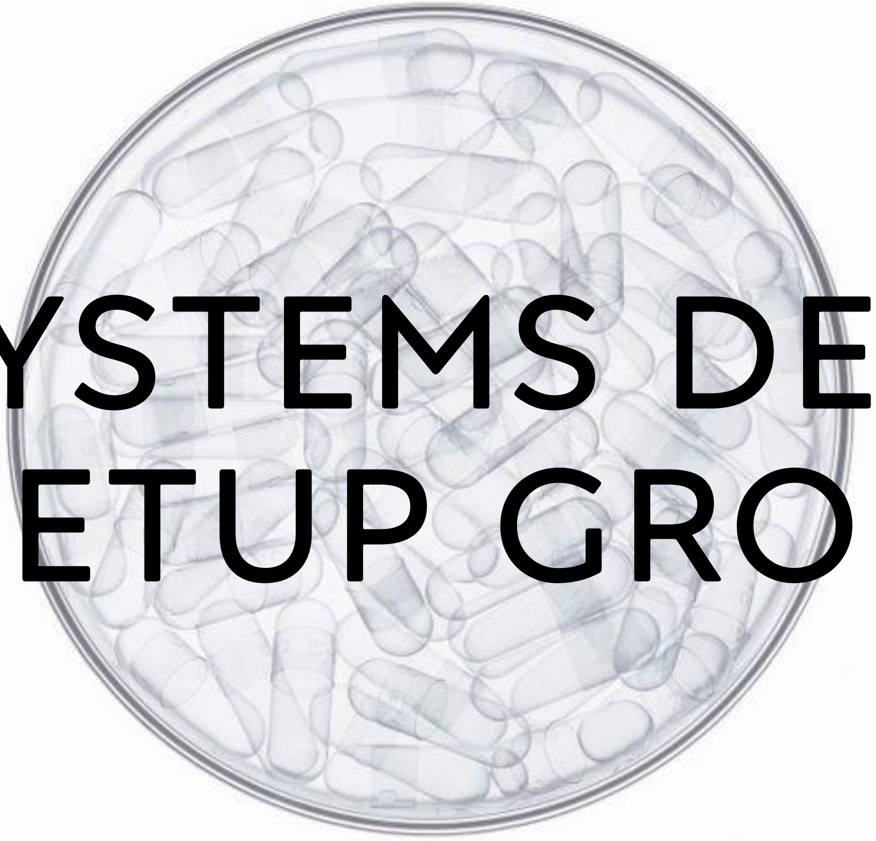
WE WILL START AT 1:05 PM



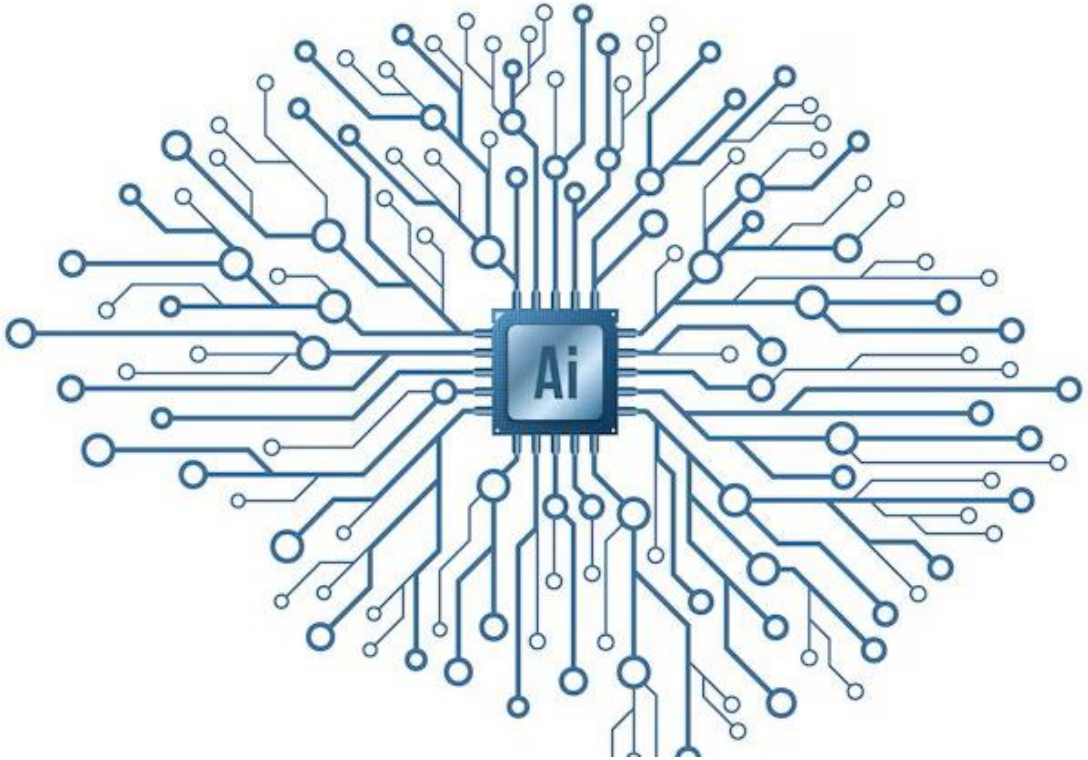
ML SYSTEMS DESIGN MEETUP GROUP

HETAV PANDYA

IMPORTANT: THIS WORKSHOP WILL BE RECORDED

HETAV PANDYA

AGENDA



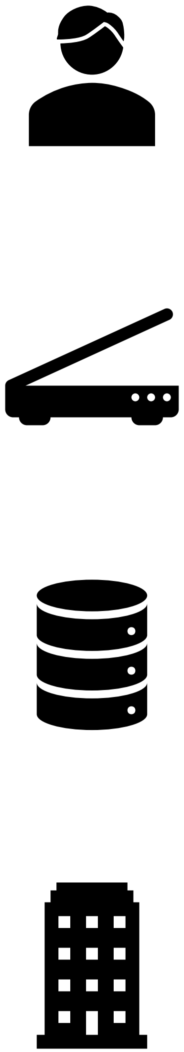
INTRODUCTION

- Welcome to ML Systems Design Meetup Group
- Designing Machine Learning Systems – Chip Huyen
- Free Access – City Library
- Frequency – Biweekly - Monthly
- Questions: Meetup Event Chat

FREE ACCESS

Via Burnaby Public Library

