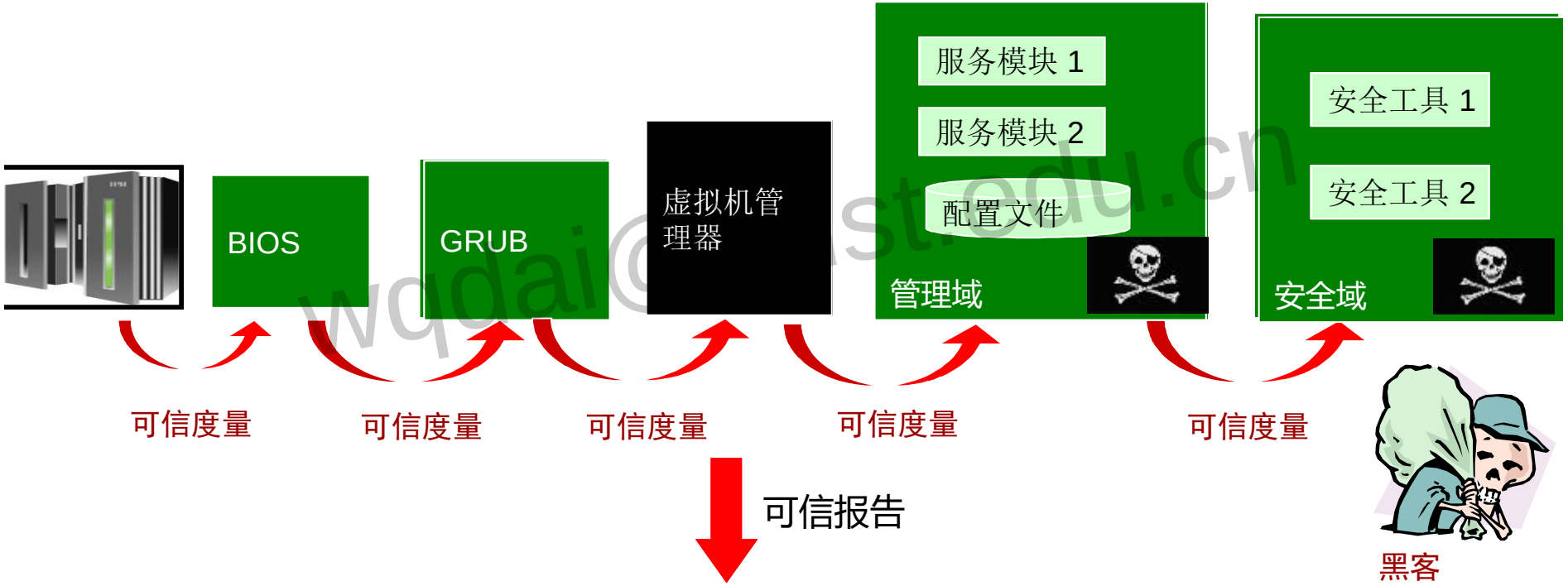
传统静态安全执行环境构建

静态方法从硬件自检开始传递可信链。



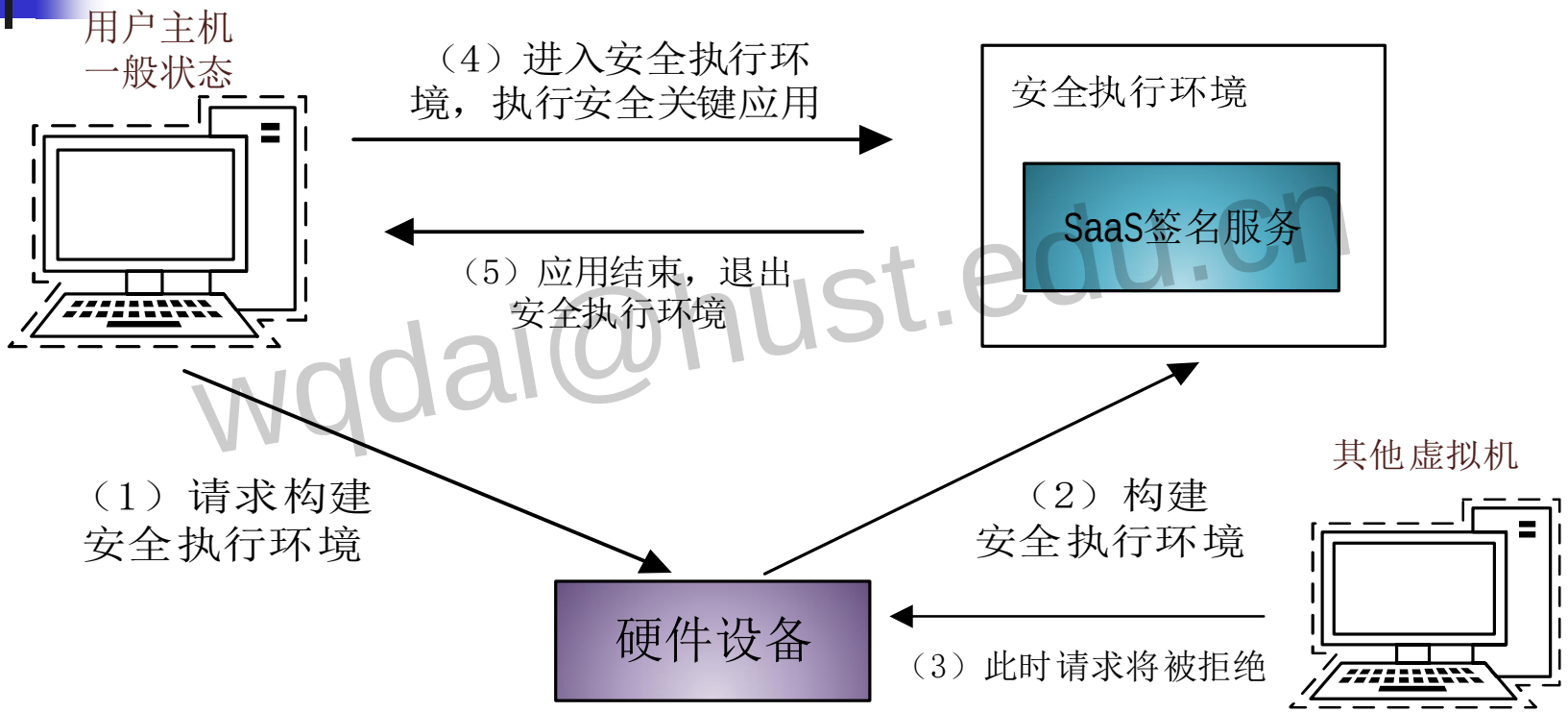
云平台的可信验证

可信验证后，黑客入侵云平台，可信的验证结果失效

wqdai@hust.edu.cn

传统动态安全执行环境构建

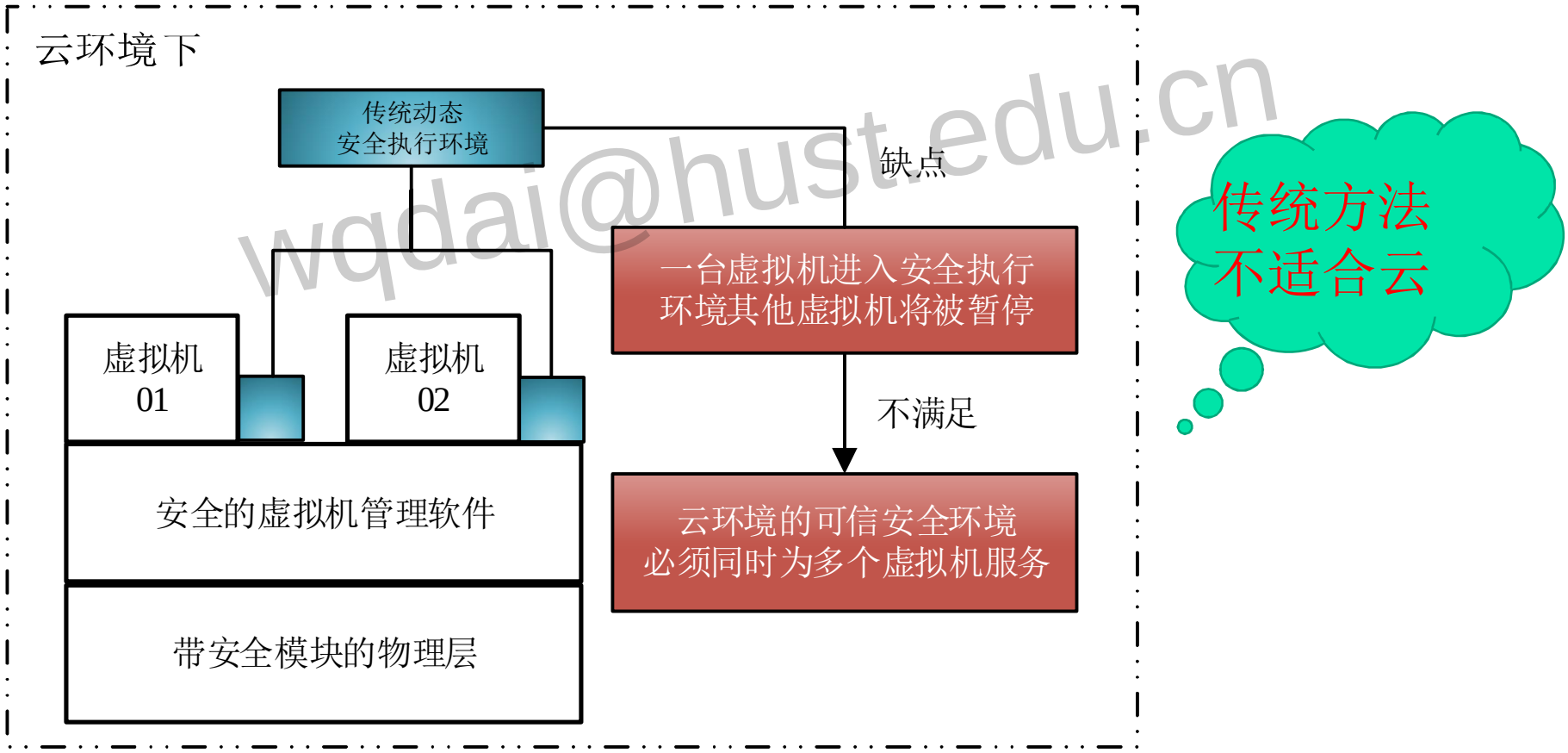
动态方法运行过程中动态创建安全环境。



wqdai@hust.edu.cn

云环境中构建安全执行环境的问题

- 传统的静态安全度量机制并不能保证运行时安全。
- 传统的动态的安全度量机制并不适合云环境。

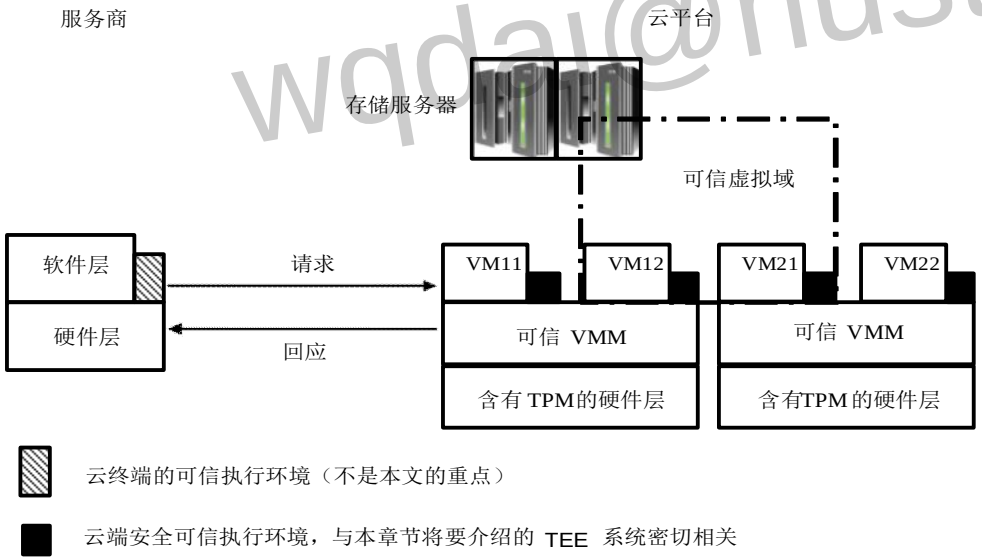


wqdai@hust.edu.cn

云环境中构建安全执行环境的问题

- 云端完全建立在虚拟机上，与此同时租户操作系统通常可以被轻易的攻破（比如探测它们的漏洞），因此并不能充分地保证其本身的安全性。
- 现有的可信执行环境构建技术都不能在安全敏感应用运行时提供足够的安全，而一些抵御恶意操作系统攻击的解决方案也不适用于云端。

wqdai@hust.edu.cn



云平台动态可信度量机制

