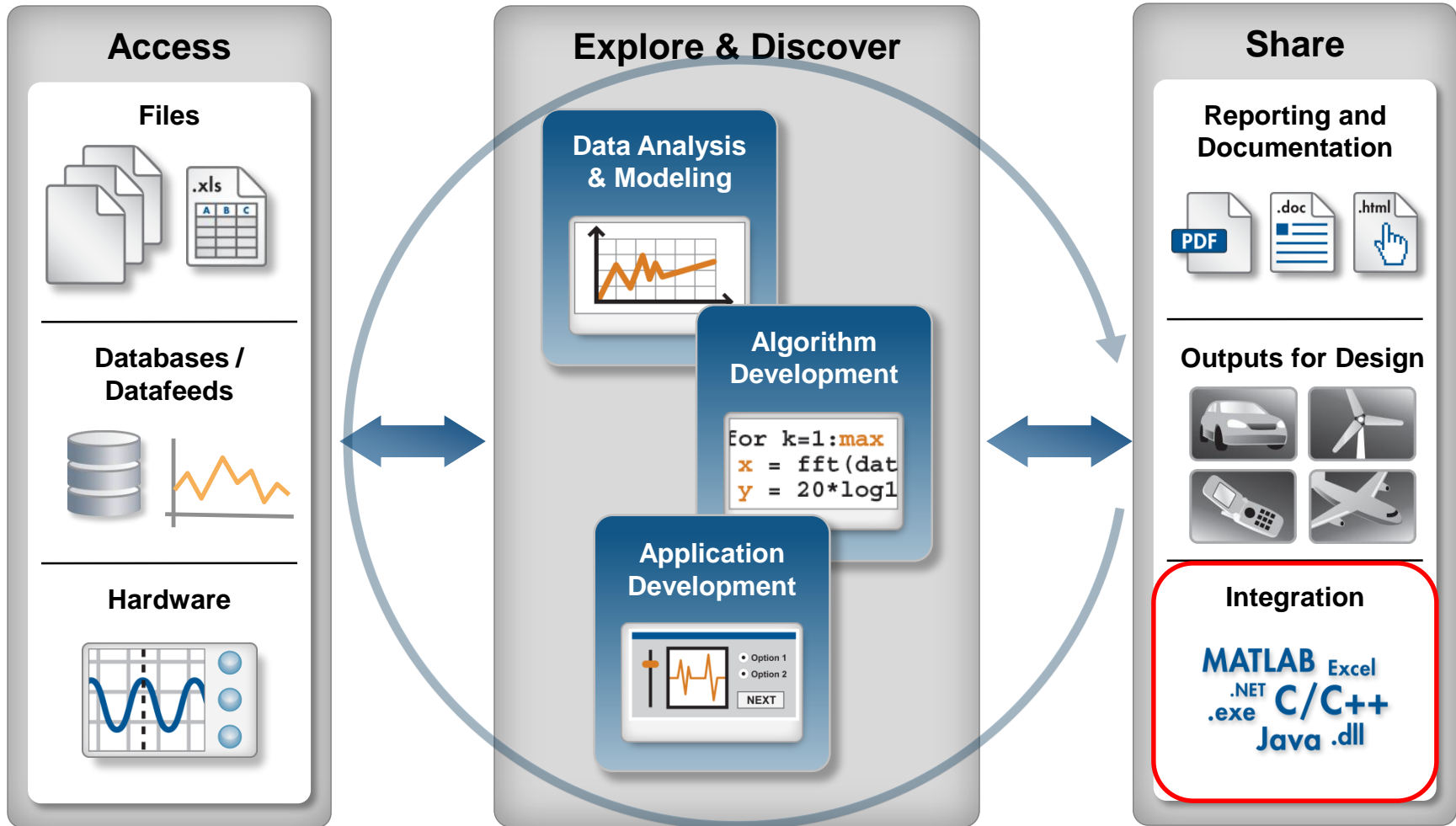


# Deploying MATLAB<sup>®</sup>-based Applications

**David Willingham**  
**Senior Application Engineer**

# Data Analytics Workflow



