

The background features a dark blue field on the left and a lighter blue field on the right, separated by a diagonal line. In the upper right, there are white, stylized wave patterns. In the lower right, there is a 3D wireframe plot with a color gradient from yellow at the top to blue at the bottom. Faint circuit-like lines are visible in the bottom right corner.

# MATLAB EXPO 2017

Integrate MATLAB Analytics into  
Enterprise Applications

Ionut Barbu, Application Engineer

# Data Analytics Workflow



**Business Data**

---

**Sensor Data**

**Data Reduction/  
Transformation**

---

**Feature Extraction**

**Model Creation**

---

**Model Validation**

**Enterprise Systems**

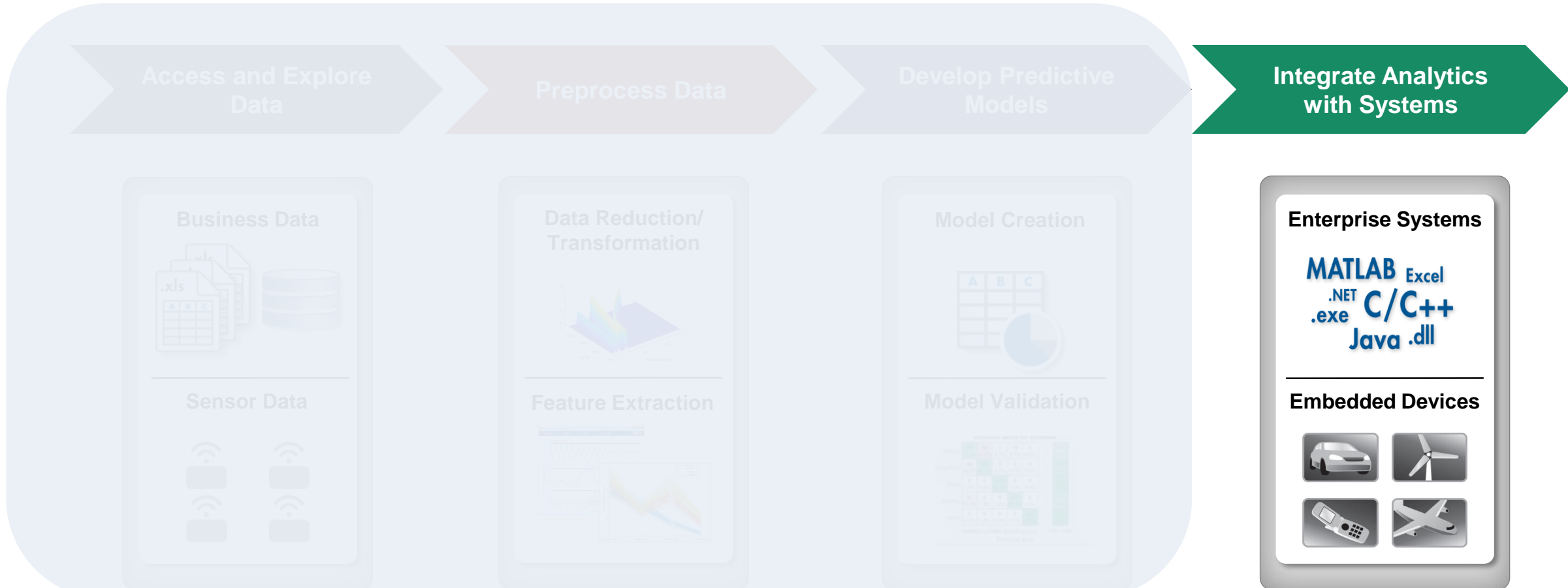
MATLAB Excel  
.NET C/C++  
.exe Java .dll

---

**Embedded Devices**

**MATLAB: Single Platform**

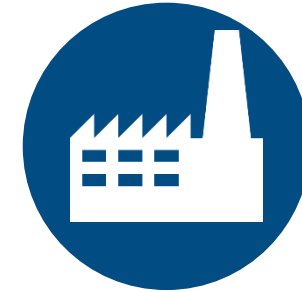
# Data Analytics Workflow



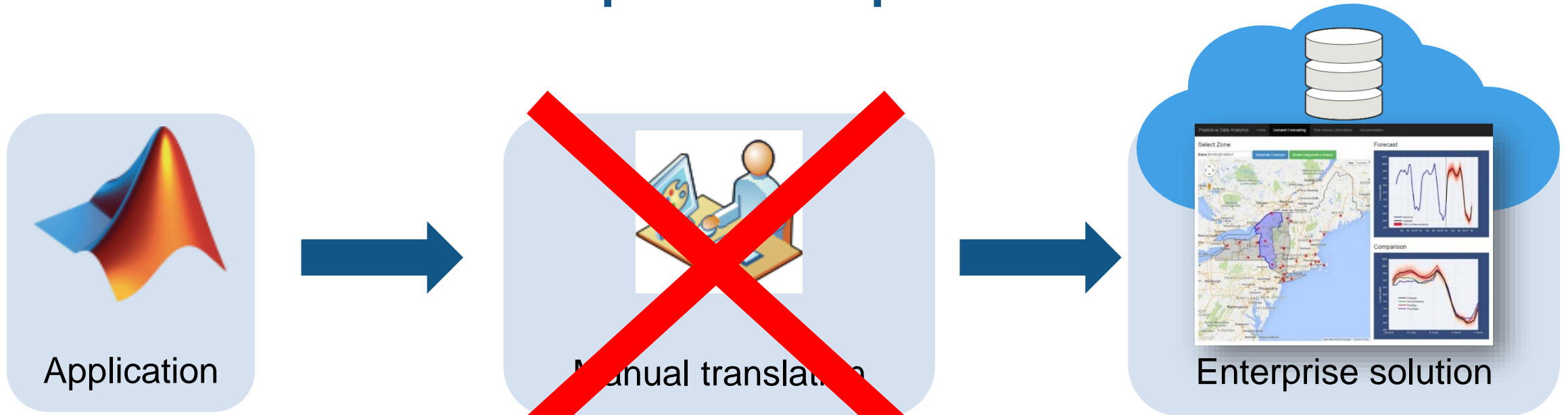
**MATLAB: Single Platform**

# Challenges

- Bridge the gap between multiple disciplines
- Integrate solutions to enterprise scale frameworks
- Deliver fast results with large volumes of data



