LINUTRONIX

Do Existing Drivers Conflict with PREEMPT_RT? A 3-year Survey

Embedded Open Source Summit
April 16-18, 2024

Real-Time Linux Summit

Ahmed S. Darwish <darwi@linutronix.de>

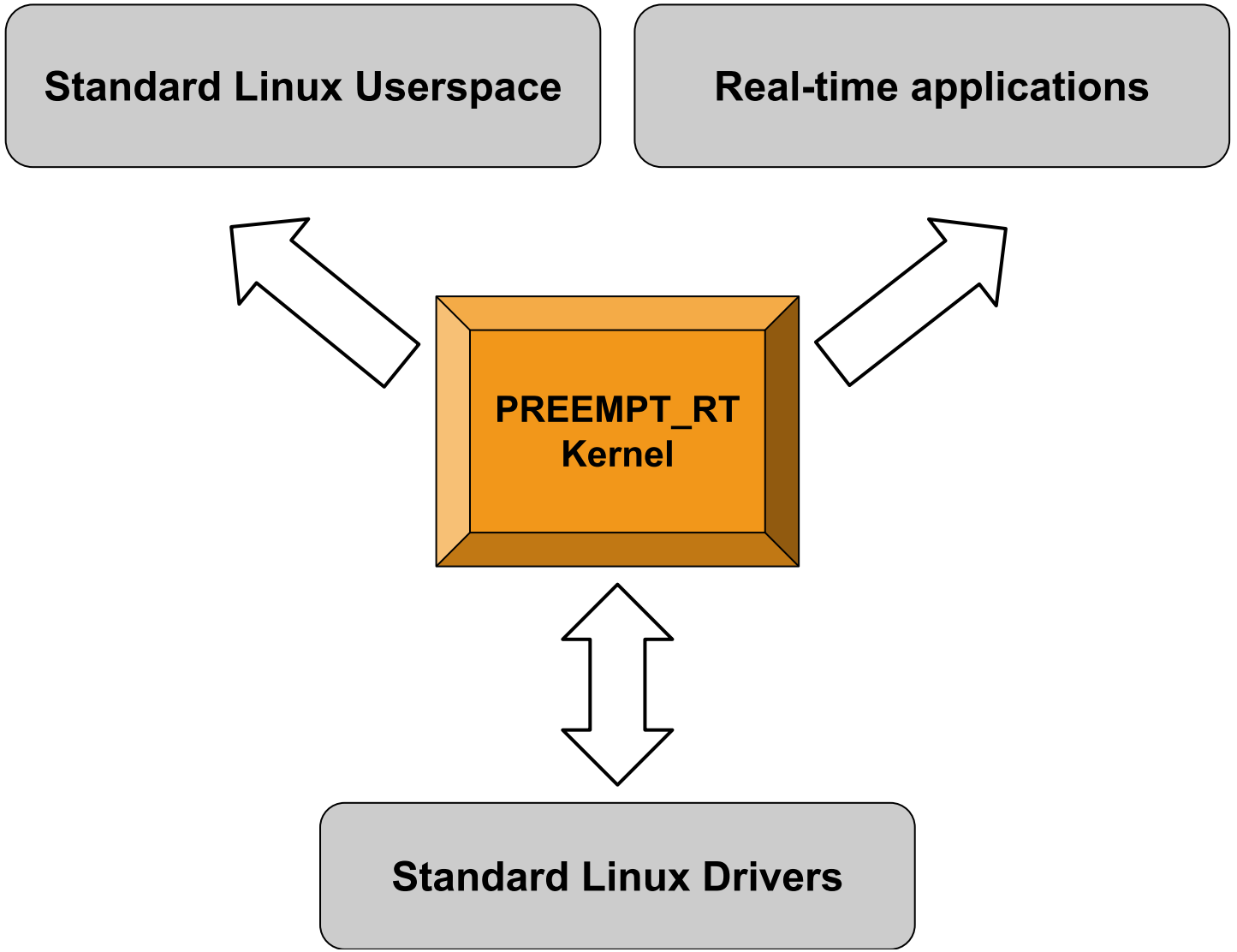
Do Existing Drivers Conflict with PREEMPT_RT? A 3-year Survey

