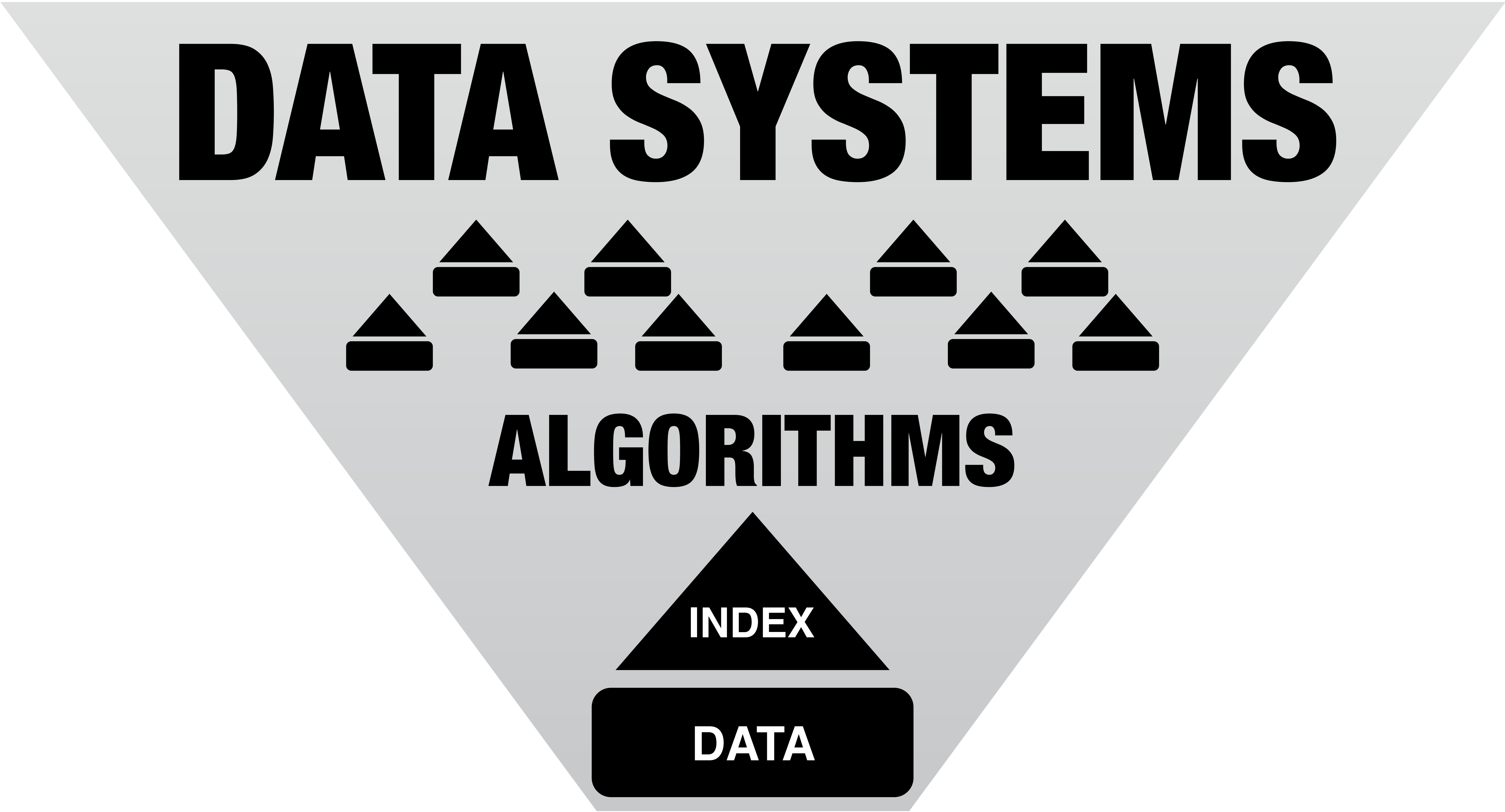
Advanced Data Management (CSCI 640/490)

Review

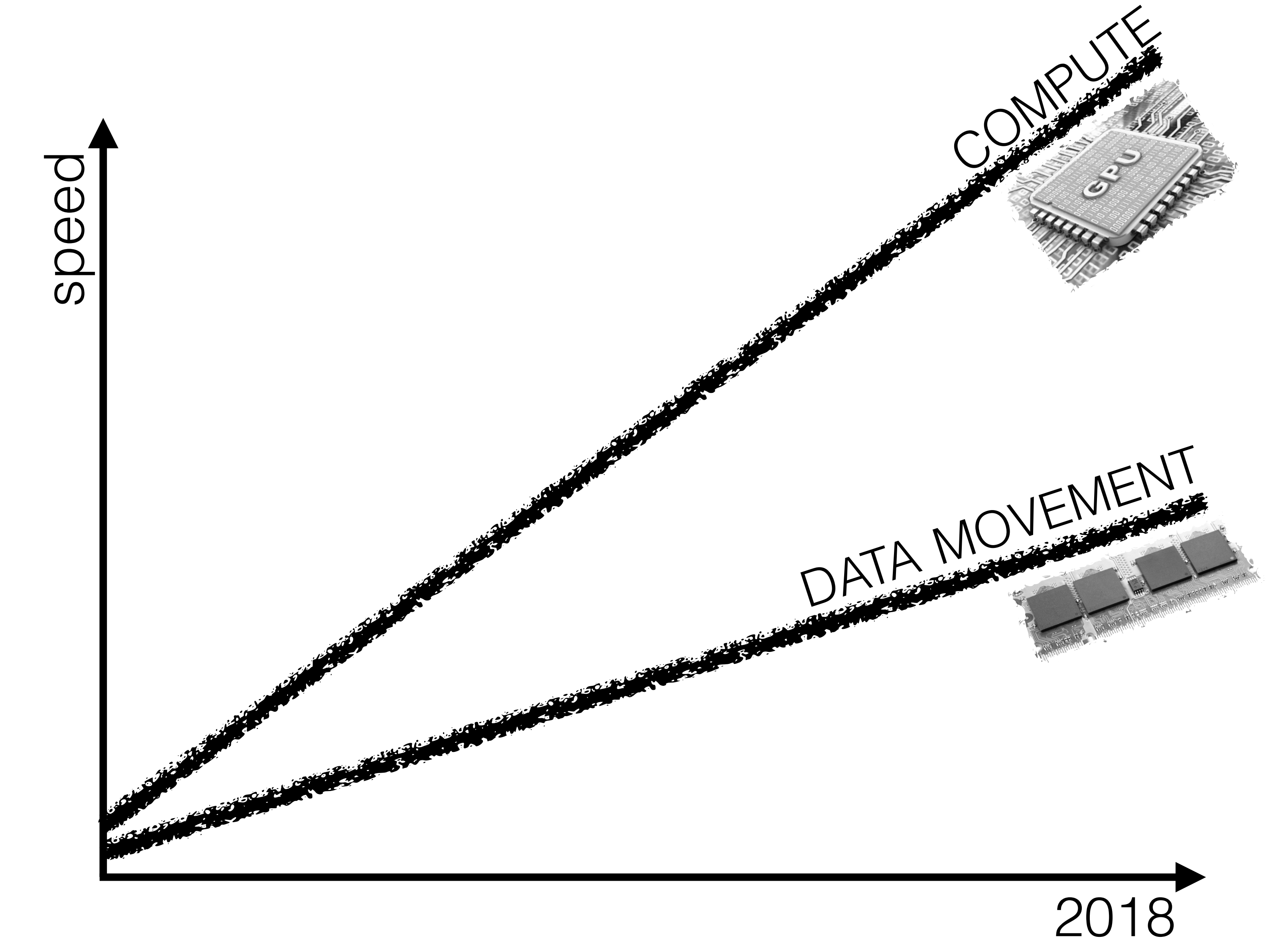
Dr. David Koop

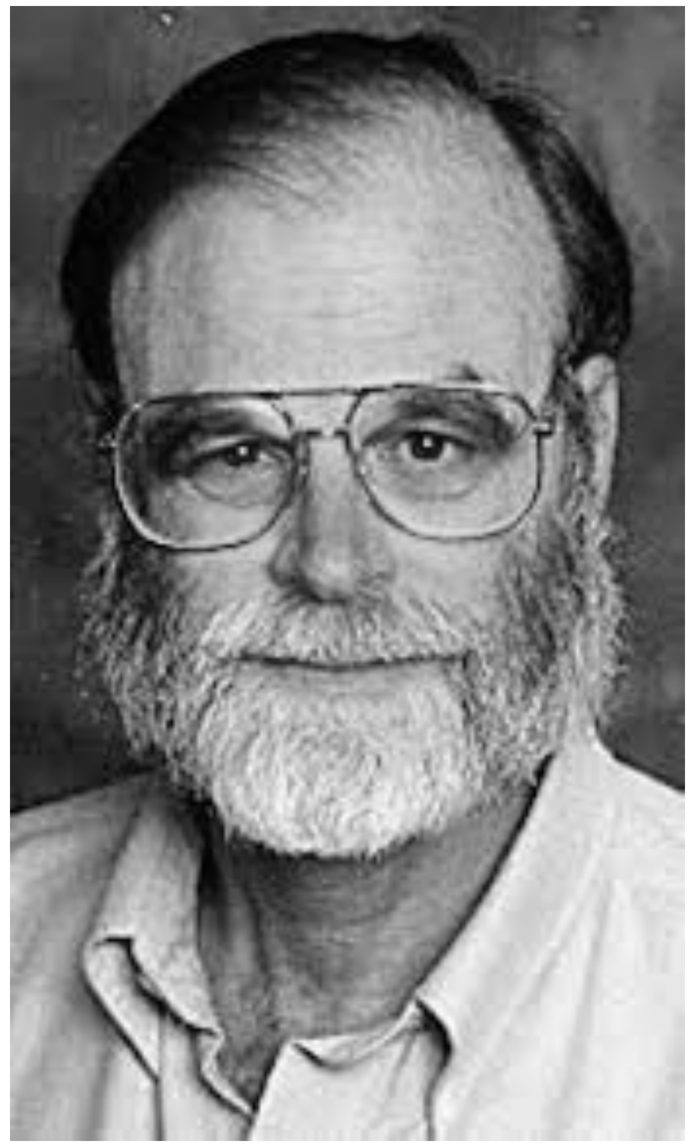
Data systems rely on algorithms



[S. Idreos, 2019]

Data structures define performance



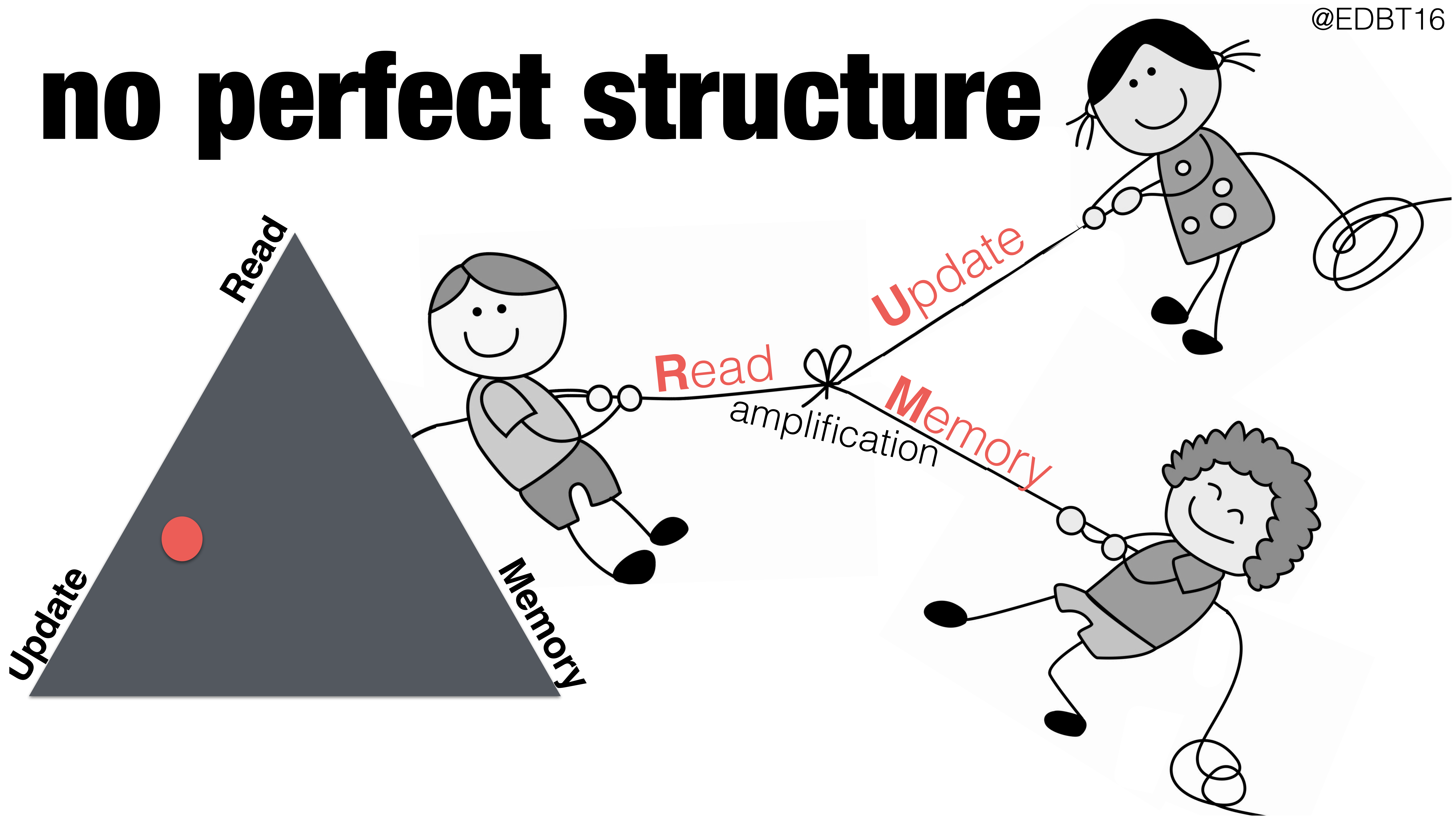


register = this room
caches = this city
memory = nearby city
disk = Pluto

Jim Gray, Turing Award 1998

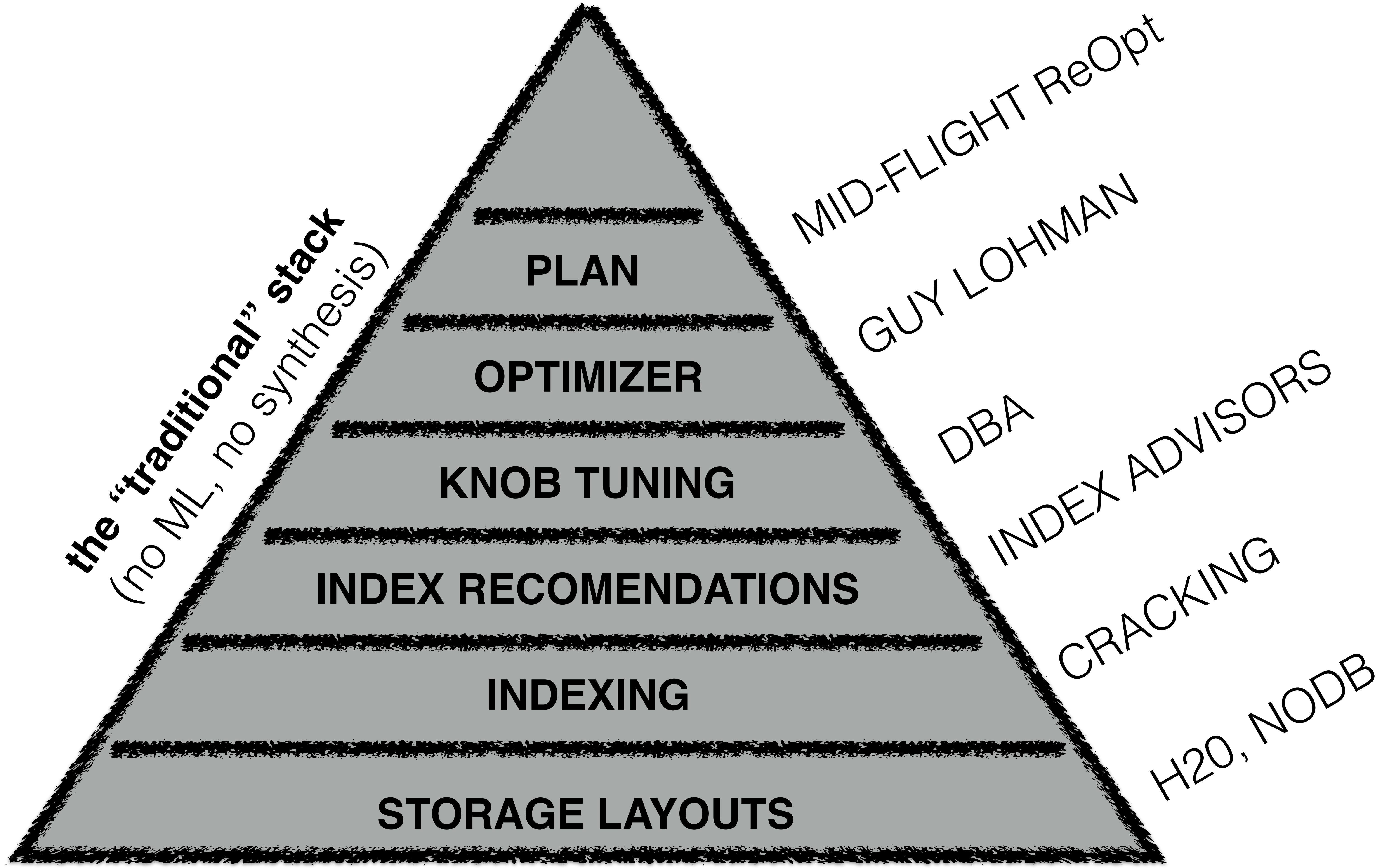
[S. Idreos, 2019]

Tradeoffs in each structure



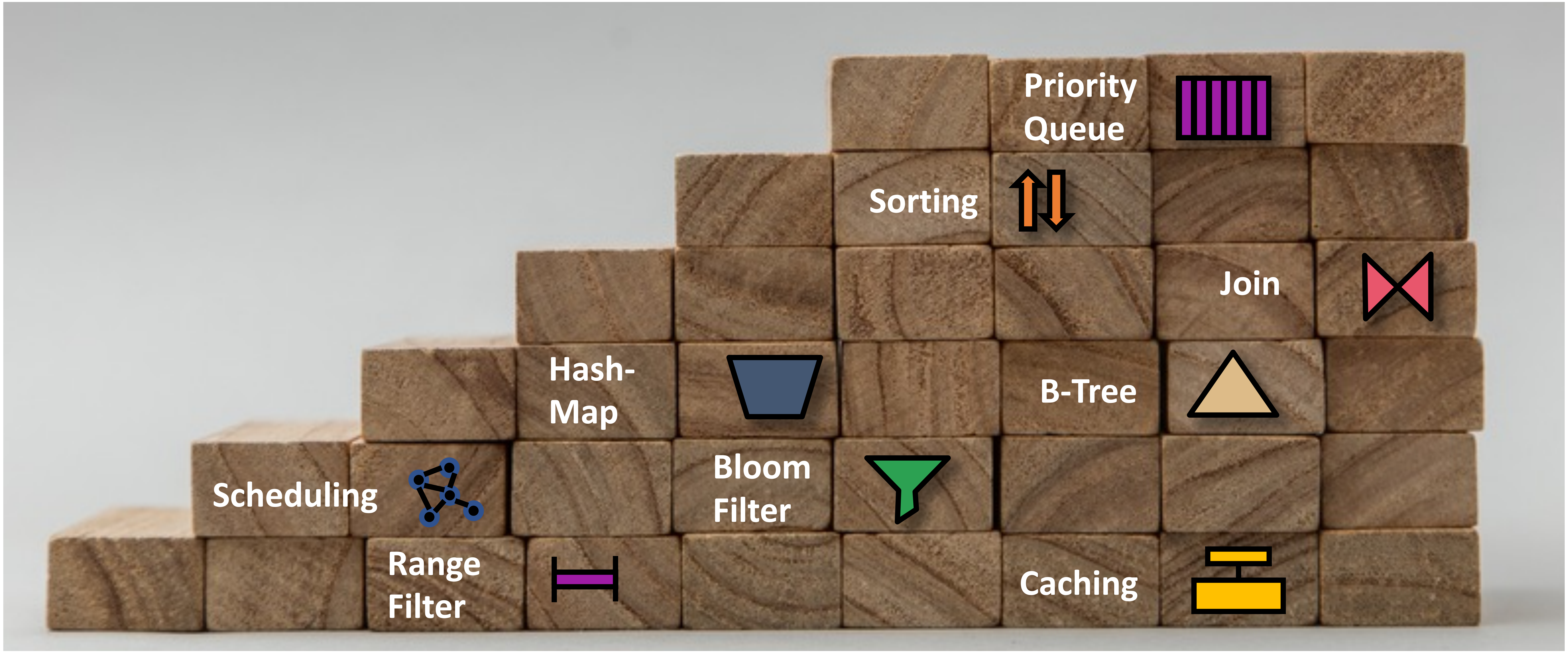
[S. Idreos, 2019]

"Traditional" Database Research

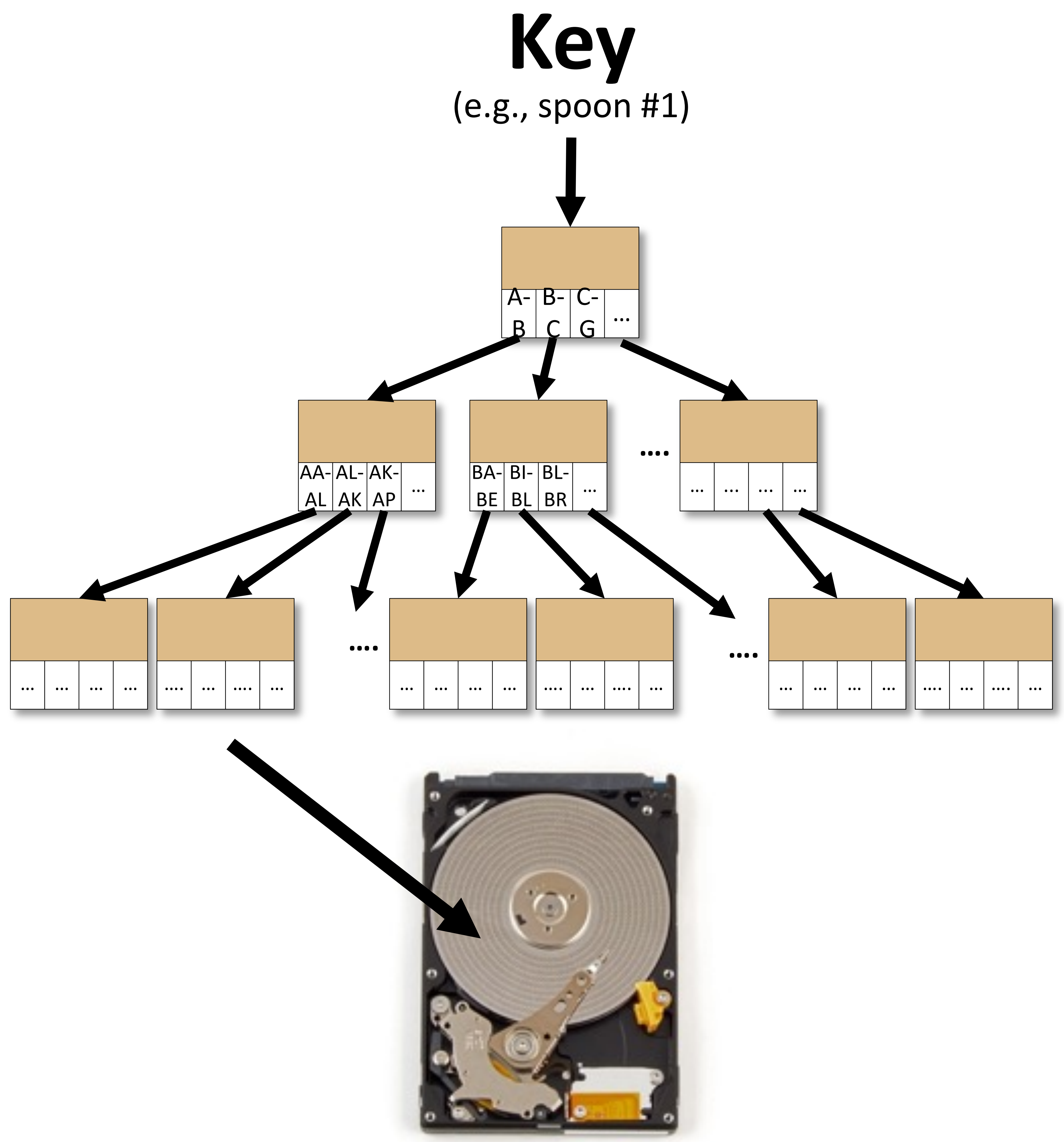


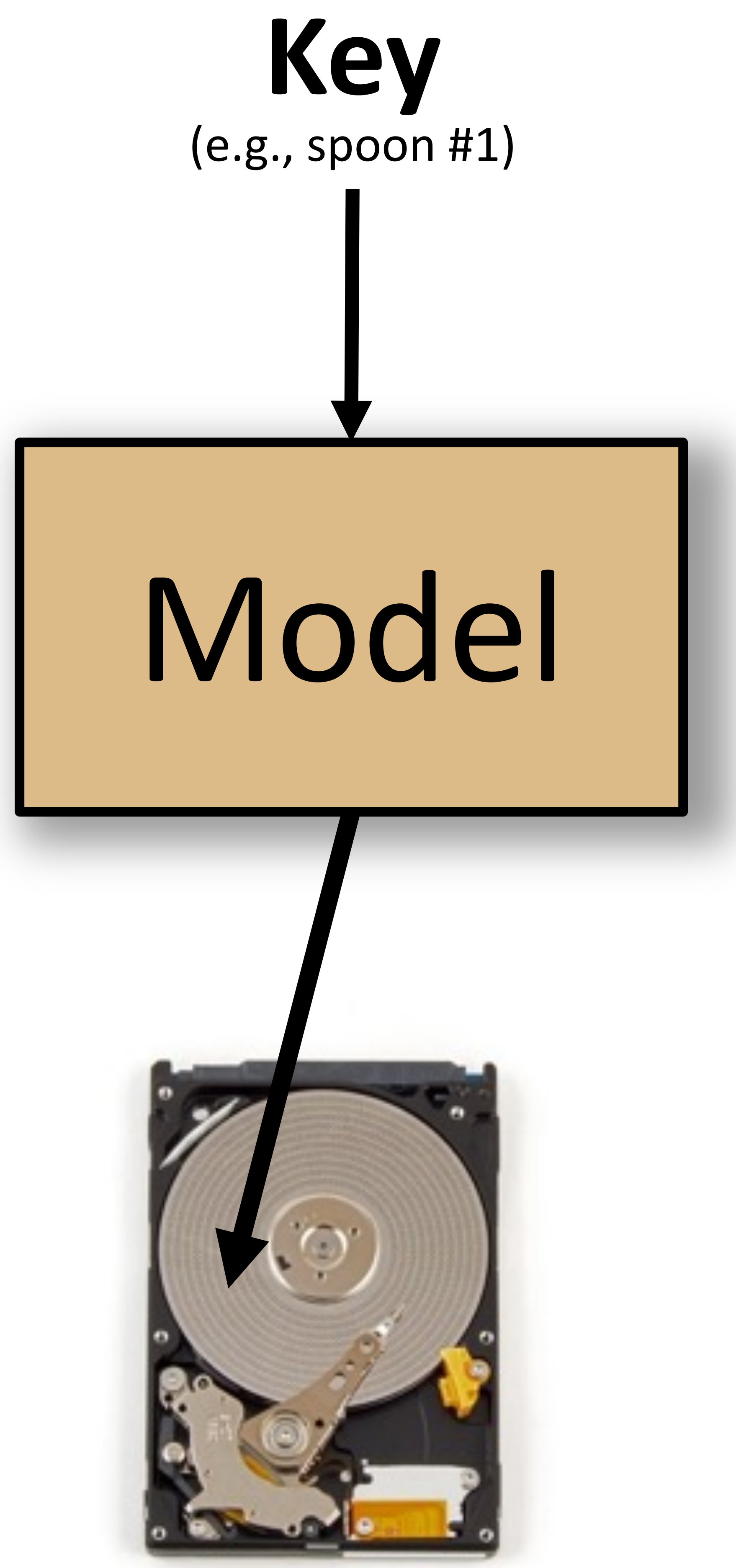
[S. Idreos, 2019]

Learned Data Structures and Algorithms



B-Tree

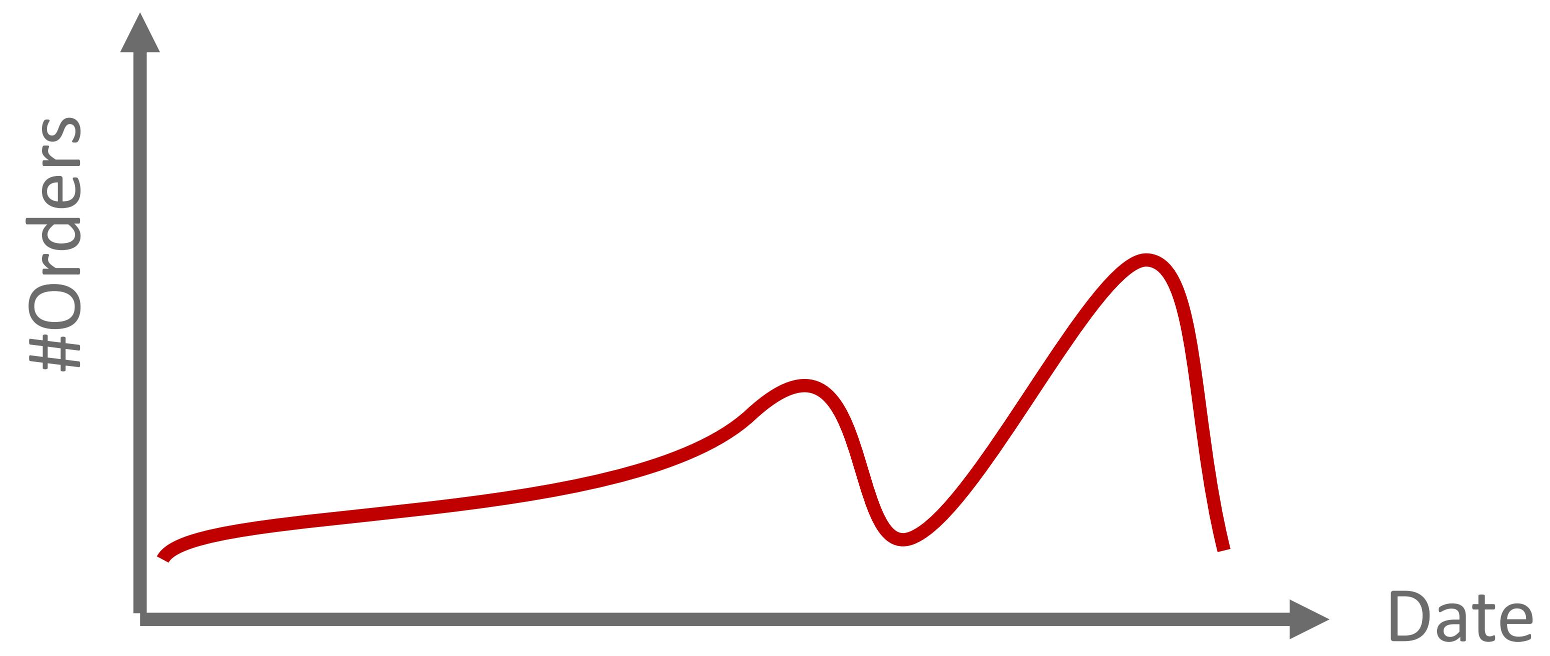




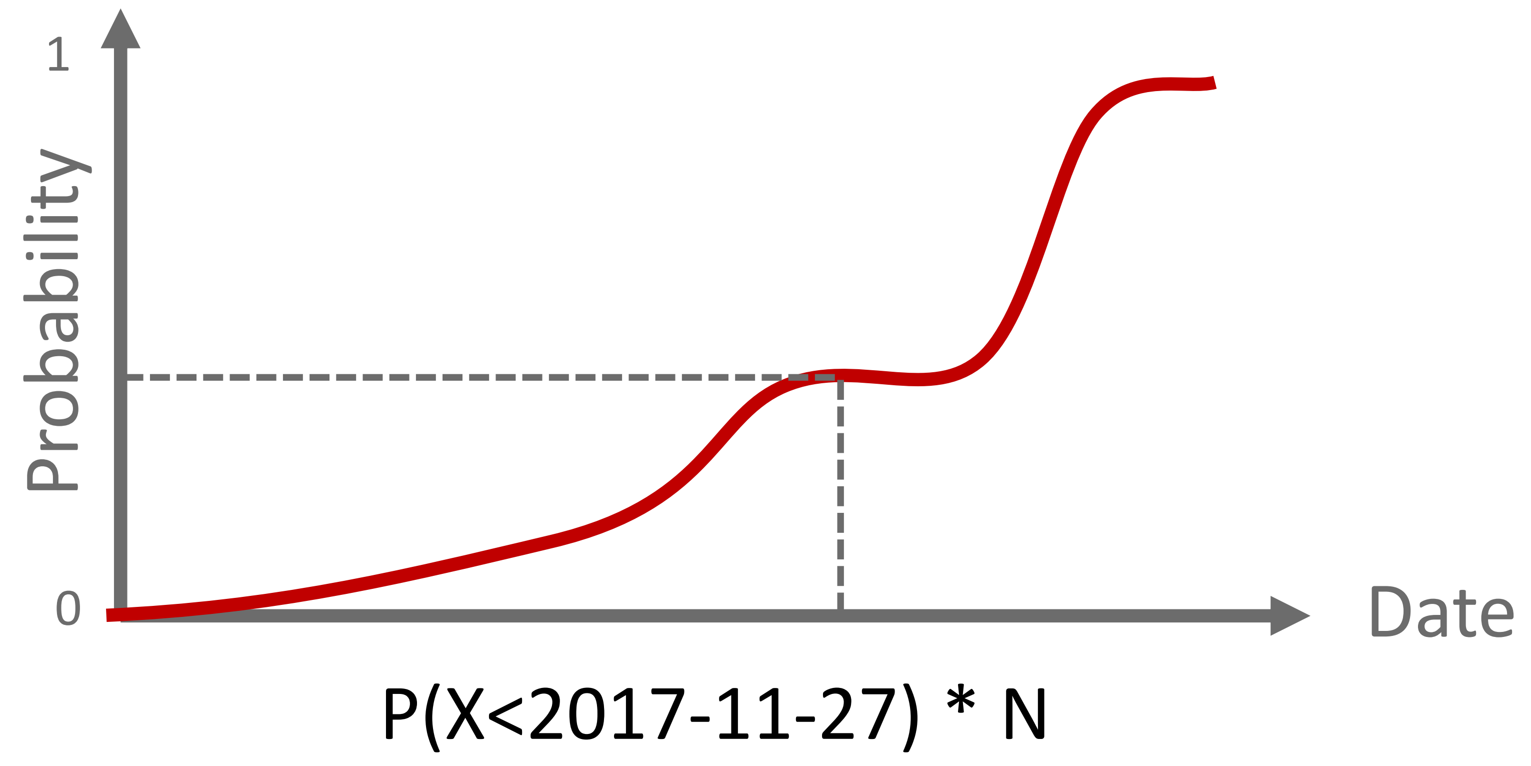
[T. Kraska, 2019]