# Different tools for development and productization

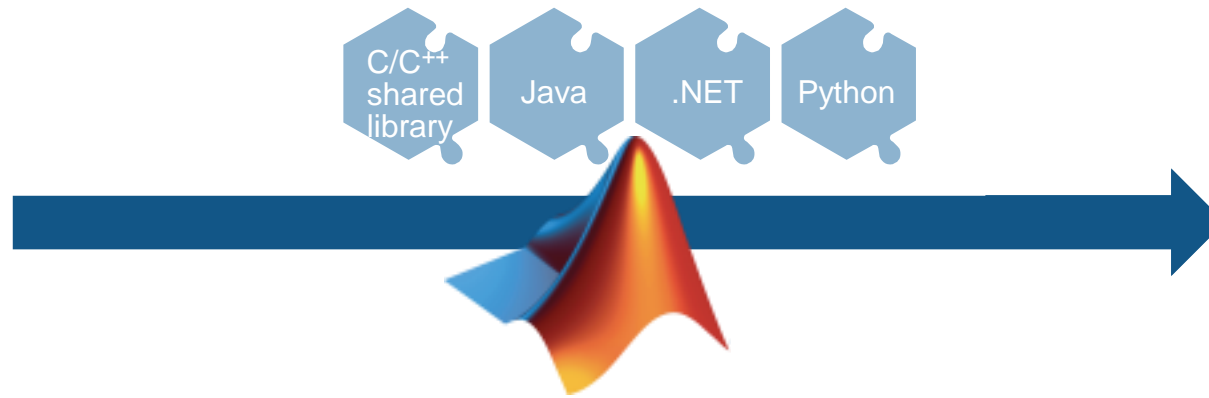


**Domain Expert**



**Solution Architect**

# What if you speed up the integration process?



**Domain Expert**

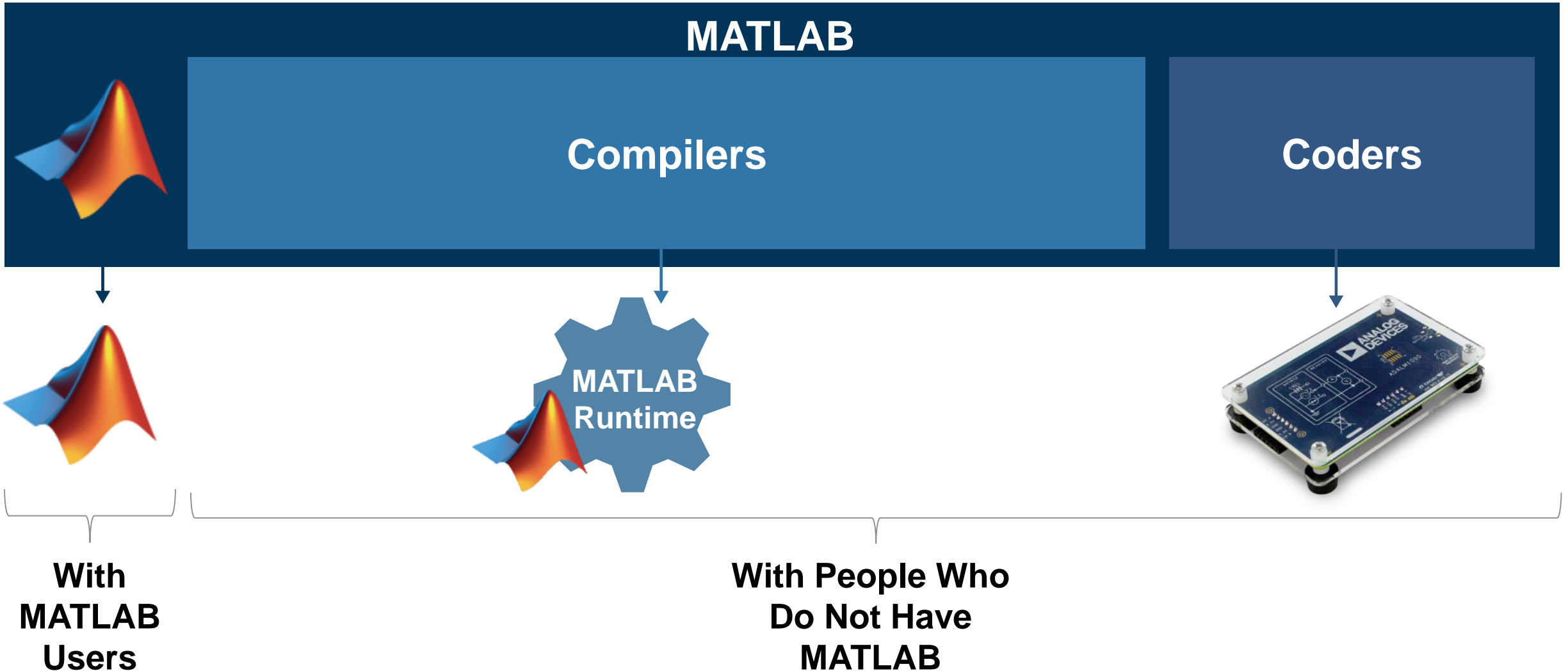
**with automatic deployment**



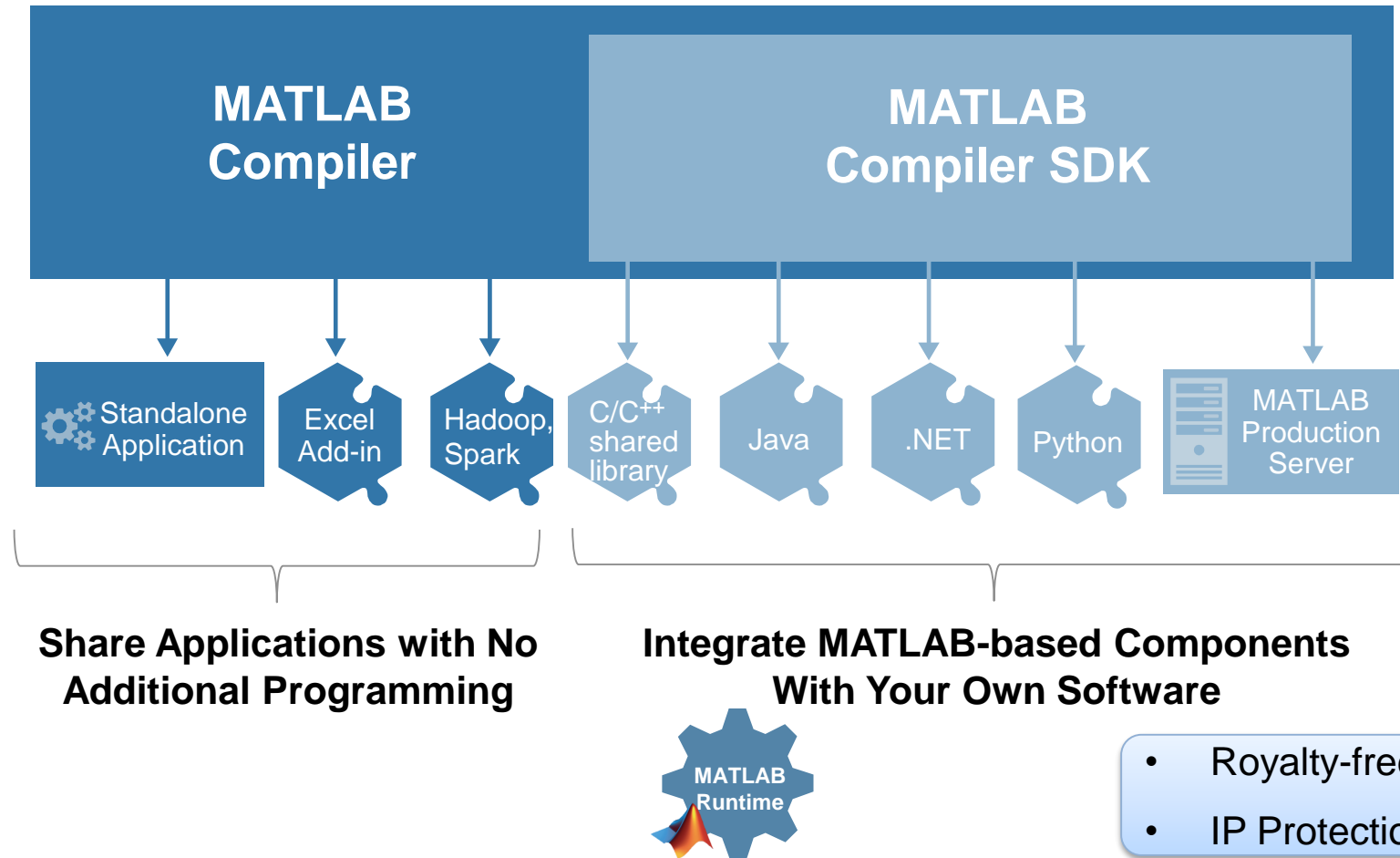
**Solution Architect**

# Sharing and Deploying MATLAB Applications

Write Your Programs Once, Then Share to Different Targets

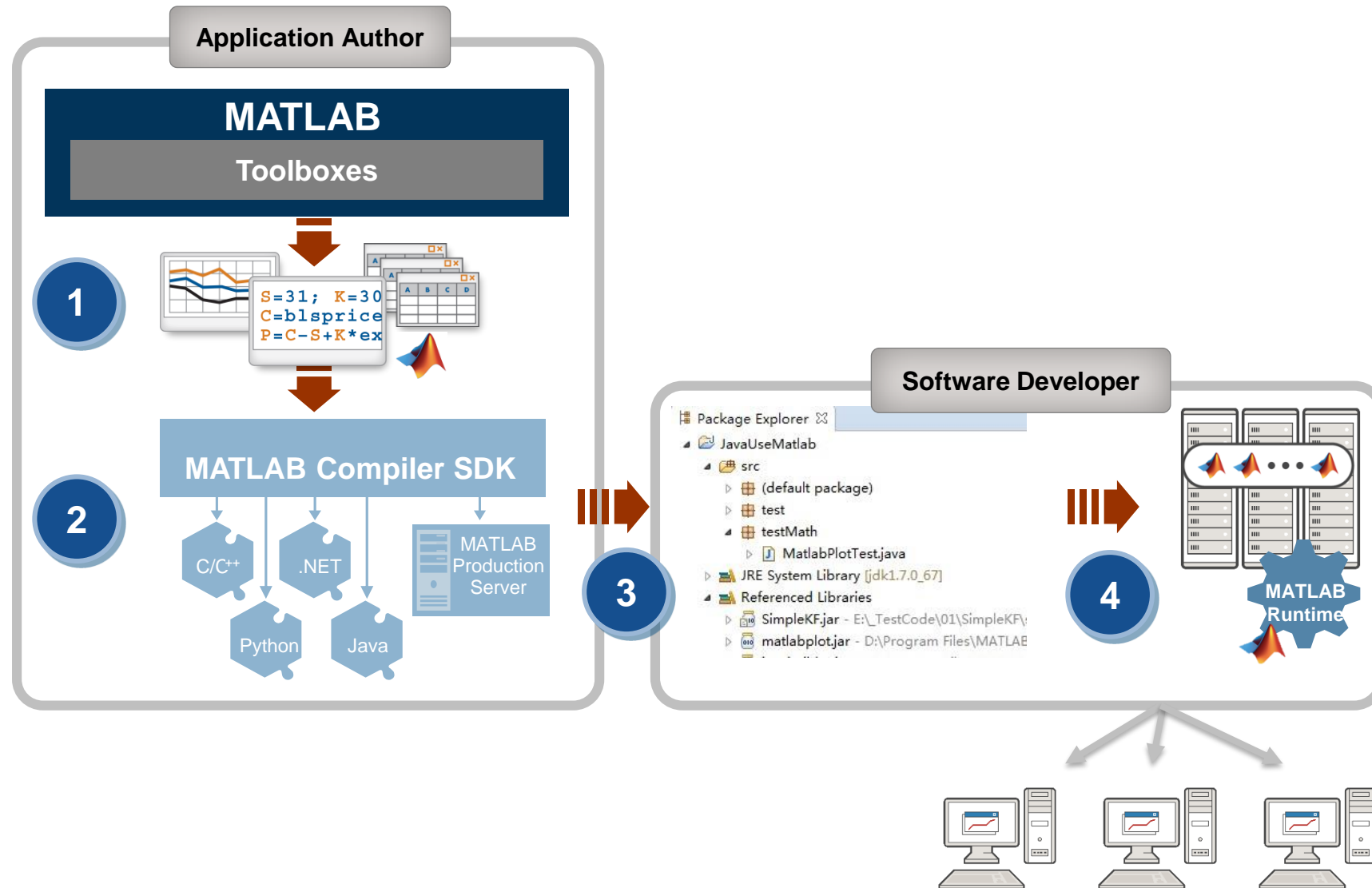


# Share with People Who Do Not Have MATLAB





# Integrate MATLAB-based Components With Your Own Software



# Using MATLAB Compiler SDK to create Java Classes

The screenshot shows the MATLAB Compiler interface for a project named 'SolarPredictorJA'. The 'COMPILER' tab is active, showing a list of exported functions: 'createPredictorFit.m', 'loadPredictorData.m', and 'makePrediction.m'. The 'PACKAGING OPTIONS' section is visible, with 'Runtime downloaded from web' selected for 'MyAppInstaller\_web' (5 MB) and 'Runtime included in package' selected for 'MyAppInstaller\_mcr' (1 GB). A 'Package' dialog box is open in the foreground, displaying three icons: a Java class file (101 010), a folder, and a package icon, each with a green checkmark. The dialog includes an 'Open output folder' button and a checked checkbox for 'Open output folder when process completes'. The background shows the 'Library information' section with fields for 'SolarPredictorJA' (version 1.0), author 'Roland Michaely', email 'Roland.Michaely@mathworks.ch', and company 'MathWorks'. Below this, there are sections for 'Additional installer options', 'Files required for your library to run' (listing 'createModelFro...', 'DavisDaily.xlsx', and 'importSolarData.m'), and 'Files installed for your end user' (listing 'doc', 'readme.txt', and 'SolarPredictorJA.j...').

