SAP R/3
A Database Application System

Alfons Kemper* Donald Kossmann* Florian Matthes+

*Universität Passau +Technische Universität Hamburg-Harburg

Outline

1. Application development interfaces of SAP R/3
2. Architecture and database interfaces of SAP R/3
3. Queries and OLAP applications on SAP R/3

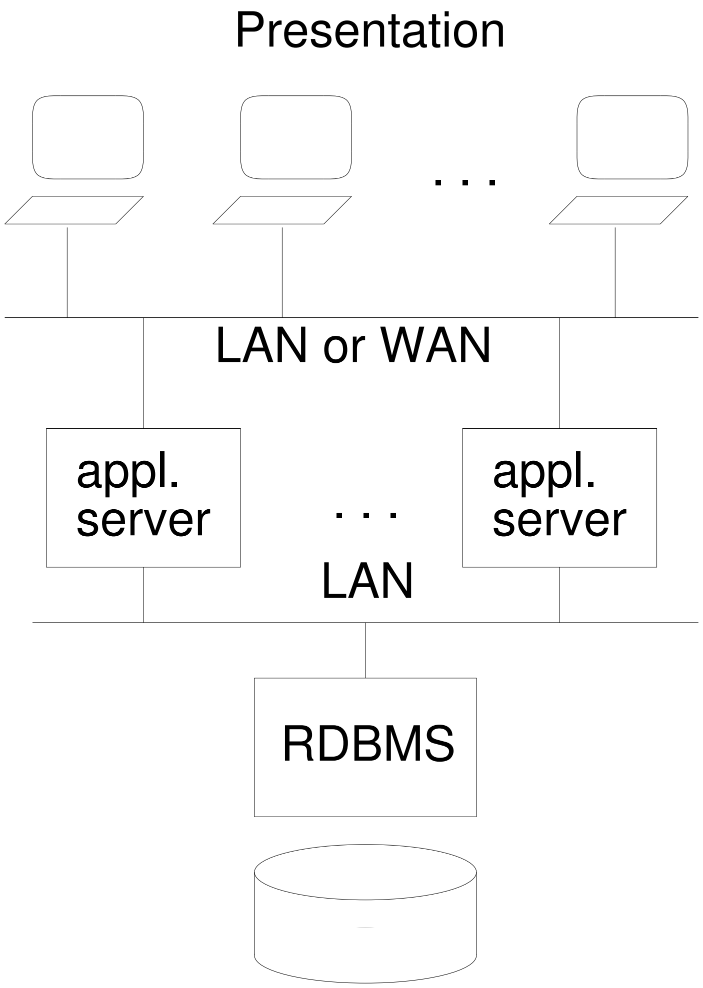
Architecture and Database Interfaces

Donald Kossmann

Outline

1. SAP's Client-server Architecture
2. Overview of the SAP Schema
3. Transaction Processing
4. Benchmarks and Results

SAP's Three-Tier Client-Server Architecture



presentation servers: control GUI

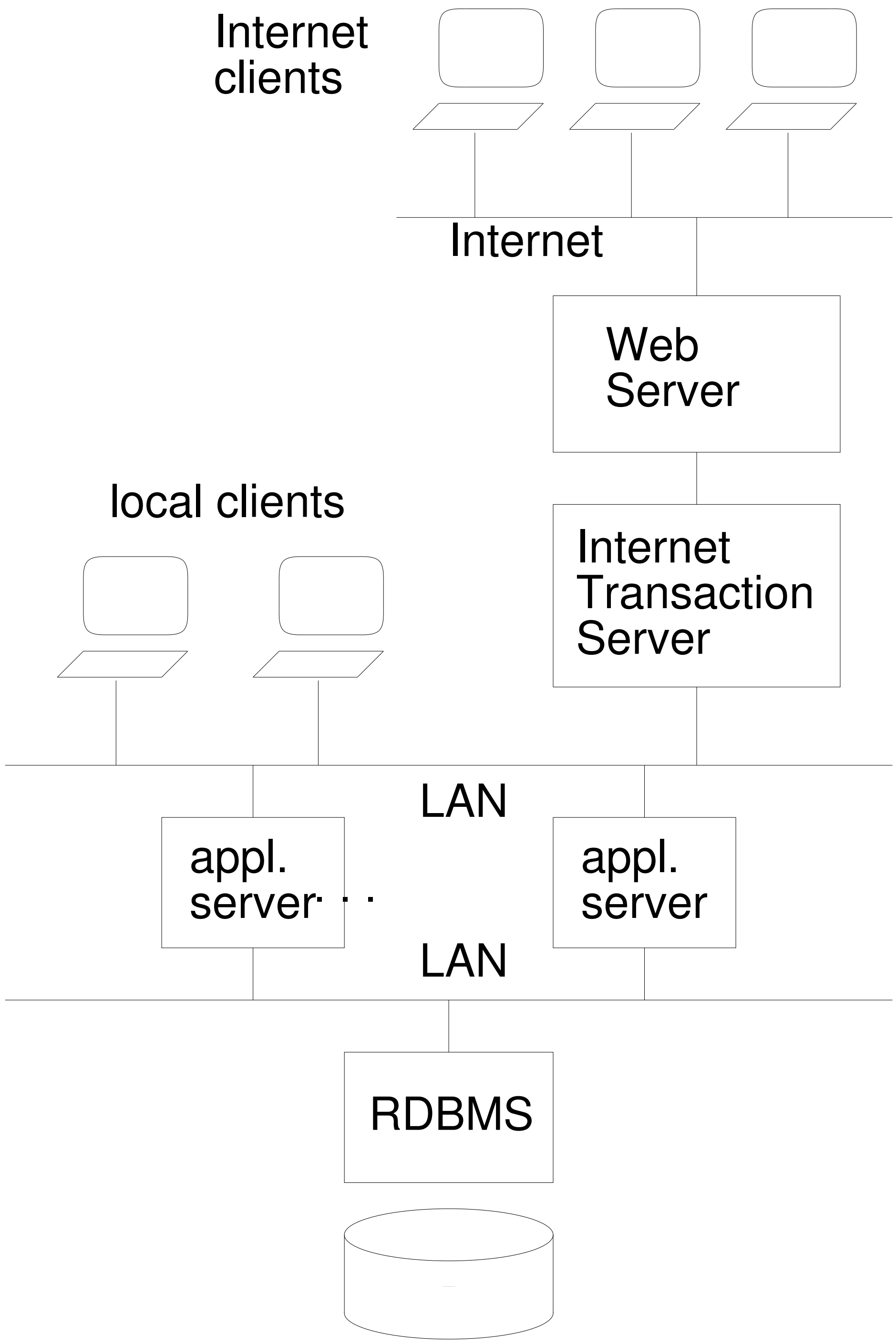
application servers: carry out ABAP programs and DynPros

RDBMS: stores all the data (including ABAP programs, DynPros, data dictionary, etc.)

Advantages of Three-Tier Architectures

1. **Scalability.** Add machines in middle tier to support more users
2. **Portability.** It is possible to use different platforms at all levels.
3. **Interoperability and openness.** Middleware serves as platform to integrate and interact with third-party products.
4. **Nice GUIs.** Presentation servers can interact with Microsoft Word, Excel, etc.

Integrating WWW Services



SAP R/3 Configurations

tiny: 1 user

- all three layers on one machine
- one of the SAP founder's golf club is run by R/3 on a laptop

small: about 10 users

- PCs for presentation
- application and database server on one (mid-range) machine
- Ethernet

Serious SAP R/3 Configurations

medium: about 100 users

- PCs, notebooks, other workstations for presentation
- a couple of machines for application servers;
- one (fairly big) database server machine;
- Ethernet for local PCs;

big: more than 1000 users

- PCs, notebooks, other workstations for presentation;
- several machines for application servers;
- a mainframe/multi-processor machine for the database
- FDDI

Additional Gimmicks

- usually, installations have separate machines for tests
- standby database server machine are recommended
- ISDN, DATEX-P, and special links to technical subsystems (e.g., sensors) are also very common

Supported Platforms

Presentation Layer

- Windows 3.1, Windows 95, Windows NT
- Java
- OSF/Motif
- OS/2
- Macintosh
- in the long run, only Windows and Java are going to be supported

Operating Systems for Application Servers

- AIX, Digital Unix, HP-UX, SINIX, SOLARIS
- Windows NT
- OS/100 (for IBM AS/400)
- OS/390 (for IBM S/390)

Database Systems

- DB2 Common Server, DB2 for AS/400, DB2 for OS/390
- Informix Online
- Oracle
- MS SQL Server
- (ADABAS D: only support for old installations)

Hardware

- Bull, IBM, SNI, SUN, Digital, HP for UNIX platforms
- many, many for Windows NT
- IBM AS/400
- IBM S/390

Communication Protocols

- TCP/IP
- LU 6.2 (for IBM mainframes)

Overview of the SAP Schema

- R/3 has more than 10,000 pre-defined tables (Version 3.x)
  - tables for data such as customers, orders, etc.
  - tables for statistics (monitoring the system)
  - tables for authorization
  - ...
  - comprehensive, generic schema for any kind of conceivable business rather than greatest common denominator
- fully normalized, (almost) no redundancy
  - good for OLTP
  - bad for OLAP (as we will see)
- users can also define their own tables

Overview of the SAP Schema (ctd.)