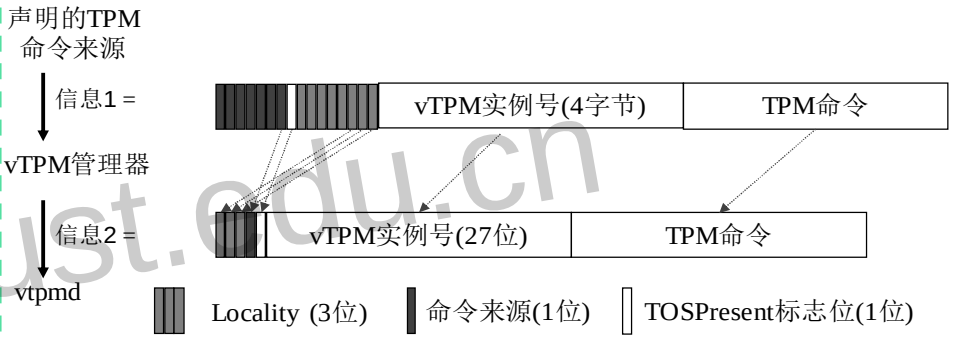
- 可信执行环境
- 兼有虚拟化和可信计算技术的优点
- 虚拟化的动态可信度量根。

虚拟动态可信度量根

（1）设计虚拟动态可信度量根的Locality

Localities	DRTM下使用者	vDRTM下使用者
Locality 4	可信硬件（DRTM）	vDRTM
Locality 3	辅助组件	辅助组件
Locality 2	可信OS的启动环境	启动时候的TEE内核
Locality 1	可信OS	启动后的TEE域
Locality 0	传统SRTM环境和其信任链	不可信客户OS

（2）设计支持vDRTM的vTPM管理器




```
1:   if 接收到的命令来源等于 1 then
2:     if 接收到的 TOSPresent 等于 0 then
3:       设置 localityModifier 为 000;
4:     end if
5:     设置 TOSPresent 位为接收值;
6:   end if
7:   if TOSPresent 等于 1 then
8:     设置 localityModifier 为接收到的 locality 位的值;
9:   end if
```

