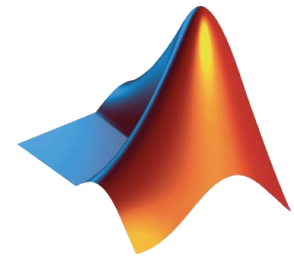


How to build a website powered by MATLAB



David Willingham
Senior Application Engineer – Data Analytics

Case Study – Mining Economics

forecasting the profitability & financial risk of a mine

Home Page - My ASP.NET x
localhost:62339

Register Login
Home About Contact

Mining Economic Forecast - Using MATLAB Production Server

This website helps users forecast the profitability of the lifecycle of an Iron Ore Mine. It has the ability to do forecast 2 main drivers of profit:

- Forecast Iron Ore prices, based off 2 types of models GBM (geometric brown motion) or HWW (mean reversion)
- Forecast Net Present Value cashflow and risk distributions

To run:

- Forecast Iron Ore Prices - Enter Values for Model Type, Start Date for the model, number of Monte Carlo Simulations, then "Simulate Iron Ore Price".
- Forecast NPV - Enter Values for Model Type, Start Date for the model, number of Monte Carlo Simulations, then "NPV Analysis"

Extra Inputs - You do have the ability to modify any of the other inputs for production, grade, recovery, costs & CAPEX

Production Period	Year0	Year1	Year2	Year3	Year4	Year5	Year6	Year7	Year8	
Ore Production (000's)		16000	16000	16000	16000	16000	16000	16000	16000	
Average Grade %		0.65	0.65	0.65	0.65	0.6	0.6	0.6	0.6	
Recovery		0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	
Costs		-291421	-445751	-698516	-348929	-320967	-384916	-620860	-65630	
Iron Price Avg (\$US/ton)		59.2340	61.4953	63.8762	66.2542	68.6970	71.1956	73.8387	76.5685	
Sales \$		597552	620364	644383	668373	639707	662974	687586	713006	
CapEx		-900000								

Model Type: gbm Start Date: 2008-04-01 Num of Sims: 100000

Plot Historical Simulate Iron Ore Price

Historical & Simulated Iron Ore Prices GBM

Num of Sims = 100000
Simulation Time = 2.1101(s)

Home Page - My ASP.NET x
localhost:62339

NPV Analysis

Net Present Value Forecast

NPV Distribution

This website uses the following:

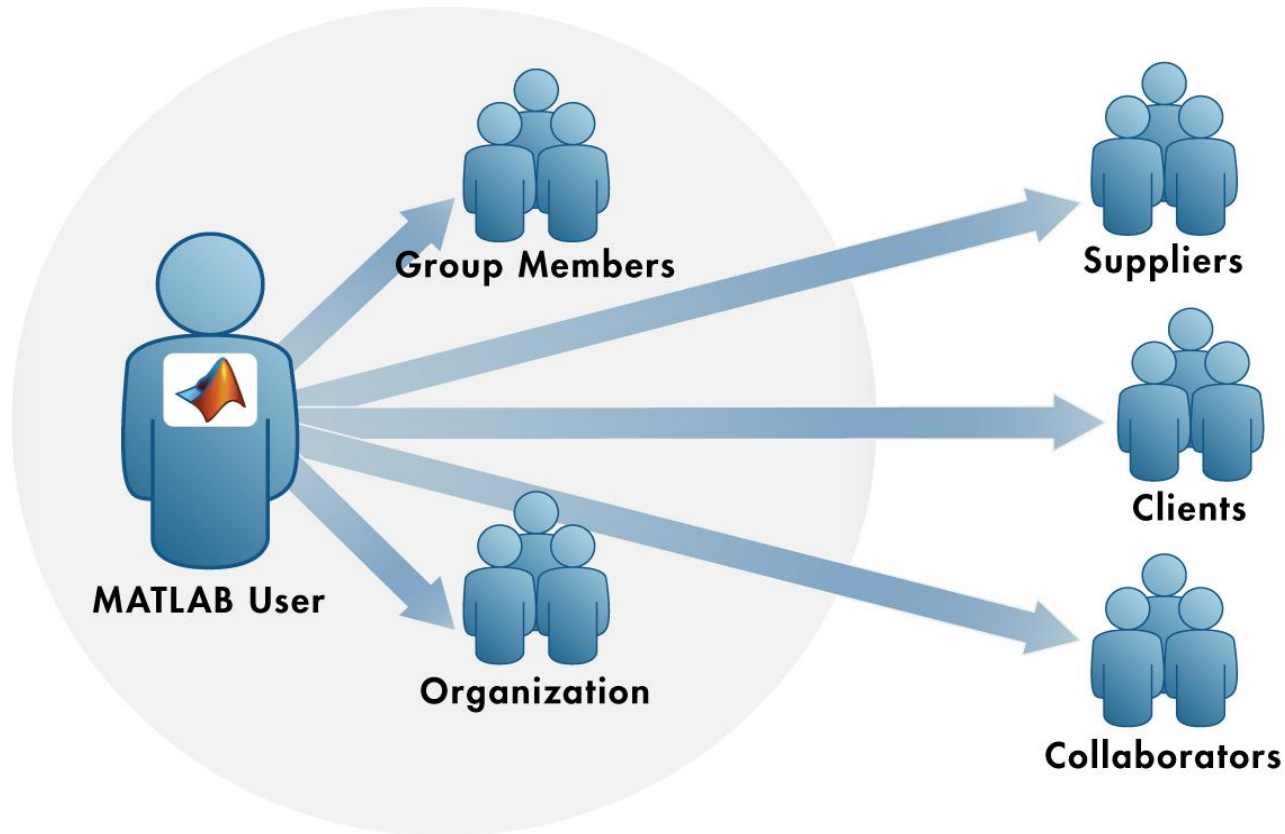
- MATLAB**
MATLAB was the data analytics environment used to develop and test the models prior to website deployment.
- MATLAB Compiler SDK**
MATLAB Compiler SDK was used for compiling the MATLAB models for deployment to this website.
- MATLAB Production Server**
MATLAB Production Server was used as the Production Environment for hosting the deployed models, which are then called in .NET

More Information:
For more information on how to deploy MATLAB models to the Web using the Compiler SDK & Production Server please refer to:
Webinar: [View Here](#)
Whitepaper: [Incorporate MATLAB Analytics into Your Business Systems](#): [Download Here](#)

Case Study - Background

- **Business Issues**
 - An analyst has developed the forecasting models
 - Only he/she can run it
 - Turn around for simulating scenarios is slow
 - IP is lost if the analyst leaves
 - Concern about manual re-coding for deployment
 - Needs to be quick
- **Business Solution**
 - Automatically deploy the models to the web with MATLAB Compiler products
 - Other users can now interact and run various scenarios
 - From any laptop/PC or mobile device, instantly
 - IP will exist long after the analyst has developed it
 - No recoding of models necessary

Share Programs Outside of MATLAB



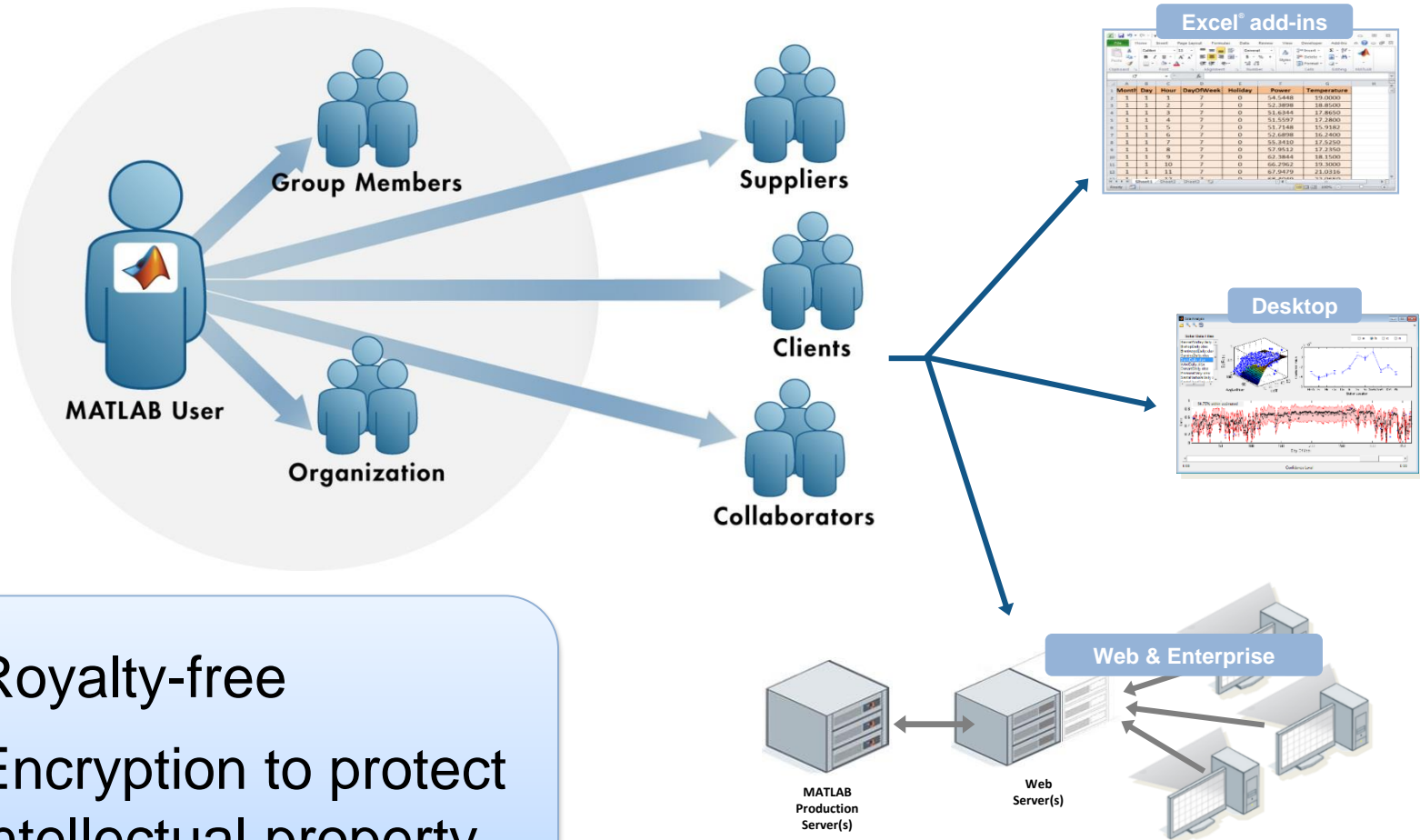
Deploy your MATLAB code to people who don't need MATLAB

