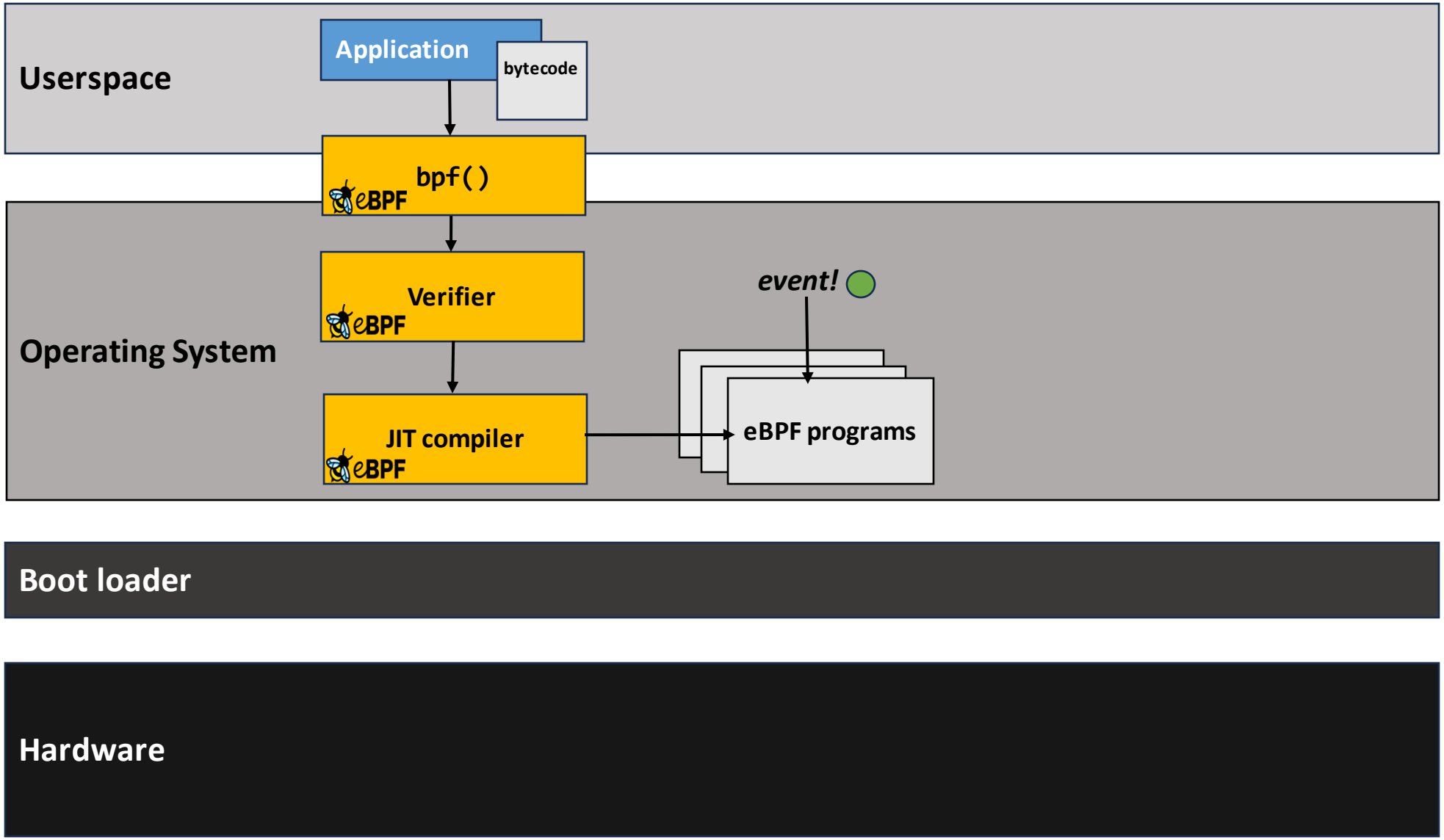
# Extending Non-Repudiable Logs with eBPF

Avery Blanchard[1], Gheorghe Almasi[2], James Bottomley[2] and Hubertus Franke[2]

