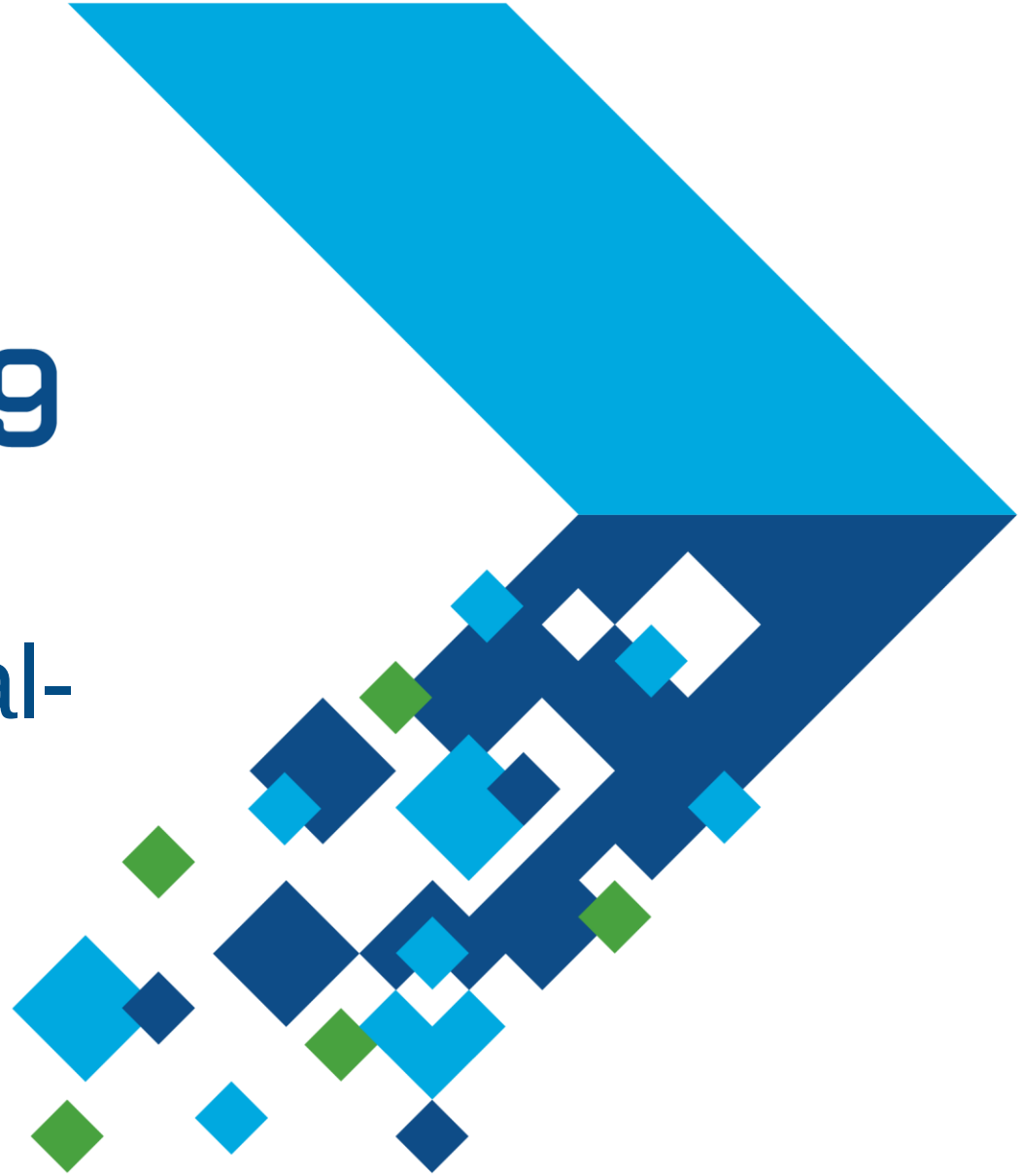
MATLAB EXPO 2019

Deploying AI for Near Real-Time Manufacturing Decisions

Pierre Harouimi

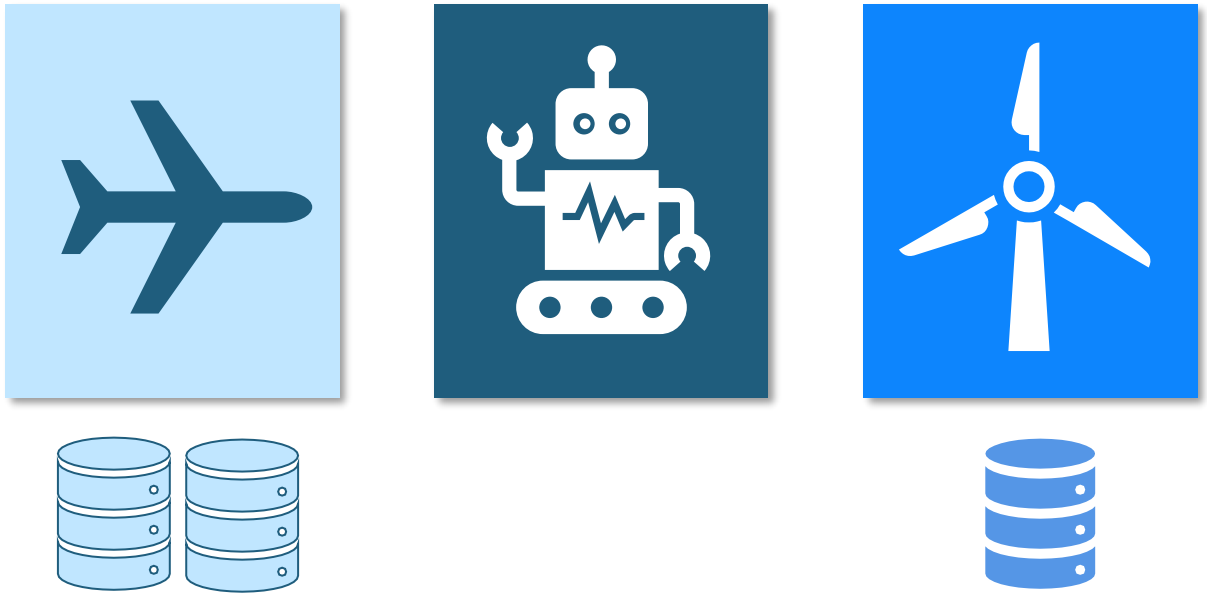


The Need for Large-Scale Streaming

Predictive Maintenance

Increase Operational Efficiency

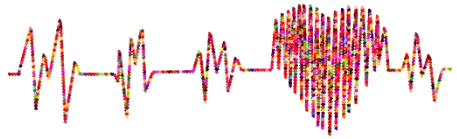
Reduce Unplanned Downtime



Medical Devices

Patient Safety

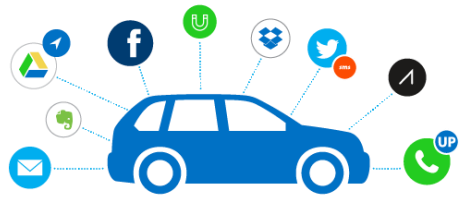
Better Treatment Outcomes



Connected Cars

Safety, Maintenance

Advanced Driving Features



Finance

High Frequency Trading

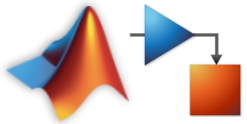
Sentiment Analysis

Example Problem: develop and operationalize a machine learning model to predict failures in industrial pumps

Current system requires Operator to manually monitor operational metrics for **anomalies**. Their expertise is required to detect and take preventative action.

