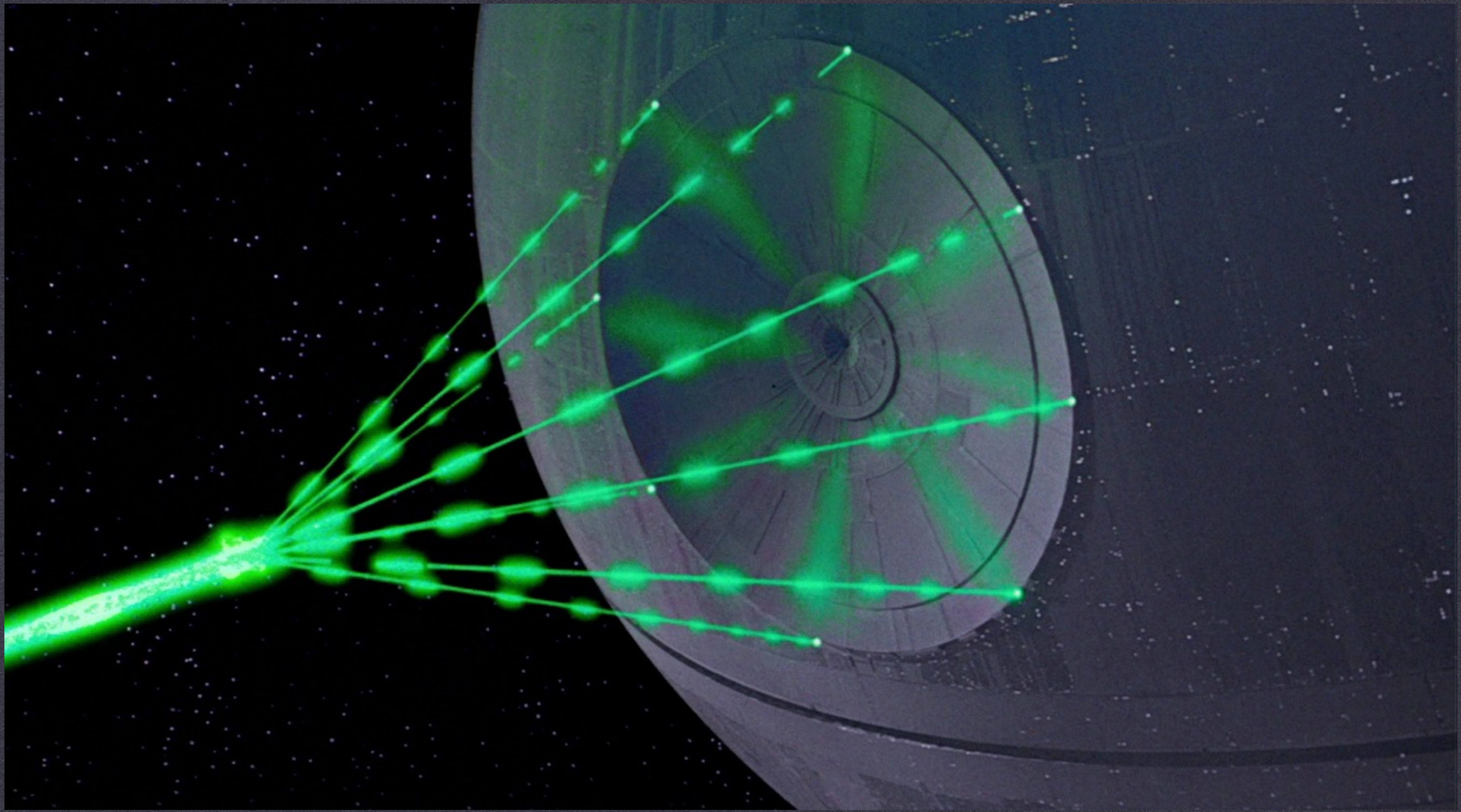


SerIoTics

Serialization metrics in an Internet of Things environment

SHAUN DONACHY

```
{“minType":"BATHROOM",
“majType”:"HUMANITARIAN,
“major_area_num”:19,
“minor_area_num":436,
“quantity”:3.7826,
“unique_id":1000,
“runtime":1443200834318,
“item_sensed”:"toilet",
“subject_measured”:"water",
“sensor_location_name”:"bathroom",
"ticks_since_turn_on":2}
```

```
{“minType":"BATHROOM",
“majType”:"HUMANITARIAN,
“major_area_num”:19,
“minor_area_num":436,
“quantity”:3.7826,
“unique_id":1000,
“runtime":1443200834318,
“item_sensed”:"toilet",
“subject_measured”:"water",
“sensor_location_name”:"bathroom",
"ticks_since_turn_on":2}
```

JSON message
~ 265 bytes

```
{“minType":"BATHROOM",
“majType”:"HUMANITARIAN,
“major_area_num”:19,
“minor_area_num":436,
“quantity”:3.7826,
“unique_id":1000,
“runtime":1443200834318,
“item_sensed”:"toilet",
“subject_measured”:"water",
“sensor_location_name”:"bathroom",
"ticks_since_turn_on":2}
```

