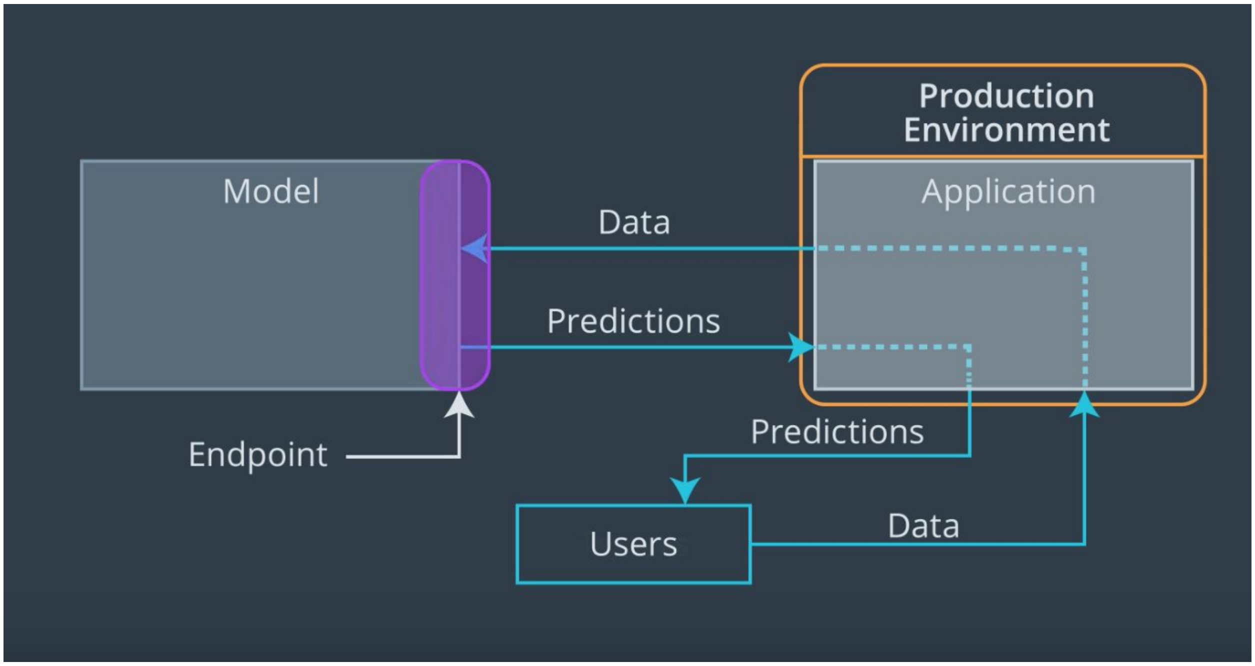
Creating and Using Endpoints

You've just learned a lot about how to use SageMaker to deploy a model and perform inference on some data. Now is a good time to review some of the key steps that we've covered. You have experience processing data and creating estimators/models, so I'll focus on what you've learned about endpoints.

An endpoint, in this case, is a URL that allows an application and a model to speak to one another.



Endpoint steps

- You can start an endpoint by calling `.deploy()` on an estimator and passing in some information about the instance.

```
xgb_predictor = xgb.deploy(initial_instance_count = 1, instance_type
= 'ml.m4.xlarge')
```

- Then, you need to tell your endpoint, what type of data it expects to see as input (like .csv).

```
from sagemaker.predictor import csv_serializer

xgb_predictor.content_type = 'text/csv'
xgb_predictor.serializer = csv_serializer
```

- Then, perform inference; you can pass some data as the "Body" of a message, to an endpoint and get a response back!

```
response = runtime.invoke_endpoint(EndpointName =
xgb_predictor.endpoint,    # The name of the endpoint we created
                                       ContentType = 'text/csv',
# The data format that is expected
                                       Body = ','.join([str(val) for
val in test_bow]).encode('utf-8'))
```

The inference data is stored in the "Body" of the response, and can be retrieved:

```
response = response['Body'].read().decode('utf-8')
print(response)
```

-

Finally, do not forget to **shut down your endpoint** when you are done using it.

```
xgb_predictor.delete_endpoint()
```

-

NEXT