APACHE KAFKA FOR THE HYBRID IOT

Data streaming platform and “traditional” messaging living together

Paolo Patierno
Principal Software Engineer, Messaging & IoT team
21/3/2019

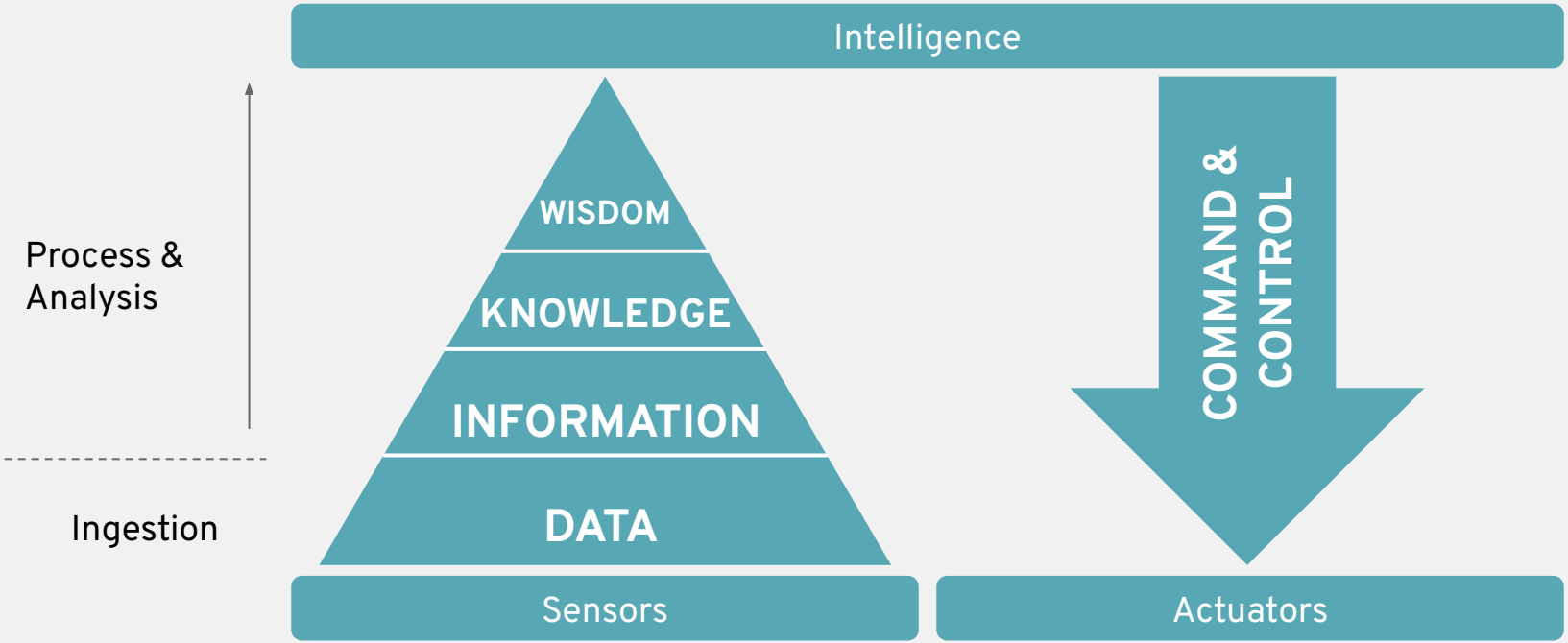
WHO AM I ?

@ppatierno

- Principal Software Engineer @ Red Hat
  - Messaging & IoT team
- Lead/Committer @ Eclipse Foundation
  - Hono, Paho and Vert.x projects
- Microsoft MVP Azure/IoT
- Hacking embedded/IoT devices in a previous life
- Dad of two, husband of one
- I love running, swimming, MotoGP, #VR46, ssc Napoli

IOT: FROM THE SENSORS AND BACK

From raw data to the intelligence … and back



redhat.