Benefits of Deploying MATLAB Code

- Domain experts maintain ownership of ideas, algorithms, and applications
- Flexibility to integrate with different programming languages
- Implement a common algorithm on different platforms
- Avoid time consuming and error prone re-coding
- Easily adopt algorithm improvements throughout lifecycle

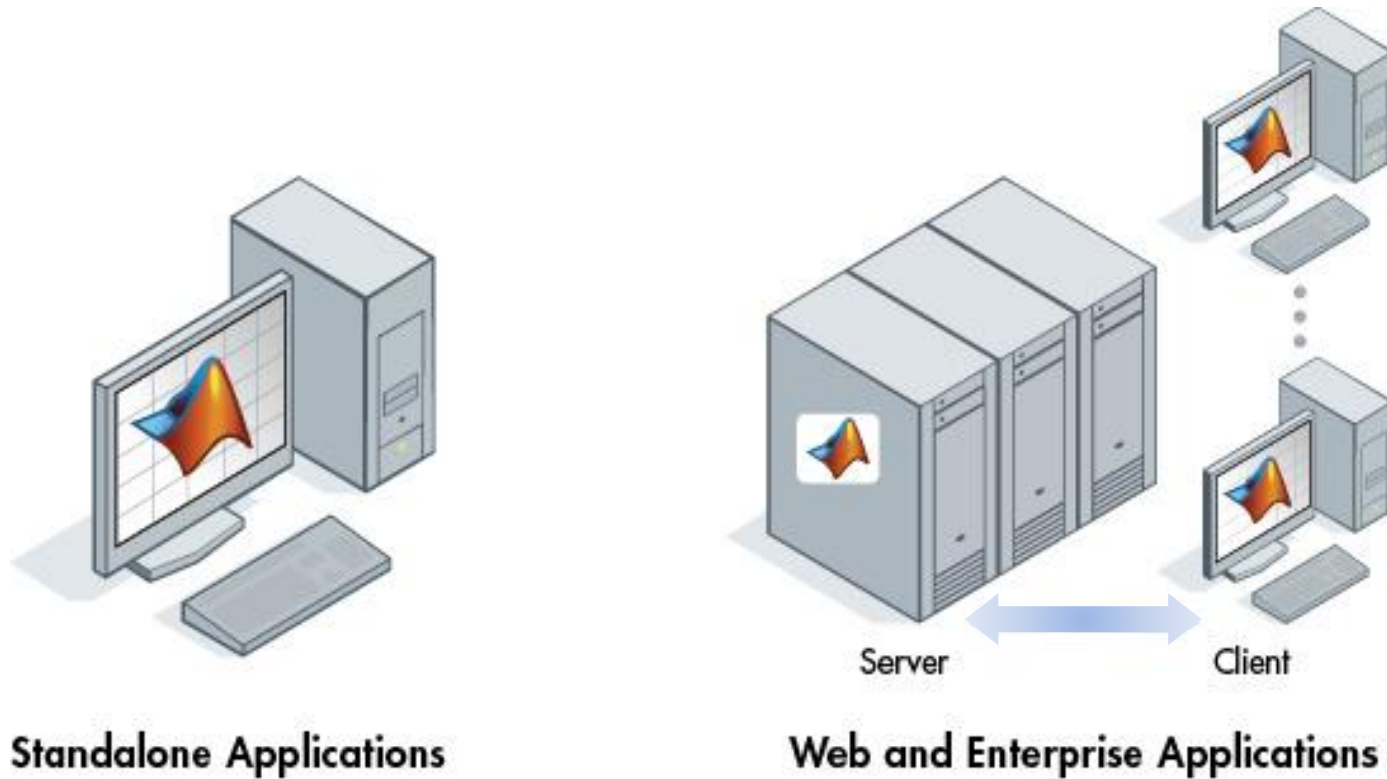


A Primer on Sharing MATLAB Programs



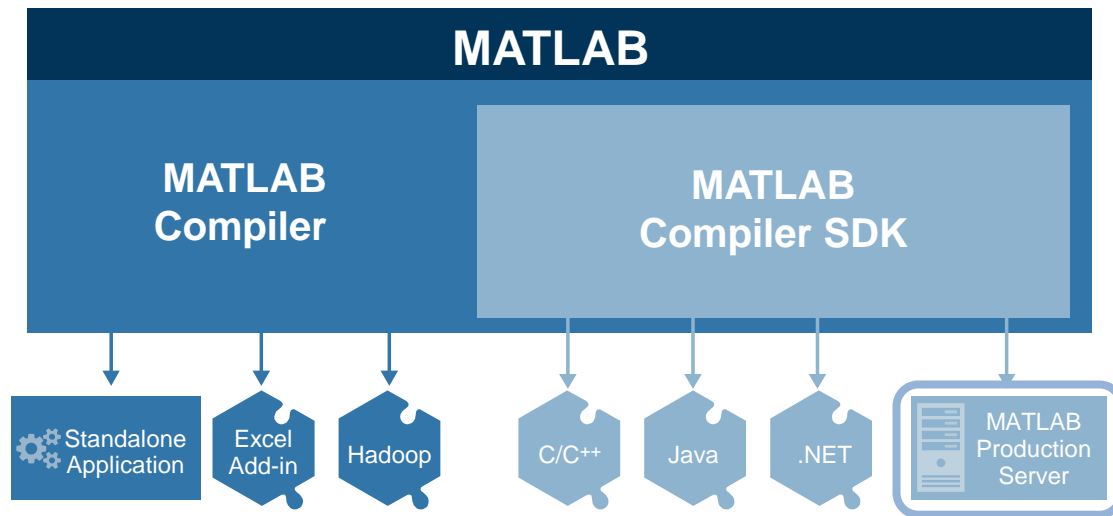
- Royalty-free
- Encryption to protect intellectual property

The Range of Application Platforms



Scale of Distribution
Application Complexity
Enterprise Integration

Which Product will Fit Your Needs?



MATLAB Compiler for sharing MATLAB programs without integration programming

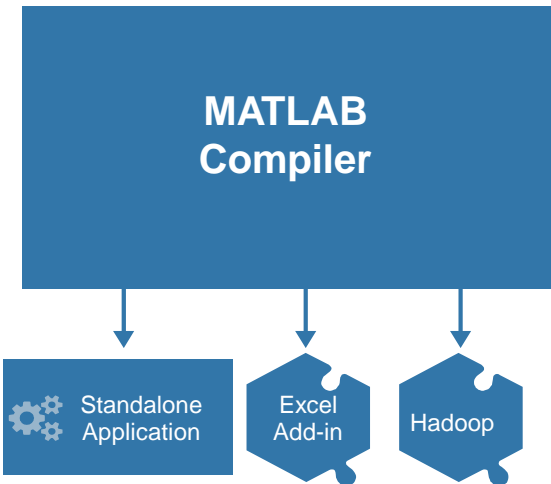
MATLAB Compiler SDK provides implementation and platform flexibility for software developers

MATLAB Production Server provides the most efficient development path for secure and scalable web and enterprise applications

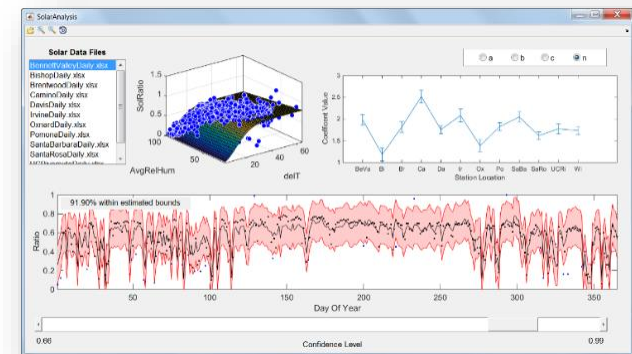
Using MATLAB Compiler

Compiled applications can be shared as:

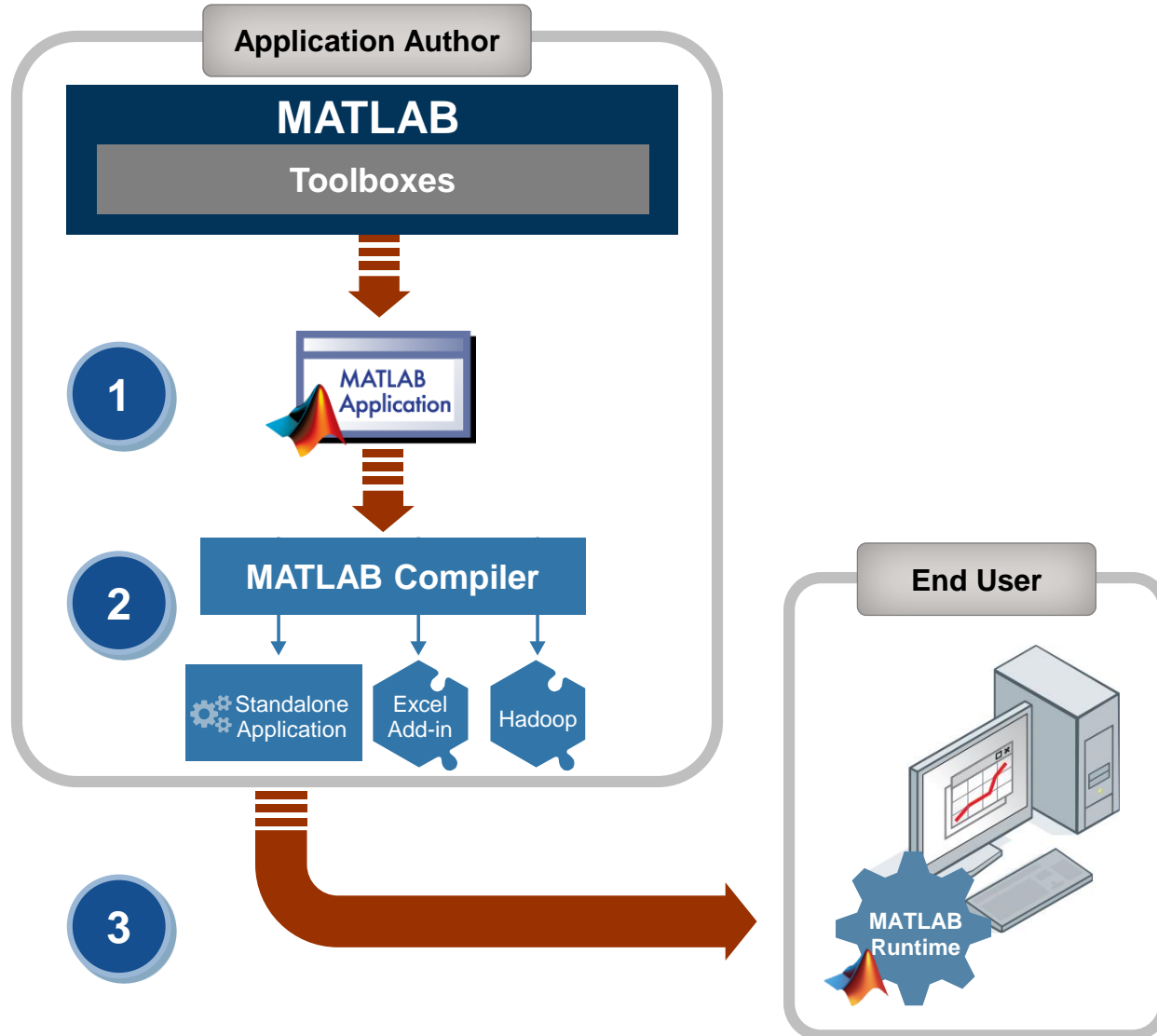
- Standalone desktop applications
- Add-ins for integration with Microsoft Excel[®] spreadsheets
- Components that run MATLAB code against Hadoop



Create professional software with customizable installers, icons, and splash screens ... without integration programming



Sharing Standalone Applications

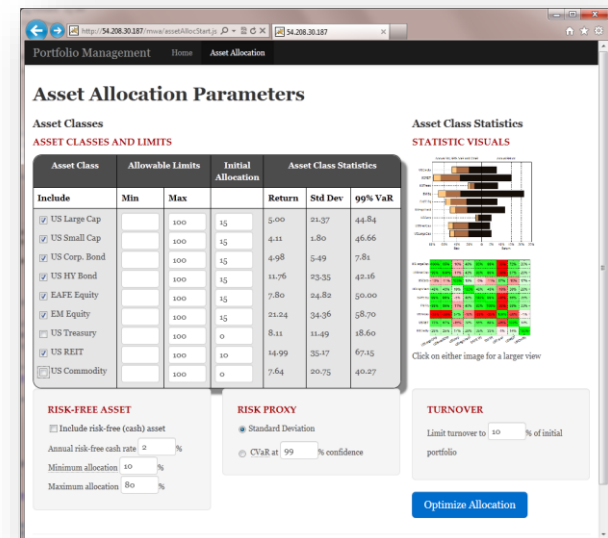
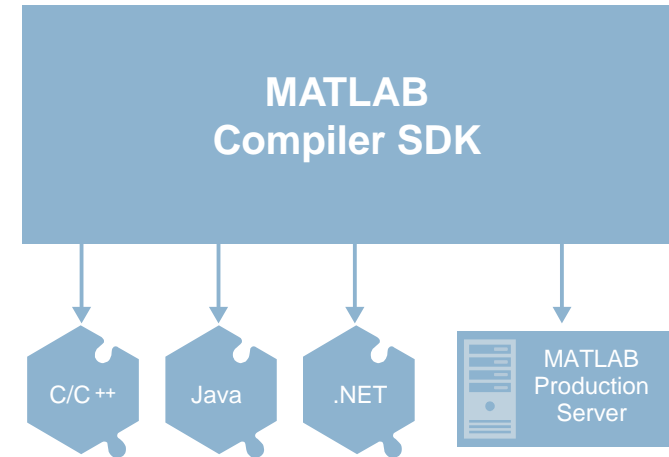


Using MATLAB Compiler SDK

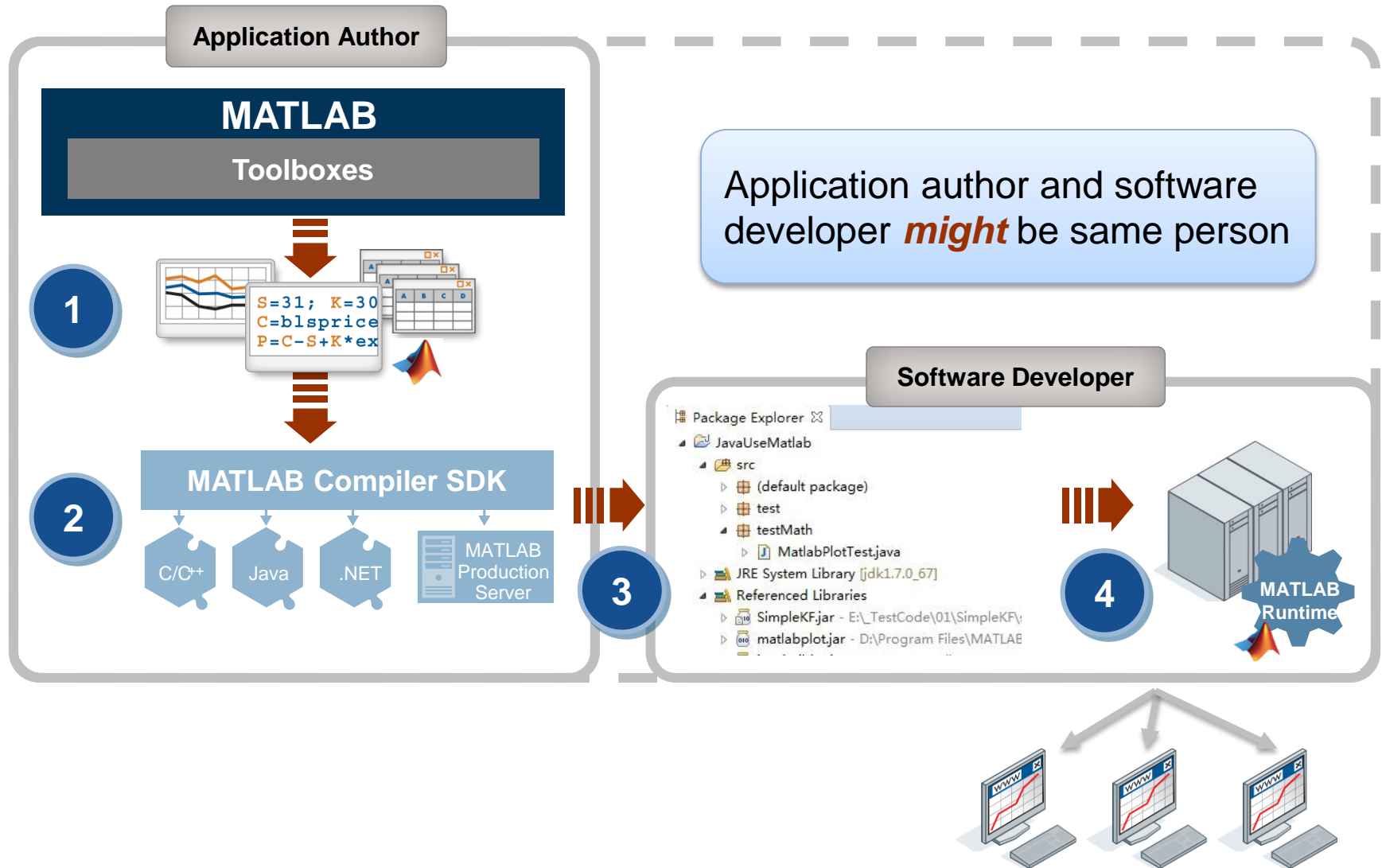
Flexible toolkit for software developers

