



BSI Sicherheitskongress 2005

sHype: Hypervisor Security Architecture

Reiner Sailer
Secure Systems Department
IBM T. J. Watson Research Center, NY

Gliederung

- Sicherheitsproblem
- Wie hilft Virtualisierungs-Infrastruktur ?
- Neues (einfacher lösbares) Sicherheitsproblem
- Eine Lösung: sHype Sicherheitsarchitektur

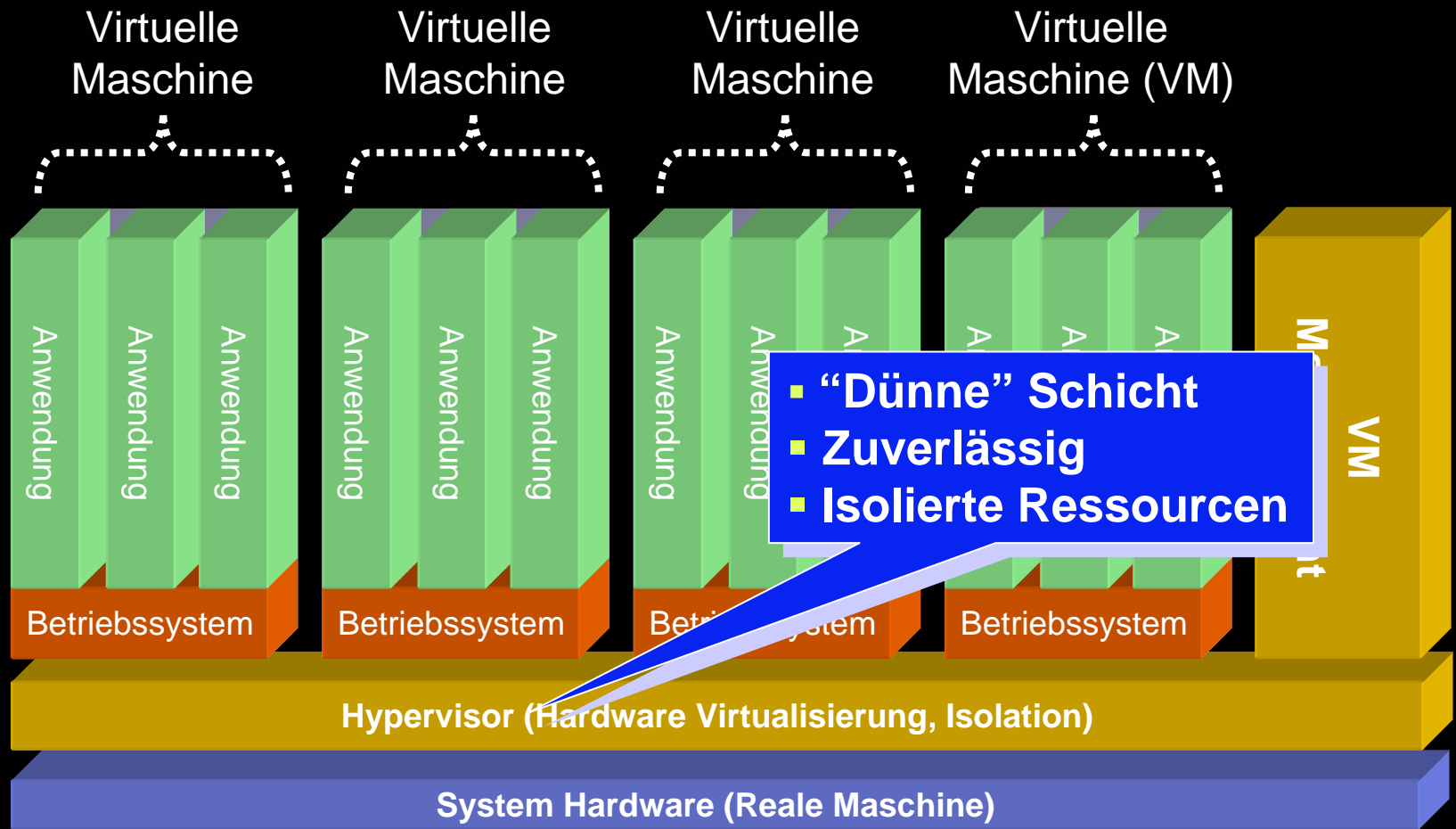
Sicherheitsproblem

- **Benutzerfreundlichkeit**, Komfort, Funktionsvielfalt, viele Einstellungen
 - Hohe Software-Komplexität
 - Hoch-dynamisch (neue „Features“)
- **Sicherheitsgarantien**
 - Niedrige Software-Komplexität (Inspektion, Fehler = $x/1000$ LOC)
 - Langlebig (einfach nachführbar)

Lösung: Unterschiedlich sensitive Workloads laufen auf demselben Betriebssystem sicher isoliert voneinander ab.

