BIG DATA INTEGRATION RESEARCH AT THE UNIVERSITY OF LEIPZIG

ERHARD RAHM, UNIV. LEIPZIG

ScaDS DRESDEN LEIPZIG

UNIVERSITY OF LEIPZIG

- Founded in 1409
- Now about 30.000 students in 14 faculties
- Computer science
  - 13 professorships and 2 junior professors
  - 150 PhD students and postdocs (120 by third party funding)

ScaDS DRESDEN LEIPZIG

GERMAN CENTERS FOR BIG DATA

Two Centers of Excellence for Big Data in Germany

- ScaDS Dresden/Leipzig
- Berlin Big Data Center (BBDC)

ScaDS Dresden/Leipzig (Competence Center for Scalable Data Services and Solutions Dresden/Leipzig)

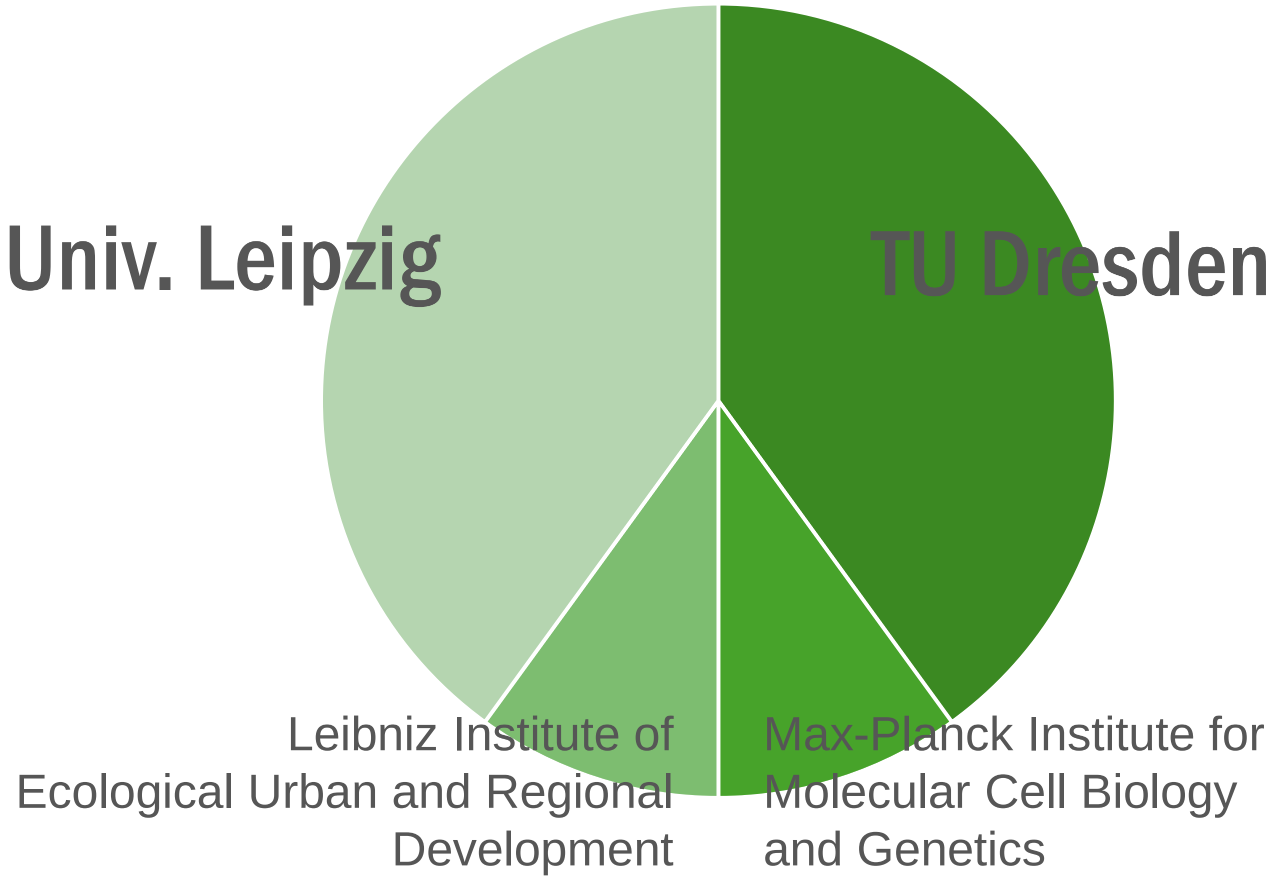
- scientific coordinators: Nagel (TUD), Rahm (UL)
- start: Oct. 2014
- duration: 4 years (option for 3 more years)
- initial funding: ca. 5.6 Mio. Euro

ScaDS DRESDEN LEIPZIG

GOALS

- Bundling and advancement of existing expertise on Big Data
- Development of Big Data Services and Solutions
- Big Data Innovations

FUNDED INSTITUTES



ScaDS DRESDEN LEIPZIG

ASSOCIATED PARTNERS

- Avantgarde-Labs GmbH
- Data Virtuality GmbH
- E-Commerce Genossenschaft e. G.
- European Centre for Emerging Materials and Processes Dresden
- Fraunhofer-Institut für Verkehrs- und Infrastruktursysteme
- Fraunhofer-Institut für Werkstoff- und Strahltechnik
- GISA GmbH
- Helmholtz-Zentrum Dresden - Rossendorf
- Hochschule für Telekommunikation Leipzig
- Institut für Angewandte Informatik e. V.
- Landesamt für Umwelt, Landwirtschaft und Geologie
- Netzwerk Logistik Leipzig-Halle e. V.
- Sächsische Landesbibliothek – Staats- und Universitätsbibliothek Dresden
- Scionics Computer Innovation GmbH
- Technische Universität Chemnitz
- Universitätsklinikum Carl Gustav Carus

ScaDS DRESDEN LEIPZIG

STRUCTURE OF THE CENTER

- Life sciences
- Material and Engineering sciences
- Environmental / Geo sciences
- Digital Humanities
- Business Data

Service center

Big Data Life Cycle Management and Workflows

Data Quality / Data Integration	Knowledge Extraktion	Visual Analytics

Efficient Big Data Architectures

ScaDS DRESDEN LEIPZIG

RESEARCH PARTNERS

- Data-intensive computing W.E. Nagel
- Data quality / Data integration E. Rahm
- Databases W. Lehner, E. Rahm
- Knowledge extraction/Data mining
  C. Rother, P. Stadler, G. Heyer
- Visualization
  S. Gumhold, G. Scheuermann
- Service Engineering, Infrastructure
  K.-P. Fähnrich, W.E. Nagel, M. Bogdan

