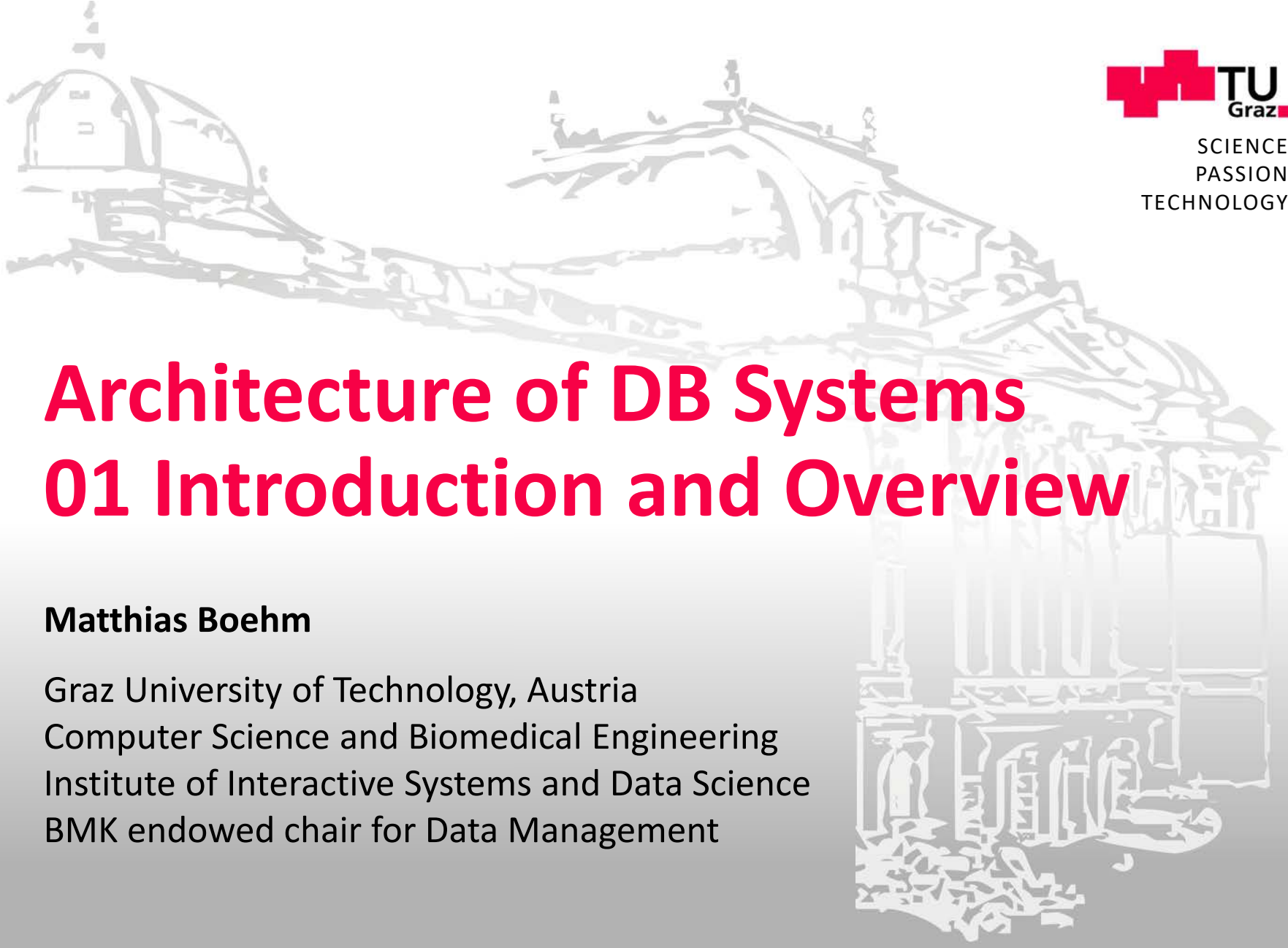


SCIENCE
PASSION
TECHNOLOGY

Architecture of DB Systems
01 Introduction and Overview

Matthias Boehm

Graz University of Technology, Austria
Computer Science and Biomedical Engineering
Institute of Interactive Systems and Data Science
BMK endowed chair for Data Management

Last update: Oct 05, 2021

Announcements/Org

- **#1 Video Recording**
  - Link in TUbe & TeachCenter (lectures will be public)
  - Optional attendance (independent of COVID)
  - Hybrid, in-person but video-recorded lectures
    - HS i5 + Webex: https://tugraz.webex.com/meet/m.boehm

- **#2 COVID-19 Precautions** (HS i5)
  - Room capacity: 24/48 (green/yellow), 12/48 (orange/red)
  - TC lecture registrations (limited capacity, contact tracing)

- **#3 Course Registration** (as of Oct 06)
  - Architecture of Database Systems (ADBS)

WS20/21: **73 (0)**
WS21/22: **94 (0)**

Agenda

- **Data Management Group**
- **Course Organization**
- **Course Motivation and Goals**
- **Course Outline and Projects**
- **Excursus: DAPHNE Project**

