

Executive Briefing:

What it takes to use machine learning in fast data pipelines



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

Data Streaming, in General

lbnd.io/fast-data-ebook

O'REILLY®

Compliments of
Lightbend

Fast Data Architectures for Streaming Applications

Getting Answers Now from
Data Sets That Never End

2nd
Edition



Dean Wampler, PhD

Streaming Engines



Microservices



Machine Learning



Intelligent Management & Monitoring and Security



Data Backplane



Storage Options

HDFS

SQL, NoSQL

Cloud Storage (S3 etc)

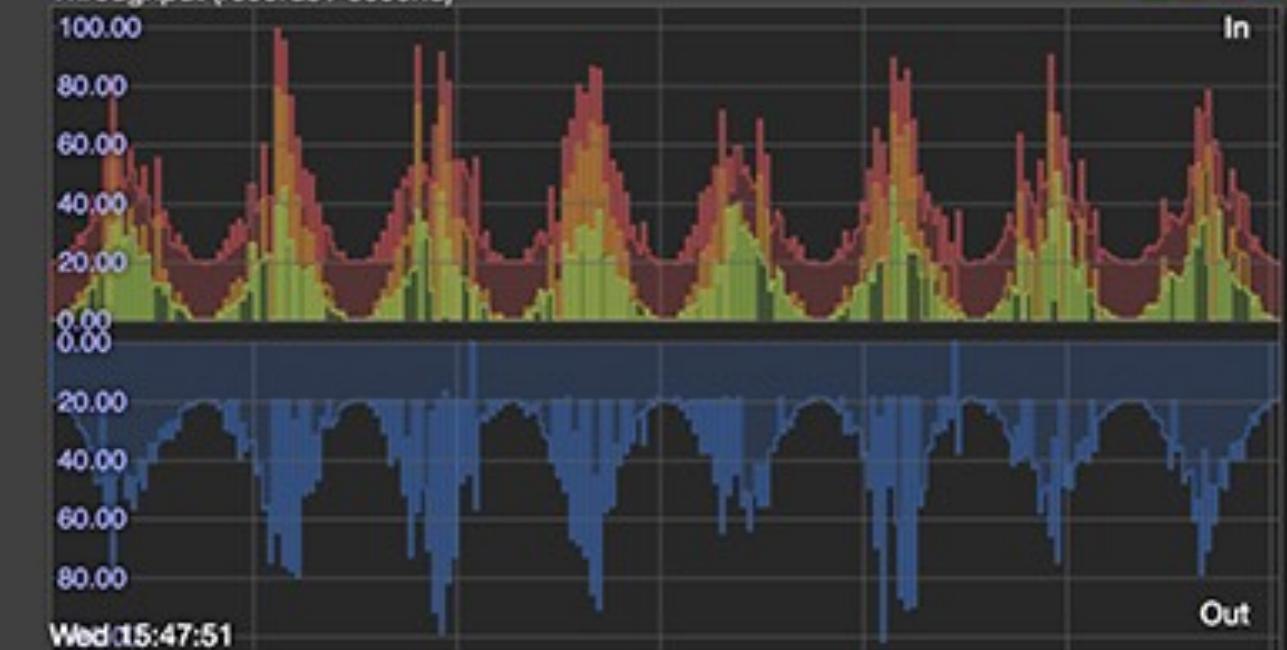
Search



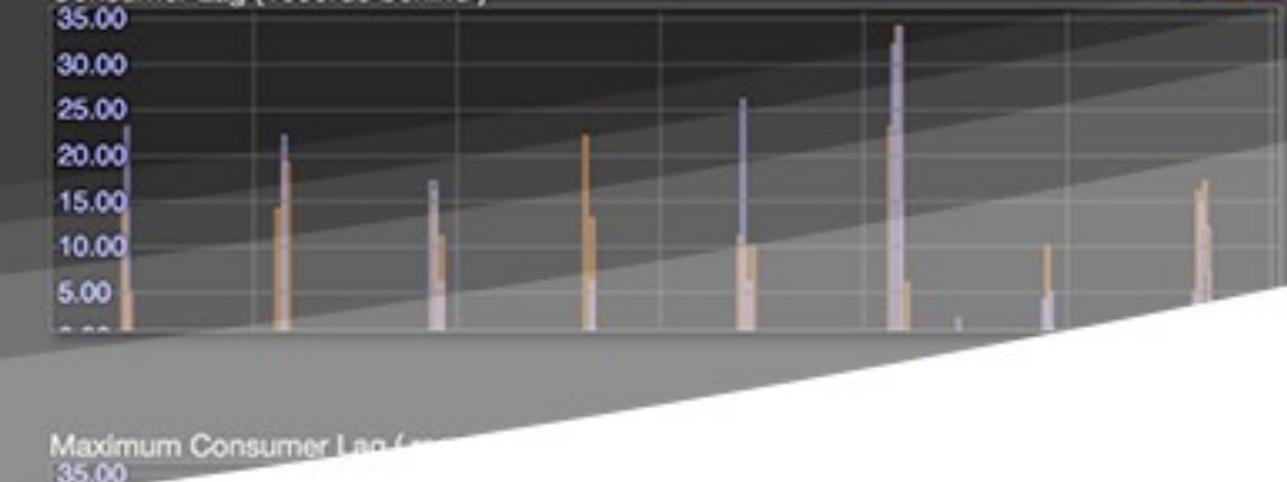


Metrics

Throughput (records / second)



Consumer Lag (records behind)



What we're up to at Lightbend...

lightbend.com/lightbend-pipelines-demo

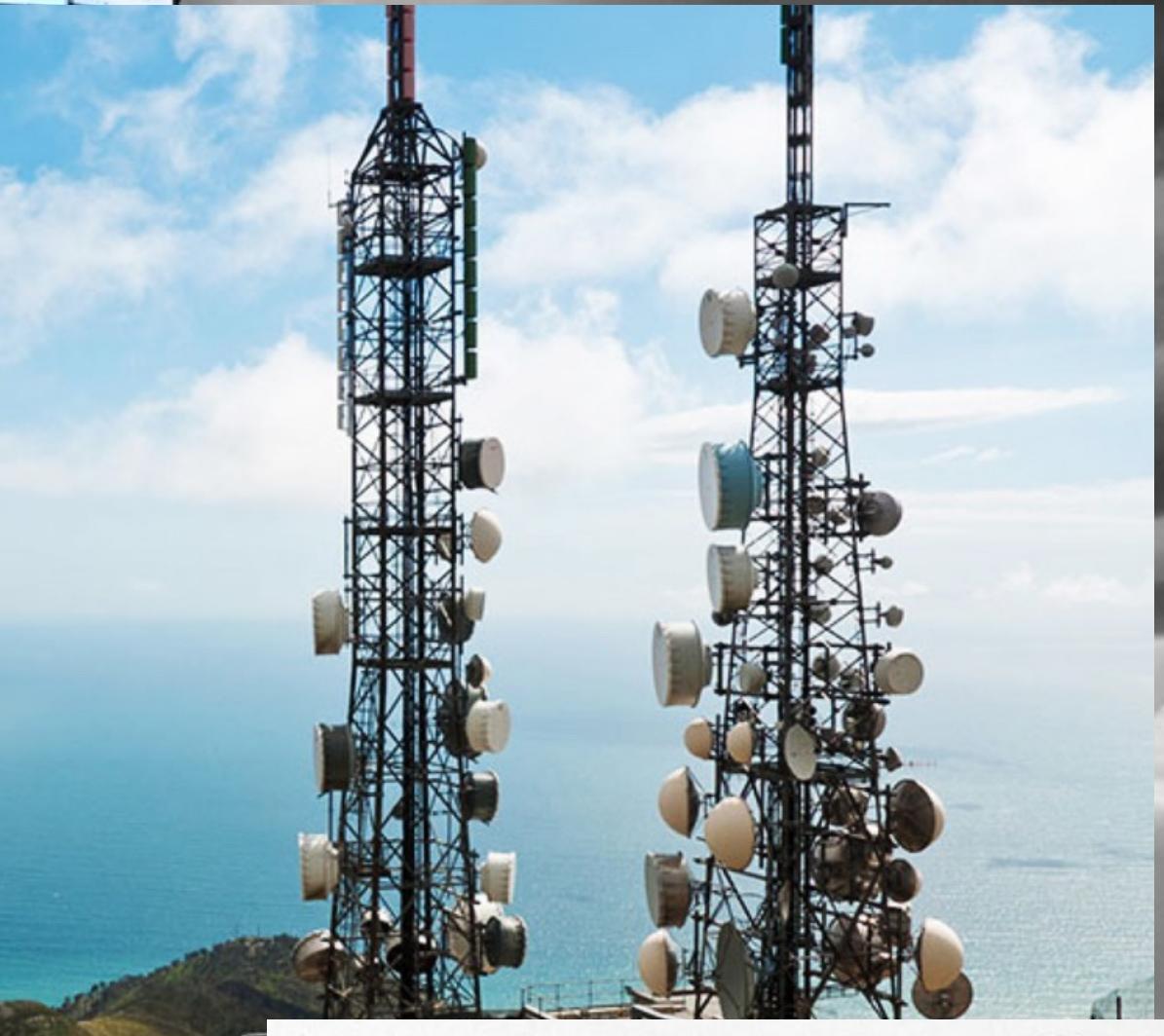
What We'll Discuss

- Batch vs. streaming... and why
- Data science vs. data engineering
- Where to use AI first
- Serving models in production
- Complete systems for ML/AI
- Pragmatic challenges