Process Engineer

Develops models in MATLAB and Simulink



System Architect

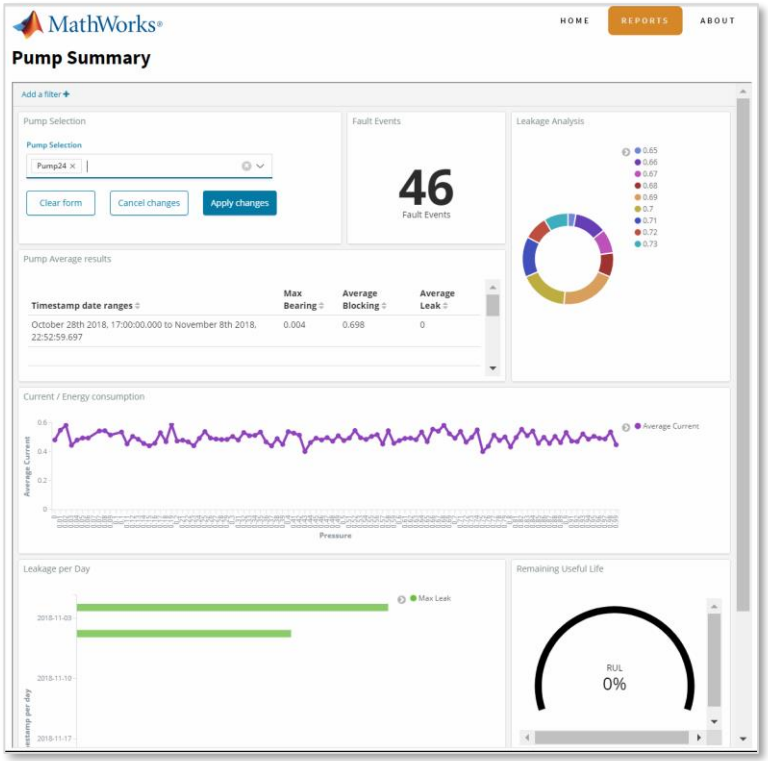
Deploys and operationalizes model on Azure cloud

Azure

Operator

Makes operational decisions based on model output

kibana



Project statement: develop end-to-end predictive maintenance system and demo in one 3-4 week sprint

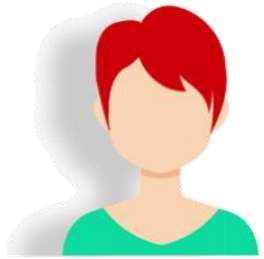
Operator

Monitor **flow**, **pressure**, and **current** of each pump so I always know their operational state

Need **alert** when fault parameters drift outside an acceptable range so I can take **immediate action**

Continuous estimate of pump's **remaining useful life (RUL) & classification** → schedule maintenance or replace the asset

