# Towards better Code Integrity on Linux.

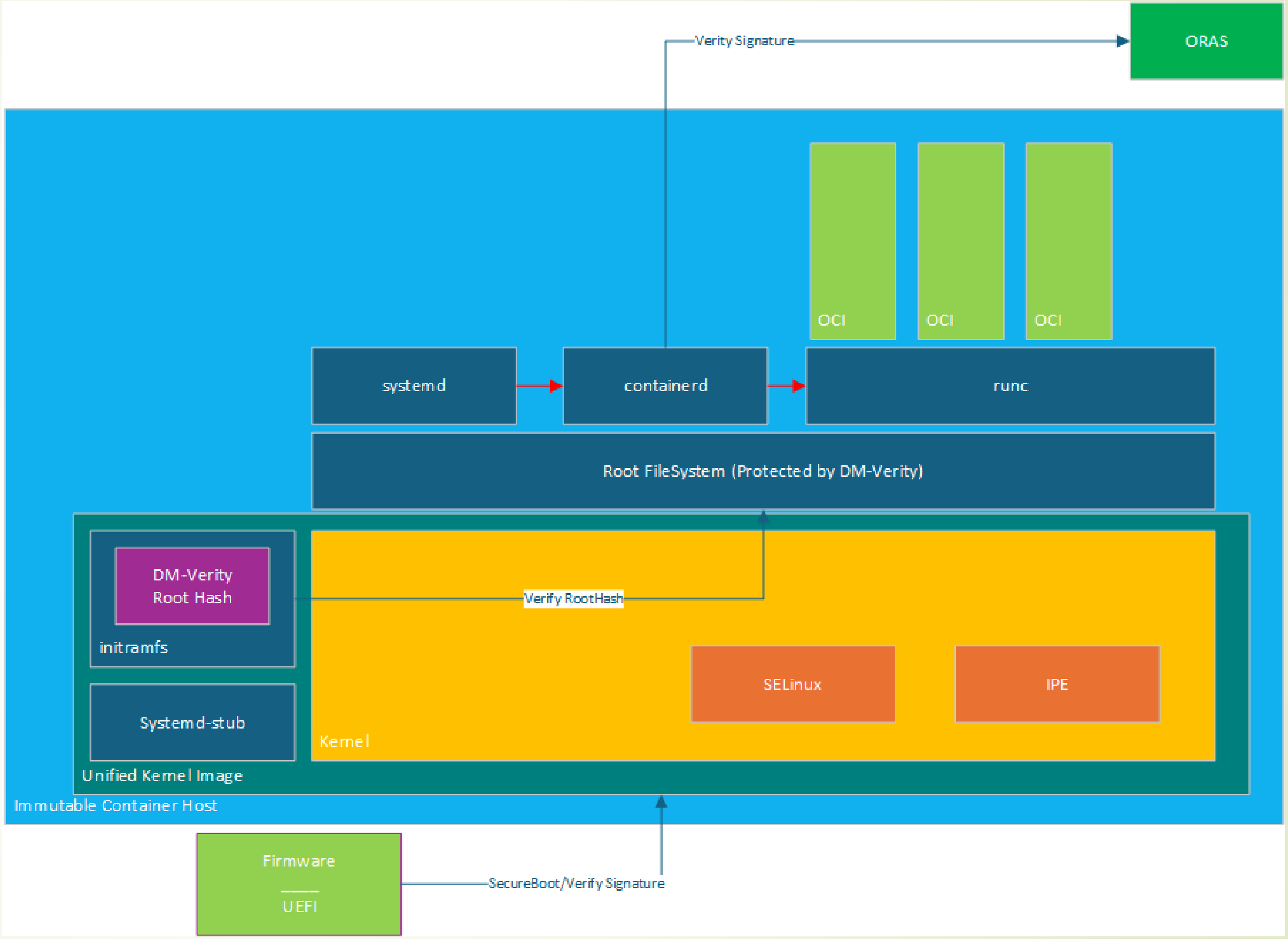
Azure Linux @ Microsoft Corporation

# Motivation

- Improve the security posture of hosts running Linux on Azure.
- Improve the security posture of hosts running container workloads on Azure.
- Improve Code Integrity for Container Host running Azure Linux.
- Improve the security posture of hosts composed of Azure Linux.
- Protect Container Host O/S from users breaking out of privileged containers.

# Immutable Container Host

- Secure Boot
  - Firmware verifies Signature of UKI
  - UKI contains Kernel, initramfs and system-stub
  - Initramfs contains DM-Verity root hash
- Kernel verifies DM-Verity Root hash before mouting RootFS
- Kernel has IPE & SELinux LSMs
- Containerd verifies signatures of OCI container workloads.



# Integrity Protection Enforcement (IPE)

- IPE is a Linux Security Module that is loaded in the Linux Kernel.
- IPE leverages the immutable system properties of system components to provide access.
  - It establishes provenance to the immutable nature of filesystem components by their support through DM-Verity, FS-Verity or their origin as being part of the initramfs (which is verified by the bootloader).
- IPE Policy File is provisioned through *securityfs* and its signature is verifiable through the kernel’s trusted key ring.
- IPE provides protection from tampered filesystem components.

# Tamper Protection for OCI Workloads

- Containerd will verify the signatures of the OCI container workloads that it downloads from OCI Registry As Storage (ORAS).
- The layers of the OCI containers will be protected through DM-Verity. Therefore, the layers are protected from offline tampering after being deployed on the Container Host.
- We have implemented a snapshotter that provides DM-Verity support for OCI containers.
- The OCI containers share the Kernel with the Contianer Host.
  - The IPE LSM loaded in the Kernel can provide security for the host as well as the containers through a single policy.
  - In the future, IPE policy can be extended to support namespaces.

# Research Topics

- Can the Kernel be allowed to trust out of tree drivers that are provided through an immutable volume trusted through IPE?
  - The key used to sign the drivers may not be chained to any of the keys in the Kernel's trusted key ring.
  - By including the out of tree driver in the volume trusted through IPE, the team intends to extend the trust given to the image attester (entity that signed the UKI and the IPE Policy) to the out of tree driver.
  - The alternative is to use the counter sign the key that was used to sign the out of tree kernel module by a key that is part of the kernel's trusted key ring, and add the resultant key to the kernel's secondary key ring.

# References

- IPE [ https://docs.kernel.org/next/admin-guide/LSM/ipe.html ]
- UKI [https://wiki.archlinux.org/title/Unified_kernel_image
- ORAS [https://github.com/oras-project/oras-go ]