Advanced Data Management (CSCI 490/680)

Scalable Database Systems

Dr. David Koop

Course Updates

- We want to help you **continue learning**… but we also realize that COVID-19 has **changed a lot** in our lives
- **I understand that this is not class as normal**
- If you have any circumstances that impact coursework, please contact me

Course Updates

- Office hours/Discussion/Questions: Blackboard Ultra Collaborate
  - **1:00-2:30pm on Tuesdays and Thursdays**
- Lectures: Online, recorded videos on Blackboard
- Discussions/Questions: Either Blackboard Ultra Collaborate or Discussions
- Reading Responses: Same as before (turn in via Blackboard)
- Assignments: Same as before (turn in via Blackboard)
- Reading Quizzes: Online via Blackboard
- Test 2 & Final Exam: Online via Blackboard

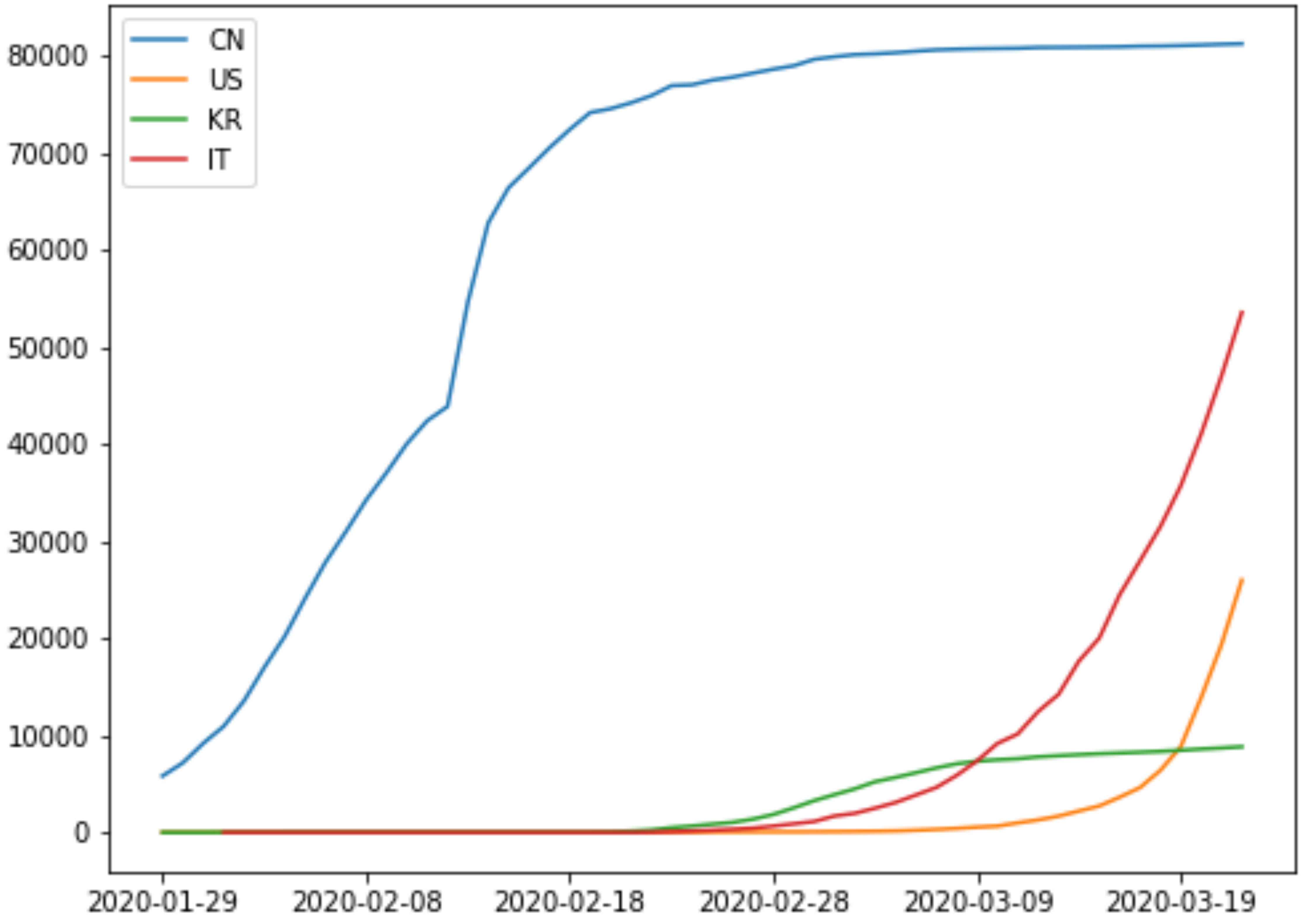
Where have we been?

- Focused on how to deal with data
- Topics
  - Tools: Python & pandas
  - Understanding Data
  - Data wrangling, cleaning, and transformation
  - Data integration & fusion
  - Data exploration & visualization

Where are we going?

- Topics:
  - Scalable databases
  - Data curation
  - Different dataset types: graphs, time series, and spatial data
  - Provenance and reproducibility
  - Data Management & Machine Learning

Assignment 4



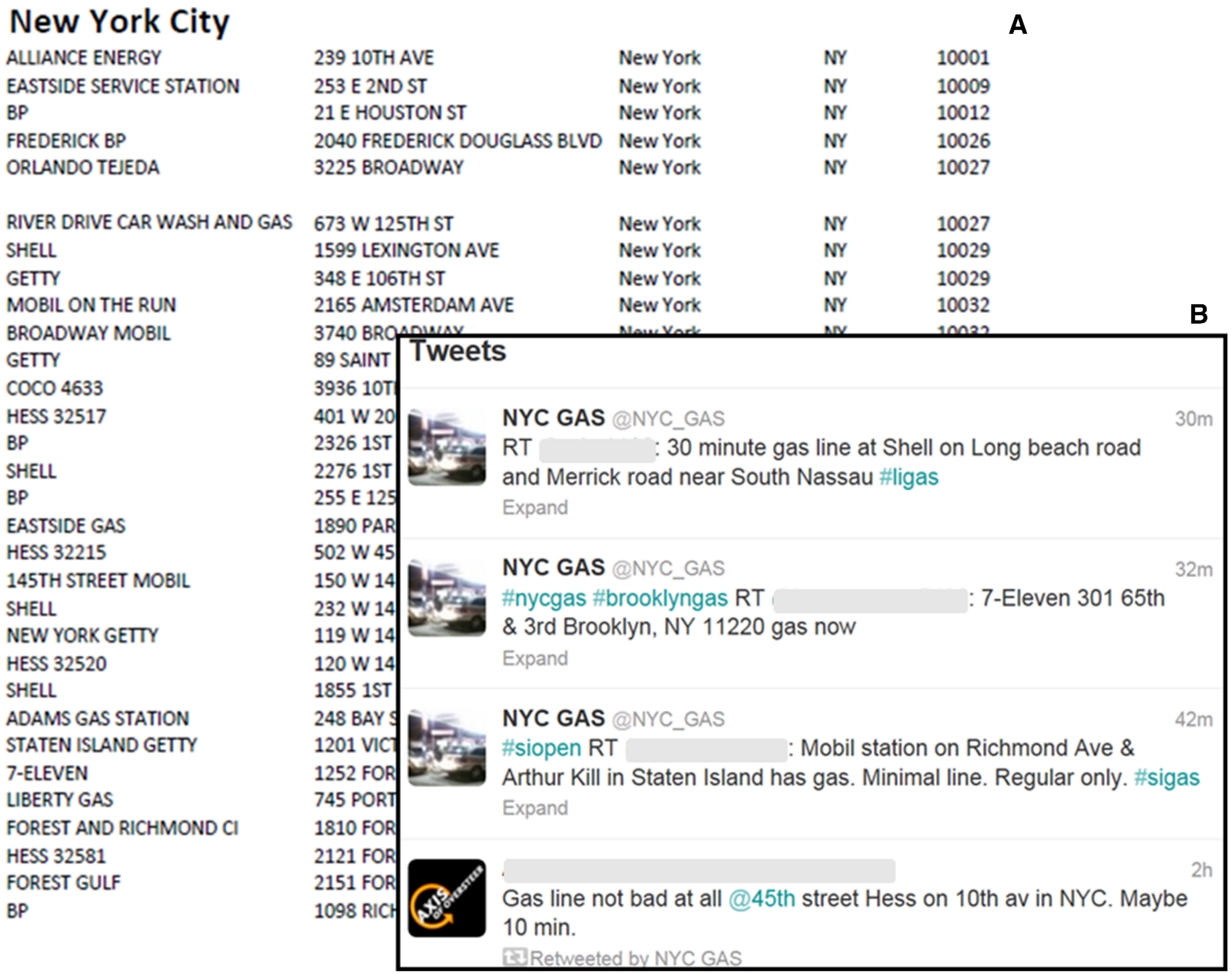
- COVID-19 data
- Data Integration
  - Population
  - Temperature
- Data Fusion:
  - Our World in Data
  - Johns Hopkins
  - Wikipedia
- Questions?

Reading Quiz

- Reading for Thursday:
  - Spanner: Google's Globally-Distributed Database
- Before watching Thursday's lecture, take Blackboard quiz on the reading
- Quiz will focus on key concepts not the details

Data Discovery and Visualization

Goal of Dataset Search: Accurate (A) vs. Timely (B)


A

New York City

ALLIANCE ENERGY	239 10TH AVE	New York	NY	10001
EASTSIDE SERVICE STATION	253 E 2ND ST	New York	NY	10009
BP	21 E HOUSTON ST	New York	NY	10012
FREDERICK BP	2040 FREDERICK DOUGLASS BLVD	New York	NY	10026
ORLANDO TEJEDA	3225 BROADWAY	New York	NY	10027
RIVER DRIVE CAR WASH AND GAS	673 W 125TH ST	New York	NY	10027
SHELL	1599 LEXINGTON AVE	New York	NY	10029
GETTY	348 E 106TH ST	New York	NY	10029
MOBIL ON THE RUN	2165 AMSTERDAM AVE	New York	NY	10032
BROADWAY MOBIL	3740 BRO			
GETTY	89 SAINT			
COCO 4633	3936 10T			
HESS 32517	401 W 20			
BP	2326 1ST			
SHELL	2276 1ST			
BP	255 E 125			
EASTSIDE GAS	1890 PAR			
HESS 32215	502 W 45			
145TH STREET MOBIL	150 W 14			
SHELL	232 W 14			
NEW YORK GETTY	119 W 14			
HESS 32520	120 W 14			
SHELL	1855 1ST			
ADAMS GAS STATION	248 BAY			
STATEN ISLAND GETTY	1201 VIC			
7-ELEVEN	1252 FOR			
LIBERTY GAS	745 PORT			
FOREST AND RICHMOND CI	1810 FOR			
HESS 32581	2121 FOR			
FOREST GULF	2151 FOR			
BP	1098 RICH			

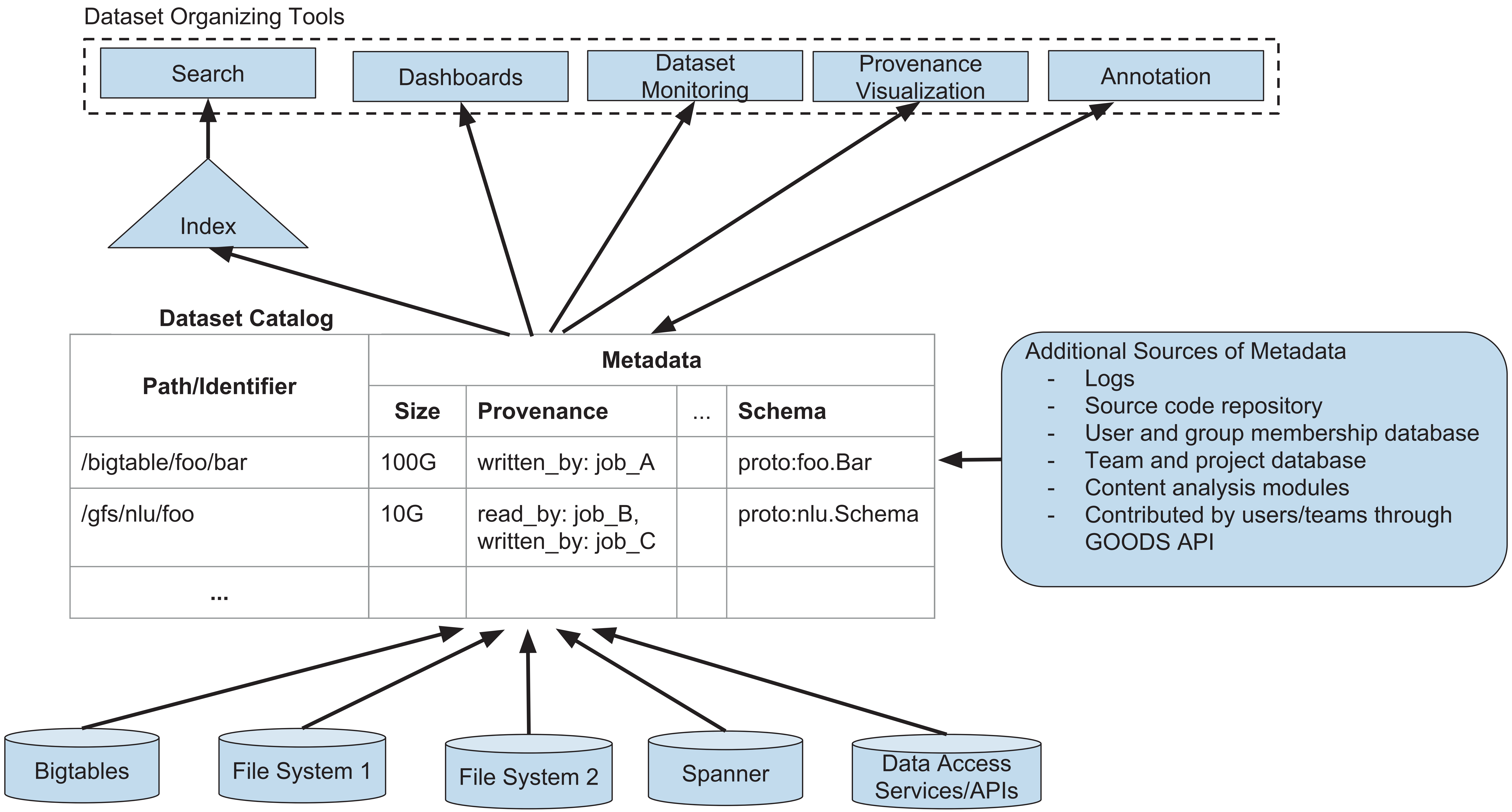
[Chapman et al., 2020]

