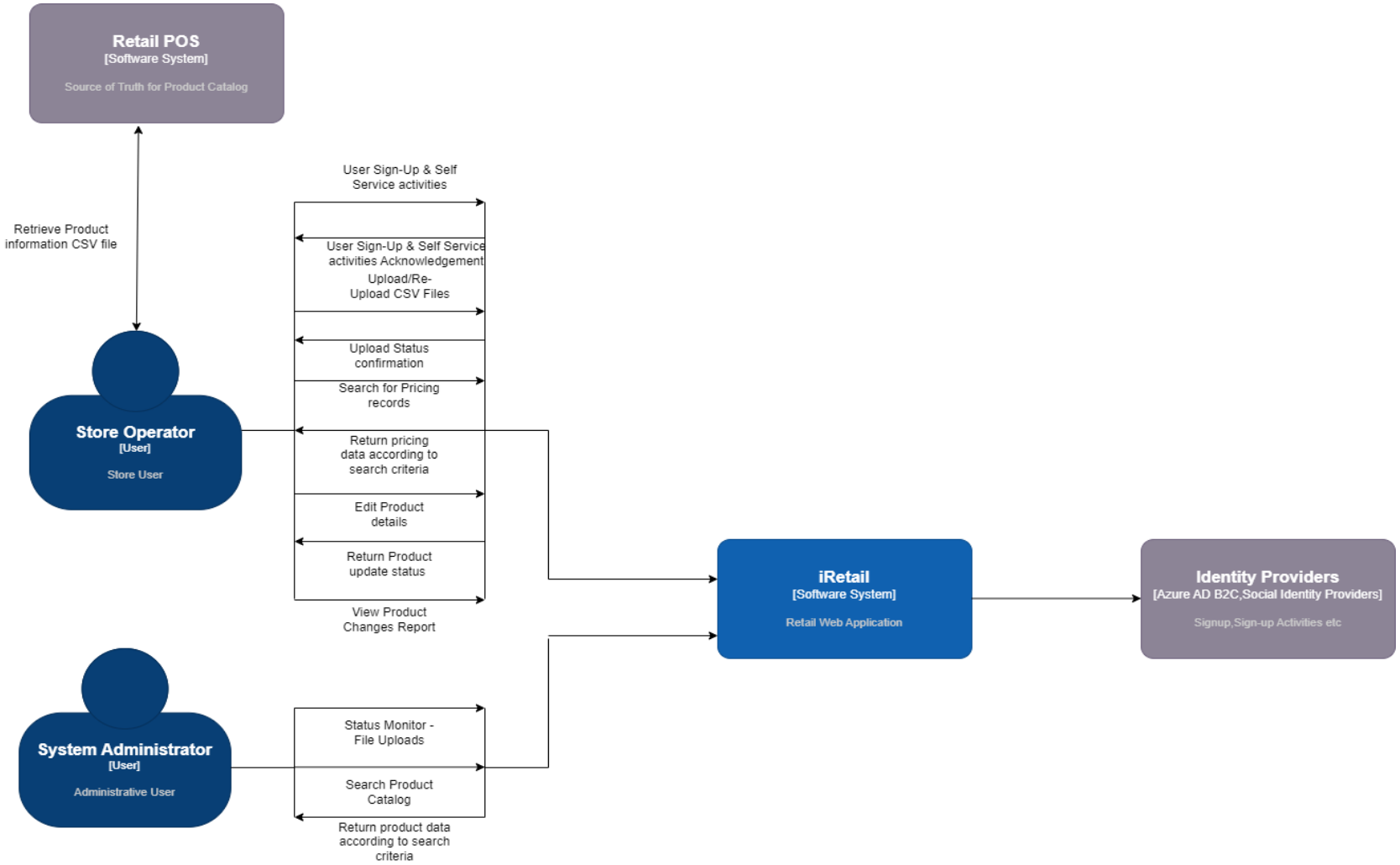
