Deep-In Shallow-Out Data Cleaning

Nan Tang

Is Real-life Data Dirty?


$121,900
TAXES DUE:
$8,122,562.19
Regards,
J Murphy
Jim Murphy,
County Treasurer
$400,000,000

Dell Ordered To Honor $15 Set To Lose $19 Million

Lifestyle July 1, 2009

Dell E1909W Widescreen Flat Panel Monitor

Dell E1909W 19" F

For business users who war
Dell E1909W offers:

- Generous screen sp
- Multiple connectivity
- High levels of energ

Gallery

New Canadian research raises concerns over number, types of transfusion errors

In all, a total of 15,134 errors were reported over 72 months. For every error that harmed a patient there were 657 errors that were detected and intercepted before the blood could reach the patient. "Wrong blood in tube" — blood drawn from the wrong patient for matching — occurred once in every 10,250 samples collected.

Problem for US Economy?

Problem for US Economy?

3000,000,000 $ per year

2X the national deficit

Problem for US Economy?

3000,000,000 $ per year

2X the national deficit

14% healthcare spending waste

Problem for US Economy?

3000,000,000 $ per year

2X the national deficit

14% healthcare spending waste

15-20% operating budget waste

Major Players

FORTUNE DATA

Big data's dirty problem

Inaccuracies, misspellings, and obsolete information makes achieving the big data utopia a slog for businesses and researchers.

A constellation of companies offer software and services for cleaning data. They range from technology giants like IBM (**IBM ▼ -0.08%**) and SAP (**SAP ▼ -0.68%**) to big data and analytics specialists like Cloudera and Talend Open Studio. A legion of start-ups are also trying to get a toehold as data janitors including Trifacta, Tamr, and Paxata.

Data Cleaning

Data

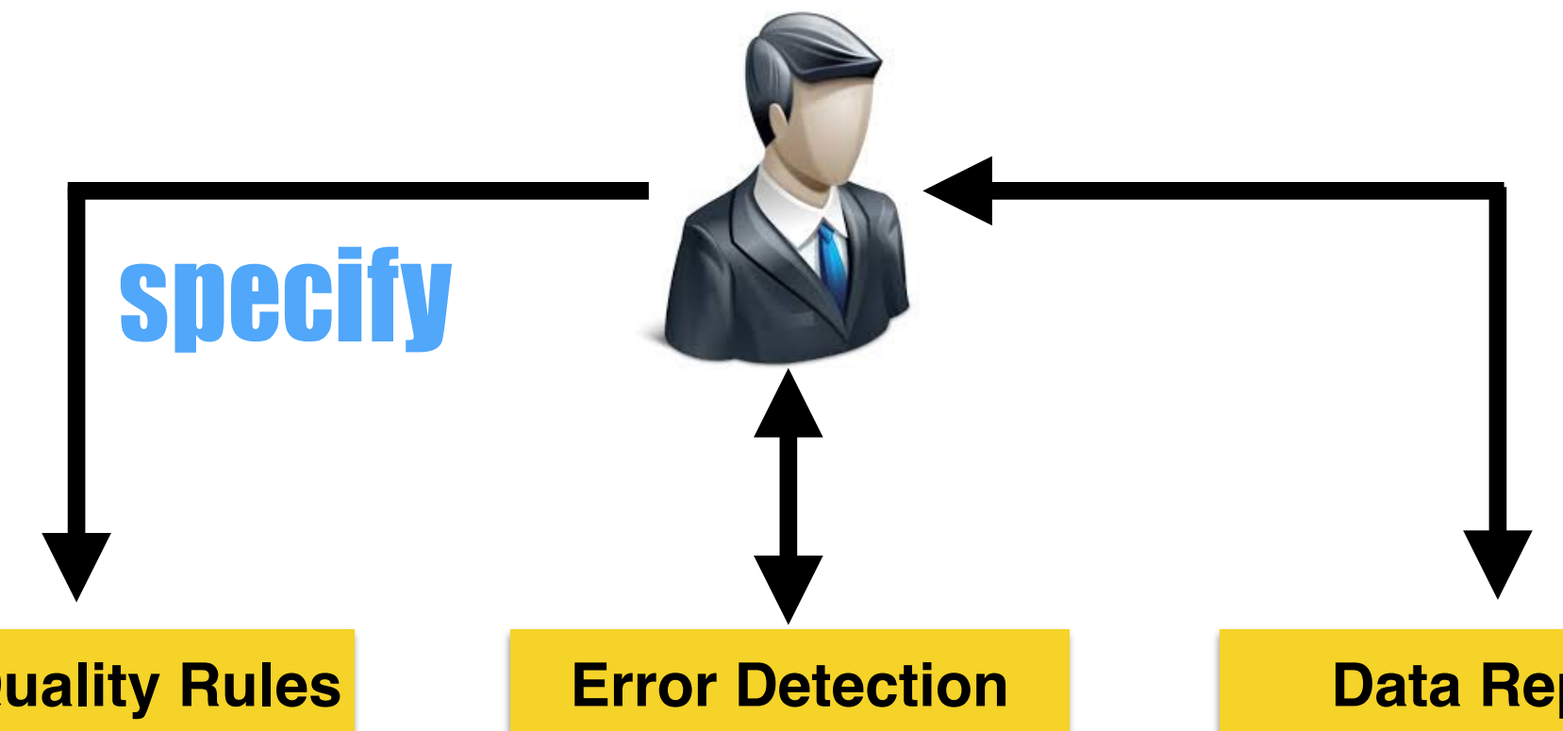
Data Quality Rules

1. You SHALL!
2. You WILL!
3. You MUST!

Error Detection

Data Repair

Data Cleaning



Data

Data Quality Rules

1. You SHALL!
2. You WILL!
3. You MUST!

Error Detection

Data Repair

Data Quality Rules

Integrity Constraints

- Functional Dependencies *(E. F. Codd 1972)*

Integrity Constraints

- Functional Dependencies *(E. F. Codd 1972)*

FD: country -> capital

country	capital	population	language
China	Beijing	10,000,000,000	Mandarin
China	Shanghai	11,000,000,000	Mandarin
Netherlands	Amsterdam	168,000,000	Dutch
Netherlands	Den Hagg	170,000,000	Dutch
Spain	Barcelona	472,700,000	Spanish

Integrity Constraints

- Functional Dependencies *(E. F. Codd 1972)*

FD: country -> capital

country	capital	population	language
China	Beijing	10,000,000,000	Mandarin
China	Shanghai	11,000,000,000	Mandarin
Netherlands	Amsterdam	168,000,000	Dutch
Netherlands	Den Hagg	170,000,000	Dutch
Spain	Barcelona	472,700,000	Spanish

Integrity Constraints

- Functional Dependencies *(E. F. Codd 1972)*

FD: country -> capital

country	capital	population	language
China	Beijing	10,000,000,000	Mandarin
China	Shanghai	11,000,000,000	Mandarin
Netherlands	Amsterdam	168,000,000	Dutch
Netherlands	Den Hagg	170,000,000	Dutch
Spain	Barcelona	472,700,000	Spanish

Conditional Functional Dependencies

- Conditional Functional Dependencies *(Wenfei et al. ICDE 2007 best paper)*

CFD1: [country = China] -> [capital = Beijing]

country	capital	population	language
China	Beijing	10,000,000,000	Mandarin
China	Shanghai	11,000,000,000	Mandarin
Netherlands	Amsterdam	168,000,000	Dutch
Netherlands	Den Hagg	170,000,000	Dutch
Spain	Barcelona	472,700,000	Spanish

Conditional Functional Dependencies

- Conditional Functional Dependencies *(Wenfei et al. ICDE 2007 best paper)*

CFD1: [country = China] -> [capital = Beijing]

country	capital	population	language
China	Beijing	10,000,000,000	Mandarin
China	Shanghai	11,000,000,000	Mandarin
Netherlands	Amsterdam	168,000,000	Dutch
Netherlands	Den Hagg	170,000,000	Dutch
Spain	Barcelona	472,700,000	Spanish

CFD2: [country = China, language] -> [capital]

Other Extensions

- Metric Functional Dependencies *(Nick et al. ICDE 2009)*

X -> Y: for t, t', if t[X] = t'[X], then d(t[Y], t'[Y]) <= μ

Other Extensions

- Metric Functional Dependencies *(Nick et al. ICDE 2009)*

X -> Y: for t, t', if t[X] = t'[X], then d(t[Y], t'[Y]) <= μ

- extended Conditional Functional Dependencies *(Bravo et al. VLDB 2008)*

Negation: if the country is not Netherlands, …
Disjunction: if the country is either China or Spain, …

Other Extensions

- Metric Functional Dependencies *(Nick et al. ICDE 2009)*

X -> Y: for t, t', if t[X] = t'[X], then d(t[Y], t'[Y]) <= μ

- extended Conditional Functional Dependencies *(Bravo et al. VLDB 2008)*

Negation: if the country is not Netherlands, …
Disjunction: if the country is either China or Spain, …

- Differential Dependencies *(Shaoxu et al. TODS 2011)*

$\Omega_L[X] \rightarrow \Omega_R[Y]$

Other Extensions

- Metric Functional Dependencies *(Nick et al. ICDE 2009)*

X -> Y: for t, t', if t[X] = t'[X], then d(t[Y], t'[Y]) <= μ

- extended Conditional Functional Dependencies *(Bravo et al. VLDB 2008)*

Negation: if the country is not Netherlands, …
Disjunction: if the country is either China or Spain, …

- Differential Dependencies *(Shaoxu et al. TODS 2011)*

$$\Omega_L[X] \rightarrow \Omega_R[Y]$$

DD1: [cardno(= 0) ∧ position(≥ 60)] → [transtime(≥ 20)]
DD2: [date(≤ 7)] → [price(≤ 100)]
DD3: [date(> 7, ≤ 30)] → [price(≥ 100, ≤ 900)]

Denial Constraints

- *Foundations of databases (Serge et al. 1994)*

not (P1 and P2 and .. . and Pm)

Denial Constraints

- *Foundations of databases (Serge et al. 1994)*

not (P1 and P2 and .. . and Pm)

for any t1, t2 not (t1[salary] > t2[salary] and t1[tax] < t2[tax])

name	salary	tax
Divy	160	60
Nan	40	8
Paolo	42	7
Sanjay	80	20

Inclusion Dependency

- As known as referential integrity
  - *Foundations of databases (Serge et al. 1994)*

$$R[A1, \ldots, An] \subseteq S[B1, \ldots, Bn]$$

Inclusion Dependency

- As known as referential integrity
  - *Foundations of databases (Serge et al. 1994)*

$$R[A1, \ldots, An] \subseteq S[B1, \ldots, Bn]$$

- Conditional Inclusion Dependencies
  - *Extending dependencies with conditions (Bravo et al. VLDB 2007)*

$$R[A1, \ldots, An] \subseteq S[B1, \ldots, Bn], \text{ if conditions}$$

Matching Dependencies

- Reasoning about record matching rules (*Wenfei et al. VLDB 2009*)

$$R1[X] \approx R2[X] \rightarrow R1[Y] \approx R2[Y]$$

Matching Dependencies

- Reasoning about record matching rules (*Wenfei et al. VLDB 2009*)

R1[X] ≈ R2[X] -> R1[Y] ≈ R2[Y]

R1[name] ≈ R2[name] -> R1[salary] ≈ R2[salary]

name	payroll
Divyakant Agraval	170
Nan Tang	80
Paolo Papotti	42
Sanjay Chawla	80

R1

name	salary	tax
Divy Agraval	160	60
Nan Tang	40	8
Paolo Papotti	42	7
Sanjay Chawla	80	20

R2

Partial Order based Constraints

- Currency constraints [*Wenfei et al. PODS 2012*]
  - For the same entity with multiple instances
  - Partial order on some attribute *A*
    - *assistant professor* $\prec$ *associate professor* $\prec$ *professor*
  - Currency constraints: $\forall t1,t2(\omega \rightarrow t1 \prec_A t2)$
    - $\forall t1,t2(t1[kids] < t2[kids] \rightarrow t1 \prec_{kids} t2)$
- Accuracy constraints
  - Determining the relative accuracy of attributes [*Cao et al. I SIGMOD 2013*]

