Introducing FlinkML

Boris Lublinsky
Stavros Kontopoulos
Lightbend

Talk's agenda

- Machine learning
- Models
- What are building?
- Results and next steps

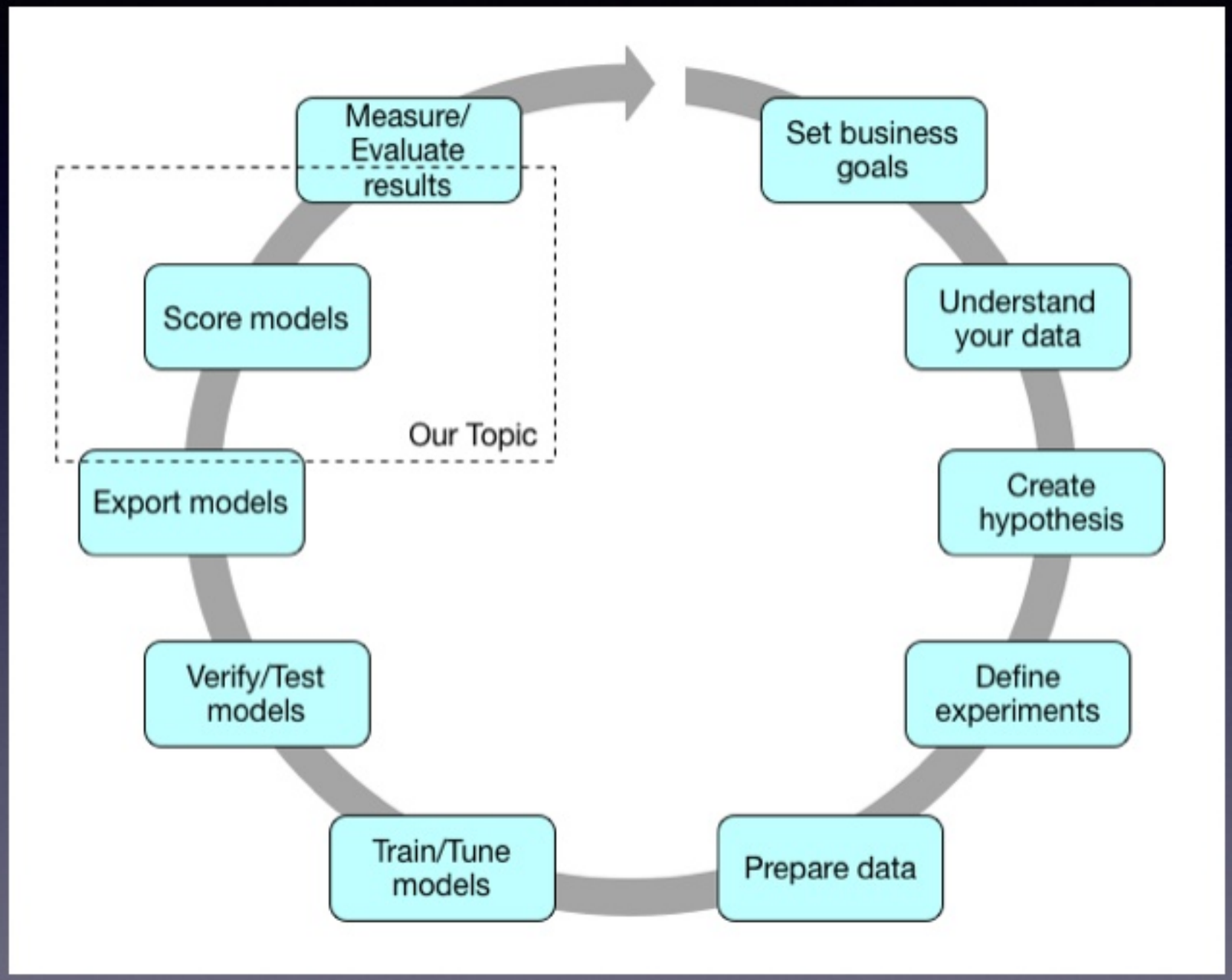
How people see ML

Data

Magic

Happiness

Reality


Measure/
Evaluate
results
Score models
Our Topic
Export models
Verify/Test
models
Train/Tune
models
Prepare data
Define
experiments
Create
hypothesis
Understand
your data
Set business
goals

What is the Model?