wqdai@hust.edu.cn

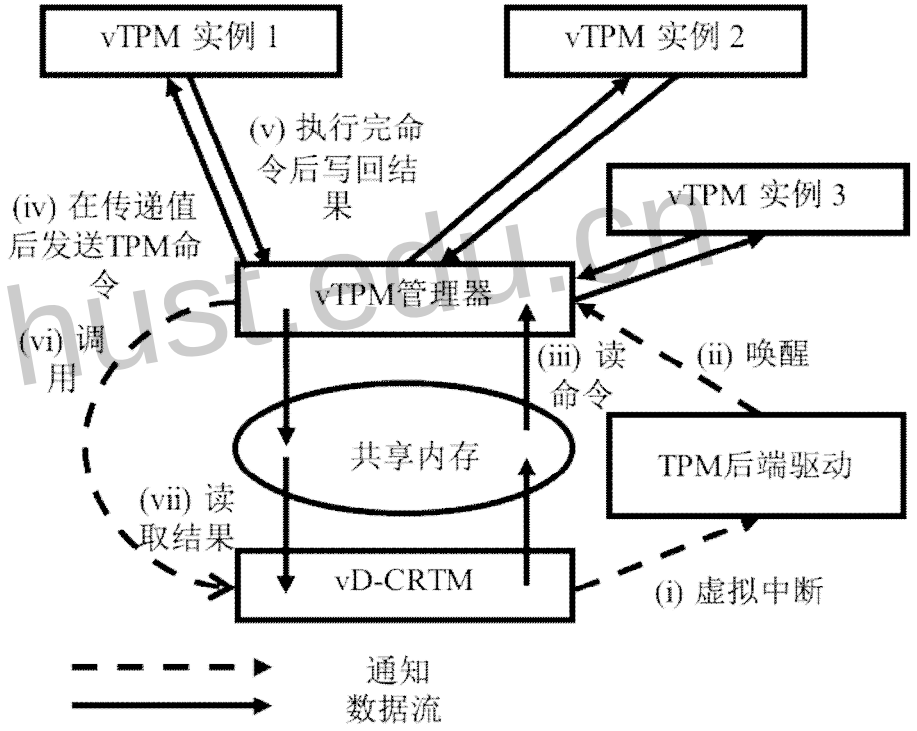
虚拟动态可信度量根

（3）设计vD-CRTM

- 接收和鉴别加载TEE的系统调用
- 度量TEE内核和应用程序
- 扩展度量值到相应PCR中

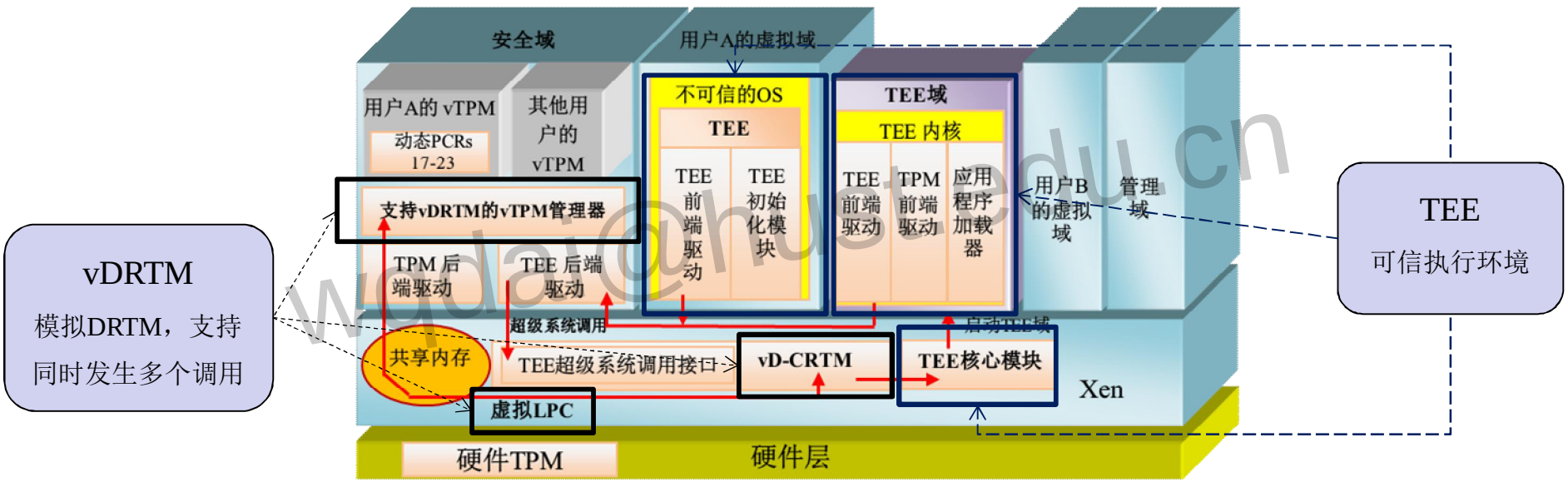
（4）设计Virtual LPC（Low Pin Count）

- 通过Xen和vTPM管理器之间共享内存实现



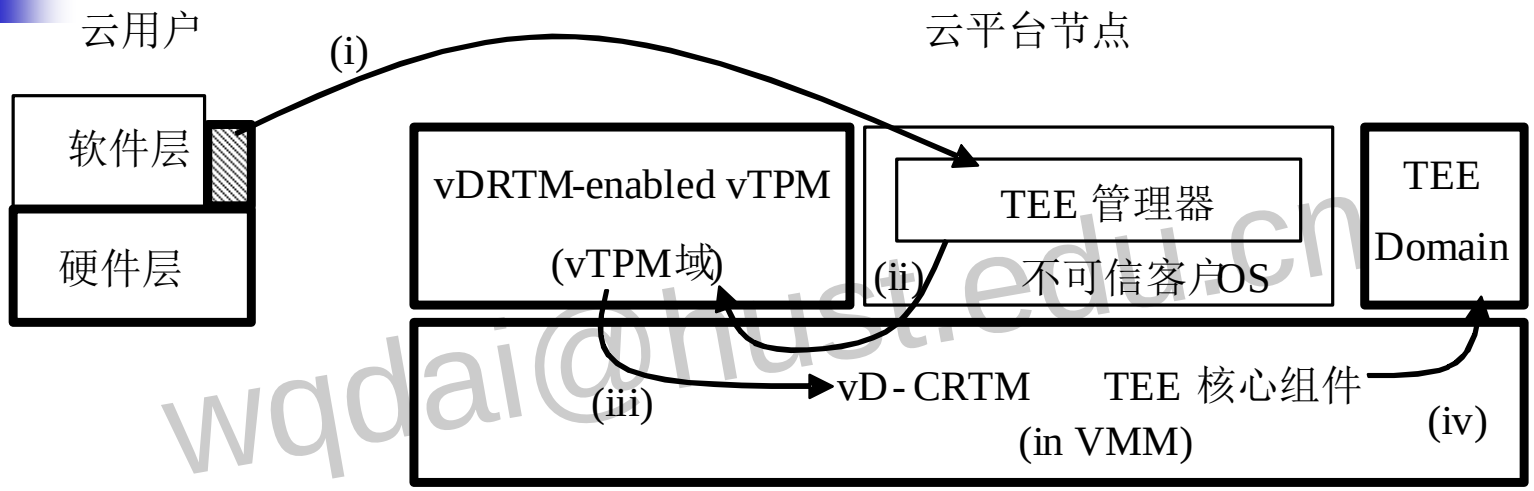
wqdai@hust.edu.cn

TEE结构



wqdai@hust.edu.cn

运行流程

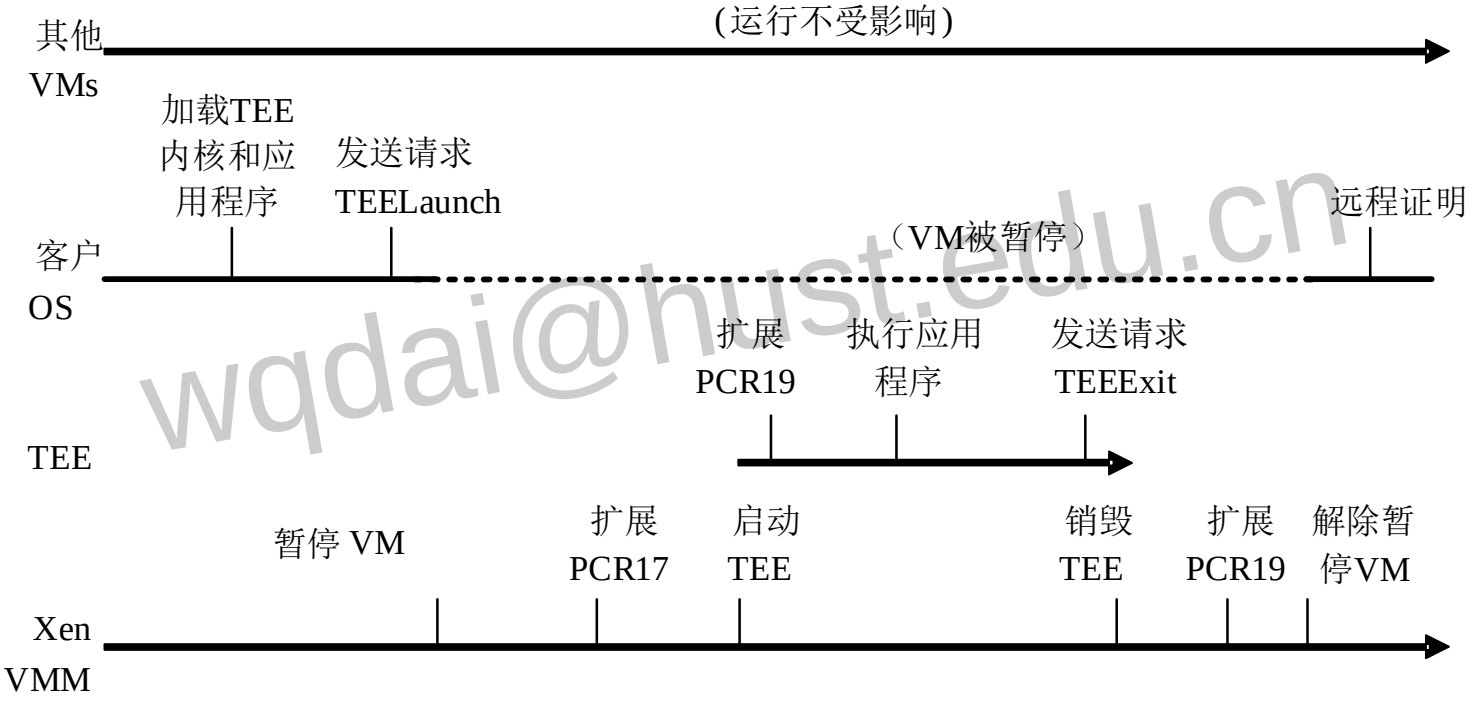


wqdai@nuaa.edu.cn

用户安全执行环境

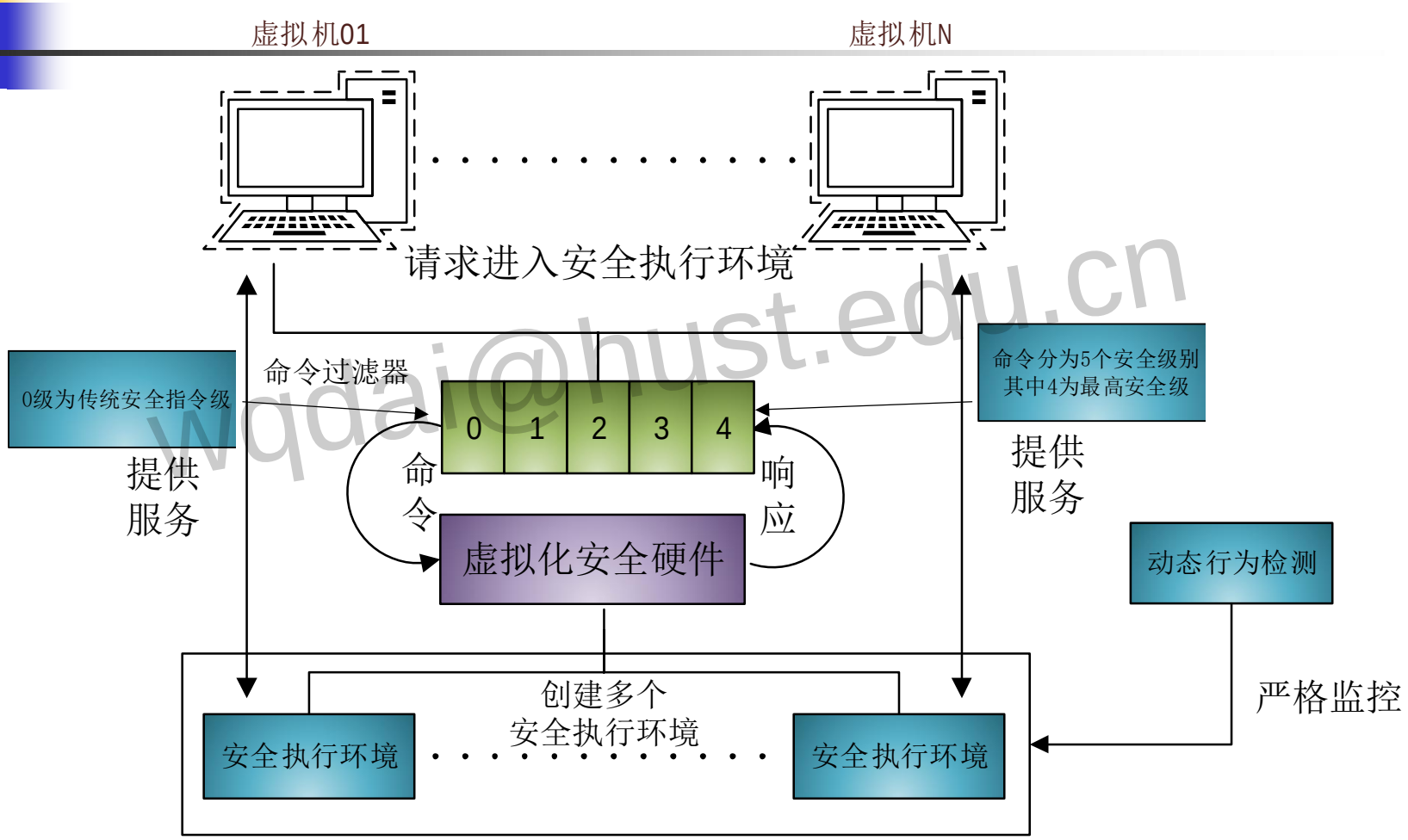
可信组件（vTPM域，vD-CRTM and TEE核心组件，TEE域）

运行流程



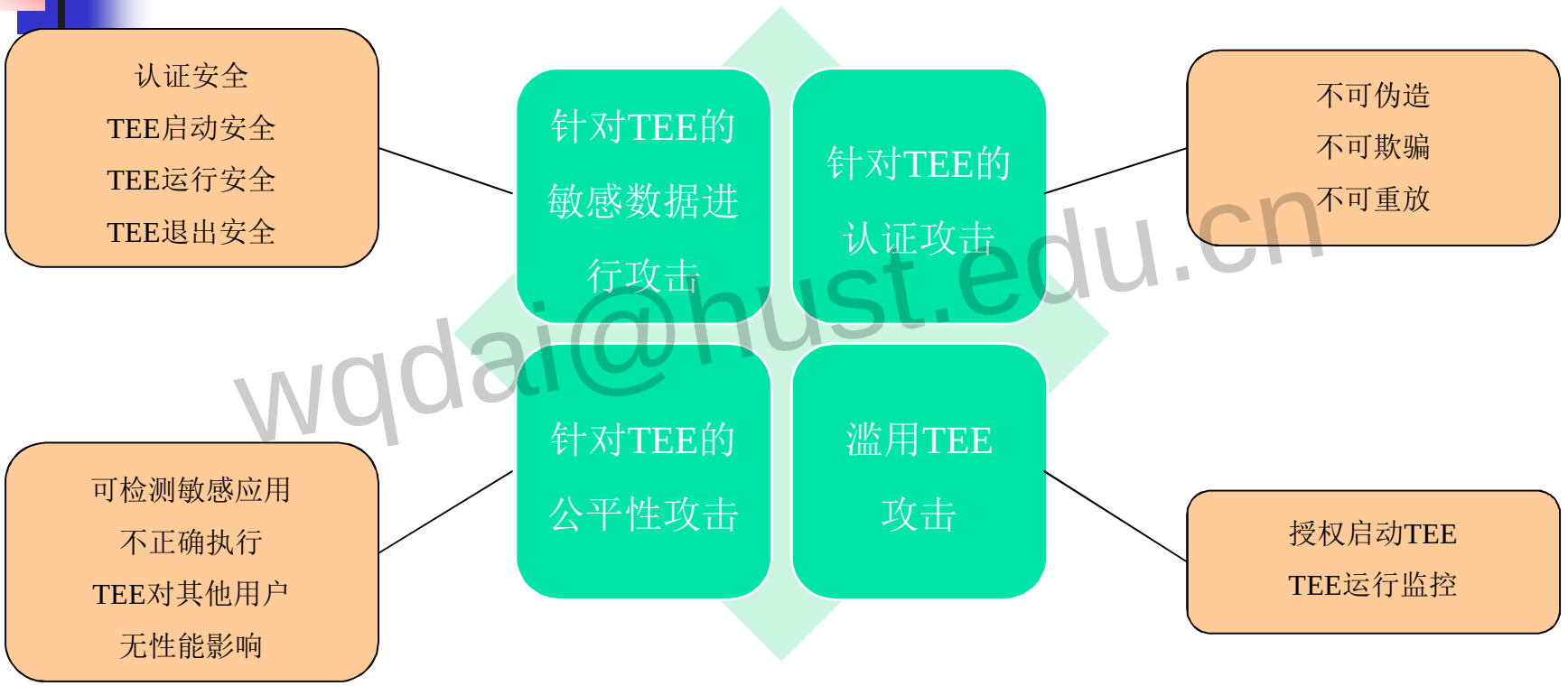
wqdai@hust.edu.cn

为云环境构建安全执行环境

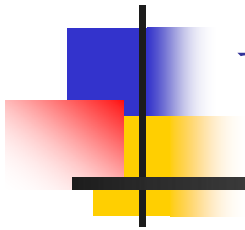


wqdai@hust.edu.cn

TEE安全性分析

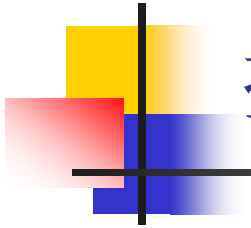