- three different kinds of SAP tables

  transparent: mapped 1:1 to RDBMS tables

  pool: mapped n:1 to RDBMS tables
  motivation: in the 80s, some RDBMS products limited the total number of tables

  cluster: mapped n:1 to RDBMS tables so that related tuples of several cluster tables are stored in one row of the RDBMS table
  motivation: sometimes good during transaction processing

Trend: make all tables transparent

Examples

1. All *comments* and *descriptions* need to be stored in separate tables in order to keep information in different languages:

$$\text{part}(\underline{\text{id}}, \ldots, \text{comment}) \quad \Rightarrow \quad \begin{array}{l} \text{part}(\underline{\text{id}}, \ldots) \\ \text{comment}(\underline{\text{partid}, \text{language}}, \text{comment}) \end{array}$$

N.B.: Actually, SAP tables come with names such as *KONV*, *STXL*, *VBAP*, *VBEP*, etc.

2. Generic way of dealing with pricing terms (customizable):

lineitem($\underline{\text{id}}$, ..., tax, discount) $\Rightarrow$ lineitem($\underline{\text{id}}$, ..., *pricing_terms*)
pricing_terms($\underline{\text{id, condition}}$, amount)

- tax and discount are stored in two different tuples (additional pricing_terms stored in additional tuples)
- to allow quick access, pricing_term tuples that belong to the same lineitem are clustered together (i.e., "terms" is a cluster table)

Schema: Observations

- SAP databases tend to be very large (due to genericity)
- Schema is the heart of SAP, but still under constant revision
  - a couple of thousand new tables with every new major release
  - a great deal of reorganization work with every upgrade

Transaction Processing in SAP

SAP's Transaction Concept

- SAP uses the term *Logical Unit of Work* (LUW) for transaction.
- Basically, an SAP LUW has the same ACID properties as SQL and any (SQL) database system:
  - an SAP LUW can span several dialog steps
  - an SAP LUW is either executed completely or not at all (i.e., atomicity)
  - ...
  - nested transactions are also possible

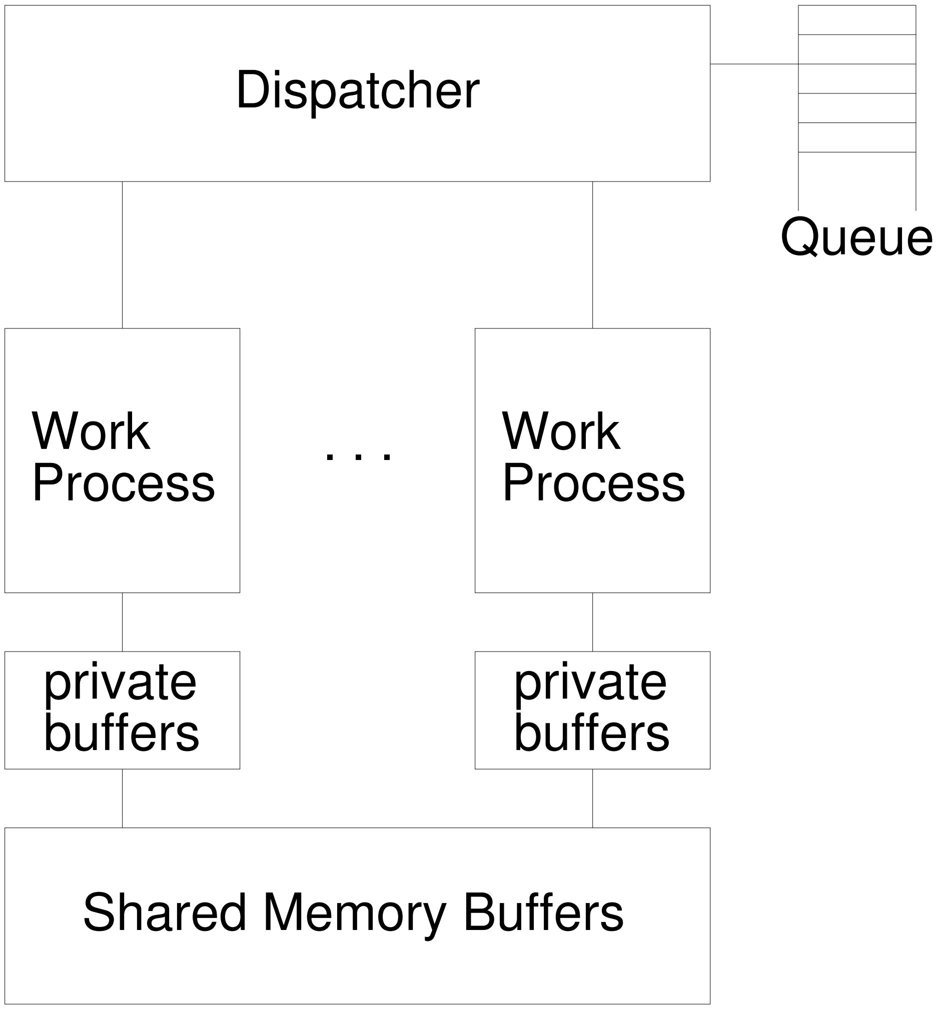
Overview of Implementation

- SAP LUWs are NOT mapped 1:1 to database transactions
- SAP implemented its own locking
  (centralized *enqueue service*)
- basically, SAP also implemented its own TP monitor
  (*message handler* and *queues* in every application server)
- online transactions and batch (overnight) queries possible
- (for comparison: PeopleSoft uses third-party TP monitors such as Tuxedo)

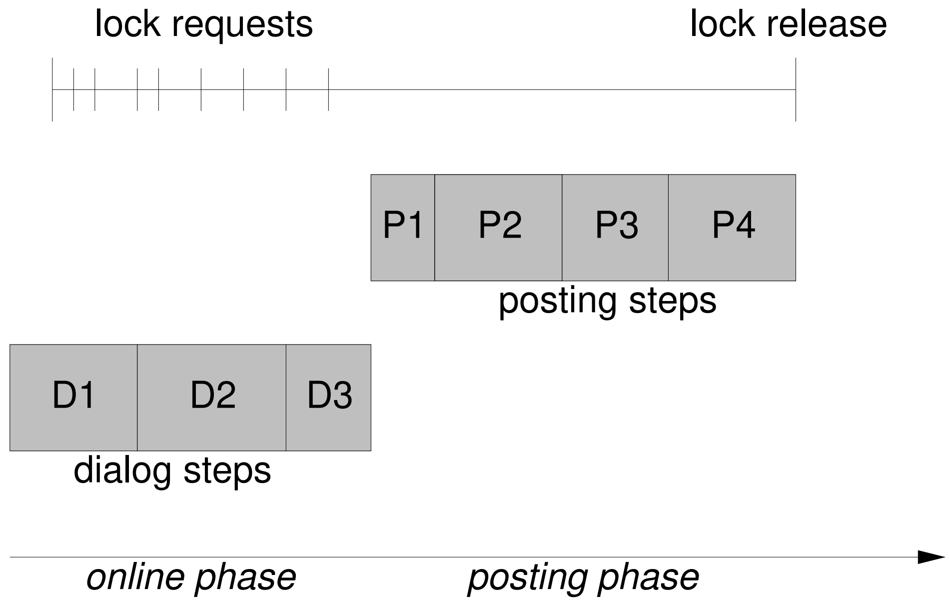
Processing Dialog Steps

1. when a user logs in, a message handler finds the application server with the smallest load (load balancing)
2. this application server handles all of the requests of that user session
3. a user session consists of several transactions, and every transaction consists of several dialog steps
4. every application server has one dispatcher process and several work processes
5. the dispatcher queues requests until a work process is available
6. a work process carries out a dialog step; rolls in relevant data and interprets DynPro and ABAP programs
7. so, every user session is handled by a single application server, but every dialog step is handled by different work processes
8. exception: transactions that involve large objects have exclusive work processes to avoid cost of rolling data in and out

Application Server



2-Phase Processing of SAP Transactions



- log records for updates are generated as part of every dialog step
- log records are propagated to RDBMS in posting phase
- locks are requested during online phase and released after the posting phase is complete (2-phase locking)
- like dialog steps, posting steps are handled by different work processes (potentially in parallel)

Why doesn't SAP directly use the RDBMS?