A model is a function transforming inputs to outputs - y = f(x),for example:

Linear regression:

$$y = a_c + a_1 * x_1 + \ldots + a_n * x_n$$

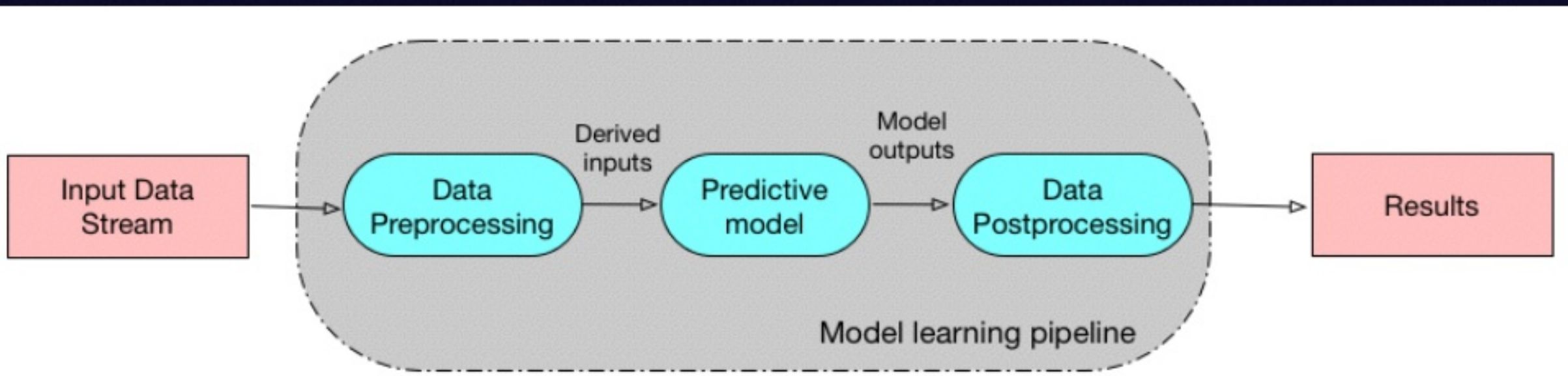
Neural network:

$$f(x) = K\left(\sum_i w_i g_i(x)\right)$$

Such a definition of the model allows for an easy implementation of model's composition. From the implementation point of view it is just function composition

Model learning pipeline

UC Berkeley AMPLab introduced machine learning pipelines as a graph defining the complete chain of data transformation.



A single pipeline can encapsulate more then one predictive model. From a serving point of view its still a single pipeline.

Models proliferation

Data Scientist

Software engineer

Model standardization

PMML
Predictive Model Markup Language

PFA
stands for
Portable Format for Analytics
Abbreviations.com

TensorFlow

Model lifecycle considerations

- Models tend to change
- Update frequencies vary greatly - from hourly to quarterly/yearly
- Model version tracking
- Model release practices
- Model update process

Customer SLA

- Response time - time to calculate prediction
- Throughput - predictions per second
- Support for running multiple models
- Model update - how quickly and easily can the model be updated
- Uptime/reliability

