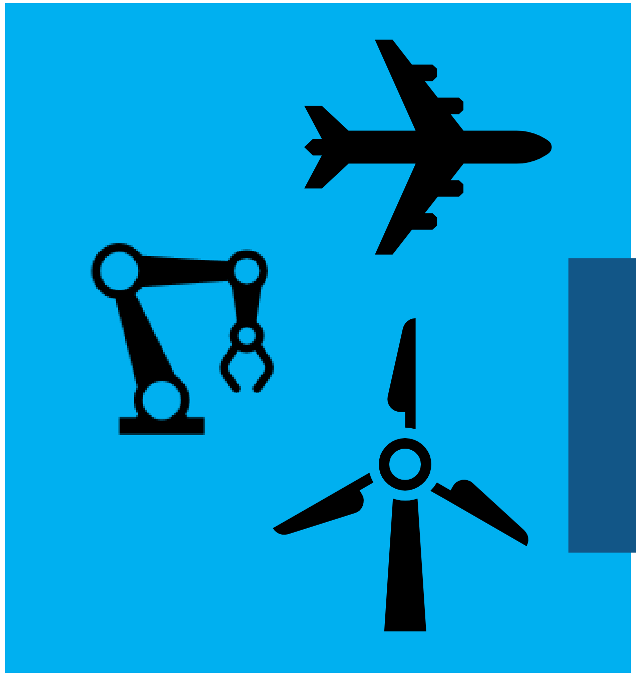
MATLAB EXPO 2019

Deploying AI

for Near Real-Time Decisions

Branko Dijkstra

The Need for Large-Scale Streaming

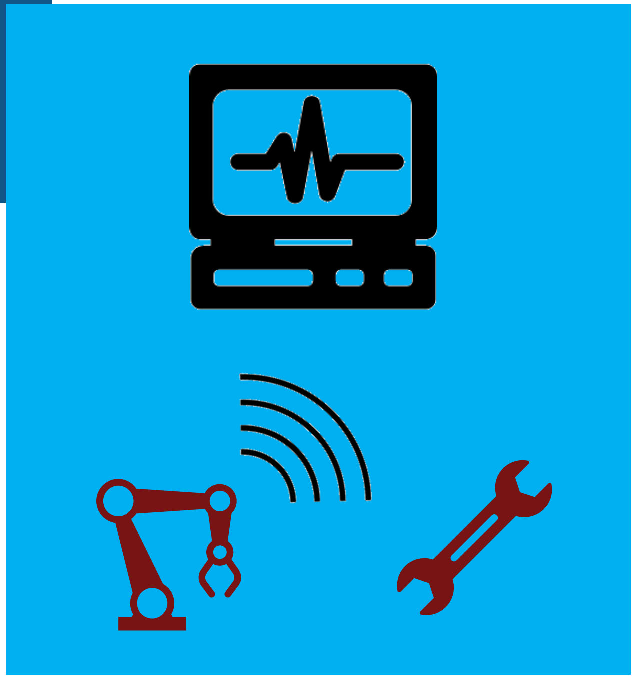


Predictive Maintenance

Increase Operational Efficiency

Reduce Unplanned Downtime

More applications require near real-time analytics



Medical Devices

Patient Safety

Better Treatment Outcomes

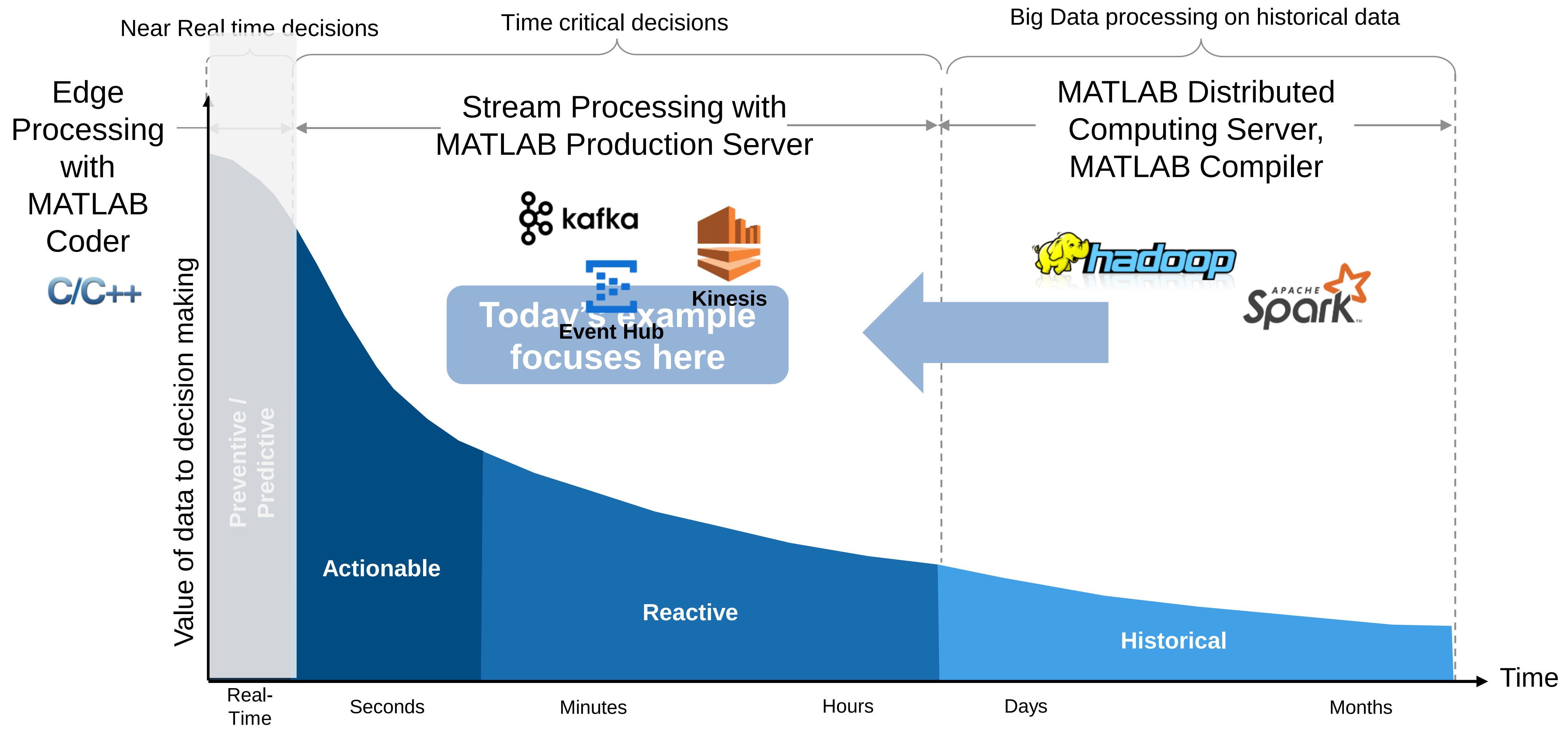
Manufacture/Processing

Process Input Variation

Maintenance Planning

Jet engine: ~800TB per day
Turbine: ~ 2 TB per day
Crusher: ~10 Mb per day
Washing Machine: ~10kb/day

Why stream processing?



Our Project: Develop and operationalize a machine learning model to predict failures in industrial pumps

Process Engineer

Develops models in MATLAB and Simulink

System Architect

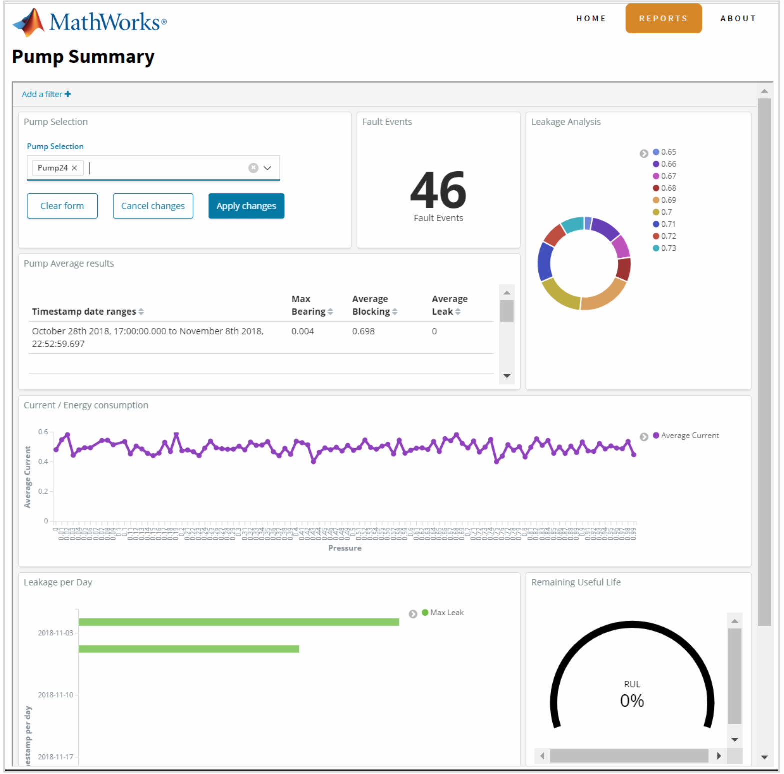
Deploys and operationalizes model on Azure cloud

Azure

Operator

Makes operational decisions based on model output

kibana



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


5"
12,000 PSI
5"
12,000 PSI
BAKER HUGHES

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

Plant Operator

1. Monitor ***flow***, ***pressure***, and ***current*** of each pump so I always know their ***operational state***
2. Need ***alert*** when fault parameters drift outside an acceptable range so I can take ***immediate corrective action***
3. Continuous estimate of each pump’s ***remaining useful life (RUL)*** so I can ***schedule maintenance or replace*** the asset

