Cool Hand Linux*
Handheld Thermal Extensions

Len Brown
len.brown@intel.com
Intel Open Source Technology Center

Hari Seshadri
harinarayanan.seshadri@intel.com
Intel Ultra-Mobile Group

Handheld Thermal Challenges

Handheld != Notebook
- Skin temperature is critical
- Fan-less
- Small, limited thermal dissipation
- non-CPU devices may dominate heat generation

ACPI 3.0 Thermal Model

Thermal relationship table
- Tells the OS the relative thermal contribution of each device to each thermal zone
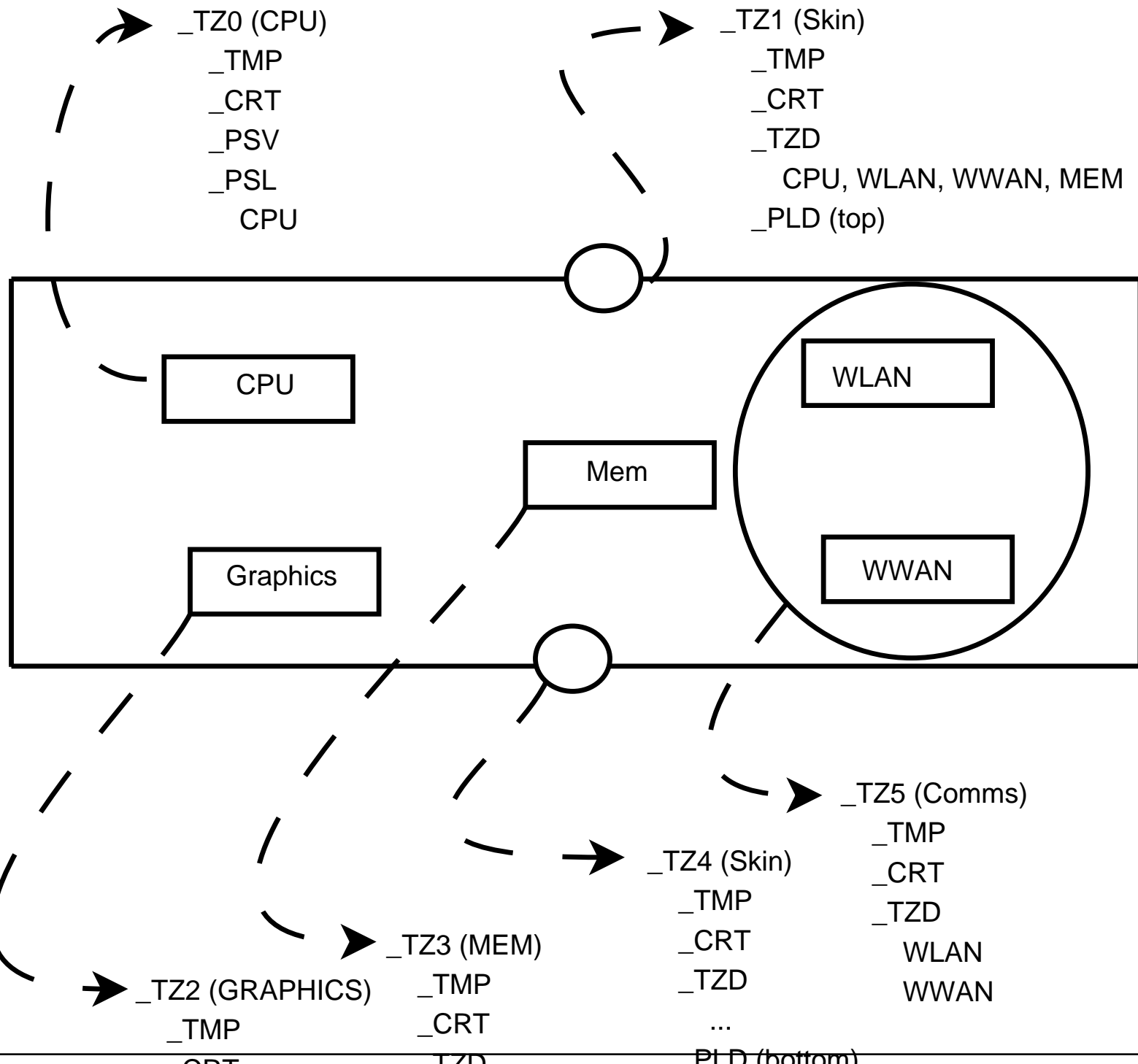- Significant system design and validation effort

