Mosaics in Big Data

Database Systems and Information Management – Trends and a Vision

Volker Markl

Professor of Computer Science, TU Berlin

Chief Scientist, DFKI Berlin

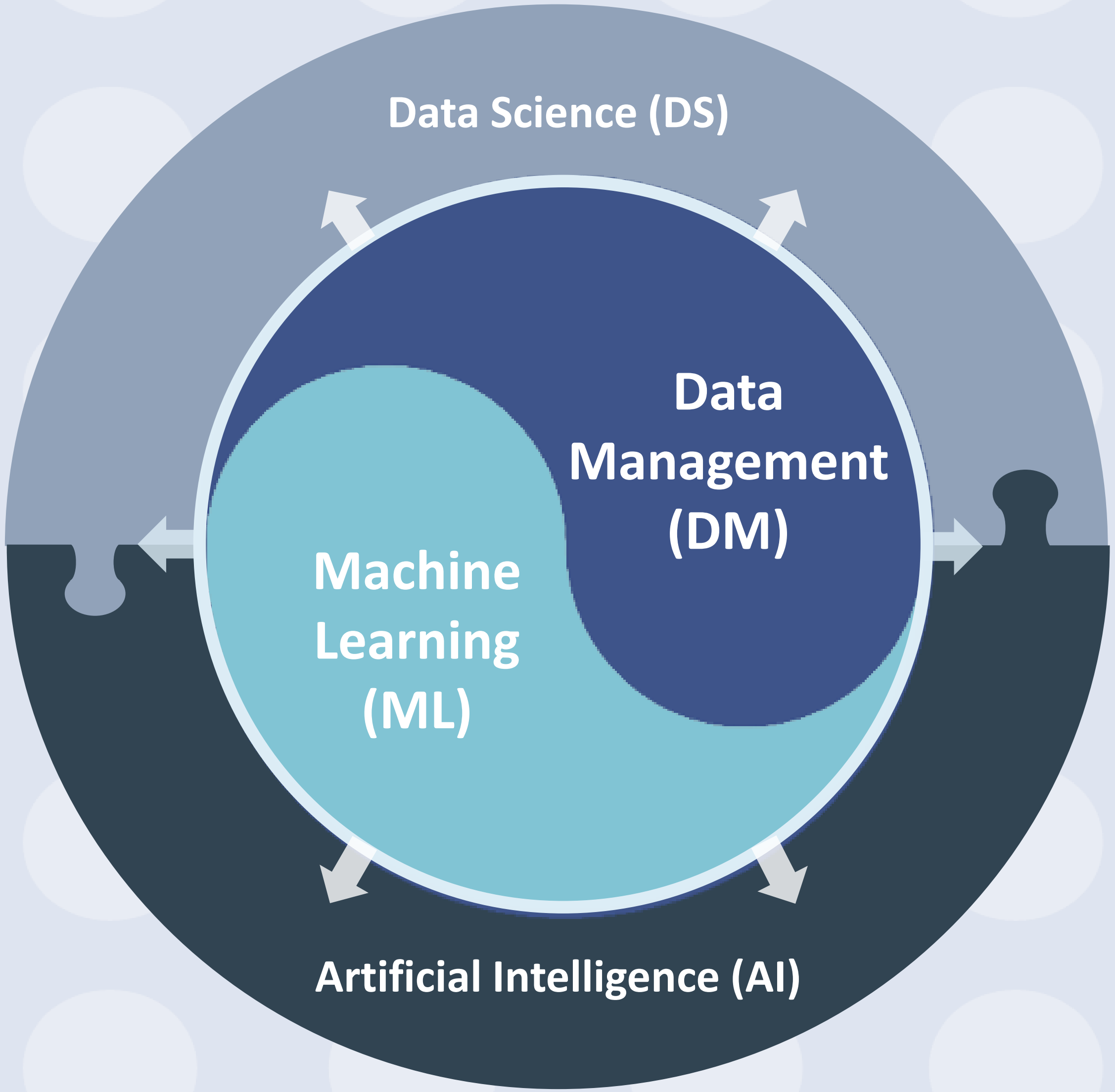
Director, Berlin Institute for the Foundations of Learning and Data (BIFOLD)

❶ Data as a Factor of Production

❷ Selected Research Contributions

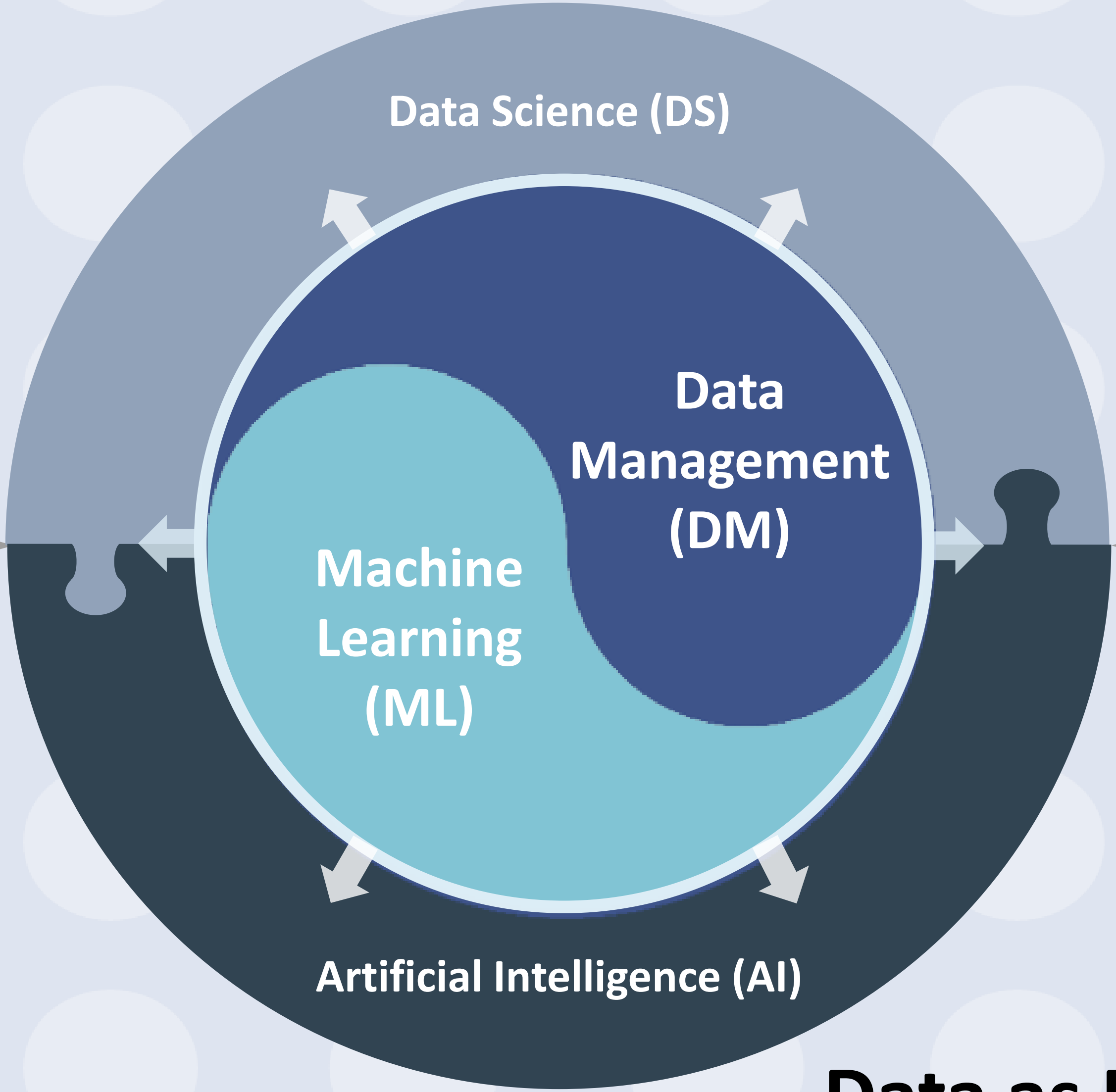
❸ Summary and Vision

Big Data and Machine Learning are the key drivers of innovation in AI and Data Science.