# Using MATLAB Compiler SDK to create Java Classes

The screenshot displays the MATLAB IDE interface with a Java application window titled "Solar Ratio Predictor Model" running. The application window includes a control panel on the left and a 3D plot area.

**Control Panel:**

- Buttons: "Use Default Data", "Load Data", "Display Model"
- Text: "Example Using MWArray API:", "Predict Solar Ratio", "DavisDaily"
- Input fields:
  - Temp. Change (F): 20
  - Rel. Humidity: 80
  - Solar Ratio: 0.52

**3D Plot:**

The plot shows a 3D scatter of blue data points and a fitted surface. The axes are labeled:
 

- Vertical axis: Solar Ratio (0 to 1)
- Horizontal axis (left): Relative Humidity (0 to 100)
- Horizontal axis (right): Temperature Change (°F) (50 to 100)

**IDE Elements:**

- Project Explorer:** Shows the project structure for "DetailedSolarAnalyzerJA", including folders like ".idea", "build", "nbproject", "out", "src", and files like "build.xml", "manifest.mf", and "SolarPredictorJA.jar".
- Run Console:** Shows the execution output for "DetailedSolarAnalyzerJA", including a warning and several lines of progress text.
- Code Editor:** Shows a snippet of Java code:
 

```
the model
...
d1.getText());
d2.getText());
...
(coefficients)
```

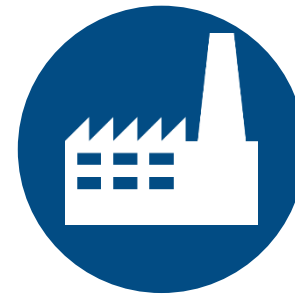
The status bar at the bottom indicates "All files are up-to-date (moments ago)", "14 chars", "255:69", "CRLF", "UTF-8", "Git: master", and other IDE settings.

# MATLAB *and* MATLAB Production Server

is the **easiest** and most **productive** environment to *take your enterprise analytics or Internet of Things solution* from **idea** to **production**



**Idea**



**Production**

# Energy Load Forecast

Electricity Demand Forecast

← → ↻ ec2-54-165-201-58.compute-1.amazonaws.com:8080/DemandForecastWeb/demandForecast.jsp

Predictive Data Analytics
Home
Demand Forecasting
Web Service Description
Documentation

### Select Zone

Zone  Generate Forecast Model Diagnostics Report

**Plattsburgh International Airport**

Station ID: KPBG

Contributes to zones: NYISO D-North(100%), NYISO F-Capitl(5%)

