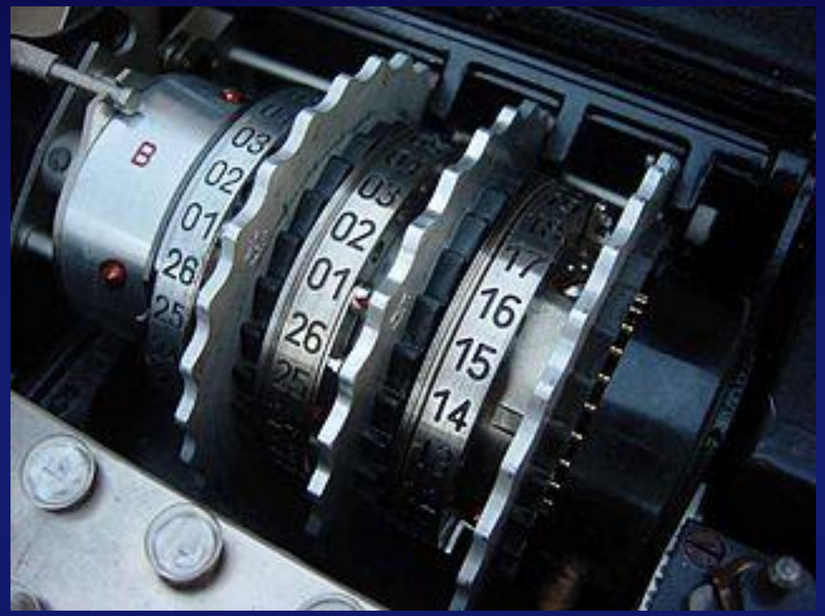
计算机安全架构演进与HarmonyOS安全设计实践

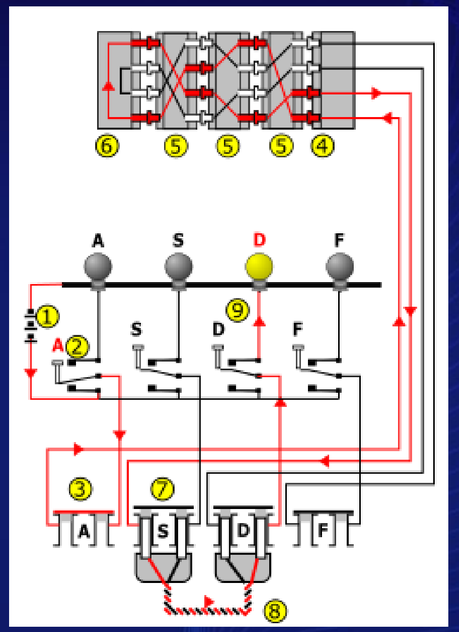
付天福

华为消费者业务 首席安全架构师

HUAWEI

第二次世界大战中最成功的密码机：恩尼格码机



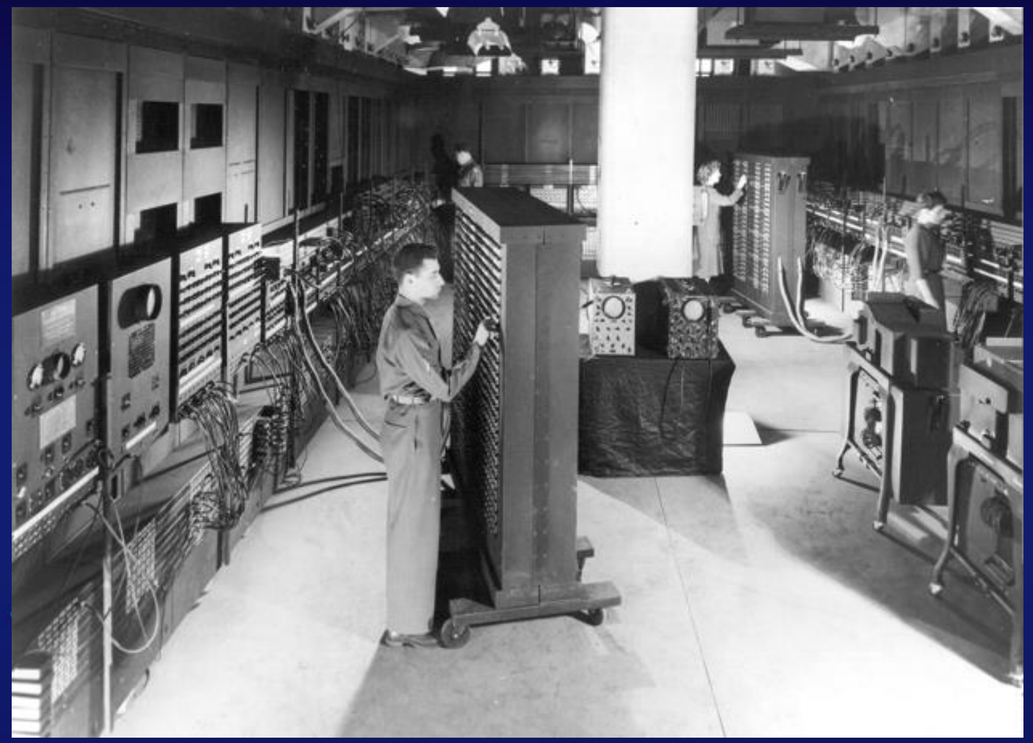


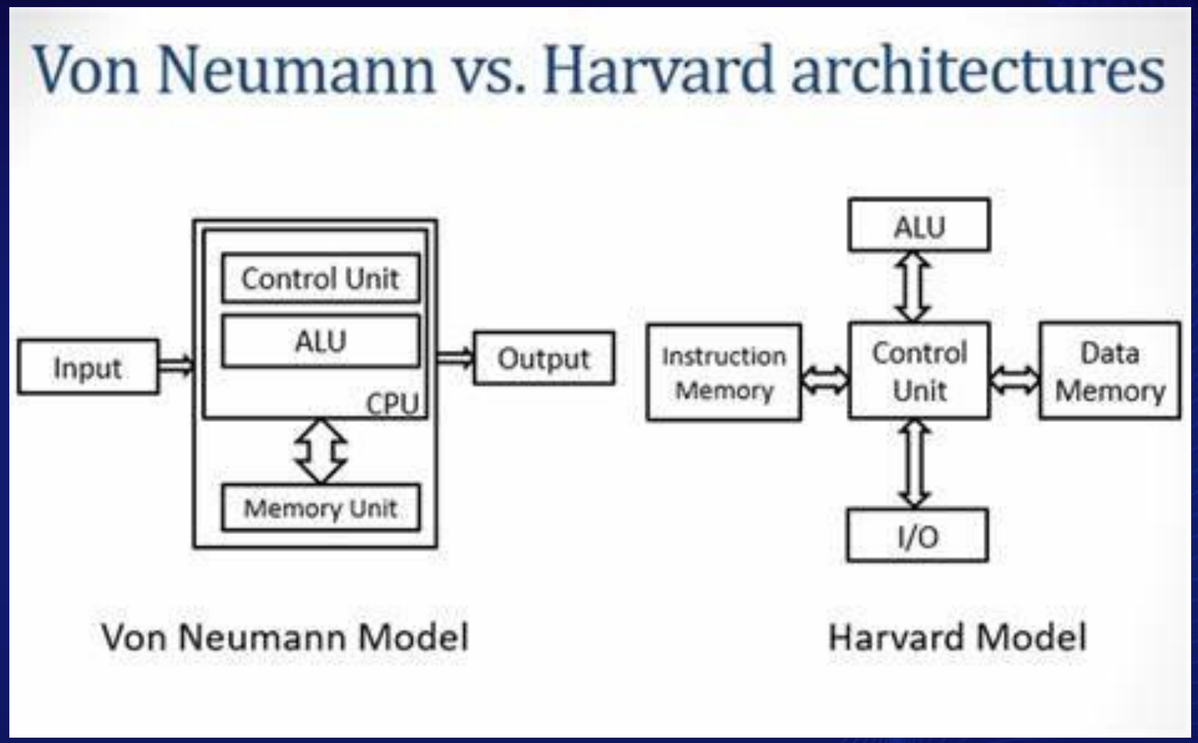
•思考：

- 如果是你来设计一台密码机，最核心的要素是什么？
- 如果是你来指挥大规模集群作战，你怎么用好密码机？

HUAWEI

第一台现代意义上的计算机ENIAC和最早的冯诺依曼架构





- **思考：**
  - 冯诺依曼架构从安全的视角，存在什么缺陷？
  - 如果你穿越到70年前去，你怎么改进？
  - 相比冯诺依曼架构，哈佛架构改进了什么？给我们的启示是什么？

HUAWEI

- **计算机安全发展演进历史**
- 典型系统安全架构模型

HUAWEI

蛮荒时代的计算机安全

•安全威胁：

➢ 输入的纸带泄密、打孔被“修改”、操作员做间谍……

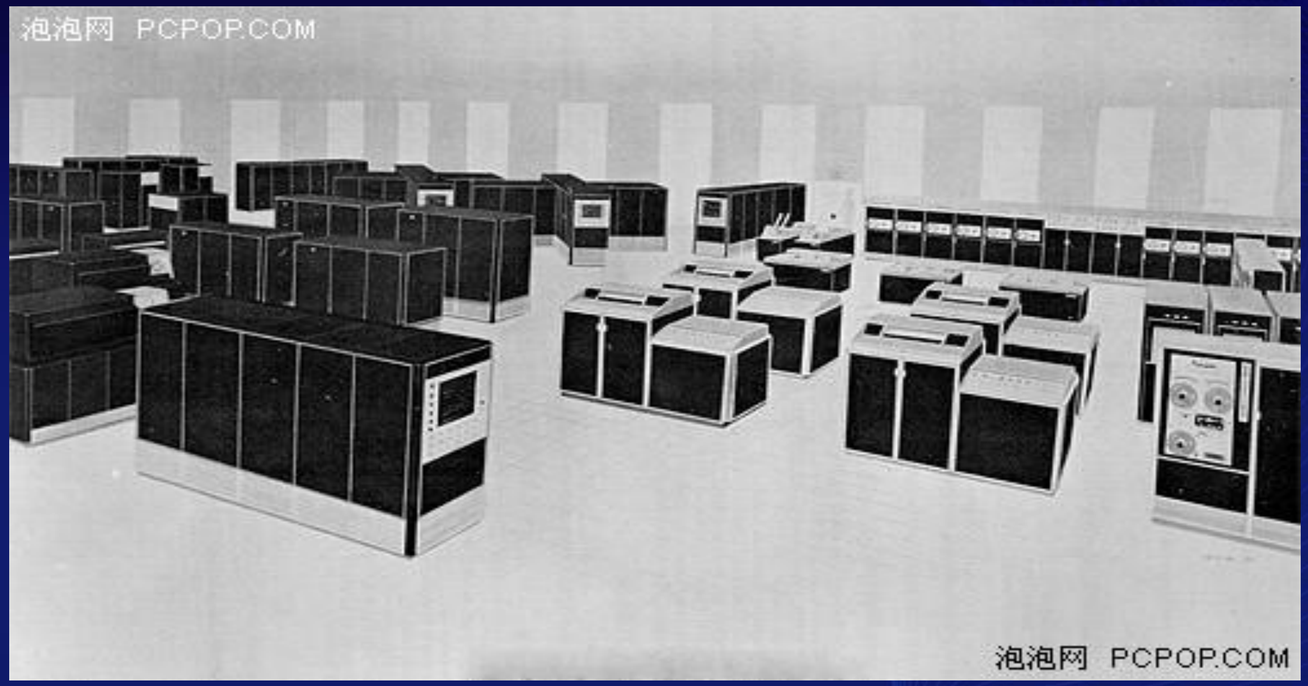
•应对办法：

➢ 物理安全：计算机被放在壁垒森严的军事保护区

➢ 人员安全：操作员要“政治过硬，踏实可靠”

