



Formula 1 telemetry processing using Quarkus and Red Hat OpenShift Streams for Apache Kafka

Paolo Patierno
Principal Software Engineer

Tom Cooper
Principal Software Engineer



@ppatierno

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

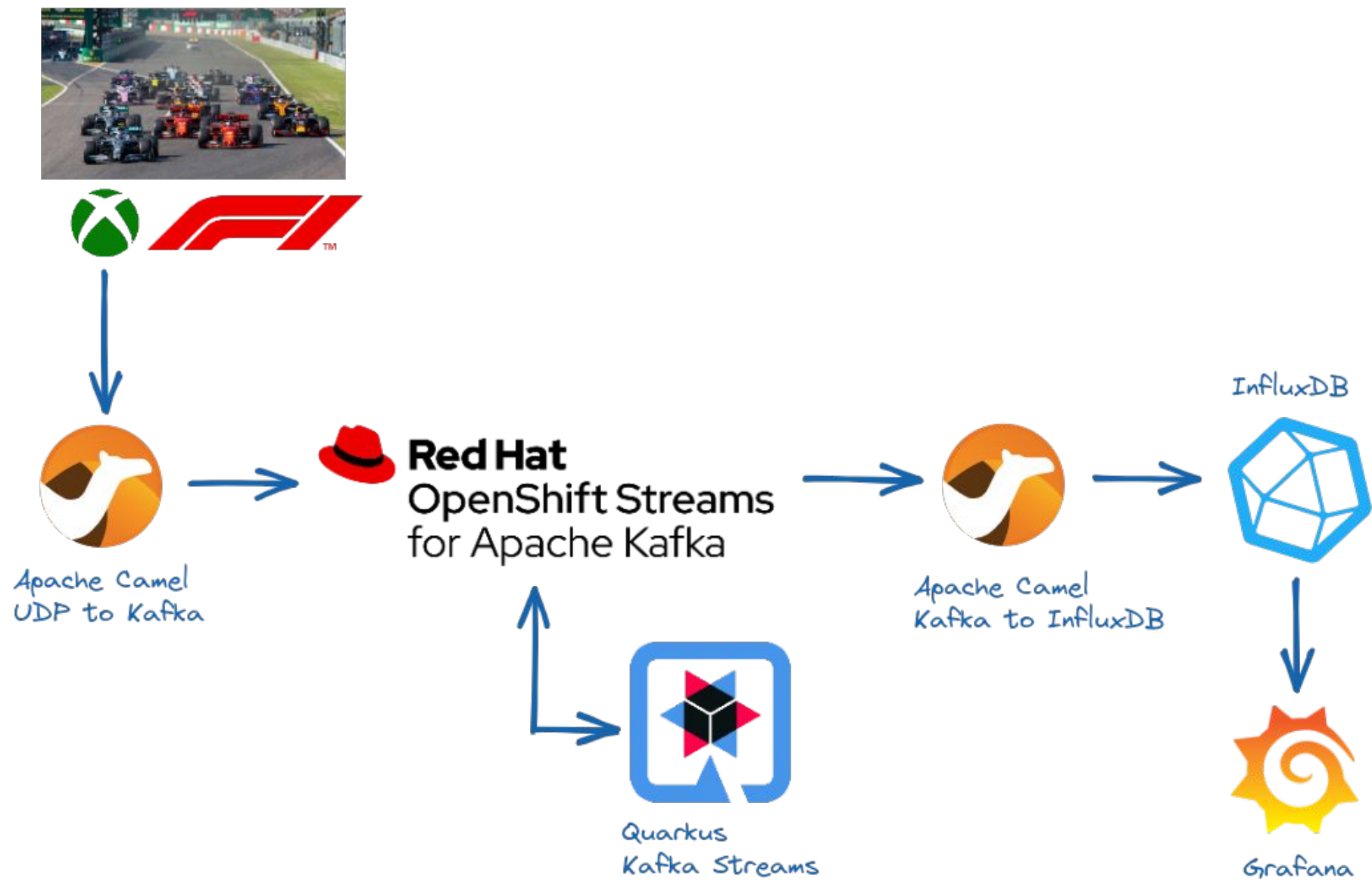
Principal Software Engineer @Red Hat
Working on Apache Kafka and Strimzi

@tomncooper



Building an event streaming 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

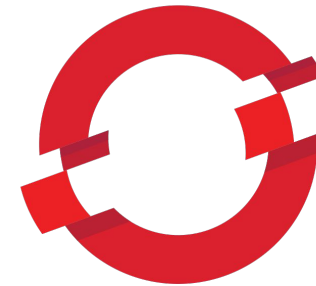




Red Hat OpenShift Streams for Apache Kafka

... RHOSAK for short ...

