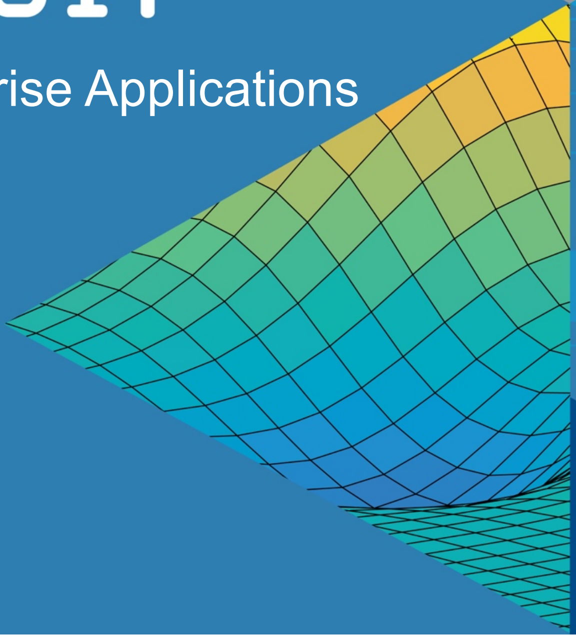
MATLAB EXPO 2017

Integrating MATLAB Analytics into Enterprise Applications



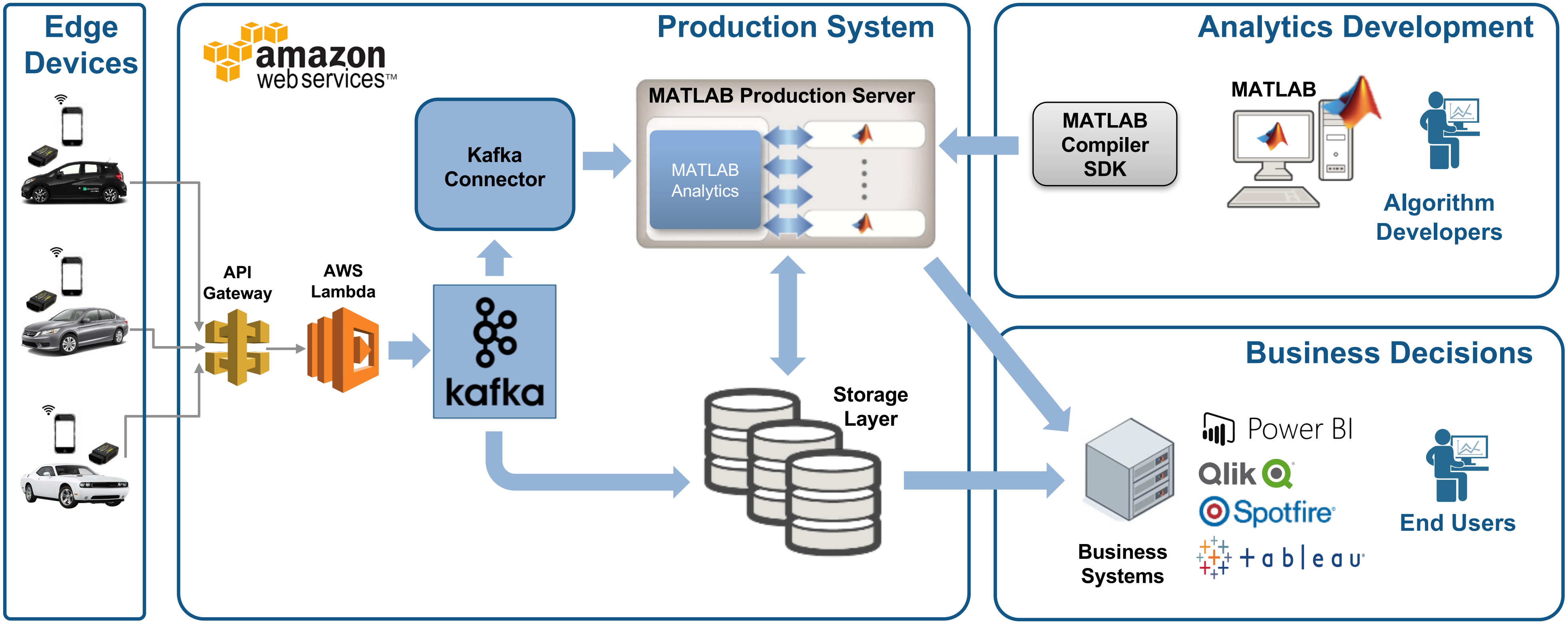
Agenda

- Example Problem
- Access and Preprocess Data
- Develop a Predictive Model
- Integrate Analytics with Production Systems
- Build a Dashboard to Visualize Results

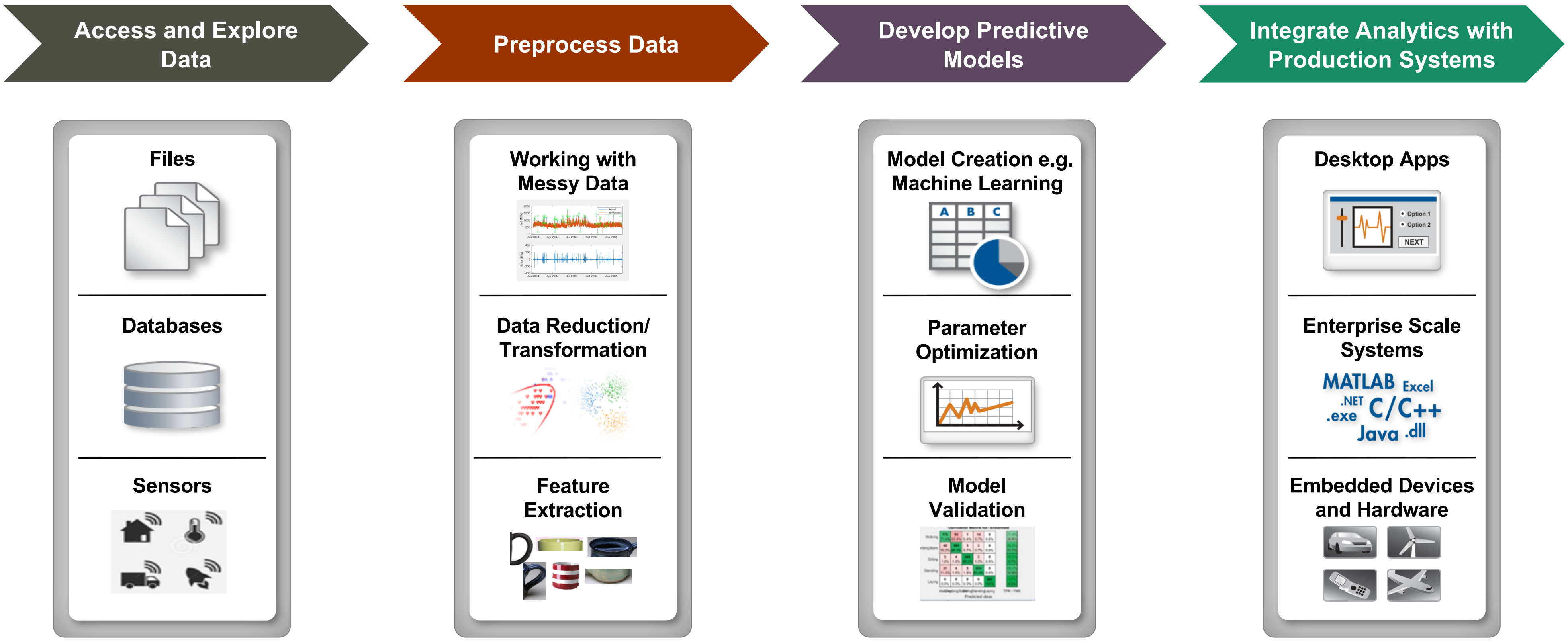
Example Problem – How's my driving?

- A group of MathWorks employees installed an OBD dongle in their car that monitors the on-board systems
- Data is streamed to the cloud where it is aggregated and stored
- I would like to use this data to score the driving habits of participants