HUAWEI

现代计算机操作系统安全的鼻祖：MULTICS



•设计目标：

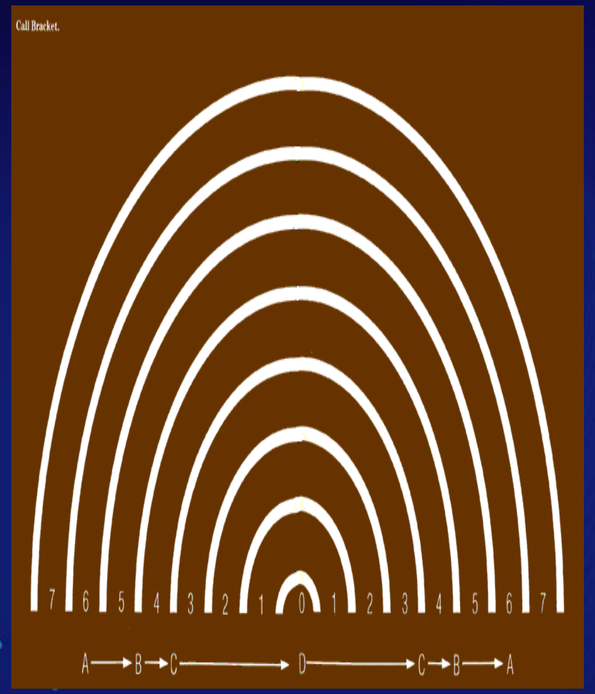
•能够支持1000个终端接入（国防部的各种研究所、各个大学、盟国的科学家等），要支持300个用户并发处理。

•美国国防部要求，MULTICS必须保证超级计算机上的**应用程序和数据的绝对安全**，因为这些程序和数据都是绝密的科研项目，一旦数据泄露，将会对国家安全带来难以估量的损失。

•MULTICS系统的一个安全假设是，**假定系统已经被攻陷，存在特洛伊木马（Trojan Horse）的情况下，仍然能够保证安全**。

HUAWEI

MULTICS安全架构模型



- Kernel resides in ring 0
- Process runs in a ring r
  - Access based on current ring
- Process accesses data (segment)
  - Each data segment has an *access bracket*: (a1, a2)
    - a1 <= a2
  - Describes read and write access to segment
    - r is the current ring
    - r <= a1: access permitted
    - a1 < r <= a2: r and x permitted; w denied
    - a2 < r: all access denied

- **Mediation**: Does interface mediate correctly?
- **Mediation**: On all resources?
- **Mediation**: Verifably?
- **Tamperproof**: Is reference monitor protected?
- **Tamperproof**: Is system TCB protected?
- **Verifiable**: Is TCB code base correct?
- **Verifiable**: Does the protection system enforce the system's security goals?

•MULTICS划时代的引入了MLS（Multi-Layer Security）的安全模型，基于这一模型，来实施系统安全的体系架构。

•MULTICS将系统划分成了64层（可以看到其目标远大，然而，也正因为这种超级复杂的分层模型，导致系统复杂度失控，最终项目失败）。

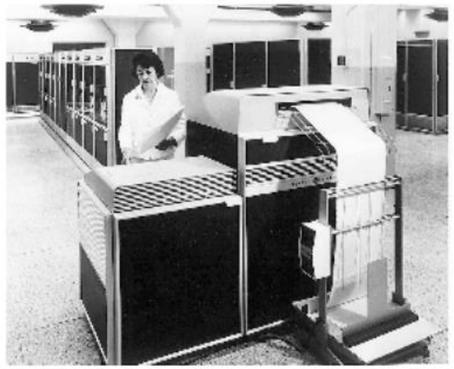
•不同层间的访问控制逻辑：r<=a1允许；a1<r<=a2不可写但是可读可执行；a2<r拒绝。

•访问控制架构模型成为后来Bell Lapadula和Biba模型的原型基础

HUAWEI

MULTICS安全架构设计目标和模型

- Secrecy
  - Multilevel security
- Integrity
  - Rings of protection
- Reference Monitoring
  - Mediate segment access, ring crossing
- Resulting system is considered a high point in secure system design



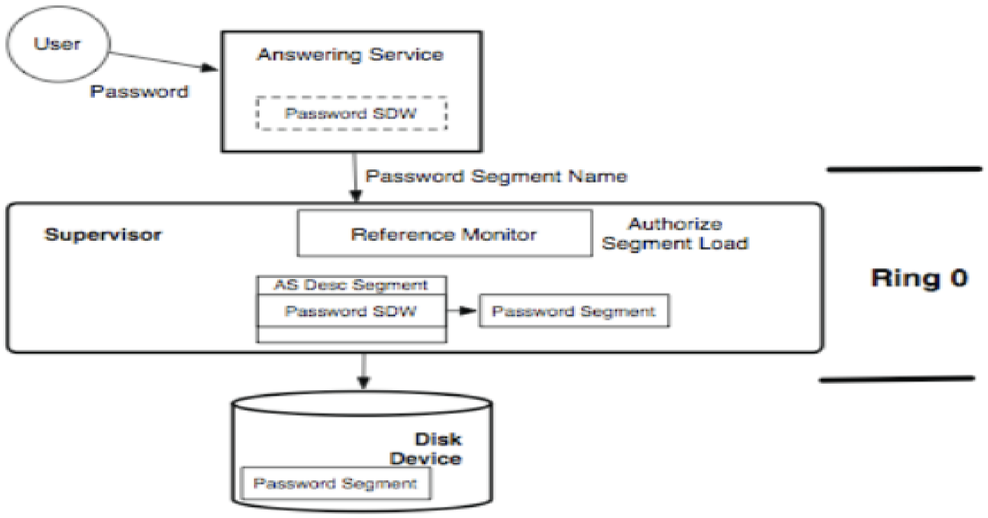


Figure 3.2: The Multics login process. The user's password is submitted to the Multics *answering service* which must check the password against the entries in the *password segment*. The Multics *supervisor* in the privileged *protection ring 0* authorizes access to this segment and adds a SDW for it to the answering service's descriptor segment. The answering service cannot modify its own descriptor segment.

- 首次将机密性和完整性保护提到了安全架构的核心
- 提出Reference Monitoring架构，成为后世安全架构设计模型的核心模型

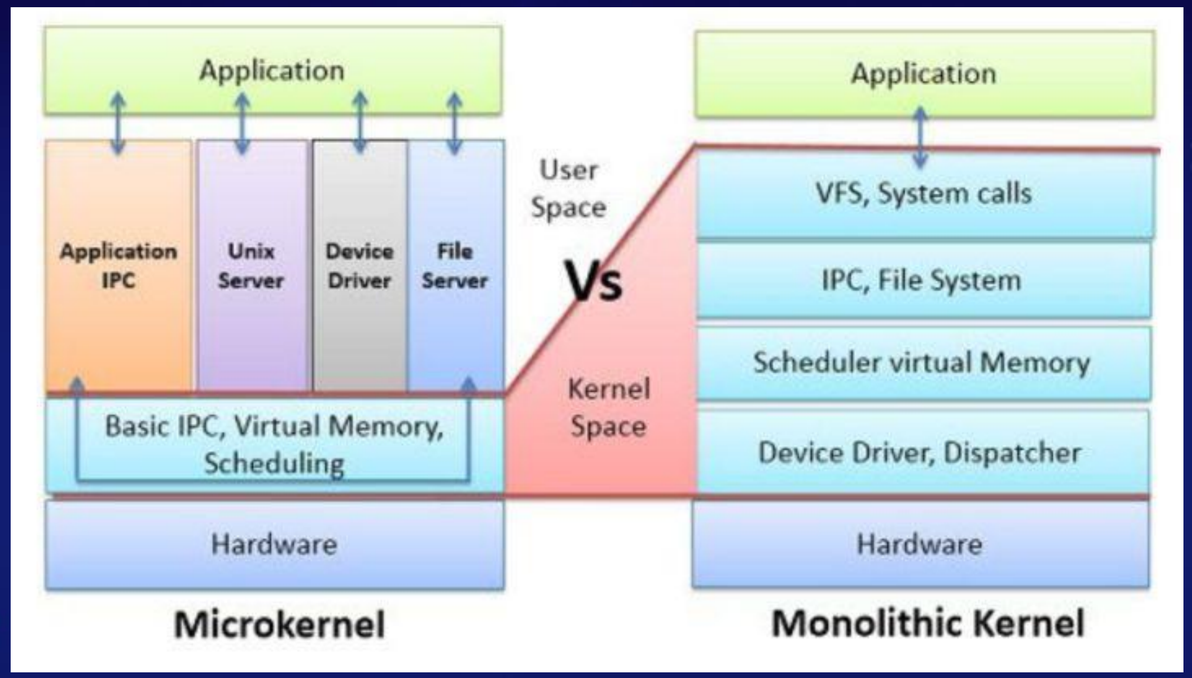
HUAWEI

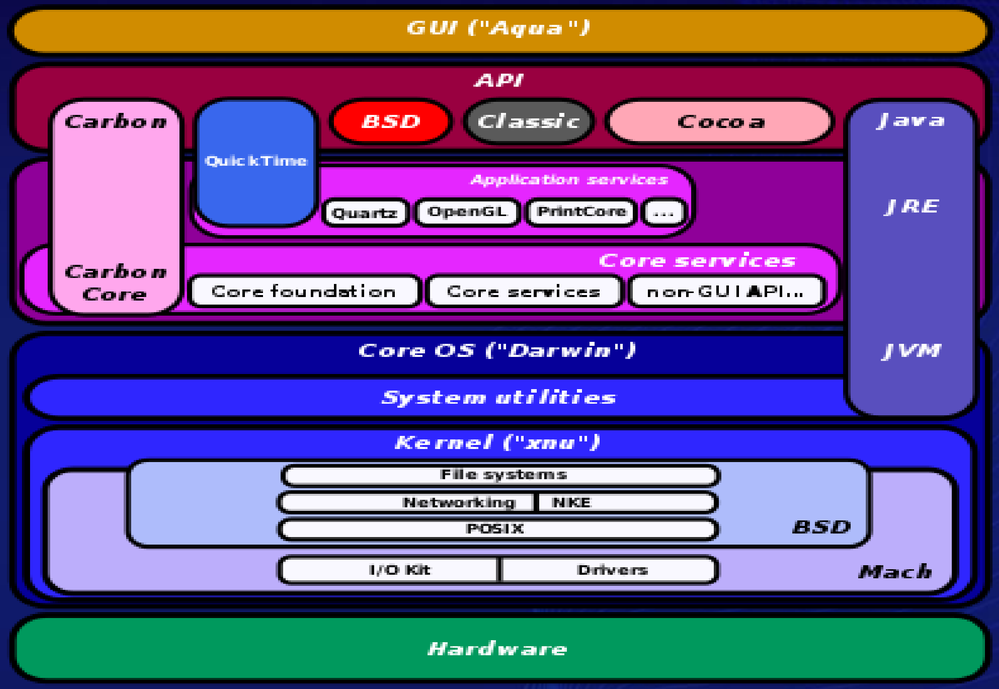
MULTICS系统的结局

由于MULTICS的设计目标，脱离了那个时代的技术水平，导致项目一再延期；然而，MULTICS划时代的安全设计思想，诞生了如下伟大的成果：

- MULTICS项目失败了，1969年，在贝尔实验室工作的Ken Thompson和Dennis Ritchie为了让自己编写的游戏能够运行起来，俩人在一台DEC小型计算机上编写了MULTICS的改编版，这就是UNIX。UNIX开启了现代操作系统的新时代。
- Roger Schell借鉴了MULTCS系统的MLS模型，推出了支持强制访问控制的MLS操作系统GEMSOS，这个操作系统的安全内核，成为后续很多安全操作系统的基石。GSMSOS首次提出了分层TCB的概念并实现了，成为第一个满足美国国防部A1级的操作系统，处理的信息级别是国家绝密。
- Paul Karger利用Reference Monitoring的架构，设计了基于Capability权能模型的安全架构，引入了OS隐蔽信道评估技术来评价OS安全等级的理论，使用虚拟机监控器的思想，成为现代虚拟化技术HyperVisor的基础。
- 1972年，James P. Anderson在《Computer Security Technology Planning Study》这篇研究中，归纳总结了MULTICS的Reference Monitor的架构模型，也提炼了MULTICS的安全可信基TCB的概念，基于这一概念，在1987年计算机桔皮书里正式提出了可信计算的思想，在2003年成立了TCGA可信计算联盟。
- 1973年，D. Bell 和L. Lapadula将MULTICS的Secrecy机密性保护的设计思想，提炼成了系统机密性保护的理论模型，后续我们称为BLP模型，其基本原则就是，高安全等级的主体不可向低安全等级写数据，防止泄密；低安全等级的主体不可向高安全等级读数据，防止窃密。
- 1975年，Kenneth J. Biba将MULTICS的Integrity完整性保护的设计思想，提炼成Biba安全模型，其基本原则就是，高安全等级的系统不可从低安全等级的系统读取程序或者数据来修改自己的系统，低安全等级的系统不可直接修改高安全等级的系统。Biba模型成为现代可信计算的核心设计原则。
- 1982年，Intel的80286处理器，正式引入了Protection Mode的模型，并在指令集上引入了Ring0~3的设计，从而在芯片和硬件层面，使能了MULTICS的MLS模型，极大的简化了使用软件来实施MLS的复杂度。
- 1985年，美国国防部发布了划时代的计算机桔皮书，引入了TCB可信计算基（Trusted Computing Base）、DAC自主访问控制（Discretionary Access Control）、MAC强制访问控制（Mandatory Access Control），成为后续安全设计的集大成。

