In Containers We Trust? Building Trust In Containerized Environments

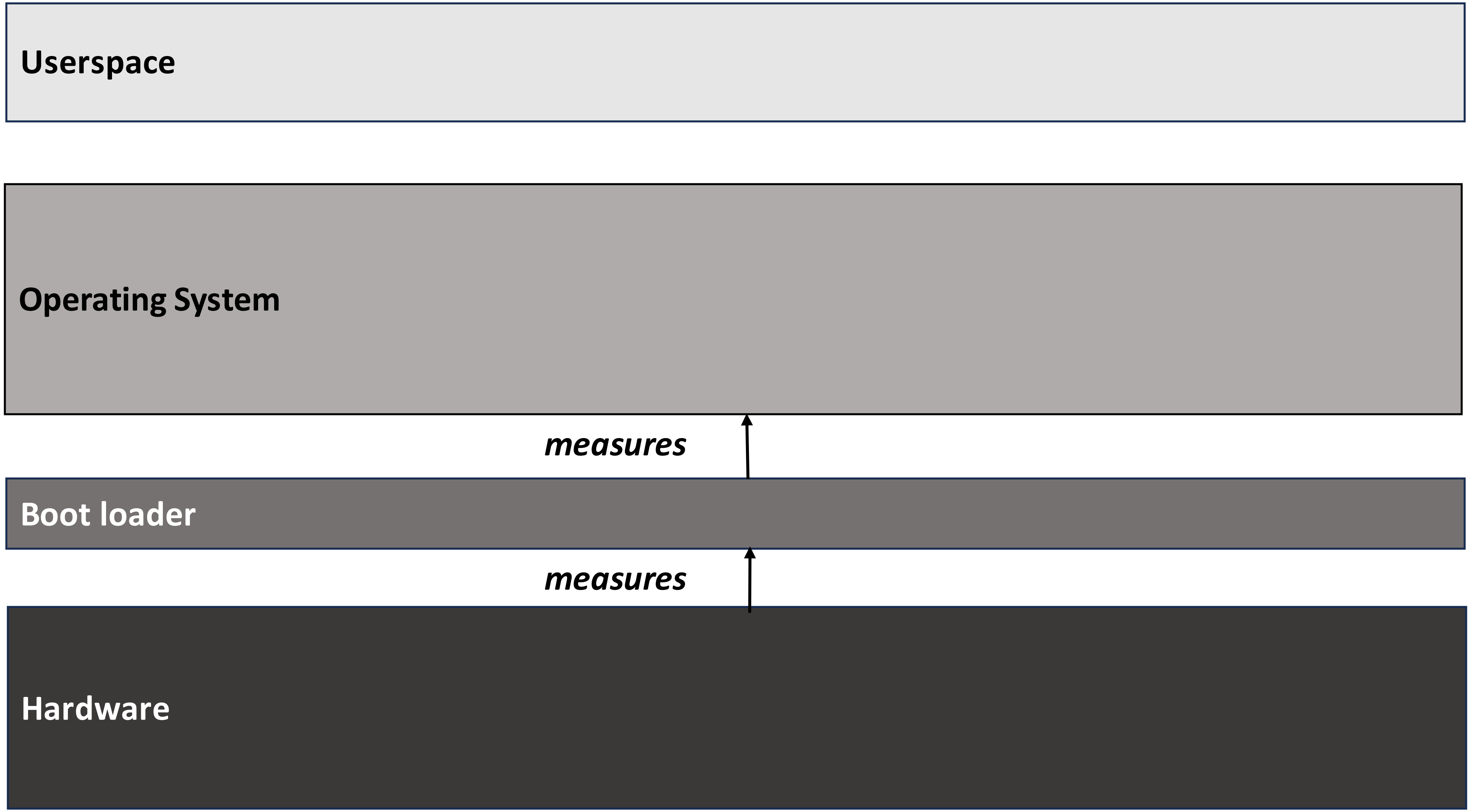
Avery Blanchard[1], Gheorghe Almasi[2], James Bottomley[2] and Hubertus Franke[2]