Preliminaries

PREEMPT_RT: Semantics

- **“PREEMPT_RT”:** The Realtime-Linux kernel releases
  - Linux Kernel + “patches”
- “**PREEMPT_RT**”: Preemption model
  - At the Realtime-linux kernel releases
  - When PREEMPT_RT patches are fully merged mainline

PREEMPT_RT: Conceptual Summary



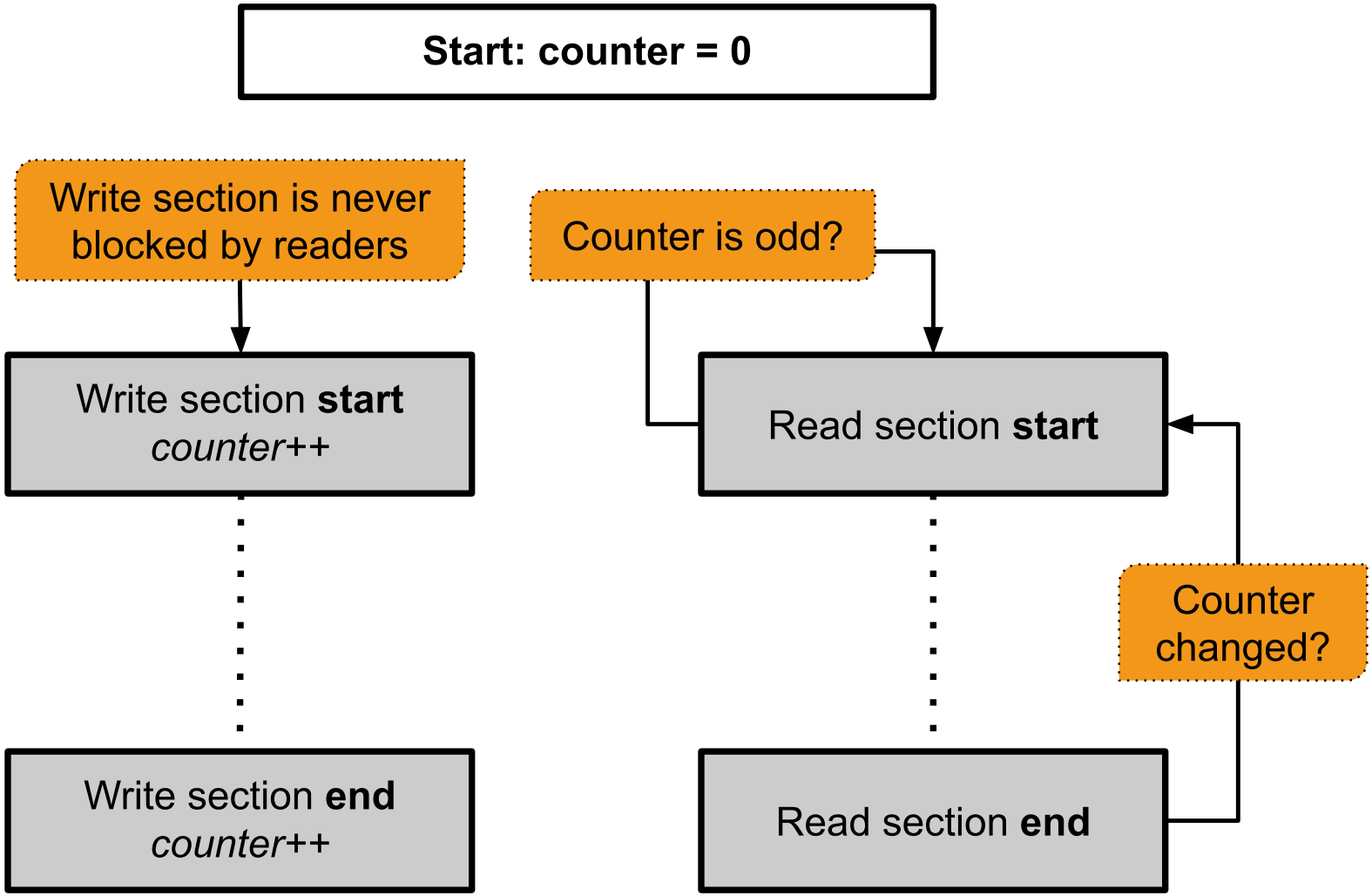
Do Existing Drivers Conflict with PREEMPT_RT? A 3-year Survey

