

# Case Study – Application Modernization

---

## Customer Problem

The Fabrikam Sports Association (FSA) is one of the largest sports franchises. They run a highly successful e-commerce website that sells merchandise to their legions of sports fans, and they have a huge passionate fan base. The current e-commerce site is built using ASP.NET and currently hosted in a co-location.

The e-commerce site hosts the shopping cart and the checkout process, and accepts payment by credit card. Owing to the high volume of transactions (tens of millions – about 50K per day) it is needed to ensure that they are PCI DSS Level 1 compliant. Fabrikam do not do the actual credit card processing but defer the credit card authorization and capture responsibilities to a third-party payment gateway, which is invoked using an API invoked over TLS (Transport Layer Security) from Fabrikam website server side logic. The API call includes the credit card holder data (name, number, and so on) and returns status indicating a success or failure in authorizing and capturing payment against the credit card. It is called after the customer clicks checkout, as a part of processing the order.

They have observed that Azure has received PCI compliance certification, and are interested in moving their solution to Azure. Fabrikam has stated that it is not clear to them from the Azure Trust Center just how Azure enables their solution for PCI compliance, and is uncertain how to proceed with quarterly datacenter audits and penetration tests once they are running in Azure.

When a customer completes a purchase, that customer is emailed a receipt. This email contains the receipt in the body of the email, as well as a PDF attachment. The PDF receipts are retained by FSA so their support staff can always review the receipts sent to a given customer. They have about 10 gigabytes (GB) of PDFs that are currently stored on disks available to the web server via a network share. They do not expect their receipt storage needs to double in less than 10 years. They currently store their customer and profile data in SQL Server 2014.

Along with the public facing e-commerce website, they have a backend website that supports their call center. Call center employees use this admin website to view customer orders. Customers can call in to the call center to place Qrders and pay for orders with their credit cards by phone. They are unsure that Azure can provide a solution that scales to meet their public demand, and is also secure for use by their call center and warehouse.

The FSA manages the order fulfillment process. When an order arrives, they store the order details in their SQL database, and also send a message for each order to their inventory management system running the warehouse. The FSA experiences a roughly 12-hour window that spans east to west coast business hours, during which they get most of their orders. The warehouse receives the message (which simply contains the order ID from the database), pulls up the order details identified in the message (by a lookup against the database), and then for each item in the order queues up a separate process to locate the item in

inventory or place an order for it with their supplier. Once this initial status for each item in the order is collected, the inventory status is updated in the database and a confirmation email is sent to the customer indicating the estimated delivery date of their completed order (and if any items are in backorder). This inventory lookup rarely takes more than a few hours and never more than a day.

They have reached a point where managing their server infrastructure is becoming a real challenge and are interested in understanding more about platform as a service (PaaS) solutions that could help them focus their efforts more on the core business value rather than infrastructure. "We're finding that with every upgrade, we're spending more and more engineering time on infrastructure and less on the experience that matters most to our fan base," says Chief Executive Officer (CEO) of Fabrikam Sports League, "we need to rebalance those efforts." As long as Fabrikam can retain their system's core functionality, they are open to new approaches for how it operates "under the covers."

According to their Chief Security Officer (CSO), one example of this is in how they manage the usernames and passwords for call center operators and support staff, as applied to the call center admin website. Today they have a homegrown solution that stores usernames and passwords in the same database used for storing merchandise information. They have experimented with other third-party solutions in the past, and their employees found it jarring to see another company's logo displayed when logging into their own call center website. In crafting their identity solution, they want to ensure they can brand the login screens with their own logo. Additionally, Fabrikam is concerned about hackers from foreign countries gaining access to the administrator site. Before they choose an identity solution, they would like to see how it indicates such attempts.

In addition to identity, Fabrikam would also like to see enhancements made possible in the migration to a PaaS solution. Director of Software Engineering, states that there is one particular feature asked of them frequently from their most loyal customers: order status alerts. Some customers, who are really fanatics, just cannot wait for an email confirmation. They want to know when their order has been processed (or if there has been a problem) via SMS. Fabrikam simply has not had the time to research how to implement this process, and is optimistic that it can be implemented by capitalizing on a service instead of homegrown code.

Finally, there is one architectural enhancement Fabrikam would like to make in the transition to a PaaS solution. When a visitor loads the home page, the page gets the list of featured products on offer (consisting of the product image, title, and URL) from the Offers service. It does this using a client-side GET request against an ASP.NET Web API 2 service that is executed as the page loads in the browser. Fabrikam anticipates growing the functionality of this service and would like to scale it independently of the website.

## Solution Outcome

Design and prepare to present a solution to the target customer audience. Follow the steps below:

### Step 1: Initial Planning

1. Identify the key stakeholders and the business goals from the customer's perspective. You could use the examples/formats for the same.

Customer Contacts	
	Name
Project Owner	
Technical Contact	
Business Decision Makers/Stakeholders	
Business Decision Makers/Stakeholders	
Business Decision Makers/Stakeholders	

#### Business Goals

- Business agility, faster time to market for business group
  - Lower Management Cost
  - Reduce Storage Cost
2. List the customer pain points, objections, other barriers for cloud, along with any identified gaps and the potential solution in Azure

### Step 2: Design and Architect Solution

Start by defining the High-level architecture without getting into the details. Envision your solution for handling the top-level requirements of the e-commerce website, call center website and inventory look up process.

Then start by addressing the following questions on the different processes and requirements:

#### Receipt management:

1. How would you propose the SLA store their receipts?
  - a. Assuming you are considering Azure Files, describe the ways in which Azure Files could be a good fit.
  - b. Instead assume you are considering Azure Blob Storage. Describe why the SLA might consider Blob Storage.
2. How would you handle migrating their historical receipts into each of the above options?
3. Considering the pros and cons of both options, ultimately which would you recommend to the customer?

4. How are existing receipts accessed by the e-commerce and call center websites? Be specific on the mechanisms used for locating the PDF associated with an order, how it is accessed, and how it is secured.
5. How would you propose that new PDFs of receipts be generated?

**Order fulfillment:**

1. How would you recommend the SLA manage the inventory lookup queues?
2. How would you help SLA decide between Azure Queues and Service Bus? Be sure to consider details implied by the SLA's requirements such as volume, message lifetime, and sizing. Explain the details of any computations you make.

**Offers Service:**

1. How would you propose Contoso meet their requirements for the Offers service?
2. What specific configurations would you need to make to support your proposed topology?

**Access Control:**

1. With respect to managing access to the call center website, explain how you would recommend Contoso implement a solution that meets their requirements. Be specific about both the implementation and the process you would use to gain acceptance of the proposed solution.

**Load testing:**

1. How would you recommend the SLA perform load testing on the e-commerce solution?
2. What tools should they use to conduct the load test and how would they configure and size them?
3. How could they monitor the solution during the load test?
4. How would you configure application performance monitoring and monitor the results? Would your solution identify issues with external resources (such as the database or payment gateway)? How?
5. Besides verifying they can handle the existing load, what other test pattern would you recommend?

**Automation:**

1. The infrastructure you have defined at this point is likely fairly complex. How would you recommend the SLA automate provisioning?
2. What steps would need to be taken to automate the provisioning in this way? Are there any Azure services you might have to provision in a different fashion?