Advanced Data Analysis Environment with PI System on MHPS-TOMONI

Hiroyasu Ishigaki
Leandro Hideo Iha

MHPS

©2018 OSIsoft, LLC

Agenda

1. MHPS-TOMONI Digital Solutions
2. Interactive Advanced Data Analysis
3. Demo
4. Conclusion

©2018 OSIsoft, LLC

1. MHPS-TOMONI Digital Solutions

MHPS Digital Solutions
TOMONI

©2018 OSIsoft, LLC

Our Background

MHPS At a glance

"Mitsubishi Heavy Industries x Hitachi"
Power plant equipment OEM / EPC contractor
The Prove of Engineering Excellence

20,000+ staffs

Over a century of experience in turbines & Boilers.

120+ years of solid background

1,000 Gas turbines,
2,600 Steam turbines
5,600 Boilers made

Headoffices in 17 countries with 48 group companies

2,000+ patents

- Service branches
- Remote Monitoring Centers
- Manufacturing sites

©2018 OSIsoft, LLC

MHPS-TOMONI™

MHPS Digital Solutions
TOMONI

A suite of digital solutions driven by user needs fueled by data analytics to make power plants digital.

©2018 OSIsoft, LLC

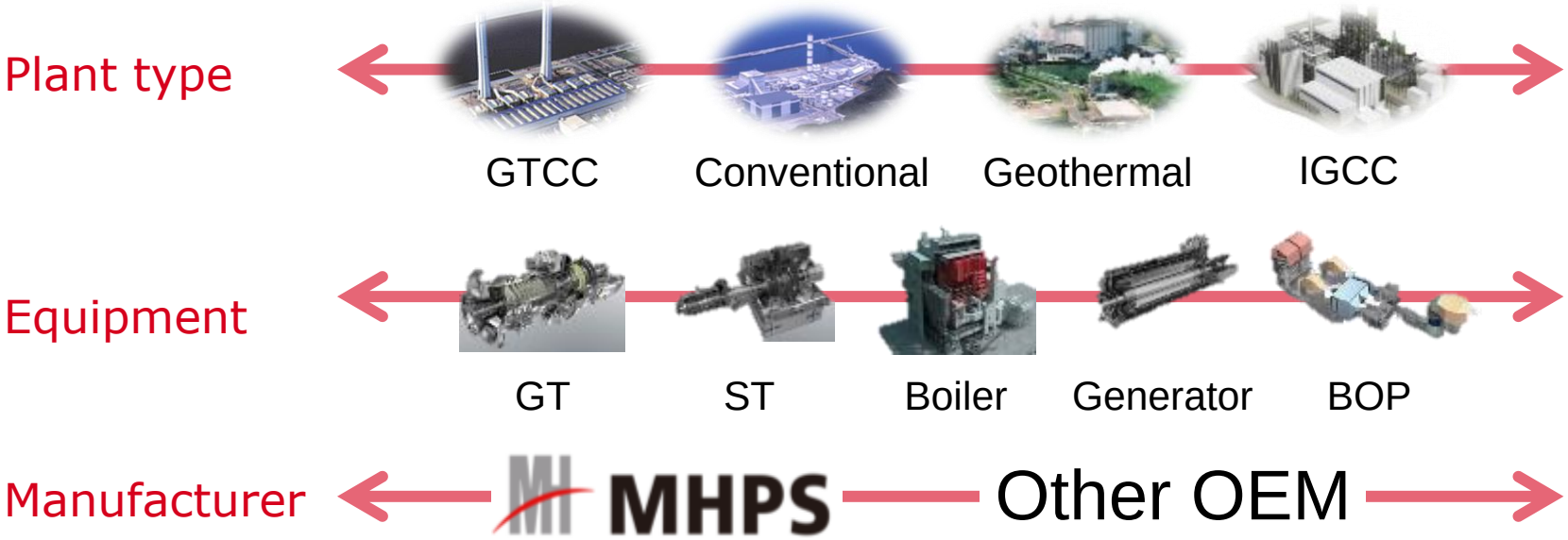
MHPS-TOMONI™

Bring in software experts to provide the best solution based on customer needs and priorities.

©2018 OSIsoft, LLC

MHPS-TOMONI™ Scope and Deployment

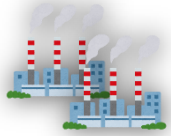
MHPS-TOMONI™ covers the entire thermal power plant, regardless of fuel and original equipment manufacturer, and is being steadily deployed around the world.