Problem: Erfordert sicheres funktionsreiches Betriebssystem (ungelöstes Problem).

Virtualisierung: Der „Virtual Machine Monitor“ (VMM)



Warum Virtualisierung ?

Virtualisierung spart Geld

Server-Bereich:

- **Consolidation:** bessere Auslastung von Systemen, 20% → >80%
- **Provisioning/Workload Balancing:** automatisiert Management und Verteilung von Nutzlasten

Client-Bereich:

- **Consolidation:** erlaubt **Ablaufen mehrerer Betriebssysteme** auf einer einzigen System-Hardware
 - unterschiedlich sensitive Workloads laufen isoliert in unterschiedlichen Betriebssystemen ab

Neues (einfacher lösbares) Sicherheitsproblem

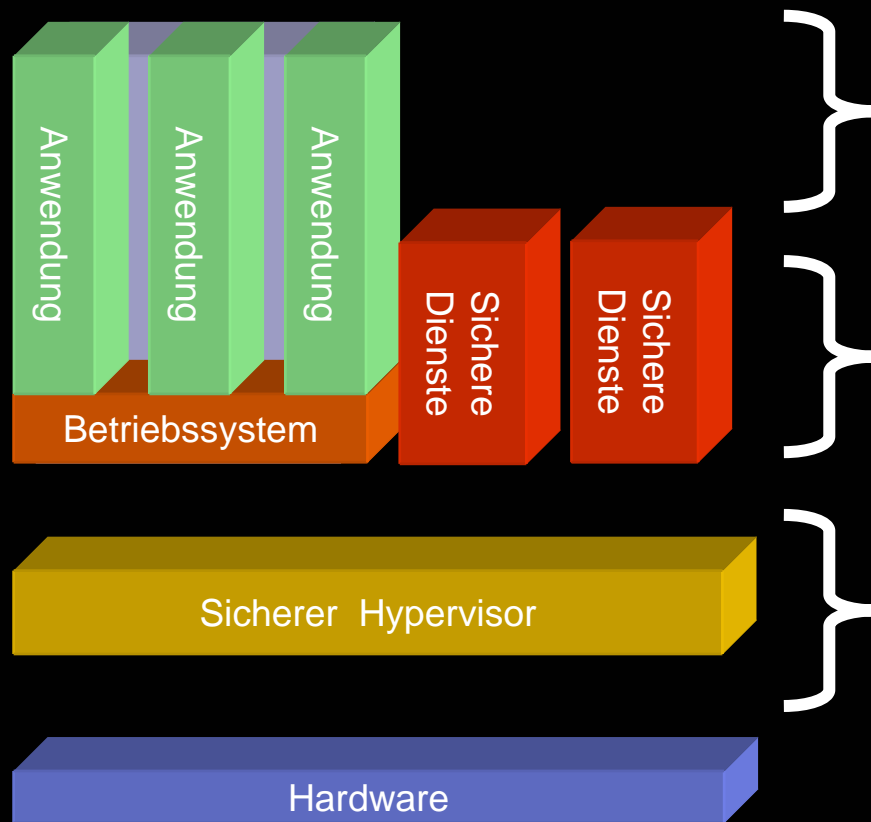
Einfacher lösbares Sicherheitsproblem (und Kern dieses Vortrags):

Wie verhindern wir Auswirkungen von Einbrüchen und Missverhalten in einem Betriebssystem auf andere lokal ablaufende Betriebssysteme?

