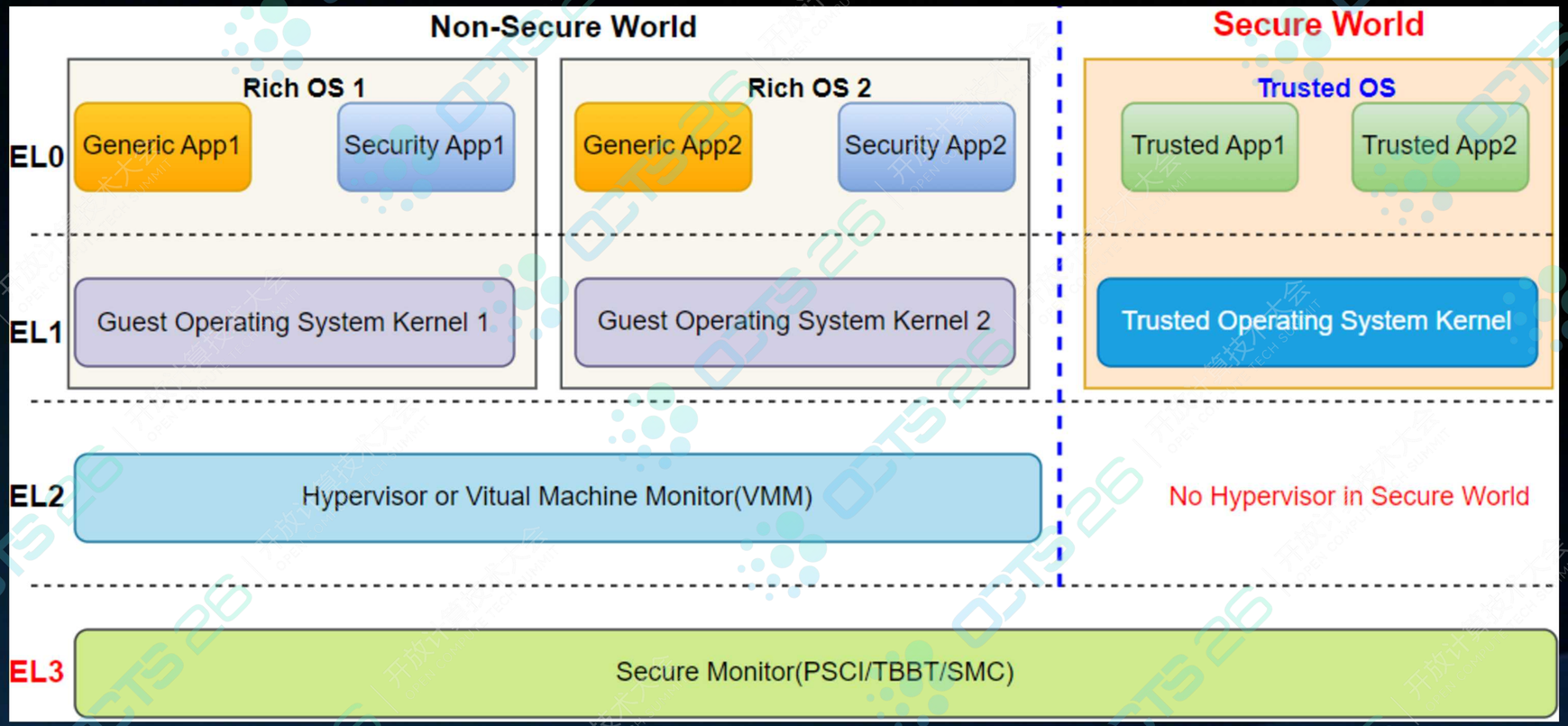
飞腾平台可信BMC的设计与实现

飞腾信息技术有限公司 王敏

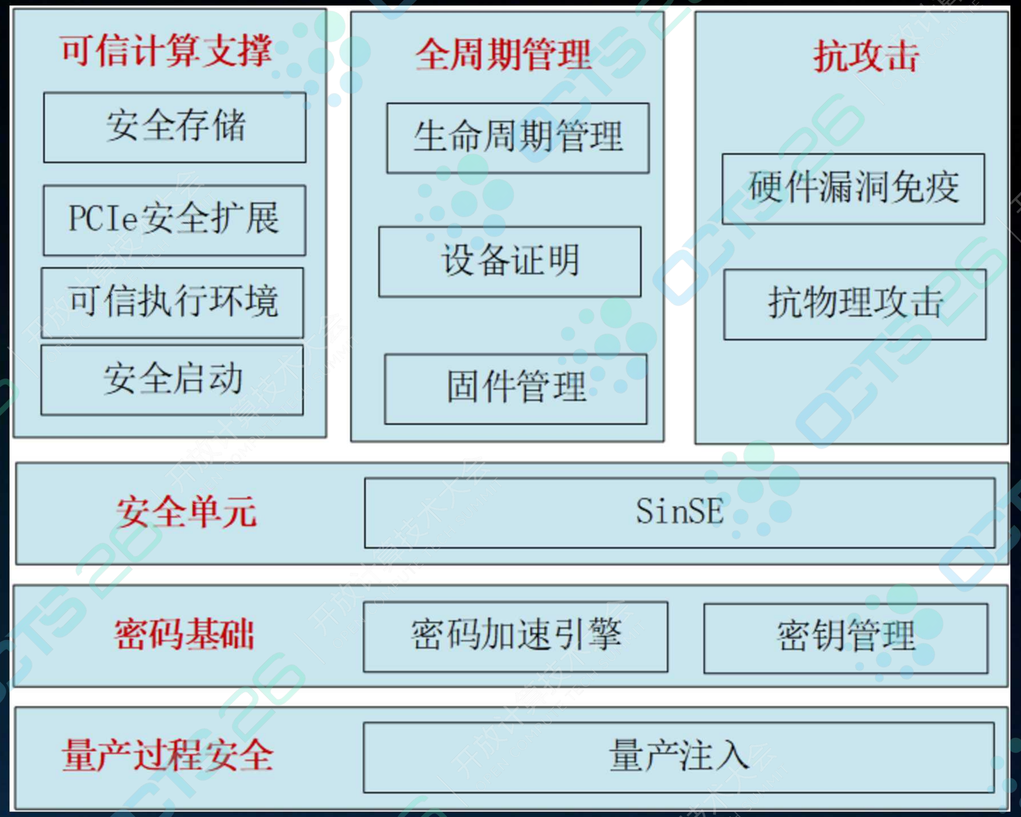
目录

ARM TrustZone



TrustZone 是 ARM 公司推出的系统级硬件安全架构；处理器及整个芯片的资源划分为 非安全世界与 安全世界。

飞腾PSPA规范



飞腾公司推出了处理器安全平台规范 PSPA (Processor Security Platform Architecture)。从下到上，分别定义了六组安全支撑能力。

