NAGP-Ecommerce Store

DAR Document

Nagarro Software Pvt. Ltd.

Prasad Bhalerao

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision History** | | | |
| Version | Date | Author/Contributor | Comments |
| 0.0.1 | 26/01/2025 | Prasad Bhalerao | Initial Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1 Introduction 4](#_Toc190015958)

[1.1 Objective and scope of document 4](#_Toc190015959)

[2 Requirements at a Glance 5](#_Toc190015960)

[3 Available tools 6](#_Toc190015961)

[3.1 Azure Front Door 6](#_Toc190015962)

[3.1.1 Features 6](#_Toc190015963)

[3.1.2 Pricing 6](#_Toc190015964)

[3.2 Azure CDN 6](#_Toc190015965)

[3.2.1 Features 6](#_Toc190015966)

[3.2.2 Pricing 7](#_Toc190015967)

[4 Comparison Analysis 7](#_Toc190015968)

[4.1 Weightage Matrix 8](#_Toc190015969)

[4.2 Comparison 8](#_Toc190015970)

[5 Recommendation 9](#_Toc190015971)

[6 Assumptions 10](#_Toc190015972)

[7 Risks 11](#_Toc190015973)

[8 Appendix 12](#_Toc190015974)

[8.1 References 12](#_Toc190015975)

# Introduction

Decision Analysis and Resolution (DAR) is critical to the development of any complex software system. DAR helps developers make informed decisions based on available options, constraints, risks, and trade-offs, ensuring the best possible course of action is taken.

## Objective and scope of document

* Objective:

The purpose of this document is to outline the importance of DAR in developing an ecommerce web application. It describes the key decisions that need to be made and the data that needs to be analyzed to inform decision-making.

* Scope:

This document is relevant to developers, project managers, and stakeholders involved in the development of the ecommerce web application.

* Key Decisions:

Key decisions that need to be made include:

* + Programming languages, frameworks, and libraries to use
  + Hosting infrastructure selection
  + Payment processing and security mechanisms to implement
  + Data Analysis:
* Data analysis is critical to inform decision-making. Relevant data to be analyzed includes:
  + User feedback
  + Market trends
  + Industry standards

# Requirements at a Glance

* Functional Requirements
* Product Catalog: Customers can browse and search products, view details, and add to cart.
* Shopping Cart: Customers can view cart contents, update quantities, and checkout.
* Checkout: Customers can enter shipping/payment information, review order, and complete purchase.
* Order Management: Customers can view order history, track status, and cancel/return items.
* Customer Account Management: Customers can create/manage account, update contact information, and change password.
* Product Management: Merchants can manage products, such as adding, updating, and managing inventory.
* Payment Gateway Integration: Web app is integrated with popular payment gateways to accept multiple forms of payment.
* Non Functional Requirments:
  + Performance: Web app handles high volume of requests with fast response times for customers.
  + Scalability: Web app able to handle increasing traffic and load as customer and order numbers grow.
  + High Availability: Web app always available to customers with minimal downtime.
  + Security: Web app protects customer data and transactions from unauthorized access and hacking attempts.
  + Compliance: Web app complies with relevant industry regulations and standards, e.g. PCI DSS for online payments.
  + Monitoring and Logging: Web app has monitoring and logging capabilities for tracking and troubleshooting.
  + Cost-efficiency: Web app is cost-efficient, utilizing cloud services and optimized resource usage.
  + Mobile Responsiveness: Web app is mobile responsive and accessible from any device.
  + User Experience: Web app has user-friendly interface, easy navigation, and provides a good user experience.

# Available tools

* Azure Front Door
* Azure CDN

## Azure Front Door

Azure Front Door functions as a global [HTTP(S)](https://www.geeksforgeeks.org/explain-working-of-https/) [load balancer](https://www.geeksforgeeks.org/load-balancing-in-cloud-computing/)and makes sure that traffic is equally spread to backends based on the defined[routing](https://www.geeksforgeeks.org/types-of-routing/). With the help of Azure Front Door, we can achieve access to a complete suite of abilities that surpass traditional content delivery. This makes it the ideal choice for applications that need developed traffic routing, load balancing, and enhanced security components.

