

# Management vernetzter IT-Systeme

## Kapitel: 10

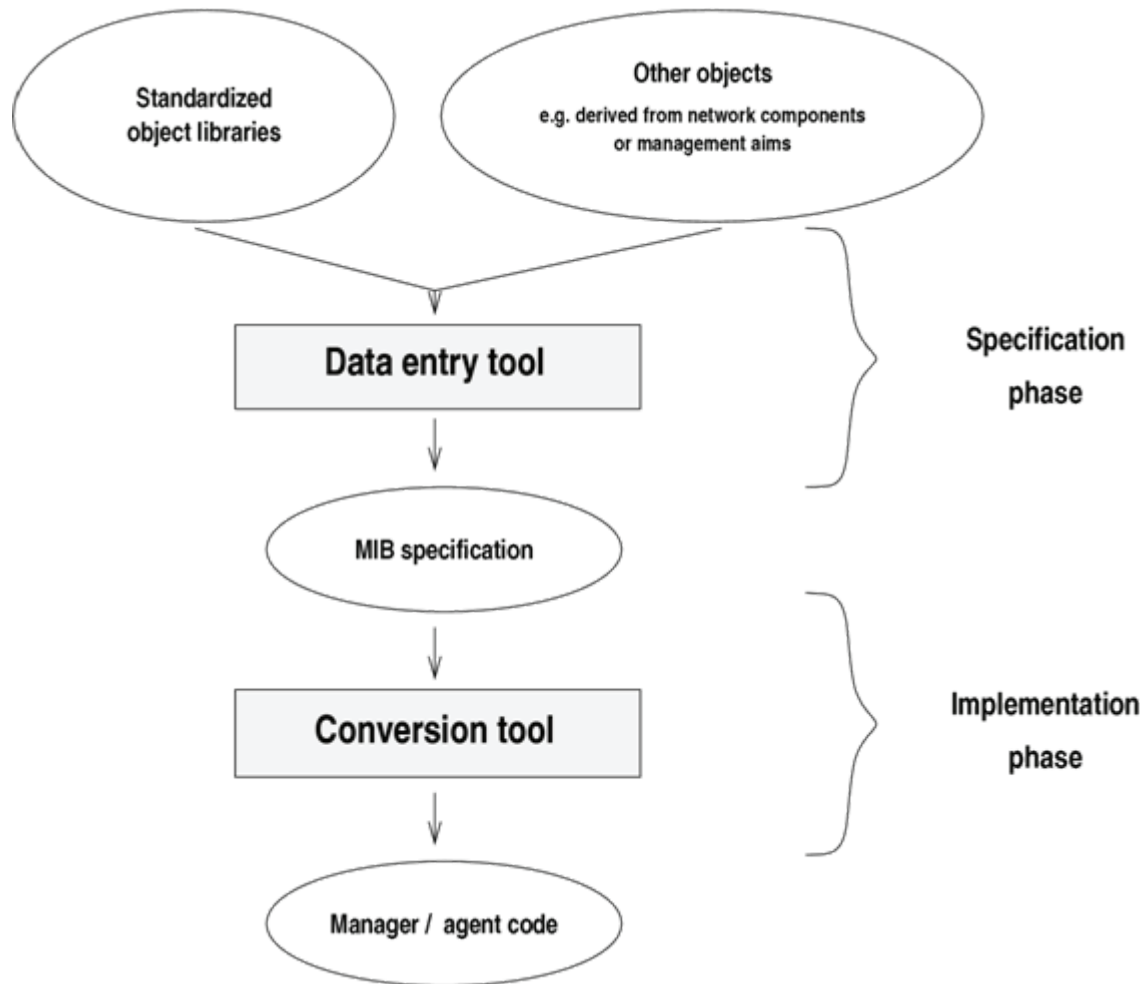
### Integration von Agenten, Architekturübergänge

# Entwicklungswerkzeuge (1)

## □ MIB-Werkzeuge

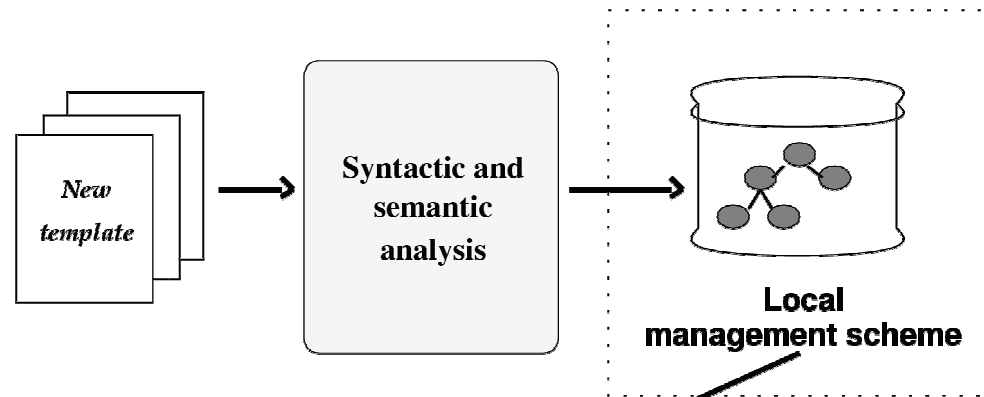
- GDMO/IDL Werkzeuge (Syntaxprüfungen, Editieren von Templates, Darstellung von Hierarchien, Navigieren, Suchen)
- Modellierungswerkzeuge für UML/CIM oder OMT (Rational Rose, StP)
- MIB Compiler, IDL Compiler, Template Compiler (Konversion von Objektspezifikationen in ausführbaren Manager- oder Agentencode)
- MIB- /Repository Administration, Klassenbibliotheken

# Development Tools for Management

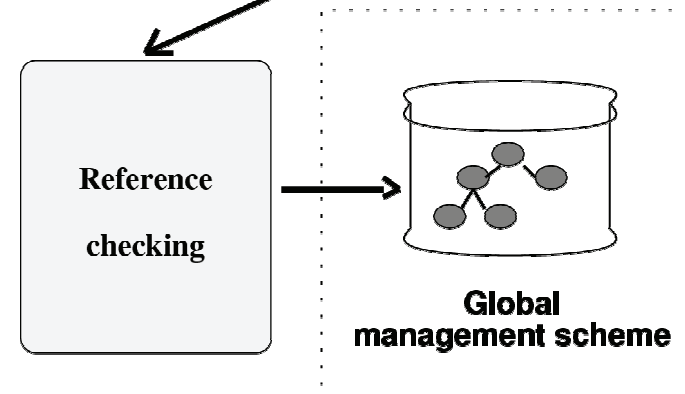


# Development Tools for Management

**Phase 1**  
*Data entry and syntactic/semantic analysis*



**Phase 2**  
*Reference checking*



# Class Browser Damocles

Expand Definition

File Help

Definition Name

Property Types  Attributes  Actions  Notifications  Behaviour

Expand Scope  Included  Inherited  Sorted

Property	Type	In Package	In Class
administrativeState	ATTR	"ISO/IEC 10165-2":logPackage	"ISO/IEC 10165-2":log
availabilityStatus	ATTR	"ISO/IEC 10165-2":availabilityStatusPackage	"ISO/IEC 10165-2":log
availabilityStatus	ATTR	"ISO/IEC 10165-2":logPackage	"ISO/IEC 10165-2":log
capacityAlarmThreshold	ATTR	"ISO/IEC 10165-2":logAlarmPackage	"ISO/IEC 10165-2":log
currentLogSize	ATTR	"ISO/IEC 10165-2":finiteLogSizePackage	"ISO/IEC 10165-2":log
discriminatorConstruct	ATTR	"ISO/IEC 10165-2":logPackage	"ISO/IEC 10165-2":log
intervalsOfDay	ATTR	"ISO/IEC 10165-2":dailyScheduling	"ISO/IEC 10165-2":log
logFullAction	ATTR	"ISO/IEC 10165-2":logPackage	"ISO/IEC 10165-2":log
logId	ATTR	"ISO/IEC 10165-2":logPackage	"ISO/IEC 10165-2":log

Apply Close

# Development Tools for Management

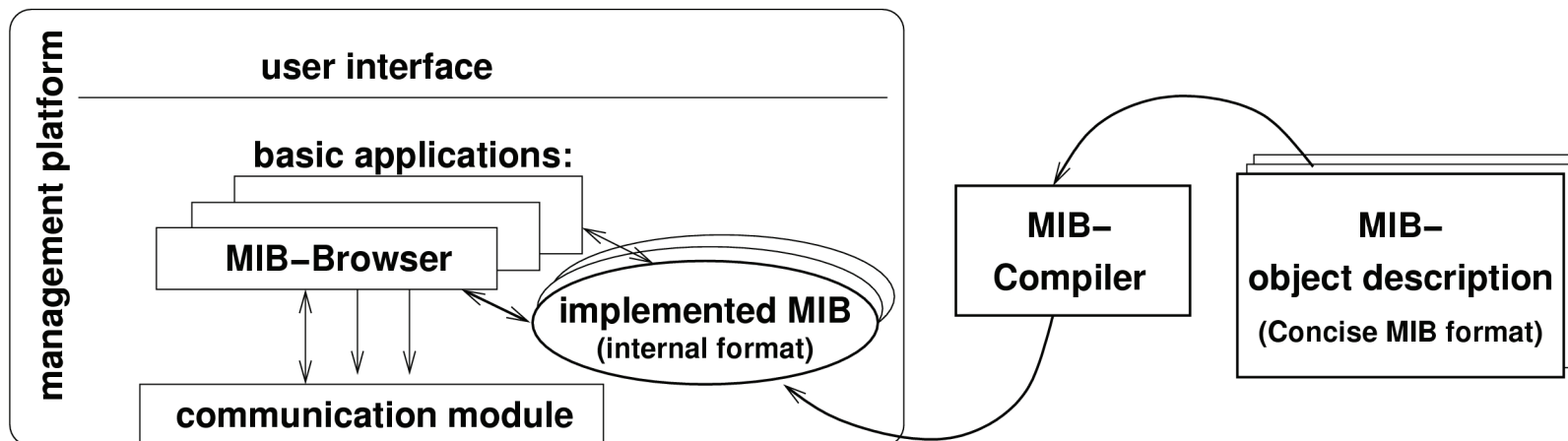
## ❑ Tools for SNMP management: MIB-Compiler

- MIB-Compilers are almost always required to make vendor and manufacturer specific MIBs available to the management platform.
  - they are necessary “support-utility”
  - Usage is complicated, and compilers & MIBs often contain errors.
  - Vendors of platforms and management applications offer a special service:
    - removal of errors in “concise format MIBs” and compiling of the MIB to produce the platform internal format
  - MIB-Compilers are also available as public domain software these simplify the access to MIB objects (often C/C++ libraries).