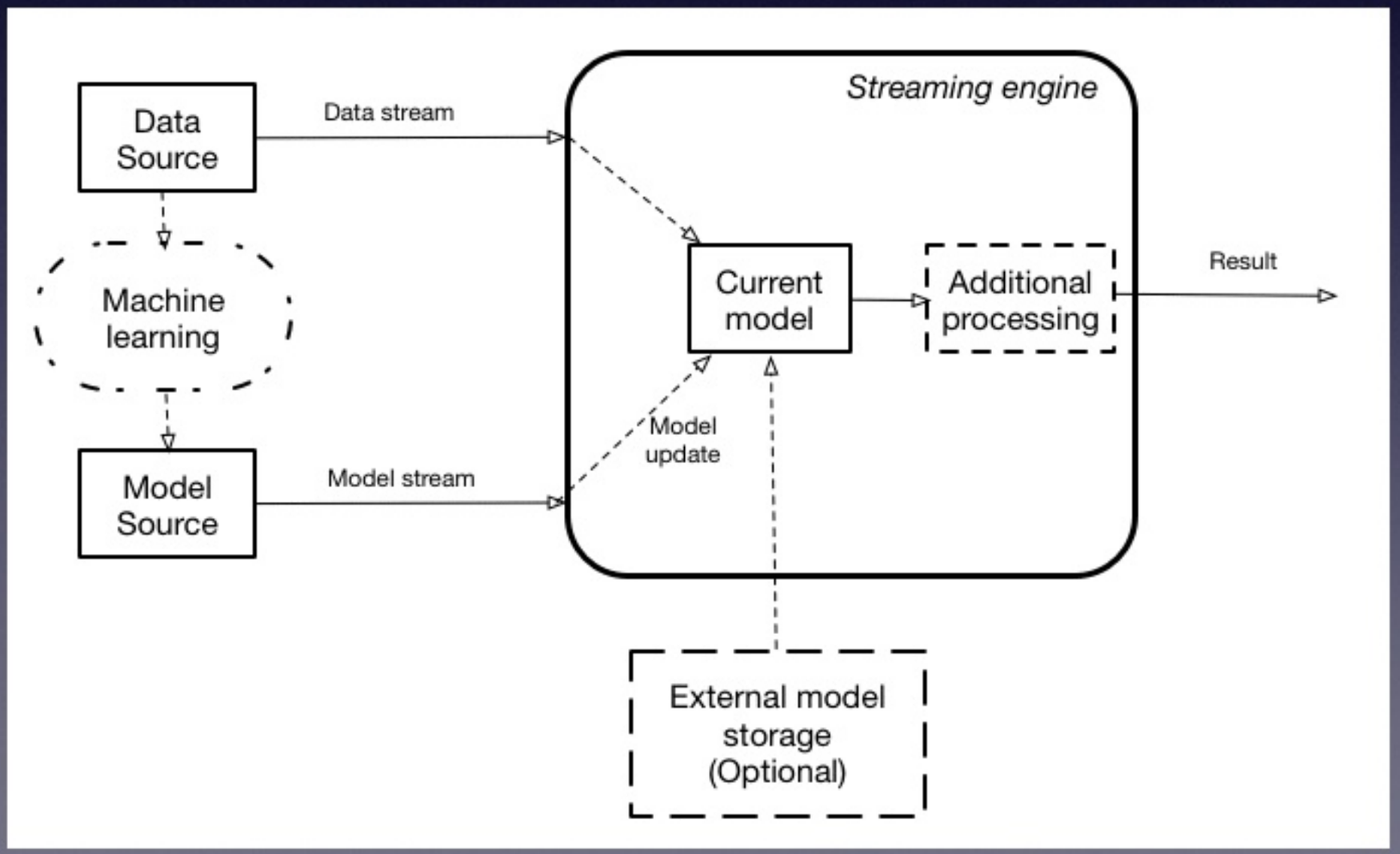
Model Governance

Models should be:

- governed by the company's policies and procedures, laws and regulations and organization's goals
- be transparent, explainable, traceable and interpretable for auditors and regulators.
- have approval and release process.
- have reason code for explanation of decision.
- be versioned. Reasoning for introduction of new version should be clearly specified.
- searchable across company

What are we building

We want to build a streaming system allowing to update models without interruption of execution (dynamically controlled stream).