1. 量产过程安全：量产注入；
2. 密码基础支撑能力：密码加速引擎、密钥管理；
3. 安全单元：SinSE；
4. 可信计算支撑能力：安全启动、可信执行环境、PCIe安全扩展、安全存储；
5. 全周期管理能力：固件管理、设备证明、生命周期管理；
6. 抗攻击能力：抗物理攻击、硬件漏洞免疫。

基本概念

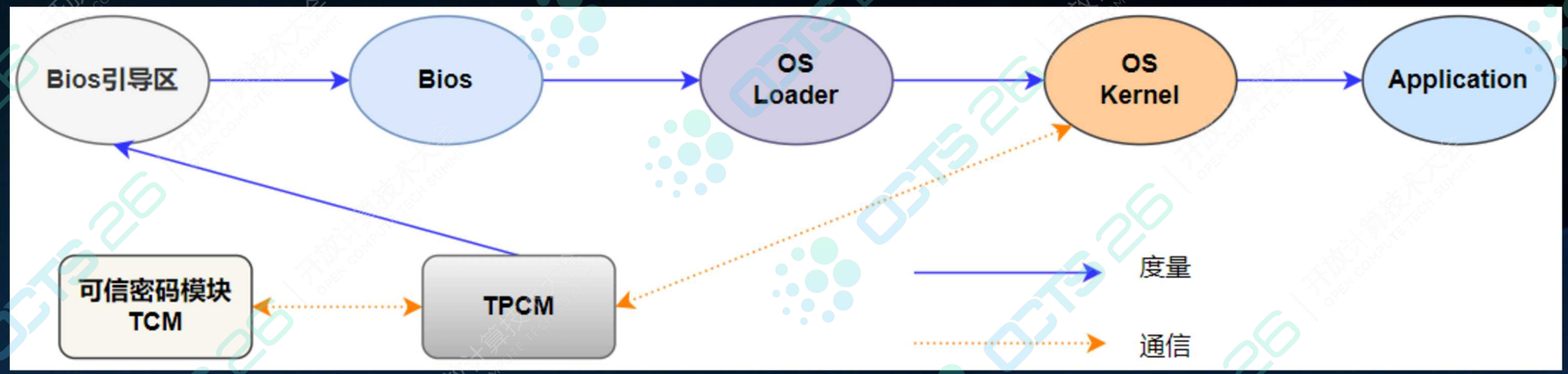
可信根： 可信计算平台的信任源点，由TPCM、TCM和TSB构成，为可信平台提供建立可信链、发起可信验证、执行可信计算、存储安全策略等功能的核心模块。

① **TPCM**（**T**rusted **P**latform **C**ontrol **M**odule）：提供完整性度量、安全存储、可信报告、密码服务等功能。

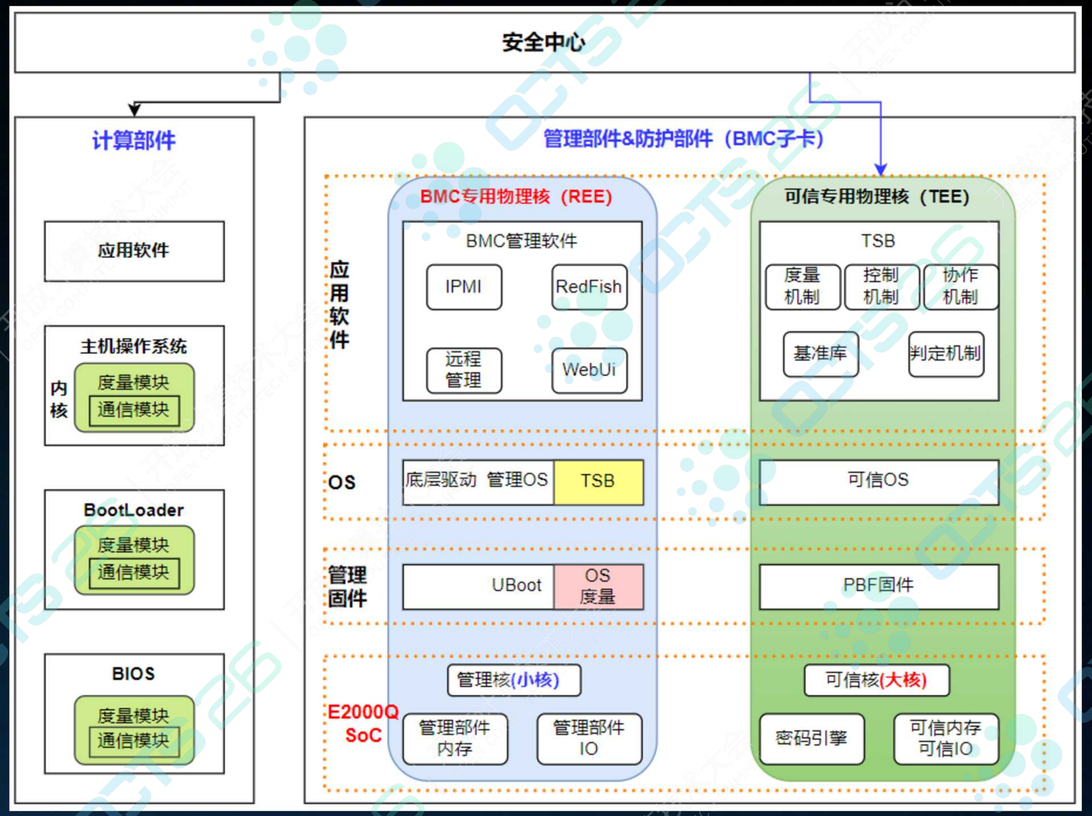
② **TCM**（Trusted Cryptography Module）：提供密码计算、密码服务。

