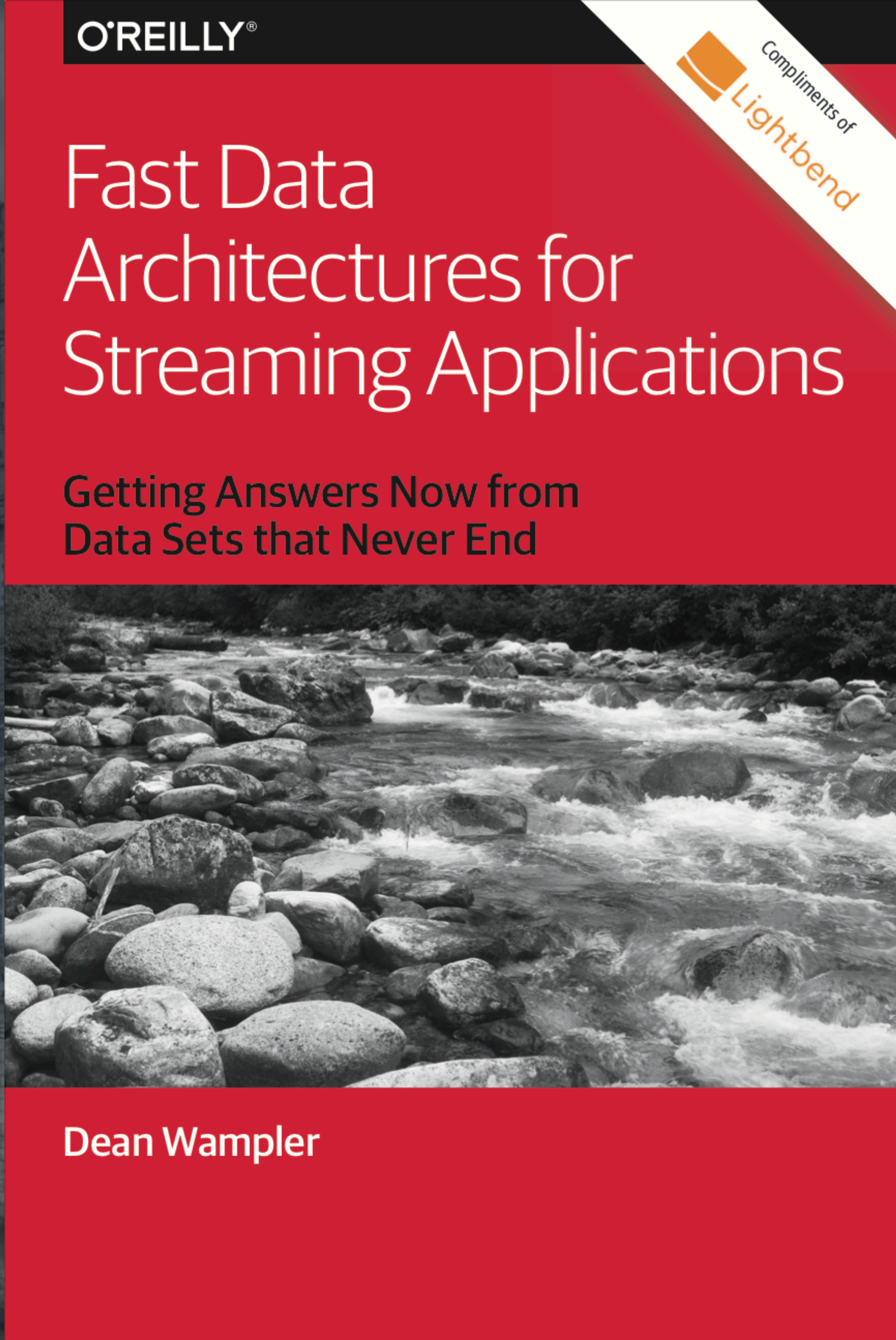
# Executive Briefing: What You Need to Know about Fast Data

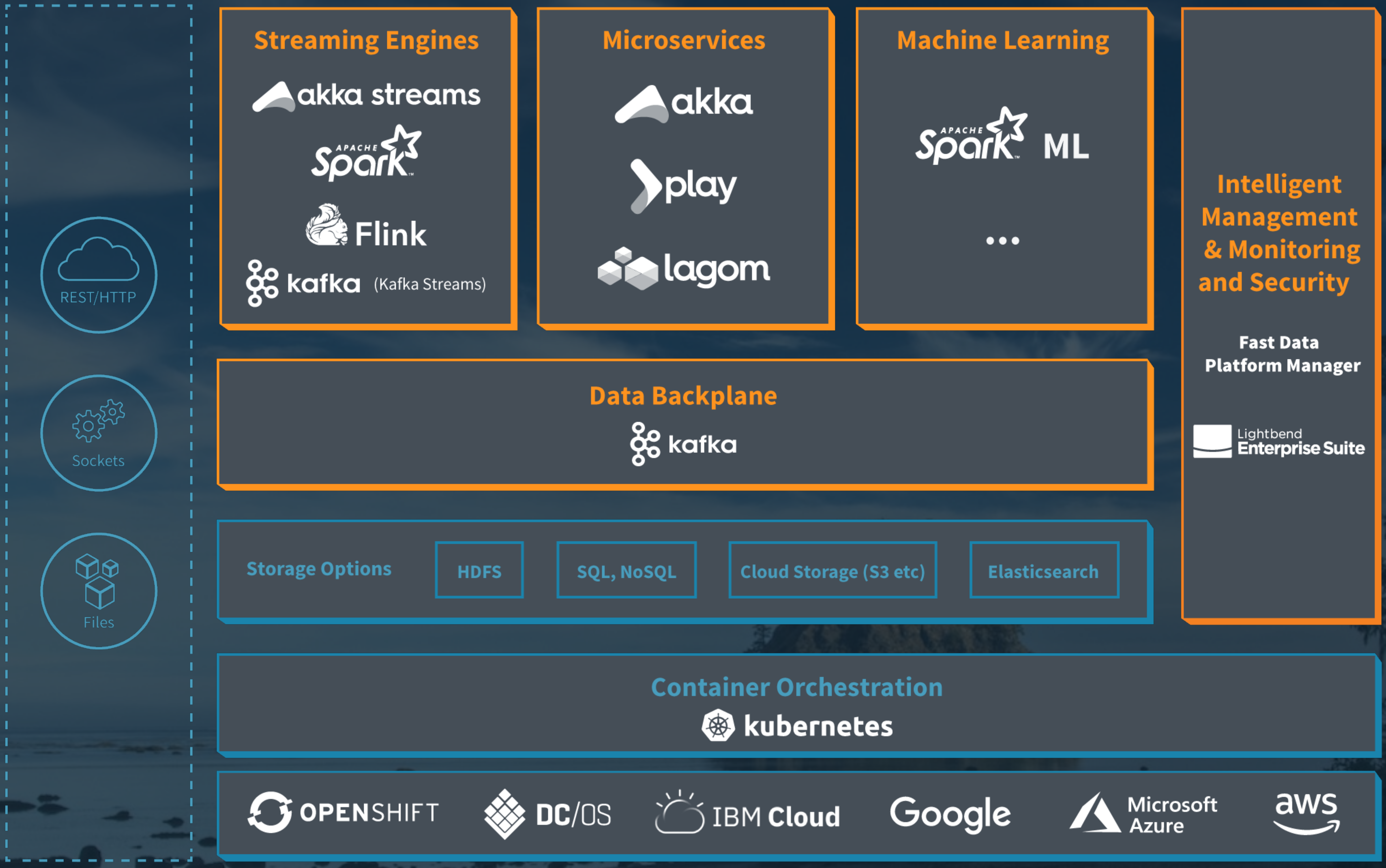
Dean Wampler, Ph.D.
dean@lightbend.com
@deanwampler
polyglotprogramming.com/talks

Lightbend

# Based on this report

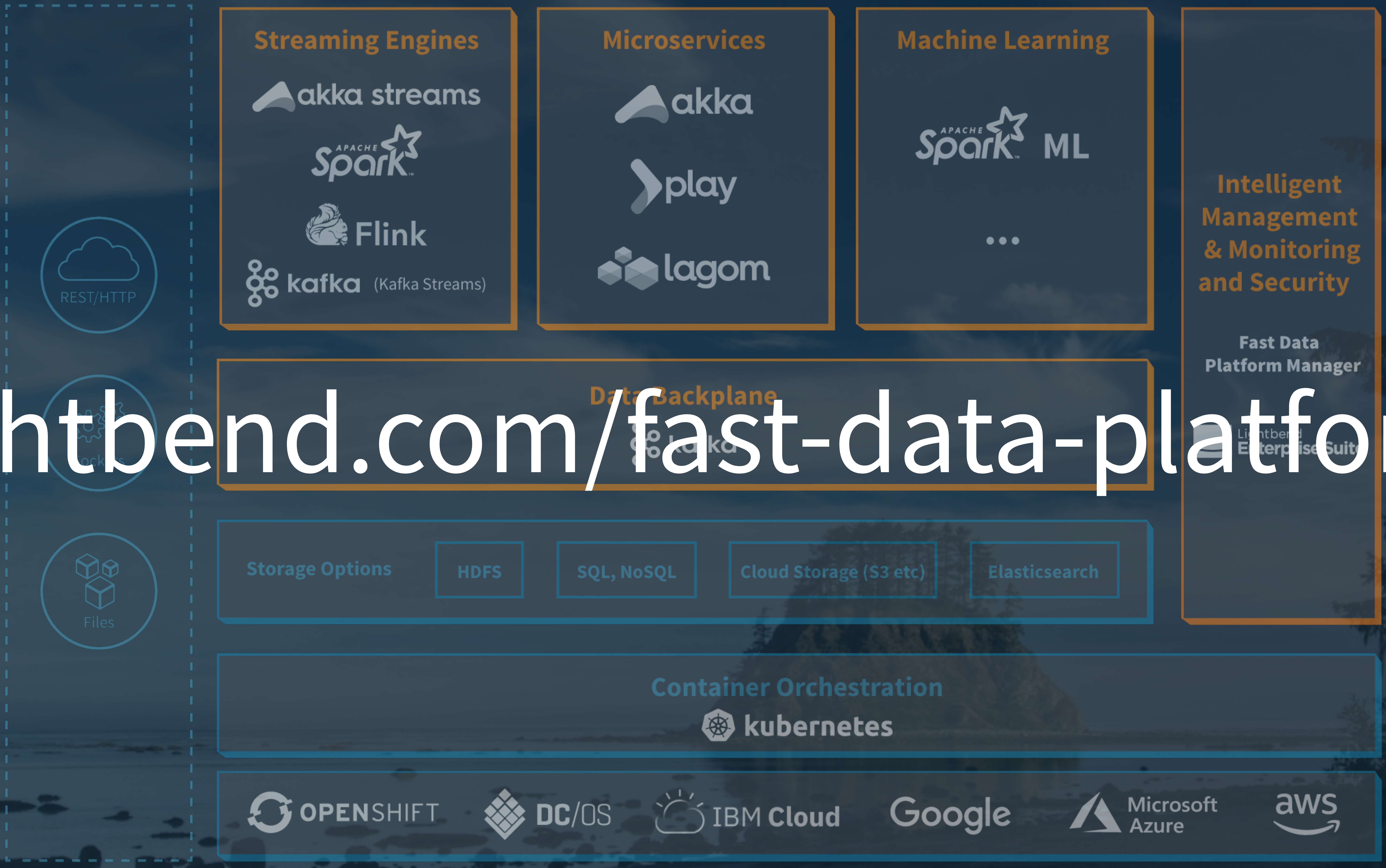
go.lightbend.com/fast-data-architectures-for-streaming-applications-oreilly-2nd-edition






I lead the Lightbend Fast Data Platform project; streaming data and microservices

# lightbend.com/fast-data-platform



# What We’ll Discuss

- Why streaming? Why now?
- How to choose technologies
- The impact streaming will have on your organization

# What We’ll Discuss


# Why Streaming?

- New opportunities that require streaming
  - Media content is obviously one ;)
- Upgrading batch applications for competitive advantage

# Why Streaming?

# Fast Data Use Cases

Similar IoT Architectures

| Predictive Analytics | IoT | Real-time Personalization | Real-time Financial Processes |
|---|---|---|---|
| Apply ML models to large volumes of device data to pre-empt failures / outages | Real-time consumer and industrial Device and Supply Chain management at scale | Real-time marketing based on behavior, location, inventory levels, product promotions, etc. | Drive better business outcomes through real-time risk, fraud detection, compliance, audit, governance, etc. |
| Hewlett Packard Enterprise | Starbucks Coffee | Royal Caribbean International | Capital One |

https://www.lightbend.com/customers

# Predictive Analytics

Hewlett Packard Enterprise

- ML models applied to device telemetry to detect anomalies
- Preemptive maintenance prevents potential failures that would impact users

# Predictive Analytics - Core Idea

Handle anomaly: move activity off component, schedule maintenance window to replace it.

Train models to look for anomalies... and score incoming telemetry.

