Eagleeye : Building Data Pipeline for Anomaly Detection

Tuhin Sharma
Senior Principal Data Scientist
Redhat

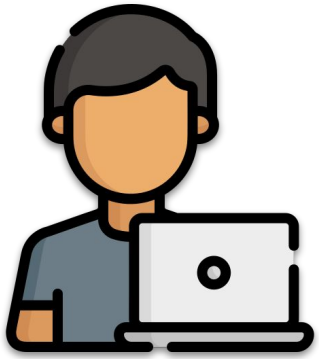
Outline

- The problem: overview
- Architecture
- Real-Time Processing
- Anomaly Detection
- Visualization
- Demo

The internet is exploding

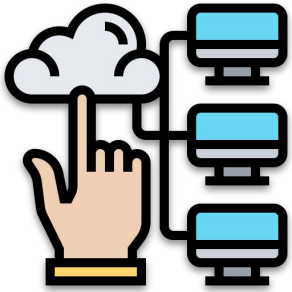
An estimated 4.1 billion people are using the Internet in 2019, reflecting a 5.3 per cent increase compared with 2018.

The global penetration rate increased from nearly 17 per cent in 2005 to over 53 per cent in 2019.



The internet is exploding

VM and container



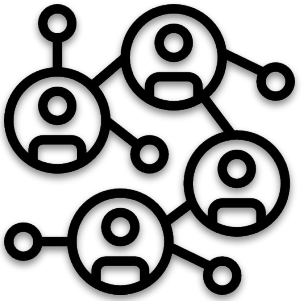
Cloud provider

Hybrid Cloud

How is it affecting Security Operations Control (SOC) analyst

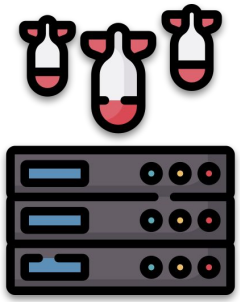
Network Policies

Network Request

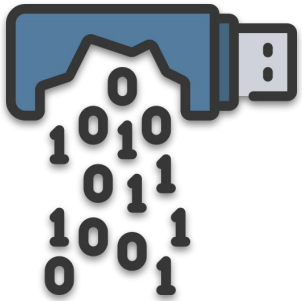


Network connection

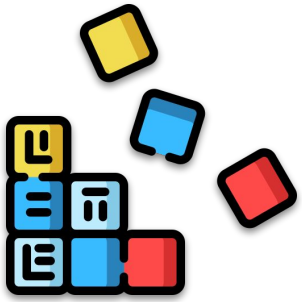
Security Challenges



