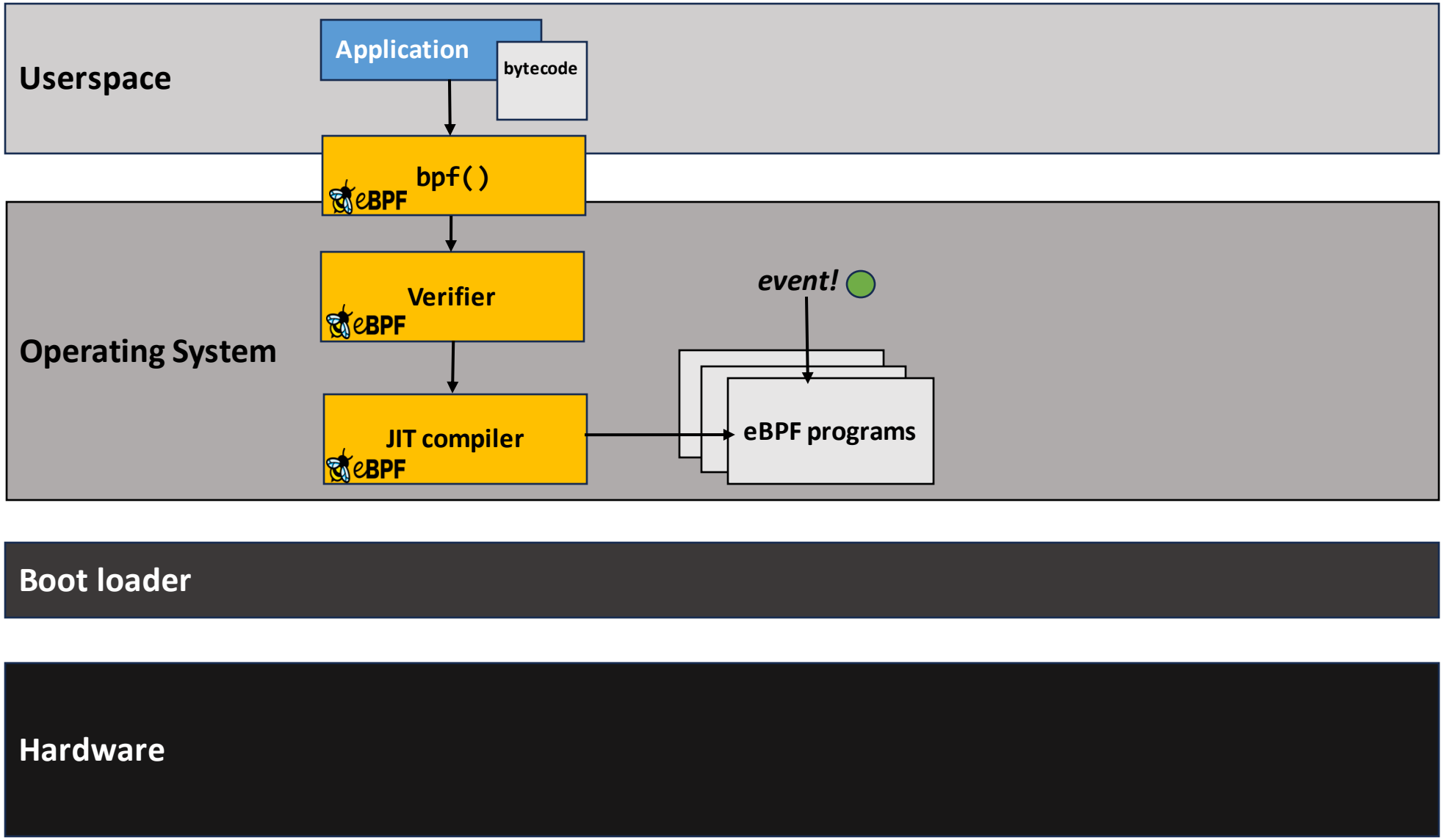
Extending Non-Repudiable Logs with eBPF

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

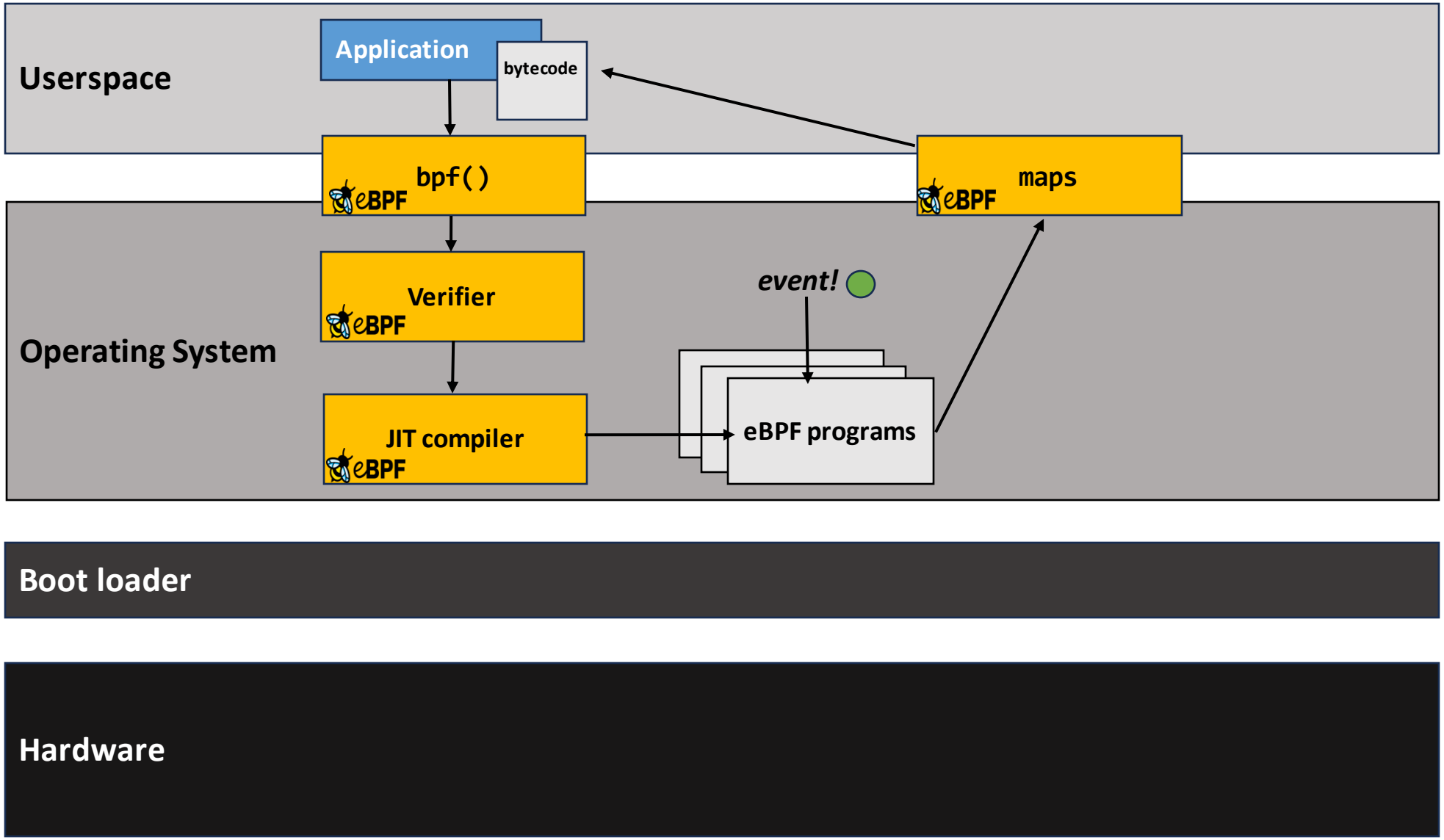
1 Duke University
