PM Infrastructure in the Linux* Kernel

Current Status And Future

Rafael J. Wysocki

Intel Open Source Technology Center

October 4, 2016

Intel OpenSource TECHNOLOGY CENTER

Outline

Intel OpenSource TECHNOLOGY CENTER

What We Are After

Objective

Use only as much energy as needed to achieve sufficient performance.

What software can do

1. Determine how much throughput/capacity is needed.
2. Determine how much latency is acceptable.
3. Enumerate the hardware PM features.
4. Use them to deliver exactly as much as necessary.

What about the Linux kernel?

Intel OpenSource TECHNOLOGY CENTER

The Kernel's Role

Provide means of control

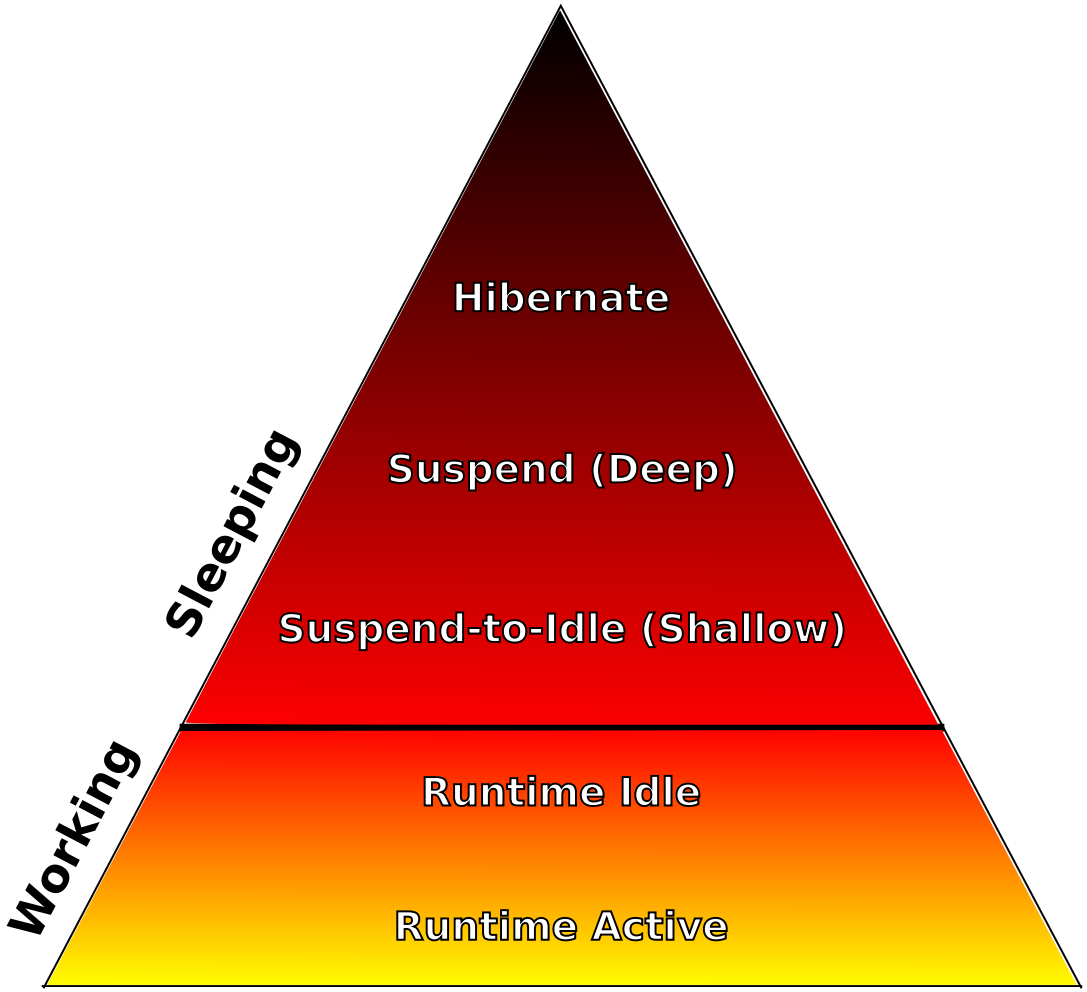
- State selection (component level).
- Carrying out transitions between states (the "mechanics").
- Response to events (e.g. wakeup).

Estimate the needs

- Use information available internally (e.g. from the CPU scheduler).
- React to the actions of user space.
- Follow trends/patterns.

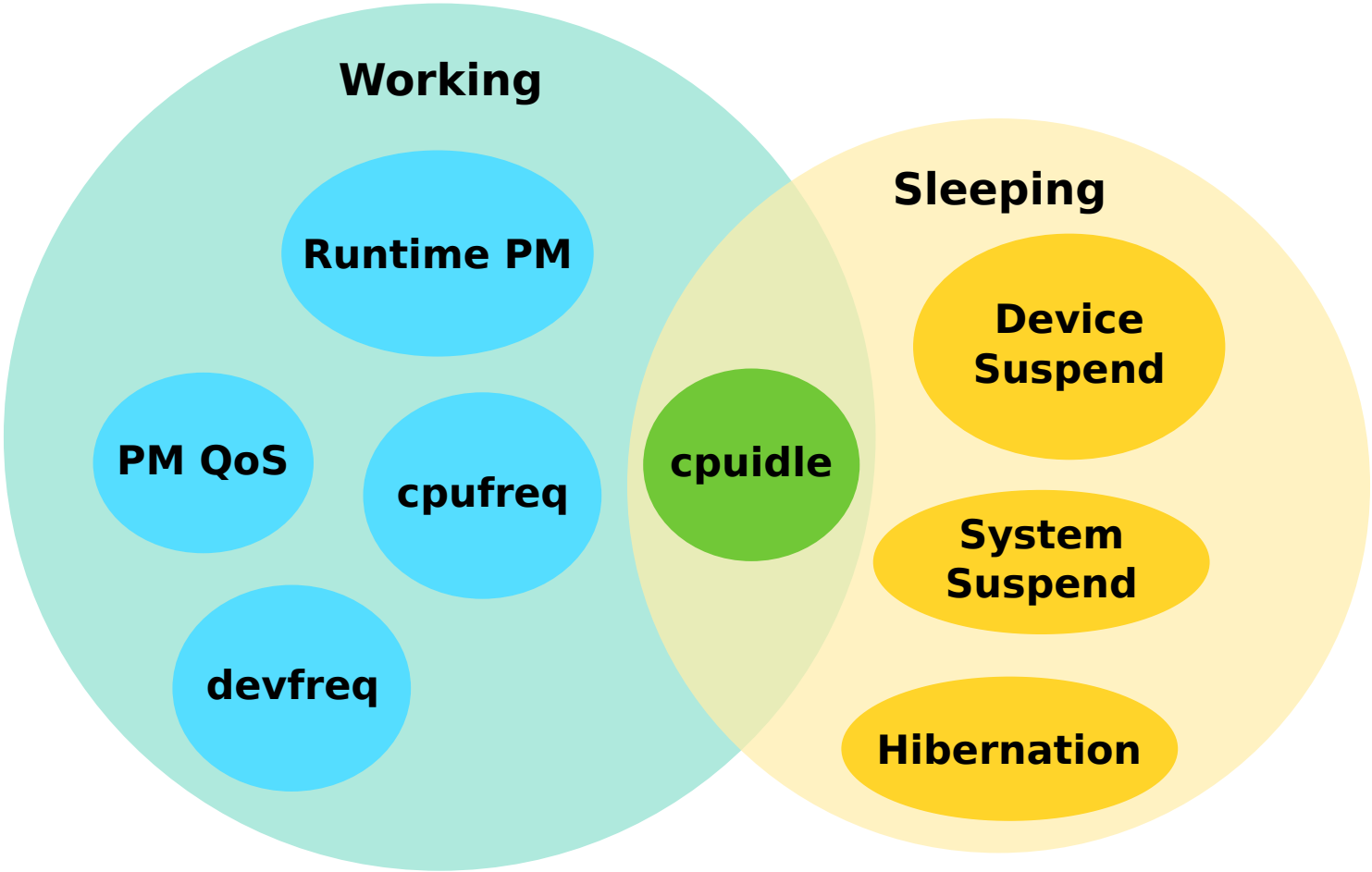
Intel OpenSource TECHNOLOGY CENTER

Working-State PM and System-Wide PM (Sleep States)



Intel OpenSource TECHNOLOGY CENTER

Power Management Frameworks in The Linux Kernel



Intel OpenSource TECHNOLOGY CENTER

System-Wide PM Overview

Global energy-saving states, "frozen" user space

Suspend-to-Idle, Standby, Suspend-to-RAM, Hibernate.