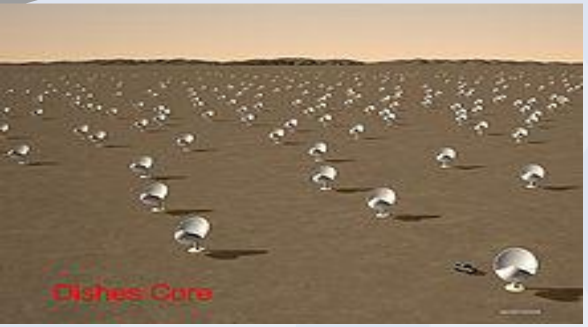


Data Management jointly with ML are disruptive in the Sciences, Humanities, and Industry.

Sciences and Humanities



Industry



4th Paradigm

Data as Factor of Production

The Fourth Paradigm – A New Standard for Research

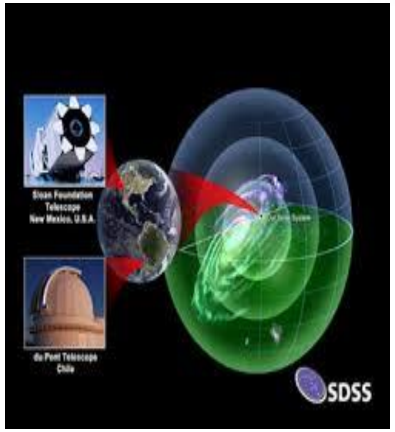
❶ 1000 Years Ago: **Empirical**
- ✓ Description of Natural Phenomena

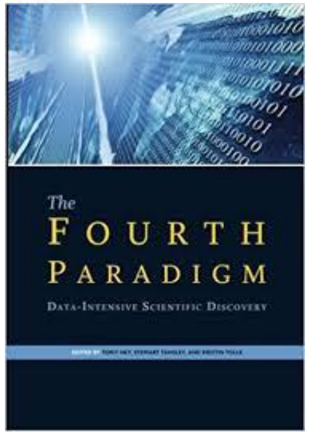
❷ The Last 100 Years: **Theoretical**
- ✓ Modeling and Generalizations

❸ The Last Decades: **Computational**
- ✓ Simulation of Complex Phenomena

❹ Nowadays: **Data Intensive**
- ✓ Massive Data Amounts Generated by Measurements and Simulation
- ✓ Data Exploration Through Software
- ✓ Information and Knowledge Stored On Computers
- ✓ Scientists Employ Databases/Files, Perform Data Management, Conduct Statistical Analysis

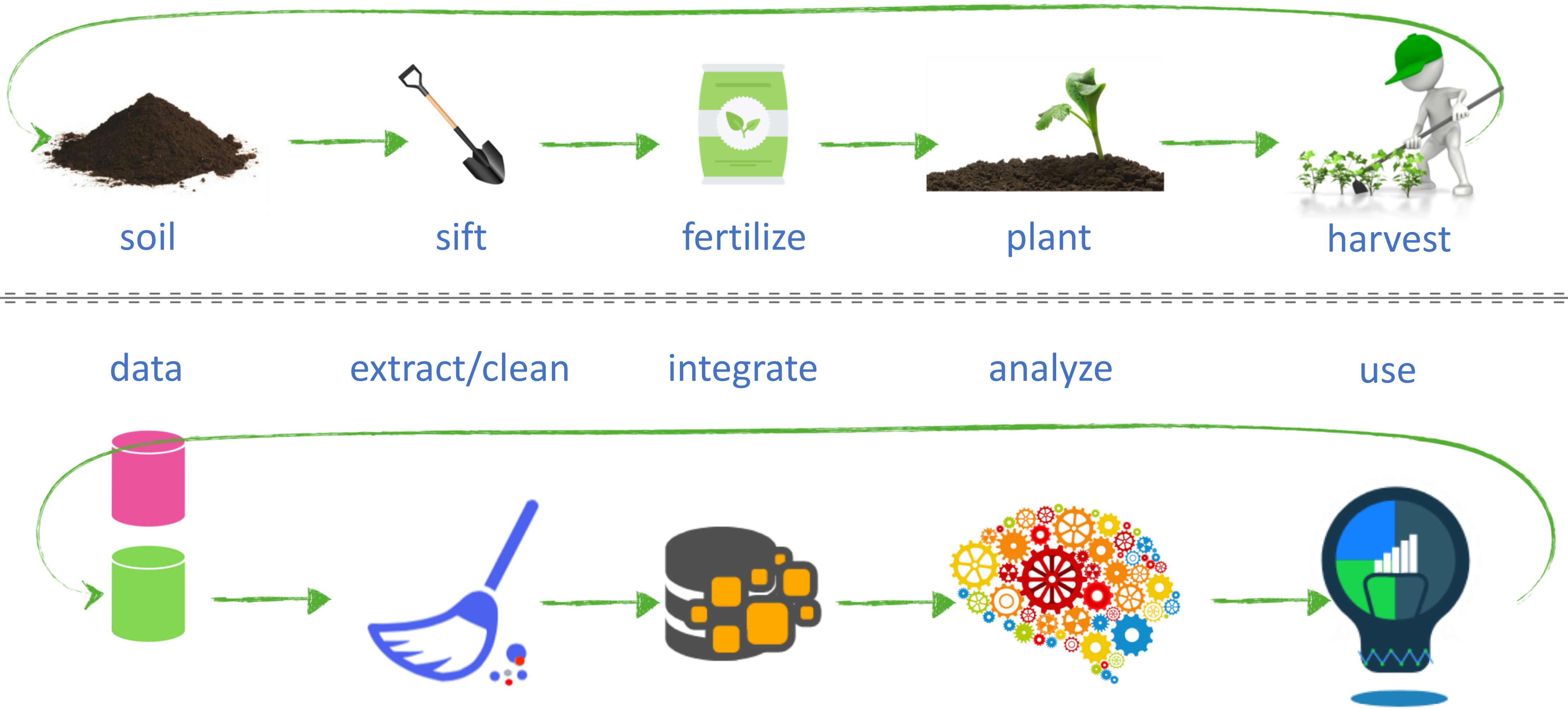
$$\nabla \cdot \mathbf{D} = \rho$$
$$\nabla \cdot \mathbf{B} = 0$$
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$
$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$$





https://www.microsoft.com/en-us/research/publication/fourth-paradigm-data-intensive-scientific-discovery
https://blogs.technet.microsoft.com/dataplatforminsider/2016/03/10/mapping-the-universe-with-sql-server/

