MICRON

MANUFACTURING IMPROVEMENTS THROUGH CUSTOM REALTIME OPTIMIZED NLP

NOT A GAN

PROJIT B – INFORMATION RETRIEVAL EXTRACTION LAB

ALOK D– LANGUAGE TECHNOLOGIES RESEARCH CENTER

UJWAL N – LANGUAGE TECHNOLOGIES RESEARCH CENTER

ATHREYA C – CENTER FOR SECURITY, THEORY, AND ALGORITHMS RESEARCH

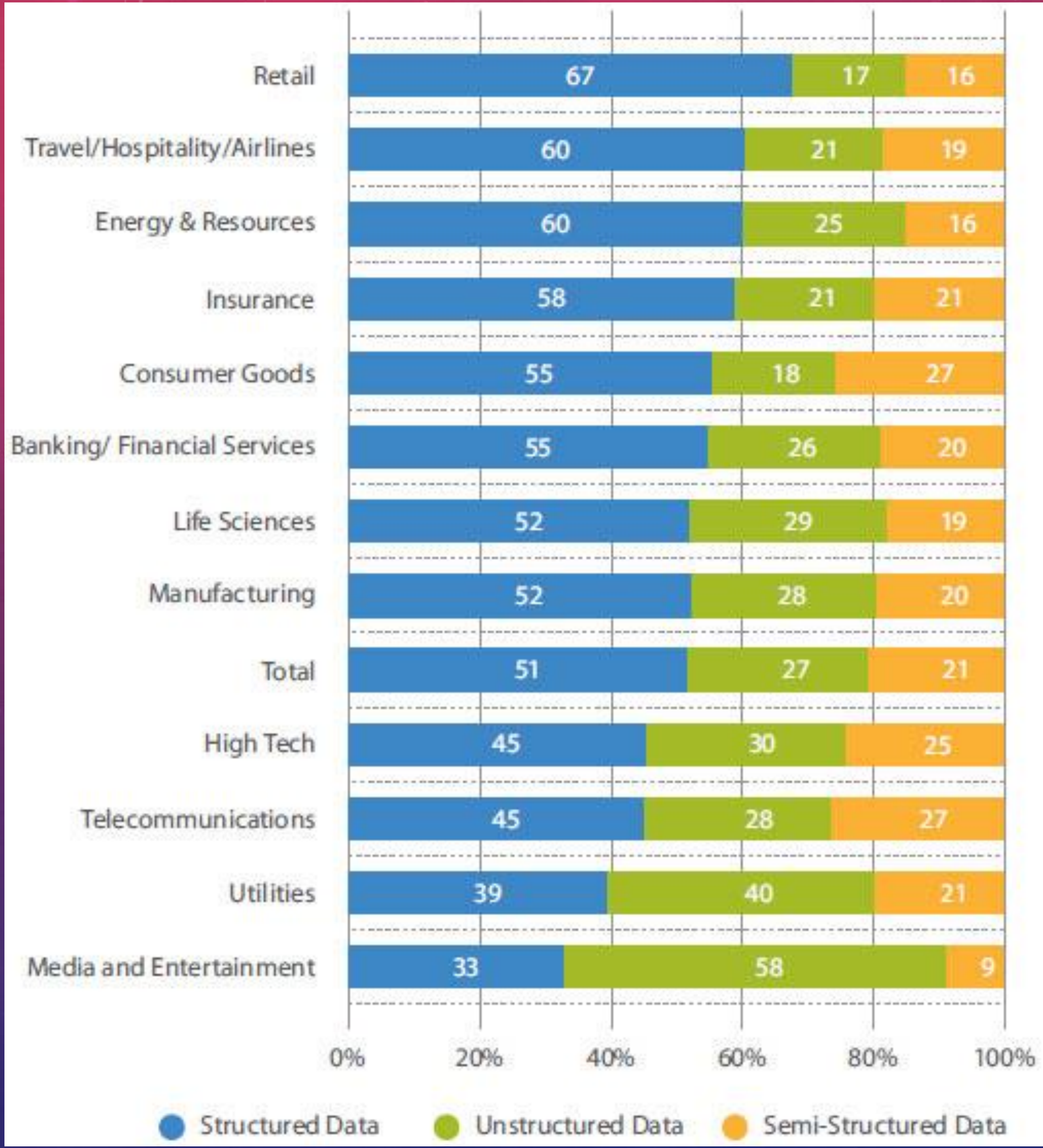
AUTO MANUFACTURING IS THE DREAM

Semi Automated Manufacturing businesses:

- Automation is a dream
- From Automobiles to FMCG such as Fruit and Vegetables, Pulp etc
- Quality Assurance, Customization are Manual Processes
- Manual processes generate manual feedback

MANUFACTURING WITHOUT STRUCTURE (-D TEXT)

Document Flow:	Supply Orders Contracts Compliance Orders **Logs**
Logs:	Machine Generated (Structured) Human Input
Benefits of Logs:	Bottleneck Identification Evaluation of Personnel and Automatons Identification of Failures



Source:
TCS, 2013: Global Trend Study.
https://sites.tcs.com/big-data-study/industries-unstructured-data/

JUST IN TIME FOR THE BEST PART :)

- Used in things like Car Manufacturing
  - JIT == TPS [Toyota Production System]
- The just-in-time (JIT) inventory system is a management strategy that minimizes inventory and increases efficiency
- The success of the JIT production process relies on steady production, high-quality workmanship, no machine breakdowns, and reliable suppliers.