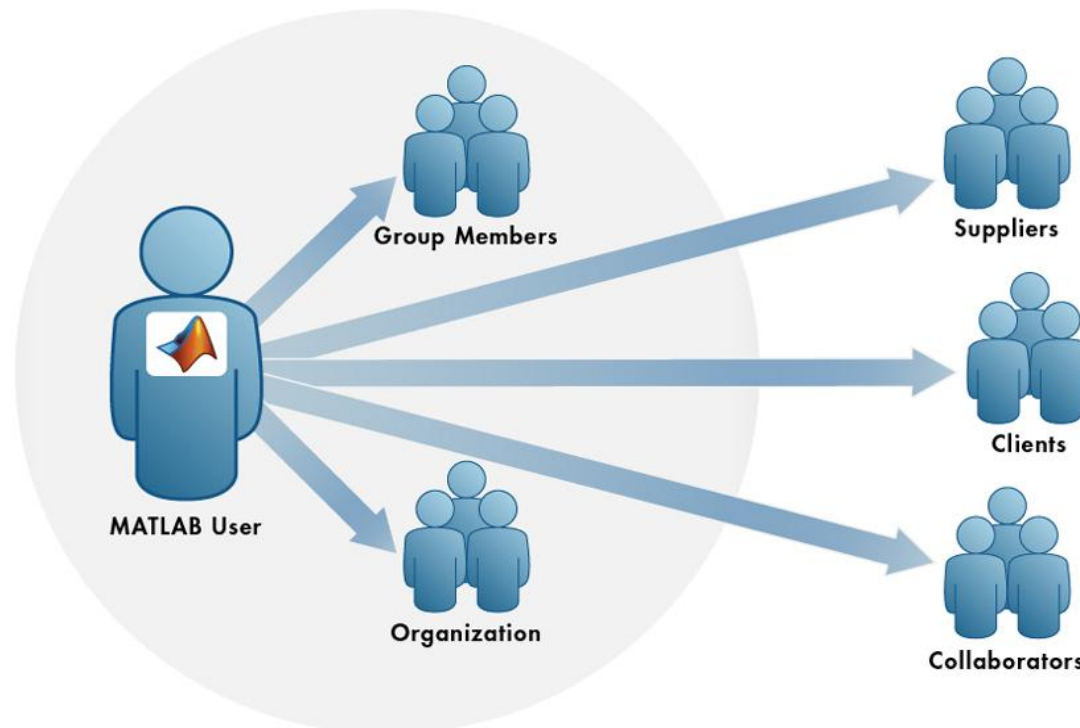
Automate

# What is *Application Deployment*?

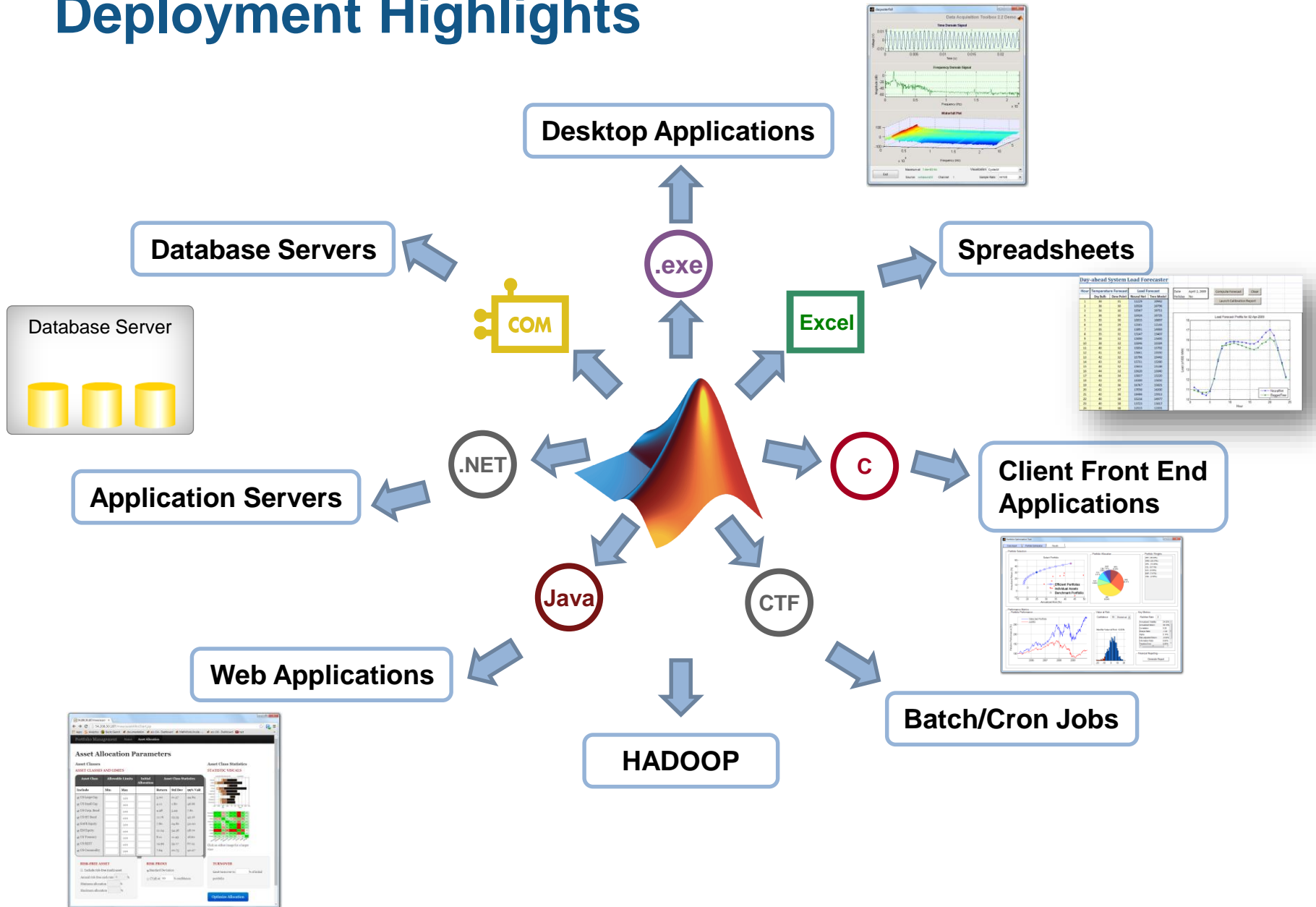
Share MATLAB programs with people who do not have MATLAB

