Formula 1 telemetry processing using Kafka Streams

Paolo Patierno
Principal Software Engineer

Tom Cooper
Senior Software Engineer

Red Hat

Who are we?

Principal Software Engineer @Red Hat
Working on Apache Kafka and Strimzi

@ppatierno

Senior Software Engineer @Red Hat
Working on Apache Kafka and Strimzi

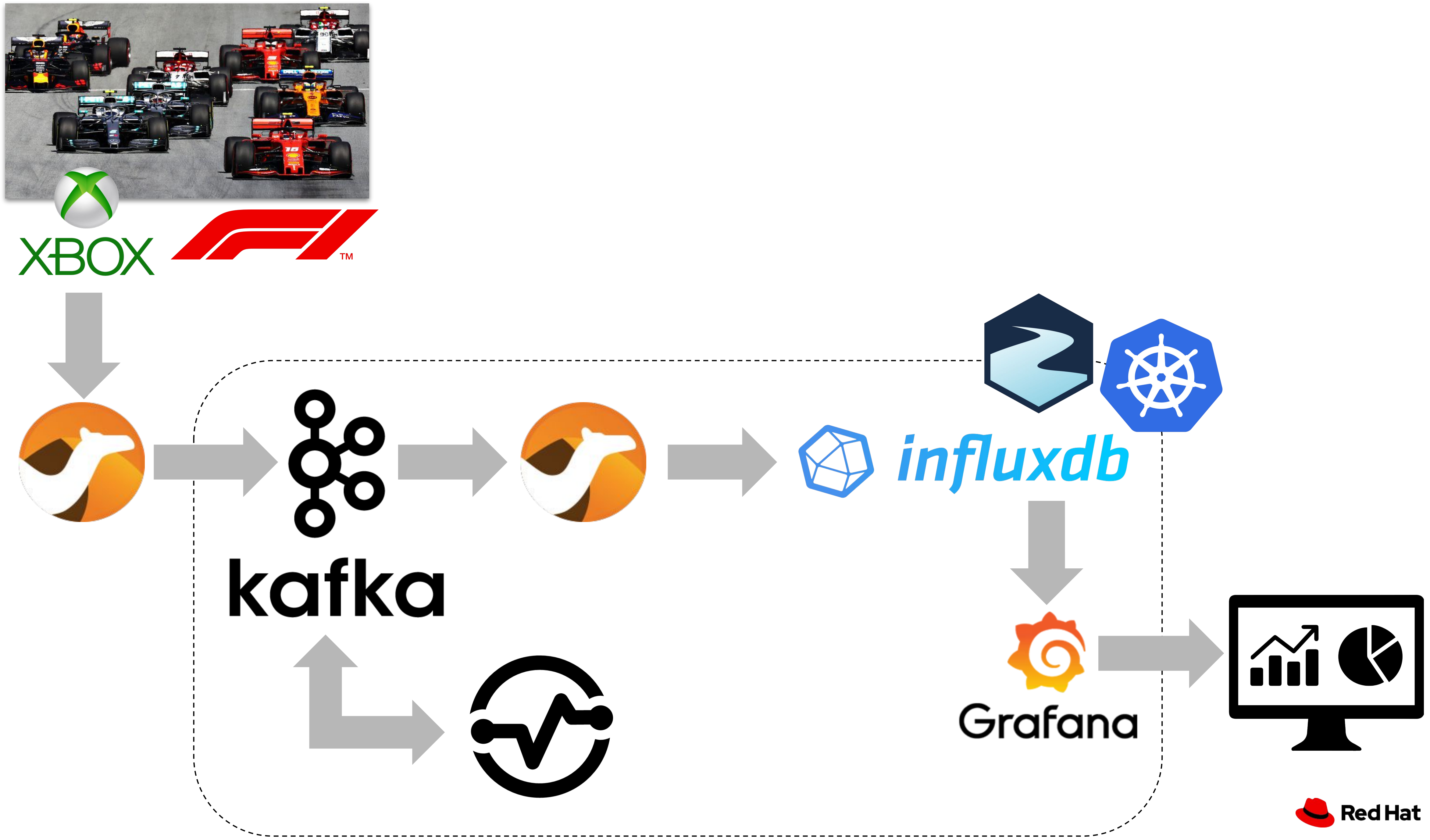