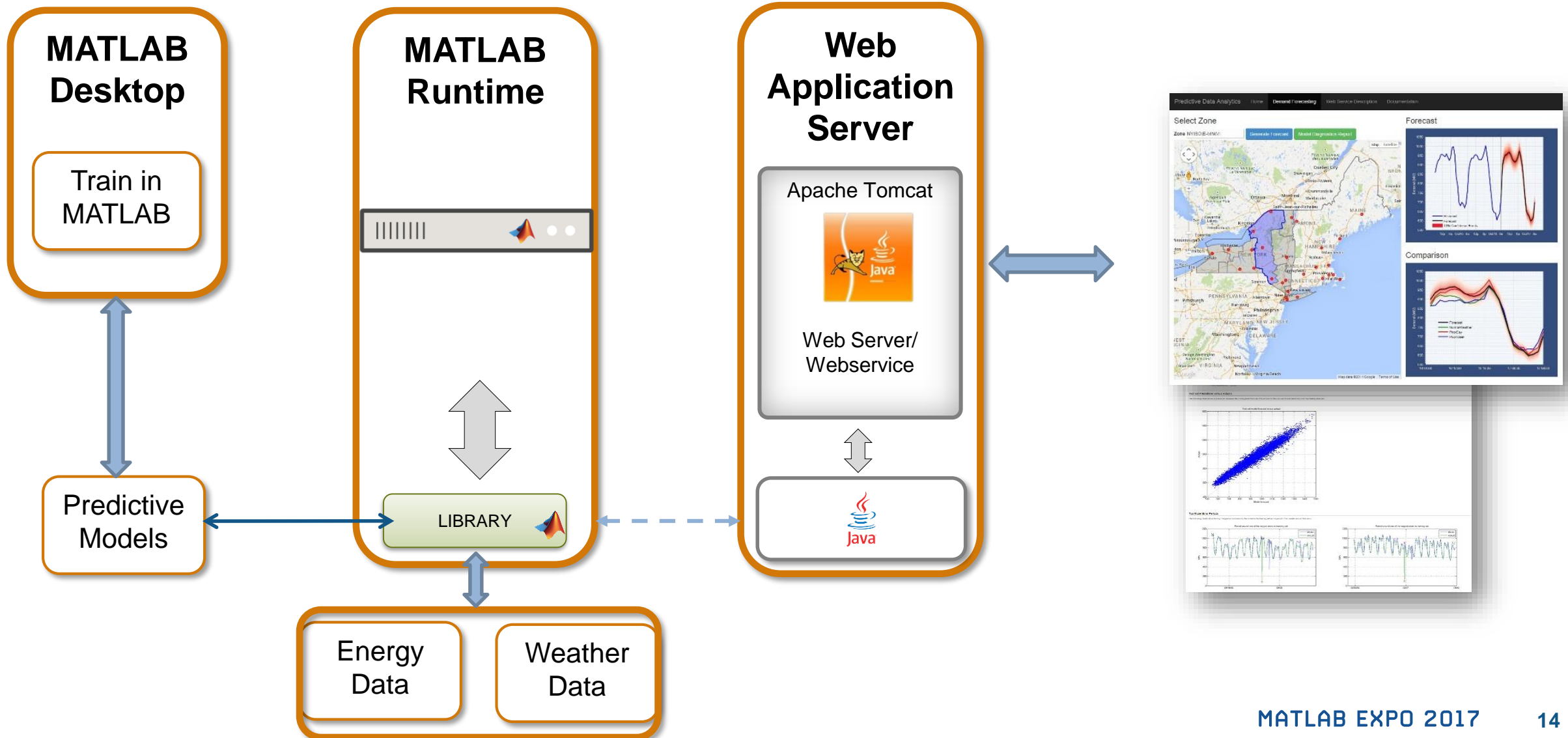
**Weather forecast**

Map Satellite

### Forecast

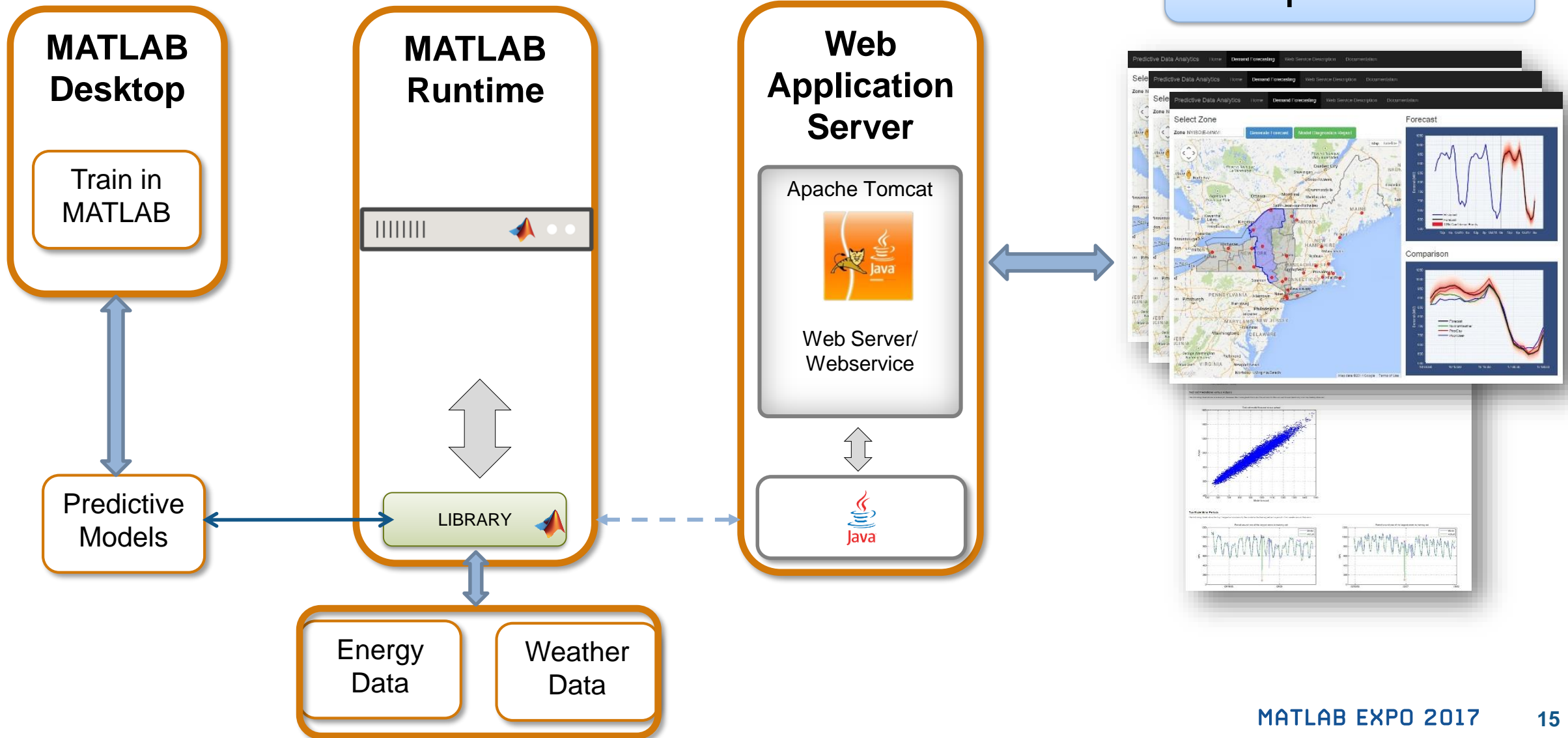
### Comparison

# Energy Load Forecast

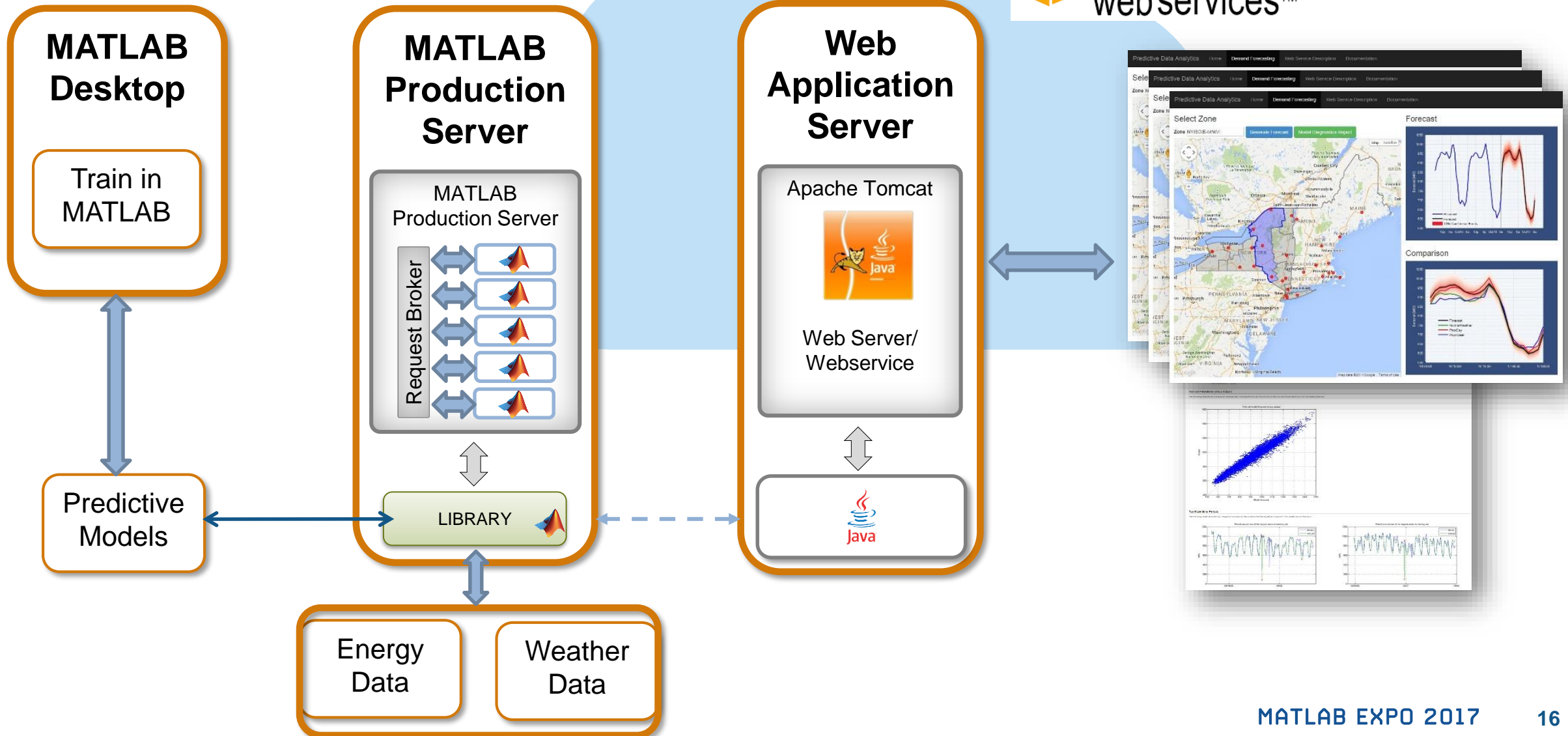


# Energy Load Forecast

Multiple users



# Energy Load Forecast

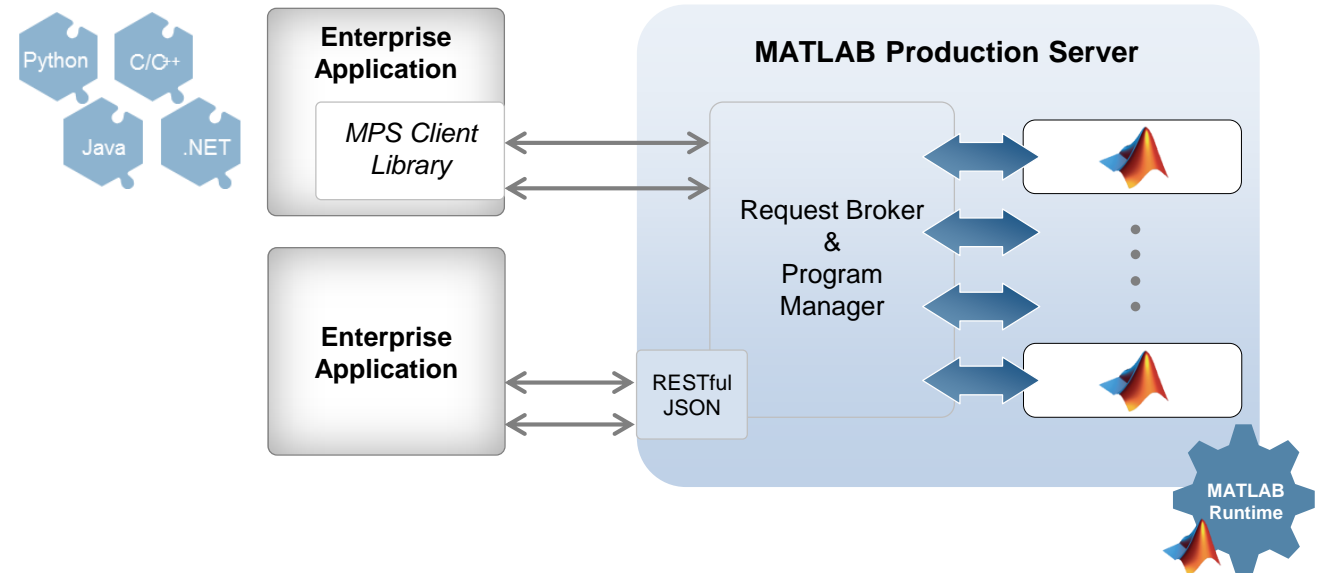




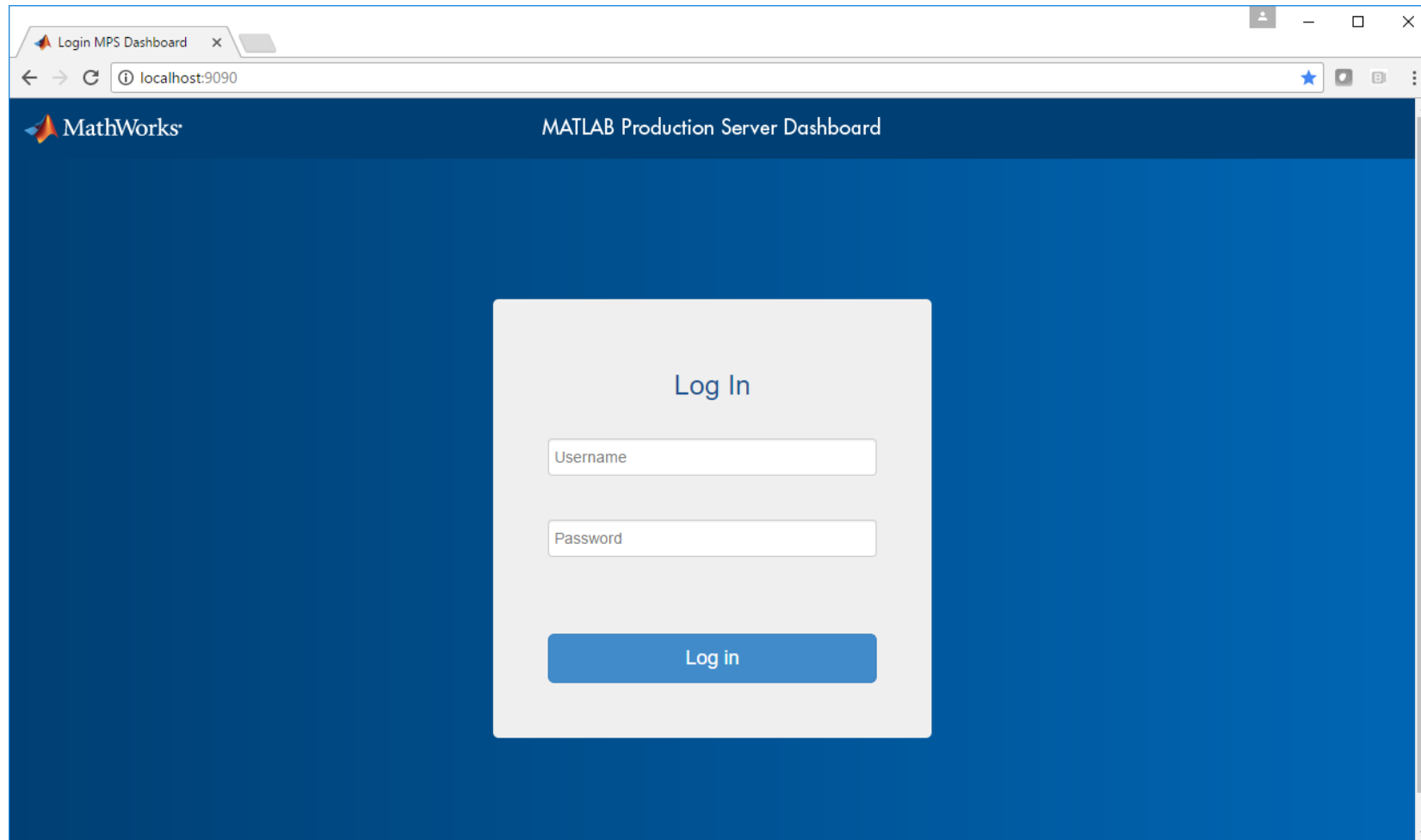
# MATLAB Production Server

## Enterprise Class Framework For Running Packaged MATLAB Programs

- Server software
  - Manages packaged MATLAB programs and worker pool
- MATLAB Runtime libraries
  - Single server can use runtimes from different releases
- RESTful JSON interface and lightweight client library (C/C++, .NET, Python, and Java)