**Anomaly Handler**

Probable Anomalies

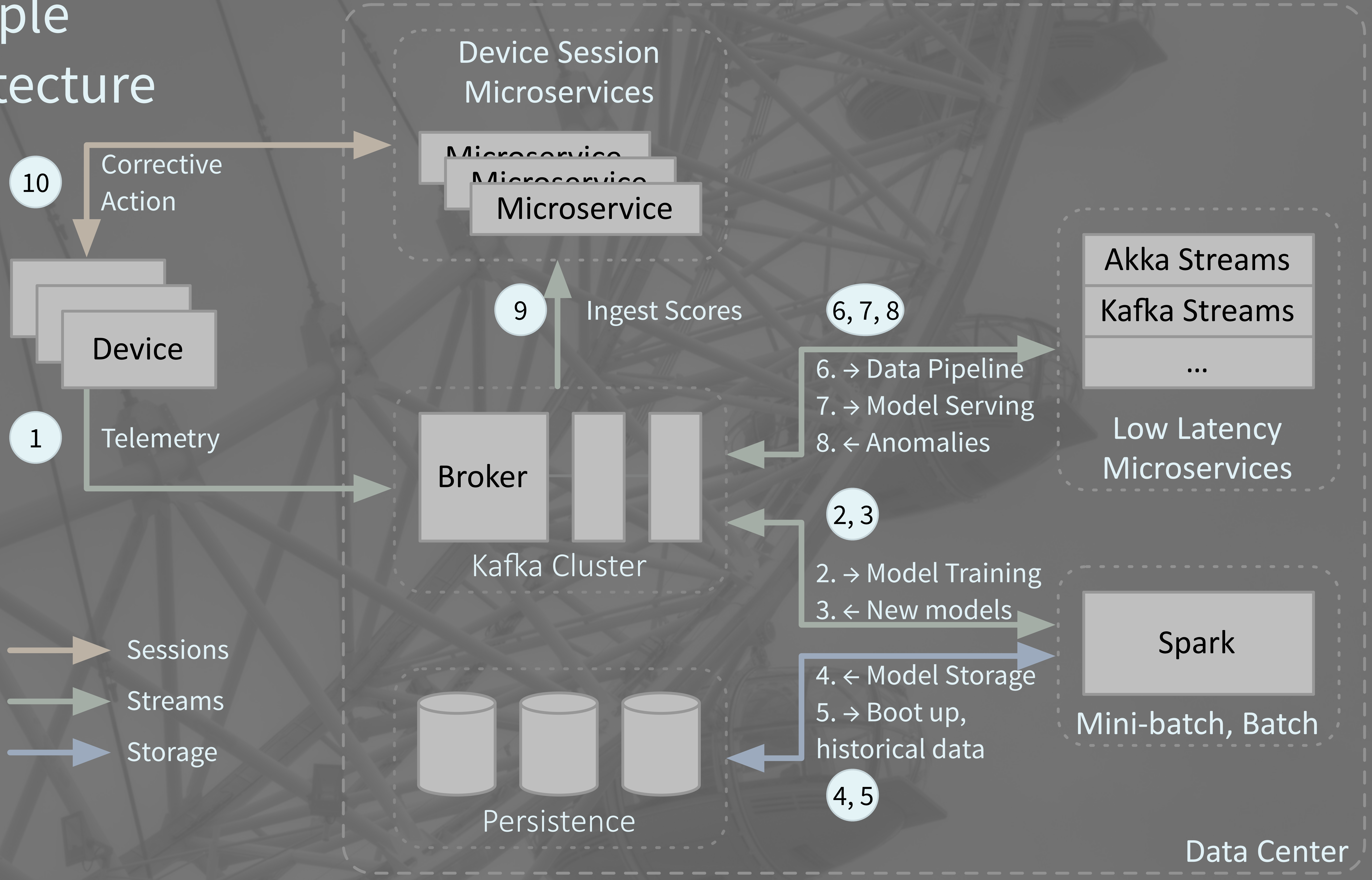
**Anomaly Detection: Model**

Corrective Actions

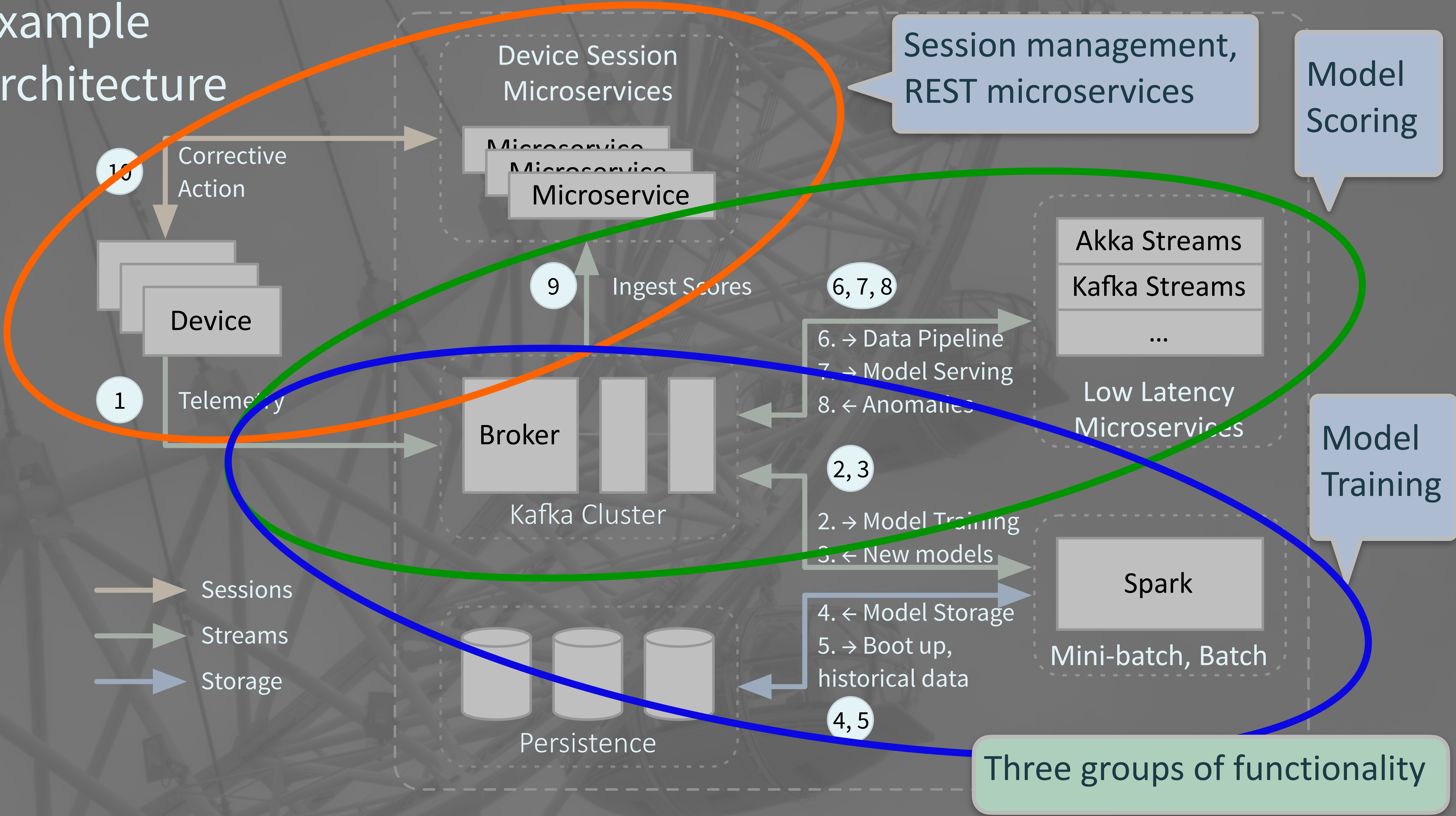

Telemetry Records

Ingest telemetry from edge devices.

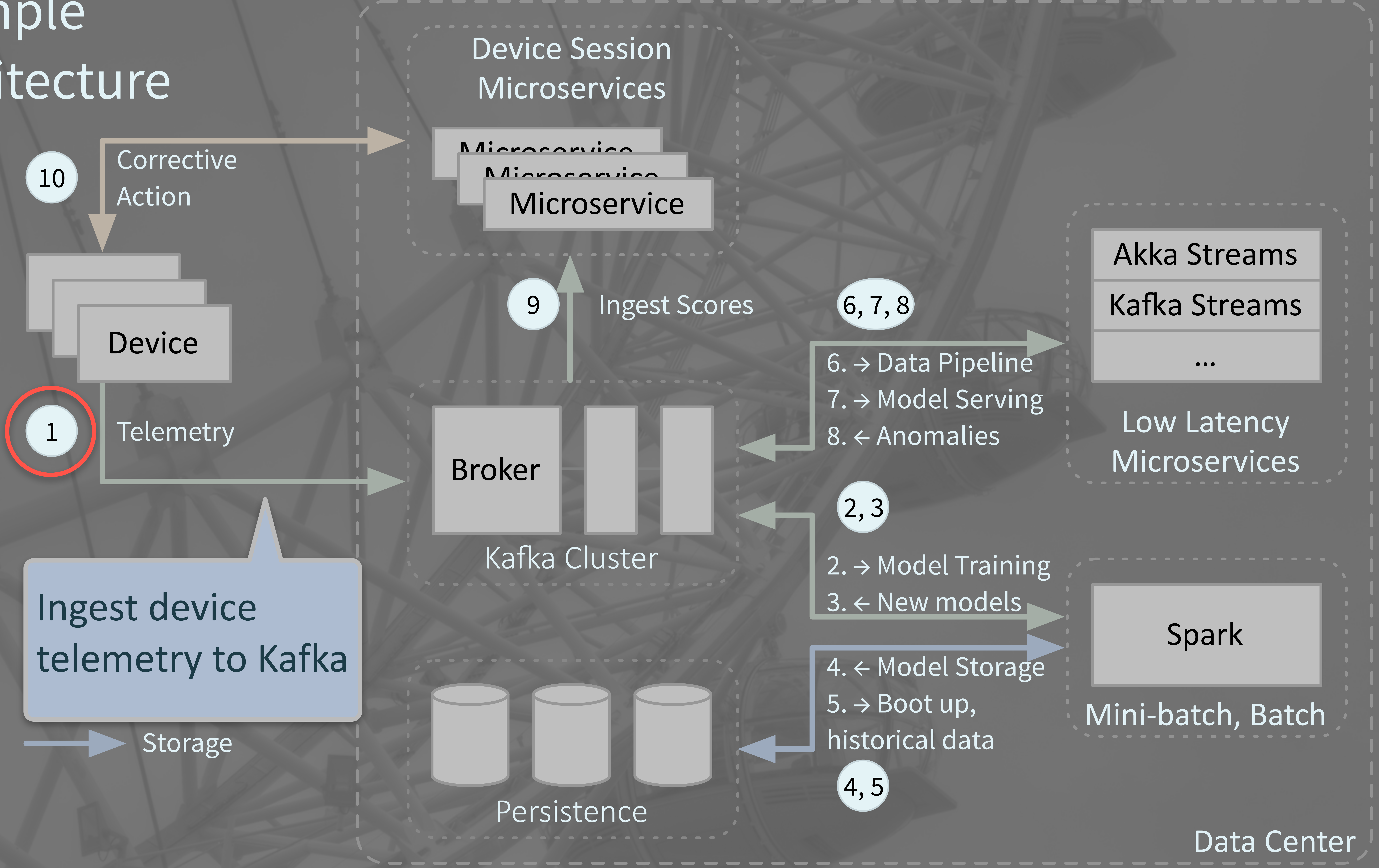
# Example Architecture



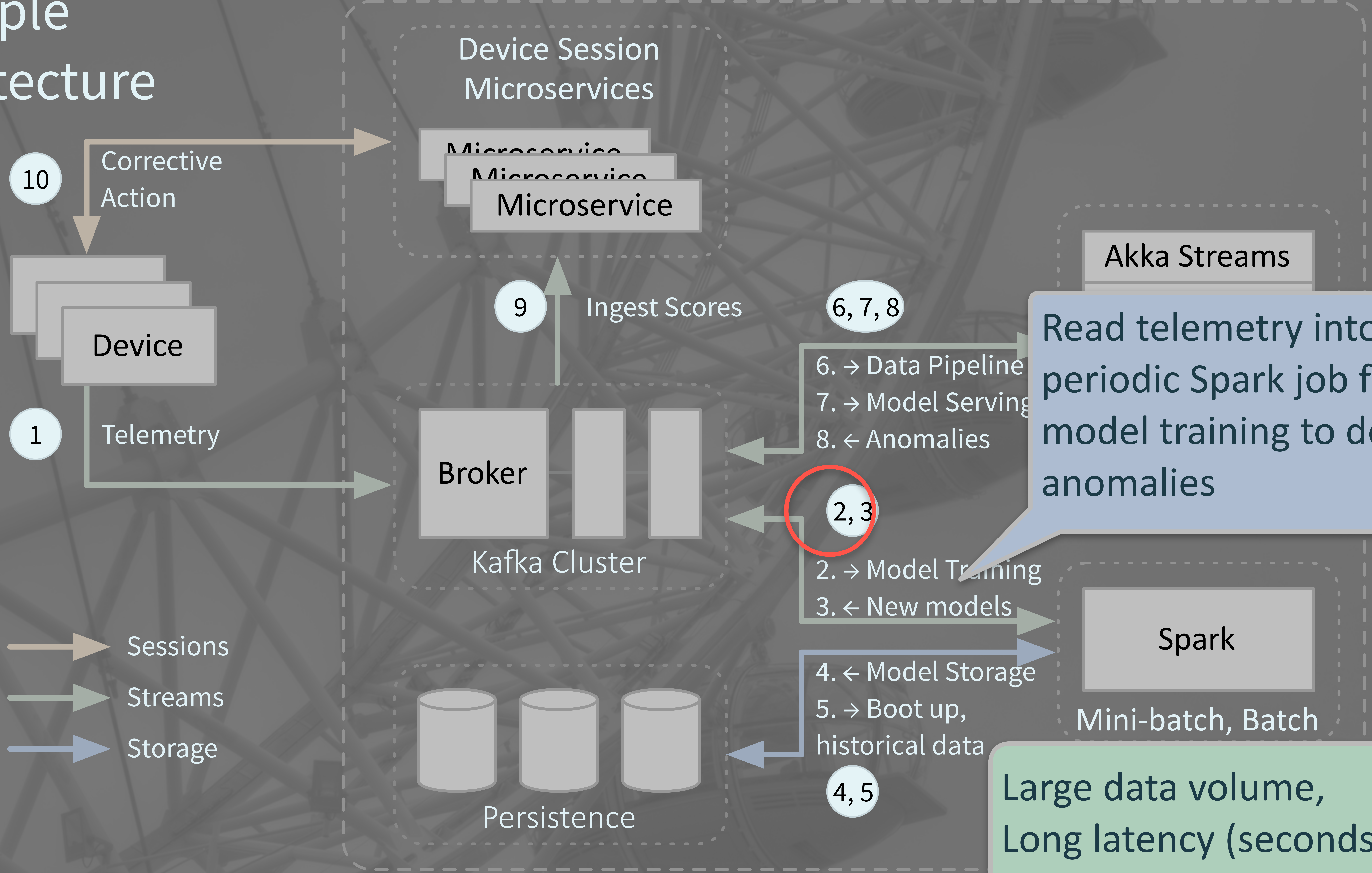
# Example Architecture



# Example Architecture



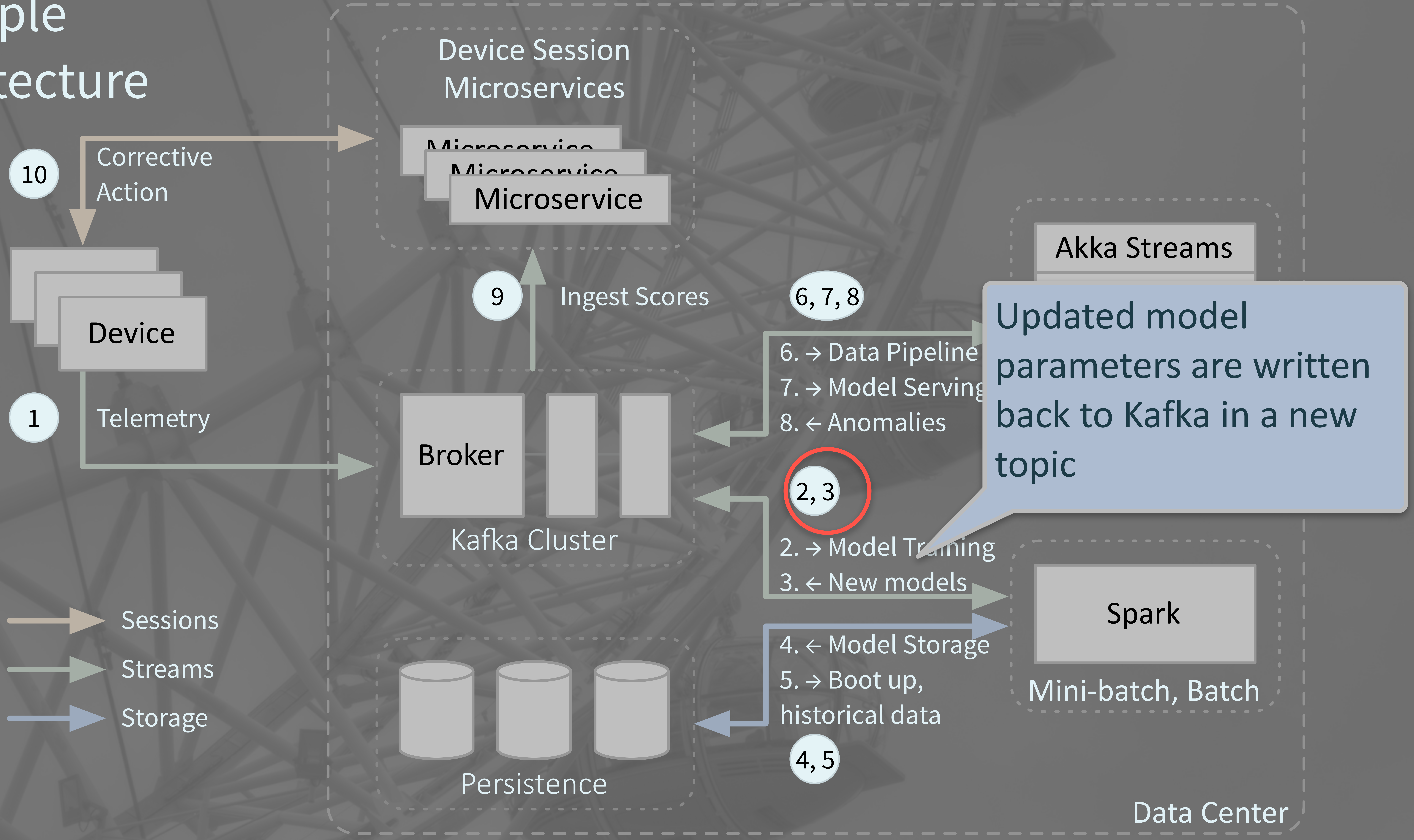
# Example Architecture



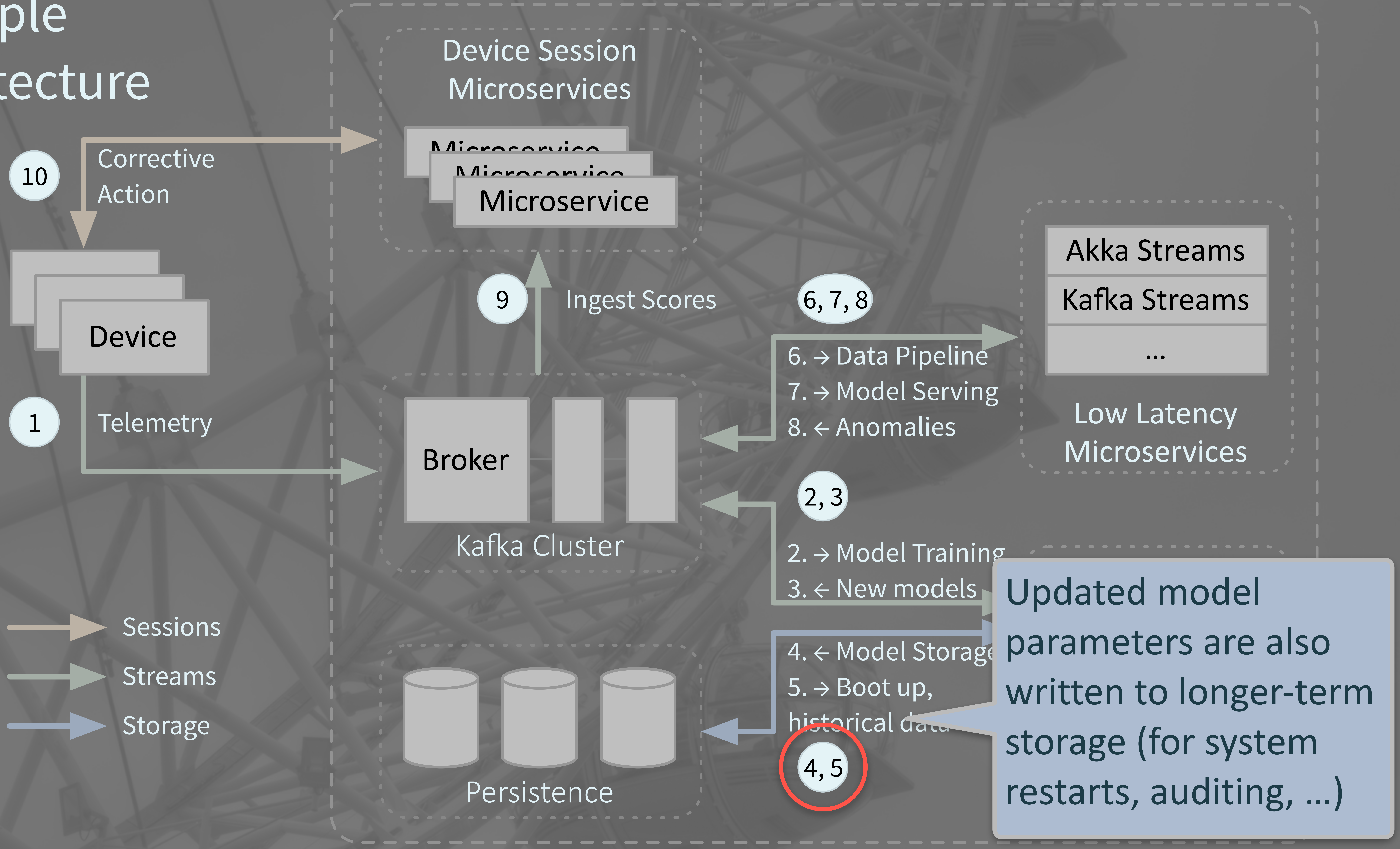

Read telemetry into a periodic Spark job for model training to detect anomalies

Large data volume,
Long latency (seconds-days)