Denial of service

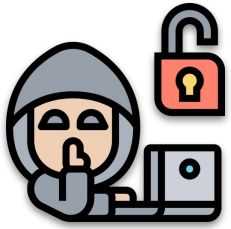


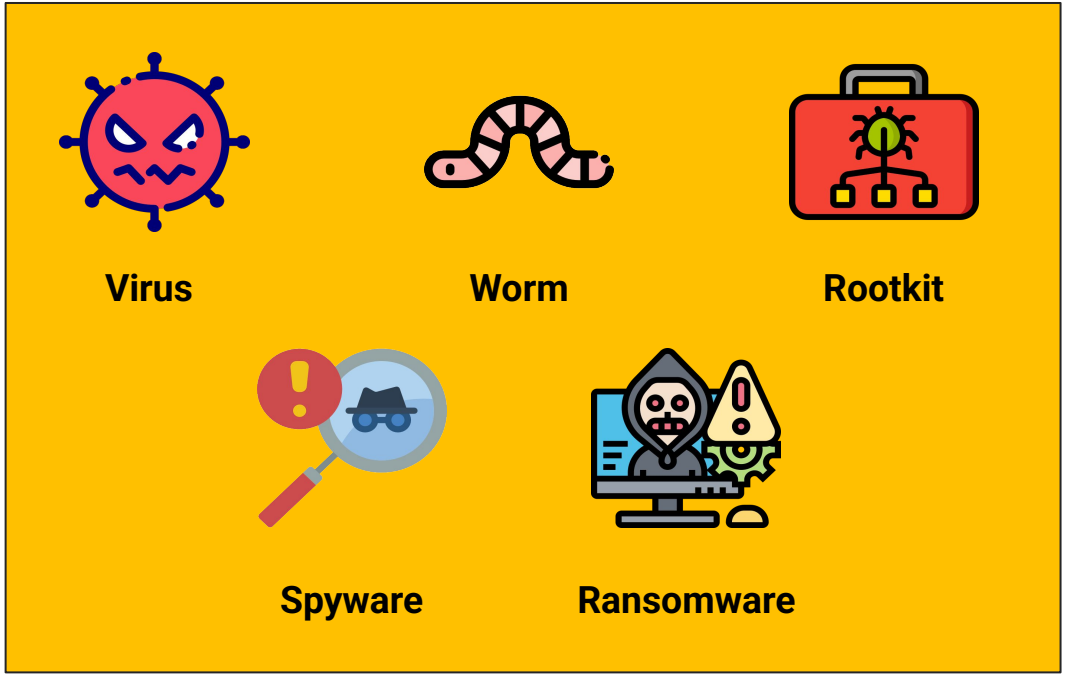
Data loss



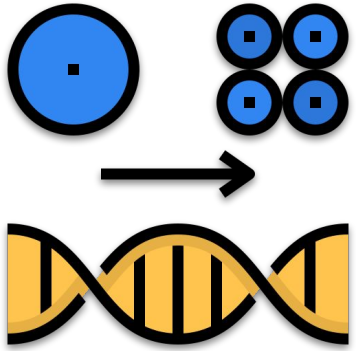
Data corruption

Existing Solutions - Malware based

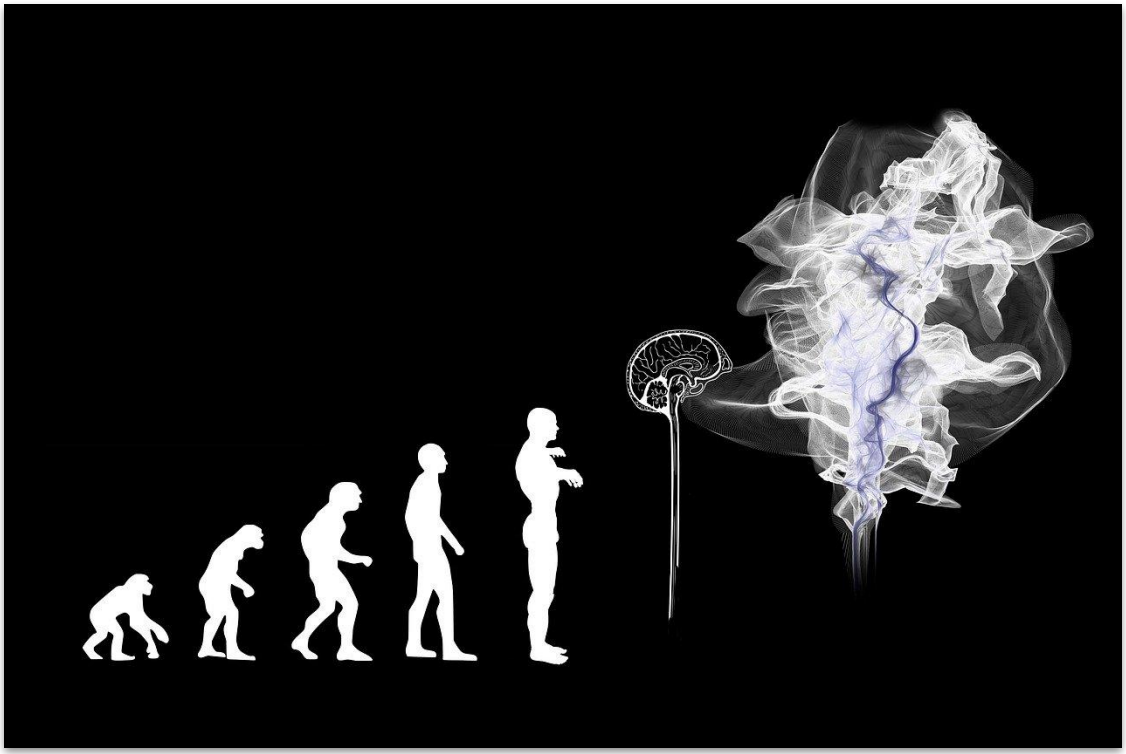




Existing Solution - Challenges



Machine Learning - WHY?



Machine Learning - How?

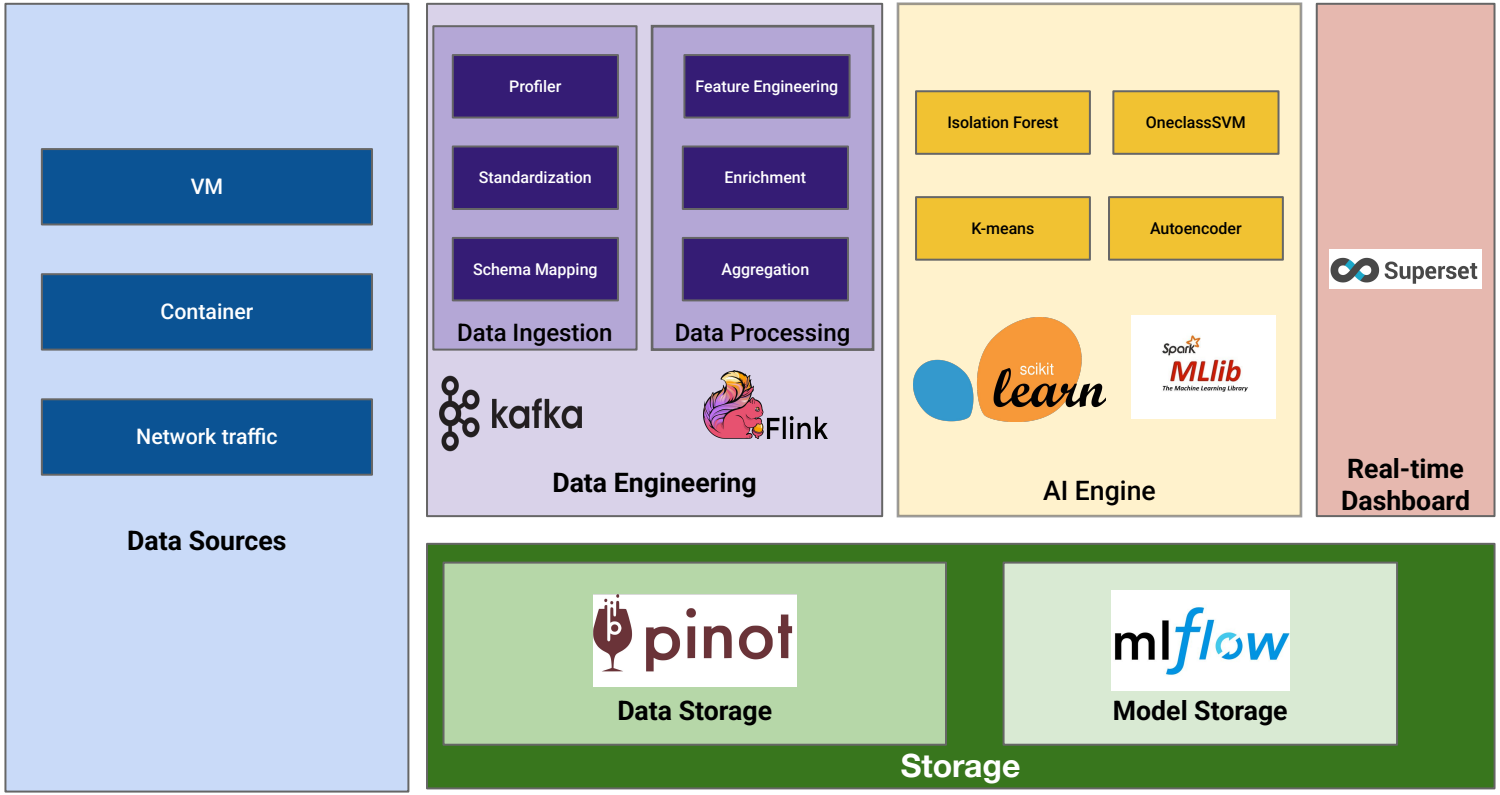
The core is a stream of time series data and the goal is to find anomalies in them