WHERE DID IT COME FROM?



Is usually malformatted

Can be potentially dangerous

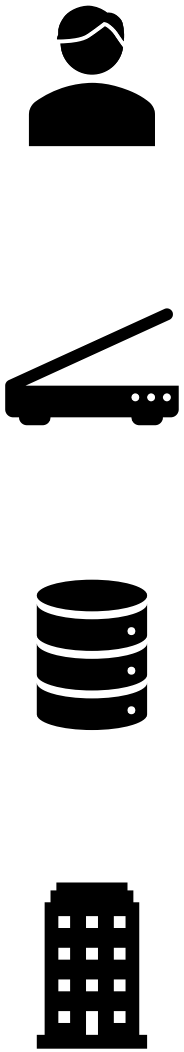
Requires fast processing

NoSQL: MongoDB, Cassandra, Redis

Blob storage: AWS S3, GCS, Azure Blob

Text inputs, form submissions, video uploads, etc.

WHERE DID IT COME FROM?



Much more safe

Lightly Structured

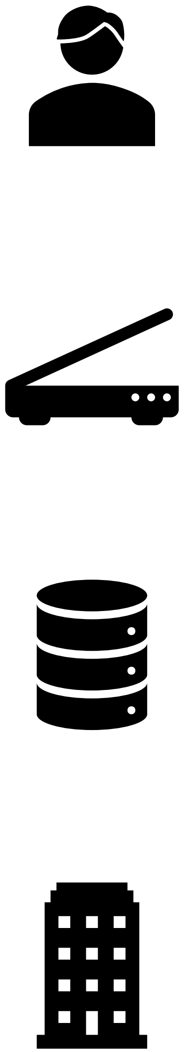
Slow processing is viable

ELK: ElasticSearch, Logstash, Kibana

Greylog for an OS solution

Logs, debug files, system metrics

WHERE DID IT COME FROM?