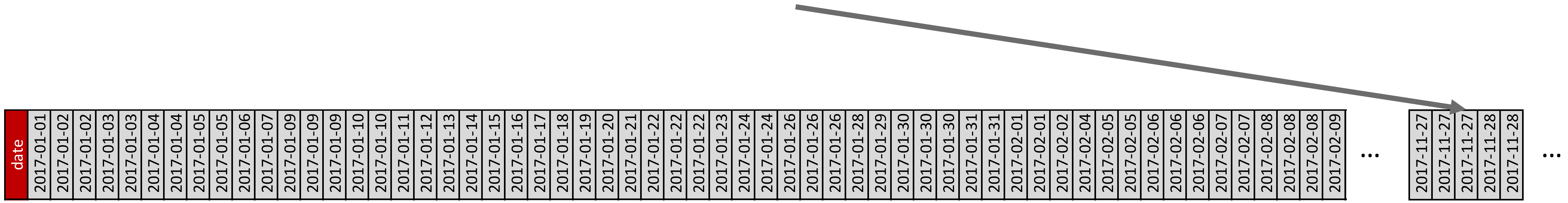
Model to Predict Data's Location on Disk

Frequency Distribution



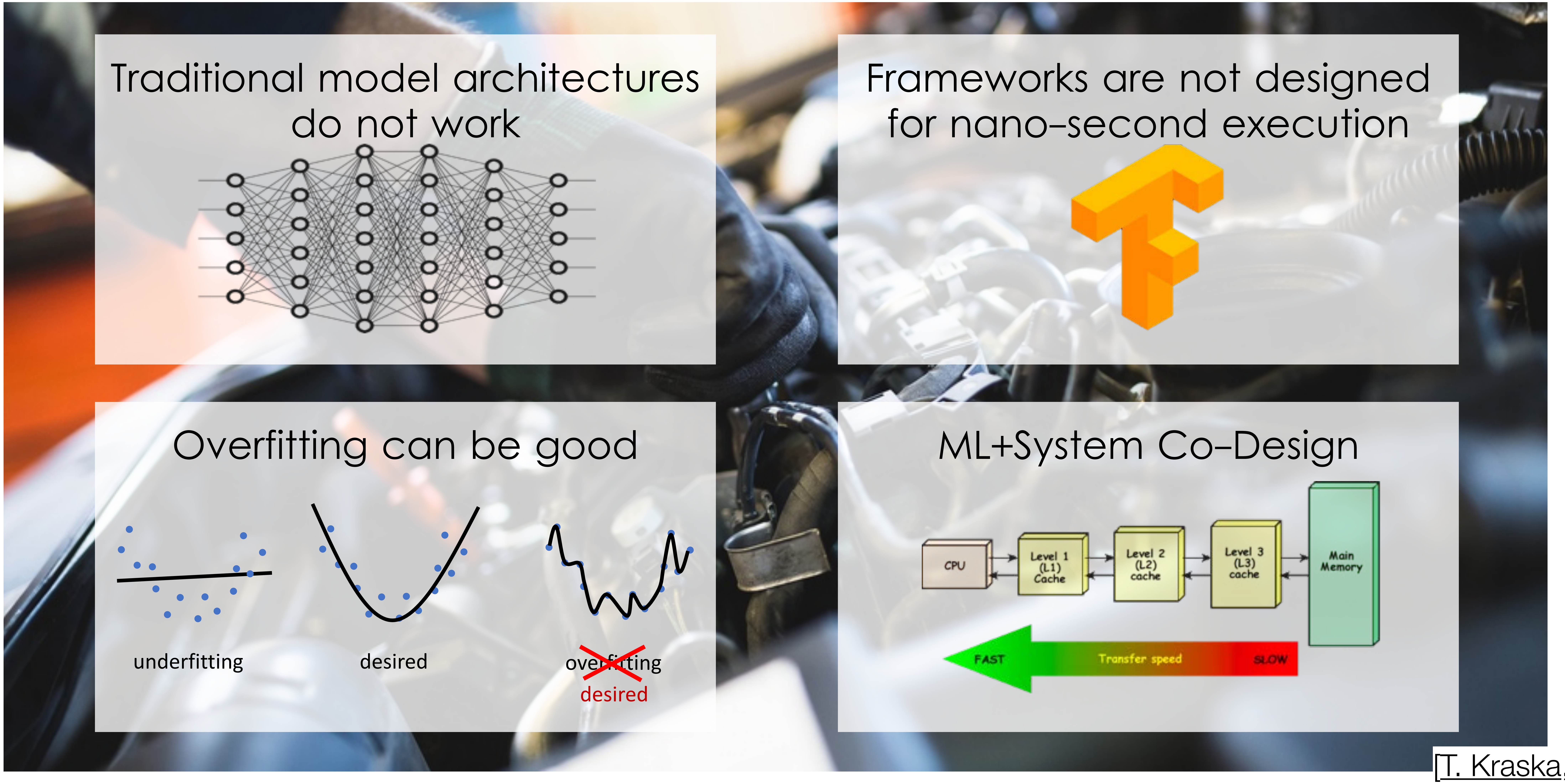
Cumulative Distribution Function (CDF)





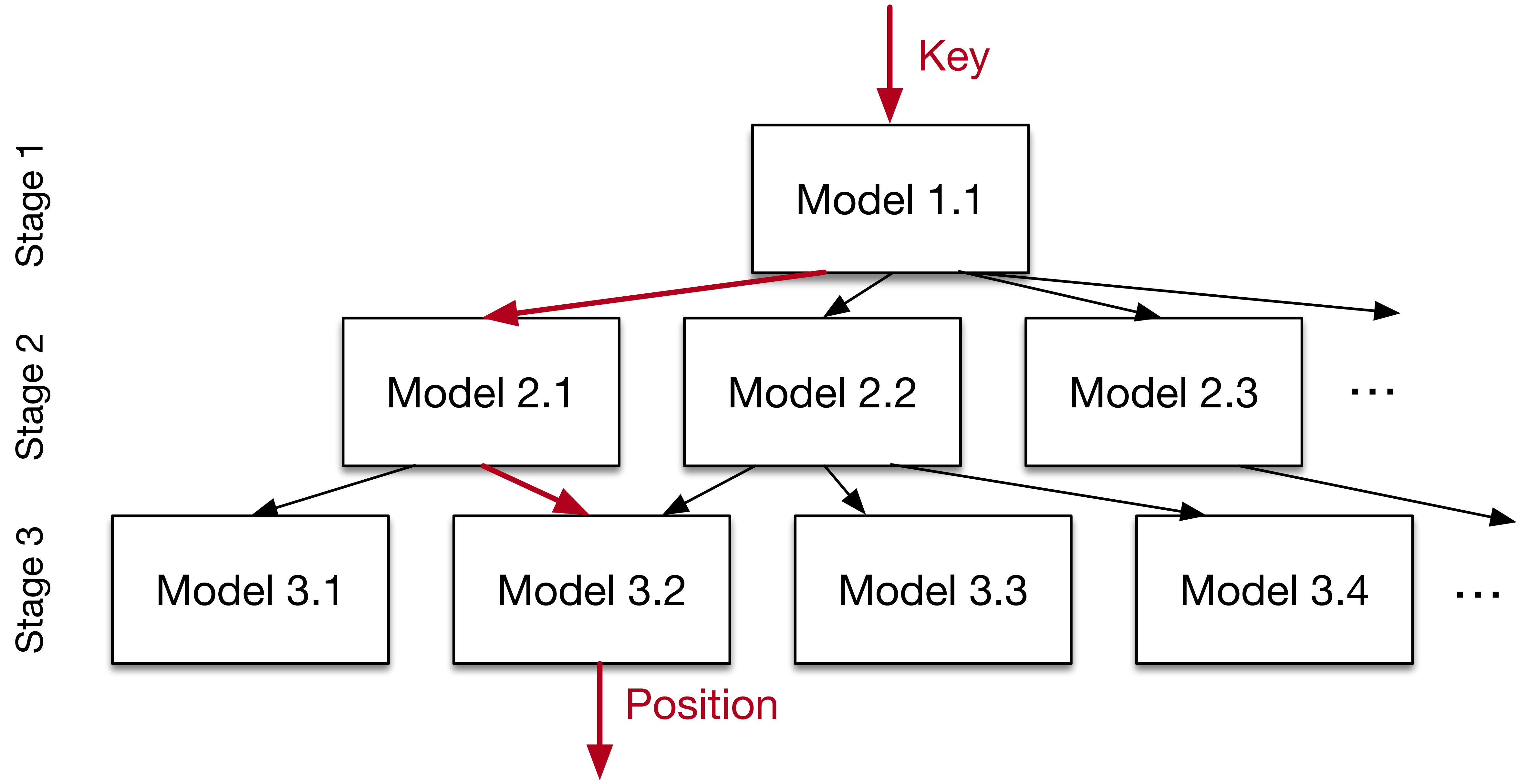
[T. Kraska, 2019]

Challenges



[T. Kraska, 2019]

Recursive Model Index (RMI)

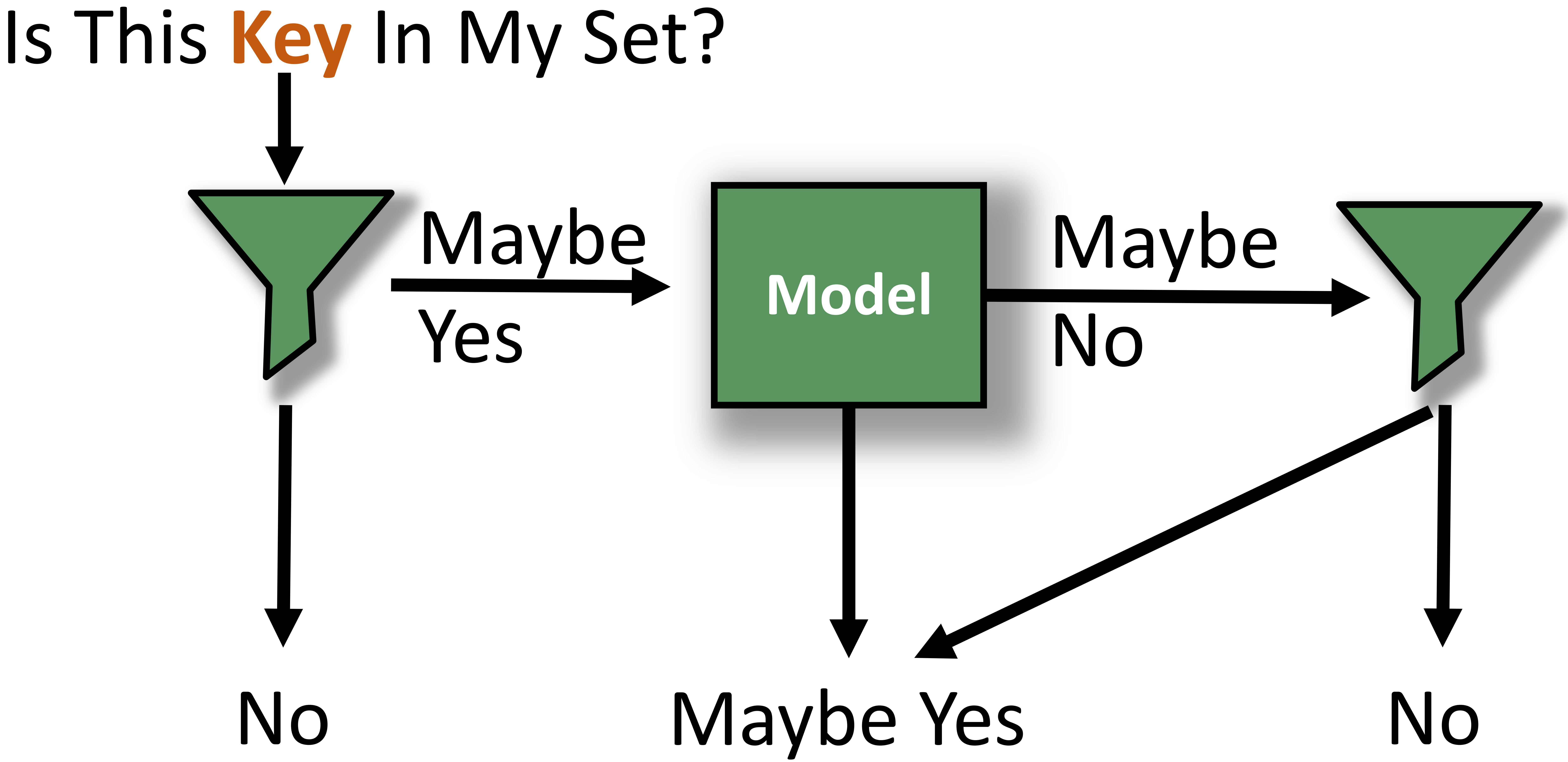


2-Stage RMI with Linear Model

```
pos₀ = a₀ + b₀ * key
pos₁ = m₁[pos₀].a + m₁[pos₀].b * key
record = local-search(key, pos₁)
```

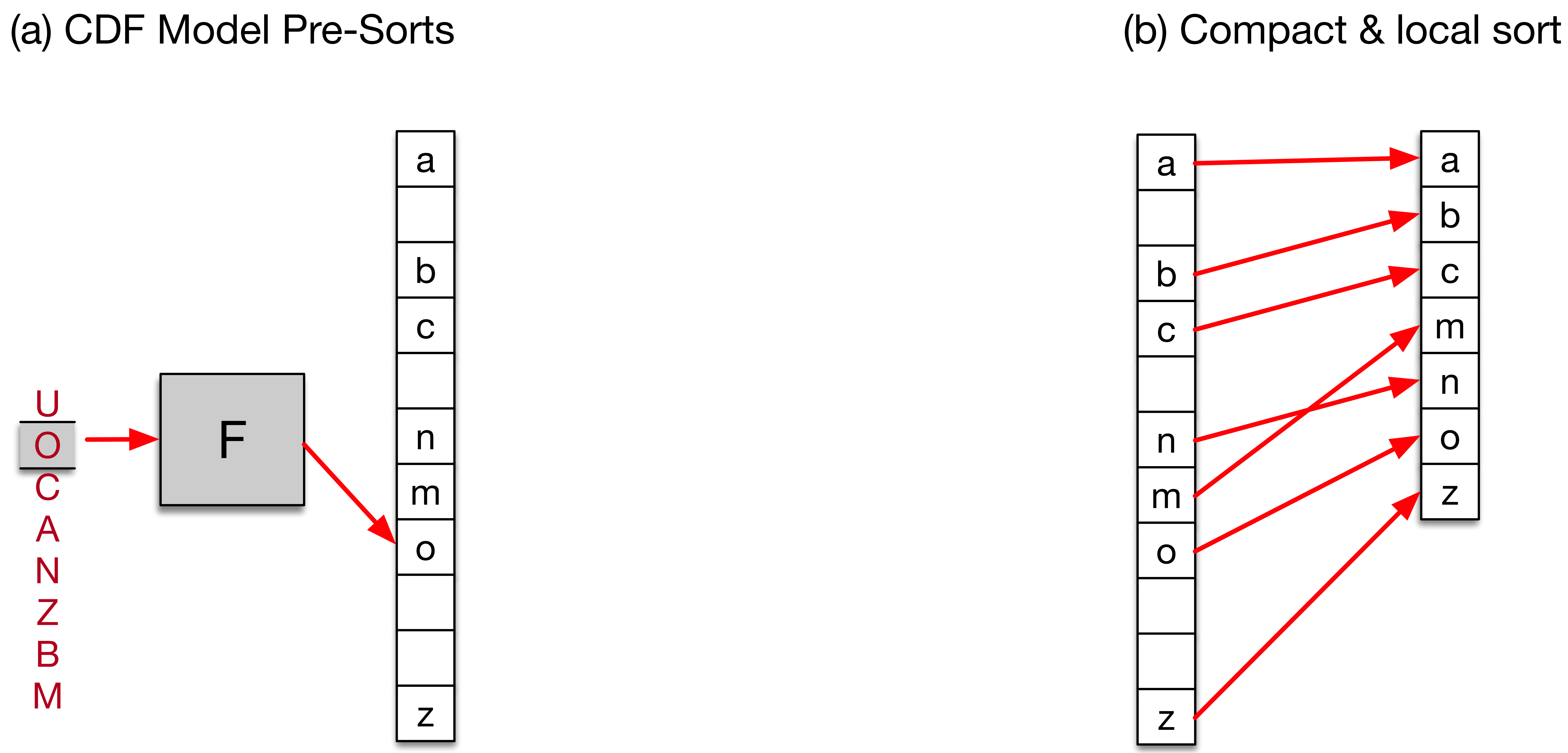
[T. Kraska, 2019]

Sandwiched Bloom Filter



[M. Mitzenmacher, 2018 via T. Kraska, 2019]

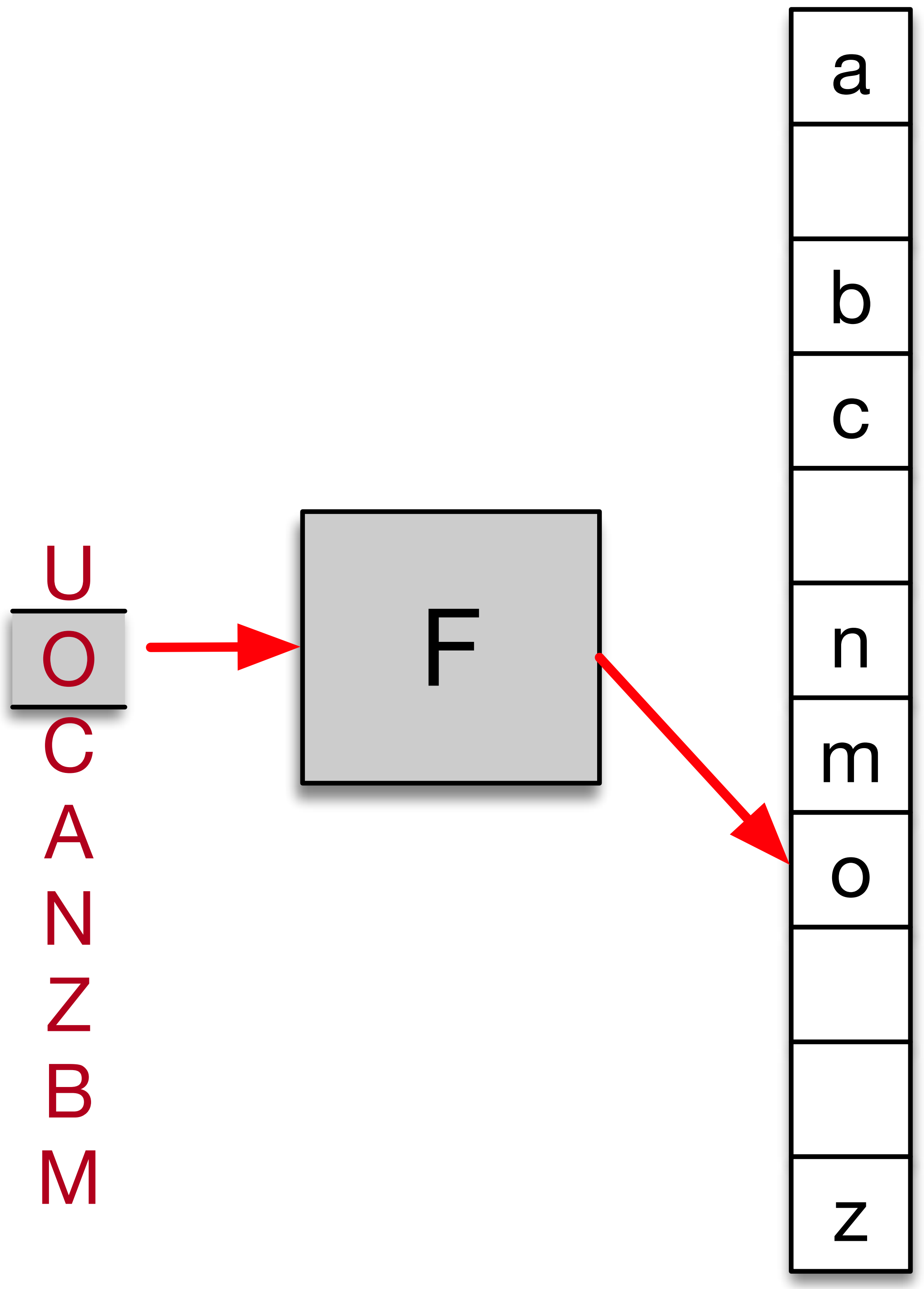
Sorting

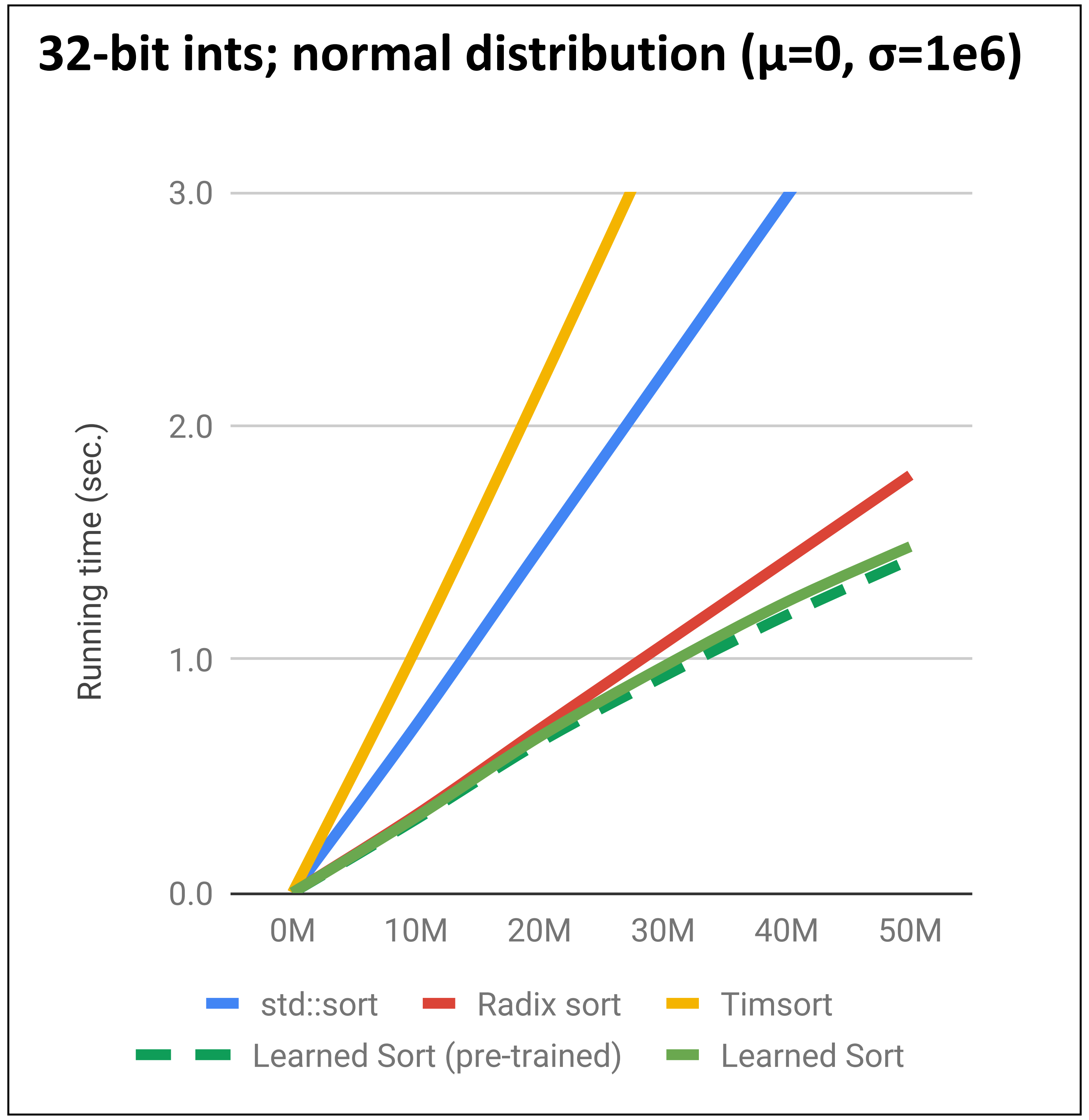


[T. Kraska, 2019]

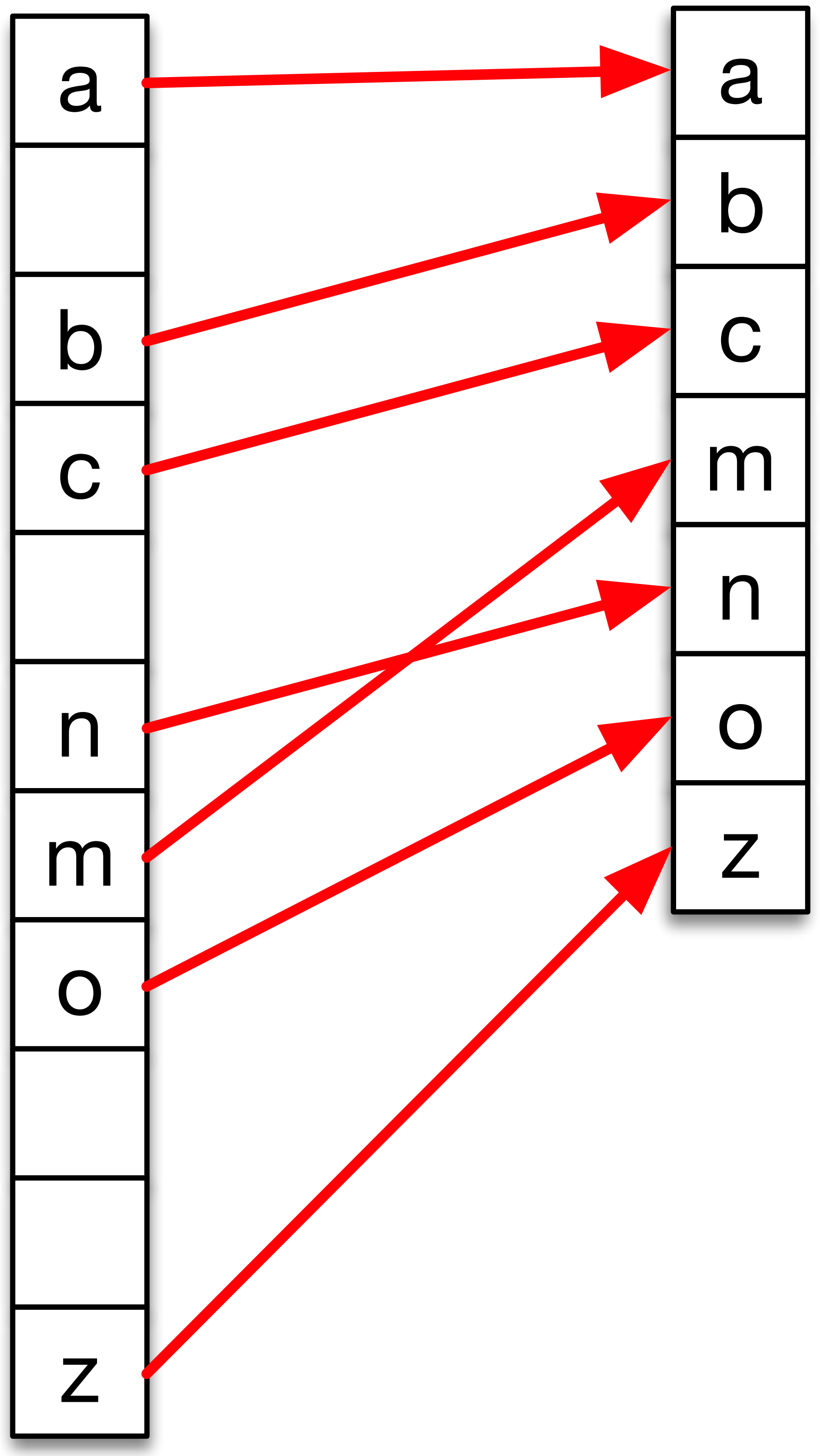
Sorting

(a) CDF Model Pre-Sorts



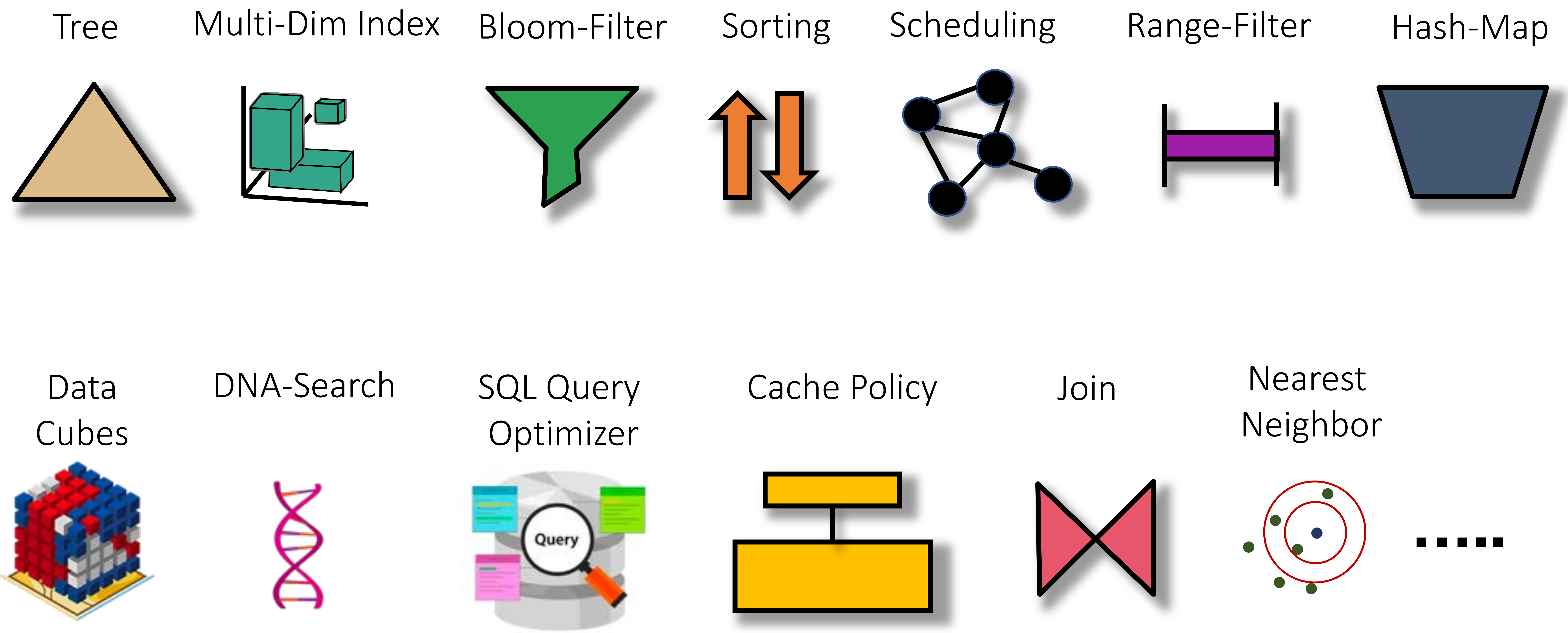


(b) Compact & local sort



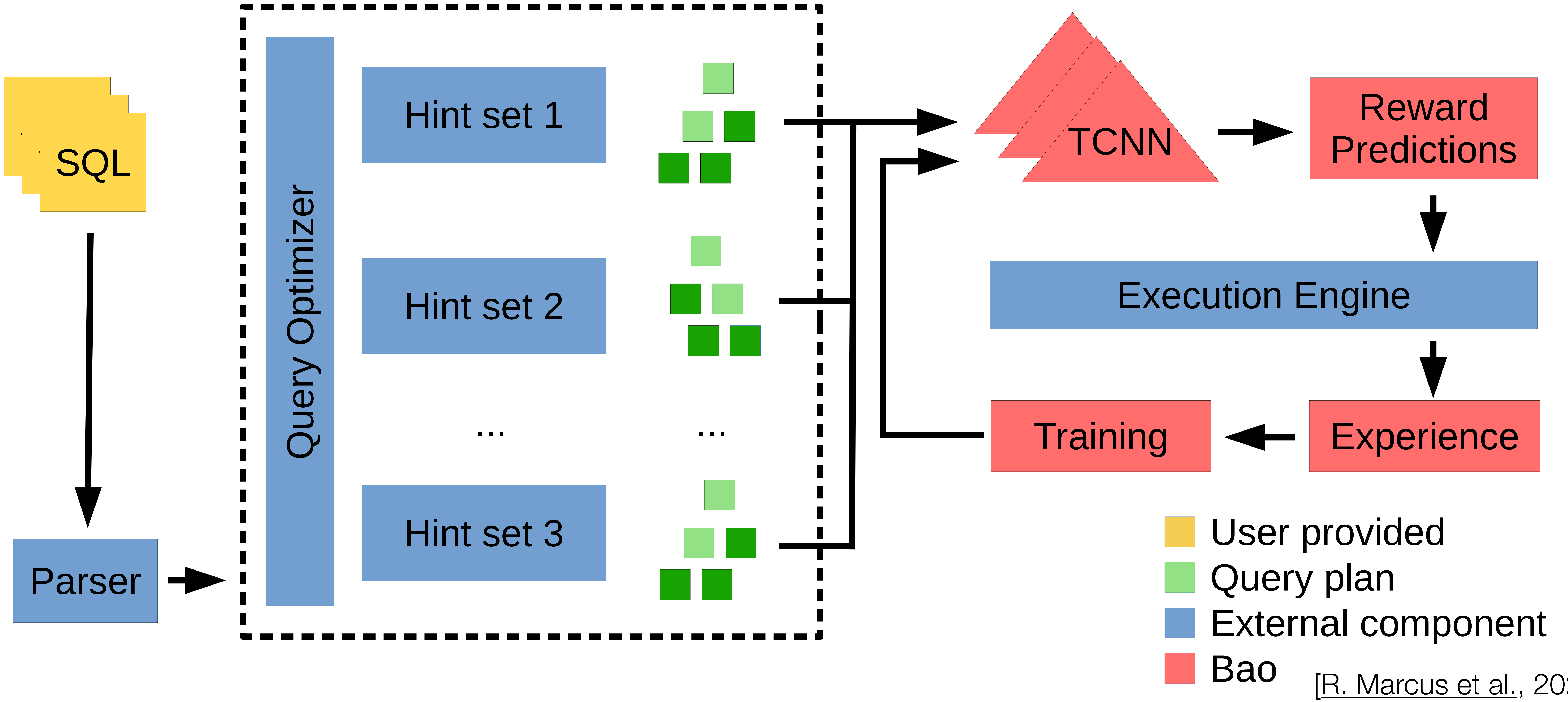
[T. Kraska, 2019]

More…



[T. Kraska, 2019]

Query Optimization



[R. Marcus et al., 2021]

Final Exam

- Wednesday, May 8, **8:00**-9:50am, PM 252
- Similar format
- More comprehensive (questions from topics covered in Test 1 & 2)
- Will also have questions from graph/spatial/temporal data, provenance, reproducibility, machine learning

Questions?