现代操作系统内核架构

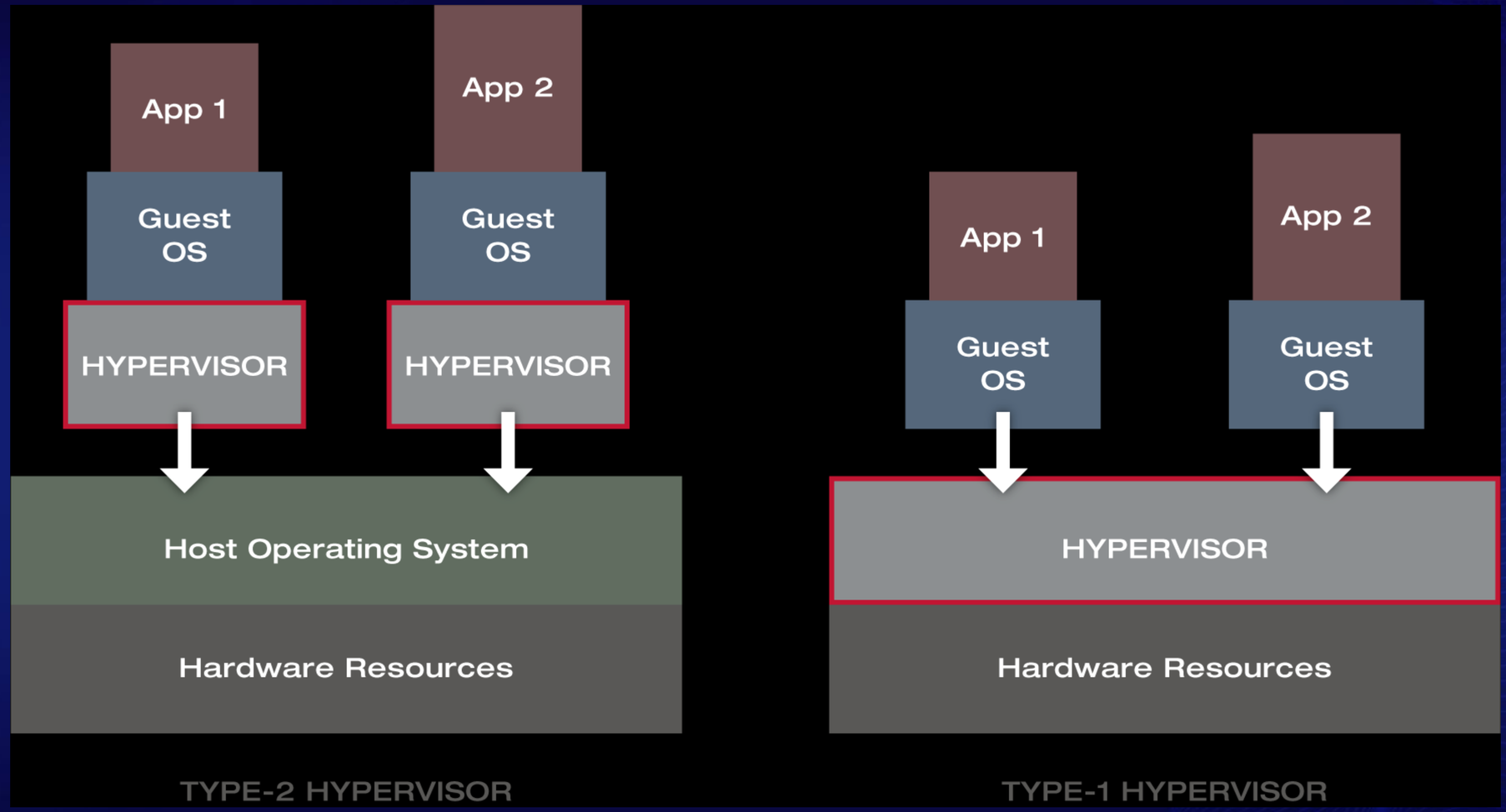




- 微内核 vs 宏内核 vs 混合内核

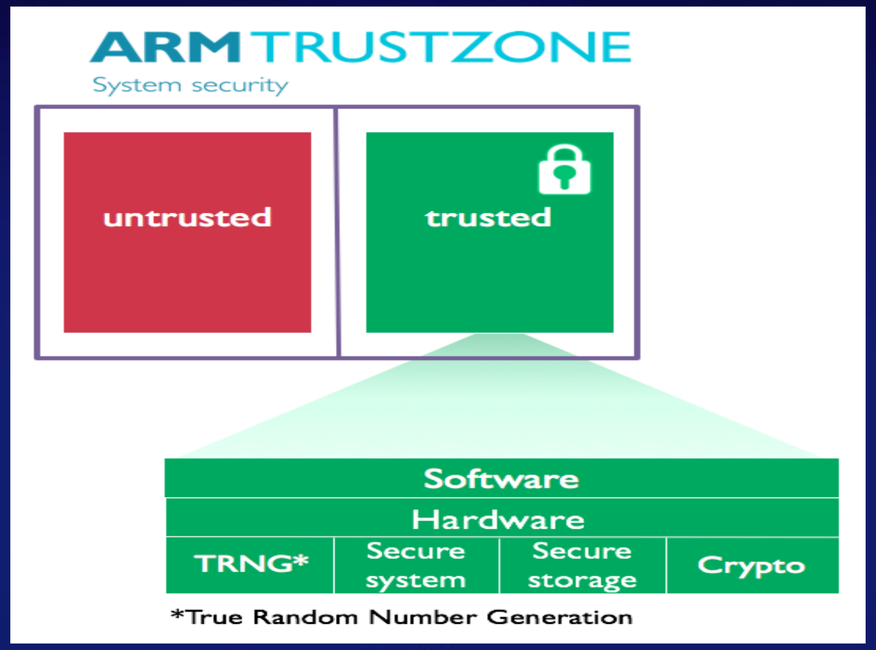
HUAWEI

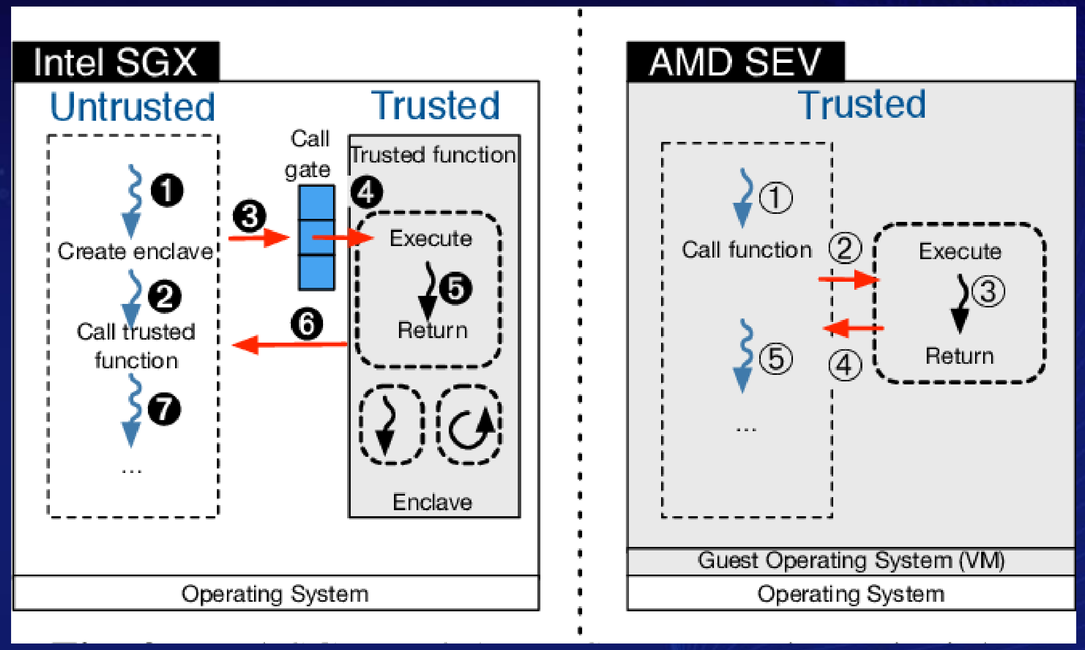
虚拟机



- 完美诠释了Reference Monitor模型

芯片使能的隔离技术：TEE、SGX、SEV

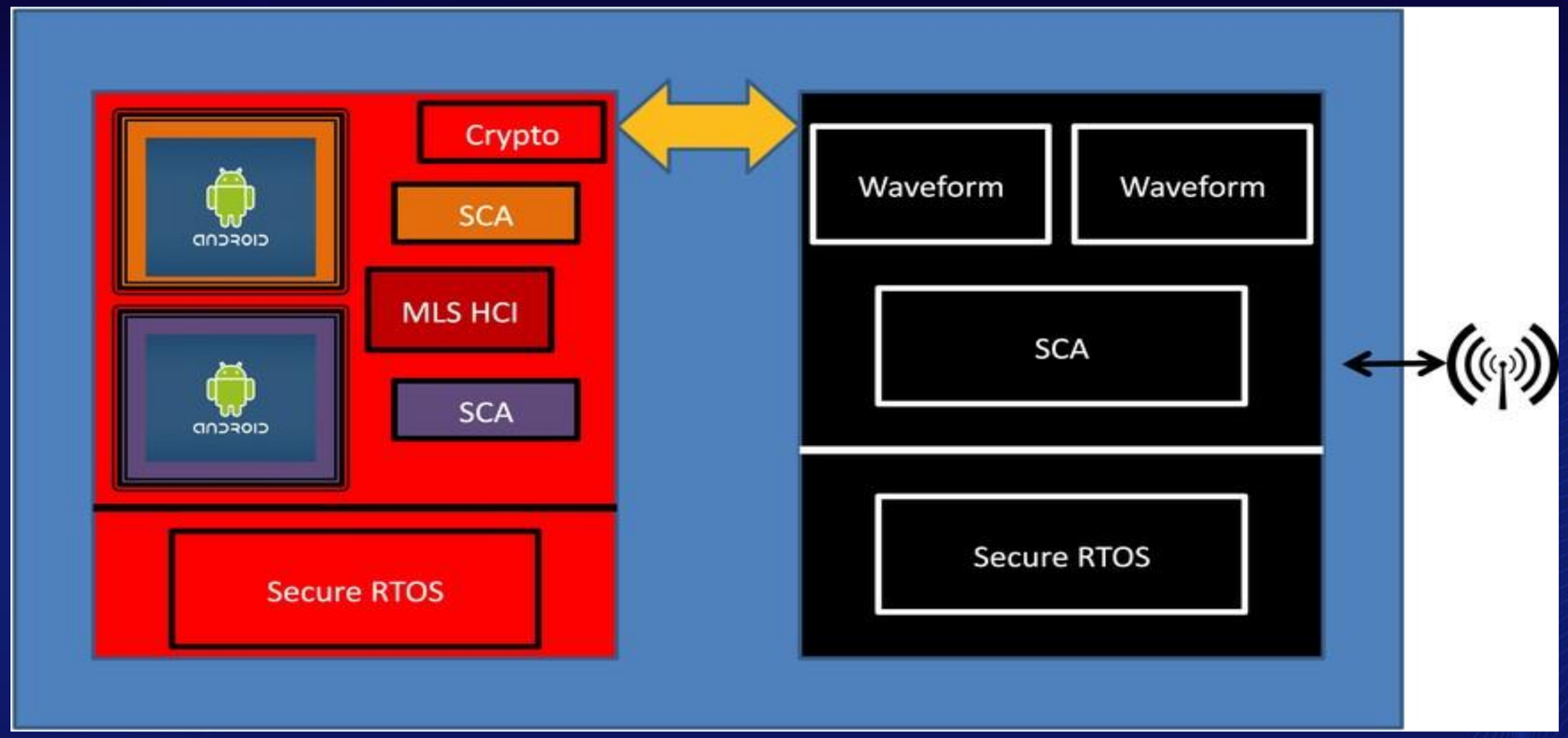




- 让处理器的控制寄存器成为OS的Reference Monitor

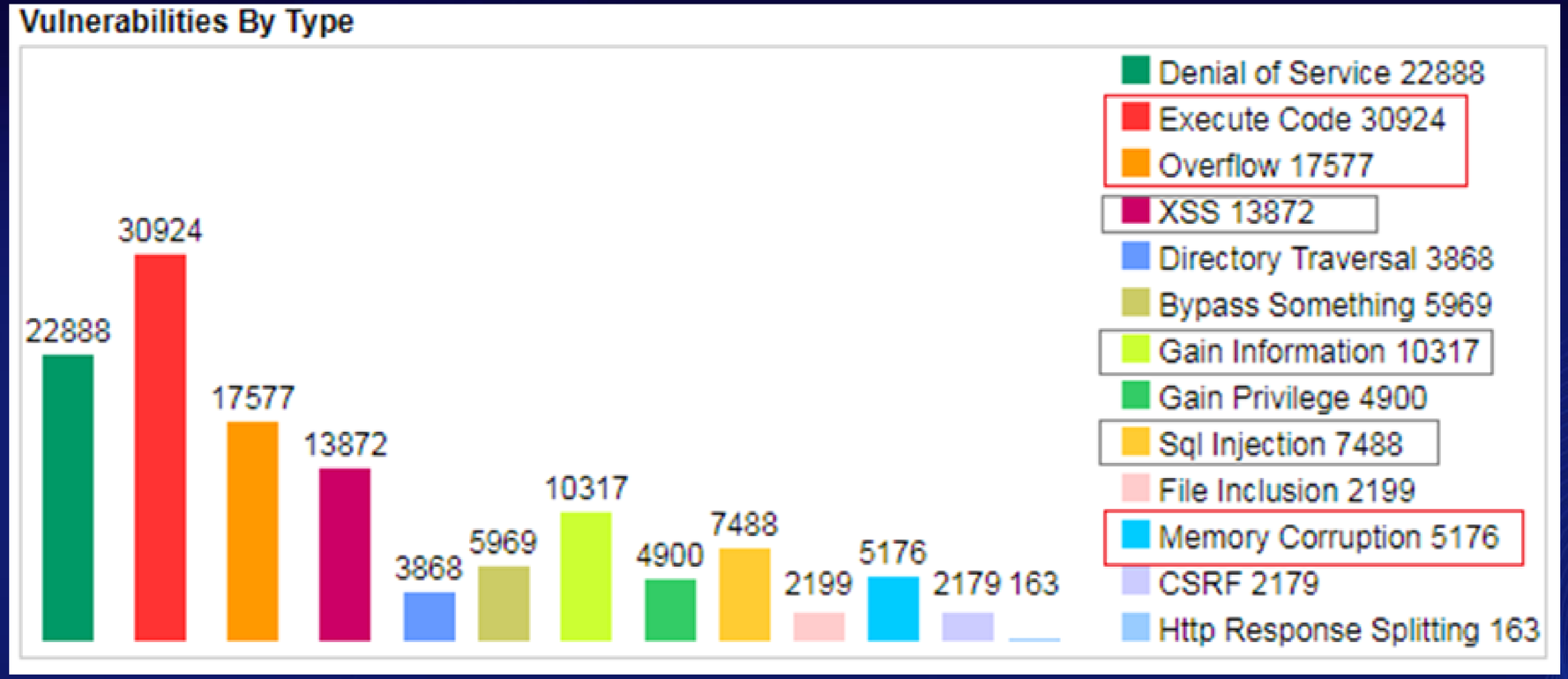
