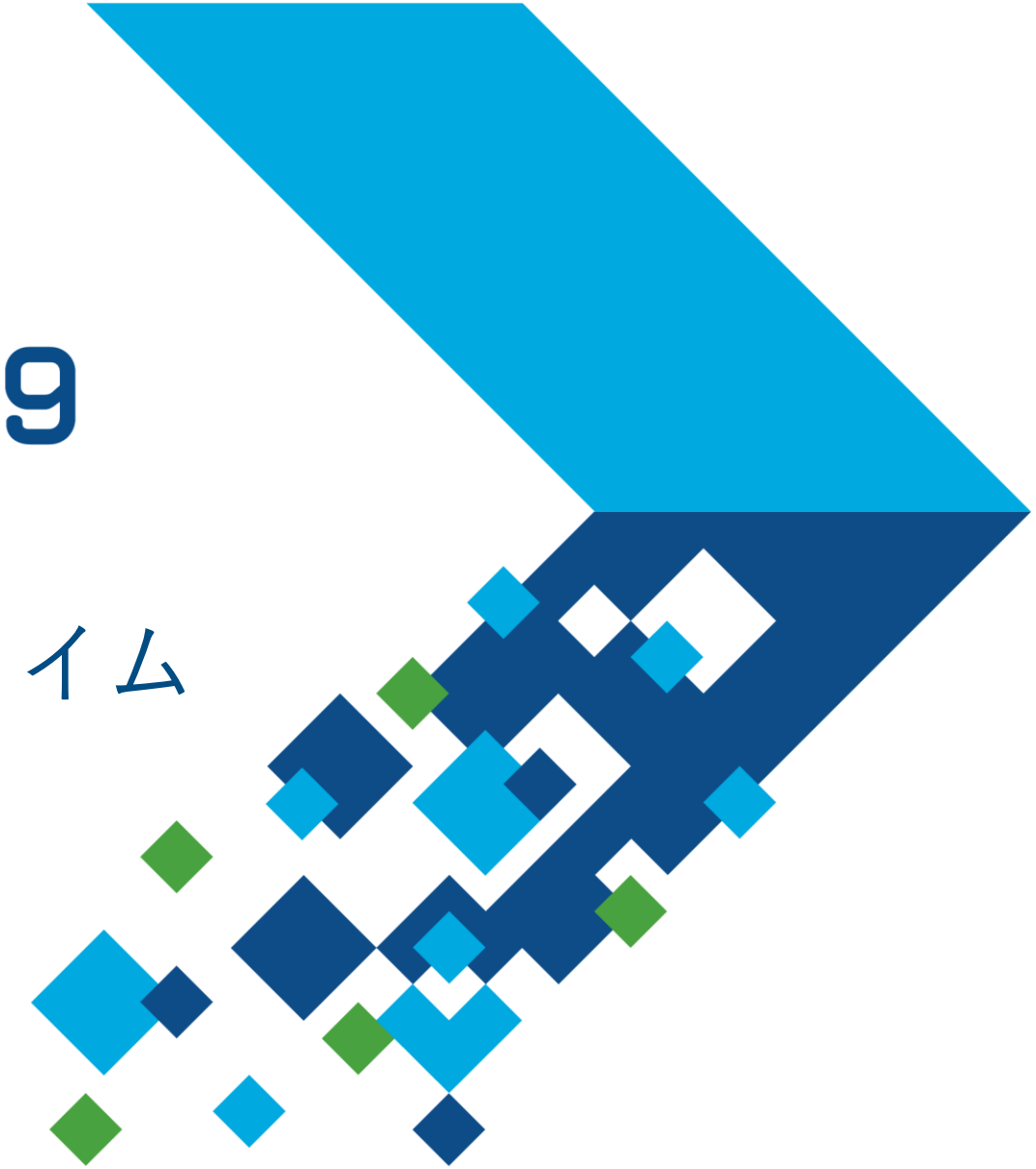
# MATLAB EXPO 2019

## モノづくりにおけるリアルタイムの意思決定にAIを展開

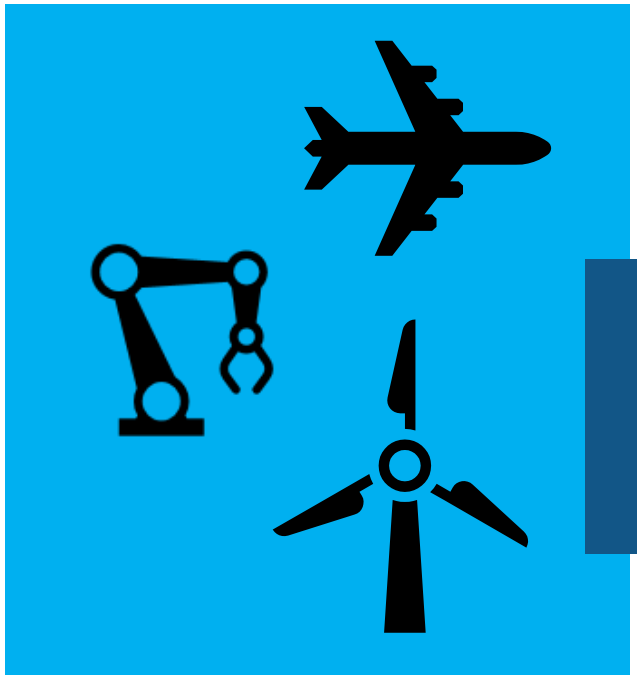
MathWorks® Japan
齊藤 甲次朗



# アジェンダ

- はじめに
- 予知保全システム開発事例
  - 概要
  - 予知保全のモデル化
  - 予知保全モデルのシステム化
  - 予知保全システムの利用
- まとめ