- Integrate with applications written in C/C++, .NET, Java
- Develop applications for MATLAB Production Server

Develop a custom application server or deploy with MATLAB Production Server



Integrating MATLAB-based Components



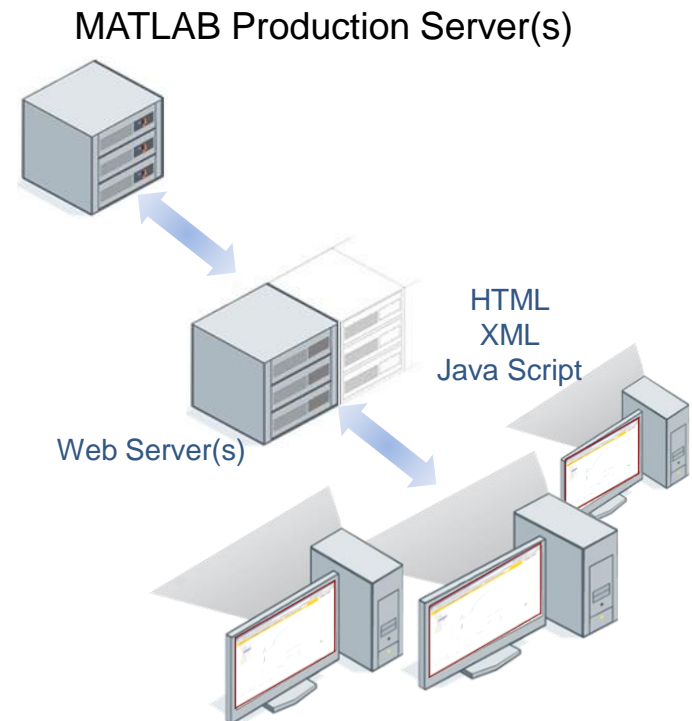
Scale up with MATLAB Production Server

Most efficient path for creating enterprise applications

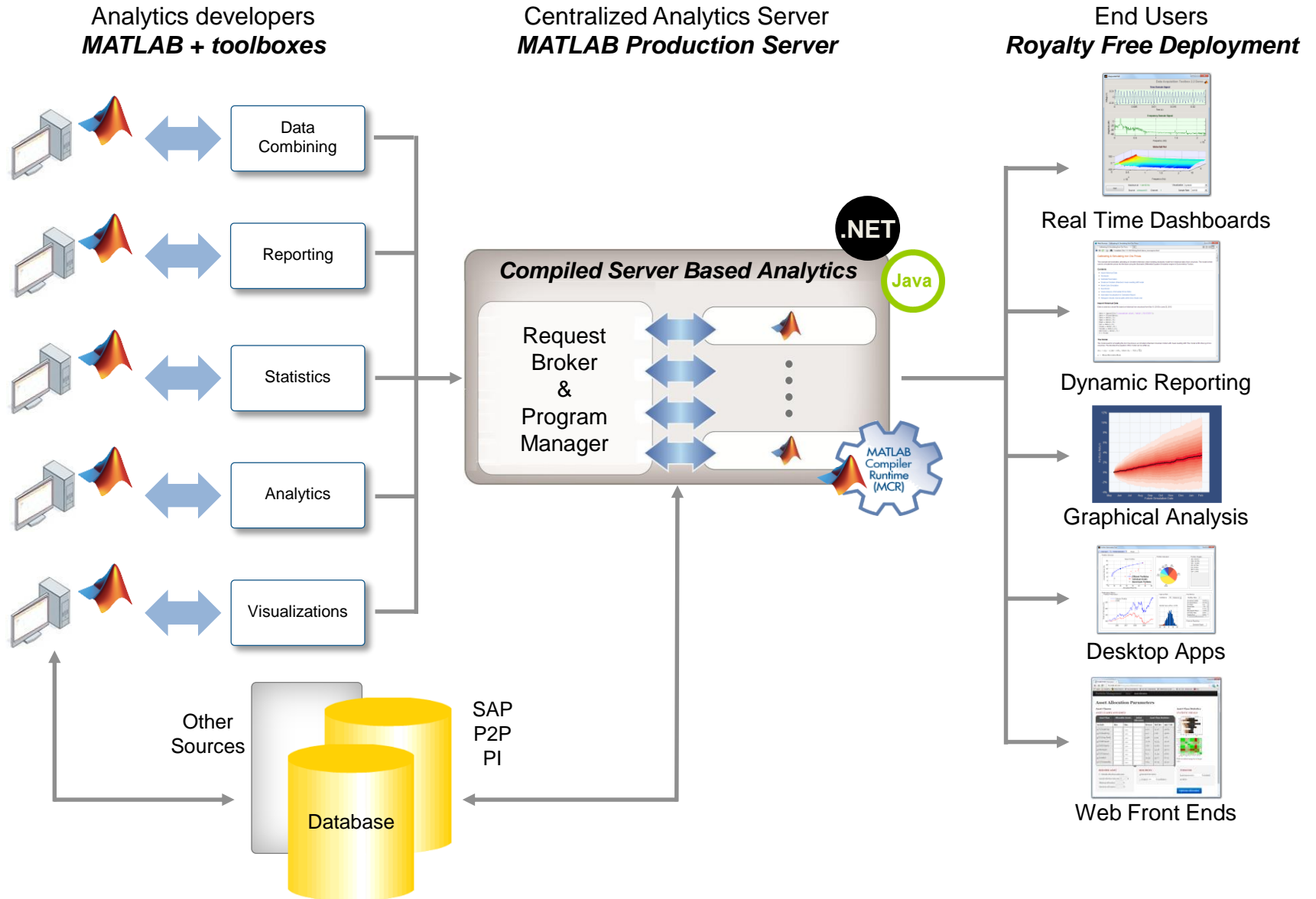
Deploy MATLAB programs into production

- 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



Enterprise Use Case Workflow



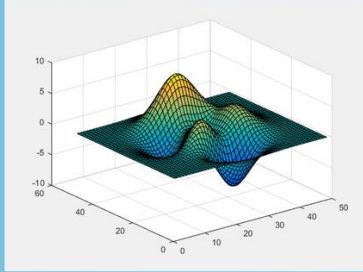
Website deployment – Hello World

- Two Hello World Examples:
 1. Output a matrix
 2. Output a MATLAB figure

MATLAB PRODUCTION SERVER EXAMPLE
 Modify this template to jump-start your ASP.NET application calling MPS.

To learn more about the MATLAB Production Server, go to: ASP.NET, visit www.mathworks.com.au. The page features videos, tutorials, and samples to help you get the most from the MATLAB Production Server.

Calling a MATLAB figure



5

Calling a MATLAB function

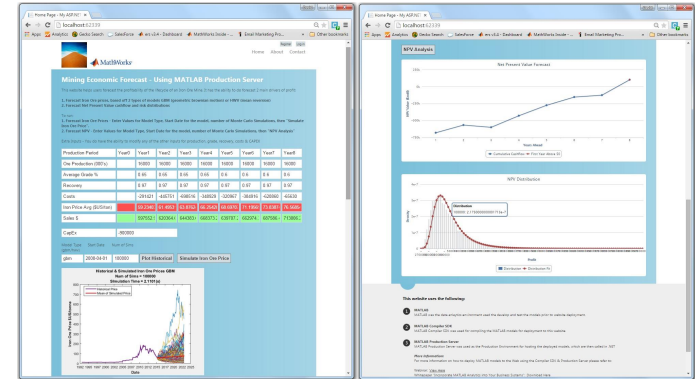
```

17 24 1 8 15
23 5 7 14 16
4 6 13 20 22
10 12 19 21 3
11 13 25 2 9
  
```

Case Study – Mining Economics

- Inputs
 - Table of Mining Production Values
 - Type of model
 - Number of Simulations

- Outputs
 - Monte Carlo Simulation of Iron Ore Price Forecast
 - With MATLAB Figures
 - Net Present Value Cashflow and Risk profile
 - With 3rd party interactive web charts