③ **TSB**（Trusted Software Base）：提供可信提供支持的软件集。

可信链：从可信根开始逐级度量下一个执行主体，建立的一种信任传递机制。



总体架构

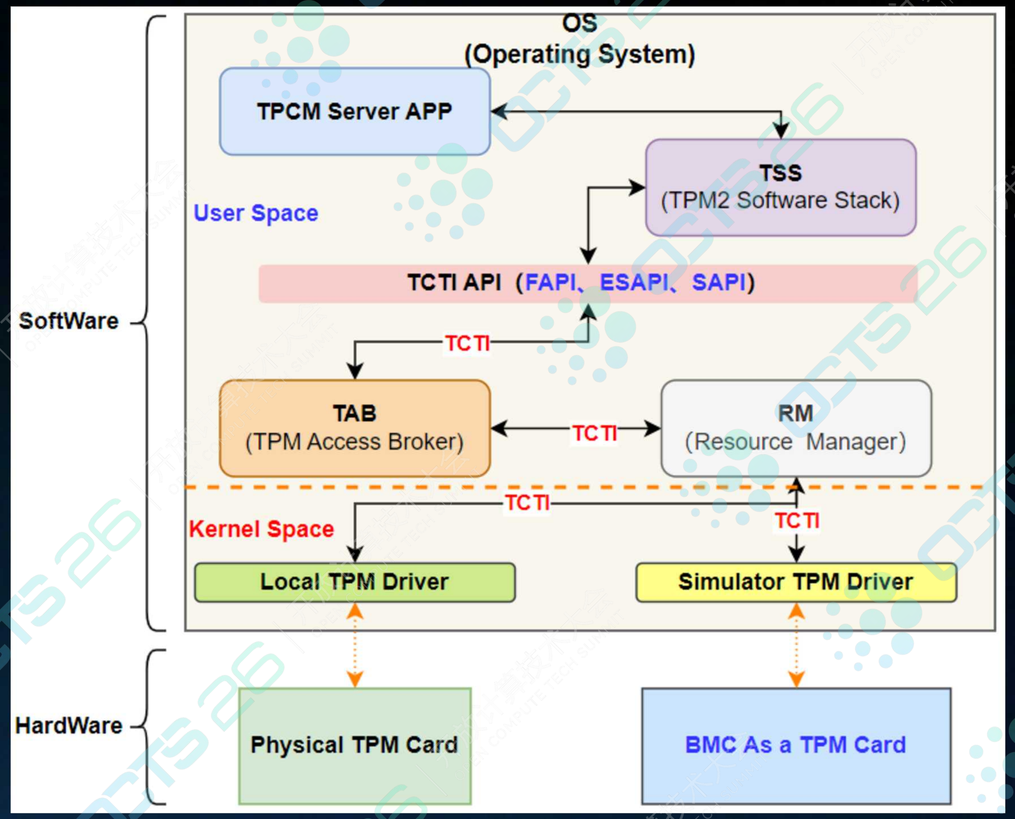


安全性高：基于硬件ARM TrustZone机制，BMC运行在富执行环境，TPCM运行在可信执行环境，具有更高级别的执行权限，保证BMC业务和TPCM业务相互隔离。

扩展性强：BMC侧提供通讯代理，确保主机到TPCM的可行通路，支持部署不同厂商的TPCM服务。

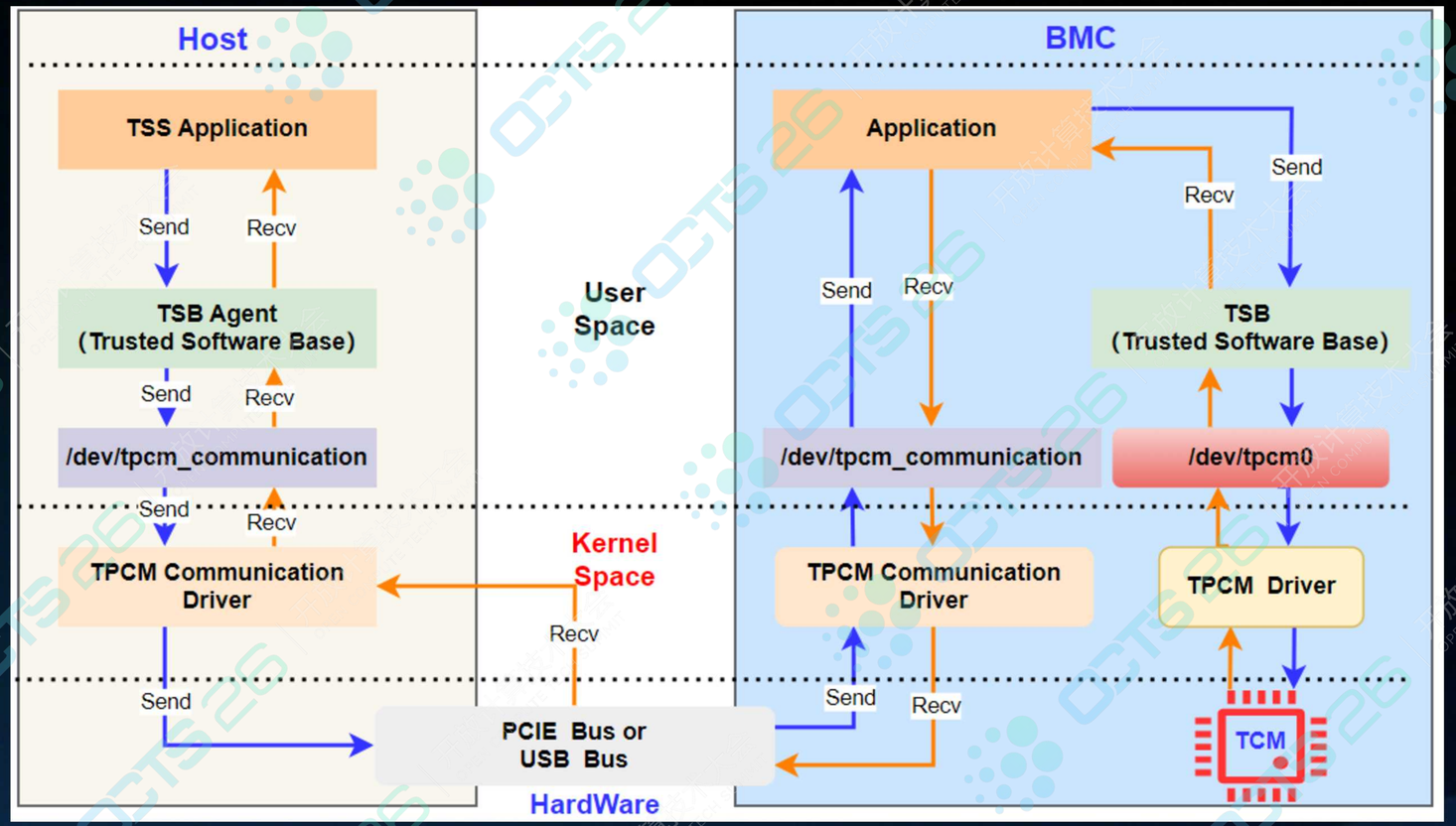
合规性高：固件级TCM服务运行在可信执行环境中，符合GMT 0012-2020 接口规范，有效地降低了硬件成本，简化板卡设计。

