March 2022

The IBM Hyper Protect Platform Generation 1

A technical overview

Table of Contents

I. Introduction

The Hyper Protect Platform, the basis for IBM's Hyper Protect Services, delivers Confidential Computing capabilities for enterprises at-scale in Hybrid Cloud deployments. While the need for protecting data is not new, Confidential Computing refers to data in use protection, a novel technology in the industry. This paper describes the approach taken by the Hyper Protect Platform Generation 1 to provide a secure environment for data at rest, data in motion and data in use. This platform offers privacy of code and data that spans the software and data lifecycles. While protecting data and code, the Hyper Protect Platform also supports a consistent developer experience that does not require special coding efforts in order to establish this level of protection.

This paper references the Confidential Computing concept about which there are various papers and surveys published (i.e. [1], [2] and [3]). The Linux® Foundation launched the umbrella project named "Confidential Computing Consortium" to act as an anchor for open collaboration, industry outreach and education initiatives for Confidential Computing [4]. This paper will thematically classify the context around this concept and elaborate on key considerations for understanding different use cases and business solutions.

This paper will also outline the *underlaying technology* provided by IBM Z® and how the requirements around Confidential Computing are considered when creating protected appliances for logical partitions (LPARs) on IBM Z. The LPAR concept is a premier and well-established technology in IBM Z. The LPAR protects against access from adjacent workloads, and such protection is equally important on the workload/container level within the protected LPAR.

Another concept explained in this paper is how the *Hyper Protect Platform* provides technical assurance. Technical assurance is achieved when a chain of trust and set of encryption keys protect against the system or platform administrator and/or service provider. This form of assurance stands in contrast with operational assurance, a type of insurance offered by most service providers. Operational assurance ensures service providers *will not* access client workloads, whereas technical assurance ensures that service providers *cannot* access client workloads.

In additional to the implementation of technical assurance, the central component of protected virtual servers leveraging this protected platform is also explored. The importance of trustworthy platform software is considered in the context of how this trusted code base is established and maintained throughout the software lifecycle.

This paper concludes with an example of how this technology is used to create a unique key management system (KMS) for a customer with direct access to a Hardware Security Module (HSM) from the protected workload. This example is a central use case to protect a customer's keys, data, workloads, and authenticity.

II. Underlaying Technology

The first generation of IBM Hyper Protect Services is based on a technology called the IBM Secure Service Container [5]. Secure Service Container provides the base infrastructure for integrating the operating system, middleware and software components into an appliance. This appliance works autonomously while providing core services focusing on consumability and security.

Appliances integrated into IBM Z with Secure Service Container technology – referred to as SSC-based appliances – are encrypted and signed for confidentiality and integrity [refer to the middle column of Figure 1]. They inherit the core reliability, performance, and overall platform characteristics of IBM Z. These appliances are also tamper-proof: access is only possible through well-defined REST interfaces over hypertext transfer protocol secure (HTTPS). An appliance administrator may use these APIs to configure storage, network, or crypto resources to appliances. Any other communication attempts are blocked, including any operating system access such as secure shell (SSH).

The IBM Z platform, in addition to the appliance, ensure that only the workload has access to its data, depicted in Figure 1 as the Appliance Workload and the Hyper Protect Workload; the Appliance Workload reflects the underlying capability of building a custom appliance and the Hyper Protect Workload embodies the consumability of SSC technology. These workloads do not permit an appliance or system administrator to access workloads or its data unless specifically authorized. The primary goal of this technology is to provide out-of-the-box protection against disclosure and manipulation of customer data by anyone except for the workload owner. In order to achieve this level of security and protection, the entire stack has been architected, implemented, and tested with a focus on offering a secure yet easy-to-use platform. This combined stack of the SSC-LPAR with the Hosting Appliance running on the IBM Z Hardware and Firmware builds up the Hyper Protect platform for HyperProtect Workloads and Services.