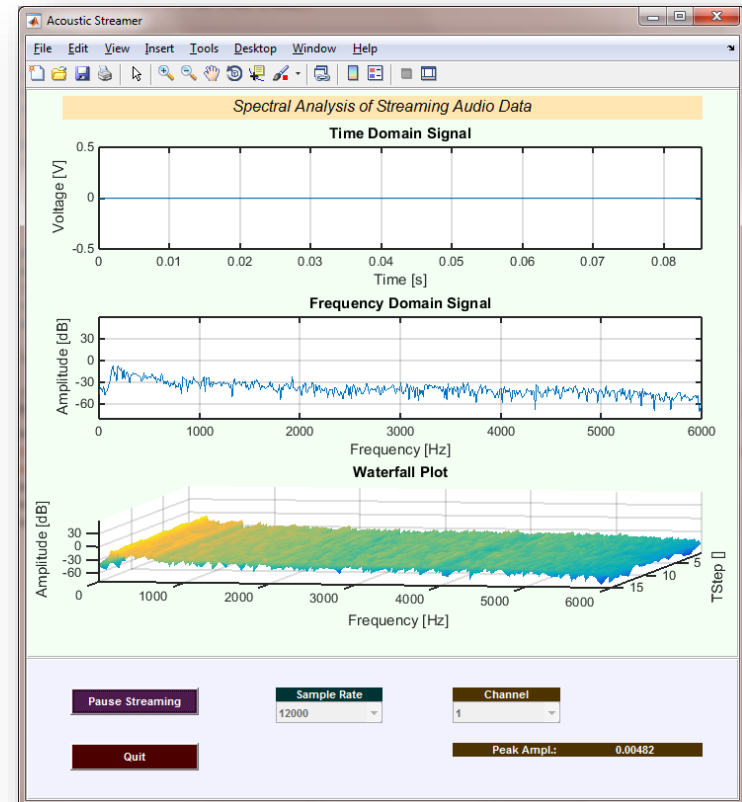


MATLAB Application Deployment

- Share MATLAB programs with people who do not have MATLAB
 - Royalty-free distribution
 - Encryption to protect your IP

- Create both standalone applications and components for integration

- Deploy to desktop, web, and enterprise applications



Learn more ...

Product and Solutions home pages

- <http://www.mathworks.com/products/compiler/>
- <http://www.mathworks.com/products/matlab-compiler-sdk/>
- <http://www.mathworks.com/products/matlab-production-server/>
- <http://www.mathworks.com/solutions/desktop-web-deployment/>

Other useful links

- <http://www.mathworks.com/examples/>
- <http://www.mathworks.com/videos/>
- <http://www.mathworks.com/company/events/webinars/index.html>
- <http://www.mathworks.com/company/newsletters/technicalarticles.html>

Thank you

Customer stories and supplemental information on following slides

Customer Stories

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 Compiler SDK 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

A2A Develops Comprehensive Risk Management Solution for Energy Markets

Challenge

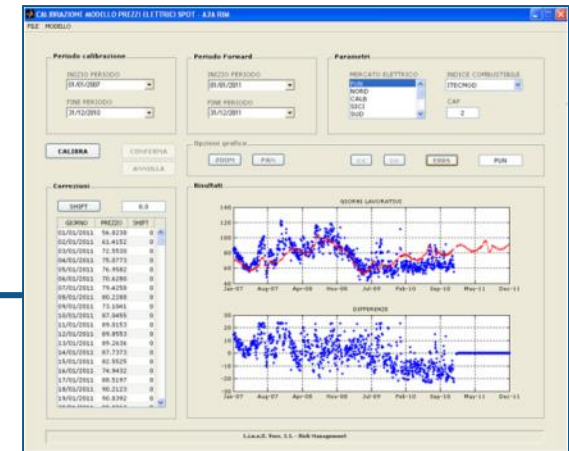
Manage and mitigate risk across markets in a large utility company

Solution

Use MATLAB and companion toolboxes to process data, develop risk and pricing models, and deploy an interactive dashboard for analysts

Results

- Hour-long calculations completed in 30 seconds
- Development time halved
- Pricing model development accelerated



A2A's application for calibrating and forecasting electricity spot price, a component of the Risk Management Dashboard.

“When you deal with numbers all day and work with sophisticated analytical models, having an integrated environment is invaluable. With MATLAB we visualize data, conduct back-testing, and plot graphs to see the results of changes we make, all in one environment, and that saves time.”

Simone Visonà
A2A

Commerzbank Develops Production Software System for Calculating Derived Market Data



Commerzbank headquarters in Frankfurt.

Challenge

Compute a variety of derived market data from raw market data

Solution

Use MATLAB to read data from a data management system in a Windows and Linux architecture, perform analyses and optimizations, visualize results, and deploy mission-critical calculations

Results

- Integration with existing system simplified
- Implementation time reduced by months
- Updates made in days, not weeks

“Our solution required a Windows client and Linux server software. We used MATLAB to rapidly develop both by taking advantage of distributed computing, a MEX-file interface to access our financial data, and fast, built-in functions for optimization, regression, and more.”

**Julian Zenglein
Commerzbank**

Fulcrum Asset Management Develops Custom Quantitative Risk Management System

Challenge

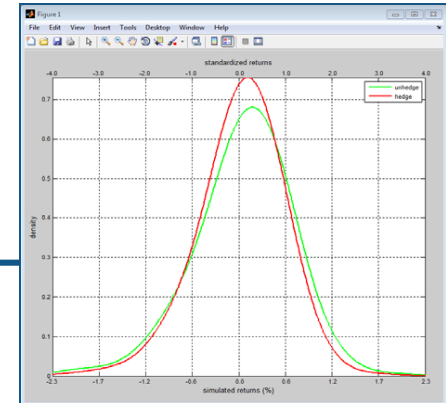
Develop an accurate, scalable, and fast risk engine for fund management

Solution

Use MATLAB to import financial data from multiple sources, develop sophisticated risk models, and run optimizations and scenarios analysis on multicore processors

Results

- Optimizations and calculations accelerated
- Risk measurement accuracy improved
- Integration with databases and datafeeds streamlined



Distribution of standardized and unstandardized simulated portfolio returns before and after hedging.

“With MATLAB we developed a risk management system that helps us implement our strategies, hedge our risks more efficiently, and respond rapidly to changes in the market. MATLAB enables us to incorporate our ongoing research and the experience of our fund managers into the risk engine.”

**Athanasios Bolmatis
Fulcrum**

Microtech Develops and Tests Implantable Blood Pressure Sensor



Microtech's submillimeter sensor, dotting the "I" on a U.S. penny.

Challenge

Create and test an implantable blood pressure sensor

Solution

Use MATLAB to develop algorithms for generating and analyzing ultrasound waves, create standalone analysis software, and control lab equipment for automated testing

Results

- Development time halved
- Hardware updates streamlined
- Productivity increased by 20%

“Using MATLAB to develop both real-time ultrasound signal processing algorithms and automated measurement software eliminates the need for a dedicated C/C++ programmer as well as the risk of introducing bugs when the algorithm is rewritten.”

**Dr. Yonathan Kozlovsky
Microtech**

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**

STIWA Increases Total Production Output of Automation Machinery



STIWA's shopfloor management system, based on MATLAB, AMS ZPoint-CI, and AMS Analysis-CI.

Challenge

Apply sophisticated mathematical methods to optimize automation machinery and increase total production output

Solution

Use AMS ZPoint-CI to collect large production data sets in near real time and use MATLAB to analyze the data and identify optimal trajectories

Results

- Total cycle time reduced by 30%
- Large data sets analyzed in seconds
- Deployment to multiple machines streamlined

“Our shopfloor management system AMS ZPoint-CI collects a huge amount of machine, process, and product data 24 hours a day. By analyzing this data immediately in MATLAB and AMS Analysis-CI we have achieved a tenfold increase in precision, a 30% reduction in total cycle time, and a significant increase in production output.”

**Alexander Meisinger
STIWA**

Ruukki Metals Improves Steel Manufacturing Processes with Standalone and Web-Based MATLAB Applications

Challenge

Enable operators to correctly set up steel coil processing lines in the manufacturing plant.

Solution

Build applications in MATLAB that enables operators to select and apply the proper settings, analyze production metrics from multiple databases, track individual coils, and refine the process

Results

- Off-gauge coil length reduced from several meters to less than 50 centimeters
- Process efficiency increased
- Waste reduced



Ruukki's web-based MATLAB application

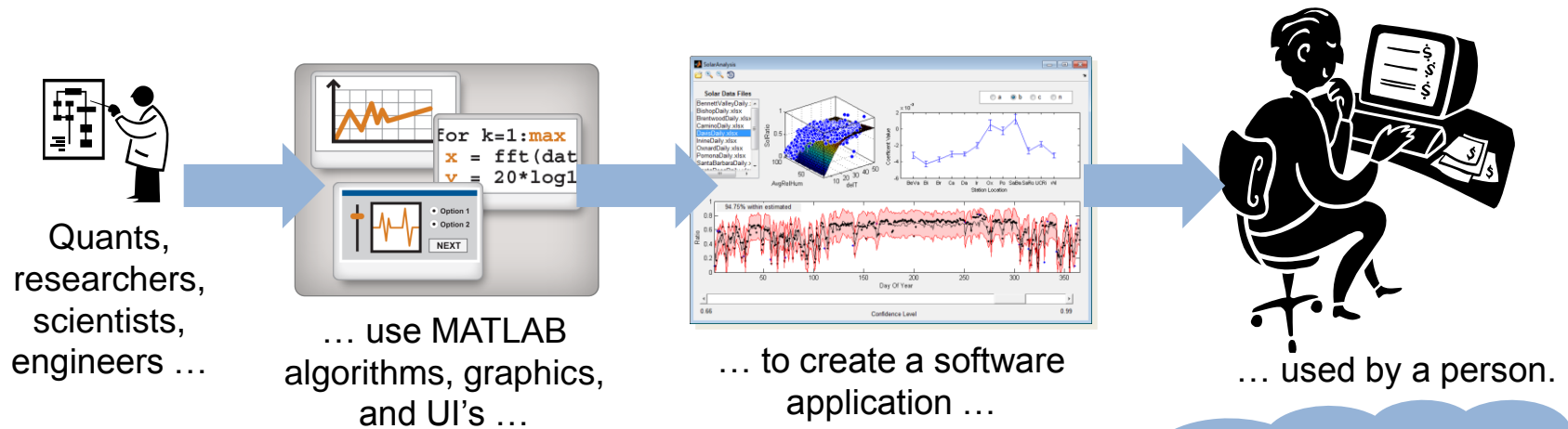
“Since deploying this application and the standalone executable that we created with MATLAB and MATLAB Compiler, we have seen fewer misalignments, less scrap, and significant increases in efficiency and consistency at the plant.”