Controlled by user space

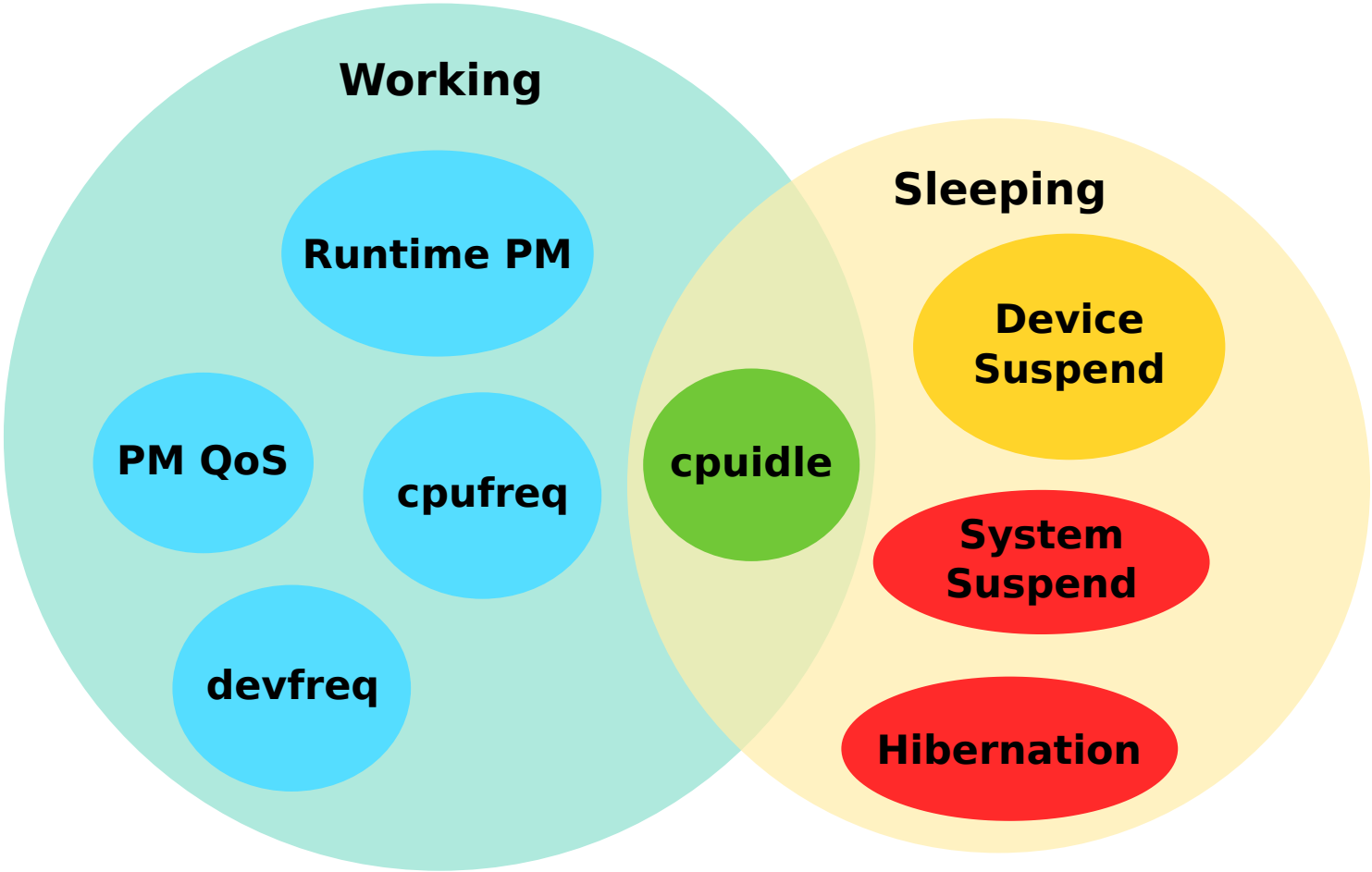
1. User space selects the target state.
2. User space decides when to start transitions.
   - Direct command (*sysfs* write).
   - Autosleep interface.

Used on many systems

Desktop distributions, Android (autosleep interface).