@tomncooper

Red Hat

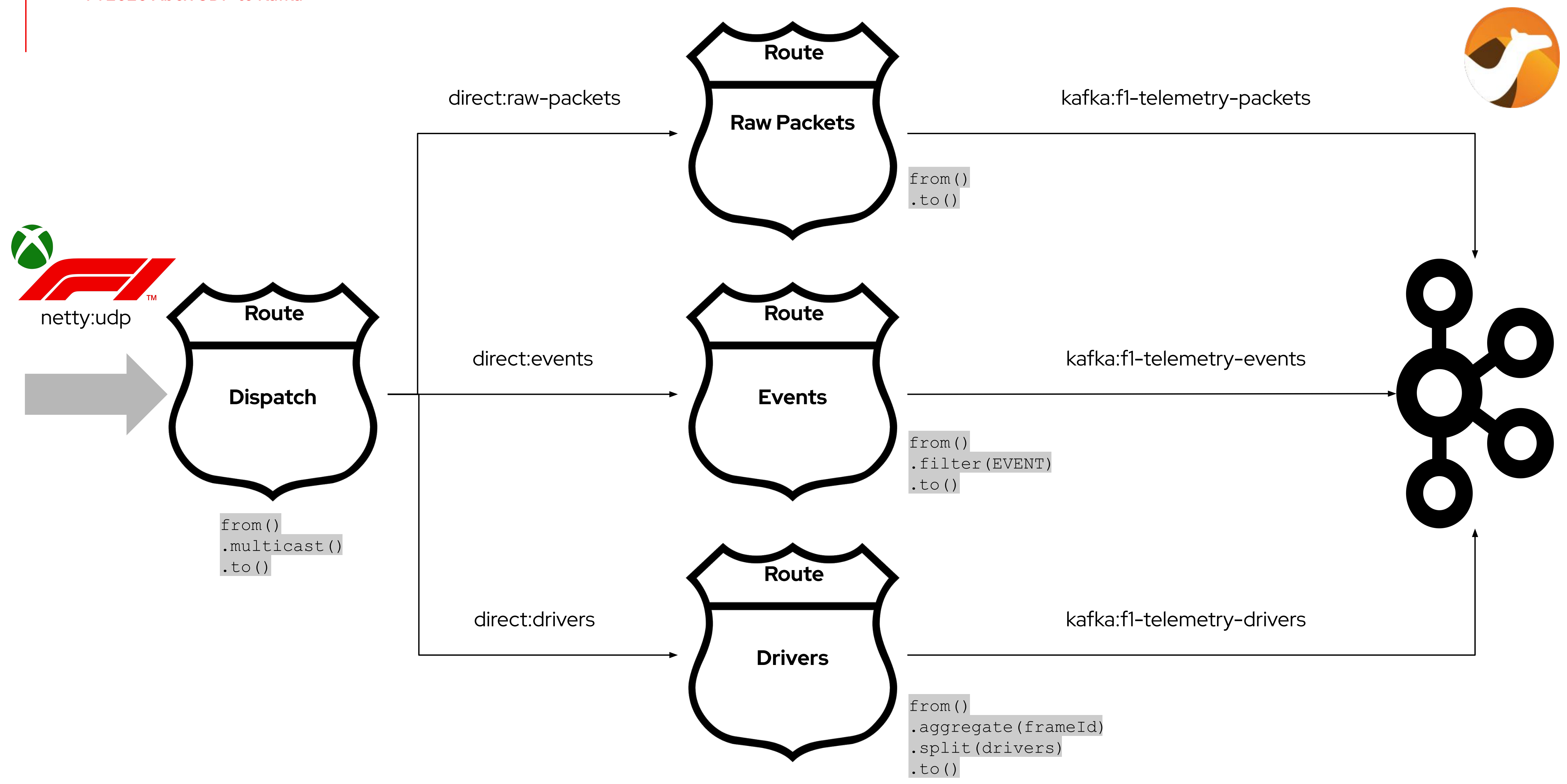
Building an events stream pipeline:

- How to ingest events reliably
- How to integrate with different systems for events ingestion (UDP) and providing output
- How to process events in real time
- How to show useful insights
- How to run and deploy the entire pipeline