Mika Judin
Ruukki Metals

Supplemental Material

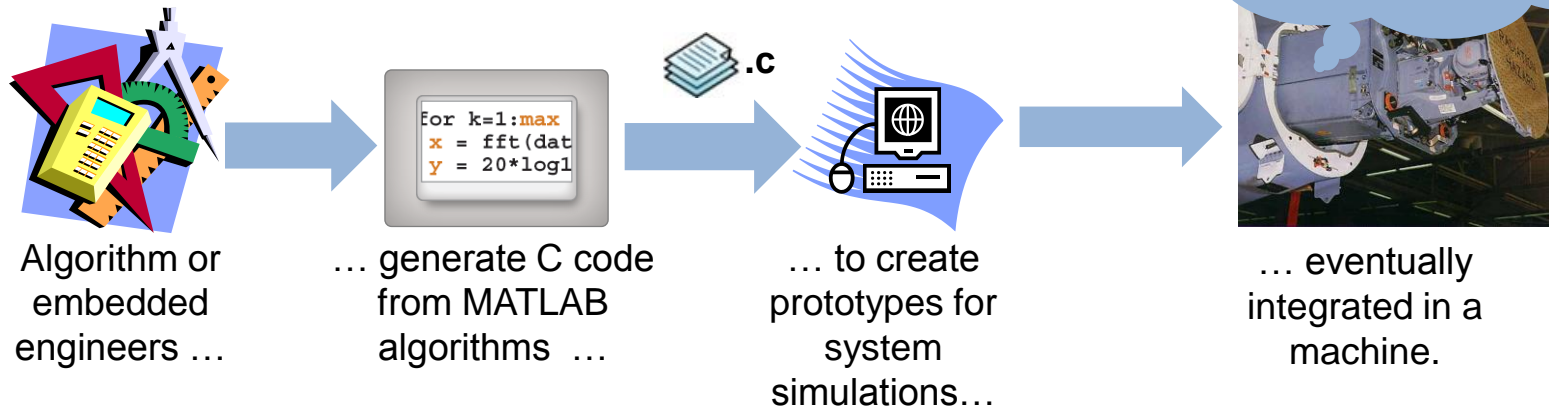
Typical Workflows for Coder and Compiler

Technical Computing: MATLAB Compiler process

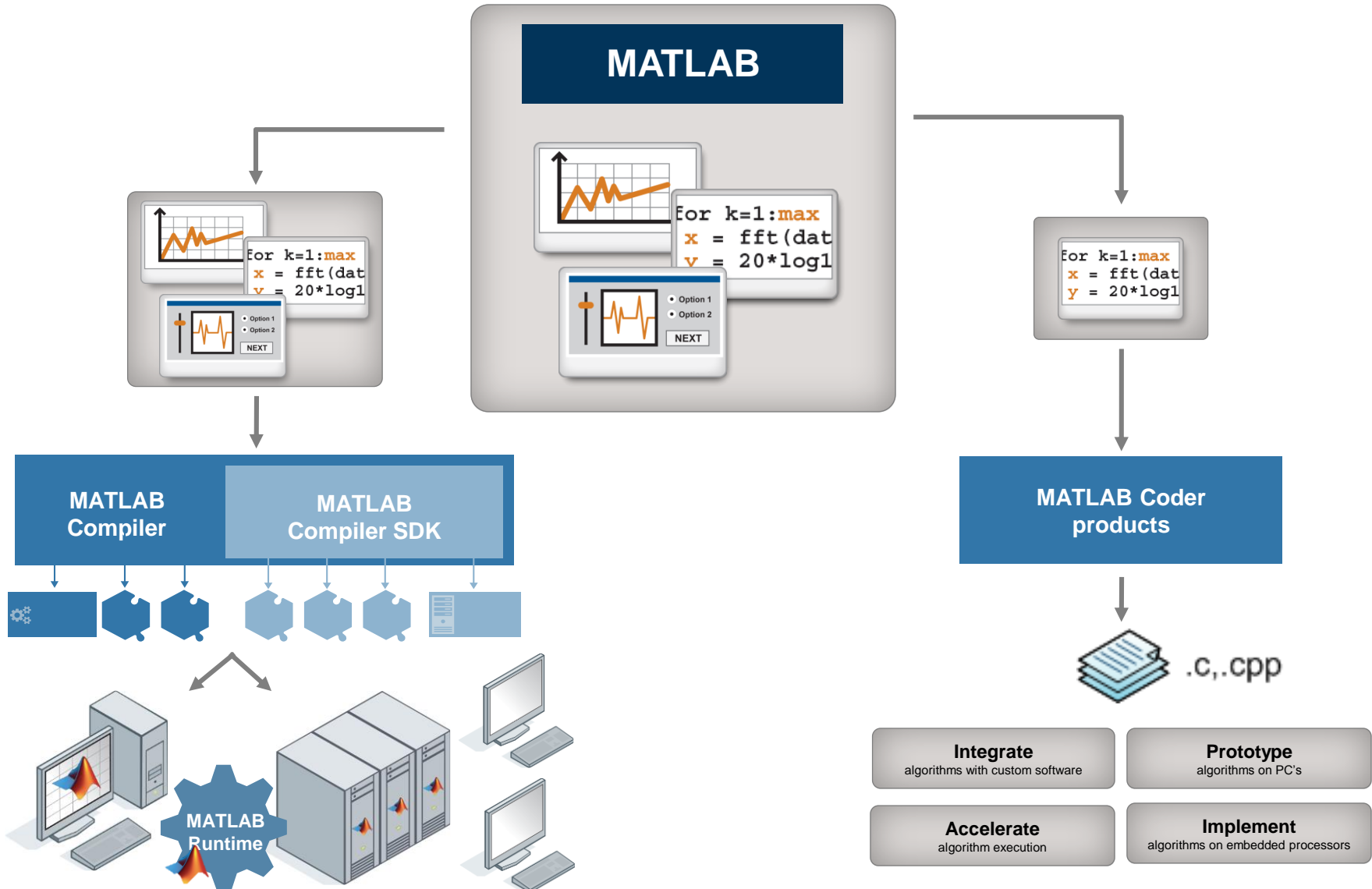


NOTE - Several images come from Google. Need legit source.

Model-based Design: MATLAB Coder process



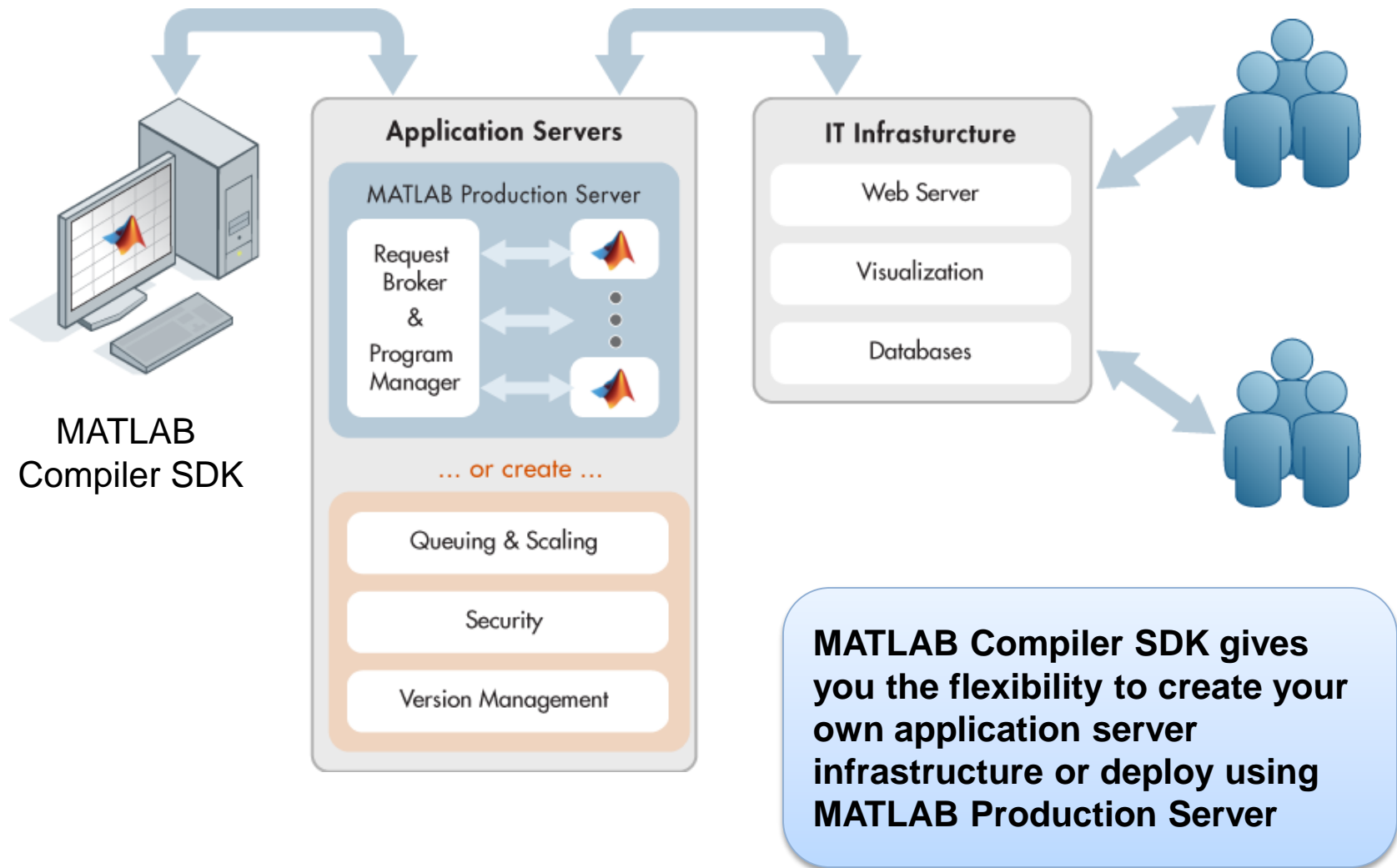
Advising on Deployment or Code Generation