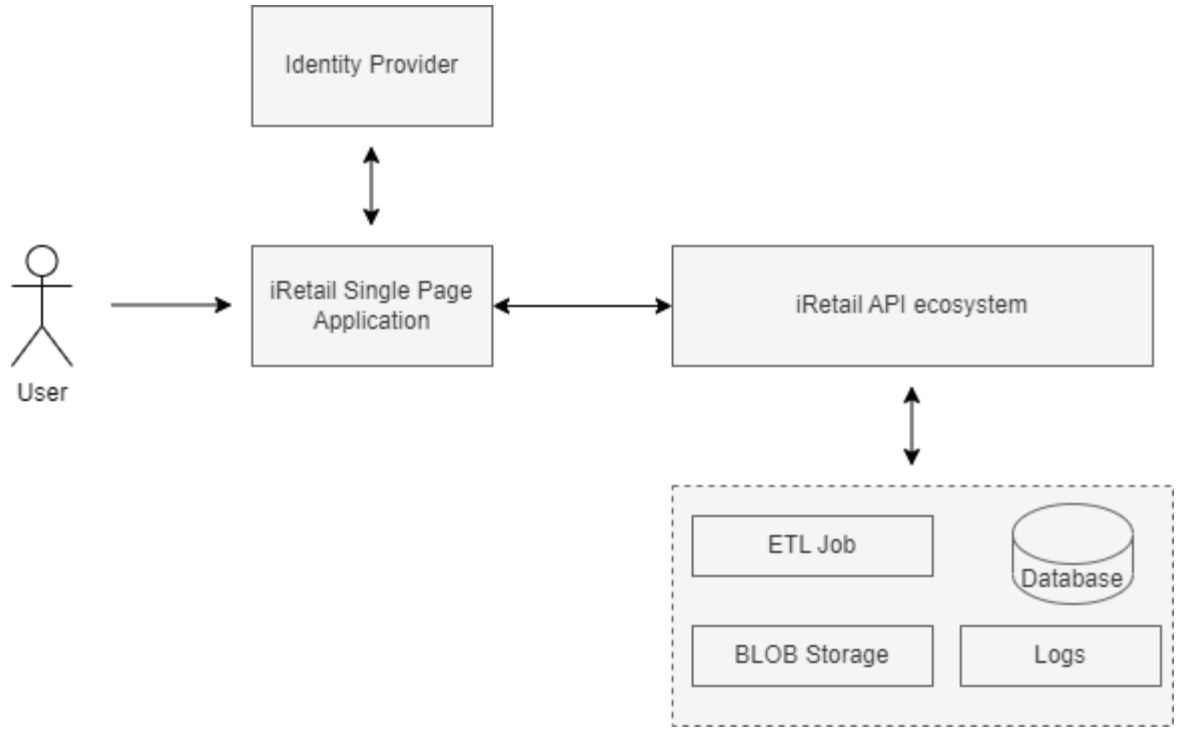
iRetail Web Application Architecture

