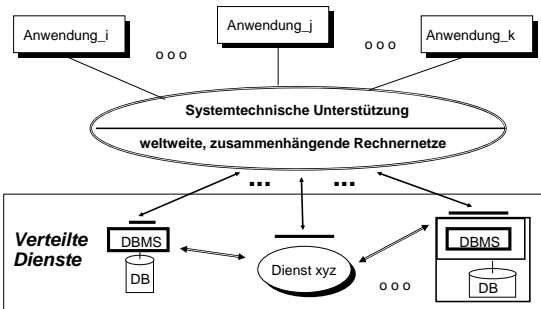


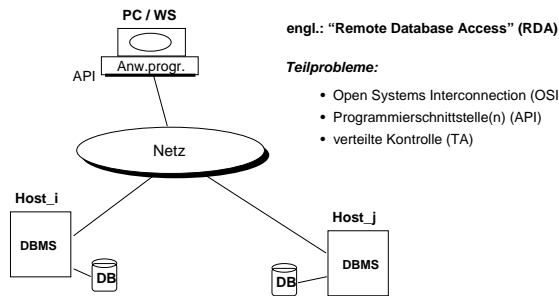
Verteilte Datenhaltung:

Zugang zu Datenbanken in Rechnernetzen
(Remote Database Access, RDA)

Allgemein: Systemtechnische Unterstützung verteilter Anwendungen in offenen Rechnernetzen



Spezielles Problem: Zugang zu Datenbanken in offenen Rechnernetzen



engl.: "Remote Database Access" (RDA)

Teilprobleme:

- Open Systems Interconnection (OSI)
- Programmierschnittstelle(n) (API)
- verteilte Kontrolle (TA)

Datenbanken in Rechnernetzen: Systemklassifikation

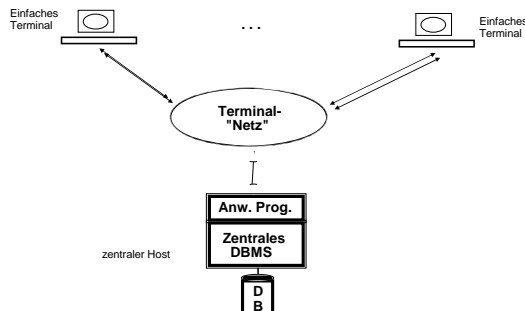
Komponenten verteilter Datenverwaltungssysteme:

- **Endbenutzerschnittstelle:** Terminals, Displays, Teller Machines, ... PCs, Workstations, ...
- **Anwendungsprogramm:** realisiert anwendungsspezifische Funktionen, benutzt DB-Dienste
- **Datenbankverwaltungssystem (DBMS):** realisiert Datenbankabfrage & -manipulationsdienste
- **Kommunikationssystem:** stellt Verbindungen zwischen verteilten Systemkomponenten her

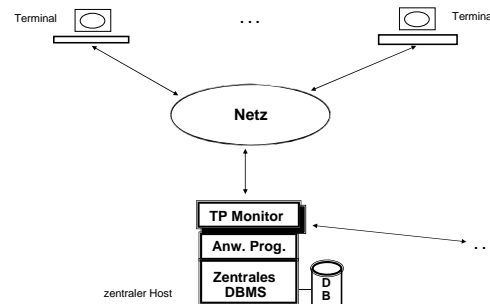
Systemalternativen:

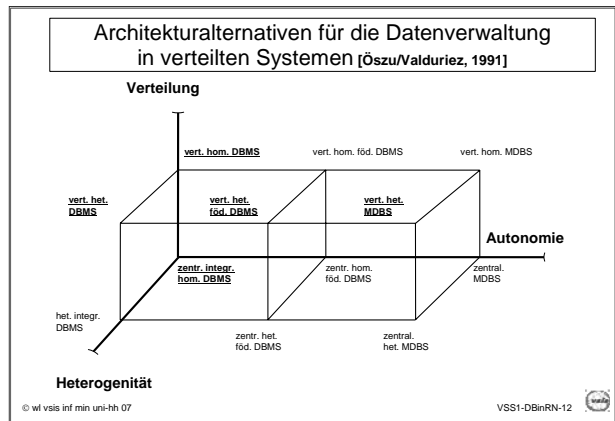
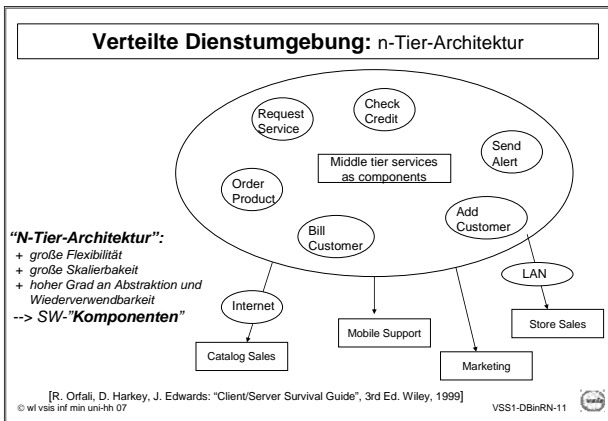
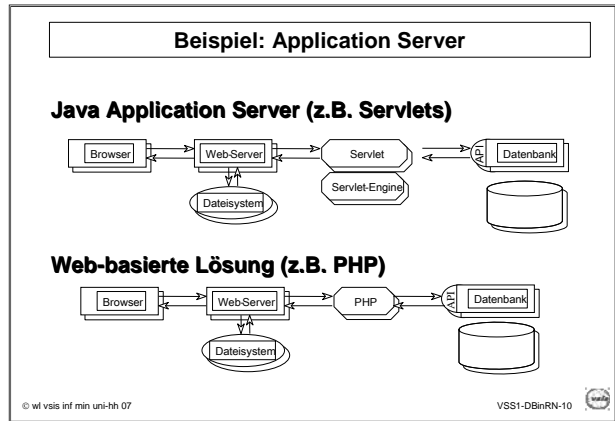
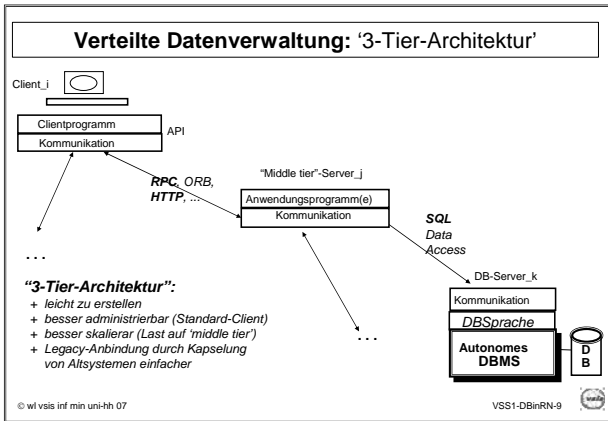
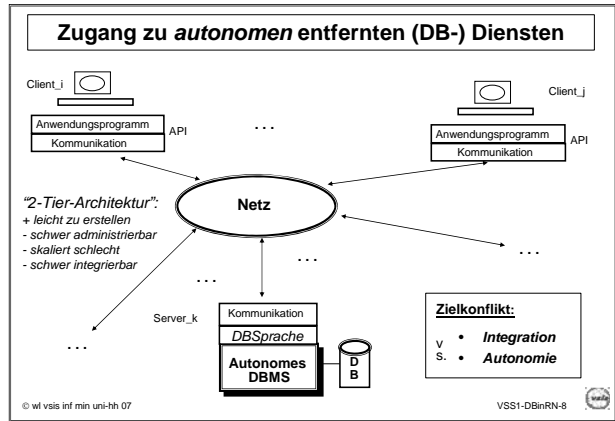
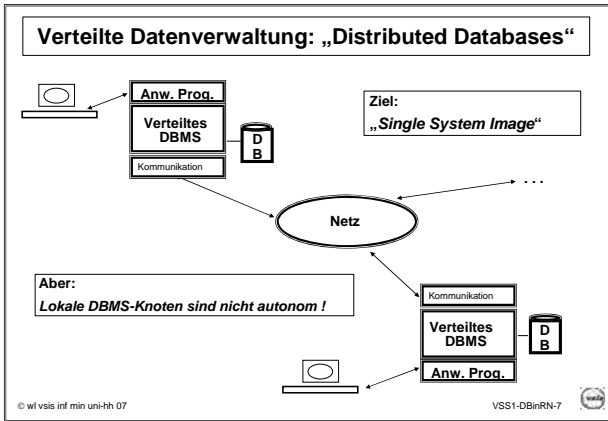
- Zentralisierte Systeme
- Transaktions- (monitor-) (TP-) Systeme
- Verteilte (Multi-, Föderative) Datenbanken
- Client / Server - Systeme