Data – A New Factor of Production



Remote Sensing

Nominated ESA–EGU EOS Excellence Award

Analysis of massive satellite data (Sentinel-2)

Labeled training data archive for AI models

TBs of multimodal data / day

Fast classification and categorization

© scihub.copernicus.eu

Enables:

- analysis of trends (e.g., deforestation)
- predictions about regions (e.g., drought)

Added to popular “big data” catalogs:

(e.g., Google Earth, Radiant MLHub, TensorFlow)

[1] BigEarthNet: A Large-Scale Benchmark Archive. IGARSS 2019.
[2] Multi-Label Remote Sensing. IEEE Access 2020.
[3] http://bigearth.net

(Begüm Demir, TU Berlin)

Industrie/Industry 4.0

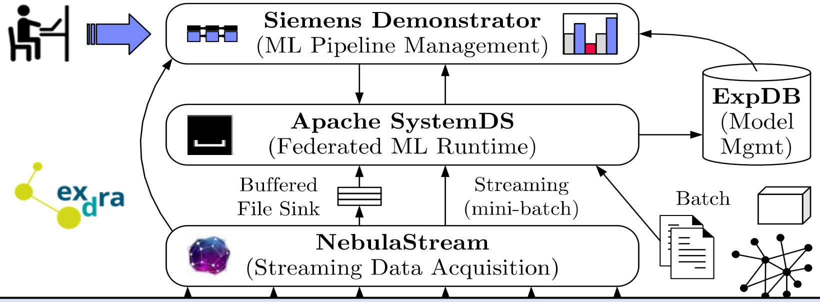
Exploratory real-time analysis of sensor data streams

Prediction of paper quality during production

97 heterogeneous sensors

low latency

complex model building + information extraction and integration



Enables:

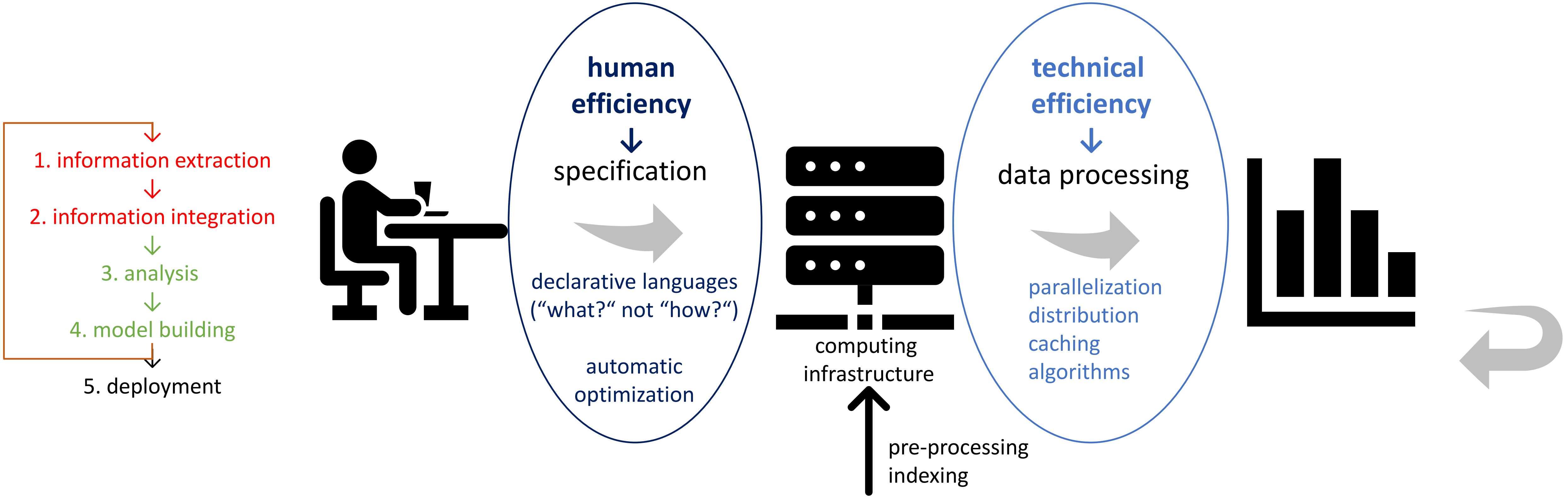
- faster reaction to paper quality issues
- cost reduction

Data science on real-time data streams

[1] The NebulaStream Platform. CIDR 2020.
[2] SystemDS. CIDR 2020.

(TU Graz, Siemens)

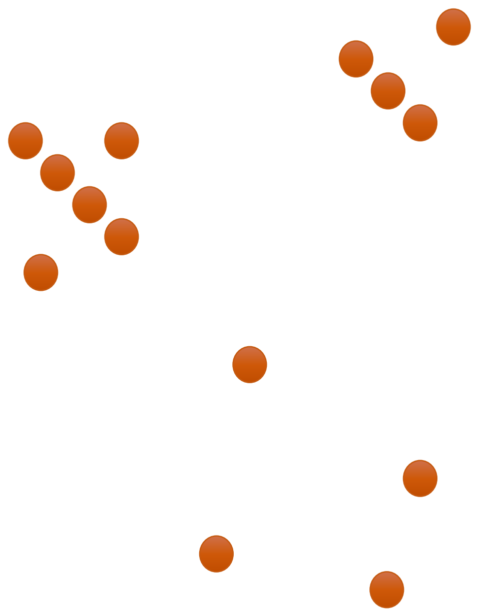
The data science process is complex and time-consuming



Goal of data management research:
scaling the process and its operationalization with respect to human and technical efficiency

Draws on results from:
computer architecture, statistics, machine learning, distributed systems, compilers, programming languages, etc.

Example: "3-Means Clustering"



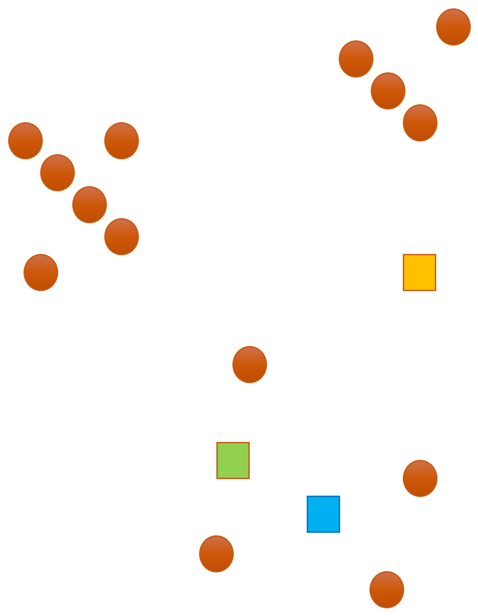
Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers

"3-Means Clustering" is a simple data analysis method that divides a dataset into *k* groups (clusters) with respect to their relative distance.
The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Example: "3-Means Clustering"