# Manage Your Server Instances Using a Dashboard Interface



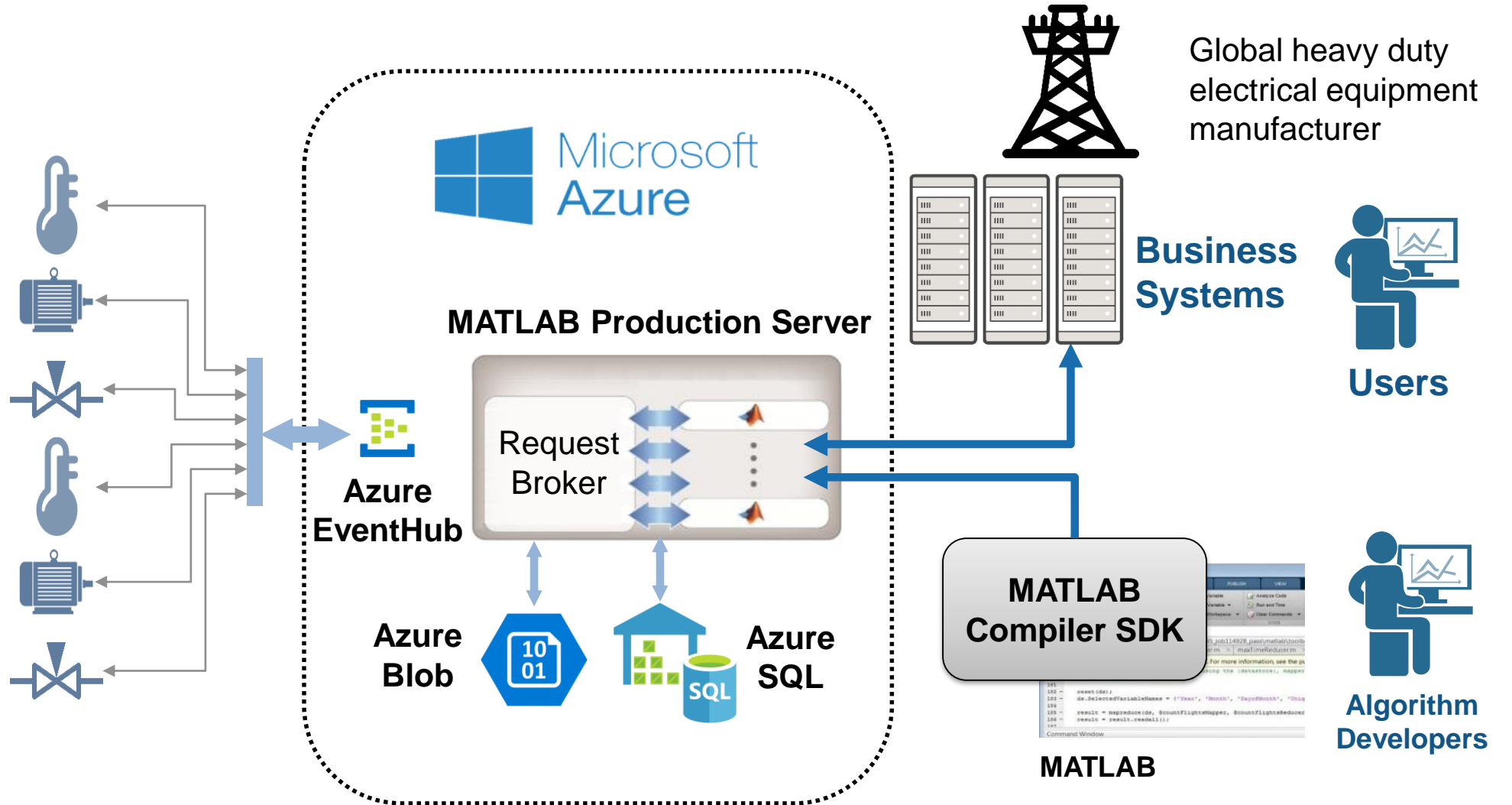
# Manage Your Server Instances Using a Dashboard Interface

The image displays two overlapping screenshots of the MATLAB Production Server Dashboard. The top screenshot shows the 'Main\_Production' instance, which is in a 'Running' state with an up time of 0:00:05:23. The bottom screenshot shows the 'myMPS' instance, also in a 'Running' state with an up time of 0:00:02:41. Both instances have detailed information panels, including their IDs (mps\_2 and mps\_1), descriptions, HTTP/HTTPS ports, creation and modification dates, and performance metrics like CPU usage, worker processes, and queue lengths. Activity charts at the bottom of each instance view show the number of requests to complete (orange bars) and available workers (blue line) over a period of time.

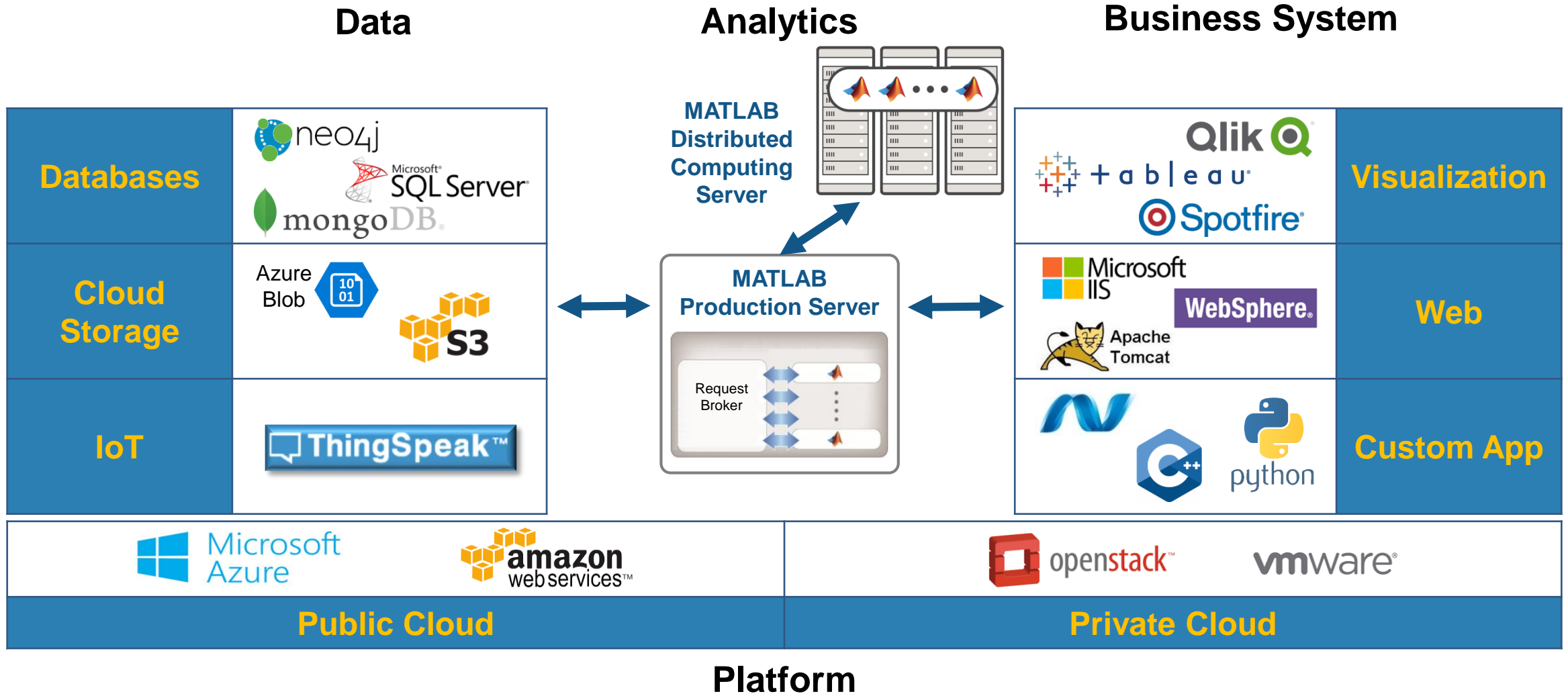
# Building Automation IoT Analytics on Azure

## Building/HVAC automation control system

- Variety of sensors and controls
- Networked communication
- Data reduction



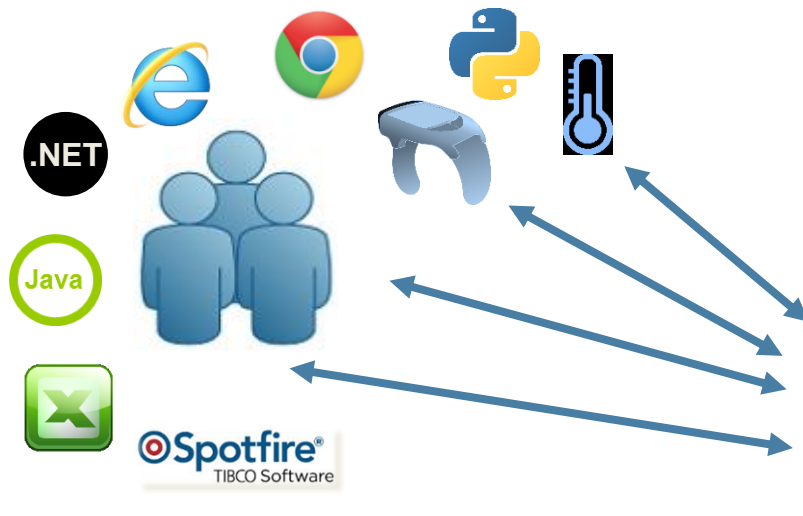
# Technology Stack



## Front-end Scalability

### Application server for MATLAB

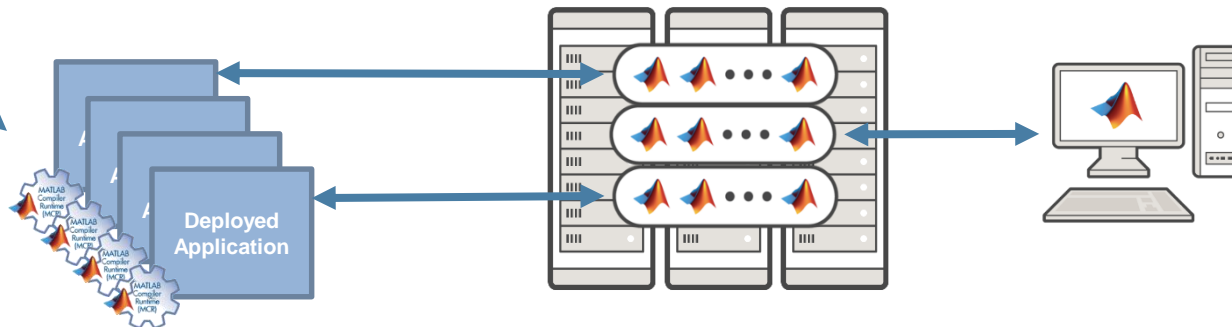
- Manage large numbers of requests to run deployed MATLAB programs



