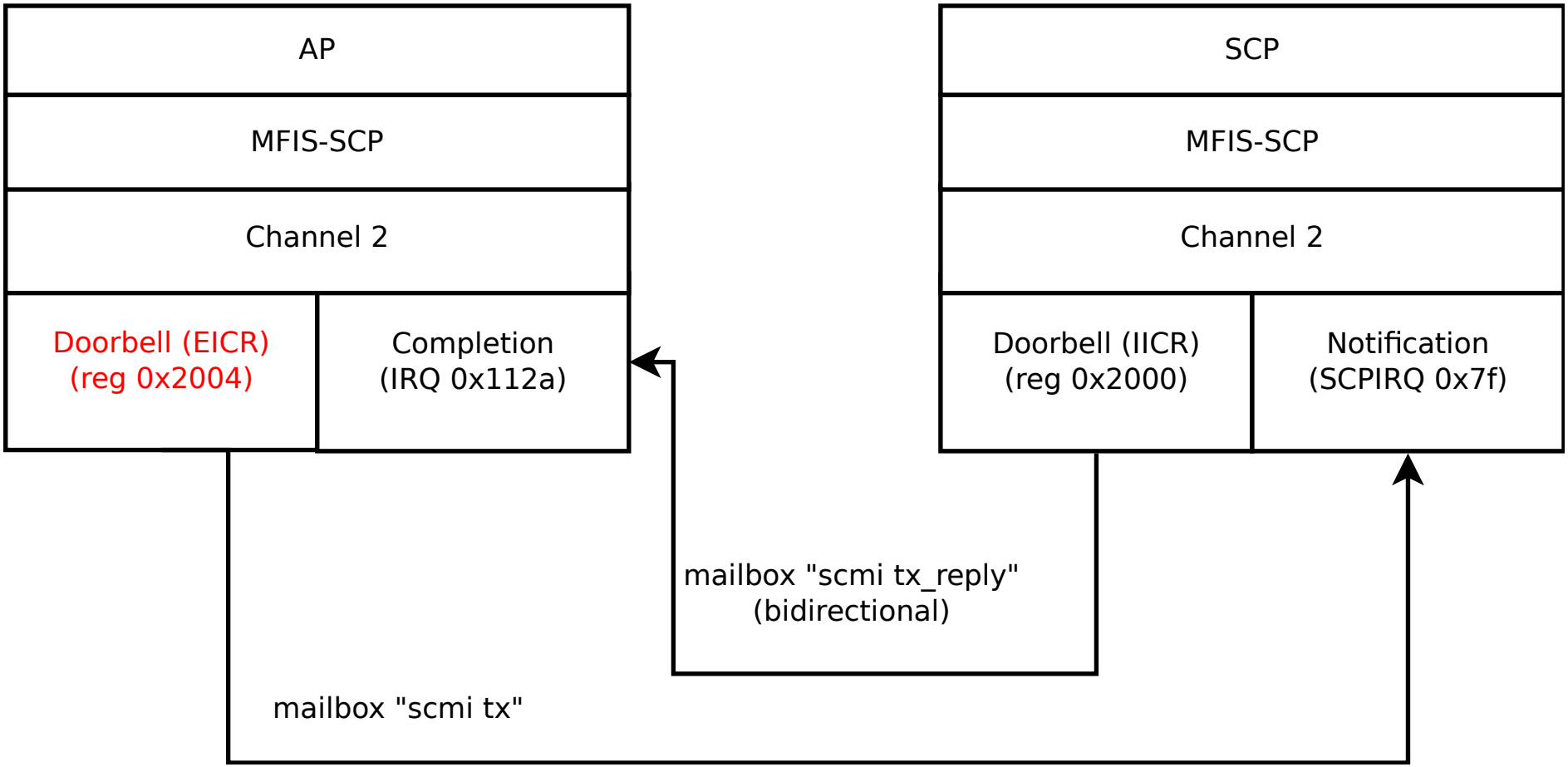
mailboxes

- generic core implemented by Jassi Brar in 2014
- live in `drivers/mailbox`
- serialization ("locking") is often inherent

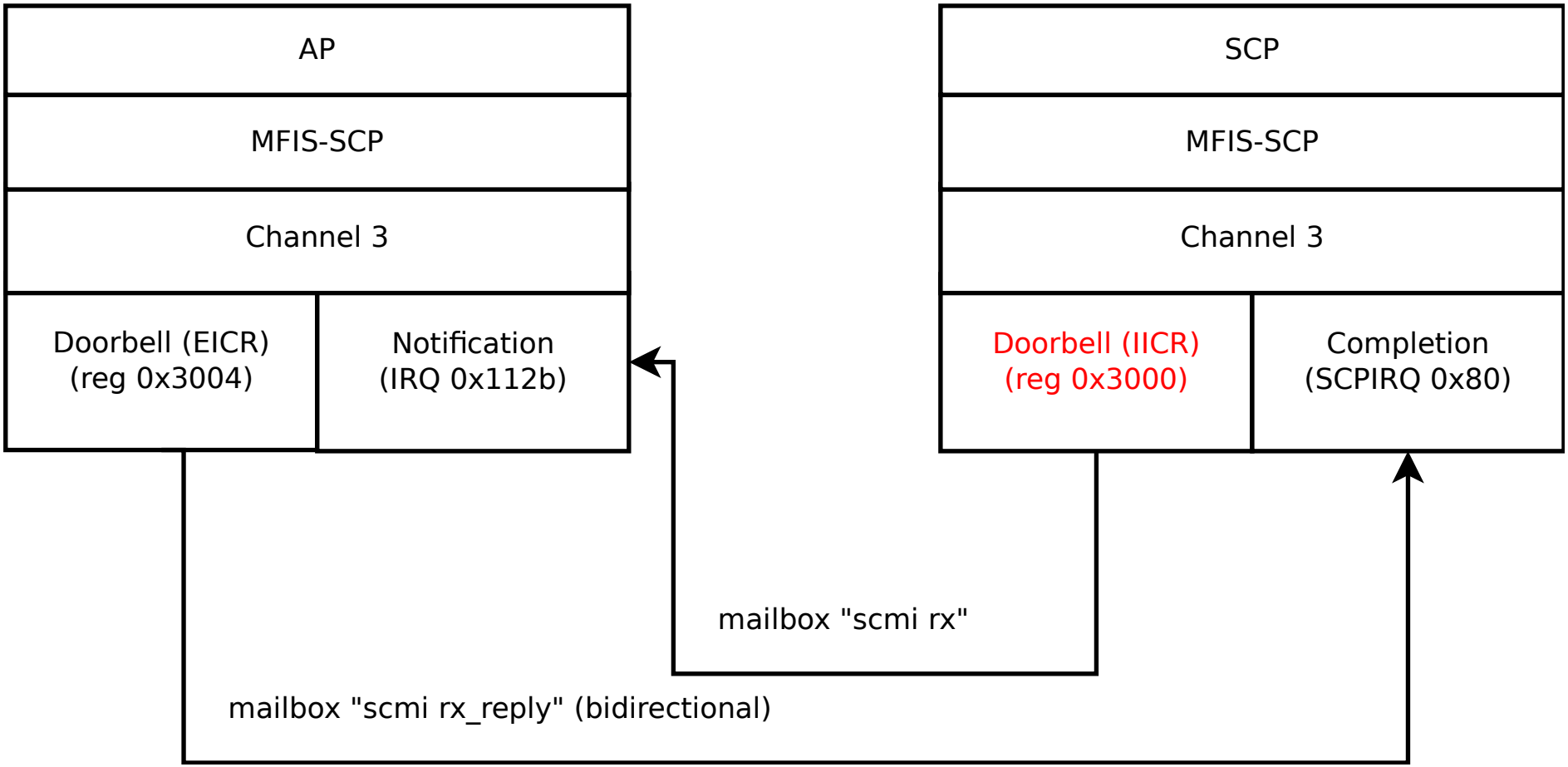
Mailbox (unidirectional)



