Implementation of Databases

Chapter 1: Architectures of Database Systems

Winter Semester 2021/2022

Lecture	Prof. Dr. Sandra Geisler
Excercises	Soo-Yon Kim, M.Sc.
	Arnab Chakrabarti, M.Sc.

1.1 Goals and Tasks of DBMS

Learning Goals

At the end of this section you will be able to

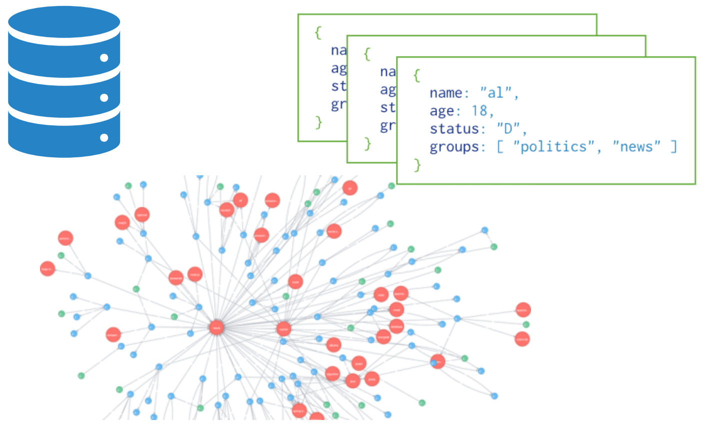
- ✓ explain, what is DBS and what is a DBMS
- ✓ name and describe the goals and tasks of a DBMS
- ✓ discuss the relationship between software applications and DBMS

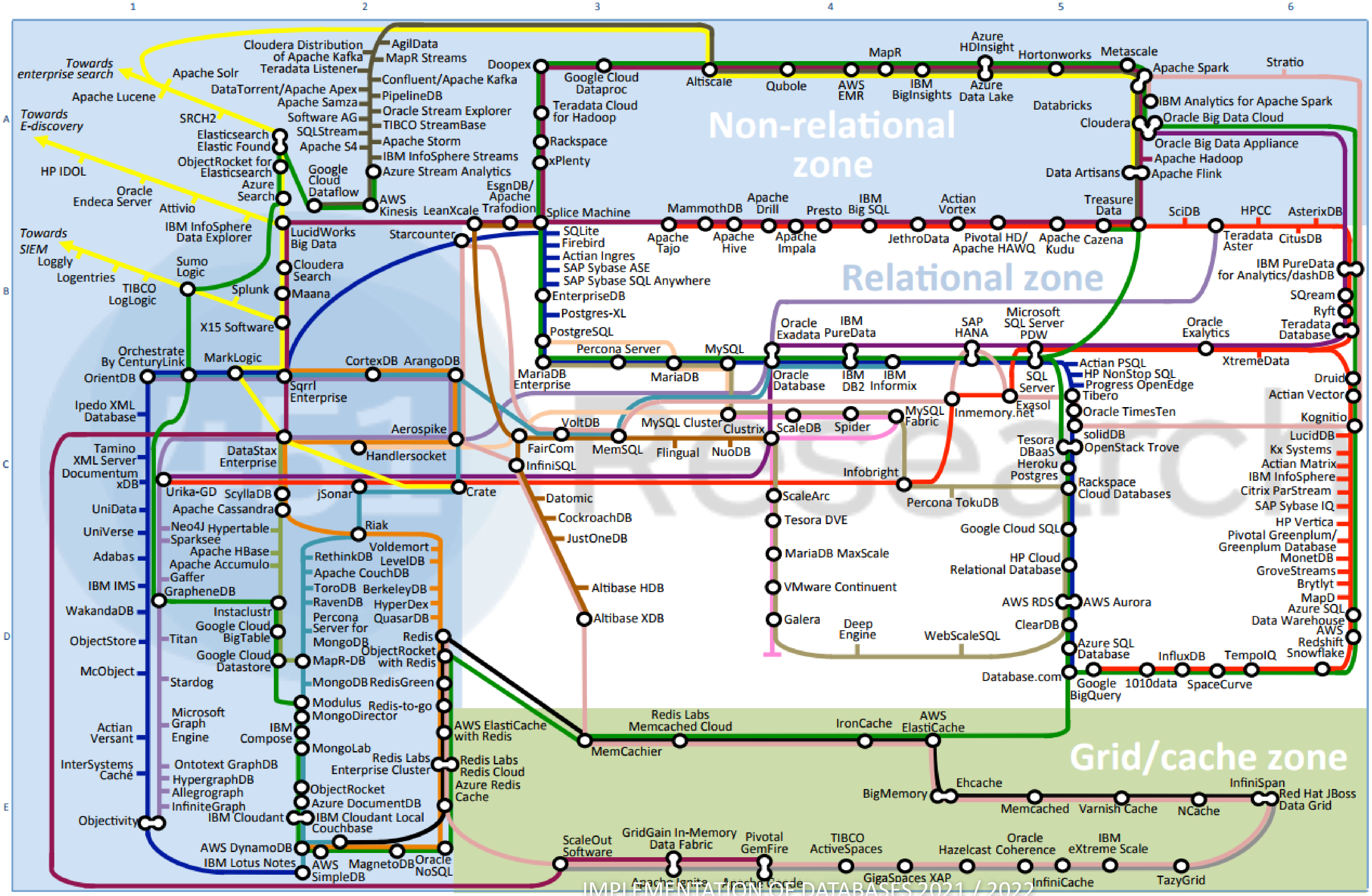
What is a Database System?

Database Management System (DBMS)
software to manage databases & data

+

Database
self-describing collection of related data





451 Research

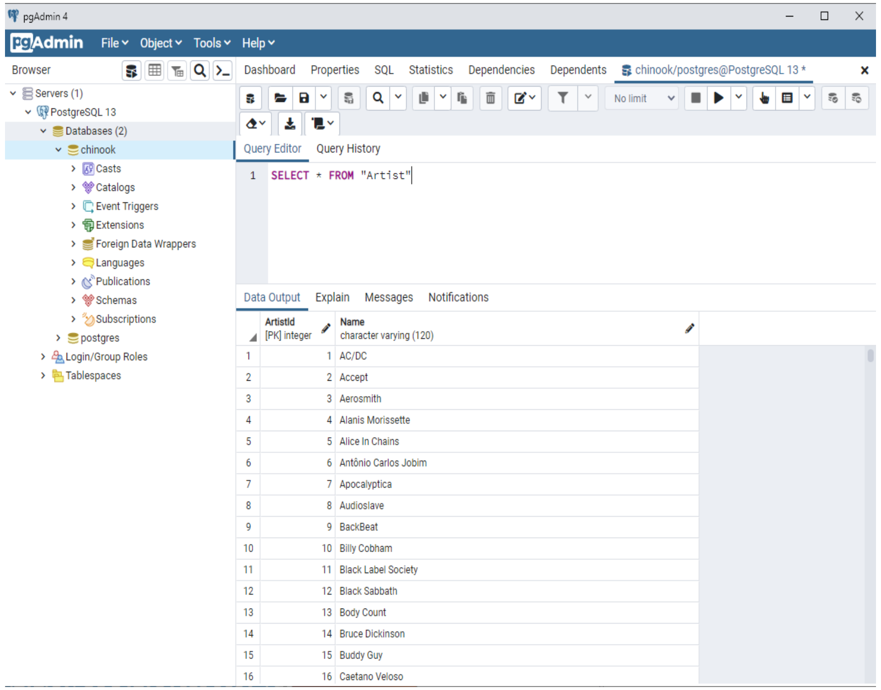
Data Platforms Map

January 2016

Key:
- General purpose
- Specialist analytic
- -as-a-Service
- BigTables
- Graph
- Document
- Key value stores
- Key value direct access
- Hadoop
- MySQL ecosystem
- Advanced clustering/sharding
- New SQL databases
- Data caching
- Data grid
- Search
- Appliances
- In-memory
- Stream processing

https://451research.com/state-of-the-database-landscape

Goals and Tasks (1)



Data Independence

- Manage data independent of applications, make data available for different applications
- **Physical data independence:** logical schema is independent of physical structure
- **Logical data independence:** external schema (for users / applications) is independent of logical schema

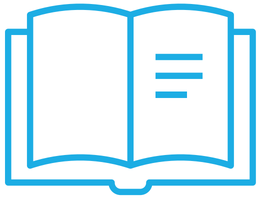
Data Manipulation & Retrieval

- DML: Data Manipulation Language
  - Retrieve / select
  - Insert / delete / update
- CRUD operations: create – read – update – delete

Goals and Tasks (2)

Structure definition and integrity assurance

- DDL: Data Definition Language
- Data dictionary / system catalog / metadata
- Integrity conditions / assertions / constraints



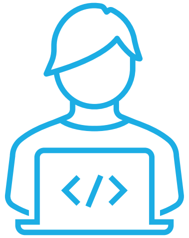
Protection of databases in multi-user mode

- Transaction management
  - **ACID**: Atomicity, Consistency, Isolation, Durability
- Recovery: Restart on error
- Data security and data protection

1.1 Goals and Tasks of DBMS (3)

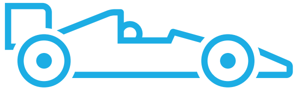
Realization of User Interfaces

- Interactive end-user interface
- API: Application Programming Interface



Performance control

- Monitoring of the system load and runtime behavior
- Index
- Clustering / data aggregation



DBMS and Applications

In contrast to OS, DBMS is an application

Databases: Consistent non-volatile memory

Application: Presentation of data, data processing

Application and DBMS often run on different computers

Communication: Connections between software systems (partially a job of OS)

ISO-OSI Reference Model for communication

AP	7	Application
	6	Presentation
OS	5	Session
	4	Transport
	3	Network
	2	Data Link
	1	Physical

Quiz

1.2 Basic Architecture of a DBMS

Learning Goals

At the end of this section you will be able to

- ✓ Explain the basic architecture of a DBMS
- ✓ Name the most important layers and interfaces of the basic and the 5 layer models and describe their tasks
- ✓ Give an example, how a query is processed throughout the layers

