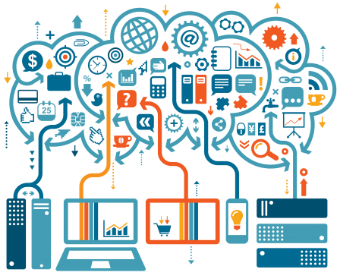


UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

Entity Resolution in the Big Data Context

Prof. Sonia Bergamaschi
Department of Engineering "Enzo Ferrari"
sonia.bergamaschi@unimore.it
www.dbgroup.unimore.it

Outline

- **Who I Am**
- From Data Integration to Big Data Integration
- Entity Resolution (a.k.a. Record Linkage)
- Privacy-Preserving Record Linkage (PPRL)
- PPRL with MOMIS

Who I Am

Prof. Sonia Bergamaschi

- Full Professor at the University of Modena and Reggio Emilia
  Engineering dept "Enzo Ferrari"
- Email: sonia.bergamaschi@unimore.it
- Leader of the Database Research Group (DBGroup)
- Dean of the ICT Doctorate
- UNIMORE Delegate for ICT Technologies
- ACM Distinguished Researcher
- IEEE Senior Member
- >300 publications in international conferences and journals
  DBLP · Google Scholar · Scopus

DBGroup at Glance

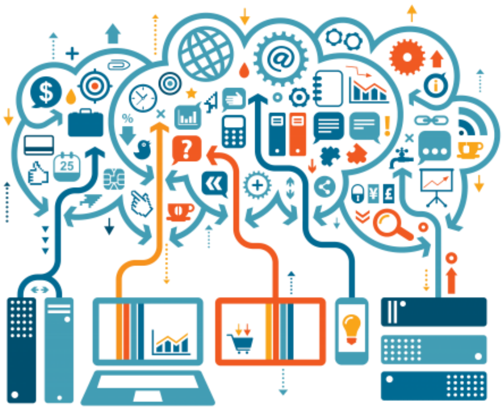
- Current Members:
  - 6 Faculty
    - Prof. Sonia Bergamaschi
    - Prof. Domenico Beneventano
    - Prof. Francesco Guerra
    - Prof. Laura Po
    - Prof. Maurizio Vincini
    - Giovanni Simonini, PhD (RTDB)
  - 1 Postdoc
    - Luca Gagliardelli, PhD (RTDA)
  - 4 ICT PhD Students
    - Luca Zecchini (2nd year, *Task-driven Big Data Integration*)
    - Adeel Aslam (1st year, *Big Data and Artificial Intelligence for the enhancement of energy virtuosity*, ER Grant)
    - Giulio De Sabbata (1st year, *Big Data and Artificial Intelligence to the efficiency of production processes in industrial manufacturing*, DataRiver)
    - Ambra Di Piano (1st year, *Deep learning in real-time on the astrophysical data obtained from the Cerenkov CTA Observatory*, INFN)
- Member of the Italian **CINI Big Data Lab**
- **DataRiver**: a spin-off (now innovative SME) to deploy the MOMIS data integration system

DataRiver
open source data management

Outline

- Who I Am
- **From Data Integration to Big Data Integration**
  - **Data Integration**
  - **Big Data**
  - **Technologies for Big Data**
    - **Big Data Management**
    - **Big Data Science**
    - **Big Data Integration**
  - **(Big) Data Integration with MOMIS**
- Entity Resolution (a.k.a. Record Linkage)
- Privacy-Preserving Record Linkage
- Some Real-World Applications

Data Integration



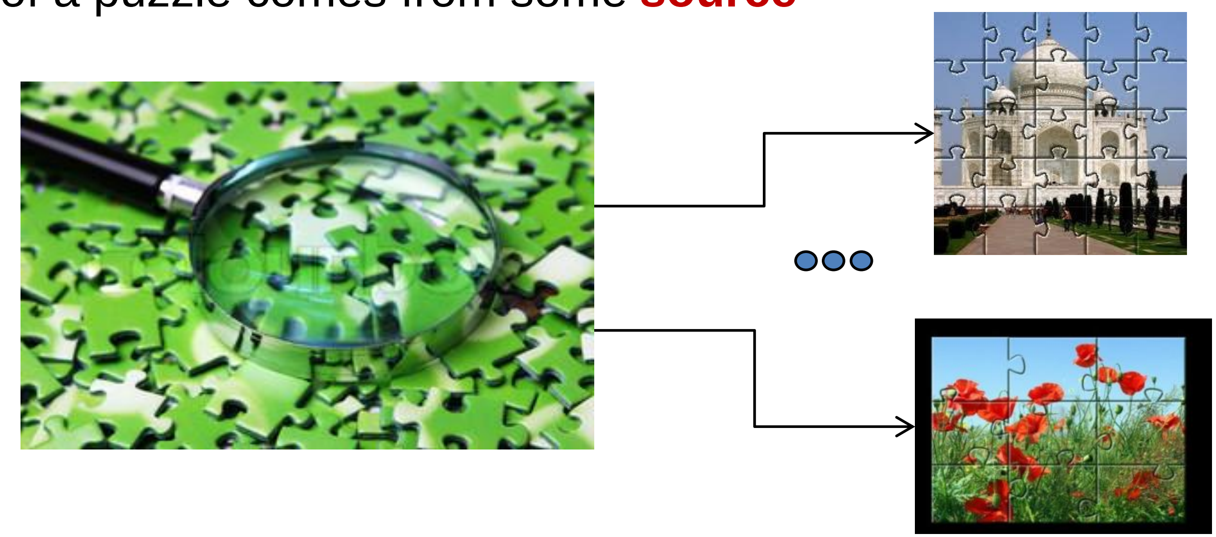
- Data Integration is the process of consolidating data from a **set of heterogeneous data sources** into a **single uniform dataset** or view on the data. *(Christian Bizer)*

- The integrated dataset should:
  - Correctly and completely represent the content of all data sources;
  - Use a single data model and a single schema;
  - Only contain a single representation of every real-world entity;
  - Not contain any conflicting data about single entities.

- To achieve this, Data Integration needs to resolve various types of **heterogeneity** that exist between data sources.

Data Integration

- The discipline of Data Integration comprises the practices, architectural techniques and tools that ingest, transform, combine and provision data across the spectrum of information types in the enterprise and beyond in order to meet the data consumption requirements of all applications and business processes.
- Applications of Data Integration:
  – Business, science, government, the Web, health… pretty much everywhere
- Data Integration = solving lots of puzzles
  – Each puzzle (e.g., Taj Mahal) is an **integrated entity**
  – Each piece of a puzzle comes from some **source**



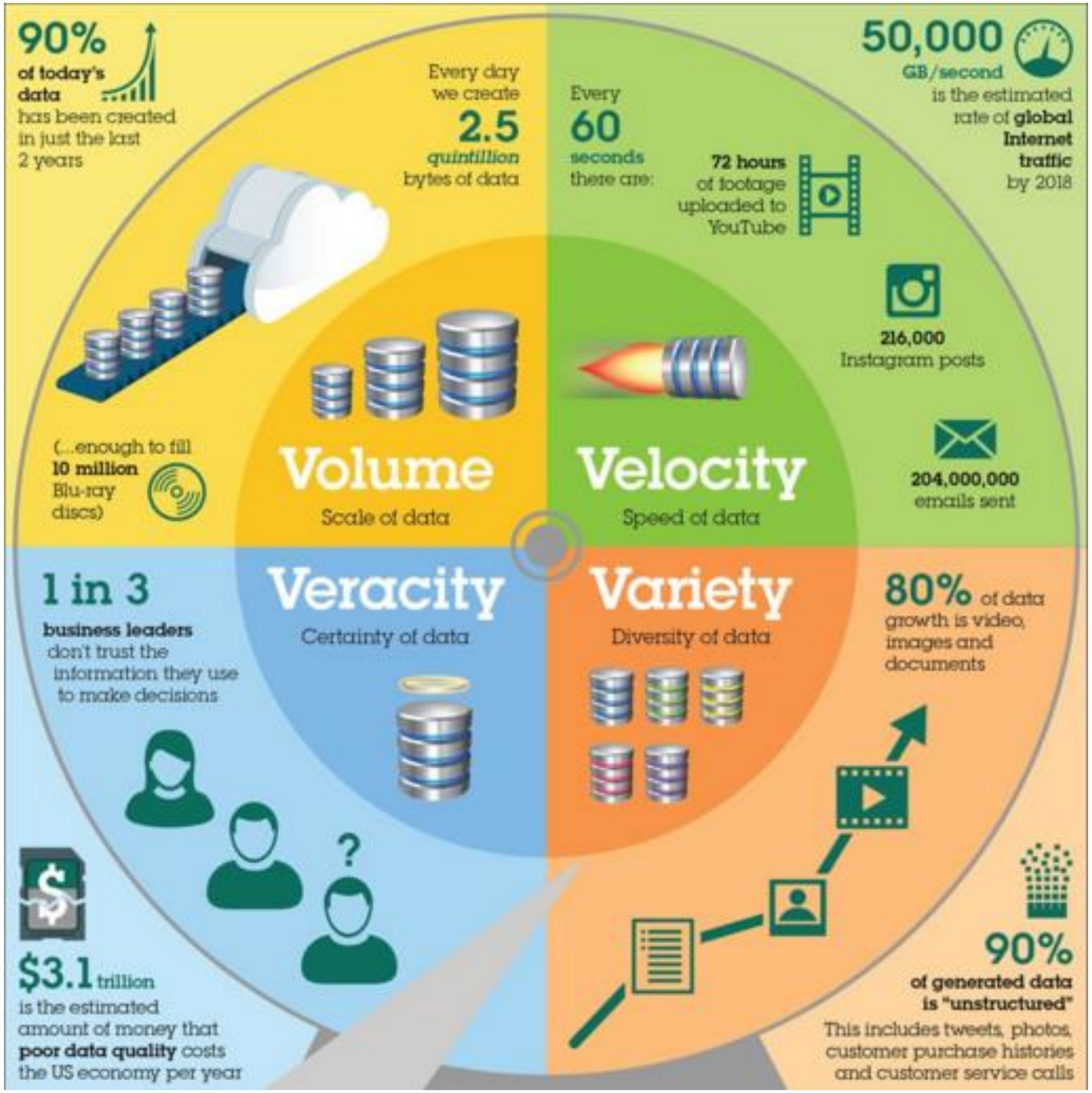
Big Data

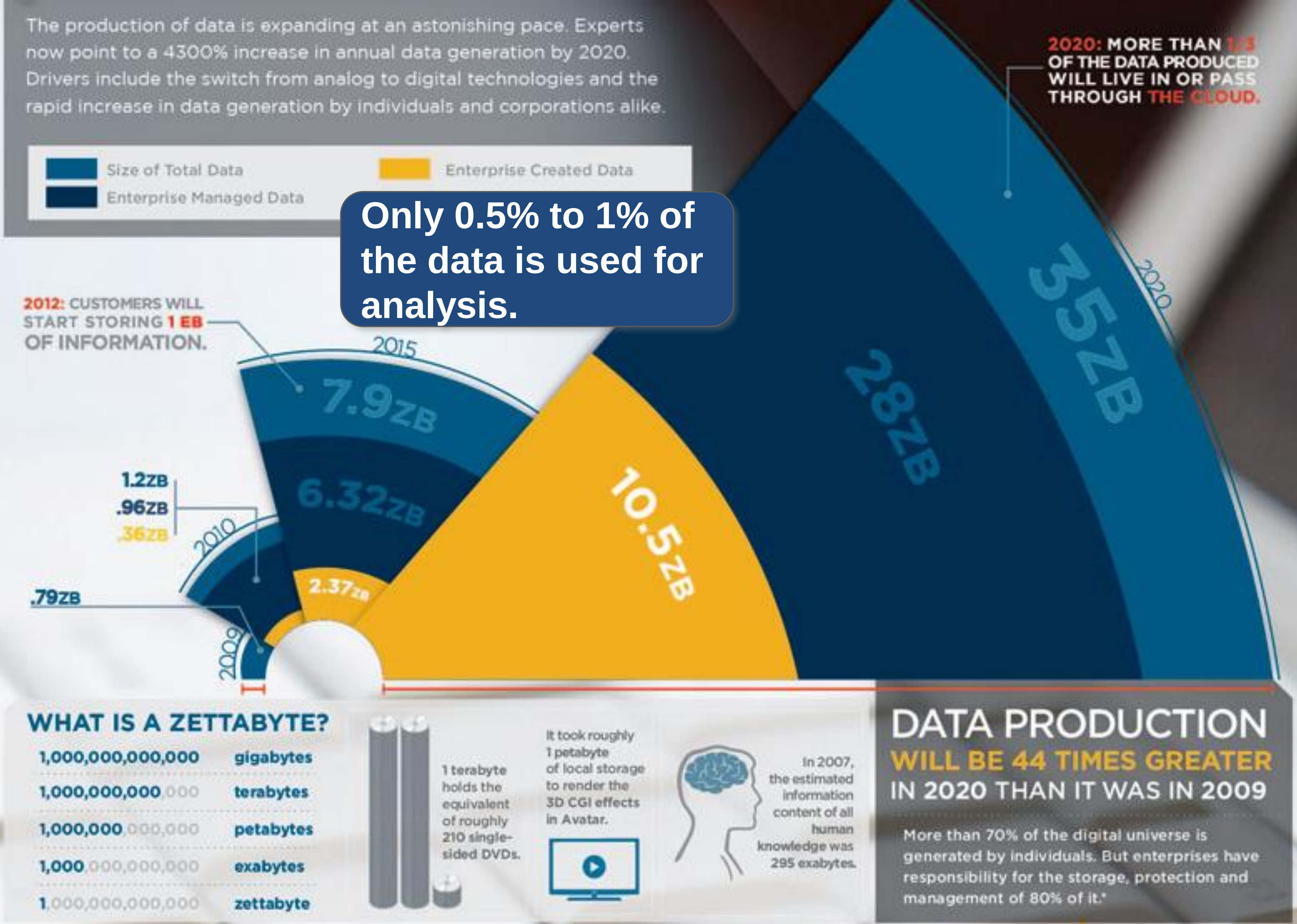
Big Data is a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools. The challenges include capture, curation, storage, search, sharing, analysis, and visualization. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found. (https://en.wikipedia.org/wiki/Big_data)

Big Data - Full faith in the power of data

The quest for knowledge used to begin with grand theories.
Now it begins with massive amounts of data. **Welcome to the Petabyte Age**!

The FOUR Vs of Big Data





Only 0.5% to 1% of the data is used for analysis.

http://www.csc.com/insights/flxwd/78931-big_data_growth_just_beginning_to_explode
http://www.guardian.co.uk/news/datablog/2012/dec/19/big-data-study-digital-universe-global-volume

Velocity

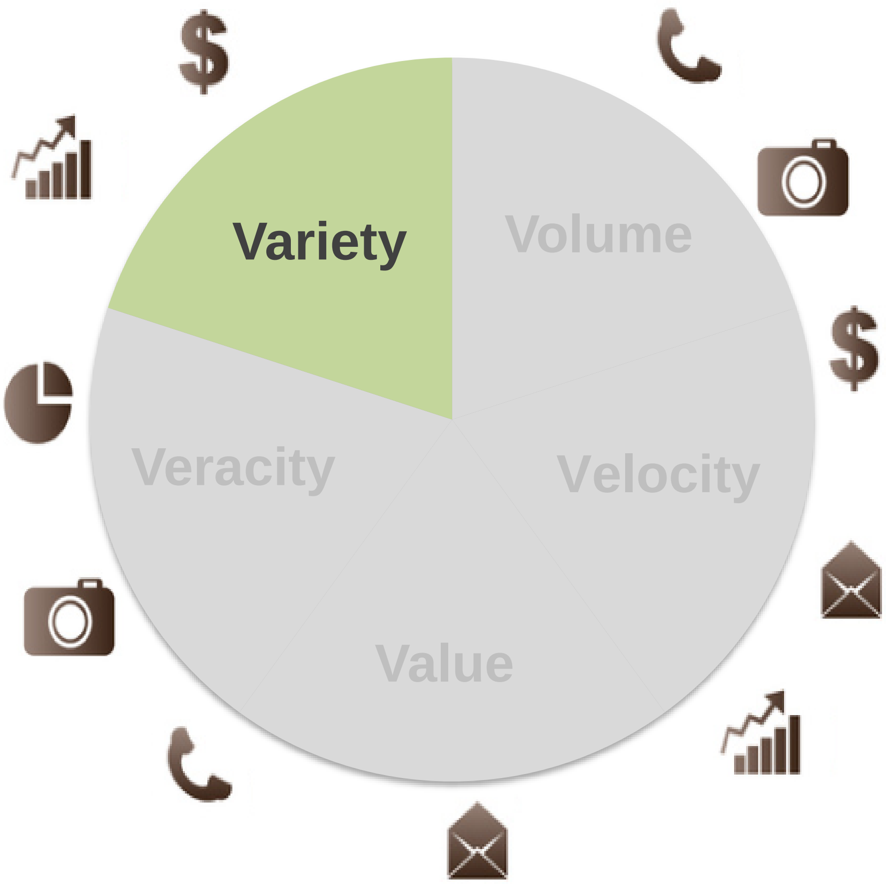
Fast Data

Rapid Changes

Real-Time/Stream Analysis

Current application examples: financial services, stock brokerage, weather tracking, movies/entertainment and online retail

Increasing Variety of data types

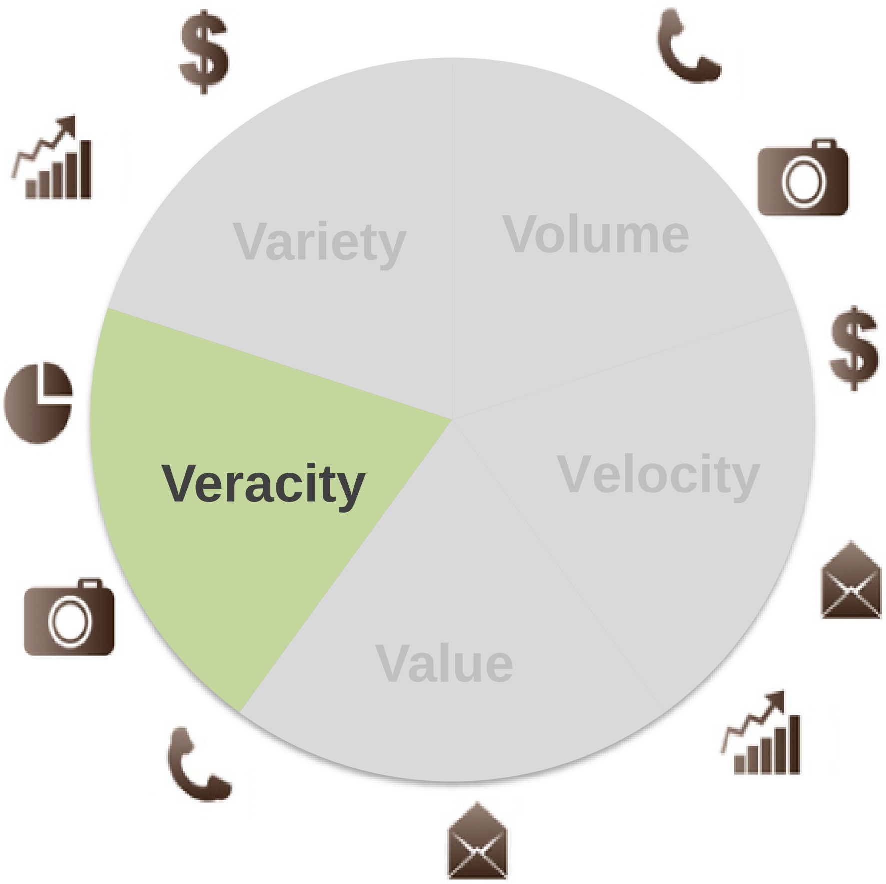


Data today comes in all types of formats:

from traditional databases to RDF data stores created by end users and OLAP systems

to text documents, email, meter-collected data, video, audio, stock ticker data and financial transactions.

We see increasing veracity (or accuracy) of data



Refers to the ***messiness*** or ***trustworthiness*** of the data. With many forms of big data ***quality*** and ***accuracy*** are ***less controllable***

(just think of Twitter posts with hash tags, abbreviations, typos and colloquial speech as well as the reliability and accuracy of content)

but technology now allows us to work with this type of data.

Value – The most important V of all!

- Then there is another V to take into account when looking at Big *Data: Value!*

- Having access to big data is no good unless we can turn it into value.

- What technologies?

Technologies for Big Data

- Big Data Management
- Big Data Science
- Big Data Integration

Database Philosophy

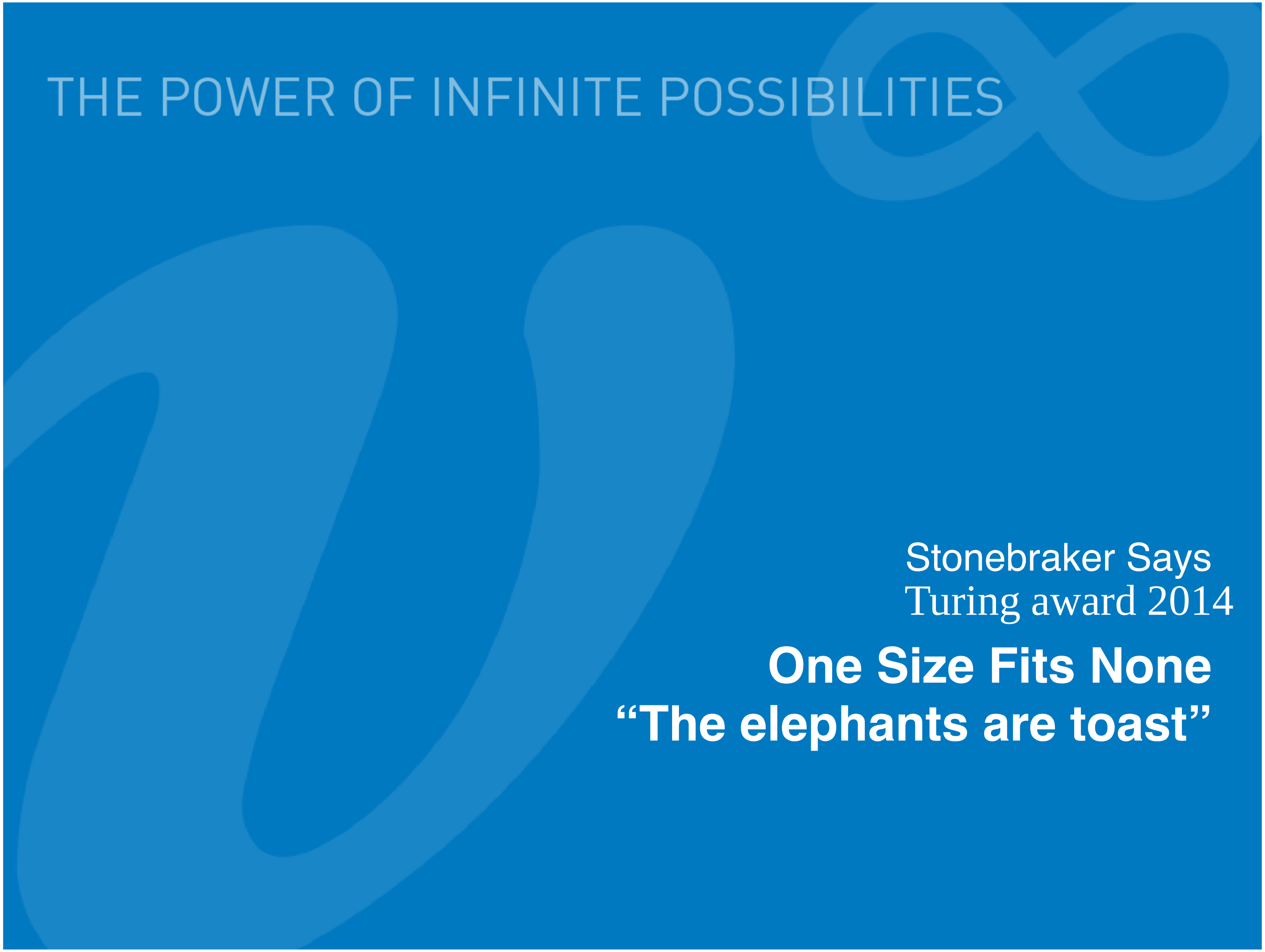
God made integers,
all else is the work of man.

(Leopold Kronecker, 19th Century Mathematician)

Codd made relations,
all else is the work of man.

(Raghu Ramakrishan, DB Textbook Author)

Traditional RDBMS



At This Point, RDBMS is “long in the tooth”'

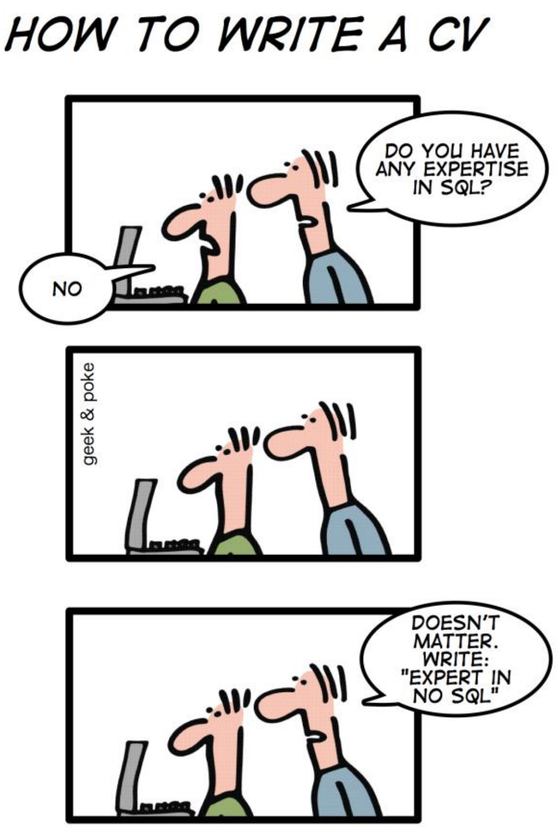
There are at least 6 (non trivial) markets where a row store can be clobbered by a specialized architecture !

- Warehouse (Vertica, Red Shift, Sybase IQ, DW Appliances, …)
- OLTP (VoltDB, HANA, Hekaton, …)
- RDF (Vertica, …)
- Text (Google, Yahoo, …)
- Scientific data (R, MatLab, SciDB, …)
- Data Streaming (Storm, Spark Streaming, InfoSphere, …)

An emerging "movement" around <u>non-relational</u> software for Big Data

- NOSQL stands for "Not Only SQL" (but is not entirely agreed upon), where SQL doesn't really mean the query language, but instead it denotes the traditional relational DBMS.

- Google **Bigtable & Mapreduce**, **Memcached**, and Amazon's **Dynamo** are the "proof of concept" that inspired many of the NOSQL systems:
  - Memcached demonstrated that in-memory indexes can be highly scalable, distributing and replicating objects over multiple nodes
  - Dynamo pioneered the idea of *eventual consistency* as a way to achieve higher availability and scalability
  - BigTable demonstrated that persistent record storage could be scaled to thousands of nodes & Mapreduce introduces parallel computation for distributed data platforms.



Leverage the NoSQL boom

Challenges – Selection of the Big Data Technology

- **Volume, Velocity**

  Calling for new **Big Data systems:**

  – **Big Data Management Systems: *NOSQL & more***

  Many more…

  – **Big Data Analysis Systems:**

  - **Batch + Streaming**

  Many more…

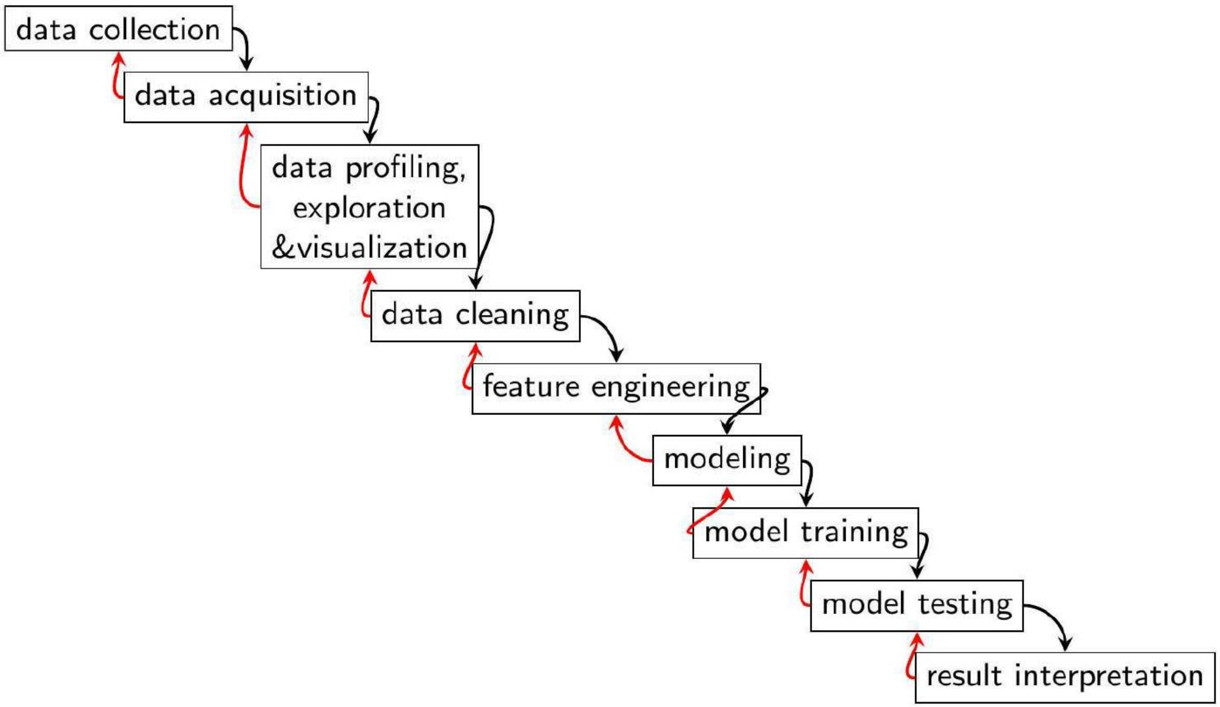
Not only Relational Database Management Systems and Business Intelligence

The Data Science Cake

Ingredients:
50g statistics
120g linear algebra
200g programming
1kg visualisation
300g software engineering

Additional skills:
creativity
out of the box thinking
grit
team spirit

The Data Science Pipeline/Waterfall Model



This is **at the same time** a process model **and** a dataflow.