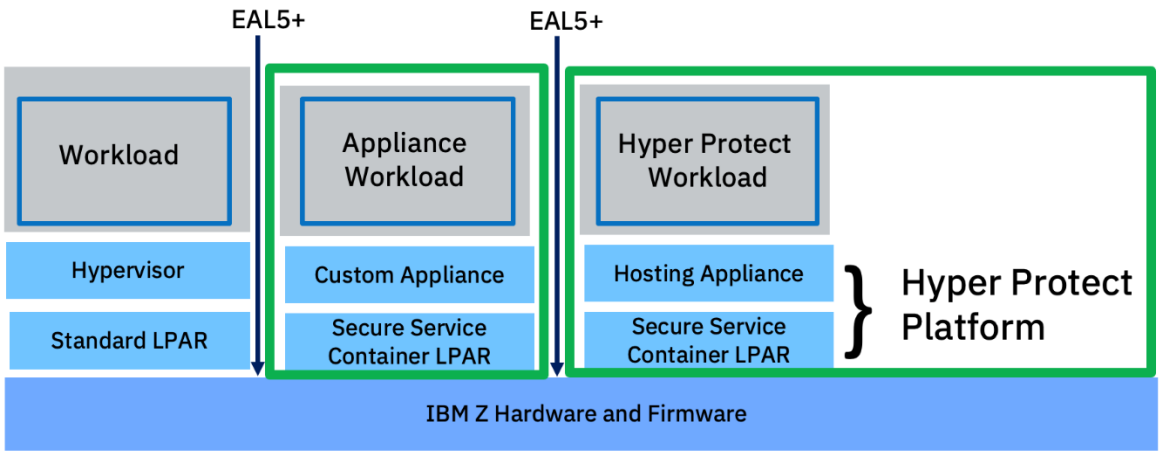

Figure 1 – Protection and isolation mechanisms of the IBM Z platform

Secure Service Container is implemented, in part, as a feature of the IBM Z platform. IBM Z mainframes use LPARs to configure resource sets for specific workloads. Each LPAR can be viewed as a Virtual Machine (VM) used for running a hypervisor or operating system. For Secure Service Container, a specific LPAR type, called an SSC-type LPAR, is available. Using IBM Z LPAR technology, an appliance implemented using Secure Service Container can be securely isolated by leveraging separation mechanisms at the hardware level. This separation is certified with an evaluation assurance level 5 (EAL5), meaning semi-formal design and test.

SSC-type LPARs differ from standard LPARs by offering additional protection to meet tamper-protection and secure access objectives. SSC-type LPARs only boot SSC-based appliances where these appliances are encrypted and protected. A special installer is used to validate and install an SSC-based appliance on the platform. The SSC Bootloader (refer to the next chapter, “The Secure Build Process toward the Hyper Protect Platform”) validates the integrity and originality of an appliance image before booting it. Furthermore, firmware prevents any memory access from outside the SSC-type LPAR. Firmware replaces the regular dump process to ensure an administrator cannot access data running in an SSC-type LPAR.

Besides security, ease-of-use was a major design goal of SSC-based appliances. The usability of this appliance starts with providing a stable and well-tested base including security patches, static and dynamic code scanning as well as penetration testing. By extending ease-of-use into runtime, an SSC-based appliance provides a convenient interface to perform storage, network, and workload configuration without requiring an administrator to know the details of performing these typical configuration tasks.

SSC-based appliances can also collect performance metrics and create audit logs. Since there is no access into the appliance, additional interfaces are available for collecting log and dump information. SSC-type appliances only collect hypervisor-level information: they capture specific logs and have additional filters in place to ensure that no sensitive data leaves the appliance. Memory dumps only contain memory information for kernel space not user space (workloads). Further, dumps are encrypted by a key to which only a small and restricted group of IBM developers have access. Processes have been put in place to ensure access to any debug data is controlled and tracked.

Chain of Trust and Boot Sequence

The chain of trust starts with a signed firmware that prevents tampering of the SSC Bootloader, including the key used to verify the appliance bootloader integrity. The firmware also has a key which allows the firmware bootloader to decrypt the appliance bootloader. This firmware key is wrapped with a hardware key that is kept in the Service Element Smart Cards for IBM z13® and IBM z14® and injected at manufacturing for IBM z15™. Manufacturing keeps the wrapping key inside the HSM.

By design, the firmware only loads the SSC Bootloader in the SSC-type LPAR. When an LPAR is defined as SSC-type, the firmware prevents any access to the memory or CPU content. As illustrated in Figure 8, the decrypted appliance bootloader does not leave the memory vulnerable.