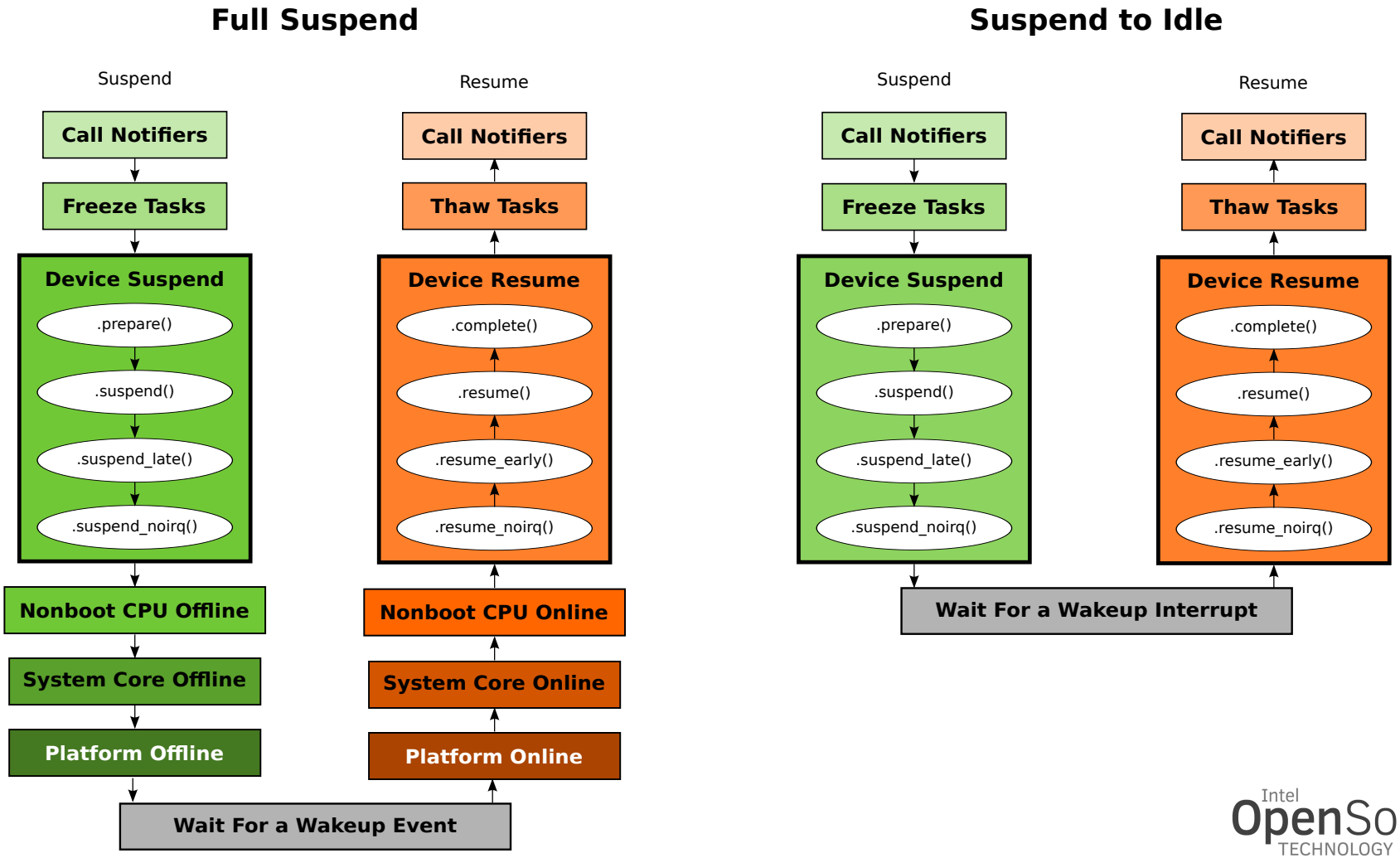
Intel OpenSource TECHNOLOGY CENTER

System-Wide PM Frameworks



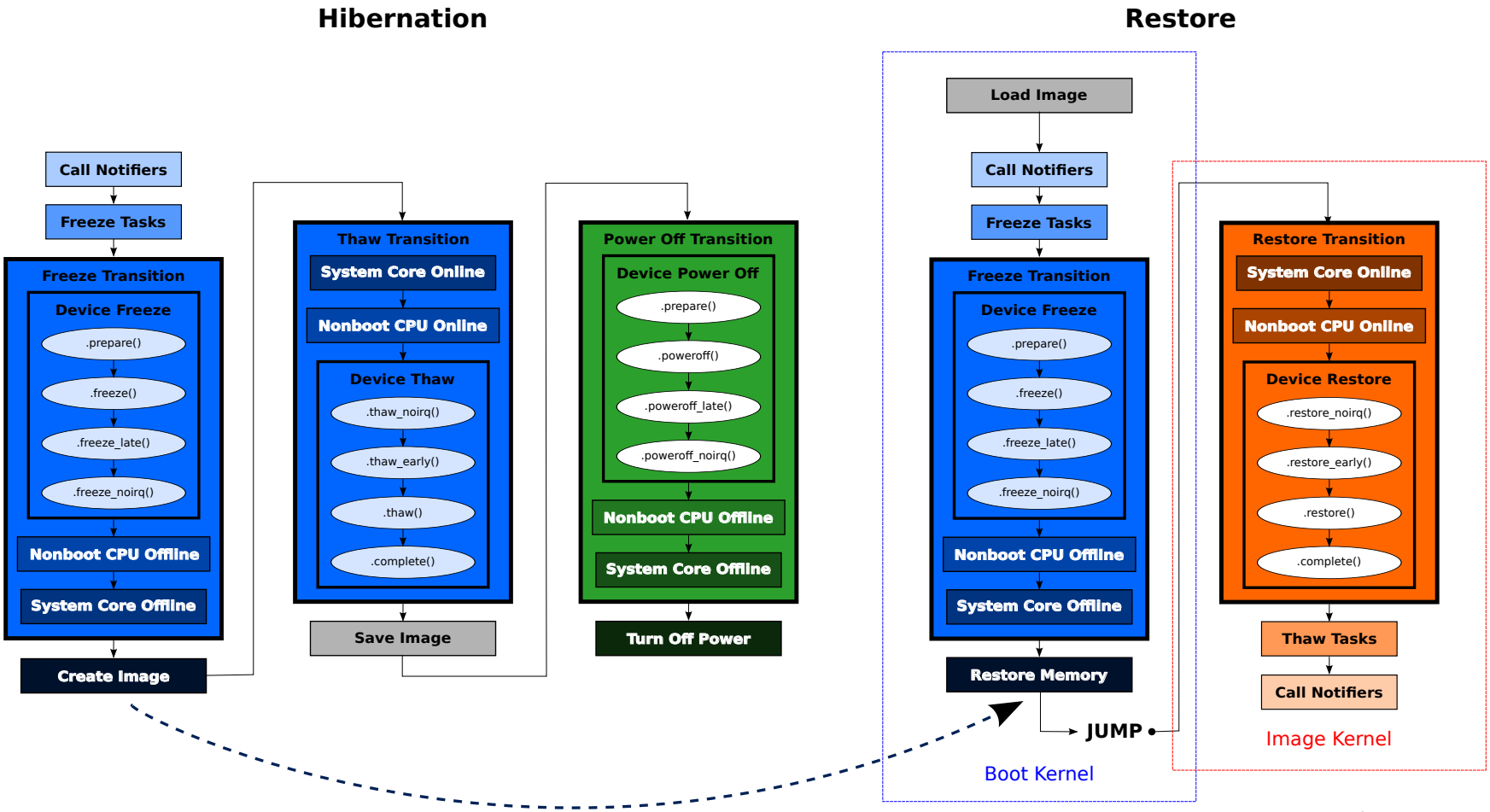
Intel OpenSource TECHNOLOGY CENTER

System Suspend/Resume Control Flow



Intel OpenSource TECHNOLOGY CENTER

System Hibernation/Restore Control Flow



Intel OpenSource TECHNOLOGY CENTER

The Future of System-Wide PM

Hibernation

- Doesn't go away (supported on ARM64 now, works with KASLR).
- Encrypted images problematic.
- Will persistent memory make it obsolete?

Future platforms may not support "platform offline"

- Suspend-to-RAM and Standby may not make sense.
- CPU offline/core offline steps unnecessary.
- Suspend-to-Idle can leverage the existing infrastructure.

Intel OpenSource TECHNOLOGY CENTER

Suspend-to-Idle vs Working-State PM

For long inactivity periods Suspend-to-Idle always uses less energy

- Suspends timekeeping (no timer interrupts).
- Longer average time between wakeups.

Different mechanisms for different use cases

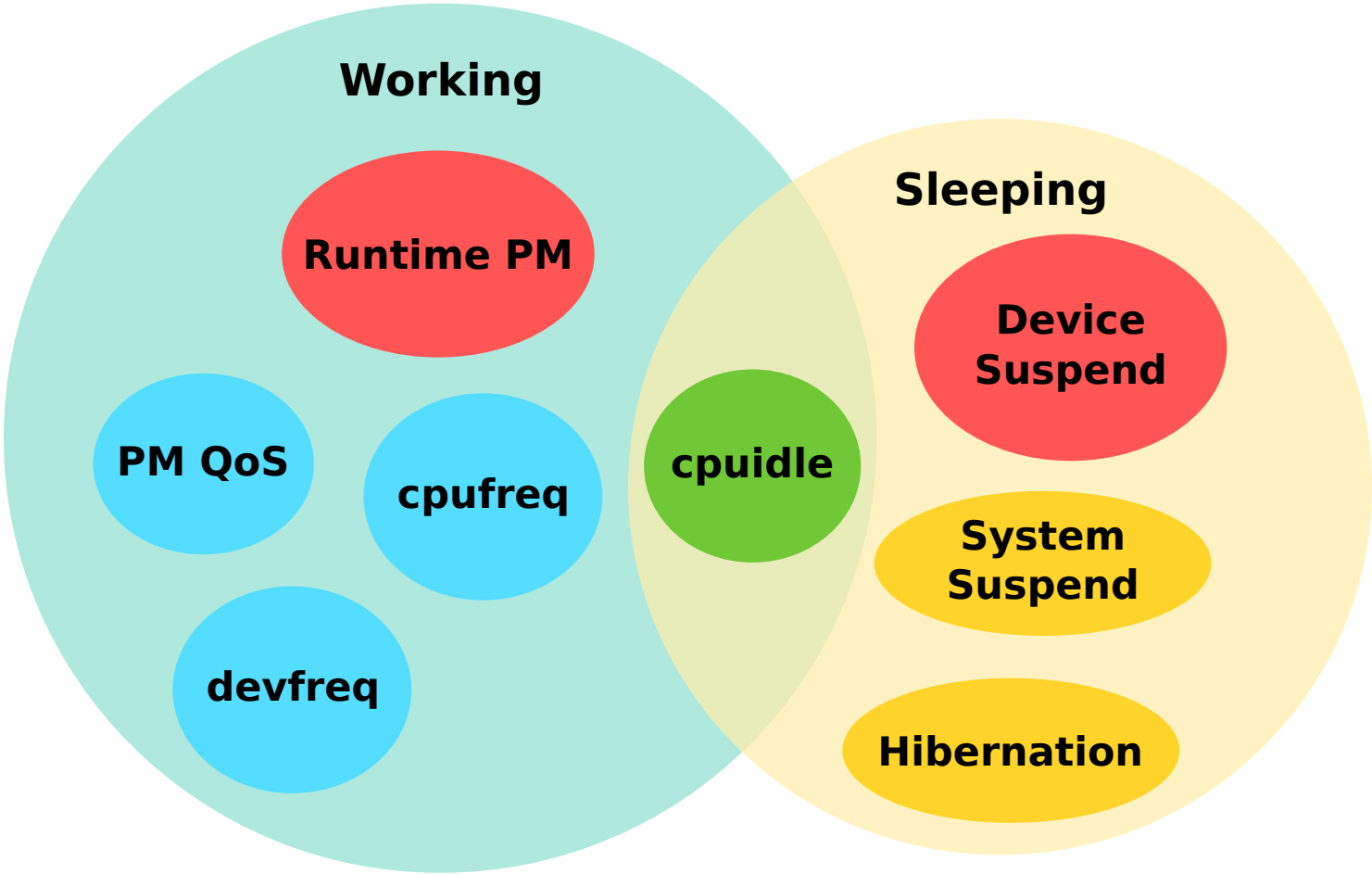
- Suspend on closing laptop lid (working-state PM insufficient).
- Opportunistic idle (Suspend-to-Idle insufficient).