Host端软件设计



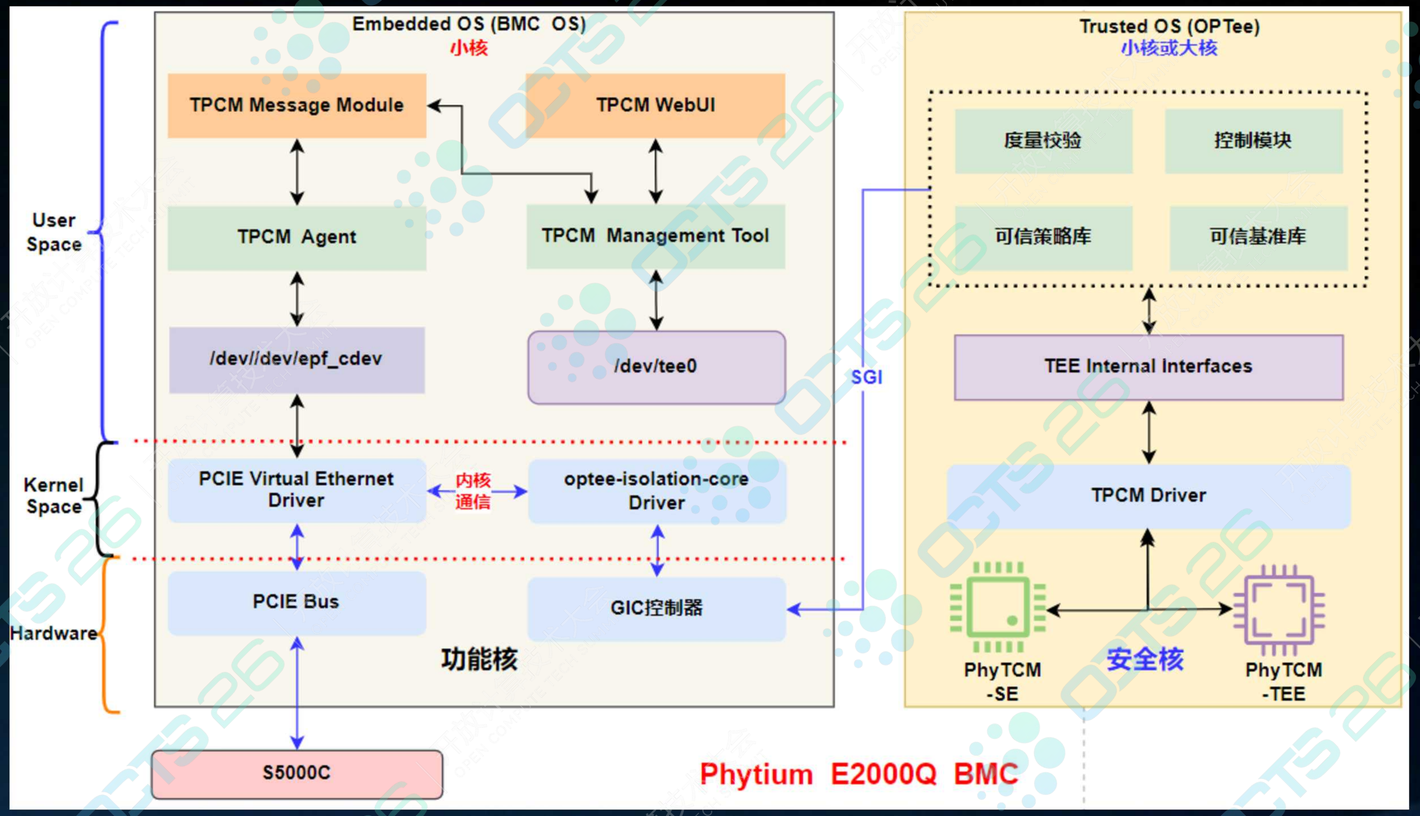
1. Host端应用层软件采用TCG开源的TPM2 软件栈TSS（TPM2 Software Stack），通过TCTI接口通信交互；
2. Host端内核层的TPM驱动，遵守Linux可信框架提供的统一接口，屏蔽硬件差异。
3. 硬件层，使用满足认证条件的TCM2.0、TPM2.0芯片，或飞腾BMC卡。

基于BMC的TCM通信



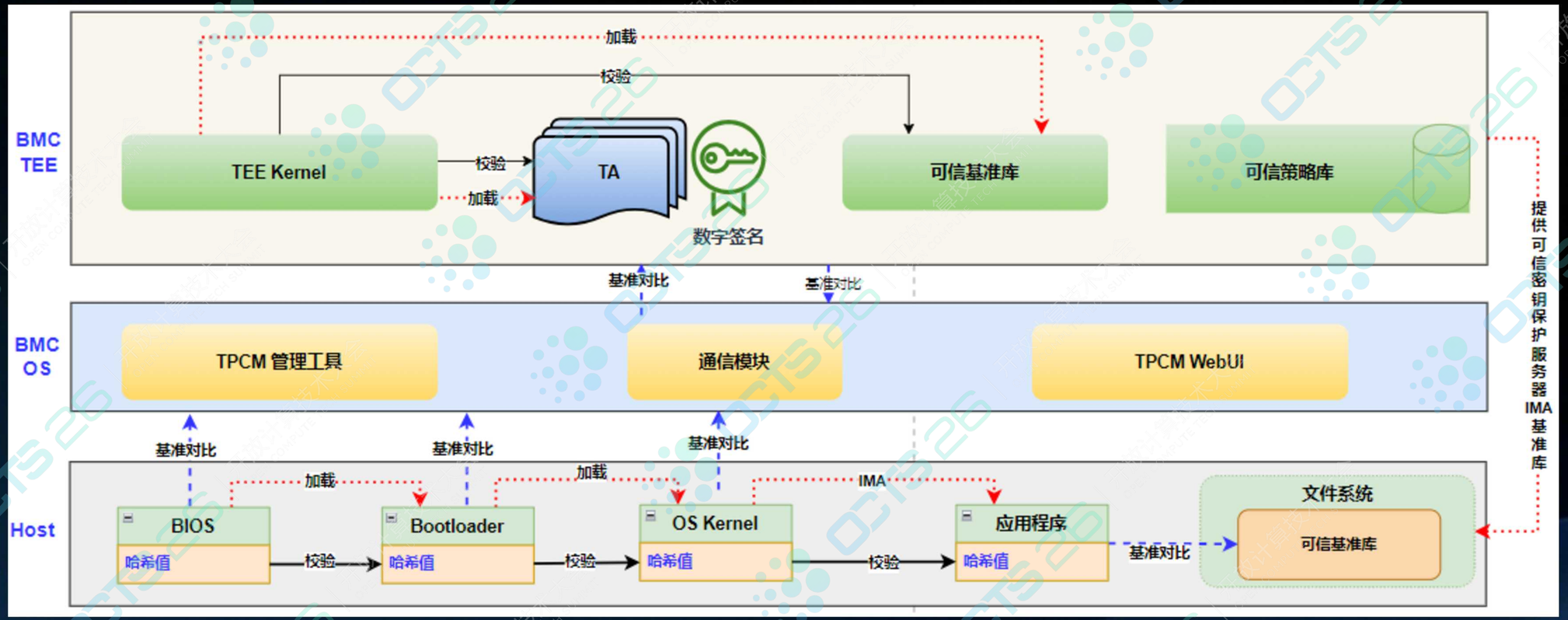
1. Host端的TSB Agent，对OS和应用程序做度量，提供度量机制；
2. BMC与Host基于PCIE总线或USB总线，实现度量过程的通信交互。