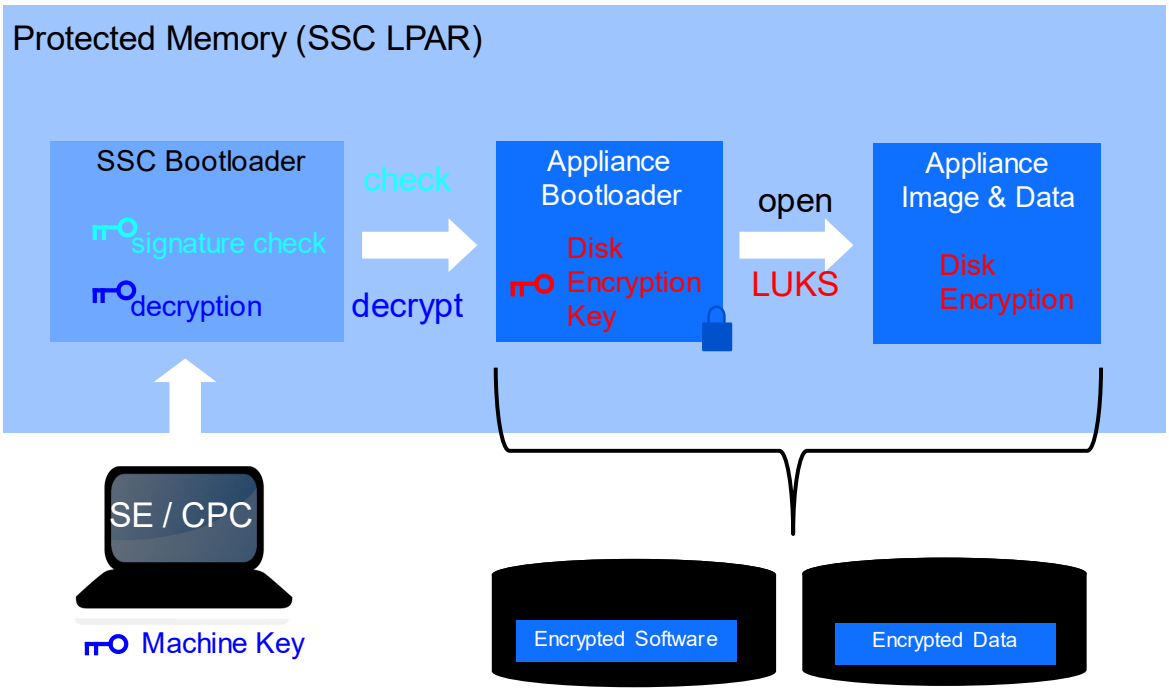

Figure 2 – Boot Sequence

III. Hosting Appliance

Confidential Computing refers to data in use protection, a novel technology in the industry. This paper describes the approach taken by the Hyper Protect Platform Generation 1 to provide a secure environment for data at rest, data in motion and data in use. The appliance used as the base for IBM Hyper Protect Services is called the Hosting Appliance [refer to Figure 3]. The Hosting Appliance provides the capability to create strongly isolated containers while ensuring data security. Data security relies on built-in data encryption with firmware-based encryption keys which are not accessible to the system administrator or those with elevated system credentials. Workloads on the Hosting Appliance are provided in the form of Docker-based images. Several of these workloads are integrated and offered as a service, such as a Hyper Protect Crypto Service, Hyper Protect DBaaS Service, or Hyper Protect Virtual Server. These services use industry-standard OCI images, and the Hosting Appliance provides convenient APIs to allow the deployment of these images.

The Hyper Protect APIs and lifecycle are designed to prevent access and facilitate loading of signed and encrypted workloads. Workload responsibility consists of providing APIs that do not leave the data within the workload as well as following proper security best practices (Security and Privacy by Design, PSIRT handling, Secure Build) to ensure the workload cannot be compromised.

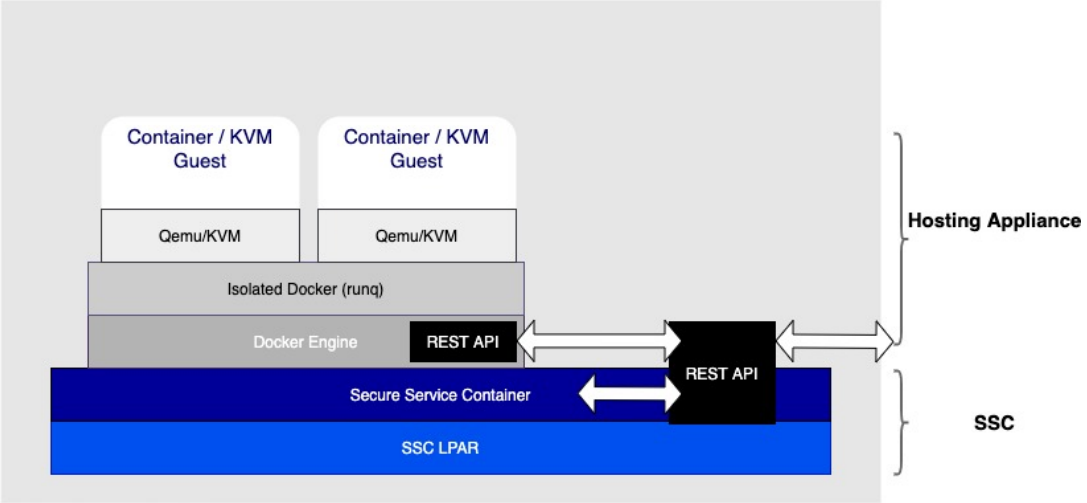

Figure 3 – Hosting Appliance

All persistent storage used by the Hosting Appliance, or its workloads is based on storage pools [refer to Figure 4]. These pools are based on logical volumes in data pool (LVM) running in RAID1 mode. One storage pool, called the SSC Root Pool, is used to store the appliance image / code. The other pool, the Appliance Data Pool, is used to store configuration and runtime data of

the appliance in addition to workload related data, including Docker images. The Appliance Data Pool is also used for the creation of Quota Groups (discussed further on page 10), which are logical volumes that can be used for containers and workload data.