- Royalty-free distribution

Provide MATLAB apps or native files directly to other MATLAB users

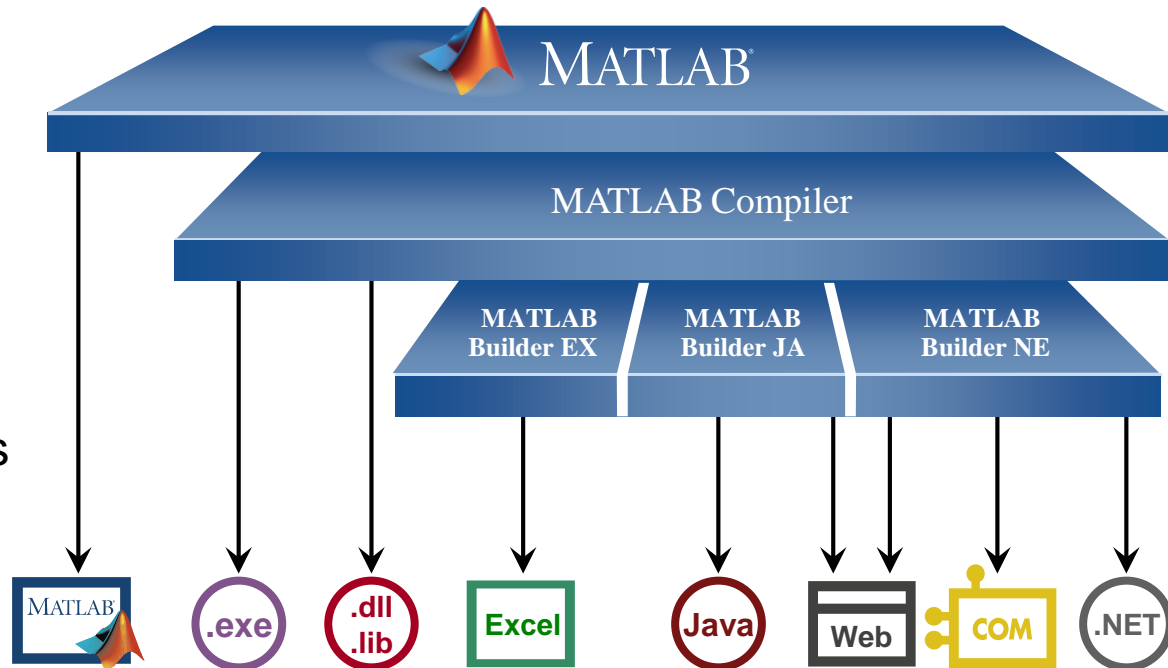


# Deployment Highlights

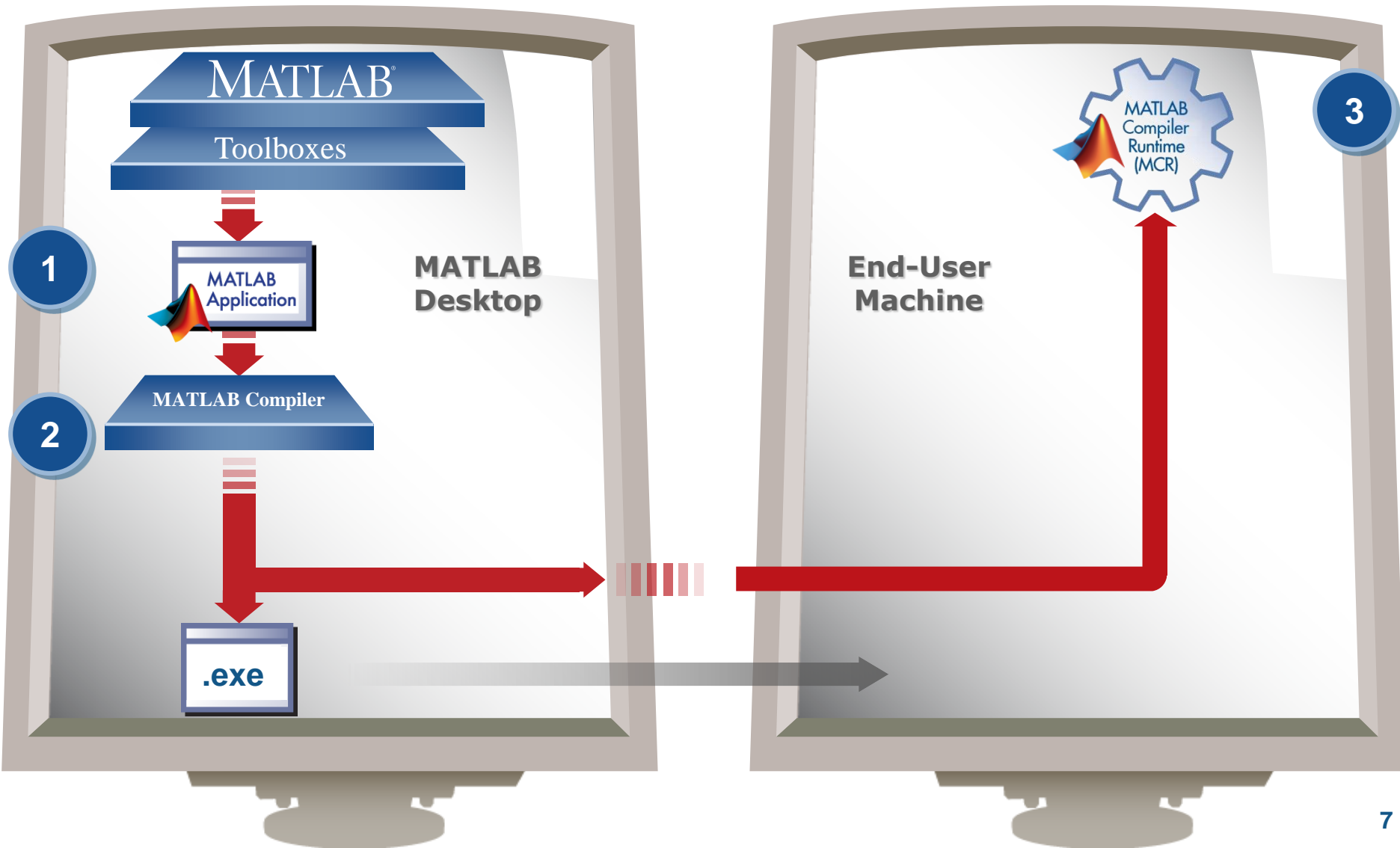


# Deploying Applications with MATLAB

- Automated deployment
- Share applications with end users who do not need MATLAB
  - Stand-alone executables
  - Shared libraries