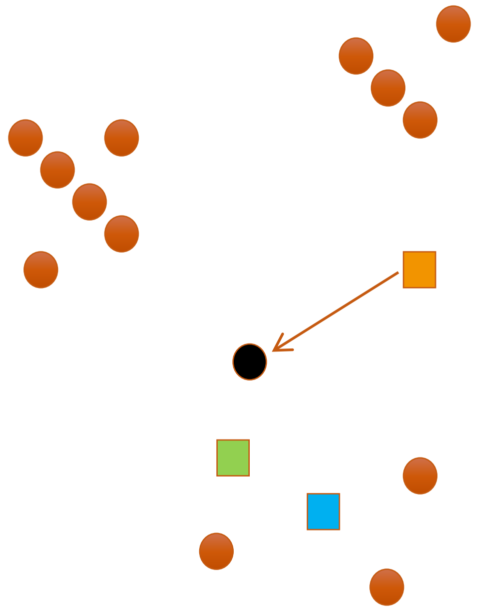
Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers

"3-Means Clustering" is a simple data analysis method that divides a dataset into *k* groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Example: "3-Means Clustering"



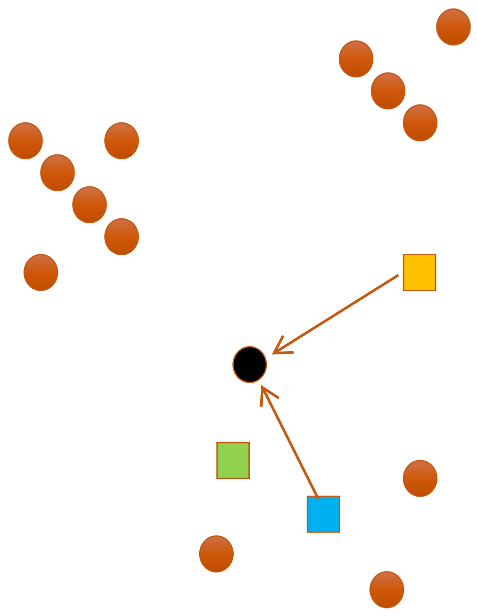
Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers

"3-Means Clustering" is a simple data analysis method that divides a dataset into *k* groups (clusters) with respect to their relative distance.
The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Example: "3-Means Clustering"



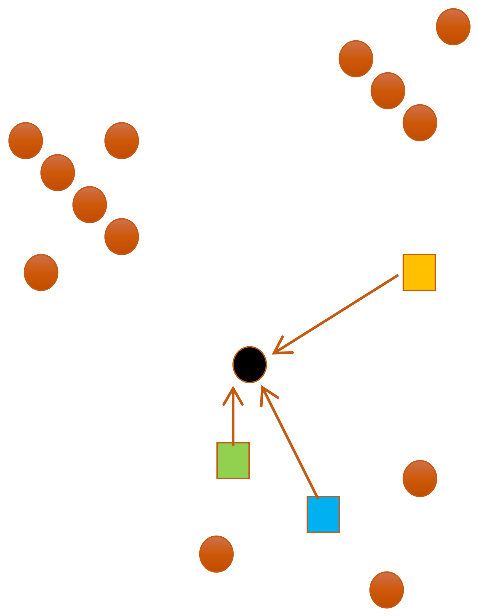
Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers

"3-Means Clustering" is a simple data analysis method that divides a dataset into *k* groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Example: "3-Means Clustering"



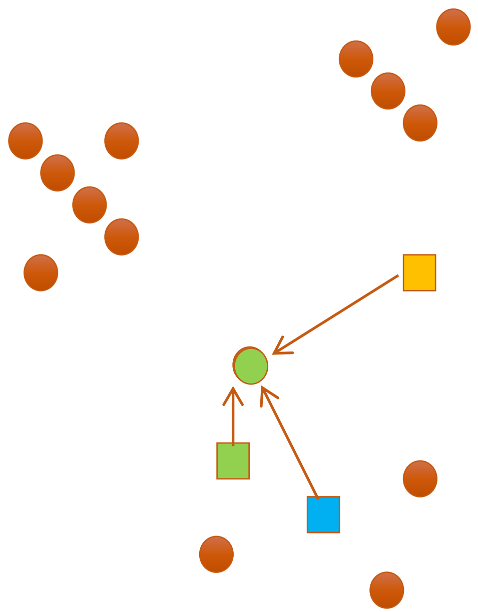
Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers

"3-Means Clustering" is a simple data analysis method that divides a dataset into *k* groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Example: "3-Means Clustering"



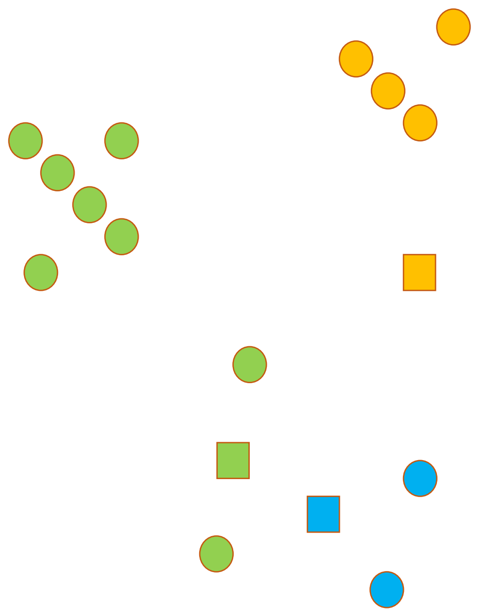
Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers

"3-Means Clustering" is a simple data analysis method that divides a dataset into *k* groups (clusters) with respect to their relative distance.
The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Example: "3-Means Clustering"



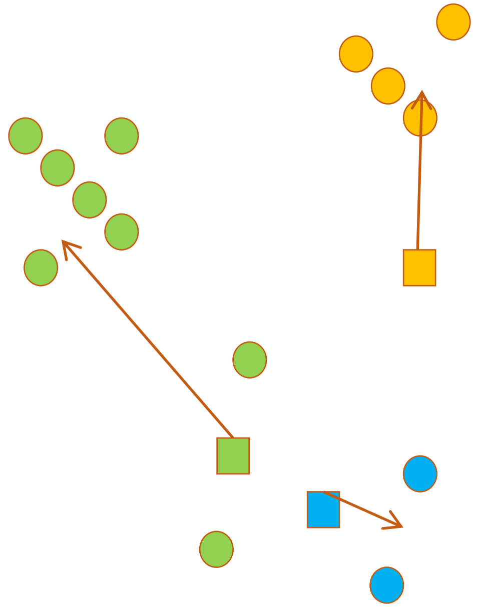
Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers

"3-Means Clustering" is a simple data analysis method that divides a dataset into *k* groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Example: "3-Means Clustering"