APPLICATION COORDINATORS

- Life sciences G. Myers
- Material / Engineering sciences M. Gude

- Environmental / Geo sciences J. Schanze
- Digital Humanities G. Heyer

- Business Data B. Franczyk

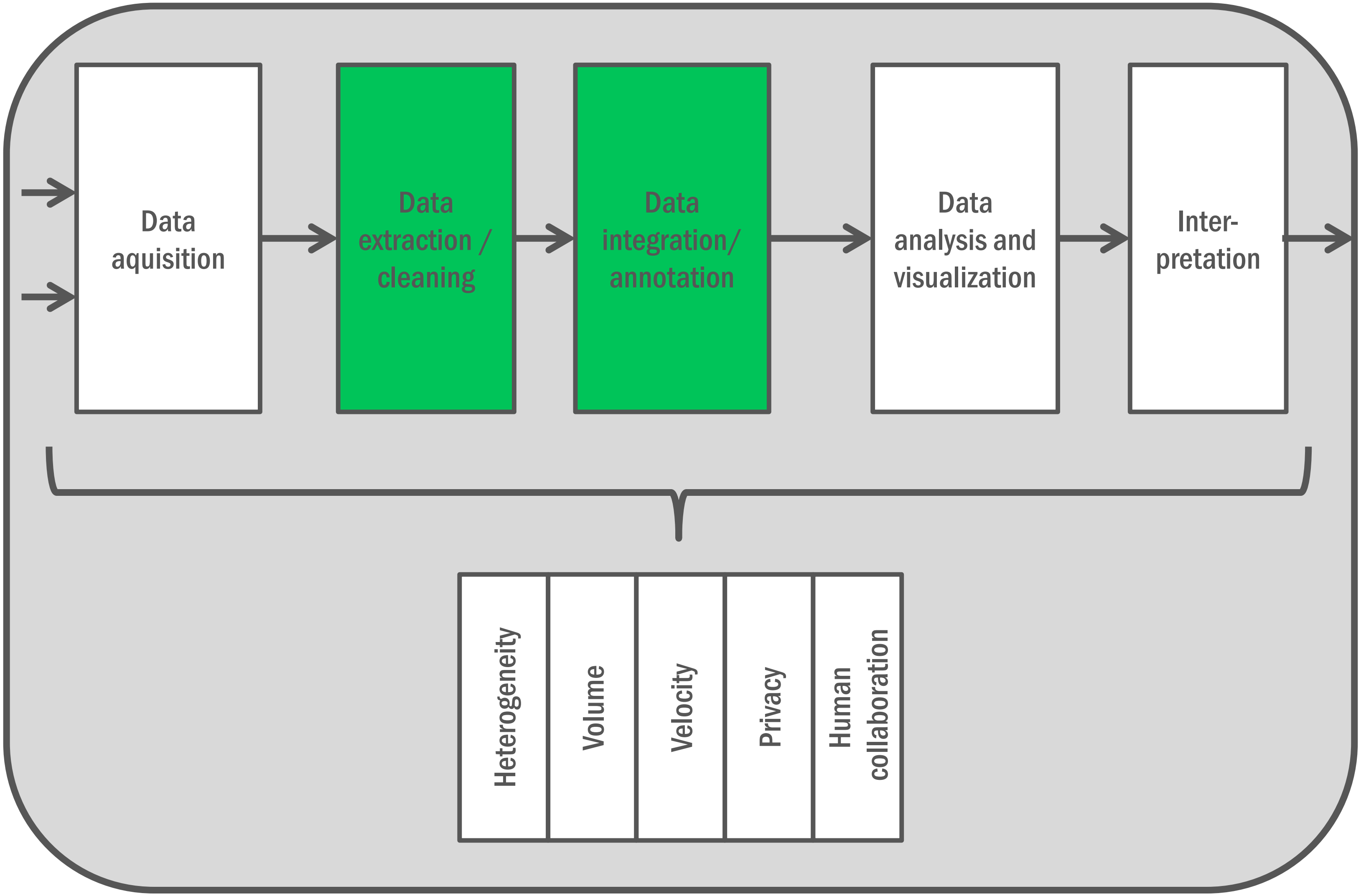
ScaDS DRESDEN LEIPZIG

AGENDA

- ScaDS Dresden/Leipzig
- Big Data Integration
  - Introduction
  - Matching product offers from web shops
  - DeDoop: Deduplication with Hadoop
- Privacy-preserving record linkage with PP-Join
  - Cryptographic bloom filters
  - Privacy-Preserving PP-Join (P4Join)
  - GPU-based implementation
- Big Graph Data
  - Graph-based Business Intelligence with BIIIG
  - GraDoop: Hadoop-based data management and analysis
- Summary and outlook

ScaDS DRESDEN LEIPZIG

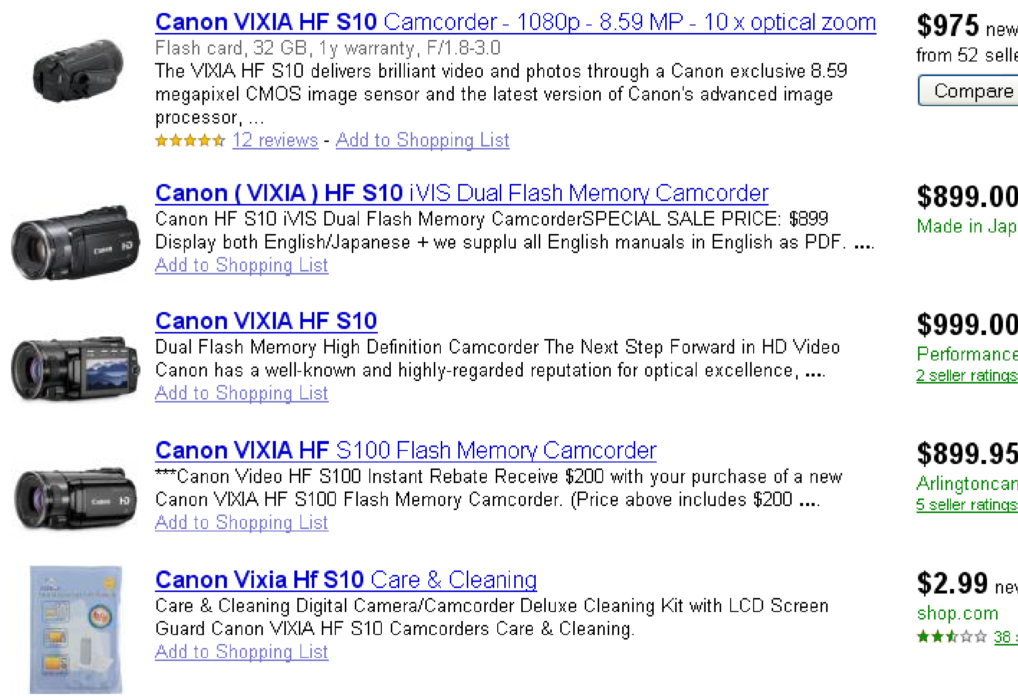
BIG DATA ANALYSIS PIPELINE



ScaDS DRESDEN LEIPZIG

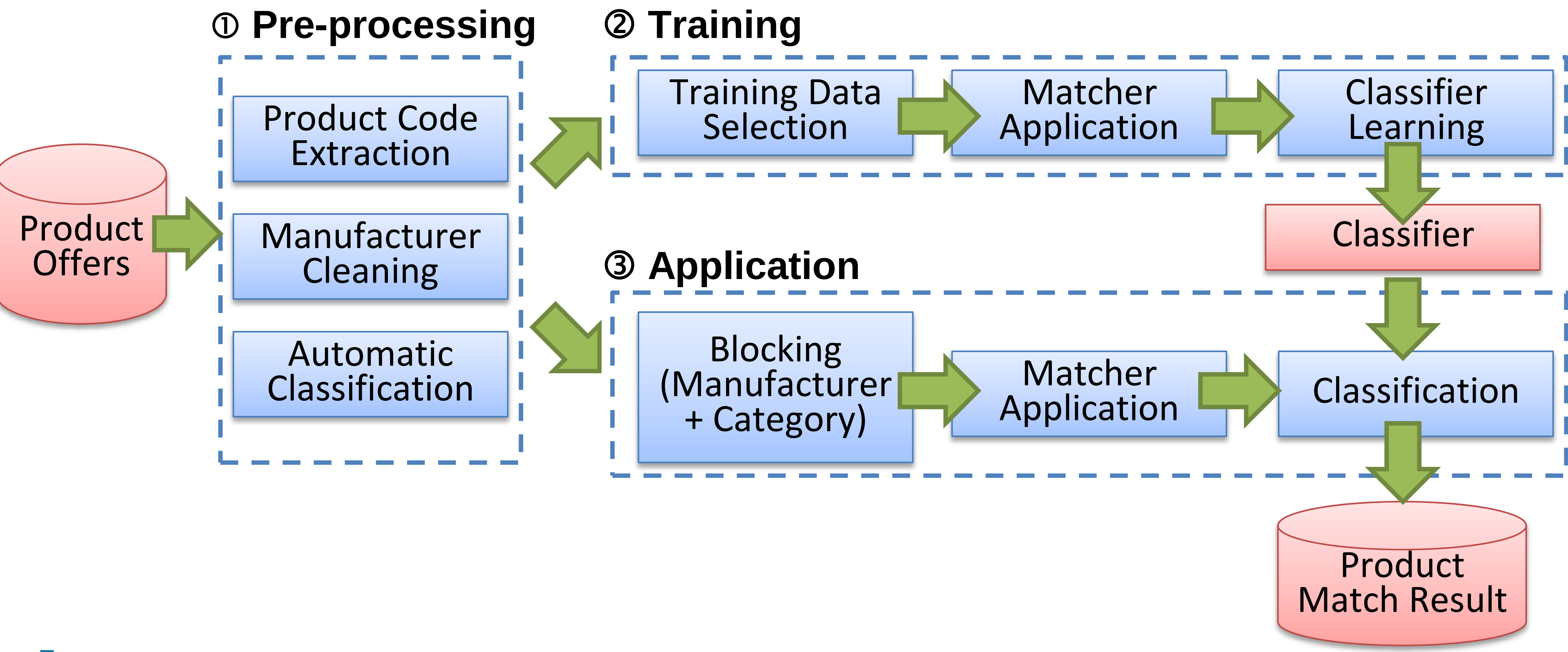
BIG DATA INTEGRATION USE CASE
INTEGRATION OF PRODUCT OFFERS IN COMPARISON PORTAL

- Thousands of data sources (shops/merchants)
- Millions of products and product offers
- Continous changes
- Many similar, but different products
- Low data quality