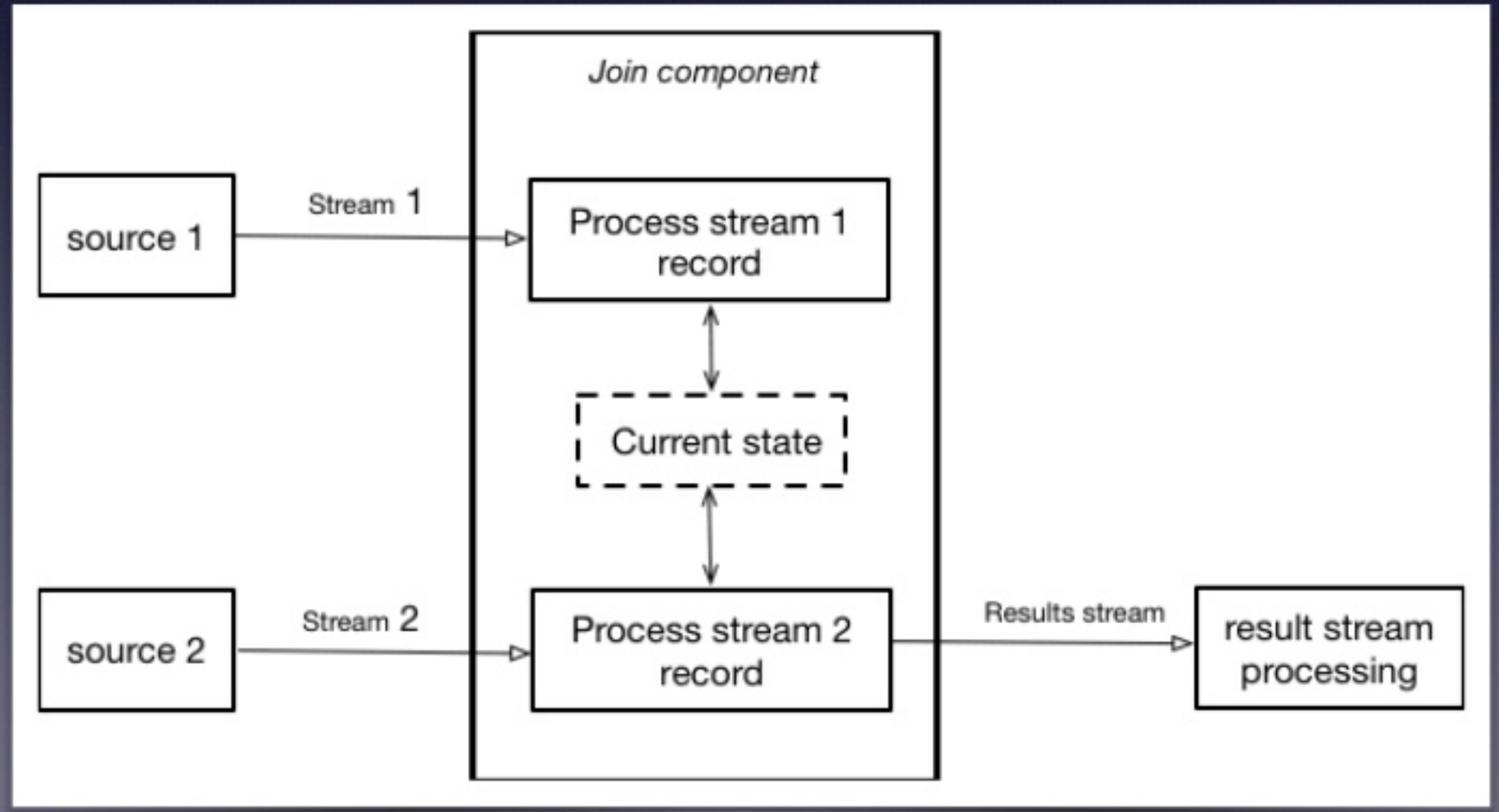
Exporting Model

Different ML packages have different export capabilities. We used the following:

- For Spark ML export to PMML - https://github.com/jpmml/jpmml-sparkml
- For Tensorflow:
  - Normal graph export
  - Saved model format

Flink Low Level Join

- Create a state object for one input (or both)
- Update the state upon receiving elements from its input
- Upon receiving elements from the other input, probe the state and produce the joined result



Model representation

On the wire

```
syntax = "proto3";

// Description of the trained model.
message ModelDescriptor {
  // Model name
  string name = 1;
  // Human readable description.
  string description = 2;
  // Data type for which this model is applied.
  string dataType = 3;
  // Model type
  enum ModelType {
    TENSORFLOW  = 0;
    TENSORFLOWSAVED = 2;
    PMML = 2;
  };
  ModelType modeltype = 4;
  oneof MessageContent {
    // Byte array containing the model
    bytes data = 5;
    string location = 6;
  }
}
```