Architecture

Data Flow Architecture

Data Sources	Data Ingestion	Data Processing	AI	Analytics
				
User/IP Network event stream Byte information	Real-time ingestion Batch ingestion Data normalization	Data enrichment Data contextualization Feature creation	Machine learning Incremental training Active learning	Rich visualization Real-time Actionable insights

Product Architecture



RealTime Processing

Apache Kafka

(Distributed event streaming platform)

- Queueing and Streaming
- Why Kafka?Kafka vs RabitMQ

Queueing vs Streaming

Message Queue	Streaming broker
Producer-Consumer model : can have one or more consumers and/or producers. In a message queue with multiple consumers, the queue will attempt to distribute the messages evenly across them, with the guarantee being that every message will only be delivered once.	**Publish-Subscribe model** : messages are organized into log files or topics. One or more consumers can subscribe to a log file or topic to receive all messages that come through that stream. With proper setup, a streaming broker will deliver the same message to every subscriber, in a specific order.
Message queues only deliver **each message once, to a single consumer**.	Streaming brokers can deliver the **same message to many consumers** without the need for replication.
Message queues process on a **first-come, first-serve** basis.	Consumers can be set up to do **different things with the same message**.
Message queues **may not deliver in the same order** messages are queued.	Streaming brokers **always deliver in the same order** messages are queued.

Why Kafka? Kafka vs RabitMQ

Feature	Kafka	RabitMQ
Message Ordering	Supported	Not Supported
Message Lifetime	Always there	Done away after consumed
Delivery Guarantee	Guarantees atomicity	Does not guarantee atomicity
Message Priorities	Not supported	Supported
Performance	High throughput with limited resource	High throughput with more resources.

Apache Flink

(Distributed RT processing engine)

- Micro-batch processing vs Stream processing?
- Why Flink? Flink vs Spark-streaming

Micro-batch processing vs Stream processing

Micro-batch processing	Stream processing
System only checks for new data **periodically**, and only processes that data when the next batch window occurs.	System is designed to **continuously** monitor for new data and dispatch processing as soon as that data is received.
Process data **with a delay**. In a data pipeline with multiple steps, those delays accumulate.	Process data **as soon it is available**.
For use cases where having the **most up-to-date data is not important** and where **tolerance for slower response time is higher**.	For use cases that require **live interaction** and **real-time responsiveness**.
Offline analysis of historical data to compute results, **identify correlations** etc.	**Financial transaction processing**, **real-time fraud detection**, and **real-time pricing** etc**.**

Why Flink? Flink vs Spark-streaming

Feature	Flink	Spark Streaming
Processing mechanism	First True streaming framework with all advanced features like event time processing, watermarks, etc	Not true streaming, not suitable for low latency requirements.
Data latency	process rows after rows of data in real time.data latency is not there	processes chunks of data, known as RDDs .data latency is always there
Parameters	Auto-adjusting, not too many parameters to tune	Too many parameters to tune. Hard to get it right.
Community	Community is not as big as Spark but growing at fast pace now	Big community and aggressive improvements

Apache Pinot

- What is OLAP?
- Why Pinot ?