Goods: Organizing Google's Datasets

- Tool for Google to help its employees find internal data
- Keep data where it is, how it is, but extract metadata to aid search
- Challenges:
  - Dataset size and scale: >26 billion datasets
  - Variety: formats (text, csv, Bigtable), storage (GoogleFS, db server)
  - Churn: ~5% of datasets deleted each day
  - Metadata uncertainty: protocol buffers, primary key identification
  - Computing importance: need to understand users
  - Recovering semantics: understanding the data aids metadata extraction

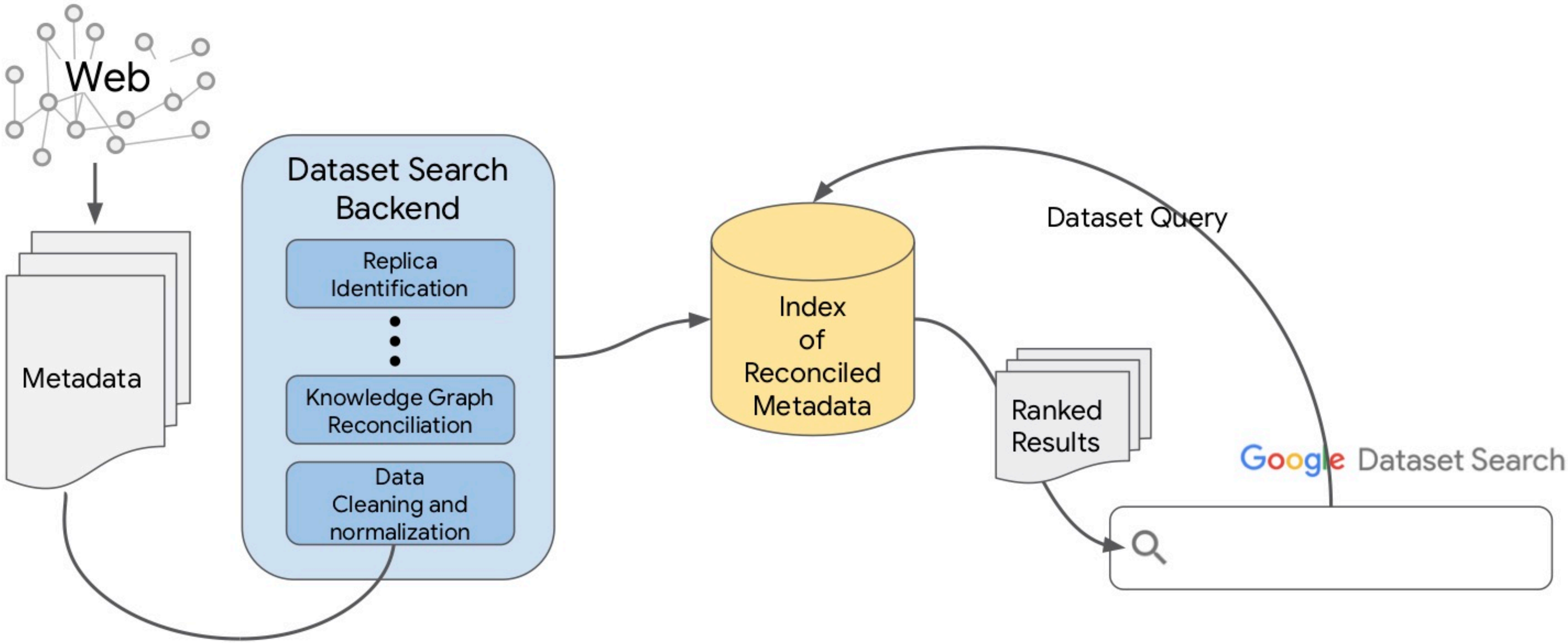
[Halevy et al., 2016]

Goods: Organizing Google's Datasets



[Halevy et al., 2016]

Google Dataset Search Overview



[N. Noy et al., 2019]

Requirements

- System must be **open** so new providers can add their own datasets
- Search is over **metadata** (a provider may require users to pay/create account)
- Metadata must be published by the data publishers themselves, adhering to a **standard**

[N. Noy et al., 2019]

Challenges

- Metadata Quality: providers don't adhere to the specs
- Metadata Duplication in Search Results: search results vs. profile pages
- Dataset Replication and Provenance: identify replicas across providers
- Churn and Stale Sites:
  - 3% deleted, 7-10% added per day
  - standard web crawlers check high-traffic sites more often
- Ranking/Relevance: data citation might help
- Multiple Dataset-Metadata Standards: schema.org vs DCAT

[N. Noy et al., 2019]

Data Visualization

What does it mean to summarize data?

	name	country	primary_fuel	capacity_mw	generation_gwh	latitude	longitude
0	PHILIPPSBURG-2	DEU	Nuclear	1468.000	12763.886134	49.2529	8.4364
1	Kvinen	NOR	Hydro	90.000	403.532191	58.9281	7.0838
2	Torrance Farm Wind Park Extension	GBR	Wind	5.700	10.342942	55.8766	-3.7452
3	Nam Phom	THA	Gas	710.000	2906.785455	16.6820	102.7422
4	Winterset	USA	Oil	13.600	0.103000	41.3372	-94.0126
5	Twin Elm Solar Farm	GBR	Solar	1.000	0.499932	51.3580	-2.7959
6	Tesoro Hawaii	USA	Oil	20.000	156.520696	21.3032	-158.0914
7	Prospector Windfarm (Burnt River)	USA	Wind	10.000	1.774076	44.4181	-117.2550
8	Caterpillar	USA	Oil	13.300	33.826000	40.4175	-86.8447
9	Bourogne	FRA	Biomass	8.862	34.817396	47.5647	6.9062
10	CED Ducor 2	USA	Solar	20.000	8.918553	35.8442	-119.0681
11	Edenderry	IRL	Oil	117.600	19.196229	53.2902	-7.0845
12	Kangaroo Valley (Shoalhaven Scheme)	AUS	Hydro	160.000	140.743389	-34.7230	150.4795
13	Wheelabrator Baltimore Refuse	USA	Waste	64.500	336.882000	39.2660	-76.6297
14	Guangxi Yulin Darongshan Wind	CHN	Wind	25.000	76.523288	22.8722	110.2332
15	Bernards Solar	USA	Solar	3.700	1.649932	40.7027	-74.5816
16	Manor Farm - Shripney	GBR	Solar	5.000	2.499661	50.8148	-0.6825
17	Saint-Cirgues-en-Montagne	FRA	Wind	18.450	35.116073	44.7679	4.0883
18	Turweston Solar Farm	GBR	Solar	16.700	8.348868	52.0328	-1.1053
...	...	...	...	...	...	...	...

Counts & Stats?

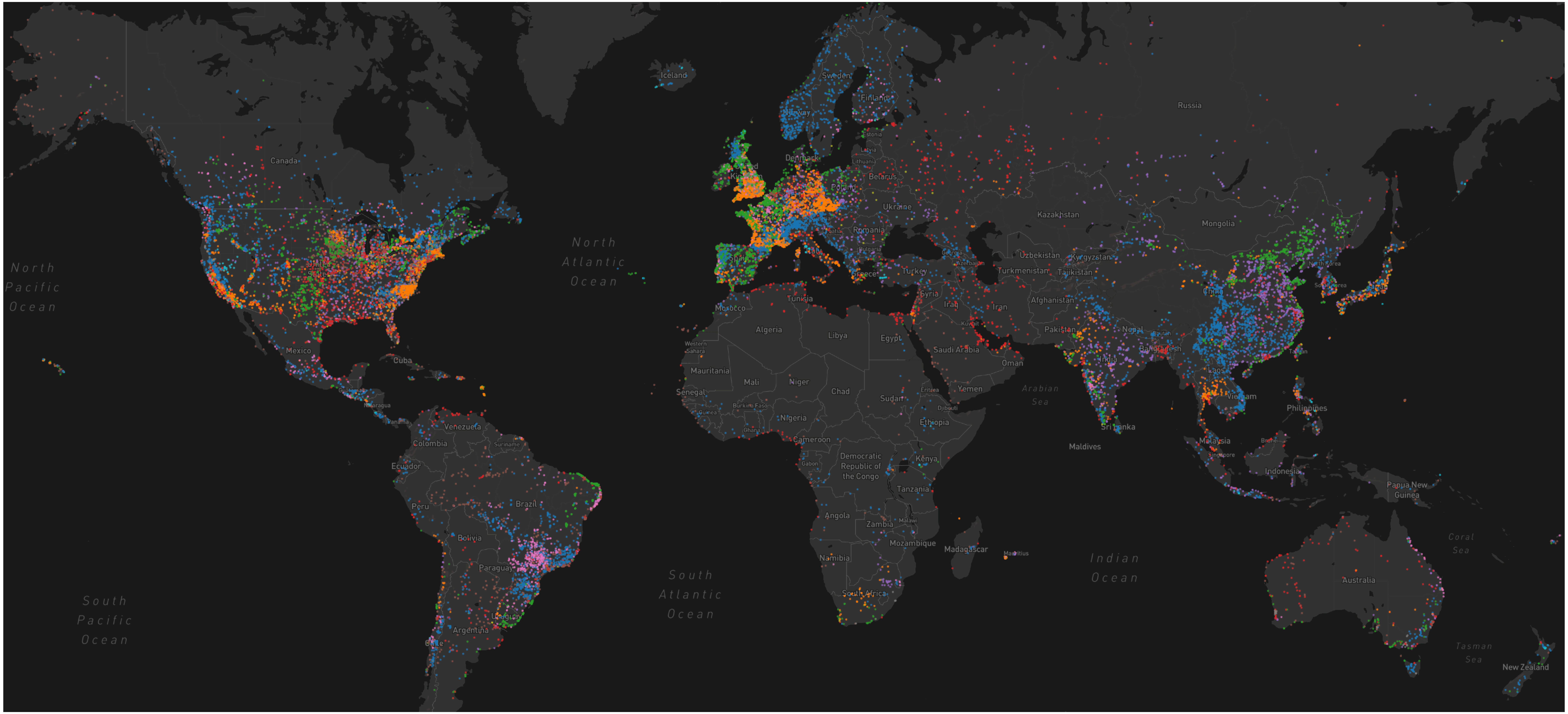
primary_fuel	num_plants
Hydro	7155
Solar	5929
Wind	5188
Gas	3922
Coal	2390
Oil	2290
Biomass	1396
Waste	1087
Nuclear	198
Geothermal	189
...	…

country	num_plants
USA	8686
CHN	3041
GBR	2536
BRA	2340
FRA	2017
CAN	1154
DEU	982
IND	861
ESP	614
RUS	505
...	…