JSON message
~ 265 bytes

ProtoBuf Byte Array
~ 69 bytes

Avro Byte Array
~ 58 bytes

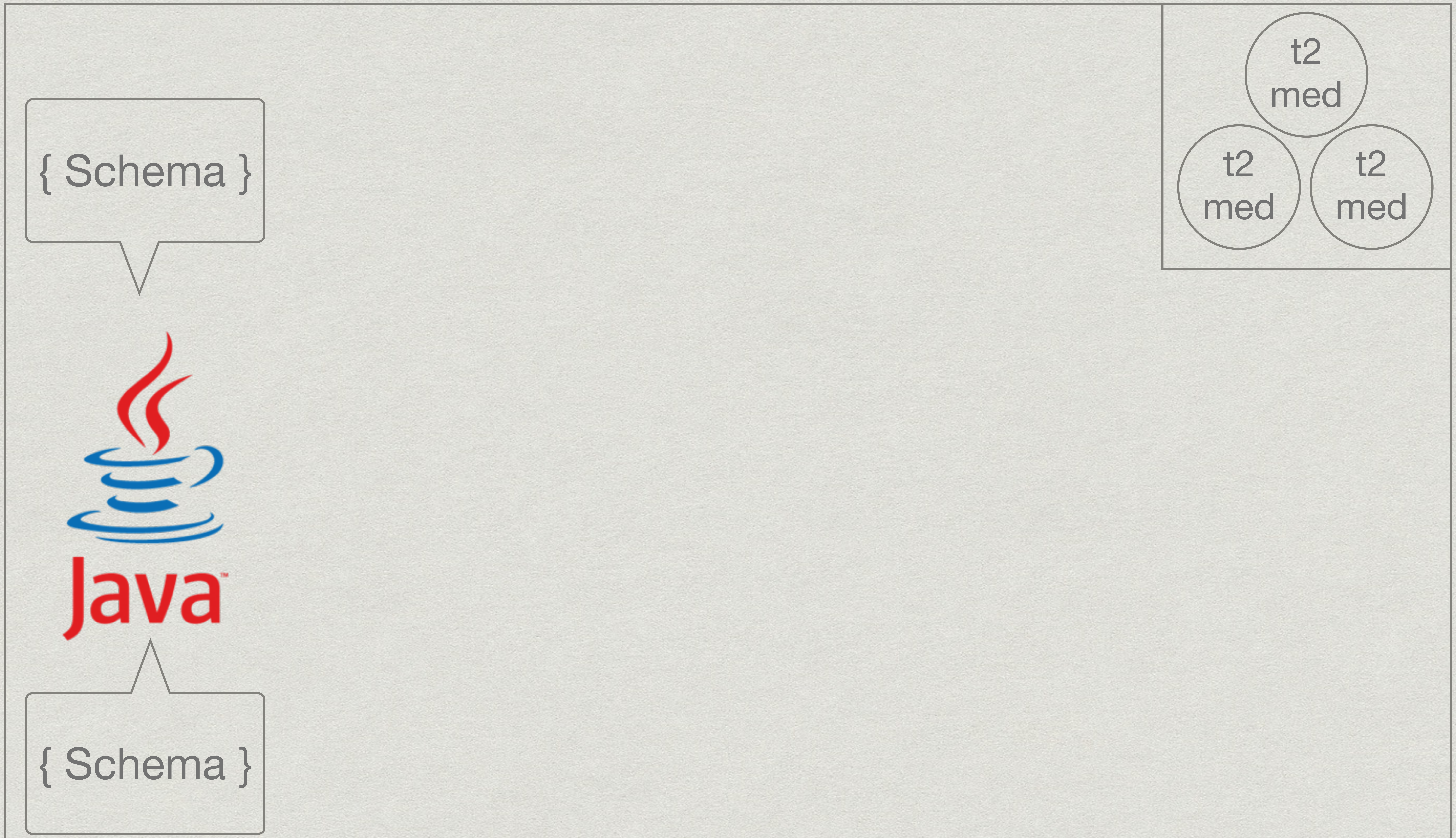
```
{“minType":"BATHROOM",
“majType”:"HUMANITARIAN,
“major_area_num”:19,
“minor_area_num":436,
“quantity”:3.7826,
“unique_id":1000,
“runtime":1443200834318,
“item_sensed”:"toilet",
“subject_measured”:"water",
“sensor_location_name”:"bathroom",
"ticks_since_turn_on":2}
```