### Features

* **Global Load Balancing**: Distributes traffic across multiple backend origins (e.g., web servers, app services) based on factors like user location, server health, and routing rules.
* **Smart Traffic Routing:** Makes sure users join to the most primary and reliable backend for an improved experience. [Routes traffic](https://www.geeksforgeeks.org/traffic-aware-routing/)established on aspects like latency, geography and priority.
* **Application Acceleration**: Optimizes the delivery of dynamic content and application logic.
* **SSL Offloading**: Terminates SSL connections at the edge, reducing the load on backend servers.
* **Web Application Firewall (WAF):** Protects applications from common web vulnerabilities and attacks.
* **Traffic Routing:** Allows for complex routing rules based on URL, headers, and other request attributes.

### Pricing

**Azure Front Door Standard** is content delivery optimized, offering both static and dynamic content acceleration, global load balancing, SSL offload, domain and certificate management, enhanced traffic analytics, and basic security capabilities.​

**Azure Front Door Premium** builds on capabilities of Azure Front Door Standard, and adds extensive security capabilities across WAF, BOT protection, Azure Private Link support, integration with Microsoft Threat Intelligence, and security analytics. WAF and Private Link pricing is included in Azure Front Door Premium.

Azure Front Door Standard/Premium billing is based on the following pricing dimensions:​

* Base Fees (i.e., fixed charge calculated on hourly basis)​
* Outbound Data Transfer from Edge to the Client
* Outbound Data Transfer from Edge to the Origin
* Requests incoming from client to Front Door's edge location
* Free data transfer from an origin in Azure data center to Front Door's edge location

## Azure CDN

Azure [Content Delivery Network (CDN)](https://www.geeksforgeeks.org/what-is-a-content-distribution-network-and-how-does-it-work/) delivering static content like images, videos, stylesheets, and scripts. It's all about giving users that high performance and super fast delivery. It can effortlessly spread content to users in a quick and efficient method. By keeping content in edge servers placed worldwide, users can easily access the content from a server that is geographically closest to them and that is how it improves the overall user experience because content delivered faster and low [latency.](https://www.geeksforgeeks.org/what-is-latency/)

Azure Content Delivery Network (CDN) is optimized for [caching](https://www.geeksforgeeks.org/caching-system-design-concept-for-beginners/)and spreading content closer to the end-users, resulting in quicker load times and decreases the server loads i.e. optimizing for caching and distribution and makes sure that faster load times and less strain on servers.

### Features

* Caching: Stores content closer to users at edge servers (Points of Presence or POPs).
* Global Reach: Distributes content across a vast network of POPs.
* Scalability: Handles high volumes of traffic and requests.
* Cost-Effective: Reduces bandwidth costs by serving content from cache.

### Pricing

Azure CDN billing is primarily based on the following pricing dimensions:

* **Outbound Data Transfers**: This is the most significant cost factor. You are charged for the data transferred from the CDN edge servers (where your content is cached) to the clients (users accessing your content). The cost per GB varies depending on the geographic location of the edge server serving the content (different regions are grouped into zones with different pricing).
* **Requests**: For certain CDN configurations (like Azure CDN from Microsoft classic), you might be charged for the number of requests made to the CDN. This is typically for features like the Rules Engine, where you can configure custom rules for how your content is delivered.

# Comparison Analysis

Azure CDN and Azure Front Door are both services offered by Microsoft Azure that help improve the performance and availability of web applications..

Key Differences:

|  |  |  |
| --- | --- | --- |
| Feature | Azure Front Door | Azure CDN |
| Primary Focus | Application delivery and load balancing | Static content delivery |
| Content Type | Static and dynamic | Static (images, videos, etc.) |
| Load Balancing | Advanced (global HTTP load balancing) | Basic (for static content) |
| Application Acceleration | Extensive | Limited |
| Security | WAF, DDoS protection, SSL offloading | Basic DDoS protection |
| Hourly Price | 130$ (500 GB Outbound, 100 GB Inbound, 2 Routing rules) | 40$ (500 GB Outbound) |

**When to Use Which**

* **Azure CDN:** Best for websites or applications with a lot of static content that needs to be delivered quickly to a global audience.
* **Azure Front Door:** Best for complex web applications that require global load balancing, application acceleration, and advanced security features.

