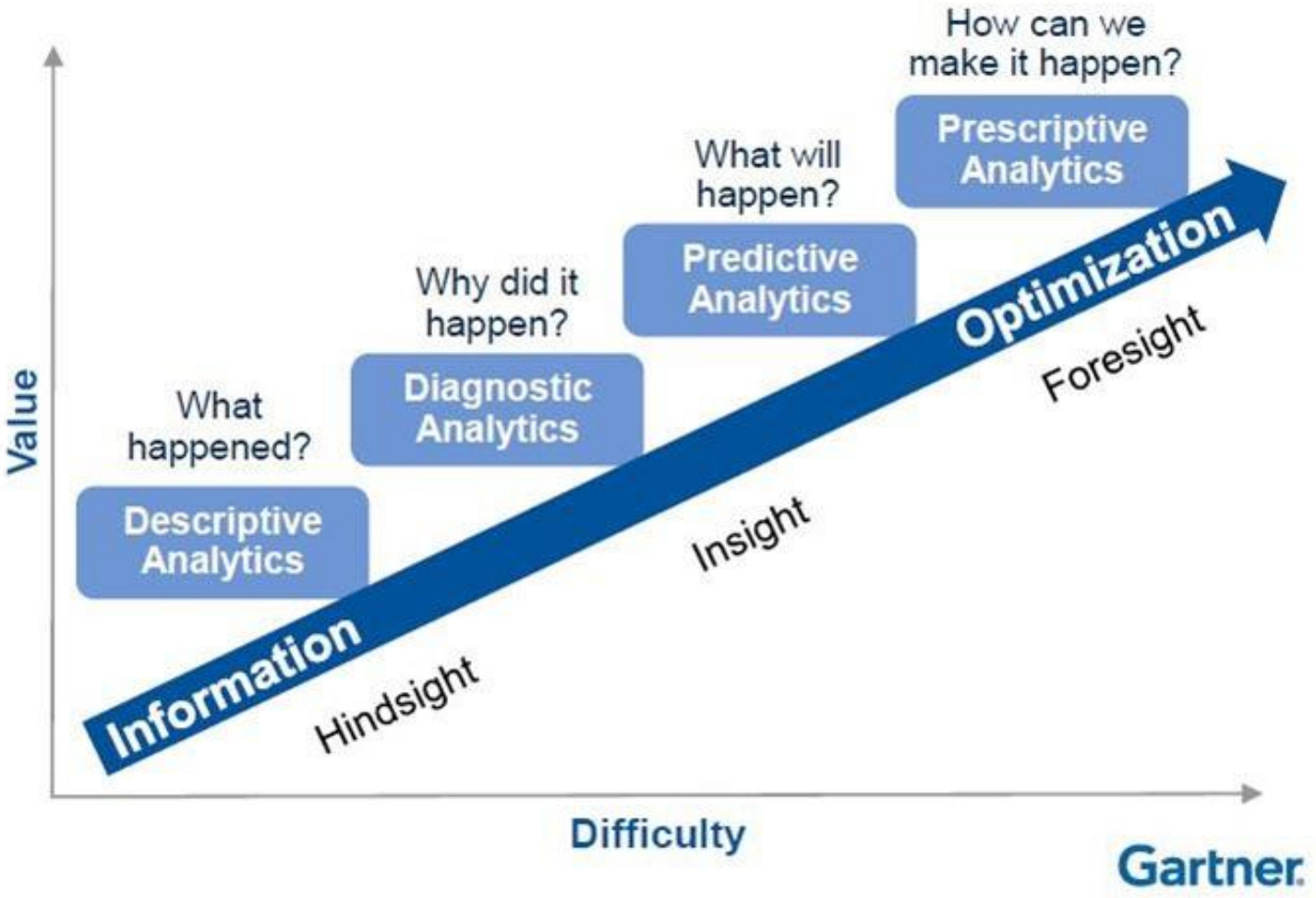
Prediktivní analytika s PI Systemem

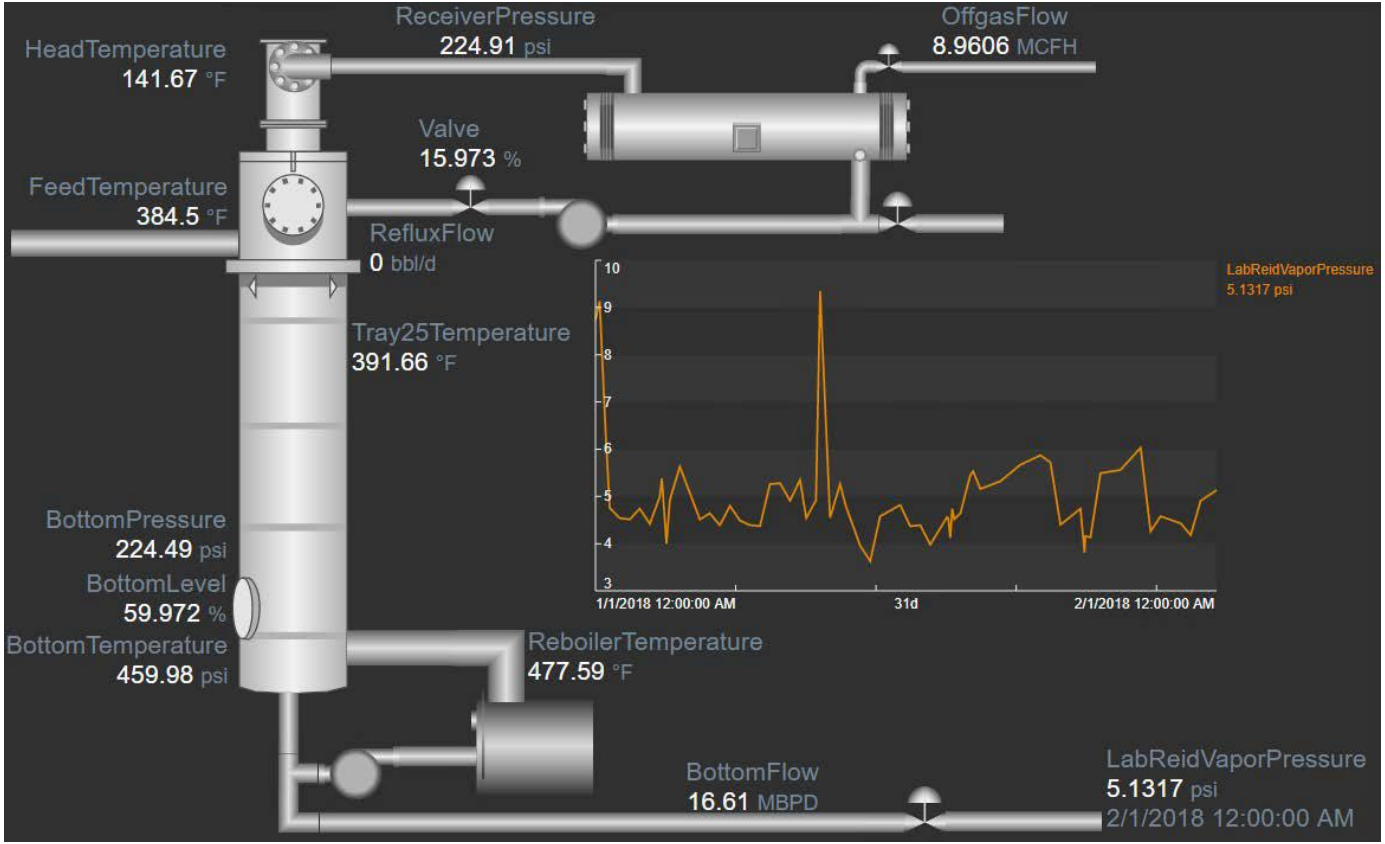
Vít Gruner

Tadeáš Marciniak

OSIsoft.