Challenges of AI Deployment

Process Engineer

We don't have a large set of failure data, and it's too costly to generate real failures in our plant for this project

Solution: Use an accurate physics-based software model for the pump to develop synthetic training sets

MathWorks

Challenges of AI Deployment

System Architect

We don’t have a large IT/hardware budget, and we need to see results before committing to a particular platform or technology

Solution: Leverage cloud platform to quickly configure and provision the services needed to build the solution, while minimizing lock-in to a particular provider

Challenges of AI Deployment

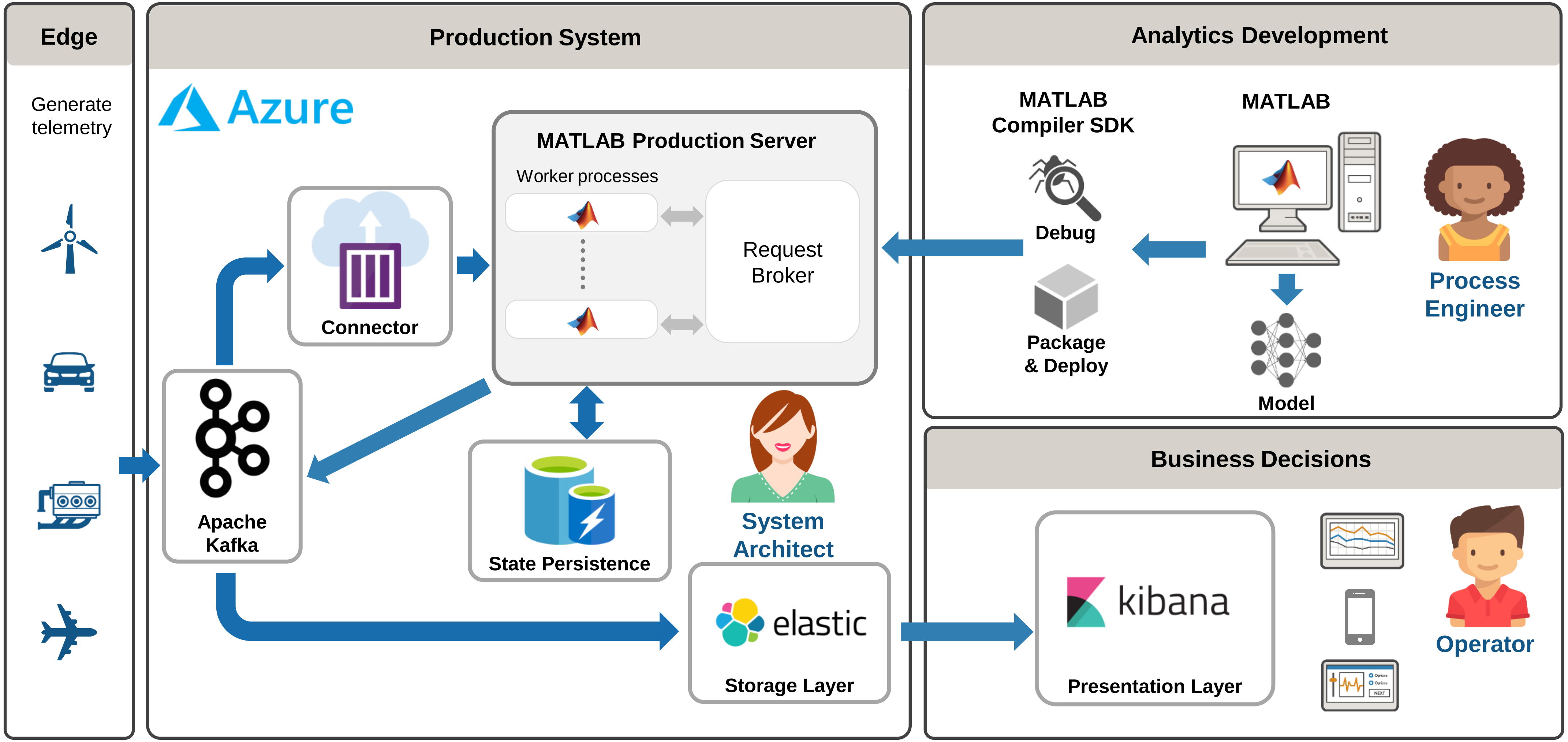
Process Engineer

Need software for multidisciplinary problem across teams, plus integration with IT

Solution: Use MATLAB and integrate with Open Source Software

MathWorks

Predictive Maintenance Architecture on Azure



Modeling approach

Process Engineer

1 Access and Explore Data

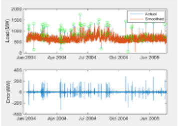
Files

Databases

Sensors

2 Preprocess Data

Working with Messy Data



Data Reduction/ Transformation

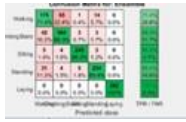
Feature Extraction

3 Develop Predictive Models

Model Creation e.g. Machine Learning

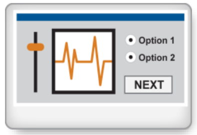
Parameter Optimization

Model Validation



4 Integrate with Production Systems

Desktop Apps



Enterprise Scale Systems

kafka

AWS Kinesis

OSIsoft.

Embedded Devices and Hardware

5 Visualize Results

3rd party dashboards

Spotfire® TIBCO Software

tableau

Qlik

Power BI

Web apps

Process Engineer

Review model requirements

Requirements From Operator

- Continuous predictions of type of fault
  - “Blocking”
  - “Leaking”
  - “Bearing”
  - Combination of above
- Continuous predictions of Remaining Useful Life [RUL]

Requirements From System Architect

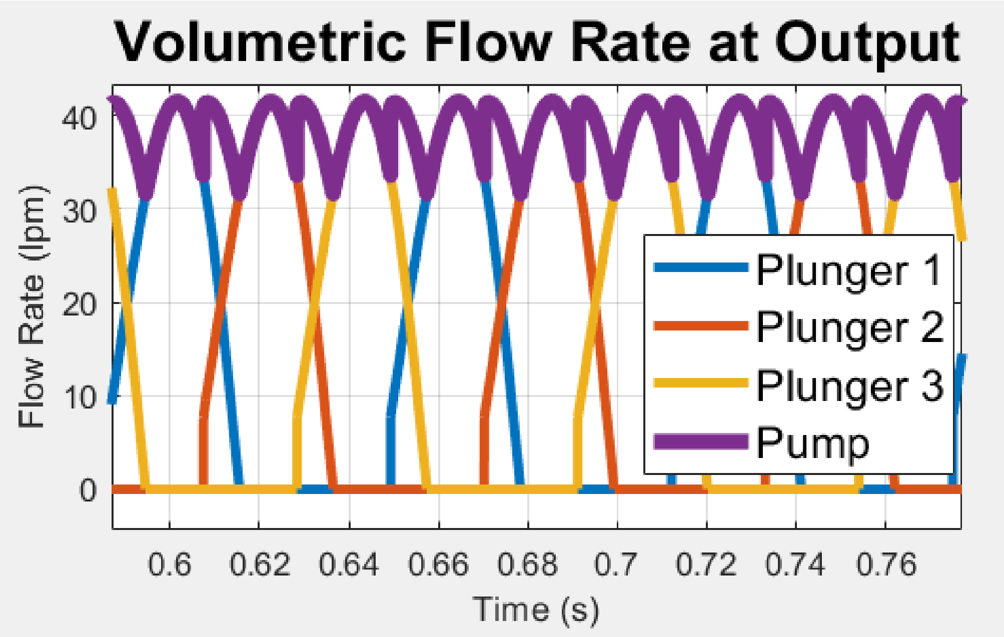
- Define window for streaming
- Define format of results, intermediate values
- Test code
- Scale code