JSON message
~ 265 bytes

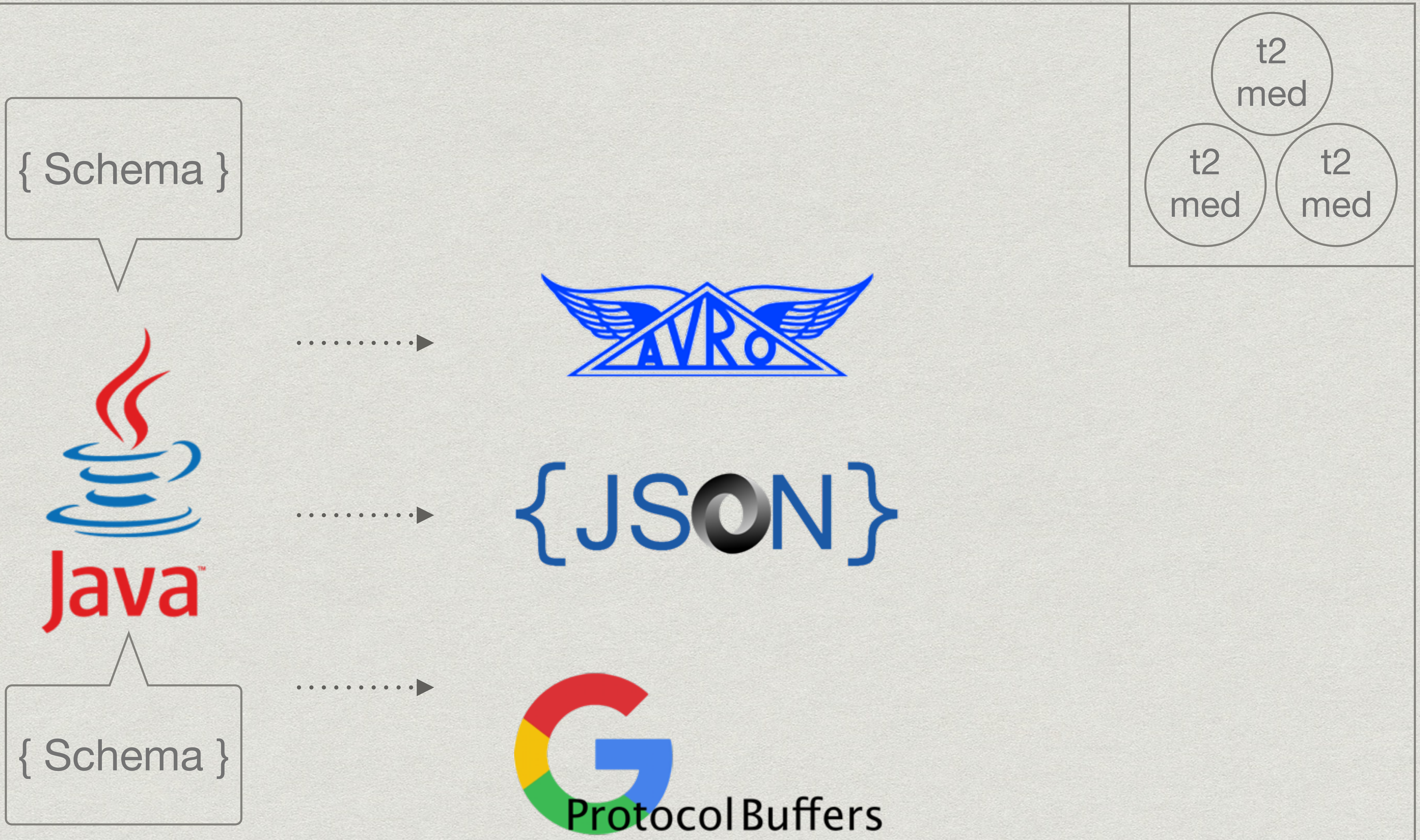
ProtoBuf Byte Array
~ 69 bytes

Avro Byte Array
~ 58 bytes

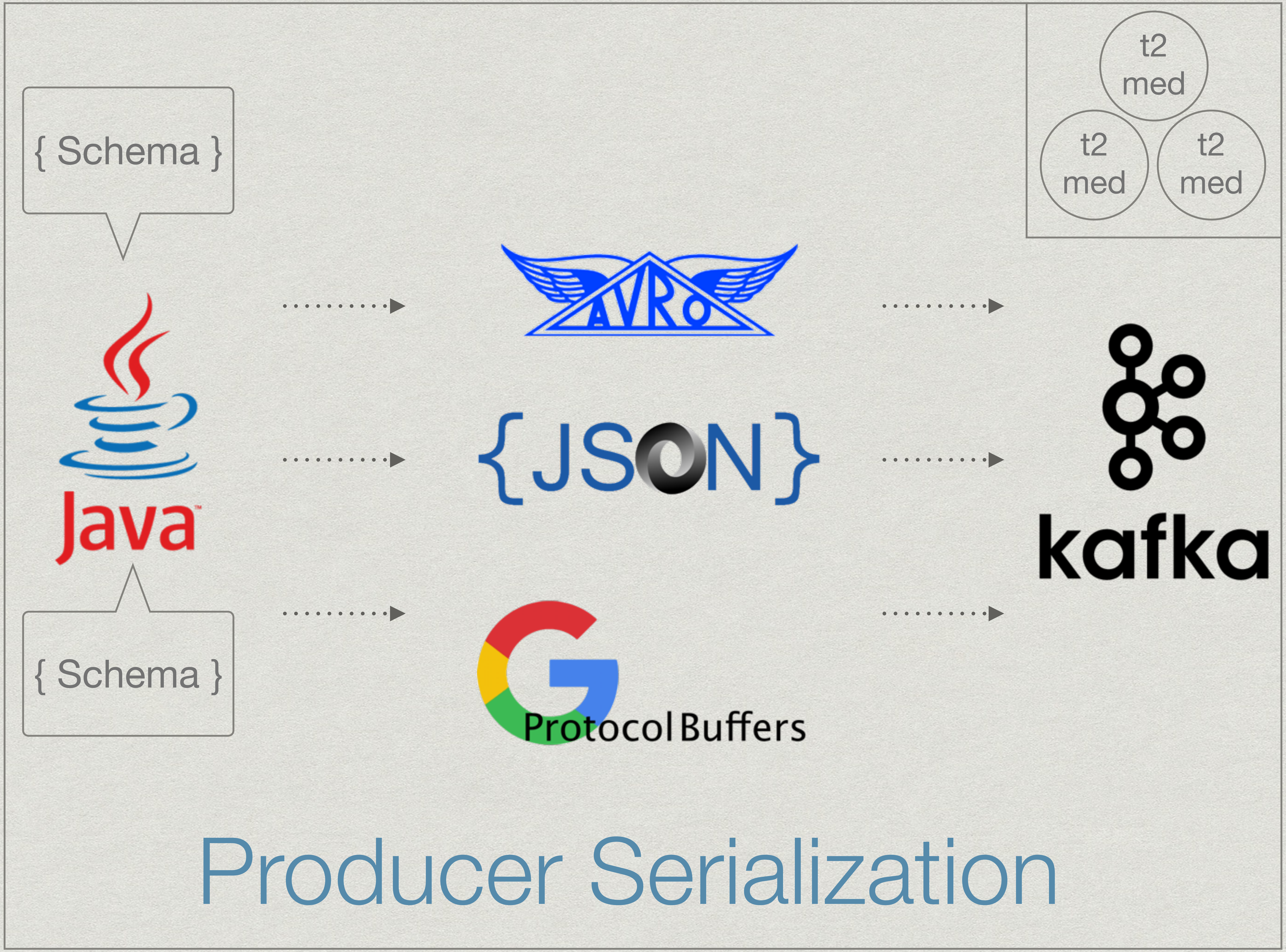
Avro Byte Array with schema header
~ 540 bytes