1 Duke University
2 IBM Research

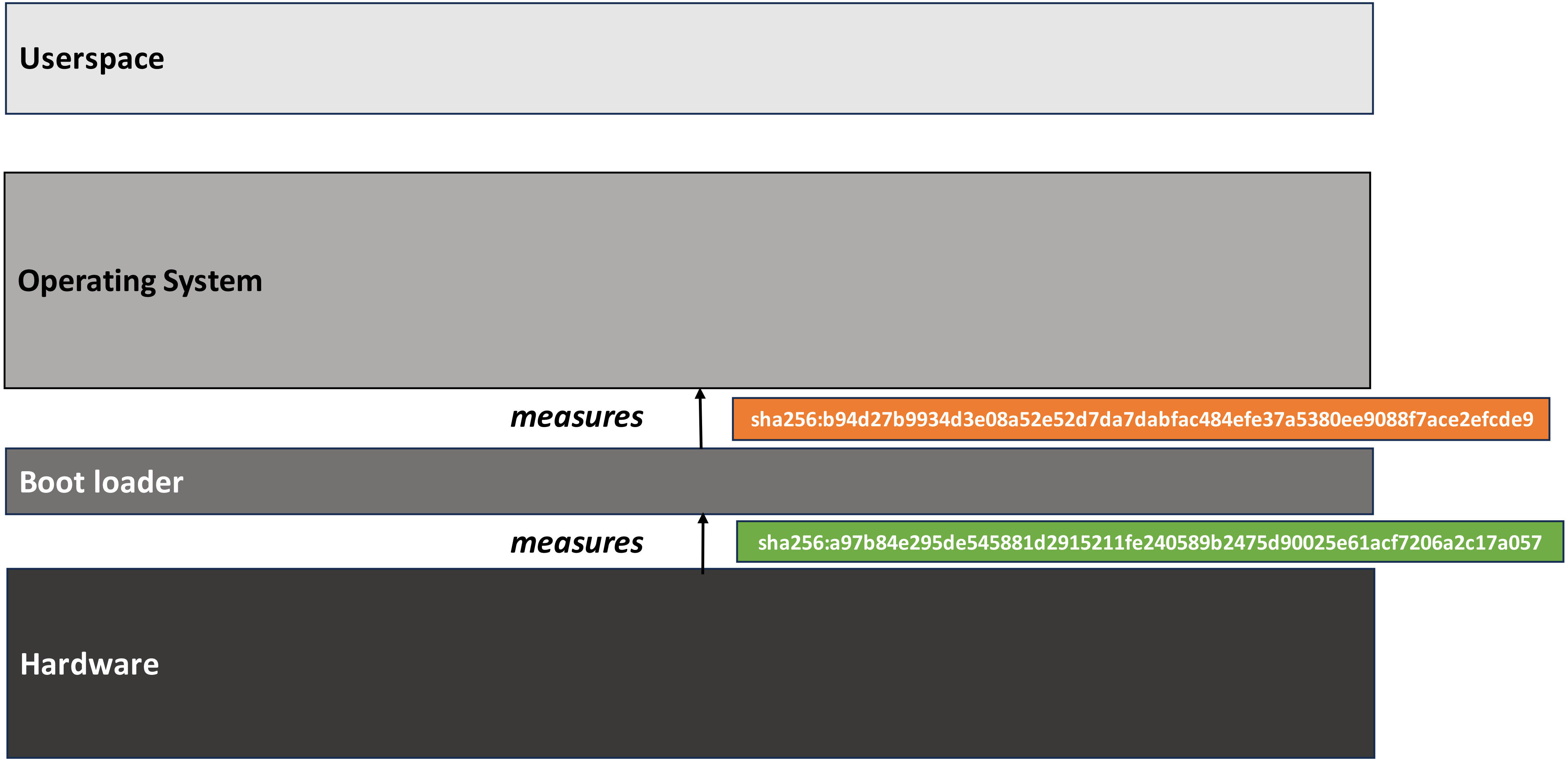
November 13th, 2023

Containers are ubiquitous and blindly trusted…

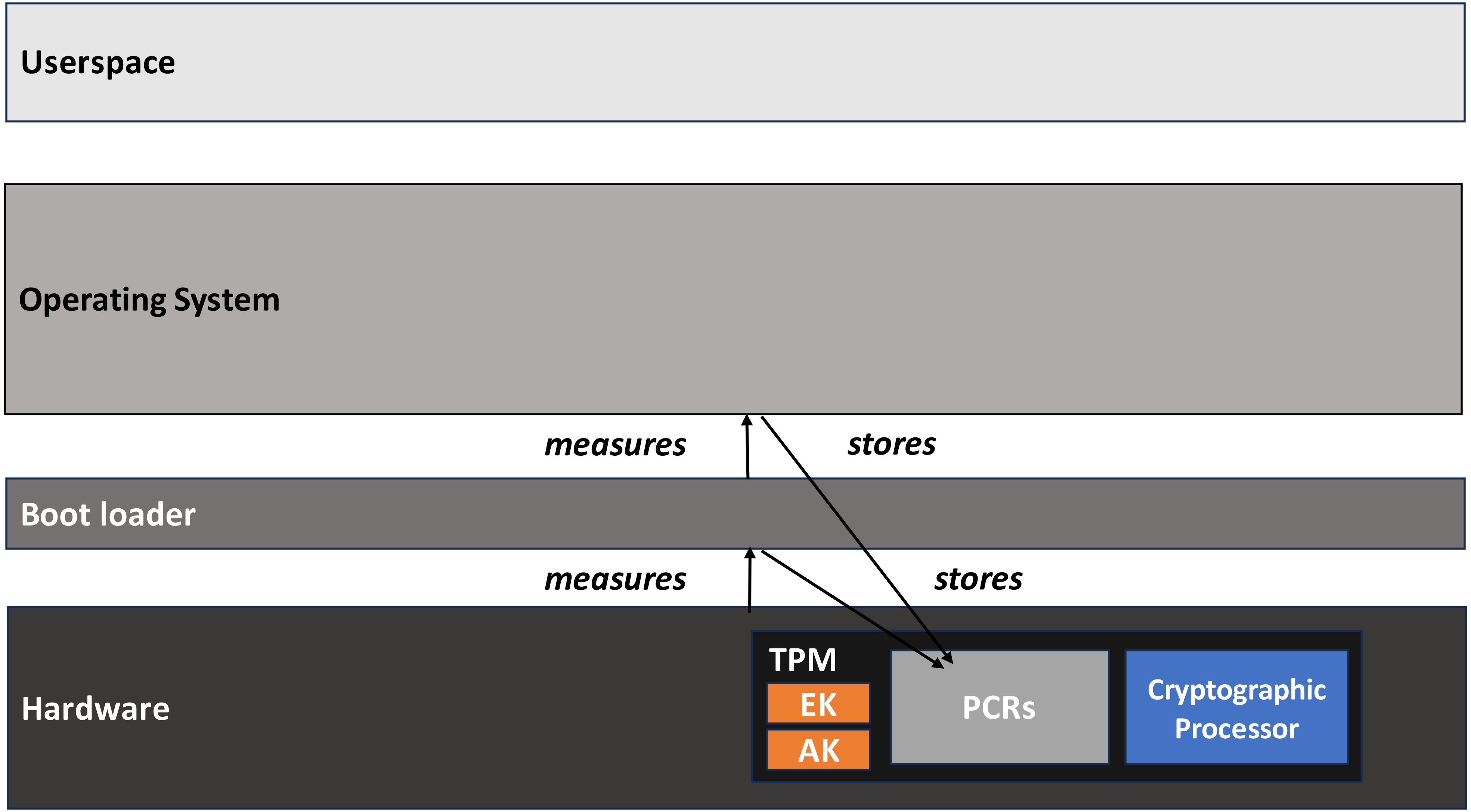
Defining Trust: Measurement



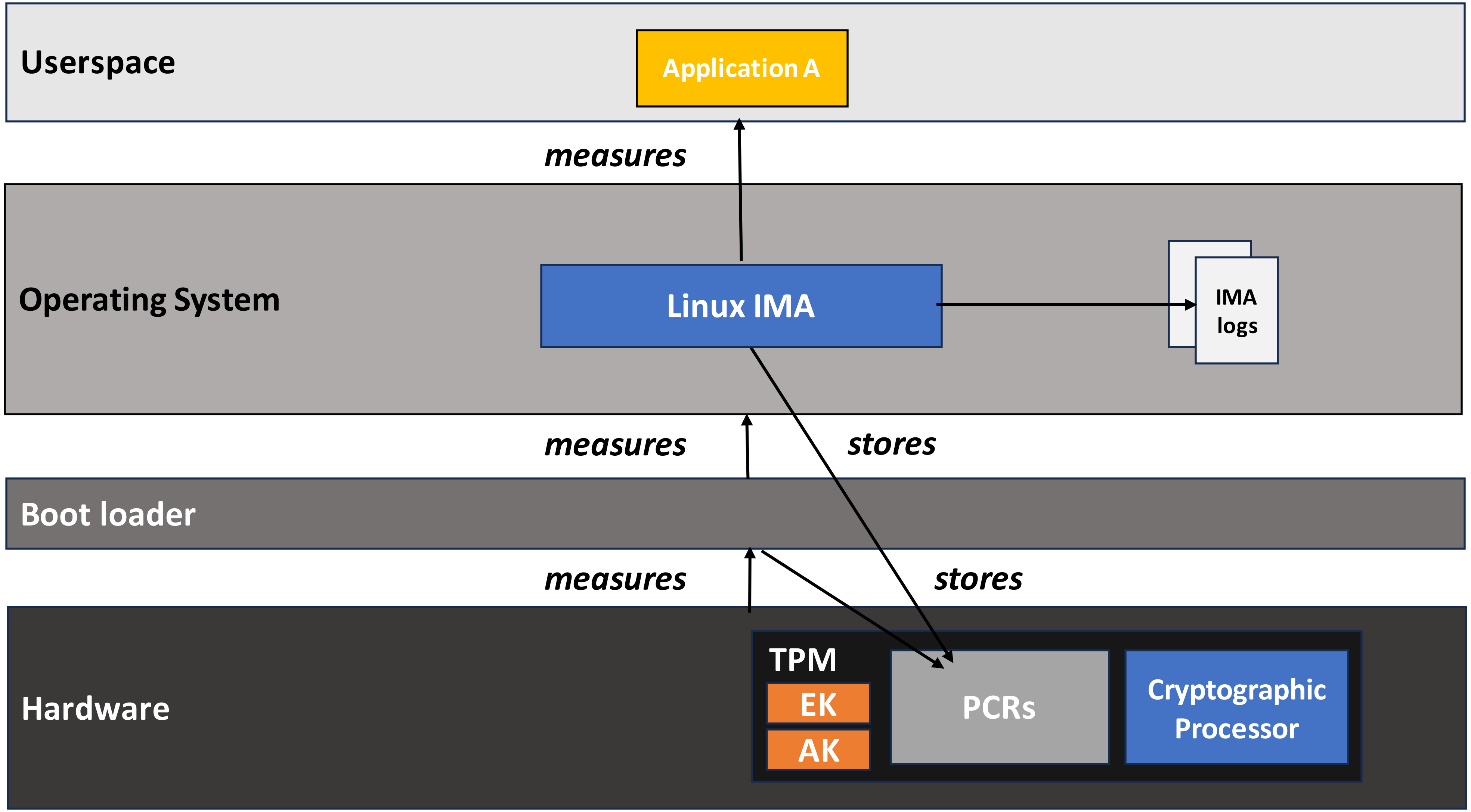
Defining Trust: Measurement


Userspace
Operating System
measures
sha256:b94d27b9934d3e08a52e52d7da7dabfac484efe37a5380ee9088f7ace2efcde9
Boot loader
measures
sha256:a97b84e295de545881d2915211fe240589b2475d90025e61acf7206a2c17a057
Hardware

