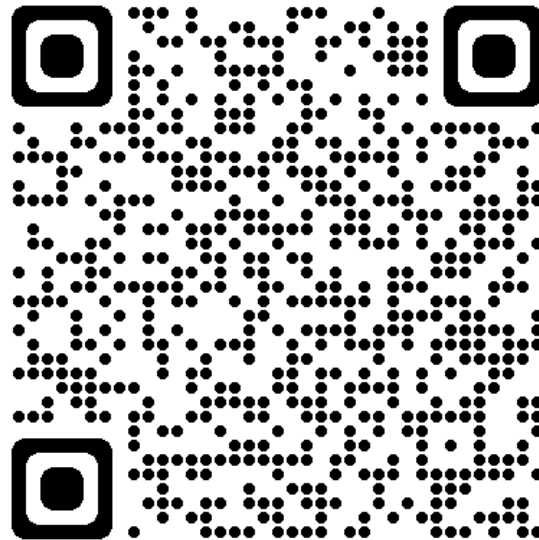# How to model stochastic behavior of failures in telco or IT systems using machine learning?
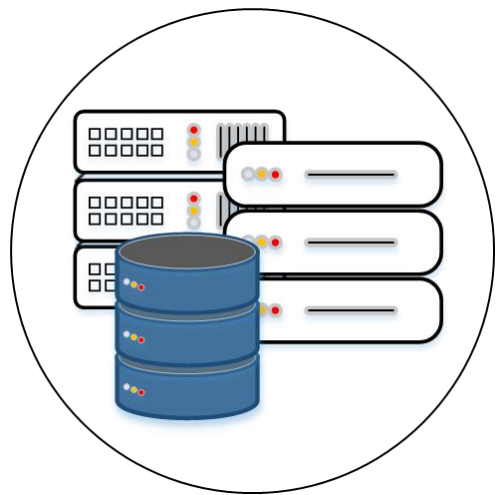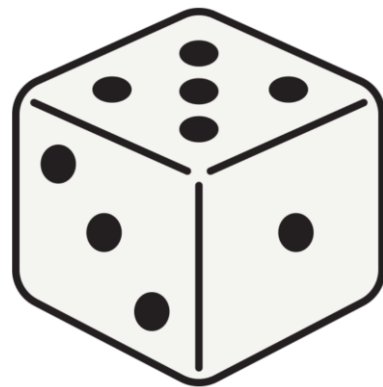
Sadegh Karimi

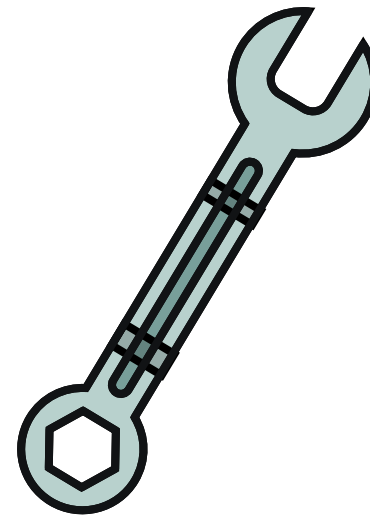# Hardware, estimation & maintenance

AirFrame Open Edge Server

AirScale baseband

Problem definition

Corrective Maintenance (CM)

- Income loss

- Reputation damage

- Less resource planning

- Expensive repairs

- Longer downtime

Preventive Maintenance (PM)

- Prevent income loss

- Prevent reputation damage

- More resource planning

- Prevent expensive repairs

- Shorter downtime
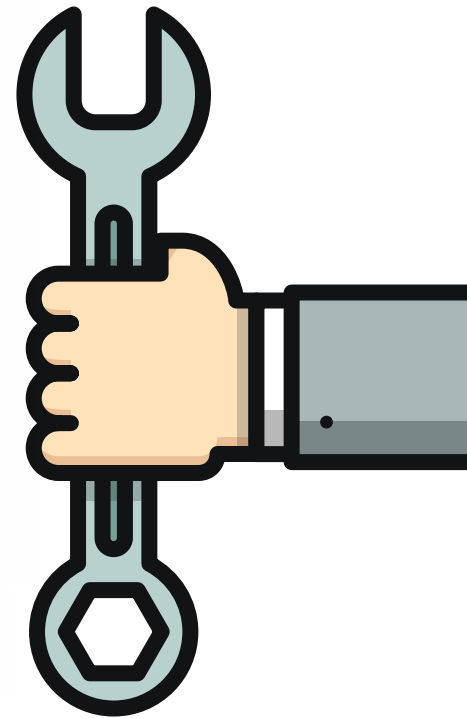
Problem definition

Conventional Method

- Failure happens, then perform the fix

- Perform PM on regular basis

- CM is playing active role

network   data/alarms   analysis   CM
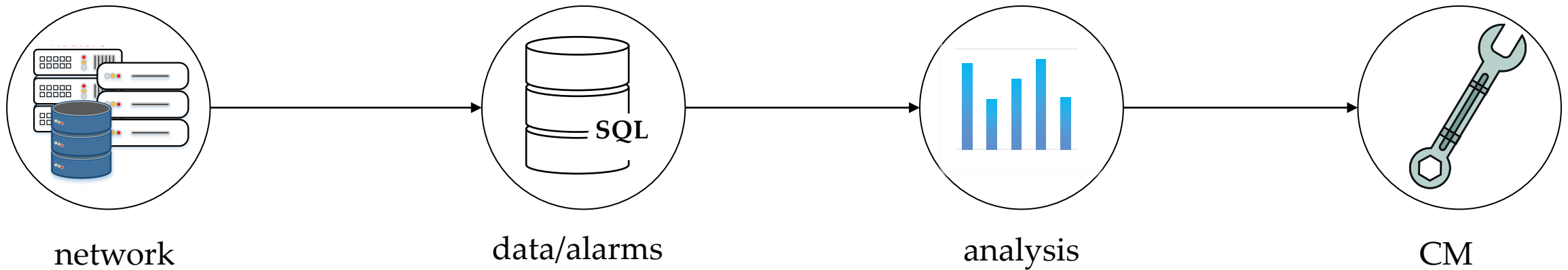
Problem definition

This method:

- Anticipate the failure, fix it before happens

- PM is playing active role

network   data/alarms   analysis   estimation   model   PM
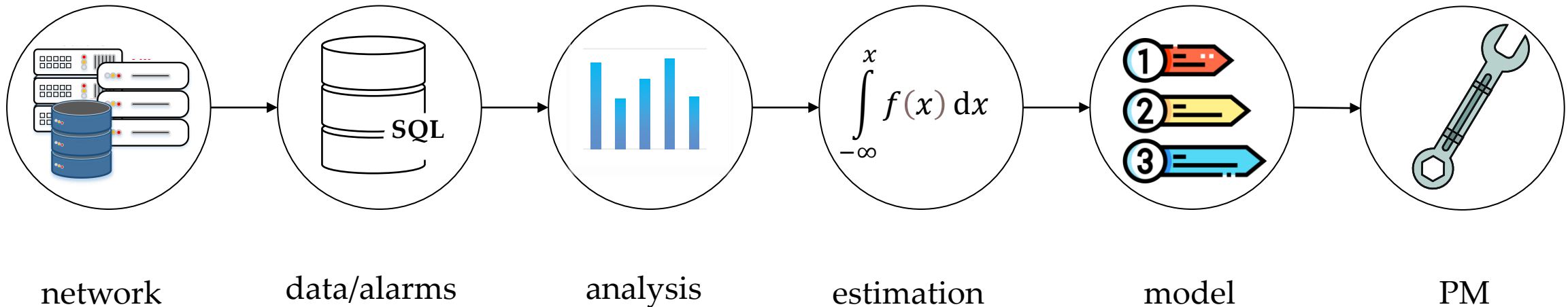
Roadmap

1 **Data collection & system design**

2 **Statistic Analysis**

3 **Estimation of failure**

4 **Modeling**
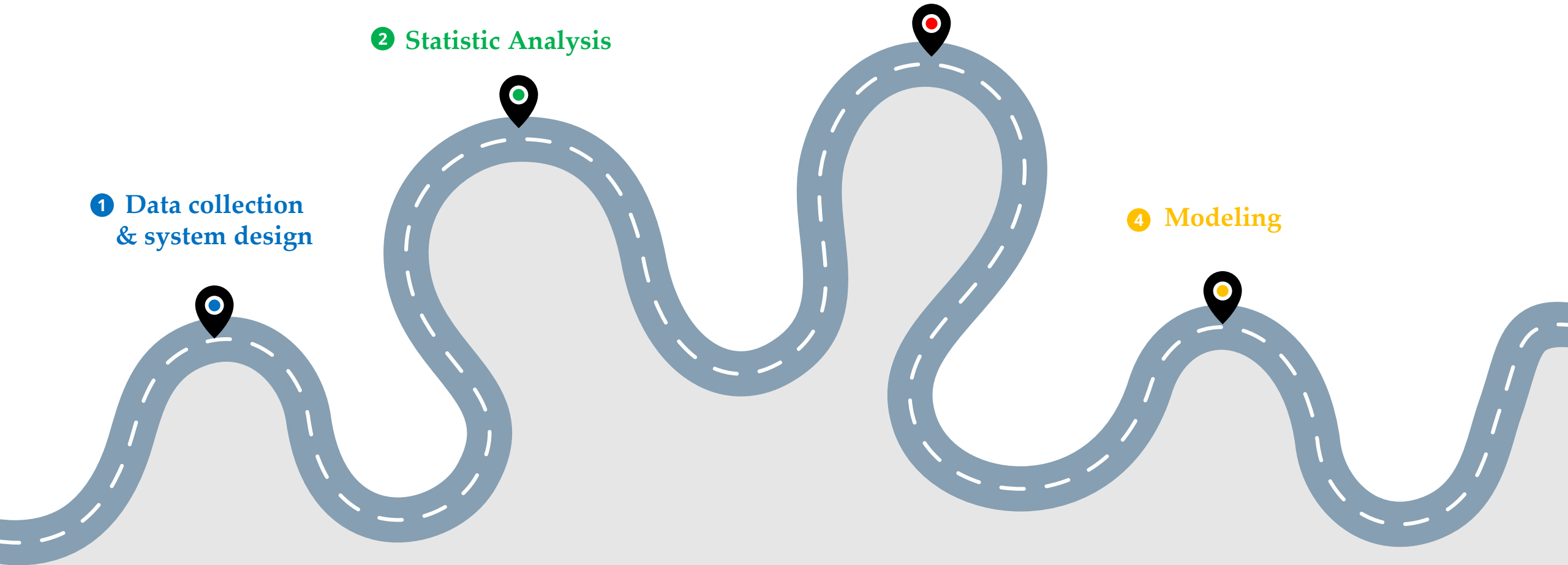
Data collection
& system design

