

MATLAB as a Financial Engineering Development Platform Delivering Financial / Quantitative Models to the Enterprise

Eugene McGoldrick

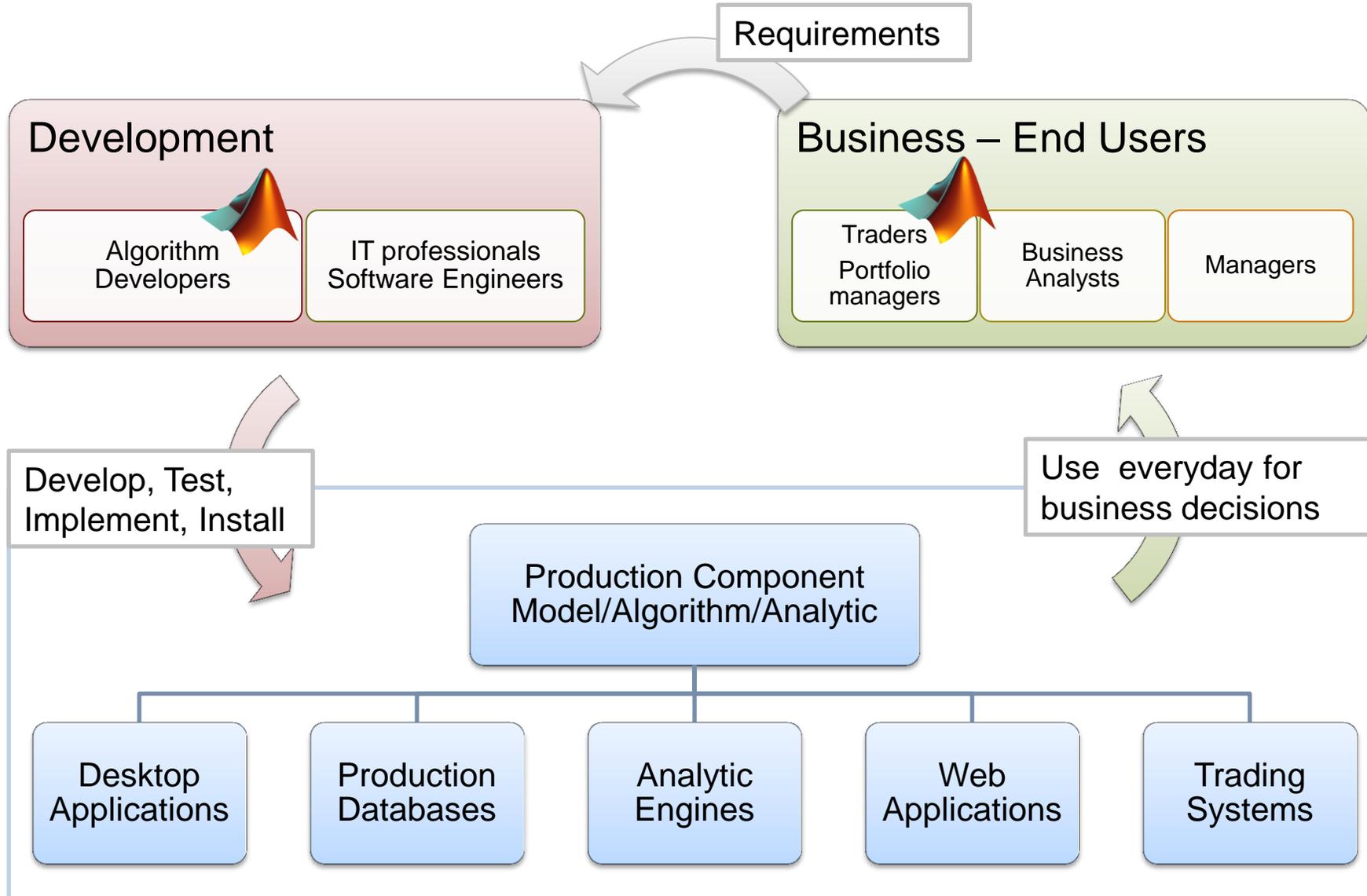
MATLAB – Development Environment for Financial Services

- Development Environment has increasing number of productivity tools built into the base product
 - Test Harness modeled on JUnit, customizable and extensible
 - Support for version control
 - Subversion
 - GIT
 - API available to support other version control systems
 - GUI development tools
 - Guide
 - App Designer
 - Performance analysis tools
 - Profiler
 - Code Analyzer

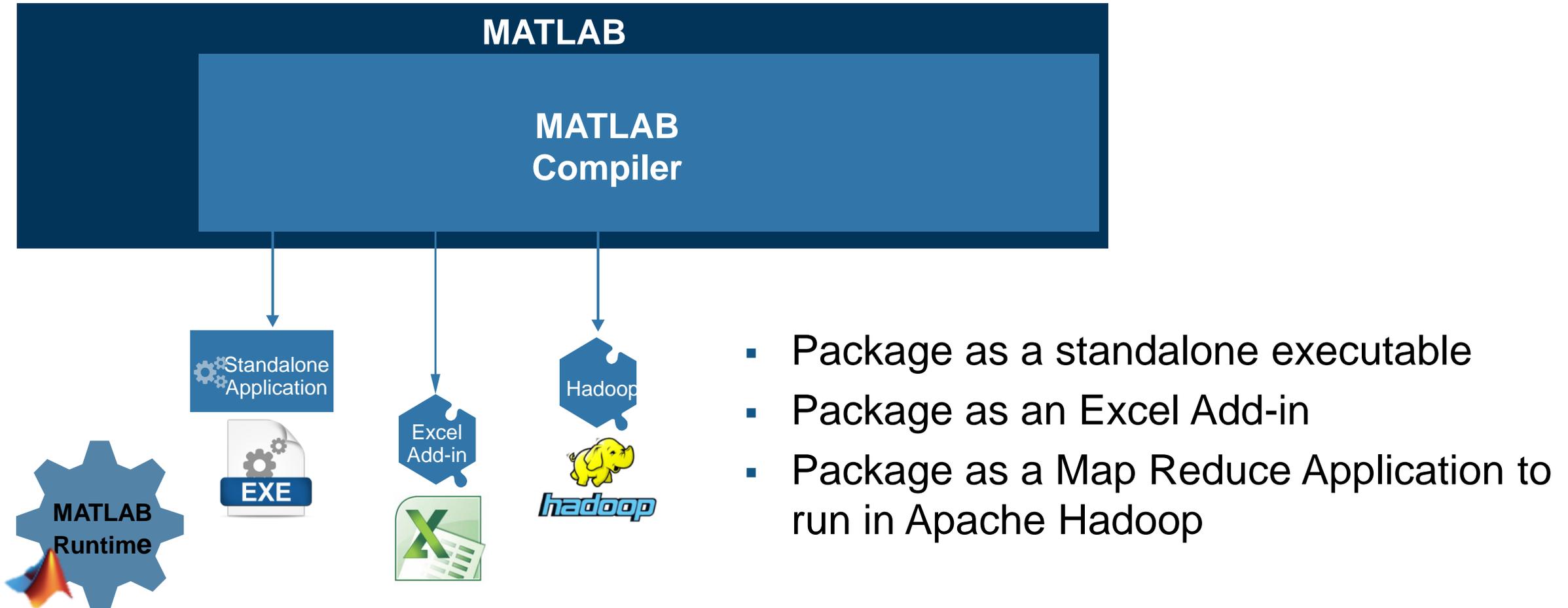
MATLAB in the Enterprise

- Goals:
 - Enable customers to rapidly develop and deploy MATLAB applications onto the desktop.
 - Seamlessly integrate MATLAB generated components into other languages, applications and enterprise production systems.