# リアルタイム解析が求められるワケ



## 予知保全

*運用の効率向上*

*計画外のダウンタイムを削減*

**リアルタイム解析を必要とする**
**アプリケーションが増えています**

ジェットエンジン: ~800 TB /day

タービン: ~ 2 TB /day

## 医療機器

*患者の安全*

*より良い治療効果*

## コネクテッドカー

*安全、保守*

*先進運転*

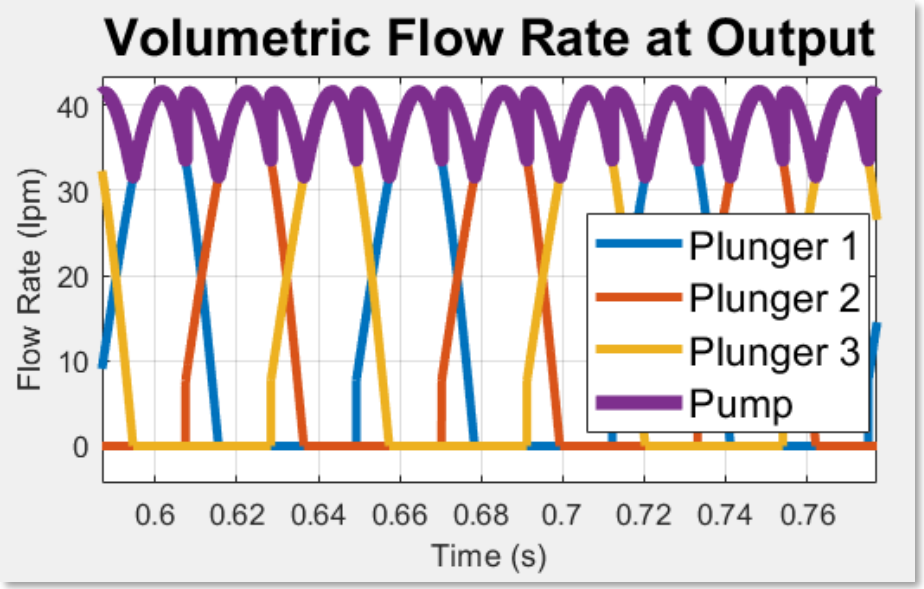
車: ~25 GB / hour

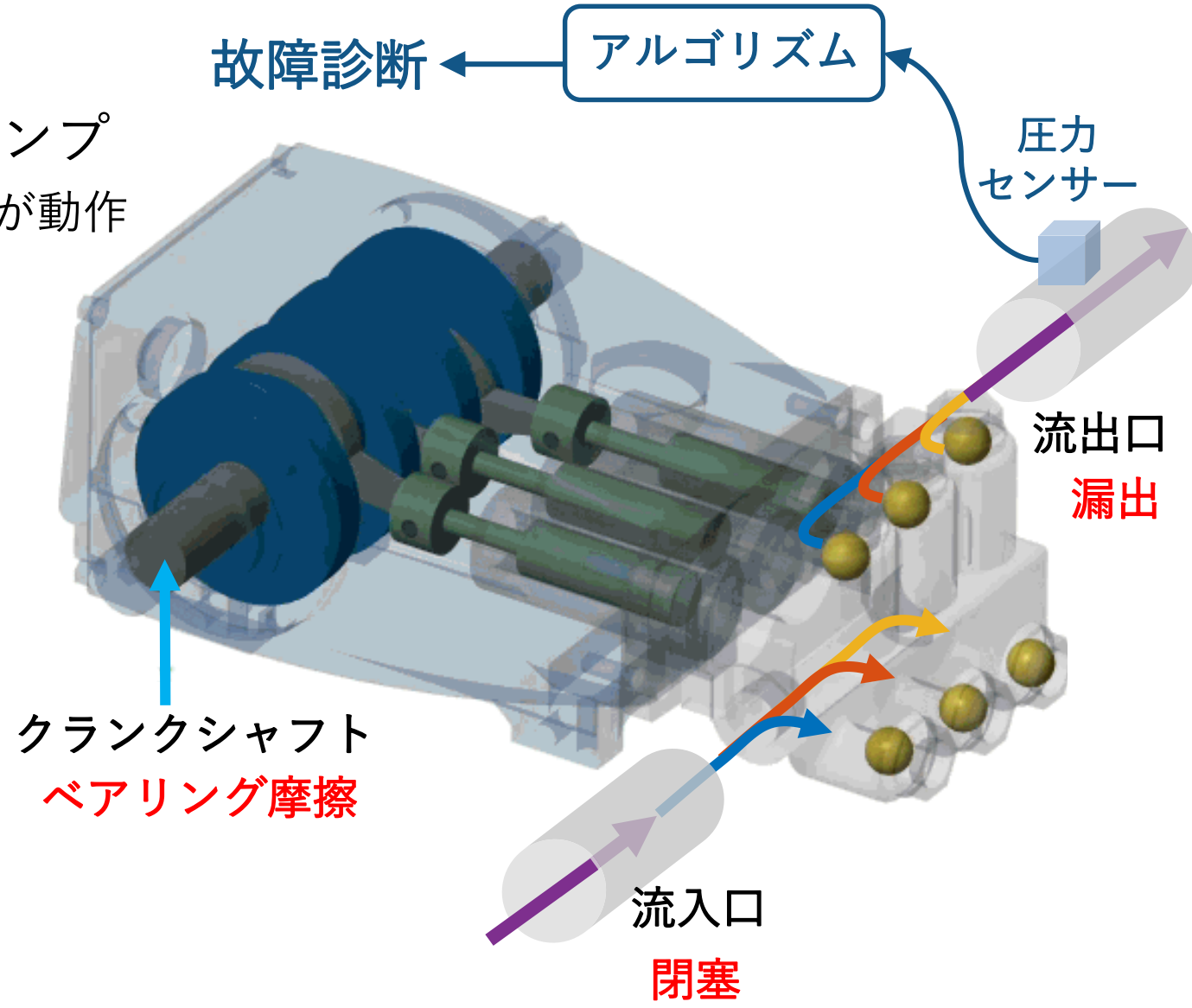
# 予知保全システム開発事例
## ～概要～


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

# 対象のポンプ

- 3つのプランジャーが動作するポンプ
  - 120°ずつの位相差でプランジャーが動作
  - 1つのチャンバーが常に排出
  - 3種類の故障



# 予知保全プロジェクトの担当者

## プロセスエンジニア

予知保全の解析モデル
を開発

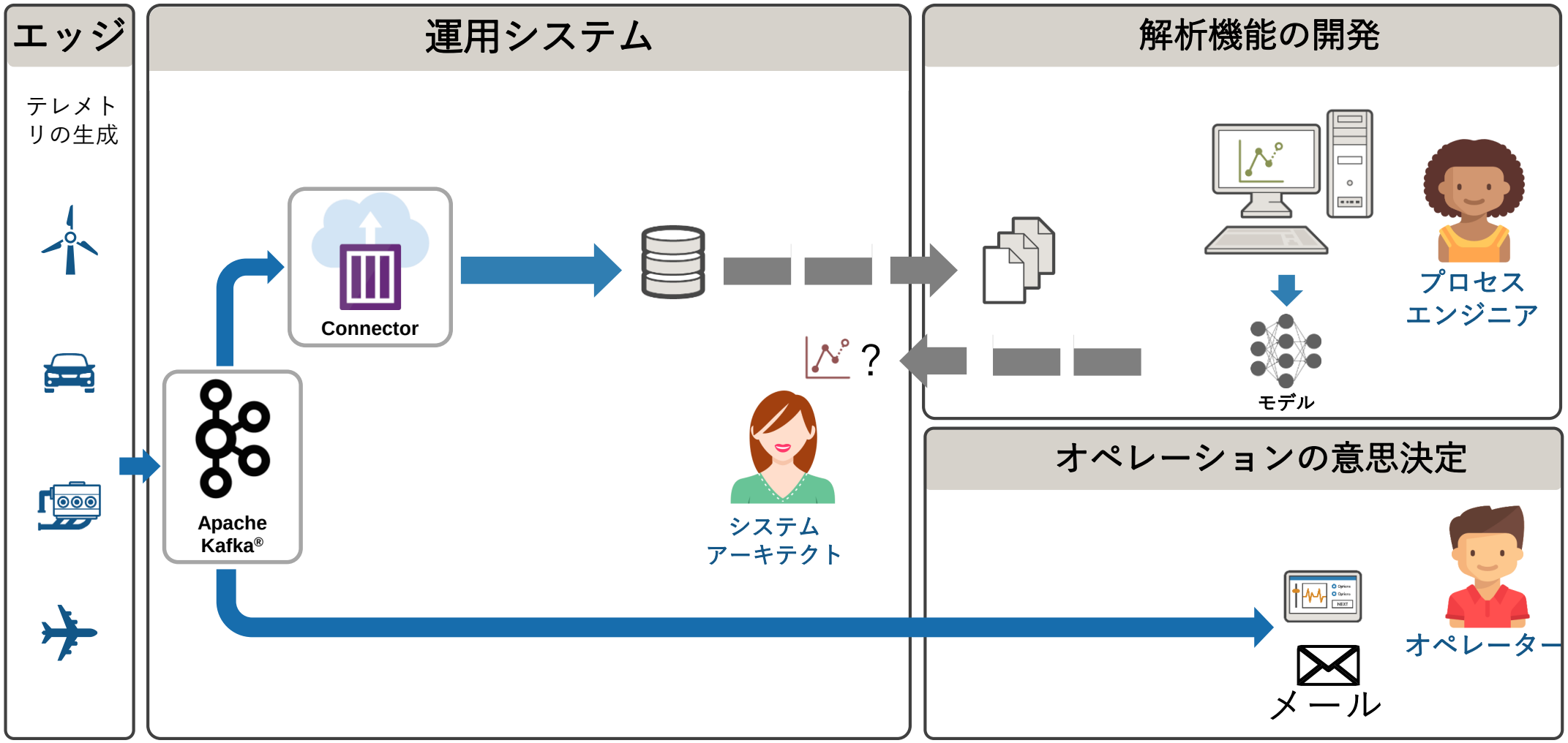
## システムアーキテクト

予知保全のシステム