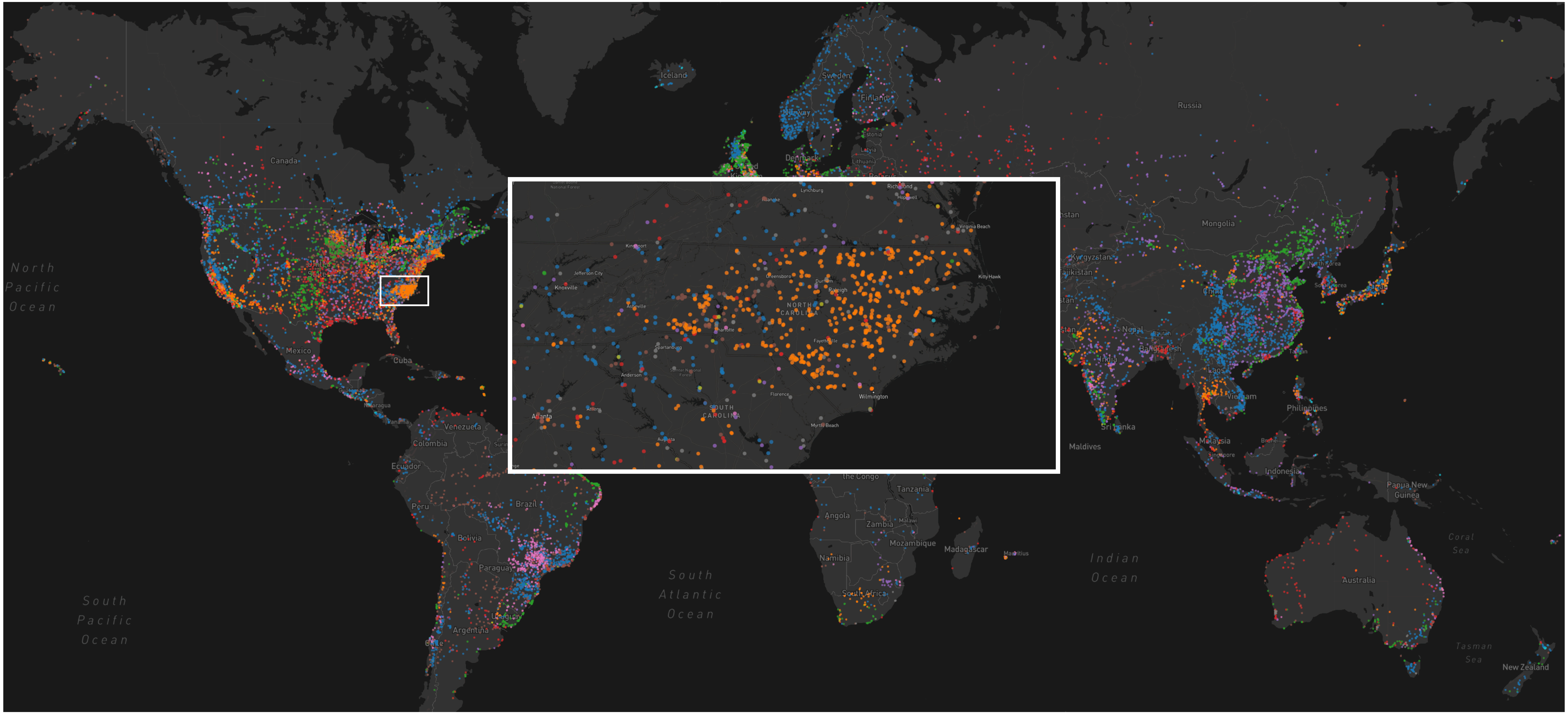
	generation_gwh
mean	806.709784
median	47.308833
std	3696.765694

	capacity_mw
mean	186.294810
median	18.900000
std	525.703572

Visual Summary



Visual Summary



Visual Summary Problems

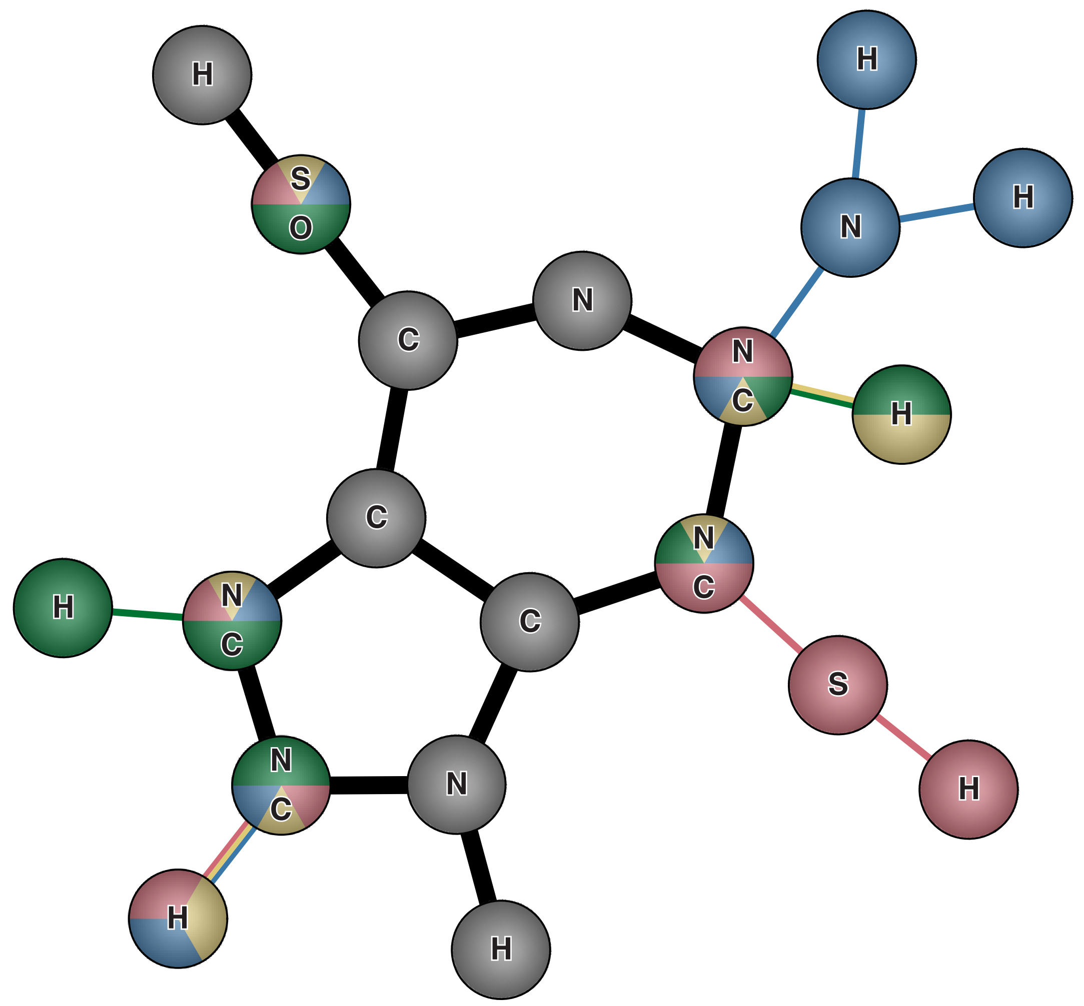
- Too much data
  - Cannot display all of it without overlap (**occlusion**)
  - A limited number of pixels
  - A limited amount of **human-resolvable** resolution
- Prioritizing the display data is non-trivial
  - Show a lot of tiny power plants and occlude
  - Show only the big power plants