## Back-end Scalability

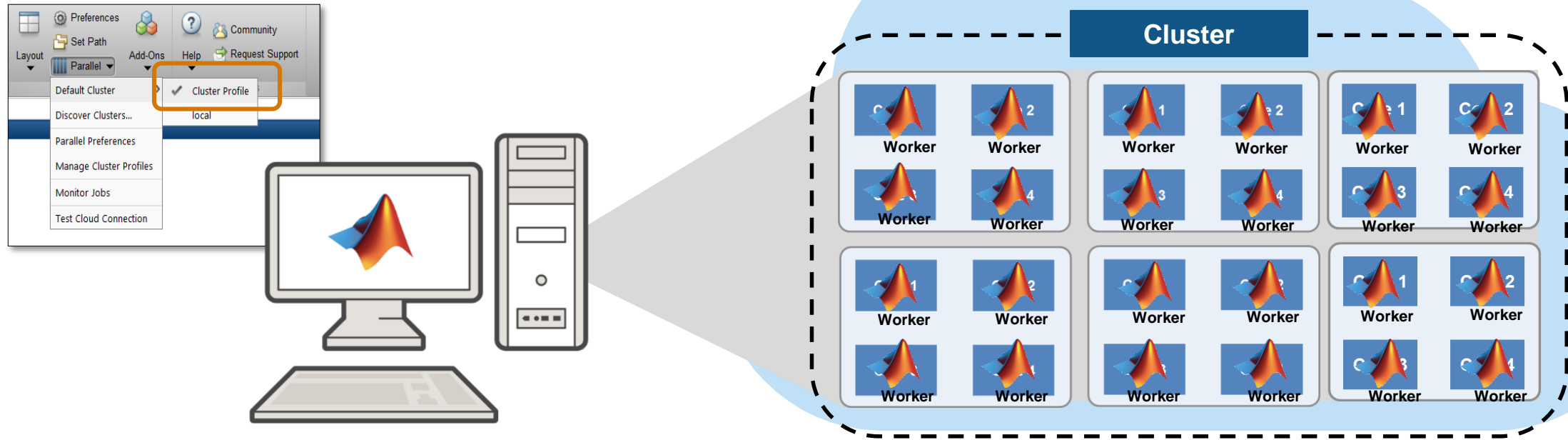
### Cluster framework for MATLAB

- Speed up computationally intensive programs on computer clusters, clouds, and grids



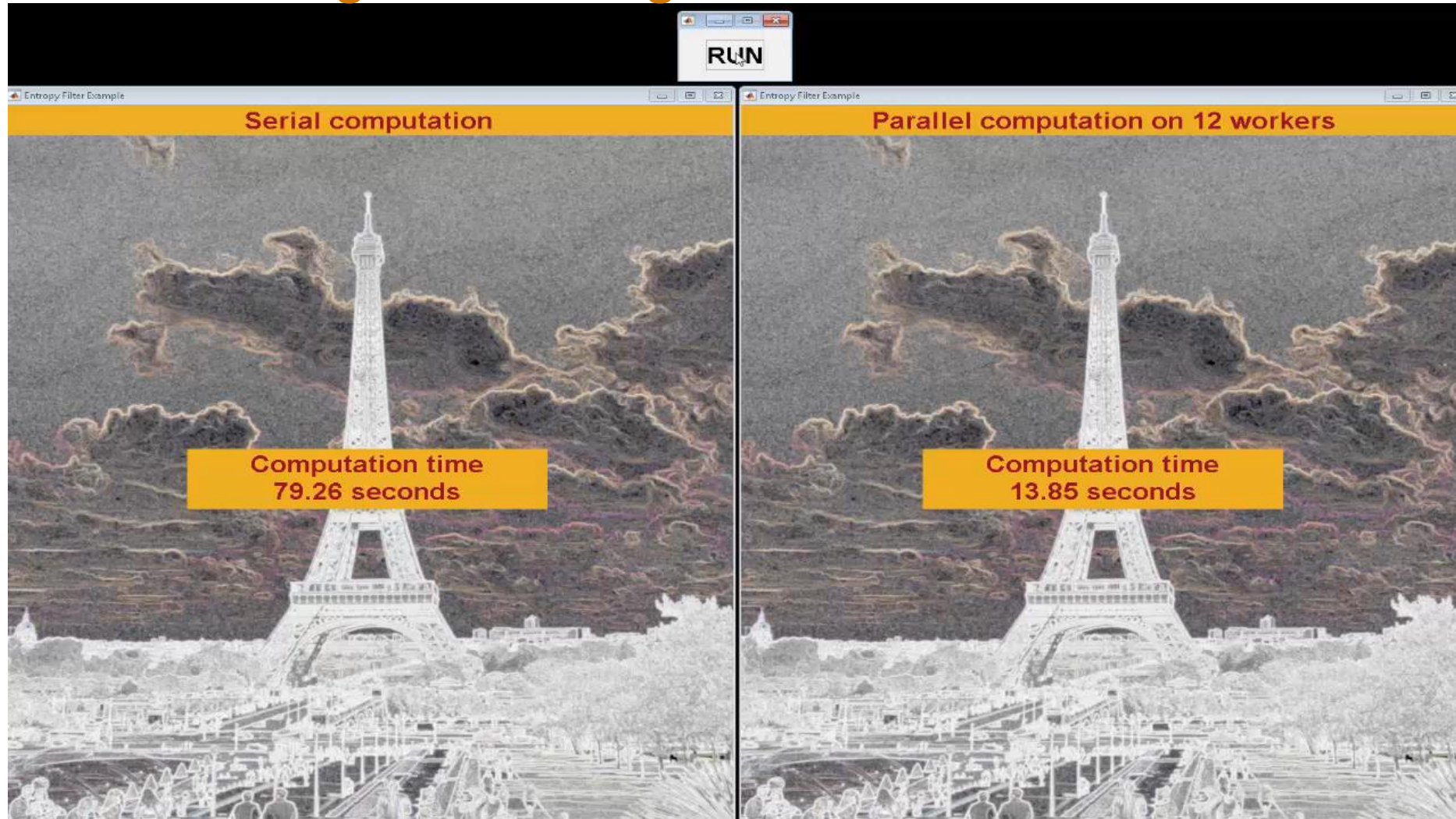
# Parallel Computing Paradigm

## Clusters



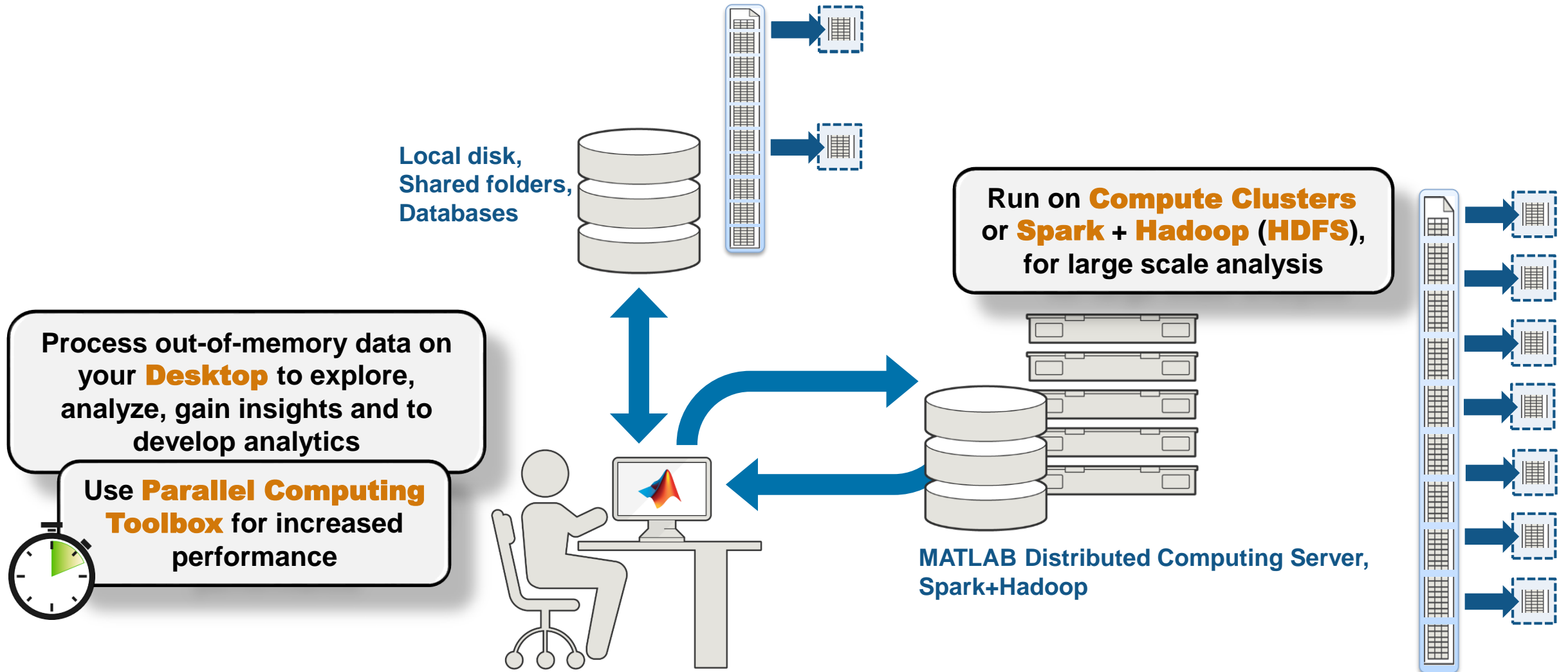
# Speed-up using Multiple Cores on the Cloud