From Jens Dittrich (Saarland University)

Data Integration

+

Data Analysis

(Business Intelligence, Statistics, Data Mining, Math

+

Data-Driven Artificial Intelligence)

Data-Driven AI

- From the Big Data era people do not focus on improving the quality of data, but just add more data to overcome errors from noisy and poor-quality information;
- In a recent talk, Andrew Ng states that 99% of the papers are model-centric;
- As a result, many models do not work well on real data;
- A recent paper from Google researchers analyzes the work of 53 AI practitioners, reporting that *"data cascades—compounding events causing negative, downstream effects from data issues—triggered by conventional AI/ML practices that undervalue data quality… are pervasive (92% prevalence), invisible, delayed, but often avoidable."*

Model-Centric	Data-Centric
- Collects as much data as possible - Iteratively **improves the model** to deal with the noise in the data	- Holds the model fixed - Iteratively **improves the quality of the data** to obtain good results

Data-Driven AI

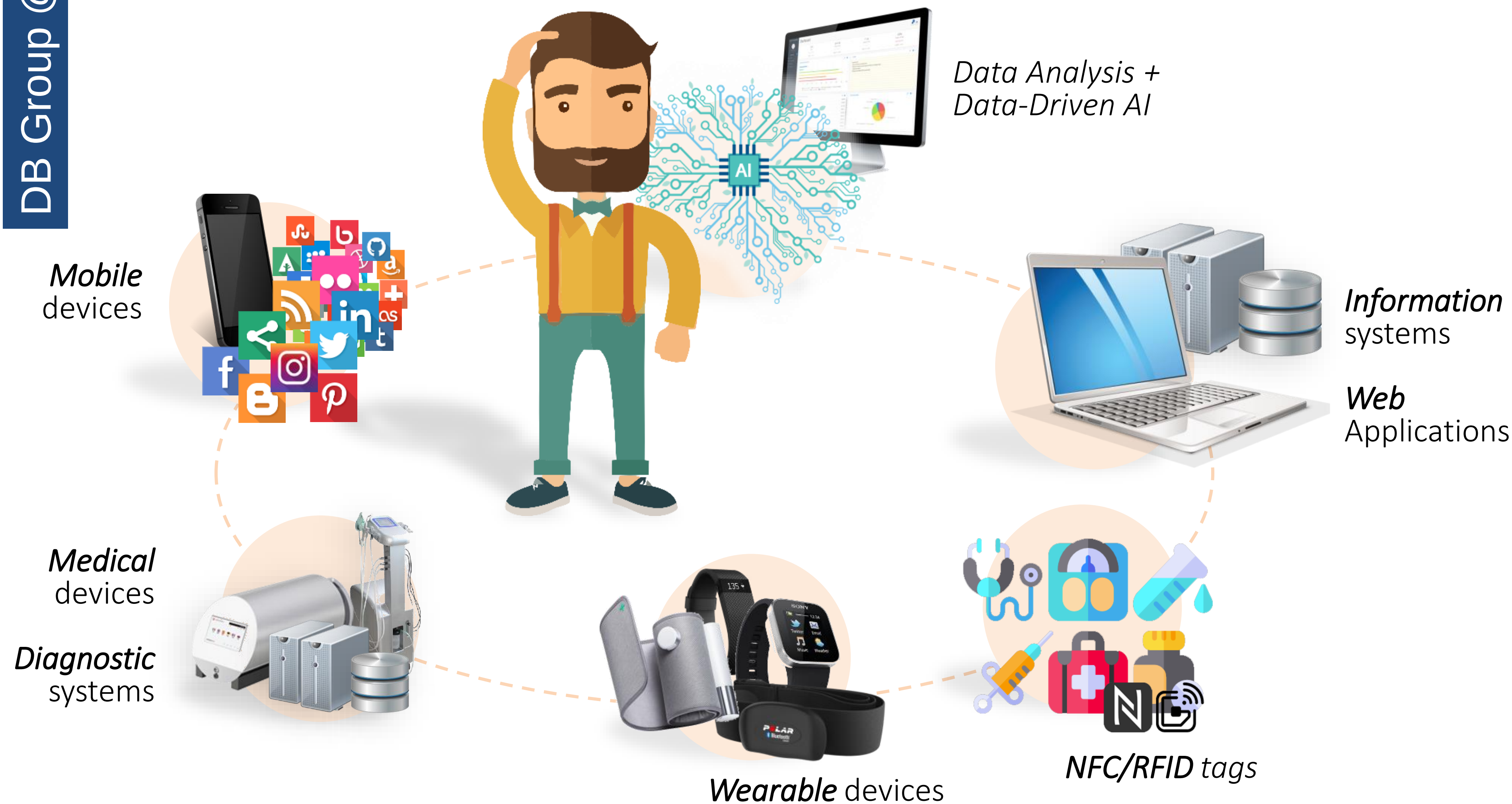
- **Data education lack** of adequate training on AI data quality, collection, and ethics. AI courses focus on toy datasets with clean values, but AI in practice requires the creation of data pipelines, often from scratch, going from ground truth to model maintenance.

- We have to define a **systematic pipeline** to improve the quality of data.

- Systematic **improvement of data quality** on a basic model is better than using the state-of-the-art models with low-quality data.

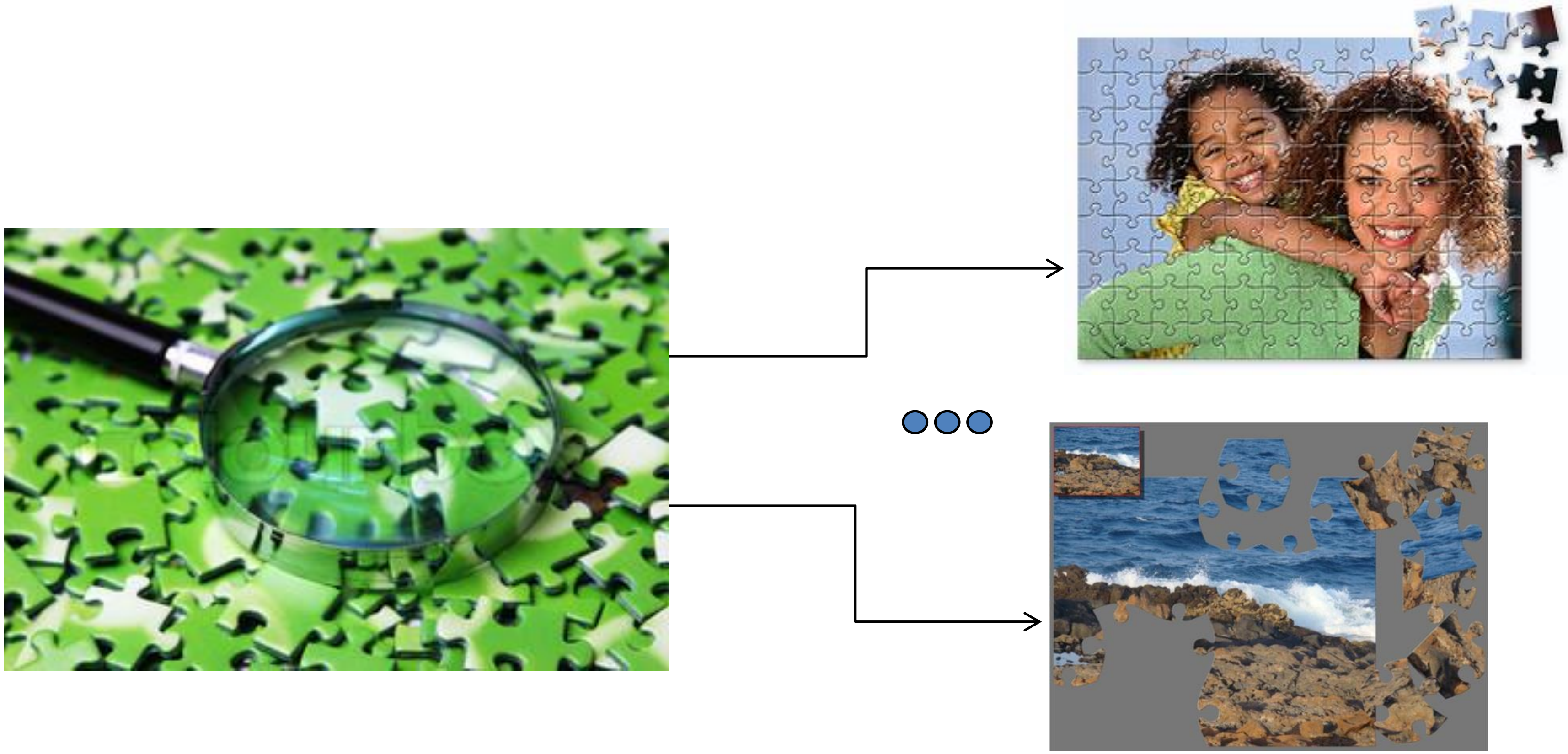
- In a recent talk, Andrew Ng states that **good data** for ML/AI:
  - Is defined consistently (the label definition is unambiguous);
  - Covers important cases (good coverage of inputs);
  - Has a feedback from the production data;
  - Is sized appropriately.

The Need for Big Data Integration: the example of eHealth



From Data Integration to Big Data Integration

- Data Integration = solving lots of puzzles
  - Big data integration → **big messy** puzzles
  - E.g., missing, duplicate, damaged pieces



(Big) Data Integration as a New Commercial Software

According to Gartner:

- ✓ Gartner estimates that the Data Integration tool market generated more than $2.7 billion in software revenue (in constant currency) at the end of 2016.
- ✓ A projected five-year compound annual growth rate of 6.32% will bring the total market revenue to around $4 billion in 2021 (see "Forecast: Enterprise Software Markets, Worldwide, 2014-2021, 2Q17 Update").
- ✓ ***$3.3 billion software revenue in 2020.***

Market Overview:

- ✓ The biggest change in the market from 2016 is the pervasive yet elusive demand for metadata-driven solutions.
- ✓ Consumers are asking for hybrid deployment not just in the cloud and on-premises but also across multiple data tiers throughout broad deployment models, plus the ability to blend data integration with application integration platforms (which is metadata driven in combination with workflow management and process orchestration) and a supplier focus on product and delivery initiatives to support these demands.

(Big) Data Integration in the Research Community

- The research community has been investigating Data Integration for more than 30 years: different research communities (database, artificial intelligence, semantic web) have been developing and addressing issues related to Data Integration:
  - Definitions, architectures, classification of the problems to be addressed;
  - Different approaches have been proposed and benchmarks developed.
- **Open issues**
  - Uncertainty, **Provenance**, and **Cleaning**;
  - **Lightweight Integration**:
  - Visualizing Integrated Data;
  - Integrating Social Media;
  - **Big Data Integration**

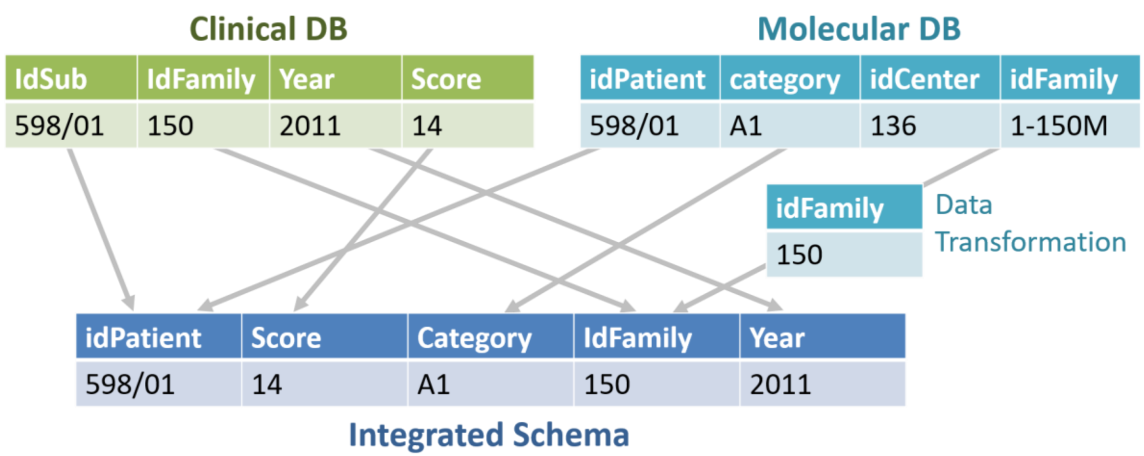
MoMIS
Mediator environment for Multiple Information Sources

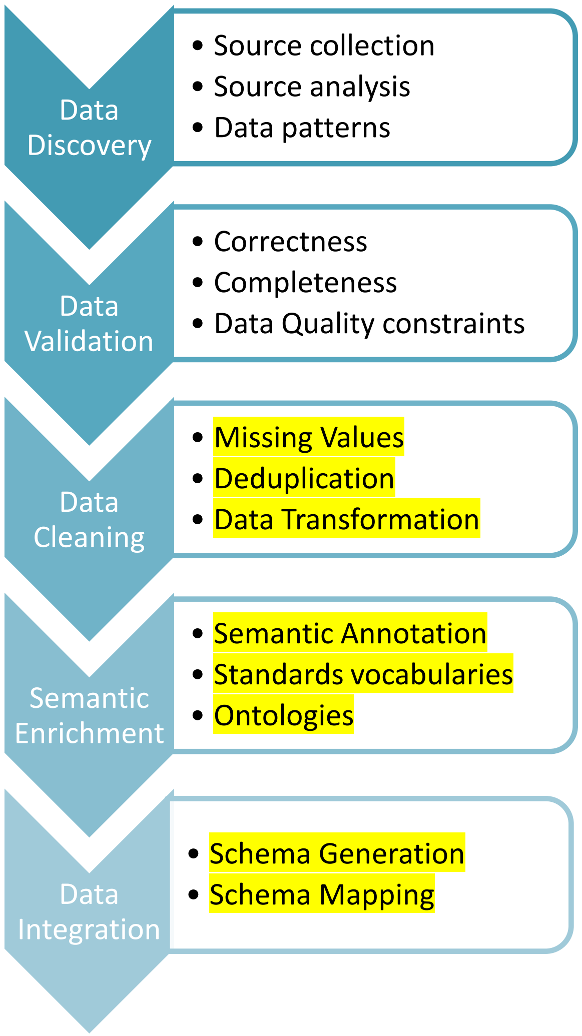
[1] S. Bergamaschi, S. Castano, M. Vincini: *Semantic Integration of Semistructured and Structured Data Sources*. ACM SIGMOD Record 28(1): 54-59 (1999)
[2] S. Bergamaschi et al.: *From Data Integration to Big Data Integration*. A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years: 43-59 (2018).

MOMIS
Mediator envirOnment for Multiple Information Sources

(Big) Data Integration with MOMIS

MOMIS is a (Big) Data Integration system able to aggregate data from heterogeneous (structured and semi-structured) and distributed sources (e.g., electronic health record, medical devices, etc.) in a semi-automatic way, exploiting the **semantic relationships** existing in the data sources (made available as open source by DataRiver).





Virtual Data Integration

MoMIS

www.datariver.it

SCHEMA ALIGNMENT

semi automatic

Global as View Mapping

Companies Mediated Schema

1. Attribute Matching
2. Companies Mediated Schema
3. Global as View mapping
4. Query

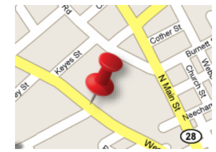
MOMIS

www.datariver.it

MOMIS

DATA FUSION

based on the same key

Name	Address	Sector	Revenue	Map
Software Inc.	Nimitz Fwy, Newark, US	Information Technology	€ 6.000 mln	
Fashion Inc.	Via Savona, Cuneo, IT	Textile	€ 930 mln	

VIRTUAL INTEGRATION

DATA CONFLICTS RESOLUTION

Data stored in Local sources

XML

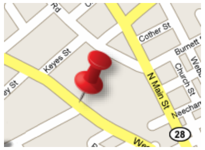
Name	Address	Sector	N° Emp.
Fashion Inc.	Via Savona, Cuneo, IT	Textile	8000
Software Inc.	Nimitz Fwy, Newark, US	Information Technology	600

Company	Location	Revenue
Software Inc.	Nimitz Fwy, Newark, US	€ 6.000 mln
Fashion Inc.	Via Libertà, Cuneo, IT	€ 930 mln

Name	Address	Latitude	Longitude
Software Inc.	Nimitz Fwy, Newark, US	37’44 N	122’13 W

ALWAYS UP TO DATE

MOMIS

Name	Address	Sector	Revenue	Map
Software Inc.	Nimitz Fwy, Newark, US	Information Technology	€ 6.000 mln	
Fashion Inc.	Via Savona, Cuneo, IT	Textile	€ 1.200 mln	

VIRTUAL INTEGRATION

Data stored in Local sources

XML

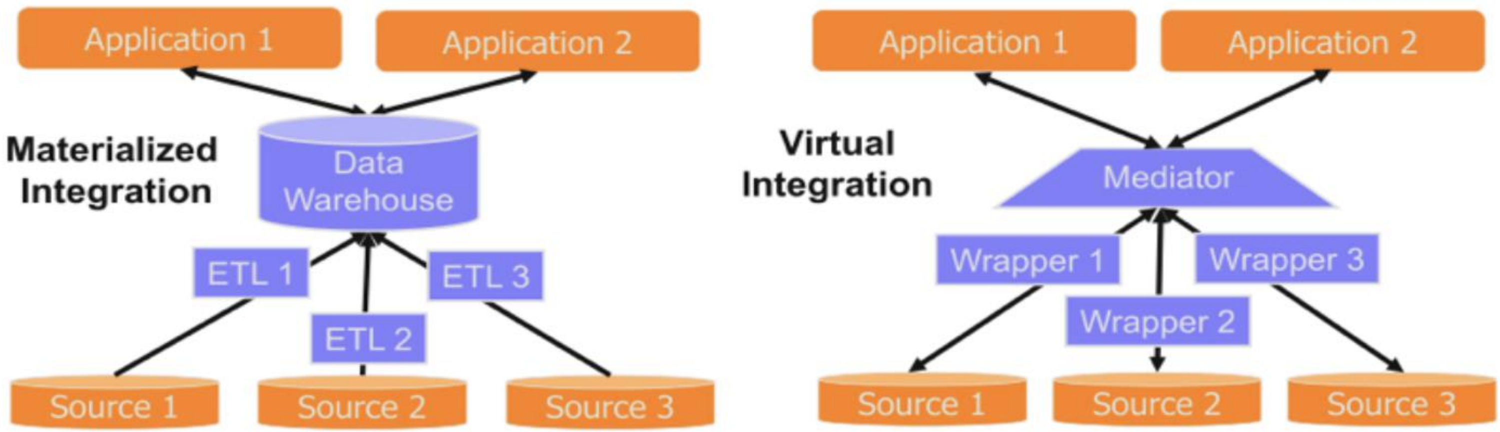
Name	Address	Sector	N° Emp.
Fashion Inc.	Via Savona, Cuneo, IT	Textile	8000
Software Inc.	Nimitz Fwy, Newark, US	Information Technology	600

Company	Location	Revenue
Software Inc.	Nimitz Fwy, Newark, US	€ 6.000 mln
Fashion Inc.	Via Libertà, Cuneo, IT	€ 1.200 mln

Name	Address	Latitude	Longitude
Software Inc.	Nimitz Fwy, Newark, US	37’44 N	122’13 W

Data Integration Architectures

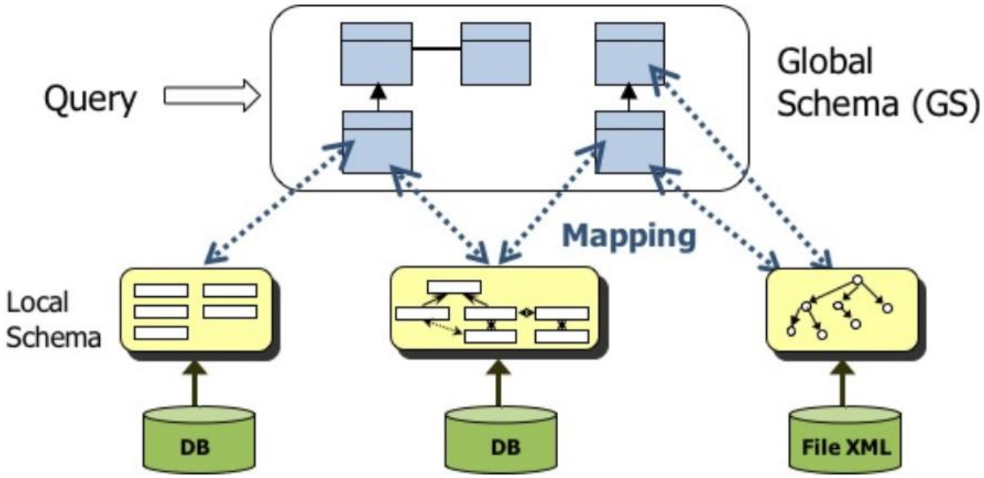
- **Materialized Integration**: integrate sources by bringing the data into a single physical database (**Data Warehouse**)
- **Virtual Integration**: leave the data at the sources and access it at query time via wrappers by supporting query over a mediated schema and by applying online query reformulation.



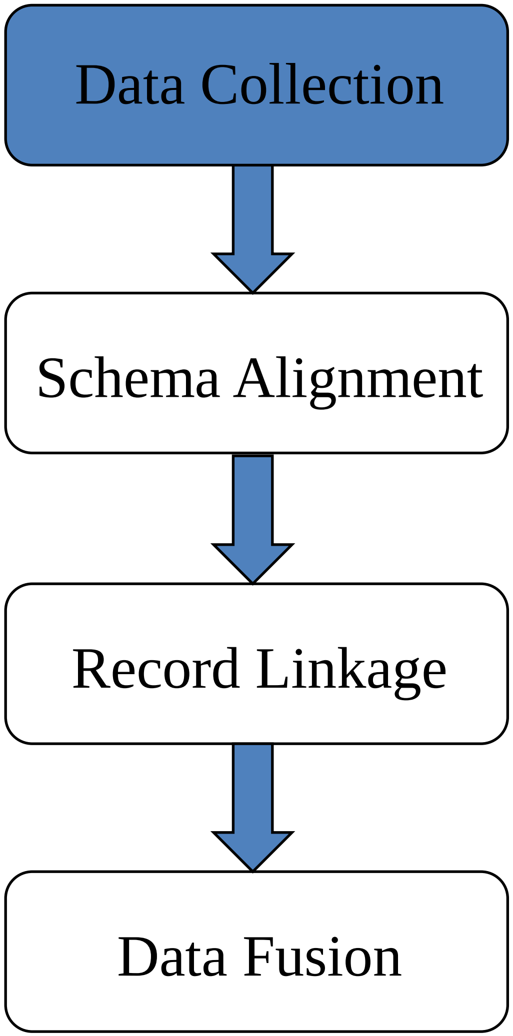
- Several intermediate architectures.

Virtual Data Integration: Mediators

- A mediator is a software module that exploits encoded knowledge about certain sets or subsets of data to create information for a higher layer of applications.
- The mediator builds a global schema of several (heterogeneous) information sources and allows a user to formulate a query on it.
- The user query is transformed in a set of sub-queries, one for each data source involved in the query.
- The results are collected by the mediator, merged and shown to the user.



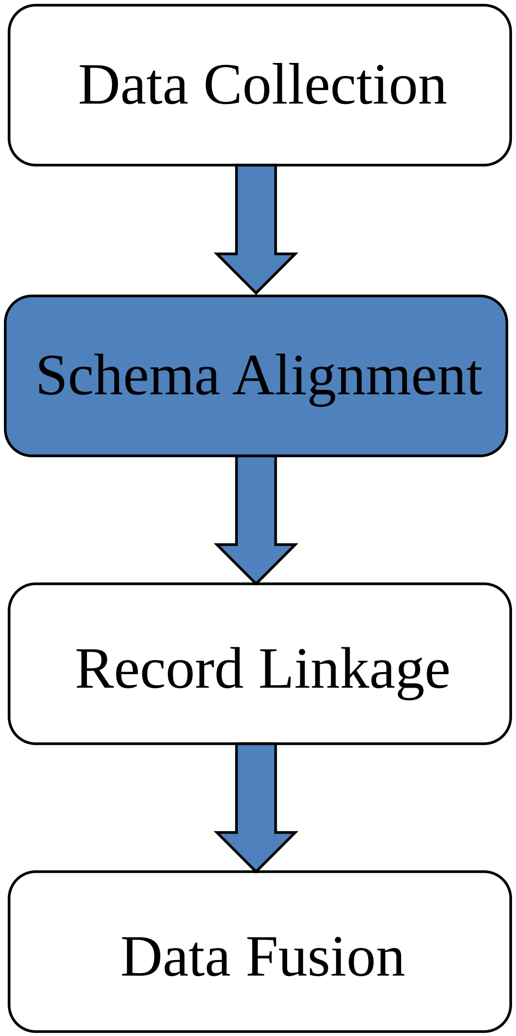
The Data Integration Process



- **Goal:** resolve technical, syntactical and data model heterogeneity so that data from all sources can be accessed/gathered and represented in the same data model.

- **Access heterogeneity** comprises all differences in the means to access data, not the data itself, e.g., Data Exchange Format (XML, JSON, CSV, …)
- **Syntactical heterogeneity** comprises all differences in the encoding of values, e.g., Character Format (ASCII, Unicode, …)
- **Data model heterogeneity** comprises differences in the data model that is used to represent data, e.g., Relational vs Object Data Model

The Data Integration Process

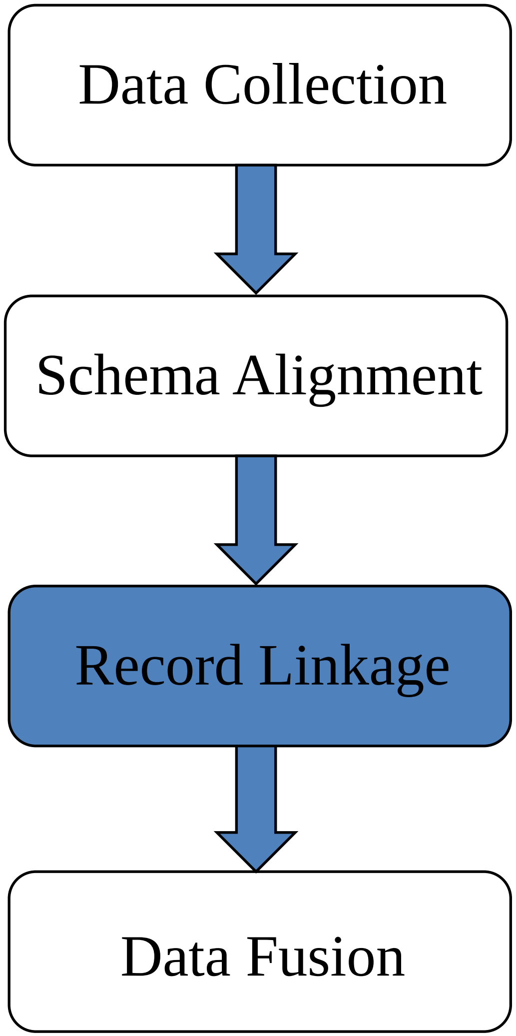


- **Goal:** resolve structural and schema-related semantic heterogeneity

- **Structural heterogeneity** comprises differences in the way different schemata represent the same part of reality,
  e.g., Alternative Modeling, Normalized vs. Denormalized

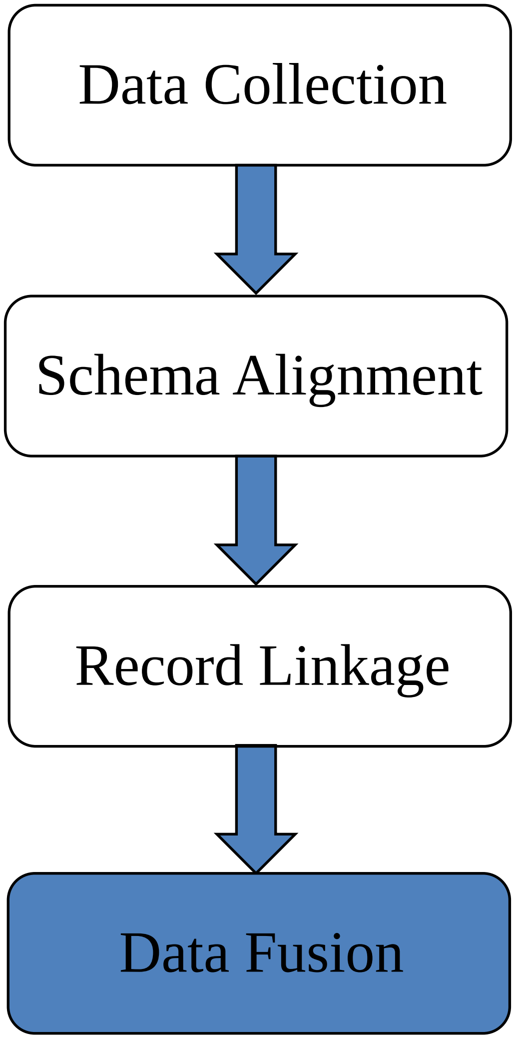
- **Semantic heterogeneity** comprises differences concerning the meaning of schema elements,
  e.g., Naming Conflicts (synonyms, homonyms, …)

The Data Integration Process



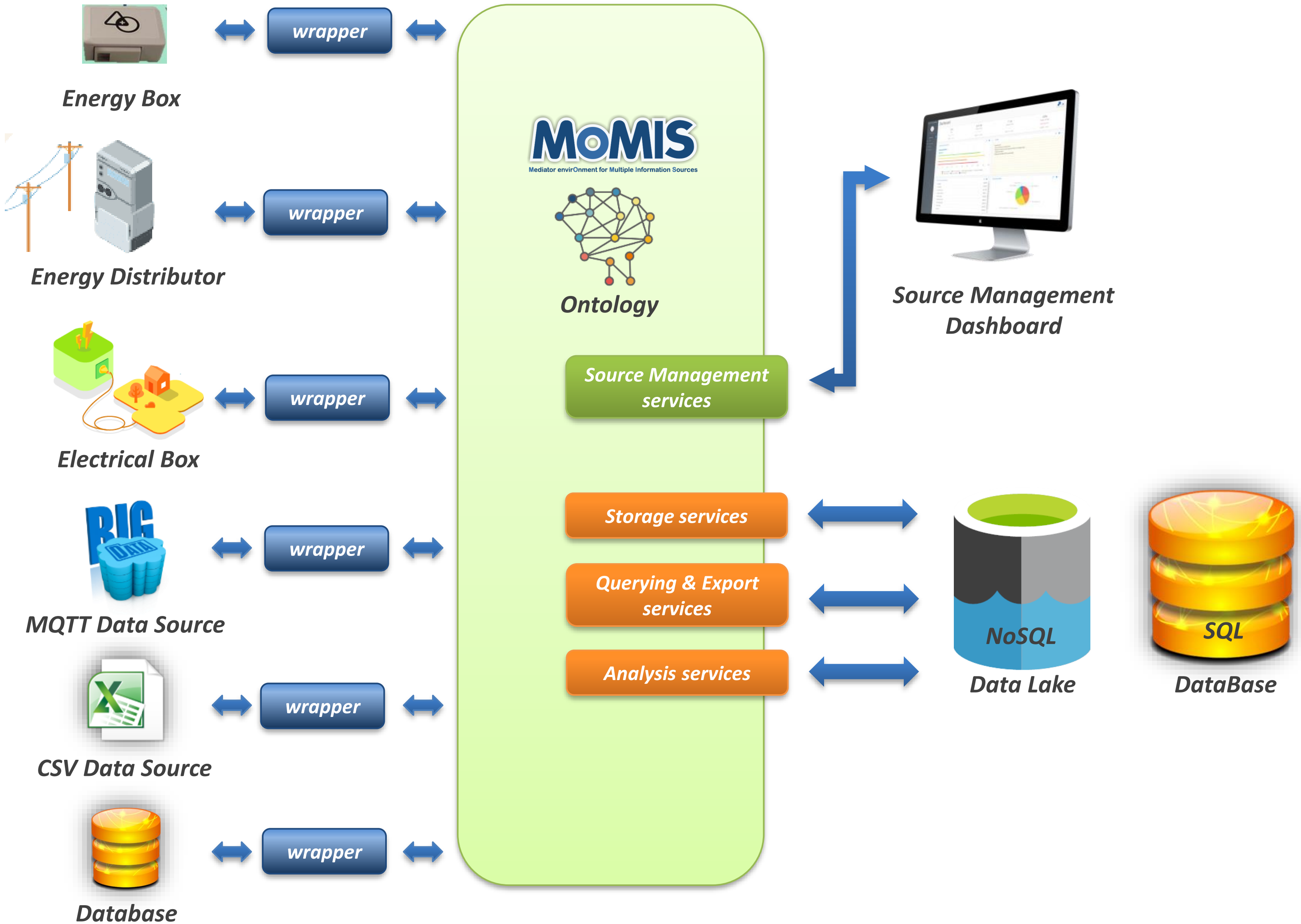
- **Goal:** resolve data related semantic heterogeneity by identifying all records in all data sources that describe the same real-world entity.
- Multiple data sources as well as multiple records within one data source may describe the same real-world entity.