Data Collection

Method: **Prometheus**

node_exporter

any_scraper

S

mtail

0111010001

1110010011

Monitoring Server

Prometheus

0110110101

0110110101
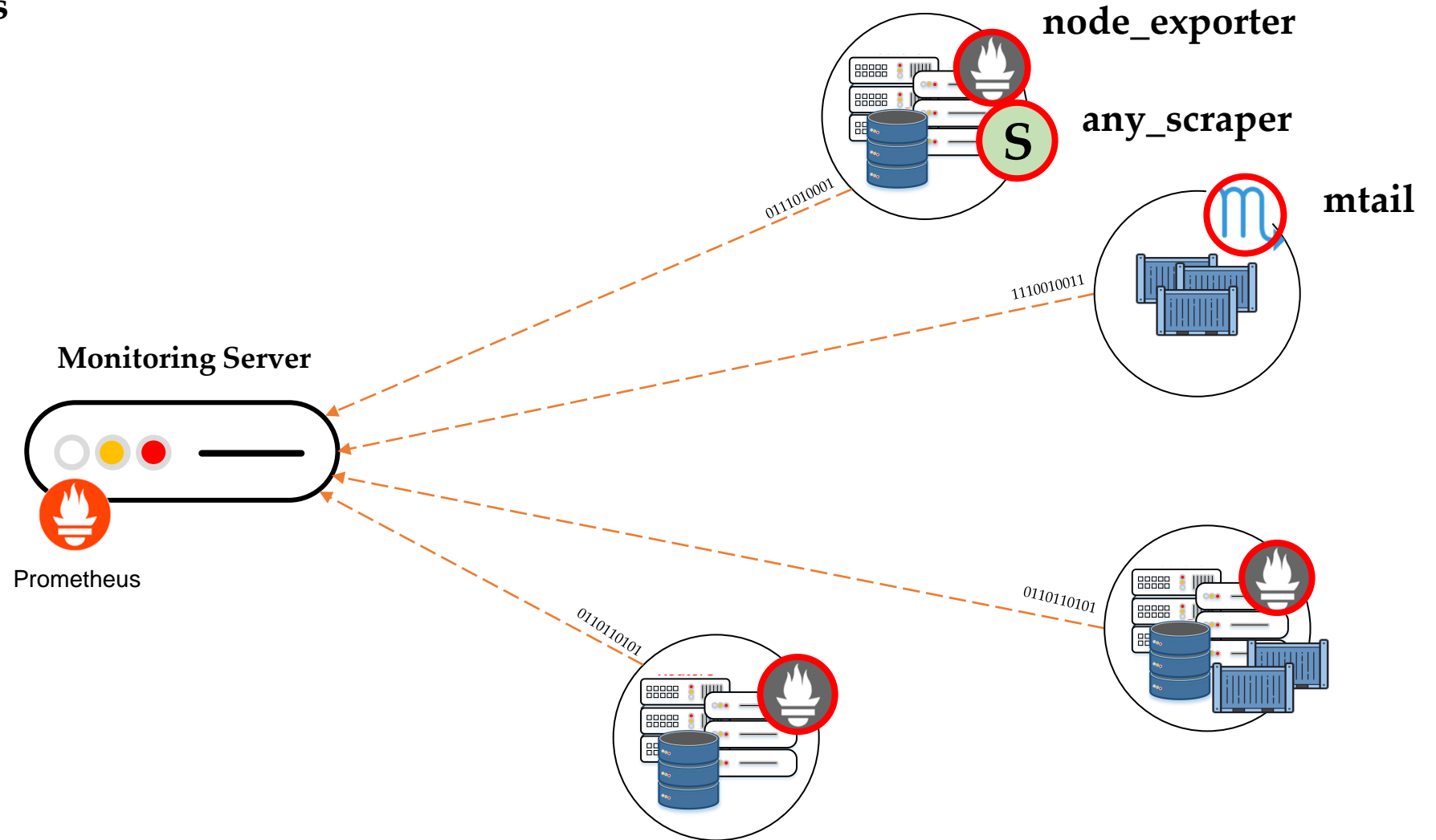
Data Collection

Method: **Network Monitoring Server (NMS)**
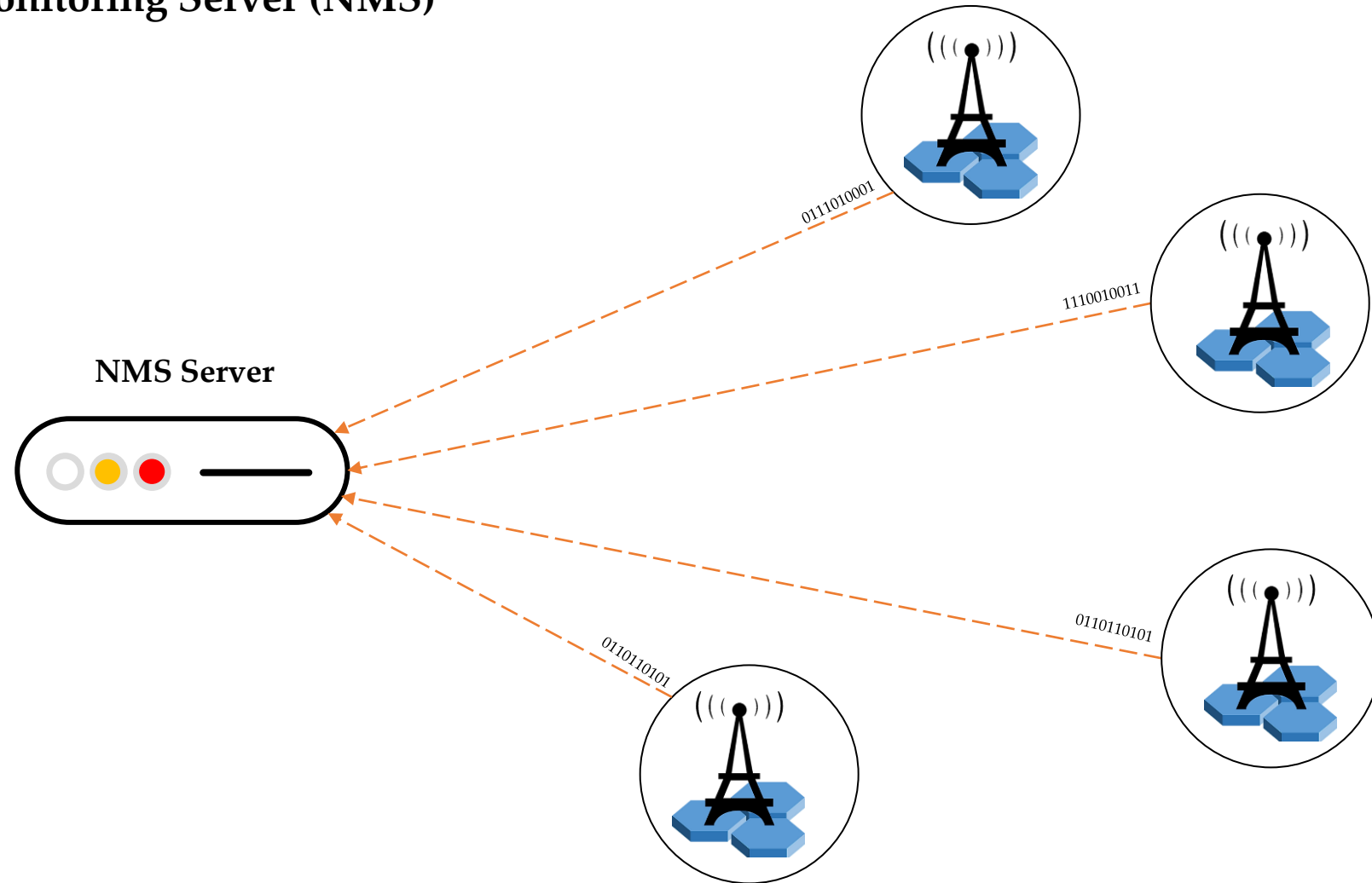
What is Prometheus?

What is Prometheus?
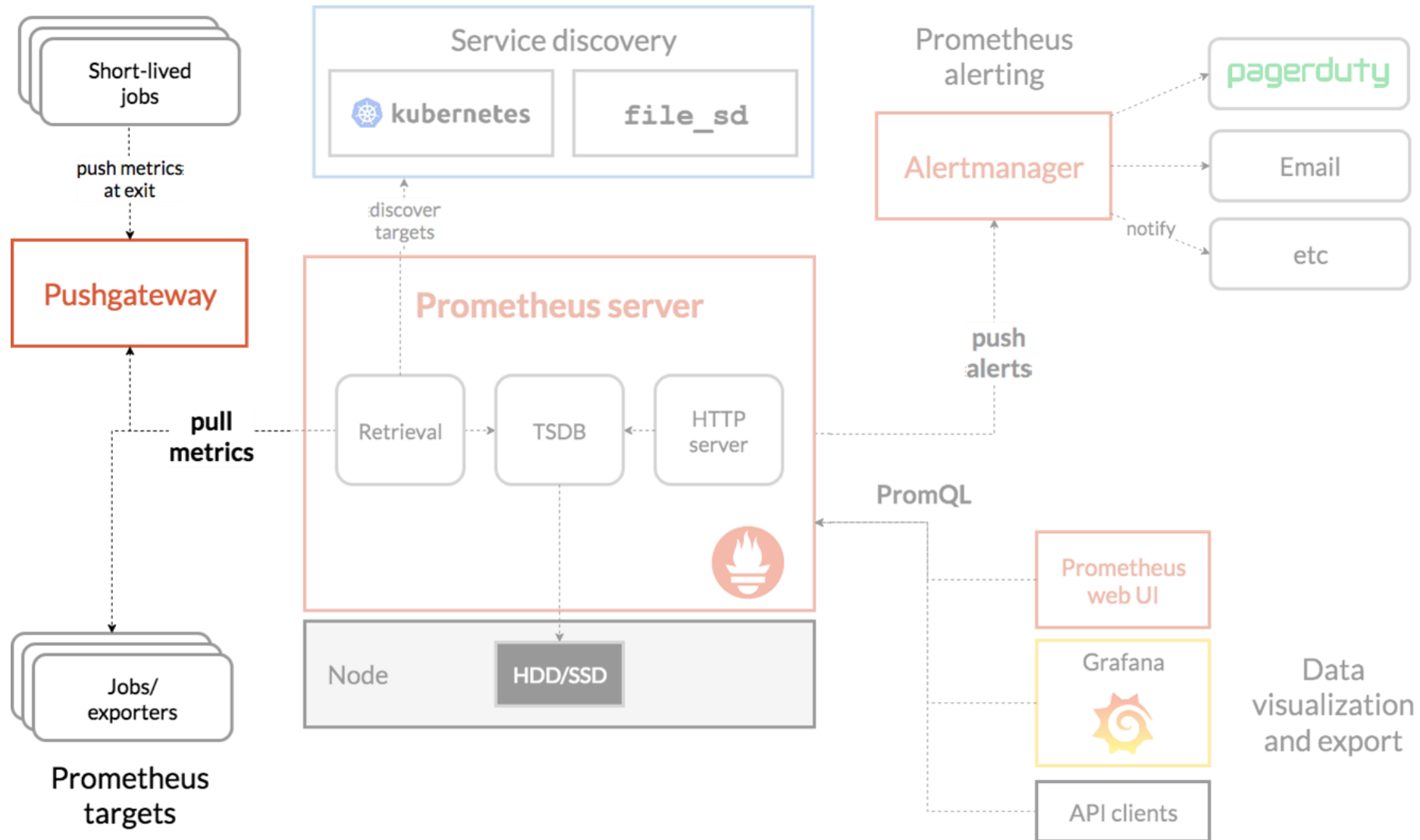
What is Prometheus?
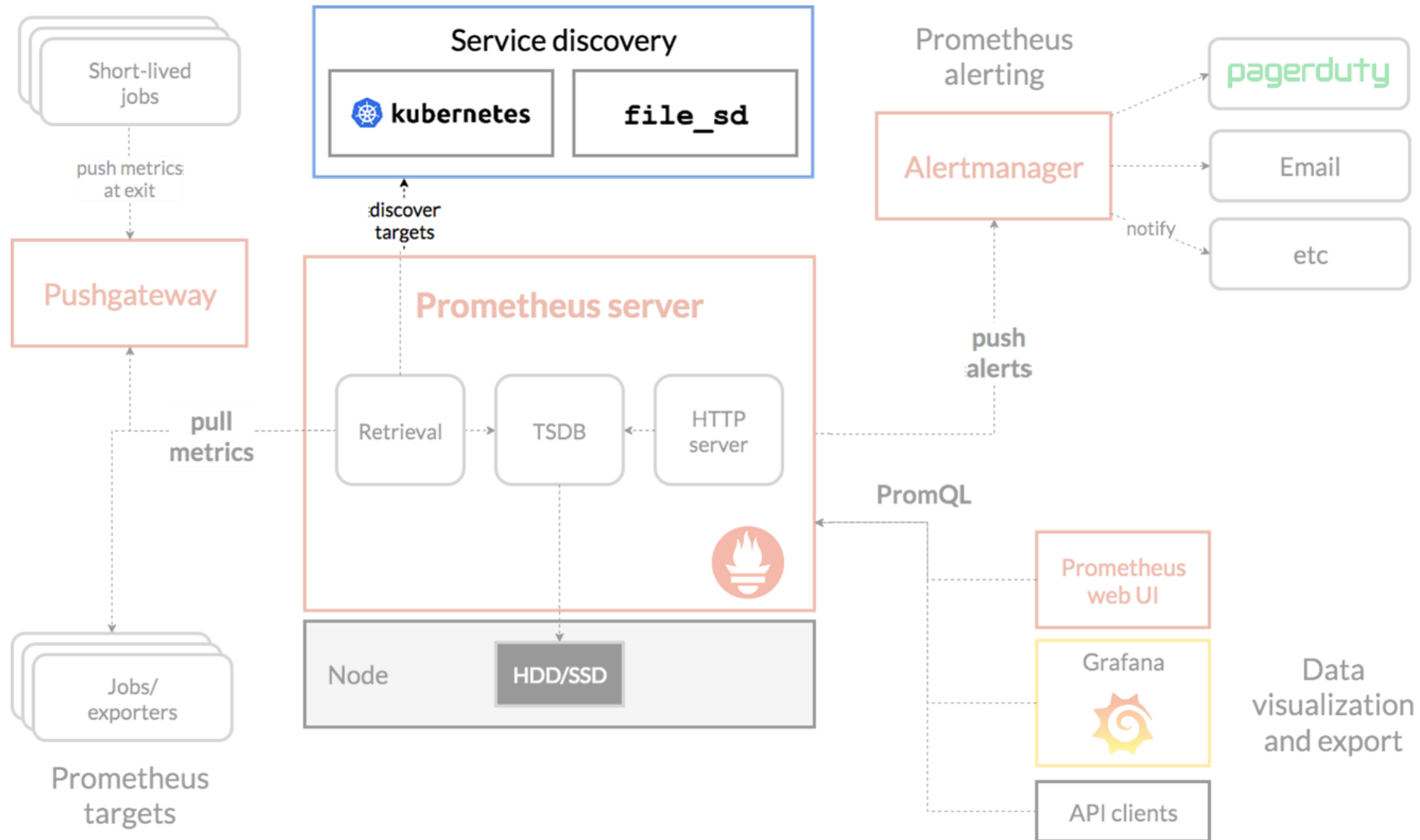
What is Prometheus?
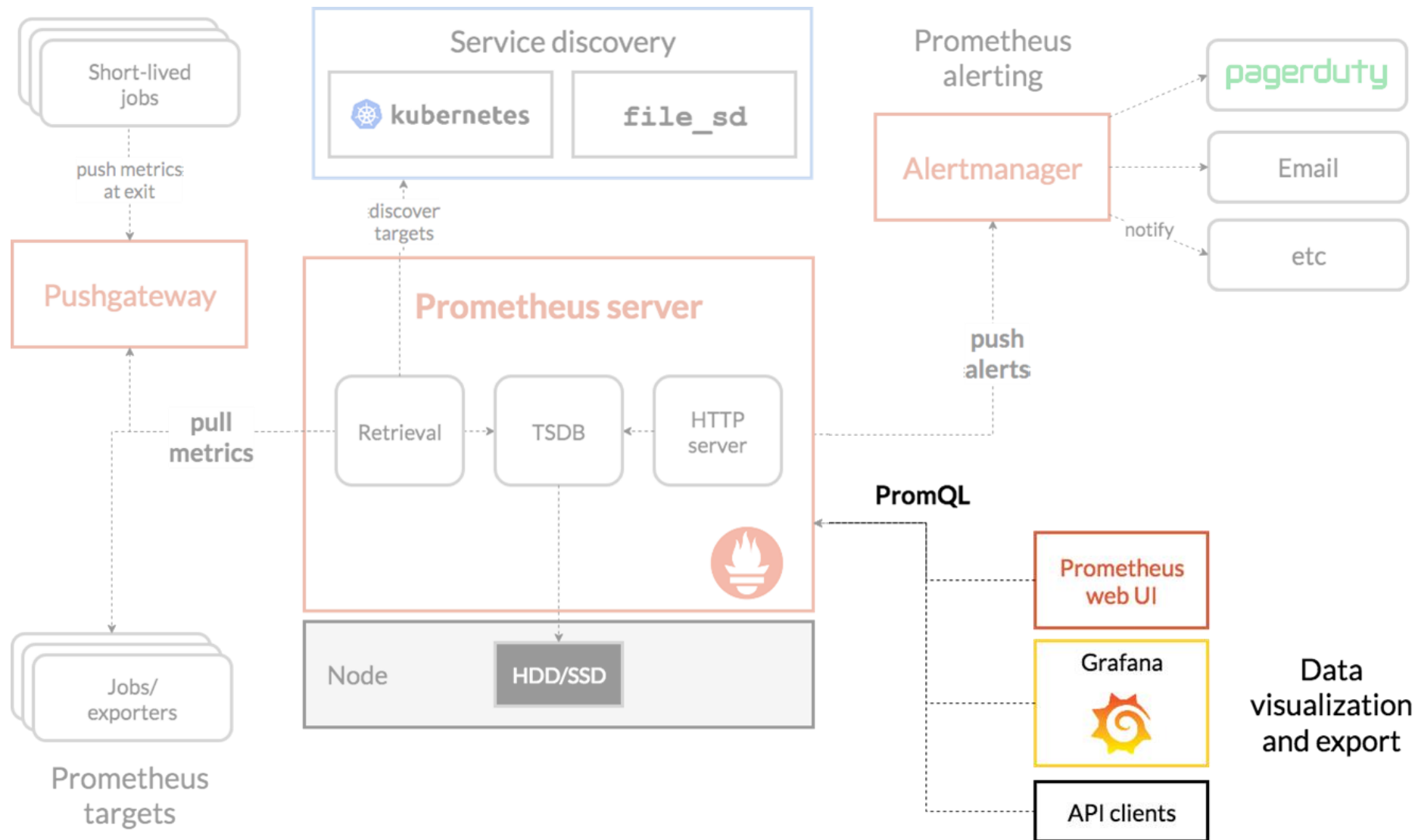
What is Prometheus?