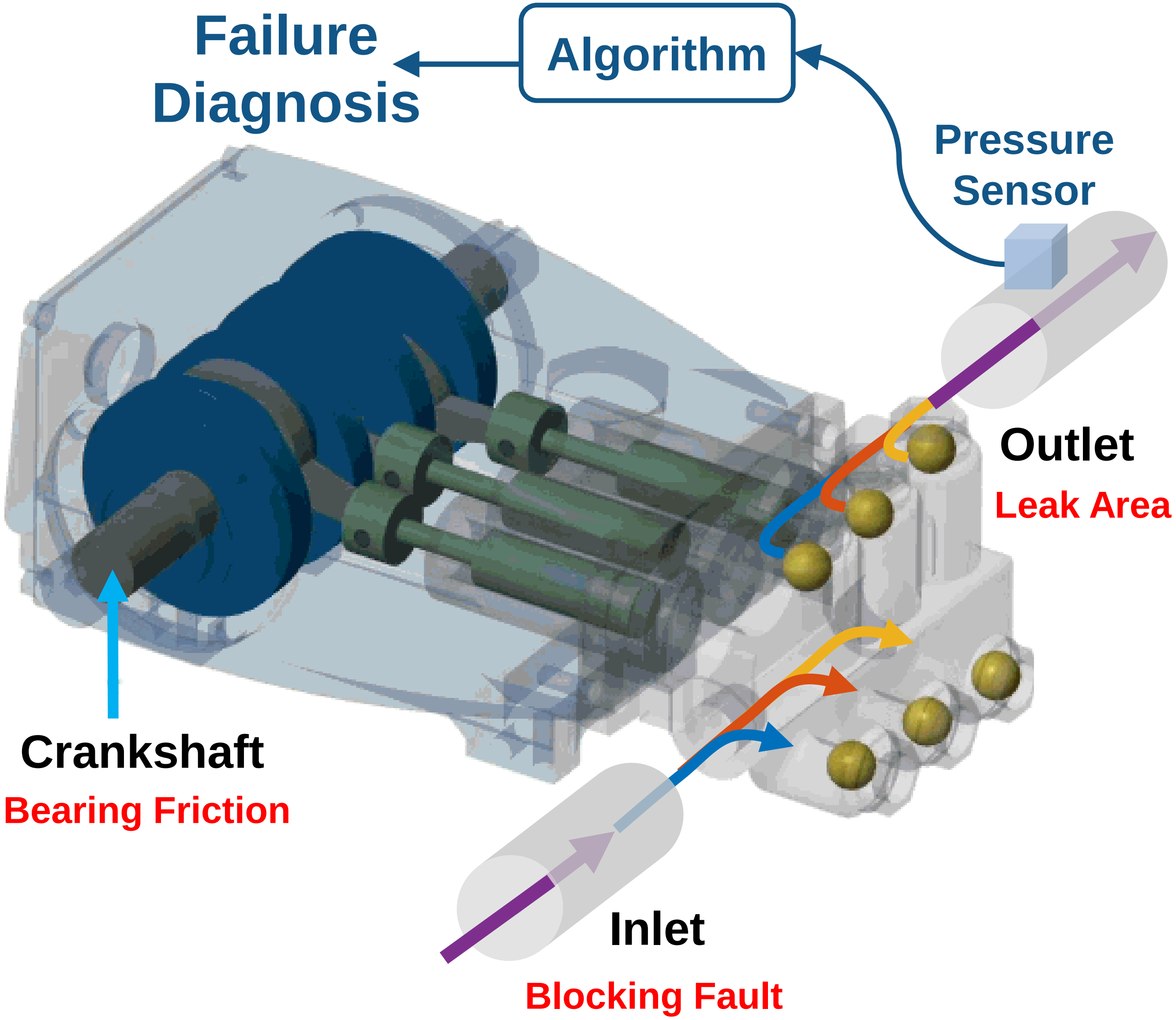

5" 12,000 PSI
5" 12,000 PSI
BAKER HUGHES

Process Engineer

Physics of Triplex Pump

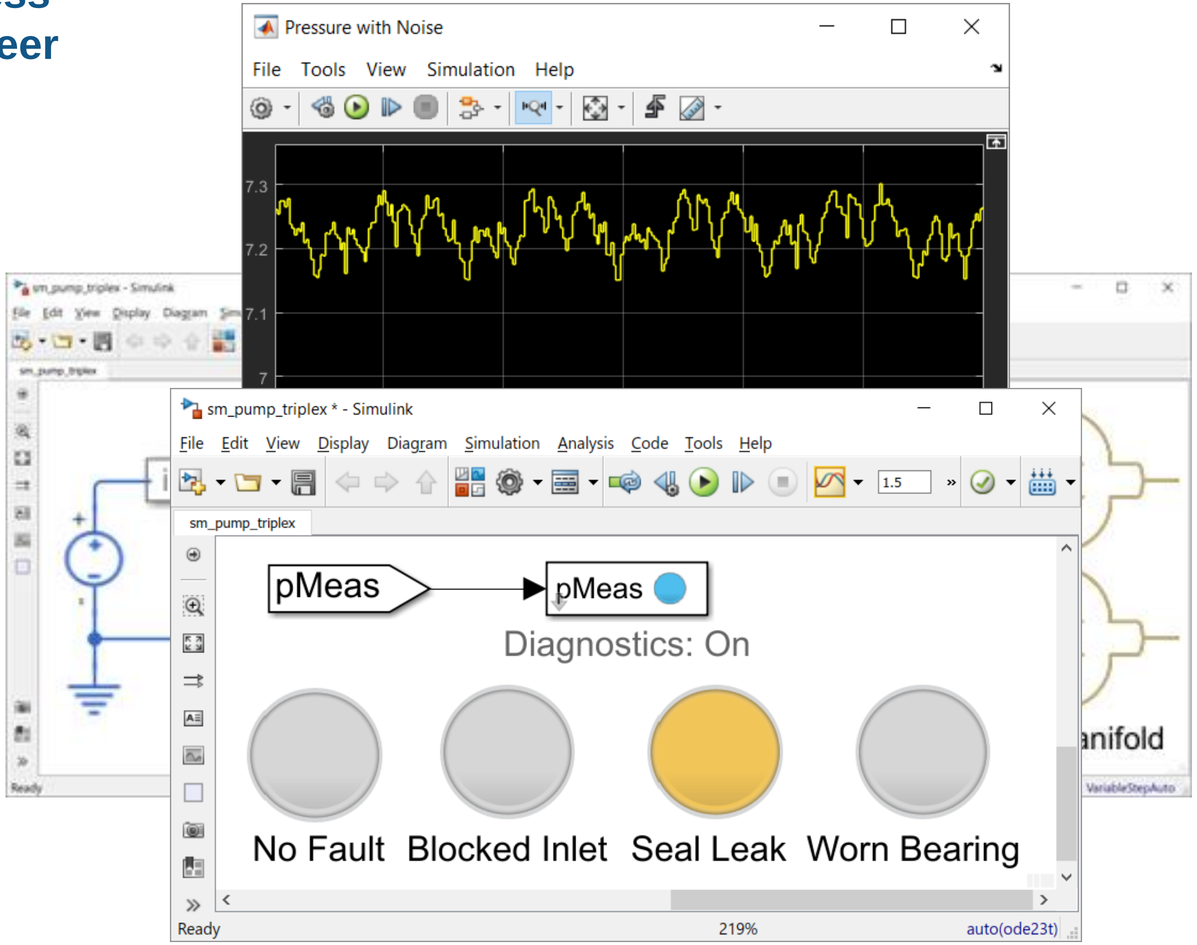
- Crankshaft drives three plungers
  - Each 120 degrees out of phase
  - One chamber always discharging
  - Three types of **failures**