Context Diagram

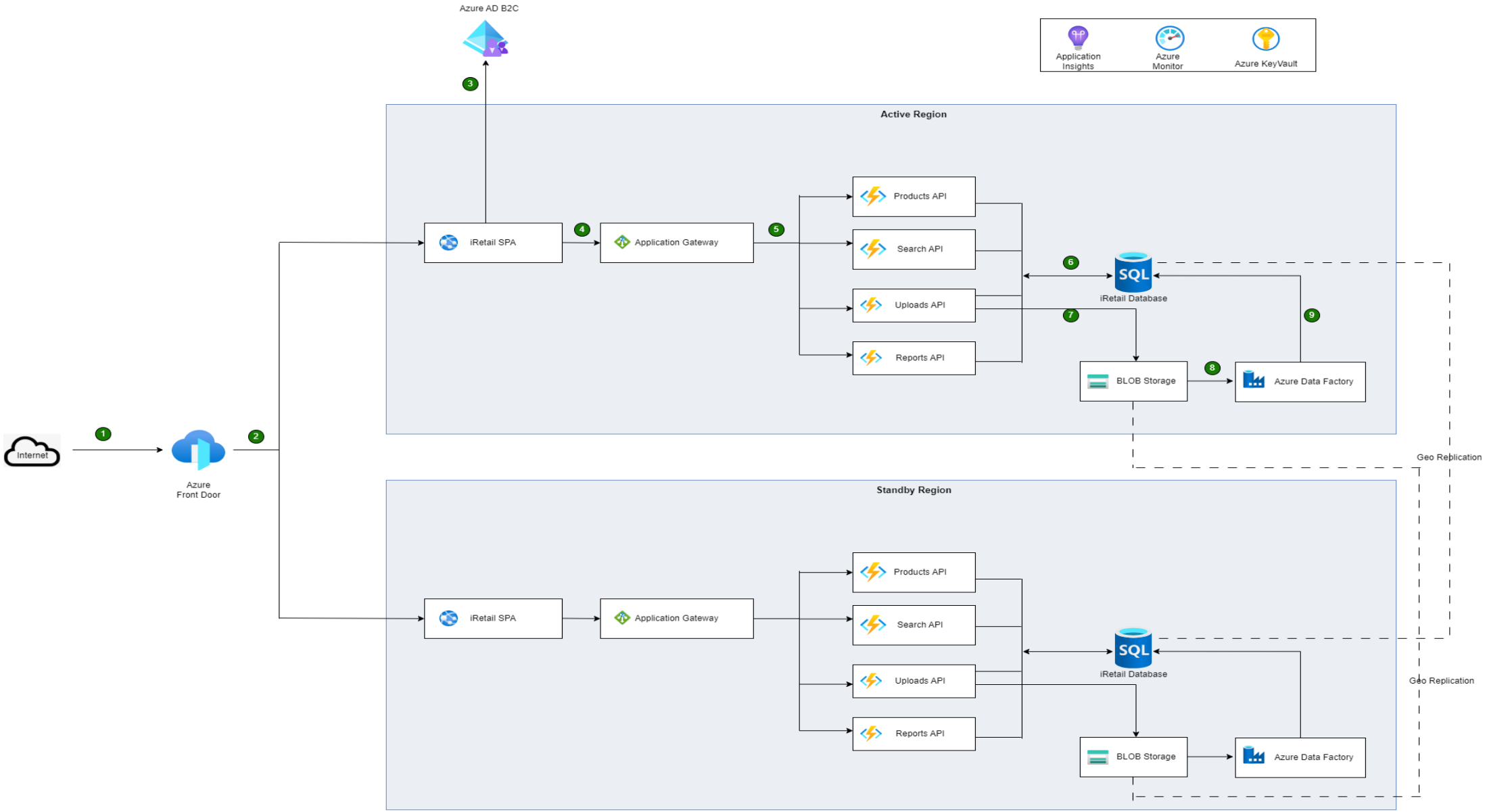


High Level Solution – Technical Composition



Component	Description
iRetail SPA	Web based Single Page Application
iRetail API	Backend Web API's to perform the required operations
ETL Job	Perform CSV Upload into the system
BLOB Storage	Maintains the CSV file
Logs	Exception,Audit,Informational log messages for the entire solution