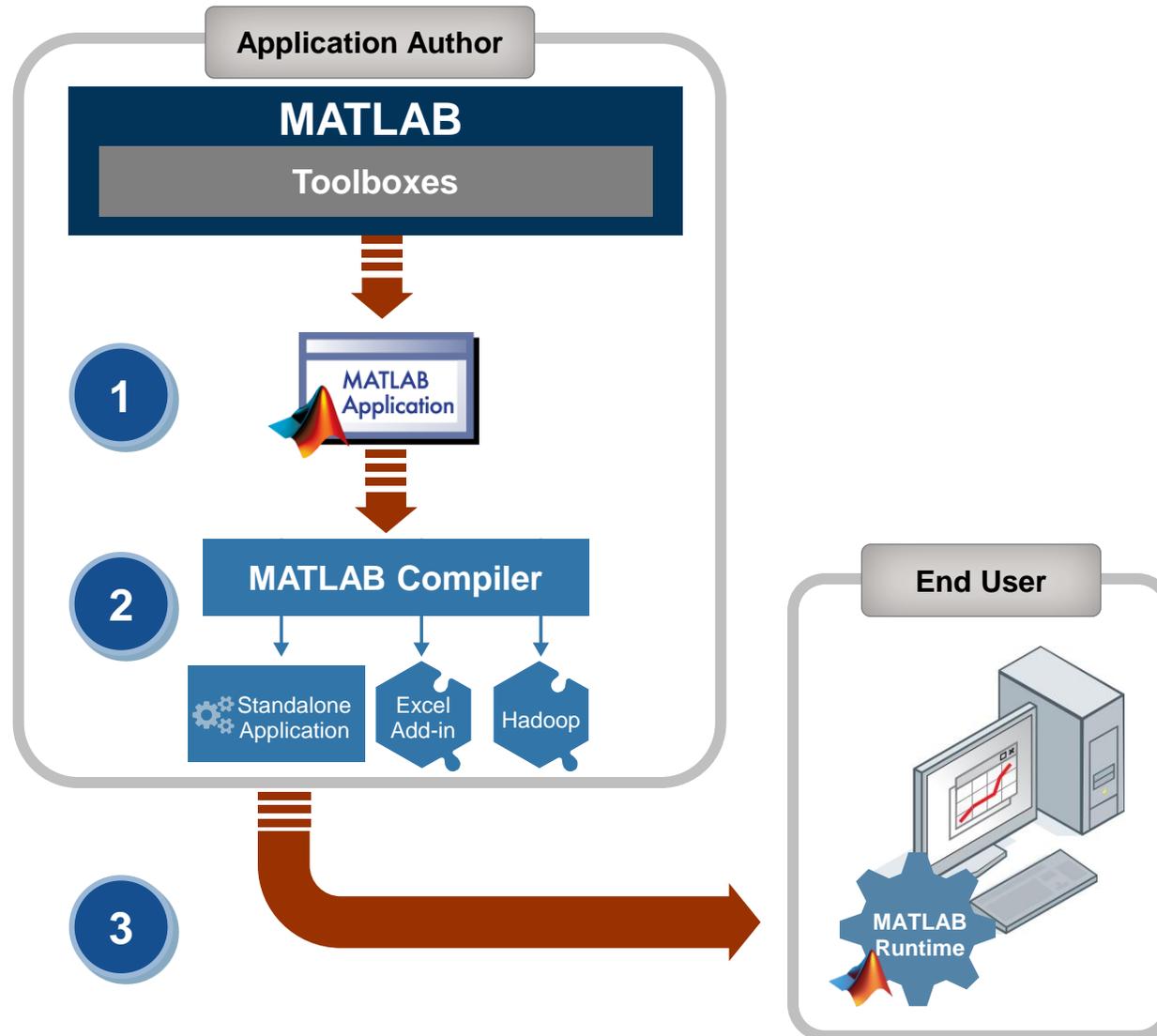
Integrating MATLAB into production systems