The Data Integration Process



- **Goal:** resolve data conflicts by combining attribute values of duplicate records into a single consolidated description of an entity.

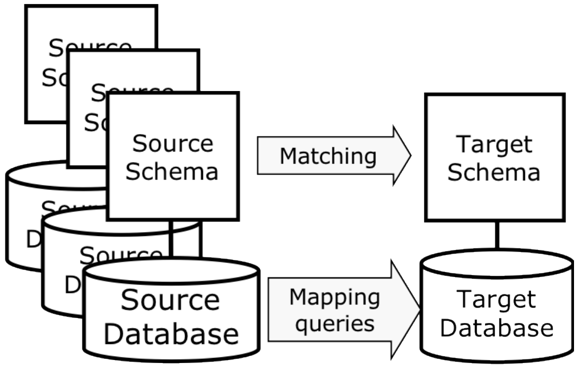
Data Collection in MOMIS: A Real-World Example



Data Integration Scenario

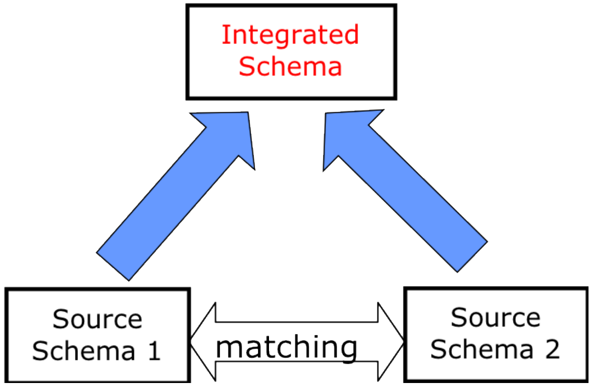
Top-down integration scenario

- **Goal:** Translate data from a set of source schemata into a given target schema.
- Triggered by concrete information need (= target schema)



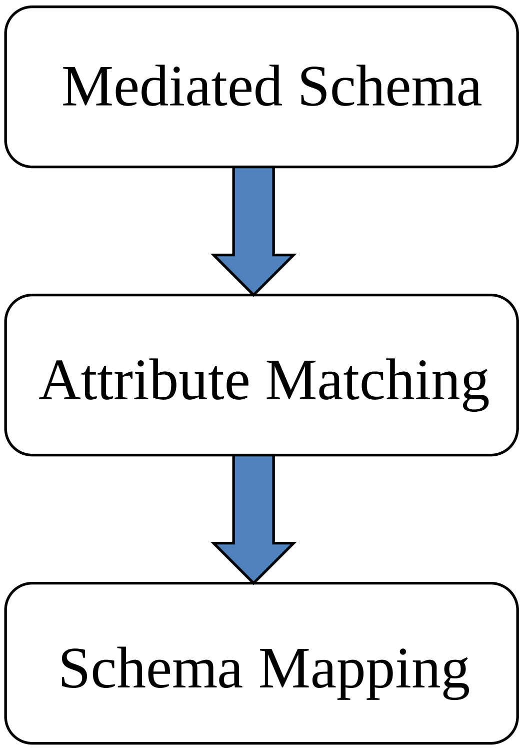
Bottom-up integration scenario

- **Goal**: Create a new integrated schema that can represent all data from a given set of source schemata (**Schema Integration**)
- Triggered by the goal to fulfill different information needs based on data from all sources.



Schema Alignment: Top-Down Integration

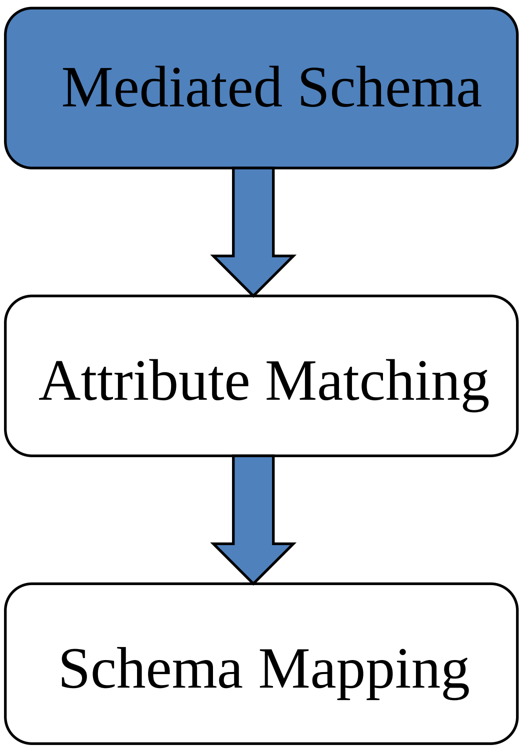
- **Top-down Integration Goal:** Translate data from a set of source schemata into a **given Mediated Schema.**
- Schema alignment: mediated schema + matching + mapping



S1	(Name, Location, Revenue, Phone number)
S2	(Name, Address, Sector, Income)
S3	(CompanyName, City, Address, Phone, Category)

Schema Alignment: Top-Down Integration

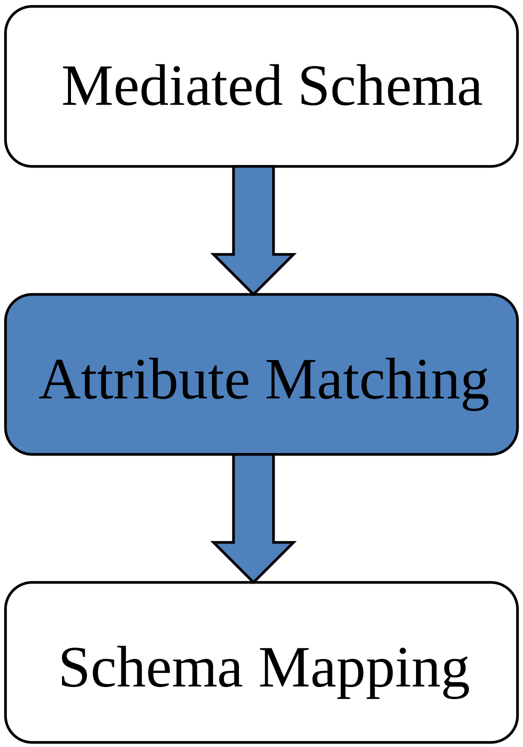
- Schema alignment: **mediated schema** + matching + mapping
  - Enables domain specific modeling



S1	(Name, Location, Revenue, Phone number)
S2	(Name, Address, Sector, Income)
S3	(CompanyName, City, Address, Phone, Category)
MS	**(Name, Address, Phone, Sector)**

Schema Alignment: Top-Down Integration

- Schema alignment: mediated schema + **matching** + mapping
  - Identifies correspondences between mediated and source schemata attributes

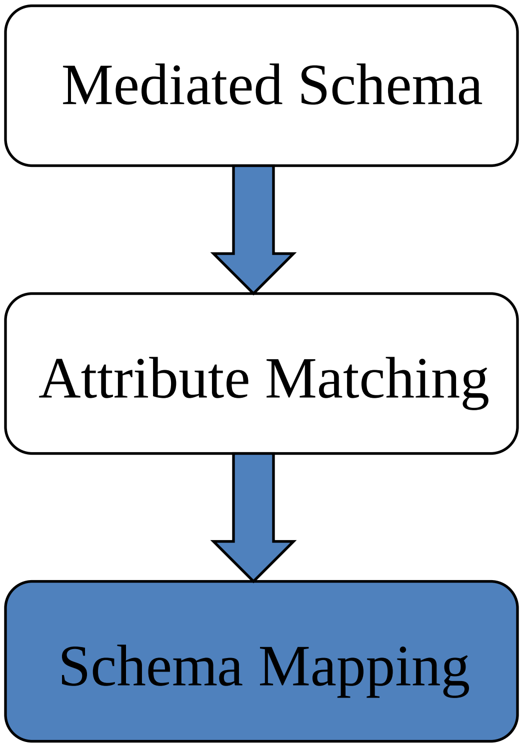


S1	(Name, Location, Revenue, Phone number)
S2	(Name, Address, Sector, Income)
S3	(CompanyName, City, Address, Phone, Category)
MS	**(Name, Address, Phone, Sector)**

MSAM	**MS.Name**: S1.Name, S2.Name, S3.CompanyName, … **MS.Address**: S1.Location, S2.Address, S3.City, S3.Address; **MS.Sector**: S2.Sector, S3.Category; **…**

Schema Alignment: Top-Down Integration

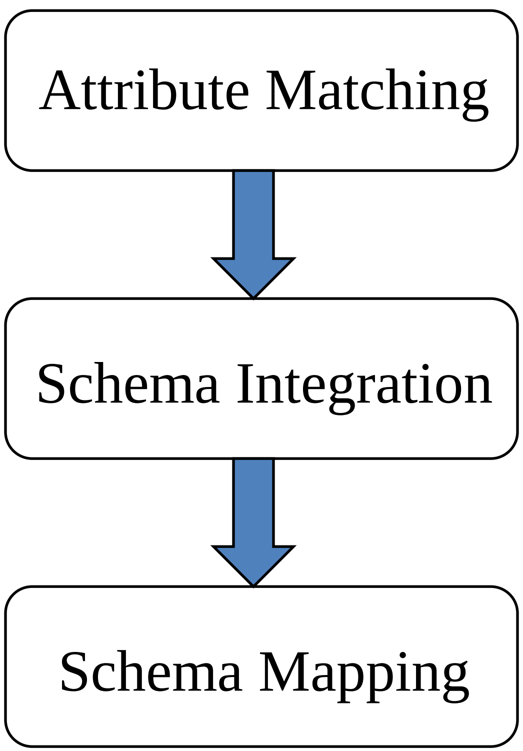
- Schema alignment: mediated schema + matching + **mapping**
  - Translate data from the set of source schemata into the mediated schema.



S1	(Name, Location, Revenue, Phone number)
S2	(Name, Address, Sector, Income)
S3	(CompanyName, City, Address, Phone, Category)
MS	**(Name, Address, Phone, Sector)**
MSAM	**MS.Name**: S1.Name, S2.Name, S3.CompanyName, … **MS.Address**: S1.Location, S2.Address, **S.Sector**: S2.Sector, S3.Category; **…**
MSSM (GAV)	(Name, Address, Phone, _):- S1(Name, Address, Phone) (Name, Address, _, Sector):- S2(Name, Address, Sector)

Schema Alignment: Bottom-Up Integration

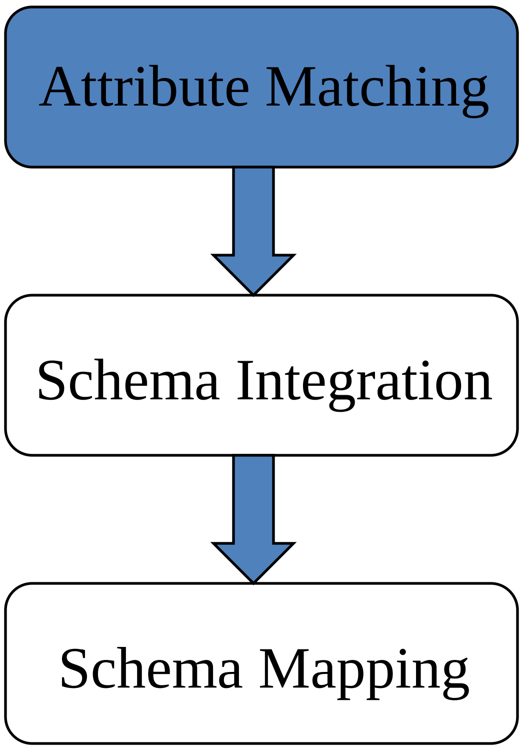
- **Bottom-up Integration Goal:** Create a new integrated schema that can represent all data from a given set of source schemata**.**
- Schema alignment: matching + schema integration + mapping



S1	(Name, Location, Revenue, Phone number)
S2	(Name, Address, Sector, Income)
S3	(CompanyName, City, Address, Phone, Category)

Schema Alignment: Bottom-Up Integration

- Schema alignment: **matching** + schema integration + mapping
  - Identifies correspondences among source schemata attributes



S1	(Name, Location, Revenue, Phone number)
S2	(Name, Address, Sector, Income)
S3	(CompanyName, City, Address, Phone, Category)

AM	S1.Name, S2.Name S2.Name, S3CompanyName … S1.Location,S2.Address S2.Address,S3.Address S2.Address,S3.City

MOMIS Attribute Matching: Common Thesaurus

Common Thesaurus : the set of correspondences between local attributes (*Attribute Matches*)

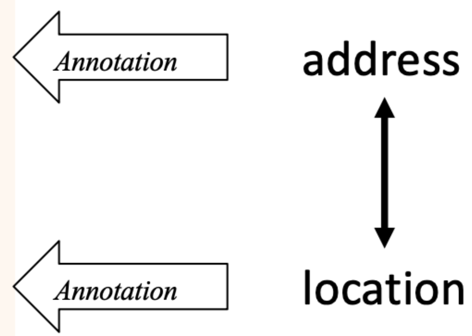
MOMIS uses a combination of semi-automatic methods:

- **Lexicon-derived** correspondences, derived by the annotation of local schemata with respect to a lexical resource, such as WordNet or other semantic resource
- **Schema-derived** correspondences
  - For example, correspondences derived from foreign keys in a relational schema
- **Inferred** correspondences**,** derived by exploiting Description Logics techniques
- **Designer supplied** correspondences
  - The designer can add/delete relationships to the Common Thesaurus

Semantic Enrichment: MOMIS Lexicon-derived Correspondences

- Lexical Annotation w.r.t. a Semantic Resource such as WordNet
  - WordNet (https://wordnet.princeton.edu) groups words into sets of synonyms (synsets), provides short definitions/examples, and records relations (Hyponymy, Hypernymy, …) among these synonym sets.

- S: (n) **address**, computer address, reference ((computer science) the code that identifies where a piece of information is stored)
- S: (n) **address** (the place where a person or organization can be found or communicated with)
  - *direct hyponym* / *full hyponym*
  - *direct hypernym* / ***inherited hypernym*** / *sister term*
    - S: (n) geographic point, geographical point (a point on the surface of the Earth)
      - S: (n) point (the precise location of something; a spatially limited location) *"she walked to a point where she could survey the whole street"*
        - S: (n) location (a point or extent in space)
          - S: (n) object, physical object (a tangible and visible entity; an entity that can cast a shadow) *"it was full of rackets, balls and other objects"*
            - S: (n) physical entity (an entity that has physical existence)
              - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))
  - *derivationally related form*
- S: (n) **address**, speech (the act of delivering a formal spoken communication to an audience) *"he listened to an address on minor Roman poets"*
- S: (n) **address** (the manner of speaking to another individual) *"he failed in*

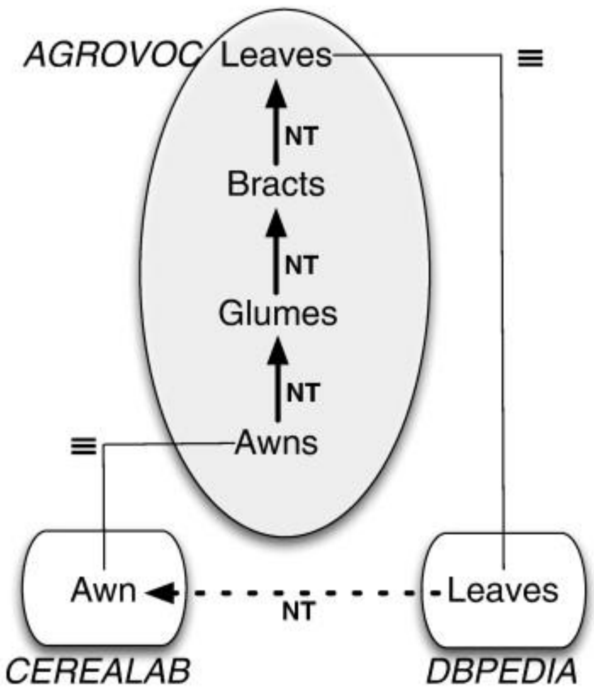


MOMIS Lexicon-derived Correspondences

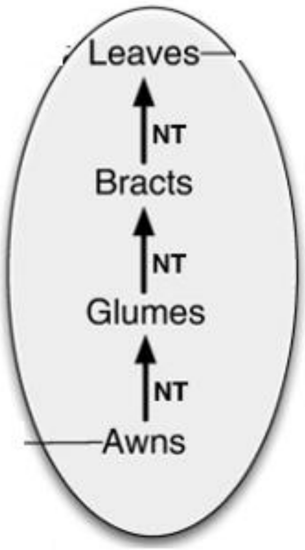
- Lexical Annotation with respect to a domain thesaurus.

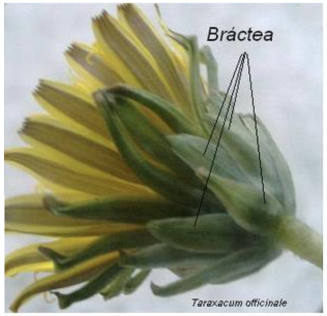
- Two local sources
  1. The CEREALAB Database
  2. The DBPEDIA Dataset

- Annotation w.r.t. AGROVOC, a thesaurus covering all areas of interest of the FAO (https://agrovoc.fao.org/browse/agrovoc/en/)



https://agrovoc.fao.org/browse/agrovoc/en/page/c_9023





Bract

A modified leaf or leaflike part just below and protecting an inflorescence

Brattea

Una foglia modificata o una parte simile a una foglia appena sotto di un'infiorescenza.

Glume

Small dry membranous bract found in inflorescences of Gramineae

Gluma

Piccola brattea membranosa secca che si trova nelle infiorescenze delle Graminacee

Awn

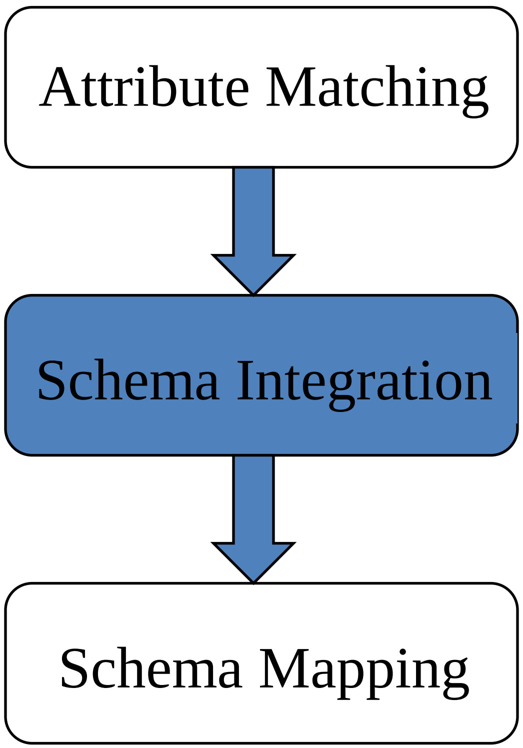
Slender bristlelike appendage found on the bracts of grasses

Arista

Appendice sottile e setolosa che si trova sulle brattee delle erbe.

Schema Alignment: Bottom-Up Integration

- Schema alignment: attribute matching + **schema integration** + mapping



- Attribute Matches of the Common Thesaurus are used to evaluate the ***affinity between local classes***
- Local classes with a given level of affinity are grouped together in ***clusters*** using a ***hierarchical clustering technique***
- For each cluster, a ***Global Class*** that represents the mediated view of all the local classes of the cluster is created

Cluster Generation: Example 1

In the following table, local attributes on the same row are matches

S1.Company	S2.Enterprise	S3.Company
Name	Name	CompanyName
Location	Address	Address, City
Phone Number		Phone, City
	Sector	Category
	Income	
Revenue		

There is no match between Revenue and Income

- S1.Company and the other two local classes do not have a sufficient affinity, thus we obtain two clusters:

{ S2.Enterprise, S3.Company}
{ S1.Company}

⇨ Two global classes

Cluster Generation: Example 2

In the following table, local attributes on the same row are matches

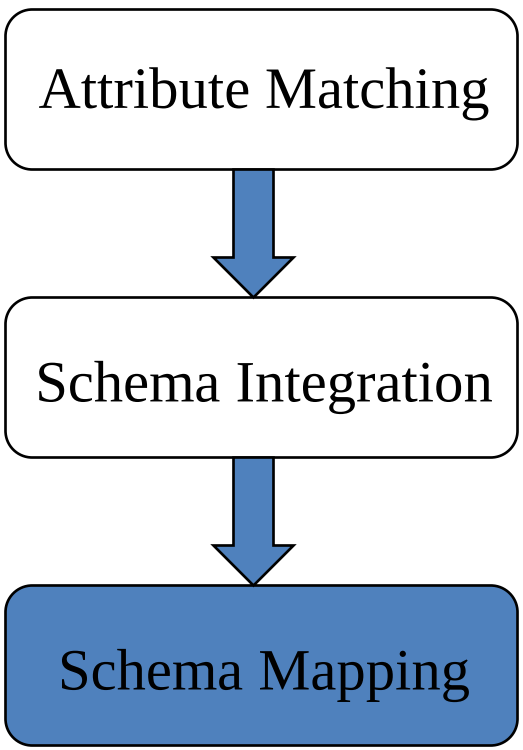
S1.Company	S2.Enterprise	S3.Company
Name	Name	CompanyName
Location	Address	Address, City
Phone Number		Phone, City
	Sector	Category
Revenue	Income	

There is a match between Revenue and Income after annotation

- Now, between S1.Company and the other two local classes there is a sufficient level of affinity, thus all the three local classes are grouped together;
  One cluster
  {S1.Company, S2.Enterprise, S3.Company} ⇨ One global class

Schema Alignment: Bottom-Up Integration

- Schema alignment: mediated schema + matching + **mapping**
  - Specifies the transformations between records in different schemas



- A Mapping Table represents the correspondences between a Global Class and its local classes ➔ *intensional* level
- How to get Global Class instances from local classes ➔ *extensional* level
- **Global as View** approach: each Global Class is defined as a view over its Local Classes

Mapping Table Generation

- For a global class G a Mapping Table MT is automatically generated, whose columns represent the local classes belonging to G and whose rows represent the global attributes of G. An element MT[GA][L] represents the set of local attributes of L which are mapped onto the global attribute GA.

	S1.Company	S2.Enterprise	S3.Company
Name	Name	Name	CompanyName
Address	Location	Address	Address, City
Phone	Phone Number		Phone, City
Sector		Sector	Category
Revenue	Revenue	Income	

- For each element MT[GA][L] a *Data Transformation Functions* can be specified to transform the local values into the global value.
  - *MT[Phone][S3.Company] = {Phone,City}:* companies from different countries, the country prefix (obtained from City) must be added to Phone.