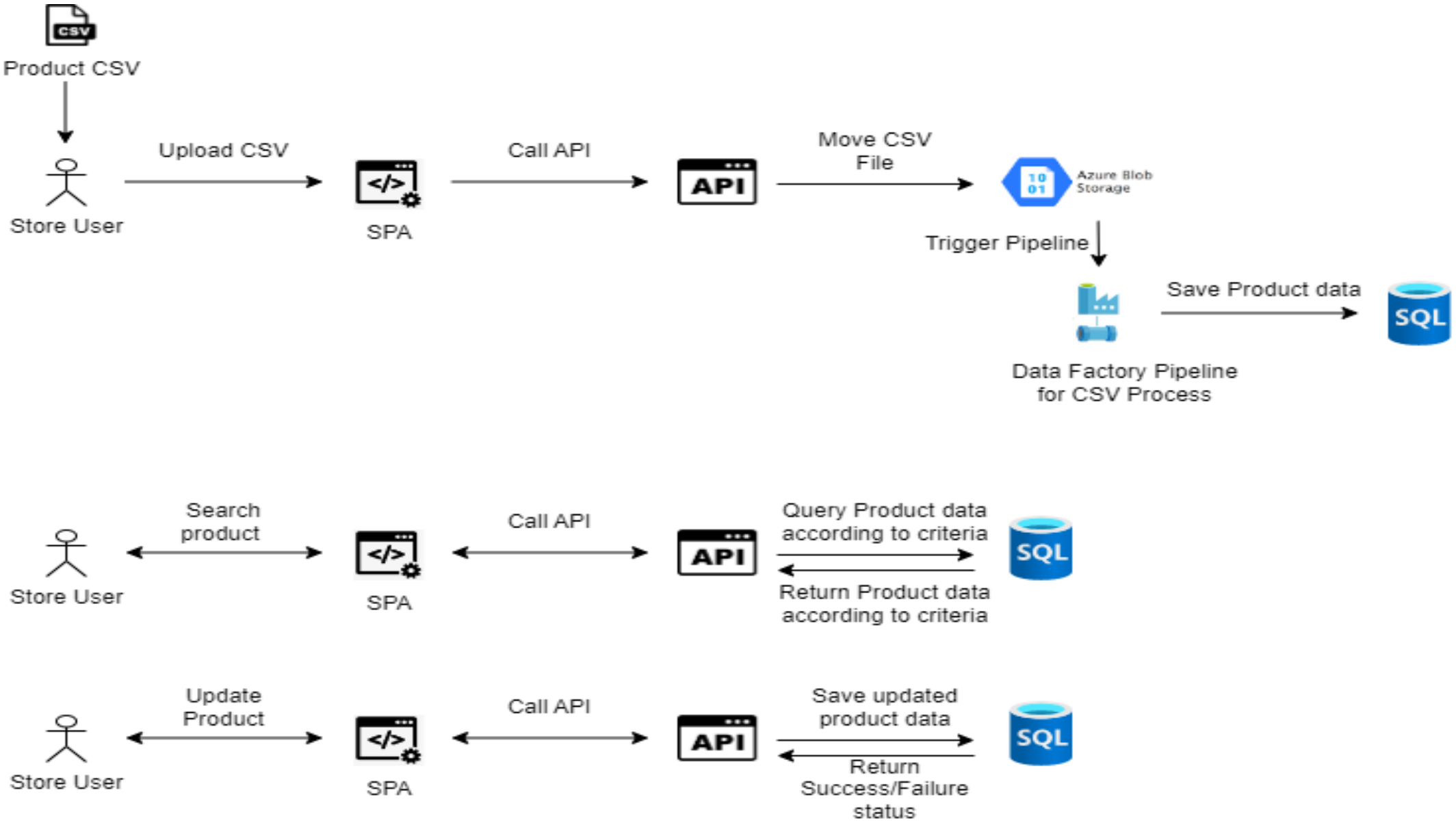
Technical Architecture



Technical Architecture – Flow of Steps involved

1. Requests from Internet reaches Azure FrontDoor
2. FrontDoor routes the request to the web app available in the primary region. If there is a failure in primary region, request will be routed to the standby region
3. iRetail SPA reaches to Azure AD B2C for Authentication and obtaining appropriate tokens/claims
4. Further SPA's calls to backend API's are routed to Application Gateway which knows which downstream API to redirect the calls to.
5. Azure Functions which are available as HTTP API's will be invoked after ensuring validating the bearer token available in the incoming request
6. The DB calls will be made by the API's will be handled by the Azure SQL DB and appropriate data will be returned if need be
7. File Handling will be handling using the Azure BLOB Storage service and files will be placed appropriate tiers(hot,cool etc)
8. The ETL process for CSV files will be triggered once the file reached BLOB and will be handled by the Azure Data Factory component
9. The Data Pipeline will update the data & status accordingly in the Azure SQL DB once the process is completed