Share with non-MATLAB Users: MATLAB Compiler



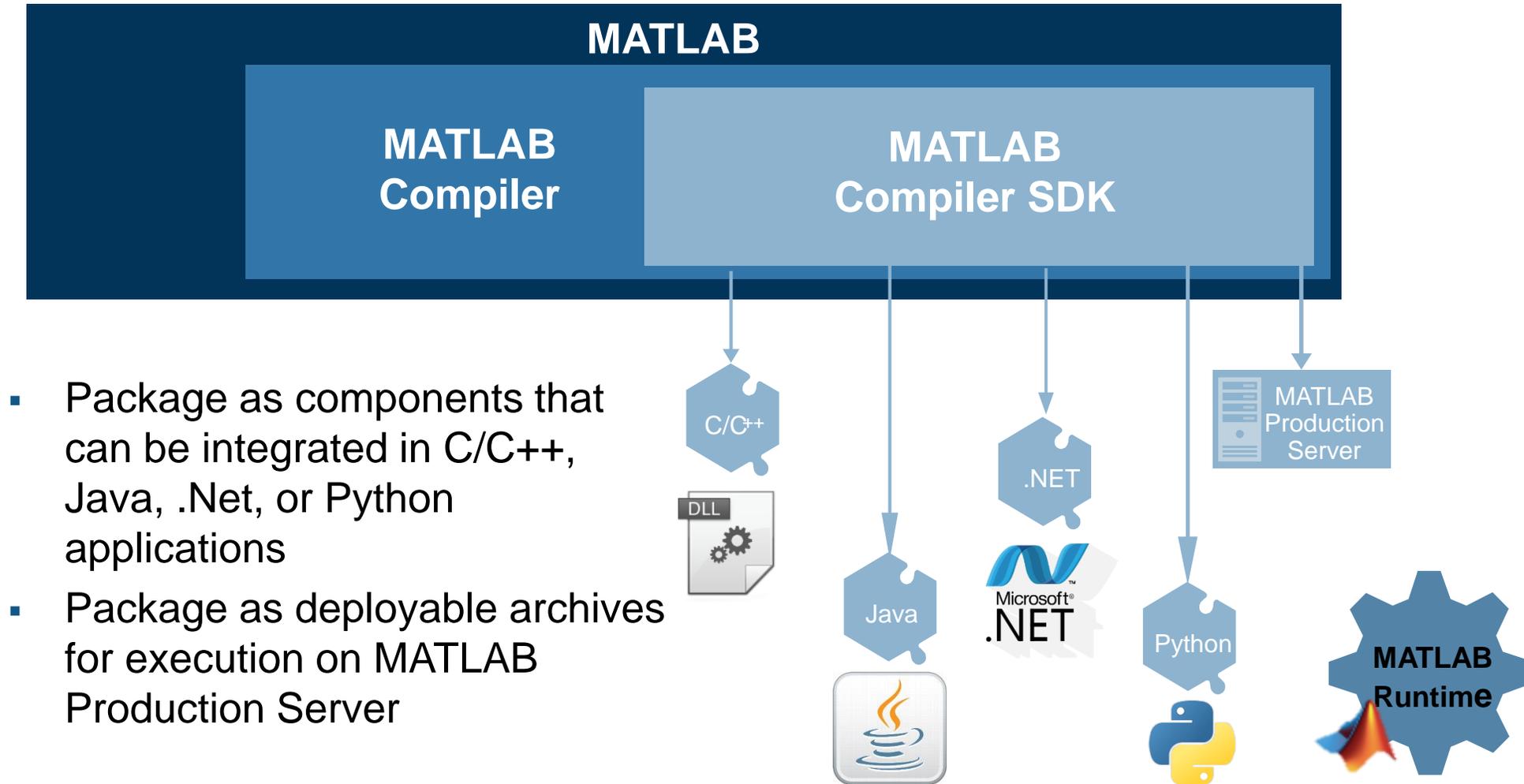
MATLAB Compiler Workflow



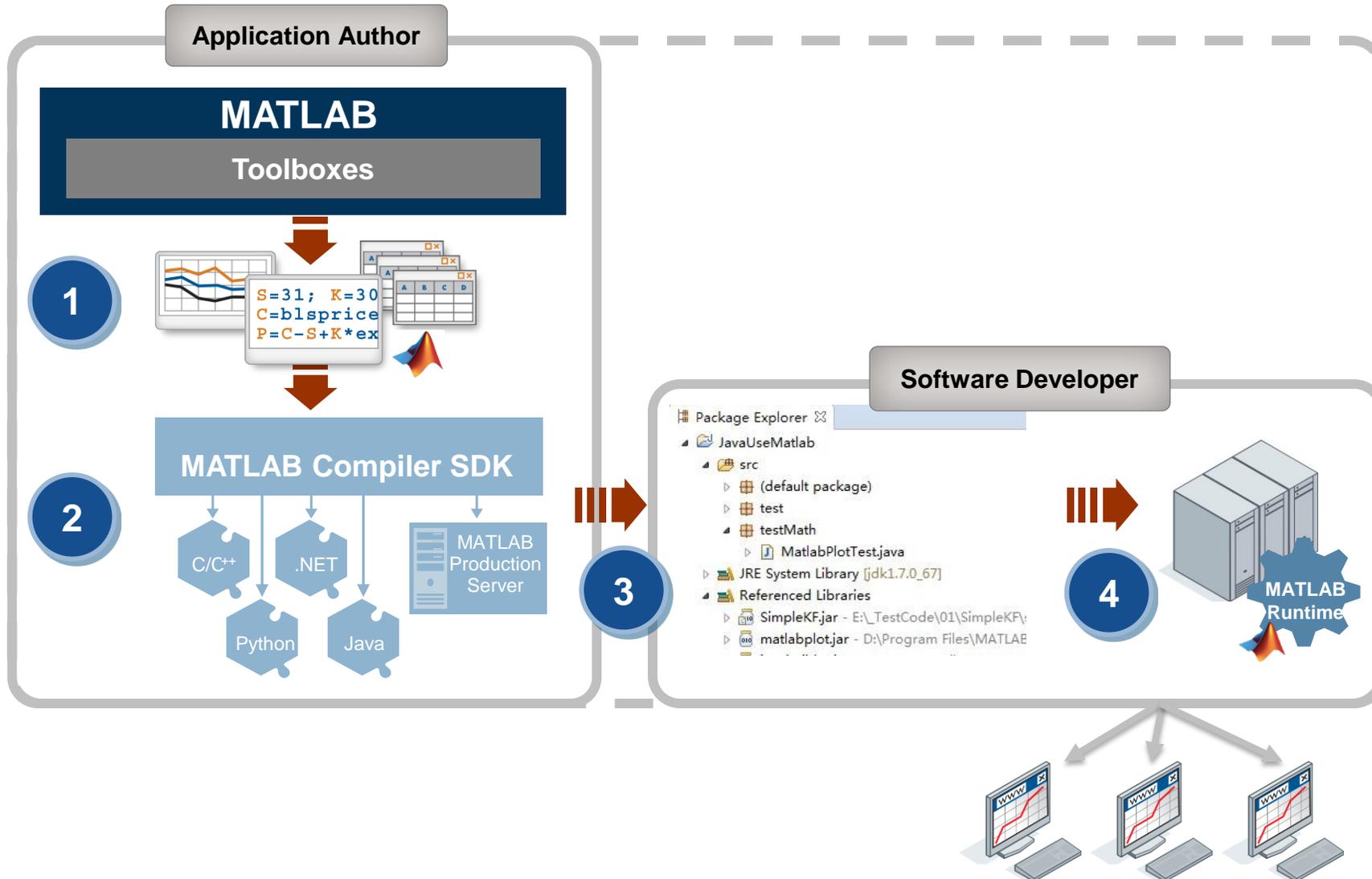
MATLAB in the Enterprise

- Goals:
 - Enable customers to rapidly develop and deploy MATLAB applications onto the desktop.
 - Seamlessly integrate MATLAB generated components into other languages, applications and enterprise production systems.