- Typical RDBMSs do not allow transactions that cross process boundaries.
  (This is necessary in SAP because the dialog steps of an LUW can be handled by different work processes)

- aborts are quite frequent (e.g., out of stock) and only carried out before posting
  as a result, no roll-back at the RDBMS, the bottleneck, is required for aborts

- SAP carries out locking in the granularity of "business objects" which are defined in the ABAP dictionary

Caching

Overview

- SAP's application servers cache ABAP programs, constraints, and operational data in order to reduce the load of the RDBMS.
- more than 90% cache hits are not unusual for SAP applications
- ABAP programs and constraints are always cached.
- Administrator decides which data to cache and in which way to cache it:
  - data that is frequently updated should not be cached at all
  - huge data that is likely to flood the cache should not be cached either
  - default settings for pre-defined tables help

Caching Examples

- *REGION* table is a classic candidate for caching
- *LINEITEM* table is a classic candidate for NOT caching

Caching Alternatives

Application servers can cache data three different granularities (set by default or system administrator for every table):

1. *complete caching of a table*
   - cache can be used for any query on that table
   - need much cache space, high cost to propagate updates
2. *tuple-wise caching*
   - cache only used for `select single` statements
   - fine-grained: i.e., good cache utilization, low cost in the presence of updates
3. *generic caching (a compromise)*
   - cache all tuples of a table with the same value of a prefix of the primary key
   - e.g., cache all tuples that belong to the same business unit

Cache consistency in configurations with several application servers

- periodic propagation
- no guarantee for serializability
- however, not a problem because only data that is almost never updated is cached or data for which inconsistencies don't matter that much

Authorization

- again, SAP implements its own authorization model and does not use the standard (SQL) model supported by the RDBMS
- users must log in with their user-id and password (identification)
- fine-grained and flexible authorization concept
  - individual fields of tables
  - specific transactions and/or reports
  - views
- bundling of authorizations
  - authorization object (set of related authorizations)
  - authorization profile (set of authorization objects; roles)
  - group profile (set of authorization profiles)
- users belong to groups and inherit group authorizations

Security

- activity logs
- encryption of all messages sent over the network
- secure transport system for batch input and migration of databases
- Kerberos and SecuDE for secure clients at the presentation layer
- only the administrator has direct access to the database and file system; everybody else must use the interfaces of the presentation layer or other external services in order to work with the system

Performance and System Monitoring

- SAP monitors the following parameters
  - queue lengths in dispatcher
  - cache hit ratio
  - database operations (scans, sorts, joins)
  - ABAP operations (sorts, etc.)
  - number of commits and rollbacks
  - CPU, disk, memory, network utilization
  - response time of dialog steps, work processes, database calls
- alerters inform administrators if problems occur
- performance statistics are stored in the database
- Early Watch service — (Big Brother is Watching You)

SAP Transaction Processing Benchmarks

- defined in 1993 (Release 1.1H)
- purpose of these benchmarks:
  - sizing of an SAP system
  - measure real SAP application operations
  - key for database systems certification
- seven dialog benchmarks (FI, MM, SD, PP, PS, WM, ?)
- two batch benchmarks (AA, HR)
- most popular benchmark is SD benchmark
- usually run by hardware vendors
  (SAP benefits from their competition)

The SD Benchmark

(Sales and Distribution)

Set Up (common to all benchmarks)

- one presentation server simulates users of several business units
- to avoid lock contention, maximum 100 users per business unit
- central configuration: one machine for application server and RDBMS
- distributed configuration: n machines for application servers, another machine for RDBMS

SD Benchmark Skript

1. create an order with five line items
2. create a delivery for this order
3. display the customer order
4. change the delivery and post goods issue
5. list fourty orders
6. create an invoice

15 dialog steps; 4 posting steps; 150 secs think time

Disclosure Report (PUBLIC)

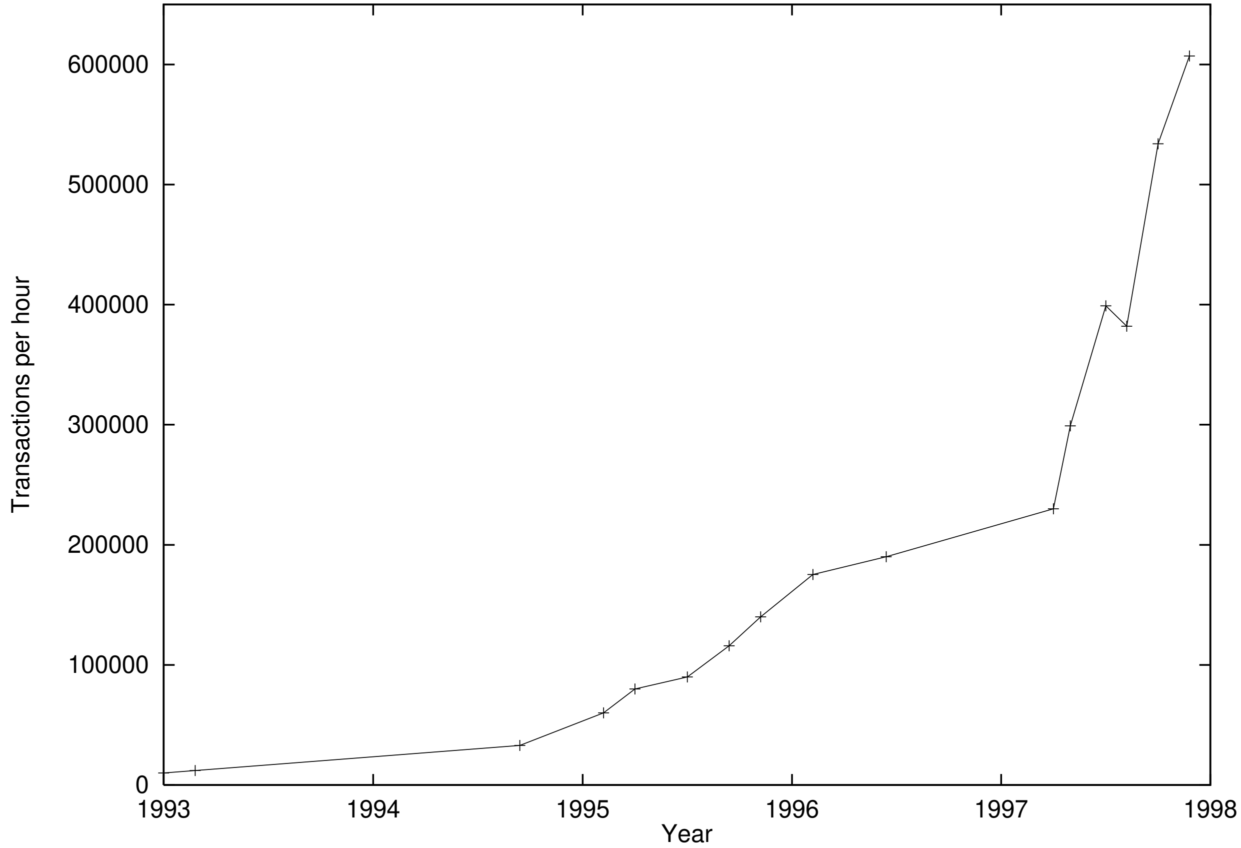
SUN Beaverton, CA, USA on March 11, 1997:

Number of benchmark users:	1,410 SD
Average dialog response time:	1.85 seconds
Throughput:	7,133 SAPS (428,000 dialog steps/h)
Equivalent to:	143,000 processed order line items per hour
Average DB request time:	0.397 (dialog), 0.397 (update)
CPU utilization:	94%
Operating System:	Solaris 2.5.1
RDBMS:	Informix Online Server 7.21 UC1
R/3 Release:	3.0E (with 3.0F kernel)
Total disk space:	350 GB

1 Central Server: 64-way UltraSparc II CPU's 250 MHz, 14 GB main memory, 1 MB level 2 cache; 15 dialog/update instances, 1 message/enqueue instance

Certification Number: 1997007

SD Benchmark Results and Trends



hardware vendors have solved many of SAP's performance problems concerning transaction processing.

References

- R. Buck-Emden and J. Galimow: SAP R/3 System – A Client/Server Technology. Addison-Wesley, 1996.
- J. Doppelhammer, T. Höppler, A. Kemper, and D. Kossmann: Database Perfromance in the Real World – TPC-D and SAP R/3. ACM SIGMOD, 1997.
- B. Lober and U. Marquard: R/3 System Standard Application Benchmarks. `http://www.sap-ag.de/products/techno/index.htm`
- B. Lober: R/3 System Standard Application Benchmarks – Published Results, Version 2.4. `http://www.sap-ag.de/products/techno/index.htm`
- SAP AG: Online Documentation. 1996-1998.
- SAP AG: SAP R/3 Software Architektur. 1994.