A Survey

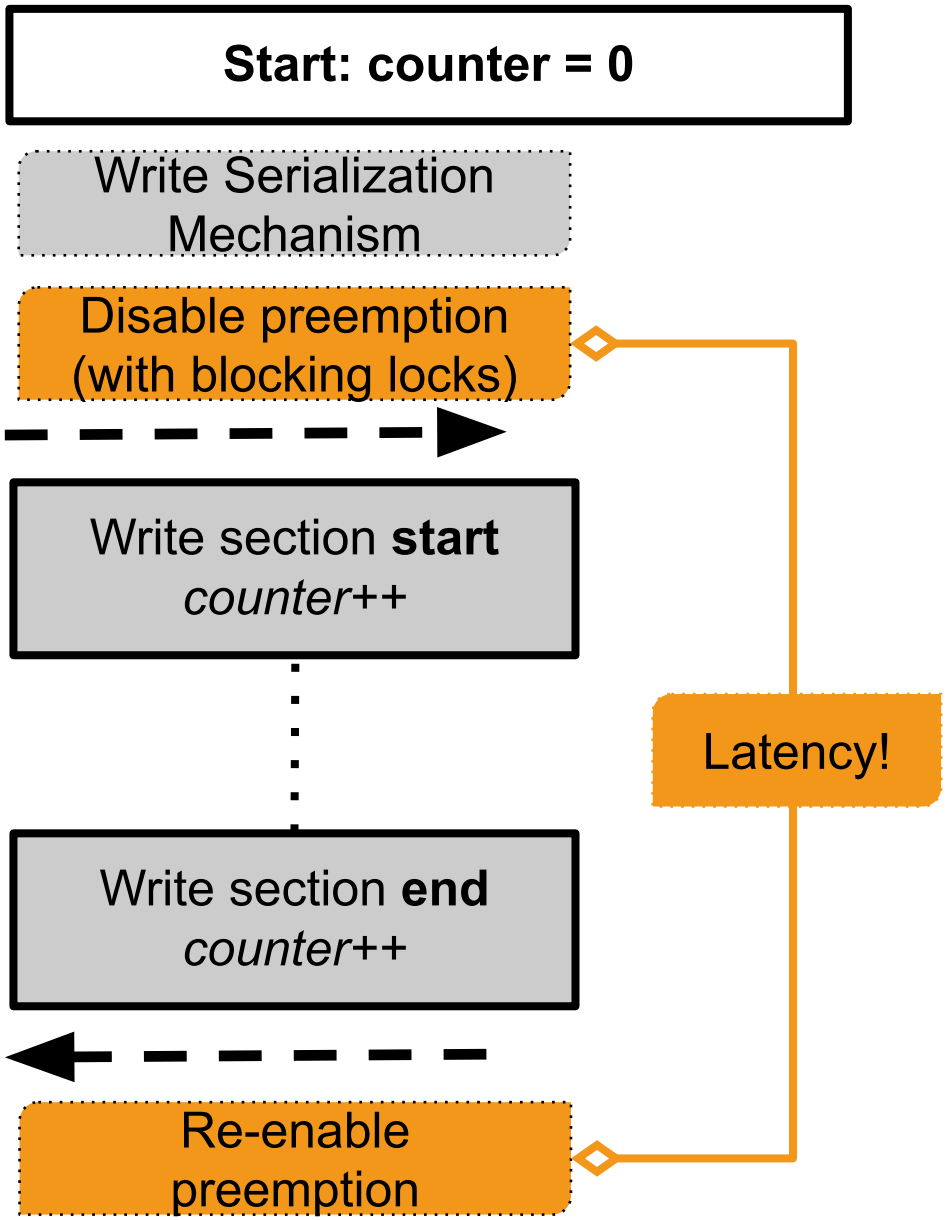
3 Years

- Sequence Counters
- Tasklet API challenges
- Context-aware code
- DRM drivers / TTY consoles