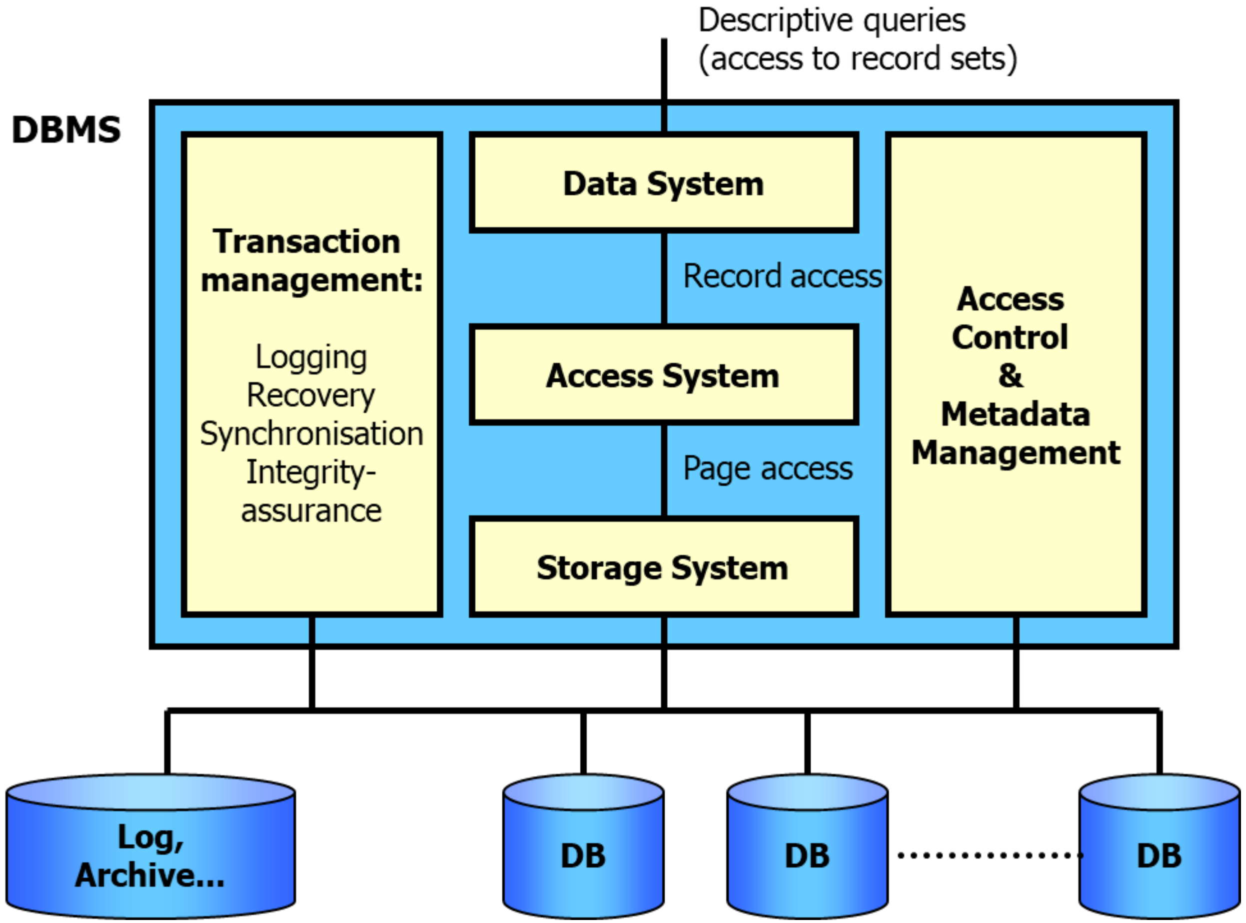
System Architecture Goal: Modularization

- Abstraction (concentrate on substantial points)
- Localization (of procedures & data)
- Information hiding principle / black box
- Completeness (on an abstract level)
- Verifiability

Concepts for modularization

- Functional abstraction
- Data abstraction
- Generic modules with objects and methods

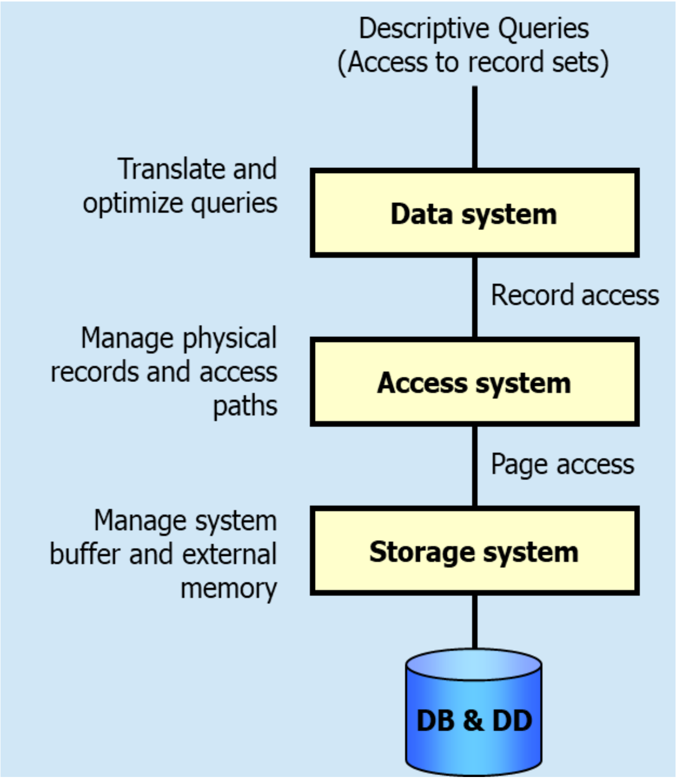
Simplified Architecture of a DBS



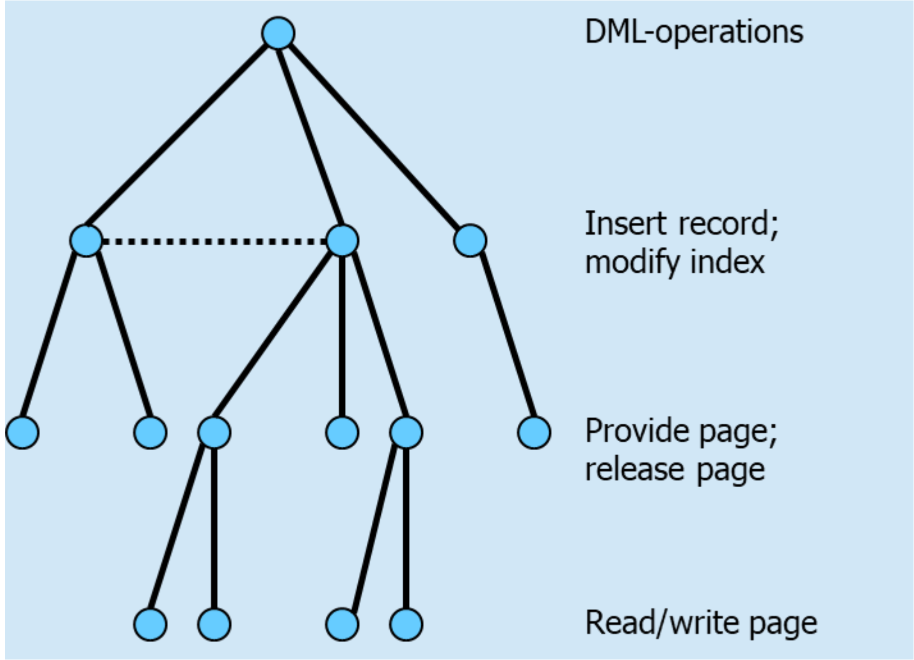
[Härder & Rahm, 2001]

Interaction of the Layers

Components



Control Flow



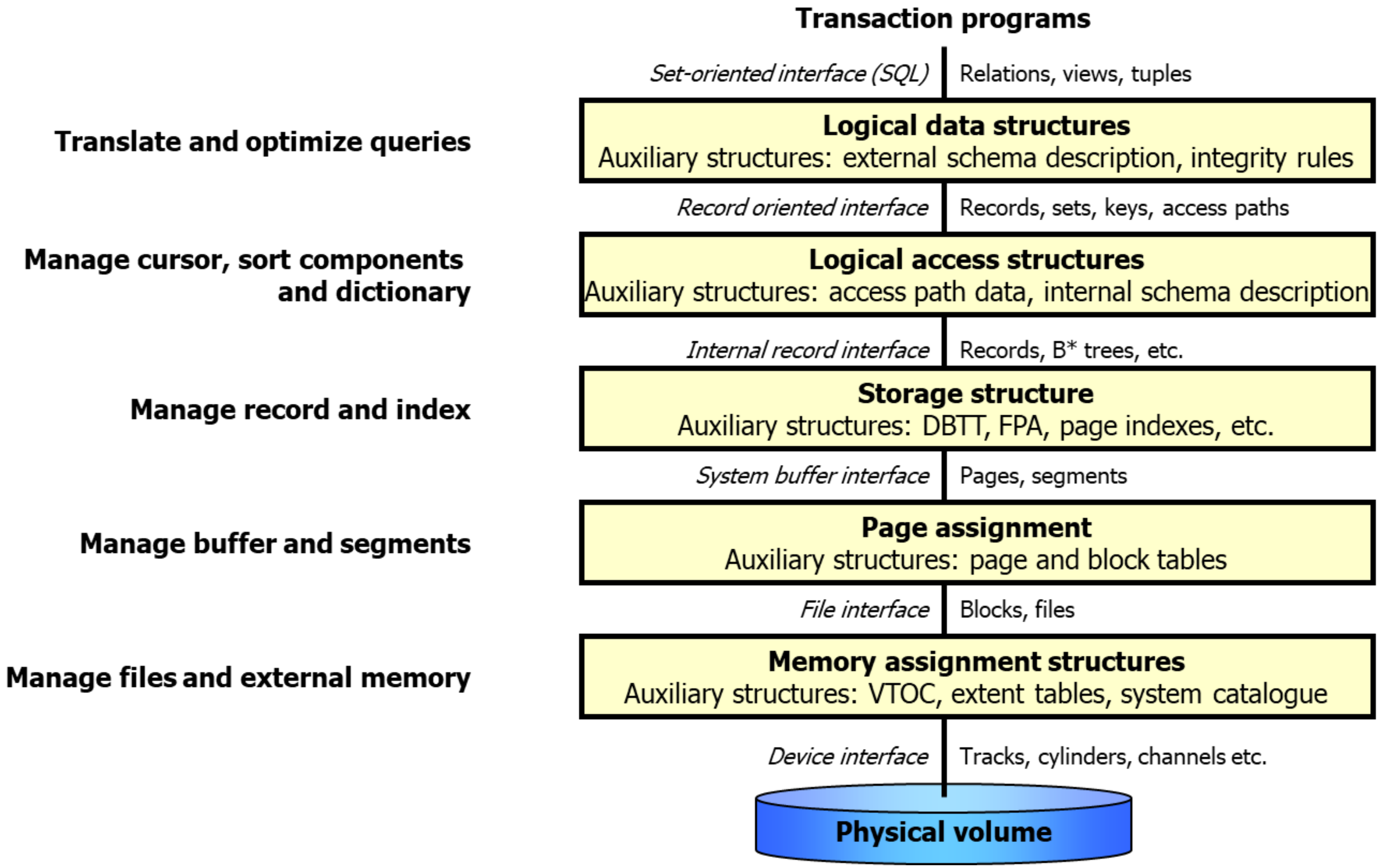
[Härder & Rahm, 2001]

Background Information on Hardware

	Capacity	Speed	Access time
L1-Cache	16-256 KB	300 GB/s	4 ns
L2-Cache	256 KB-4 MB	300 GB/s	10 ns
L3-Cache	few MBs	~50 GB/s	5-15 ns
RAM	GBs	Up to 10.000 MB/s	10-20 ns
Solid State Disc	GB – TB	200 MB/s	~0.1 ms
Hard disc	TBs	Up to 1.000 MB/s	7 ms
CD/DVD	640 MB – 20 GBs	10 MB/s	150 ms
Streamer	4 GB - >100 GB	2-10 MB/s	100 ms - >10 s
Network	-	1/10/100/1000 MB/s	ca. 1 ms