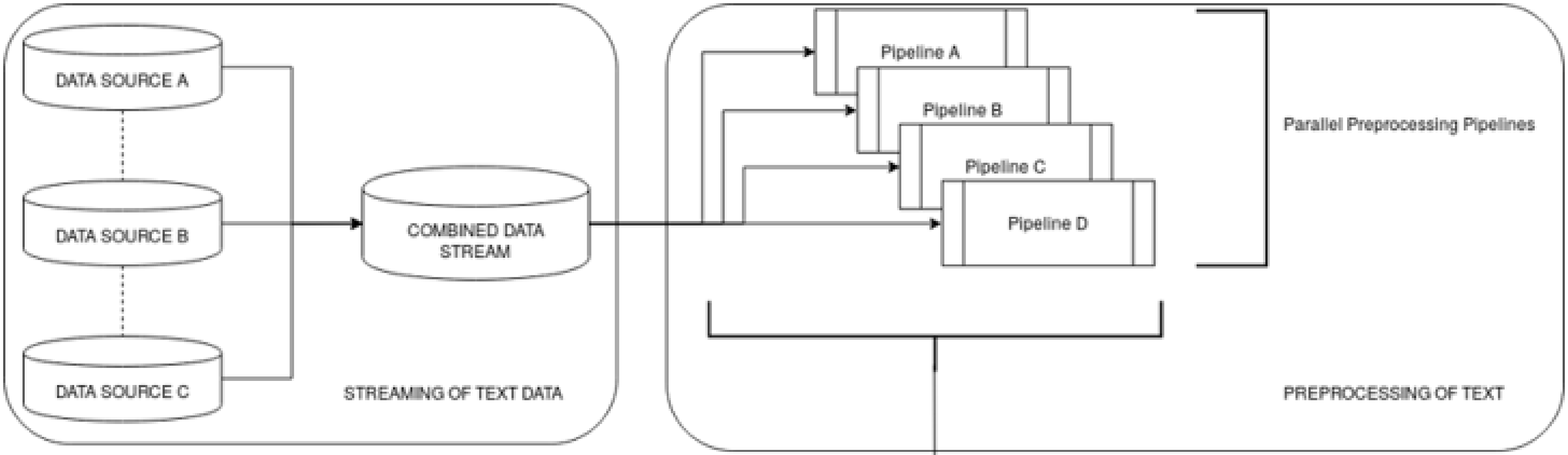
ONLY TIME WILL TELL...

Short Term

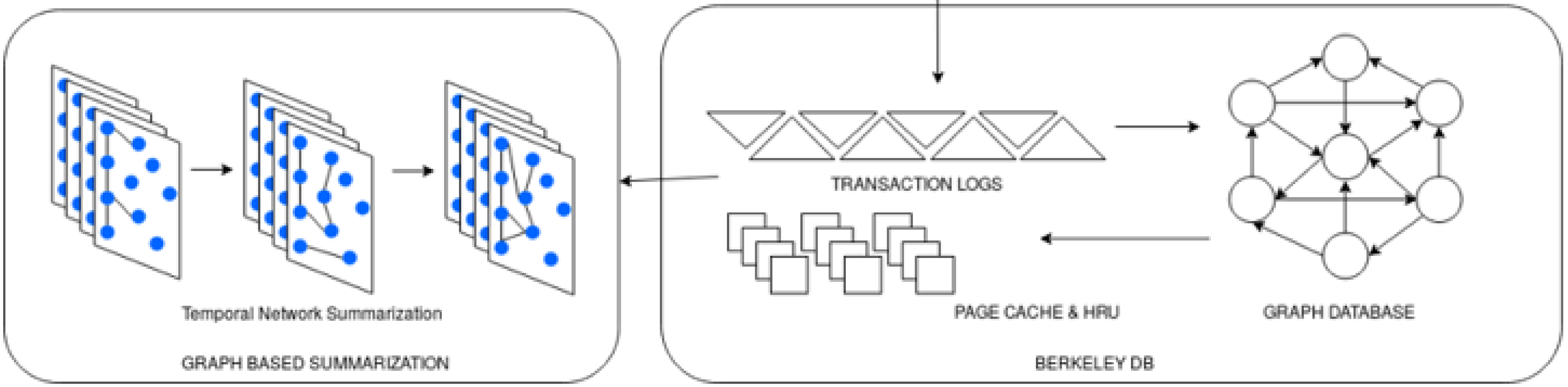
- Decision Makers don't have the time to read all the information
- Multiple parties may be affected by multiple things.
  - Supply Manager would want to know delays in supplies
  - QA wants to know of potential product defects observed
  - Management may want to know of employee issues on the floor