Fleet Analytics Architecture



Data Analytics Workflow



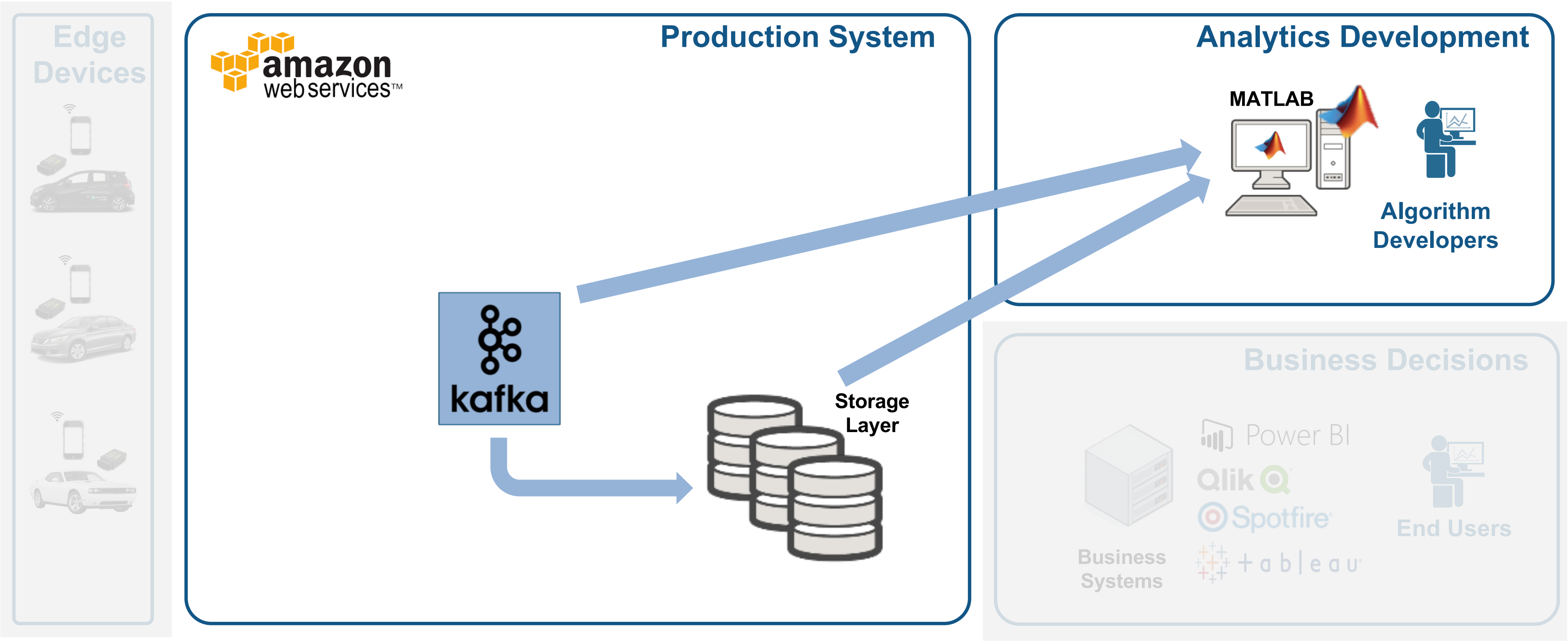
Access and Explore Data
- Files
- Databases
- Sensors

Preprocess Data
- Working with Messy Data
- Data Reduction/ Transformation
- Feature Extraction

Develop Predictive Models
- Model Creation e.g. Machine Learning
- Parameter Optimization
- Model Validation

Integrate Analytics with Production Systems
- Desktop Apps
- Enterprise Scale Systems
- Embedded Devices and Hardware

Access and Preprocess Data



The Data: Timestamped messages with JSON encoding

Key	Value

{
"vehicles_id":
{"$oid":"55a3fd0069702d5b41000000"}
}
	{
	"time” : {"$date":"2015-07-13T18:01:35.000Z"},
	"kc” : 1975.0, "kff1225" : 100.65293, "kff125a" : 110.36619,
	…..
	}

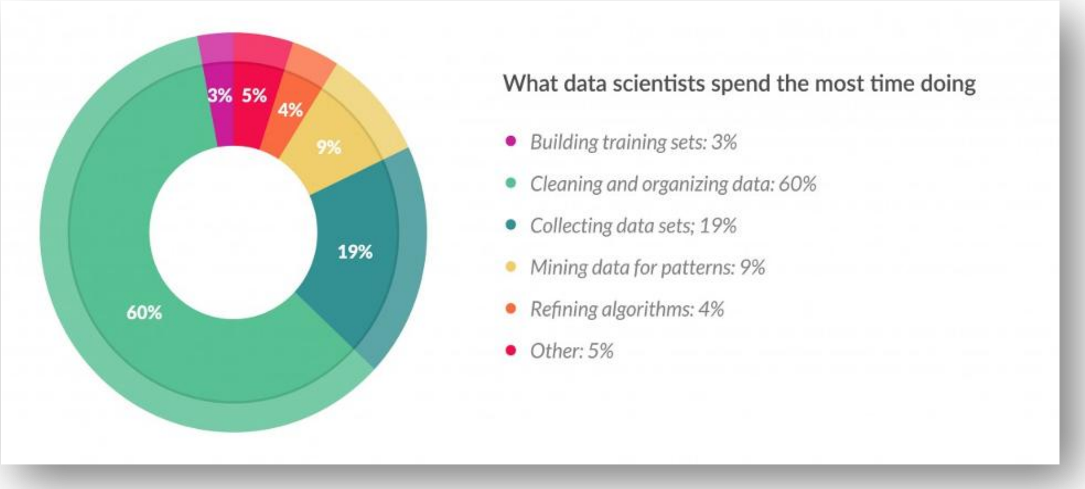
{
"vehicles_id":
{"$oid":"55a3fe3569702d5c5c000020"}
}
	{
	"time":{"$date":"2015-07-13T18:01:53.000Z"},
	"kc” : 2000.0, "kff1225" : 109.65293, "kff125a" : 115.36619,
	…..
	}

{
"vehicles_id":
"$oid":"55a4193569702d115b000001"}
}
	{
	"time":{"$date":"2015-07-12T19:04:04.000Z"}
	"kc":2200.0, "kff1225" : 112.65293, "kff125a" : 112.36619,
	…..
	}

Data Access and Preprocessing – Challenges

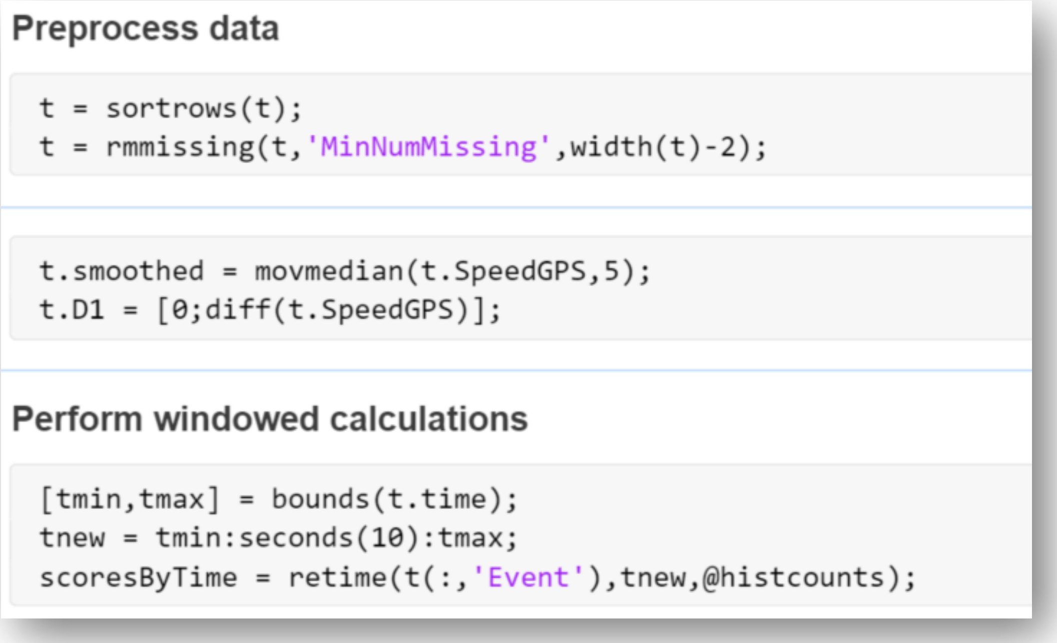
Challenges

- Data aggregation
  - Different sources (files, web, etc.)
  - Different types (images, text, audio, etc.)
- Data clean up
  - Poorly formatted files
  - Irregularly sampled data
  - Redundant data, outliers, missing data etc.
- Data specific processing
  - Signals: Smoothing, resampling, denoising, Wavelet transforms, etc.
  - Images: Image registration, morphological filtering, deblurring, etc.
- Dealing with out of memory data (big data)



Data preparation accounts for about **80%** of the work of data scientists - Forbes

