arm

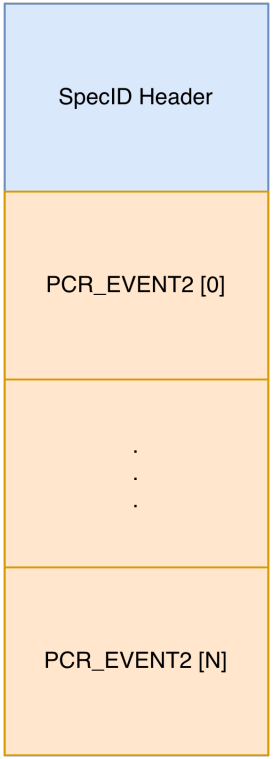
Event Log Library

Lightweight driver for handling TPM Event Logs in Firmware.

Harrison Mutai
26/11/2025 23:28

What is an event log?

- An **event log** is a structured record of security-relevant measurements.
- It is used during **device attestation** to prove that a system booted into a trusted state.

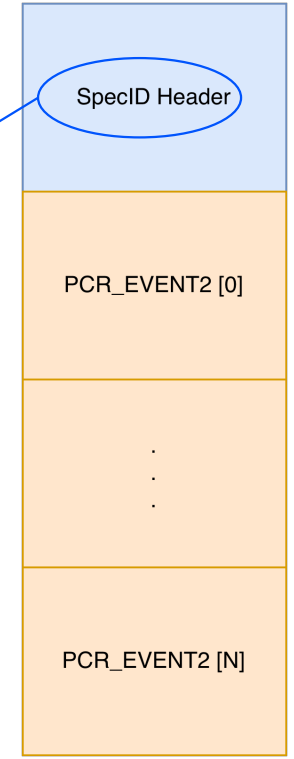


What is an event log?

TCG_EfiSpecIdEvent

- Defines how to parse the rest of the event log.
- Declares supported hash algorithms.
- May also contain optional vendor information, these are OEM-defined and not standard.

```
[NOTICE] TCG_EfiSpecIDEvent:
[NOTICE]   PCRIndex           : 0
[NOTICE]   EventType          : 3
[NOTICE]        Digest        : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[NOTICE]                      : 00 00 00 00
[NOTICE]   EventSize          : 41
[NOTICE]   Signature          : Spec ID Event03
[NOTICE]   PlatformClass      : 0
[NOTICE]   SpecVersion        : 2.0.2
[NOTICE]   UintnSize          : 1
[NOTICE]   NumberOfAlgorithms : 3
[NOTICE]   DigestSizes        :
[NOTICE]     #0 AlgorithmId   : SHA256
[NOTICE]        DigestSize    : 32
[NOTICE]     #1 AlgorithmId   : SHA384
[NOTICE]        DigestSize    : 48
[NOTICE]     #2 AlgorithmId   : SHA512
[NOTICE]        DigestSize    : 64
[NOTICE]   VendorInfoSize     : 0
```

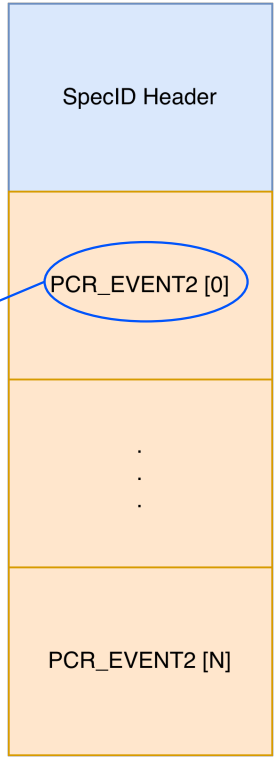


What is an event log?