Data Management Group

https://damslab.github.io/

About Me

- **09/2018 TU Graz**, Austria
  - BMK endowed chair for data management
  - **Data management for data science**
    (ML systems internals, end-to-end data science lifecycle)

https://github.com/apache/systemds

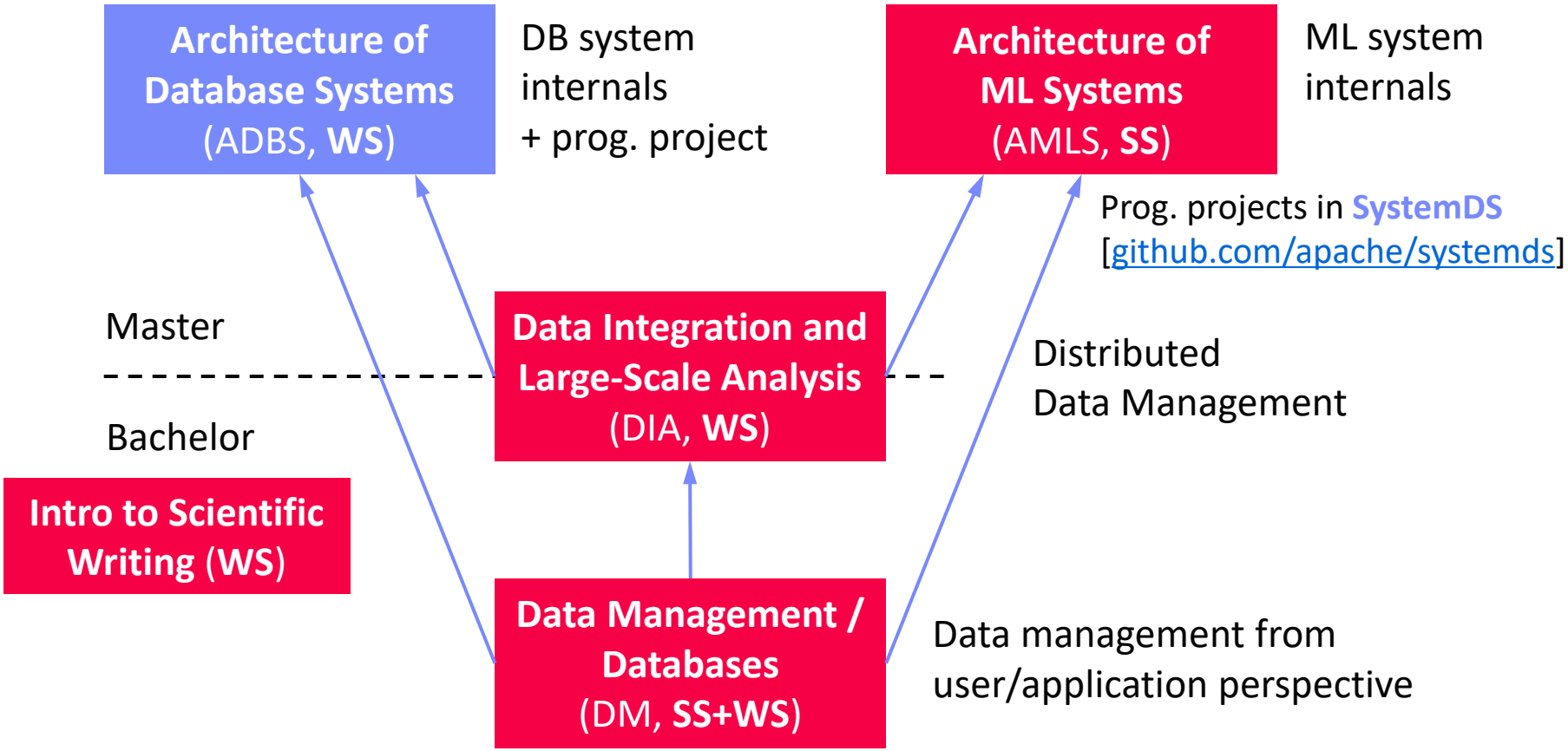
- **2012-2018 IBM Research – Almaden**, USA
  - Declarative large-scale machine learning
  - Optimizer and runtime of **Apache SystemML**

- **2011 PhD TU Dresden**, Germany
  - Cost-based optimization of integration flows
  - Systems support for time series forecasting
  - In-memory indexing and query processing

TECHNISCHE UNIVERSITÄT DRESDEN

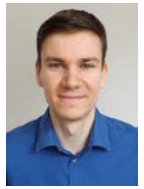
DB group

Data Management Courses



Course Organization

Basic Course Organization



- **Staff**
  - Lecturer: Univ.-Prof. Dr.-Ing. Matthias Boehm, ISDS
  - Assistants: Dr.-Ing. Patrick Damme, M.Tech. Arnab Phani
- **Language**
  - Lectures and slides: **English**
  - Communication and examination: **English**/**German**
- **Course Format**
  - VU 2/1, **5 ECTS** (2x 1.5 ECTS + 1x 2 ECTS), bachelor/master
  - **Weekly lectures** (**Wed 6.15pm**, including **Q&A**), **attendance optional**
  - **Mandatory programming project** (2 ECTS)
  - **Recommended papers** for additional reading on your own
- **Prerequisites**
  - **Preferred:** course Data Management / Databases is very good start
  - **Sufficient:** basic understanding of SQL / RA (or willingness to fill gaps)
  - Basic programming skills in low-level language (C, C++)

Course Logistics

- **Website**
  - https://mboehm7.github.io/teaching/ws2122_adbs/index.htm
  - All course material (lecture slides) and dates
- **Video Recording Lectures (TUbe)**

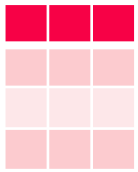
- **Communication**
  - **Informal language** (first name is fine)
  - Please, **immediate feedback** (unclear content, missing background)
  - Newsgroup: N/A – email is fine, summarized in following lectures
  - **Office hours:** by appointment or after lecture
- **Exam**
  - **Completed programming project** (checked by me/staff)
  - **Final written exam** (oral exam if <15 students take the exam)
  - **Grading** (30% project/exercises completion, 70% exam) **NEW**

Course Logistics, cont.