MOMIS Schema Alignment

S1.Company			
Name	Location	Revenue	Phone Number
IBM Corp	New York	131	469805361
Apple Inc	Cupertino, CA	158	777805361
GE	Boston, MA	77	

S2.Enterprise			
Name	Address	Sector	Income
IBM	NY	IT	140
Apple	CA	IT	160
Electric Co	MD	Electric	3

S3.Company				
Company Name	City	Address	Phone	Category
General Electric	Boston	Farnsworth Str.	443-3000	Electric
IBM Corporation	New York		980-5350	Information

Schema matching: Mapping Table of G

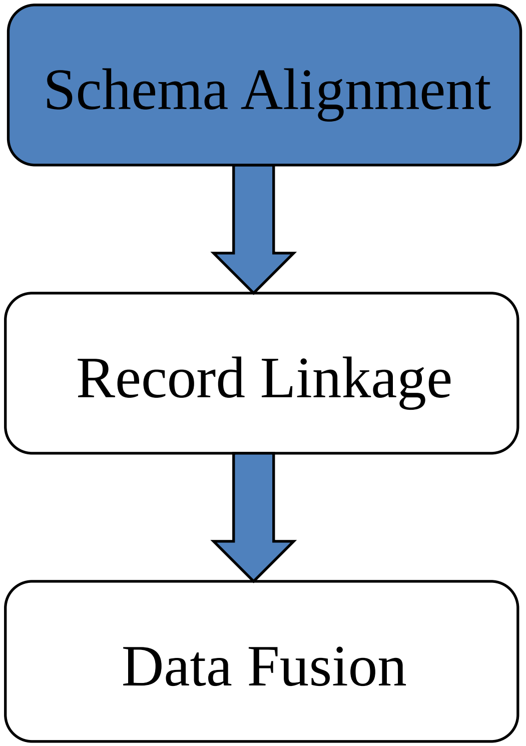
	S1.Company	S2.Enterprise	S3.Company
Name	Name	Name	CompanyName
Address	Location	Address	Address, City
Phone	Phone Number		Phone, City
Sector		Sector	Category
Revenue	Revenue	Income	

Global Instance

	Name	Address	Phone	Sector	Revenue
S1	IBM Corp	New York	469805361		131
	Apple Inc	Cupertino, CA	777805361		158
	GE	Boston, MA			77
S2	IBM	NY		IT	140
	Apple	CA		IT	160
	Electric Co	MD		Electric	3
S3	General Electric	Boston, Farnsworth Str.	56-443-3000	Electric	
	IBM Corporation	New York	77-980-5350	Information	

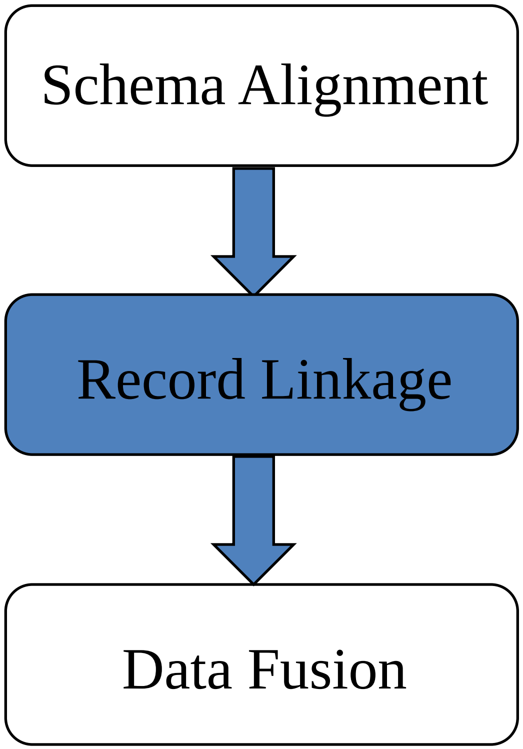
- The Mapping Table with the correspondences between the *global class G* and its local classes *{S1.Company, S2.Enterprise, S3.Company}* ➔ *intensional* level
- **Global as View** approach: The instances of the global class G are defined by a view over the local class instances ➔ *extensional* level

The Data Integration Process



	Name	Address	Phone	Sector	Revenue
S1	IBM Corp	New York	469805361		131
	Apple Inc	Cupertino, CA	777805361		158
	GE	Boston, MA			77
S2	IBM	NY		IT	140
	Apple	CA		IT	160
	Electric Co	MD		Electric	3
S3	General Electric	Boston, Farnsworth Str.	56-443-3000	Electric	
	IBM Corporation	New York	77-980-5350	Information	

The Data Integration Process



	Name	Address	Phone	Sector	Revenue	
	IBM Corp	New York	469805361		131	E1
S1	Apple Inc	Cupertino, CA	777805361		158	
	GE	Boston, MA			77	
	IBM	NY		IT	140	E1
S2	Apple	CA		IT	160	
	Electric Co	MD		Electric	3	
S3	General Electric	Boston, Farnsworth Str.	56-443-3000	Electric		
	IBM Corporation	New York	77-980-5350	Information		E1

- ***Simple*** case: we can define one or more attributes, called ***Join Attributes*** to identify the same entity
- ***Complex*** (real) case: no join attributes → ***Entity Resolution / Record Linkage***

Simple Record Linkage: Join Attributes

Join Attribute

	S1.Company	S2.Enterprise	S3.Company
Name	Name	Name	CompanyName
Address	Location	Address	Address, City
Phone	Phone Number		Phone, City
Sector		Sector	Category
Revenue	Revenue	Income	

Two records of *Si* and *Sj* represent the same real entity if and only if they satisfy the ***join condition***

Si.Name=Sj.Name

	Name	Address	Phone	Sector	Revenue
S1	IBM Corporation	New York	469805361		131
	Apple Inc	Cupertino, CA	777805361		158
	General Electric	Boston, MA			77
S2	IBM Corporation	NY		IT	140
	Apple Inc	CA		IT	160
	Electric Co	MD		Electric	3
S3	General Electric	Boston, Farnsworth Str.	56-443-3000	Electric	
	IBM Corporation	New York	77-980-5350	Information	

	Name	Address	Phone	Sector	Revenue
E1	IBM Corporation	New York	469805361		131
	IBM Corporation	NY		IT	140
	IBM Corporation	New York	77-980-5350	Information	
E2	Apple Inc	Cupertino, CA	777805361		158
	Apple Inc	CA		IT	160
E3	General Electric	Boston, MA			77
	General Electric	Boston, Farnsworth Str.	56-443-3000	Electric	
E4	Electric Co	MD		Electric	3

The operation to be performed is a ***Full Outer Join***

- to join records on the basis of join conditions
- to include into the result all the records of all local sources

The Data Integration Process

Schema Alignment → Record Linkage → **Data Fusion**

	Name	Address	Phone	Sector	Revenue	
S1	IBM Corp	New York	469805361		131	E1
	Apple Inc	Cupertino, CA	777805361		158	
	GE	Boston, MA			77	
S2	IBM	NY		IT	140	E1
	Apple	CA		IT	160	
	Electric Co	MD		Electric	3	
S3	General Electric	Boston, Farnsworth Str.	56-443-3000	Electric		
	IBM Corporation	New York	77-980-5350	Information		E1

	Name	Address	Phone	Sector	Revenue
E1	IBM Corp	New York	469805361		131
	IBM	NY		IT	140
	IBM Corporation	New York	77-980-5350	Information	
E2	Apple Inc	Cupertino, CA	777805361		158
	Apple	CA		IT	160
E3	GE	Boston, MA			77
	General Electric	Boston, Farnsworth Str.	56-443-3000	Electric	
E4	Electric Co	MD		Electric	3

Resolution Functions

longest → Name; *voting* → Address; *prefer S3* → Phone; *prefer S2* → Sector; *avg* → Revenue

	Name	Address	Phone	Sector	Revenue
E1	IBM Corporation	New York	77-980-5350	IT	135,5
E2	Apple Inc	Cupertino, CA	777805361	IT	159
E3	General Electric	Boston, Farnsworth Str.	56-443-3000	Electric	77
E4	Electric Co	MD		Electric	3

To Summarize...

Local Classes

S1.Company			
Name	**Location**	**Revenue**	**Phone Number**
IBM Corp	New York	131	469805361
Apple Inc	Cupertino, CA	158	777805361
GE	Boston, MA	77	

S2.Enterprise			
Name	**Address**	**Sector**	**Income**
IBM	NY	IT	140
Apple	CA	IT	160
Electric Co	MD	Electric	3

S3.Company				
Company Name	**City**	**Address**	**Phone**	**Category**
General Electric	Boston	Farnsworth Str.	443-3000	Electric
IBM Corporation	New York		980-5350	Information

Global Class Company - Mapping Table

	S1.Company	S2.Enterprise	S3.Company
Name	Name	Name	CompanyName
Address	Location	Address	Address, City
Phone	Phone Number		Phone, City
Sector		Sector	Category
Revenue	Revenue	Income	

Global Class Company – Instance (*virtual*)

Name	Address	Phone	Sector	Revenue
IBM Corporation	New York	77-980-5350	IT	135,5
Apple Inc	Cupertino, CA	777805361	IT	159
General Electric	Boston, Farnsworth Str.	56-443-3000	Electric	77
Electric Co	MD		Electric	3

Global Query - To query the integrated data

Example: name and revenue for companies with address "New York" and sector "IT"

The Querying Problem

- How to answer global queries?

```
SELECT  Name, Revenue
FROM    Company
WHERE   Address = "New York" and Sector = "IT"
```

- In a Virtual Data Integration system, data reside at the data sources then the query processing is based on Query rewriting: a global query has to be expressed as an equivalent set of queries on the local data sources (local queries).

- **Global as View** approach:
  - Instances of a global class G are defined by a view over its local class instances
  - rewriting is performed by **unfolding**, i.e., by expanding a global query on G according to the view associated to G

➢ In the following an intuitive example of query unfolding.

Query Unfolding: A simple example

Mapping table of the global class **Company** (with only two local classes)

	S1.Company	S2.Enterprise
Name	Name	Name
Address	Location	Address
Phone	Phone Number	
Sector		Sector
Revenue	Revenue	Income

Global query

```
SELECT  Name, Revenue
FROM Company
WHERE Phone like "77*" and Sector = "IT"
```

Local queries

```
SELECT  Name, Revenue FROM S1.Company
WHERE   Phone Number like "77*"
```

Name	Revenue
Apple Inc	158
IBM Corporation	131

```
SELECT  Name, Income FROM S2.Enterprise
WHERE   Sector = "IT"
```

Name	Income
IBM Corporation	140
Apple Inc	160

Local queries results are

1) Transformed by using the Mapping Table to obtain the global attributes
2) Joined by using the join attribute *Name*
3) Fused by using the Resolution Functions (average)

Name	Income
IBM Corporation	135,5
Apple Inc	159

Outline

- Who I Am
- From Data Integration to Big Data Integration
- **Entity Resolution (a.k.a. Record Linkage)**
  - **Entity Resolution**
  - **Entity Resolution Pipeline**
    - Blocking
    - Block Cleaning
    - Entity Matching
    - Entity Clustering
    - Data Fusion
    - Beyond Traditional Batch ER
- Privacy-Preserving Record Linkage (PPRL)
- PPRL in E-Eath domain

Entity Resolution (ER)

Data Discovery
- Source collection
- Source analysis
- Data patterns

Data Validation
- Correctness
- Completeness
- Data Quality constraints

Data Cleaning
- Missing Values
- **Deduplication**
- Data Transformation

Semantic Enrichment
- Semantic Annotation
- Standards vocabularies
- Ontologies

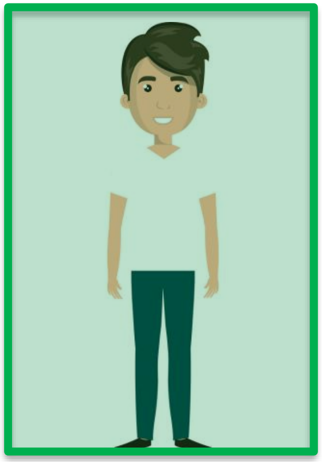
Data Integration
- Schema Generation
- Schema Mapping

Deduplication
a.k.a.
Entity Resolution
Data Matching
Record Linkage

Entity Resolution (ER)

Given one or more data sources, Entity Resolution (ER) is the task of identifying the **records (entity profiles)** that refer to the **same real-world object (entity)**.

We will refer to *entity profiles* simply as **profiles**.



Data Source A

	Name	Surname	Address	Sex
r1	Mary-Ann	White	West Main Street 29, 12068, Fonda, NY, New York	F
r2	Thomas J.	Franklin	50 Liverpool Street, London	M

Data Source B

Name	Residence	Age	Gender	
Franklin, Tom	London (UK)	25	Male	r3
Withe, Mary Ann	New York (USA)	29	Female	r4

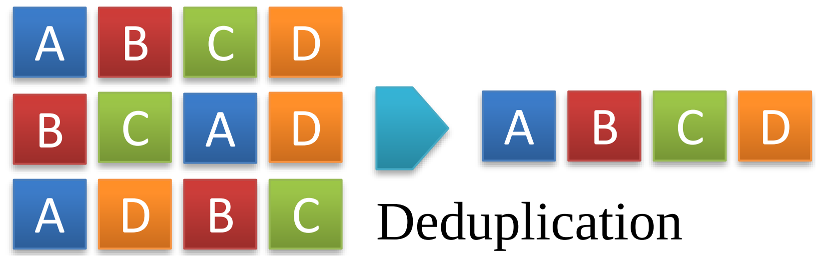
Entity Resolution (ER)

ER is a hard task, since data can be **dirty and ambiguous**:
- some words can be **written in different ways** (or even **misspelled**);
- cases of **homonymy** and **synonymy**;
- **missing** or wrong values.

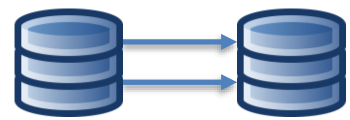
Name	Surname	Date of birth	Address
Richard	**Wright**	08/06/1996	Main Street, 12
Anne Marie	Thompson	04-09-1998	St James Blvd 4
Richard	**Wright**	Dec 15, 1968	Hill Street 98
Rick	**Wirgth**	Aug 6, 1996	Main St. 12
Nick	Mason	**NULL**	NULL
Anne-Marie	Thompson	April 9, 1998	Saint James Boulevard, 12

ER: Why is it so important?

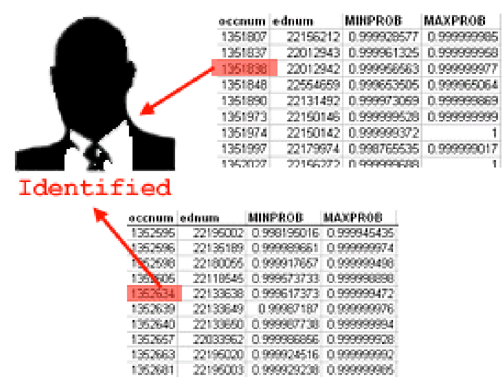
- Data Integration
- Deduplication
- Record Linkage
- Fraud Detection
- Catalogs Fusion
- Reducing the size of stored data
- …



Deduplication



Record Linkage



Fraud Detection

The «Father» of Entity Resolution

Halbert L. Dunn, M.D. (1896-1975), the leading figure in establishing a **national vital statistics** system in the United States.
Vital statistics: births, deaths, migration, marriages, divorces, etc.

- Medical doctor and statistician;
- Chief of the National Office of Vital Statistics from 1935 through 1960;
- Co-founder of the National Association for Public Health Statistics and Information Systems (NAPHSIS) and the Inter-American Statistics Institute (IASI).

H. Dunn: *Record Linkage*. American Journal of Public Health (AJPH) 36(12): 1412-1416 (1946)

Main focus on **death clearances**.

Record Linkage (1946)

EACH person in the world creates a Book of Life. This Book starts with birth and ends with death. Its pages are made up of the records of the principal events in life. Record linkage is the name given to the process of assembling the pages of this Book into a volume.

The Book has many pages for some and is but a few pages in length for others. In the case of a stillbirth, the entire volume is but a single page.

The person retains the same identity throughout the Book. Except for advancing age, he is the same person. Thinking backward he can remember the important pages of his Book even though he may have forgotten some of the words. To other persons, however, his identity must be proven. "Is the John Doe who enlists today in fact the same John Doe who was born eighteen years ago?"

Events of importance worth recording in the Book of Life are frequently put on record in different places since the person moves about the world throughout his lifetime. This makes it difficult to assemble this Book into a single compact volume. Yet, sometimes it is necessary to examine all of an individual's important records simultaneously. No one would read a novel, the pages of which were not assembled. Just so, it is necessary at times to link the various important records of a person's life.

The two most important pages in the Book of Life are the first one and the last one. Consequently, in the process of record linkage the uniting of the fact-of-death with the fact-of-birth has been given a special name, "death clearance."

ER Formalization (1969)

Ivan P. Fellegi (1935), a Hungarian-Canadian statistician, Chief Statistician of Canada from 1985 to 2008.

I. Fellegi, A. Sunter: *A Theory for Record Linkage*. Journal of the American Statistical Association (JASA), 64(328), 1183-1210 (1969)

A mathematical model is developed to provide a theoretical framework for a computer-oriented solution to the problem of recognizing those records in two files which represent identical persons, objects or events (said to be *matched*).

A comparison is to be made between the recorded characteristics and values in two records (one from each file) and a decision made as to whether or not the members of the comparison-pair represent the same person or event, or whether there is insufficient evidence to justify either of these decisions at stipulated levels of error. These three decisions are referred to as *link* (A_1), a *non-link* (A_3), and a *possible link* (A_2). The first two decisions are called positive dispositions.

The two types of error are defined as the error of the decision A_1 when the members of the comparison pair are in fact unmatched, and the error of the decision A_3 when the members of the comparison pair are, in fact matched. The probabilities of these errors are defined as

$$\mu = \sum_{\gamma \epsilon \Gamma} u(\gamma) P(A_1 \mid \gamma)$$

and

$$\lambda = \sum_{\gamma \epsilon \Gamma} m(\gamma) P(A_3 \mid \gamma)$$

respectively where $u(\gamma)$, $m(\gamma)$ are the probabilities of realizing γ (a comparison vector whose components are the coded agreements and disagreements on each characteristic) for unmatched and matched record pairs respectively. The summation is over the whole comparison space Γ of possible realizations.

A *linkage rule* assigns probabilities $P(A_1|\gamma)$, and $P(A_2|\gamma)$, and $P(A_3|\gamma)$ to each possible realization of $\gamma \epsilon \Gamma$. An optimal linkage rule $L(\mu, \lambda, \Gamma)$ is defined for each value of (μ, λ) as the rule that minimizes $P(A_2)$ at those error levels. In other words, for fixed levels of error, the rule minimizes the probability of failing to make positive dispositions.

A theorem describing the construction and properties of the optimal linkage rule and two corollaries to the theorem which make it a practical working tool are given.

A Real Use Case Scenario

- A logistic company that works with medical supplies acquires new customers.
- These customers already own their product catalogs with thousands of products.
- In each catalog the same product has a different identifier, and a similar but not equal description.
- The logistic company wants to unify the catalogs (i.e., giving a unique id to the same product) in order to better organize its warehouse.

Catalog A

PID	Title	Description
P123X	Syringe 10x10	Syringe 10 ml – 10 pack
P123Y	Syringe 10x100	Syringe 10 ml – 100 pack
P456A	Insulin needle 4x10	Hypodermic insulin needle 4 mm – 10 pack
…	…	…

Catalog B

ID	Name	Description
1	Syringes 10 ml	Syringe 10 ml x 10 pieces
2	Syringes 10 ml big pack	Syringe 10 ml x 100 pieces
3	Small needle	Needle for insulin 4 mm 10 pieces
…	…	…

A Real Use Case Scenario

Solutions

- Merge the catalogs manually:
  - High effort;
  - Requests a lot of time;
  - High error risk.
- Use a deduplication tool exploiting the products description:
  - Faster;
  - Accurate;
  - Combine man work with the tool.

Master Data Management

- **[Master Data Management (MDM)]** is a business-led program for ensuring that the organization's shared data (**master data**) is consistent and accurate. MDM programs include the people, processes, and systems used to keep master data accurate and consistent.

- MDM creates a **single master record** (a.k.a. **golden record**) which serves as a **trusted view of business-critical data** (a customer, location, product, supplier, etc.) upon which a business or organization relies. Master data can be managed and shared across the business to promote accurate reporting, reduce data errors, remove redundancy, and help workers make better-informed business decisions.

Master Data Management

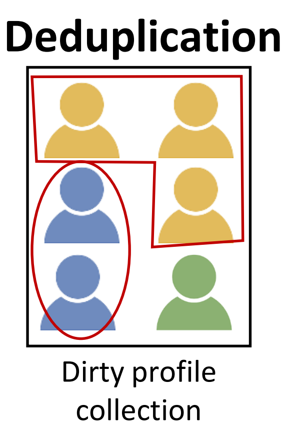
- To create master data, information coming from across internal (e.g., silos) and external data sources and applications has been **deduplicated, reconciled and enriched**, becoming a consistent and reliable source.
- As a **discipline**, MDM relies on the principles of data governance, with the goal of creating a trusted and authoritative view of a company's data.
- As a **technology**, MDM solutions automate how business-critical data is governed, managed, and shared throughout applications used by lines of business, brands, departments, and organizations. MDM applies data integration, reconciliation, enrichment, quality, and governance to create master records.

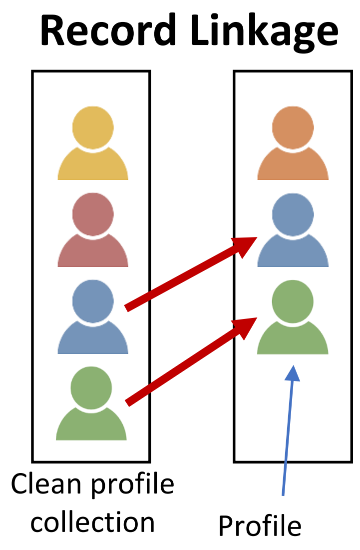
Types of Entity Resolution

- The input of ER consists of profile collections that can be of two types:
  - **Clean**: each collection is duplicate-free;
  - **Dirty**: each collection contains duplicates.

- Based on the input, we distinguish ER into 3 sub-tasks:
  - Clean-Clean ER (a.k.a. **Record Linkage**)
  - Dirty-Clean ER
  - Dirty-Dirty ER

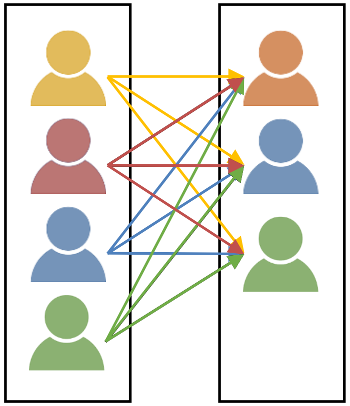
  a.k.a. **Deduplication**





Entity Resolution Complexity

- ER is an inherently quadratic problem $O(n^2)$: every profile must be compared with each other.
- ER does not scale well to large profile collections (e.g., Big Data).



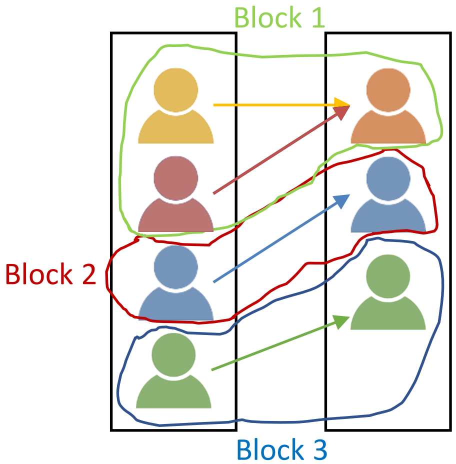
12 comparisons! ***(n x m)*** **in clean-clean scenario or** ***n x (n-1)*** **in a dirty scenario**

➡ **It is not feasible to compare all possible pairs of profiles!**

Entity Resolution Complexity: A Possible Solution

Blocking is needed in order to reduce the number of comparisons:

- Group similar profiles into blocks by defining for every profile a blocking key or a set of blocking keys (profiles with the same keys are placed in the same blocks);
- Execute comparisons only inside each block.
  - Complexity now is quadratic to the size of the block (much smaller than dataset size!)



4 comparisons!

But how to define the blocking keys?
This is a very difficult and error-prone task!

Entity Resolution Complexity: A Possible Solution

But how to define the blocking keys?
This is a very difficult and error-prone task!

Dataset 1

ID	Name	Surname	Location
1	Thomas	Jones	Hills street
2	Richie	William	Main street 9

Dataset 2

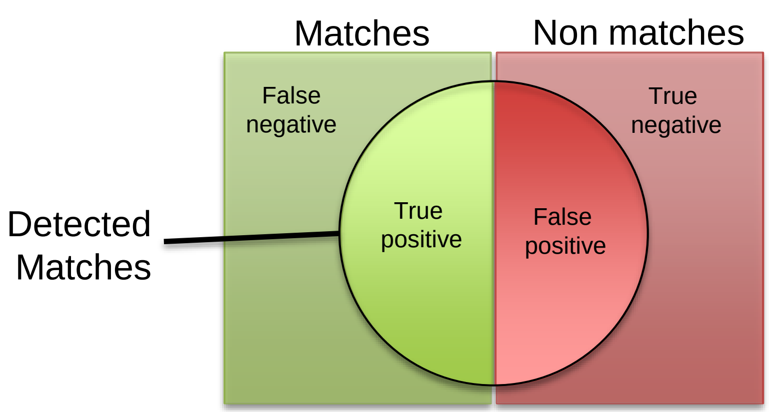
ID	Nominative	Address
3	Tom Jones	Hill St
4	Rick Williams	Main street 9

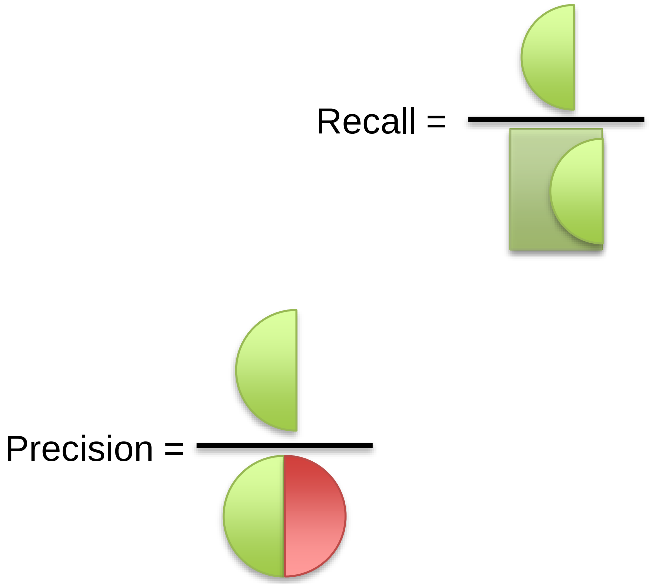
Blocking Key	Records
Street	1, 2, 4
Main	2, 4
9	2, 4
Jones	1, 3

Measuring the Blocking Performance

Recall: how many duplicates are selected over the existing ones?

Precision: how many selected pairs are true duplicates?

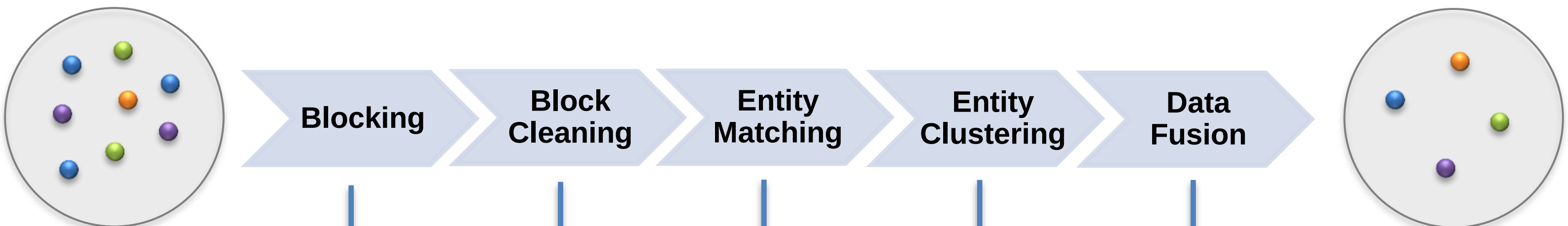




The "Standard" ER Approach

Dirty Data (duplicate records)

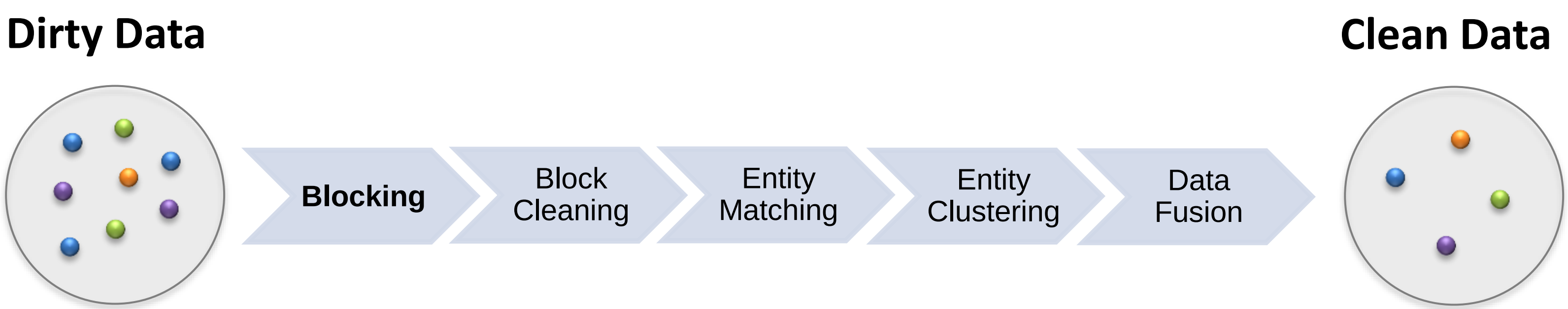
Clean Data (representative records)



- **Blocking**: Cluster together similar records (***blocking function***)
- **Block Cleaning**: Refine blocks to increase precision without affecting recall
- **Entity Matching**: Compare the candidate pairs of records (***matching function***)
- **Entity Clustering**: Partition the retained records into real-world entities
- **Data Fusion**: Obtain from each cluster a single clean record representative of the entity (***conflict resolution function***)

BLOCKING

Cluster together similar profiles

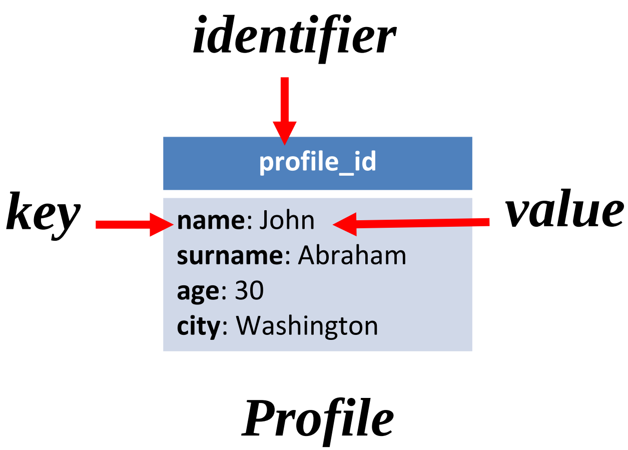


Blocking: Assumptions

1. Every profile is a uniquely identified set of name-value pairs;

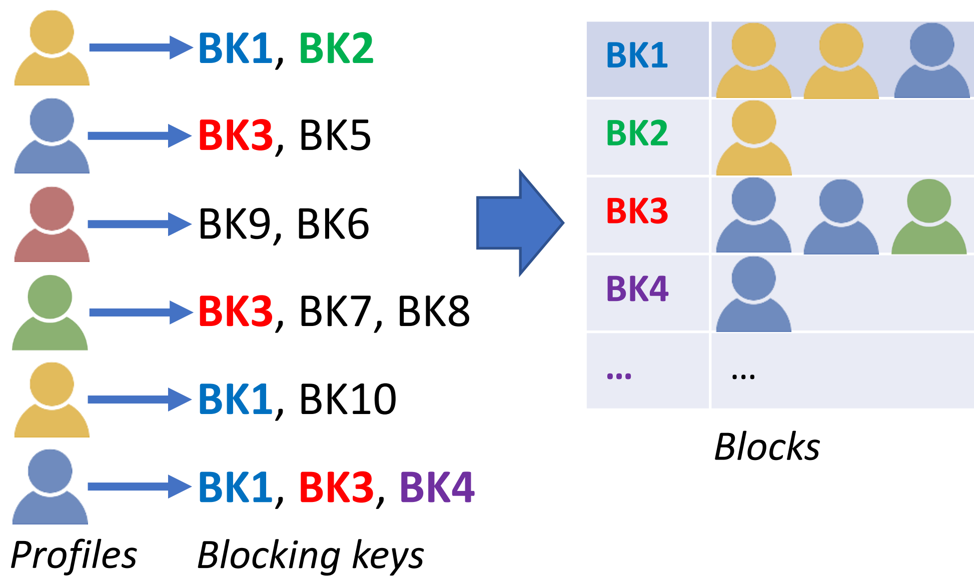
2. Every profile corresponds to a single real-world object;

3. Two matching profiles are detected as matches only if they co-occur in at least one block.



Blocking: General Principles

1. Each profile is described by one or more **blocking keys**;
2. All profiles having the **same** blocking key are placed in the same block.