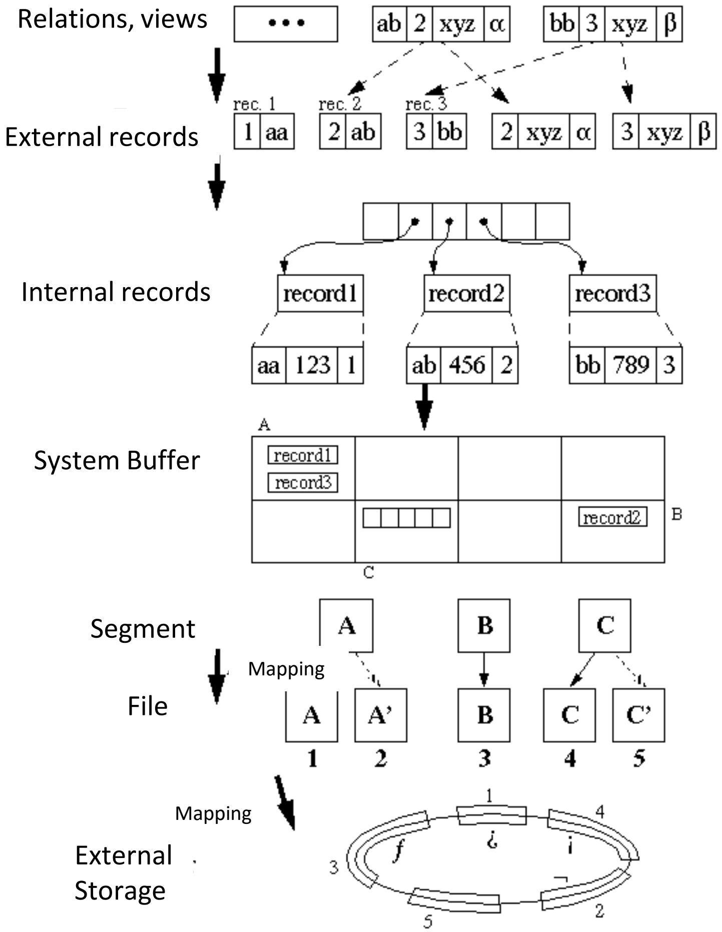
→ Prefer main memory, reduce disk accesses!

Five layer model of a DBS



[Härder & Rahm, 2001]

Example



[Härder & Rahm, 2001]

```
SELECT p.Name FROM Professor p, Vorlesung v
WHERE      p.Rang='C4' AND v.Titel='Logik'
           AND p.PersNr = v.gelesenVon
```

{ p.name | p∈Professor ∧ ∃v∈Vorlesung ∧ p.Rang='C4' ∧ v.Titel='Logik' ∧ p.PersNr=v.gelesenVon }

$$\pi_{Name}(\sigma_{Titel=Logik \wedge Rang=C4}(Professor \bowtie_{PersNr=gelesenVon} Vorlesung))$$

$$\pi_{Name}(\sigma_{Rang=C4}(Professor) \bowtie_{PersNr=gelesenVon} \sigma_{Titel=Logik}(Vorlesung))$$

OPEN CURSOR Vorlesung(Titel='Logik')
FIND NEXT record …
OPEN CURSOR Professor(Rang='C4')
…

B+-TREE-SEARCH Vorlesung(Titel='Logik')
FETCH RECORD Vorlesung(…,gelesenVon)
…
B+-TREE-SEARCH Professor(PersNr=gelesenVon)
…

LOAD PAGE 123
WRITE PAGE 345
…

Data Independence: An Overview

Layer	What is hidden?
Logical data structures	Position indicator and explicit relations in the schema
Logical access paths	Number and kind of the physical access paths; internal representation of records
Storage structures	Management of buffers, logging
Page assignment structures	File mapping, indirect page assignment
Memory assignment structures	Technical features and technical details of external storage media

Problems:

Due to high specialization, functionality of operating system often not usable

- Segment-file mapping
- Paging
- Shadow memory
- Buffer management
- Dispatching

Quiz

1.3 Evolution of DBMS

Learning Goals

At the end of this section you will be able to

- ✓ name the most important stages in DBMS evolution
- ✓ classify a DBMS along different dimensions
- ✓ name different client-server architectures and discuss pros and cons
- ✓ explain tasks of client and server and name three different server types
- ✓ explain distributed database principles
- ✓ explain what is a big data architecture

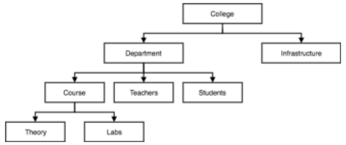
Evolution Time Line

1960s – 1980s

Flat files

1970s – 1990s

Hierarchical & Network Databases

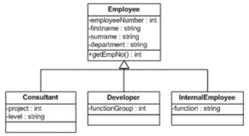


1980s – today

Relational Databases

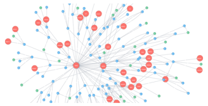
1990s – today

Object-oriented and object-relational Databases



2000 – today

Internet and Graph Databases



2008 – today

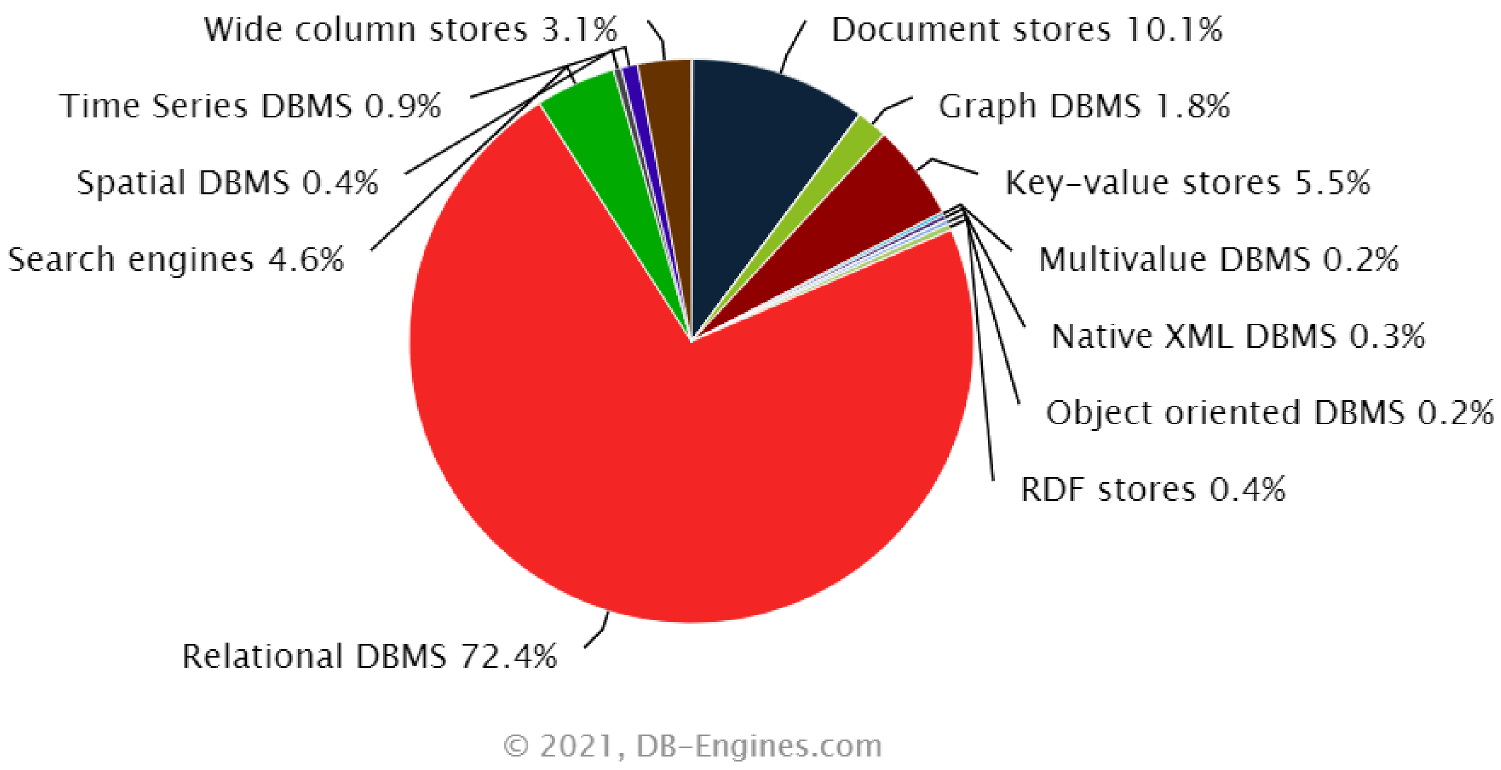
NoSQL & Big Data

mongoDB

DynamoDB

APACHE kafka

Data Model Popularity Today

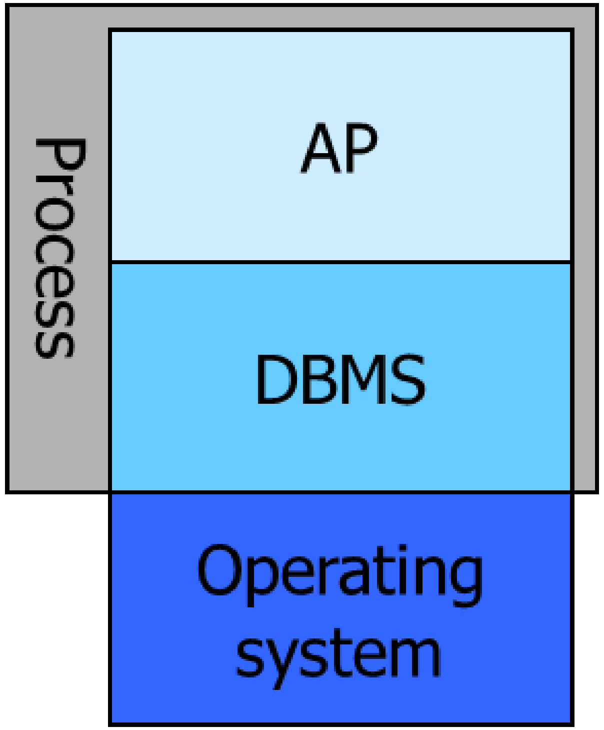


Classification Dimensions of DBMS [Elmasri & Navathe, 2017]

- Data Model
- Number of users
- Number of sites
- Costs & License
- Types of access path
- Purpose

Embedded DBMS:
1-Layer-Architecture, Single User DBMS

- One process, one address space
- No concurrency control
- Simple crash recovery