2 IBM Research

November 13th, 2023

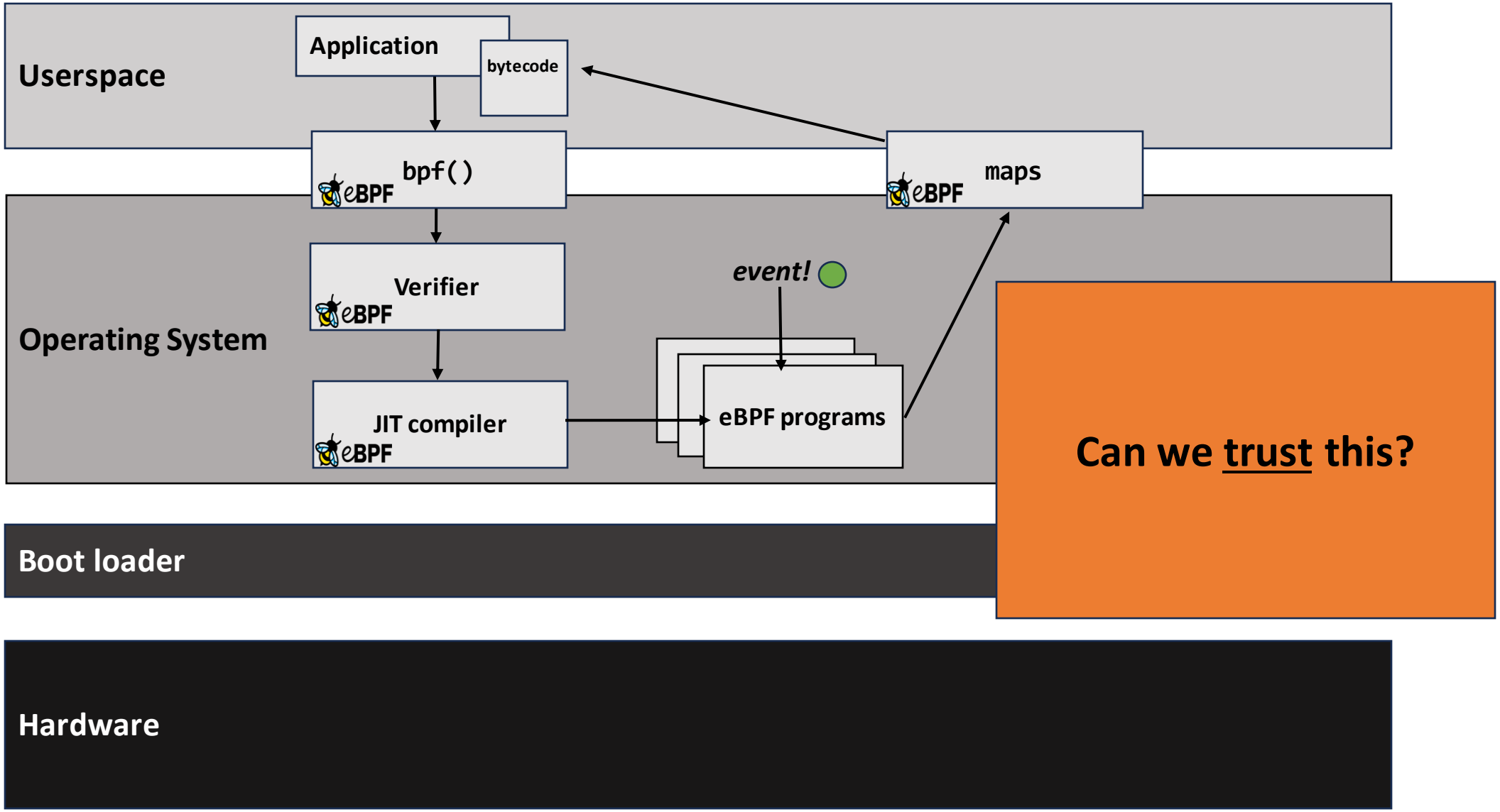
Visibility into System State with eBPF



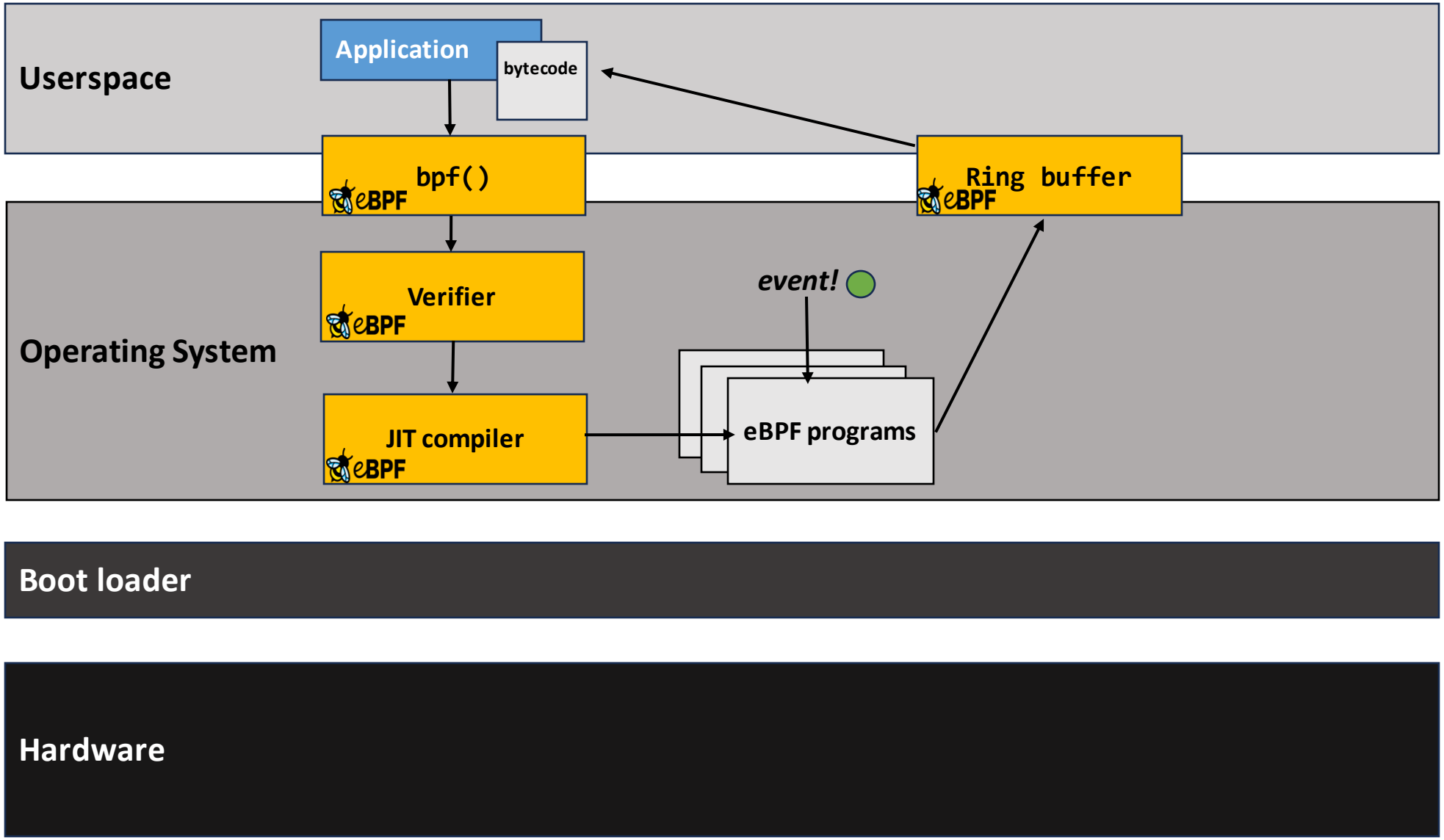
Visibility into System State with eBPF