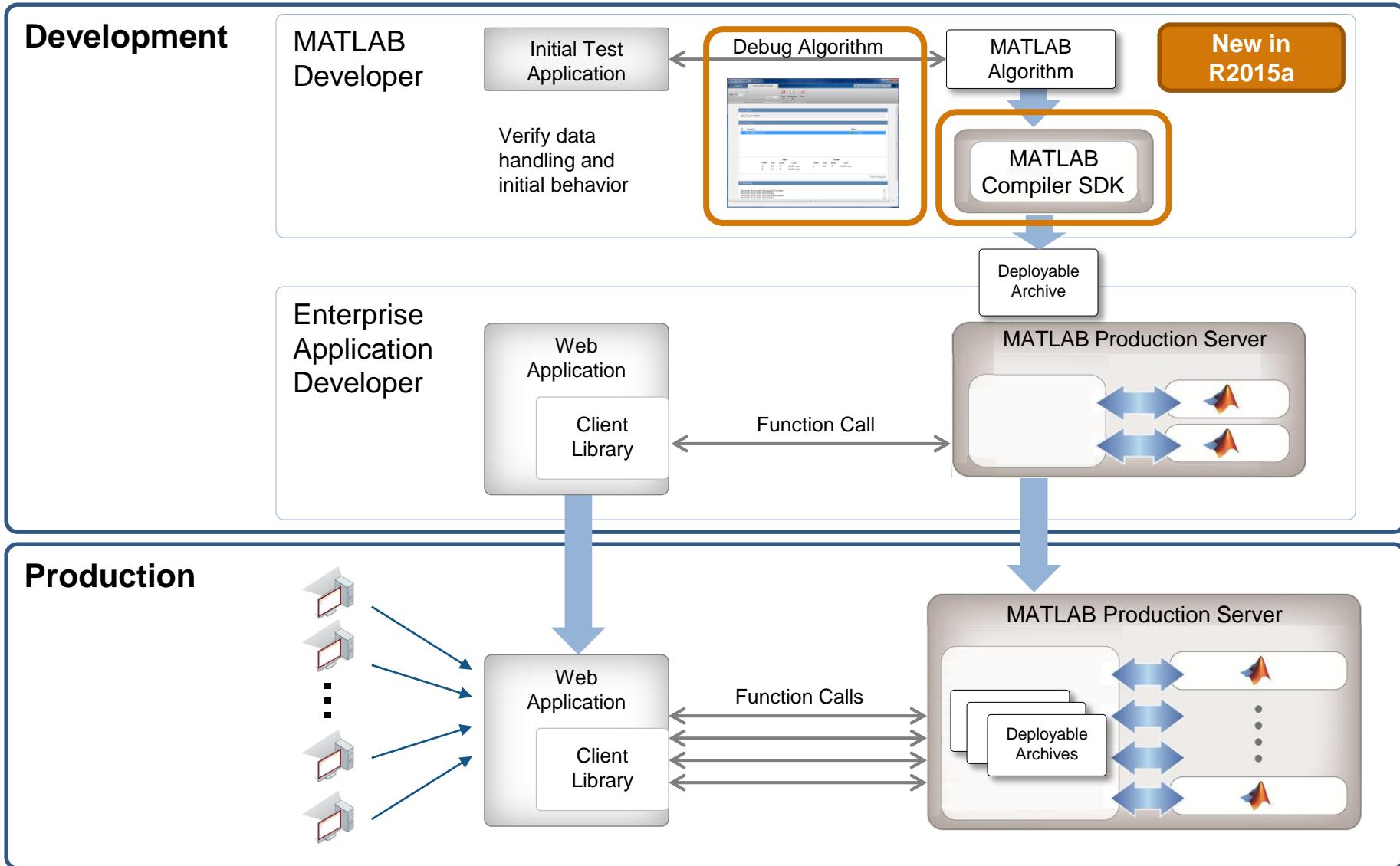
MATLAB Compiler SDK and MATLAB Coder

	MATLAB Compiler MATLAB Compiler SDK	MATLAB Coder
Output	Software components	Portable and readable C source code
MATLAB language support	Full	Subset
Additional libraries	MATLAB Runtime	None
Supported toolboxes	Most toolboxes	Some toolboxes
License model	Royalty-free	Royalty-free
Extensions	MATLAB Production Server	Embedded Coder

Options for Web/Enterprise Application Servers



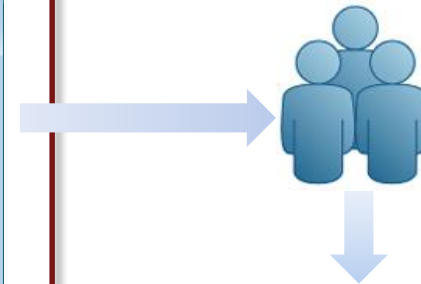
Production Deployment Workflow



Typical Process for Standalone Applications

Application author

The screenshot shows the MATLAB environment with a script editor containing MATLAB code for solar analysis. The code includes variables for temperature difference, solar ratio, and a function to create a solar model. Overlaid on this is the MATLAB Compiler window, which is used to package the application into a standalone executable. The 'Application Information' tab is visible, showing fields for application name, email, and company.

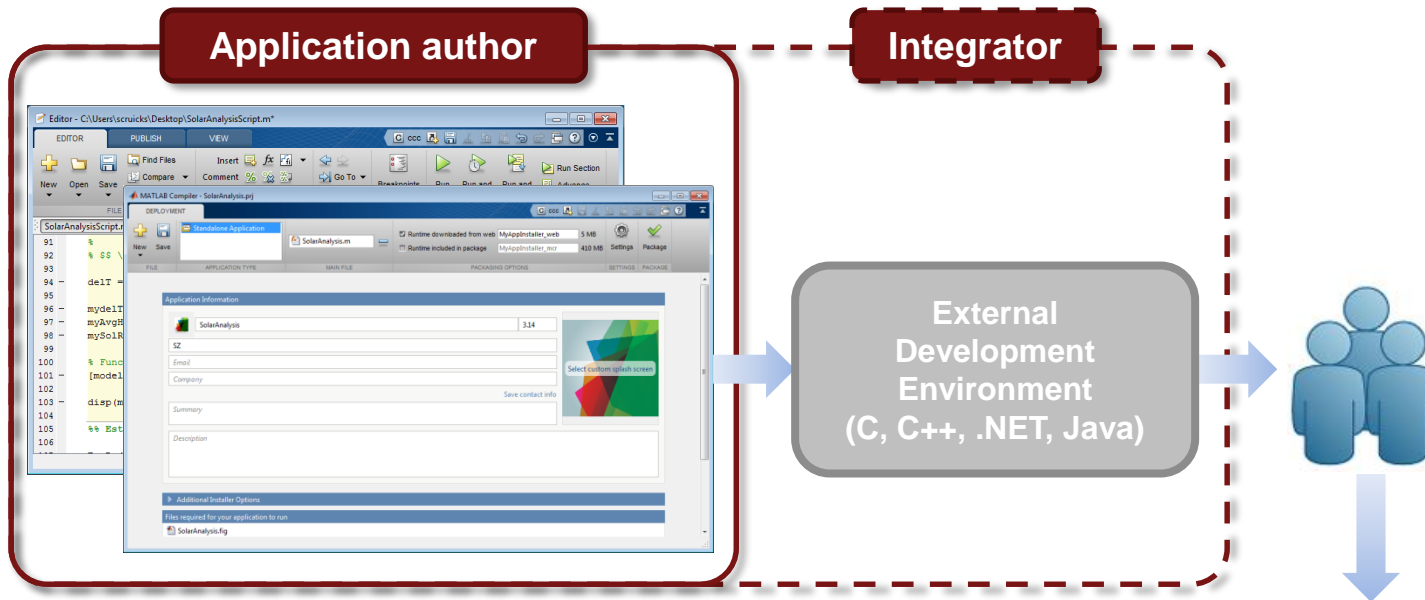


- 1.) Create MATLAB algorithms
- 2.) Define the user interface
- 3.) Package the application using MATLAB Compiler
- 4.) Give the application installer to someone