©2018 OSIsoft, LLC

MHPS-TOMONI : Roadmap

Autonomous Operation



Advanced O&M

O&M Support

Monitoring

Higher Reliability

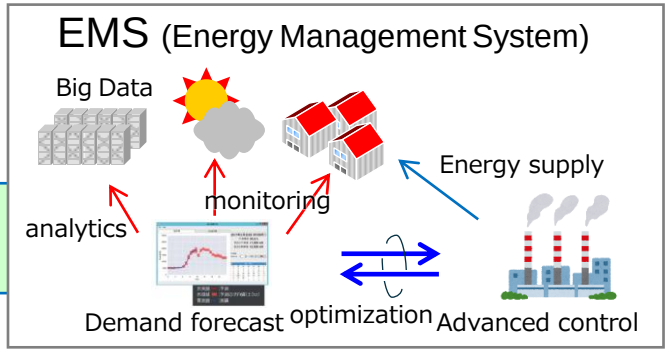
Shorter Outages

Extended Outage Intervals

Optimized Performance

Remote Operation

Optimize the overall fleet portfolio



Advanced Remote Monitoring

Automated Boiler Combustion Tuning

Global Service Center (Philippines)
Remote monitoring and O&M support

Predictive Analytics by diagnostics of motor current

©2018 OSIsoft, LLC

MHPS-TOMONI architecture

AI CPFM

Auto Vlv. tuning

Boiler AI tuning

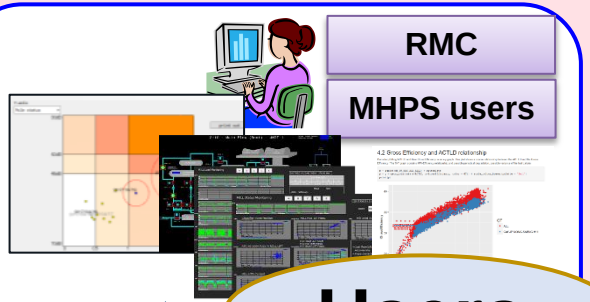
GT AI flow modulation

Edge application

SecureGW

Power plant

Edge



Users

KPI Analyst

Pre Act

Plant performance

GT parts lifetime

ST/GEN Predict.

Boiler diagnosis

Customer application

Internal application

PI System

OSIsoft.