基于BMC的TPCM可信根

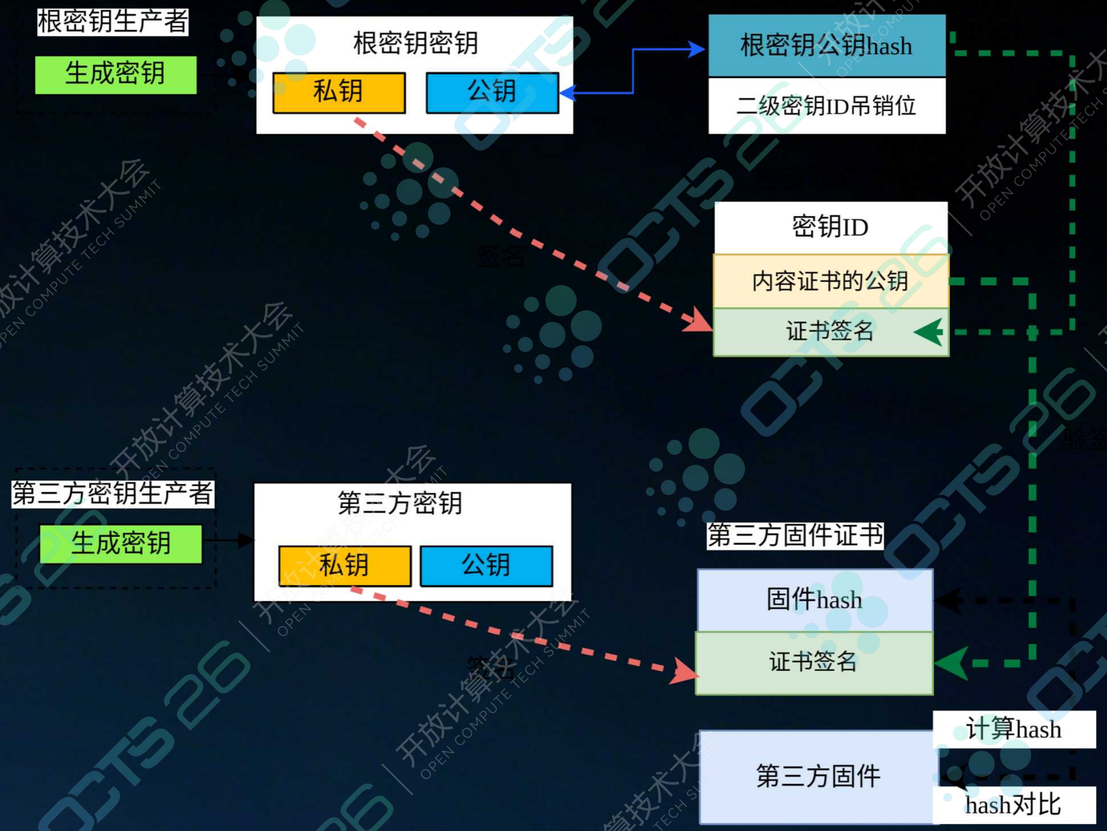


1. BMC内部分为嵌入式系统BMC OS和安全态OpTee系统，通过SGI软中断，实现BMC内部的功能核和可信核之间的通信交互。
2. BMC内部的功能核与可信核，通过共享内存，实现度量对象的数据交互。

基于BMC的服务器可信启动控制



创建根密钥

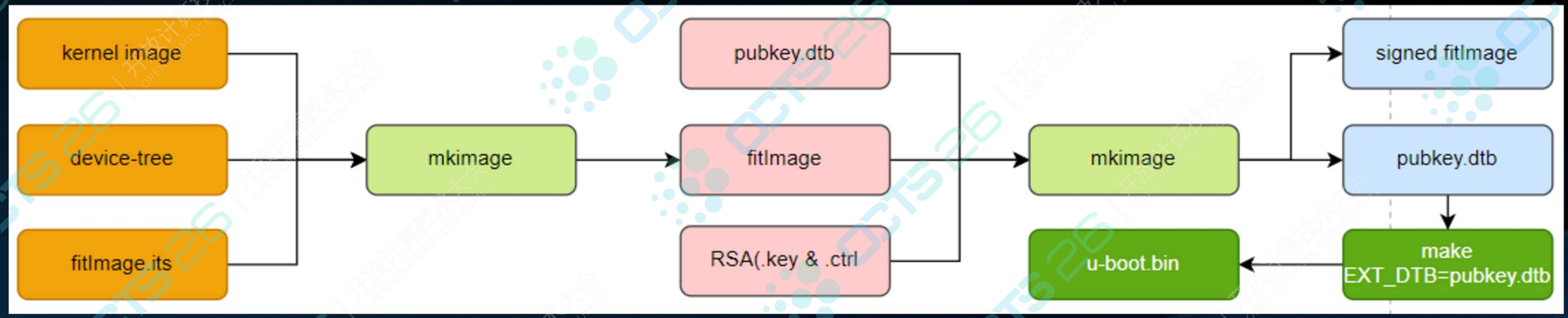


1. 签名的过程就是一个加密的过程，就是使用私钥以某种签名算法对数据摘要进行运算产生数字签名的过程。
2. 密钥证书通过上一级的私钥签名生成，UEFI的安全启动过程中，Shim、Grub、Kernel均需要在镜像编译制作过程中使用相同私钥/证书进行签名。
3. 上一级的公钥验签下一级的公钥证书，得到下一级公钥的哈希值，然后对比下一级公钥的哈希值，是否相等。

制作安全内核的fitImage

基于fitImage框架的安全启动，Kernel的公钥保存在u-boot的dtb文件里，证书在fitImage，使用mkimage制作BMC的安全引导固件u-boot和带有签名的fitImage。

make EXT_DTB=~/bmc/openbmc_11/make-fitImage/u-boot-pubkey.dtb



设备树配置

```
diff --git a/arch/arm64/boot/dts/phytium/pe2201.dtsi b/arch/arm64/boot/dts/phytium/pe2201.dtsi
index 684b1fc96..701c38b5b 100644
--- a/arch/arm64/boot/dts/phytium/pe2201.dtsi
+++ b/arch/arm64/boot/dts/phytium/pe2201.dtsi
@@ -24,6 +24,13 @@
                enable-method = "psci";
                clocks = <&scmi_dvfs 2>;
        };
+       cpu1: cpu@1 {
+                       device_type = "cpu";
+                       compatible = "phytium,ftc310", "arm,armv8";
+                       reg = <0x0 0x201>;
+                       enable-method = "psci";
+                       clocks = <&scmi_dvfs 2>;
+               };
 };

 &soc {
diff --git a/arch/arm64/boot/dts/phytium/pe220x.dtsi b/arch/arm64/boot/dts/phytium/pe220x.dtsi
index 2b281f30e..854b97ad9 100644
--- a/arch/arm64/boot/dts/phytium/pe220x.dtsi
+++ b/arch/arm64/boot/dts/phytium/pe220x.dtsi
@@ -49,6 +49,11 @@
                        #thermal-sensor-cells = <1>;
                };
        };
+
+       optee {
+                       compatible = "linaro,optee-tz";
+                       method = "smc";
+               };
 };

 thermal-zones {
```

```
configurations {
        default = "conf-phytium_phytium-bmc-e2ps23.dtb";
        conf-phytium_phytium-bmc-e2ps23.dtb {
                description = "1 Linux kernel, FDT blob, ramdisk";
                kernel = "kernel-1";
                fdt = "fdt-phytium_phytium-bmc-e2ps23.dtb";
                ramdisk = "ramdisk-1";
                hash-1 {
                        algo = "sha256";
                };
                signature-1 {
                        algo = "sha256,rsa2048";
                        key-name-hint = "kernel-dev";
                        sign-images = "kernel", "fdt", "ramdisk";
                };
        };
};
```