ScaDS DRESDEN LEIPZIG

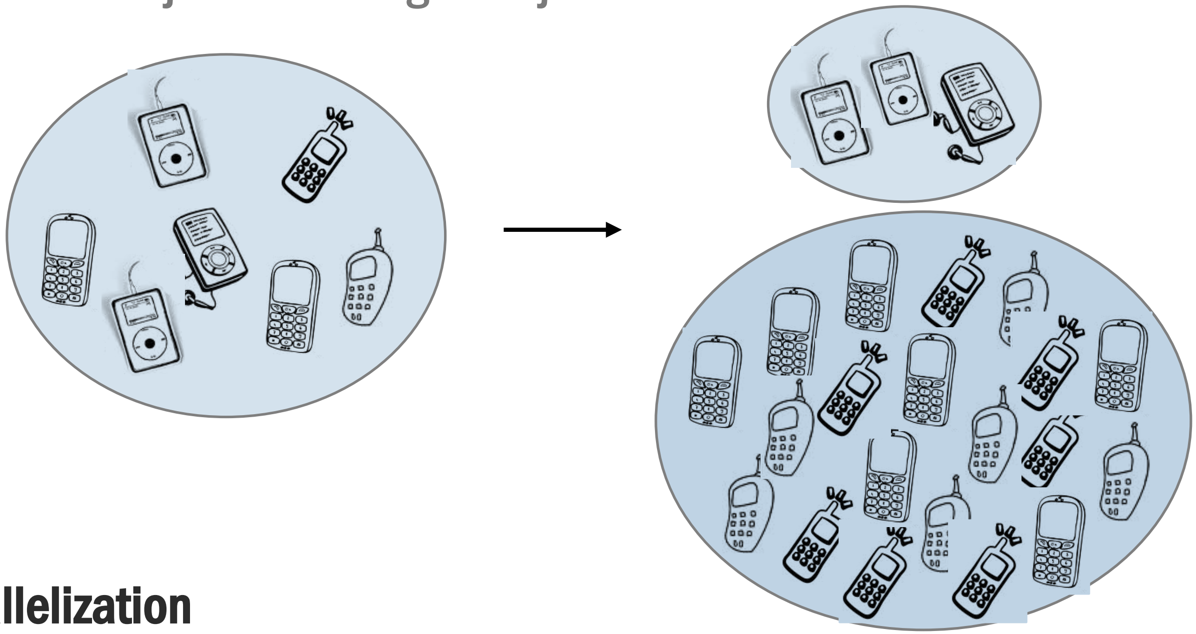
LEARNING-BASED MATCH APPROACH



ScaDS DRESDEN LEIPZIG

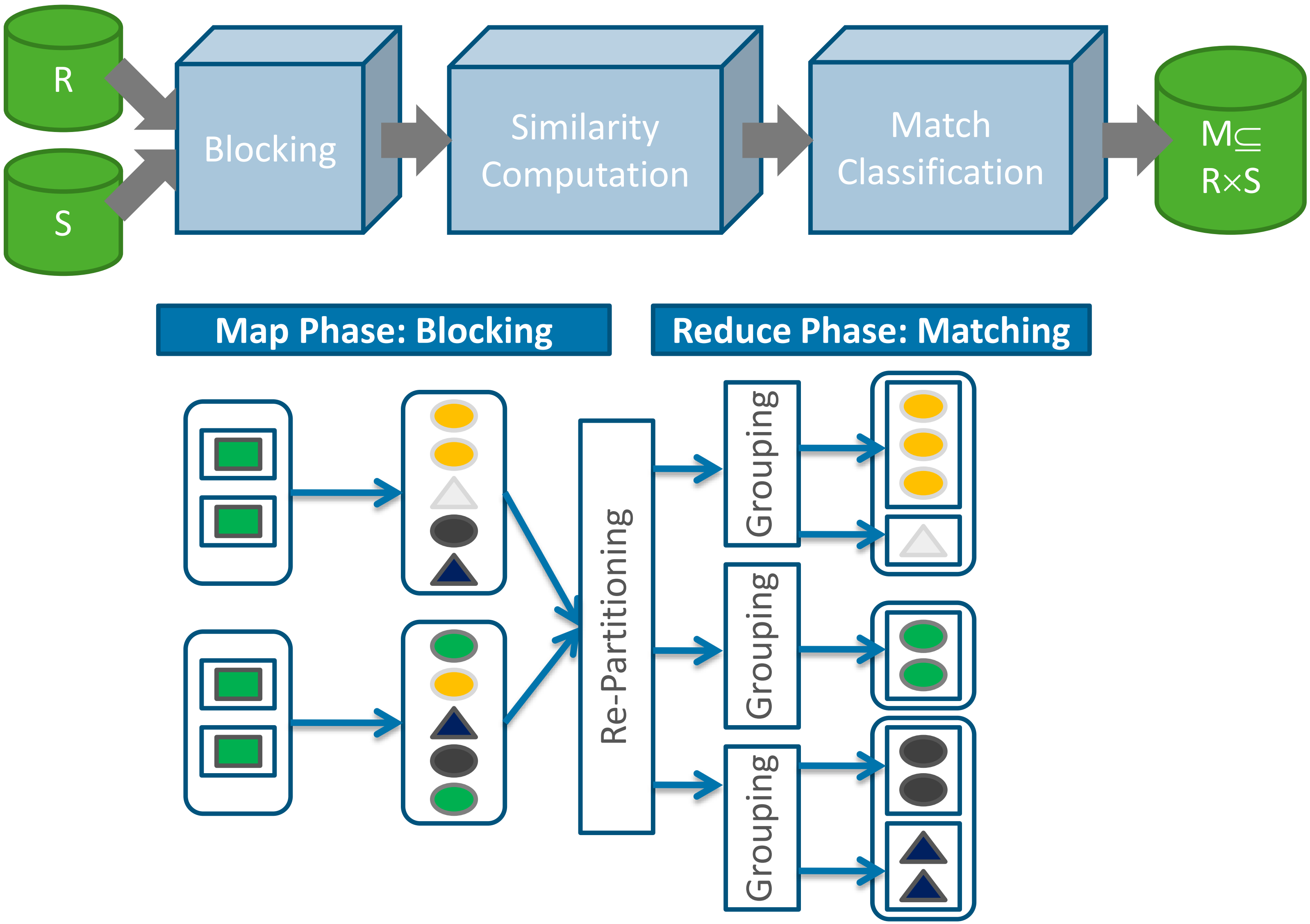
HOW TO SPEED UP OBJECT MATCHING?

- **Blocking** to reduce search space
  - group similar objects within blocks based on *blocking key*
  - restrict object matching to objects from the same block



- **Parallelization**
  - split match computation in sub-tasks to be executed in parallel
  - exploitation of Big Data infrastructures such as Hadoop (Map/Reduce or variations)

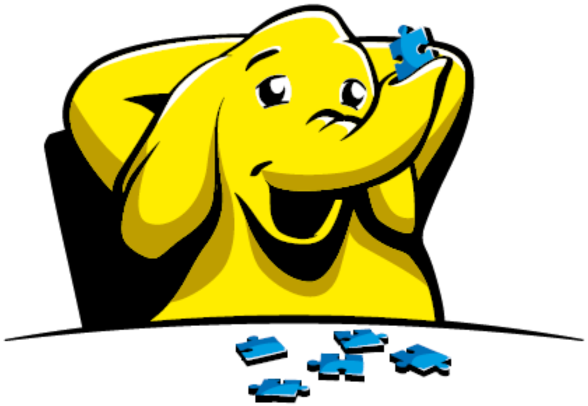
GENERAL OBJECT MATCHING WORKFLOW



ScaDS DRESDEN LEIPZIG

DEDOOP: EFFICIENT DEDUPLICATION WITH HADOOP

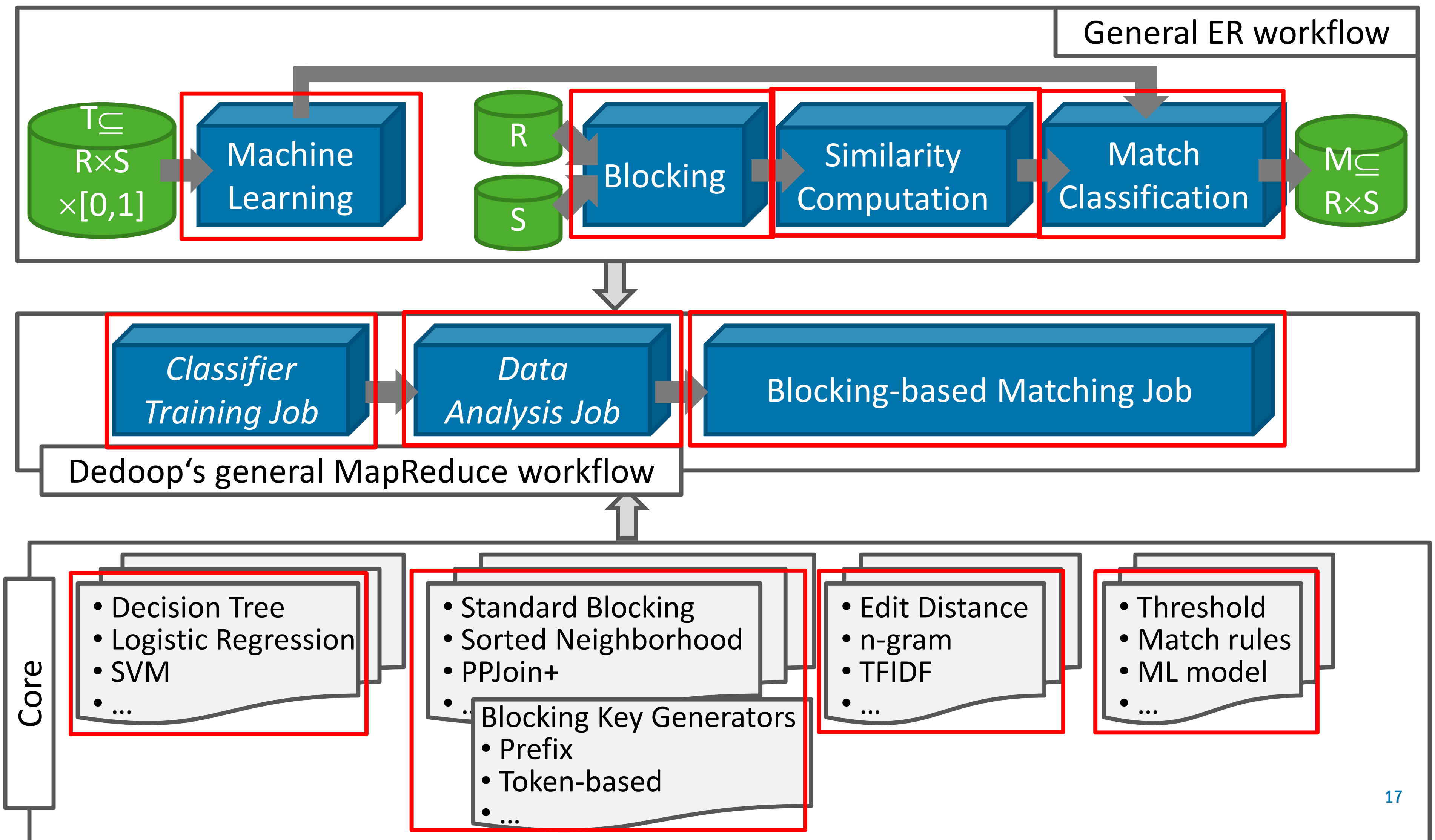
- Parallel execution of data integration/ match workflows with Hadoop
- Powerful library of match and blocking techniques
- Learning-based configuration
- GUI-based workflow specification
- Automatic generation and execution of Map/Reduce jobs on different clusters
- Automatic load balancing for optimal scalability
- Iterative computation of transitive closure (extension of MR-CC)