を開発

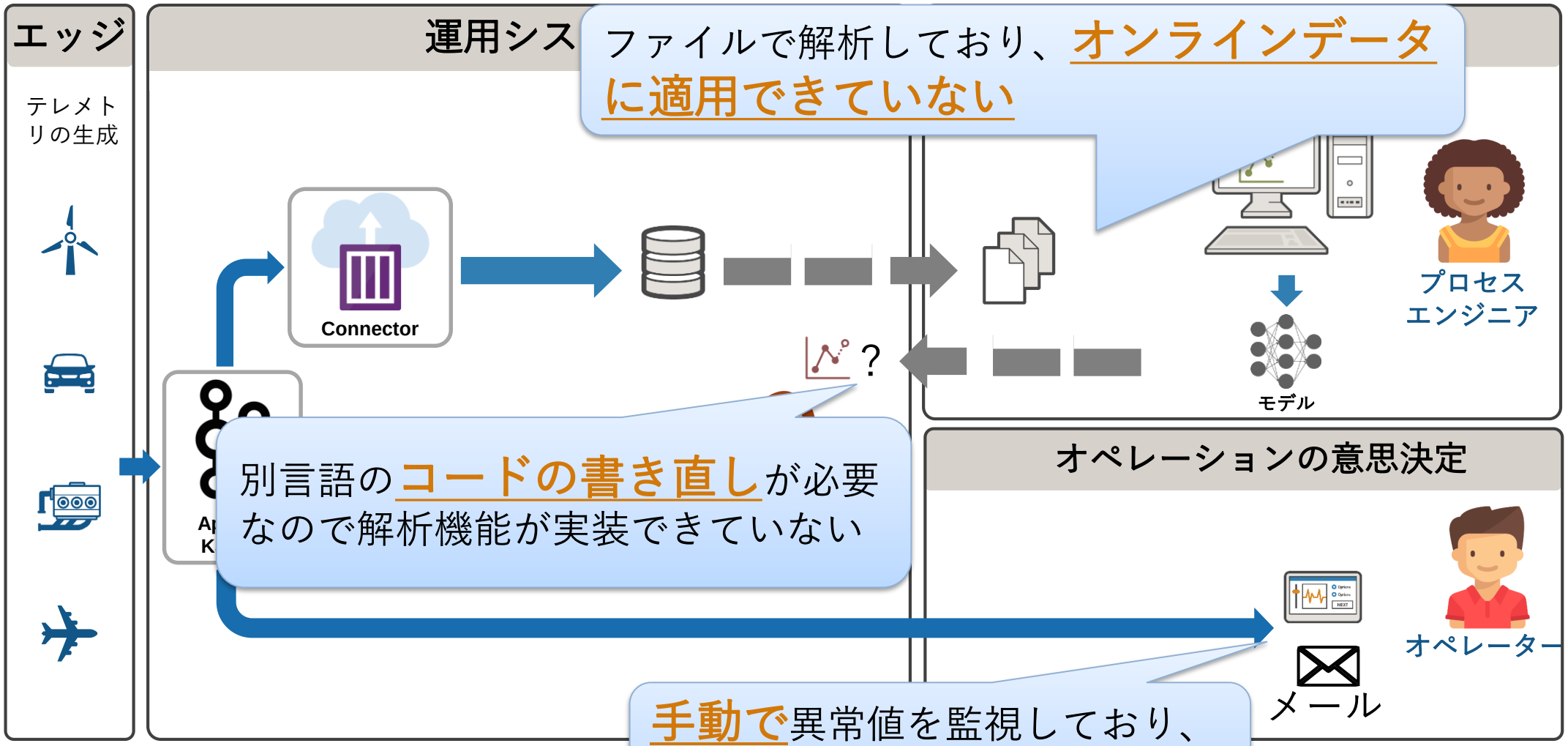
## プラントオペレーター

システムを利用して
オペレーションの意思決定

# 現状の予知保全システムと課題



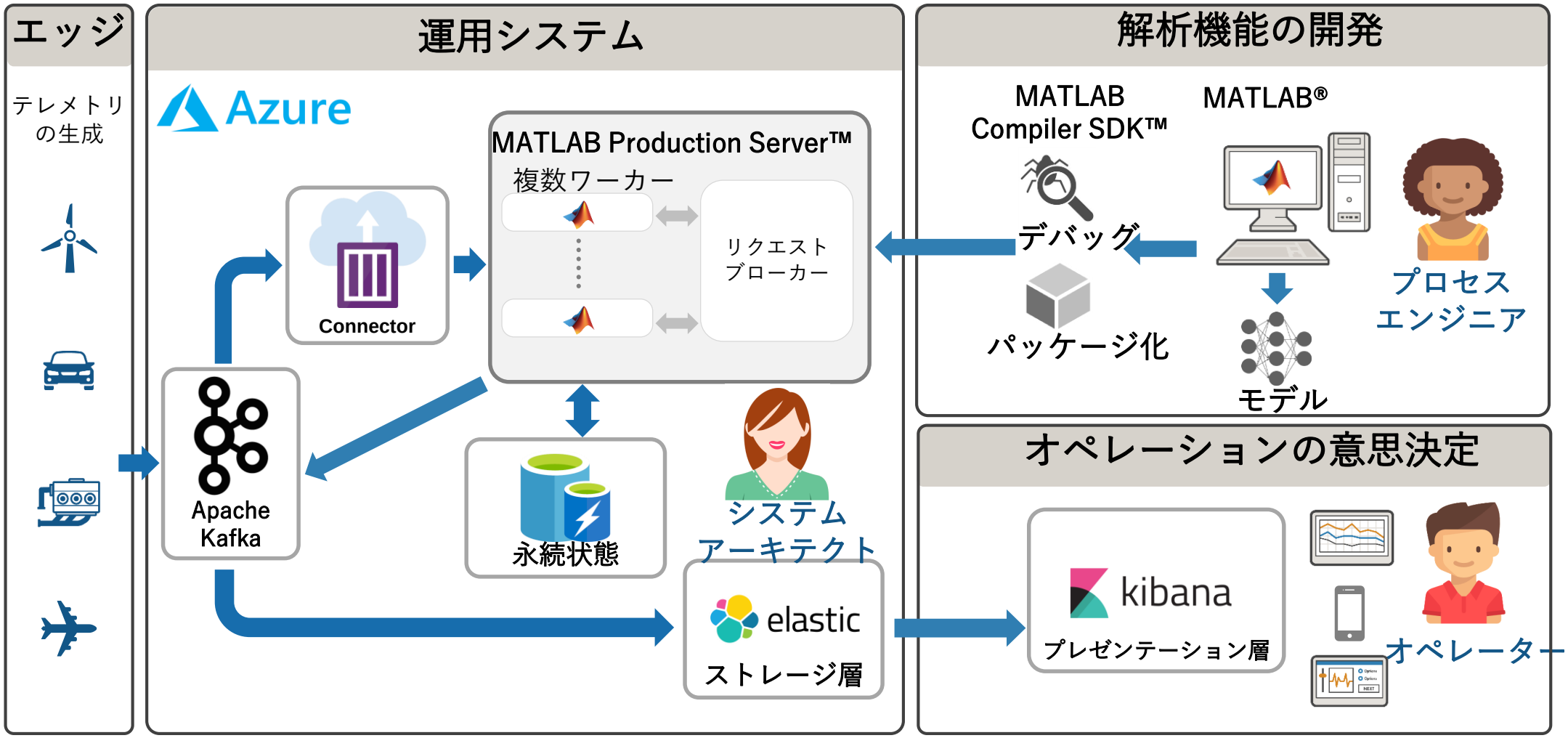
# 現状の予知保全システムと課題



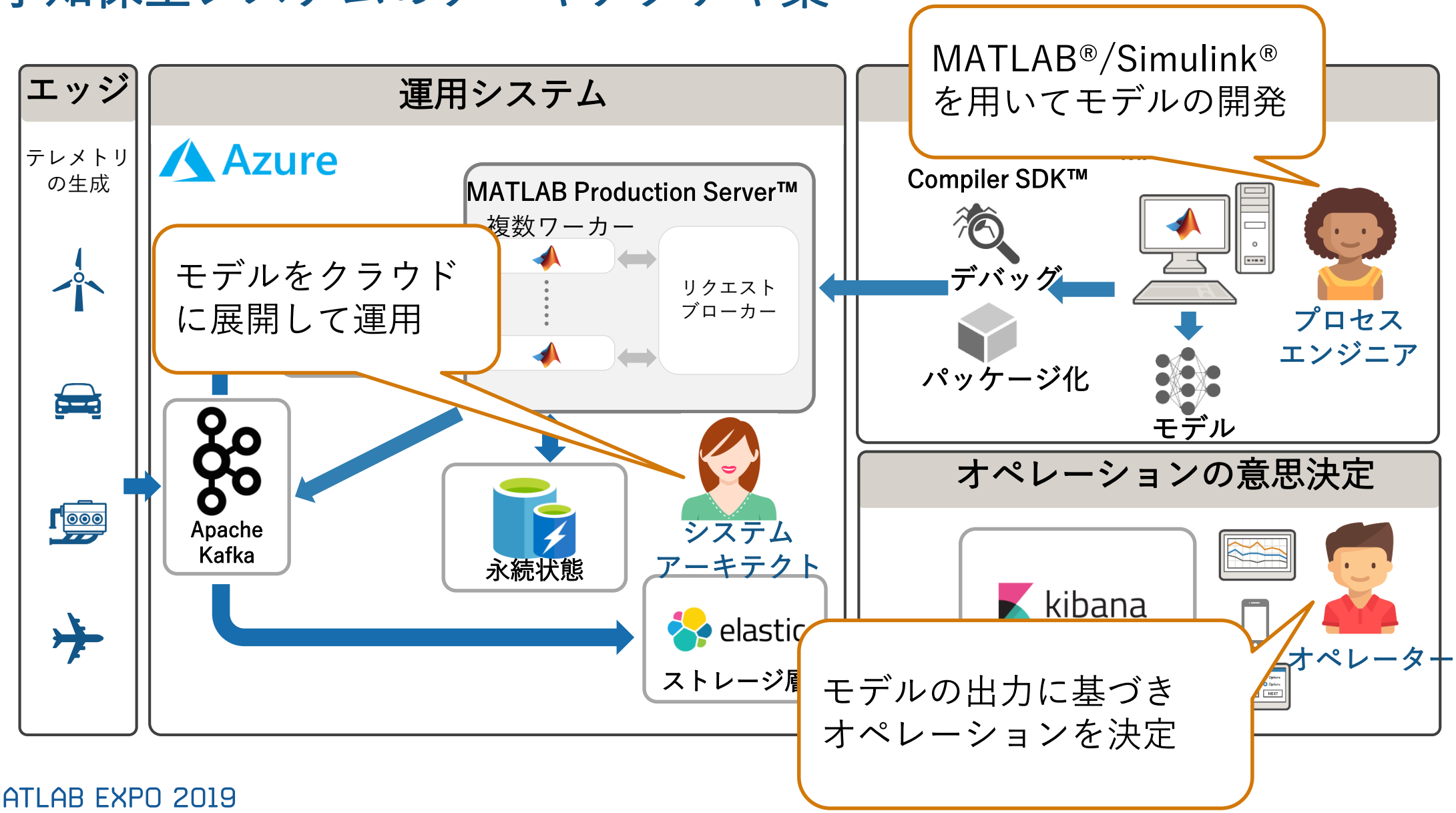
# 予知保全システム開発のプロジェクト概要

| 項目 | 概要 |
| --- | --- |
| 期間 | 1ヶ月 |
| 内容 | • 予知保全のアルゴリズム開発(モデル化)<br>• モデルのシステム化<br>• ストリーミング処理 |
| 要求 | **オペレーションからの要求**<br>• 全ポンプの流量、圧力、電流を監視してオペレーションの状態を常時把握できること<br>• パラメータが範囲外にドリフトした際にすぐにリアクションができるようにすること<br>• 全ポンプの故障までの寿命を常時推定し、保守や交換をスケジューリングできること<br><br>**システムアーキテクトからの要求**<br>• ITの予算が限られているが、投資する前にソリューションがうまくいくか把握できること |

