System Control and Management Interface

Charles García-Tobin
Agenda

- What’s this all about
- Requirements of SCMI
- What’s the status
A nice future

We’d like a common methodology for system management functions

This requires a method that is:

1. Extensible
2. OS agnostic:
   - Virtualizable so that guest or host drivers look the same
   - So we can run other OSs
3. Wide coverage: work across segments from mobile to enterprise and anywhere in between
We are not really there 😞 but there are opportunities 😊

- PSCI fixes this for core power management, but not performance management or peripheral device management
- ACPI covers many of the gaps in ACPI but has low level of adoption outside enterprise

- Everybody has to implement the same functions
- In many segments everybody is doing this in similar ways
  - There is a strong trend to use system level controllers, all with very similar interfaces

- ARM has published the Power System Control Architecture (PSCA) which covers the use of system controllers with ARM IP
SCMI system control and management interface

- SCMI is an extensible interface covering performance and power functions.
- Aim is to create an interface that can be driven from ACPI, or directly from kernel.
- Builds on strong trend in the industry towards µC based platform controllers.
- Candidate version for review:
SCMI structure

SCMI is split into two layers:

- **Protocols**: The services you want to provide:
  - Device Power/Perf management
  - Core perf management
  - Sensor management

- **Transports**: The mechanism by which the service is communicated between the OS and system controllers
## Protocols examples

<table>
<thead>
<tr>
<th>Discovery protocol</th>
<th>Power domain protocol</th>
<th>Performance domain protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PROTOCOL_ATTRIBUTES&lt;br&gt;• PROTOCOL_VERSION&lt;br&gt;• PROTOCOL_MESSAGE_ATTRIBUTES&lt;br&gt;• DISCOVER_VENDOR&lt;br&gt;• DISCOVER_SUB_VENDOR&lt;br&gt;• DISCOVER_IMPLEMENTATION_VERSION&lt;br&gt;• DISCOVER_LIST_PROTOCOLS&lt;br&gt;• DISCOVER_AGENT</td>
<td>• PROTOCOL_ATTRIBUTES&lt;br&gt;• PROTOCOL_VERSION&lt;br&gt;• PROTOCOL_MESSAGE_ATTRIBUTES&lt;br&gt;• POWER_DOMAIN_ATTRIBUTES&lt;br&gt;• POWER_STATE_GET&lt;br&gt;• POWER_STATE_SET&lt;br&gt;• POWERT_STATE_NOTIFY&lt;br&gt;• POWER_STATE_CHANGED</td>
<td>• PROTOCOL_ATTRIBUTES&lt;br&gt;• PROTOCOL_VERSION&lt;br&gt;• PROTOCOL_MESSAGE_ATTRIBUTES&lt;br&gt;• PERFORMANCE_DOMAIN_ATTRIBUTES&lt;br&gt;• PERFORMANCE_DESCRIBE_LEVELS&lt;br&gt;• PERFORMANCE_LIMITS_GET&lt;br&gt;• PERFORMANCE_LIMITS_SET&lt;br&gt;• PERFORMANCE_LEVEL_GET&lt;br&gt;• PERFORMANCE_LEVEL_SET&lt;br&gt;• PERFORMANCE_NOTIFY_LIMIT&lt;br&gt;• PERFORMANCE_NOTIFY_LEVEL&lt;br&gt;• PERFORMANCE_LIMIT_CHANGED&lt;br&gt;• PERFORMANCE_LEVEL_CHANGED</td>
</tr>
</tbody>
</table>

• Optional Shared Memory for stats

• There are additional protocols to cover sensors, clocks and binary transfer
Transports

- Method by which parameters/results are passed between the caller of the interface (an agent) and the implementer (the platform)

- Have the notion of channels
  - Platform uses these to identify calling agent
  - Different channels depending on who initiates communication, agent or platform

- Today the specification only defines a mailbox transport
  - Uses a shared memory area and doorbell interrupts
  - Supports interrupts or polling
  - Very similar to ACPI PCC
  - This could be extended in the future
Agenda

- What’s this all about
- Requirements of SCMI
- What’s the status
Extensibility and OS agnosticism

- The design is inherently extensible allowing more protocols or transports to be added over time.

- OS agnostic
  - Different OSs same Firmware

- Virtualizable
  - In practice this generally means memory based transport e.g Mailbox

- The current draft of the specification provides a mailbox transport
  - In future an SMC transport could be added if we enable ways of isolating management from security
Wide coverage: How it all comes together: linux - mobile
Wide coverage: How it could all come together: enterprise
Wide coverage: How it could all come together: enterprise
Agenda

- What’s this all about
- Requirements of SCMI
- What’s the status
Status and plans

● There are remaining challenges
  ○ Interaction with ACPI CPPC
  ○ Use of SCMI from ASL

● We are prototyping
  ○ In Juno we have basic power control and performance management working
  ○ We have looked at interaction with CPPC

● We’d like to release first public spec end of this year
  ○ Concentrating on mobile
  ○ Feedback welcome

● We are planning to upstream SCMI code following spec publication
  ○ In linux and ARM TF and SCP reference FW

● Candidate version for review
Thank You

#LAS16
For further information: www.linaro.org
LAS16 keynotes and videos on: connect.linaro.org
Wide coverage: SCMI is not just for application processor cores

- The spec mainly describes communications between AP and platform
- Other system agents could use the same interface, e.g. a modem in mobile or an management controller in enterprise
- Self managed agents must use private channels