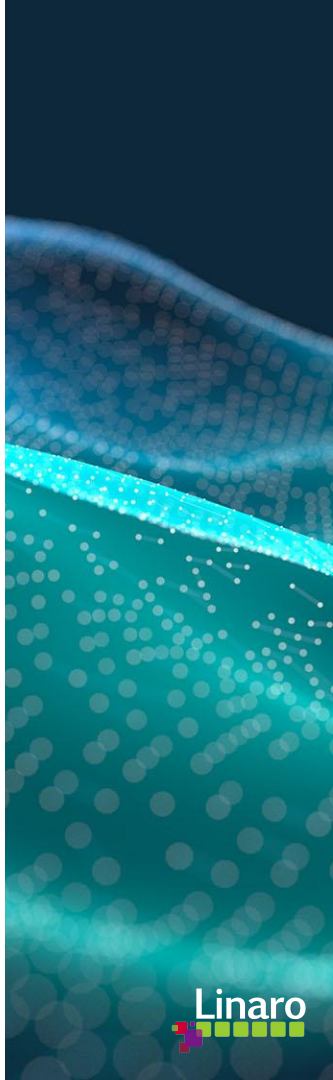
BoF:
Multiple system-wide low power-states (S2I/S2R)

Ulf Hansson, Linaro
ulf.hansson@linaro.org

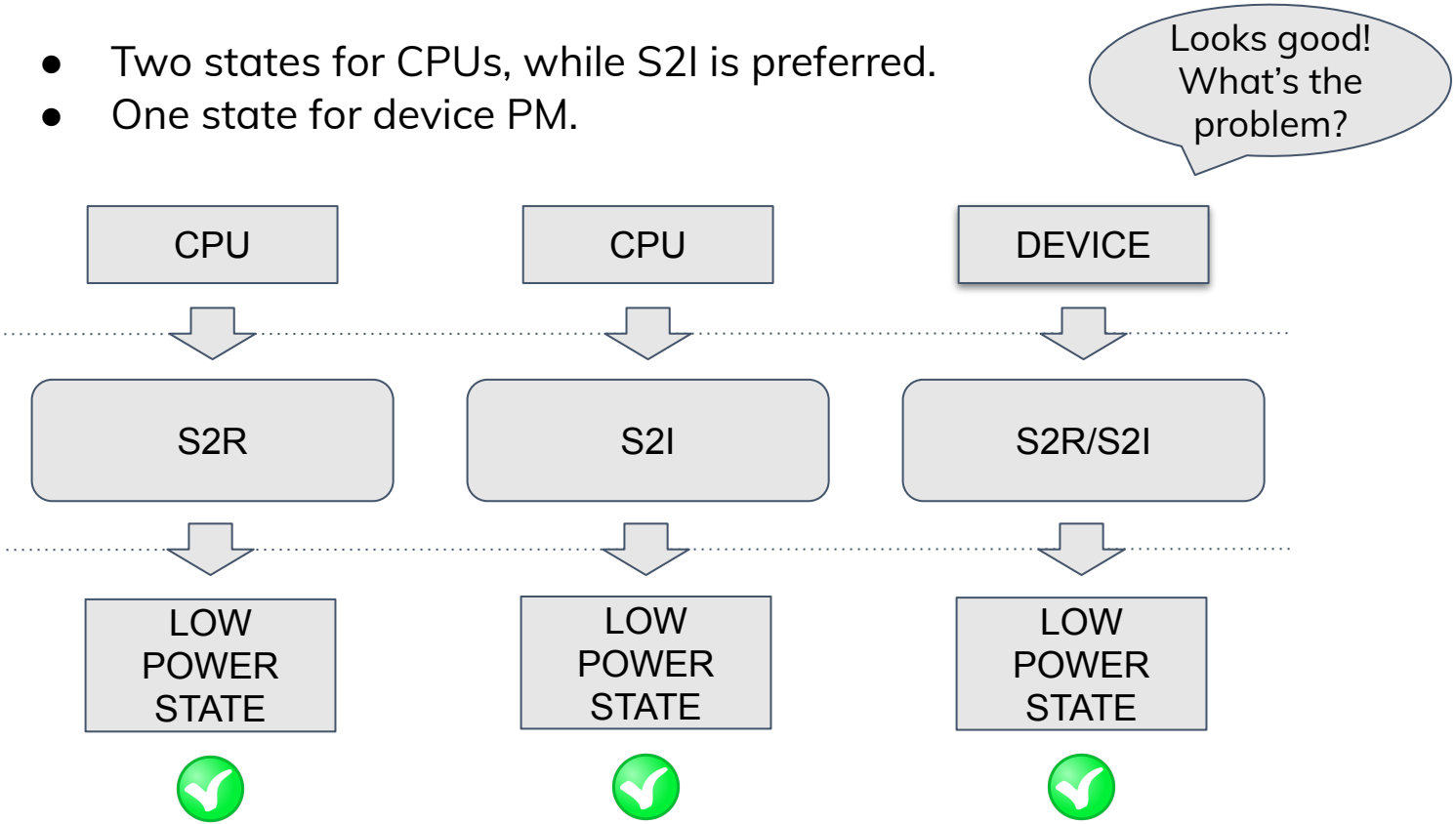
Agenda

- Background
- Use-cases - discussions
- What’s the way forward?



Background

- Two states for CPUs, while S2I is preferred.
- One state for device PM.



#1 PM co-processor

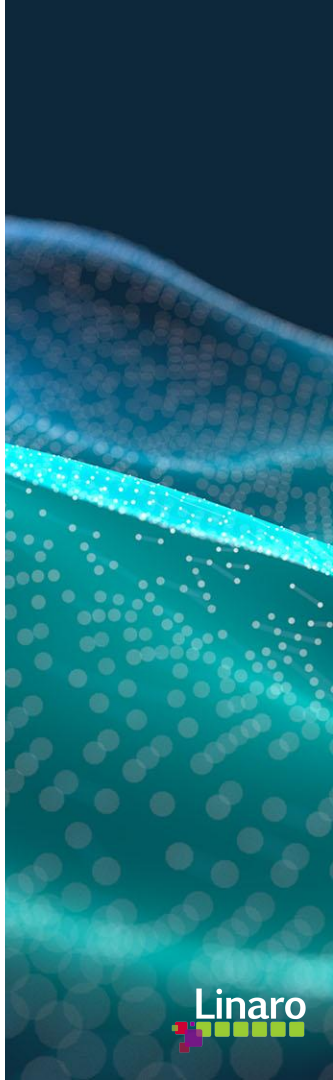
- Controls power-domain(s) for peripheral devices.
- Manages configurations for system wakeups.