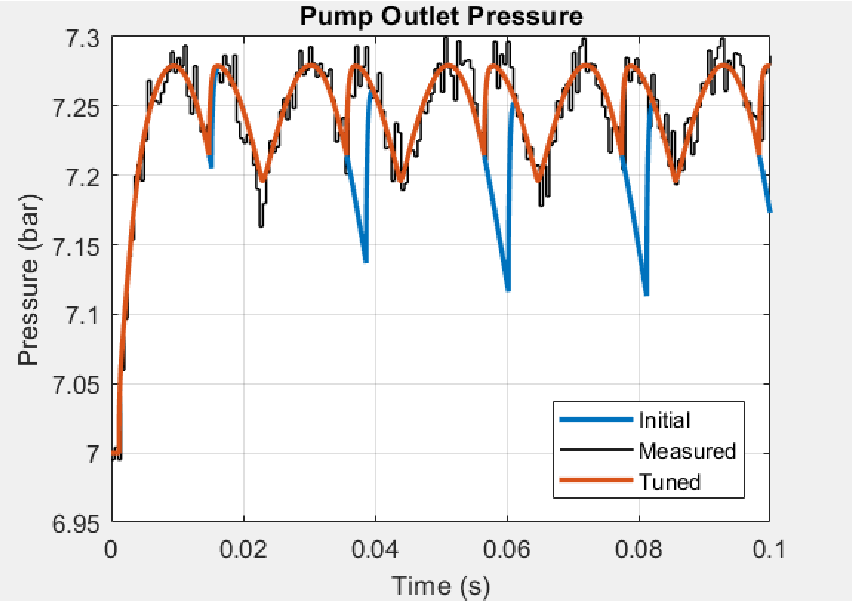




1 Access and Explore Data

Use sensor data from pump to identify levels of failure





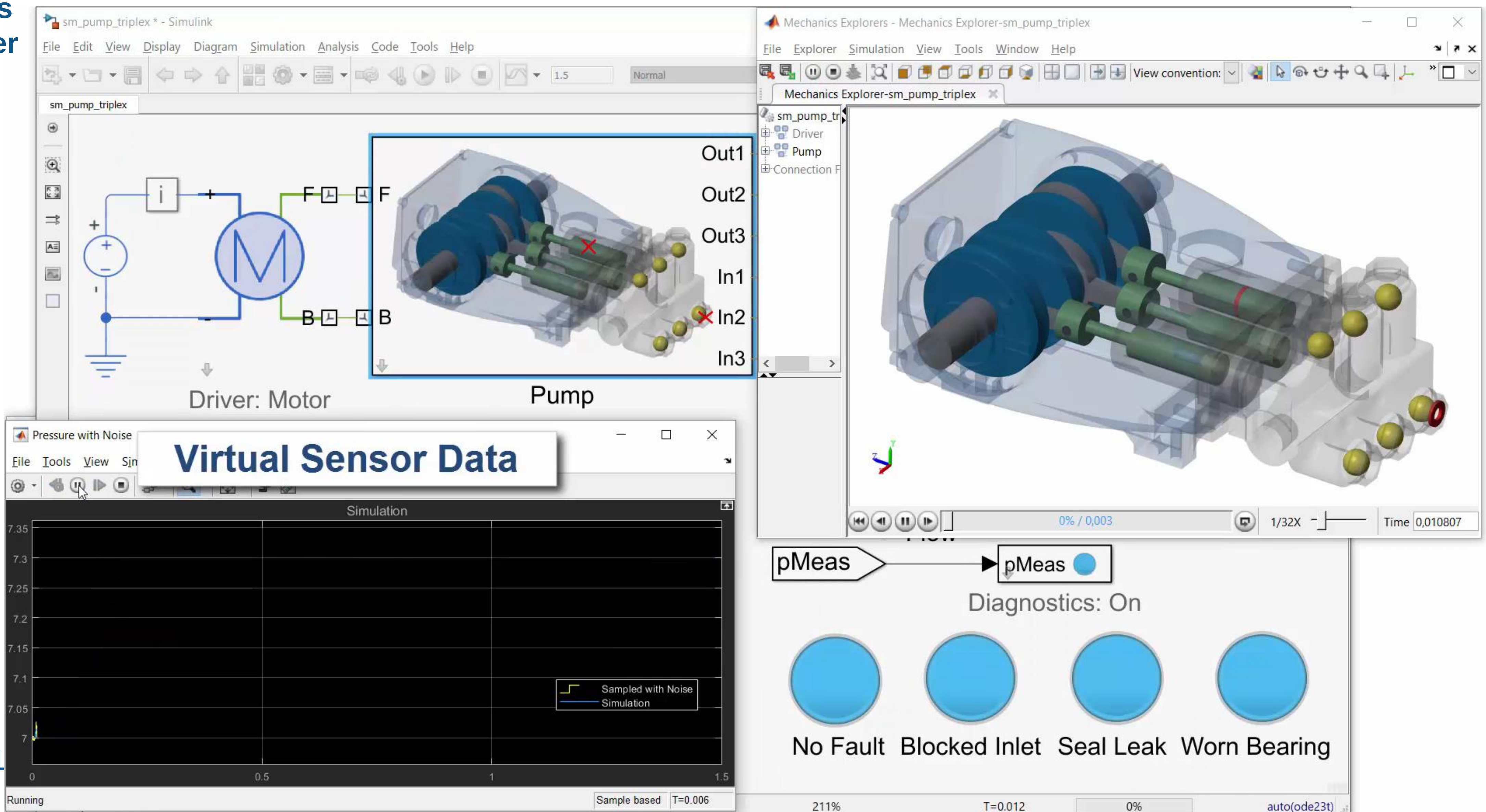
Simulate faults

Pump sensor data

1 Access and Explore Data

Build digital twin and generate sensor data

Process Engineer

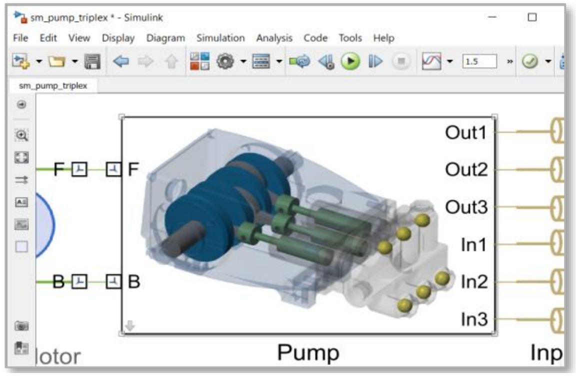


1

Access and Explore Data

Simulate data with many failure conditions

Process Engineer



Leak Area = [1e-9 0.036]

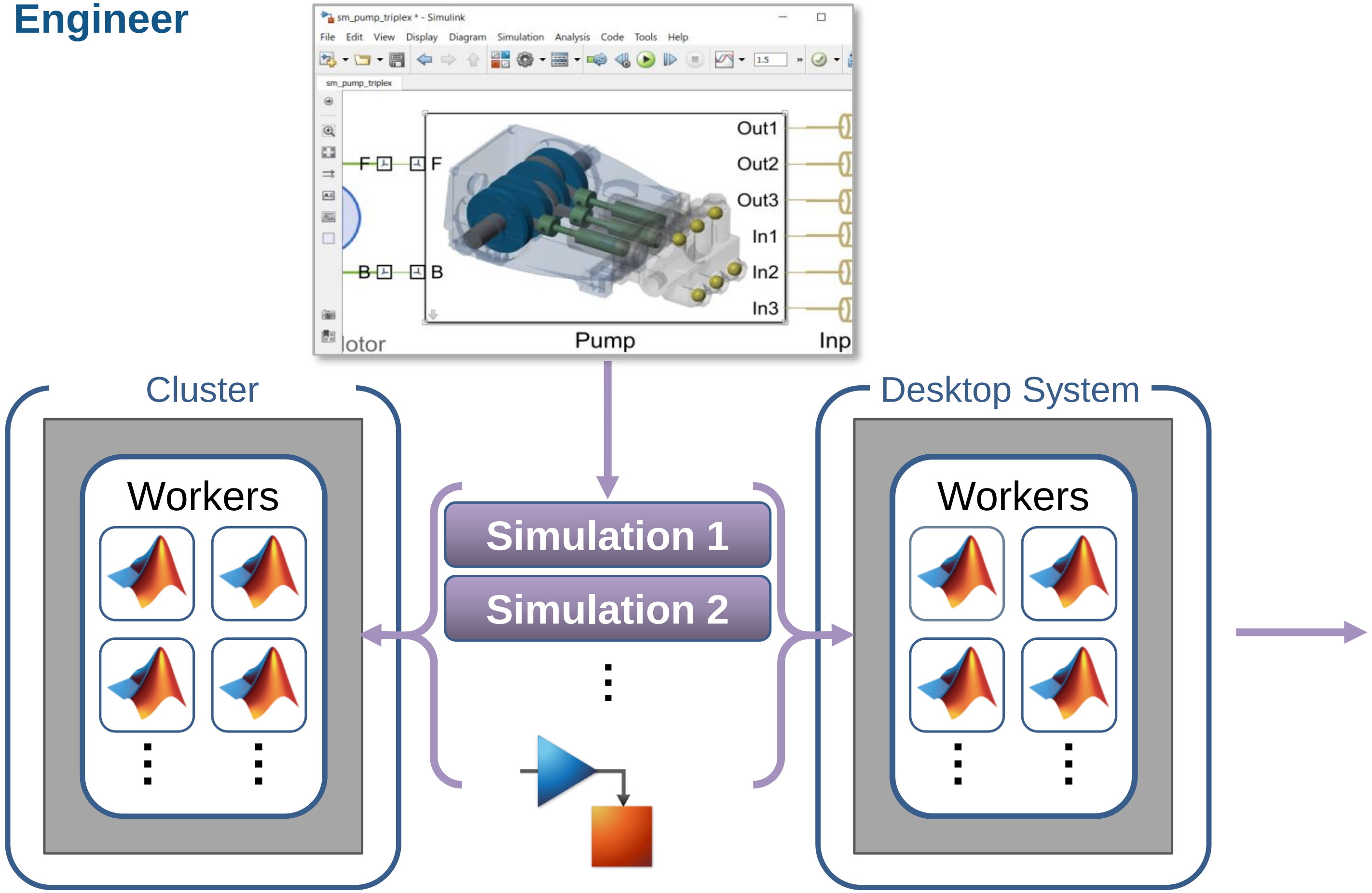
Bearing Friction = [0 6e-4]

Blocking Fault = [0.5 0.8]

Simulate data with many failure conditions

1 Access and Explore Data

Process Engineer



Run parallel simulations

Access Data

```
ens = simulationEnsembleDatastore(location)
```

```
ens =
  simulationEnsembleDatastore with properties:

           DataVariables: [25×1 string]
    IndependentVariables: [0×0 string]
      ConditionVariables: [0×0 string]
       SelectedVariables: [25×1 string]
                ReadSize: 1
              NumMembers: 702
          LastMemberRead: [0×0 string]
                   Files: [702×1 string]
```