TCG_PCR_EVENT2

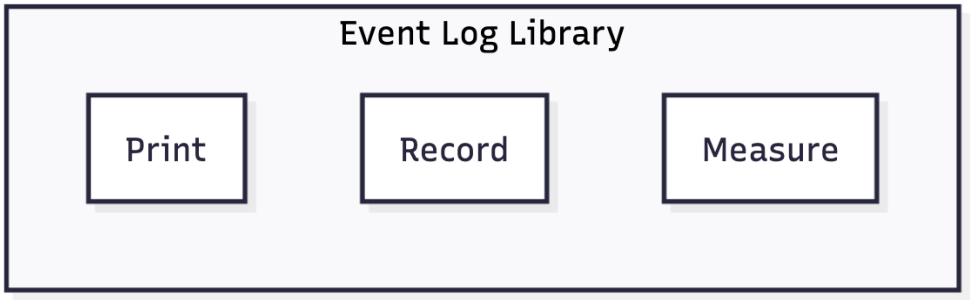
- Records a single measurement event and includes:
  - PCR Index
  - Counted list of tagged digests, each containing:
    - The Algorithm ID
    - The Digest value computed over the event data

```
[NOTICE] PCR_Event2:
[NOTICE]   PCRIndex           : 0
[NOTICE]   EventType          : 1
[NOTICE]   Digests Count      : 3
[NOTICE]     #0 AlgorithmId   : SHA256
[NOTICE]        Digest        : 28 3b 0a 75 2b 88 1b 5f 50 0f ad b5 90 10 e6 3d
[NOTICE]                      : 1b 66 4f 8b da 2d bf 33 cb 6b e2 1c 8e b3 ec a9
[NOTICE]     #1 AlgorithmId   : SHA384
[NOTICE]        Digest        : d9 d5 bf 14 4c 08 e9 57 7e d0 d1 e5 e5 60 87 51
[NOTICE]                      : 09 b3 40 98 05 80 47 3d bc 2e 68 9a 3b e8 38 e7
[NOTICE]                      : 7a 0a 33 48 fe 96 0e c9 bf 81 da 36 f1 86 8c a5
[NOTICE]     #2 AlgorithmId   : SHA512
[NOTICE]        Digest        : d2 47 88 fa 4c 0c 77 8b f0 d1 23 14 28 54 95 63
[NOTICE]                      : 65 16 cf 40 86 1b 3d 73 7f d3 5d bb 59 1c 5b 5d
[NOTICE]                      : 25 91 6e b1 d8 61 76 b1 4e 0e 67 d2 d0 39 57 f0
[NOTICE]                      : cf 6c 87 83 4b f3 28 54 05 88 36 0b a7 c7 c5 f8
[NOTICE]   EventSize          : 8
[NOTICE]   Event              : IMAGE_1
```