"This tool by far shows the most mature use of MapReduce for data deduplication"
www.hadoopsphere.com

ScaDS DRESDEN LEIPZIG

DEDOOP OVERVIEW



ScaDS DRESDEN LEIPZIG

AGENDA

- ScaDS Dresden/Leipzig
- Big Data Integration
  - Introduction
  - Matching product offers from web shops
  - DeDoop: Deduplication with Hadoop
- Privacy-preserving record linkage with PP-Join
  - Cryptographic bloom filters
  - Privacy-Preserving PP-Join (P4Join)
  - GPU-based implementation
- Big Graph Data
  - Graph-based Business Intelligence with BIIIG
  - GraDoop: Hadoop-based data management and analysis
- Summary and outlook

ScaDS DRESDEN LEIPZIG

PRIVACY FOR BIG DATA

- Need for comprehensive privacy support ("privacy by design")
  - Privacy-preserving publishing of datasets
  - Privacy-preserving record linkage
  - Privacy-preserving data mining
- Privacy-preserving record linkage
  - object matching with encrypted data to preserve privacy
  - conflicting requirements: high privacy, scalability and match effectiveness
  - use of central linking unit (Trusted third party) vs. symmetric approaches (Secure Multiparty Computing)

ScaDS DRESDEN LEIPZIG

PPRL WITH BLOOM FILTERS

- effective and simple encryption uses cryptographic bloom filters (Schnell et al, 2009)
- tokenize all match-relevant attribute values, e.g. using bigrams or trigrams
  - typical attributes: first name, last name (at birth), sex, date of birth, country of birth, place of birth
- map each token with a family of one-way hash functions to fixed-size bit vector (fingerprint)
  - original data cannot be reconstructed
- match of bit vectors (Jaccard similarity) is good approximation of true match result

SIMILARITY COMPUTATION - EXAMPLE

thomas

tho hom oma mas

tho
hom
oma

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	0	0	1	1	0	1	0	0	0	0	1	1	0	1

mas

h1(mas)= 3 | h2(mas)= 7 | h3(mas)= 11

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	0	1	1	1	0	1	0	0	0	1	1	1	0	1

thoman

tho hom oma man

tho
hom
oma

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	0	0	1	1	0	1	0	0	0	0	1	1	0	1

man

h1(man)= 2 | h2(man)= 0 | h3(man)= 13

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	0	1	1	0	1	0	0	0	0	1	1	0	1

$$\mathbf{Sim_{Jaccard}\,(r1, r2) = (r1 \land r2)\ /\ (r1 \lor r2)}$$

$$\mathbf{Sim_{Jaccard}\,(r1, r2) = 7/11}$$

PP-JOIN: POSITION PREFIX JOIN (XIAO ET AL, 2008)

- one of the most efficient *similarity join* algorithms
  - determine all pairs of records with $sim_{Jaccard}(x,y) \geq t$
- use of filter techniques to reduce search space
  - length, prefix, and position filter
- relatively easy to run in parallel
- good candidate to improve scalability for PPRL
- evaluate set bit positions instead of (string) tokens

ScaDS DRESDEN LEIPZIG

LENGTH FILTER

- matching records pairs must have similar lengths

$$\boxed{\mathrm{Sim}_{\mathrm{Jaccard}}(x, y) \geq t \Rightarrow |x| \geq |y| * t}$$

- length / cardinality: number of set bits in bit vector

- Example for minimal similarity t = 0,8:

ID	Bit vector	card.
B	1 0 1 0 0 0 0 0 1 1 0 0 0	4
C	0 0 0 1 1 1 1 1 1 1 0 0 0	7
A	0 1 0 1 1 1 1 1 1 1 0 0 0	8

length filter
$7 * 0.8 = 5.6 > 4$

  - record B of length 4 cannot match with C and all records with greater length (number of set positions), e.g., A

ScaDS DRESDEN LEIPZIG

PREFIX FILTER

- Similar records must have a minimal overlap α in their sets of tokens (or set bit positions)

$$Sim_{Jaccard}(x, y) \geq t \iff Overlap(x, y) \geq \alpha = \left\lceil \left(\frac{t}{1+t} * (|x|) + |y|) \right) \right\rceil$$

- Prefix filter approximates this test
  - reorder bit positions for all fingerprints according to their overall frequency from infrequent to frequent
  - exclude pairs of records without any overlap in their prefixes with

$$prefix\_length(x) = \lceil ((1-t)*|x|) + 1 \rceil$$

- Example (t= 0.8)

ID	reordered fingerprint	card.	prefix fingerprint
B	1 0 1 0 0 0 0 0 1 1 0 0 0 0	4	1 0 1
C	0 0 0 1 1 1 1 1 1 1 0 0 0 0	7	0 0 0 1 1 1
A	0 1 0 1 1 1 1 1 1 1 0 0 0 0	8	0 1 0 1 1

AND operation on prefixes shows non-zero result for C and A so that these records still need to be considered for matching

ScaDS DRESDEN LEIPZIG

P4JOIN: POSITION FILTER

- improvement of prefix filter to avoid matches even for overlapping prefixes
  - estimate maximally possible overlap and checking whether it is below the *minimal overlap α* to meet threshold t
  - *original position filter* considers the position of the last common prefix token

- revised position filter
  - record x, prefix 1 1 0 1 length 9
  - record y, prefix 1 1 1 length 8
  - highest prefix position (here fourth pos. in x) limits possible overlap with other record: the third position in y prefix cannot have an overlap with x
  - maximal possible overlap = #shared prefix tokens (2) + min (9-3, 8-3)= 7 < minimal overlap α = 8

ScaDS DRESDEN LEIPZIG

EVALUATION

- comparison between NestedLoop, P4Join, MultiBitTree
  - MultiBitTree: best filter approach in previous work by Schnell
    - applies length filter and organizes fingerprints within a binary tree so that fingerprints with the same set bits are grouped within sub-trees
    - can be used to filter out many fingerprints from comparison
- two input datasets R, S
  - determined with FEBRL data generator
    N=[100.000, 200.000, …, 500.000]. $|R|=1/5 \cdot N$, $|S|=4/5 \cdot N$
  - bit vector length: 1000
  - similarity threshold 0.8

ScaDS DRESDEN LEIPZIG

EVALUATION RESULTS

- runtime in minutes on standard PC

Approach	Dataset size N				
	100.000	200.000	300.000	400.000	500.000
NestedLoop	6,10	27,68	66,07	122,02	194,77
MultiBitTree	4,68	18,95	40,63	78,23	119,73
P4 Length filter only	3,38	20,53	46,48	88,33	140,73
P4 Length+Prefix	3,77	22,98	52,95	99,72	159,22
P4 Length+Prefix+Position	**2,25**	**15,50**	**40,05**	**77,80**	125,52

- similar results for P4Join and Multibit Tree
- relatively small improvements compared to NestedLoop

ScaDS DRESDEN LEIPZIG

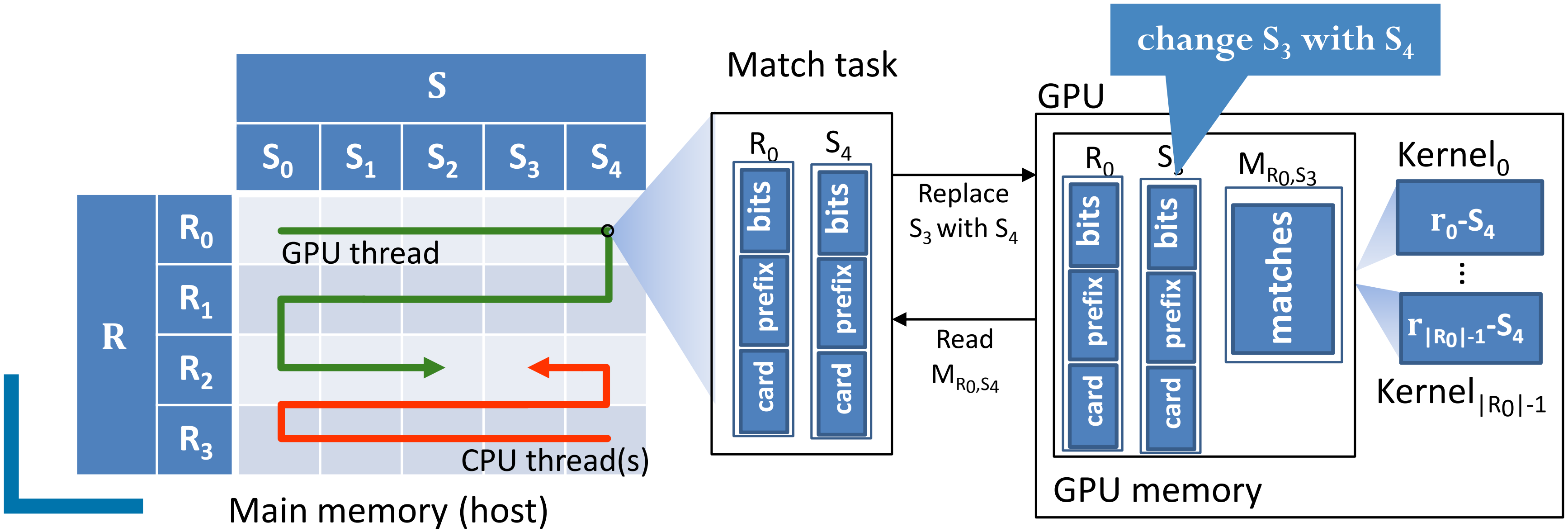
GPU-BASED PPRL

- Operations on bit vectors easy to compute on GPUs
  - Length and prefix filters
  - Jaccard similarity
- Frameworks CUDA und **OpenCL** support data-parallel execution of general computations on GPUs
  - program („kernel“) written in C dialect
  - limited to base data types (float, long, int, short, arrays)
  - no dynamic memory allocation (programmer controls memory management)
  - important to minimize data transfer between main memory and GPU memory

