Let's Do Data Lineage in Kafka, Flink and Druid by Tracking Aircraft!

Hellmar Becker, Senior Sales Engineer

About Me

Hellmar Becker
Sr. Sales Engineer at Imply
Lives near Munich

hellmar.becker@imply.io
https://www.linkedin.com/in/hellmarbecker/
https://blog.hellmar-becker.de/

Agenda

- About Streaming Data Governance
- Kafka Headers: A lesser known feature
- Stream ETL with Flink using Kafka Headers
- Quick intro to Apache Druid - A Streaming Analytics Database
- Tracking Aircraft Radar data with Raspberry Pi
- Let's put it all together
- Live Demo
- Q&A

Streaming Data Governance

Data governance is a collection of standards, processes, roles, and metrics that ensure that data is usable, accessible, and effective.

Data governance with respect to streaming is about

- Stream Quality
- Stream Catalog
- Stream Lineage

Today, let's shine a light on the **Lineage** aspect!

Kafka Headers: A lesser known feature

Record headers give you the ability to add some metadata about the Kafka record. They are collections of arbitrary key/value pairs.

Kafka views headers as Kafka headers are key-value pairs, where the header key is a `java.lang.String` type and the header value is a byte array.

Flink models headers as a `MAP<BYTES, BYTES>` virtual (metadata) column. Use `DECODE/ENCODE` with known character set (usually UTF-8) if you need to process the string values

Druid can read Kafka record headers natively:

- Autogenerate table columns from header keys. For ingestion, you specify a prefix to create column names
- For decoding the values, specify the character set

Stream ETL with Flink using Kafka Headers

Flink views Kafka headers as *metadata* columns

```
CREATE TABLE `adsb-raw` (
  `kafka_timestamp` TIMESTAMP_LTZ(3) METADATA FROM 'timestamp',
  `kafka_headers` MAP<BYTES, BYTES> NOT NULL METADATA FROM 'headers',
  `kafka_key` STRING,
  `val` STRING
) WITH (
  'connector' = 'kafka',
  'topic' = 'adsb-raw',
  'key.format' = 'csv',
  'key.fields' = 'kafka_key',
  'key.fields-prefix' = 'kafka_',
  'value.format' = 'raw',
  …
);
```

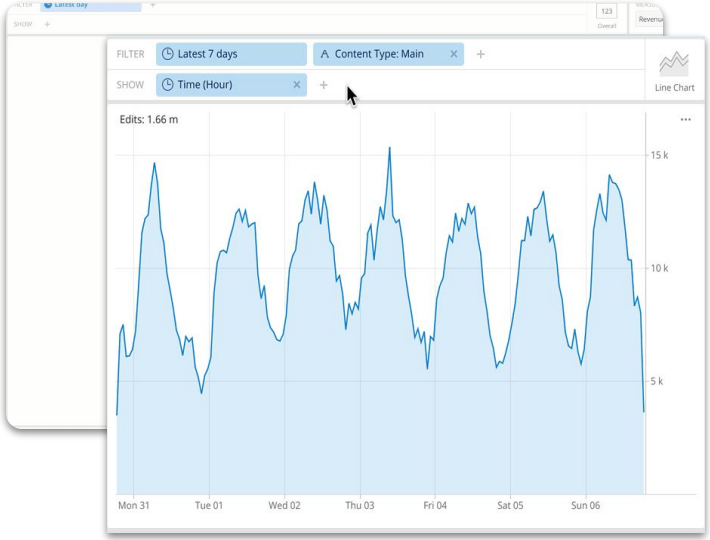
Metadata columns can be read and written (unless declared virtual)