Visual Information-Seeking Mantra

- Overview First
- Zoom & Filter
- Details on Demand

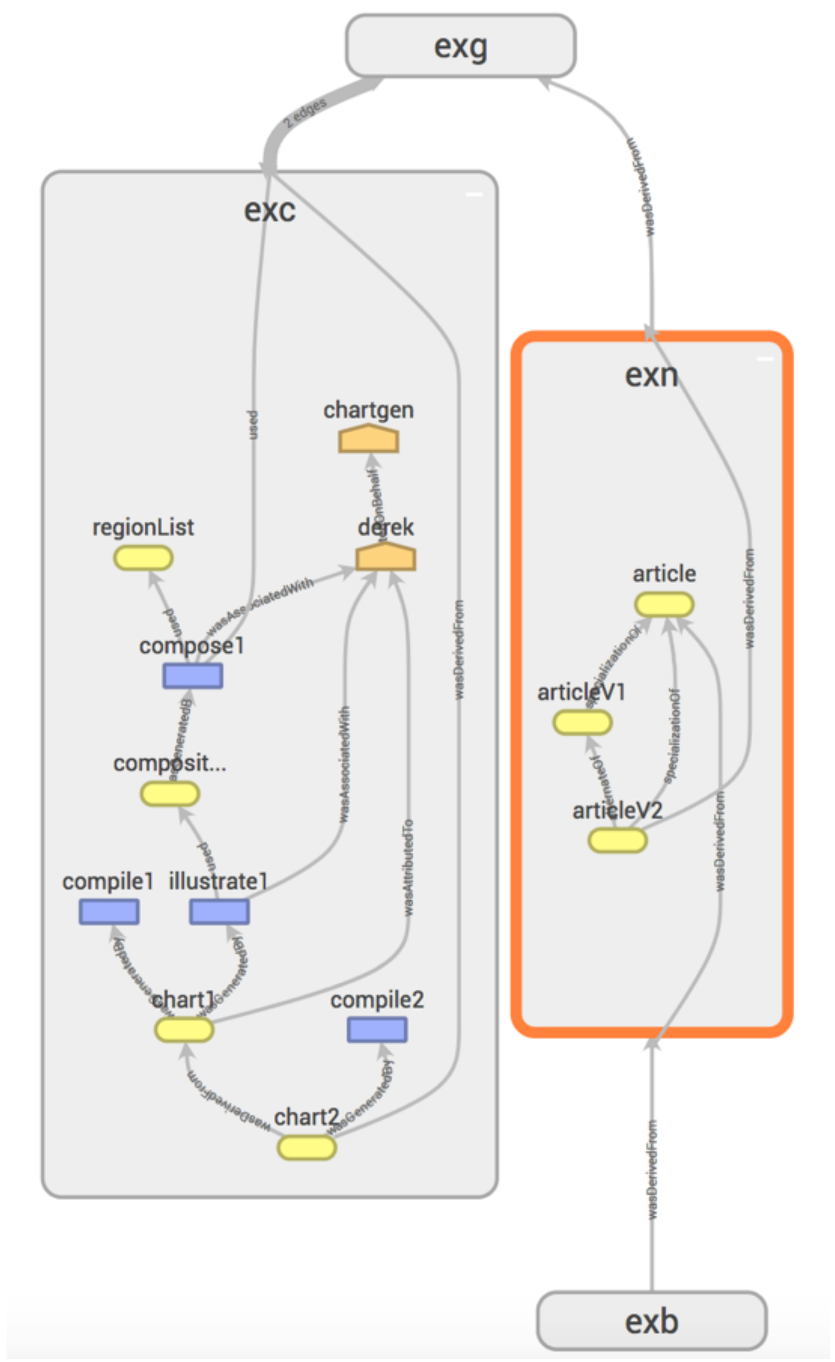
–Schneiderman, 1996

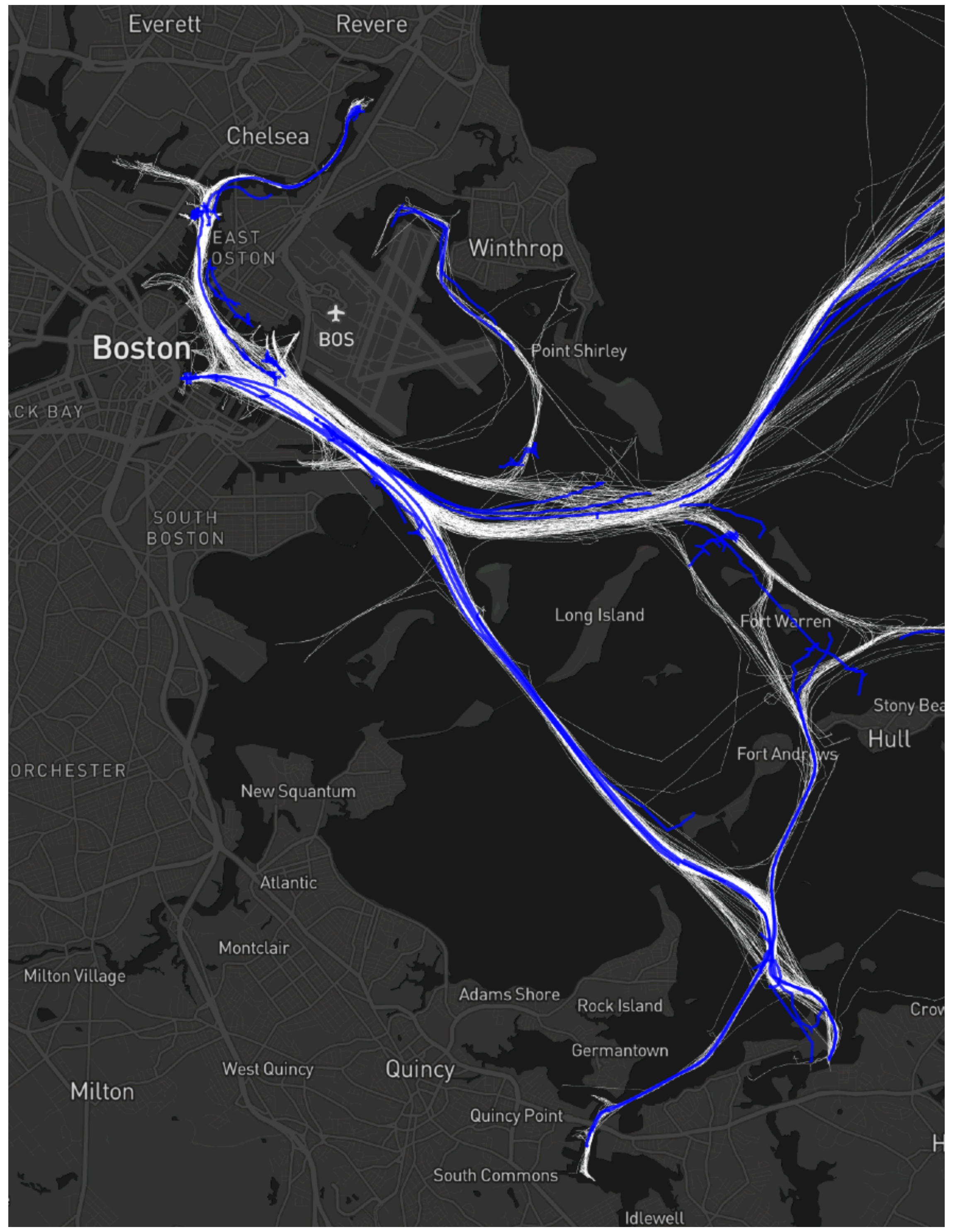
Visual Summarization Projects



Graph Collection Summaries

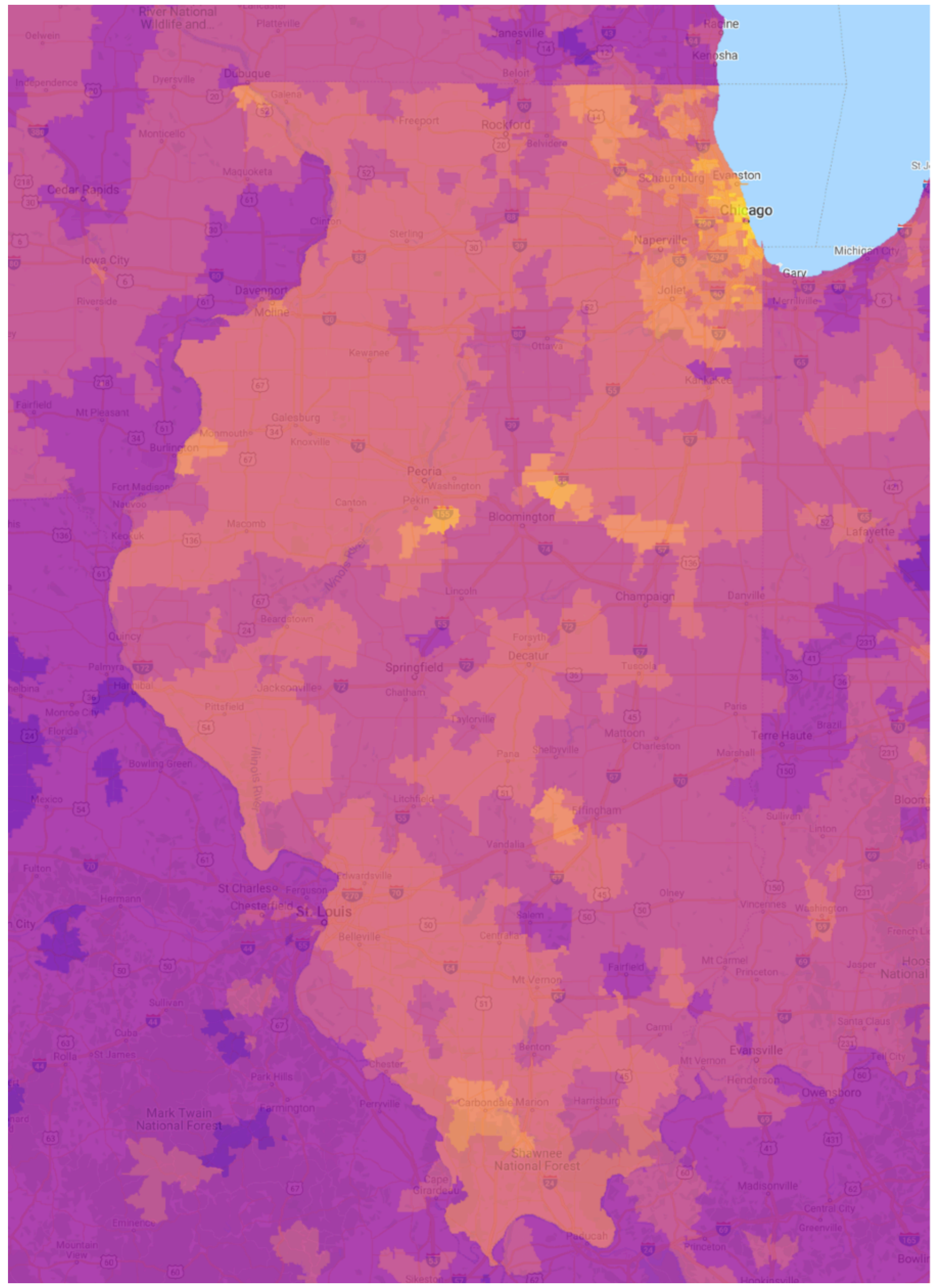
Graph Summarization





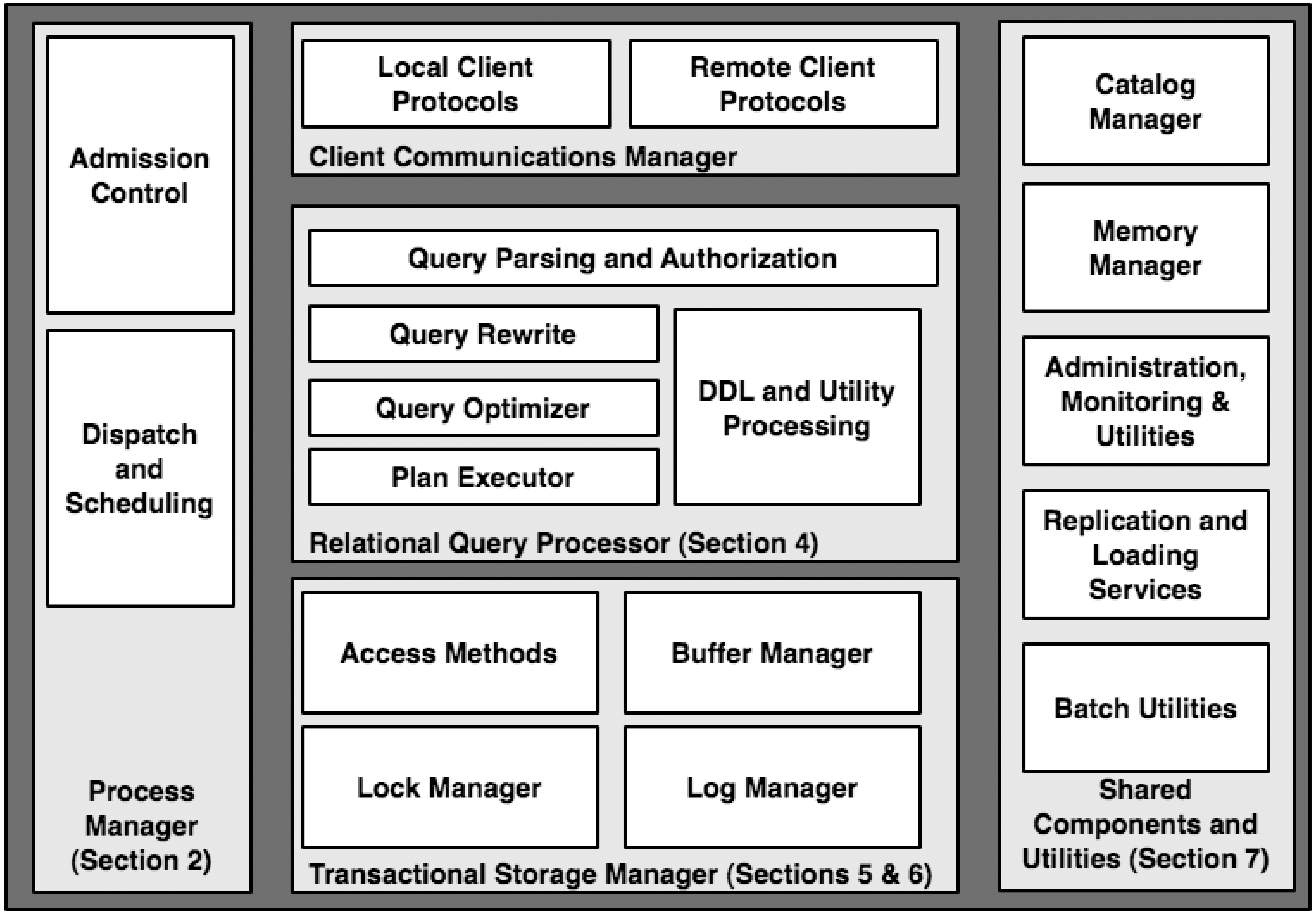
Trajectory Summarization

Map Summarization



Scalable Database Systems

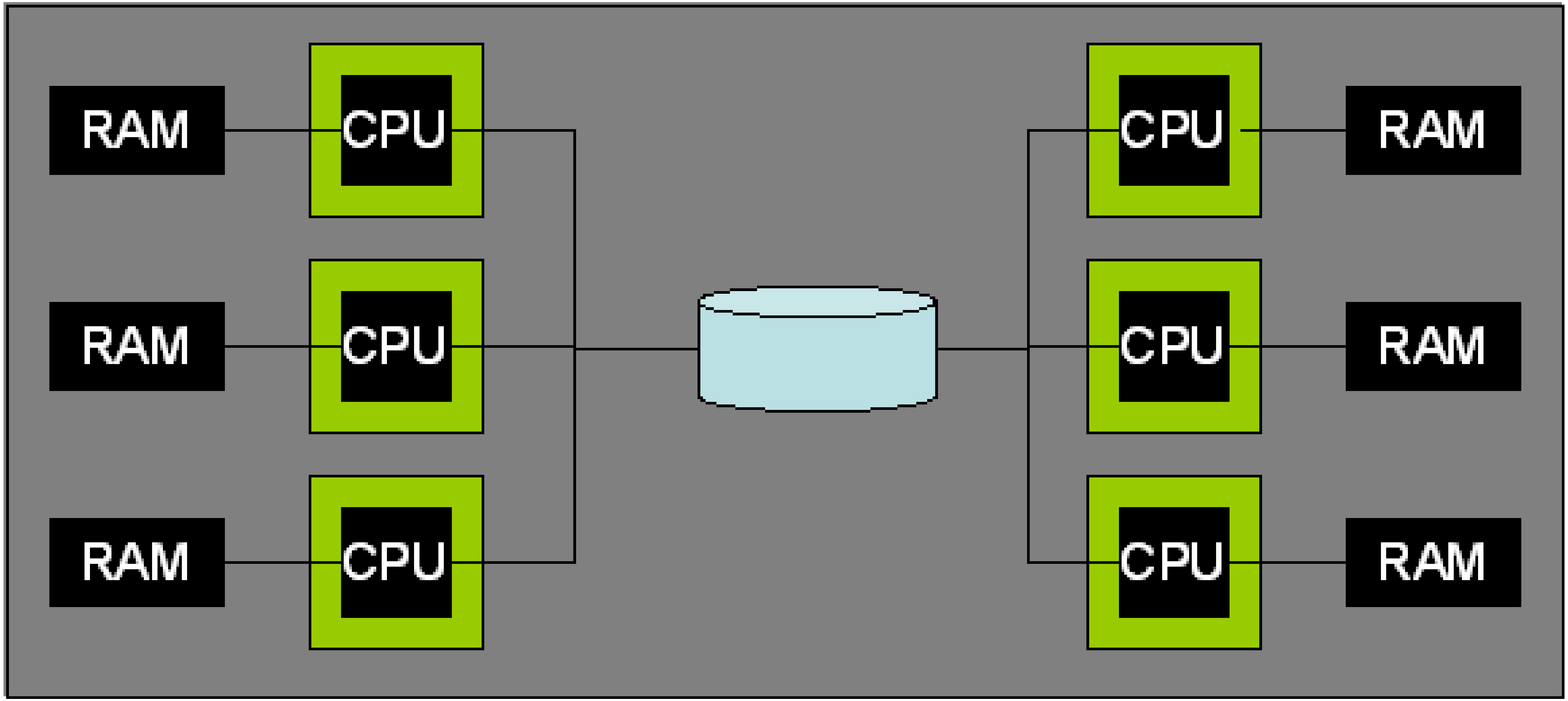
Relational Database Architecture



[Hellerstein et al., Architecture of a Database System]