What is OLAP

Roll up

Drill down

Slice & Dice

Why Pinot?

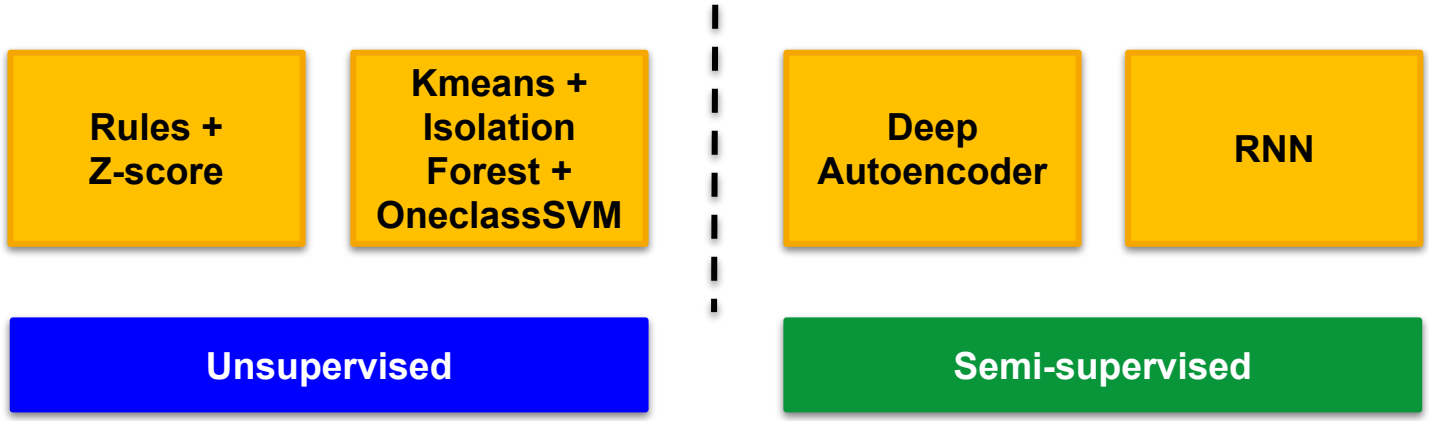
- Blazing Fast
- Pluggable Indexing
- Realtime Ingestion
- Horizontally Scalable
- PQL
- Joins using Trino and PrestoDB
- Hybrid Tables
- Anomaly Detection

Anomaly Detection

AI Model

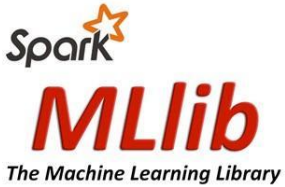
- Our journey
- Tools

Our Journey


Rules + Z-score
Kmeans + Isolation Forest + OneclassSVM
Deep Autoencoder
RNN
Unsupervised
Semi-supervised

Tools


scikit
learn


Spark
MLlib
The Machine Learning Library


PyTorch

MLflow

- Model management & mlflow

What is MLflow

MLflow Tracking

Record and query experiments: code, data, config and results

MLflow Projects

Package DS code in a format to reproduce runs on any platform

MLflow Models

Deploy machine learning models in diverse serving environments

MLflow Registry

Deploy machine learning models in diverse serving environments

Visualization

Apache Superset

- What is it?
- Near real-time dashboard

Near Real-time Dashboard

Explore

Explore your data using the array of data visualizations.