# 予知保全システムのアーキテクチャ案



# 予知保全システムのアーキテクチャ案



# 予知保全システム開発のアプローチ

1. データへのアクセスと探索
2. データの前処理
3. 予測モデルの開発
4. システムへの統合
5. 結果の可視化

**1. データへのアクセスと探索**

- ファイル
- データベース
- センサー

**2. データの前処理**

- 乱雑データの扱い
- データ削減／変換
- 特徴抽出

**3. 予測モデルの開発**

- モデル作成
- パラメータ最適化
- モデル検証

**4. システムへの統合**

- デスクトップアプリ
- エンタープライズシステム
- 組込機器とハードウェア

**5. 結果の可視化**

- サードパーティダッシュボード
- Webアプリ

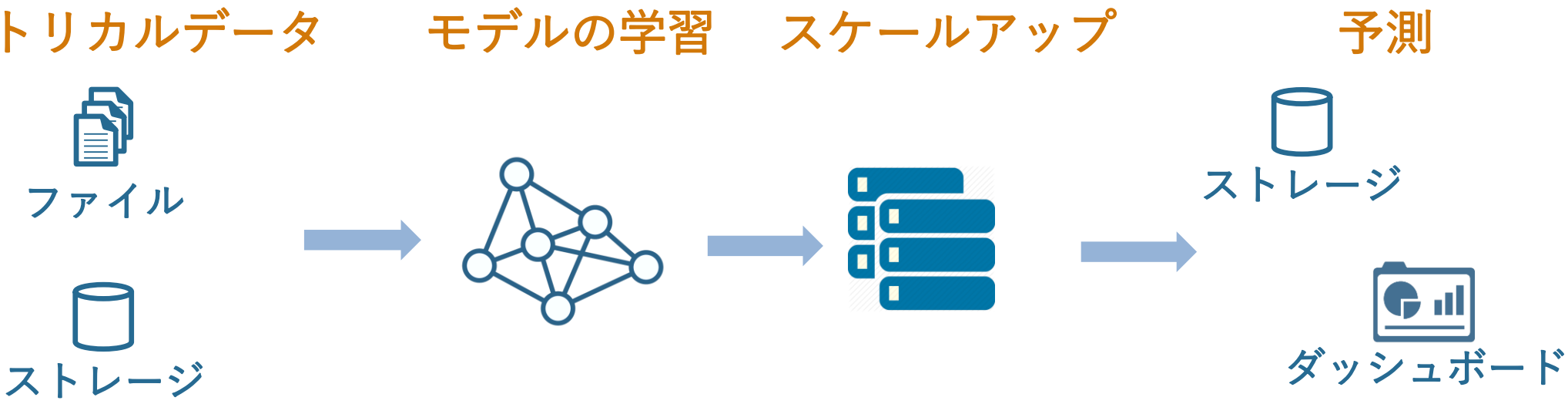
# 予知保全システム開発事例
## ～予知保全のモデル化～

1 データへのアクセスと探索

プロセス
エンジニア

# ヒストリカルデータの利用

## ①バッチ処理：ヒストリカルデータでモデルの開発とテスト

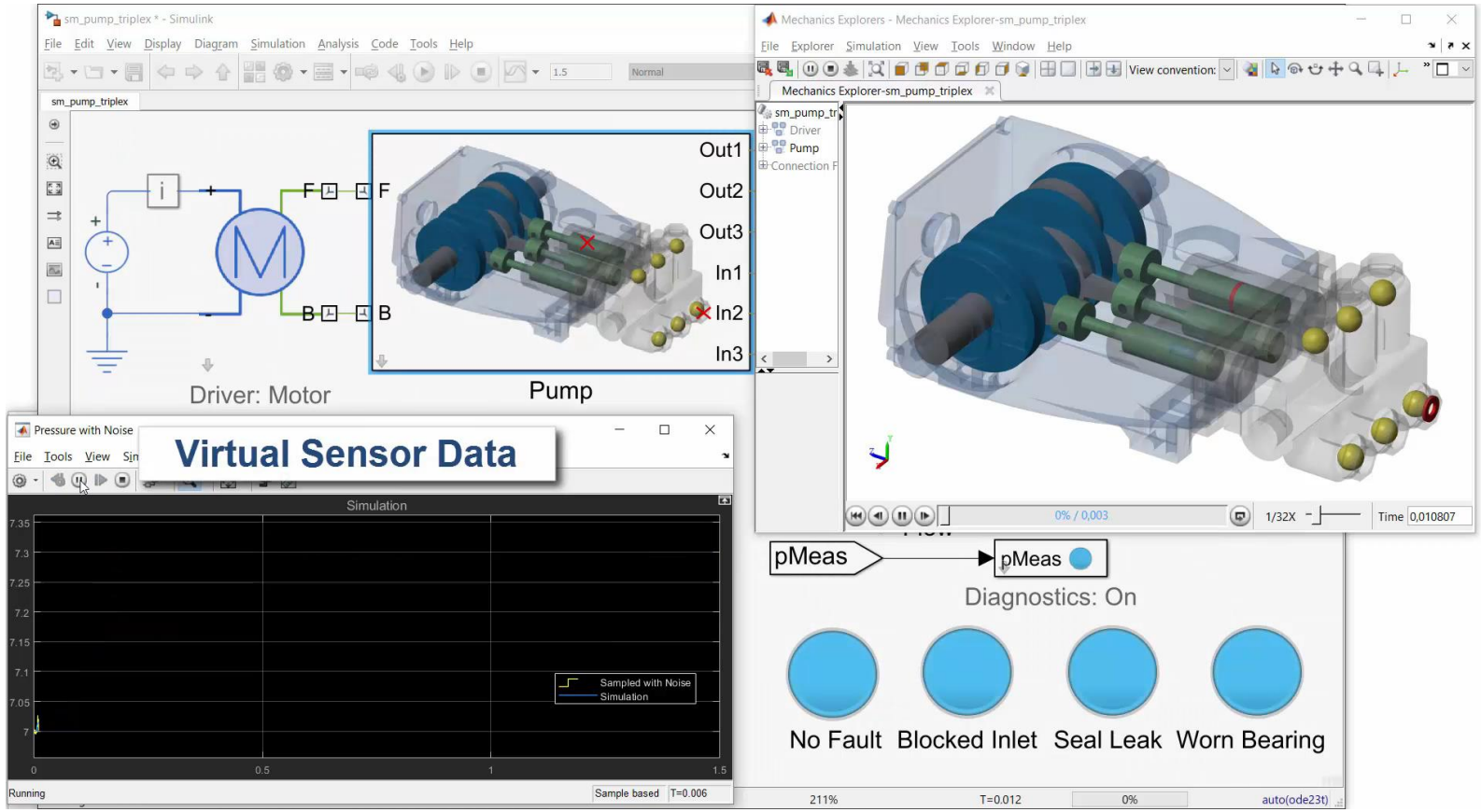


# 故障レベルの特定にセンサデータを利用

プロセス
エンジニア

ファイル
(正常データ)

シミュレーション
(異常データ)