Zugang zu „zentralisierten“ (DB-) Diensten in offenen Umgebungen



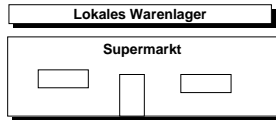
DB / DC Systeme: „TP Monitore“





Kooperationsalternativen verteilter Verarbeitung

A) "Supermarkt"-Modell (Verteilte Datenbanken)



- Kunde nimmt selbst, was er braucht
- Waren alle gleichzeitig präsent
- Waren gleichartig präsentiert
- Warenbestand relativ fix
- Wenig Angestellte (geringer Aufwand)
- Viel Aufwand bei Warenrepräsentation

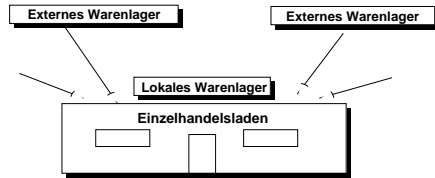
==> **Einfach zu verstehen !**

© wlvsl inf min uni-hh 07

VSS1-DBinRN-13

Kooperationsalternativen verteilter Verarbeitung

B) "Einzelhandelsladens"-Modell (Client/Server)



- Kunde gibt nur Bestellung auf
- Angestellte besorgen Waren
- Waren verschiedenartig / gar nicht präsentiert
- Waren nur zum Teil lokal präsent (wenig Aufwand)
- Viele Angestellte (höherer Aufwand)
- Sehr flexibles Warenangebot

==> **Einfach zu betreiben !**

© wlvsl inf min uni-hh 07

VSS1-DBinRN-14

Beispiel: erste ISO/OSI- Standardisierung des "Remote Database Access" (RDA)

European Computer Manufacturers Association: ECMA TC 22 (Datenbanken)

- Erste RDA-Vorschläge 83-85 und 86-88
- Später aktive Teilnahme an der ISO-Arbeit

International Standards Organization: ISO / IEC JTC1 (Computer Science)

- Wichtigstes offizielles internationales Normungsgremium - mit
- **SC 21:** Open Systems Interconnection, Databases, Open Systems (!) - mit
- **WG 3:** Datenbanken: **Rapporteur Group RDA** - darin
- Aktive Mitglieder: USA, CDN, UK, F, D, AUS, J, ...
- 1. CD zum RDA in 1988/89, 2. CD 1990, DIS 1991/2, IS 1993

Nationale Standardisierungsgremien: ANSI X3H2, DIN NI 21.3, AFNOR, BSI, ...