wqdai@hust.edu.cn



可信启动

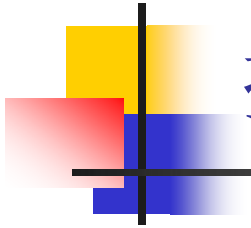
wqdai@hust.edu.cn



提纲

- Linux的启动过程
- TrustGrub
- Grub-IMA
- OSLO
- Tboot

wqdai@hust.edu.cn



提纲

- Linux的启动过程
- TrustGrub
- Grub-IMA
- OSLO
- Tboot

wqdai@hust.edu.cn

Linux启动过程概述

- Linux启动过程指的是从打开计算机电源直到显示用户登录画面的全过程。
- Linux启动的大致流程：

wqdai@hust.edu.cn

（1）打开电源，CPU进入实模式

- X86架构处理器的模式：
  - 实模式（Real Mode）：内存寻址同8086处理器（16位段寄存器和偏移量），寻址范围为1M。
  - 保护模式（Protection Mode）：采用32位段寄存器和偏移量，最大寻址空间4GB，CPU分为4个特权级（0-3）。
  - 系统管理模式（System Management Mode）：拥有独立的地址空间，用来执行电源管理方面的指令。
  - 虚拟8086模式（Virtual-8086 Mode）：允许在保护模式中运行传统的8086程序。
- 开机时系统处于实模式，操作系统（例如Linux、Windows）运行于保护模式。