Store data on HDFS

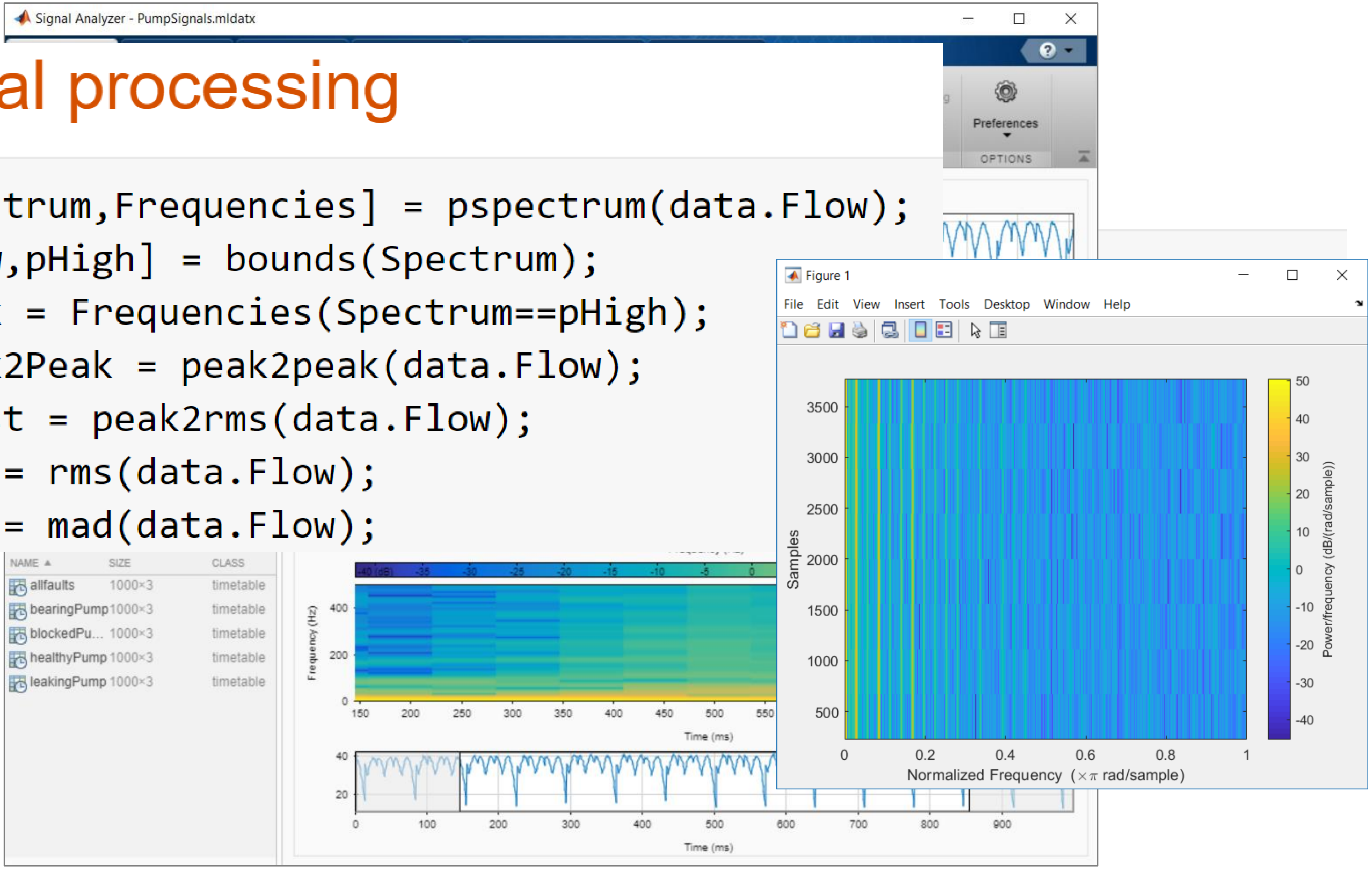
Process Engineer

2 Preprocess Data

Represent signal information

Signal processing

```
[Spectrum,Frequencies] = pspectrum(data.Flow);
[pLow,pHigh] = bounds(Spectrum);
fPeak = Frequencies(Spectrum==pHigh);
qPeak2Peak = peak2peak(data.Flow);
qCrest = peak2rms(data.Flow);
qRMS = rms(data.Flow);
qMAD = mad(data.Flow);
```



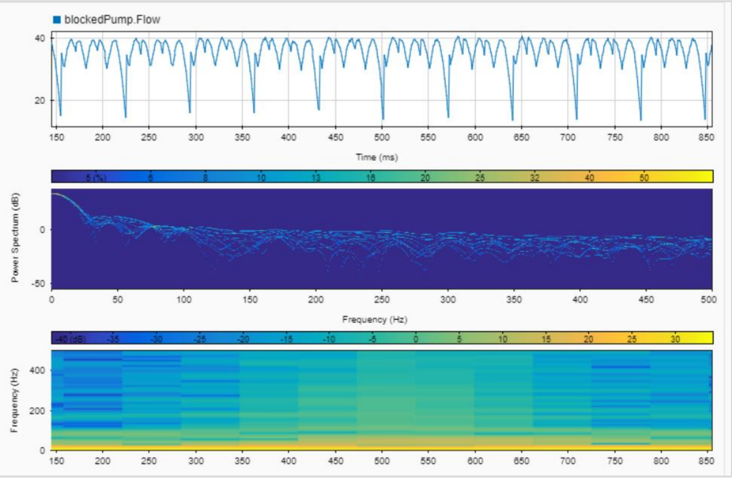
Develop Predictive Models in MATLAB

3 Develop Predictive Models

Process Engineer

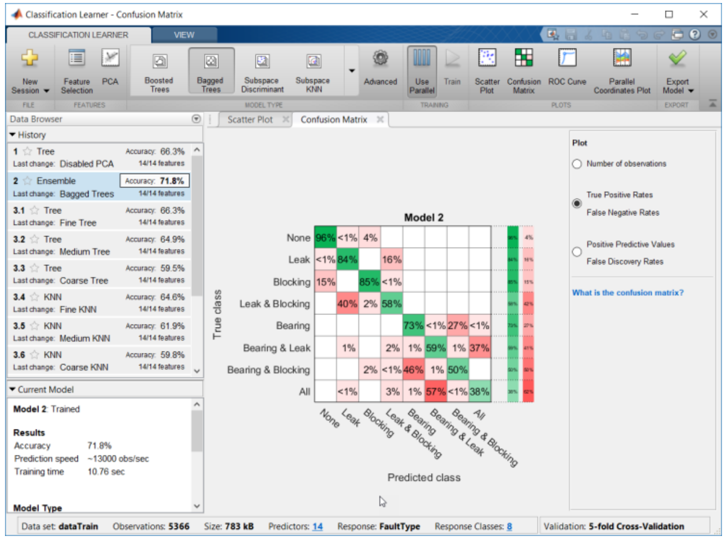
	Time	1 LeakFault	2 BlockingFault	3 BearingFault	4 FaultType
1	0 sec	2.8472	-0.1477	1.8000	All
2	0.001 sec	-0.1498	-0.4207	1.3103	Bearing & Blocking
3	0.002 sec	0.6511	1.6521	-0.5557	Leak
4	0.003 sec	0.1469	-0.2775	1.0074	All
5	0.004 sec	-0.6480	0.7065	-0.8878	Blocking
6	0.005 sec	-0.8165	-0.5434	-0.3079	Blocking
7	0.006 sec	-1.0061	1.2083	0.0661	Bearing
8	0.007 sec	1.0125	-1.9098	-0.7027	Leak & Blocking

Label Faults



Represent Signals

Train Model



Validate Model

Scale

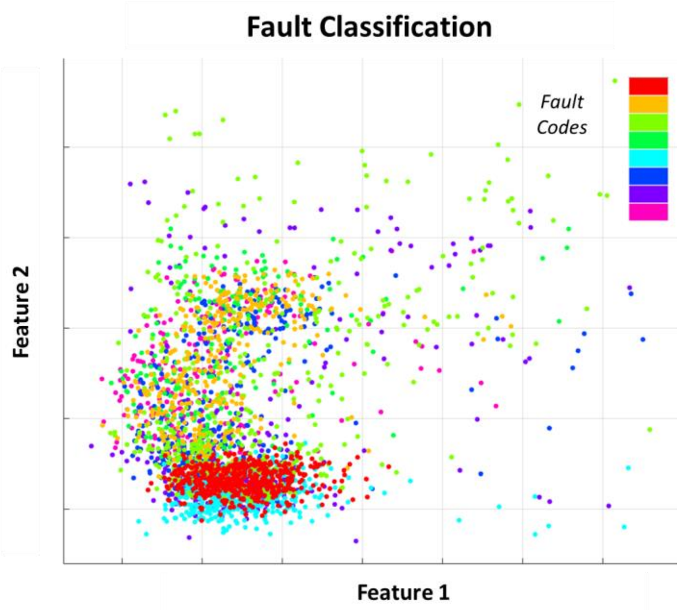
```
tt = tall(ds);
tt = preprocessData(tt);
model = TreeBagger(50,tt,'Event');
```

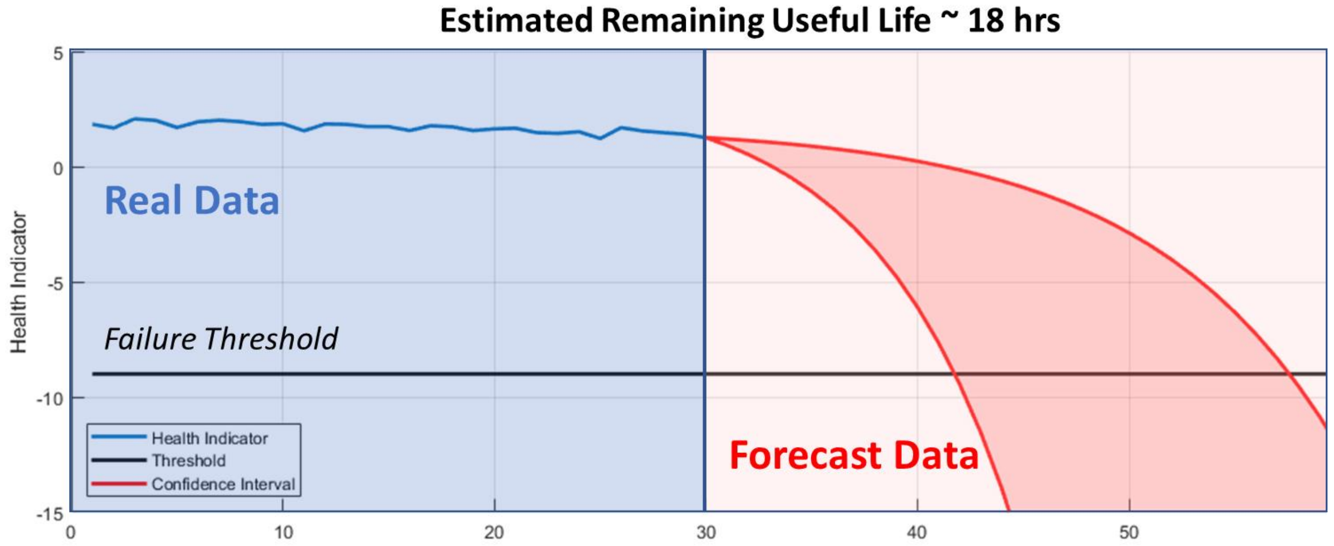
```
Evaluating tall expression using the Spark Cluster:
- Pass 1 of 2: Completed in 11 sec
- Pass 2 of 2: Completed in 2.3333 min
Evaluation completed in 2.6167 min
```

3 Develop Predictive Models

Process Engineer

Develop Predictive Models in MATLAB





Type of Fault (Classification)