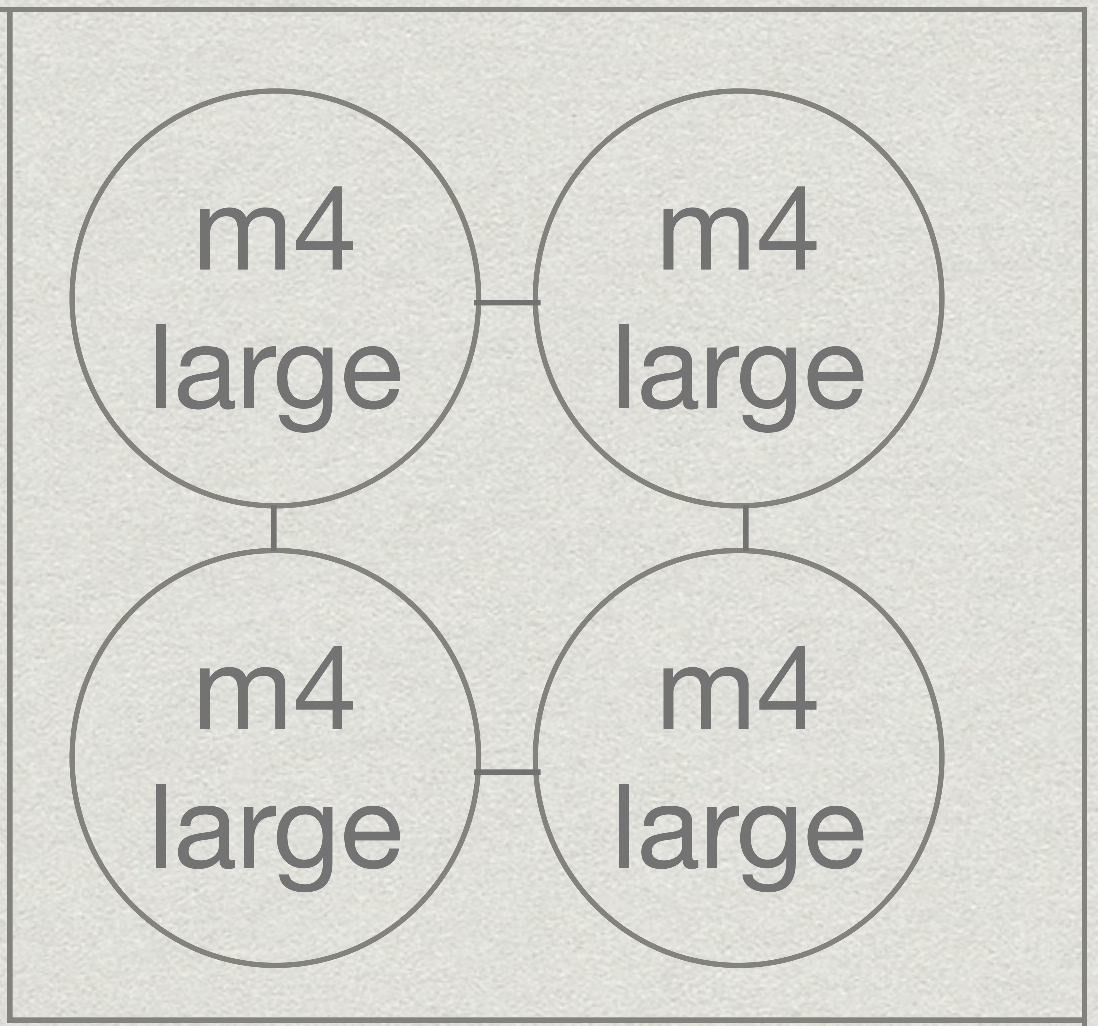
Producer Serialization



Producer Serialization



Producer Serialization

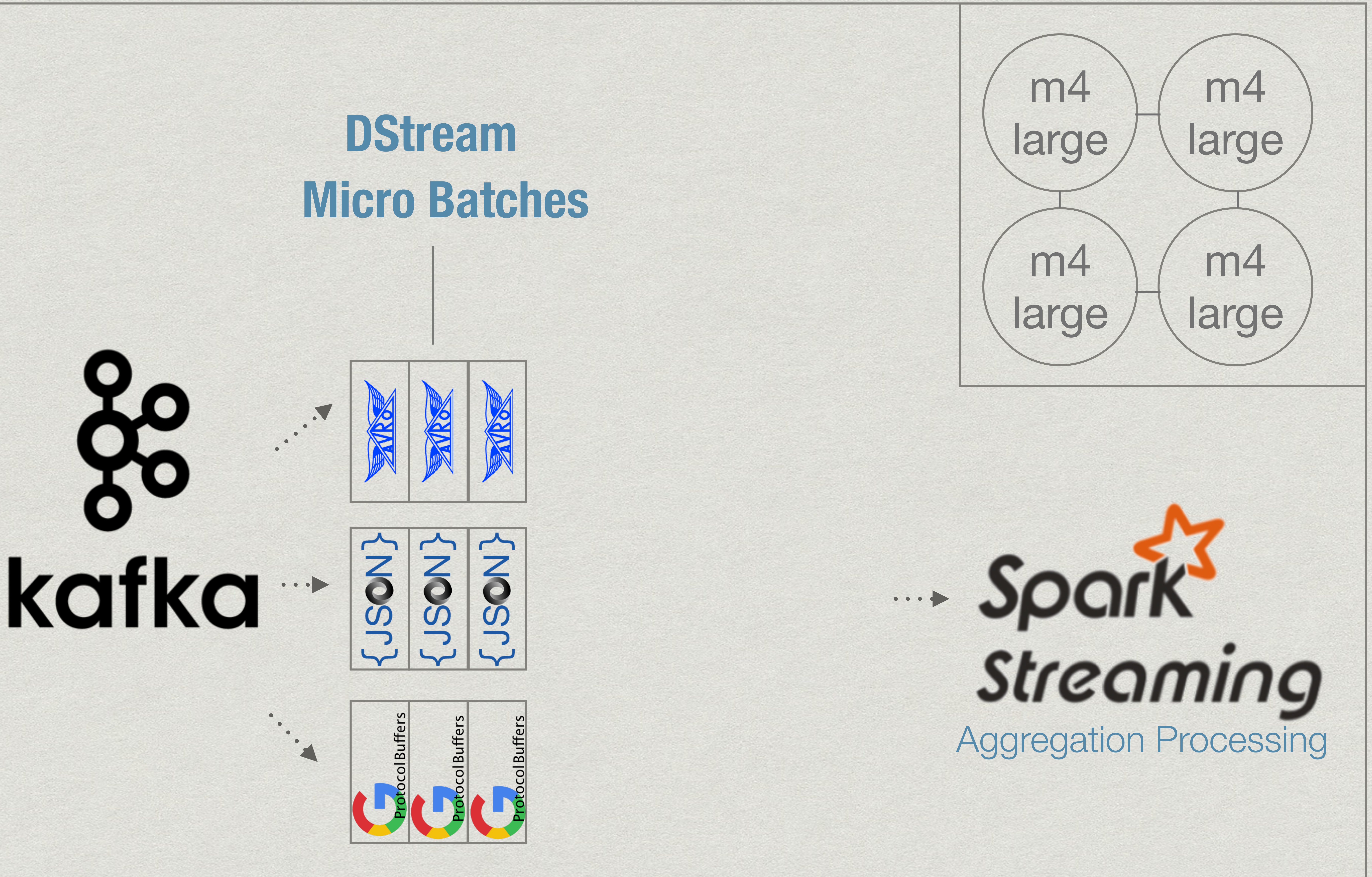


kafka

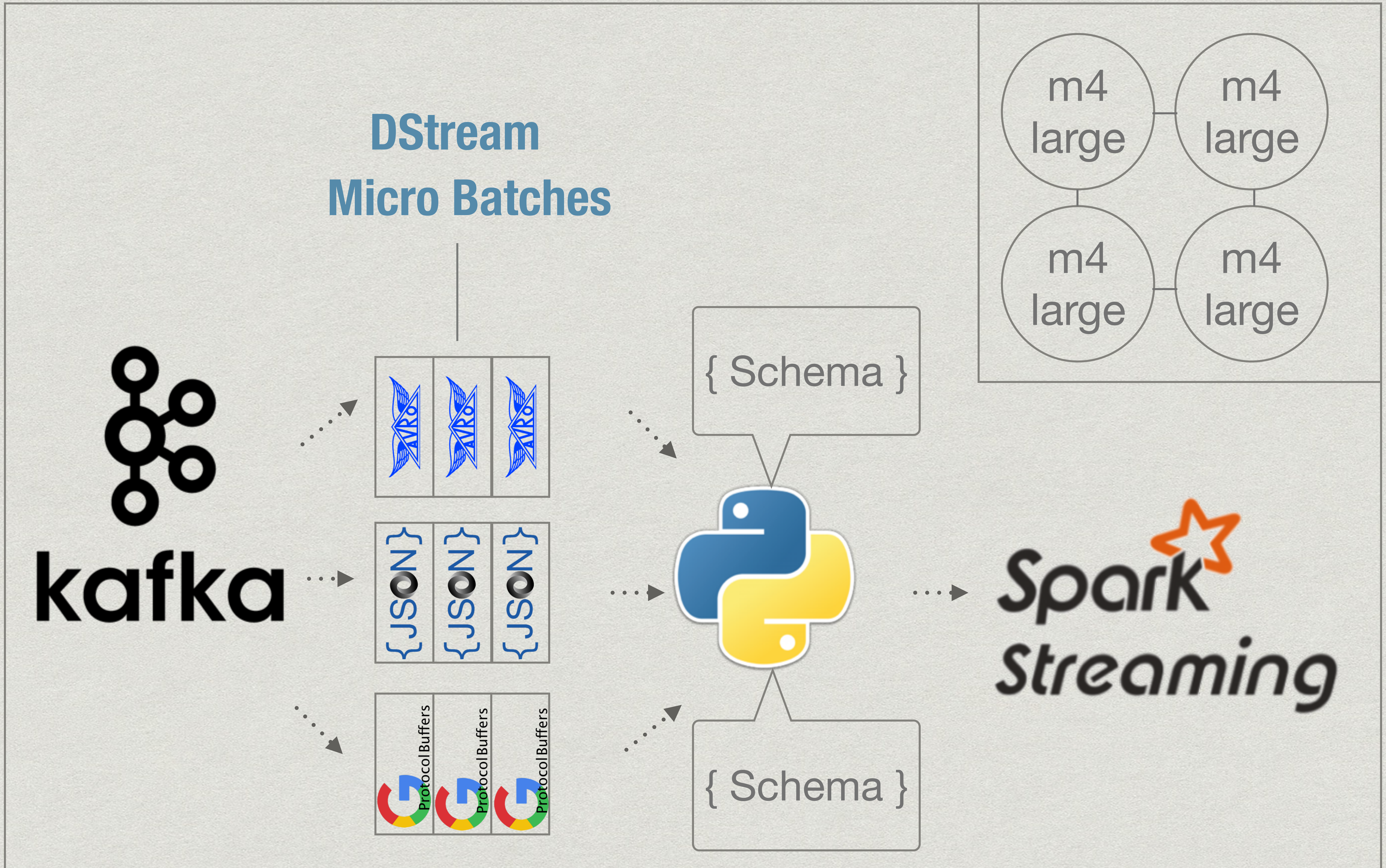