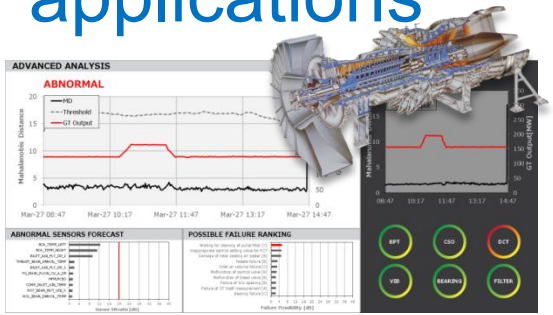
Cloud

Microsoft Azure

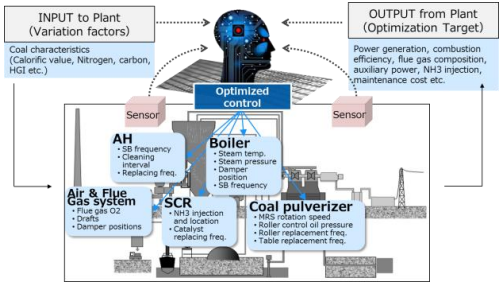
©2018 OSIsoft, LLC

MHPS-TOMONI™ applications

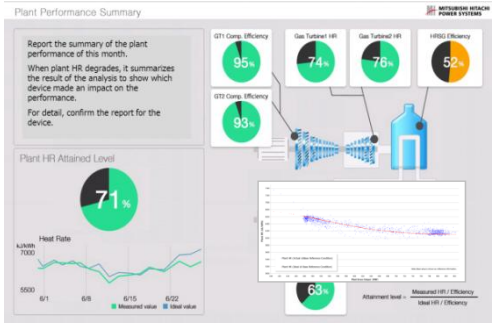
24/7 Remote monitoring center



Predictive analysis

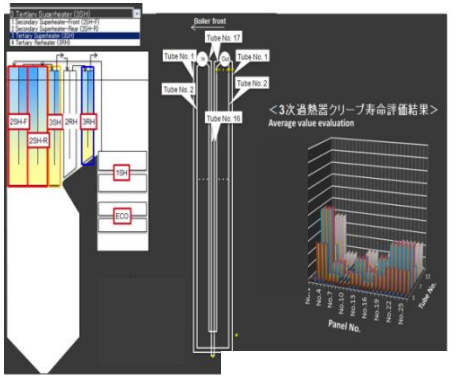


AI combustion tuning



Plant performance analysis

KPI Analyst



Boiler lifetime assessment

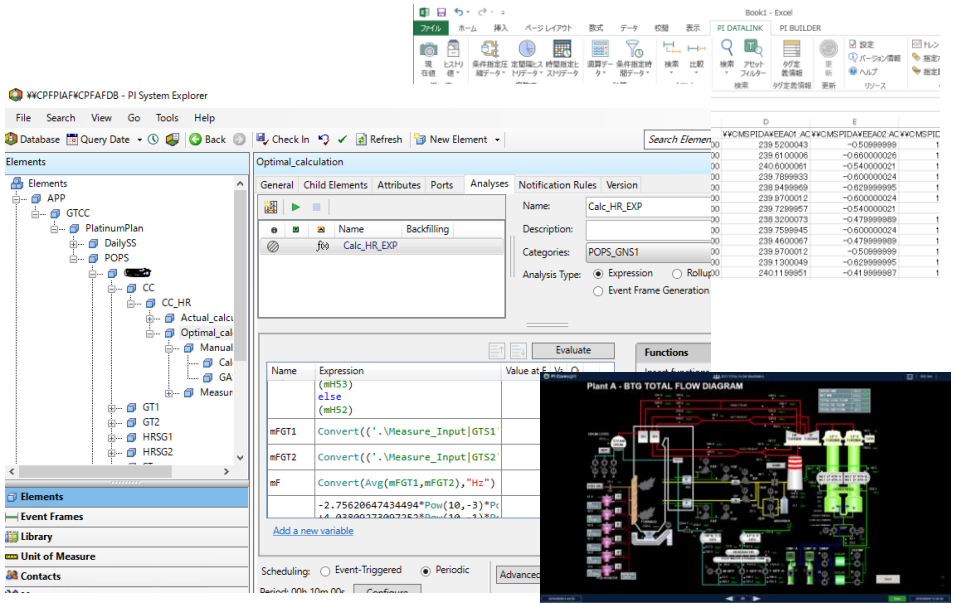
… and more

©2018 OSIsoft, LLC

Important PI components

Calculation / Visualization

- **PI Asset Framework**
  Handling multiple power plants with structure/templates
- **PI ProcessBook / PI DataLink**
  Data visualization, easy,
  rich functionality
- **PI Vision**
  Easy, Ad-hoc, mobility visualization
- **PI AF Analytics**
  Various data analysis including
  Heat rate, equipment efficiency



©2018 OSIsoft, LLC

We need more powerful tools !!