Problem Statement: Refinery Stabilization Column



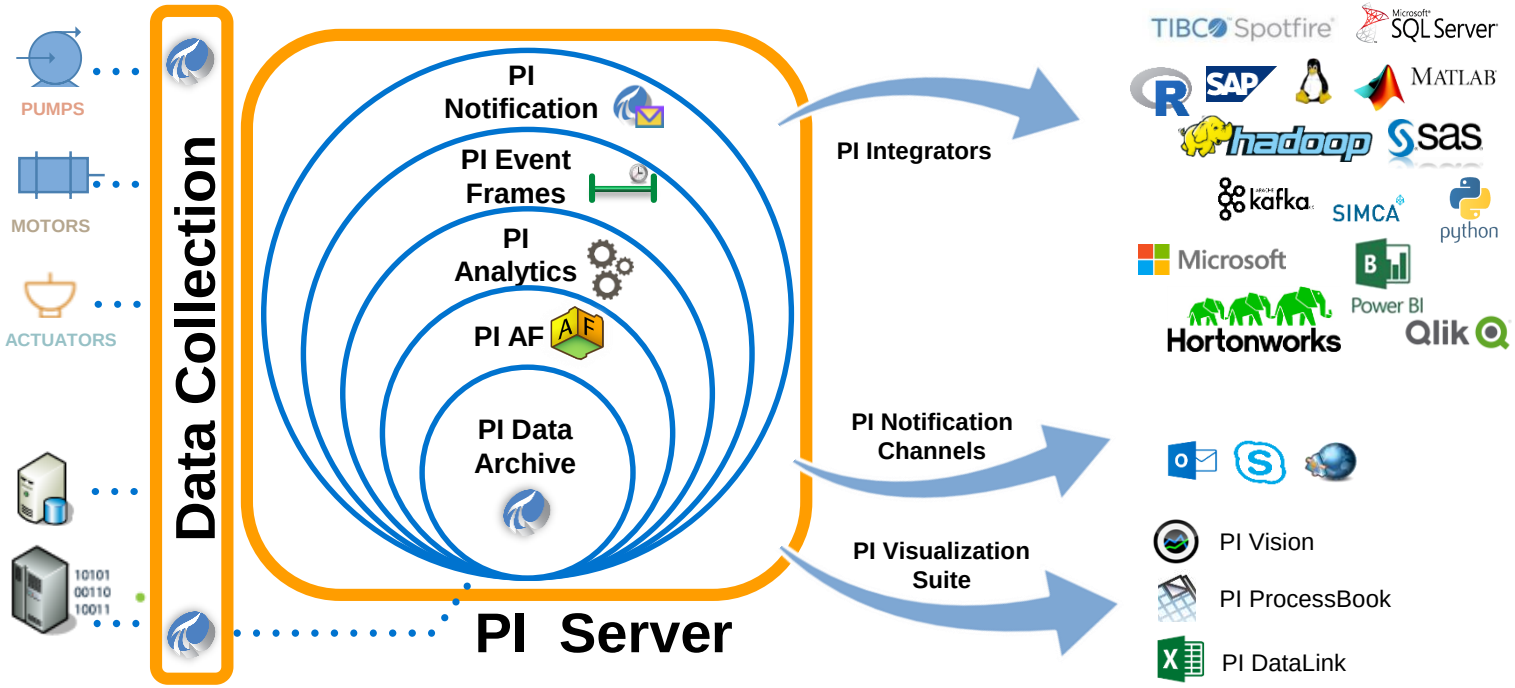
Problem Statement: Refinery Stabilization Column

- The product has to be stabilized by controlling its Reid Vapor Pressure (RVP)
- The stabilizer columns does not have on-line RVP analyzer
  - manual samples (taken twice a day) are analyzed by the laboratory
- More than 12 on-line measurements are collected each 6 minutes

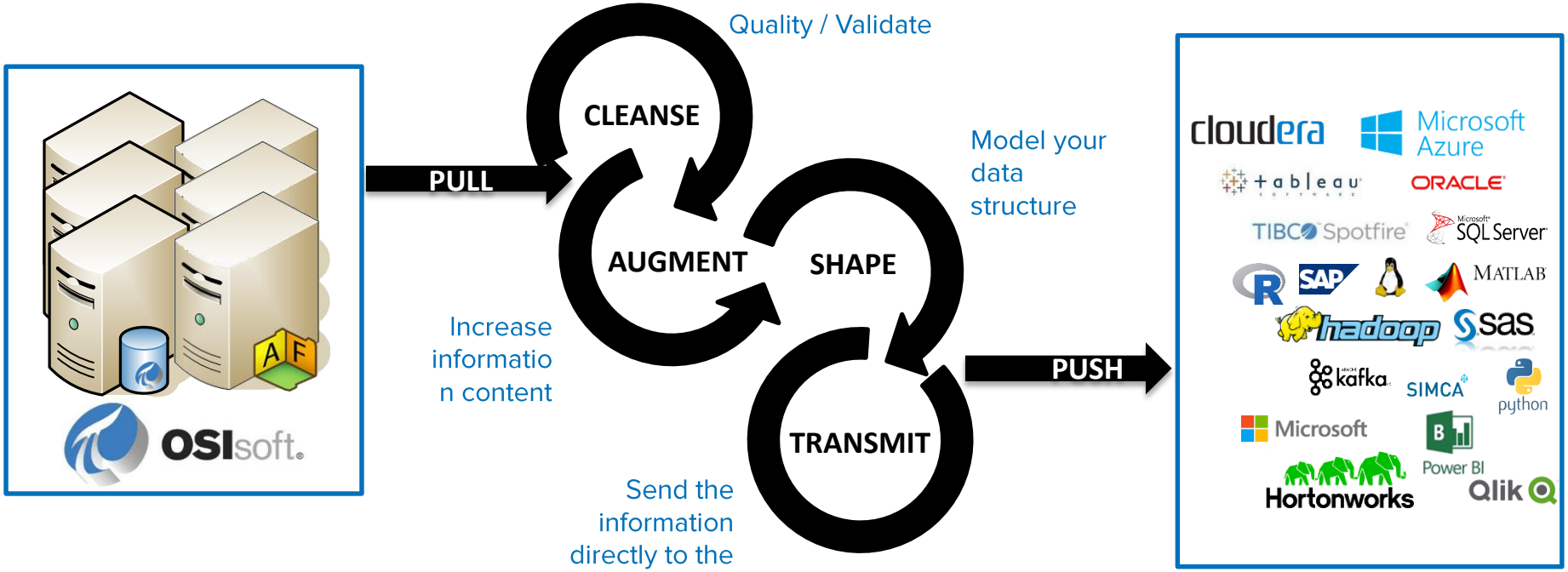
The goal:

- Use the process data collected in the PI System along with the laboratory measurements to generate a model predicting the RVP