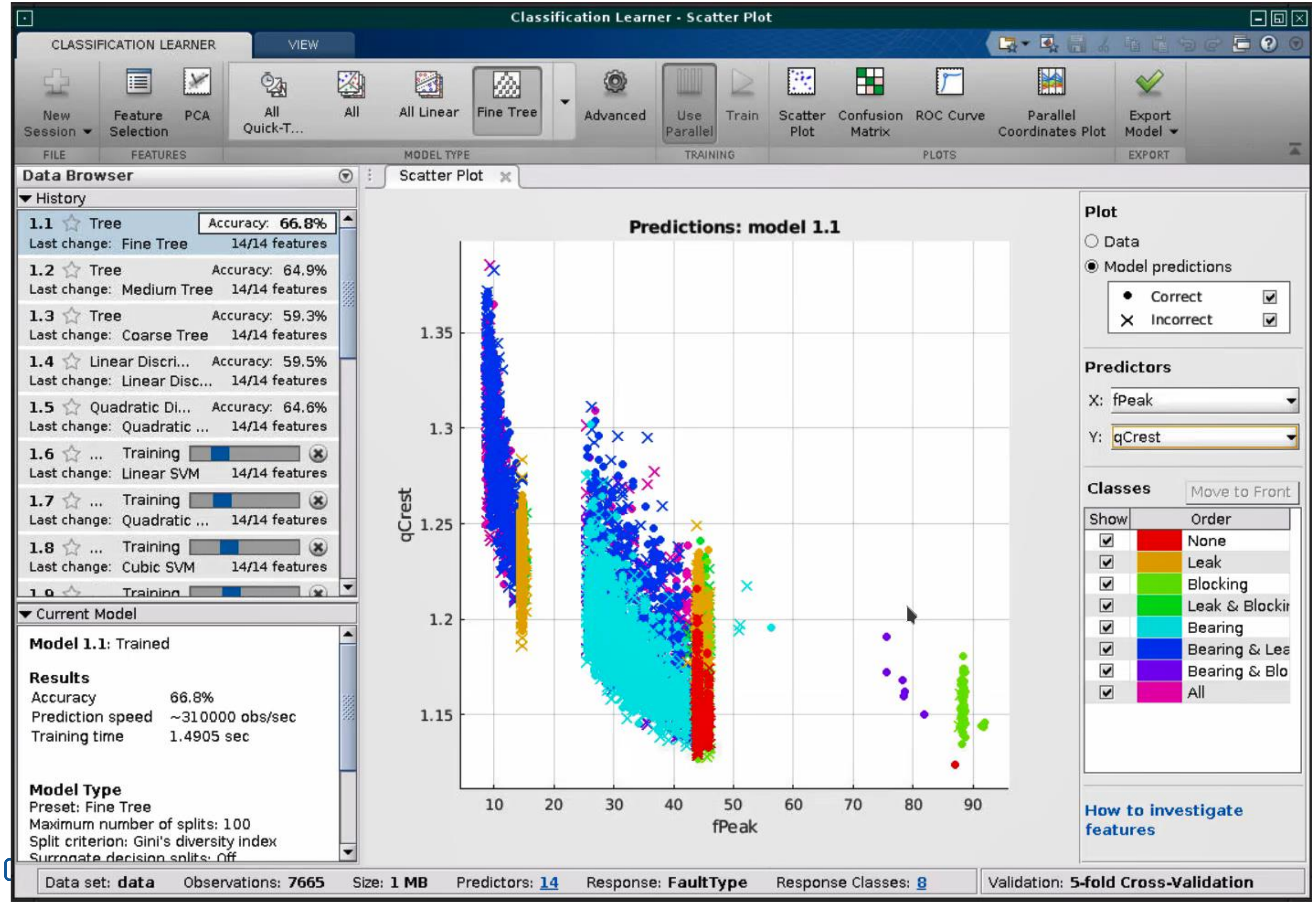
Remaining Useful Life (Regression)