What is Prometheus?

Prometheus configuration

MonitorServer

Prometheus

OS metrics

log file

targetServer
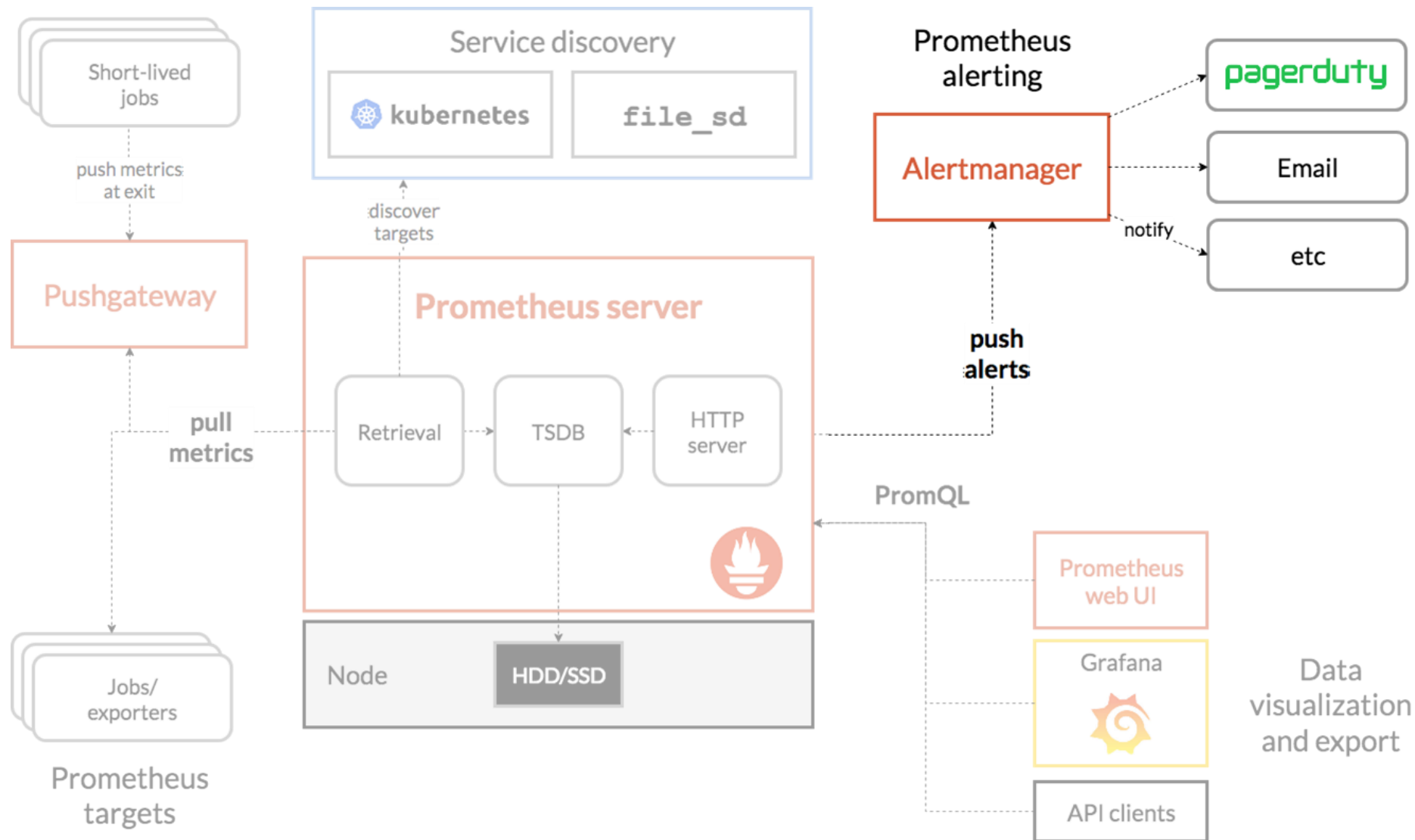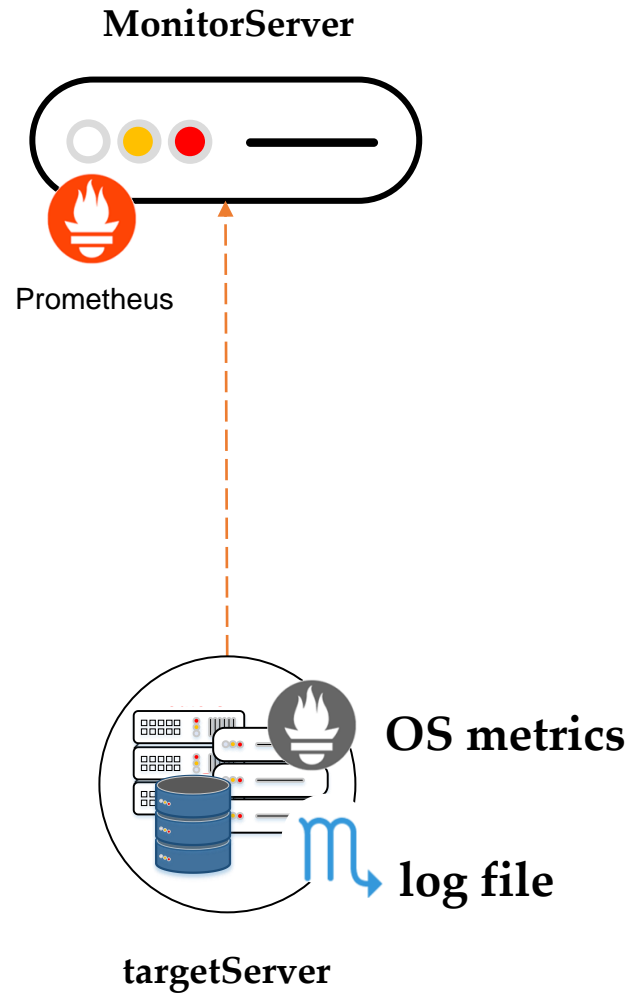
Prometheus configuration

```
Terminal – MonitorServer

codedive@MonitorServer:~$ sudo wget https://github.com/prometheus/prometheus/releases/
download/v2.37.2/prometheus-2.37.2.linux-amd64.tar.gz
codedive@MonitorServer:~$
codedive@MonitorServer:~$ sudo mkdir /usr/local/bin/Prometheus
codedive@MonitorServer:~$
codedive@MonitorServer:~$ mv prometheus-2.37.2.linux-amd64.tar.gz /usr/local/bin/Prometheus
codedive@MonitorServer:~$ cd /usr/local/bin/Prometheus
codedive@MonitorServer:~$
codedive@MonitorServer:~$ tar xvfz prometheus-2.37.2.linux-amd64.tar.gz
codedive@MonitorServer:~$ cd prometheus-2.37.2.linux-amd64
codedive@MonitorServer:~$
codedive@MonitorServer:~$ vim prometheus.yml
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$ ufw allow 9090/tcp
codedive@MonitorServer:~$
codedive@MonitorServer:~$ ./prometheus --config.file=prometheus.yml > /dev/null 2>&1 &
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$ vim /etc/systemd/system/prometheus.service
```
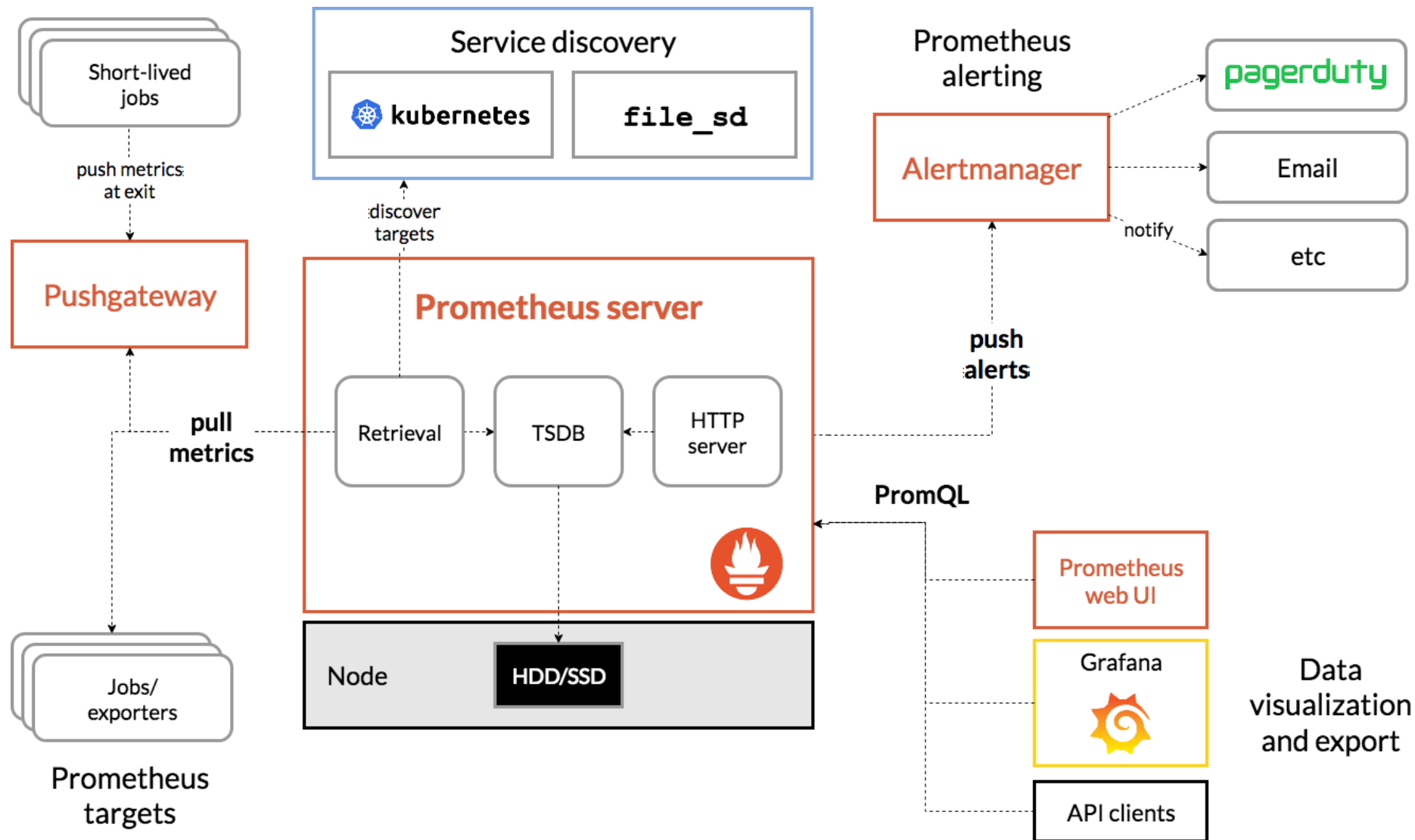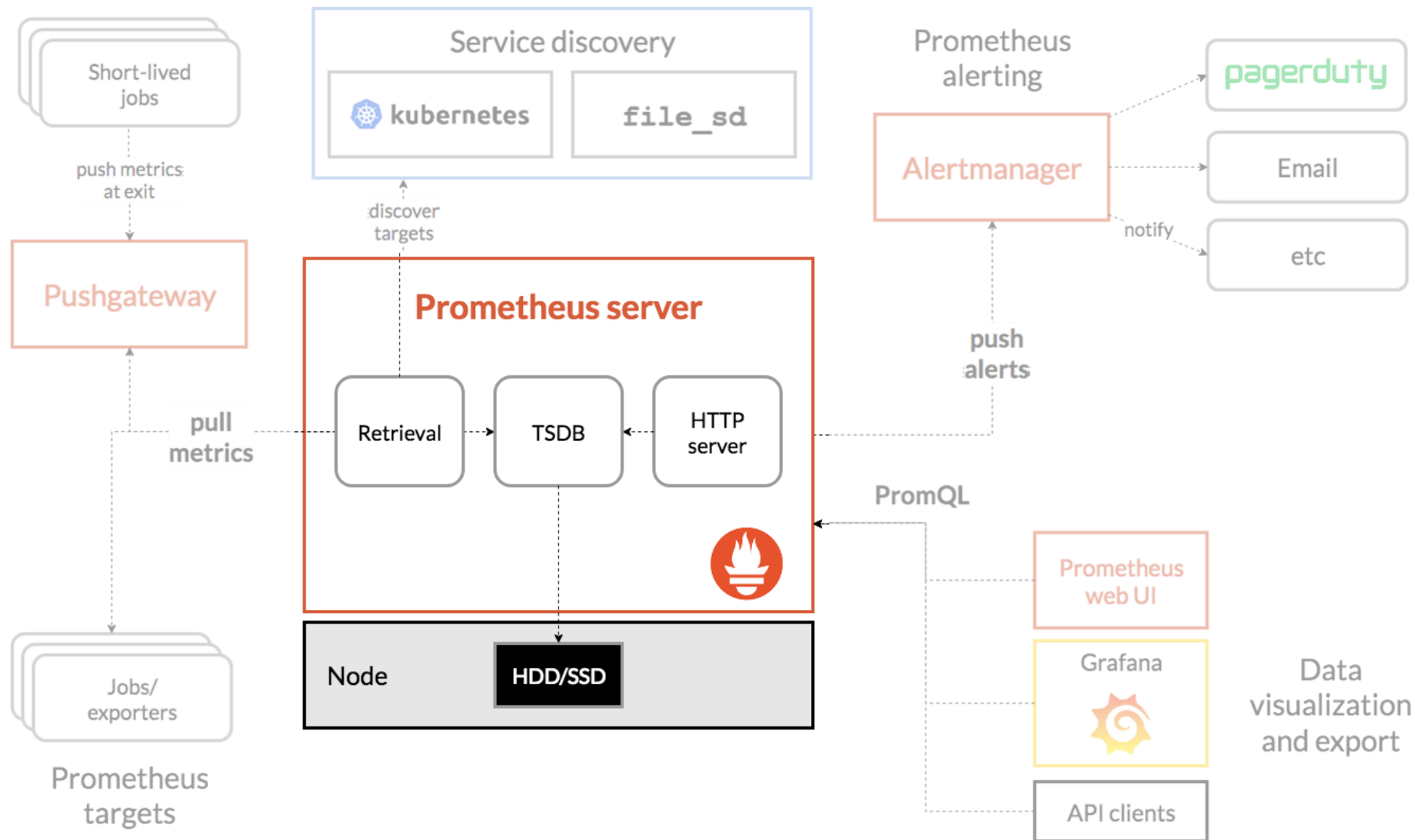
MonitorServer

