Implementing Microservices and Message Queueing

Stephen Haunts
DEVELOPER, LEADER, AUTHOR AND TRAINER

@stephenhaunts www.stephenhaunts.com

Overview

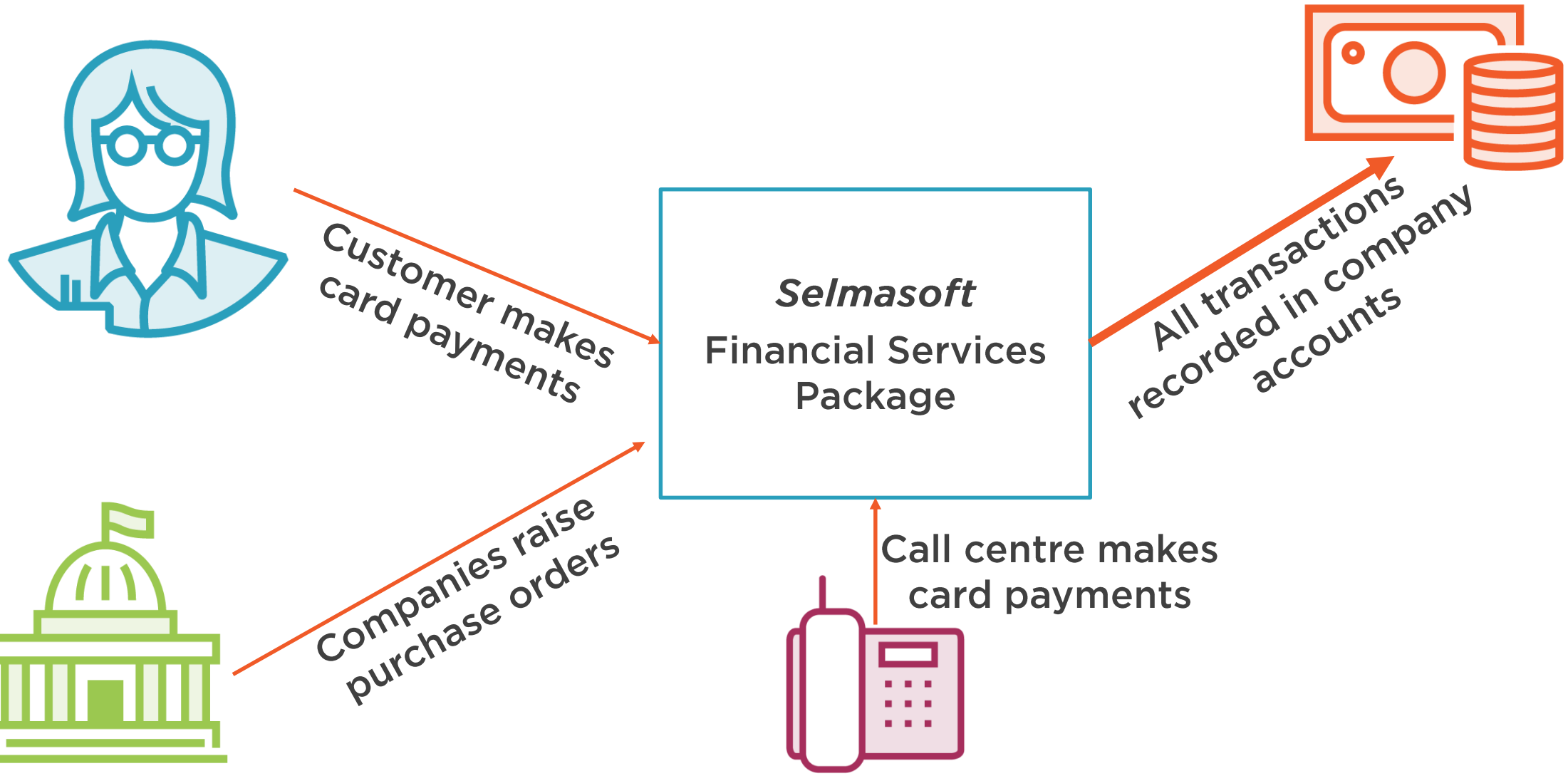
What does Selmasoft want to build?

High level design

Build supporting code

Demonstration of support code

What Do We Want to Build?