wqdai@hust.edu.cn

（2）BIOS开机自检

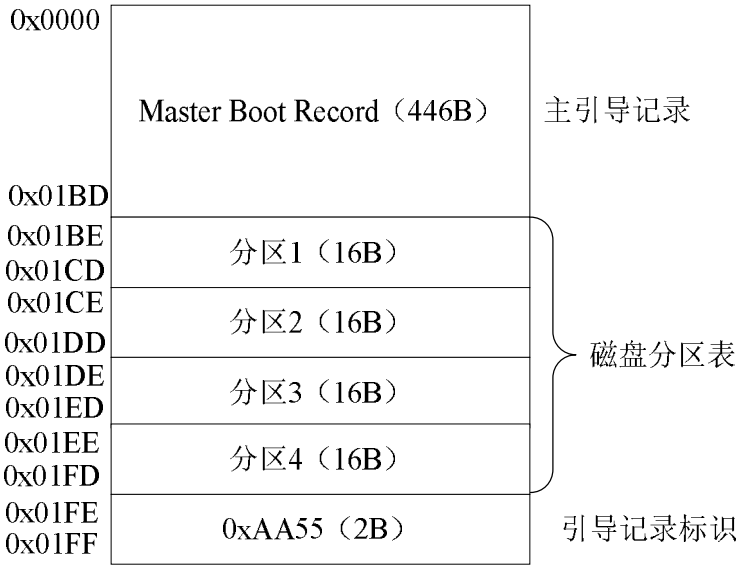
- BIOS一般被存放在只读存储器（Read-Only Memory，简称ROM）之中，即使在关机或掉电以后，这些代码也不会消失。BIOS的代码是直接与硬件打交道，它为操作系统提供了控制硬件的基本功能。BIOS包括有系统BIOS（即常说的主板BIOS）、显卡BIOS和其它设备（例如IDE控制器或者网卡等）的BIOS。其中计算机的启动过程是在系统BIOS的控制下进行的。系统BIOS的启动代码要做的事情就是进行加电后自检（Power-On Self Test，POST）。POST的主要任务是检测系统中一些关键设备是否存在和能否正常工作，例如内存和显卡等设备。

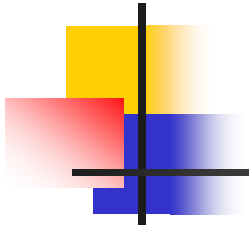
wodai.cqust.edu.cn

（3）启动引导程序

- 硬盘的第1个扇区被称为主引导扇区，它在所有的系统分区之前，不属于任何系统分区，其包括：
  - 主引导记录（Master Boot Record，简称MBR）
  - 磁盘分区表（Disk Partition Table，简称DPT）
  - 引导记录标识（Boot Record ID）

wqdai@hust.edu.cn





引导加载程序

- 引导加载程序（boot loader）
  - LILO：作为一个较老的Linux 引导加载程序，它在Linux社区的支持下发展，并始终能够充当一个可用的现代引导加载程序。有一些新的功能，比如增强的用户界面，以及对能够突破原来1024-柱面限制的新BIOS 功能的利用。
  - Grub：与LILO相比，Grub（Grand Unified Boot Loader）拥有强大的交互式命令界面，支持网络引导，而且更加可靠、灵活和易于使用。Grub负责装入内核并引导Linux系统，此外还可以引导其它操作系统，如FreeBSD、OpenBSD、DOS和Windows系列。

wqdai@hust.edu.cn

Grub

- Grub包含2个阶段：Stage1和Stage2。
  - Stage1：加载Stage2并跳转执行。
  - Stage2：加载其他操作系统。

wqdai@hust.edu.cn

```
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
#hiddenmenu

title Fedora (2.6.23.1-42.fc8)
        root (hd0,0)
        kernel /boot/vmlinuz-2.6.23.1-42.fc8 ro root=LABEL=/ rhgb quiet
        initrd /boot/initrd-2.6.23.1-42.fc8.img
title Windows XP
        rootnoverify (hd0,1)
        chainloader +1
```

（4）加载内核

- 根据Grub设定的启动项，读取内核映像，并进行解压缩操作。

- 系统将解压后的内核放置在内存之中，并调用start_kernel()函数来启动一系列的初始化函数并初始化各种设备，完成Linux核心环境的建立。start_kenrel()定义在init/main.c中，它就类似于一般可执行程序中的main()函数，系统在此之前所做的仅仅是一些能让内核程序最低限度执行的初始化操作，真正的内核初始化过程是从这里才开始。函数start_kerenl()将会调用一系列的初始化函数，用来完成内核本身的各方面设置，目的是最终建立起基本完整的Linux核心环境。

wqdai@hust.edu.cn

（5）完成系统初始化

- 内核被加载后，第一个运行的程序便是/sbin/init，该文件会读取/etc/inittab文件，并依它来设置Linux的运行等级。

设置默认的运行等级

```
id:5:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
```

- Linux的运行等级设定如下：
  - 0：关机
  - 1：单用户模式
  - 2：无网络支持的多用户模式
  - 3：有网络支持的多用户模式
  - 4：保留，未使用
  - 5：图形的多用户模式（X11）
  - 6：重启

wqdai@hust.edu.cn

（5）完成系统初始化

- 在设定了运行等级后，Linux系统执行的脚本程序是/etc/rc.d/rc.sysinit，其包括：
  - 设置环境变量
  - 读取/etc/sysconfig/network
  - 使用fsck检测文件系统
  - 启动信息存入/var/log/dmesg
  - 使用/etc/sysconfig/clock文件初始化clock
  - 检测系统参数proc并设置PNP
  - 运行rc.serial对串行端口初始化
  - 安装root、proc文件系统以及其它文件系统
  - 设置字体，启动Swapping等操作

wqdai@hust.edu.cn

（5）完成系统初始化

- **启动内核模块：**根据/etc/modprobe.conf文件来装载内核模块。

- **执行不同运行级别的脚本程序：**根据运行级别的不同，系统会运行rc0.d到rc6.d中的相应的脚本程序，来完成相应的初始化工作和启动相应的服务。

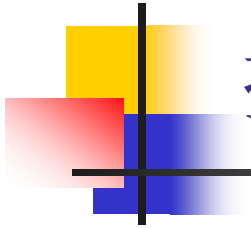
- **执行/etc/rc.d/rc.local：**Linux留给用户进行个性化的地方。你可以把你想设置和启动的东西放到这里。

wqbai@hust.edu.cn

（6）打开终端供用户登录

- 启动mingetty，其主要包括：
  - 打开终端，并设置模式。
  - 输出登录界面及提示，接受用户名的输入。
  - 以该用户名作为login的参数，加载login程序。
- 当用户登录通过了这些检查后，login将搜索/etc/passwd文件（必要时搜索 /etc/shadow文件）用于匹配密码、设置主目录和加载shell。如果没有指定主目录，将默认为根目录；如果没有指定shell，将默认为/bin/sh。在设置好shell的用户ID(uid)、组ID(gid)，以及PATH等环境变量以后，进程加载shell。