Much more …

- Check constraints
- Aggregate constraints
- Neighborhood constraints (*Shaoxu et al. VLDB 2014*)
- … … …

Fundamental Problems

- Consistency (*a.k.a.* satisfiability)
- Implication
- Inference

- Please find more theoretical results from *“Foundations of Data Quality Management”* by Wenfei Fan and Floris Geerts

Fundamental Problems

- Consistency (*a.k.a.* satisfiability)
- Implication
- Inference

- Please find more theoretical results from *"Foundations of Data Quality Management"* by Wenfei Fan and Floris Geerts

Most of data cleaning problems are with high complexity, ranging from PTIME to NP-complete, coNP-complete, and beyond

Statistical Methods

- Anomaly detection

Open Problem

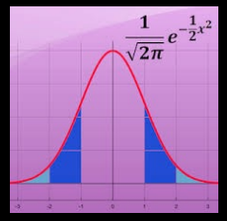
Data

Open Problem

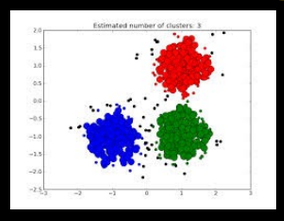
Error Detection

Frequency

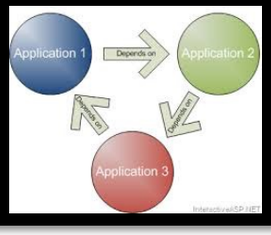
Distribution



Clustering



Dependency



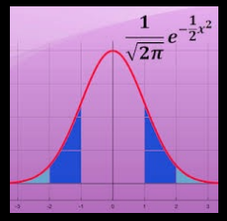
Data

Open Problem

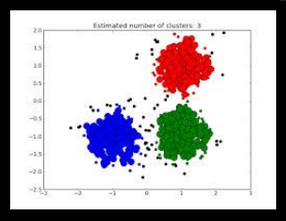
Error Detection

Frequency

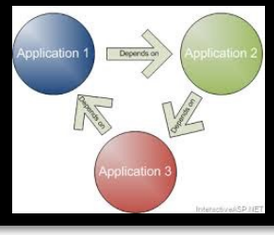
Distribution



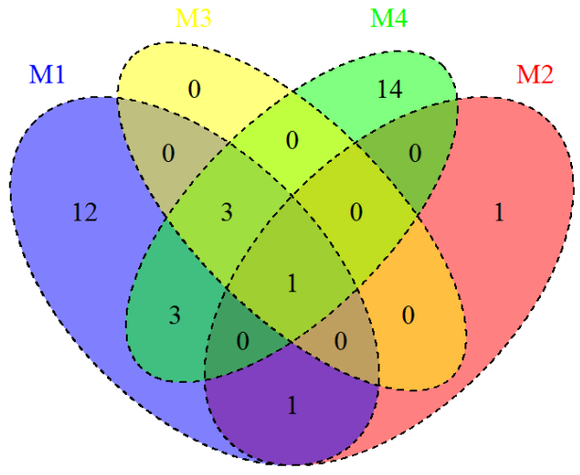
Clustering



Dependency



Data

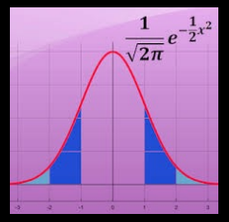


Open Problem

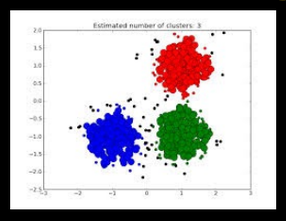
Error Detection

Frequency

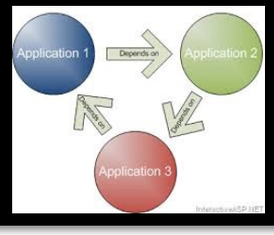
Distribution

