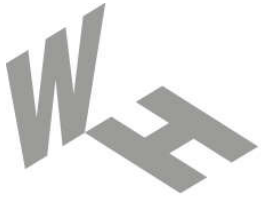


Westfälische Hochschule
Gelsenkirchen Bocholt Recklinghausen
University of Applied Sciences

Trusted Computing Base (TCB) / Trusted Computing Platform
→ Overview

Prof. Dr. (TU NN)
Norbert Pohlmann

Institute for Internet Security - if(is)
University of Applied Sciences Gelsenkirchen
http://www.internet-sicherheit.de

Content

- Aim and outcomes of this lecture
- Trusted Computing Base
- Operating System
- Virtualization
- Trusted Platform
- Summary

Content

- **Aim and outcomes of this lecture**
- Trusted Computing Base
- Operating System
- Virtualization
- Trusted Platform
- Summary

TCB / Trusted Computing Platform
→ Aims and outcomes of this lecture

Aims

- To introduce in the topic Trusted Computing Base (TCB) and Trusted Computing Platform (TCP)
- To explore the general idea of TCB and TCP
- To analyze the goals of TCB and TCP

At the end of this lecture you will be able to:

- Understand what the basic idea of TCB and TCP is.
- Know something about the approach to TCB and TCP.
- Understand the need of TCB and TCP.

Content

- Aim and outcomes of this lecture
- **Trusted Computing Base**
- Operating System
- Virtualization
- Trusted Platform
- Summary

Trusted Computing Bases
→ Idea

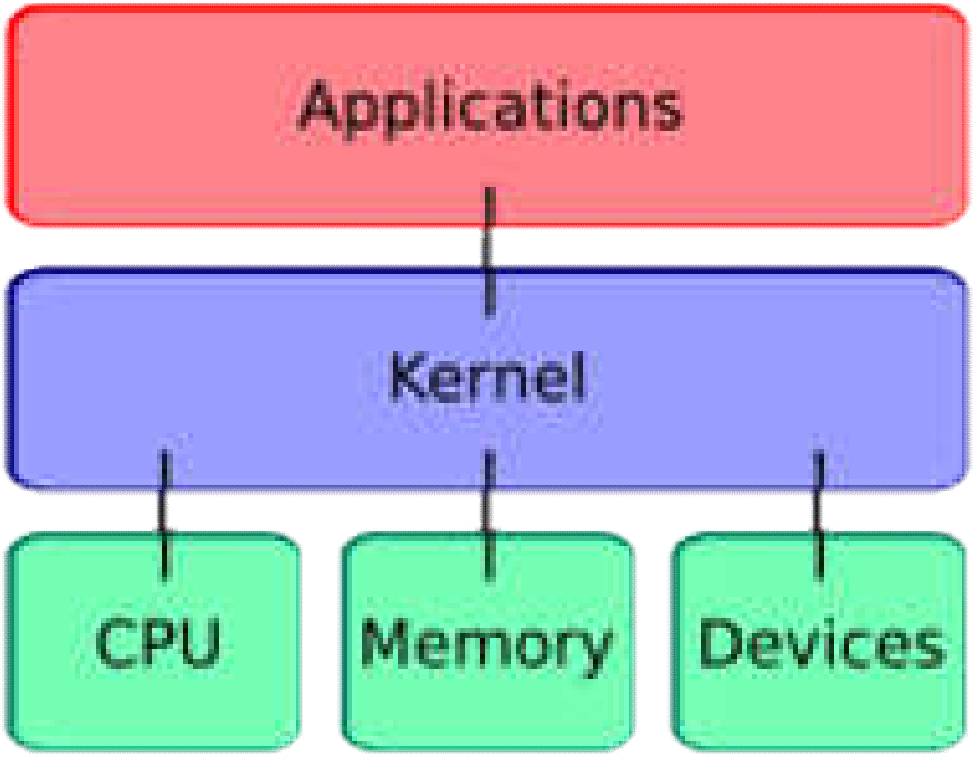
- The **Trusted Computing Base (TCB)** of a computer system is the set of all hardware, firmware, and/or software components that are critical to its security.
- In the sense that bugs occurring inside the TCB might threaten the security properties of the entire system.
- By contrast, parts of a computer system outside the TCB supposedly cannot misbehave in a way that would leak any more privileges than was granted to them in the first place in accordance to the security policy.
- The **careful design** and **implementation** of a system's trusted computing base (TCB) is paramount to its overall security.
- Modern operating systems strive to reduce the size of the TCB so that an exhaustive examination of its code base becomes feasible [5].
  - Manual or computer-assisted **software audit** or **program verification**
  - Size: < 100.000 line of code (normal OS 10.000.000 line of code)