Share with non-MATLAB Users: MATLAB Compiler SDK

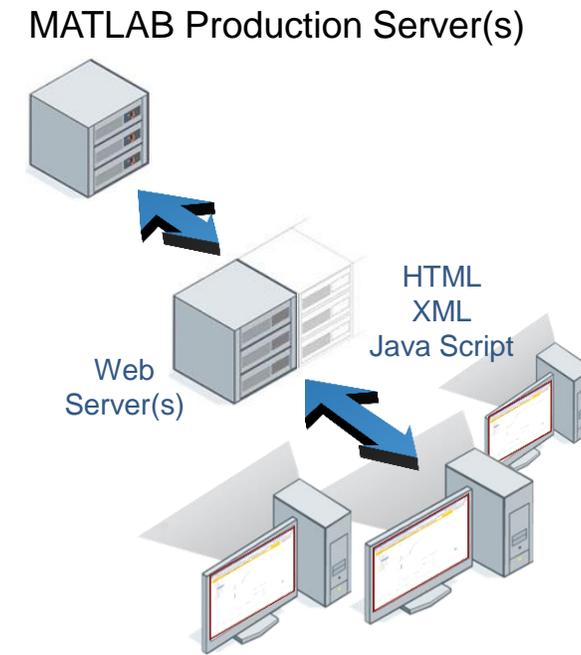


MATLAB Compiler SDK Workflow

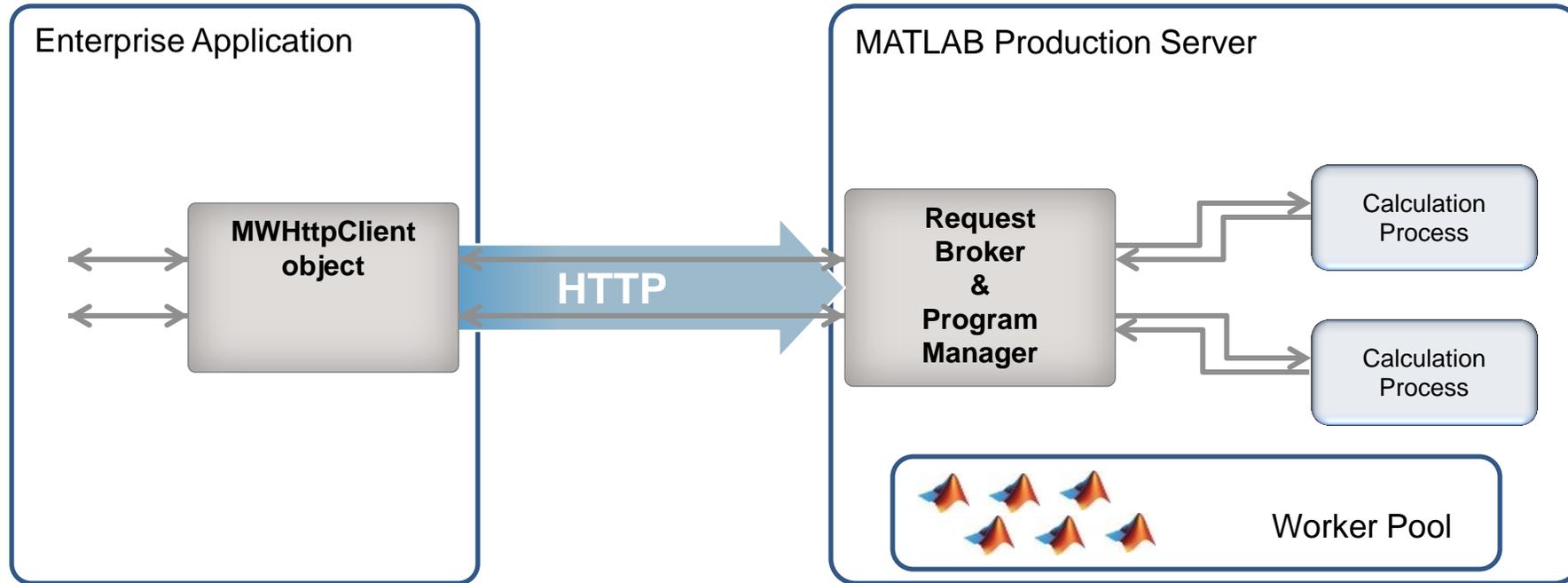


MATLAB Production Server

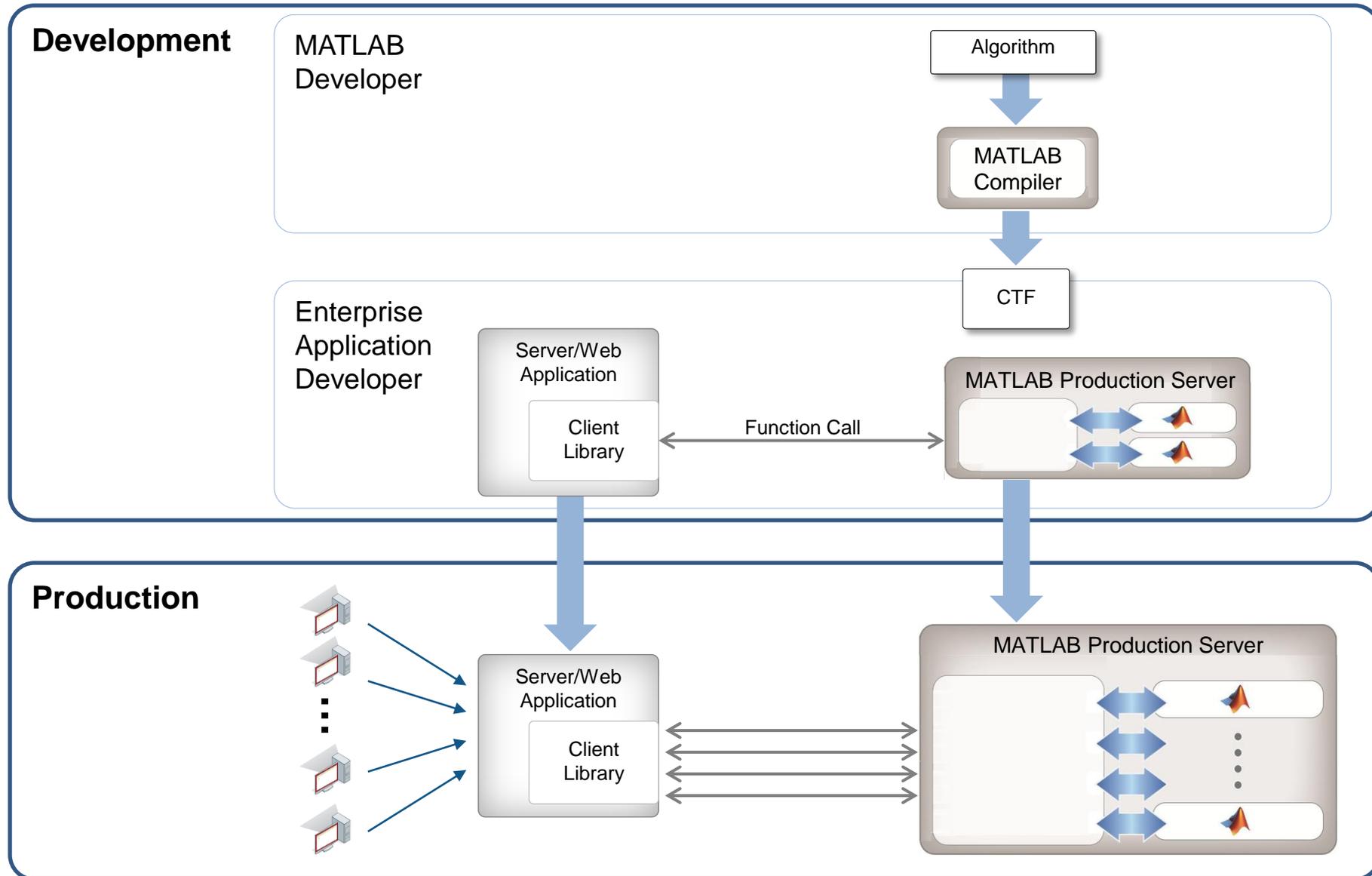
- Directly deploy MATLAB programs into production
 - Centrally manage multiple MATLAB programs & MCR versions
 - Automatically deploy updates without server restarts
- Scalable & reliable
 - Service large numbers of concurrent requests
 - Add capacity or redundancy with additional servers
- Use with web, database & application servers
 - Lightweight client library isolates MATLAB processing
 - Access MATLAB programs using native data types



Calling Functions



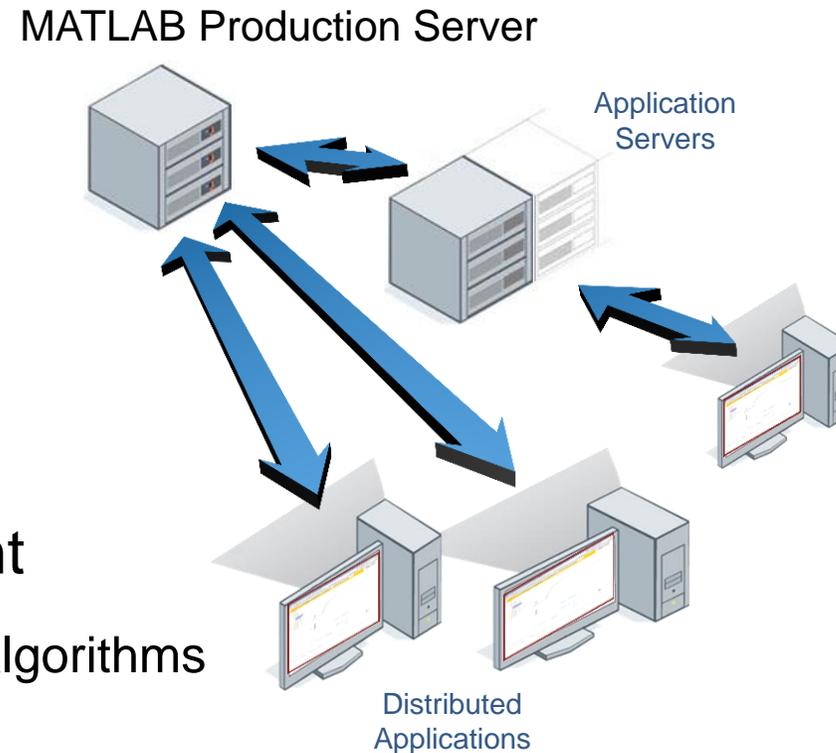
Production Deployment Workflow



Central Management

Centrally run and manage numerical algorithms

- Simplifies applications
 - Analytics run within datacenter
 - UI and business functionality
- Simplifies change management
 - Independent update of numerical algorithms



Benefits of the MATLAB Production Server

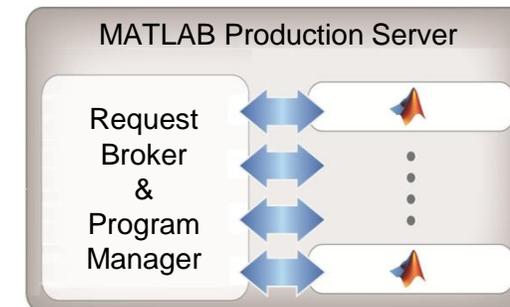
- Enterprise class framework for running packaged MATLAB programs
- Server software
 - Manages packaged MATLAB programs & worker pool
- Manages MATLAB Runtime libraries for multiple releases
 - MATLAB Compiler Runtime (MCR) for various versions of MATLAB from R2012b onward live on the server.
 - Compiled MATLAB analytics from different versions of MATLAB from R2012b onward can co-exist on the server.
- Lightweight client library for .NET and Java frameworks, C/C++, and Python are supported.
- Reduces the Total Cost Of Ownership for building and supporting in-house financial analytics development and deployment.

Flexible System to Manage

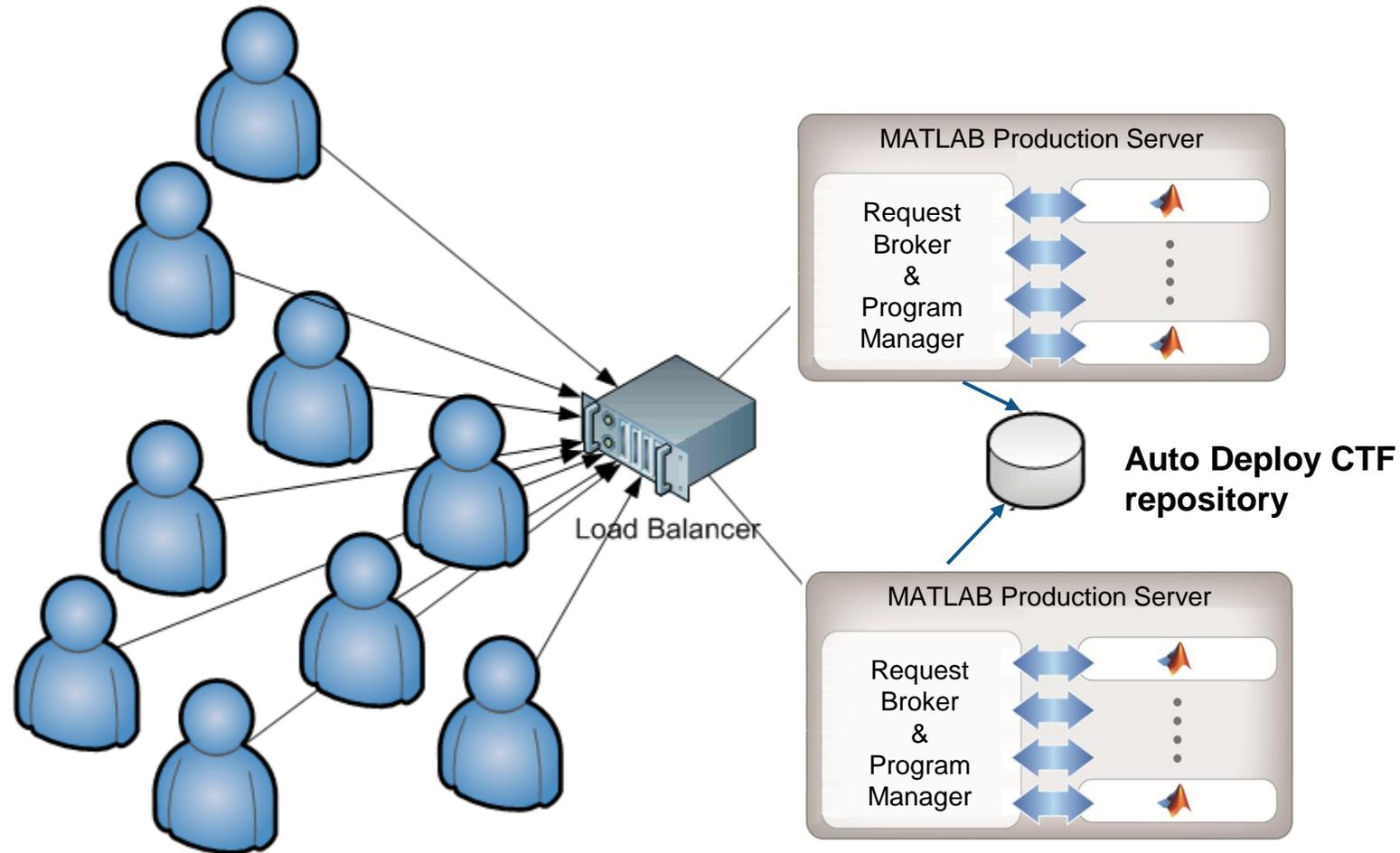
- Licensed on workers/worker threads not on Broker process
- Infinitely configurable to take advantage of existing in-house hardware
- Hosted Analytics platform that is installed in house enabling rapid updating and deployment of analytics/models
- Accessed by any front end application by means of thin client communications library, or through JSON/Restful interface