Project statement: constraints & solution



Process Engineer

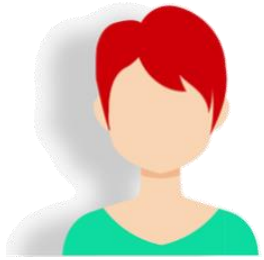
I have few or zero failure data

Generate realistic synthetic data / use Machine Learning models

Architect IT

I have a limited budget, and don't know the adjusted platform

Leverage cloud platform to quickly configure it

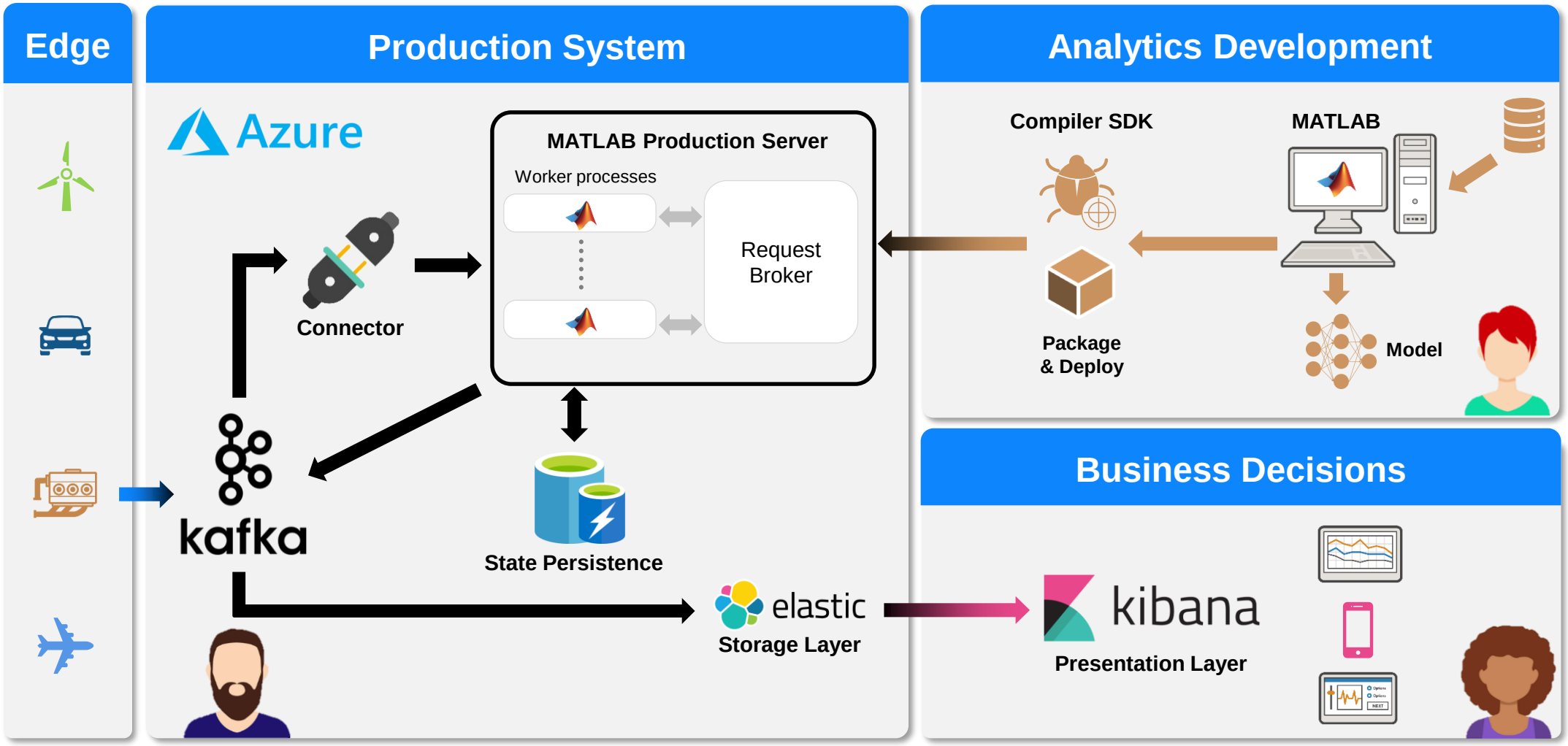


Process Engineer

We need multiple tools for multidisciplinary problems

Use MATLAB and integrate with other environments

Predictive Maintenance Architecture on Azure



Review model requirements

Type of fault

RUL

Operator

Time-windowing
Out-of-order delivery

Test code

Scalable code

System Architect

Process Engineer

A complete end-to-end workflow

Access Data

Files

Database & Cloud

Sensors

Preprocess

Multiple formats

Messy Data

Arrange data

Identify Features

Data Transformation

Feature Extraction

Feature Selection

Predictive Analytics

Model Creation Machine Learning

Parameter Optimization

Model Validation

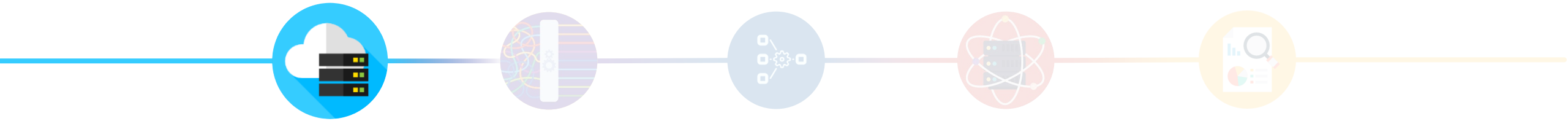
Deploy & Integrate

Enterprise Scale System

Visualization

Web Apps

Access/Generate data

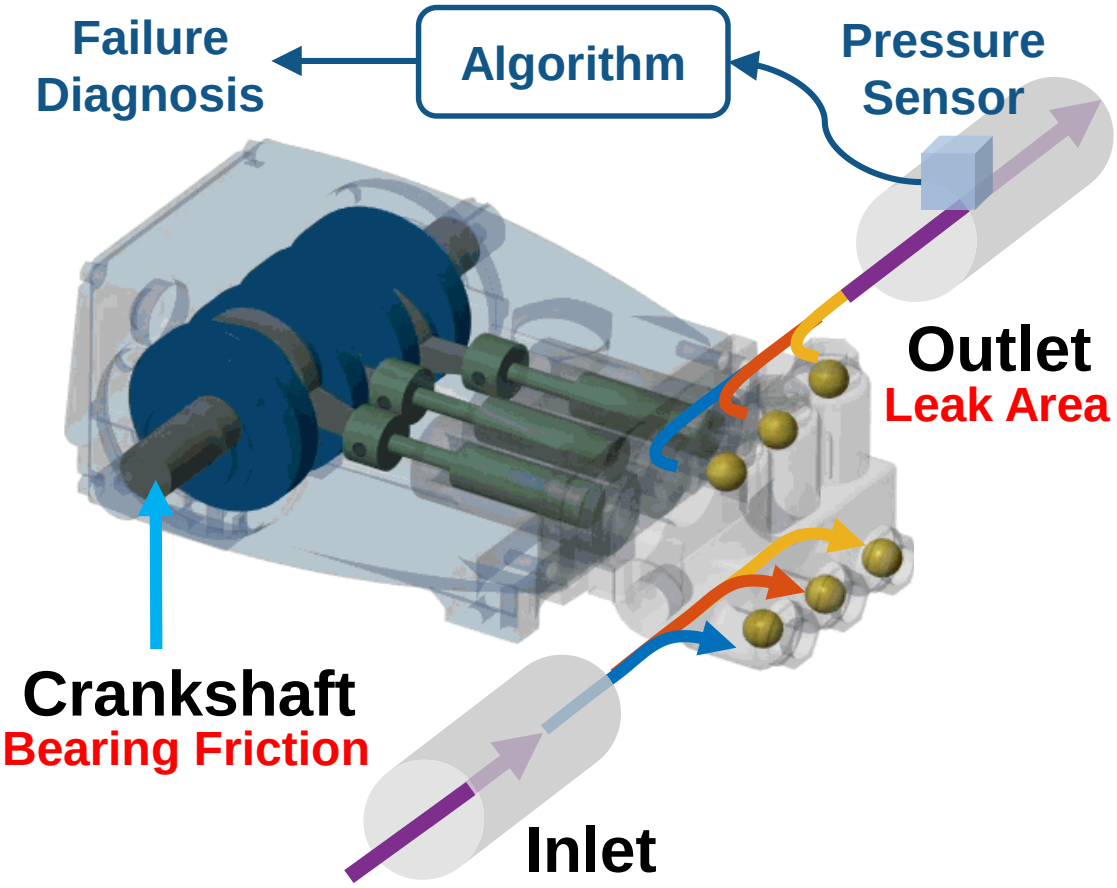


Access/
Generate Data

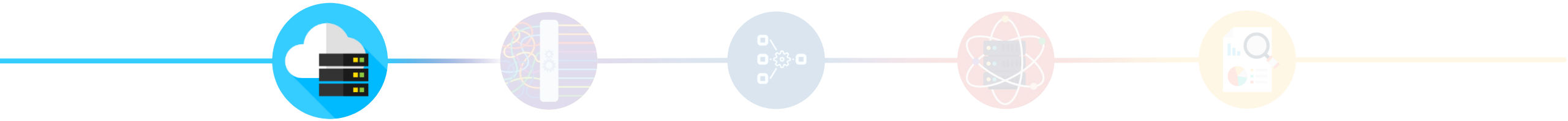
Crankshaft drives three plungers

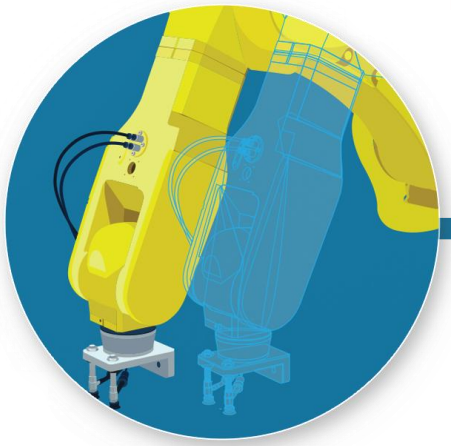
➔ Three types of **failures**

Process Engineer



Access/Generate data





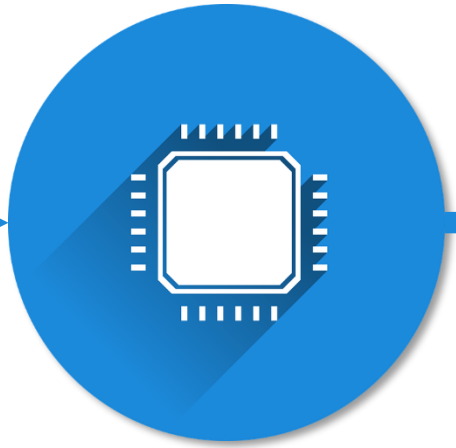
Digital Twin

Bearing Friction

Blocking Fault

Leak Area

Data & **Failures**
Simulation



Parallel Computing

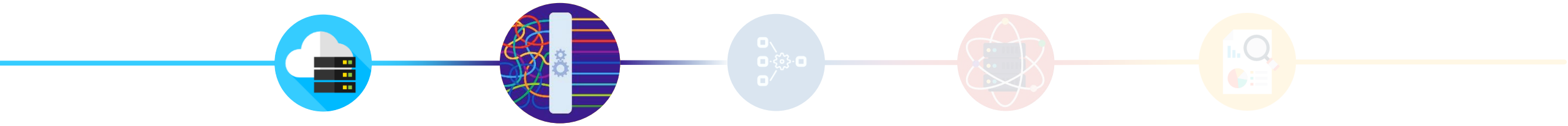
hadoop

Run many
parallel simulations

```
simulationEnsembleDatastore
```

Process Engineer

Preprocessing data

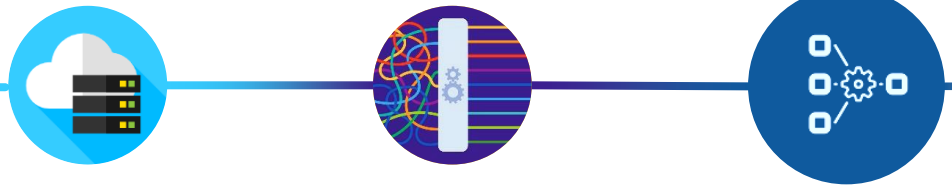


`timetable`

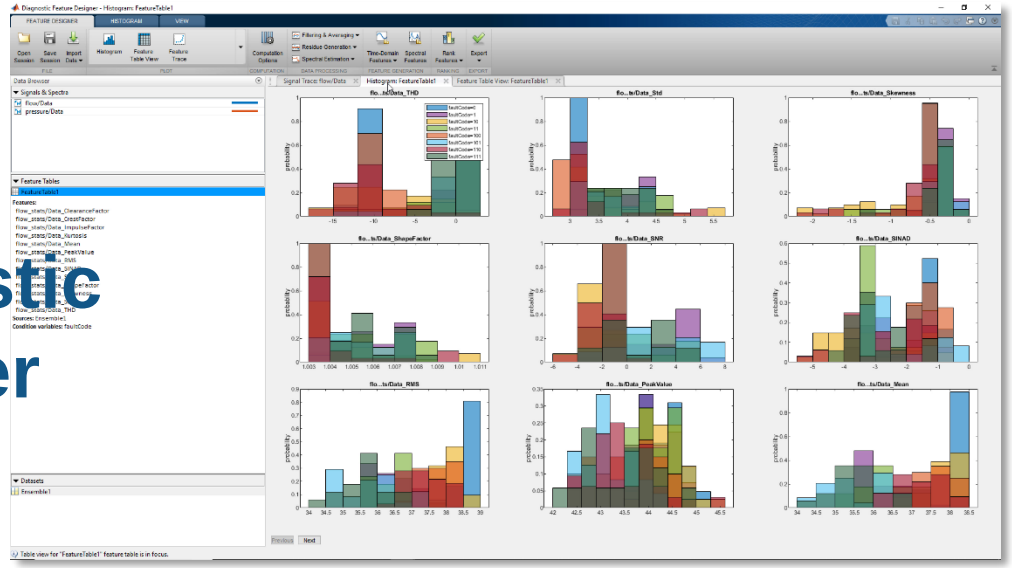
```
data = synchronize(Flow,Pressure,Current,t,'linear');
data = normalize(data,'center');
```