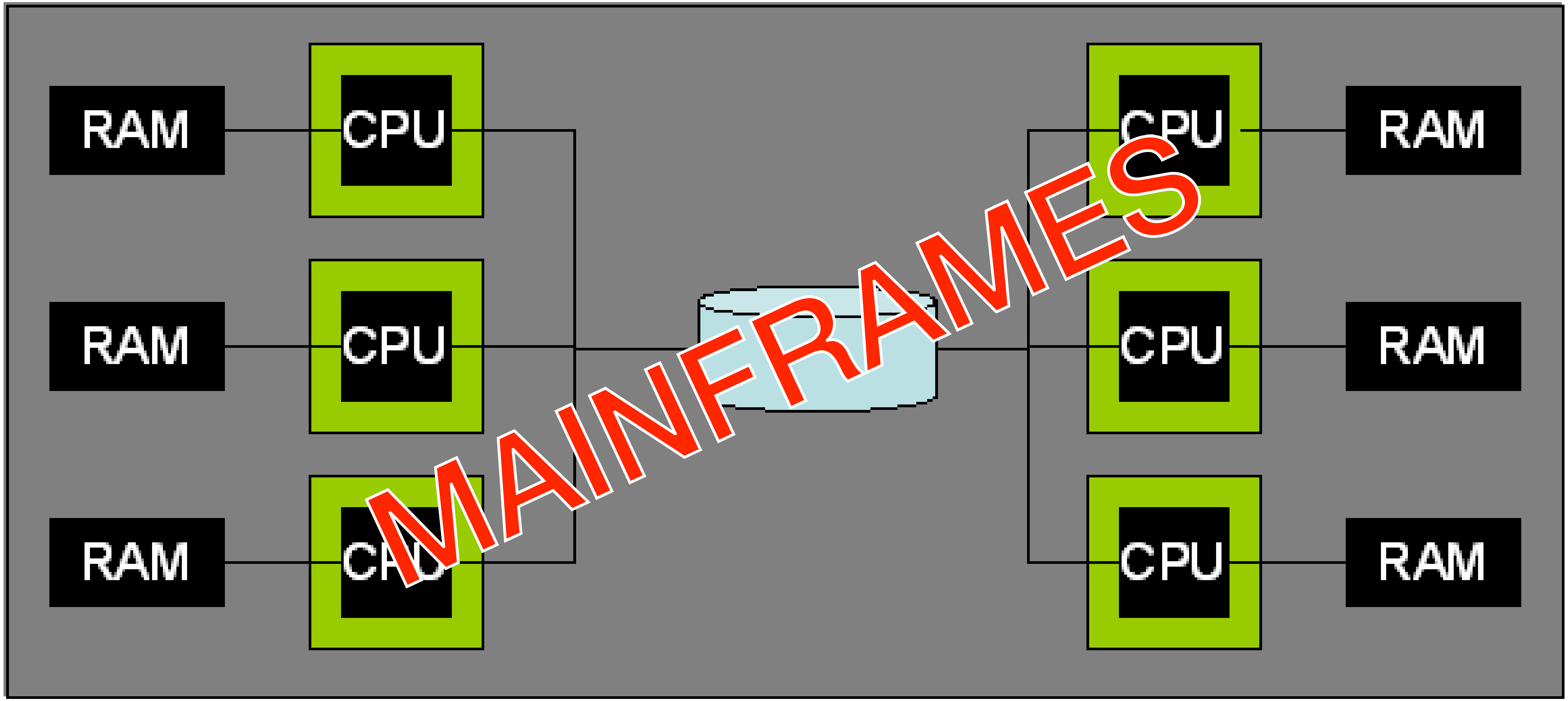
How to Scale Relational Databases?

Parallel DB Architecture: Shared Disk



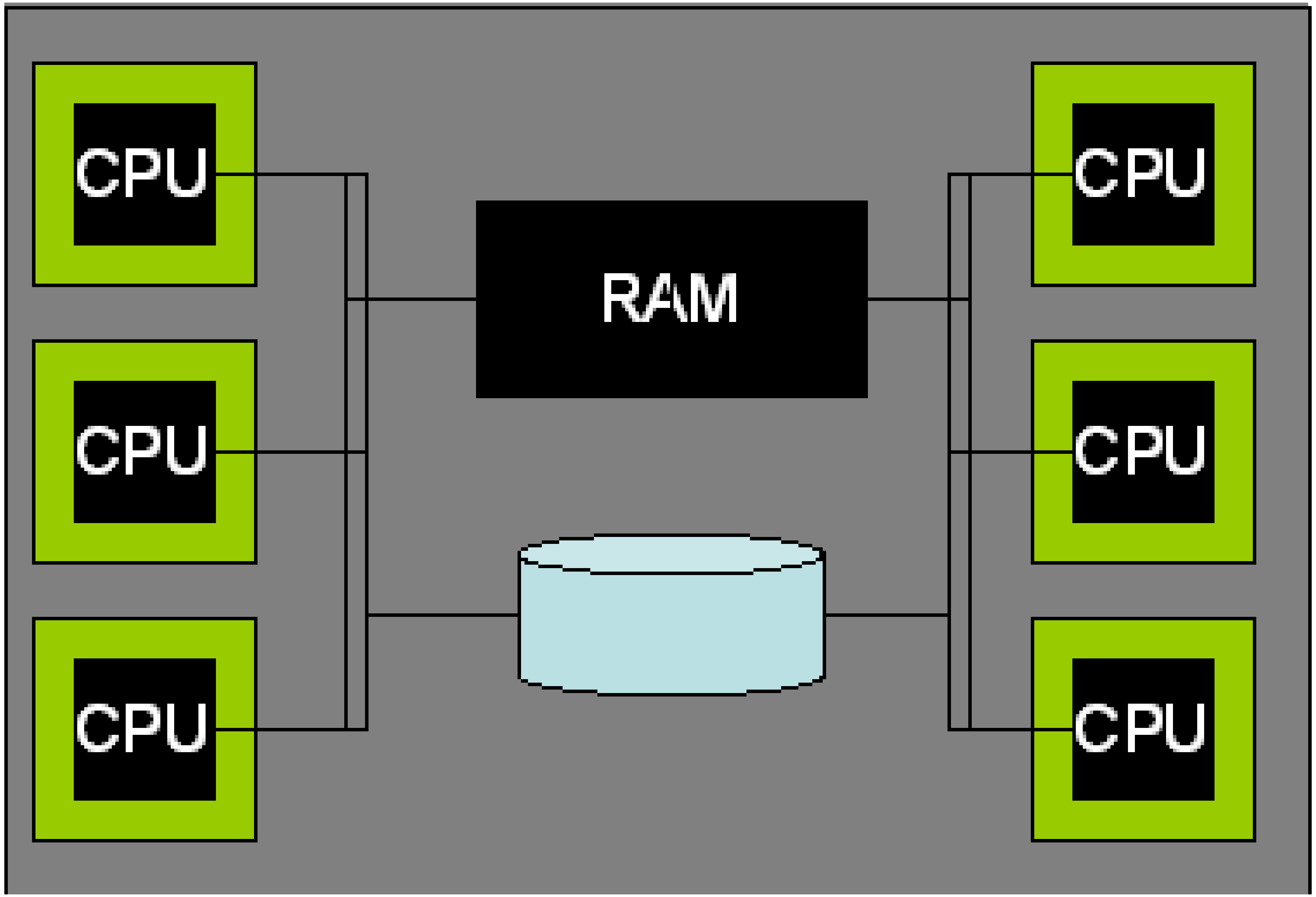
[Hellerstein et al., Architecture of a Database System]

Parallel DB Architecture: Shared Disk



[Hellerstein et al., Architecture of a Database System]

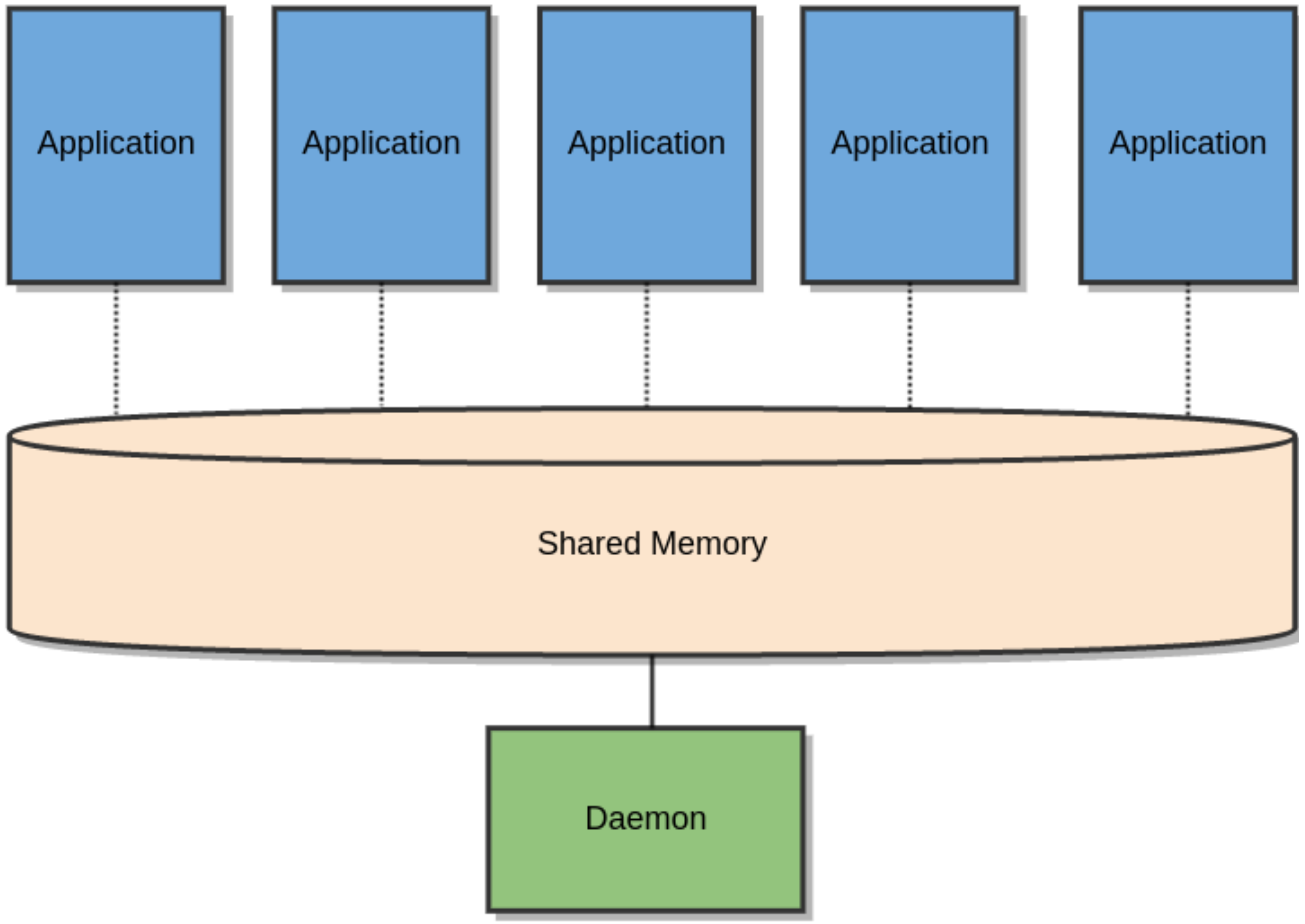
Parallel DB Architecture: Shared Memory



[Hellerstein et al., Architecture of a Database System]

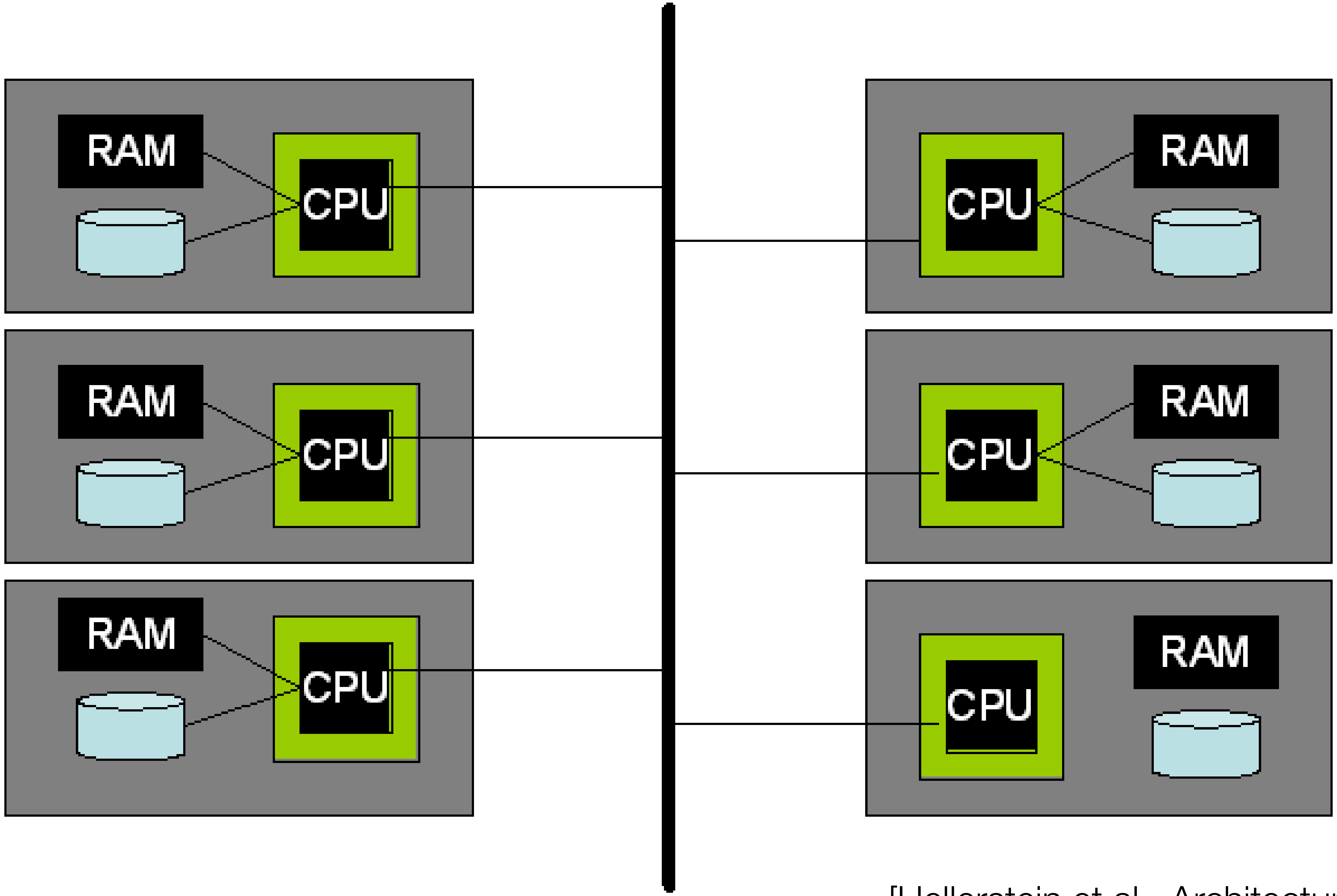
TrafficDB: Shared-Memory Data Store

- Traffic-aware route planning
- Want up-to-date data for all
- Thousands of requests per second
  - High-Frequency Reads
  - Low-Frequency Writes
- "Data must be stored in a region of RAM that can be shared and efficiently accessed by *several* different application processes"