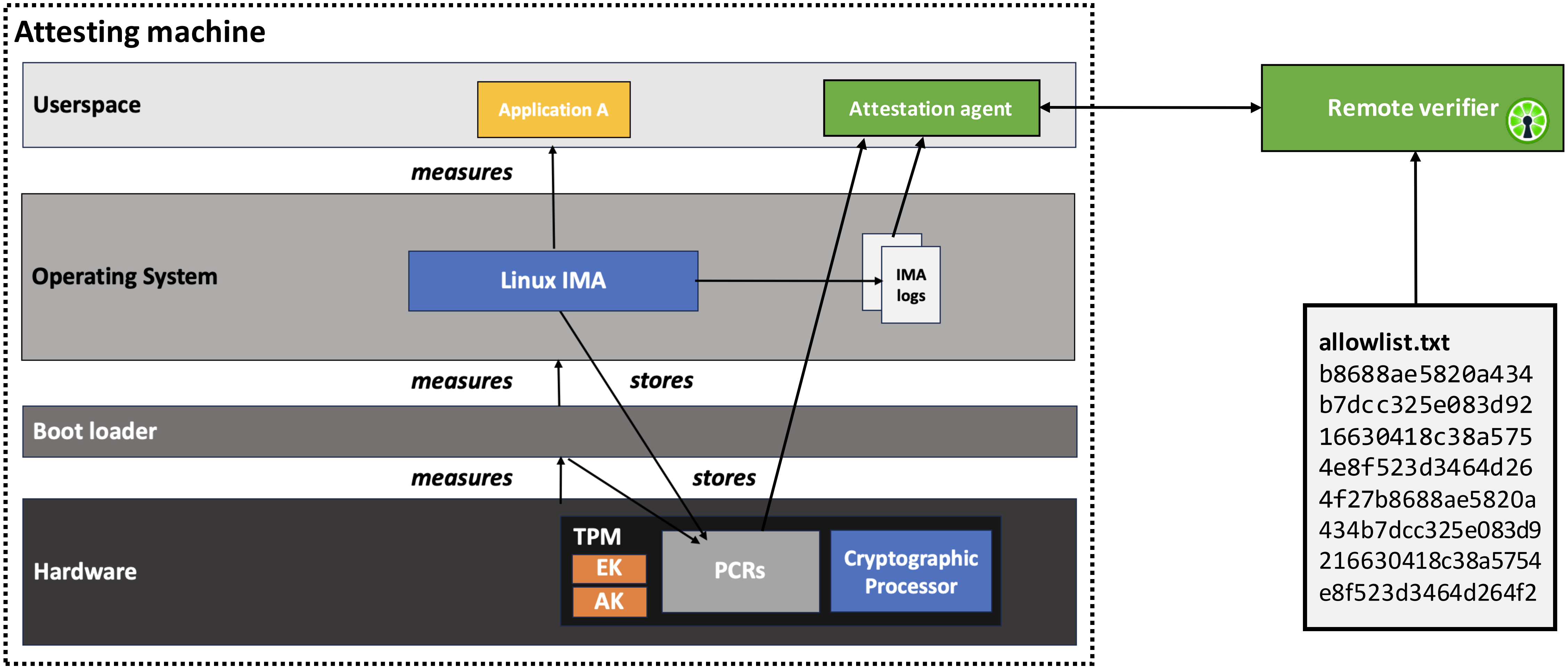
Building Trust from Hardware



Extending Integrity Measurements in Runtime

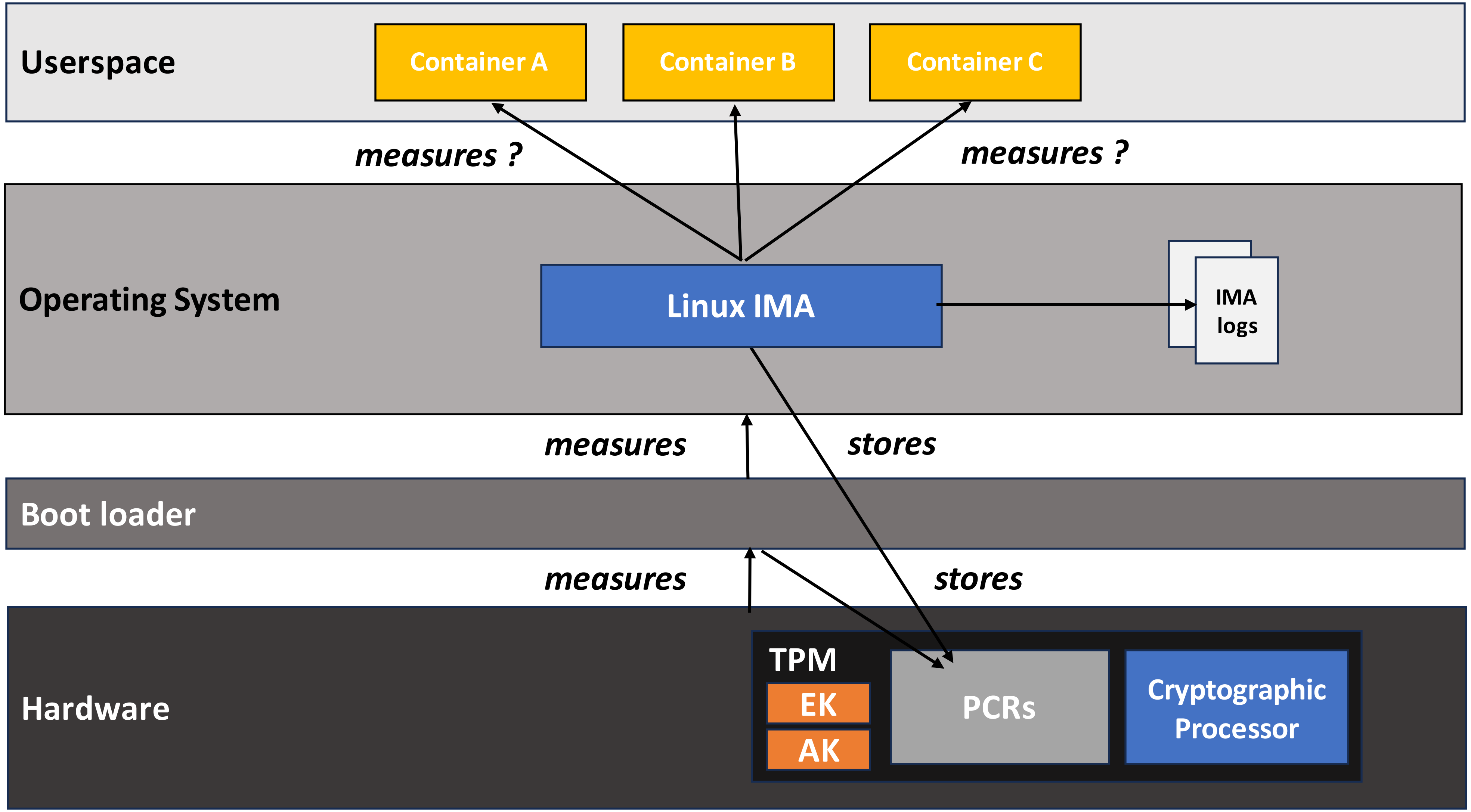


Building Trust in Remote Environments



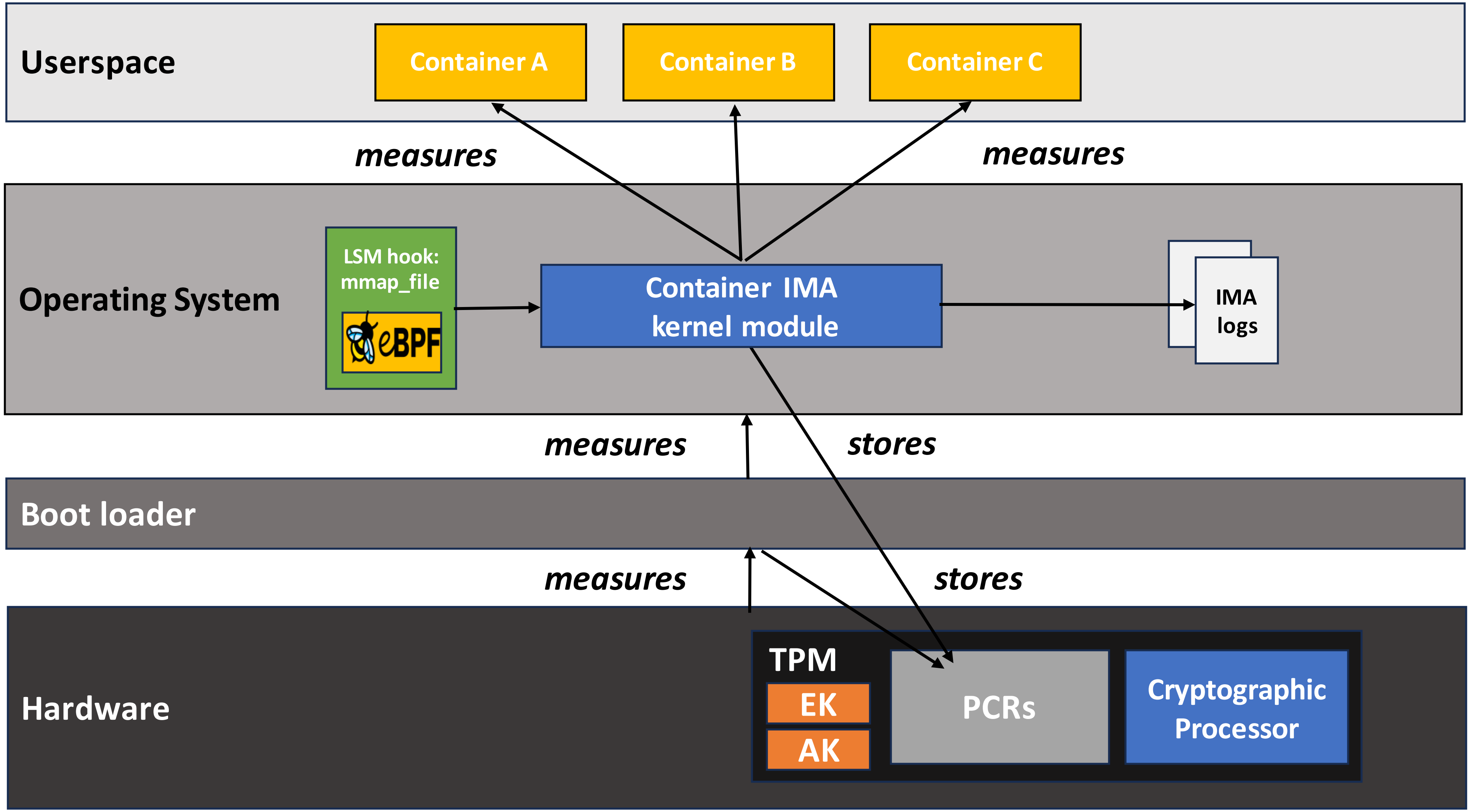
```
allowlist.txt
b8688ae5820a434
b7dcc325e083d92
16630418c38a575
4e8f523d3464d26
4f27b8688ae5820a
434b7dcc325e083d9
216630418c38a5754
e8f523d3464d264f2
```