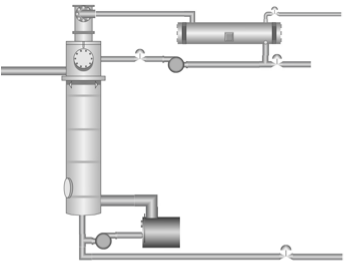
The PI System Infrastructure



PI Integrators



Refinery Stabilization Column – PI Architecture



Data Collection

PI Notification

PI Event Frames

PI Analytics

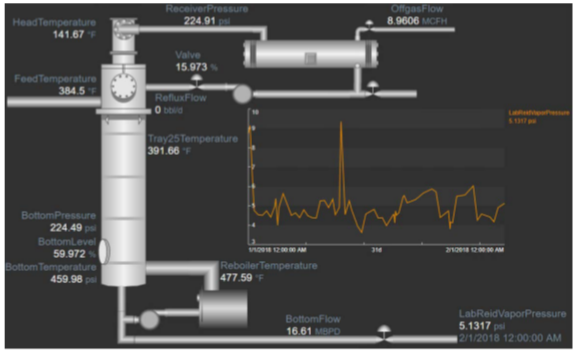
PI AF

PI Data Archive

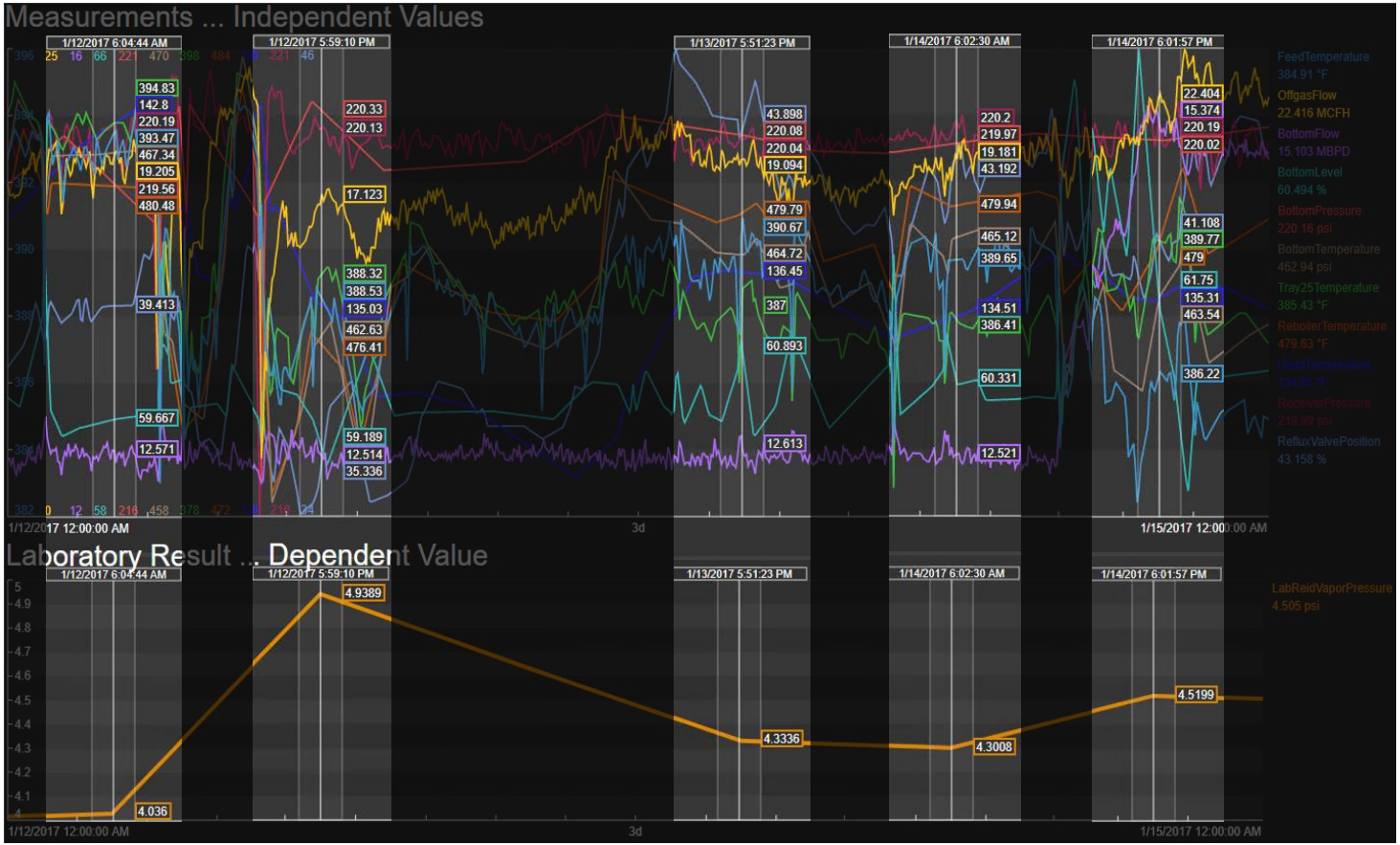
PI Server

PI Visualization Suite

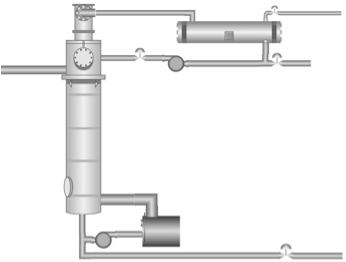
PI Vision



Machine Learning Model = Multivariate Function



Refinery Stabilization Column Predictive ML Models



Data Collection

PI Notification

PI Event Frames

PI Analytics

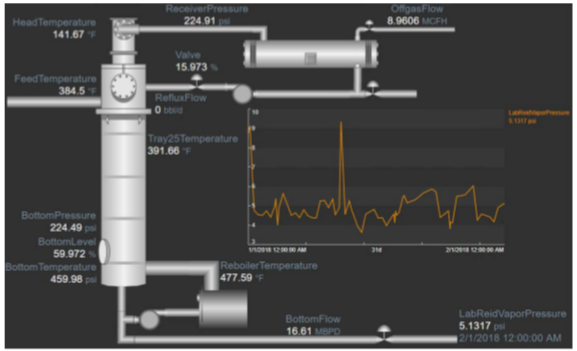
PI AF

PI Data Archive

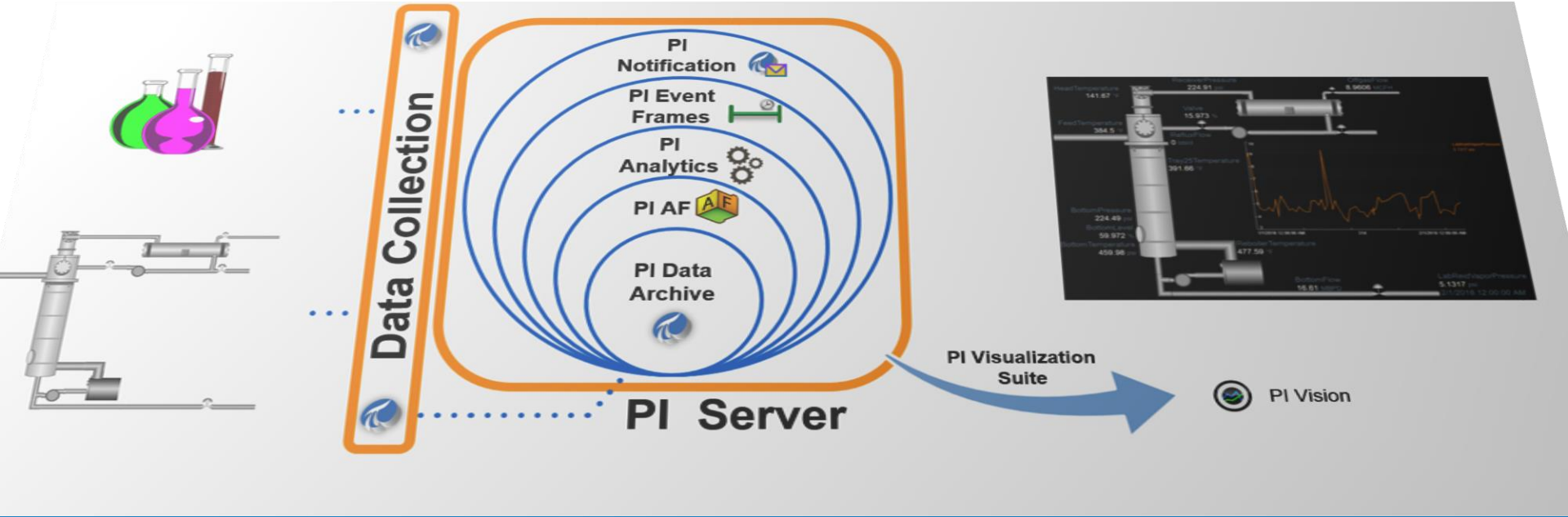
PI Server

PI Visualization Suite

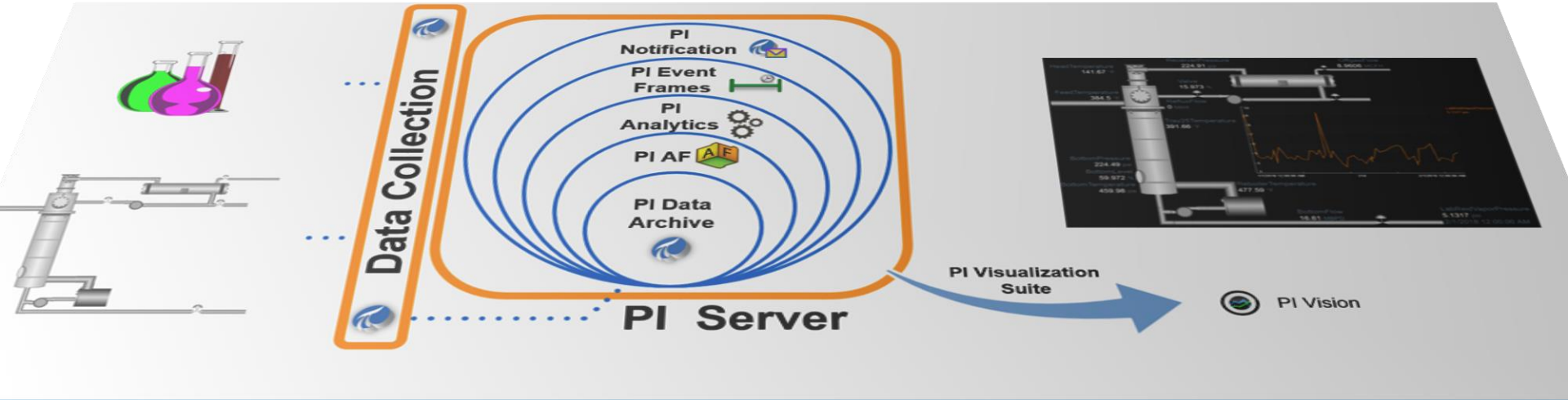
PI Vision



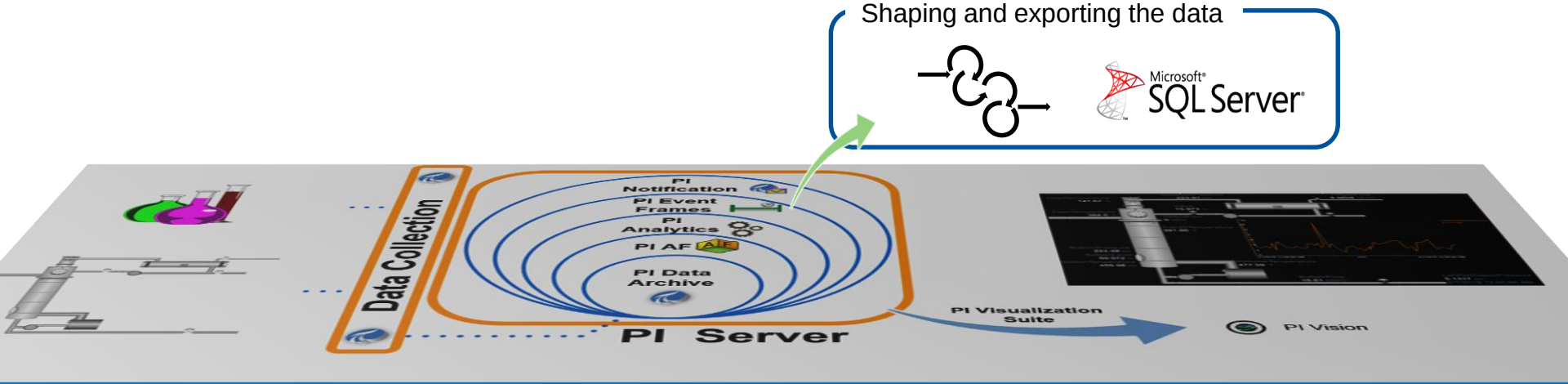