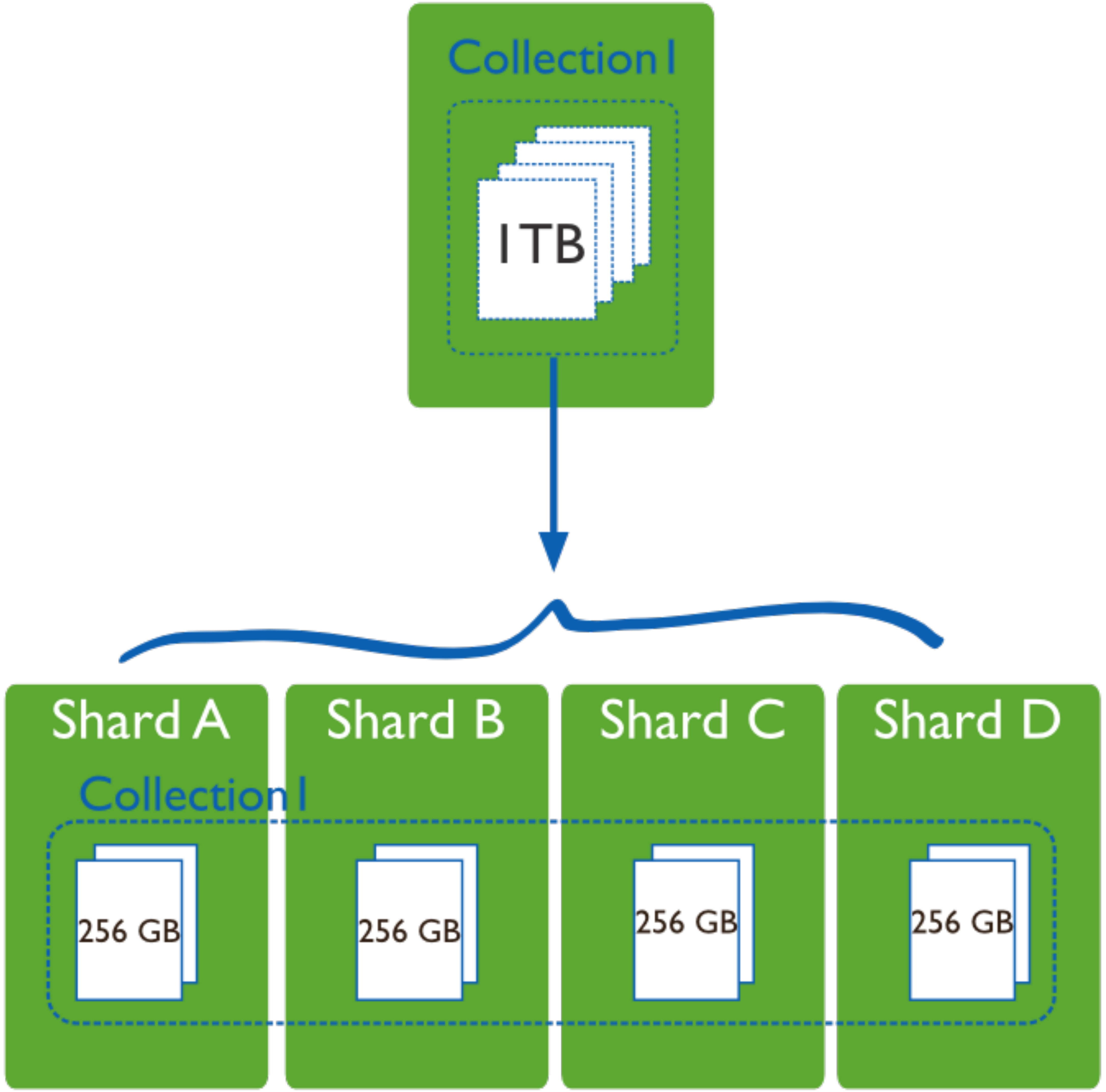
[R. Fernandes et al., 2016]

Parallel DB Architecture: Shared Nothing



[Hellerstein et al., Architecture of a Database System]

Sharding



[MongoDB]

Relational Databases: One size fits all?

- Lots of work goes into relational database development:
  - B-trees
  - Cost-based query optimizers
  - ACID (Atomicity, Consistency, Isolation, Durability)
- Vendors have stuck with this model since the 1980s
- Having different systems leads to business problems:
  - cost problem
  - compatibility problem
  - sales problem
  - marketing problem

[Stonebraker and Çetinetmel, 2005]

ACID Transactions

- Make sure that transactions are processed reliably
- **A**tomicity: leave the database as is if some part of the transaction fails (e.g. don't add/remove only part of the data) using rollbacks
- **C**onsistency: database moves from one valid state to another
- **I**solation: concurrent execution matches serial execution
- **D**urability: endure hardware failures, make sure changes hit disk

Stonebraker: The End of an Architectural Era

- "RDBMSs were designed for the business data processing market, which is their sweet spot"
- "They can be beaten handily in most any other market of significant enough size to warrant the investment in a specialized engine"
- Changes in markets (science), necessary features (scalability), and technology (amount of memory)
- RDBMS Overhead: Logging, Latching, and Locking
- Relational model is not necessarily the answer
- SQL is not necessarily the answer

Row Stores

Primary Key

Row

id	scientist	death_by	movie_name
1	Reinhardt	Crew	The Black Hole
2	Tyrell	Roy Batty	Blade Runner
3	Hammond	Dinosaur	Jurassic Park
4	Soong	Lore	Star Trek: TNG
5	Morbius	The machine	Forbidden Planet
6	Dyson	SWAT	Terminator 2: Judgment Day

[J. Swanhart, Introduction to Column Stores]

OLTP vs. OLAP

- Online Transactional Processing (OLTP) often used in business applications, data entry and retrieval transactions
- OLTP Examples:
  - Add customer's shopping cart to the database of orders
  - Find me all information about John Hammond's death
- OLTP is focused on the day-to-day operations while Online Analytical Processing (OLAP) is focused on analyzing that data for trends, etc.
- OLAP Examples:
  - Find the average amount spent by each customer
  - Find which year had the most movies with scientists dying

Inefficiency in Row Stores for OLAP

select sum(metric) as the_sum from fact

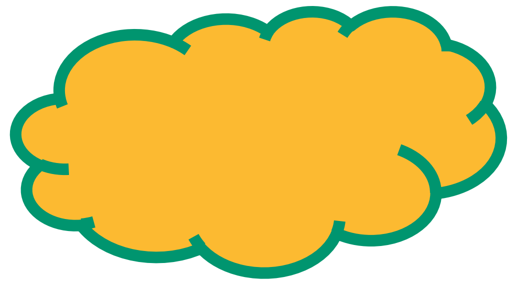
1. Storage engine gets *a whole row* from the table

6	15	on_hold	247	122	9	72	76	5	66

2. SQL interface extracts only requested portion, adds it to "the_sum"

247

3. IF all rows scanned, send results to client, else GOTO 1



[J. Swanhart, Introduction to Column Stores]

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]

Horizontal Partitioning vs. Vertical Partitioning

Original Table

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
1	TAEKO	OHNUKI	BLUE
2	O.V.	WRIGHT	GREEN
3	SELDA	BAĞCAN	PURPLE
4	JIM	PEPPER	AUBERGINE

[M. Drake]

Horizontal Partitioning vs. Vertical Partitioning

Vertical Partitions

VP1

CUSTOMER ID	FIRST NAME	LAST NAME
1	TAEKO	OHNUKI
2	O.V.	WRIGHT
3	SELDA	BAĞCAN
4	JIM	PEPPER

VP2

CUSTOMER ID	FAVORITE COLOR
1	BLUE
2	GREEN
3	PURPLE
4	AUBERGINE

Original Table

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
1	TAEKO	OHNUKI	BLUE
2	O.V.	WRIGHT	GREEN
3	SELDA	BAĞCAN	PURPLE
4	JIM	PEPPER	AUBERGINE

Horizontal Partitions

HP1

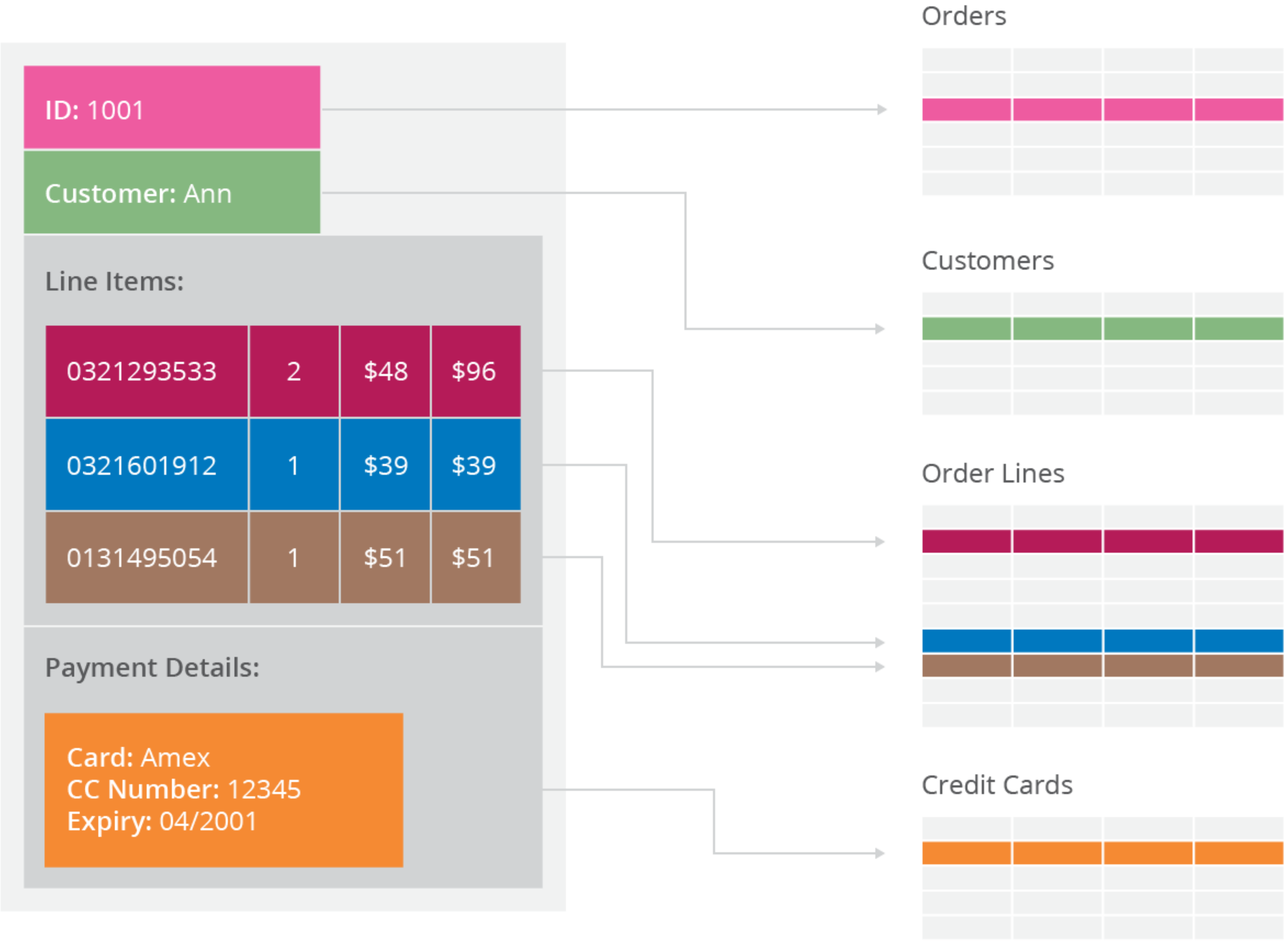
CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
1	TAEKO	OHNUKI	BLUE
2	O.V.	WRIGHT	GREEN