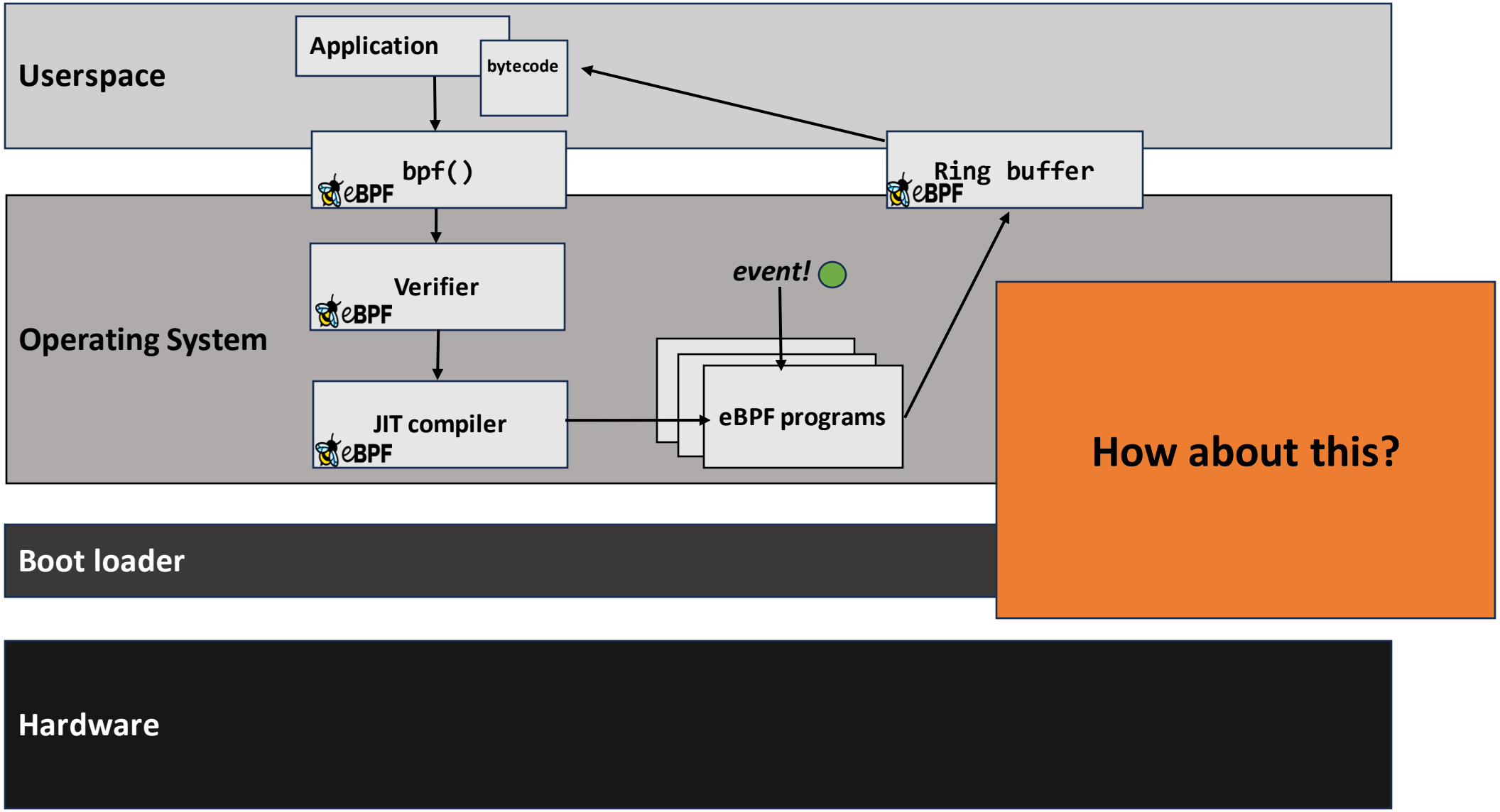
Logging System State



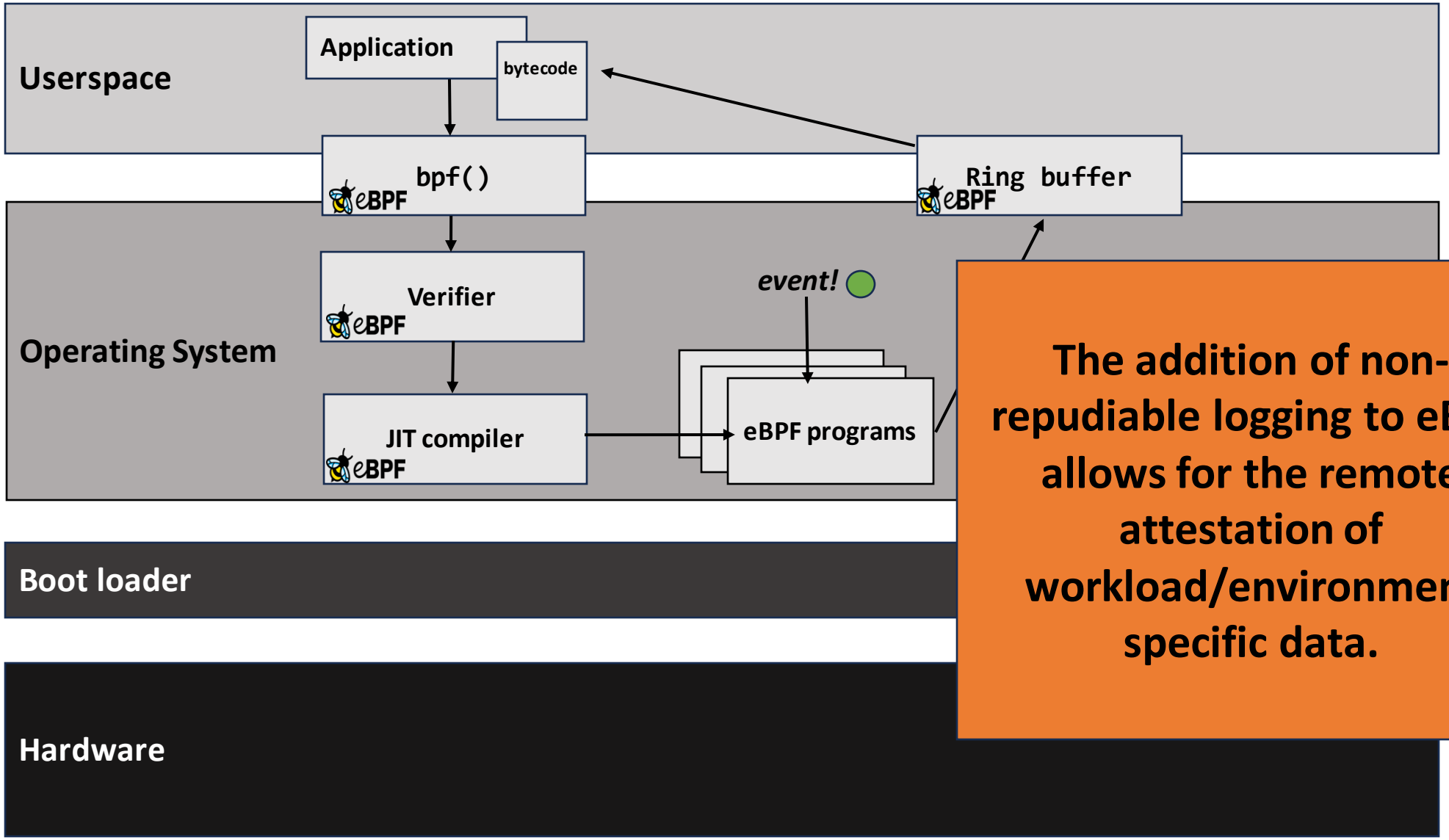
Visibility into System State with eBPF



Logging System State

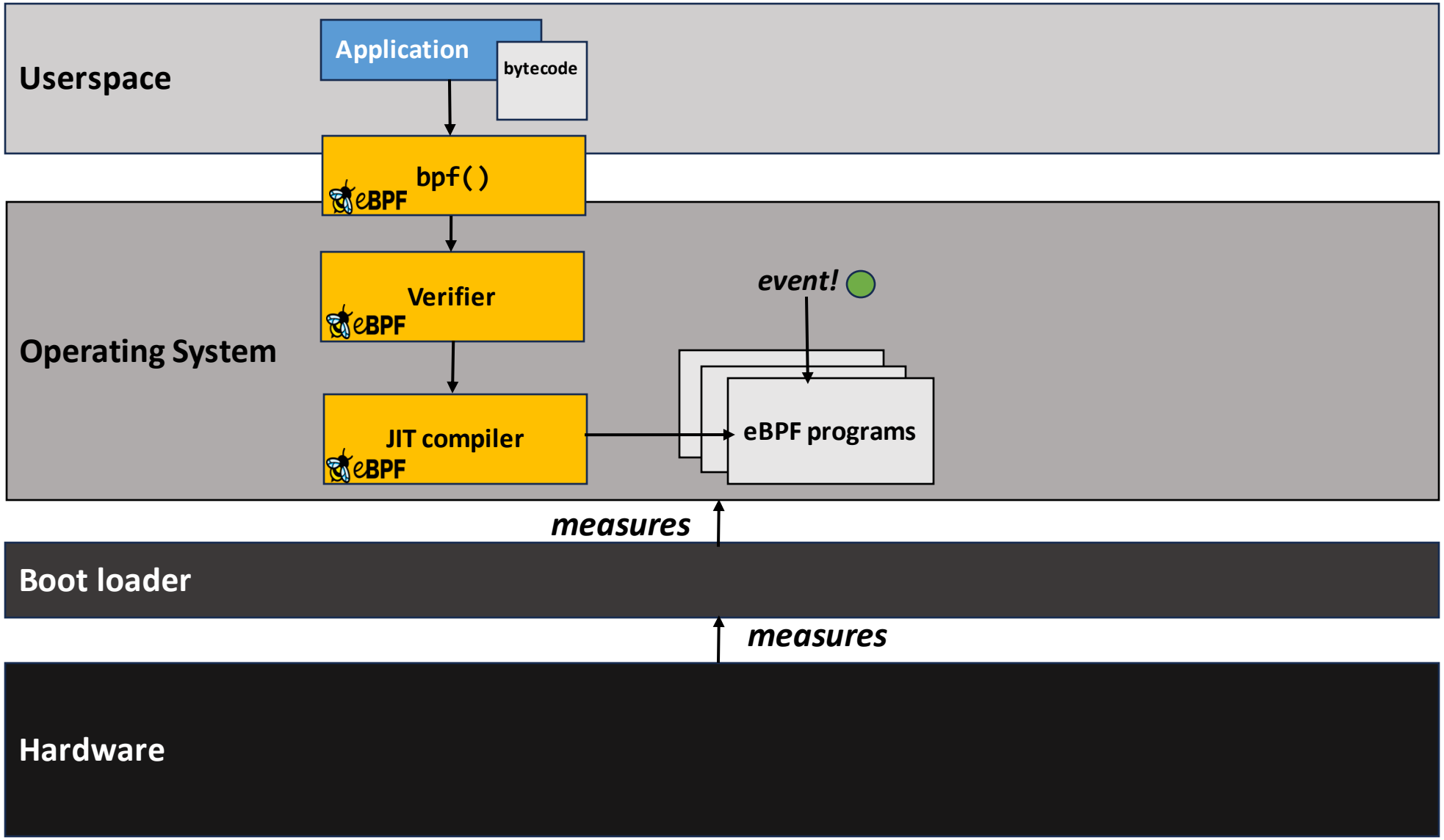