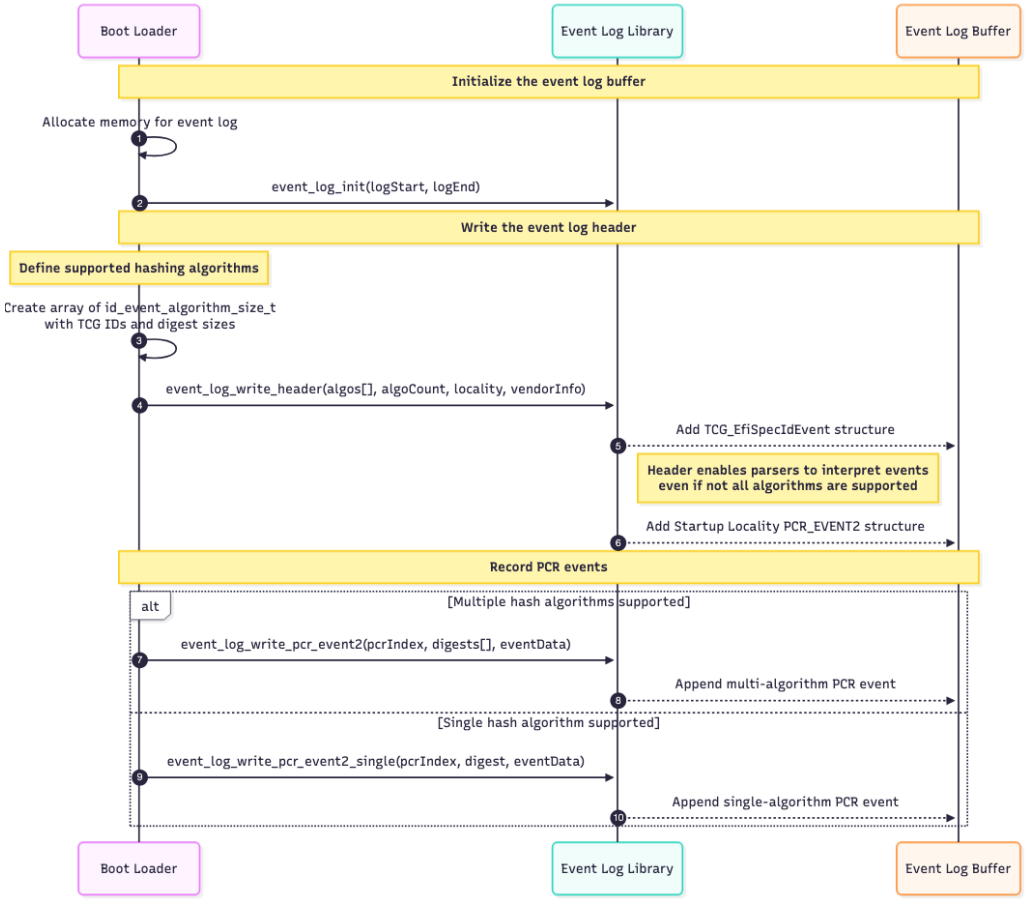
Event Log Library Capabilities

- The library provides a high-level interface for creating and managing TPM event logs, enabling firmware and software components to:
  - Initialize and extend a TPM event log directly in memory.
  - Record key event types including SpecID, PCR_EVENT2, and Startup Locality.
  - Use multiple hashing algorithms (e.g., SHA-256, SHA-384, and SHA-512) for flexible measurement policies.
  - Measure and log firmware components or configuration data.
  - Print the event log for validation, and attestation replay.



Integration Into Platform Builds

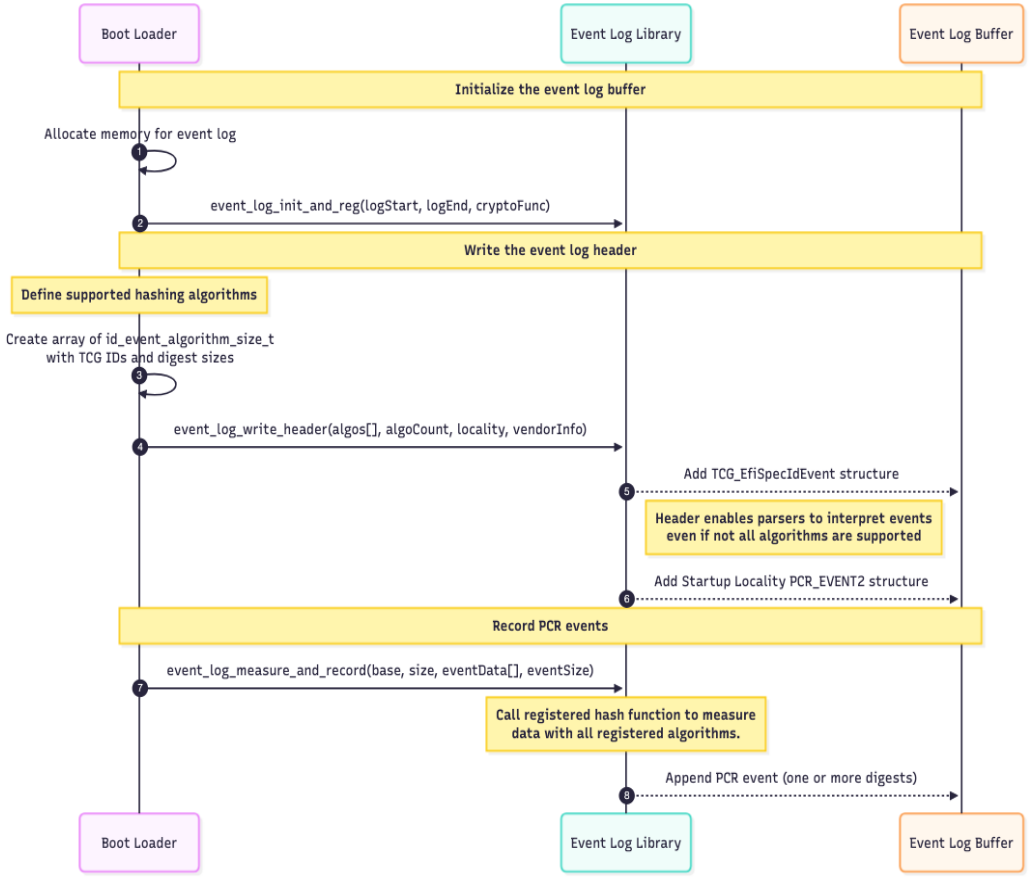
1. **Initialize the event log buffer**
2. **Define supported hashing algorithms**
3. **Write the event log header**
4. **Record PCR events**