IOT WORKLOAD

- Huge amount of data to ingest
- Unbounded dataset
- Need for real time analytics
- Need for real time reactions
- Sometimes the need to join :
    - with static data
    - with historical data

STREAM PROCESSING
A NEW PARADIGM FOR DATA PROCESSING

STREAM PROCESSING

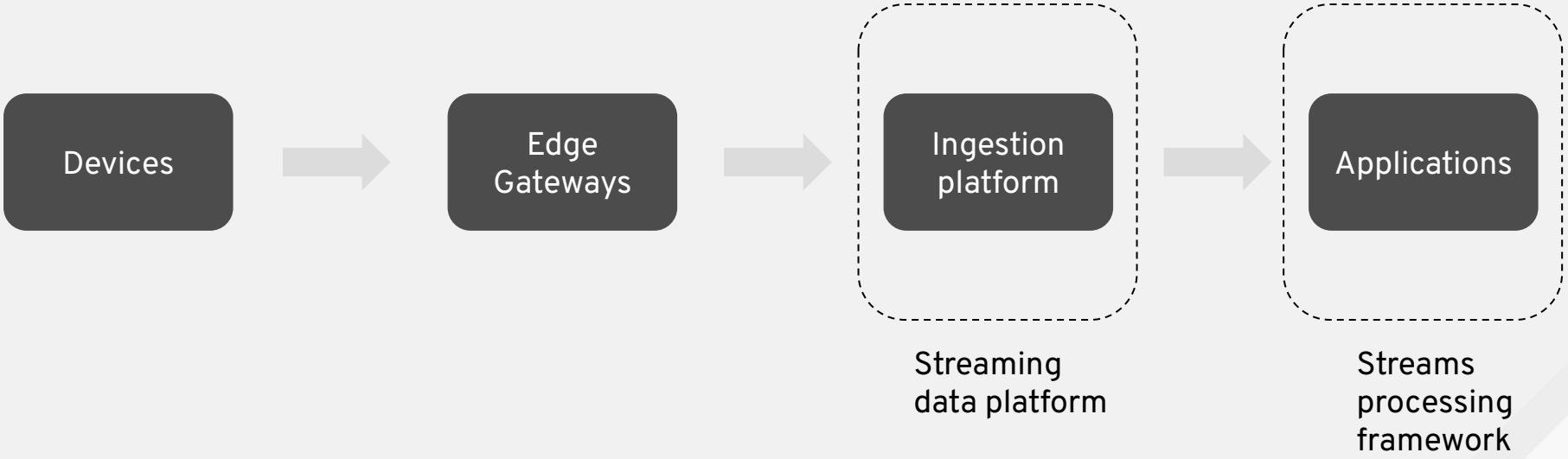
- A type of data processing engine that is designed with infinite datasets in mind
- Working with data as they arrive
- Working with an infinite stream of data
    - Moving boundaries with data in “windows”
- Tradeoffs between latency/cost/correctness

STREAM PROCESSING & IOT

What are the good parts for IoT?

- Timing
  - Supporting “windowing” for getting analytics in a specific time window
  - Using “event time” (vs “processing time”) which makes much sense in IoT
  - Handling “out of order” data when the device is not able to guarantee order
- State management
  - Be fast on joining real time data with metadata to enrich information content
  - Be fast on data aggregation
  - Be able to join different streams of data, from different devices
- Re-processing
  - Supporting the possibility to re-read the stream of data
- Scalability/Partitioning
  - Handle data from more devices and burst of traffic

STREAM PROCESSING & IOT



redhat.

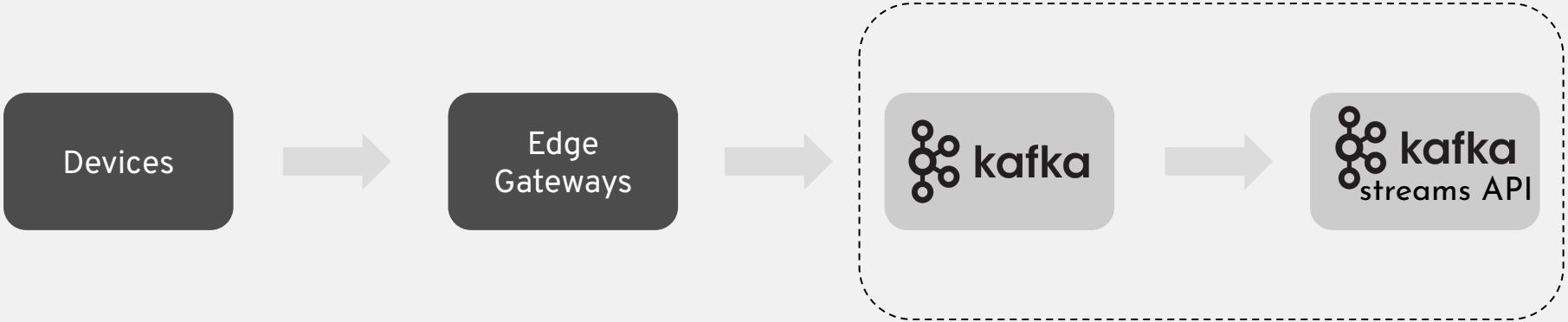
STREAM PROCESSING & IOT

It's a wild west out there

- Streaming data platform
  - Apache Kafka
- Streams processing framework
  - Apache Spark (streaming)
  - Apache Samza
  - Apache Flink
  - …

INGESTION + PROCESSING, DO WE REALLY NEED MORE THAN ONE PLATFORM IN PLACE?