Prometheus

targetServer

Prometheus configuration

```
[Unit]
    Description=Prometheus
    User=root
    Group=root
    After=local-fs.target network-online.target network.target
    Wants=local-fs.target network-online.target network.target

[Service]
    Type=simple
    Restart=on-failure
    ExecStart=/usr/local/bin/prometheus/prometheus-2.37.2.linux-amd64/prometheus --config.file=
    /usr/
    local/bin/prometheus/prometheus-2.37.2.linux-amd64/prometheus.yml

[Install]
    WantedBy=multi-user.target
```

MonitorServer

Prometheus

targetServer

Prometheus configuration

```
Terminal – MonitorServer

codedive@MonitorServer:~$ systemctl daemon-reload
codedive@MonitorServer:~$ systemctl start Prometheus
codedive@MonitorServer:~$ systemctl status prometheus
codedive@MonitorServer:~$ systemctl enable prometheus
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
codedive@MonitorServer:~$
```

MonitorServer

Prometheus

targetServer

Prometheus configuration

```
Terminal – TargetServer

codedive@TargetServer:~$
codedive@TargetServer:~$ wget https://github.com/prometheus/node_exporter/releases/download
/v1.4.0/node_exporter-1.4.0.linux-amd64.tar.gz
codedive@TargetServer:~$ tar xvfz node_exporter-1.4.0.linux-amd64.tar.gz
codedive@TargetServer:~$
codedive@TargetServer:~$ cd node_exporter-1.4.0.linux-amd64
codedive@TargetServer:~$ mv node_exporter /usr/local/bin/node_exporter
codedive@TargetServer:~$
codedive@TargetServer:~$ ufw allow 8081/tcp
codedive@TargetServer:~$
codedive@TargetServer:~$ node_exporter --web.listen-address 0.0.0.0:8081 > /dev/null 2>&1 &
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
```

MonitorServer

Prometheus

targetServer

Prometheus configuration


```
Terminal — TargetServer

codedive@TargetServer:~$ cat anylogfile.log
time, ip,hw alarms, cpu errors, low volt, overheat alarm, fan degraded, priority, failure
2022211500      86.245.244.226 0        0      2      1      1      3      1
2022220912      242.254.113.85 0        0      1      3      3      3      1
2022231054      180.67.216.32  2        0      2      0      0      2      2
2022251123      17.164.18.174  2        0      1      2      2      3      2
2022261254      16.75.147.107  1        0      1      1      1      4      1
2022281333      166.42.237.79  0        0      2      1      0      4      1
2022291531      240.151.80.141 1        0      3      0      0      1      2
2022271723      251.187.13.90  0        0      4      3      3      2      2
2022241324      32.40.200.86   0        0      1      0      0      1      0
2022011136      106.129.137.98 0        0      2      4      3      2      2
2022031252      101.25.58.95   1        0      2      3      3      1      2
2022041521      16.10.150.63   1        2      1      3      3      2      2
2022102043      168.147.131.18 1        0      2      2      1      3      2
2022122245      230.12.118.189 1        1      1      4      3      2      3
2022141247      17.226.234.190 1        0      1      0      0      3      0
2022191539      119.159.65.0   1        0      4      0      0      3      2
2022201848      129.255.137.20 0        0      2      0      0      1      1
2022232346      172.91.66.180  1        0      1      3      3      3      2
2022232339      154.92.117.111 1        0      3      1      1      2      1
2022251449      12.103.196.116 1        0      1      3      3      2      2
2022271049      90.205.195.153 0        0      2      0      0      1      1
2022271943      98.95.46.105   1        0      1      3      3      3      2
2022281157      158.120.7.77   1        0      3      1      1      2      1
2022292021      20.28.39.208   1        0      1      3      3      2      2
```
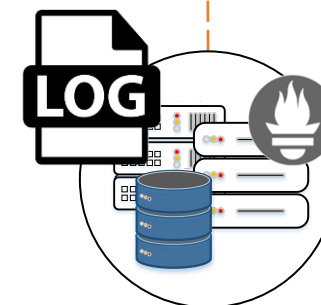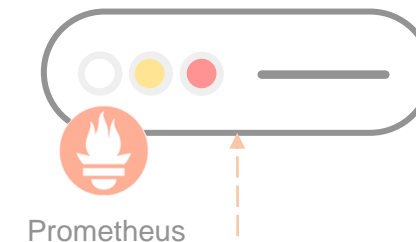
MonitorServer

Prometheus

targetServer

Prometheus configuration


```
Terminal – TargetServer

codedive@TargetServer:~$ apt-get update
codedive@TargetServer:~$ apt-get install mtail
codedive@TargetServer:~$ systemctl enable mtail
codedive@TargetServer:~$
codedive@TargetServer:~$ ufw allow 3903/tcp
codedive@TargetServer:~$
codedive@TargetServer:~$ vim /etc/mtail/whateverlogfine.mtail
```

MonitorServer

Prometheus

targetServer

Prometheus configuration

```
Terminal — vim /etc/mtail/whateverlogfine.mtail

counter hardware_alarm by ip
counter cpu_error by ip

# log_format mtail '$time, $ip, $"hw alarms", $"cpu errors" , $"low volt"'
#                  '$"overheat alarm", $"fan degraded", $priority, $failure';


/^/ +
#
/\s(?P<transfertime>\d+)/ +

# remote IP
/\s(?P<remoteIP>\d+\.\d+\.\d+\.\d+)/ +

# hw alarms
/\s(?P<hw_alarms>\d+)/ +

# cpu errors
/\s(?P<cpu_error>\d+)/ +

...
...
...

{
    hardware_alarm[$ip]++
    cpu_error[$ip]++
}
```
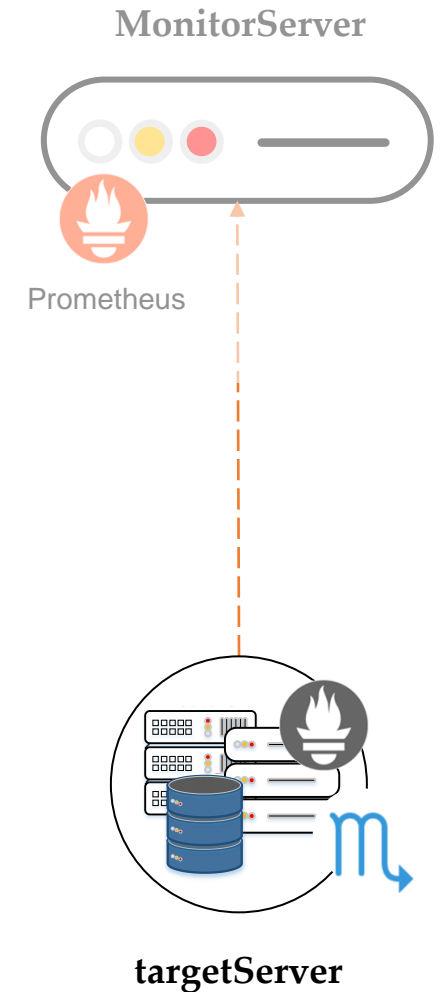
MonitorServer

Prometheus

targetServer

Prometheus configuration

```
Terminal – TargetServer

codedive@TargetServer:~$
codedive@TargetServer:~$ mtail  --logs /var/log/anyservice/ anylogfile.log  --progs
/etc/mtail/ > /dev/null 2>&1 &
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
codedive@TargetServer:~$
```

MonitorServer

Prometheus

targetServer

Prometheus configuration


```
Terminal — MonitorServer

codedive@MonitorServer:~$
codedive@MonitorServer:~$ cd /usr/local/bin/prometheus/prometheus-2.37.2.linux-amd64/
codedive@MonitorServer:~$
codedive@MonitorServer:~$ vim prometheus.yml
```

MonitorServer

Prometheus

targetServer

Prometheus configuration

```
..........
..........
scrape_configs:
..........
..........
  - job_name: "nodes"
    # metrics definition
    scrape_interval: 5s
    static_configs:
        - targets: ["63.32.45.224:8081", "63.32.45.224:3903"]
          labels:
                group: 'nodes'
```
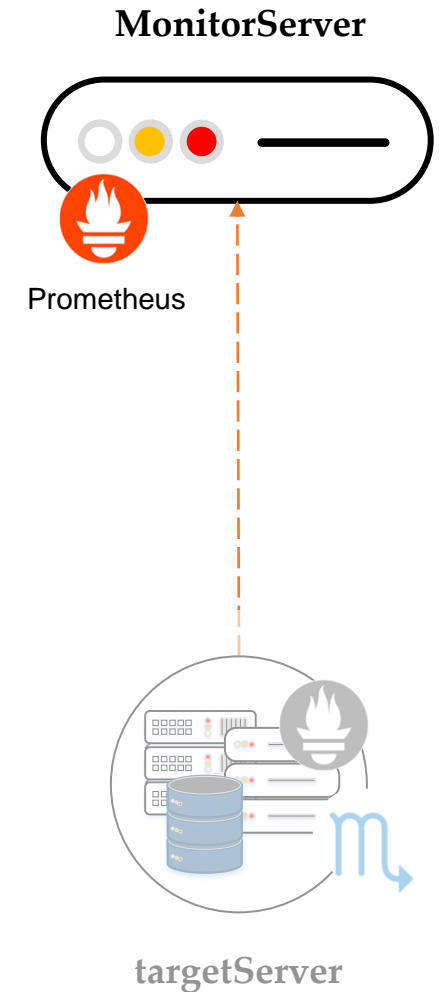
Terminal – vim prometheus.yml