Gegeben: Isolation von virtuellen Ressourcen gegeneinander:

- lokale Betriebssysteme können sich nur beeinflussen, wenn sie Ressourcen gemeinsam Nutzen (Informationsfluss)
- **sHype kontrolliert das gemeinsame Nutzen von Ressourcen zwischen Betriebssystemen (VMs) auf Hypervisor-Ebene**

sHype – Ein Geschichteter Sicherheitsansatz



Policy - Durchsetzung

- Middleware / Anwendung
→ z.B. Web Services Policy
- Betriebssystem: Intra-VM Policy
→ z.B. SELinux, Type Enforcement, Role-Based Access Control
- sHype: Inter-VM Policy
→ z.B., Chinese Wall, TE, (MLS)

Gliederung

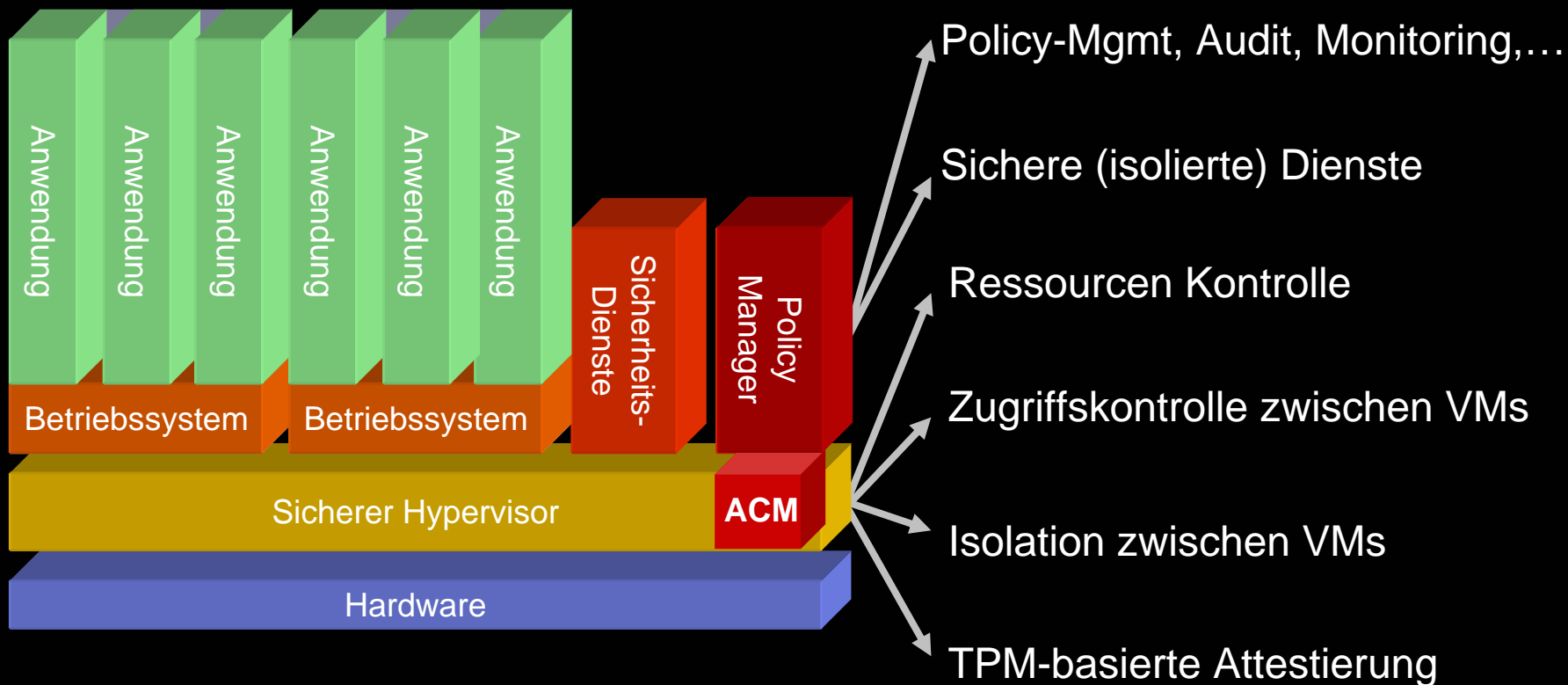
- Sicherheitsproblem
- Wie hilft Virtualisierungs-Infrastruktur ?
- Neues (einfacher lösbares) Sicherheitsproblem
- Eine Lösung: **sHype** Sicherheitsarchitektur
schafft nachhaltig abgrenzbare Laufzeit-Umgebungen