BMC u-boot安全启动

```
     Load Address: 0x82000000
     Hash algo:    sha256
     Hash value:   a2a081cefe5fd46ee4c4a5d2e1da215e168159dac0e786455512b61806b1d14b
   Verifying Hash Integrity ... sha256+ OK
   Loading fdt from 0x3871abf8 to 0x82000000
   Booting using the fdt blob at 0x82000000
   Uncompressing Kernel Image
   Loading Ramdisk to f3a90000, end f3e2b00c ... OK
   Loading Device Tree to 00000000f3a86000, end 00000000f3a8fe27 ... OK
auth_mod_init
Wait SW initialization done!
Software initialization done!
Using crypto library 'mbed TLS'
Auth img_id = 31 , img_base = 0x16f408 , img_size = 0x1f1
auth_signature start ...
auth_nvctr start ...
auth_image_internal end
Auth img_id = 30 , img_base = 0x16f008 , img_size = 0x68
auth_hash start ...
auth_image_internal end
Auth img_id = 29 , img_base = 0x16f808 , img_size = 0x1f1
auth_signature start ...
auth_nvctr start ...
auth_image_internal end
Auth img_id = 28 , img_base = 0x38300000 , img_size = 0x7bd0b4
auth_hash start ...
auth_image_internal end

Starting kernel ...
```

uboot公钥验签内核的公钥证书，获取证书里公钥hash值，然后计算内核公钥的hash值。

1. mg id= 3l ,img base = 0x16f408 内核公钥证书，由uboot私钥签名生成。
2. img id = 30 ,img base = 0x16f008 内核公钥存放的位置，内核厂商生成的公钥。

Host 端加载USB TCM

CONFIG_USB_PHYTIUM=y

CONFIG_CRYPTO_SM3=y

CONFIG_TCG_TPM=y

CONFIG_EVM=y

CONFIG_IMA_APPRAISE=y

CONFIG_INTEGRITY=y

CONFIG_IMA=y

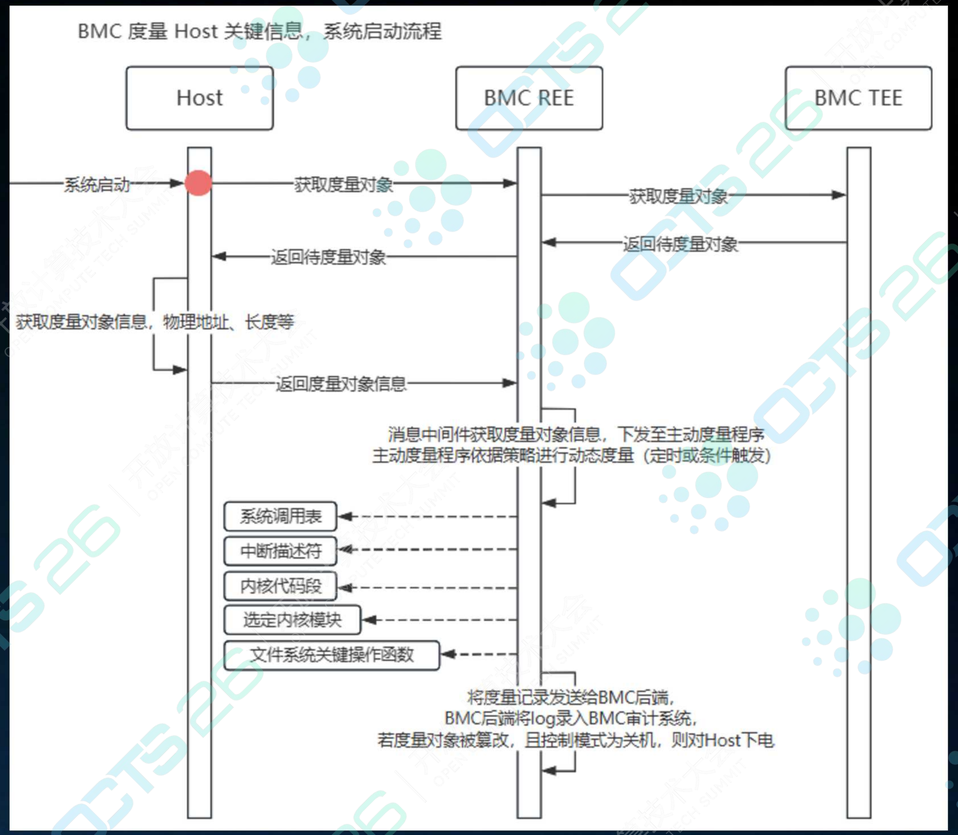
CONFIG_IMA_NG_TEMPLATE=y

CONFIG_IMA_DEFAULT_HASH_SM3=y

```
[root@localhost ~]# lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 004: ID 1d6b:0104 Linux Foundation Multifunction Composite Gadget
Bus 003 Device 003: ID 4875:0100 E2000 TCM USB TEXT Phytium usb tcm
Bus 003 Device 002: ID 4989:0100
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
[root@localhost ~]# lsusb -s 3:3 -vvv

Bus 003 Device 003: ID 4875:0100 E2000 TCM USB TEXT Phytium usb tcm
Device De
          bDescriptorType         1
  bcdUSB               2.00
  bDeviceClass          255 Vendor Specific Class
  bDeviceSubClass         0
  bDeviceProtocol         0
  bMaxPacketSize0        64
  idVendor           0x4875
  idProduct          0x0100
  bcdDevice            5.15
  iManufacturer           1 E2000 TCM USB TEXT
  iProduct                2 Phytium usb tcm
  iSerial                 0
  bNumConfigurations      1
  Configuration Descriptor:
```

可信动态度量交互流程



1. BMC获取Host端需要动态度量的对象进程（运行中的程序）；
2. Host端的TPCM Agent，将待度量进程的内存物理地址、度量内存大小等信息，发送给BMC解析；
3. BMC调用TSB 共享库，通过SMC通信进入TEE，执行可信计算，得到可信策略；
4. 返回到REE态，记录保存度量结果，录入BMC的审计日志。