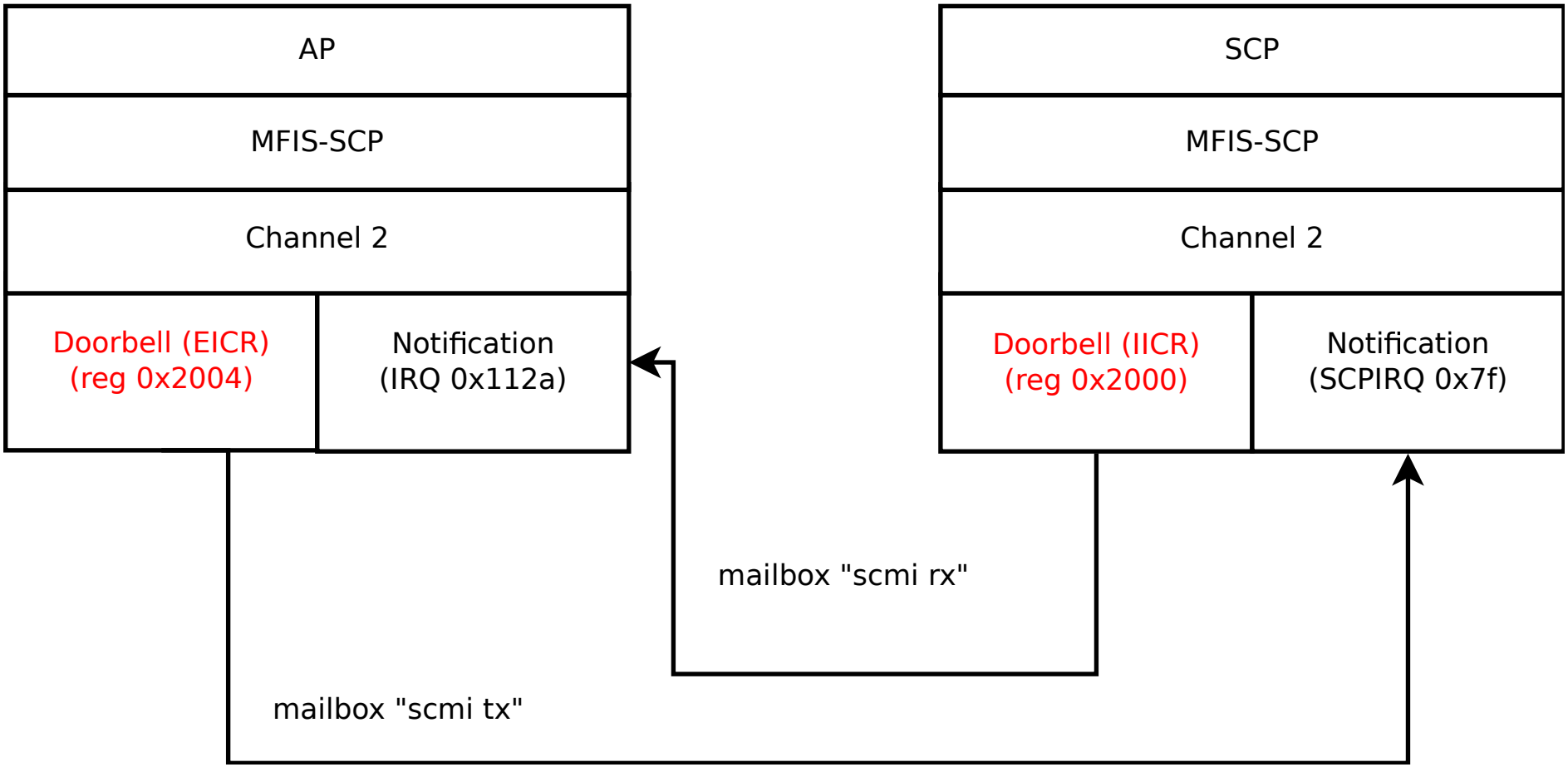
Mailbox (bidirectional)



Second Pair of Mailboxes (makes SCMI bidirectional)



One bidirectional mailbox treated as two unidirectional mailboxes (SCMI bidirectional)



remoteproc

- implemented by Ohad Ben-Cohen in 2012
- abstraction layer to start / stop remote cores
- starting means to upload firmware to the core
- definition of remote resources in the binary
- live in `drivers/remoteproc`
- exists in U-Boot as well

virtio

- protocol for communication between frontends and backends

  or drivers and devices or ...
- open standard
- communication via shared memory

  virtqueues are ring buffers
- virtqueues can be kicked / flushed
- transport is PCI or MMIO (as of now)

remoteproc II

- remote firmware can include virtio devices
- also has a `kick()` callback
- virtqueues are kicked

  most use mailboxes for this kick

rpmsg

- transfers messages to the remote core
- represented as a bus with remote endpoints
- can have custom transports
- most use virtio-based transfers supported by remoteproc

  you get it for free if you have a `kick()` callback
- implemented by Ohad Ben-Cohen in 2012
- live in `drivers/rpmsg`

Take a breath

Okay so far?

virtio-msg

- transport layer for virtio

  virtio over messages (e.g. via trusted zones)
- not messages over virtio

  that would be rpmsg
- rpmsg over virtio-msg is technically possible

  Holodeck-in-Holodeck anyone?
- latest patches here

  WIP, not upstream yet

DeviceTree

System DeviceTree

- one DT describes all(!) the hardware
- a tool will create subsets for each OS/domain/...
- one solution, but THE solution?

Firmware DT bindings

- some remote controllers do not need a DT
- but talking to them needs description
- stay in this room for Geert's talk

Building Blocks Health Status

Why this talk

wanted to upstream hwspinlock driver

Why this talk

wanted to upstream hwspinlock driver

not possible outside `hwspinlock` directory because of internal header

Why this talk

wanted to upstream hwspinlock driver

not possible outside `hwspinlock` directory because of internal header

wanted to refactor internal header

Why this talk

wanted to upstream hwspinlock driver

not possible outside `hwspinlock` directory because of internal header

wanted to refactor internal header

needed core refactoring first to prevent making too much internal data public

Why this talk II

wanted to refactor core

Why this talk II

wanted to refactor core

Well, while here we could refactor everything so allocations are finally done in the core and not in drivers

Why this talk II

wanted to refactor core

Well, while here we could refactor everything so allocations are finally done in the core and not in drivers

wanted to refactor everything

Why this talk II

wanted to refactor core

Well, while here we could refactor everything so allocations are finally done in the core and not in drivers

wanted to refactor everything

- difficult because there are no debug capabilities to prove I didn't break anything
- got dozens of Sashiko reports about life-cycle issues, too

Why this talk III

wanted to add debug capabilities

Why this talk III

wanted to add debug capabilities

Needed an extension of the radix-tree API

Why this talk III

wanted to add debug capabilities

Needed an extension of the radix-tree API

wanted to extend the radix-tree API

Why this talk III

wanted to add debug capabilities

Needed an extension of the radix-tree API

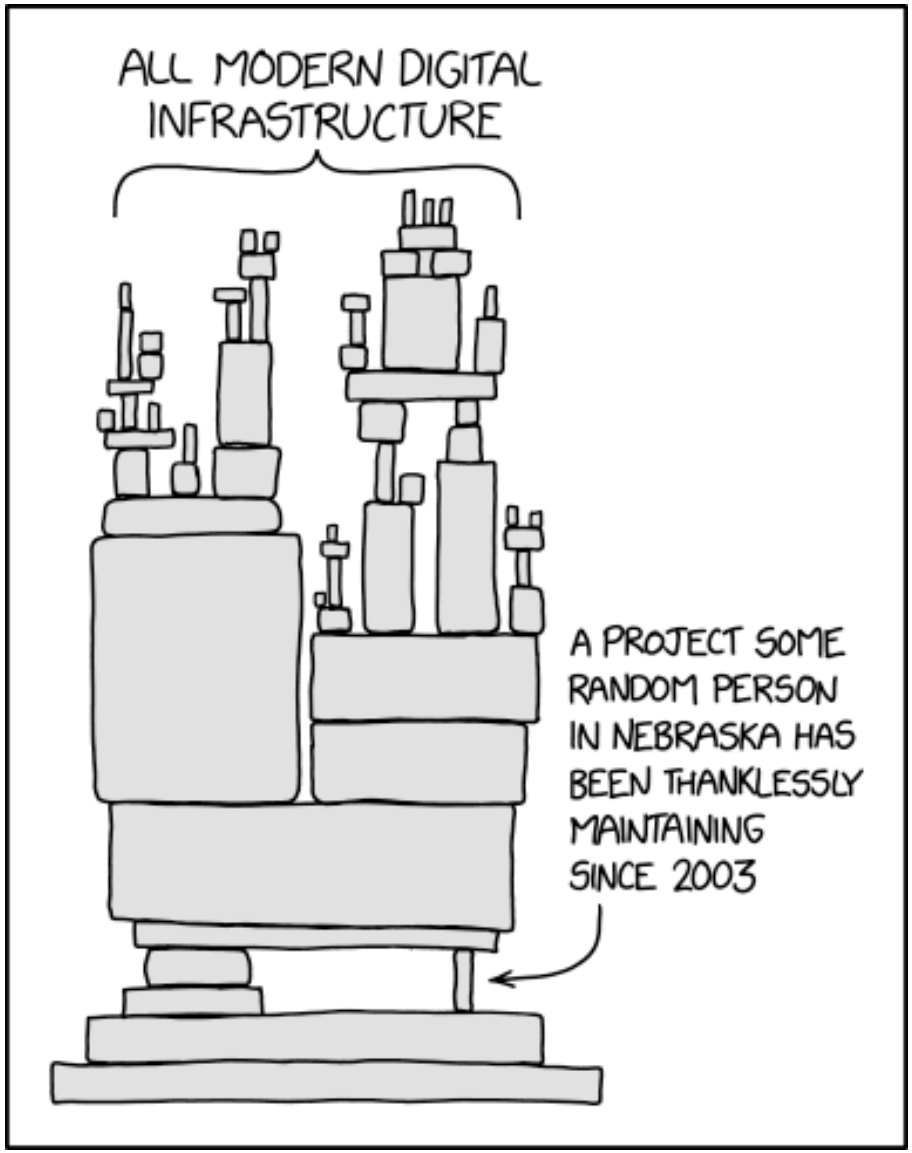
wanted to extend the radix-tree API

Rejected. radix-tree is deprecated.

