OMAP Thermal Software

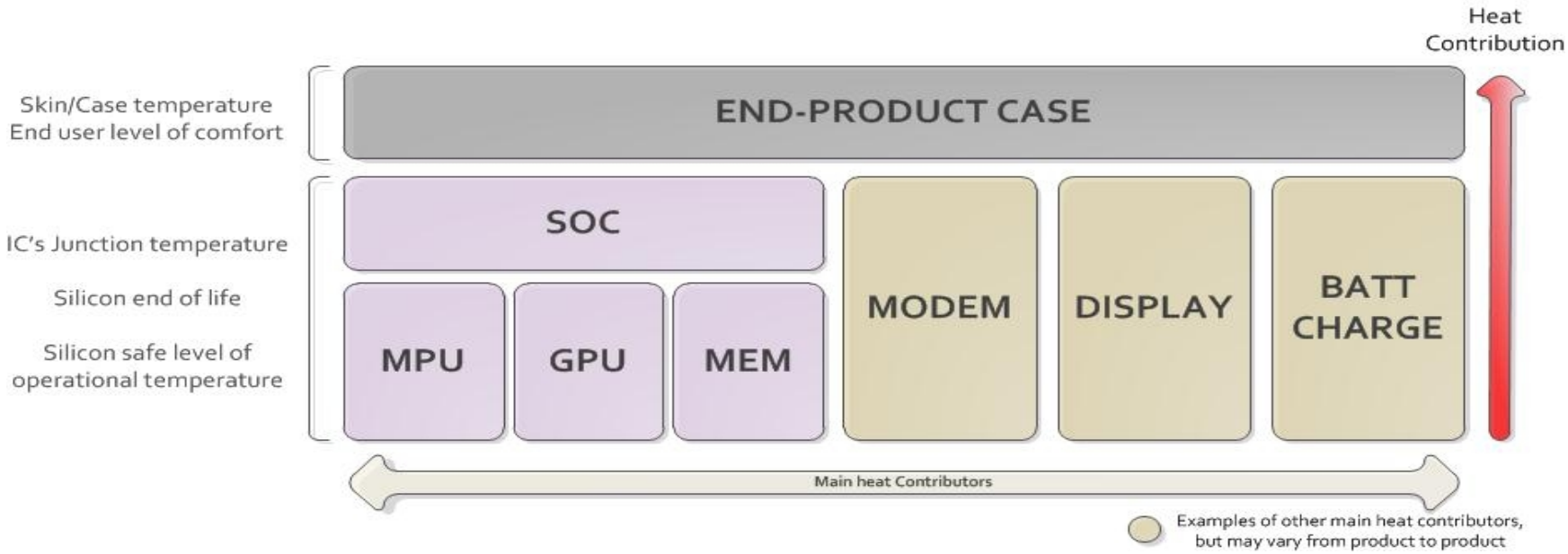
August 30th, 2012

Eduardo Valentin

Content

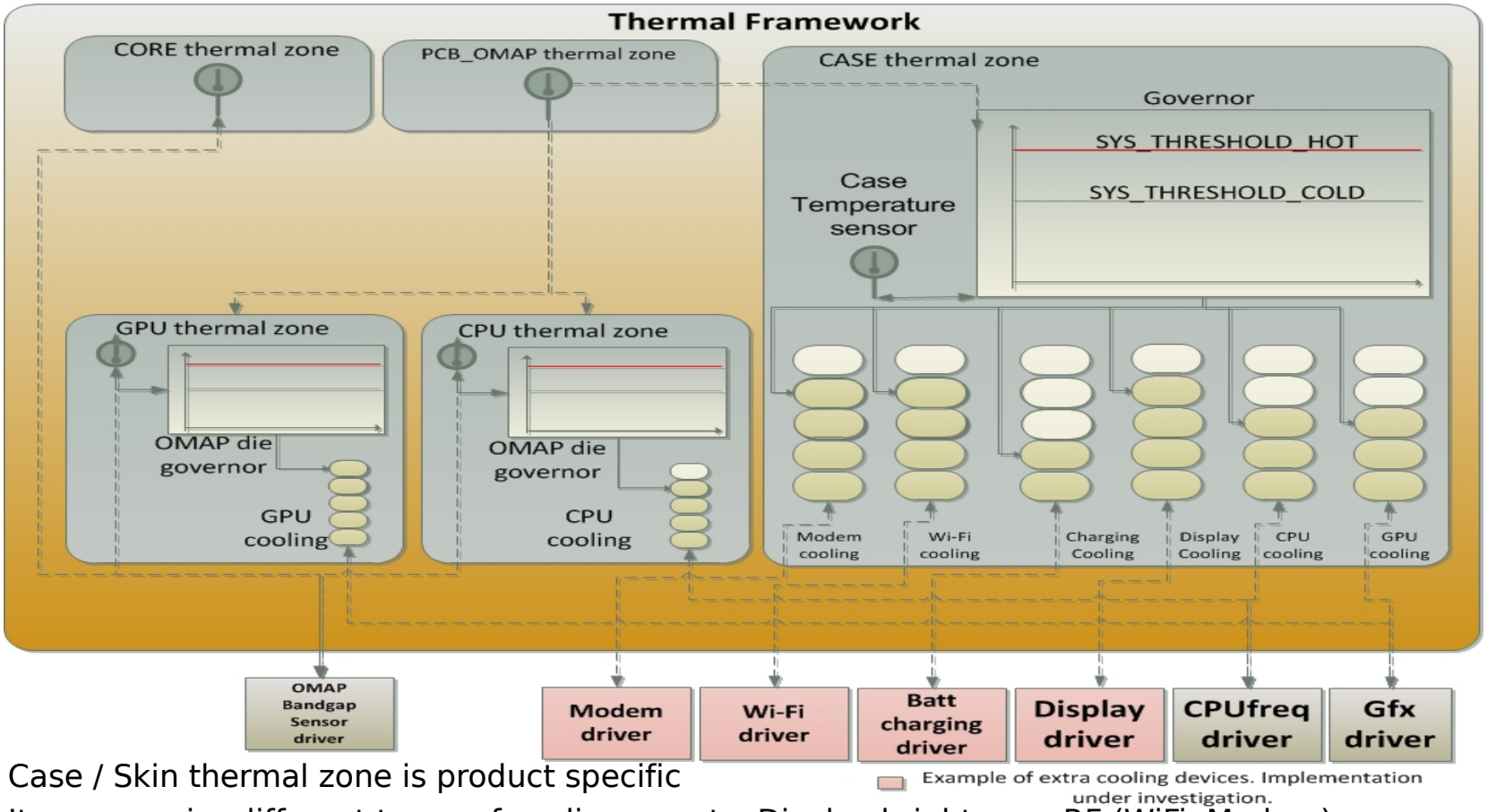
- Software design decisions and constraint overview
  - Case / Skin temperature management
  - Junction temperature management
  - Linux Kernel Thermal FW overview
- Not well solved Use Cases, scenarios
  - Passive CPU cooling, what can be done today
  - UC I: CPUfreq policy update
  - UC II: What if need to re-use CPU cooling in different zones?
  - UC III: CPUfreq governor vs. boost interfaces
  - UC IV: How about combo power rails?
  - UC V: Removal of high performance states
- API changes proposals: open discussion

Overall design principles



- Setup of policies that manages major heat sources for skin/case temperature management
- Software needs to take care also of the IPs junction temperature management and the constraints that they bring
- The constraint management has to be aware of the list of constraints generated by different policies, per device.