**Simulinkモデルで故障をシミュレーション**
(故障データがあまり無く、実際に装置を故障させるのはコストが掛かりすぎるため)

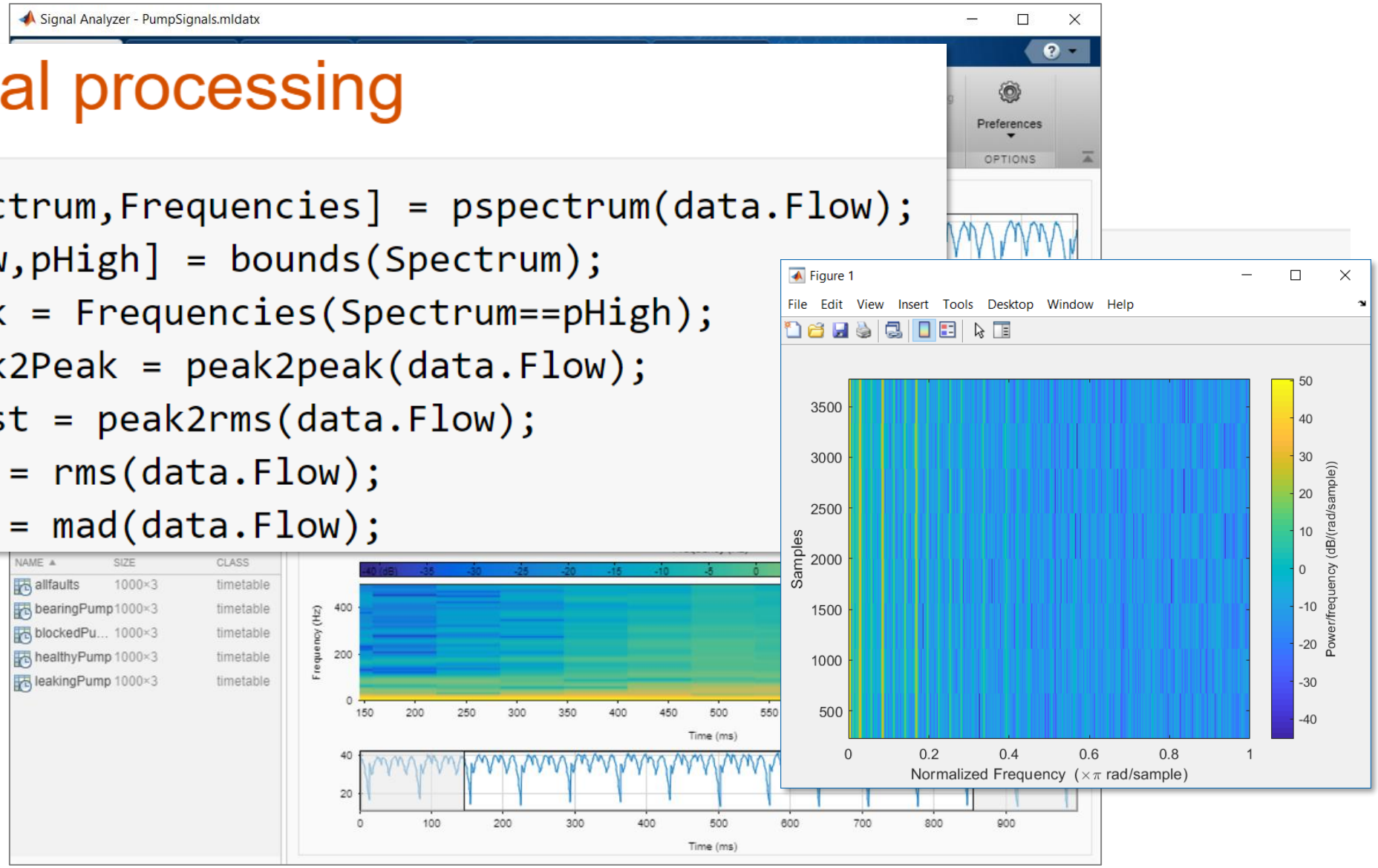
プロセス
エンジニア

2 データの前処理

# 信号データの前処理

## Signal processing

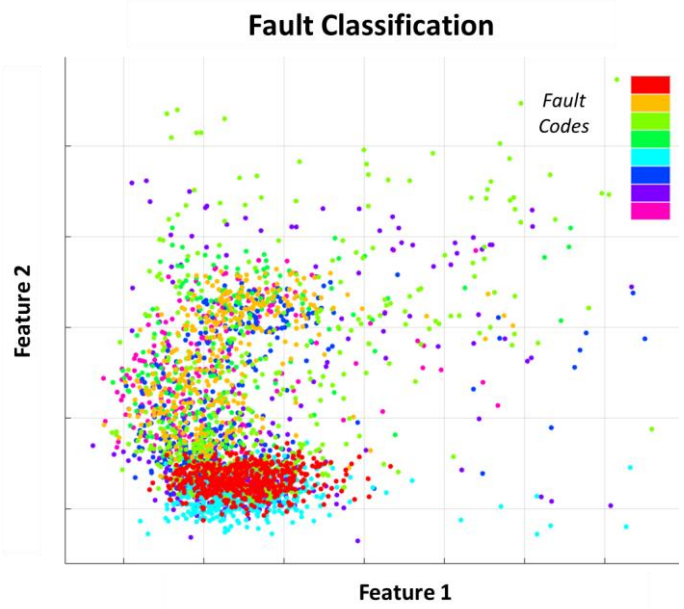
```
[Spectrum,Frequencies] = pspectrum(data.Flow);
[pLow,pHigh] = bounds(Spectrum);
fPeak = Frequencies(Spectrum==pHigh);
qPeak2Peak = peak2peak(data.Flow);
qCrest = peak2rms(data.Flow);
qRMS = rms(data.Flow);
qMAD = mad(data.Flow);
```