Technical Requirements

Asynchronous Payments

No blocking when making a payment for a customer

Synchronous Payments

Instant payments for the call centre

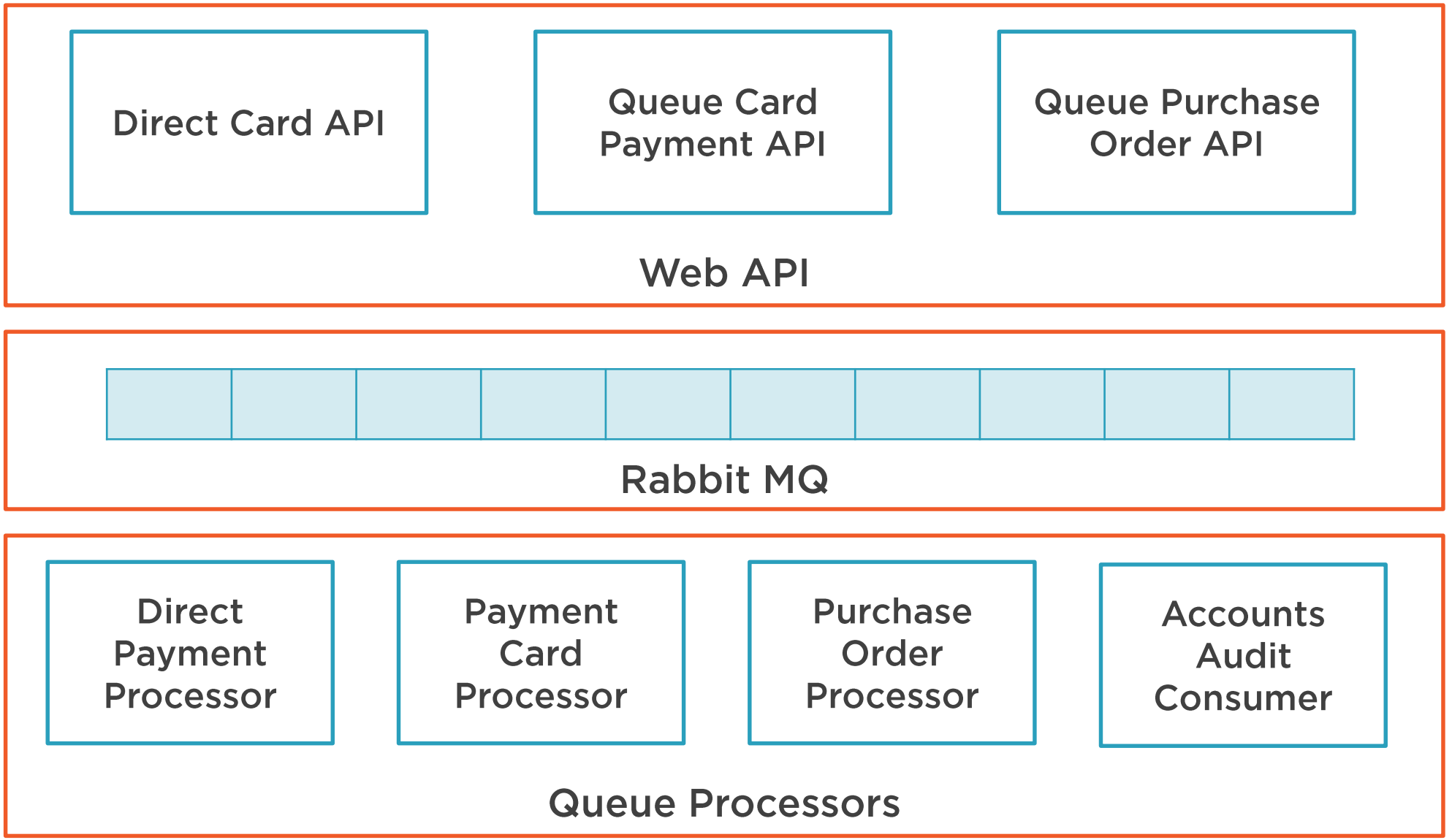
Queue Persistence

Messages persisted to disc for resiliency

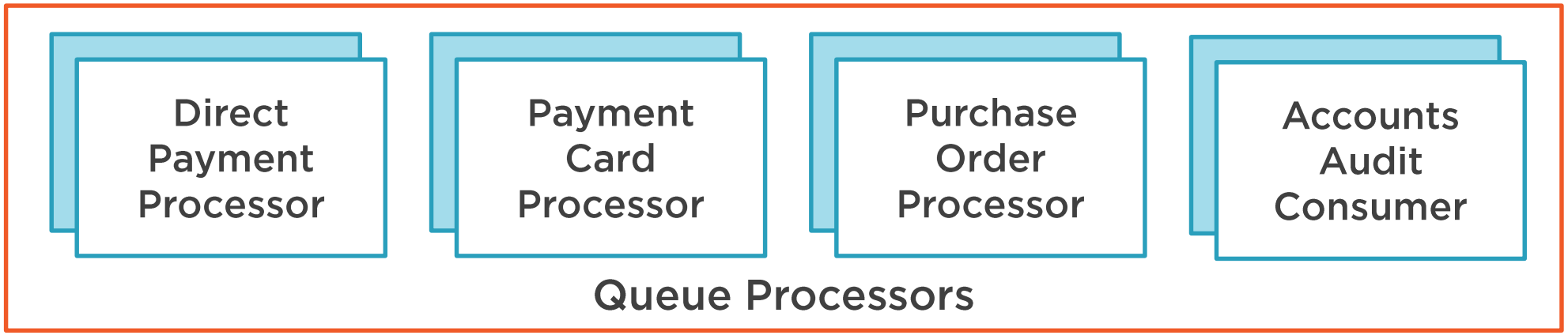
Scale Out Consumers

Easily scale consumers to cope with demand

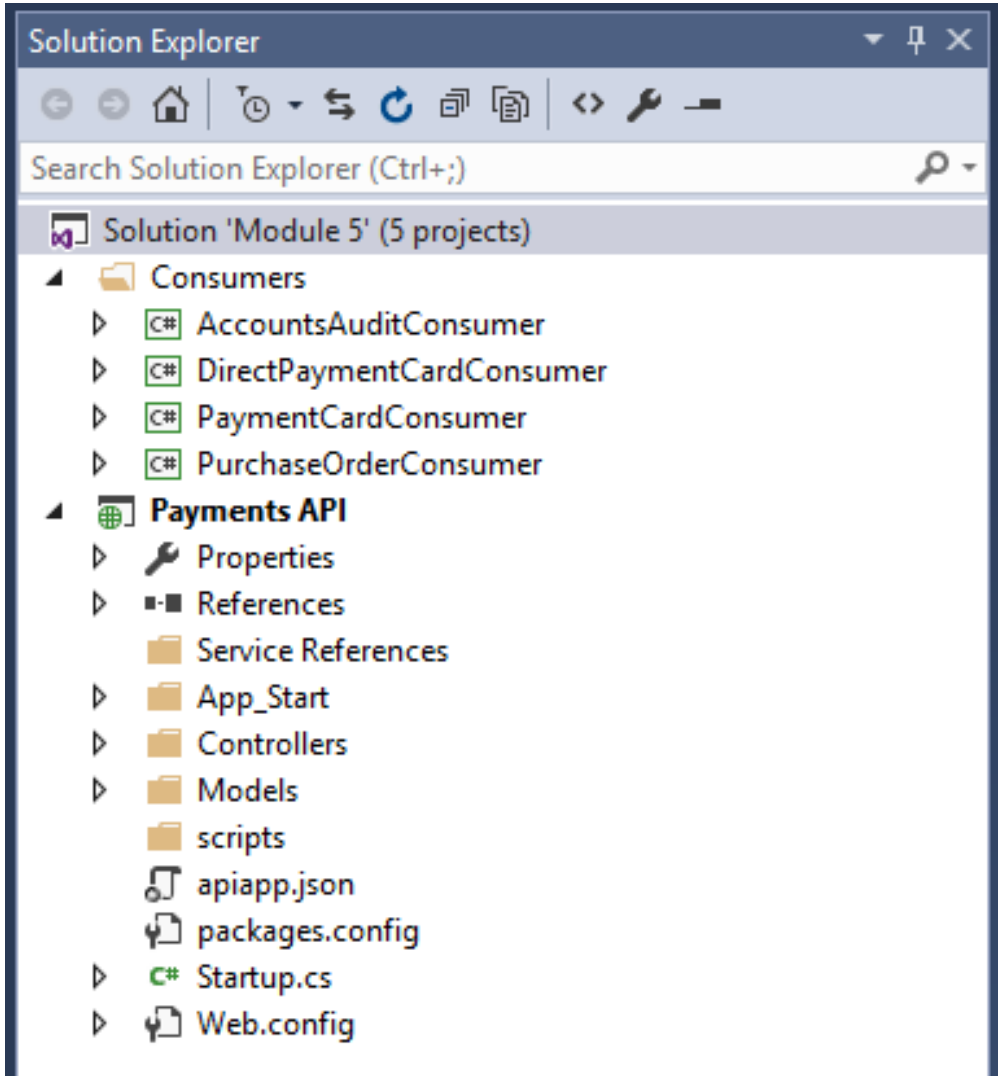
High Level Architecture



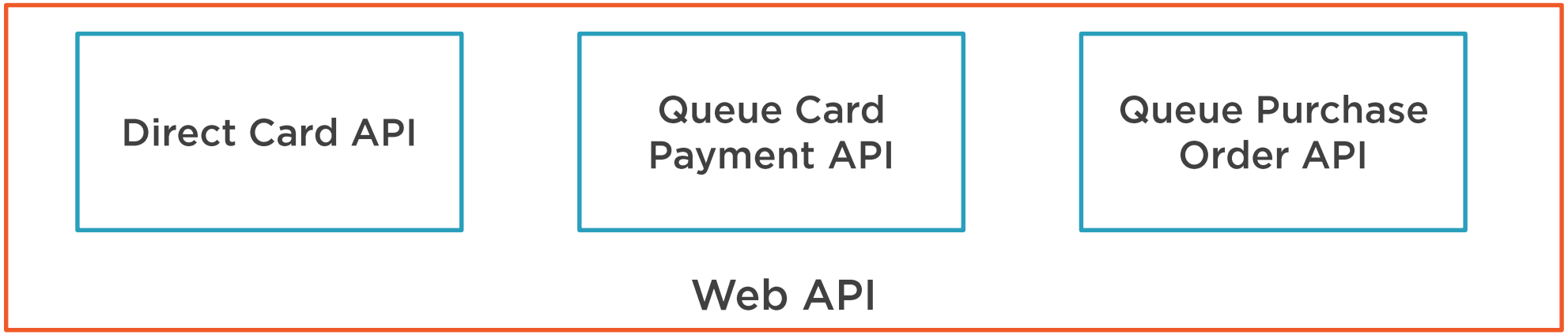
High Level Architecture