Suspend-to-Idle might be more efficient than it is today

Framework/driver issues to fix.

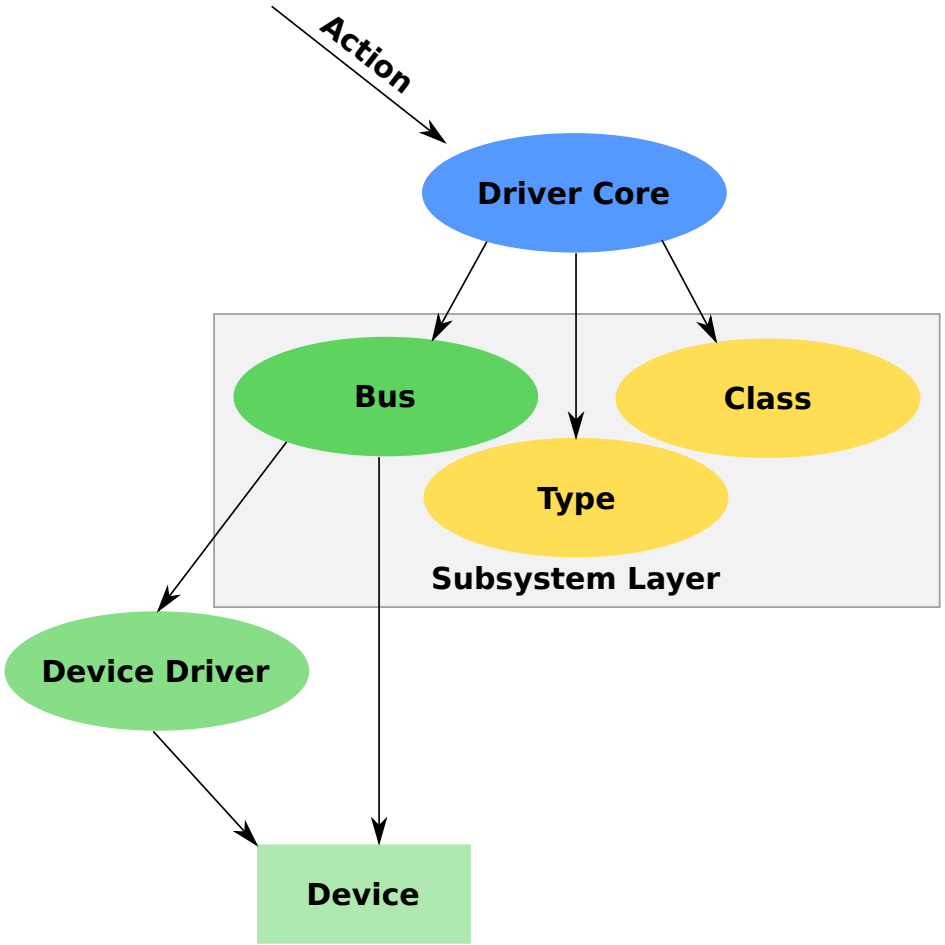
Intel OpenSource TECHNOLOGY CENTER

Suspend/Resume of I/O Devices



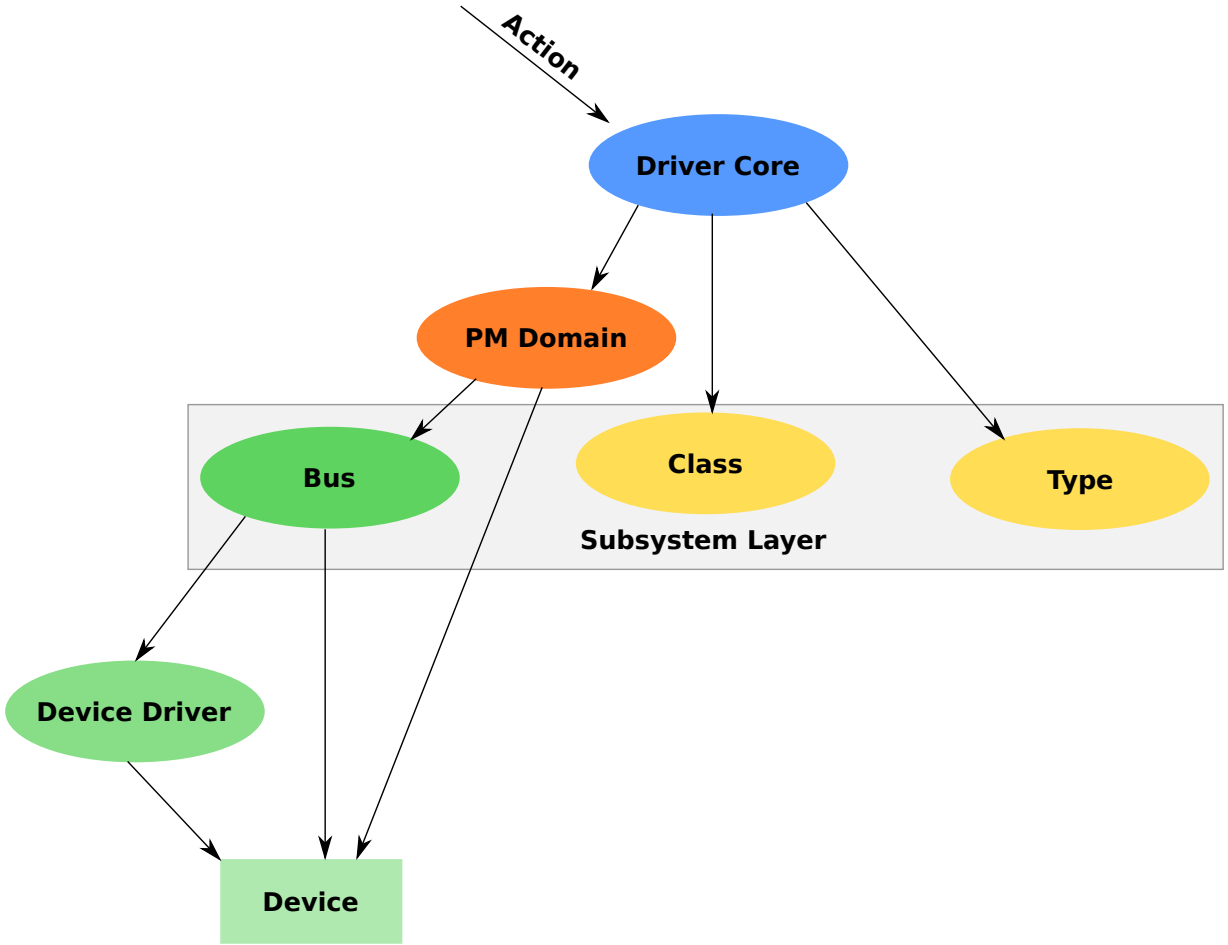
Intel OpenSource TECHNOLOGY CENTER

Device PM Operations and The Driver Core



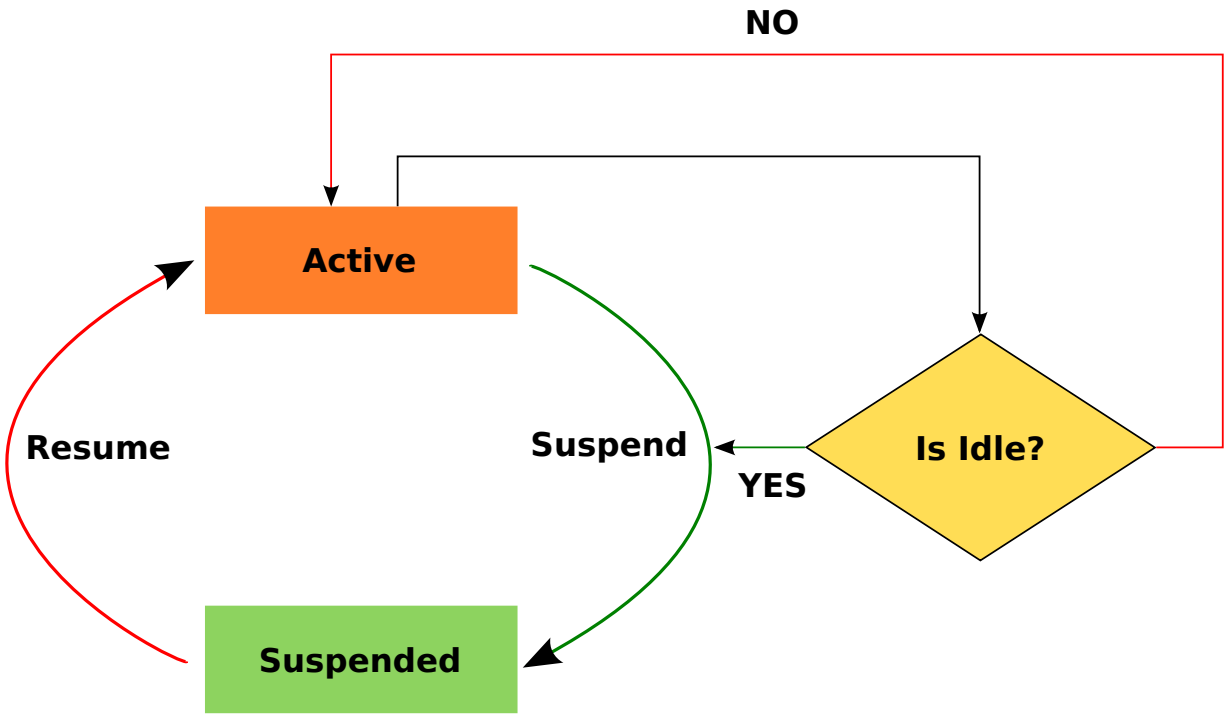
Intel OpenSource TECHNOLOGY CENTER

Device PM Operations and PM Domains