Spark Streaming

Aggregation Processing

Consumer Deserialization

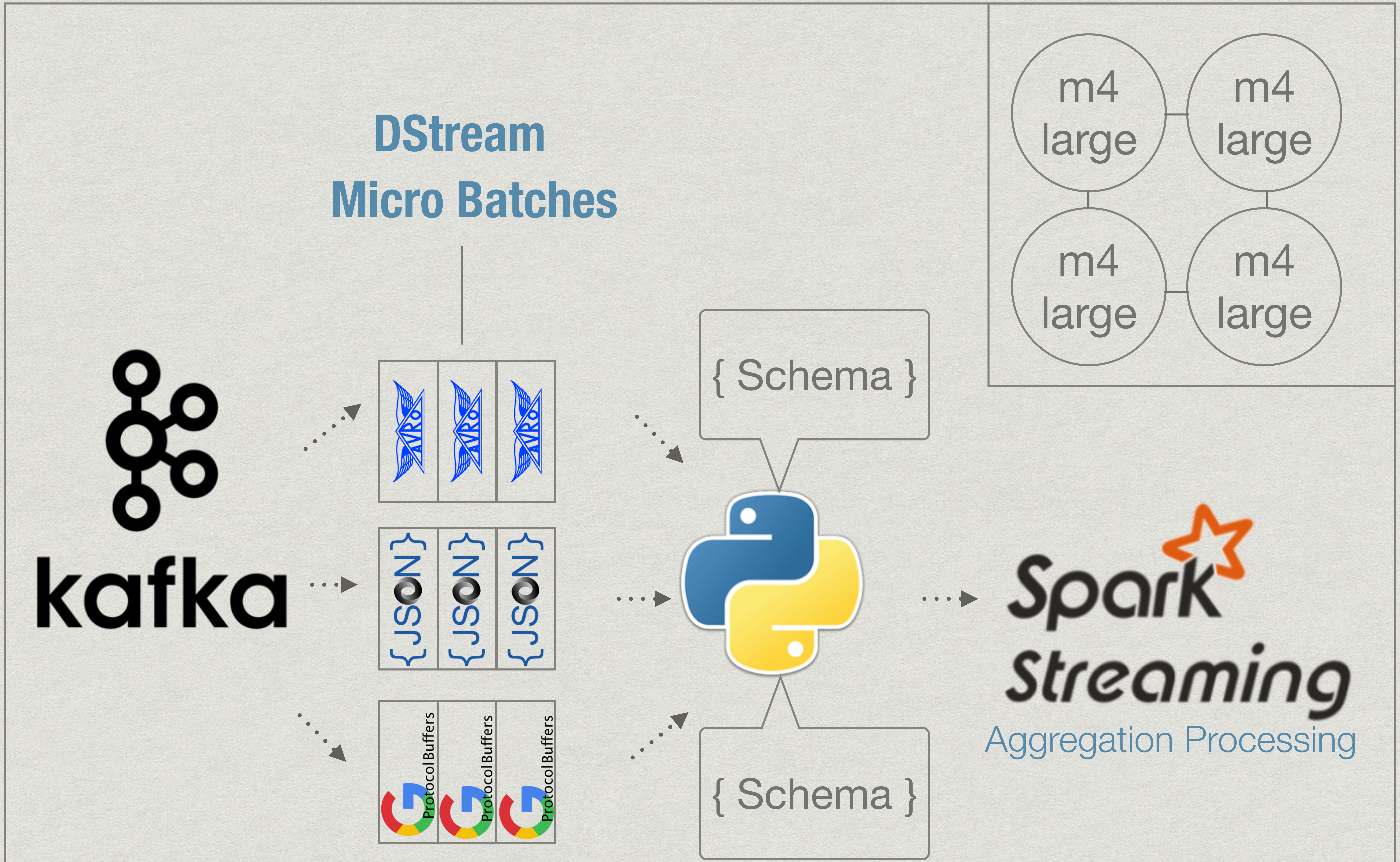


Consumer Deserialization

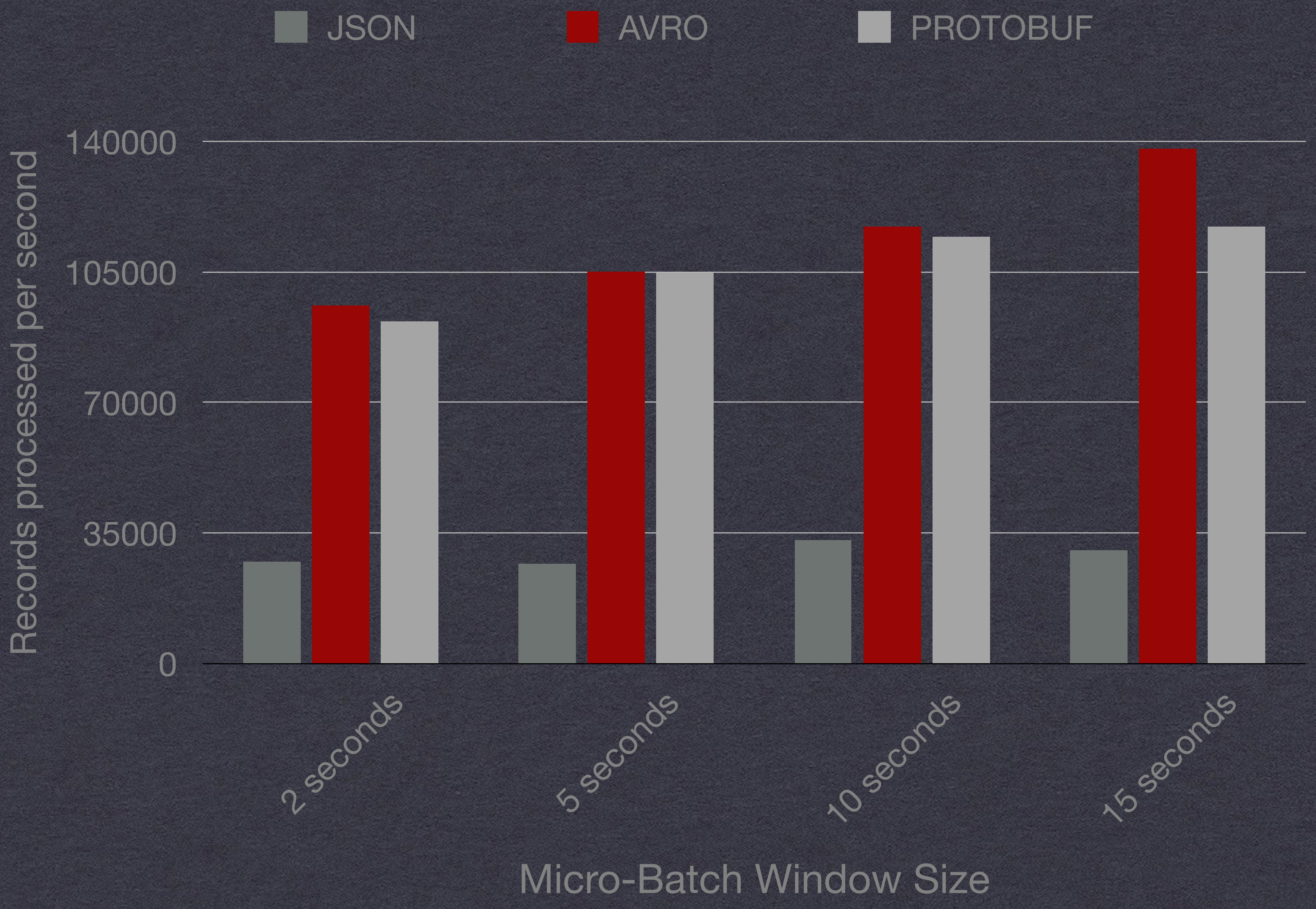