Block performance is measured with:

- $\text{Recall} = \frac{detected\ matches}{existing\ matches}$
- $\text{Precision} = \frac{detected\ matches}{executed\ comparisons}$

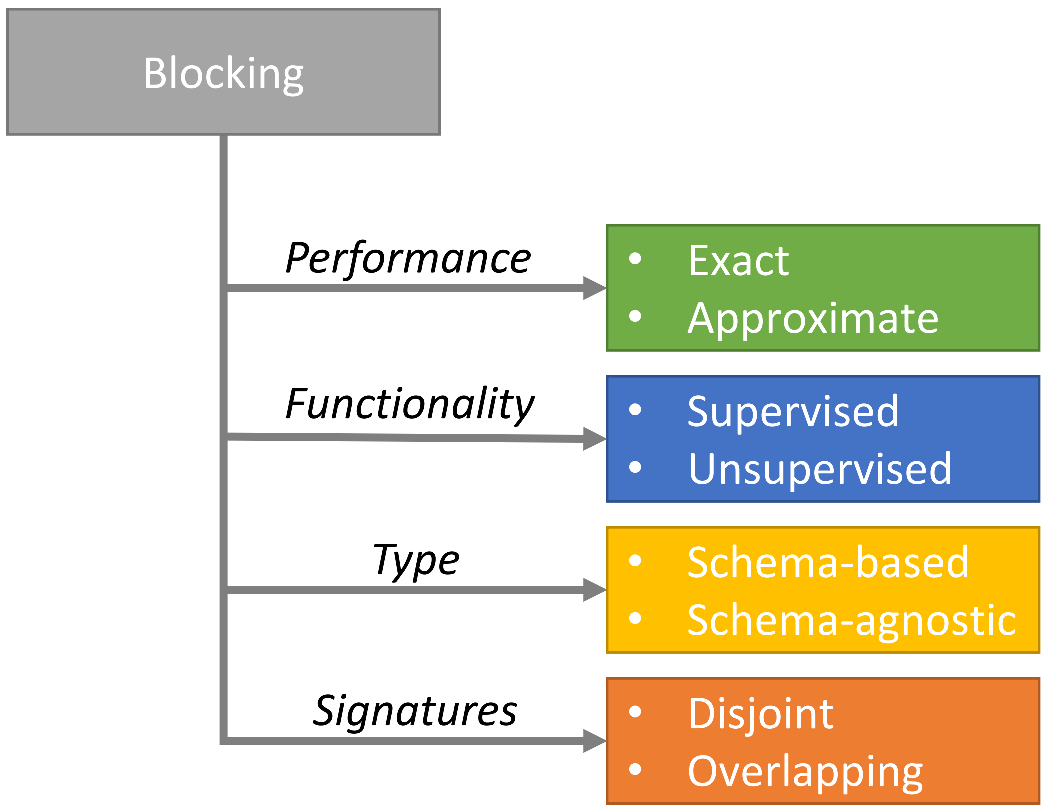
Blocking: Goal

The goal of blocking is to cluster together profiles coming from one dirty (Dirty ER) or two (or more) clean (Clean-Clean ER) data sources, **maximizing** both *Recall* and *Precision*.

Note: there is an **emphasis on *Recall***, since if two entities are matching, then they should co-occur in at least one block.

Blocking: Categorization

Blocking can be categorized based on several parameters.



Blocking: Performance

Exact blocking methods: provide exact results, i.e., given a *similarity predicate*, find all the pairs of profiles that satisfy it.

Similarity join techniques are an example of exact blocking methods.

Maximize the precision (~100%).

Imply *closed-world assumption*: a statement that is true is also known to be true.

This is used also in DBMS: if the DBMS does not have an information about a query (i.e., it does not know the answer), it always replies “false”.

ID	Name
1	Tom Jones
2	Rick Williams
3	Angela Jones
4	Rick Wiliams

Edit Distance (s.Name, r.Name) $\leq$ *2*

ID1	ID2
2	4

Similarity Joins

- Given a set of documents ***D***, a similarity function ***sim*** and a threshold ***t***, a similarity join retrieves all the pairs $r, s \in D \mid sim(r, s) \geq t$

- Similarity joins can be used in many applications such as:
  - Record Linkage
  - Deduplication
  - Data Cleaning
  - Fraud Detection (e.g., identify plagiarism)
  - …

Usage Example: Data Integration

Patients Data

ID	Name	Surname	Age	…

Similarity Join

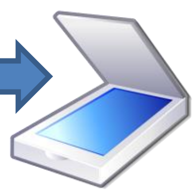
$sim(r, d) \geq 0.9$

$\bowtie$

Matching Records

Patient ID	Clinical record ID

Clinical Records



Digitalization

Optical Character Recognition

Text Documents

Similarity Measures

Token based

- Jaccard similarity
  - $J(x,y) = \frac{|x \cap y|}{|x \cup y|}$
- Cosine similarity
  - $C(x,y) = \frac{x \cdot y}{\|x\| \cdot \|y\|} = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2} \cdot \sqrt{\sum_i y_i^2}}$
- Dice similarity
  - $D(x,y) = \frac{2 \cdot |x \cap y|}{|x| + |y|}$
- Overlap similarity
  - $O(x,y) = |x \cap y|$

Character based

- Edit distance
  - Edit distance between two string is the minimum number of
    - Insertion
    - Deletion
    - Substitution
  - Needed to transform one into the other

Note: all these similarity measures can be traced back to the overlap similarity

Type of Joins	Measure	Definition	Equivalent Overlap Threshold
character-based	Edit Distance	# character transformations	$\max(\lvert x\rvert, \lvert y\rvert)+1-(1+\theta) \times q$
token-based	Overlap	$\lvert \mathbf{x} \cap \mathbf{y} \rvert$	θ
	Cosine	$\lvert \mathbf{x} \cap \mathbf{y} \rvert / \sqrt{\lvert x\rvert \cdot \lvert y\rvert}$	$\theta \times \sqrt{\lvert x\rvert \cdot \lvert y\rvert}$
	Dice	$2 \cdot \lvert \mathbf{x} \cap \mathbf{y} \rvert / (\lvert x\rvert + \lvert y\rvert)$	$\theta \times (\lvert x\rvert + \lvert y\rvert)/2$
	Jaccard	$\lvert \mathbf{x} \cap \mathbf{y} \rvert / (\lvert x\rvert + \lvert y\rvert - \lvert \mathbf{x} \cap \mathbf{y} \rvert)$	$\theta \times (\lvert x\rvert + \lvert y\rvert)/(1+\theta)$

Similarity Joins: Indexing and Filtering

- It is not possible to compare all the possible pairs, due to the time required to perform all the comparisons;

- **Indexing** and **filtering** techniques are employed to reduce the number of comparisons discarding all the pairs that for sure cannot reach the request threshold;

- The most common used indexing technique is called ***prefix index*** (or ***prefix filter***)

- The most common used filters are:
  - ***Length filter***
  - ***Positional filter***

- <u>***See the appendices for more details on these techniques.***</u>

Blocking: Performance

Approximate blocking methods: provide approximate results.

They try to maximize the recall maintaining a high precision.

Imply *open-world assumption*: a statement that is true may be true irrespective of whether or not it is known to be true.

For example, *Token Blocking* [1].

[1] P. Christen: *A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication*. IEEE Transactions on Knowledge and Data Engineering (TKDE) 24(9): 1537-1555 (2012)

ID	Name
1	Tom Jones
2	Rick Williams
3	Angela Jones
4	Rick Wiliams

→ *Token Blocking*

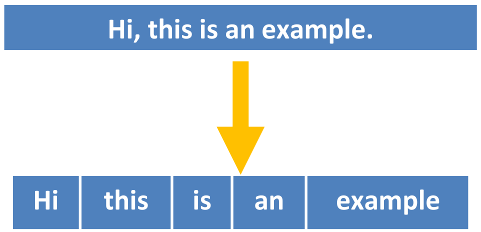
Blocking key	Records
Rick	2, 4
Jones	1, 3

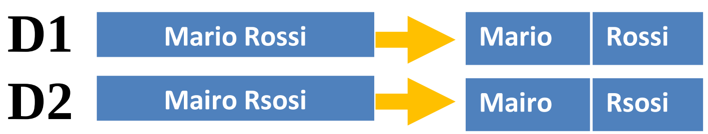
Tokenization

- Many techniques could be used, for example:
  - Word tokenization
  - Q-grams
  - Soundex
  - More complex algorithms

Tokenization: Word Tokenization

- Is the simplest technique, splits the text by punctuation and white spaces;
- Could be useful to compare large documents;
- Do not detect misspelled words.





Jaccard sim (D1, D2) = 0

Tokenization: Q-Grams

- Splits the text into chunks of length *q*, called *q-grams*;
- Useful to detecting misspelled words;
- In large documents generates a lot of tokens.

Example of 3-grams generation

D1	Mario	→	##m	#ma	mar	ari	rio	io#	o##
D2	Mairo	→	##m	#ma	mai	air	iro	ro#	o##

Jaccard sim (D1, D2) = 0.3

Tokenization: Soundex

- Soundex is a phonetic algorithm for indexing names by sound, as pronounced in English;
- The goal is for homophones to be encoded to the same representation so that they can be matched despite minor differences in spelling;
- Soundex is the most widely known of all phonetic algorithms because it is implemented by the most popular DBMS (e.g., MySQL, SQL Server, PostgreSQL, DB2, etc.).

D1 Mario → M600

D2 Mairo → M600

Blocking: Functionality

Supervised: the user learns the best blocking key, identifying the best combination of attribute names and transformations (require high user effort). For example, *Magellan* [1].

[1] A. Doan et al.: *Magellan: toward building ecosystems of entity matching solutions*. Communications of the ACM (CACM) 63(8): 83-91 (2020)

Data Source A

	Name	Surname	Address
r1	John	Doe	Abraham street
r2	Abraham	Lincoln	High street

Data Source B

```
{                                   r3
    fullname: "john doe",
    location: "Abraham st"
}
```

```
{                                   r4
    fullname: "Abraham Lincoln",
    location: "high str"
}
```

Transformation

ID	Name+Surname	Address
r1	John Doe	Abraham street
r2	Abraham Lincoln	High street

ID	fullname	location
r3	John Doe	Abraham st
r4	Abraham Lincoln	High str

Blocking keys: A.name+surname, B.fullname

Block ID	Profiles
John Doe	r1, r3
Abraham Lincoln	r2, r4

user

Blocking: Functionality

Unsupervised: general methods, do not require user intervention.

For example, *Token Blocking* [1].

[1] P. Christen: *A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication*. IEEE Transactions on Knowledge and Data Engineering (TKDE) 24(9): 1537-1555 (2012)

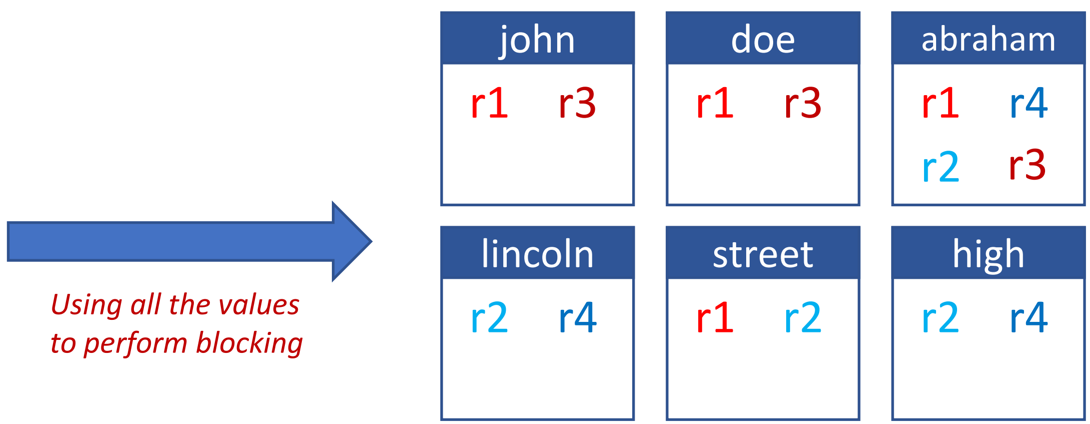
Data Source A

	Name	Surname	Address
r1	John	Doe	Abraham street
r2	Abraham	Lincoln	High street

Data Source B

```
{                                   r3
    fullname: "john doe",
    location: "Abraham st"
}

{                                   r4
    fullname: "Abraham Lincoln",
    location: "high str"
}
```

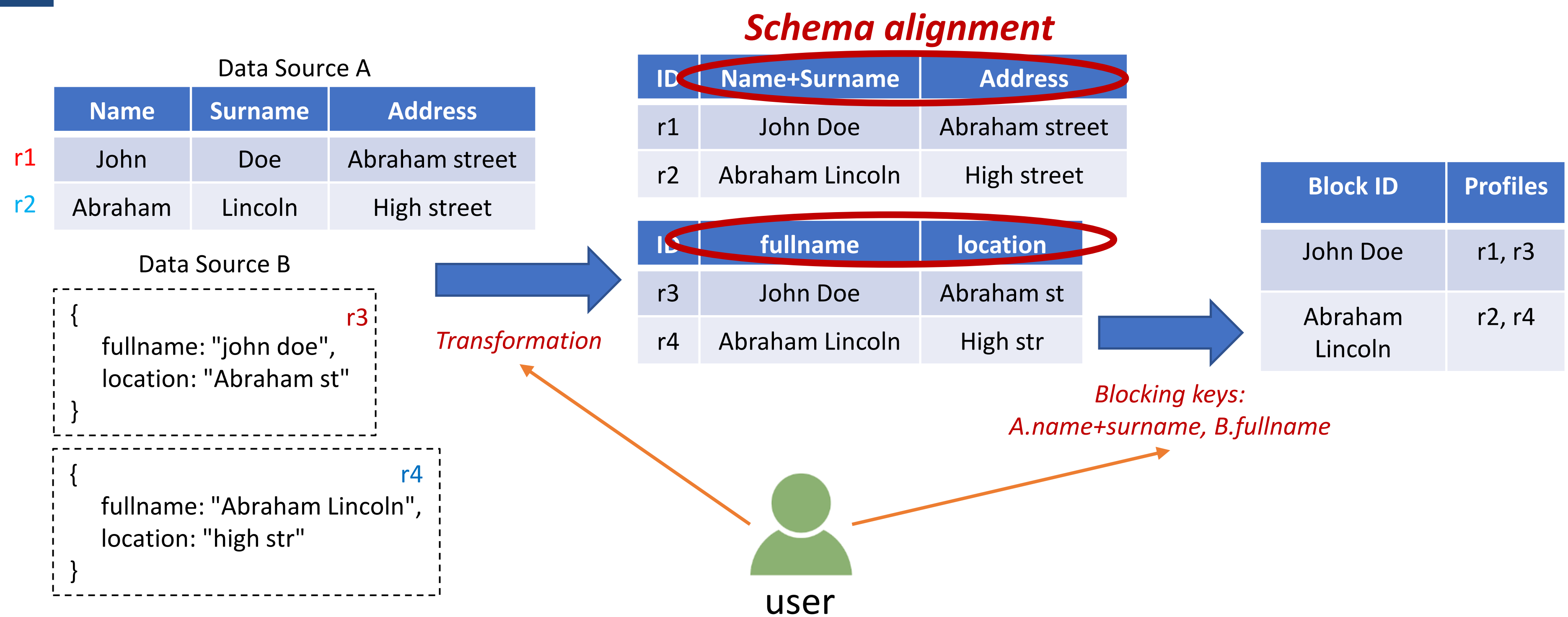


Blocking: Type

Schema-based: require schema alignment. It might be a hard task, especially with heterogenous data. For example, *Magellan* [1].

In the example below, the schemas have been aligned; to do so, the original schema was modified by merging the attributes name and surname.

[1] A. Doan et al.: *Magellan: toward building ecosystems of entity matching solutions*. Communications of the ACM (CACM) 63(8): 83-91 (2020)



Blocking: Type

Schema-agnostic: do not require schema alignment, since the schema is not considered. For example, *Token Blocking* [1].

In the example below, only the values are used to generate the blocks.

[1] P. Christen: *A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication*. IEEE Transactions on Knowledge and Data Engineering (TKDE) 24(9): 1537-1555 (2012)

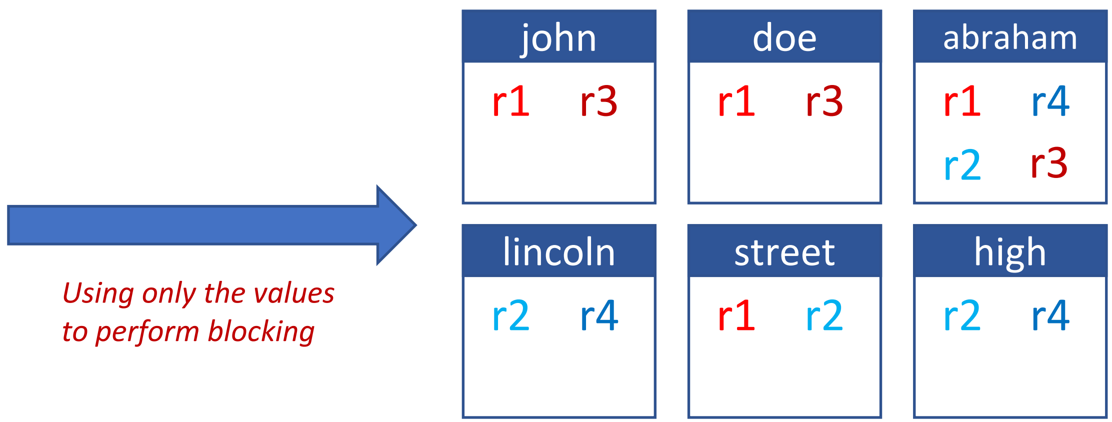
Data Source A

	Name	Surname	Address
r1	John	Doe	Abraham street
r2	Abraham	Lincoln	High street

Data Source B

```
{                                    r3
    fullname: "john doe",
    location: "Abraham st"
}
```

```
{                                    r4
    fullname: "Abraham Lincoln",
    location: "high str"
}
```



Blocking: Signatures

Disjoint: only one blocking key is assigned to each profile, so that it is contained in only one block. For example, *Standard Blocking* [1].

Standard blocking selects the most appropriate attribute(s) w.r.t. noise and distinctiveness, transforming the corresponding values into a blocking key.

[1] G. Papadakis, T. Palpanas: *Web-scale, Schema-Agnostic, End-to-End Entity Resolution*. Tutorial at the ACM International Web Conference (WWW) (2018)

Data Source A

	Name	Surname	Address
r1	John	Doe	Abraham street
r2	Abraham	Lincoln	High street

Data Source B

```
{                                   r3
    fullname: "john doe",
    location: "Abraham st"
}
```

```
{                                   r4
    fullname: "Abraham Lincoln",
    location: "high str"
}
```

Blocking keys: A.name+surname, B.fullname

ID	Blocking key
r1	John Doe
r2	Abraham Lincoln
r3	John Doe
r4	Abraham Lincoln

Blocking

Block ID	Profiles
John Doe	r1, r3
Abraham Lincoln	r2, r4

Blocking: Signatures

Overlapping: multiple blocking keys are assigned to each profile, so that a profile can be contained in multiple blocks. For example, *Token Blocking* [1].

Token Blocking uses all the tokens as blocking keys.

[1] P. Christen: [A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication](). IEEE Transactions on Knowledge and Data Engineering (TKDE) 24(9): 1537-1555 (2012)

Data Source A

	Name	Surname	Address
r1	John	Doe	Abraham street
r2	Abraham	Lincoln	High street

Data Source B

r3
```
{
    fullname: "john doe",
    location: "Abraham st"
}
```

r4
```
{
    fullname: "Abraham Lincoln",
    location: "high str"
}
```

Token generation

ID	Blocking keys
r1	john, doe, abraham, street
r2	abraham, lincoln, high, street
r3	john, doe, abraham, st
r4	abraham, lincoln, high, str

Blocking

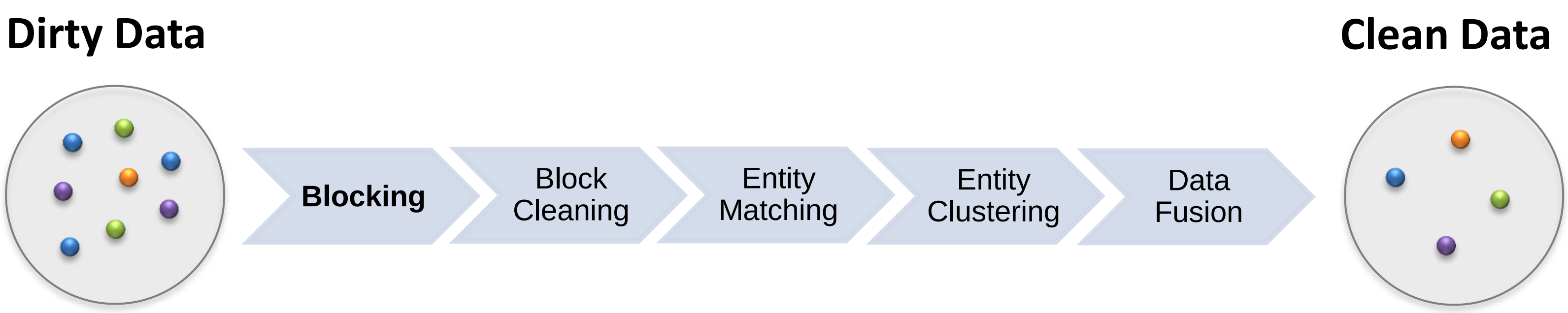
john	doe	abraham
r1 r3	r1 r3	r1 r4 r2 r3

lincoln	street	high
r2 r4	r1 r2	r2 r4

BLOCKING TECHNIQUES

The techniques presented here are ***approximated****,* ***unsupervised****,* ***schema-agnostic****, and* ***overlapping****.*

These techniques are the most suitable when dealing with data with high heterogeneity, including unstructured data sources, and data changing with high frequency.



Schema-agnostic Token Blocking

Functionality:

1. Given a profile, extract all the tokens that are contained in its attribute values;
2. Create one block for every distinct token: each block contains all profiles with the corresponding token (note that each block should contain at least two profiles).

[1] G. Papadakis, T. Palpanas: *Web-scale, Schema-Agnostic, End-to-End Entity Resolution*. Tutorial at the ACM International Web Conference (WWW) (2018)

Schema-agnostic Token Blocking: Example

Data Source A

	Name	Surname	Address
r1	John	Doe	Abraham street
r2	Abraham	Lincoln	High street

Data Source B

```
{                                   r3
   fullname: "john doe",
   location: "Abraham st"
}
```

```
{                                   r4
   fullname: "Abraham Lincoln",
   location: "high str"
}
```

john	doe	abraham	lincoln	street	high
r1 r3	r1 r3	r1 r4 r2 r3	r2 r4	r1 r2	r2 r4

Since the Token Blocking is schema-agnostic, note that the **Abraham** block contains both profiles using “Abraham” as a person name and profiles using “Abraham” as an address.

Loosely Schema-aware Token Blocking

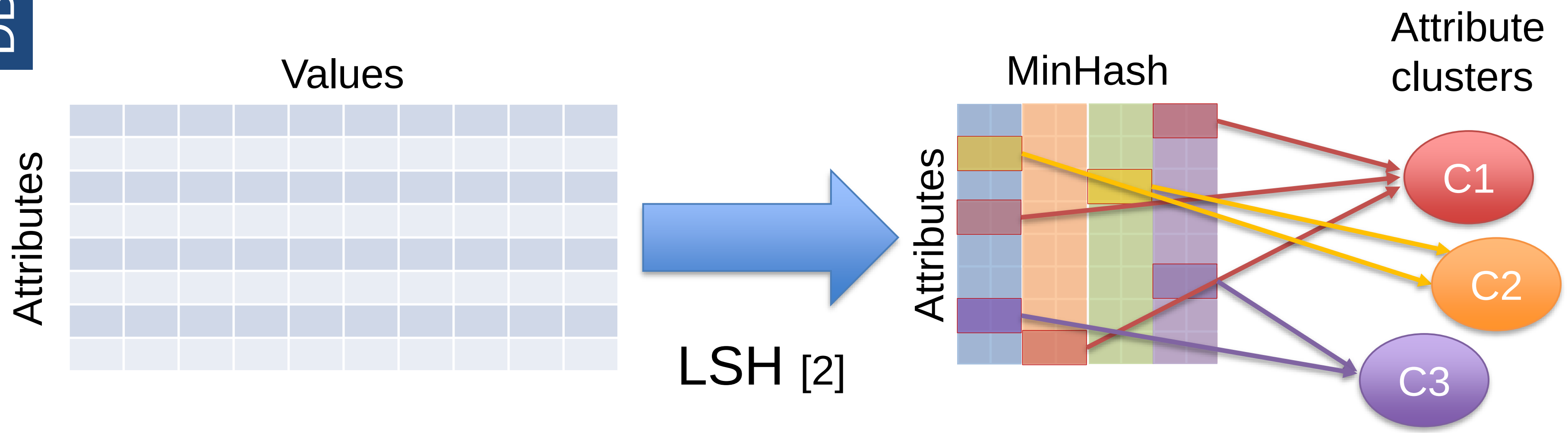
Functionality:

1. Automatically cluster together similar attributes (Attribute Clustering);
2. Given a profile, extract all the tokens that are contained in its attribute values, taking into account the generated clusters (i.e., by adding the cluster id to each token);
3. Create one block for every distinct token: each block contains all profiles with the corresponding token.

[1] G. Simonini, S. Bergamaschi, H. V. Jagadish: *BLAST: a Loosely Schema-aware Meta-blocking Approach for Entity Resolution*. Proceedings of the VLDB Endowment (PVLDB) 9(12): 1173-1184 (2016)

Attribute Clustering

Intuition: similar attributes have similar values



[1] G. Papadakis et al.: *Meta-Blocking: Taking Entity Resolution to the Next Level*. IEEE Transactions on Knowledge and Data Engineering (TKDE) 26(8): 1946-1960 (2014)

[2] J. Leskovec et al.: *Mining of Massive Datasets*. Cambridge University Press (2014)

Loosely Schema-aware Token Blocking: Example

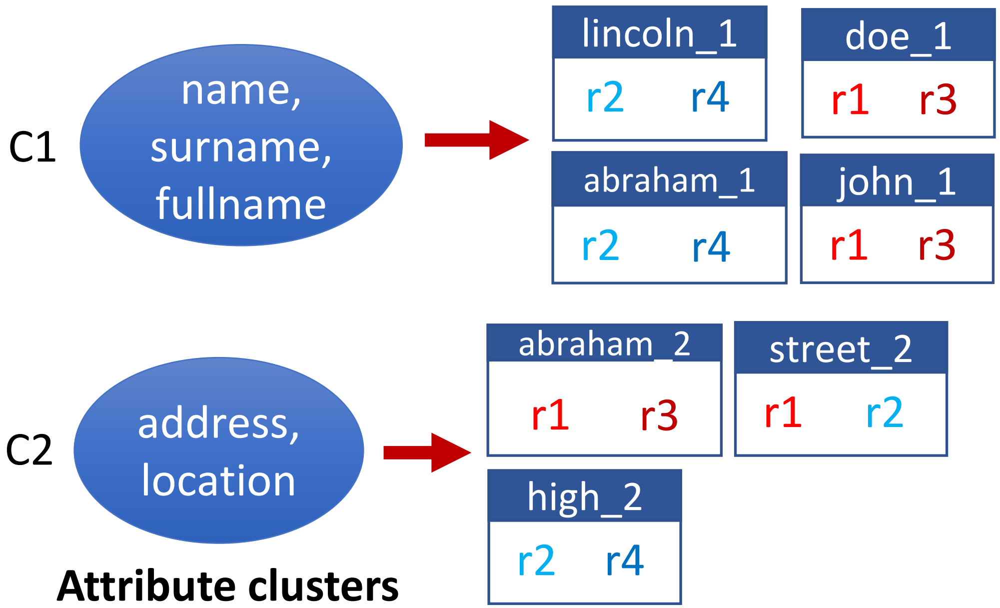
Data Source A

	Name	Surname	Address
r1	John	Doe	Abraham street
r2	Abraham	Lincoln	High street

Data Source B

```
{                                   r3
   fullname: "john doe",
   location: "Abraham st"
}
```

```
{                                   r4
   fullname: "Abraham Lincoln",
   location: "high str"
}
```

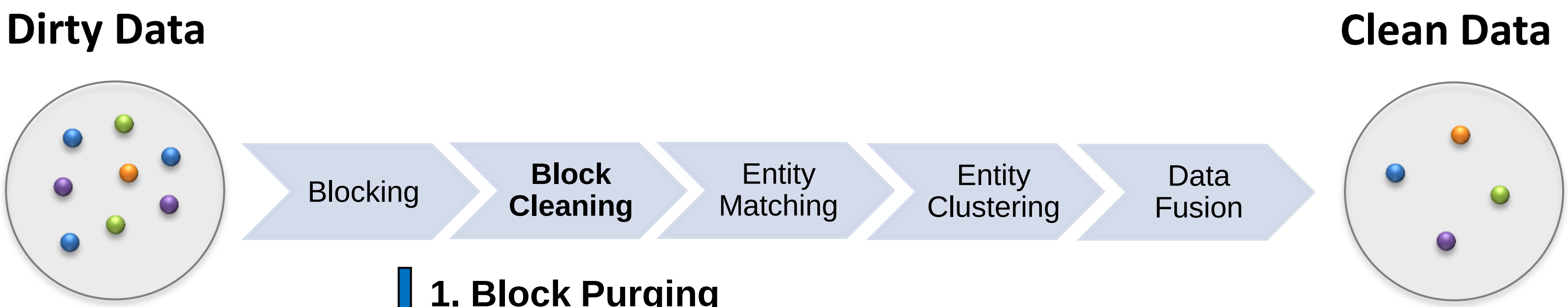


Attribute clusters

Now it is possible to see that using the Attribute Clustering information the token “Abraham” used as an address produces a different block from the token “Abraham” used as a person name.