- Multiple low power-states are supported.
- Modelled as a generic PM domain (genpd provider).

Problem:

- Entering the deepest state can break the ***system wakeup latency constraint*** for a running use-case.

Thoughts:

- Do we need a new genpd governor?
- What corresponds to the **system wakeup latency**?
- Does **cpu_dma_latency** correspond to the constraint?



#2 CPUs, S2I and PREEMT_RT

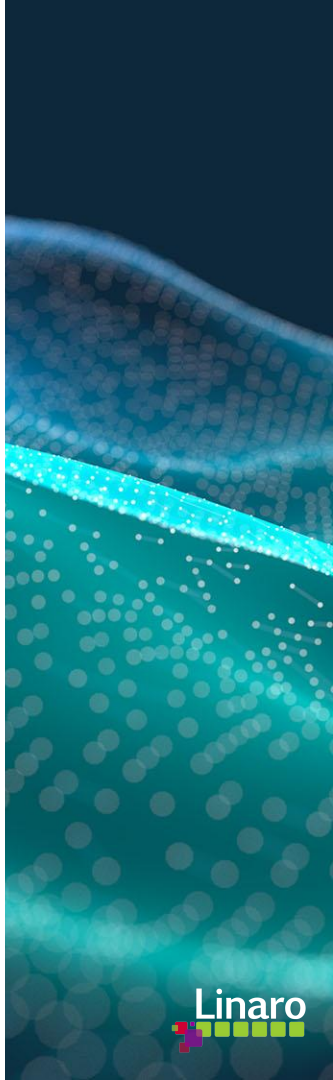
- Multiple low power-states are supported.
- Deeper states are disabled for CPUIdle.