- **Course Applicability**
  - **Master** programs computer science (CS), as well as software engineering and management (SEM)
    - Catalog Data Science (elective course in major/minor)
    - Catalog Software Technology (elective course in major/minor)
  - **Free subject course** in any other study program or university

Course Motivation and Goals

History 1970/80s (relational)



SQL Standard (SQL-**86**)

Oracle, IBM DB2, Informix, Sybase → MS SQL

SEQUEL

QUEL

System R @ IBM Research – Almaden (Jim Gray et al., Turing Award '98)

Ingres @ UC Berkeley (Stonebraker et al., Turing Award '14)

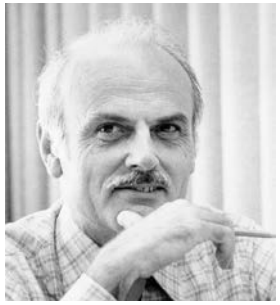
Tuple Calculus

Relational Algebra

Relational Model

Goal: Data Independence (physical data independence)

- Ordering Dependence
- Indexing Dependence
- Access Path Dependence



Edgar F. "Ted" Codd @ IBM Research (Turing Award '81)

[E. F. Codd: A Relational Model of Data for Large Shared Data Banks. Comm. ACM 13(6), **1970**]

Success of SQL / Relational Model

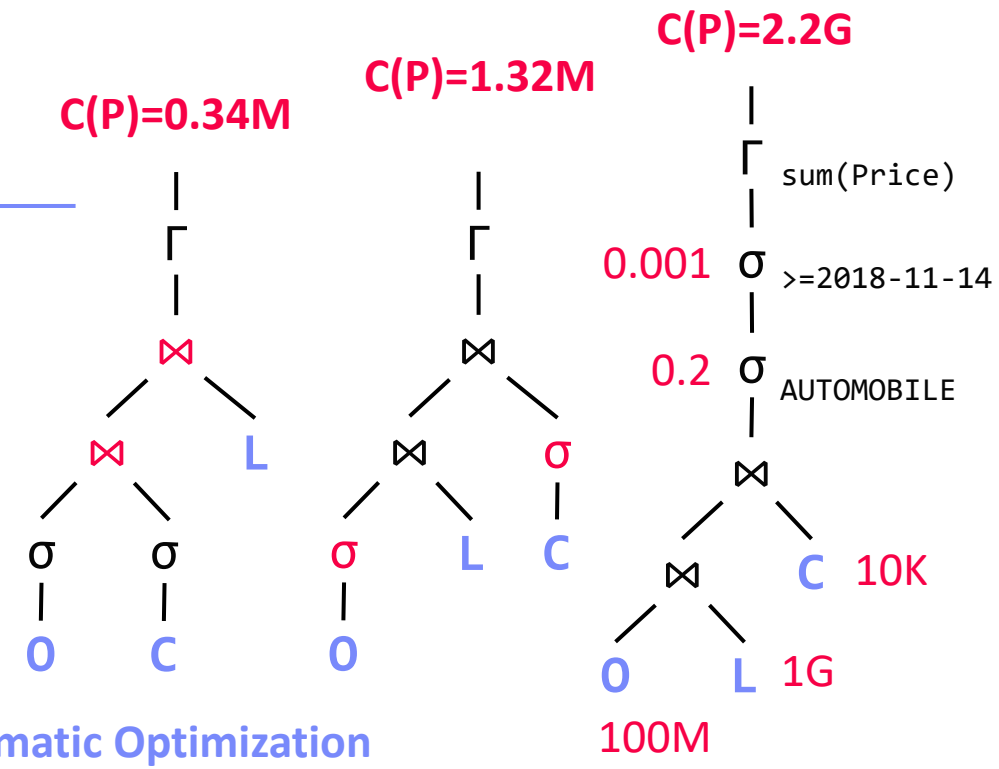
Query:

```
SELECT O_OID, sum(L_Price)
FROM Orders, Lineitem, Customer
WHERE O_OID = L_OID AND O_CID = C_CID
   AND O_Odate >= '2018-11-14'
   AND C_Msegment = 'AUTOMOBILE'
GROUP BY O_OID
```