wddai@hust.edu.cn



提纲

- Linux的启动过程
- TrustGrub
- Grub-IMA
- OSLO
- Tboot

wqdai@hust.edu.cn

基于TCG扩展的引导过程

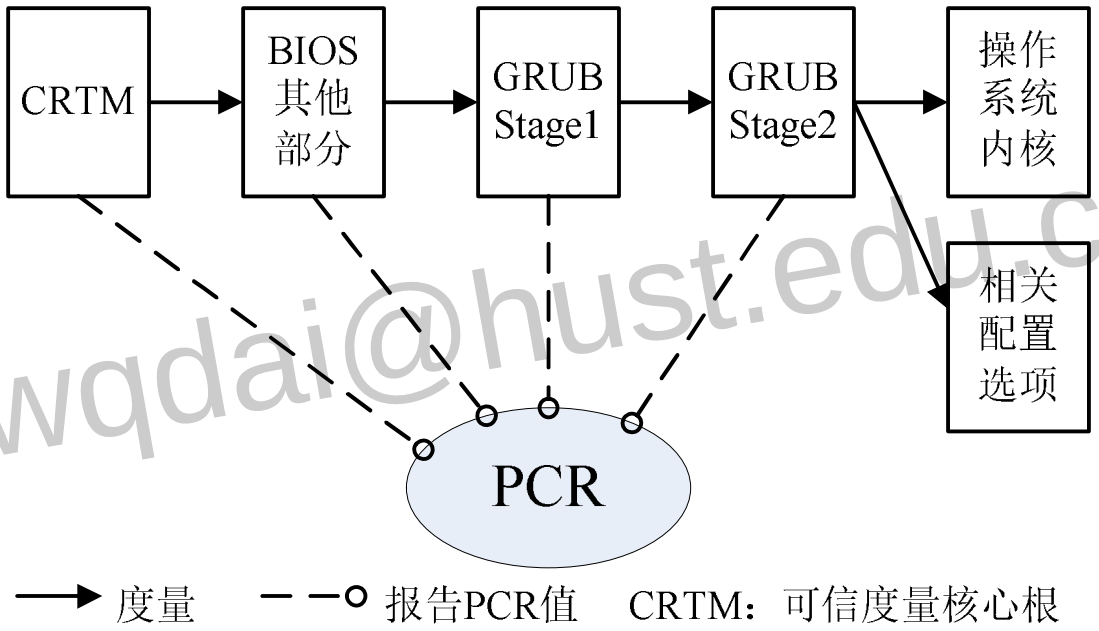
- 可信链（chain-of-trust）

- 可信根（Root of Trust）：默认被信任的实体。在上例中，实体A被认为是可信度量根（Root of Trust for Measurement，简称RTM），因为它被用来度量其他实体。

- 可信度量核心根（Core Root of Trust for Measurement，简称CRTM）是BIOS的引导块代码，这段代码被认为是可信平台的基础。它是BIOS的不可修改部分，在平台的整个生命过程中保持不变。

wodai@hust.edu.cn

基于TCG扩展的度量过程



wqdai@hust.edu.cn

- 信任链从CRTM开始，在加载BIOS其他部分之前，先报告其PCR值，然后将执行权交给BIOS其他部分。在Grub的Stage2，为了支持多操作系统启动，除了度量选择的操作系统内核，还需要度量相关的配置信息（内核模块等）。

引导加载程序的度量方法

- 引导加载程序度量方法：
  - 静态可信度量根（Static Root of Trust for Measurement，简称SRTM）：由基于TCG扩展的BIOS提供，例如TrustedGrub和Grub-IMA。
  - 动态可信度量根（Dynamic Root of Trust for Measurement，简称DRTM）：由特定的CPU指令（Intel的SENTER和AMD的SKINIT）提供，例如OSLO和Tboot。

- 两种不同的方法，度量值都被保存在TPM的PCR中。

wqdai@hust.edu.cn

TrustedGrub

- TrustedGrub是Sourceforge支持的开源项目，它扩展了原始的Grub引导程序。它的主要特性是支持TPM提供的可信计算功能，在启动过程中度量任意文件并将完整性度量结果扩展到相应的PCR中。

- 在TrustedGrub的引导过程中，它度量了BIOS、Grub和操作系统内核并报告PCR值。BIOS度量主引导扇区，然后加载并执行。TrustedGrub扩展了Stage1，可以度量Stage2的第1个扇区（第1部分），Stage2的其他部分被称为第2部分。为了保证启动过程的完整性，TrustedGrub必须度量Stage2的输入参数，包括内核，启动模块和相关配置信息。Grub的默认配置选项在配置文件（grub.conf）中指定，由Stage2加载。然而，用户能够改变和扩展Grub提供shell的配置选项。配置选项的指令并不是立即被执行，而是存储在一个列表中并不执行直到用户选择配置来启动。

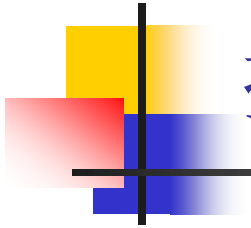
wdcj@hust.edu.cn

TrustedGrub（续）

- TrustedGrub遵守基于TCG扩展的PC客户端规范。TrustedGrub通过度量需要加载的多启动模块扩展了PCR，所有与操作系统内核和模块相关的度量值保存在PCR14中。

wqdai@hust.edu.cn

PCR	用途
4	Stage1
8	Stage2的第1部分
9	Stage2的第2部分
12	命令行参数
13	checkfile度量的程序
14	内核和模块



提纲

- Linux的启动过程
- TrustGrub
- Grub-IMA
- OSLO
- Tboot

wqdai@hust.edu.cn

Grub-IMA

- GRUB-IMA（Integrity Measurement Architecture）是由IBM提出的基于Linux的完整性度量架构，它建立了一个可信的平台并能够向远程用户报告平台和其上软件的完整性。
- 它是对Linux的引导加载器－GRUB增加了TCG度量的功能，它支持客户端PC的TCG 1.1b规范，并为TCG软件栈（TSS）的实例－TrouSers提供基础。当前平台的所有度量的程序形成了一个度量值列表，它反映了平台及其应用程序的状态。

wddai@hust.edu.cn

Grub-IMA（续）

- GRUB-IMA集成了对BIOS，GRUB和操作系统的度量值，并支持在远程系统上查看这些Hash值。此外，它还支持通过事件类型将度量值存储在ACPI度量日志中。

wqdai@hust.edu.cn

PCR	用途
4	Stage1和Stage2
5	事件日志
8	操作系统组件（内核，模块等）

Grub-IMA（续）

- 在GRUB启动之后，GRUB-IMA还可以度量配置文件（grub.conf）中指定的程序和操作系统内核相关的模块。下图中在加载Linux内核和ramdisk之前，先度量两个脚本程序rc.sysinit和rc.local。Linux内核和ramdisk的Hash值被扩展到PCR8，而rc.sysinit和rc.local的Hash值被扩展到PCR9。

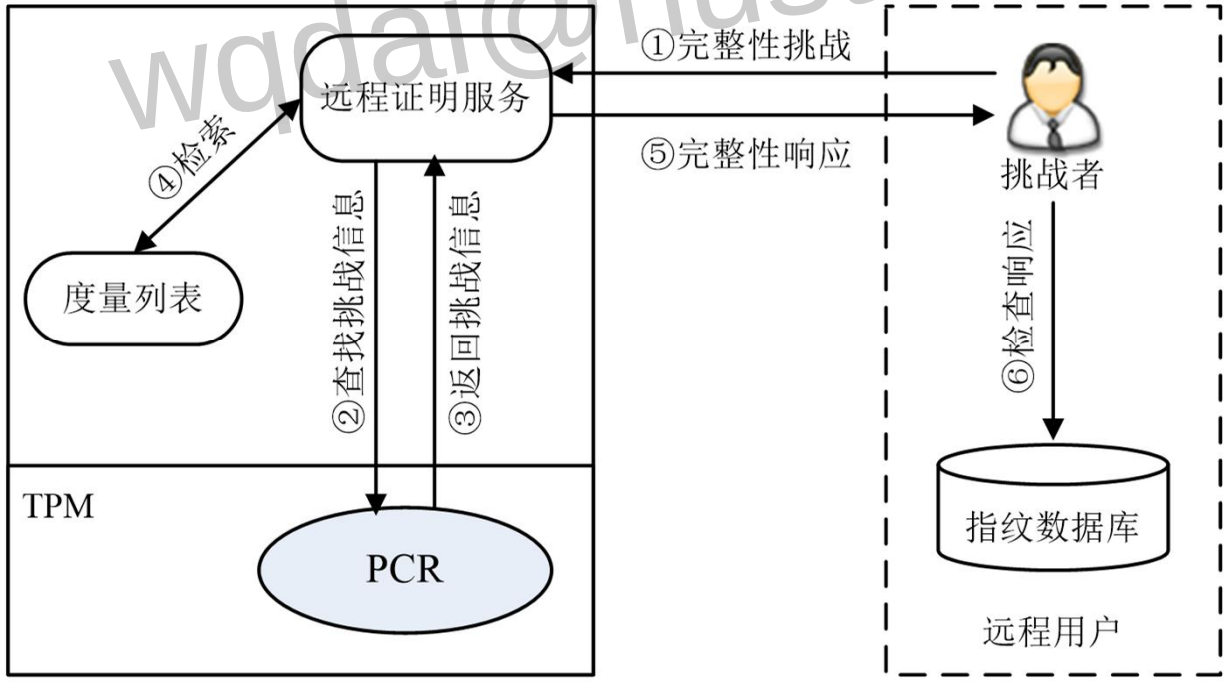
```
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
#hiddenmenu

title Fedora (2.6.23.1-42.fc8)
        root (hd0,0)
      measure (hd0,0)/etc/rc.sysinit 9
      measure (hd0,0)/etc/rc.local 9
        kernel /boot/vmlinuz-2.6.23.1-42.fc8 ro root=LABEL=/ rhgb quiet 8
        initrd /boot/initrd-2.6.23.1-42.fc8.img
title Windows XP
        rootnoverify (hd0,1)
        chainloader +1
```

