Data Modelling and Databases

Donald Kossmann
Systems Group
ETH Zürich
www.systems.ethz.ch

Schedule

- Lectures
  - Mondays: 10:00 – 12:00
  - Wednesdays: 8:00 – 10:00
  - Held in German (English slides)
- Exercise Groups (Start March 5)
  - Tuesdays: 8:00 – 10:00
  - Fridays: 8:00 - 10:00
  - Held in English and German
- **Please, register during the break today:**
  - lists at the front desk of lecture room

Literature

- Kemper, Eickler: Datenbanksysteme: Eine Einführung. Oldenbourg Verlag, 7. Auflage, 2009.

or

- Garcia-Molina, Ullman, Widom: Database Systems: The Complete Book. Pearson, 2. Auflage, 2008.

Overview

- **How to use a database system?**
  - **Data modelling (ER, UML, theory)**
  - **Database programming (SQL)**
- How to build a database system?
  - Query optimization
  - Transaction management
- What next?
  - Object-orientierted, object-relational databases
  - Data Warehousing, Decision Support, Data Mining
  - XML & WWW

Detailed Schedule

Week No.	Date (Mi)	Topic Lecture	Topic Exercises
1	20.2.2013	Introduction	---
2	27.2.2013	ER, UML	---
3	6.3.2013	Relational Model	ER
4	13.3.2013	SQL I	Start project
5	20.3.2013	Guest Lecture, SQL II	Relational Model
6	27.3.2013	Integrity Constraints	---
7	3.4.2013	---	---
8	10.4.2013	Normal forms I	SQL
9	17.4.2013	Normal forms II	IC, Project: Part I
10	24.4.2013	Query Processing I	Normal forms
11	1.5.2013	(Query Processing II)	Normal forms, Proj.
12	8.5.2013	Transactions	Query Processing
13	15.5.2013	Synchronization	Transactions
14	22.5.2013	Security	Synchronization
15	29.5.2013	Object-relational Databases	End Project: Part 2

Exercises & Exam

- Project
  - Part 1: Build an App; Part 2: Build a DB
  - Groups of three students
  - Not graded
- Exercise Sheets
  - Handout in the week before it is discussed
  - Not graded
  - Please, do them *before* they are discussed!
- Sessionsprüfung (written, 90min, closed book)

Internships in India (Accenture)

- If you are interested in an internship in India, please, contact: **donaldk@ethz.ch**
- Duration: Juni - September 2013 (no classes)
- Requirements
  - doing well with regard to your study plan
  - English
  - Programming languages: Java, SQL
  - Bachelor and Master students
- Application deadline: **15. March 2013 (by E-Mail)**
- This is an adventure!!!
  - (I would do it myself, if I met the requirements.)
- **„Session Exams" can be taken in India**

What is a Database System (DBMS)?

- A DBMS is a tool that helps develop and run data-intensive applications:
  - large databases
  - large data streams

Vision

- Store all data and make it available and useful to all authorized people, anytime and anywhere.
- Google ‘s mission statement:
  Organize all the information of the world.
- Status: Technology is there (card boxes). The model is missing (labels).

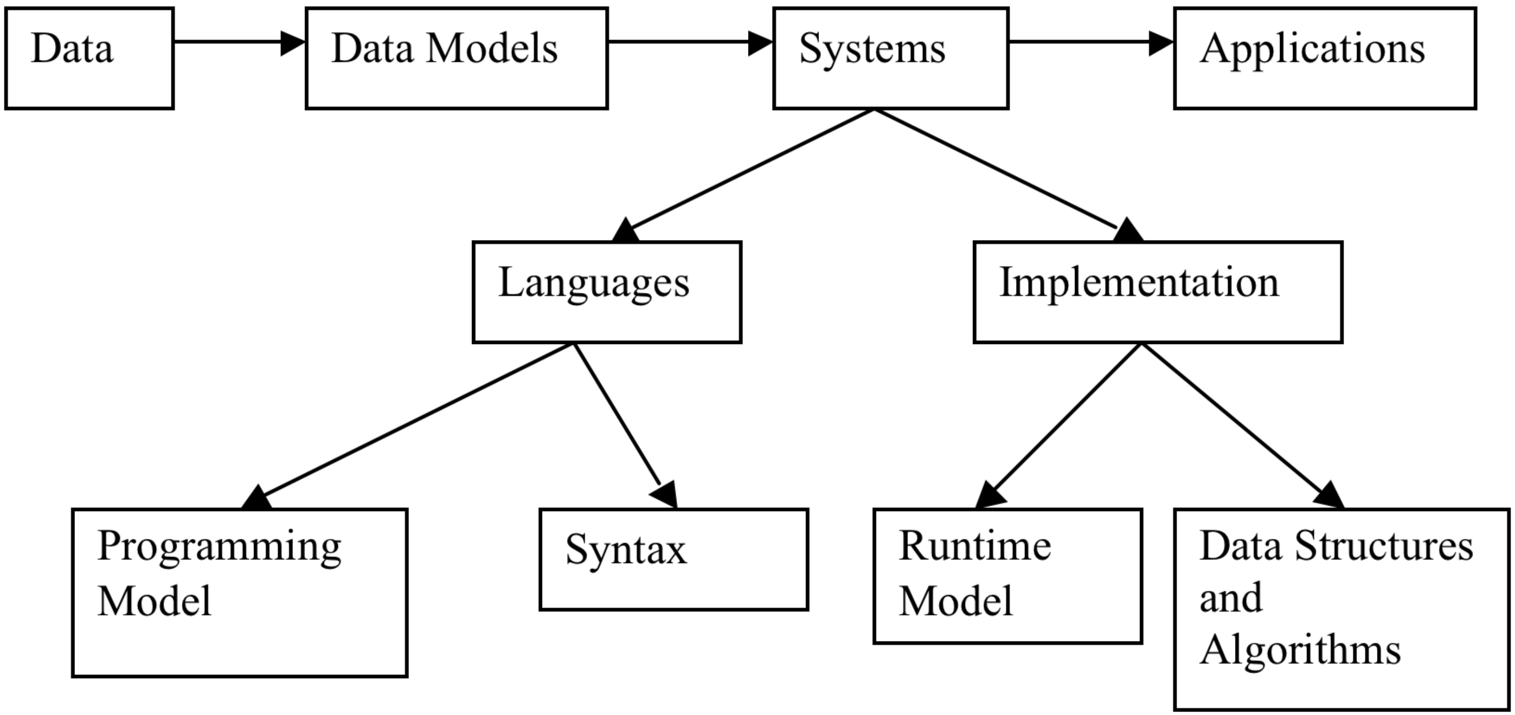
Simple Truths

- „Power of data“
  - the more data the merrier (GB -> TB)
  - data comes from everywhere in all shapes
  - value of data often discovered later
  - data has no owner within an organization (no silos!)

- Services turn data into $
  - the more services the merrier
  - need to adapt quickly

- E.g.: Google, Amadeus, Disney, Walmart, BMW, ...
- Platforms: Oracle, MS, SAP, Google, ..., 28msec