ScaDS DRESDEN LEIPZIG

EXECUTION SCHEME

- partition inputs R and S (fingerprints sorted by length) into equally-sized partitions that fit into GPU memory
  - generate match tasks per pair of partition
  - only transfer to GPU if length intervals per partition meet length filter
  - optional use of CPU thread to additionally match on CPU

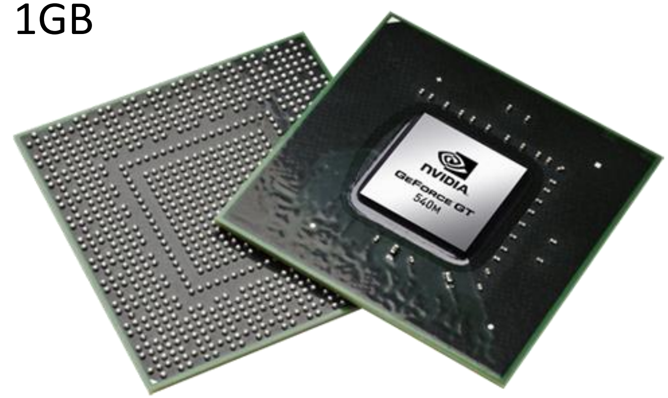


ScaDS DRESDEN LEIPZIG

GPU-BASED EVALUATION RESULTS

GeForce GT 610
- 48 Cuda Cores@810MHz
- 1GB
- 35€

GeForce GT 540M
- 96 Cuda Cores@672MHz
- 1GB



	100.000	200.000	300.000	400.000	500.000
GForce GT 610	0,33	1,32	2,95	5,23	8,15
GeForce GT 540M	0,28	1,08	2,41	4,28	6,67

- improvements by up to a factor of 20, despite low-profile graphic cards
- still non-linear increase in execution time with growing data volume

ScaDS DRESDEN LEIPZIG

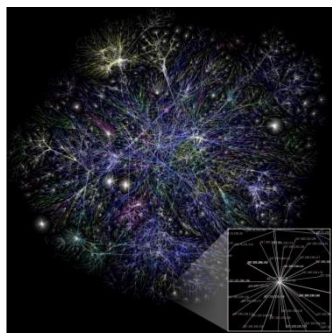
AGENDA

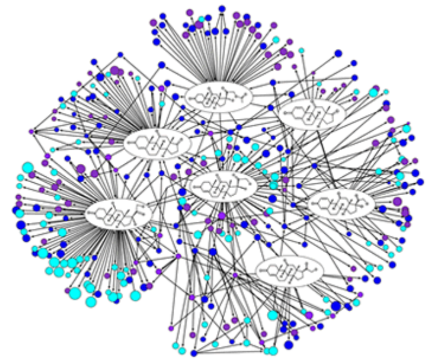
- ScaDS Dresden/Leipzig
- Big Data Integration
  - Introduction
  - Matching product offers from web shops
  - DeDoop: Deduplication with Hadoop
- Privacy-preserving record linkage with PP-Join
  - Cryptographic bloom filters
  - Privacy-Preserving PP-Join (P4Join)
  - GPU-based implementation
- Big Graph Data
  - Graph-based Business Intelligence with BIIG
  - GraDoop: Hadoop-based data management and analysis
- Summary and outlook

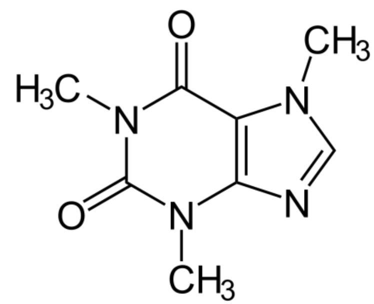
ScaDS DRESDEN LEIPZIG

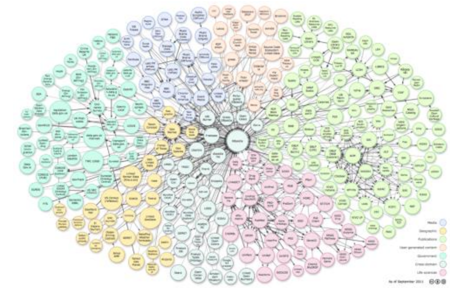
„GRAPHS ARE EVERYWHERE“

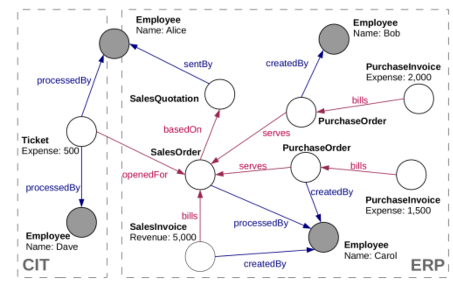
Social science	Engineering	Life science	Information science











Facebook
- ca. 1.3 Billion users
- ca. 340 friends per user

Twitter
- ca. 300 Million users
- ca. 500 Million Tweets per day

Internet
- ca. 2.9 Billion Users

Gene (human)
- 20,000-25,000
- ca. 4 Million individuals

Patients
- > 18 Millionen (Germany)

Illnesses
- > 30.000

World Wide Web
- ca. 1 Billion Websites

LOD-Cloud
- ca. 31 Billion Triples

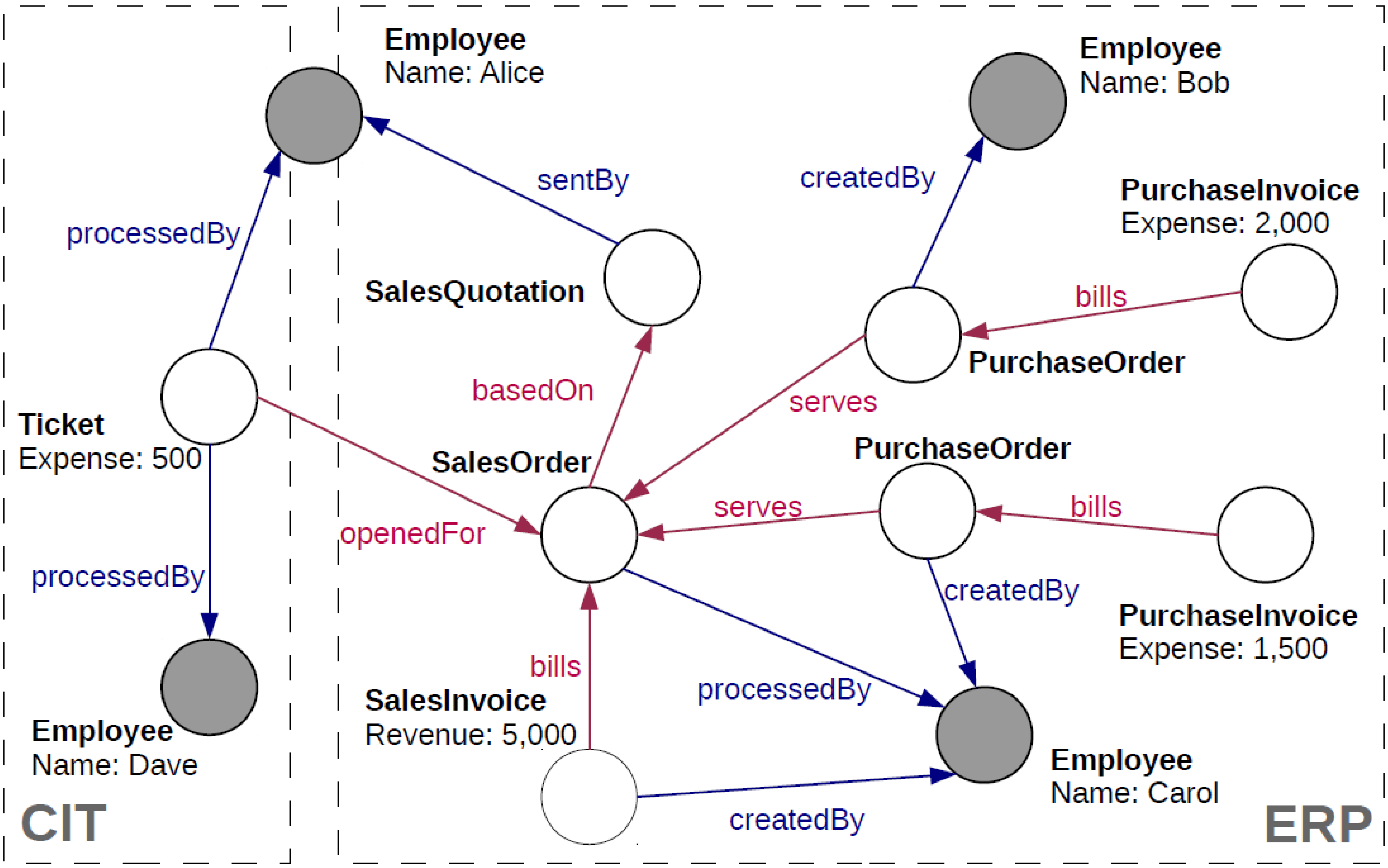
ScaDS DRESDEN LEIPZIG

USE CASE: GRAPH-BASED BUSINESS INTELLIGENCE

- Business intelligence usually based on relational data warehouses
  - enterprise data is integrated within dimensional schema
  - analysis limited to predefined relationships
  - no support for relationship-oriented data mining
- Graph-based approach (BIIIG)
  - Integrate data sources within an instance graph by preserving original relationships between data objects (transactional and master data)
  - Determine subgraphs (business transaction graphs) related to business activities
  - Analyze subgraphs or entire graphs with aggregation queries, mining relationship patterns, etc.