No plans to implement on Linux

ACPI 2.0 Thermal Model

Thermal zone
- Temperature
- Trip points
  - CRT - Critical Shutdown
  - HOT - Hibernate
  - PSV - Passive
    - PSL - Associated Passive List (of Processors)
  - ACx - (multiple) Active
- TZD - Associated Thermal Zone Devices

Mapping Sensors to Thermal Zones

_TZ0 (CPU)
   _TMP
   _CRT
   _PSV
   _PSL
      CPU

_TZ1 (Skin)
   _TMP
   _CRT
   _TZD
      CPU, WLAN, WWAN, MEM
   _PLD (top)

CPU

WLAN

Mem

Graphics

WWAN

_TZ5 (Comms)
   _TMP
   _CRT
   _TZD
      WLAN
      WWAN

_TZ4 (Skin)
   _TMP
   _CRT
   _TZD
      ...

_TZ3 (MEM)
   _TMP
   _CRT
      _TZD

_TZ2 (GRAPHICS)
   _TMP
   _CRT

_PLD (bottom)

ACPI Embedded Controller (EC)

Polls inexpensive dumb sensors
Tracks trip-point state
Sends events to CPU

ACPI is not special, "native" EC can do this too...
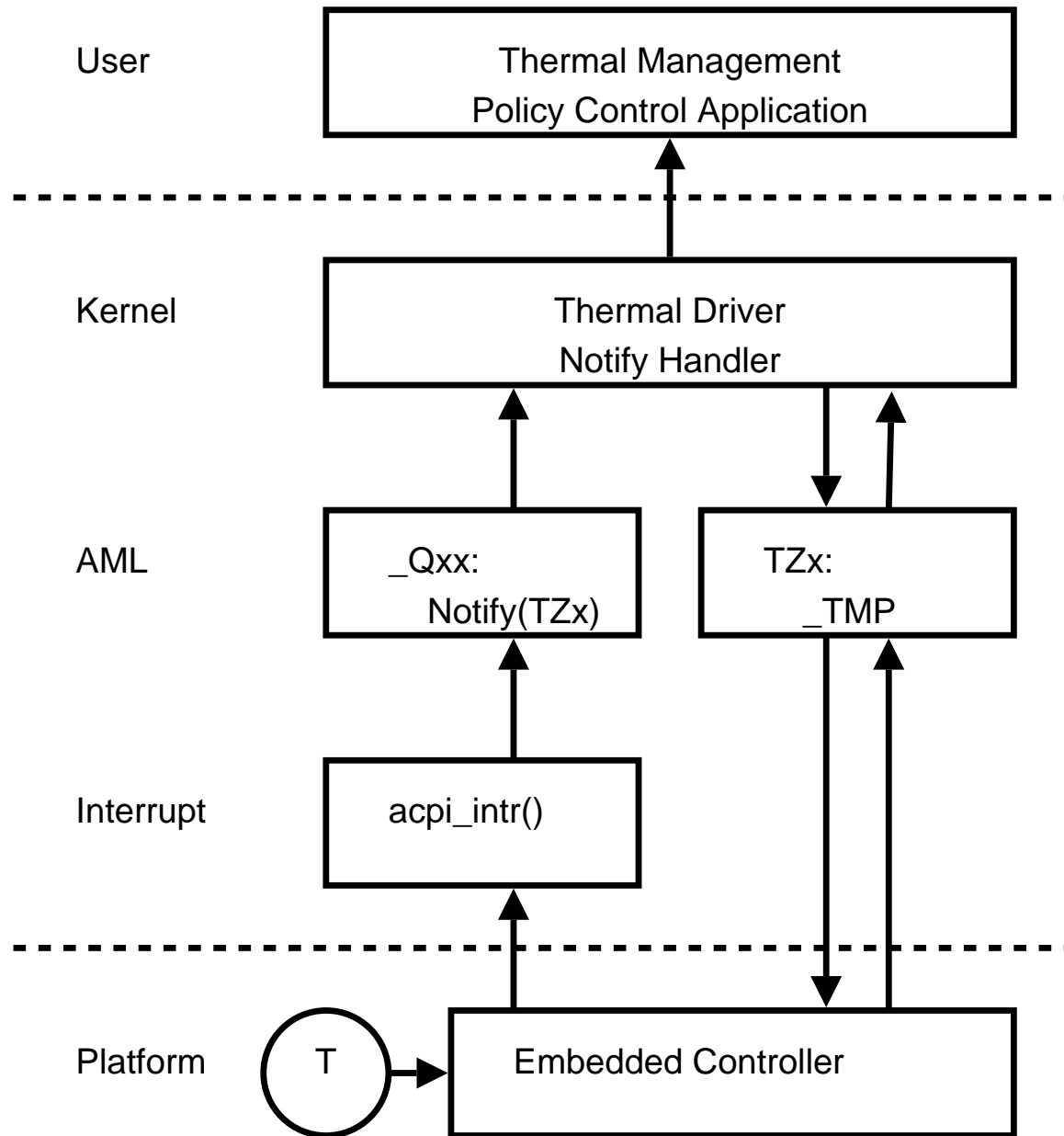
Using ACPI for Handheld Thermal Events