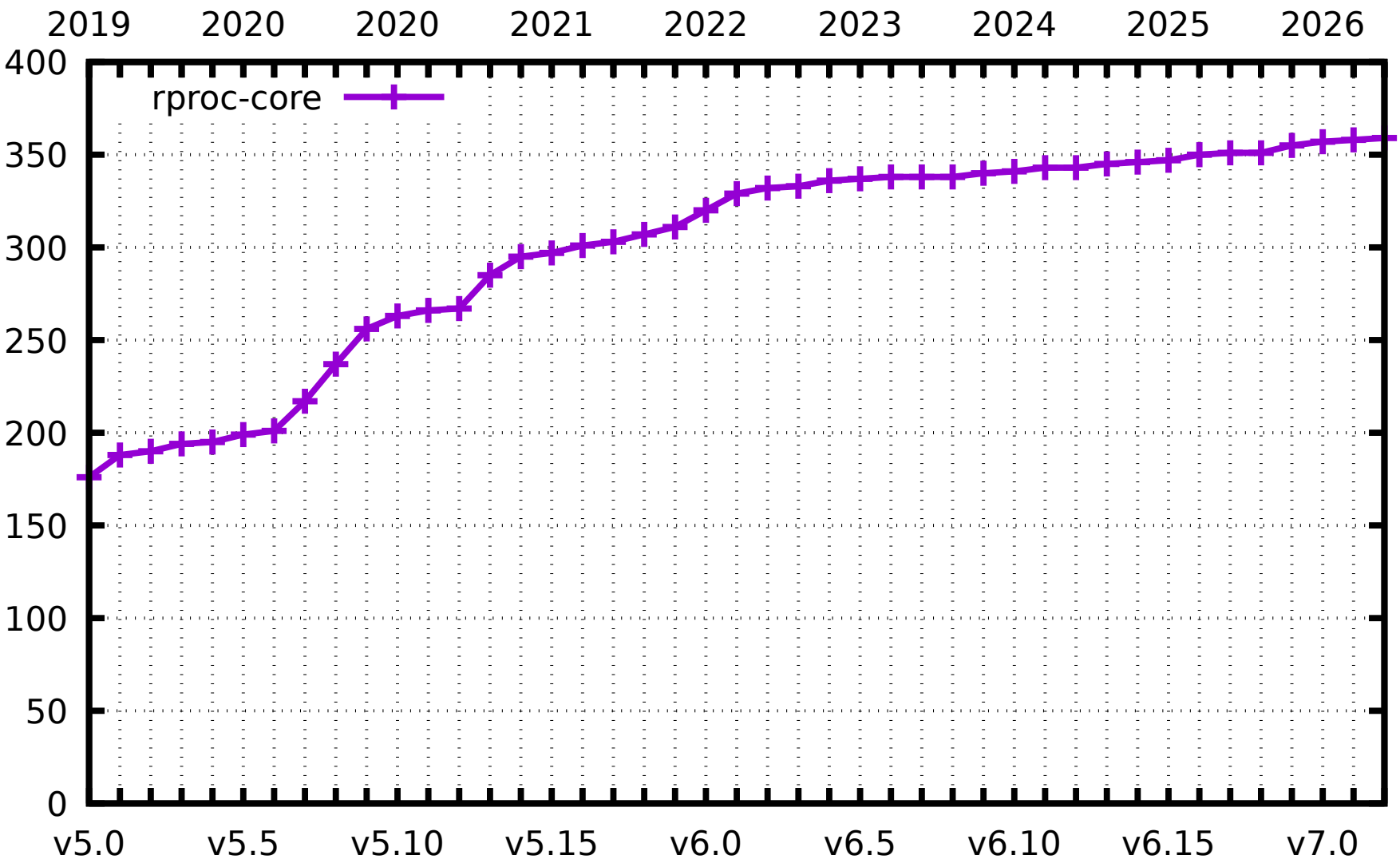
So, I wondered...

On top of this, we want to do AMP??

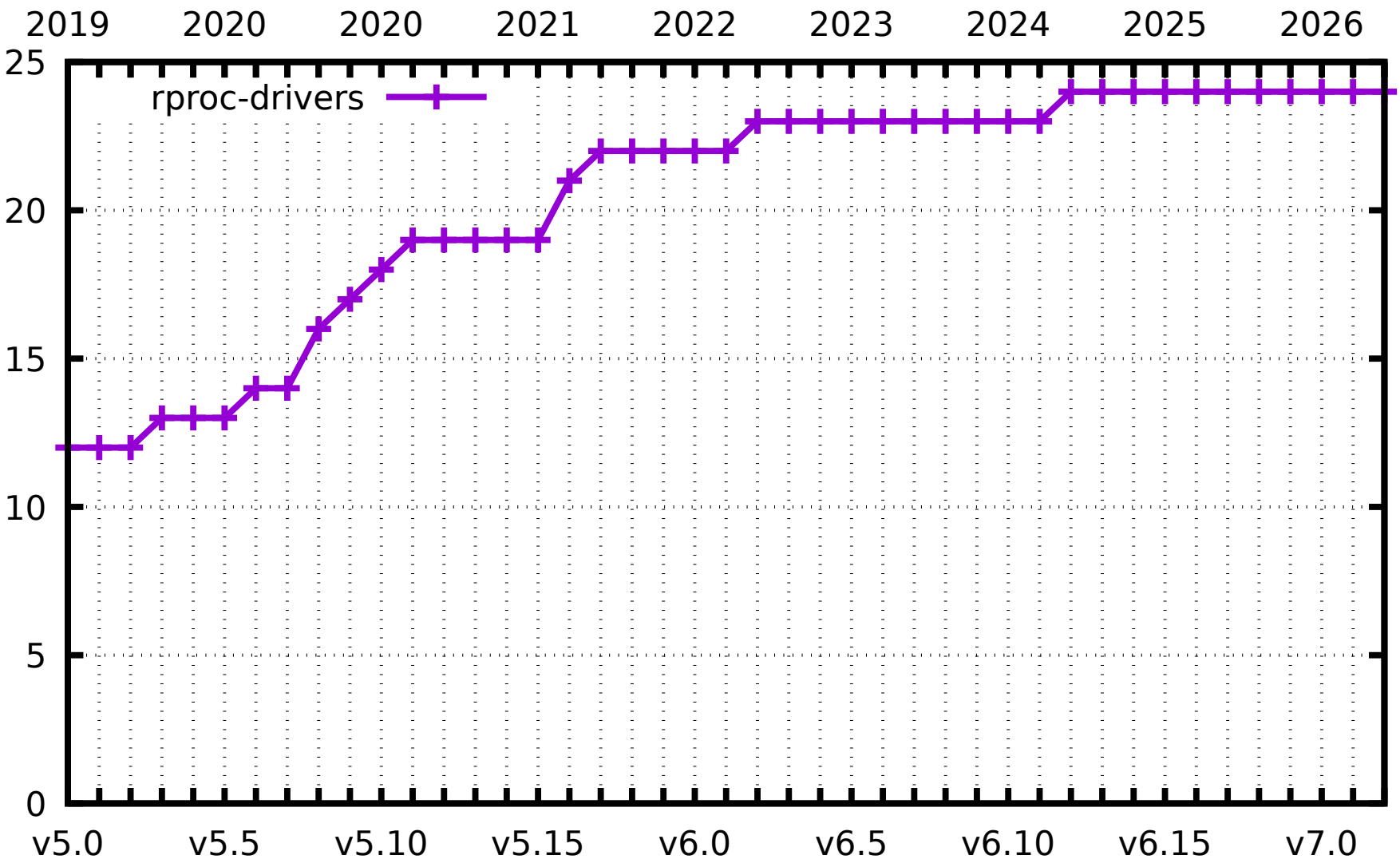
The obvious xkcd #2347



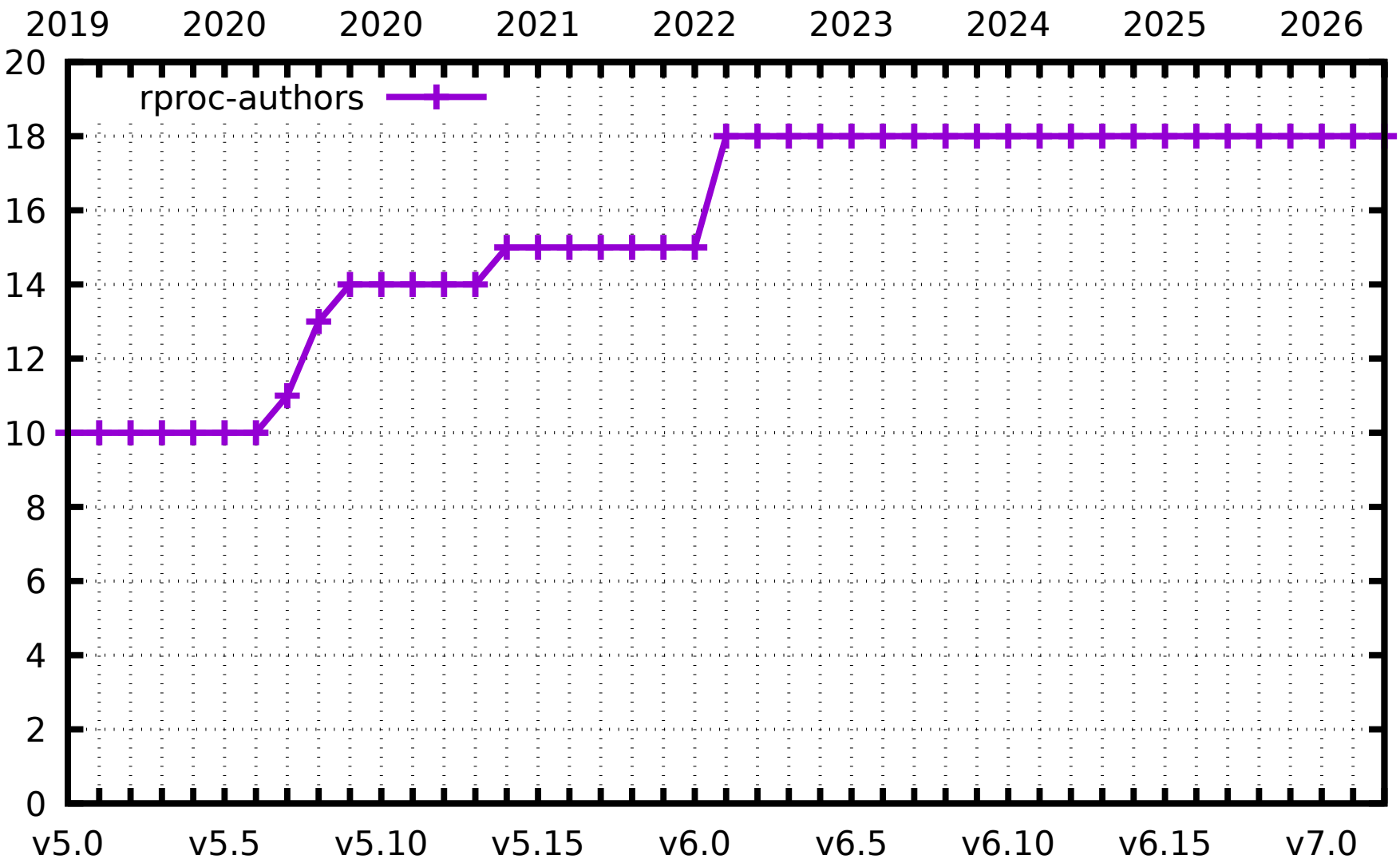
remoteproc (# core commits)