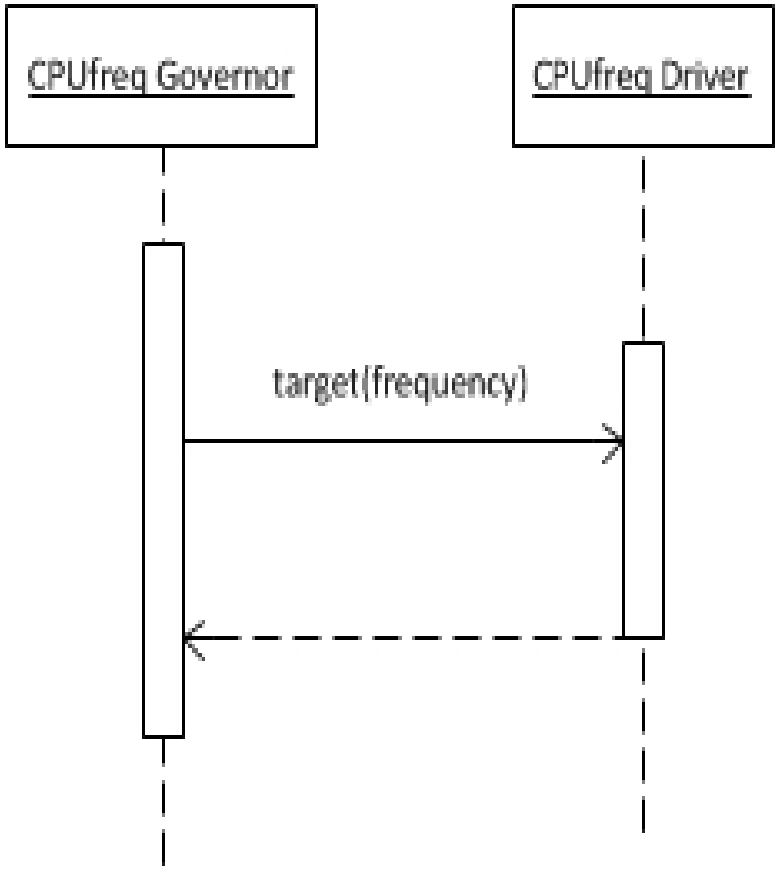
Thermal constraints and SW



- Case / Skin thermal zone is product specific
- It may require different types of cooling agents: Display brightness, RF (WiFi, Modem) throughput control, CPU and GPU DVFS, etc
- Policy definition needs to be followed up by a characterization work
- Junction temperature management policy makes sure temperature is within silicon's operating limits

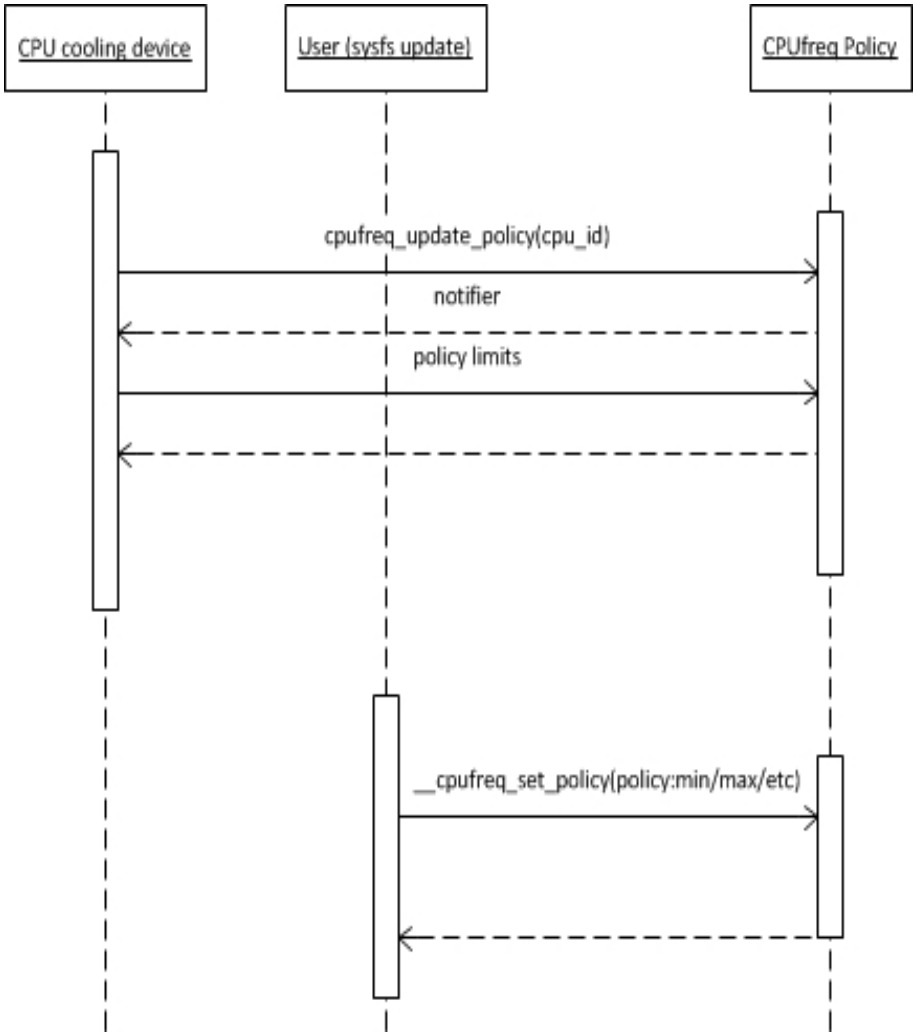
Passive CPU cooling, what can be done today

