AZ 305
Azure subscription best practices

www.linkedin.com/in/saraswathilakshman
saraswathilakshman.medium.com

Azure Subscription Design Best Practices Guide

Core Principles

Azure subscription design is crucial for governance, security, cost management, and operational efficiency. The subscription acts as a boundary for billing, access control, and resource management.

Key Best Practices with Real-World Examples

1. **Subscription per Environment Strategy**

Principle: Separate environments using different subscriptions to provide clear boundaries and prevent accidental changes.

Real-World Example: A retail company "ShopCorp" uses:

- **Production Subscription**: `ShopCorp-Prod-001`
  - Contains live e-commerce website, customer databases
  - Strict access controls (only senior engineers and operations team)
  - High-tier SLAs and backup policies
- **Development Subscription**: `ShopCorp-Dev-001`
  - Contains development resources, test databases
  - Broader access for development team
  - Lower-cost SKUs and relaxed policies
- **Staging Subscription**: `ShopCorp-Staging-001`
  - Mirror of production for final testing
  - Production-like configuration but separate billing

Benefits: Cost isolation, security separation, independent scaling, clear accountability.

2. **Business Unit Separation**

Principle: Create separate subscriptions for different business units or departments with distinct requirements.

Real-World Example: A multinational corporation "TechGlobal" organizes by business unit:

- **TechGlobal-Finance-Prod**: Financial systems, compliance-heavy workloads
- **TechGlobal-Marketing-Prod**: CRM systems, analytics platforms
- **TechGlobal-HR-Prod**: HRIS, payroll systems with strict privacy requirements
- **TechGlobal-Engineering-Prod**: Development tools, CI/CD pipelines

Benefits: Independent billing, unit-specific governance, tailored compliance policies.

3. **Geographic/Regional Strategy**

Principle: Separate subscriptions by geographic regions for data sovereignty, latency optimization, or regulatory compliance.

Real-World Example: A global SaaS company "DataFlow" uses regional subscriptions:

- **DataFlow-US-East-Prod**: Serves North American customers
- **DataFlow-EU-West-Prod**: Serves European customers (GDPR compliance)
- **DataFlow-APAC-Southeast-Prod**: Serves Asia-Pacific customers
- **DataFlow-Shared-Global**: Global services like DNS, CDN management

Benefits: Data residency compliance, reduced latency, regional cost optimization.

4. **Workload-Based Separation**

Principle: Isolate different types of workloads that have varying requirements, SLAs, or lifecycle patterns.

Real-World Example: A healthcare provider "MedTech Solutions":

- **MedTech-PatientPortal-Prod**: Patient-facing applications (high availability)
- **MedTech-Analytics-Prod**: Data analytics and machine learning workloads
- **MedTech-Archive-Prod**: Long-term data storage and compliance systems
- **MedTech-Integration-Prod**: Third-party integrations and APIs

Benefits: Workload-specific optimization, independent scaling, targeted security policies.

5. **Naming Convention Standards**

Principle: Implement consistent, descriptive naming conventions across all subscriptions.

Real-World Example: Standard format: `{Company}-{BusinessUnit/Purpose}-{Environment}-{Region?}-{Number}`

Examples:

- `Contoso-Finance-Prod-EastUS-001`
- `Contoso-DevOps-Shared-Global-001`
- `Contoso-Marketing-Dev-WestEU-001`
- `Contoso-Compliance-Prod-CentralUS-001`

Benefits: Easy identification, automated governance, simplified management.

6. **Subscription Quotas and Limits Management**

Principle: Understand and plan for Azure subscription limits to avoid hitting resource constraints.

Real-World Example: An ISV "CloudApp Inc" hit the 980 VNet limit in their main subscription:

Problem: Single subscription approach led to:

- 980+ VNets (approaching 1000 limit)
- 15,000+ VMs (approaching 25,000 limit)
- Complex resource management

Solution: Redesigned architecture:

- **CloudApp-Compute-Prod-001**: Primary compute resources
- **CloudApp-Compute-Prod-002**: Additional compute when limits approached
- **CloudApp-Network-Prod-001**: Dedicated networking resources
- **CloudApp-Storage-Prod-001**: Dedicated storage accounts

7. **Hub-and-Spoke Network Architecture**

Principle: Use a connectivity subscription for shared networking resources with spoke subscriptions for workloads.