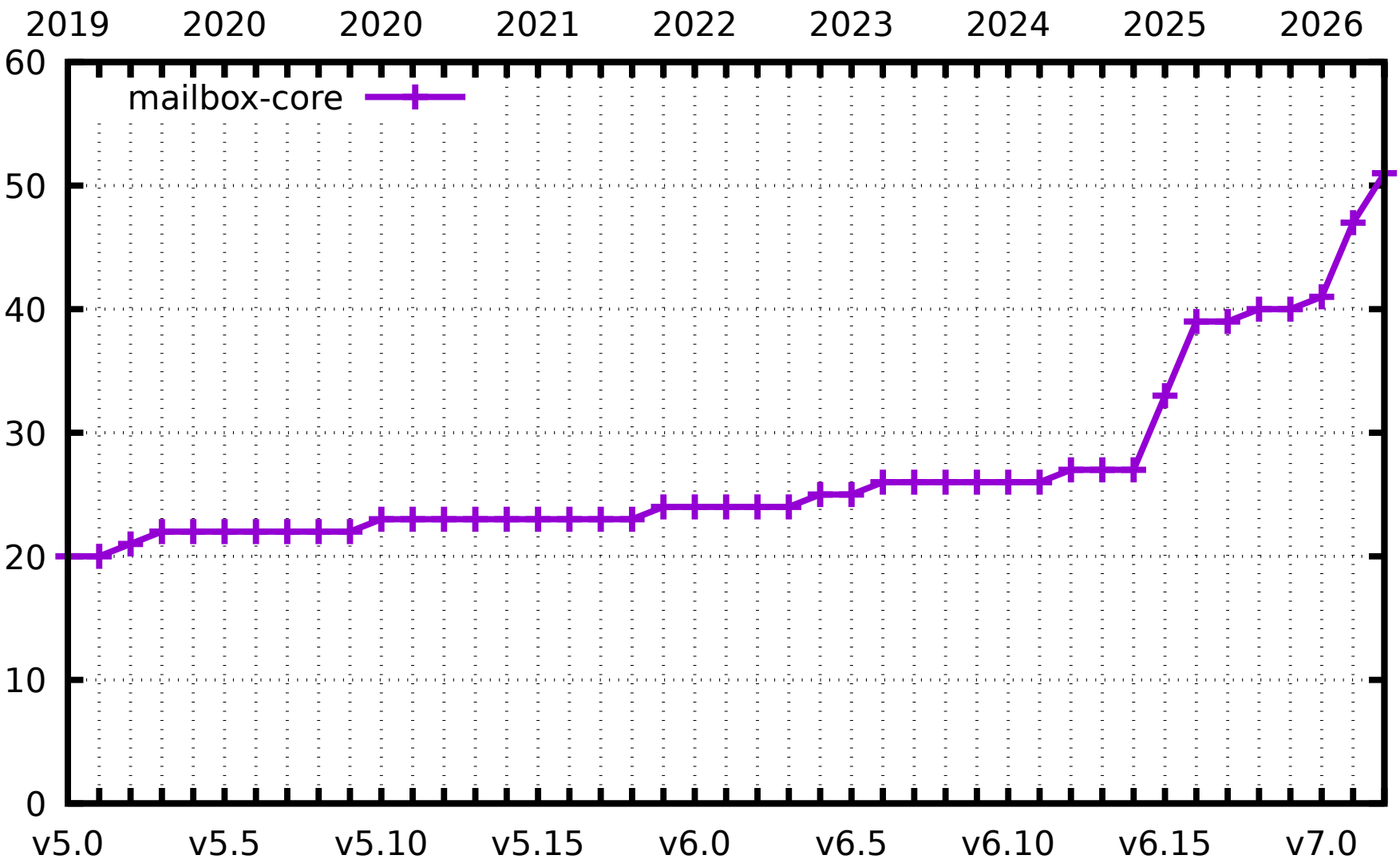
remoteproc (# drivers)



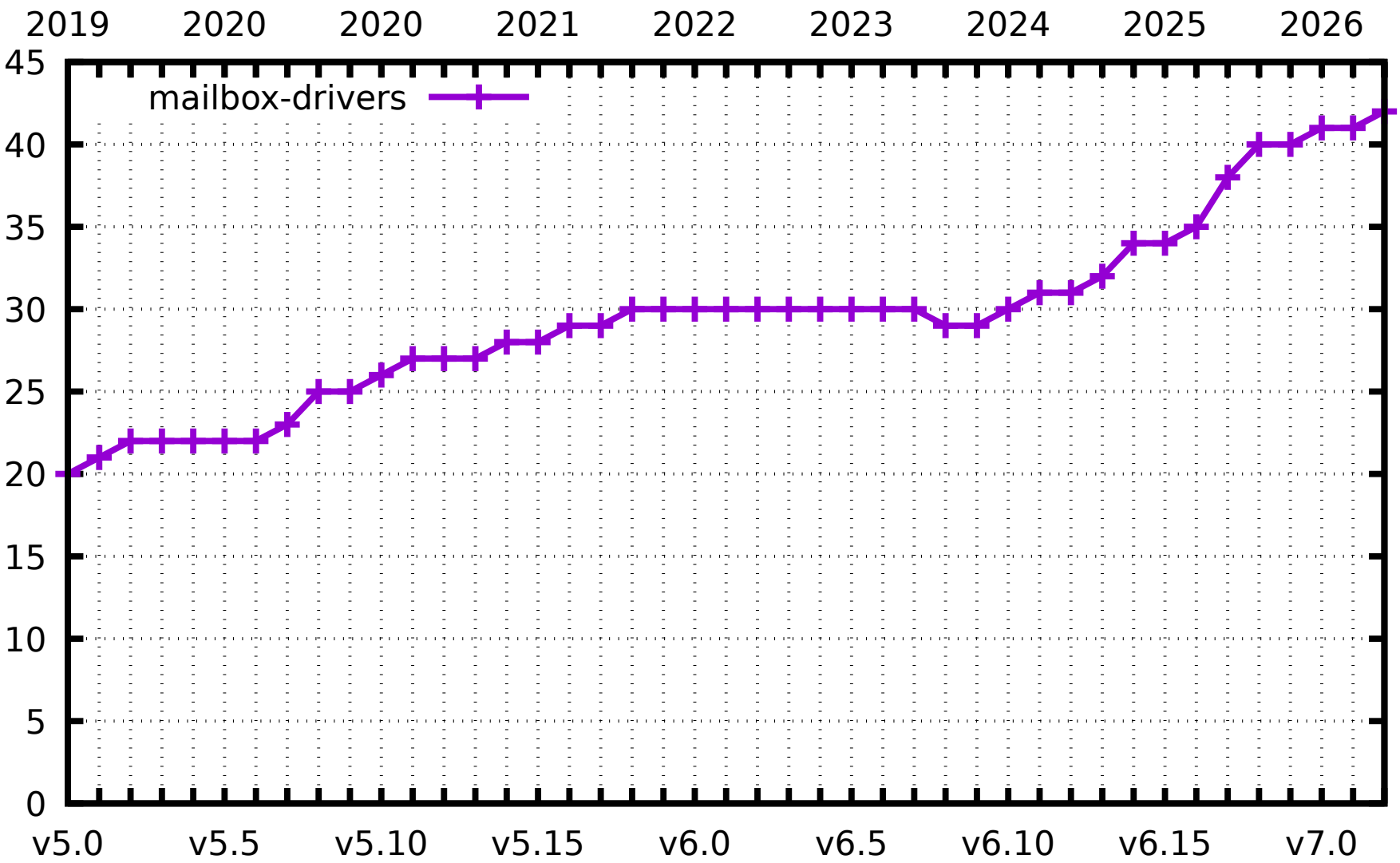
remoteproc (#authors >5 patches)