Apache Druid - A Streaming Analytics Database

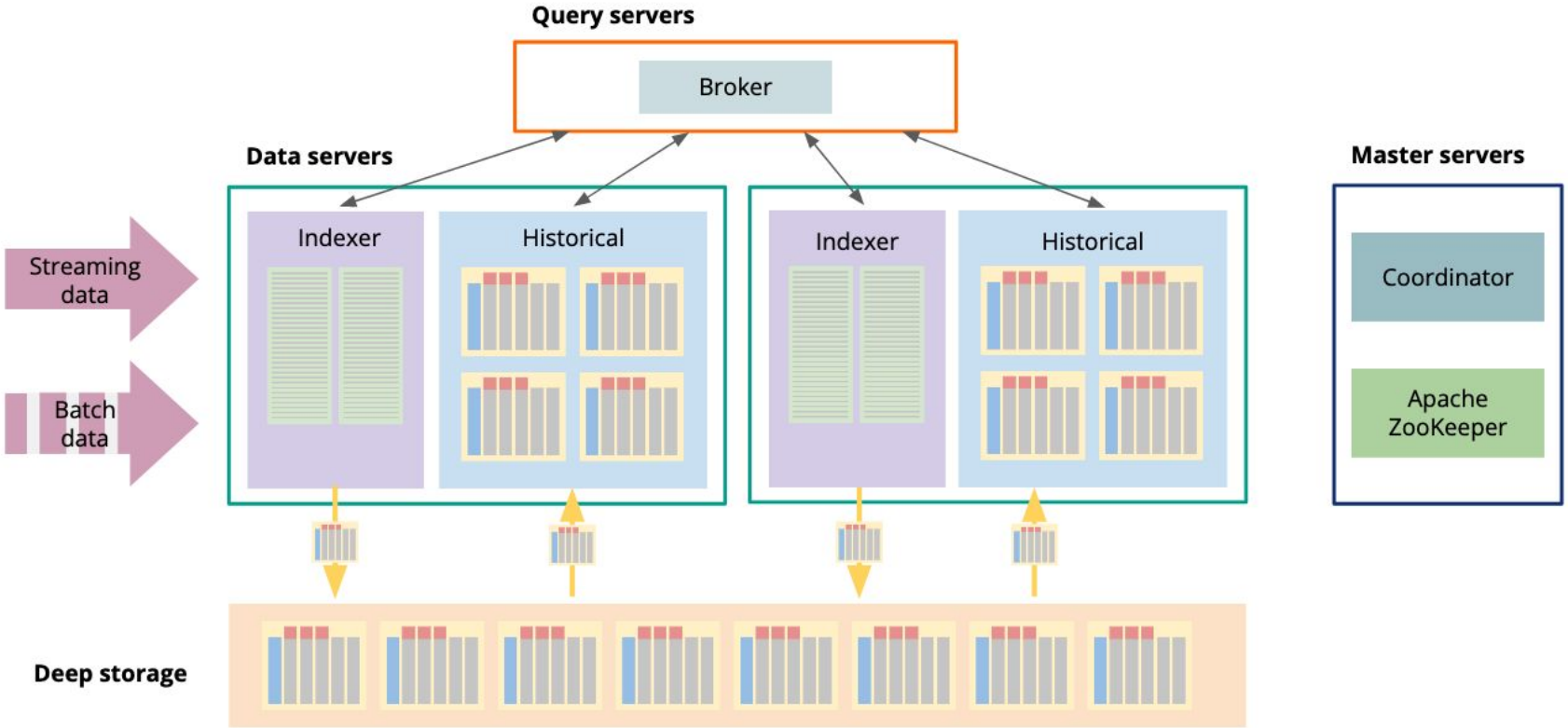
For analytics applications that require:

1. **Sub-second queries at any scale**
   Interactive analytics on TB-PBs of data
2. **High concurrency at the lowest cost**
   100s to 1000s QPS via a highly efficient engine
3. **Real-time and historical insights**
   True stream ingestion for Kafka and Kinesis