Calculation

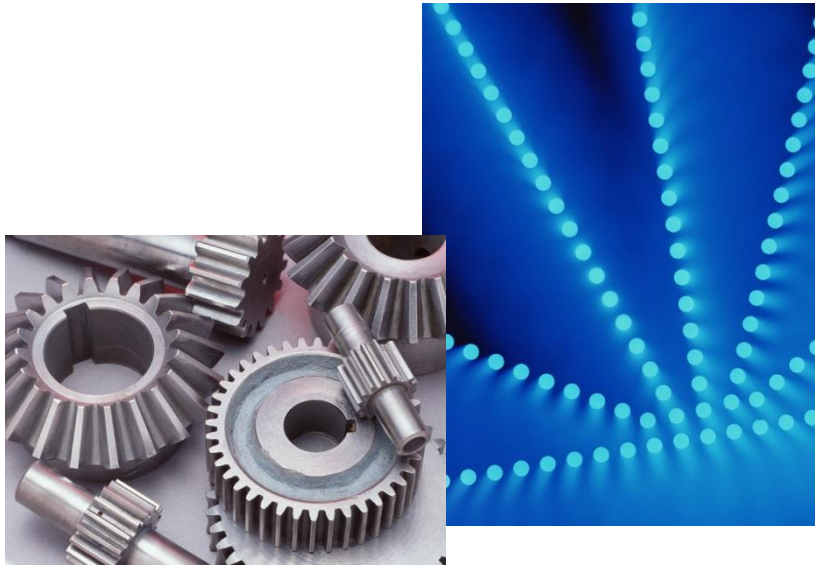
R, Python

AI/Machine learning

Visualization

Web based interactive framework

Easy to develop



©2018 OSIsoft, LLC

2. Interactive Advanced Data Analysis

MHPS Digital Solutions

TOMONI

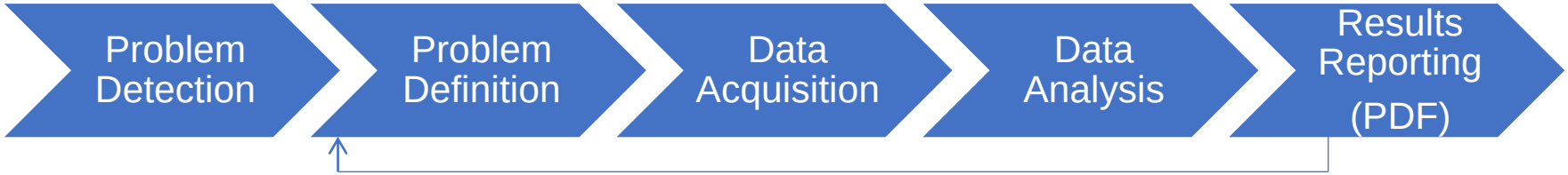
©2018 OSIsoft, LLC

Change of Paradigm for Data Analysis Report

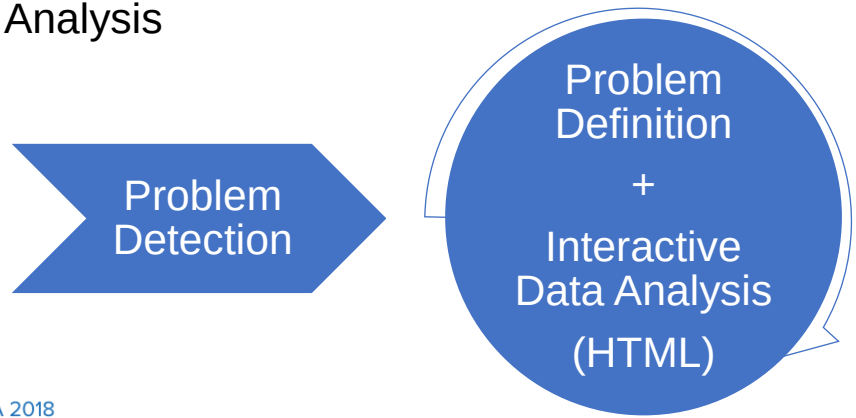
MHPS have shifted from Classical Data Analysis Report to Interactive Data Analysis.

Communication between issue owner and data scientists is improved.

Classic Data Analysis



Interactive Data Analysis



©2018 OSIsoft, LLC

Our Tools

Consistent PI historical data is leveraged by data analyst's tools like R and Python.

Mechanism is simple. Fast GUI implementation by short code.

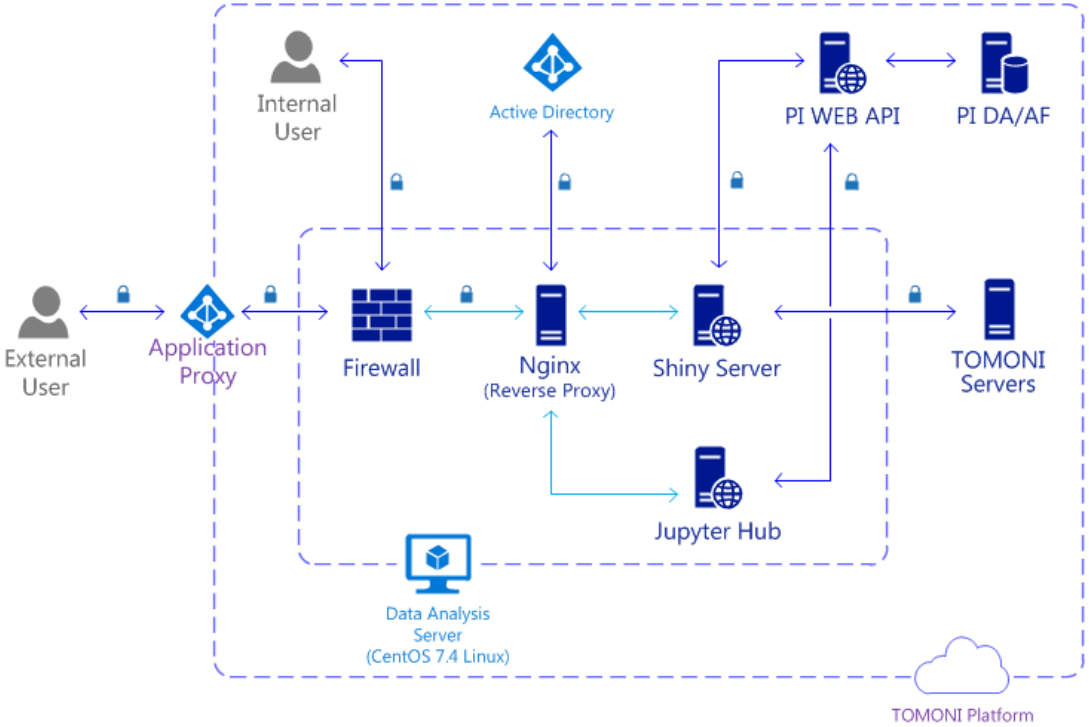
Data Consistency ▶ Analyst's Tools ▶ Fast GUI implementation ▶ Interactive Analysis

©2018 OSIsoft, LLC

Data Analysis System Architecture

Connections between Data Analysis Server and outside servers are secured connections.