sHype: Lektionen aus der Vergangenheit

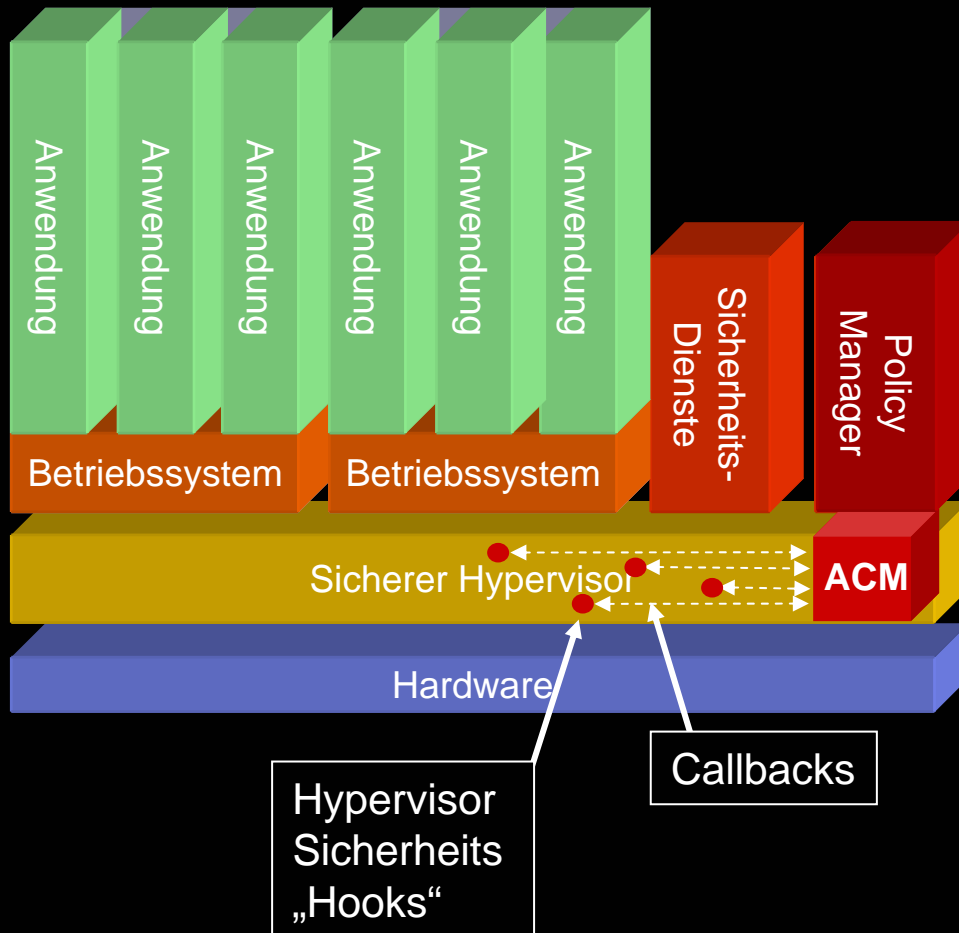
Kernalized VM/370 (Research Project) / Secure VAX/VMM (Research Project)

- **KVM/370** (1979, Performance 10-50%) [Retrofit: 50% neuer Code]
- **VAX/VMM** (Plan '82, Booten '84, Stabil '89, Performance 30-90%) [neuer code]
- **Lektionen:**
 - (i) **Leistung ist der entscheidende Faktor für breite Einsetzbarkeit**
 - (ii) **Minimale Code-Schnittstellen zwischen Sicherheits- und Hypervisor-Code erhöhen die nachhaltige Einbringung von Sicherheit.**
- **sHype zielt auf mittlere Assurance** unter Berücksichtigung von (i) + (ii) ab, während historische Ansätze auf höchste Assurance abzielten
- Sicherheit ist effektiv gegen Angriffe, wenn sie tatsächlich eingesetzt wird
 - breite Verfügbarkeit in Basis-Technologie (Einbringung + Pflege)
 - COTS Sicherheitstechnologie („Mainstream“)