Non-repudiable Logging

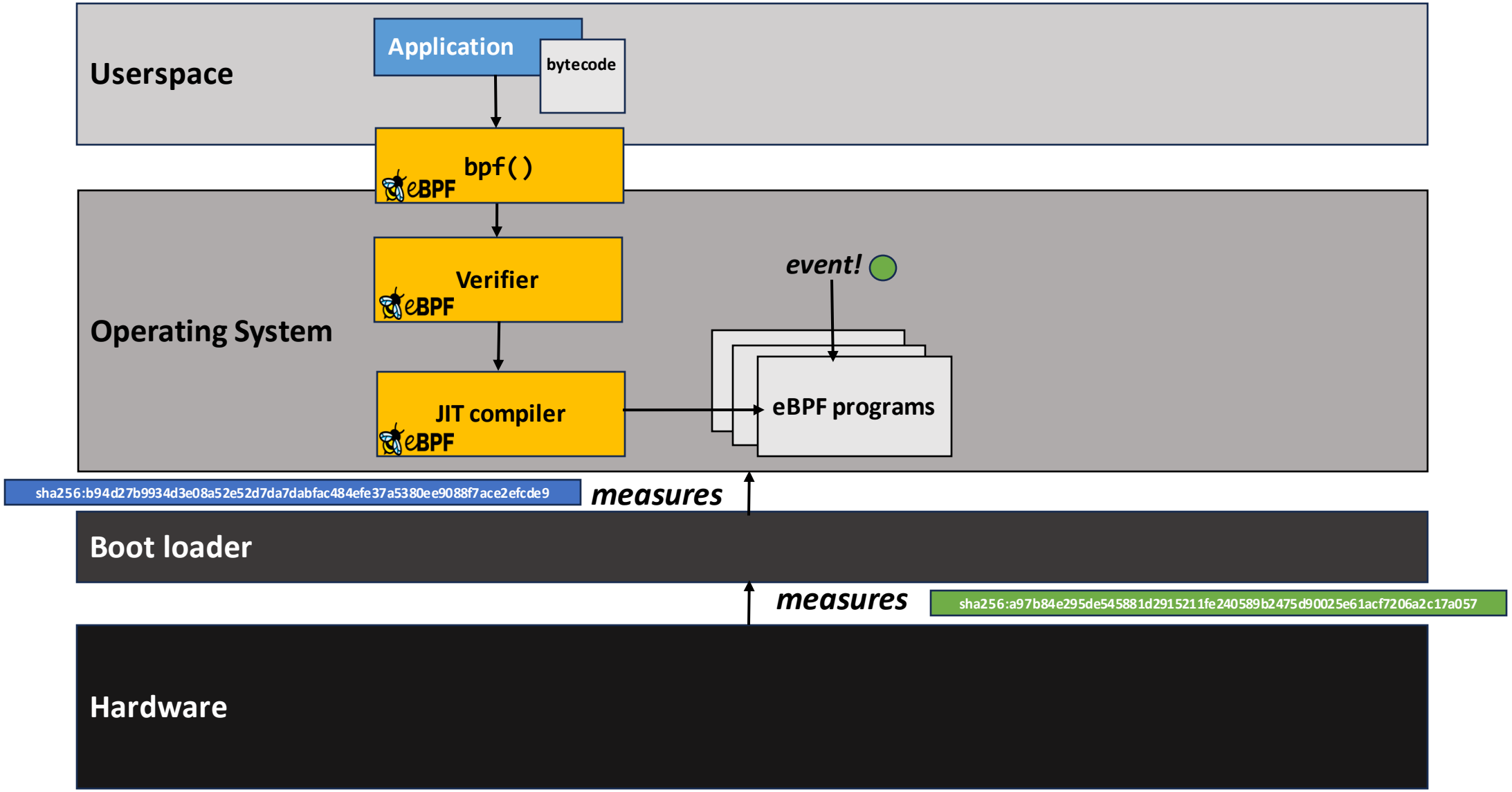


The addition of non-repudiable logging to eBPF allows for the remote attestation of workload/environment specific data.