HP2

CUSTOMER ID	FIRST NAME	LAST NAME	FAVORITE COLOR
3	SELDA	BAĞCAN	PURPLE
4	JIM	PEPPER	AUBERGINE

[M. Drake]

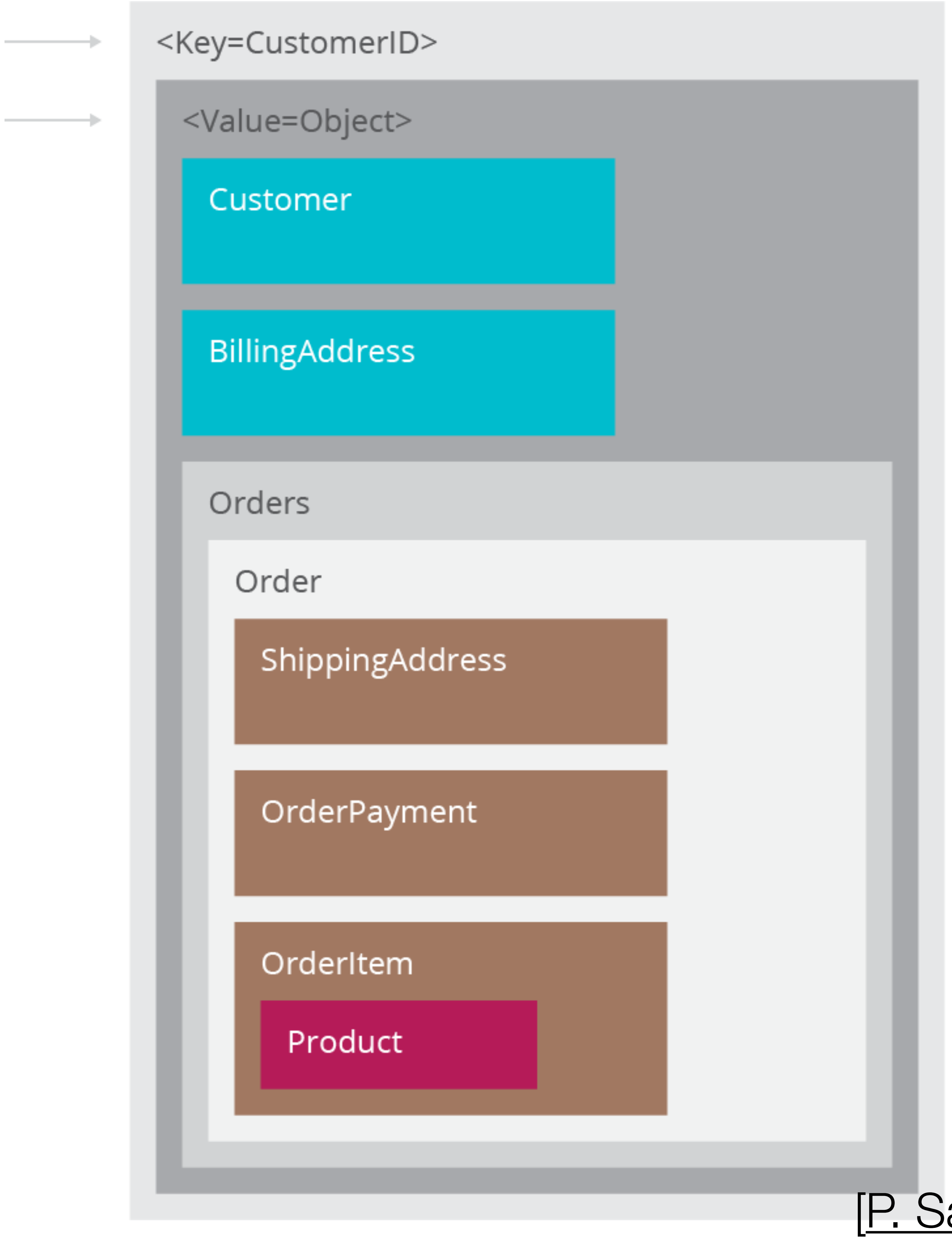
Problems with Relational Databases



[P. Sadalage]

NoSQL: Key-Value Databases

- Always use primary-key access
- Operations:
  - Get/put value for key
  - Delete key
- Examples
  - Memcached
  - Amazon DynamoDB
  - Project Voldemort
  - Couchbase



[P. Sadalage]

NoSQL: Document Databases

- Documents are the main entity
  - Self-describing
  - Hierarchical
  - Do not have to be the same
- Could be XML, JSON, etc.
- Key-value stores where values are "examinable"
- Can have query language and indices overlaid
- Examples: MongoDB, CouchDB, Terrastore

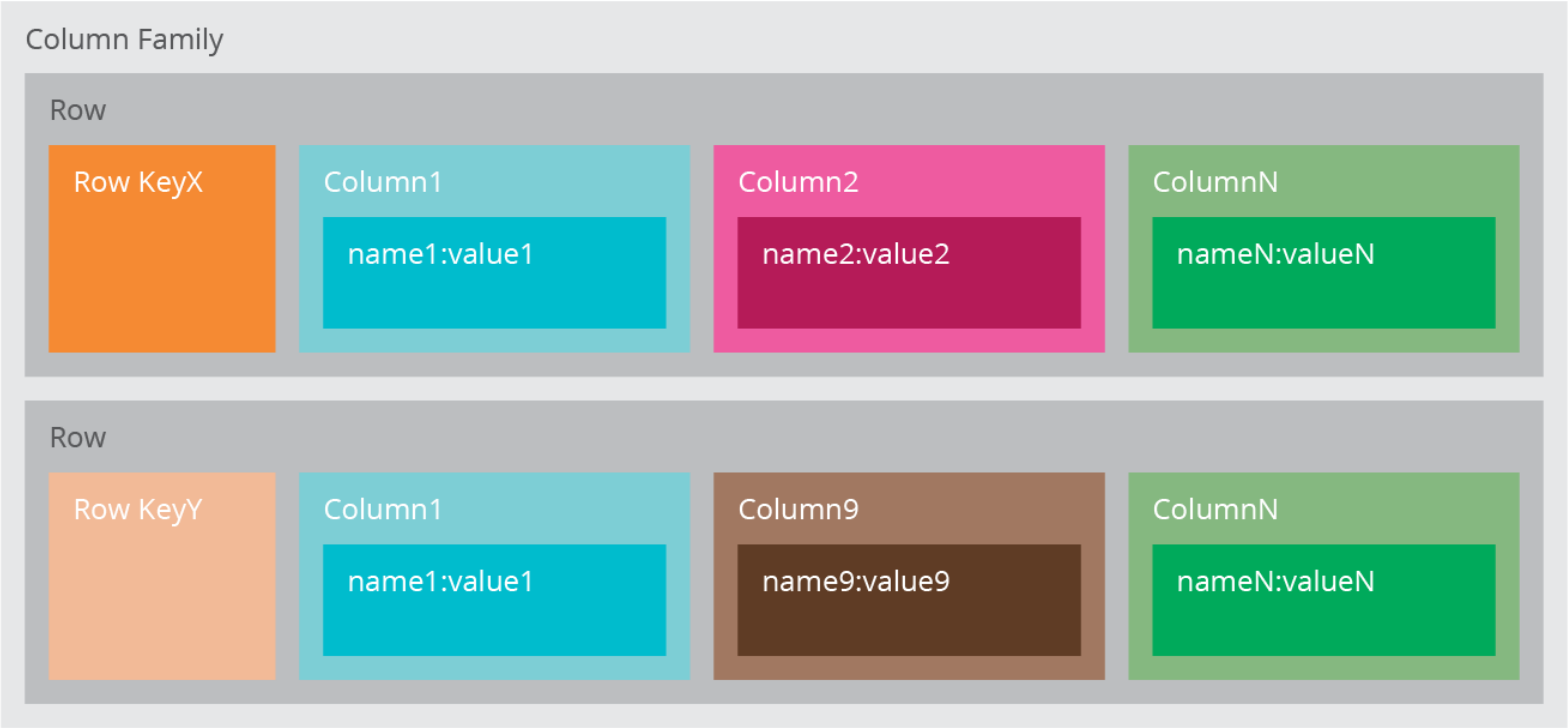
<Key=CustomerID>

```
{
    “customerid”: “fc986e48ca6”
    “customer”:
    {
    “firstname”: “Pramod”,
    “lastname”: “Sadalage”,
    “company”: “ThoughtWorks”,
    “likes”: [ “Biking”,”Photography” ]
    }
    “billingaddress”:
    { “state”: “AK”,
        “city”: “DILLINGHAM”,
        “type”: “R”
      }
}
```

[P. Sadalage]

NoSQL: Column Stores

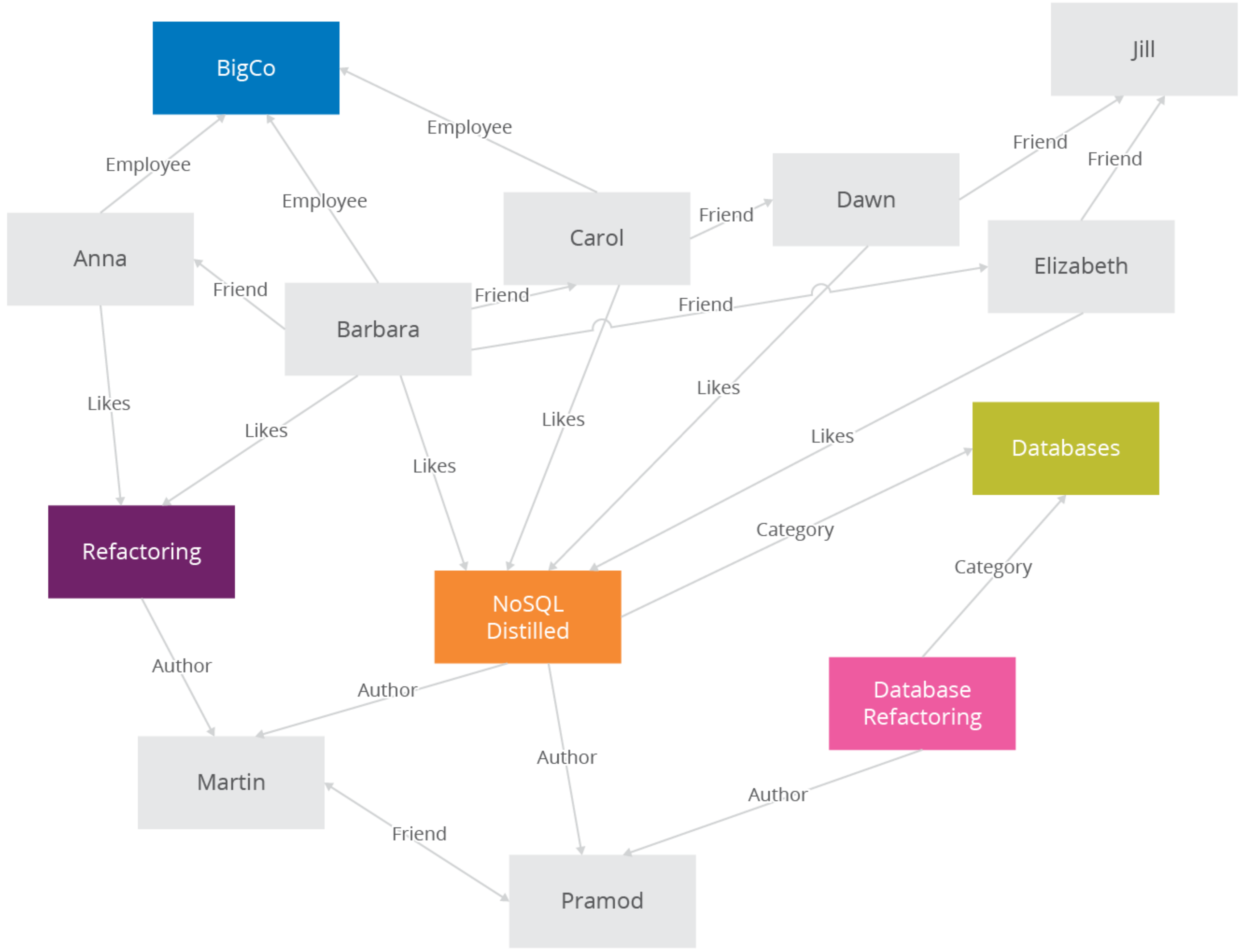
- Instead of having rows grouped/sharded, we group columns
- …or families of columns
- Put similar columns together
- Examples: Cassandra, HBase



[P. Sadalage]

NoSQL: Graph Databases

- Focus on entities and relationships
- Edges may have properties
- Relational databases required a set traversal
- Traversals in Graph DBs are faster
- Examples:
  - Neo4j
  - Pregel