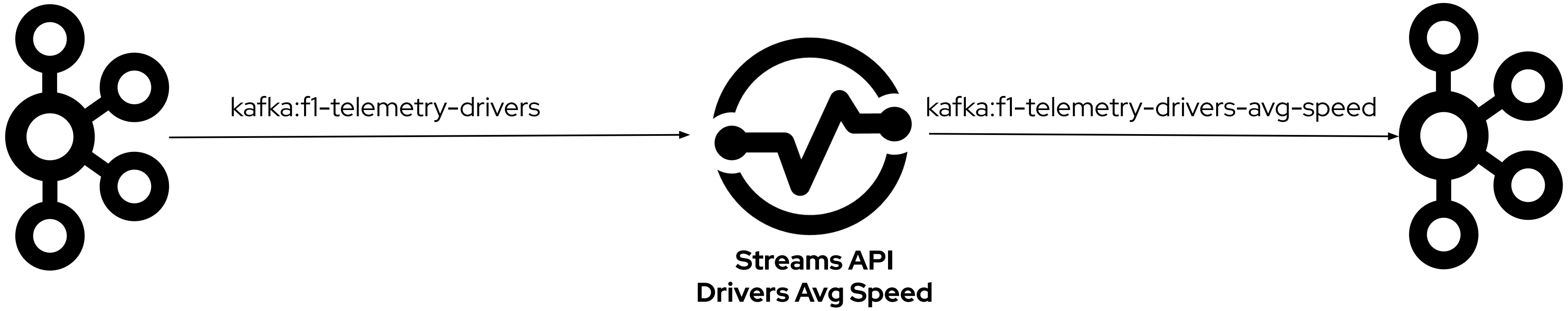
Red Hat


XBOX
kafka
influxdb
Grafana

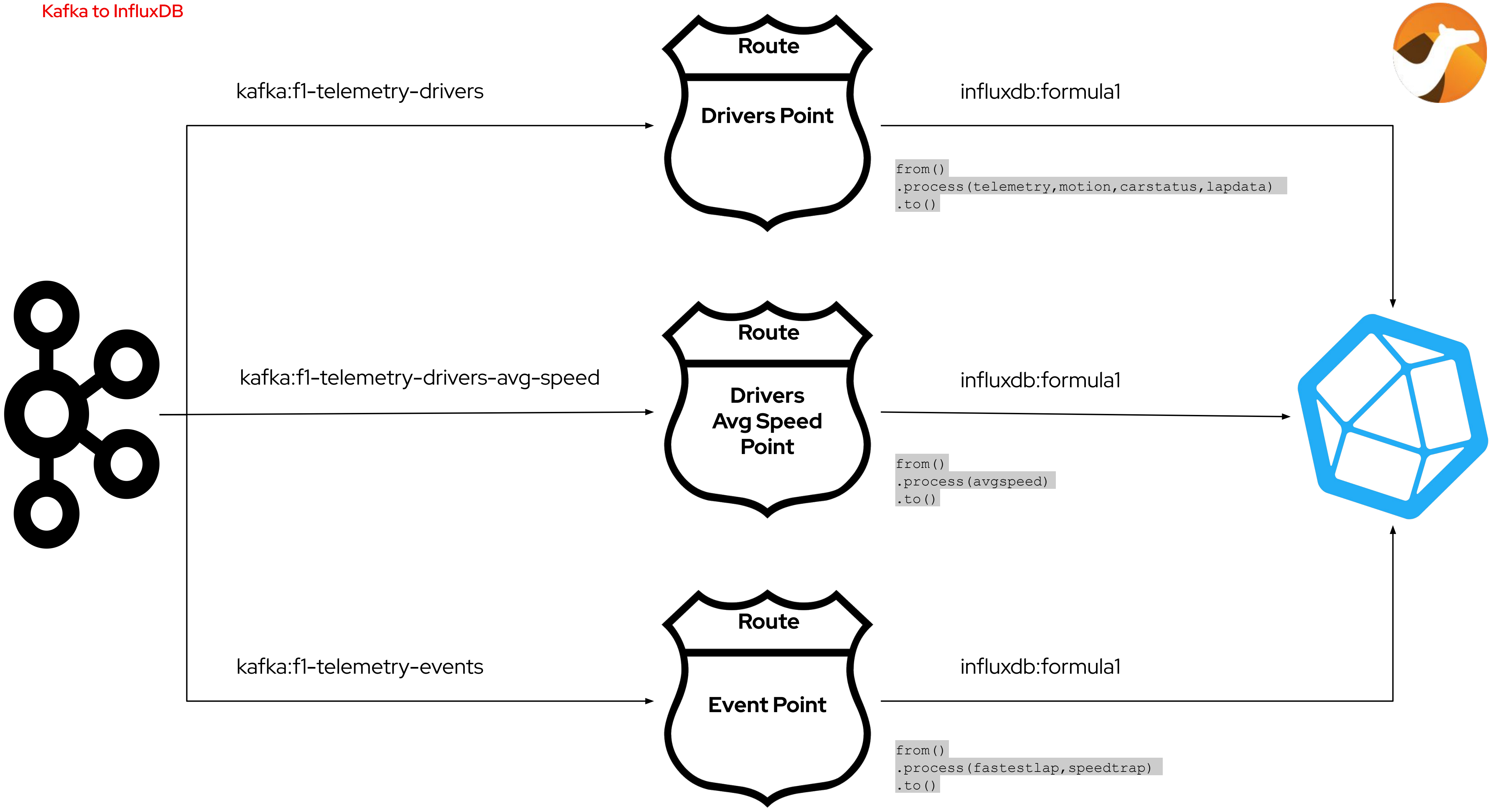
F1 2020 Xbox UDP to Kafka



Red Hat


kafka:f1-telemetry-drivers
kafka:f1-telemetry-drivers-avg-speed
Streams API
Drivers Avg Speed