#1 Declarative:
what not how

#2 Flexibility:
closure property
→ composability

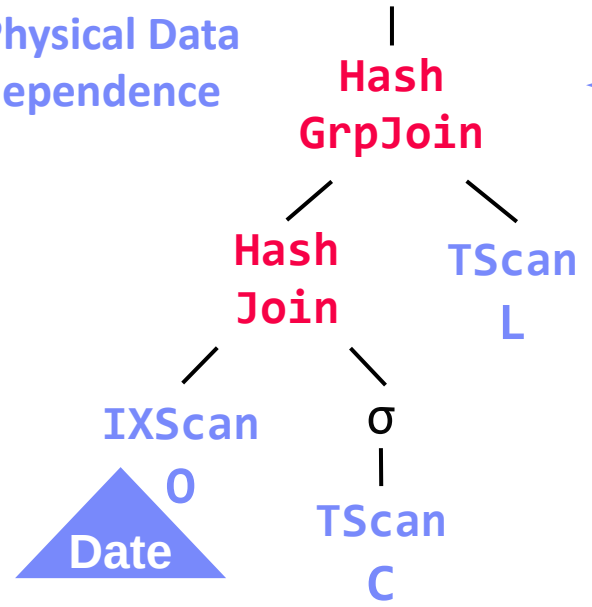
Logical Query Plans



#3 Automatic Optimization

#4 Physical Data Independence

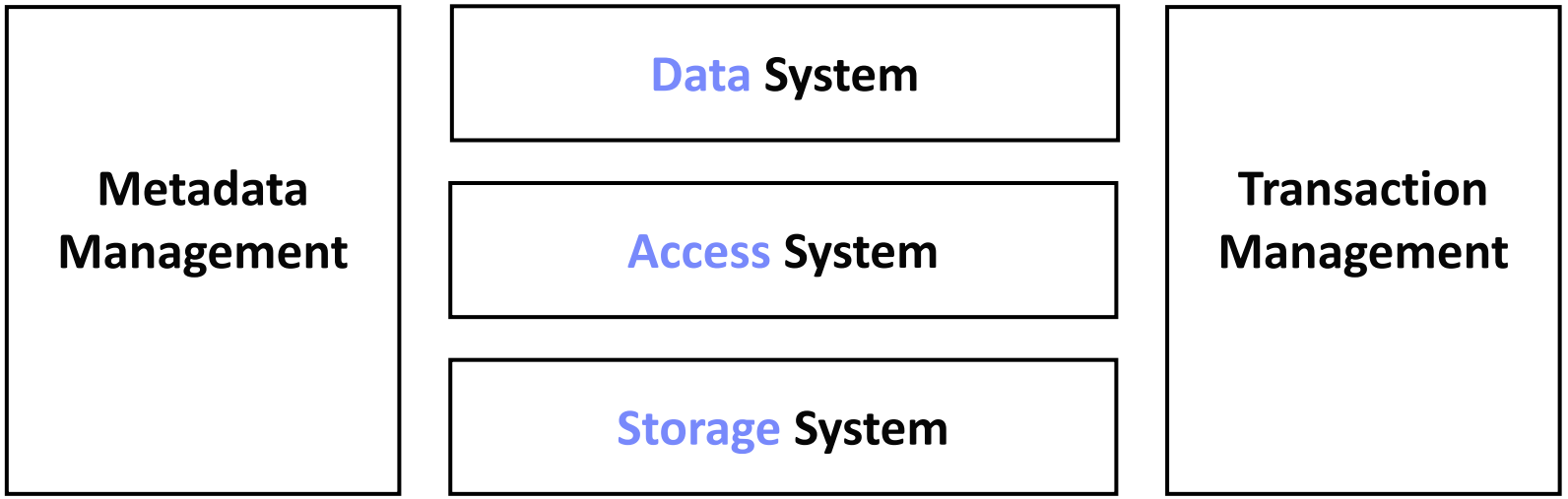
Physical Query Plan



DBMS Architecture

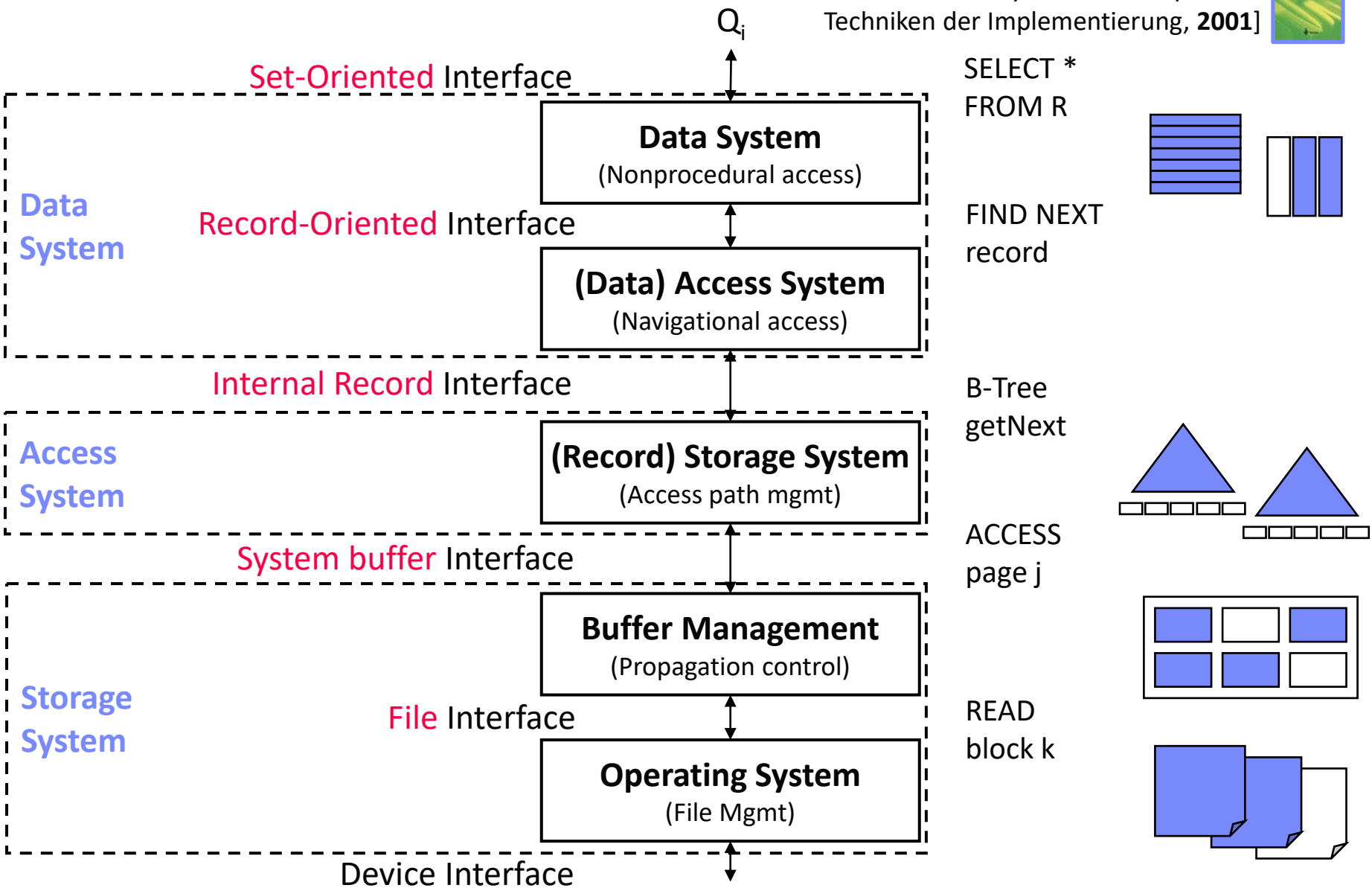
[Theo Härder, Erhard Rahm: Datenbanksysteme: Konzepte und Techniken der Implementierung, **2001**]