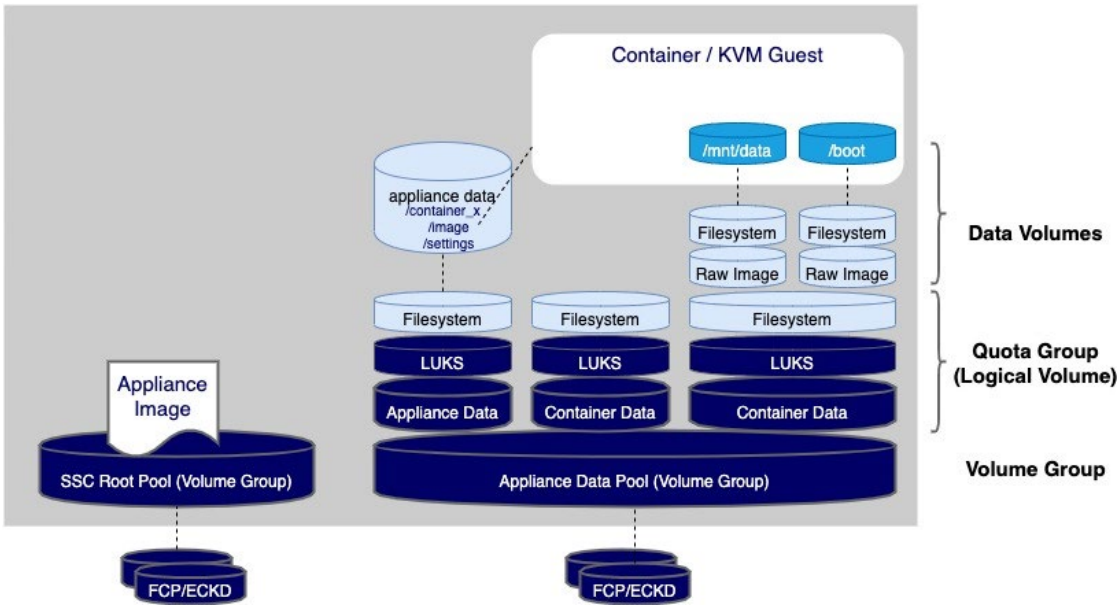

Figure 4 – Hosting Appliance Storage Stack

Data Protection

The appliance itself is a regular embedded Linux build without ssh. Management APIs design was carefully implemented to operate while avoiding threat such as disallowing ssh, capturing debug data in first errors, auto recovering, and more, all without granting an administrator access to the embedded Linux.

The Hosting Appliance uses dm-crypt (LUKS) with random keys for data at rest protection for boot and data volumes. Keys are created and managed automatically within an appliance and transparent to the user or administrator of an appliance. Optionally, Hosting Appliance offers Bring-Your-Own-Key (BYOK) for data volumes. This feature is particularly important if an administrator wants to be able to revoke access to data by protecting volumes with an external key.

Additionally, Hosting Appliance uses btrfs for most of its internal data and data volumes. Btrfl offers better protection and detects tampering even during runtime. An exception to this protocol is the boot partition, where the Appliance Bootloader is stored and the signing as well as encryption is done for the individual files (kernel, initrd, kernel parameters).

IV. Encryption Flow and Keys

This section will look at encryption flows and keys, starting with the key pairs for data protection. A description of Registration Files, Quota Groups and then OCI Containers will follow. To begin, the encryption and protection of secrets is based upon three keys, which are inserted at build time [refer to Figure 5].

The **appliance-to-appliance key pair** is used to protect data created to be stored by an appliance for later retrieval. Any backup, for example, is protected by this key pair. Only an appliance of the same type has the same key pair and is able to restore an entire appliance with its configuration.

The **appliance-to-vendor key pair** is used to sign and encrypt data to be sent to IBM. A virtual appliance encrypts first failure data capture (FFDC) information for diagnostic purposes. Only specific IBM developers have access to decrypted FFDC data.

The **vendor-to-appliance key pair** is used for sending data to a virtual appliance. This key pair is for example used for the registration of workloads with a virtual appliance. In order to create and run a workload, an appliance administrator needs to register the workload via a special file. This special file is encrypted and signed with the vendor-to-appliance key pair. The entire process ensures that only certain workloads and updates to these workloads get access to data created by the workload.