# Development Tools for Management

## ❑ Tools for SNMP management MIB-Compiler:

- Management information made available by agents is defined in the Concise MIB Format (RFC 1212) and provided as ASCII files when buying a network device or another managed module.
- The MIB-Compiler transforms the Concise MIB Format into a platform internal representation of the management information.



## Entwicklungswerkzeuge (2)

- ❑ Werkzeuge zur Entwicklung von Agenten (zur Interaktion zwischen Agent und Ressource)
  - für OSI-Agenten (z.B. TMN Workbench)
  - für SNMP-Agenten
    - (z.B. MIB-Compiler, MIB-Stubs, Extensible Agents)
  - zu DMTF-Agenten
  - für CORBA-Agenten
  - für JAVA-Agenten (z.B. JDMK)
- ❑ Umfassen Agent Core, Protocol Adapters, Basic Services (e.g. notification, filtering, persistent storage)

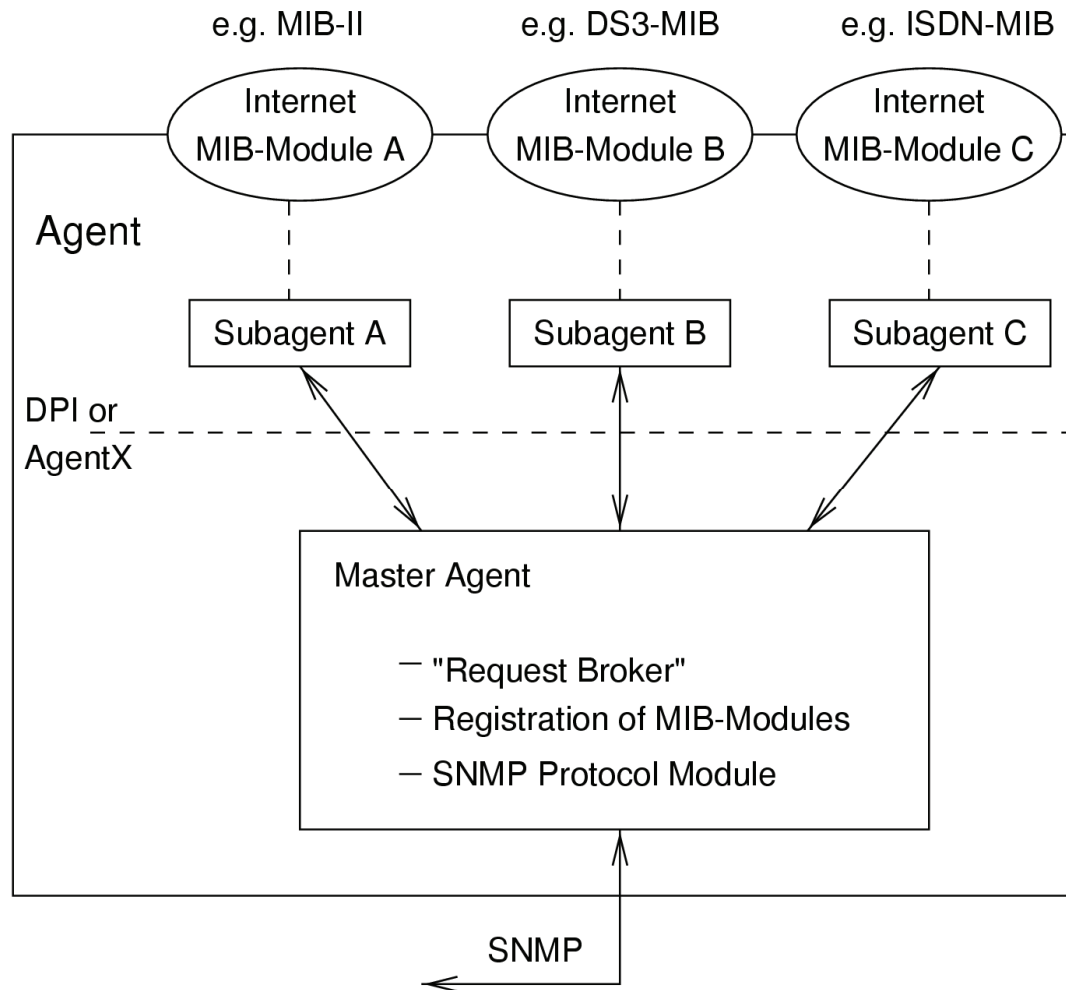


# Managementagenten

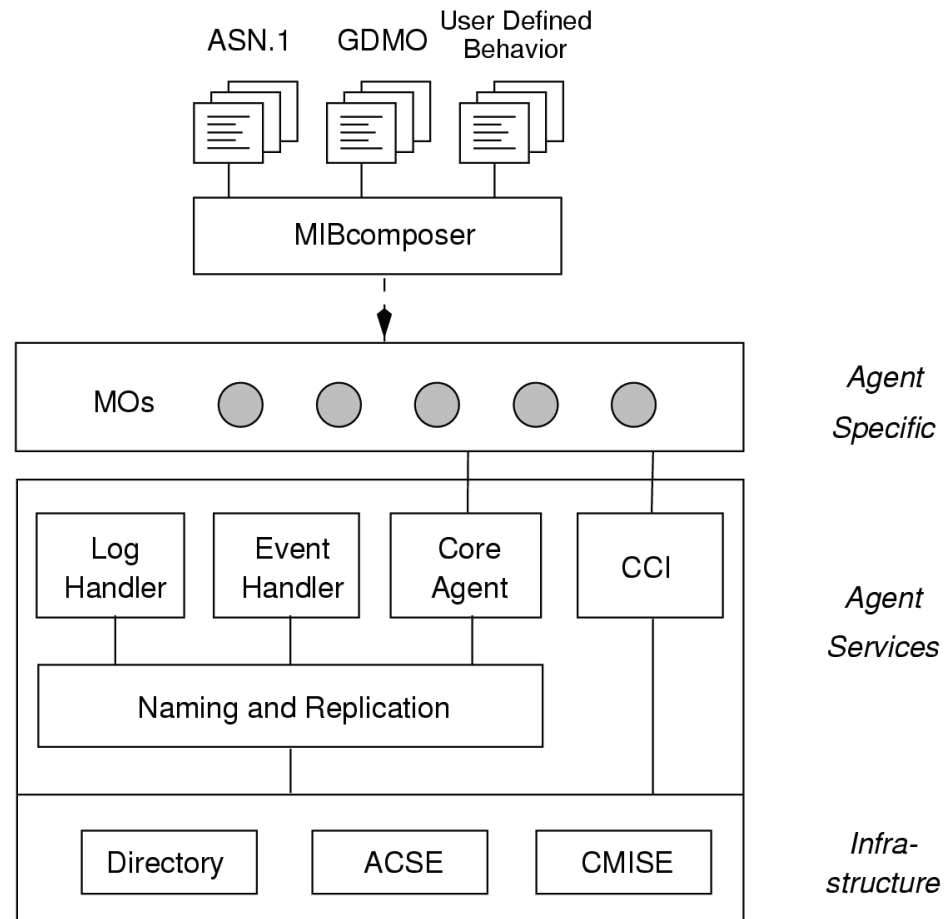
## □ Beispiele von Agenten:

- SNMP-Agenten und SMNP MIBs für  
Cisco-Router 12000, Cisco 7000, Cisco 8000 ...  
3Com-Switches: SuperStack II 3300, SuperStack II 9300  
und alle sonstigen Network Devices
  
- ⇒ Primär pure MIB-Managementinformation, wenig  
Managementfunktionalität, globales und umfassendes  
Management fast unmöglich!

