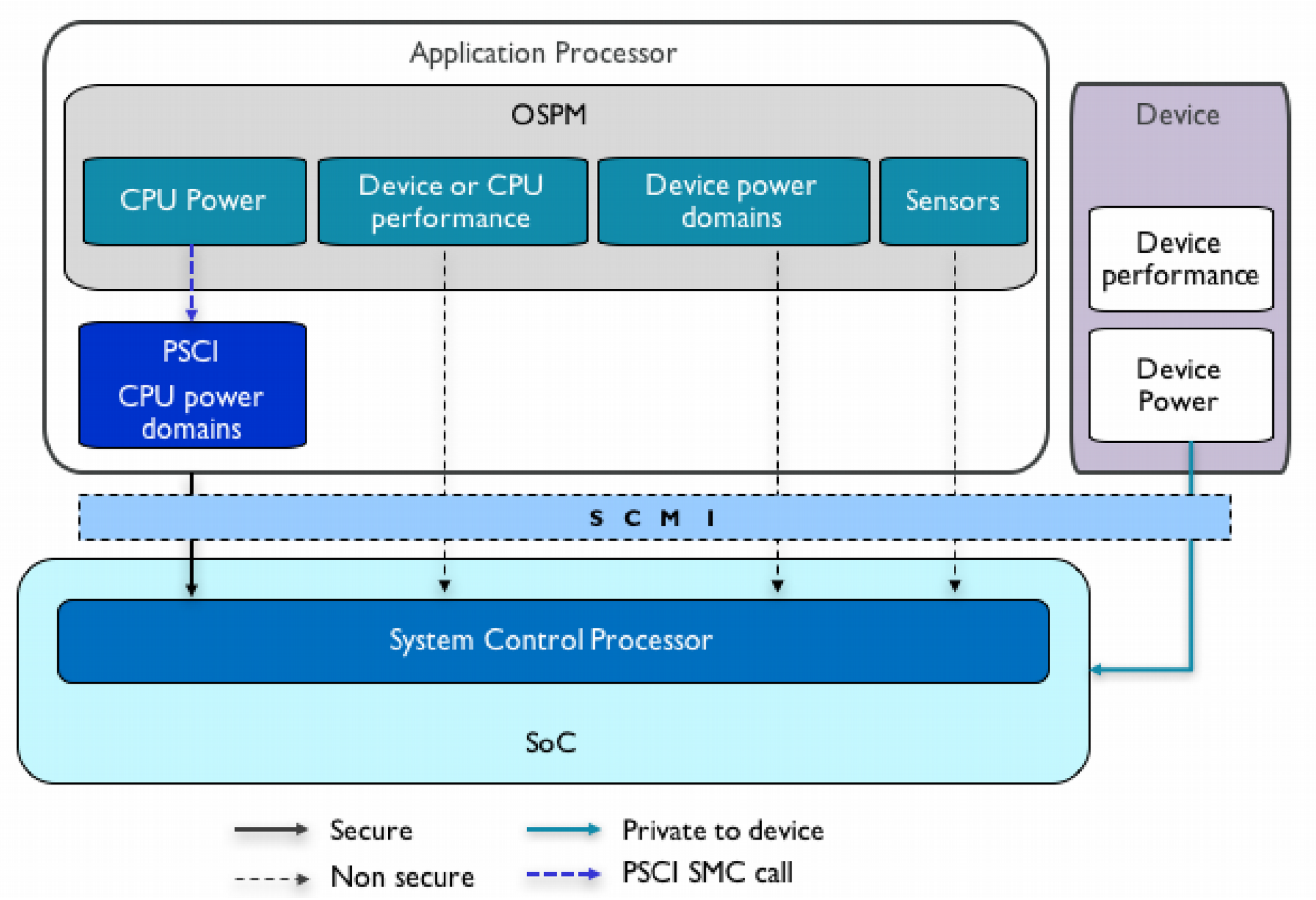
SCMI Virtual Sensors Implementation

For GENIVI Hypervisor Project

System Control and Management Interface

- Replacement for ARM SCPI, TI SCI, QCOM RPM, Nvidia Tegra BPMP, etc.
- Designed to minimize the dependency on the mailbox/transport hardware
- Implements
  - Discovery and self-description of the interfaces it supports
  - Power domain management, which is the ability to place a given device or domain into the various power-saving states that it supports
  - Performance management, which is the ability to control the performance of a domain that is composed of compute engines such as application processors (APs), GPUs, or other accelerators
  - Clock management, which is the ability to set and inquire rates on platform-managed clocks
  - Sensor management, which is the ability to read sensor data, and be notified of sensor value changes