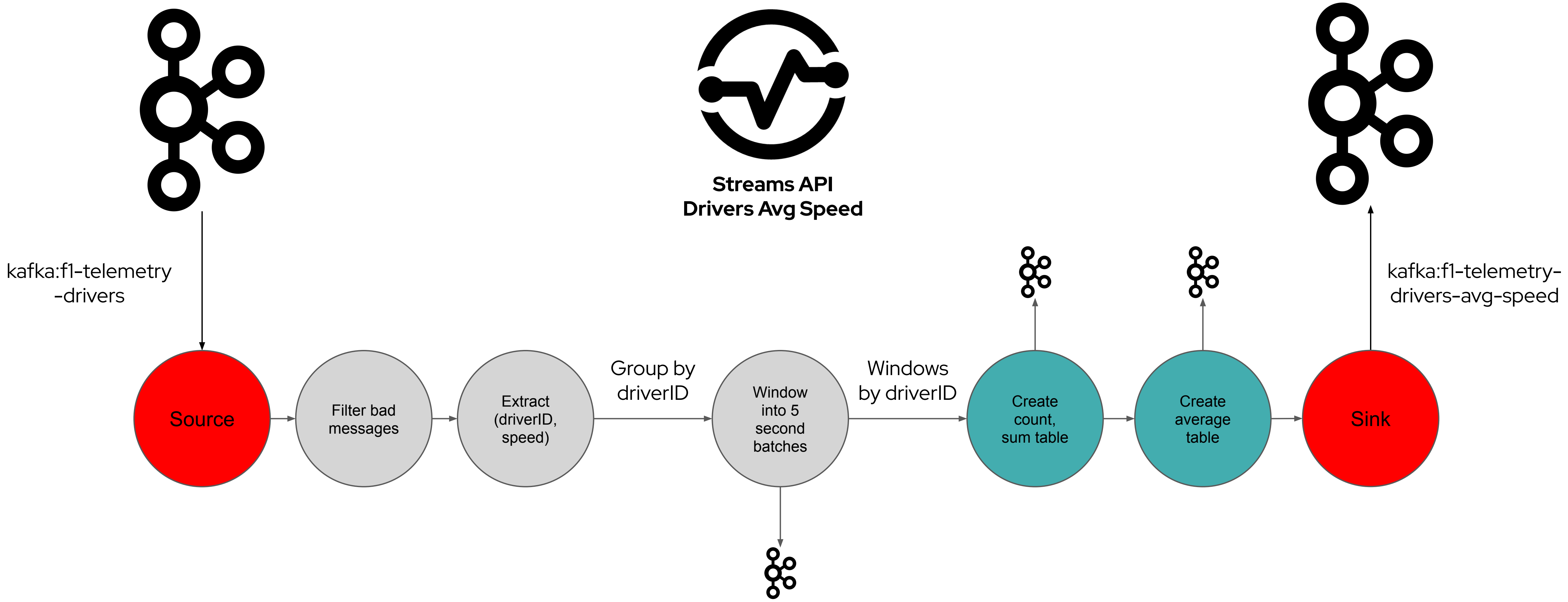
Kafka to InfluxDB



```
from()
.process(telemetry,motion,carstatus,lapdata)
.to()
```

```
from()
.process(avgspeed)
.to()
```

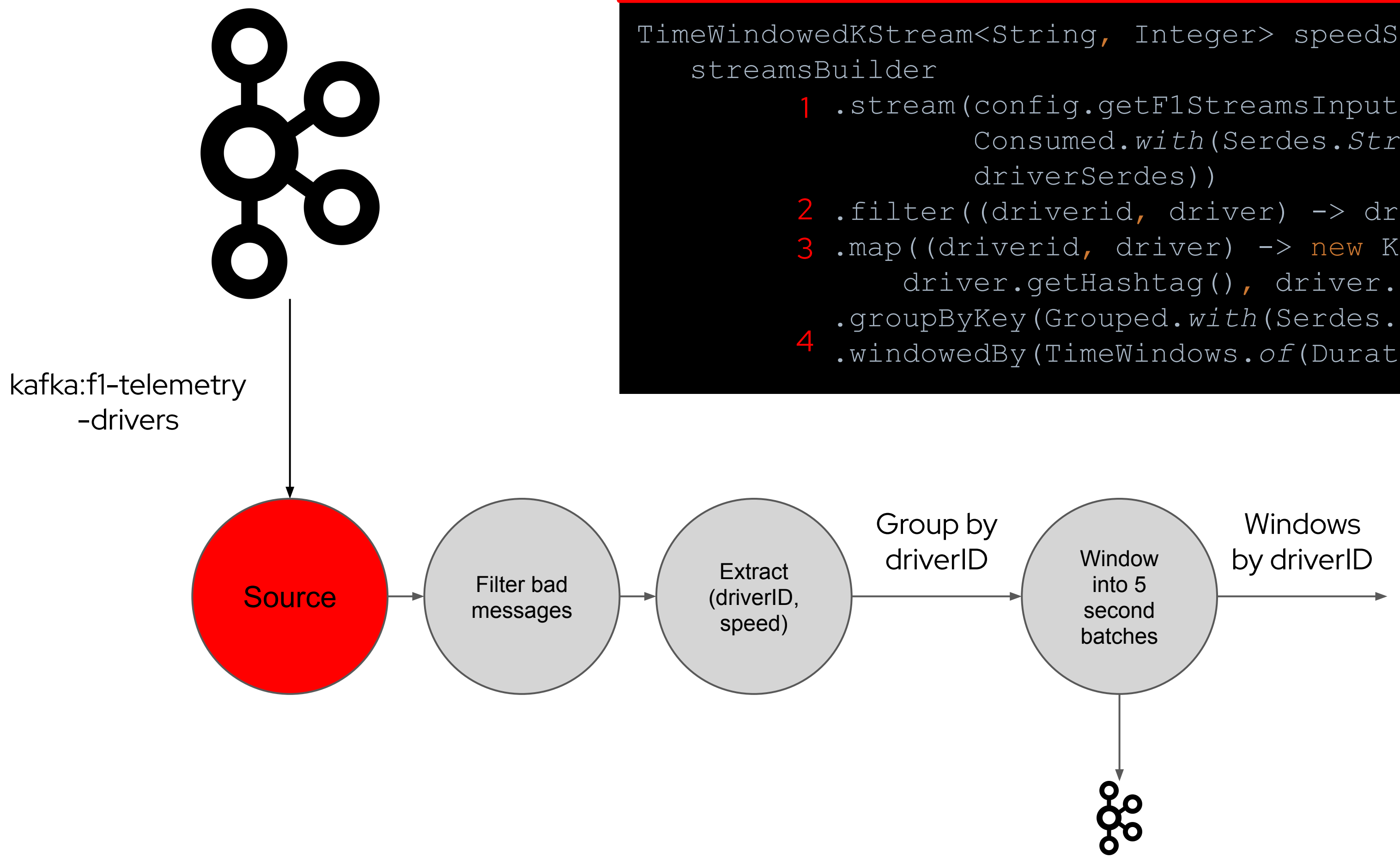
```
from()
.process(fastestlap,speedtrap)
.to()
```