IMA度量

- 配置IMA度量策略

  echo “measure func=BPRM_CHECK mask=MAY_EXEC” >/sys/kernel/security/integrity/ima/policy”

- 查看IMA度量记录

```
[root@localhost ~]# [  864.976852] TCM_DEBUG: [tcm_usb_recv:141]: Get Response: 802000130000 len is 19
cat /sys/kernel/security/ima/ascii_runtime_measurements
10 9c97ab82aecb35a2658b68ea1f46addd1fe9a85c ima-ng sm3:ce8a55facc311aa812685457a8991e7c9598624af06c397f1edb8a37c7b98b95 boot_aggregate
10 526b6bb5d03210491649c223ec921774b5986131 ima-ng sm3:916c2791f8367a440043296b280327d5707adb677366a11a647f710dec46c913 /usr/bin/kmod
10 cf81fd2710f36bc94c0d6f2460e6b8b67037e20a ima-ng sm3:151782ee1352fbec547bccd9727a6b75075ec689598fa1cd879012d1066aa619 /usr/lib/systemd/systemd-cgroups-agent
10 b920fb8e069e9786ba7bcefd9fc1d182d5553b6e ima-ng sm3:046df343d744f65690ee6e3bac55400bd3d54731e137d6f12623dc007d118a85 /usr/bin/bash
10 9138cc03d682eb84a0fdf52c2d150a98ba8ea4d1 ima-ng sm3:e996ad4c30346bf67afee469a440e67d92453d67955f6d982fe0f4798bd2f971 /usr/bin/xkbcomp
10 c62fdc74c678cc62ba57ddc2e17d4593075ff89a ima-ng sm3:8239864b63cda579d47ac65bb62fb0165bc5726b1cb7cdf3a388a4aba03f0f06 /usr/sbin/lightdm
10 10ca95af6302f5a3f1ddfdcf08550856aed073e6 ima-ng sm3:8b584cfed0ec4b8a821e284815592beee77e2bebc27807ebcd7f7937a1e30121 /usr/sbin/ukui-greeter
10 21491535838197f13c3c4b15b8e5380fc3069532 ima-ng sm3:7879f3e2f475d3045ea4c0026db0d047e667777cb873c58ccf1bef5e7893e571 /usr/bin/xhost
10 0adcb72fcbc3eecfa5abb41e6baf198a62d588e0 ima-ng sm3:1df0a048ec1cf5dcb11b91ec4a0fe59f5a5873dd018de7e87f97a190dcc2ecfc /usr/bin/pkla-check-authorization
10 6562d937cce09623c5f6c4b6c0947396817fbc63 ima-ng sm3:539af5fdd476bbb67301e4472b50510a882207433cacd007ea92dcf0378c14f5 /usr/bin/locale
10 3a217507a2815c9267829b7103ddf34ed7ea8c74 ima-ng sm3:b5048ab02bbb84ef2f2c36ec67be17e2a150be0b320e87a92925c928a6791baf /usr/bin/xrandr
10 7e9ff851783ccb1ee47571a8e353b10e94d7f157 ima-ng sm3:bd1927ad08b077ebe1f24d0e6f2a99cb9bb53dc93923852a44f8669402e75a69 /usr/lib/systemd/systemd-coredump
10 277c118c61e7d343132cca1848ee9d2a7c7e48a2 ima-ng sm3:680edf22d9920c1022b84f829c863cbd8298c22dc6dfccbd158ec991f4450a41 /usr/bin/Xorg
10 e21d080bd35347c7bf5349ecc64b41f502cb1921 ima-ng sm3:bcee7b32f91749687974fda932b3d18a1da8b1aef914a42c732b6c55da299b95 /usr/libexec/Xorg.wrap
10 b2a086dd1425e7e23a64020f98a6ced5bbe1c5a1 ima-ng sm3:8aea04fdc982e54e88e6b018404d9d42d564595711a7820425675c943ab9e691 /usr/libexec/Xorg
10 54a2bed09865180d451b17f163ecf4915e5fee62 ima-ng sm3:f83d8c6bfc55cfc2e1f624564f6f0573c9d741b4b0bf53da86f2ada67d94d0fe /usr/bin/cat
10 372e5fa48ad33b392bfb4d311b4a89b616ee0d5c ima-ng sm3:a875945436ce2b9ef6f29797c65db71fd4c916fdc2a723bcb83c0769df55bfbb /usr/bin/tpm2
10 c392bd7d3246cc4ace6725c6fa53dcc990191bd8 ima-ng sm3:64d4872cf770fa40d1cee2243e3c2e466f27e1b9268291ea1581d0d576281f8a /usr/bin/kylin_kms_daemon
10 6e283c81687c791caaa81e7c178d702d2f5d5369 ima-ng sm3:27d1efcb4f6937881f1ce0c58f5add16065dbdf00c6d0275b592ae82bb9696e2 /usr/bin/ls
```