Refinery Stabilization Column Predictive ML Models



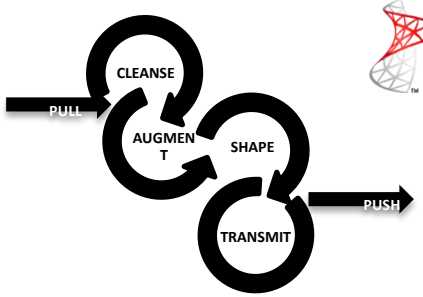
Refinery Stabilization Column Predictive ML Models



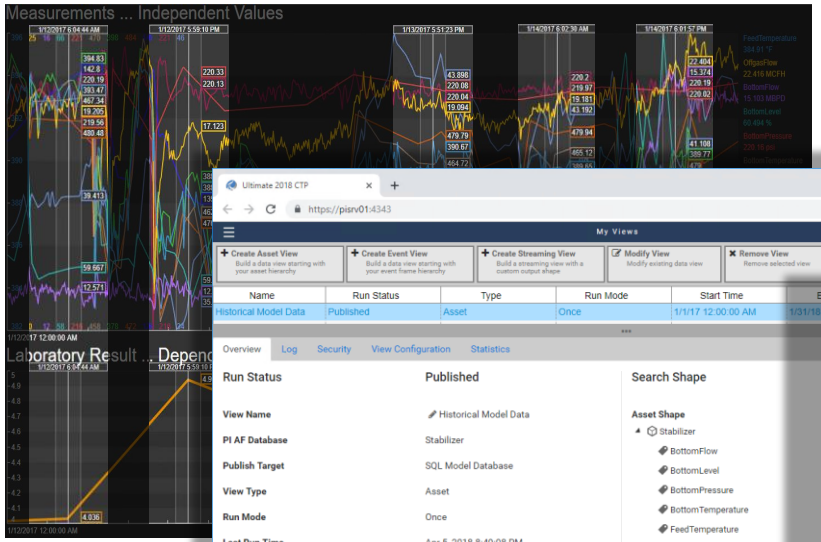
Refinery Stabilization Column Predictive ML Models



Shaping and exporting the data



Microsoft® SQL Server®



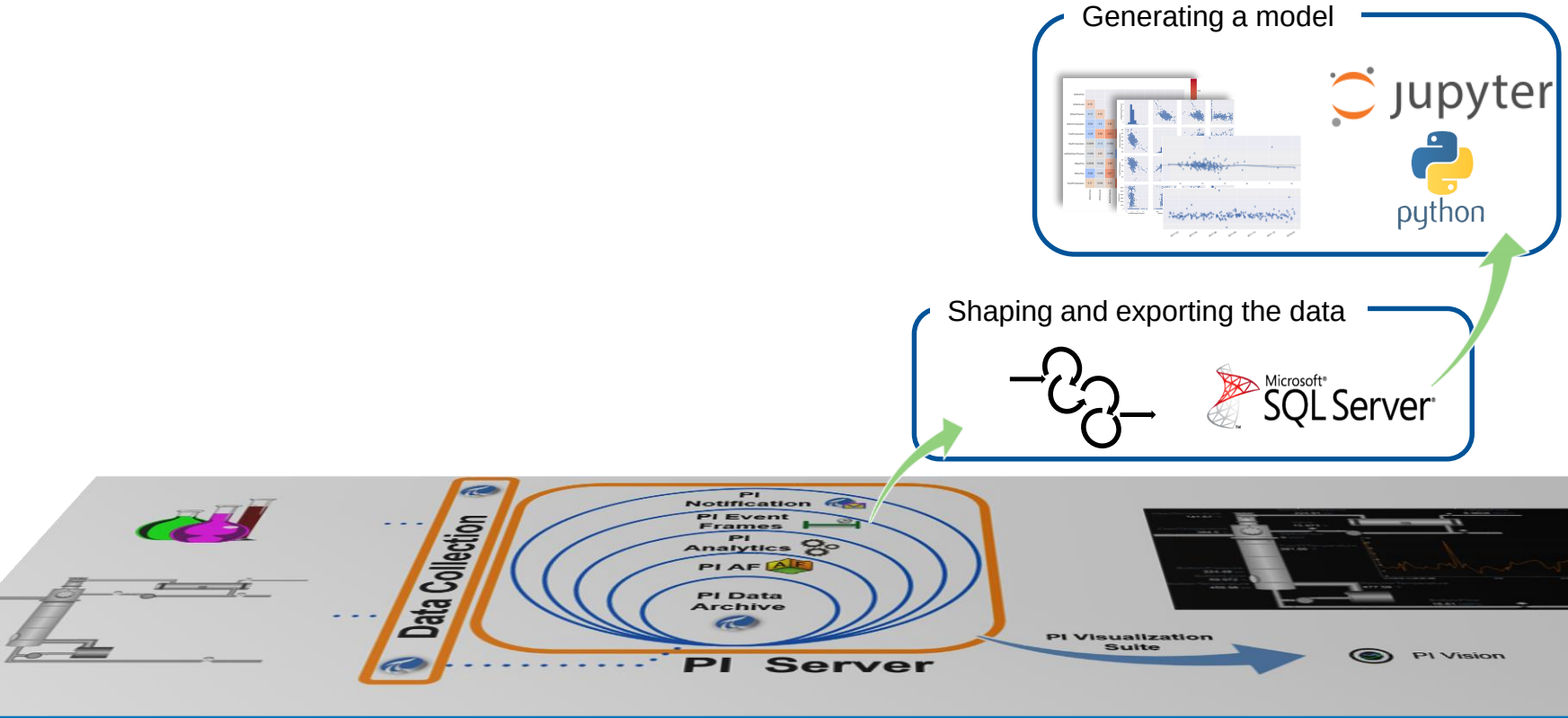

```
SELECT TOP 1000 [Id]
      ,[Stabilizer]
      ,[TimeStamp]
      ,[BottomFlow]
      ,[BottomLevel]
      ,[BottomPressure]
      ,[BottomTemperature]
      ,[FeedTemperature]
      ,[HeadTemperature]
      ,[OffgasFlow]
      ,[ReboilerTemperature]
      ,[ReceiverPressure]
      ,[RefluxFlow]
      ,[RefluxValvePosition]
      ,[Tray25Temperature]
      ,[LabReidVaporPressure]
      ,[PIIntTSTicks]
      ,[PIIntShapeID]
  FROM [ModelData].[dbo].[Historical Model Data]
```