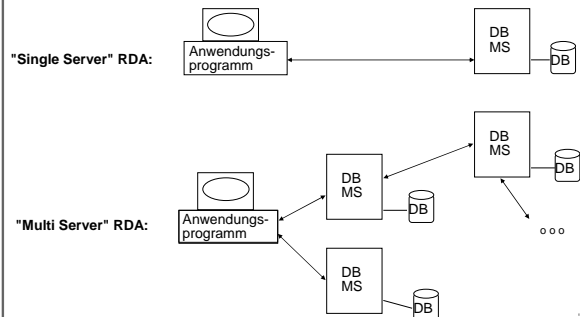
- Untergruppe(n) zu RDA
- wählende, aktive Mitglieder der ISO-Gremien

+ weitere (z.B. industrielle) Interessengruppen - wie „**SQL-Access Group**“ (SAG)

© wlvsl inf min uni-hh 07

VSS1-DBinRN-15

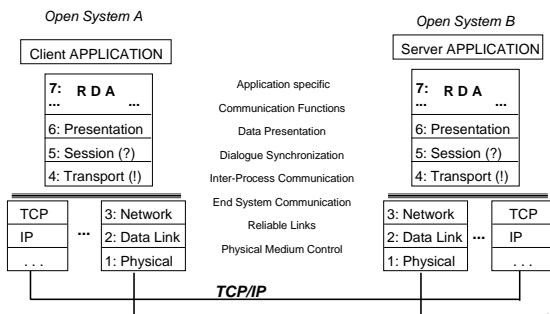
Alternative Komplexitätsstufen des RDA



© wlvsl inf min uni-hh 07

VSS1-DBinRN-16

RDA - Kommunikationsunterstützung: Varianten



© wlvsl inf min uni-hh 07

ISO/OSI

VSS1-DBinRN-17

ISO / OSI RDA: Beispiel einer Dienstauffolge

"Generischer" RDA (Part 1)

- R-Initialize
- R-Open
- R-BeginTransaction
- R-Execute

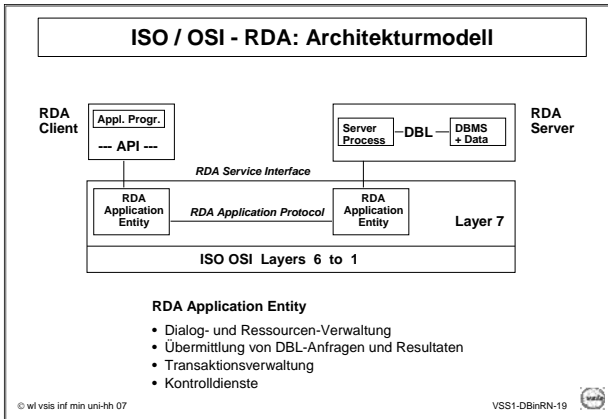
```
* Select..., Find..., Fetch...
* Insert..., Update..., Delete...
* ...
```

RDA "SQL - Spezialisierung" (Part 2)

- R-Status
- ...
- R-Commit
- ...
- R-Terminate

© wlvsl inf min uni-hh 07

VSS1-DBinRN-18



Übersicht über RDA-Dienstelemente: Datenbanksprachdienste

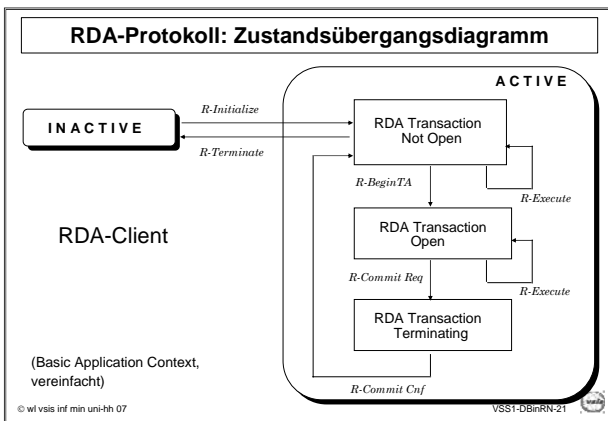
Immediate Execution DBL Functional Unit

Dienstelement	Beschreibung	Client	Server
R-Execute	Ausführung einer DBL-Operation	req cnf	ind rsp