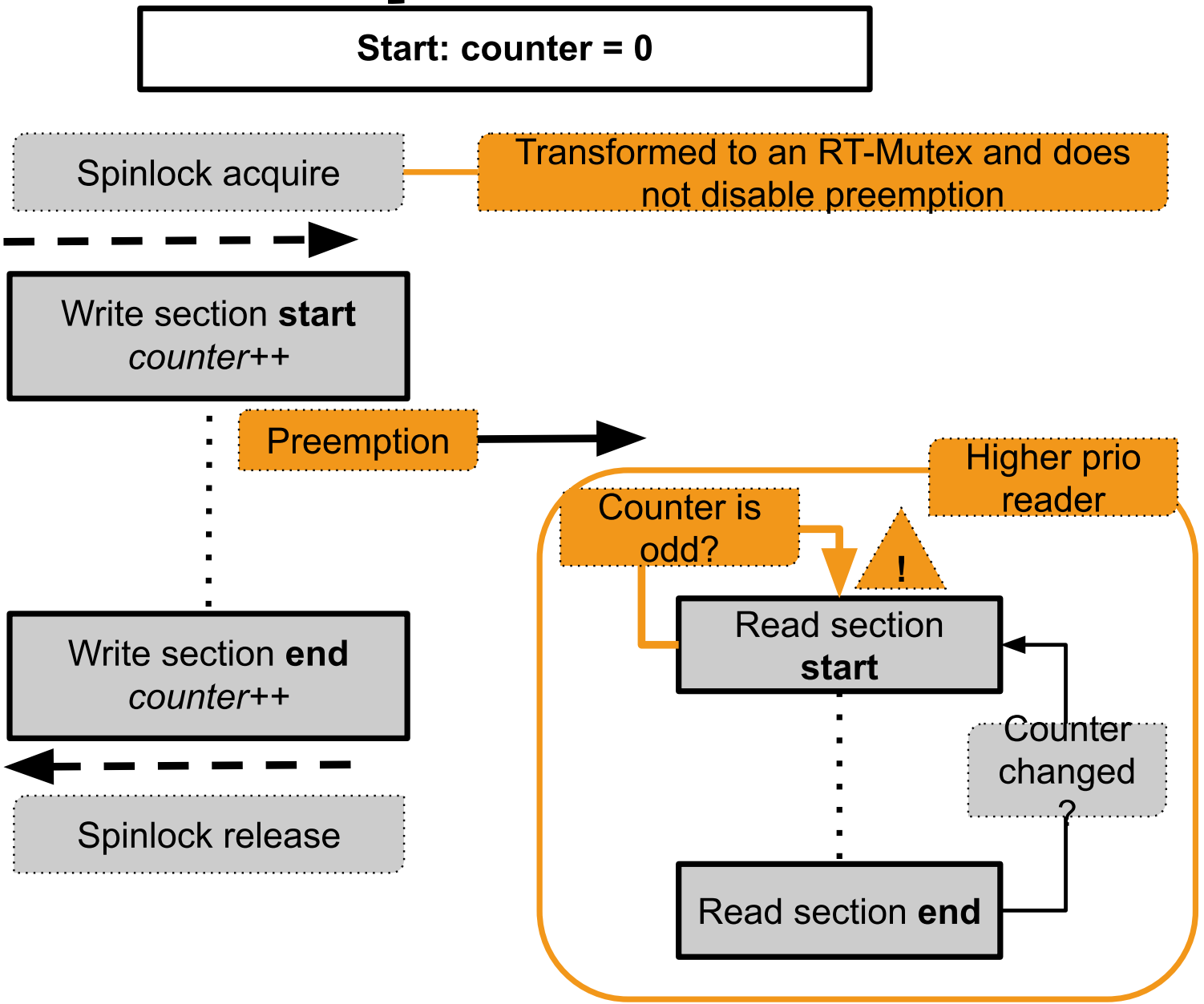
Sequence Counter APIs



Sequence Counters

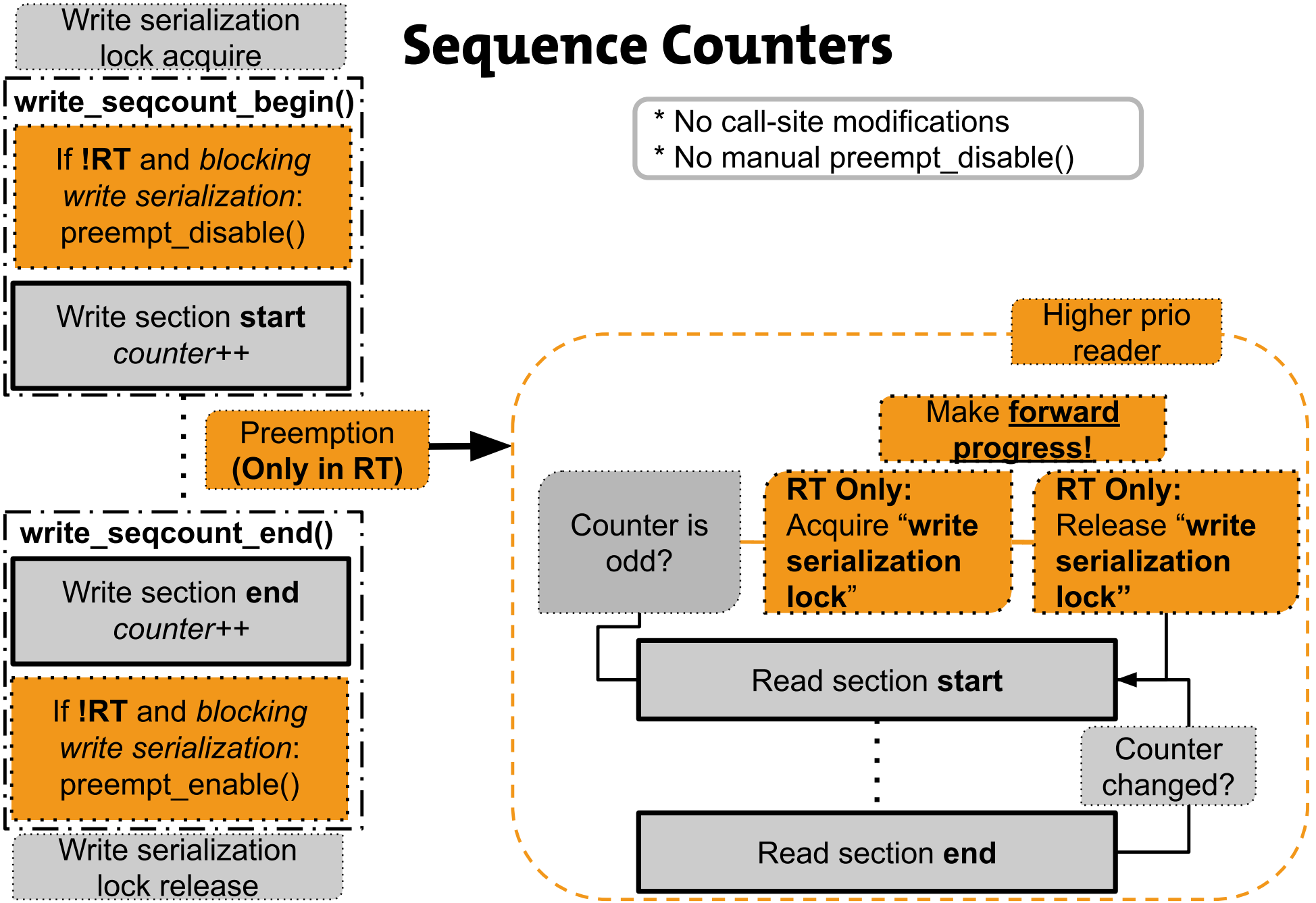


Sequence Counters



Sequence Counters

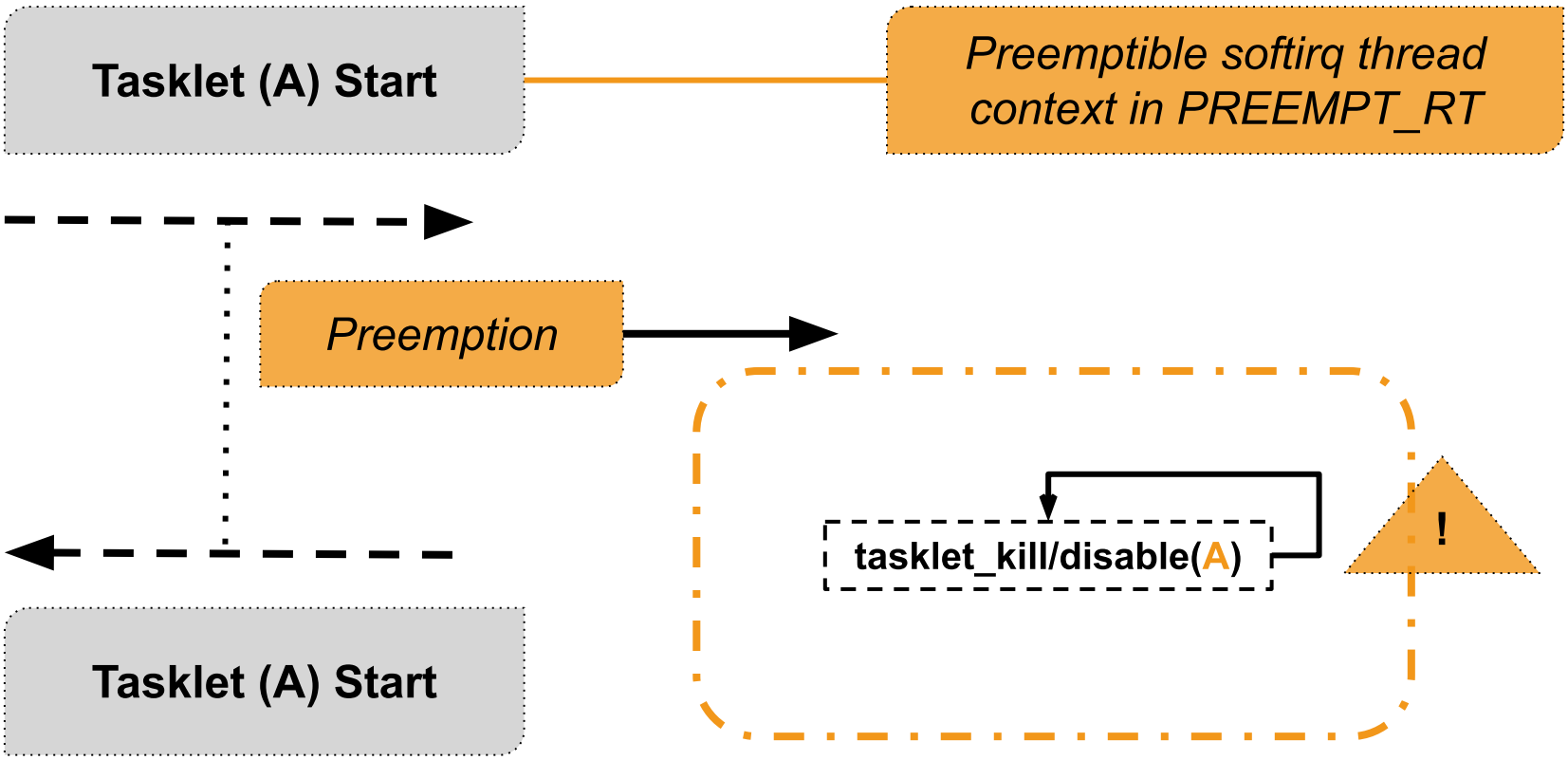
* No call-site modifications
* No manual preempt_disable()



Sequence Counter APIs

- Sequence Counter and Sequental Lock APIs with attached write sequential lock
  - More safety / Lockdep-awareness
- Latch Sequence Counters
  - Used in multiple call sites afterwards, including printk

Tasklet APIs



Tasklet APIs

- We surveyed 400+ call-sites for context preemptibility (Google Sheet link.)
- All except 10 call sites were calling tasklet deletion from preemptible contexts, so the initial busy-loop tasklet deletion was substituted with conditional variables.
- The other call sites were buggy and subsequently fixed.
- Work has been long in-progress to remove tasklets support from the kernel.

Contextual Checks

- The macros:
  - preemptible()
  - in_irq(), in_softirq(), in_serving_softirq()
  - Began creeping into drivers and driver subsystems
- Changes in SCSI libsas, USB, etc.

Questions / Comments

Thank you for your attention.

darwi@linutronix.de
info@linutronix.de