... powered by



OPENS SHIFT

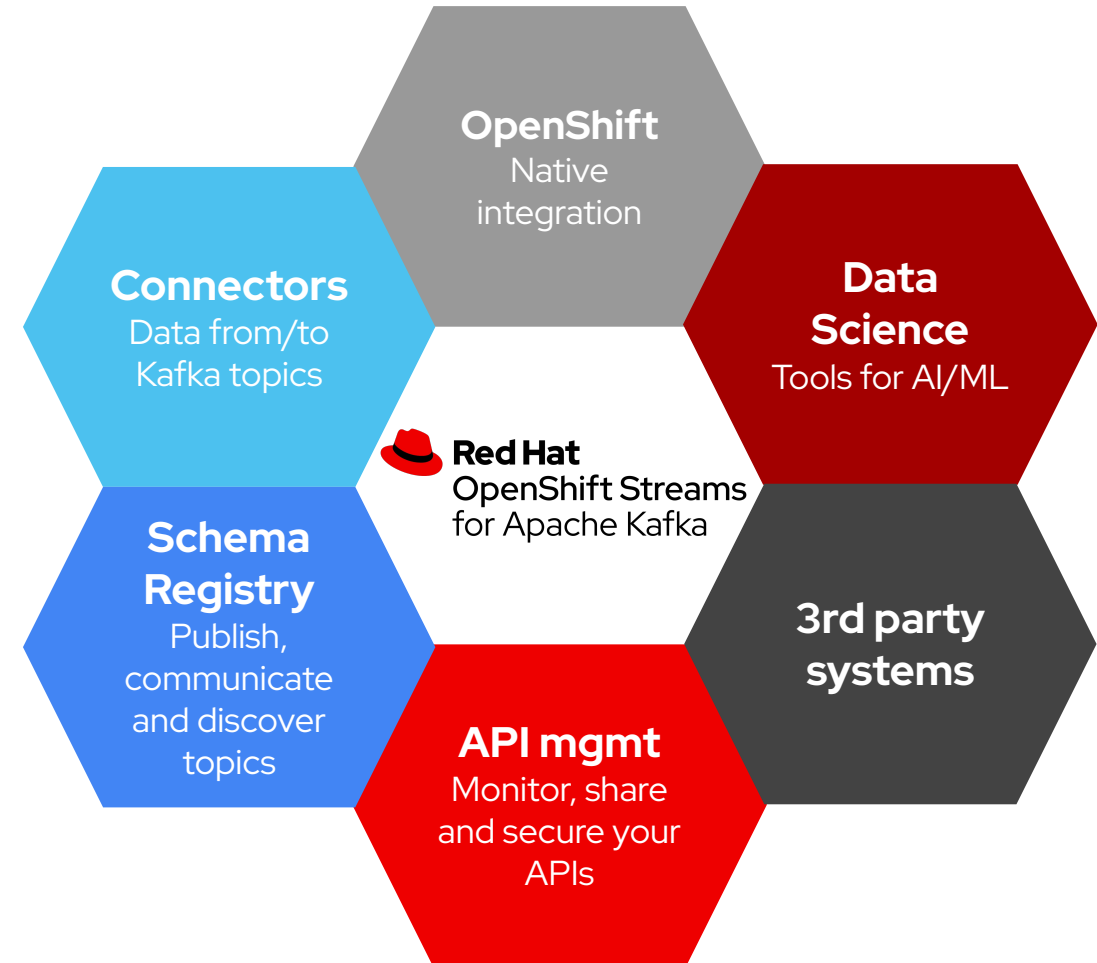


STRIMZI

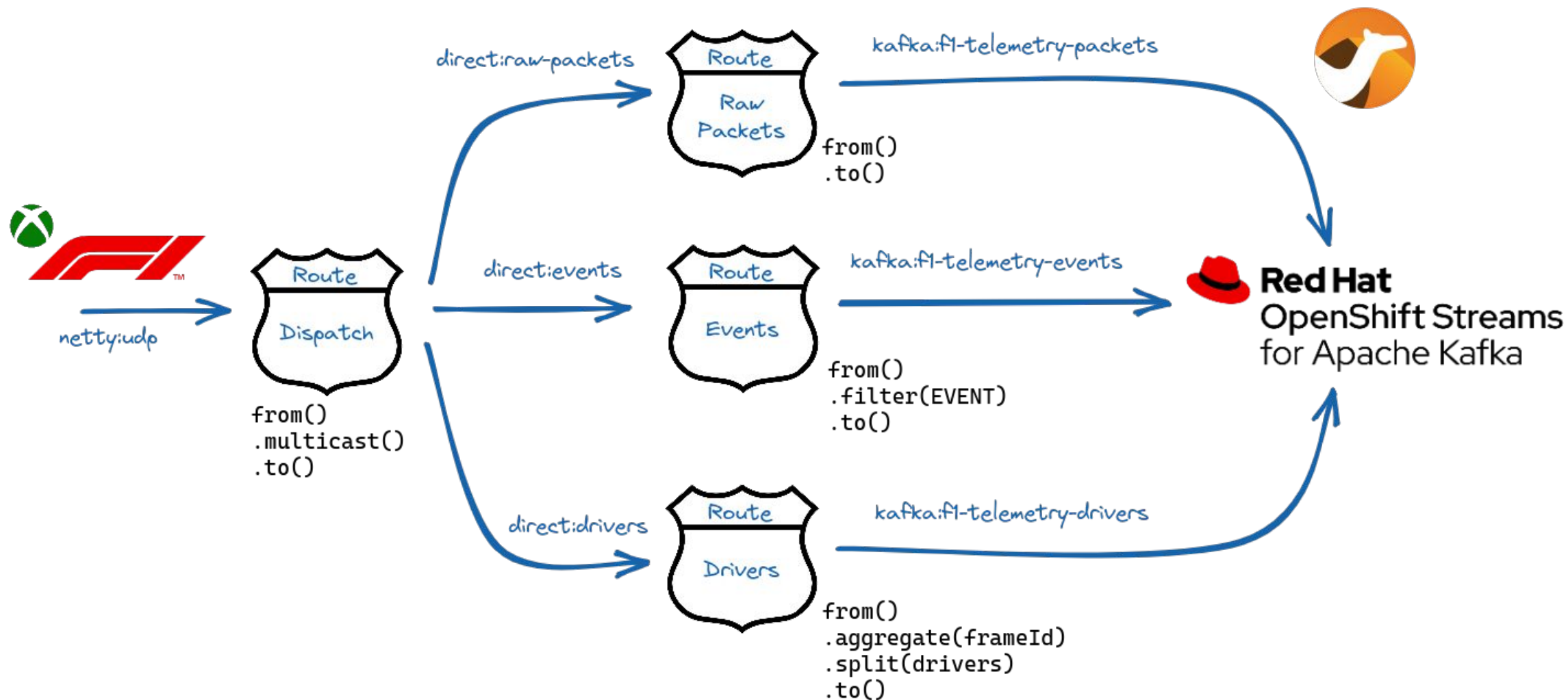


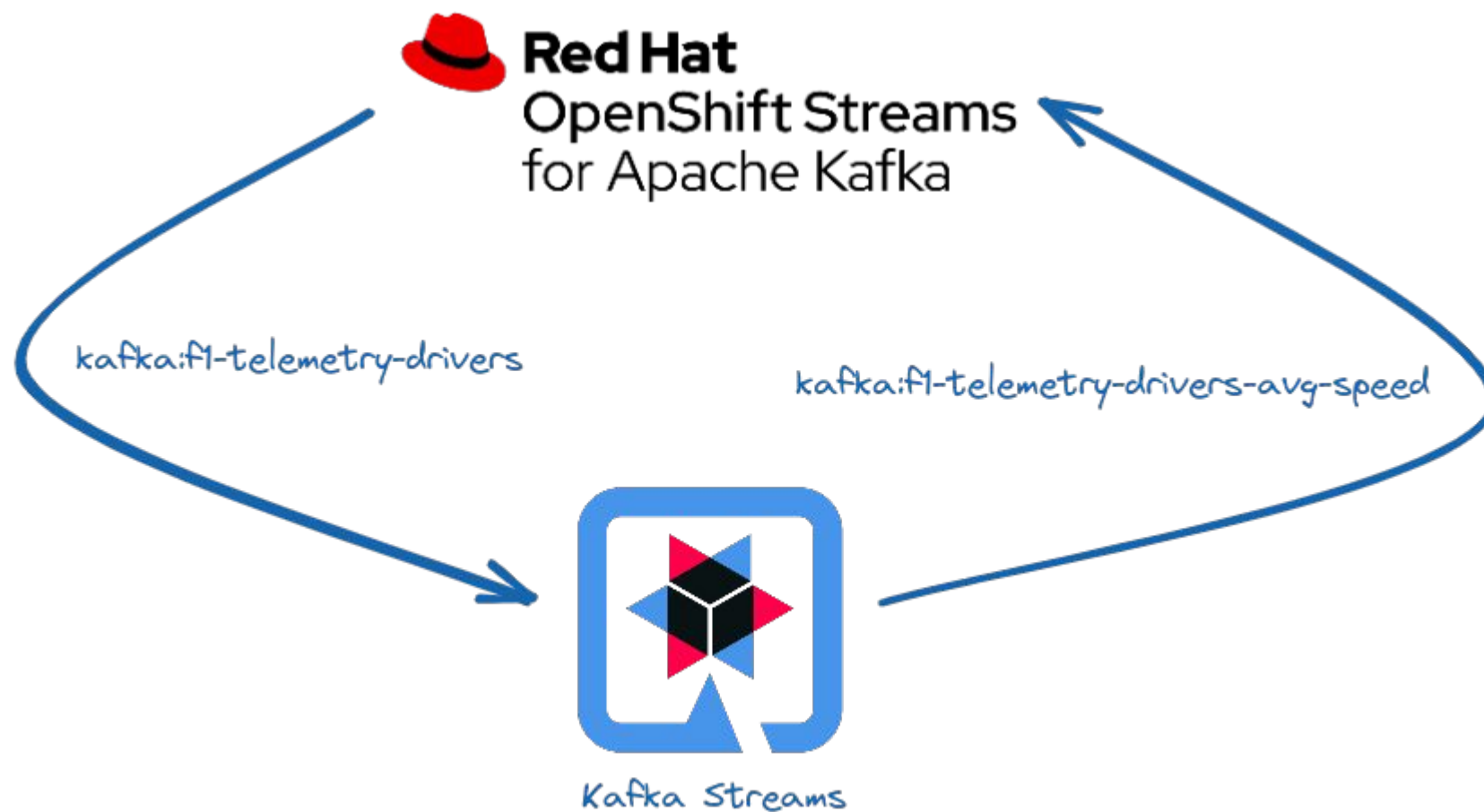
Red Hat

- ▶ Fully hosted and managed Kafka service for stream-based applications
- ▶ Based on Strimzi project to deploy Kafka
- ▶ Reduces ...
 - operational cost and complexity ...
 - ... of delivering real-time applications across hybrid-cloud environments
- ▶ Consumption-based pricing
 - Cluster, storage, data transfer
- ▶ Part of Red Hat Application Services
 - Well integrated within an entire ecosystem