Consumer Deserialization

Consumer Deserialization



COMPARISON OF SERIALIZATION AND BATCH SIZE
3 instances, 3 JVMs per instance



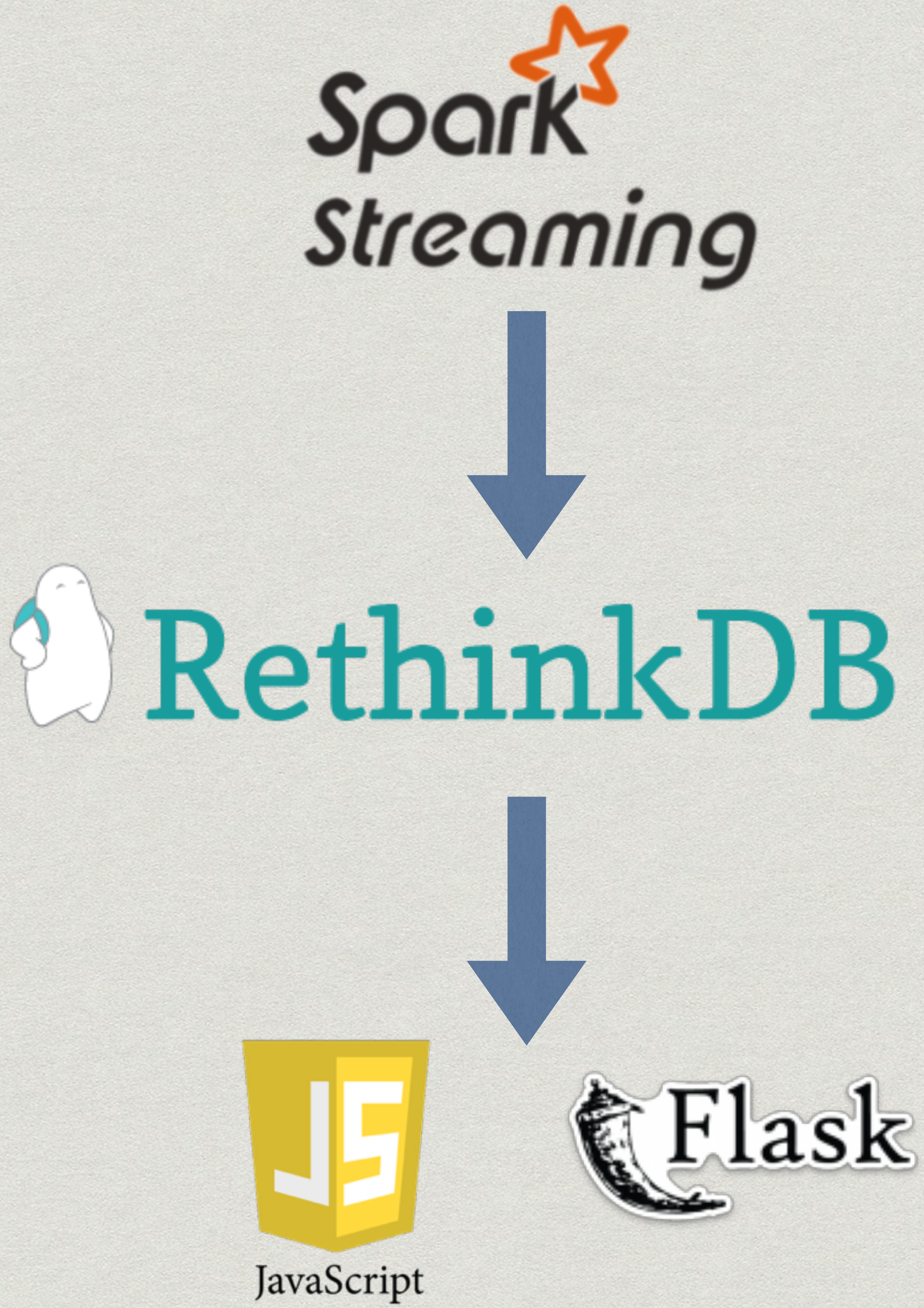
Challenges

- Serializing in Java and de-serializing in Python - no docs
- Java kafka producer API is new - conflicting docs
- Computing throughput
- Debugging in Pyspark
- Parallel reads with receiver-based consumer