- **Coarse-grained System Architecture**



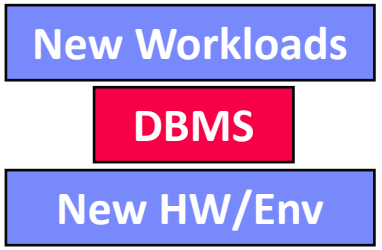
DBMS Architecture, cont.

[Theo Härder, Erhard Rahm: Datenbanksysteme: Konzepte und Techniken der Implementierung, **2001**]



Course Goals

- **Constantly Changing Environment**
  - New application and **data analysis workloads**
  - Heterogeneous and changing **hardware characteristics**



- **#1 Architecture and internals of traditional/modern DB systems**
- **#2 Understanding of DB characteristics → better evaluation / usage**
- **#3 Understanding of effective techniques → build/extend DB systems**
  (these fundamental techniques are **broadly applicable** in other systems)

Course Outline and Projects

Course Outline

A: System Architecture and Data Access

- **01 Introduction and Overview** [Oct 06]
- **02 DB System Architectures** [Oct 13]
- **03 Data Layouts and Bufferpool Management** [Oct 20]
- **04 Index Structures and Partitioning** [Oct 27]
- **05 Compression Techniques** [Nov 03, Patrick]

B: Query Processing and Optimization

- **06 Query Processing (operators, execution models)** [Nov 10]
- **07 Query Compilation and Parallelization** [Nov 17]
- **08 Query Optimization (rewrites, costs, join ordering)** [Nov 24]
- **10 Adaptive Query Processing** [Dec 01]

Course Outline, cont.

C: Emerging Topics

- **11 Cloud Database Systems** [Dec 15]
- **12 Modern Concurrency Control** [Jan 12, Arnab]
- **13 Modern Storage and HW Accelerators** [Jan 19]
- **14 Selected Trends & Project Results** [Jan 26, Patrick & Arnab]

Overview Programming Project

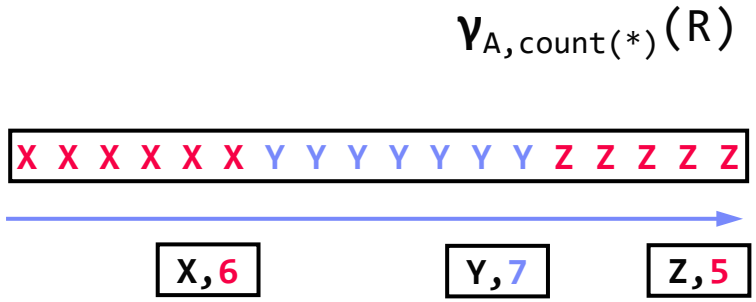
- **Team**
  - **1-3 person teams** (w/ clearly separated responsibilities)
- **Task: Efficient Group-by Aggregation**
  - Column-oriented frame storage w/ materialized intermediates
  - Multi-threaded group-by aggregation w/ multiple group-by columns, additive aggregation functions, different data types / characteristics
  - C test / performance suites → correct and minimum perf
  - Programming language: no restrictions, but **C or C++** recommended
- **Timeline**
  - **Oct 19:** Test drivers, reference implementation available
  - **Jan 21, 11.59pm:** Final programming project deadline
- **Prices Top-k Submissions**
  - Research assistant positions / payed master theses in **DAPHNE project**

Overview Programming Project, cont.

- **Recap: Classification of Aggregates (DM, DIA)**
  - Additive, semi-additive, additively-computable, others

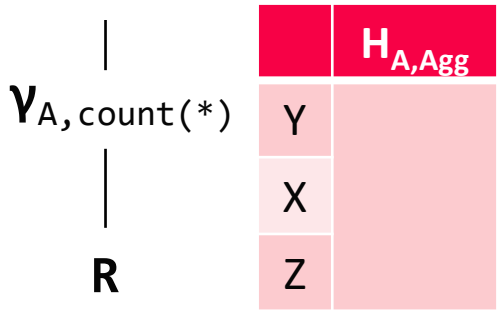
- **#1 Sort Group-By**
  - Similar to sort-merge join (`Sort`, `GroupAggregate`)
  - Sorted group output

sort
O(N log N)
aggregate
O(N)