Container Present a Gap in Trust and Integrity

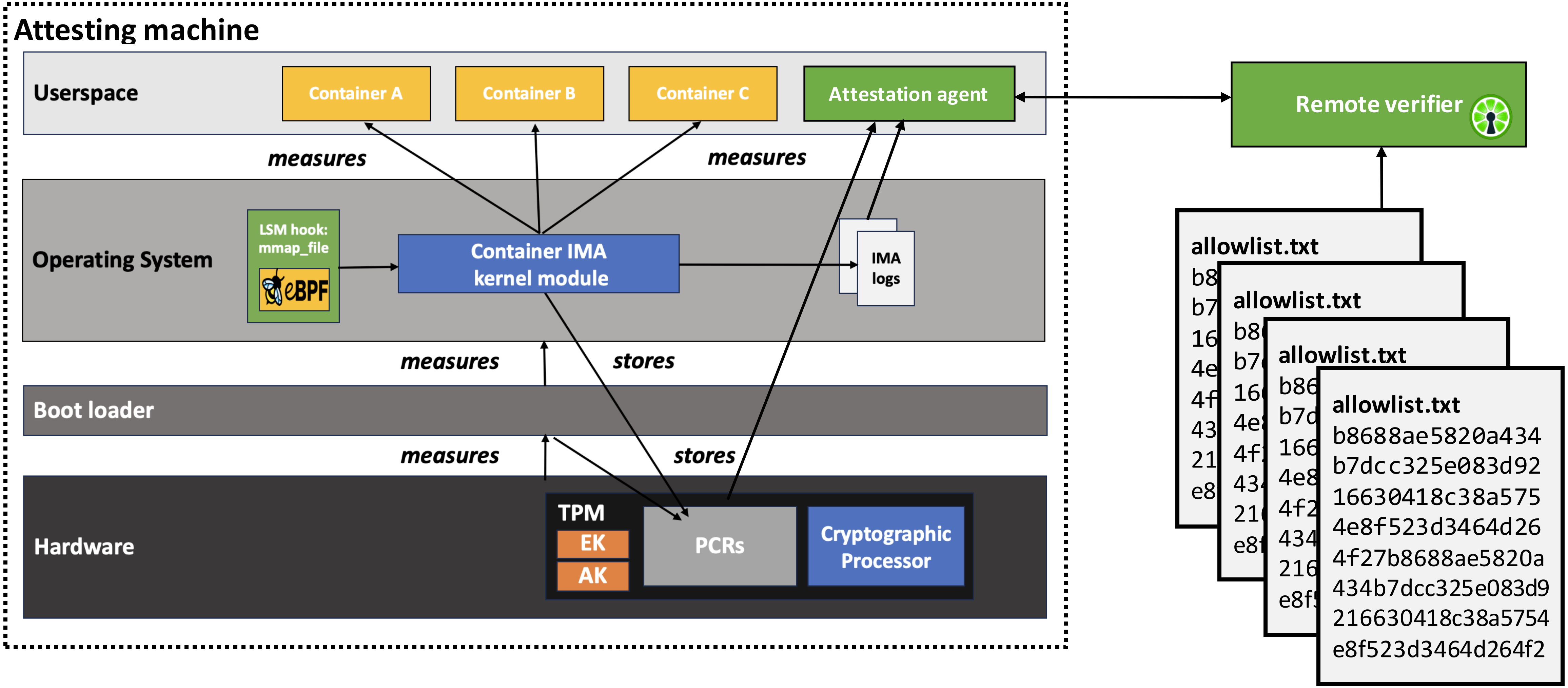


Enabling Container Attestation: A Preliminary Approach

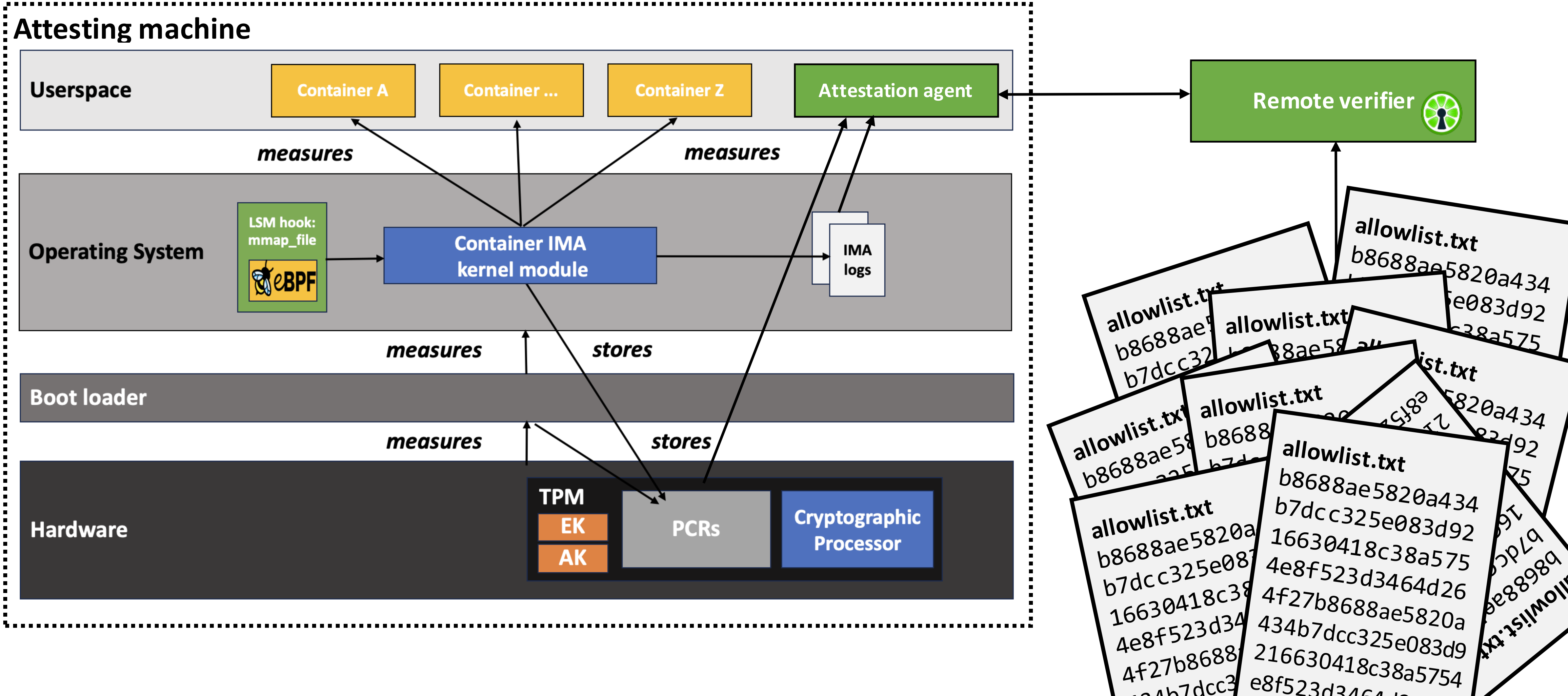
Extending IMA to Containers using eBPF



Attesting Container File Integrity



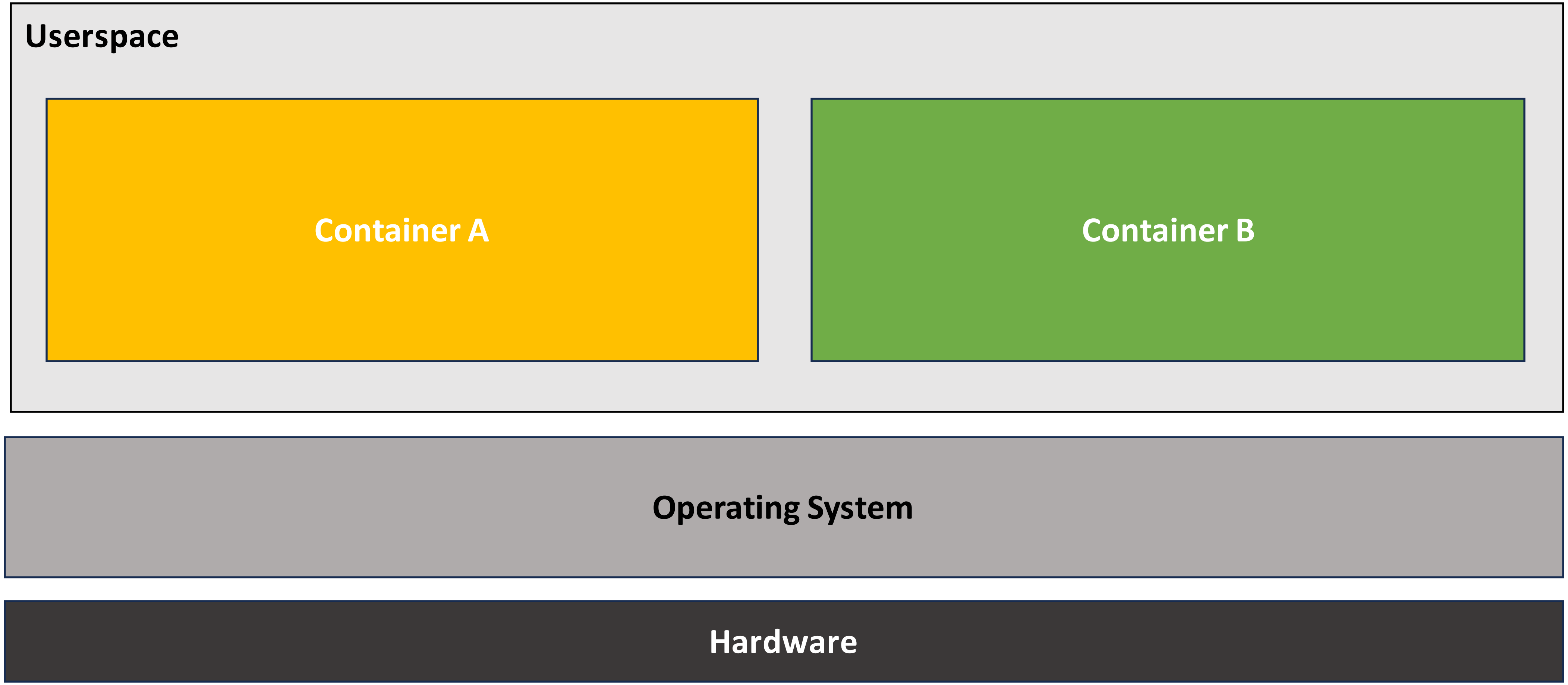