100 % ▾

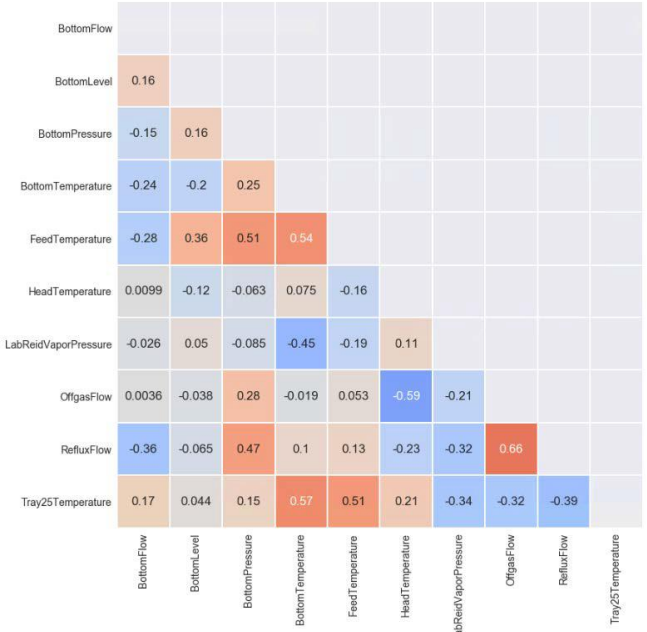
Results | Messages

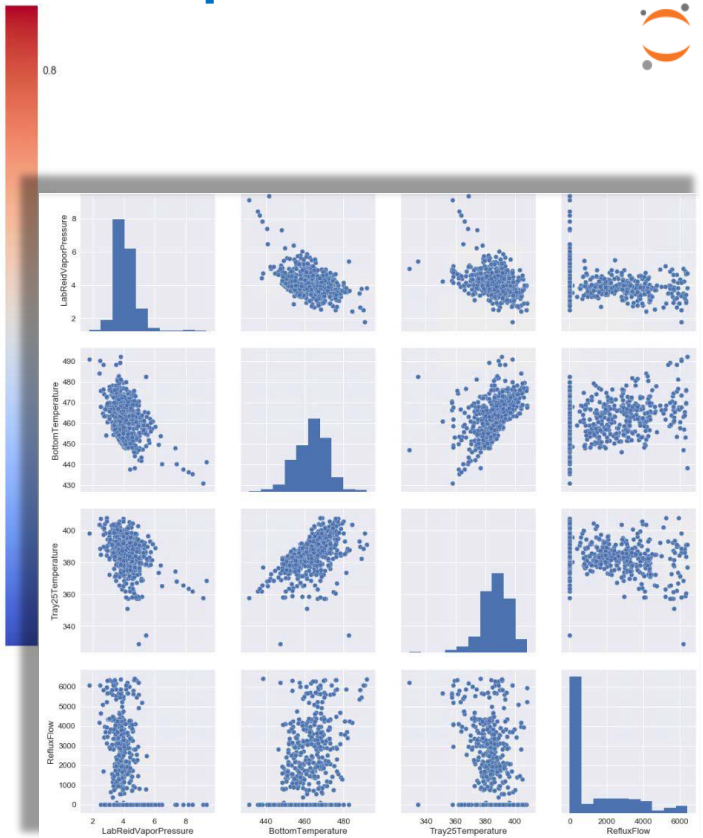
	Id	Stabilizer	TimeStamp	BottomFlow	BottomLevel	BottomPressure	BottomTemperature	FeedTemperature	HeadTemperature	OffgasFlow	ReboilerTemperature	ReceiverPressure
1	1	Stabilizer	2017-01-01 00:00:00.000	15.842270851135	59.196228027344	219.337173461914	474.98486328125	391.834045410156	273.827606201171	0	489.051330566406	220.084854125976
2	2	Stabilizer	2017-01-01 05:00:00.000	15.885624885559	59.916400909424	219.303970336914	473.329071044921	390.567321777343	278.574523925781	0	488.744567871093	219.991622924804
3	3	Stabilizer	2017-01-01 17:00:00.000	15.974317550659	60.005424499512	219.375656127929	476.650268554687	395.234008789062	285.700134277343	0	491.168548583984	220.089187622070
4	4	Stabilizer	2017-01-02 05:00:00.000	15.956583976746	60.170280456543	219.24119567871	473.580413818359	390.820220947265	304.855102539062	0	488.891845703125	220.061889648437
5	5	Stabilizer	2017-01-02 17:00:00.000	16.099704742432	60.614753723145	219.322891235351	477.060485839843	392.138671875	309.9111328125	0	491.012756347656	220.051177978515
6	6	Stabilizer	2017-01-03 05:00:00.000	16.084686275297	60.213829040527	219.376861572265	472.732940673828	388.80078125	308.165283203125	0	487.556884765625	220.022308349609
7	7	Stabilizer	2017-01-03 17:00:00.000	18.021993637085	59.679718017578	219.384857177734	468.795471191406	385.704498291015	337.381866455078	0	486.283538818359	219.95703125
8	8	Stabilizer	2017-01-04 05:00:00.000	17.819311141968	59.81600189209	219.308288574218	470.496032714843	385.128021240234	293.908905029296	0	488.33950805664	219.978088378906
9	9	Stabilizer	2017-01-04 17:00:00.000	18.046827316284	60.434307098389	220.0929602539	471.326843261718	386.597534179687	135.924972534179	15.770408630371	489.874267578125	219.831878662109
10	10	Stabilizer	2017-01-05 05:00:00.000	17.828824996948	59.848300933838	222.097885131835	470.463653564453	386.406951904296	149.846054077148	28.346687316895	489.171112060546	221.638473510742
11	11	Stabilizer	2017-01-05 17:00:00.000	18.001346588135	59.724563598633	220.756240844726	469.090118408203	385.562371826171	141.367645263671	26.619531631469	489.484771728515	220.731796264648
12	12	Stabilizer	2017-01-06 05:00:00.000	17.930227279663	59.714340209960	220.509017944335	469.944671630859	384.742736816406	135.592086791992	26.178880691528	491.318115234375	220.340022460937
13	13	Stabilizer	2017-01-06 17:00:00.000	18.015043258667	60.018531799316	220.488983154296	472.224884033203	387.214324951171	139.747817993164	25.742961883544	493.324859619140	219.599609375
14	14	Stabilizer	2017-01-07 05:00:00.000	17.901599884033	60.599925994873	220.731796264648	470.341064453125	385.073730468750	133.522094726562	23.684286117554	491.268829345703	219.652481079101