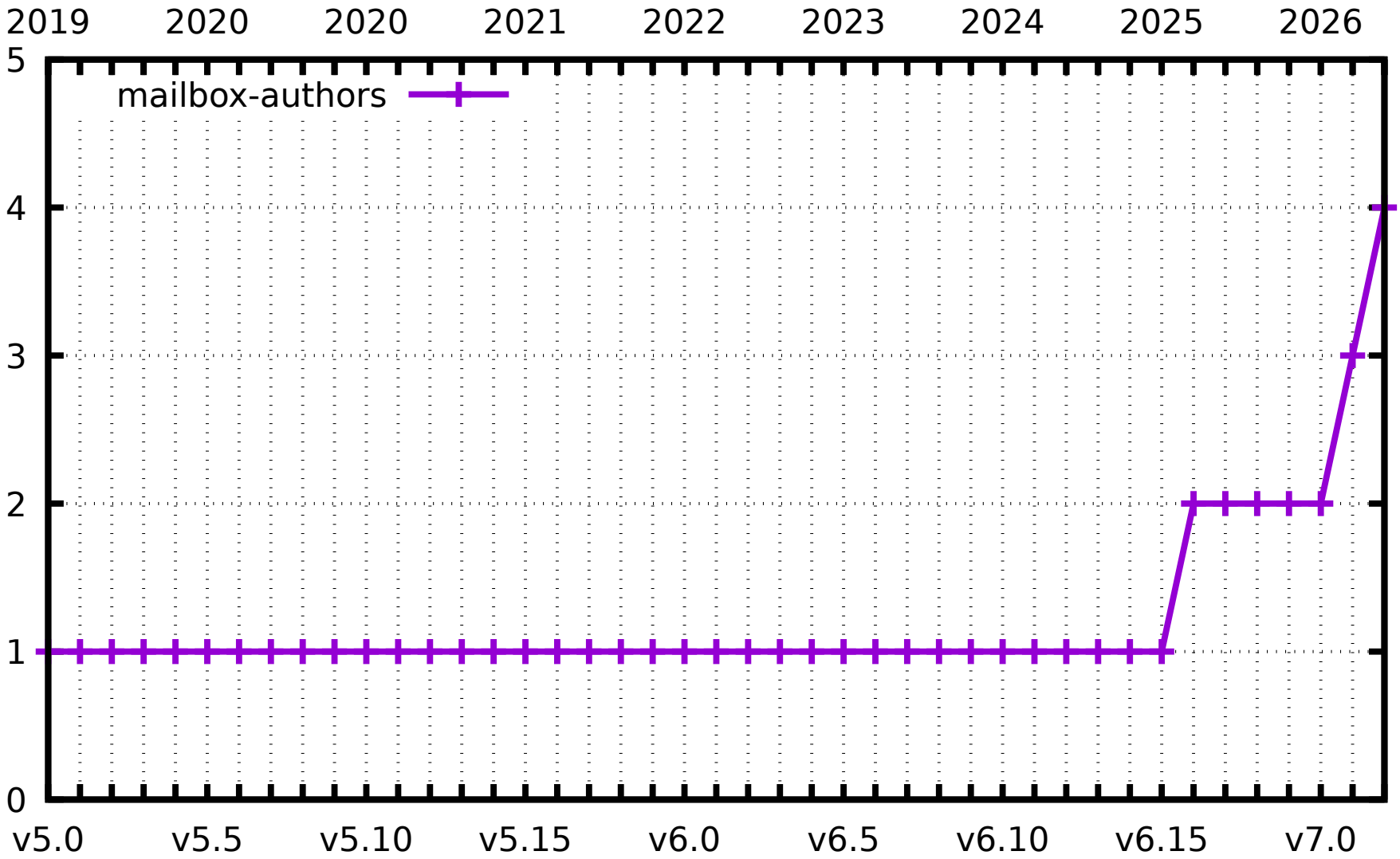
mailbox (# core commits)



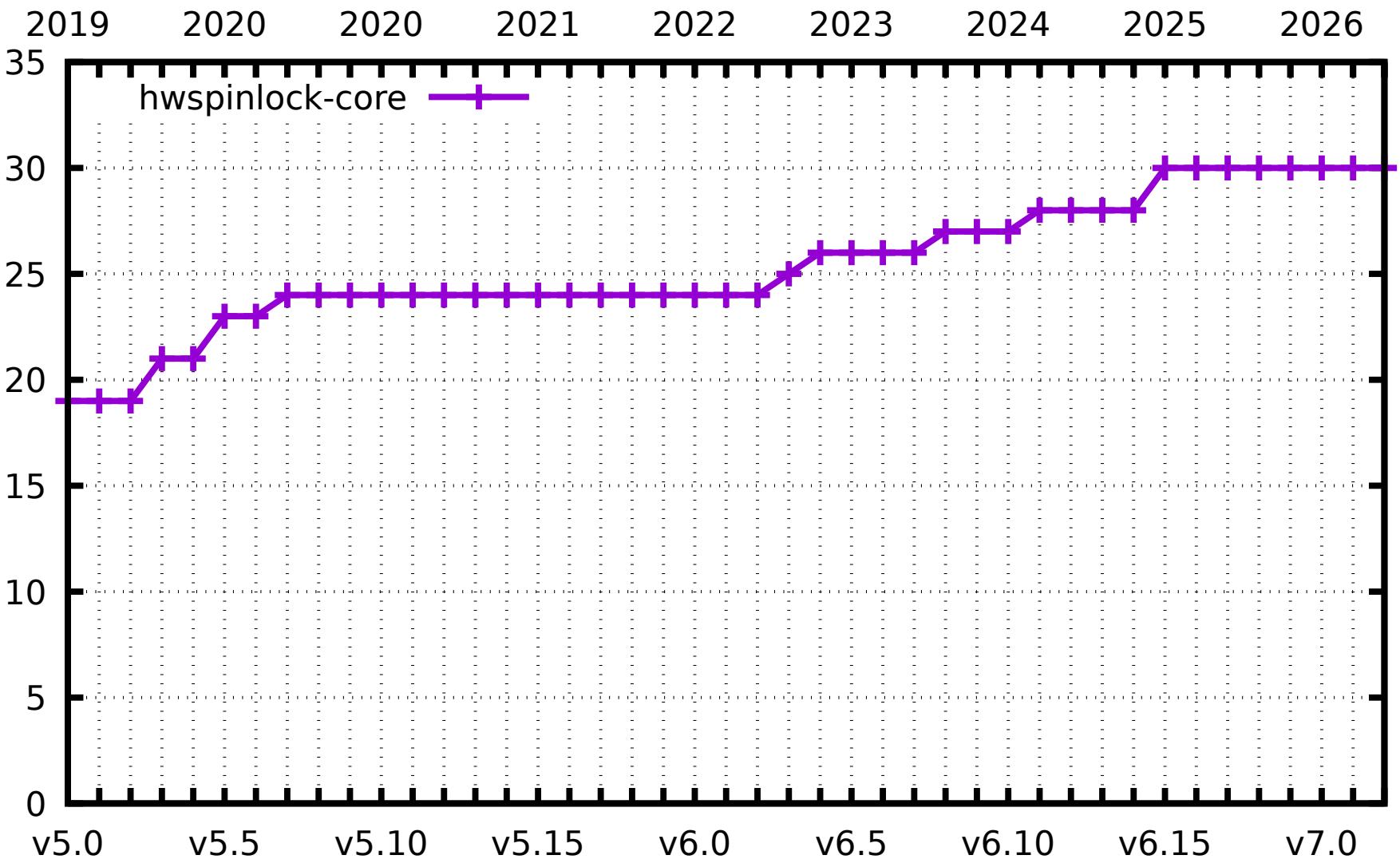
mailbox (# drivers)