Logical Flow along with use cases



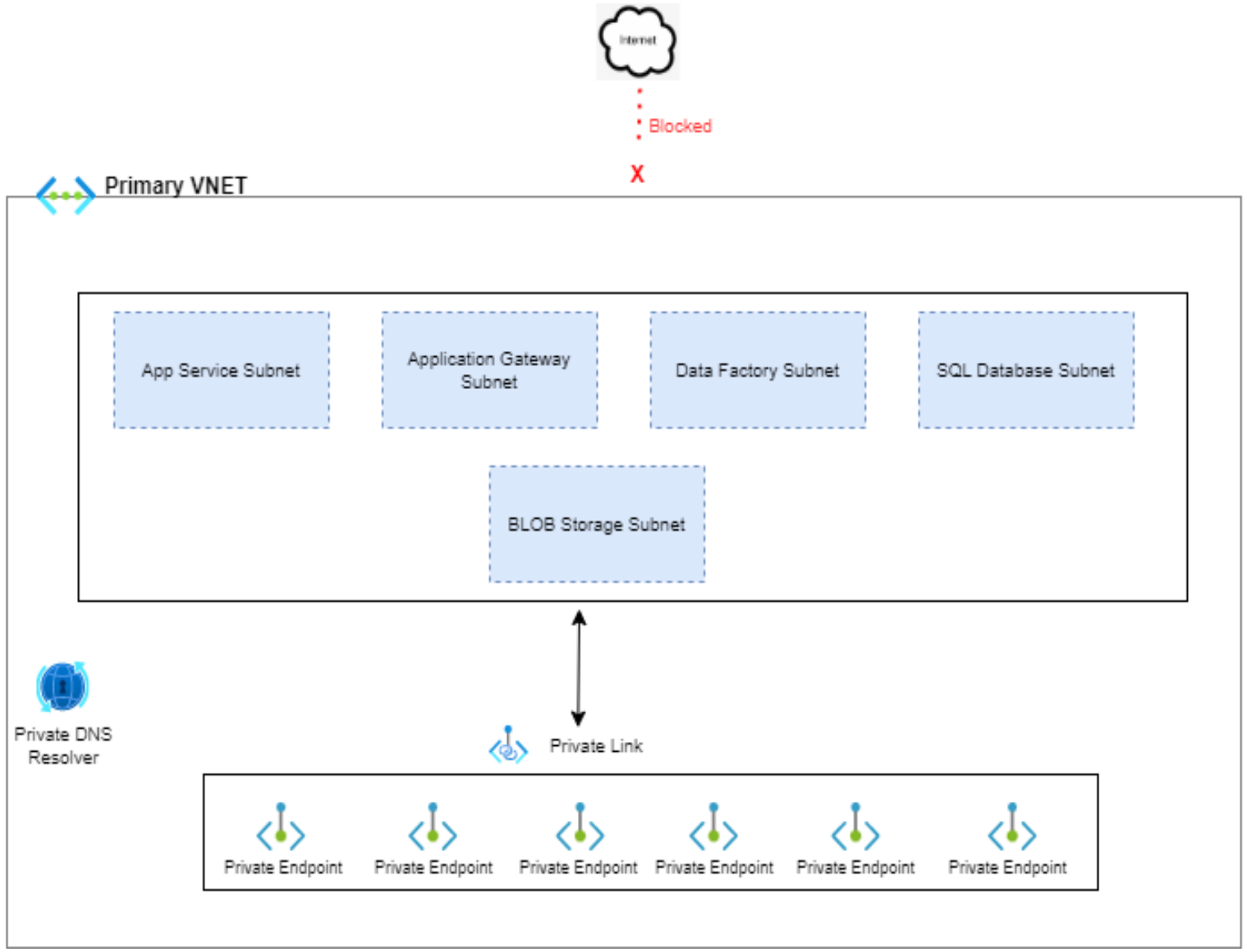
Design Decisions

- ✓ As a solution, its targeted to be a Microservices architecture and due to very few services it may not be a complete Microservices architecture at this moment. But this solution consists of required components to be evolve into a complete Microservices solution going forward
- ✓ The hosting platform will be completely on Microsoft Azure cloud.
- ✓ The solution will be hosted in two regions (Azure regions) comprising Primary and Standby region to ensure High Availability of the solution.
- ✓ Backend API's are designed considering functional aspects as well as Single Responsibility principle
- ✓ Backend API's are supposed to be Serverless since there will be not heavy business logic handled at the API level
- ✓ Azure FrontDoor: Since this solution need to be rolled out to multiple countries and considering the features like Web Application Firewall, Global Routing,CDN etc FrontDoor had been chosen.
- ✓ Azure App Service : This will be the hosting service for the iRetail SPA as there will be more options like OS,runtime, Scaling features etc can be utilized
- ✓ Azure Application Gateway: It validates the requests from FrontEnd App and routes them further as well as additional responsibility like Health Monitoring of backend API's, Termination of TLS etc

Design Decisions

- ✓ Azure Functions: It will be utilized with App Service Plan as the tier in consideration of scaling aspects
- ✓ Azure SQL: Since there is no unstructured data, RDBMS DB is selected to store the Master & Detail transactional data involved in the solution.
- ✓ Azure BLOB Storage: The incoming CSV files need to be processed and moved to a location where it can stay for a while. Considering the volume of the CSV files as well as Maintenance perspective, Azure BLOB Storage had been chosed
- ✓ Azure Data Factory: The CSV File upload process can be handled another API (Serverless function) but if there is any additional validation,transformation logic need to be handled it will be heavy on the service and lot of code need to be written. Considering the volume,global regions,processing time etc, Azure Data Factory had been chosen