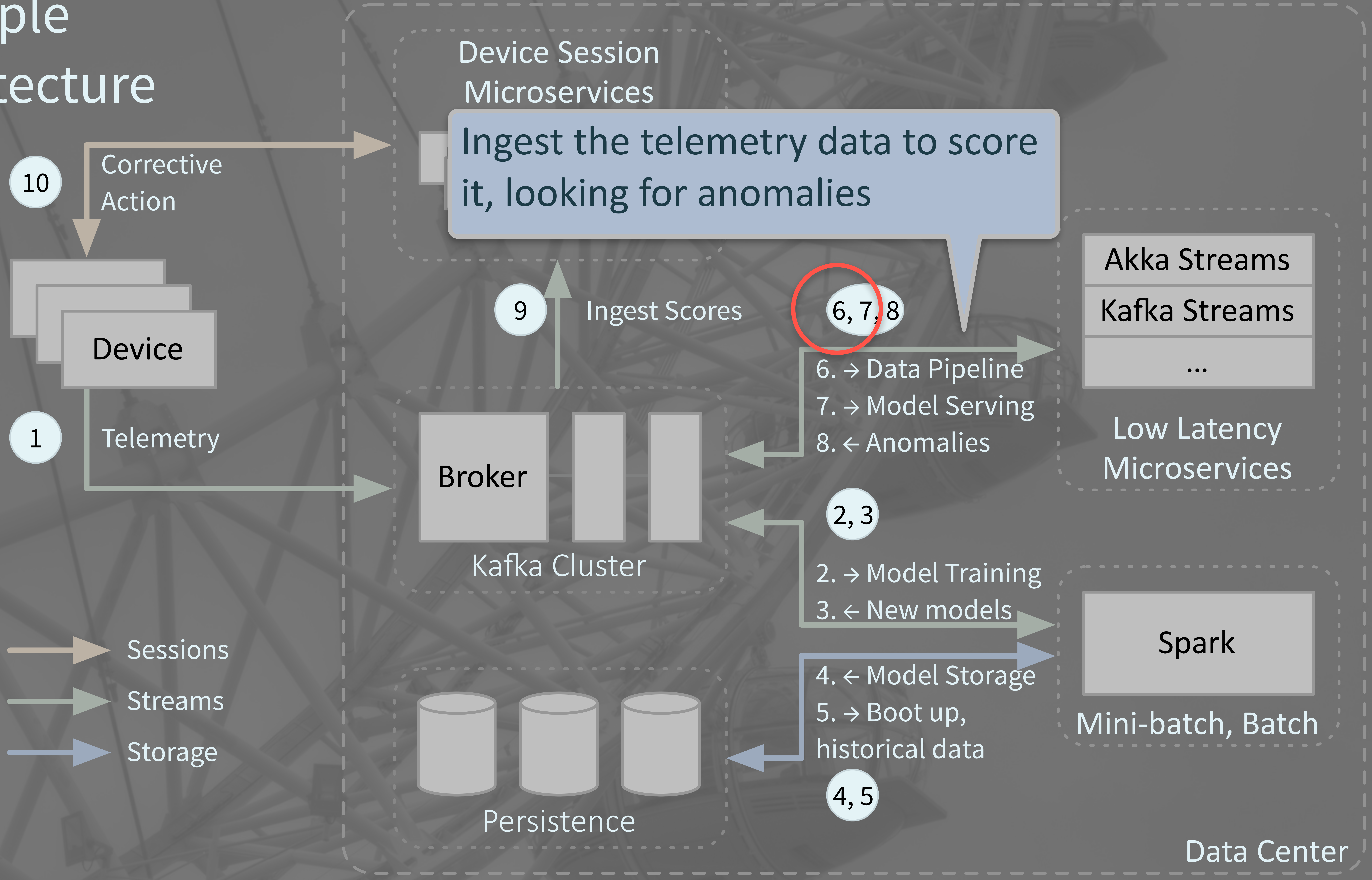
# Example Architecture



# Example Architecture



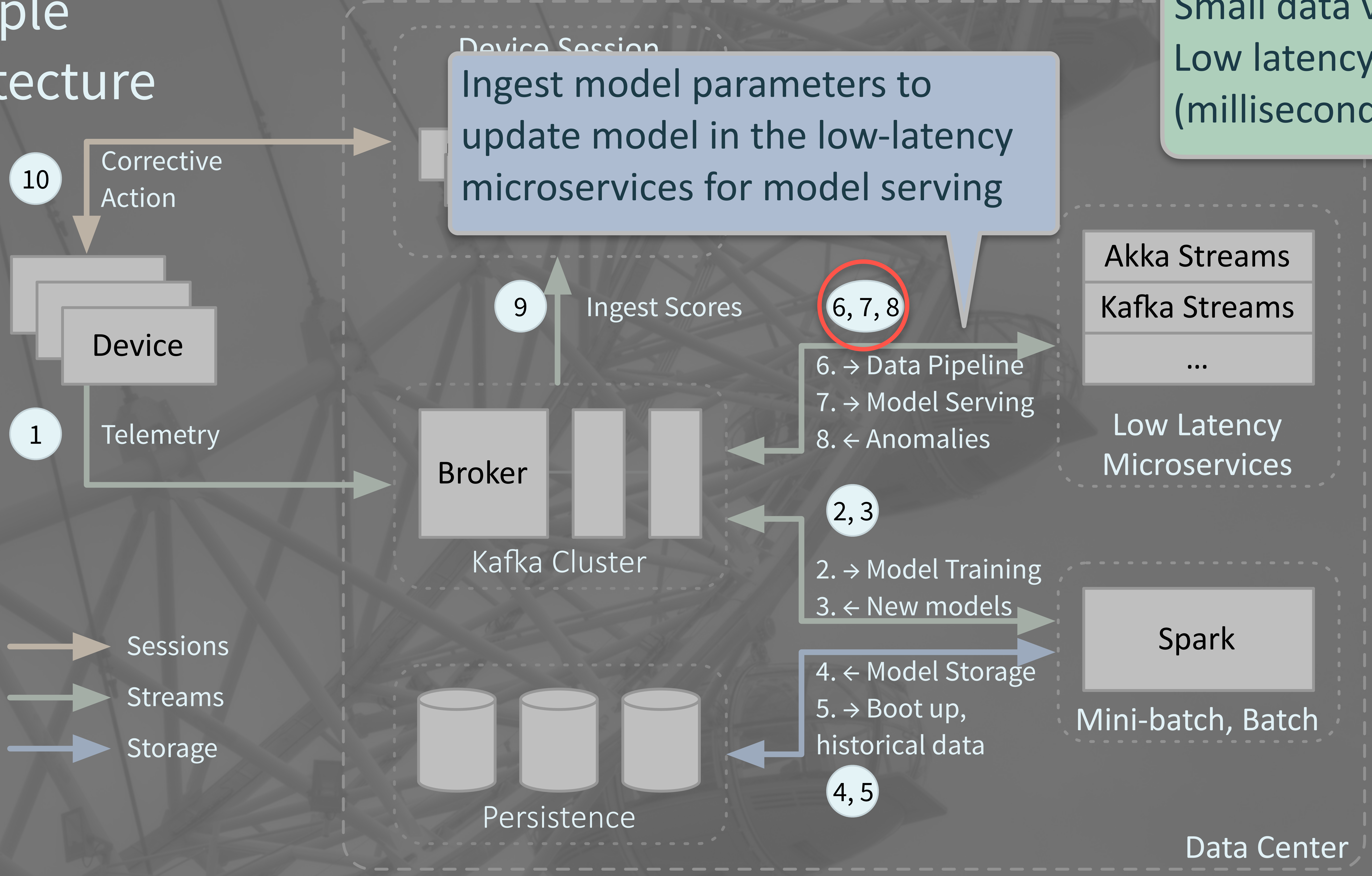
# Example Architecture



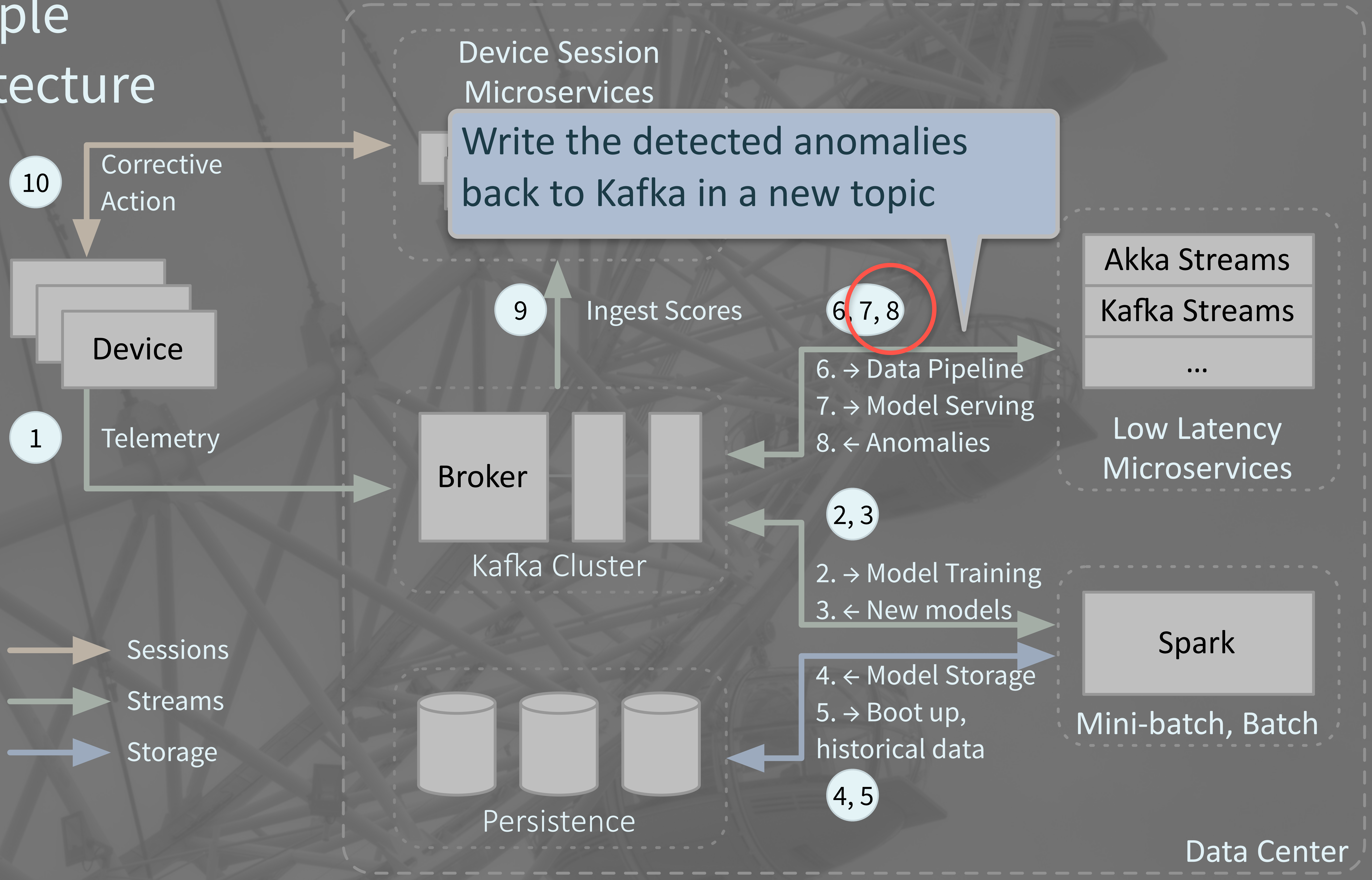
# Example Architecture

Small data volume, Low latency (milliseconds-...)

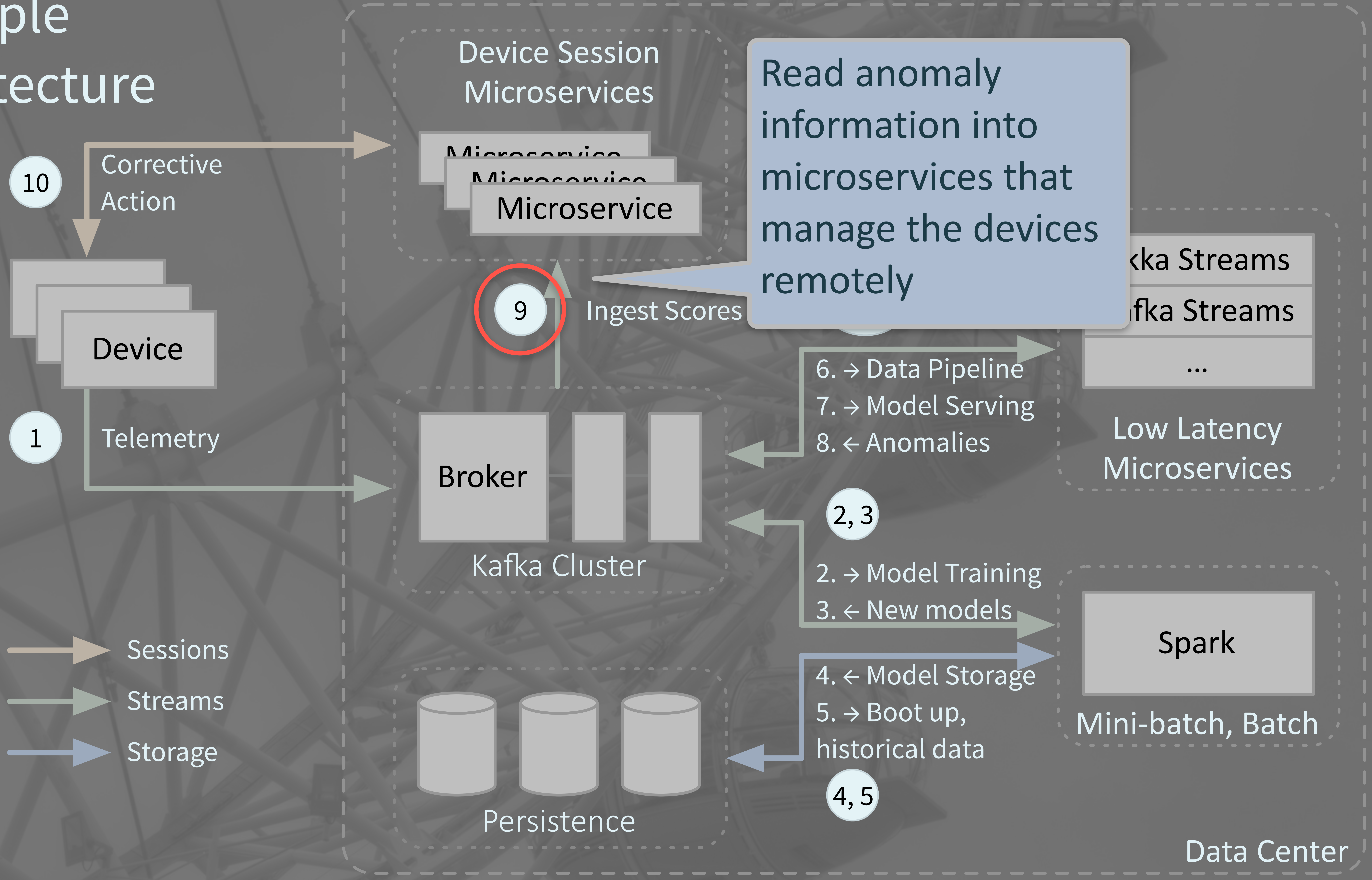
Ingest model parameters to update model in the low-latency microservices for model serving