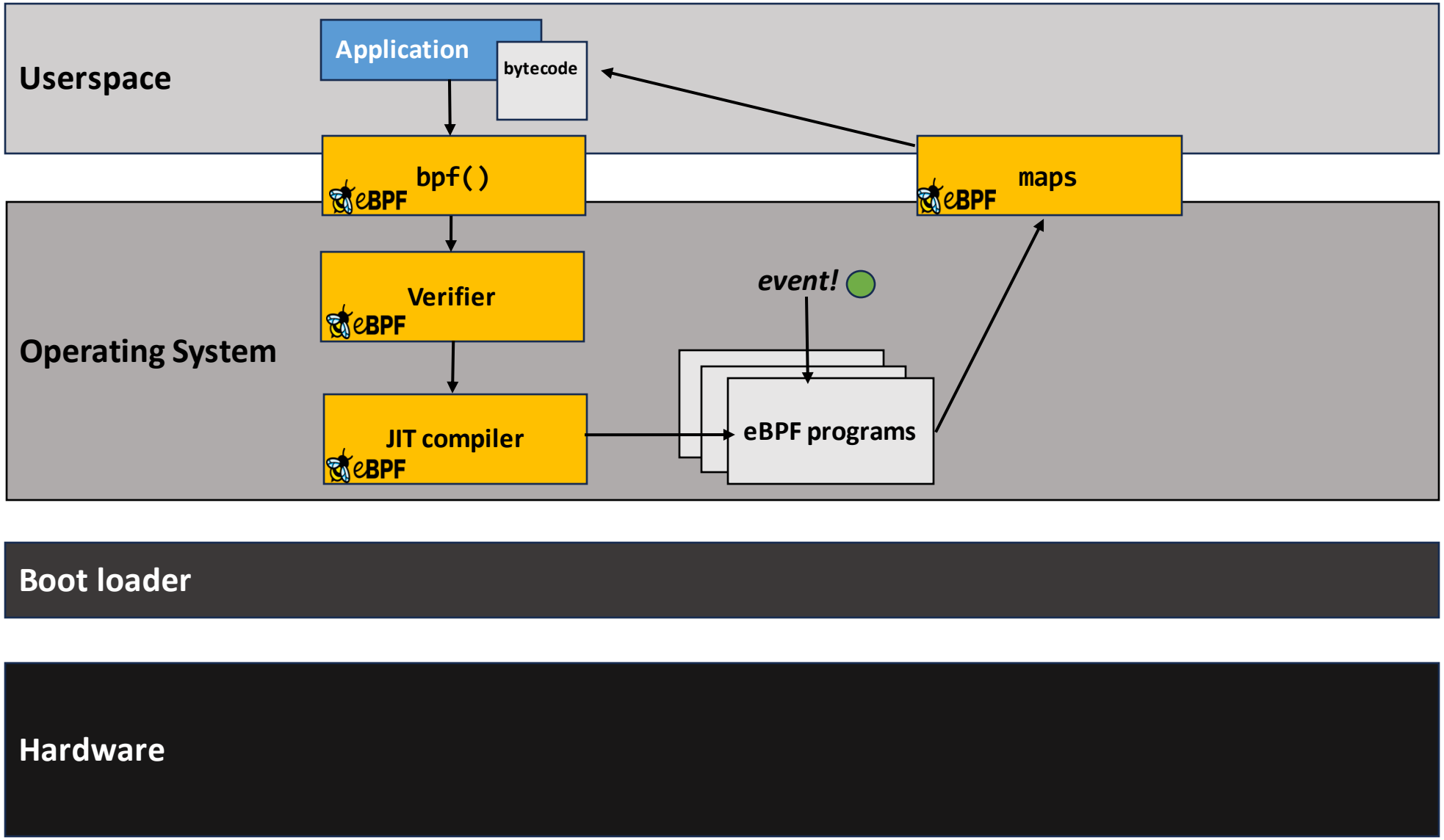
1 Duke University
2 IBM Research

November 13th, 2023

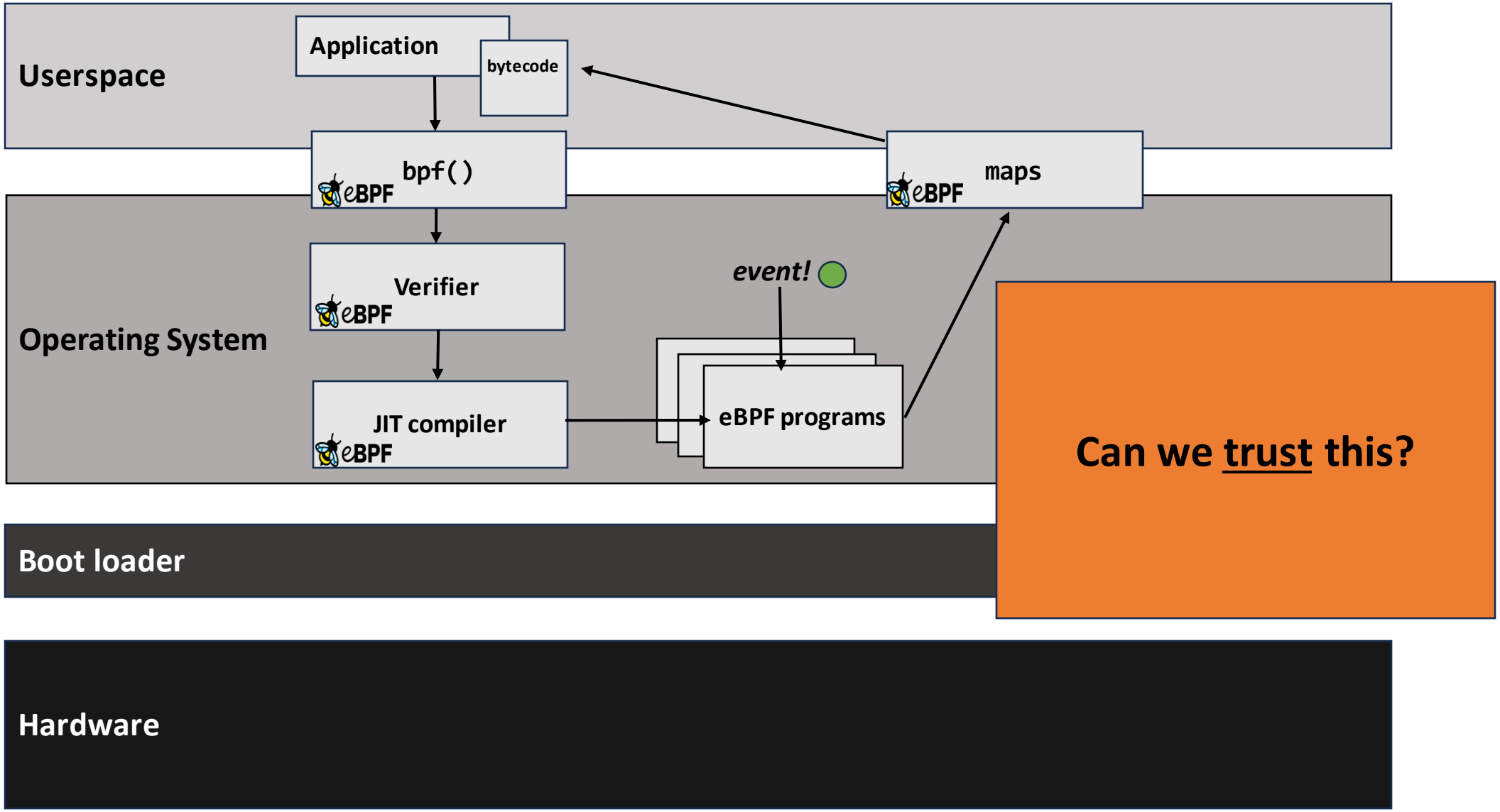
# Visibility into System State with eBPF



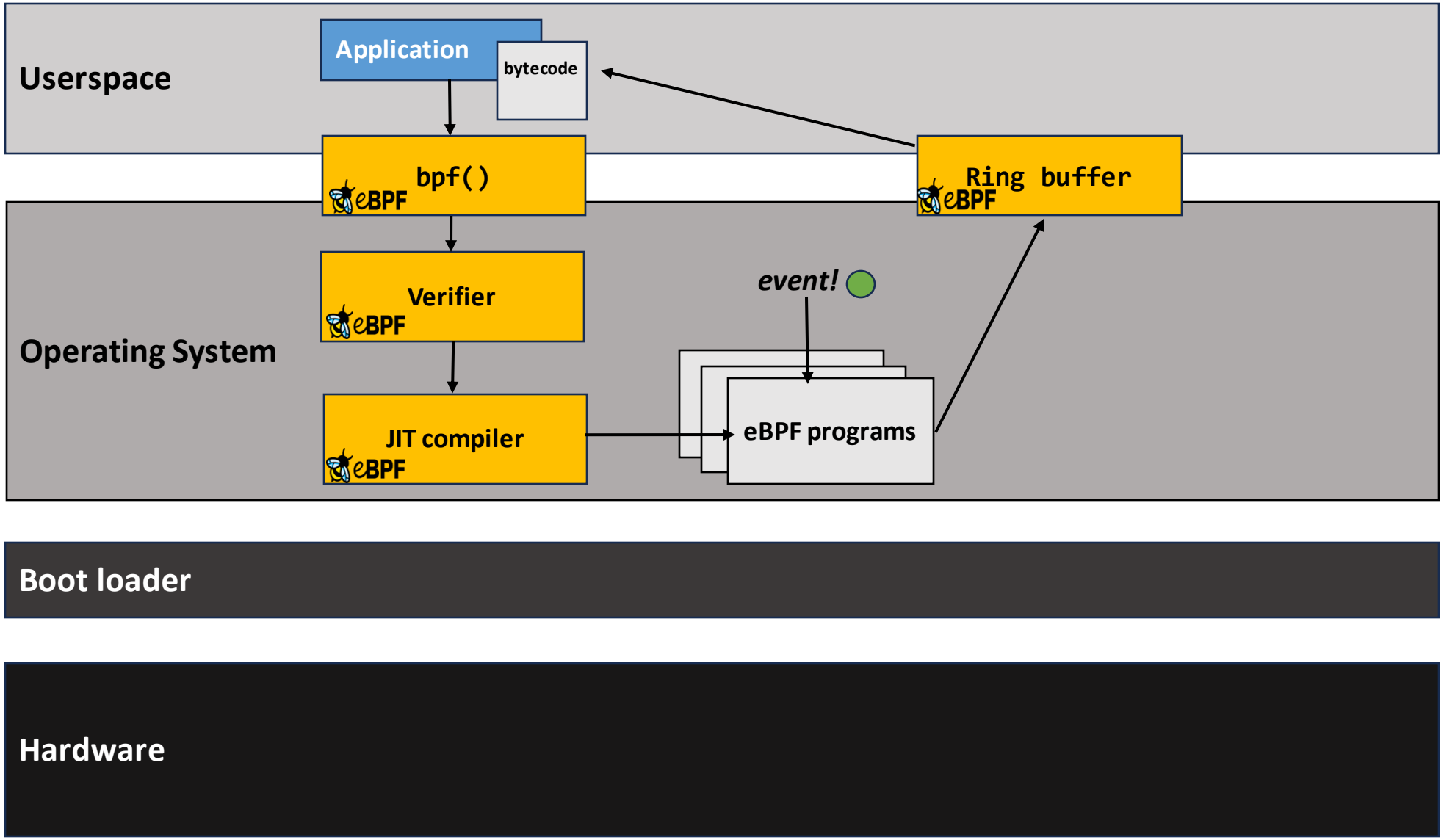
# Visibility into System State with eBPF