Try it! red.ht/TryKafka

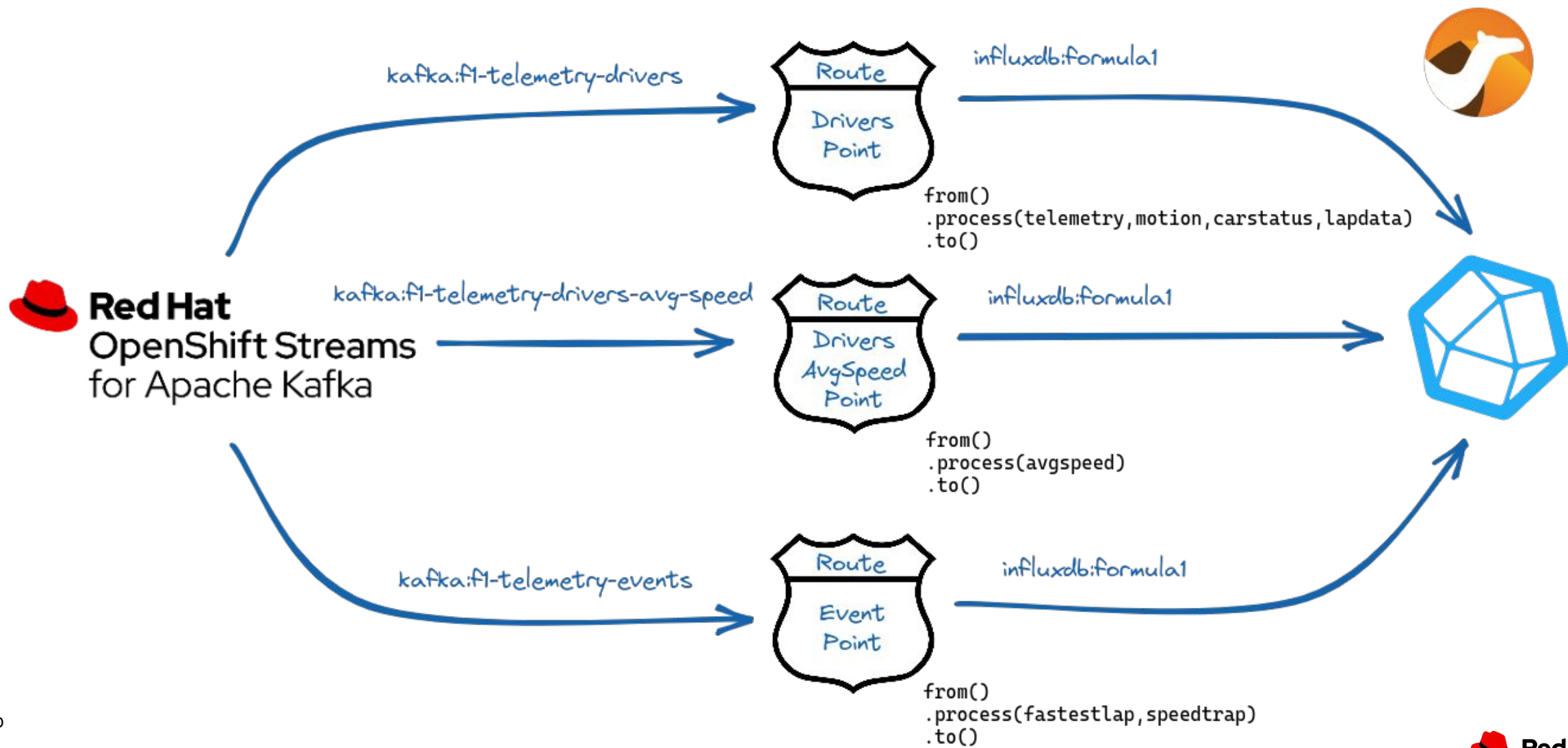




Kafka Streams with Quarkus

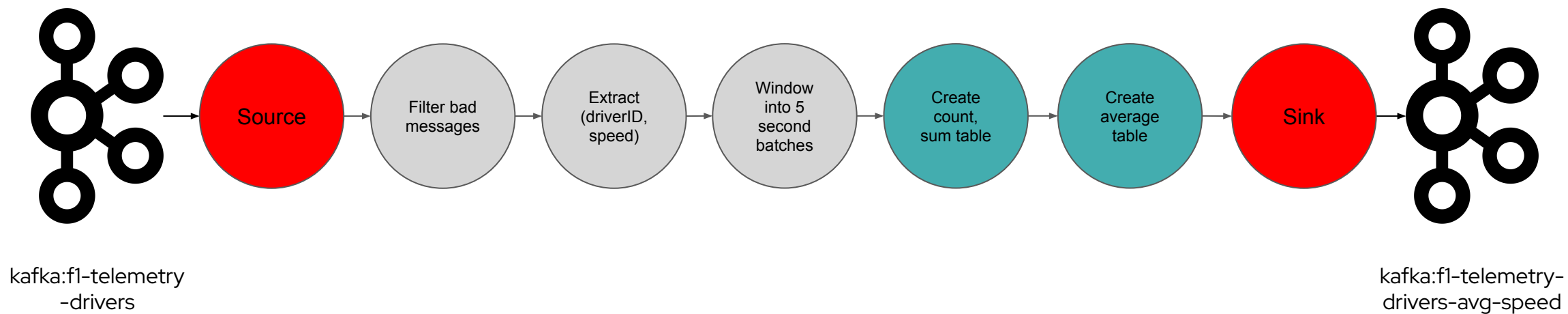
- ▶ A Kubernetes native Java stack
- ▶ Based on the best Java libraries and standards
- ▶ Fast boot-time and low memory usage
- ▶ Native build with GraalVM
- ▶ Both imperative and reactive programming paradigms
- ▶ Focussed on developer joy and rich tooling