**Can They Be Used Together?**

Yes, Azure CDN and Azure Front Door can be used together to create a comprehensive content delivery and application acceleration solution. Azure Front Door can act as the entry point for all traffic, routing requests to the appropriate backend origins, including Azure CDN endpoints for static content.

## Weightage Matrix

|  |  |
| --- | --- |
| Feature | Points |
| Primary Focus | 5 |
| Content Type | 5 |
| Load Balancing | 5 |
| Application Acceleration | 3 |
| Security | 3 |
| Pricing | 3 |

## Comparison

|  |  |  |
| --- | --- | --- |
| Feature | Azure Front Door | Azure CDN |
| Primary Focus | **5** | **2** |
| Content Type | **5** | **3** |
| Regional Load Balancing | **5** | **2** |
| Application Acceleration | **3** | **1** |
| Security | **3** | **1** |
| Pricing | **1** | **3** |

# Recommendation

For our application, we recommend Azure Front Door over Azure CDN primarily due to its advanced traffic routing capabilities. While CDN excels at static content delivery, our application requires intelligent routing based on factors like user location, backend health, and real-time performance. Front Door's smart routing features enable us to:

1. **Optimize performance**: Route users to the closest healthy backend, minimizing latency and improving user experience.
2. **Enhance availability**: Automatically failover to alternate backends if a region or server becomes unavailable.
3. **Implement complex traffic management**: Support A/B testing, canary deployments, and other advanced routing scenarios.
4. While Azure CDN is a strong option for caching and static content delivery, Front Door's intelligent routing is crucial for meeting our application's performance, availability, and traffic management requirements.

# Assumptions

1. **Global User Base**: Our e-commerce platform anticipates users worldwide, requiring low latency and high availability across geographical regions.
2. **Dynamic Content**: Beyond static assets, our application handles dynamic content like product catalogs, user accounts, shopping carts, and personalized recommendations, requiring intelligent routing to application servers.
3. **Multiple Backend Origins**: We plan to deploy our application across multiple regions/data centers for redundancy and scalability.
4. **Complex Traffic Management**: We foresee the need for A/B testing new features, canary deployments for releases, and potentially traffic shaping during peak seasons.

# Risks

* **Increased Complexity:** Implementing and managing Front Door introduces additional complexity compared to solely using CDN. This includes configuring routing rules, health probes, and managing multiple backend origins.
* **Cost:** Front Door has a base cost and usage-based charges. While the benefits might outweigh the cost, it's essential to carefully estimate and monitor expenses to ensure cost-effectiveness.
* **Configuration Errors:** Incorrectly configured routing rules or health probes can lead to traffic misdirection, service disruptions, or performance issues. Thorough testing and validation are crucial.
* **Dependency:** Relying on Front Door introduces a dependency on the service. Any issues or outages with Front Door can impact the availability of the entire application.
* **Learning Curve:** There is a learning curve associated with mastering Front Door's features and configurations. The team needs to be adequately trained or have access to expertise.
* **Integration Challenges:** Integrating Front Door with existing infrastructure and services might present challenges. Careful planning and coordination are required.

# Appendix

## References

1. Microsoft Azure Front Door documentation: https://docs.microsoft.com/en-us/azure/cosmos-db/
2. Getting started with Azure Front Door: <https://docs.microsoft.com/en-us/azure/cosmos-db/create-sql-api-dotnet>
3. Azure CDN Search Applications: https://www.elastic.co/enterprise-search/search-applications
4. Data modeling in Azure Front Door: https://docs.microsoft.com/en-us/azure/cosmos-db/modeling-data
5. Querying data in Azure Front Door: https://docs.microsoft.com/en-us/azure/cosmos-db/sql-query-getting-started
6. Performance tuning in Azure Front Door: https://docs.microsoft.com/en-us/azure/cosmos-db/performance-tips
7. Azure CDN as a primary database: [https://dev.to/er\_dward/Azure CDN-as-a-primary-database-15a5#:~:text=Azure CDN%20is%20not%20a%20transactional,It%20a%20good%20AP%20sytem](https://dev.to/er_dward/elasticsearch-as-a-primary-database-15a5#:~:text=Elasticsearch%20is%20not%20a%20transactional,It%20a%20good%20AP%20sytem).