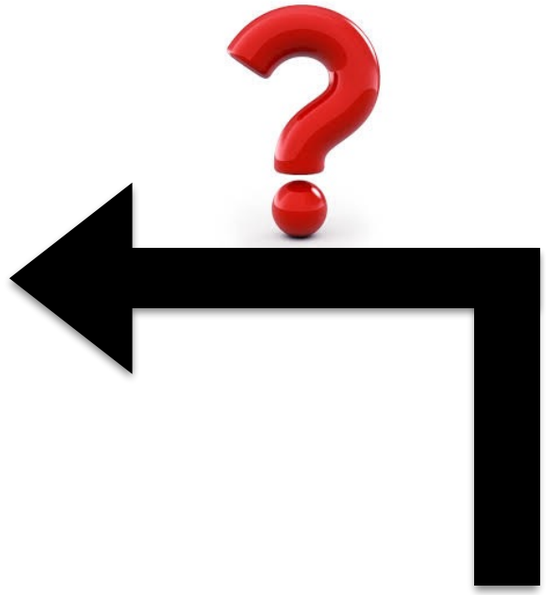


Clustering

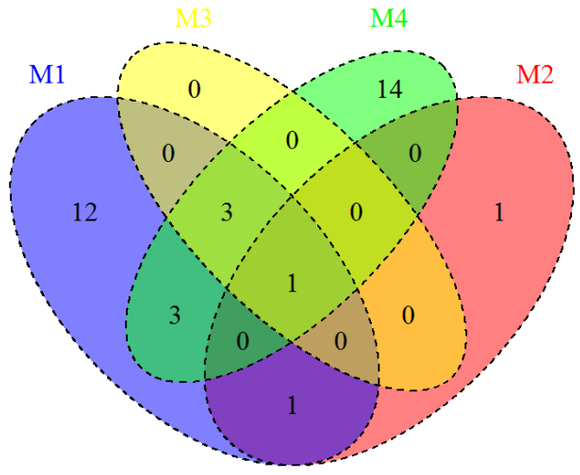


Dependency

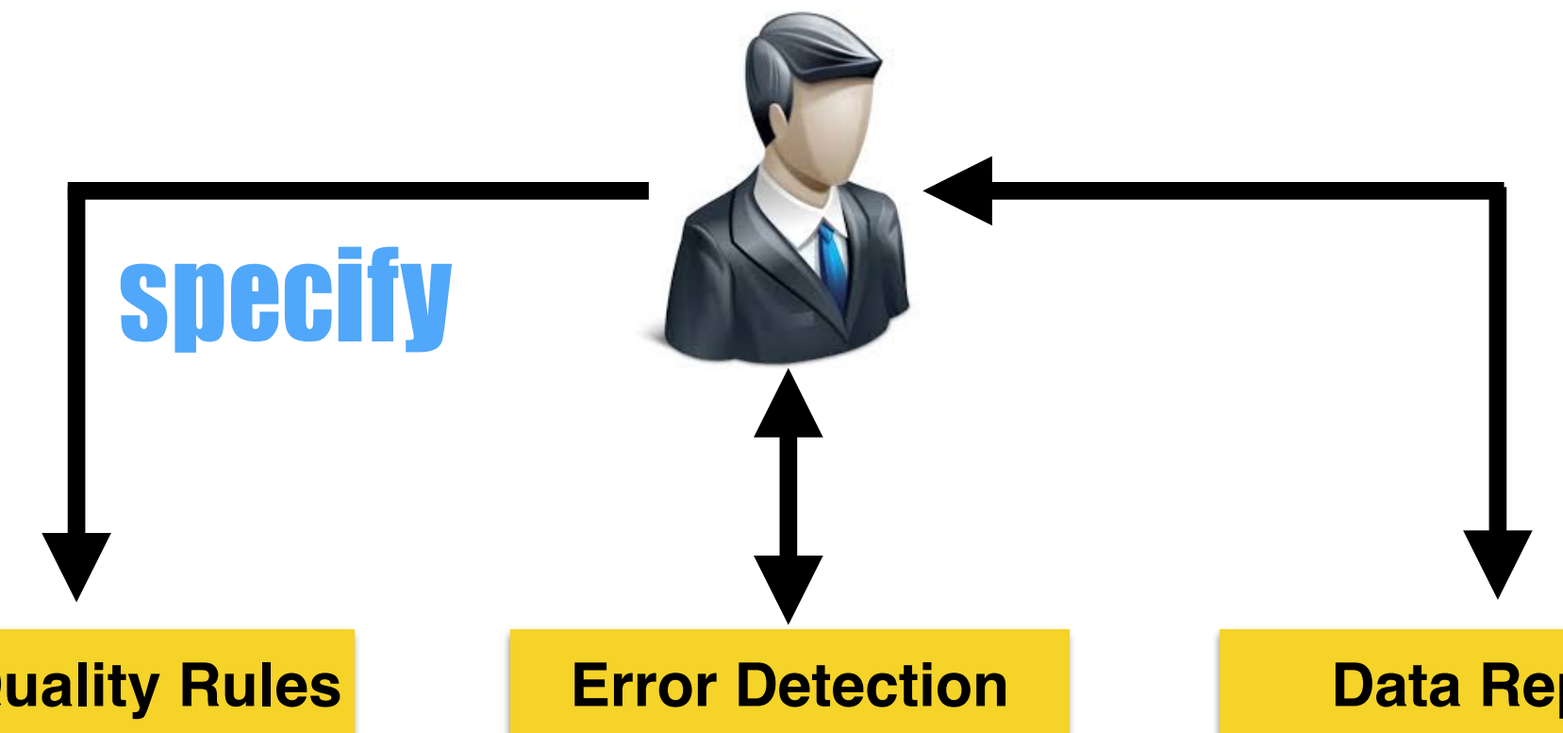




Data



Data Cleaning



Data

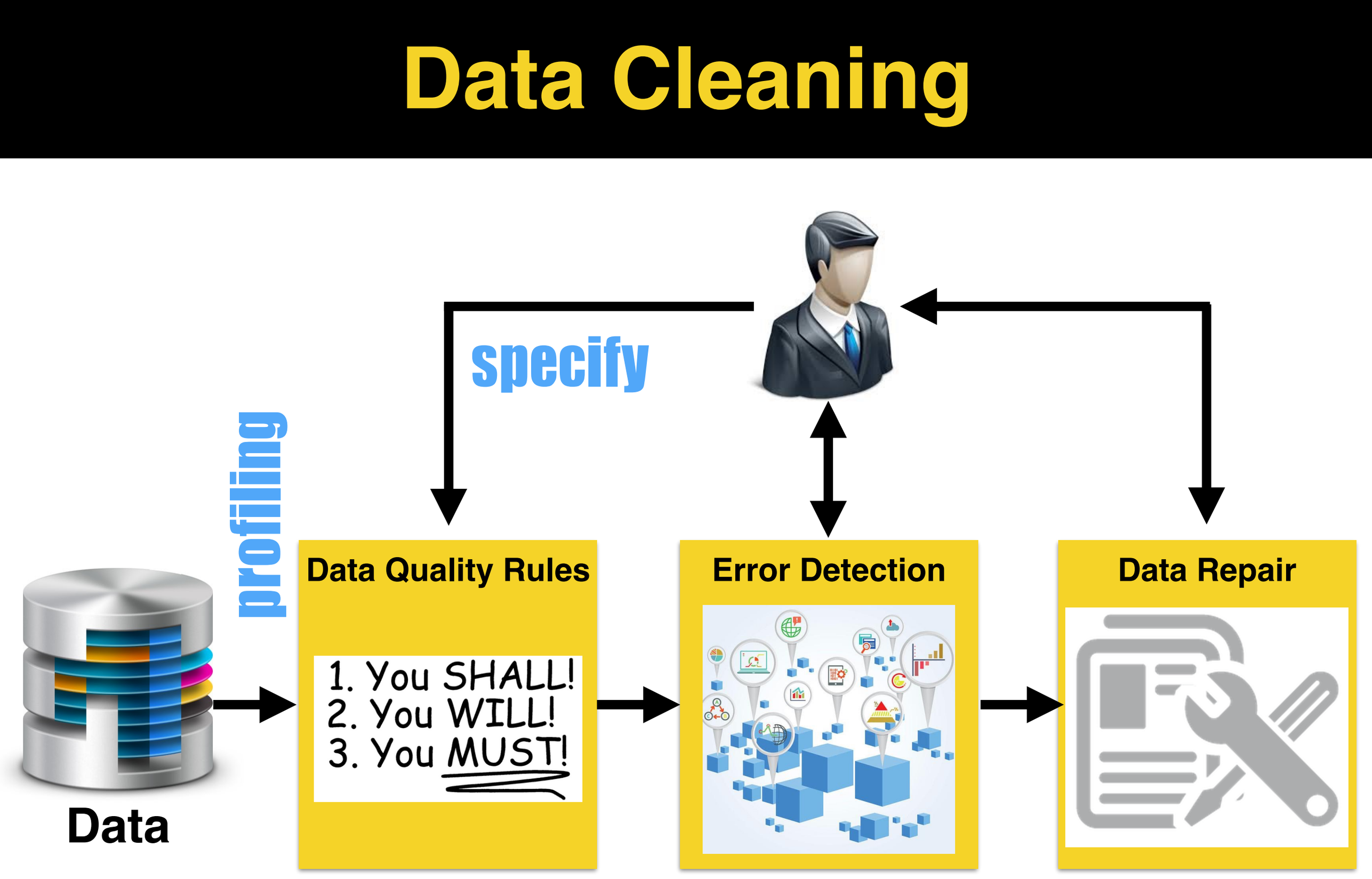
Data Quality Rules

1. You SHALL!
2. You WILL!
3. You MUST!

Error Detection

Data Repair

Data Cleaning



Profiling for Data Quality Rules

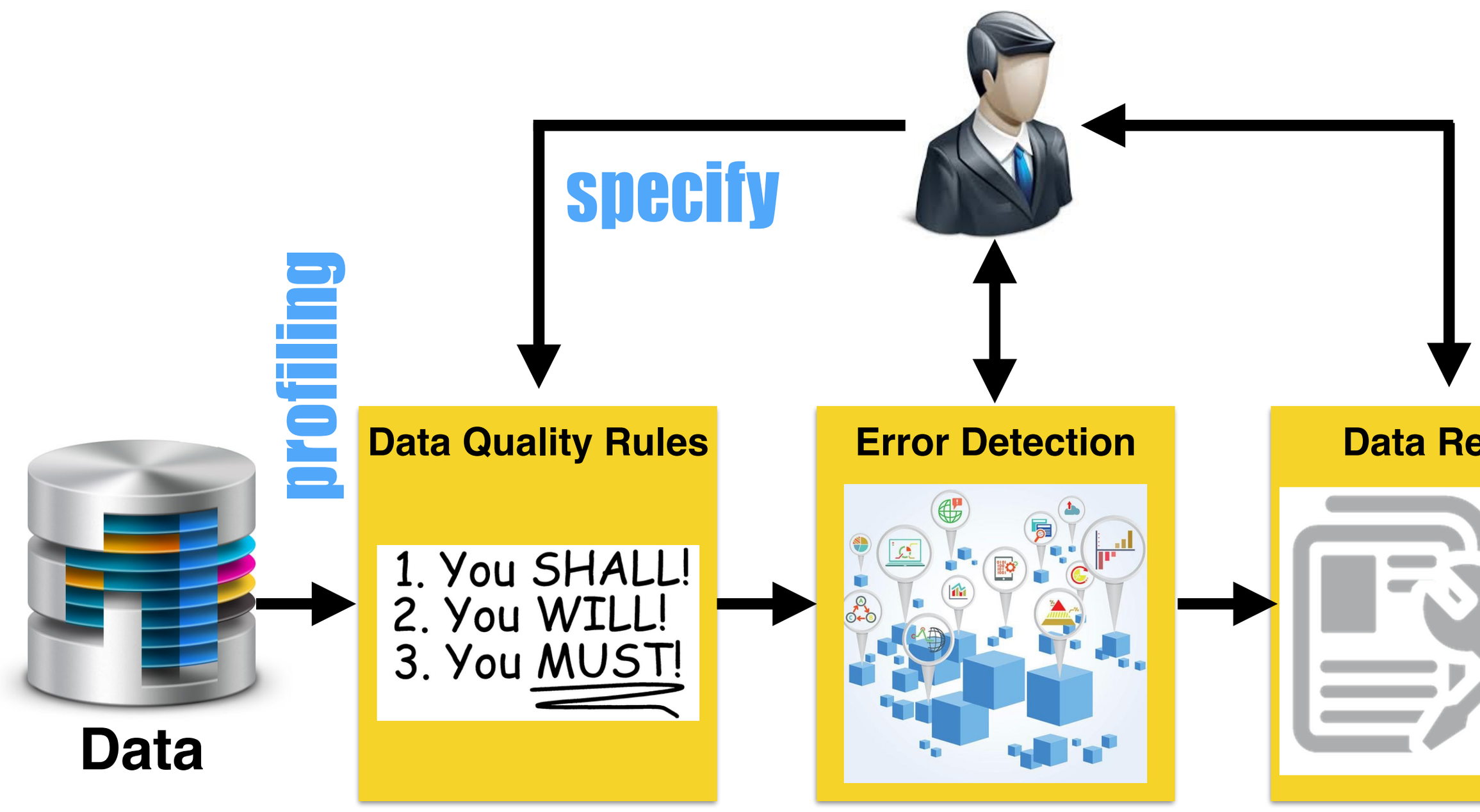
- Functional dependencies, Tane (*Huhtala et al. 1999*)
- CFDs (*Chiang et al. 2008, Wenfei et al. 2011*)
- Correlations and soft FDs, Cords (*Ihab et al. 2004*)
- Unique column combinations (*Arvid et al. 2013, Ziawasch et al. 2014*)
- Inclusion dependencies (*Papenbrock et al. 2015*)
- Matching dependencies (*Shaoxu et al. 2009*)
- Denial constraints (*Chu et al. VLDB 2014*)

Profiling for Data Quality Rules

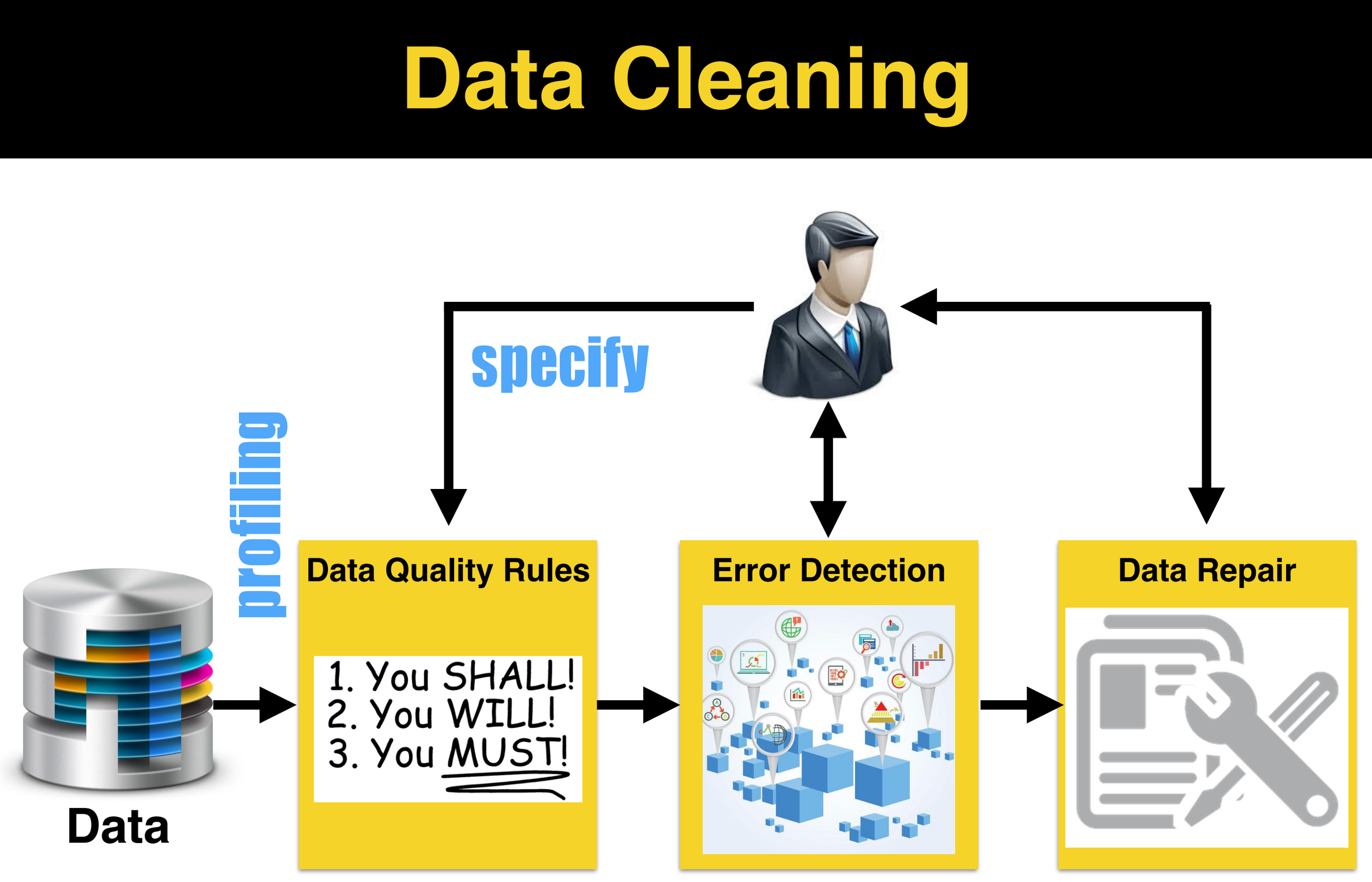
- Functional dependencies, Tane (*Huhtala et al. 1999*)
- CFDs (*Chiang et al. 2008, Wenfei et al. 2011*)
- Correlations and soft FDs, Cords (*Ihab et al. 2004*)
- Unique column combinations (*Arvid et al. 2013, Ziawasch et al. 2014*)
- Inclusion dependencies (*Papenbrock et al. 2015*)
- Matching dependencies (*Shaoxu et al. 2009*)
- Denial constraints (*Chu et al. VLDB 2014*)

1. Most of these approaches share the idea of (fault-tolerant) frequent pattern mining
2. The quality of profiling results heavily depends on the quality of data

