Confidential Computing with Secure Execution (IBM Z)

—

Jakob Naucke
Cloud Development for Linux and OpenShift
on IBM Z & LinuxONE

IBM Secure Execution for Linux

IBM Z & LinuxONE/ s390x/“mainframe” used for Red Hat OpenShift workloads

Hardware confidential computing support since z15 (September 2019) & LinuxONE III

Necessarily based on Linux KVM virtualization

Other virtualized confidential computing technologies include IBM Power’s PEF, AMD SEV, and Intel TDX

How do you know your workload runs in a secured context?

```
$ ssh my-secure-domain
Hi, this is the motd from your
cloud provider! I am totally
running this inside Secure
Execution!
```

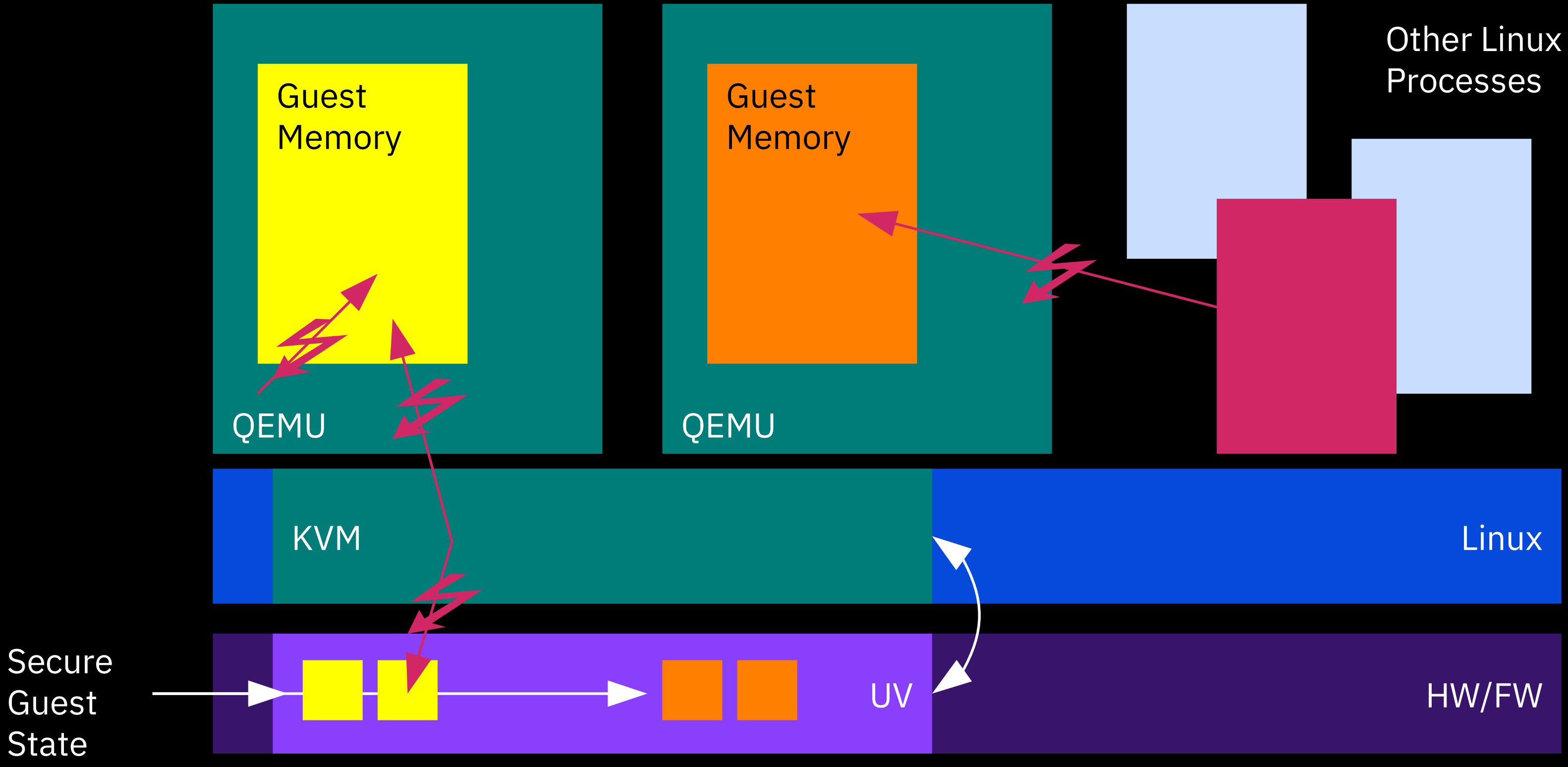
This can be achieved through attestation and smaller encrypted key containers.