Queries and OLAP Applications on SAP R/3

Alfons Kemper

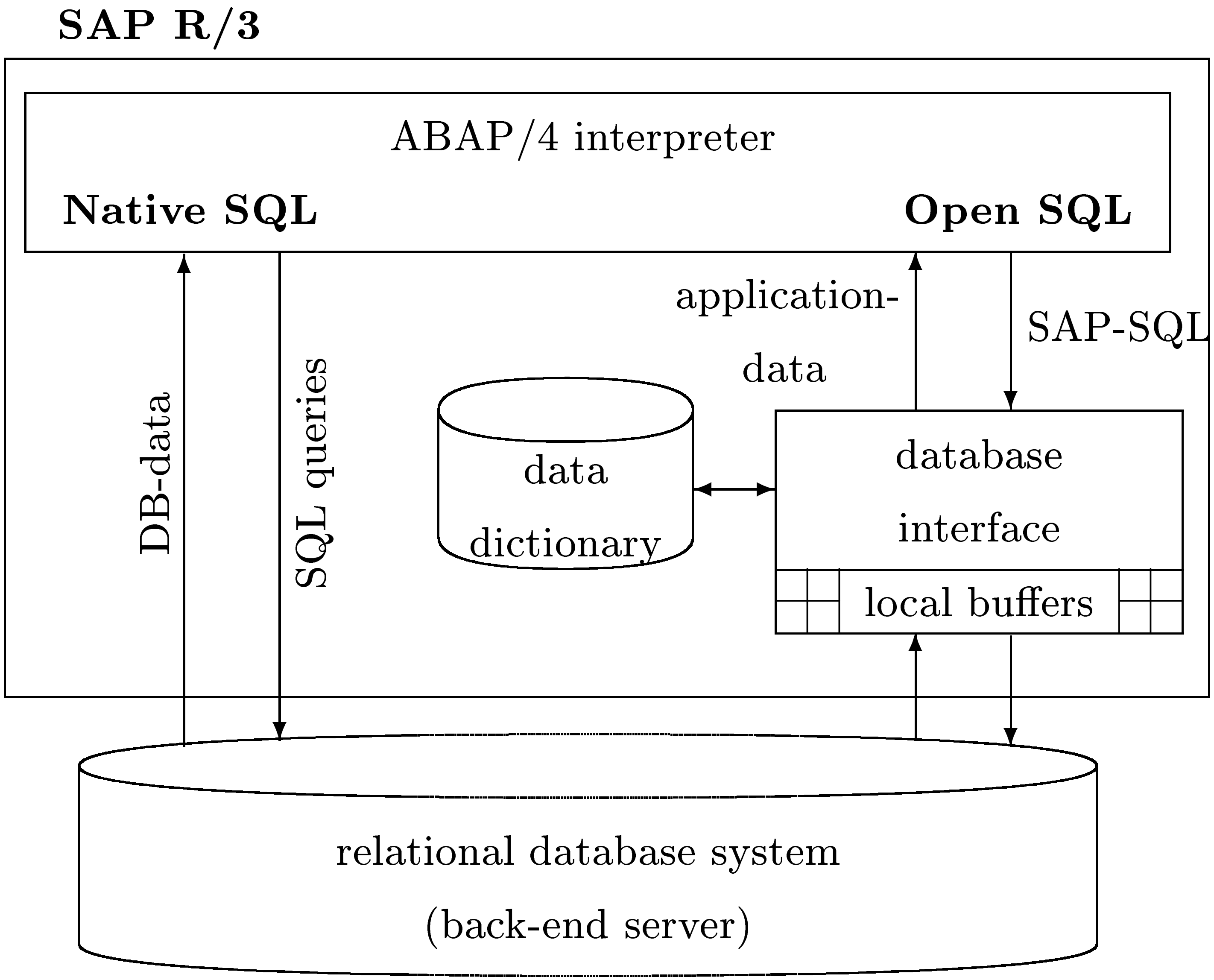
Outline

- Query interfaces of SAP R/3
- Decision Support Queries on SAP R/3
- SAP Data Warehouse Products

Querying/Reporting in SAP R/3

- all business data is entered via OLTP-applications into SAP R/3
- SAP has its own predefined schema
- SAP R/3 is used to ask queries, e.g.:
  - for finding particular information as part of OLTP processing: *what happened to a particular order issued by customer 'Smith' in March 1998*
  - for generating business reports for decision support: *how did our new line of products sell in different regions over the last six months*
- SAP R/3 has many predefined business reports that can be invoked “by the click of the mouse”
- SAP R/3 also allows to “program” customized queries/reports

Query Interfaces of SAP R/3



Open SQL versus Native SQL

Open SQL

- SQL-92 style syntax (some limitations!)
- built-in operators, such as SORT
- materialize temporary results
- caching of query results
- cursor caching to optimize repeated database access
- in recent versions SAP has significantly improved the database interface
- before Version 3.0 it was not possible to “push” a join to the database server; it had to be implemented as nested SFW’s within the ABAP-processor (i.e., within the application server).

Open SQL versus Native SQL–cont'd

Native SQL

- bypass ABAP/4 processor (and caching)
- bypass SAP data dictionary
- therefore not possible to access non-transparent tables
- advantage: often better performance
  - exploit (non-standard) features of the RDBMS
  - avoid certain problems of the ABAP processor
- disadvantage: not portable and potentially unsafe
  - cannot use different RDBMSs
  - programmer needs expert knowledge

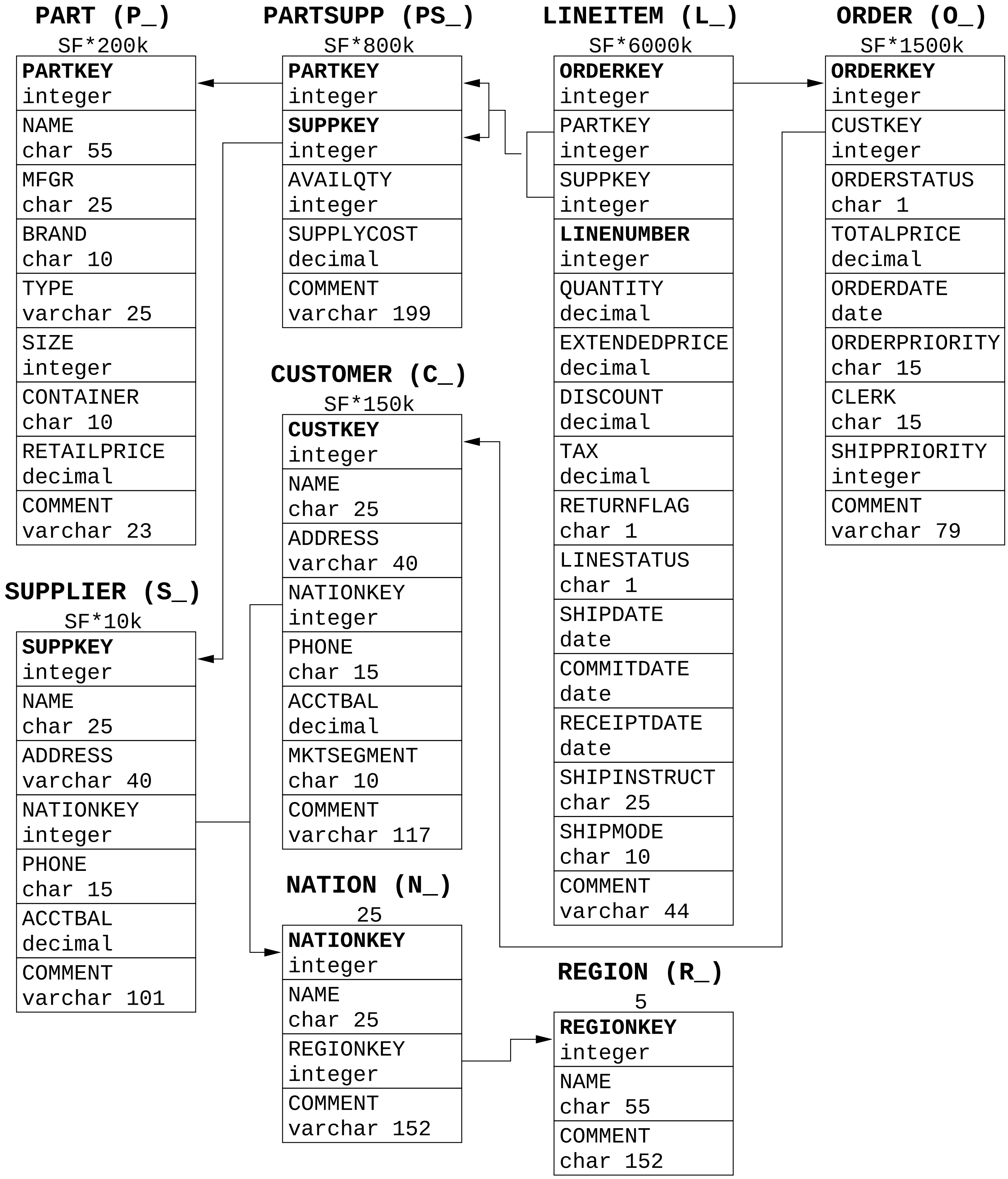
$\Rightarrow$ SAP recommends use of Open SQL

  - all built-in business applications use Open SQL

Assessment of Query Processing in SAP R/3

- use of the TPC-D Benchmark
- results were reported in: Jochen Doppelhammer, Thomas Höppler, Alfons Kemper, Donald Kossmann: Database Performance in the Real World - TPC-D and SAP R/3 (Experience Paper). SIGMOD Conference 1997: 123-134
- load data into SAP's predefined tables
- wrote 17 TPC-D queries in a) Native SQL and b) Open SQL
- implemented 2 update functions using SAP R/3's batch input facility
- Sun Sparc Station with 2×60MHz processors, 256 MB main memory, 5×4GB disks, commercial RDBMS as back-end
- for comparison: we implemented the "original" TPC-D benchmark directly on commercial RDBMS using the same hardware and the same configuration

The TPC-D Database Schema



SAP Tables used in the TPC-D Benchmark

- **8** tables in the TPC-D schema versus **17** tables in SAP R/3

SAP Table	**Description**	**Orig. TPC-D Tab.**
T005	Country: general info	NATION
T005T	Country: Names	NATION
T005U	Regions	REGION
MARA	Parts: general info	PART
MAKT	Parts: description	PART
A004	Parts: terms	PART
KONP	Terms: positions	PART
LFA1	Supplier: general info	SUPPLIER
EINA	Part-Supplier: general info	PARTSUPP
EINE	Part-Supplier: terms	PARTSUPP
AUSP	properties	PART, SUPP, P.-S.
KNA1	Customer: general info	CUSTOMER
VBAK	Order: general info	ORDER
VBAP	Lineitem: position	LINEITEM
VBEP	Lineitem: terms	LINEITEM
KONV	Pricing Terms	LINEITEM
STXL	Text of comments	all

Size of Database in KB (SF = 0.2)

	Original TPC-D DB	**SAP DB** (Version 2.2)
Data	271,139	2,813,216
Indexes	102,822	841,008

Why is SAP database so big?