The Data Management Universe



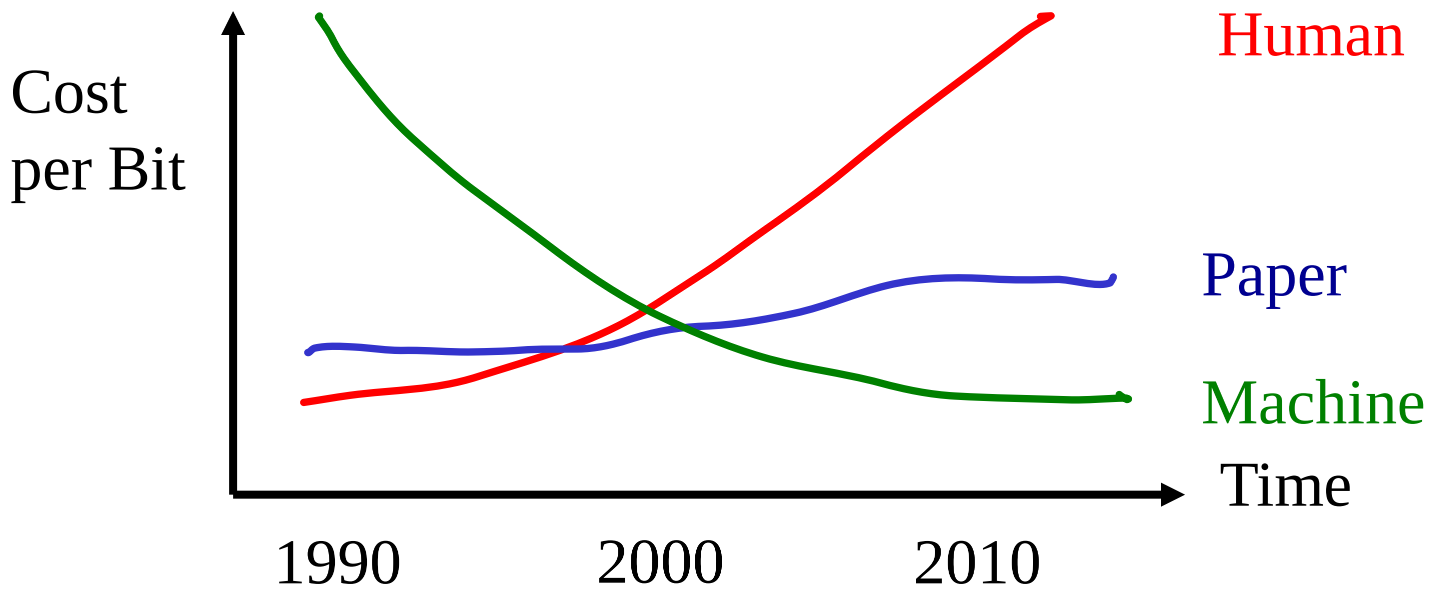
Data and Data Models

- Formats
  - XML, serialized Java objects, binary, ...
- Structures / Models
  - Tuples, hierarchies, relationships, lists, unstructured, ...
- Examples
  - Lecture notes
  - Financial accounts
  - Emotions (?): love, taste, ...

Systems

- Software platforms that store & organize data
  - File system: Windows, ...
  - Relational database systems: Postgres, Oracle, ...
  - Other database systems: Sausalito (;-)), OODB, ...
  - Key/value stores: HBase, AWS S3, MongoDB, ...
  - Interpreters: JVM, .NET, ...
  - Human intelligence
- Hardware that stores & organizes data
  - HDD, SSD, main memory, ...
  - Paper
  - Human brain

Where is data stored today?



Mechanical Turk: Prices for humans going down again. How come?

Typical Applications (data / operations)

- Bank (Accounts / „Money Transfer“)
- Library (Books / „Lend Book“)
- Content Management System (docs, „show“)
- E-Business (Catalogue, „search“)
- ERP (Order, „delivery“)
- Decision Support (Order, „emp of the month“)
- Facebook, Twitter, … (Friends, „post tweet“)

Why use a DBMS?

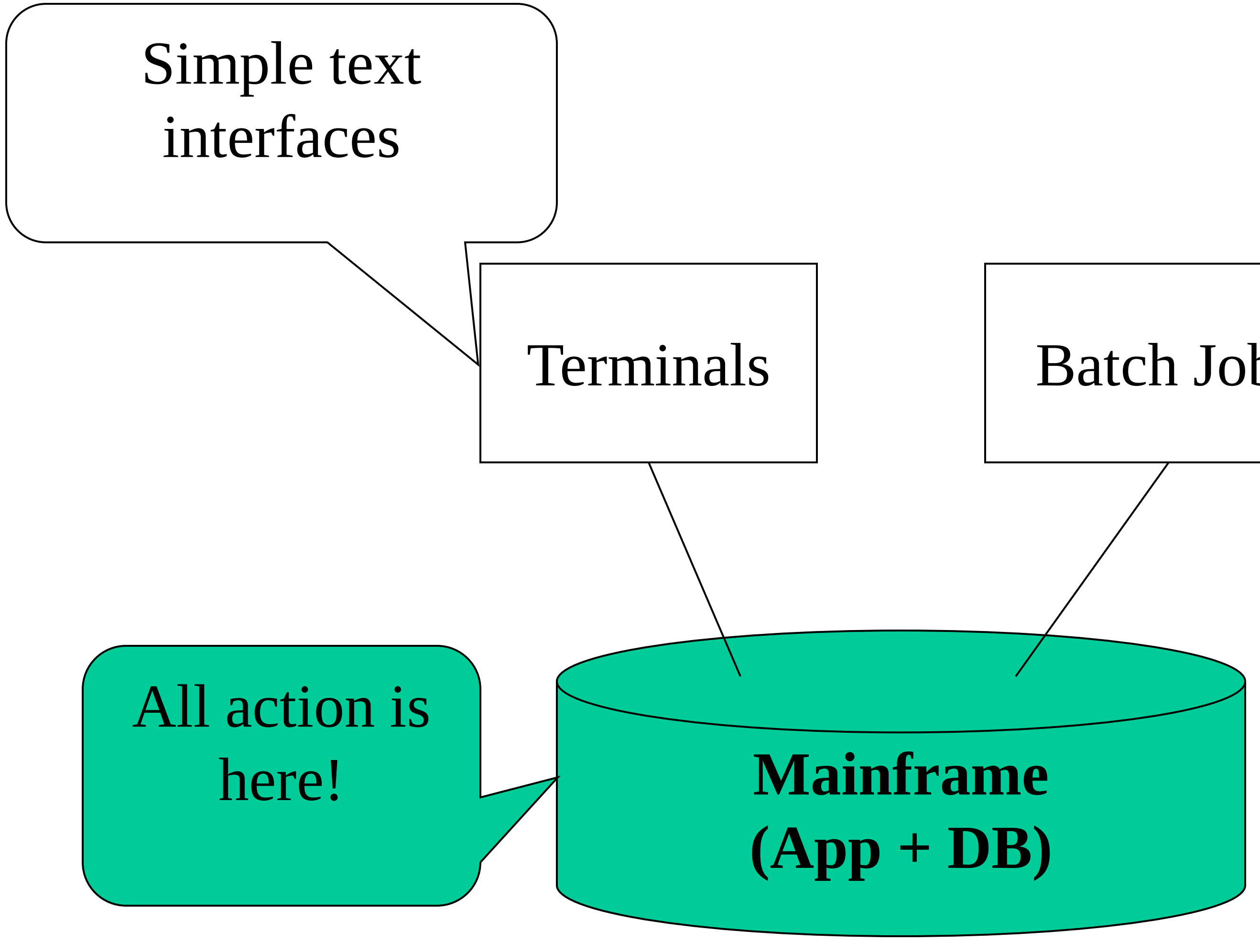
- Avoid redundancy and inconsistency
- Rich (declarative) access to the data
- Synchronize concurrent data access
- Recovery after system failures
- Security and privacy

- Reduce cost and pain to do something useful
  - There is always an alternative!!!

DBMS Architectures

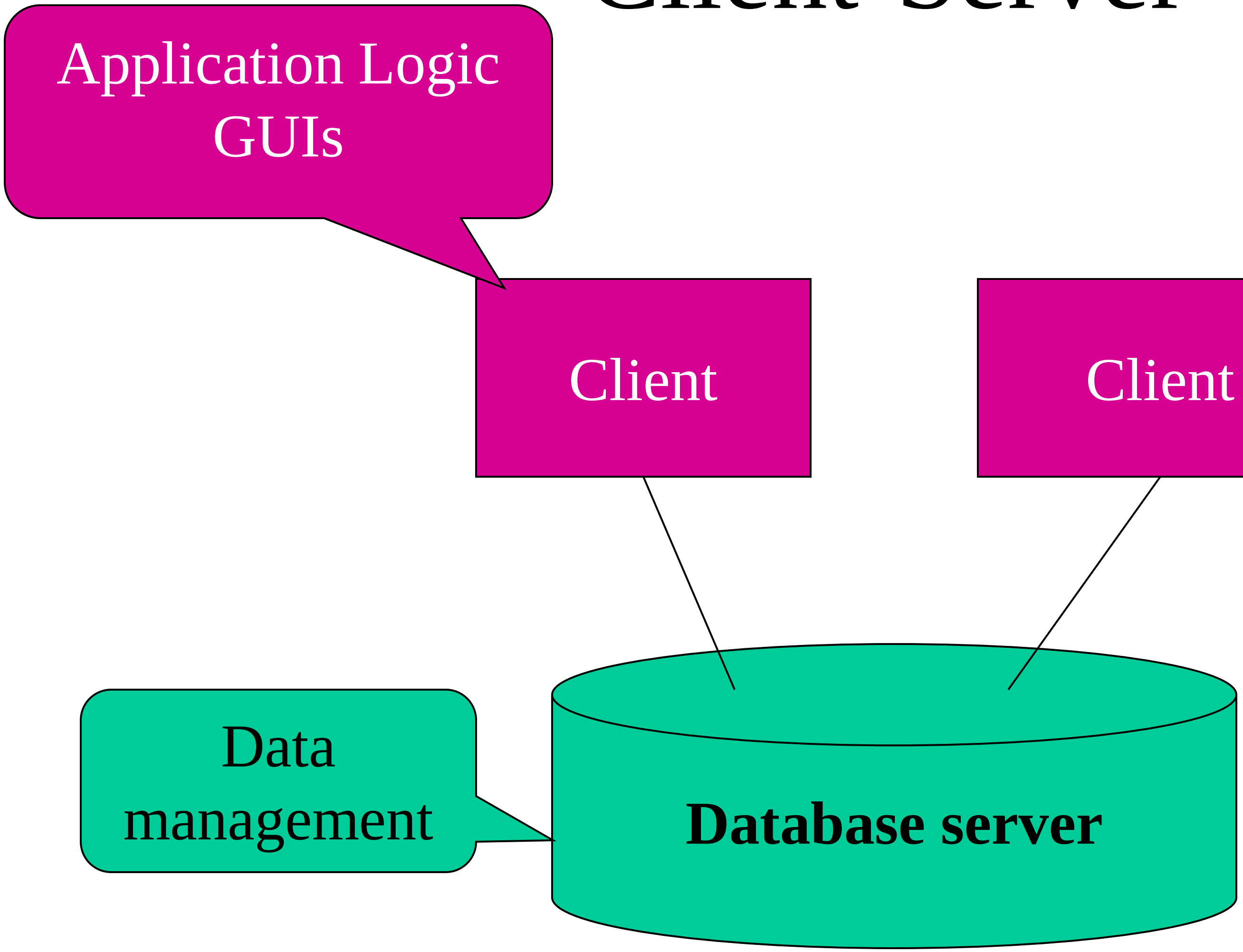
- Question 1: How to slice functionality?
  - Presentation, application logic, data management
- Question 2: How is functionality mapped to HW?
  - Storage, processing, network
- History:
  - Mainframes
  - Client-Server Architecture, Multi-Tier Architectures
  - Parallel Database Systems
  - Data Stream Management Systems
  - Cloud