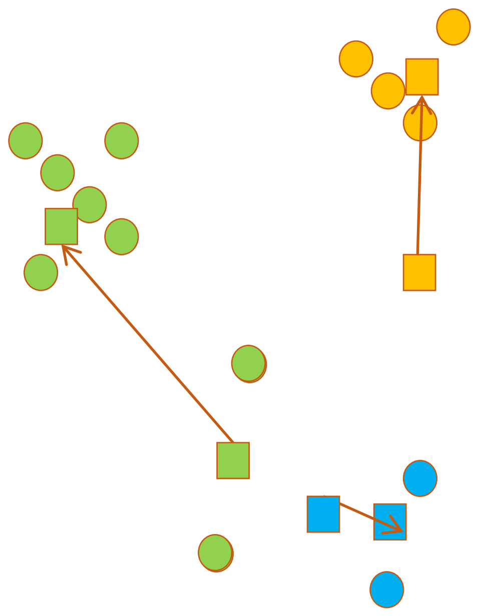
Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers

"3-Means Clustering" is a simple data analysis method that divides a dataset into *k* groups (clusters) with respect to their relative distance.
The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Example: "3-Means Clustering"



Choose 3 random cluster centers

Iterate until convergence:

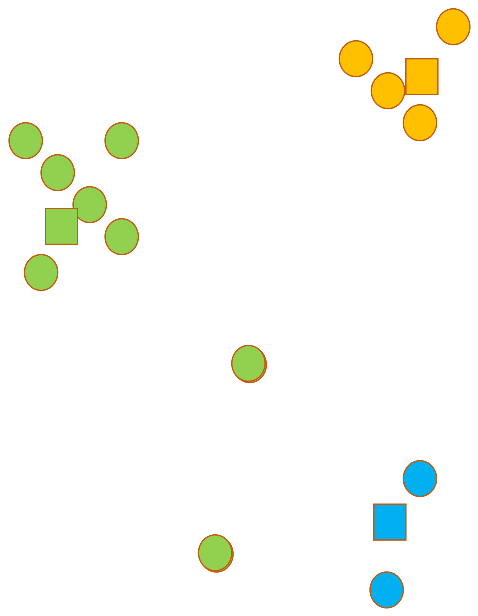
Compute distance of each point to each center

Assign each point to the closest cluster

Move centers

"3-Means Clustering" is a simple data analysis method that divides a dataset into *k* groups (clusters) with respect to their relative distance. The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Example: "3-Means Clustering"



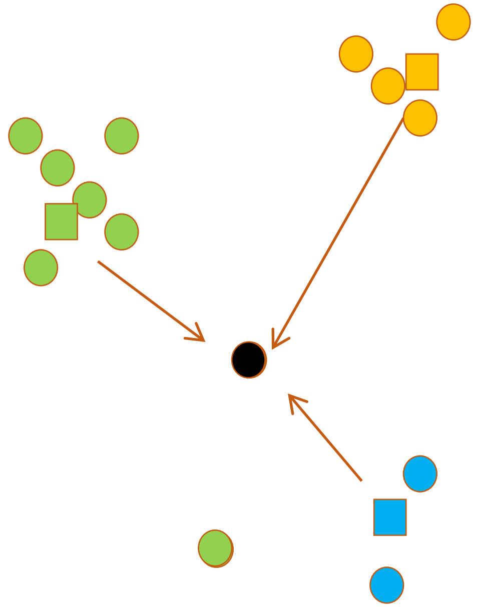
Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers

"3-Means Clustering" is a simple data analysis method that divides a dataset into *k* groups (clusters) with respect to their relative distance.
The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Example: "3-Means Clustering"



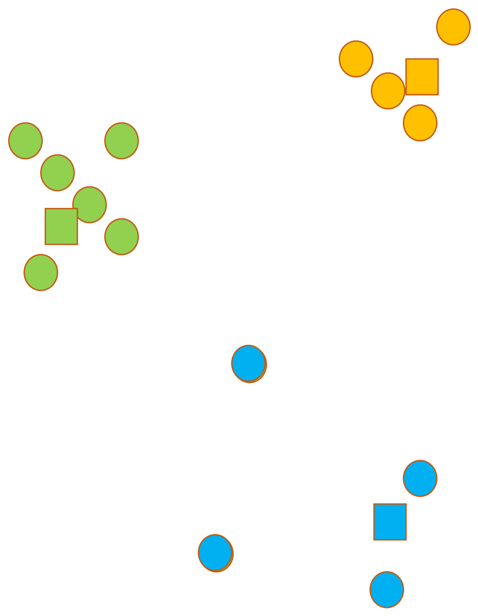
Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers

"3-Means Clustering" is a simple data analysis method that divides a dataset into *k* groups (clusters) with respect to their relative distance.
The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Example: "3-Means Clustering"



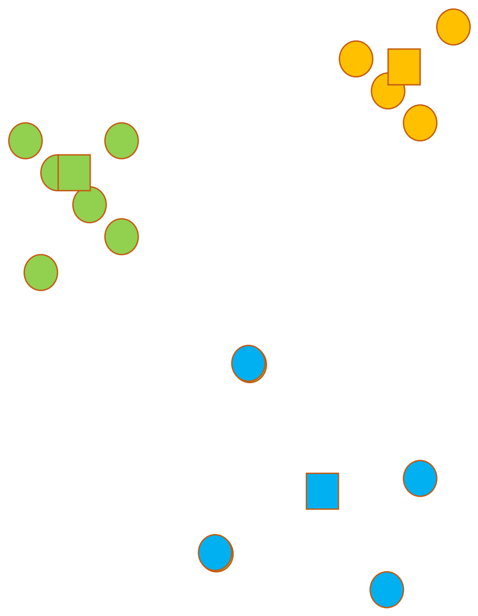
Choose 3 random cluster centers

Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers

"3-Means Clustering" is a simple data analysis method that divides a dataset into *k* groups (clusters) with respect to their relative distance.
The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Example: "3-Means Clustering"



Choose 3 random cluster centers

Iterate until convergence:

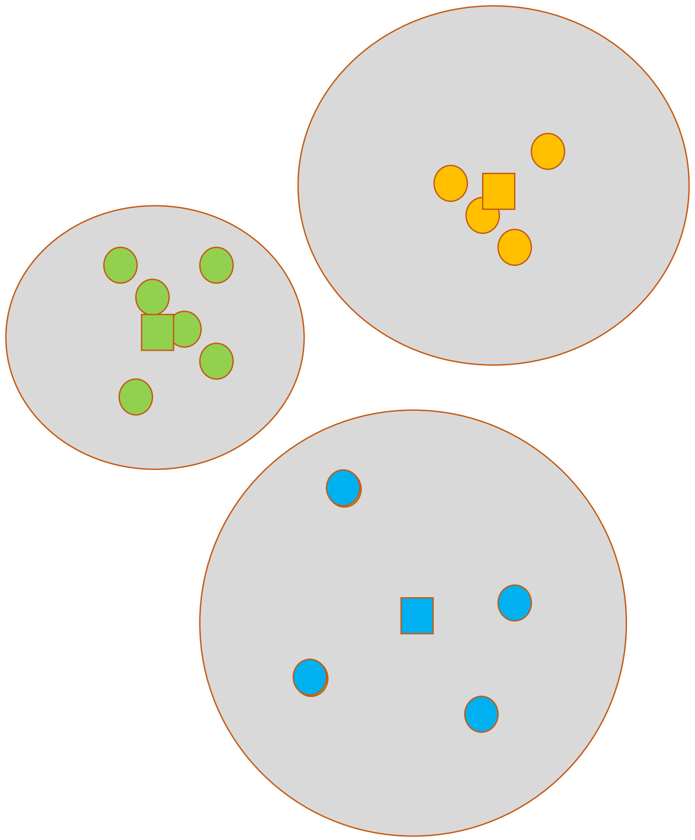
Compute distance of each point to each center