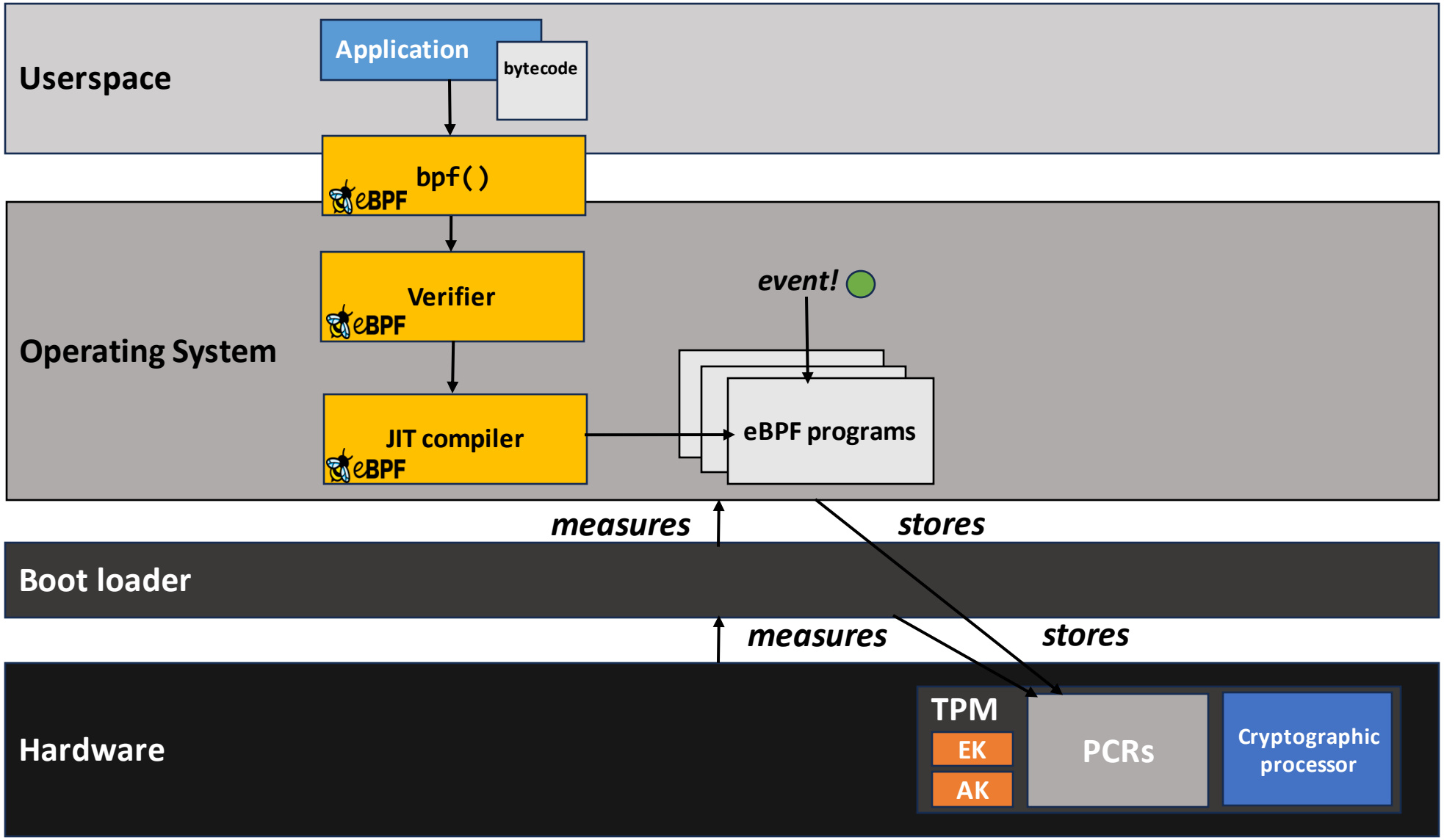
Building a Chain of Trust



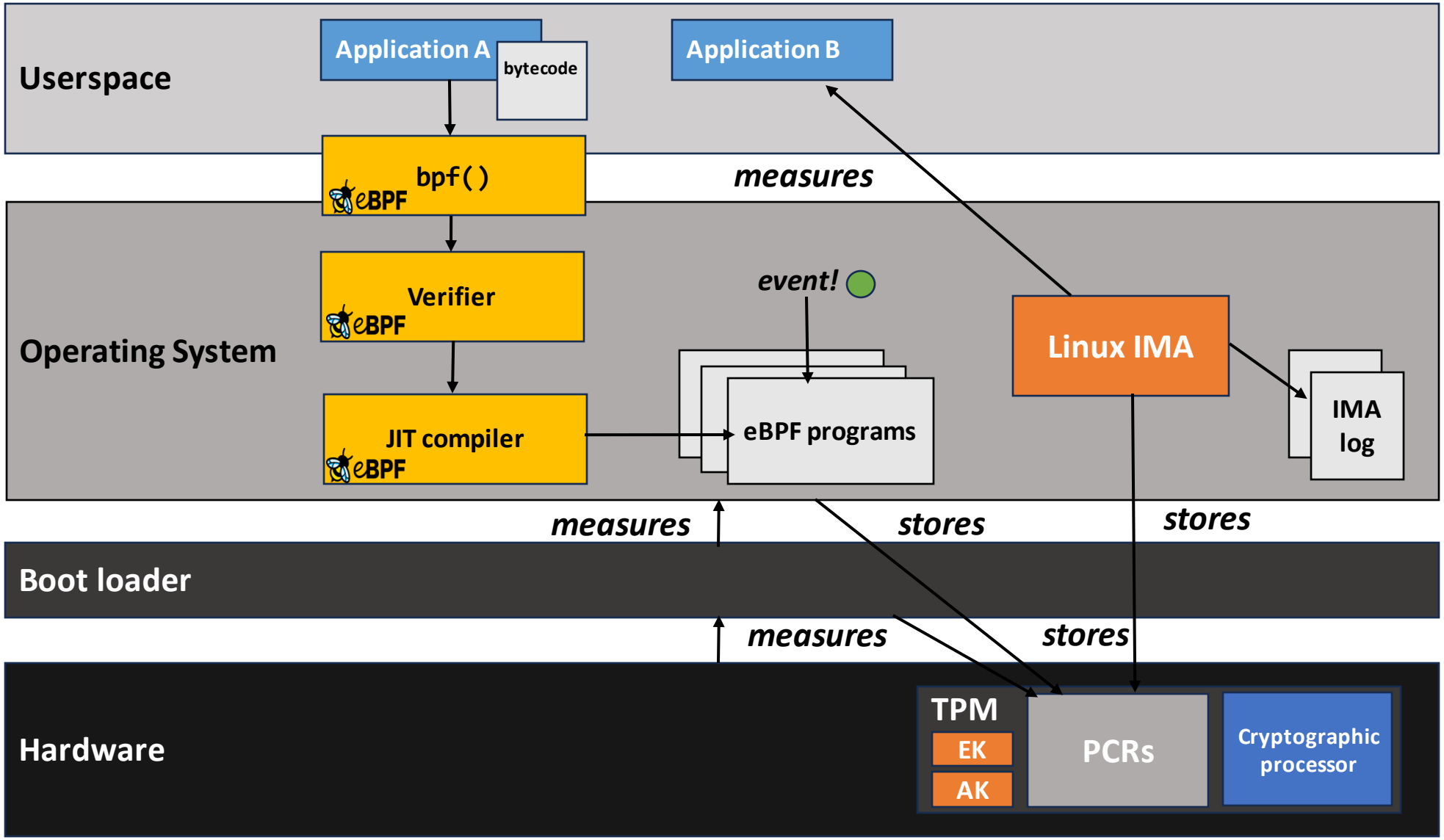
Building a Chain of Trust



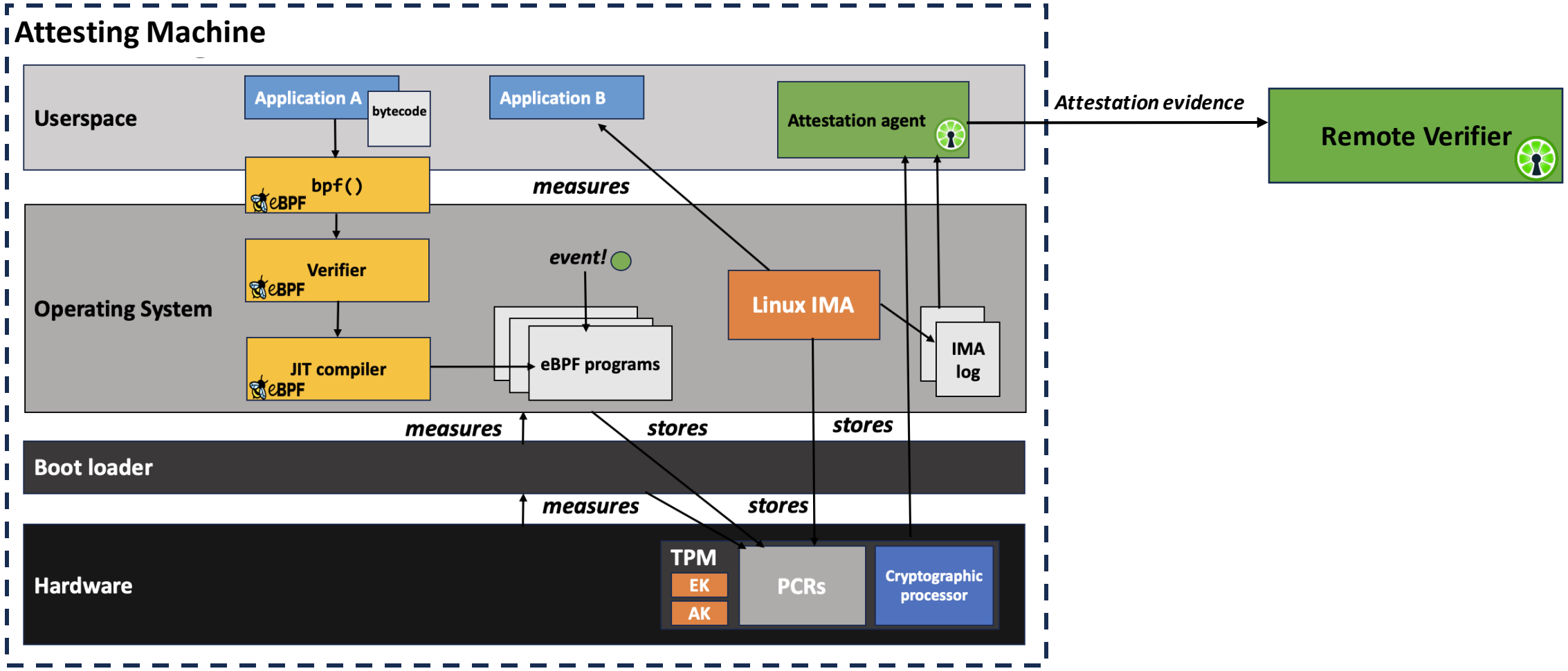
Rooting Trust in Hardware



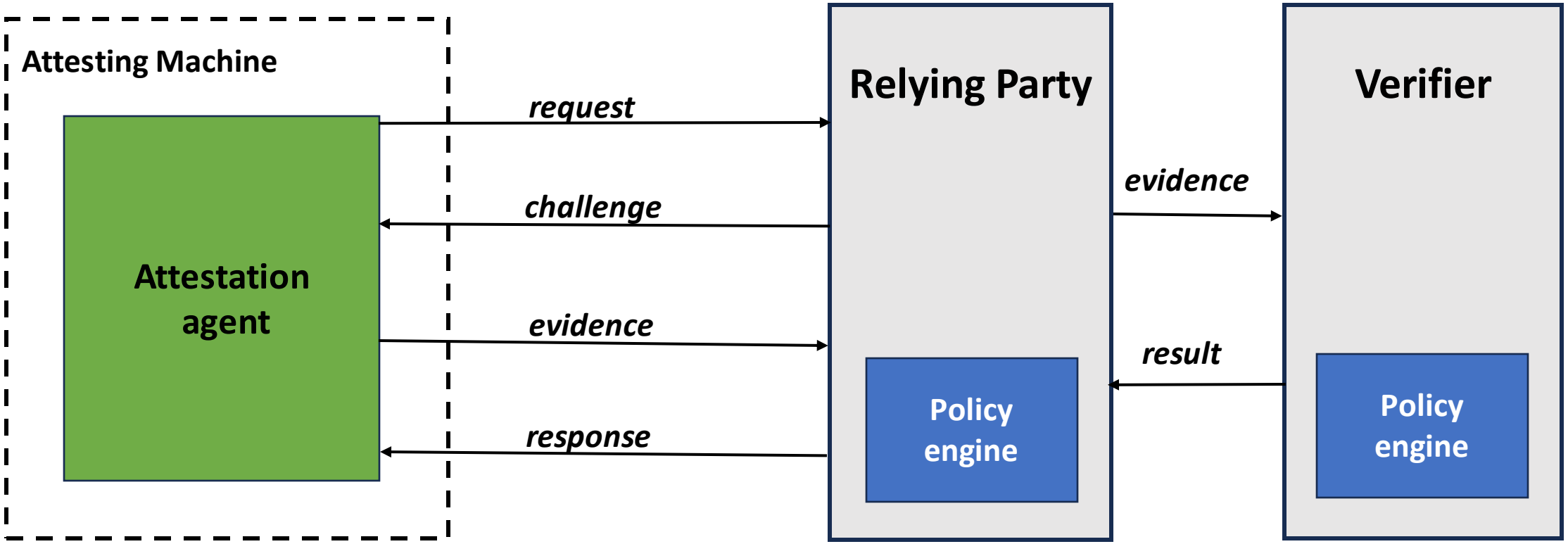