- **#2 Hash Group-By**
  - Similar to hash join (`HashAggregate`)
  - Higher temporary memory consumption
  - Unsorted group output
  - **#1** w/ **tuple grouping**
  - **#2** w/ **direct aggregation** (e.g., count)
  - **Beware:** cache-unfriendly if many groups (size(H) > L2/L3 cache)

build & agg
O(N)



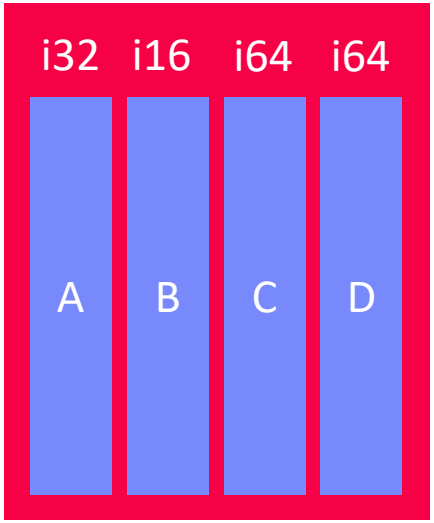
Overview Programming Project, cont.

- **API Sketch**
  - Materialized inputs, outputs
  - Multiple group by attributes
  - Multiple aggregated attributes
  - Aggregation functions: sum/min/max

$$\gamma_{sum(C)}(R)$$
$$\gamma_{A,sum(C)}(R)$$
$$\gamma_{A,B,sum(C)}(R)$$
$$\gamma_{A,B,sum(C),sum(D)}(R)$$

- **Data Characteristics**
  - Frames w/ column-oriented storage
  - Data Types: INT16, INT32, INT64
  - Varying # distinct values, skew, missing values

```
getNumRows()
getNumCols
getRow(size_t r)
getCol(size_t c)
```



- **Performance Target**
  - Relative to [naïve, tuned] reference impl (TBD)
  - **Perf target scaled by team size**

DAPHNE:

Integrated **D**ata **A**nalysis **P**ipelines for Large-Scale DM, **H**PC, and ML

Motivation, Vision, and System Architecture

https://daphne-eu.github.io/

DAPHNE

[Louvre, Paris]

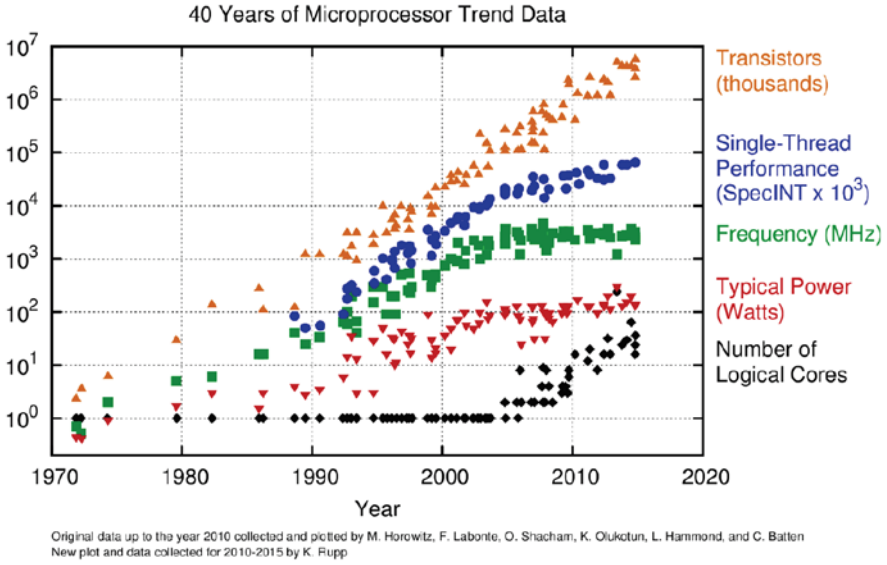
HW Challenges

[S. Markidis, E. Laure, N. Jansson, S. Rivas-Gomez and S. W. D. Chien: Moore's Law and Dennard Scaling]

- **#1 End of Dennard Scaling** (~2005)
  - **Law:** power stays proportional to the area of the transistor
  - Ignored leakage current / threshold voltage
    → **increasing power density S^2** (power wall, heat) → stagnating frequency

$P = \alpha CFV^2$ (power density 1)
(P .. Power, C .. Capacitance, F .. Frequency, V .. Voltage)

- **#2 End of Moore's Law** (~2010-20)
  - **Law:** #transistors/performance/ CPU frequency doubles every 18/24 months
  - Original: # transistors per chip doubles every two years **at constant costs**
  - Now increasing costs (10/7/5nm)