Mainframes



Simple text interfaces
Terminals
Batch Jobs
All action is here!
Mainframe (App + DB)

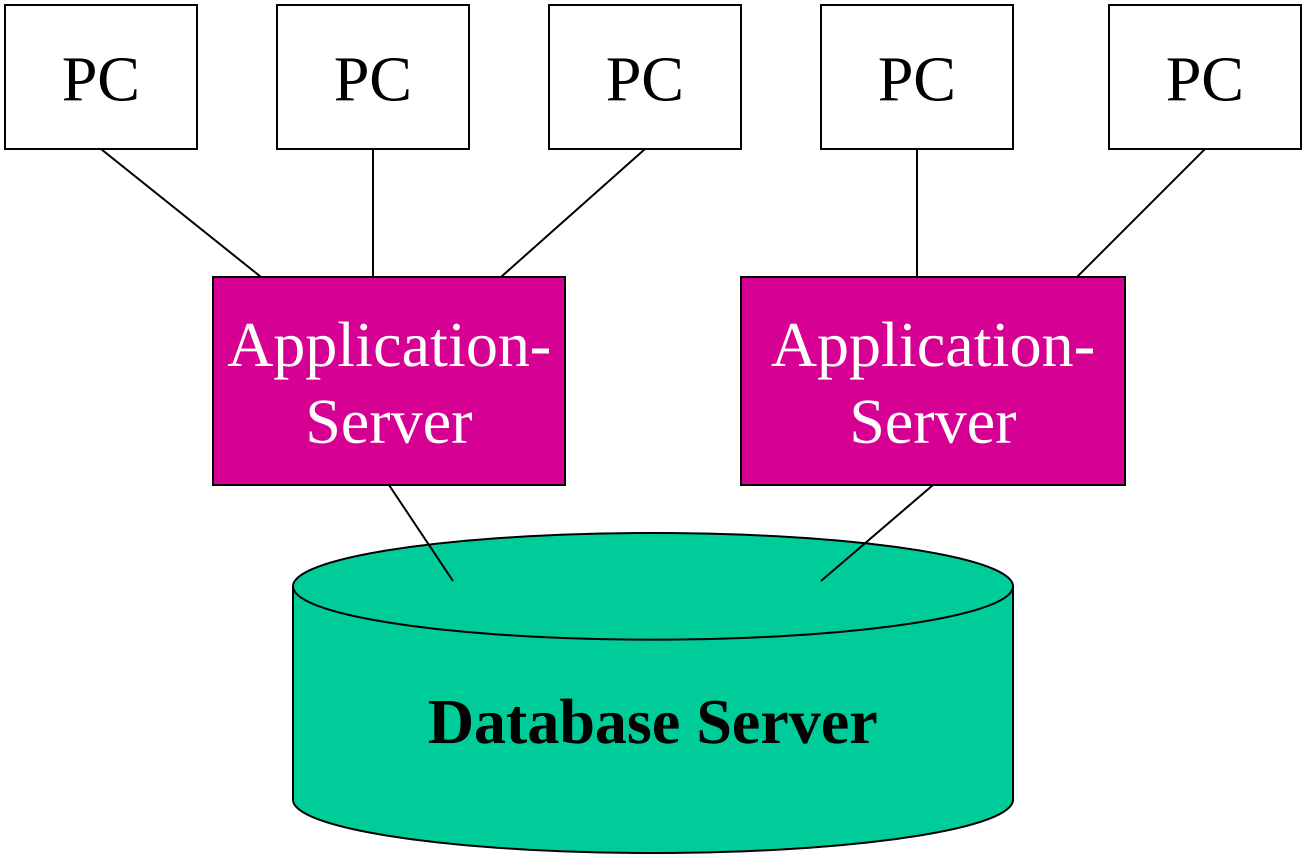
Client-Server



Why Client/Server?

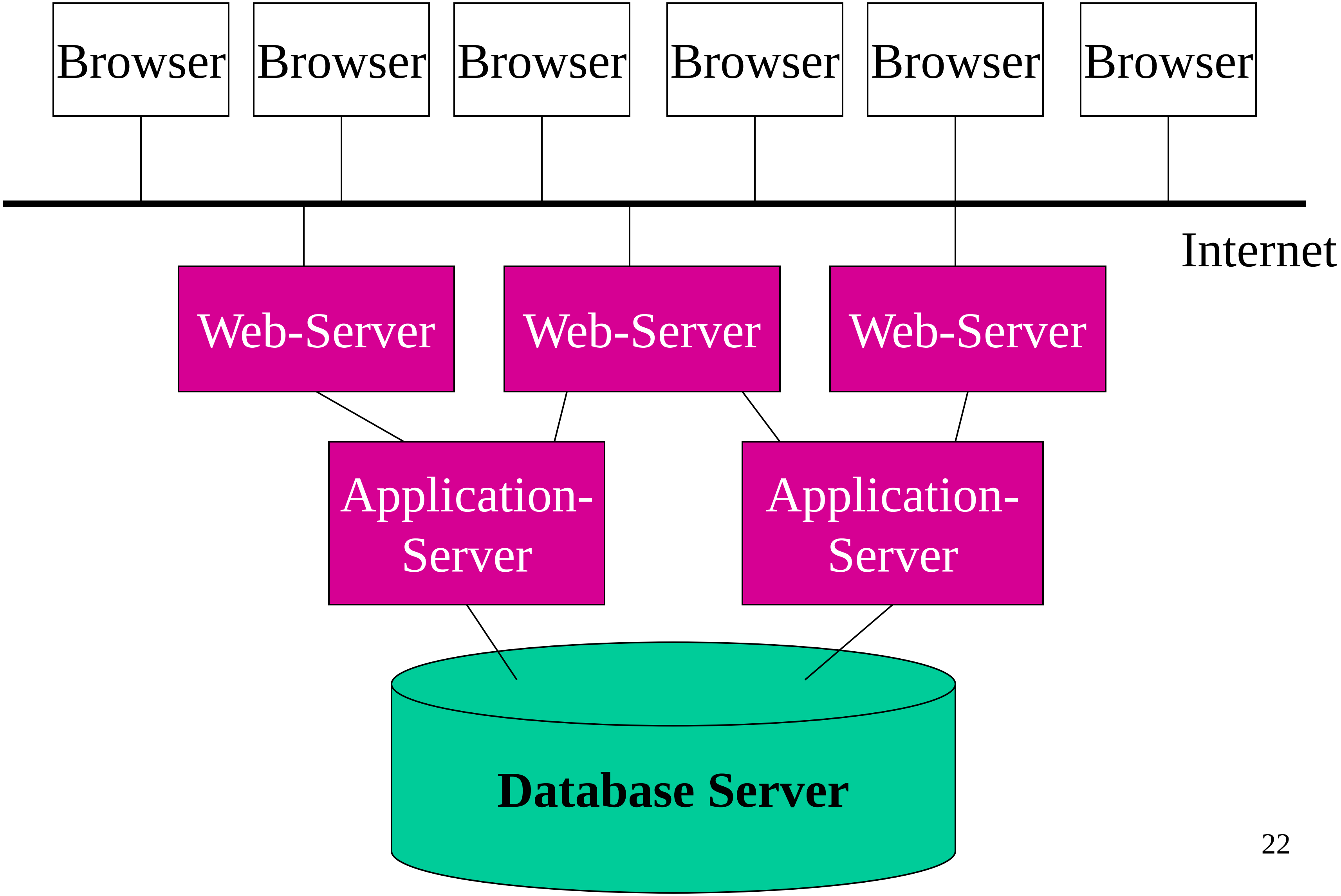
- Scalability: use resources of client machines
  - The more clients / users: the more resources
- Security: server dedicated to protect data
  - No trojan horses possible at server
- Centralized Availability, Administration
  - Same as for mainframe
- Con: Complexity (Caching, etc.)
  - Higher communication cost than mainframe