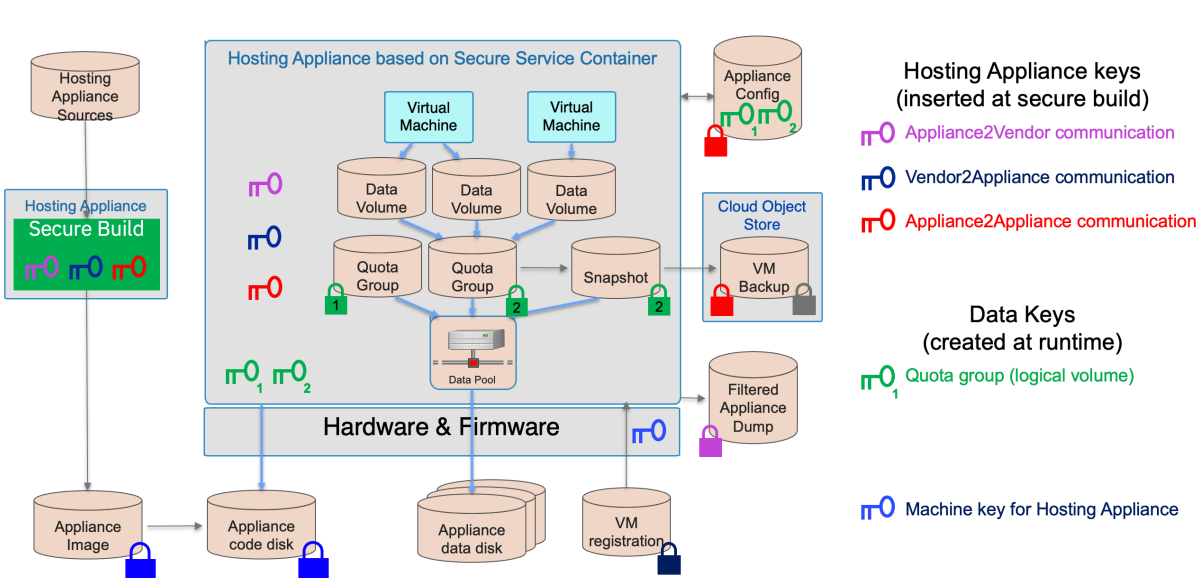

Figure 5 – Code and data encryption flow

Registration Files

Prior to loading container OCI images, they need to be registered within the Hosting Appliance using a matching registration file [refer to Figure 6]. The registration file contains several components including: 1) the public key used to sign the OCI image, 2) the URL and credentials required to download it, 3) the required data volumes, 4) their mount points and 5) the allowed environment variables. Optionally, secrets such as TLS certificates might be included in the registration file. These are made available to the container at runtime, allowing the container to securely communicate with other containers or prove it is of a particular vendor running in a Hyper Protect enclave. This process is possible because the encryption chain protects the secret from being leaked to the administrator. The registration file is then signed with same key as they OCI image and encrypted by a vendor-to-appliance key so that only the Hosting Appliance can decrypt it. Using the registration file, OCI images cannot be tampered with or started outside of the Hosting Appliance.

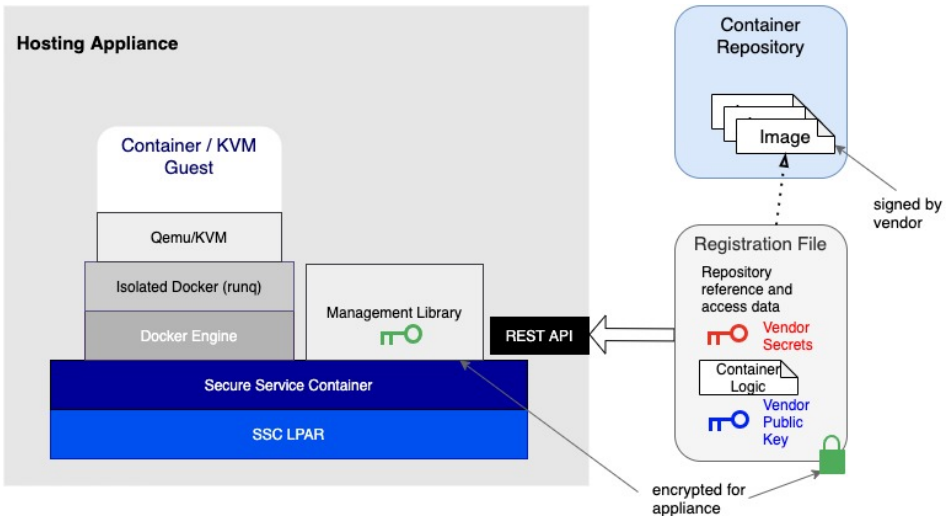

Figure 6 – Registering new containers with a registration file

Quota Groups

Quota groups are logical volumes that can be used for containers and workload data [refer to Figure 7]. The appliance provides APIs to create additional logical volumes in data pool (LVM) to store container data volumes. These volumes are encrypted with unique random LUKS keys and are stored in the data settings LVM. Quota groups support snapshotting and exporting encrypted snapshots. They do this by using a random key encrypted with the appliance-to-appliance key and a password provided by the administrator. This allows for disaster recovery implementation while ensuring the quota group data is only accessible by the Hosting Appliances.

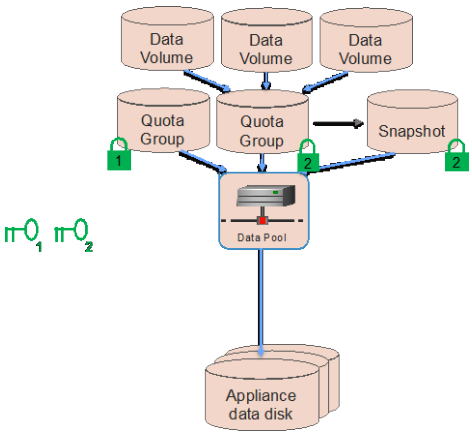

Figure 7 – Quota groups and Disk to data volume mapping

OCI Containers

The Hosting Appliance includes a container runtime based on runq. Runq was created to run each container in a dedicated micro VM, and therefore reduce the risk of a malicious container breaking into the Hosting Appliance. The Hosting Appliance pins each data volume to the container signing key, significantly ensuring that data can only be accessed by the containers that created them.

According to the Confidential Computing definition, each OCI container running in runq inside the Hosting Appliance is a KVM guest running in an enclave (LPAR). Because the KVM guest is managed using an OCI runtime, it is possible to deploy existing OCIs to the enclaves so far they have been compiled for s390x (as are all official Docker Hub containers). The KVM guest can scale from 1 vCPU and 16 MB of RAM to 240 vCPUs and 16 TB of RAM. Additionally, based on the quota groups, hundreds of TBs of persistent and encrypted volumes can be attached to a single OCI container.

OCI images are previously registered with a registration file and then managed with an API to start the containers. OCI image signing keys and their content may be protected with the OCI Secure Build Process described earlier in this paper. Finally, since the Hosting Appliance run in micro VMs and are not allowed to share any data volumes, it does not assume trust between individual containers.