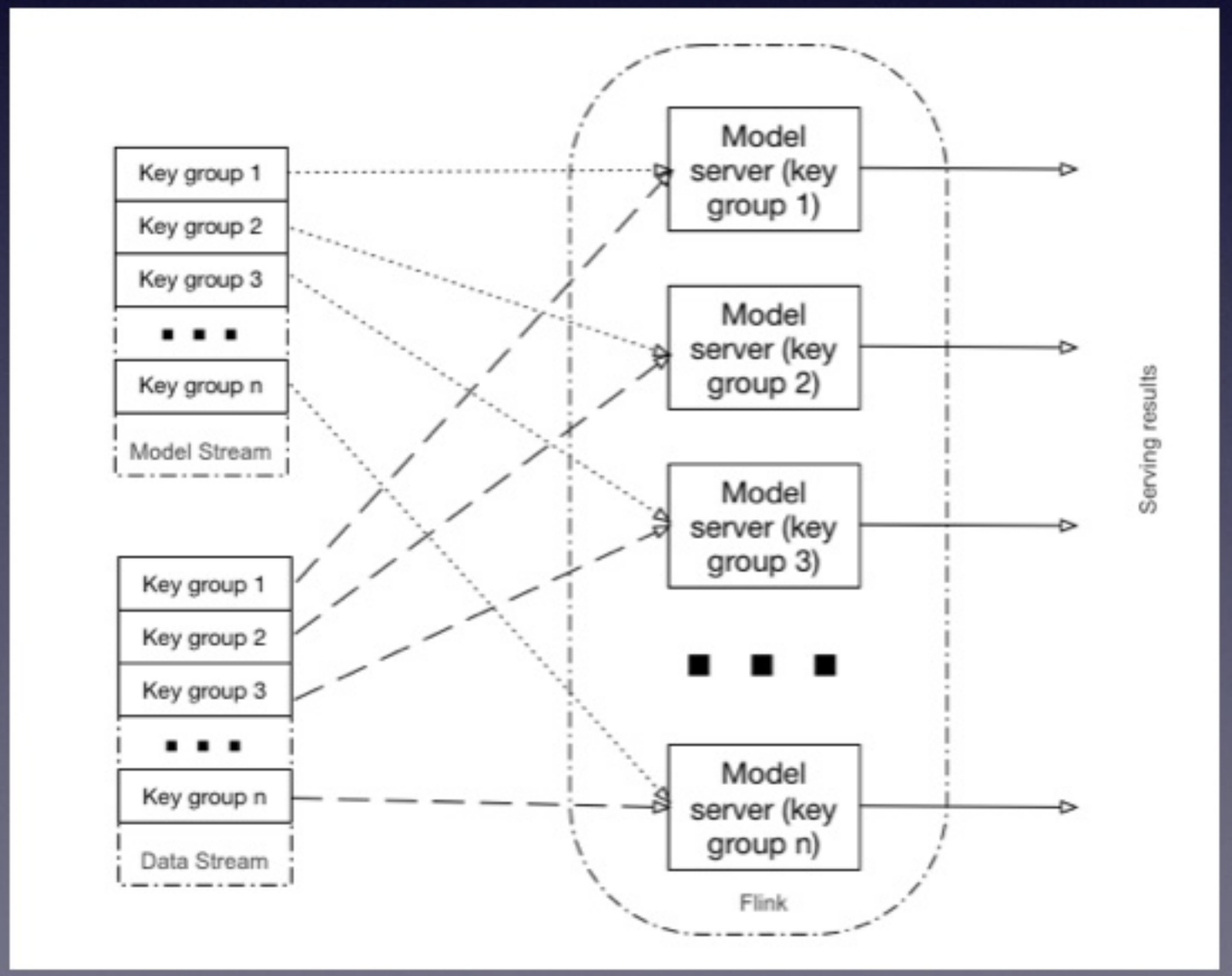
Internal

```
trait Model {
 def score(input : AnyVal) :
AnyVal
 def cleanup() : Unit
 def toBytes() : Array[Byte]
 def getType : Long
}

trait ModelFactoryl {
 def create(input :
ModelDescriptor) : Model
 def restore(bytes : Array[Byte]) :
Model
}
```