Three-Tier



PC
PC
PC
PC
PC
Application-Server
Application-Server
Database Server

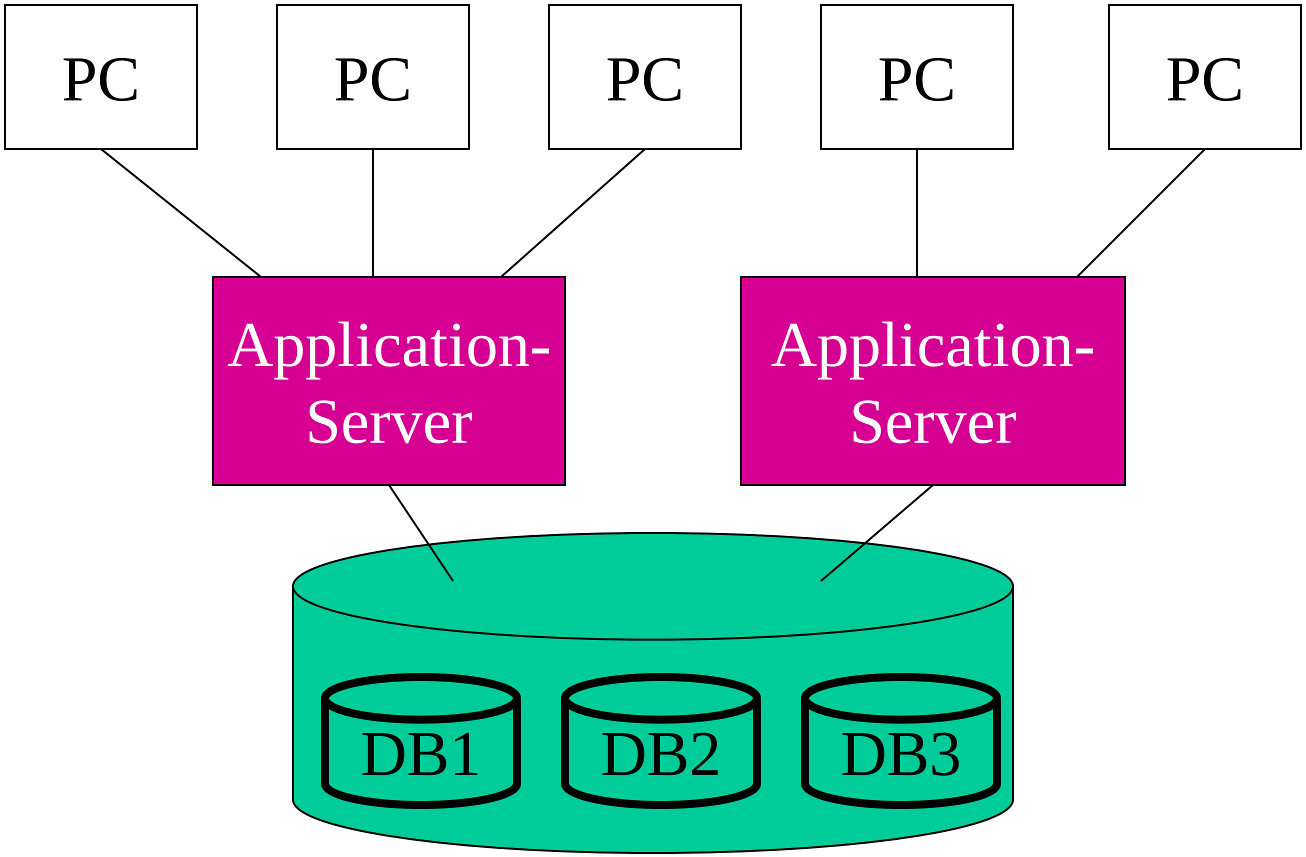
Databases on the Web



Multi-Tier-Architectures

- Software Layering:
  - Every layer implements a different functionality (database, application logic, communication, GUI, ...)
  - Best of breed at each layer (Oracle database, SAP app, Apache Web server, Mac, ...)
- Hardware Layering:
  - Each layer may run on different machines
  - Several layers can run on the same machine
  - Dedicated HW: high IO for DB server
- Scalability
  - Scales great at every layer, except DB

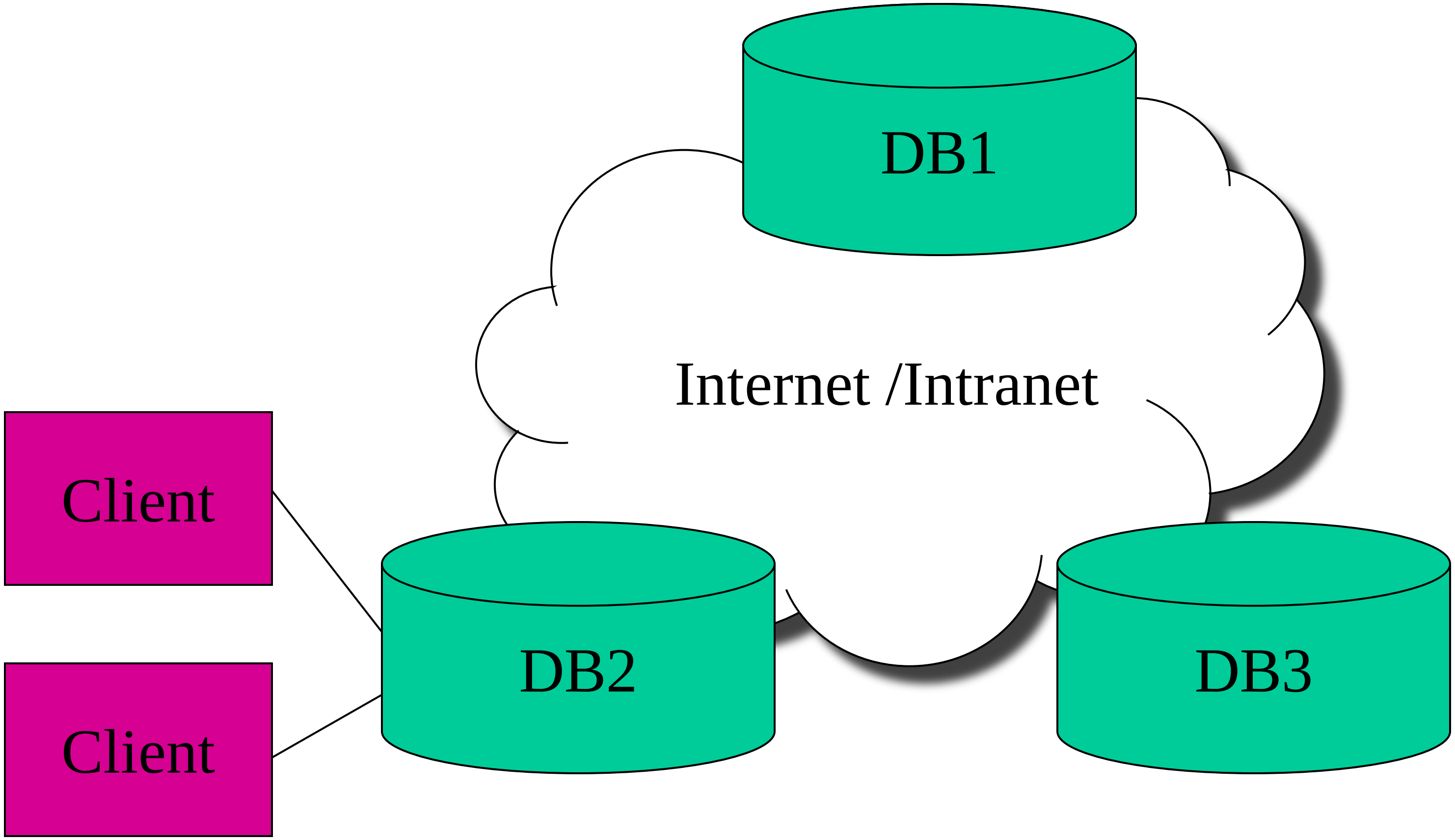
Parallel Database System



Parallel Database System

- Properties:
  - The database system runs on multiple HW nodes
  - Network interconnect assumed to be fast
- Goals:
  - Increase throughput (Inter-Query Parallelism)
  - Reduce latency (Intra-Query Parallelism)
  - Increase availability
  - Reduced cost, Extensibility, Scalability
- Key concern: Transparency
  - Automatic parallelization