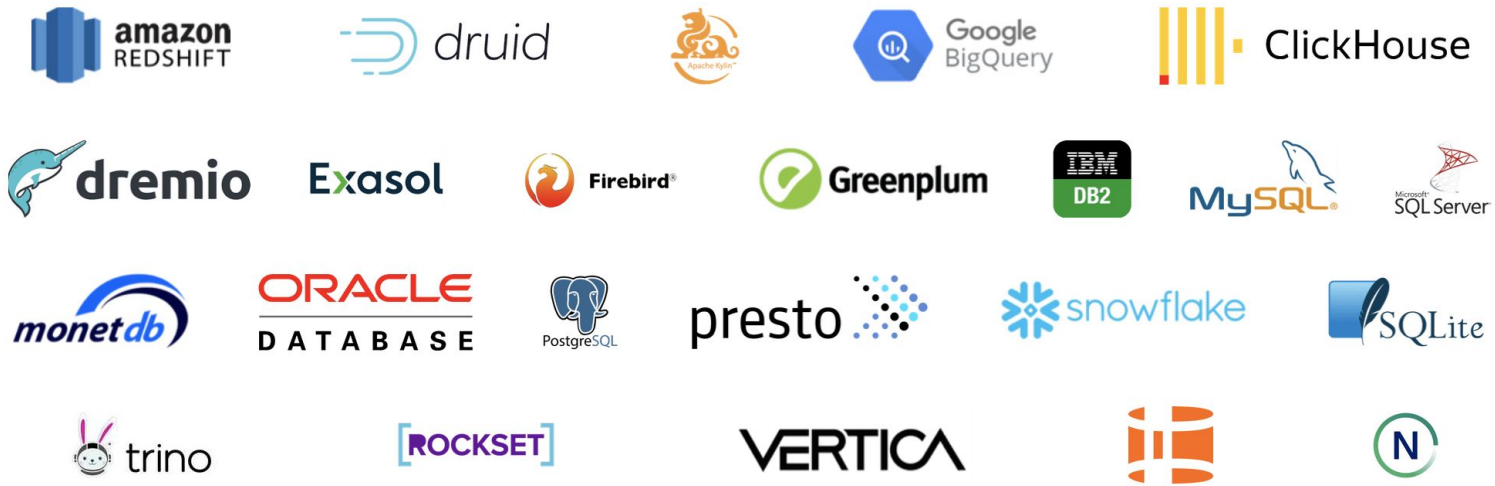
View

View your data through interactive dashboards