Batch vs. streaming... and why

Telecom



Finance



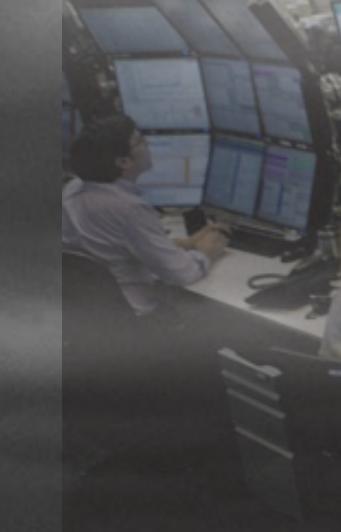
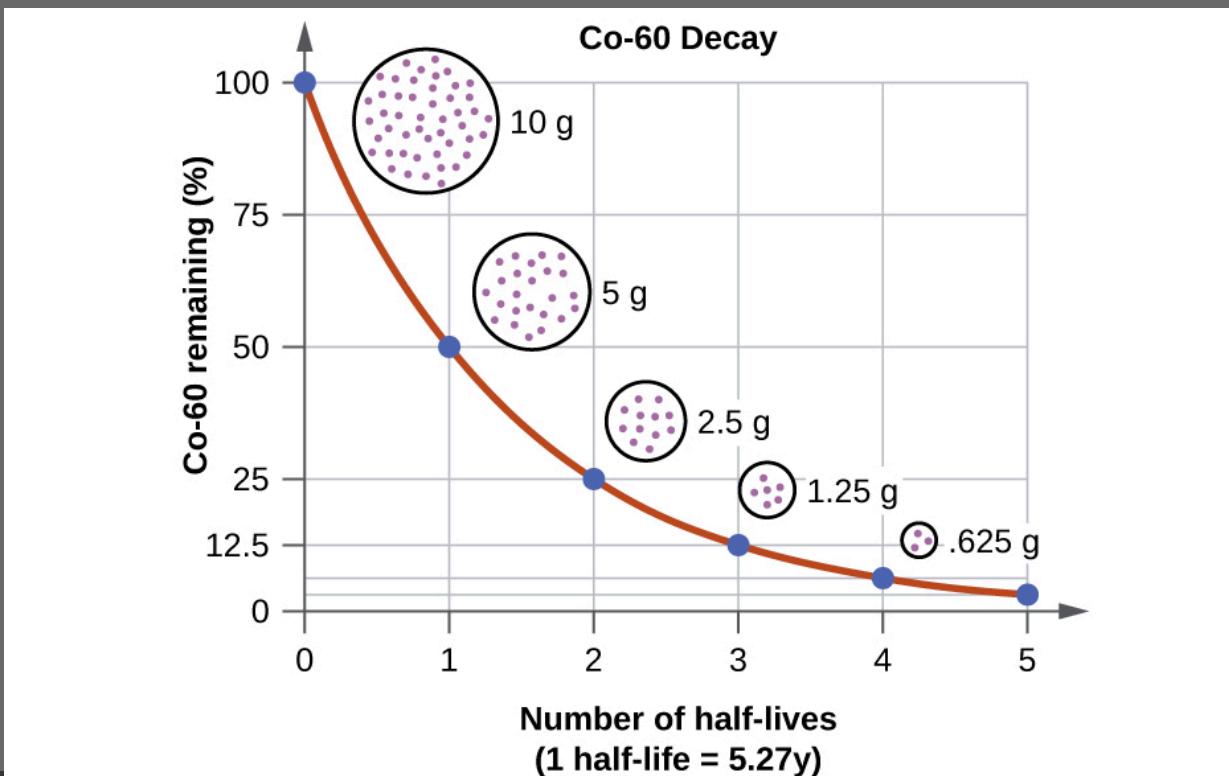
Energy

... and IoT

State of the art phone!



Information value has a half life; it decays with time





Data Science vs. Data Engineering



Data Science toolbox



Software
Engineering toolbox

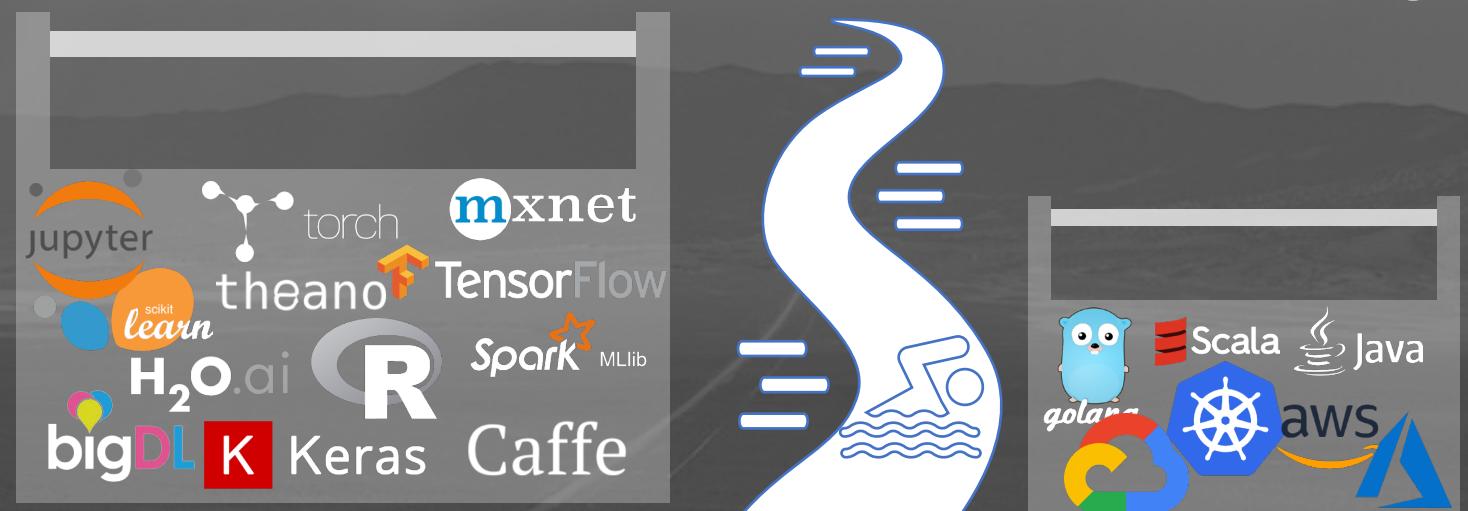
Data Scientists

- Comfortable with uncertainty
- Less process oriented
 - Iterative, experimental

Data Engineers

- Uncomfortable with uncertainty
- Process oriented
 - Agile Manifesto
 - ... which does not mention data!

<https://derwen.ai/s/6fqt>

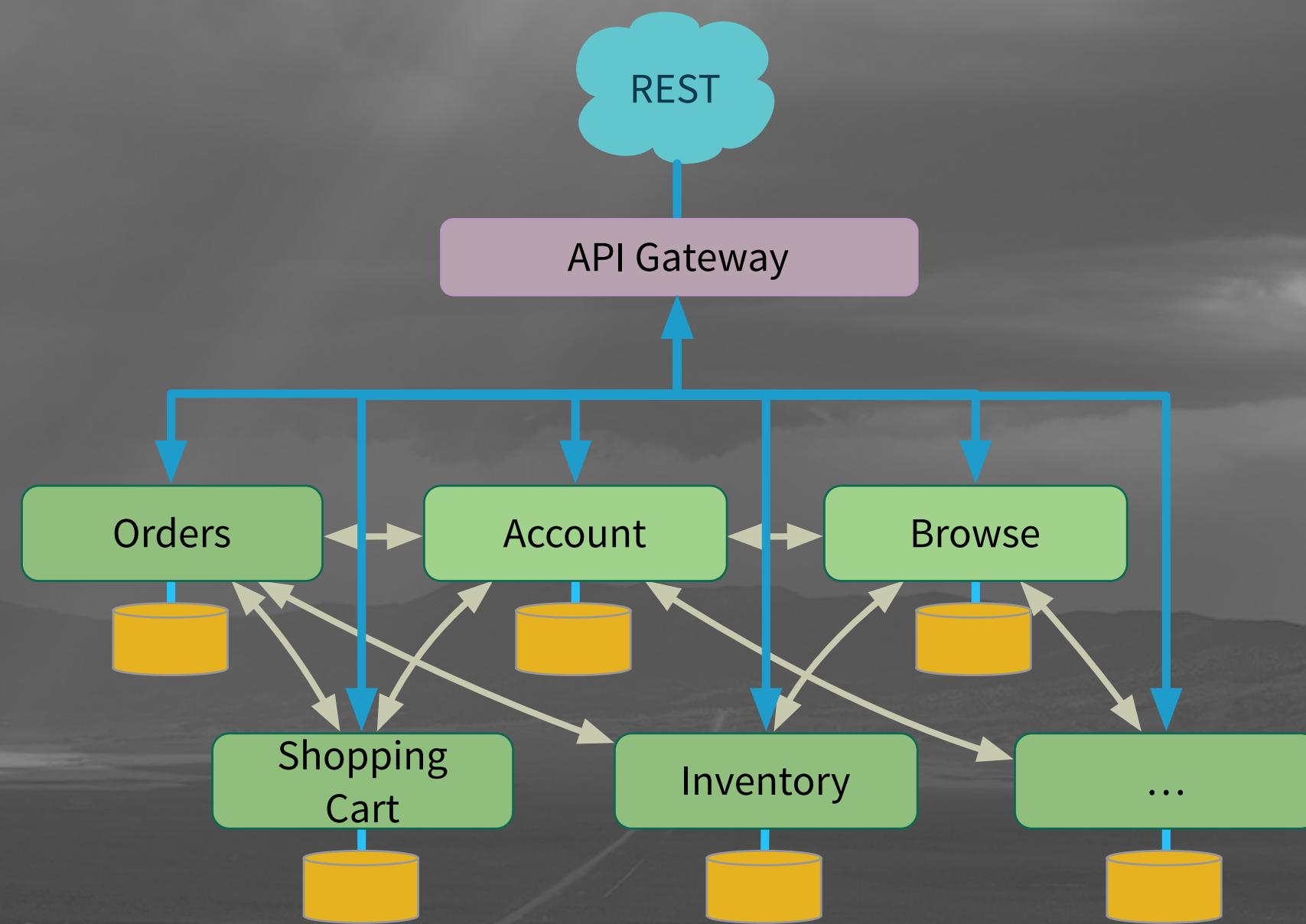




Furthermore, Streaming
Imposes New Requirements

- Reliability - fault and “surprise” tolerant
- Availability - “always on”
- Low latency - for some definition of “low”
- Scalability - up and down
- Adaptability - ideally without restarts

Reminds Me of Microservices





Where to Apply AI First

- “AI in the enterprise will build upon existing analytic applications.”
 - Hybrid systems: enhance existing analytics with ML/AI models.

<https://www.oreilly.com/ideas/9-ai-trends-on-our-radar>

- Should we integrate legacy “expert systems” and how?