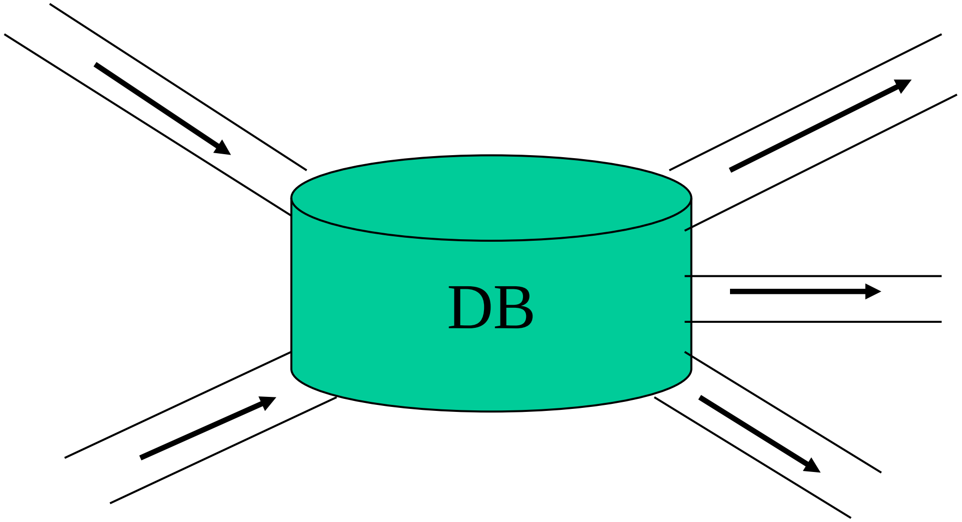
Distributed Database System



Distributed Database System

- Properties:
  - Database system runs on muliple HW nodes
  - Network interconnect is assumed to be slow / expens.
  - Each node behaves autonomously (SOA)
- Goal:
  - Minimize communication cost
- Idea:
  - Store data where it is needed (partition data)
  - Possibly replicate data
- Key concern: Transparency
  - Automatic partitioning and replication

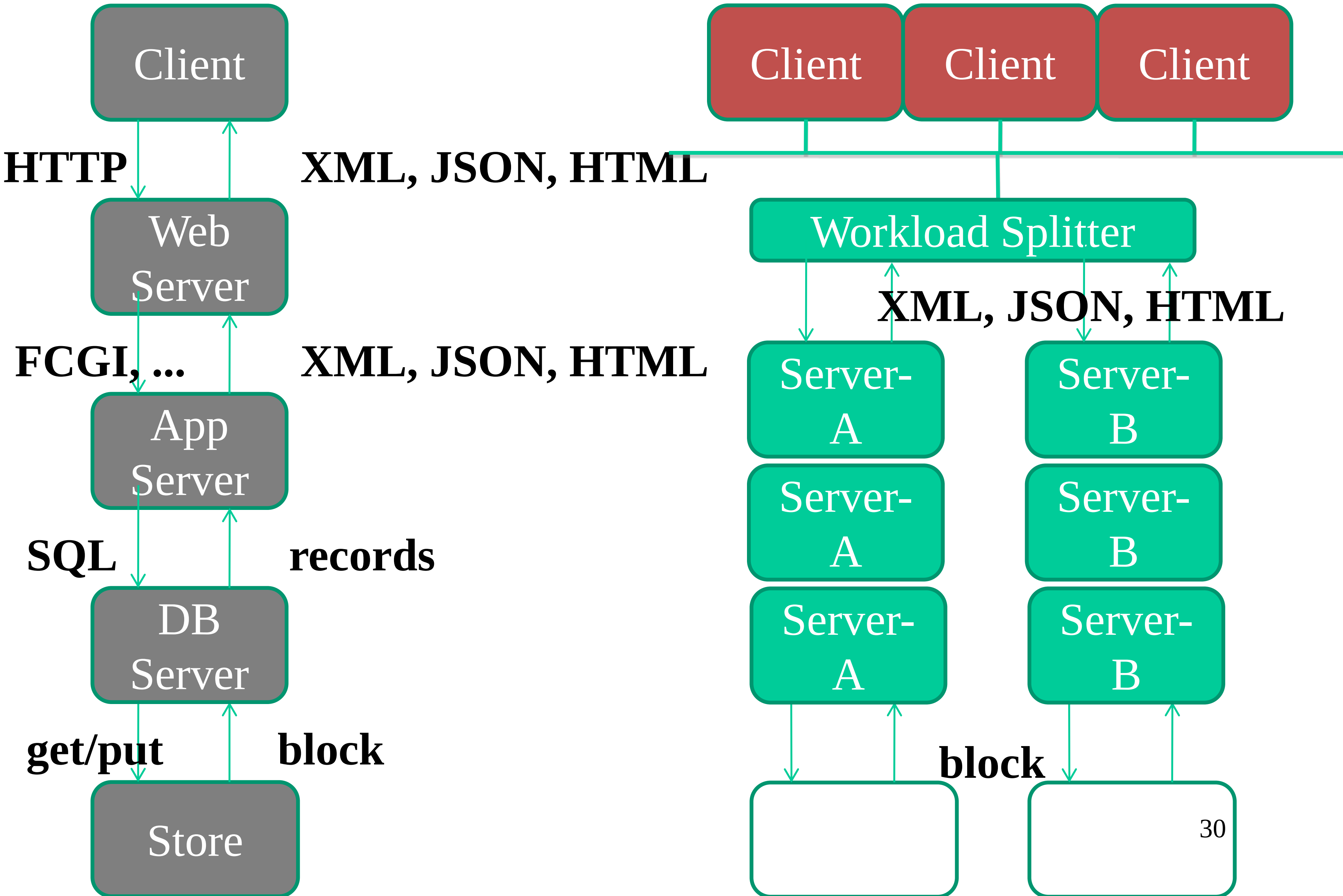
Stream Data Management (Hub)



Stream Data Management

- „Am Anfang war das Wasser…“
  „Alles ist im Fluss …“
  - Actions (and data) are triggered by a stream of events (e.g., purchase orders, sensor measurements, …)
- Data is processed „on the fly“
  - Process events as they occur
  - As needed, store and archive events / affects in DB
  - As needed, precompute reports in DB
- Same abstractions for DSMS as for DBMS
  - E.g., SQL or XQuery as programming languages