A two-factor authentication is used to control access to TOMONI & Data Analysis Server.

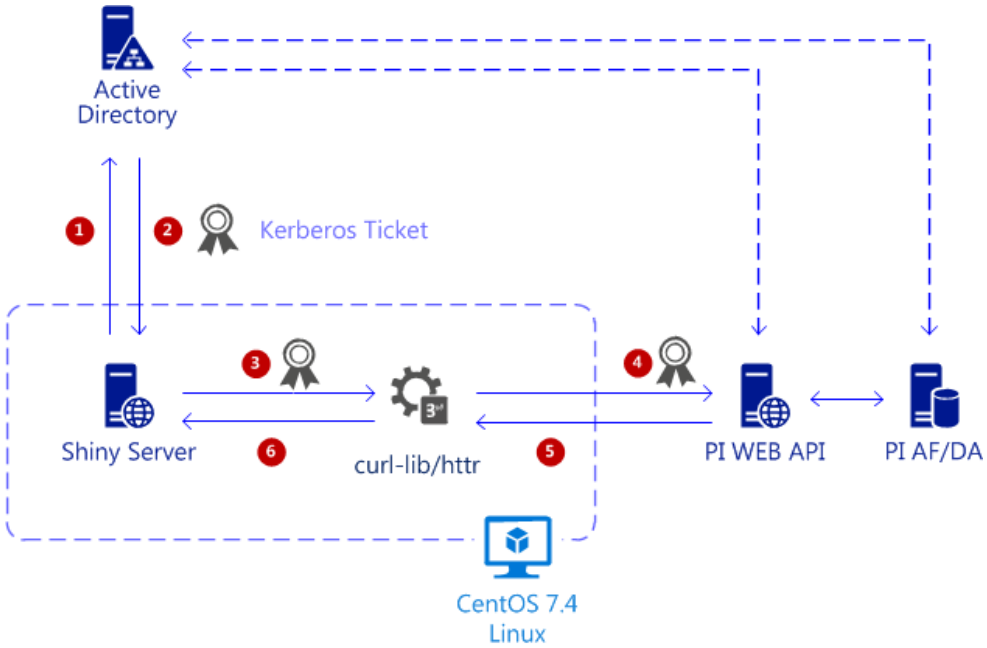


©2018 OSIsoft, LLC

Integration between Shiny Server and PI System

Shiny applications access PI data using PI Web API.

Security is hardened by using Kerberos instead of plain user / password.



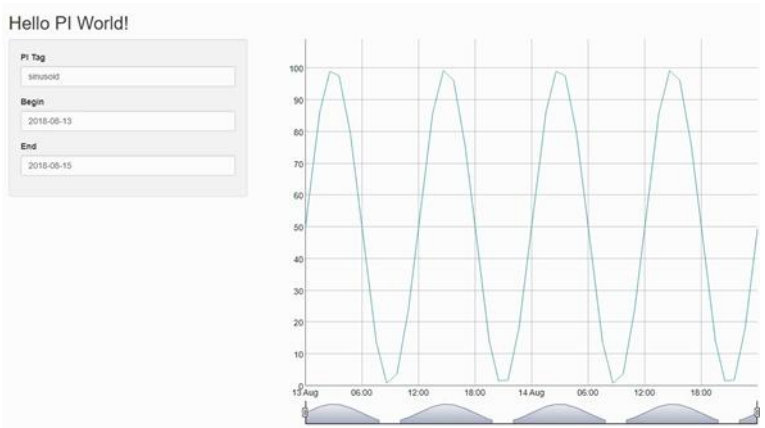
©2018 OSIsoft, LLC

Fast GUI implementation with Customized Logic - Shiny

Shiny applications access PI data using PI Web API.

Security is hardened by using Kerberos instead of plain user / password.

```
library(shiny)
library(dygraphs)
library(piwebapi)
library(xts)

server <- shinyServer(
    function(input, output) {
        output$dygraphsPlot <- renderDygraph({
            ### Secured access to PI System ###
            piWebApiService <- piwebapi$new("https://piserver.diasysnetmation.com/piwebapi/",
                                            TRUE, NULL, NULL, FALSE, FALSE)
            res <- piWebApiService$data$getInterpolatedValues(paste0("pi:\\\\SERVERDA\\", input$tagId),
                                                              startTime = format(input$dateBegin),
                                                              endTime = format(input$dateEnd), interval = "60s")
            ### Type formatting ###
            res$timestamp <- as.POSIXct(res$timestamp, format="%Y-%m-%dT%H:%M:%S", tz='GMT')
            res <- xts(res, order.by = res$timestamp)
            ### Plot drawing ###
            dygraph(res[,2:2]) %>% dyOptions(includeZero = TRUE) %>% dyRangeSelector()
        })
    }
)

ui <- fluidPage(
    titlePanel("Hello PI World!"),
    sidebarLayout(
        sidebarPanel(
            textInput("tagId", label = "PI Tag", value = "sinusoid"),
            dateInput("dateBegin", label = "Begin", value = (Sys.Date() - 7)),
            dateInput("dateEnd", label = "End", value = Sys.Date())
        ),
        mainPanel(
            dygraphOutput("dygraphsPlot", width = "100%", height = "640px")
        )
    )
)
### Shiny Application Ready ###
shinyApp(ui = ui, server = server, options = list(host = "0.0.0.0"))
```