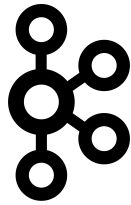


Streams API Drivers Avg Speed





+



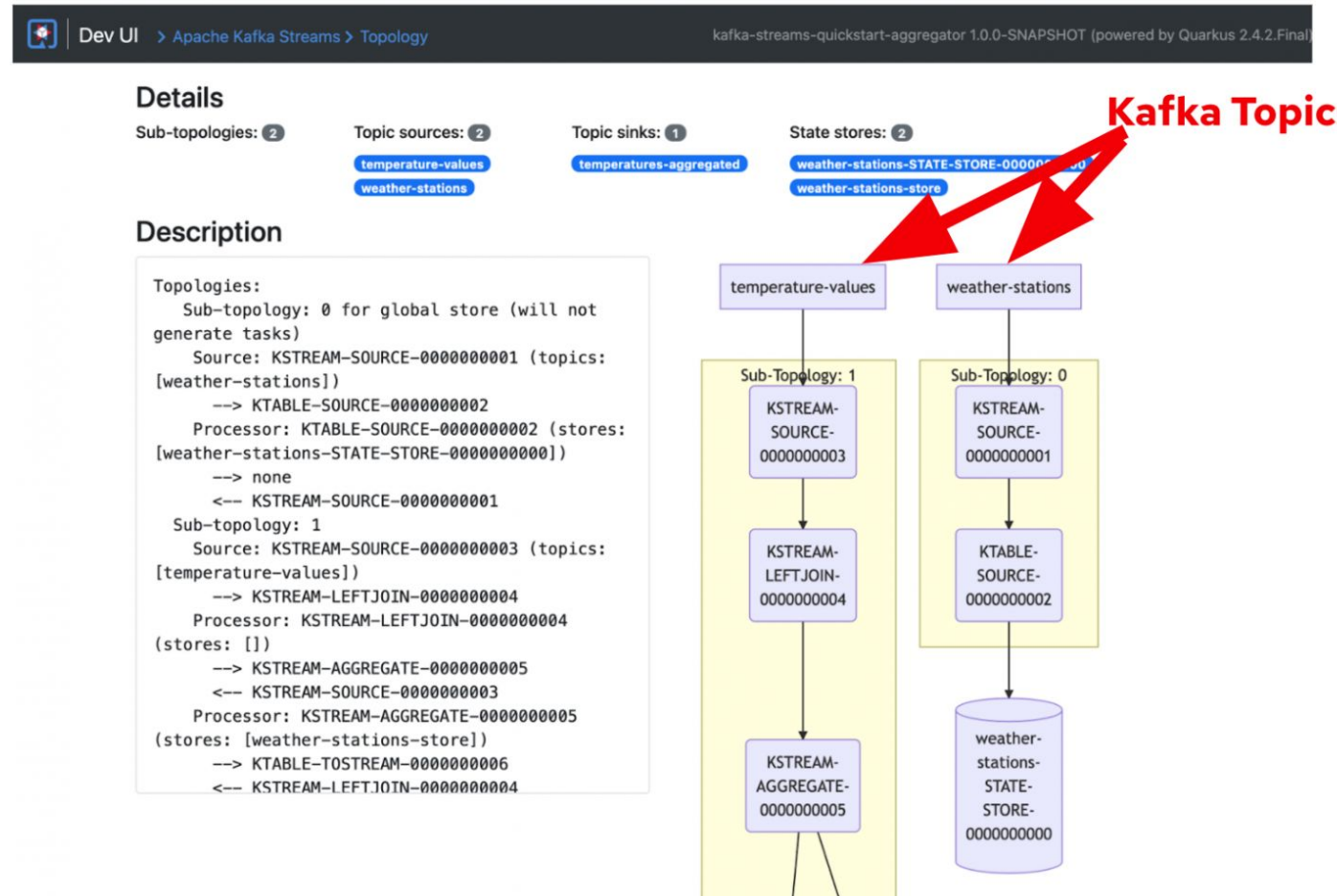
Kafka
Streams

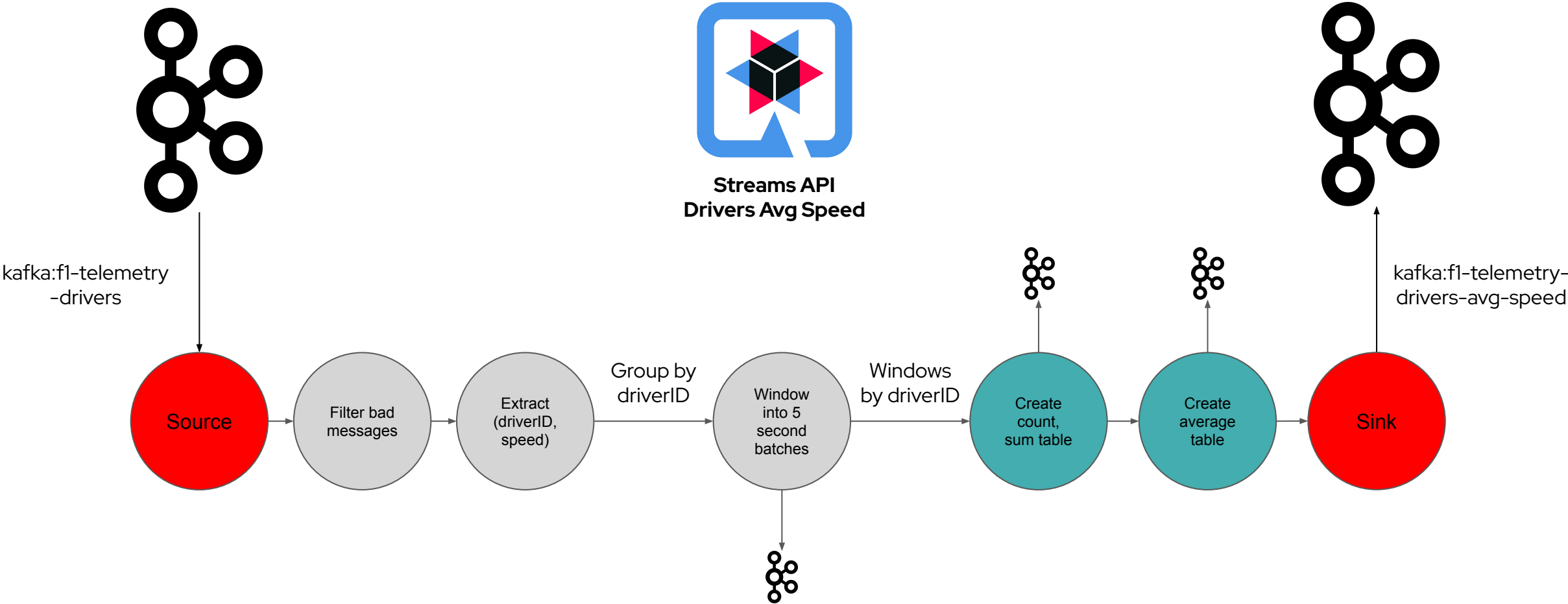
Configure via:

`application.properties`

Test quickly using:

`./mvnw compile quarkus:dev`

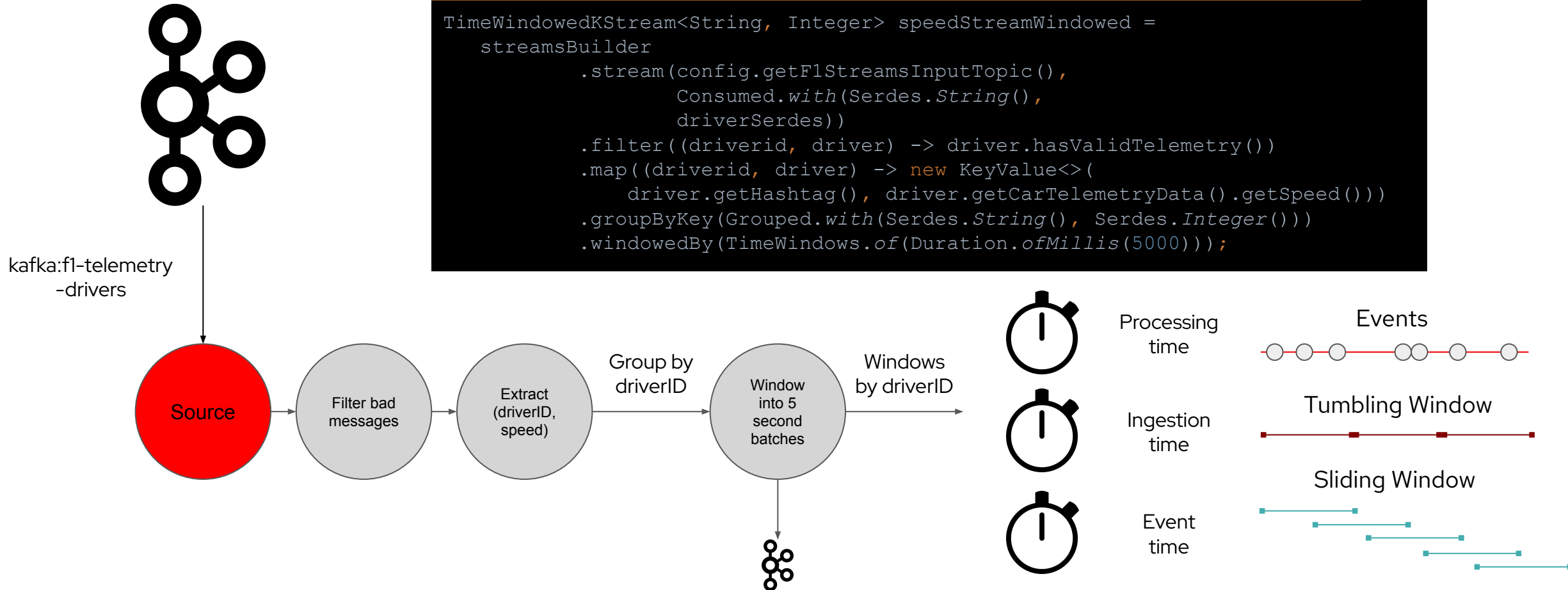




Setup

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

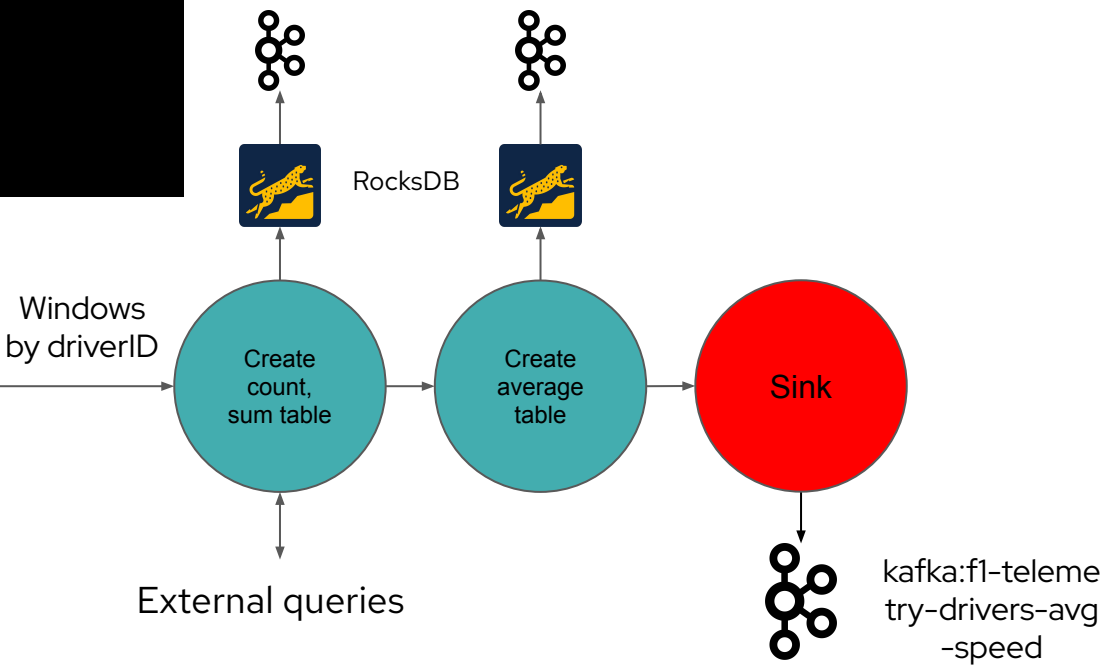
TimeWindowedKStream<String, Integer> speedStreamWindowed =
    streamsBuilder
        .stream(config.getF1StreamsInputTopic(),
            Consumed.with(Serdes.String(),
                driverSerdes))
        .filter((driverid, driver) -> driver.isValidTelemetry())
        .map((driverid, driver) ->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