Access a Sample of Data and Develop a Preprocessing Function



Preprocess data

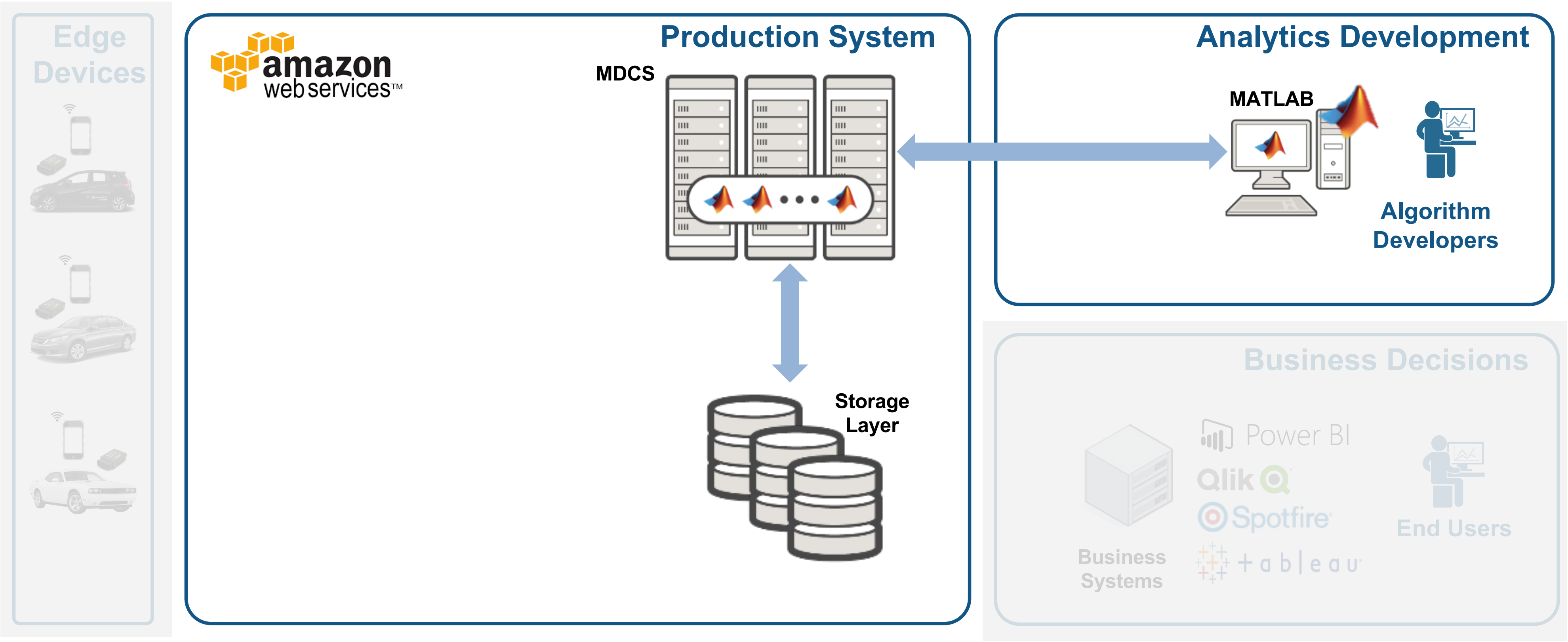
```
t = sortrows(t);
t = rmmissing(t,'MinNumMissing',width(t)-2);
```

```
t.smoothed = movmedian(t.SpeedGPS,5);
t.D1 = [0;diff(t.SpeedGPS)];
```

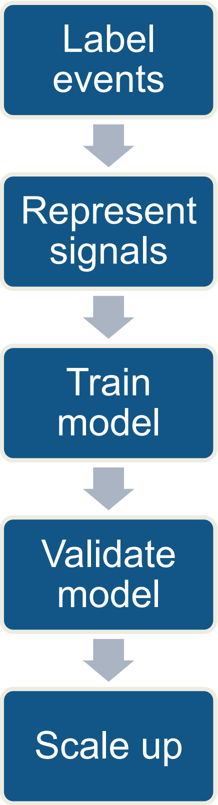
Perform windowed calculations

```
[tmin,tmax] = bounds(t.time);
tnew = tmin:seconds(10):tmax;
scoresByTime = retime(t(:,'Event'),tnew,@histcounts);
```

Develop a Predictive Model



Develop a Predictive Model in MATLAB



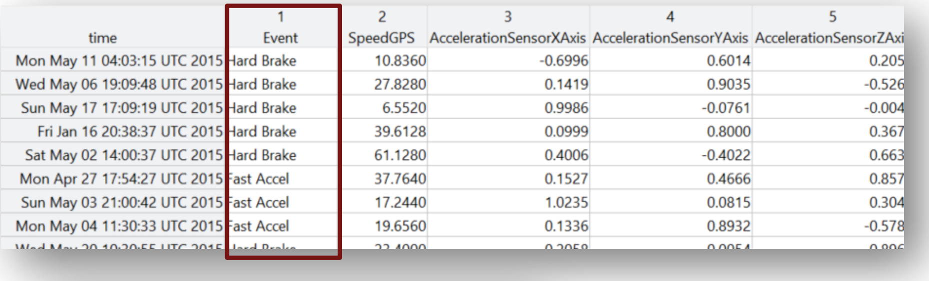
Label events

Represent signals

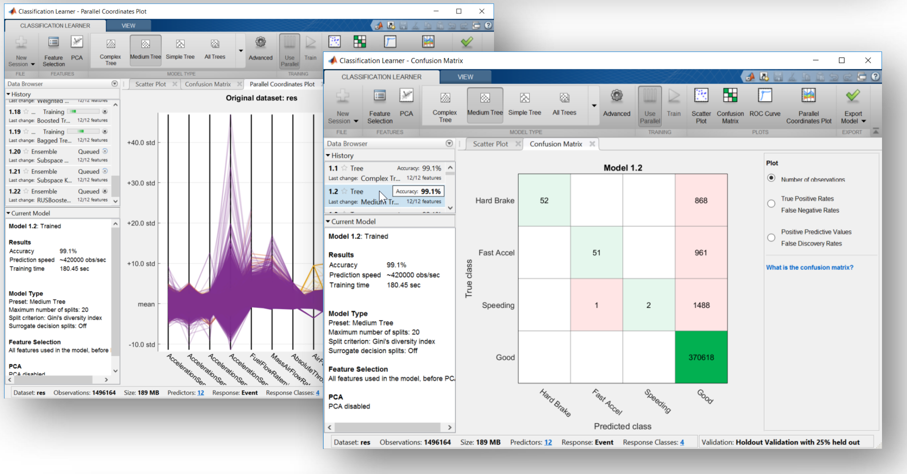
Train model

Validate model

Scale up



	1	2	3	4	5
time	Event	SpeedGPS	AccelerationSensorXAxis	AccelerationSensorYAxis	AccelerationSensorZAxi
Mon May 11 04:03:15 UTC 2015	Hard Brake	10.8360	-0.6996	0.6014	0.205
Wed May 06 19:09:48 UTC 2015	Hard Brake	27.8280	0.1419	0.9035	-0.526
Sun May 17 17:09:19 UTC 2015	Hard Brake	6.5520	0.9986	-0.0761	-0.004
Fri Jan 16 20:38:37 UTC 2015	Hard Brake	39.6128	0.0999	0.8000	0.367
Sat May 02 14:00:37 UTC 2015	Hard Brake	61.1280	0.4006	-0.4022	0.663
Mon Apr 27 17:54:27 UTC 2015	Fast Accel	37.7640	0.1527	0.4666	0.857
Sun May 03 21:00:42 UTC 2015	Fast Accel	17.2440	1.0235	0.0815	0.304
Mon May 04 11:30:33 UTC 2015	Fast Accel	19.6560	0.1336	0.8932	-0.578



History

1.1 Tree Accuracy: 99.1%
Last change: Complex Tr... 12/12 features

1.2 Tree Accuracy: 99.1%
Last change: Medium Tr... 12/12 features

Current Model

Model 1.2: Trained

Results

Accuracy 99.1%
Prediction speed ~420000 obs/sec
Training time 180.45 sec

Model Type

Preset: Medium Tree
Maximum number of splits: 20
Split criterion: Gini's diversity index
Surrogate decision splits: Off

Feature Selection

All features used in the model, before PC

PCA

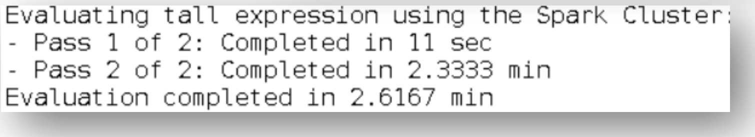
PCA disabled

Model 1.2

True class \ Predicted class	Hard Brake	Fast Accel	Speeding	Good
Hard Brake	52			868
Fast Accel		51		961
Speeding		1	2	1488
Good				370618

● Number of observations
○ True Positive Rates False Negative Rates
○ Positive Predictive Values False Discovery Rates

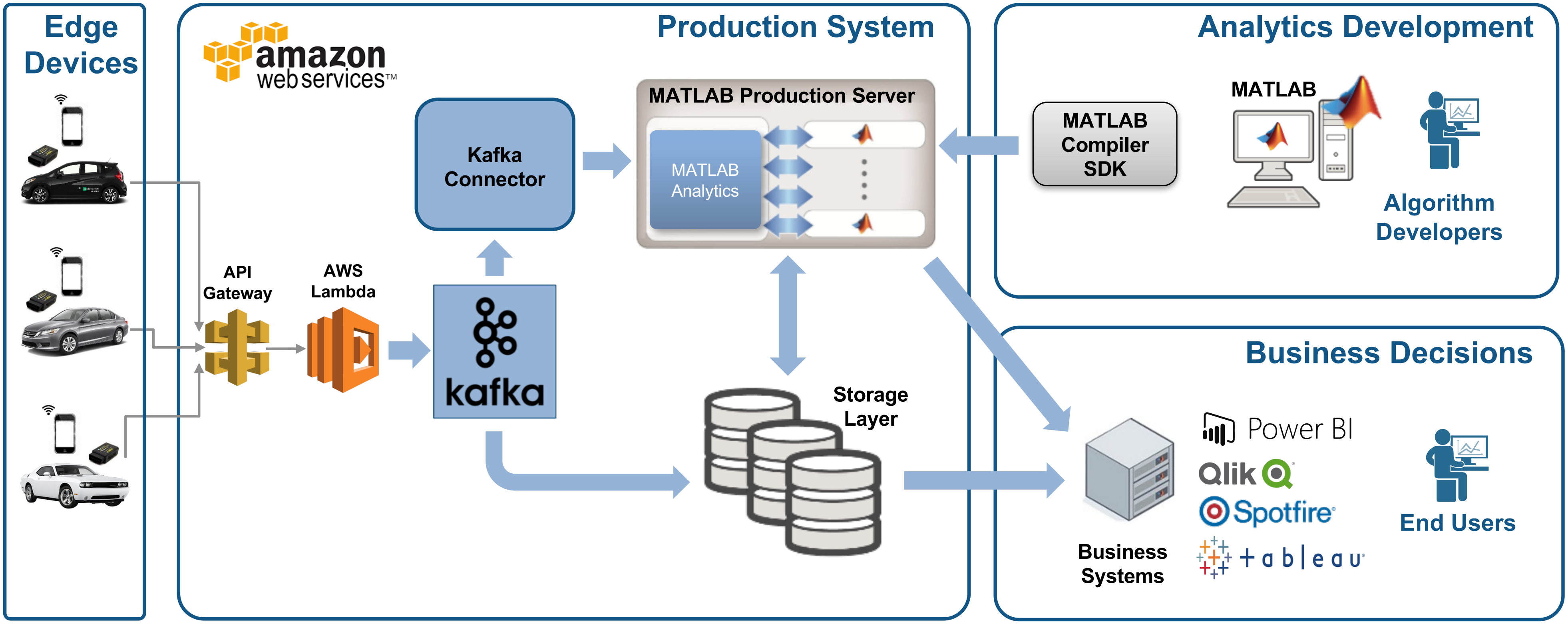
Dataset: **res** Observations: **1496164** Size: **189 MB** Predictors: 12 Response: **Event** Response Classes: 4 Validation: **Holdout Validation with 25% held out**



```
Evaluating tall expression using the Spark Cluster:
- Pass 1 of 2: Completed in 11 sec
- Pass 2 of 2: Completed in 2.3333 min
Evaluation completed in 2.6167 min
```