# AgentX und DPI



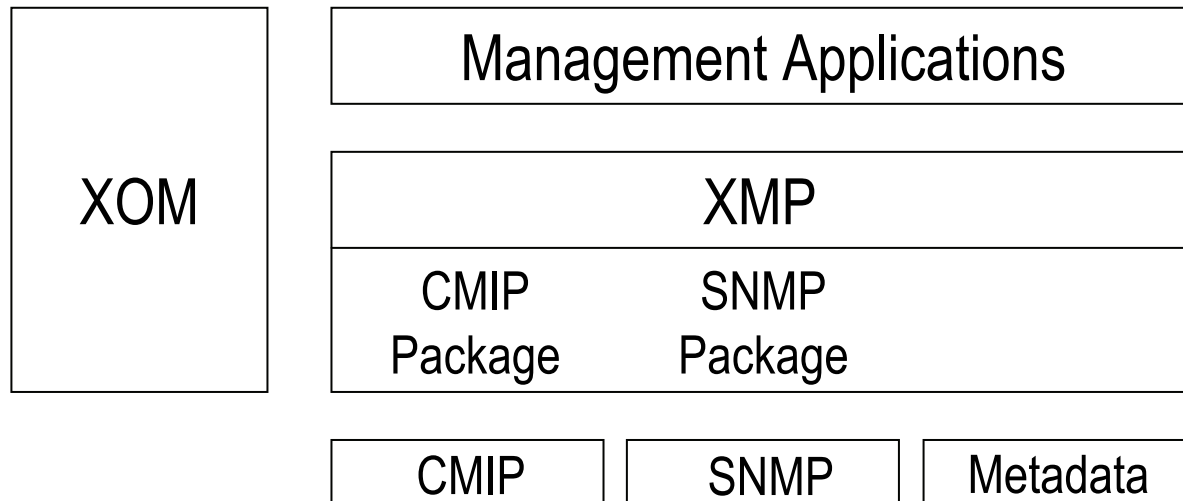
# Entwicklung von OSI-Agenten



# Entwicklungswerkzeuge (3)

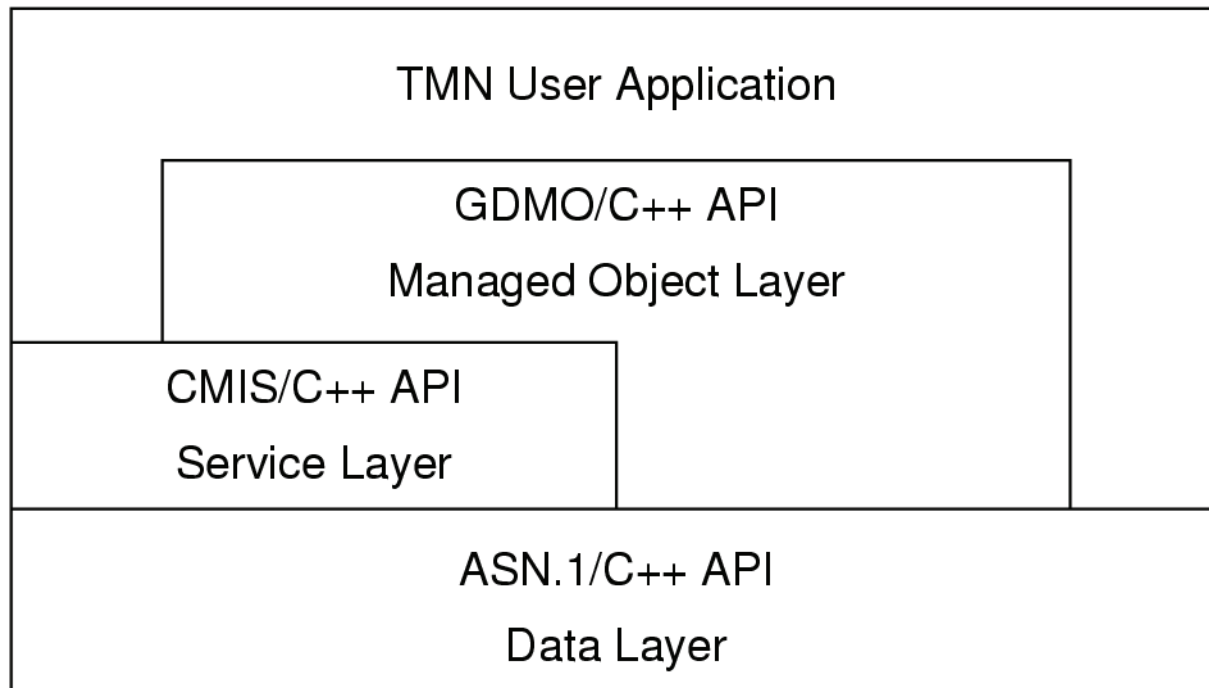
- ❑ Werkzeuge zur Entwicklung von Managementanwendungen
  - APIs: XMP/XOM, TMN/C++API, SNMP++API, Java-API
  - Schnittstellen zu Datenbanken, einheitlicher agent view und database view auf MOs
  - Verfeinerte Event-Dienste
  - Stubs für MO-Zugriff
- ❑ Werkzeuge zur Oberflächengestaltung
  - für X-Windows
  - für JAVA

# X/Open XMP and XOM:

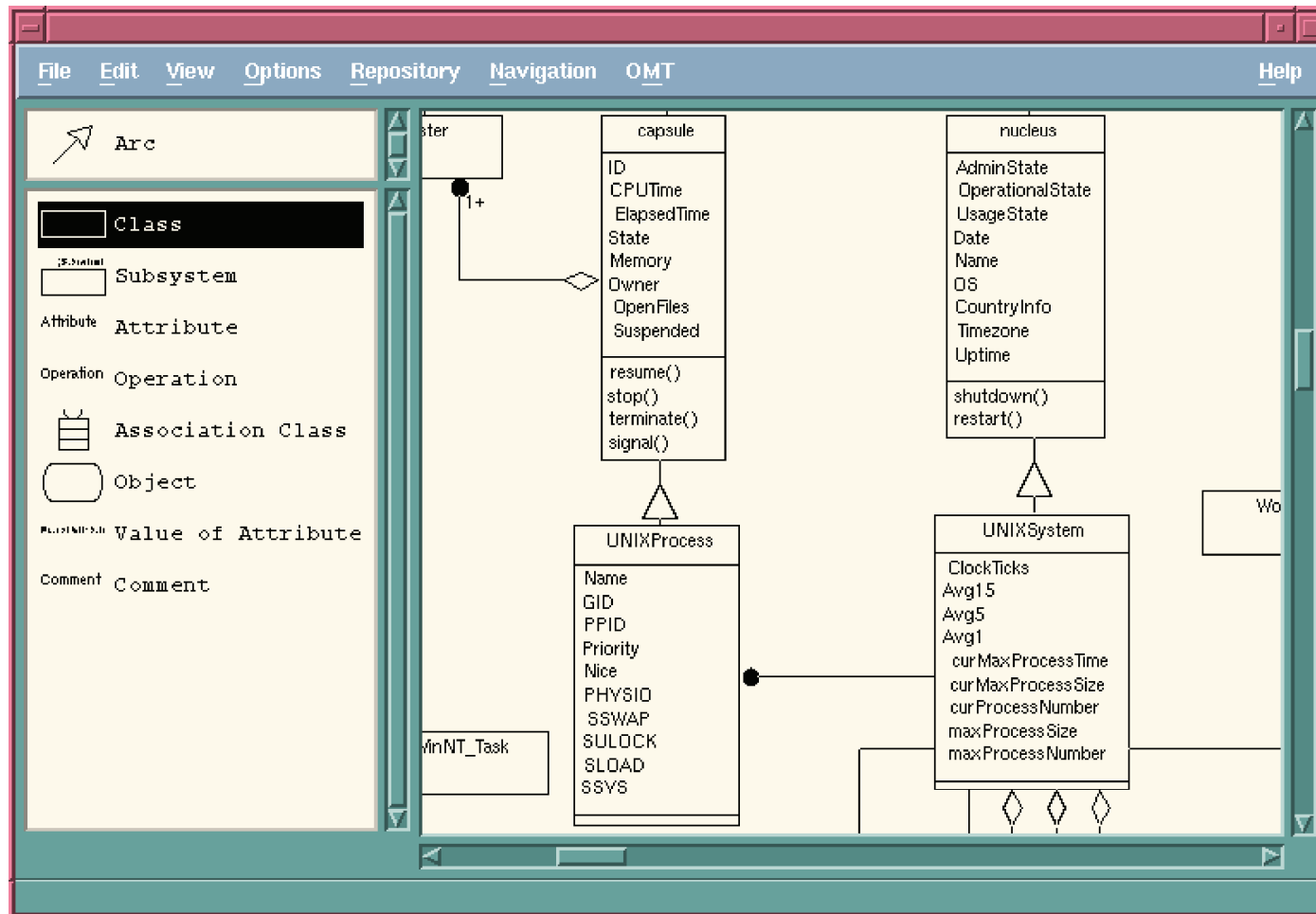


- ❑ XMP: API for a (syntactical) unified access to the protocol functionality of CMIP and SNMP
- ❑ XOM: API for local manipulation of ASN.1 data types (e.g., for translation of parameters during access to XMP)

# Architektur von TMN/C++ API



# StP-Objektmodell-Editor



# Konfiguration für Systemtabelle

Object Display Defaults	
<b>Display Properties</b>	
Type:	<input type="radio"/> Glyph
	Display Status Information: <input checked="" type="radio"/> Yes <input type="radio"/> No
	<input checked="" type="radio"/> <b>Status Control</b>
Display Type:	Graph
Direction:	Horizontal
Graph Type:	Line
<b>Status Control Configuration</b>	
Variable:	Disk Utilization
File System:	/export/home
<input checked="" type="radio"/> <b>Warning</b>	High: 80 to 90 Low: to
<input checked="" type="radio"/> <b>Critical</b>	High: 91 to 100 Low: to
<input type="radio"/> <b>Failure</b>	High: to Low: 0 to 0
Sample Rate:	20 times per Minute



# Managementspezifische Oberflächengestaltung

- ❑ Ziel: Integration von Werkzeugen unter einer Oberfläche
- ❑ Vorherrschend: X-Windows, NT-Oberfläche, Web-Browser
- ❑ Elemente:
  - Darstellung (Icons) für Ressourcen
  - Aufbau von Objekthierarchien und Maps
  - Objekt-Attribute (Property books)
  - tabellarische Darstellungen
  - Darstellung veränderlicher Werte
  - Konfiguration von Darstellungen
- ❑ Werkzeuge: HP-OV-Windows, Java AWT, SWING/JFC

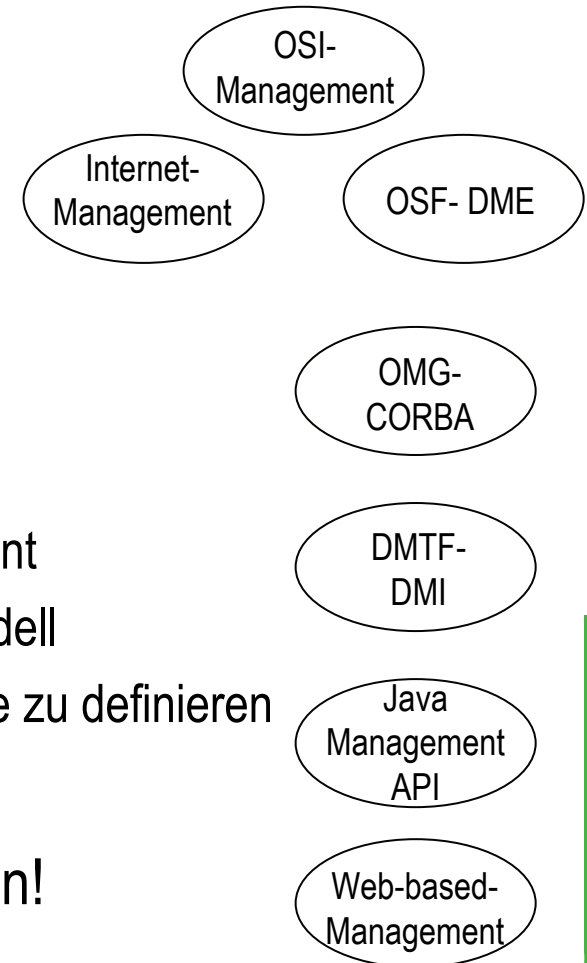
# Notwendigkeit von Architektur-Übergängen