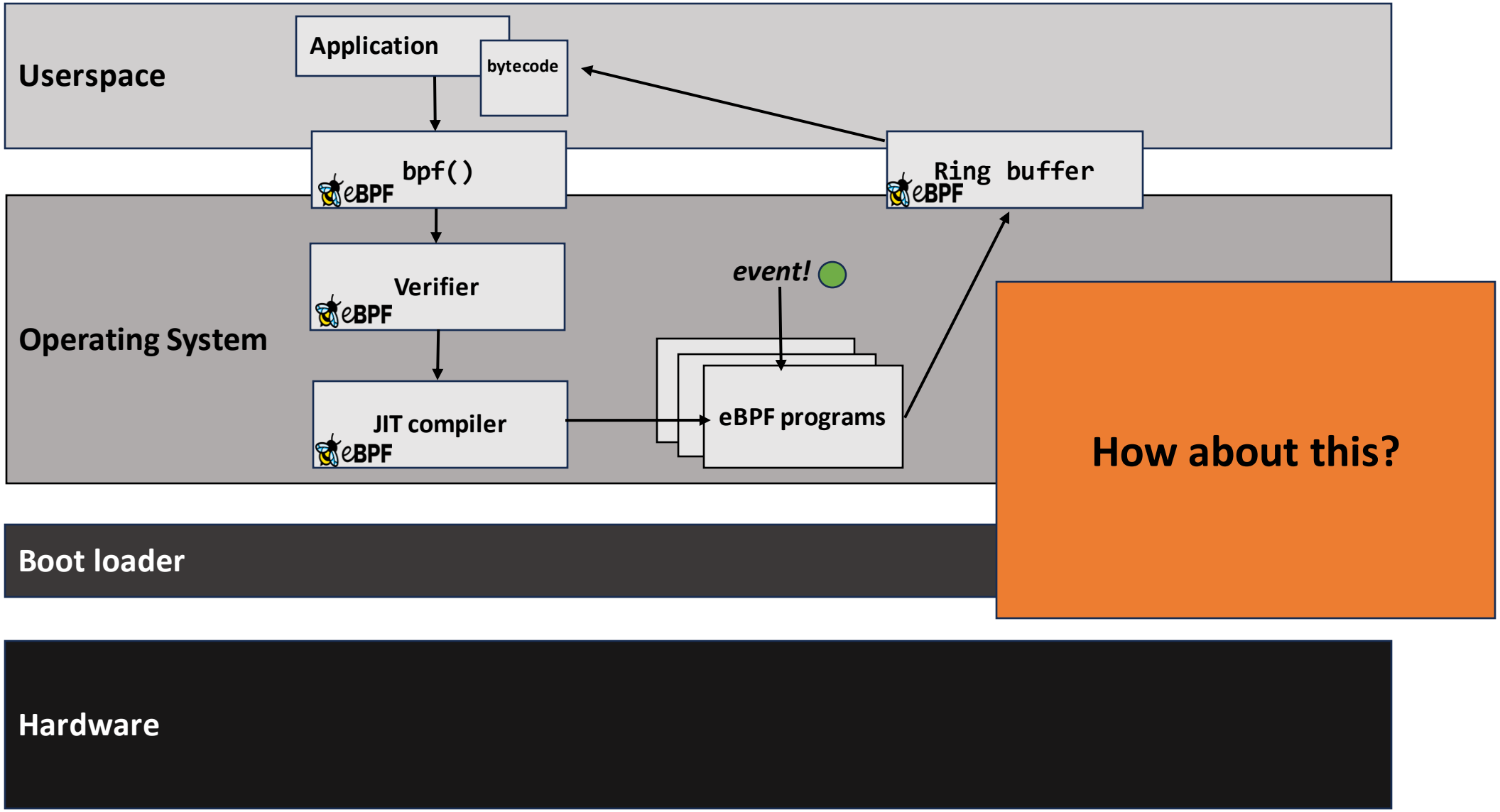
# Logging System State



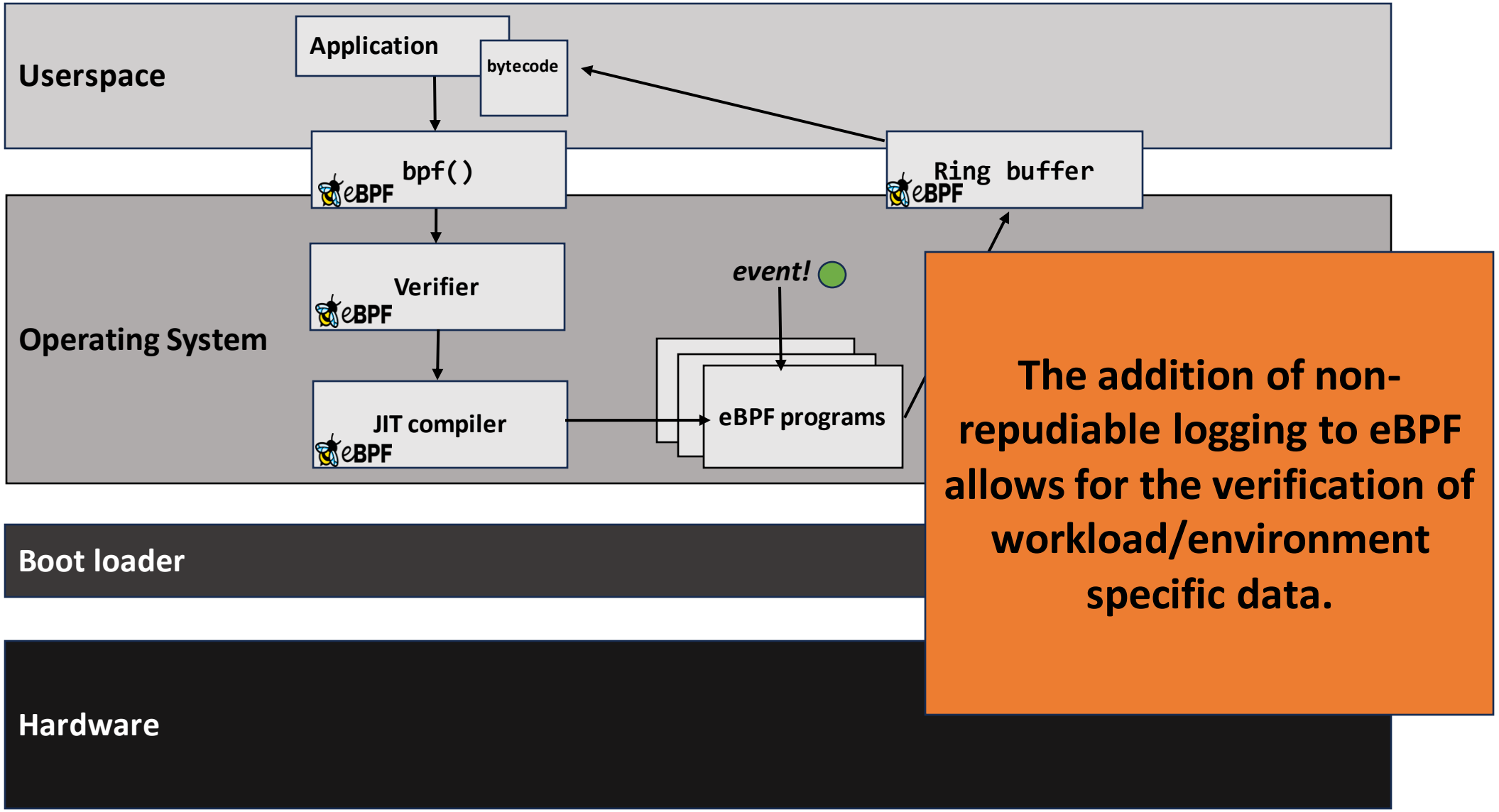
# Visibility into System State with eBPF