Long Term

- Identify problematic areas in the system over time
- Aggregate analysis
- Association rule mining



WHERE THE ENDS MEET



GRAPHS ARE SUMMARIES, SUMMARIES ARE GRAPHS

https://github.com/AlokDebnath/Megathon19

IEEE Xplore, 2015: Adhikari et al; Propagation based Temporal Network Summarization

h

- Graph based summarization looks at accessible input nodes from which to gather information
- It is a highly optimized summarization method for large amounts of data
- Uses keywords, semantic weights and contextual information to determine the importance of certain parts of the text
- Can be personalized to the user requesting it
- Graph Summaries can be updated based on temporal information

WHAT THE THING MUST DO

- Not just a ML/DL task.
- System building needs to take into account:
    - Relevant information to a particular entity in the system
    - Model connections between entities (could be derived from the dataset)
    - Context to a particular entry if required
    - Updating certain links as required

THE MAP-REDUCE

Mapping Summarization to Pipeline

- Don’t need full set of information
- Easy query links between entities
- Needs to be fast and efficient.

Solution

- HRU

HRU OR ITS VARIANTS

Alternatives like A Reduced Lattice Algorithm exist, but they trade off on the performance guarantee or space

Reference: https://web.eecs.umich.edu/~jag/eecs584/papers/implementing_data_cube.pdf

GREEDY ALGORITHM, THUS MUCH FASTER.

PERFORMANCE GUARANTEE, NO WORSE THAN ¾ OF OPTIMAL SOLUTION.

CAN'T MATERIALIZE ALL DATA, BECAUSE OF PETABYTES OF DATA

PROVIDES EXCELLENT TRADEOFF BETWEEN RUN TIME AND SPACE COMPLEXITY

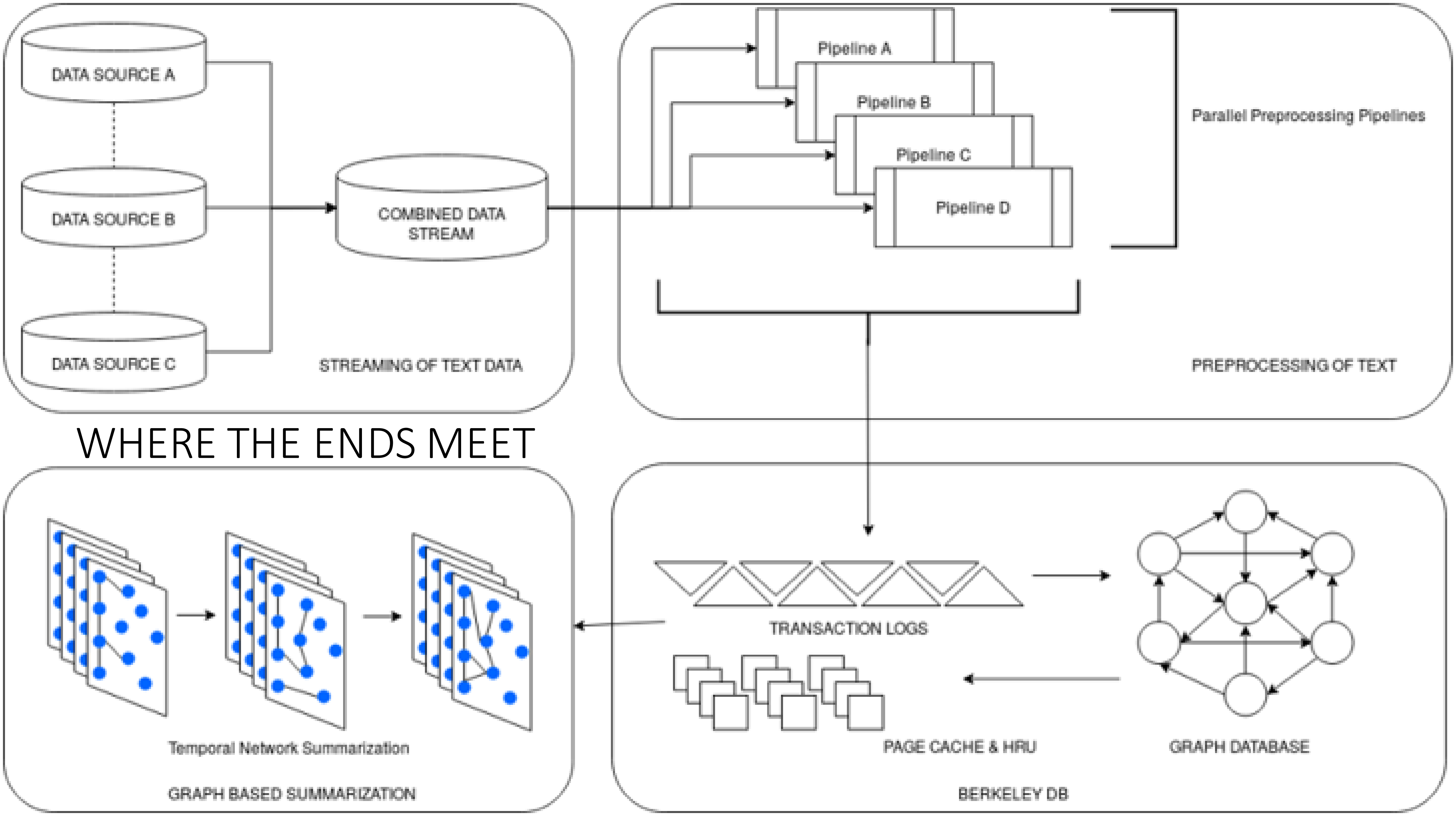
WHY BERKELEYDB?