Serving Models in Production



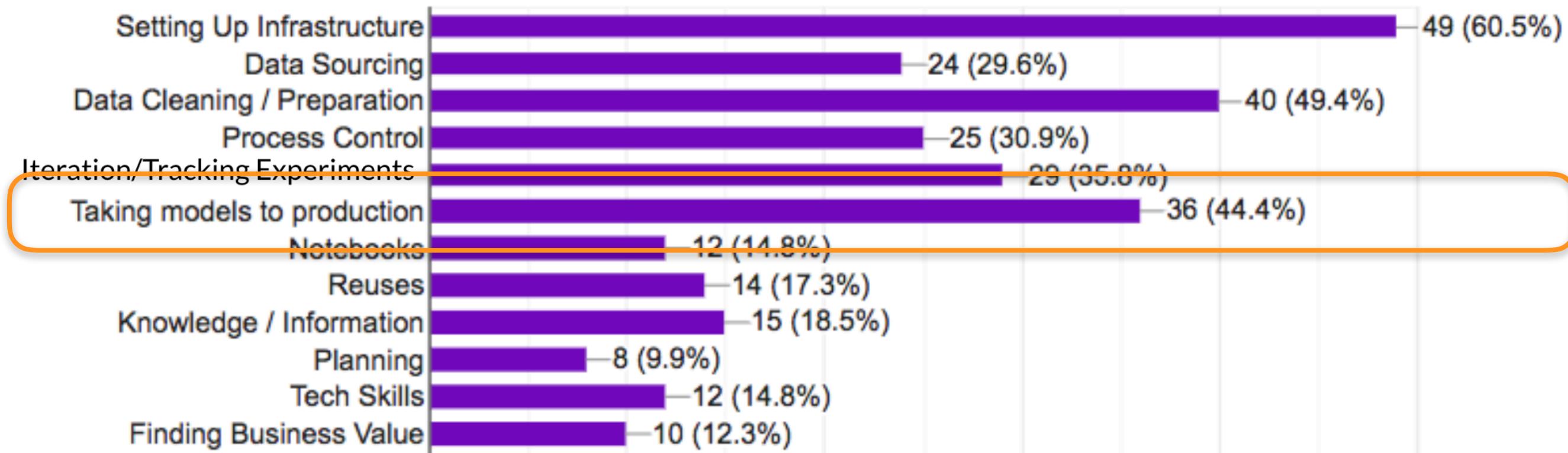
Lightbend

@deanwampler

A Recent Kubeflow User Survey

What are the major pain points in your ML workflows today? Check all that apply.

81 responses



Kuberflow User Survey - 1Q2019

Lack of Tool/Process Integration

- ~60% worry about missed opportunities
- ~50% worry about loss of data team productivity
- ~45% worry about slow time-to-market
- ~40% worry about customer dissatisfaction

CI/CD Processes Required (1/3)

- Version control - for models and code
- Automated builds, tests and other quality checks, artifact delivery
- Launch Configurations: “dark” launches, A/B and other testing scenarios

CI/CD Processes Required (2/3)

- Monitoring
 - Performance overhead
 - Quality metrics match expectations from training?

CI/CD Processes Required (3/3)

- Auditing
 - Which model used to score this record?
 - Which records used to train this model?
 - Who accessed this model and when?

CI/CD Processes Required (3/3)

- All Models Are Data
- All Data Is Modelable
- All Modelable Data Is Modelable

But What's Different? (1/2)

- Automation of model search, experimentation
 - E.g., hyperparameter tuning
- Data safety, fairness, and lineage
- Automation needs to measure reliability, SLAs,
- Reproducibility

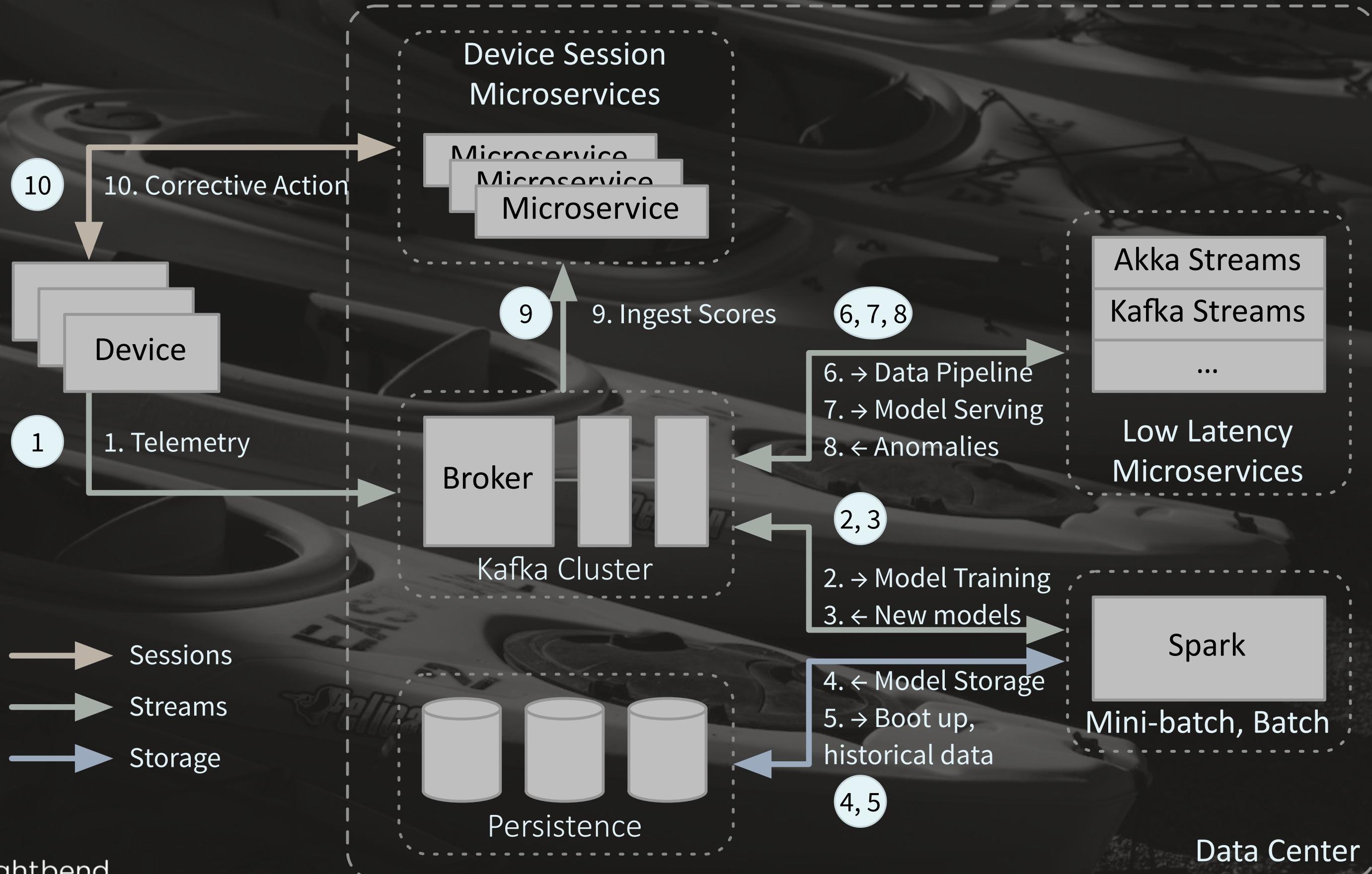
<https://www.oreilly.com/ideas/9-ai-trends-on-our-radar>

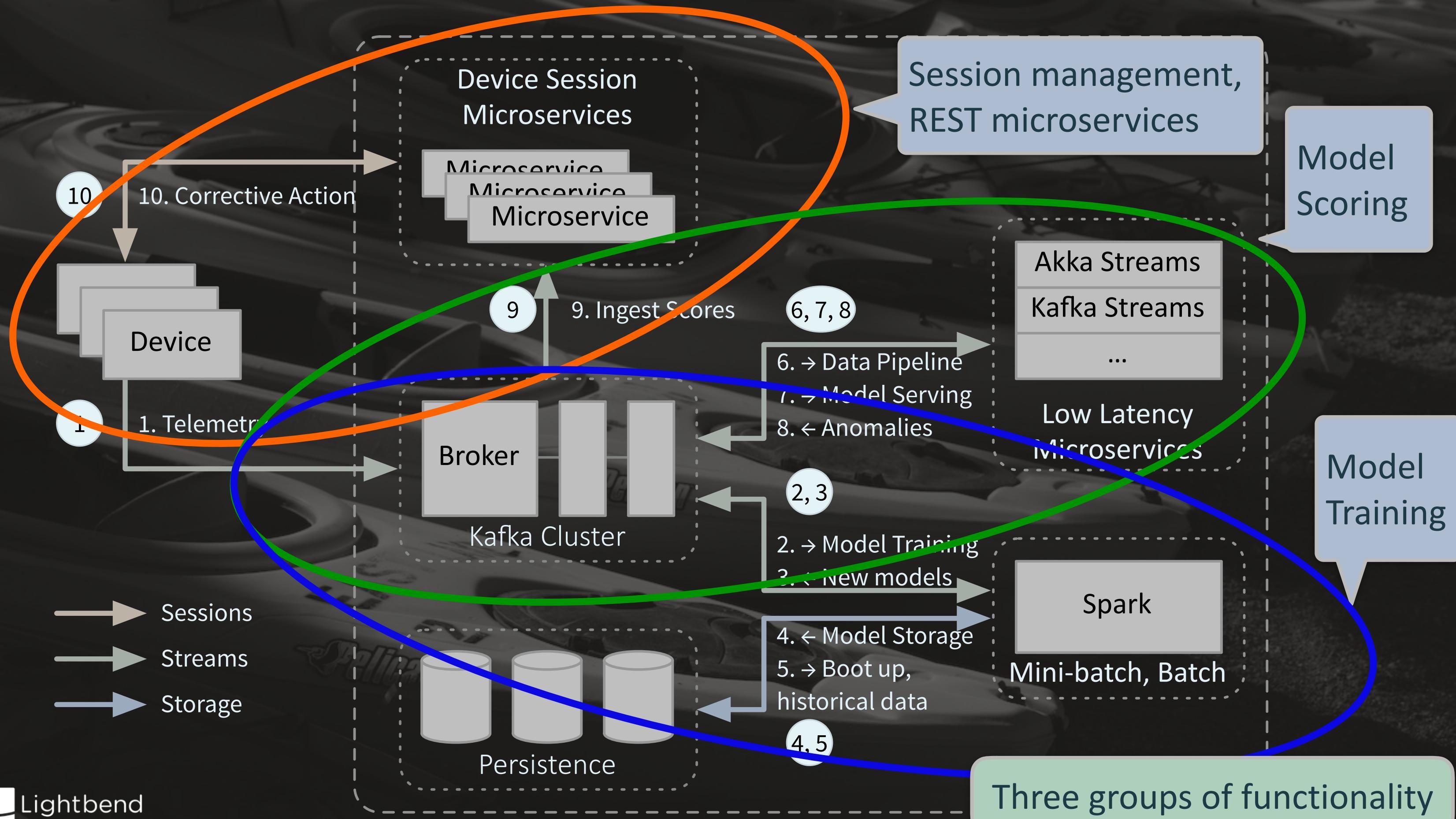
But What's Different? (2/2)

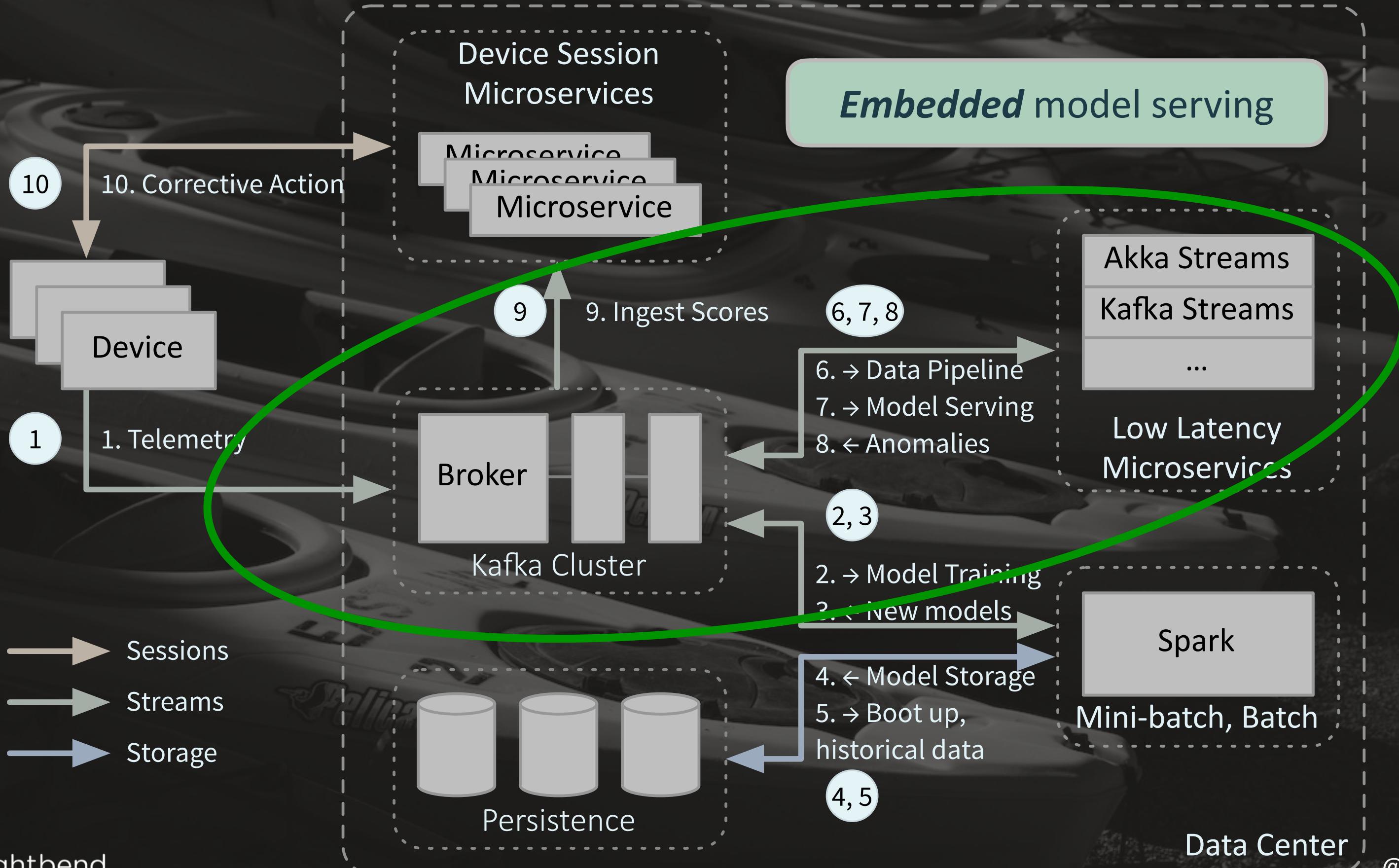
How to integrate the extra indeterminacy in a DevOps world that prefers determinacy?

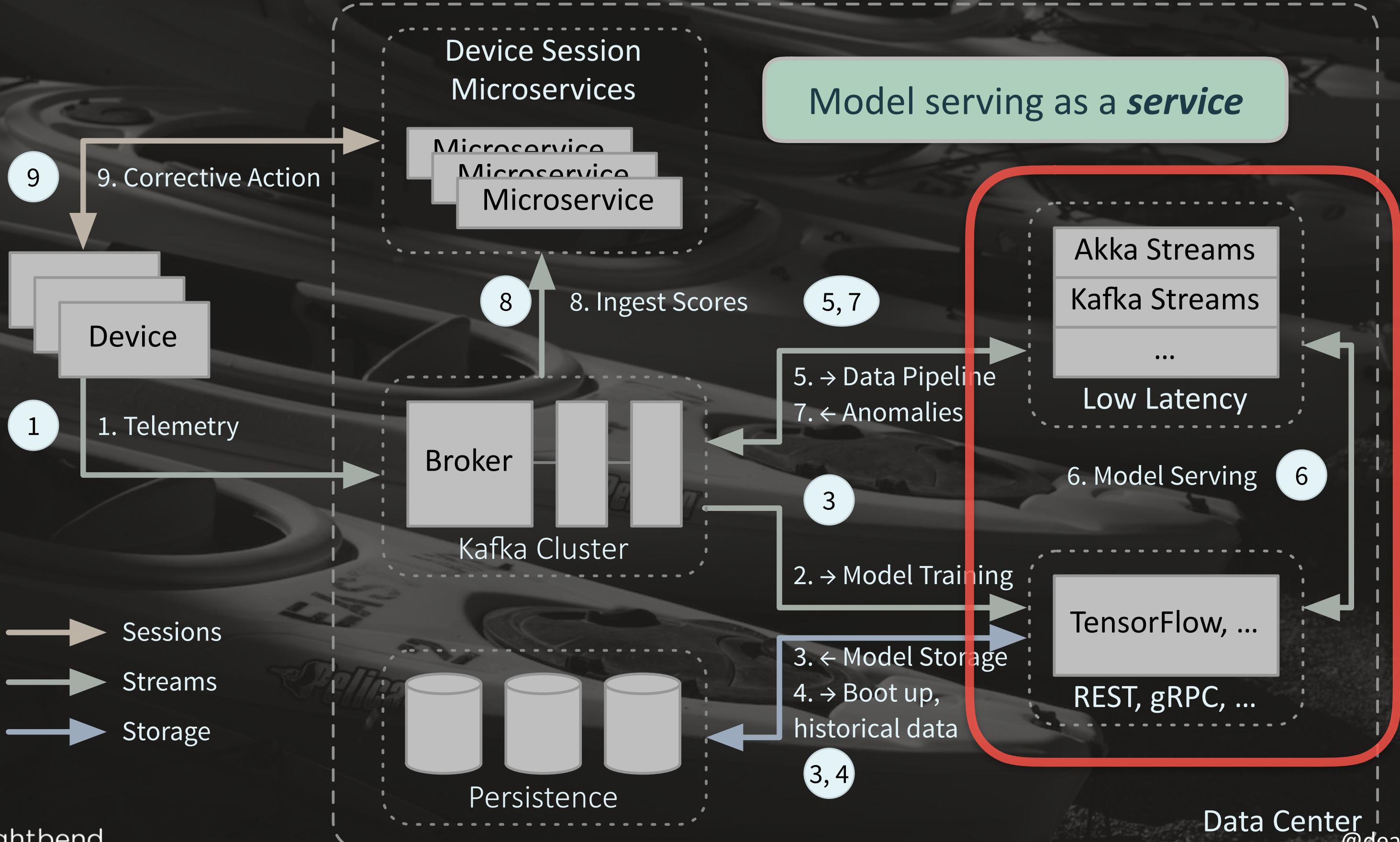
Example Architectures



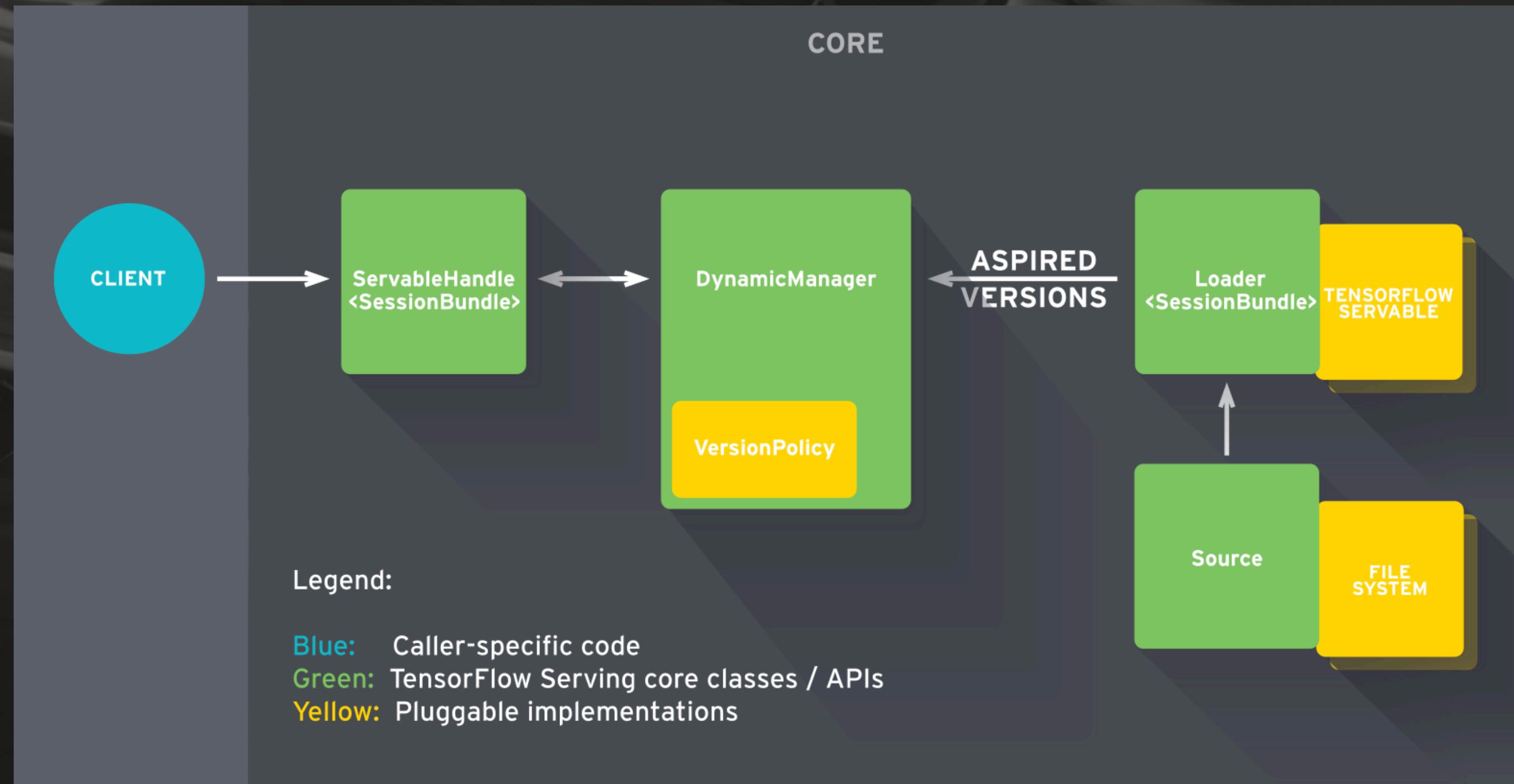






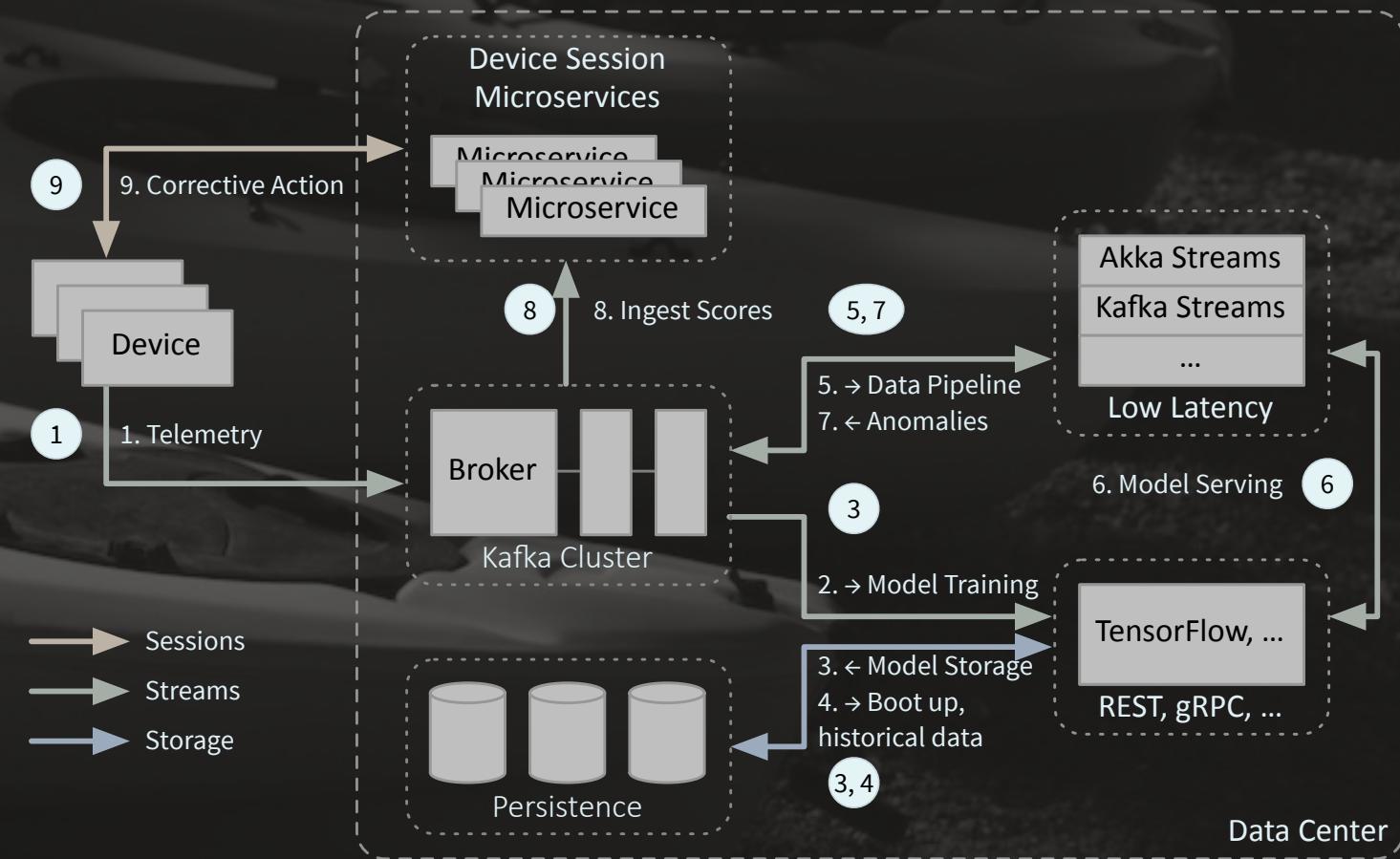


TensorFlow Serving



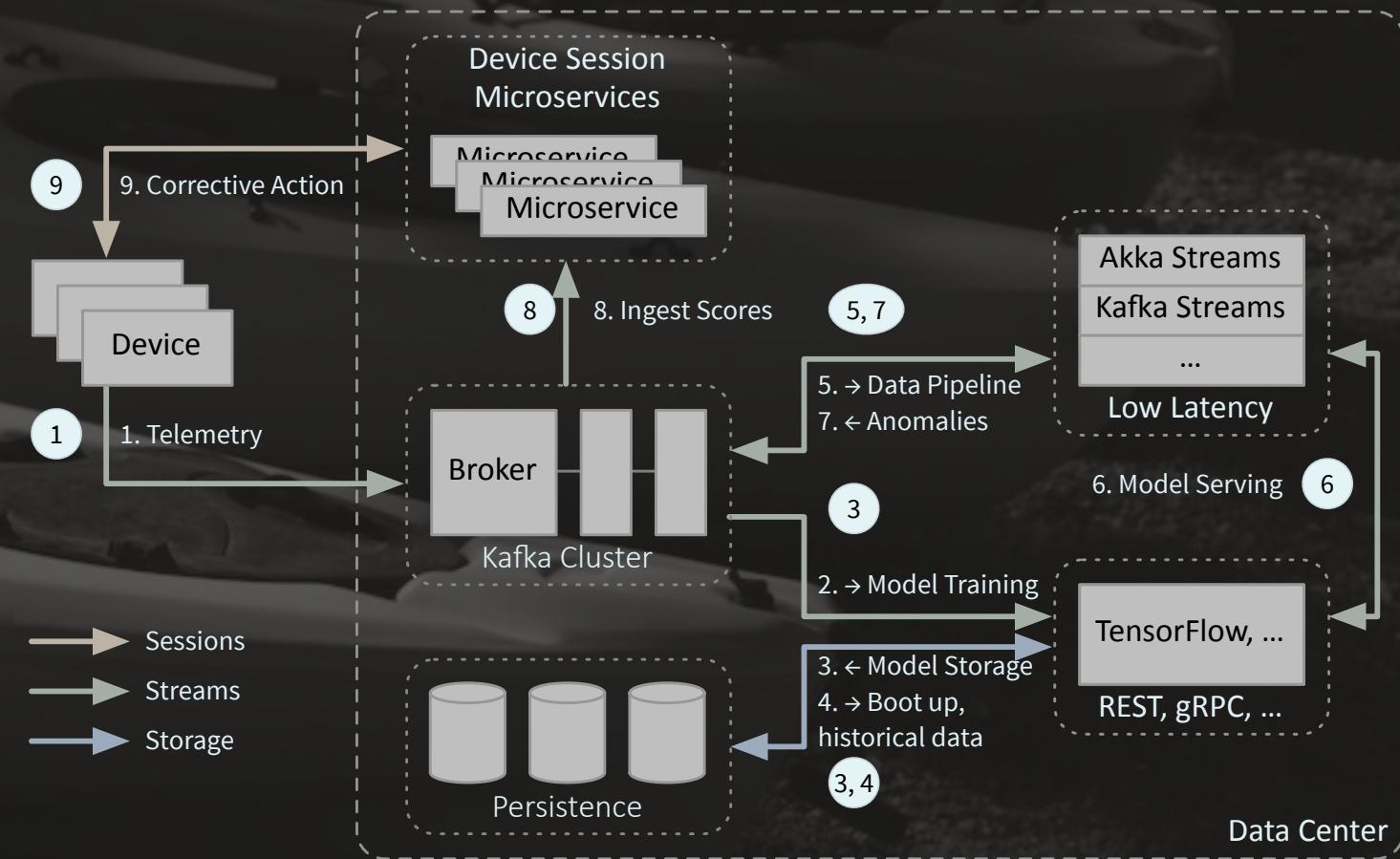
Model Serving as a Service

- Pros:
 - A familiar integration pattern
 - Decouples “concerns”: AI tools, scalability, upgradability, ...



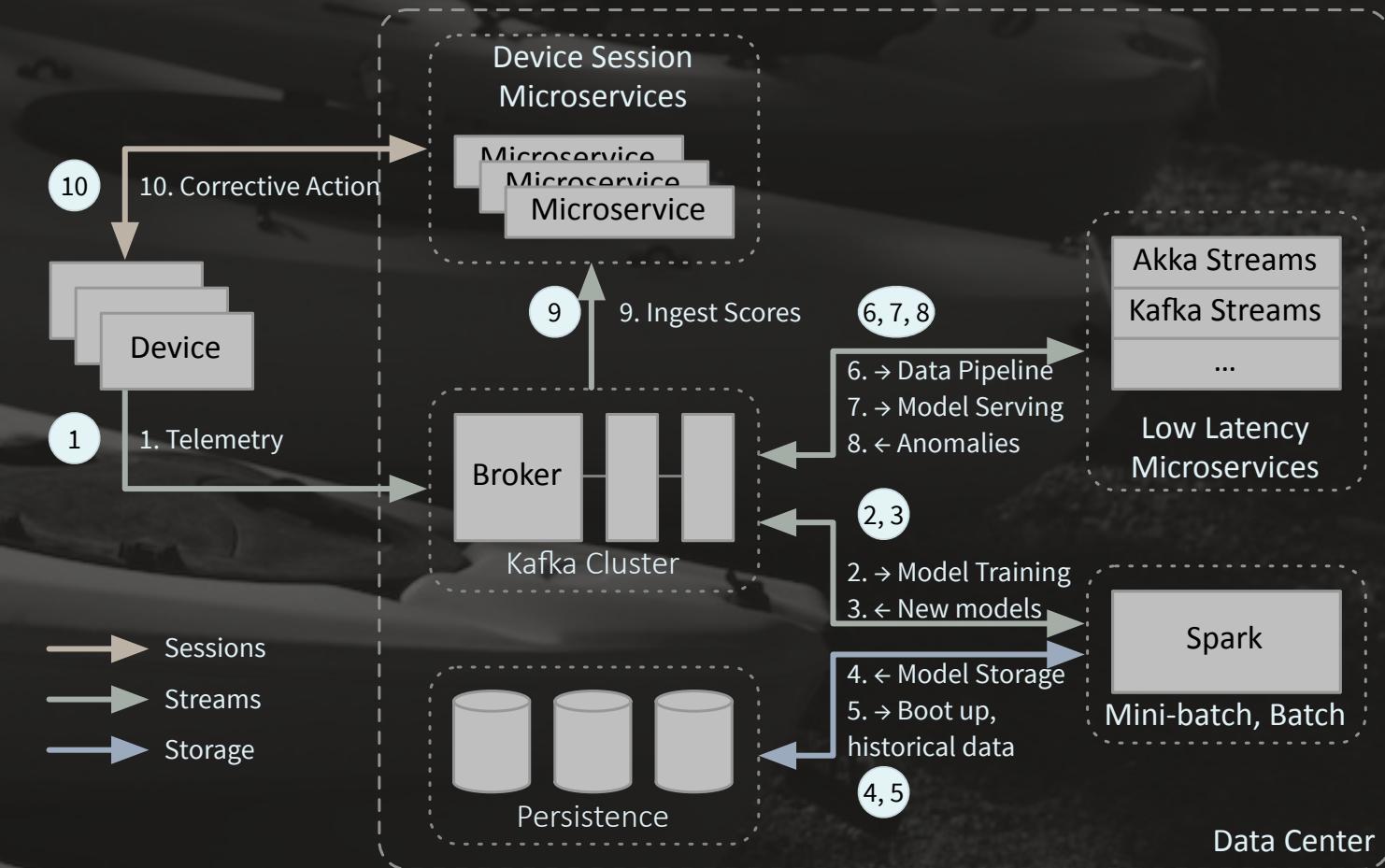
Model Serving as a Service

- Cons:
 - Overhead of invocation, e.g., REST
 - ML Pipeline becomes a unique production workflow



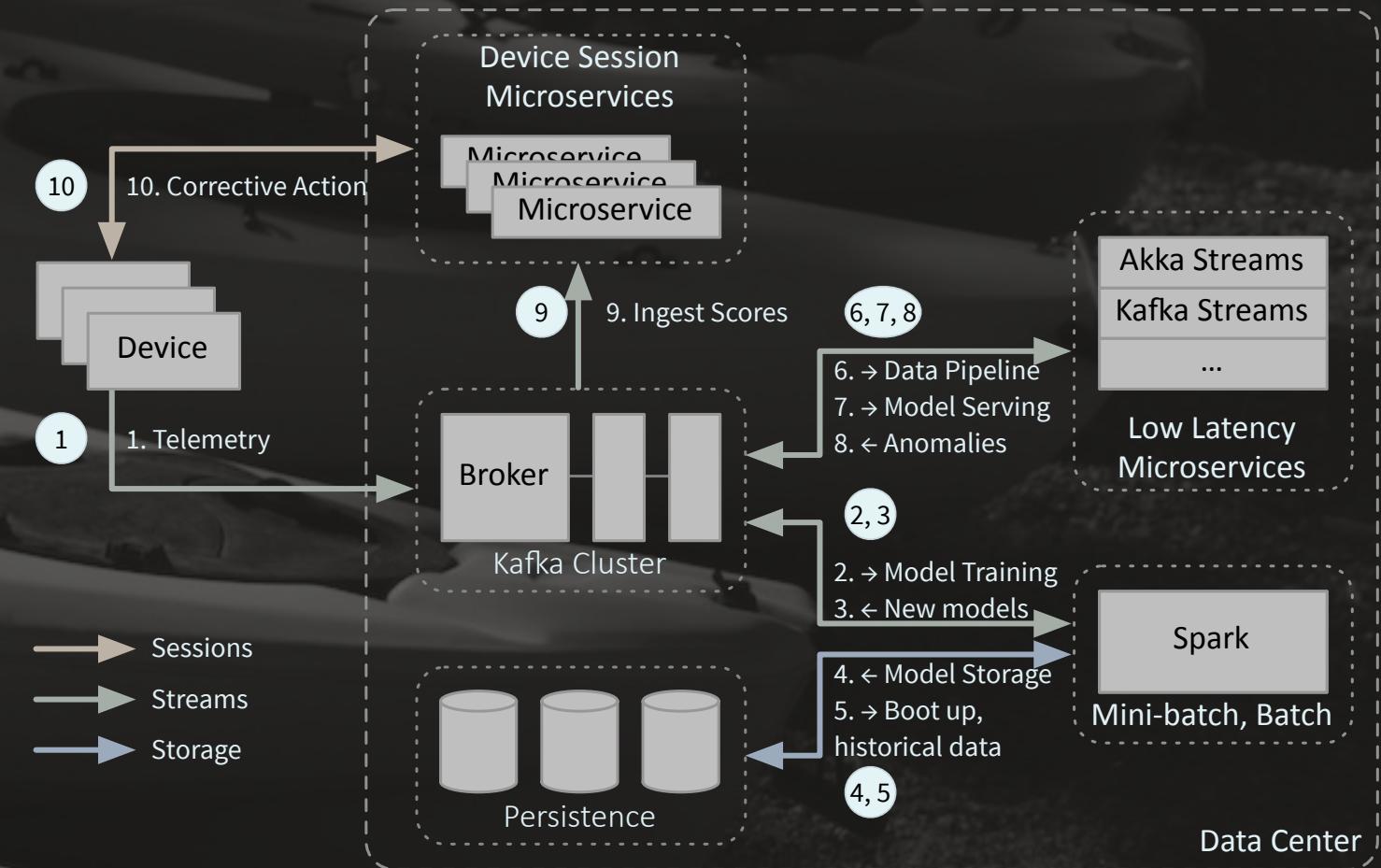
Embedded Model Serving

- Pros:
 - Lowest scoring overhead - interprocess communication only used for model updates
 - Performance tuning focuses on one system, the data pipeline



Embedded Model Serving

- Cons:
 - Model parameters must be serialized
 - Model serving library must be “compatible” with training system:
 - Algorithms, quality



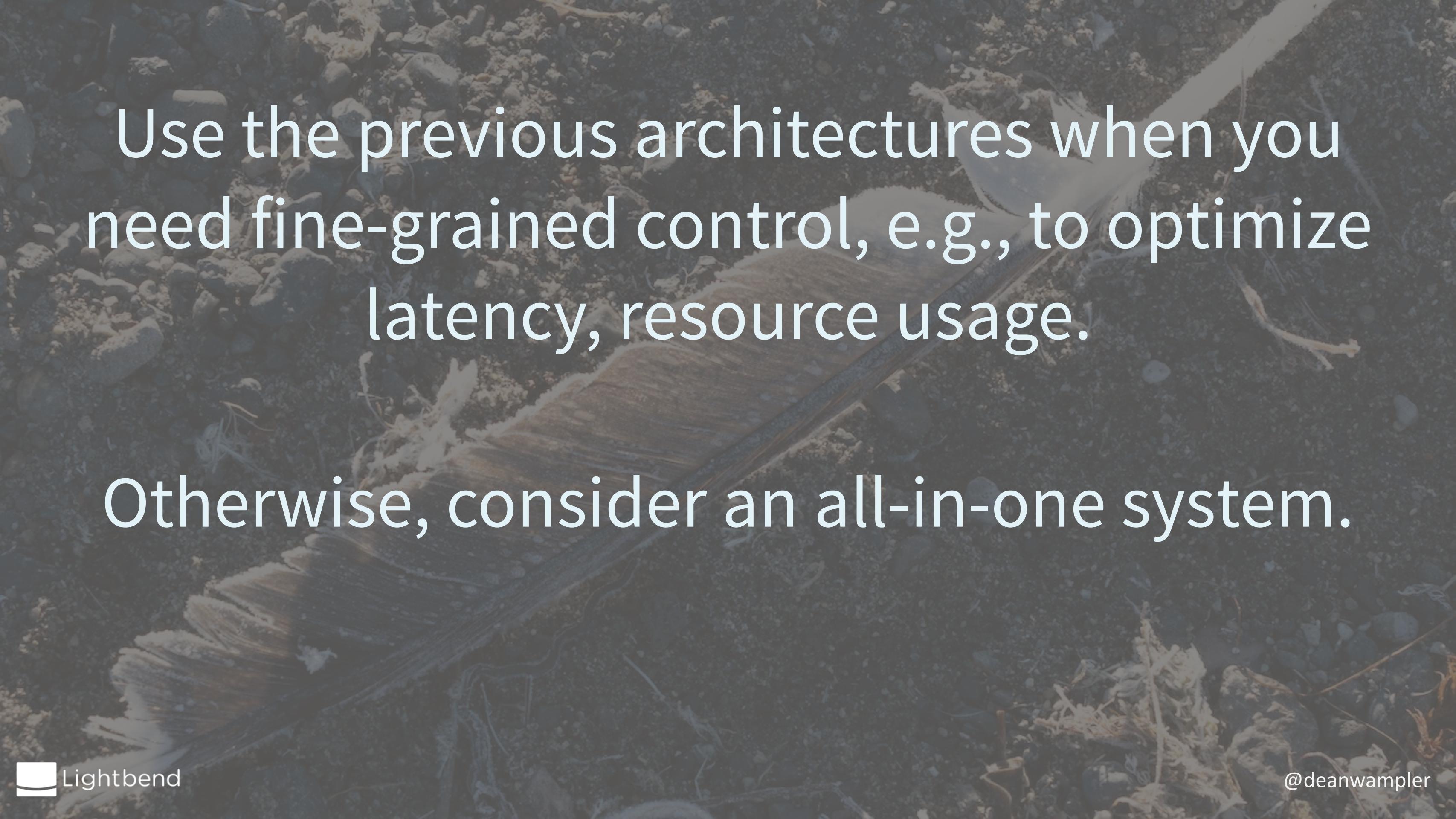


ML/AI Systems



Lightbend

@deanwampler

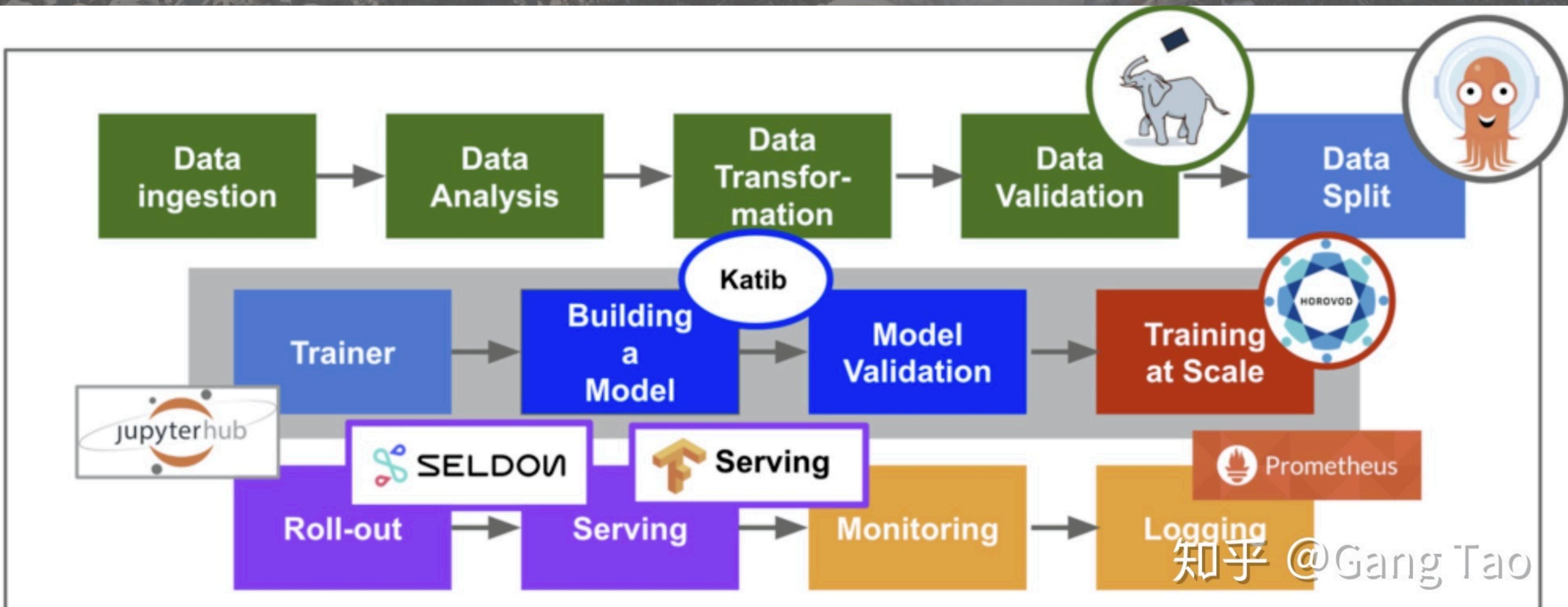


Use the previous architectures when you need fine-grained control, e.g., to optimize latency, resource usage.

Otherwise, consider an all-in-one system.

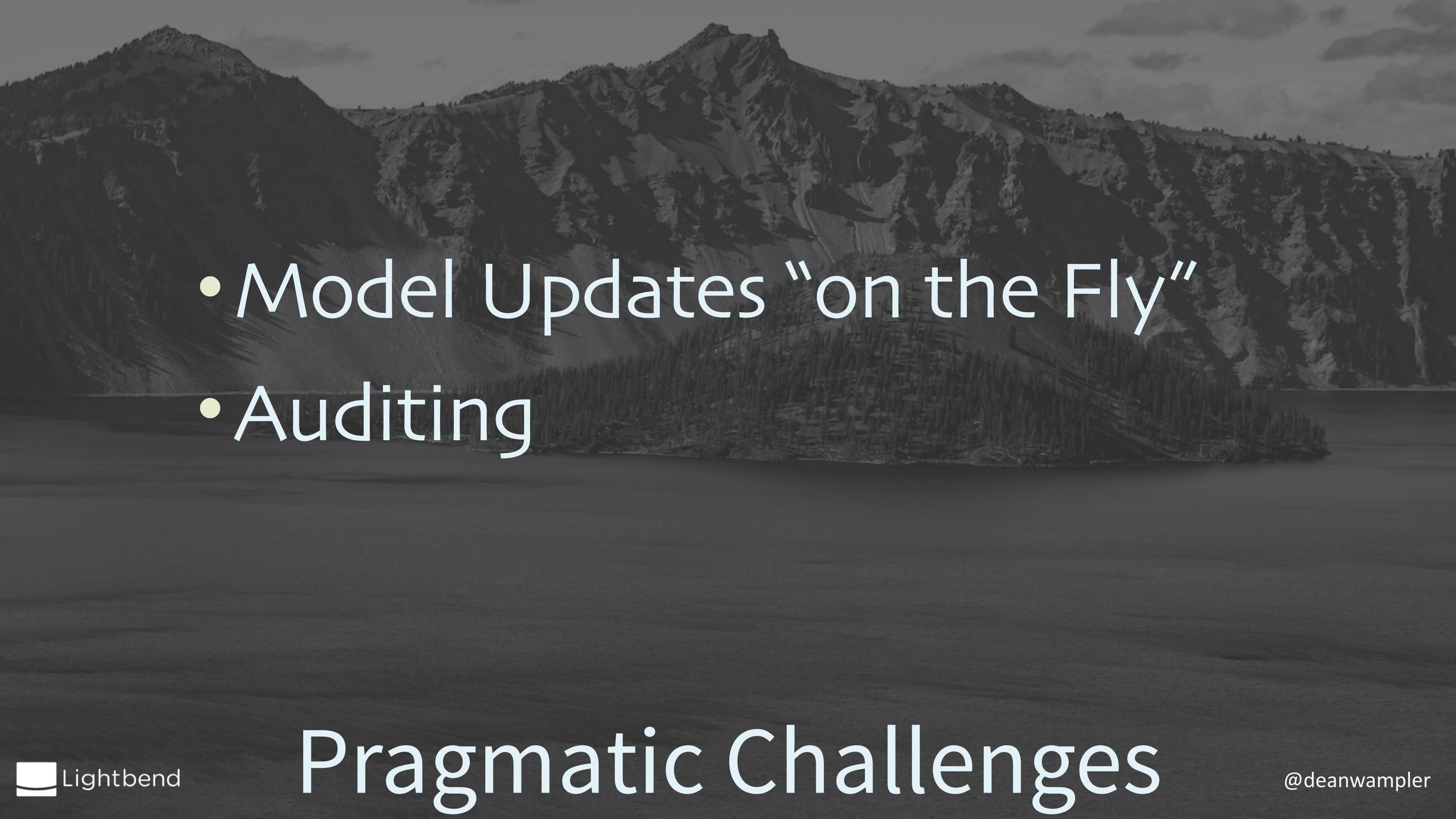
- Kubeflow - for Kubernetes
- SageMaker - for AWS users
- MLFlow - from the Spark community
- ...

Systems - Kubeflow





Pragmatic Challenges

- 
- Model Updates “on the Fly”
 - Auditing

Model Updates “on the Fly”

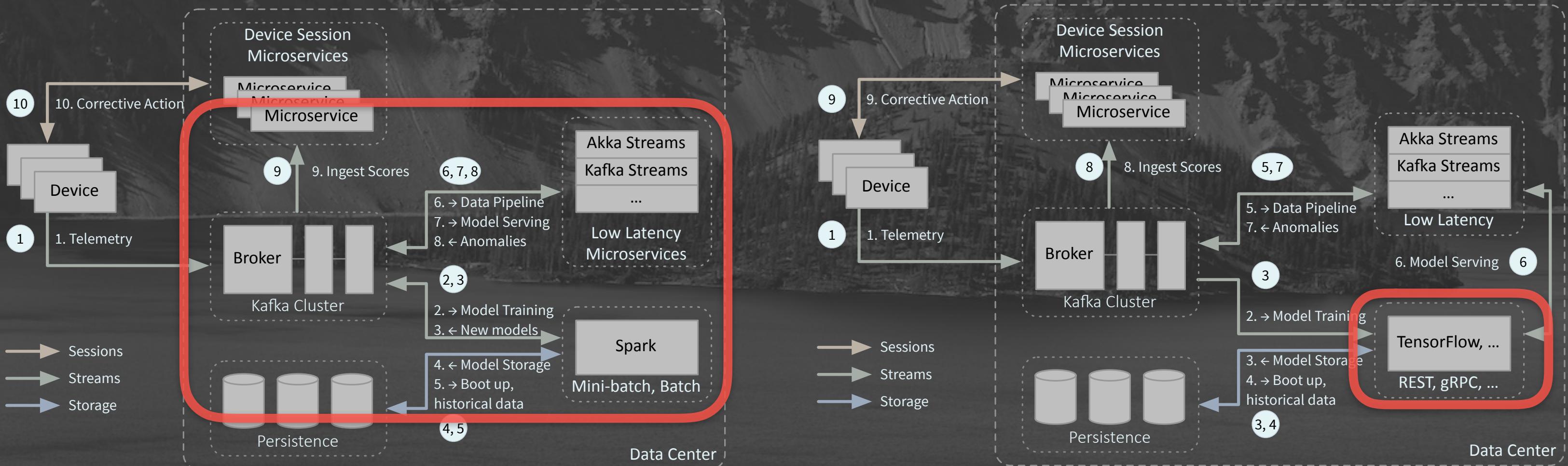
- “Concept Drift” - models grow stale
 - They have a “half life”, too
- Periodically retrain then serve the new model without downtime
 - (Continuous training is a research topic)

Retraining Considerations

- Trade off model performance vs. retraining cost
- How far back in the data set do you go?

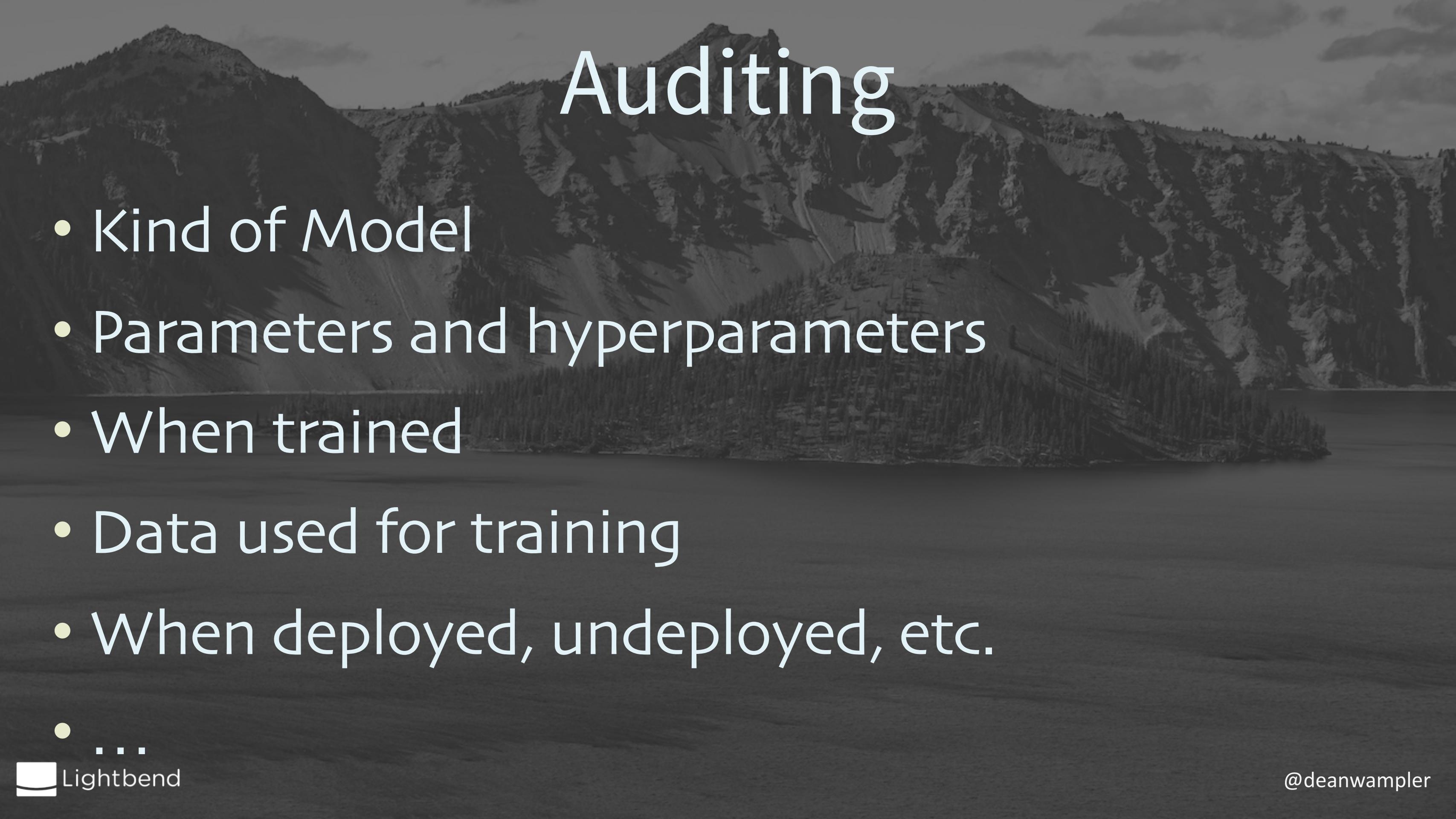
Retraining Considerations

- Approaches:
 - Run “periodic” batch jobs - most mature
 - How far back in the data set do you go?
 - Online and incremental algorithms
 - Auto-adaptive ML (i.e., continual learning) - future capability



Complex to update
embedded models!

Model updates can be
straightforward



Auditing

- Kind of Model
- Parameters and hyperparameters
- When trained
- Data used for training
- When deployed, undeployed, etc.
- ...

Auditing

- Quality metrics
- Serving metrics (how many records, scoring times...)
- Provenance of decision to retrain
 - The metrics gathered above that were used to decide when to retrain



Dusty Milky Way

Mars last Summer



Lightroom

@deanwampler

References

- Ideas:
 - [Ben Lorica on 9 AI Trends](#)
 - [Paco Nathan's Data Governance Talk](#)

You can get these slides with the links here:
polyglotprogramming.com/talks

References

- Ideas:
 - O'Reilly Radar: [Data, AI, others](#)
 - [distill.pub](#)
 - [The Algorithm](#)
 - [The Gradient](#)

References

- Work on the unique challenges of integrating ML/AI in SW development processes
- SEMLA workshops: [2018](#), [2019](#)

References

- A few research papers, etc.
- Incremental training
- an example
- Continual learning
- Explainability

References

- Kubeflow
- MLFlow
- DVC
- AWS SageMaker
- Fiddler (explainable AI)

References

- General Information about Stream Processing
- [My O'Reilly Report on Architectures](#)
- [Streaming Systems Book](#)
- [Stream Processing with Apache Spark](#)
- [Designing Data-Intensive APPS book](#)

References

- Other Talks
 - [Strata Talk on ML in a Streaming Context](#)
 - [Stream All the Things! \(video\)](#)
 - [Streaming Microservices with Akka Streams and Kafka Streams \(video\)](#)

References

- Tutorials
 - [Model serving in streams](#)
 - [Stream processing with Kafka and microservices](#)

Controls

GRAFANA WORKLOADS

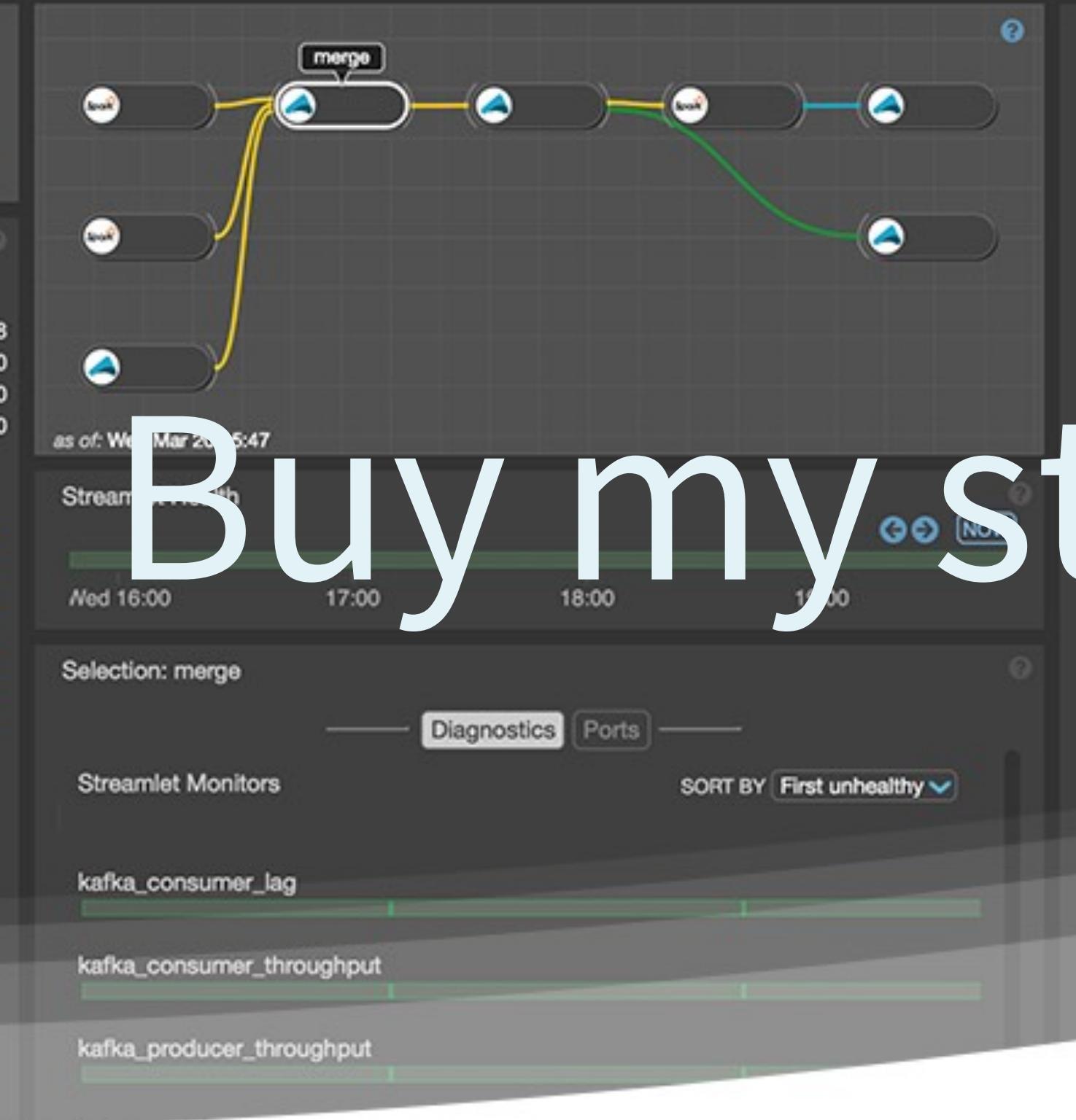
Application Details

Streamlet Current Health

Healthy	8
Warning	0
Critical	0
Unknown	0

Streamlet Health Events

- cdr-validator
- cdr-aggregator
- merge
- console-egress
- error-egress
- cdr-generator1
- cdr-generator2
- cdr-ingress



Buy my.stuff!!

What we're up to at Lightbend...
lightbend.com/lightbend-pipelines-demo



Questions?

Dean Wampler, Ph.D.
dean@lightbend.com
[@deanwampler](https://twitter.com/deanwampler)
lightbend.com/lightbend-platform
polyglotprogramming.com/talks