HUAWEI

最强隔离：密码学加持的“红黑”隔离



- 将计算机体系架构的安全隔离、访问控制，转换成了密钥和加解密的隔离与访问控制

漏洞：成为安全架构头顶的乌云

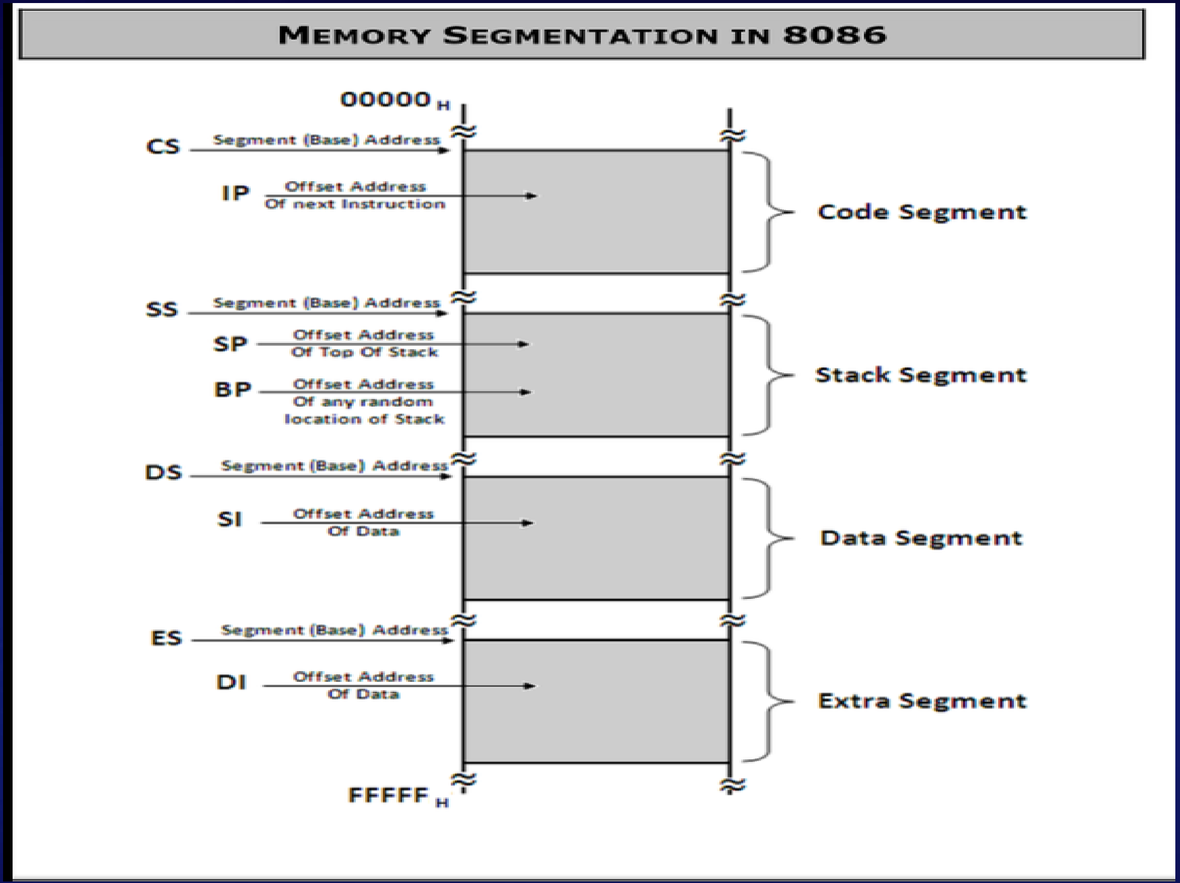


来源：CVEdetails网，1999年~2019年的CVE漏洞统计数据

- **思考：**
- **产生软件漏洞的根源是什么？**
- **可能的防御办法有哪些？**

HUAWEI

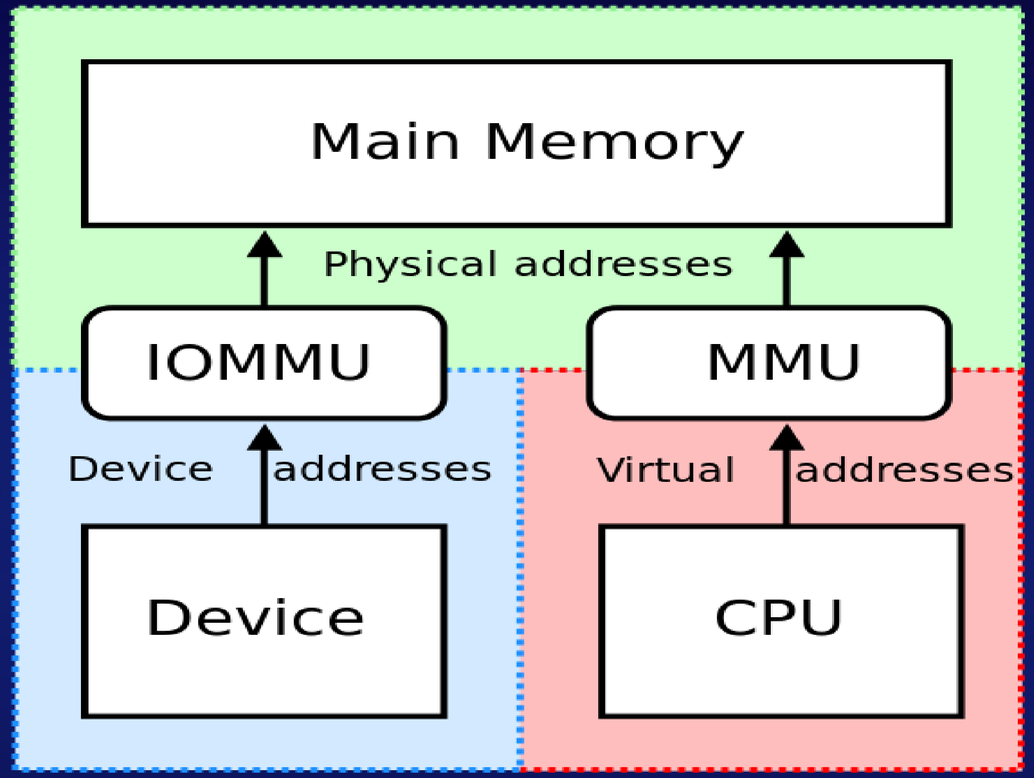
被内存分段：井水不犯河水



- 代码归代码，数据归数据，堆栈归堆栈
- 这样就足够了吗？

HUAWEI

内存隔离



- “眼不见心不烦”