- SAP keeps info which is not modelled in the TPC-D benchmark
  - given default values in our experiments
- (strong) partitioning of the database tables
  - space to store foreign keys
- in SAP, keys (e.g. Customer-No) are 16 byte strings.
- in original TPC-D benchmark, keys are 4 byte integers.

Loading Time for the Benchmark Data (SF = 0.2)

	Loading Time
REGION	—
NATION	—
SUPPLIER	18m
PART	15h 56m
PARTSUPP	30h 24m
CUSTOMER	7h 33m
ORDER+LINEITEM	25d 19h 55m

- in all, it took about **1 month!**
- Reasons:
  - SAP carries out extensive consistency checks for every record individually
  - SAP R/3 inserts records into the database one-at-a-time rather than making use of the RDBMS bulk load facilities
- it appears to be very difficult (impossible?) to bypass SAP R/3 and insert *R/3-consistent* data into the RDBMS tables directly

TPC-D Power Test

Query Update	**RDBMS (TPC-D-DB)**	**Native SQL (SAP DB)**	**Open SQL (SAP DB)**
Q1	6m 09s	58m 59s	56m 18s
Q2	53s	3m 09s	34s
Q3	4m 03s	9m 02s	11m 51s
Q4	1m 45s	6m 18s	6m 38s
Q5	6m 39s	14m 42s	37m 27s
Q6	1m 20s	7m 28s	14m 06s
Q7	9m 03s	23m 05s	29m 24s
Q8	1m 54s	19m 04s	16m 37s
Q9	8m 42s	31m 33s	1h 7m 14s
Q10	5m 18s	33m 06s	57m 49s
Q11	5s	4m 37s	2m 23s
Q12	3m 15s	9m 48s	9m 36s
Q13	8s	19s	25s
Q14	6m 23s	10m 25s	21m 54s
Q15	3m 25s	13m 51s	28m 31s
Q16	13m 24s	3m 16s	3m 22s
Q17	11s	1m 50s	2m 13s
UF1	1m 40s	1h 46m 54s	1h 46m 54s
UF2	1m 48s	11m 35s	11m 35s
Total (quer.)	1h 12m 37s	4h 10m 32s	6h 06m 22s
Total (all)	1h 16m 05s	6h 09m 01s	8h 04m 51s

- more details (all queries in Open and Native SQL):

http://www.db.fmi.uni-passau.de/projects/SAP/

Why is SAP query processing more expensive?

- three (native SQL) to five (Open SQL) times more expensive queries
  - ten times larger database (due to partitioning, large keys, default values)
- partitioning of the data leads to $N \times M$-way joins versus M-way joins in original TPC-D benchmark
  - Example: Query 1
  - TPC-D: single table scan
  - SAP: Lineitem $\bowtie$ Lineitem $\bowtie$ Lineitem
- updates are very expensive because of the consistency checks; just as for initial loading

Why is Open SQL even more expensive?

- Limitations of Open SQL prohibit "pushing down" certain expensive operations, e.g., complex aggregation
  - increased communication costs between database server and application server
- poor query evaluation plans
  - partly due to RDBMS
  - partly due to SAP R/3
- cannot exploit non-standard features of the RDBMS
  - for example proprietary string manipulation routines
- SAP generates parameterized queries rather than passing constants of selection predicates to the RDBMS optimizer
  - may result in poor optimization choices by the optimizer
  - e.g., choosing an index scan instead of a table scan because the selectivity of a predicate is not known due to param.

Conclusions

- Complex queries for decision support are too expensive in SAP R/3
- The highly partitioned schema is optimized for OLTP applications, not for OLAP applications
- (Extensive) query processing penalizes the operational OLTP system

$\Rightarrow$ Data Warehouse is the way to go

SAP EIS (Executive Information System)

- SAP's first attempt in data warehousing
- part of the regular R/3 system
- uses the same system (database server and application server) as the OLTP applications
- therefore, incresed load on the OLTP system
- based on a rudimentary star schema
  - a fact table
  - several dimension tables
- however, SAP EIS has limitations: e.g., only three attributes per dimension table

SAP EIS—cont'd

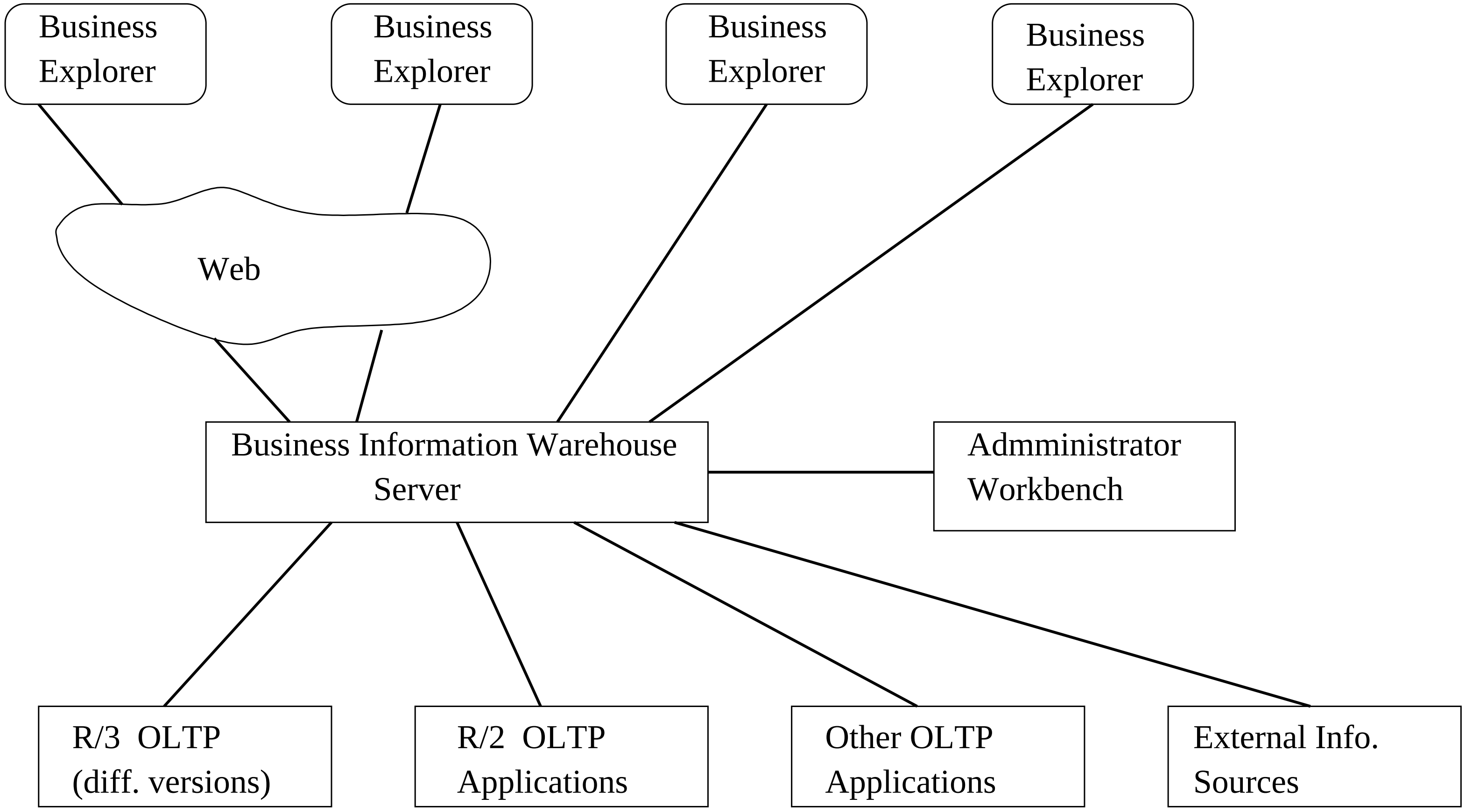
- limitations in query processing
  - only 8 of the 17 TPC-D queries could be expressed in SAP EIS
- the performance of these eight queries improved by a factor of two as compared to the Open SQL reports
- high loading time: about 50 hrs to load the TPC-D database with SF=0.2 from R/3 into EIS
  - even though, EIS was part of the same R/3 system
- merely extracting the TPC-D database from SAP R/3 into ASCII files takes 6 hours

The Business Information Warehouse (BIW)