- Normal CPUfreq operation is described on the right side.
  - There is a policy descriptor to hold CPU frequency constraints (min/max, frequencies available, etc)
- Two ways to update the policy constraints:
  - In kernel:
    `cpufreq_update_policy(cpu_id)`
    `notifier:`
    `CPUFREQ_POLICY_NOTIFIER`
  - Userland: sysfs nodes under /sys/devices/system/cpu/cpu*/cpufreq/*
- CPU cooling is done by means of updating the CPUfreq policy
  - from kernel space
  - based on cpu thermal zone definitions.



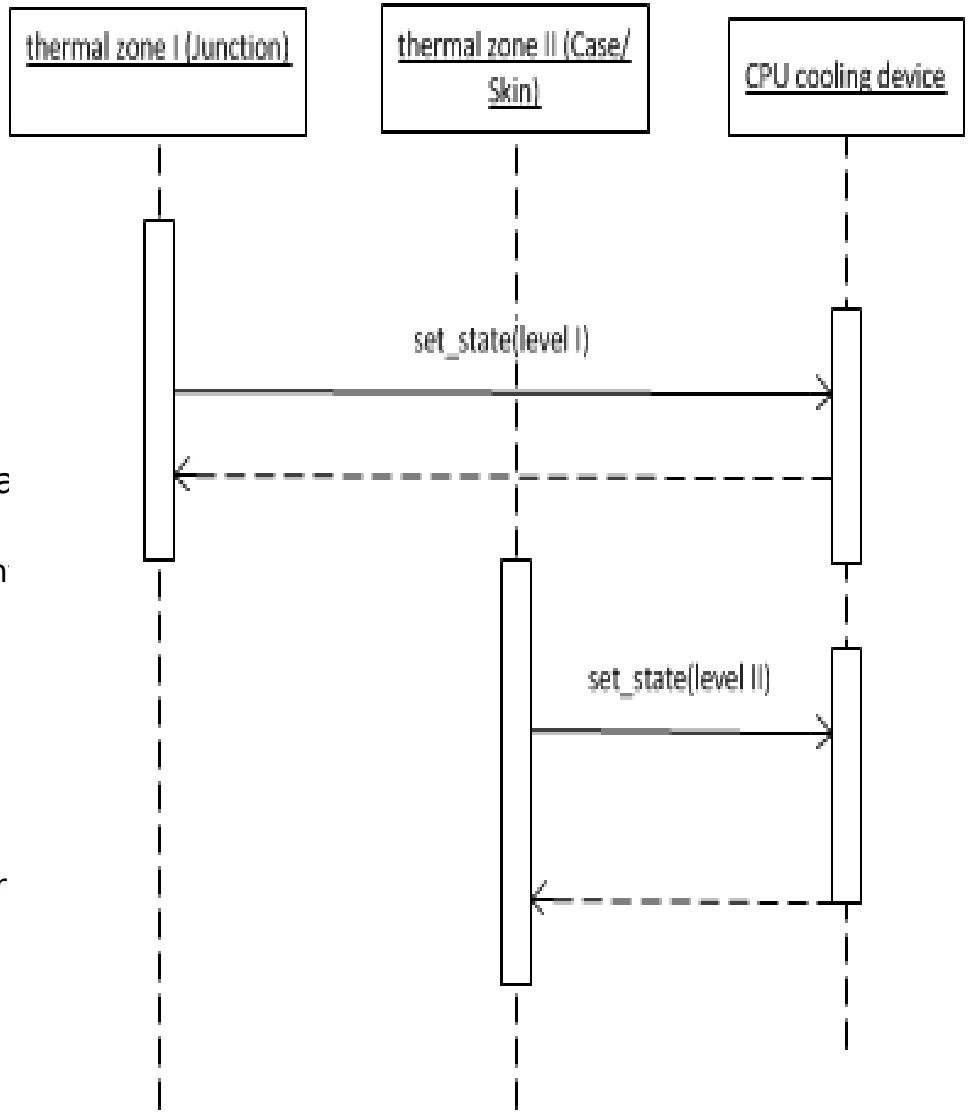
UC I: CPUfreq policy update

- What is the current synchronization between limits set by user (sysfs) and in kernel constraints on CPU frequency?
- Scenario:
  - A. System is under heavy CPU load:
    - CPU thermal gets into critical thresholds
  - B. Thermal zone constraints are applied within kernel
  - C. Malware, or even not well written applications may abuse the sysfs interfaces
    - Busy loop requesting for maximum frequency
  - Ideally thermal constraint must be the one selected in this scenario.
  - How do we handle such situation today?
- Highlight: other types of devices may also suffer of the same issue: constraint coming from userspace + thermal limitations
- **Requirement:** Device constraint synchronization between userspace and kernel space



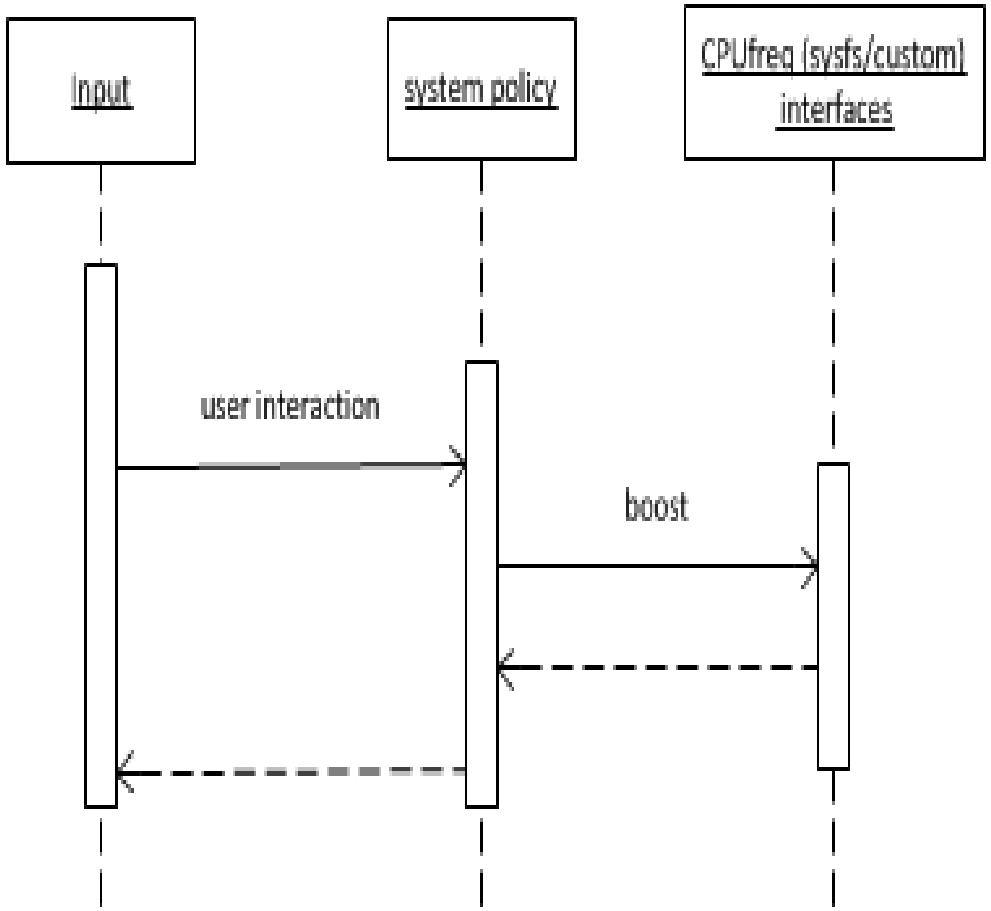
UC II: What if need to re-use CPU cooling in different zones?