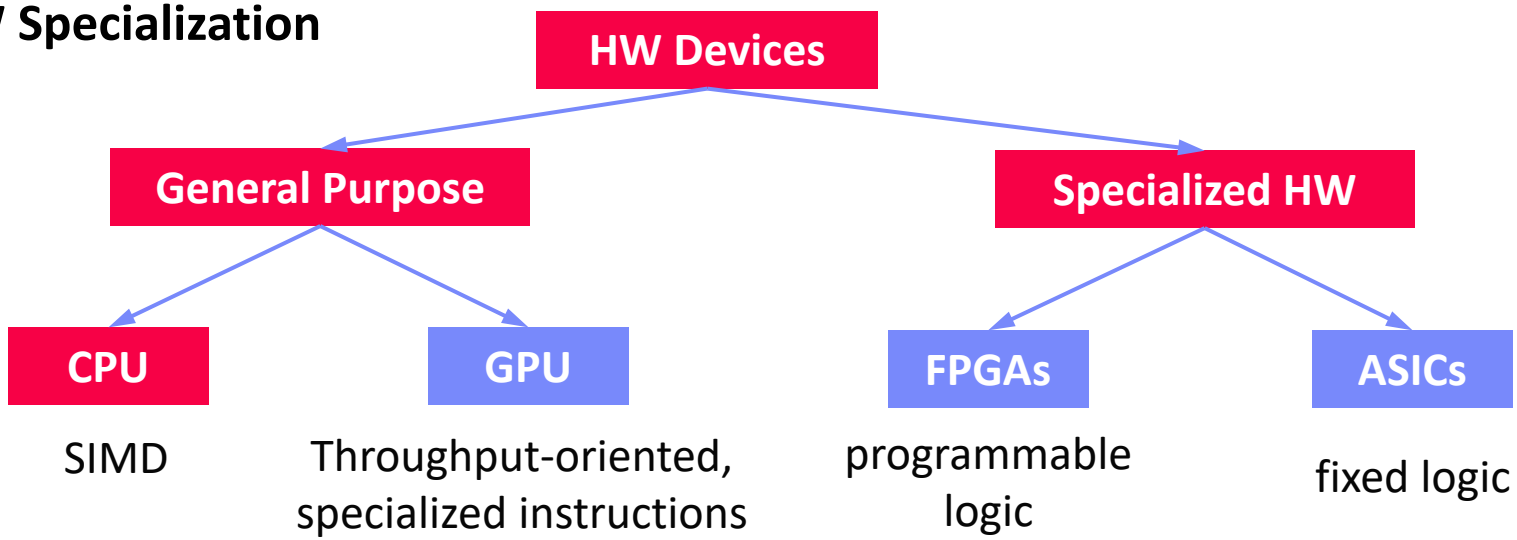


- **#3 Amdahl's Law** (speedup limitations)

→ Dark Silicon and Specialization

HW Challenges, cont.

- **HW Specialization**



- **Additional Specialization**
  - **Data Transfer & Types:** e.g., low-precision, quantization; new data types
  - **Sparsity Exploitation:** e.g., sparsification, exploit across operations, defer weight decompression just before instruction execution
  - **Near-Data Processing:** e.g., operations in main memory, storage class memory (SCM), secondary storage (e.g., SSDs), and tertiary storage (e.g., tapes)

➔ **Heterogeneity and Utilization Challenges**

Productivity & Overhead Challenges

- **Productivity and Systems Support**
  - ML pipelines and HPC still require substantial manual effort
  - Different programming models, cluster environments, redundancy
- **Overhead and Low Utilization**
  - Separate, statically provisioned clusters
  - Lack of interoperability, coarse-grained file exchange
- **Lack of Common System Infrastructure**
  - Conceptual ideas reused, but redundantly implemented
  - Open-source systems in DM, ML, HPC often company-controlled

Productivity & Specialization
from use cases to HW infrastructure

➔ DAPHNE Overall Objective:
An open and extensible systems infrastructure (DM/ML/HPC)

DAPHNE Consortium

- **Data Management**
- **High-Performance Computing (HPC)**
- **ML Systems**
- **ML/NLP/ Sim&Optimization**

- **Application Domains**
- **Academia and Industry**

➔ **Advancements through Creation**

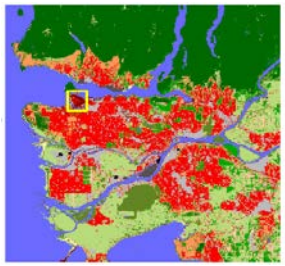
- Know-Center GmbH (**coordinator**), Austria
- AVL List GmbH, Austria
- Deutsches Zentrum fuer Luft- und Raumfahrt e.V., Germany
- Eidgenoessische Technische Hochschule Zuerich, Switzerland
- Hasso-Plattner-Institut for Digital Engineering gGmbH, Germany
- Institute of Communication and Computer Systems, Greece
- Infineon Technologies Austria AG, Austria
- intel , Intel Technology Poland sp. z o.o., Poland
- IT-Universitetet i København, Denmark
- Kompetenzzentrum Automobil- und Industrieelektronik GmbH, Austria
- Technische Universität Dresden, Germany
- Univerza v Mariboru, Slovenia (EuroHPC Center)
- Universitaet Basel, Switzerland

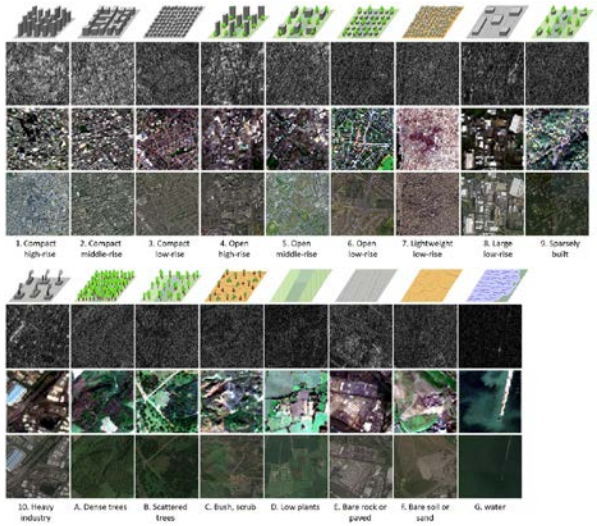
Use Cases

[Xiao Xiang Zhu et al: So2Sat LCZ42: A Benchmark Dataset for the Classification of Global Local Climate Zones. **GRSM 8(3) 2020**]