Attestation vs. Containers



Measuring Container Image Integrity in the Kernel

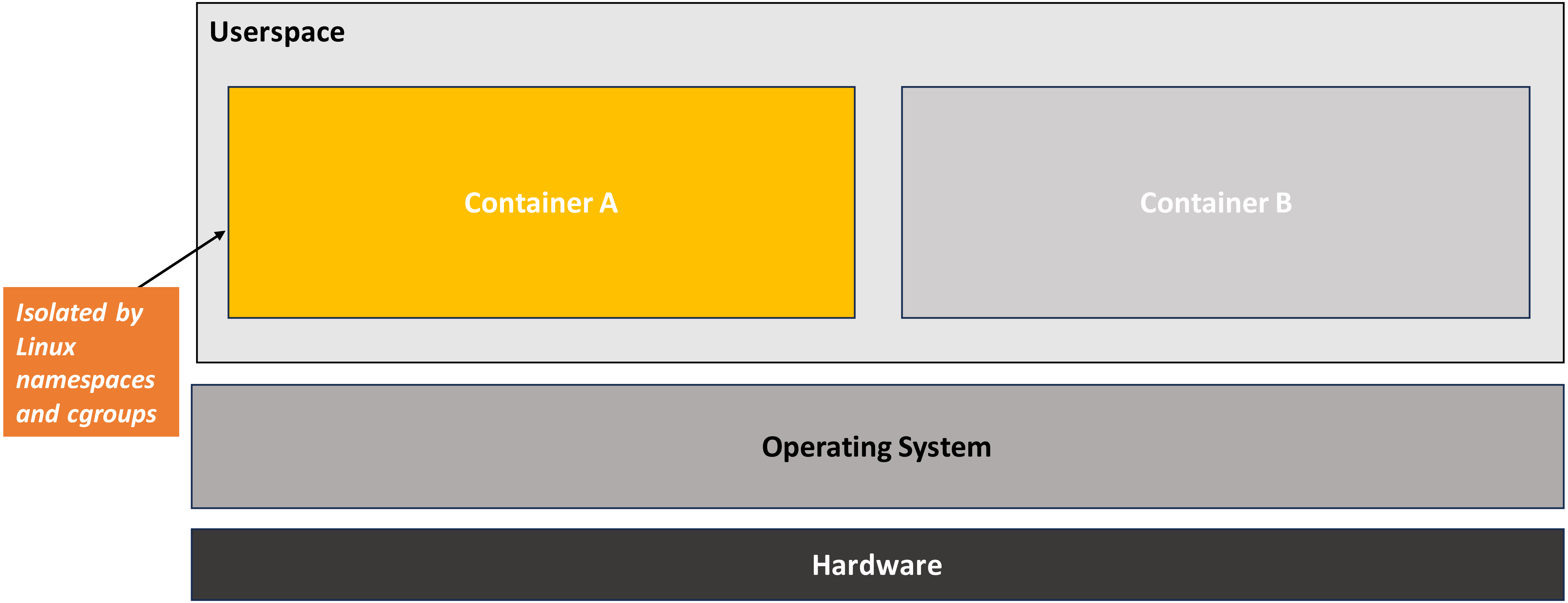
Container Image Integrity

- Goal: allow of attestation of ***container images***

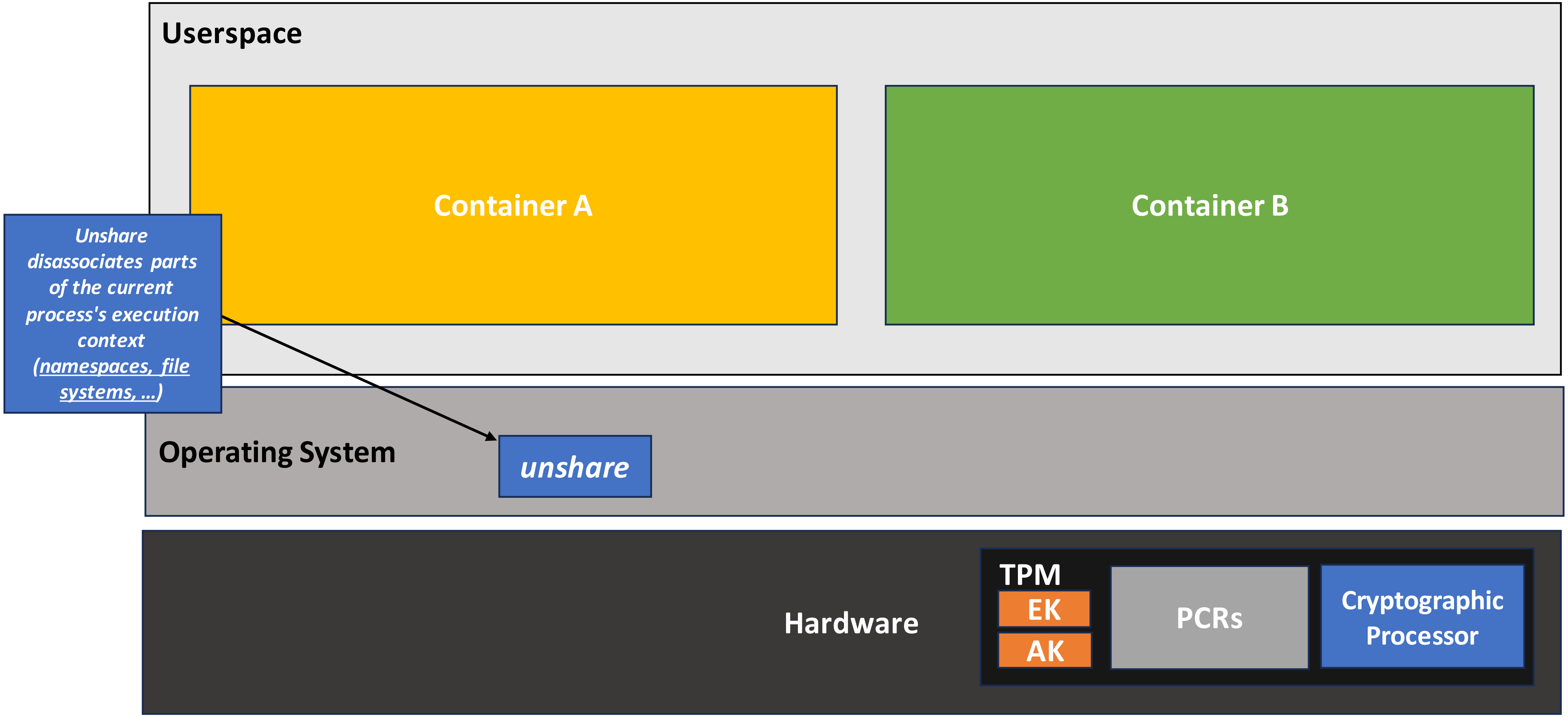


Container Image Integrity