Content

- Aim and outcomes of this lecture
- Trusted Computing Base
- **Operating System**
- Virtualization
- Trusted Platform
- Summary

Operating System
→ Kernel (1/2)

- **Kernel** is the central component of most computer operating systems (OS).
- Its responsibilities include managing the system's resources (the communication between hardware and software components).
- As a basic component of an operating system, a kernel provides the lowest-level abstraction layer for the resources (especially memory, processors and I/O devices) that application software must control to perform its function.
- It typically makes these facilities available to application processes through **inter-process communication** mechanisms and **system calls** [5].



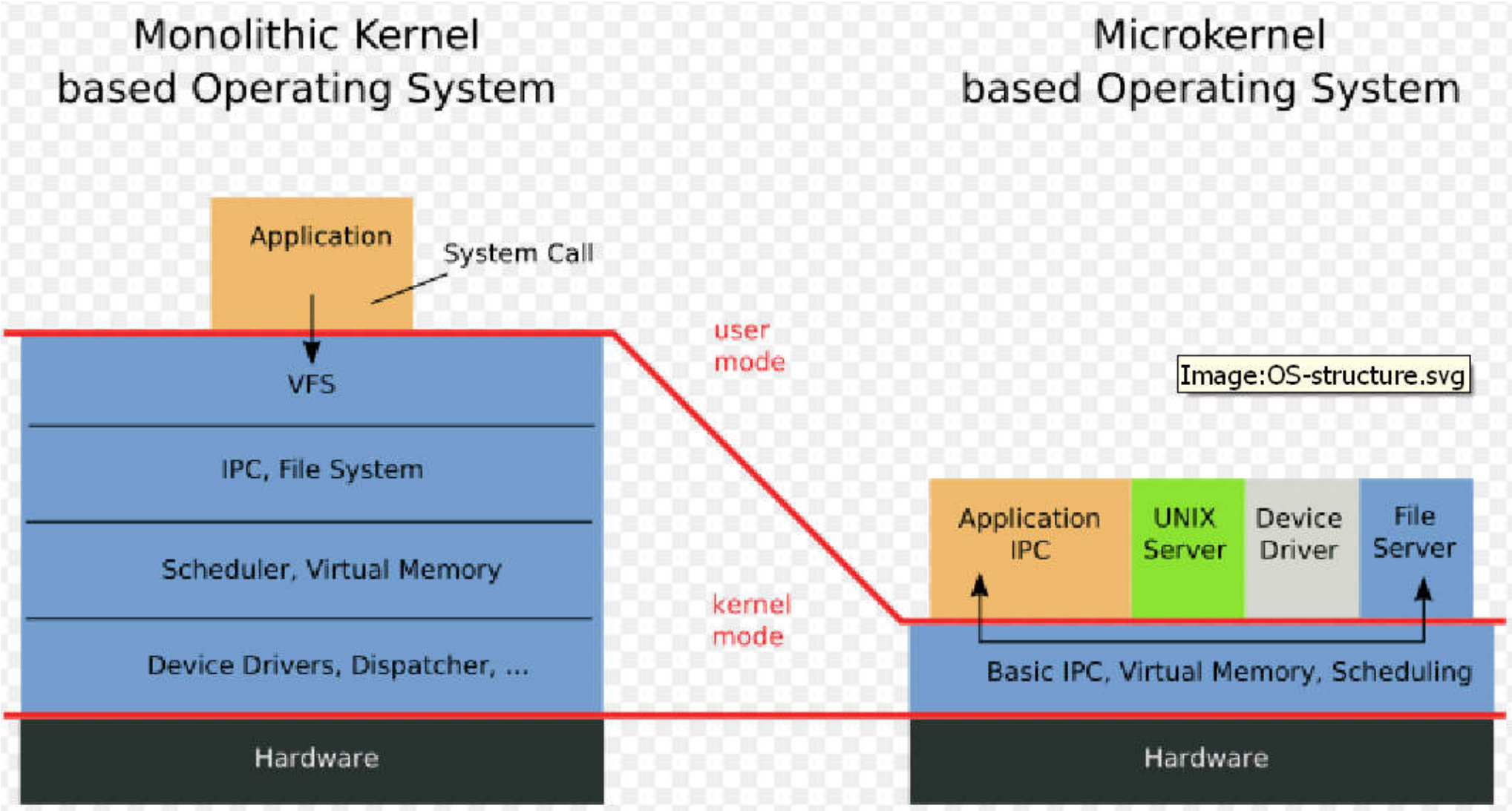
Operating System
→ Kernel (2/2)