Process Engineer

Identify Condition Indicators



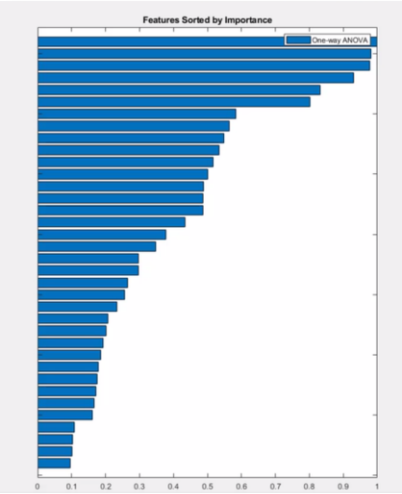
Feature Diagnostic Designer

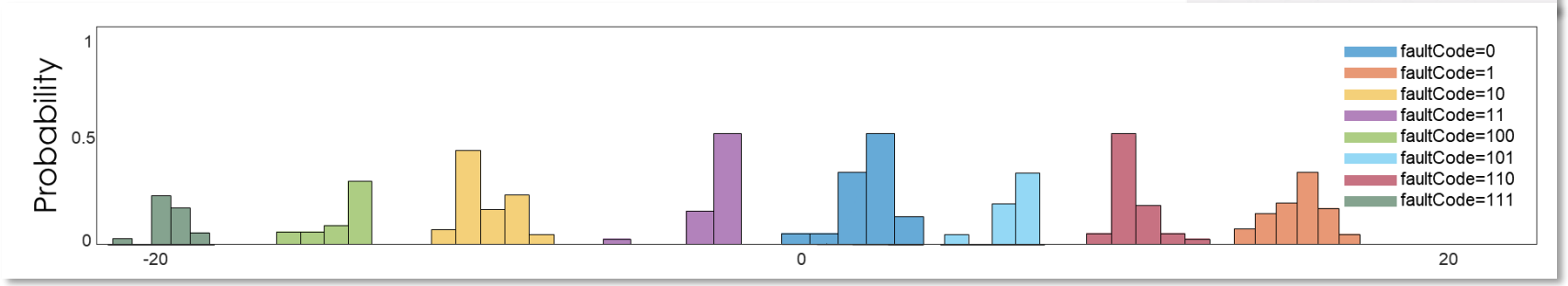


Visualize data

Extract features

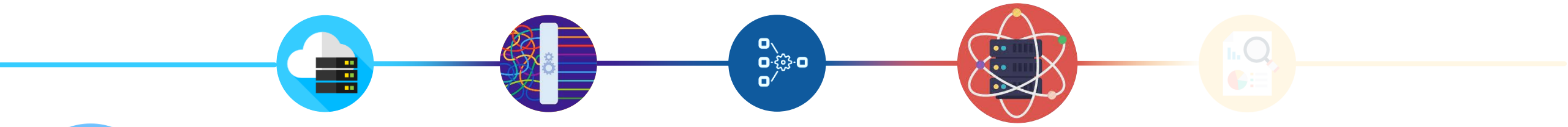
Select the most useful features





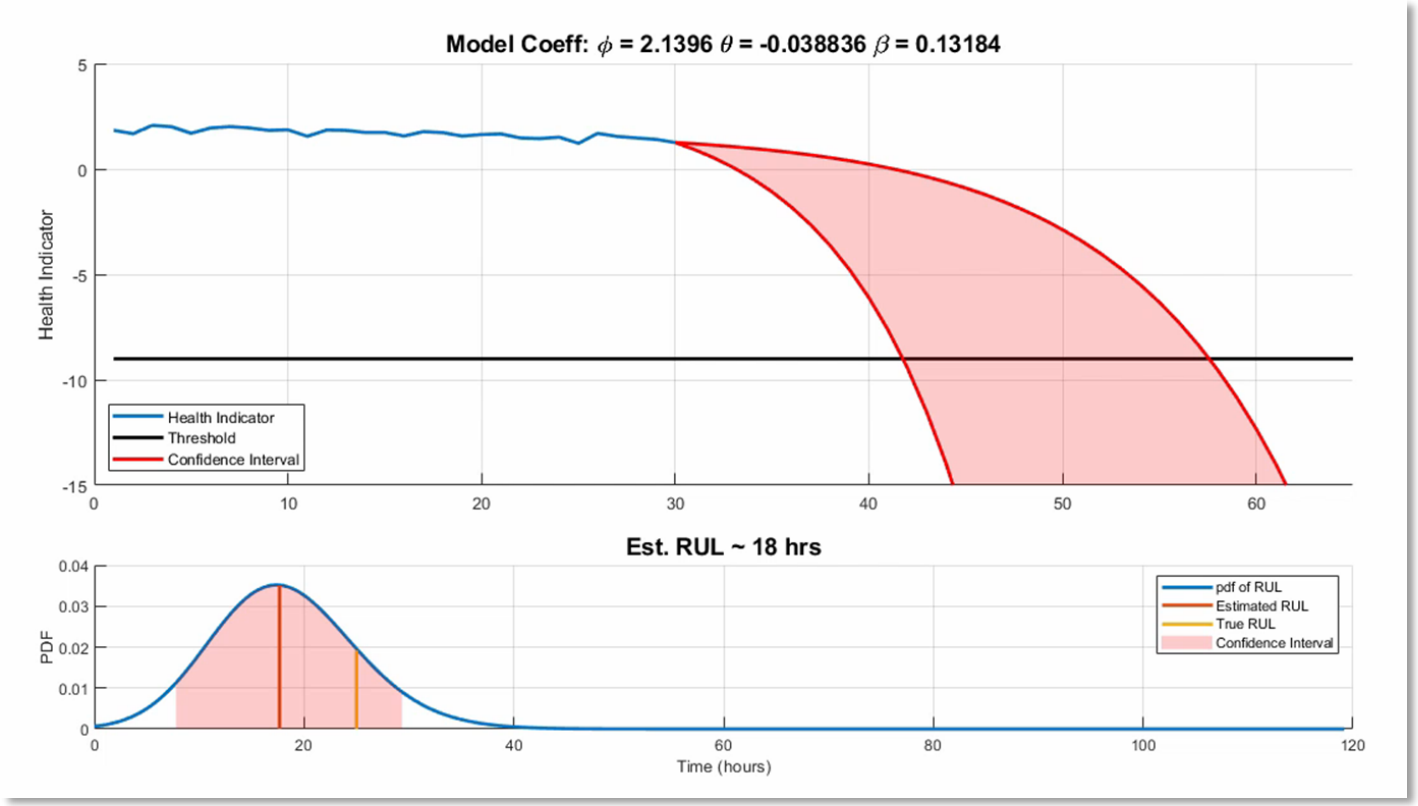
Process Engineer

Predictive Analytics: regression



= *Remaining Useful Life*

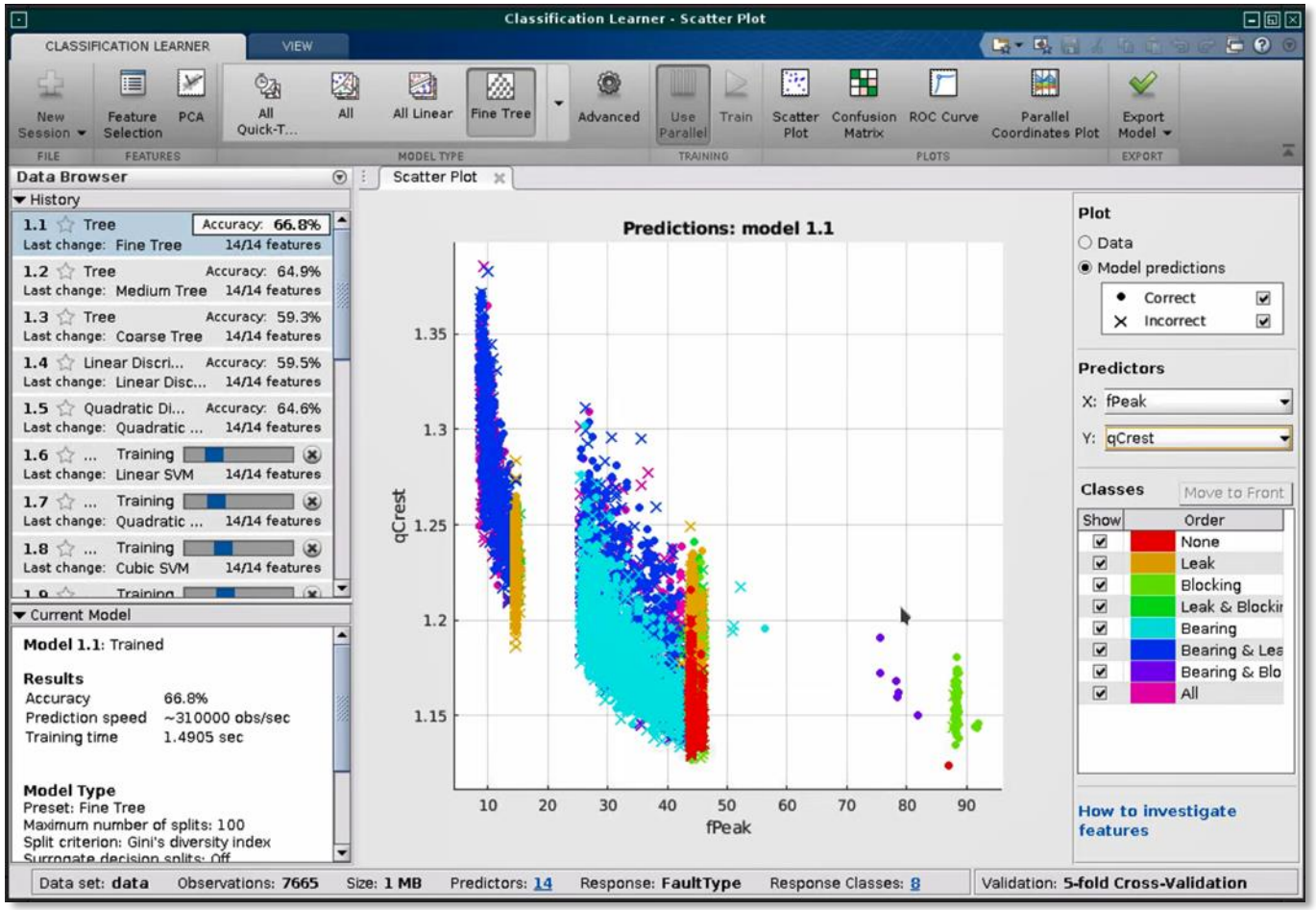
RUL



Process Engineer

Predictive Analytics: classification

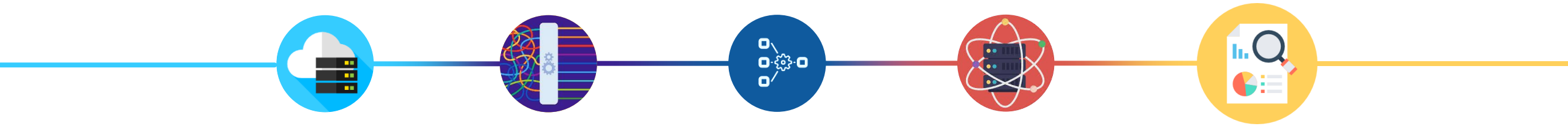
Type of fault



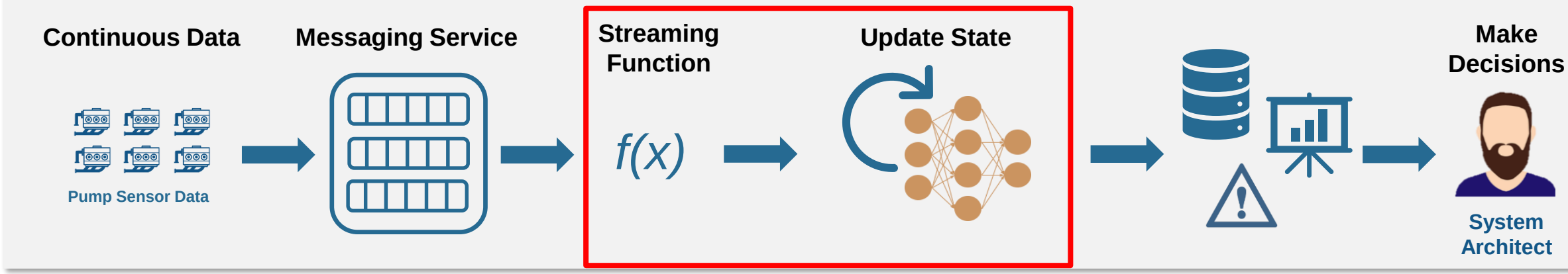
Classification Learner App

Process Engineer

Integrate with Production Systems

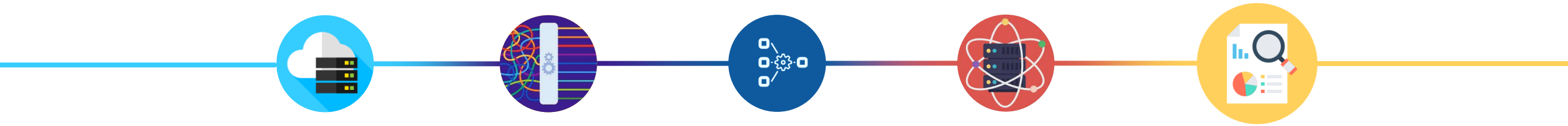