Problems:

- Deeper states would work for S2I, to not waste power.
- To enable deeper states, the ***system wakeup latency constraint*** must be guaranteed, for the running use-case.

Thoughts:

- Do we need a governor for S2I?
- What corresponds to the **system wakeup latency**?
- Does **cpu_dma_latency** correspond to the constraint?



#3 NVMe - storage

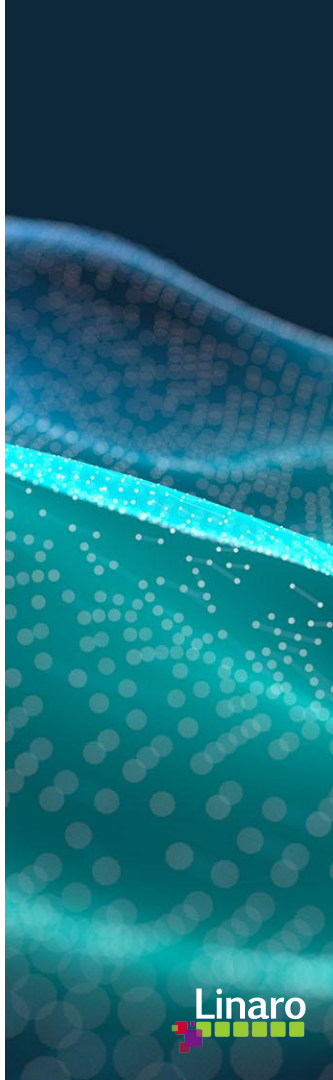
- Autonomous Power State Transition (APST) and powered-off.
- APST is used when applicable.

Problems:

- APST or not - NVMe may consume power.
- Power-off instead of APST - but when?
- Frequently doing power-off/on could hurt durability.

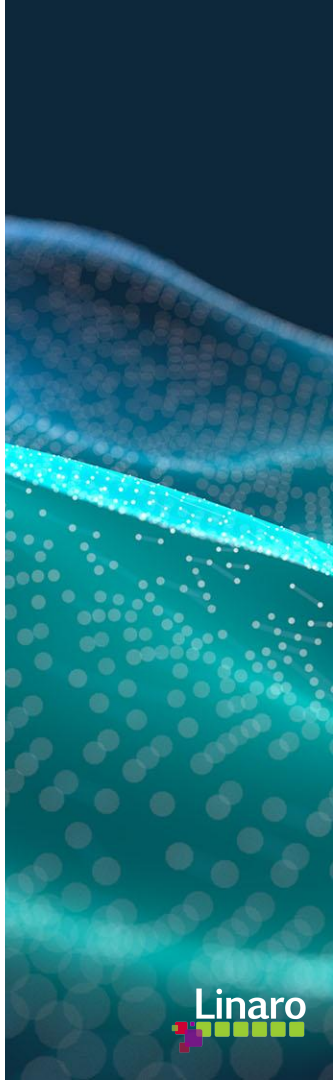
Thoughts:

- Should user-space decide based on the use-case?
- Device specific or generic (**sleep duration**)?
- Similar problem exists for eMMC and SD.