  - Software components
  - Encrypted
- Uses MATLAB Component Runtime Libraries

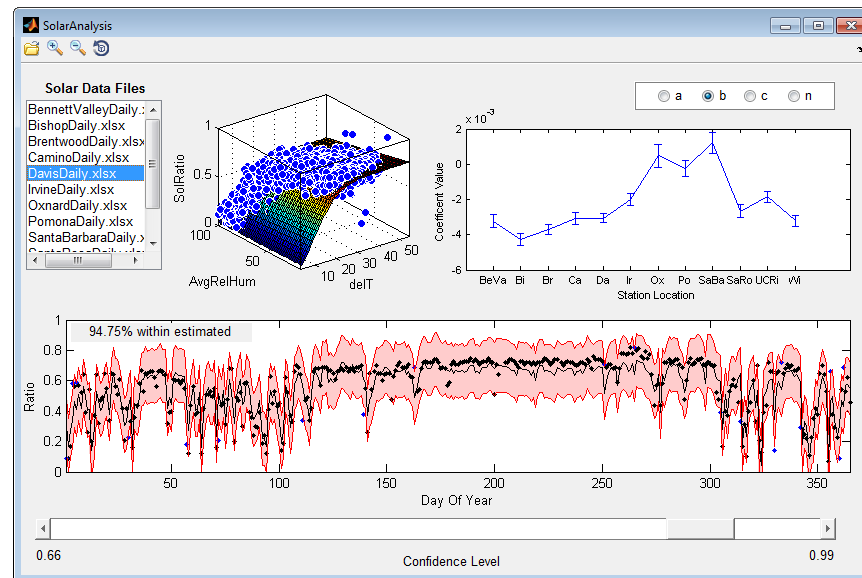


# Deploying Applications with MATLAB



# Capabilities of MATLAB Compiler™

- Package MATLAB programs as standalone applications or shared libraries
- Create professional software with customizable installers, icons, and splash screens
- Encrypt your intellectual property