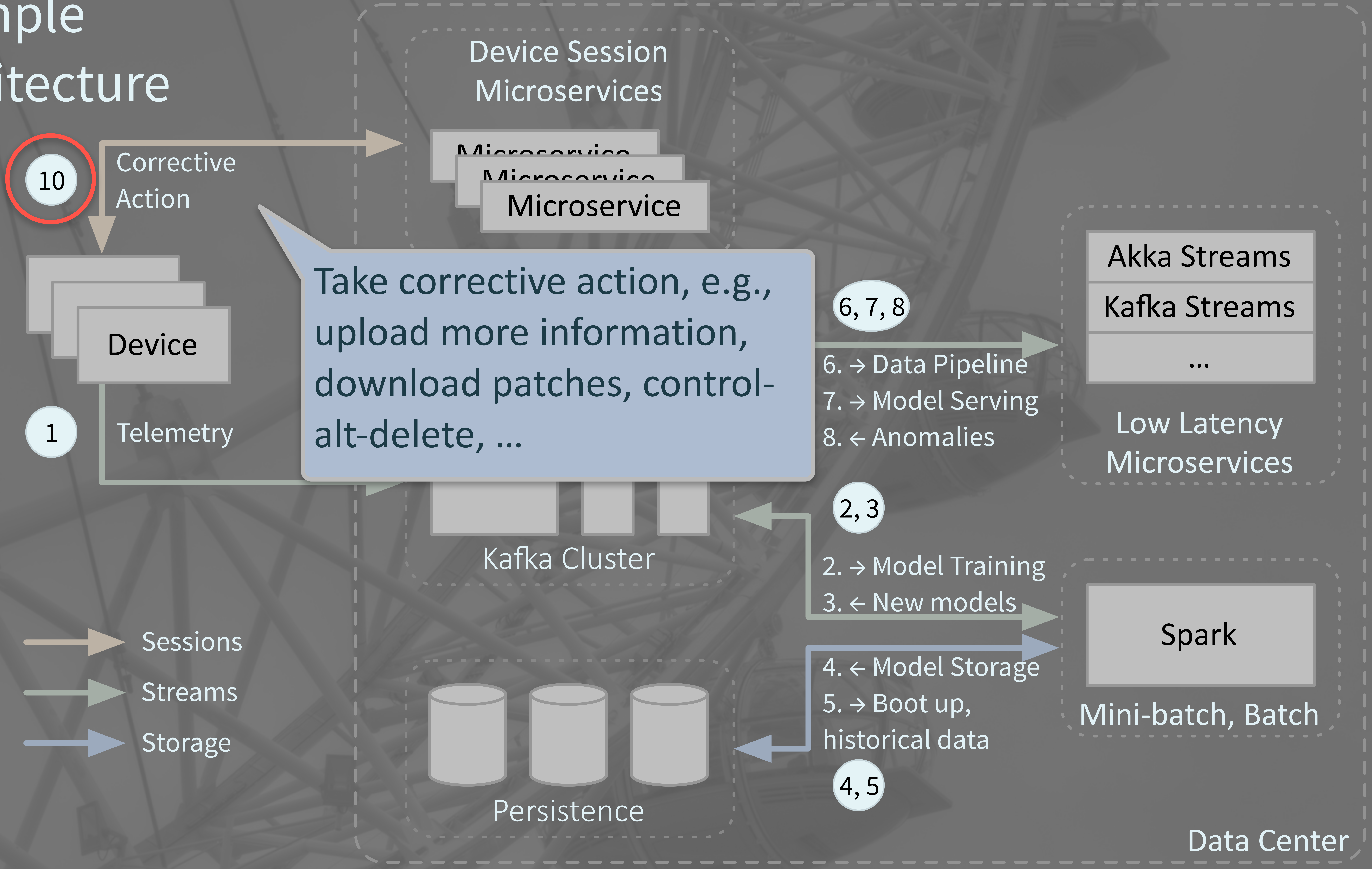
# Example Architecture



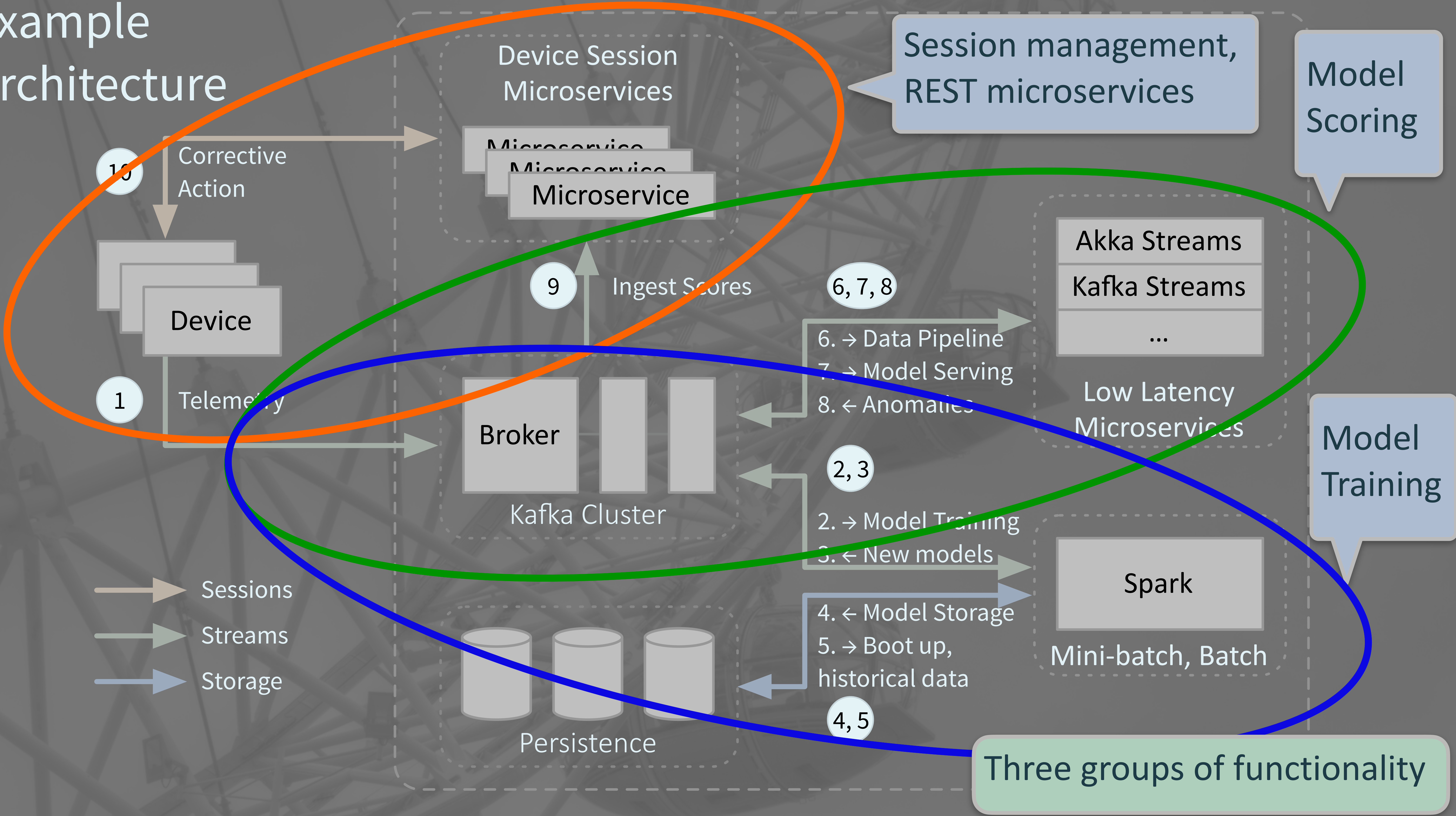
# Example Architecture

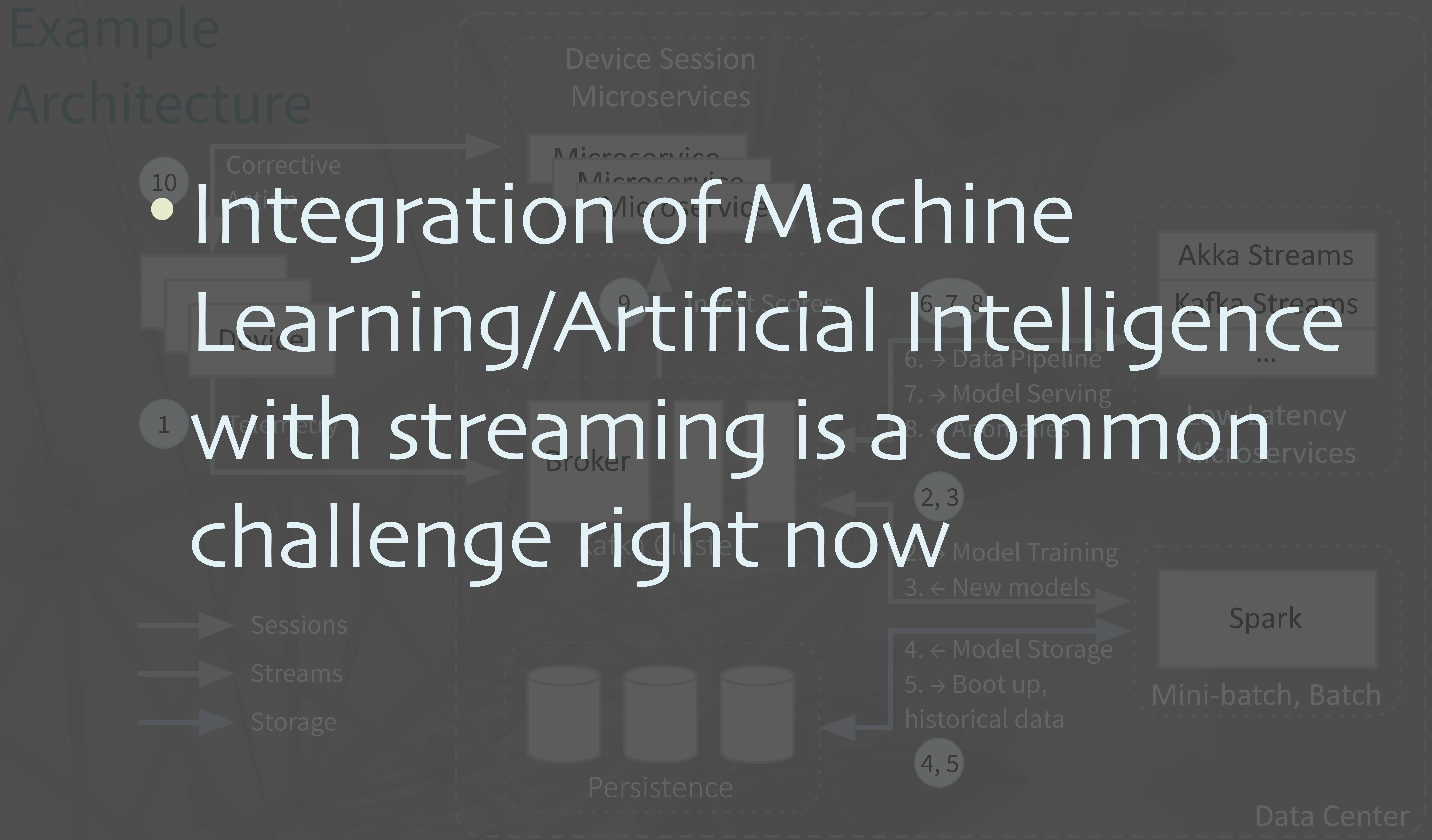


# Example Architecture



# Example Architecture

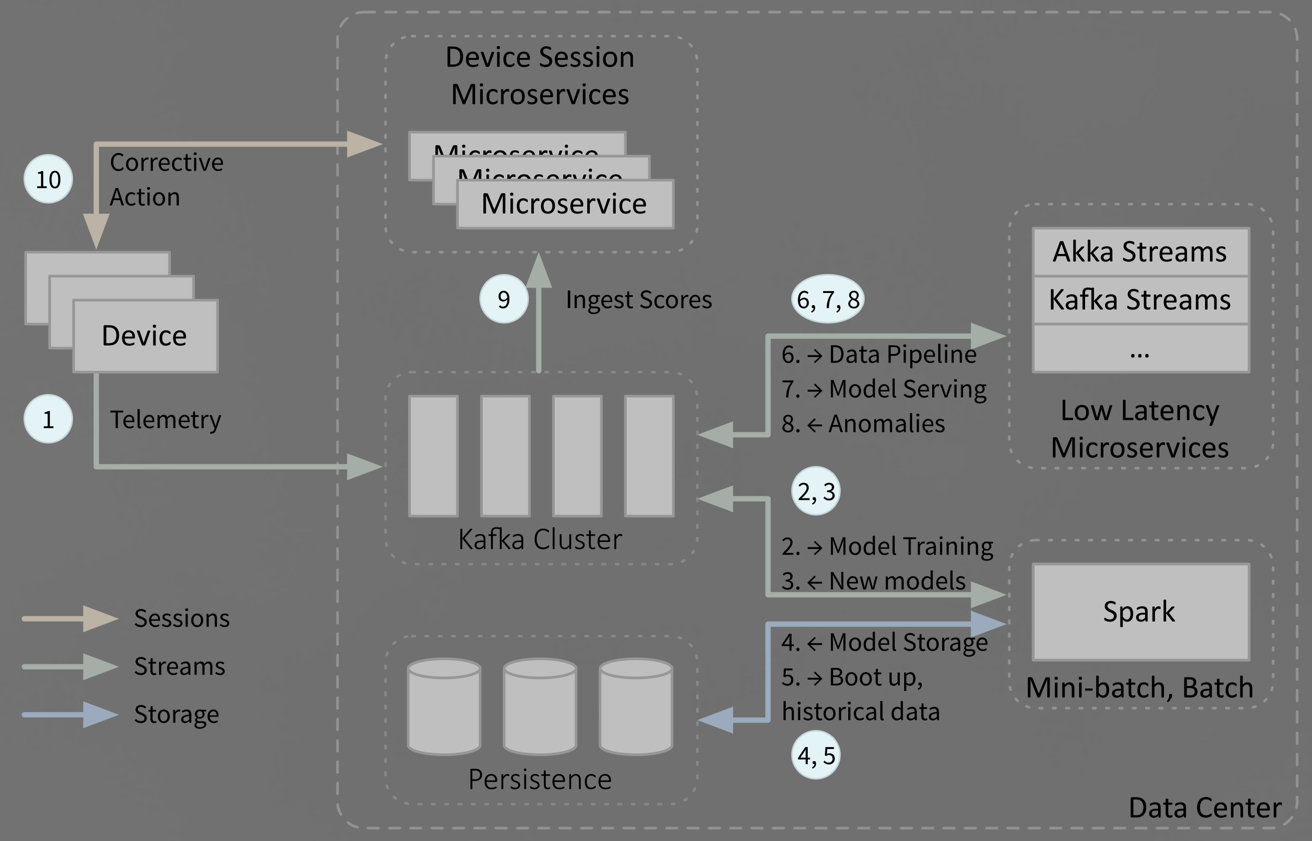





- Integration of Machine Learning/Artificial Intelligence with streaming is a common challenge right now

# Challenges

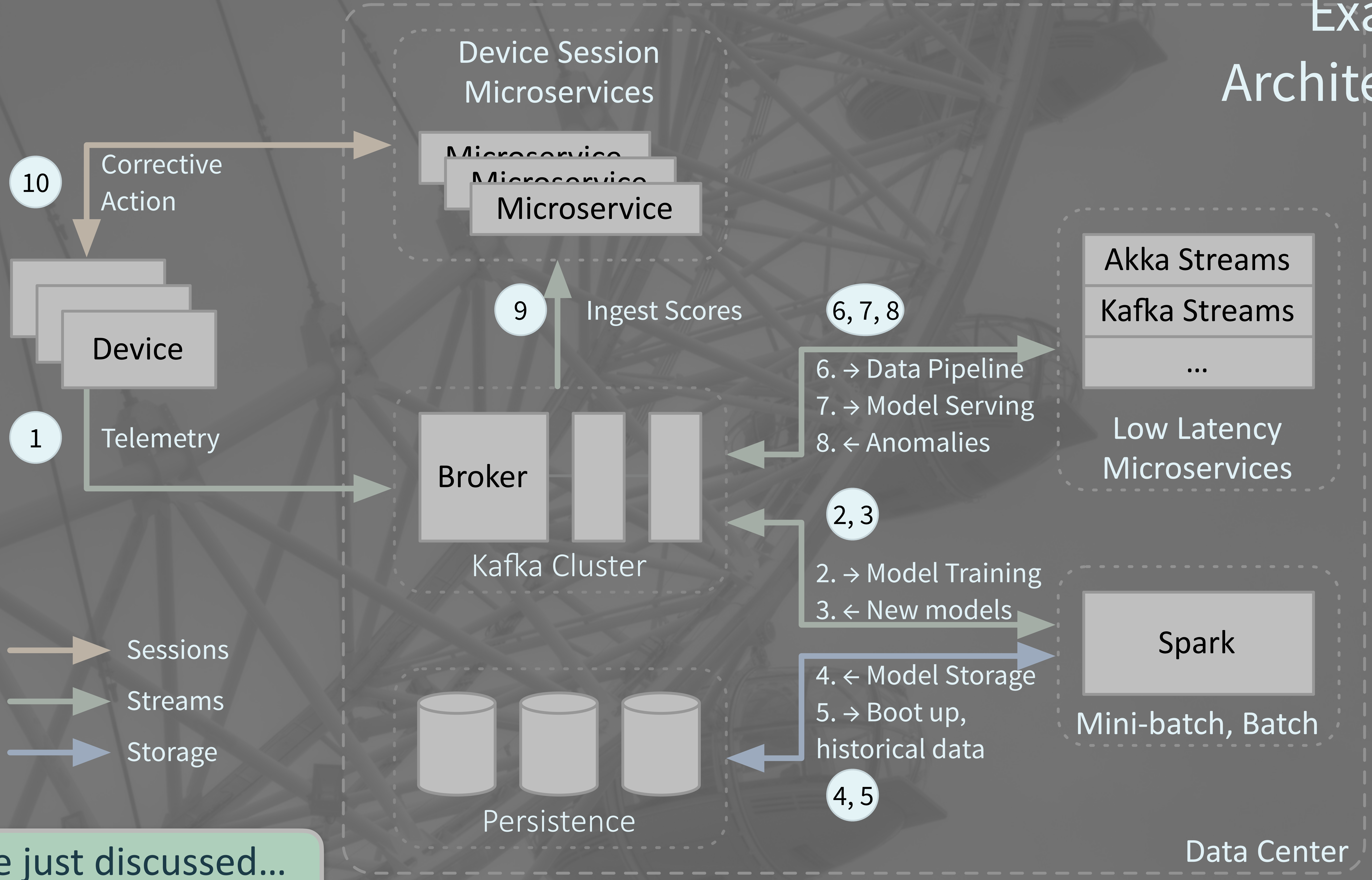
- Network overhead for telemetry ingestion too high?
- Model serving latency too long?
- Datacenter unavailable?
- Idea: Serve models on the device!