Processor Thermal Zone
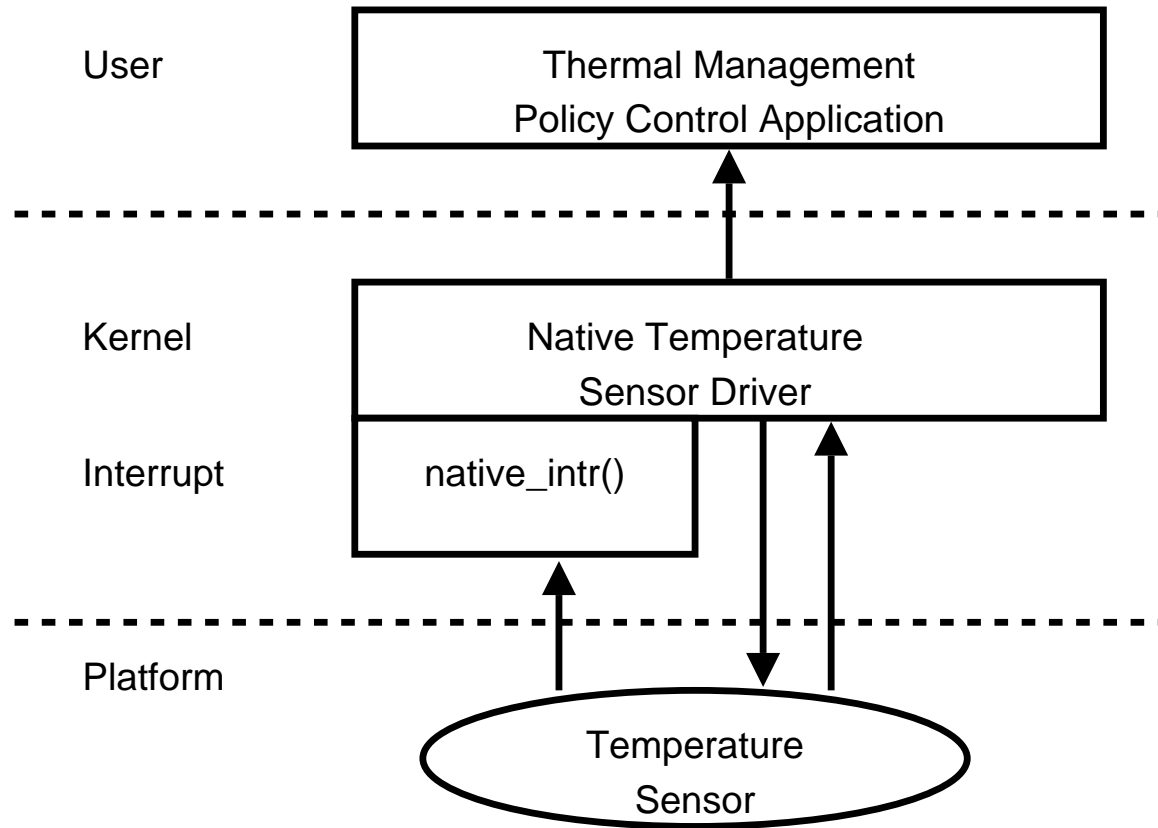- ☐ Use existing critical shutdown
- ☐ Use existing passive trip and throttling

Non-processor Thermal Zones
- ☐ Use existing critical shutdown
- ☐ Report "interesting" temperature change events

Thermal Event Delivery via ACPI

Thermal Event Delivery via Native Driver

User

Thermal Management
Policy Control Application

Kernel

Native Temperature
Sensor Driver

Interrupt

native_intr()