*A deployed application created with MATLAB Compiler*

# Typical Process for Standalone Applications

**Application author**

The screenshot shows the MATLAB environment with a script editor containing MATLAB code for solar analysis. A separate window displays a graphical user interface with various controls like buttons and sliders. Another window shows the 'MATLAB Compiler - SolarAnalysis.m' dialog, where the application is configured as a standalone application.



- 1.) Create MATLAB algorithms
- 2.) Define the user interface
- 3.) Package the application using MATLAB Compiler
- 4.) Give the application installer to someone

They will install the application ... and run it on their desktop

The screenshot shows the 'Installation Options' dialog box where the installation folder is set to 'C:\Program Files\SolarAnalysis'. Below it, the application's user interface is displayed, featuring a list of solar data files, a 3D surface plot of solar radiation, a line graph of coefficient values, and a time-series plot of solar radiation over a year.



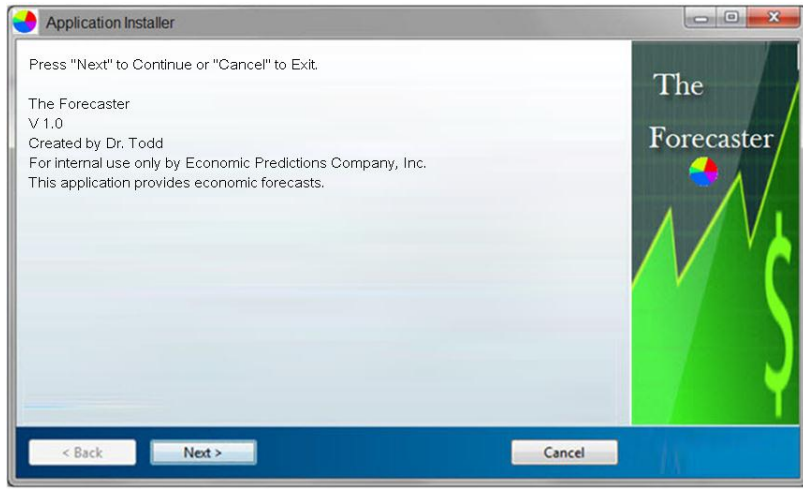
# Customizations for your Applications

The screenshot shows the MATLAB Compiler interface for configuring an application. Key features and callouts include:

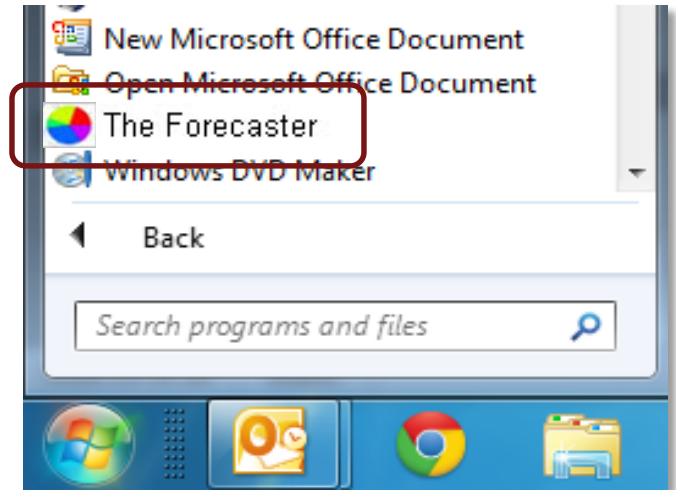
- Icons:** A callout box labeled "Icons" points to a dialog box on the left showing three icon sizes (48x48, 32x32, 16x16) and a "Select icon" button.
- Runtime downloaded from web:** A callout box labeled "Runtime downloaded from web" points to a checkbox in the "PACKAGING OPTIONS" section.
- Metadata:** A central callout box labeled "Metadata" has arrows pointing to the "Application Name", "Author Name", "Email", "Company", "Summary", and "Description" fields in the "Application Information" section.
- Graphics for splash screen:** A callout box labeled "Graphics for splash screen" points to a "Select custom splash screen" button next to a colorful abstract graphic.
- Graphics for installer:** A callout box labeled "Graphics for installer" points to a "Select custom logo" button next to a similar colorful abstract graphic.
- Installed applications:** A callout box labeled "Installed applications accessible from Windows Start menu and Add/Remove Programs" points to the application's entry in the Windows Start menu.