©2018 OSIsoft, LLC

Fast GUI implementation with Customized Logic - Shiny

Shiny applications access PI data using PI Web API.

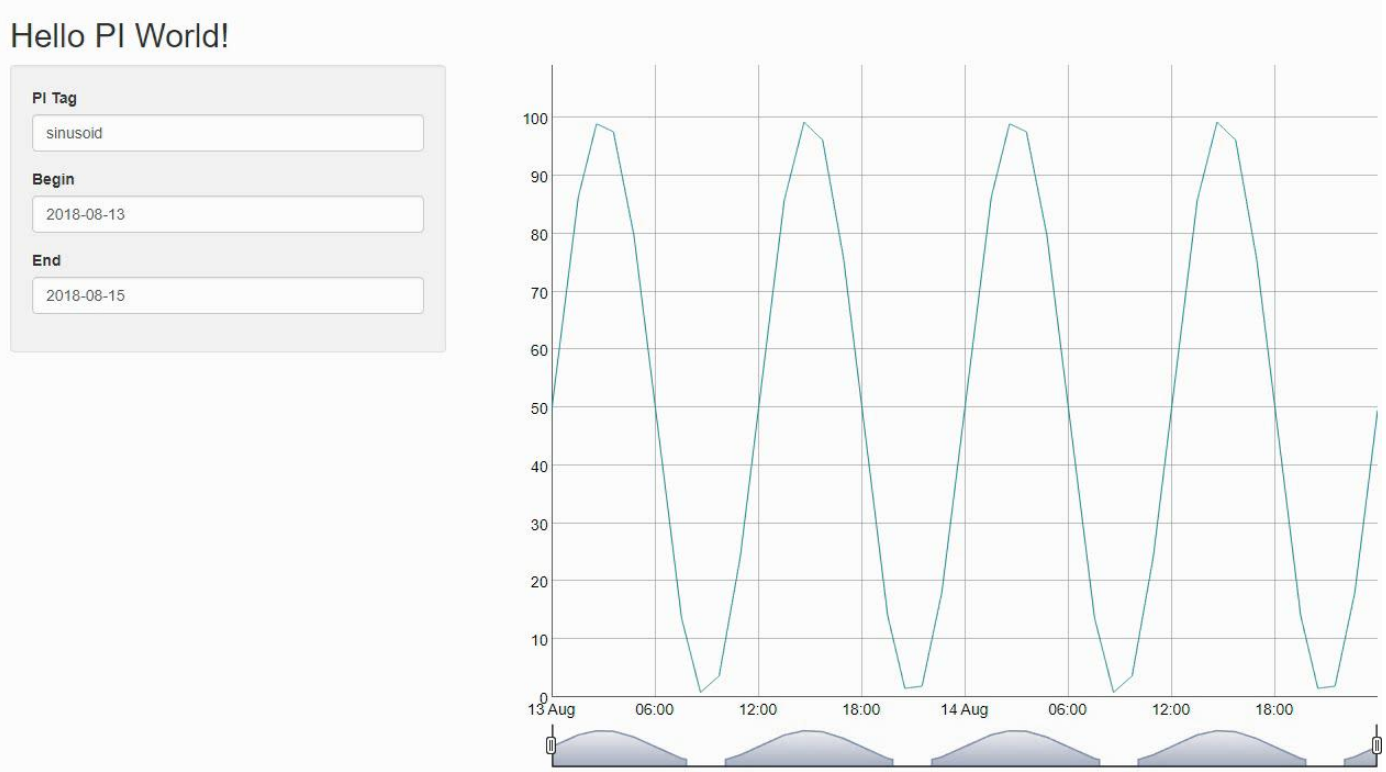
Security is hardened by using Kerberos instead of plain user / password.

```
library(shiny)
library(dygraphs)
library(piwebapi)
library(xts)

server <- shinyServer(
    function(input, output) {
        output$dygraphsPlot <- renderDygraph({
            ### Secured access to PI System ###
            piWebApiService <- piwebapi$new("https://piserver.diasysnetmation.com/piwebapi/",
                                            TRUE, NULL, NULL, FALSE, FALSE)
            res <- piWebApiService$data$getInterpolatedValues(paste0("pi:\\\\SERVERDA\\", input$tagId),
                                                              startTime = format(input$dateBegin),
                                                              endTime = format(input$dateEnd), interval = "60s")

            ### Type formatting ###
            res$timestamp <- as.POSIXct(res$ti
            res <- xts(res, ord
            ### Pi
```

```
piWebApiService <- piwebapi$new(
                            "https://piserver.diasysnetmation.com/piwebapi/",
                            TRUE, NULL, NULL, FALSE, FALSE)

            res <- piWebApiService$data$getInterpolatedValues(
                   paste0("pi:\\\\SERVERDA\\", input$tagId),
                            startTime = format(input$dateBegin),
                            endTime = format(input$dateEnd), interval = "60s")
```