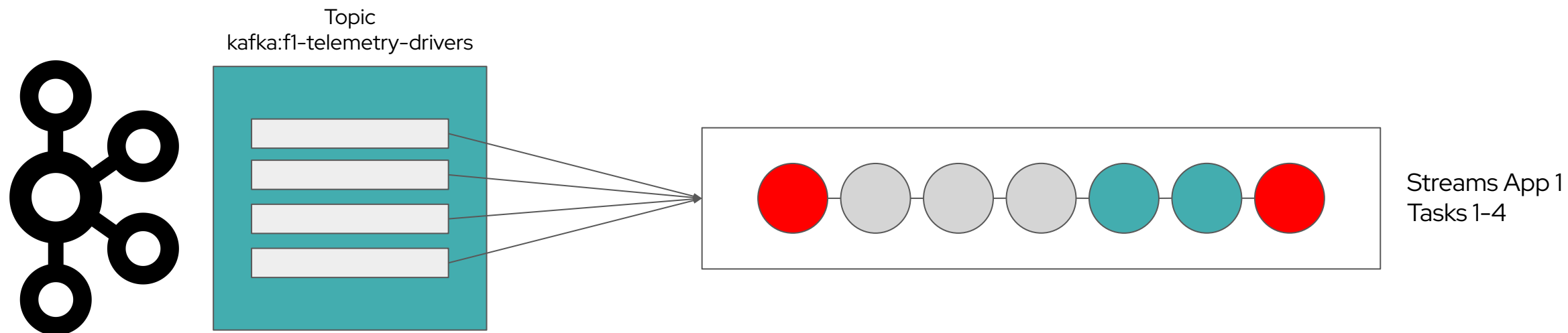


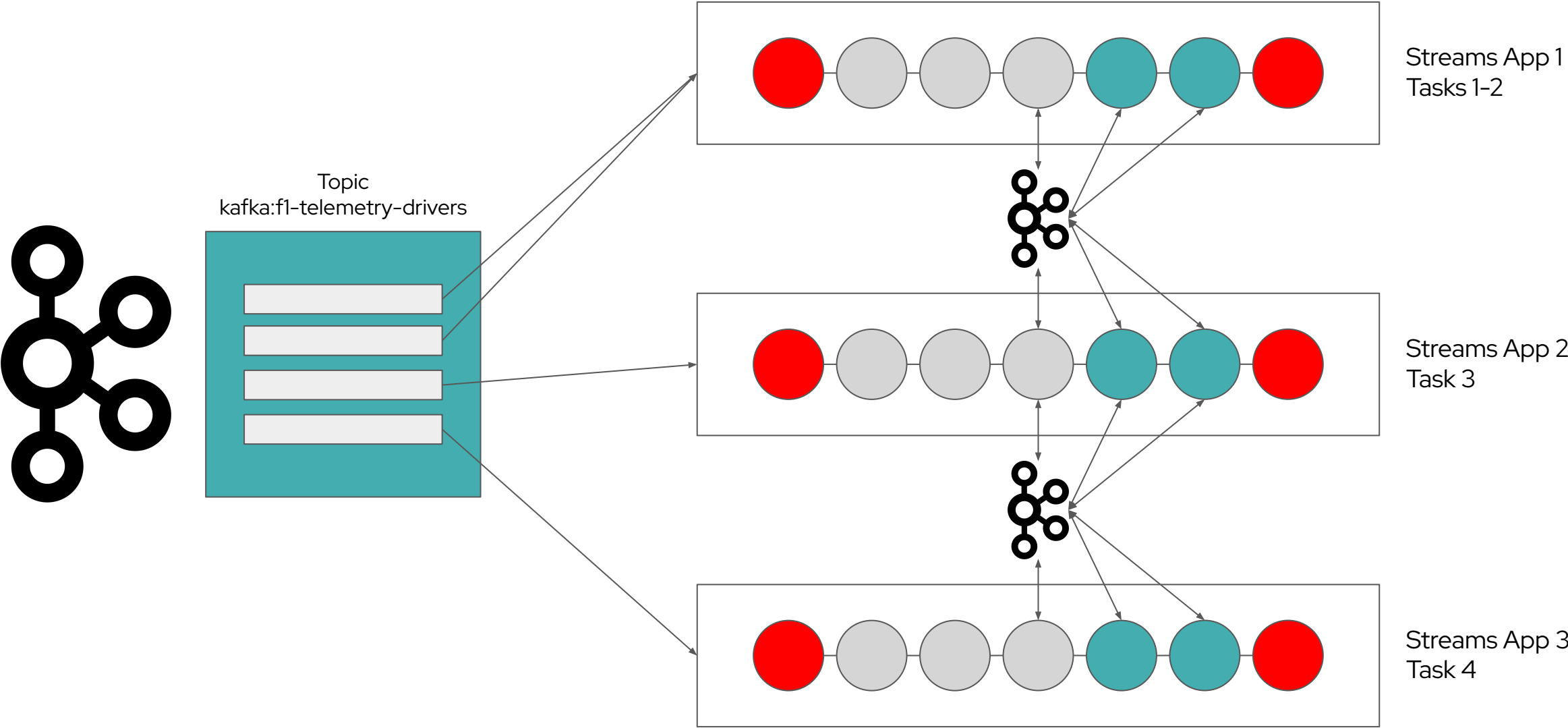
```
KTable<Windowed<String>, SpeedCountAndSum> speedCountAndSum =
    speedStreamWindowed.aggregate(new Initializer<SpeedCountAndSum>() {
        @Override
        public SpeedCountAndSum apply() {
            return new SpeedCountAndSum(0, 0);
        }
    }, new Aggregator<String, Integer, SpeedCountAndSum>() {
        @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();
            }
        }
    );
```