ScaDS DRESDEN LEIPZIG

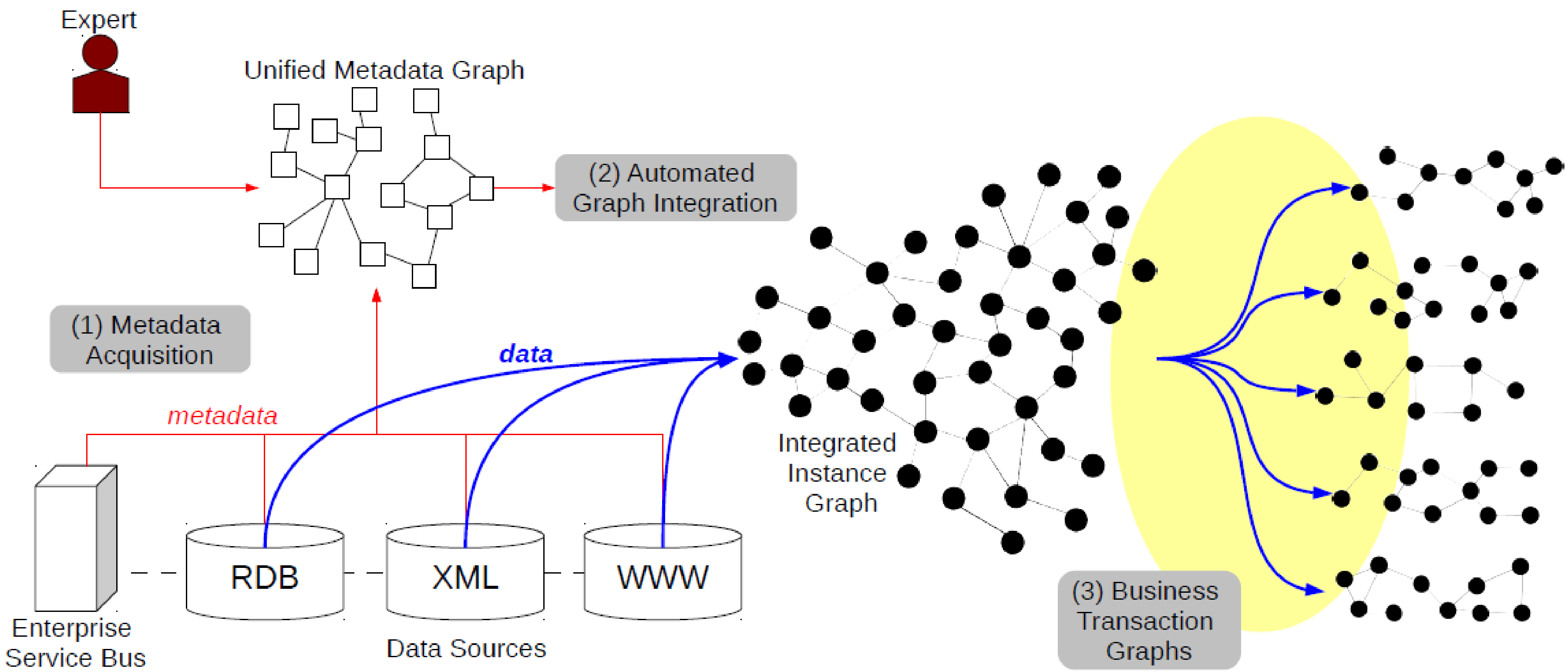
SAMPLE GRAPH



ScaDS DRESDEN LEIPZIG

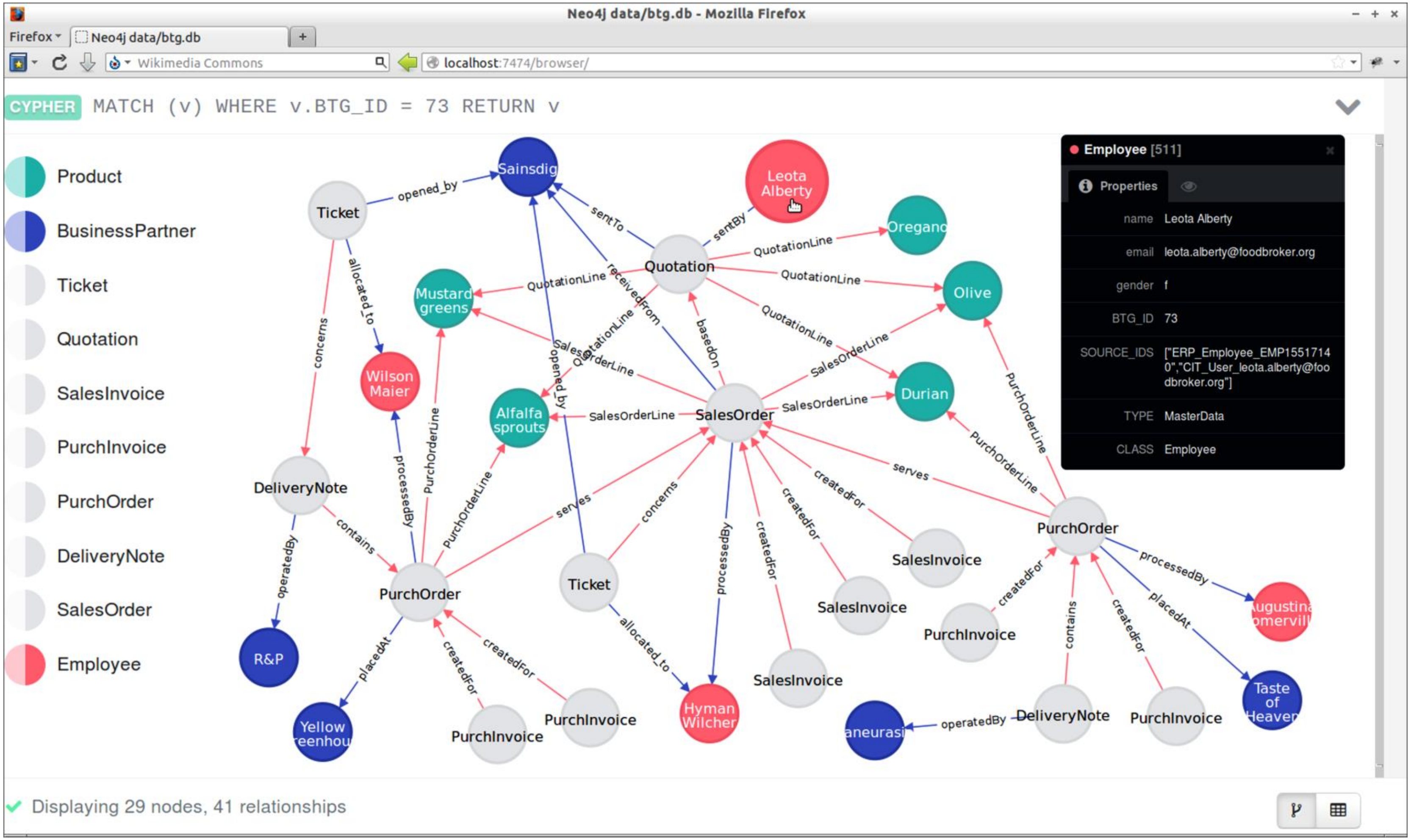
BIIIG DATA INTEGRATION AND ANALYSIS WORKFLOW

„Business Intelligence on Integrated Instance Graphs“



ScaDS DRESDEN LEIPZIG

SCREENSHOT FOR NEO4J IMPLEMENTATION



ScaDS DRESDEN LEIPZIG

GRAPH DATA MANAGEMENT

- Relational database systems
  - store vertices and edges in tables
  - utilize indexes, column stores, etc.
- Graph database system, e.g. Neo4J
  - use of property graph data model: vertices and edges have arbitrary set of properties (represented as key-value pairs)
  - focus on simple transactions and queries
- Distributed graph processing systems, e.g., Google Pregel, Apache Giraph, GraphX, etc.
  - In-memory storage of graphs in Shared Nothing cluster
  - parallel processing of general graph algorithms, e.g. page rank, connected components, ...

ScaDS DRESDEN LEIPZIG

WHAT'S MISSING?