- These tasks are done differently by different kernels, depending on their design and implementation.
- **Monolithic kernels** will try to achieve these goals by executing all the code in the same address space to **increase the performance** of the system.
- **Microkernels** run most of their services in user space, aiming to improve **maintainability** and **modularity** of the codebase (higher trustworthiness).
- A range of possibilities exists between these two extremes [5].

Operating System
→ Monolithic Kernel

- A **monolithic kernel** is a kernel architecture where the entire kernel is run in kernel space in supervisor mode.
- In common with other architectures (microkernel, hybrid kernels), the kernel defines a high-level virtual interface over computer hardware.
- With a set of primitives or system calls to implement operating system services such as process management, concurrency, and memory management in one or more modules.
- Even if every module servicing these operations is separate from the whole, the code integration is very tight and difficult to do correctly, and, since **all the modules run in the same address space**, a bug in one module can bring down the whole system.
- However, when the implementation is complete and trustworthy, the tight internal integration of components allows the low-level features of the underlying system to be effectively utilized, making a good monolithic kernel **highly efficient**.
- In a monolithic kernel, all the systems such as the file system management run in an area called the kernel mode [5].

Operating System
→ Overview Kernel (2/2)



Operating System
→ Microkernel (1/3)

- A **microkernel** is a minimal computer operating system kernel which, in its purest form, provides no operating-system services at all, only the mechanisms needed to implement such services, such as low-level address space management, thread management, and inter-process communication (IPC).
- If the hardware has a kernelmode-usermode distinction, the microkernel is the only part of the system executing in a kernel mode.
- The actual operating-system services are provided by "user-mode" servers.
- These include device drivers, protocol stacks, file systems and user-interface code.
- This results in a system structure that is drastically different from the monolithic kernels of the mass market.
- The **monolithic kernels** have a **vertically-layered structure**, where applications obtain services by performing a specific system call for each service [5].

Operating System
→ Microkernel (2/3)

- In contrast, a **microkernel-based** system features a **horizontal structure**, where system services are obtained by executing an IPC system call addressed to a particular server.
- They also have much in common with **hypervisors**, but the latter make no claim to minimality, and are specialized to supporting virtual machines.
- In computing, a hypervisor, also called **virtual machine monitor**, is a virtualization platform that allows multiple operating systems to run on a host computer at the same time.
- The **L4 microkernel** is frequently used as a hypervisor, which indicates that a microkernel is a possible implementation of a hypervisor.
- The term nanokernel is historically used to differentiate from earlier microkernels which contained actual system services, but the minimallity principle used by Jochen Liedtke in the design of the L4 microkernel implies that these terms have the same meaning; microkernel is the modern terminology [5].