Implementation	DB Access(s)	Query Time(s)
CouchDB	955.025	956.606
MongoDB	811.560	846.286
BerkeleyDB	0.515	117.759

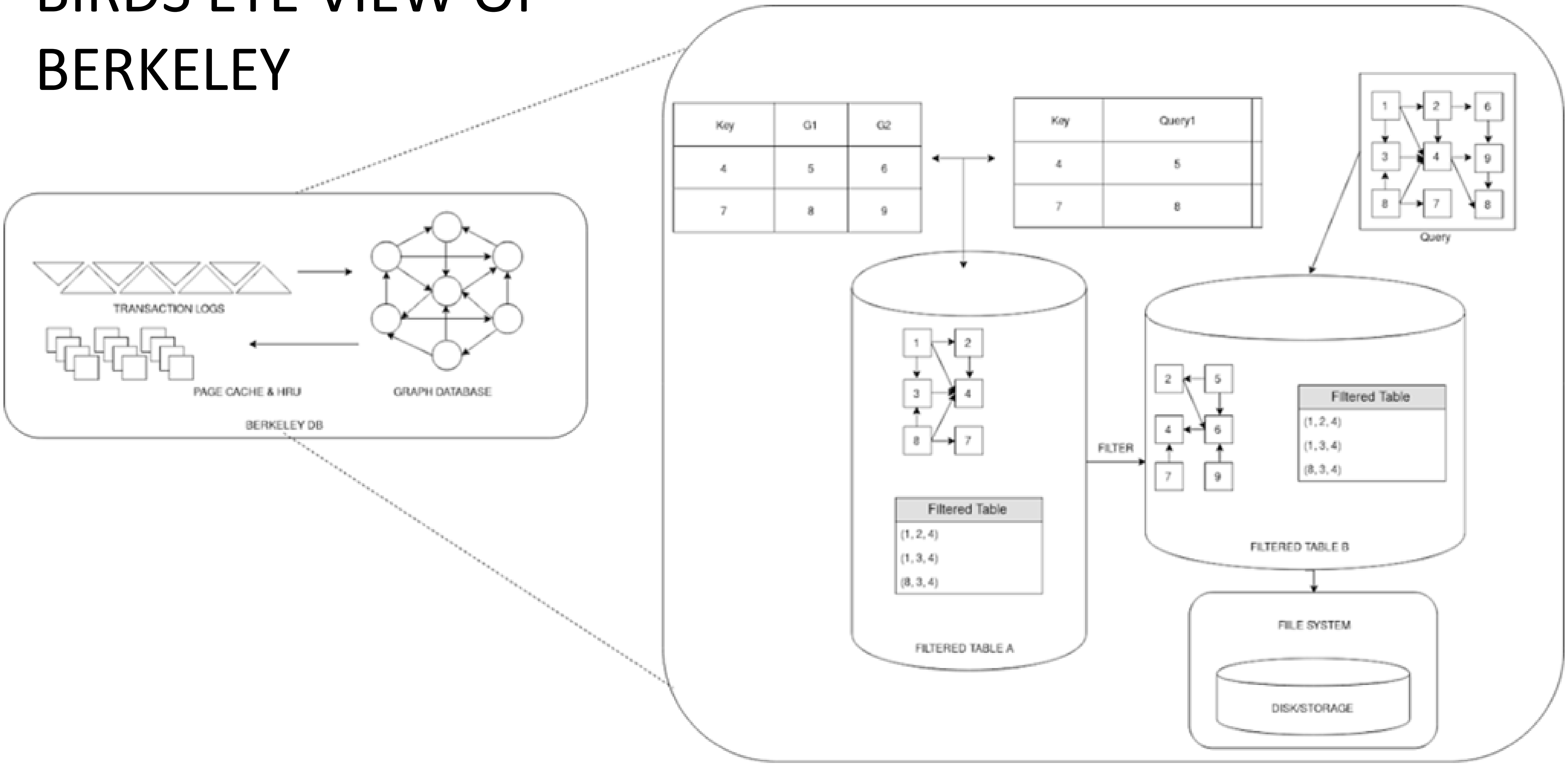
- High Performance
- Embedded
- Distributed
- NoSQL