- There are cases in which makes sense to receive constraints from different thermal zones:
  - Maintain CPU within functional operating temperature
  - Sustain product case/skin temperature by reducing power produced by CPU (passive cooling)
- In such cases, each constraint coming from different thermal zones must have a life time:
  - Possibility to add and remove the constrain
  - It doesn't make sense to keep last applied
  - Arbitration based on priority definition is needed
- Again, the constraint coming from different parts of the system must be solved in the same way:
  - UC I: we must consider also what the user says
- **Requirement**: Device constraint management in a centralized layer.



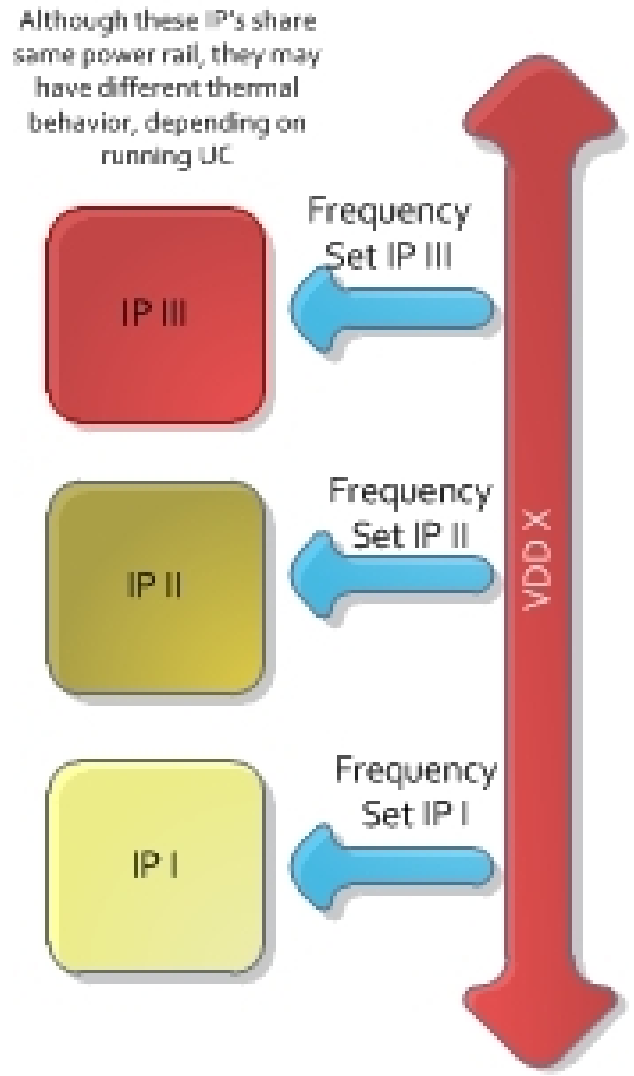
UC III: CPUfreq governor vs. boost interfaces

- Sometimes you may want to boost CPU frequency, targeting performance, interactiveness
- Constraint set coming from user space, may be similar situation as in UC I
- There are some proposals to allow device drivers to request also high frequencies on CPU:
  - Interrupt handling overhead
- Case: What to do if someone (userspace or kernel space) requests for boost frequency when the CPU thermal zone is at critical thresholds?
- **Requirement**: Device constraint synchronization between userspace and kernel space.