内存页加密

Figure 4-8. Protection Key Register Format

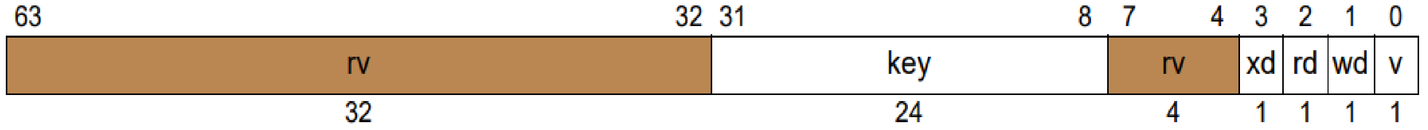


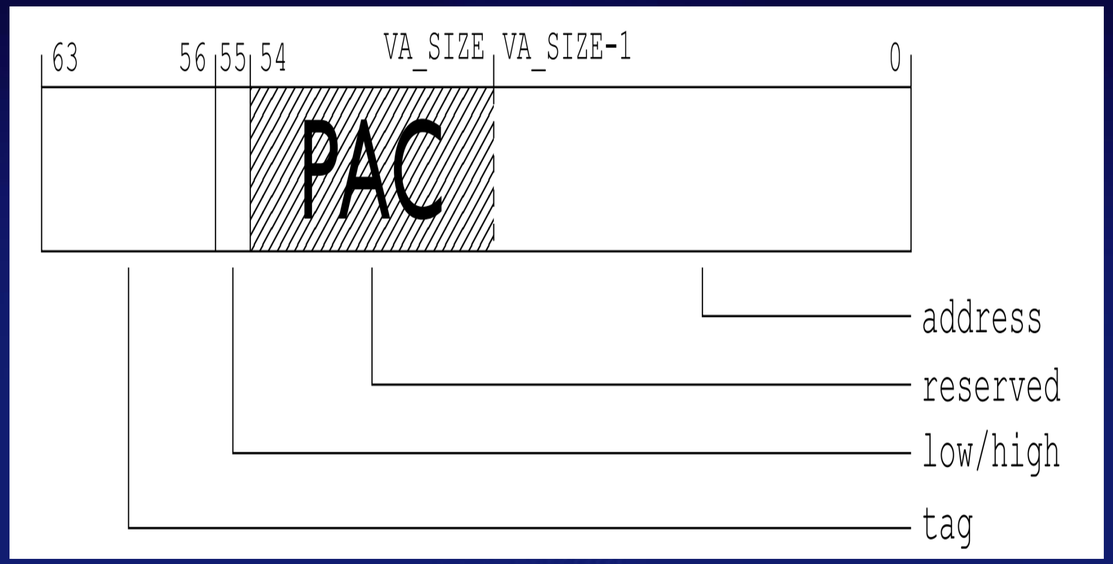
Table 4-6. Protection Register Fields

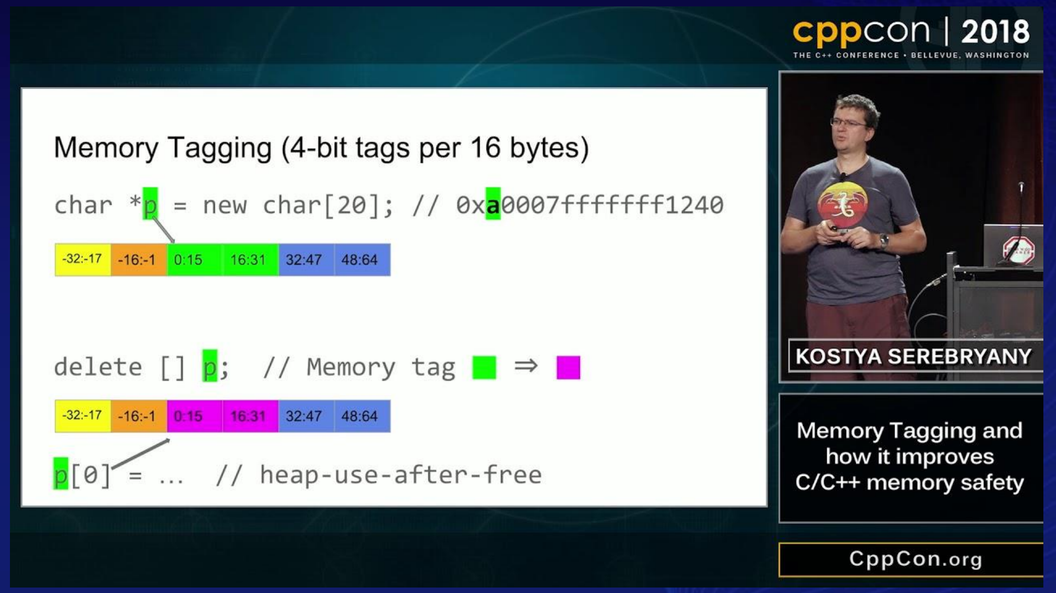
Field	Bits	Description
v	0	Valid – When 1, the Protection Register entry is valid and is checked by the processor when performing protection checks. When 0, the entry is ignored.
wd	1	Write Disable – When 1, write permission is denied to translations in the protection domain.
rd	2	Read Disable – When 1, read permission is denied to translations in the protection domain.
xd	3	Execute Disable – When 1, execute permission is denied to translations in the protection domain.
key	31:8	Protection Key – uniquely tags translation to a given protection domain.
rv	7:4,63:32	reserved

- MPK Memory Page Key内存页加密
- 密钥访问控制，只要密钥不泄露，内存就不可滥用

HUAWEI

指针加密





- 把指针的某些位进行加密或者染色，不泄密就不会滥用

HUAWEI

基于权能的内存访问控制

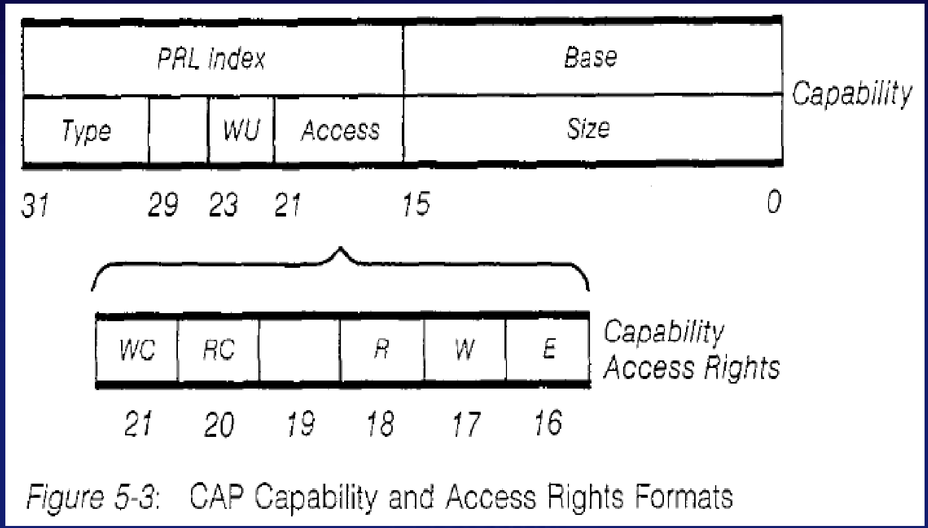
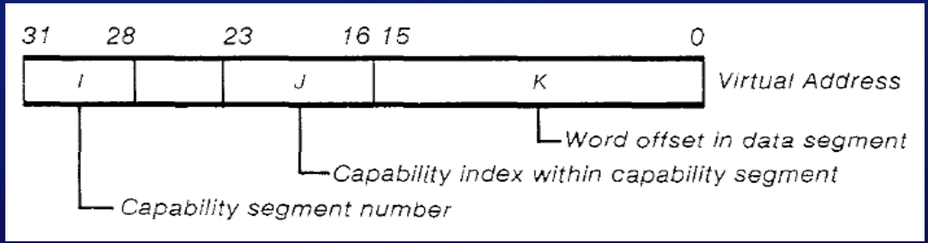
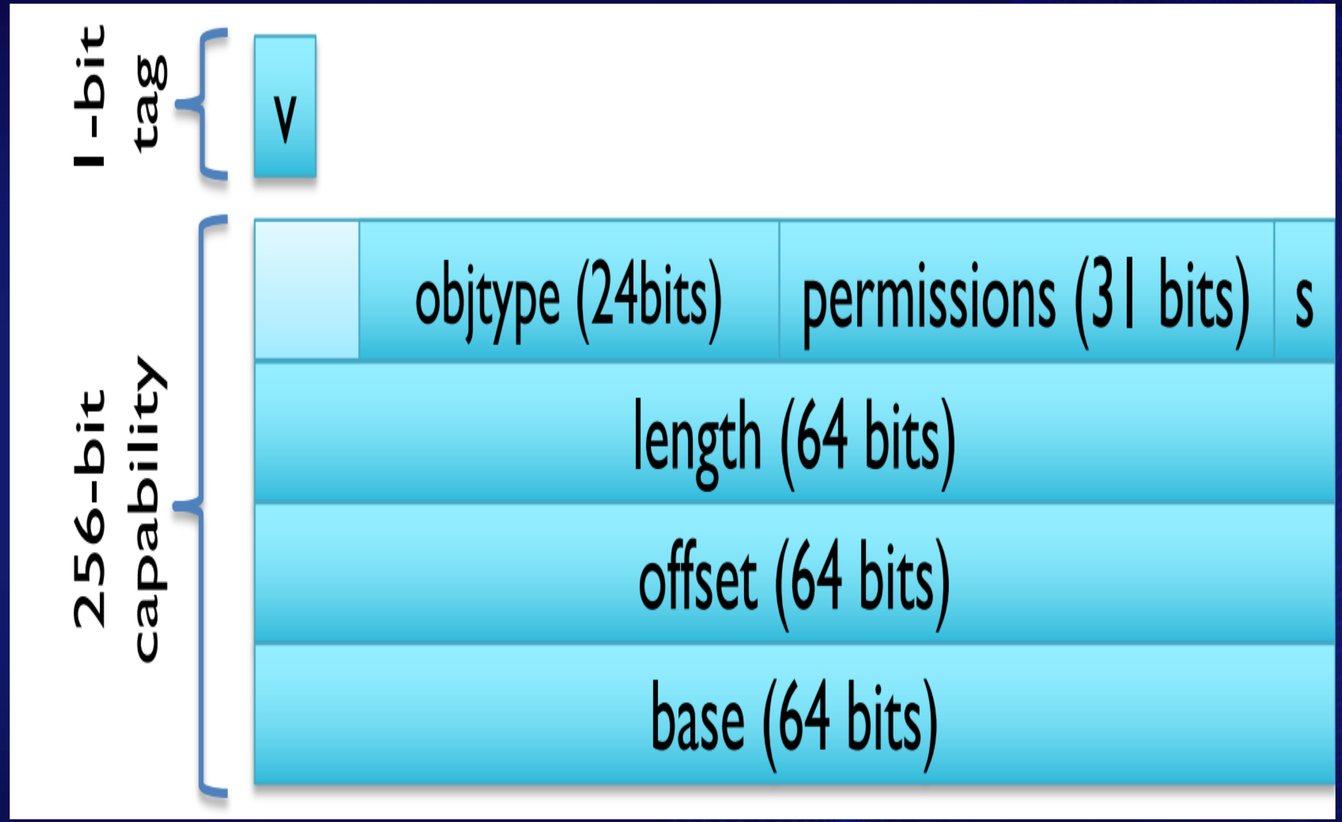