Real-World Example: "Global Manufacturing Corp" network design:

Hub Subscription: `GMC-Connectivity-Prod-001`

- Central hub VNet with Azure Firewall
- VPN Gateway for on-premises connectivity
- Shared services like DNS, monitoring

Spoke Subscriptions:

- `GMC-ERP-Prod-001`: ERP system with spoke VNet
- `GMC-CRM-Prod-001`: CRM system with spoke VNet
- `GMC-Analytics-Prod-001`: Analytics platform with spoke VNet

All spokes peer to the hub for centralized security and connectivity.

8. **Cost Management Strategy**

Principle: Structure subscriptions to enable granular cost tracking and optimization.

Real-World Example: "StartupTech" cost optimization approach:

Before: Single subscription - difficult to track costs per product **After**: Multiple subscriptions:

- `StartupTech-ProductA-Prod-001`: $15,000/month - profitable
- `StartupTech-ProductB-Prod-001`: $8,000/month - break-even
- `StartupTech-ProductC-Prod-001`: $12,000/month - needs optimization
- `StartupTech-Shared-Prod-001`: $3,000/month - shared services

This enabled product-level P&L analysis and targeted cost optimization.

9. **Security and Compliance Boundaries**

Principle: Use subscriptions to create security boundaries for different compliance requirements.

Real-World Example: "SecureBank" compliance-driven design:

- **SecureBank-PCI-Prod-001**: Payment processing (PCI DSS compliance)
  - Isolated network, enhanced monitoring
  - Limited access, additional encryption

- **SecureBank-General-Prod-001**: General banking applications
    - Standard compliance requirements
    - Regular security controls
- **SecureBank-Analytics-Prod-001**: Customer analytics (anonymized data)
    - Relaxed controls for development agility
    - No sensitive customer data

10. **Subscription Lifecycle Management**

Principle: Plan for subscription creation, management, and eventual decommissioning.

Real-World Example: "AgileDevCorp" project-based subscription lifecycle:

Creation Process:

1. Project "Phoenix" gets subscription `AgileDevCorp-Phoenix-Dev-001`
2. Automated governance policies applied
3. Budget limits and alerts configured
4. Development team granted access

Lifecycle Management:

- **Active Development**: Full resource allocation
- **Maintenance Mode**: Reduced resource allocation
- **End-of-Life**: Data backup, resource cleanup, subscription cancellation

Decommissioning Process:

1. Data retention policy execution
2. Resource inventory and cleanup
3. Cost analysis and lessons learned
4. Subscription marked for deletion

Implementation Roadmap

Phase 1: Assessment (Weeks 1-2)

- Audit current subscription usage
- Identify business requirements and constraints
- Map existing resources and dependencies

Phase 2: Design (Weeks 3-4)

- Create subscription taxonomy
- Define naming conventions
- Plan network architecture
- Design governance framework

Phase 3: Implementation (Weeks 5-12)

- Create new subscriptions following best practices
- Implement Azure Policy and RBAC
- Set up cost management and monitoring
- Migrate resources as needed

Phase 4: Optimization (Ongoing)

- Monitor usage patterns and costs
- Adjust subscription boundaries as needed
- Implement automation and governance improvements
- Regular reviews and updates

Common Anti-Patterns to Avoid

1. **Single Subscription for Everything**: Creates management complexity and security risks
2. **Too Many Subscriptions**: Increases administrative overhead unnecessarily
3. **Inconsistent Naming**: Makes management and automation difficult
4. **Ignoring Limits**: Leads to unexpected constraints and downtime
5. **Poor Network Planning**: Results in complex connectivity and security challenges

Monitoring and Governance

Key Metrics to Track

- **Cost per subscription**: Monthly spend analysis
- **Resource utilization**: Identify optimization opportunities
- **Security compliance**: Policy adherence monitoring
- **Performance metrics**: SLA achievement tracking

Automation Opportunities

- Subscription provisioning workflows
- Policy enforcement automation
- Cost anomaly detection
- Resource lifecycle management

This subscription design strategy provides a foundation for scalable, secure, and cost-effective Azure operations while maintaining flexibility for future growth and changes.

🌍 Azure Multi-Region Deployments: One Subscription or Many? Clearing the Confusion

A practical guide to subscription strategy for global applications

"Do I need separate Azure subscriptions for each region where I deploy my application?"