MonitorServer

Prometheus

targetServer
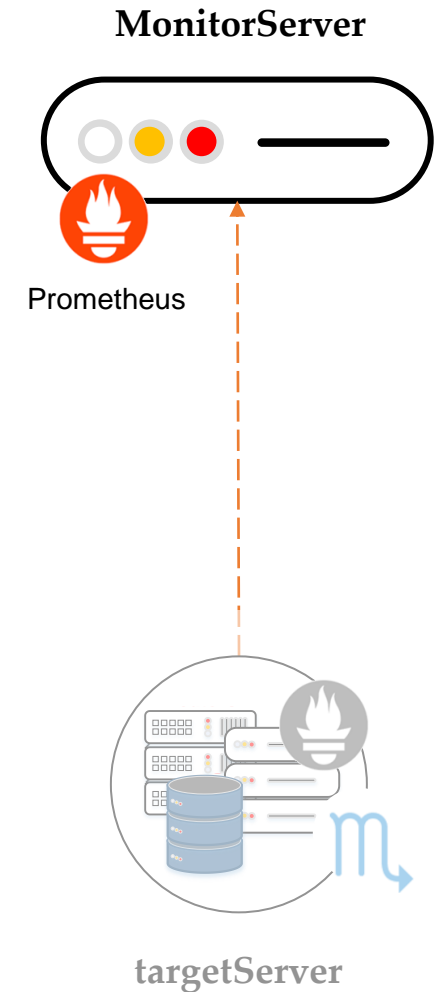
Statistic Analysis

Insight over dataset

```
codedive@MonitorServer:~$ python
Python 3.9.5 (default, May 18,2021,14:42:02) (AMD64)::Anaconda, Inc.
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>>
>>> import pandas as pd
>>> df = pd.read_csv('dataset/dataset.csv')
>>> print(df[:2000].describe(include='all').T)

                             count      mean        std     min    max
hardware alarms             2000.0    0.8970   0.854843   0.0    3.0
cpu errors                  2000.0    0.0480   0.246427   0.0    2.0
low voltage                 2000.0    1.5595   1.155910   0.0    4.0
overheating alarm           2000.0    1.3440   1.249177   0.0    5.0
fan degradaded              2000.0    1.0665   1.044329   0.0    3.0
baseband processing fault   2000.0    0.2760   0.633265   0.0    3.0
synchronization failure     2000.0    0.0685   0.299754   0.0    2.0
low disk performace         2000.0    0.0730   0.318939   0.0    2.0
tlpm                        2000.0  122.0190  89.775478  10.0  359.0
tlcm                        2000.0  132.2450  62.102724   7.0  331.0
number of PM                2000.0    1.8815   0.942816   1.0    4.0
priority                    2000.0    2.5120   1.007156   1.0    4.0
other alarms                2000.0    0.2220   0.694231   0.0    4.0
failure                     2000.0    1.4625   1.180801   0.0    6.0
```
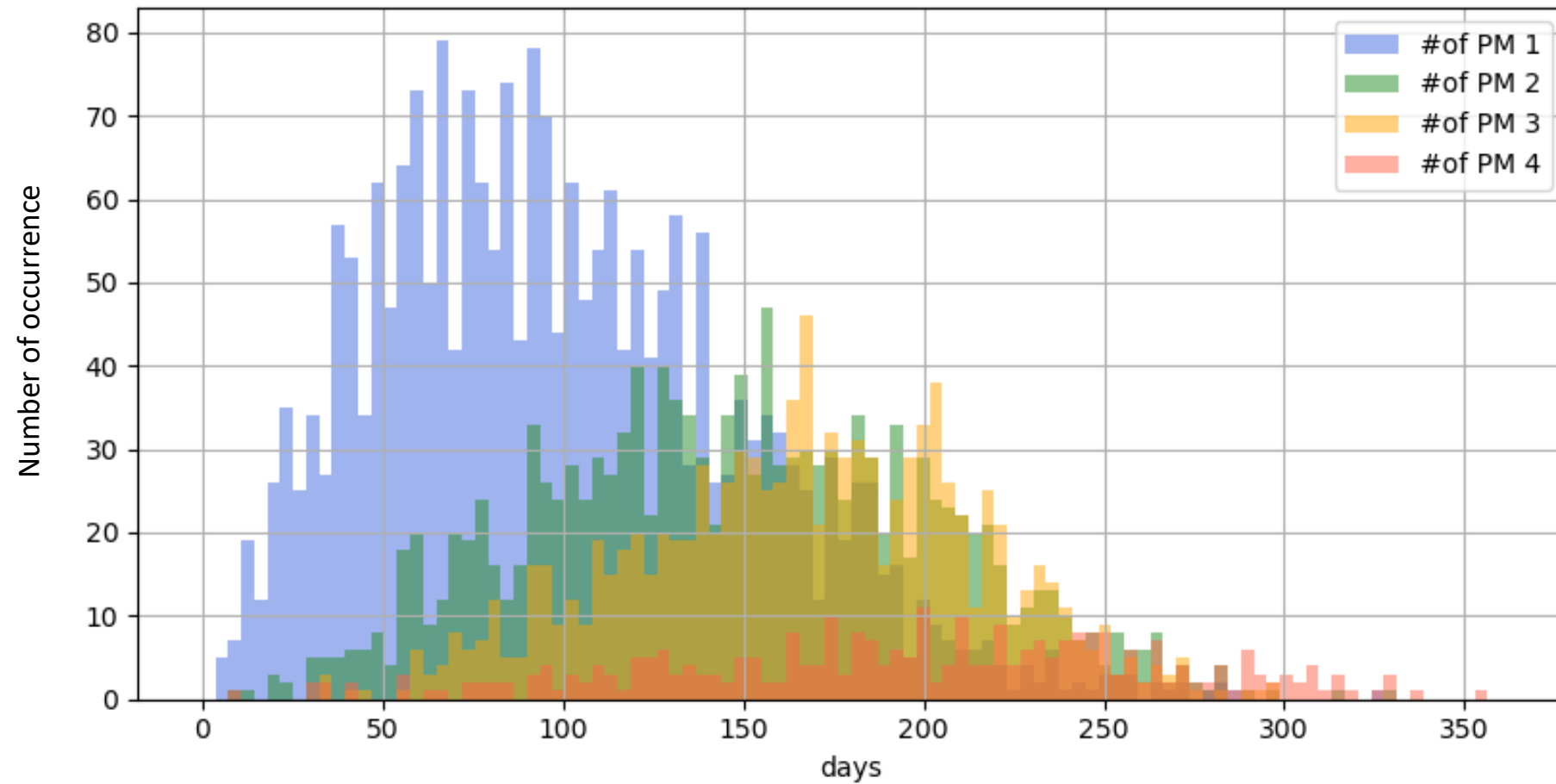
List of features

hardware alarms
cpu errors
low voltage
overheating alarm
fan degradaded
baseband processing fault
synchronization failure
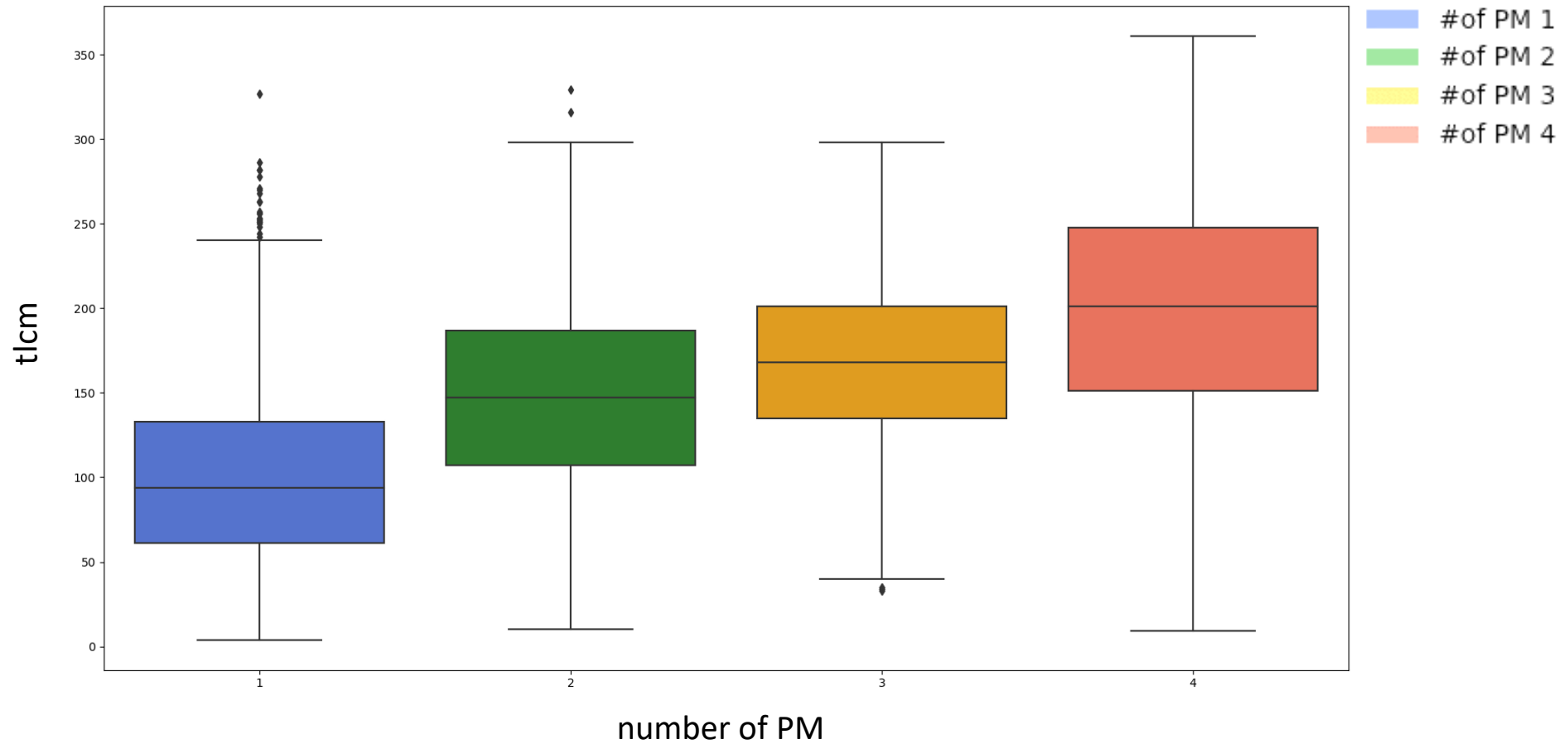low disk performace
tlpm
tlcm
number of PM
priority
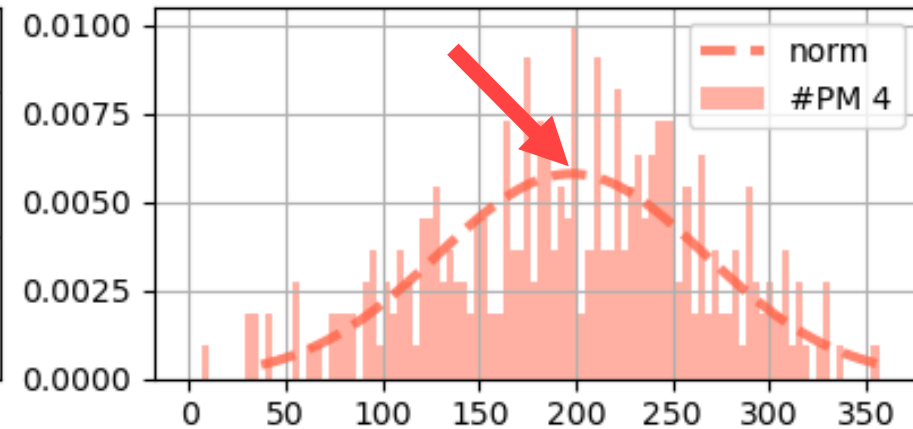other alarms
failure

Histogram
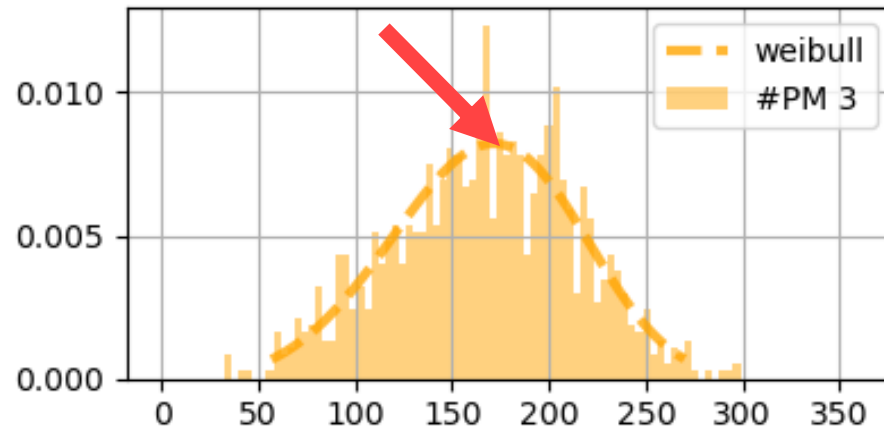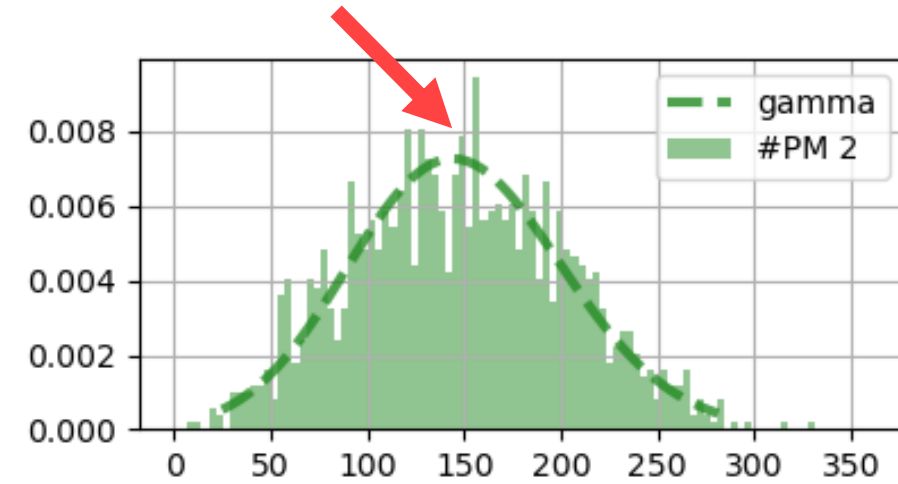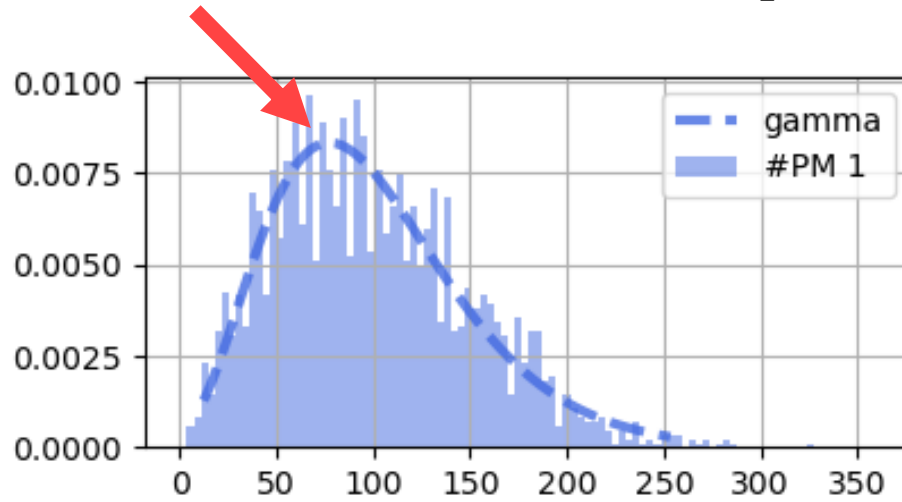
Time to last corrective maintenance