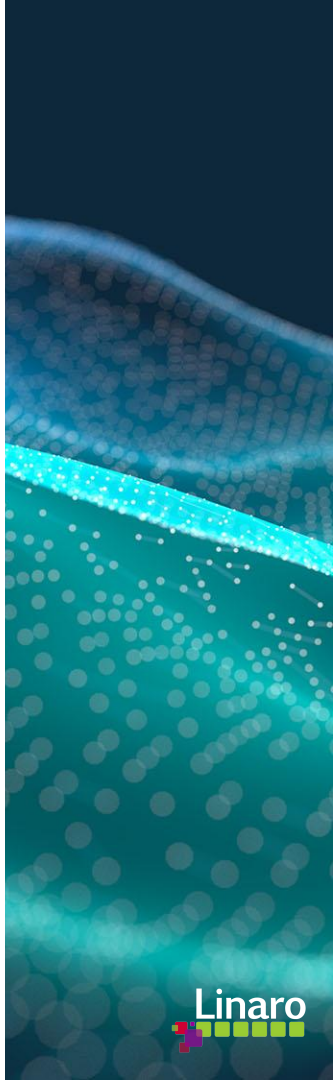


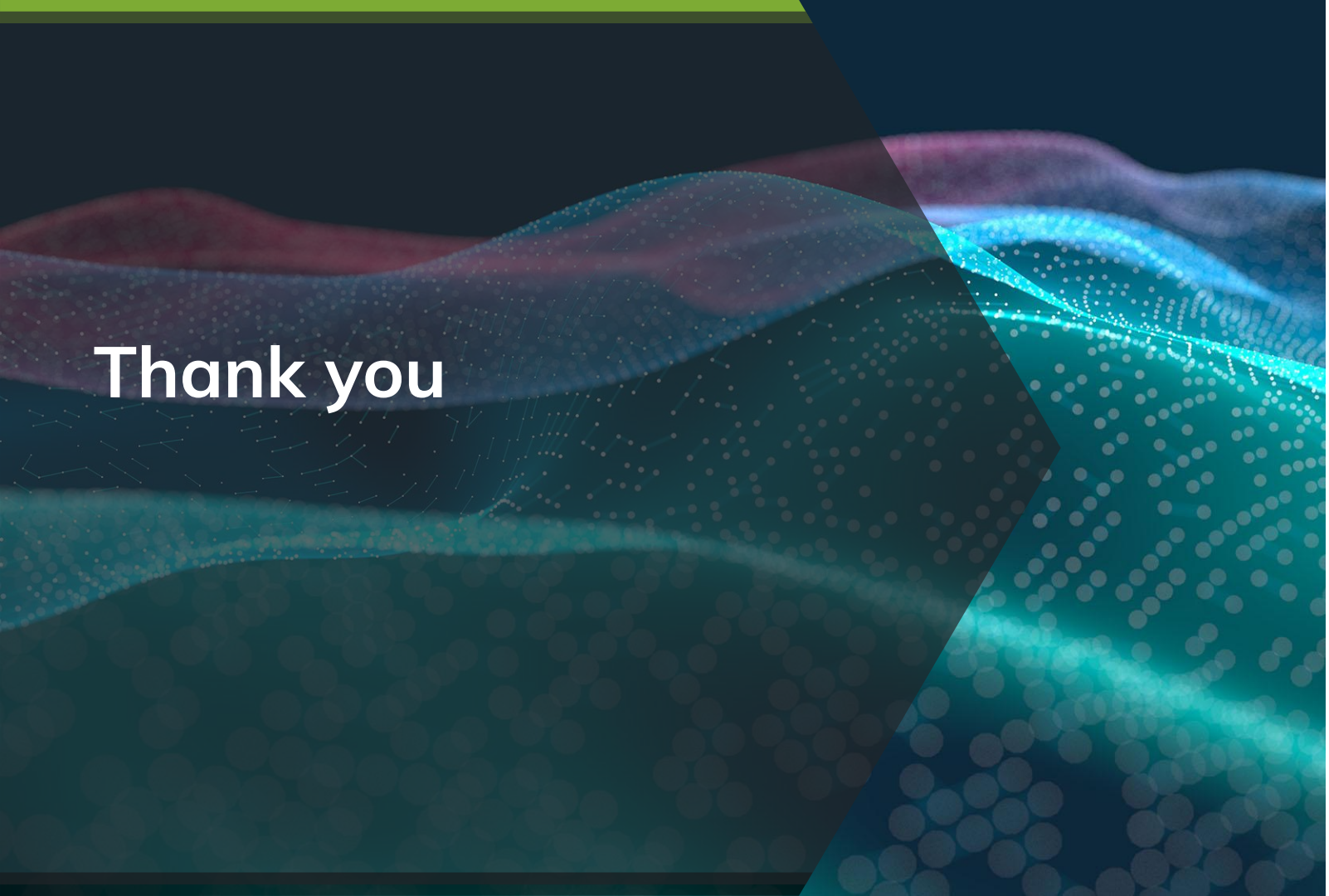
#4 More use-cases?

-



What’s the way forward?





Linaro