Refinery Stabilization Column Predictive ML Models



Generating a model – Data Preparation





Generating a Linear Regression Model

Model Training

```
from sklearn.linear_model import LinearRegression
linReg = LinearRegression()

linReg.fit(X_train, y_train)
```

	LinReg
const	21.643687
BottomFlow	-0.040710
BottomLevel	-0.032544
BottomPressure	0.030519
BottomTemperature	-0.018783
FeedTemperature	0.014180
HeadTemperature	0.004406
OffgasFlow	0.019082
RefluxFlow	-0.000348
Tray25Temperature	-0.048495

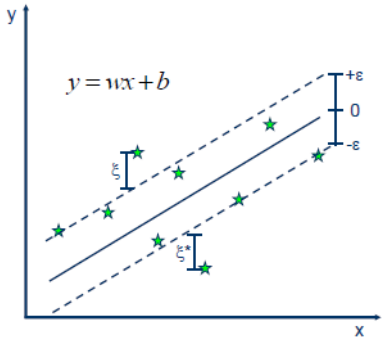
LinReg Model:

RVP = 21.64 +
-0.041*BottomFlow +
-0.033*BottomLevel +
0.031*BottomPressure +
...

Generating a Support Vector Regression Model

Support Vector Regression

Image Source



Fit as many instances as possible on the street (wx+b

- Minimize:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

- Constraints:

$$y_i - wx_i - b \le \varepsilon + \xi_i$$

$$wx_i + b - y_i \le \varepsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \ge 0$$

Modeling

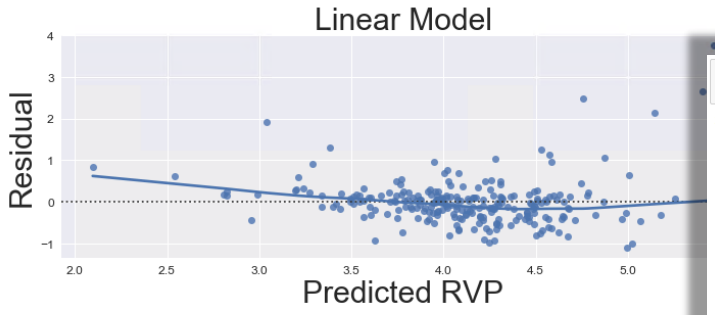
```
from sklearn.svm import SVR
svrModel = SVR(kernel='rbf')
svrModel.fit(X_train, y_train)
```

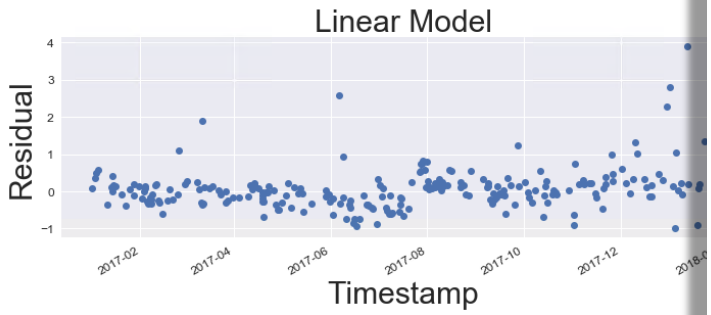
```
SVR(C=1.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma='auto',
  kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)
```

Model Verification

```
evaluateModel(X_test, y_test, linReg, 'Linear')
```

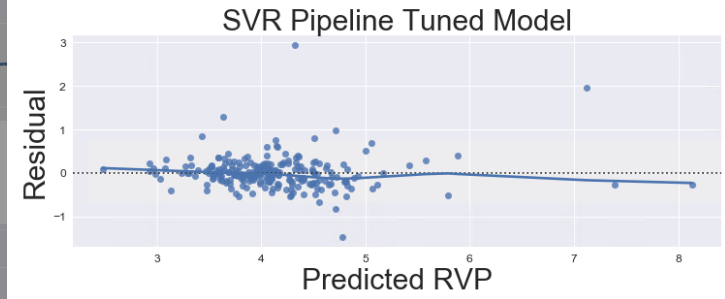
```
Linear Model RMSE = 0.5648484654589196
```

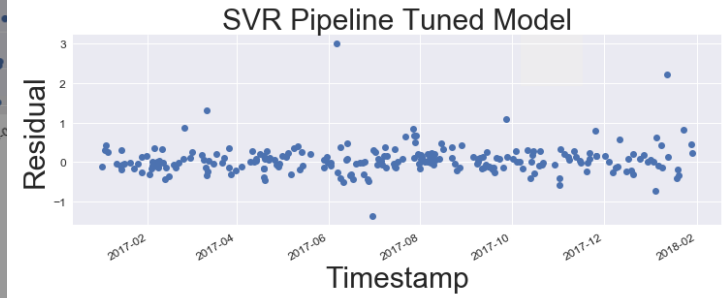




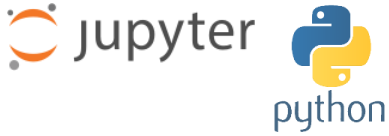

```
svrPipelineTuned = svrPipelineCV.best_estimator_
evaluateModel(X_test, y_test, svrPipelineTuned, 'SVR Pipeline Tuned')
```

```
SVR Pipeline Tuned Model RMSE = 0.3770256681670012
```