Operating System
→ Microkernel (3/3)

- **Pros and Cons of a Microkernel:**
  - **Pros**
    - Higher robustness
    - Higher modularity
    - Higher flexibility
    - Higher IT security
      … because of more inter-process communication, …
    - Less needed storage
    - …
  - **Cons**
    - Less performance, because of more inter-process communication

Content

- Aim and outcomes of this lecture
- Trusted Computing Base
- Operating System
- **Virtualization**
- Trusted Platform
- Summary

Trusted Computing Bases
→ Virtualization

- **In general, virtualization is a broad term that refers to the abstraction of computer resources.**
- **Platform virtualization**, which separates an operating system from the underlying platform resources.
  - **Full Virtualization**
    In full virtualization, the virtual machine simulates enough hardware to allow an unmodified "guest" OS (one designed for the same instruction set) to be run in isolation.
  - **Hardware-assisted Virtualization**
    In hardware-assisted virtualization, the hardware provides architectural support that facilitates building a virtual machine monitor and allows guest OSes to be run in isolation. In 2005 and 2006, Intel and AMD provided additional hardware to support virtualization.
  - **Paravirtualization**
    In paravirtualization, the virtual machine does not necessarily simulate hardware, but instead (or in addition) offers a special API that can only be used by modifying the "guest" OS [5].
  - …

Content

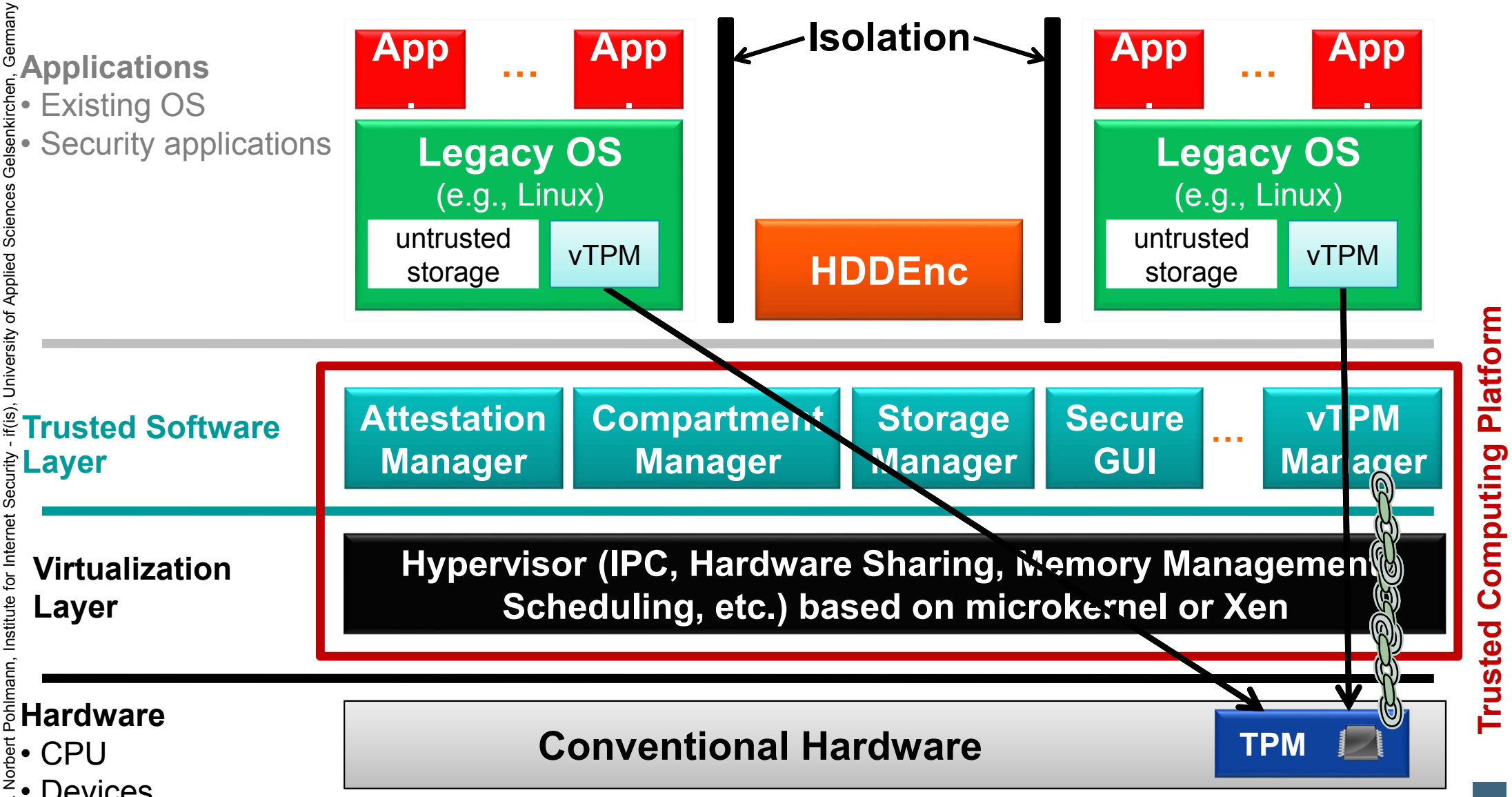
- Aim and outcomes of this lecture
- Trusted Computing Base
- Operating System
- Virtualization
- **Trusted Platform**
- Summary