Streams API
Drivers Avg Speed
kafka:f1-telemetry
-drivers
kafka:f1-telemetry-
drivers-avg-speed
Source
Filter bad messages
Extract (driverID, speed)
Group by driverID
Window into 5 second batches
Windows by driverID
Create count, sum table
Create average table
Sink

Setup

```
StreamsBuilder streamsBuilder = new StreamsBuilder();
Serde<Driver> driverSerdes = Serdes.serdeFrom(
    new DriverSerializer(), new DriverDeserializer());
Serde<SpeedCountAndSum> speedCountAndSumSerde = Serdes.serdeFrom(
    new SpeedCountAndSumSerializer(), new SpeedCountAndSumDeserializer());

TimeWindowedKStream<String, Integer> speedStreamWindowed =
   streamsBuilder
         1 .stream(config.getF1StreamsInputTopic(),
                   Consumed.with(Serdes.String(),
                   driverSerdes))
         2 .filter((driverid, driver) -> driver.hasValidTelemetry())
         3 .map((driverid, driver) -> new KeyValue<>(
                driver.getHashtag(), driver.getCarTelemetryData().getSpeed()))
           .groupByKey(Grouped.with(Serdes.String(), Serdes.Integer()))
         4 .windowedBy(TimeWindows.of(Duration.ofMillis(5000)));
```