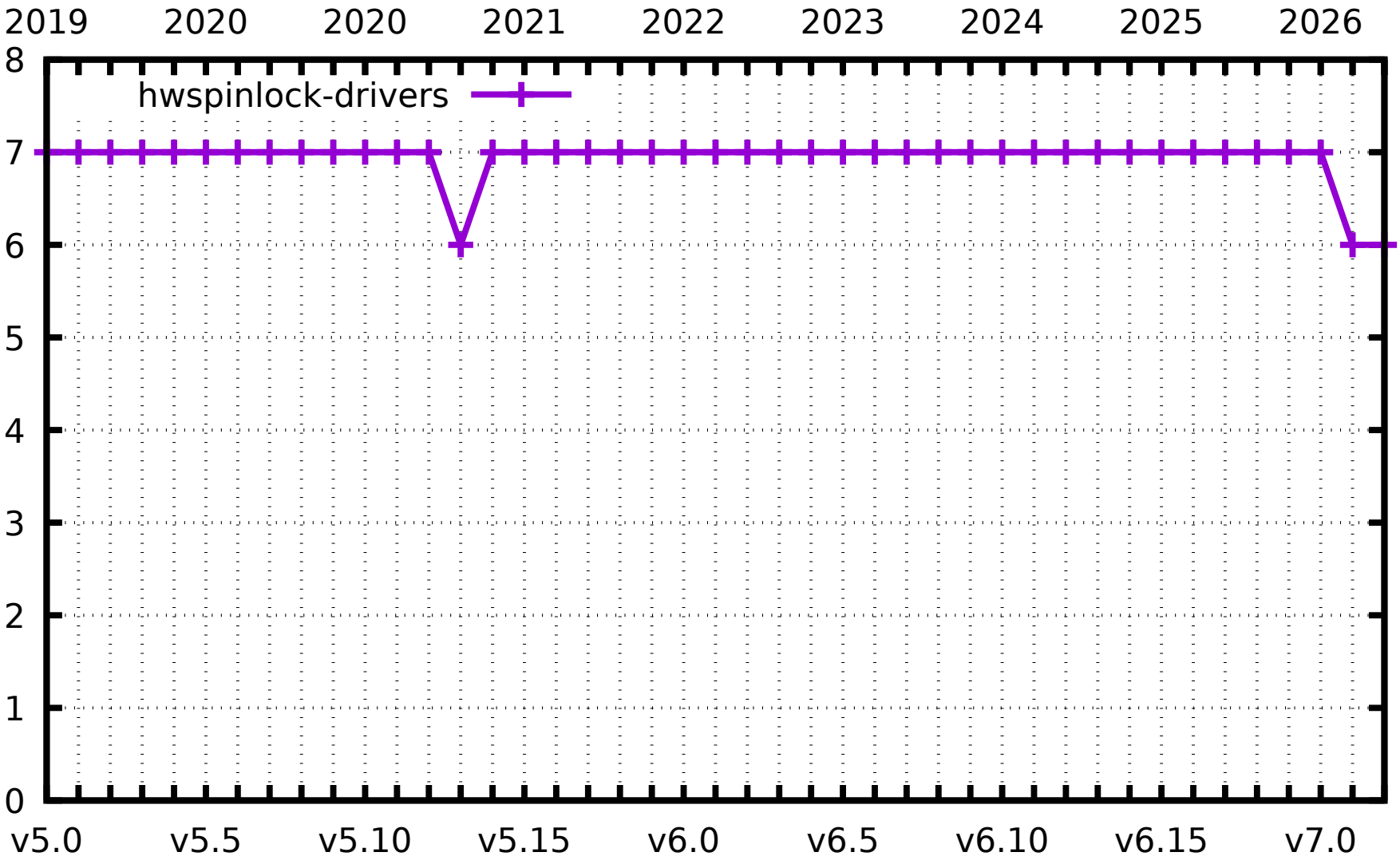
mailbox (#authors >5 patches)



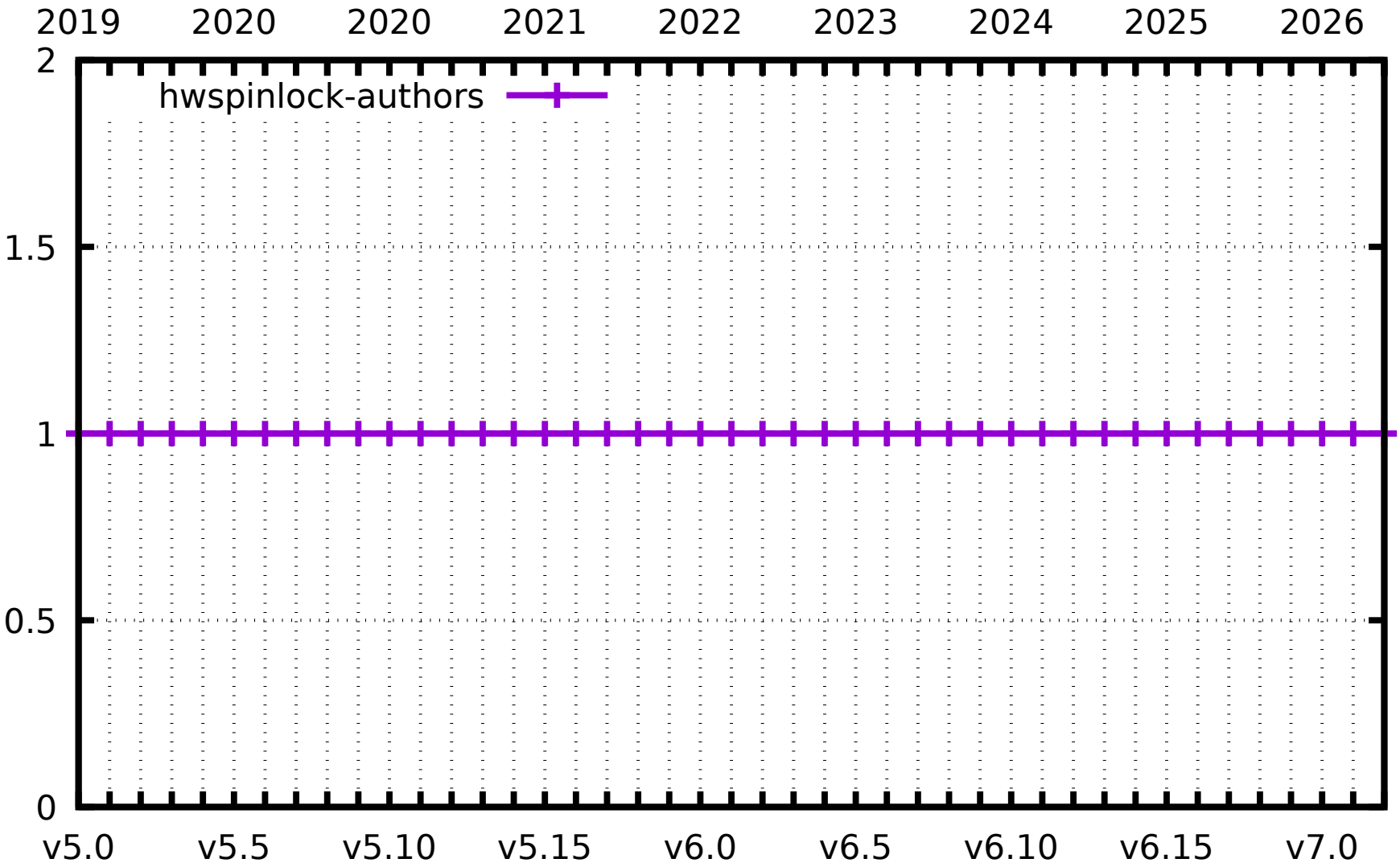
hwspinlock (# core commits)



hwspinlock (# drivers)



hwspinlock (#authors >5 patches)

