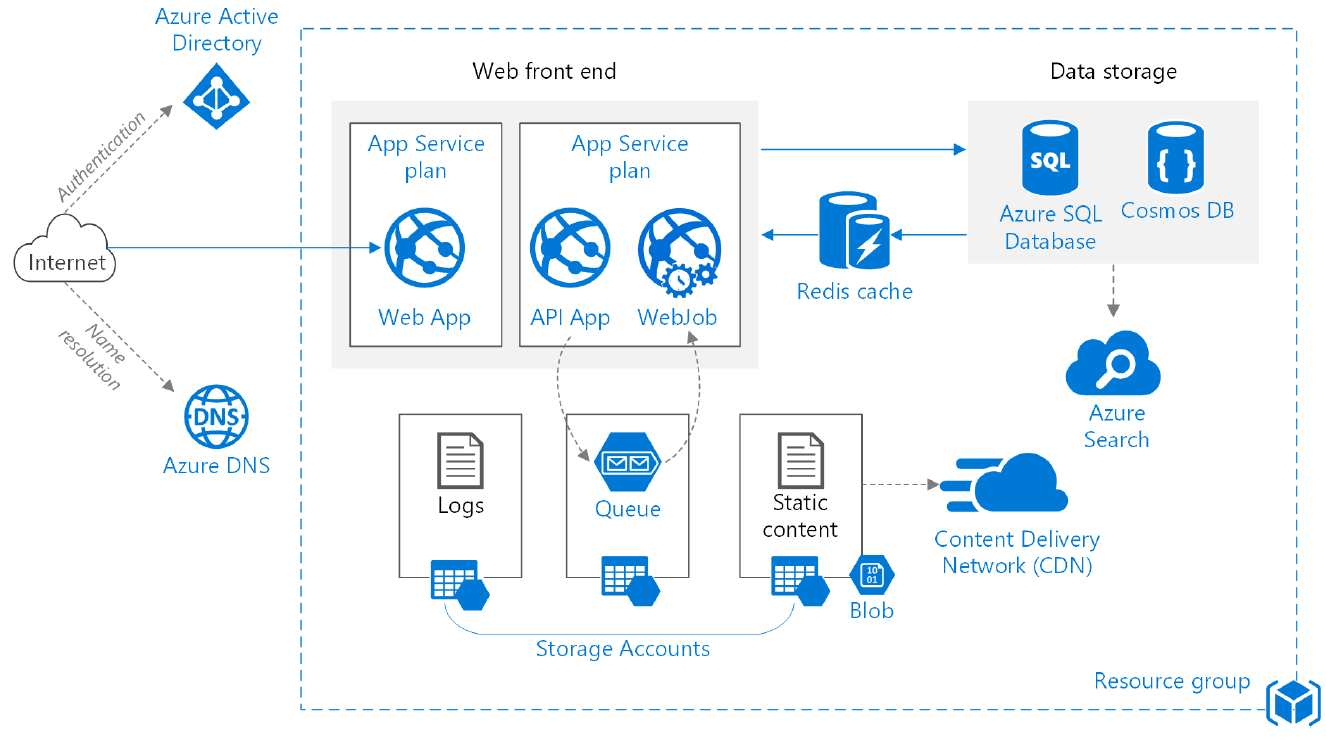
# Functional Architecture (self explanatory)

# Technical Architecture



* **Resource group**. A resource group is a logical container for Azure resources.
* **Web app** and **API app**. A typical modern application might include both a website and one or more RESTful web APIs. A web API might be consumed by browser clients through AJAX, by native client applications, or by server-side applications. For considerations on designing web APIs
* **WebJob**. Use WebJobs to run long-running tasks in the background. WebJobs can run on a schedule, continuously, or in response to a trigger, such as putting a message on a queue. A WebJob runs as a background process in the context of an App Service app.
* **Queue**. In the architecture shown here, the application queues background tasks by putting a message onto an Queue storage queue. The message triggers a function in the WebJob. Alternatively, you can use Service Bus queues.
* **Cache**. Store semi-static data in Redis Cache.
* **CDN**. Use Azure Content Delivery Network (CDN) to cache publicly available content for lower latency and faster delivery of content.
* **Data storage**. Use Azure SQL Database for relational data. For non-relational data, consider a NoSQL store, such as Cosmos DB.
* **Azure Search**. Use Search to add search functionality such as search suggestions, fuzzy search, and language-specific search. Azure Search is typically used in conjunction with another data store, especially if the primary data store requires strict consistency. In this approach, store authoritative data in the other data store and the search index in Azure Search. Azure Search can also be used to consolidate a single search index from multiple data stores.
* **Email/SMS**. Use a third-party service such as SendGrid or Twilio to send email or SMS messages instead of building this functionality directly into the application.
* **DNS**. DNS is a hosting service for DNS domains, providing name resolution using Microsoft Azure infrastructure.