Boxplot

Time to last corrective maintenance

Fitting the Probability function

most probable time to failure

Insight over dataset

```python
1  import numpy as np
2  import pandas as pd
3  from scipy.stats import gamma, weibull_min, norm
4  import matplotlib.pyplot as plt
5  from fitter import Fitter
6
7  df = pd.read_csv('dataset/dataset.csv')
8
9  plt.figure(1)
10 plt.hist(df.loc[df['number of PM'] == 1,'tlcm'], bins = 100,
11            range=(0,360),alpha=0.5, label='#of PM 1', density=False,
12            facecolor='royalblue')
13
14 plt.hist(df.loc[df['number of PM'] == 2,'tlcm'], bins = 100,
15            range=(0,360),alpha=0.5, label='#of PM 2', density=False,
16            facecolor='forestgreen')
17
18 plt.hist(df.loc[df['number of PM'] == 3,'tlcm'], bins = 100,
19            range=(0,360),alpha=0.5, label='#of PM 3', density=False,
20            facecolor='orange')
21
22 plt.hist(df.loc[df['number of PM'] == 4,'tlcm'], bins = 100,
23            range=(0,360),alpha=0.5, label='#of PM 4', density=False,
24            facecolor='tomato')
25
26
27 plt.xlabel('days')
28 plt.ylabel('probability')
29 plt.grid()
30 plt.legend(loc='best')
31
32
```
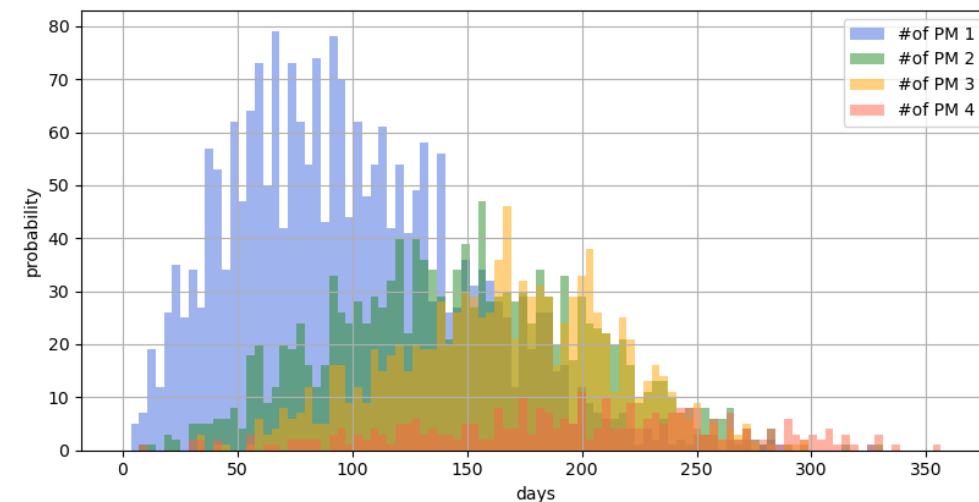
Insight over dataset

```python
33  # plot and fitting the pdf based on different #number of PMs
34  # ------------------------------- PM = 1 ----------------------------------
35  plt.figure(2)
36  plt.subplot(2, 2, 1)
37  dataIn = df.loc[df['number of PM'] == 1,'tlcm']
38  f = Fitter(dataIn, distributions=['gamma', "weibull_min", "norm"])
39  f.fit()
40  print('parameter for #PM=1 are: {}'.format(f.get_best(method = 'sumsquare_error')))
41  bFitPara = f.get_best(method = 'sumsquare_error')
42  a = bFitPara['gamma']['a']
43  l = bFitPara['gamma']['loc']
44  s = bFitPara['gamma']['scale']
45
46  x = np.linspace(gamma.ppf(0.01, a, loc = l, scale = s), gamma.ppf(0.99, a, loc = l,
47  scale = s), 100)
48
49  plt.plot(x, gamma.pdf(x, a,loc = l, scale = s), '--', color='royalblue', lw=3,
50              alpha=0.8, label='gamma')
51
52  plt.hist(dataIn, bins = 100, range=(0,360), alpha=0.5, label='#PM 1', density=True,
53              facecolor='royalblue')
54
55  plt.grid()
56  plt.legend(loc='best')
57
58
59  # plot for the remaining values of PM = {1,2,3,4} are almost same
60
61
62
63
64
```
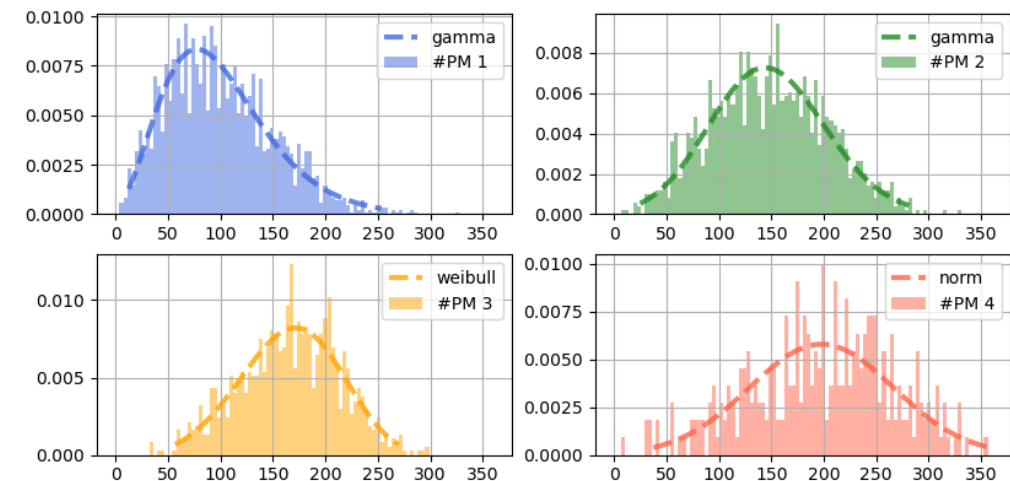
Estimation of failures

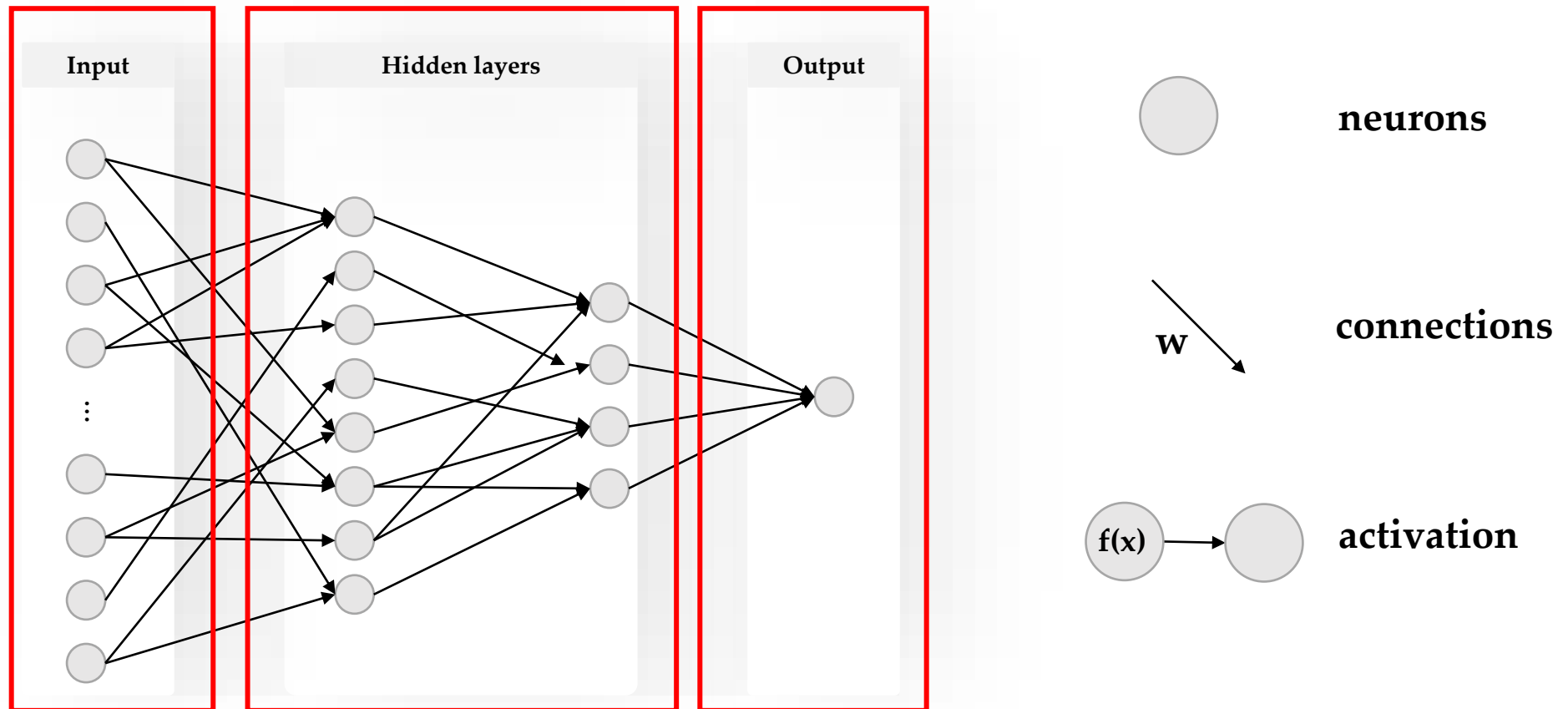# Goal of estimation

Goal:

Anticipate the most probable failures

Methods:

Method 1: fit probability function over data

Method 2: use the neural network for estimation

What is neural network?