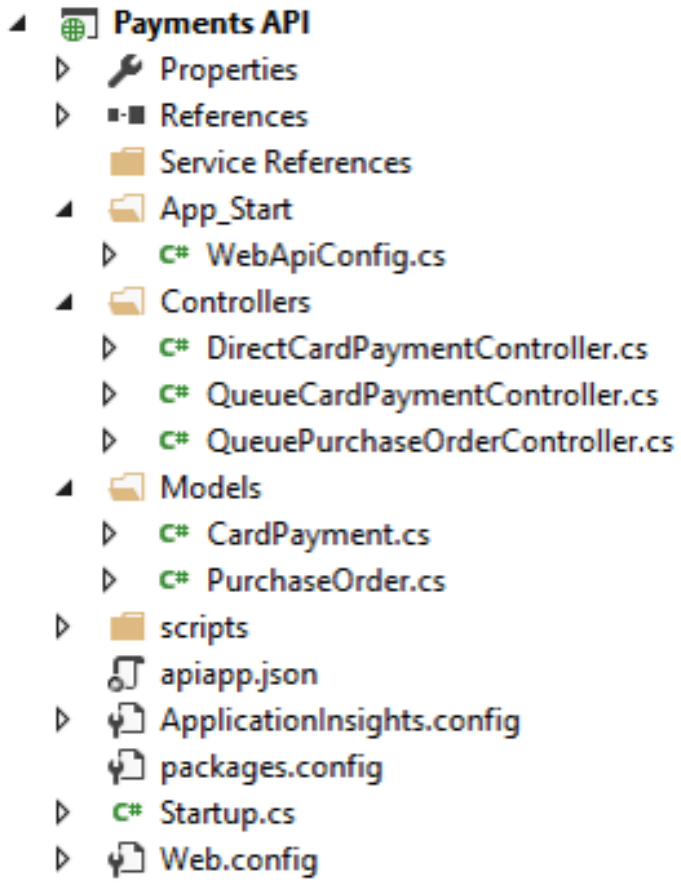
Exploring the Sample Code



Web API


Direct Card API
Queue Card Payment API
Queue Purchase Order API
Web API

Web API


Payments API
Properties
References
Service References
App_Start
WebApiConfig.cs
Controllers
DirectCardPaymentController.cs
QueueCardPaymentController.cs
QueuePurchaseOrderController.cs
Models
CardPayment.cs
PurchaseOrder.cs
scripts
apiapp.json
ApplicationInsights.config
packages.config
Startup.cs
Web.config

Web API

```
public class QueueCardPaymentController : ApiController
{
      [HttpPost]
      public IHttpActionResult MakePayment([FromBody]
                                           CardPayment payment)
    {
        return Ok(payment);
    }
}
```

Web API

```
public class QueueCardPaymentController : ApiController
{
      [HttpPost]
      public IHttpActionResult MakePayment([FromBody]
                                           CardPayment payment)
    {
        return Ok(payment);
    }
}
```