Investigate

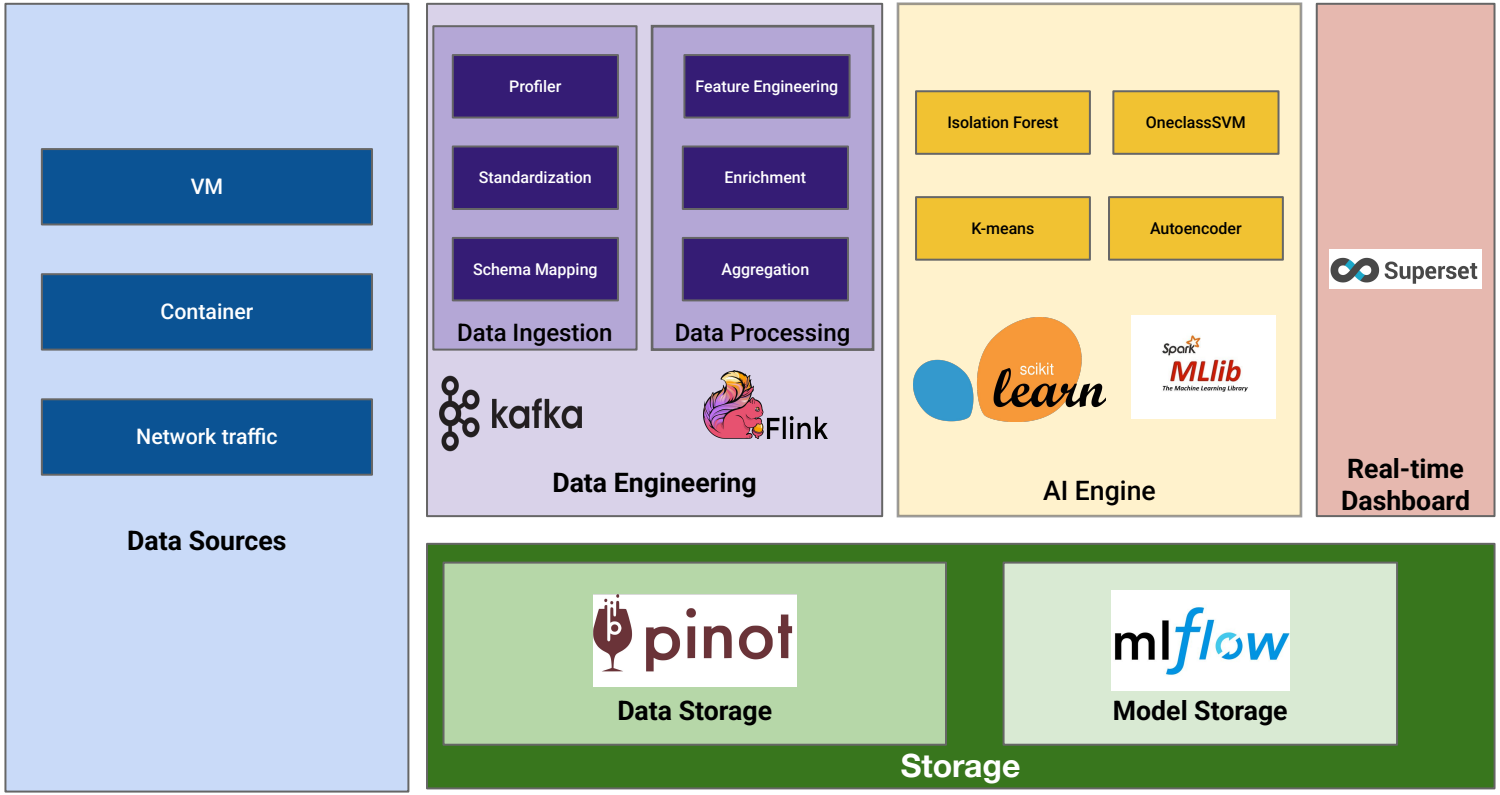
Use SQL Lab to write queries to explore your data