Data Cleaning



Data Cleaning



Efficient Error Detection

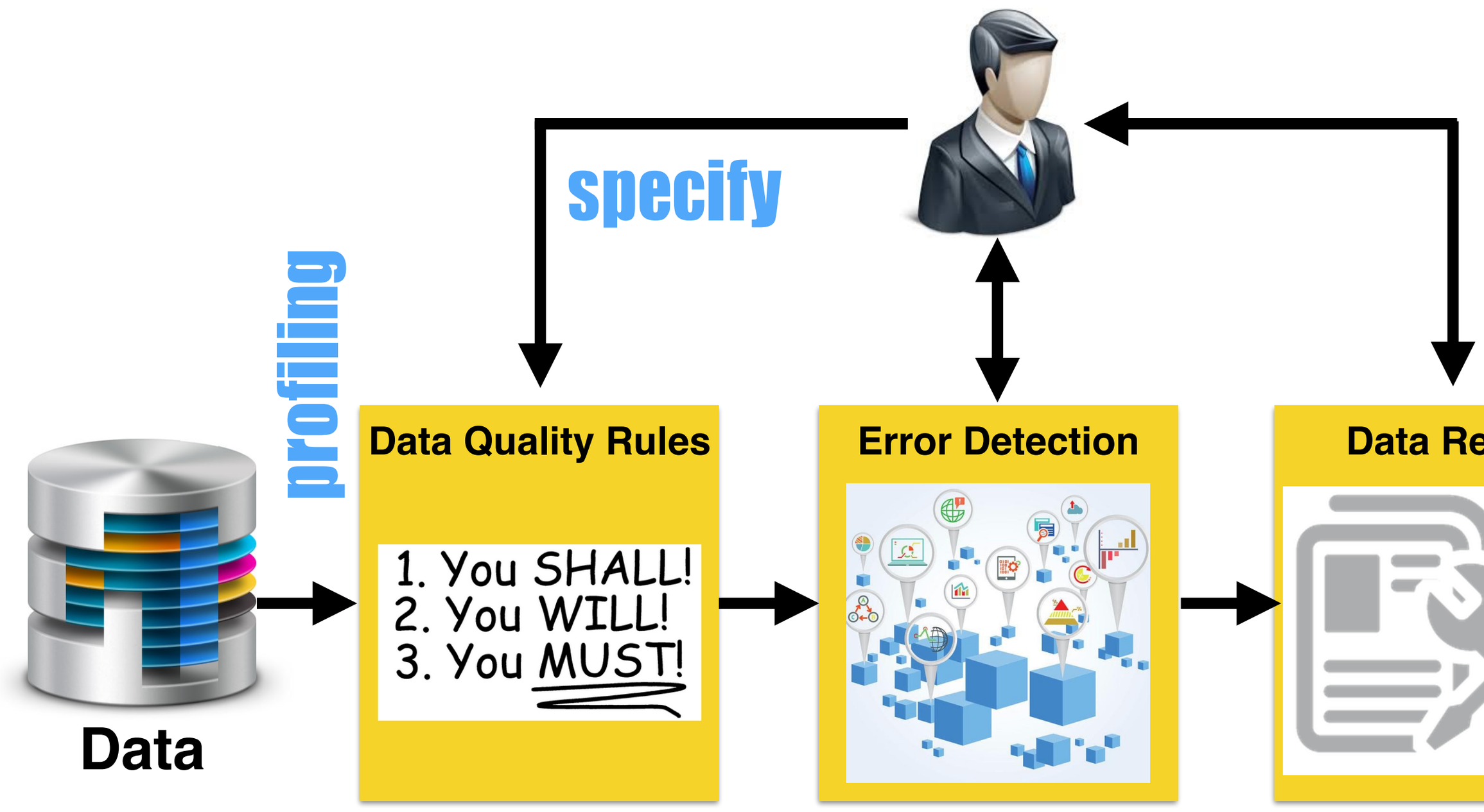
- CFDs in distributed environment
  - *Incremental detection of inconsistencies in distributes environment (Wenfei et al. ICDE 2012)*
- Generic error detection on Spark
  - *BigDansing: A system for big data cleansing (Zuhair et al. SIGMOD 2015)*

Efficient Error Detection

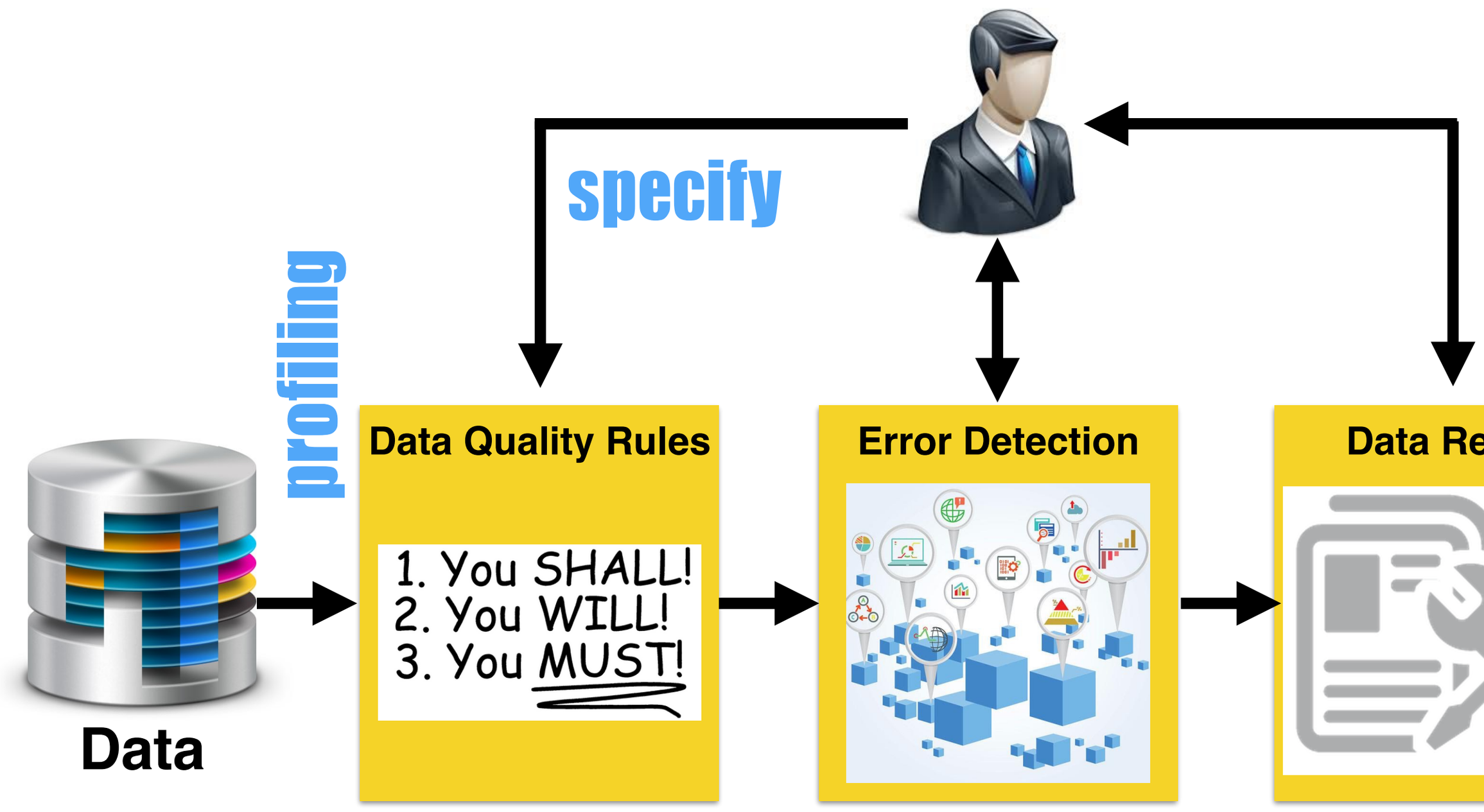
- CFDs in distributed environment
  - *Incremental detection of inconsistencies in distributes environment (Wenfei et al. ICDE 2012)*
- Generic error detection on Spark
  - *BigDansing: A system for big data cleansing (Zuhair et al. SIGMOD 2015)*

Most error detection methods are deterministic, hence the research challenges are mainly in the context of efficiency

Data Cleaning



Data Cleaning



Automatic Repair

Consistent Query Answering

- Consistent query answers in inconsistent databases (*Arenas et al. PODS 1999*)
  - Repair
    - a database that satisfies the integrity constraints
    - difference from the given database is minimal
  - A tuple (a1, . . ., an) is a consistent query answer to a query Q (x1, . . ., xn) in a database *R* if it is an element of the result of Q in every repair of r.

Consistent Query Answering

- Consistent query answers in inconsistent databases (*Arenas et al. PODS 1999*)
  - Repair
    - a database that satisfies the integrity constraints
    - difference from the given database is minimal
  - A tuple (a1, . . ., an) is a consistent query answer to a query Q (x1, . . ., xn) in a database *R* if it is an element of the result of Q in every repair of r.

Queries	Functional dependencies		Denial constraints
	$\|F\| = 1$	$\|F\| \geq 2$	
$\sigma, \times, -, \cup$	PTIME	PTIME	PTIME
$\pi, \sigma, \times$ (no join)	PTIME	co-NP-complete	co-NP-complete
$\pi, \sigma, \times$ (join)	co-NP-complete	co-NP-complete	co-NP-complete

A Consistent Minimum Repair

- Find a repair with the minimum cost (*Bohannon et al. SIGMOD 2005*)
- NP-complete problem: find a repair with cost at most *m*

A Consistent Minimum Repair

- Find a repair with the minimum cost (*Bohannon et al. SIGMOD 2005*)
- NP-complete problem: find a repair with cost at most *m*

FD1: A -> B; FD2: B -> C

	A	B	C
t1	a1	b1	c1
t2	a1	b2	c2
t3	a1	b1	c1
t4	a2	b2	c3

A Consistent Minimum Repair

- Find a repair with the minimum cost (*Bohannon et al. SIGMOD 2005*)
- NP-complete problem: find a repair with cost at most *m*

FD1: A -> B; FD2: B -> C

	A	B	C
t1	a1	b1	c1
t2	a1	b2	c2
t3	a1	b1	c1
t4	a2	b2	c3

A Consistent Minimum Repair

- Find a repair with the minimum cost (*Bohannon et al. SIGMOD 2005*)
- NP-complete problem: find a repair with cost at most *m*

FD1: A -> B; FD2: B -> C

	A	B	C
t1	a1	b1	c1
t2	a1	b2	c2
t3	a1	b1	c1
t4	a2	b2	c3

t1[B] = t2[B]

t2[B] = t3[B]

t2[C] = t4[C]

Equivalence Class

- (*Bohannon et al. SIGMOD 2005*)

FD1: A -> B; FD2: B -> C

	A	B	C
t1	a1	b1	c1
t2	a1	b2	c2
t3	a1	b1	c1
t4	a2	b2	c3

t1[B] = t2[B]

t2[B] = t3[B]

t2[C] = t4[C]

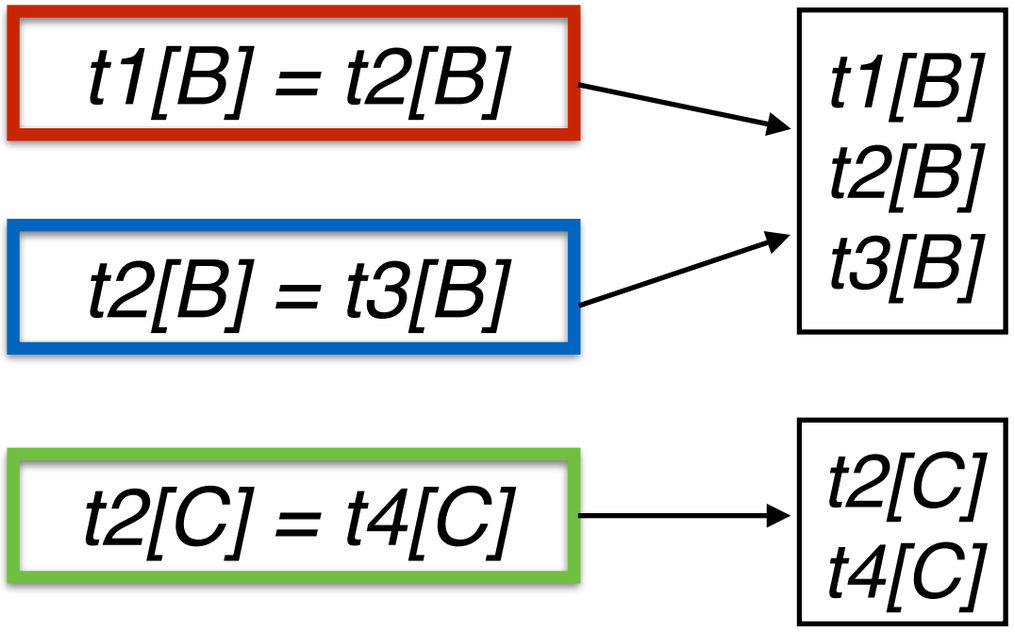
Equivalence Class

- (*Bohannon et al. SIGMOD 2005*)