Secure Execution (like PEF) relies on a fully **encrypted boot image** that can house anything.

The asymmetric key is tied to the machine and can be verified through a certificate authority.

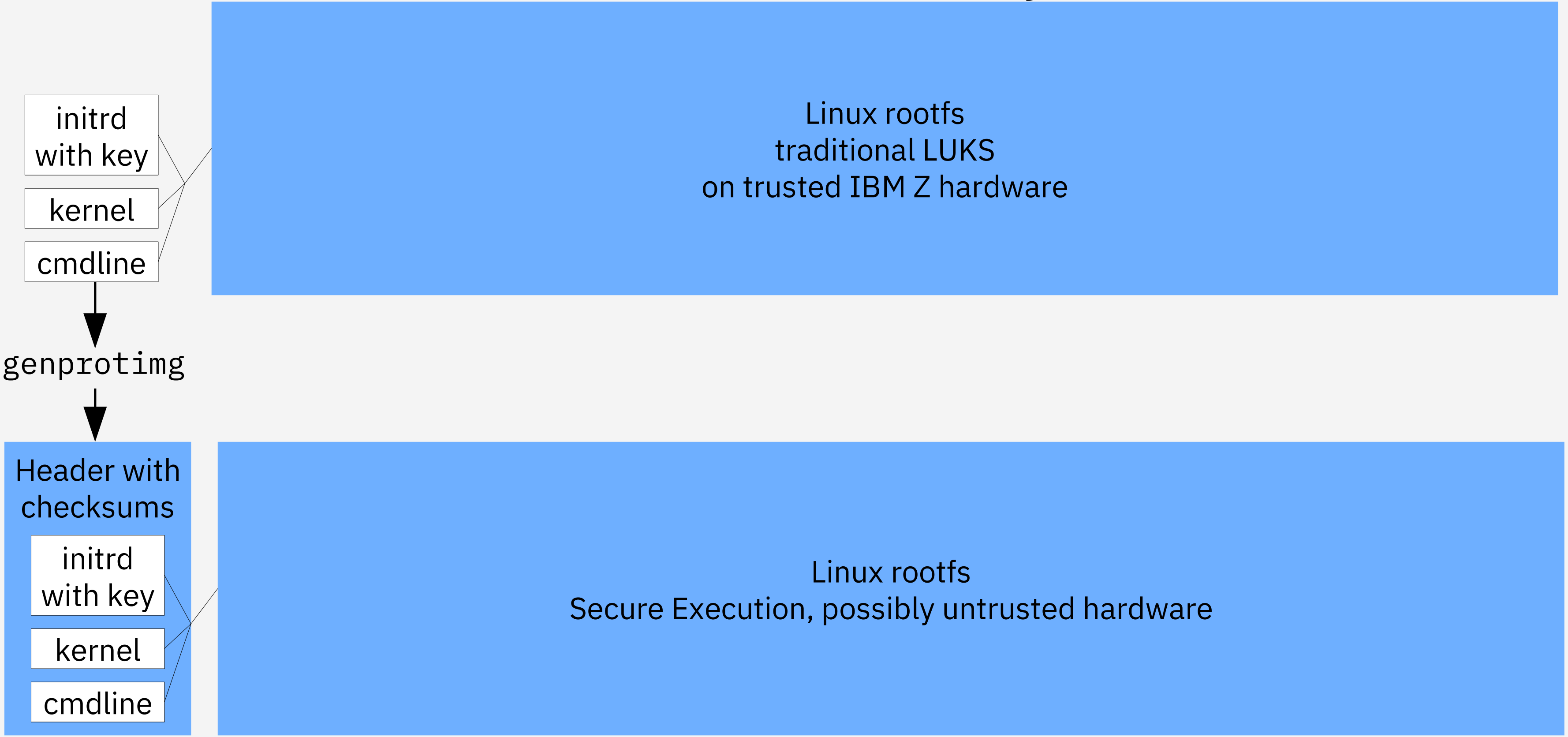
But how can the machine retrieve the private key for decryption? If the *hypervisor* could simply read it, you haven't gained anything.