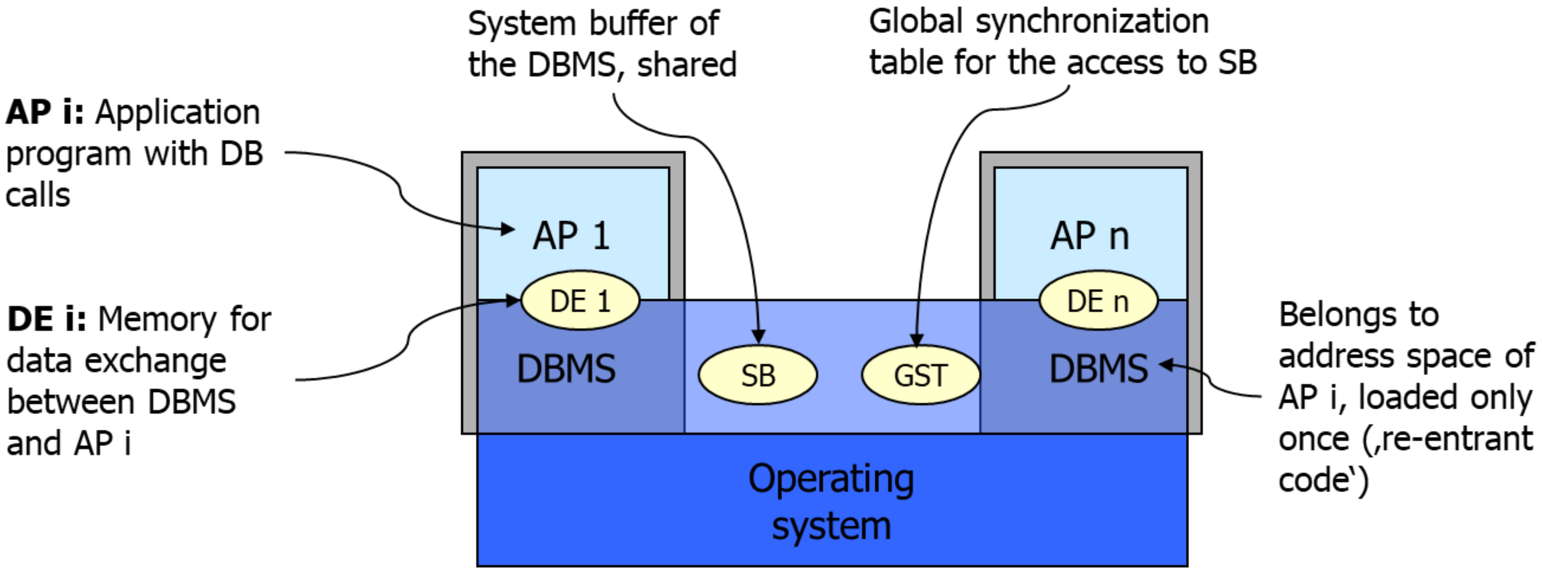
AP: Application program with DB calls

DBMS: Belongs to the address space of the AP

Example: PC database systems (MS Access)

Embedded DBMS: 1-Layer-Architecture – Multi-user DBMS

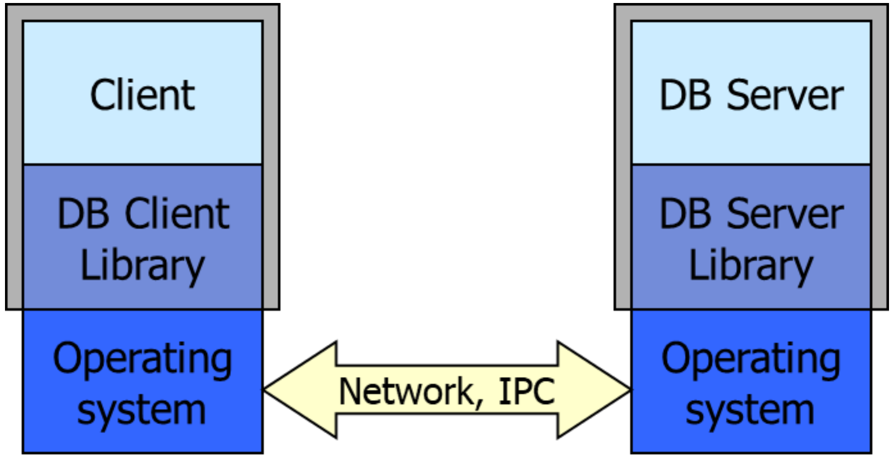
- Multiple processes, communication via shared address space
- Very efficient data exchange via shared memory, but
- No security concerning errors in the AP



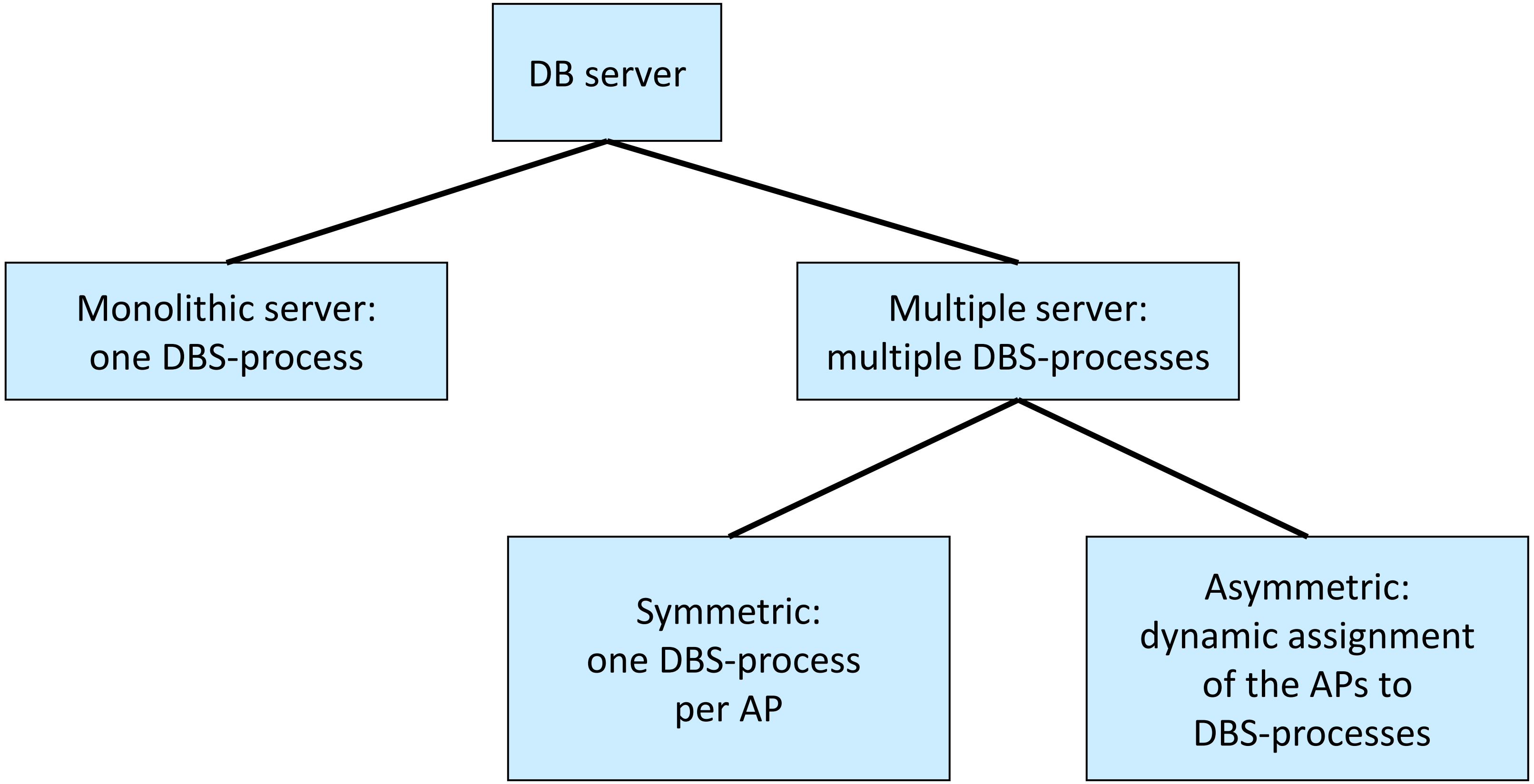
Example: IBM System/R* – research prototype

Embedded DBMS: 2-Layer-Architecture ("Client/Server")

- Client and server totally separated $\Rightarrow$ distributed access to DB
- Communication among clients and servers via a network or IPC.
- Specialized protocols used (JDBC, Net8, TCP/IP)
- Clear separation of client and server



Classification in Client/Server Architecture



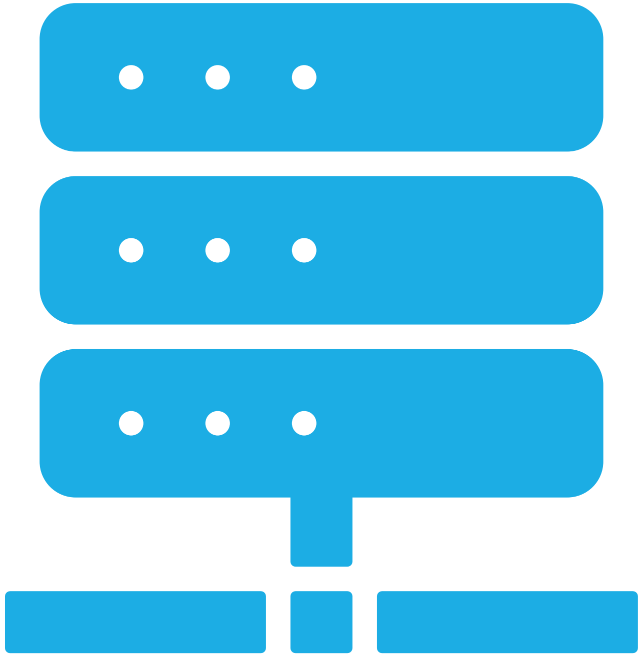
Multiple DB Server Processes

- Communication among the servers via “shared memory”
- Communication among clients and servers/dispatcher via OS-mechanisms (IPC) or network software

- **Symmetric assignment:**

  Each client assigned to exact one server process.
  Static assignment, fixed number n of servers stated in advance
  $\Rightarrow$ maximal degree of parallelism is n

- **Asymmetric assignment:**
  Each client assigned to a server process by a dispatcher.
  Fixed number n of servers stated in advance, but degree of parallelism can be higher.

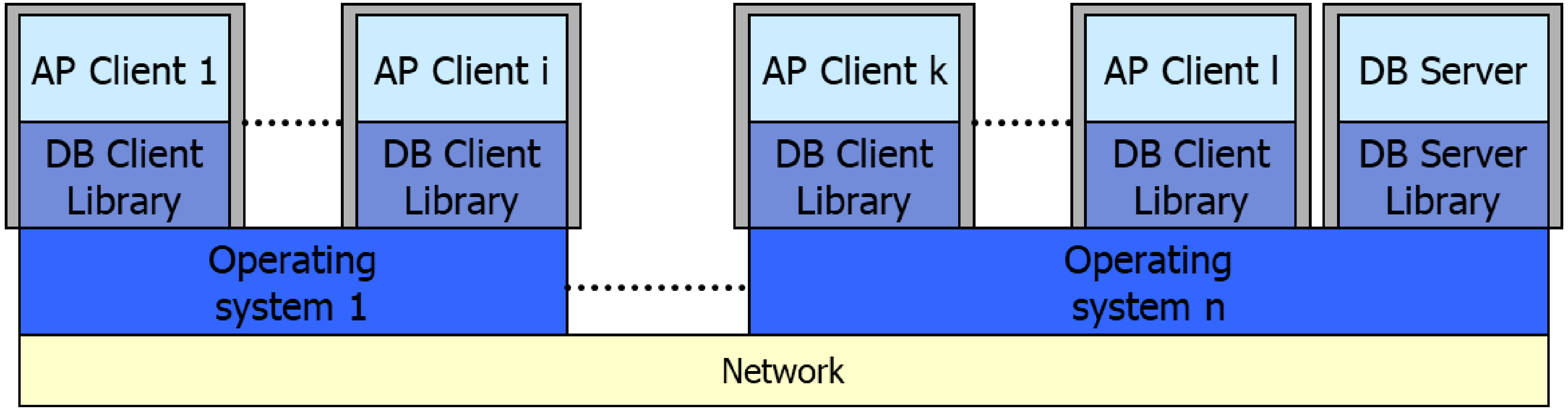


Single DB Server Process

- Synchronized access to system buffer and central system tables
- Server uses multi-threading (“re-entrant code”)
- Only one server process for many clients
- DB server process is preferred by OS