Review

Re

What did we do this semester?

What's involved in dealing with data?

Data Acquisition
- Structured data
- Unstructured data
- Event processing
- Sensor networks
- Protocols
- Real-time
- Data streams
- Multimodality

Data Analysis
- Stream mining
- Semantic analysis
- Machine learning
- Information extraction
- Linked Data
- Data discovery
- ‘Whole world’ semantics
- Ecosystems
- Community data analysis
- Cross-sectorial data analysis

Data Curation
- Data Quality
- Trust / Provenance
- Annotation
- Data validation
- Human-Data Interaction
- Top-down/Bottom-up
- Community / Crowd
- Human Computation
- Curation at scale
- Incentivisation
- Automation
- Interoperability

Data Storage
- In-Memory DBs
- NoSQL DBs
- NewSQL DBs
- Cloud storage
- Query Interfaces
- Scalability and Performance
- Data Models
- Consistency, Availability, Partition-tolerance
- Security and Privacy
- Standardization

Data Usage
- Decision support
- Prediction
- In-use analytics
- Simulation
- Exploration
- Visualisation
- Modeling
- Control
- Domain-specific usage

[Big Data Value Chain, Curry et al., 2014]

Python!

- Just assign expressions to variables, no typing

```
a = 12
a = "abc"
b = a + "de"
```

- Functions defined using def, called using parenthesis:

```
def hello(name1="Joe", name2="Jane"):
    print(f"Hello {name1} and {name2}")
hello(name2="Mary")
```

- Always indent blocks (if-else-elif, while, for, etc.):

```
z = 20
if x > 0:
    if y > 0:
        z = 100
else:
    z = 10
```

Python Containers

- List: `[1,"abc",12.34]`
- Tuple: `(1, "abc", 12.34)`
- Indexing/Slicing:
  - `x[0], x[:-1], x[1:2], x[::2]`
- Set: `{1, "abc", 12.34}`
- Dictionary: `{'x': 1, 'y': "abc", 'z': 12.34}`
- Mutable vs. Immutable
- Stored by reference
- Iterators: objects that traverse containers, just know how to get next element
- You cannot index/slice an iterator (`d.values()[-1]` doesn't work)

Comprehensions

- List Comprehensions:
  - `squares = [i**2 for i in range(10)]`
- Dictionary Comprehensions:
  - `squares = {i: i**2 for i in range(10)}`
- Set Comprehensions:
  - `squares = {i**2 for i in range(10)}`
- Comprehensions allow filters:
  - `squares = [i**2 for i in range(10) if i % 2 == 0]`

JupyterLab

- An interactive, configurable programming environment
- Supports many activities including notebooks
- Runs in your web browser
- Notebooks:
  - Originally designed for Python
  - Supports other languages, too
  - Displays results (even interactive maps) inline
  - You decide how to divide code into executable cells
  - Shift+Enter to execute a cell

Relational Algebra

- Definition: A procedural language consisting of a set of operations that take one or two relations as input and produce a new relation as their result.
- Six basic operators
  - select: σ
  - project: $\prod$
  - union: $\cup$
  - set difference: –
  - Cartesian product: x
  - rename: ρ

[A. Silberschatz et al.]

Components of SQL

- **Data Definition Language (DDL)**: the specification of information about relations, including schema, types, integrity constraints, indices, storage
- **Data Manipulation Language (DML)**: provides the ability to query information from the database and to insert tuples into, delete tuples from, and modify tuples in the database.
- An SQL relation is defined using the create table command:

`create table r` $(A_1\ D_1,\ A_2\ D_2,\ \ldots,\ A_n\ D_n,\ (C_1),\ \ldots,\ (C_k))$

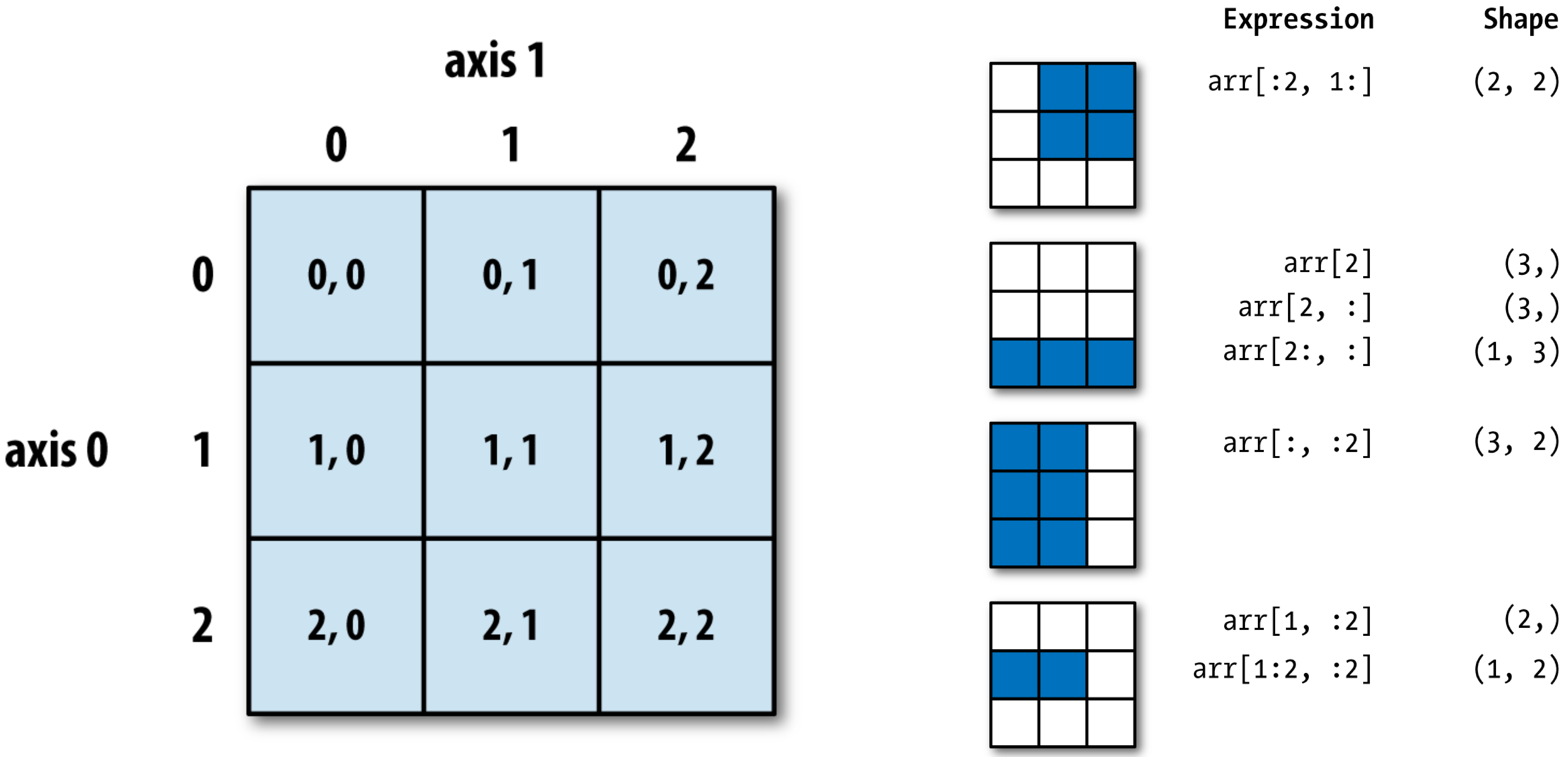
- A typical SQL query has the form:

`select` $A_1, A_2, \ldots, A_n$
`from` $r_1, r_2, \ldots, r_m$
`where` P

- A_i is an **attribute**
- D_i is the **data type**
- r_i represents a **relation**
- P is a **predicate**

[A. Silberschatz et al.]

NumPy arrays and slicing

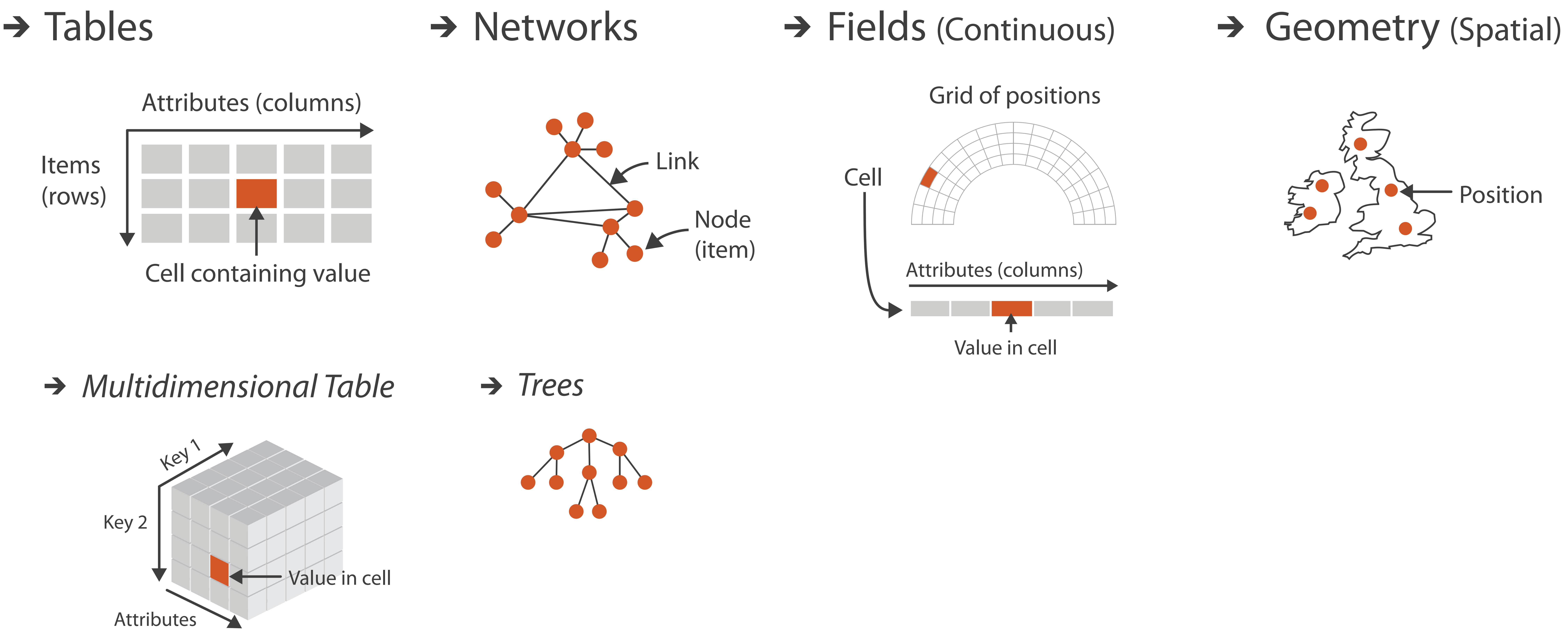


[W. McKinney, Python for Data Analysis]

Boolean Indexing

- `names == 'Bob'` gives back booleans that represent the element-wise comparison with the array `names`
- Boolean arrays can be used to index into another array:
  - `data[names == 'Bob']`
- Can even mix and match with integer slicing
- Can do boolean operations (`&`, `|`) between arrays (just like addition, subtraction)
  - `data[(names == 'Bob') | (names == 'Will')]`
- Note: `or` and `and` do not work with arrays
- We can set values too! `data[data < 0] = 0`

What is Data?



[Munzner (ill. Maguire), 2014]

Categorial, Ordinal, and Quantitative

A	B	C	S	T	U
Order ID	Order Date	Order Priority	Product Container	Product Base Margin	Ship Date
3	10/14/06	5-Low	Large Box	0.8	10/21/06
6	2/21/08	4-Not Specified	Small Pack	0.55	2/22/08
32	7/16/07	2-High	Small Pack	0.79	7/17/07
32	7/16/07	2-High	Jumbo Box	0.72	7/17/07
32	7/16/07	2-High	Medium Box	0.6	7/18/07
32	7/16/07	2-High	Medium Box	0.65	7/18/07
35	10/23/07	4-Not Specified	Wrap Bag	0.52	10/24/07
35	10/23/07	4-Not Specified	Small Box	0.58	10/25/07
36	11/3/07	1-Urgent	Small Box	0.55	11/3/07
65	3/18/07	1-Urgent	Small Pack	0.49	3/19/07
66	1/20/05	5-Low	Wrap Bag	0.56	1/20/05
69	6/4/05	4-Not Specified	Small Pack	0.44	6/6/05
69	6/4/05	4-Not Spec		0.6	6/6/05
70	12/18/06	5-Low		0.59	12/23/06
70	12/18/06	5-Low		0.82	12/23/06
96	4/17/05	2-High		0.55	4/19/05
97	1/29/06	3-Medium		0.38	1/30/06
129	11/19/08	5-Low		0.37	11/28/08
130	5/8/08	2-High	Small Box	0.37	5/9/08
130	5/8/08	2-High	Medium Box	0.38	5/10/08
130	5/8/08	2-High	Small Box	0.6	5/11/08
132	6/11/06	3-Medium	Medium Box	0.6	6/12/06
132	6/11/06	3-Medium	Jumbo Box	0.69	6/14/06
134	5/1/08	4-Not Specified	Large Box	0.82	5/3/08
135	10/21/07	4-Not Specified	Small Pack	0.64	10/23/07
166	9/12/07	2-High	Small Box	0.55	9/14/07
193	8/8/06	1-Urgent	Medium Box	0.57	8/10/06
194	4/5/08	3-Medium	Wrap Bag	0.42	4/7/08

quantitative
ordinal
categorical

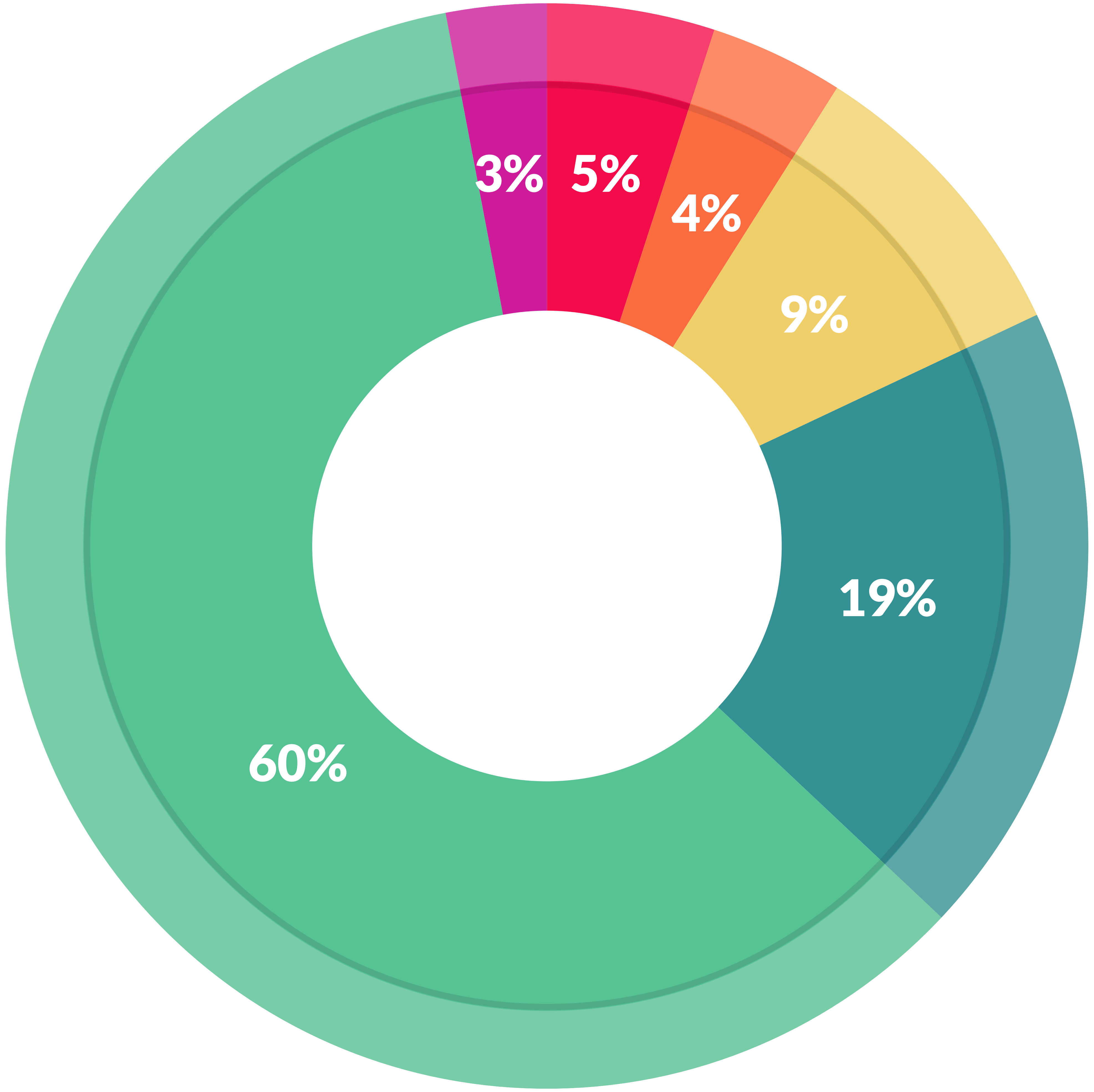
Pandas and Data Frames

	Inspection ID	DBA Name	AKA Name	License #	Facility Type	Risk	Address	City	State	Zip	Inspection Date	Inspection Type	Results
0	2356580	UNCOOKED LLC	UNCOOKED LLC	2709319.0	NaN	All	210 N CARPENTER ST	CHICAGO	IL	60607.0	01/13/2020	License	Not Ready
1	2356551	MOJO 33 NORTH LASALLE LLC	MOJO 33 NORTH LASALLE LLC	2689550.0	Restaurant	Risk 1 (High)	33 N LA SALLE ST	CHICAGO	IL	60602.0	01/13/2020	License Re-Inspection	Pass
2	2356492	LA BIZNAGA #2	LA BIZNAGA #2	2708992.0	NaN	Risk 1 (High)	2949 W BELMONT AVE	CHICAGO	IL	60618.0	01/10/2020	License	Not Ready
3	2356432	LAS TABLAS	LAS TABLAS	1617900.0	Restaurant	Risk 1 (High)	4920 W IRVING PARK RD	CHICAGO	IL	60641.0	01/09/2020	Canvass	Pass
4	2356423	GIORDANO'S OF BEVERLY	GIORDANO'S OF BEVERLY	2074456.0	Restaurant	Risk 1 (High)	9613 S WESTERN AVE	CHICAGO	IL	60643.0	01/09/2020	Canvass	Pass
...	...	...	...	...	...	...	...	...	...	...	...	...	...
199687	112321	PANDA EXPRESS #236	PANDA EXPRESS #236	1801495.0	Restaurant	Risk 1 (High)	77 W JACKSON BLVD	CHICAGO	IL	60604.0	02/18/2010	Suspected Food Poisoning	Pass
199688	74300	KENNYS RIBS & CHICKEN	UNCLE JOE'S	81030.0	Restaurant	Risk 1 (High)	1453 E HYDE PARK BLVD	CHICAGO	IL	60615.0	02/08/2010	Complaint	Pass
199689	70314	Cafe Marbella	Cafe Marbella	2016764.0	Restaurant	Risk 1 (High)	5527-5531 N Milwaukee AVE	CHICAGO	IL	60630.0	01/28/2010	License Re-Inspection	Pass
199690	78309	WALGREENS # 07876	WALGREENS # 07876	2004292.0	Grocery Store	Risk 3 (Low)	7544 S STONY ISLAND AVE	CHICAGO	IL	60649.0	02/18/2010	TASK FORCE LIQUOR 1474	Pass
199691	150209	YSABEL'S FILIPINO CUISINE	YSABEL'S GRILL ASIAN CUISINE	2013419.0	Restaurant	Risk 1 (High)	4908 W Irving Park RD	CHICAGO	IL	60641.0	01/12/2010	License Re-Inspection	Pass

199692 rows × 17 columns

- Data Frames are tables with many database-like operations
- Index shared across all columns
- Can select, project, merge (join), and more
- Read and write many file formats

How do data scientists spend their time?



What data scientists spend the most time doing

- *Building training sets: 3%*
- *Cleaning and organizing data: 60%*
- *Collecting data sets; 19%*
- *Mining data for patterns: 9%*
- *Refining algorithms: 4%*
- *Other: 5%*

[CrowdFlower Data Science Report, 2016]

Data Wrangling

- Automated Transformation Suggestions
- Editable Natural Language Explanations
  - Fill **Bangladesh** by **copying** values from **above**
  - Fill **Bangladesh** by [averaging / ✓ copying / interpolating] values from **above**
  - Fill **Bangladesh** by **averaging** the **5** values from **above**
- Visual Transformation Previews
- Transformation History

split	split1	split2	split3	split4
	2004	2004	2004	2003
STATE	Participation Rate 2004	Mean SAT I Verbal	Mean SAT I Math	Participation Rate
New York	87	497	510	82
Connecticut	85	515	515	84
Massachusetts	85	518	523	82
New Jersey	83	501	514	85
New Hampshire	80	522	521	75
D.C.	77	489	476	77
Maine	76	505	501	70
Pennsylvania	74	501	502	73
Delaware	73	500	499	73
Georgia	73	494	493	66

split	fold	fold1	value
New York	2004	Participation Rate 2004	87
New York	2004	Mean SAT I Verbal	497
New York	2004	Mean SAT I Math	510
New York	2003	Participation Rate 2003	82
New York	2003	Mean SAT I Verbal	496
New York	2003	Mean SAT I Math	510
Connecticut	2004	Participation Rate 2004	85
Connecticut	2004	Mean SAT I Verbal	515
Connecticut	2004	Mean SAT I Math	515
Connecticut	2003	Participation Rate 2003	84
Connecticut	2003	Mean SAT I Verbal	512

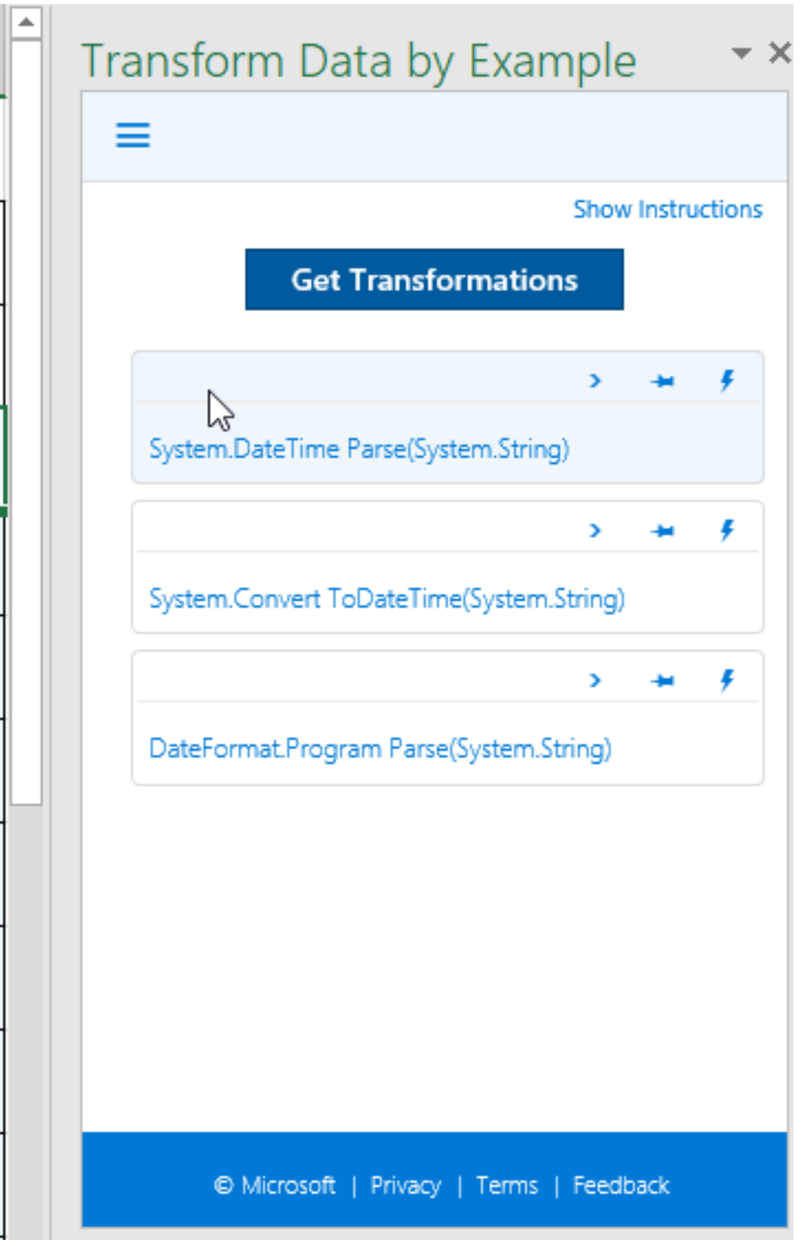
[S. Kandel et al., 2011]

TDE: Transform Data by Example

C	D
Transaction Date	**output**
Wed, 12 Jan 2011	2011-01-12-Wednesday
Thu, 15 Sep 2011	2011-09-15-Thursday
Mon, 17 Sep 2012	
2010-Nov-30 11:10:41	
2011-Jan-11 02:27:21	
2011-Jan-12	
2010-Dec-24	
9/22/2011	
7/11/2012	
2/12/2012	

→

C	D
Transaction Date	**output**
Wed, 12 Jan 2011	2011-01-12-Wednesday
Thu, 15 Sep 2011	2011-09-15-Thursday
Mon, 17 Sep 2012	2012-09-17-Monday
2010-Nov-30 11:10:41	2010-11-30-Tuesday
2011-Jan-11 02:27:21	2011-01-11-Tuesday
2011-Jan-12	2011-01-12-Wednesday
2010-Dec-24	2010-12-24-Friday
9/22/2011	2011-09-22-Thursday
7/11/2012	2012-07-11-Wednesday
2/12/2012	2012-02-12-Sunday



[Y. He et al., 2018]

Transform by Pattern: Automating Unify/Repair

- Auto-Unify

S-timestamp	S-phone	S-coordinates
2019-12-23	(425) 882-8080	(38°57'N, 95°15'W)
2019-12-24	(425) 882-8080	(38°61'N, 95°21'W)
2019-12-23	(206) 876-1800	(39°19'N, 95°18'W)
2019-12-24	(206) 876-1800	(39°26'N, 95°23'W)
2019-12-23	(206) 903-8010	(39°42'N, 96°38'W)

R-timestamp	R-phone	R-coordinates
Nov. 16 2019	650-853-1300	N37°31′ W122°14′
Nov. 17 2019	650-853-1300	N37°18′ W122°19′
Nov. 16 2019	425-421-1225	N37°48′ W122°17′
Nov. 17 2019	425-421-1225	N37°60′ W123°08′
Nov. 16 2019	650-253-0827	N37°01′ W123°72′

- Auto-Repair

Date	Opponents
January 12, 1997	Venezuela
February 12, 1997	Peru
April 2, 1997	Colombia
1997-06-04	United States
1997-06-11	Chile
1997-06-14	Ecuador

(a) EN-Wiki: Dates

Year	Artist	Issue Price (BU)
1989	John Mardon	$16.25
1990	D.J. Craig	$16.75
1991	D.J. Craig	$16.75
1992	Karsten Smith	17.50
1993	Stewart Sherwood	$17.50
1994	Ian D. Sparkes	$17.95

(b) EN-Wiki: Currency values

Women's winner	Time
Anikó Kálovics	2:31:24
Lenah Cheruiyot	2:27:02
Lenah Cheruiyot	2:33.44
Emily Kimuria	2:28.42
Jane Ekimat	2:32.08

(c) EN-wiki:time

#	Original air date[1]
12	March 23, 2008
13	March 30, 2008
14	April 6, 2008
15	13 April 2008
16	20 April 2008

(d) EN-Wiki: Date

TBP: Learning from Tables

T_1

Name	#	Born	Died
Washington, George	USA President (1)	02/22/1732	12/14/1799
Adams, John	USA President (2), VP (1)	10/30/1735	07/04/1826
Jefferson, Thomas	USA President (3), VP (2)	04/13/1743	07/04/1826
Madison, James	USA President (4)	03/16/1751	06/28/1836
Monroe, James	USA President (5)	04/28/1758	07/04/1851

T_2

Date of birth	President	Birthplace	State† of birth
February 22, 1732	George Washington	Westmoreland County	Virginia†
October 30, 1735	John Adams	Braintree	Massachusetts†

T_3

30.	**George Washington**	–	57y, 10d	22.02.1732	14.12.1799
31.	**John Quincy Adams**	Nat-Rep	57y, 7m, 20d	11.07.1767	23.02.1848
32.	**Thomas Jefferson**	Dem-Rep	57y, 10m, 18d	13.04.1743	04.07.1826
33.	**James Madison**	Dem-Rep	57y, 11m, 15d	16.03.1751	28.06.1836
34.	**James Monroe**	Dem-Rep	58y, 10m, 3d	28.04.1758	04.07.1831

T_4

1.	George Washington	Virginia	Feb. 22, 1732	Dec. 14, 1797
3.	Thomas Jefferson	Virginia	Apr. 13, 1743	July 4, 1826
4.	James Madison	Virginia	Mar. 16, 1751	June 28, 1836
6.	John Quincy Adams	Massachusetts	July 11, 1767	Feb. 23, 1848

T_5

	Name and (party)[1]	Term	State of birth	Born	Died	Religion[2]	Age at inaug.	Age at death
1.	Washington (F)[3]	1789–1797	Va.	2/22/1732	12/14/1799	Episcopalian	57	67
2.	J. Adams (F)	1797–1801	Mass.	10/30/1735	7/4/1826	Unitarian	61	90

T_6

PRESIDENT	BIRTH DATE	BIRTH PLACE	DEATH DATE	LOCATION OF DEATH
George Washington	Feb 22, 1732	Westmoreland Co., Va.	Dec 14, 1799	Mount Vernon, Va.
John Adams	Oct 30, 1735	Quincy, Mass.	July 4, 1826	Quincy, Mass.