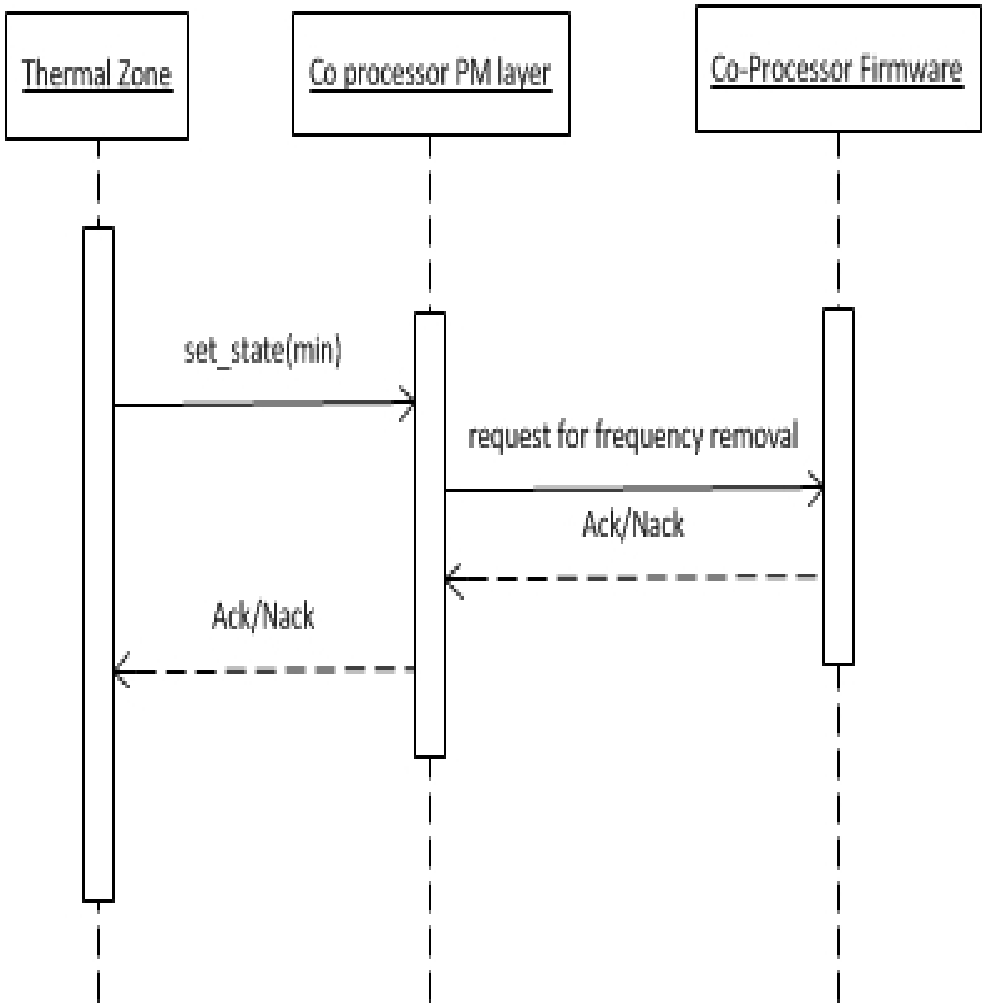
UC IV: How about combo power rails?

- Cooling devices based on frequency scaling are usually bound to voltage scaling as well.
- Frequency + Voltage scaling is more effective than frequency only scaling, from cooling aspects
- Devices may have their own frequency domain and frequency set
- Devices may also share same power rail
  - All devices need to downscale frequency in order to downscale voltage at the voltage domain
- **Requirement**: 1 Constraint to N Device constraints representation.



UC V: Removal of high performance states

- Case: Multimedia co-processor is activated with high frequency due to use case requirement, but due to thermal constraint, high frequency may be not available anymore
  - This can also overlap with UC IV, what if we need to remove high frequency due to thermal constraint from IP sharing same power rail
- **Requirement**: Handshaking/Notification when high performance states are enforced to be removed