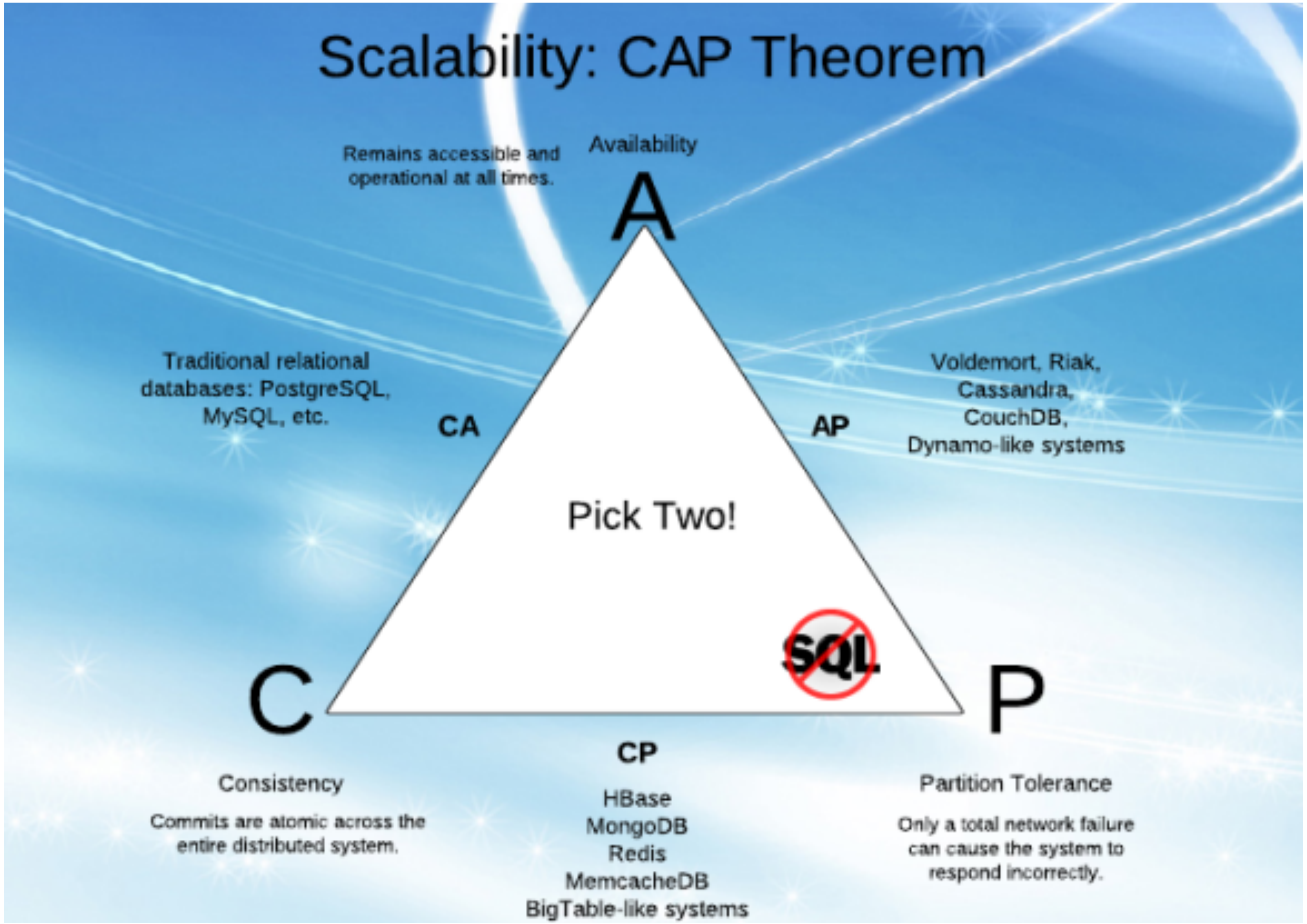
[P. Sadalage]

Distributing Data

- Aggregate-oriented databases
- Sharding (horizontal partitioning): Sharding distributes different data across multiple servers, so each server acts as the single source for a subset of data
- Replication: Replication copies data across multiple servers, so each bit of data can be found in multiple places. Replication comes in two forms,
    - Master-slave replication makes one node the authoritative copy that handles writes while slaves synchronize with the master and may handle reads.
    - Peer-to-peer replication allows writes to any node; the nodes coordinate to synchronize their copies of the data.

[P. Sadalage]

CAP Theorem



[E. Brewer]

CAP Theorem

- Consistency: every read would get you the most recent write
- Availability: every node (if not failed) always executes queries
- Partition tolerance: system continues to work even if nodes are down
- Theorem (Brewer): It is impossible for a distributed data store to simultaneously provide more than two of Consistency, Availability, and Partition Tolerance

Think about RDBMS Transactions…

Cassandra: A Decentralized Structured Storage System

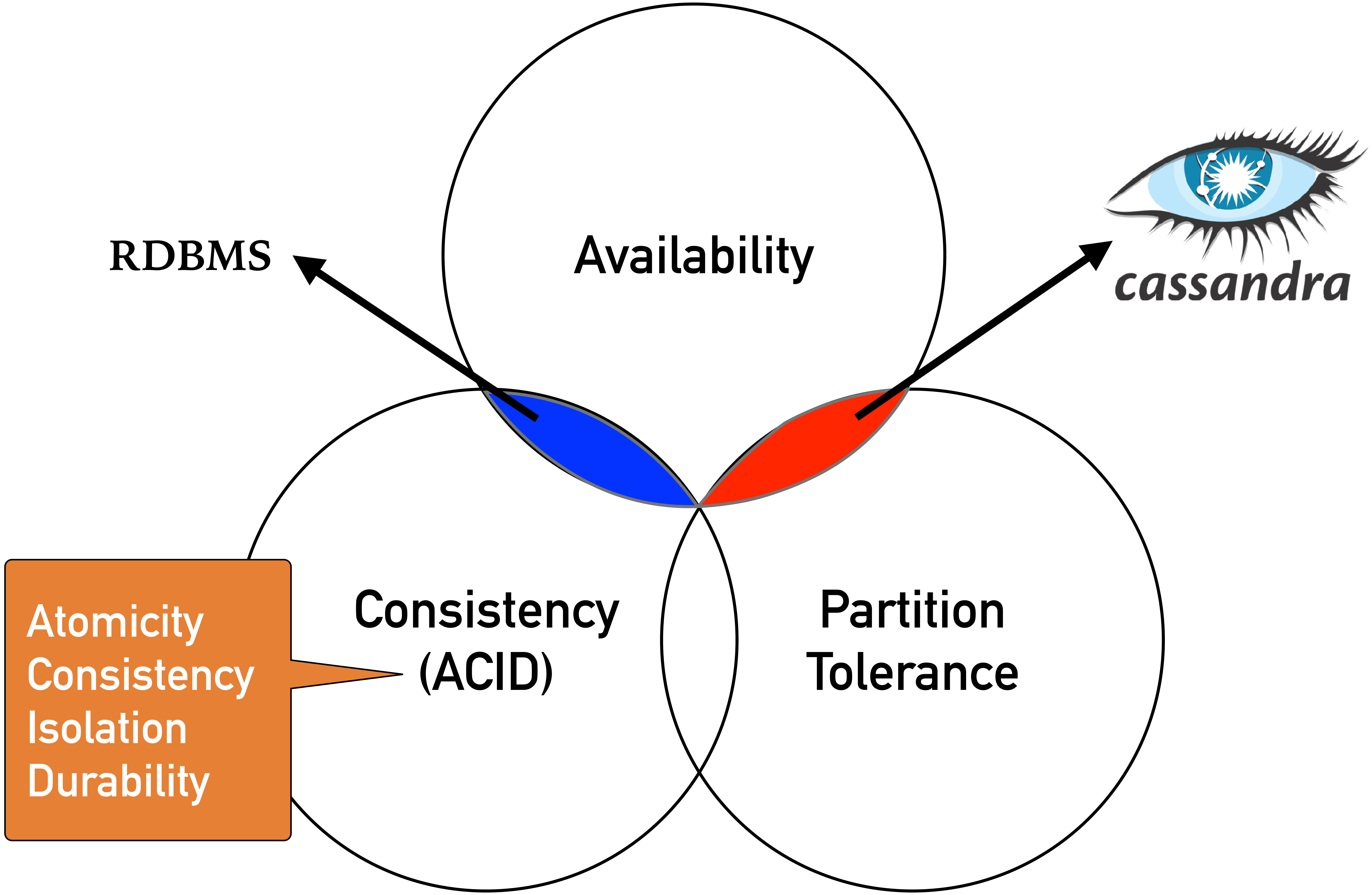
A. Lakshman and P. Malik

What is Cassandra?

- Fast Distributed (Column Family NoSQL) Database
  - High availability
  - Linear Scalability
  - High Performance
- Fault tolerant on Commodity Hardware
- Multi-Data Center Support
- Easy to operate
- Proven: CERN, Netflix, eBay, GitHub, Instagram, Reddit

[G. Atil]

Cassandra and CAP



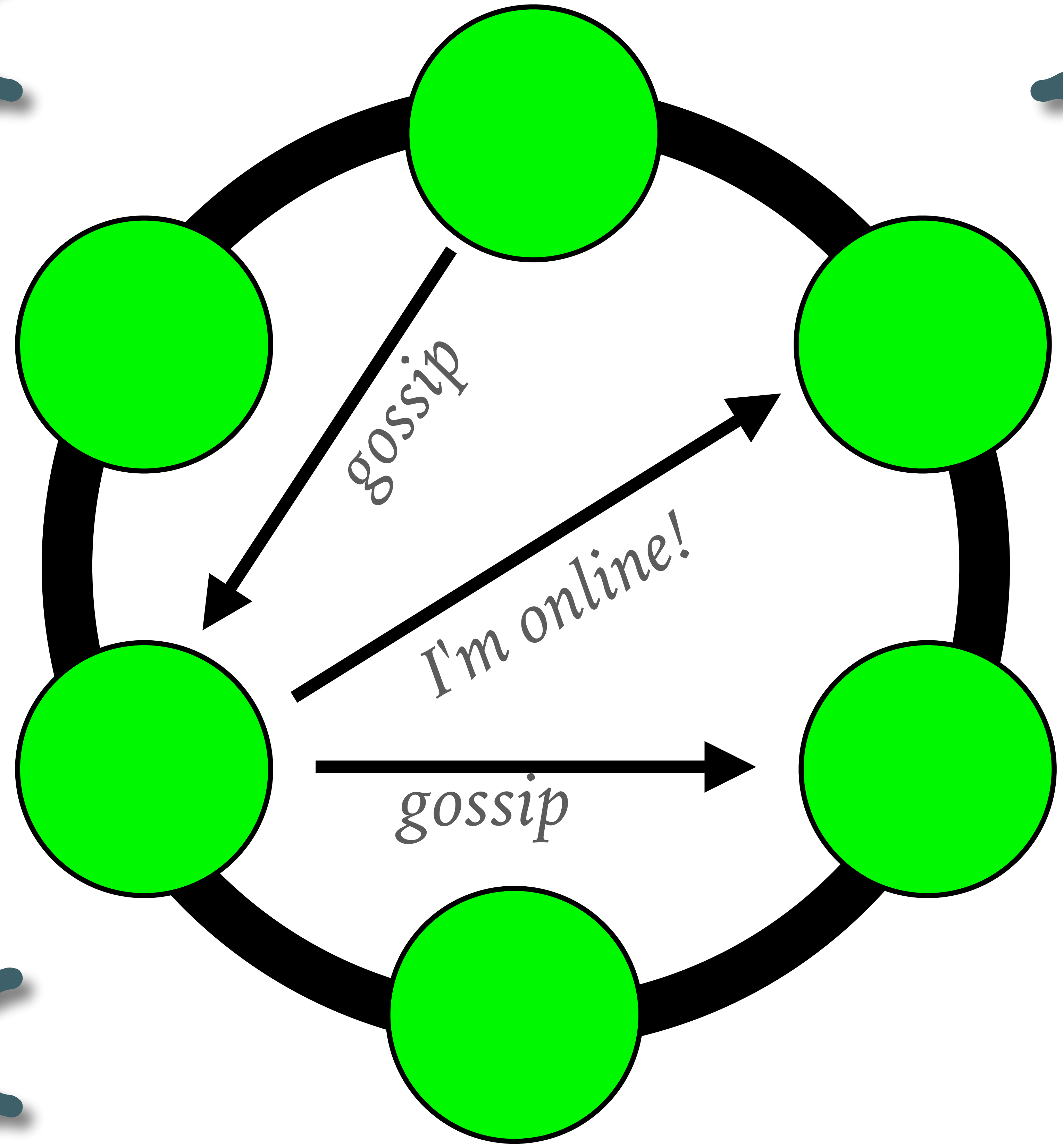
[G. Atil]

Cassandra: Ring for High Availability

NO MASTER

NO SLAVE

PEER TO PEER



[G. Atil]

Slides: Introduction to Cassandra

Robert Stupp

Next Class's Reading & Quiz

- Spanner: Google's Globally-Distributed Database
- Quiz available on Thursday:
  - Will focus on main concepts in the paper, not details
  - On Blackboard