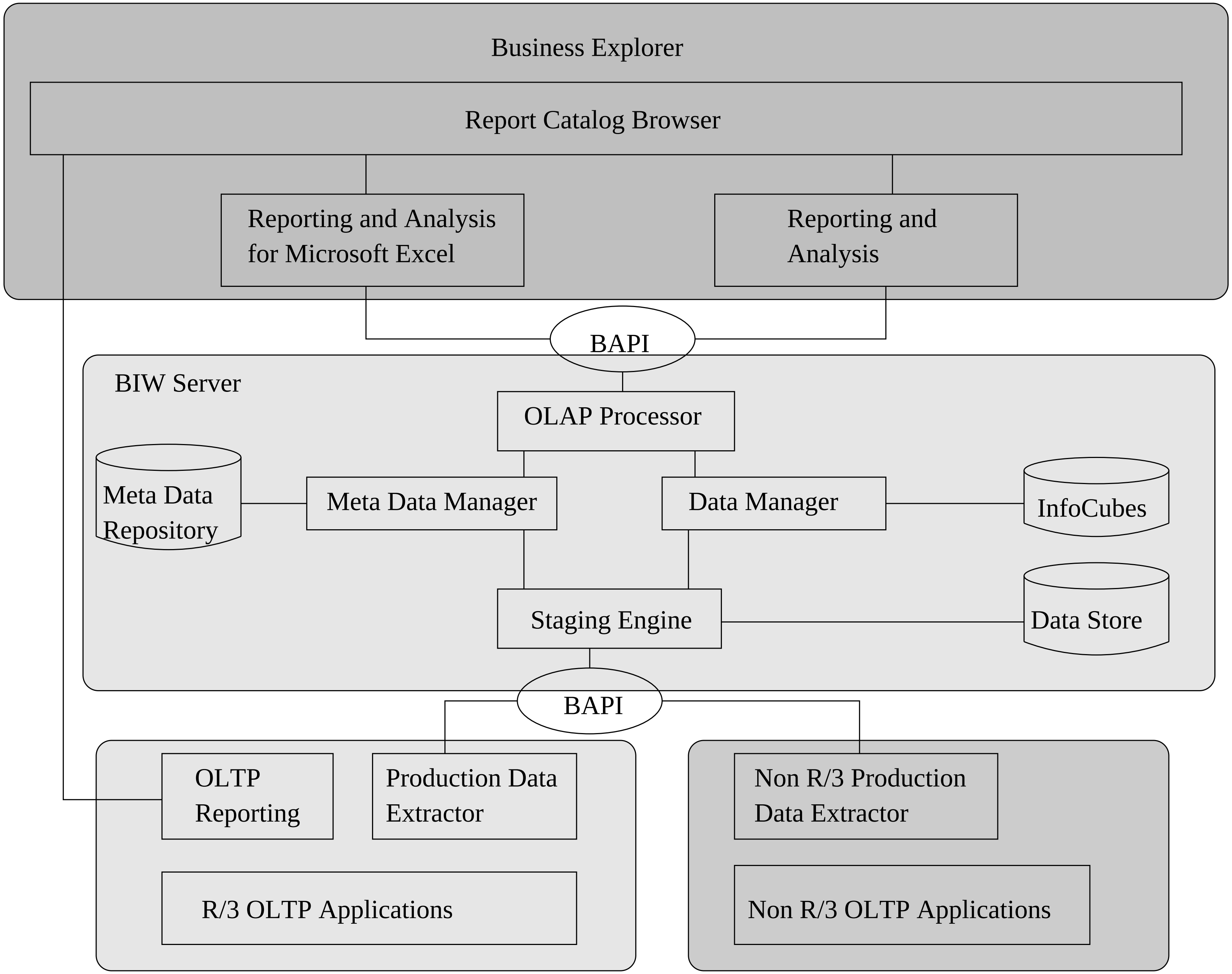
- SAP's new data warehouse product
- first beta tests; general availability starting in July
- ROLAP architecture (Relational OLAP)
- RDBMS independent, i.e., several RDBMS products are supported as back-end server (developed on Oracle and MS SQL Server)
- bundled with the SAP R/3 system, Version 4.0
- but can also be acquired separately as a stand-alone system
- first SAP stand-alone component (outside of R/3)
- part of SAP's Business Framework

  $\leadsto$ componetization of R/3
- BIW can handle data extraction from various other systems, not only from SAP R/3 and R/2