MATLAB Production Server ... Customer Configurations (1)

- Request Broker and 24 worker processes
- Can have multiple instances of the MATLAB Production Server
 - 2 request brokers and twelve worker servers
 - 3 request brokers and eight worker servers
 - 4 request brokers and six worker servers
 - 6 request brokers and four worker servers
- Increase capacity by increasing number of servers and combining them
 - One request broker and 48 worker processes



MATLAB Production Server ... Customer Configurations (2)



Easy Integration

- IT can efficiently integrate models/analytics in to production system
- Seamless integration into .NET and Java development environments ... only a few lines of code required
- Time to deploy greatly reduced
 - Only need to supply function signature from Quant to IT for implementation into Enterprise system
 - Updates easily implemented and redeploy new model version

Integration Example ... Java

- Reference client library
- Define function signatures
- Define connection (server & CTF)

MATLAB Function

```
function B = BlackScholes(CP,S,X,T,r,v)

d2=d1-v*sqrt(T);
if CP=='c'
B = (S*normcdf(d1)-X*exp(-r*T))*normcdf(d2)-noise;
```

Enterprise Application

```
import com.mathworks.mps.client.MWClient;
import com.mathworks.mps.client.MWHttpClient;
import com.mathworks.mps.client.MATLABException;;

public interface BlkSchInterface
{
    double BlackScholes(string C, double S, double X, double T, double r, double v); }

MWClient client = new MWHttpClient();
BlkSchInterface blksch_1 = client.CreateProxy(new URL("http://192.168.240.220:9910/BlkSch1"), BlkSchInterface);
double optionprice = blksch_1.BlackScholes("c", BasePrice.Value, 1, 1, 1, Volatility.Value));
```

Integration ExampleNET

- Reference client library
- Define function signatures
- Define connection (server & CTF)

MATLAB Function

```
function B = BlackScholes(CP,S,X,T,r,v)

d2=d1-v*sqrt(T);
if CP=='c'
B = (S*normcdf(d1)-X*exp(-r*T))*normcdf(d2)-noise;
```

Enterprise Application

```
using Mathworks.MATLAB.ProductionServer.Client;

public interface BlkSchInterface
{
    double BlackScholes(string C, double S, double X, double T, double r, double v); }

MWClient client = new MWHttpClient();
BlkSchInterface blksch_1 = client.CreateProxy<BlkSchInterface>(new Uri("http://192.168.240.220:9910/BlkSch1"));
double optionprice = blksch_1.BlackScholes("c", BasePrice.Value, 1, 1, 1, Volatility.Value));
```

JSON/Restful API

- Easy API to use
- No client library required

MATLAB Function

```
function B = BlackScholes(CP,S,X,T,r,v)

d2=d1-v*sqrt(T);
if CP=='c'
B = (S*normcdf(d1)-X*exp(-r*T)*normcdf(d2))-noise;
```

Enterprise Application

```
var cp = parseFloat(document.getElementById('coupon_payment_value').value);
var np = parseFloat(document.getElementById('num_payments_value').value);
var ir = parseFloat(document.getElementById('interest_rate_value').value);
var vm = parseFloat(document.getElementById('facevalue_value').value);
```

```
// A new XMLHttpRequest object
var request = new XMLHttpRequest();
```

```
//Use MPS RESTful API to specify URL
var url = "http://localhost:9910/BondTools/pricecalc";
```

```
//Use MPS RESTful API to specify params using JSON
var params = { "nargout":1,"rhs": [vm, cp, ir, np] };
```

```
request.open("POST", url);
```

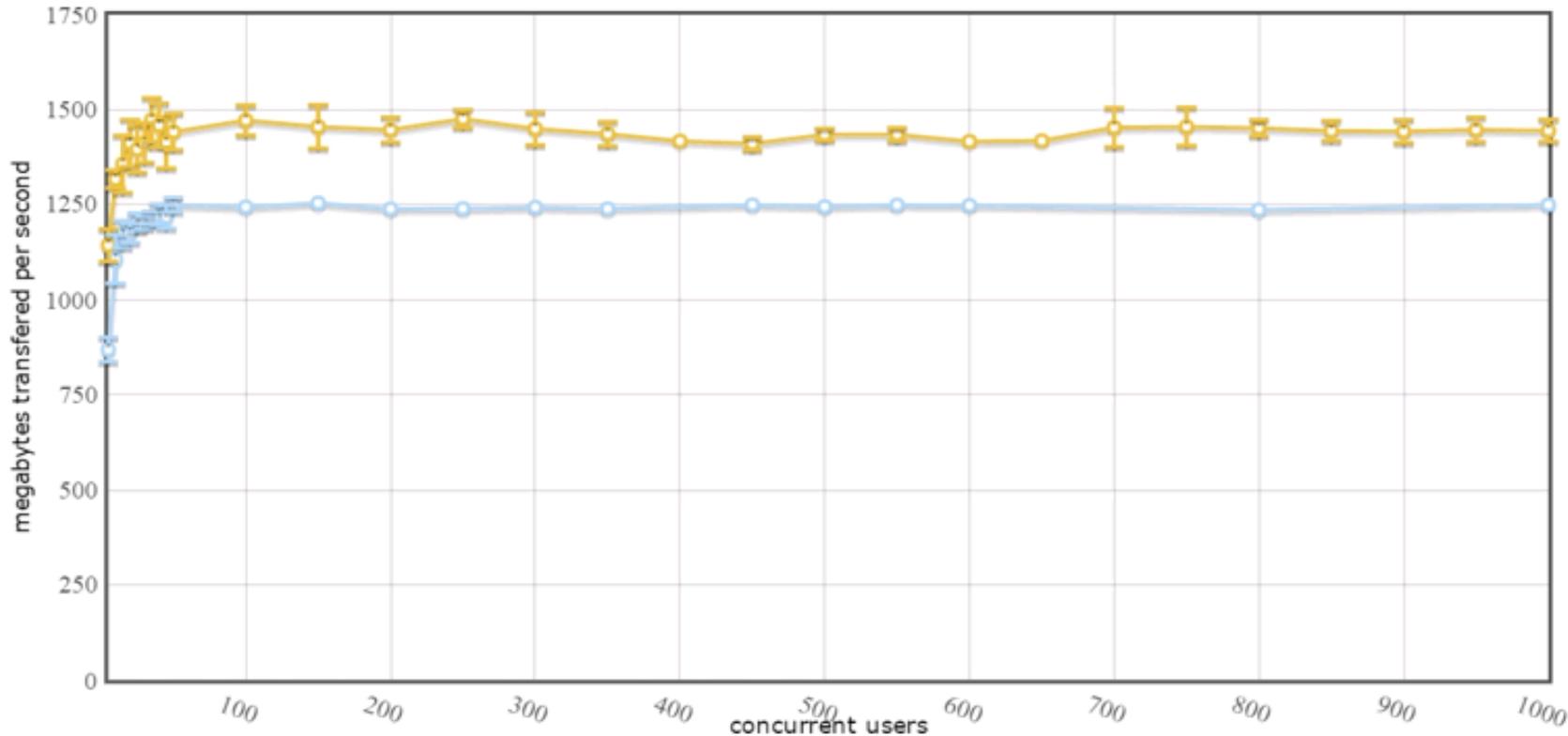
```
//Use MPS RESTful API to set Content-Type
request.setRequestHeader("Content-Type", "application/json");
```

```
request.send(JSON.stringify(params));
```