Tidy Data

	treatmenta	treatmentb
John Smith	—	2
Jane Doe	16	11
Mary Johnson	3	1

Initial Data

name	trt	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

Tidy Data

	John Smith	Jane Doe	Mary Johnson
treatmenta	—	16	3
treatmentb	2	11	1

Transpose

[H. Wickham, 2014]

AutoSuggest

- Automate "Complex" Data Preparation steps
- Focus on frame transformations (not per-cell transformations)
- Learn from Jupyter Notebooks
- Two Types of Predictions:
  - Single-Operator Prediction
  - Next-Operator Prediction

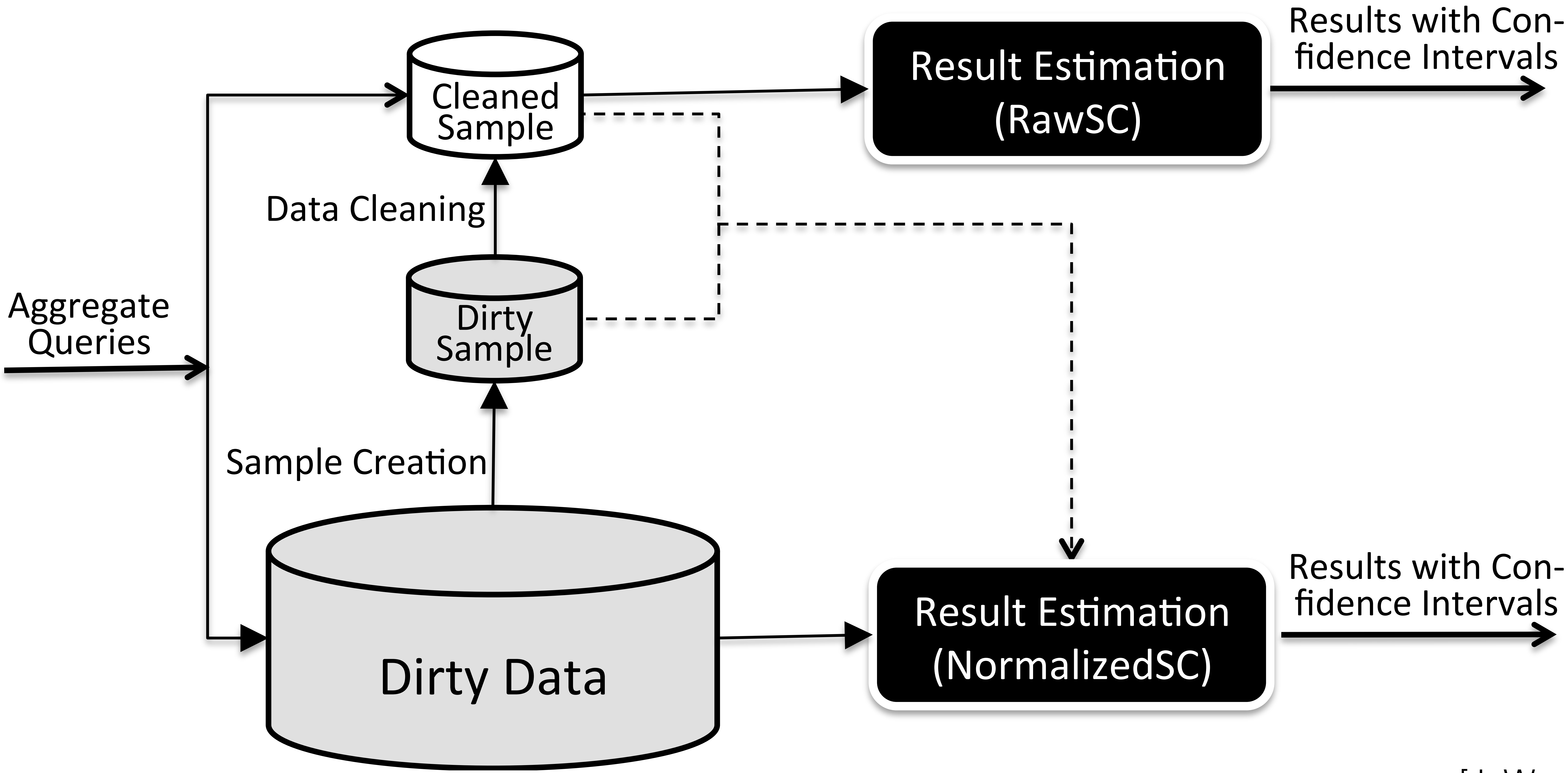
Sector	Ticker	Company	Year	Quarter	Market Cap	Revenue
Aerospace	AJRD	AEROJET ROCKETD	2006	Q1	1442.67	472.07
Aerospace	AJRD	AEROJET ROCKETD	2006	Q2	1514.80	489.22
...	...	...	...	...	...	...
Aerospace	BA	BOEING CO	2006	Q1	343.41	210.66
...	...	...	...	...	...	...
Utilities	YORW	YORK WATER CO	2008	Q4	600.19	271.73

Sector	Ticker	Company	2006	2007	2008
Aerospace	AJRD	AEROJET ROCKETD	6218.09	6342.45	7088.62
	ATRO	ASTRONICS CORP	1050.97	1071.99	1198.11
Business Services	HHS	HARTE-HANKS INC	2473.75	2523.22	2820.07
	NCMI	NATL CINEMEDIA	856.92	874.06	976.89
Consumer Staples	YTEN	TIELD10 BIOSCI	533.13	543.79	607.77
	...	...	...	...	...
Utilities	YORW	YORK WATER CO	1902.37	1940.42	2168.70

Ticker	Company	Year	Aerospace	Business Services	...	Utilities
AJRD	AEROJET ROCKETD	2006	6218.09	NULL	...	NULL
AJRD	AEROJET ROCKETD	2007	6342.45	NULL	...	NULL
AJRD	AEROJET ROCKETD	2008	7088.62	NULL	...	NULL
ATRO	ASTRONICS CORP	2006	1050.97	NULL	...	NULL
...	...	...	...	...	...	...
HHS	HARTE-HANKS INC	2006	NULL	2473.75	...	NULL
...	...	...	...	...	...	...
YORW	YORK WATER CO	2008	NULL	NULL	...	2168.7

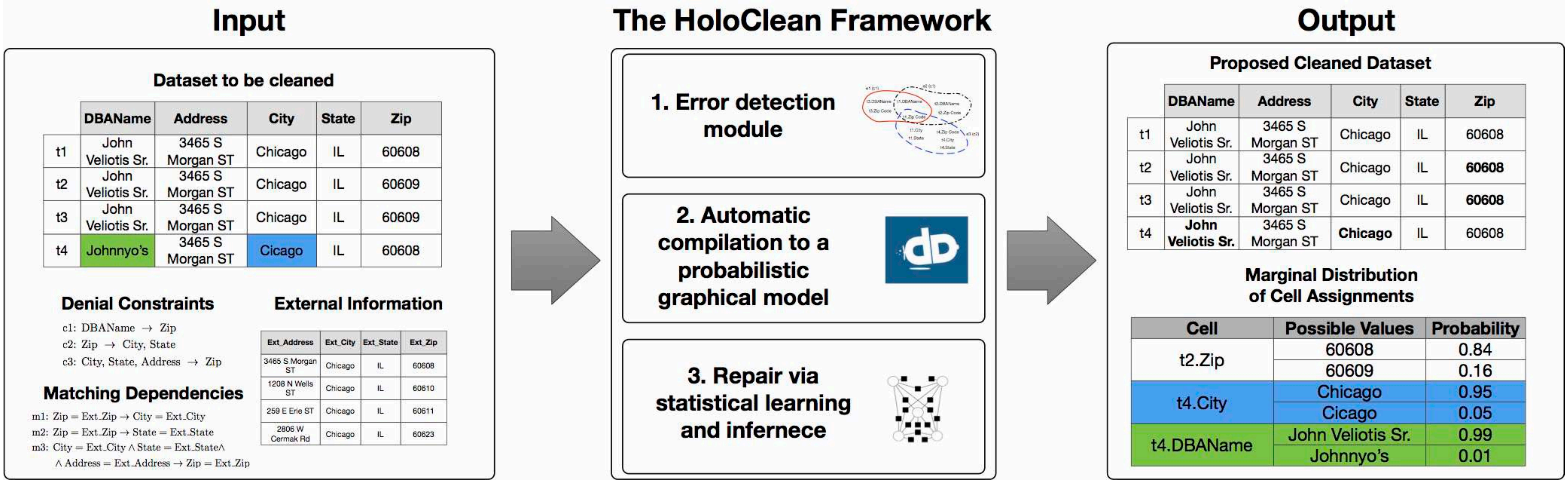
[C. Yan & Y. He]

Data Cleaning: SampleClean



[J. Wang et al., 2014]

Data Cleaning: HoloClean



[T. Rekatsinas et al., 2017]

Merges (aka Joins)

- Need to merge data from one DataFrame with data from another DataFrame
- Example: Football game data merged with temperature data

Game

Id	Location	Date	Home	Away
0	Boston	9/2	1	15
1	Boston	9/9	1	7
2	Cleveland	9/16	12	1
3	San Diego	9/23	21	1

Weather

wId	City	Date	Temp
0	Boston	9/2	72
1	Boston	9/3	68
…	…	…	…
7	Boston	9/9	75
…	…	…	…
21	Boston	9/23	54
…	…	…	…
36	Cleveland	9/16	81

No data for San Diego

Inner Strategy

Merged

Id	Location	Date	Home	Away	Temp	wId
0	Boston	9/2	1	15	72	0
1	Boston	9/9	1	7	75	7
2	Cleveland	9/16	12	1	81	36

No San Diego entry

Outer Strategy

Merged

Id	Location	Date	Home	Away	Temp	wId
0	Boston	9/2	1	15	72	0
NaN	Boston	9/3	NaN	NaN	68	1
…	…	…	…	…	…	…
1	Boston	9/9	1	7	75	7
NaN	Boston	9/10	NaN	NaN	76	8
…	…	…	…	…	…	…
NaN	Cleveland	9/2	NaN	NaN	61	22
…	…	…	…	…	…	…
2	Cleveland	9/16	12	1	81	36
…	…	…	…	…	…	…
3	San Diego	9/23	21	1	NaN	NaN

Data Integration

Movie: Title, director, year, genre
Actors: title, actor
Plays: movie, location, startTime
Reviews: title, rating, description

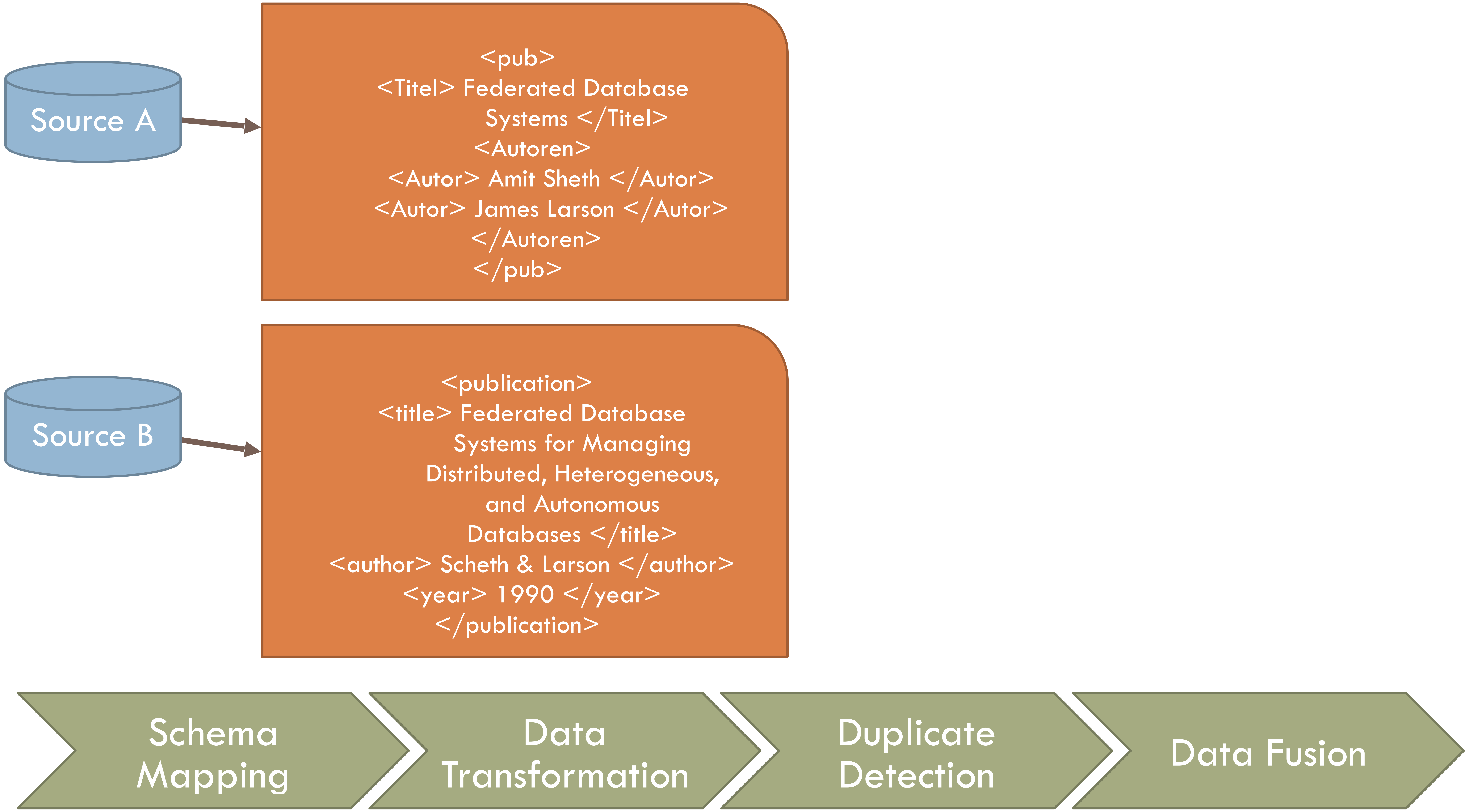
```
select title, startTime
  from Movie, Plays
where Movie.title=Plays.movie AND
    location="New York" AND
     director="Woody Allen"
```

Sources S1 and S3 are relevant, sources S4 and S5 are irrelevant, and source S2 is relevant but possibly redundant.

S1	S2	S3	S4	S5
Movies: name, actors, director, genre	Cinemas: place, movie, start	Cinemas in NYC: cinema, title, startTime	Cinemas in SF: location, movie, startingTime	Reviews: title, date grade, review

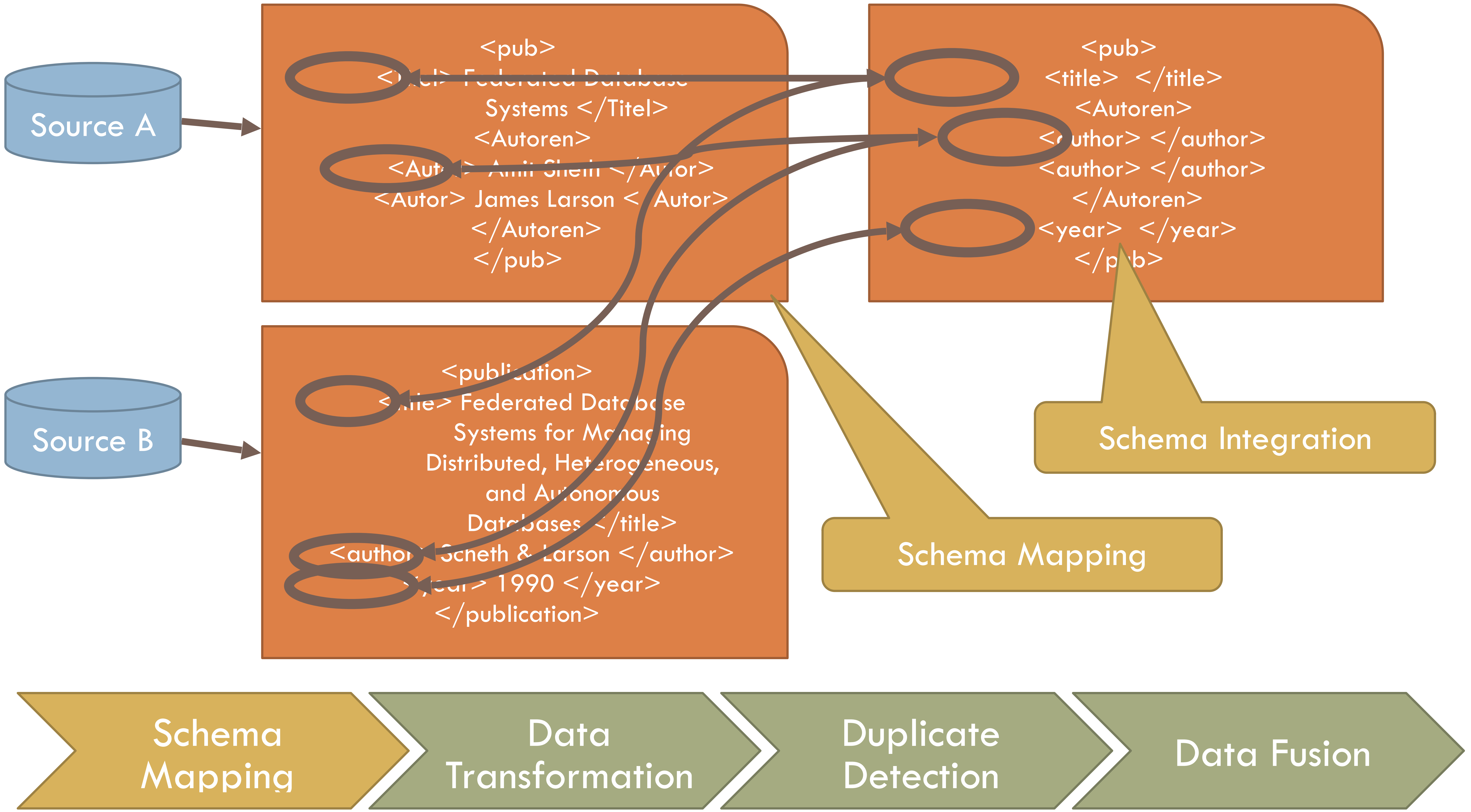
[AH Doan et al., 2012]

Information Integration



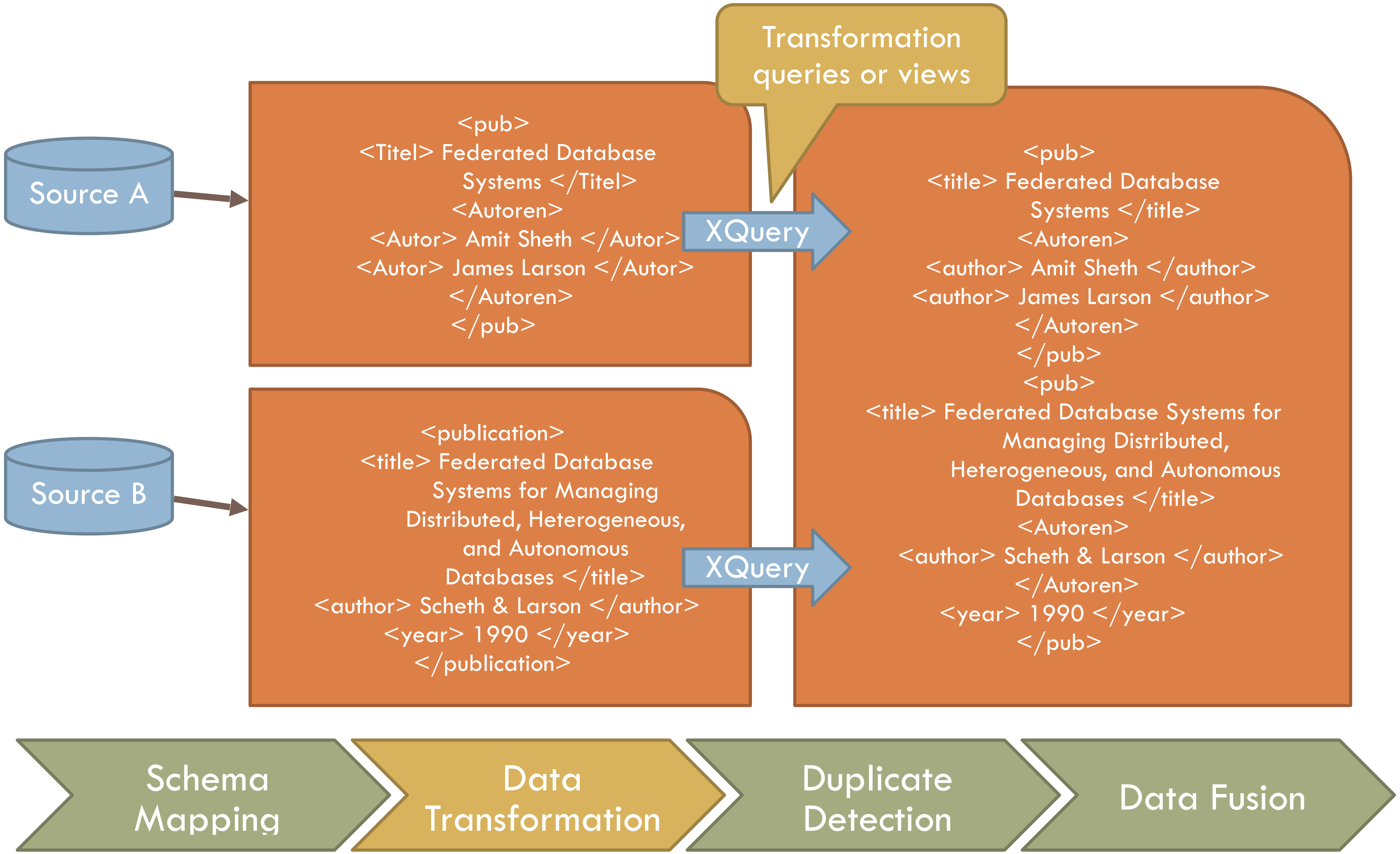
[L. Dong and F. Naumann, 2009]

Information Integration



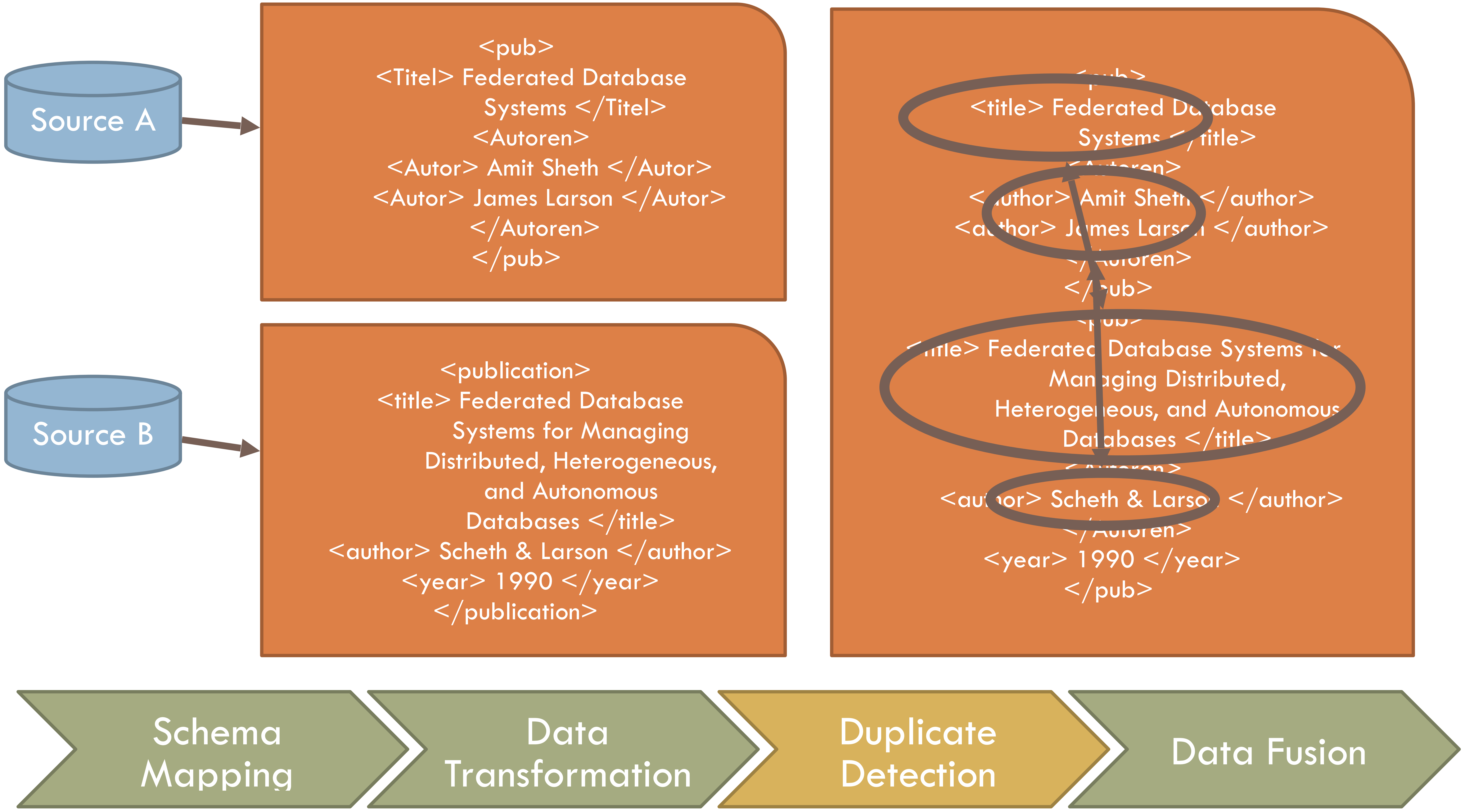
[L. Dong and F. Naumann, 2009]

Information Integration



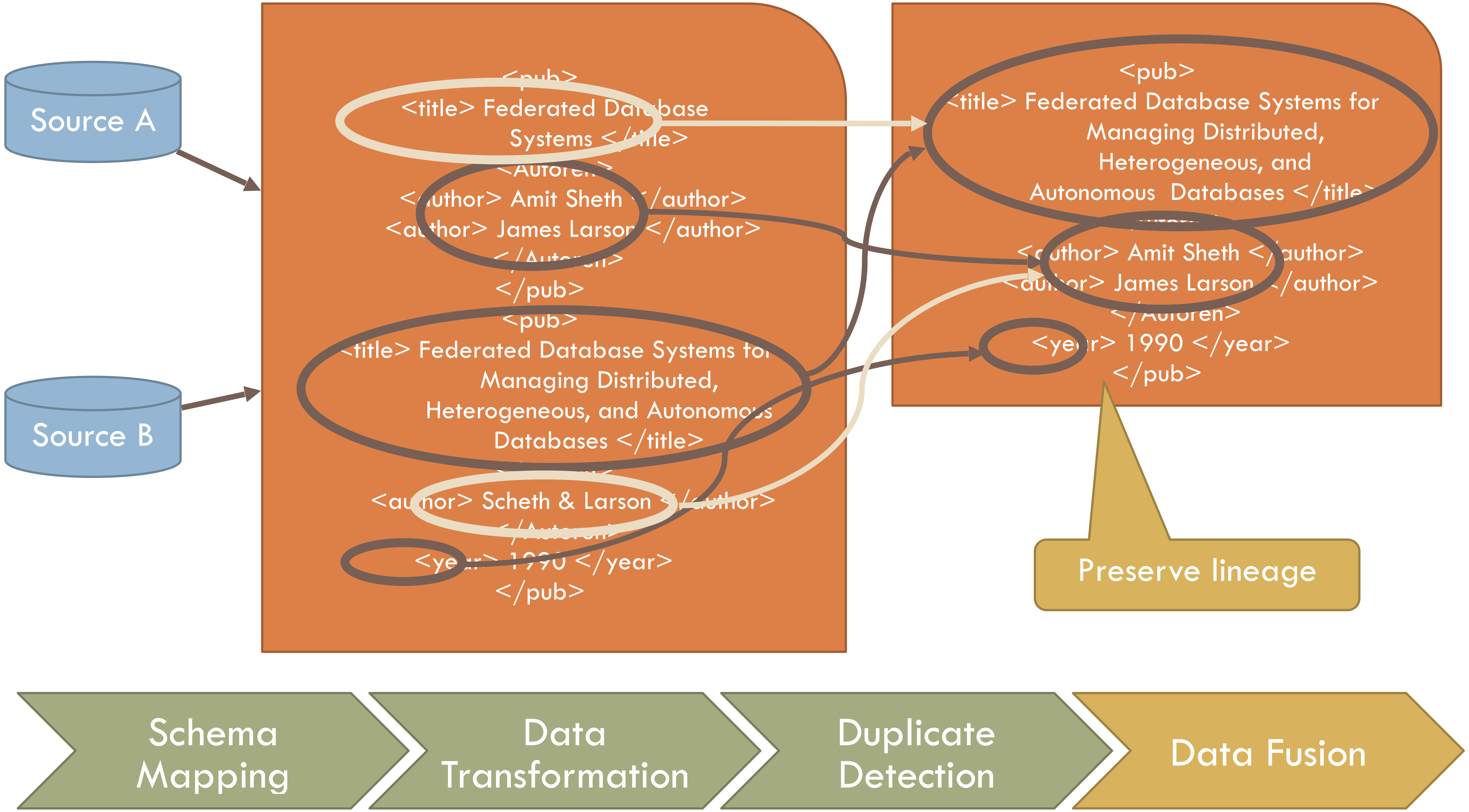
[L. Dong and F. Naumann, 2009]

Information Integration



[L. Dong and F. Naumann, 2009]

Information Integration



[L. Dong and F. Naumann, 2009]

Challenges in Data Fusion when Sources Copy

	S1	S2	S3	S4	S5
Stonebraker	MIT	Berkeley	MIT	MIT	MS
Dewitt	MSR	MSR	UWisc	UWisc	UWisc
Bernstein	MSR	MSR	MSR	MSR	MSR
Carey	UCI	AT&T	BEA	BEA	BEA
Halevy	Google	Google	UW	UW	UW

[X L Dong et al., 2009]

Challenges in Data Fusion when Sources Copy

	S1	S2	S3	S4	S5
Stonebraker	MIT	Berkeley	MIT	MIT	MS
Dewitt	MSR	MSR	UWisc	UWisc	UWisc
Bernstein	MSR	MSR	MSR	MSR	MSR
Carey	UCI	AT&T	BEA	BEA	BEA
Halevy	Google	Google	UW	UW	UW

[X L Dong et al., 2009]

Challenges in Data Fusion when Sources Copy

2. With only a snapshot it is hard to decide which source is a copier.

	S1	S2	S3	S4	S5
Stonebraker	MIT	Berkeley	MIT	MIT	MS
Dewitt	MSR	MSR	UWisc	UWisc	UWisc
Bernstein	MSR	MSR	MSR	MSR	MSR
Carey	UCI	AT&T	BEA	BEA	BEA
Halevy	Google	Google	UW	UW	UW

[X L Dong et al., 2009]

Challenges in Data Fusion when Sources Copy

1. Sharing common data does not in itself imply copying.

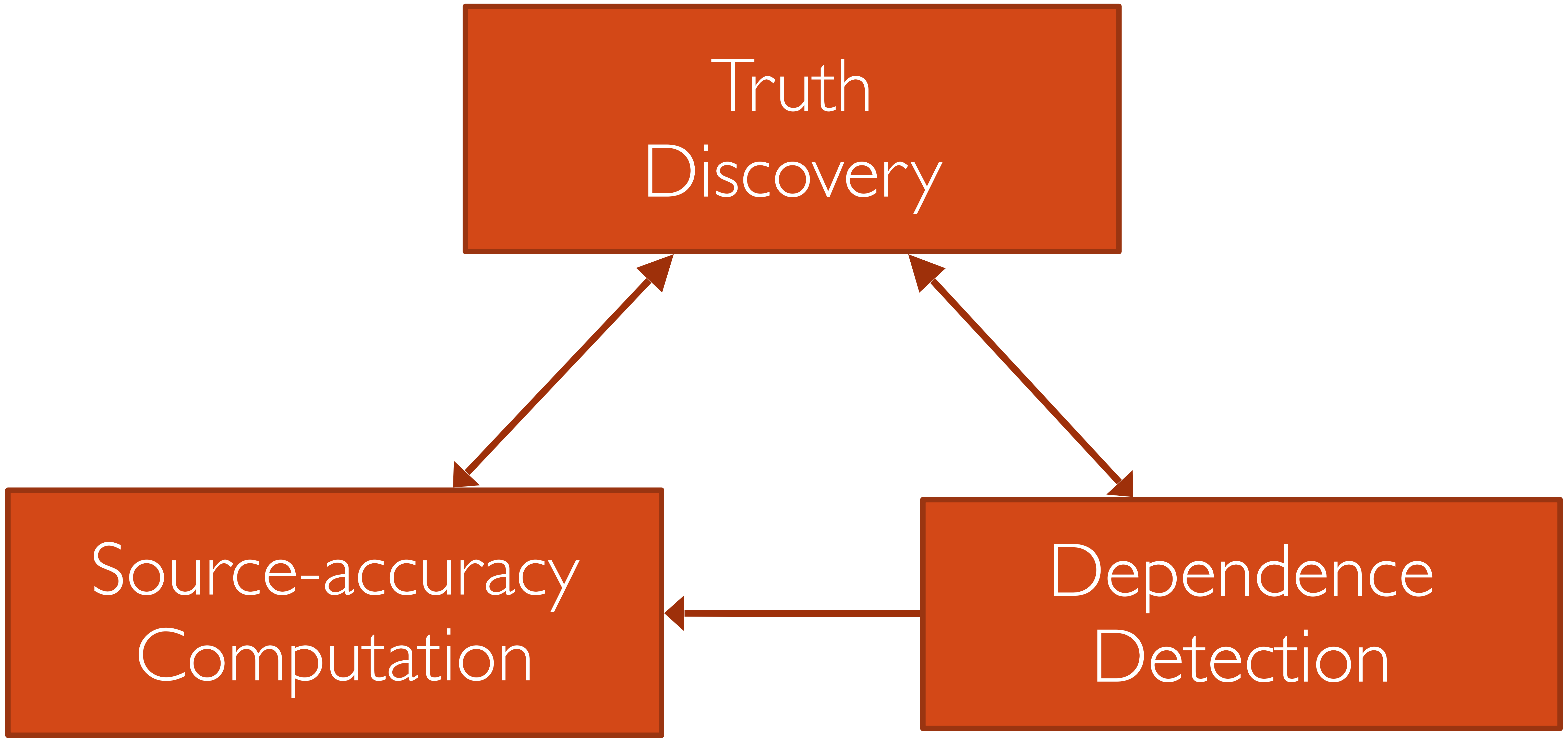
2. With only a snapshot it is hard to decide which source is a copier.