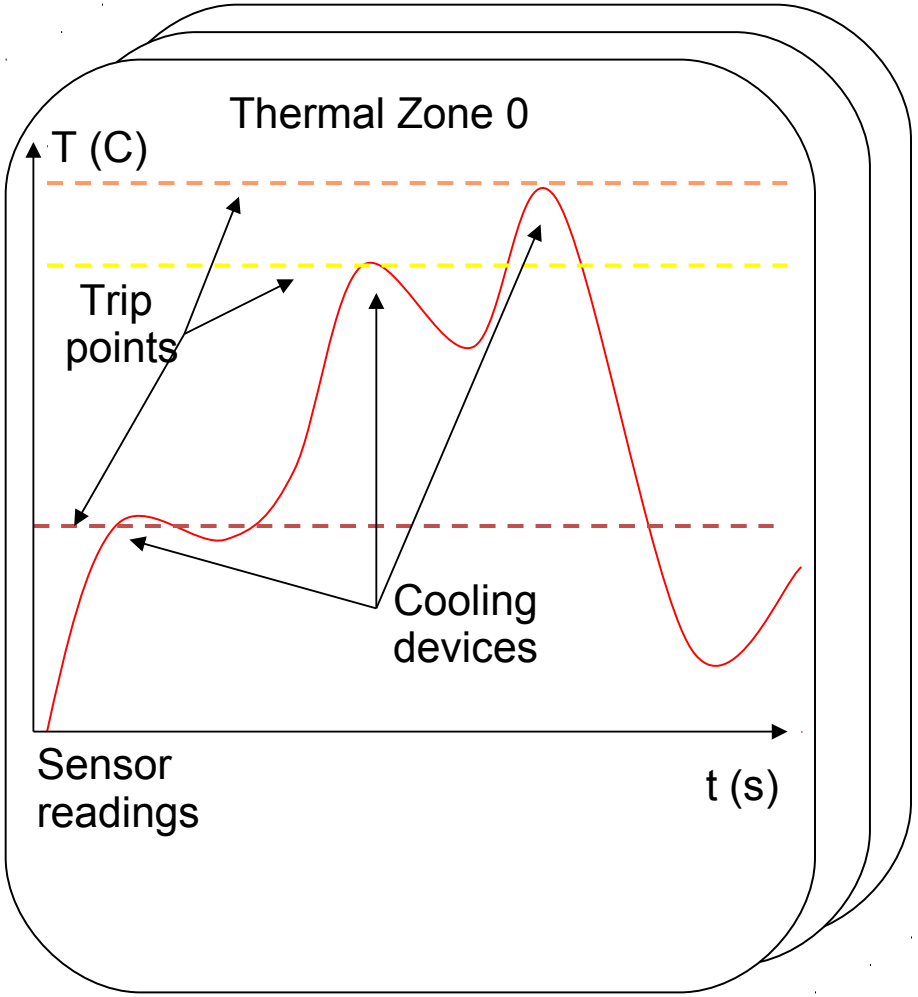
API changes proposals: open discussion

- **Requirement:** Device constraint management in a centralized layer.
  - Upper limits on performance/throughput constraints
  - Arbitration between different types of constraints Upper and Lower limits
- **Requirement:** Device constraint synchronization/arbitration between userspace and kernel space
  - Presented Showcase: CPU.
    - Thermal constraint should not get overwritten by CPUfreq userspace interfaces (constraints need to be aggregated and arbitrated properly).
  - But applies also to other cooling aspects: RF power, audio volume, LCD brightness, co-processor frequency, etc
    - e.g.: change in system audio volume by user vs. thermal constraints on audio device
- **Requirement**: Handshaking/Notification when high performance states are enforced to be removed
  - Not always the removal of high performance state is possible at request time
  - e.g.: Need to propagate the thermal constraint that gets added while running video rec UC. But not always is possible to remove the performance level straight away.
- **Requirement**: Provide a way to map 1 Constraint to N Device constraints representation.
  - There could be constraints shared by several devices: Combo power rails

Back-up Slides

Linux Generic Thermal Framework

- Core code under:
  drivers/thermal/thermal_sys.c
  include/linux/thermal.h
  Documentation/thermal/
- Included in the Linux Kernel in 2008 (by Intel)
- Provides linking between
  - Thermal zones (sensors)
  - Cooling devices (fan, processor speed, etc)
- Standard notification to userland
- Who uses it:
  - **Intel** has an userland based thermal using the interfaces provided by this framework [1]
  - **ACPI** thermal is built on top of this framework
  - **ST Spear** has thermal setup under drivers/thermal
  - **Samsung** is proposing (via Linaro) thermal support for **Exynos4** under drivers/thermal (not yet merged)
  - TI has sent to Linux Kernel staging area a version of the O4/5 thermal driver which uses this API
- Improvements under discussion in linux-pm and linux-acpi



OMAP Software architecture using Linux Thermal FW