# End Customer sees a Professional Application

## Installer



## Application in Start menu



## Splash Screen

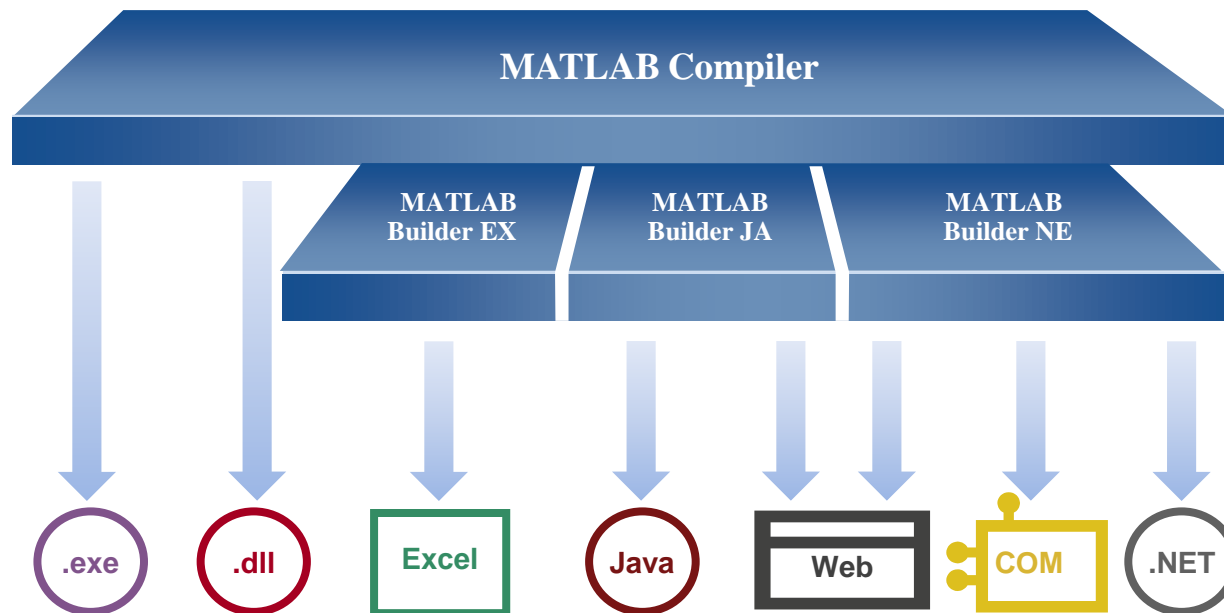


## Icon



# MATLAB Builder™ Products

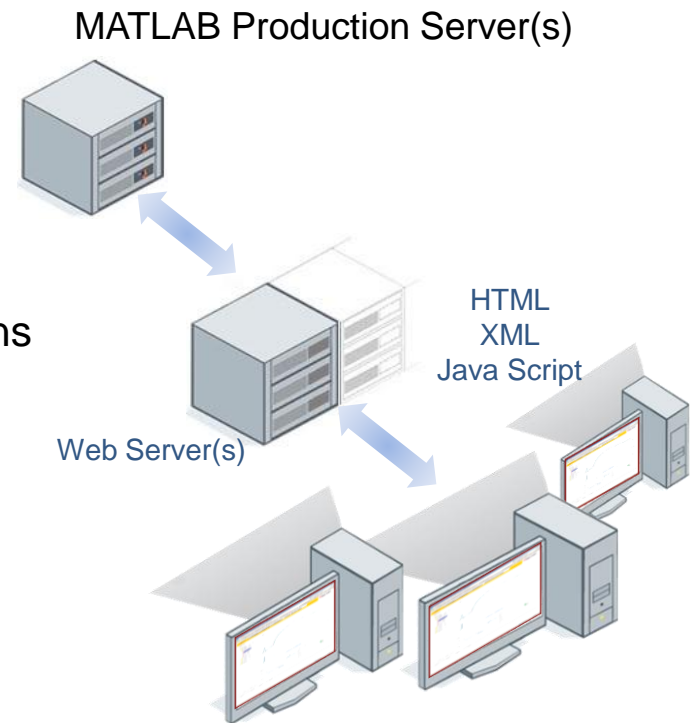
- Generate add-ins, components, and libraries to integrate with Microsoft Excel®, .NET, and Java™
- Support web technologies such as ASP.NET, SOAP, XML, JavaScript, and HTML
- Scalable applications via Java RMI and .NET remoting