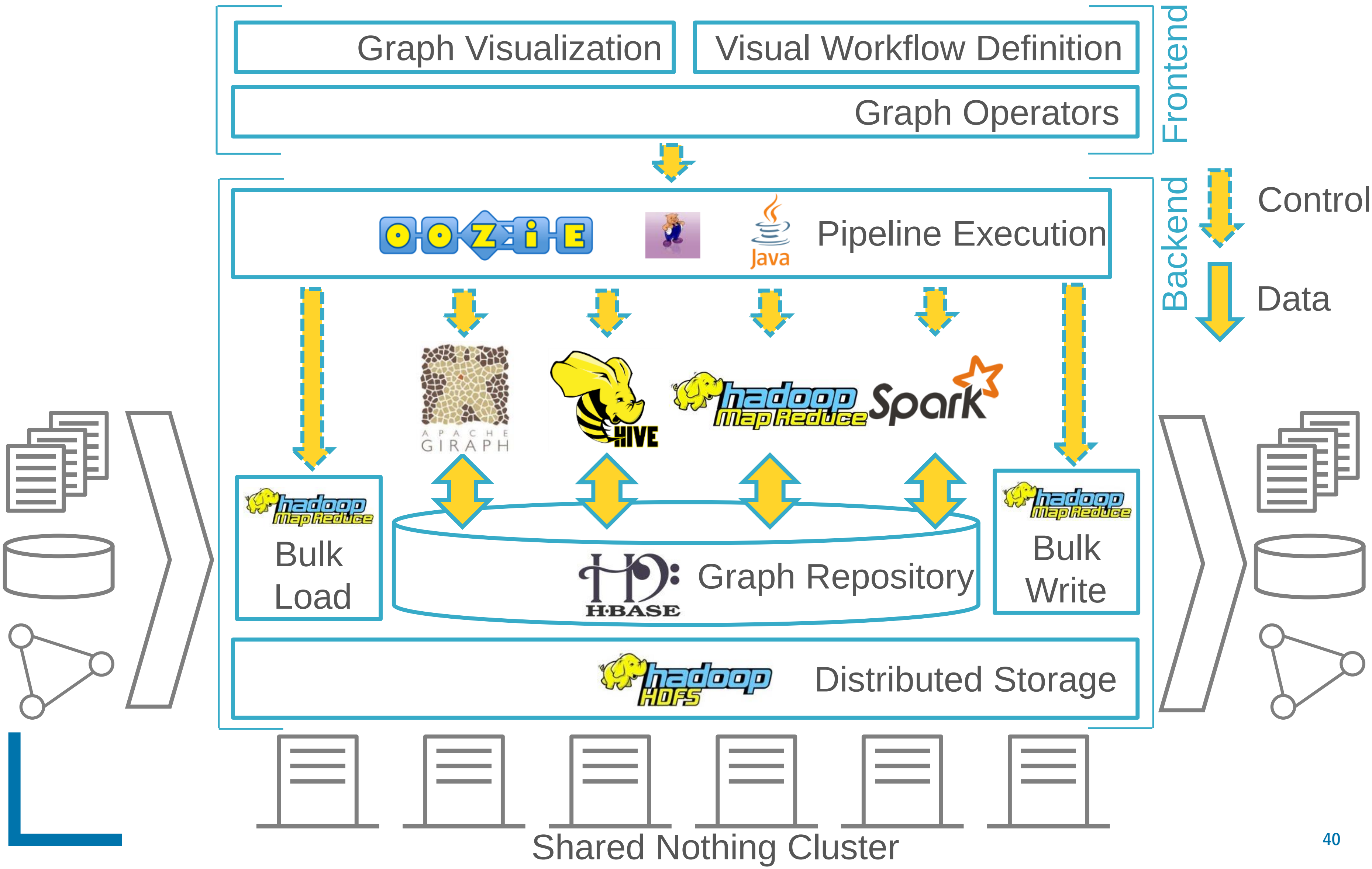
A comprehensive framework and research platform for efficient, distributed and domain independent graph analytics.

GRADOOP CHARACTERISTICS

- Hadoop-based framework for graph data management and analysis
- Graph storage in scalable distributed store, e.g., HBase
- Extended property graph data model
  - operators on graphs and sets of (sub) graphs
  - support for semantic graph queries and mining
- Leverages powerful components of Hadoop ecosystem
  - MapReduce, Giraph, Spark, Pig, Drill ...
- New functionality for graph-based processing workflows and graph mining

ScaDS DRESDEN LEIPZIG

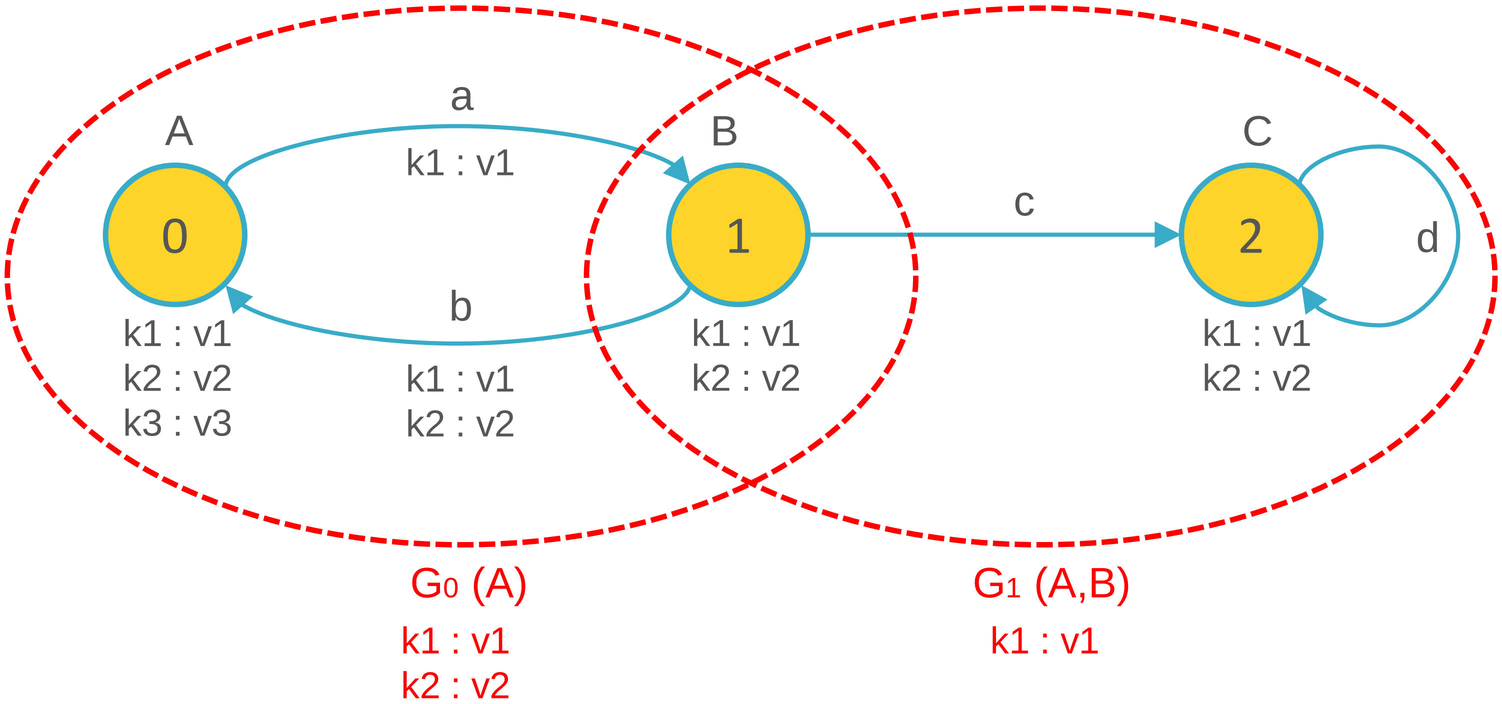
GRADOOP – HIGH LEVEL ARCHITECTURE



ScaDS DRESDEN LEIPZIG

EXTENDED PROPERTY GRAPH MODEL

Partitioned Directed Labeled Attributed Multigraph



ScaDS DRESDEN LEIPZIG

GRADOOP OPERATORS

Single Graph Operations

Operator	Input	Output
Aggregation $\gamma: \mathcal{G} \to (\mathbb{R} \cup \Sigma)$ $G \mapsto g$	Graph G	Number/String g
Subgraph Discovery $\theta_{\upsilon,\epsilon}: \mathcal{G} \to \mathbb{G}$ $G \mapsto \mathcal{G}$	Graph G Vertex map $\upsilon: V \to \mathbb{G}$ Edge map $\epsilon: E \to \mathbb{G}$	Graph set $\mathcal{G}$

- Summarization
- Pattern Match
- Projection

Graph Set Operations

Operator	Input	Output
Selection $\sigma_\varphi: \mathbb{G} \to \mathbb{G}$ $\mathcal{G} \mapsto \mathcal{G}'$	Graph set $\mathcal{G}$ Predicate $\varphi: \mathcal{G} \to \{0,1\}$	Graph set $\mathcal{G}'$

- Map
- Union
- Intersect
- Difference

Binary Graph Comparison

Operator	Input	Output
Similarity $\sim: \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ $\langle G_1, G_2 \rangle \mapsto s$	Graphs G_1, G_2	Similarity s