SHAUN DONACHY

MS COMPUTER SCIENCE, VIRGINIA COMMONWEALTH UNIVERSITY

VCU
VIRGINIA COMMONWEALTH UNIVERSITY

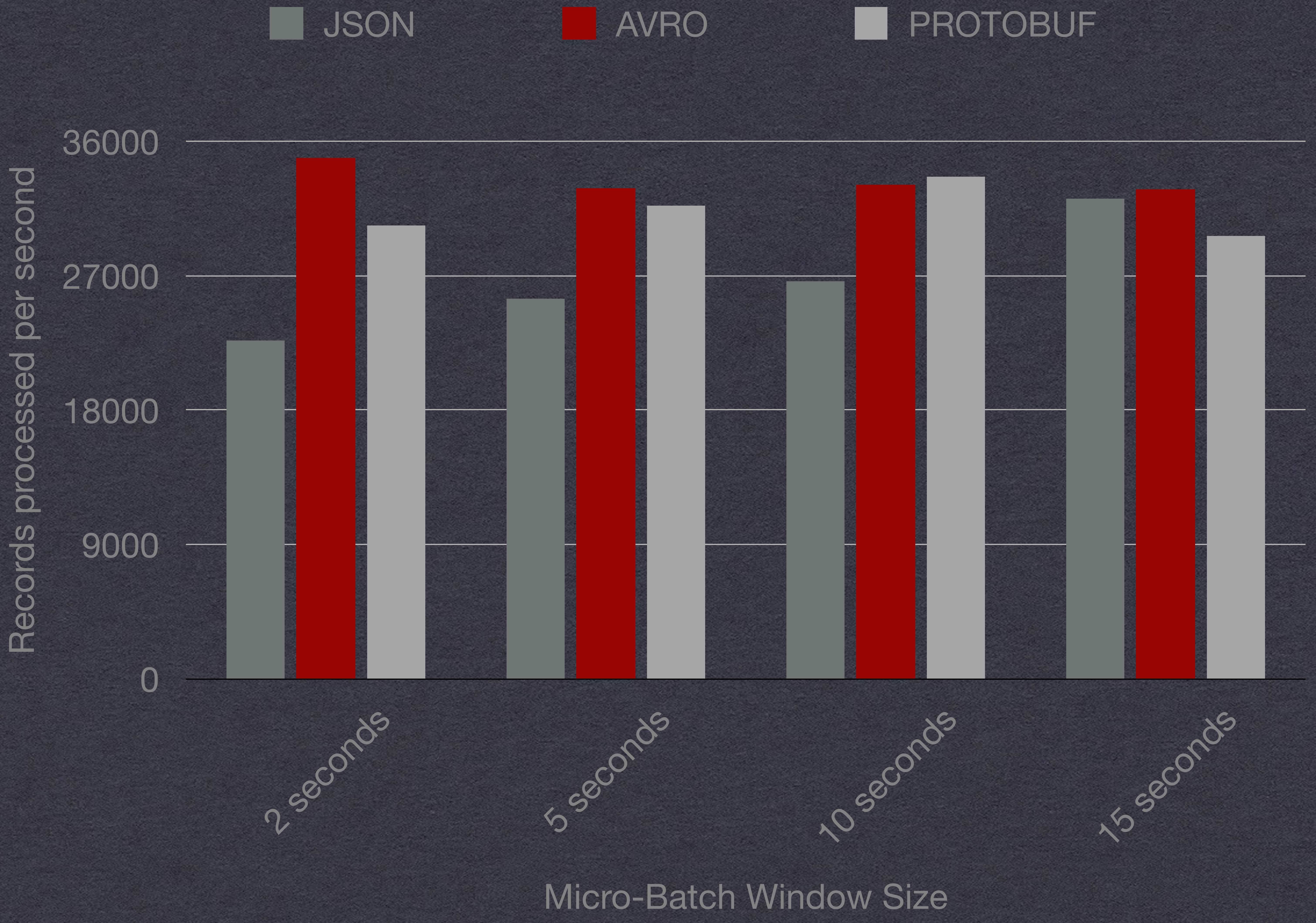

Spark Streaming
RethinkDB
JavaScript
Flask

Experimental Parameters

Producer Machines	Producer jvm per machine	Thread spawn rate	Total Sensors per jvm
2/3	1/3	5 sec	125,000

Kafka Partitions/ Replication	Spark Read Parallelism	Spark Compute Parallelism	Spark micro-batch window
6/2	6	6	2/5/10/15 sec

COMPARISON OF SERIALIZATION AND BATCH SIZE
2 instances, 1 JVM per instance



Findings

- Avro and Protobuf serialization greatly improves throughput over JSON
- Overall Avro and Protobuf show similar performance, the differences are in their characteristics

Image credits

- Disney, Star Wars
- BrainlessTales.com
- Settlers of Catan
- Apache: Spark, Avro, Kafka
- Python, rethinkDB, flask