Sichere Hypervisor Funktionen I

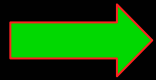


Sichere Hypervisor Funktionen II



- **Flexible Architektur**
Unterstützt unterschiedliche Sicherheits-Policies
- **Access Control Module (ACM)**
Fällt Entscheidungen
- **Hypervisor Instrumentierung mit Callbacks in das ACM**
Kontrolliert inter-VM Ressourcen
Setzt Entscheidungen des ACM durch
- **Unterstützte Plattformen**
Xen/OpenHype, PHYP

sHype – Architektur-Bausteine

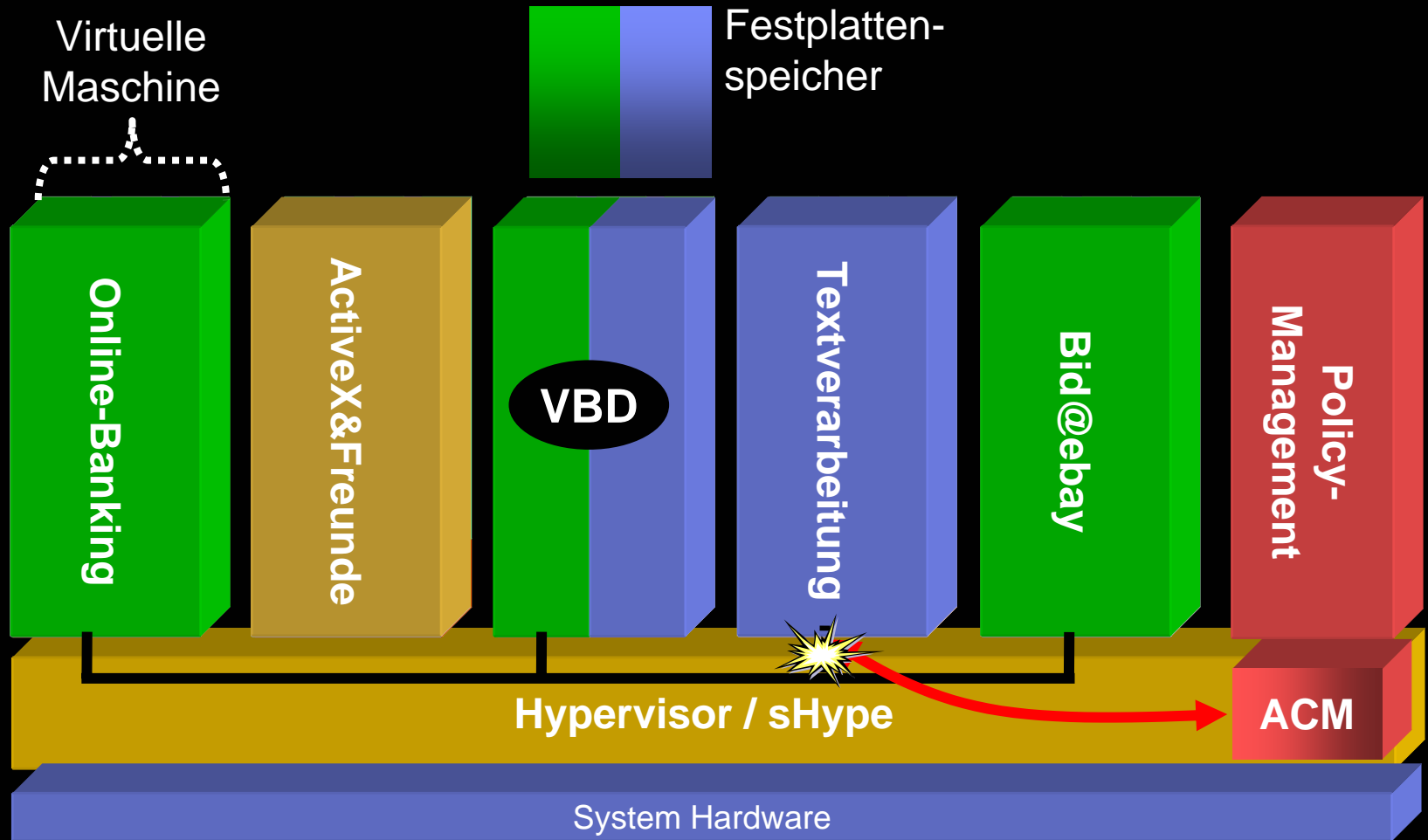


- **Sicherheits-Policies** angepasst an den Bedarf auf VM-Ebene
 - **Chinese Wall Policy:**
Autorisierung des Startens von Betriebssystemen (Workloads)
 - **Type Enforcement Policy:**
Autorisierung der Ressourcen-Teilung zwischen Betriebssystemen