This question comes up in almost every cloud architecture discussion I have. The answer isn't straightforward—it depends on your specific requirements. Let me break down the scenarios with real-world examples to help you make the right decision.

🤔 The Common Confusion

Many architects assume that "geographic distribution" automatically means separate subscriptions per region. This isn't always the case. The subscription strategy should align with your **compliance, operational, and business requirements**—not just geography.

📋 Three Distinct Scenarios

Scenario 1: Same App, Single Subscription

Focus: Performance & Availability

Example: Global e-commerce platform "ShopFast"

```
✅ One Subscription: ShopFast-Global-Prod
  📍 East US: Primary database + web app
  📍 West Europe: Read replica + web app
  📍 Southeast Asia: Read replica + web app
```

Database Strategy: Primary database with regional read replicas **Use Case**: When data can freely cross borders and you want centralized management

Scenario 2: Same App, Multiple Subscriptions

Focus: Data Sovereignty & Compliance

Example: Banking app "GlobalBank" with GDPR requirements

```
🏢 US Subscription: GlobalBank-US-Prod
  📍 East US: Independent US customer database

🏢 EU Subscription: GlobalBank-EU-Prod
  📍 West Europe: Separate EU customer database
```

Database Strategy: Completely isolated databases per region **Use Case**: When regulations require data residency (GDPR, banking laws, healthcare)

Scenario 3: Different Apps per Region

Focus: Regional Business Requirements

Example: Multinational corp with region-specific needs

```
US US Subscription: Payroll system (US labor laws)
EU EU Subscription: GDPR compliance portal
JP APAC Subscription: Local payment integrations
```

🎯 Decision Framework

Choose SINGLE subscription when:

- ✅ Same application serving global users
- ✅ No strict data residency requirements
- ✅ Centralized operations team
- ✅ Shared cost allocation is acceptable

Choose MULTIPLE subscriptions when:

- 🔒 Data sovereignty requirements (GDPR, HIPAA)
- 🏢 Independent regional operations teams
- 💰 Need separate billing per region
- ⚖️ Different compliance requirements
- 🎯 Different SLA requirements per region

🗄️ Database Sharing Patterns

Pattern 1: Global + Regional Replicas

Perfect for Netflix-style content platforms:

- **Shared**: Content catalog (can be global)
- **Regional replicas**: For performance optimization
- **Use case**: Social media, streaming, e-commerce

Pattern 2: Complete Regional Isolation

Essential for regulated industries:

- **US Database**: US customer data only
- **EU Database**: EU customer data only
- **Use case**: Banking, healthcare, government

Pattern 3: Hybrid Approach

Best of both worlds:

- **Global**: Product catalogs, shared reference data
- **Regional**: Customer data, transaction records
- **Use case**: E-commerce with compliance requirements

📊 Real-World Example: StreamCorp Architecture

```
🎬 Content Distribution (Single Subscription)
  Global content catalog + regional CDN

👥 User Data (Separate Subscriptions)
  US US subscription: US user profiles
  EU EU subscription: EU user profiles (GDPR)
  🌏 APAC subscription: APAC user profiles
```

Why this works: Content can be shared globally, but user data must stay regional for compliance.

💡 Key Takeaways

1. **Geography ≠ Automatic Subscription Split**: Base decisions on compliance and operational needs, not just location
2. **Database Strategy Matters**: Consider data sharing laws before architecting your data layer
3. **Start Simple, Scale Smart**: Begin with single subscription if possible, split when compliance or operations require it
4. **Plan for Growth**: Design your subscription hierarchy to accommodate future regions and requirements

🚀 Action Items

- **Audit your current multi-region setup**: Are you over-complicating with unnecessary subscription splits?
- **Review compliance requirements**: Do you actually need data residency, or just performance optimization?
- **Assess operational needs**: Can your team manage global resources, or do you need regional ownership?

What's your experience with Azure multi-region deployments? Have you faced similar subscription strategy decisions?

Drop a comment below—I'd love to hear about your real-world challenges and solutions! 👇

#Azure #CloudArchitecture #MultiRegion #GDPR #CloudStrategy #TechLeadership

---

Found this helpful? Follow me for more cloud architecture insights and real-world Azure best practices.