	S1	S2	S3	S4	S5
Stonebraker	MIT	Berkeley	MIT	MIT	MS
Dewitt	MSR	MSR	UWisc	UWisc	UWisc
Bernstein	MSR	MSR	MSR	MSR	MSR
Carey	UCI	AT&T	BEA	BEA	BEA
Halevy	Google	Google	UW	UW	UW

3. A copier can also provide or verify some data by itself, so it is inappropriate to ignore all of its data.

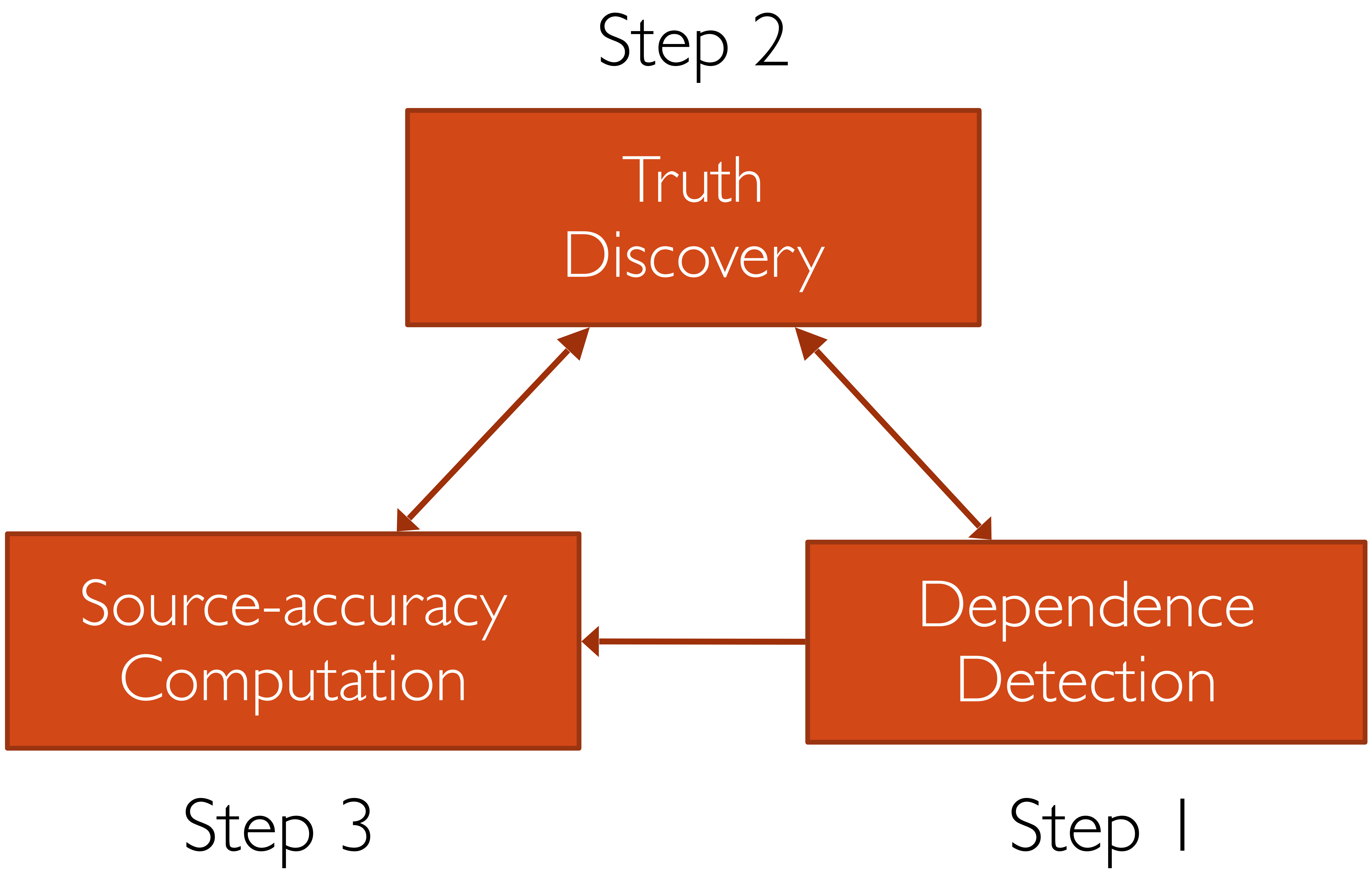
[X L Dong et al., 2009]

Source Dependence: Iteration on Truth and Sources



[X L Dong et al., 2009]

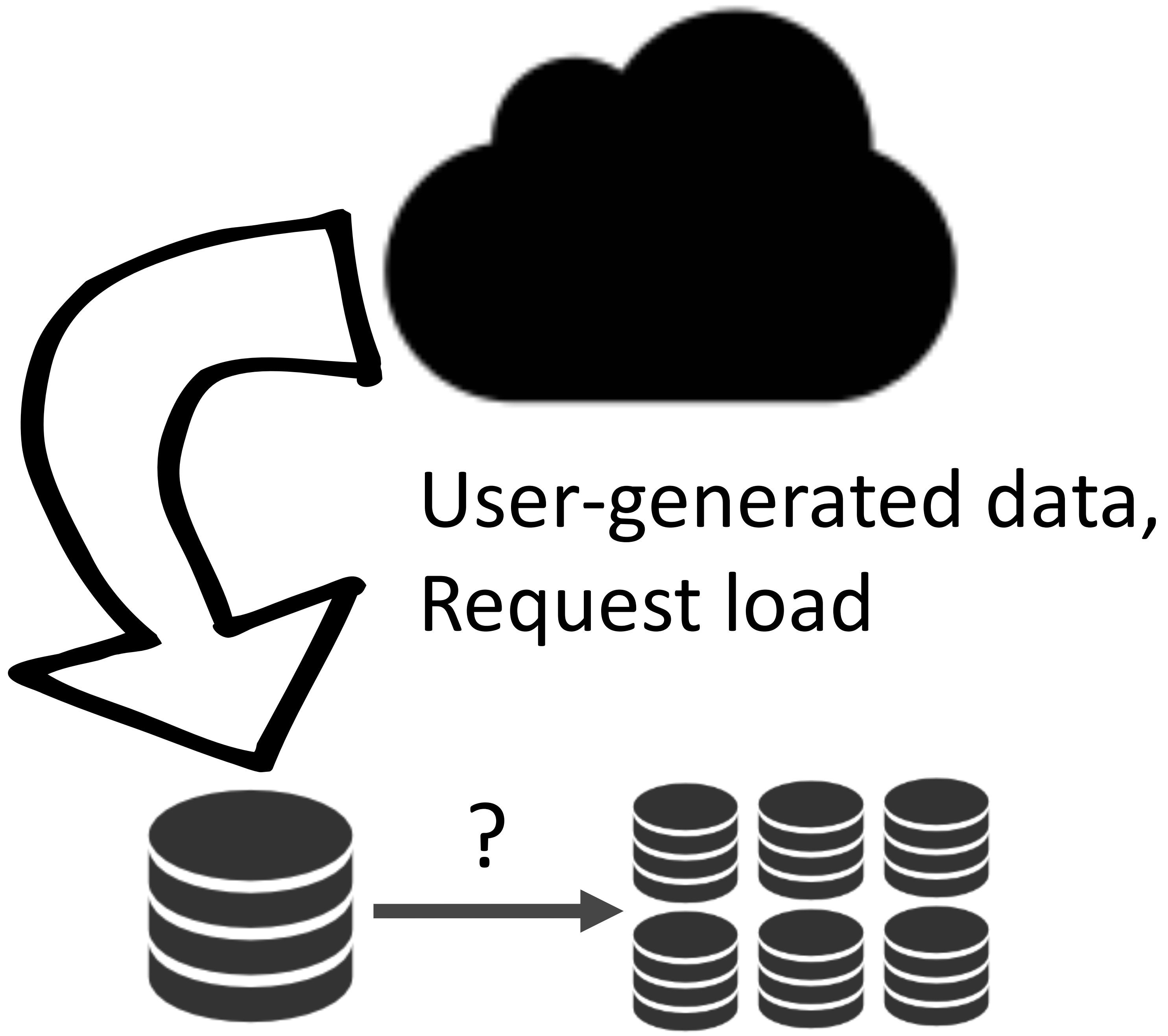
Source Dependence: Iteration on Truth and Sources

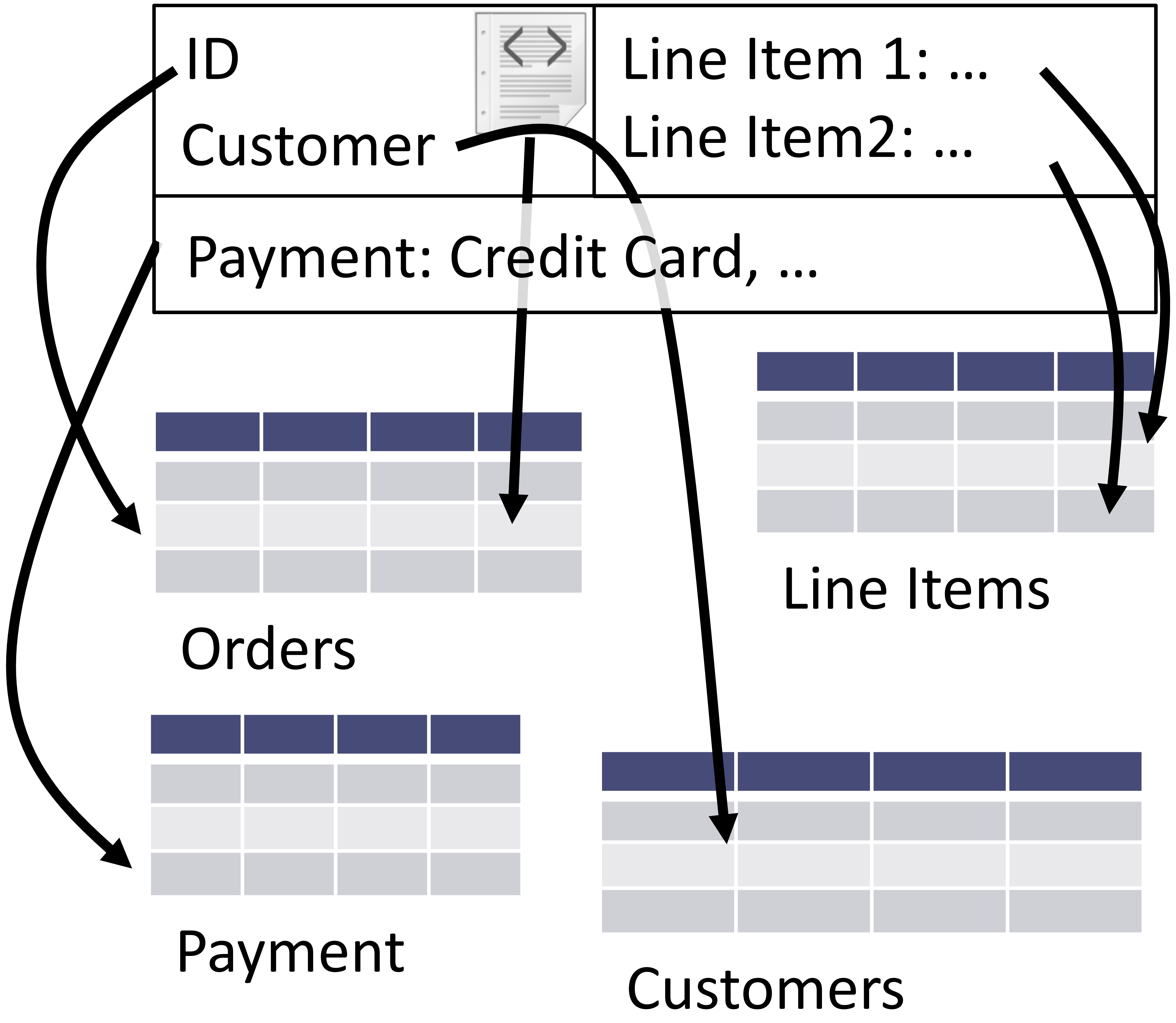


[X L Dong et al., 2009]

NoSQL Motivation

Scalability	Impedance Mismatch





[F. Gessert et al., 2017]

Column Stores

row id = 1

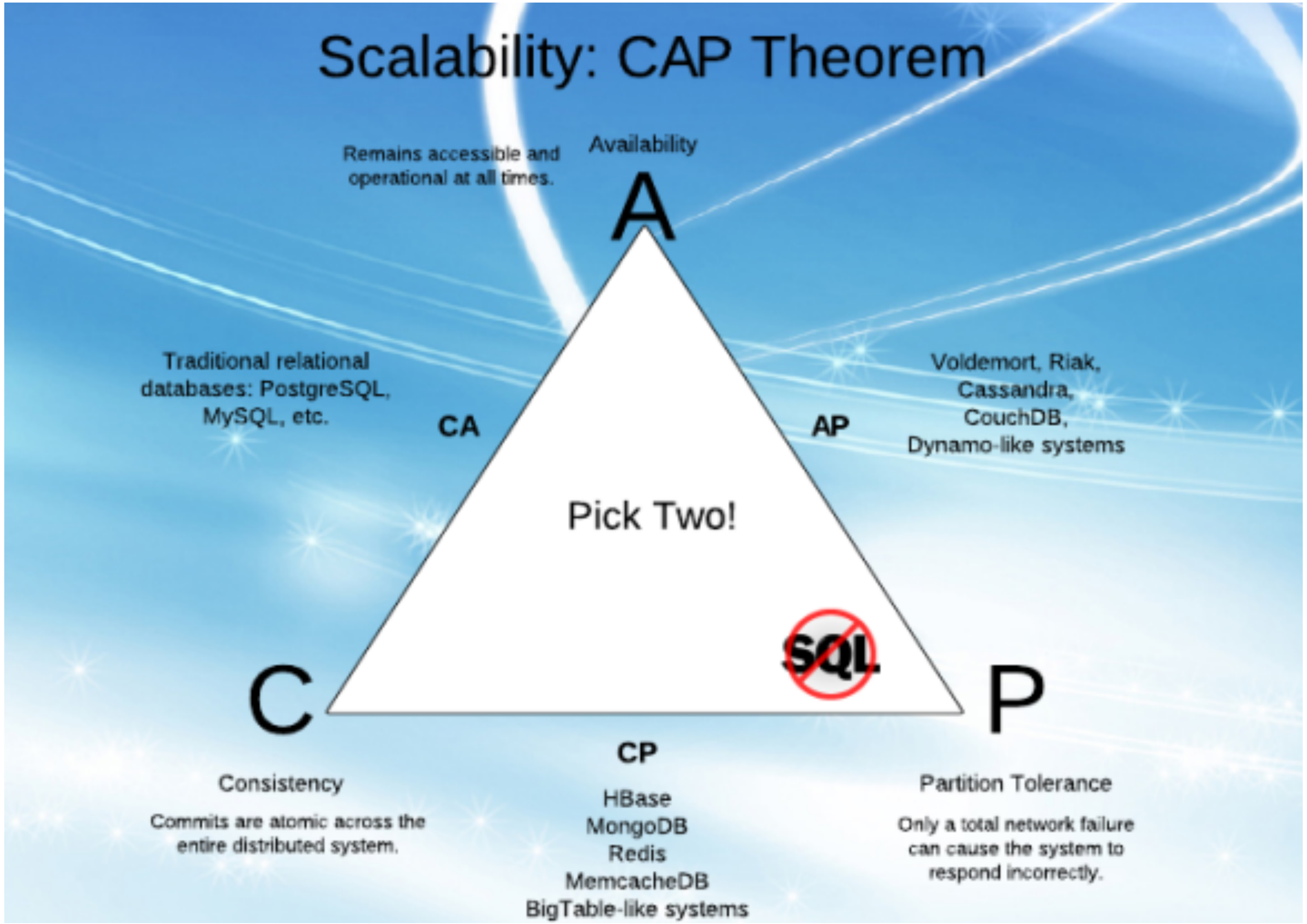
row id = 6

id	Title	Person	Genre
1	Mrs. Doubtfire	Robin Williams	Comedy
2	Jaws	Roy Scheider	Horror
3	The Fly	Jeff Goldblum	Horror
4	Steel Magnolias	Dolly Parton	Drama
5	The Birdcage	Nathan Lane	Comedy
6	Erin Brokovitch	Julia Roberts	Drama

Each column has a file or segment on disk

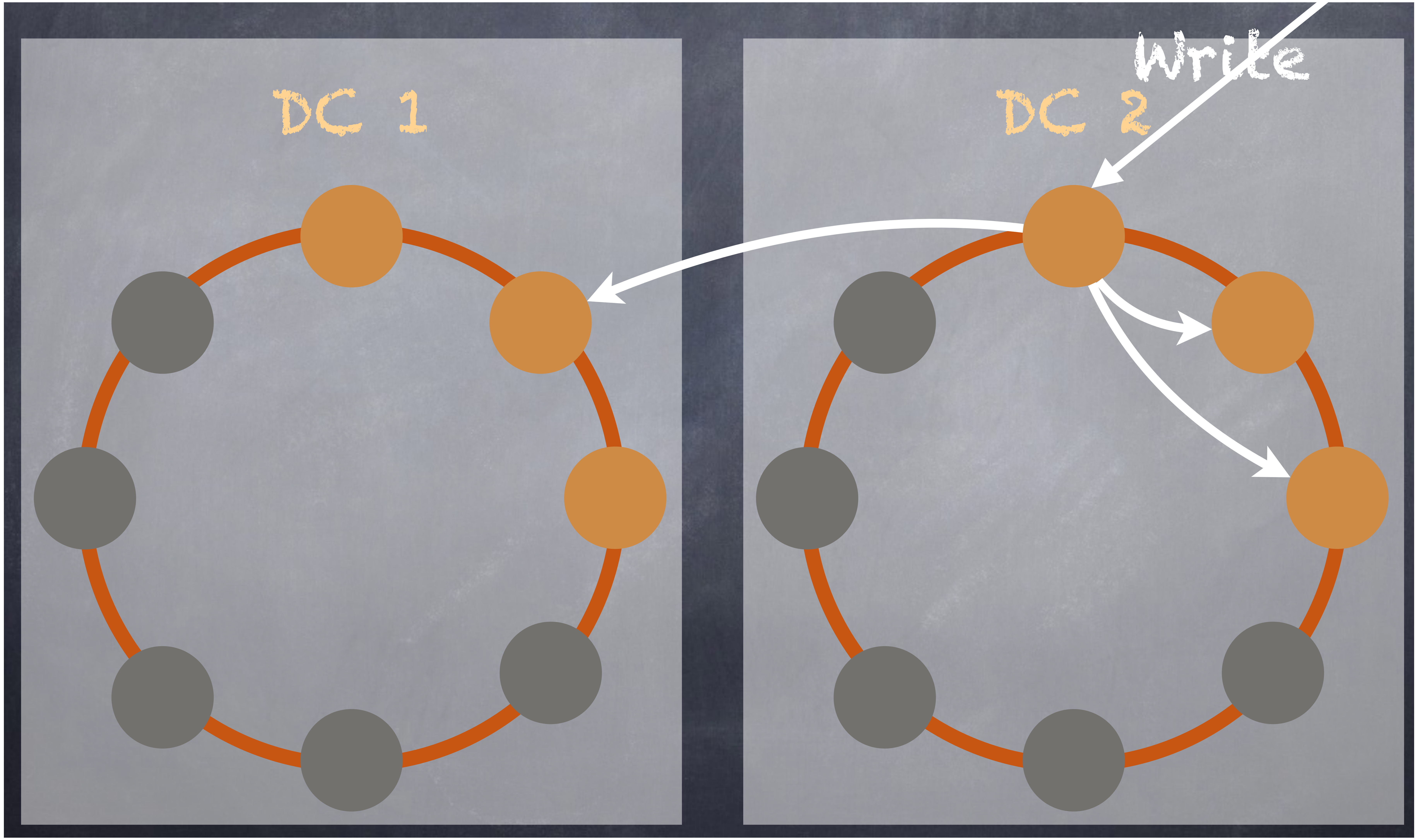
[J. Swanhart, Introduction to Column Stores]

CAP Theorem



[E. Brewer]

Cassandra: Replication and Consistency



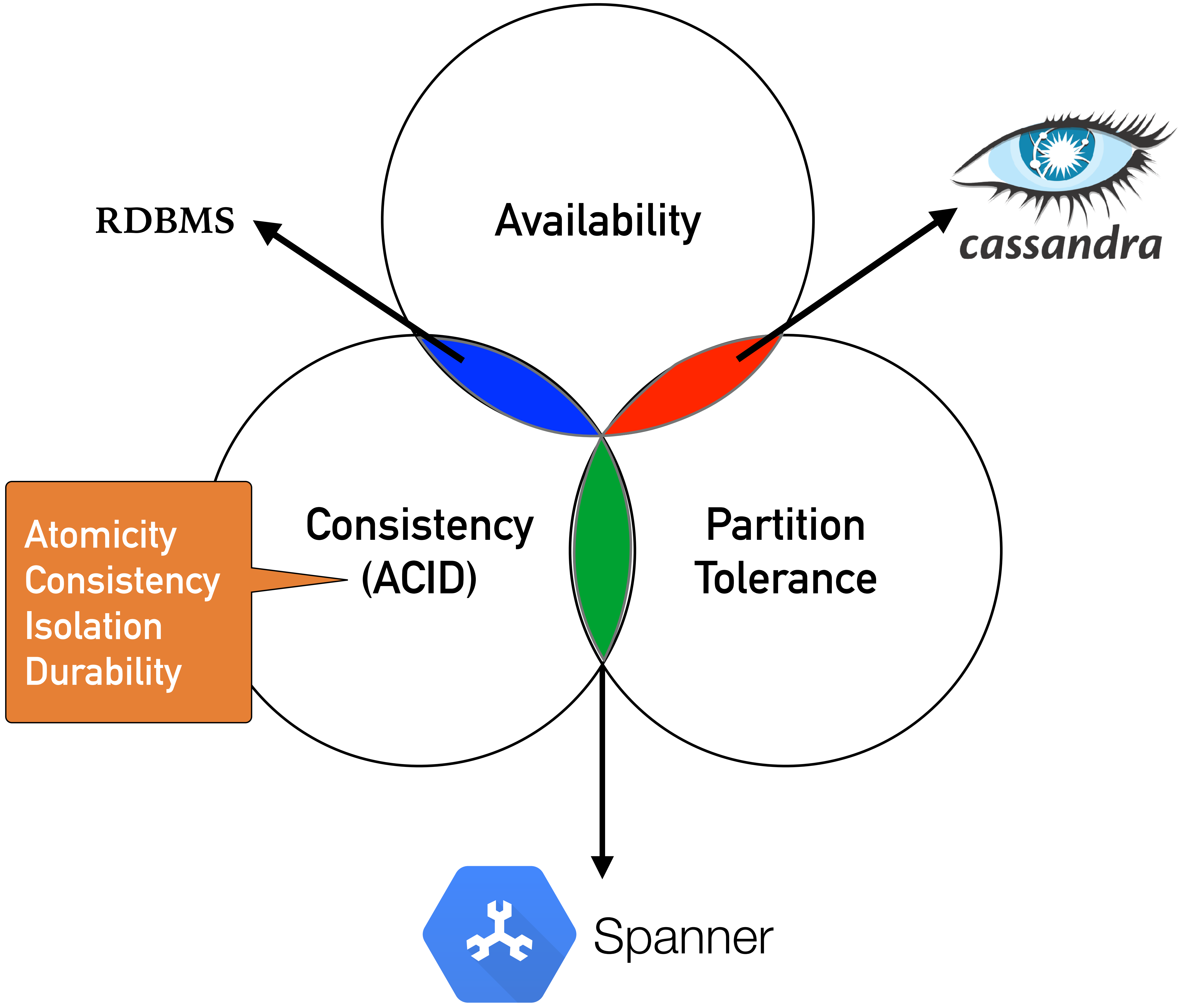
[R. Stupp]

Three Types of NewSQL Systems

- New Architectures
  - New codebase without architectural baggage of legacy systems
  - Examples: VoltDB, Spanner, Clustrix
- Transparent Sharding Middleware:
  - Transparent data sharding & query redirecting over cluster of single-node DBMSs
  - Examples: citusdata, ScaleArc (usually support MySQL/postgres wire)
- Database-as-a-Service:
  - Distributed architecture designed specifically for cloud-native deployment
  - Examples: xeround, GenieDB, FathomDB (usually based on MySQL)

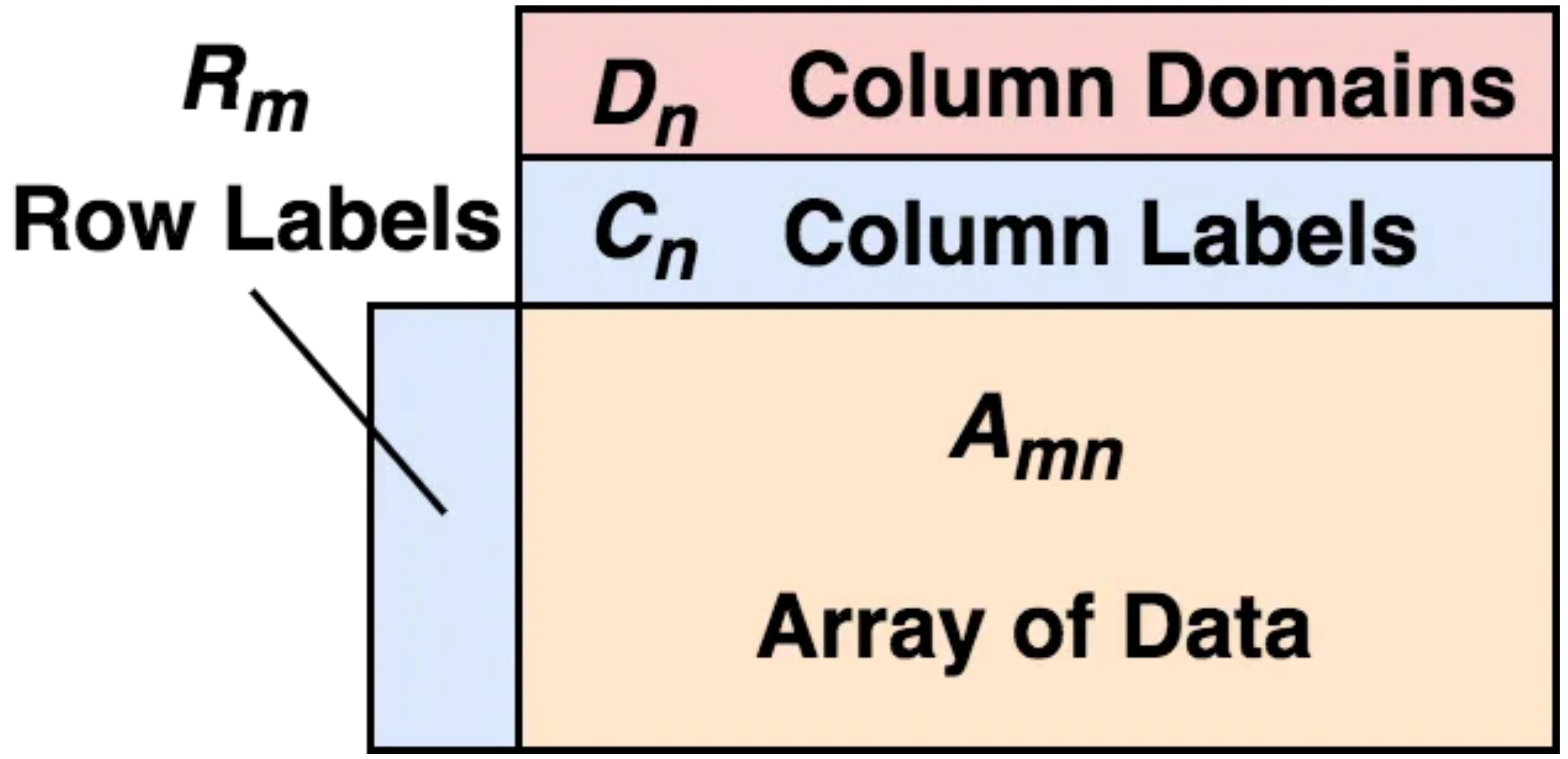
[A. Pavlo]

Spanner: Google's NewSQL Cloud Database



- Which type of system is Spanner?
  - C: consistency, which implies a single value for shared data
  - A: 100% availability, for both reads and updates
  - P: tolerance to network partitions
- Which two?
  - CA: close, but not totally available
  - So actually **CP**

Dataframe Data Model



- Combines parts of matrices, databases, and spreadsheets
- Ordered, but not necessarily sorted
  - Rows and columns
- No predefined schema necessary
  - Types can be induced at runtime
- Typed Row/column labels
  - Labels can become data
- Indexing by label or row/column number
  - "Named notation" or "Positional notation"

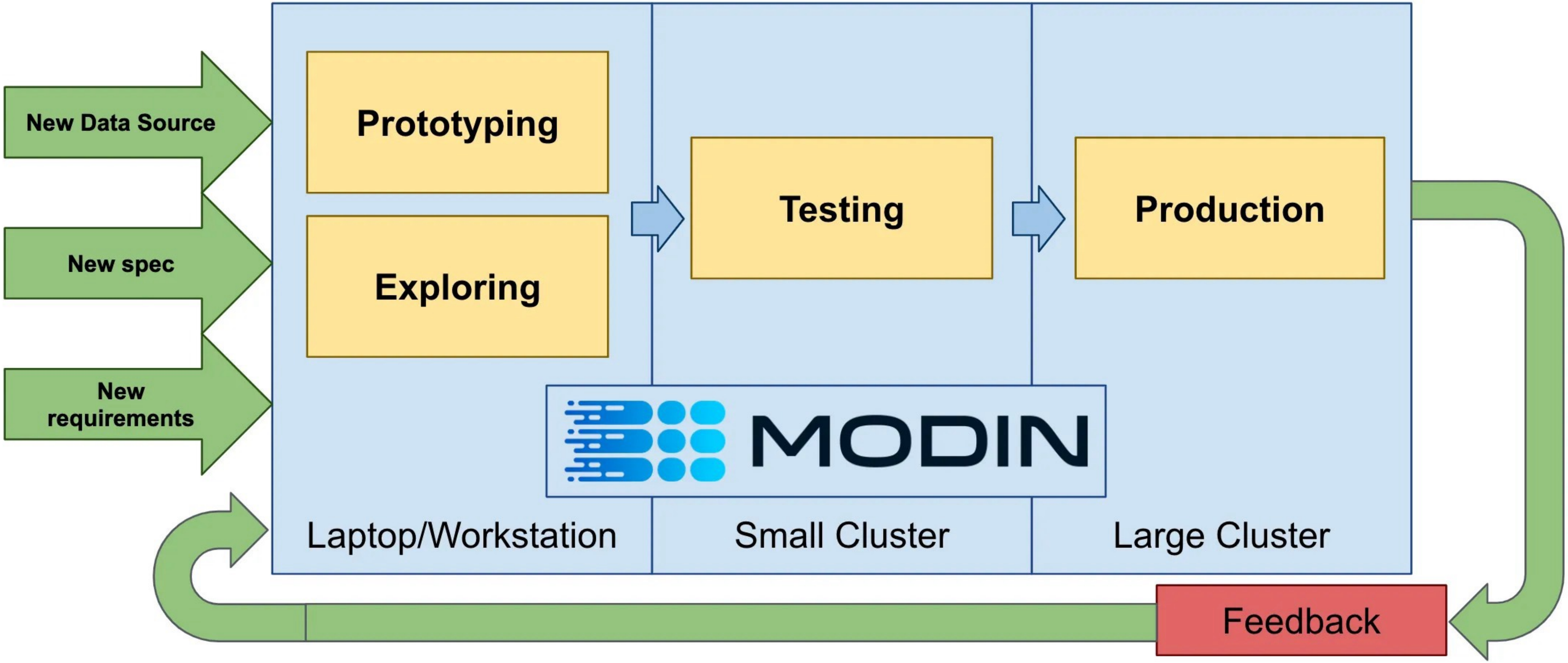
[D. Petersohn]

Differences between Databases & Dataframes

Convenience	Entire query at once	Incremental + inspection
Flexible	Strict schema	Mixed types, R/C and data/metadata equiv.
Versatility	SFW or bust	600+ functions

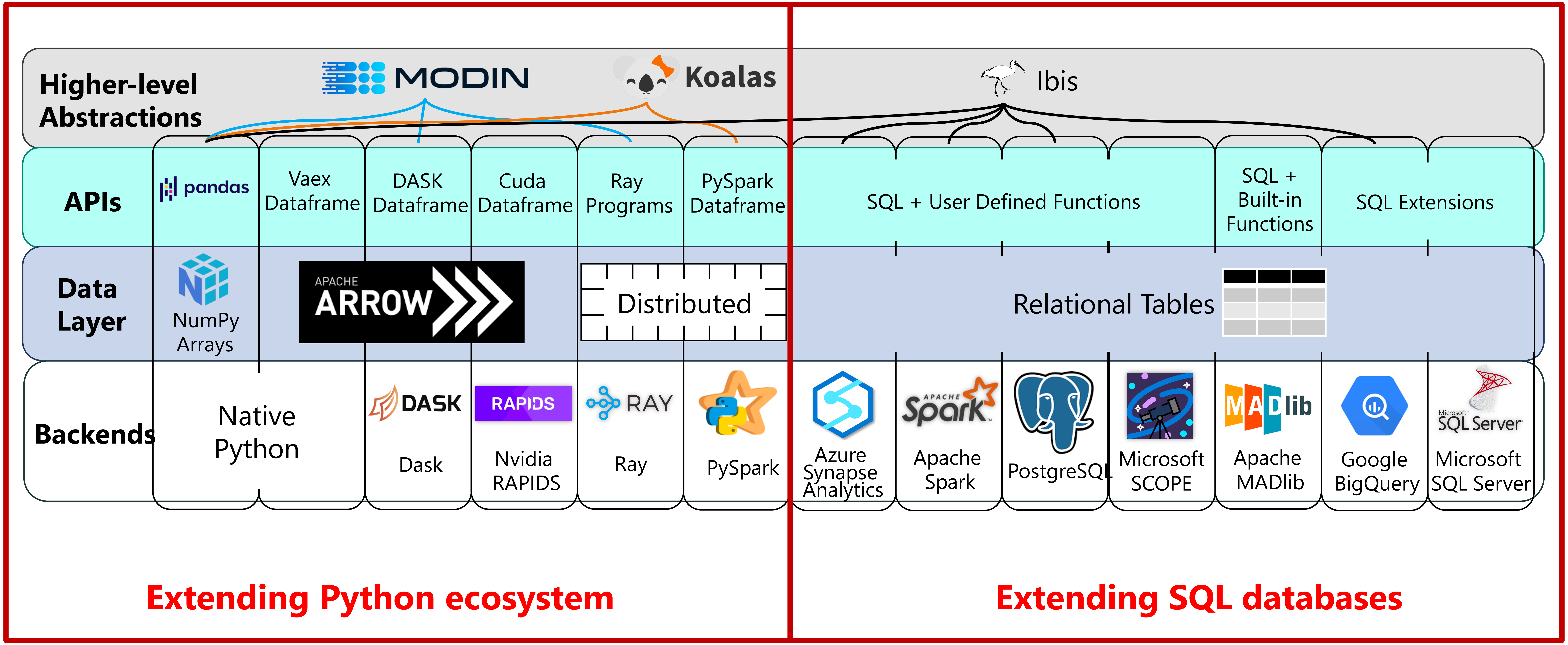
[D. Petersohn, 2022]