Example system that implements SCMI



SMI transport support in HW

- At the moment, only Mailbox Transport is defined for SCMI in order to implement communication between APs and SCP – which can be abstracted on OS level
- Currently there are no (?) SCP-ready designs, BUT…
- …in a virtualized system, remote domain controlling clocks, sensors, etc. can be viewed as an SCP – i.e. same protocol can be used with hypervisor-based transport implementing "virtual mailbox", not depending on HW support
  - Xen implementation for ARM (done in context of DVFS support for virtualized systems): https://lists.xenproject.org/archives/html/xen-devel/2017-11/msg00664.html
- SCMI and its transport is currently defined for ARM, but there (seemingly) is no obstacle to use it on x86 with virtualized system

Protocol-defined sensors

0 None	15 kPa	30 Cubic Feet	45 Gauss	60 Bits	75 Collisions
1 Unspecified	16 PSI	31 Meters	46 Gilberts	61 Bytes	76 Packets
2 Degrees C	17 Newtons	32 Cubic Centimeters	47 Henries	62 Words (data)	77 Messages
3 Degrees F	18 CFM	33 Cubic Meters	48 Farads	63 Doublewords	78 Characters
4 Degrees K	19 RPM	34 Liters	49 Ohms	64 Quadwords	79 Errors
5 Volts	20 Hertz	35 Fluid Ounces	50 Siemens	65 Percentage	80 Corrected Errors
6 Amps	21 Seconds	36 Radians	51 Moles	66 Pascals	81 Uncorrectable Errors
7 Watts	22 Minutes	37 Steradians	52 Becquerels	67 Counts	82 Square Mils
8 Joules	23 Hours	38 Revolutions	53 PPM (parts/million)	68 Grams	83 Square Inches
9 Coulombs	24 Days	39 Cycles	54 Decibels	69 Newton-meters	84 Square Feet
10 VA	25 Weeks	40 Gravities	55 DbA	70 Hits	85 Square Centimeters
11 Nits	26 Mils	41 Ounces	56 DbC	71 Misses	86 Square Meters
12 Lumens	27 Inches	42 Pounds	57 Grays	72 Retries	
13 Lux	28 Feet	43 Foot-Pounds	58 Sieverts	73 Overruns/Overflows	
14 Candelas	29 Cubic Inches	44 Ounce-Inches	59 Color Temperature Degrees K	74 Underruns	255 OEM Unit

Protocol-defined operation modes

- Synchronous Access – This method is recommended for sensors whose data is immediately available or is internally cached by the platform, and can be returned immediately to the requesting agent. Examples include platform event counters, or sensor data samples that are stored in internal memory within the platform.
- Asynchronous Access – This method is recommended for sensors whose data is not cached by the platform or for sensors that are slow to read. An example of this could be an on-die thermal sensor.
- Event Notification – The agent can register for receiving notifications on specific sensor values, conditions, or states of interest.
- Shared Memory – In this scheme, the platform periodically updates the sensor value in an area of memory that is shared between agents and the platform.

Implementation

- Linux kernel 4.17 https://patchwork.ozlabs.org/patch/879925/
- Currently no support in virtio
- Xen reference generic PV implementation in progress
  - DT defines sensors configuration in virtual domain
  - Kernel SCMI agent implements application access to virtualized sensors
  - Xen virtual mailbox implements intra-domain communication
  - Userspace backend implements “platform” protocol interface with HW
- Biggest question – is it ok to use on x86?