Much more safe

Heavily Structured

Needs to be persistent

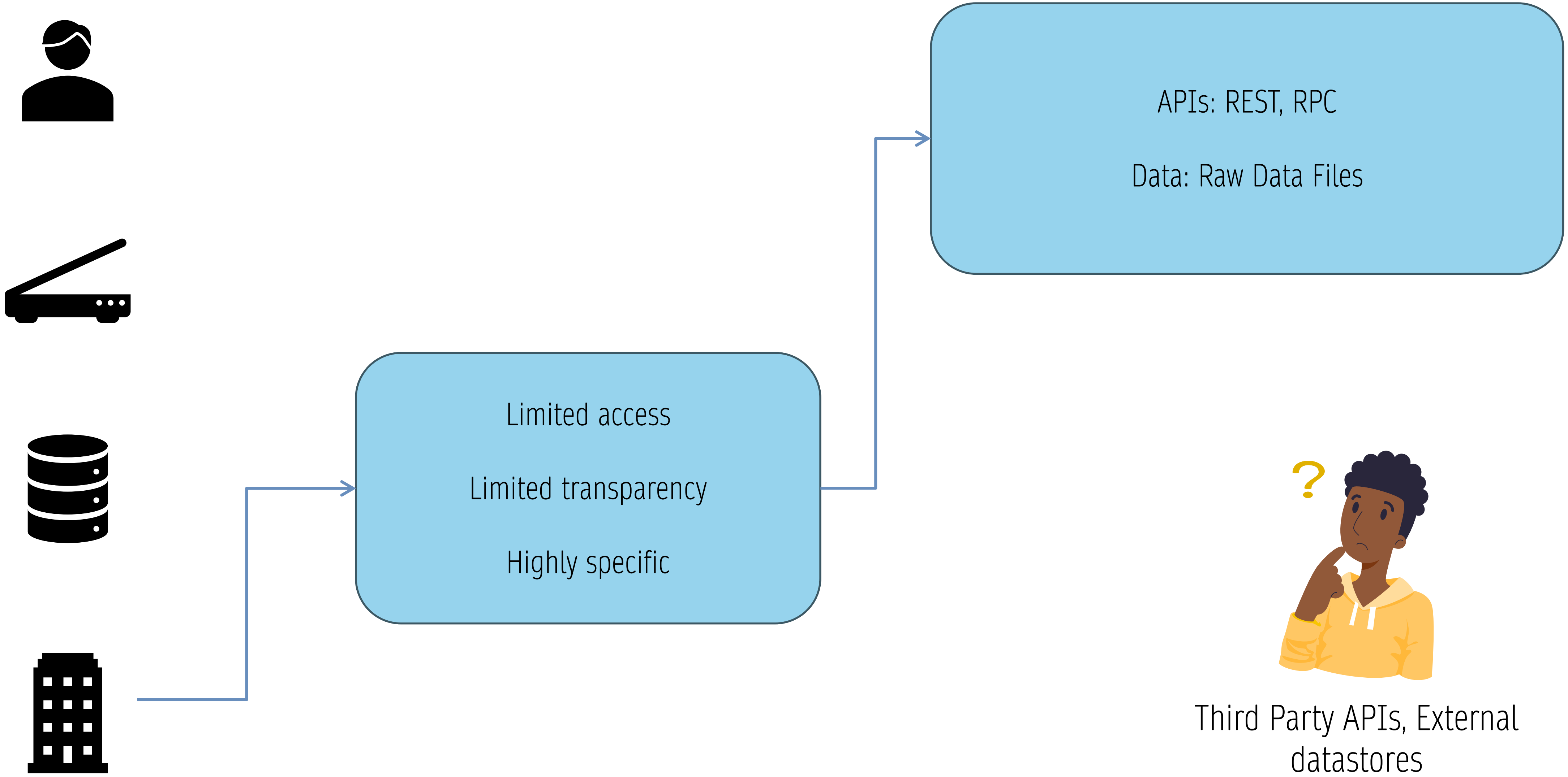
RDBMS: MySQL, PostgreSQL

ETL: Apache Kafka, AWS Glue

Data Warehouses: AWS Redshift, Google BigQuery

Transactions, orders, static details

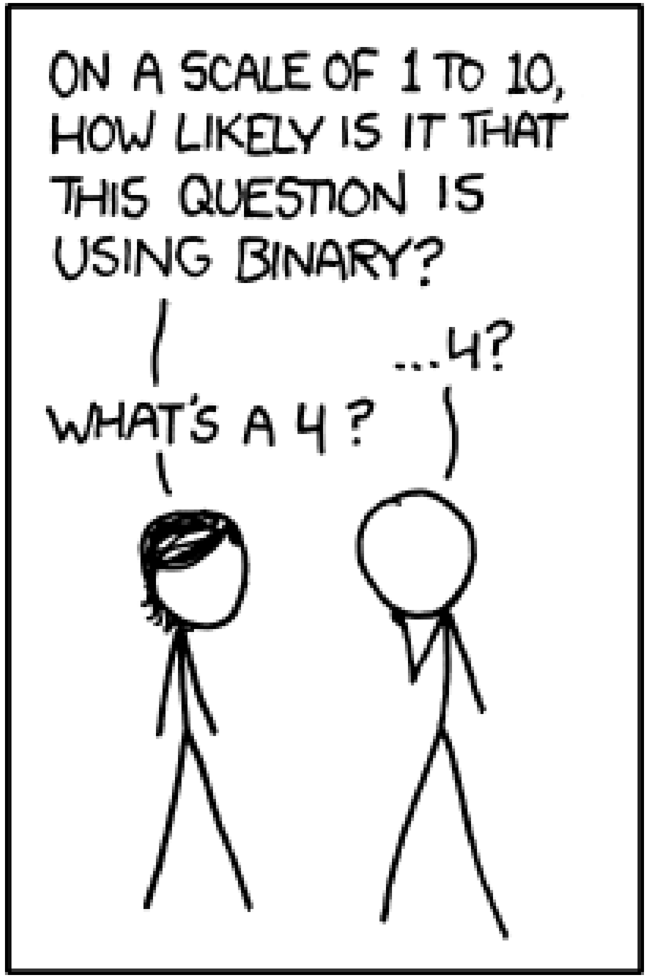
WHERE DID IT COME FROM?