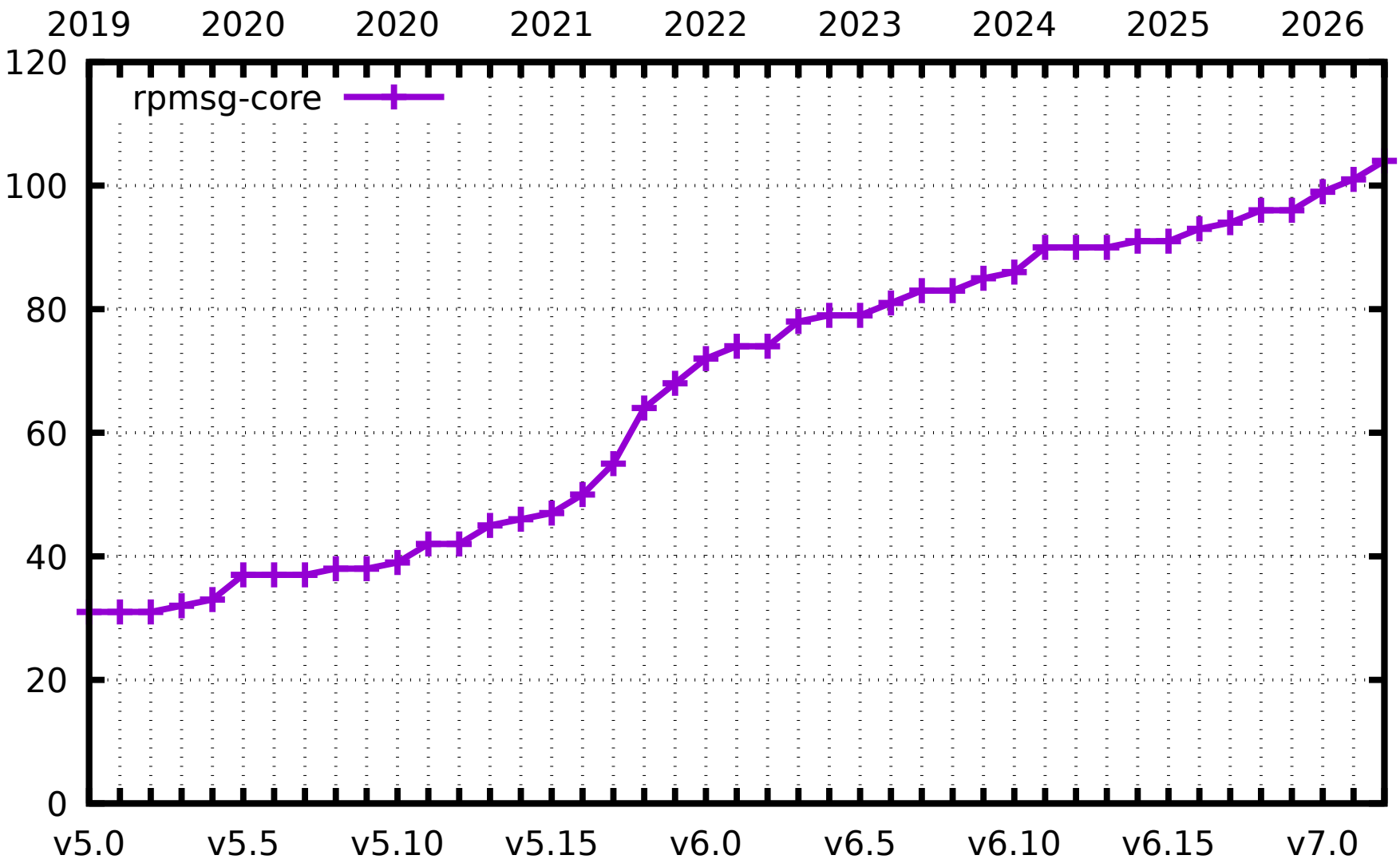


personal remarks

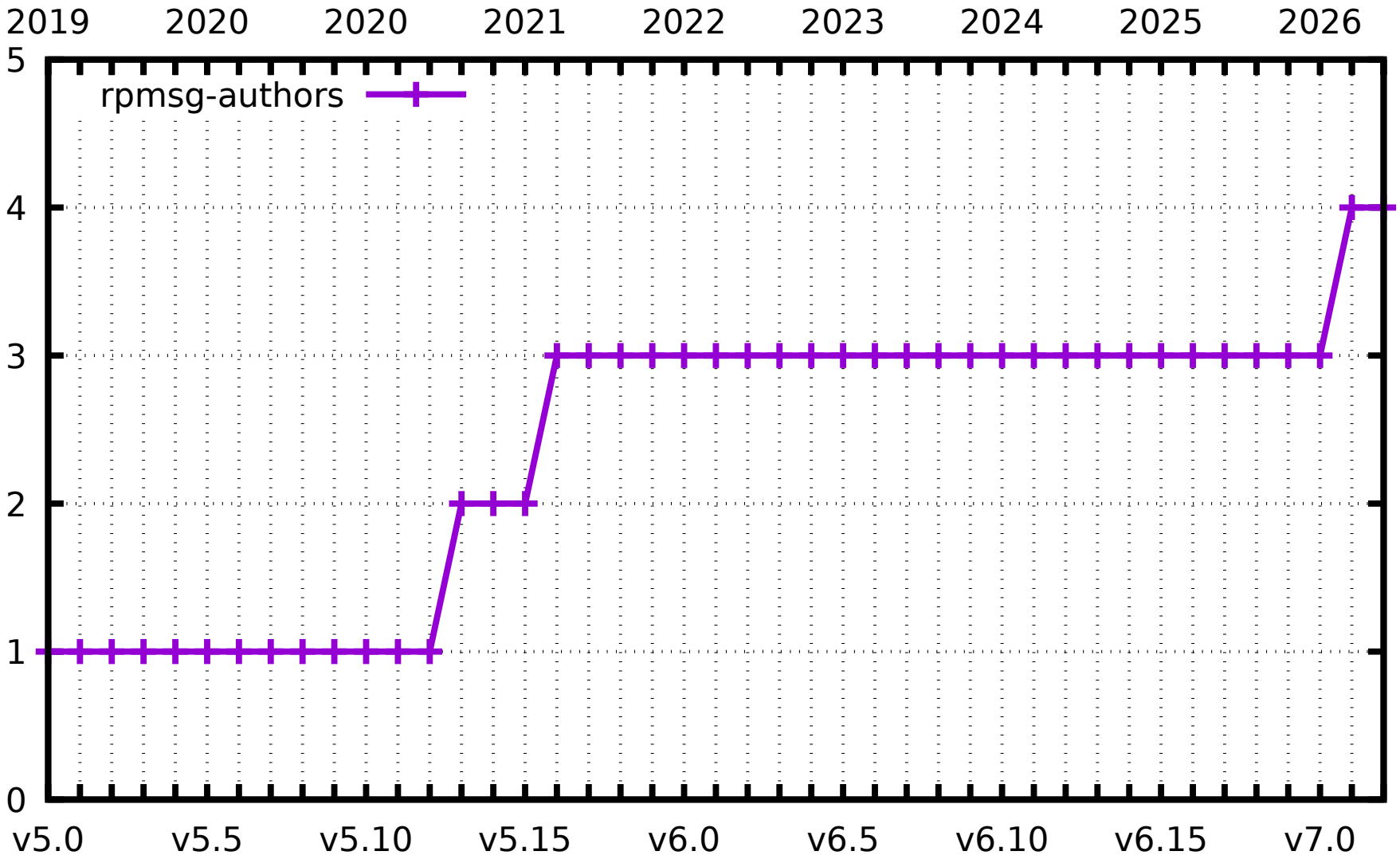
- worked with the `hwspinlock` and `mailbox` subsystems personally
- `hwspinlock` was slow, `mailbox` was fast
- review was always thorough
- subsystem pain points are already known
- usual suspects, maintainer overload

Still good so far?

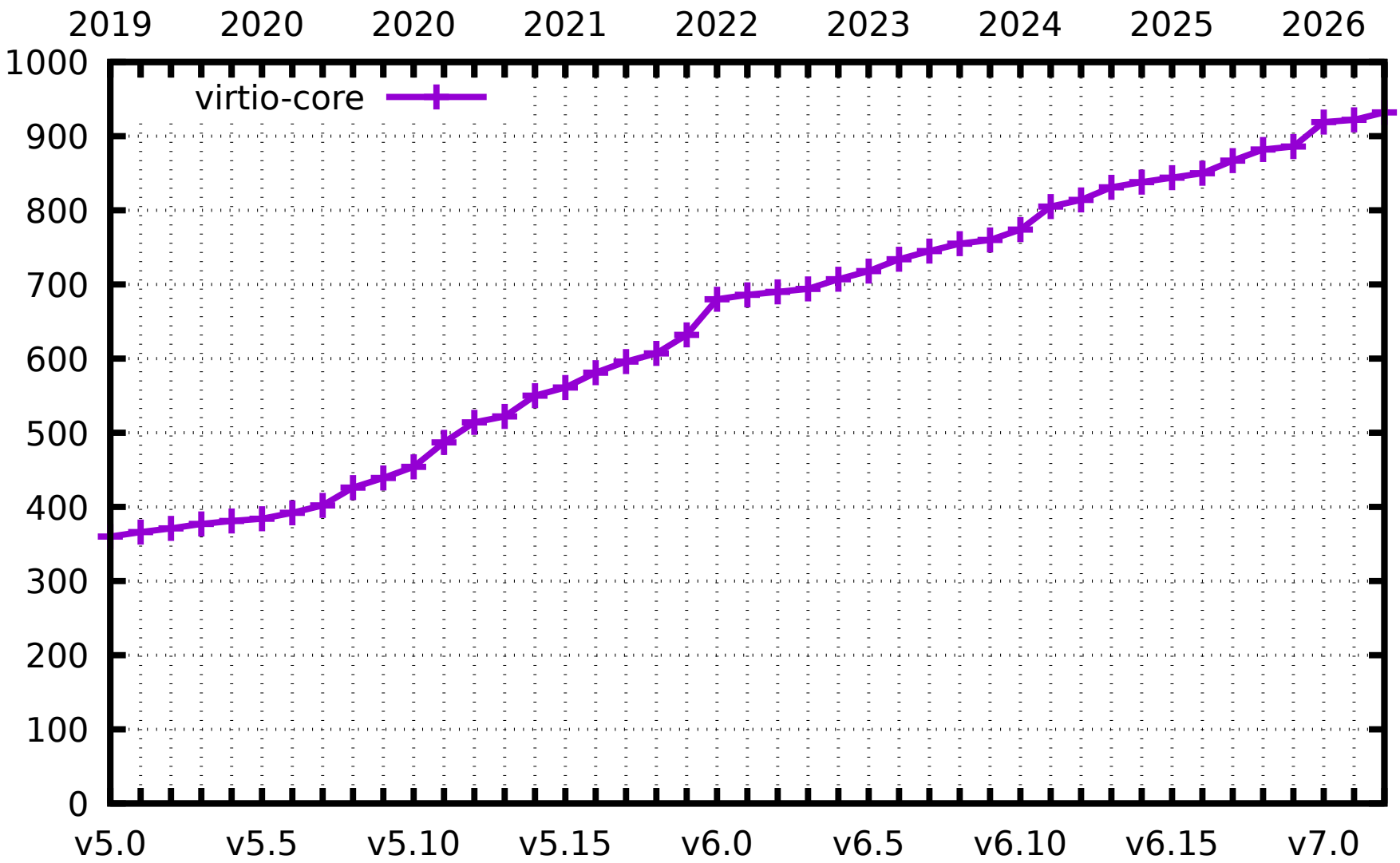
rpmsg (# core commits)