Where we are today with MATLAB Production Server

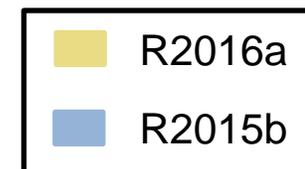
R2016a

- Improved throughput and overall performance



1.5 Gb/sec transfer rate for 1 through 1000 concurrent users

50 μ sec latency for 1 through 1000 concurrent users

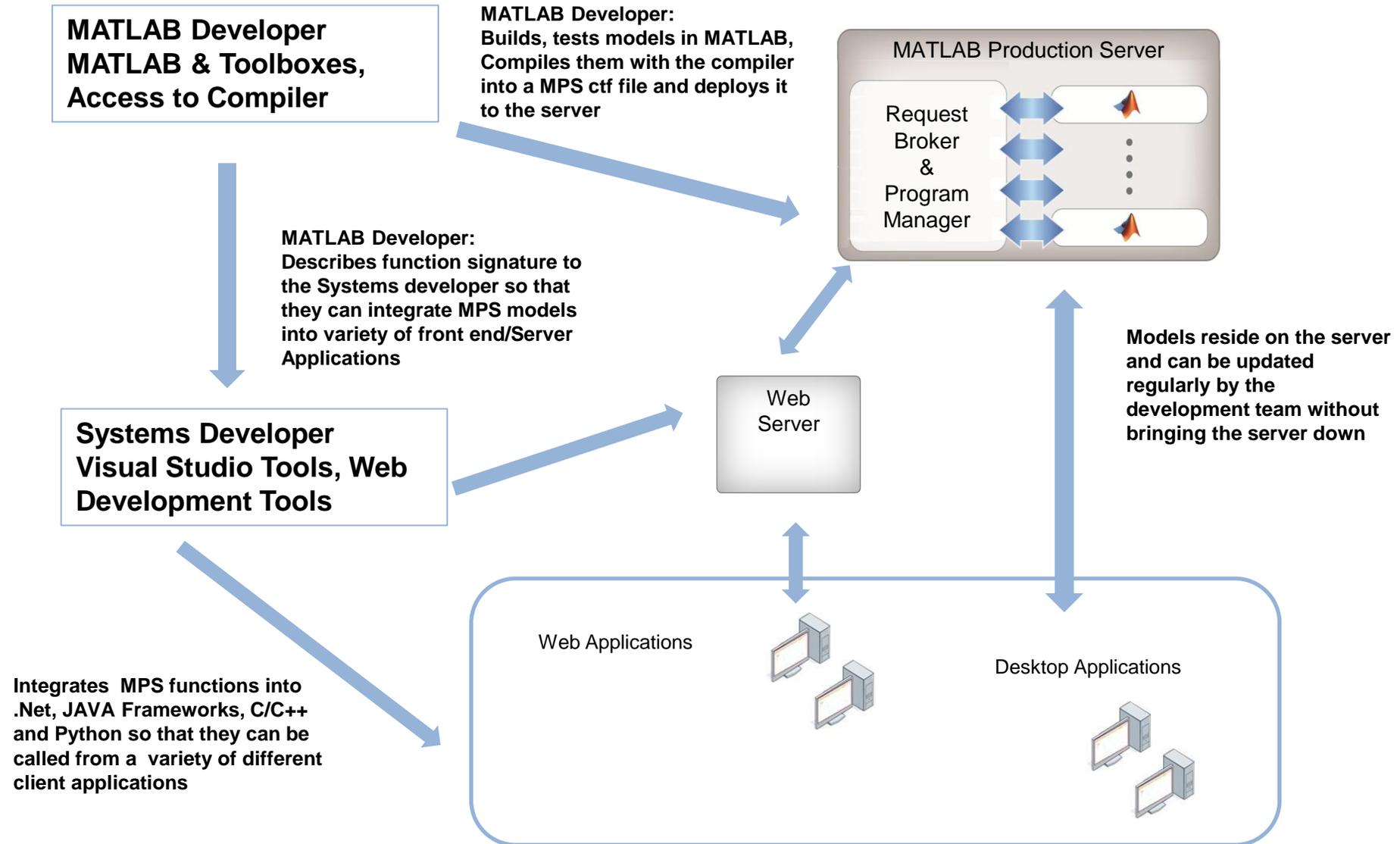


MATLAB Development to Production Workflow

Reduce Cost of Building and Deploying In-House Analytics

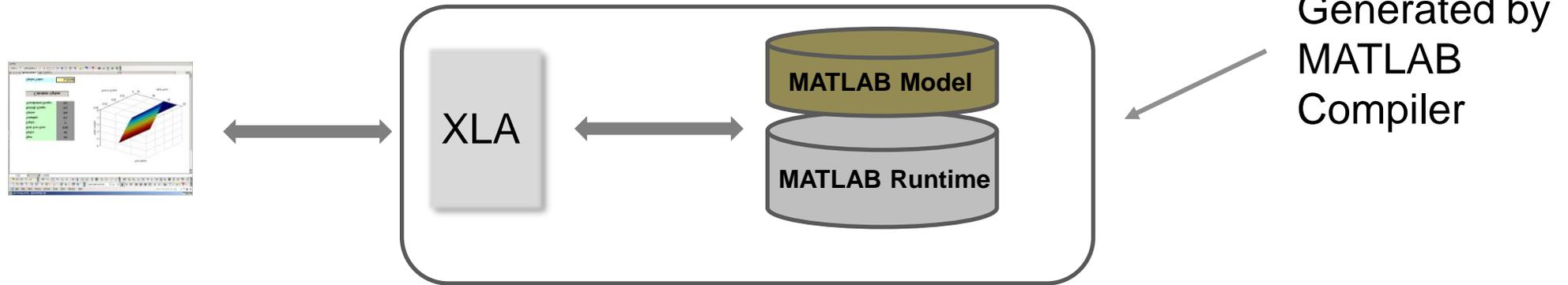
- Single development environment for model development and testing.
- Quants/Analysts/Financial Modelers do not have to rewrite code in another language.
- IT can efficiently integrate models/analytics in to production system
- Time to deploy greatly reduced
 - Only need to supply function signature from Quant to IT for implementation into Enterprise system
 - Updates easily implemented and redeploy new model version