THE FORMAT MATTERS!

Format	Binary/Text	Human-readable	Example use cases
JSON	Text	Yes	Everywhere
CSV	*Text*	*Yes*	*Everywhere*
Parquet	*Binary*	*No*	*Hadoop, Amazon Redshift*
Avro	Binary primary	No	Hadoop
Protobuf	Binary primary	No	Google, TensorFlow (TFRecord)
Pickle	Binary	No	Python, PyTorch serialization

DEMO TIME



Google Collab File

ROW MAJOR VS COLUMN MAJOR

Row Major	Column Major
Faster row reads	Faster column reads
Row data is sequentially stored on disk	Column data is sequentially stored on disk
Example: Numpy (by default)	Example: Pandas

SPEED -> TIME -> MONEY

In NumPy, the major order can be specified. When an ndarray is created, it's row-major by default if you don't specify the order.

People coming to pandas from NumPy tend to treat DataFrame the way they would ndarray, e.g., trying to access data by rows, and find DataFrame slow.

DEMO TIME (SUBJECT TO TIME)

Google Collab File

STRUCTURED VS UNSTRUCTURED

PostgreSQL

redis

cassandra

Structured data	Unstructured data
Schema clearly defined	Data doesn't have to follow a schema
Easy to search and analyze	Fast arrival
Can only handle data with a specific schema	Can handle data from any source
Schema changes will cause a lot of troubles	No need to worry about schema changes (yet), as the worry is shifted to the downstream applications that use this data
Stored in data warehouses	Stored in data lakes