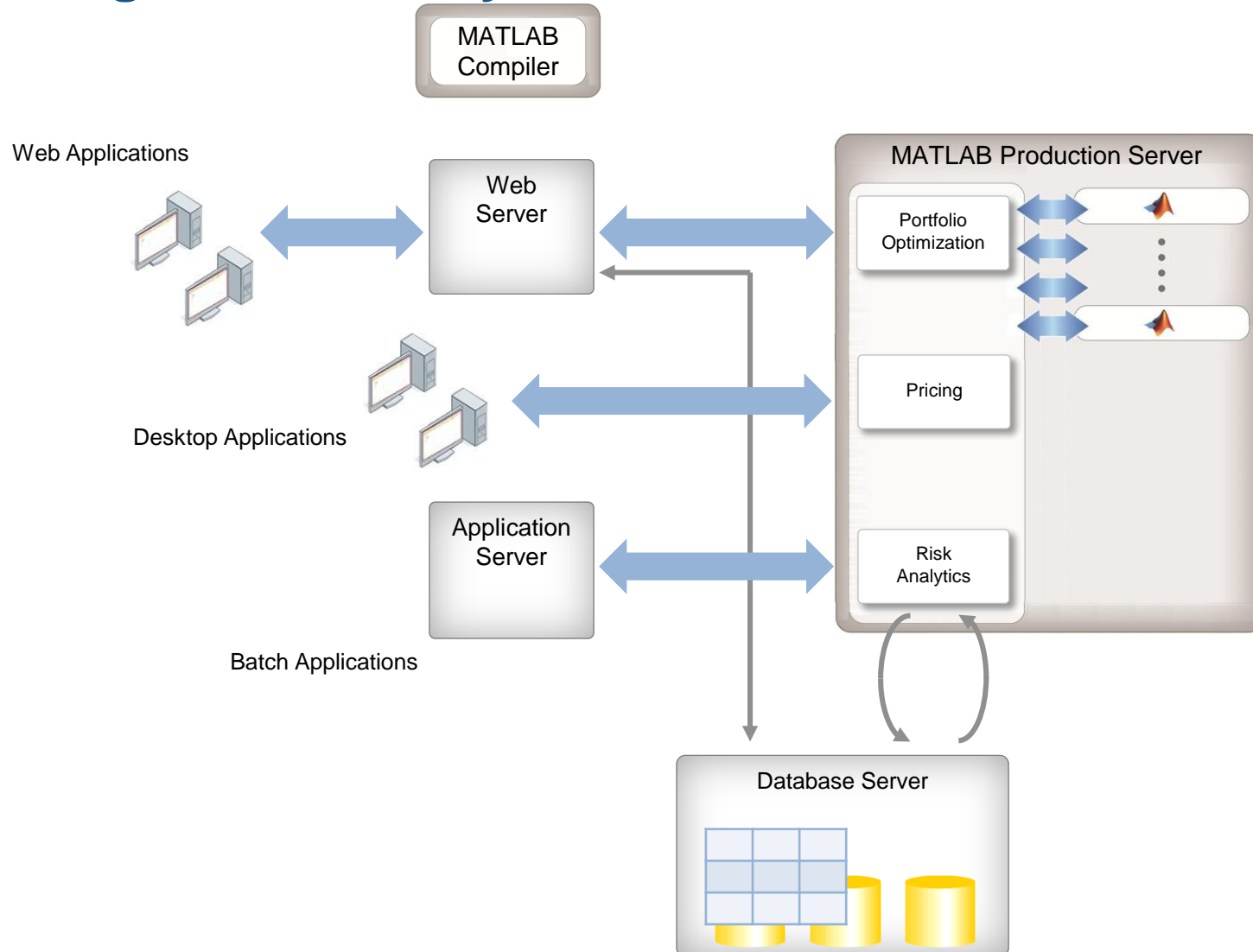
# MATLAB Production Server

## Scale up & centralize analytics

- Most efficient path for enterprise applications
- Deploy MATLAB programs into production
  - Manage multiple MATLAB programs and versions
  - Update programs without server restarts
  - Reliably service large numbers of concurrent requests
- Integrate with web, database, and application servers