Supported Databases

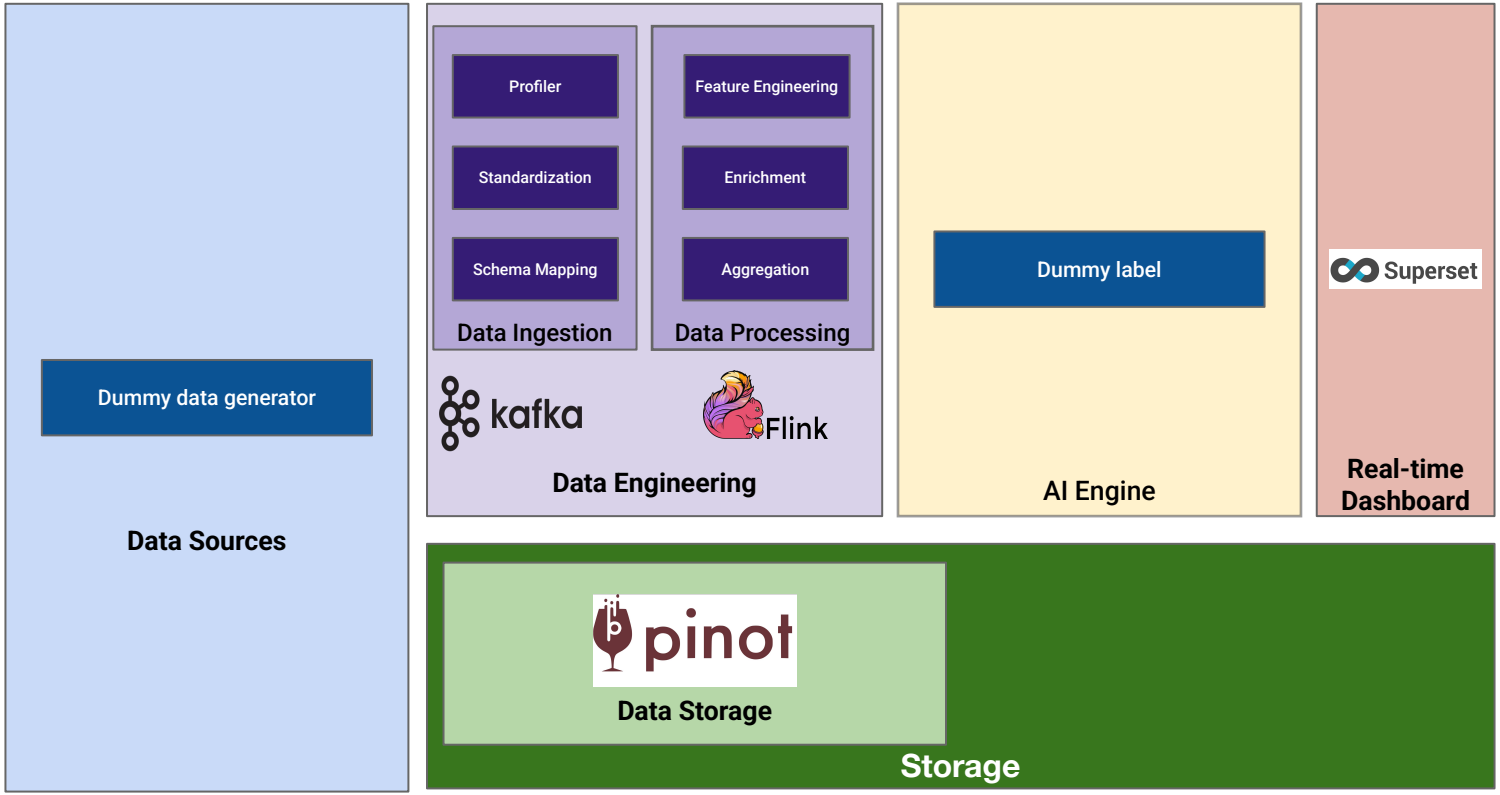


Demo!!

Product Architecture (We discussed)



Demo Setup


Dummy data generator
Data Sources
Profiler
Standardization
Schema Mapping
Data Ingestion
Feature Engineering
Enrichment
Aggregation
Data Processing
kafka
Flink
Data Engineering
Dummy label
AI Engine
Superset
Real-time Dashboard
pinot
Data Storage
Storage

Sample Data

Raw network event


```
{
  "ip": "32.534.234.12",
  "eventTimestamp": "28-01-2022 11:03:04",
  "isFile": "false",
  "stateCode": "US-FL",
  "bytes": 106975,
  "year": "2022",
  "month": "01",
  "day": "28",
  "hour": "11",
  "minute": "03",
  "second": "04"
}
```


Aggregated IP profile


```
{
  "windowStart": "28-01-2022 11:03:00",
  "windowEnd": "28-01-2022 11:03:15",
  "ip": "32.534.234.12",
  "numBytesSent": 1000401,
  "numRequestSent": 7,
  "numFilesSent": 1,
  "threatLevel": "low"
}
```

References

- Message Queue vs Streaming
- Kafka vs. RabbitMQ: Architecture, Performance & Use Cases
- Spark Streaming vs Flink vs Storm vs Kafka Streams
- MLflow
- Apache Superset
- Apache Pinot
- Apache Flink
- Apache Kafka

Code is available at

- https://github.com/tuhinsharma121/eagleeye

Thank you!!