Amazon S3

Elasticsearch + Logstash + Kibana

ETL FRAMEWORK

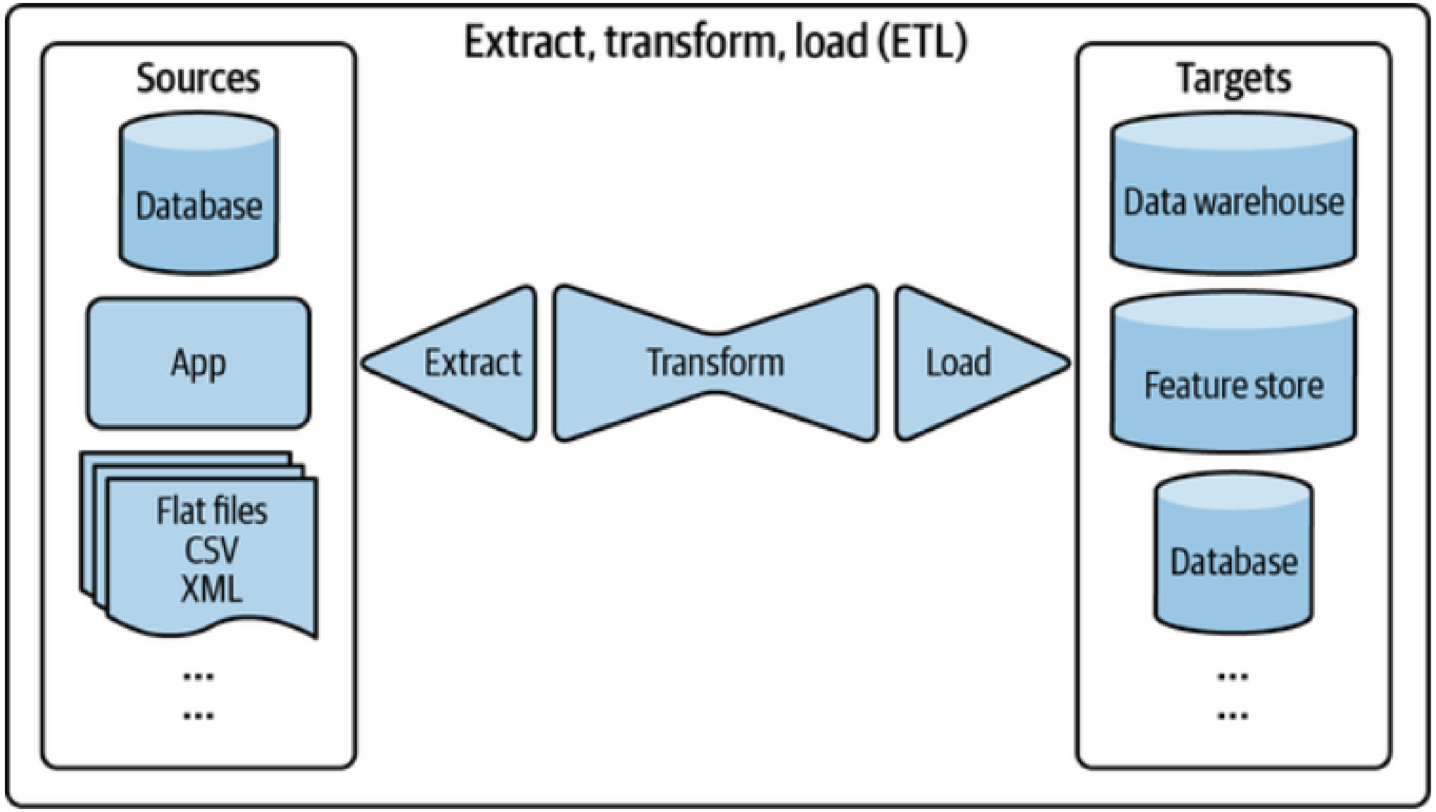
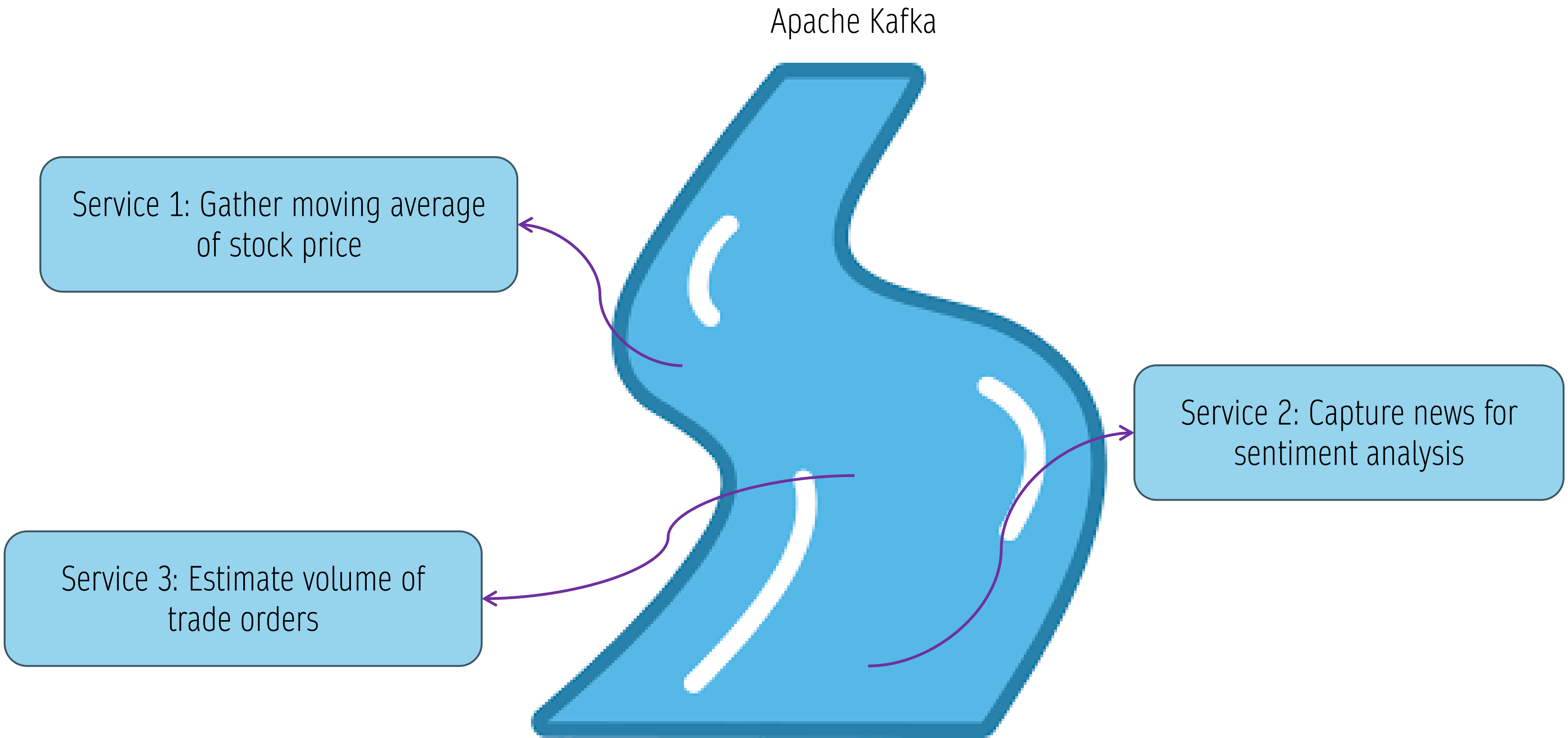