Platform

Temperature
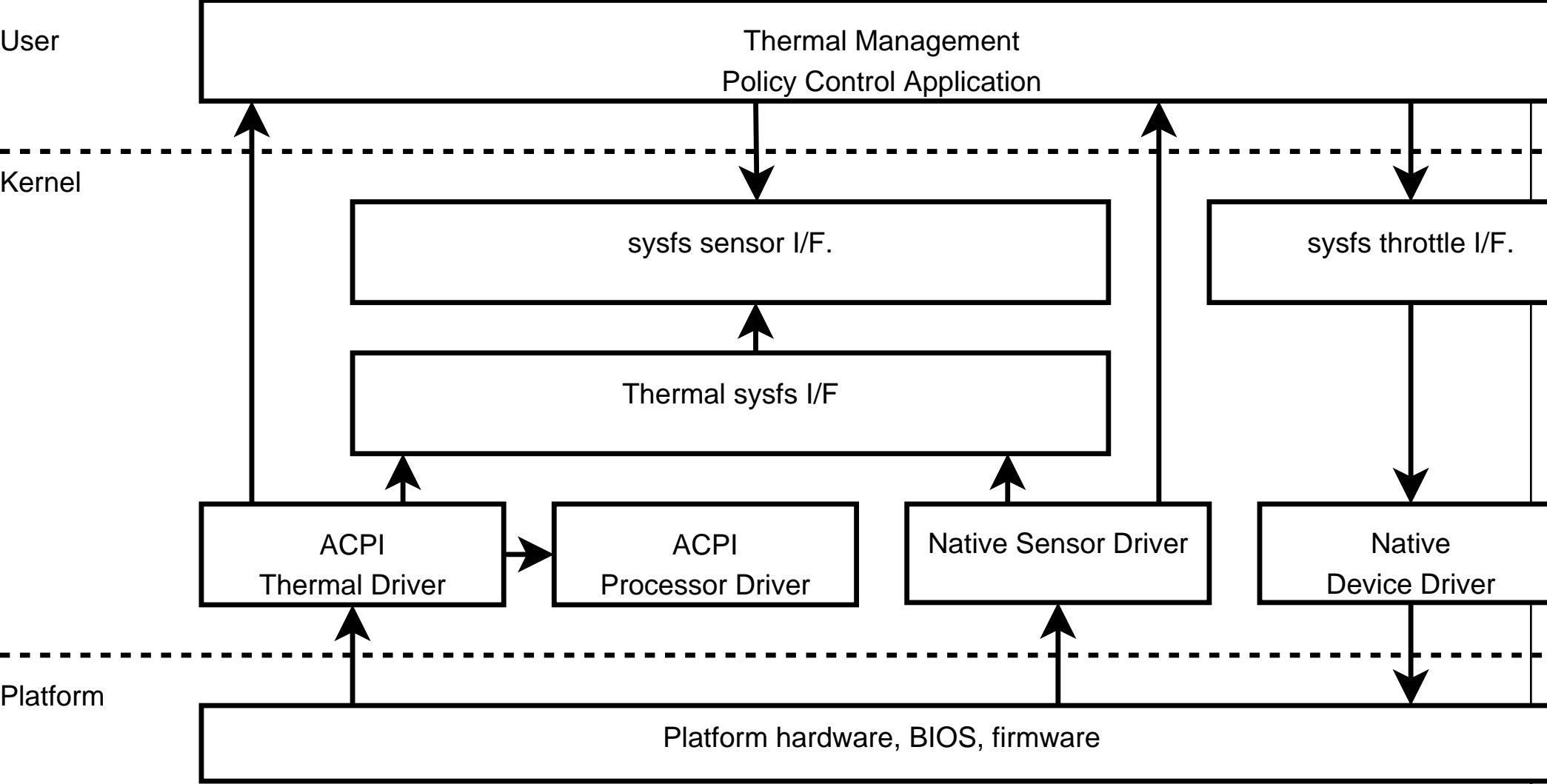**Sensor**

Design Decisions

Use low-power EC to poll inexpensive temperature sensors
Policy decisions made by user-space
Kernel provides mechanism only
  ☐ Deliver events to user-space
  ☐ Communicate throttling decisions to native drivers

Software Architecture

User	**Thermal Management Policy Control Application**

Kernel

sysfs sensor I/F.	sysfs throttle I/F.

| Thermal sysfs I/F | |

ACPI Thermal Driver	ACPI Processor Driver	Native Sensor Driver	Native Device Driver

Platform

Platform hardware, BIOS, firmware

Thermal event notification

netlink message from thermal-zone driver to user-space
status is also available via sysfs