Separation of Roles in Building and Deploying In-House Analytics

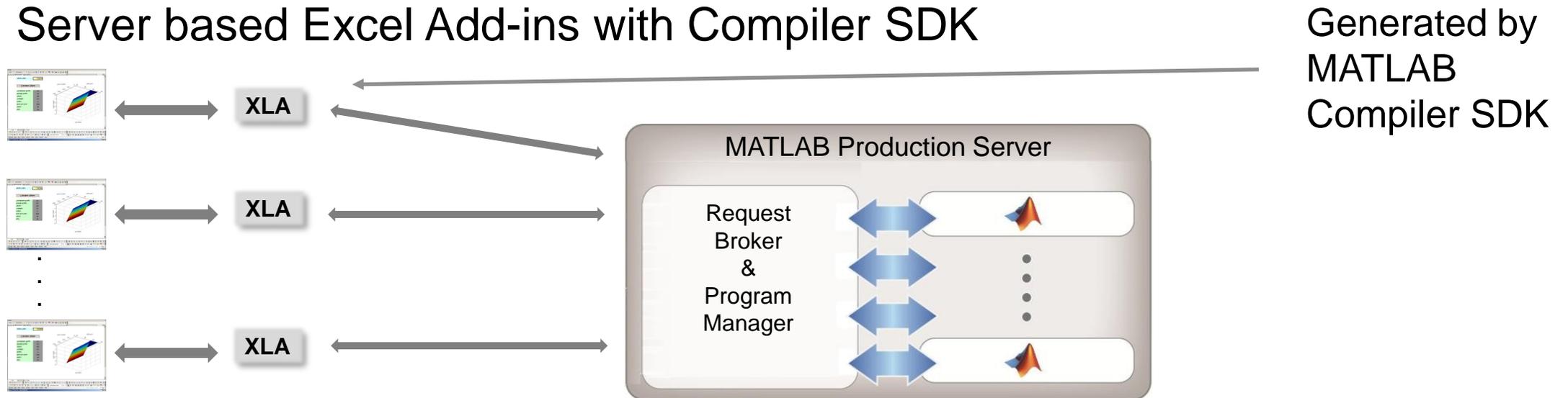


Desktop and Server Based Excel Add-ins

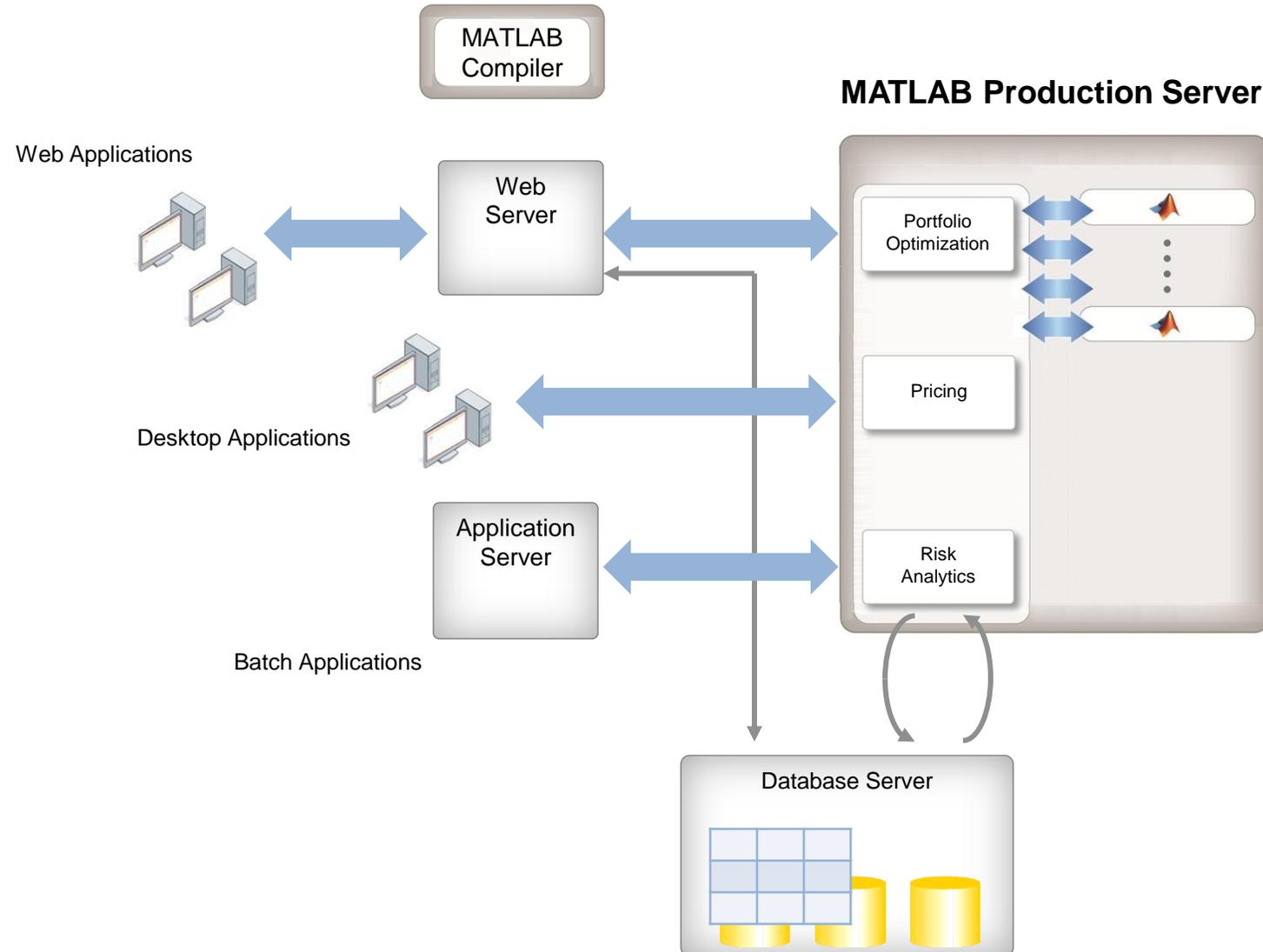
- Desktop Excel Add-ins with compiler



- Server based Excel Add-ins with Compiler SDK

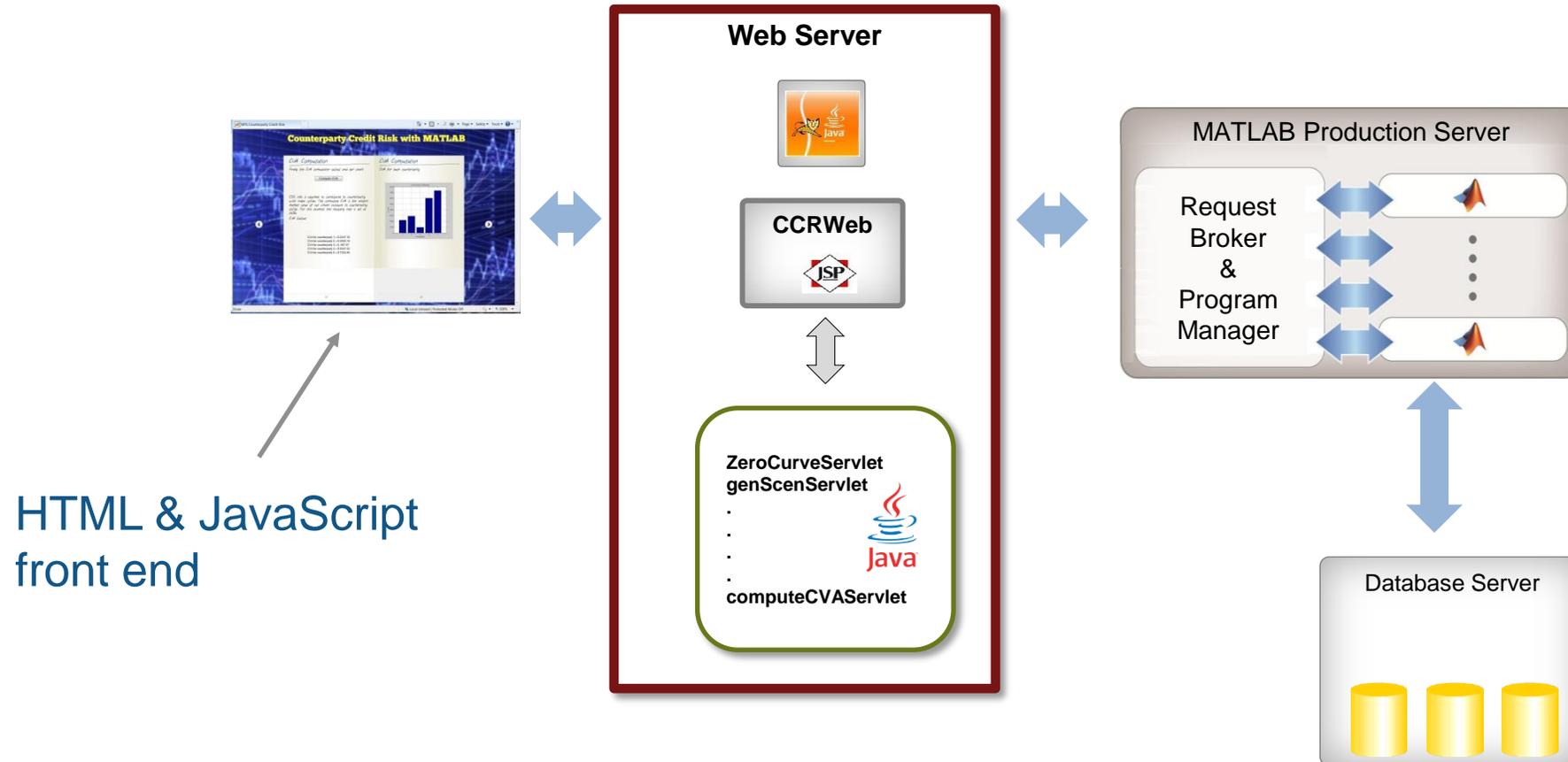


Sharing algorithms across the organization



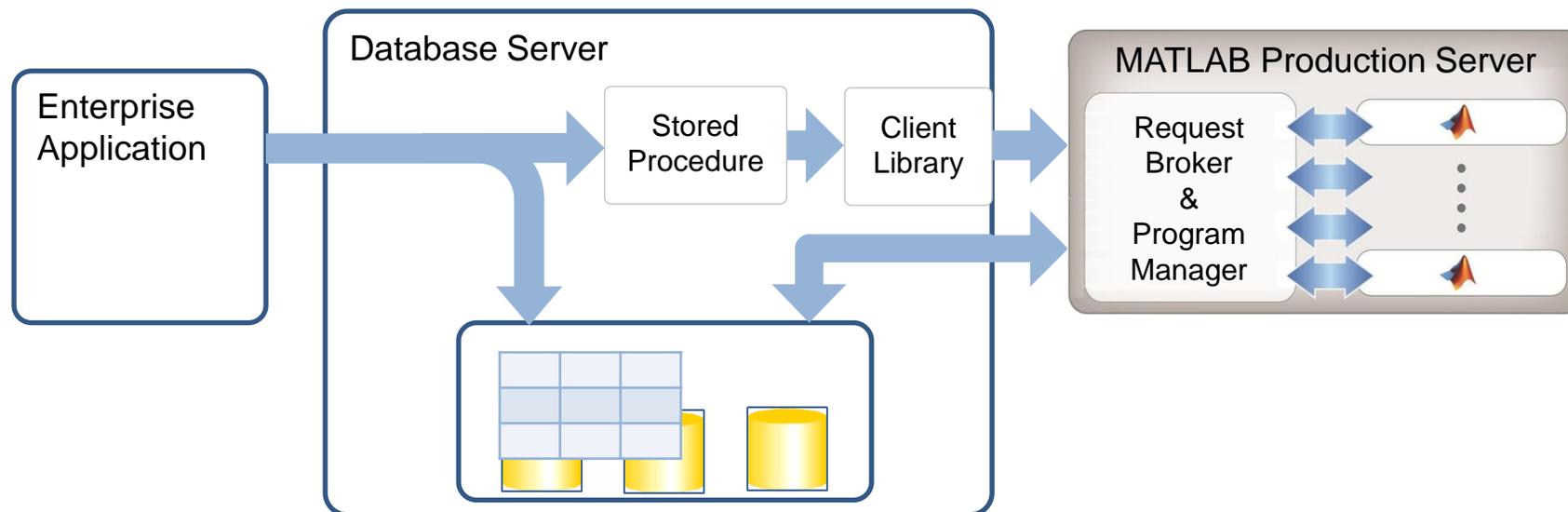
Web Applications - Using JAVA, C# client libraries

- Architecture



Integration with Databases

- Optimize numerical processing within databases
 - Request MATLAB analytics directly from database servers
 - Trigger requests based upon database transactions
- Minimize data handling using Database Toolbox