Extending Measurements Through Runtime



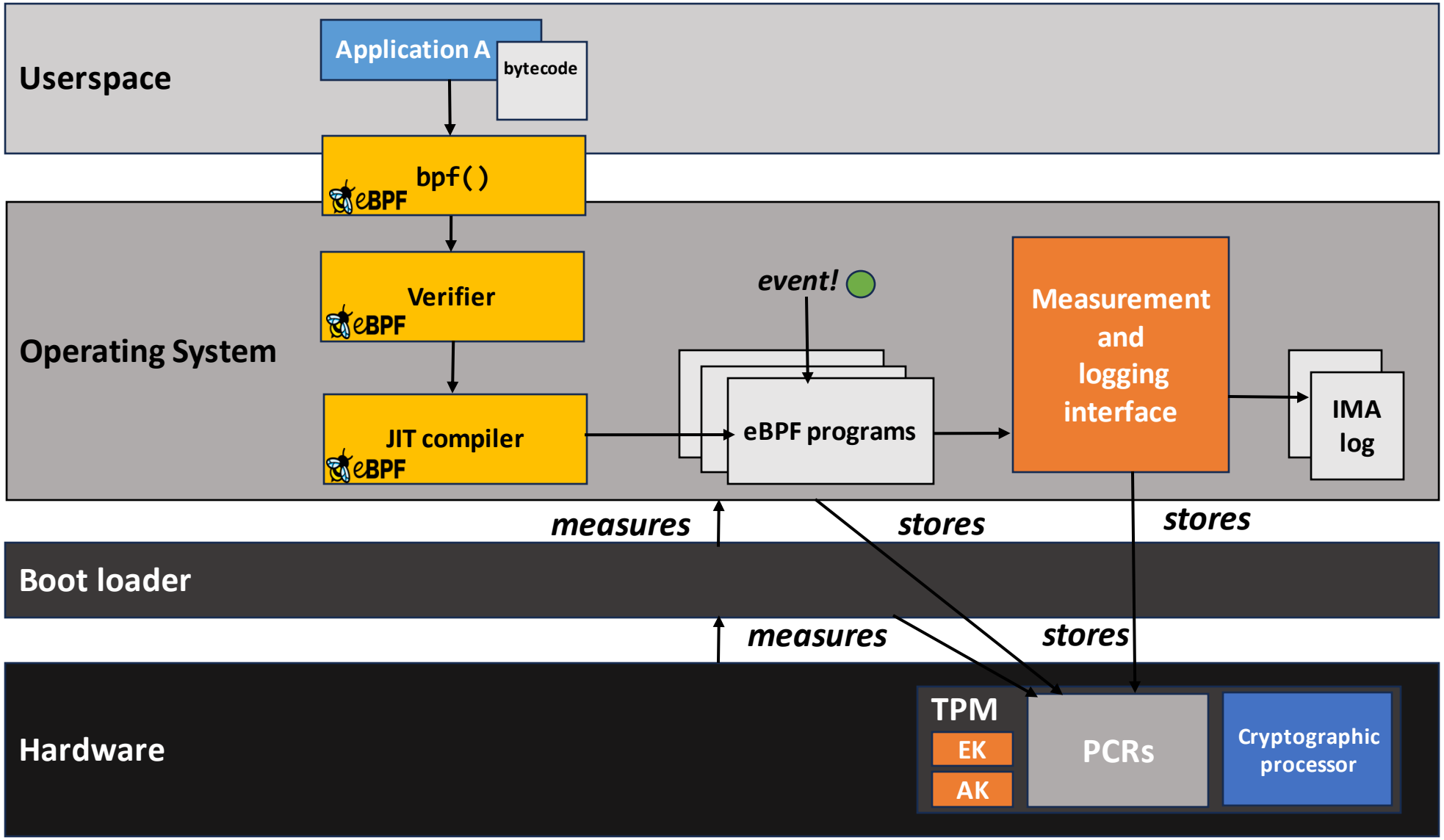
Building Trust in Environments



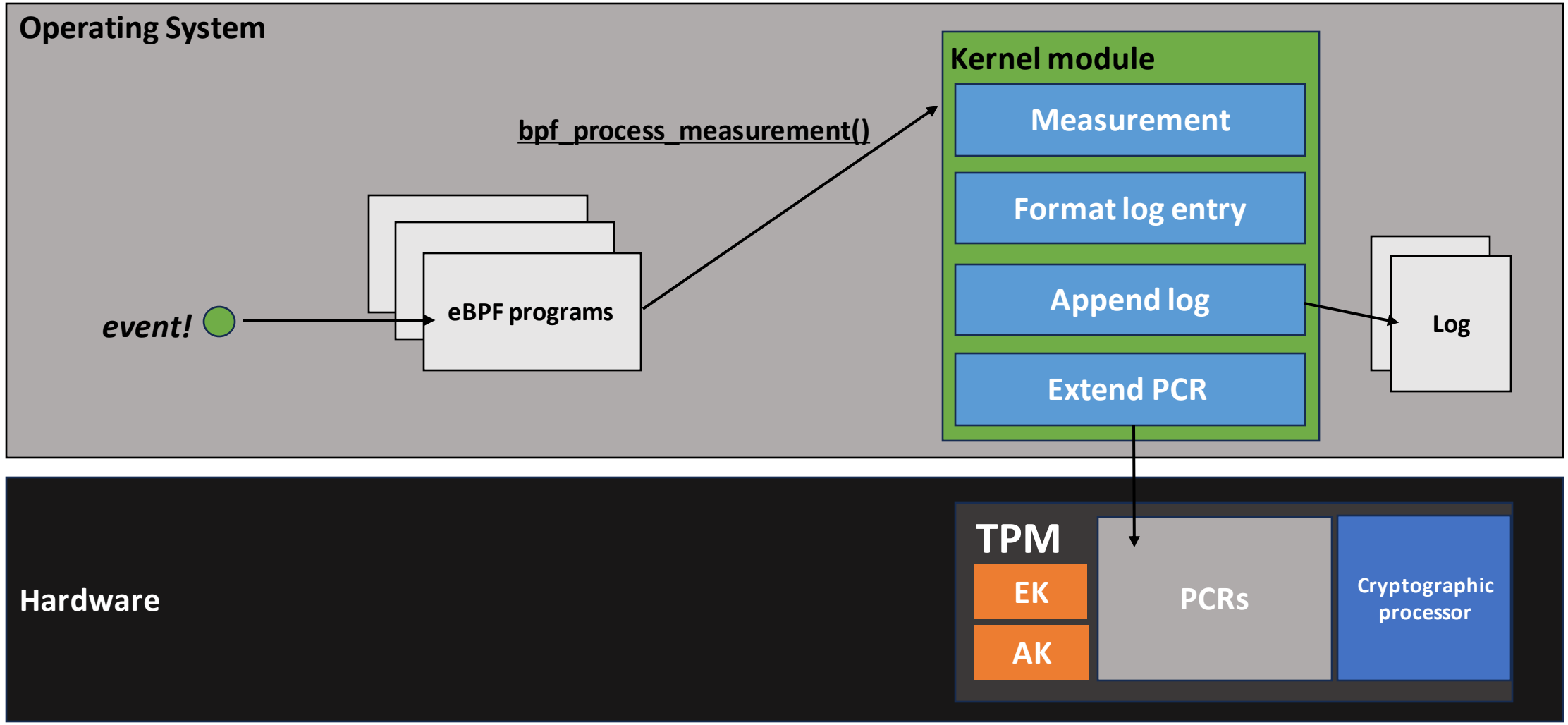
Attesting System Properties



Non-repudiable Logging in eBPF Programs



Measurement Interface

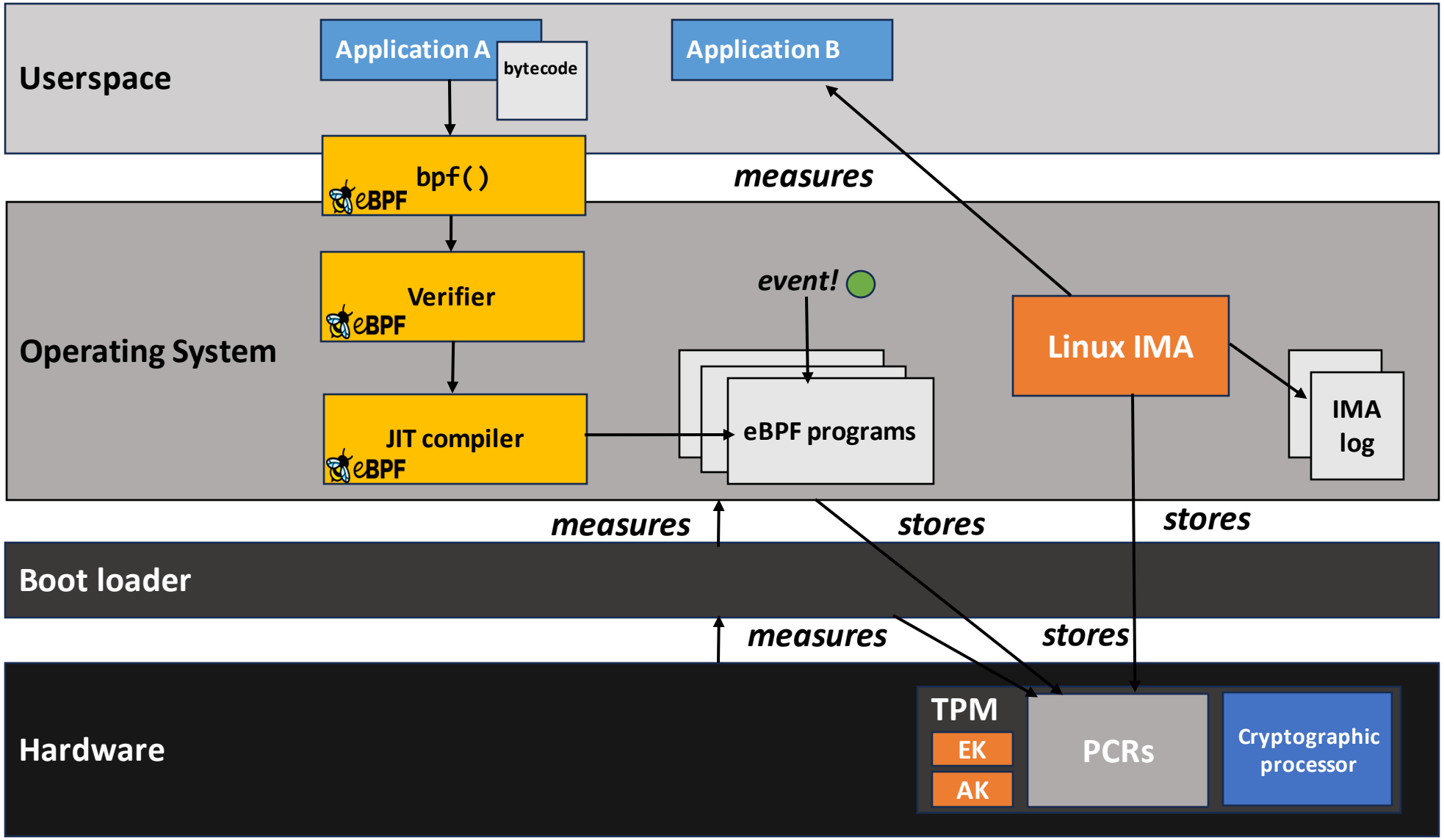