Enter the Ultravisor



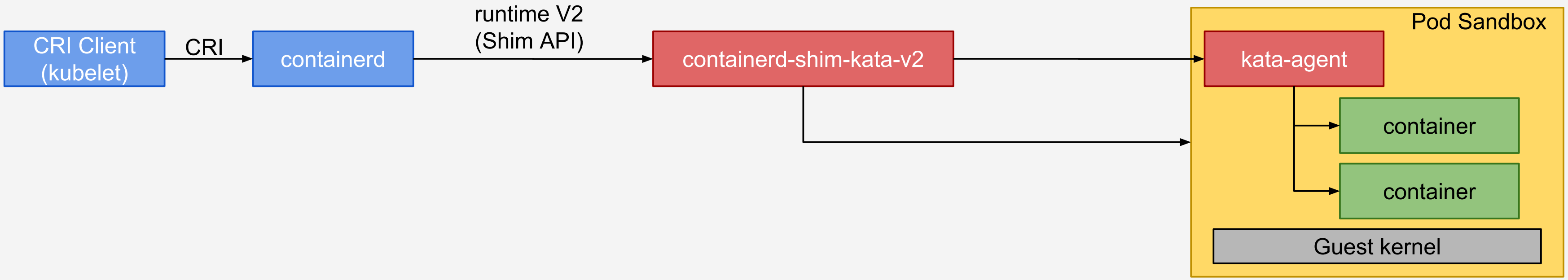
“Classical” Secure Execution

...but what if you want containers?



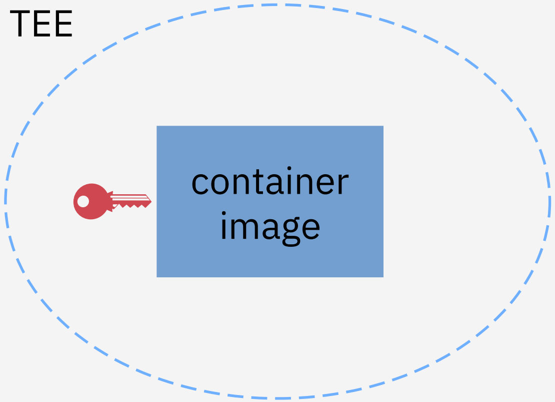
Enter Kata Containers

"The speed of containers, the security of VMs"



github.com/kata-containers/kata-containers, Apache 2.0

How do you achieve confidential computing with Kata Containers?



Utilize hardware. Lock the agent.

As a first, basic solution, we can put anything we might want to use into a custom, encrypted image.

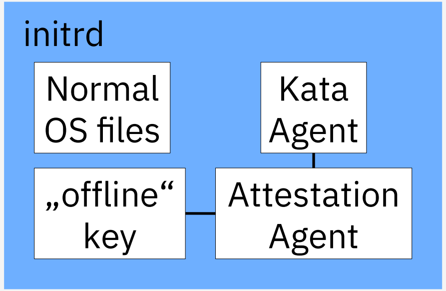
This image is pulled upon creating a container.

Where is the key to decrypt it?

Integrating the current Secure Execution workflow with the Attestation Agent

“Bake-in” approach

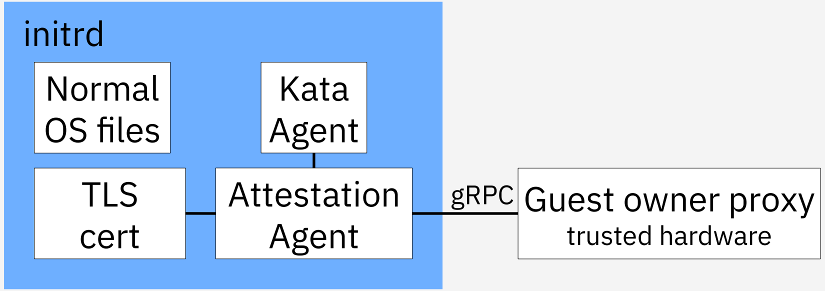
Integrate the keys to decrypt image layers



Simple, but inflexible

“Key fetch” approach

Classical authentication



(Somewhat) more flexible
TLS as substitute for runtime attestation

Thank you

Jakob Naucke
Cloud Development for Linux and OpenShift on IBM Z & LinuxONE
—
jakob.naucke@ibm.com
github.com/Jakob-Naucke
ibm.com