Figure 5-3: CAP Capability and Access Rights Formats

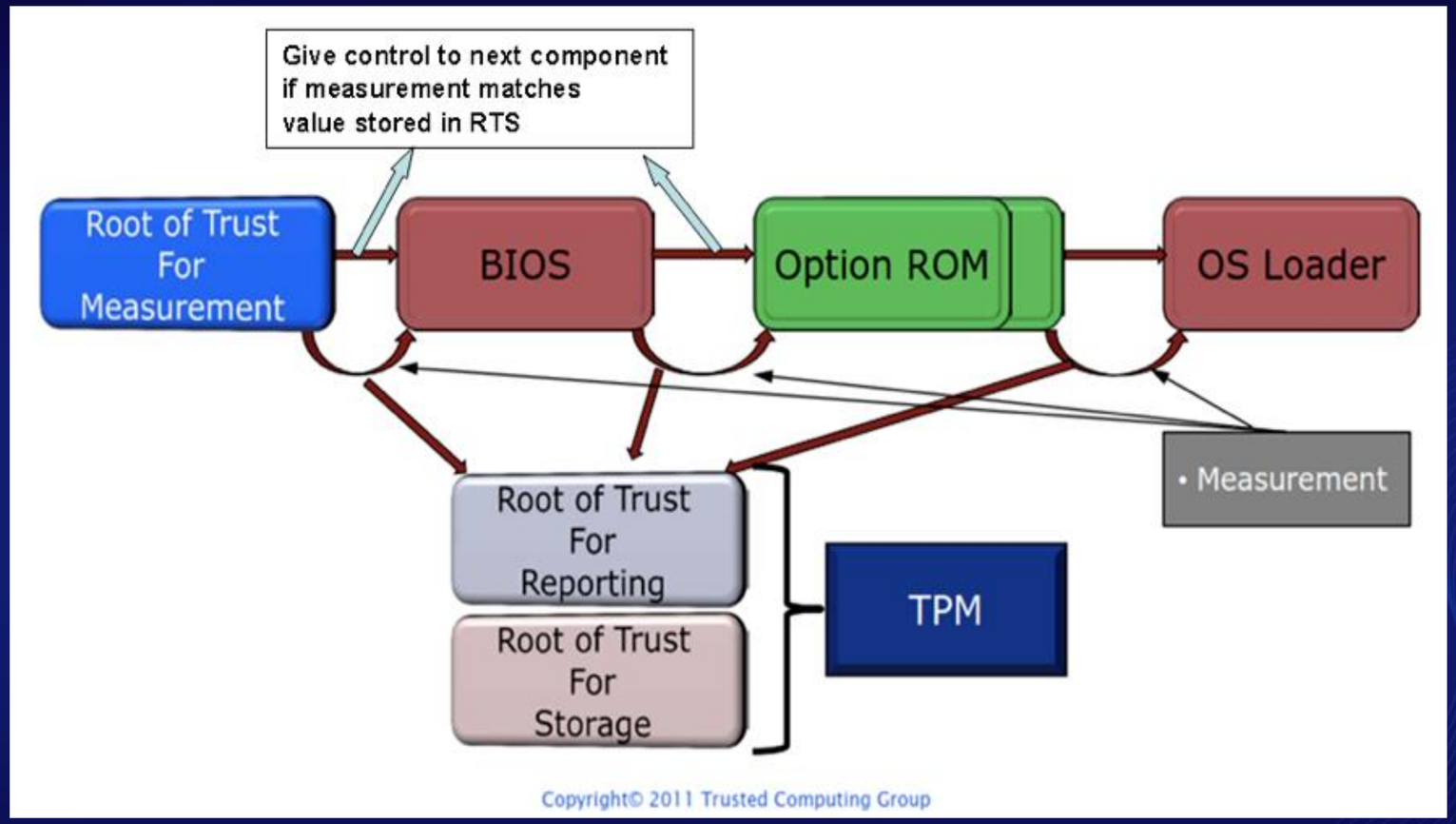




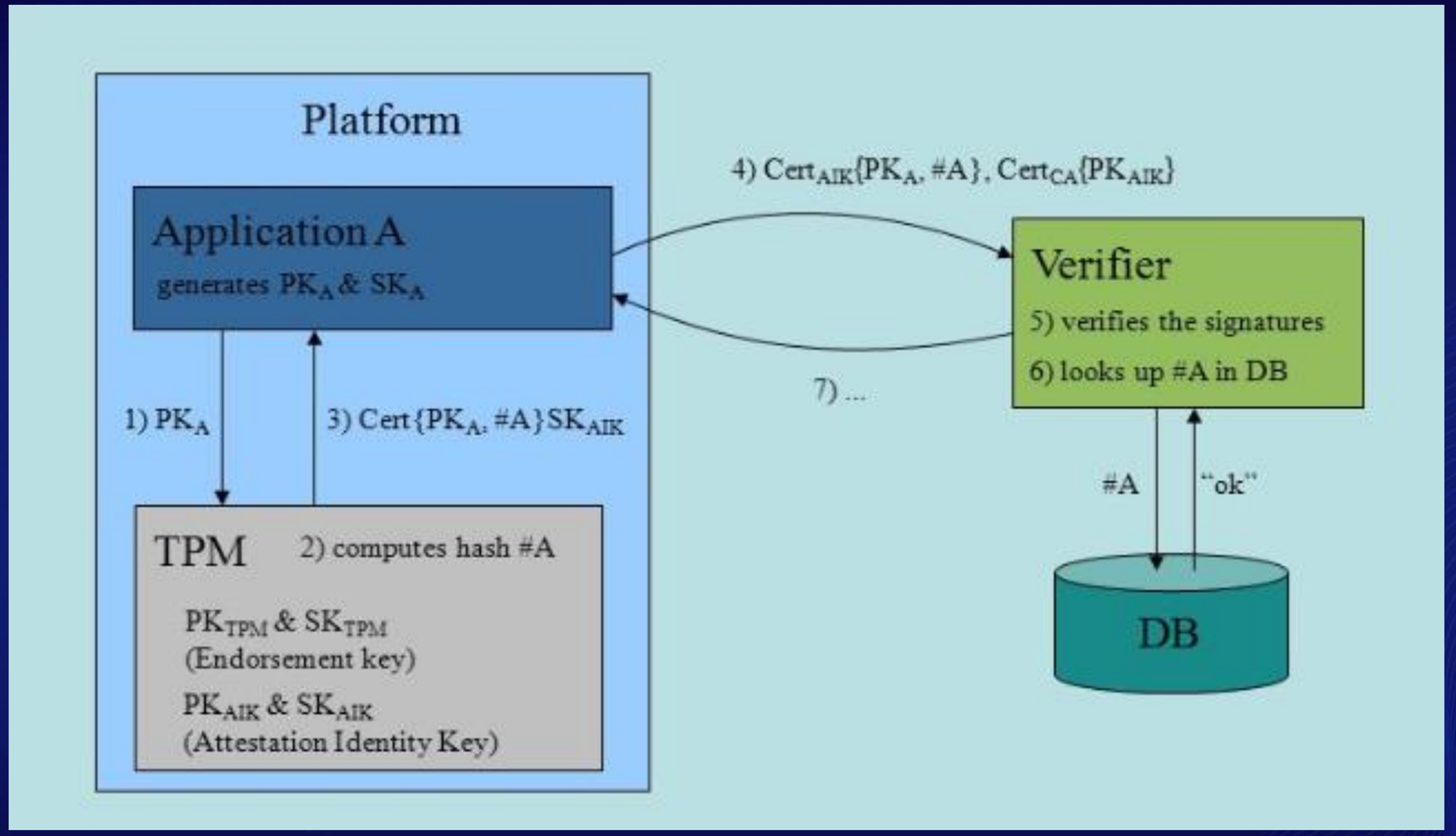
- CHERI会成为内存安全的终结者吗?

HUAWEI

可信启动


Give control to next component if measurement matches value stored in RTS
Root of Trust For Measurement
BIOS
Option ROM
OS Loader
• Measurement
Root of Trust For Reporting
Root of Trust For Storage
TPM
Copyright© 2011 Trusted Computing Group

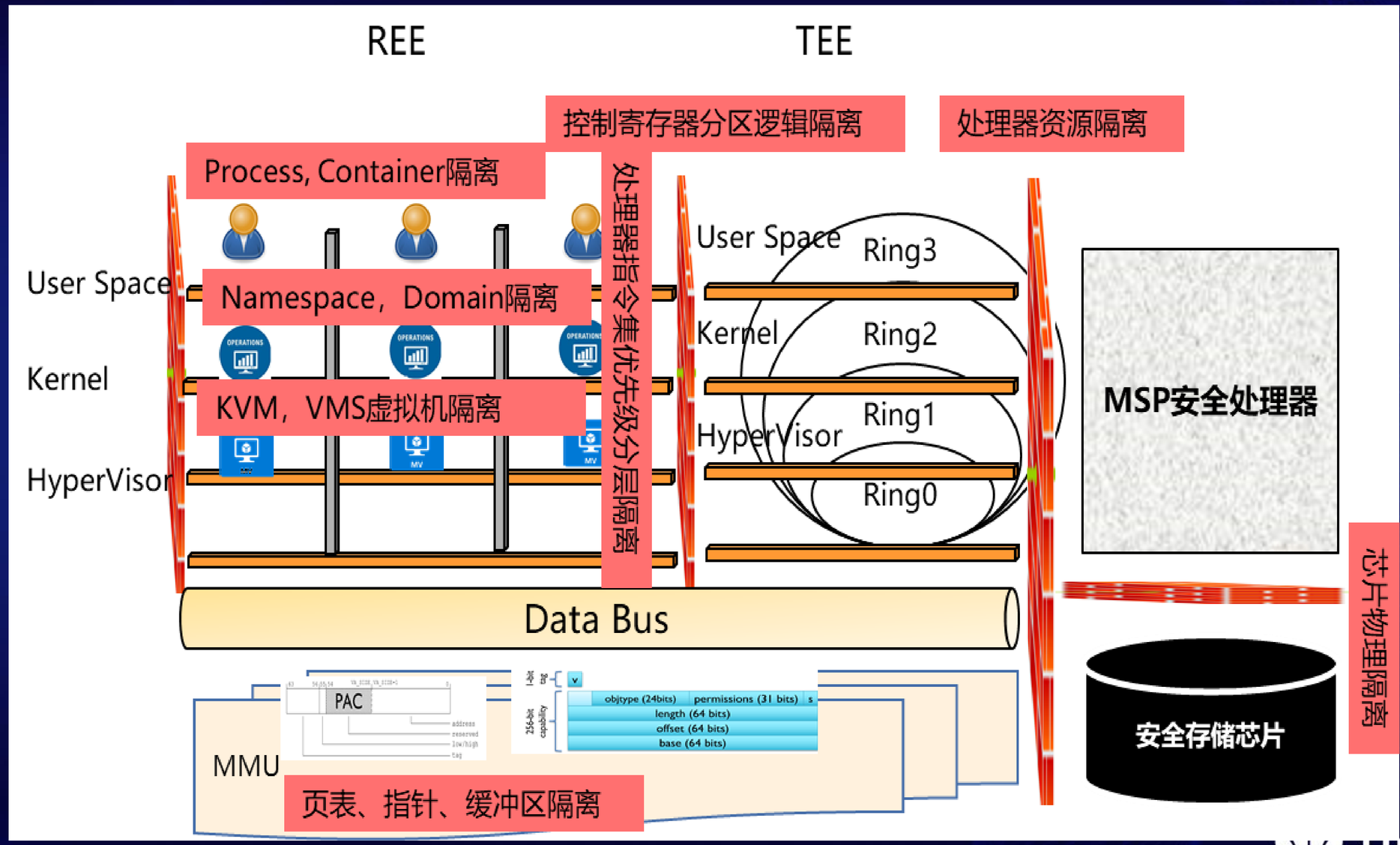
远程证明


Platform
Application A
generates PK_A & SK_A
1) PK_A
3) Cert{PK_A, #A}SK_{AIK}
TPM
2) computes hash #A
PK_{TPM} & SK_{TPM}
(Endorsement key)
PK_{AIK} & SK_{AIK}
(Attestation Identity Key)
4) $Cert_{AIK}\{PK_A, \#A\}, Cert_{CA}\{PK_{AIK}\}$
7) ...
Verifier
5) verifies the signatures
6) looks up #A in DB
#A
"ok"
DB