Figure 3-7. An overview of the ETL process

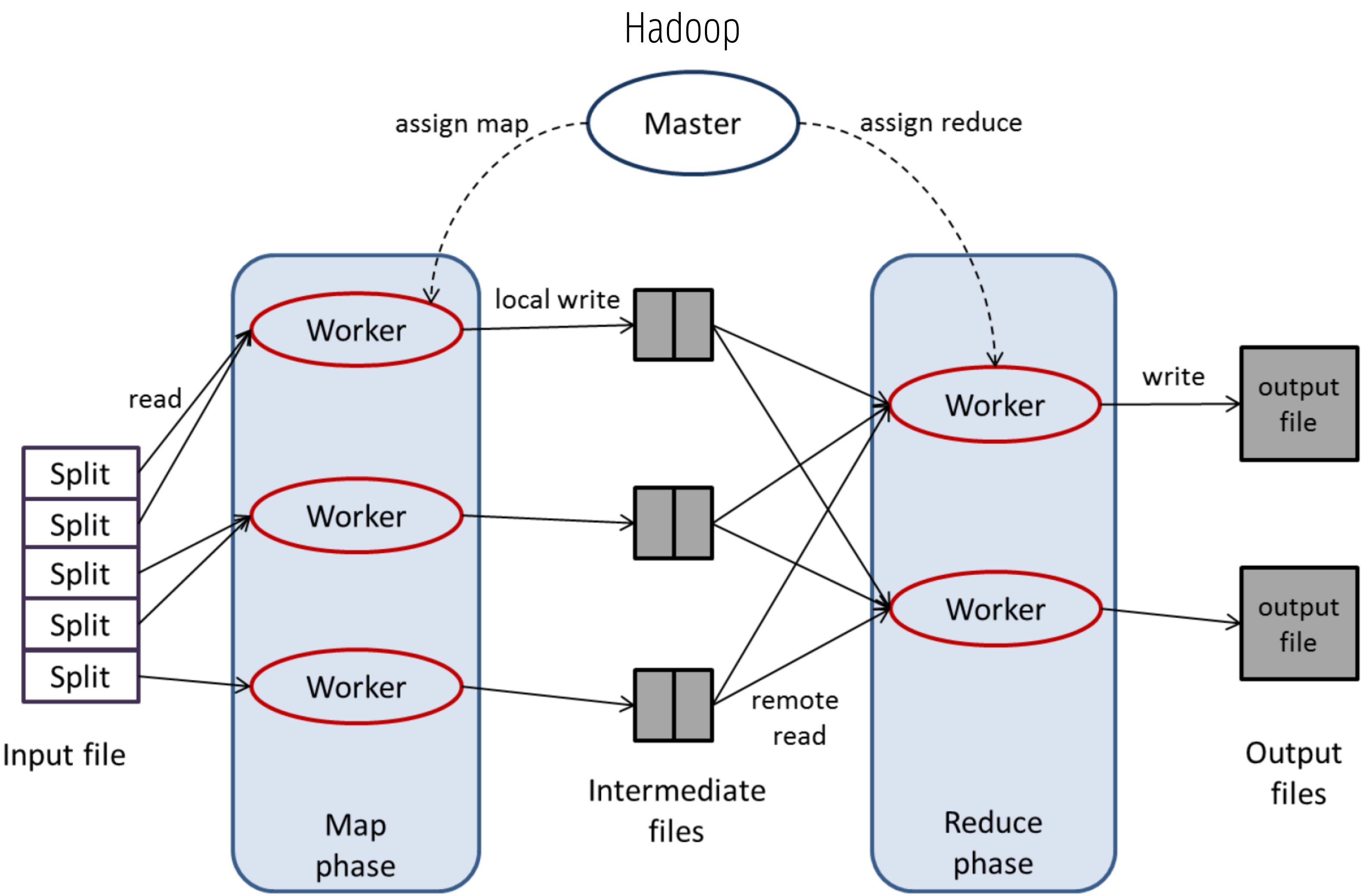
STREAM VS BATCH PROCESSING

Feature	Batch Processing	Stream Processing
Data Arrival	Data is collected over a period and processed later.	Data is processed in real-time as it arrives.
Processing Latency	High latency; data processed in batches (minutes to hours).	Low latency; data processed near real-time (milliseconds to seconds).
Data Volume	Handles large volumes of data at once (batch sizes).	Handles continuous data streams in small chunks.