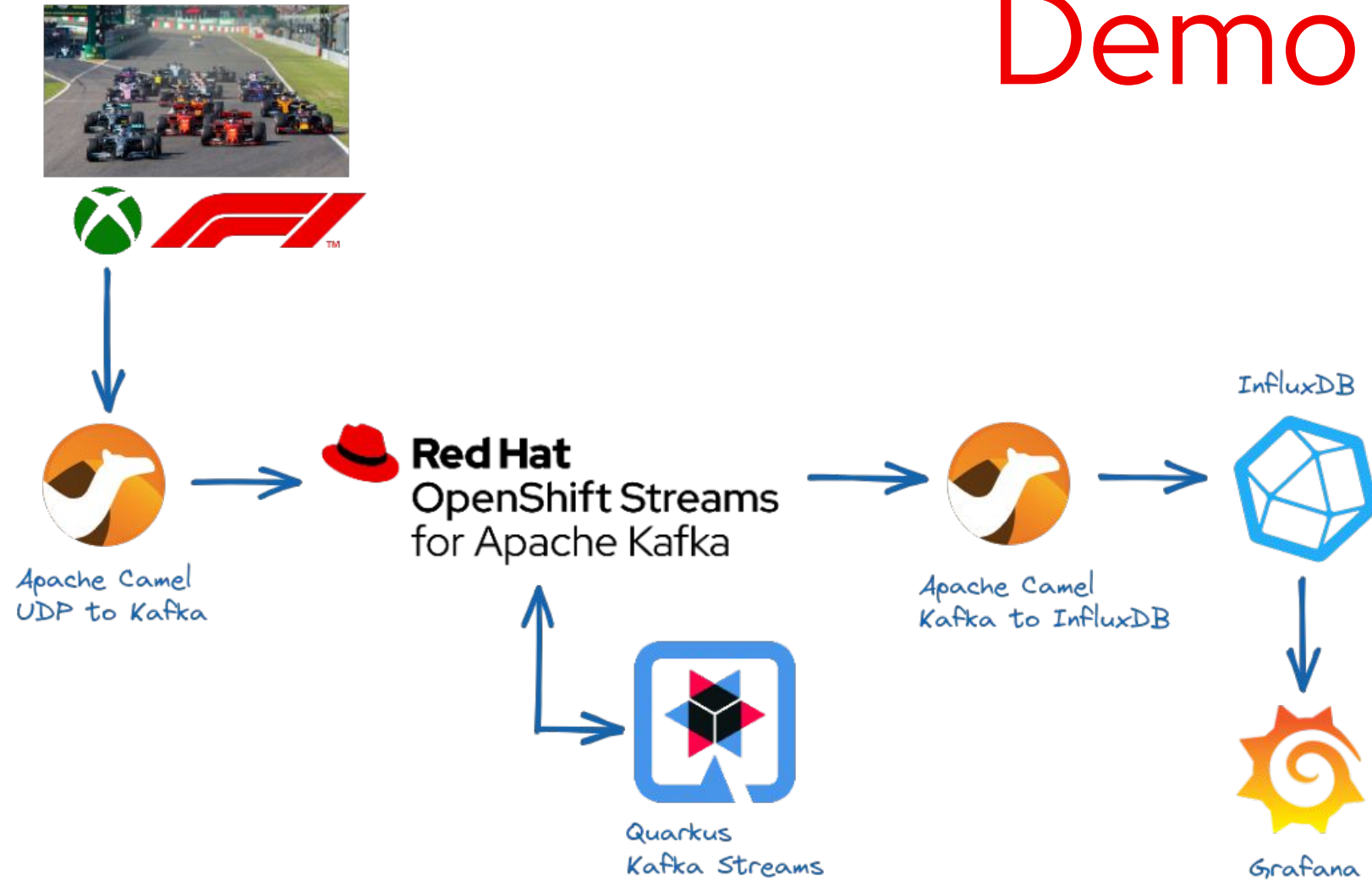
Driver ID	Count	Sum
1234	300	21450.0
5678	278	21322.6
9101	284	22634.8







Demo



Resources

- ▶ [Red Hat OpenShift Streams for Apache Kafka](#)
- ▶ [Quarkus](#)
- ▶ [Apache Kafka](#)
- ▶ [Strimzi](#)
- ▶ [Apache Camel](#)
- ▶ [Formula 1 Telemetry with Apache Kafka](#)
- ▶ [Formula 1 Telemetry with RHOSAK](#)

Red Hat
Summit

Thank you



linkedin.com/company/red-hat



facebook.com/redhatinc



youtube.com/user/RedHatVideos



twitter.com/RedHat