Assign each point to the closest cluster

Move centers

"3-Means Clustering" is a simple data analysis method that divides a dataset into *k* groups (clusters) with respect to their relative distance.
The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

Example: "3-Means Clustering"



Choose 3 random cluster centers

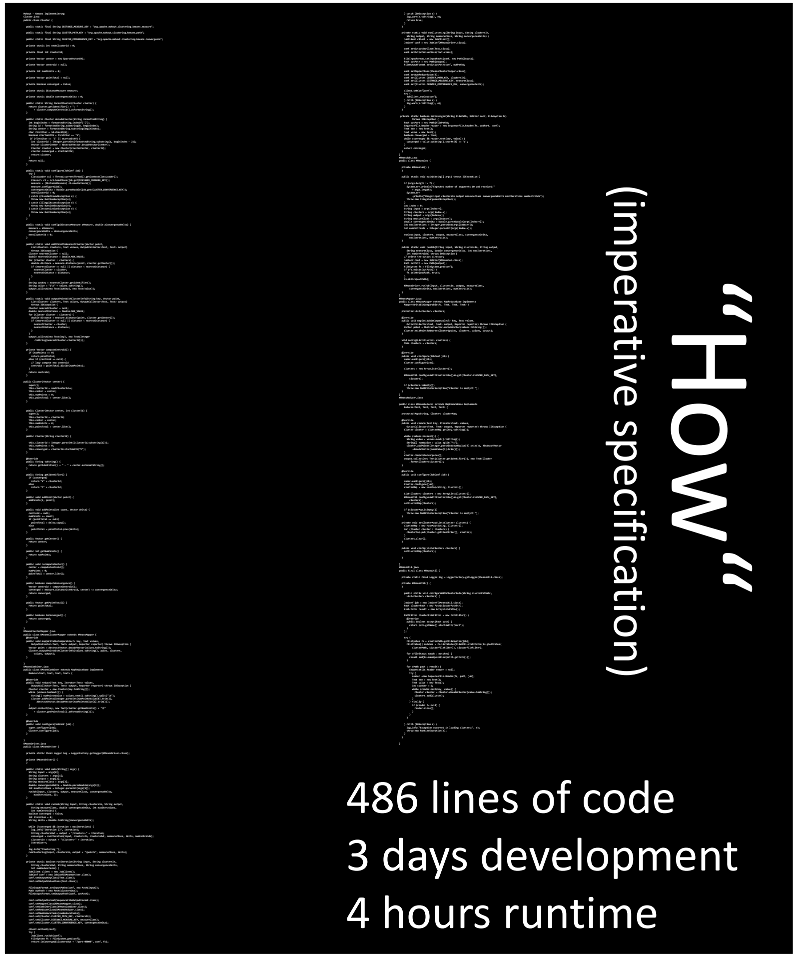
Iterate until convergence:

- Compute distance of each point to each center
- Assign each point to the closest cluster
- Move centers

"3-Means Clustering" is a simple data analysis method that divides a dataset into *k* groups (clusters) with respect to their relative distance.
The example illustrates an iterative algorithm to determine three groups (clusters) for a set of points according to a Euclidean distance.

“What“ not “How“

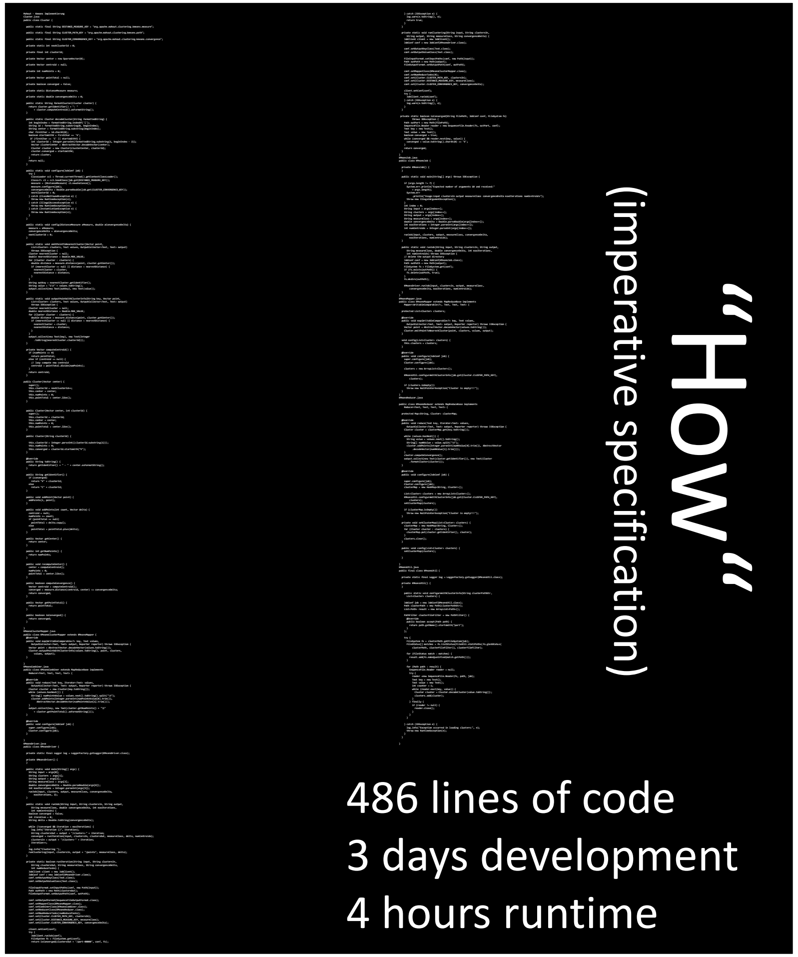
Example: “k-Means Clustering“



Hand-optimized code
(data-, workload- and system-dependent)

“What“ not “How“:
Example: “k-Means Clustering“

Declarative data flow program with automatic optimization, parallelization, and hardware adaption



Hand-optimized code
(data-, workload- and system-dependent)

❶ Data as a Factor of Production

❷ Selected Research Contribution: Reduction of Human and Technical Latency

❸ Summary and Vision

Apache Flink: Data Programmability and Scalable Data Stream Analytics

human efficiency

```
case class Path (from: Long, to:
Long)
val tc = edges.iterate(10) {
  paths: DataSet[Path] =>
    val next = paths
      .join(edges)
      .where("to")
      .equalTo("from") {
        (path, edge) =>
          Path(path.from, edge.to)
      }
      .union(paths)
      .distinct()
    next
  }
```