BLOCK CLEANING

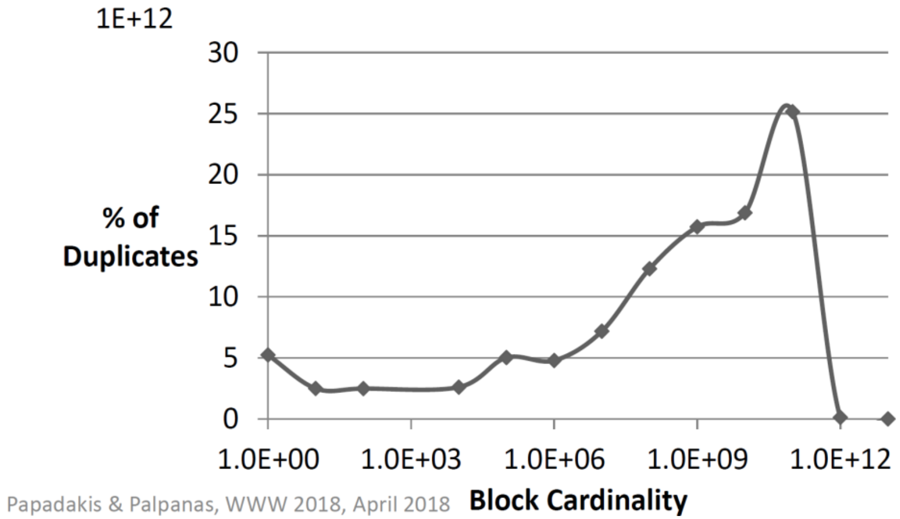
Refine blocks to increase precision without affecting recall



1. Block Purging
2. Block Filtering
3. Meta-Blocking

Block Purging

- Removes **oversized blocks** (i.e., many comparisons, no duplicates).
- Discards them by setting an upper limit on the **cardinality** (i.e., number of comparisons) of each block [1].



[1] G. Papadakis et al.: *Meta-Blocking: Taking Entity Resolution to the Next Level*.
IEEE Transactions on Knowledge and Data Engineering (TKDE) 26(8): 1946-1960 (2014)

Block Filtering

Ideas:

- Each profile has a different relevance in different blocks;
- Larger blocks are less likely to contain unique duplicates (i.e., duplicates that are not generated by other blocks), thus less relevant.

Note that this requires **overlapping signatures**

Blocks

abraham	high	john	doe	lincoln	street
r1 r4 r2 r3	r2 r4	r1 r3	r1 r3	r2 r4	r1 r2

Entity Index

r1 -> {john, doe, street, ~~abraham~~}
r2 -> {high, lincoln, street, ~~abraham~~}
r3 -> {john, doe, ~~abraham~~}
r4 -> {high, lincoln, ~~abraham~~}

- **For each profile, an inverted index "profile -> blocks" is built;**
- **the blocks are sorted by their cardinality;**
- **the biggest blocks are removed (usually the 80% of the smallest ones is kept)**

Filtered Blocks

high	john	doe	lincoln	street
r2 r4	r1 r3	r1 r3	r2 r4	r1 r2

Blocks are reconstructed

[1] G. Papadakis et al.: *Meta-Blocking: Taking Entity Resolution to the Next Level*.
IEEE Transactions on Knowledge and Data Engineering (TKDE) 26(8): 1946-1960 (2014)

Meta-Blocking

- **Main idea**: the more blocks two entity profiles share, the more likely they match.

- **Goal**: restructure a **redundancy-positive** block collection B into a new one B' in which $recall(B') \cong recall(B)$ and $precision(\mathrm{B}') \gg precision(B)$

[1] G. Papadakis et al.: *Meta-Blocking: Taking Entity Resolution to the Next Level*.
IEEE Transactions on Knowledge and Data Engineering (TKDE) 26(8): 1946-1960 (2014)

Schema-agnostic Meta-Blocking

Data Source A

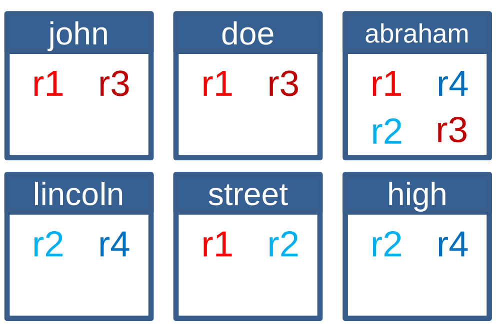
	Name	Surname	Address
r1	John	Doe	Abraham street
r2	Abraham	Lincoln	High street

Data Source B

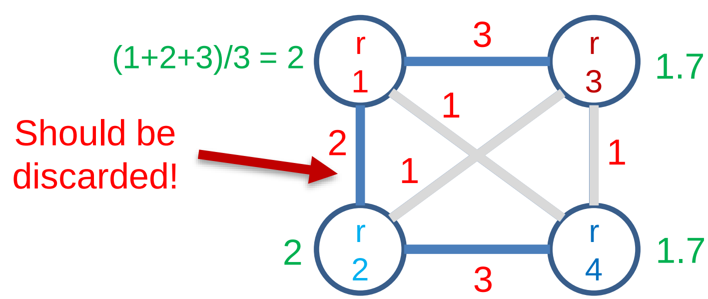
```
{                                   r3
   fullname: "john doe",
   location: "Abraham st"
}
```

```
{                                   r4
   fullname: "Abraham Lincoln",
   location: "high str"
}
```

Blocking (e.g. Token Blocking)



Meta-Blocking



[1] G. Papadakis et al.: *Meta-Blocking: Taking Entity Resolution to the Next Level*.
IEEE Transactions on Knowledge and Data Engineering (TKDE) 26(8): 1946-1960 (2014)

Loosely Schema-aware Meta-Blocking

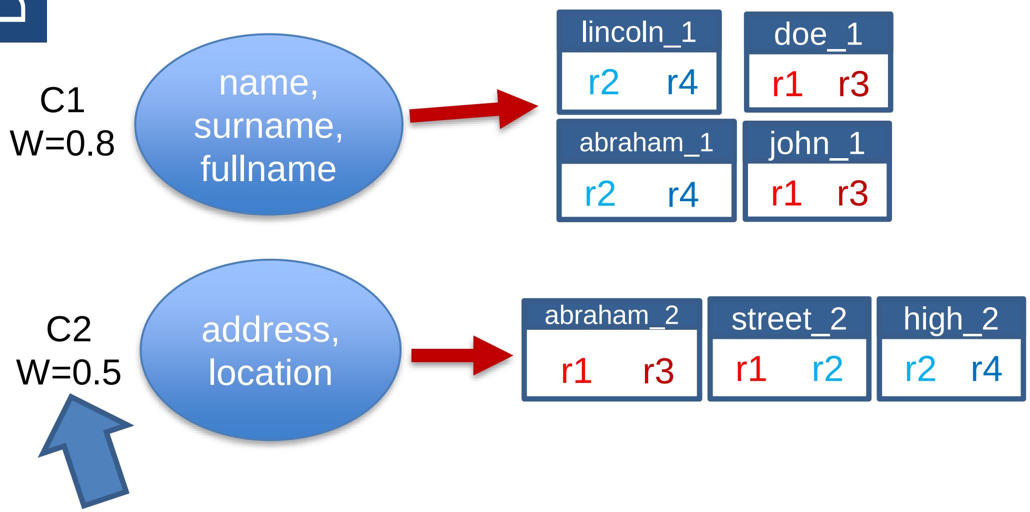
Data Source A

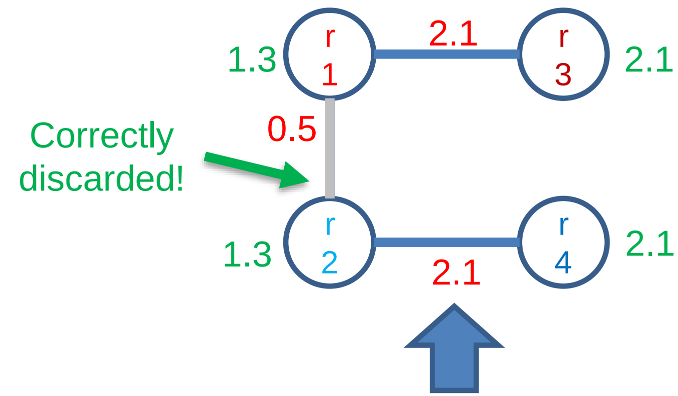
	Name	Surname	Address
r1	John	Doe	Abraham street
r2	Abraham	Lincoln	High street

Data Source B

```
{                                  r3
   fullname: "john doe",
   location: "Abraham st"
}
```

```
{                                  r4
   fullname: "Abraham Lincoln",
   location: "high str"
}
```





Using the Loosely Schema-aware Meta-Blocking it is possible to associate a weight to each cluster (attribute cluster token entropy)

And then use it to weight the edges according to which cluster the comparison belongs

[1] G. Simonini. S. Bergamaschi, H. V. Jagadish: *BLAST: a Loosely Schema-aware Meta-blocking Approach for Entity Resolution*. Proceedings of the VLDB Endowment (PVLDB) 9(12): 1173-1184 (2016)

Meta-Blocking Performances

Billion Triple Challenge 2012 dataset: it is composed by two datasets, one contains the data of DBpedia 3.7 (4.2M of profiles), the other one the data of Freebase (3.7M of profiles).

Configuration: Intel Xeon E5-2670v2 2.50 GHz (20 cores) and 128 GB of RAM

	Recall (before ER)	Number of comparisons	Overhead	Entity Matching time (0.05 ms/comparison)
Without blocking	100%	$1.6 \cdot 10^{13}$	-	~25 years
TB+WNP * [1]	81%	$4.8 \cdot 10^{10}$	70 min	28 days
BLAST [2]	81%	$3.8 \cdot 10^{8}$	90 min	5 hours

** Token Blocking + Weight Node Pruning*

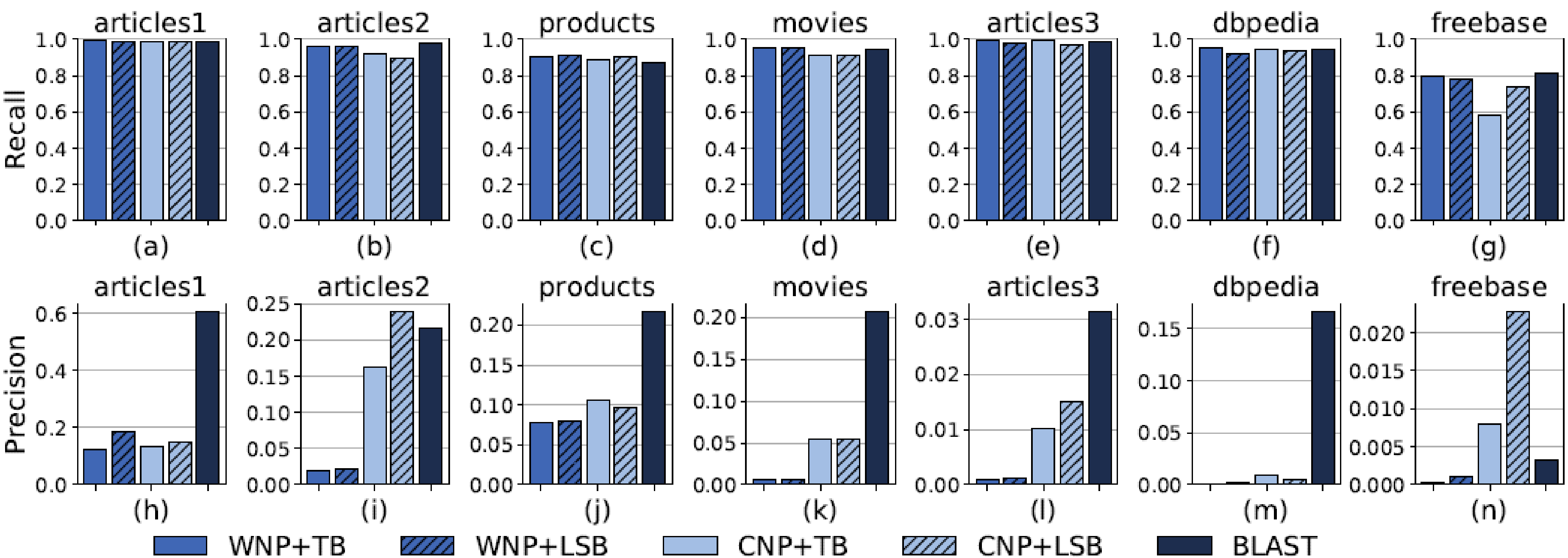
[1] G. Papadakis et al.: *Meta-Blocking: Taking Entity Resolution to the Next Level*. IEEE Transactions on Knowledge and Data Engineering (TKDE) 26(8): 1946-1960 (2014)

[2] G. Simonini, S. Bergamaschi, H. V. Jagadish: *BLAST: a Loosely Schema-aware Meta-blocking Approach for Entity Resolution*. Proceedings of the VLDB Endowment (PVLDB) 9(12): 1173-1184 (2016)

Experimental Results

Datasets

	Size	$\lvert\mathcal{P}_1\rvert - \lvert\mathcal{P}_2\rvert$	$\lvert\mathcal{A}_1\rvert - \lvert\mathcal{A}_2\rvert$	$\lvert\mathcal{D}_P\rvert$
articles1 (*)	small	2.6k - 2.3k	4 - 4	2.2k
articles2 (*)	small	2.5k - 61k	4 - 4	2.3k
products (*)	small	1.1k - 1.1k	4 - 4	1.1k
movies	small	28k - 23k	4 - 7	23k
articles3 (*)	large	1.8M - 2.5M	7 - 7	0.6M
dbpedia	large	1.2M - 2.2M	30k - 50k	0.9M
freebase	large	4.2M - 3.7M	37k - 11k	1.5M



SparkER: Scaling Entity Resolution in Spark

SparkER [3, 4] is an ER framework developed in Scala for **Apache Spark**.

It implements for Spark the Meta-Blocking techniques described in the previous slides and in [1, 2].

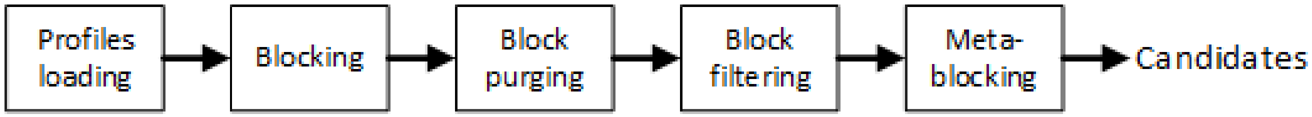
[1] G. Papadakis et al.: *Meta-Blocking: Taking Entity Resolution to the Next Level*. IEEE Transactions on Knowledge and Data Engineering (TKDE) 26(8): 1946-1960 (2014)

[2] G. Simonini, S. Bergamaschi, H. V. Jagadish: *BLAST: a Loosely Schema-aware Meta-blocking Approach for Entity Resolution*. Proceedings of the VLDB Endowment (PVLDB) 9(12): 1173-1184 (2016)

[3] L. Gagliardelli, G. Simonini, D. Beneventano, S. Bergamaschi: *SparkER: Scaling Entity Resolution in Spark*. International Conference on Extending Database Technology (EDBT), 602-605 (2019)

[4] G. Simonini, L. Gagliardelli, S. Bergamaschi, H. V. Jagadish: *Scaling entity resolution: A loosely schema-aware approach*. Information Systems (IS) 83: 145-165 (2019)

SparkER: Scaling Entity Resolution in Spark

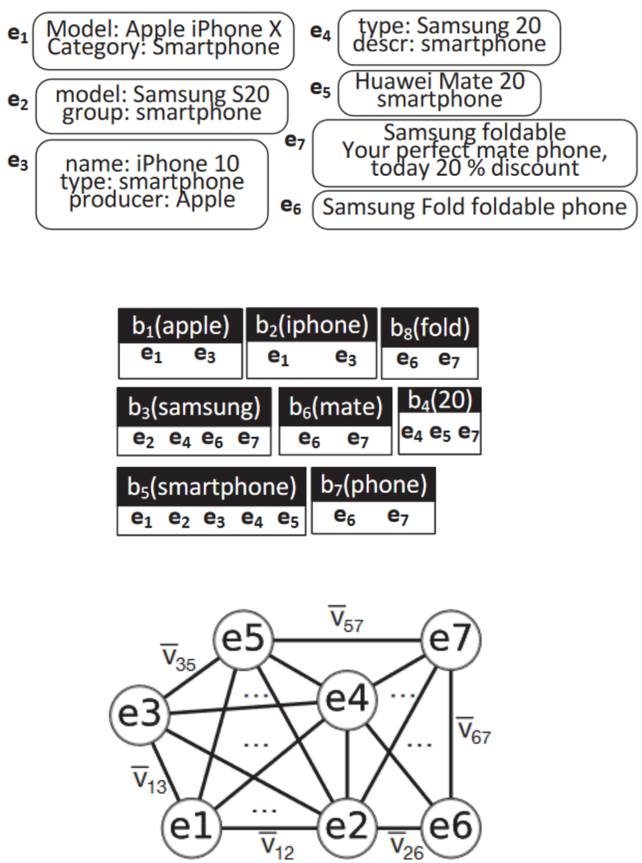


The process is composed of different stages:

- **Profile Loading**: loads the data (supports CSV, JSON and serialized formats) into profiles;
- **Blocking**: performs the blocking, Token Blocking or Loose Schema Blocking;
- **Block Purging**: removes the biggest blocks, which usually represent stopwords or very common tokens that do not provide significant relations;
- **Block Filtering**: for each profile, filters out the biggest blocks;
- **Meta-Blocking**: performs the Meta-Blocking, producing as results the list of candidate pairs that might be matches.

The source code of **SparkER** is available on GitHub and its complete and detailed documentation can be found on Read the Docs.

Our Latest Work: Generalized Supervised Meta-Blocking



The profiles collection contains three duplicate pairs: (e1, e3), (e2, e4), (e6, e7)

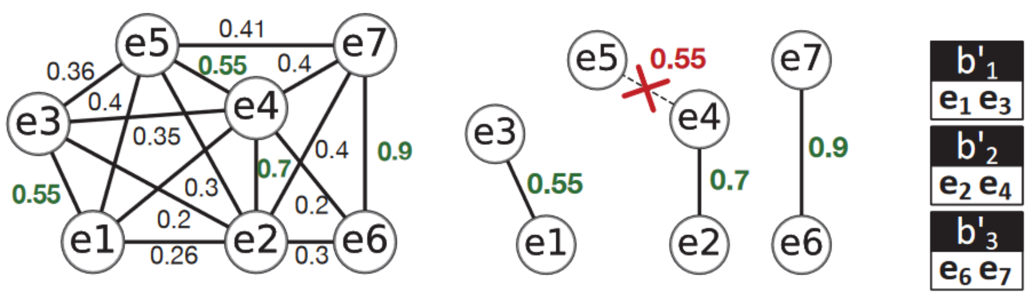
1. The profiles are clustered together by using Token Blocking.
2. The meta-blocking graph is generated as follows: each entity profile is represented as a node; two nodes are connected by an edge if the corresponding profiles co-occur in at least one block; each edge is associated with a feature vector that contains several metric regarding the corresponding profiles (e.g., number of shared blocks, Jaccard Similarity, etc.).

[1] L. Gagliardelli, G. Papadakis, G. Simonini, S. Bergamaschi, T. Palpanas: *Generalized Supervised Meta-blocking*. Proceedings of the VLDB Endowment (PVLDB) 15 (2022)

VLDB 2022

Our Latest Work: Generalized Supervised Meta-Blocking

3. By using a balanced sample of edges (i.e., 50% matches, 50% non matches), a probabilistic classifier (e.g., logistic regression) is trained to predict if an edge is a match or not. Then, each edge is weighted with the probability of being a match.
4. The pruning is performed as in unsupervised meta-blocking by considering only the edges with a probability of being a match equal or greater than 0.5
5. Finally, the pruned blocking collection is obtained.

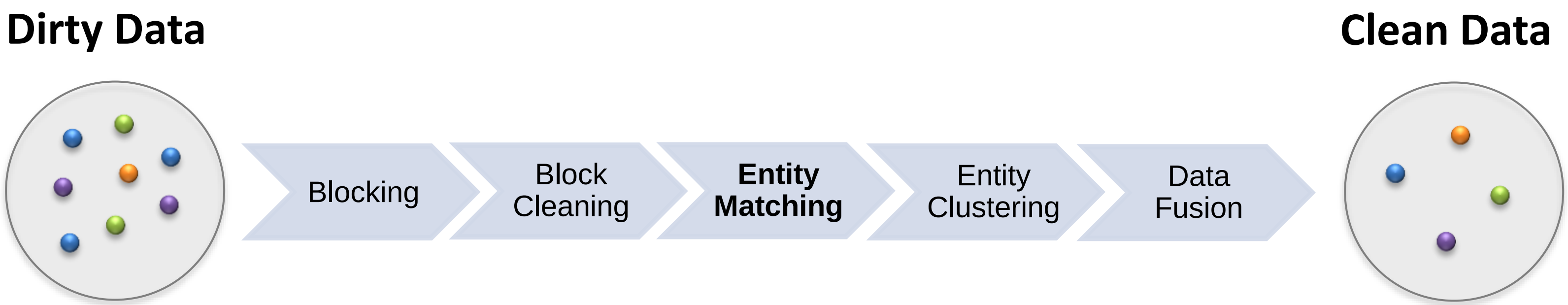


[1] L. Gagliardelli, G. Papadakis, G. Simonini, S. Bergamaschi, T. Palpanas: *Generalized Supervised Meta-blocking*. Proceedings of the VLDB Endowment (PVLDB) 15 (2022)

VLDB 2022

ENTITY MATCHING

Establishing if each candidate pair of profiles refers or not to the same real-world entity



Entity Matching

- Meta-Blocking provides **candidate pairs** of profiles
- Entity Matching is needed in order to **decide whether a pair is a match or not**
- Different techniques can be used:
  - Supervised
    - Crowdsourcing
    - Classifiers
    - …
  - Unsupervised
    - User Defined Functions
    - Heuristics (e.g., Cosine Similarity, Jaccard Similarity, etc.)
    - …
  - Human in the loop

State-of-the-art Entity Matching Solutions

State-of-the-art frameworks for Entity Matching rely on:

- **Machine Learning (ML)** (e.g., *Magellan* [1])
- **Deep Learning (DL)** (e.g., *DeepMatcher* [2])

[1] A. Doan et al.: *Magellan: toward building ecosystems of entity matching solutions*. Communications of the ACM (CACM) 63(8): 83-91 (2020)

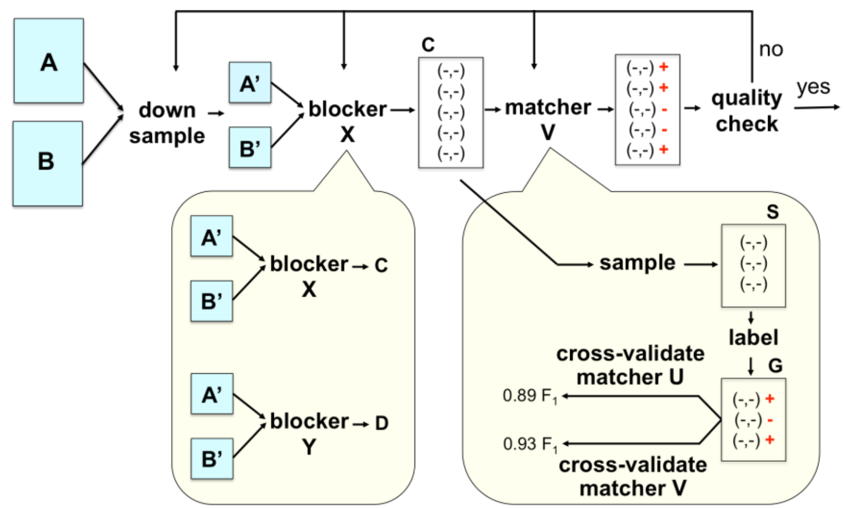
[2] S. Mudgal et al.: *Deep Learning for Entity Matching: A Design Space Exploration*. ACM International Conference on Management of Data (SIGMOD), 19-34 (2018)

Magellan

Project led by the group of **AnHai Doan** (see on GitHub); GreenBay Technologies (born to commercialize Magellan) acquired by Informatica in 2020

Entity Matching solutions based on rules and on Machine Learning (Decision Tree, Random Forest, Naïve Bayes, Logistic Regression, Linear Regression, SVM, XGBoost)

Includes also blocking functions



Select the best blocker: X, Y

Select the best matcher: U, V

[1] P. Konda et al.: *Magellan: Toward Building Entity Matching Management Systems*. Proceedings of the VLDB Endowment (PVLDB) 9(12): 1197-1208 (2016)

[2] A. Doan et al.: *Magellan: toward building ecosystems of entity matching solutions*. Communications of the ACM (CACM) 63(8): 83-91 (2020)

DeepMatcher

Another project led by the group of **AnHai Doan** (see on [GitHub]())

Entity Matching solutions based on Deep Learning:

- **SIF**: considers the words present in each attribute value pair (no word order).
- **RNN**: considers the sequences of words present in each attribute value pair.
- **Attention**: considers the alignment of words present in each attribute value pair (no word order).
- **Hybrid**: considers the alignment of sequences of words present in each attribute value pair (default).

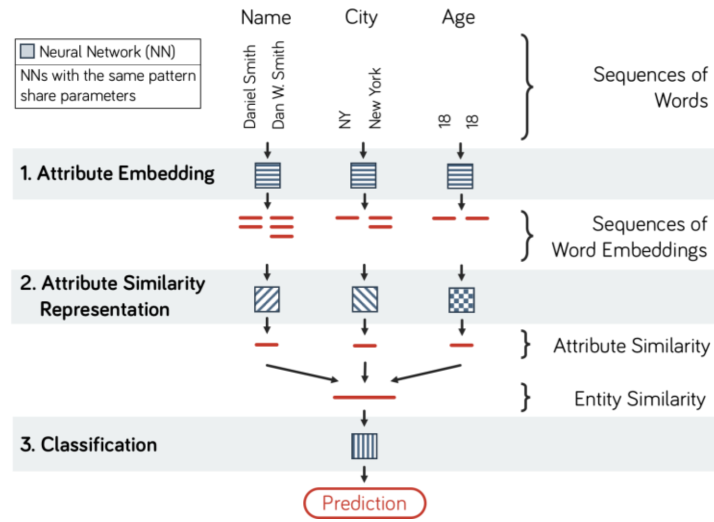
Example Input:

t_1

Name	City	Age
Daniel Smith	NY	18

t_2

Name	City	Age
Dan W. Smith	New York	18



[1] S. Mudgal et al.: *[Deep Learning for Entity Matching: A Design Space Exploration]()*. ACM International Conference on Management of Data (SIGMOD), 19-34 (2018)

The Labeling Issue: Reducing Human Effort

How to deal with the **need for labeled data** (requiring a significant **human effort**) to train the models?

Two main research directions:

- **Transfer Learning (TL)**
- **Active Learning (AL)**

Research Directions: Transfer Learning

Storing knowledge gained while solving a problem and applying it to a different but related problem (**pre-trained EM models**) (e.g., *Ditto* [1])

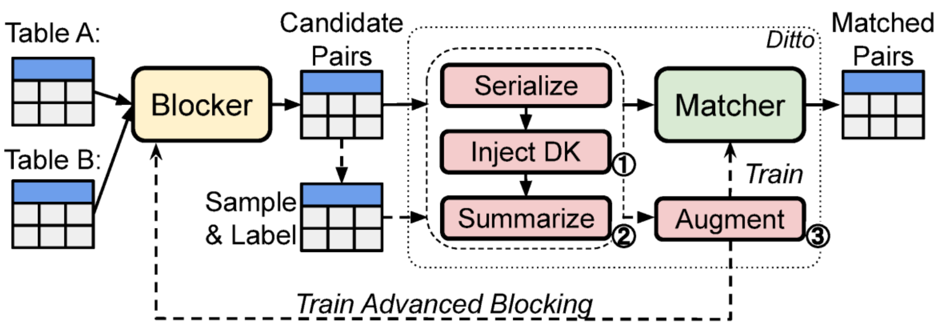
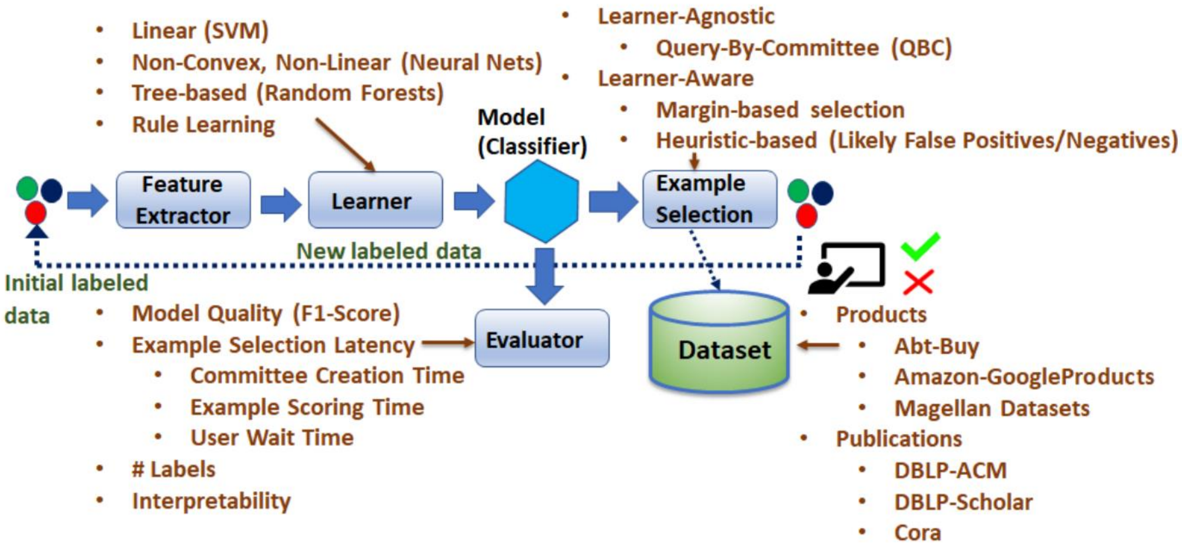


Figure 2: An EM system architecture with DITTO as the matcher. In addition to the training data, the user of DITTO can specify (1) a method for injecting domain knowledge (DK), (2) a summarization module for keeping the essential information, and (3) a data augmentation (DA) operator to strengthen the training set.