- ✓ Azure AD B2C: To have an identity mechanism (AuthN & AuthZ), Azure AD B2C had been chosen since there are features like Signup/Signin, Self Service features like Password Reset etc, Custom Policies, Customized Login Page etc.
- ✓ As of now, there is no need to employ an “Hub & Spoke” Network Topology or any other approach to enable secured communication between components in Azure.

Secured Communication between Azure Components



S.No	Description
1.	Public Internet connectivity/access will be blocked
2.	All components will be placed in separate subnets
3.	Communication will happen with the help of Private Link Service & dedicated Private EndPoints
4.	Private DNS Resolver is the other significant part in the communication process

Primary Database Tables involved

Table Name	Description	Primary Attribute	Relationship attributes
SellerMaster	Holds the primary seller information	SellerID	N/A
SellerStores	Holds the store information of sellers	SellerStoreID	SellerID – Relationship with SellerMaster table
ProductMaster	Holds the product information of sellers	ProductID	SellerID – Relationship with SellerMaster table
ProductImages	Contains the Product Image URL's	ProductImageID	ProductID – Relationship with ProductMaster
ProductDetails	Contatins additional Product information	ProductDetailID	ProductID – Relationship with ProductMaster
ProductsUpload_Staging	Used for Products CSV ETL Process	StagingID	N/A

Proposed Technology Stack

Cloud Provider – Platform as a Service	FrontEnd App (SPA) Development
Azure	ReactJS
Database	**Backend Development**
Microsoft SQL Server	.NET Core
DevOps (CI/CD & IaC)	
Azure DevOps	Azure Biceps
sonarqube	

DevOps Approach

✓ Azure Infrastructure Provisioning & Maintenance:

- Azure PaaS service instances will be provisioned using Infrastructure as Code Approach (IaC)
- It will be easy & quick to rollout infrastructure to other region as well
- Azure Biceps will be utilized to create/modify infrastructure definitions