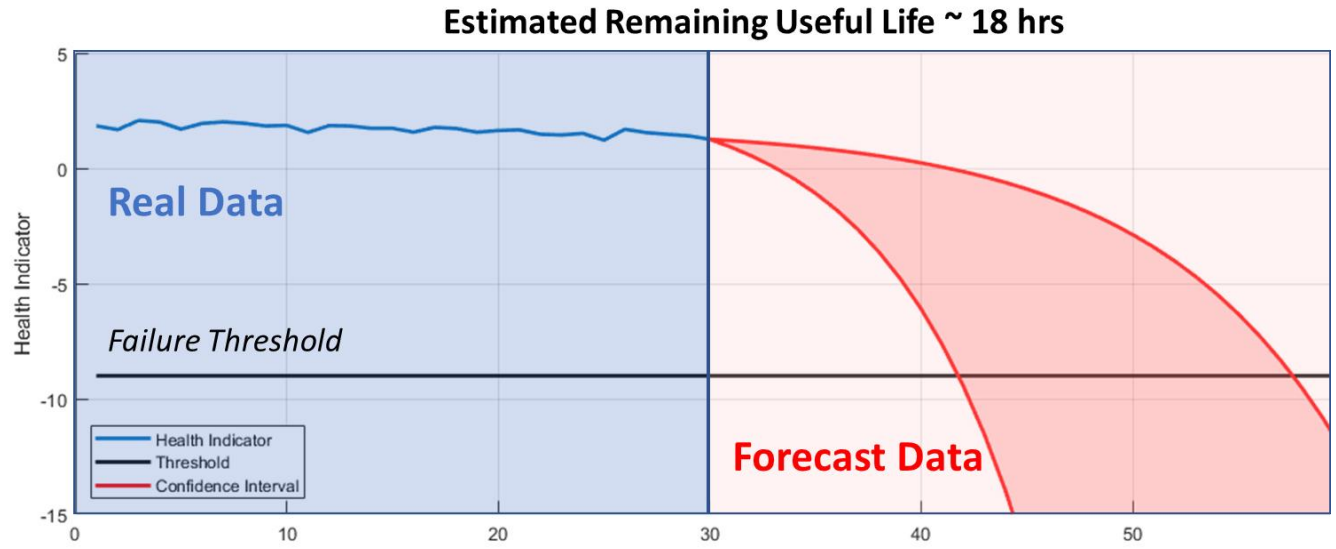


プロセス
エンジニア

3 予測モデルの開発

# MATLABでの予測モデルの開発




故障の種類
→分類

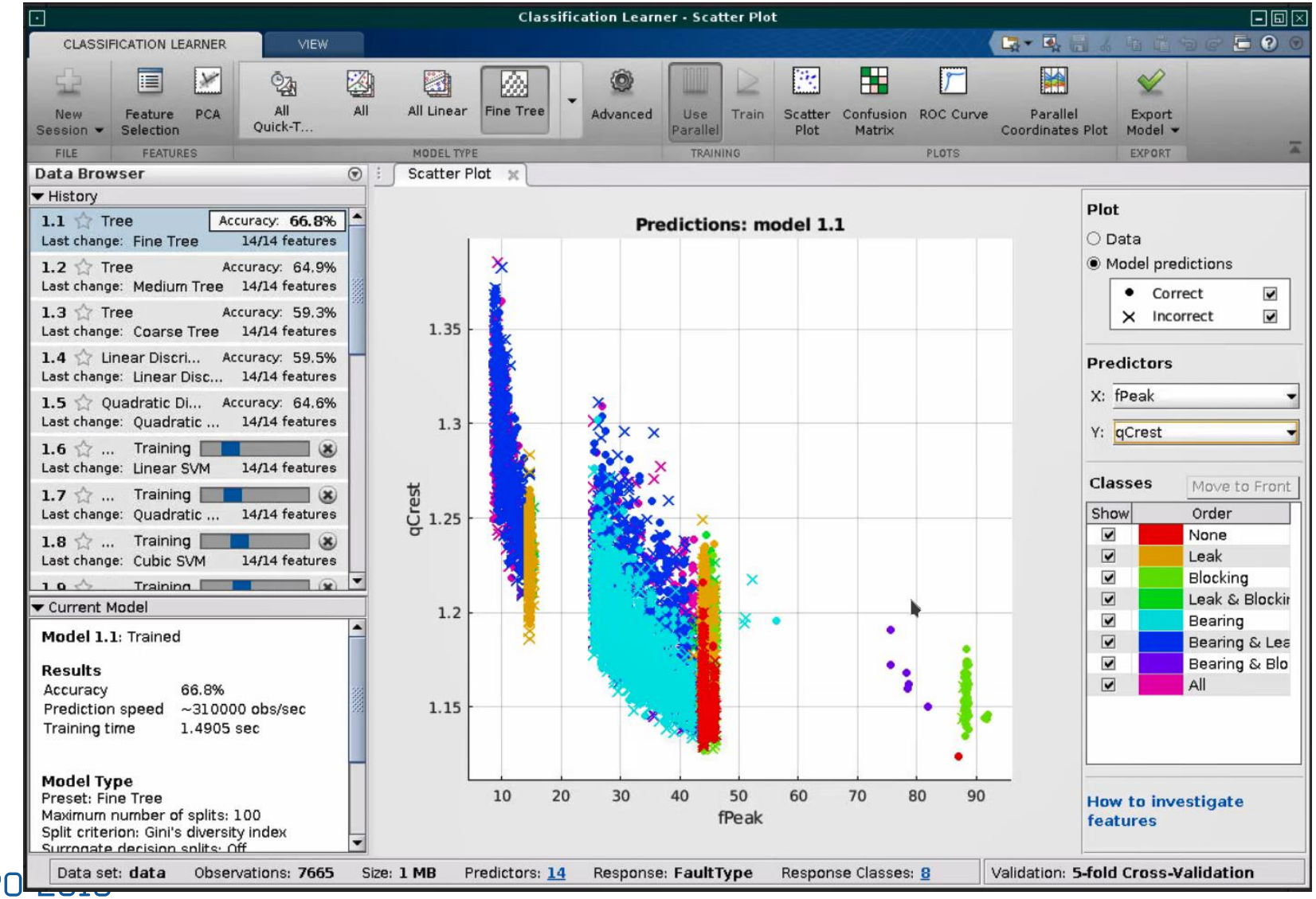
オペレーターに
必要な情報

故障までの予測時間
→回帰

プロセス
エンジニア

3 予測モデルの開発

# 機械学習モデル(分類)の開発

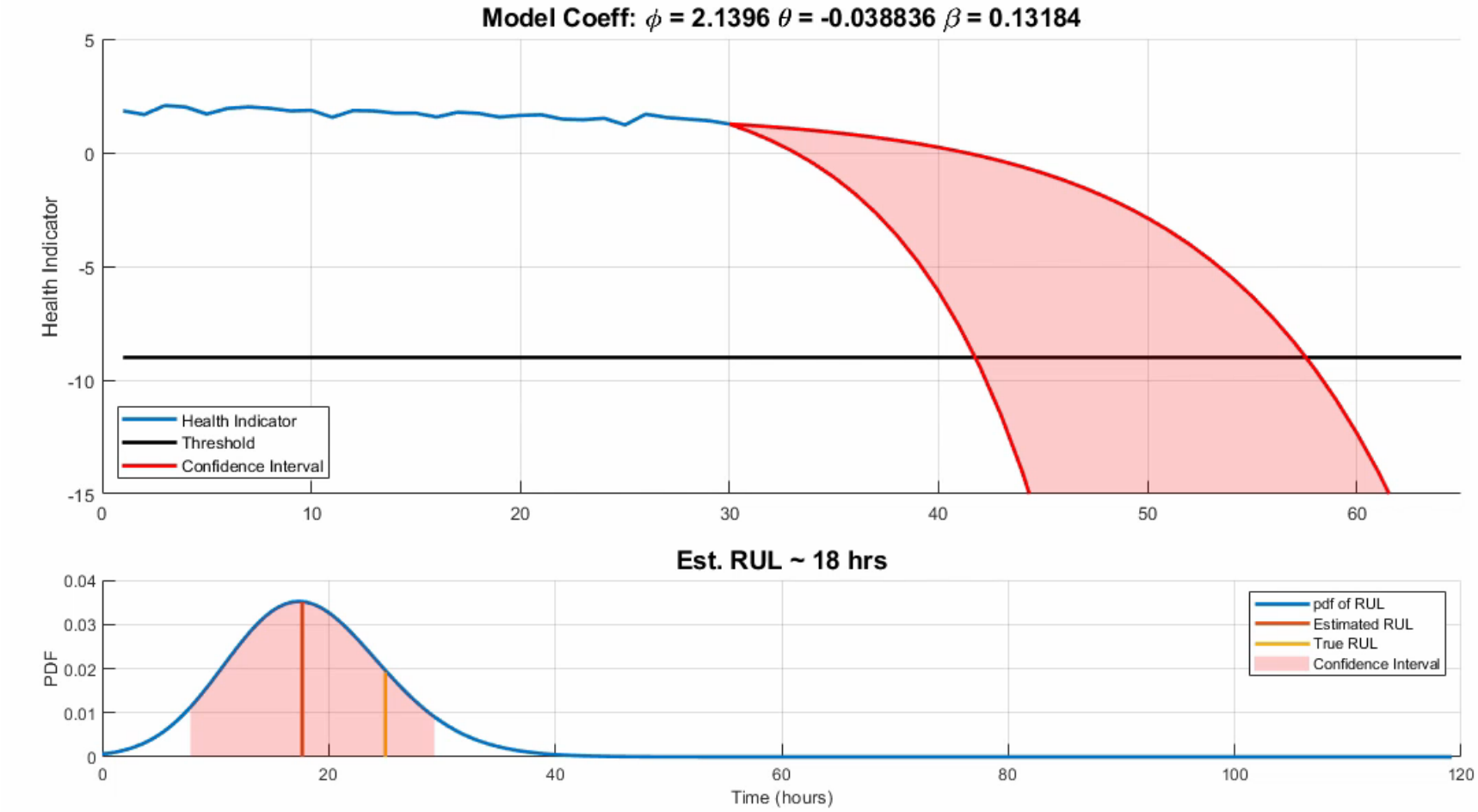



分類ラベル
- Blocking
- Leaking
- Bearing
- 上記組合せ

プロセス
エンジニア

3 予測モデルの開発

# 故障までの時間を予測(回帰)



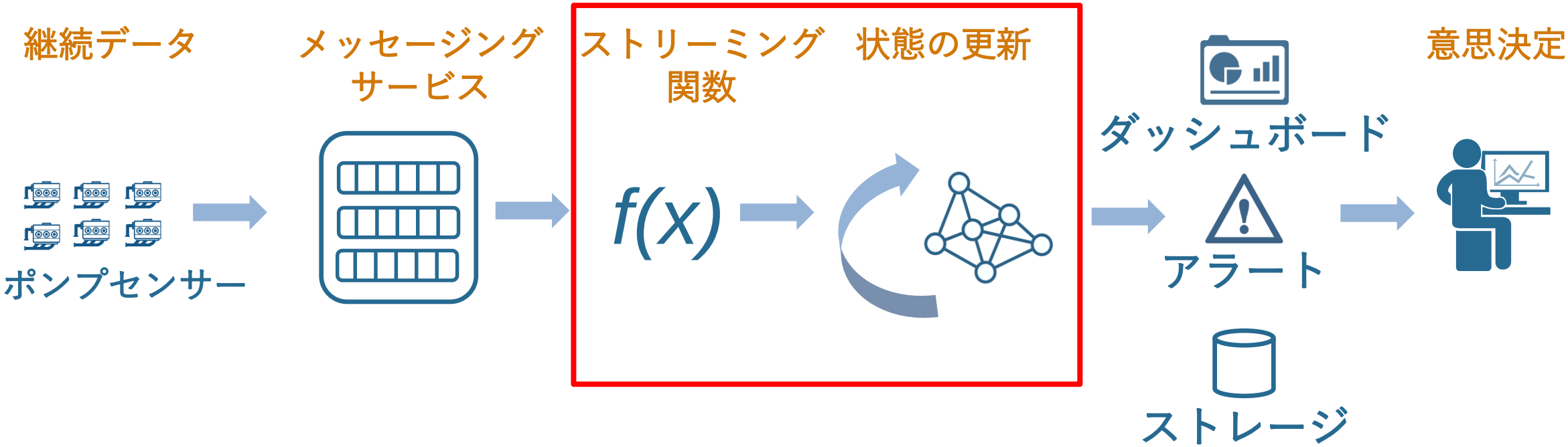

$$S(t) = \phi + \theta(t)\, e^{(\beta(t)t + \epsilon(t) - \frac{\sigma}{2})}$$

プロセス
エンジニア

4 システムへの統合