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

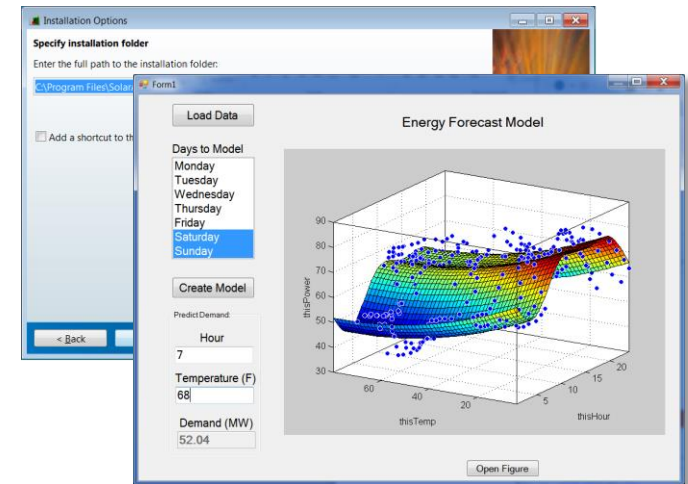
The screenshot shows the 'Installation Options' dialog box for the application. It includes a 'Specify installation folder' field with a 'Browse...' button. Below this is a preview window titled 'Solar Analysis' which displays a 3D surface plot of solar ratio, a line graph of coefficient values for different station locations, and a time-series plot of solar ratio over the day of year with 90.00% confidence bounds.

Typical Process for Integrated Applications

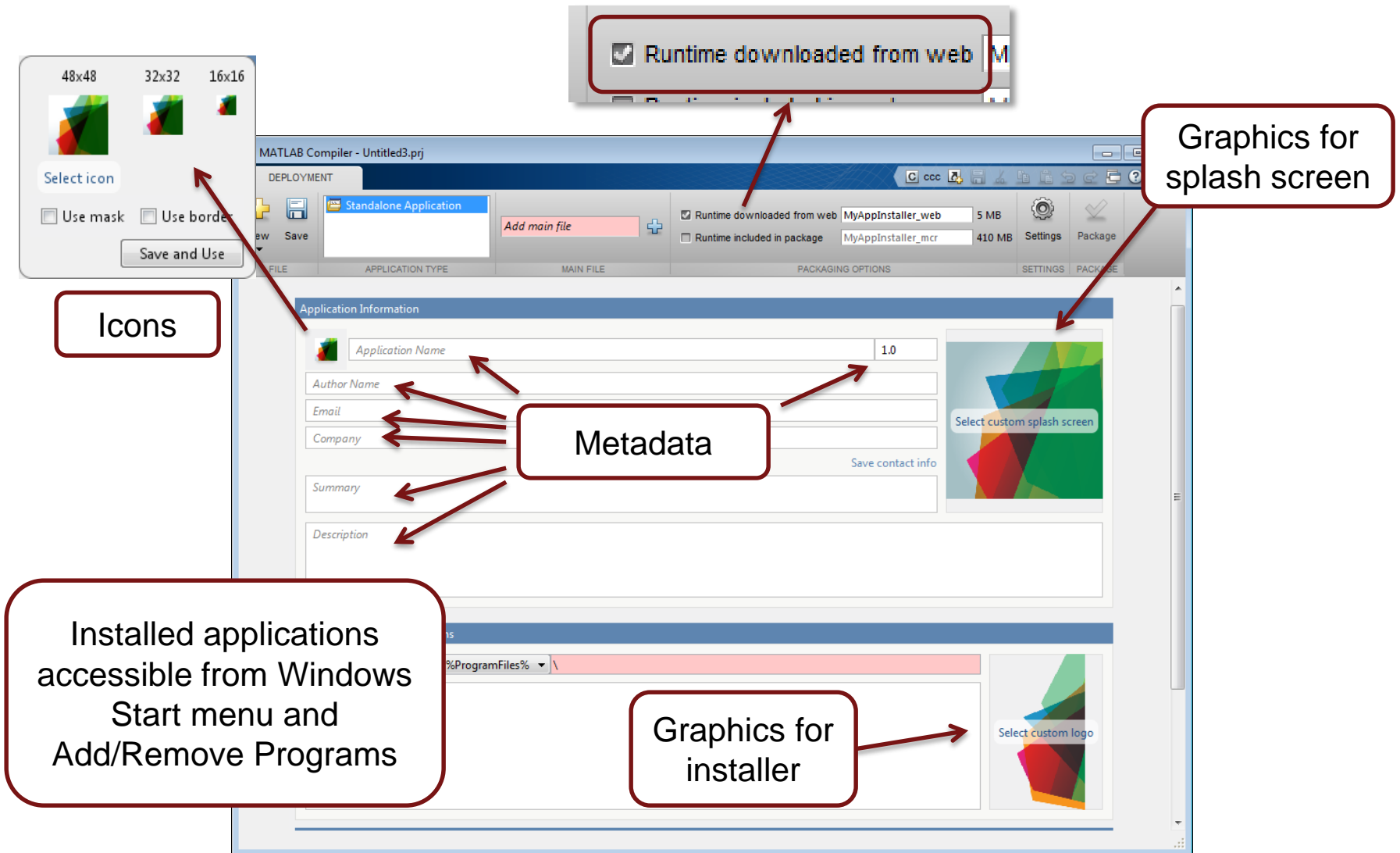


- 1.) Create MATLAB algorithms
- 2.) Package component libraries using MATLAB Compiler SDK
- 3.) Use an external development environment to develop integrated application, including user interface
- 4.) Give the final application to someone

They will install the application on desktops or servers
 ... for their own use or for use by others



Customizations for your Applications



The screenshot shows the MATLAB Compiler interface for configuring a standalone application. Several key areas are highlighted with callouts:

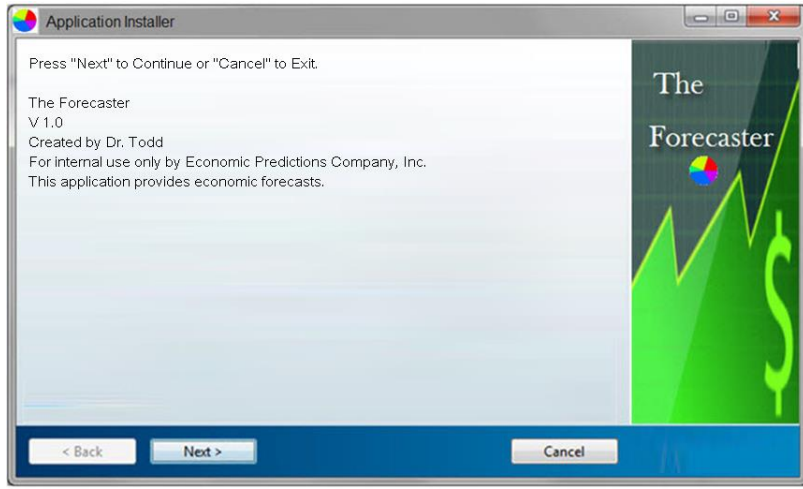
- Icons:** A callout points to the icon selection dialog, which offers sizes of 48x48, 32x32, and 16x16 pixels, and options to use a mask or border.
- Runtime downloaded from web:** A callout points to the checkbox in the packaging options section, which is checked.
- Metadata:** A central callout points to the 'Application Information' section, which includes fields for Application Name, Author Name, Email, Company, Summary, and Description.
- Graphics for splash screen:** A callout points to the 'Select custom splash screen' button, which is accompanied by a colorful abstract graphic.
- Graphics for installer:** A callout points to the 'Select custom logo' button, which is also accompanied by a colorful abstract graphic.

Additional callouts include:

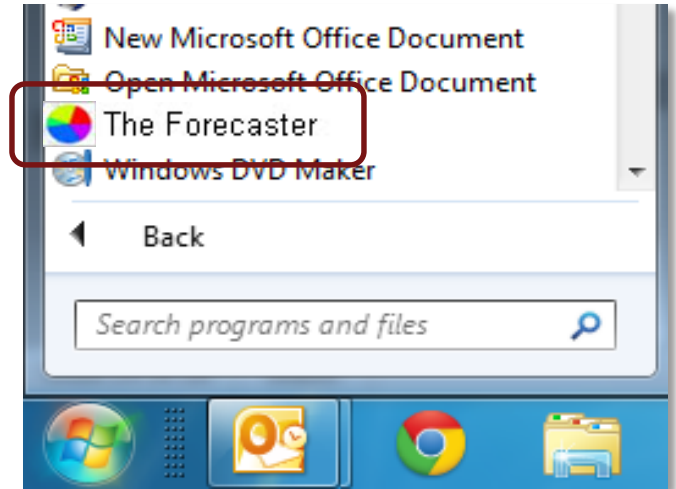
- Installed applications accessible from Windows Start menu and Add/Remove Programs:** A large callout at the bottom left.
- Settings and Package buttons:** Located in the top right of the deployment options section.

End Customer sees a Professional Application

Installer



Application in Start menu



Splash Screen



Icon



MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See 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.
© 2015 The MathWorks, Inc.