FD1: A -> B; FD2: B -> C

	A	B	C
t1	a1	b1	c1
t2	a1	b2	c2
t3	a1	b1	c1
t4	a2	b2	c3

Equivalence classes

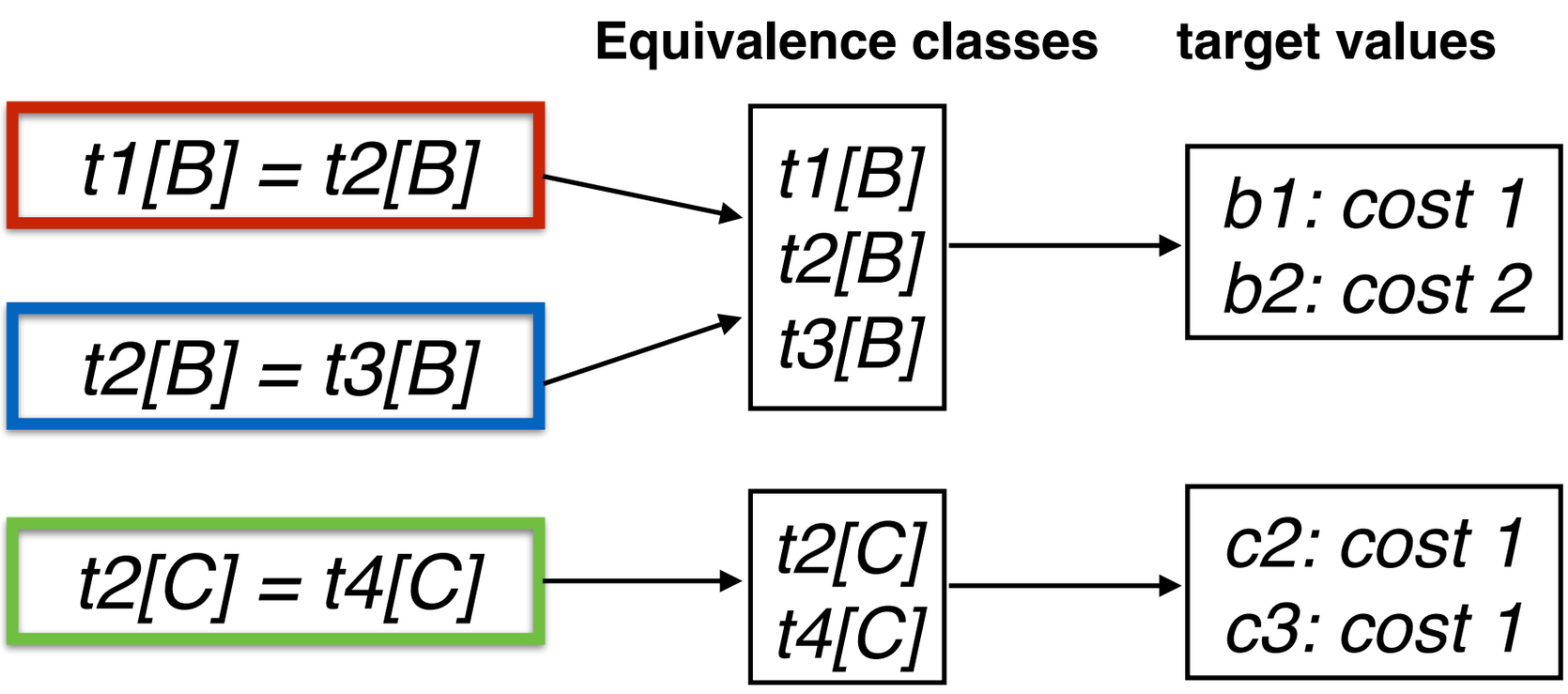


Equivalence Class

- (*Bohannon et al. SIGMOD 2005*)

FD1: A -> B; FD2: B -> C

	A	B	C
t1	a1	b1	c1
t2	a1	b2	c2
t3	a1	b1	c1
t4	a2	b2	c3

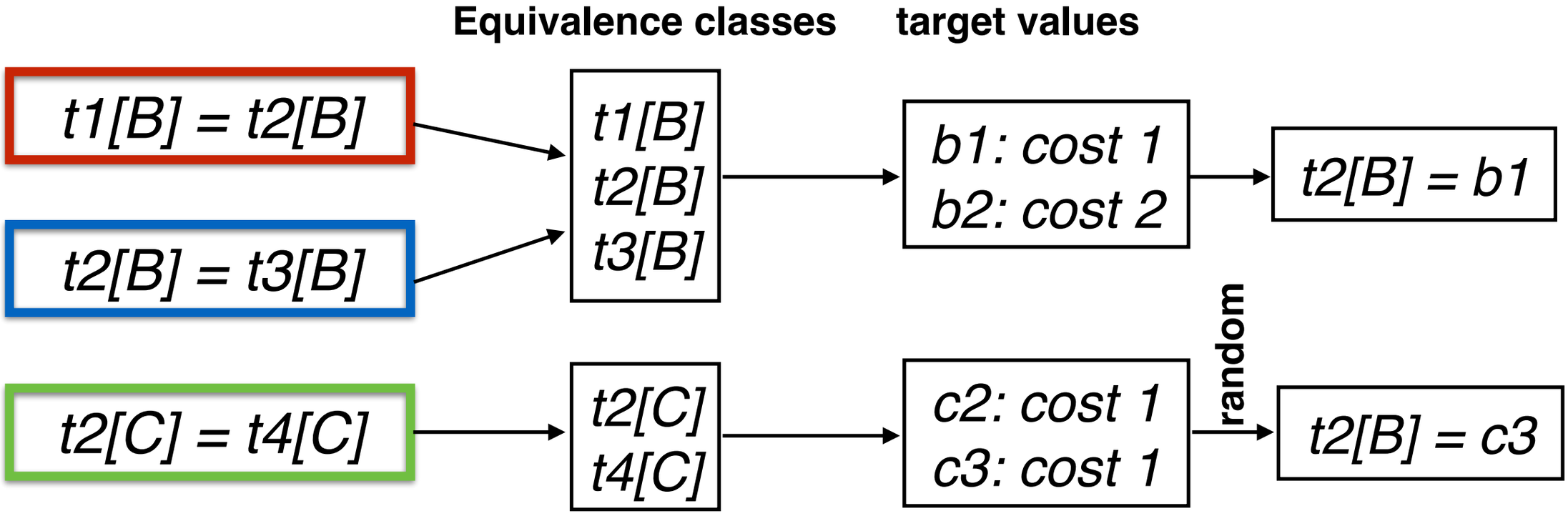


Equivalence Class

- (*Bohannon et al. SIGMOD 2005*)

FD1: A -> B; FD2: B -> C

	A	B	C
t1	a1	b1	c1
t2	a1	b2	c2
t3	a1	b1	c1
t4	a2	b2	c3

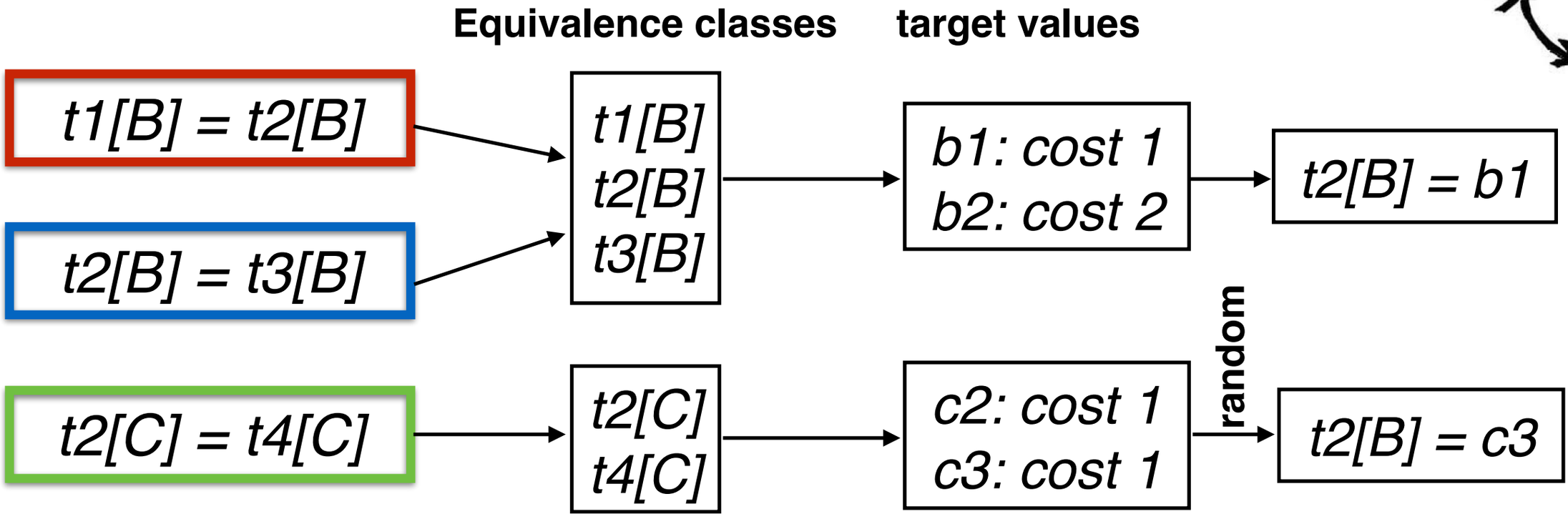


Equivalence Class

- (*Bohannon et al. SIGMOD 2005*)

FD1: A -> B; FD2: B -> C

	A	B	C
t1	a1	b1	c1
t2	a1	b2	c2
t3	a1	b1	c1
t4	a2	b2	c3



Holistic Data Cleaning

FD1: A -> B; FD2: B -> C

	A	B	C
t1	a1	b1	c1
t2	a1	b2	c2
t3	a1	b1	c1
t4	a2	b2	c3

- Holistic cleaning (minimum set cover)
  - Holistic data cleaning: putting violations into contexts (*Chu et al. ICDE 2013*)
- SAT-solver
  - NADEEF: a commodity data cleaning system (*Michele et al. SIGMOD 2013*)

Possible Repairs

- Modeling and querying possible repairs in duplicate detection (George et al. VLDB 2009)

- Uncertain models

- Multiple possible repairs with probabilities

Machine Learning based Approached

- Don’t be scared: use scalable automatic repairing with maximum likelihood and bounded changes (*Mohamed et al. SIGMOD 2013*)

Integrity Constraints Can Also Be Wrong

- Unified repair for data and constraints (*Chiang et al. ICDE 2011*)

- Relative trust of problematic data and constraints (*George et al. VLDB 2013*)

- Continuous data cleaning (*Christina et al. ICDE 2014*)

Concerns of Automatic Repair

- Minimality is nothing about finding the ground truth
- All methods work based on the redundancy of data
- Changing *Beijing* to *Shanghai* has the lowest cost

FD: country -> capital

country	capital	population	language
China	Beijing	10,000,000,000	Mandarin
China	Shanghai	11,000,000,000	Mandarin
China	Shanghai	12,000,000,000	Mandarin

Reliable External Sources

Confidence Values

- Some data may come from reliable data sources
- Users can place confidence values on the data

FD: country -> capital

country	capital	population	language
China (1.0)	Beijing (0.9)	10,000,000,000	Mandarin
China (1.0)	Shanghai (0.4)	11,000,000,000	Mandarin
China (1.0)	Shanghai (0.5)	12,000,000,000	Mandarin

Confidence Values

- Some data may come from reliable data sources
- Users can place confidence values on the data

FD: country -> capital

country	capital	population	language
China (1.0)	Beijing (0.9)	10,000,000,000	Mandarin
China (1.0)	Shanghai (0.4)	11,000,000,000	Mandarin
China (1.0)	Shanghai (0.5)	12,000,000,000	Mandarin

- Improving data quality: consistency and accuracy (*Cong et al. VLDB 2008*)
- Interaction between record matching and data repairing (*Wenfei et al. SIGMOD 2011*)

Master Data and Users

	name	country	capital	city	conf
r1	George	China	Beijing	Beijing	SIGMOD
r2	Ian	China	Shanghai	Hongkong	ICDE
r3	Peter	China	Tokyo	Tokyo	ICDE
r4	Mike	Canada	Toronto	Toronto	VLDB

country	capital
China	Beijing
Canada	Ottawa
Japan	Tokyo

- *Towards certain fixes with editing rules and master data (Wenfei et al. VLDB 2010 best paper)*

Master Data and Users

editing rule: ((country, country) -> (capital, capital))

	name	country	capital	city	conf
r1	George	China	Beijing	Beijing	SIGMOD
r2	Ian	China	Shanghai	Hongkong	ICDE
r3	Peter	China	Tokyo	Tokyo	ICDE
r4	Mike	Canada	Toronto	Toronto	VLDB

country	capital
China	Beijing
Canada	Ottawa
Japan	Tokyo

- *Towards certain fixes with editing rules and master data (Wenfei et al. VLDB 2010 best paper)*

Master Data and Users

editing rule: ((country, country) -> (capital, capital))

	name	country	capital	city	conf
r1	George	China	Beijing	Beijing	SIGMOD
r2	Ian	China	Shanghai	Hongkong	ICDE
r3	Peter	China	Tokyo	Tokyo	ICDE
r4	Mike	Canada	Toronto	Toronto	VLDB

country	capital
China	Beijing
Canada	Ottawa
Japan	Tokyo

- *Towards certain fixes with editing rules and master data (Wenfei et al. VLDB 2010 best paper)*

Master Data and Users

editing rule: ((country, country) -> (capital, capital))

	name	country	capital	city	conf
r1	George	China	Beijing	Beijing	SIGMOD
r2	Ian	China	Shanghai	Hongkong	ICDE
r3	Peter	China	Tokyo	Tokyo	ICDE
r4	Mike	Canada	Toronto	Toronto	VLDB

country	capital
China	Beijing
Canada	Ottawa
Japan	Tokyo

Is r2[country] China?
YES.