- 计算机安全发展演进历史
- 典型系统安全架构模型

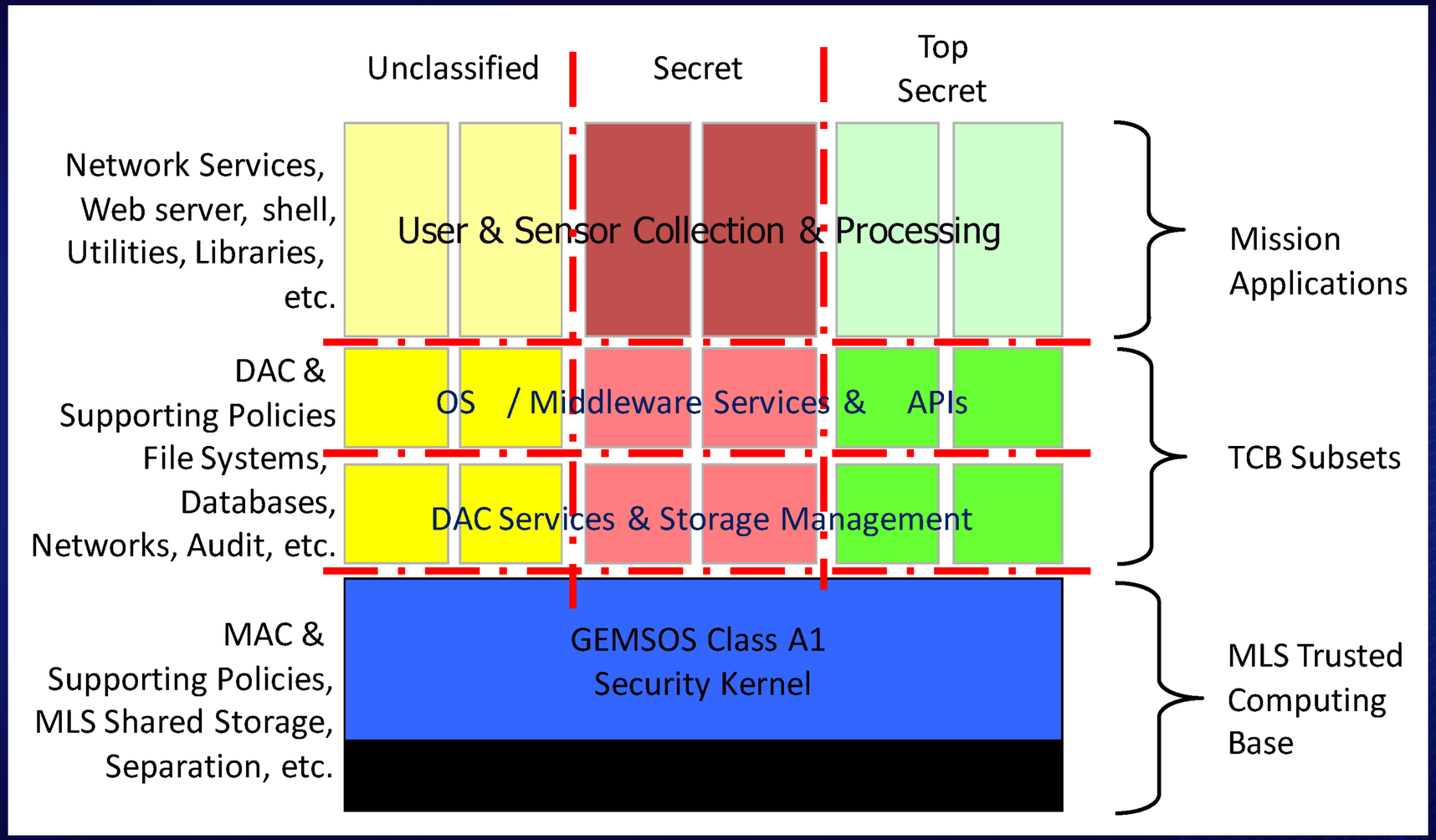
HUAWEI

隔离与访问控制模型



HUAWEI

史上第一个A1级安全OS架构：GEMSOS



HUAWEI

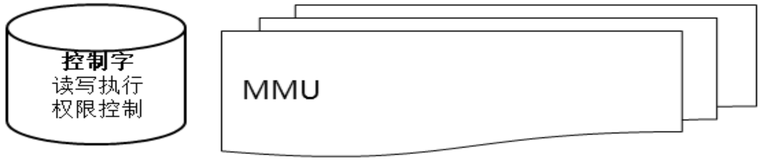
微观隔离与访问控制模型

内存分段隔离

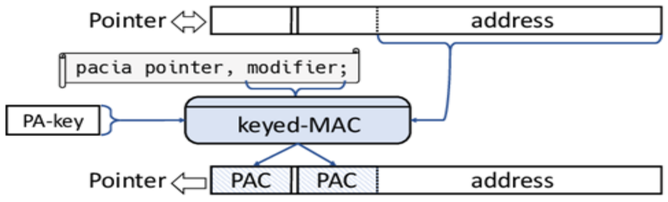
只读，可执行，不可写　只读，可执行，不可写　可读写，不可执行

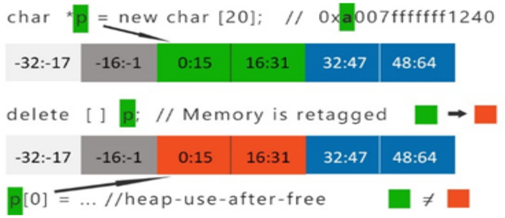
代码段	堆栈段	数据段	扩展段

内存分页隔离



内存权限隔离



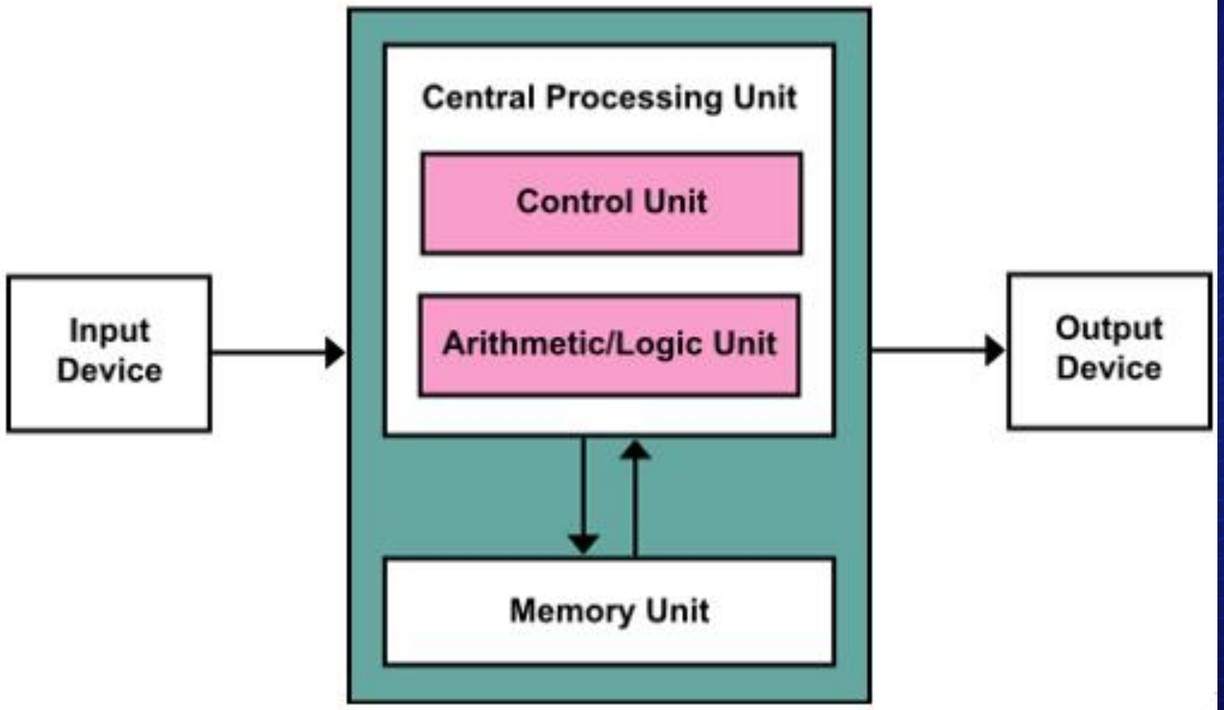


内存Buffer隔离

HUAWEI

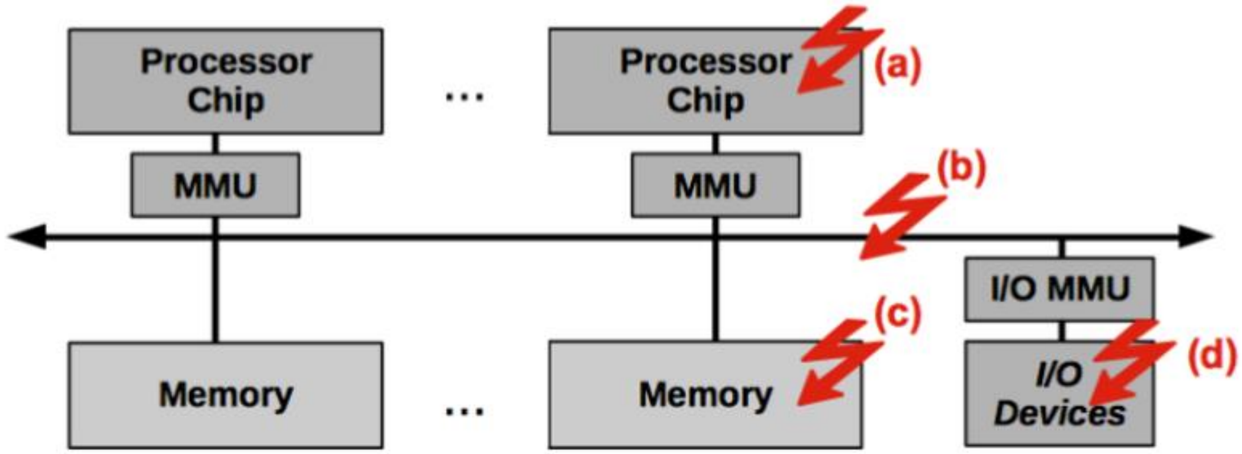
SL0：石器时代

最初始状态的处理器的架构如下：

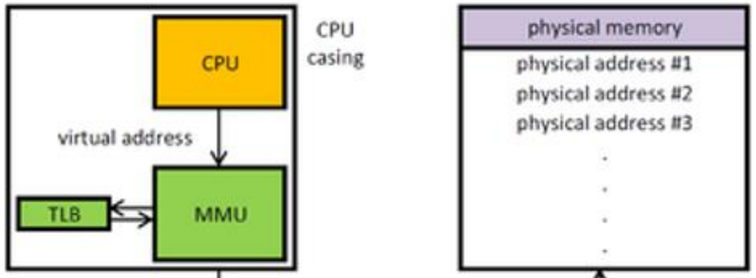


HUAWEI

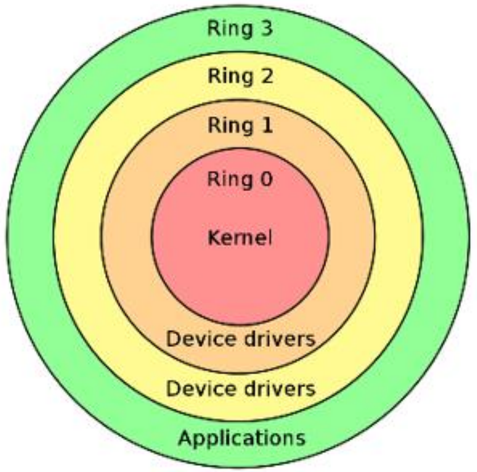
SL1：铁器时代

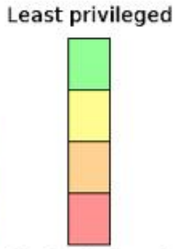


Tutorial on Principles of Secure Processor Architecture Design



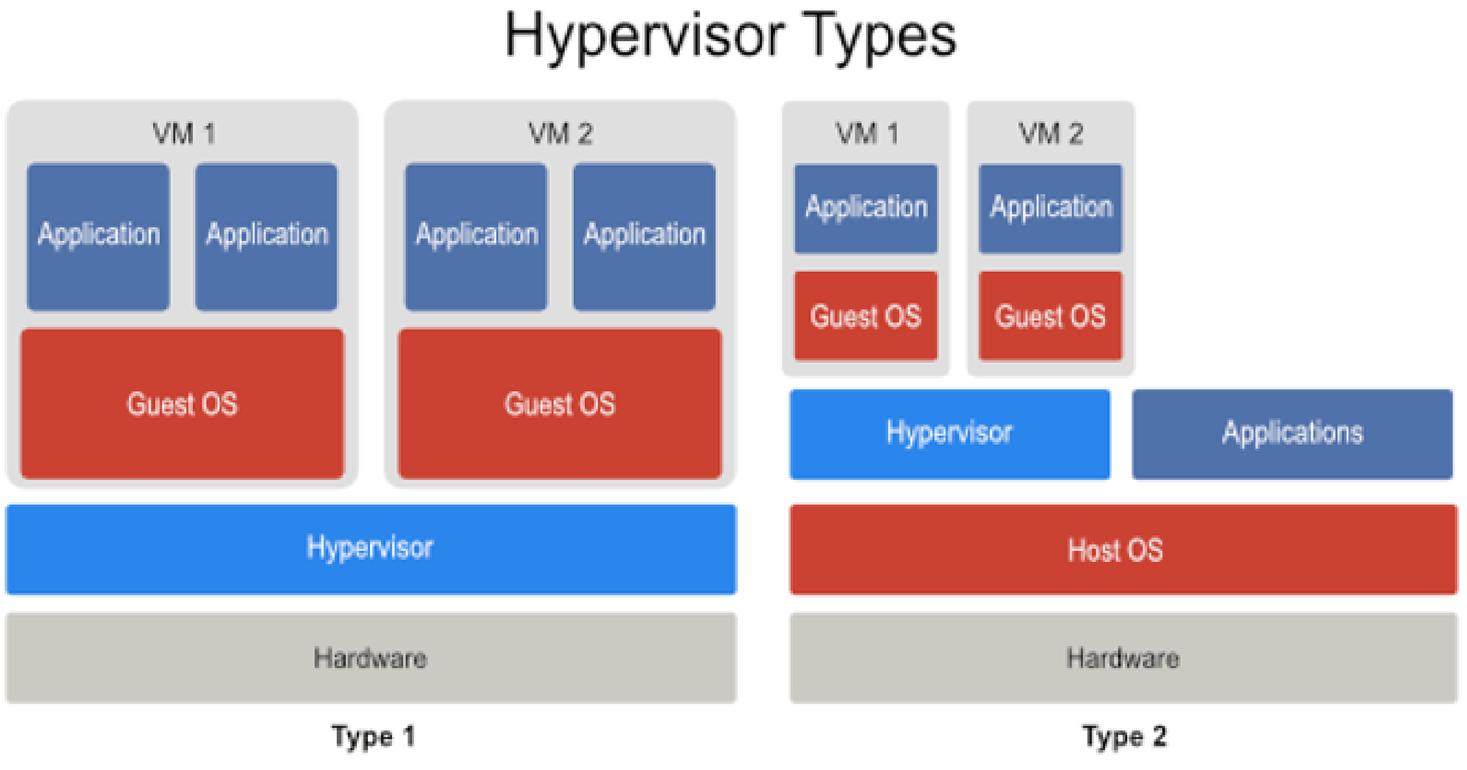
CPU: Central Processing Unit
MMU: Memory Management Unit