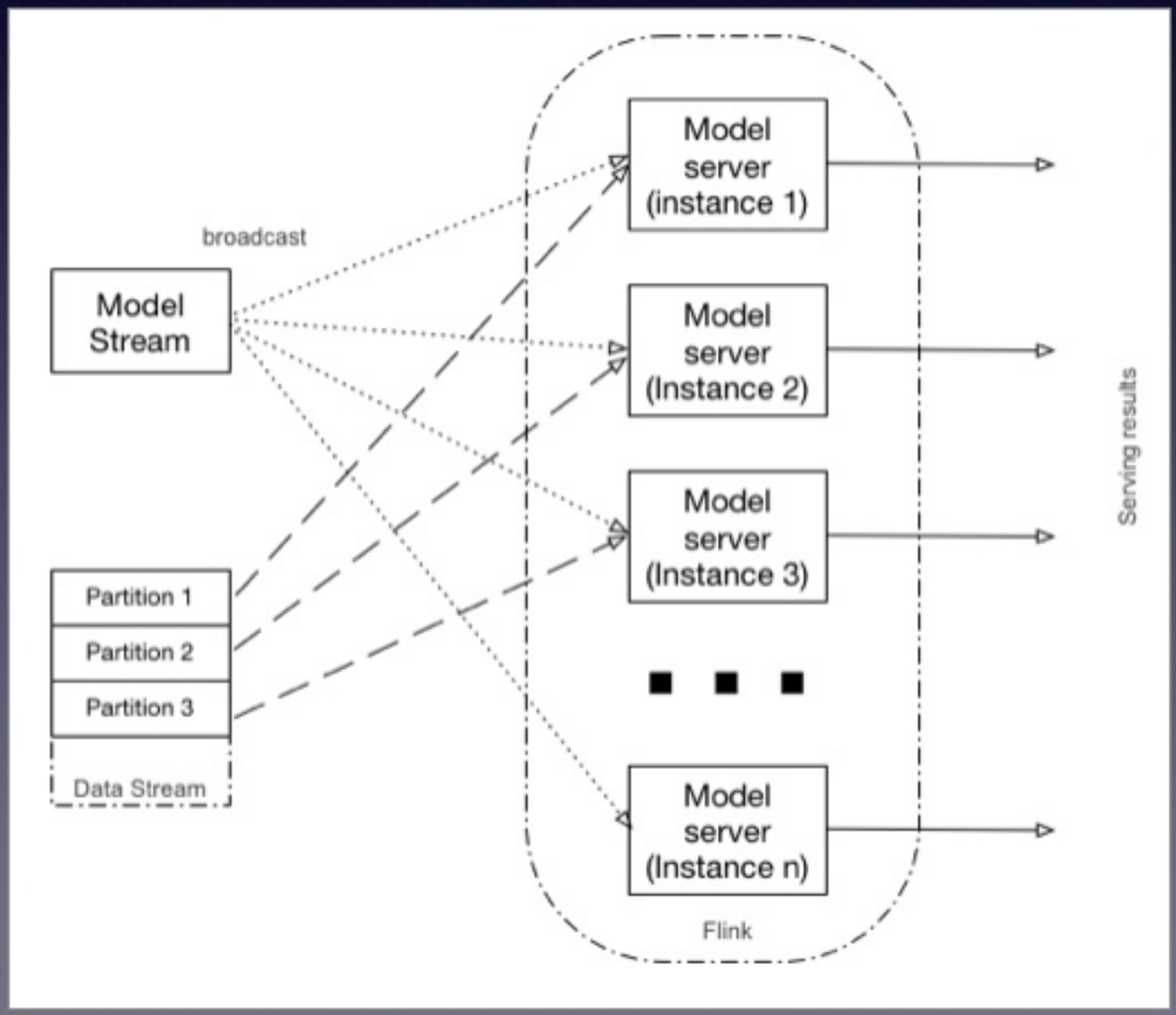
Key based join

Flink's *CoProcessFunction* allows key-based merge of 2 streams. When using this API, data is key-partitioned across multiple Flink executors. Records from both streams are routed (based on key) to the appropriate executor that is responsible for the actual processing.



Partition based join

Flink's *RichCoFlatMapFunction* allows merging of 2 streams in parallel (based on parralelization parameter). When using this API, on the partitioned stream, data from different partitions is processed by dedicated Flink executor.