- *Towards certain fixes with editing rules and master data (Wenfei et al. VLDB 2010 best paper)*

Master Data and Users

editing rule: ((country, country) -> (capital, capital))

	name	country	capital	city	conf
r1	George	China	Beijing	Beijing	SIGMOD
r2	Ian	China	**Beijing**	Hongkong	ICDE
r3	Peter	China	Tokyo	Tokyo	ICDE
r4	Mike	Canada	Toronto	Toronto	VLDB

country	capital
China	Beijing
Canada	Ottawa
Japan	Tokyo

Is r2[country] China?
YES.

- *Towards certain fixes with editing rules and master data (Wenfei et al. VLDB 2010 best paper)*

Master Data and Users

editing rule: ((country, country) -> (capital, capital))

	name	country	capital	city	conf
r1	George	China	Beijing	Beijing	SIGMOD
r2	Ian	China	**Beijing**	Hongkong	ICDE
r3	Peter	China	Tokyo	Tokyo	ICDE
r4	Mike	Canada	Toronto	Toronto	VLDB

country	capital
China	Beijing
Canada	Ottawa
Japan	Tokyo

Is r2[country] China?
YES.

Is r1[country] China?

Is r3[country] China?

Is r4[country] Canada?

… … … …

- *Towards certain fixes with editing rules and master data (Wenfei et al. VLDB 2010 best paper)*

User Guided

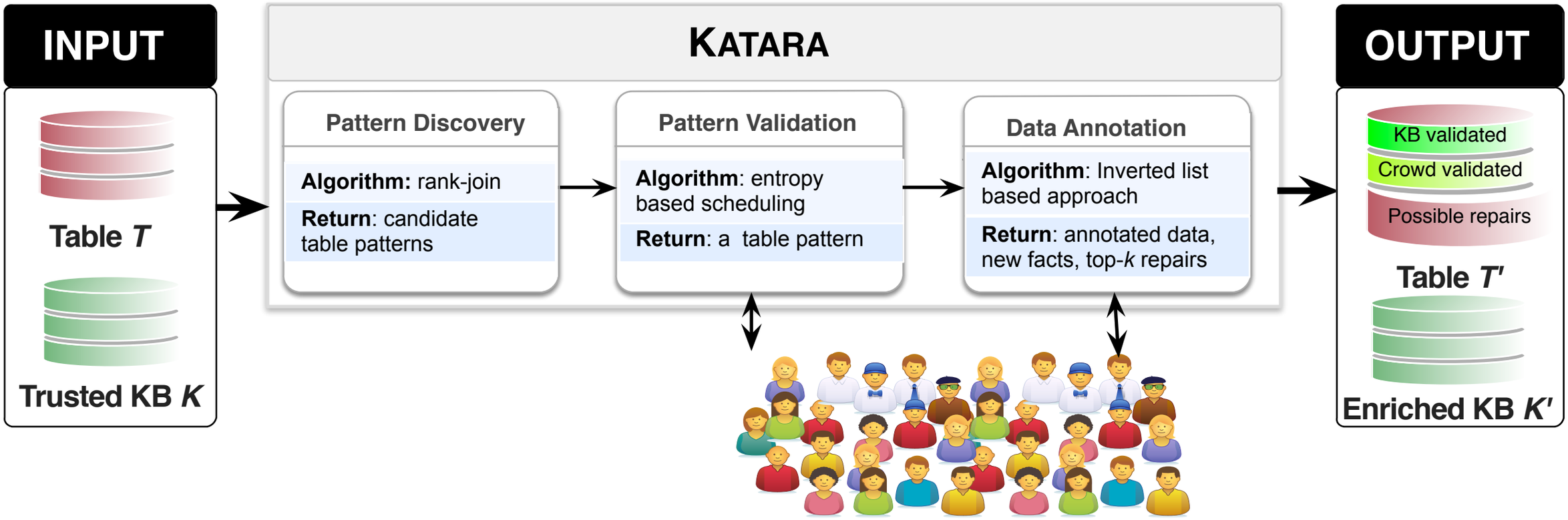
- *Towards certain fixes with editing rules and master data (Wenfei et al. VLDB 2010)*

- *Guided data repair (Mohamed et al. VLDB 2011)*

- *Continuous data cleaning (Christina et al. ICDE 2014)*

- *Conflict resolution with data currency and consistency (Wenfei et al. JDIQ 2014)*

Knowledge Bases and Crowdsourcing



- *KATARA: a data cleaning system powered by knowledge bases and crowdsourcing (Xu et al. SIGMOD 2015)*

precision: +
recall: ++

Heuristic
(Automated)

precision: ++
recall: ++

Certain
(User guided)

precision: +
recall: ++

Heuristic (Automated)

precision: ++
recall: +

Fixing Rules (Automated)

precision: ++
recall: ++

Certain (User guided)

Fixing rules

- **Challenge**
  - Automated and dependable data repairing

- *Towards dependable data repairing with fixing rules (Jiannan et al. SIGMOD 2014)*

Fixing rules

- **Challenge**
  - Automated and dependable data repairing

- *Towards dependable data repairing with fixing rules (Jiannan et al. SIGMOD 2014)*

Fixing rules

- **Challenge**
  - Automated and dependable data repairing

- **Observation**
  - Certain ***data patterns*** of semantically related values can provide evidence to capture and rectify data errors

- *Towards dependable data repairing with fixing rules (Jiannan et al. SIGMOD 2014)*

Fixing rules

- **Challenge**
  - Automated and dependable data repairing

- **Observation**
  - Certain ***data patterns*** of semantically related values can provide evidence to capture and rectify data errors

(China, Shanghai)

- *Towards dependable data repairing with fixing rules (Jiannan et al. SIGMOD 2014)*

Fixing rules

- **Challenge**
  - Automated and dependable data repairing

- **Observation**
  - Certain ***data patterns*** of semantically related values can provide evidence to capture and rectify data errors

(China, Shanghai)

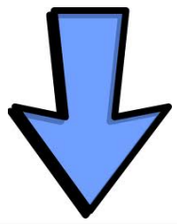
- *Towards dependable data repairing with fixing rules (Jiannan et al. SIGMOD 2014)*

Fixing rules

- **Challenge**
  - Automated and dependable data repairing

- **Observation**
  - Certain ***data patterns*** of semantically related values can provide evidence to capture and rectify data errors

(China, Shanghai)



(China, Beijing)

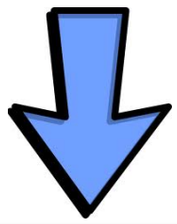
- *Towards dependable data repairing with fixing rules (Jiannan et al. SIGMOD 2014)*

Fixing rules

- **Challenge**
  - Automated and dependable data repairing

- **Observation**
  - Certain ***data patterns*** of semantically related values can provide evidence to capture and rectify data errors

(China, Shanghai)



(China, Beijing)

(China, Tokyo)

- *Towards dependable data repairing with fixing rules (Jiannan et al. SIGMOD 2014)*

Fixing Rules (cont’)

- **Syntax**

fR1: (([country], [China]), (capital, {Shanghai, Hongkong})) -> Beijing

Fixing Rules (cont')

- **Syntax**

fR1: (([country], [China]), (capital, {Shanghai, Hongkong})) -> Beijing

country	{capital$^-$}	capital$^+$
China	Shanghai	Beijing
	Hongkong	

Fixing Rules (cont')

- **Syntax**

fR1: (([country], [China]), (capital, {Shanghai, Hongkong})) -> Beijing

evidence **negative**

country	{capital⁻}	capital⁺
China	Shanghai Hongkong	Beijing

Fixing Rules (cont')

- **Syntax**

fR1: (([country], [China]), (capital, {Shanghai, Hongkong})) -> Beijing

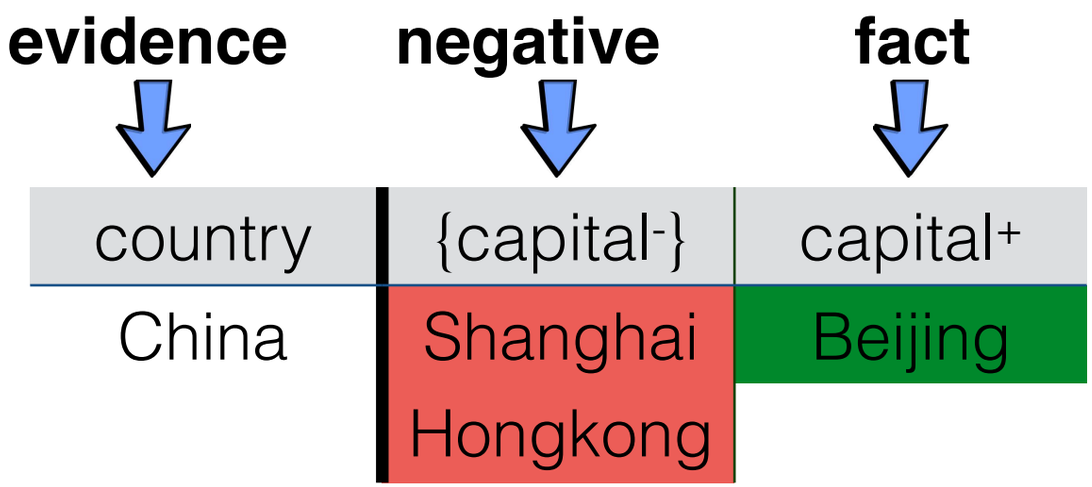
evidence **negative** **fact**

country	{capital⁻}	capital⁺
China	Shanghai Hongkong	Beijing

Fixing Rules (cont')

- **Syntax**

fR1: (([country], [China]), (capital, {Shanghai, Hongkong})) -> Beijing



	name	nationality	capital	bornAt
r1	Nan	China	Beijing	Shenyang
r2	Yan	China	Shanghai	Hangzhou
r3	Si	China	Beijing	Changsha
r4	Miura	China	Tokyo	Kyoto

Fixing Rules (cont')

- **Syntax**

fR1: (([country], [China]), (capital, {Shanghai, Hongkong})) -> Beijing

evidence **negative** **fact**

country	{capital⁻}	capital⁺
China	Shanghai Hongkong	Beijing

	name	nationality	capital	bornAt
r1	Nan	China	Beijing	Shenyang
r2	Yan	China	Shanghai	Hangzhou
r3	Si	China	Beijing	Changsha
r4	Miura	China	Tokyo	Kyoto

Fixing Rules (cont')

- **Syntax**

fR1: (([country], [China]), (capital, {Shanghai, Hongkong})) -> Beijing

evidence **negative** **fact**

country	{capital⁻}	capital⁺
China	Shanghai Hongkong	Beijing

	name	nationality	capital	bornAt
r1	Nan	China	Beijing	Shenyang
r2	Yan	China	Beijing	Hangzhou
r3	Si	China	Beijing	Changsha
r4	Miura	China	Tokyo	Kyoto