STREAM PROCESSING & IOT



Apache Kafka + Kafka Streams API

APACHE KAFKA

What is that?

“ … a publish/subscribe messaging system …”

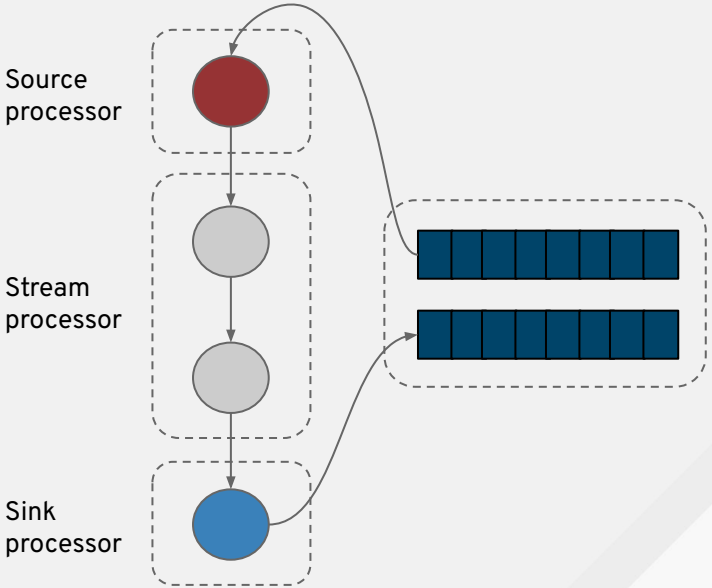
“ … a streaming data platform …”

“ … a distributed, horizontally-scalable, fault-tolerant, commit log …”

APACHE KAFKA

The Streams API

- Stream processing framework, just a Java lib!
- Streams are Kafka topics (as input and output)
- Scaling the stream application horizontally
- Creates a topology of processing nodes (filter, map, join etc) acting on a stream
    - Low level processor API
    - High level DSL
    - Using “internal” topics (when re-partitioning is needed or for “stateful” transformations)



THEN, IT COMES TO HANDLING THE FLOW TO DEVICES FOR CONTROL

DIFFERENT NEEDS

… compared to the data ingestion

- Correlation …
  - … between the command and the related result
- Commands executed just one time …
  - … no need for re-playing a commands “history”
- Most of the times no need for storing …
  - … but executing commands only if the device in online
- Embedded devices are mostly low power …
  - …. and IoT/messaging protocols fit better

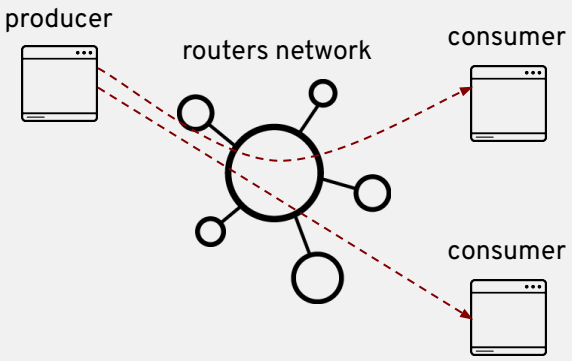
“TRADITIONAL” MESSAGING

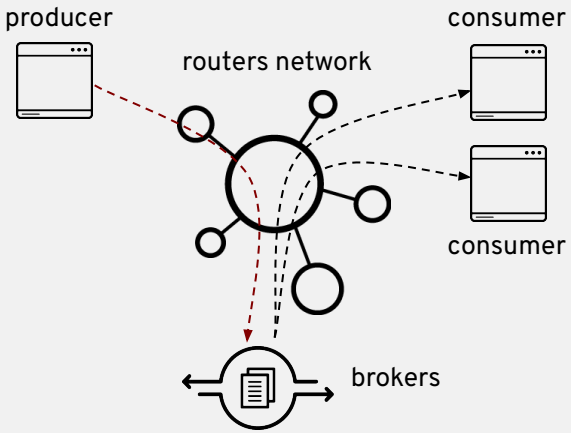
… with AMQP 1.0

- Allows to handle request/reply pattern
  - Sending a command, receiving the execution status/result
- Most of the times, storing isn’t really needed
  - The command has to be executed now, not later…
  - … or using storage but with a TTL on the message command …
  - … and you can also check the status of delivery and cancel
- Filtering messages
  - Getting only specific messages based on headers values