From the eBPF side

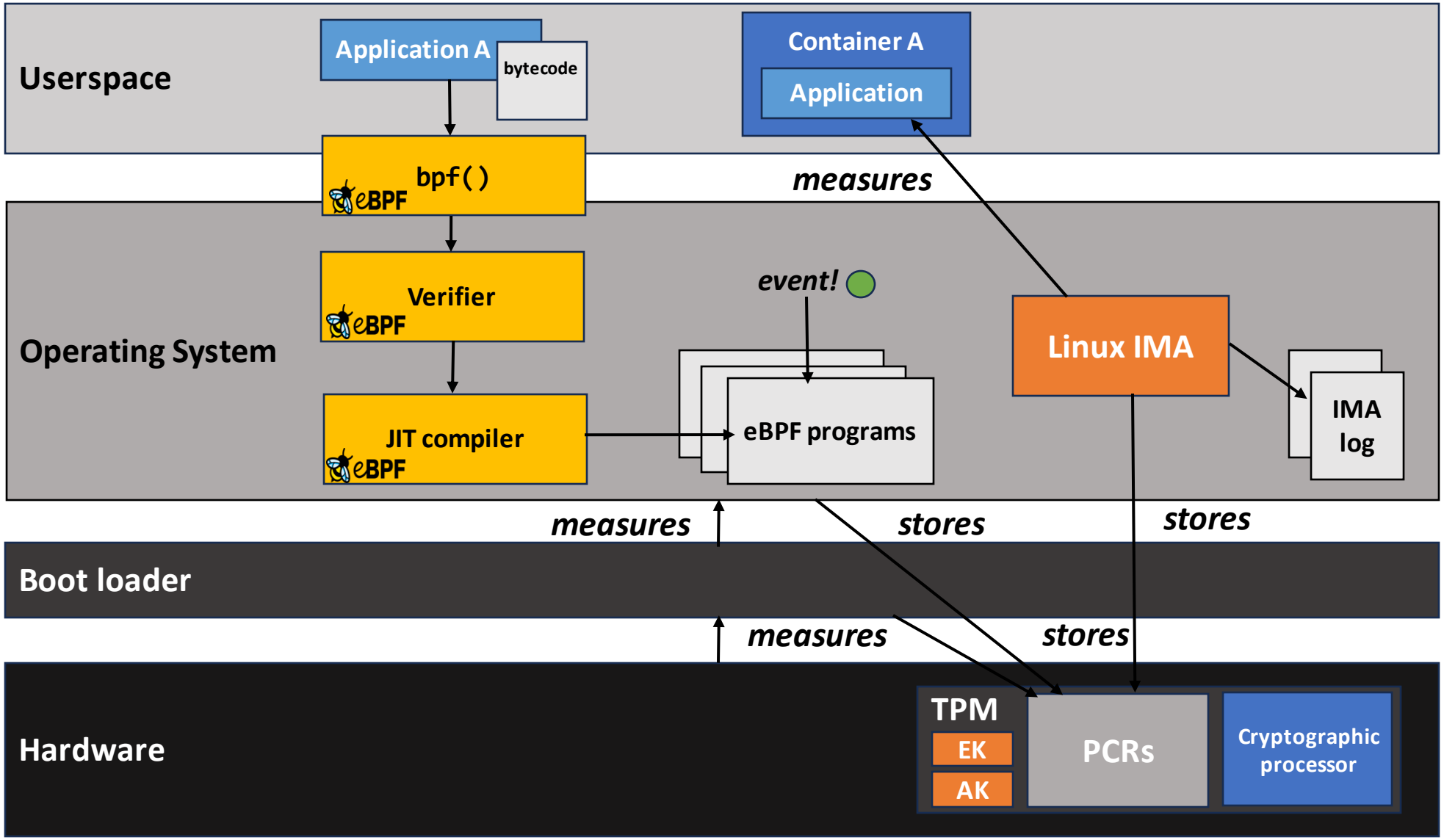
- Available to sleepable eBPF programs
- Programs can provoke the measurement and storage of formatted data and files

```
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;

}
```