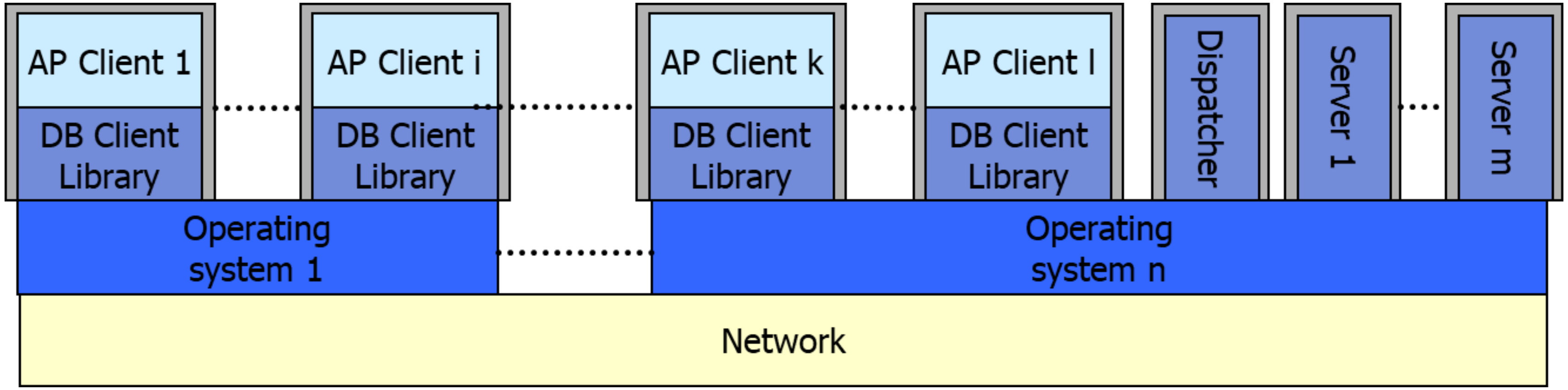
Monolithic Server

- Own resource management, duplicates OS functions
- Simple communication in the server via shared memory
- Example: PostgreSQL



Multiple Servers

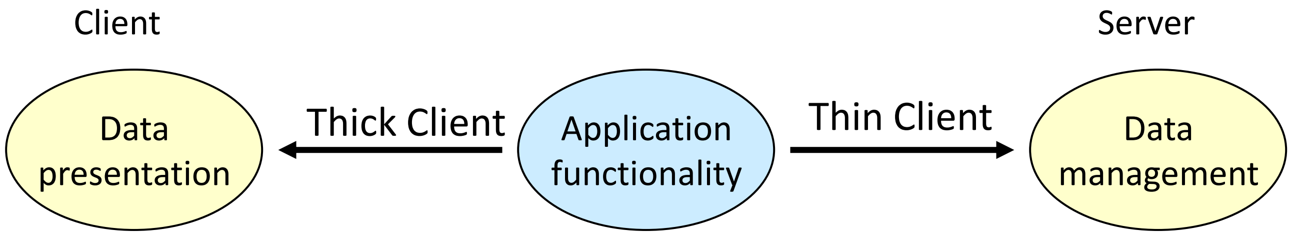
- DBMS is a compound of different processes
- Communication via operating system or network
- Process scheduling by OS, advantageous in multi-processor computers, because OS manages processor allocation.
- Example: Oracle, IBM DB2, MySQL, MS SQL Server, Sybase



Tasks of Client and Server

Tasks of a server

- Data management
  - Relation Server
  - Object Server
  - Page Server
- Application functionality

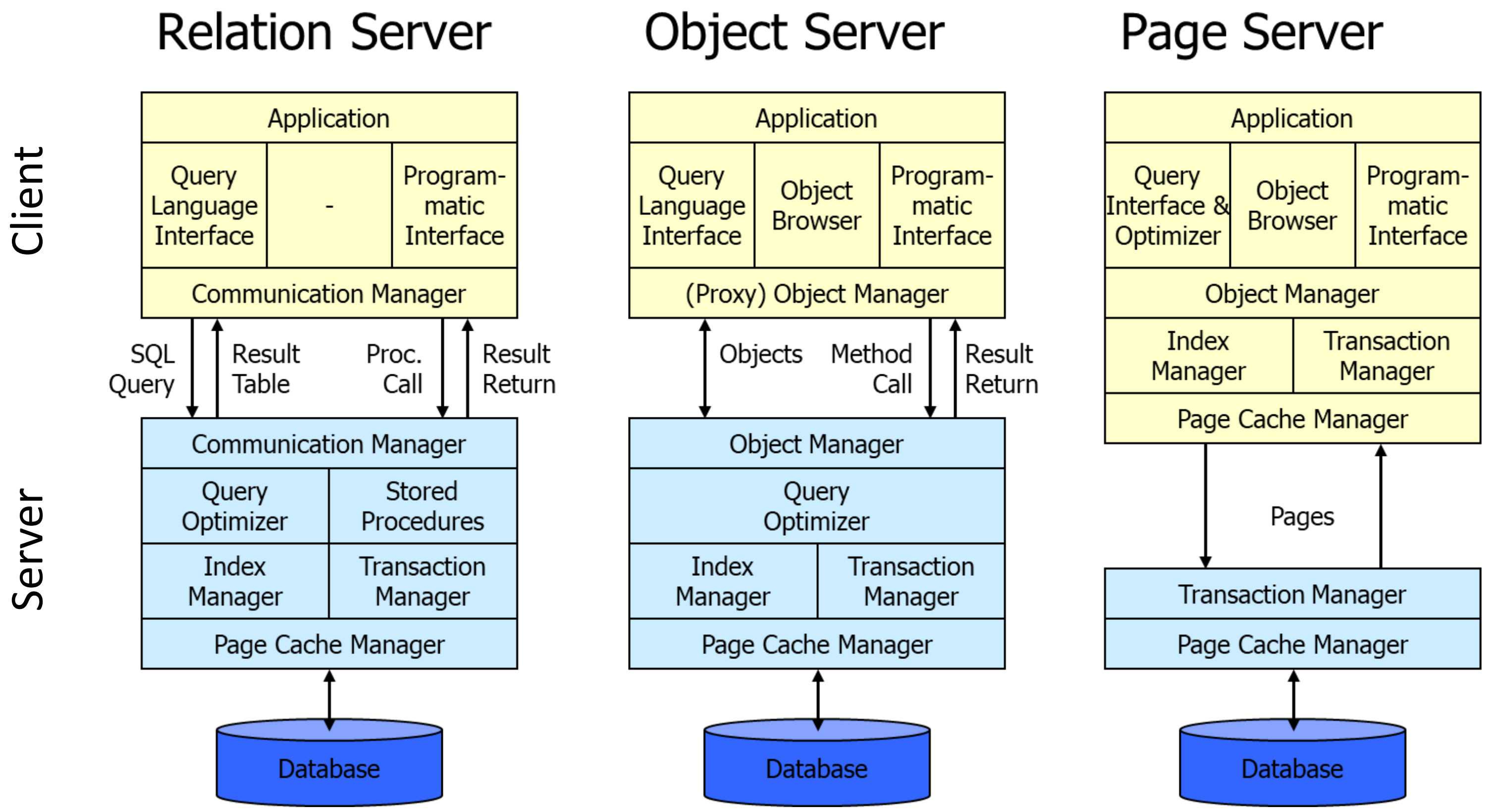


Tasks of a client

- Presentation of data
- Application functionality

Server Types

[Härder & Rahm, 2001]



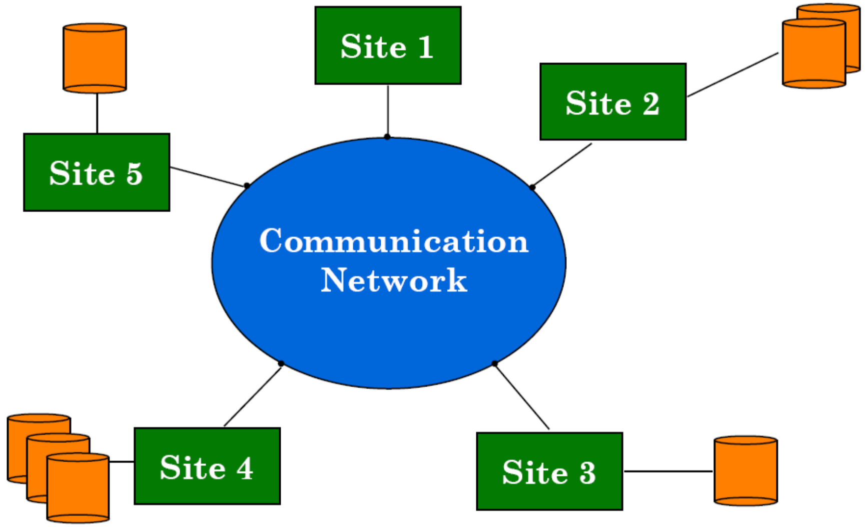
Quiz

Distributed Database Systems

- A distributed database (DDB) is a collection of multiple, *logically interrelated* databases distributed over a *computer network*
- A distributed database management system (D–DBMS) is the software system that permits the management of the distributed database and makes the distribution *transparent* to the users

Distributed database system (DDBS)

DDBS = DDB + D–DBMS



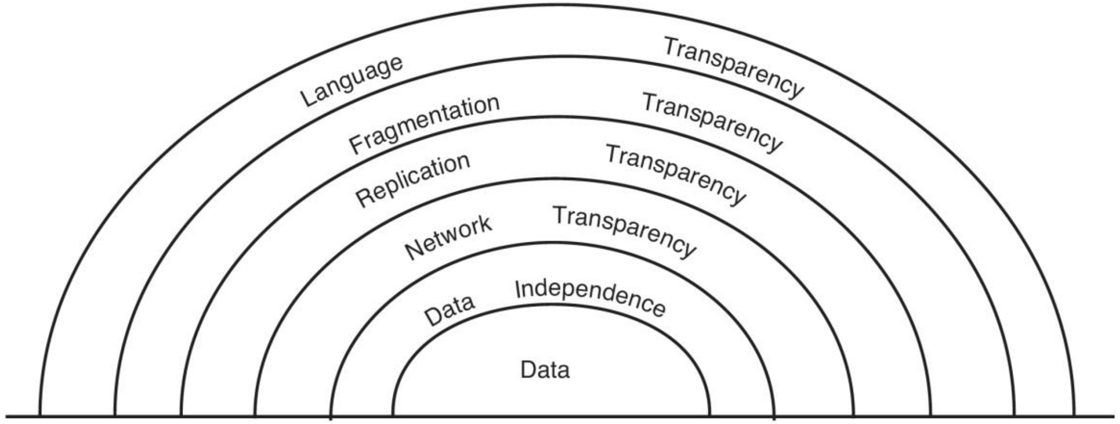
[Özsu & Valduriez, 2011]

Promises of D-DBMS

- Transparent management of distributed, fragmented, and replicated data
- Improved reliability/availability through distributed transactions
- Improved performance
- Easier and more economical system expansion
  → Scalability