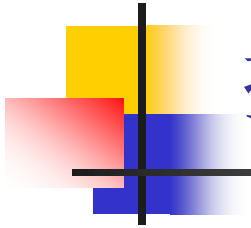
wqdai@hust.edu.cn

GRUB-IMA远程证明

- GRUB-IMA支持向远程用户证明平台可信性的功能。GRUB-IMA通过使用AIK对PCR签名，然后将度量值列表与平台绑定并以加密的形式发送给挑战者。



wqdai@hust.edu.cn



提纲

- Linux的启动过程
- TrustGrub
- Grub-IMA
- OSLO
- Tboot

wqdai@hust.edu.cn

静态可信度量

- 静态可信度量根是以基于TCG的BIOS为基础的，将下一个需要引导实体进行度量并将度量值存储到TPM中。虽然静态可信度量很容易实现，但是也存在一些无法回避的问题：
  - 实体（程序或者数据）是在加载时度量其完整性的，而不是在运行过程中度量。
  - 在引导过程的各个阶段，当某个阶段发生变化（例如升级或者打补丁）时，这将影响该阶段以及其后所有阶段的度量值。
  - 由于引导过程中，都是前一阶段度量后一阶段，严格地依赖于启动顺序。

wddai@hust.edu.cn

DRTM使用的PCR及其用途

- 与静态可信度量相比，动态可信度量使用了不同的PCR，用途也不同。

PCR	用途	复位
16	调试	1
17	Locality 4	1
18	Locality 3	1
19	Locality 2	1
20	Locality 1	1
21	T/OS指定	1
22	T/OS指定	1
23	应用程序指定	1

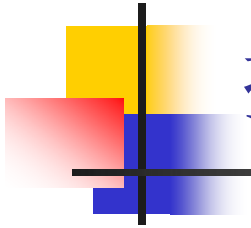
wqdai@hust.edu.cn

OSLO

- OSLO（Open Secure LOader）是基于AMD64处理器的支持动态可信度量根的开源引导加载器。它通过SKINIT指令来创建动态可信度量根，替代了基于BIOS的静态可信度量根和可信链。OSLO从多启动加载器开始作为内核的一部分，它初始化TPM，调用SKINIT指令，hash所有被加载的文件，并扩展到TPM的动态PCR中。OSLO通过TPM设备驱动来实现与TPM通信。

wqdai@ustc.edu.cn

PCR	用途
17	度量
19	Hash命令行



提纲

- Linux的启动过程
- TrustGrub
- Grub-IMA
- OSLO
- Tboot

wqdai@hust.edu.cn

Tboot

- Tboot（Trusted Boot）是一种开源的启动过程度量软件。其通过Intel TXT来度量和检查内核/虚拟机管理器（Virtual Machine Monitor）的启动过程。

wqdai@hust.edu.cn

- Tboot使用的PCR：

PCR	用途
17	度量MLE
18	Tboot及其策略控制值

Tboot（续）

- Tboot的主要功能：
  - 度量启动过程
  - 撤销度量的环境
  - 重置数据保护
  - 保护TXT内存范围
  - Intel TXT启动控制策略工具
  - 校验启动

wodai@hust.edu.cn

Tboot度量实例

- 下图说明了Tboot的启动配置文件（grub.conf）实例，Tboot最先启动，Linux内核和初始化RAM盘（initrd）以模块的形式加载。

```
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
#hiddenmenu

title Fedora / Intel TXT
        root (hd0,0)
        kernel /tboot.gz
        module /boot/vmlinuz-2.6.23.1-42.fc8 ro root=LABEL=/ rhgb quiet
        module /boot/initrd-2.6.23.1-42.fc8.img
        module /SINIT_17.BIN

title Windows XP
        rootnoverify (hd0,1)
        chainloader +1
```

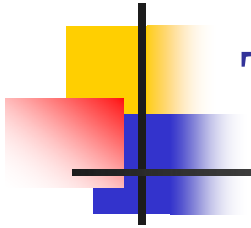
wqdai@hust.edu.cn

What is Trusted Boot (tboot)?

- Open source, pre-kernel/VMM module
- Uses Intel TXT to perform verified launch of OS kernel/VMM
  - Today only supports Xen
- Project also contains tools for policy creation and Provisioning
  - Intel TXT Launch Control Policy (LCP)
  - Tboot Verified Launch policy

wqdai@hust.edu.cn

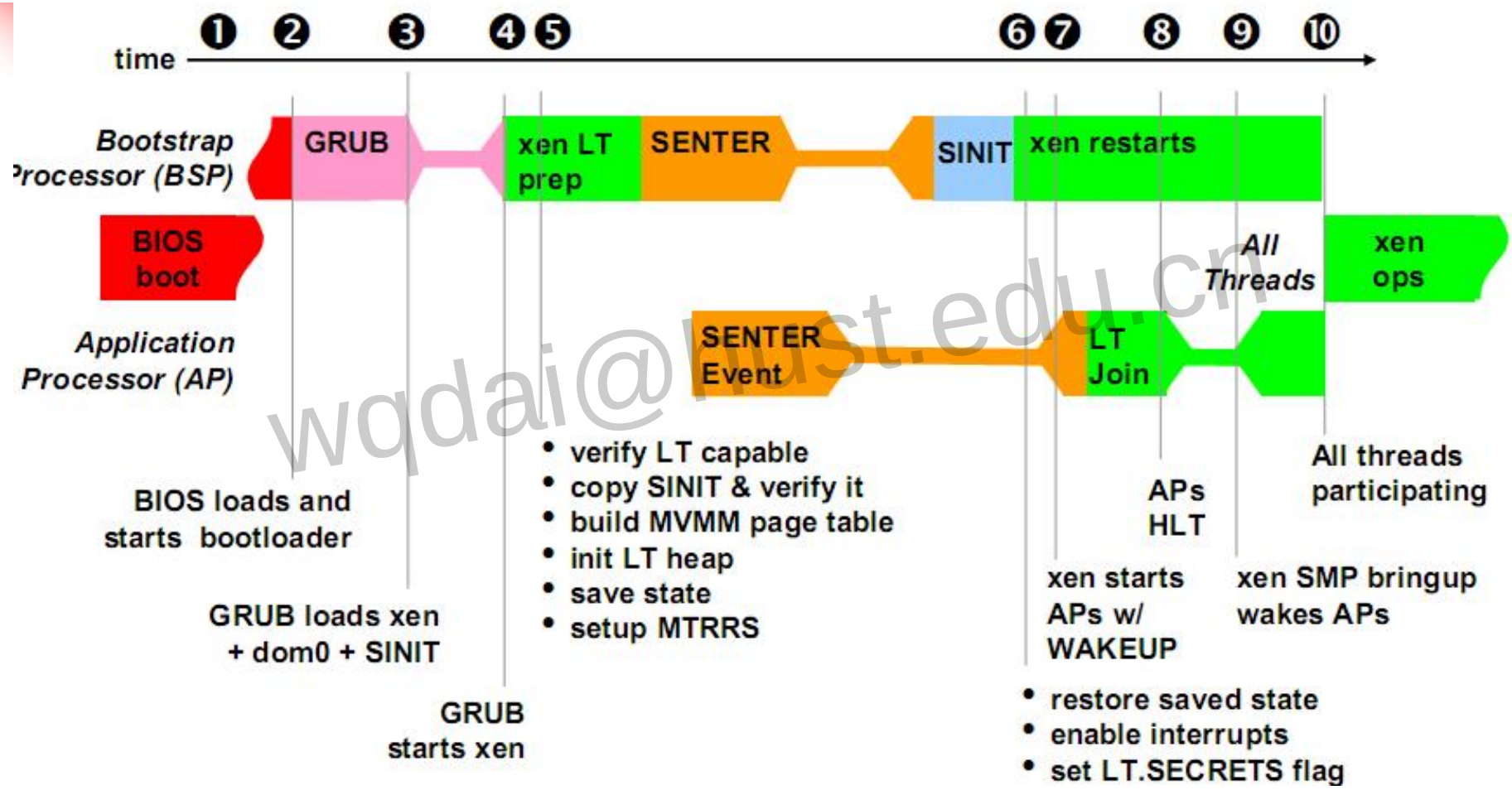
Trusted Boot provides a foundation for a Trusted Xen



