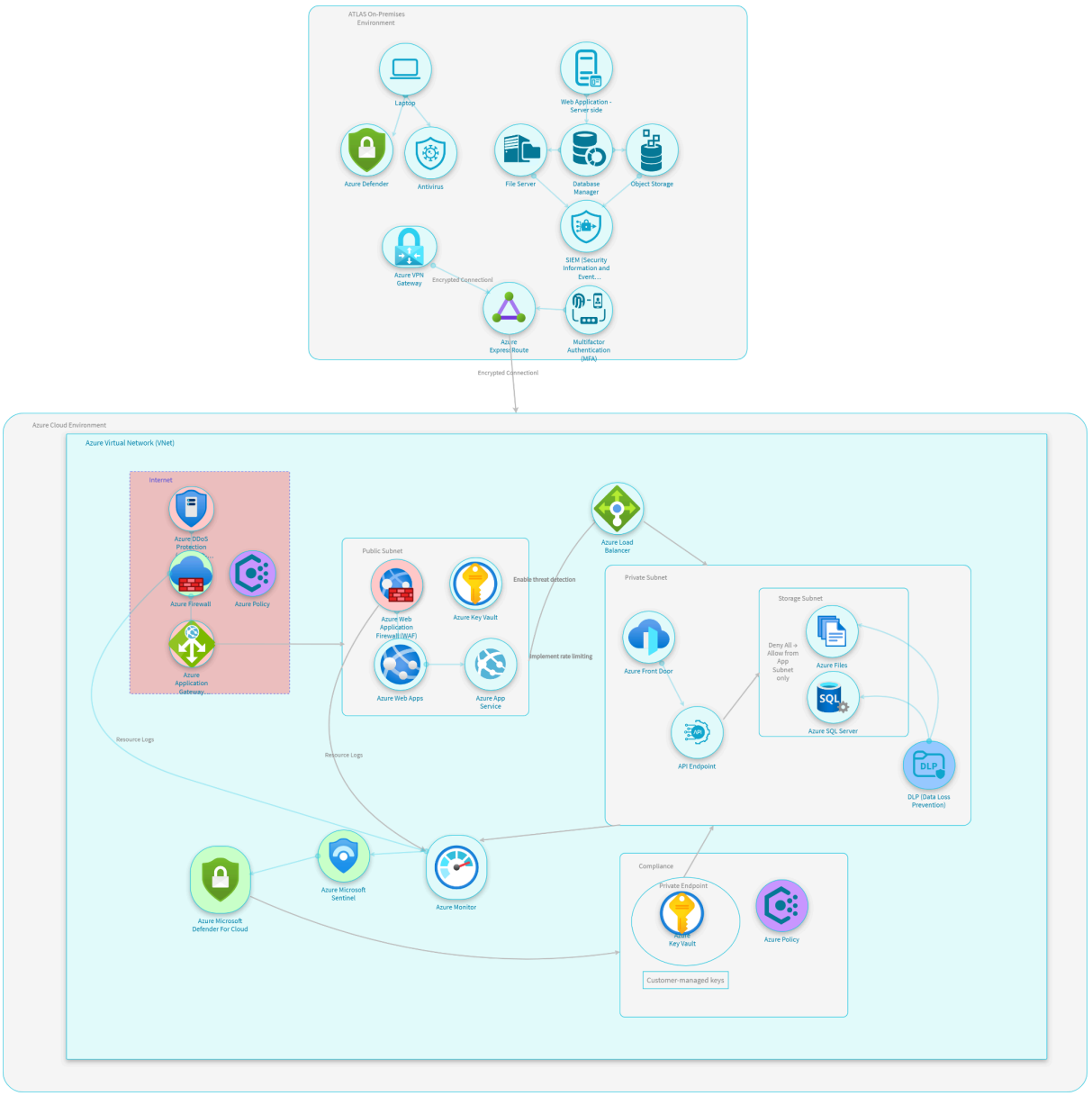
Cloud Migration Security Report for ATLAS

Date: April.16 2025

CSCL1020
Xuanyu Zhu

Cloud Network Architecture Diagram



Executive Summary

ATLAS, one of the leading financial and insurance service providers, is pursuing a strategic cloud migration to modernize its IT infrastructure and enhance operating flexibility. The migration involves moving on-premises systems—applications, servers, databases, and file servers—to a cloud service provider (Microsoft Azure). Currently, ATLAS employs standard security controls such as Host-Based Intrusion Prevention Systems (HIPS), Next-Generation Firewalls (NGFW), Web Application Firewalls (WAF), and a Security Information and Event Management (SIEM) tool, all configured at default. Our intention is to tailor these controls to a cloud environment and add more controls to protect sensitive personal information (PII) on the public site and protect the multi-tenant cloud environment.

Migration key points:

1. Azure Native Security Integration

Replaced third-party tools (e.g., Akamai Guardicore) with **Microsoft Defender for Cloud** and **Azure Policy**, enabling centralized threat detection, automated compliance, and seamless protection for Azure workloads.

2. Data Encryption & Sovereignty

Introduced **customer-managed keys (CMK)** and **Azure Key Vault** to control encryption keys, ensuring compliance (GDPR/HIPAA) and eliminating hard-coded credential risks.

3. AI-Driven Threat Detection

Deployed **Azure Sentinel** (SIEM/SOAR) and **Defender for Cloud** for proactive threat hunting, automated incident response, and cross-platform hybrid workload protection.