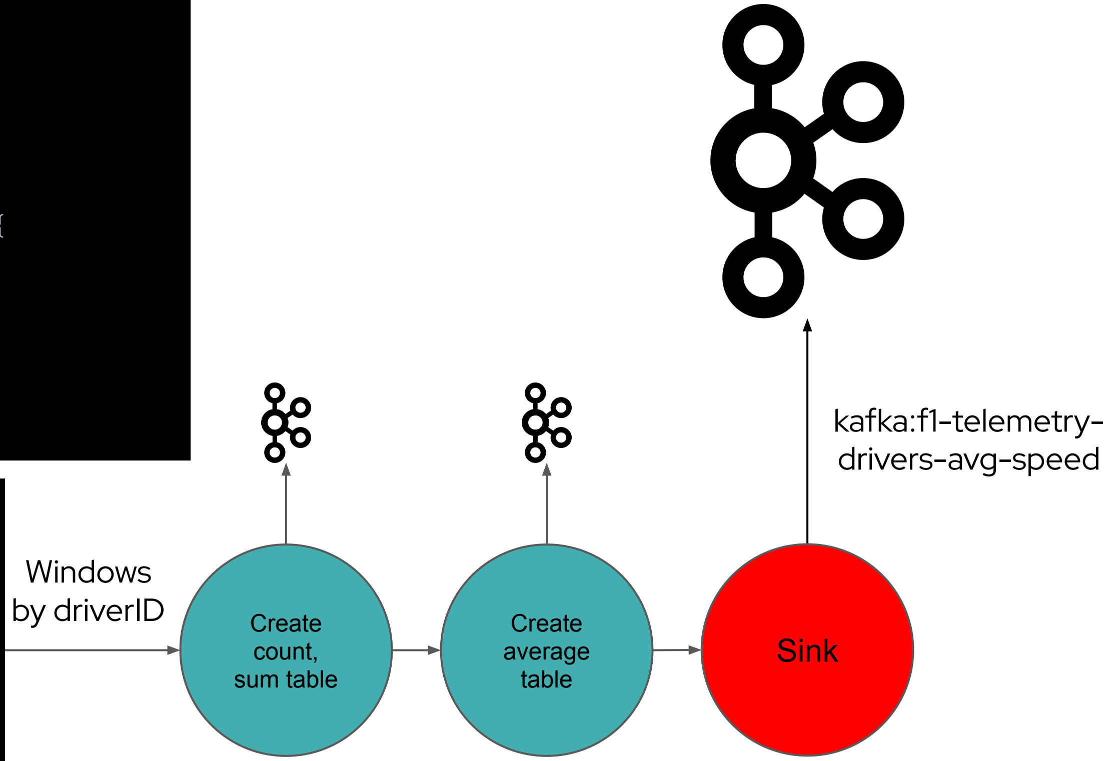
Processing time

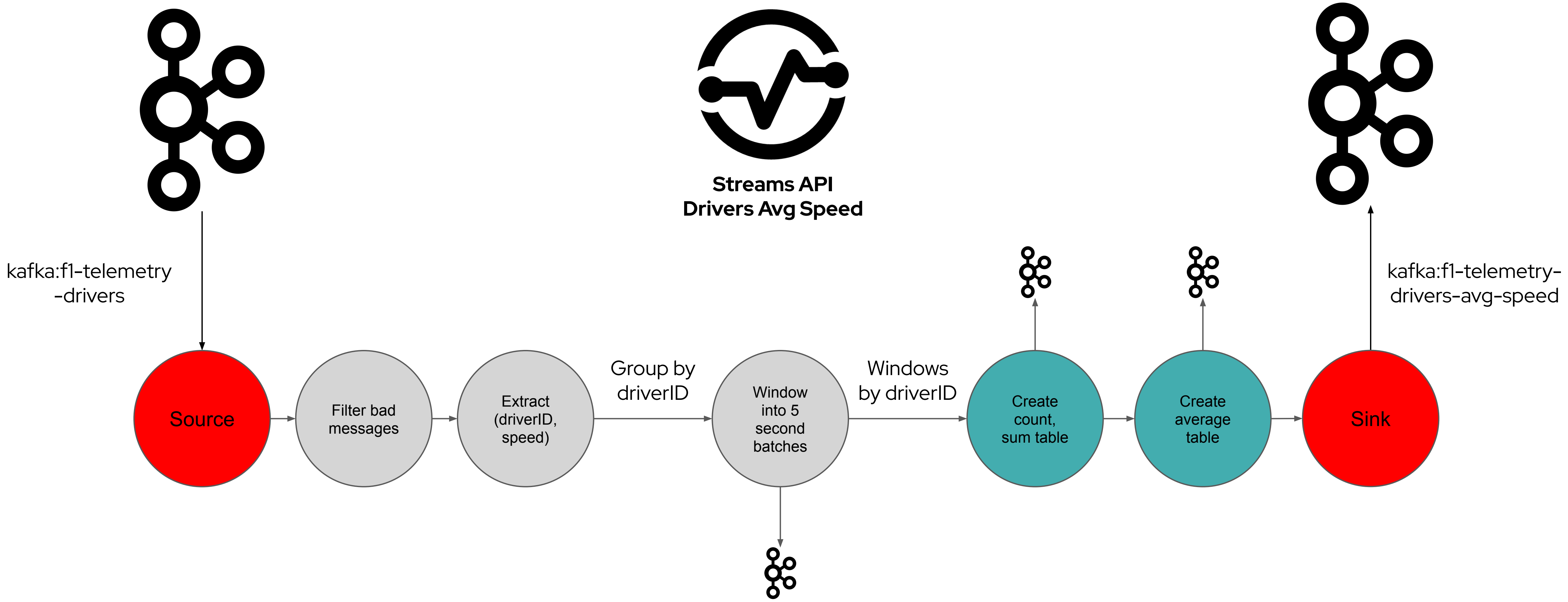
Ingestion time

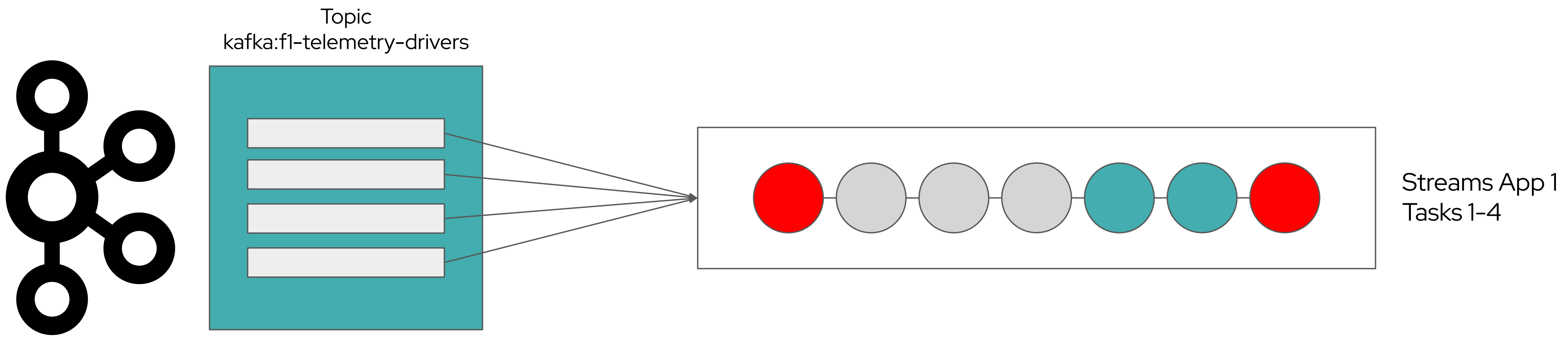
Event time

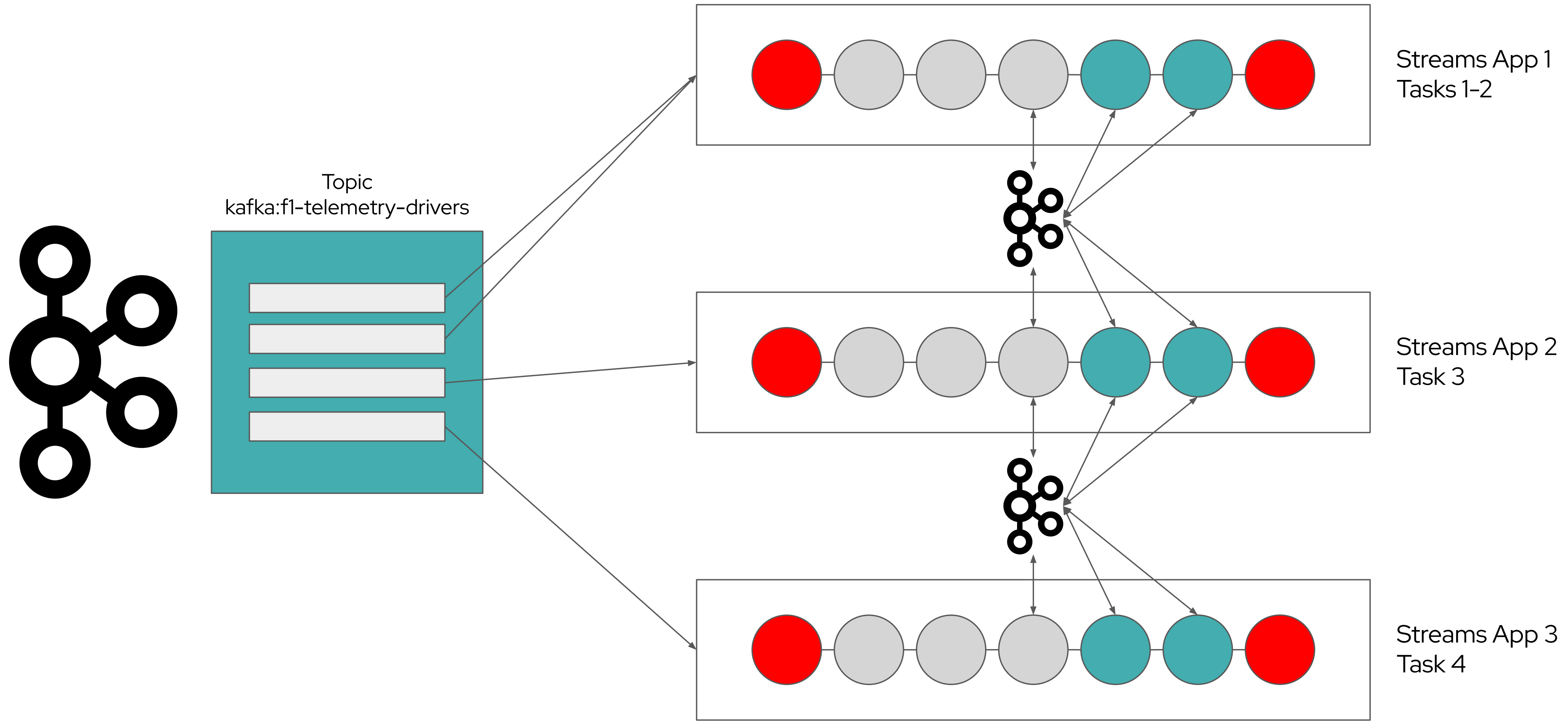
```
KTable<Windowed<String>, SpeedCountAndSum> speedCountAndSum =
    speedStreamWindowed.aggregate(new Initializer<SpeedCountAndSum>() {
        @Override
        public SpeedCountAndSum apply() {
            return new SpeedCountAndSum(0, 0);
        }
    }, new Aggregator<String, Integer, SpeedCountAndSum>() {
        @Override
        public SpeedCountAndSum apply(String key,
                                      Integer value,
                                      SpeedCountAndSum aggregate) {
            aggregate.setCount(aggregate.getCount() + 1);
            aggregate.setSum(aggregate.getSum() + value);
            return aggregate;
        }
    }, Materialized.with(Serdes.String(), speedCountAndSumSerde));
```