# Logging System State

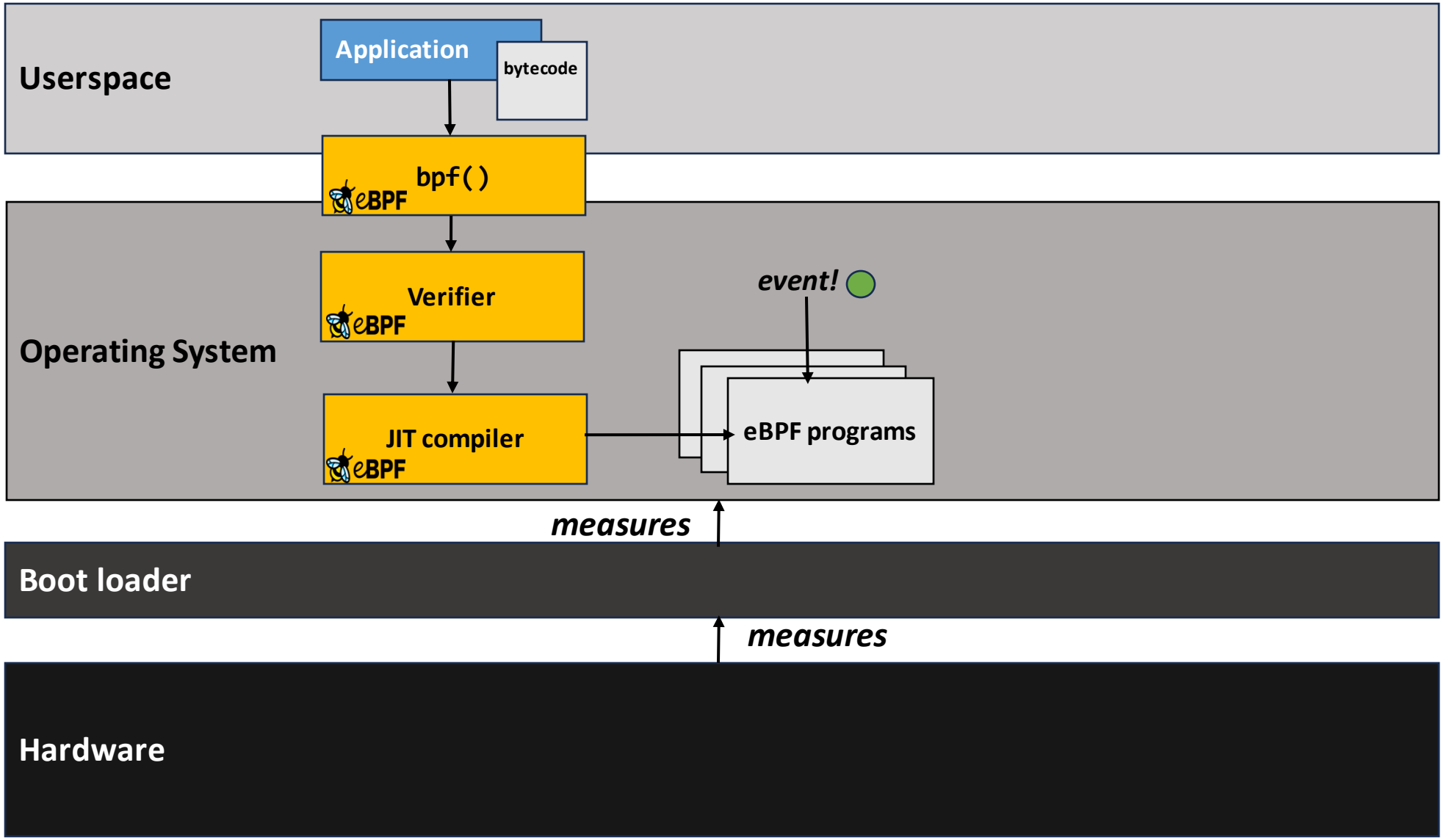


# Non-repudiable Logging

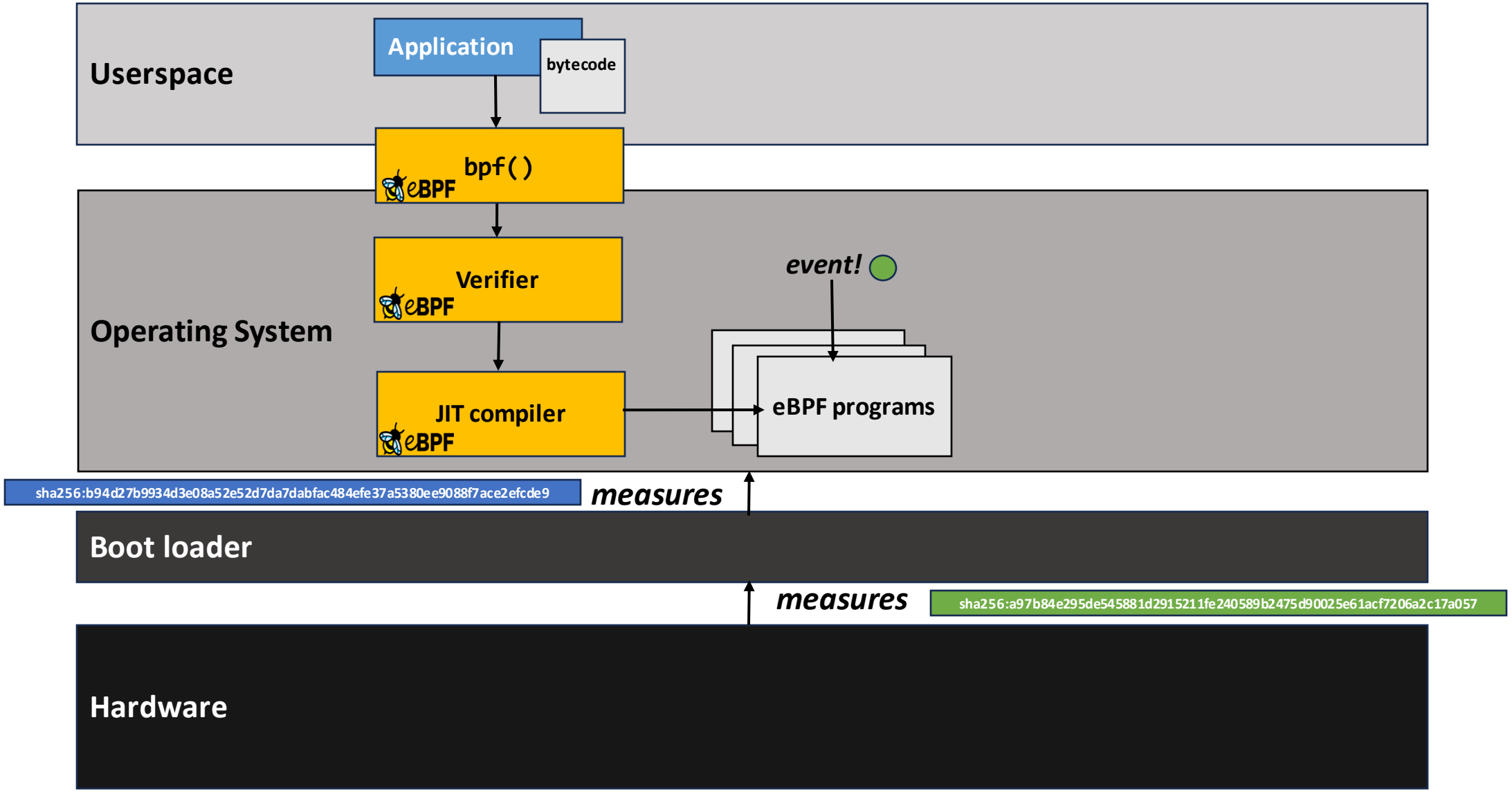



**The addition of non-repudiable logging to eBPF allows for the verification of workload/environment specific data.**

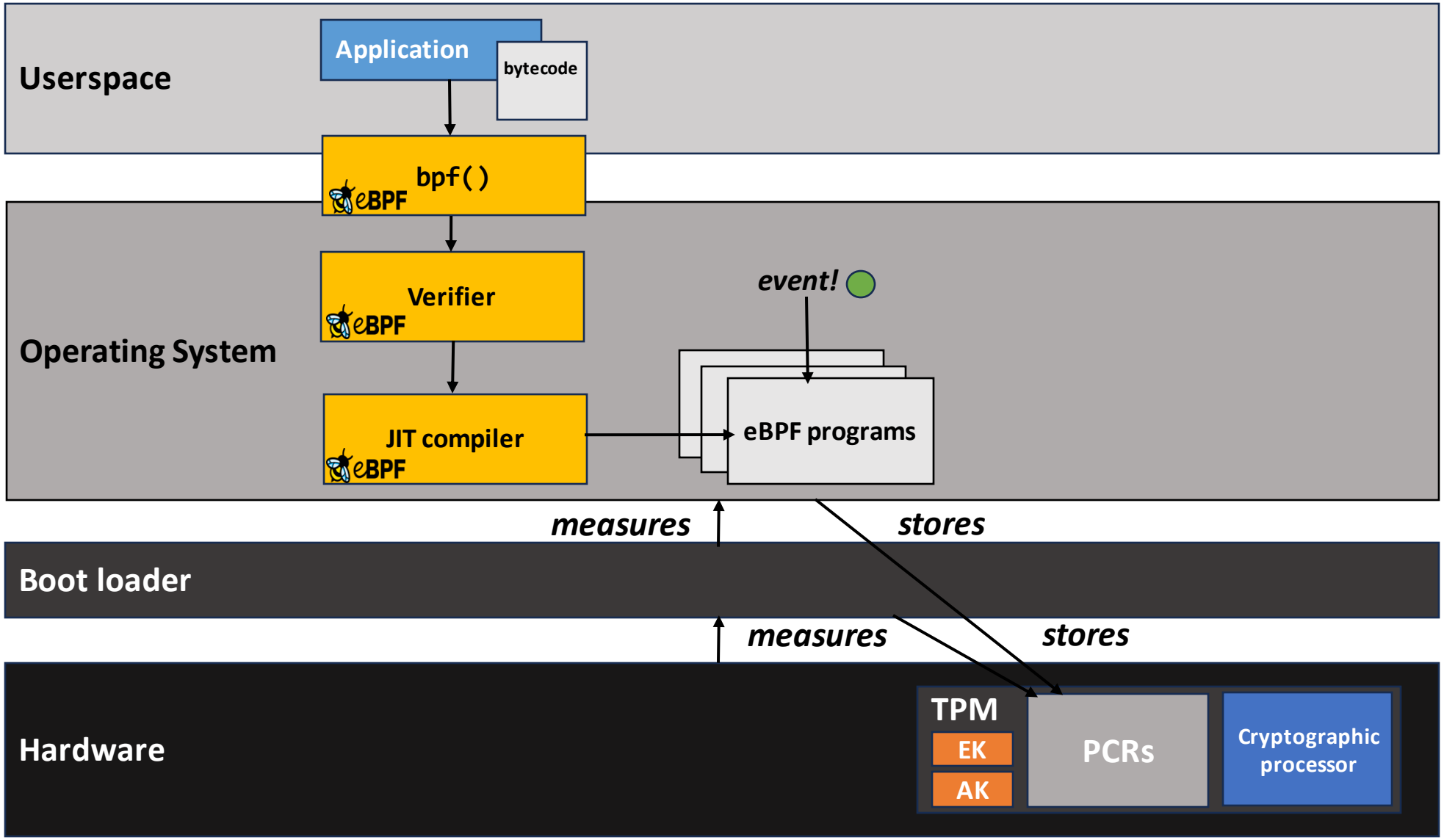