❑ Diverse verschiedene Managementarchitekturen vorhanden oder im Entstehen

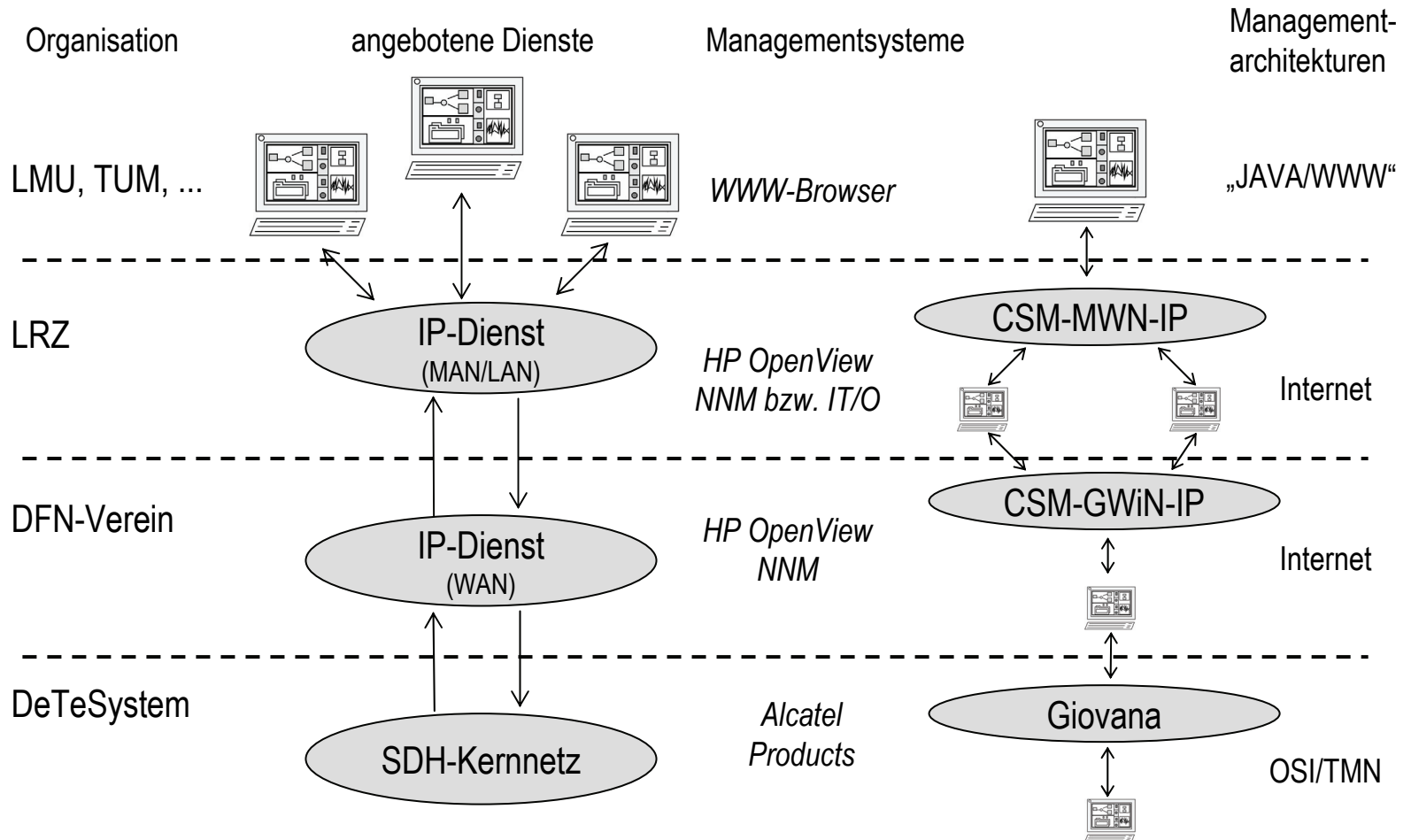
❑ Grundsätzliche Varianten:

- Spezielle Managementarchitekturen
  - spezielles Protokoll und Informationsmodell
  - spezielle Dienste, Funktionen
- Allgemeine Architektur, verfeinert für Management
  - allg. Protokoll, Informations- bzw. Objektmodell
  - Management-Dienste, -Information, -Objekte zu definieren

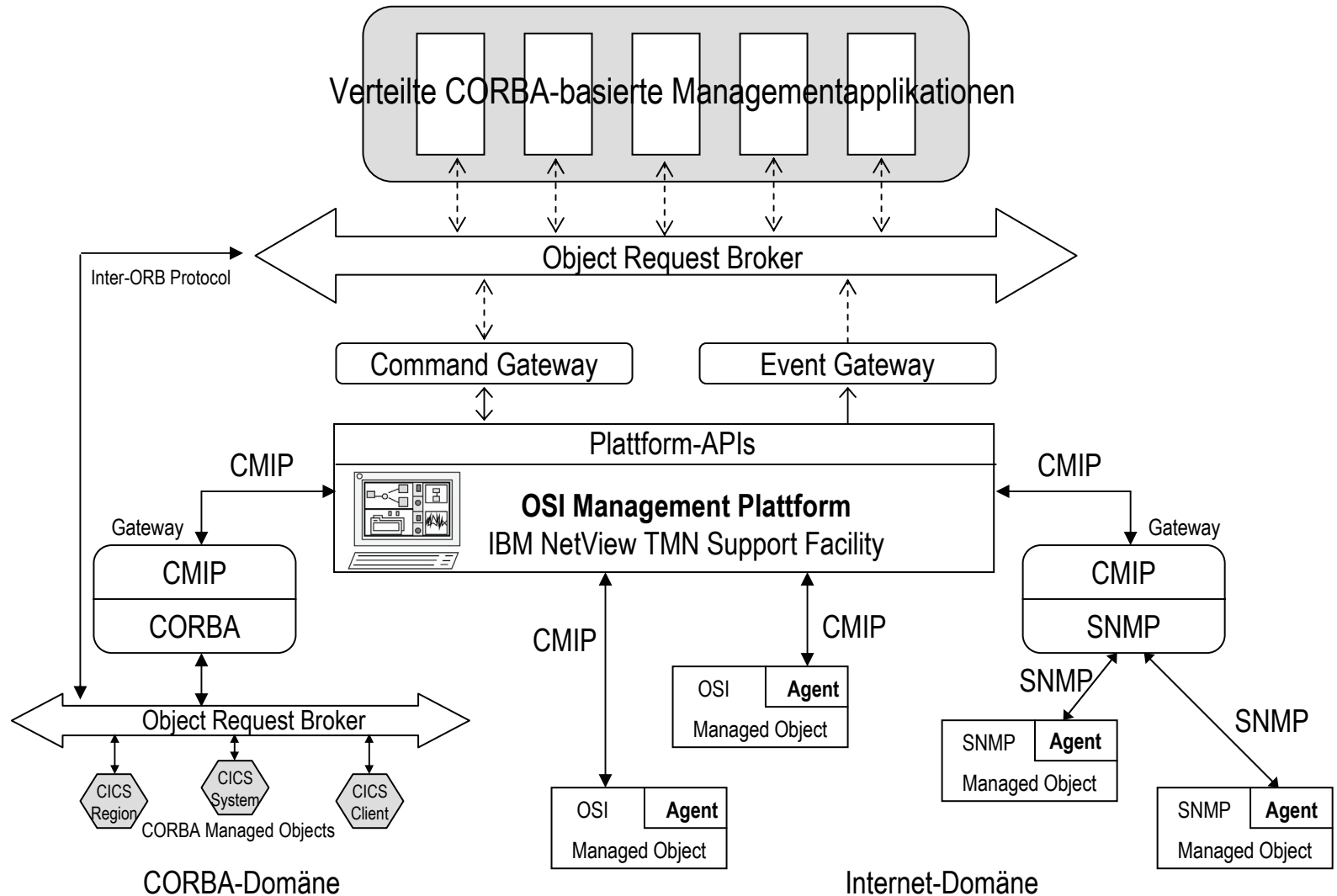
❑ Interoperabilität sichern, Übergänge schaffen!