Implicit Assumptions

Data stored at a number of sites	Each site *logically consists of a single processor.*
Processors at different sites are interconnected by a computer network	no multiprocessors → parallel database systems
Distributed database is a database, not a collection of files	Data is logically related as exhibited in the users' access patterns → relational data model
D-DBMS is a full-fledged DBMS	→ not remote file system, not a Transaction Processing system



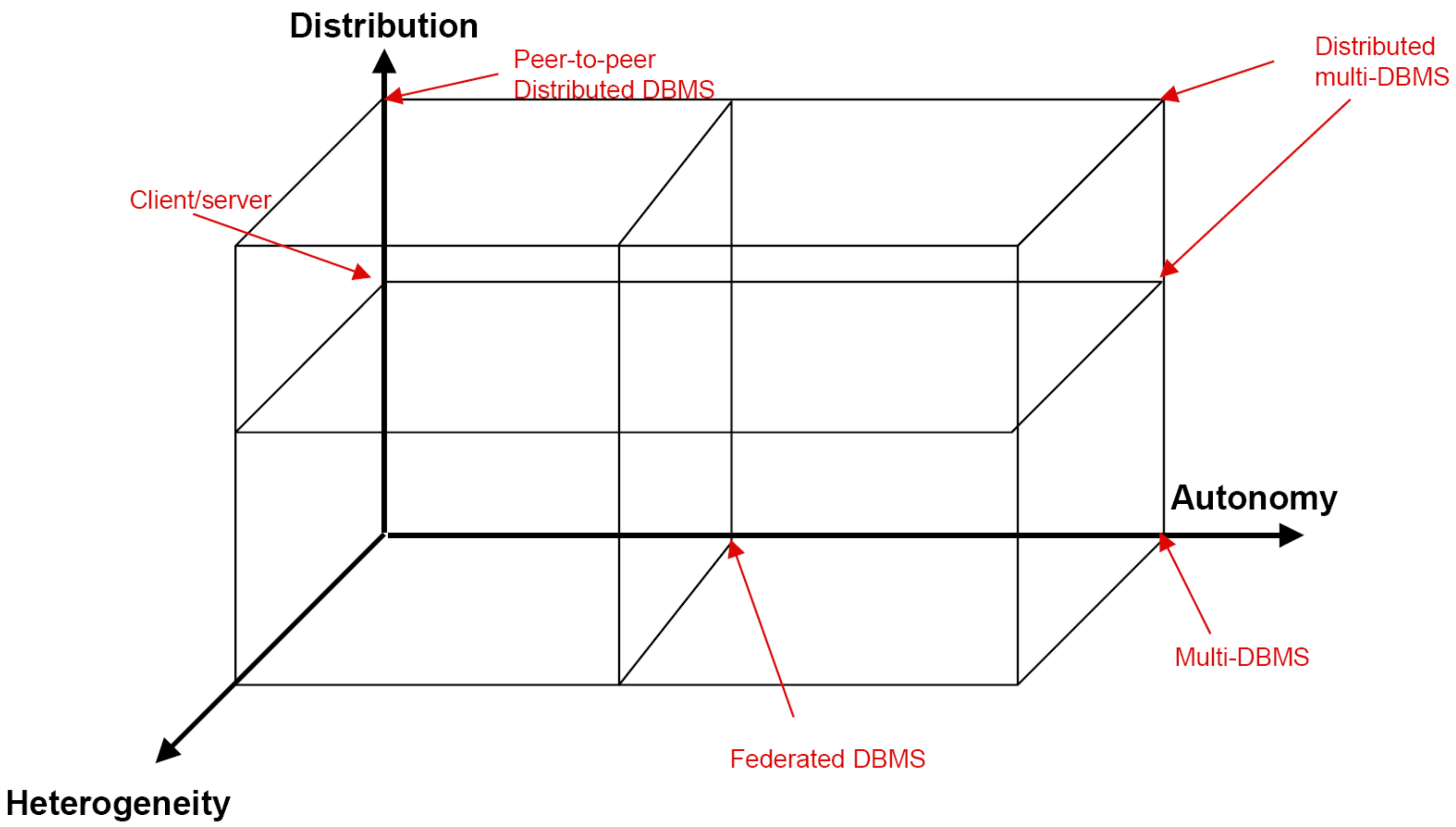
Transparency

… is the separation of higher-level semantics of a system from lower-level implementation issues

Fundamental issue is to provide **data independence** in the distributed environment

- Network (distribution) transparency
- Replication transparency
- Fragmentation transparency
  - horizontal fragmentation: selection
  - vertical fragmentation: projection
  - hybrid

D-DBMS: Implementation Alternatives



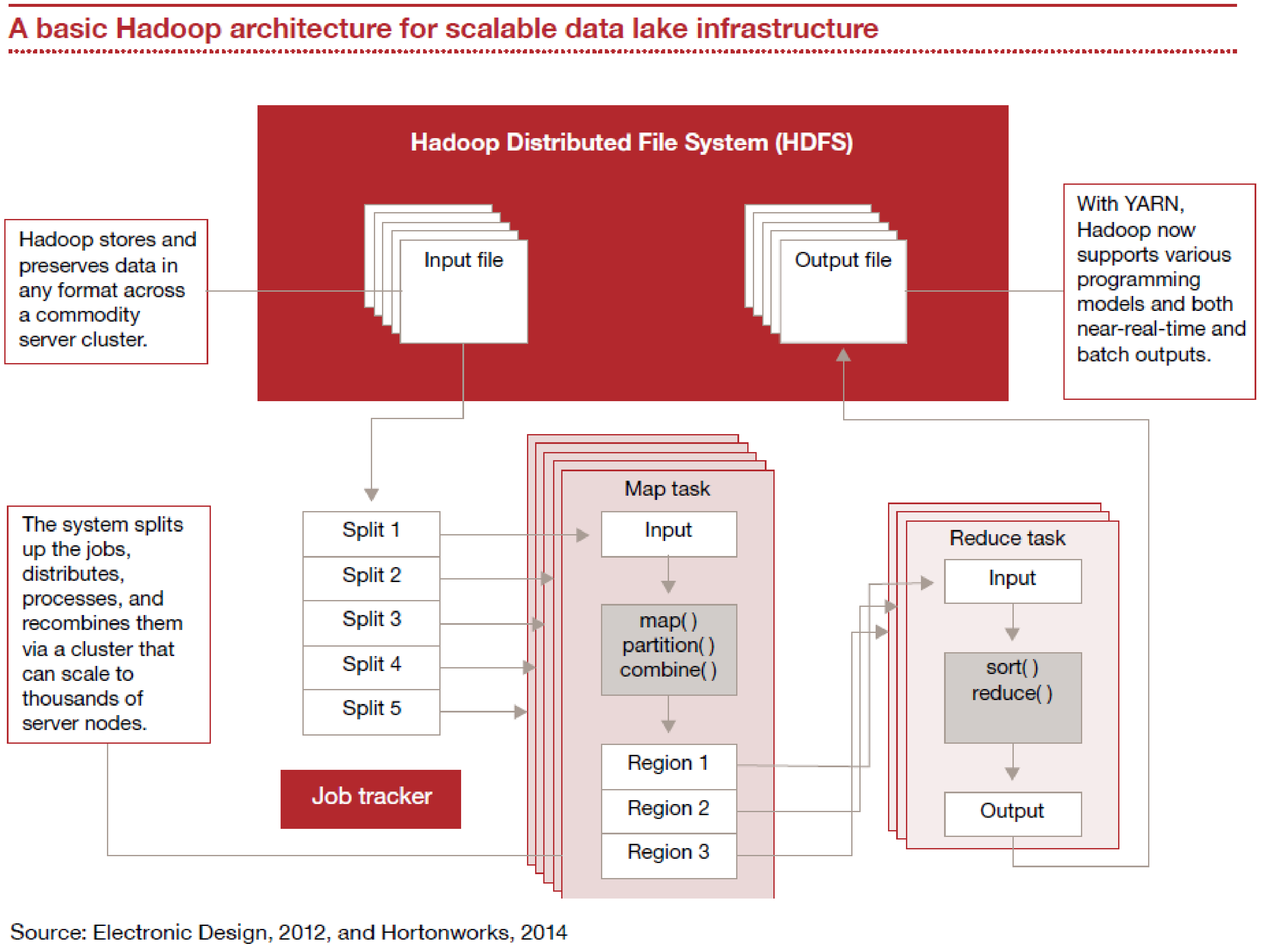
Big Data Architectures

- Big Data requires large, scalable, distributed architectures
- Four/Six Vs : Volume, Velocity, Variety, Veracity, [Hofstee & Nowka, 2013], (Value, Variability)
- Heterogeneity
  - Sources
  - Systems
  - Requirements
  - Client Applications

➔Big Data Systems are complex eco-systems, independent components have to be integrated

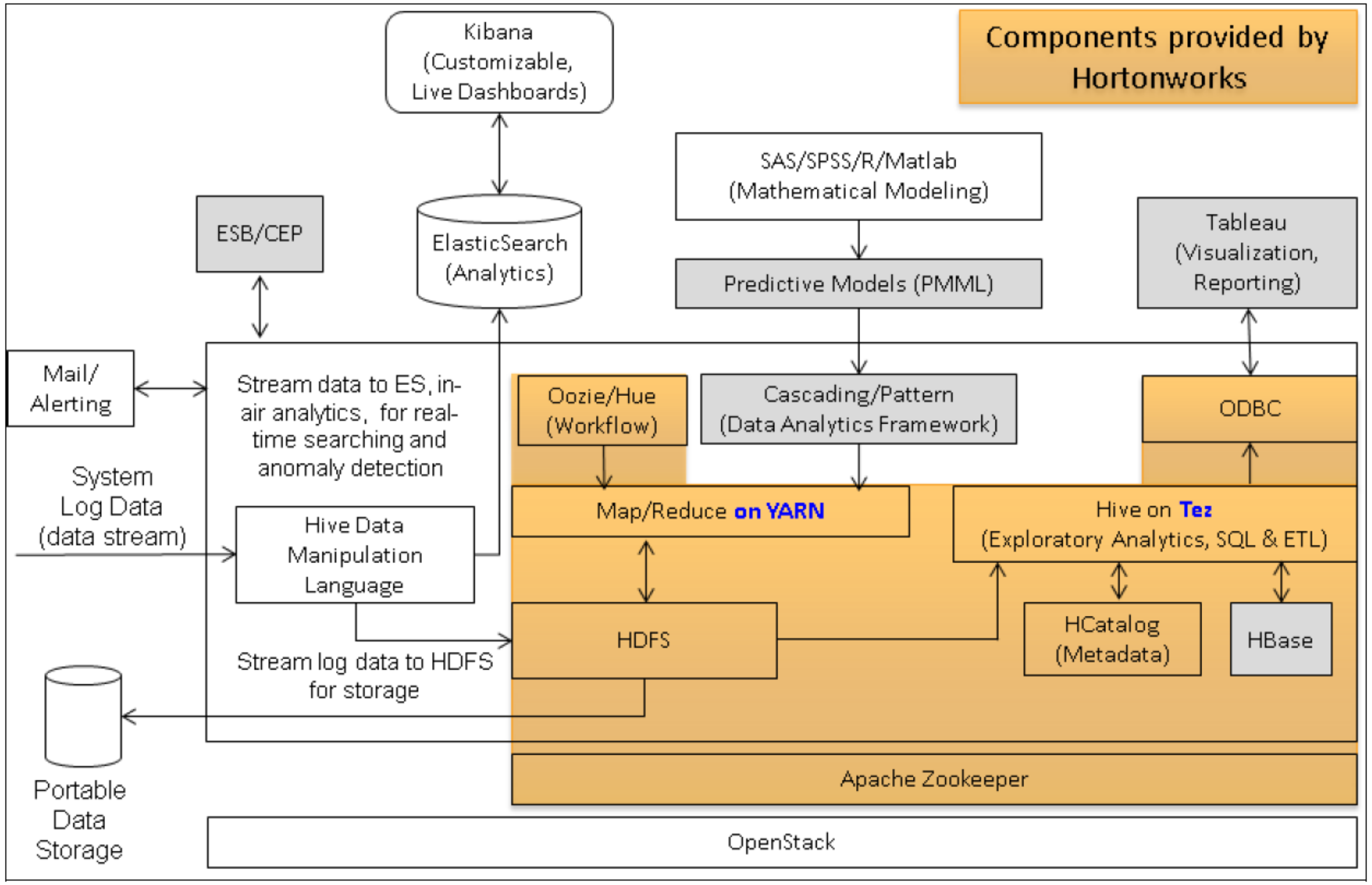
➔Hadoop is not a Big Data system, it is just a component