- **Sicherheits-Management**

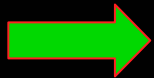
- **Sicherheits-Durchsetzung**

TypeEnforcement – Kontrolliert Nutzung gemeinsamer Ressourcen



sHype – Architektur-Bausteine

- Sicherheits-Policies

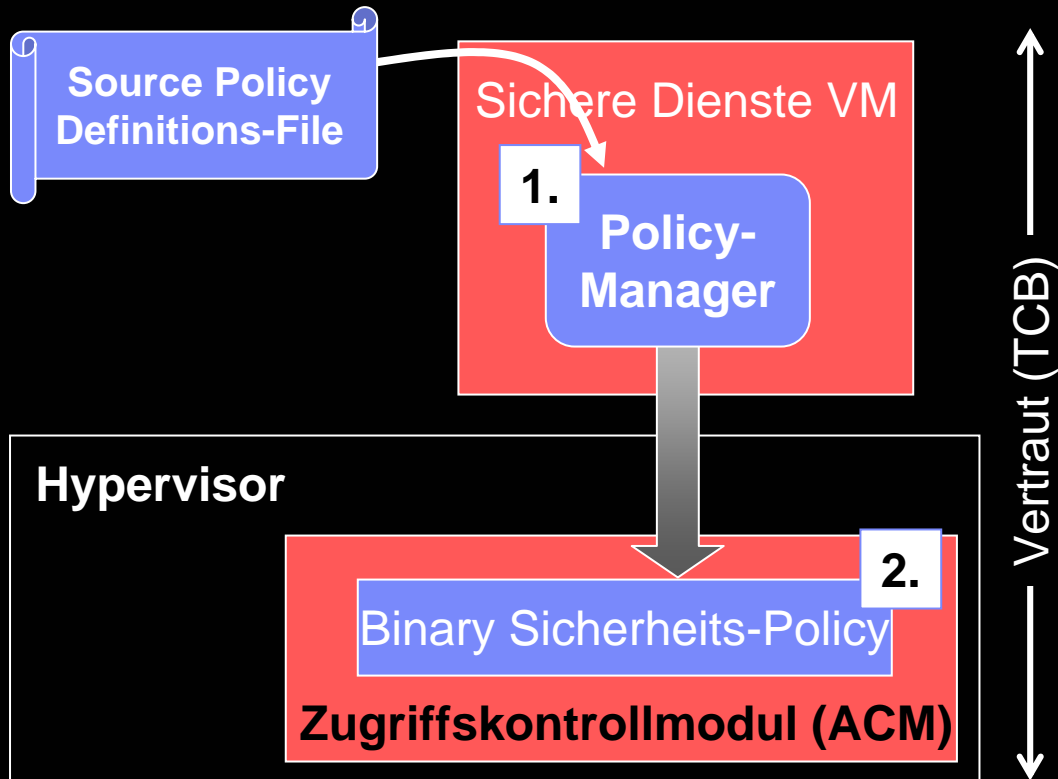


- **Sicherheits-Management**

Nahtlose Integration in existierende Virtualisierungsinfrastruktur

- Sicherheits-Durchsetzung

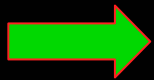
Etablierung einer Sicherheits-Policy



- 1. Policy-Management außerhalb des Hypervisors minimiert Sicherheits-Code im Hypervisor (P-Mgmt nicht notwendig zum Betrieb)**
- 2. Binäre Policy im Hypervisor vereinfacht Zugriffskontrolle im Hypervisor**