V. The Secure Build Process towards the Hyper Protect Platform

This section describes how the chain of trust is built for the Hosting Appliance based on Secure Service Container. Based on the established through the trusted Firmware and the SSC Bootloader the following considerations are made to extend the chain of trust towards the HyperProtect Platform.

The Secure Build Process of SSC-Based Appliances

To ensure the security and integrity of SSC-based appliances, another important component is needed: the Secure Build process. The Secure Build process provides an environment for building, encrypting and signing a virtual appliance, as depicted in Figure 2. Encryption is done based on an IBM platform key, ensuring only the SSC Bootloader can decrypt and start a virtual appliance image. Similar to a virtual appliance, the Secure Build environment is a protected, and secured environment – allowing a build administrator to only perform well-defined actions. Examples of the secure build process include: the building of a new appliance (based on a new appliance key set), updating an appliance, or rolling the appliance key set for an appliance.

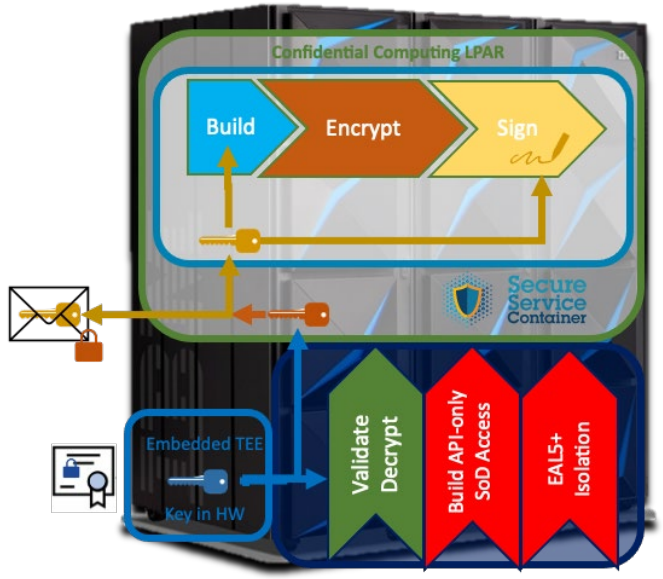

Figure 8 – The Secure Build Process

Hosting Appliance Bootloader

At build time, the Hosting Appliance Bootloader is signed and encrypted using keys that are kept inside the Hosting Appliance Secure Build. The various keys are described here:

- The encryption key is randomly generated and then encrypted with the machine public key
- The signing is done with a key that was generated and persisted within the Secure Build. This key gets periodically rotated
- The signing key is put into a vendor record and signed by the firmware team in a secure room with the firmware production key for SSC verification. The vendor record includes expiration dates and whether or not the record is valid for certain machine Serial Numbers.

The Secure Build runs within a Hosting Appliance as part of IBM Continuous Integration. At the same time, the new Hosting Appliance is built and encrypted with another random key which is then inserted in the Appliance bootloader prior to encrypting it. This protocol ensures that the Appliance decryption key cannot be accessed from outside the SSC-type LPAR.

VI. Leveraging a Protected Workload Platform

Secure Build Server (SBS)

Secure Build Server (SBS) provides the capability to build a trusted container image within a secure enclave provided by Hyper Protect Virtual Servers. The SBS running within an enclave is highly isolated, meaning developers can access the container only with a specific API. A cloud administrator cannot access the contents of the container, ensuring that the created container image can be highly trusted. The SBS process starts by cryptographically signing the image and employing a manifest. The manifest verifies whether what is running on the product system matches with what is being built by the SBS. A manifest also contains the source code as well as build scripts, build logs and git commit logs, allowing this material to be available for audit purposes.

Since the enclave protects signing keys inside the SBS, the signatures can be used to verify whether the image and manifest are from the SBS. Docker Content Trust (DCT) is used to establish trust between the consumer and image producer. It also ensures the keys for signing DCT never leave the SBS. DCT is well supported by container registries such as Docker Hub or IBM Cloud Container Registry (ICR). Additionally, an encrypted registration file is used as a workload specification, thereby ensuring that the administrator has no way of knowing about the workload or its metadata.

Bring Your Own Image (BYOI)

BYOI provides a mechanism for users to securely deploy workloads on to Hyper Protect Virtual Servers, making it possible to deploy any OCI image. These images may be existing images on a container registry, new images built with base images or those built with the SBS process described above. BYOI allows developers to configure container images per workload requirements. The BYOI requires a registration file, which specifies the container registry, the container image, and the required credentials. The registration file is encrypted using a provided public key and only can be decrypted by Hyper Protect Virtual Servers.

The cloud administrator cannot access the information that is included in the registration file, namely the container registry credentials, because the registration file is encrypted. Consequently, the cloud administrator cannot download or access the application image or any secrets that are contained in the image as long as the container registry access controls are correctly set up and the credentials are not exposed.

VII. Crypto Card Management within the IBM Hyper Protect Platform

The IBM Cloud Hyper Protect Crypto Services provides Key Management System (KMS) and Hardware Security Module (HSM) capabilities to protect key material using a FIPS 140-2 Level 4 certified HSM. Using Secure Service Container, it is possible to isolate each customer within the enclave and attach a dedicated HSM domain. The FIPS 140-2 Level 4-certified hardware (using IBM CEX6S and CEX7S) provides full key access to the customer. All key material is protected by the HSM and not vulnerable to the public.