©2018 OSIsoft, LLC

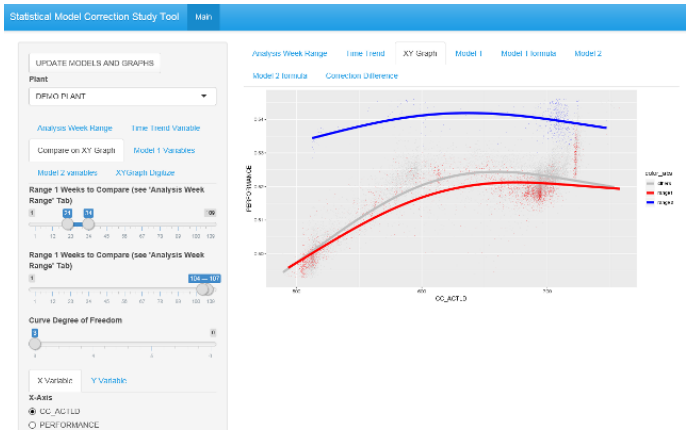
Fast GUI implementation with Customized Logic - Shiny



©2018 OSIsoft, LLC

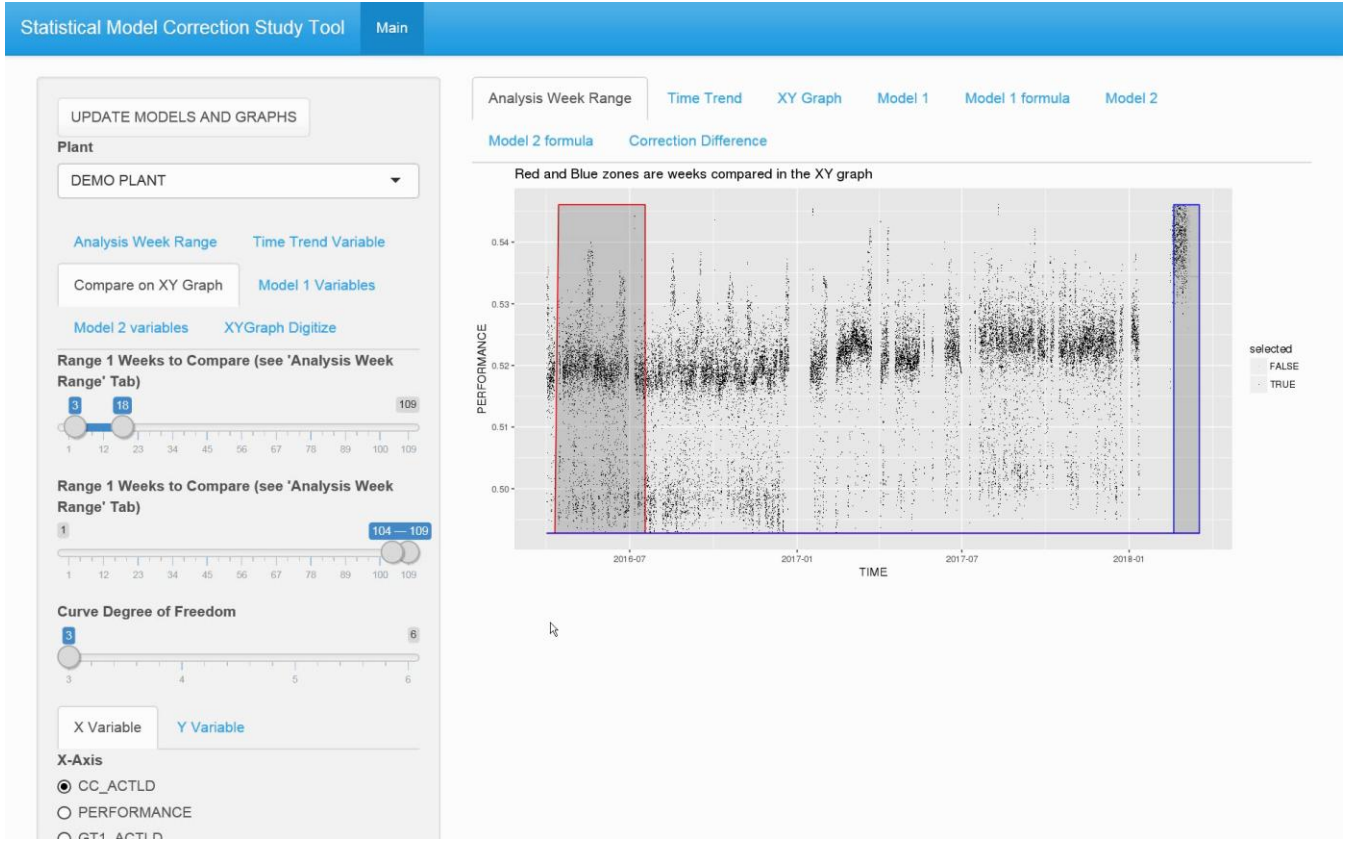
Example: Comparison of Plant Performance