Stored Execution DBL Functional Unit

Dienstelement	Beschreibung	Client	Server
R-Define DBL	Validierung und Speicherung einer DBL-Operation	req cnf	ind rsp
R-Invoke DBL	Aufruf einer vor-defin. DBL-Op.	req cnf	ind rsp
R-Drop DBL	Löschen einer vor-defin. DBL-Op.	req cnf	ind rsp

© wlv isis inf min uni-hh 07 VSS1-DBinRN-20



ISO / OSI 'Common Application Service Elements' (CASE) - Übersicht über Komponentenstandards für RDA

Application Layer Structure (ALS):
Grundsätzliche Beschreibung der Struktur von Anwendungsstandards der ISO/OSI-Ebene 7

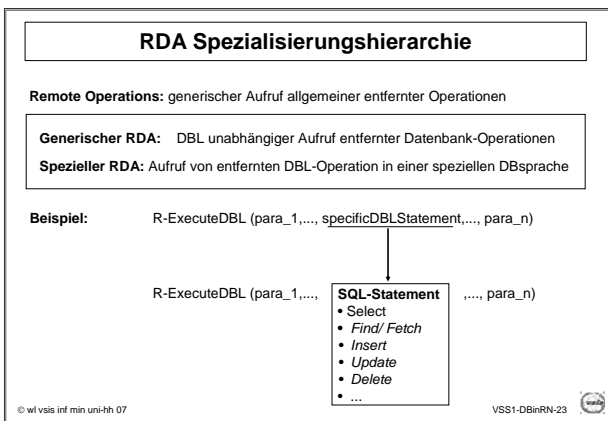
Association Control Service Element (ACSE):
Auf- und Abbau einer Association zwischen zwei kommunizierenden Prozessen; Fehlermeldungen

Remote Operations (ROSE):
Art und Weise, Dienste und (minimales) Protokoll für die Ausführung entfernter Operationen in verteilten Systemen zu spezifizieren; bei RDA nur der Idee nach - nicht als separate ASE - verwandt

Commitment Concurrency and Recovery (CCR):
Protokoll für den Austausch von Nachrichten zwischen direkt benachbarten Knoten, die an verteilten Transaktionen beteiligt sind, Basis für TP

Distributed Transaction Processing (TP):
Vollständiger Dienst zum Auf- und Abbau sowie zur Verwaltung aller notwendigen Kommunikationsprimitive bei der Abwicklung von verteilten Transaktionen mittels 2-Phase-Commit-Protokoll und CCR

© wlv isis inf min uni-hh 07 VSS1-DBinRN-22



Folgearbeiten der RDA-Standardisierung

Historie des RDA : • ISO/OSI RDA (IS 9579): 1. Version fertig (IS seit Dez. 1993)

Weitere RDA/SQL-Standarddokumente:

- RDA 'PICS-Proforma' (DIS 94/5) / 'Defect Reports'
- RDA SQL2 Spec. ('Proposed Draft Amendment #1')
- SQL: 'Call-Level Interface' (CLI): DIS (94/5), 'Persistent Stored Modules' (CD 95), 'Global Transactions' (CD 95)

Mögliche Erweiterungen:

- Erweiterte Funktionen, z.B. Dialogkonzept (z.B. 'Recovery', 'suspend/resume', ...)
- Zusätzliche / erweiterte Datenbanksprachen (NDL?, SQL3, ...)
- Alternative & erweiterte Transaktionskonzepte (?)
- Unterstützung voll verteilter Datenbanken (?)
- Zugriff auf spezielle Datenbanken (z.B. IRDS: objektorientiert (ODMG?), Multimedia, ...)
- Anpassungen an Charakteristika neuartiger (z.B. HS-) Netze
- Weitergehendes Ziel: „Portable“ Anwendungsprogramme (d.h. Standard-APIs) --->

Übergang von „OSI“ (Open Systems Interconnection) zu „OS“ (Open Systems)

d.h. \Rightarrow ISO/OSI RDA + Remote SQL - API

\Rightarrow ISO/OSI TP + Distributed TP - API (s)