4. Modernized Network Security

Upgraded to **Azure Firewall**, **WAF**, and **DDoS Protection**, providing layered defense (L3-L7), scalability, and cost-efficient threat mitigation.

5. Zero Trust Identity Controls

Enforced **Azure AD Conditional Access** and **MFA** universally, blocking unauthorized access and enforcing least privilege via granular RBAC.

6. Automated Compliance

Leveraged **Azure Policy** and **Compliance Dashboard** to auto-remediate misconfigurations, streamline audits, and enforce policy-as-code governance.

Risk Summary

This following risk report complements the technical threat assessment, highlighting ongoing vulnerabilities and the effectiveness of implemented countermeasures. Despite partial mitigations, critical risks persist across Azure components, requiring prioritized remediation. Key insights include:

Current Risk: 68% (High), with Projected Risk dropping to 4% (Low) if all countermeasures are implemented.

Top Residual Risks: Misconfigured RBAC roles, unencrypted data flows, insecure certificate management, and inadequate logging.

Critical Components: Azure Key Vault, SQL Server, App Service, and Front Door remain high-priority due to exposed credentials, SQL injection risks, and DDoS vulnerabilities.

Assumptions

Countermeasures marked as "REQUIRED" are partially implemented but not fully validated.

Microsoft's shared responsibility model is assumed, but gaps in customer-managed controls (e.g., RBAC, encryption) persist.

Threat actors include external attackers (e.g., DDoS, credential stuffing) and insider threats.

Residual Risk Summary

1. Identity & Access Management

Risks: Too many permissions assigned (e.g., managed identities with multiple roles) and weak backup MFA methods like SMS.

Impact: Hackers love over privileged accounts—it's like giving them a master key. SMS can be intercepted (e.g., SIM-swapping scams). They could steal accounts, access sensitive data, or move freely in the system.

Fix: Tighten access: Give people/robots only the permissions they actually need.

Ditch SMS: Use physical security keys (FIDO2) or app-based codes.

2. Data Security

Risks: Data stored without encryption (Azure Files/SQL) and passwords hardcoded in scripts (e.g., Azure Front Door).

Impact: Data leaks, legal fines (GDPR/PCI DSS), and loss of customer trust.
Fix: Encrypt everything: Turn on Azure's built-in encryption tools. Lock up secrets: Use Azure Key Vault—no more hardcoded passwords. Rotate keys automatically.

3. Network & Endpoint Security
Risks: Publicly open services (VPN, Firewall) and weak network rules (NSGs). DDoS protection is weak or missing. DDoS attacks could crash systems.
Impact: Service outages, hacked data, or ransomware attacks.
Fix: Lock the gates: Use Azure Private Link to hide services from the public internet. Prepare for storms: Enable Azure's DDoS Protection and a Web Application Firewall (WAF).

4. Compliance & Monitoring
Risks: Missing logs and policy violations (e.g., unapproved cloud services).
Impact: No logs = flying blind during an attack. Rule-breaking = audit nightmares and fines. Slow response to attacks, failed audits, and financial penalties.
Fix: Automate rules: Use Azure Policy to block non-compliant setups. Centralize logs: Dump everything into Log Analytics/Sentinel. Set alerts for weird activity.

5. Third-Party & Supply Chain
Risks: Outdated plugins (e.g., Database Manager) and insecure DevOps pipelines.
Impact: Hackers target third-party tools—they're the weak link. A single compromised plugin can infect everything. Malware infections, system takeovers, or downtime.
Fix: Scan dependencies: Tools like SCA catch risky plugins. Secure pipelines: Never store deployment secrets in plain text. Enforce device firmware updates.

These gaps could lead to breaches, fines, or operational chaos. Simple fixes—like stricter access rules, encryption, and automated monitoring—will cut risks significantly. Focus on high-impact changes first (e.g., killing SMS MFA, encrypting data) to avoid worst-case scenarios.

Expanded Recommendations
1. **Kill SMS MFA**
Disable SMS as a fallback option in Azure AD Conditional Access policies.
Notify users to switch to stronger methods (e.g., FIDO2 keys, Microsoft Authenticator app).
Train users on phishing-resistant MFA, SMS is the "old lockpickable door" of authentication. Phishing-resistant MFA (like FIDO2) stops 99% of credential theft attempts. It's a one-time pain for long-term security.

2. **Encrypt Everything**
Enable encryption by default, for Azure Files/SQL: Turn on Transparent Data Encryption (TDE) and use customer-managed keys (CMK) instead of Microsoft-managed keys.
Ensure Azure Backup Vaults use CMK. Use tools like GitGuardian or Azure's built-in code scanning to find passwords in code/config files. Replace them with Key Vault references. Set Key Vault to rotate keys every 90 days. Test workflows to avoid app outages during rotations. Encrypted data is useless to hackers—even if they steal it. CMK ensures you control access, not a third party.

3. **Hide Public Services**
Get off the public internet, use Azure Private Link for VPNs, databases, and storage accounts. This makes them accessible only through private networks (no public IPs). For must-have

public endpoints (e.g., customer-facing apps), restrict access with NSG rules (e.g., allow only from specific IP ranges). Run Azure’s Security Center recommendations to find accidentally exposed services (e.g., a test VM left public). Shut them down or move them to private networks. Deploy Azure Firewall + WAF, block known malicious IPs and SQL injection attempts. Set WAF to "Prevention Mode"—no "Detection Mode" half-measures.