# Internet of Things

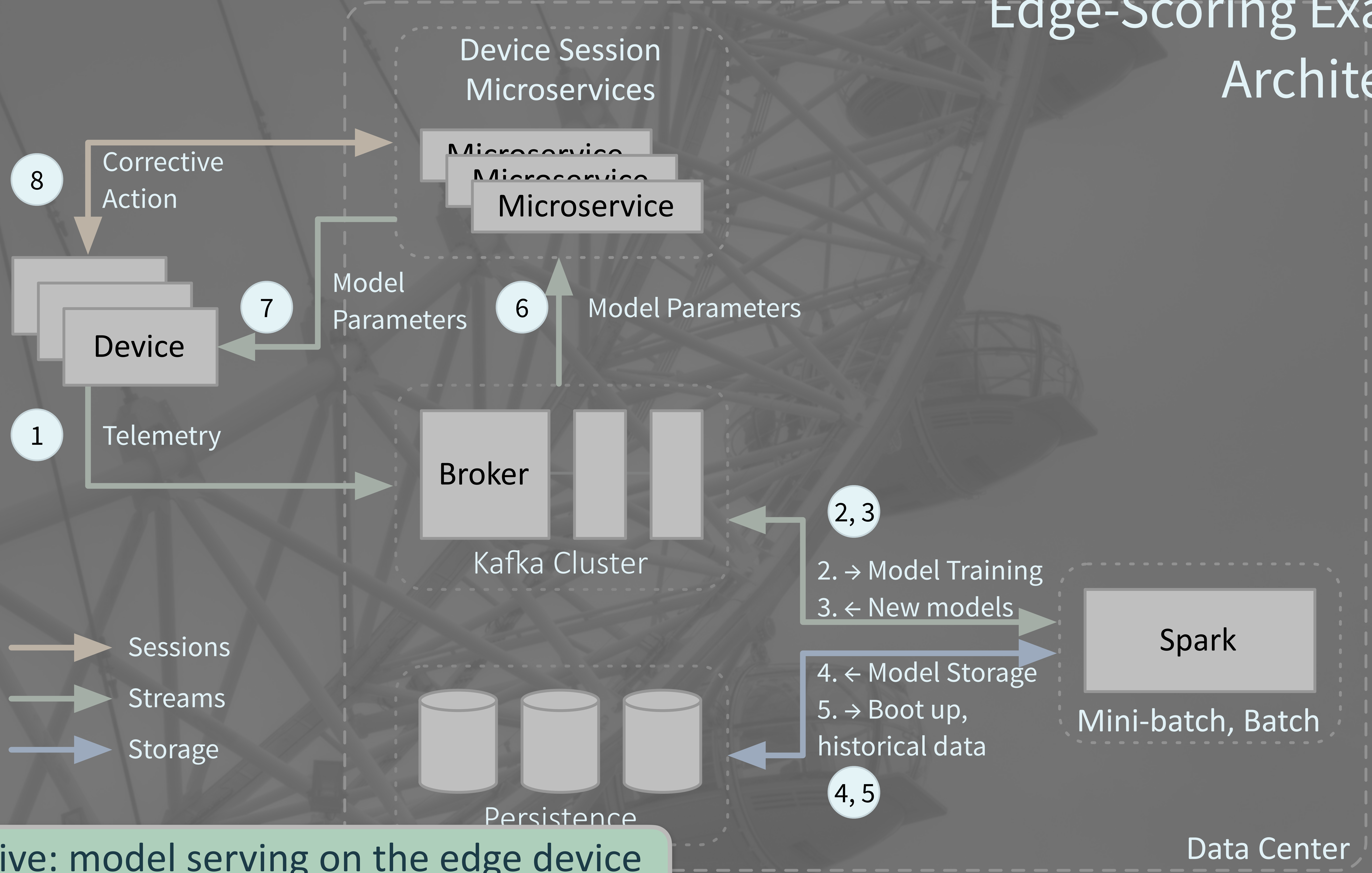
- Real-time consumer and industrial device and supply chain management at scale

# Example Architecture



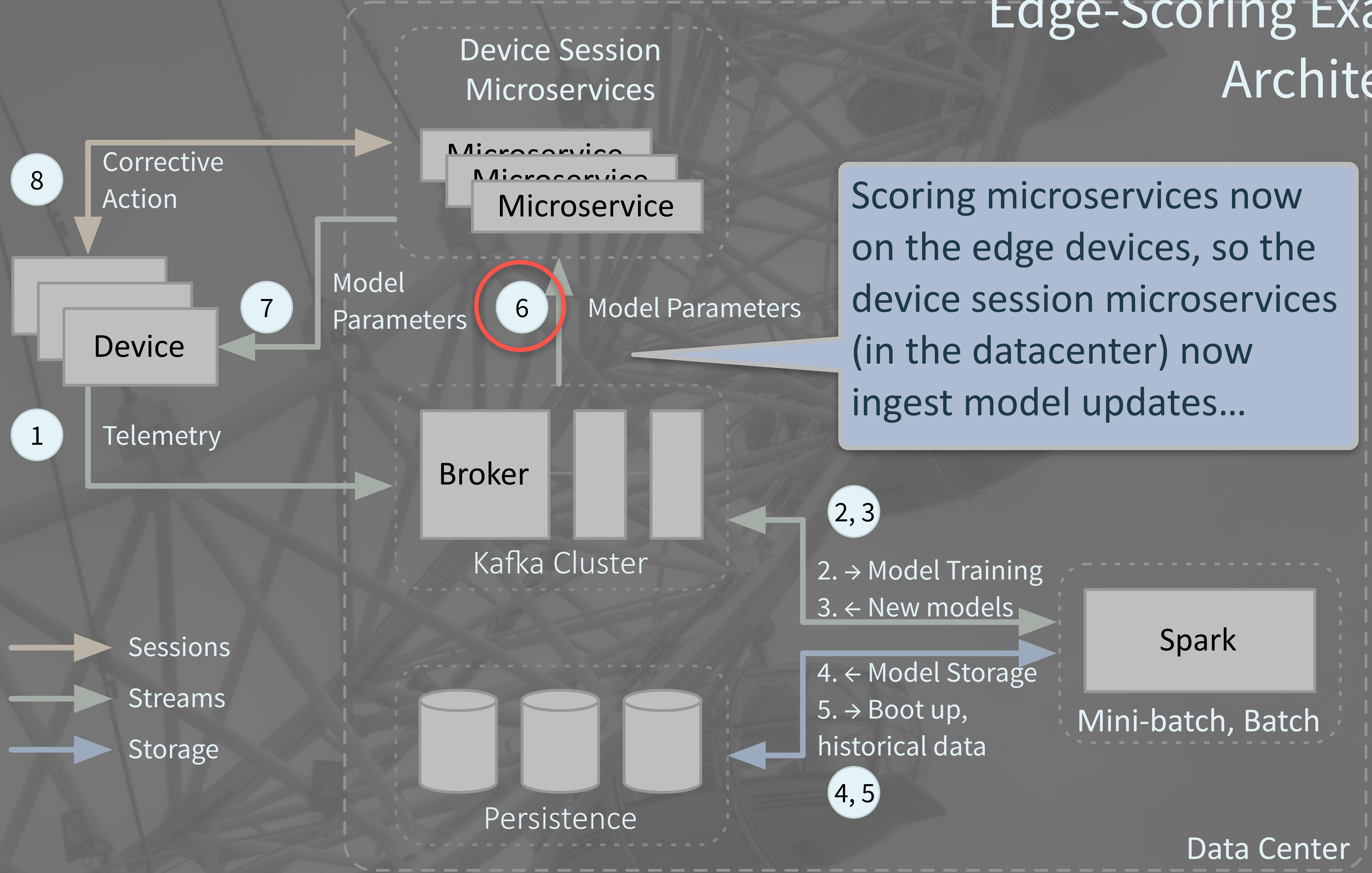

What we just discussed...

# Edge-Scoring Example Architecture



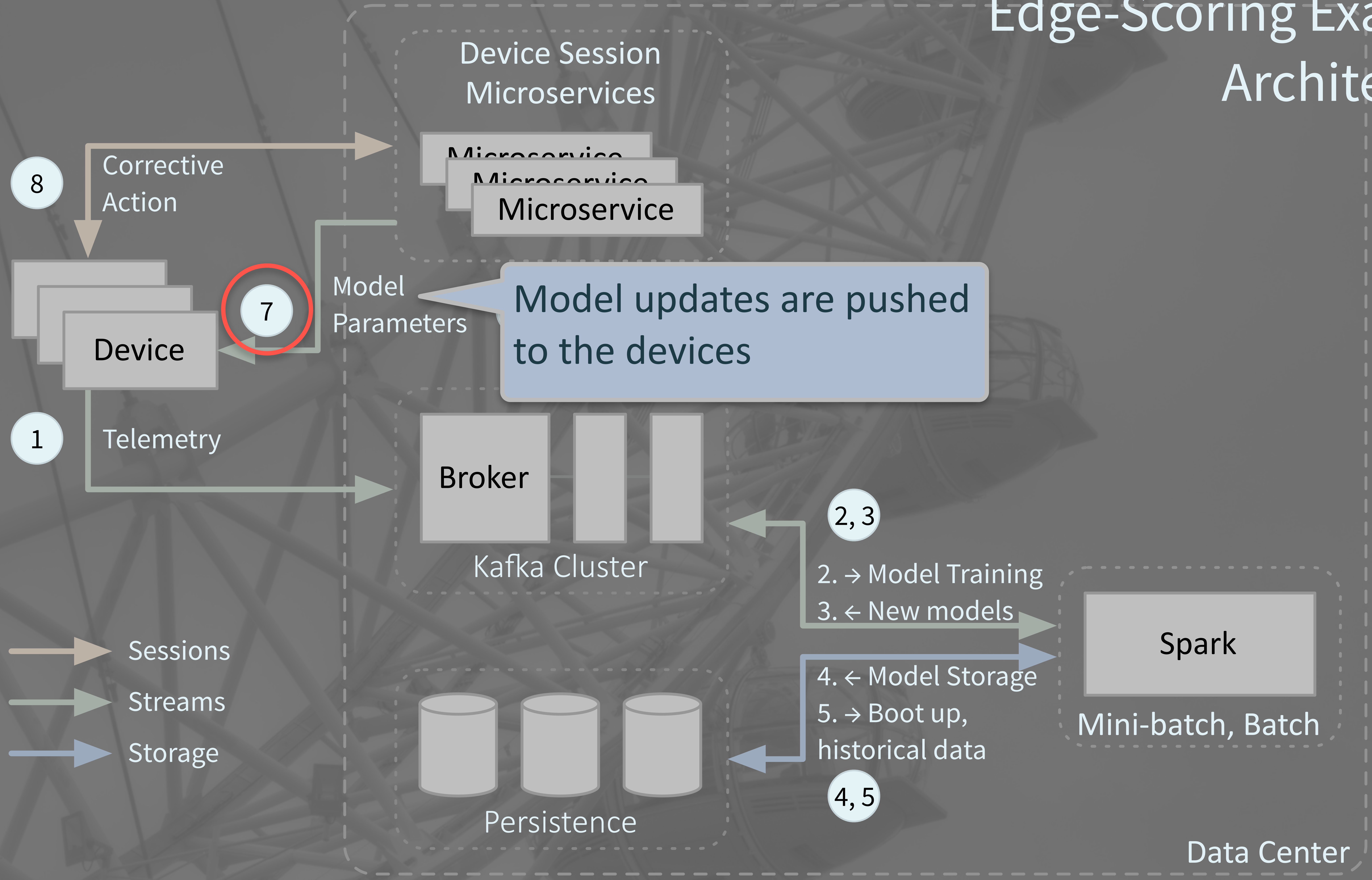

Alternative: model serving on the edge device

# Edge-Scoring Example Architecture

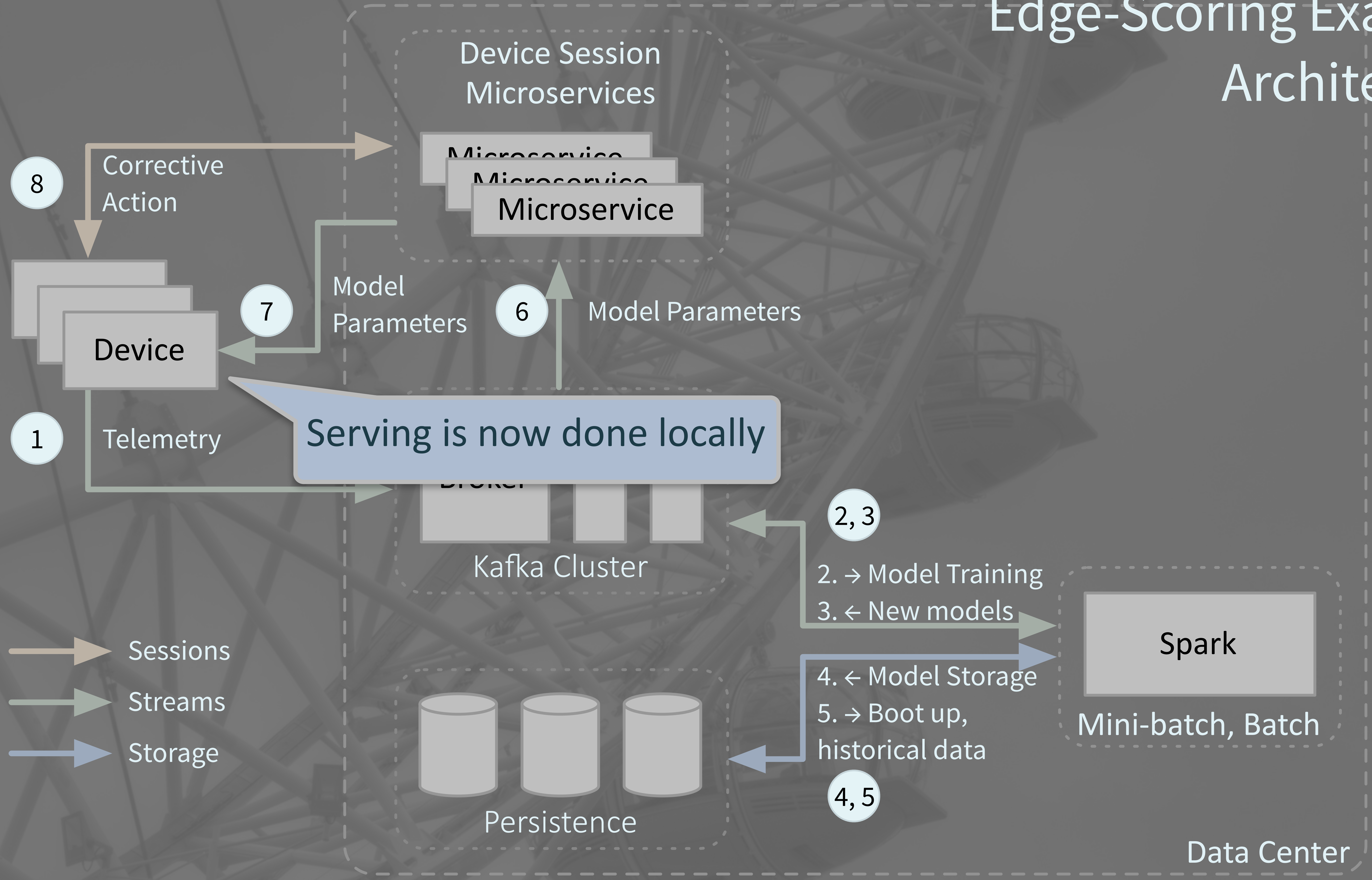



Scoring microservices now on the edge devices, so the device session microservices (in the datacenter) now ingest model updates…