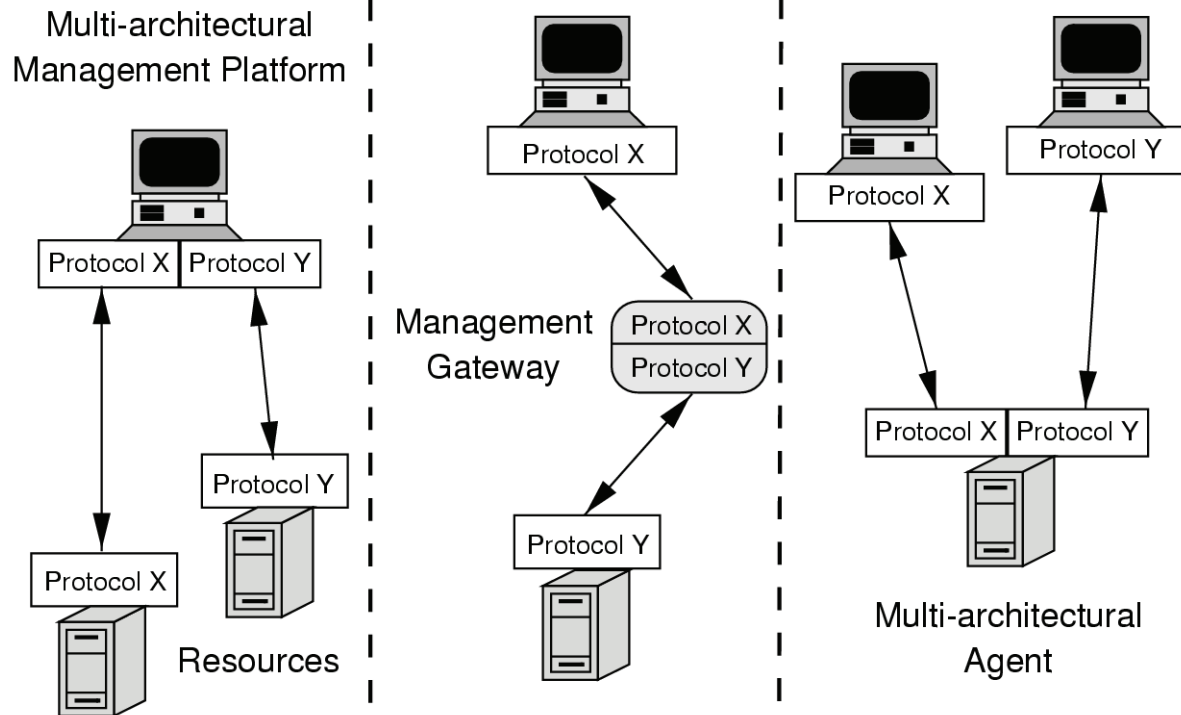
# Managementsysteme und -architekturen



# Anwendung von Management-Gateways im TMN-Management



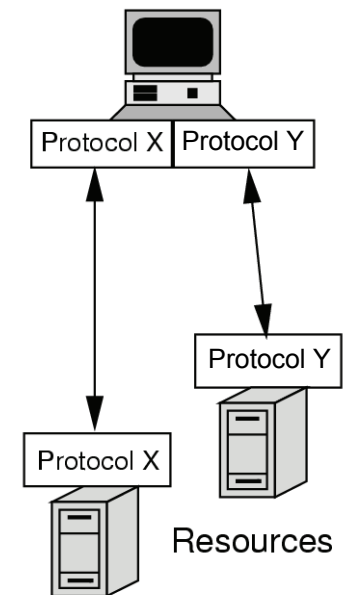
# Architekturübergänge



# Multiarchitekturelle Plattform

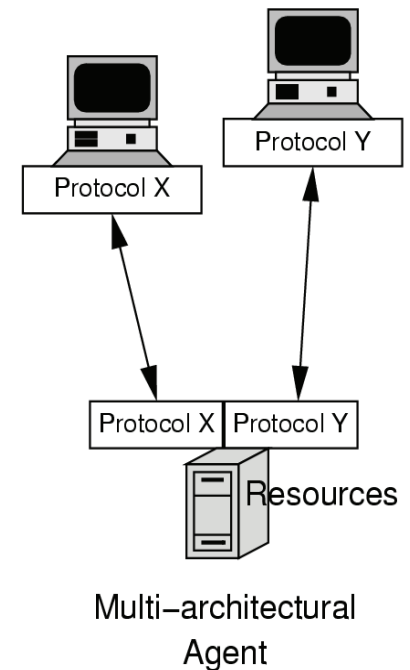
- ❑ Architekturübergang in Management-Plattform
- ❑ Management-Plattform: realisiert (Teile d.) Architektur
  - Kern (Kommunikation, Datenbasis ...)
  - Ereignis-, Topologiemanagement ...
  - GUI
- ❑ Multiarchitekturelle Plattform:
  - „spricht alle Management-Sprachen“
- ❑ keine direkte Umsetzung der Protokolle nötig, nur Abbildung auf Komm.-API der Plattform
- ❑ Unterschiede der Architekturen „schlagen häufig bis zur Anwendung durch“
- ❑ Nebeneinander unter gemeinsamer Oberfläche

Management Platform



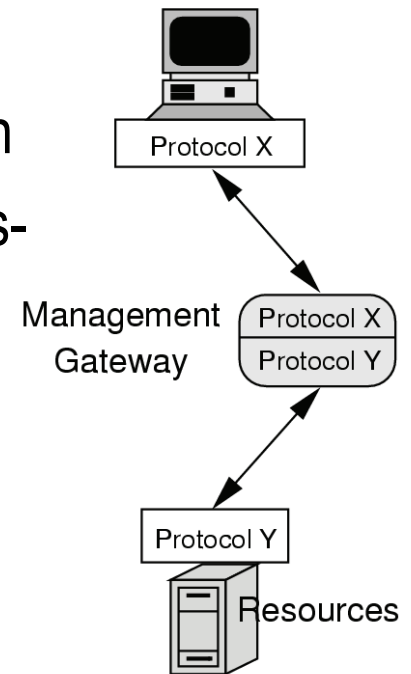
# Multiarchitektureller Agent

- ❑ Architektur-Übergang im Agenten
- ❑ Multiarchitektureller Agent:  
„spricht mehrere Management-Sprachen“
- ❑ keine Umsetzung der Protokolle nötig
- ❑ keine Umsetzung der Informationsmodelle nötig
- ❑ (eventuell) Probleme beim Zugriff auf  
gemeinsame Ressourcen
- ❑ Aufwendig wegen i.a. großer Zahl an Agenten
- ❑ Trotzdem relativ häufig:  
„mitgelieferter Standard-Agent“ +  
nachinstallierte „Spezial-Agenten“



# Management-Gateway

- ❑ Architektur-Übergang in (logischem) Zwischensystem
- ❑ kein reiner „Protokoll-Konverter“
- ❑ Unterschiede in Inf.-modellen zu berücksichtigen
- ❑ Statischer Aspekt: Übersetzung der Informationsmodelle – „Specification Translation“
- ❑ Dynamischer Aspekt: Umsetzung der Protokolle bzw. deren Elemente: „Interaction Translation“
- ❑ Weitgehend transparent für Managementanwendungen und -Agenten