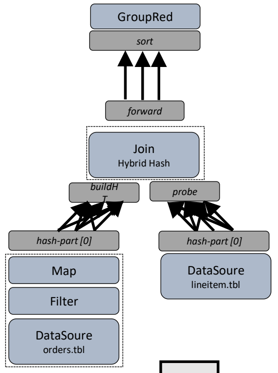
Declarative data-analysis program

algebraic model

cost model

Automatic optimization, parallelization and hardware adaption

data flow graph



specification time

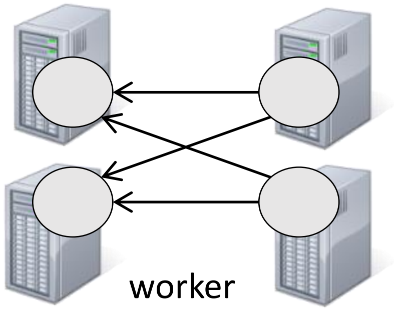
runtime

memory management

out-of-core algorithms

data stream processing

state management



worker

Distribution of operators

Monitoring of operation

fault-tolerance

resource management

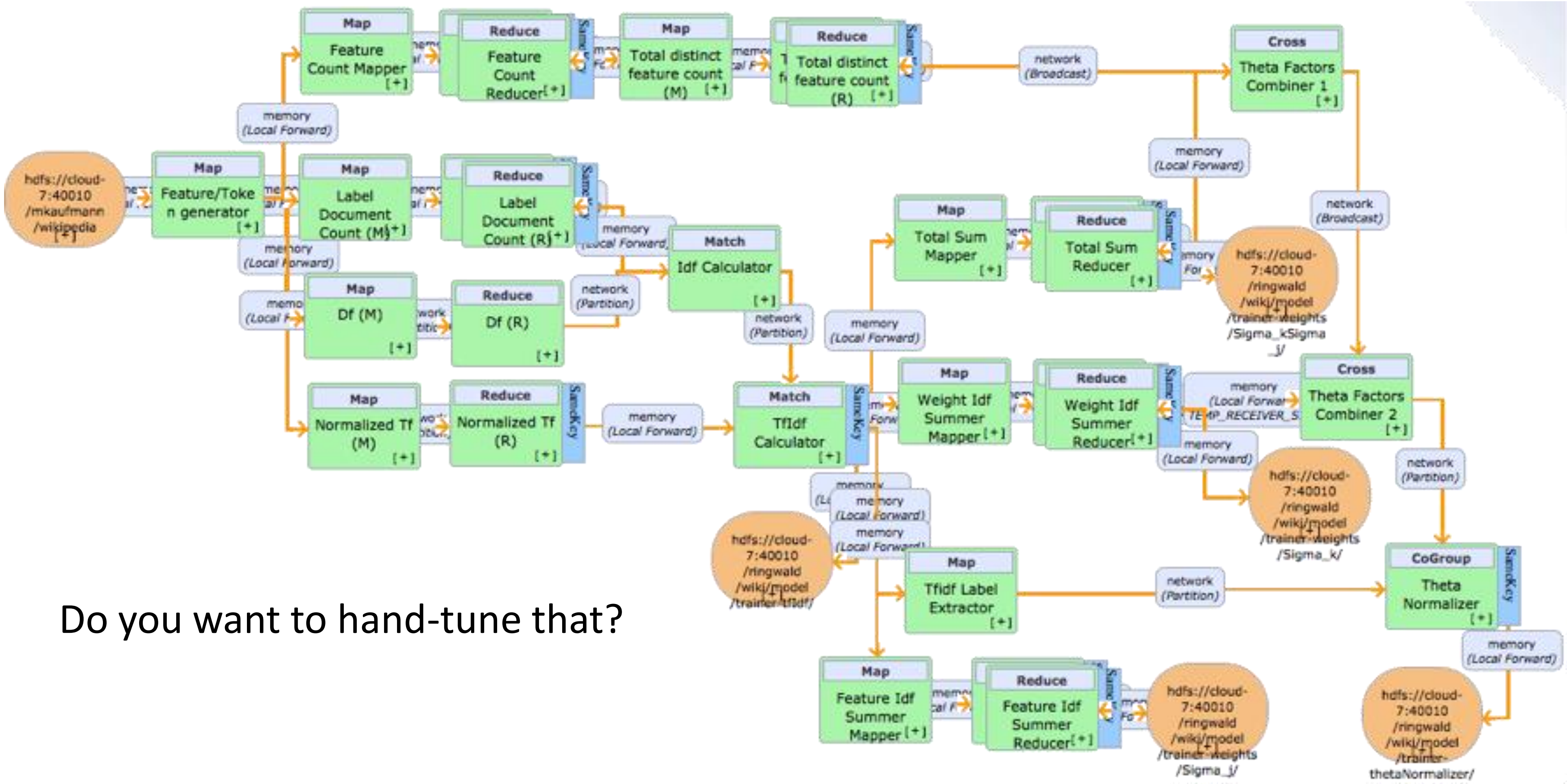
coordinator

technical efficiency

D. Battré, S. Ewen, F. Hueske, O. Kao, V. Markl, D. Warneke: *Nephele/PACTs: a programming model and execution framework for web-scale analytical processing. SoCC 2010: 119-130*

P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, K. Tzoumas: *Apache Flink™: Stream and Batch Processing in a Single Engine. IEEE Data Eng. Bull. 38(4): 28-38 (2015)*

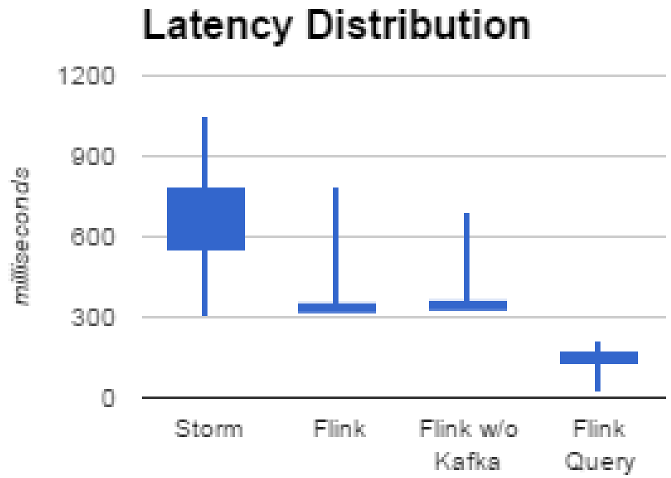
Why Optimization?



Do you want to hand-tune that?

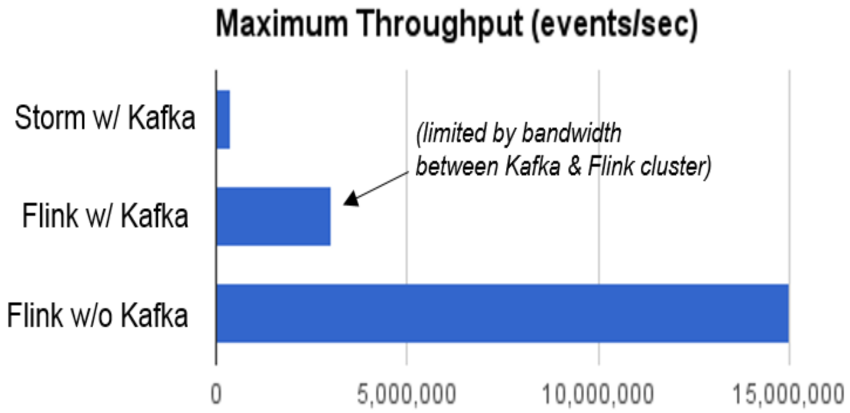
F. Hueske, M. Peters, A. Krettek, M. Ringwald, K. Tzoumas, V. Markl, J.C. Freytag: Peeking into the optimization of data flow programs with MapReduce-style UDFs. ICDE 2013: 1292-1295