Trusted Platform
→ Motivation

- **Hardware virtualization** a (re-discovered) useful means to reduce to total cost
  - Apparent in corporate data centers
  - However, workloads should be processed separately due to diversity of security objectives involved parties

- Combine **hypervisors** (Virtual Machine Monitors) with hardware-based **root of trust (Trusted Platform)**
  - **Hypervisors** provide isolations of workloads
    - mediating access to physical resources by virtual machines
  - Hardware **root of trust** is resistant to software attacks and provides a basis for reasoning about the integrity of SW ruing on a (trusted) platform

Trusted Platform
→ Possible Architecture



Trusted Platform
→ Components (1/2)

- **TC enabled hardware**
  - Ensuring authentication of the access requestor and the

- **Trusted Software Layer**

  - ***Trust Manager:*** controls access to TPM interface

  - ***Compartment Manager***:
    - Manages creation, updates, and deletion of compartments
    - Measures compartments and assigns unique IDs to them

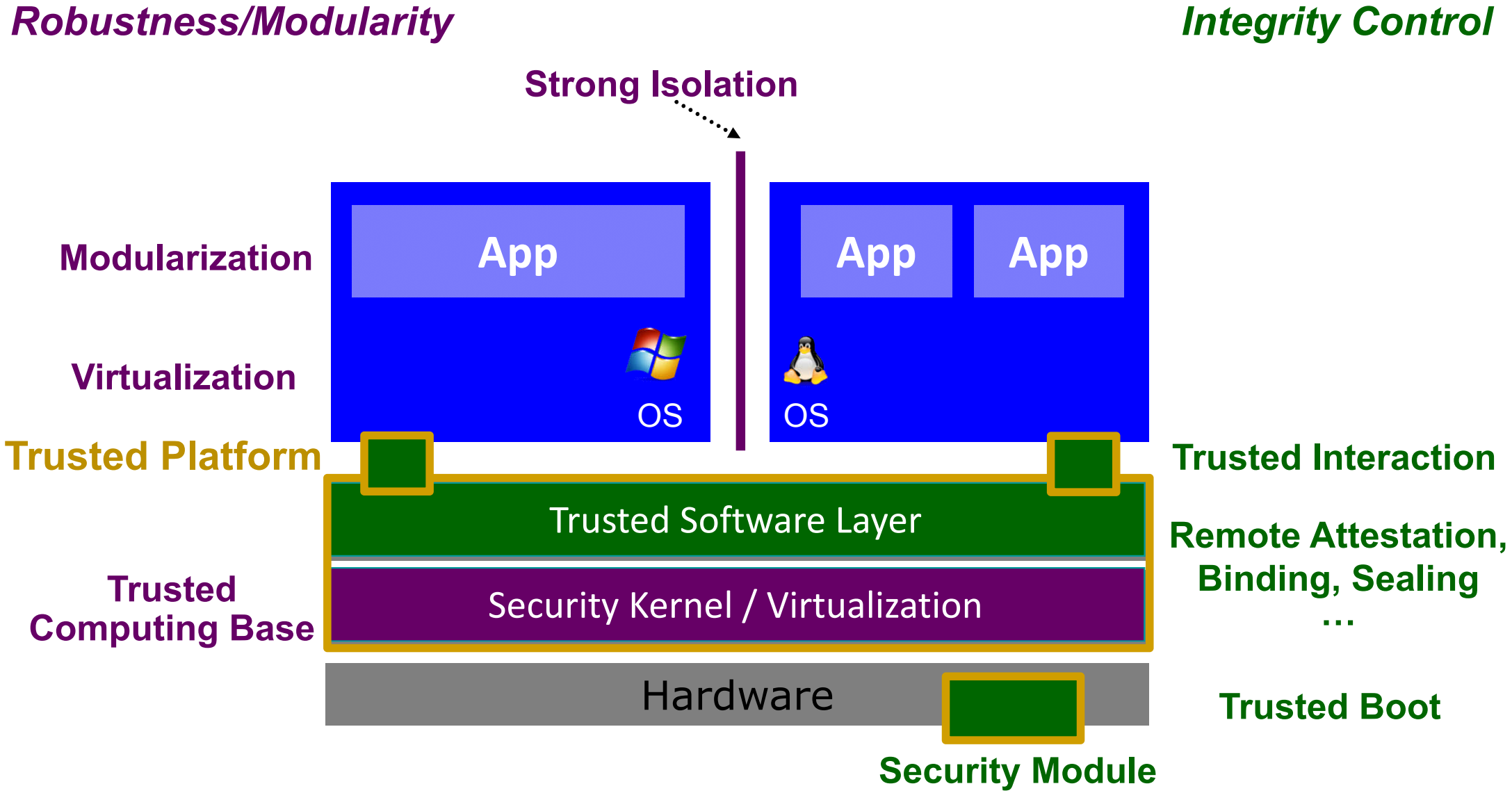
  - ***Storage manager***
    - Guarantees trusted storage, i.e., authenticity, confidentiality and integrity (and freshness) of stored data
    - Has access to configuration of clients it is communicating to over trusted channel