Source:
https://therin.me/uploads/calc-apriori-realtime-nosql_tcirwin.pdf

WHERE THE ENDS MEET

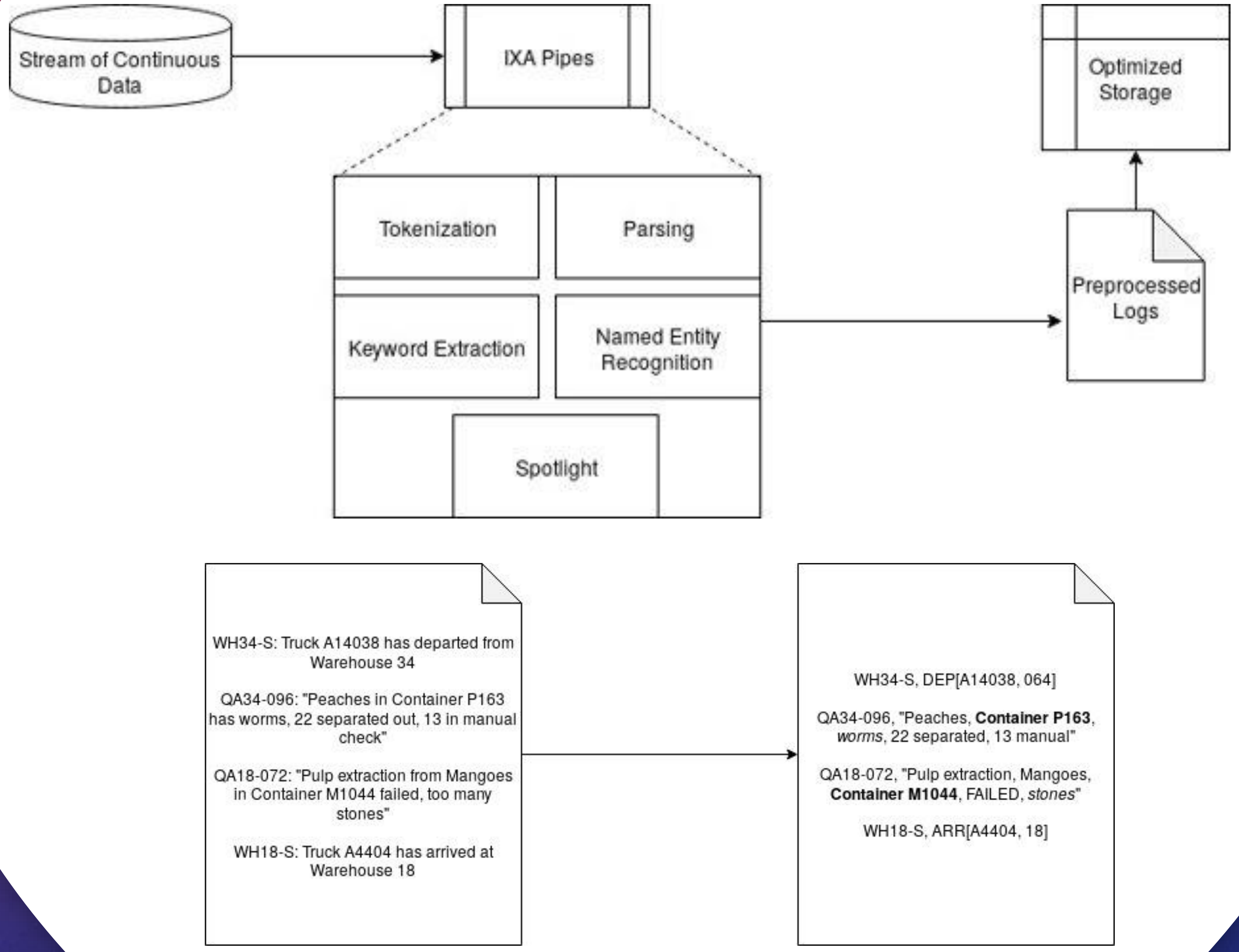


BIRDS EYE VIEW OF BERKELEY

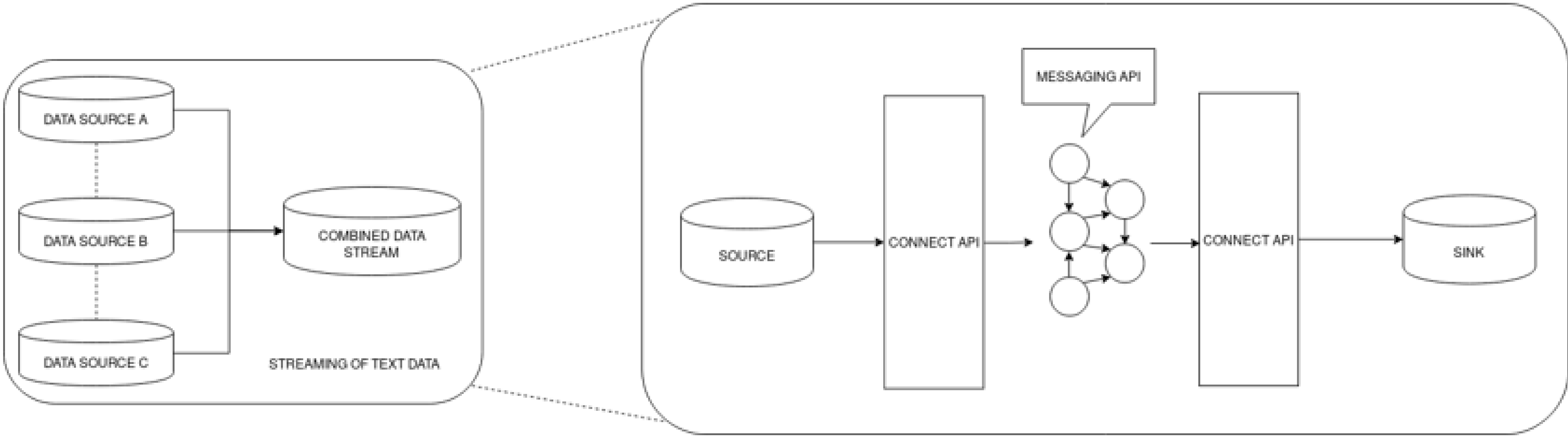


WHAT COMES BEFORE PROCESSING (SPOILER ALERT… PREPROCESSING)