- Root of trust is in hardware: Intel TXT dynamic launch
- Tboot is verified by Intel TXT Launch Control Policy (LCP)
  - Part of measured launch
- Tboot Verified Launch verifies Xen and Dom0 (+ initrd)

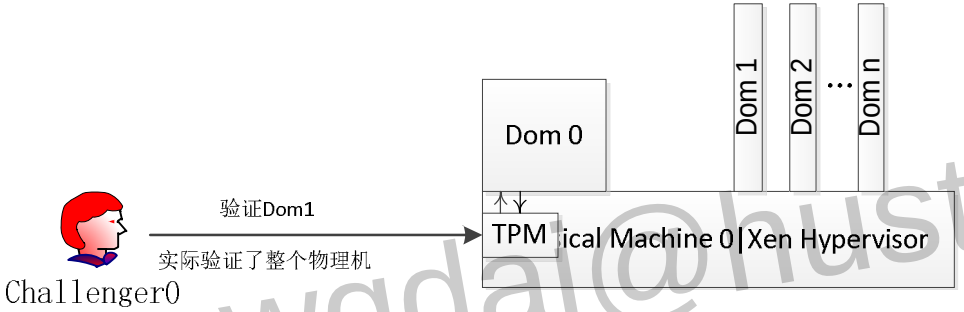
wqdai@hust.edu.cn

Tboot for Xen

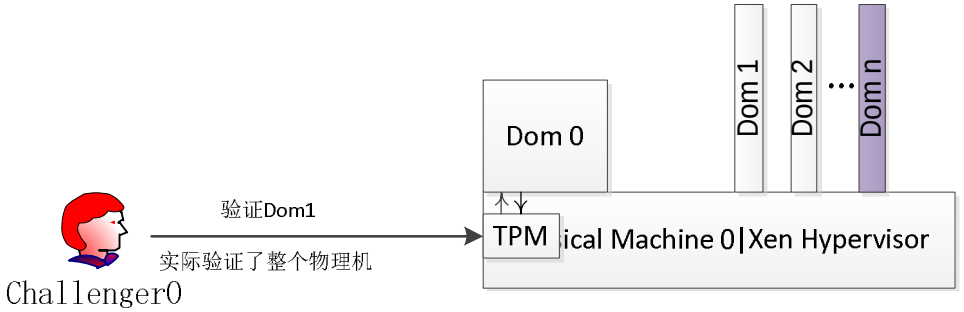


wqdai@nust.edu.cn

传统TPM下的虚拟机



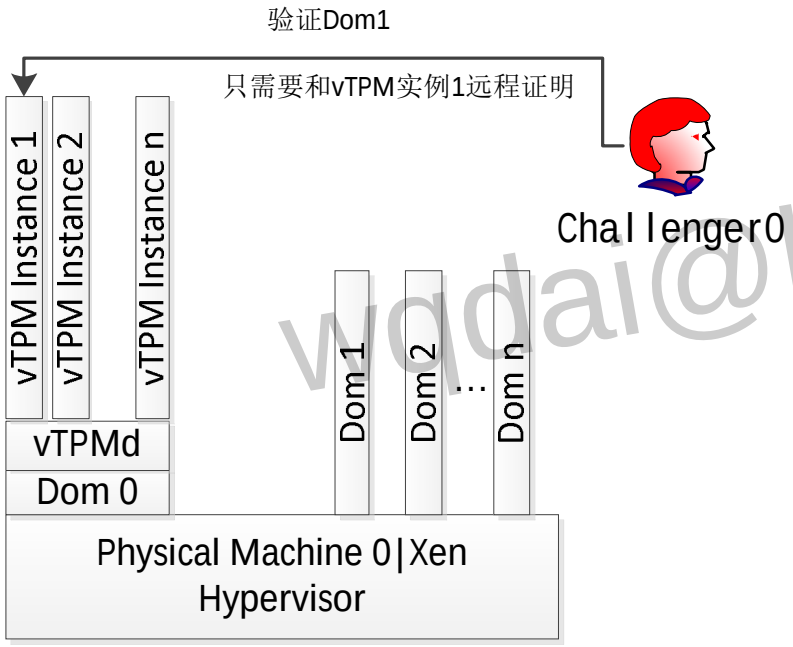
当一个远程验证这想验证Dom 1的安全状态，他将只能验证整个物理机。当这些虚拟机的状态都符合要求时，验证将会成功。



然而，如果一台虚拟机的状态不符合要求，如图中的Dom n，此次远程证明将失败。然而事实上，远程证明方想要验证的Dom1的状态是安全的。

wqdai@hust.edu.cn

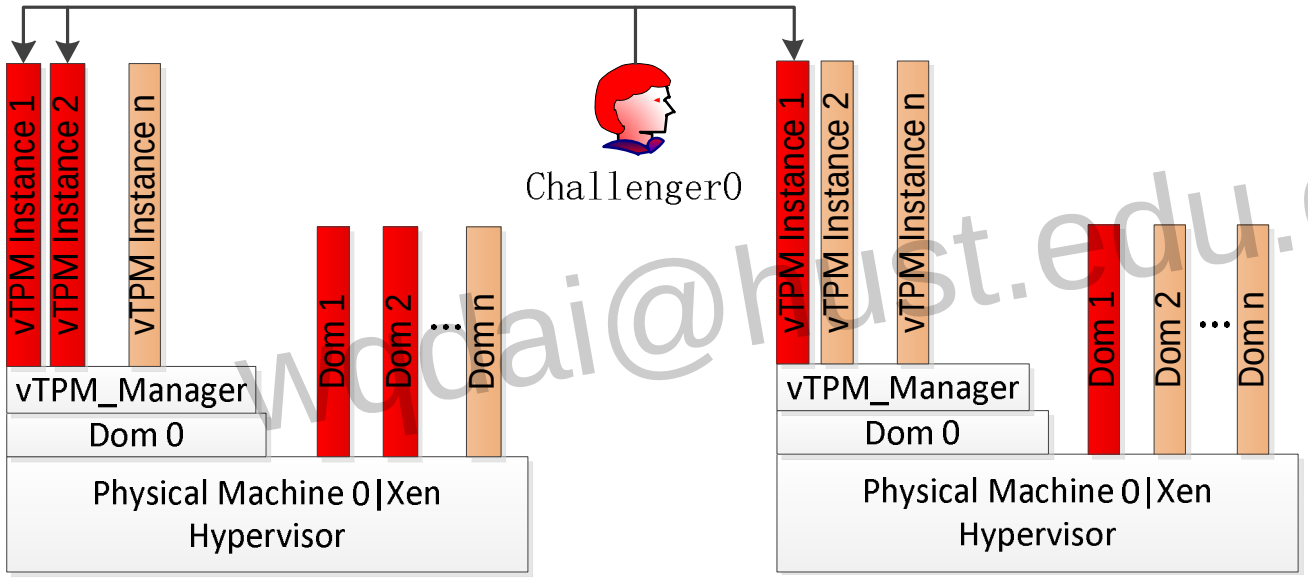
vTPM所带来的变化



在使用vTPM的系统中，每一个使用vTPM的虚拟机都有一个与其对应的vTPM实例。当远程用户要验证Dom 1是否处于安全状态时，只需要对vTPM 实例1进行远程证明即可。

wqdai@hust.edu.cn

vTPM下对域进行远程证明



如图所示，同样颜色的虚拟机表示属于同一个TVD。当远程用户希望验证红色TVD是否处于安全状态时，他只能分别对这个TVD中的三个虚拟机所属的vTPM实例提出远程证明。这个过程将非常耗时，也增加了相关应用的设计难度。

wqdai@hust.edu.cn