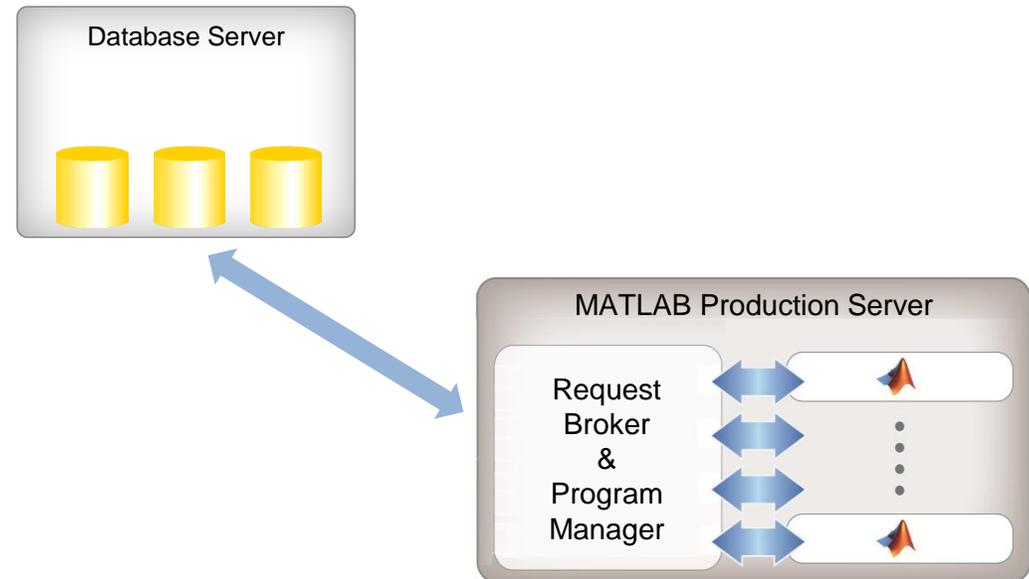


MATLAB Components in Production Databases

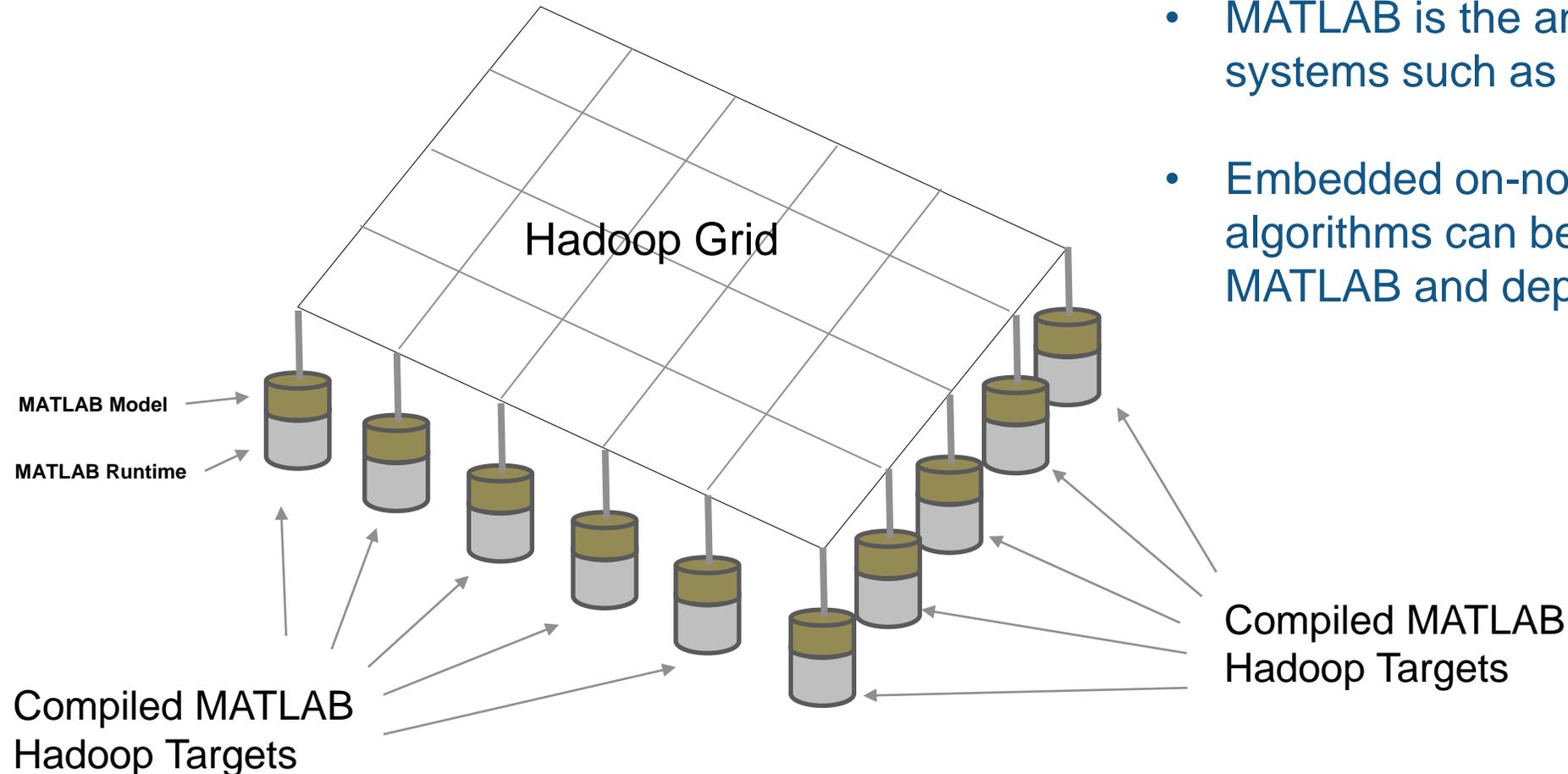
MATLAB Production Server can provide predictive analytics in the database

- Oracle (Java, .NET)
- Microsoft SQL Server (.NET)
- Microsoft Access (.NET)
- Netezza (JAVA)
- SAS (JAVA)
- Teradata (JAVA)

- Thin client with MPS
 - Java and .NET supported
- Central repository for models ... Simplifies change management



MATLAB Big Data Analytic Components - Hadoop



- MATLAB is the analytics for big data systems such as Hadoop
- Embedded on-node custom algorithms can be developed in MATLAB and deployed to the grid

Compiler generated Hadoop components on each node of Hadoop Grid

MATLAB Production Server Use Cases