Stream Processing: apply model to sensor data in near real-time



Process Engineer

Develop a streaming function



```
function new_state = streamingFunction(data,old_state)
```

Preprocess signals

```
[data,features] = preprocessData(data);
```

Predict faults

```
[Leak,Blocking,Bearing] = predictFaultValues(features);
FaultType = predictFault(features);
[RUL,Model] = predictUpdateRUL(data.Timestamp,data.Flow,500);
```

Update state

```
new_state = updateState(data,old_state);
```

Write results

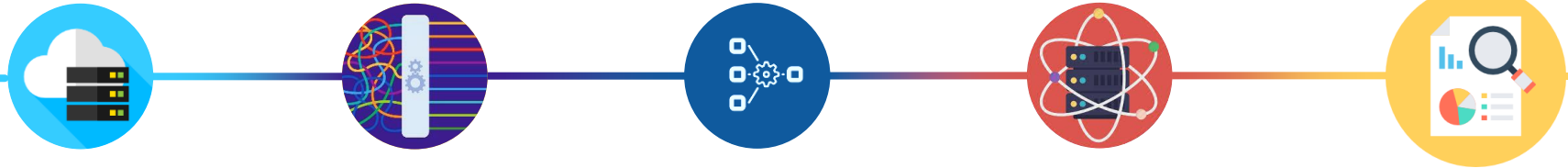
```
writeResults(Leak,Blocking,Bearing,FaultType,RUL,Model)
end
```

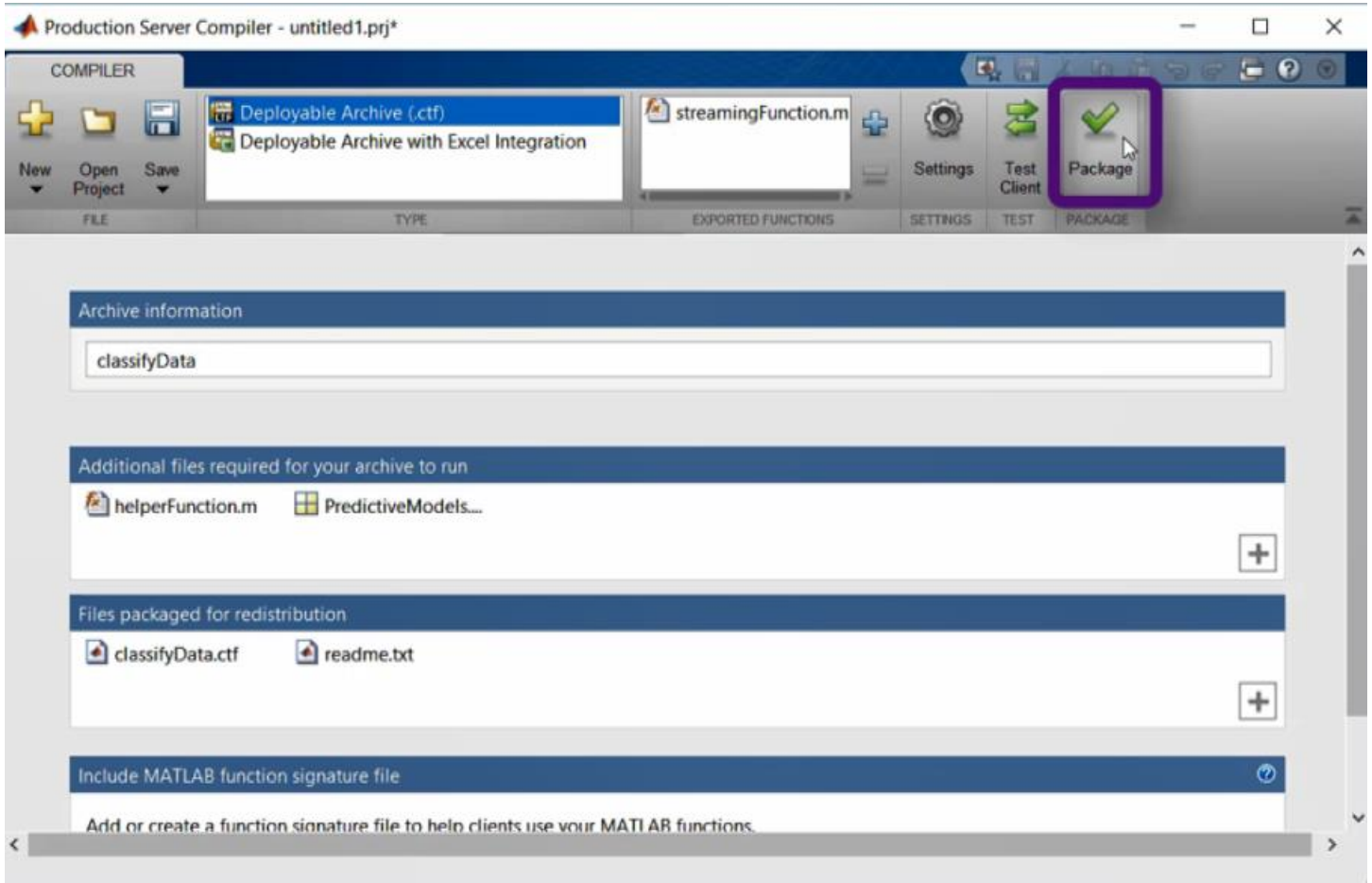
Process each window of data as it arrives