# Building a Chain of Trust



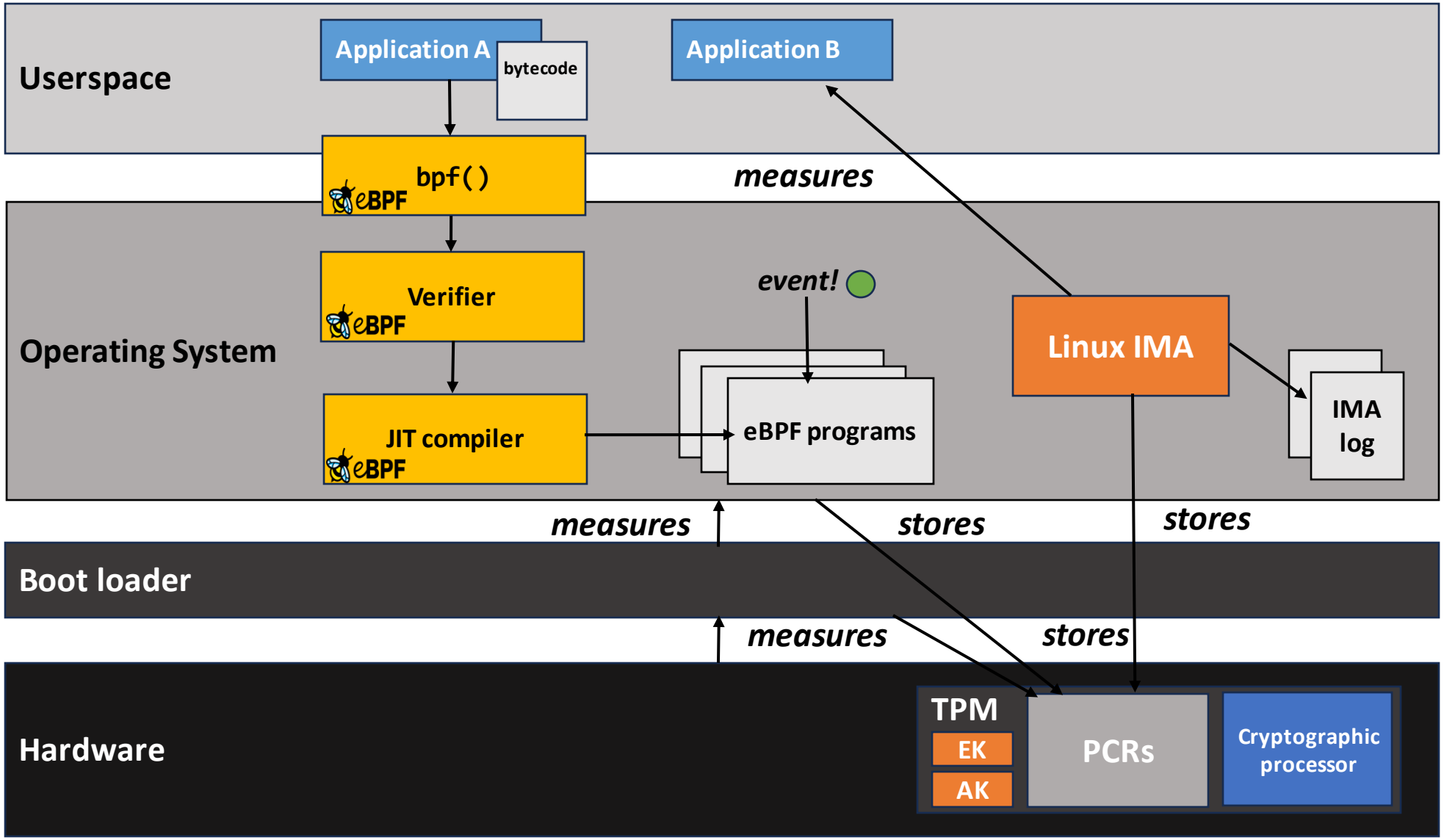
# Building a Chain of Trust



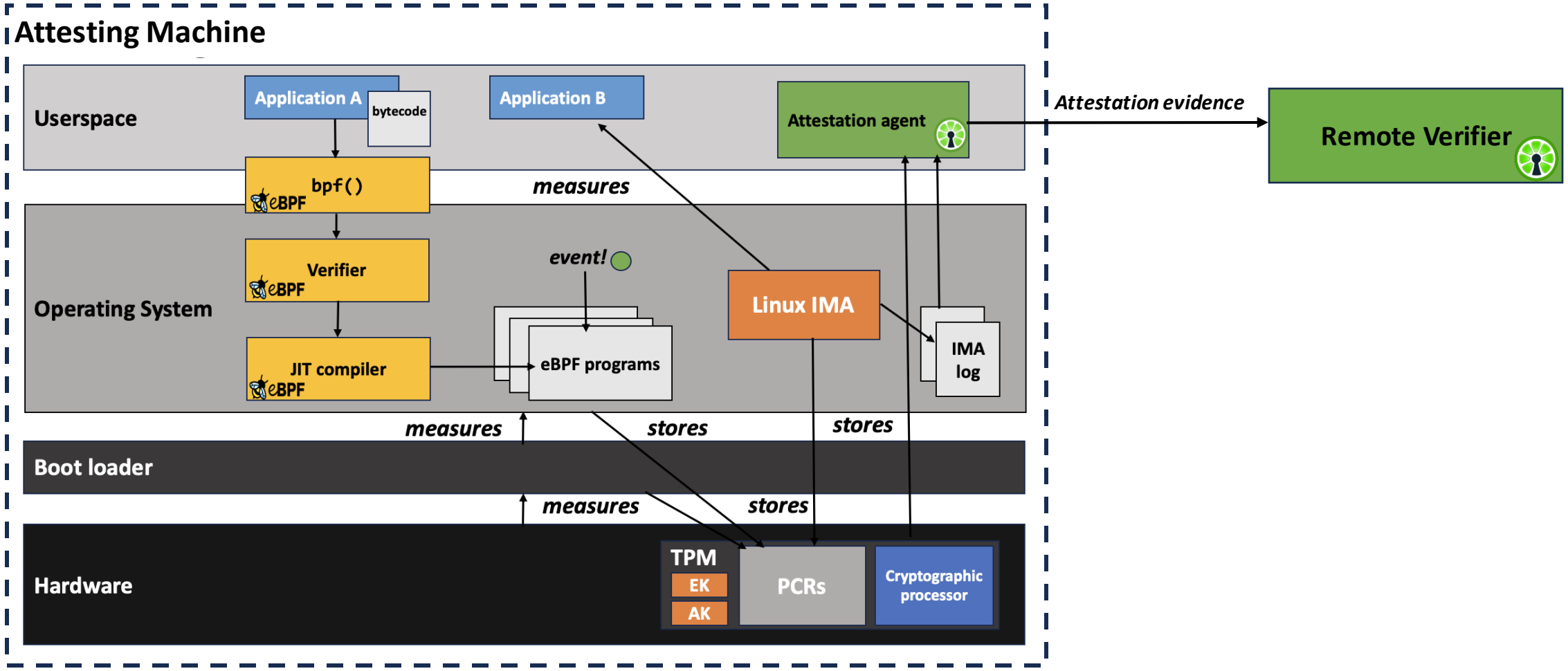
# Rooting Trust in Hardware



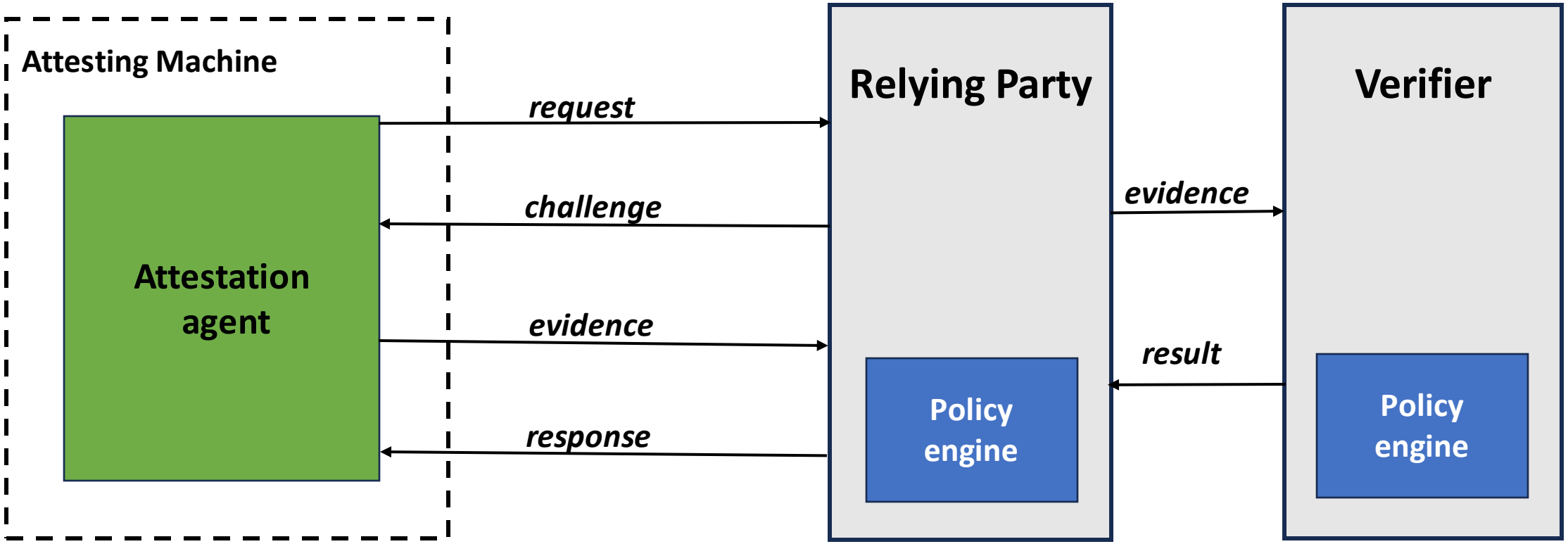
# Extending Measurements Through Runtime