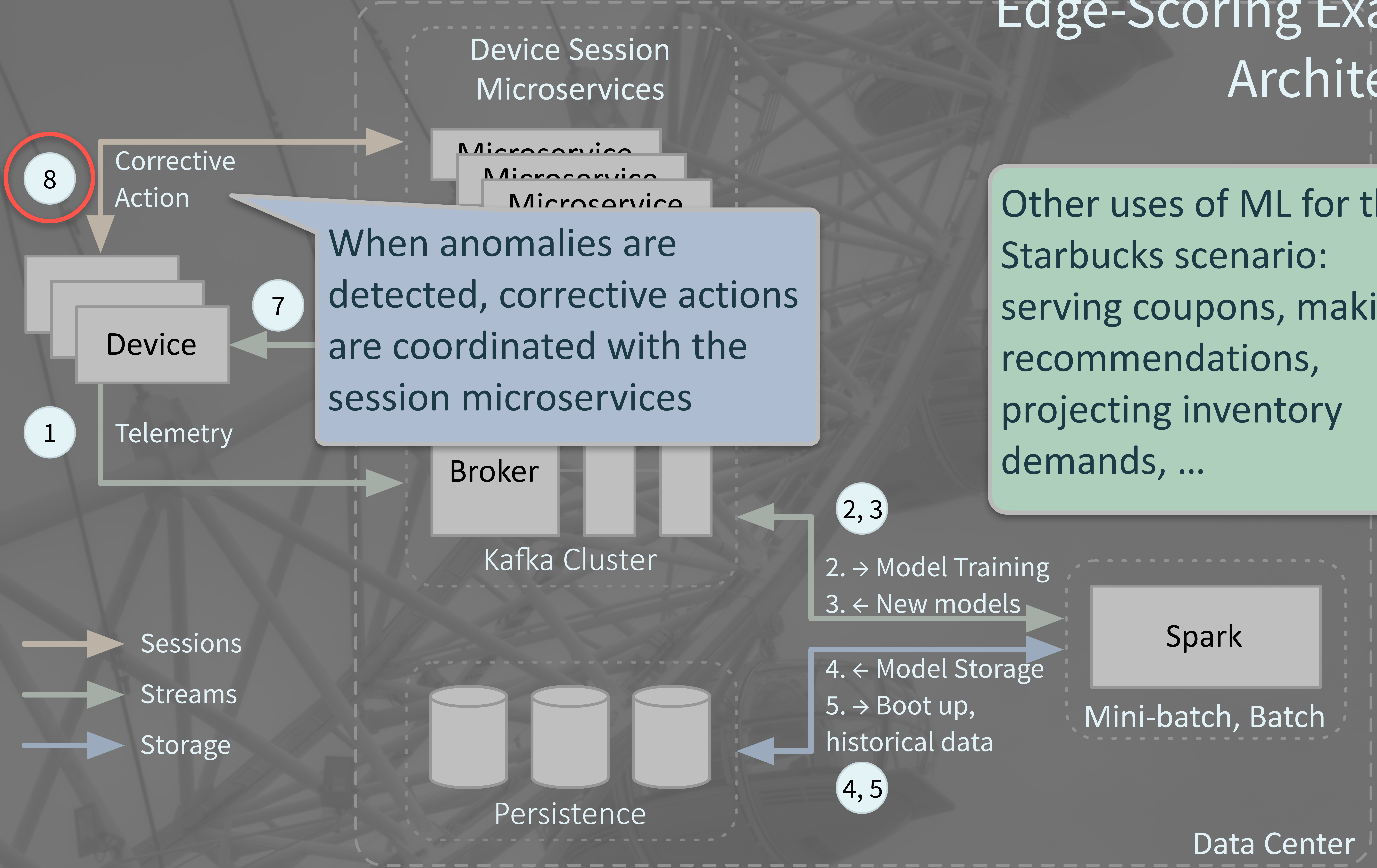
# Edge-Scoring Example Architecture



# Edge-Scoring Example Architecture



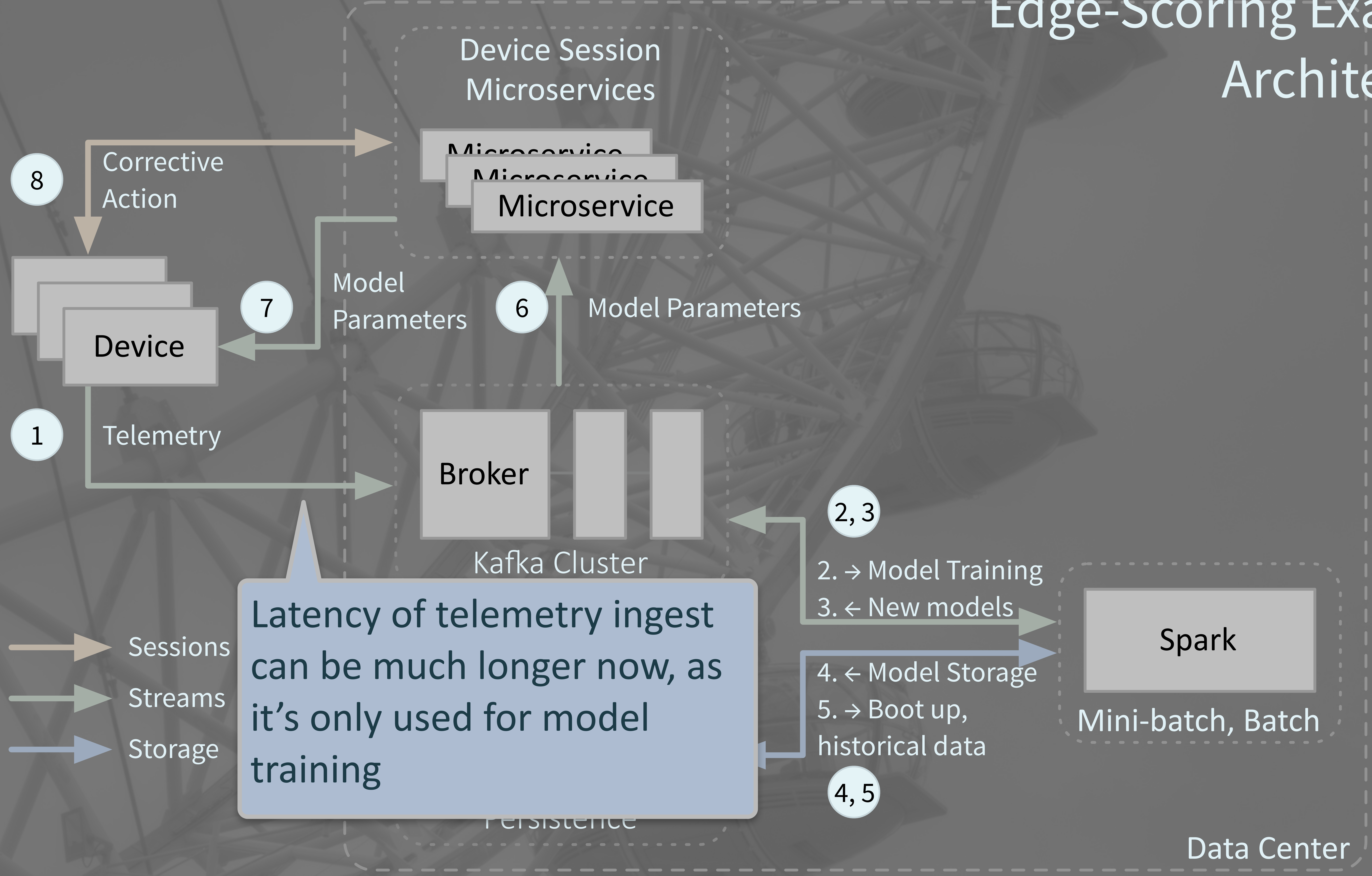
# Edge-Scoring Example Architecture



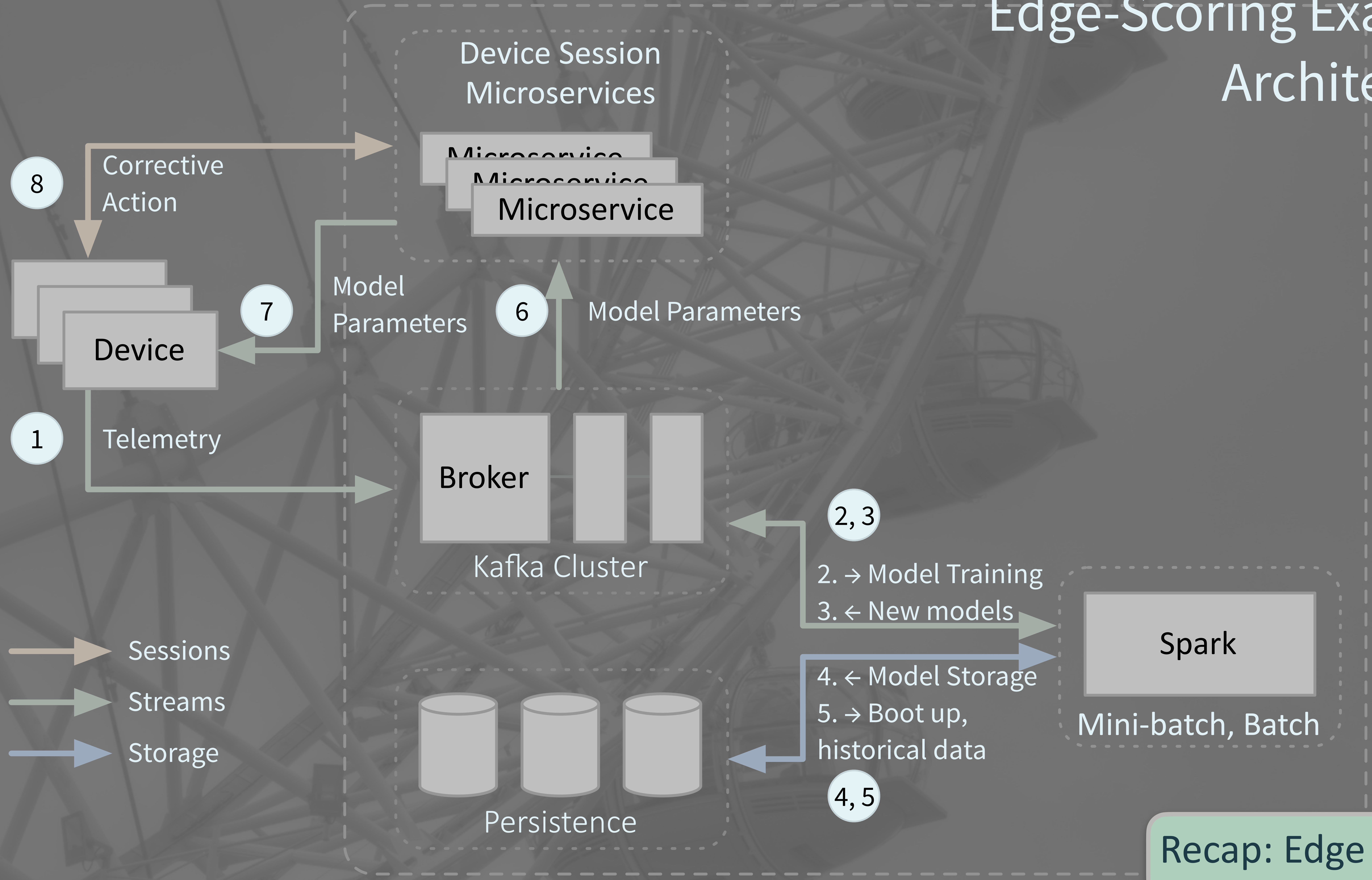

When anomalies are detected, corrective actions are coordinated with the session microservices

Other uses of ML for this Starbucks scenario: serving coupons, making recommendations, projecting inventory demands, …

# Edge-Scoring Example Architecture



# Edge-Scoring Example Architecture




Recap: Edge Serving

# Fas… …ases

Batch changed to streaming for competitive advantage

| Predictive Analytics | IoT | Real-time Personalization | Real-time Financial Processes |
|---|---|---|---|
| Apply ML models to large volumes of device data to pre-empt failures / outages | Real-time consumer and industrial Device and Supply Chain management at scale | Real-time marketing based on behavior, location, inventory levels, product promotions, etc. | Drive better business outcomes through real-time risk, fraud detection, compliance, audit, governance, etc. |
| Hewlett Packard Enterprise | Starbucks Coffee | Royal Caribbean International | Capital One |

https://www.lightbend.com/customers


# Technology Choices

# Technology Choices

- More than “faster” Hadoop…
- New architectures that merge data processing with microservices

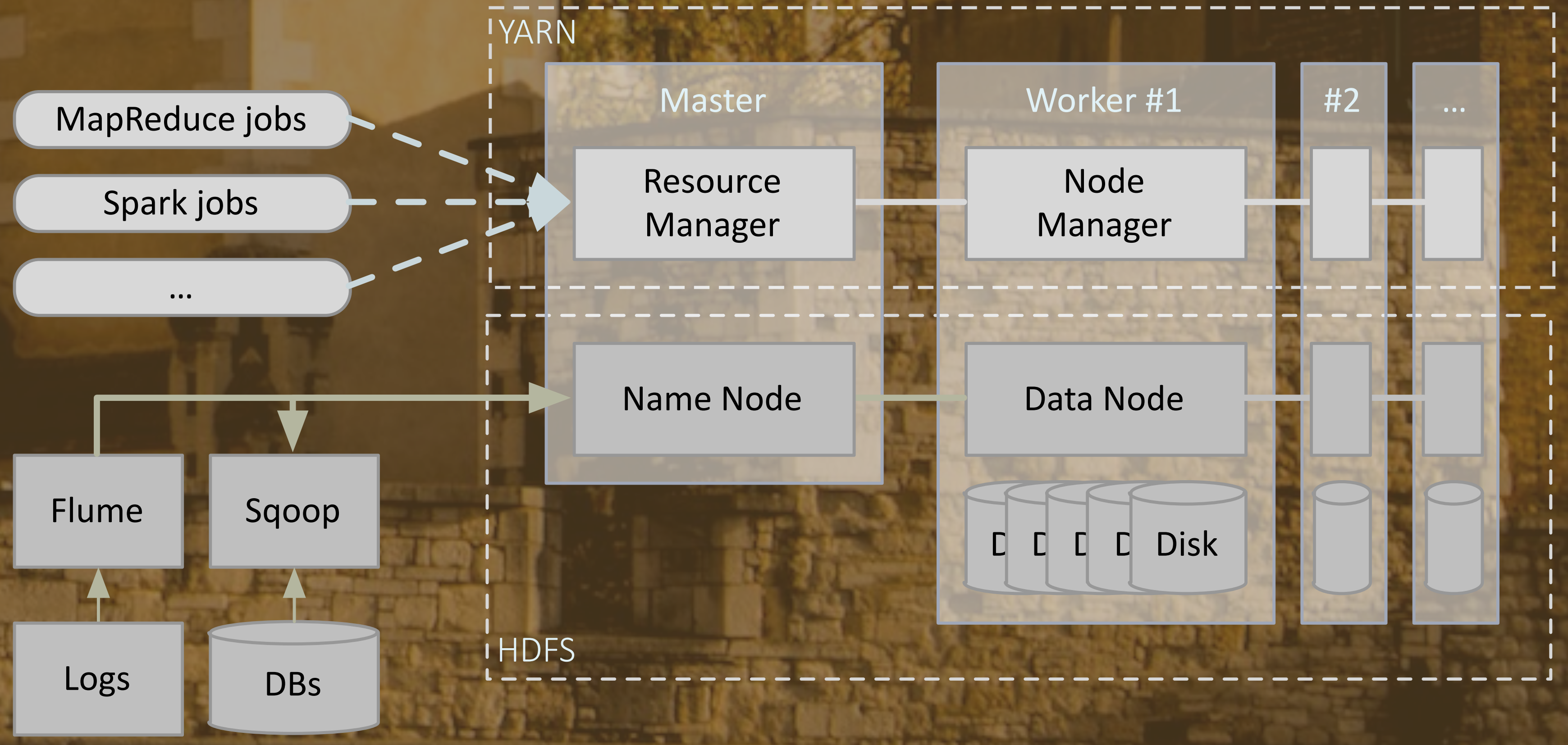

Recall Hadoop…

- Data warehouse replacement
- Historical analysis
- Interactive exploration
- Offline training of machine learning models
- …

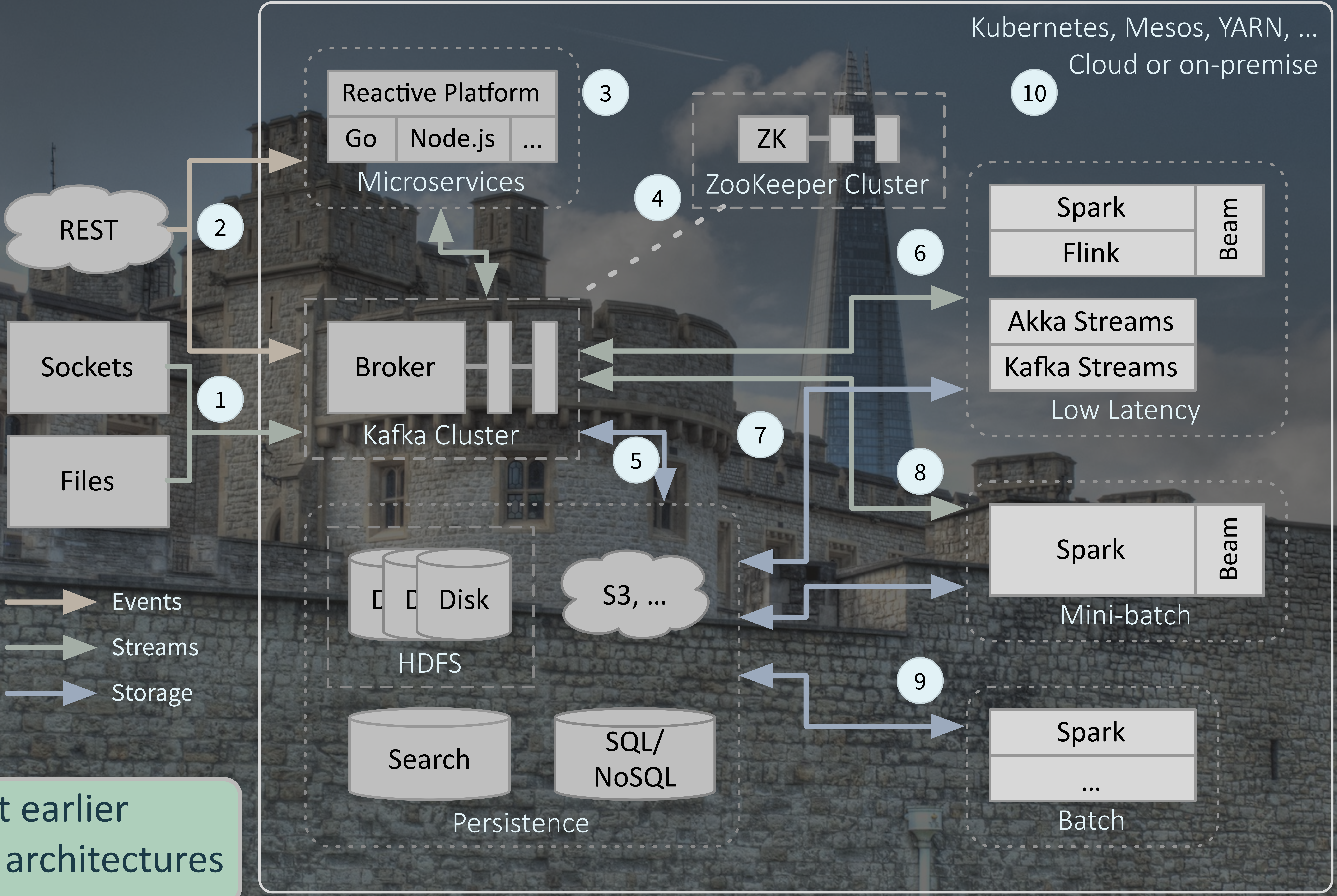
MapReduce jobs
Spark jobs
...
submit to...
YARN
Master Node
Resource Manager
Worker Node #1
Node Manager
#2
...
Name Node
Data Node
Disk
HDFS

Resource Management
Compute
MapReduce jobs
Spark jobs
...
submit to...
YARN
Master Node
Resource Manager
Worker Node #1
Node Manager
#2
...
Name Node
Data Node
Disk
HDFS
Storage
Optimized for storing lots of data *at rest*, with subsequent processing, but not optimized for data *in motion*.