The user can choose and shift the sampled data to compare plant performance between before and after upgrade event occur. Easy curve fitting and external effect removal is easy using statistic model.



©2018 OSIsoft, LLC

Video Demo



©2018 OSIsoft, LLC

In the future…

- Data analysis for all! Let's empower our engineers with data literacy!
- Continuous training to get more out of our data.

PI Web API could…

- Implement claim based authentication for PI Web API for R/Python. (Eg.: Oauth)
- Return data frames from PI Web API with adequate data types.
- Improve caching for repeated queries.

Transformation with data and cloud

©2018 OSIsoft, LLC

Mitsubishi Hitachi Power Systems

ADVANCED DATA ANALYSIS ENVIRONMENT WITH PI SYSTEM ON MHPS-TOMONI

CHALLENGE

Build interactive applications with advanced data analysis within a short time

- Difficult to obtain data from field
- Multiple programming languages / Multiple programmers needed

SOLUTION

Access PI System using PI Web API in R and developing web application using Shiny package

- PI System in the cloud provides real time data
- PI Web API interfaces with the programming language of your choice
- One programming language from data extraction to web application

RESULTS

Time to deploy interactive data analysis application reduced from 2 months to 2 days

- Application development needs one person instead of three (Field Engineer, Data Scientist, Web Developer)
- Improved communication between issue owner and data scientist

©2018 OSIsoft, LLC

- Hiroyasu Ishigaki
- General Manager, ICT department
- Mitsubishi Hitachi Power Systems, LTD.
- hiroyasu_ishigaki@mhps.com

- Leandro Hideo Iha
- Software and Control Engineer
- Mitsubishi Hitachi Power Systems, LTD.
- leandrohideo_iha@mhps.com

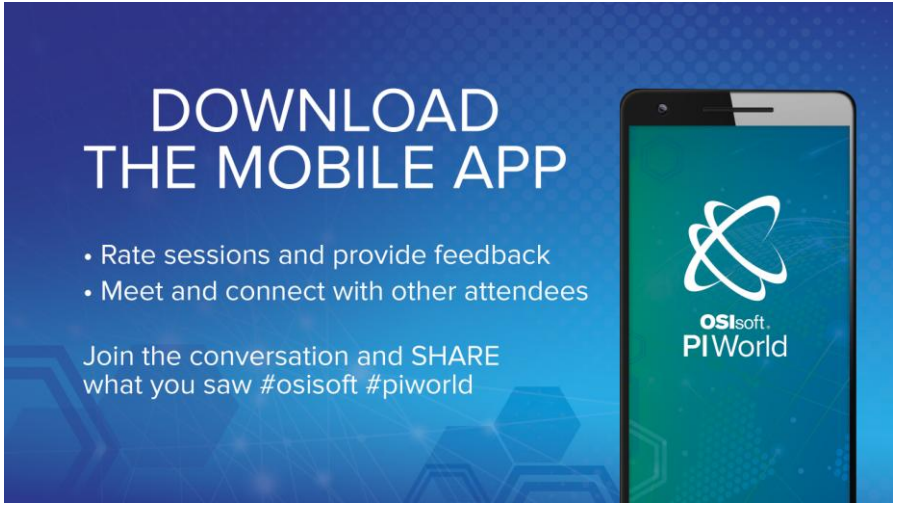
©2018 OSIsoft, LLC

Questions?

Please wait for the **microphone**

State your **name & company**

Please rate this session in the mobile app!



Download on the App Store

GET IT ON Google Play

©2018 OSIsoft, LLC

©2018 OSIsoft, LLC