[1] Y. Li et al.: *Deep Entity Matching with Pre-Trained Language Models*. Proceedings of the VLDB Endowment (PVLDB) 14(1): 50-60 (2020)

Research Directions: Active Learning

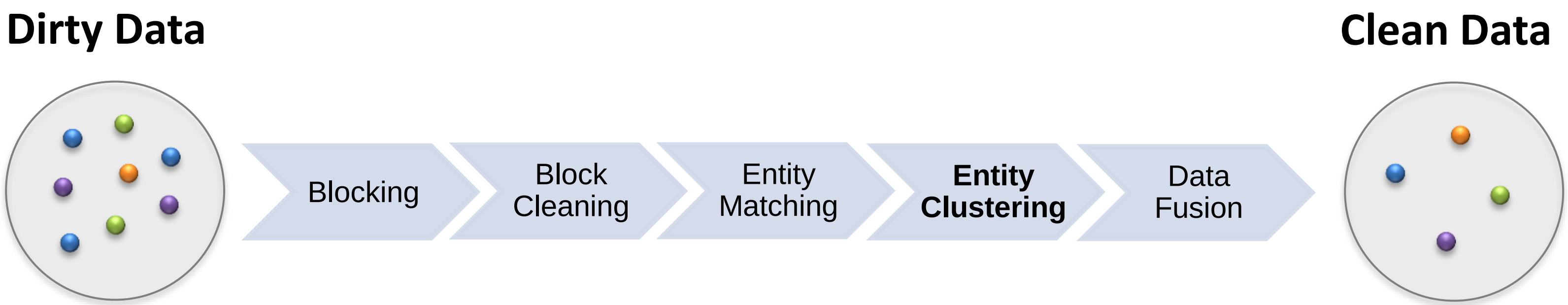
The learning algorithm interactively queries a user or a source (**oracle**) to label dynamically collected ambiguous examples in order to refine the learned model (classifier) upon them



[1] V. Meduri et al.: *A Comprehensive Benchmark Framework for Active Learning Methods in Entity Matching*. ACM International Conference on Management of Data (SIGMOD), 1133-1147 (2020)

ENTITY CLUSTERING

From pairs of matching profiles to consistent clusters of matches, each one referring to a different entity



Entity Clustering

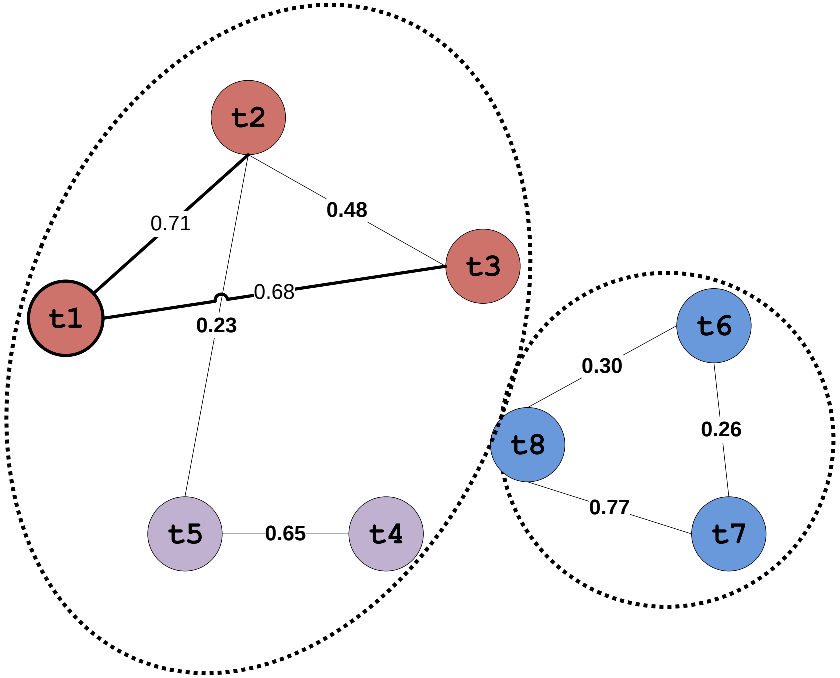
- Entity Matching provides **pairs of profiles that are identified as true matches**
- These pairs may refer to the same entity
- Entity Clustering partitions matched pairs (correspondences) into equivalence clusters

- **Input**
  - Matched pairs of profiles = Similarity Graph:
    - Nodes → Profiles, Edges → Candidate matches, Edge weights → similarity
- **Output**
  - Equivalence Clusters

Entity Clustering Algorithms

- Connected Components (Transitive Closure)
- CENTER
- MERGE-CENTER

The standard approach
But
- May put together many dissimilar profiles (low threshold)
- May split many similar profiles (high thresholds)
- Very sensitive to the value of the threshold used for the similarity join

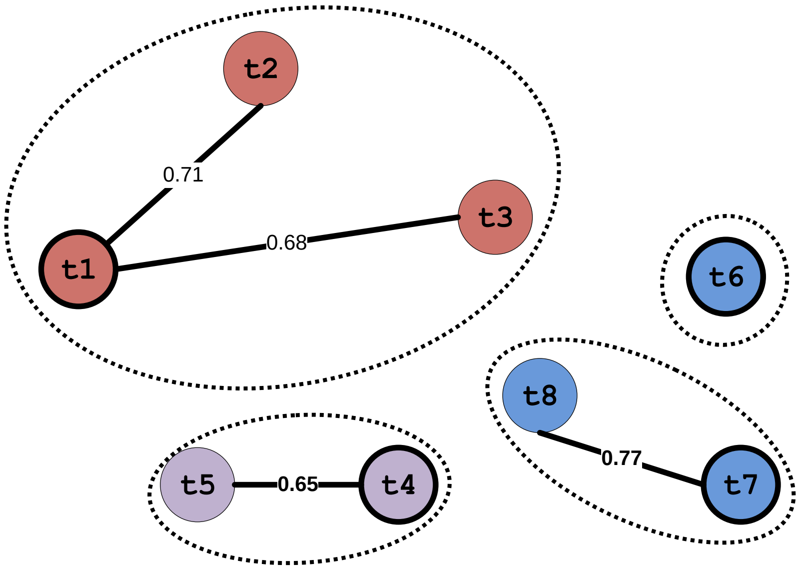


[1] G. Papadakis, T. Palpanas: *Web-scale, Schema-Agnostic, End-to-End Entity Resolution*. Tutorial at the ACM International Web Conference (WWW) (2018)

Entity Clustering Algorithms

- Connected Components (Transitive Closure)
- **CENTER**
- MERGE-CENTER

The CENTER algorithm performs clustering by partitioning the similarity graph into clusters that have a center, and all profiles in each cluster are similar to the center of the cluster.

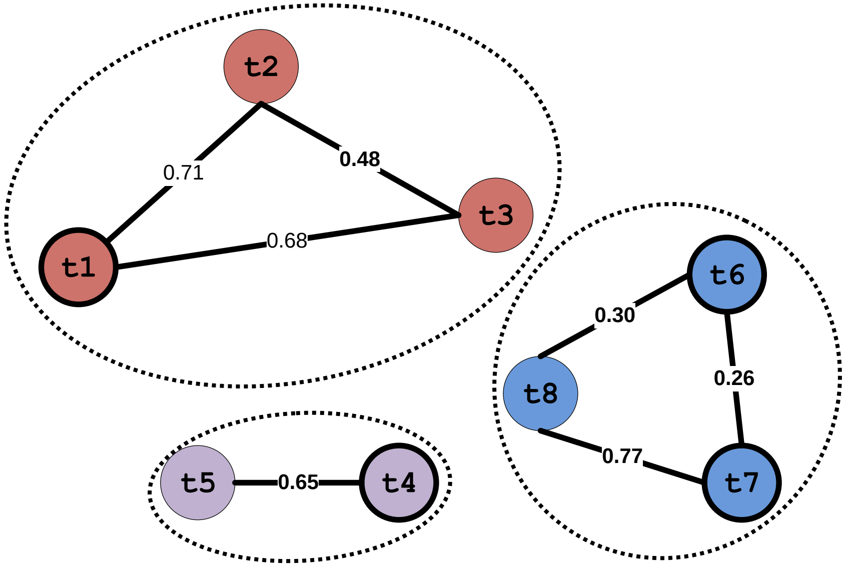


[1] G. Papadakis, T. Palpanas: *Web-scale, Schema-Agnostic, End-to-End Entity Resolution*. Tutorial at the ACM International Web Conference (WWW) (2018)

Entity Clustering Algorithms

- Connected Components (Transitive Closure)
- CENTER
- **MERGE-CENTER**

It performs similar to CENTER, but merges two clusters c_i and c_j whenever a profile similar to the center node of c_j is in the cluster c_i, i.e., a profile that is similar to the center of the cluster c_i is similar to the center of c_j.



[1] G. Papadakis, T. Palpanas: *Web-scale, Schema-Agnostic, End-to-End Entity Resolution*. Tutorial at the ACM International Web Conference (WWW) (2018)

JedAI: An End-to-End ER Framework

JedAI can be used in three ways:

1. As an **open-source library** that implements numerous state-of-the-art methods for all steps of an established end-to-end ER workflow.
2. As a **desktop application** for ER with an intuitive Graphical User Interface that is suitable for both expert and lay users.
3. As a **workbench** for comparing all performance aspects of various (configurations of) end-to-end ER workflows.

JedAI

[1] G. Papadakis, G. Mandilaras, L. Gagliardelli, G. Simonini, E. Thanos, G. Giannakopoulos, S. Bergamaschi, T. Palpanas, M. koubarakis: *Three-dimensional Entity Resolution with JedAI*. Information Systems (IS) 93: 101565 (2020)

- Project website: http://jedai.scify.org
- GitHub repository: https://github.com/scify/JedAIToolkit

138

JedAI: An End-to-End ER Framework

JedAI implements the following **schema-agnostic, end-to-end workflow** for both Clean-Clean and Dirty ER:

Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7
Data Reading →	**Block Building** →	**Block Cleaning** →	**Comparison Cleaning** →	**Entity Matching** →	**Entity Clustering** →	**Evaluation & Storing**
Reads files containing the entity profiles and the golden standard.	Creates overlapping blocks.	Optional step that cleans blocks from useless comparisons (repeated, superfluous).	Optional step that operates on the level of individual comparisons to remove the useless ones.	Executes all retained comparisons.	Partitions the similarity graph into equivalence clusters.	Stores and presents performance results w.r.t. numerous measures.

JedAI: An End-to-End ER Framework

Magellan

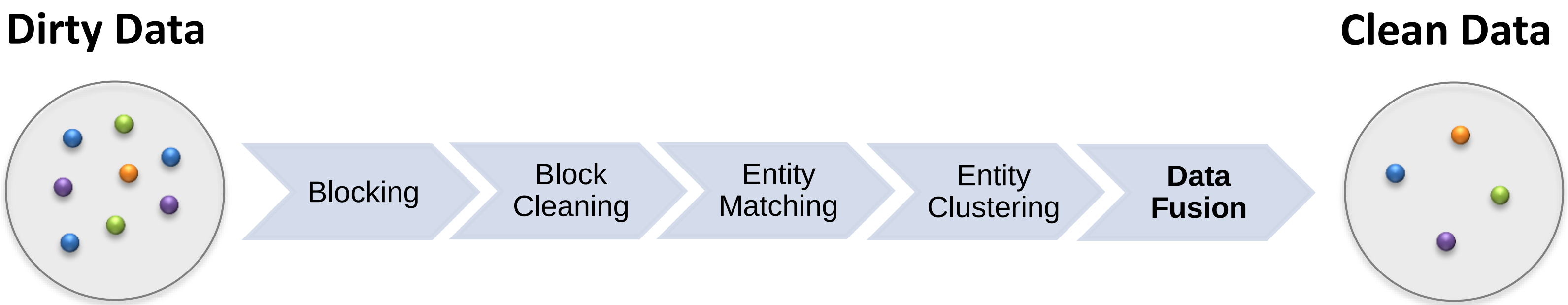
- × limited variety of (blocking) methods
- × restricted to relational data only
- × targeted to expert users, focusing on development of tailor-made methods
- × offers command-line interface, no GUI

JedAI

- ✓ rich variety available methods for every step in the end-to-end workflow
- ✓ applies to both structured and non-structured data
- ✓ hands-off functionality through default configuration of every method, but also extensible
- ✓ intuitive GUI with guidelines even for novice users
- ✓ multi-core execution (SPARKER)

DATA FUSION

From clusters of matches to single clean representative records



Data Fusion

- Obtain a **single record representative of the entity** from each cluster of matching records
- Application of a **conflict resolution function** (i.e., **aggregation function**) to each attribute

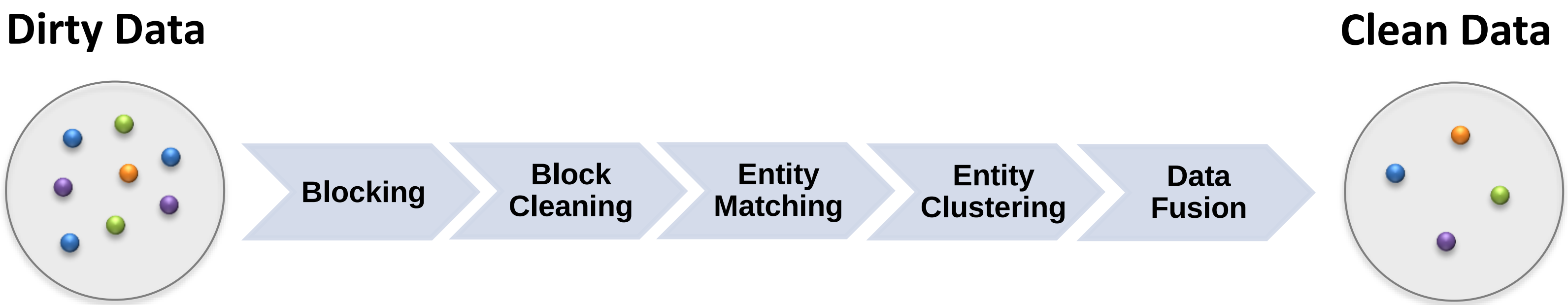
		Majority Voting	MAX	AVG

Brand	Model	Megapixels	Price ($)
canon	eos 400d	10.0	185.00
cannon	rebel xti	10.1	150.00
canon	eos 400d	10.1	115.00

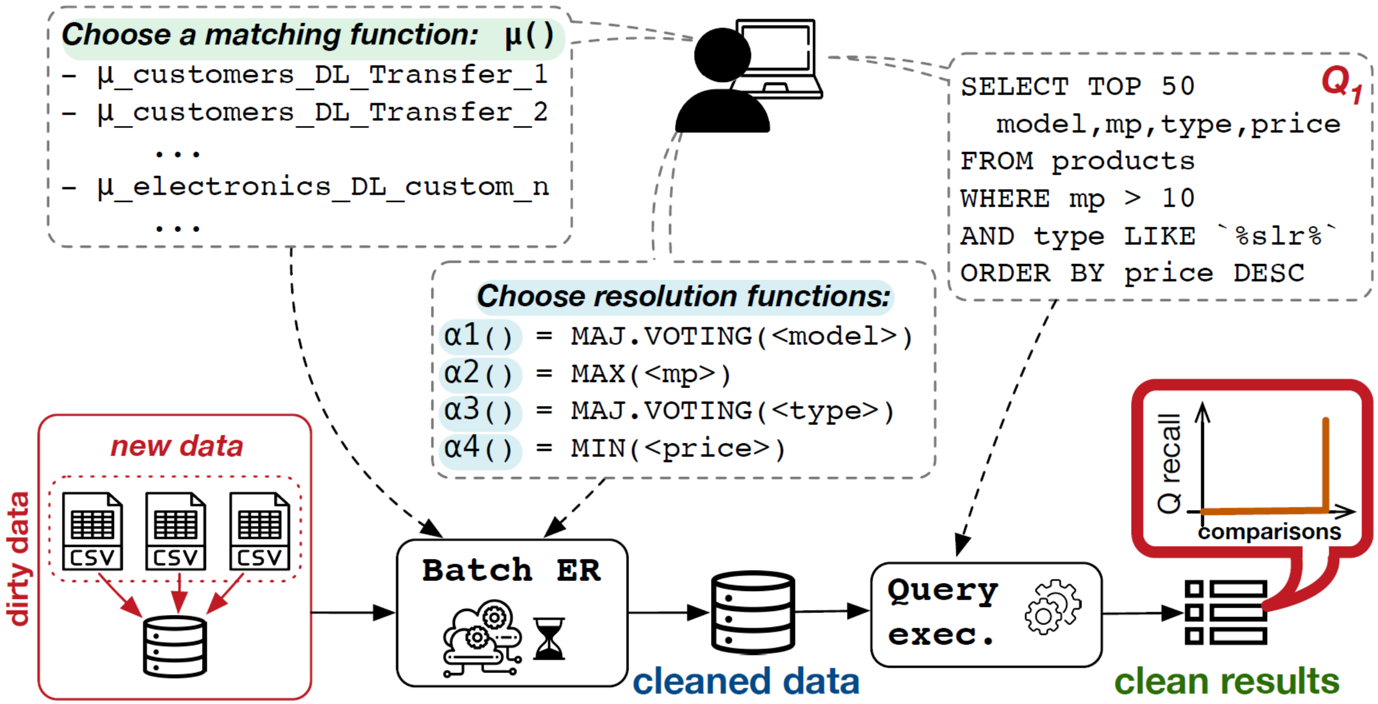
Brand	Model	Megapixels	Price ($)
canon	eos 400d	10.1	150.00

BEYOND TRADITIONAL BATCH ER

From the established ER pipeline to pay-as-you-go approaches



Batch ER



Useless comparisons (produce entities that will surely not appear in the result of the query)

Our time, resources and affordable costs (e.g., pay-as-you-go in cloud) are often **limited**

Beyond Traditional Batch ER: Query-Driven Approaches

Reduce the **number of comparisons** needed to answer the query (discard useless candidates)

Batch approaches, but fewer comparisons to be performed

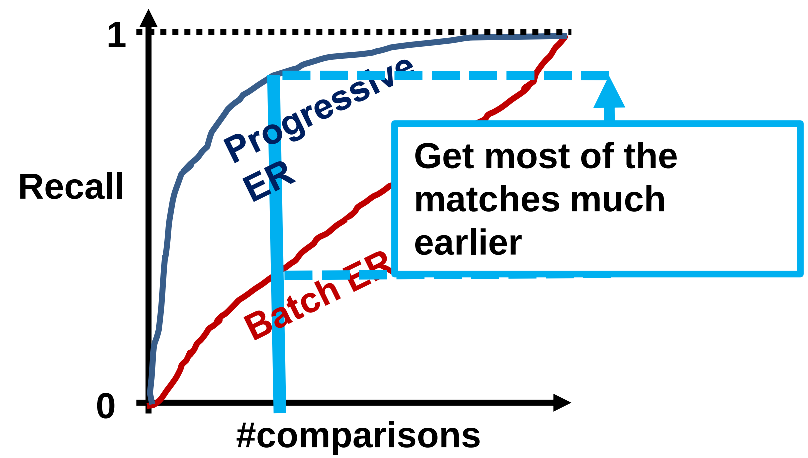
[1] H. Altwaijry et al.: *Query-Driven Approach to Entity Resolution*.
Proceedings of the VLDB Endowment (PVLDB) 6(14): 1846-1857 (2013)

[2] H. Altwaijry et al.: *QuERy: A Framework for Integrating Entity Resolution with Query Processing*.
Proceedings of the VLDB Endowment (PVLDB) 9(3): 120-131 (2015)

Beyond Traditional Batch ER: Progressive Approaches

Maximize the number of retrieved **matches** in a limited amount of time (driven by **matching likelihood**) – ***See the appendices for more details***

- Use cases:
  - Limited resources
  - Exploratory ER
- Tries to approximate the **optimal comparison order**
  - Candidate pairs are ordered
  - Applies the Matching Function following that order
- Maximize **Progressive Recall**
- **Tradeoff:** $t_{start\text{-}progressive} > t_{start\text{-}batch}$



[1] S. Whang et al.: *Pay-As-You-Go Entity Resolution*.
IEEE Transactions on Knowledge and Data Engineering (TKDE) 25(5): 1111-1124 (2013)

[2] T. Papenbrock et al.: *Progressive Duplicate Detection*.
IEEE Transactions on Knowledge and Data Engineering (TKDE) 27(5): 1316-1329 (2015)

[3] G. Simonini, G. Papadakis, T. Palpanas, S. Bergamaschi: *Schema-agnostic Progressive Entity Resolution*. IEEE International Conference on Data Engineering (ICDE): 53-64 (2018)

A Common Real-World Data Science Scenario

A data scientist wants to query a **dirty** dataset of products acquired from the Web (**data exploration**)

- Information needs and business priorities → Expressed using a **query**
- Data changes with a high frequency → **Time constraints**

Pay-as-you-go: obtain the resulting entities as soon as possible and return them as soon as they are available

A common situation: data exploration, trading, on-demand extraction and cleaning in data lakes, Web data, etc.

id	brand	model	type	mp	price
R1	canon	eos 400d	dslr	10.1	165.00
R2	canon	rebel xti	reflex	1.01	185.00
R3	eos canon	400 d	dslr	10.1	115.00
R4	nikon	d-200	dslr	-	150.00
R5	nikon	d200	-	10.2	130.00
R6	nikon	d40	digital	-	100.00
R7	kodak	dc3200	dslr	1.3	75.00
R8	kodak	dc-3200	-	1.3	80.00

```
SELECT TOP 50 brand, model, type, mp, price
FROM products
WHERE type LIKE '%slr%'
AND mp > 10
ORDER BY price DESC
```

The Shortcomings of the Existing Approaches

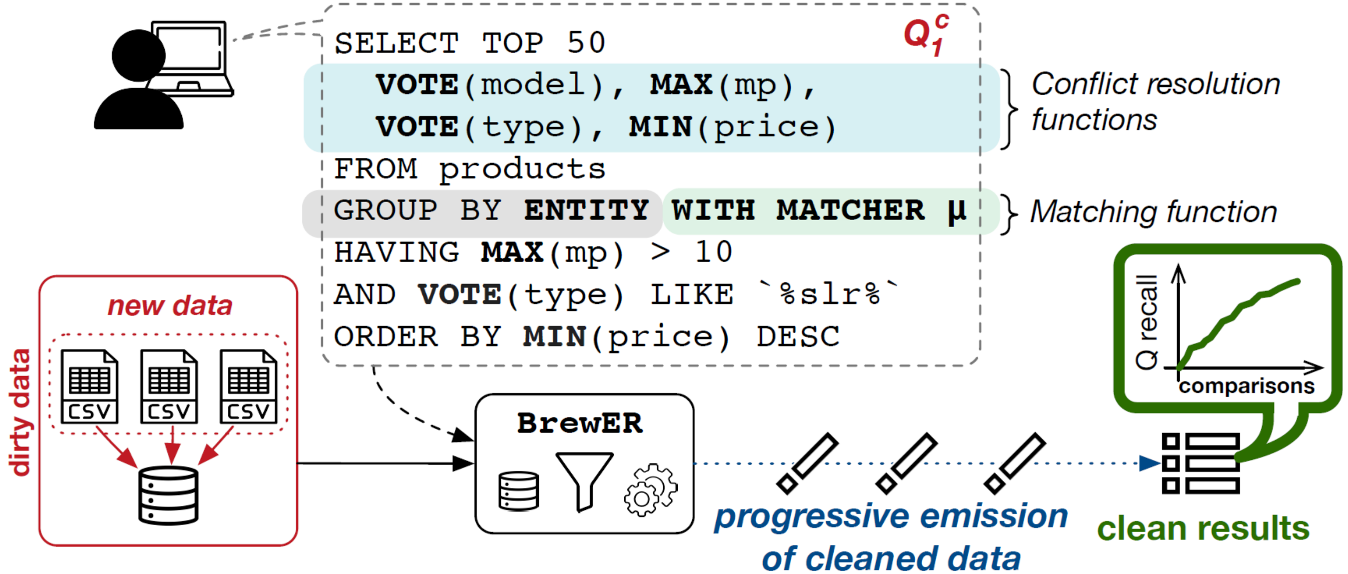
PROGRESSIVE ENTITY RESOLUTION

- **Maximize** the number of retrieved **matches** in a limited amount of time
- Driven by **matching likelihood**, do not support **user-defined priorities**
- Focus on matches, not on entities (**approximate result**, i.e., partially resolved entities)

QUERY-DRIVEN ENTITY RESOLUTION

- Reduce the **number of comparisons** needed to answer the query (discard useless candidates)
- **Batch** approach: not designed for a progressive execution, do not support the **ORDER BY** clause

BrewER: Entity Resolution On-Demand



Agnostic approach to blocking and matching functions

Clean queries on dirty data

- **QUERY-DRIVEN**: ER only on the portion of dataset useful to answer the query (according to the HAVING clauses)
- **PROGRESSIVE**: return the entities in the result in the right order as soon as they are obtained (according to the ORDER BY clause)

[1] G. Simonini, L. Zecchini, S. Bergamaschi, F. Naumann: *Entity Resolution On-Demand*. Proceedings of the VLDB Endowment (PVLDB) 15(7): 1506-1518 (2022)

VLDB 2022

BrewER in action: (1) Filtering the blocks

id	brand	model	type	mp	price
R1	canon	eos 400d	dslr	10.1	165.00
R2	canon	rebel xti	reflex	1.01	185.00
R3	eos canon	400 d	dslr	10.1	115.00
R4	nikon	d-200	dslr	-	150.00
R5	nikon	d200	-	10.2	130.00
R6	nikon	d40	digital	-	100.00
R7	kodak	dc3200	dslr	1.3	75.00
R8	kodak	dc-3200	-	1.3	80.00

```
SELECT TOP 50 VOTE(brand), VOTE(model), VOTE(type),
              MAX(mp), AVG(price)
FROM products
GROUP BY ENTITY WITH MATCHER μ
HAVING VOTE(type) LIKE '%slr%'
AND MAX(mp) > 10
ORDER BY AVG(price) DESC
```

«canon»
R1, R2, R3

«nikon»
R4, R5, R6

«kodak»
R7, R8

Which blocks can produce useful entities?

R1	canon	eos 400d	dslr	10.1	165.00
R2	canon	rebel xti	reflex	1.01	185.00
R3	eos canon	400 d	dslr	10.1	115.00

R4	nikon	d-200	dslr	-	150.00
R5	nikon	d200	-	10.2	130.00
R6	nikon	d40	digital	-	100.00

R7	kodak	dc3200	dslr	1.3	75.00
R8	kodak	dc-3200	-	1.3	80.00

BrewER in action: (2) Initializing the priority queue

R1	canon	eos 400d	dslr	10.1	165.00
R2	canon	rebel xti	reflex	1.01	185.00
R3	eos canon	400 d	dslr	10.1	115.00

R4	nikon	d-200	dslr	-	150.00
R5	nikon	d200	-	10.2	130.00
R6	nikon	d40	digital	-	100.00

SELECT TOP 50 VOTE(brand), VOTE(model), VOTE(type),
MAX(mp), AVG(price)
FROM products
GROUP BY ENTITY WITH MATCHER μ
HAVING VOTE(type) LIKE '%slr%'
AND MAX(mp) > 10
ORDER BY AVG(price) DESC

head

R2	R1	R3	R4	R5	R6
185.00	165.00	115.00	150.00	130.00	100.00

Priority queue

BrewER in action: (3) Iterating on the priority queue

Iteration 1

head

R2	R1	R3	R4	R5	R6
185.00	165.00	115.00	150.00	130.00	100.00

head

E1	R4	R5	R6
155.00	150.00	130.00	100.00

Iteration 2

head

E1	R4	R5	R6
155.00	150.00	130.00	100.00

id	brand	model	type	mp	price
E1	canon	eos 400d	dslr	10.1	155.00

✅

Iteration 3

head

R4	R5	R6
150.00	130.00	100.00

head

E2	R6
140.00	100.00

Iteration 4

head

E2	R6
140.00	100.00

id	brand	model	type	mp	price
E2	nikon	d200	dslr	10.2	140.00

✅

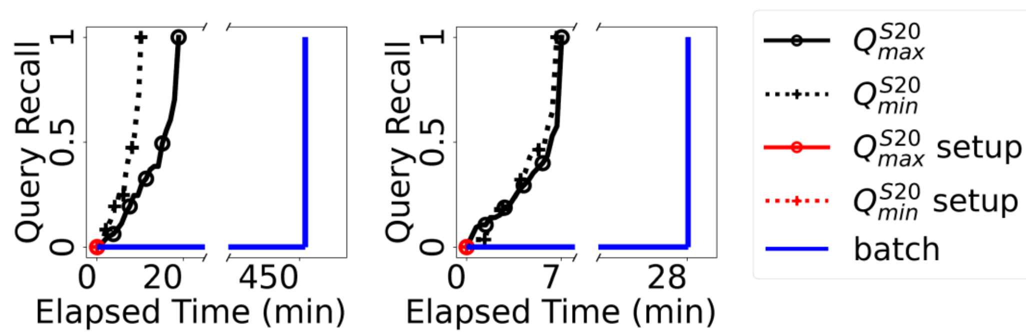
BrewER in action: clean results emitted progressively

id	brand	model	type	mp	price
R1	canon	eos 400d	dslr	10.1	165.00
R2	canon	rebel xti	reflex	1.01	185.00
R3	eos canon	400 d	dslr	10.1	115.00
R4	nikon	d-200	dslr	-	150.00
R5	nikon	d200	-	10.2	130.00
R6	nikon	d40	digital	-	100.00
R7	kodak	dc3200	dslr	1.3	75.00
R8	kodak	dc-3200	-	1.3	80.00

id	brand	model	type	mp	price
E1	canon	eos 400d	dslr	10.1	165.00
E2	nikon	d200	dslr	10.2	140.00

```
SELECT TOP 50 VOTE(brand), VOTE(model), VOTE(type),
       MAX(mp), AVG(price)
FROM products
GROUP BY ENTITY WITH MATCHER μ
HAVING VOTE(type) LIKE '%slr%'
AND MAX(mp) > 10
ORDER BY AVG(price) DESC
```

1. **Preliminary filtering of the blocks**
2. **Initialization of the priority queue**
3. **Loop on the priority queue**



(a) SIGMOD20 (BL) (b) SIGMOD21

Outline

- Who I Am
- From Data Integration to Big Data Integration
- Entity Resolution (a.k.a. Record Linkage)
- **Privacy-Preserving Record Linkage (PPRL)**
- PPRL in MOMIS

General Data Protection Regulation (GDPR)

Whenever sensitive personal data about individuals are to be integrated, privacy and confidentiality have to be considered.