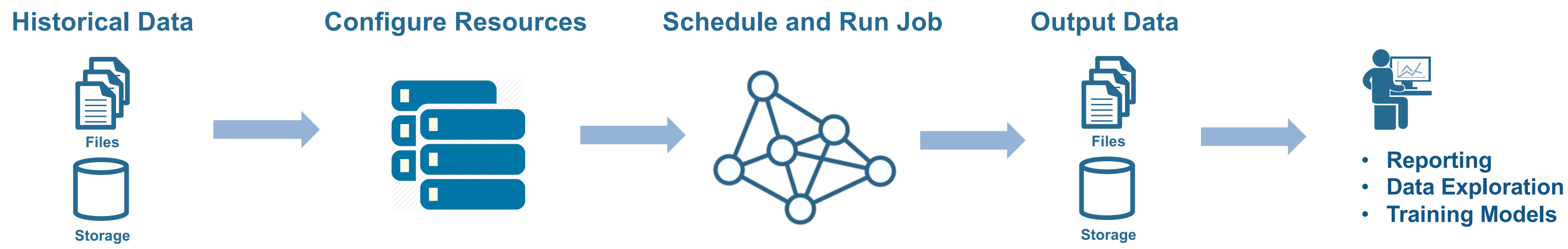
MATLAB EXPO 2017

Integrate Analytics with Production Systems



A quick Intro to Stream Processing

- ***Batch Processing*** applies computation to a finite sized historical data set that was acquired in the past



Historical Data

Files

Storage

Configure Resources

Schedule and Run Job

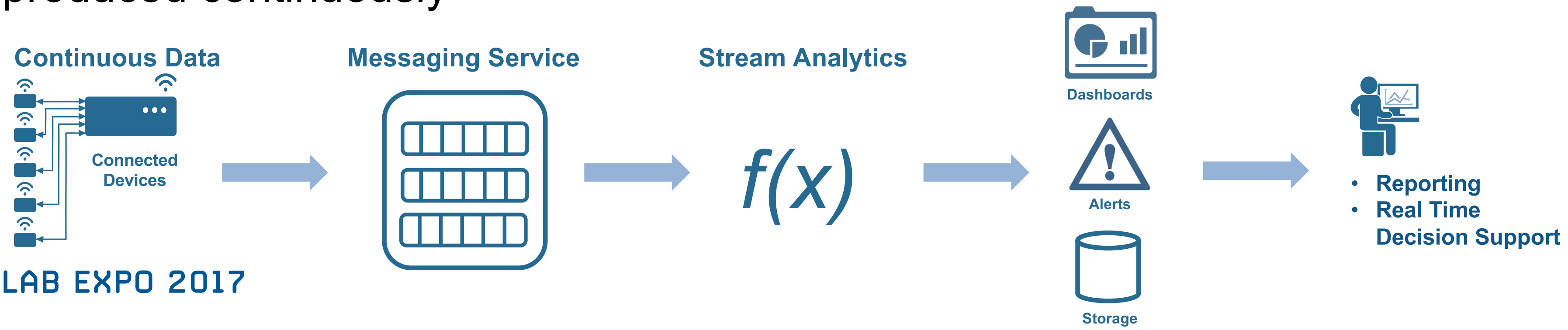
Output Data

Files

Storage

- Reporting
- Data Exploration
- Training Models

- ***Stream Processing*** applies computation to an unbounded data set that is produced continuously



Continuous Data

Connected Devices

Messaging Service

Stream Analytics

$f(x)$

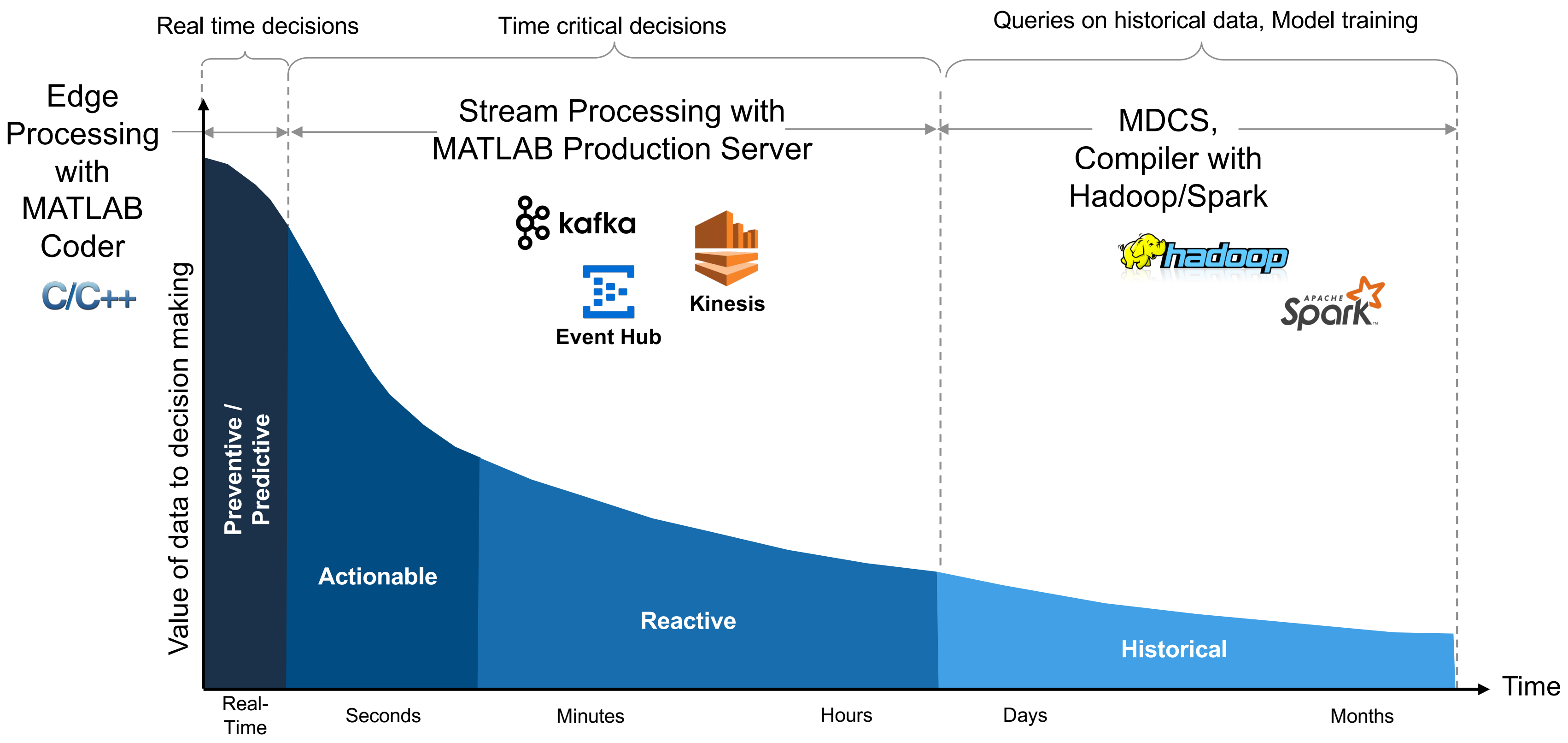
Dashboards

Alerts

Storage

- Reporting
- Real Time Decision Support

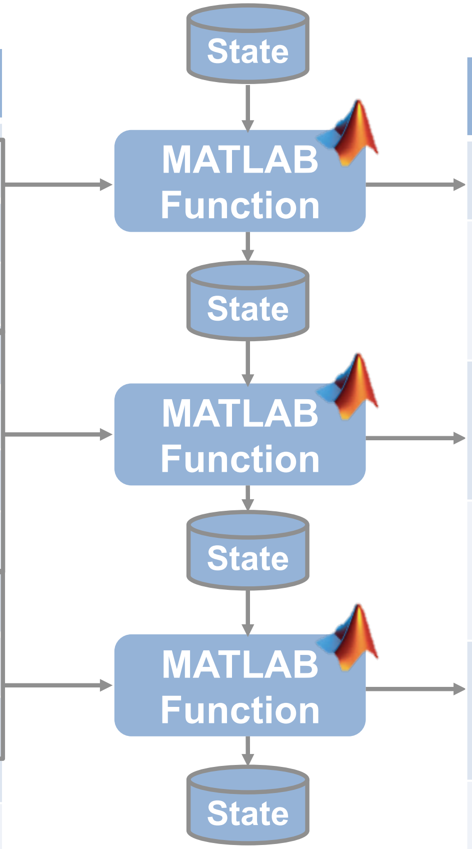
Stream processing exploits the fact that recent data tends to be more valuable