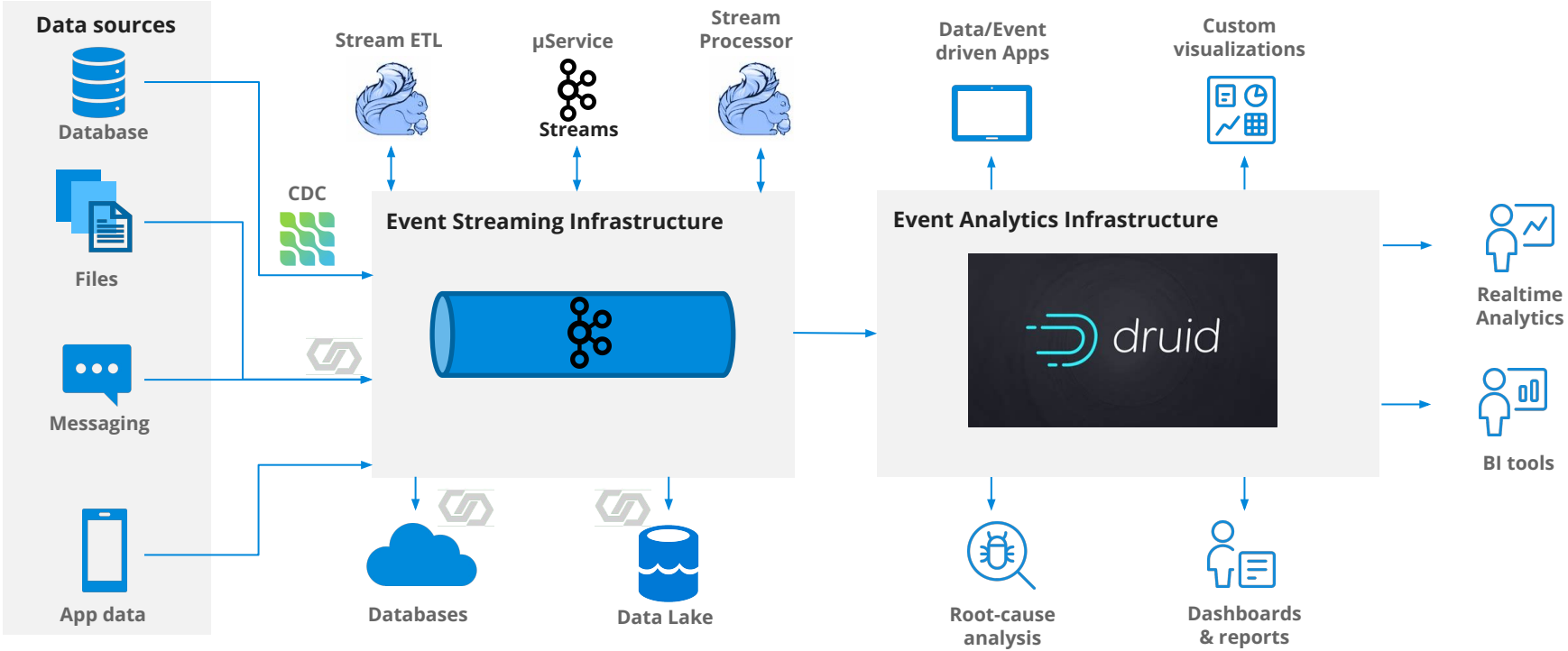
★ Plus, **non-stop reliability** with automated fault tolerance and continuous backup



Apache Druid - A Streaming Analytics Database



KFD Stack - Kafka Flink Druid



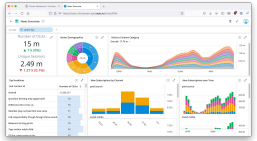
Tracking Aircraft Radar data with Raspberry Pi


```
#!/bin/bash

CC_BOOTSTRAP="<confluent cloud bootstrap server>"
CC_APIKEY="<api key>"
CC_SECRET="<secret>"
CC_SECURE="-X security.protocol=SASL_SSL -X sasl.mechanism=PLAIN -X
sasl.username=${CC_APIKEY} -X sasl.password=${CC_SECRET}"
CLIENT_ID="<client id>"
CLIENT_TIMEZONE=$(date +"%Z")
LON="0.0"
LAT="0.0"
TOPIC_NAME="adsb-raw"

nc localhost 30003 \
       | awk -F "," '{ print $5 "|" $0 }' \
       | kcat -P \
       -t ${TOPIC_NAME} \
       -b ${CC_BOOTSTRAP} \
       -H "ClientID=${CLIENT_ID}" \
       -H "ClientTimezone=${CLIENT_TIMEZONE}" \
       -H "ReceiverLon=${LON}" \
       -H "ReceiverLat=${LAT}" \
       -K "|" \
       ${CC_SECURE}
```

Let's put it all together

	Data Production	Delivery	Processing	Storage	Query	Visualisation
Analytics Pipeline	 					 
	Aircraft transponder data (ADS-B)	**Kafka** as an event streaming platform - Track provenance using header fields	Preprocessing with **Apache Flink**: - Filter - Enrich - Transform - Process header fields	**Apache Druid** - Highly scalable, built in DR - Native Kafka connectivity - Analyze Kafka header fields	**Apache Druid** Analytical queries against realtime, detail data	**Imply Pivot** - a data exploration and adhoc analytics GUI for Druid

imply

Live Demo

Learnings

- Lineage is a crucial part of Data Governance
- We can track lineage by adding metadata to each data record
- One way to do this is by using Kafka record headers
- The KFD (Kafka Flink Druid) stack supports tracking data lineage with Kafka headers end-to-end
- Don't be scared of "enterprisey" things, you can try this at home with public data sources such as aircraft data!

Links

- Streaming Governance training by Confluent: https://developer.confluent.io/courses/governing-data-streams/overview/
- Catalogs in Flink SQL (Decodable blog): https://www.decodable.co/blog/catalogs-in-flink-sql-a-primer
- Kafka headers: https://www.confluent.io/blog/5-things-every-kafka-developer-should-know/#tip-5-record-headers
- Github repo: https://github.com/hellmarbecker/plt-airt-2000
-

Questions

hellmar.becker@imply.io
https://www.linkedin.com/in/hellmarbecker/
https://blog.hellmar-becker.de/