Intel OpenSource TECHNOLOGY CENTER

Device Runtime PM Operations



Intel OpenSource TECHNOLOGY CENTER

Runtime PM on Future Systems

Hardware design trends

- Increasing integration of components.
- Increasing level of support for aggressive PM.

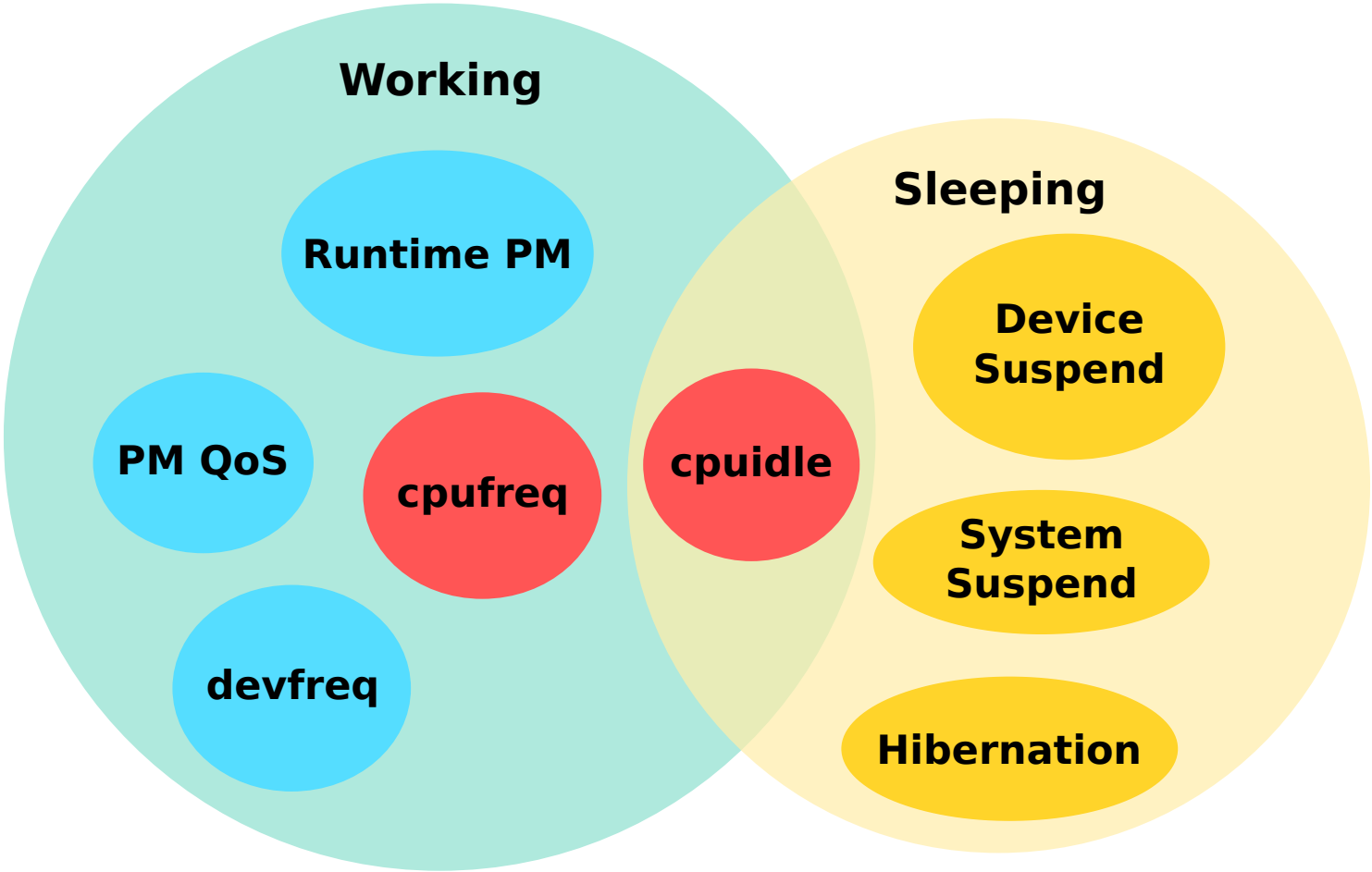
System-on-a-Chip (SoC) configurations

- CPU packages contain I/O devices.
- Package low-power states depend on I/O devices.
- PM features of different components are interdependent.