Real time decisions
Time critical decisions
Queries on historical data, Model training
Edge Processing with MATLAB Coder
C/C++
Stream Processing with MATLAB Production Server
kafka
Event Hub
Kinesis
MDCS, Compiler with Hadoop/Spark
hadoop
APACHE Spark
Value of data to decision making
Preventive / Predictive
Actionable
Reactive
Historical
Time
Real-Time
Seconds
Minutes
Hours
Days
Months

Streaming data is treated as an unbounded table

Input Table

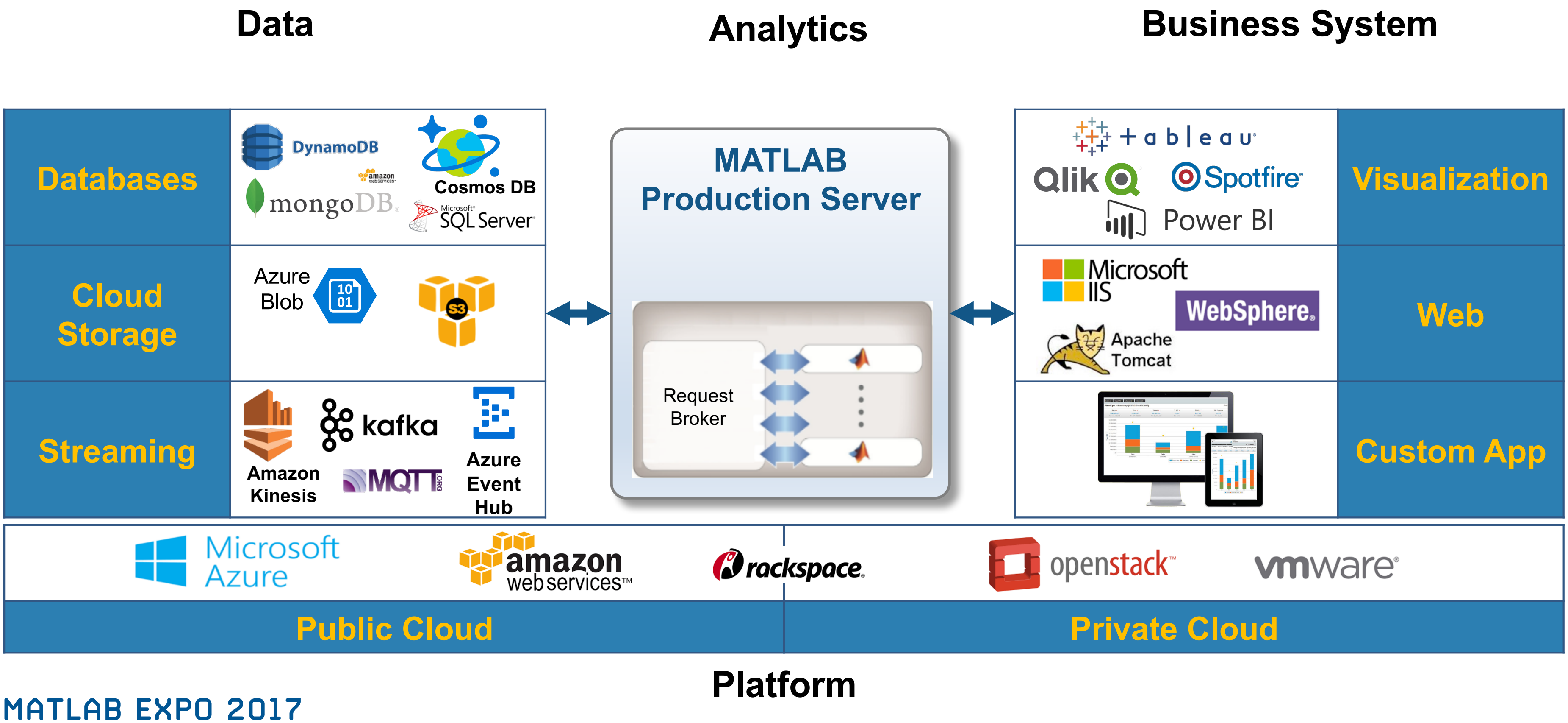
Event Time	Vehicle	RPM	Torque	Fuel Flow
18:01:10	55a3fd	1975	100	110
18:10:30	55a3fe	2000	109	115
18:05:20	55a3fd	1980	105	105
18:10:45	55a3fd	2100	110	100
18:30:10	55a419	2000	100	110
18:35:20	55a419	1960	103	105
18:20:40	55a3fe	1970	112	104
18:39:30	55a419	2100	105	110
18:30:00	55a3fe	1980	110	113
18:30:50	55a3fe	2000	100	110
…	…	…	…	…
…	…	…	…	…



Output Table

Time window		Vehicle	Score
…		…	…
18:00:00	18:10:00	55a3fd	5
		55a3fe	…
		55a419	…
18:10:00	18:20:00	55a3fd	7
		55a3fe	3
		55a419	…
18:20:00	18:30:00	55a3fd	…
		55a3fe	4
		55a419	…
18:30:00	18:40:00	55a3fd	…
		55a3fe	5
		55a419	8
…		…	…

Introducing MATLAB Production Server



Data

Analytics

Business System

Databases: DynamoDB, amazon webservices, mongoDB, Cosmos DB, Microsoft SQL Server

Cloud Storage: Azure Blob, S3

Streaming: Amazon Kinesis, kafka, MQTT.ORG, Azure Event Hub

MATLAB Production Server

Request Broker

Visualization: tableau, Qlik, Spotfire, Power BI

Web: Microsoft IIS, WebSphere, Apache Tomcat

Custom App

Microsoft Azure, amazon web services, rackspace, openstack, vmware

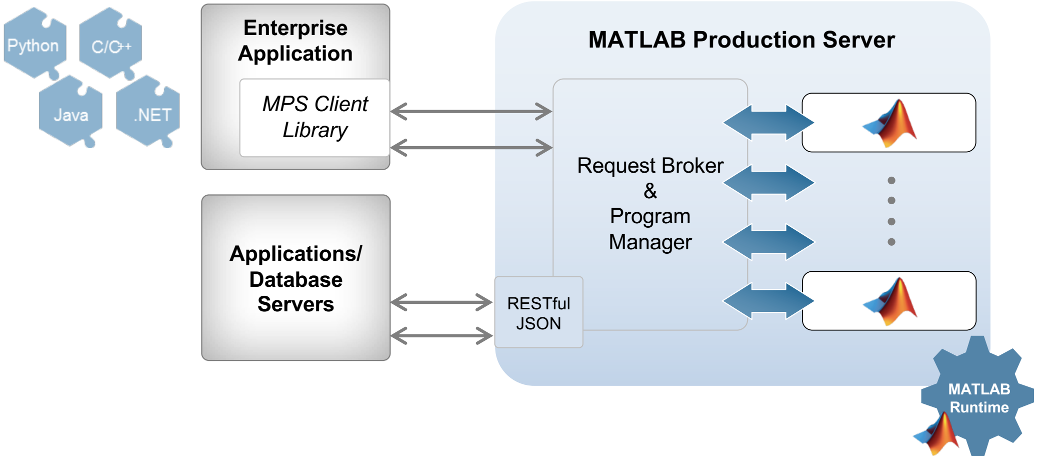
Public Cloud

Private Cloud

Platform

Introducing MATLAB Production Server

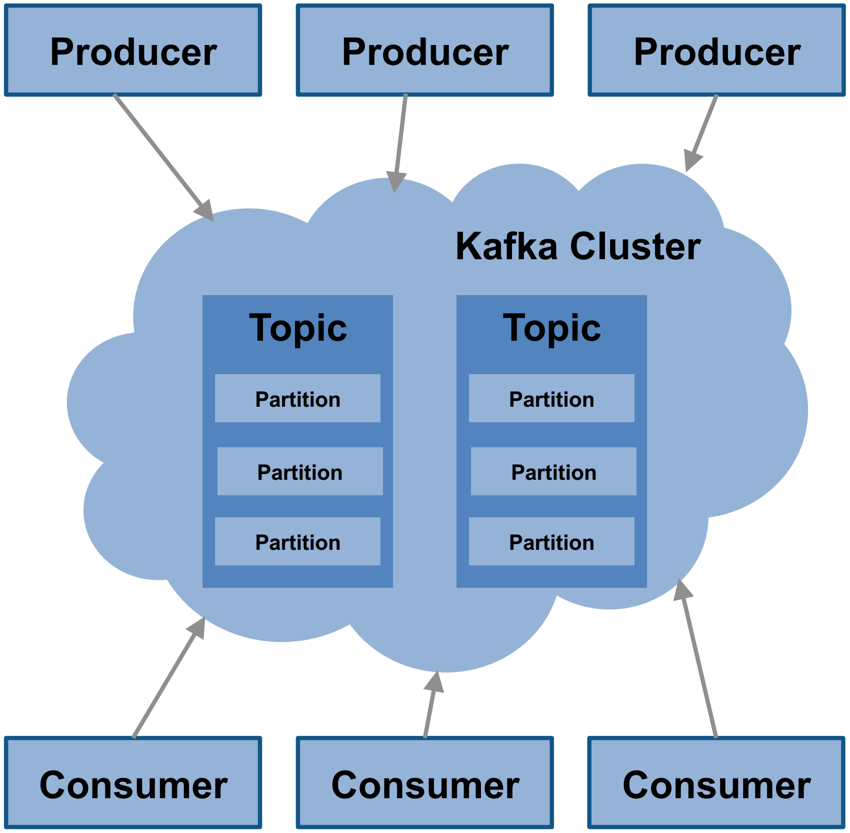
- Server software
  - Manages packaged MATLAB programs and worker pool
- MATLAB Runtime libraries
  - Single server can use runtimes from different releases
- RESTful JSON interface
- Lightweight client libraries
  - C/C++, .NET, Python, and Java