Plant Operator

Process Engineer

3 Develop Predictive Models

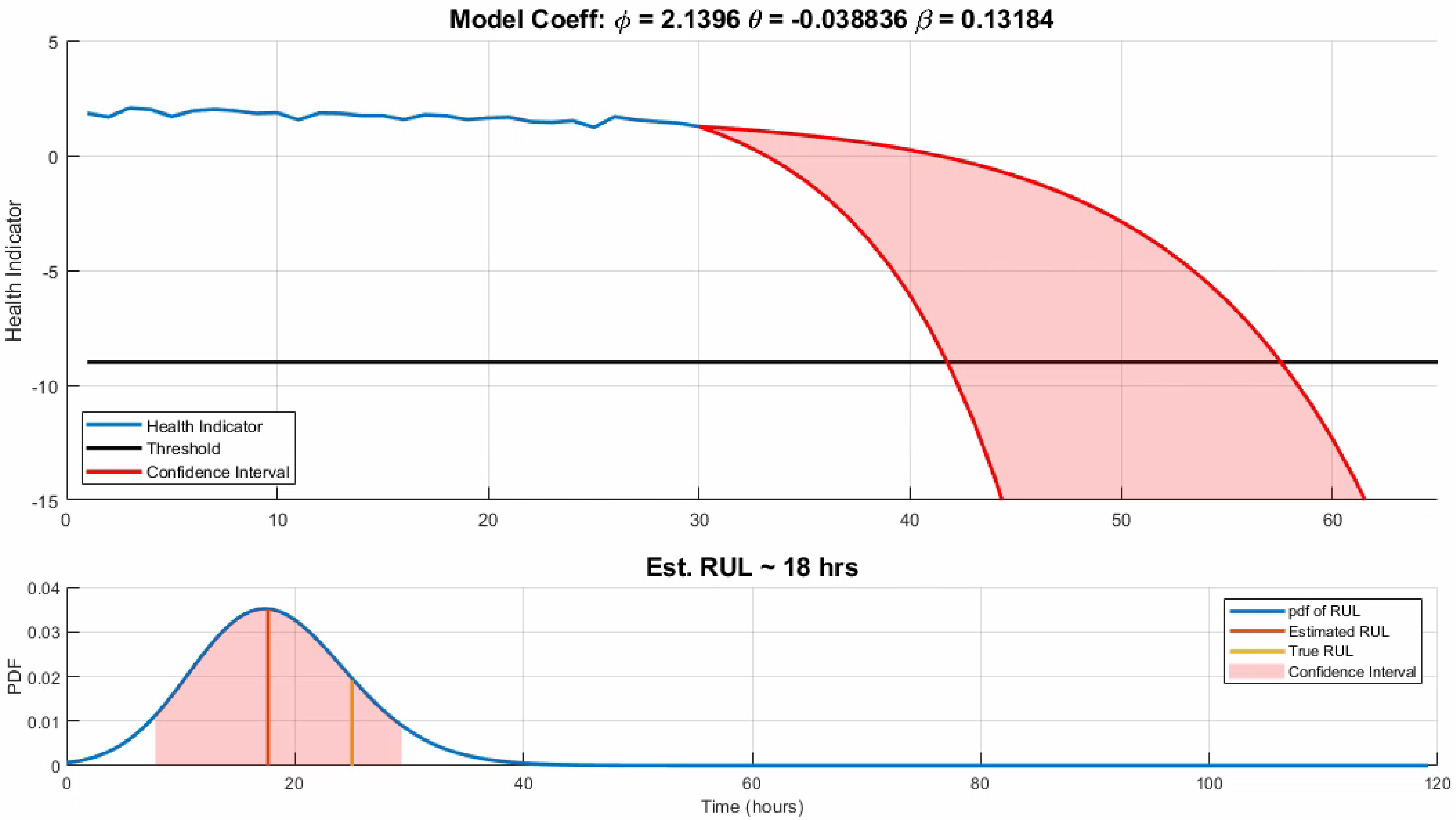
Develop Machine Learning Models



Process Engineer

3 Develop Predictive Models

Estimate Remaining Useful Life



$$S(t) = \phi + \theta(t)\, e^{(\beta(t)t+\epsilon(t) -\frac{\sigma}{2})}$$

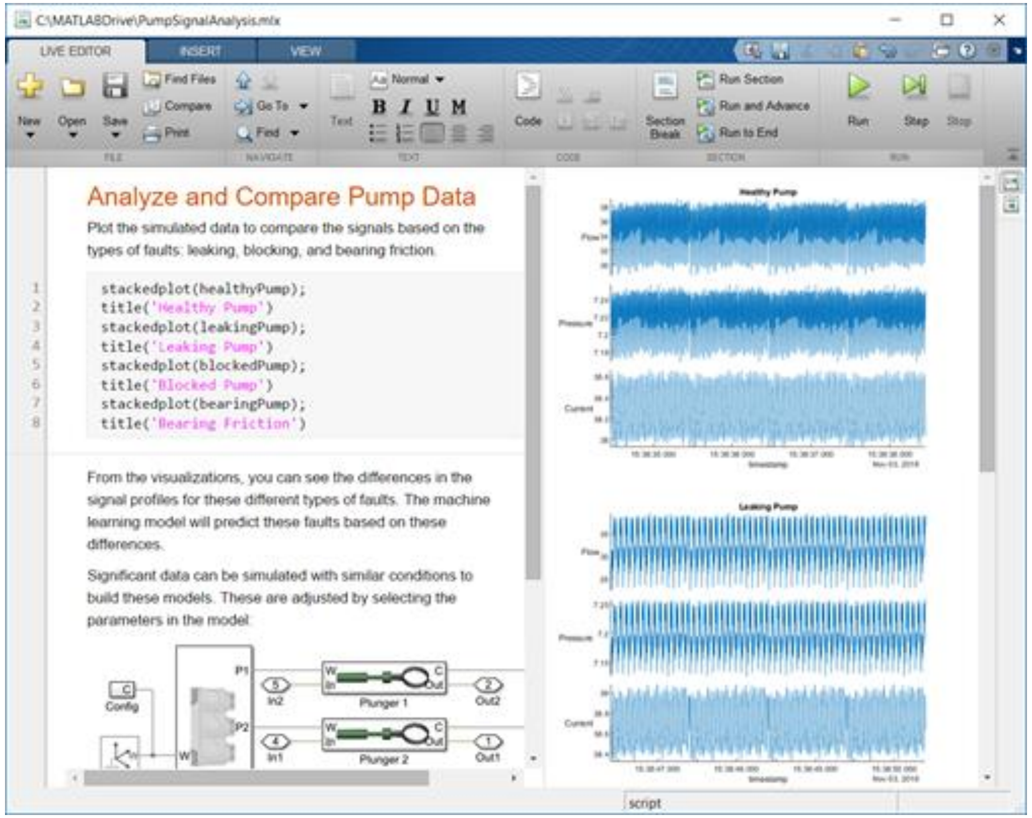
4 Integrate with Production Systems

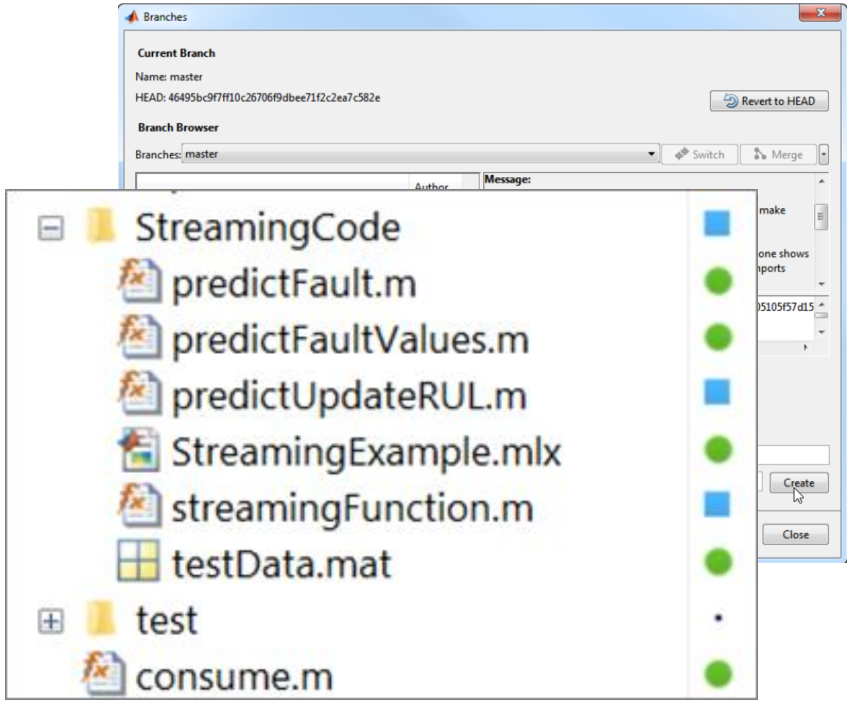
Share with the team

Process Engineer

Review results with Operator

Share code with System Architect





.pdf, html, LaTeX

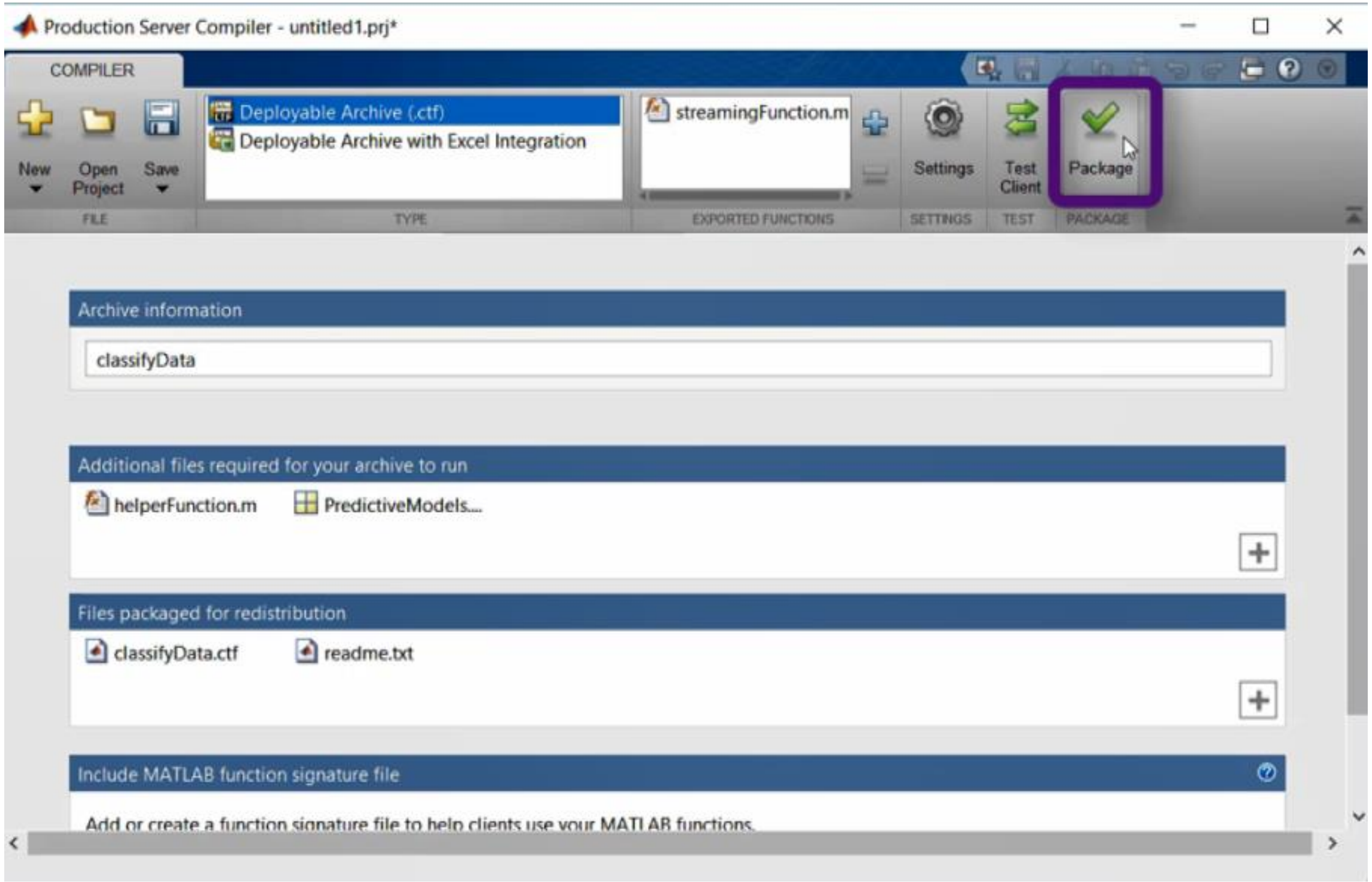
Source Control

MathWorks

4 Integrate with Production Systems

Package Stream Processing Function

Process Engineer



System Architect

4 Integrate with Production Systems

Review System Requirements

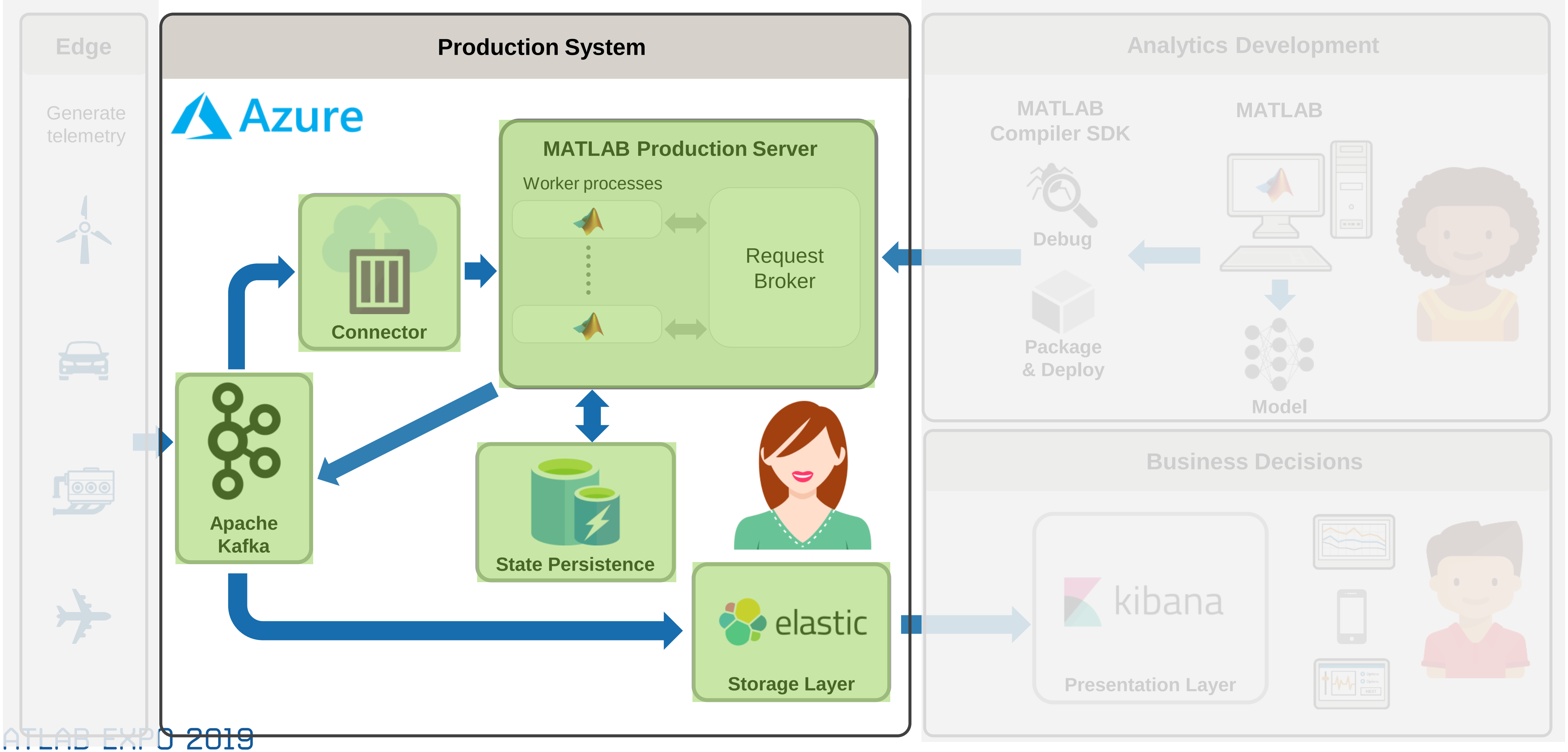
- Requirements from the Process Engineer
  - Every millisecond, each pump generates a time-stamped record of flow, pressure, and current
  - Model expects 1 sec. window of data per pump
  - Initially, 1’s – 10’s of devices, but quickly scale to 100’s