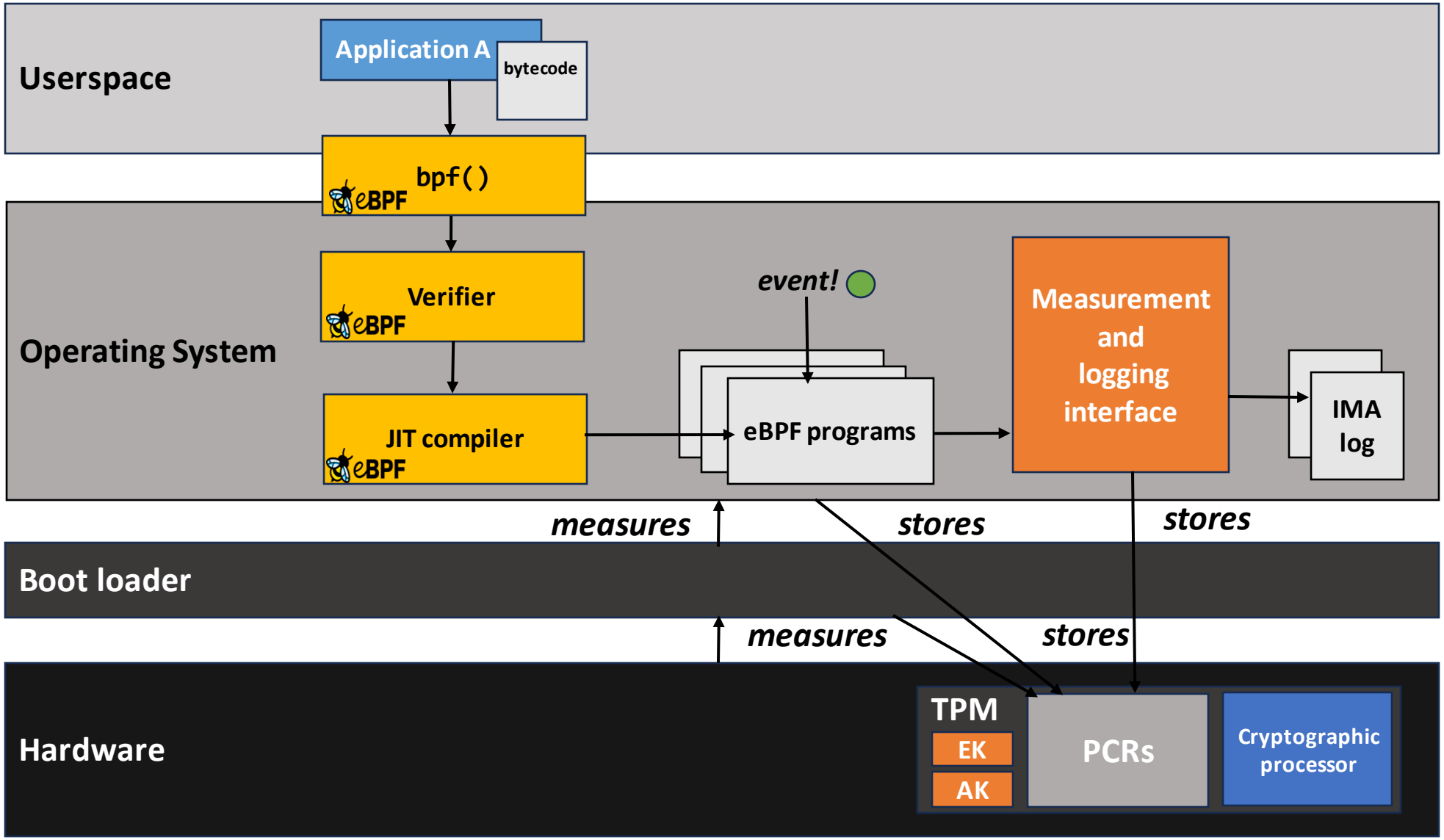
# Building Trust in Environments



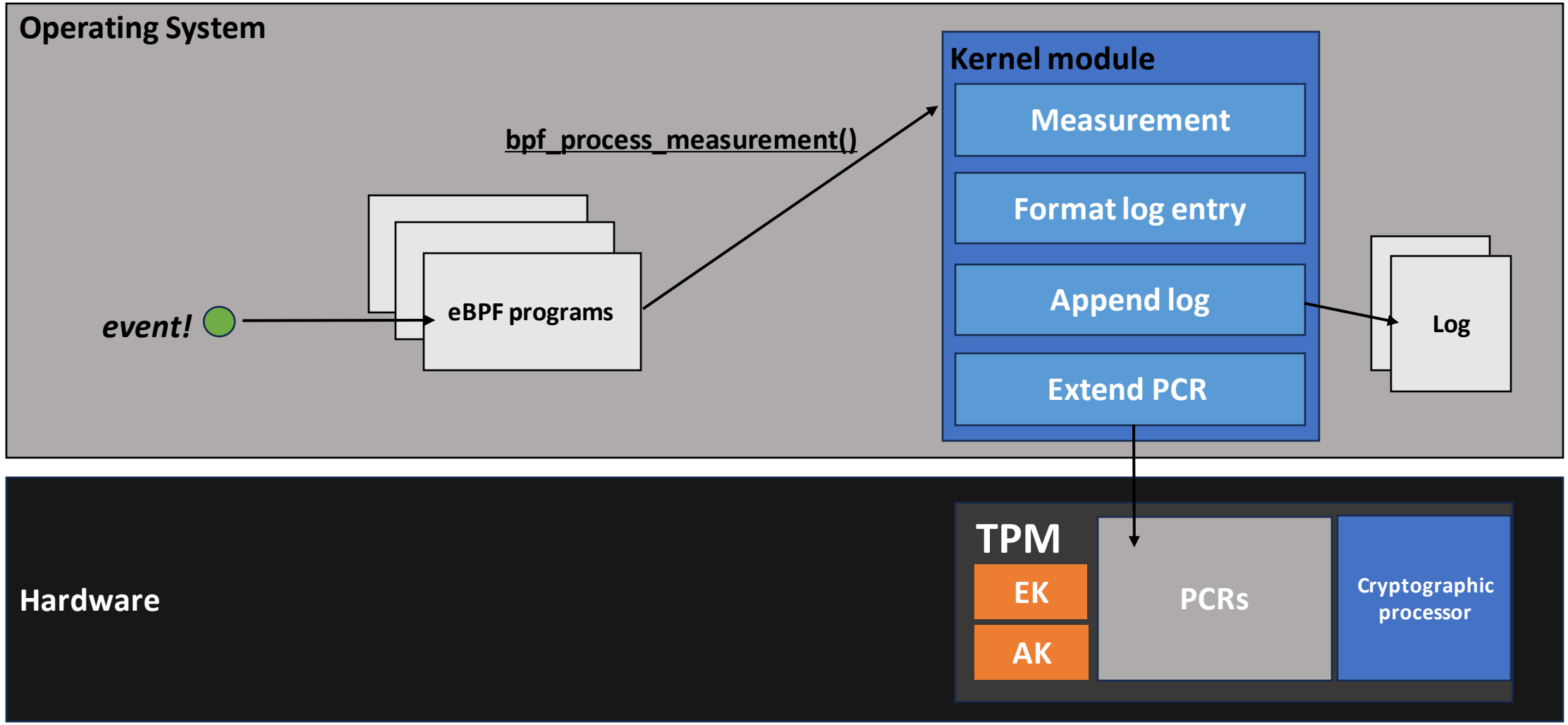
# Attesting System Properties



# Non-repudiable Logging in eBPF Programs



# Measurement Interface

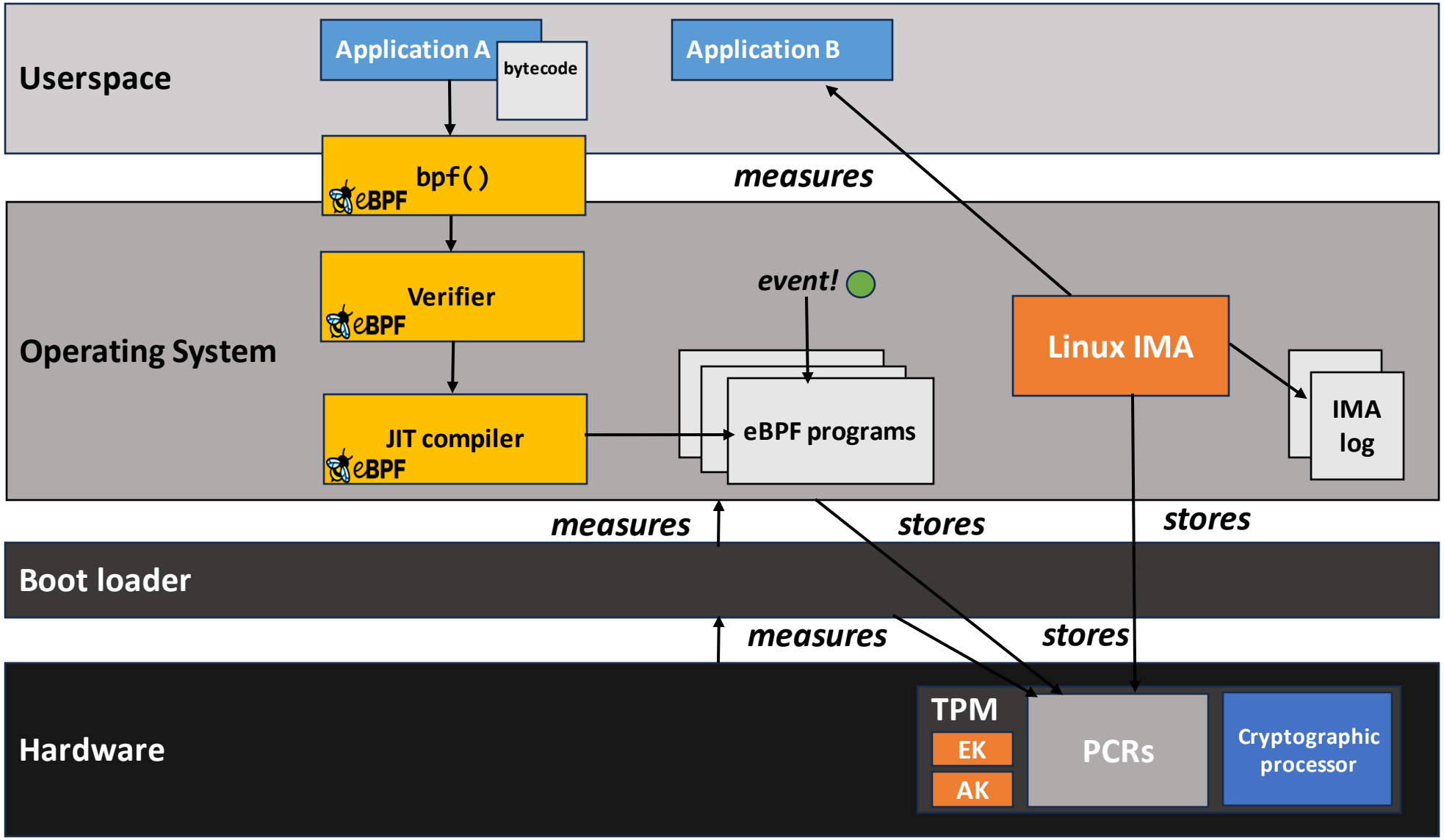


# From the eBPF side

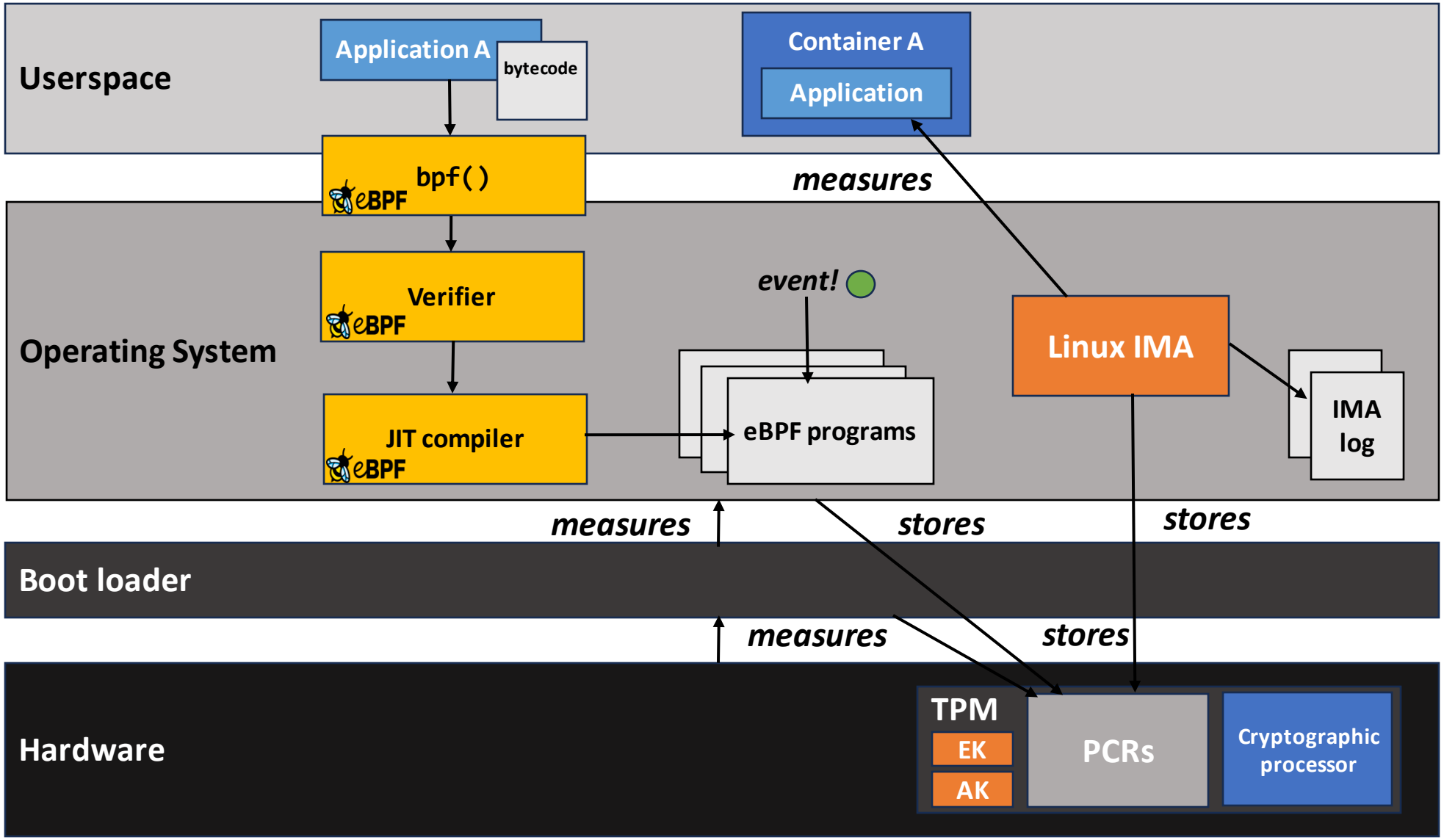
- Available to sleepable eBPF programs
- Programs can provoke the measurement and storage of formatted data and files

```
struct ebpf_data {
        struct file *file;
        unsigned int ns;
};

extern int bpf_process_measurement(void *, int) __ksym;
extern int measure_file(struct file *) __ksym;

SEC("lsm.s/mmap_file")
int BPF_PROG(mmap_hook, struct file *file, unsigned int reqprot,
               unsigned int prot, int flags)
{
    struct task_struct *task;
    u32 key;
    unsigned int ns;
    int ret;

    if (!file)
        return 0;

    if (prot & PROT_EXEC || reqprot & PROT_EXEC) {

        task = (void *) bpf_get_current_task();
        ns = BPF_CORE_READ(task, nsproxy, uts_ns, ns.inum);

        struct ebpf_data data = { .file = file, .ns = ns };

        ret = bpf_process_measurement((void *) &data,
                        sizeof(&data));


    }


    return 0;

}
```