© wlv isis inf min uni-hh 07 VSS1-DBinRN-24

US RDA-Standardisierung: „SQL-Access Group“ (SAG)

SAG: Private Vereinigung von US DBMS-Herstellern zur Förderung des "RDA"
Mitglieder: Bull, DEC, Fujitsu, HP, SUN, Tandem, u.a. ... (ca. 40)

Hauptziel der **Datenbankentwickler:** **Interoperabilität** (inkl. Gateways)
 Hauptziel der **Anwendungsentwickler:** **Portabilität**

Randbedingung für RDA: Harmonisierung mit ISO-Standards:

- ISO/OSI - RDA : Dienste, Nachrichtenformate, Protokoll, ...
- ISO/DBL - SQL: Standard-API für (relationale) Datenbanken, Katalog, ...
- SQL-Access-Group: Multi-Vendor-Prototyp, Implementor-Vereinbarungen, Conformance Tests, Erweiterungen ... -> RDA API (X/Open+ISO): **Call Level Interface (CLI)**

BEISPIEL für SQL CLI:

"debit" (:stmt, :source, :delta)

```
{ connect to "Remote_DB" @ "a.b.c.d"
  as "Alternate_DB" user "jim" password "paris";
  prepare debit from :stmt;
  execute debit using :source, :delta;
  disconnect "Remote_DB" }
```

Microsoft „Open Database Connectivity“ (ODBC)

Windows Open Service Architecture (WOSA)

Anwendung: „Information At Your Fingertips“

Application Interface (API)
 Betriebssystem: Windows/MS-DOS oder Windows/NT + ... (?)

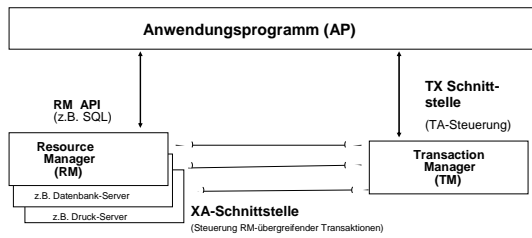
Service Provider Interface (SPI)



ODBC

- SQL-basiertes *Call Level Interface (CLI)* nach erweiterter SAG-Spezifikation
- „standardisiertes“ API (X/Open) für den Zugriff auf heterogenen Datenquellen
- Unterschiedliche *Conformance Levels* (von SAG SQL bis zu erw. DBMS-Funktionalität) für maximale DBMS-Funktionalität: jeweils *eigener Treiber* pro DBMS-Zugang
- ODBC 1.0 seit 9/92, 75 Hersteller, ca. 40 DBMS-Treiber
- ODBC 2.0: 250 Teiln. in Development Conference, Beta-Version seit 1993 verfügbar

Schnittstellenspezifikation der X/Open -> Open Group für „Distributed Transaction Processing“ (DTP)

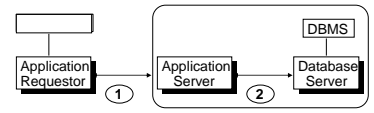


Ziel: Standardisierung von **Syntaxschnittstellen** zur Integration heterogener **Ressourcen-Manager** (z.B. DBMS) in verteilte Transaktionsverwaltung

Beispiel für proprietäre Architektur: IBM's „Distributed Relational Database Architecture“ (DRDA)

DRDA:

IBM's interner Standard für Dienst und Protokoll zum Zugang zu entfernten relationalen Datenbanken in SNA-Netzen



1 : Application Support Protocol
 2 : Database Support Protocol

DRDA basiert auf IBM's Architekturen für

- **Kommunikation:** Systems Network Architecture (SNA)
- **Datenbanken:** SQL and Distributed Data Management (DDM)
- **Präsentation:** Formatted Data Object Content Architecture (FD/OCA)
- **TA-Verwaltung:** Distributed (SNA) Transaction Processing (LU 6.2)