On this Rock I will build my System

Why Open-Source Firmware matters

Lucas Stach – l.stach@pengutronix.de
About me

- Member of Pengutronix graphics and kernel team
- Low-level infrastructure guy
- Working on projects with long maintenance times
Where do we come from?

- Minimalistic Firmware
  - Even got used to call them „bootloaders“

- Usually no run-time interaction with kernel
Old model – the good

- Full control over system operation in Linux kernel
  - Helps when debugging complex interactions
- Easy update story
Old model – the bad

- High complexity in Linux kernel
  - As bad as it sounds? Likely not...

- Some features hard to implement in a generic kernel

- Things start to look different when virtualization is added
Enter PSCI

Power State Coordination Interface

- Moves system global and CPU power state into abstract interface
  - Trap based (like syscall)

- Bare-metal and virtualized kernel see same thing
Trusted Firmware

- Help vendors in implementing PSCI and other basic platform functions in a standard compliant way

- Started by ARM, now a Linaro project

- BSD licensed
# PSCI – collision with reality

Real-world Hardware struggles: shared Registers

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Device1</td>
<td>Device0</td>
<td>CPU1</td>
<td>CPU0</td>
</tr>
<tr>
<td>power control</td>
<td>power control</td>
<td>power control</td>
<td>power control</td>
</tr>
</tbody>
</table>
PSCI – collision with reality

Real-world Hardware struggles: shared Interfaces

SoC

PMIC

VDDx

VDDy

I2C
PSCI – collision with reality

- Lots of hardware not designed to provide the separation required by PSCI
- May not even be possible entirely
Down the rabbit hole: SCMI

System Control and Management Interface

- Move lots more functionality to SCMI provider
  - Device power/performance states
  - Clocks
  - Sensors
  - System control
Down the rabbit hole: SCMI

- Linux kernel implementation gets easy

- Firmware get much more complex
  - Lots of runtime interactions between OS and FW
  - Sometimes hard to reason about
Can I decide to just don’t care?
Maybe...
Can I decide to just don’t care?
Probably not!
Asymmetric SoCs

Application Processor

Minion Core

RT Core

Shared System Resources
Asymmetric SoCs
Takeaways

- Firmware taking over more functionality is here to stay

- Incentive for chip vendors to open-source not as high as with Linux kernel

- Real risk of loosing some control