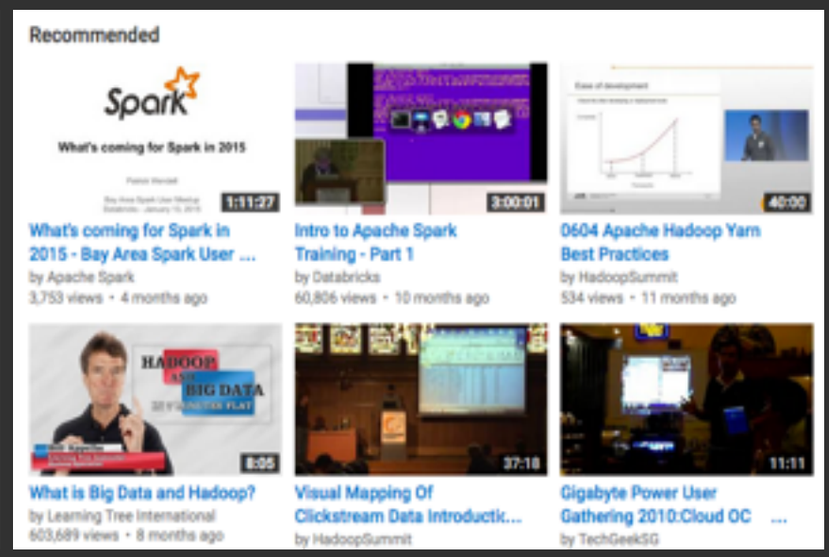
Apache PredictionIO
End-to-End Machine Learning Server
with Apache Spark

What is Machine Learning?

What is Machine Learning?

Examples of machine learning applications:

- Facial recognition software
- E-mail spam detectors
- Automated personalized recommendations