- Goal: allow of attestation of ***container images***

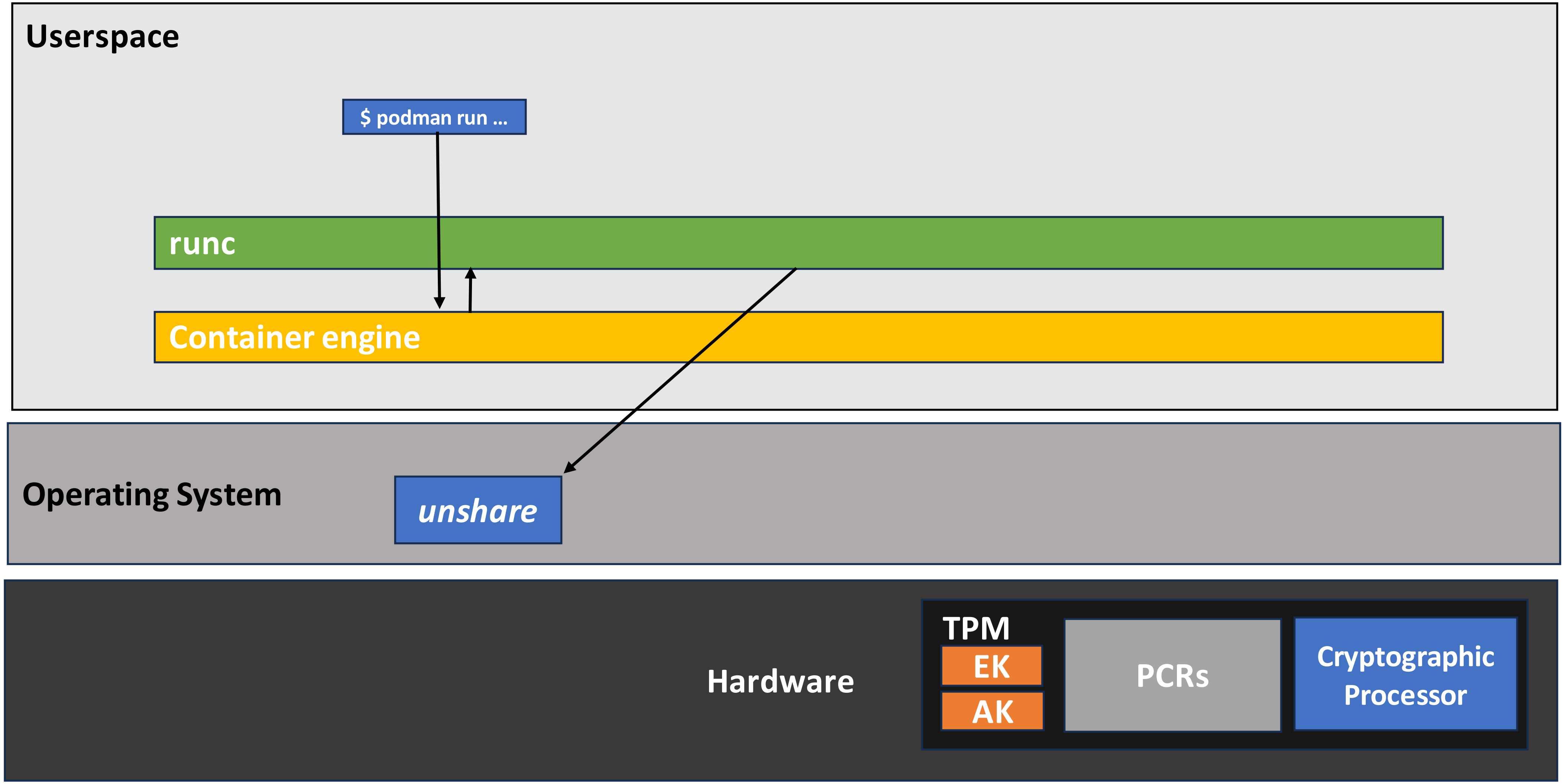


Visibility of Container Creation in the Kernel

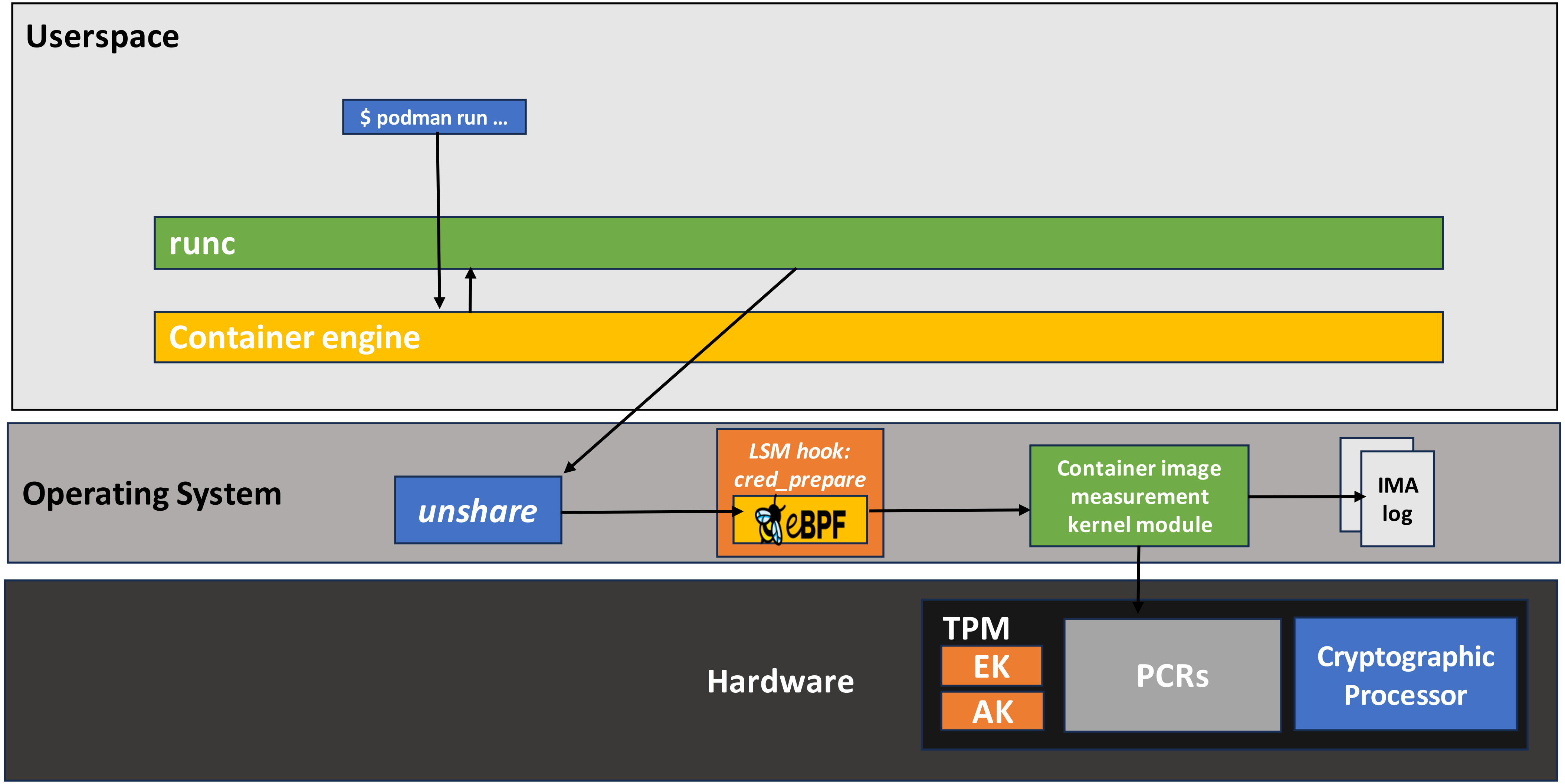


Unshare disassociates parts of the current process's execution context (namespaces, file systems, …)