Modin as a Way to Scale Dataframes



[D. Petersohn]

Data Science Jungle

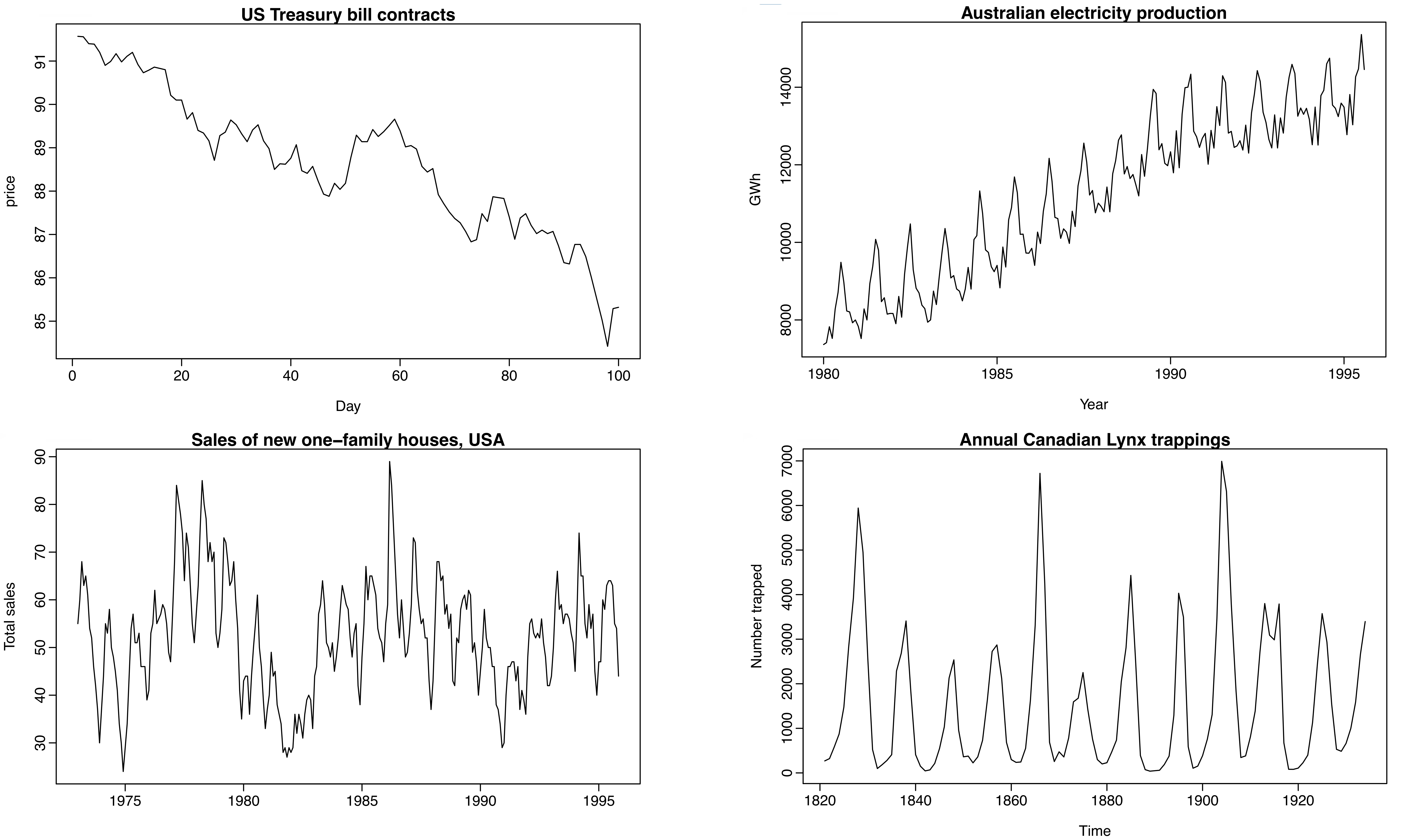


[A. Jindal et al., 2021]

Magpie Goals

Layer		Goal
Pythonic Environment	python	Familiar Python surface
Unified Dataframe API	pandas	Ongoing standardization
Magpie Middleware	PyFroid Compiler	Batching Pandas into large query expressions
	Cross Optimization	Backend selection using past workloads
	Common Data Layer — APACHE ARROW	Cache commonly seen dataframes
Polyengines & Mappers: Azure Synapse Analytics, Koalas, SQL Alchemy	Native Python (python)	Multi-backend environments and libraries
Database Backends: Microsoft SCOPE, Apache Spark, PostgreSQL, Apache MADlib, Google BigQuery, SQL Server	pandas, intel Scalable Dataframe Compiler	Cloud backends

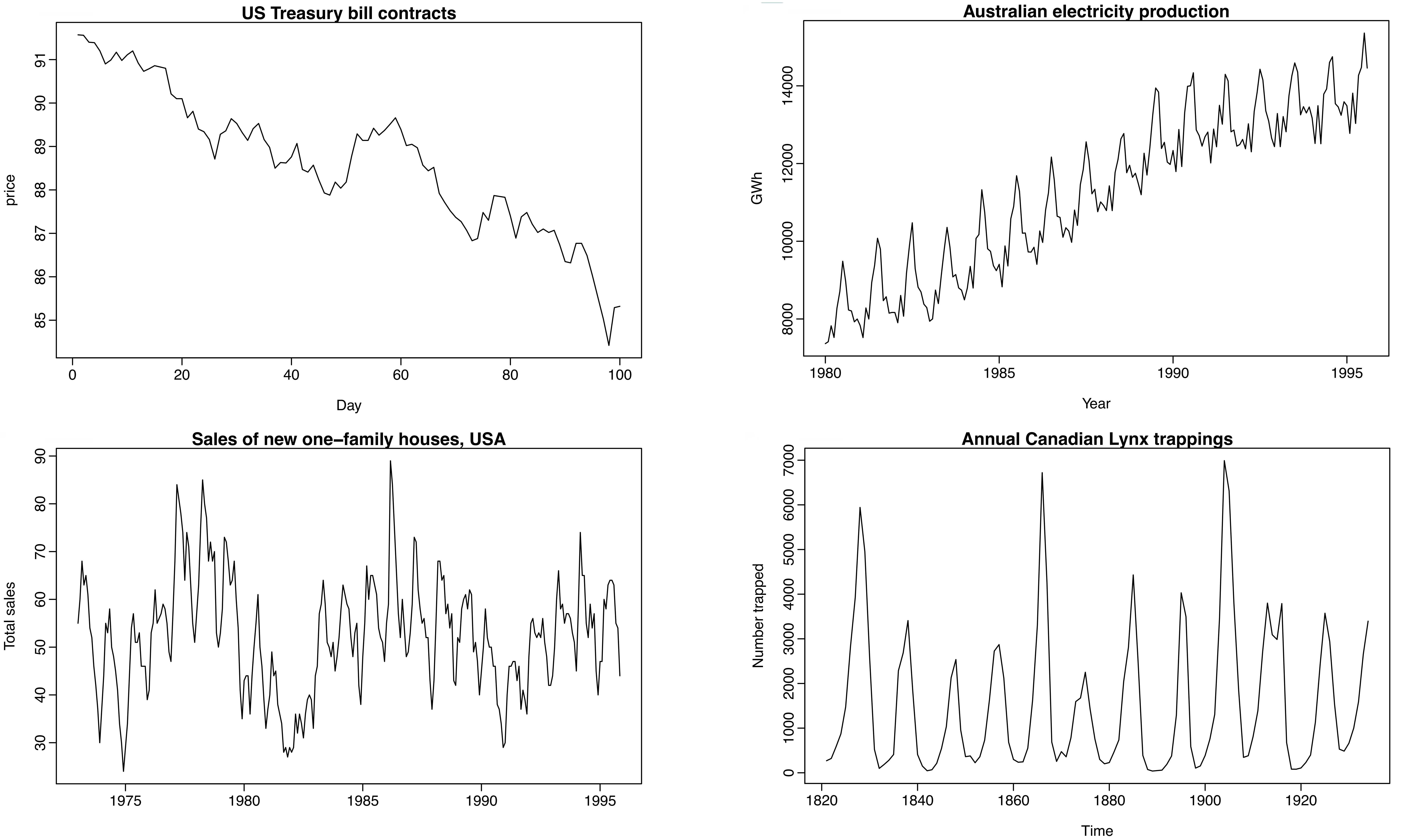
Time Series Data



[R. J. Hyndman]

Time Series Data

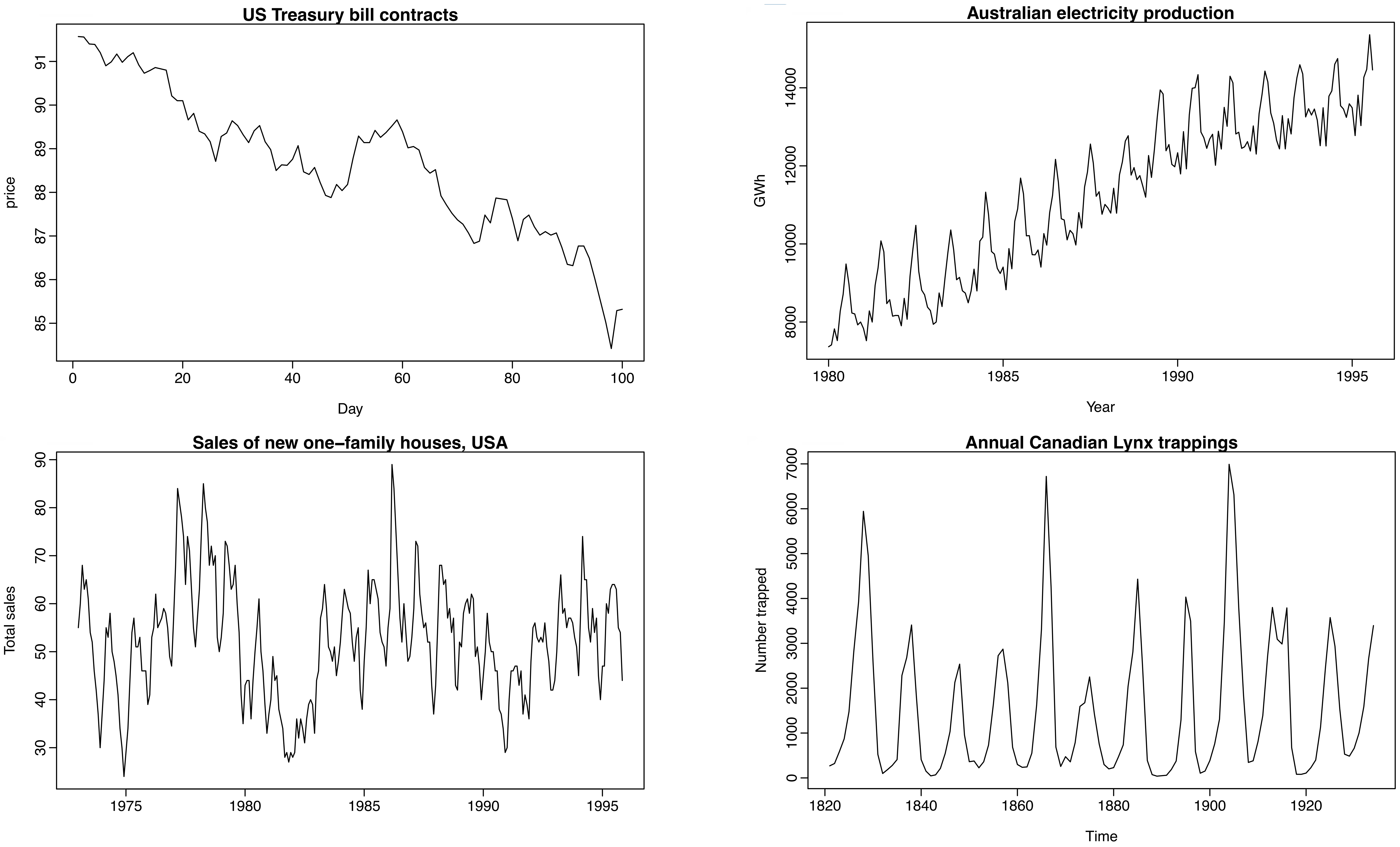
Trend



[R. J. Hyndman]

Time Series Data

Trend

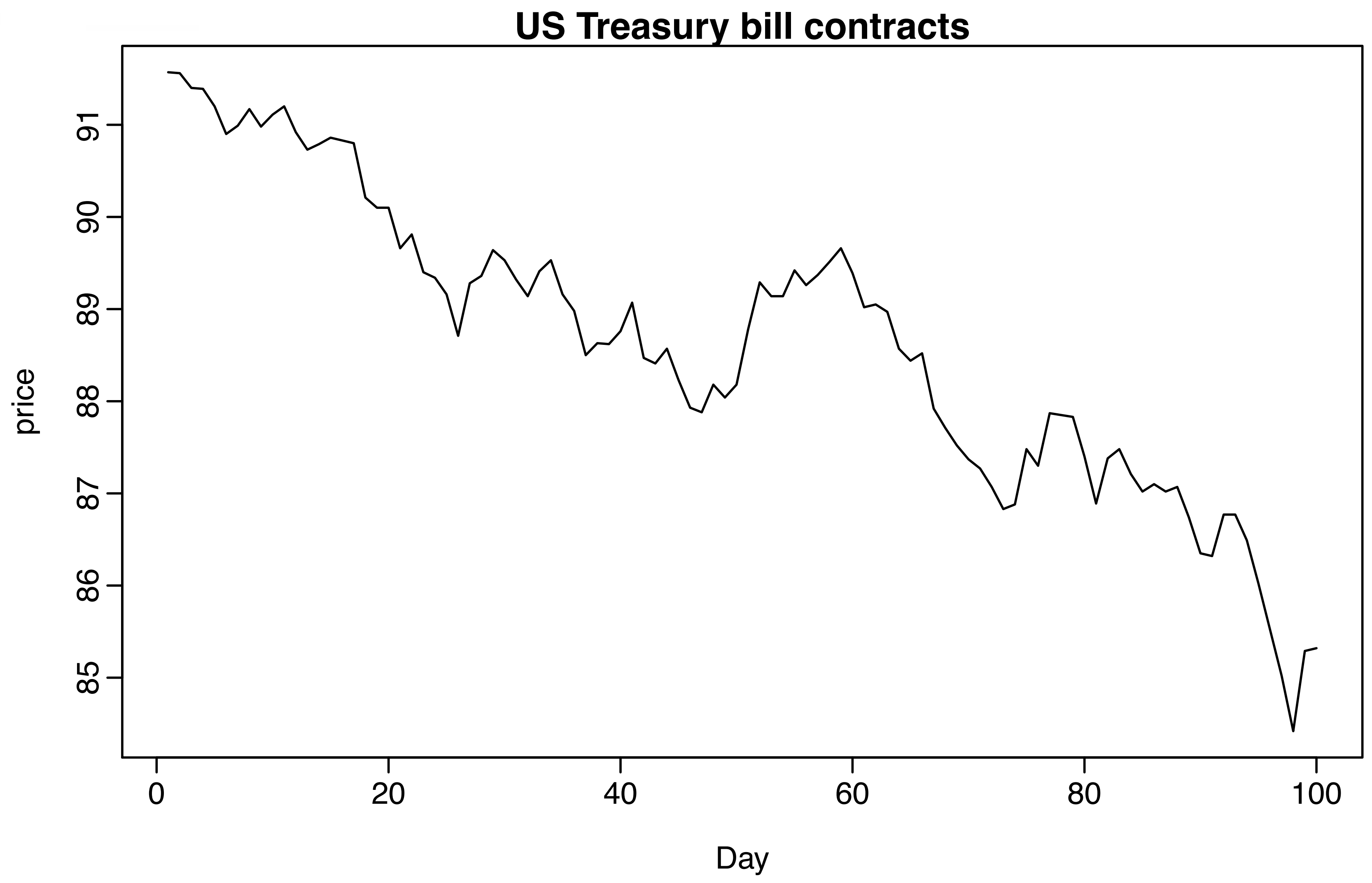


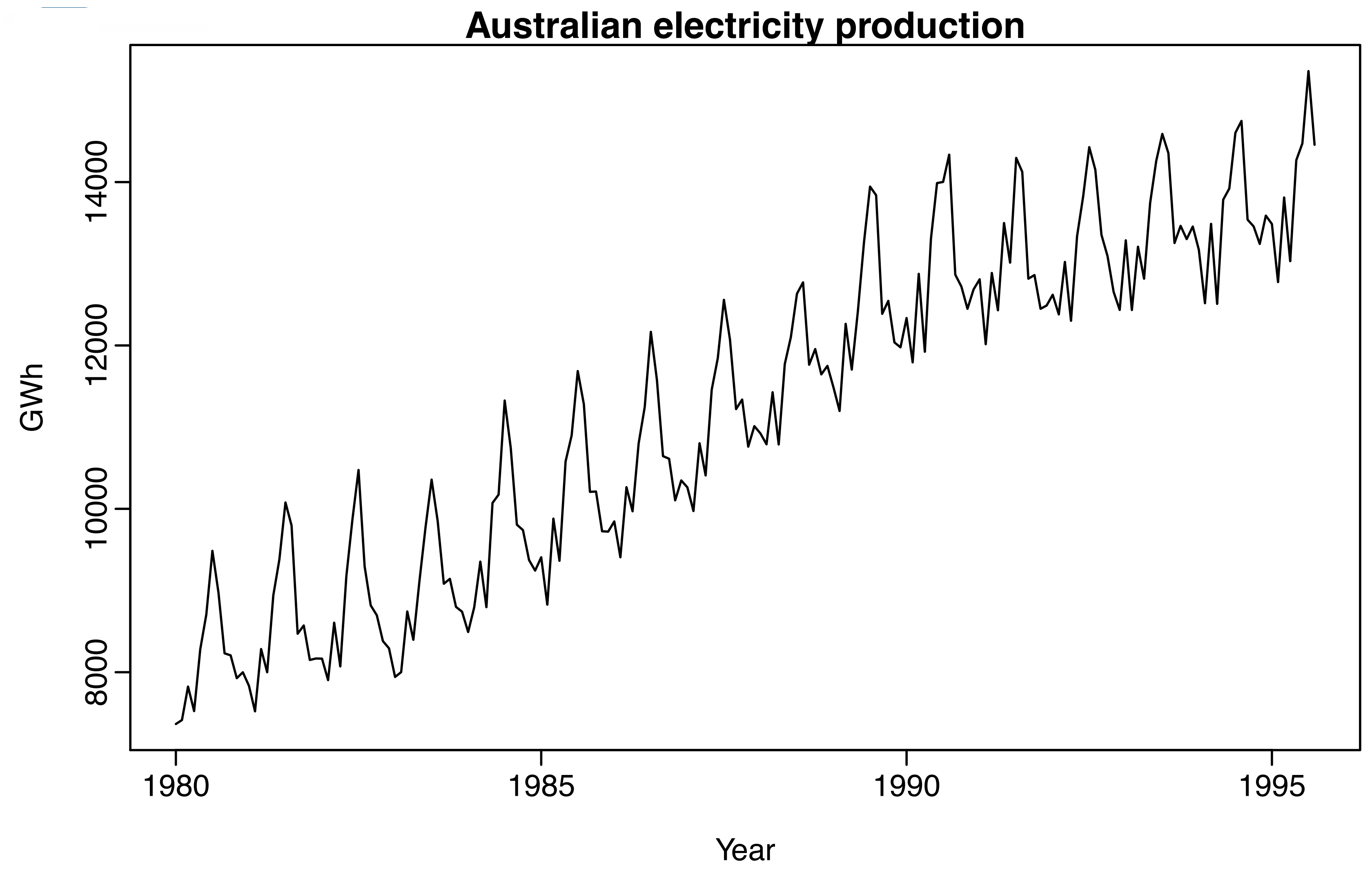
Trend + Seasonality

[R. J. Hyndman]

Time Series Data

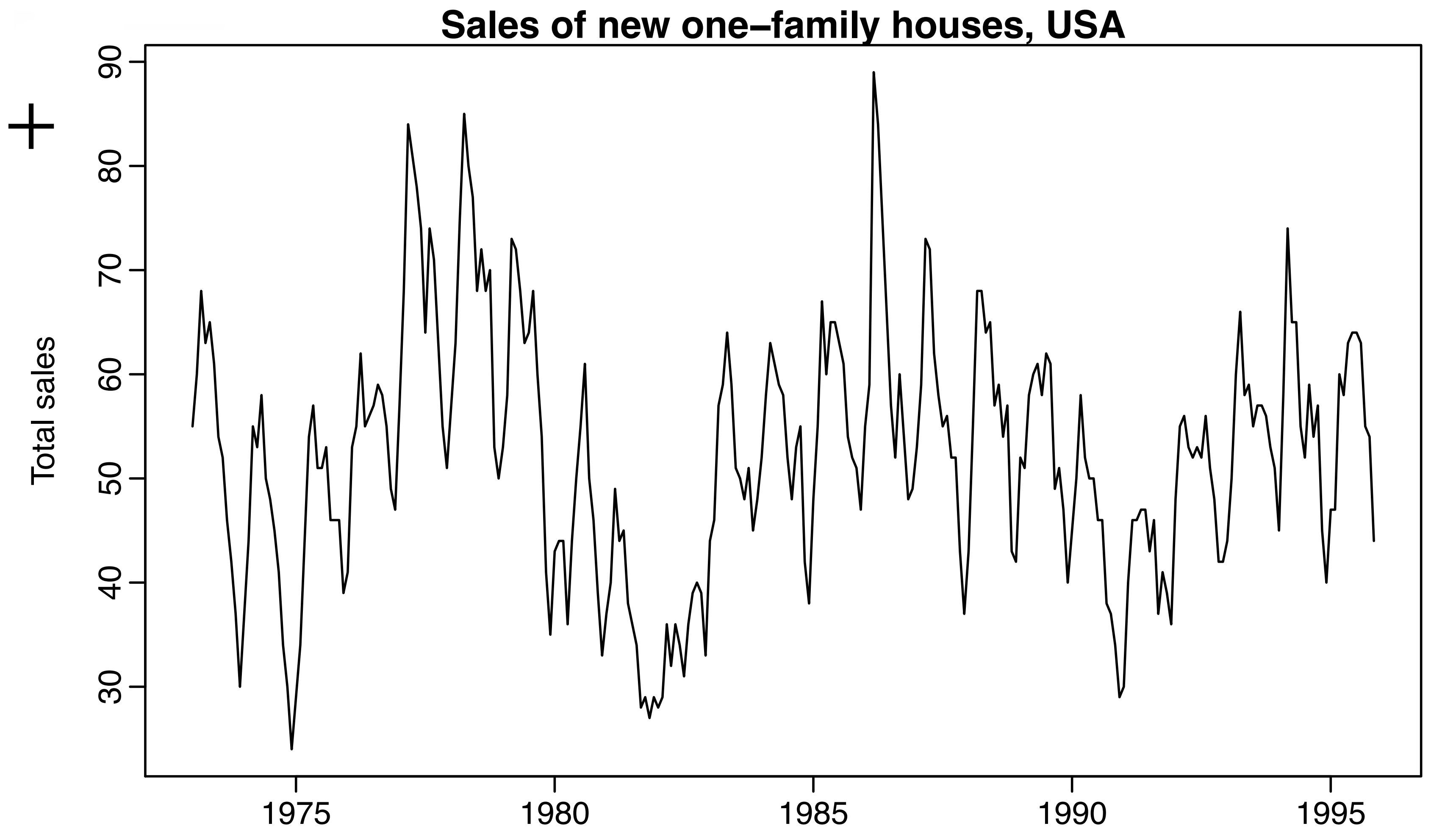
Trend

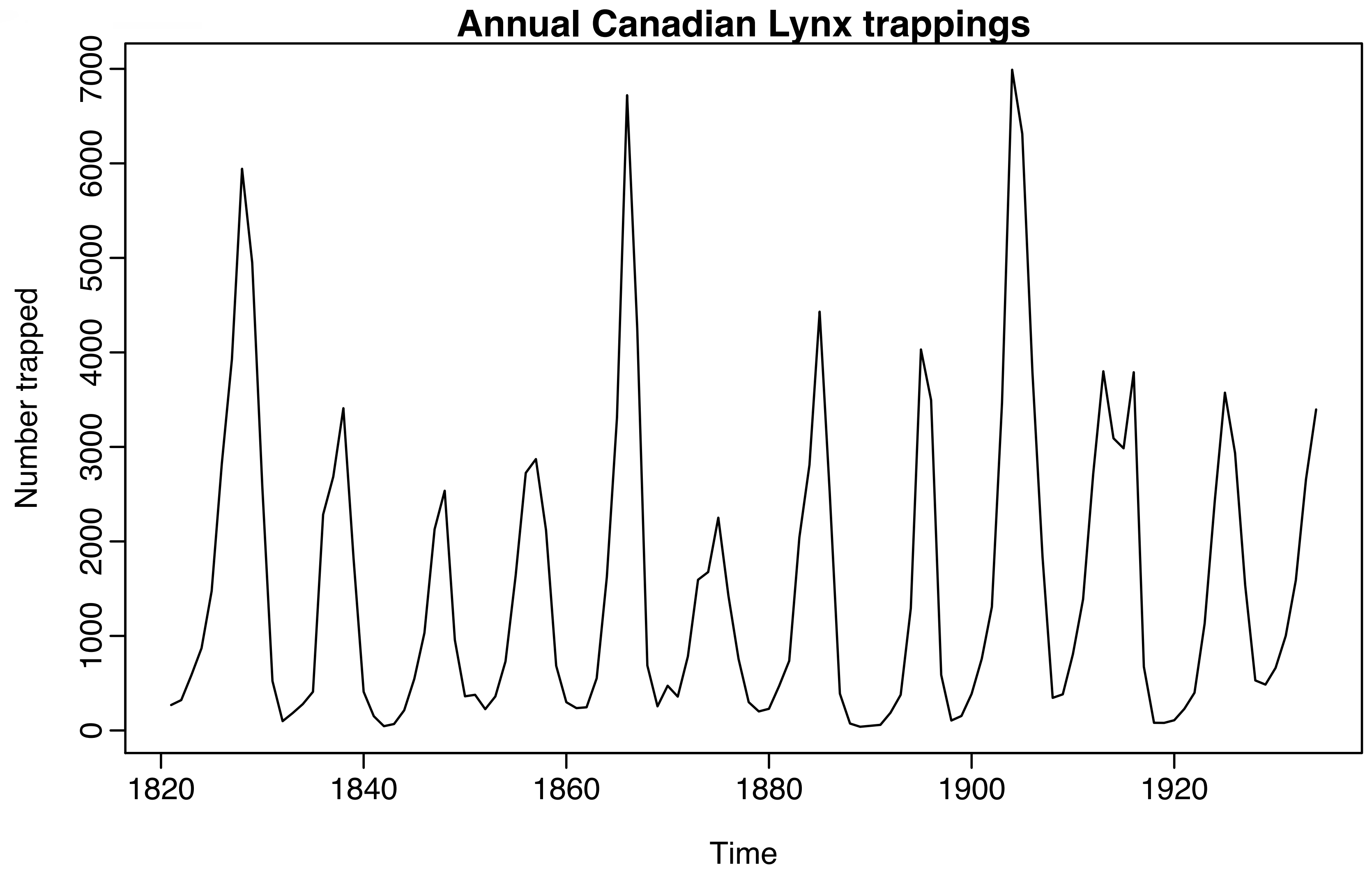




Trend + Seasonality

Seasonality + Cyclic

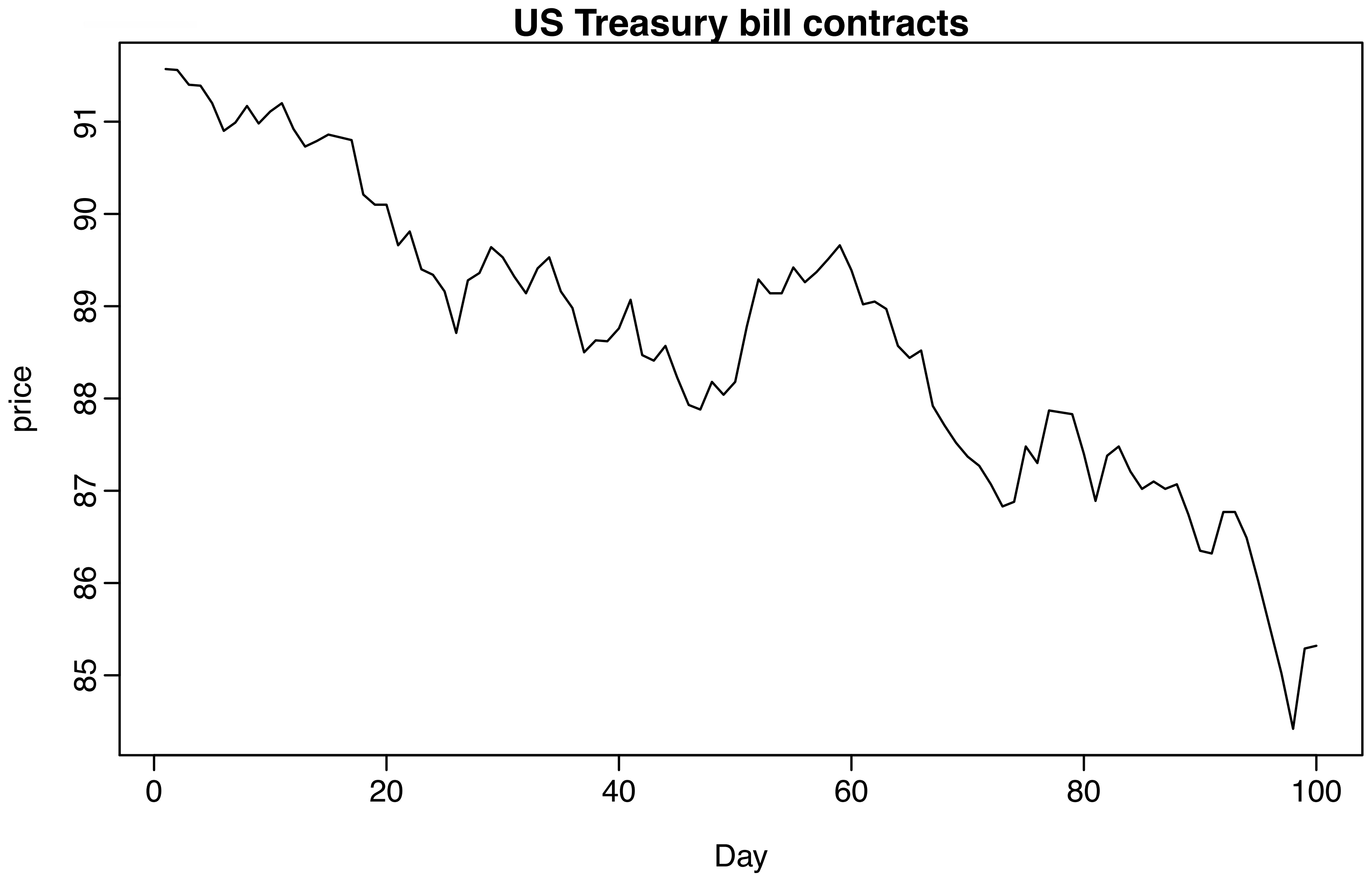




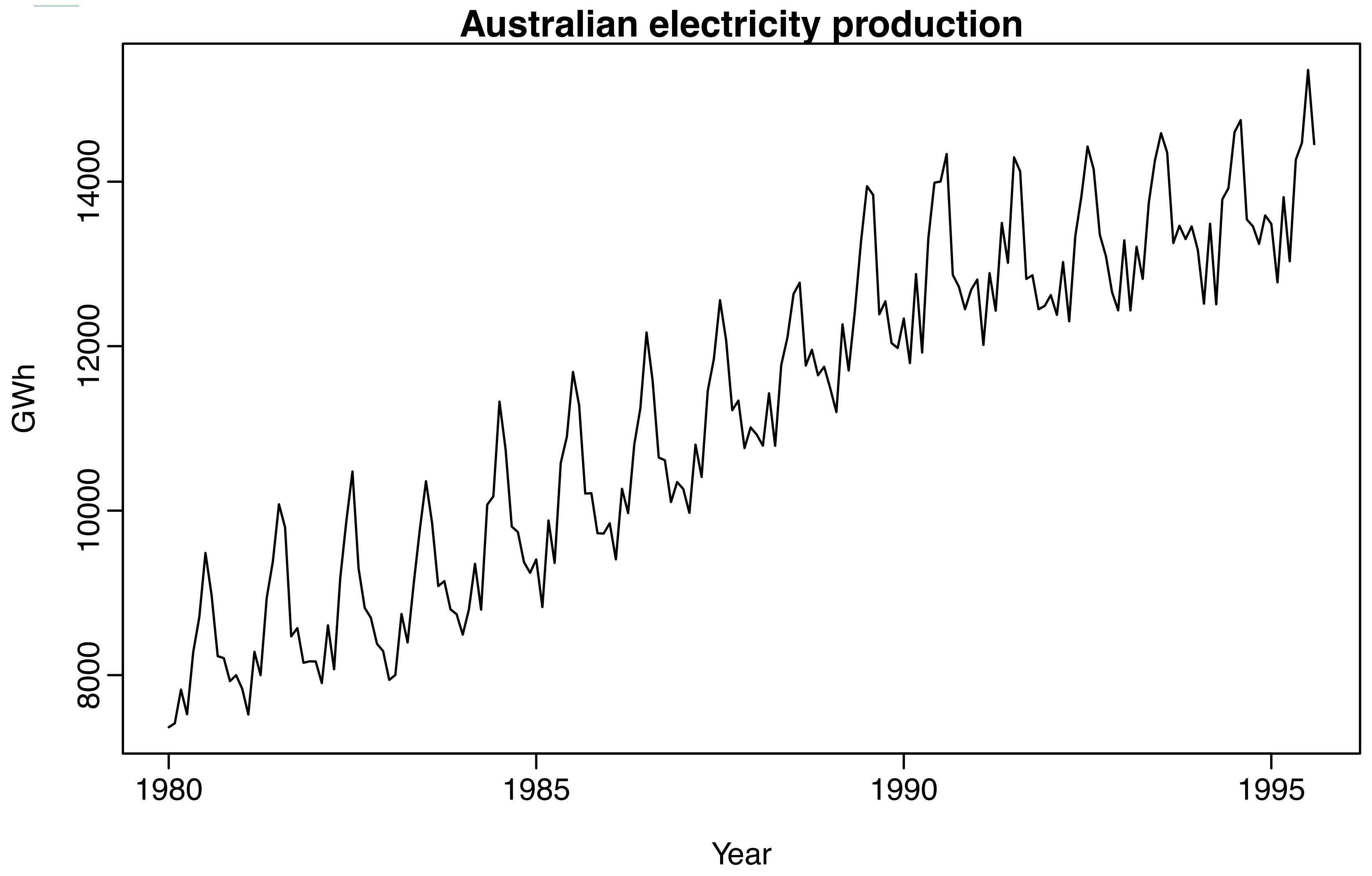
[R. J. Hyndman]