“TRADITIONAL” MESSAGING

Direct vs Store-and-forward





- Only if the device is connected
- The “sender” always knows that message has reached the final destination

- Even if the device isn’t connected; TTL helps for stale messages
- The “senders” only knows that message has reached the intermediary

LET'S PUT THEM TOGETHER TO CREATE THE "HYBRID" IOT

MOVING THE IOT INTO THE “FOG”

The role of an IoT gateway

- The protocols Babel tower
  - Protocol translation (BLE, ZigBee, MQTT, AMQP 1.0, HTTP, …)
- Reducing cloud workload
  - Offload part of the processing at the edge
- Real time reaction
  - Avoiding the cloud latency
- Offline handling
  - Connection isn’t reliable or the bandwidth is low
- Security and privacy
  - Keep data at the edge, avoiding the cloud
- Reducing price
  - Paying for less devices connected, less data transferred

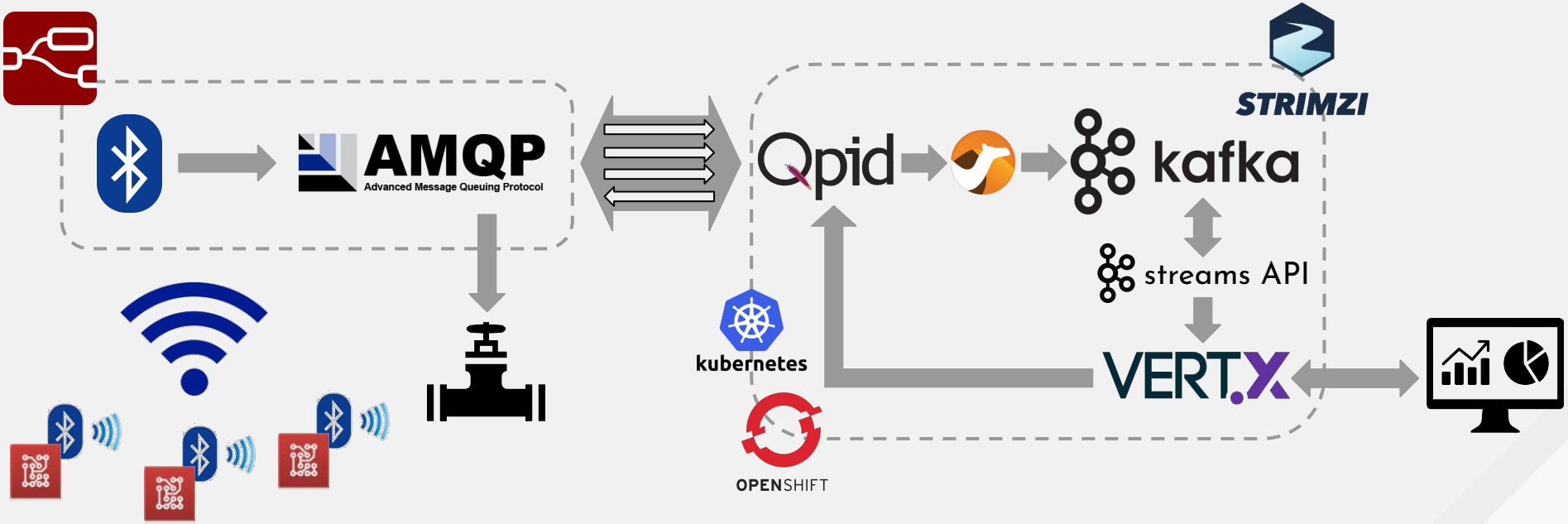
AMQP 1.0 ON THE GATEWAY

AMQP
Advanced Message Queuing Protocol

- Kafka protocol needs more connections to all brokers
  - The IoT gateway has to provide all these connection
  - Exposing all the brokers can be cumbersome
  - Request/reply doesn't fit so well
- With AMQP 1.0 …
  - … the IoT gateway can have just one connection to the cloud
  - … a dispatch router can be used to handle the scalability and failures
  - … within one connection, different sensors traffic can be multiplexed with “links”
  - … within one connection, different “sessions” (and “links”) for ingestion and control

DEMO

An “hybrid” IoT pipeline



redhat.

THANK YOU!

@ppatierno

redhat.