Introducing Apache Kafka

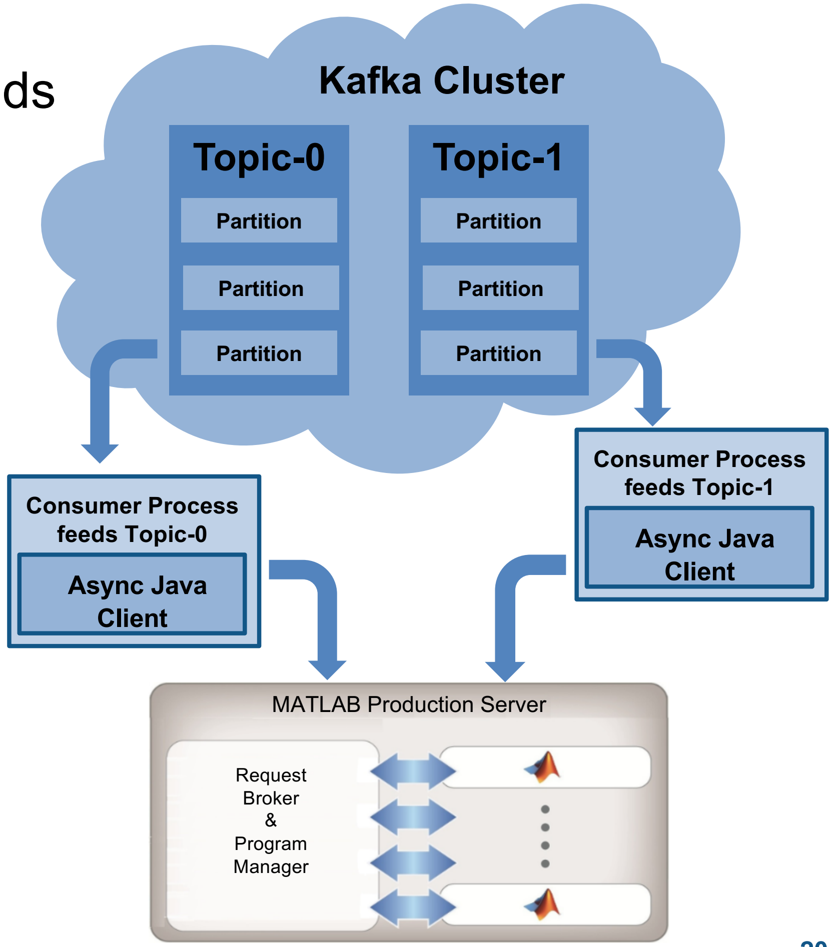
APACHE kafka®
A distributed streaming platform

- Kafka is a high through-put distributed messaging system
- Originally developed at LinkedIn and open sourced in 2011
- Kafka is architected as a massively scalable publish/subscribe message queue
- Well suited for large scale streaming applications

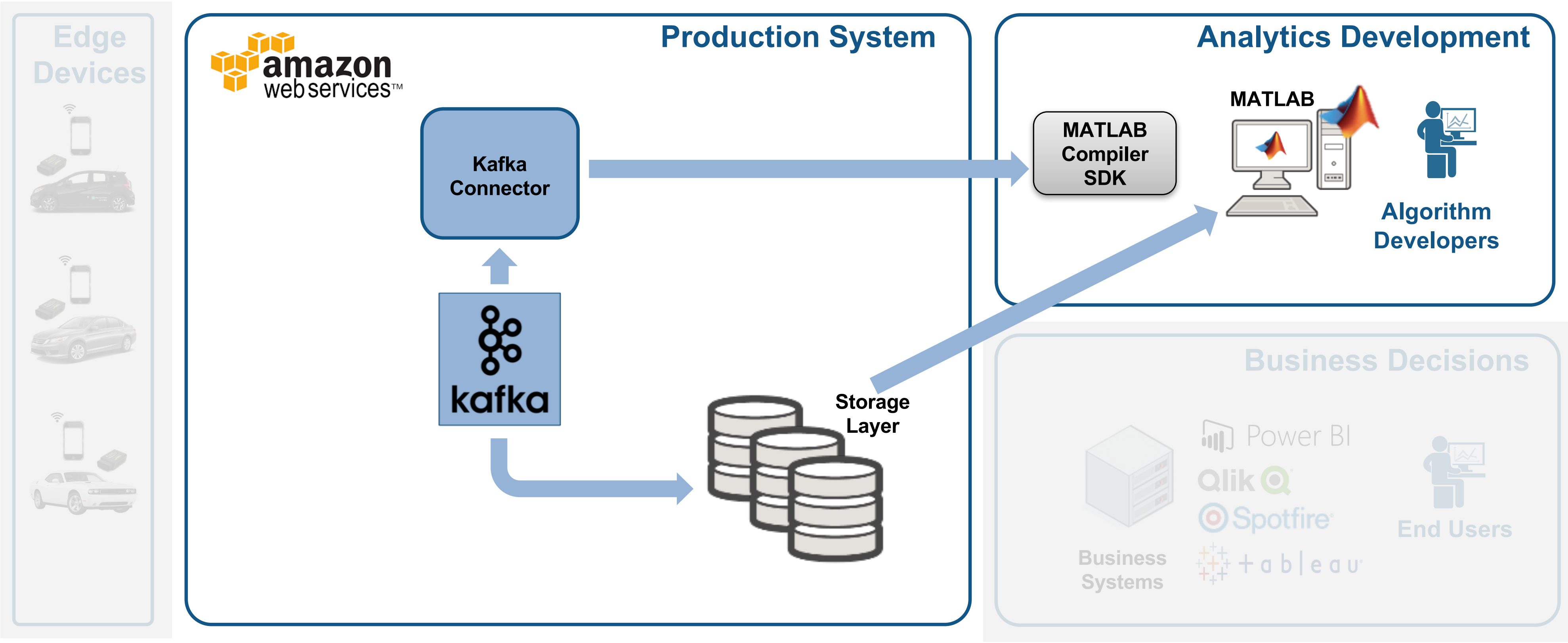


Connecting MATLAB Production Server to Apache Kafka

- Kafka client for MATLAB Production Server feeds topics to functions deployed on the server
- Configurable batch of messages passed as a MATLAB Timetable
- Each consumer process feeds one topic to a specified function
- Drive everything from a simple config file
  - No programming outside of MATLAB!