Integration Into Platform Builds

1. **Initialize the event log buffer**
2. **Define supported hashing algorithms**
3. **Write the event log header**
4. **Record PCR events**

With measurement APIs…



Repository Location & Testing Workflow

- The project is hosted on a Gerrit review server, with a mirrored read-only copy available on GitHub for broader visibility.
  - Gerrit: https://review.trustedfirmware.org/plugins/gitiles/shared/libEventLog
  - GitHub Mirror: https://github.com/TF-Shared/event-log-library/
- All submitted patches go through a Jenkins-based CI pipeline, which performs:
  - Code formatting checks
  - Build matrix testing across Clang, GCC, and Arm GNU toolchains
  - 32-bit and 64-bit build verification
- On TF-A side we run emulated remote attestation tests, which:
  - Parse the textual event log dump
  - Recompute digests manually
  - Compare recomputed values with logged digests to ensure correctness

Repository Location & Testing Workflow

Long term goals

- Integrate static analysis tools (e.g., cppcheck) into the CI workflow
- Unit testing
- MISRA compliance automated rule checking
- Deploy hardware-based testing using Raspberry Pi boards connected to a LAVA lab, enabling real-hardware validation of event logging and attestation flows

Roadmap & Community Contributions

- Patches are currently under review to formalize a multi-digest event format, enabling each event to carry more than one digest.
- Supporting patches are being prepared for several platforms to make use of the new multi-algorithm format, including:
  - Raspberry Pi
  - QEMU
  - NXP i.MX
  - Arm platforms (Juno, FVP, etc.)
- These patches enable platforms to declare and measure using multiple algorithms during boot.