Data Structure	Typically structured and stored in databases/files.	Data may be semi-structured or unstructured streams.
Use Cases	Periodic reporting - Historical analysis	Real-time analytics - Fraud detection
Services/Technologies	Hadoop, Spark, Hive	Apache Kafka, AWS Kinesis, Apache Flink, Spark Streaming
Machine Learning	Training models on historical data batches.	Online learning from streaming data. Real-time model inference.

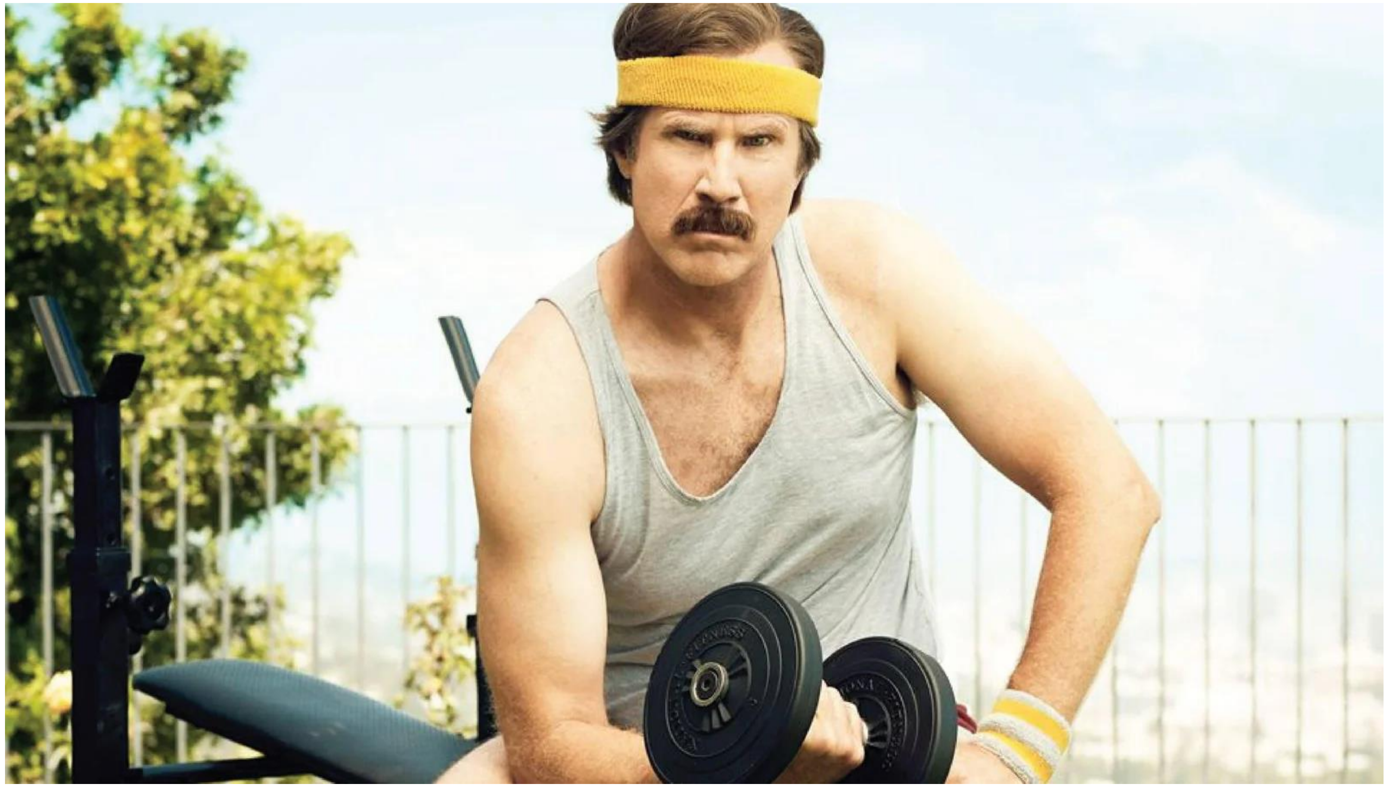
STREAM PROCESSING



BATCH PROCESSING



WHAT DO WE DO WITH THE DATA?