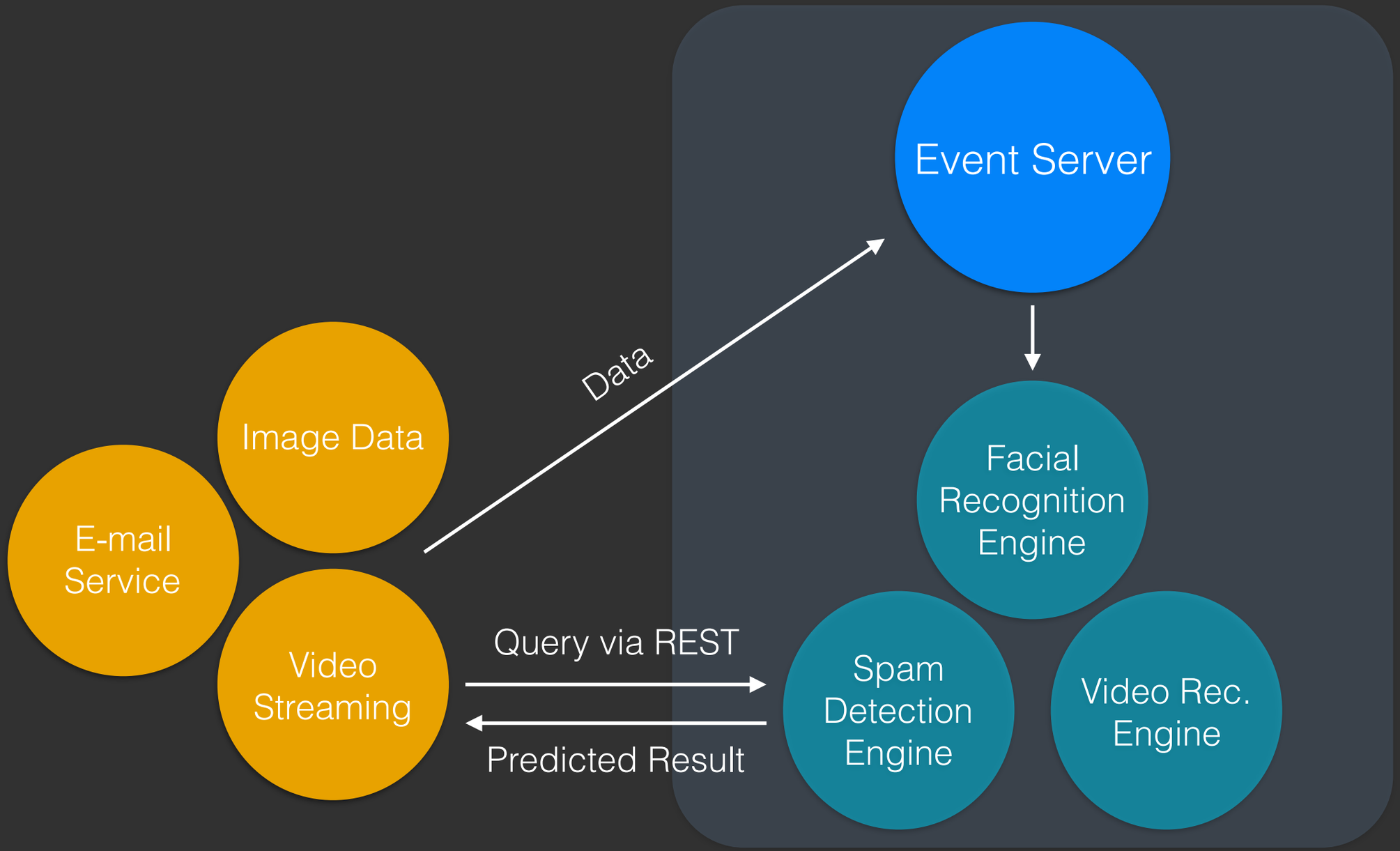


PredictionIO

What is Apache PredictionIO?

PredictionIO

What is Apache PredictionIO?



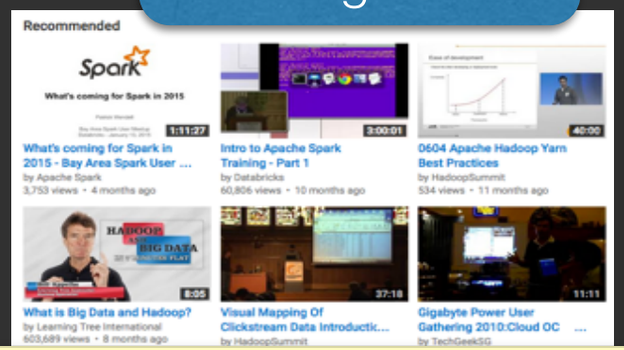
PredictionIO

An **Engine Template** is the basic skeleton of an engine.

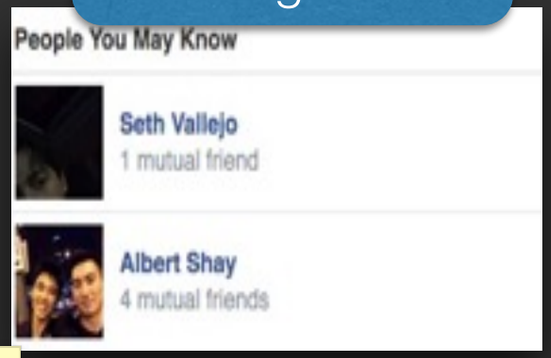
Recommendation Engine Template

Text Classification Engine Template

Item Recommendation Engine



User Recommendation Engine



Semantic Analysis Engine

Spam Detector Engine

Engine user usage: quick, ready, and customizable ML solutions

PredictionIO

Engine

Building an Engine with Separation of Concerns (SoC)

- DASE - the "MVC" for Machine Learning
- Data: Data Source and Data Preparator
- Algorithm(s)
- Serving
- Evaluator

Engine Functions of an Engine

A. Train deployable predictive model(s)

B. Respond to dynamic query

C. ~~Evaluation~~

Engine A. Train predictive model(s)