- Hadoop is ideal for batch and interactive apps
- … but also constrained by that model



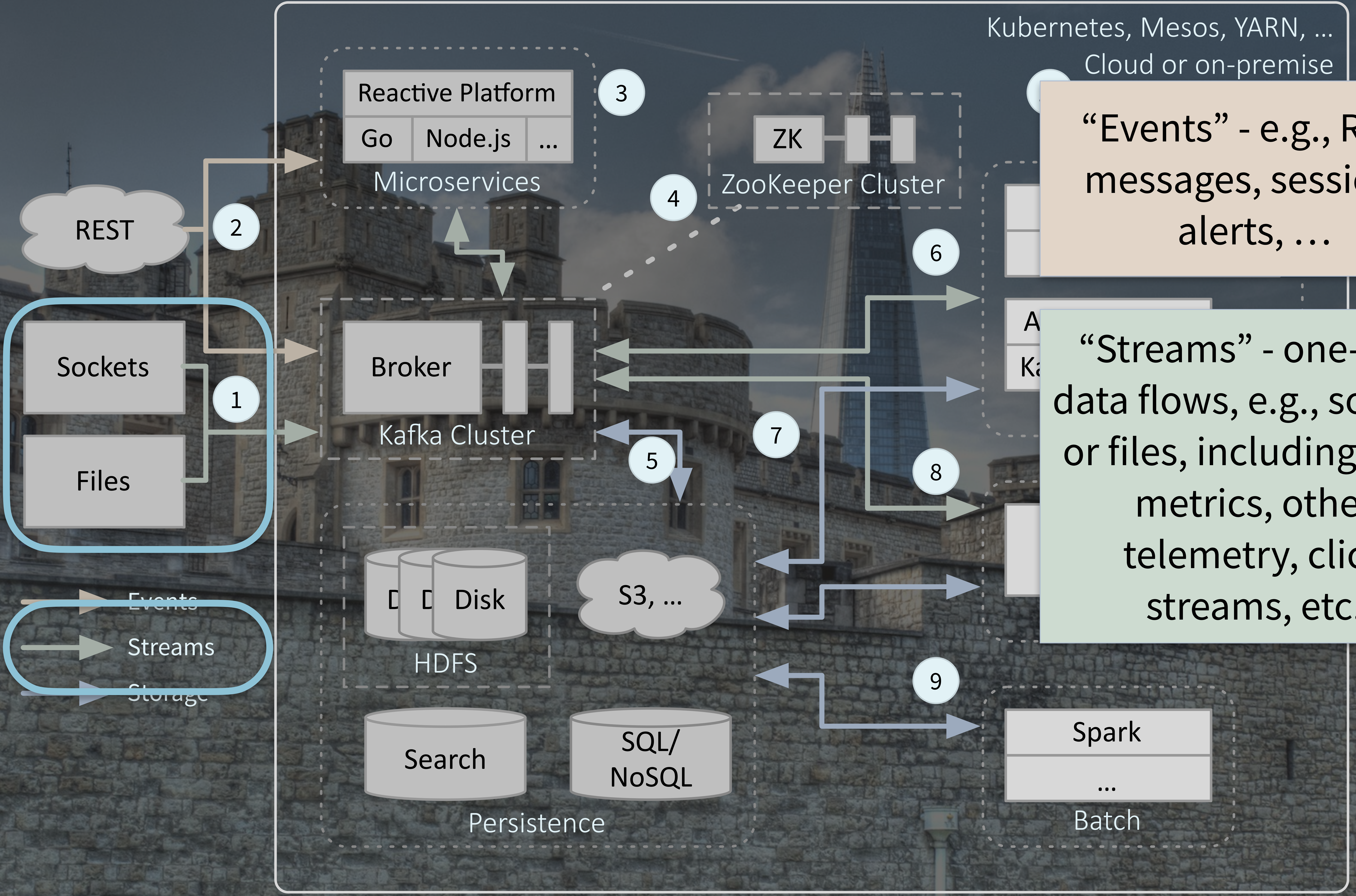
# New Fast Data Architecture




Flesh out earlier example architectures


Kubernetes, Mesos, YARN, …
Cloud or on-premise
10
Kubernetes and Mesos provide the job and resource management needed for dynamic, heterogenous work loads
While YARN can be used, it's not flexible enough for today's dynamic workloads
Deploy in the cloud or on premise
REST
2
Sockets
1
Files
Broker
Kafka Cluster
eeper Cluster
5
7
ms
ms
ency
Spark
Beam
Mini-batch
D
D
Disk
HDFS
S3, …
Events
Streams
Storage
9
Spark
…
Batch
Search
SQL/
NoSQL
Persistence


Kubernetes, Mesos, YARN, …
Cloud or on-premise
"Events" - e.g., REST messages, sessions, alerts, …
Reactive Platform
Go
Node.js
…
Microservices
3
ZK
ZooKeeper Cluster
4
REST
2
Sockets
1
Files
Broker
Kafka Cluster
6
Akka Streams
Kafka Streams
Low Latency
7
5
8
Spark
Beam
Mini-batch
Disk
S3, …
HDFS
Events
Streams
Storage
9
Spark
…
Batch
Search
SQL/
NoSQL
Persistence




"Events" - e.g., REST messages, sessions, alerts, …

"Streams" - one-way data flows, e.g., sockets or files, including logs, metrics, other telemetry, click streams, etc.

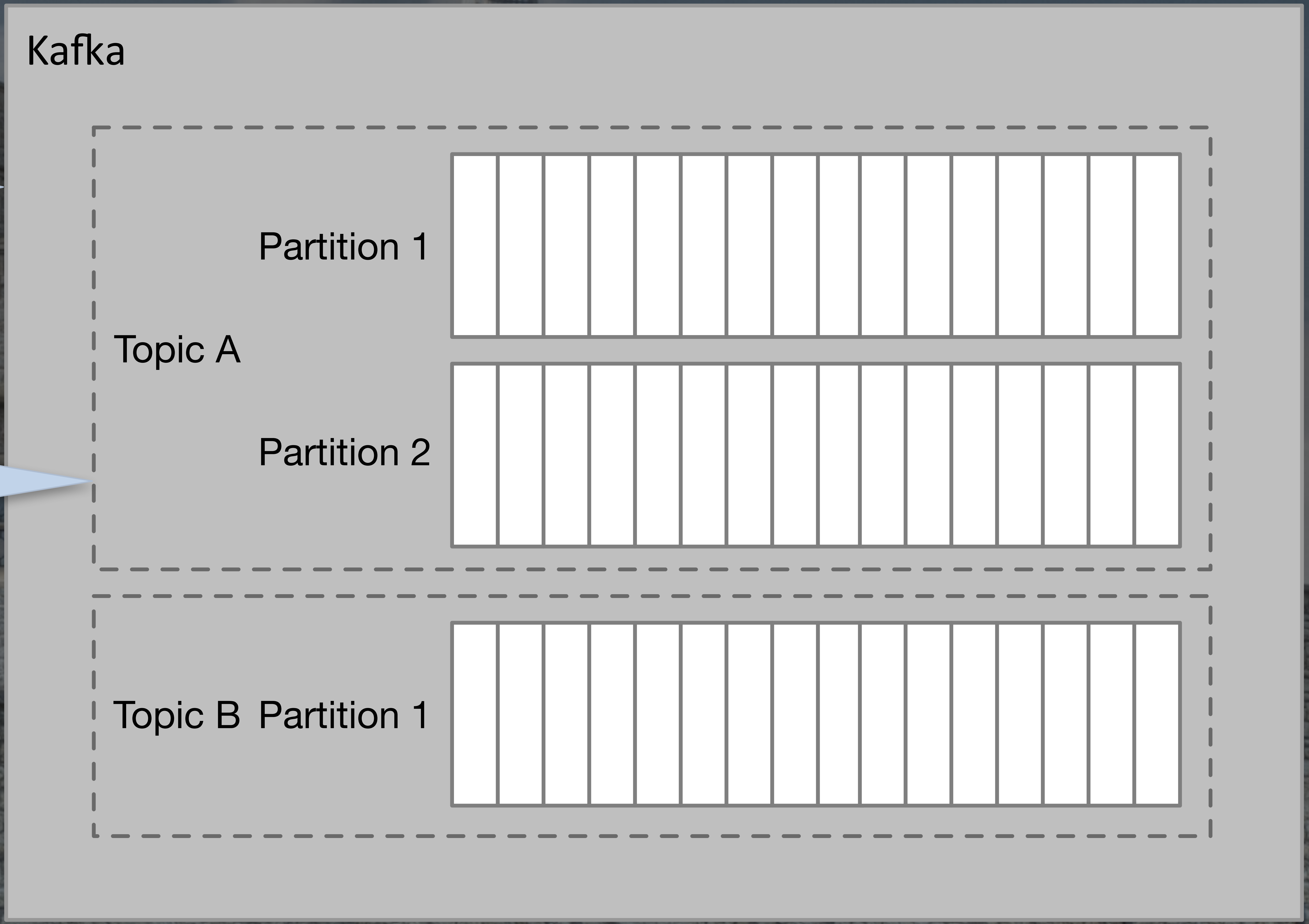

Kubernetes, Mesos, YARN, …
Cloud or on-premise
Reactive Platform
Go
Node.js
...
Microservices
3
ZK
ZooKeeper Cluster
4
REST
2
6
Sockets
1
Files
Broker
Kafka Cluster
7
5
8
Disk
HDFS
S3, …
9
Events
Storage
Search
SQL/
Batch
"Events" - e.g., REST messages, sessions, alerts, …
"Streams" - one-way data flows, e.g., sockets or files, including logs, metrics, other telemetry, click streams, etc.
"Storage" - JDBC, async reads/writes to storage
Each has different volumes, velocities, latency characteristics, protocols, etc.

Kubernetes, Mesos, YARN, ...
Cloud or on-premise
10
Reactive Platform
Go
Node.js
...
Microservices
3
ZK
ZooKeeper Cluster
4
REST
2
Sockets
1
Files
Broker
Kafka Cluster
5
Kafka: the data backplane. Deployed as a cluster of “Brokers” for scalability, resiliency.
Beam
8
Spark
Beam
Mini-batch
9
Spark
...
Batch
Disk
HDFS
S3, ...
Search
SQL/ NoSQL
Persistence
Events
Streams
Storage

# Why Kafka?

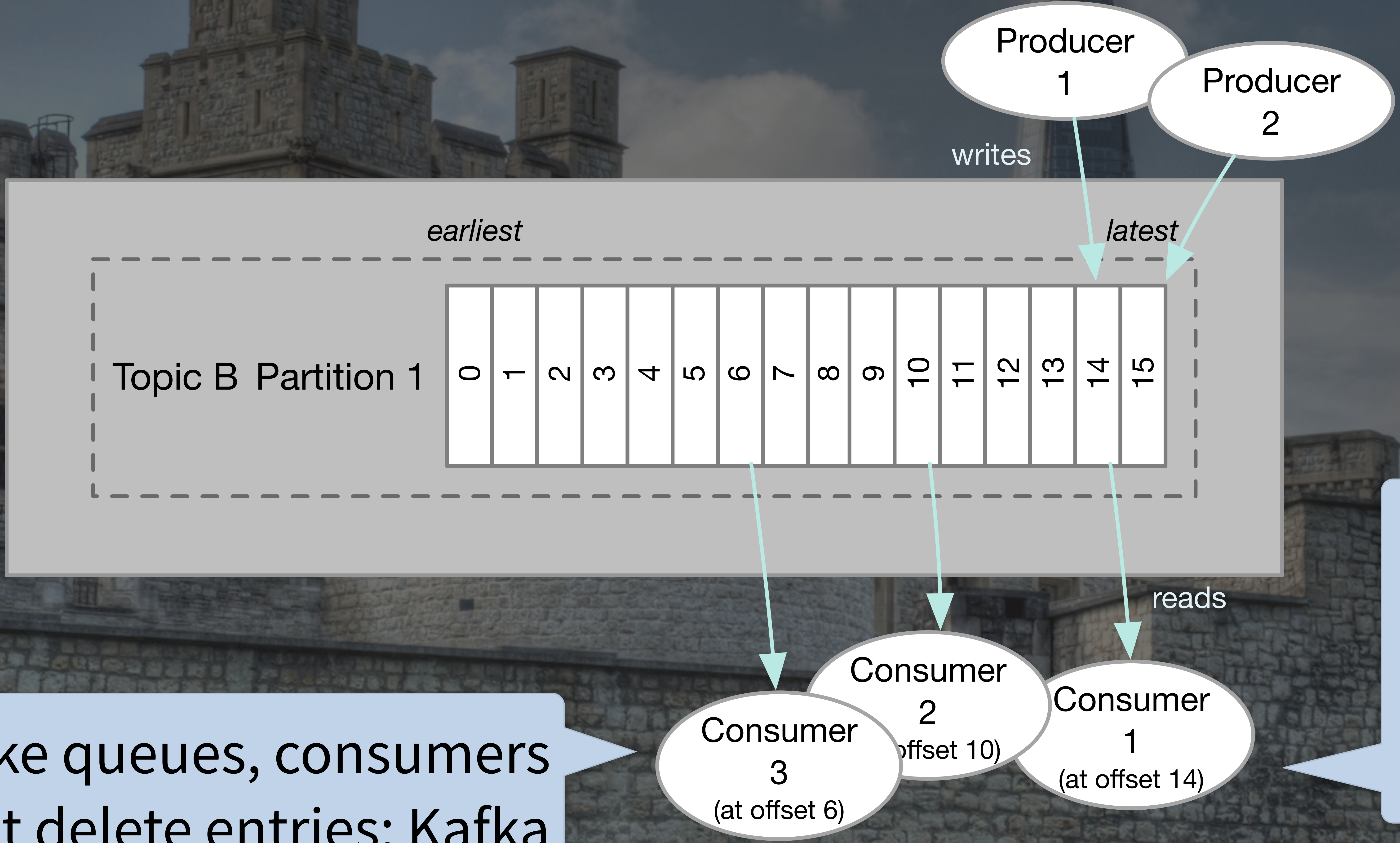
Organized into topics

Topics are partitioned, replicated, and distributed



# Why Kafka?

Logs, *not* queues!

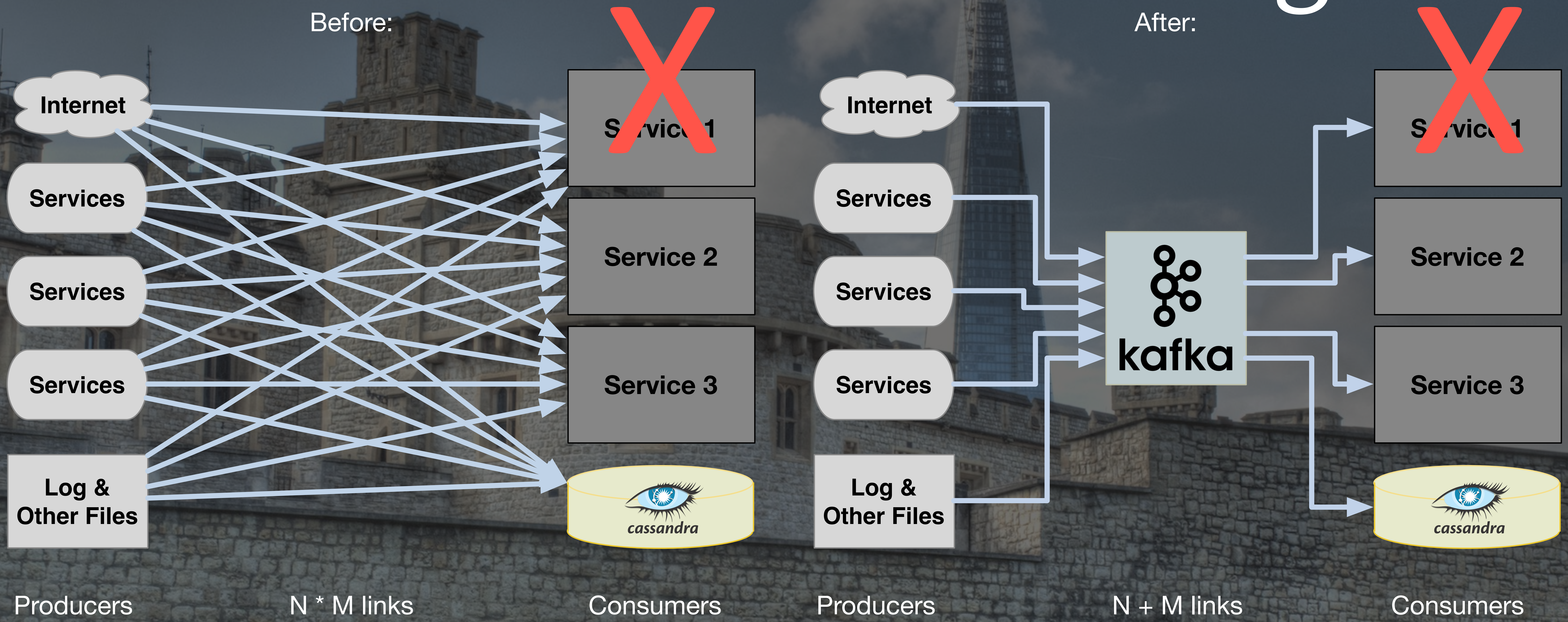



M Producers

N Consumers, who start reading where they want

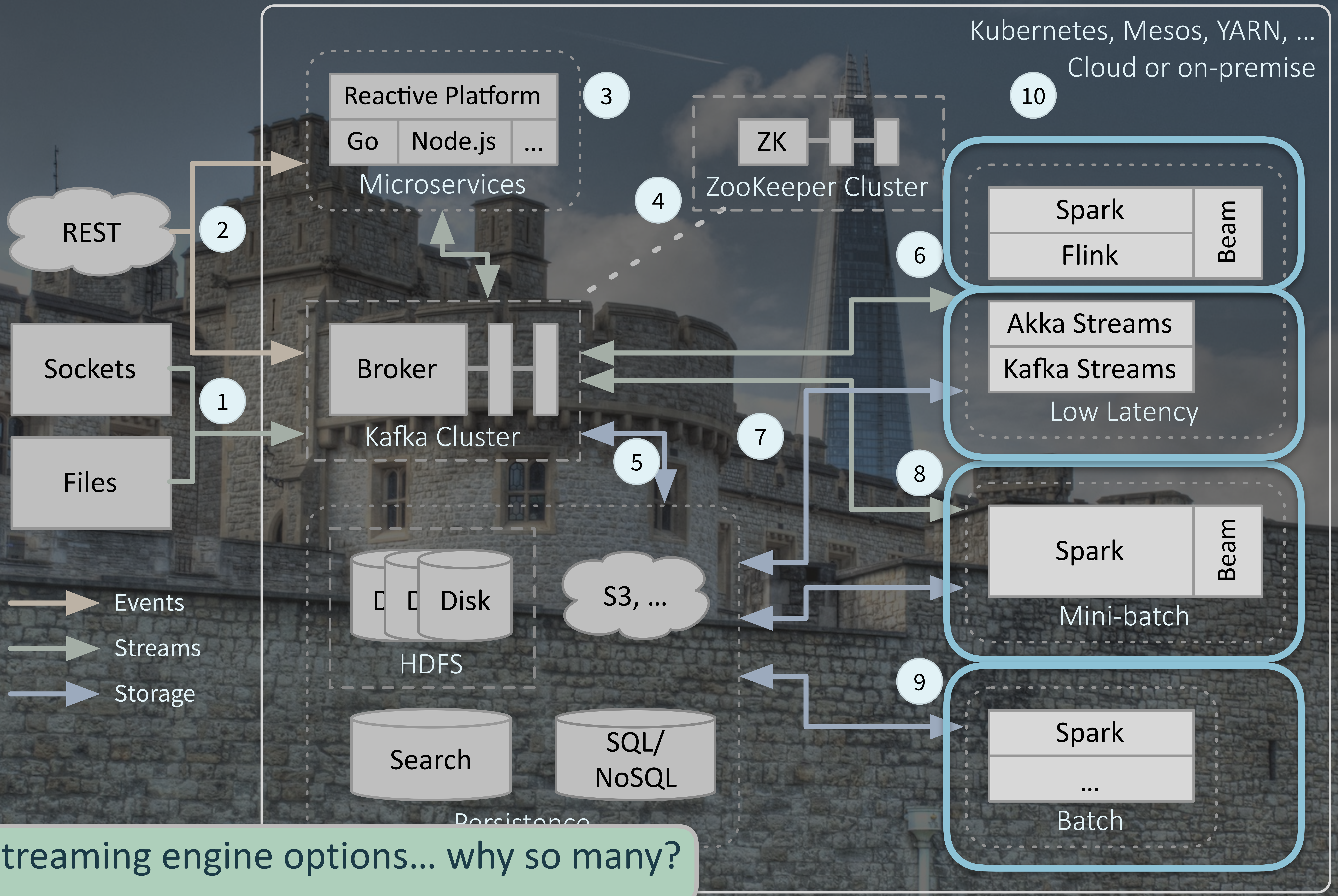
Unlike queues, consumers don't delete entries; Kafka manages their lifecycles

# Using Kafka




Messy and fragile; what if "Service 1" goes down?

Simpler and more robust! Loss of Service 1 means no data loss.




Lots of streaming engine options… why so many?

# You need choices

- Latency: how low?
- Volume per unit time: how high?
- Data processing: which kinds?
- Build, deploy, and manage services: what are your preferences?

The streaming engines form two groups:
Run as distributed services
You submit jobs, they are partitioned into tasks
Kubernetes, Mesos, YARN, …
Cloud or on-premise
10
Reactive Platform
Go
Node.js
Microservices
REST
2
Sockets
1
Files
Broker
Kafka Cluster
6
Spark
Flink
Beam
Akka Streams
Kafka Streams
Low Latency
8
Spark
Beam
Mini-batch
9
Spark
…
Batch
Disk
HDFS
S3, …
Search
SQL/ NoSQL
Persistence
Events
Streams
Storage

The streaming engines form two groups:
Kubernetes, Mesos, YARN, …
Cloud or on-premise
10
Reactive Platform
Go
Node.js
…
Microservices
3
ZK
ZooKeeper Cluster
4
REST
2
Sockets
1
Files
Broker
Kafka Cluster
Libraries you embed in your microservices
6
Spark
Flink
Beam
Akka Streams
Kafka Streams
Low Latency
8
Spark
Beam
Mini-batch
Disk
HDFS
S3, …
Search
SQL/ NoSQL
Persistence
9
Spark
…
Batch
Events
Streams
Storage

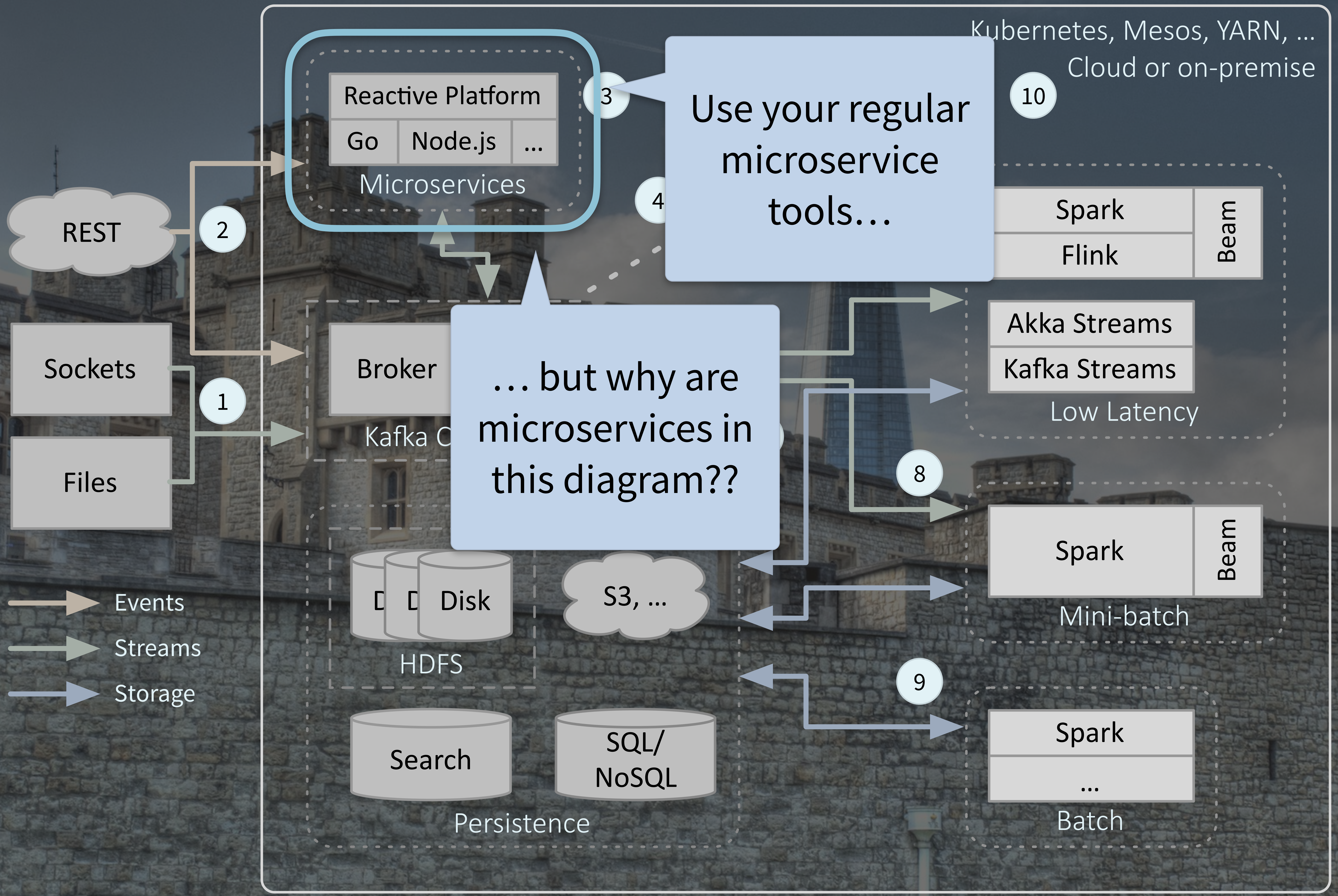
Kubernetes, Mesos, YARN, …
Cloud or on-premise
10
Reactive Platform
Go
Node.js
Microservices
REST
2
Sockets
1
Files
Broker
Kafka Cluster
New, low-latency *Structured Streaming*
Spark
Beam
6
Akka Streams
Kafka Streams
Low Latency
Older *mini-batch* streaming
8
Spark
Beam
Mini-batch
Disk
HDFS
S3, …
Events
Streams
Storage
Full batch support (replaced MapReduce)
9
Spark
…
Batch
Search
Rich SQL and Machine Learning options.

Kubernetes, Mesos, YARN, …
Cloud or on-premise
10
Reactive Platform
Go
Node.js
Microservices
3
luster
REST
2
Sockets
1
Files
Broker
Kafka Cluster
5
Low latency Spark alternative. Also uses services to partition the data and tasks across a cluster
Spark
Flink
Beam
6
Akka Streams
Kafka Streams
Low Latency
8
Spark
Beam
Mini-batch
9
Spark
…
Batch
Disk
HDFS
S3, …
Search
SQL/ NoSQL
Persistence
Events
Streams
Richer streaming semantics, more mature low-latency support

Kubernetes, Mesos, YARN, …
Cloud or on-premise
10
Reactive Platform
Go
Node.js
…
Microservices
3
ZK
ZooKeeper Cluster
4
REST
2
Sockets
1
Files
Broker
Kafka Cluster
Part of the rich Akka ecosystem for general microservices
6
Spark
Flink
Beam
Akka Streams
Kafka Streams
Low Latency
8
Spark
Beam
Mini-batch
9
Spark
…
Batch
Disk
HDFS
S3, …
Search
SQL/ NoSQL
Persistence
Events
Streams
Storage


Kubernetes, Mesos, YARN, …
Cloud or on-premise
10
Reactive Platform
Go
Node.js
…
Microservices
3
ZK
ZooKeeper Cluster
4
REST
2
Sockets
1
Files
Broker
Kafka Cluster
Kafka-centric library for common data processing tasks
Spark
Flink
Beam
6
Akka Streams
Kafka Streams
Low Latency
8
Spark
Beam
Mini-batch
S3, …
Disk
HDFS
9
Spark
…
Batch
Search
SQL/
NoSQL
Persistence
Events
Streams
Storage

Kubernetes, Mesos, YARN, …
Cloud or on-premise
10
Reactive Platform
Go
Node.js
…
Microservices
3
ZK
ZooKeeper Cluster
4
REST
2
Sockets
1
Files
Broker
Kafka Cluster
5
Spark
Flink
Beam
6
Akka Streams
Kafka Streams
Low Latency
7
8
Spark
Beam
Mini-batch
9
Spark
…
Batch
Disk
HDFS
S3, …
Search
SQL/
NoSQL
Persistence
Standard APIs allow almost any storage you want



Kubernetes, Mesos, YARN, …
Cloud or on-premise
Reactive Platform
Go
Node.js
...
Microservices
Use your regular microservice tools…
… but why are microservices in this diagram??
REST
Sockets
Files
Broker
Kafka C
Spark
Flink
Beam
Akka Streams
Kafka Streams
Low Latency
Spark
Beam
Mini-batch
Spark
...
Batch
Disk
HDFS
S3, ...
Search
SQL/
NoSQL
Persistence
Events
Streams
Storage
1
2
3
4
8
9
10

# Why Microservices in Fast Data?

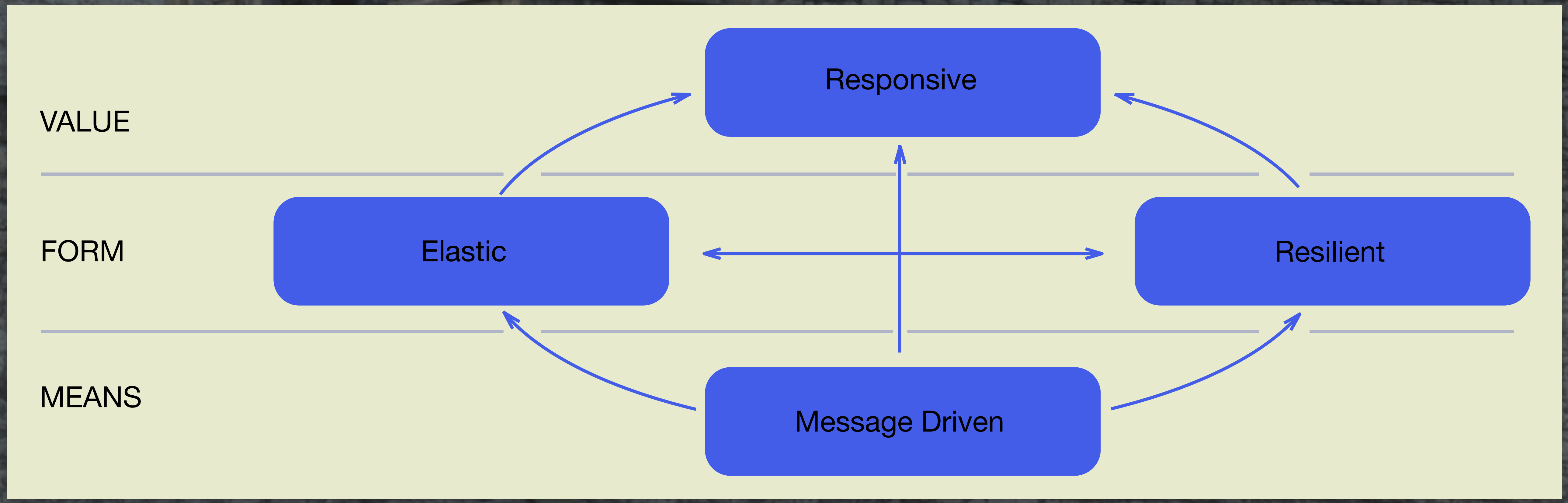
1. The trend is to run everything in big clusters using Kubernetes or Mesos
   - In the cloud or on-premise

# Why Microservices in Fast Data?

2. If streaming gives you information faster…
   - … you'll want quick access to it in your other services!

# Why Microservices in Fast Data?

3. Streaming raises the bar on data services
   - Compared to batch services, long-running streaming services must be more:
     - Scalable
     - Resilient
     - Flexible




https://www.reactivemanifesto.org/

# Why Microservices in Fast Data?

4. This leads to our last major point…

# Organizational Impact

# Organizational Impact

- Data engineers have to become good at highly-available microservices
- Microservice engineers have to become good at data
- … and Data scientists have to understand production issues

# The Past


Some overlap in concerns, architecture

# The Present


Much more overlap

# The Future?

Why? Since streams process data incrementally, there is less need for large-scale tools like Spark, Flink

… and using microservices for everything simplifies development, deployment, and operations

Microservices *for* Fast Data

## Much more microservice focused?


Lightbend
Fast Data Platform
lightbend.com/fast-data-platform


lightbend.com/fast-data-platform

lightbend.com/fast-data-platform


Dean Wampler, Ph.D.
dean@lightbend.com
@deanwampler
polyglotprogramming.com/talks
Lightbend