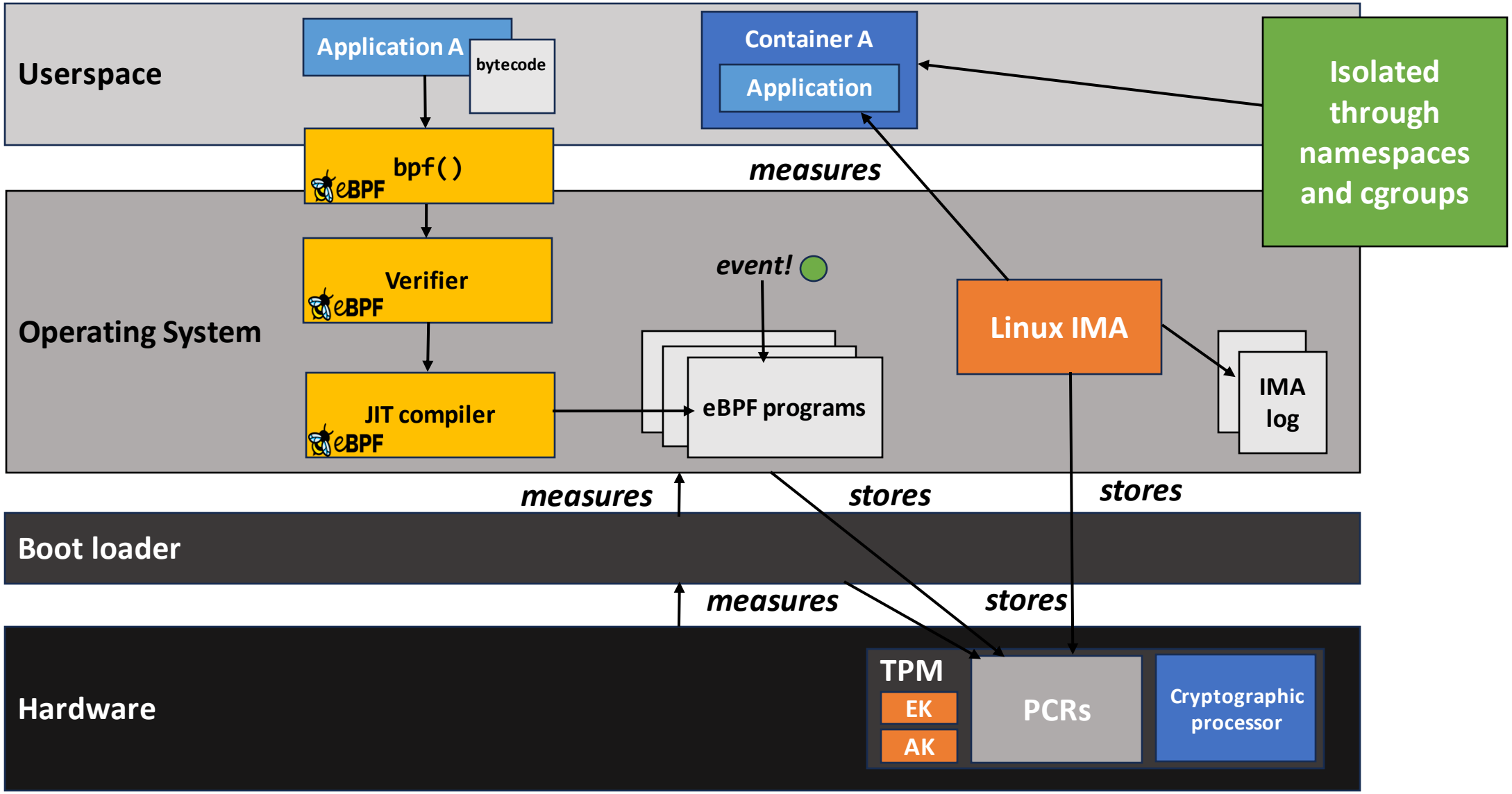
Example Use Case



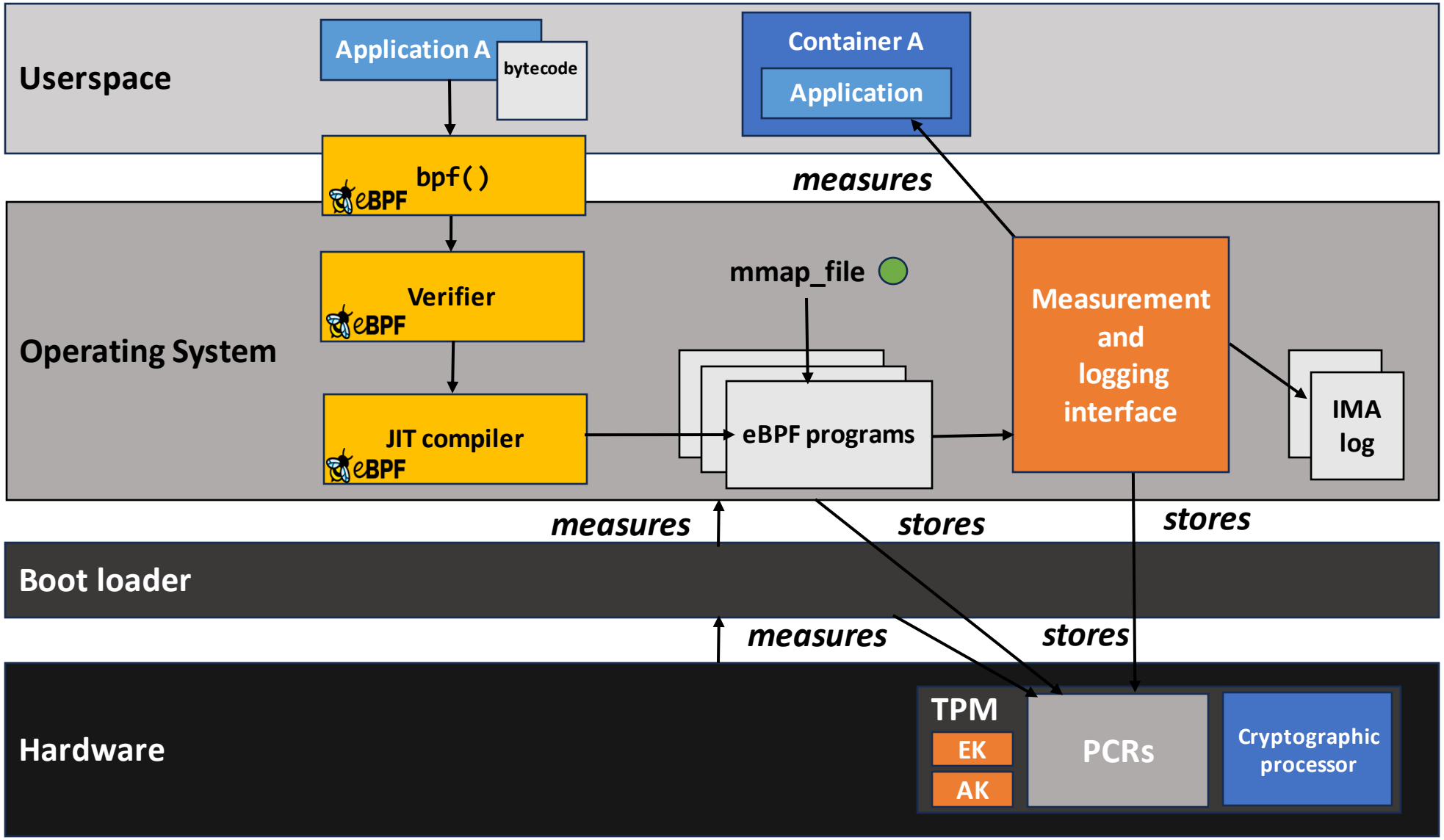
Extending Linux IMA to Containers



Extending Linux IMA to Containers



Adding Namespace Support to IMA



Resulting IMA Log

Evalution

Enabling attestation of workload/platform specific system properties using eBPF.

avery.blanchard@duke.edu

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