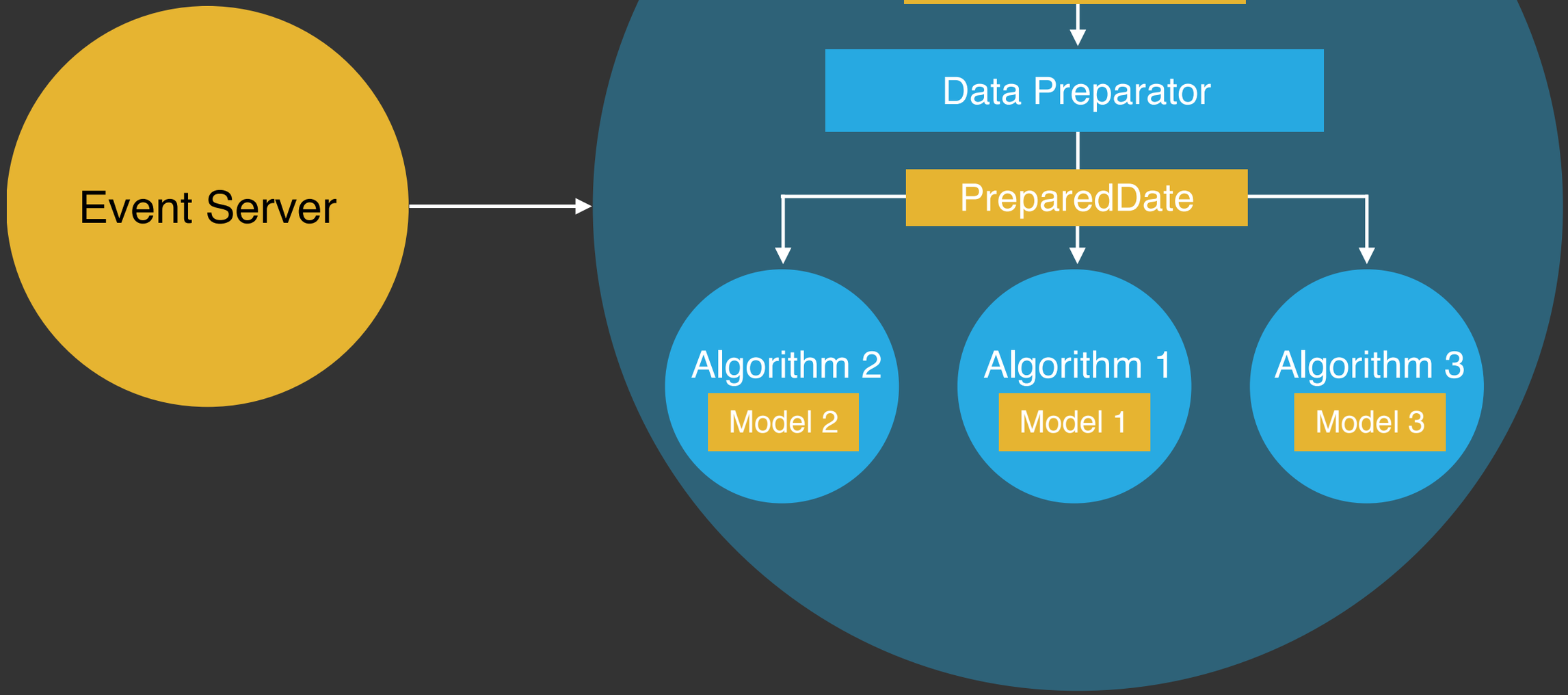
class DataSource(…) extends PDataSource
def readTraining(sc: SparkContext)
==> trainingData

$ pio train

class Preparator(…) extends PPreparator
def prepare(sc: SparkContext, trainingData: TrainingData)
==> preparedData

class Algorithm1(…) extends PAlgorithm
def train(prepareData: PreparedData)
==> Model

Engine A. Train predictive model(s)