Fixing Rules (cont')

- **Syntax**

fR1: (([country], [China]), (capital, {Shanghai, Hongkong})) -> Beijing

evidence	negative	fact
country	{capital⁻}	capital⁺
China	Shanghai Hongkong	Beijing

	name	nationality	capital	bornAt
r1	Nan	China	Beijing	Shenyang
r2	Yan	China	Beijing	Hangzhou
r3	Si	China	Beijing	Changsha
r4	Miura	China	Tokyo	Kyoto

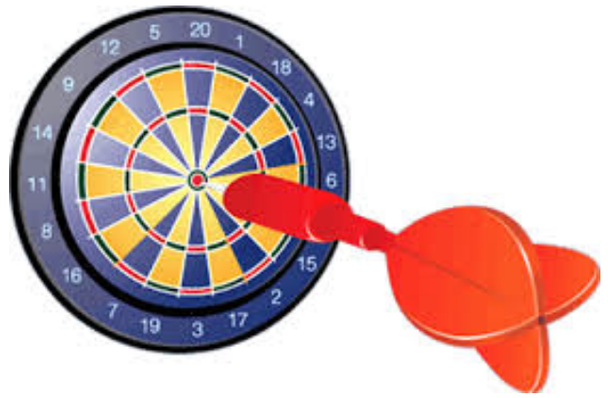
Fixing Rules (cont')

- **Syntax**

fR1: (([country], [China]), (capital, {Shanghai, Hongkong})) -> Beijing

evidence **negative** **fact**

country	{capital⁻}	capital⁺
China	Shanghai Hongkong	Beijing



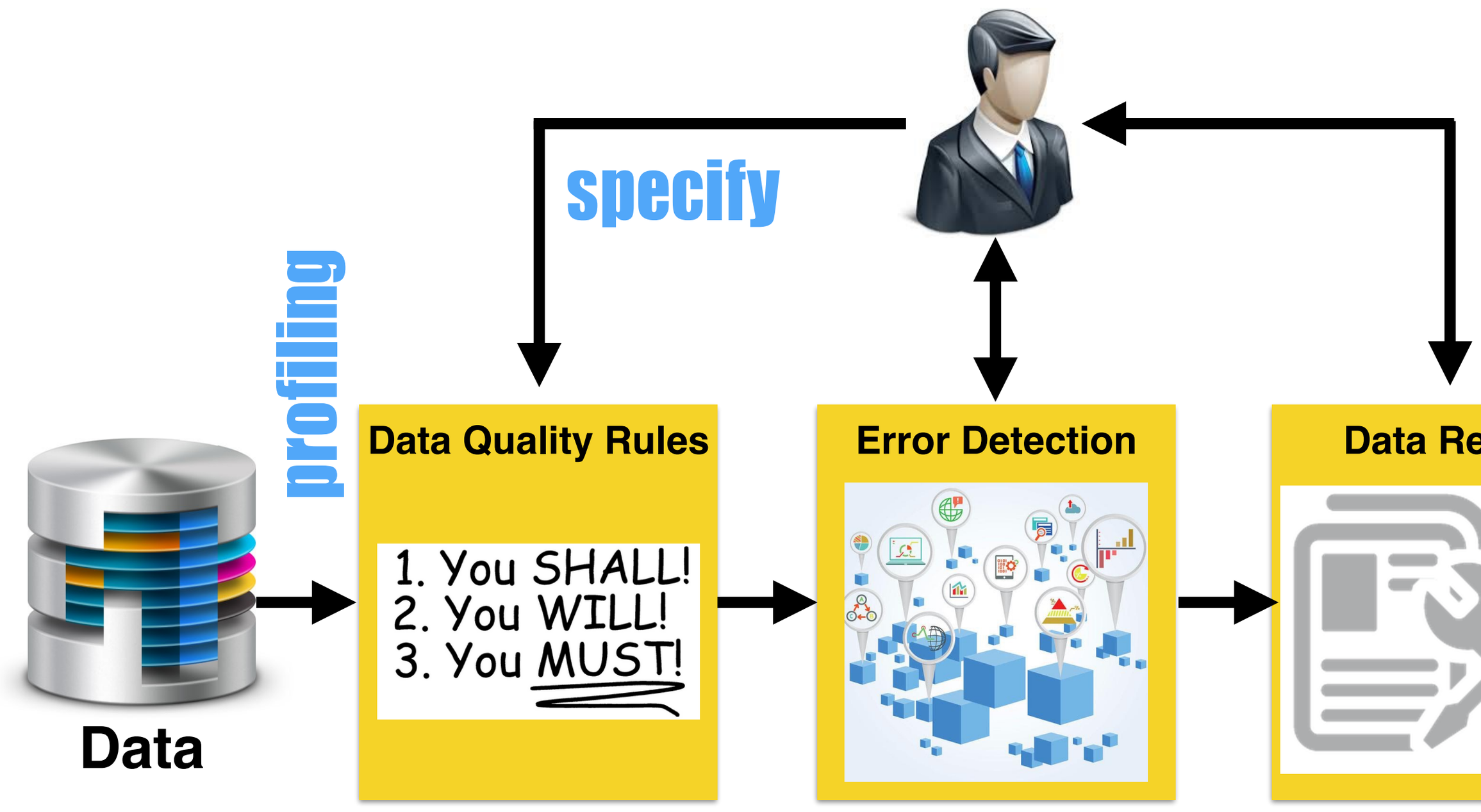
	name	nationality	capital	bornAt
r1	Nan	China	Beijing	Shenyang
r2	Yan	China	Beijing	Hangzhou
r3	Si	China	Beijing	Changsha
r4	Miura	China	Tokyo	Kyoto

Sherlock Rules: Proof Positive and Negative

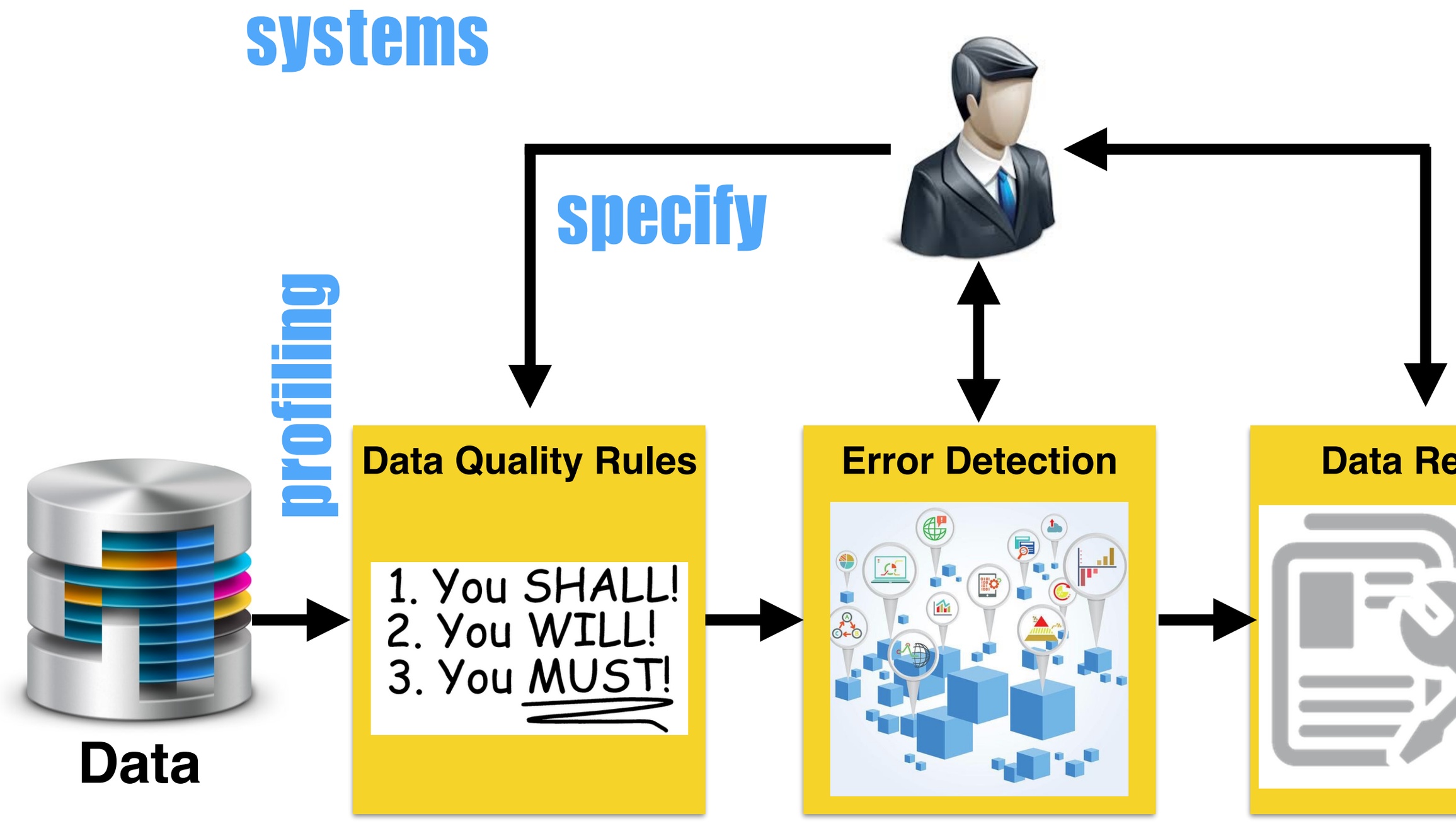
Fundamental Problems for Rule-based Repair