Effect of Optimization:
Lower latency and higher throughput in particular for streaming applications



2x to 4x lower latency

50x higher throughput



[1] https://www.ververica.com/blog/extending-the-yahoo-streaming-benchmark

[1] https://www.ververica.com/blog/extending-the-yahoo-streaming-benchmark

compression of data streams for visualization

implicit parallelism through deep language embedding

latency reduction by result sharing

exploiting fast interconnects for heterogeneous processing

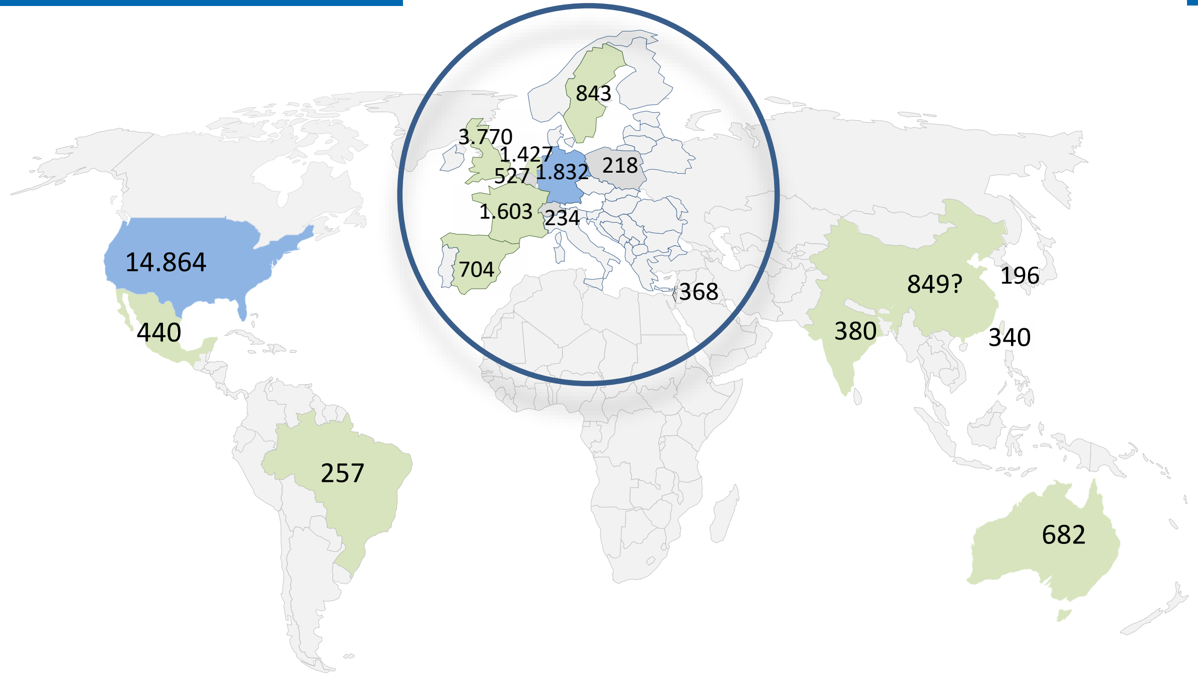
processing control flow in data flow systems

fast data loading using RDMA and GPUs

Apache Flink

https://www.meetup.com/topics/apache-flink/
https://flink.apache.org/poweredby.html
https://github.com/apache/flink

Flink Community



Flink Contributors

29,500+	Meetup Members Worldwide	18	Countries that Regularly Hold Meetups
870+	Open Source Contributors/Developers	49+	Companies using Apache Flink
49	Meetup Groups Worldwide		Startup data Artisans (now Ververica)

Last updated: May 2021

Some Highly Engaged Users

Largest job has > 20 operators, runs on > 5000 vCores in 1000-node cluster, processes millions of events per second

Complex jobs of > 30 operators running 24/7, processing 30 billion events daily, maintaining state of 100s of GB with exactly-once guarantees

30 Flink applications in production for more than one year; 10 billion events (2TB) processed daily

Courtesy of Kostas Tzoumas

Apache Flink Users

https://cwiki.apache.org/confluence/display/FLINK/Powered+by+Flink

❶ Data as a Factor of Production

❷ Selected Research Contribution

❸ Summary and Vision

Evolution of Big Data Platforms

FIRST GENERATION

Data Warehouses,
e.g. , **relational DBMS**

SECOND GENERATION

Scale-out, Map/Reduce, UDFs,
e.g. , **Apache Hadoop**

THIRD GENERATION

In-memory Performance and
Improved Programming Model,
e.g. **Apache Spark**

FOURTH GENERATION

In-memory + Out of Core Performance,
Declarativity, Optimization of Iterative
Algorithms, True Streaming
e.g. , **Apache Flink**

FIFTH GENERATION

Heterogeneous, Distributed, and Federated Architecture;
Management of Data and Algorithms;
Infrastructure for Publishing, Sharing, and Billing,
e.g., **NebulaStream, SystemDS, Apache Wayang, BigEarthNet**

Post-Doc and PhD Student Opportunities

Berlin, the (digital) capital of Germany, is a **young, cosmopolitan, international city** in the heart of Europe, with a very large research and science industry as well as a dynamic and **thriving startup scene**, in particular in the creative and information technology space.

Pursue a DATA MANAGEMENT, DATA SCIENCE, AND DATA ENGINEERING career within

- ✓ Doctoral and postdoctoral positions
- ✓ Topics include Large Scale Streaming Infrastructures for IoT and Fog, Data Infrastructures, Responsible Data Management

Questions and application submissions (including cover letter, CV, transcripts, and copies of your academic degrees) should be sent to: jobs@dima.tu-berlin.de.

Reference Pages

The DIMA Research Group, https://tu.berlin/en/dima/

BIFOLD, https://bifold.berlin

Prof. Volker Markl, https://tu.berlin/en/dima/about-us/prof-dr-volker-markl

Conclusion

❶ Data are a new **factor of production** for sciences, humanities, and industry.
- Critical success factor is the increase in **human efficiency** through intuitively usable systems.
- Improving the **technical efficiency** enables real-time analysis.

❷ Data management research is **interdisciplinary** within and outside computer science.

❸ Data management research has created important **technological foundations, new systems,** and **novel applications**:
- **Methods** for improving efficiency via automatic optimization, parallelization, and hardware adaption.
- **Systems** for efficient-, distributed-, and compliant analysis of large data sets and streams.
- **Applications** in the **Sciences and Humanities**, **Industry** and **Society** (“Citizen Science“).

❹ **Current research challenges** in data management include:
- Methods and systems for the management and analysis of **massive distributed, heterogeneous data streams** (“IoT“, “Industrie 4.0“).
- Technologies for **data management infrastructures for open and protected, collaborative AI innovations**.

❺ The scientific community in Berlin is tackling these challenges in a comprehensive **ecosystem**:
- **Foundational research** (BIFOLD, TU Berlin).
- **Applications** in mathematics/sciences, healthcare/biotech, humanities, as well as industry and startups.