Dynamic
Fast
Inherently parallelizable



CONTEMPLATING KAFKA



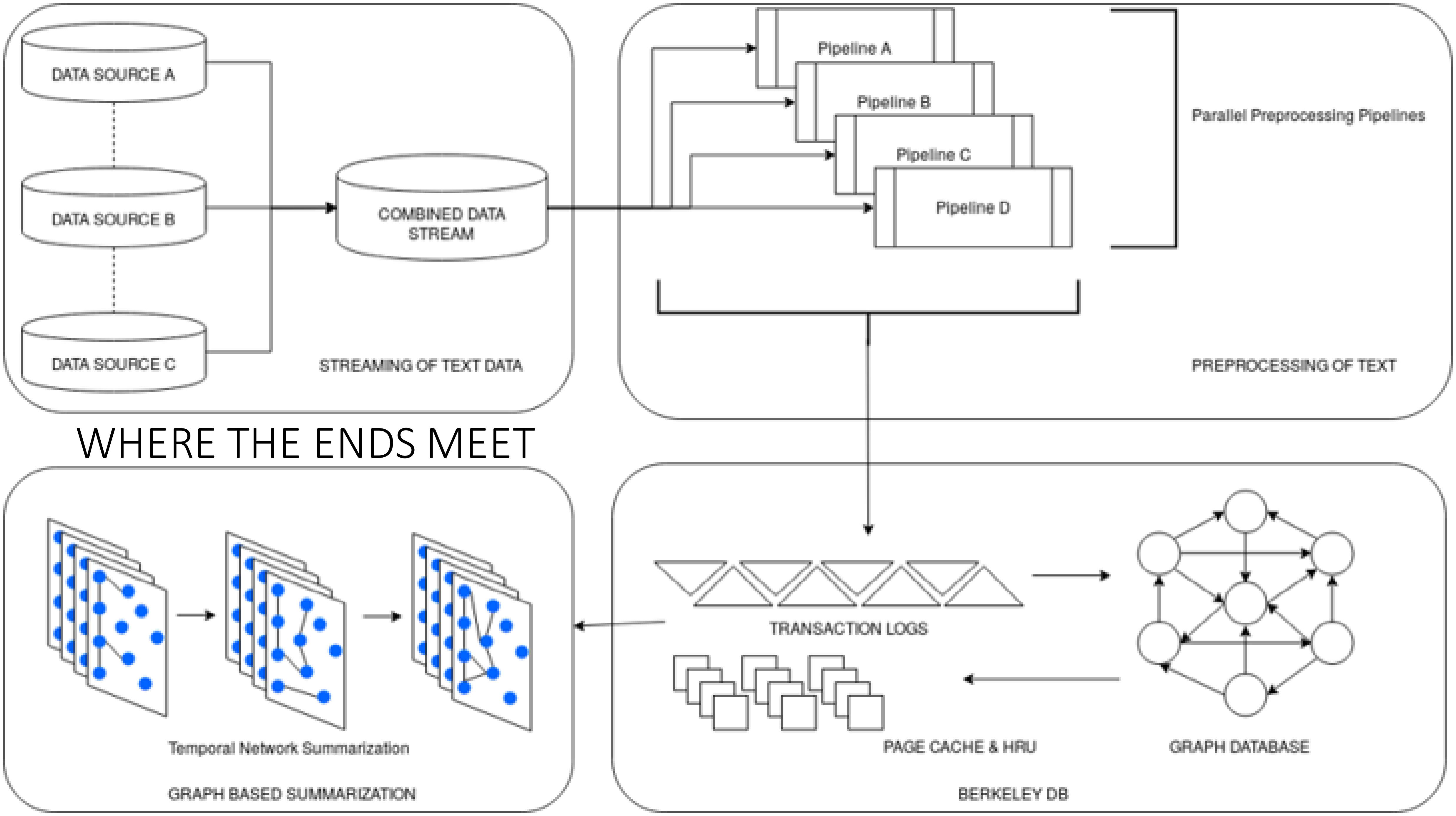
THE TRIAL OF KAFKA

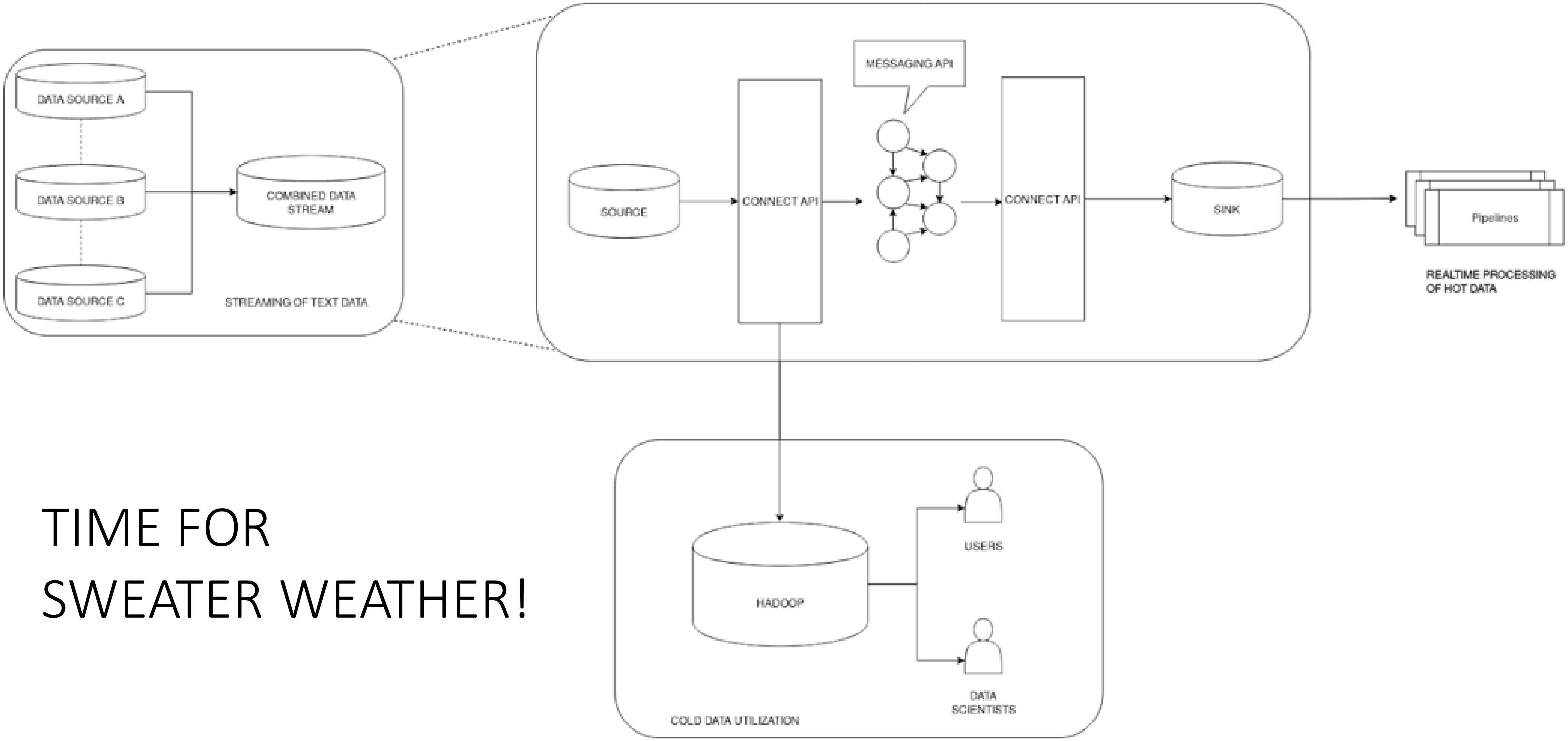
- Kafka – Distributed streaming for logs.
- Why Stream?
  - Modular
  - Uniform interface to multiple data processing pipelines.
- Per record stream processing with millisecond latency and over 50,000 messages/s
- Elastic, highly scalable, fault tolerant
- Equally viable for small, medium and large-scale clusters

Kafka lets you: Connect the input data stream to various pipelines by plugging in various stream consumers

Source: Kreps, Jay, Neha Narkhede, and Jun Rao. "Kafka: A distributed messaging system for log processing." Proceedings of the NetDB. 2011.