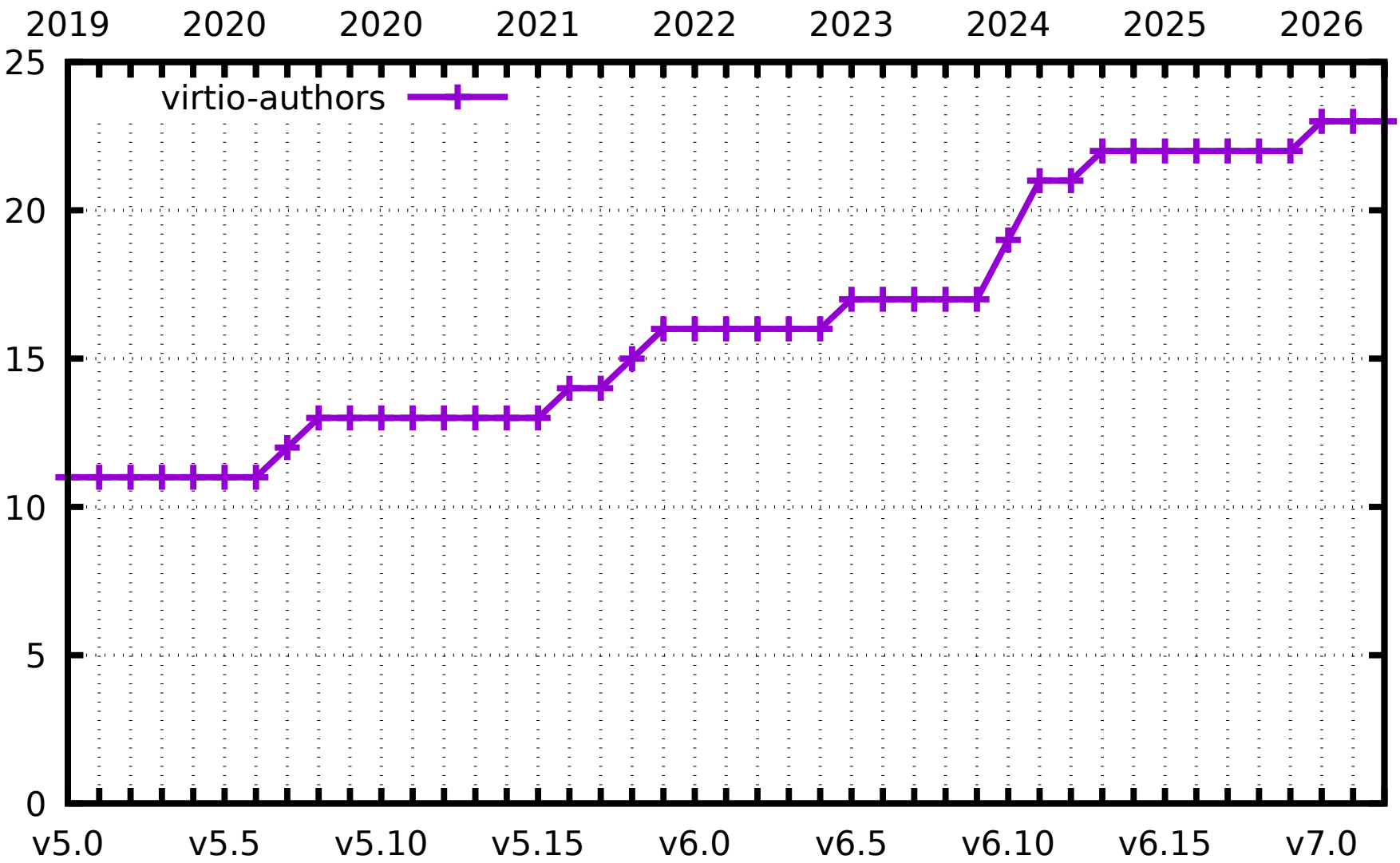
rpmsg (#authors >5 patches)



virtio (# core commits)



virtio (#authors >5 patches)



virtio-msg

- constant effort to get it upstream
- specification is harder than implementation

DeviceTree

System DeviceTree

- `lopper` as a System DT tool is under active development
- generic approach still under discussion

Firmware DT bindings

- did I mention Geert's talk already?

Off-the-Shelf?

Building Blocks

ideally

- one Kernel config
- one e.g. Zephyr config
- standard AMP tools in userspace

often like this

Testing SUNXI hwspinlocks

For testing this driver you need to build a modified crust firmware from my crust fork ...

There in the debug menu you need to enable the hwspinlock test loop.

After that you need to build an u-boot version with curst/SCP support. A u-boot fork with crust support in the crust branch is available here...

OpenAMP tries to unify

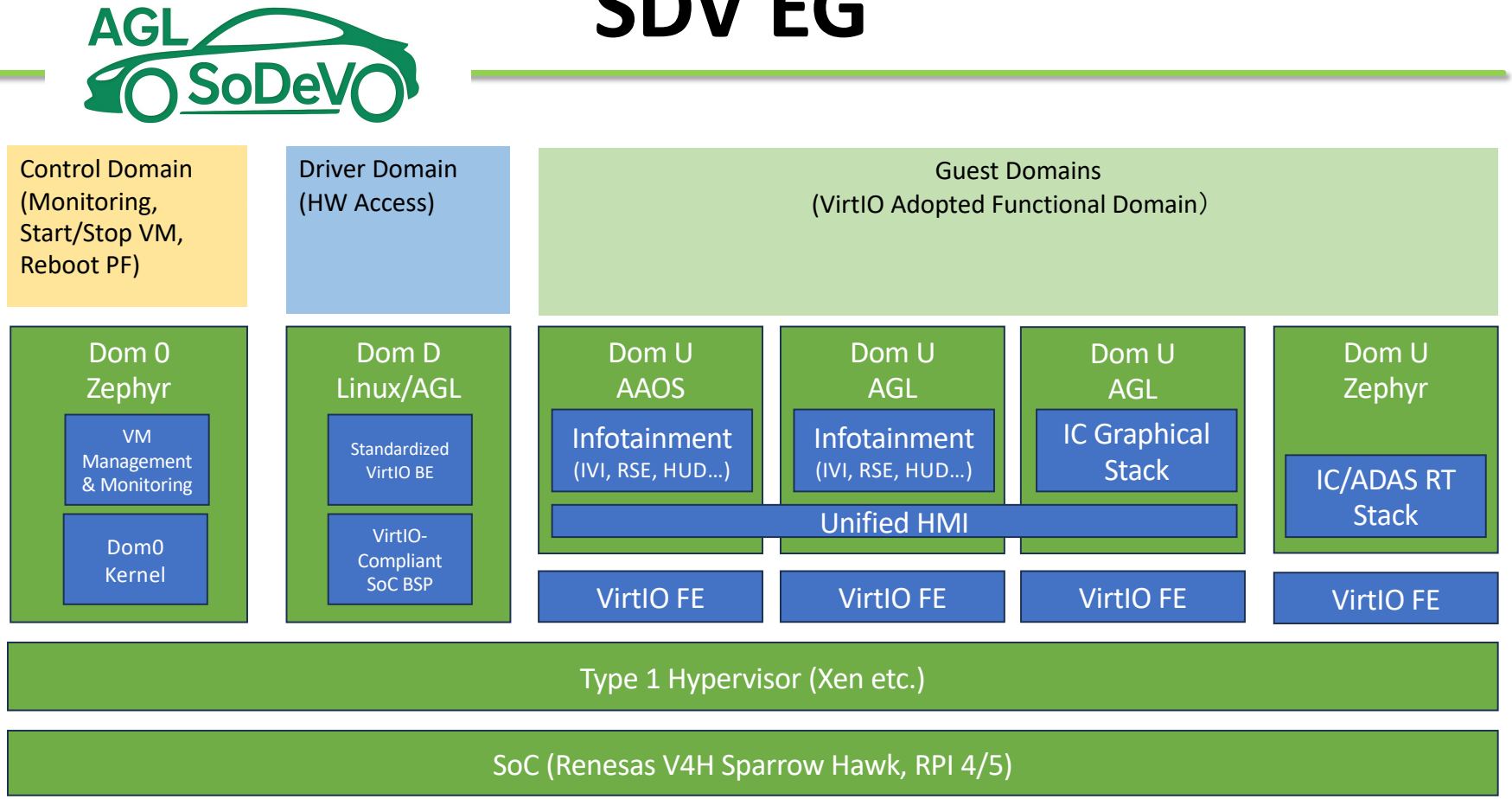
Examples

- Build instructions
- Test case

but it is still everywhere

- RPMsg example in Zephyr codebase

SDV EG


AGL SoDeVo
Control Domain (Monitoring, Start/Stop VM, Reboot PF)
Driver Domain (HW Access)
Guest Domains (VirtIO Adopted Functional Domain)
Dom 0 Zephyr
VM Management & Monitoring
Dom0 Kernel
Dom D Linux/AGL
Standardized VirtIO BE
VirtIO-Compliant SoC BSP
Dom U AAOS
Infotainment (IVI, RSE, HUD…)
Dom U AGL
Infotainment (IVI, RSE, HUD…)
Dom U AGL
IC Graphical Stack
Dom U Zephyr
IC/ADAS RT Stack
Unified HMI
VirtIO FE
VirtIO FE
VirtIO FE
VirtIO FE
Type 1 Hypervisor (Xen etc.)
SoC (Renesas V4H Sparrow Hawk, RPI 4/5)

Level 2 examples

Build instructions

- AGL SoDeV for SparrowHawk (WIP)
- AGL SoDeV for RPi4/5 (more WIP)
- currently hard to build
- ... but bleeding edge at home!

Conclusion

Building blocks

- are in general OK
- `hwspinlock` is not (but was an outlier)

  Not much used, but I will fix it nonetheless!
- stress-testing must be improved
- unification of non-Kernel parts would be great

Upper layers

- we can build the AMP buildings shown before
- upper layers are still in flux

  there be dragons!
- the tech needs to be more accessible

  amazing first steps have been done

Thank you!



The End

Questions? Comments?

Questions?

- Right here, right now
- At the conference
- wsa@kernel.org

  (the above two are better)