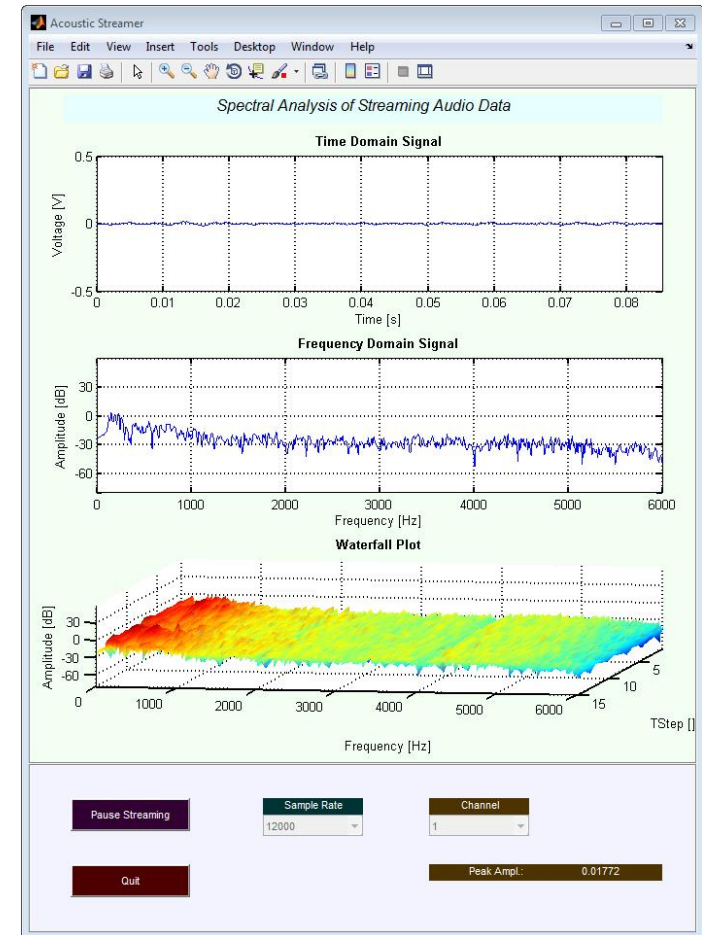


# Centralized Analytics Integrate with IT systems



# MATLAB Application Deployment

- Share MATLAB programs with people who do not have MATLAB
  - Royalty-free distribution
  
- Create both standalone applications and shared libraries
  
- Deploy to desktop, web, and enterprise applications



# Thank you

**Customer stories on following slides**

# UniCredit Bank Austria Develops and Rapidly Deploys a Consistent, Enterprise-Wide Market Data Engine

## Challenge

Improve risk management operations throughout a multinational financial institution

## Solution

Use MATLAB, MATLAB Compiler, and MATLAB Builder JA to build and rapidly deploy a consistent enterprise-wide data warehouse into J2EE Web Architecture

## Results

- Development time reduced by 50%
- Risk management improved across the bank
- Operational, audit, and maintenance costs reduced



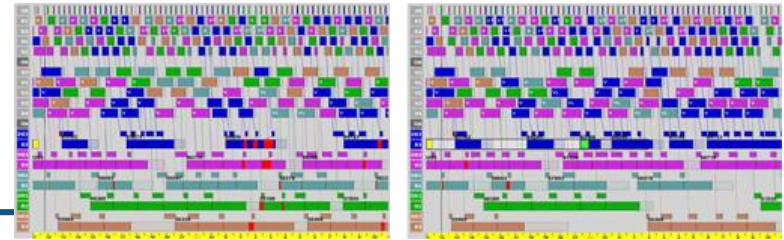
Zero-coupon yield curve plot in UniCredit Bank Austria's UMD environment.

**“ With MATLAB, we can focus on business logic instead of implementation details. We can deploy an algorithm in a Java environment the same day, without any additional coding. This approach enabled us to cut our development time in half, if not more weeks, instead of months.”**

**Peter W. Schweighofer**  
UniCredit Bank Austria



# HKM Optimizes Just-in-Time Steel Manufacturing Schedule



Manually reviewed plant schedule (left) and plant schedule automatically optimized with MATLAB genetic algorithms (right). The optimized schedule minimizes schedule conflicts (in red), meets delivery dates, and achieves the target utilization rate.

## Challenge

Optimize a steel production process to enable consistent, just-in-time delivery

## Solution

Use MATLAB, global optimization, and parallel computing to maximize throughput of more than 5 million tonnes of steel annually

## Results

- Algorithm development accelerated by a factor of 10
- Optimization time cut from 1 hour to 5 minutes
- Customer satisfaction increased

**“C++, Java, or third-party optimization solutions would have required us to spend significantly more time in development or to simplify our constraints. Only MATLAB provided the flexibility, scalability, development speed, and level of optimization that we required.”**

**Alexey Nagaytsev  
Hüttenwerke Krupp Mannesmann**

# Halliburton Makes Oil Exploration Safer Using MATLAB and Neural Network Toolbox



## Challenge

To improve the ability to detect detonation of explosives used to perforate the well bore

## Solution

Use MathWorks products to develop an adaptive, predictive neural network filter that cleanses the detonation signal of contaminating noise from onsite machinery

## Results

- Authentic simulation on the desktop
- An accurate, production-standard algorithm
- Dramatic time savings

**“Using MATLAB and MATLAB Compiler, I can develop an application at least 100 times faster than I could with Visual Basic or C. The time we saved on the very first application that we wrote in MATLAB more than paid for the software.”**

**Roger Schultz  
Halliburton Energy Services**