Time Series Data

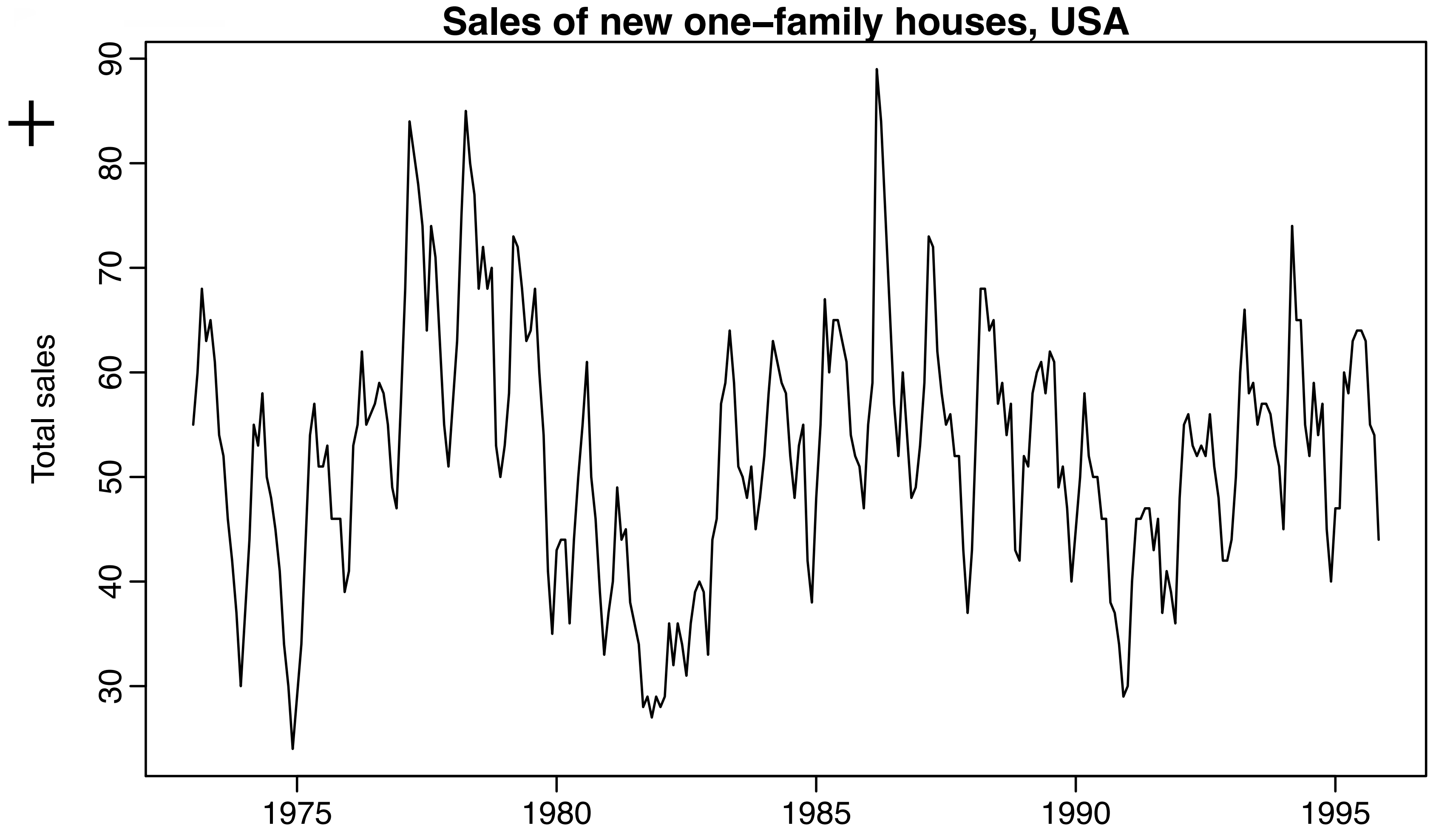
Trend



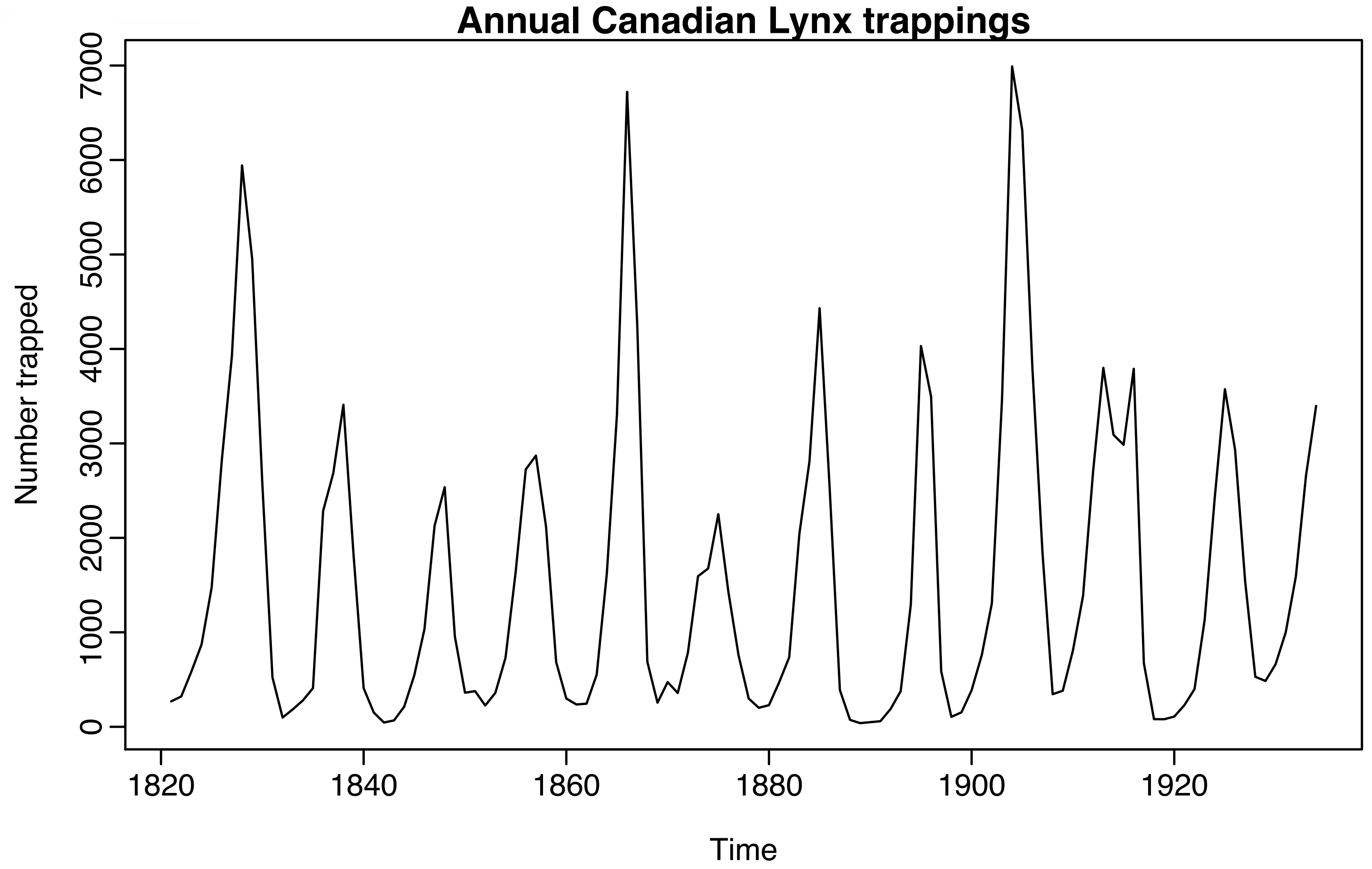
Trend + Seasonality



Seasonality + Cyclic

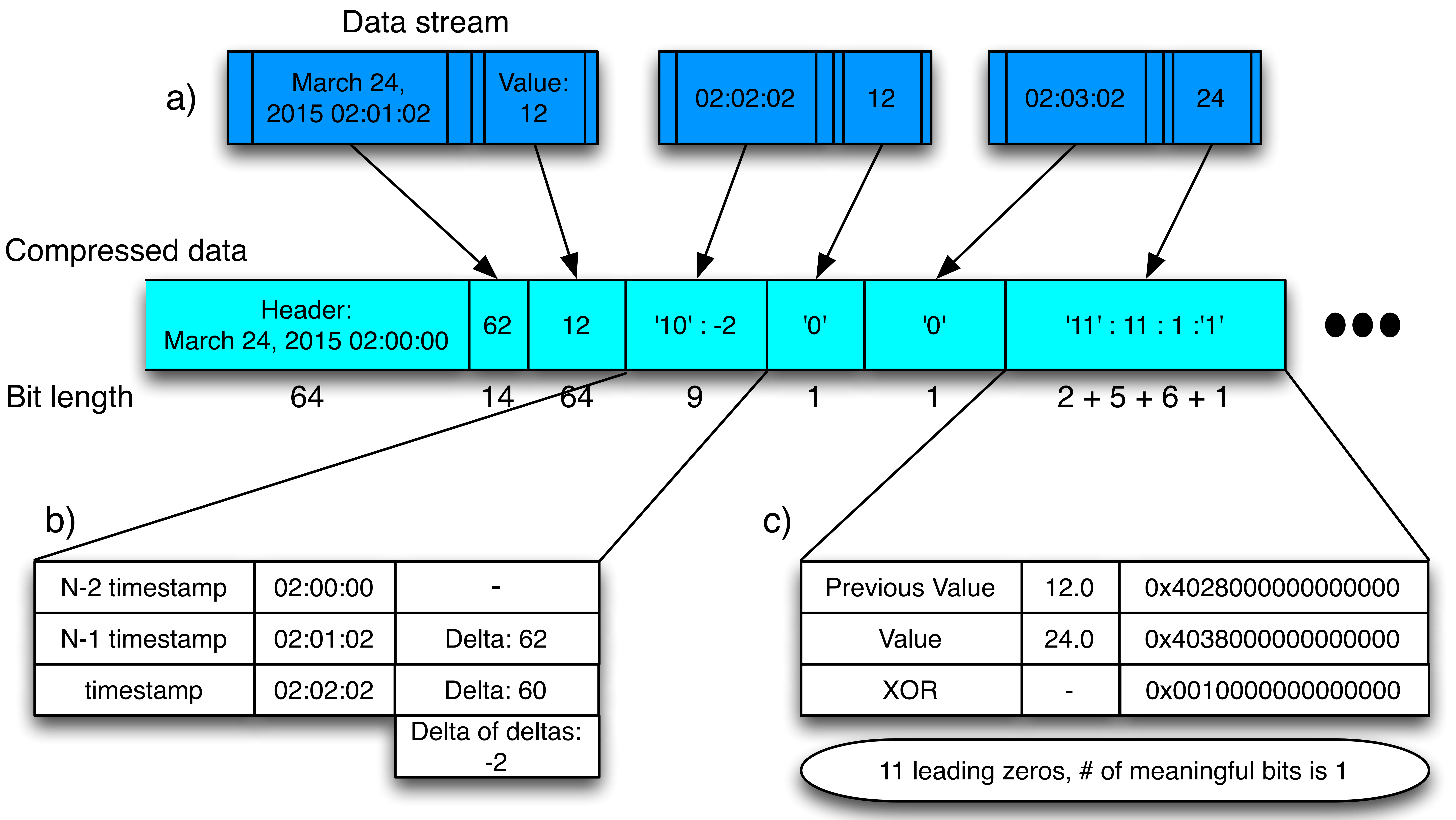


Stationary



[R. J. Hyndman]

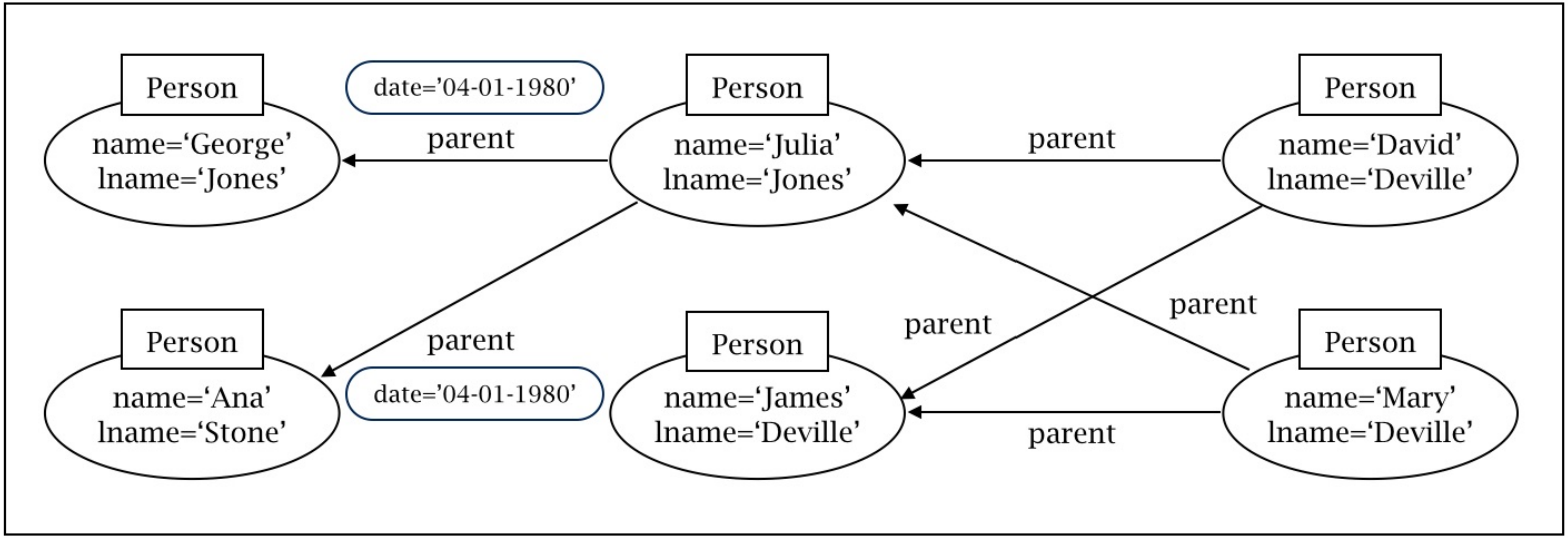
Gorilla Time Series Data Compression



[Pelkonen et al., 2015]

Graph Databases focus on relationships

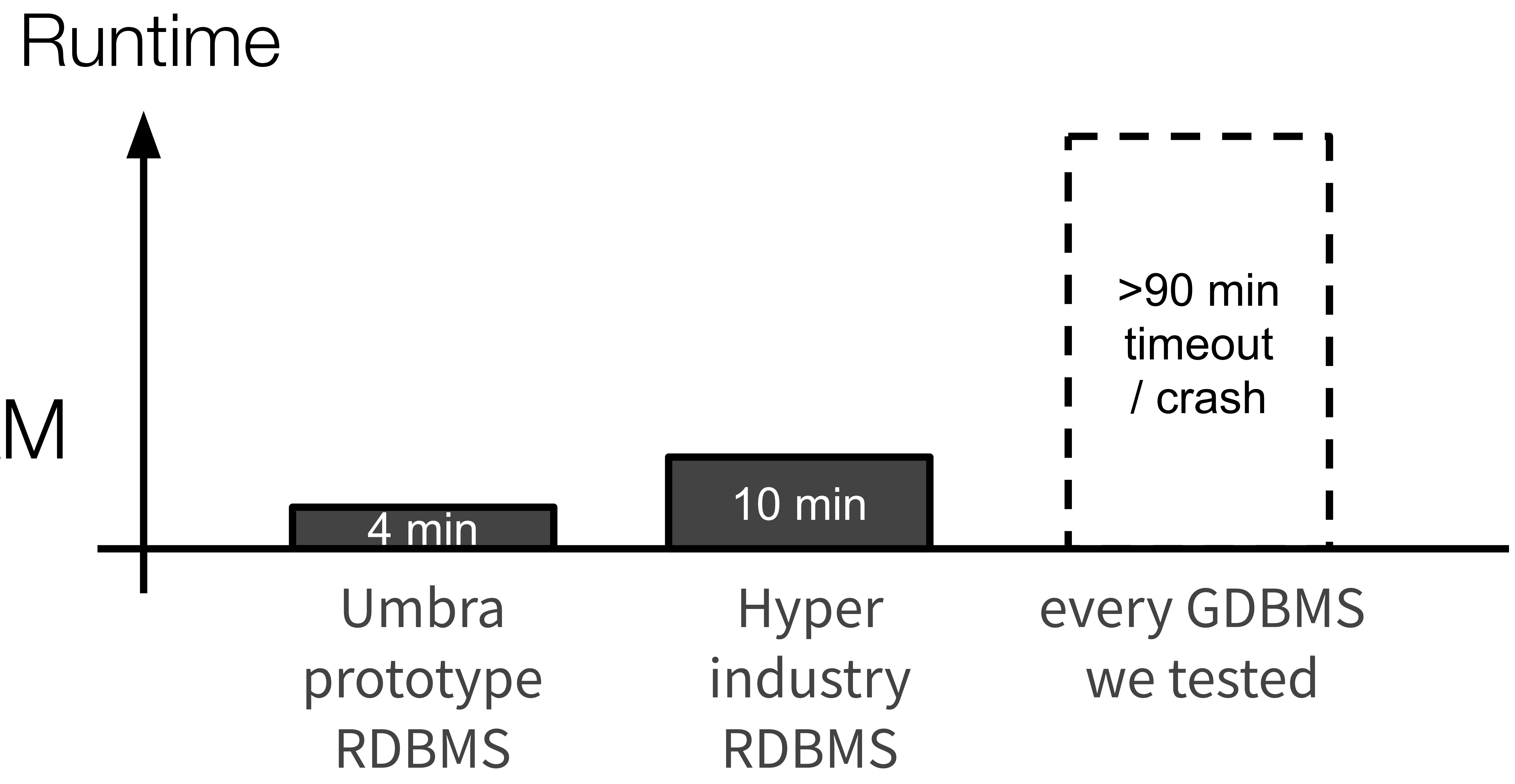
- Directed, labelled, attributed multigraph
- Properties are **key/value pairs** that represent metadata for nodes and edges



[R. Angles and C. Gutierrez, 2017]

Graph DBMS Problems

- performance
  - Slow loading speeds
  - Query speeds over magnitude slower than RDBMS
- scalability
  - Low datasize limit, typically << RAM
  - Little benefit from parallelism
- reliability
  - Loads never terminate
  - Query run out of memory or crash
  - Bugs



[P. Boncz, 2022]

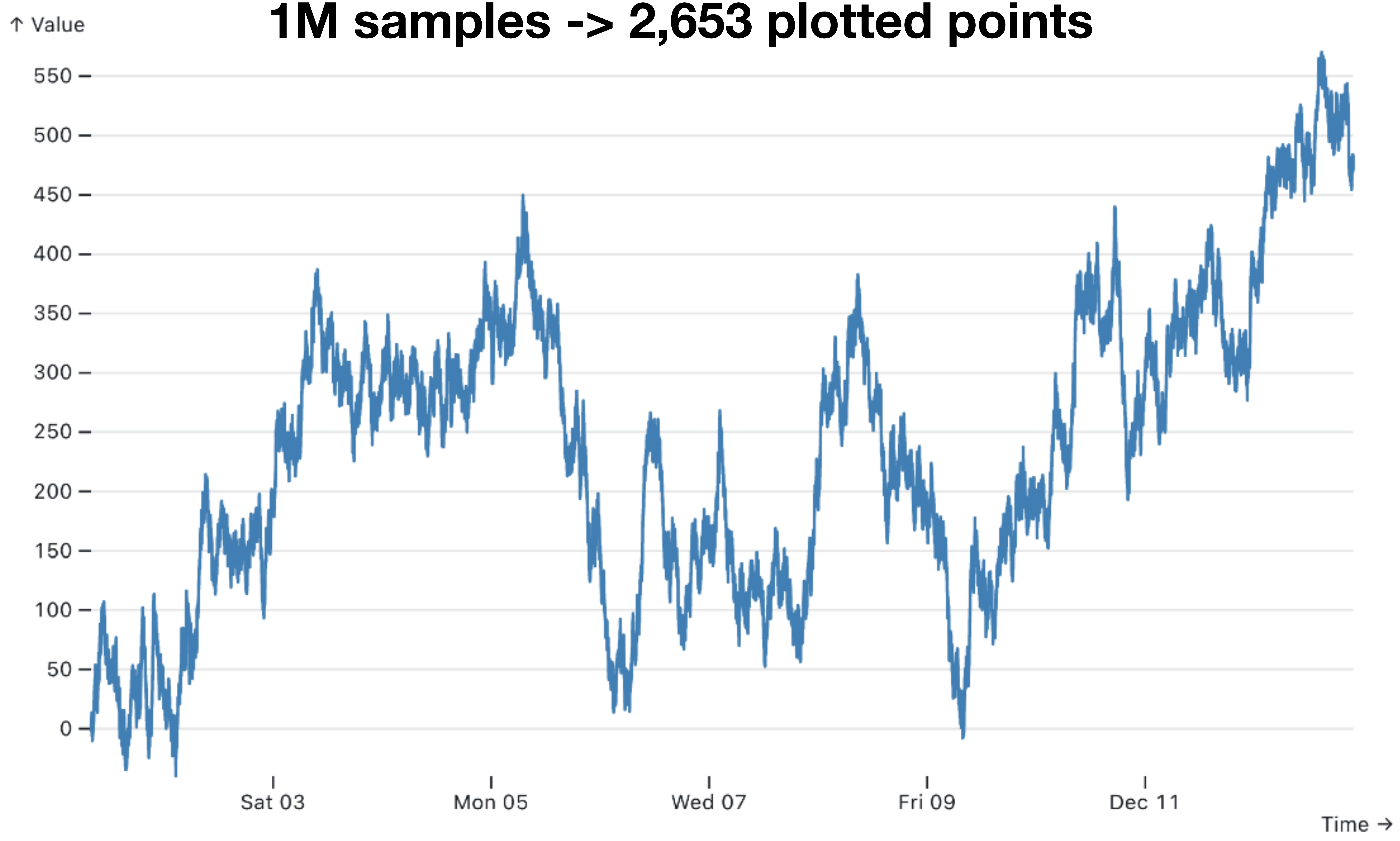
Supporting Scalable Visualization

- Two Problems:
  - **Lots of data**, how to display (encode) it
  - User **interaction** is key to gaining insight, requires **low latency**
- Addressing big data:
  - Encoding should focus on **available resolution**, not size of data
  - Approaches:
    - Sampling
    - Modeling
    - **Binning**
  - Bin → Aggregate (→ Smooth) → Plot

[J. Heer]

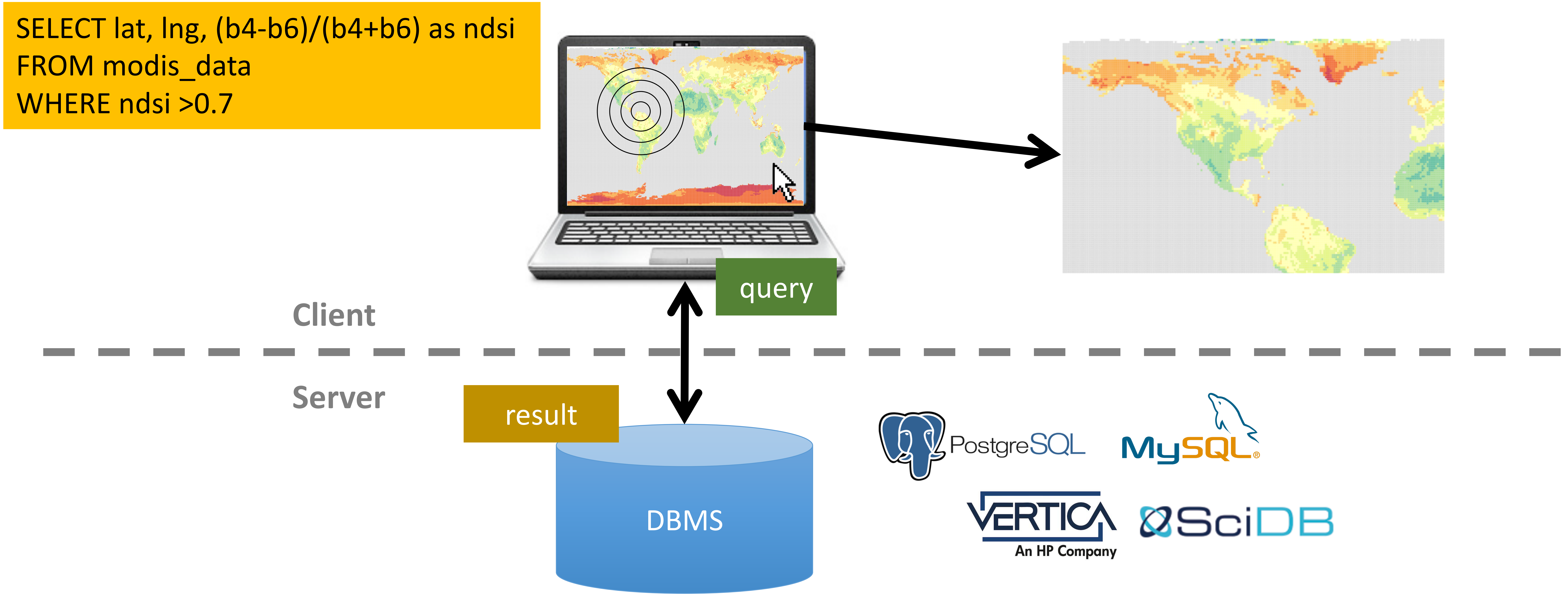
Time Series Aggregation

1M samples -> 2,653 plotted points



[J. Heer]

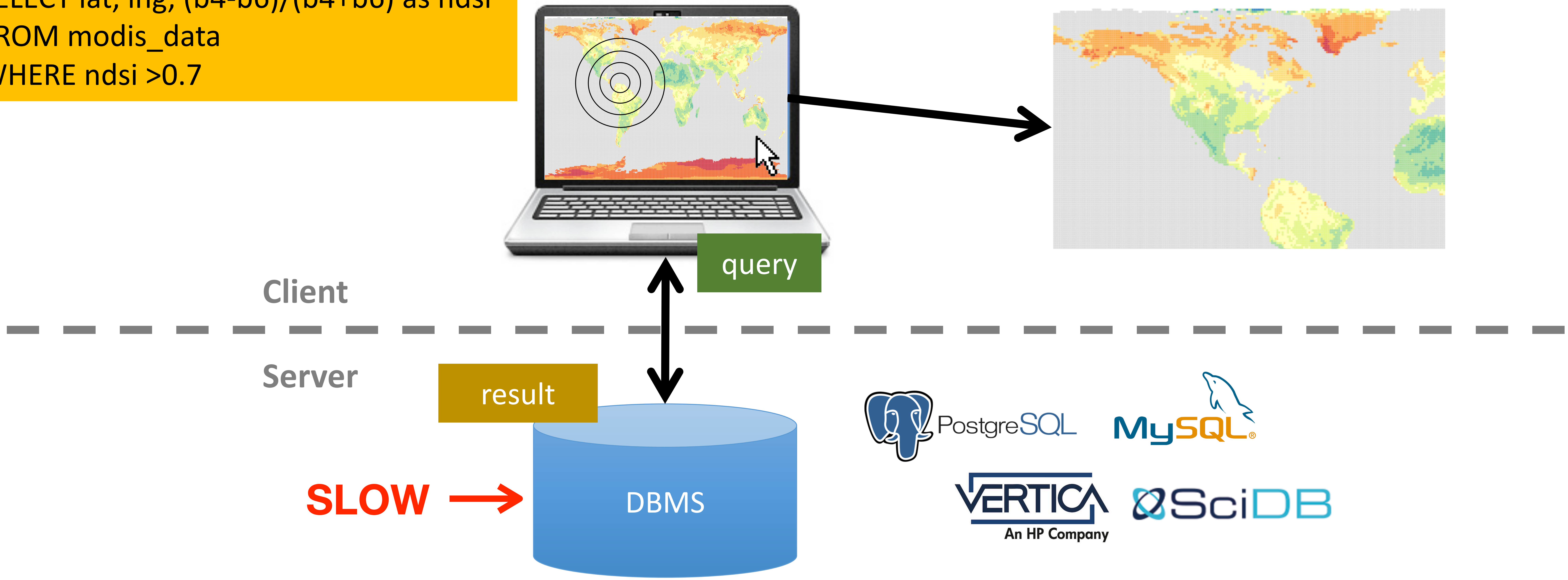
Interactive Exploration of Spatial Data



[L. Battle, 2017]

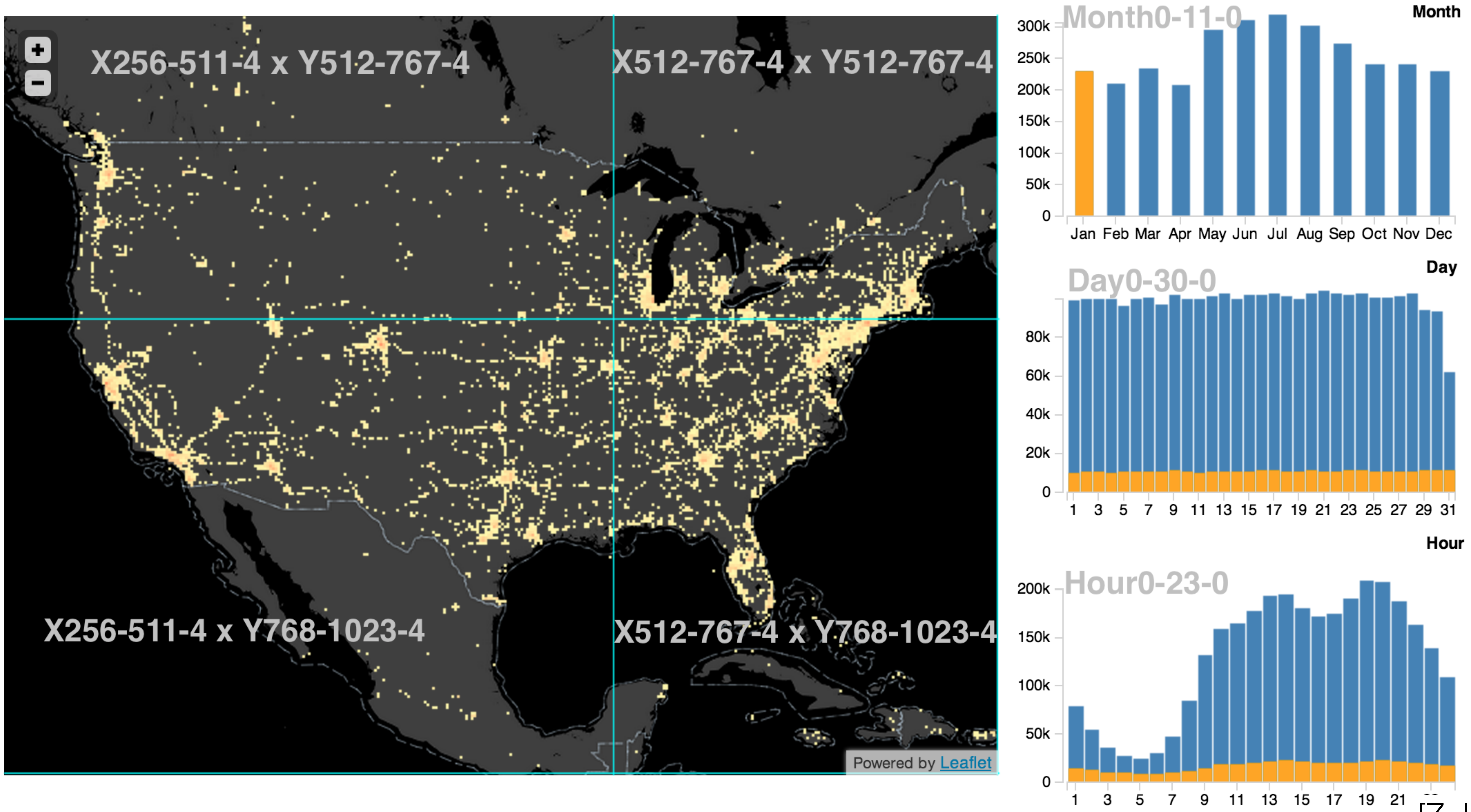
Interactive Exploration of Spatial Data

```
SELECT lat, lng, (b4-b6)/(b4+b6) as ndsi
FROM modis_data
WHERE ndsi >0.7
```