[So2Sat LC42 Dataset https://mediatum.ub.tum.de/1454690]

- **DLR Earth Observation**
  - **ESA Sentinel-1/2** datasets → 4PB/year
  - Training of local climate zone classifiers on **So2Sat LCZ42** (15 experts, 400K instances, 10 labels each, 85% confidence, ~55GB H5)
  - **ML pipeline:** preprocessing, ResNet20, climate models



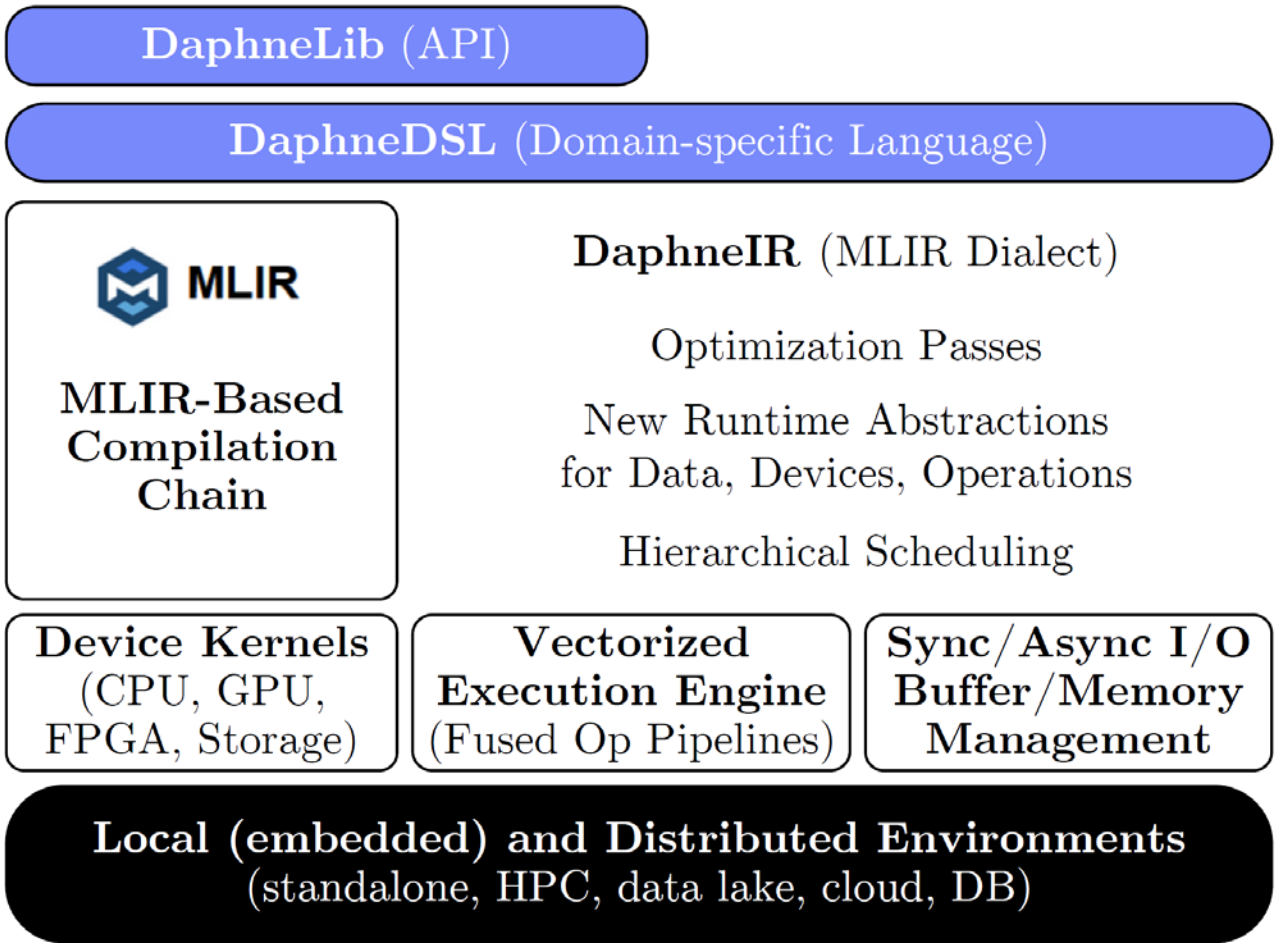


- **KAI Semiconductor Device Reliability**
- **IFAT Semiconductor Ion Beam Tuning**
- **AVL Vehicle Dev Process** (ejector geometries, fuel cells)

- **ML-assisted simulations** (e.g., fluids, materials) + **data analysis**

System Architecture

DAPHNE



Summary and Q&A

- **Course Goals**
  - **#1** Architecture and internals of traditional/modern DB systems
  - **#2** Understanding of DB characteristics → **better evaluation / usage**
  - **#3** Understanding of effective techniques → **build/extend DB systems** (these fundamental techniques are **broadly applicable** in other systems)
- **Programming Project**
  - Column-oriented frame storage w/ materialized intermediates
  - **Multi-threaded group-by aggregation** w/ multiple group-by columns, additive aggregation functions, different data types / characteristics
- **Next Lectures**
  - 02 **DB System Architectures** [Oct 13]
  - 03 **Data Layouts and Bufferpool Management** [Oct 20]
  - 04 **Index Structures and Partitioning** [Oct 27]
  - 05 **Compression Techniques** [Nov 03]