The Business Information Warehouse's Role



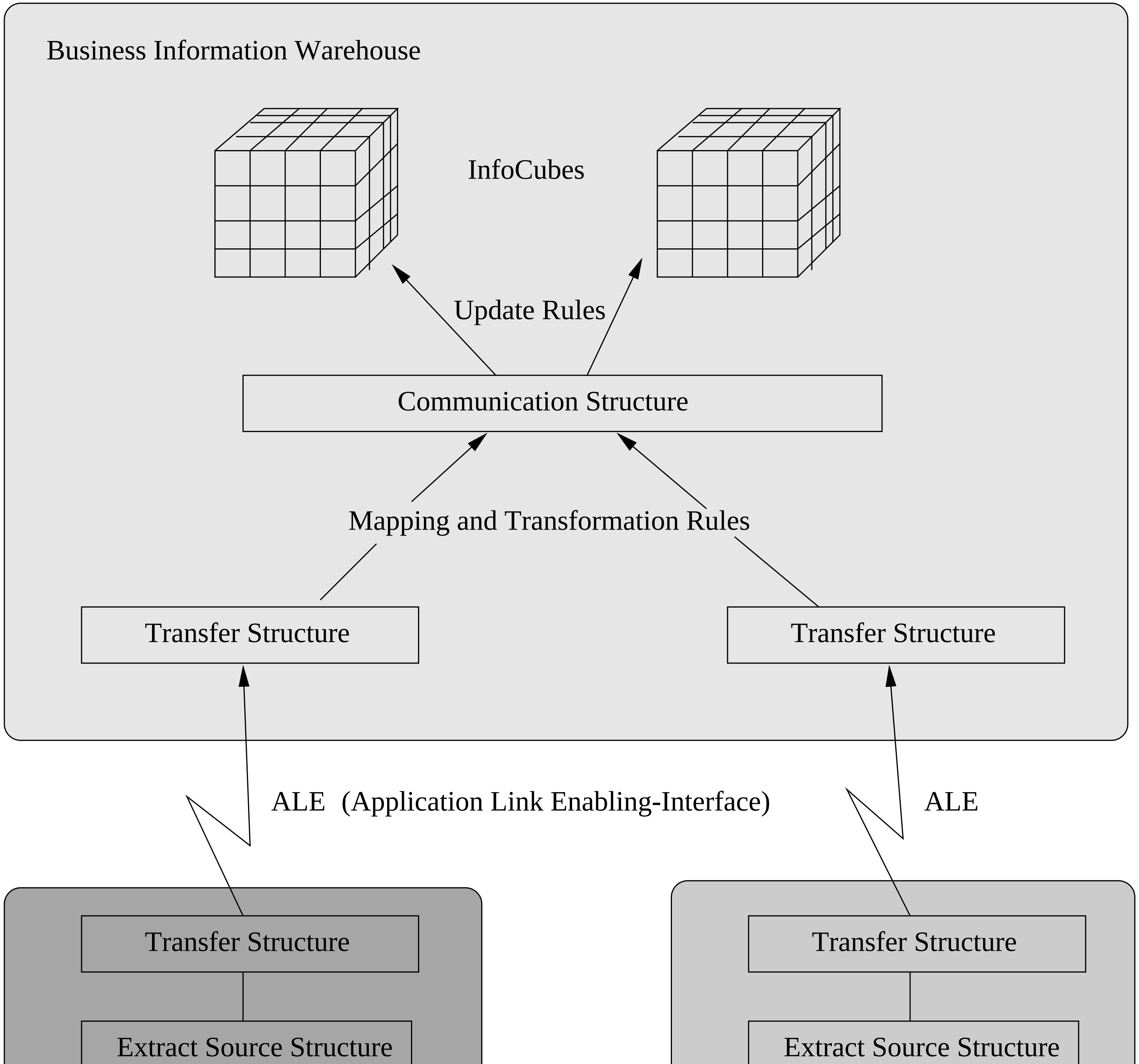
The BIW's Architecture



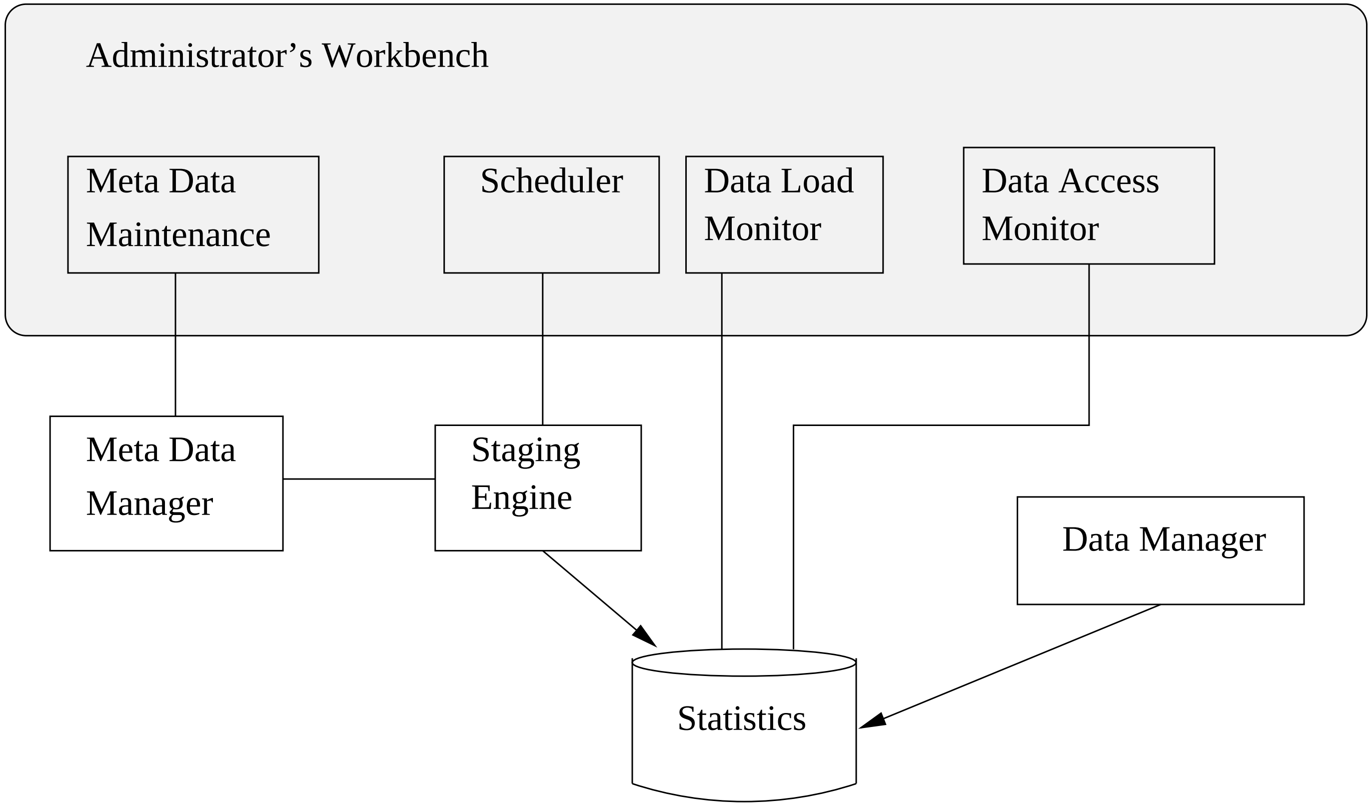
Key Features of the BIW Server

- open Business Application Programming Interfaces (BAPIs) for
  - data loading: to accommodate the extraction of data from non-SAP systems
  - OLAP processing: e.g., for third party visualization tools
- pre-configured meta data repository: InfoCubes catalog, report catalog, information source catalog
- many pre-defined InfoCubes for common business applications: e.g., market segment analyses, profitability analyses, stock inventory analyses, corporate indicator systems
- predefined extraction routines to obtain this data from R/3 OLTP systems (with incremental updates!)