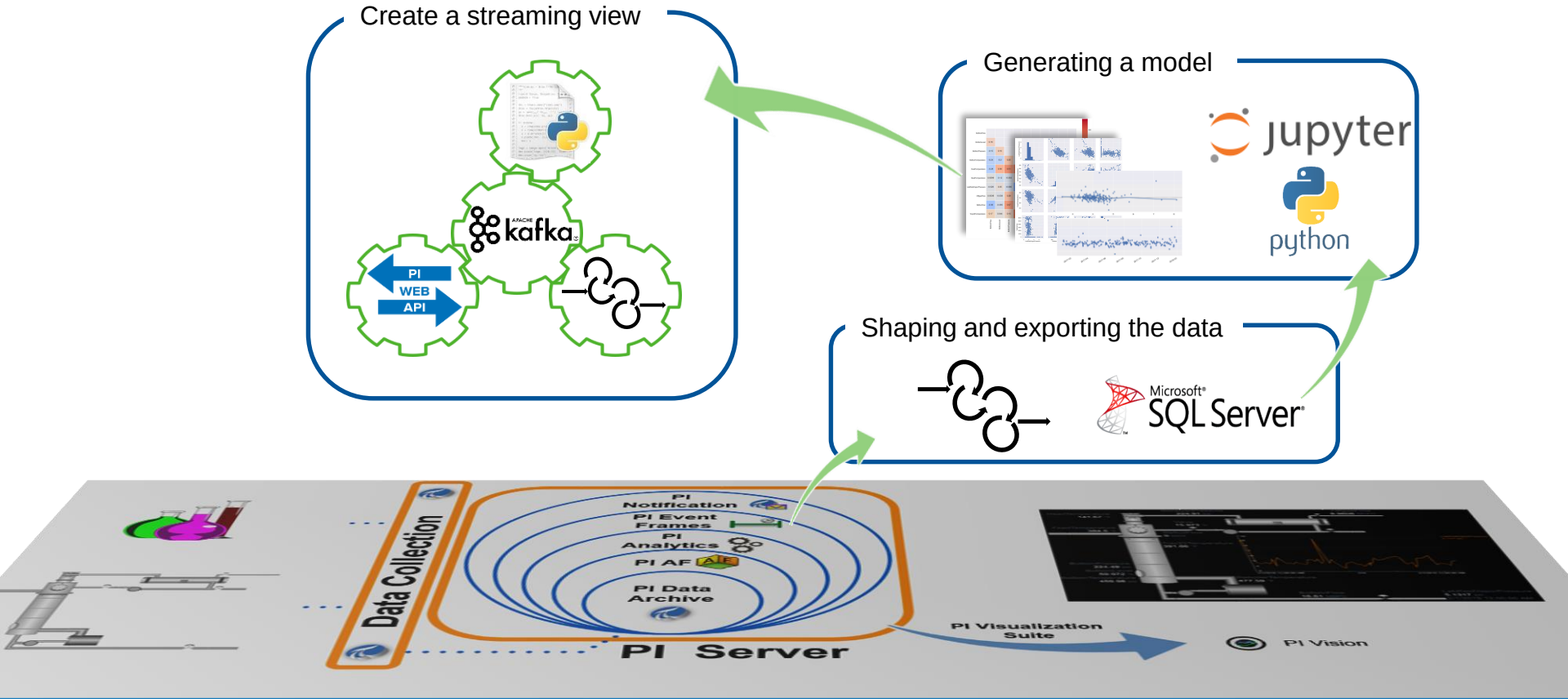
Create a Streaming View



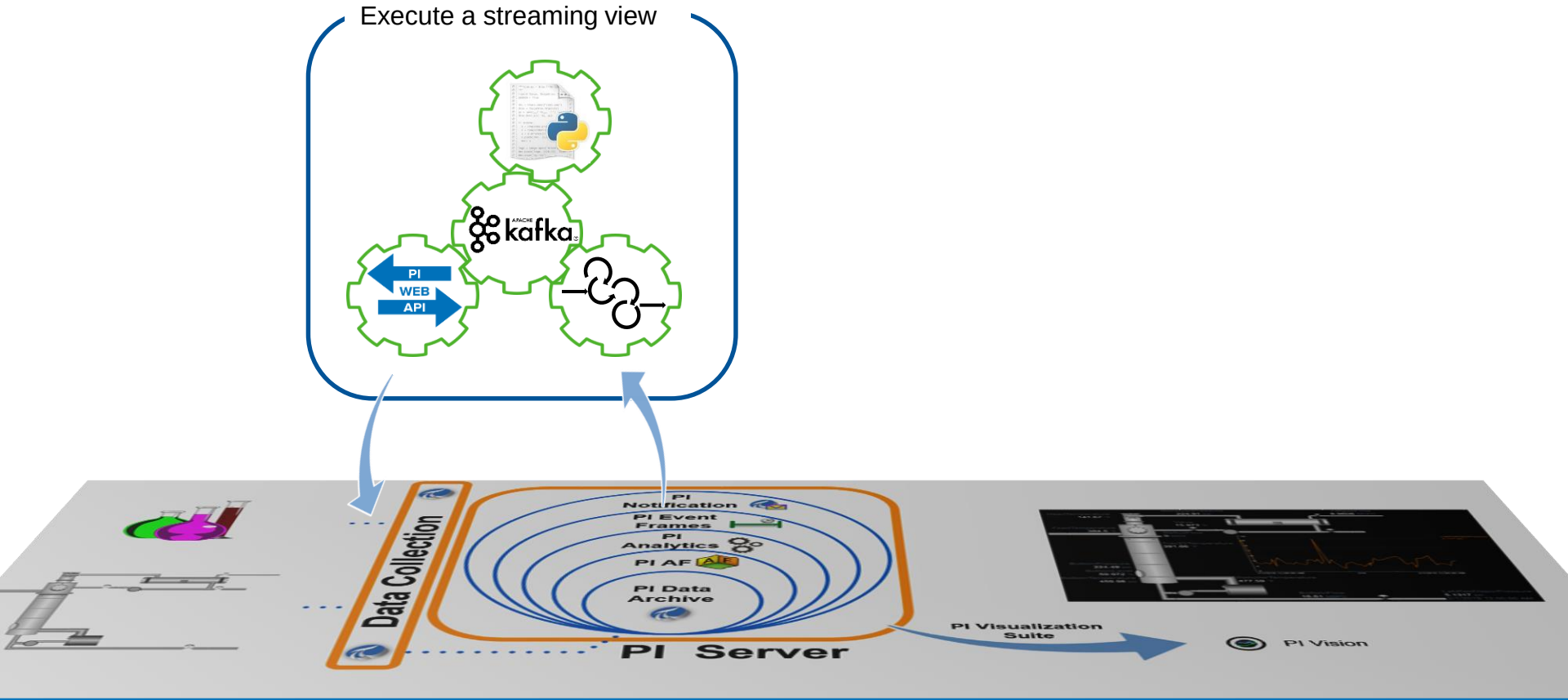
Streaming

```
from requests.packages import urllib3
urllib3.disable_warnings()
import requests
from requests_kerberos import HTTPKerberosAuth, OPTIONAL
import base64

def writeValuesToPI(time, linReg_prediction, svr_prediction):

    timeISO8601 = time[:19] + 'Z'


    elementPath = 'PISRV01\Stabilizer\Houston\Stabilizer'
    linPredPath = 'PISRV01\Stabilizer\Houston\Stabilizer|LinReg_RVP'
    svrPredPath = 'PISRV01\Stabilizer\Houston\Stabilizer|SVR_RVP'


    elementWebId = 'P1Em' + base64.b64encode(elementPath.encode()).decode('ascii')
    svrPredWebId = 'P1AbE' + base64.b64encode(svrPredPath.encode()).decode('ascii')
    linPredWebId = 'P1AbE' + base64.b64encode(linPredPath.encode()).decode('ascii')

    url = f'https://pisrv01/piwebapi/streamsets/{elementWebId}/value?webIDType=PathOnly'

    body = [
      {
        "WebId": svrPredWebId,
        "Value": {
          "Timestamp": timeISO8601,
          "Value": svr_prediction
        }
      },
      {
        "WebId": linPredWebId,
        "Value": {
          "Timestamp": timeISO8601,
          "Value": linReg_prediction
        }
      }
    ]

    r = requests.post(
        url,
        str(body),
        auth=HTTPKerberosAuth(mutual_authentication=OPTIONAL),
        verify=False
    )

    print(f'PI Web API Response Status: {r.status_code}')
```

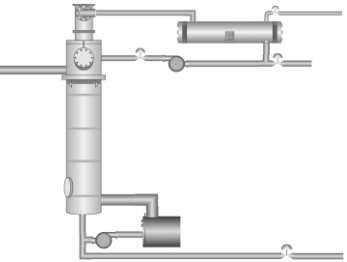
Refinery Stabilization Column Predictive ML Models