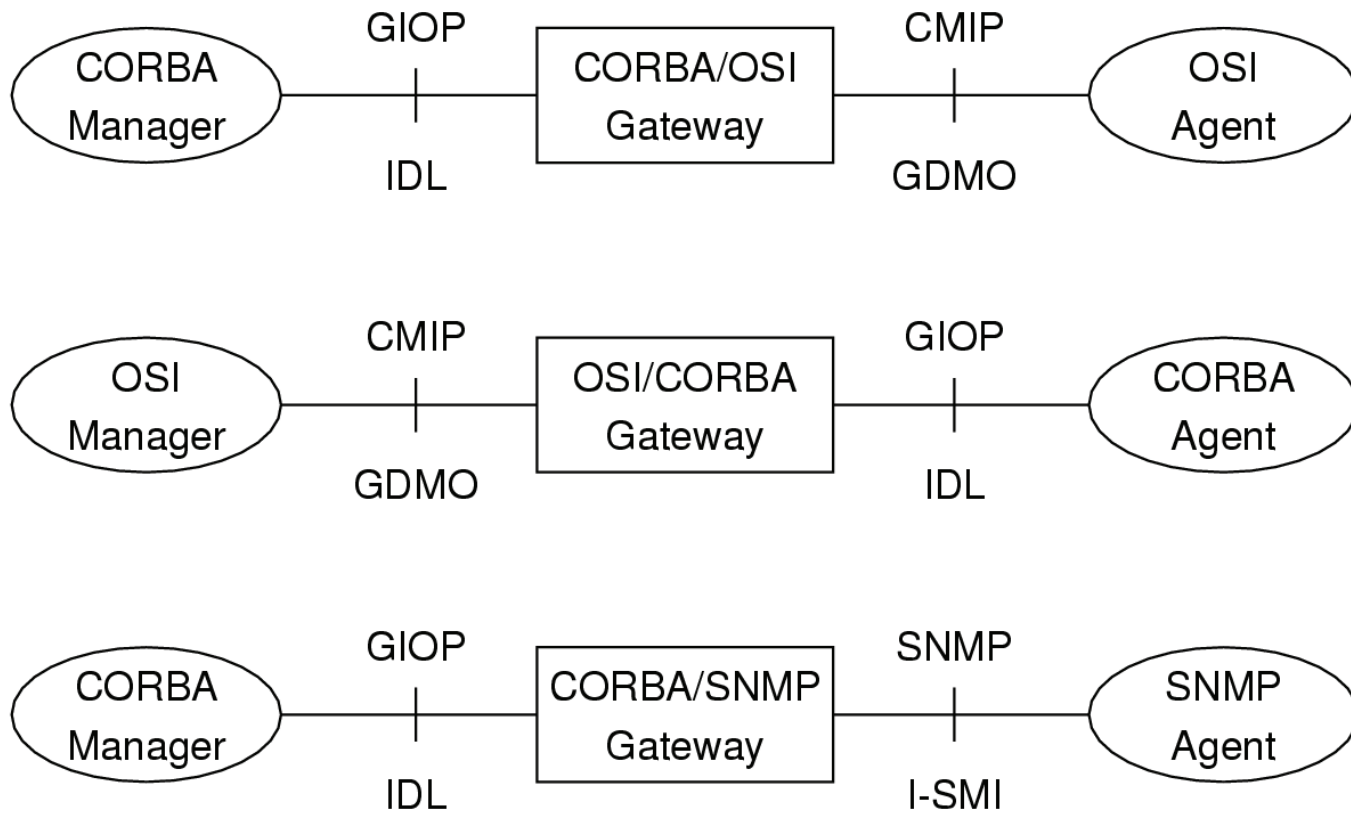




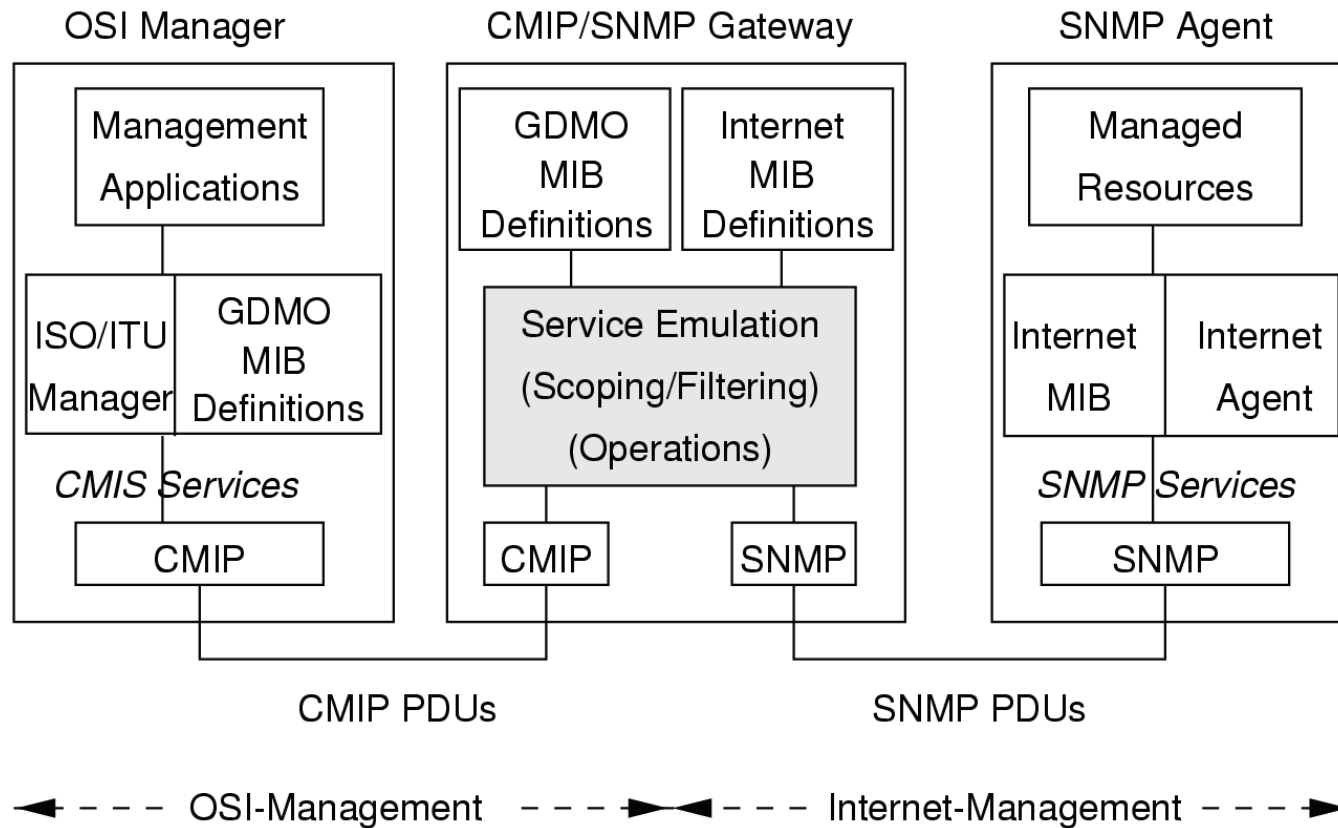
# Management-Gateways

- ❑ Übersetzung der Managementinformation
  - auf syntaktischer Ebene
  - auf semantischer Ebene
- ❑ Protokollumsetzung
  - Namensabbildung
  - Dienstabbildung
  - zustandbehaftetes/zustandsloses Gateway
- ❑ Beispiel: IIMC (ISO/Internet)
- ❑ Beispiel: JIDM (Übergang zu CORBA)

# JIDM-Szenarien



# CMIP/SNMP-Gateway



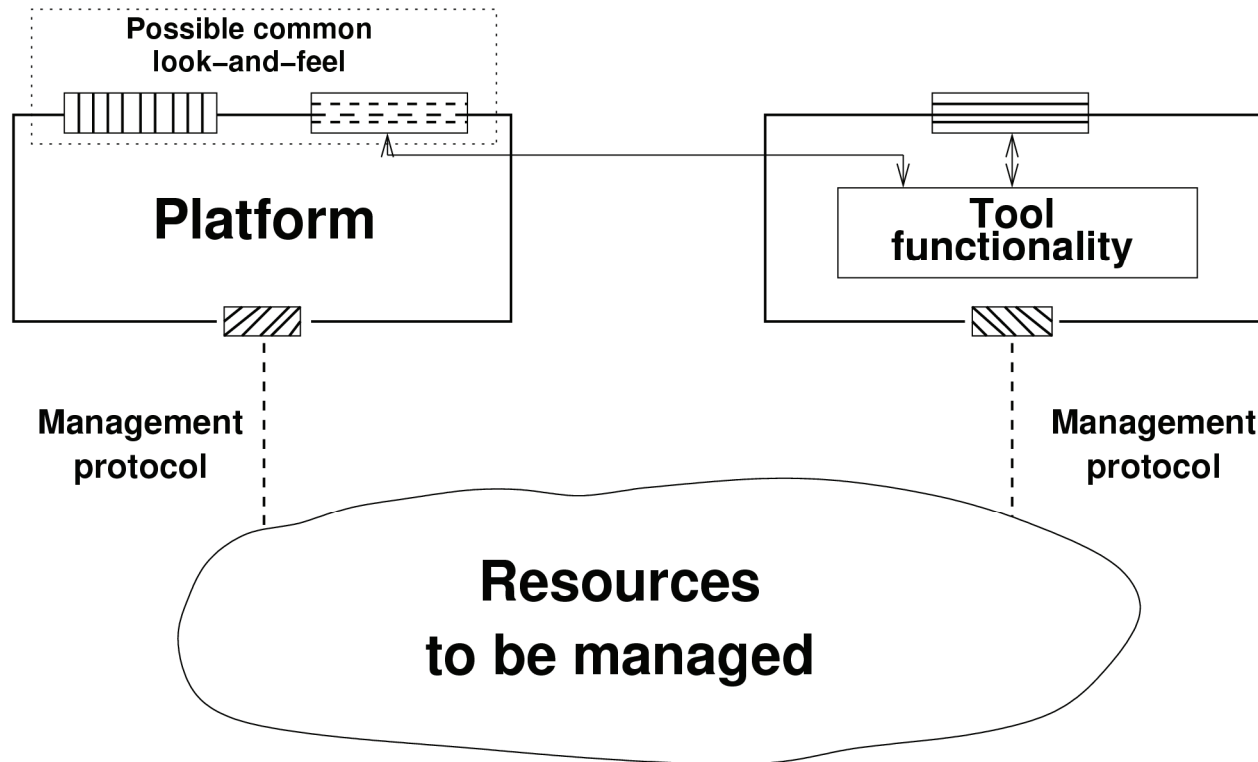
# Techniken der Werkzeugintegration

□ Kriterium:

Welche Dienste der Plattform werden benutzt?

- Oberflächenintegration
- Integration über Proxy-Agenten
- Integration mittels Datenaustausch
- Vollständige Integration

# Oberflächenintegration



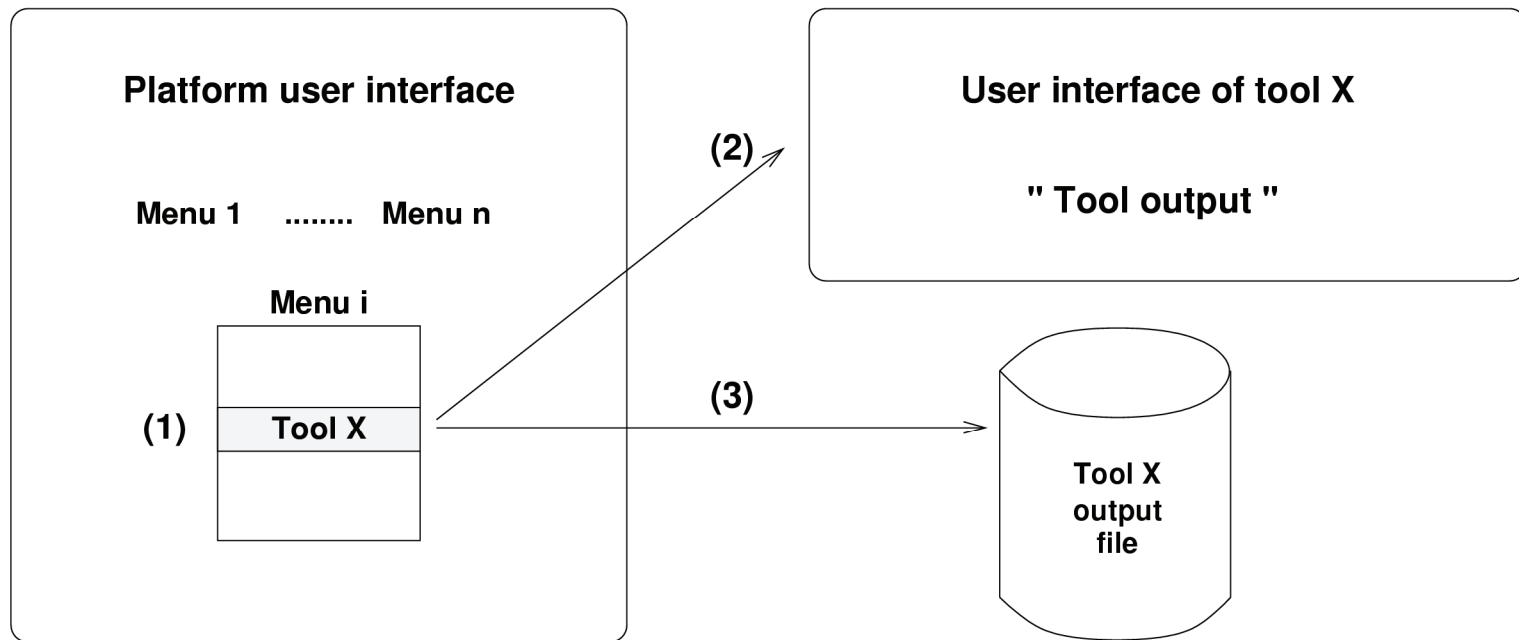
# User Interface Integration

Integration of management applications through the use of a common user interface:

