# Azure Architecture Design Document

## 1. Overview of Architecture

This document outlines the architectural design for a production environment hosted in Azure. The environment is designed for three separate subscriptions to support EMEA, APAC, and AMER regions, each with a dedicated setup for data residency and cost optimization. The architecture includes Azure Front Door (AFD), API Management (APIM), App Service Environment v3 (ASE v3), Functions, App Service, Service Bus, Logic Apps, and a NAT Gateway. The design ensures high availability, security, and compliance with GDPR requirements.

## 2. Component Descriptions

• \*\*Azure Front Door (AFD)\*\*: Provides global load balancing and web application firewall (WAF) capabilities, improving application performance and security.  
• \*\*API Management (APIM)\*\*: Manages APIs across the environment, handling authentication, authorization, and rate limiting.  
• \*\*App Service Environment v3 (ASE v3)\*\*: An isolated environment within a virtual network for securely hosting web applications.  
• \*\*Azure Functions\*\*: Serverless compute service for running event-driven code with scaling based on demand.  
• \*\*App Service\*\*: Hosts web applications with scaling and management features.  
• \*\*Service Bus\*\*: Messaging infrastructure for reliable communication between distributed systems.  
• \*\*Logic Apps\*\*: Facilitates workflow automation and integration with various services.  
• \*\*NAT Gateway\*\*: Manages outbound connections for resources within the virtual network.

## 3. Cost Estimates

The following table provides cost estimates for each component in the three regions (EMEA, APAC, AMER). These estimates are based on Azure's regional pricing and should be verified using the Azure Cost Estimator. Recommendations for cost-saving measures and performance optimizations are also included.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Component | Configuration | EMEA Cost | APAC Cost | AMER Cost |
| Azure Front Door (AFD) | Global distribution, WAF enabled | $... | $... | $... |
| API Management (APIM) | Standard tier | $... | $... | $... |
| App Service Environment (ASE v3) | Isolated environment | $... | $... | $... |
| Azure Functions | Consumption-based | $... | $... | $... |
| App Service | Standard tier | $... | $... | $... |
| Service Bus | Standard tier | $... | $... | $... |
| Logic Apps | Consumption-based | $... | $... | $... |
| NAT Gateway | Standard configuration | $... | $... | $... |

\*\*Total Estimated Monthly Cost (per region):\*\*

## 4. Cost-Saving and Performance Considerations

### Azure Front Door (AFD)

* • Enable caching to reduce backend load and lower request costs.
* • Optimize WAF rules to avoid unnecessary inspection of low-risk traffic.
* • Use single Front Door instance globally to reduce regional deployments.
* • Reduce data egress by minimizing cross-region traffic.
* • Consider traffic compression for improved data transfer efficiency.

### API Management (APIM)

* • Consolidate APIs where possible to avoid redundant instances.
* • Use caching at APIM to reduce backend calls and improve latency.
* • Optimize tier based on actual usage to avoid over-provisioning.
* • Limit API calls with rate limiting to avoid unnecessary consumption.
* • Enable developer tiers for test environments to reduce production costs.

### App Service Environment (ASE v3)

* • Scale only during peak times to save on resources during off-hours.
* • Consider sharing app service plans among applications when possible.
* • Use reserved instances to lock in lower rates if demand is predictable.
* • Optimize resource usage by monitoring and adjusting plan sizes.
* • Configure autoscaling thresholds carefully to avoid rapid scaling.

### Azure Functions

* • Use Consumption Plan for dynamic workloads to avoid idle costs.
* • Apply pre-warmed instances only for critical functions to reduce latency.
* • Avoid overusing HTTP triggers to prevent high egress costs.
* • Monitor function duration and optimize code to reduce execution time.
* • Consolidate related functions into single function apps when possible.

### App Service

* • Consolidate applications in shared App Service Plans to optimize usage.
* • Leverage autoscaling features to manage workload variations efficiently.
* • Use lower-tier plans for non-critical services to minimize cost.
* • Optimize app startup time to reduce scaling costs.
* • Disable idle instances during non-peak times to save on compute costs.

### Service Bus

* • Use Basic tier for non-critical workloads where possible.
* • Limit message retention and cleanup unused topics to save on storage.
* • Optimize partitioning to avoid over-provisioning based on usage patterns.
* • Use short-lived sessions for low-latency messaging at minimal cost.
* • Monitor and adjust throughput units based on usage trends.

### Logic Apps

* • Consolidate workflows to avoid redundant Logic Apps for similar tasks.
* • Use recurrence triggers only when necessary to reduce executions.
* • Optimize workflows by reducing unnecessary steps for lower costs.
* • Set up alerts for high consumption workflows to catch inefficiencies early.
* • Consider Azure Functions for lightweight workflows to minimize costs.

### NAT Gateway

* • Limit the NAT Gateway usage to essential resources only.
* • Avoid unnecessary outbound internet connections to reduce costs.
* • Optimize NSG rules to minimize traffic through NAT Gateway.
* • Monitor data usage and set alerts for high traffic to control egress costs.
* • Use private endpoints to avoid NAT Gateway usage for internal traffic.

## 5. Disaster Recovery (DR) Strategy

The disaster recovery strategy for this architecture is designed to ensure compliance with GDPR and maintain high availability across EMEA, APAC, and AMER regions. Key DR considerations include:  
• \*\*Data Replication\*\*: Use geo-redundant storage options for data replication across Azure paired regions.  
• \*\*Backup\*\*: Implement automated backups for critical data in databases and storage accounts.  
• \*\*Failover\*\*: Use Azure Traffic Manager or Front Door to direct traffic to backup regions in case of a failure.  
• \*\*Testing and Monitoring\*\*: Conduct regular failover testing and monitor DR performance for readiness.  
• \*\*Data Residency\*\*: Ensure all data remains within the specified region (EMEA, APAC, AMER) for compliance.