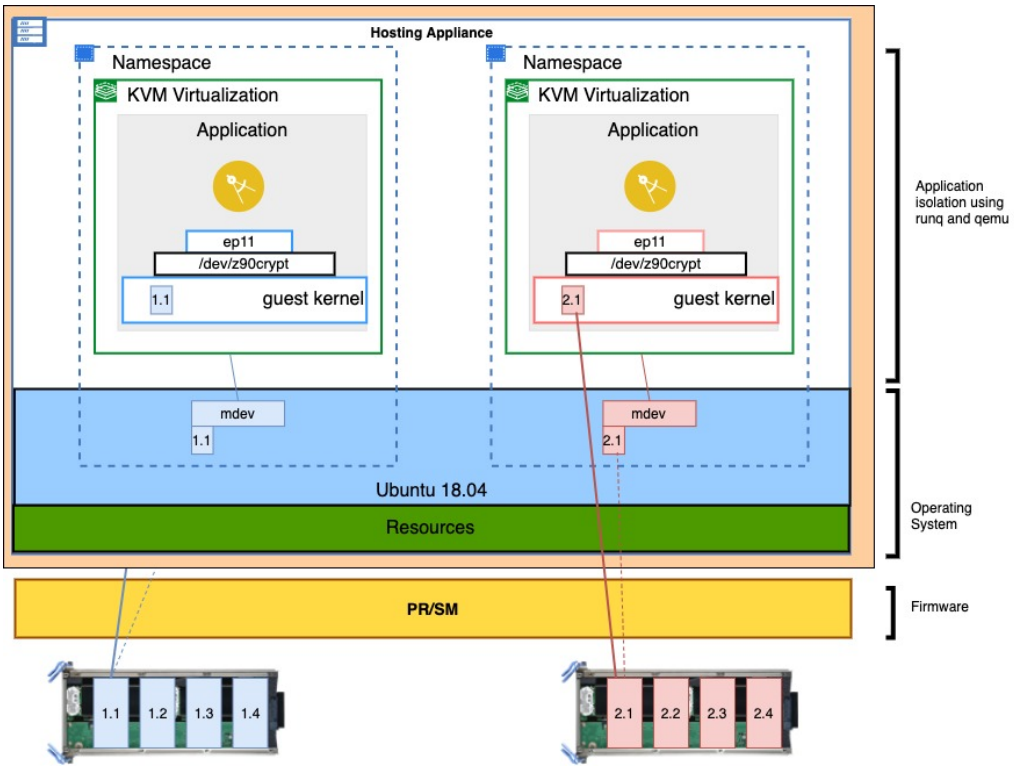

Figure 9 – Crypto Adapter passthrough to the Hosting Appliance on IBM LinuxONE

Two customers running applications within an enclave can access their own exclusive domains on attached HSMs. The data exchanged between application and crypto cards are protected from the underlying Host. The application module builds a passthrough method that passes the data directly to the underlying firmware/millicode while bypassing the Host [refer to the schematic runtime in Figure 9].

Whenever the guest issues one of the three crypto instructions, the millicode verifies the valid access to the card via the KVM and LPAR. Only if both levels mark an adapter/domain tuple as being active, access is granted. As these instructions are implemented in millicode, the hypervisor is not involved in any communication between the guest OS and the crypto hardware.

Instead, the millicode will interact with the system firmware to pass along the cryptographic requests and return the results when the guests queries.

Using the passthrough method and the Secure Service Container enclave, customer data is protected from an end-to-end perspective without a cloud administrator being able to manipulate or gain access to the data.

VIII. Summary

The growth of cloud workloads has been shaped by legitimate concerns about data privacy. Now businesses are better equipped to address these concerns with sophisticated encryption technologies protecting data at rest and data in transit as well as a new generation of technology protecting data in use. By protecting data in use, Confidential Computing enables all types of valuable data to be processed in a Hybrid Cloud model. This technology also empowers multi-party collaboration scenarios, ones that have been previously difficult to establish due to privacy, security, and regulatory requirements. Confidential Computing's use of hardware-based techniques to isolate data in use increases its application and potential to become an important attribute of all types of Hybrid Cloud services.

However, not all Confidential Computing solutions deliver the same levels of security, flexibility, and seamless developer support. Learning about the platforms and the breadth of services they offer allows for selection of a platform that supports data privacy and regulatory needs of solutions today and in the future. A key aspect to consider is the end-to-end capabilities of such solutions from integration to deployment and life cycle management. Considerations must include the secure build of a workload, embedded processes to ensure a trustworthy supply chain, separation of duty as mandatory or technical assurance throughout the lifecycle as preferrable concept. IBM Hyper Protect Platform's hardware-based technical assurances provide true data confidentiality and integrity that allows enterprises to safely scale their Hybrid Cloud deployments.