Challenge: Take dependencies into account

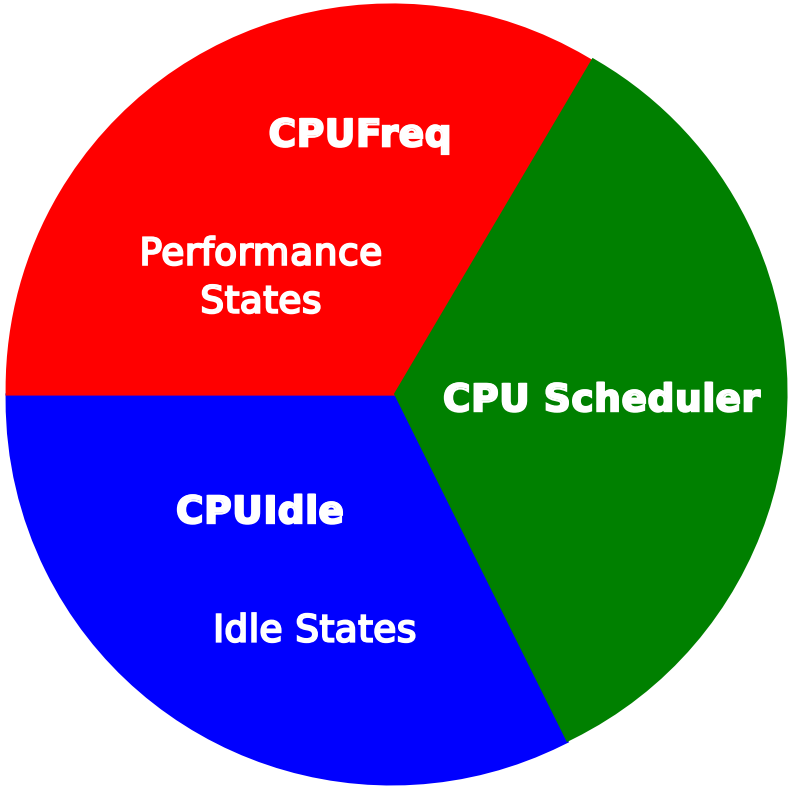
Intel OpenSource TECHNOLOGY CENTER

CPU Power Management Frameworks



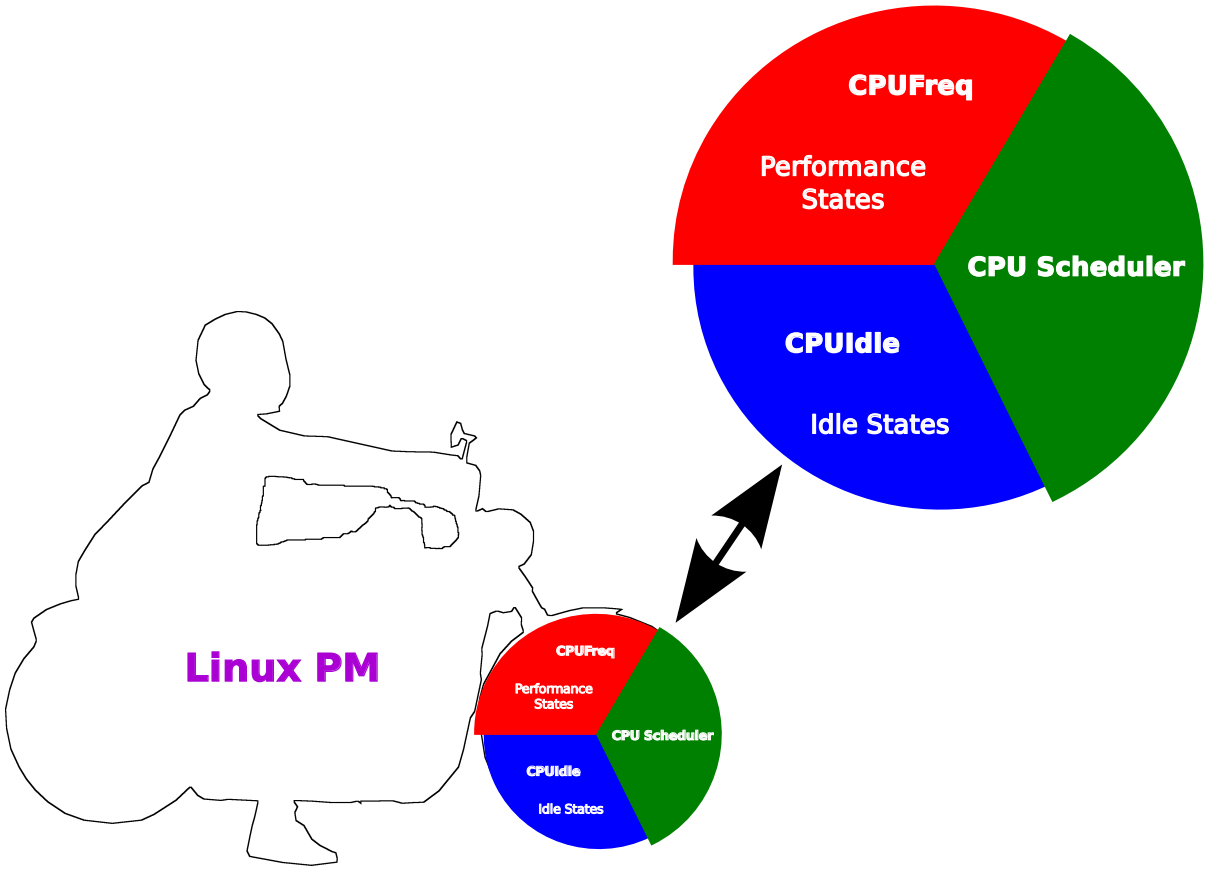
Intel OpenSource TECHNOLOGY CENTER

CPU Power Management Overview



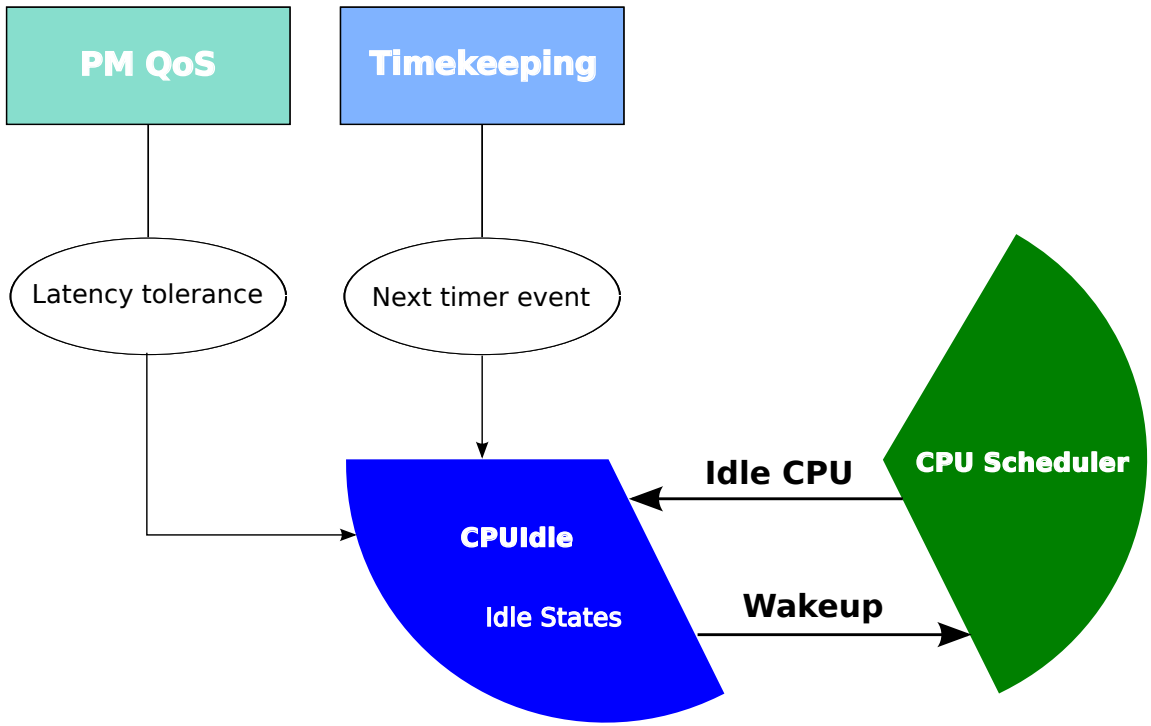
Intel OpenSource TECHNOLOGY CENTER

Cooperation Between Components Is Key



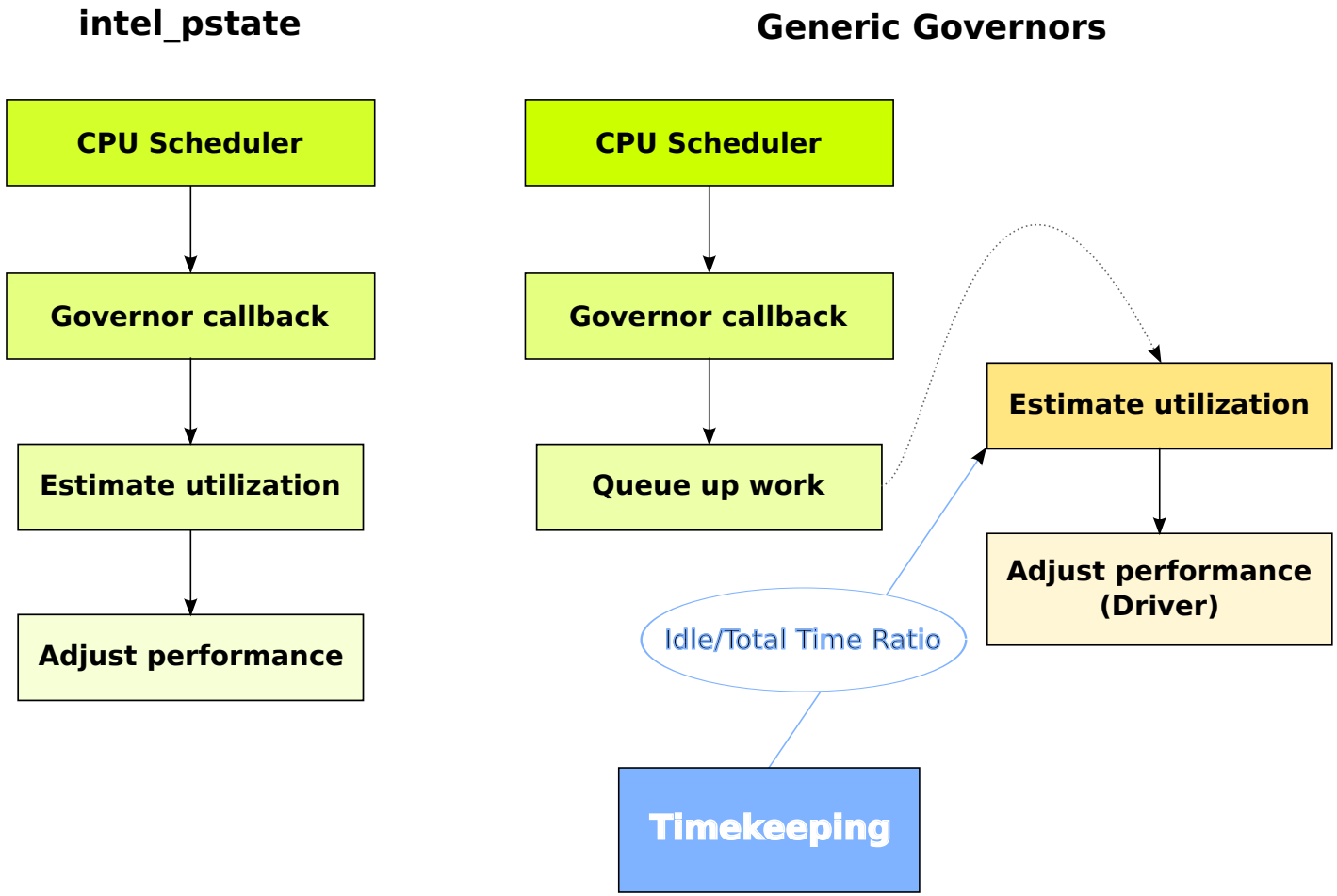
Intel OpenSource TECHNOLOGY CENTER

CPUIdle Workflow



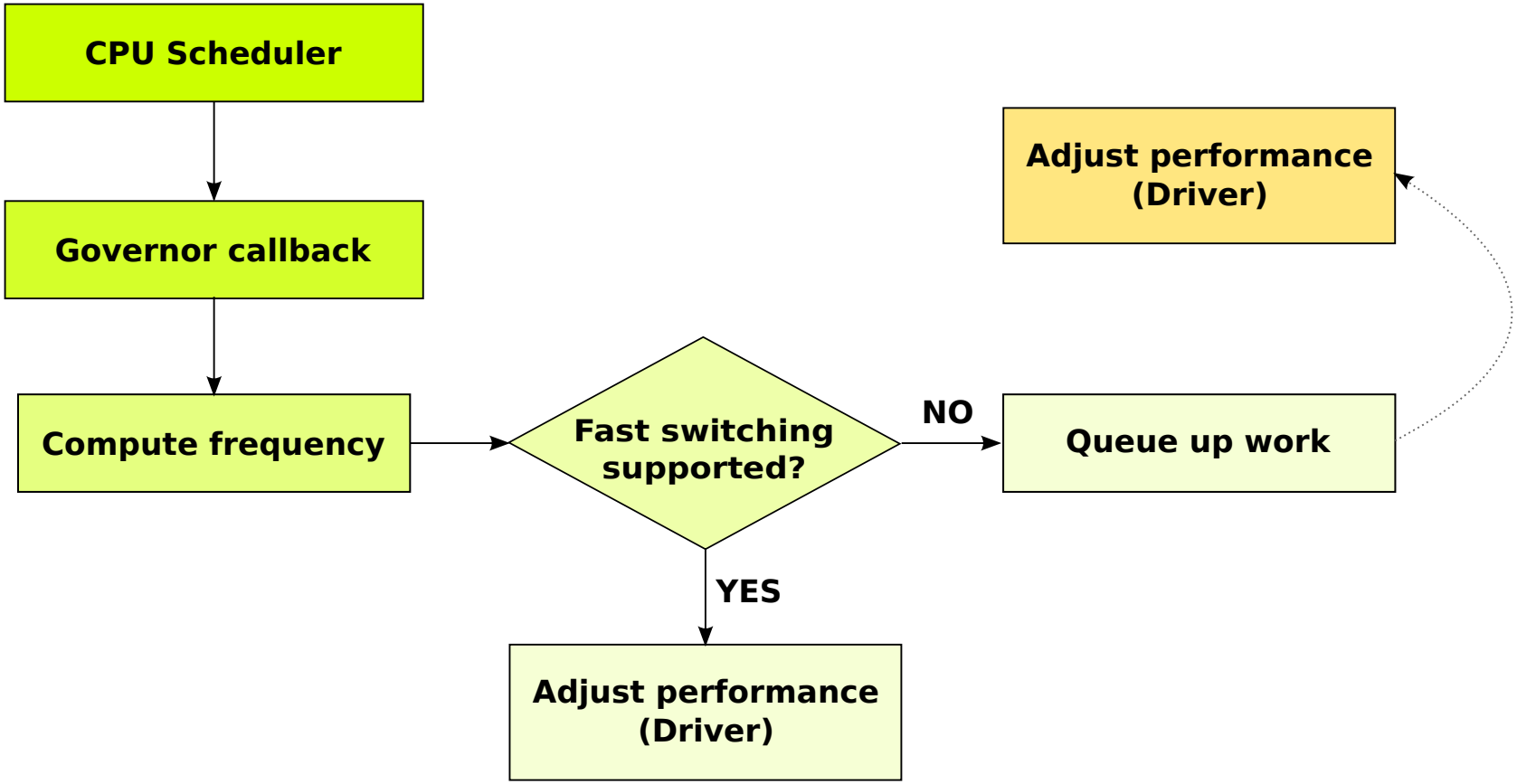
Intel OpenSource TECHNOLOGY CENTER

CPUFreq: Old-Style Governors and intel_pstate



Intel OpenSource TECHNOLOGY CENTER

CPUFreq: `schedutil` Governor



Intel OpenSource TECHNOLOGY CENTER

CPUFreq: Scheduler Hints, Cross-CPU Updates

Observations

- It is good to take blocking on I/O (*IOwait*) into account.
- Different scheduling classes require different handling.

Solution: Scheduler hints (work in progress)

- Pass hints from the CPU scheduler to governor callbacks.
- A hint can represent the reason update or similar.

Observation

In some cases it is beneficial to invoke governor callbacks cross-CPU.

Intel OpenSource TECHNOLOGY CENTER

The Future of CPU Power Management

Complex topologies (hardware threads, modules, packages)

Many different scheduling strategies are potentially viable.