Input

Hidden layers

Output

neurons

w   connections

f(x) → activation

Neural Network

```python
1   import pandas as pd
2   from matplotlib import pyplot as plt
3   from tensorflow.keras import models, layers, metrics
4   import shap
5   import tensorflow as tf
6
7   tf.random.set_seed(5)
8
9   df = pd.read_csv('dataset/dataset.csv', index_col=None)
10
11  # create neural network model
12  model = models.Sequential()
13  model.add(layers.Dense(name="Hidden 1", input_dim=13, units=8, activation='linear'))
14  model.add(layers.Dropout(name="drop1", rate=0.15))
15  model.add(layers.Dense(name="Hidden 2", units=4, activation='linear'))
16  model.add(layers.Dropout(name="drop2", rate=0.15))
17  model.add(layers.Dense(name="Output", units=1, activation='linear'))
18
19  model.compile(optimizer='Adadelta', loss='mse',
20               metrics=[tf.keras.metrics.MeanSquaredError()])
21
22  dataOut = df['failure']
23  dataIn  = df.drop(['failure'], axis=1)
24
25  x_training =  dataIn[0:1000].values
26  y_training = dataOut[0:1000].values
27
28  training = model.fit(x=x_training, y=y_training, batch_size=8, epochs=3000,
29               shuffle=True, verbose=1, validation_split=0.3)
30
31
32
```
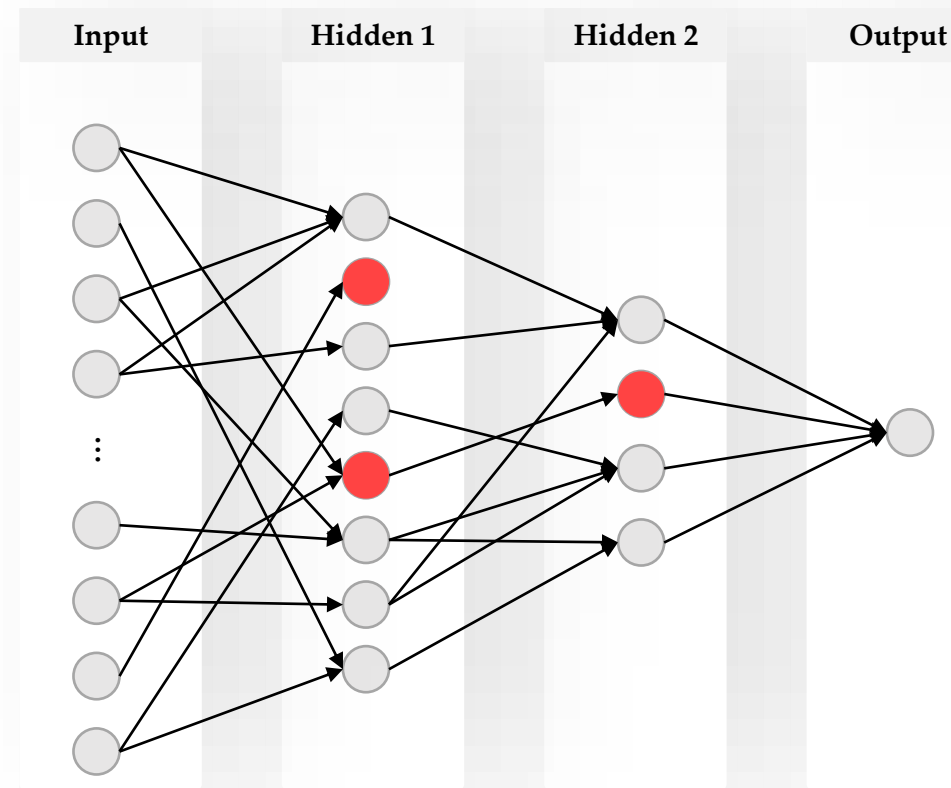
Neural Network

```python
33  # plots for training and validation on dataset
34  fig, ax = plt.subplots(nrows=1, ncols=2, sharey=True)
35
36  plt.figure(1)
37  # training plot
38  ax[0].set(title="Training data")
39  ax11 = ax[0].twinx()
40  lns01 = ax[0].plot(training.history['loss'], color='tomato', label='loss')
41  ax[0].set_xlabel('epochs')
42  ax[0].set_ylabel('loss', color='tomato')
43  lns02 = ax11.plot(training.history[metric], label=metric, color='royalblue')
44  ax11.set_ylabel("score", color='royalblue')
45
46  lns = lns01+lns02
47  labs = [l.get_label() for l in lns]
48  ax[0].legend(lns, labs, loc='upper right')
49  ax[0].grid(alpha=0.4)
50
51  # validation plot
52  ax[1].set(title="Validation data")
53  ax22 = ax[1].twinx()
54  lns11 = ax[1].plot(training.history['val_loss'], color='tomato', label='loss')
55  ax[1].set_xlabel('epochs')
56  ax[1].set_ylabel('loss', color='tomato')
57  lns12 = ax22.plot(training.history['val_'+metric], label=metric, color='royalblue')
58  ax22.set_ylabel("score", color="royalblue")
59
60  lns = lns11+lns12
61  labs = [l.get_label() for l in lns]
62  ax[1].legend(lns, labs, loc='upper right')
63  ax[1].grid(alpha=0.4)
64
```
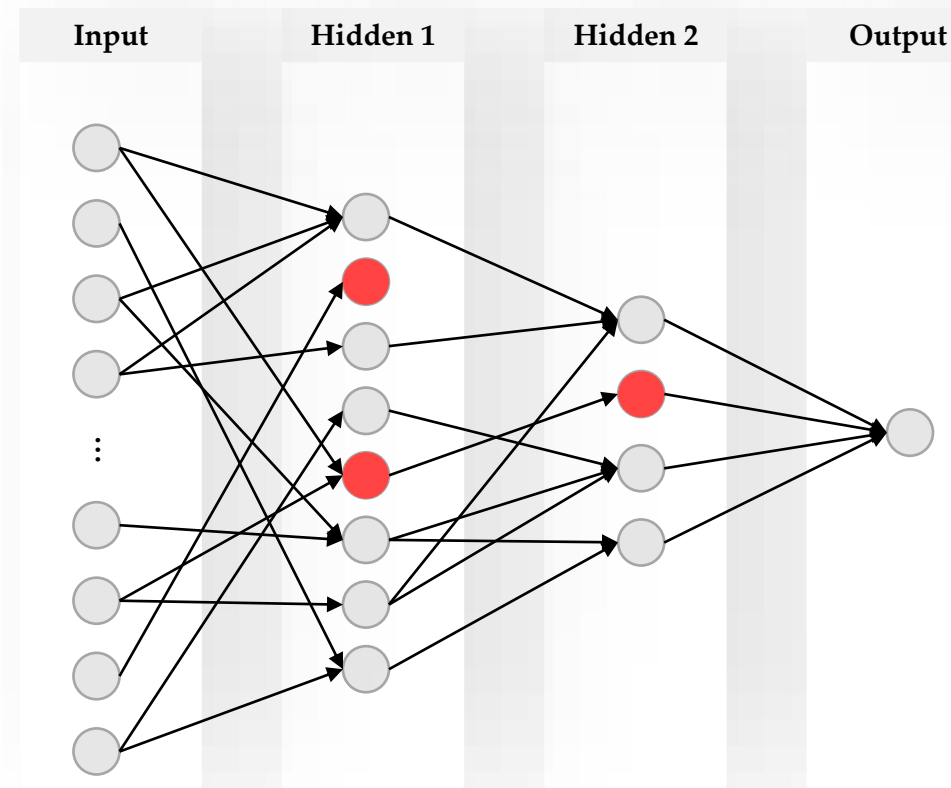
learning process vs. epochs
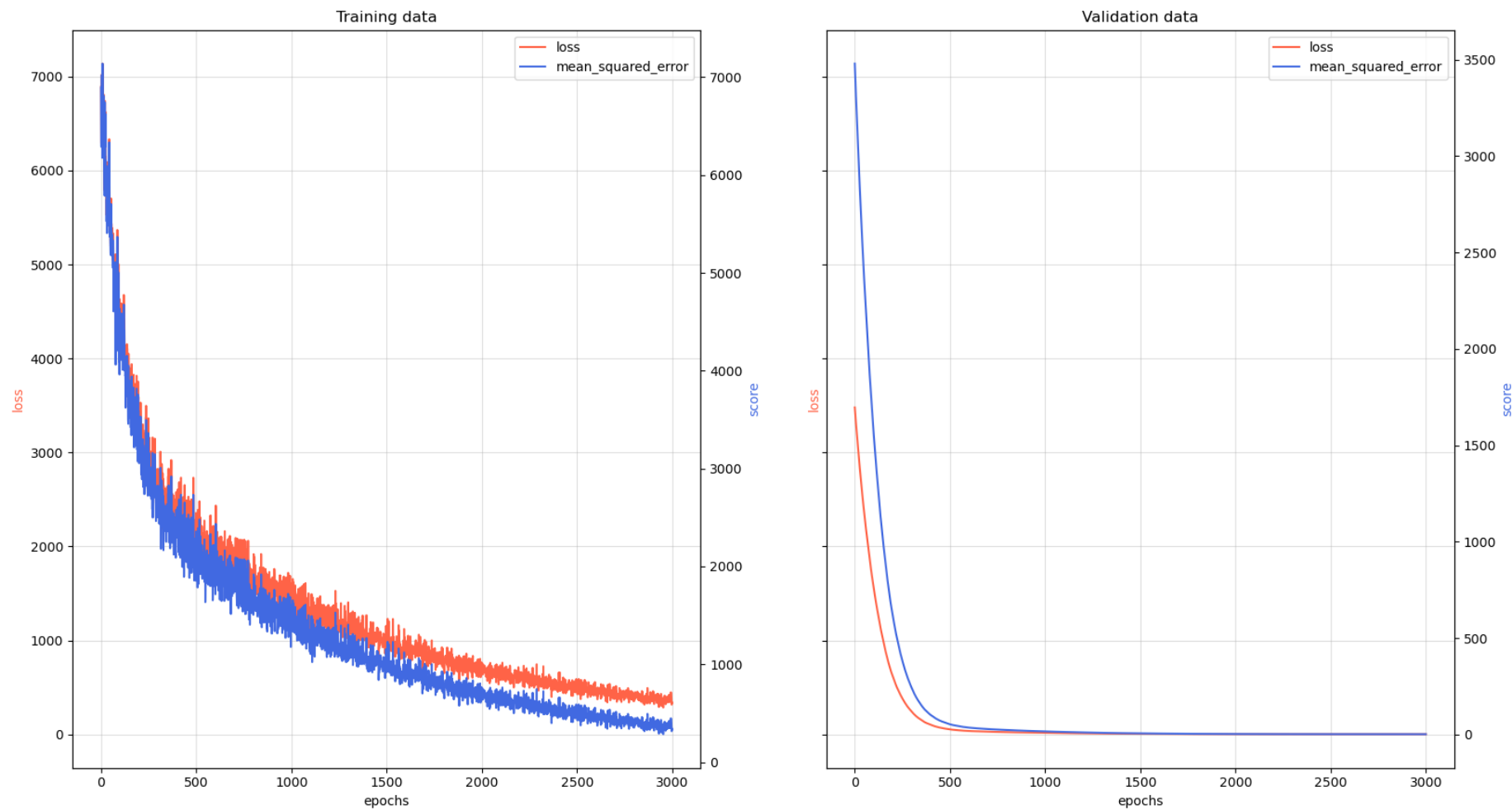
Neural Network

```python
# impact of feature to the value of target
dataInExplainer = dataIn[-1:].values
dataOutExpected = dataOut[-1:]

explainer = shap.Explainer(model, x_training)
shap_values = explainer(dataInExplainer)

values = shap_values.values[0]
base_values = shap_values.base_values[0][0]
datas = df[-1:].values[0]
columns = list(df.columns)[:-1] # exclude failure column

plt.figure(2, figsize=(10, 8), dpi=80)
exp = shap.Explanation(values, base_values, data=dataInExplainer[0],
feature_names=columns)
shap.plots.waterfall(exp, max_display=13, show=False)
plt.rcParams["font.size"] = "12"
plt.title('expected value is: {}'.format(dataOutExpected.values[0]))
plt.grid()
plt.show()
```

Neural Network

expected value is: 2

$f(x) = 2.324$

277 = tlpm	+1.3
3 = fan degradaded	−0.21
160 = tlcm	+0.17
2 = low voltage	−0.17
3 = overheating alarm	−0.13
0 = baseband processing fault	−0.12
1 = number of PM	−0.06
0 = low disk performace	+0.05
2 = priority	−0.04
0 = hardware alarms	−0.03
0 = other alarms	−0.03
0 = synchronization failure	+0
0 = cpu errors	+0

$E[f(X)] = 1.595$
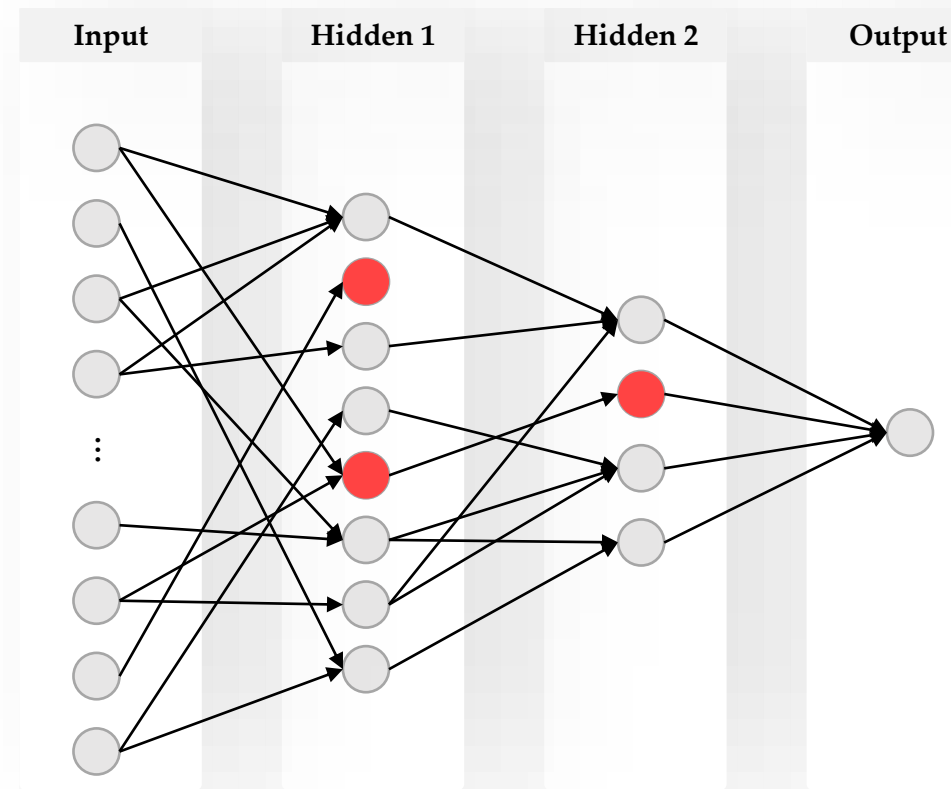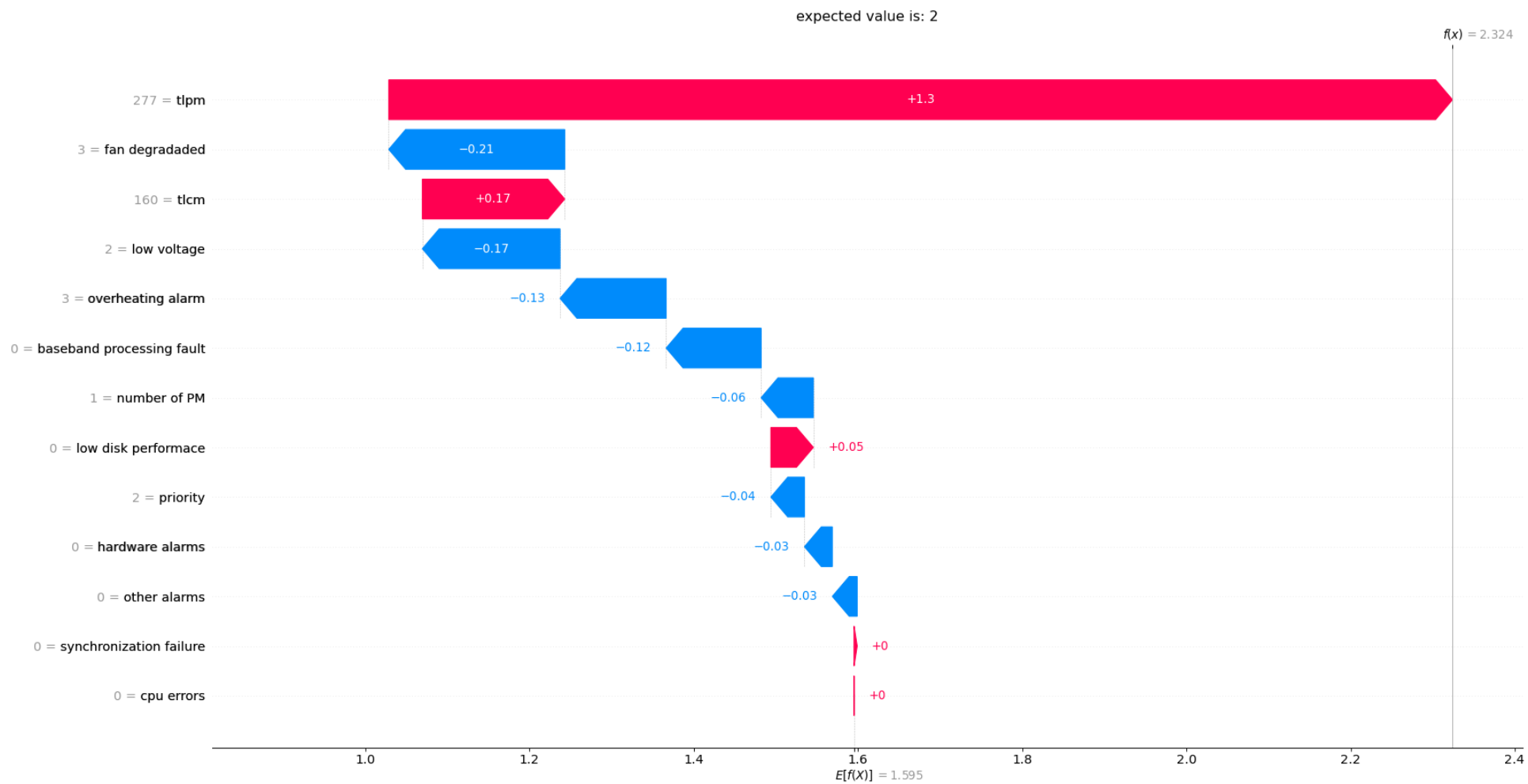
Modeling

Modeling

Probable financial loss:

```
for each node:
    p = find probability of failure
    c = find income per day
    Ci = p * c
```