WHERE THE ENDS MEET

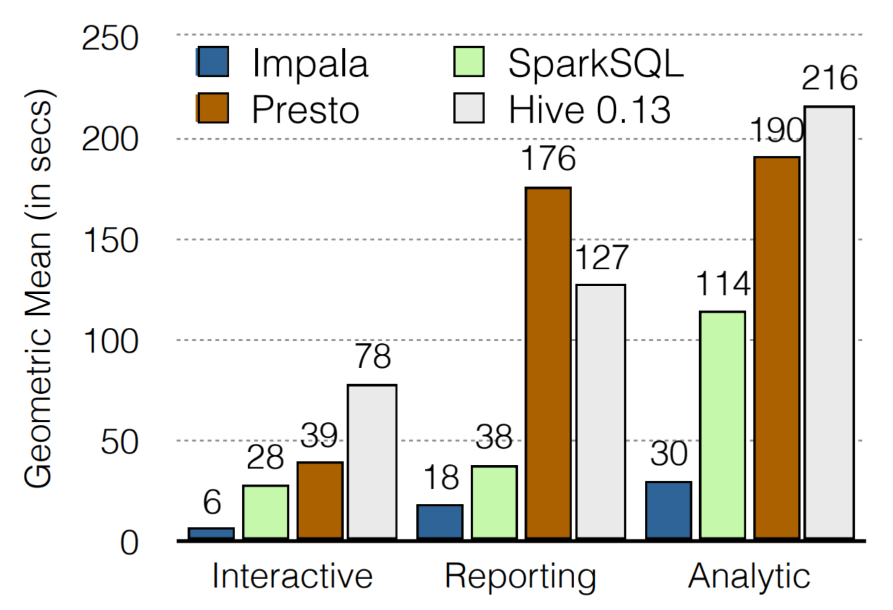


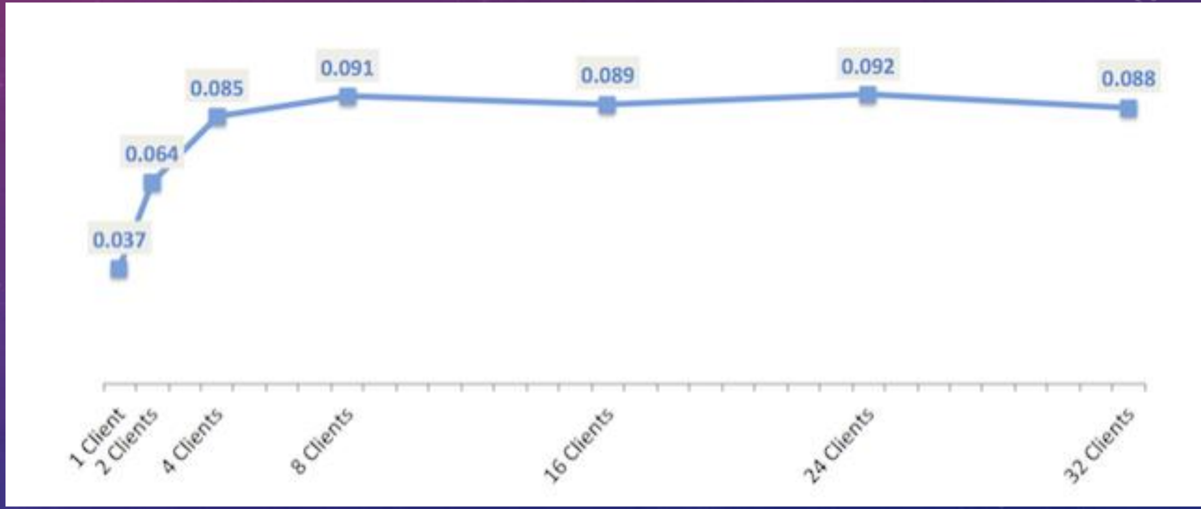


TIME FOR SWEATER WEATHER!

TAMING THE IMPALA

- Impala + Parquet on Hadoop.
- Parquet: I/O will be reduced as we can efficiently scan only a subset of the columns while reading the data. Better compression also reduces the bandwidth required to read the input.
- Impala: Massively Parallel Processing has none of the overheads associated with MapReduce. Up to a 68x latency improvement for single user. Low latency growth rate over multiple users.





Latency for different query types/multiple clients (lower is better)

Source: Kornacker, Marcel, et al. "Impala: A Modern, Open-Source SQL Engine for Hadoop." Cidr. Vol. 1. 2015.

TIL

- Almost **50%** of data in Manufacturing is semi-structured and unstructured
- Manual **text** logs and records are an irreplaceable part of Semi-Automated Manufacturing Industries
- Unstructured text captures **human insight** into the manufacturing process
- **NLP is hard**: Not trivial to extract relevant information but adds significant value in
- Semi-Automated logs can follow the same pipeline for **personalized real-time** feeds

TIL CONTINUED....(LEARNED A LOT)

- Customized Pipeline is essential and necessary for having the most **efficient handling** of big data
- Take a look at Google, Uber, Facebook.
- Generic solutions and piecing together different modules cannot be as fast as an **integrated, problem-specific pipeline**.

THANK YOU!

TEAM: NOT A GAN

Link to our repo: https://github.com/AlokDebnath/Megathon19