PM-Aware Scheduling Conjecture

Energy efficiency may be improved without hurting performance by making the CPU scheduler drive CPU power management as a whole.

Predictions (usual disclaimers apply)

- PM-aware scheduling will enter the mainline kernel.
- CPUFreq and CPUIdle will be combined.

Intel OpenSource TECHNOLOGY CENTER

Conclusion

- The Linux kernel supports power management in a number of ways.
- Both system-wide and working state (runtime) PM are supported.
- Support for system-wide PM in device drivers is generally better.
- I/O device runtime PM support is improving.
- CPU PM is well supported, consolidation in progress.
- Hardware design trends increase PM complexity.
- Interdependencies between PM features are challenging.

Questions?

Intel OpenSource TECHNOLOGY CENTER

References


R. J. Wysocki, *CPUfreq and The Scheduler: Revolution in CPU Power Management* (http://events.linuxfoundation.org/sites/events/files/slides/cpufreq_and_scheduler_0.pdf).


Neil Brown, *Improvements in CPU frequency management* (http://lwn.net/Articles/682391/).


Len Brown, *Suspend/Resume at the Speed of Light* (http://events.linuxfoundation.org/sites/events/files/slides/Brown-Linux-Suspend-at-Speed-of-Light-LC-EU-2015.pdf).


R. J. Wysocki, *Getting More Out Of System Suspend In Linux* (http://events.linuxfoundation.org/sites/events/files/slides/linux_suspend.pdf).


R. J. Wysocki, *Power Management in the Linux Kernel – Current Status and Future* (http://events.linuxfoundation.org/sites/events/files/slides/kernel_PM_plain.pdf).


Jonathan Corbet, *The cpuidle subsystem* (http://lwn.net/Articles/384146/).


R. J. Wysocki, *Why We Need More Device Power Management Callbacks* (https://events.linuxfoundation.org/images/stories/pdf/lfcs2012_wysocki.pdf).

Intel OpenSource TECHNOLOGY CENTER

Documentation and Source Code

- Documentation/cpu-freq/*
- Documentation/cpuidle/*
- Documentation/power/devices.txt
- Documentation/power/pci.txt
- Documentation/power/runtime_pm.txt
- include/linux/cpufreq.h
- include/linux/cpuidle.h
- include/linux/device.h
- include/linux/pm.h
- include/linux/pm_runtime.h
- include/linux/suspend.h
- drivers/acpi/processor_idle.c
- drivers/base/power/*
- drivers/cpufreq/*
- drivers/cpuidle/*
- kernel/power/*

Intel OpenSource TECHNOLOGY CENTER

Legal Information

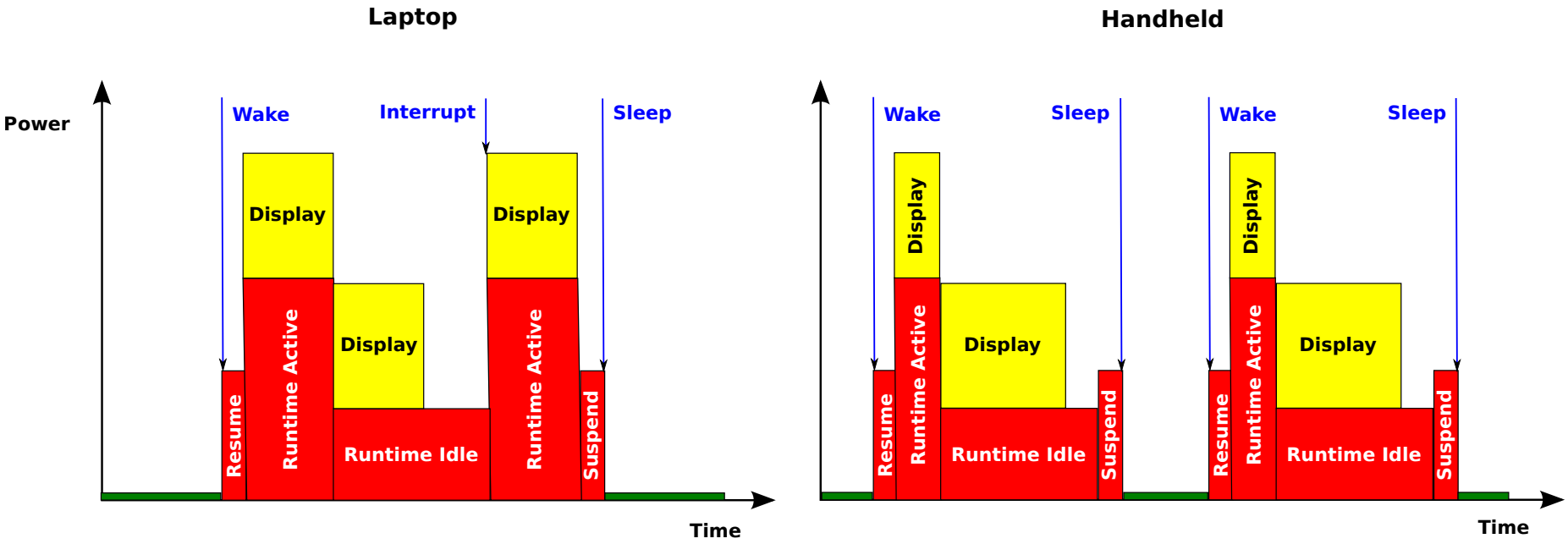
Intel OpenSource TECHNOLOGY CENTER

Thanks!

Thank you for attention!

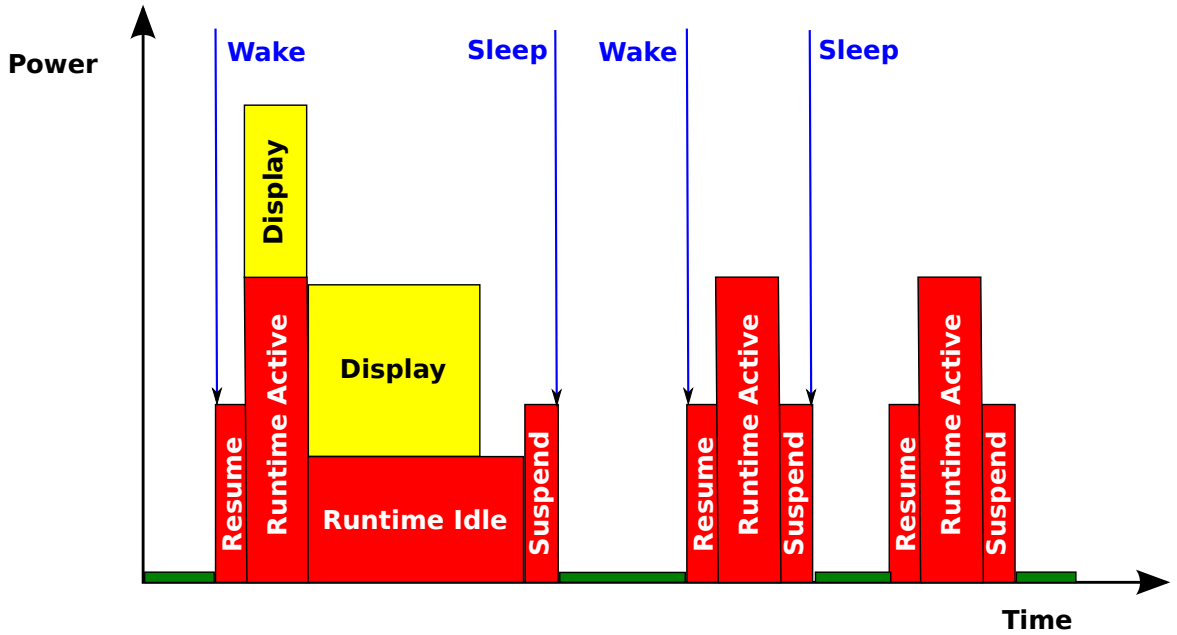
Intel OpenSource TECHNOLOGY CENTER

Laptop/Handheld Usage Scenarios



Intel OpenSource TECHNOLOGY CENTER

Dark Resume Scenario



Intel OpenSource TECHNOLOGY CENTER