- Edit Steps
- Equivalence
- Equality

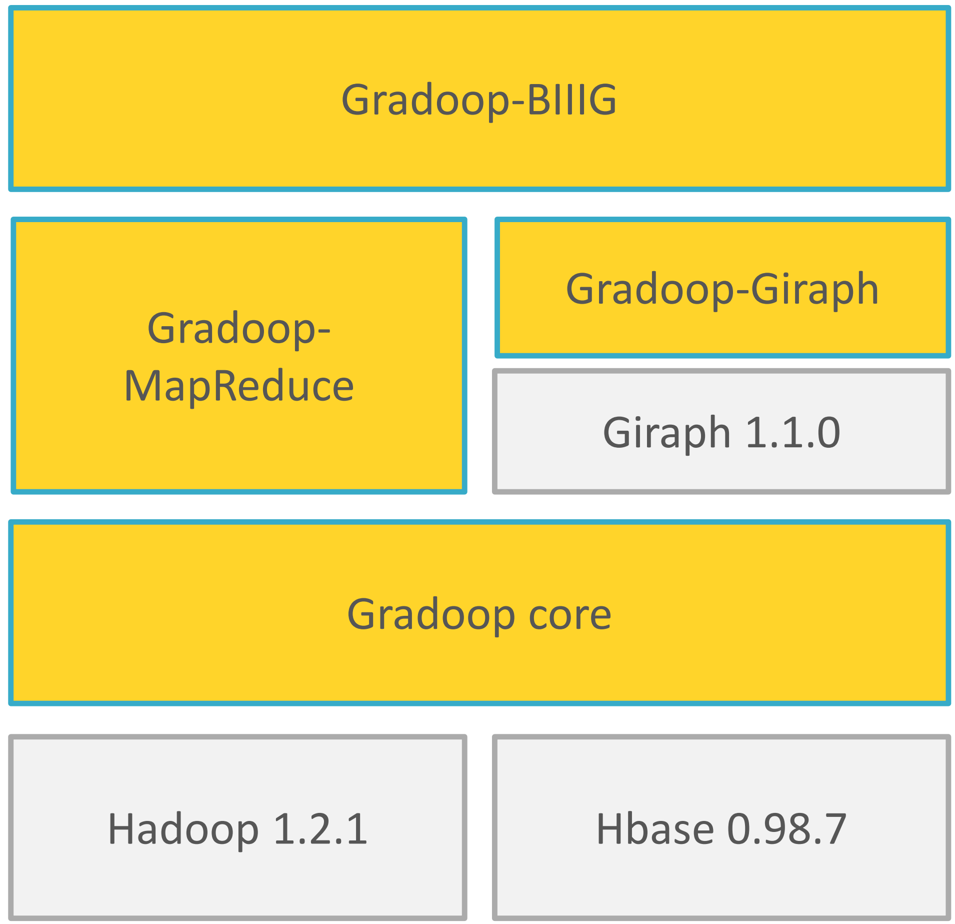
n-ary Graph Comparison

Operator	Input	Output
Frequent Subgraphs $\phi_t: \mathbb{G} \to \mathbb{G}$ $\mathcal{G} \mapsto \mathcal{G}'$	Graph set $\mathcal{G}$ Treshold $0 \leq t \leq 1$	Graph set $\mathcal{G}'$

- Inner Join
- Outer Join

ScaDS DRESDEN LEIPZIG

IMPLEMENTATION STATUS



BTG Analysis Pipeline
Data Import

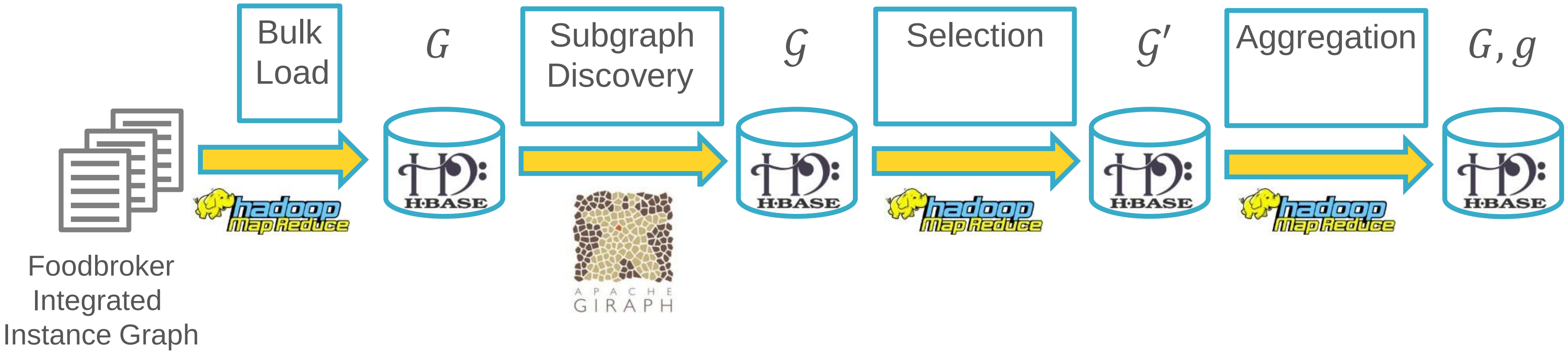
Subgraph Discovery
I/O Formats

Selection
Aggregation

EPG Model
HBaseGraphStore
Bulk Load
I/O Formats

ScaDS DRESDEN LEIPZIG

BIIG WITH GRADOOP



ScaDS DRESDEN LEIPZIG

AGENDA

- ScaDS Dresden/Leipzig
- Big Data Integration
  - Introduction
  - Matching product offers from web shops
  - DeDoop: Deduplication with Hadoop
- Privacy-preserving record linkage with PP-Join
  - Cryptographic bloom filters
  - Privacy-Preserving PP-Join (P4Join)
  - GPU-based implementation
- Big Graph Data
  - Graph-based Business Intelligence with BIIIG
  - GraDoop: Hadoop-based data management and analysis
- Summary and outlook

ScaDS DRESDEN LEIPZIG

SUMMARY

- ScaDS Dresden/Leipzig
  - Research focus on data integration, knowledge extraction, visual analytics
  - broad application areas (scientific + business-related)
  - solution classes for applications with similar requirements
- Big Data Integration
  - Big data poses new requirements for data integration (variety, volume, velocity, veracity)
  - comprehensive data preprocessing and cleaning
  - Hadoop-based approaches for improved scalability, e.g. Dedoop
  - Usability: machine-learning approaches, GUI, ...

ScaDS DRESDEN LEIPZIG

SUMMARY (2)

- Scalable Privacy-Preserving Record Linkage
  - bloom filters allow simple, effective and relatively efficient match approach
  - Privacy-preserving PP-Join (P4JOIN) achieves comparable performance to multibit trees but easier to parallelize
  - GPU version achieves significant speedup
  - further improvements needed to reduce quadratic complexity
- Big Graph Data
  - high potential of graph analytics even for business data (BIIIG)
  - GraDoop: infrastructure for entire processing pipeline: graph acquisition, storage, integration, transformation, analysis (queries + graph mining), visualization
  - leverages Hadoop ecosystem including graph processing systems
  - extended property graph model with powerful operators

ScaDS DRESDEN LEIPZIG

OUTLOOK

- Parallel execution of more diverse data integration workflows for text data, image data, sensor data, etc.
  - learning-based configuration to minimize manual effort (active learning, crowd-sourcing)
- Holistic integration of many data sources (data + metadata)
  - clustering across many sources
  - N-way merging of related ontologies (e.g. product taxonomies)
- Improved privacy-preserving record linkage
  - better scalability, also for n-way (multi-party) PPRL
- Big Graph data management
  - complete processing framework
  - improved usability

ScaDS DRESDEN LEIPZIG

REFERENCES

- H. Köpcke, A. Thor, S. Thomas, E. Rahm: *Tailoring entity resolution for matching product offers*. Proc. EDBT 2012: 545-550
- L. Kolb, E. Rahm: *Parallel Entity Resolution with Dedoop*. Datenbank-Spektrum 13(1): 23-32 (2013)
- L. Kolb, A. Thor, E. Rahm: Dedoop: *Efficient Deduplication with Hadoop*. PVLDB 5(12), 2012
- L. Kolb, A. Thor, E. Rahm: *Load Balancing for MapReduce-based Entity Resolution*. ICDE 2012: 618-629
- L. Kolb, Z. Sehili, E. Rahm: *Iterative Computation of Connected Graph Components with MapReduce*. Datenbank-Spektrum 14(2): 107-117 (2014)
- A. Petermann, M. Junghanns, R. Müller, E. Rahm: *BIIIG : Enabling Business Intelligence with Integrated Instance Graphs*. Proc. 5th Int. Workshop on Graph Data Management (GDM 2014)
- A. Petermann, M. Junghanns, R. Müller, E. Rahm: *Graph-based Data Integration and Business Intelligence with BIIIG.* Proc. VLDB Conf., 2014
- E. Rahm, W.E. Nagel: *ScaDS Dresden/Leipzig: Ein serviceorientiertes Kompetenzzentrum für Big Data*. Proc. GI-Jahrestagung 2014: 717
- R.Schnell, T. Bachteler, J. Reiher: *Privacy-preserving record linkage using Bloom filters*. BMC Med. Inf. & Decision Making 9: 41 (2009)
- Z. Sehili, L. Kolb, C. Borgs, R. Schnell, E. Rahm: *Privacy Preserving Record Linkage with PPJoin*. Proc. BTW Conf. 2015
- C. Xiao, W. Wang, X. Lin, J.X. Yu: *Efficient Similarity Joins for Near Duplicate Detection*. Proc. WWW 2008