Proposed Thermal Zone sysfs interface

temp1_input, temp1_alarm
  ☐ Current temperature [_TMP] (RO)
  ☐ Temperature change occurred (RW)
temp1_crit, temp1_crit_alarm
  ☐ Critical alarm temperature [_CRT] (RO)
  ☐ Critical alarm occurred (RW)
temp1_passive, temp1_passive_alarm
  ☐ Passive alarm temperature [_PSV] (RO)
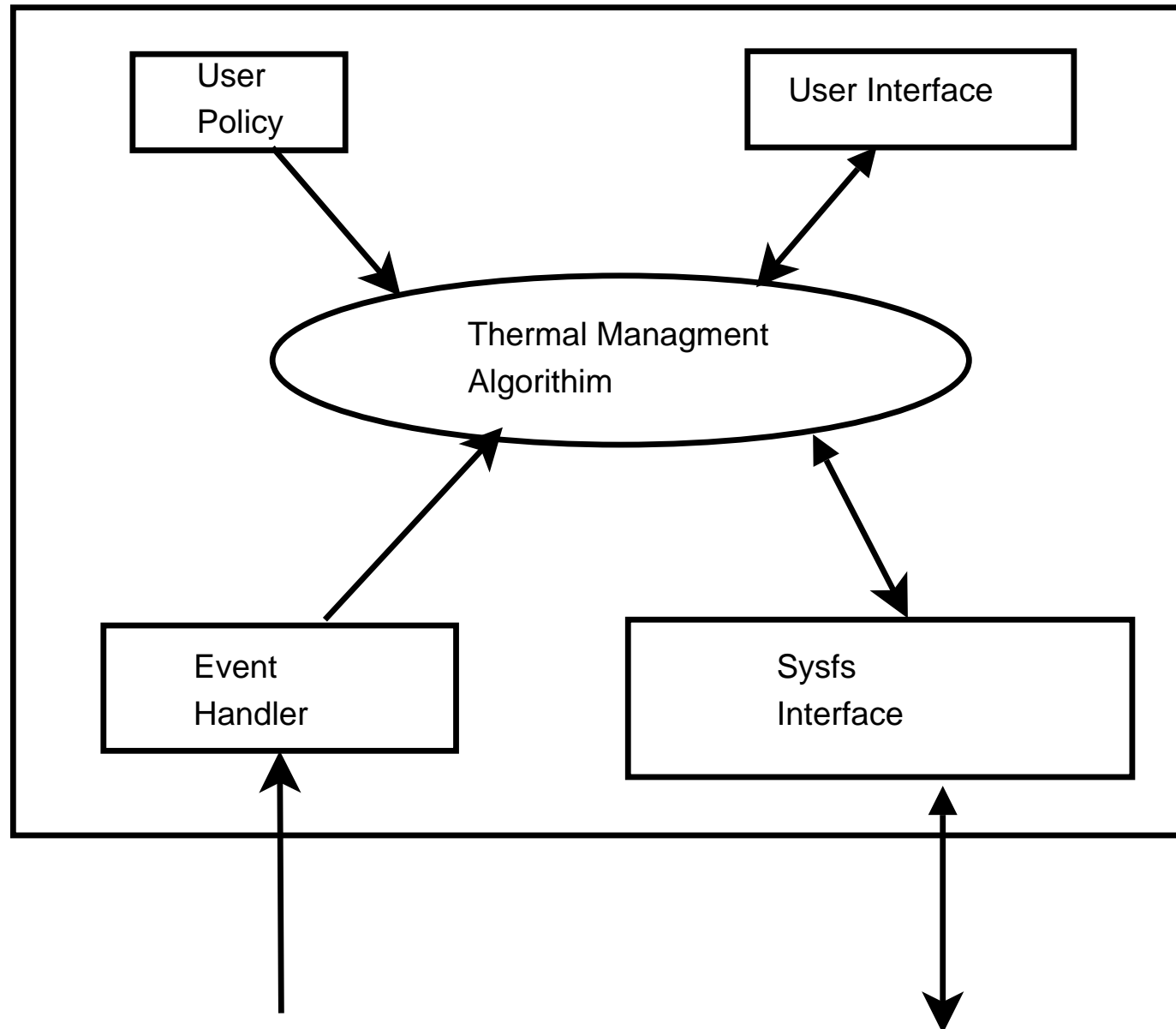 ☐ Passive alarm occurred (RW)
<device_name1>
  ☐ Link to device1 for zone (RO)

Throttling sysfs properties

throttling
throttling_max

Thermal Policy Control Application

Opens

EC must know granularity for "interesting" temp changes
  □ Add API for OS to tell the EC?

Summary

User-space owns policy decisions
Kernel acts as communication conduit
- thermal-driver to user-space
- user-space to native throttling driver

Simple approach applicable both ACPI and native systems

Thank you!

Intel Open Source Technology Center