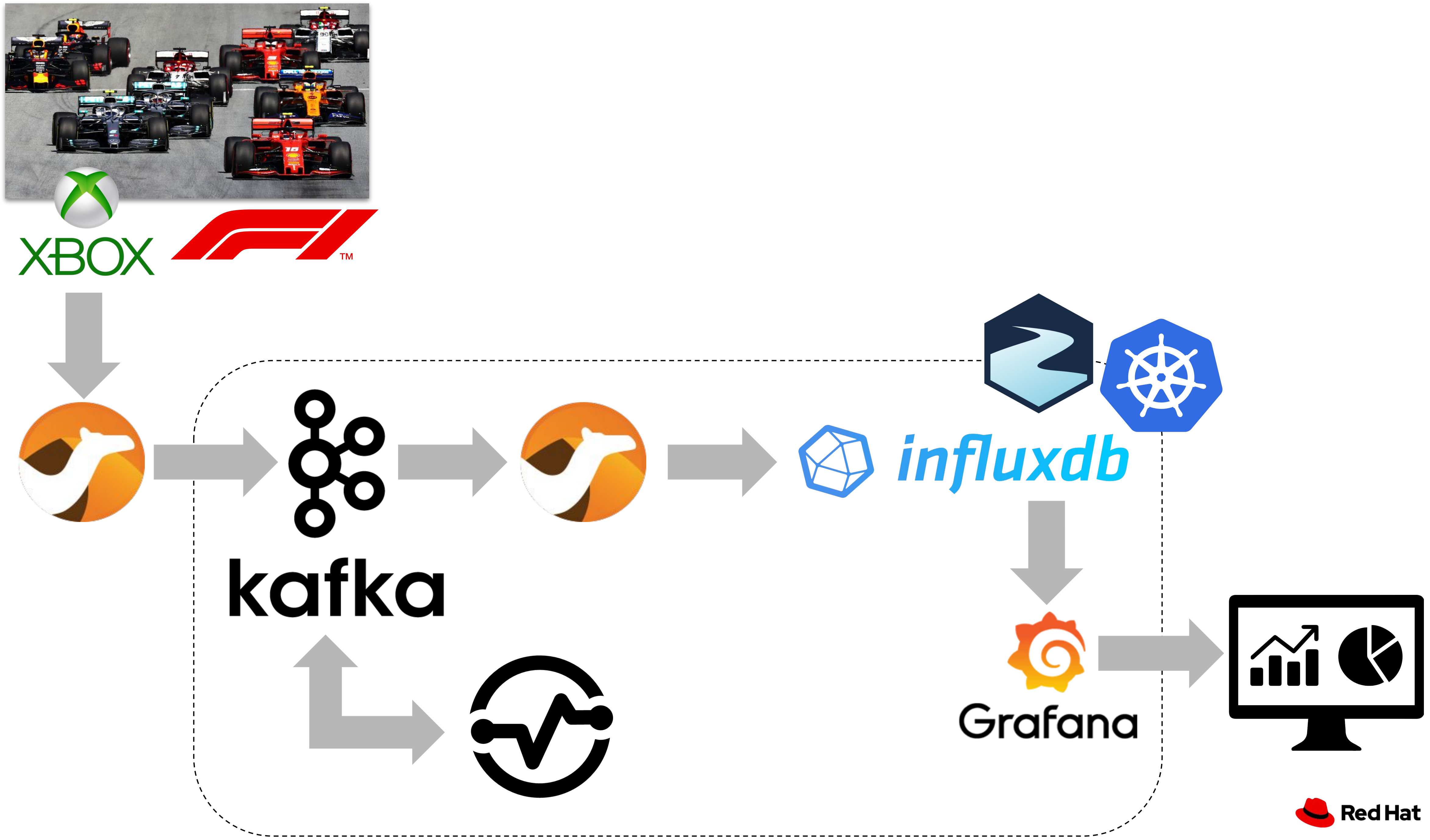
```
KTable<Windowed<String>, Integer> speedAvarage =
    speedCountAndSum.mapValues(
            new ValueMapper<SpeedCountAndSum, Integer>() {
        @Override
        public Integer apply(SpeedCountAndSum speedCountAndSum) {
            return speedCountAndSum.getSum() /
                        speedCountAndSum.getCount();
        }
    });
```




Streams API
Drivers Avg Speed
kafka:f1-telemetry -drivers
Source
Filter bad messages
Extract (driverID, speed)
Group by driverID
Window into 5 second batches
Windows by driverID
Create count, sum table
Create average table
Sink
kafka:f1-telemetry-drivers-avg-speed


Topic
kafka:f1-telemetry-drivers
Streams App 1
Tasks 1-4


Topic
kafka:f1-telemetry-drivers
Streams App 1
Tasks 1-2
Streams App 2
Task 3
Streams App 3
Task 4


XBOX
TM
kafka
influxdb
Grafana

Demo

Resources:

- Blog post: https://grafana.com/blog/2021/02/02/real-time-monitoring-of-formula-1-telemetry-data-on-kubernetes-with-grafana-apache-kafka-and-strimzi/
- F1 decoding library: https://github.com/ppatierno/formula1-telemetry
- F1 Kafka project: https://github.com/ppatierno/formula1-telemetry-kafka
- Video demo: https://www.youtube.com/watch?v=Re9LOAYZi2A

Red Hat

- F1 2020 Codemasters game provides telemetry packets on UDP
  - Specification is available online
- Kubernetes / OpenShift
  - Deploying the Apache Kafka cluster through Strimzi project
  - Running Apache Camel applications, InfluxDB and Grafana
- Apache Camel
  - Ingesting telemetry packets to Apache Kafka
  - Gets telemetry data and race events from Apache Kafka; store into InfluxDB
- InfluxDB
  - Time-series database to provide data to Grafana
- Grafana
  - Showing all telemetry and events on specific dashboards