sHype – Architektur-Bausteine

- Sicherheits-Policies
- Sicherheits-Management

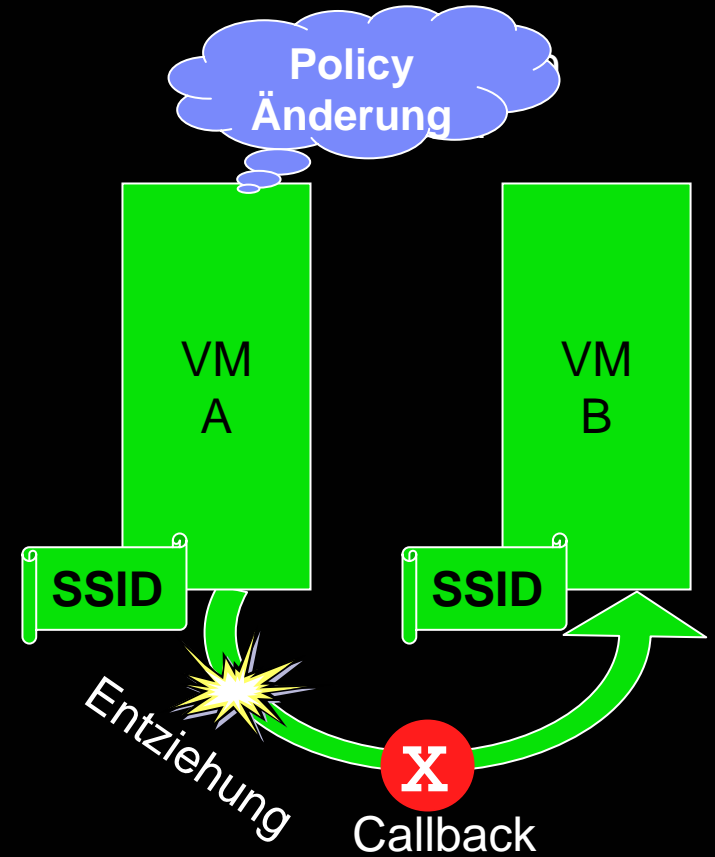
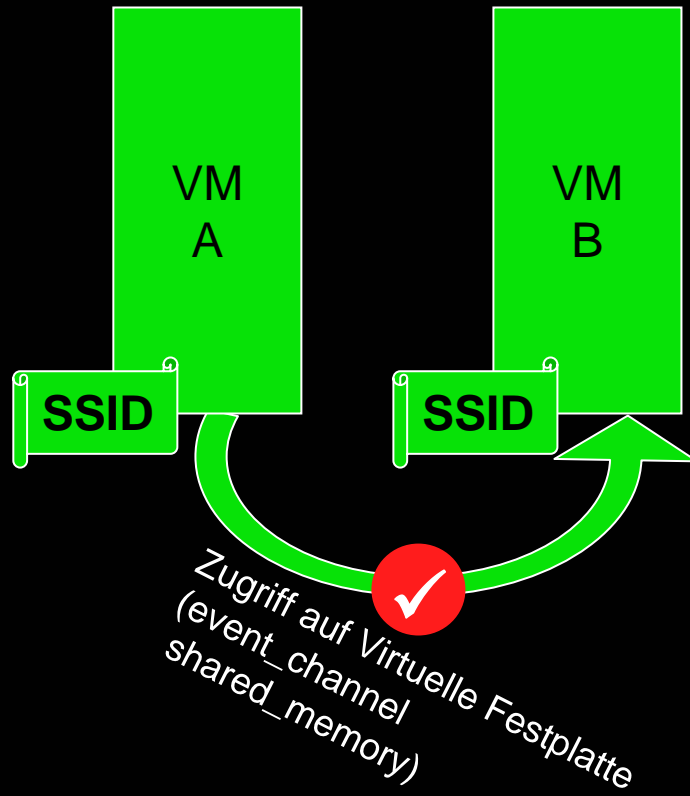


- **Sicherheits-Durchsetzung**

**Autorisierung während des ersten Zugriffs auf Ressourcen
(„Bind-time Authorization“)**

Autorisierung während des Bindens von Ressourcen (TE)

Gemeinsame Ressourcen



... sHype Kontrolle

sHype: „Bind-time Authorization“

✓ Vorteile

- + grob-granulare, einfache, Maschinen-unabhängige Policies
- + minimale Code-Schnittstelle (in nicht-optimierten Code-Teilen)
- + vernachlässigbare Leistungsverminderung (<1%)

✗ Nachteile

- Entzug von Ressourcen kann komplex sein (Shared Memory)

Implementierung

- sHype ist integraler Bestandteil der Virtualisierung, nicht optional
- Größe: 3K Lines Of Code
- Policies: Null (Default), Chinese Wall, und Type Enforcement Policy
- Verfügbarkeit: Xen Mailing Liste
(xen-devel@lists.xensource.com)

Zusammenfassung

sHype integriert sich nahtlos in die Hardware-Virtualisierung

- Ideal für die Abgrenzung von lokalen Betriebssystemen (inter-VM)
- „Safety-Net“ für Sicherheitsprobleme in Betriebssystemen

Gegenwärtige Arbeit im Zugriffskontroll-Bereich

- sHype-Unterstützung mehrerer realer Systeme (Cluster, Server Farm)
- Erhaltung der Sicherheitsgarantien während der Migration von VMs

Weitere Informationen

- http://www.research.ibm.com/secure_systems_department
- sailer@us.ibm.com