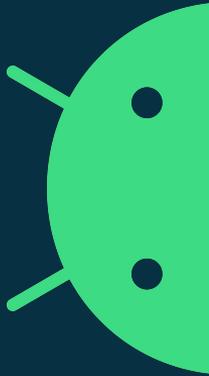# Solving issues associated with modules and supplier-consumer dependencies

Saravana Kannan

android

Some base terms

Driver

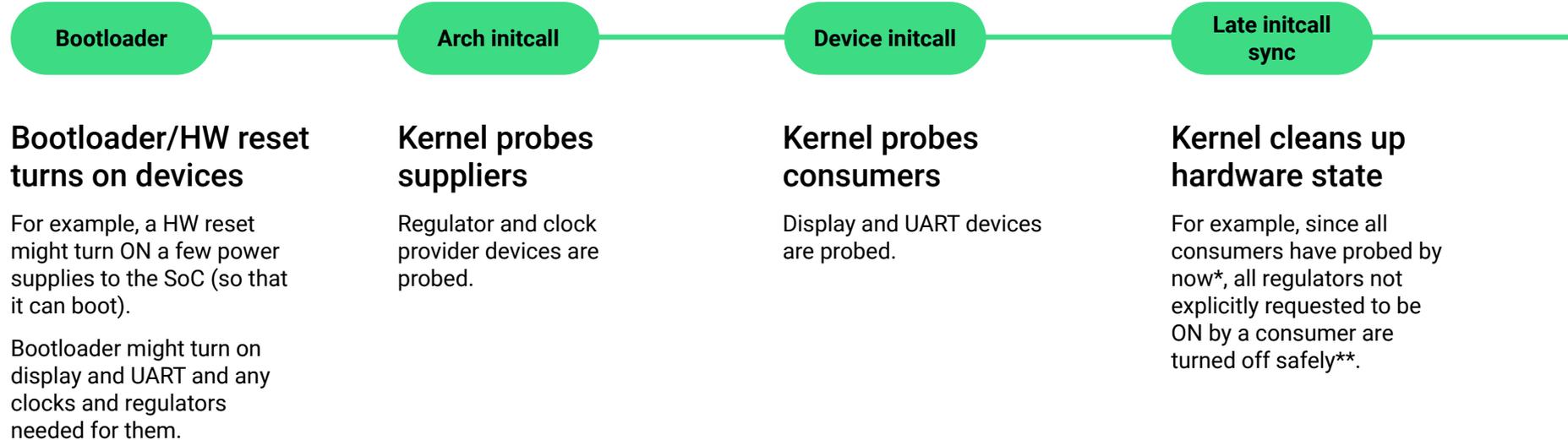A driver is a piece of software that knows how to operate a specific hardware IP.

Device

A device is an instance of a hardware IP and there can be more than one.

Device Probe

A device is probed by a driver. The probe succeeds if the driver knows how to operate the device and gets all the resources it needs.

android

Simplified boot sequence

Bootloader — **Arch initcall** — **Device initcall** — **Late initcall sync**

Bootloader/HW reset turns on devices

For example, a HW reset might turn ON a few power supplies to the SoC (so that it can boot).

Bootloader might turn on display and UART and any clocks and regulators needed for them.

Kernel probes suppliers

Regulator and clock provider devices are probed.

Kernel probes consumers

Display and UART devices are probed.

Kernel cleans up hardware state

For example, since all consumers have probed by now*, all regulators not explicitly requested to be ON by a consumer are turned off safely**.

android

* Not really!
** System crashes or misbehaves :(

android

Delicate dependencies

- Initcall link order is used to manually order device probe.

- But not really, initcalls only order driver registration.

- Falls apart if, same driver probes two devices with different ordering needs.

- Deferred probe throws initcall ordering out the window.

- Completely breaks down for modules.

android

One kernel to boot them all

Generic Kernel Image

One generic kernel that works across all ARM64 based systems

Modules for hardware

SoC vendors/OEMs can supply modules for their hardware.

android

Device links to the rescue!

android

Well, almost

- Upstream API/functionality in driver core

- Can track supplier-consumer dependencies

- Can order device probes based on dependencies

- Keeps track of supplier and consumer device state (successfully probed or not)

android

Missing pieces

- Doesn't have a mechanism to let the supplier know when all its consumers have probed

- Needs some other entity to add the device links first

- No way to know when all the consumers have been linked to a specific supplier

android

Solving the puzzle

Update device links to add callback

Device links already tracks consumer state. So simply add a driver/bus callback for when all the consumers are probed and the hardware state can be synced to the software state -- sync_state()

Get dependency info from firmware

Firmware such as devicetree and ACPI already have the dependency information. Just let them add the device links.

android

Why not have driver create device links?

- No way of knowing when device links to all the consumers have been added. Maybe the module hasn't been loaded?

- No way to make sure all consumers get a chance to add a link to the supplier before the supplier check for "all consumers have probed".

android

Firmware adding device links

○   Driver being loaded as a module becomes irrelevant to the problem

○   We can know when device links for all consumers have been added.

○   Easy to make sure consumers get a chance to add a link to the supplier before the supplier check for "all consumers have probed".

android

Patches?

android

Here it is!

- https://lore.kernel.org/lkml/20190904211126.47518-1-saravanak@google.com/

- More subtle details in the commit text/documentation.

android

Questions?

android