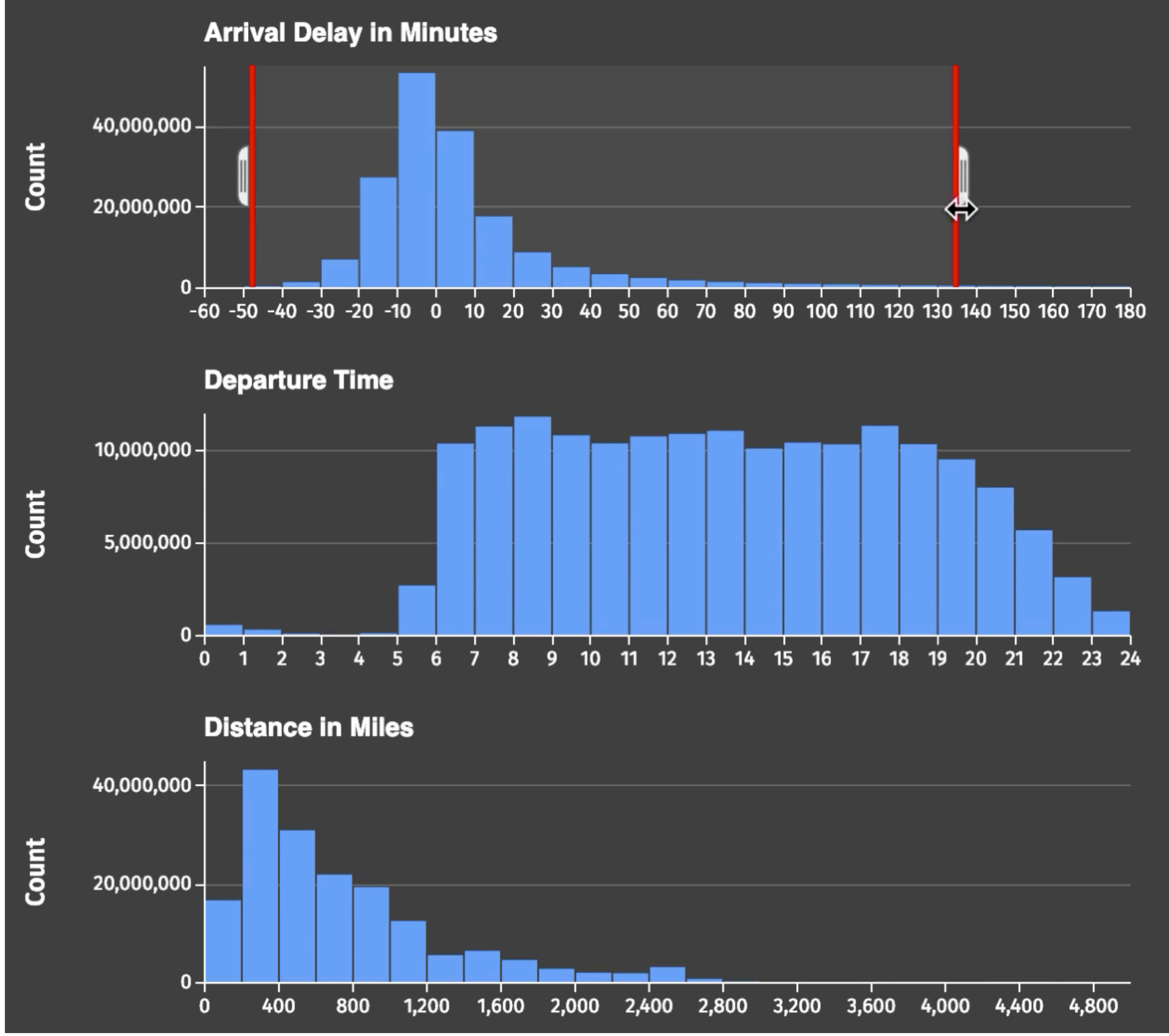
[L. Battle, 2017]

Visualization: Minimize Latency



[Z. Liu et al., 2013]

Visualization: Task-Prioritized Prefetching



brushes in the precomputed view

serves requests from a data cube

Data Cube. Gray et al. *1997.*

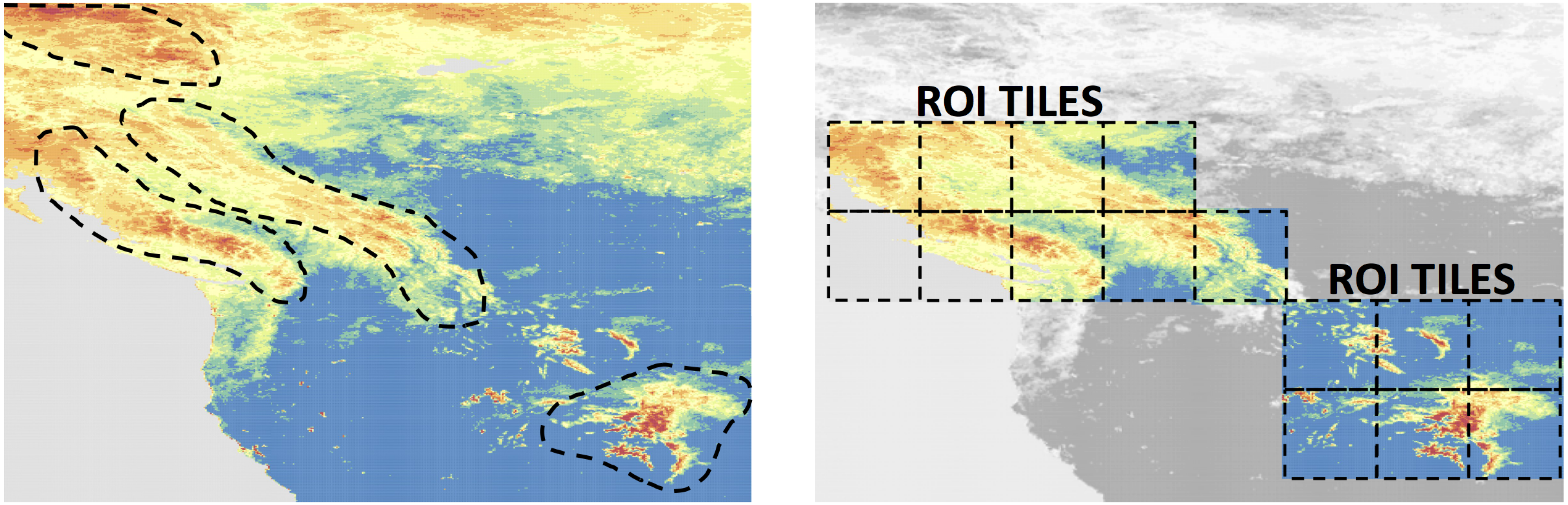
interacts with a new view

query for new data cubes

[D. Moritz et al. via J. Heer]

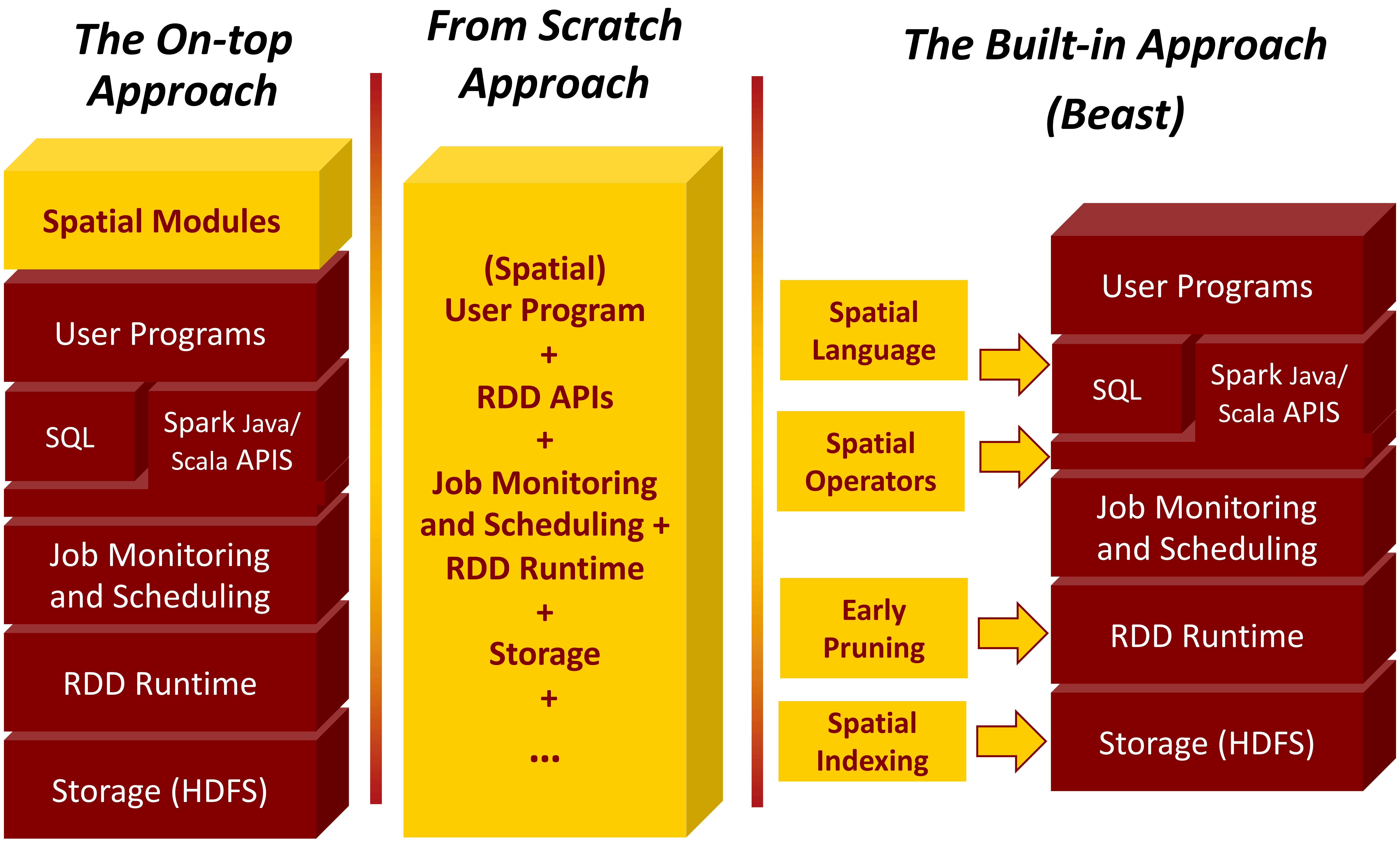
Visualization: Prefetching

- Predict which tiles a user will need next and prefetch those
  - Use common patterns (zoom, pan)
  - Use regions of interest (ROIs)



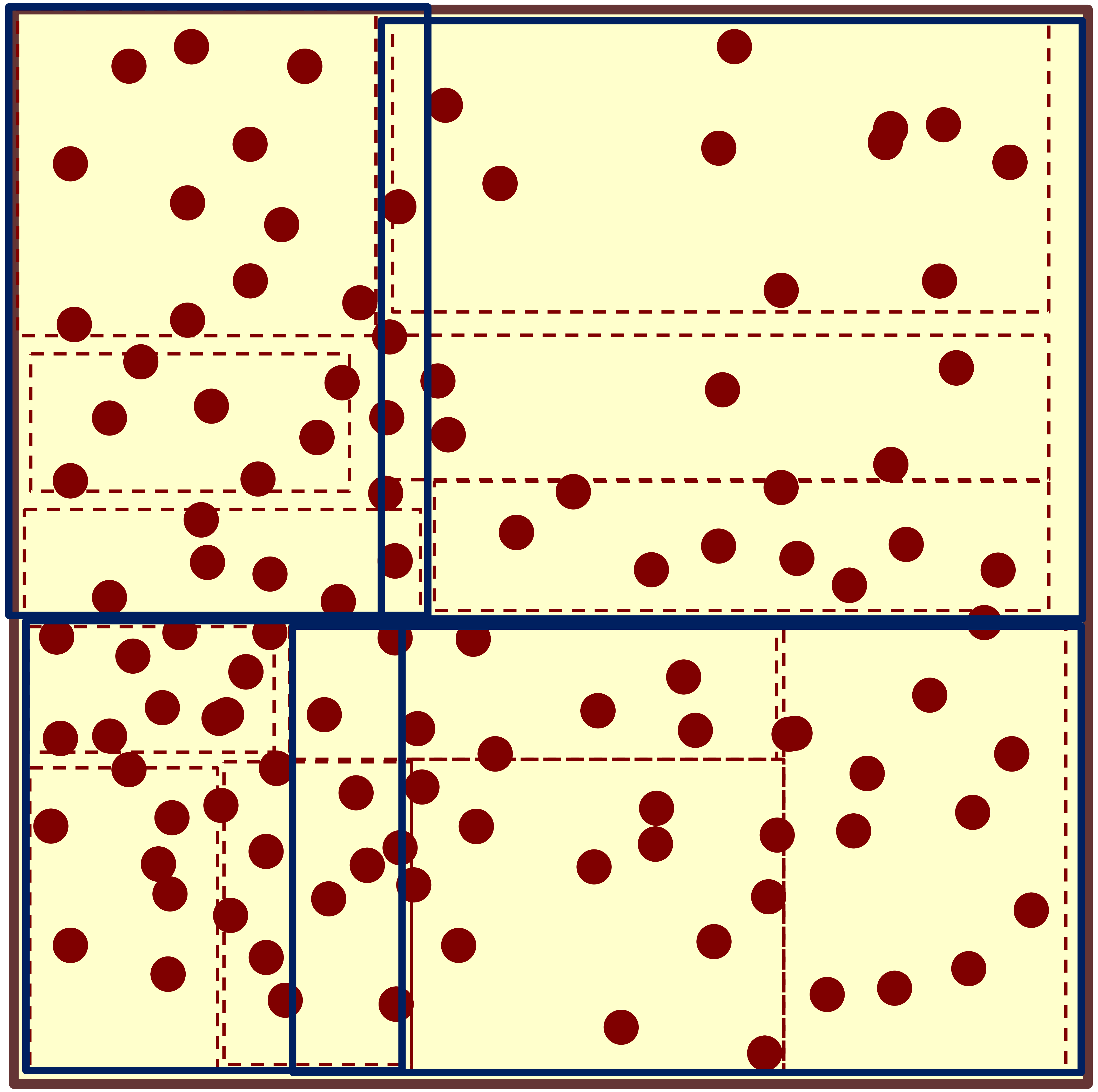
[Battle et al., 2016]

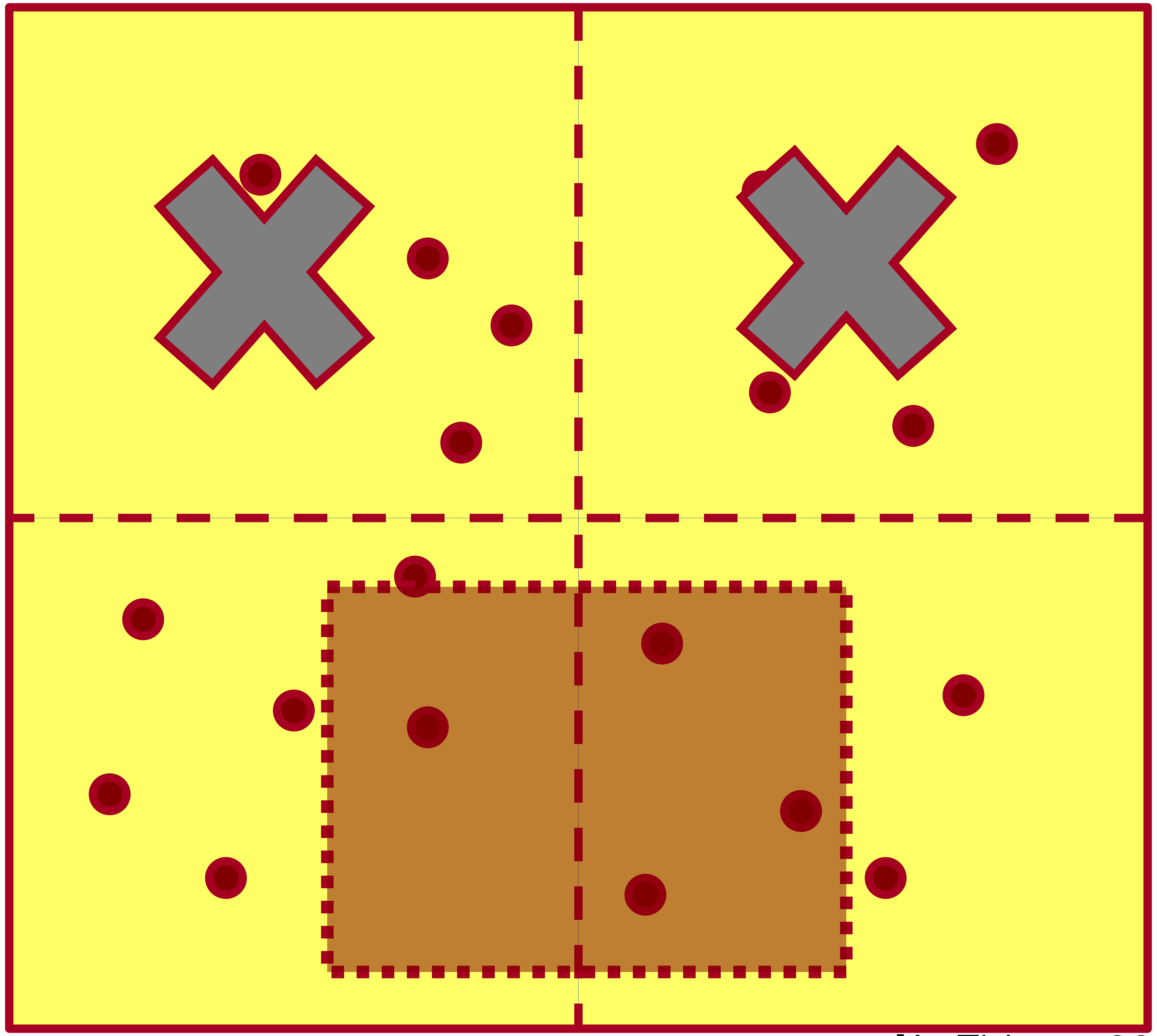
Spatial Data: Beast Architecture



[A. Eldawy, 2021]

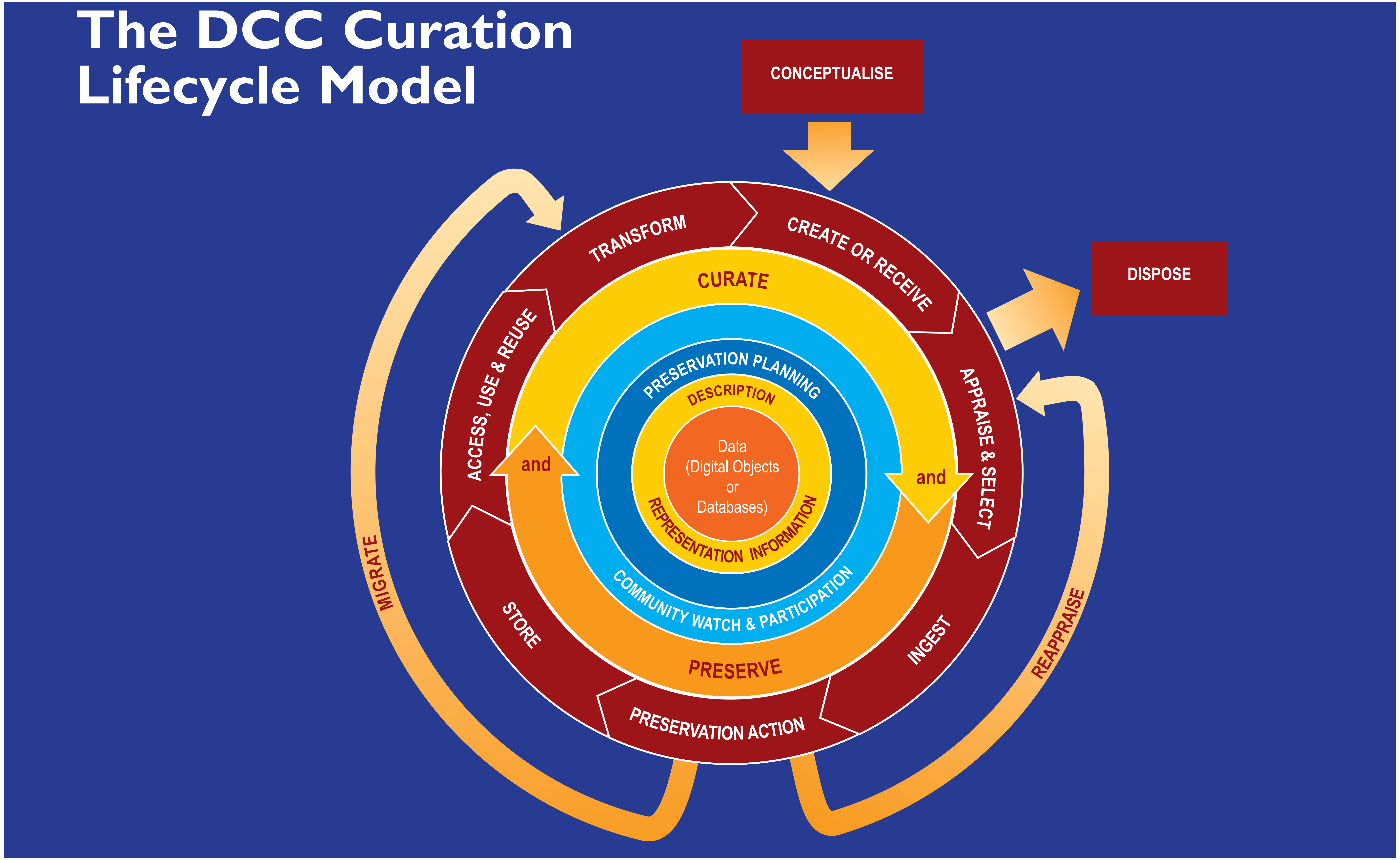
Spatial Data: Partitioning/Indexing & Range Query





[A. Eldawy, 2021]

Data Curation



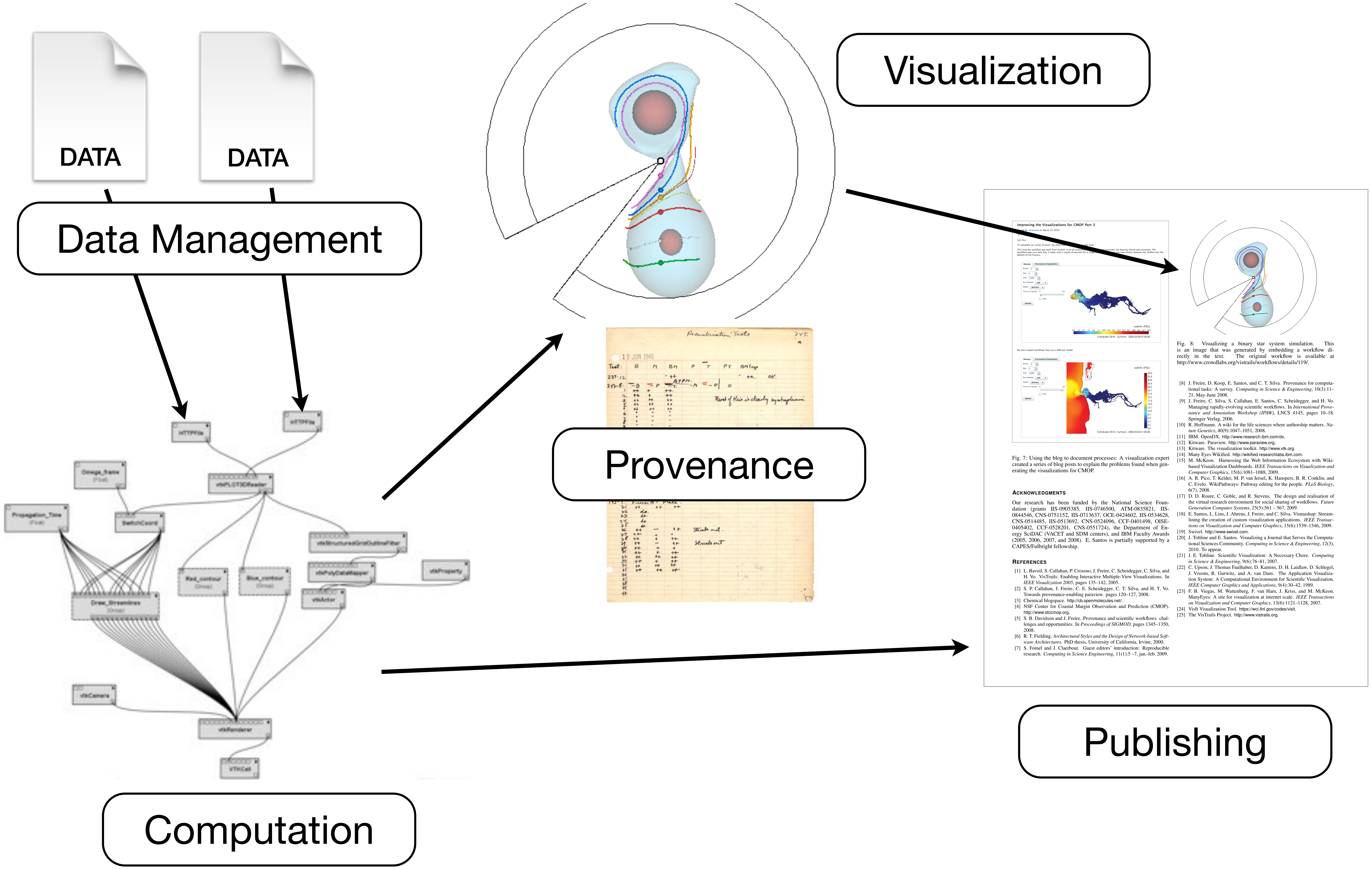
[DCC]

Data Curation: FAIR Principles

- Findable: Metadata and data should be easy to find for both humans and computers
- Accessible: Users need to know how data can be accessed, possibly including authentication and authorization
- Interoperable: Can be integrated with other data, and can interoperate with applications or workflows for analysis, storage, and processing
- Reusable: Optimize the reuse of data. Metadata and data should be well-described so they can be replicated and/or combined in different settings

[GO FAIR]

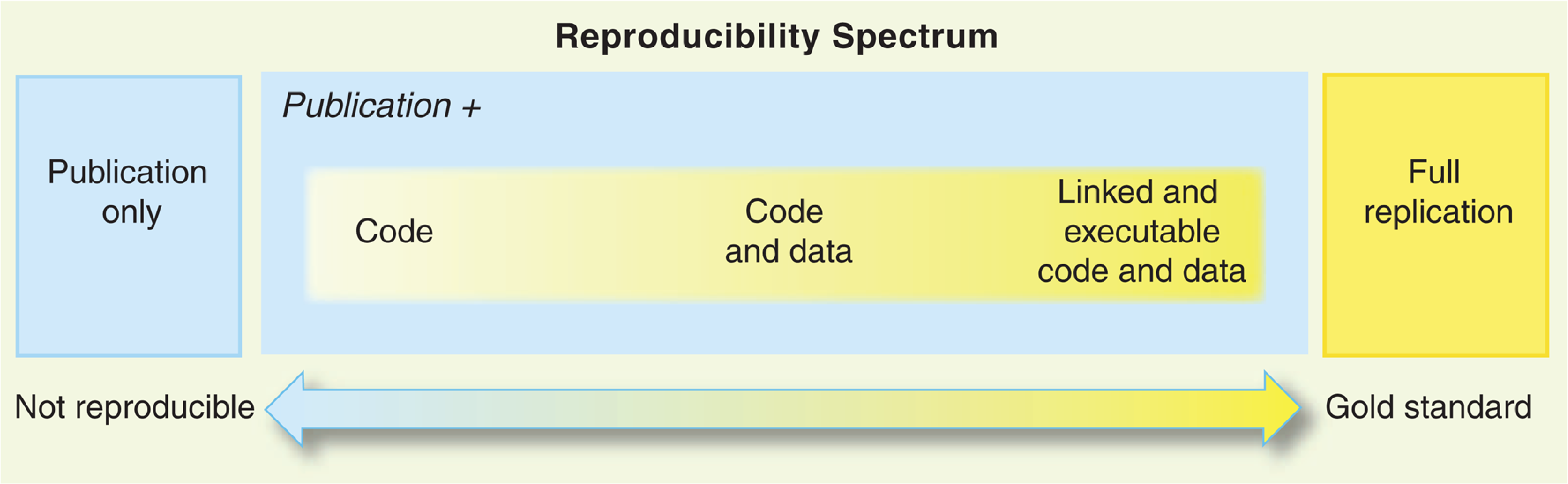
Provenance


DATA
DATA
Data Management
Visualization
Provenance
Computation
Publishing

Prospective and Retrospective Provenance

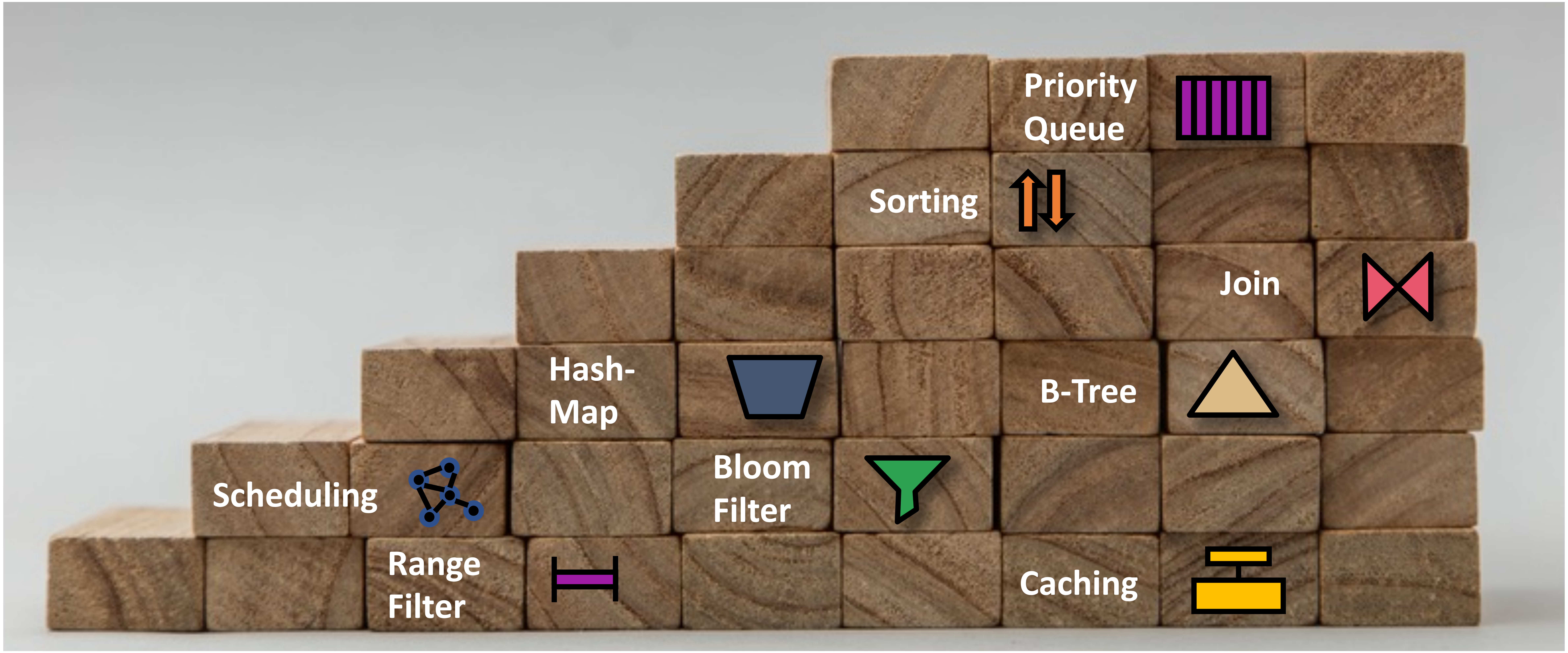
- Recipe for baking a cake versus the actual process & outcome
- Prospective provenance is what was specified/intended
  - a workflow, script, list of steps
- Retrospective provenance is what actually happened
  - actual data, actual parameters, errors that occurred, timestamps, machine information
- **Do not need** prospective provenance to have retrospective provenance!

Reproducibility



[R. D. Peng]

Machine Learning and Databases



[T. Kraska, 2019]

Questions?

Final Exam

- Wednesday, May 8, **8:00**-9:50am, PM 252
- Similar format
- More comprehensive (questions from topics covered in Test 1 & 2)
- Will also have questions from graph/spatial/temporal data, provenance, reproducibility, machine learning