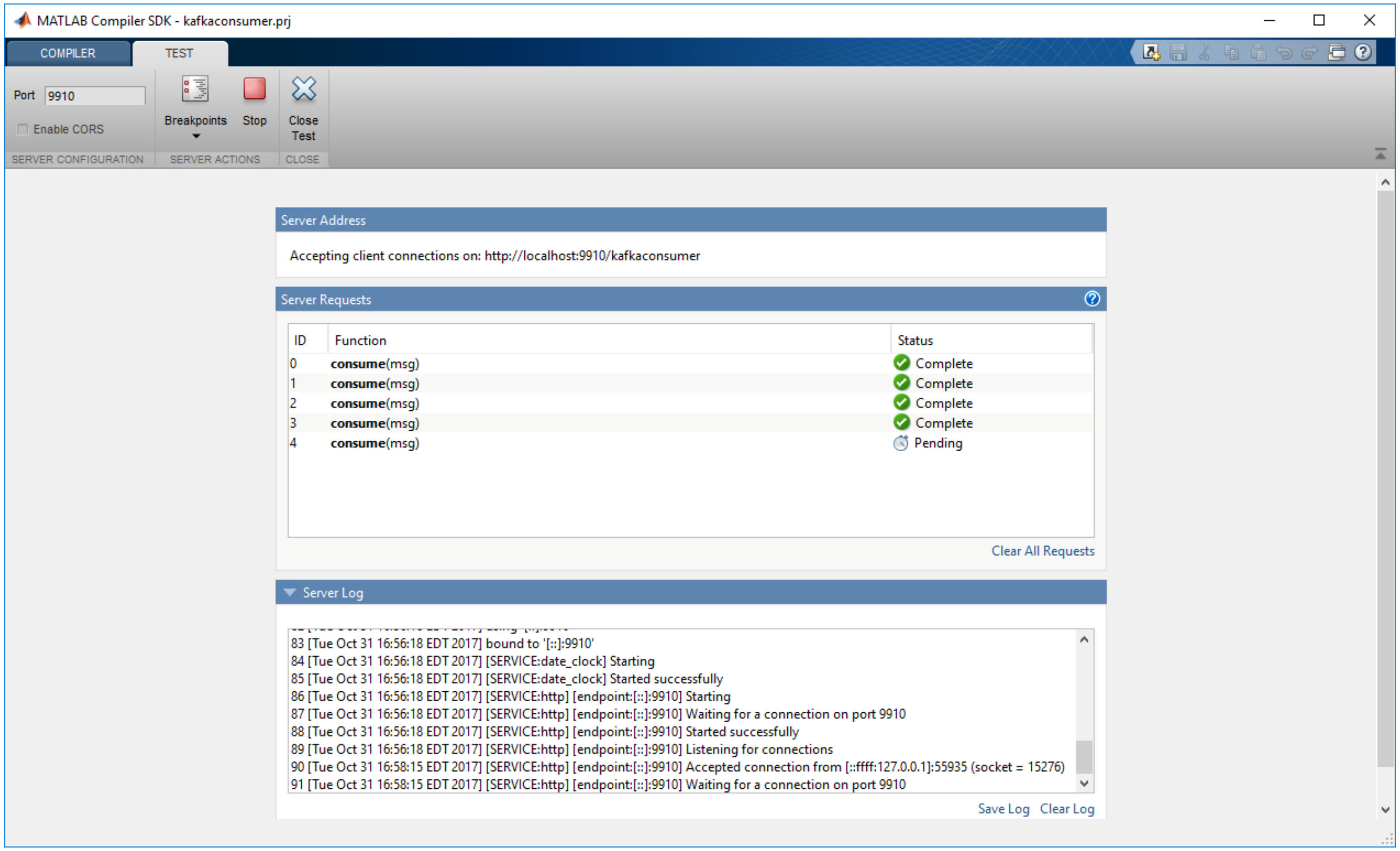
Develop, Test, and Deploy a Stream Processing Function



Develop a Stream Processing Function in MATLAB

DEMO

Test Your Stream Processing Function on Live Data

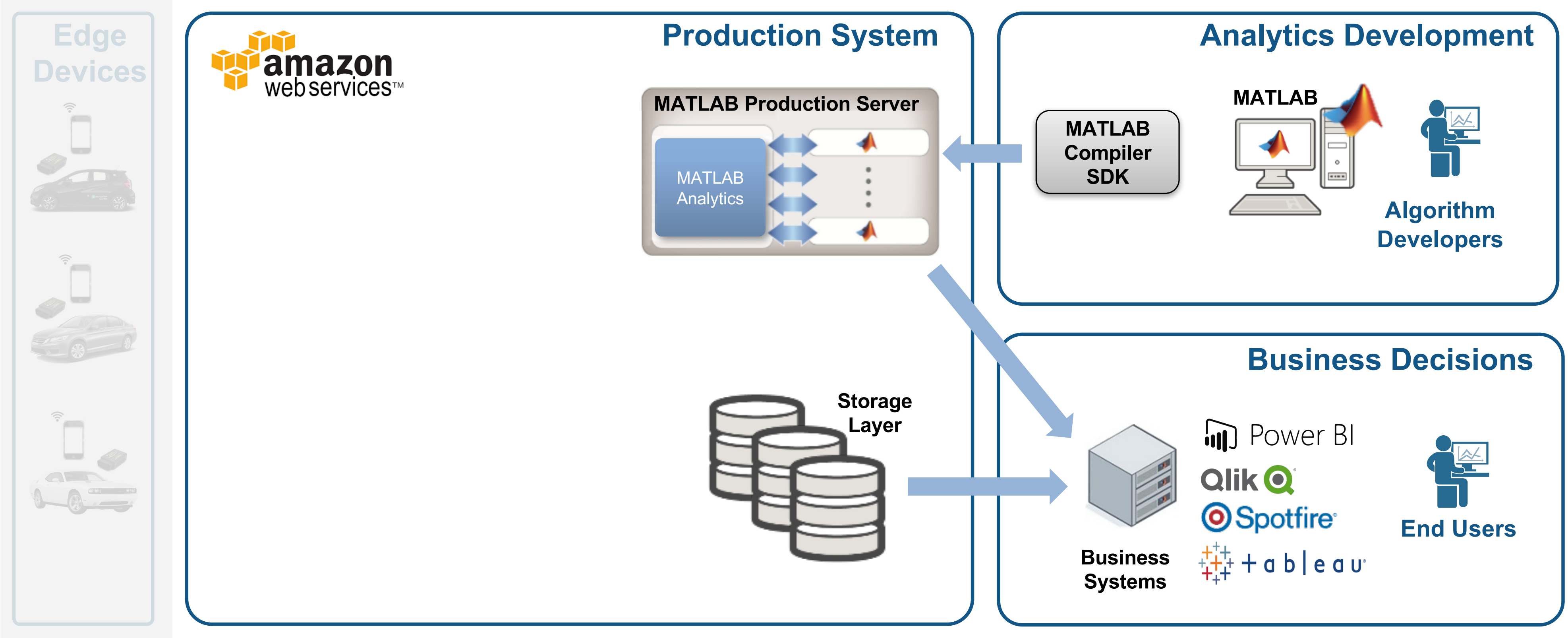


Accepting client connections on: http://localhost:9910/kafkaconsumer

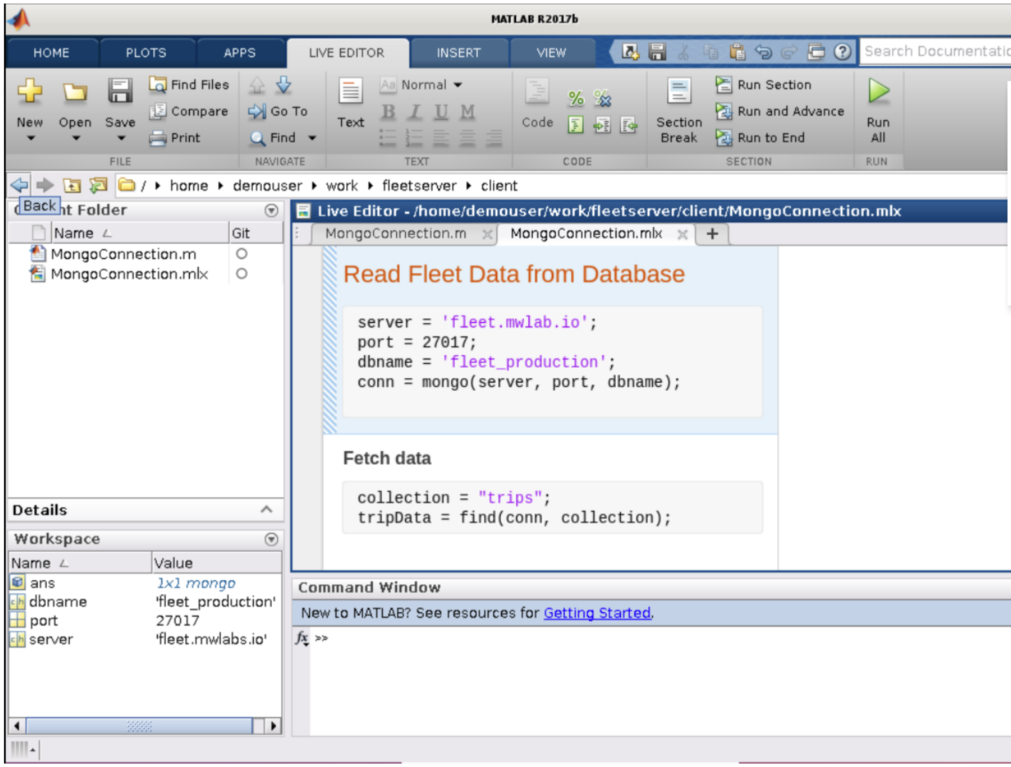
ID	Function	Status
0	**consume**(msg)	✓ Complete
1	**consume**(msg)	✓ Complete
2	**consume**(msg)	✓ Complete
3	**consume**(msg)	✓ Complete
4	**consume**(msg)	Pending

```
83 [Tue Oct 31 16:56:18 EDT 2017] bound to '[::]:9910'
84 [Tue Oct 31 16:56:18 EDT 2017] [SERVICE:date_clock] Starting
85 [Tue Oct 31 16:56:18 EDT 2017] [SERVICE:date_clock] Started successfully
86 [Tue Oct 31 16:56:18 EDT 2017] [SERVICE:http] [endpoint:[::]:9910] Starting
87 [Tue Oct 31 16:56:18 EDT 2017] [SERVICE:http] [endpoint:[::]:9910] Waiting for a connection on port 9910
88 [Tue Oct 31 16:56:18 EDT 2017] [SERVICE:http] [endpoint:[::]:9910] Started successfully
89 [Tue Oct 31 16:56:18 EDT 2017] [SERVICE:http] [endpoint:[::]:9910] Listening for connections
90 [Tue Oct 31 16:58:15 EDT 2017] [SERVICE:http] [endpoint:[::]:9910] Accepted connection from [::ffff:127.0.0.1]:55935 (socket = 15276)
91 [Tue Oct 31 16:58:15 EDT 2017] [SERVICE:http] [endpoint:[::]:9910] Waiting for a connection on port 9910
```