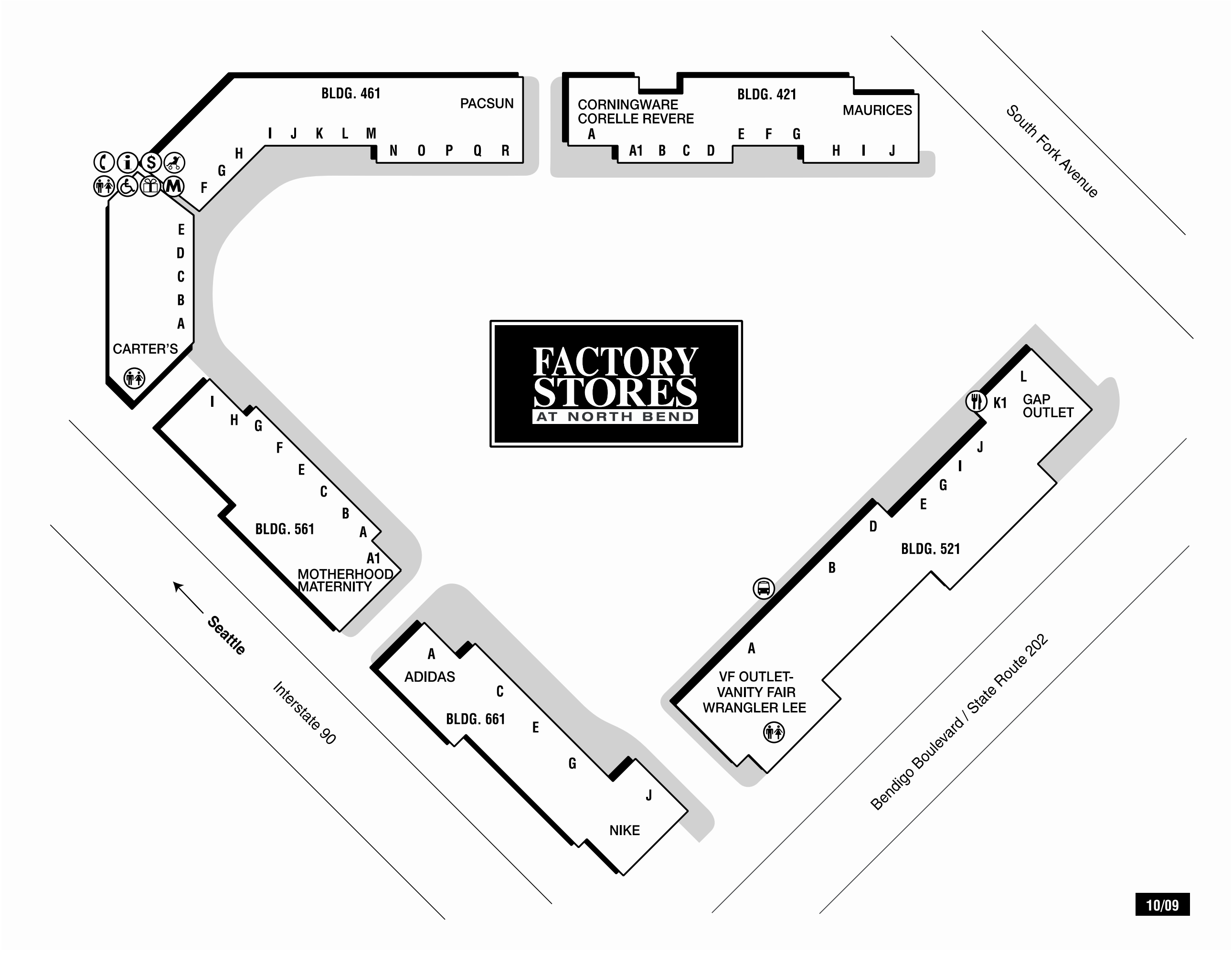
Variant I: Partition Workload by „Tenant“



Partition Workload by „Tenant“

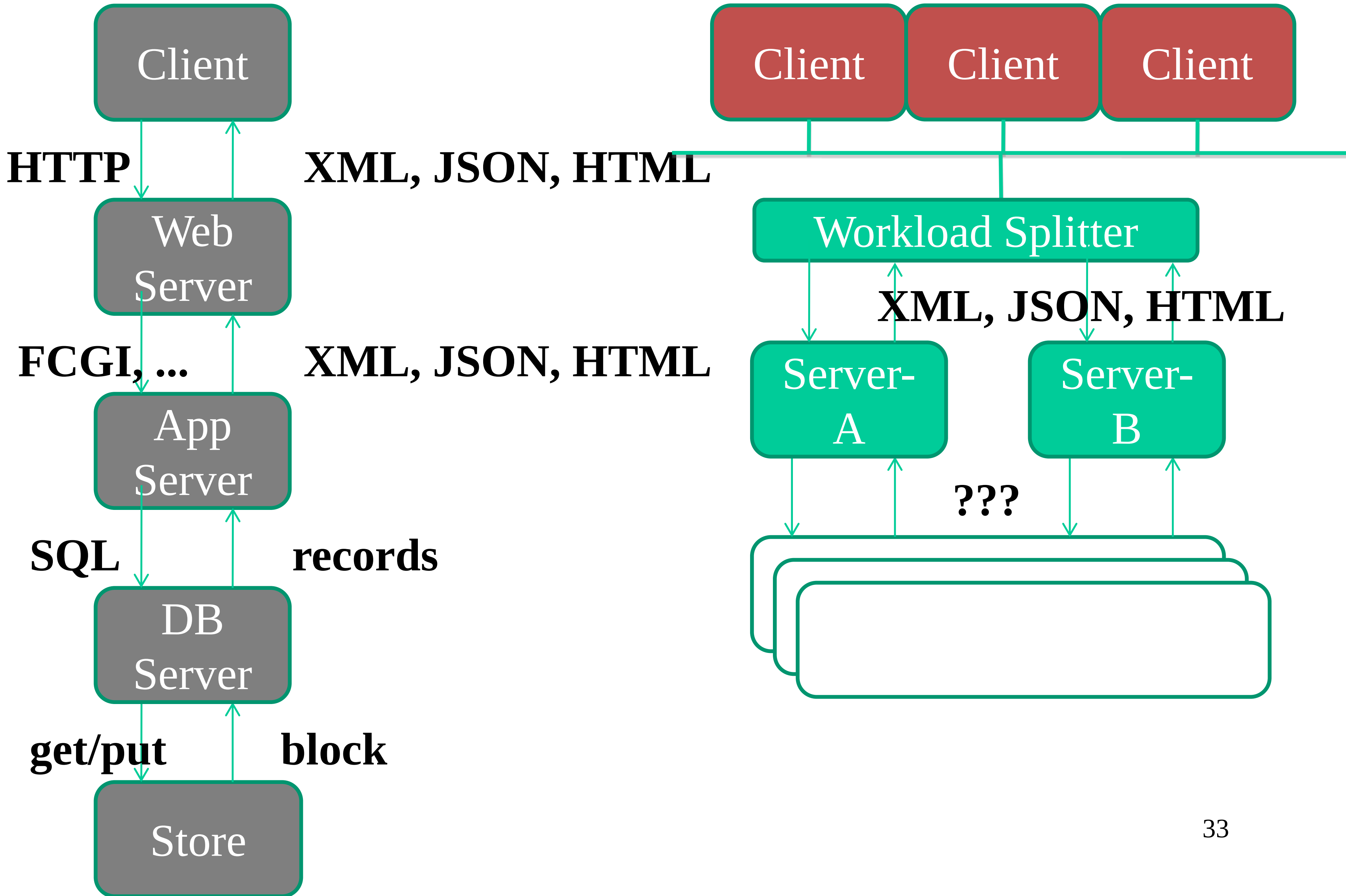
- Principle
  - partition data by „tenant“
  - route request to DB of that tenant
- Advantages
  - reuse existing database stack (RDBMS)
  - flexibility to use DAS or SAN/NAS
- Disadvantages
  - multi-tenant problem *[Salesforce]*
    - optimization, migration, load balancing, fix cost
  - silos: need DB federator for inter-tenant requests
  - expensive HW and SW for high availabilty

Metaphor: Shopping Mall

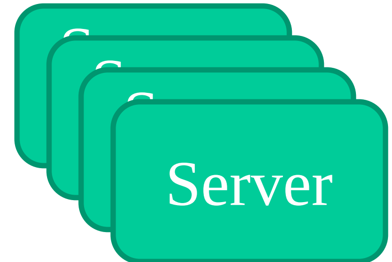


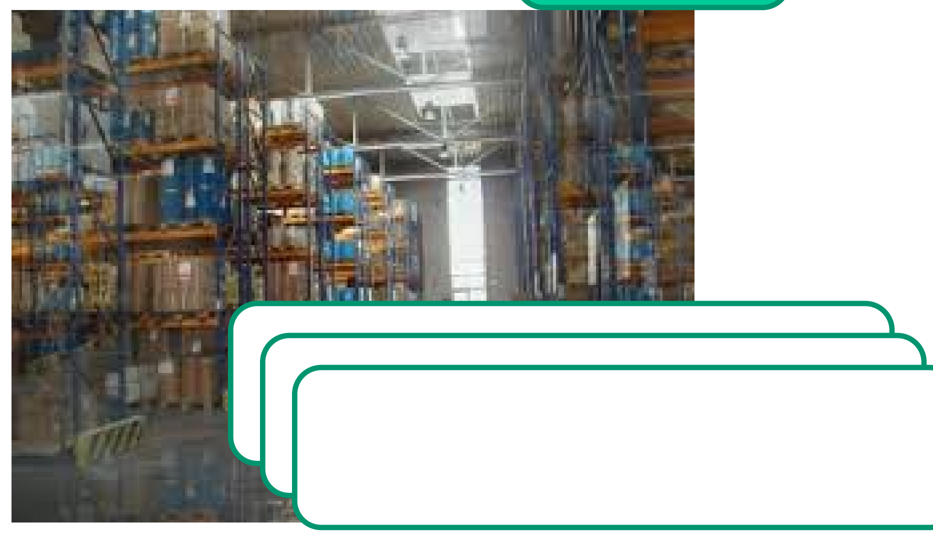
- If a shop is successful, you need to move it!
  - (popularity vs. growth of product assortment)

Variant II: Partition Workload by „Request“



Metaphor: Internet Department Store





- If a product is successful, you stock up its supply
  - Transparent and fine-grained reprovisioning
  - Cost of reprovisioning much lower!!!