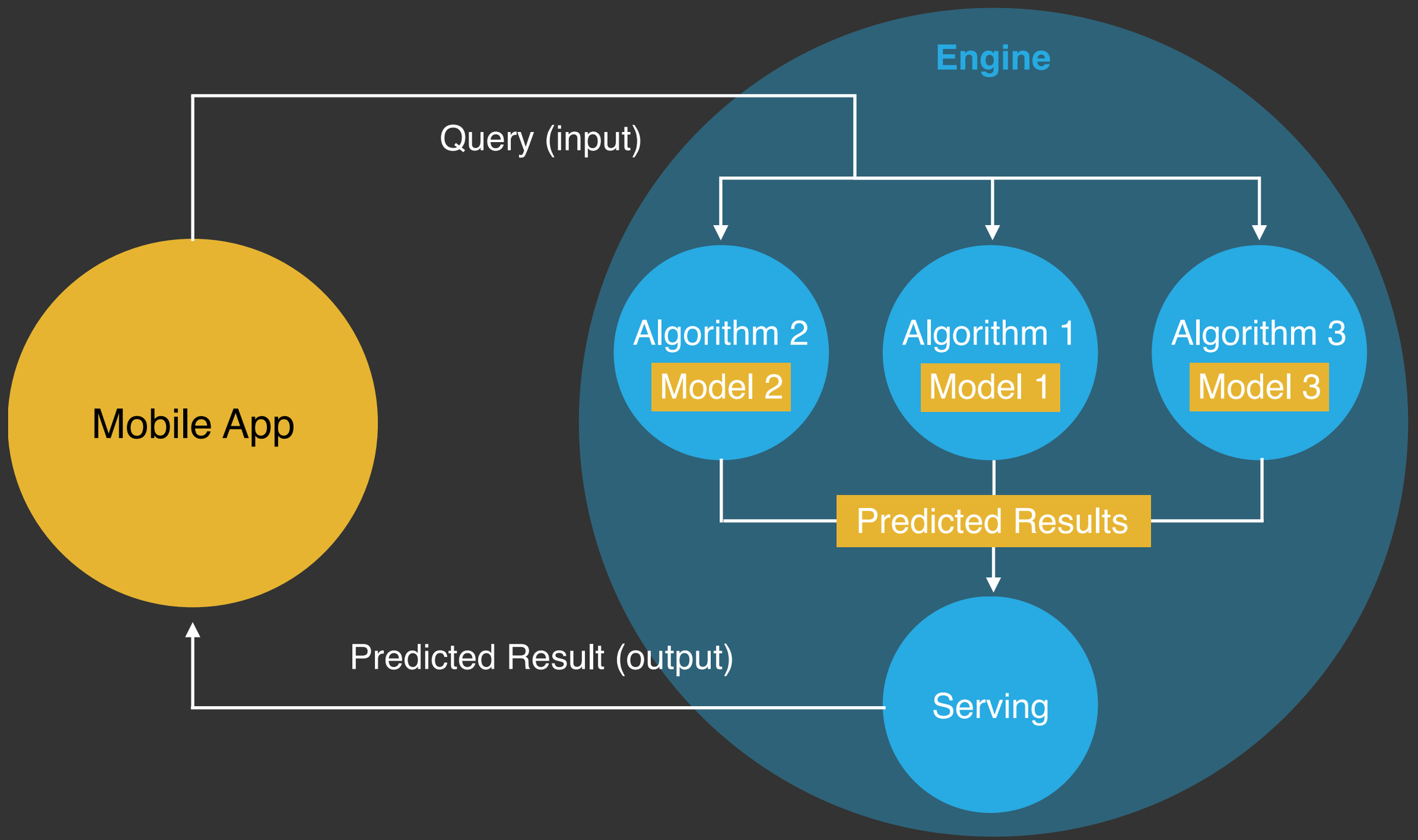


Engine B. Respond to dynamic query

class Algorithm1(…) extends PAlgorithm
def predict(model: ALSModel, query: Query)
==> predictedResult

class Serving extends LServing
def serve(query: Query, predictedResults: Seq[PredictedResult])
==> predictedResult

Engine B. Respond to dynamic query



Engine DASE Factory

```
object RecEngine extends IEngineFactory {
  def apply() = {
    new Engine(
      classOf[DataSource],
      classOf[Preparator],
      Map("algo1" -> classOf[Algorithm1]),
      classOf[Serving])
  }
}
```

Running on Production

- Install PredictionIO
  $ bash -c "$(curl -s http://install.prediction.io/install.sh)"

- Start the Event Server
  $ pio eventserver

- Deploy an Engine
  $ pio build; pio train; pio deploy

- Update Engine Model with New Data
  $ pio train; pio deploy

Production Features

- Model Updating and Versioning
- Offline and Online Evaluation
- Multiple Engine Variants
- Query and Prediction Tracking

PredictionIO

Universal Recommender

https://templates.prediction.io

PredictionIO

More Information

Website: http://predictionio.incubator.apache.org/

Github: https://github.com/apache/incubator-predictionio

PredictionIO