Complete Your Application



Complete Your Application



Name	Git
MongoConnection.m	○
MongoConnection.mlx	○

Live Editor - /home/demouser/work/fleetserver/client/MongoConnection.mlx

MongoConnection.m × MongoConnection.mlx ×

Read Fleet Data from Database

```
server = 'fleet.mwlab.io';
port = 27017;
dbname = 'fleet_production';
conn = mongo(server, port, dbname);
```

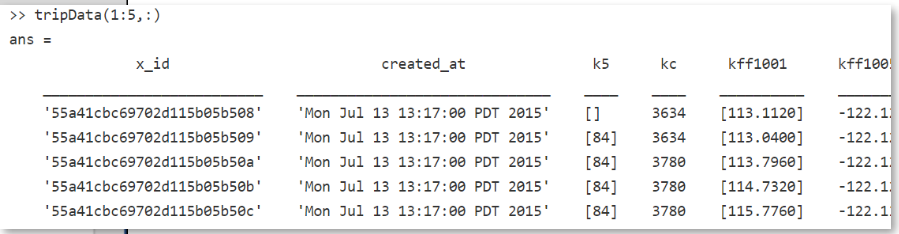
Fetch data

```
collection = "trips";
tripData = find(conn, collection);
```

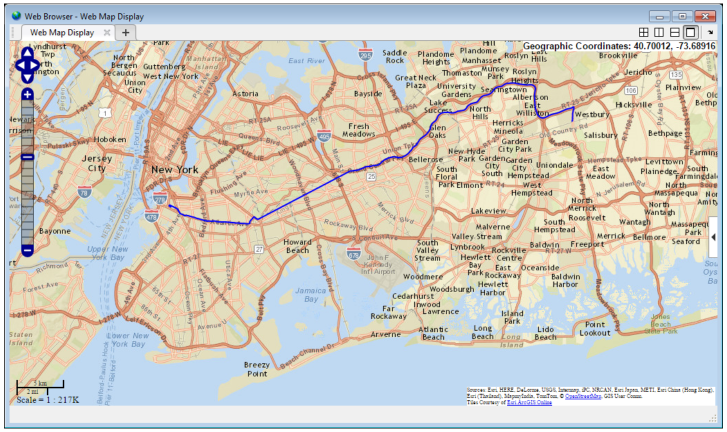
Workspace

Name	Value
ans	1x1 mongo
dbname	'fleet_production'
port	27017
server	'fleet.mwlab.io'

New to MATLAB? See resources for Getting Started.



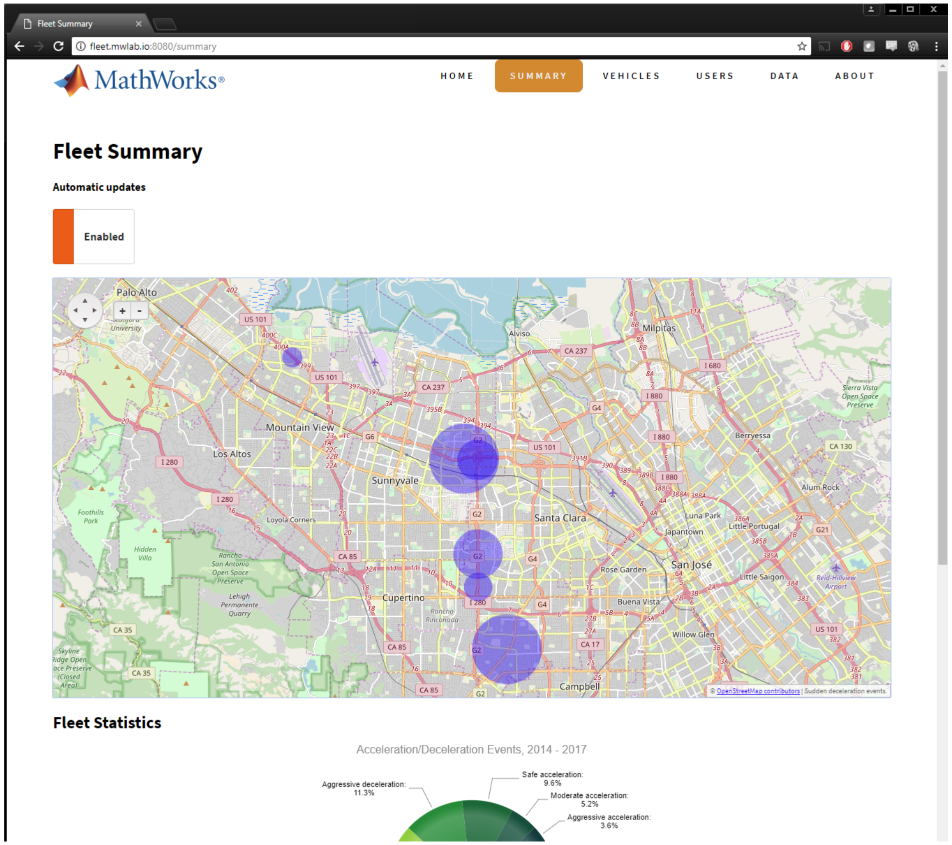
```
>> tripData(1:5,:)
ans =
              x_id                            created_at                k5      kc       kff1001      kff100
    ___________________________    ________________________________    ____    ____    __________    _______
    '55a41cbc69702d115b05b508'    'Mon Jul 13 13:17:00 PDT 2015'    []      3634    [113.1120]    -122.1
    '55a41cbc69702d115b05b509'    'Mon Jul 13 13:17:00 PDT 2015'    [84]    3634    [113.0400]    -122.1
    '55a41cbc69702d115b05b50a'    'Mon Jul 13 13:17:00 PDT 2015'    [84]    3780    [113.7960]    -122.1
    '55a41cbc69702d115b05b50b'    'Mon Jul 13 13:17:00 PDT 2015'    [84]    3780    [114.7320]    -122.1
    '55a41cbc69702d115b05b50c'    'Mon Jul 13 13:17:00 PDT 2015'    [84]    3780    [115.7760]    -122.1
```



Web Browser - Web Map Display
Web Map Display
Geographic Coordinates: 40.70012, -73.68916
New York
Jersey City
Hoboken
Bayonne
Astoria
Scale = 1 : 217K

Reference: https://www.mathworks.com/help/database/ug/import-and-analyze-data-from-mongodb.html

Go Live!



Fleet Summary

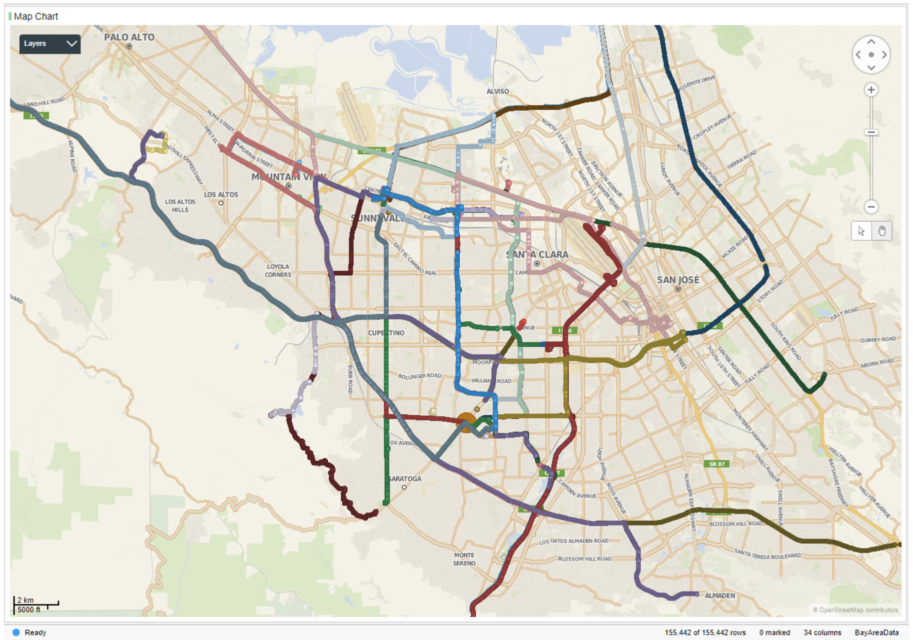
Automatic updates

Enabled

Fleet Statistics

Acceleration/Deceleration Events, 2014 - 2017

Aggressive deceleration: 11.3%
Safe acceleration: 9.6%
Moderate acceleration: 5.2%
Aggressive acceleration: 3.6%



Ready

155,442 of 155,442 rows 0 marked 34 columns BayAreaData

Key Takeaways

- MATLAB Connects directly to your data so you can quickly design and validate algorithms

- MATLAB's high-level language and apps enable fast design iterations

- MATLAB Production Server enables easy integration of your MATLAB algorithms with enterprise production systems

- This enables you to spend your time understanding the data and designing algorithms

Resources to learn and get started

- Data Analytics with MATLAB
- MATLAB Compiler SDK
- MATLAB Production Server
- Database Toolbox
  - https://www.mathworks.com/help/database/ug/import-and-analyze-data-from-mongodb.html

MathWorks®