Previous state

Current window of data to be processed

Package Stream Processing Function easily





Review System Requirements

Alerts

Type of fault

Operator

High frequency

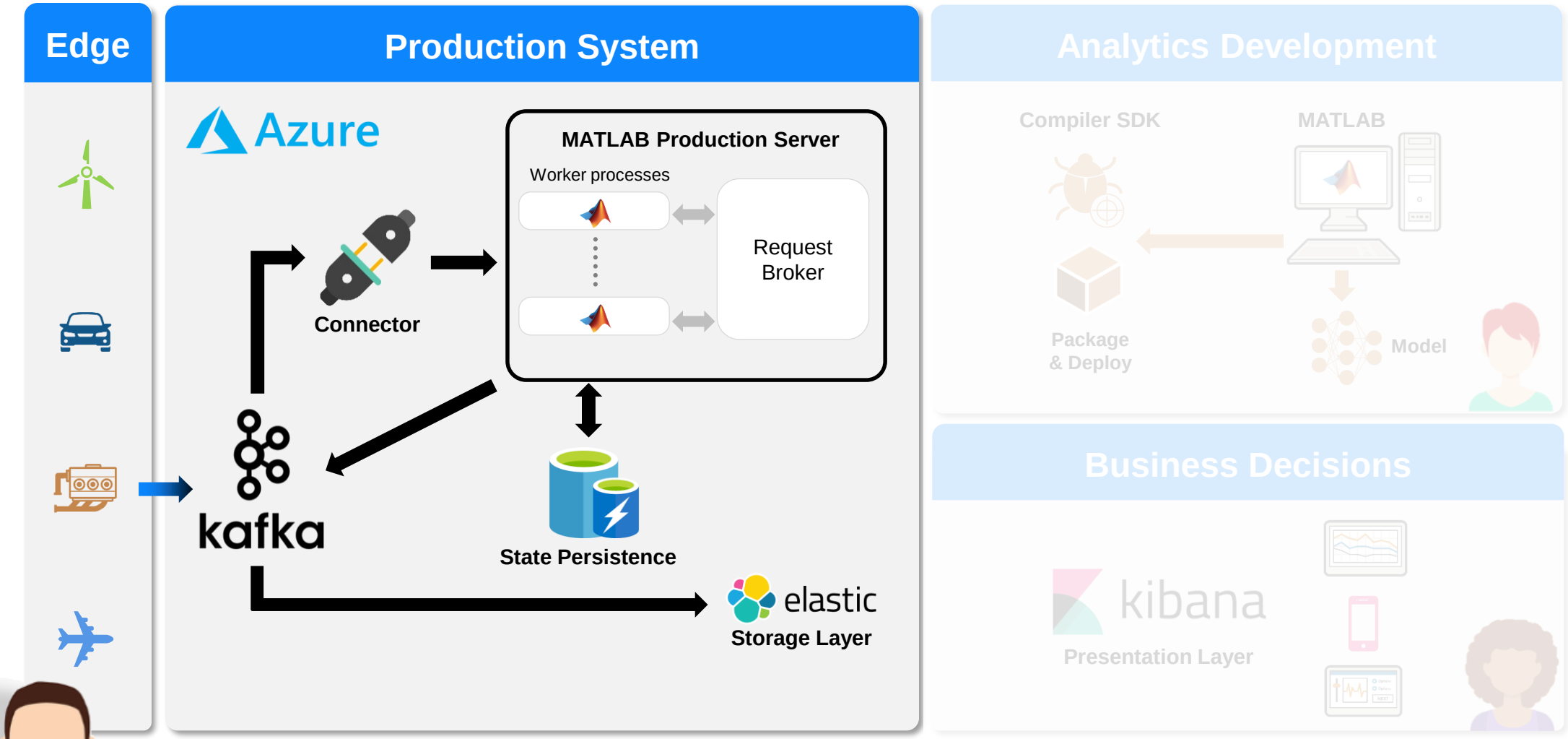
Big Data

Scalablility

Engineer

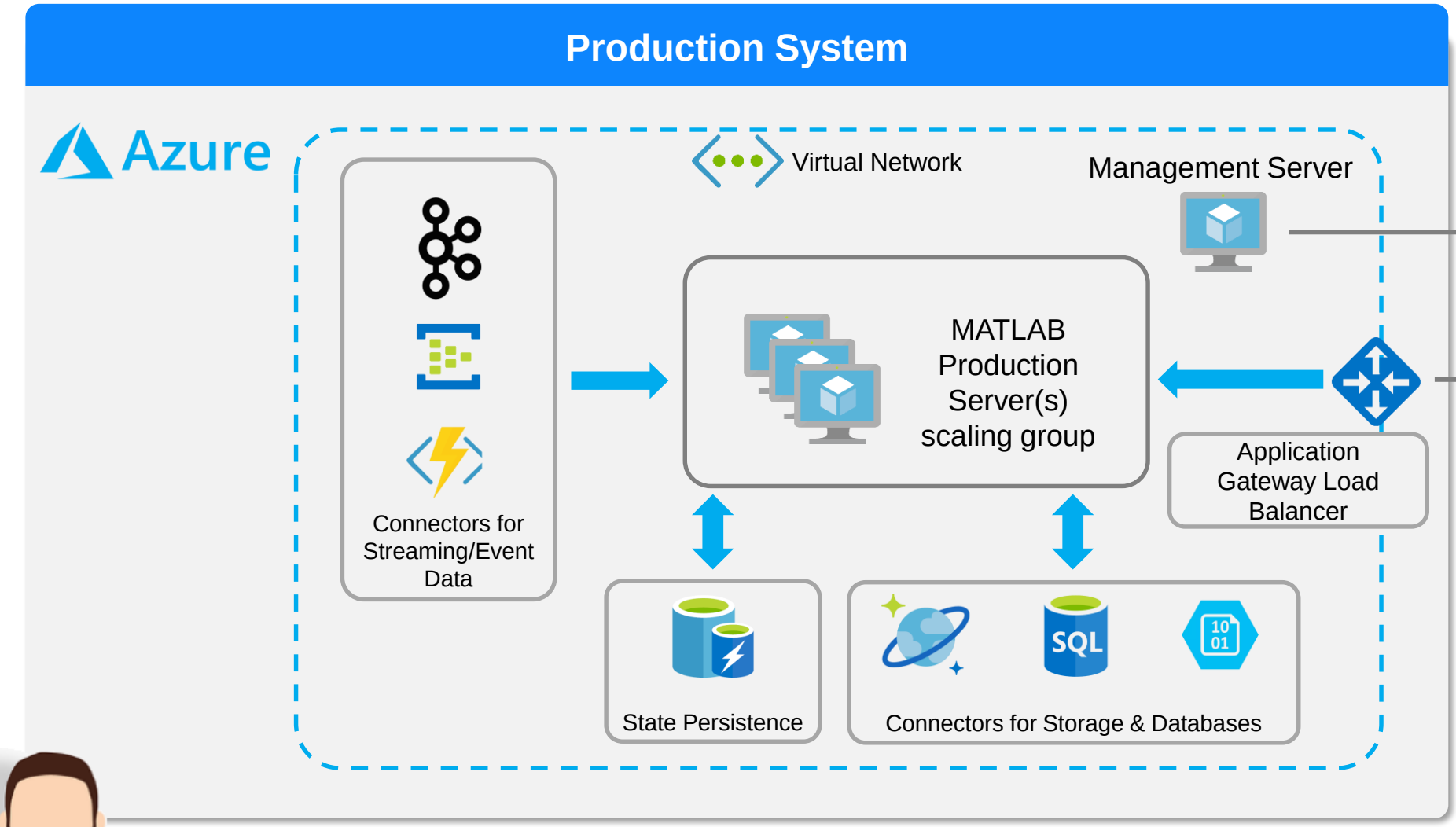
System Architect

Integrate Analytics with Production Systems



System Architect

Configure MATLAB Production Server in the cloud



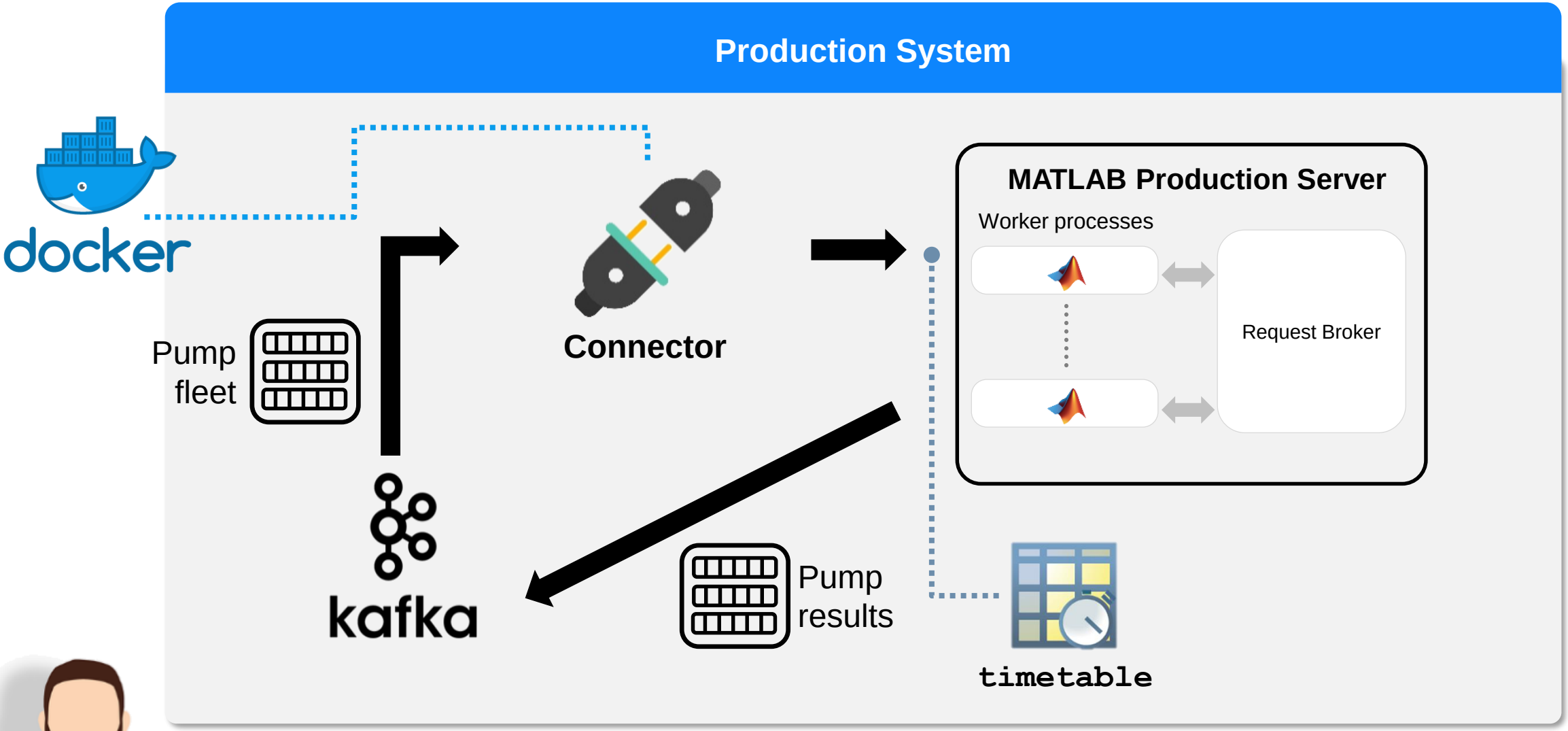
https management endpoint

Enterprise Applications

System Architect