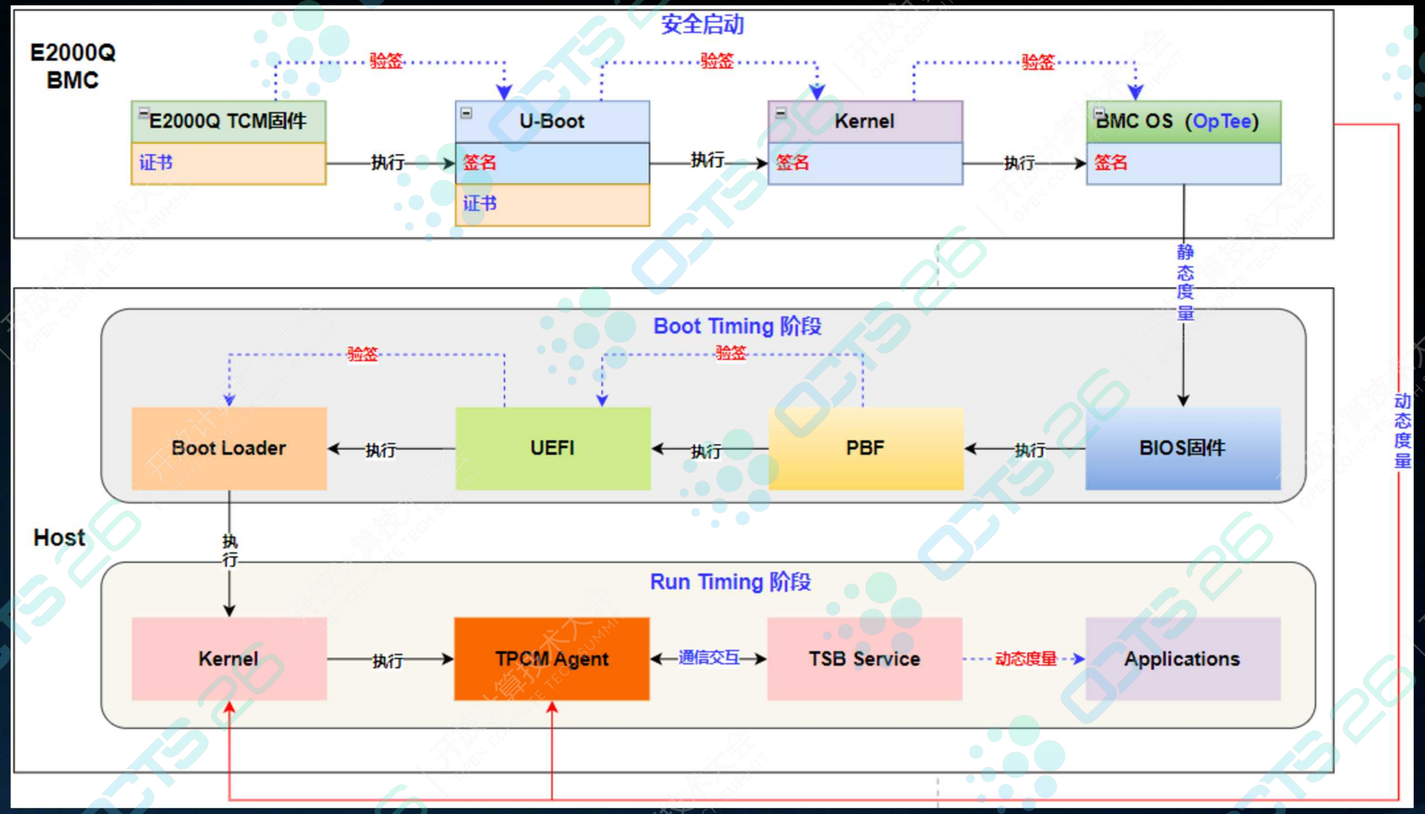
保存TCM PCR度量数据

读取已保存的TCM PCR度量的数据。

```
sm3_256:
  0 : 0x0000000000000000000000000000000000000000000000000000000000000000
  1 : 0x0000000000000000000000000000000000000000000000000000000000000000
  2 : 0x0000000000000000000000000000000000000000000000000000000000000000
  3 : 0x0000000000000000000000000000000000000000000000000000000000000000
  4 : 0x0000000000000000000000000000000000000000000000000000000000000000
  5 : 0x0000000000000000000000000000000000000000000000000000000000000000
  6 : 0x0000000000000000000000000000000000000000000000000000000000000000
  7 : 0x0000000000000000000000000000000000000000000000000000000000000000
  8 : 0x0000000000000000000000000000000000000000000000000000000000000000
  9 : 0x0000000000000000000000000000000000000000000000000000000000000000
 10: 0xE5D153160609A538EC3BDFF755FC547D07E8C1C20BBDC86A44792172C724AE2C
 11: 0x0000000000000000000000000000000000000000000000000000000000000000
 12: 0x0000000000000000000000000000000000000000000000000000000000000000
 13: 0x0000000000000000000000000000000000000000000000000000000000000000
```

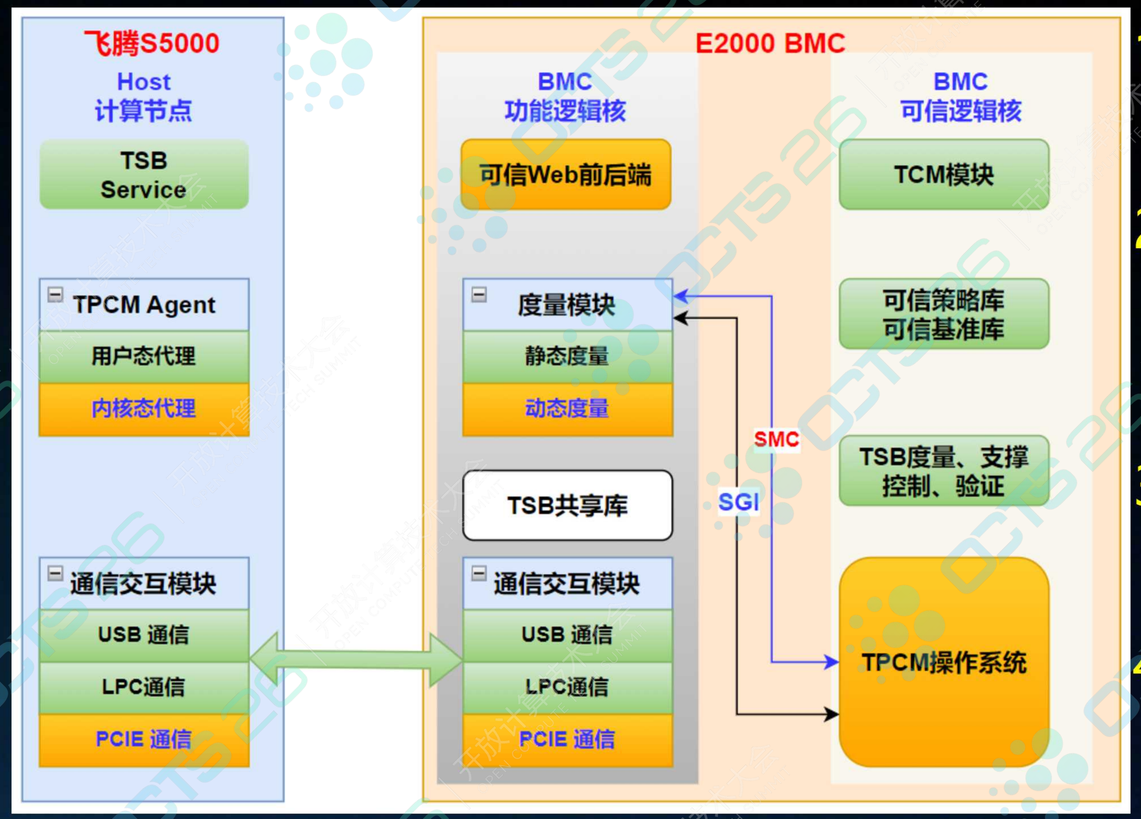
TPCM的Web管理界面

可信度量流程



1. BMC端实现安全启动，静态度量Host端启动阶段使用的Bios固件；
2. BMC功能核动态度量、主动度量Host端运行阶段的内核和应用程序；

飞腾S5000基于BMC的度量交互



1. Host端基于飞腾S5000C，实现TSB服务、用户态和内核态的TPCM Agent；
2. Host端与BMC端的通信交互模块，已基于USB Bus和LPC Bus实现静态度量的通信，动态度量使用**PCIE DMA**方式；
3. BMC端BMC对Host主动度量的策略、基准设置接口，并新增度量部件类型。
4. BMC端通过**SMC**实现REE到TEE的切换。REE与TEE采用**SGI**（软中断）模式通信。

THANKS

THANKS

THANKS

THANKS