## High Resolution Image Processing

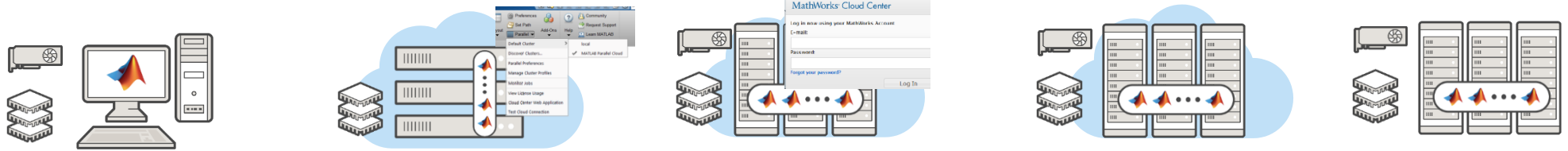




# Big Data Workflow



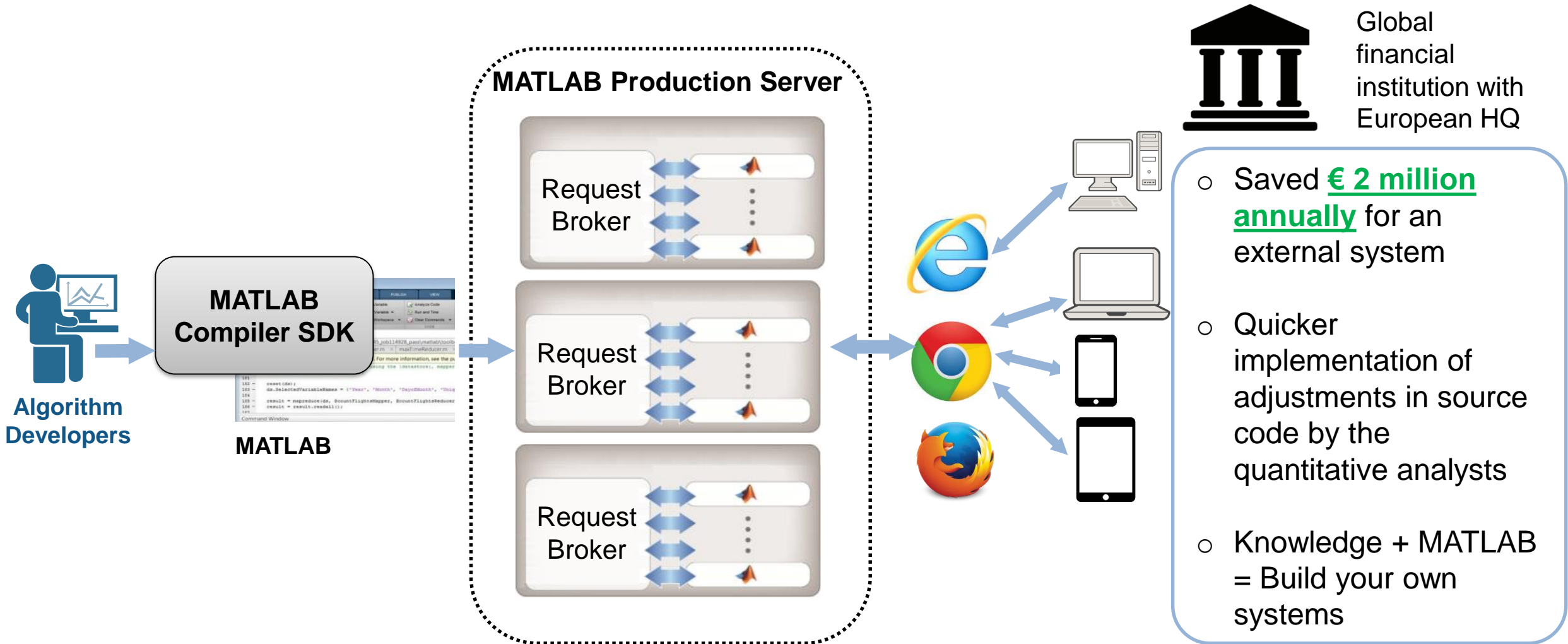
# Scale your Applications Beyond the Desktop



Option	Parallel Computing Toolbox	MATLAB Parallel Cloud	MATLAB Distributed Computing Server for Amazon EC2	MATLAB Distributed Computing Server for Custom Cloud	MATLAB Distributed Computing Server
Description	Explicit desktop scaling	Single-user, basic scaling to cloud	Scale to EC2 with some customization	Scale to custom cloud	Scale to clusters
Maximum workers	No limit	16	256	No limit	No limit
Hardware	Desktop	MathWorks Compute Cloud	Amazon EC2	Amazon EC2, Microsoft Azure, Others	Any
Availability	Worldwide	United States and Canada	United States, Canada and other select countries in Europe	Worldwide	Worldwide

Learn More: [Parallel Computing on the Cloud](#)

# Customer Example: Financial Customer Advisory Service



Global financial institution with European HQ

- Saved **€ 2 million annually** for an external system
- Quicker implementation of adjustments in source code by the quantitative analysts
- Knowledge + MATLAB = Build your own systems

# How to get started?



- Data Analytics
- Application Development
- Code Generation

# MATLAB<sup>®</sup>

## Data Analytics

Data Processing and visualization  
 Statistics  
 Machine Learning  
 Optimization Techniques  
 Parallel Computing

## Application-Specific

Control System Design  
 Signal Processing  
 Communication Systems  
 LTE Systems

## Application Development

Programming Techniques  
 Building Interactive Applications  
 Object-Oriented Programming

## Computational Finance

Risk Management  
 Time-Series Modelling

## Code Generation

MATLAB Coder  
 Interfacing with C-code

## Signal Processing

Using MATLAB  
 Using Simulink

## Image and Video Processing

Image Processing  
 Computer Vision

# SIMULINK<sup>®</sup>

## Model-Based Design

Implementing MBD Workflow  
 Model Management and Architecture  
 Verification and Validation

## Code Generation

Rapid Prototyping and HIL-Simulation  
 Embedded Systems  
 FPGA Design  
 Generating HDL Code  
 Xilinx Zynq SoCs  
 AUTOSAR

## STATEFLOW<sup>®</sup>

Event-Based Modeling

## Code Integration

Integrating C and MATLAB

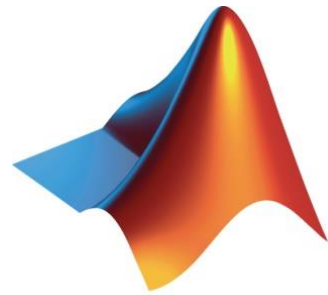
## Simscape<sup>™</sup>

General Simscape<sup>™</sup>  
 Simscape Multibody<sup>™</sup>  
 Simscape Driveline<sup>™</sup>  
 Simscape Fluids<sup>™</sup>  
 Simscape Power Systems<sup>™</sup>

## Polyspace<sup>®</sup>

Polyspace Code Prover<sup>™</sup>

<https://nl.mathworks.com/services/training.html>



# MathWorks®

*Accelerating the pace of engineering and science*

© 2017 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.