Refinery Stabilization Column Predictive ML Models



Refinery Stabilization Column Predictive ML Models



Data collection

PI Notification

PI Event Frames

PI Analytics

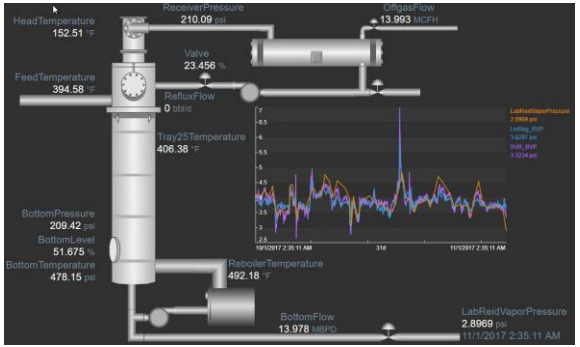
PI AF

PI Data Archive

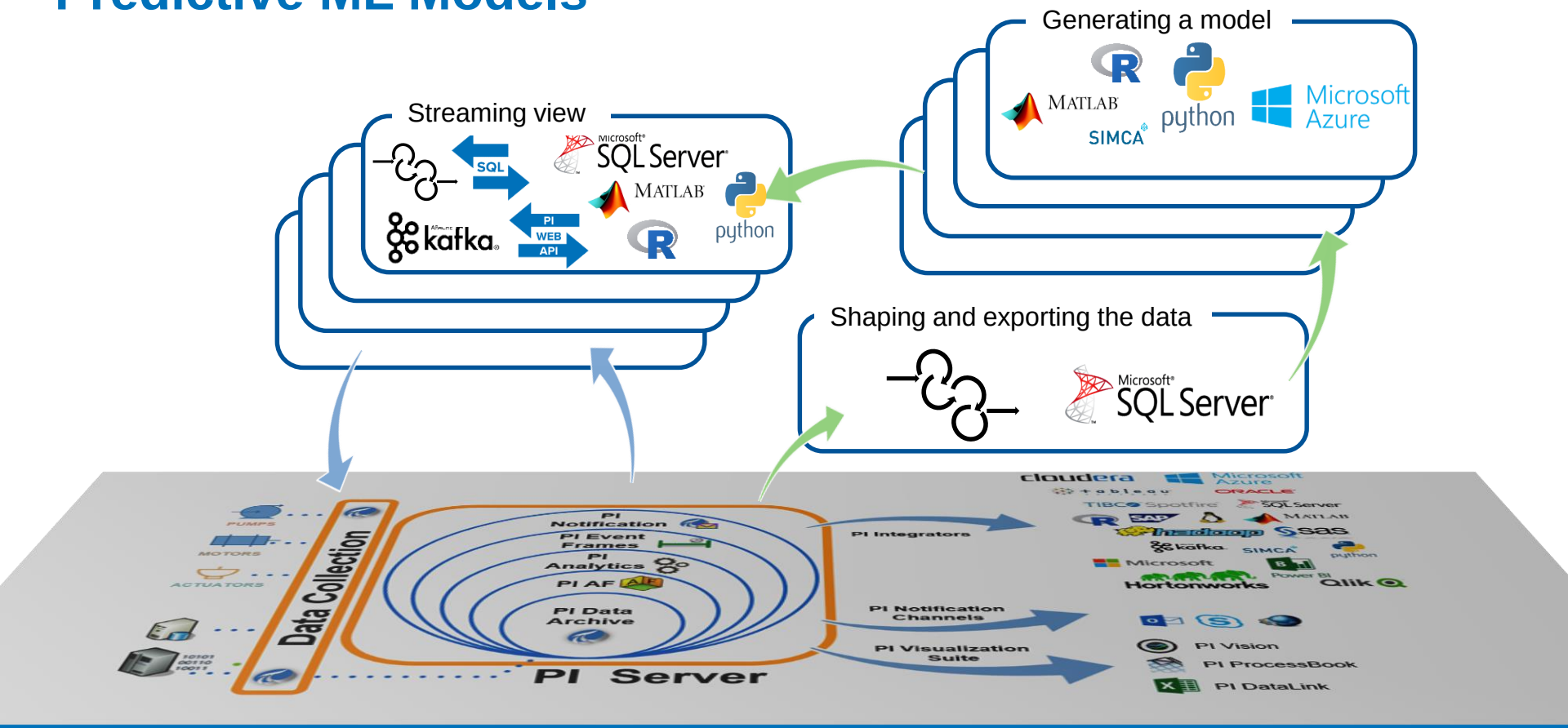
PI Server

PI Visualization Suite

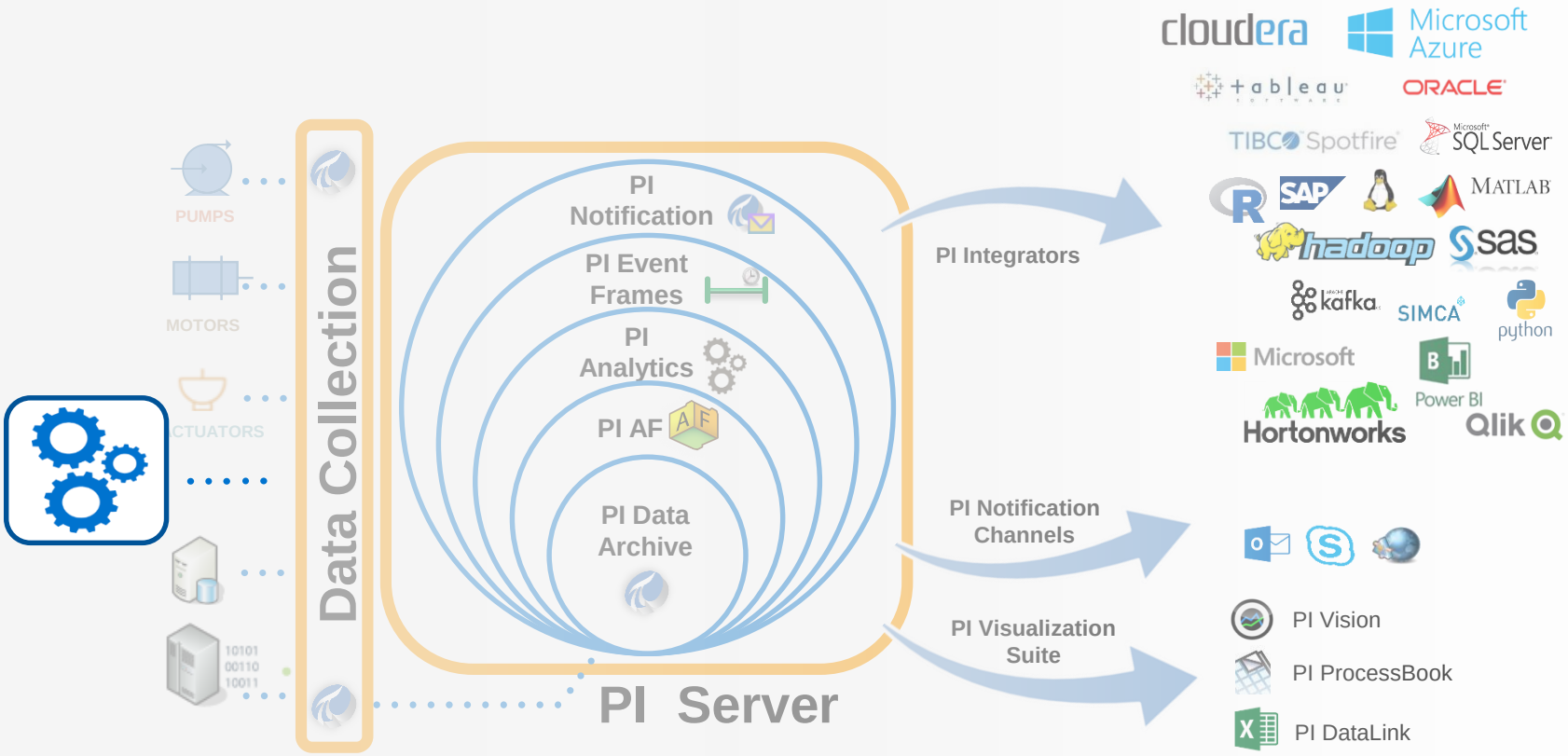
PI Vision



Predictive ML Models



ML Models in PI System Infrastructure



Examples

Deschutes Brewery - Brewing Beer the Smart Way
(https://www.osisoft.com/Presentations/Brewing-Beer-the-Smart-Way/)

Boehringer Ingelheim - A Case Study in Biopharmaceutical Data Analytics
(https://www.osisoft.com/Presentations/A-Case-Study-in-Biopharmaceutical-Data-Analytics--Using-Asset-Framework-and-Event-Frames-for-MVDA/)

Vitens N.V. - Data Science with R and the PI System
(https://www.osisoft.com/Presentations/Data-Science-with-R-and-the-PI-System/)

Using MATLAB with PI System for Analysis and Process Monitoring
(https://www.mathworks.com/videos/using-matlab-with-pi-system-for-analysis-and-process-monitoring-81859.html)

Challenges

- Pharma, food & beverage - prediction of fermentation performance
- Chemistry - prediction of reaction outcomes
- Metallurgy - liquid metal composition
- All industries – condition based maintenance (degradation curve prediction)
- …

Thank You