Hadoop as a Basic Big Data Platform

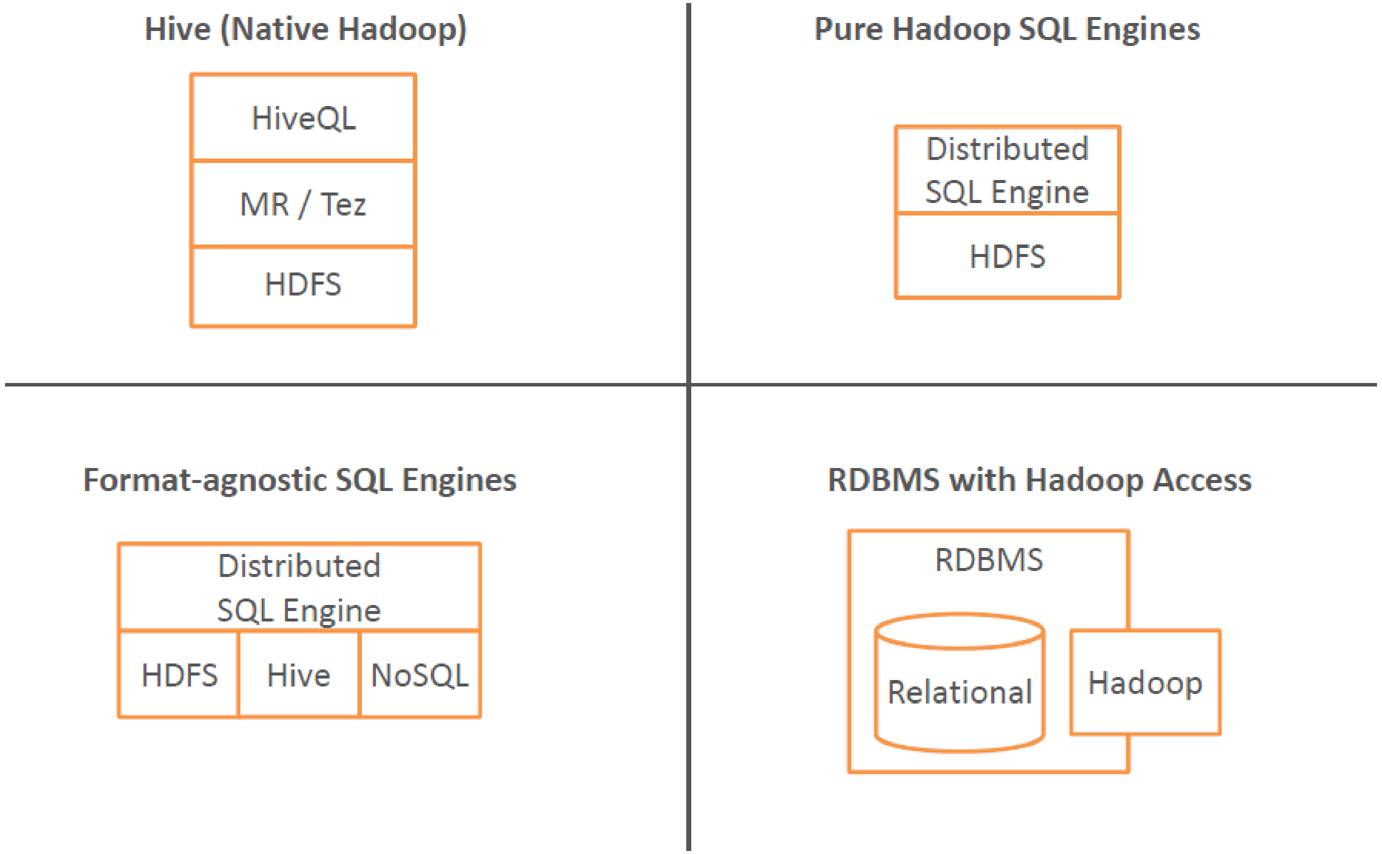


Source: https://www.pwc.com/us/en/technology-forecast/2014/cloud-computing/assets/pdf/pwc-technology-forecast-data-lakes.pdf

Example Big Data Architecture [Boci & Thistlethwaite, 2015]



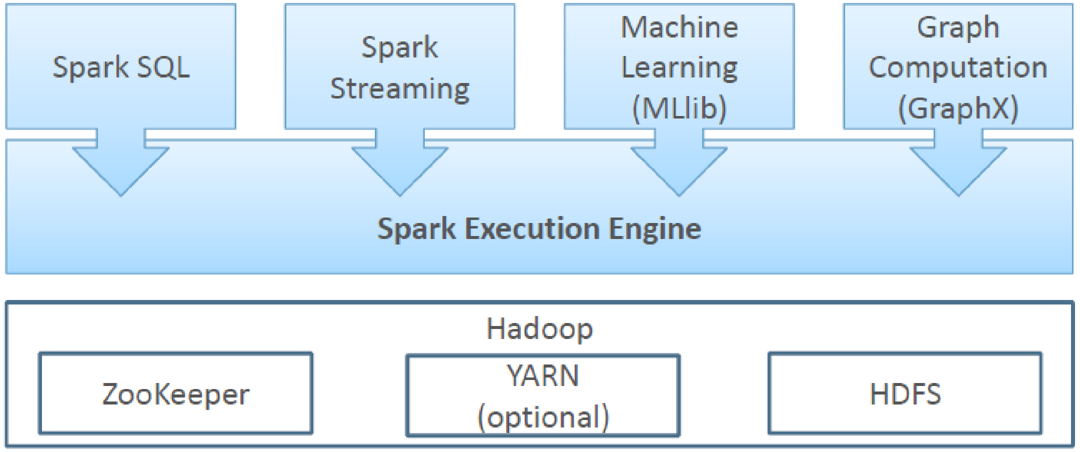
Architectures for Hadoop with SQL



[Albrecht, 2015]

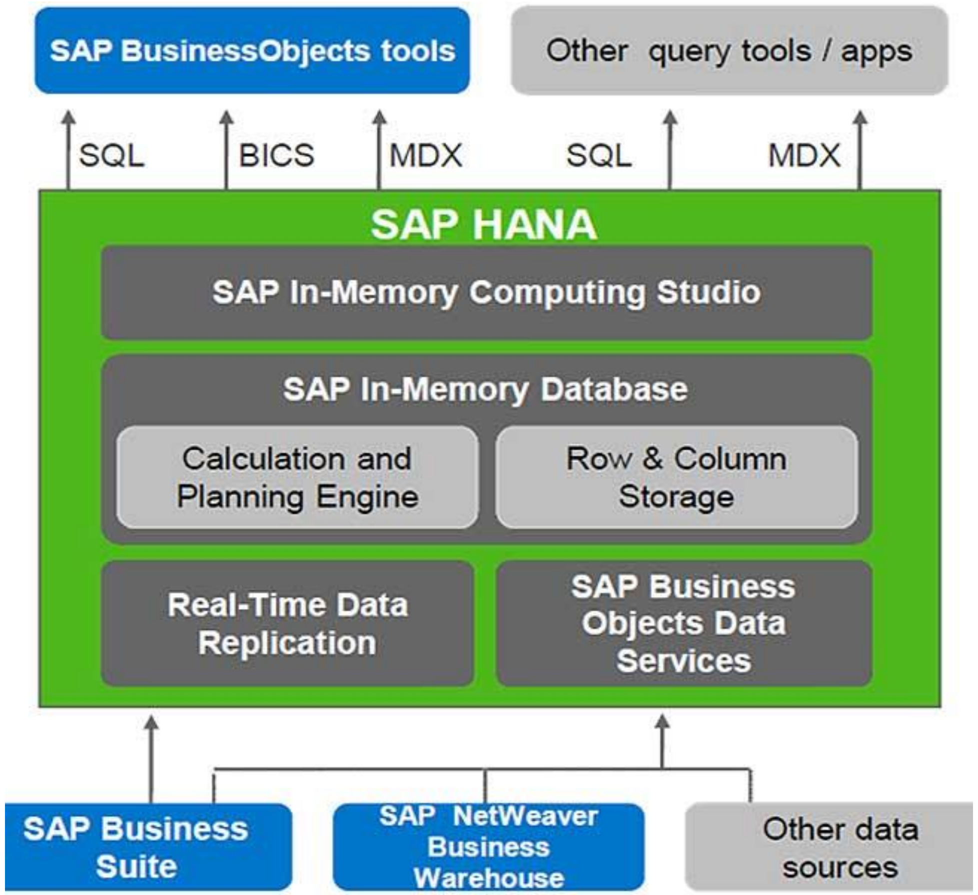
In-Memory Database Systems

- Distributed data processing, but do as much as possible in-memory and avoid I/O operations (to disk or distributed file system)
- Example: Apache Spark
  - Distributed In-Memory Computing Framework
  - General framework for all kinds of SQL and non-SQL analytics