Data protection in Europe is set off by the European General Data Protection Regulation (**GDPR**) which became active in May 2018 and is a comprehensive legal framework that sets guidelines for the collection and processing of personal information from individuals who live in the European Union (EU).

An Appropriate technique to implement data-protection principles in a effective manner is Pseudonymization.

This applies to the use of tolerant **privacy-preserving techniques** to create **pseudonyms** of the data to be integrated.

Privacy-Preserving Record Linkage (PPRL)

Identifying and linking the **records (profiles)** that refer to the **same real-world object (entity),** across several data sources held by different parties, in a manner that prevents both the computation and the output of the computation from revealing (to any **internal parties** involved in the process and **external adversaries**) any private sensitive information about the entities represented in the data.

Pseudonym	Blood test
1234 (Alice)	$BloodTest_A$
5678 (Bob)	$BloodTest_B$

Hospital 1

+

Name	ECG
3456 (Alice)	ECG_A
7890 (Carl)	ECG_C

Hospital 2

Pseudonym	Blood test	ECG
???? (Alice)	$BloodTest_A$	ECG_A
???? (Bob)	$BloodTest_B$	
???? (Carl)		ECG_C

PPRL & Databases

In a privacy-aware setting, databases may contain different kinds of data that require to be handle in different way to ensure protection of identity and sensitive data of individuals:

- **Personally Identifiable Information (PII):** an attribute which is a unique identifier for each entity of a population; e.g., Personal Identification Number (PID).
- **Quasi-identifiers (QID):** an attribute which contains information that potentially identifies record owners when joined with other information; e.g., names, dates of birth, addresses.
- **Sensitive Data:** an attribute which represents sensitive individual-specific information that must be protected against privacy disclosure; e.g., disease or income, religion or political opinions.
- **Non-Sensitive Data:** an attribute that contains information which does not deserve protection; e.g., metadata.

Classifying data according to identifiability in a real scenario is not an easy task because different kinds of data can overlap.

PPRL requirements

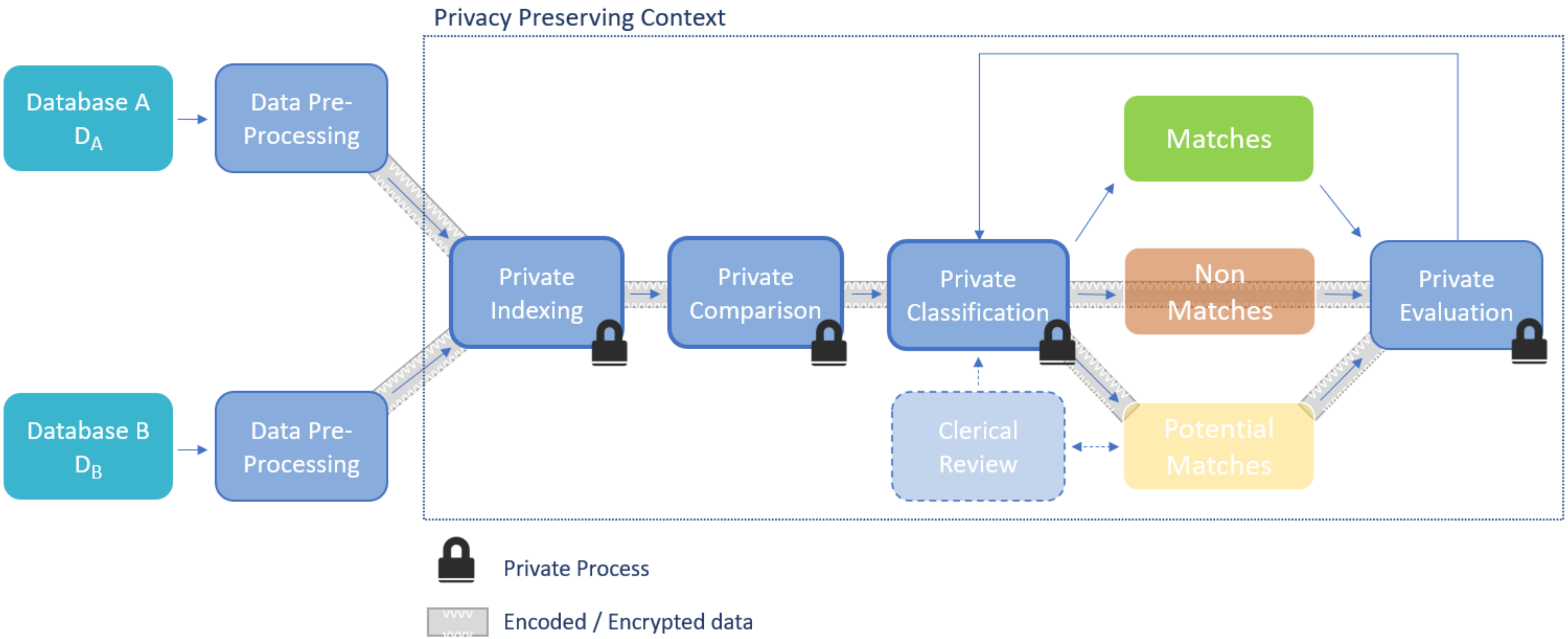
The PPRL requirements are:

- **Allow linkage of data related to the same subject:**
  achieve high linkage quality on encrypted data even if original data (plaintext) has typos, misspellings and other errors.

- **Avoid re-identification of subject's identity**:
  In order to meet the GDPR requirements, both internal and external parties should not be able to derive subject’s identities without additional information.
  A basic measure is to encrypt data at local sources and prevent unencrypted data leaves the local storage.

- **Allow the possibility of re-identification in particular cases**:
  It should be possible to inform a subject about data breach or relevant research results. Therefore, the possibility of re-identification is needed.

The “Standard” PPRL Approach



The "Standard" PPRL Approach

- **Data pre-processing and masking** is crucial for linkage quality outcomes; it resolves inconsistencies in data.

Data masking is conducted by encrypting sensitive data and identifiers in such a way that only limited information about record is revealed.
This step can be conducted independently at each data source; however, some exchange of information between the parties about what data pre-processing and masking approaches is required.

- **Blocking** (indexing or filtering) is crucial for scalability; it reduces the number of comparisons that need to be conducted between records and generates candidate record pairs (or sets).

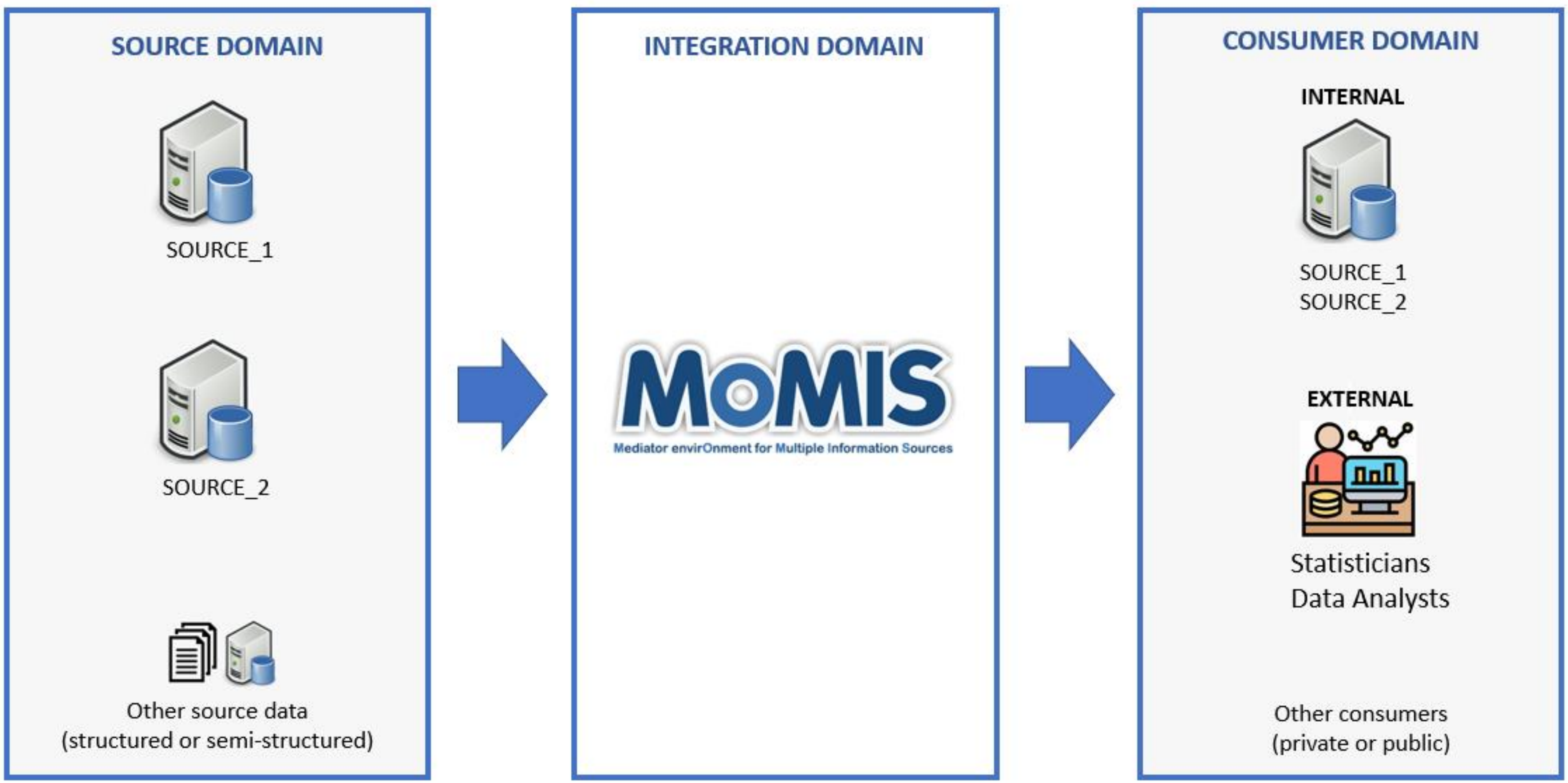
- **Comparison and Classification** compare candidate record pairs and classify them into match or not match.

Outline

- Who I Am
- From Data Integration to Big Data Integration
- Entity Resolution (a.k.a. Record Linkage)
- Privacy-Preserving Record Linkage
- PPRL in MOMIS

PPRL in MOMIS

PPRL of two different sources (Source domain) within the healt domain, using Momis as Integration domain, in order to provide aggregated results to Consumer Domain



PPRL in MOMIS

- RL could be performed on a global identifier (PII) or set of QIDs
- A QID must be present in all local sources and set/subset of QIDs must uniquely identify each subject
- Sensitive Data can be present both in a single source and in all local sources (data fusion)
- It is importatant to take into account the trade-off between privacy (possibility of re-identification by adversaries) and utility (analysis capacity for research purpose).

SOURCE_1
Name
Surname
Birthday
Nationality
Gender
Residence
Disease
Therapy

QIDs

?

Sensitive Data

SOURCE_2
NAME SURNAME
BIRTHDAY
NATIONALITY
SEX
WEIGHT
MEDICAL TEST
DISEASE
DRUG

PPRL in MOMIS

- For example Weight can be used to conduct the analysis separately and produce aggregated results (e.g. statistical analysis of diabetic patients depending on weight), thus is classified as sensitive data.
- Residence is suitable for further analysis but the analysis phase requires that consumers uses the Sensitive data in plaintext and Residence could be used to re-identify the subject, so it will be excluded.

SOURCE_1
Name
Surname
Birthday
Nationality
Gender
~~Residence~~
Disease
Therapy

QIDs

Sensitive Data

SOURCE_2
NAME SURNAME
BIRTHDAY
NATIONALITY
SEX
WEIGHT
MEDICAL TEST
DISEASE
DRUG

PPRL Mapping Table

- The MOMIS Mapping Table is used to define the Global attributes between the two global schema, the type of attributes in the PPRL process and the Data Trasformation Functions that local sources should apply in Pre-processing Phase to facilitate comparison and data aggregation.

	GLOBAL ATTRIBUTE	SOURCE_1	SOURCE_2
QIDs	Name	Name	NAME SURNAME
	Surname	Surname	NAME SURNAME
	Birthday	Birthday	BIRTHDAY
	Nationality	Nationality	NATIONALITY
	Gender	Gender	SEX
Sensitive Data	Weight		WEIGHT
	Medical test		MEDICAL TEST
	Disease	Disease	DISEASE
	Therapy	Therapy	DRUG

Data Pre-processing

- Data transformation (pre-processing) performed at local sources

S1	Name	Surname	Birthday	Nationality	Gender	Residence	Disease	Therapy
S1_1	john	Smith	20th May 1996	american	Male	West Main Street 29, 12068, Fonda, NY, New York	Diabetes mellitus	acarbose (Precose)
S1_2	Rossi	paolo	1° Luglio 1940	italiano	Maschio	Via Torriani Napo 29, 20124 Milano MI	Morbo di Alzheimer	memantine (Namenda)
S1_3	Katia	Anderson	13th June 1966	british	Female	West Main Street 29, 12068, Fonda, NY, New York	Breast cancer	Radiation Therapy

↓ Data Pre-processing Functions

S1	Name	Surname	Birthday	Nationality	Gender	Disease	Therapy
S1_1	John	Smith	20/03/1996	USA	M	Diabetes	acarbose (Precose)
S1_2	Rossi	Paolo	01/07/1940	Italy	M	Alzheimer	memantine (Namenda)
S1_3	Katia	Anderson	13/06/1996	UK	F	Breast cancer	Radiation Therapy

Data Pre-processing

- Data transformation (pre-processing) performed at local sources

S2	NAME SURNAME	BIRTHDAY	NATIONALITY	SEX	WEIGHT	MEDICAL TEST	DISEASE	DRUG
S2_1	John Smith	20/03/1996	USA	M	98 kg	A1C test	Diabetes	Precose
S2_2	Johnathan Monette	03/12/1954	UK	M	68 kg	CDR Test	Alzheimer	AChE
S2_3	Kathy Anderson	12/06/1996	UK	F	50 kg	CT scan	Cancer	Chemotherapy

↓ Data Pre-processing Functions

S2	Name	Surname	Birthday	Nationality	Gender	Weight (kg)	Medical Test	Disease	Therapy
S2_1	John	Smith	20/03/1996	USA	M	98	A1C test	Diabetes	Precose
S2_2	Johnathan	Monette	03/12/1954	UK	M	68	CDR Test	Alzheimer	AChE
S2_3	Kathy	Anderson	12/06/1996	UK	F	50	CT scan	Cancer	Chemotherapy

Context Specific Pseudonimyzation

- Pseudonimization is performed at local sources
- Pseudonymization (ex. hash encoding) transforms QIDs (name + surname + ...) and the context (+ source) into local pseudonyms (LP).

Ex JohnSmithSource1 -> d57199db0a8b0fce530d9c48413d4e32.

S1	Name	Surname	....
S1_1	John	Smith	....
S1_2	Rossi	Paolo	....
S1_3	Katia	Anderson	....

S2	Name	Surname	...
S2_1	John	Smith	...
S2_2	Johnathan	Monette	...
S2_3	Kathy	Anderson	...

Context Specific Pseudonimization

S1	Local Pseudonym
LP1_1	d57199db0a8b0fce530d9c48413d4e32
LP1_2	36a9263eb76656786fe8b29b4dd5c710
LP1_3	f065eaa2087ce8acc043a9d6f8798953

S2	Local pseudonym
LP2_1	1ae7c92ab815317a63711cd6d3a83632
LP2_2	74868a832d2e6604da3645971d73f333
LP2_3	011b76605f7a576c50252c119bcb94c0

Phonetic Hashing

- Phonetic encoding is performed at local source
- Phonetic encoding (based on pronunciation) avoids deduplication of one and the same patient and allows scalability. Ex. Soundex transforms QIDs (name) + (surname) into a string that can be used for comparison (or like blocking key) -> QID_PHON. Ex JohnSmith -> J500S530

S1	Name	Surname
S1_1	John	Smith
S1_2	Rossi	Paolo
S1_3	Katia	Anderson

S2	Name	Surname
S2_1	John	Smith
S2_2	Johnathan	Monette
S2_3	Kathy	Anderson

S1	Phonetic QID
PH1_1	J500S530
PH1_2	R200P400
PH1_3	K300A536

S2	Phonetic QID
PH2_1	J500S530
PH2_2	J535M530
PH2_3	K300A536

Asymmetric encryption

- In order to allow re-identification a local source additionally encrypt QIDs with asymmetric encryption -> QID_CRYPT
- Asymmetric encryption uses a key pair (private and public) so that anyone can encrypt a message using the public key while only a third trusted part (the MOMIS mediator) can decrypt it with the private key (private key is kept secret and is never transmitted).

Ex. JohnSmith ->
Owi8OHkdDaRyU0PNfI/9yWyPaKkjHKUaH0mIE2JB6yNL/AZcroEzFuFo5
RsjFcisLELSDgD8ZuM0F9I8taqhyQ==

Public key:
MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBALeNFp2vUK3tRFdb4eibVgyw8UJ5Y2eBUOyB0vpFLxhuE3FJZRE4RxpMjY
d5c4kR9w+zYYWa62bCvJKFPxRILPMCAwEAAQ==

Private key:
MIIBVQIBADANBgkqhkiG9w0BAQEFAASCAT8wggE7AgEAAkEAt40Wna9Qre1EV1vh6JtWDLDxQnljZ4FQ7IHS+kUvGG4T
cUllEThHGkyNh3lziRH3D7NhhZrrZsK8koU/FEgs8wIDAQABAkEAhXuJMutH1PRzesRLKYmtrlUPXrRAYglc/GH9OBwP/8buf
Xn7PMAzV7R53u7MHUR9tO9Ds9mXcBqCSTFV5VqaoQIhAPuIL3KhQVIxPvHL5ubqVIhlMCM6Oxrx9Cyz1dXQOwWVAiEA
us/Ec8l59BKr7Tr/32zIarn/np5oq454vGk0U1YNtmcCIQD5E11w1K/7dRqQk8pdxZPZ0OG/MI2Q3CFgFuDcLqwTlQIgY3S3x1k
tdzb1l3BAx2d37/IkWANIAIXyW4S7Gd8Hn+MCIFTpYUmTZykNAEf/7nO4e/VGqfyXcmhw6PNOc4HgP9Zs

Data acquisition

Local sources send LP, Sensitive data, QID_PHON and QID_CRYPT to MOMIS (through an encrypted communication channel).

MOMIS performs PPRL using QID_PHON and assigns LP, QID_PHON and QID_CRYPT to a Global Pseudonym (GP). -> Matching Table

When inserting a new patient, MOMIS checks if the patient is present in other contexts to which the new LP can be linked, otherwise it creates a new GP.

Metadata database (matching table)

Global Pseudonyms	LP1	LP2	QID_CRIPY	QID_PHON
GP_1	LP1_1	LP2_1	...	J500S530
GP_2	LP1_2		...	R200P400
GP_3		LP2_2	...	J535M530
GP_4	LP1_3	LP2_3	...	K300A536

Matching Table & Sensitive Data

- It is important that the Sensitive Data and their LPs are stored in a different location than the metadata describing the correlation between the different LPs. This reduces the hazards of unintended re-identification and increases data security.

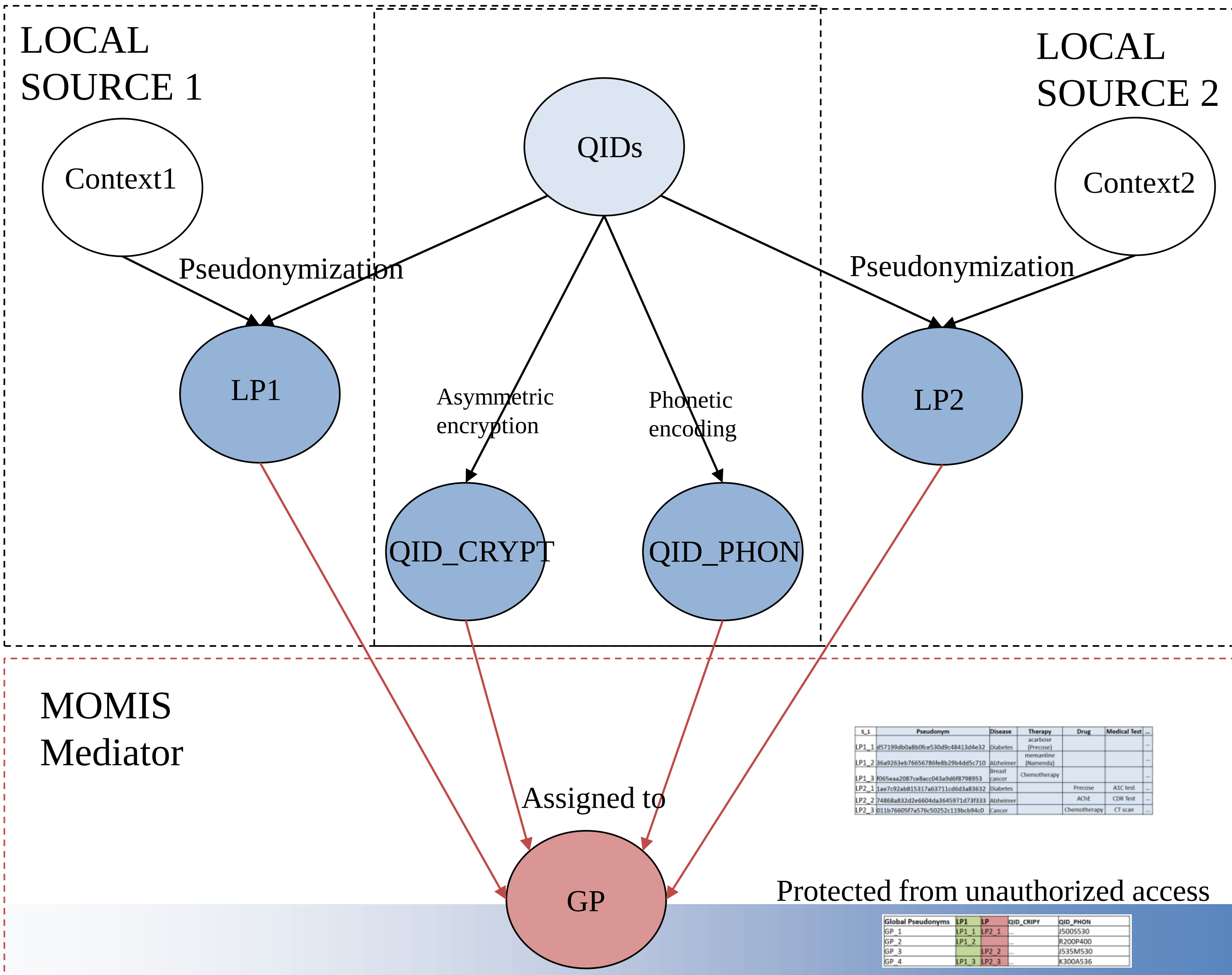
Metadata database (Matching Table)

Global Pseudonyms	LP1	LP2	QID_CRIPY	QID_PHON
GP_1	LP1_1	LP2_1	...	J500S530
GP_2	LP1_2		...	R200P400
GP_3		LP2_2	...	J535M530
GP_4	LP1_3	LP2_3	...	K300A536

Repository (Sensitive Data)

	Pseudonym	Disease	Therapy	Weight (kg)	Medical Test
LP1_1	d57199db0a8b0fce530d9c48413d4e32	Diabetes	acarbose (Precose)		
LP1_2	36a9263eb76656786fe8b29b4dd5c710	Alzheimer	memantine (Namenda)		
LP1_3	f065eaa2087ce8acc043a9d6f8798953	Breast cancer	Chemotherapy		
LP2_1	1ae7c92ab815317a63711cd6d3a83632	Diabetes	Precose	98	A1C test
LP2_2	74868a832d2e6604da3645971d73f333	Alzheimer	AChE	68	CDR Test
LP2_3	011b76605f7a576c50252c119bcb94c0	Cancer	Chemotherapy	50	CT scan

METADATA GENERATION



S_1	Pseudonym	Disease	Therapy	Drug	Medical Test	...
LP1_1	d57199db0a8b0fce530d9c48413d4e32	Diabetes	acarbose (Precose)			...
LP1_2	36a9263eb76656786fe8b29b4dd5c710	Alzheimer	memantine (Namenda)			...
LP1_3	f065eaa2087ce8acc043a9d6f8798953	Breast cancer	Chemotherapy			...
LP2_1	1ae7c92ab815317a63711cd6d3a83632	Diabetes		Precose	A1C test	...
LP2_2	74868a832d2e6604da3645971d73f333	Alzheimer		AChE	CDR Test	...
LP2_3	011b76605f7a576c50252c119bcb94c0	Cancer		Chemotherapy	CT scan	...

Global Pseudonyms	LP1	LP	QID_CRIPY	QID_PHON
GP_1	LP1_1	LP2_1	...	J500S530
GP_2	LP1_2		...	R200P400
GP_3		LP2_2	...	J535M530
GP_4	LP1_3	LP2_3	...	K300A536

Exported data

- An aggregation of Sensitive data from the repository is initiated by a query carried out by the user (Data Aggregation). The desired data is linked and collated by MOMIS utilising the Matching Table.

Matching table | **Sensitive data**

Global Pseudonyms	LP1	LP2	QID_CRIPY	QID_PHON	Disease	...
GP_1	LP1_1	LP2_1	...	J500S530	Diabetes	...
GP_2	LP1_2		...	R200P400	Alzheimer	...
GP_3		LP2_2	...	J535M530	Alzheimer	...
GP_4	LP1_3	LP2_3	...	K300A536	Breast cancer	...

- Exported data will be assigned to a new pseudonym.

Exported Pseudonyms	Disease	Therapy	Weight (kg)	Medical Test
EP_1	Diabetes	acarbose (Precose)	98	A1C test
EP_2	Alzheimer	memantine (Namenda)		
EP_3	Alzheimer	AChE	68	CDR Test
EP_4	Breast cancer	Chemotherapy	50	CT scan

- When data is to be transferred to another system, the re-usage of an existing identifier from source systems should be avoided.

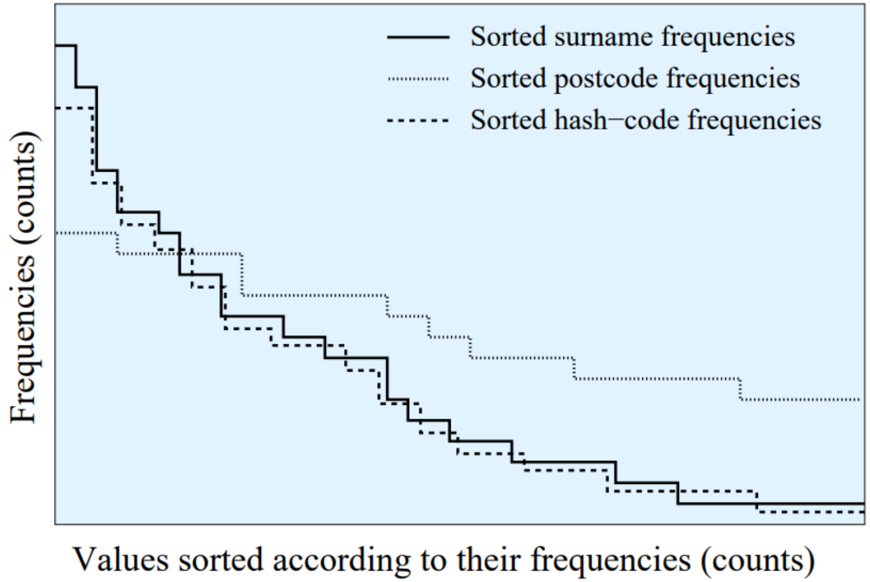
Attack methods

- **Dictionary attacks**

  An adversary encodes a list of known values using existing encoding functions until a matching encoded value is identified (a keyed encoding approach, like HMAC, can help prevent this attack)

- **Frequency attacks**

  Frequency distribution of encoded values is matched with the distribution of known values.

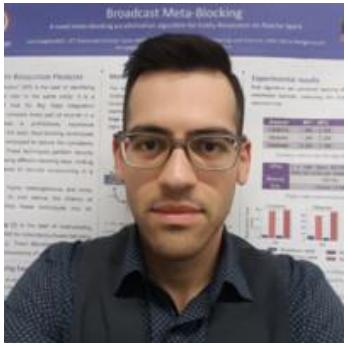


- **Collusion**

  A set of parties collude with the aim to learn about another party's data.

Thanks to…

Domenico
BENEVENTANO



Luca
GAGLIARDELLI

Giovanni
SIMONINI

Lisa
TRIGIANTE

Luca
ZECCHINI

THANK YOU
FOR THE ATTENTION!