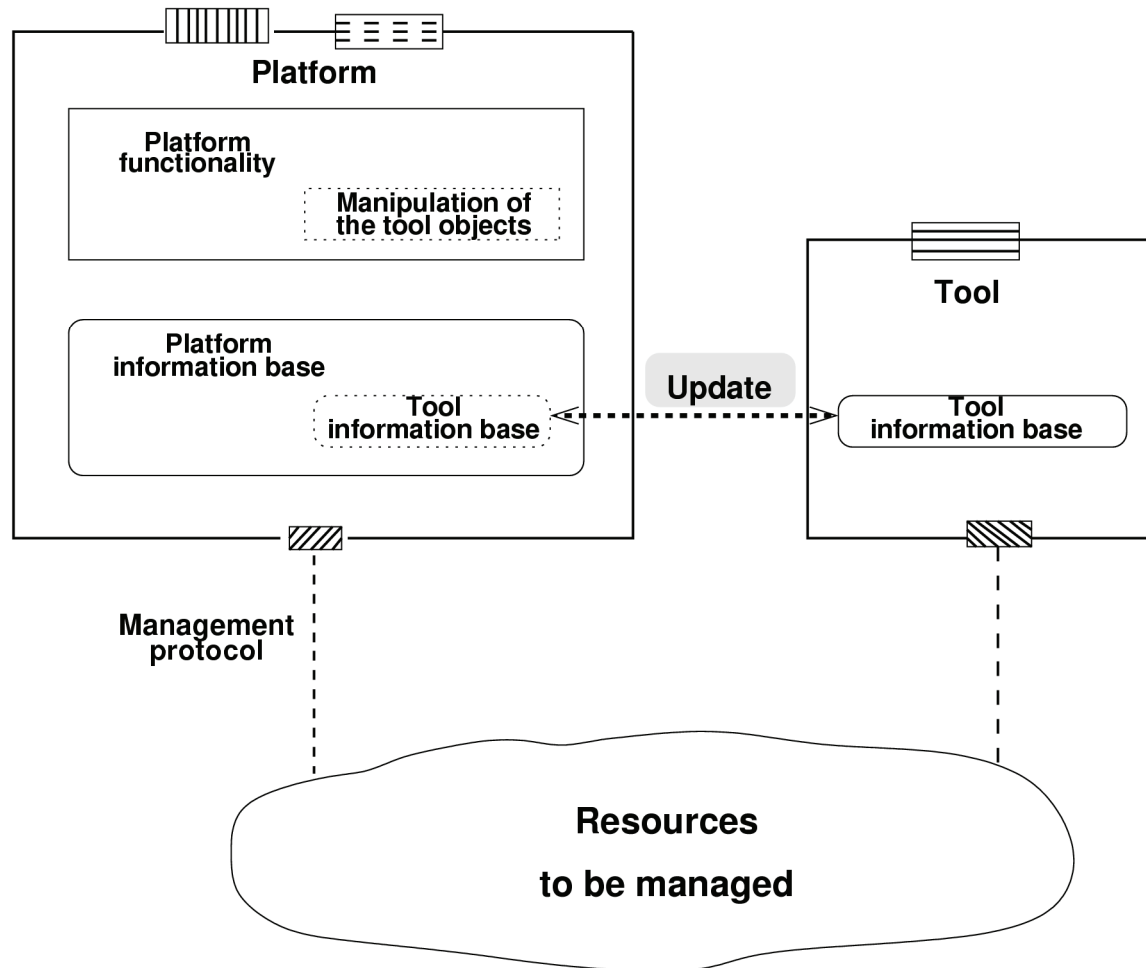
„Menu integrations“:

- tool is activated by a corresponding command and passing parameters from the platform to the tool
- separate windows are used to display the tool and its output
- the output of the tool may be piped into files

# User Interface Integration



# Methods of Integration: Data Integration





# Data Integration

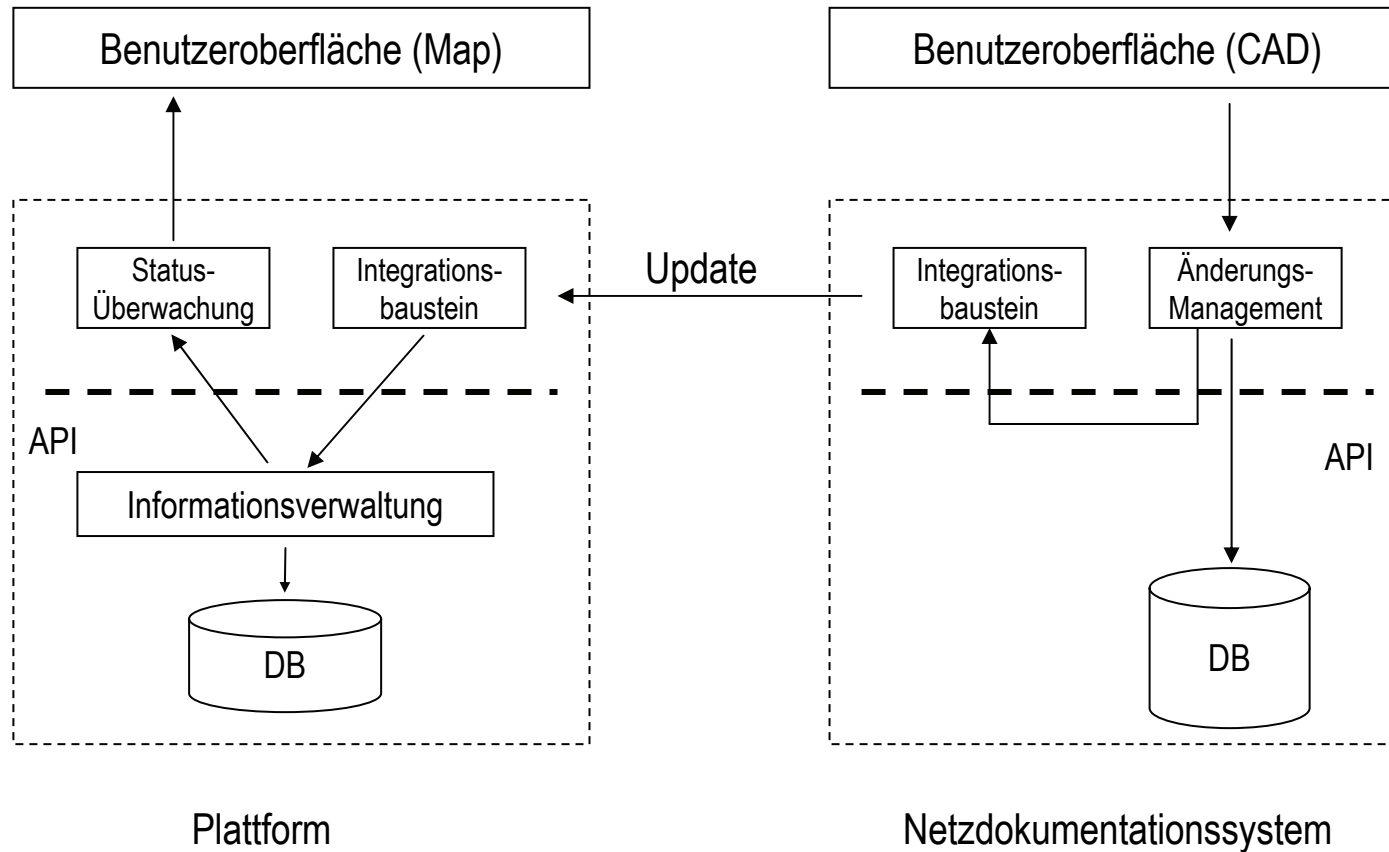
Aspects:

Mapping of different information structures

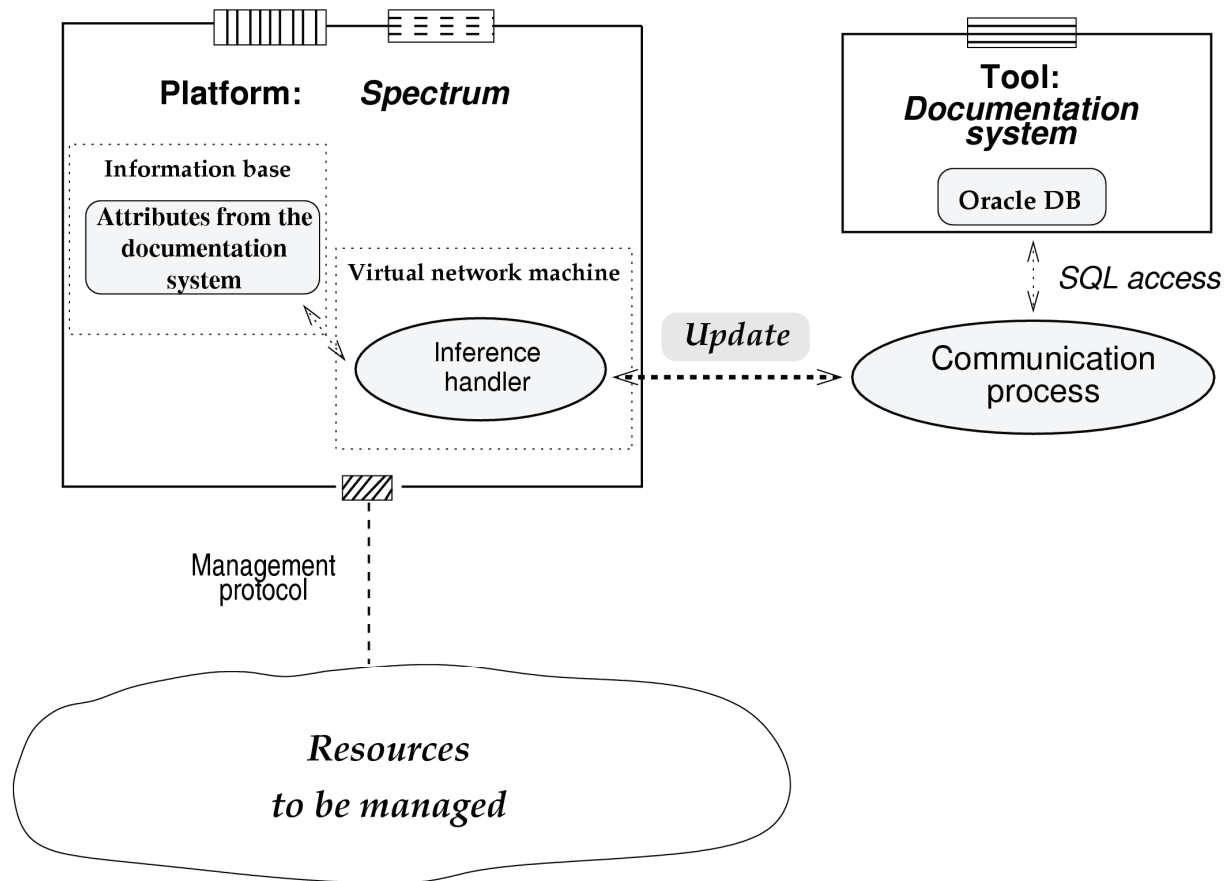
Update mechanisms:

- whenever data in an information base is changed
- at fixed intervals
- on explicit request
- on access to data