✓ Web Application & Azure Functions

- Both SPA & .NET Core based Azure functions will be deployed into Azure Platform using dedicated Build & Release pipelines (CI/CD).
- The Pipelines will be based on Pipeline as Code Approach (YAML files)
- Additional steps like Unit Tests validation, Static Code Analysis using SonarQube will be included in the Build pipeline

Non Functional Requirements

S.No	NFR Description	How this NFR had been Considered,Implemented
1.	Availability	At Solution Level: The solution will be deployed in two regions namely 1) Primary 2) Secondary/Standby with Azure FrontDoor covering the routing. Database: Database is replicated with geo replication strategy (Zone redundant replication)
2.	Scalability	App Service: Automatic scaling can be set by defining the auto scaling rules in App Service based on CPU Time etc Azure Functions: Since the azure functions are going to be based on “App Service” plan similar scaling rules & no. of instances can be defined
3.	Performance	CDN : Any static content can be server with the help of CDN Cache : Response caching will be implemented for appropriate use cases. Database: Appropriate indexing can be done to improve the data retrieval time

Non Functional Requirements (Continued)

S.No	NFR Description	How this NFR had been Considered,Implemented
4.	Security	Solution Level: Components will be grouped and placed accordingly into appropriate VNET/Subnets. Private Endpoints will be utilized to ensure communication between Azure Components. All the communication with open internet will be disabled using appropriate firewall rules User Level: Azure AD B2C based policies will be utilized to handling AuthN/AuthZ
5.	Resilient & Fault Tolerance	In the event of exceptions or any unexpected failures, it will be handled gracefully at the appropriate component of the solution. For Ex: If there is an failure while retrieving the status of CSV Upload, it will be propagated with appropriate status code by the API to the FrontEnd and the SPA will display a user friendly message. Retry Behaviour: Retry attempts will be implemented at all the layers to ensure the use case is met.

Non Functional Requirements (Continued)

S.No	NFR Description	How this NFR had been Considered,Implemented
6.	Maintainability	Each Azure component will be provisioned using Infrastructure As Code (IaC) Mechanism so it will be quick to add/update any configurations,provision new instances etc
7.	Reliability	As of now, the solution doesn't have any business hour limitation and its available 24/7/365. Failures at each component level will be handled and the solution will be back online appropriately after failures
8.	Auditability	All the operations related to a request are traced end-to-end (from FrontEnd App to BackEnd to Database) using ApplicationInsights logging. CorrelationID will be available to track a request end to end. Additionally any custom events,metrics can be tracked and logged into App Insights.

Non Functional Requirements (Continued)

S.No	NFR Description	How this NFR had been Considered,Implemented
		Change Data Capture (CDC) – DML changes for required tables can be audited at field level by enabling CDC feature in SQL Server. Temporal Tables – If the entire table needs to be audited, temporal table feature in SQL Server can be utilized.
9.	Accessibility	SPA will be designed/developed to cater to comply with disability regulations
10.	Compatibility	Browser Compatibility : The SPA will be compatible with prominent browsers like Chrome,Edge etc. Responsive Design : SPA will be designed in such a way so that the UI is viewable in any device, Desktop,Tab,Mobile etc

Assumptions

S.No	Description	Category
1.	Store user who ever uploads the data, they only can search & view the product data. Other users will not able to search for it.	Functional/Business
2.	The CSV File will have a size limit imposed so that it can be handled during upload	Technical
3.	Store Meta Data need to be maintained as a pre-requisite before uploading the CSV Files	Technical
4.	Once CSV Process is completed, the file will be moved to different tier in the BLOB Storage	Technical
5.	Any validation,transformation in the CSV file will be handled at the data pipeline level	Technical
6.	Region based deployment will be strategized and rollout can happen accordingly	Technical
7.	On successful registration, the users will receive email from Azure ADB2C.	Technical

Thank You