- Termination
- Consistency
- Implication
- Determinism

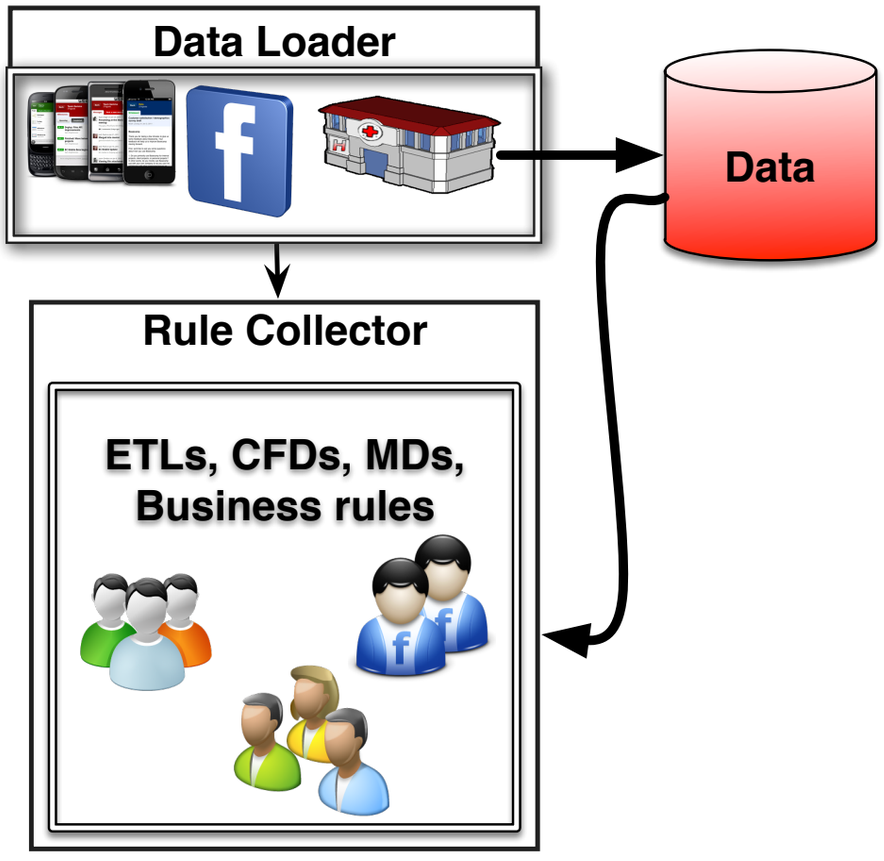
Data Cleaning



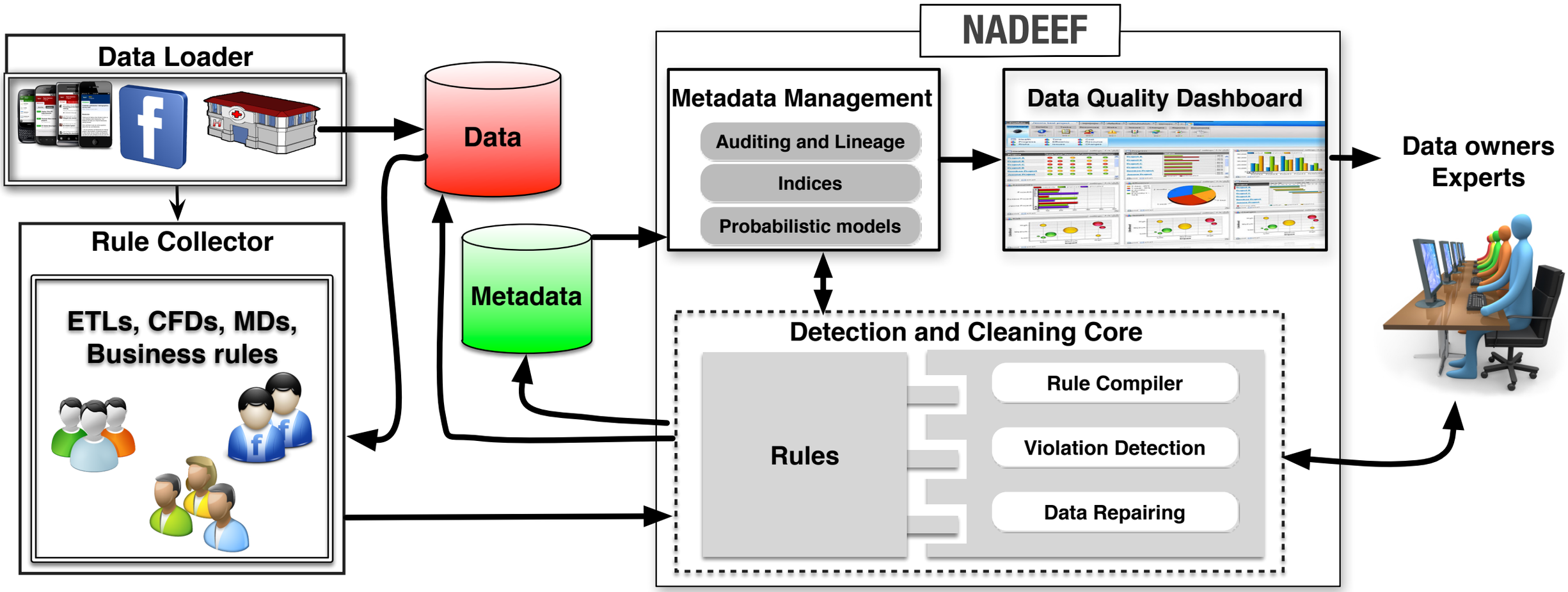
Data Cleaning



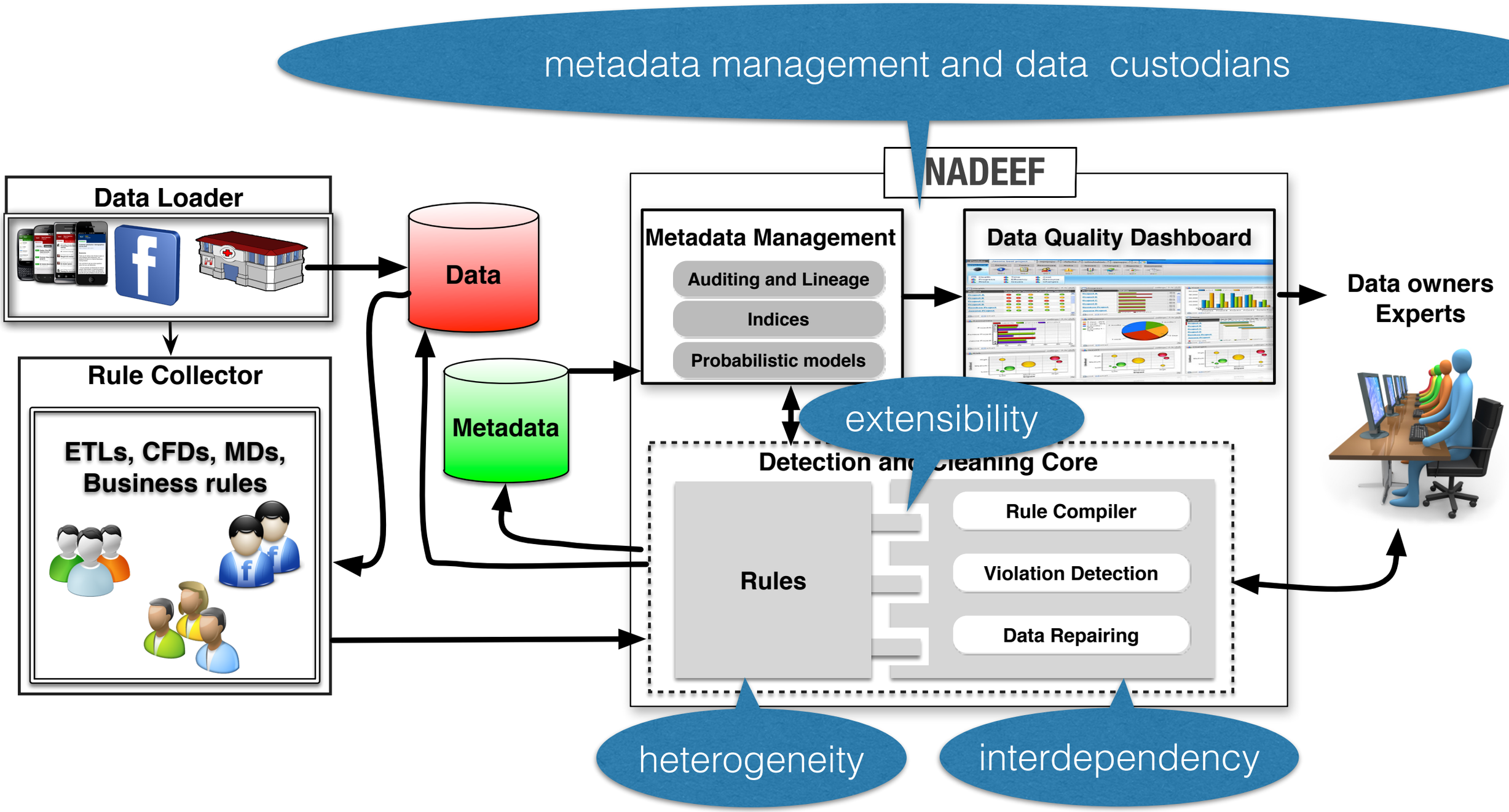
NADEE: A Commodity Data Cleaning System



NADEE: A Commodity Data Cleaning System



NADEE: A Commodity Data Cleaning System



NADEEF

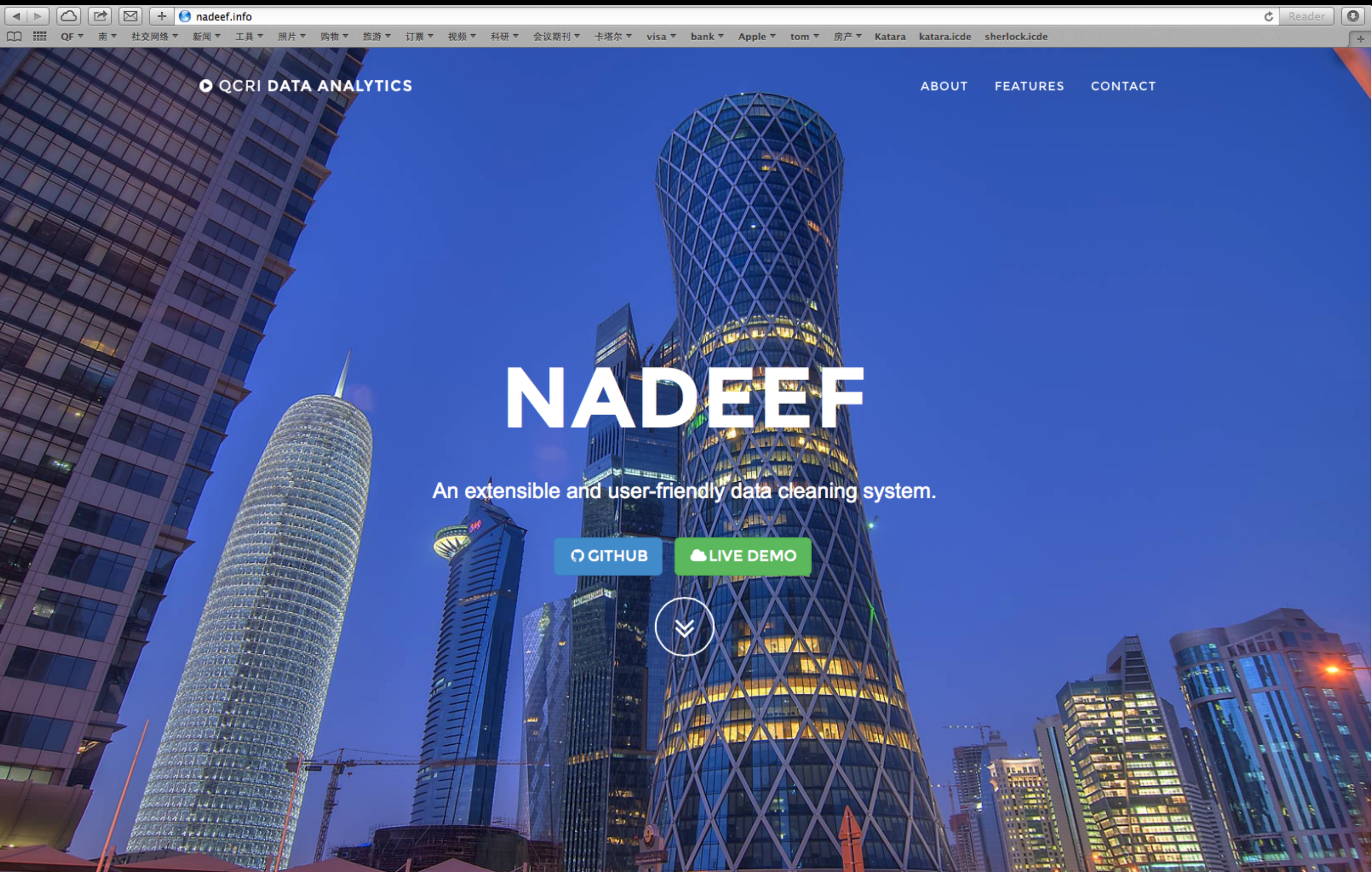
Rule Editor

Detect
Repair
Block
Iterator

```
 8      @Override
 9      public Collection<Violation> detect(TuplePair tuplePair) {
10          List<Violation> result = new ArrayList<>();
11          Tuple left = tuplePair.getLeft();
12          Tuple right = tuplePair.getRight();
13
14          if (
15              Metrics.getEqual(
16                  left.get("name"), right.get("name")) == 1.0 &&
17              Metrics.getLevenshtein(
18                  left.get("address"), right.get("address")) > 0.8 &&
19              Metrics.getEqual(
20                  left.get("gender"), right.get("gender")) == 1.0
21          ) {
22              Violation v = new Violation(getRuleName());
23              v.addTuple(left);
24              v.addTuple(right);
25              result.add(v);
26          }
27          return result;
28      }
29
30
```

Close | Save changes

NADEEF Code Release



Llunatic

Other Topics

Generating Explanations

- *Descriptive and prescriptive data repairing with fixing rules (Chalamalla et al. SIGMOD 2014)*