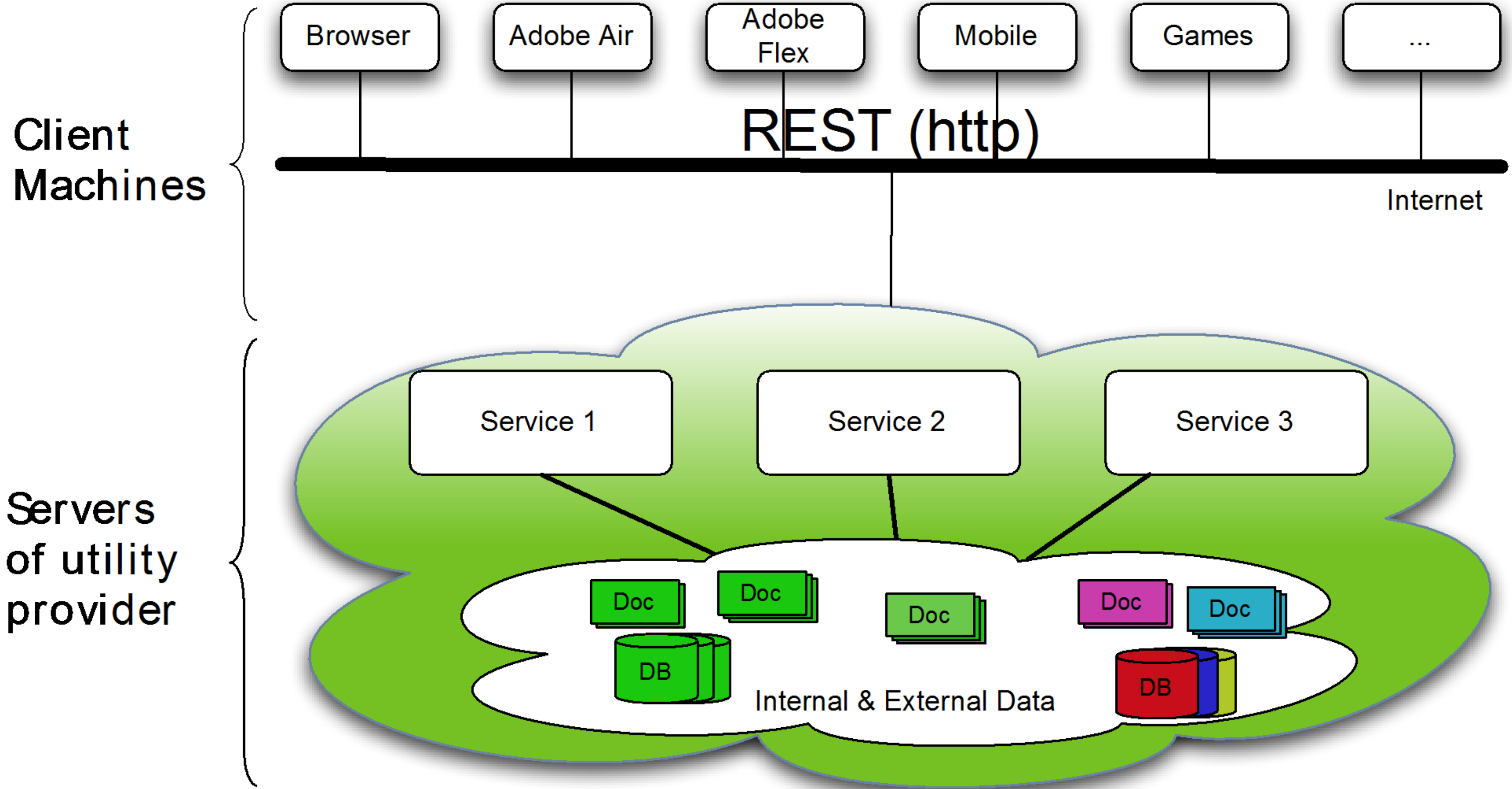
Partition Workload by „Request“

- Principle
  - fine-grained data partitioning by page or object
  - any server can handle any request
  - implement DBMS as a library (not server)
- Advantages
  - avoids disadvantages of Variant I
- Disadvantages
  - new synchronization problem (CAP theorem)
  - whole new breed of systems
  - caching not effective

Mega Trends (IT)

- Cloud Computing
  - logically centralized data
  - physically distributed data
  - commoditization of computing

- Web
  - standardizes representation of data (XML, Unicode)
  - references to data (URI, URL)
  - access to data (HTTP: get, put, post, delete)
  - search for data (Google)
  - extends to physical world: your car keys have a URI

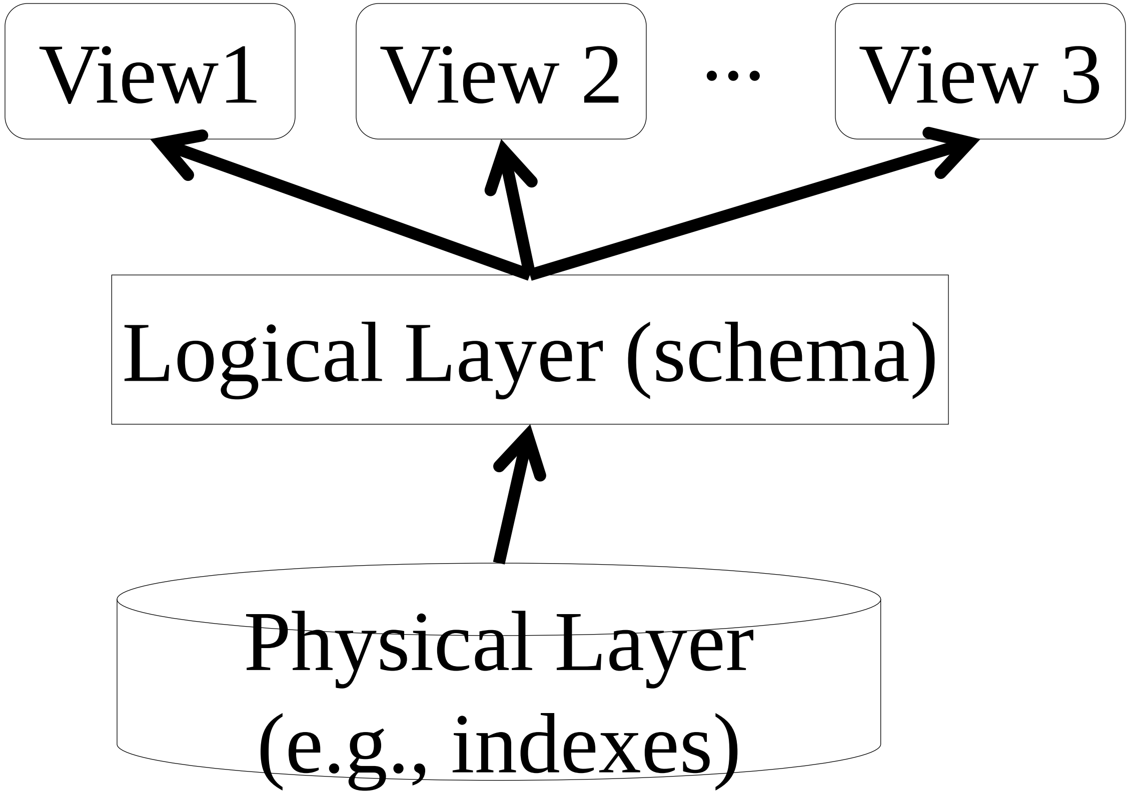

Client Machines
Browser
Adobe Air
Adobe Flex
Mobile
Games
...
REST (http)
Internet
Servers of utility provider
Service 1
Service 2
Service 3
Doc
Doc
Doc
Doc
Doc
DB
DB
Internal & External Data