Probable non-financial loss:
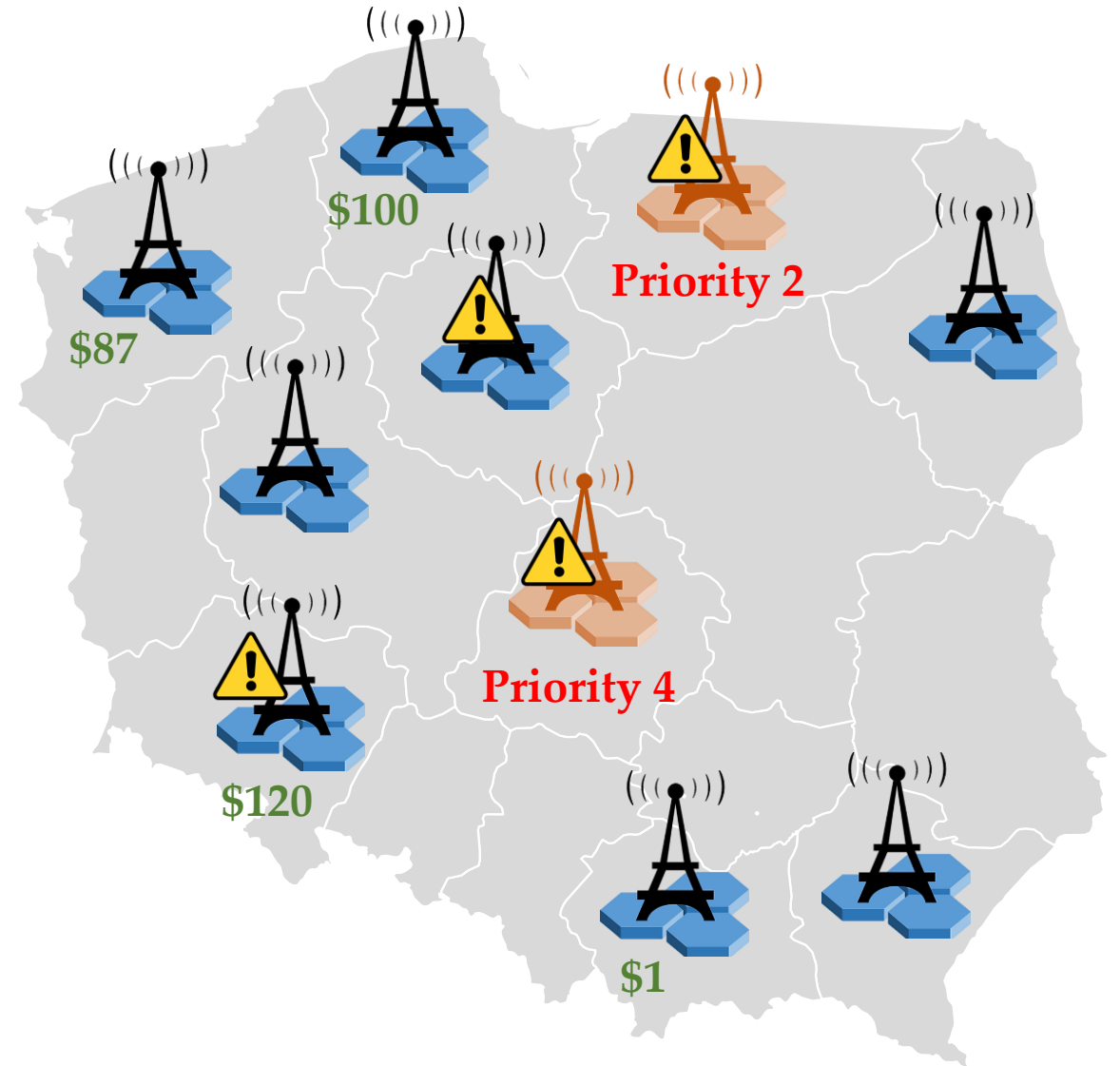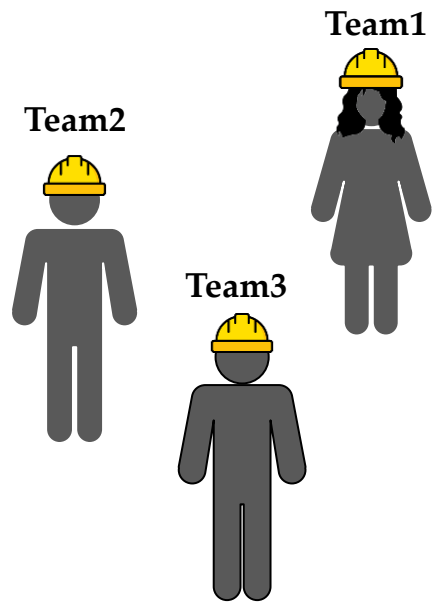
```
for each node:
    p = find probability of failure
    m = rate the node based on its importance 0-100
    Pi = p * m
```

Modeling

Sets and Indices

I : Set of all available task, $\quad i \in I$

M : Set of all available Workforces, $\quad m \in M$

Parameters

c_i : Probable financial loss due to task i

p_i : Probable non-financial loss due to task i

T_m : Available time in hours for workforce m

a_{im} : time consumed by workforce m to complete task i

Variables

y_{im} : is equal to 1 if workforce m is assigned to task i

w_i : is equal to 1 if task i is selected to be performed

Cost function

$$Max\{\alpha_1 F_1(x) + \alpha_2 F_2(x)\}$$

Where:

$$F_1(x) = \sum_{i \in I} c_i w_i \qquad \text{Cost function of financial loss}$$

$$F_2(x) = \sum_{i \in I} p_i w_i \qquad \text{Cost function of non-financial loss}$$

Constraints

$$w_i = \{0,1\} \quad \forall i \in I$$

$$y_{im} = \{0,1\} \quad \forall i \in I, m \in M$$

$$\sum_{m \in M} y_{im} = w_i \quad \forall i \qquad \text{assign only a task to a workforce}$$

$$\max \sum_{i \in I} a_{im} y_{im} \leq T_m \quad \forall m \in M \qquad \text{workforce time limitation}$$

Modeling

```python
import pyomo.opt as po
import pyomo.environ as pe

I  = ['Server01', 'Server03', 'Server04', 'Server05', 'Server08', 'Server10']
M  = ['Team01', 'Team02', 'Team03']
Ci = {'Server01': 109, 'Server03': 89, 'Server04': 23, 'Server05': 54,
         'Server08': 86, 'Server10': 68}
Pi = {'Server01': 17, 'Server03': 2, 'Server04': 100, 'Server05': 21,
         'Server08': 71, 'Server10': 34}

Tm = {'Team01': 2, 'Team02': 2, 'Team03': 3}
Aim = {
    ('Server01', 'Team01'): 1, ('Server01', 'Team02'): 3, ('Server01', 'Team03'): 1.5,
    ('Server03', 'Team01'): 2, ('Server03', 'Team02'): 5, ('Server03', 'Team03'): 0.5,
    ('Server04', 'Team01'): 3, ('Server04', 'Team02'): 4, ('Server04', 'Team03'): 4,
    ('Server05', 'Team01'): 1, ('Server05', 'Team02'): 1, ('Server05', 'Team03'): 5,
    ('Server08', 'Team01'): 3, ('Server08', 'Team02'): 9, ('Server08', 'Team03'): 3,
    ('Server10', 'Team01'): 4, ('Server10', 'Team02'): 5, ('Server10', 'Team03'): 1
}

# set solver and create model
solver = po.SolverFactory('glpk')
model = pe.ConcreteModel()

# Set & Indices definition
model.i = pe.Set(initialize=I)
model.m = pe.Set(initialize=M)

# Parameter Declaration
model.ci = pe.Param(model.i, initialize=Ci)
model.pi = pe.Param(model.i, initialize=Pi)
model.tm = pe.Param(model.m, initialize=Tm)
model.aim = pe.Param(model.i, model.m, initialize=Aim)
```

Sets and Indices

I : Set of all available skills, $i \in I$

M : Set of all available Workforces, $m \in M$

Parameters

c_i : Probable financial loss due to task i

p_i : Probable non-financial loss due to task i

T_m : Available time in hours for workforce m

a_{im} : time consumed by workforce m to complete task i

Modeling

```python
33  # Variable declaration
34  model.yim = pe.Var(model.i, model.m, domain=pe.Binary)
35  model.wi = pe.Var(model.i, domain=pe.Binary)
36
37  # cost function
38  def costfunc(model):
39      alphaC = 0.7; alphaP = 0.3
40      fC = sum(model.ci[i] * model.wi[i] for i in model.i)
41      fP = sum(model.pi[i] * model.wi[i] for i in model.i)
42      return alphaC * fC + alphaP * fP
43
44  model.obj = pe.Objective(rule = costfunc, sense = pe.maximize)
45
46  # contstraint definition
47  def ruleC1(model, i):
48      return sum(model.yim[i,m] for m in model.m) == model.wi[i]
49  model.C1 = pe.Constraint(model.i, rule=ruleC1)
50
51  def ruleC2(model, m):
52      return sum(model.aim[i,m] * model.yim[i,m] for i in model.i) <= model.tm[m]
53  model.C2 = pe.Constraint(model.m, rule=ruleC2)
54  # solve the model
55  solver.solve(model)
56
57  # print outout of the model
58  model.pprint()
59
60
61
62
63
64
```

Variables

y_{im} : is equal to 1 if workforce m is assigned to task i

w_i : is equal to 1 if task i is selected to be performed

Cost function

$$Max\{\alpha_1 F_1(x) + \alpha_2 F_2(x)\}$$

Where:

$$F_1(x) = \sum_{i \in I} c_i w_i \qquad \text{Cost function of financial loss}$$

$$F_2(x) = \sum_{i \in I} p_i w_i \qquad \text{Cost function of non-financial loss}$$

Constraints

$$w_i = \{0,1\} \quad \forall i \in I$$

$$y_{im} = \{0,1\} \quad \forall i \in I, m \in M$$

$$\sum_{m \in M} y_{im} = w_i \quad \forall i \qquad \text{assign only a task to a workforce}$$

$$\max \sum_{i \in I} a_{im} y_{im} \leq T_m \quad \forall m \in M \qquad \text{workforce time limitation}$$

Modeling

```
codedive@MonitorServer:~$ python ORMModel.py

...


2 Var Declarations
    wi : Size=6, Index=i
    Key   : Lower : Value : Upper : Fixed : Stale : Domain
    BTS01 :     0 :   1.0 :     1 : False : False : Binary
    BTS03 :     0 :   1.0 :     1 : False : False : Binary
    BTS04 :     0 :   0.0 :     1 : False : False : Binary
    BTS05 :     0 :   1.0 :     1 : False : False : Binary
    BTS08 :     0 :   0.0 :     1 : False : False : Binary
    BTS10 :     0 :   1.0 :     1 : False : False : Binary


    yim : Size=18, Index=yim_index
    Key                 : Lower : Value : Upper : Fixed : Stale : Domain
    ('BTS01', 'Team01') :     0 :   0.0 :     1 : False : False : Binary
    ('BTS01', 'Team02') :     0 :   0.0 :     1 : False : False : Binary
    ('BTS01', 'Team03') :     0 :   1.0 :     1 : False : False : Binary
    ('BTS03', 'Team01') :     0 :   1.0 :     1 : False : False : Binary
    ('BTS03', 'Team02') :     0 :   0.0 :     1 : False : False : Binary
    ('BTS03', 'Team03') :     0 :   0.0 :     1 : False : False : Binary
    ('BTS04', 'Team01') :     0 :   0.0 :     1 : False : False : Binary
    ('BTS04', 'Team02') :     0 :   0.0 :     1 : False : False : Binary
    ('BTS04', 'Team03') :     0 :   0.0 :     1 : False : False : Binary
    ('BTS05', 'Team01') :     0 :   0.0 :     1 : False : False : Binary
    ('BTS05', 'Team02') :     0 :   1.0 :     1 : False : False : Binary
    ('BTS05', 'Team03') :     0 :   0.0 :     1 : False : False : Binary
    ('BTS08', 'Team01') :     0 :   0.0 :     1 : False : False : Binary
    ('BTS08', 'Team02') :     0 :   0.0 :     1 : False : False : Binary
    ('BTS08', 'Team03') :     0 :   0.0 :     1 : False : False : Binary
    ('BTS10', 'Team01') :     0 :   0.0 :     1 : False : False : Binary
    ('BTS10', 'Team02') :     0 :   0.0 :     1 : False : False : Binary
    ('BTS10', 'Team03') :     0 :   1.0 :     1 : False : False : Binary
```
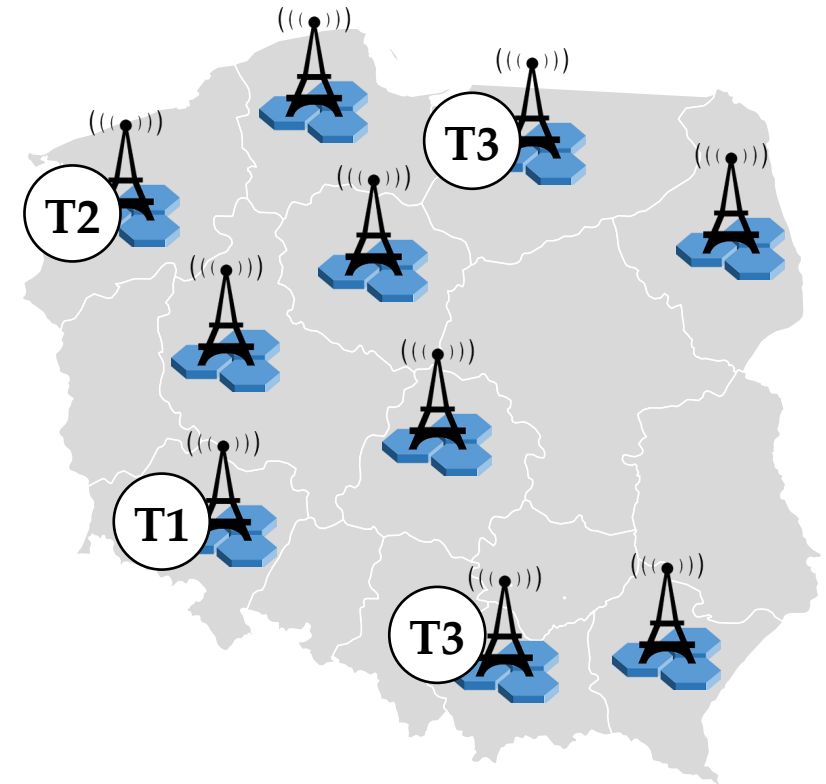
Thank you