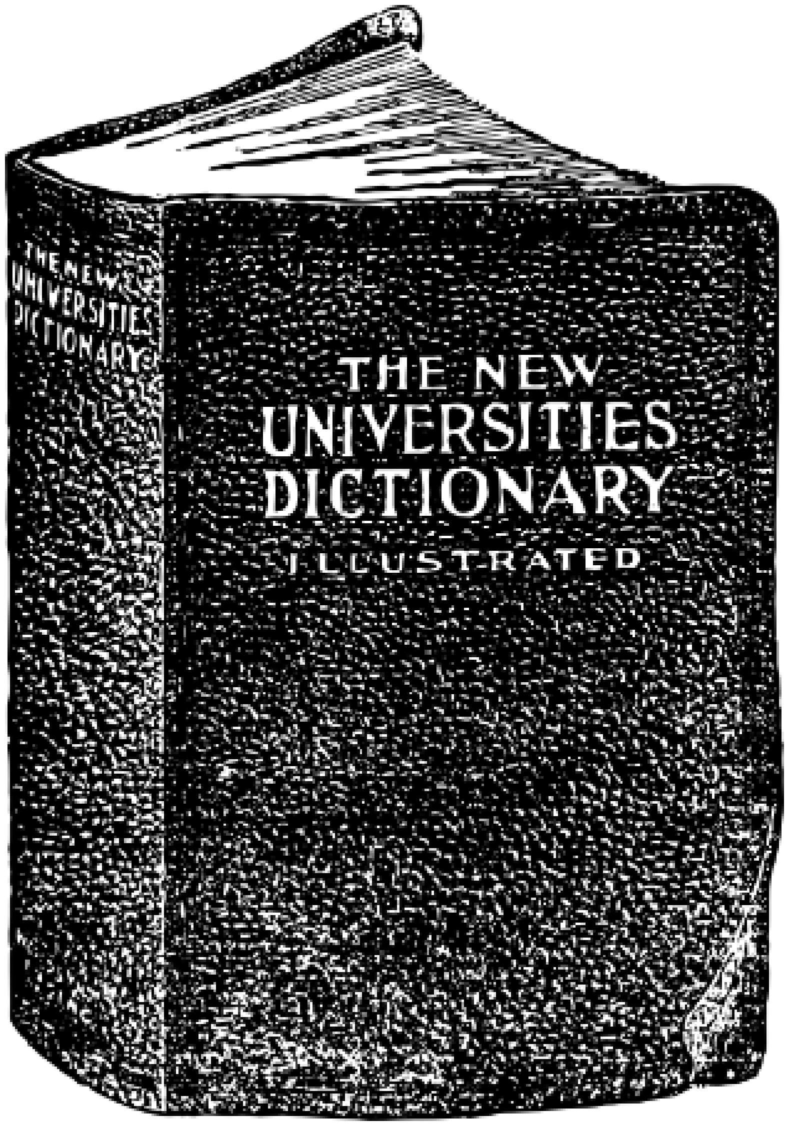


[Albrecht, 2015]

Another Example for an In-Memory Database System



https://www.stechies.com/overview-sap-hana-database-architecture/



DBMS with Data Dictionary

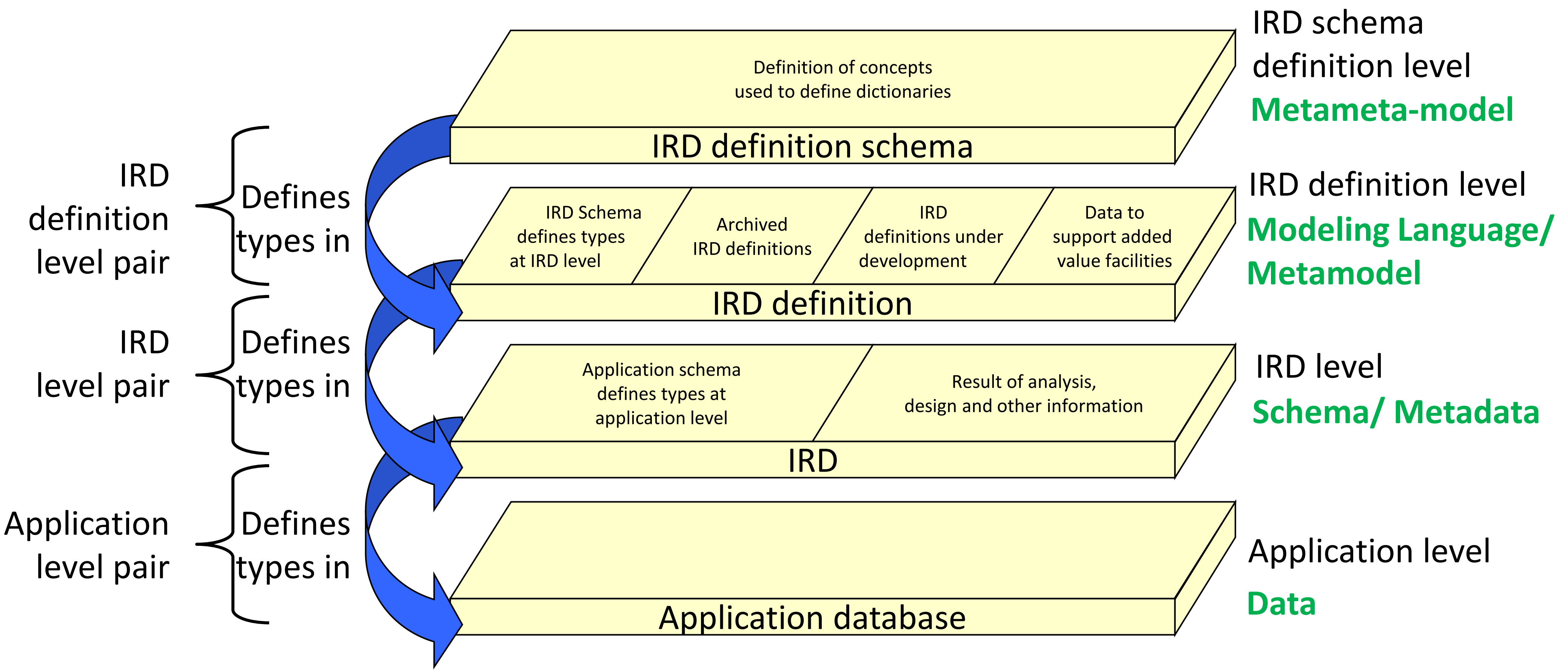
Problem: Each mapping step means loss of semantics

- Realisation of operations requires information about
  - Schemas
  - Integrity constraints
  - Index structures
  - Access authorization
  - …

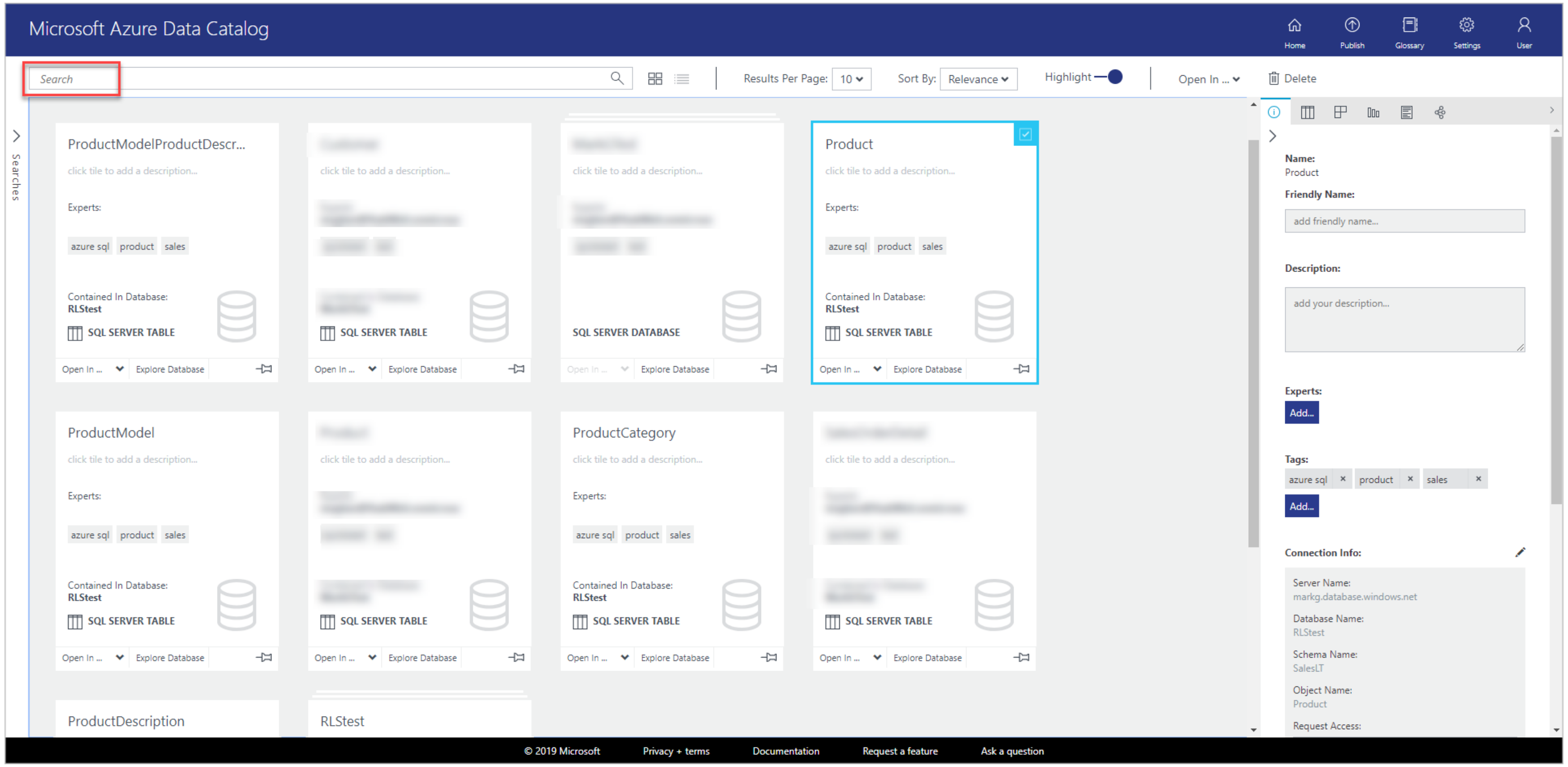
Approach: Comprehensive management by data dictionary

- Internal in DB (uniform model)
- Stand-alone module (fast and specialized service)

IRDS Framework Standard (ISO 10027:1990)



Example: Azure Data Catalog



https://docs.microsoft.com/de-de/azure/data-catalog/register-data-assets-tutorial

Quiz

Chapter 1 - Review Questions

- Explain the concept of data independence!
- What are the layers of the five layered DBMS architecture?
- How is a query processed in a DBMS?
- Why is transaction management not assigned to a single layer in the DBMS architecture?
- What are the four levels in the IRDS architecture?
- What is distribution transparency in distributed database systems?
- What are basic characteristics of Big Data?

References & Further Reading

Parts of the slides are based on course material by

- Prof. Dr. Matthias Jarke (Information Systems and Databases, RWTH Aachen University)
- Prof. Dr. Christoph Quix (Wirtschaftsinformatik und Data Science, Hochschule Niederrhein)

Further Reading

[Albrecht, 2015] J. Albrecht: Processing Big Data with SQL on Hadoop. TDWI, 2015.

[Boci & Thistlethwaite, 2015] Boci, E. & Thistlethwaite, S.: A novel big data architecture in support of ADS-B data analytic *Proc. Integrated Communication, Navigation, and Surveillance Conference (ICNS),* **2015**, C1-1-C1-8

[Elmasri & Navathe, 2017] Elmasri, R., & Navathe, S. (2017). *Fundamentals of database systems* (Vol. 7). Pearson.

[Härder und Rahm, 2001] Härder, T. & Rahm, E. Datenbanksysteme: Konzepte und Techniken der Implementierung Springer Heidelberg, 2001

[Hofstee & Nowka, 2013] Hofstee, P. and Nowka, K. J. (2013). The Big Deal about Big Data - A Perspective from IBM Research.

Presentation at IEEE NAS Conference, Xi'An China.

[IRDS Framework Standard, ISO 10027:1990] https://www.iso.org/obp/ui/#iso:std:iso-iec:10027:ed-1:v1:en

[Kemper & Eickler, 2015] Kemper, A., & Eickler, A. (2015). Datenbanksysteme; 10., akt. u. erw. Aufl. (in German)

[Özsu & Valduriez, 2011] Özsu, M. T. & Valduriez, P. Principles of distributed database systems Springer, 2011