Database Abstraction Layers

Data Independence

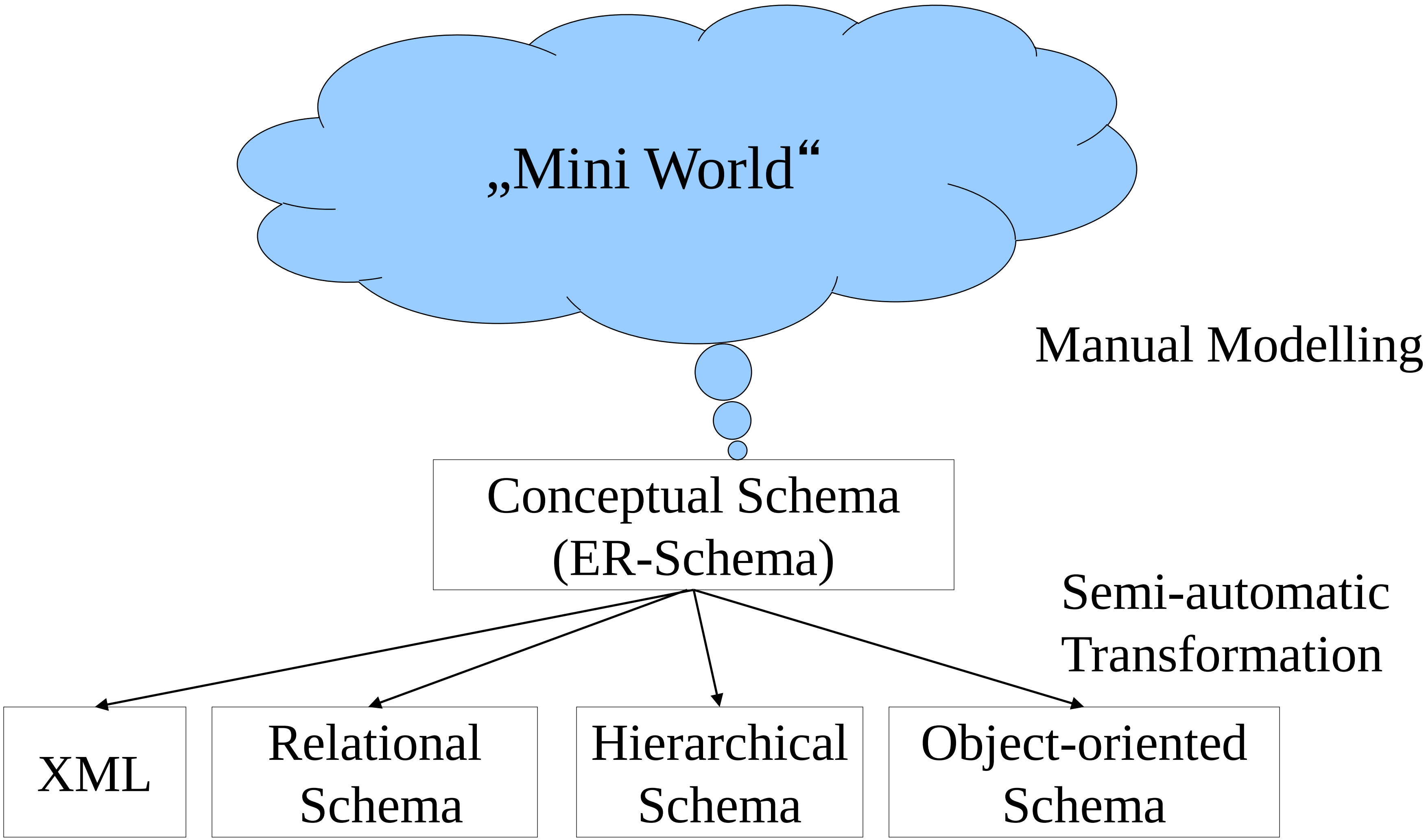
Logical Data Independence

Physical Data Independence

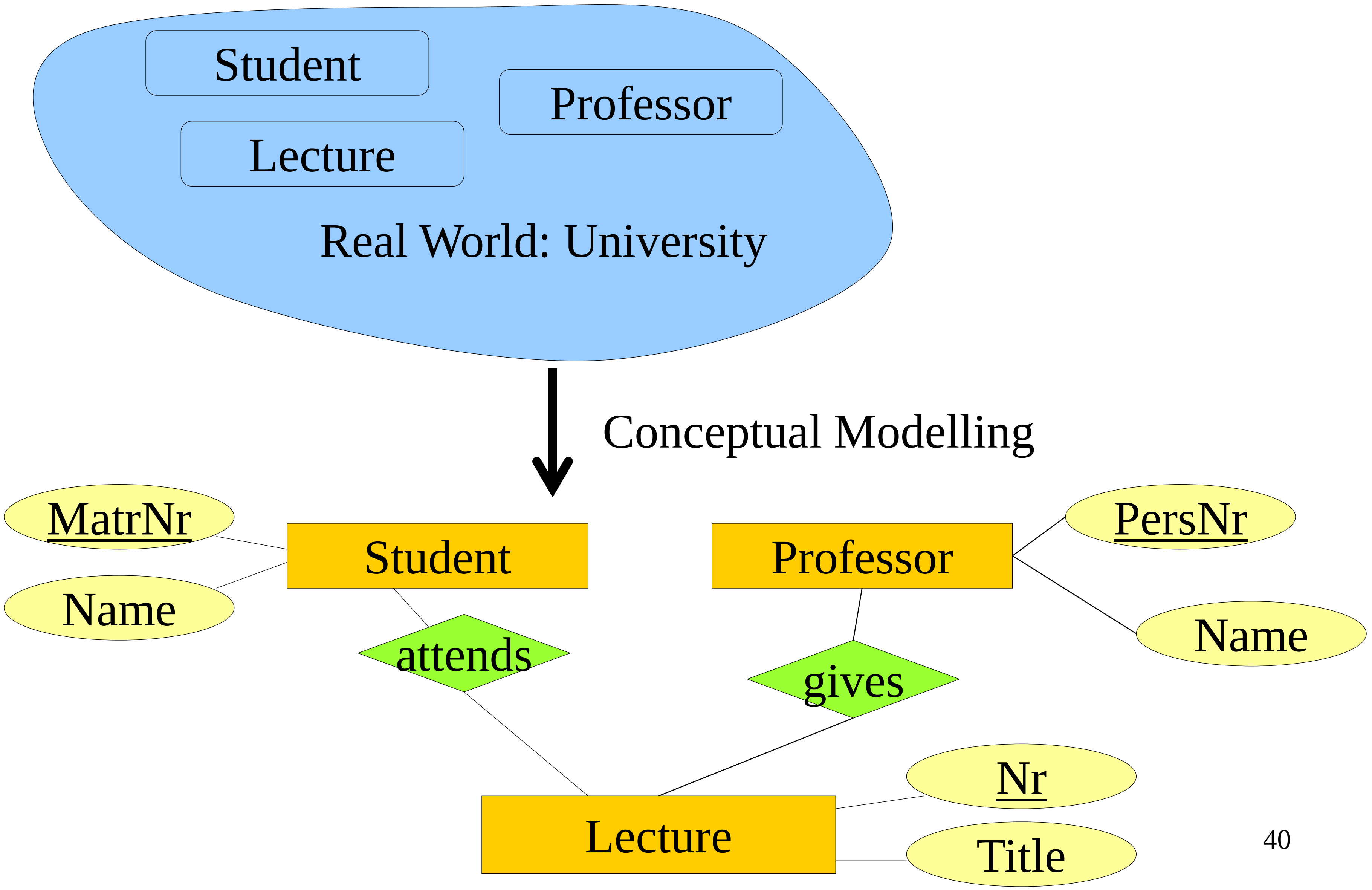


Changes at one layer do not affect another layer!

Data Modelling



Example



Student
Professor
Lecture
Real World: University
Conceptual Modelling
MatrNr
Name
Student
attends
Professor
PersNr
Name
gives
Nr
Lecture
Title

Overview of Data Models

- Network model (e.g., CODASYL/COBOL)
- Hierarchical model (IBM IMS/FastPath)
- Relational model (SQL)
- Object-oriented model (ODMG 2.0)
- Semi-structured model (XML Infoset)
- Deductive model (Datalog, Prolog)

Relational Data Model

Student	
Legi	Name
26120	Fichte
25403	Jonas
...	...

attends	
Legi	Lecture
25403	5022
26120	5001
...	...

Lecture	
Nr	Title
5001	Grundzüge
5022	Glaube und Wissen
...	...

```
Select Name
From Student, attend, Lecture
Where Student.Legi= attend.Legi and
      attend.Lecture= Lecture.Nr and
      Lecture.Title = `Grundzüge´;
```

```
Update   Lecture
   set   Title = `Grundzüge der Logik´
 where   Nr = 5001;
```

Components of a Database System