TLB: Translation lookaside buffer



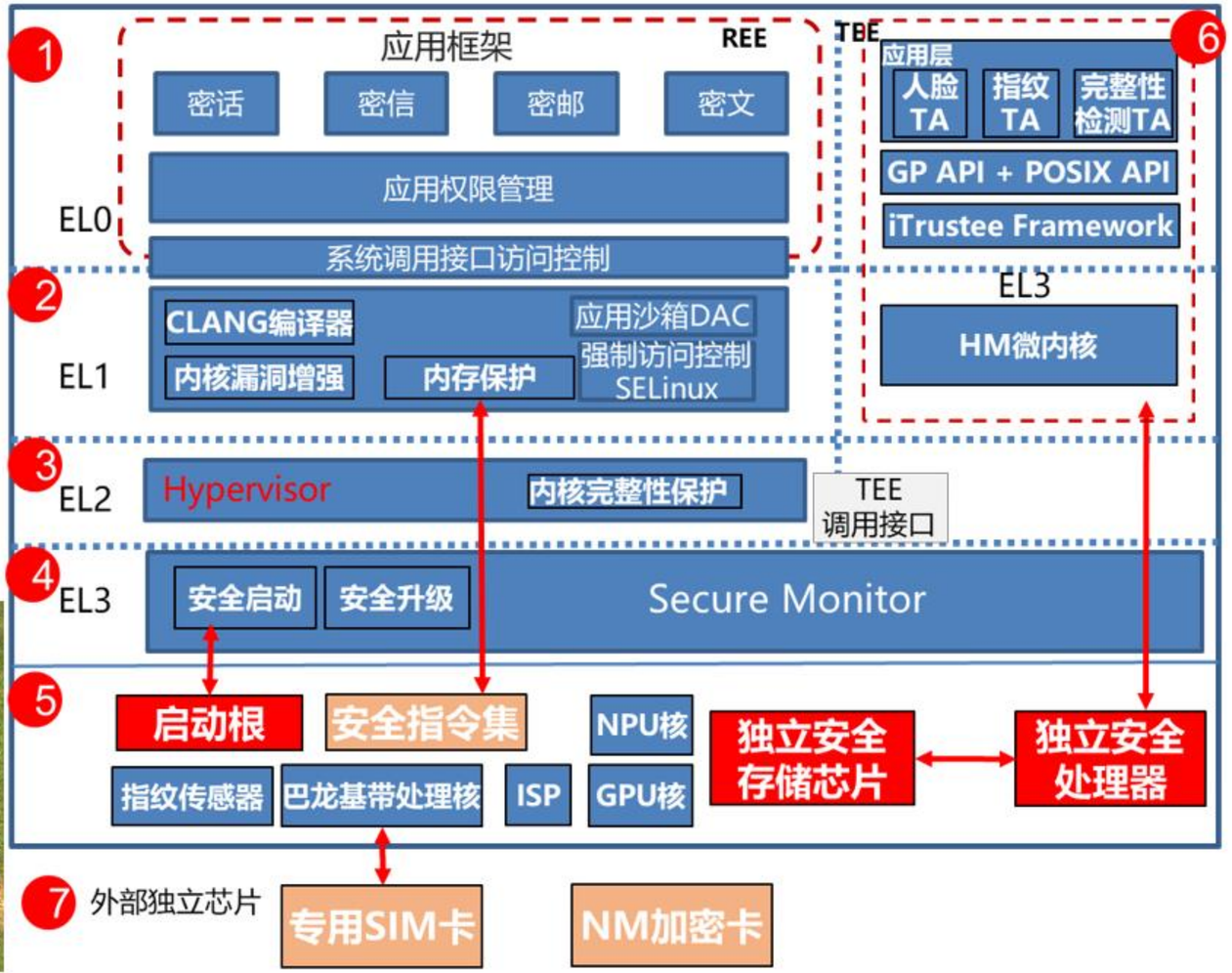


HUAWEI

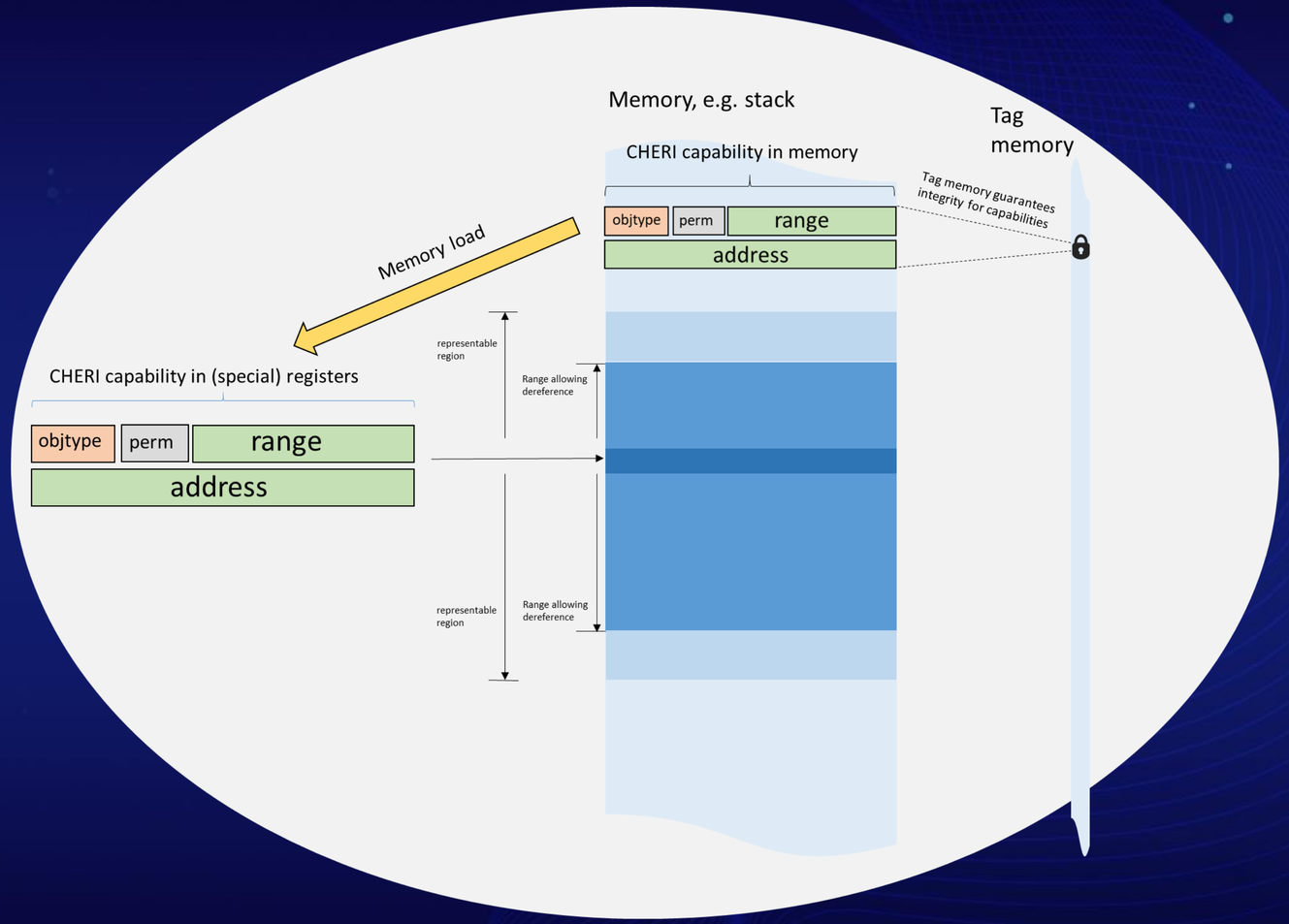
SL2：蒸汽时代


Hypervisor Types
VM 1
Application
Application
Guest OS
VM 2
Application
Application
Guest OS
Hypervisor
Hardware
Type 1
VM 1
Application
Guest OS
VM 2
Application
Guest OS
Hypervisor
Applications
Host OS
Hardware
Type 2

SL3：电气时代


1
应用框架
REE
密话
密信
密邮
密文
应用权限管理
EL0
系统调用接口访问控制
TBE
6
应用层
人脸TA
指纹TA
完整性检测TA
GP API + POSIX API
iTrustee Framework
EL3
HM微内核
2
CLANG编译器
应用沙箱DAC
内核漏洞增强
内存保护
强制访问控制 SELinux
EL1
3
EL2
Hypervisor
内核完整性保护
TEE 调用接口
4
EL3
安全启动
安全升级
Secure Monitor
5
启动根
安全指令集
NPU核
独立安全存储芯片
独立安全处理器
指纹传感器
巴龙基带处理核
ISP
GPU核
7
外部独立芯片
专用SIM卡
NM加密卡

SL-x：巴别塔?



https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-850.pdf

HUAWEI

接下来：

HarmonyOS安全设计实践

HUAWEI