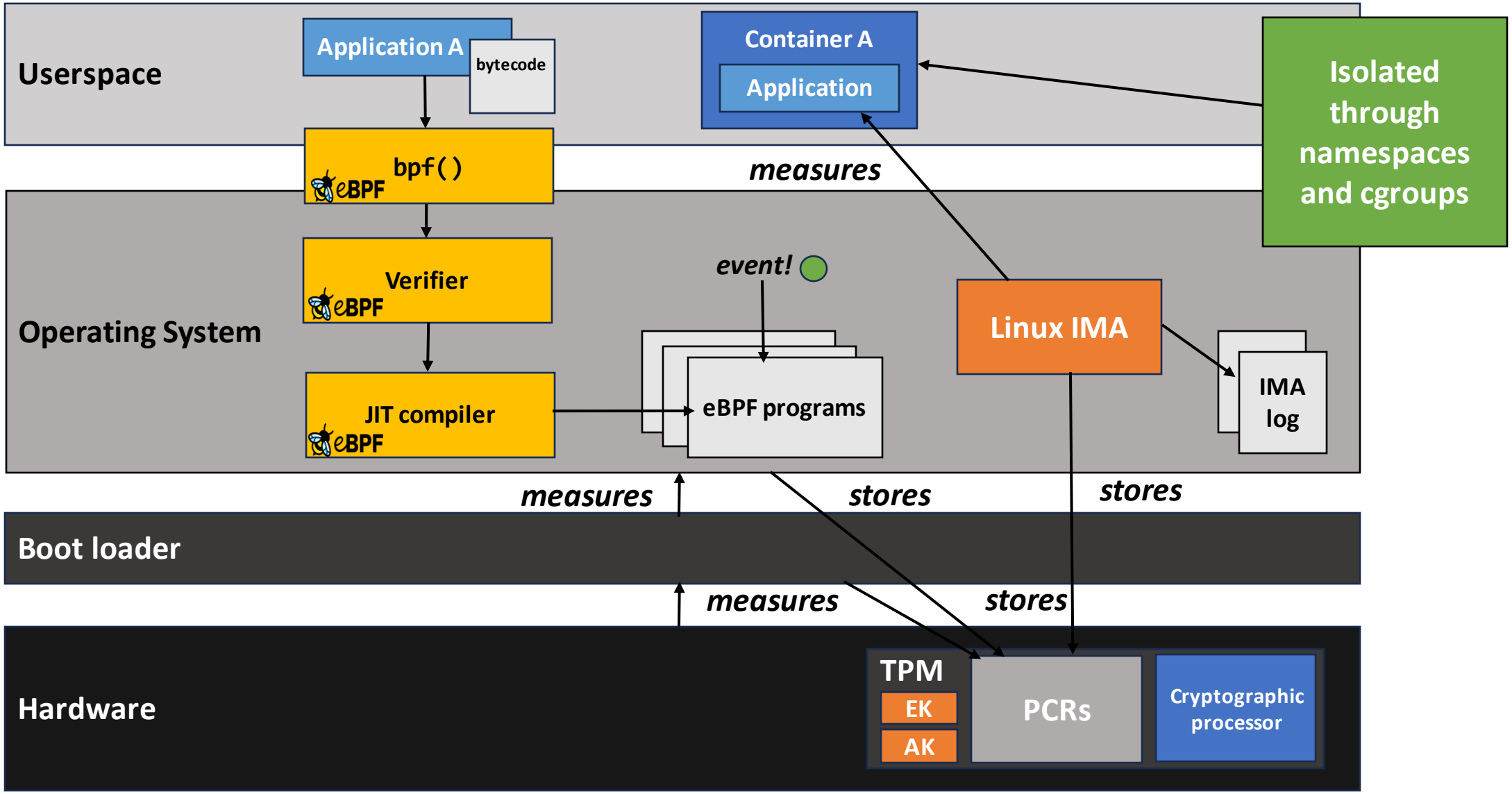
# Example Use Case



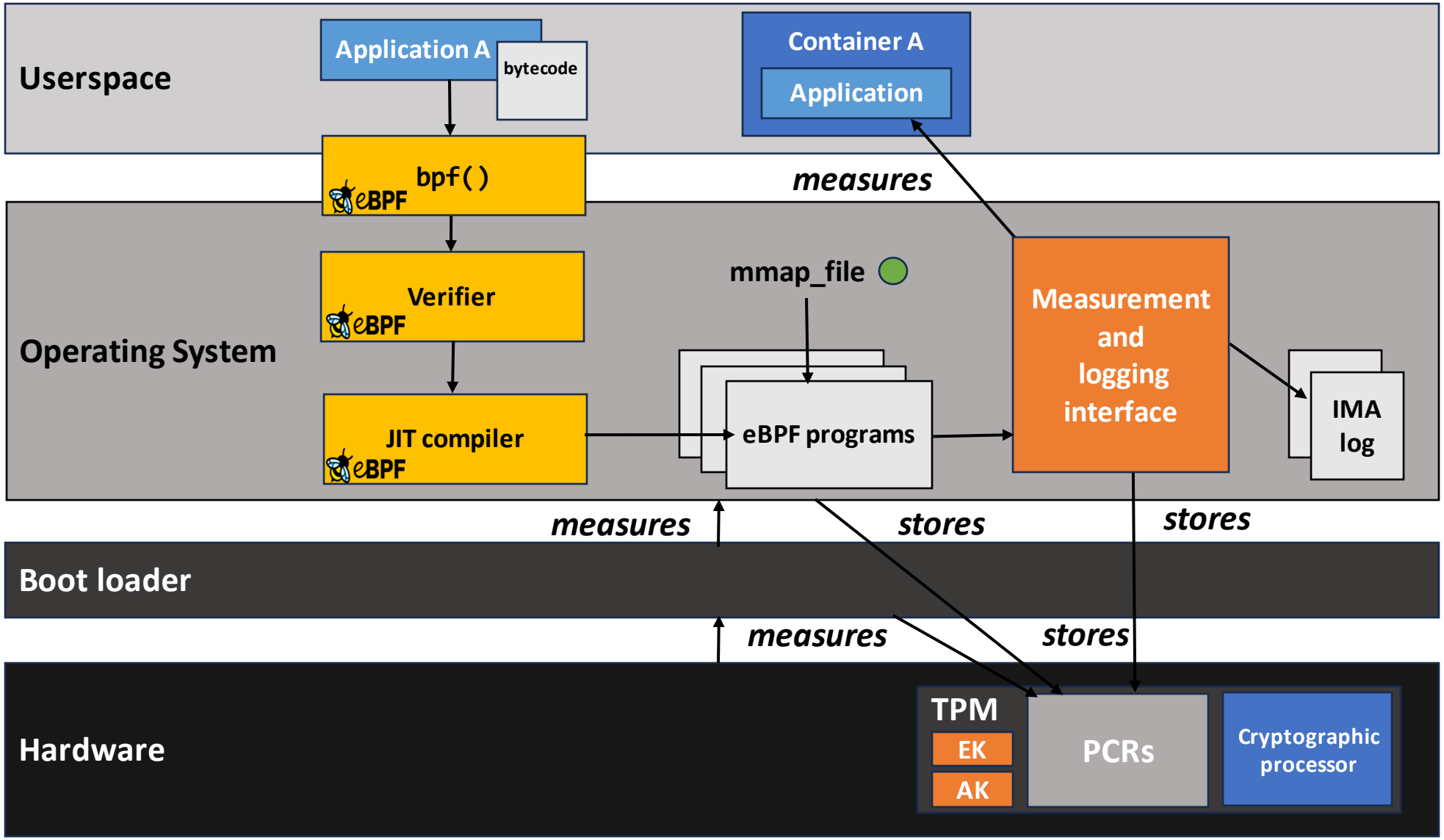
# Extending Linux IMA to Containers



# Extending Linux IMA to Containers



# Adding Namespace Support to IMA



# Resulting IMA Log

```
[[avery@fedora container-ima]$ sudo tail -n 25 /sys/kernel/security/ima/ascii_runtime_measurements
11 56c23fc8baf7155d5e50797c55b0a0436eb4f1e0 ima-ng sha256:5353978a6cc92dad314d0ef0bfbdec1f84241b77aa12b06aba221893fee8c728 4026532423:/usr/lib64/libpcre2-8.so.0.11.2
11 cf5f9b53ad9292dd5759fcccc49a643442deb583 ima-ng sha256:0b50ab927bd942ef6a7434e16f8819caedcdbfc5445176a7eba6990d2cb2e233 4026532423:/usr/lib64/libcap-ng.so.0.0.0
10 5fb55ea1517fb800aea13744cf6f8b9b366ebf0d ima-ng sha256:0efb18f93ab7c680ba28ba9bc50c3fabed9fa49d22e18a76b381039cfd01d4cf /usr/bin/containerd-shim-runc-v2
10 5c4f7bddfc3988228ca9d50629d524b4dacb1454 ima-ng sha256:d3f7d10e296e5c626ea78539cb38cd8dffd043bea3627da35ce3b20c0ac68014 /hello
11 eec36e004d3dd397484f9302bfc209b5e98a91b7 ima-ng sha256:aa15dcbe503ee00f9f72924e8bea3b0c9bd42ca5205c40e70dde9d7c963e56e0 4026532981:/hello
11 f6422a7bd7c8cdab8de4130492ecc3b88918447a ima-ng sha256:0458c4bf9471e7b2083b4fc1d3f8b7632b2b2fe1f54fbaa2b43d8b37b1e53690 4026532896:/usr/bin/bash
11 bf122367ae1321e9d50a35a9578fd9c78f6af526 ima-ng sha256:21d54feee92cd42390a9fa151f8950813ecd5eaf8607b3e353aa4742a3cff08d 4026532896:/usr/lib64/ld-linux-x86-64.so.2
10 bfc24c544fc3f85107c65e69856f43cc54ef72b4 ima-ng sha256:415a5f6f063d6b6c0183947708651e5424b3c38d68357fd23103ad7c56a2a3cf /usr/lib64/ld-linux-x86-64.so.2
11 828866f89825ac2b847c97b5189a74ba38151729 ima-ng sha256:8046a139aa8590f8004da1319b64c09194596dd5a0abf59020a8568e9b6d61f1 4026532896:/usr/lib64/libtinfo.so.6.4
11 c79143cc2fc5bf4780b9a35a02acaad81e9a28f7 ima-ng sha256:be13ff2194f060dff73c96f317d16e3a2c380ba3beedc1b485af866b3b7e4729 4026532896:/usr/lib64/libc.so.6
10 7393b51a9384e4254a9c8df419309a41fac0d5f5 ima-ng sha256:9f2e4d479a9a7d7e27e639701da4ffbcfb8ff40512b5c676ac42801058847c28 /usr/lib64/libc.so.6
11 3f81e07160ed9c5a0306d04070f222e73e25f405 ima-ng sha256:e5fef662f3f426b94c10d288a7ce06caa14bea7b2452976232b8d20b99d3c61e 4026532896:/usr/bin/grep
11 35d240dbf1f63cef20c3a374c2f02055da7477e0 ima-ng sha256:5353978a6cc92dad314d0ef0bfbdec1f84241b77aa12b06aba221893fee8c728 4026532896:/usr/lib64/libpcre2-8.so.0.11.2
11 509eed1e60450cacb7a50135b0b5bad68bfdbea4 ima-ng sha256:696e58e522641b5e1392f8927f1ec6da7dc2227ff224f9341bf795108603f9d2 4026532896:/usr/bin/dircolors
10 60776682416701d59dc629e089d78c6e5b09c9a5 ima-ng sha256:9f4a5f1f38860c5479da080378a632636d818675b3f19849a4c43fe5242beadc /usr/bin/locale
11 a8644b70da2f5c21375daf0282b54ba14a664475 ima-ng sha256:e9d92cd921a0aff13e408f1a2584d1d1bf6d6e385e4587ed315750de2e3a4afd 4026532896:/usr/bin/locale
11 244db78d5a27491ff64dd2e4151504d87f5dab8e ima-ng sha256:f67b8429b6c88aed90d42f0ae5c10cbd0db870999f07f9ab64b81920557cf243 4026532896:/usr/bin/sed
11 9b1628bd57965c0e533792d7387d7aed7f325c7b ima-ng sha256:2bc581d5f250e8cc107293dc20146c5ca4c284542fe2a710ddd7a86d18922689 4026532896:/usr/lib64/libacl.so.1.1.2301
11 4f95cb7d1f8eeea324de0daf4f991ae7f4d8d5e9 ima-ng sha256:f60ce3ddcc706168ed8af61c585782e841e242d3053132bfd60e01f9980b776a 4026532896:/usr/lib64/libselinux.so.1
11 930798123fa89441440093cd8f9f1226a07a6e63 ima-ng sha256:29bd22f15758028d3a11ed853b9fb93156cc19915fdb844b73e3075d3ce6510b 4026532896:/usr/lib64/libattr.so.1.1.2501
10 39e08422189f153283aafba2fa32240fb2149177 ima-ng sha256:d91502c3a044c776ae0c9b799b59df4fb1901aa84dcb4c26c4dcbe55cf5951be /usr/bin/sha256sum
11 07d6e54b15a424d098754beb53826dc4302e47cb ima-ng sha256:45c8aa2c7fbac7881cc7edd30d12e4584ce83ad4733d03d80eaf1a53a3b555e5 4026532896:/usr/bin/sha256sum
11 e55e3e1e1b707e9b0a7833dae9ef518f9cdf5d86 ima-ng sha256:a034c7cb9eaa990a1e71ff0b72b689b5a311de8117a1bdf8191cce5f6157fce7 4026532896:/usr/lib64/libcrypto.so.3.0.9
11 c4c94f9689565e866c0fd79e53da7ca46f3deb7b ima-ng sha256:d6c93839cc0e7c29fab74d08bfda639ce776c594ac77a4704160bd5069b9a7e8 4026532896:/usr/lib64/libz.so.1.2.13
10 1844bd922c570347569afc8ca2551b0c26302661 ima-ng sha256:8a56e729fd7764215090c1a02781c465ddf534a50b602f76b3cc33c19e013bbd /usr/bin/tail
[avery@fedora container-ima]$
```

# Evalution

# Enabling non-repudiable logging of workload/platform specific system properties using eBPF.

avery.blanchard@duke.edu

https://github.com/avery-blanchard/container-ima

Duke UNIVERSITY

IBM