  - ***Secure GUI:*** guarantee a trusted path to application

Trusted Platform
→ Components (2/2)

- **Virtualization Layer**
  - Provides abstraction of physical machine
  - Provides isolation between virtual machines

Paradigm Shift
→ Trusted Platform



Content

- Aim and outcomes of this lecture
- Trusted Computing Base
- Operating System
- Virtualization
- Trusted Platform
- **Summary**

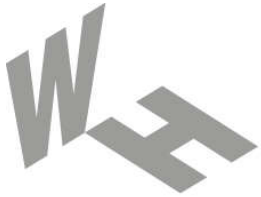
TCB / Trusted Computing Platform
→ Summary

- Trusted Computing Platform is an approach to bring the ideas of
  - **Trusted Computing (root of trust - TCG)** and
  - **Trusted Computing Base (TCB)**

  together.

- Increase the level of trustworthiness.

- It is a very good basic for modern application.



Westfälische Hochschule
Gelsenkirchen Bocholt Recklinghausen
University of Applied Sciences

Trusted Computing Base (TCB) / Trusted Computing Platform → Overview

Thank you for your attention!
Questions?

Prof. Dr. (TU NN)
Norbert Pohlmann

Institute for Internet Security - if(is)
University of Applied Sciences Gelsenkirchen
http://www.internet-sicherheit.de

if(is) internet security.

TCB / Trusted Computing Platform
→ Literature

- [1] **Prof. Dr.-Ing. Ahmad Reza Sadeghi**
  http://www.trust.rub.de/home/
- [2] N. Pohlmann, A.-R. Sadeghi, C. Stüble: "European Multilateral Secure Computing Base", DuD Datenschutz und Datensicherheit – Recht und Sicherheit in Informationsverarbeitung und Kommunikation, Vieweg Verlag, 09/2004
- [3] N. Pohlmann, H. Reimer: „Trusted Computing – eine Einführung", in "Trusted Computing - Ein Weg zu neuen IT-Sicherheitsarchitekturen", Hrsg.: N. Pohlmann, H. Reimer; Vieweg-Verlag, Wiesbaden 2008
- [4] M. Linnemann, N. Pohlmann: "An Airbag for the Operating System – A Pipedream?", ENISA Quarterly Vol. 3, No. 3, July-Sept 2007
- [5] http://en.wikipedia.org/wiki/Main_Page

Links:

Institute for Internet Security:
http://www.internet-sicherheit.de/forschung/aktuelle-projekte/trusted-computing/