https://github.com/mathworks-ref-arch

Zoom on Kafka connector to MPS

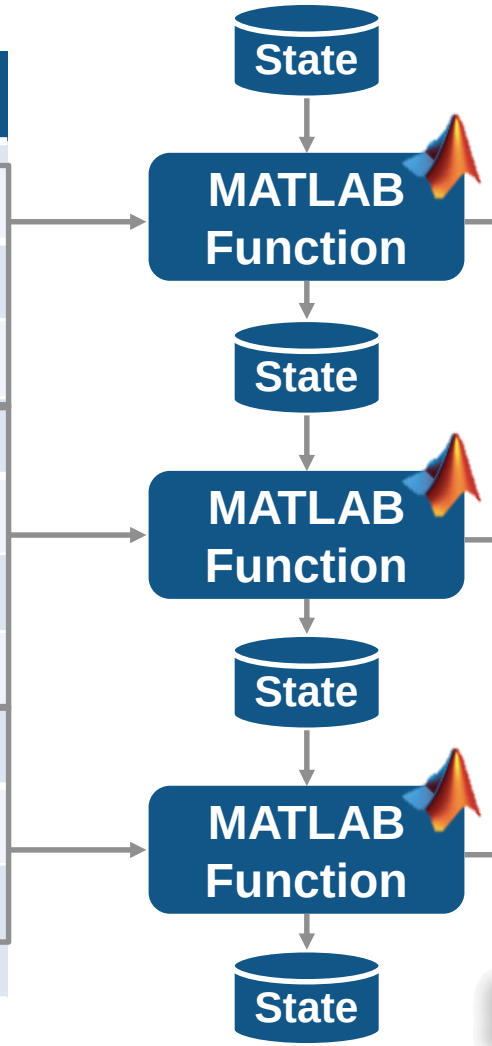


System Architect

Streaming data is treated as an unbounded Timetable

Input Stream

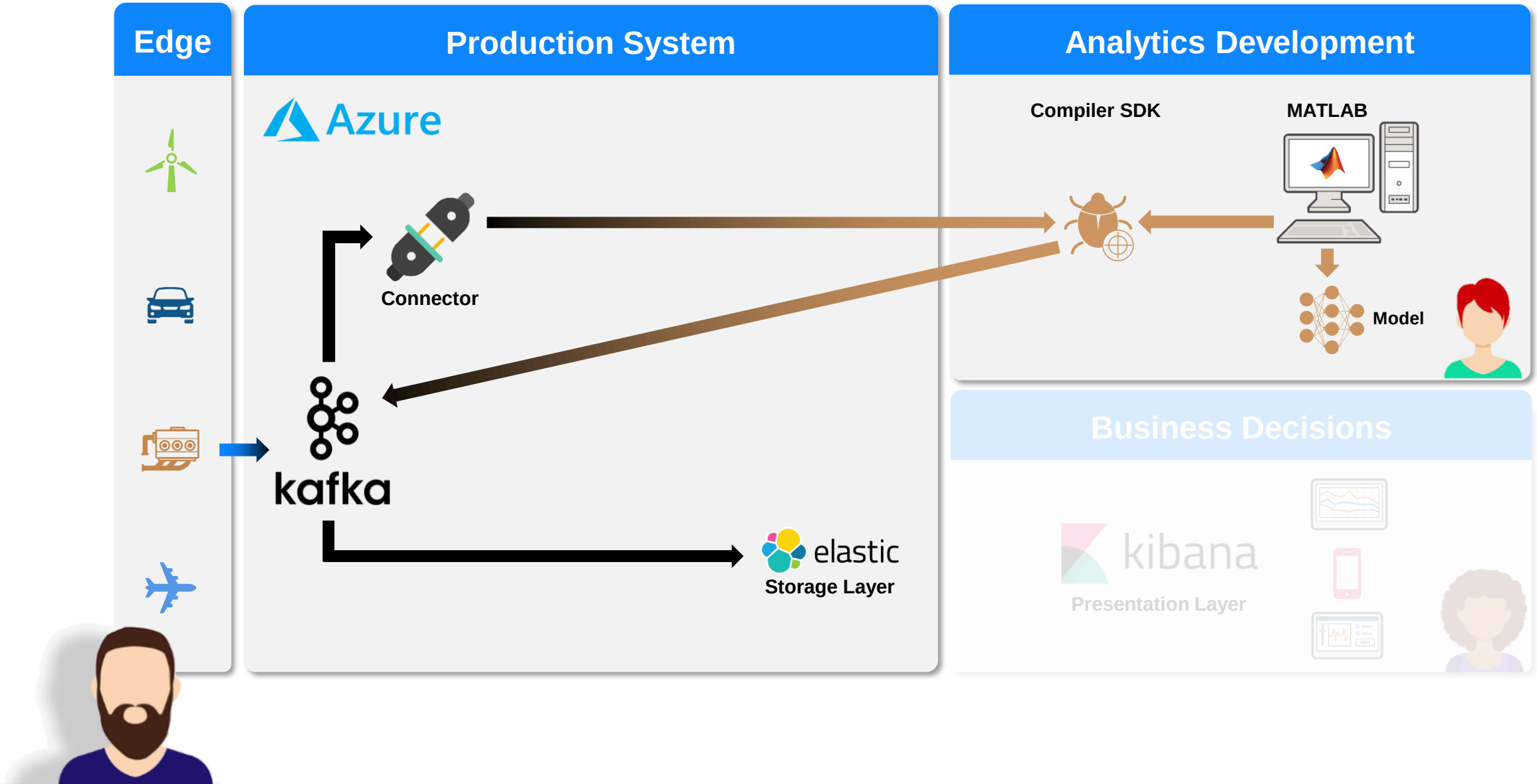
Event Time	Pump Id	Flow	Pressure	Current
18:01:10	Pump1	1975	100	110
18:10:30	Pump3	2000	109	115
18:05:20	Pump1	1980	105	105
18:10:45	Pump2	2100	110	100
18:30:10	Pump4	2000	100	110
18:35:20	Pump4	1960	103	105
18:20:40	Pump3	1970	112	104
18:39:30	Pump4	2100	105	110
18:30:00	Pump3	1980	110	113
18:30:50	Pump3	2000	100	110
…	…	…	…	…