WHAT DO WE DO WITH THE DATA?

- Your model is as good as your data
- Sampling is a method of gathering data for training
- There are many types of sampling methods
    - Convenience sampling
    - Snowball sampling
    - Judgement sampling
    - Quota sampling
    - Simple random sampling
    - Weighted sampling
    - Reservoir sampling

RESERVOIR SAMPLING

Reservoir sampling is a fascinating algorithm that is especially useful when you have to deal with streaming data, which is usually what you have in production.

1. Put the first k elements into the reservoir.
2. For each incoming n^{th} element, generate a random number i such that $1 \le i \le n$.
3. If $1 \le i \le k$: replace the i^{th} element in the reservoir with the n^{th} element. Else, do nothing.

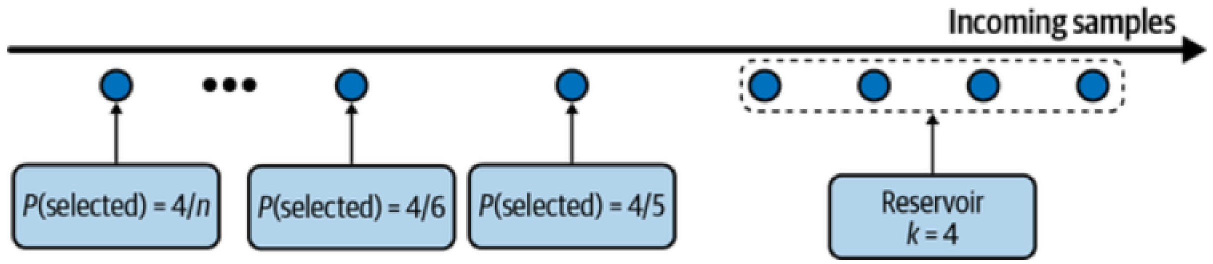


Figure 4-2. A visualization of how reservoir sampling works

WHAT TO EXPECT NEXT…

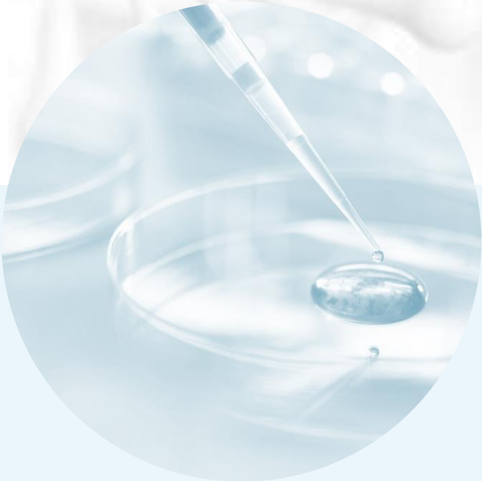
- Data labelling
- Labelling functions
- Natural labelling
- Weak supervision models
- Semi-supervised learning
- Transfer learning
- Class imbalance problems
- ROC curves
- Resampling
- Cost sensitive learning

YOUR VOICE MATTERS!

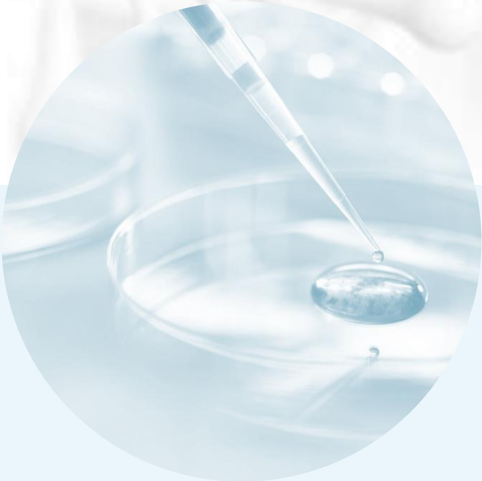
Please take some time to fill up our very very short feedback form ☺

If you would like to connect with me, feel free to scan this!

THANK YOU!



Q&A TIME



HOW CAN ORGANIZATIONS BALANCE THE NEED FOR COLLECTING DIVERSE DATA WITH THE ETHICAL CONCERNS AROUND PRIVACY, BIAS, AND CONSENT?