Process Engineer

- Requirements from the Operator
  - Alerts when parameters drift outside the expected ranges
  - Continuous estimating of RUL for each pump

Operator

Integrate Analytics with Production Systems



System Architect

4 Integrate with Production Systems

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
…	…	…	…	…

State

MATLAB Function

State

MATLAB Function

State

MATLAB Function

State

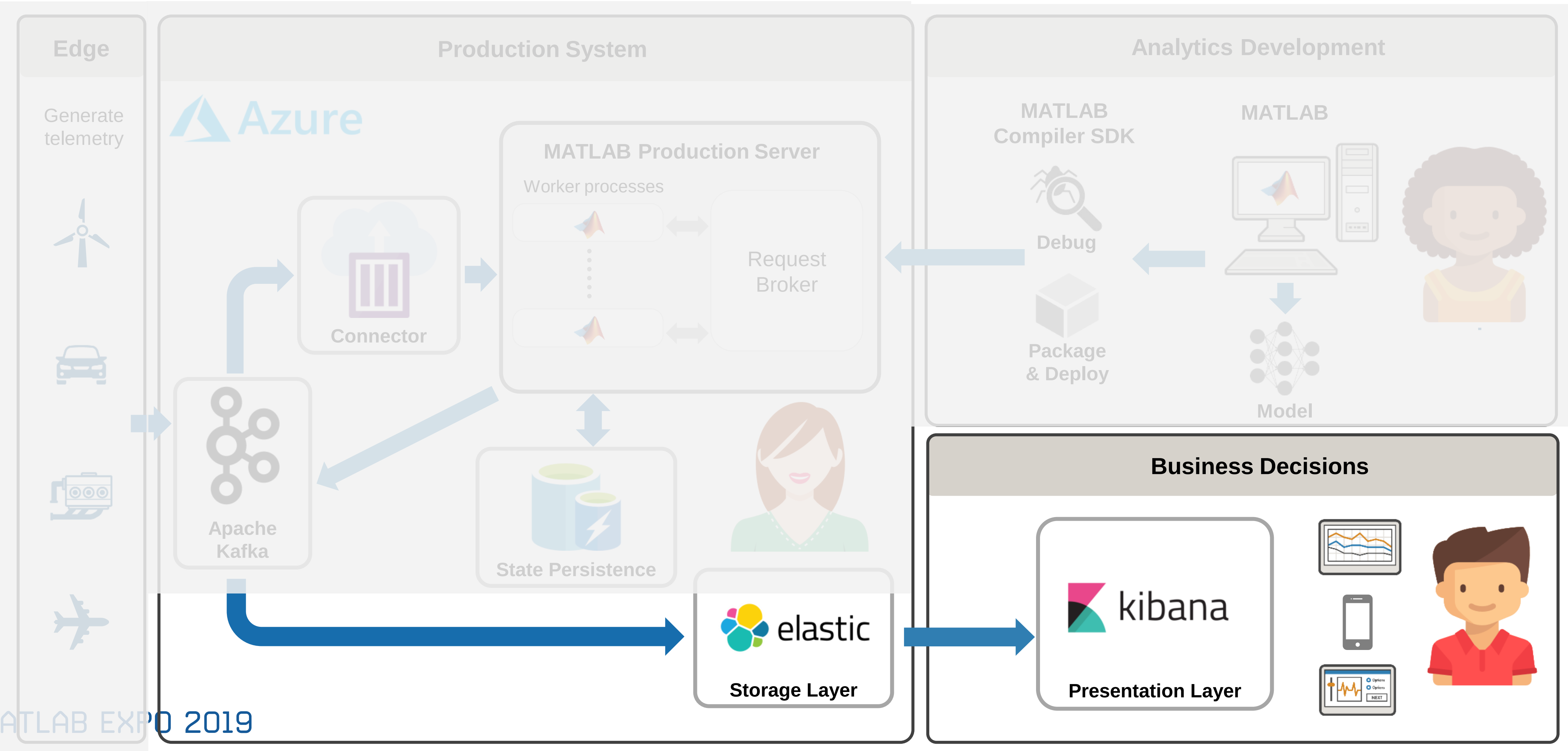
Output Stream

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	8

Complete your application

System Architect

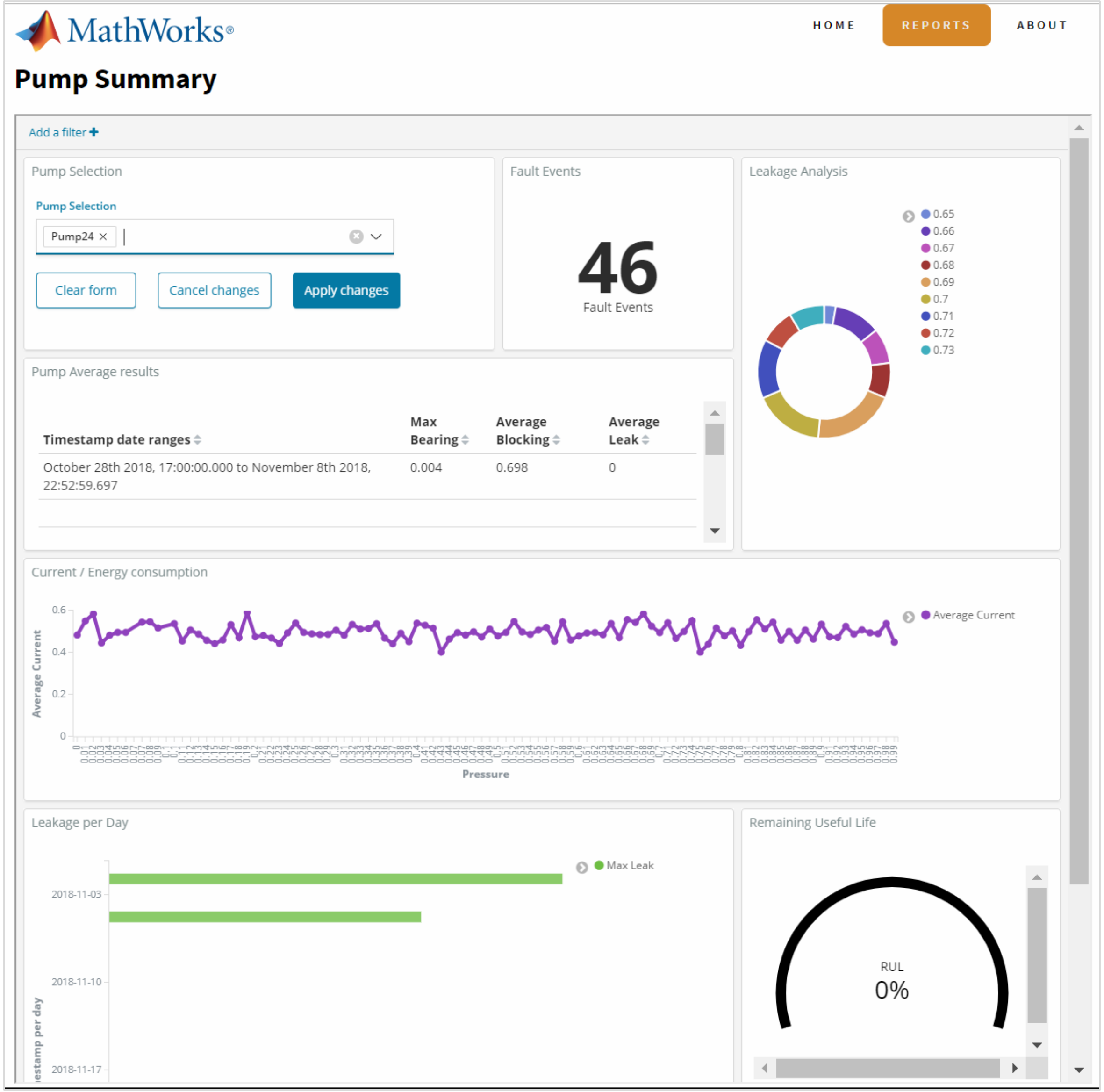
4 Integrate with Production Systems



Plant Operator

5 Visualize Results

Complete Your Application



Team Retrospective

- Completed demo of full system in 3 week sprint
- Successfully used digital twin to generate faults and train models
- Fast prototyping of physical and AI models with MATLAB and Simulink. Easy integration with OSS
- Cloud platform enabled faster IT setup

Takeaways

- You will face streaming data at some point
- Infrastructure may vary - MATLAB is always there
- Large or Small data, the concepts are the same
- Talk to us: www.mathworks.com

Resources to learn and get started

- [GitHub: MathWorks Reference Architectures]()
- [Working with Enterprise IT Systems]()
- [Data Analytics with MATLAB]()
- [Simulink]()