IX. Bios

Stefan Amann - *IBM Systems, Böblingen, Germany (stefan.amann@de.ibm.com).*
Stefan. Amann is Senior Technical Staff Member at the IBM Lab in Boeblingen. He studied Computer Engineering at the Vocational Academy Stuttgart. He graduated in 1993 and then joined IBM at the Research and Development Laboratory in Boeblingen as an R&D engineer. He worked as architect on several projects for IBM Z. He is the lead architect for IBM Secure Service Container, and Hosting Appliance since 2018.
He is author or coauthor of 32 patents and several technical papers.

Rebecca Gott - *IBM Systems, Poughkeepsie NY USA (gott@us.ibm.com).*
Dr. Rebecca Gott is a Distinguished Engineer in IBM Z and LinuxONE, responsible for Hyper Protect Private Cloud, blockchain and digital assets offerings on the platforms. Previously, she had worked across HPC, POWER® and IBM Z, having worked within the hardware development organization for the last 5 generations of IBM Z systems. Dr. Gott received her Ph.D.in Electrical Engineering from Tulane University in 1995 where she focused on signal processing. Prior to joining IBM, she served on active duty in the U.S. Air Force.

Stefan Liesche – *IBM Systems, Böblingen, Germany* (liesche@de.ibm.com).
Stefan Liesche is the Architect for IBM Hybrid Cloud and Hyper Protect Services on Z. Stefan is focused on security, transparency and protection of data and services in flexible cloud environments. Stefan worked in various areas as Technical leader within IBM, most recently as Chief Architect for IBM Cloud Hyper protect Services and IBMs Watson Talent Portfolio where Stefan was building AI driven solutions that transform recruiting and career decisions within global organizations, that not only enhances quality of decisions, but also allows HR functions to enhance fairness and tackle biases. Stefan also innovated within the Exceptional Web Experience products for several years with a focus on open solutions and integration. Stefan has more than 20 years of experience as technical leader, collaborating with partners and customers through joint projects, as well as within IBM's product development organization.

Anbazhagan Mani - *IBM Systems, Bangalore, India (manbazha@in.ibm.com).*
Anbazhagan Mani is a Master Inventor and Senior Technical Staff Member in the IBM Z and LinuxONE. Anbazhagan Mani leads the design and development of Hyper Protect Virtual Servers offering on IBM Cloud and on-premises. He has over 20 years of experience at IBM mainly in the systems management & cloud areas. He holds a masters degree from National Institute of Technology (NIT), Trichy, India.

Nicolas Mäding - *IBM Systems, Boeblingen, Germany (nmaeding@de.ibm.com).*
Nicolas Mäding is a Product Manager at the IBM Lab in Boeblingen. He received his Dipl. Ing. Degree in Electrical and Information Technology at the Technical University of Chemnitz. He joined IBM in 2001 and worked in various development and management positions in IBM Systems Hardware Development. After that he joined the Z-as-a-Service organization as Release Manager of Hyper Protect Hosting Appliance in 2018 and became the Product Manager for the Hyper Protect Platform and Confidential Computing with Linux on Z. He is author or coauthor of 12 patents and several technical papers.

Angel Nuñez Mencias - *IBM System, Böblingen, Germany (anunez@de.ibm.com).*
Angel Nuñez Mencias is Senior Technical Staff Member in the Mainframe Firmware Organization. In 2002, he received an Engineer degree in telecommunications from the Polytechnic University of Madrid and in 2004 an M.S. degree in electrical engineering and computer science from the University of Stuttgart. He completed an M.B.A. degree from the National University of Distance Education in 2005. That year, he joined the IBM Development Laboratory in Boeblingen, Germany. He created the original architecture of the Secure Service Container and is now the lead architect for the Blockchain Solutions on Z. He is author or coauthor of 54 patents and 15 technical papers.

Stefan Schmitt - *IBM System, Böblingen, Germany (st.schmitt@de.ibm.com).*
Stefan Schmitt is a Senior Technical Staff Member in the Hyper Protect Services and IBM Z organization. Stefan is focused on delivering confidential computing services within the IBM Cloud®, like IBM Cloud Hyper Protect DBaaS and IBM Cloud Hyper Protect Crypto services. Stefan has more than 20 years of experience as technical leader, collaborating with partners and customers through joint projects. His experience spans from Security, Compliance, Web Development as well as enterprise grade cloud native applications.

X. References

[1- ARM / PULSE] Confidential Computing: A Pulse Survey on the Future of Security Technology - https://armkeil.blob.core.windows.net/developer/Files/pdf/graphics-and-multimedia/confidential-computing-pulse-survey.pdf

[2-Intel] Confidential Computing: Security for the Cloud Age - https://www.intel.com/content/www/us/en/now/your-data-on-intel/confidential-computing-security-for-cloud-age-article.html

[3-Google (AMD)] Confidential Computing: Encrypt data in-use with Confidential VMs - https://cloud.google.com/confidential-computing

[4-Linux Foundation] New Cross-Industry Effort to Advance Computational Trust and Security for Next-Generation Cloud and Edge Computing - https://www.linuxfoundation.org/en/press-release/new-cross-industry-effort-to-advance-computational-trust-and-security-for-next-generation-cloud-and-edge-computing/

[5-IBM] Integrating solutions on IBM Z with Secure Service Container https://www.ibm.com/downloads/cas/6JWREBWX

IBM Corporation
New Orchard Road
Armonk, NY 10504
U.S.A.
03/22