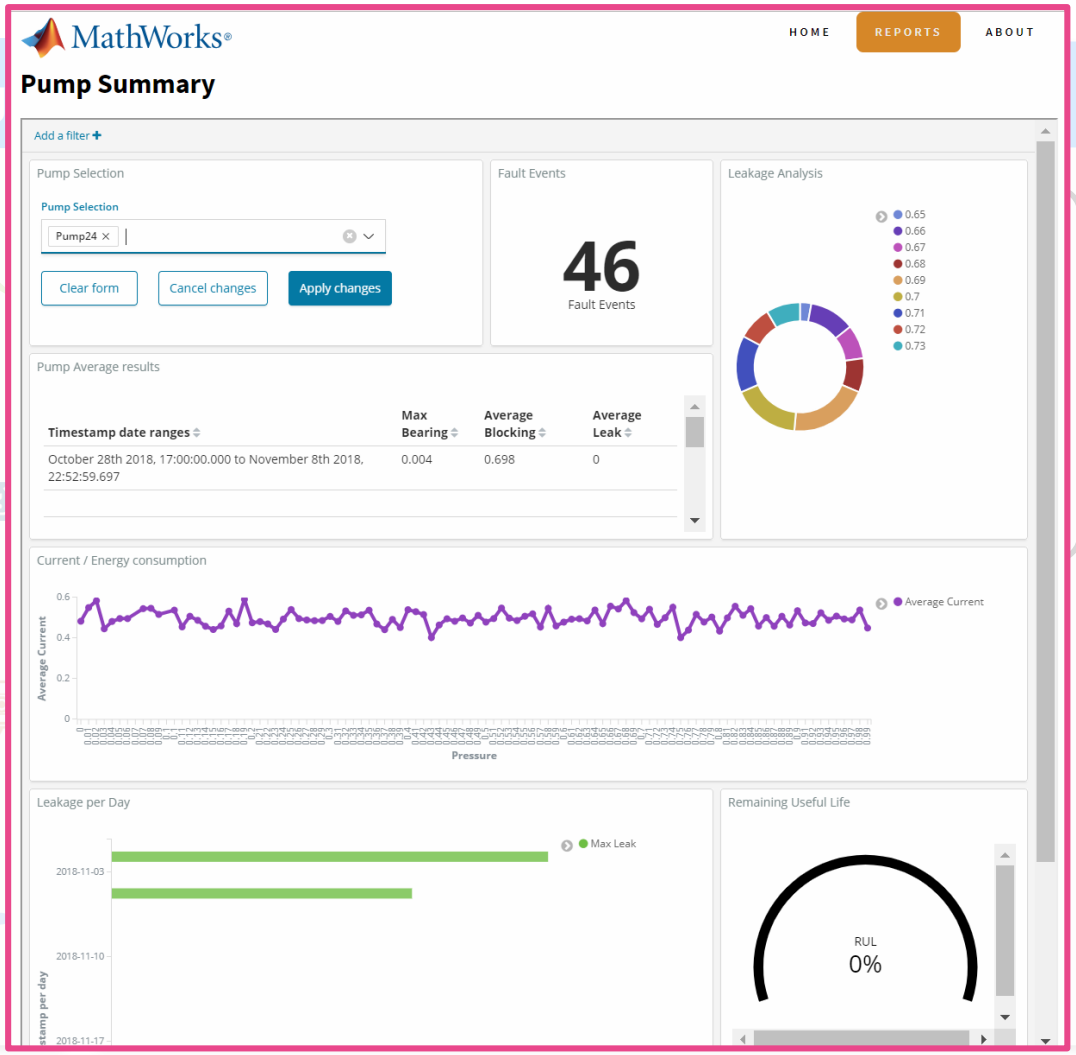
Output Stream

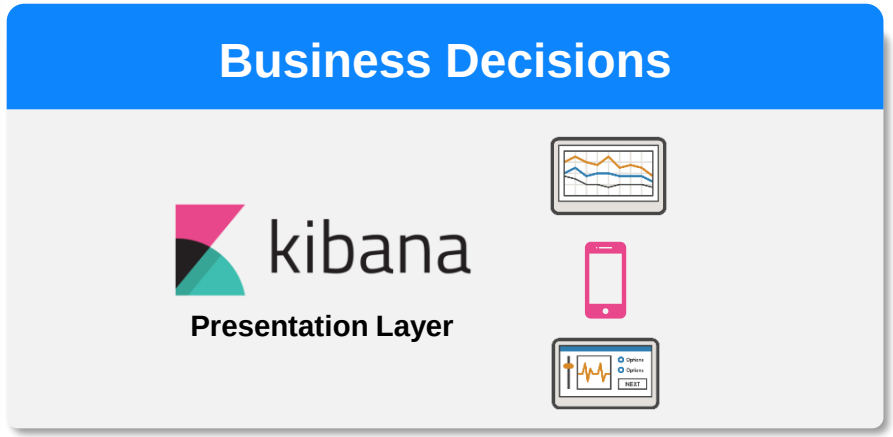
Time window		Pump Id	Bearing Friction
…		…	…
18:00:00	18:10:00	Pump1	5
		Pump3	…
		Pump4	…
18:10:00	18:20:00	Pump2	7
		Pump3	3
		Pump4	…
18:20:00	18:30:00	Pump1	…
		Pump3	4
		Pump4	…
18:30:00	18:40:00	Pump5	…
		Pump3	5
		Pump4	9

Debug your streaming function on live data

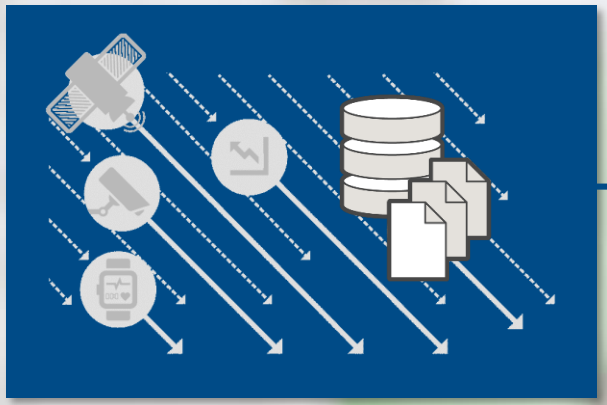


Complete your application





Operator



Access Data

Build Machine Learning models

MATLAB®

Deployment

MathWorks

Baker Hughes Develops Predictive Maintenance Software for Gas and Oil Extraction Equipment Using Data Analytics and Machine Learning

By Gulshan Singh, Engineer Manager

Challenge:

Reduce pump equipment costs & downtime

Solution:

Use MATLAB to analyze 1 TB of data and create a neural network to predict machine failures

Results:

Savings of more than $10 million projected

Development time reduced **tenfold**

Ease of use
Multiple types of data

> *We saw **three advantages** in using MATLAB […]. The first is **speed**; development in C or other language would have taken longer. The second is **automation**. The third is the wide **variety** of technologies.*

Follow this link to read the complete user story of Gulshan Singh

BAKER HUGHES a GE company