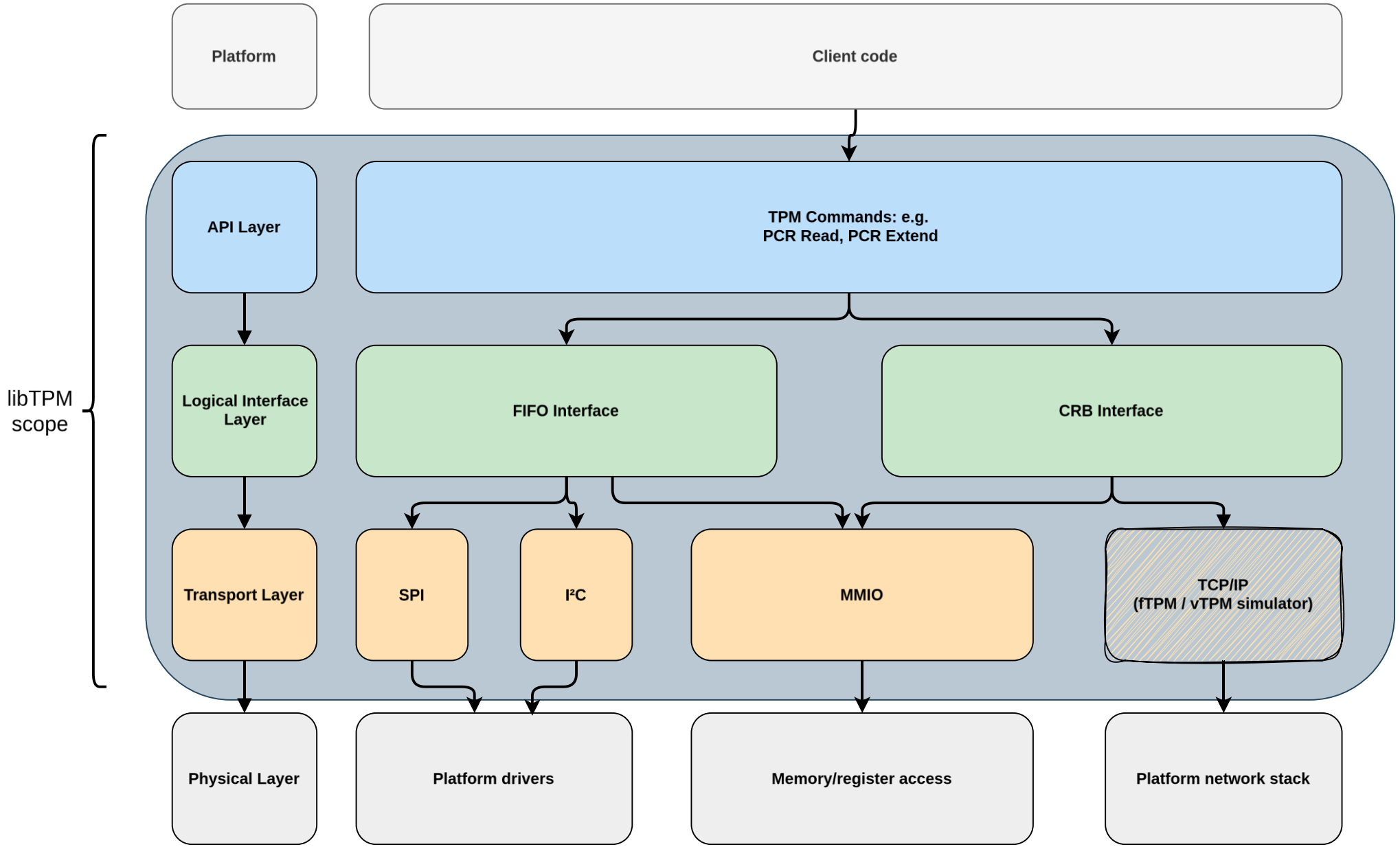
arm

TPM Library

Refactored from TF-A

Matthew Ellis
27 November 2025

TPM Host Interface Architecture



Why roll our own?

Survey of existing TPM libraries

Library / Source	Memory Model	OS / Kernel Coupling	Stack Level	Licensing / Support	Notes
wolfTPM	❌ Requires `malloc`	➖ Minimal OS coupling	✓ Transport + upper layers	✓ Commercial support GPL3	Lightweight but not static-memory friendly
Linux TPM Driver	❌ Requires dynamic allocation	❌ Strong kernel integration	✓ Transport	✓ Maintained GPL	Not portable outside Linux kernel
FreeBSD TPM	❌ Requires dynamic allocation	➖ Some OS dependencies	✓ Transport	✓ Maintained BSD	Less entangled than Linux but still not firmware-ready
Zephyr TPM2 TSS Module	✓Static	✓ RTOS-integrated	✓ Transport	❓ Community support, unknown	Unknown long-term viability and license
Most TPM2 Software Stacks	❌ Heavy dynamic memory	❌ User-space assumptions	❌ Upper TPM layers	❓ Varies	Not designed for firmware integration

Development and testing hardware

Raspberry Pi 3B – Advantages

Platform

- Well-supported ecosystem
- Stable tooling
- Easy UART/serial access
- Runs Linux with TPM driver

Hardware Practicality

- Small, low-power
- Easy to rack in multiples
- SD card switch
- Off the shelf TPM modules

Question – Will it support more than one TPM?

Do We Need It?

- TF-A doesn't require multi-TPM support
- But system use cases exist:
  - Mission-critical failover TPM
  - Telecom partitions with independent roots of trust

Architectural Implications

- Build-time vs run-time TPM selection
- Global state vs per-instance context
- Effects on API design and build system

Soft issues

- Library adoption and fragmentation
- Code complexity and maintenance
- Long term support and testing burden