# Integration: Datenaustausch



# Data Integration

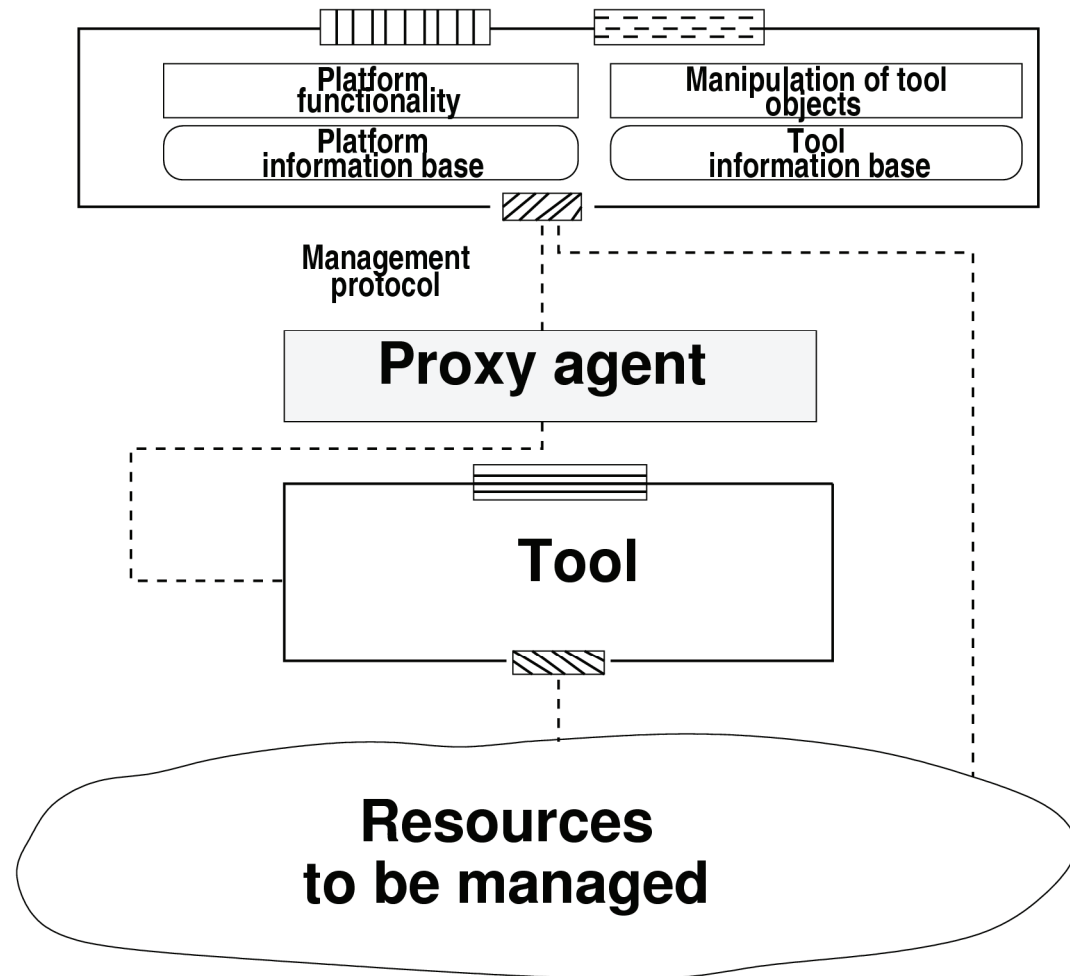


# Integration via a Proxy Agent

- ❑ Properties of „proxy integrations“:
  - the tool as such is preserved (i.e., no modifications),
  - in addition to the user interface, parts of the functionality of the tool is integrated in the platform
  
- ❑ Tasks of the proxy:
  - represent the tool and its functionality
  - ensure consistency between tool-object and reality

# Methods of Integration: Proxy Agent

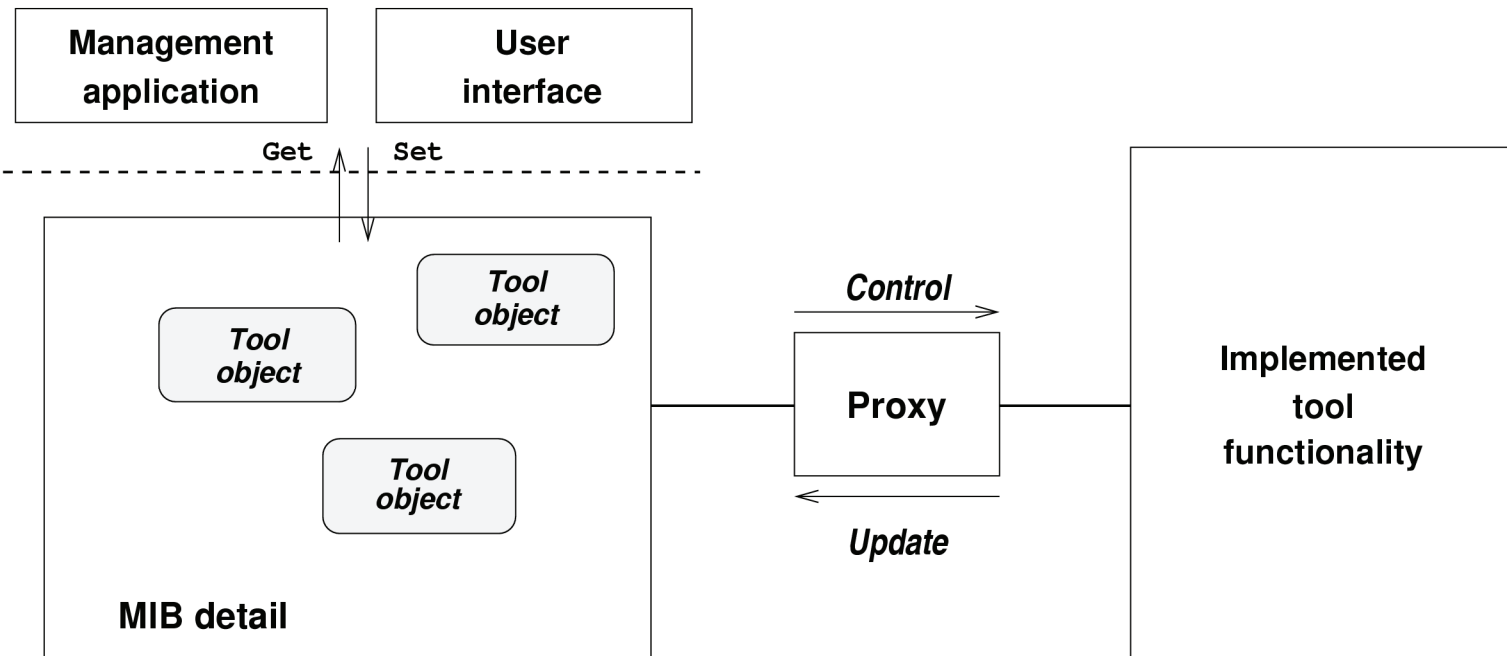
Integration via a proxy agent:



# Integration via a Proxy Agent

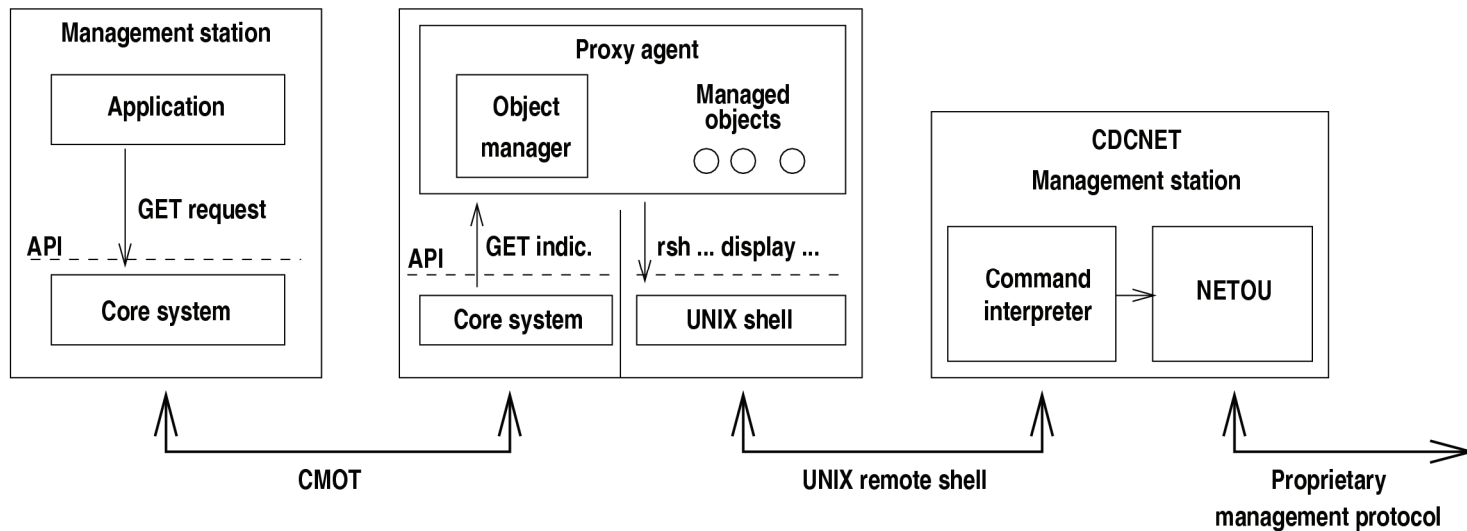
*Platform side*

*Tool side*



# Integration durch Proxy-Agent

Plattform wird übergeordnet:



# Integration via a Proxy Agent

## CD NETOU:

`display_osi_transport_status`

## OSI Transport Status

number of SAPs = 2

number of class 0 connections = 2

number of class 2 connections = 0

number of class 4 connections = 8

## HP OpenView:

CDCTransportEntity MANAGED OBJECT CLASS

DERIVED FROM top

CHARACTERIZED BY:

BEHAVIOUR DEFINITION CDCTransportBehaviour

ATTRIBUTES

NumberOFSAPs GET

Class0Connections GET

....

Registered as {...}



# Integration durch Proxy-Agent

Plattform wird untergeordnet:

