

What it takes to use machine learning in fast data pipelines

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



Lightbend

© Dean Wampler & Lightbend, 2007-2019

@deanwampler

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

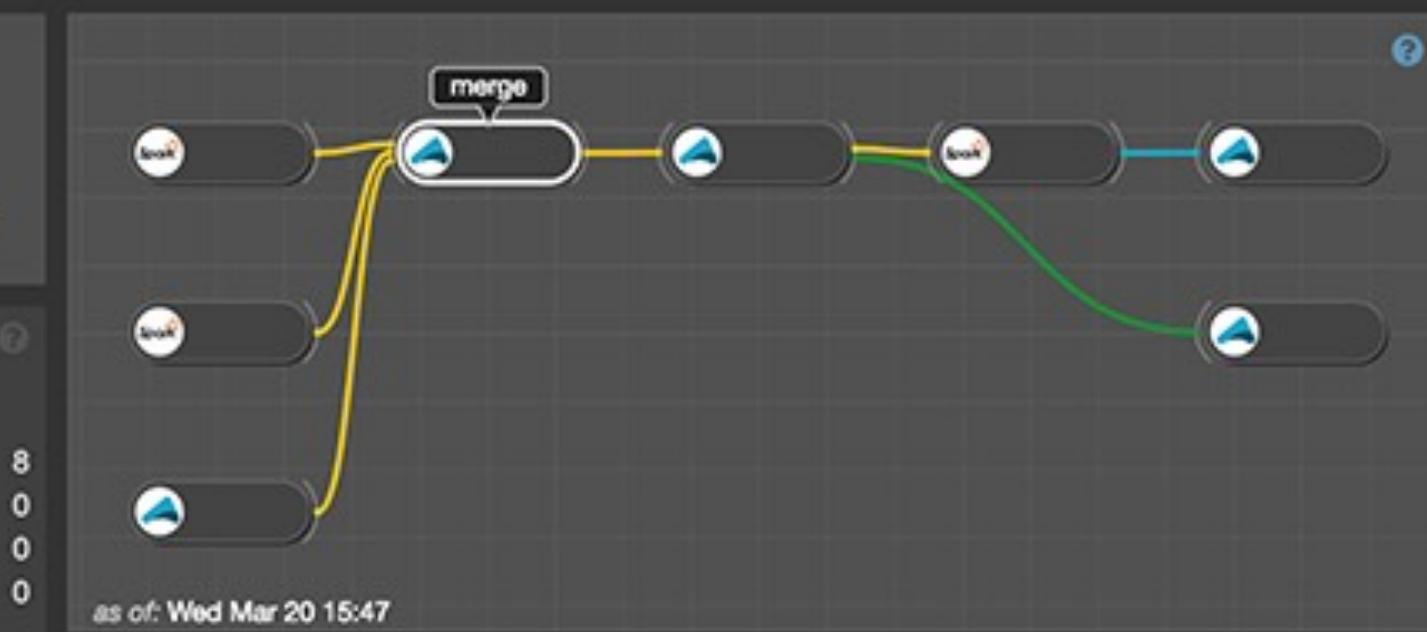
HDFS

SQL, NoSQL

Cloud Storage (S3 etc)

Search





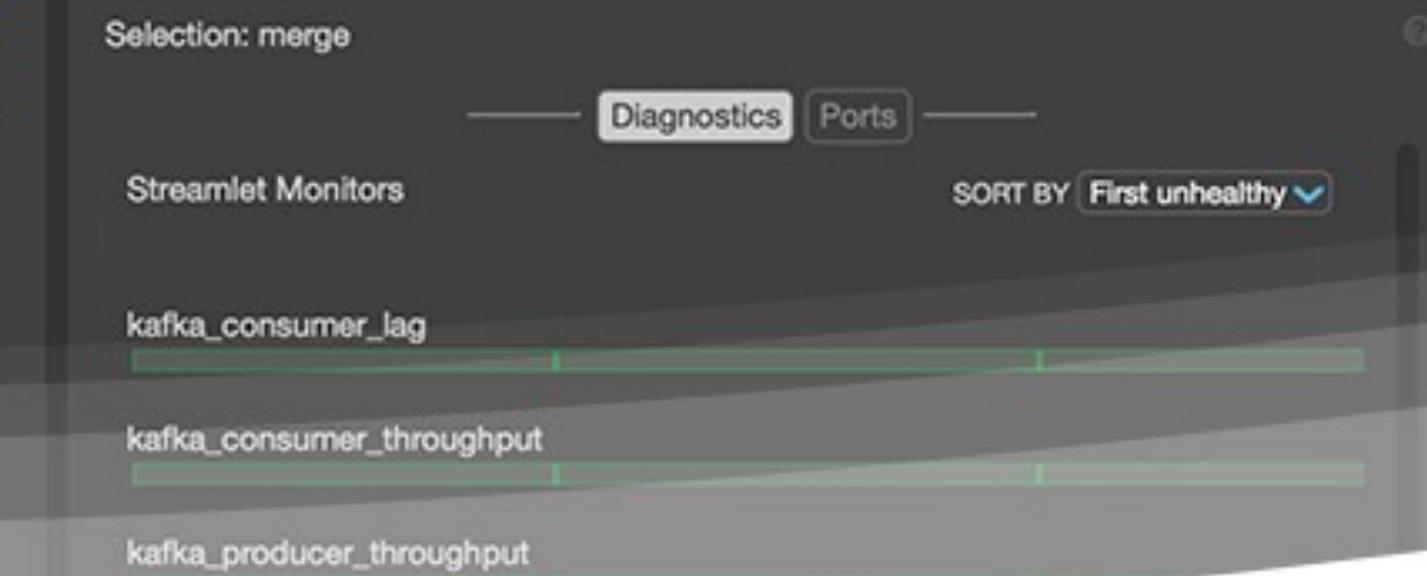
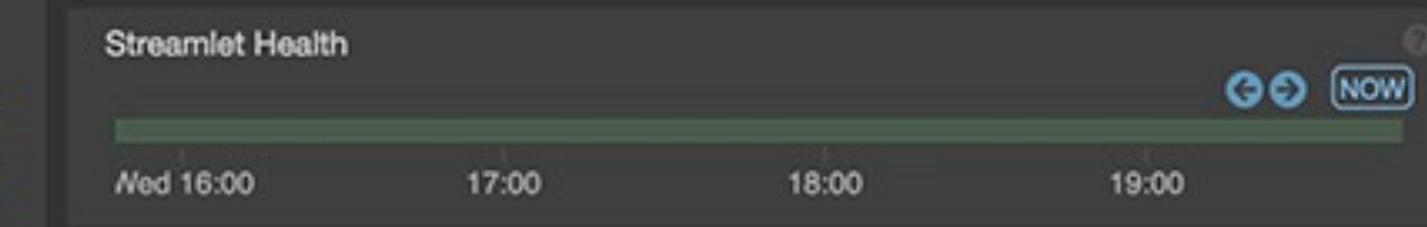
Application Details

Streamlet Current Health

Healthy	8
Warning	0
Critical	0
Unknown	0

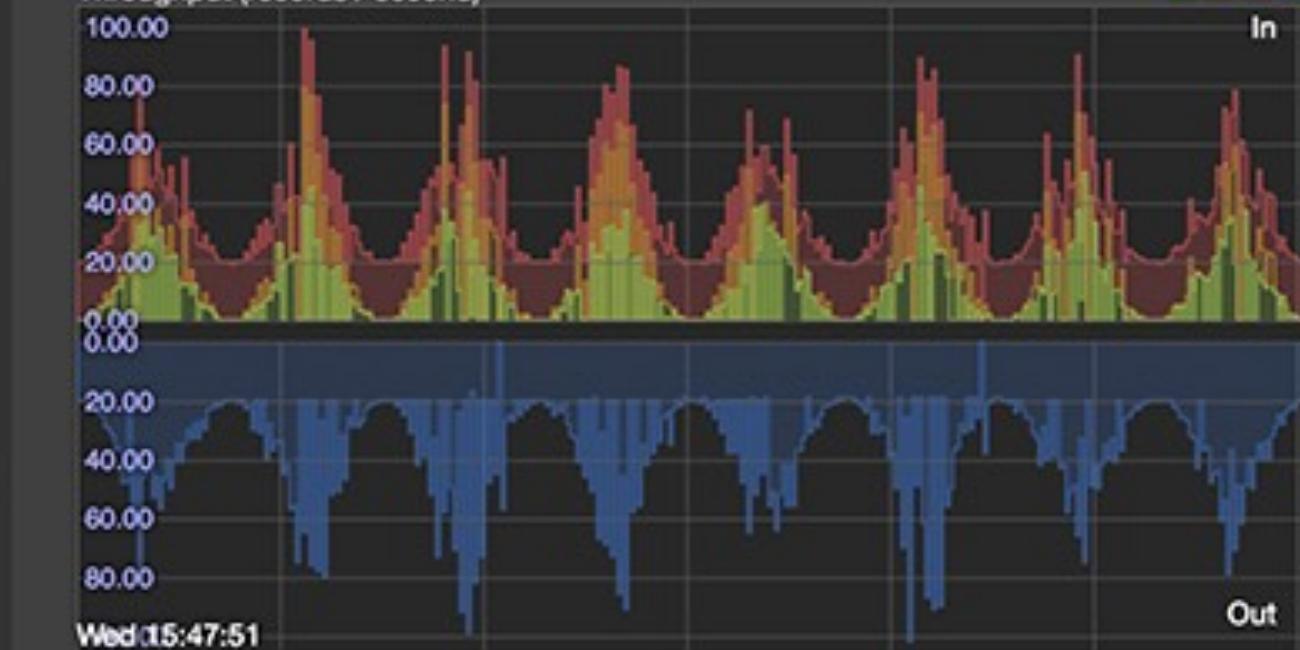
Streamlet Health Events

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

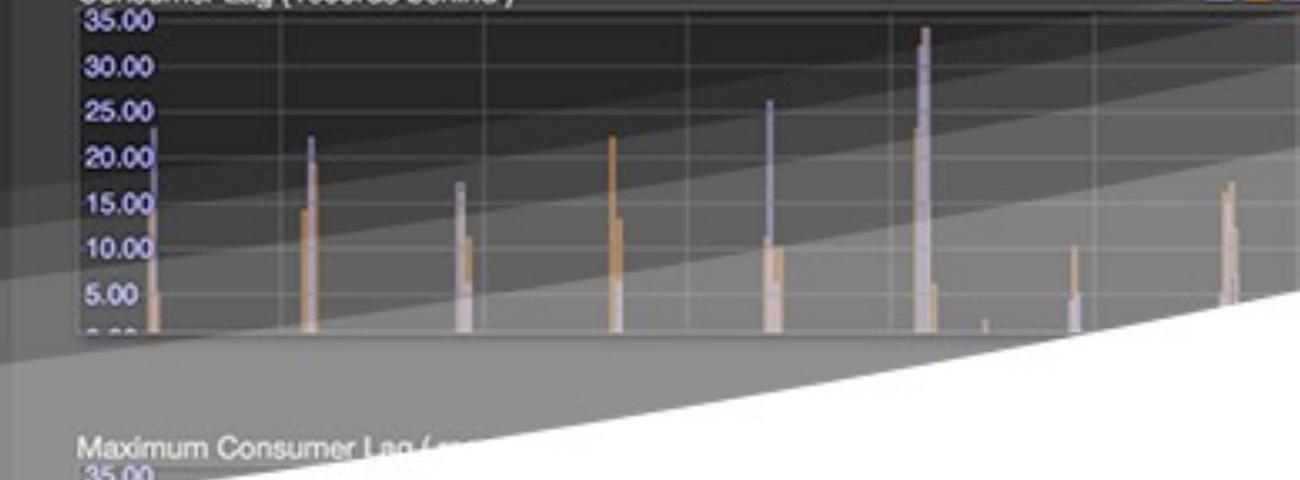


Metrics

Throughput (records / second)



Consumer Lag (records behind)



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



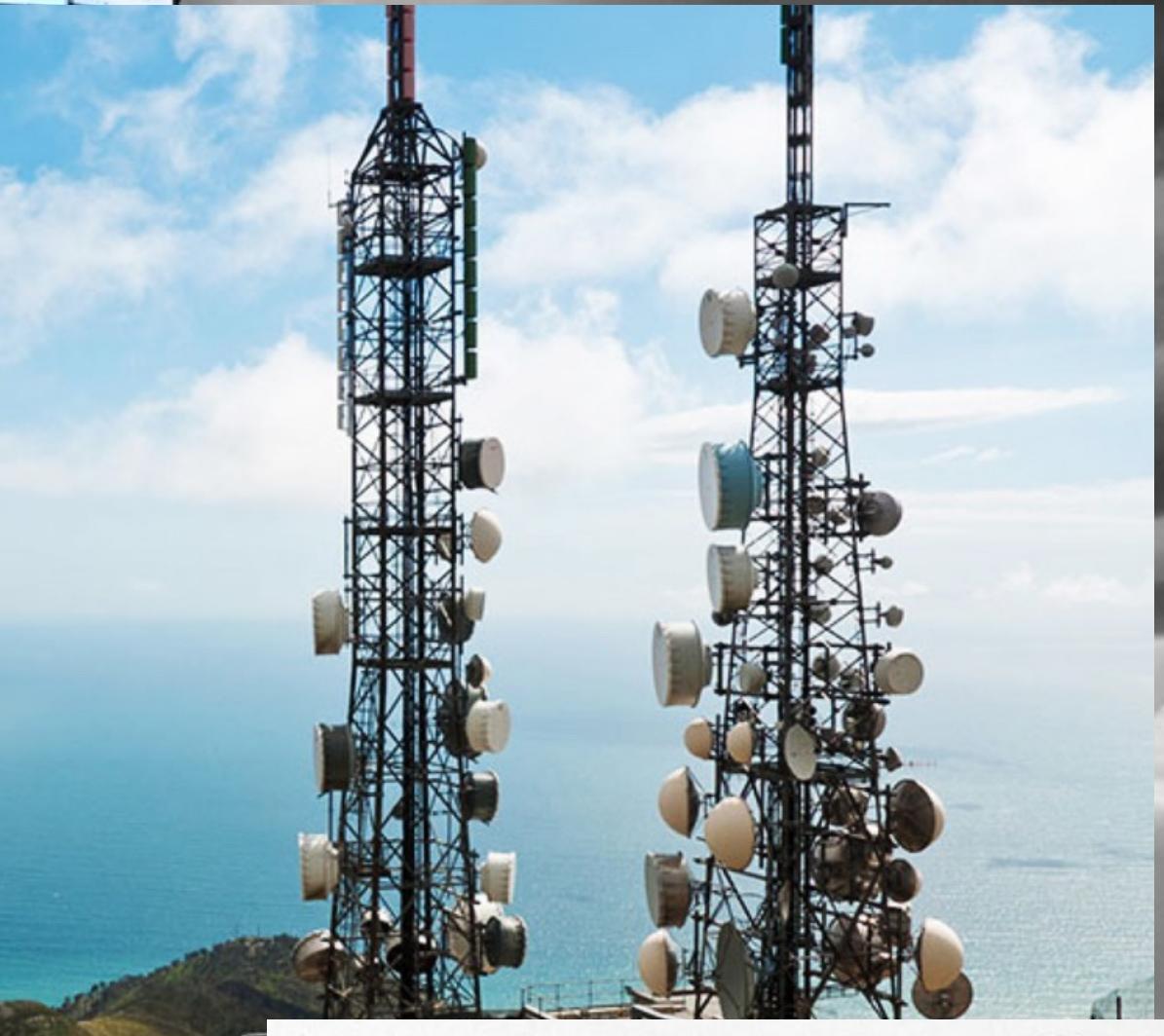
What We'll Discuss

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



Batch vs. streaming... and why

Telecom



Finance



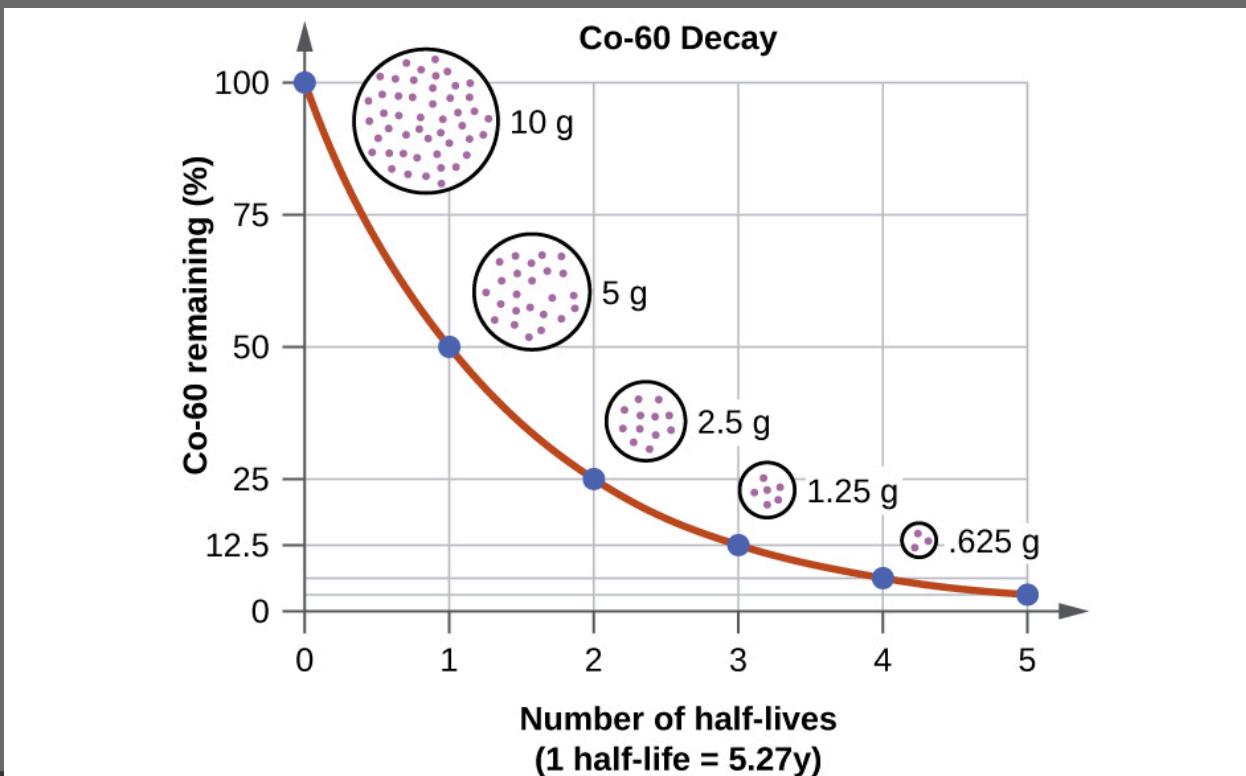
Energy

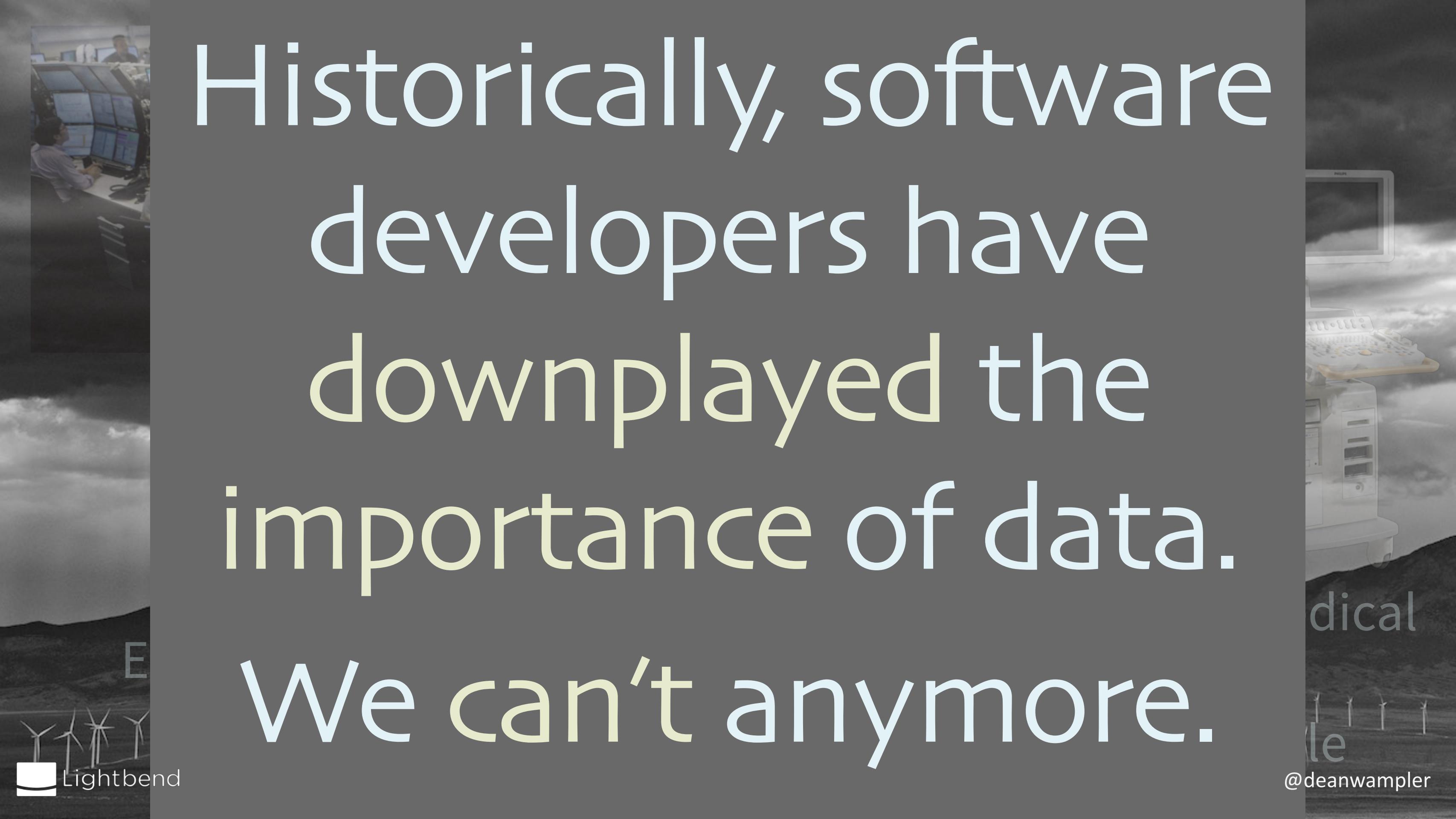
... and IoT

State of the art phone!



Information value has a half life; it decays with time





Historically, software
developers have
downplayed the
importance of data.
We can't anymore.



Lightbend

@deanwampler



Data Science vs. Data Engineering



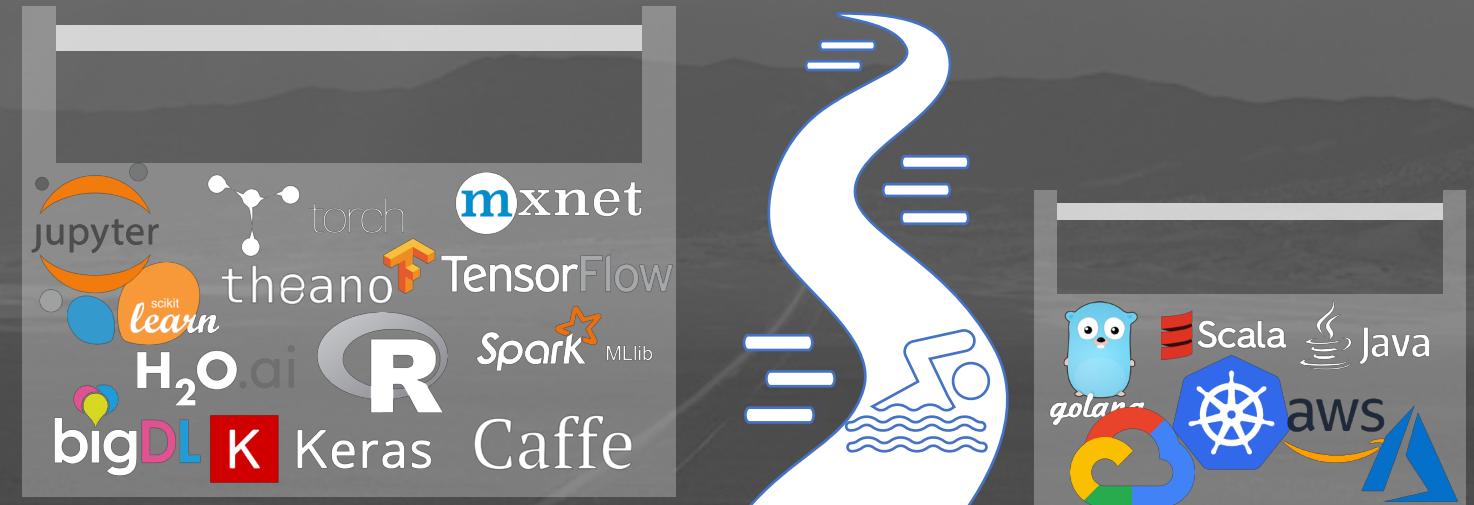


Data Science toolbox



Software
Engineering toolbox

- Comfortable with uncertainty
- Less process oriented
 - Iterative, experimental
- Uncomfortable with uncertainty
- Process oriented
 - Agile Manifesto
 - ... which does not mention data!



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



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

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
- 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?

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



Tools



Lightbend

@deanwampler

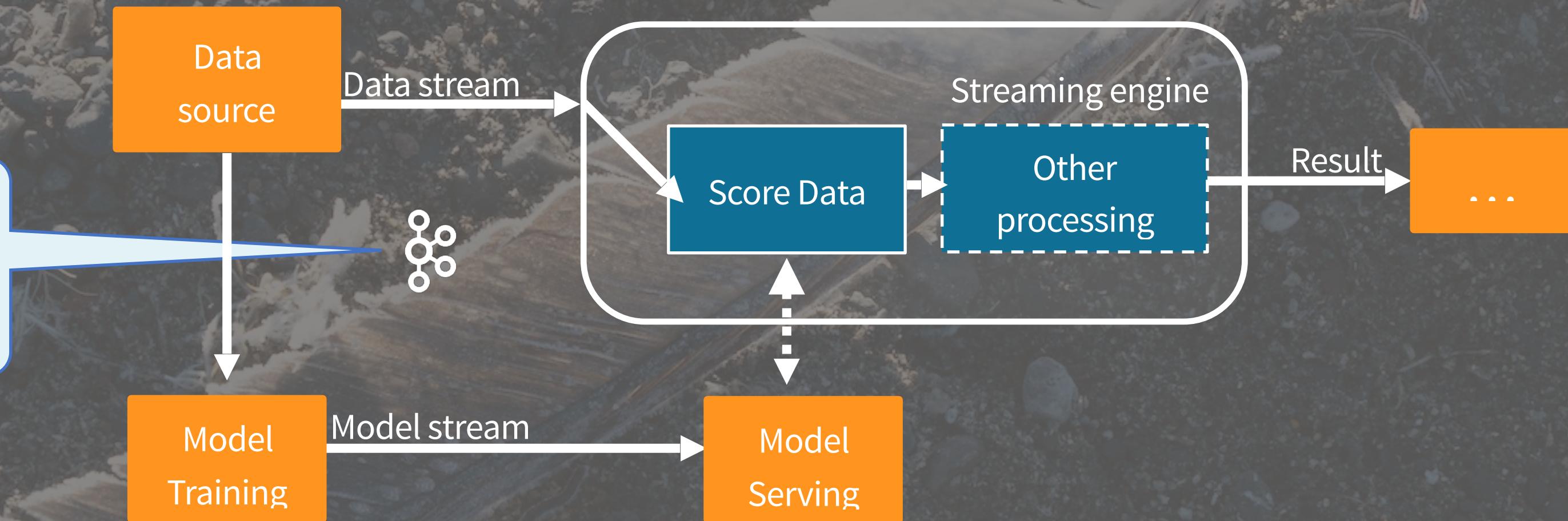
Tools

- Approaches:
 - Model Serving as a Service
 - Embedded within Microservices
- Systems
 - Kubeflow, MLFlow, ...



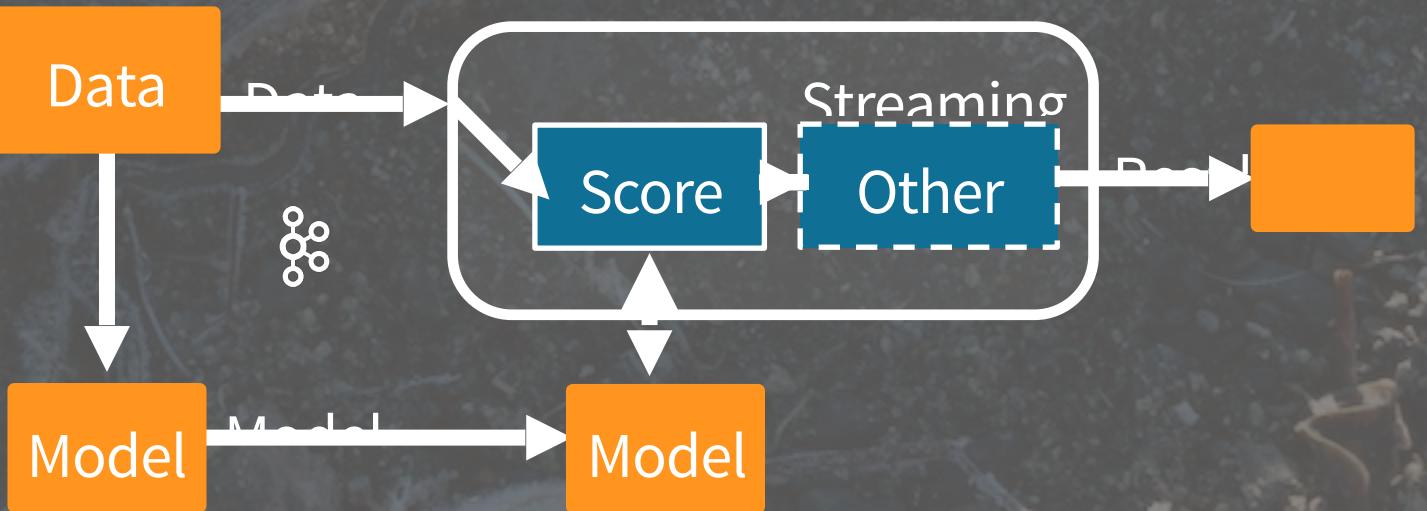
Model Serving as a Service

We'll use Kafka as the “log” system.



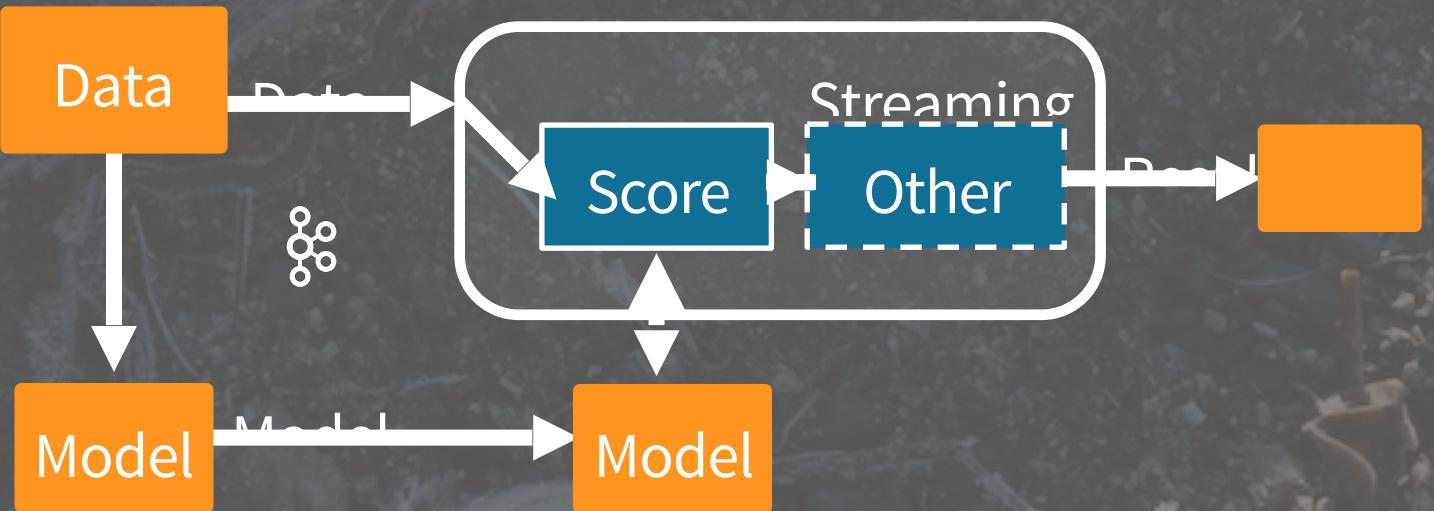
Serving

- Pros:
 - Decouples “concerns”: tools, scalability, upgradability, ...
 - One system for training and scoring

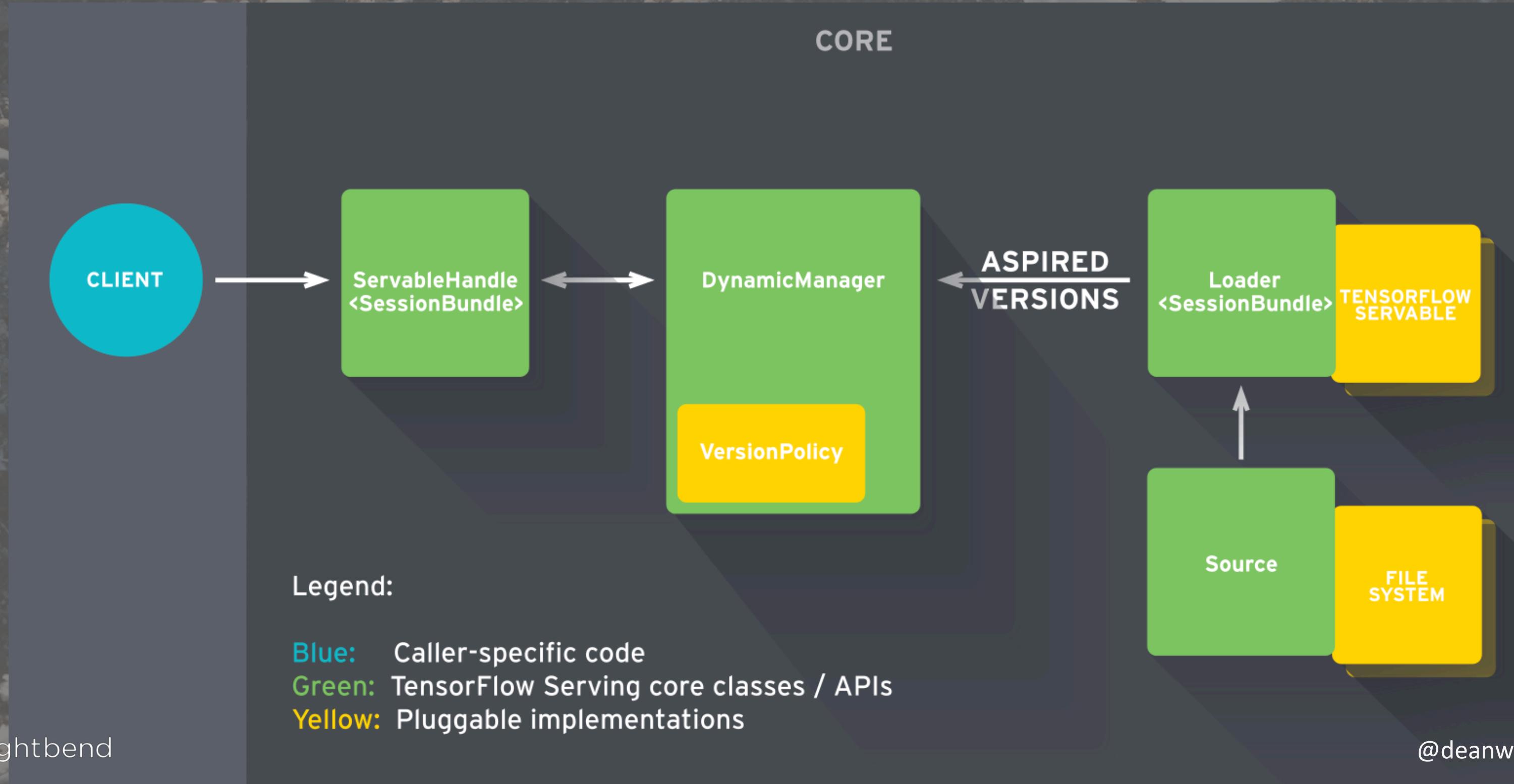


Serving

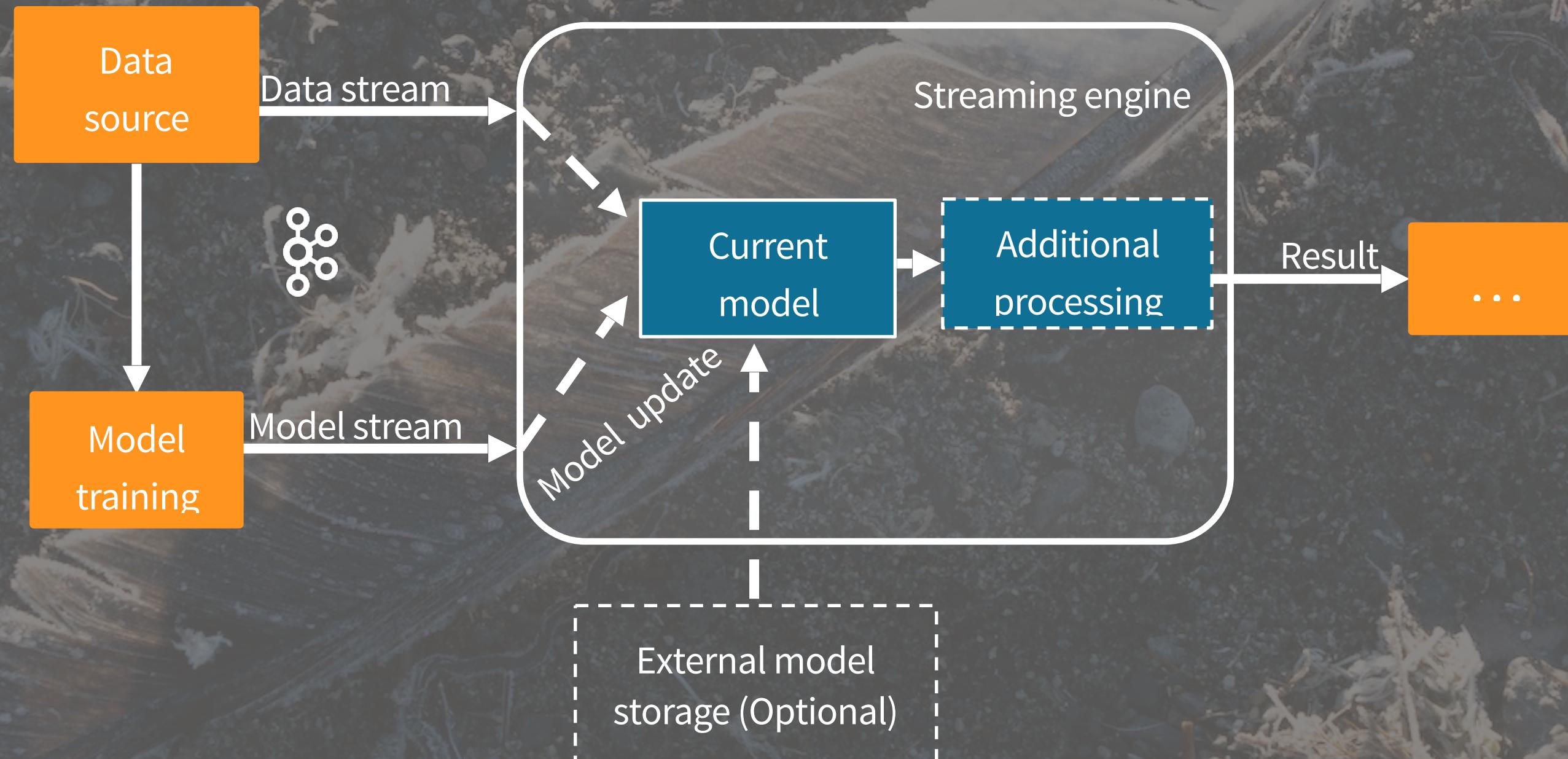
- Cons
 - Overhead of invocation, e.g., REST
 - ML Pipeline becomes a unique production work flow



TensorFlow Serving

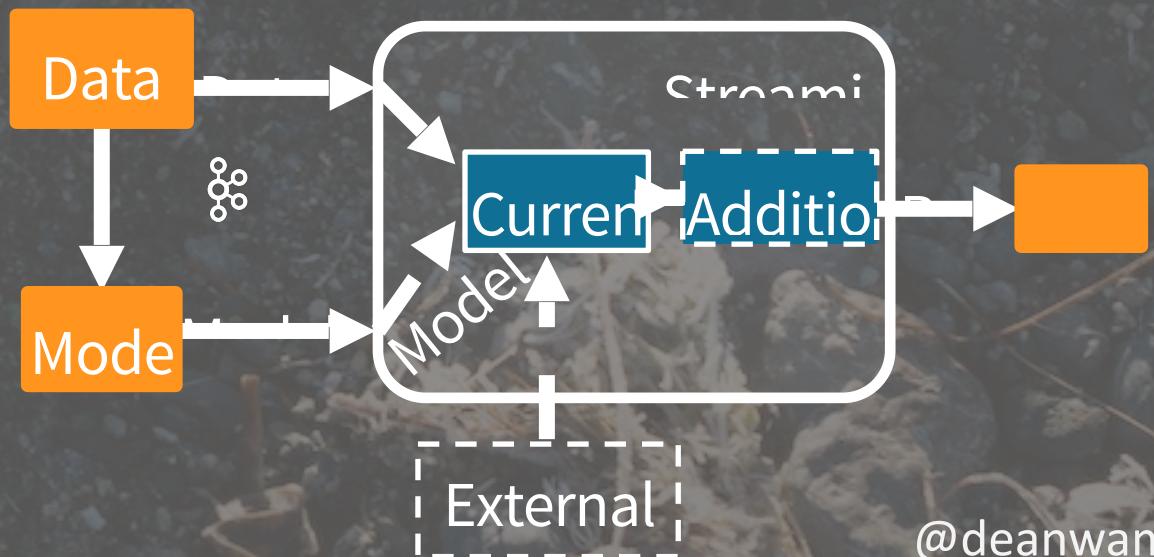


Embedded in Microservices



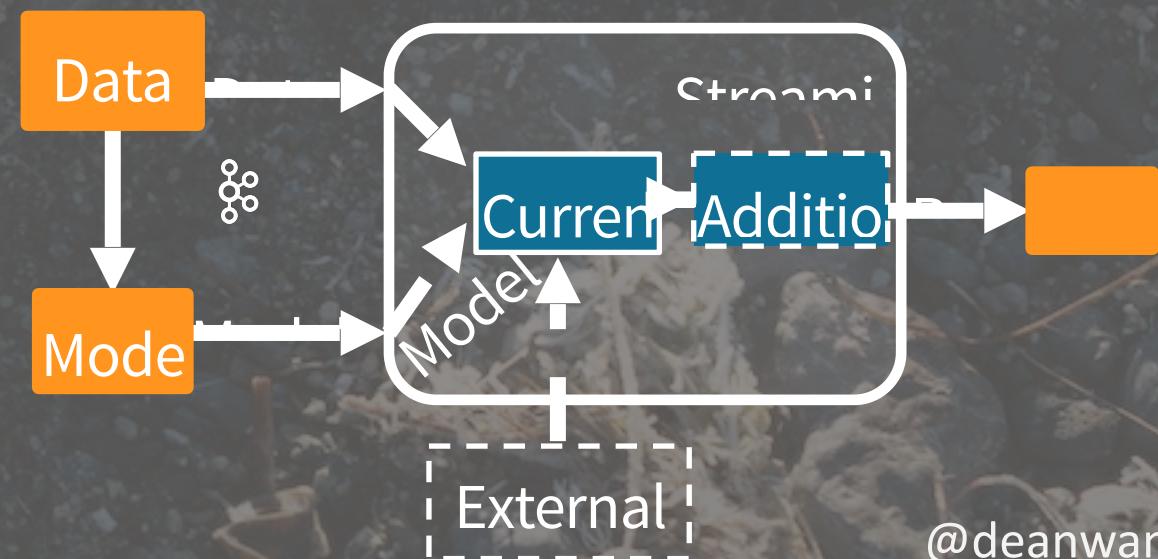
Embedded

- Pros:
 - Minimal interprocess communication overhead
 - Performance tuning focuses on one system, the data pipeline



Embedded

- Cons
 - Model parameters must be serialized
 - Model serving library must be “compatible” with training system:
 - Same algorithms
 - Same performance

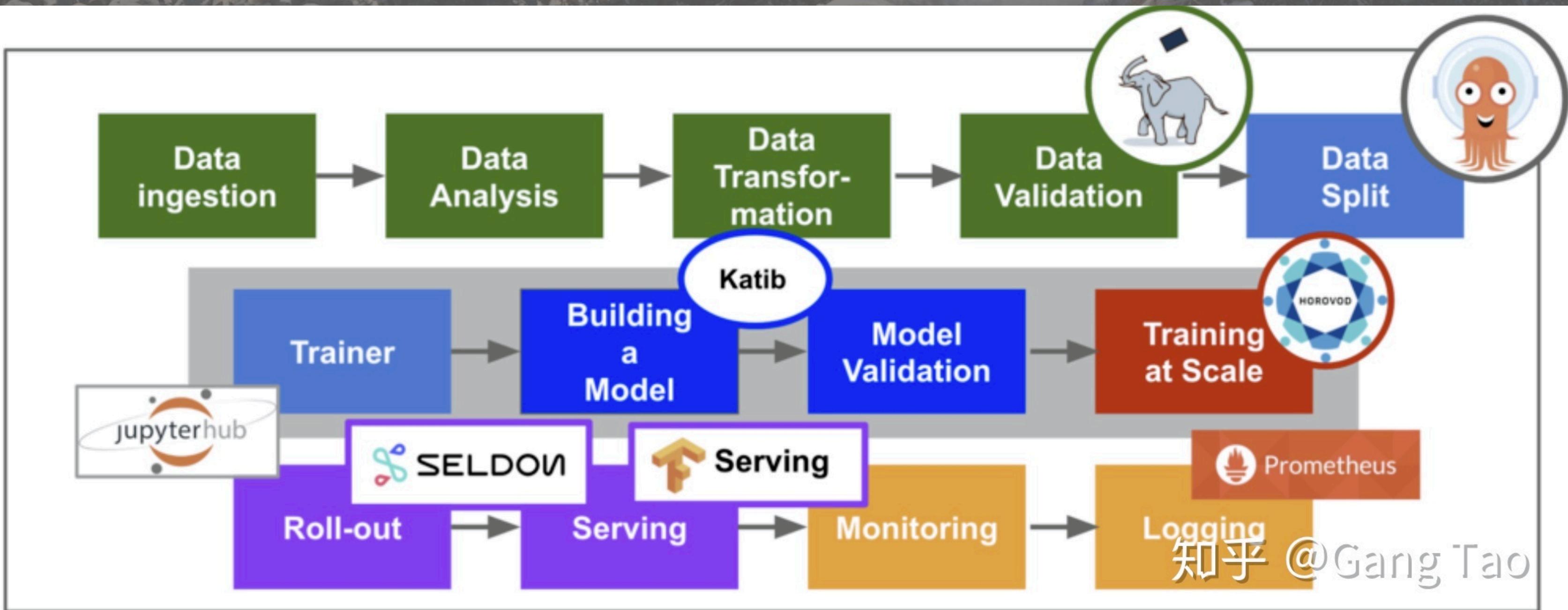


Systems

- Kubeflow - for example
 - For Kubernetes
- Others
 - MLflow
 - AWS SageMaker
 - ...



Systems - Kubeflow





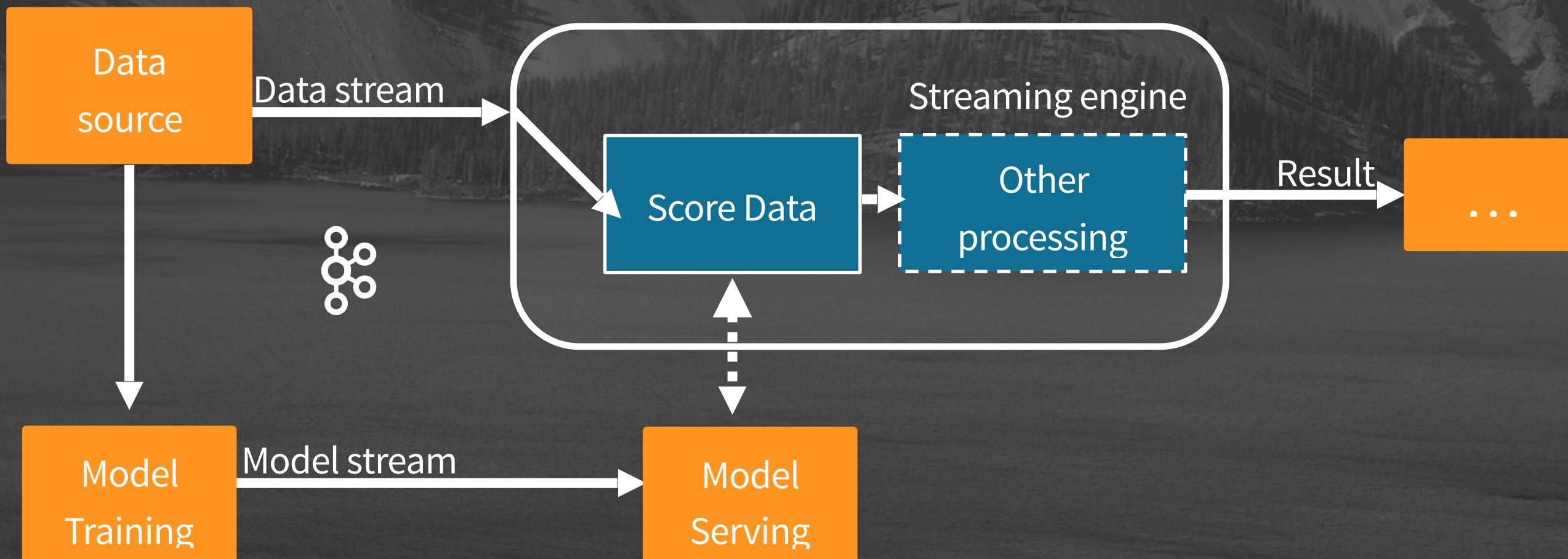
Pragmatic Challenges

Model Updates “in situ”

- “Concept Drift” - models grow stale
 - They have a “half life”, too
- Periodically retrain then serve the new model
 - (Continuous training a research topic)

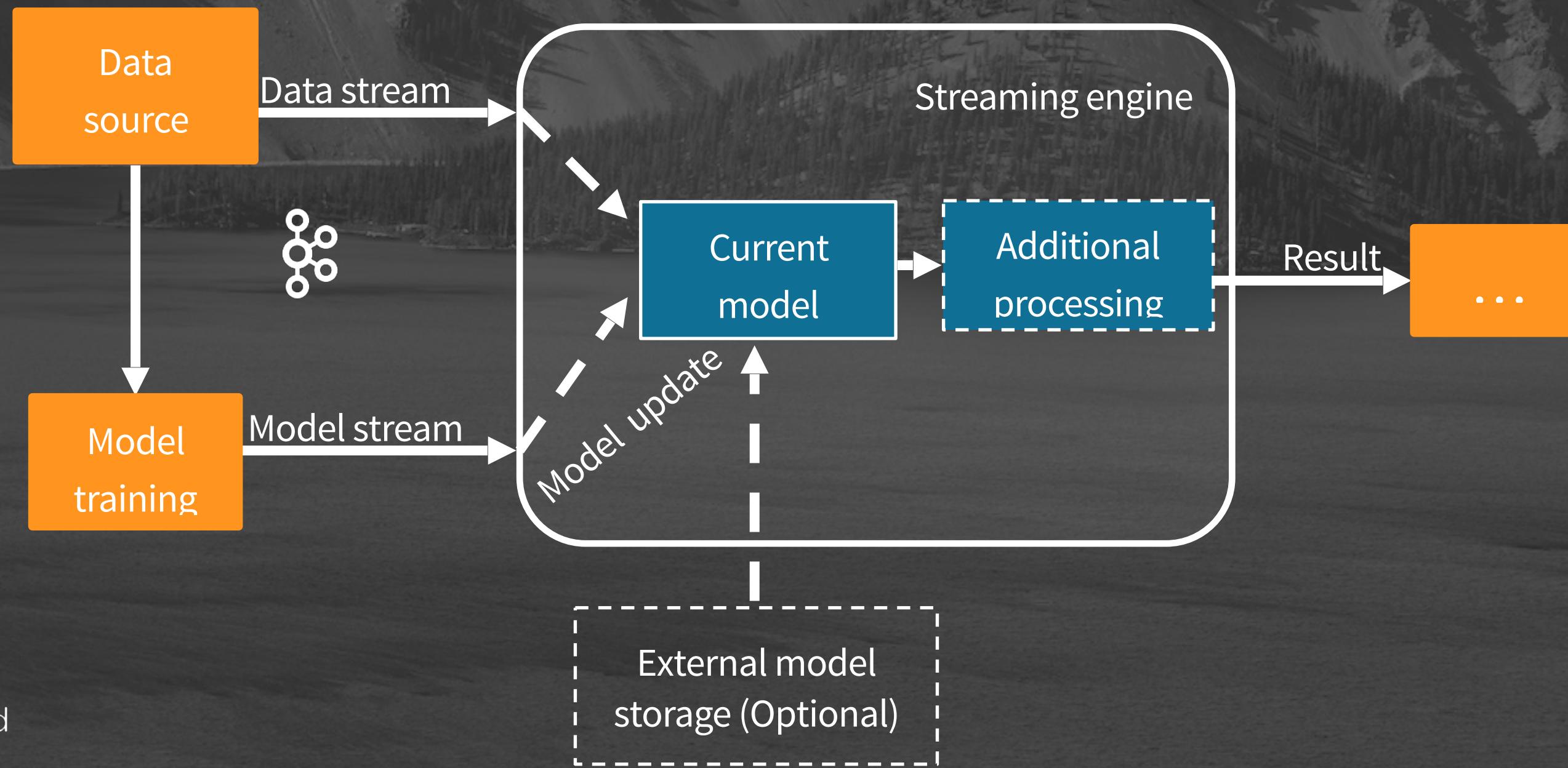
Model Updates “in situ”

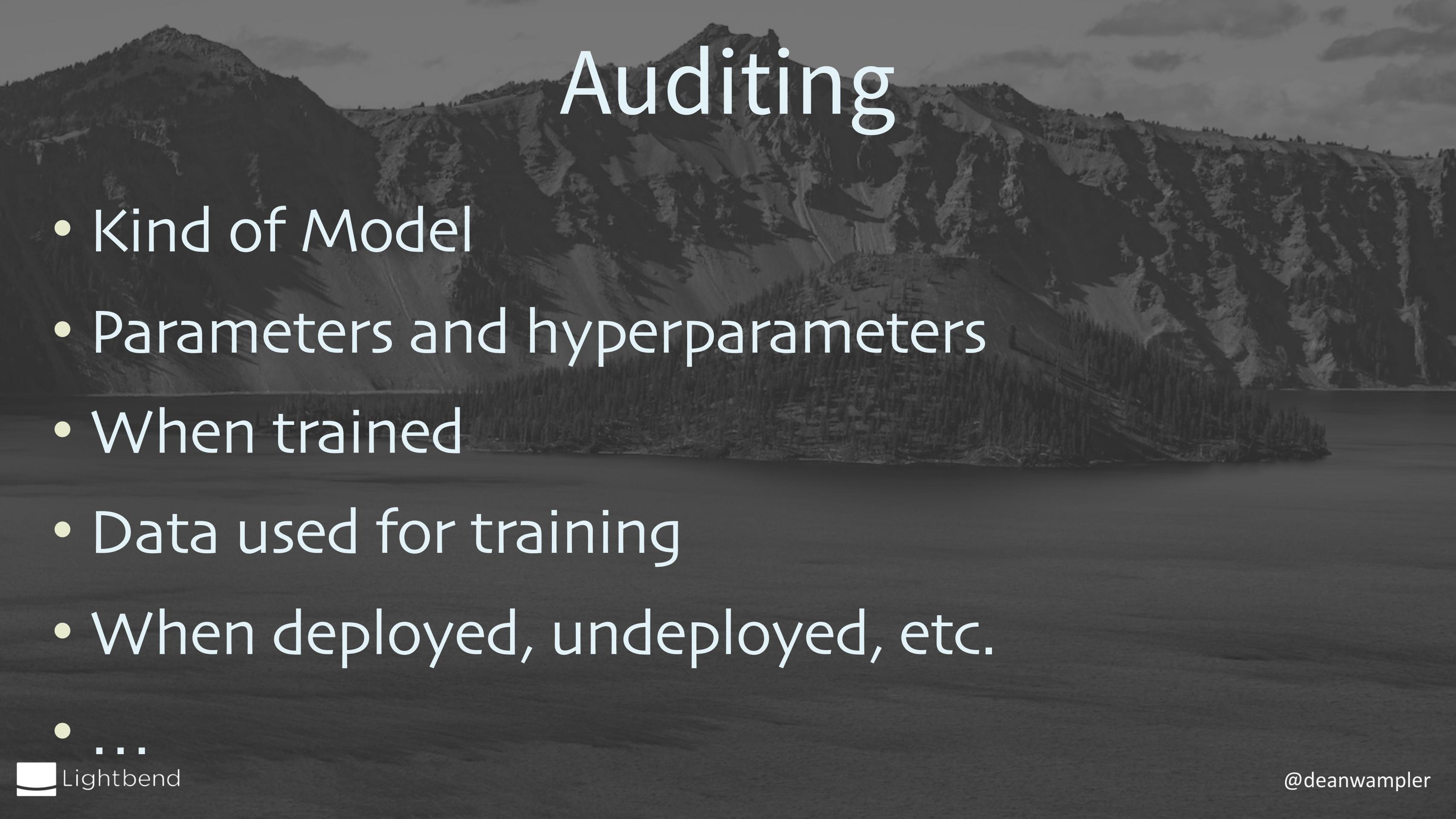
- Model updates and MSaaS



Model Updates “in situ”

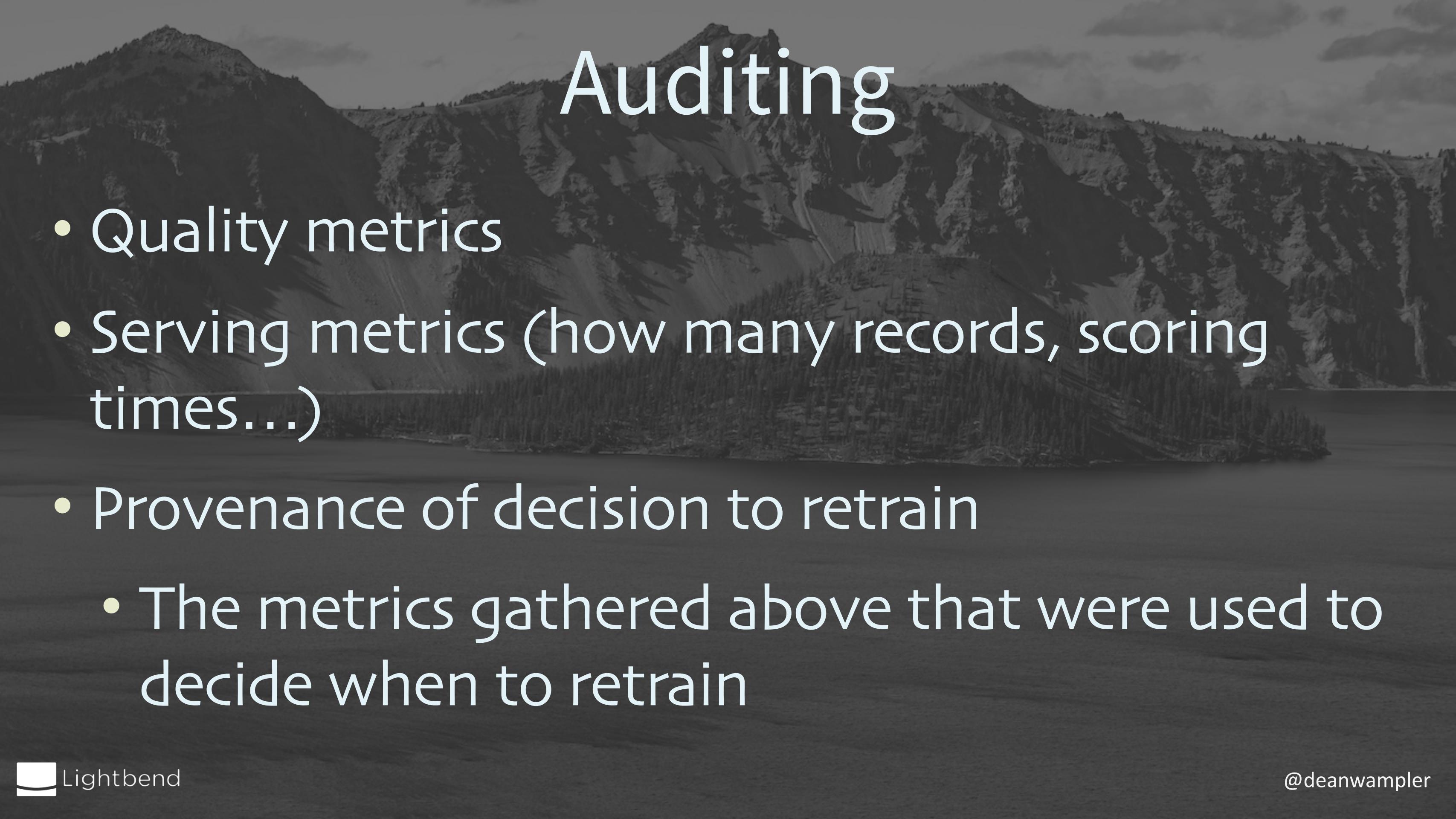
- Embedded model updates





Auditing

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

A black and white photograph of a mountain range with a winding road in the foreground.

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

Retraining Considerations

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

Retraining Considerations

- Approaches:
 - Mature: “periodic” Batch jobs
 - Online and incremental algorithms
 - Future: Auto-adaptive ML (i.e., continual learning). Successor to AutoML



Dusty Milky Way

Mars last Summer



Lightroom

@deanwampler

References

- Ideas:
 - [Ben Lorica on 9 AI Trends](#)
 - [Paco Nathan's Data Governance Talk](#)
 - [distill.pub](#)
 - [The Algorithm](#)
 - [The Gradient](#)

References

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

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

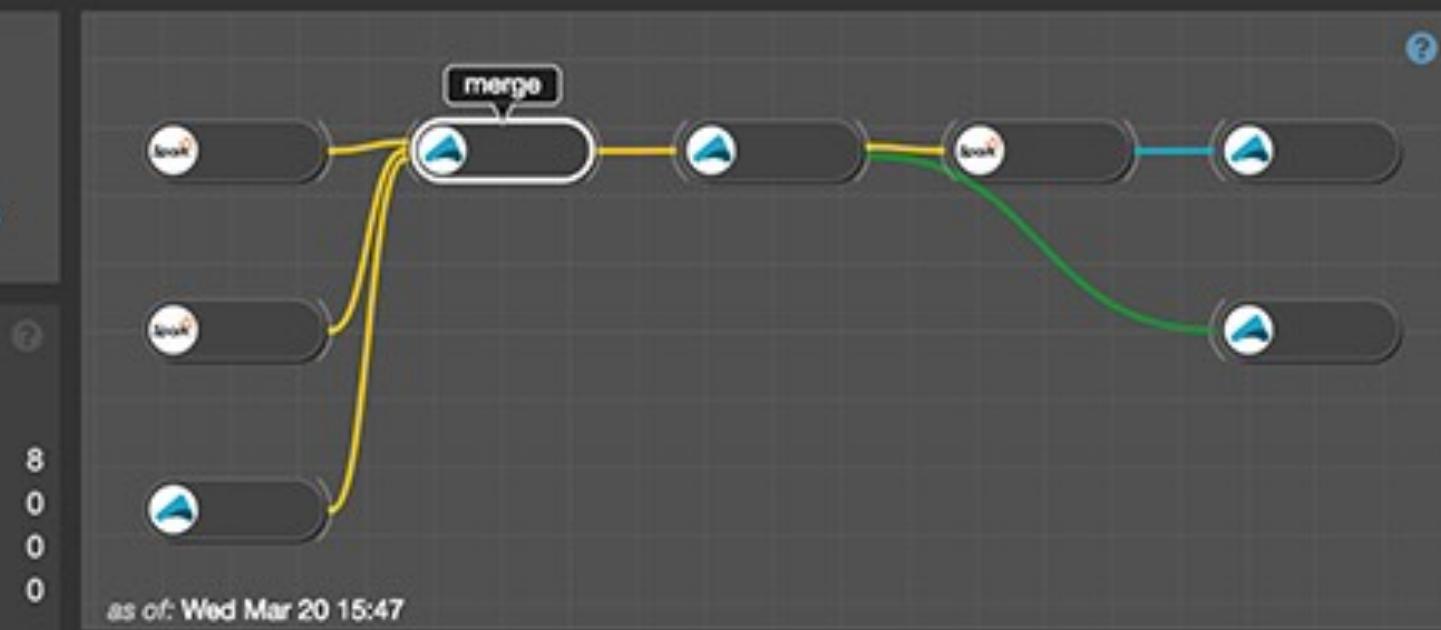
Application Details

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	---



Streamlet Health

Wed 16:00

Buy my stuff!!

Selection: merge

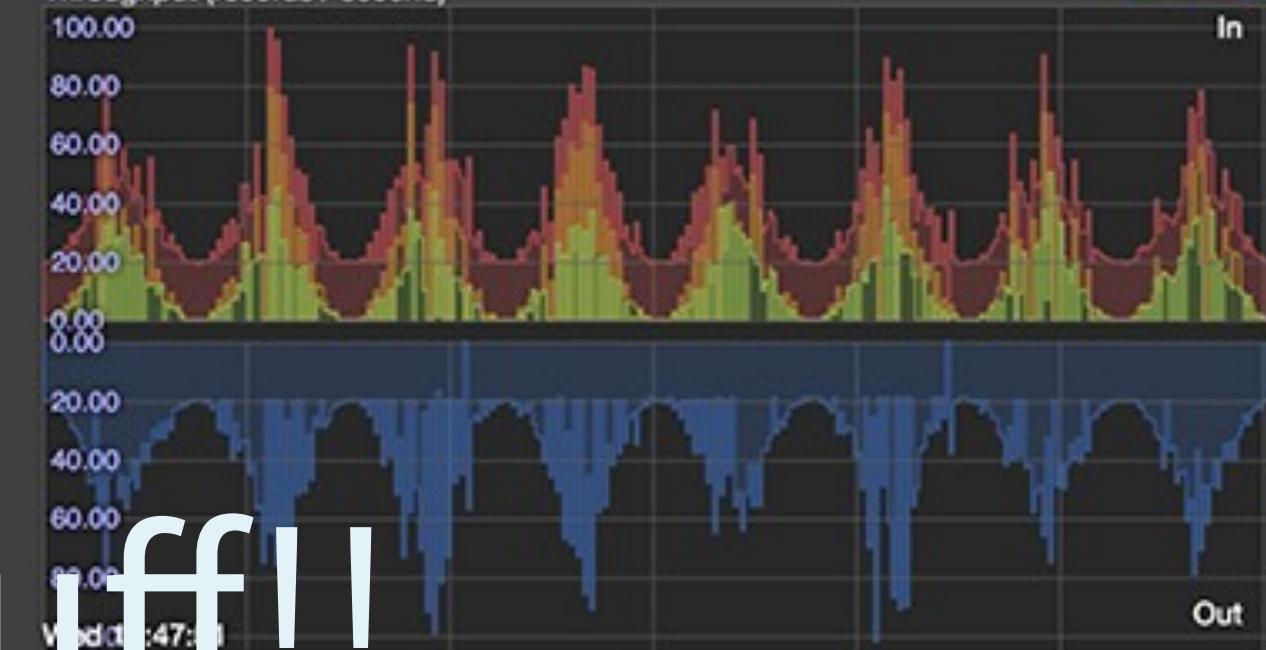
Diagnostics Ports

Streamlet Monitors SORT BY First unhealthy

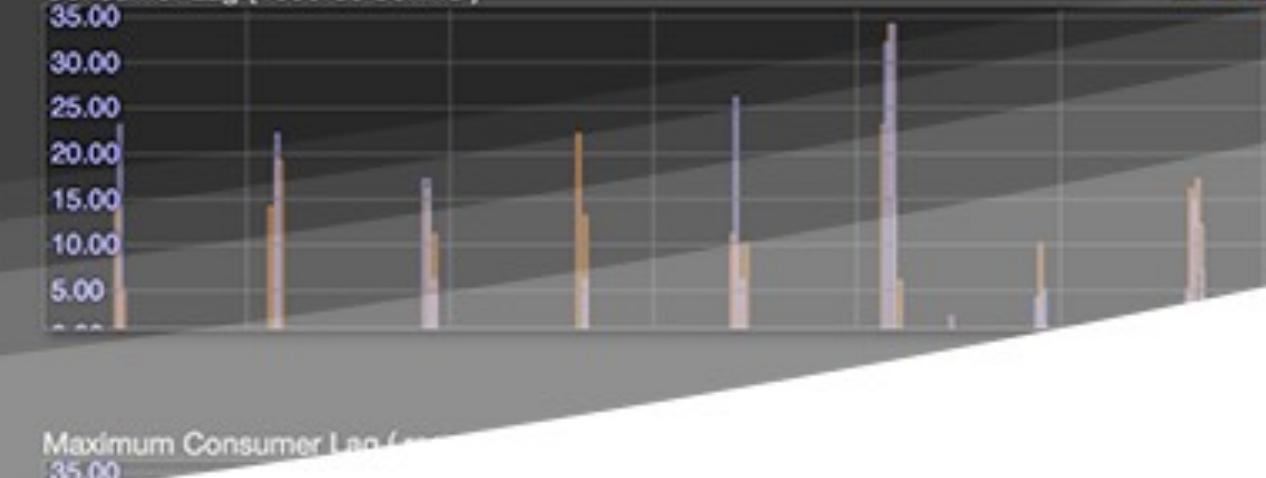
kafka_consumer_lag	green
kafka_consumer_throughput	green
kafka_producer_throughput	green

Metrics

Throughput (records / second)



Consumer Lag (records behind)



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