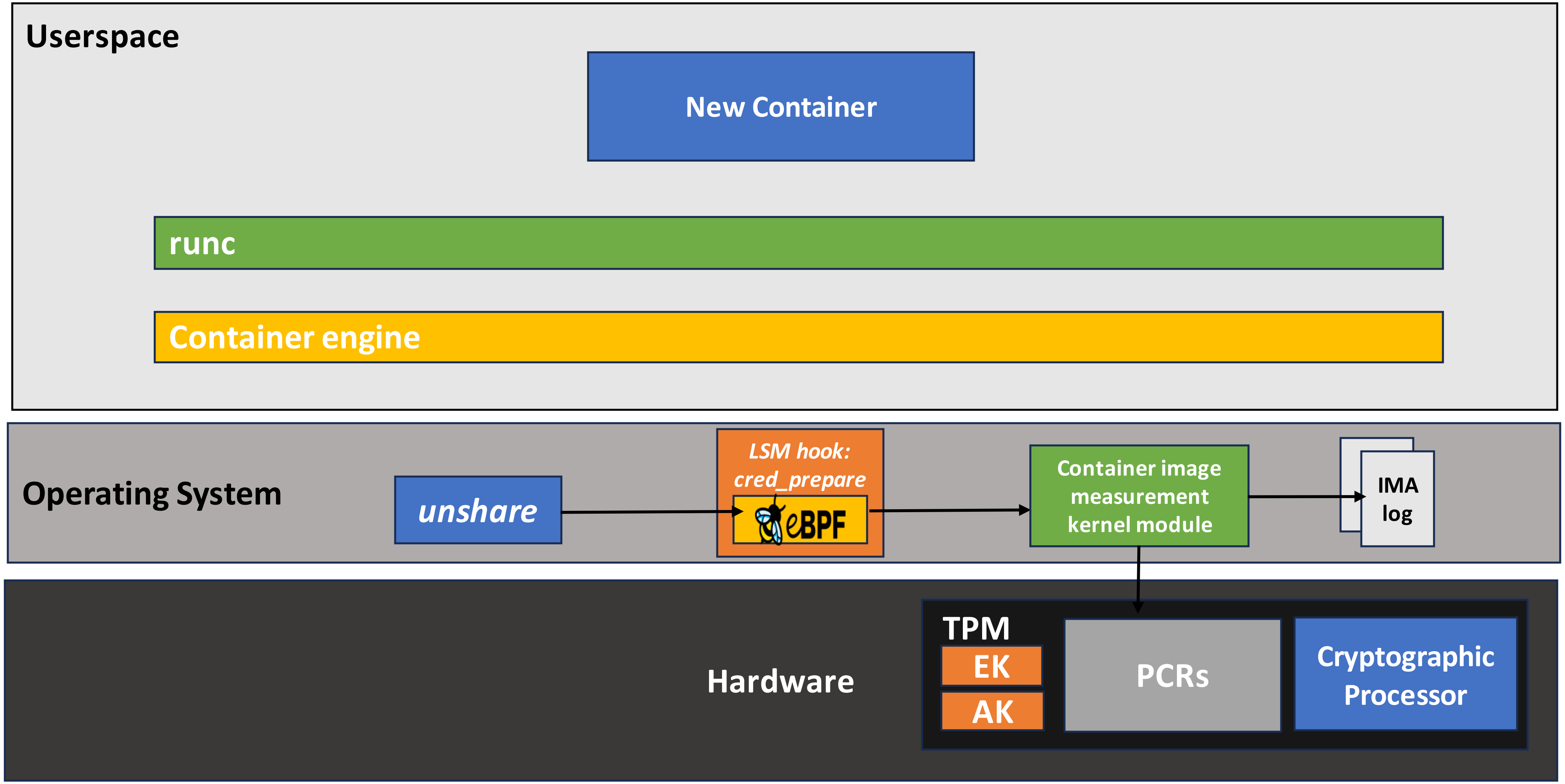
Visibility Through the Unshare System Call



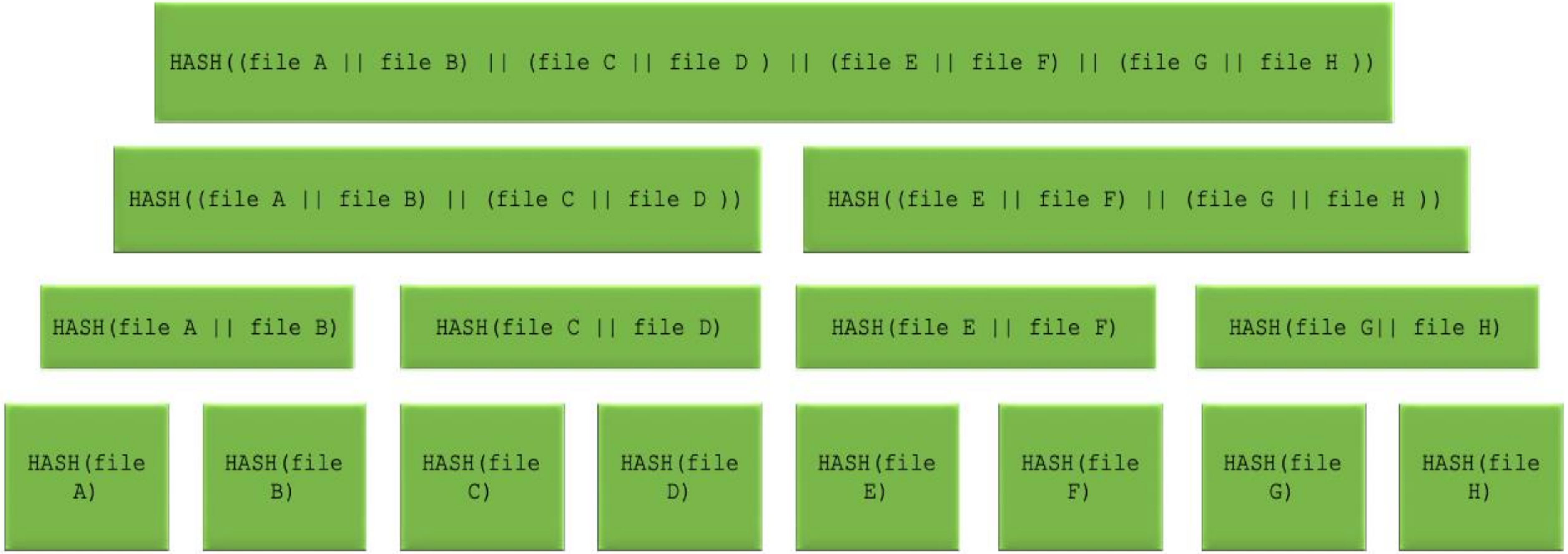
Visibility Through the Unshare System Call



Measuring Container Images From the Kernel



Kernel-Verifiable Image Digests



HASH((file A || file B) || (file C || file D) || (file E || file F) || (file G || file H))

HASH((file A || file B) || (file C || file D))

HASH((file E || file F) || (file G || file H))

HASH(file A || file B)

HASH(file C || file D)

HASH(file E || file F)

HASH(file G|| file H)

HASH(file A)

HASH(file B)

HASH(file C)

HASH(file D)

HASH(file E)

HASH(file F)

HASH(file G)

HASH(file H)

Extending the IMA log with Image Digests

- TODO

Evaluating Measurement Overhead

- EVALS TODO

Demo Environment

- TODO

Enabling Container Integrity Verification

- Current image digests are dependent on images layers, manifest files, image ids, ...
- Kernel-verifiable digests need to be provided to extend the chain of trust from hardware up to each container instance

Building trust in containers through image integrity measurements leveraging trusted hardware.

avery.blanchard@duke.edu

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