# ストリーム処理関数を開発

②*ストリーム処理:* リアルタイムでセンサーデータにモデルを適用



プロセス
エンジニア

4 システムへの統合

# ストリーム処理関数を開発

## 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
```

データが送られてきたら
全ての窓を処理

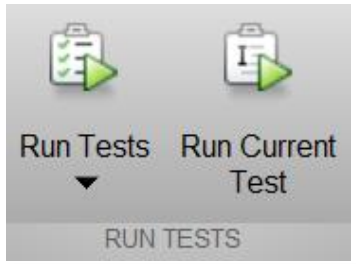
1つ前の状態

処理される現在のデータ窓

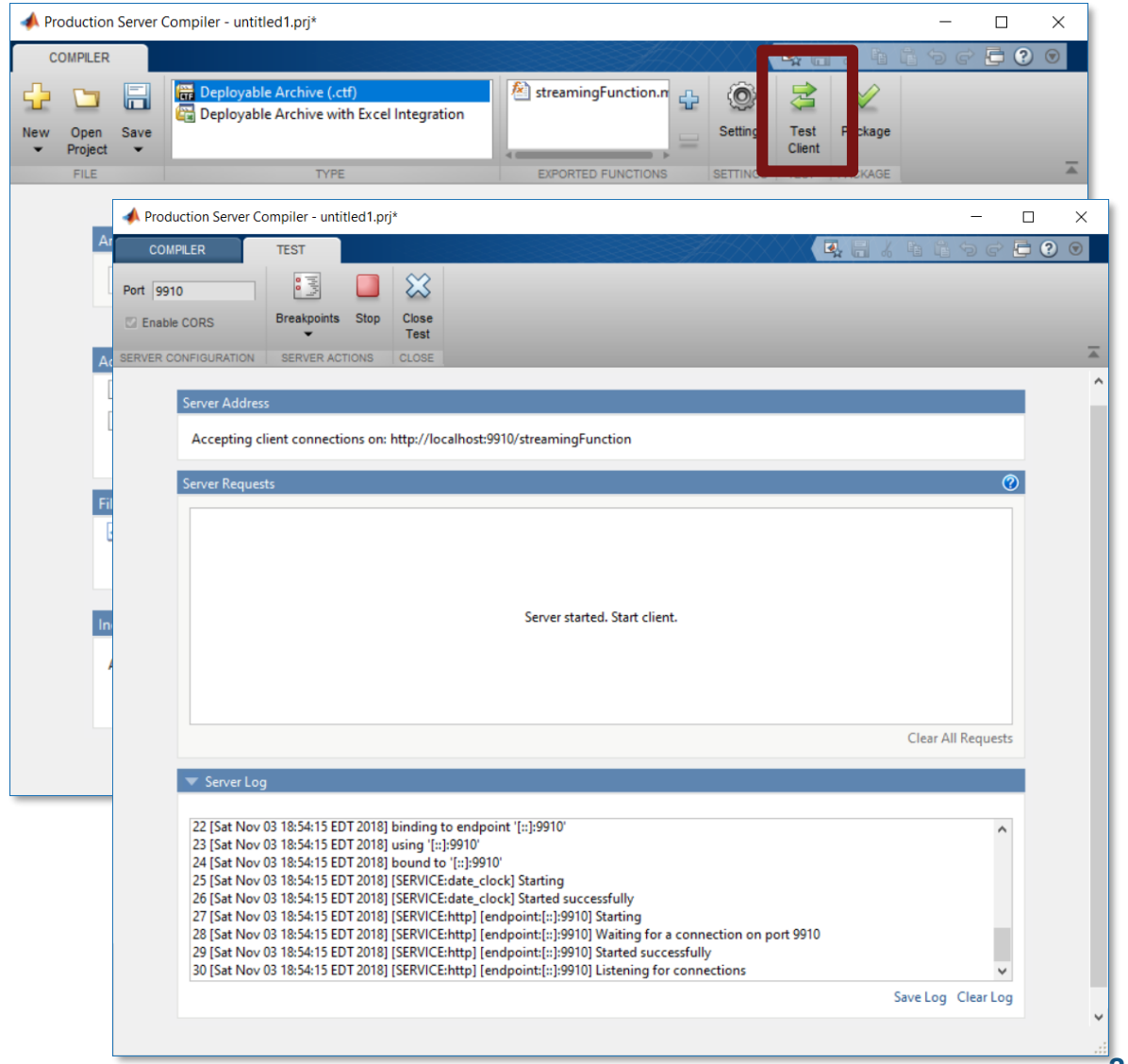
プロセス
エンジニア

4 システムへの統合

# ストリーム処理関数のテスト



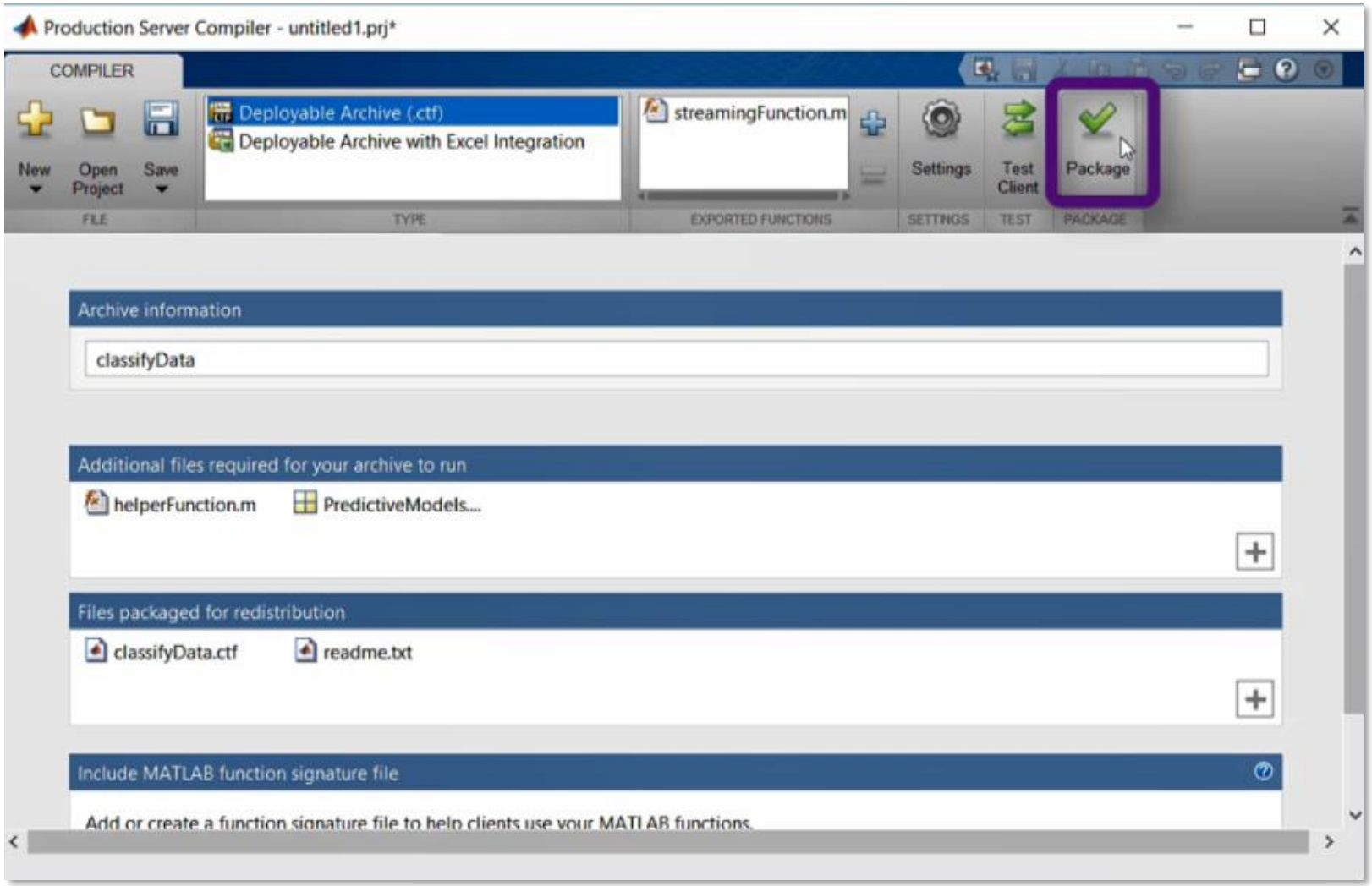

```
results = runtests('predictFaults_tests')

Running predictFaults_tests
....
Done predictFaults_tests
__________
results =
  1×4 TestResult array with properties:

    Name
    Passed
    Failed
    Incomplete
    Duration
    Details
Totals:
   4 Passed, 0 Failed, 0 Incomplete.
   0.01614 seconds testing time.
```



プロセス
エンジニア

4 システムへの統合

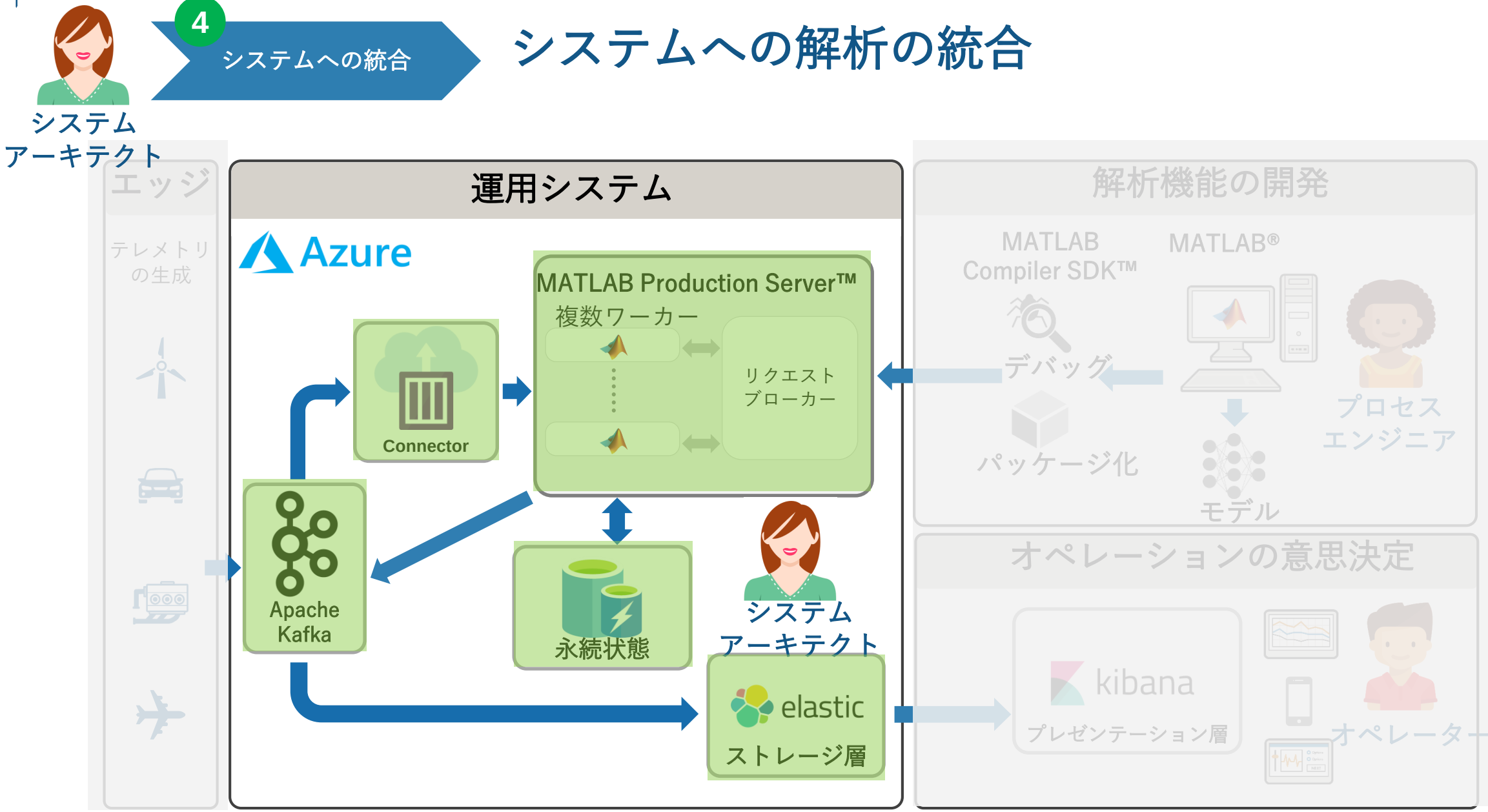
# ストリーム処理関数のパッケージ化



# 予知保全システム開発事例
## ～予知保全モデルのシステム化～

4 システムへの統合

# システムへの解析の統合



システム
アーキテクト

4 システムへの統合

# Kafkaを用いたストリーミングデータの入出力

## 入力ストリーム

| Event Time | Pump Id | Flow | Pressure | Current |
|---|---|---|---|---|
| 18:01:10 | Pump1 | 1975 | 100 | 110 |
| 18:10:30 | Pump3 | 2000 | 109 | 115 |
| 18:0… | | | | |
| 18:1… | | | | |
| 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 |
| … | … | … | … | … |

バッチ処理で作成したモデル
をストリーム処理に利用

状態 → MATLAB 関数 → 状態 → MATLAB 関数 → 状態 → MATLAB 関数 → 状態

## 出力ストリーム

| 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 |

システム
アーキテクト

4 システムへの統合

# ダッシュボードの作成

エッジ

テレメトリの生成

運用システム

Azure

MATLAB Production Server™

複数ワーカー

Connector

リクエスト
ブローカー

Apache
Kafka

永続状態

システム
アーキテクト

解析機能の開発

MATLAB
Compiler SDK™

MATLAB®

デバッグ

パッケージ化

モデル

プロセス
エンジニア

オペレーションの意思決定

elastic
ストレージ層

kibana
プレゼンテーション層

オペレーター

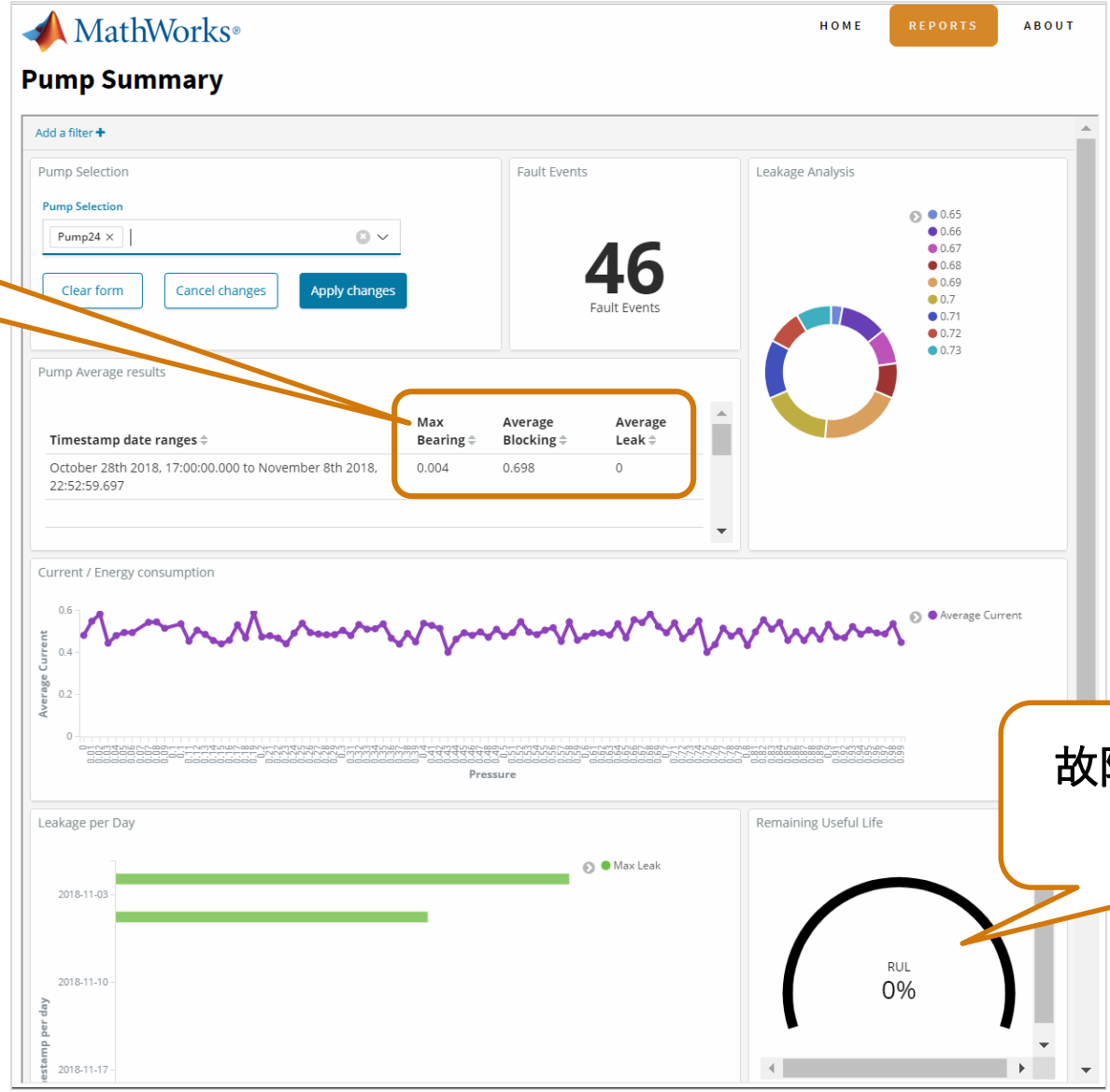
# 予知保全システム開発事例
## ～予知保全システムの利用～

プラント
オペレーター

5 結果の可視化

# ダッシュボードの利用

故障の種類を
予測




故障までの寿命の
予測

# まとめ

# Key Takeaways

- MATLABの**アプリを活用した**予知保全モデルの開発
- 実際に故障した際の機器のデータが無くても、
  Simulinkを用いて**故障した場合のシミュレーションデータ**を作成
- プロトタイプ作成から、ストリーミングプラットフォームKafkaとクラウドを用いたシステム化までを**３週間**で実現

# 参考：使用したToolbox

| 項目 | 使用した製品 |
| --- | --- |
| 予知保全のモデル化 | • MATLAB®<br>• Predictive Maintenance Toolbox™<br>• Signal Processing Toolbox™<br>• Statistics and Machine Learning Toolbox™ |
| ポンプのシミュレーション<br>データ作成 | • Simulink®<br>• Simscape™<br>• Simscape™ Electrical™<br>• Simscape™ Fluids™<br>• Simscape™ Multibody™<br>• Predictive Maintenance Toolbox<br>• Parallel Computing Toolbox™ |
| 予知保全モデルのシステム化 | • MATLAB® Compiler SDK™<br>• MATLAB Production Server™ |

# 参考：補足情報

- GitHub: MathWorks Reference Architectures
  https://github.com/mathworks-ref-arch
- MATLABとIT システムの連携
  https://jp.mathworks.com/solutions/enterprise-it-systems.html
- データサイエンス向けMATLAB
  https://jp.mathworks.com/solutions/data-science.html