Now let’s briefly analyze the design of a trusted virtual machine monitor (TVMM) called Terra [131].

The novel ideas of this design are:

• The TVMM should support not only traditional operating systems, by exporting the hardware abstraction

for open-box platforms, but also the abstractions for closed-box platforms discussed in Section

9.5. Note that the VM abstraction for a closed-box platform does not allow the contents of the system

to be either manipulated or inspected by the platform owner.

• An application should be allowed to build its software stack based on its needs.Applications requiring

a very high level of security, e.g., financial applications and electronic voting systems, should run

under a very thin OS supporting only the functionality required by the application and the ability to

boot. At the other end of the spectrum are applications demanding low information assurance23 but

a rich set of OS features; such applications need a commodity operating system.

• Support additional capabilities to enhance system assurance:

• Provide trusted paths from a user to an application.We saw in Section 9.5 that such a path allows

a human user to determine with certainty the identity of the VM it is interacting with and, at the

same time, allows the VM to verify the identity of the human user.

• Support attestation, which is the ability of an application running in a closed box to gain trust

from a remote party by cryptographically identifying itself.

• Provide airtight isolation guarantees for the TVMM by denying the platform administrator root

access.

The management VM is selected by the owner of the platform but makes a distinction between a

platform owner and a platform user. The management VM formulates limits to the number of guest

VMs running on the platform, denies access to guest VMs that are deemed unsuitable to run, and grants

access to I/O devices to running VMs and limits their CPU, memory, and disk usage. Guest VMs expose

a raw hardware interface, including virtual network interfaces to virtual devices. The TVMM runs at the

highest privilege level and is secure even from the actions of the platform owner; it provides application

developers with the semantics of a closed-box platform.

A significant challenge to the security of a TVMM comes from the device drivers used by different

VMs running on the platform. Device drivers are large or very large software components, especially

the drivers for high-end wireless cards and video cards. There is also a large variety of such drivers,

many hastily written to accommodate new hardware features. Typically, the device drivers are the

lowest-quality software components found in the kernel of an operating system; thus, they pose the

highest security risks. To protect a TVMM, the device drivers should not be allowed to access sensitive

information and their memory access should be limited by different hardware protection mechanisms.

Malicious I/O devices can use different hardware capabilities, such as DMA, to modify the kernel.