The Staging of Information

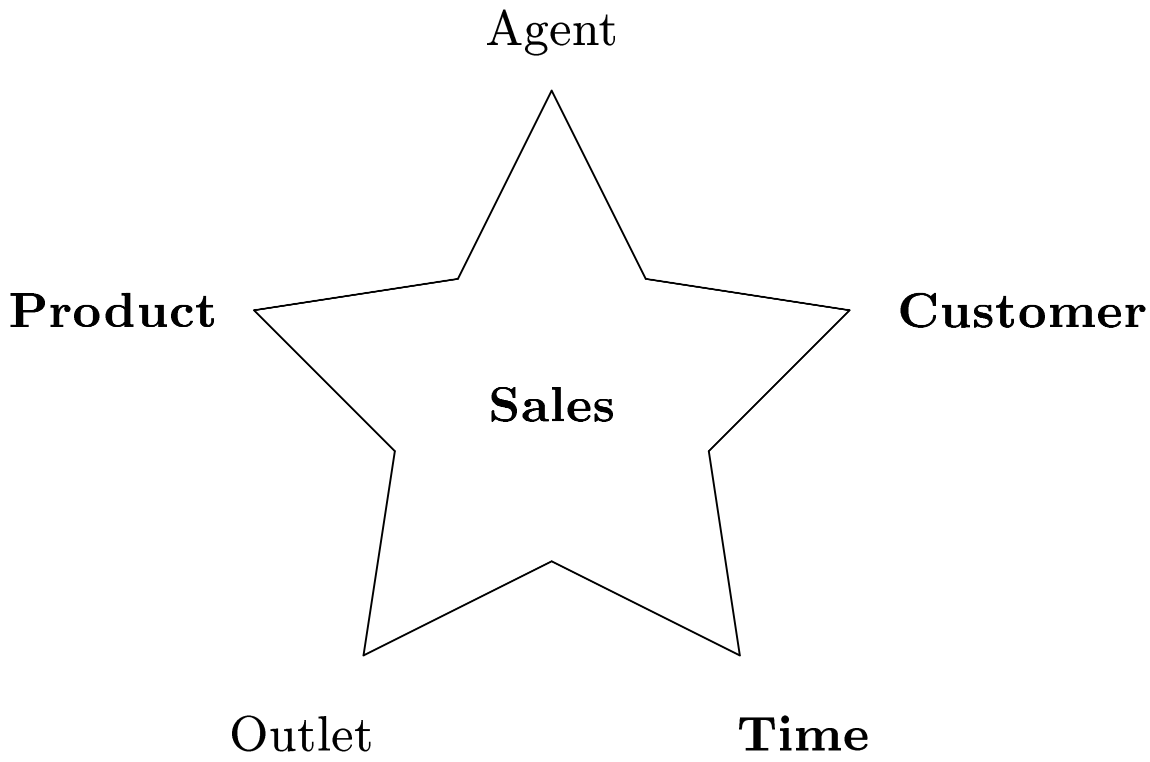


The Administrator's Workbench



- GUI-based
- sophisticated scheduler

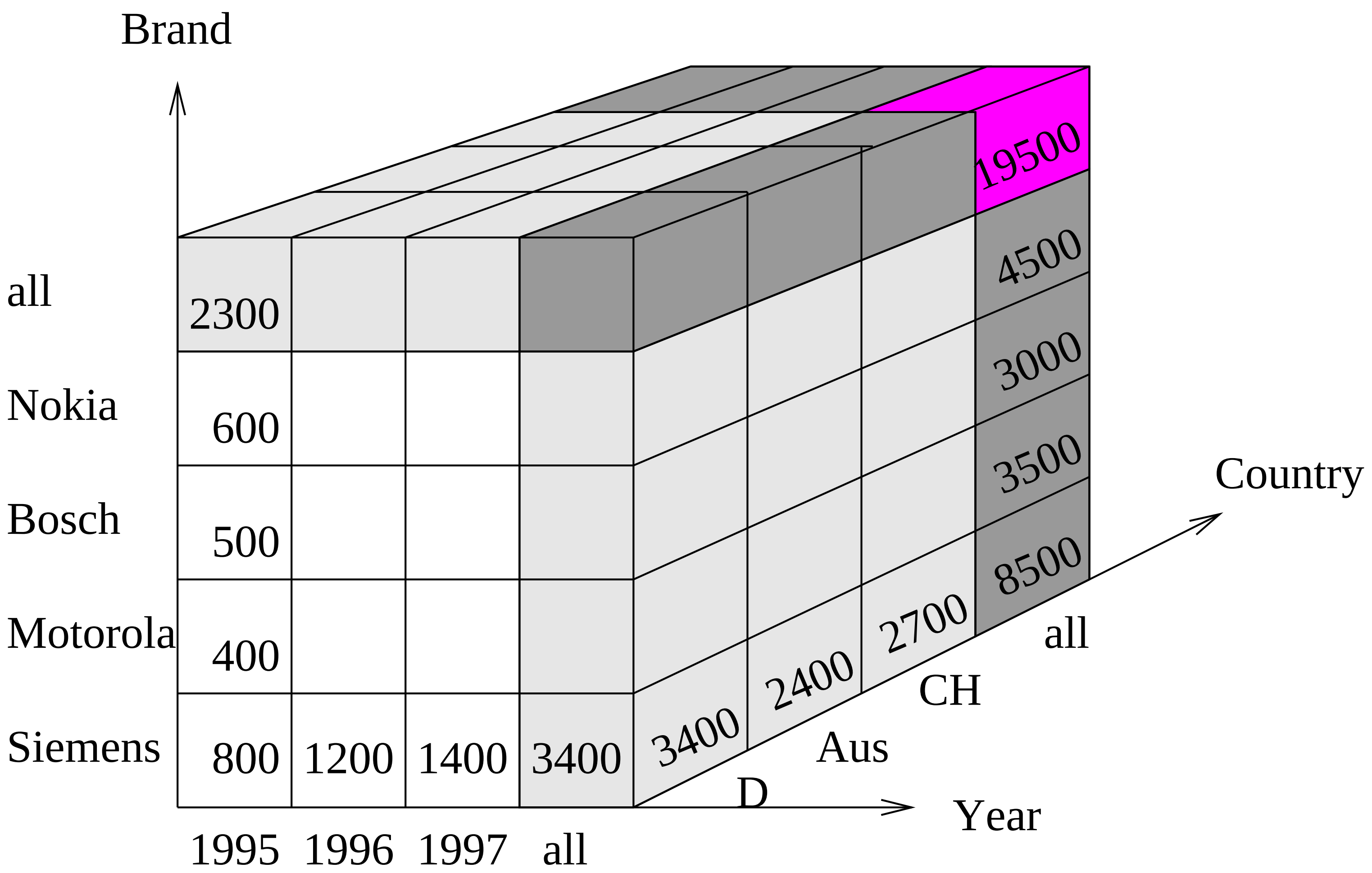
InfoCubes: Star Schema



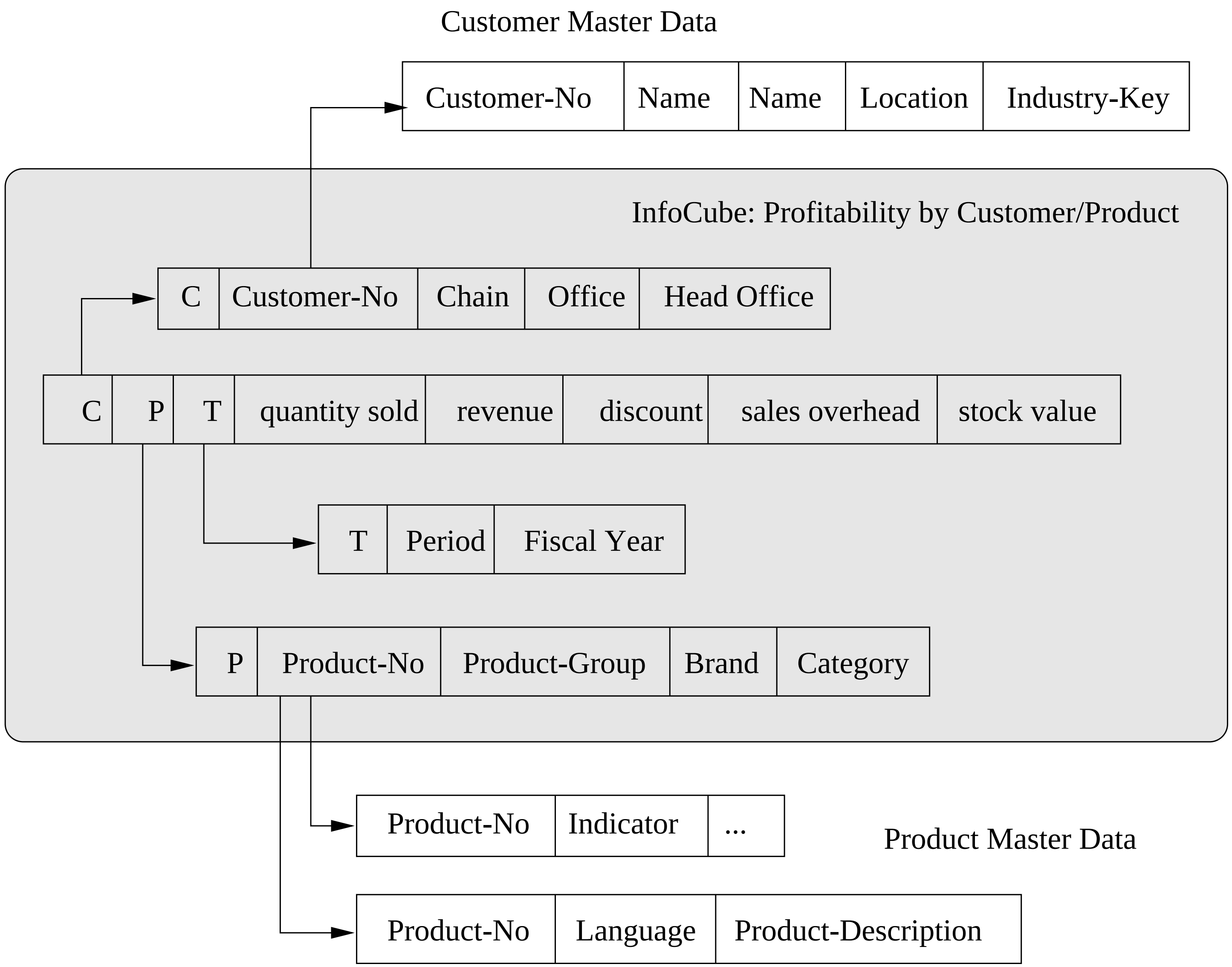
- one normalized fact table
- several dimension tables
  - possibly non-normalized to model hierarchical characteristics
  - e.g., customers' city→ county→ country→ region

Cube Representation

- (Handy) Sales by Brand/Year/Country

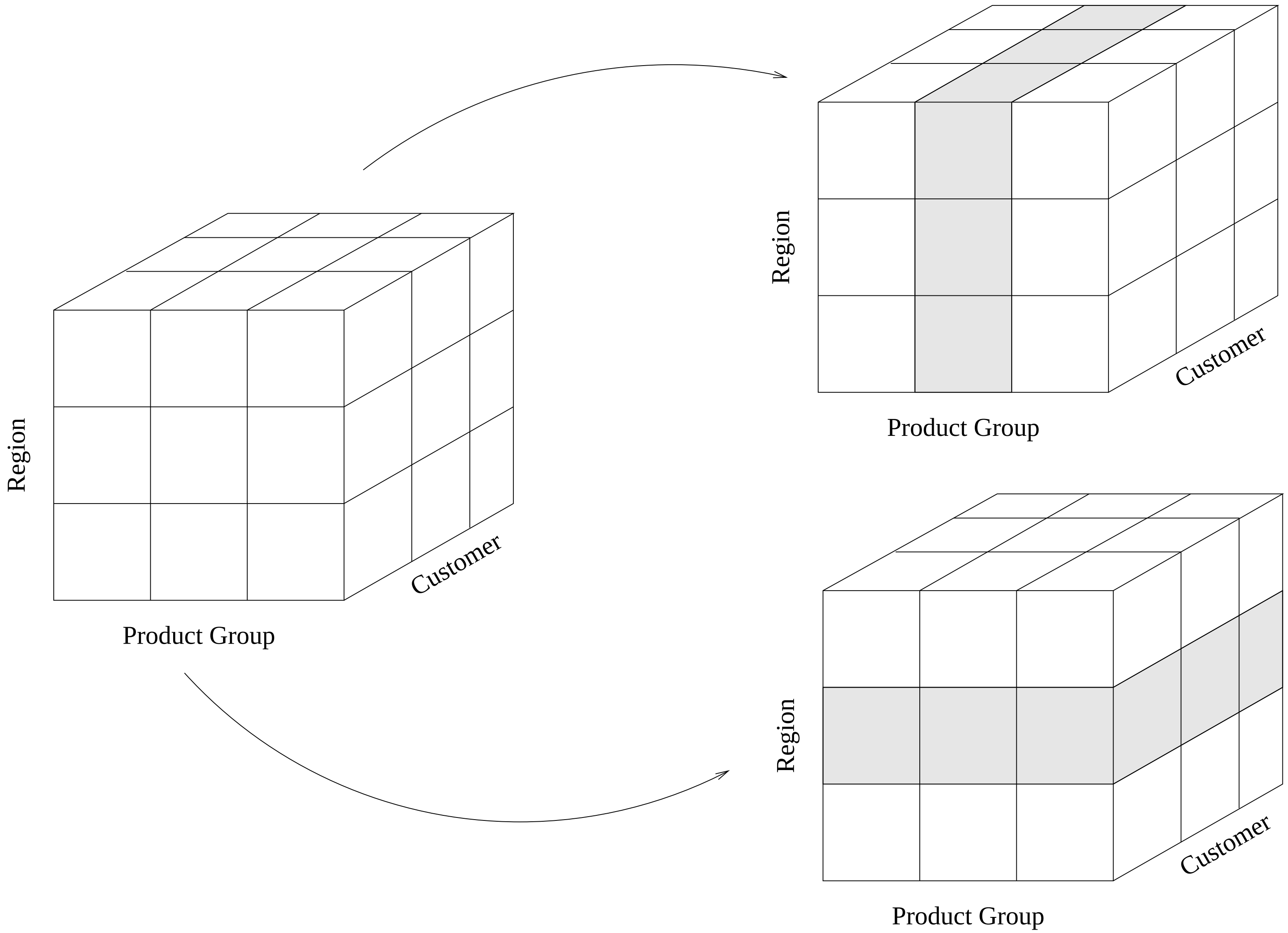


Relational Representation of an "InfoCube"



Flexibly Exploring Business Data

- slice and dice
- drill down/roll up



Business Analysis Library

- aggregate functions: sum, min, max, average, ...
- comparison functions: difference, ratio, percent,, share, correlation, ...
- sequence functions: top/last n, cumulative sum, tertiles, quartiles, ABC analysis, ...
- exception conditions: absolute values, top/last n, top/last n%, trends, ...
  - used to highlight exceptions
- financial functions: currency conversion, business period conversion, ...
  - very important for international enterprises

Performance

- no benchmark results yet
- dedicated server
  - separate from OLTP server
  - can be configured as a three-tier client server configuration (like SAP R/3)
- InfoCube summary levels: precomputed aggregations
- persistent report cache
- batch reports

Literature

- Jochen Doppelhammer, Thomas Höppler, Alfons Kemper, Donald Kossmann: Database Performance in the Real World - TPC-D and SAP R/3 (Experience Paper). SIGMOD Conference 1997: 123-134 `http://www.db.fmi.uni-passau.de/projects/SAP/`
- SAP-AG: SAP Business Information Warehouse – Information Center. `http://www.sap-ag.de/products/biw/index.htm`
- Rüdiger Kretschmer and Wolfgang Weiss, SAP's R/3 Applications with ABAP/4. Sybex, 1996, Alameda, CA, USA.
- Transaction Processing Performance Council TPC, TPC Benchmark D (Decision Support), Standard Specification 1.0, Transaction Processing Performance Council (TPC), 1995, `http://www.tpc.org/`