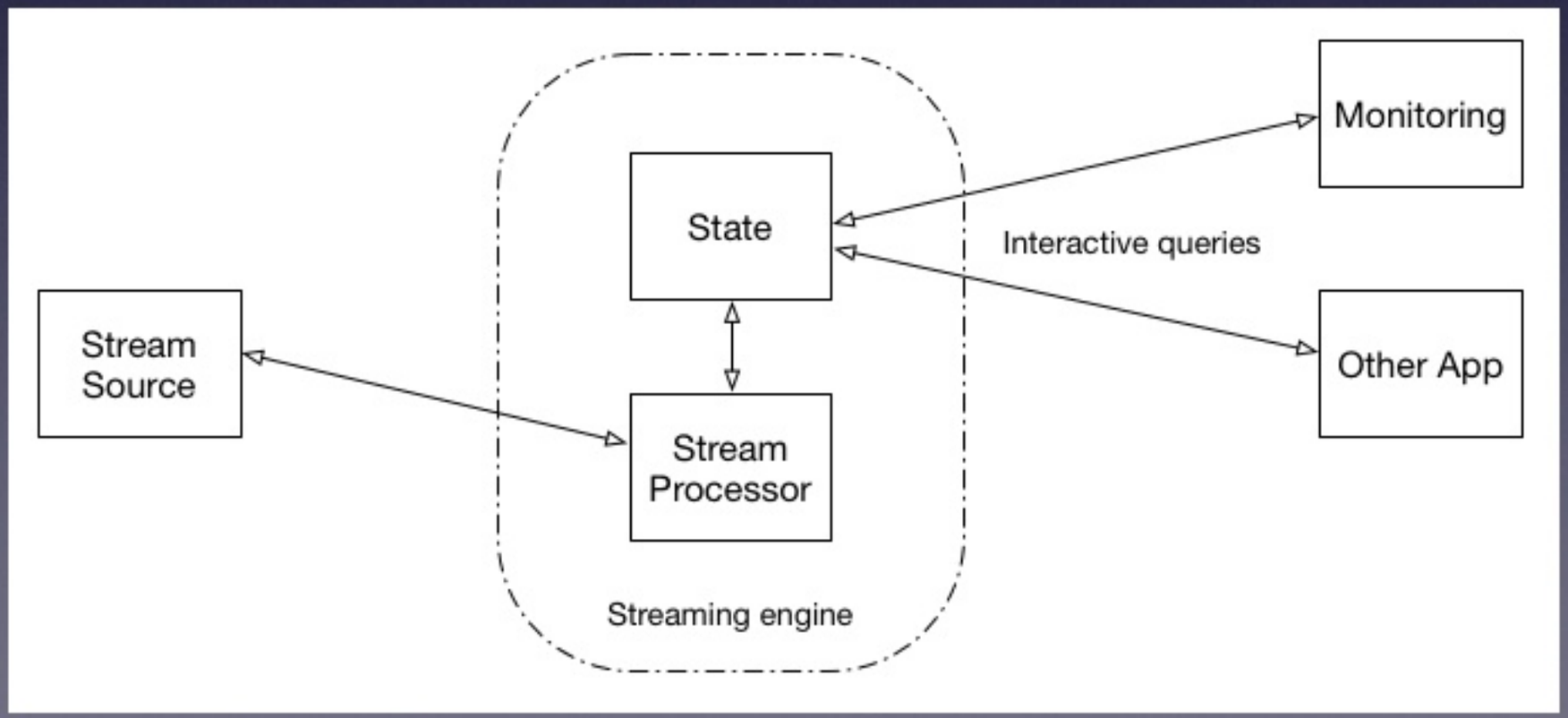
Monitoring

Model monitoring should provide information about usage, behavior, performance and lifecycle of the deployed models

```
case class ModelToServeStats(
        name: String,                                   //  Model name
        description: String,                            // Model descriptor
        modelType: ModelDescriptor.ModelType,           // Model type
        since : Long,                                   // Start time of model usage
        var usage : Long = 0,                           // Number of servings
        var duration : Double = .0,                     // Time spent on serving
        var min : Long = Long.MaxValue,                 // Min serving time
        var max : Long = Long.MinValue                  // Max serving time
)
```

Queryable state

Queryable state (interactive queries) is an approach, which allows to get more from streaming than just the processing of data. This feature allows to treat the stream processing layer as a lightweight embedded database and, more concretely, *to directly query the current state* of a stream processing application, without needing to materialize that state to external databases or external storage first.



Where are we now?

- Proposed solution does not require modification of Flink proper. It rather represents a framework build leveraging Flink capabilities. Although extension of the queryable state will be helpful

- Flink improvement proposal (Flip23) - model serving is close to completion - https://docs.google.com/document/d/1ON_t9S28_2LJ91Fks2yFw0RYyeZvIvndu8oGRPsPuk8).

- Fully functioning prototype of solution supporting PMML and Tensorflow is currently in the FlinkML Git repo (https://github.com/FlinkML)

Next steps

- Complete Flip23 document
- Integrate Flink Tensorflow and Flink PMML implementations to provide PMML and Tensorflow specific model implementations - work in progress
- Look at bringing in additional model implementations
- Look at model governance and management implementation

Thank you

Any Questions?