Table of Contents

# Azure Agentic AI Support Bot - Comprehensive Design Package

# Reference Architecture diagram + description showing agents, data sources, tools, identity/permissions, runtime, and guardrails

# Azure Agentic AI Support Bot - Reference Architecture

## Overview

The Azure Agentic AI Support Bot is a production-ready system designed to troubleshoot Windows VM RDP connectivity issues using Microsoft’s latest AI technologies. The architecture leverages Azure OpenAI, Semantic Kernel, and Azure-native services to provide intelligent, automated support with comprehensive safety and governance controls.

## Reference Architecture Diagram

┌─────────────────────────────────────────────────────────────────────────────────┐  
│ Azure Agentic AI Support Bot │  
├─────────────────────────────────────────────────────────────────────────────────┤  
│ │  
│ ┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐ │  
│ │ Azure Portal │ │ Bot Service │ │ Web Chat │ │  
│ │ Support │ │ Integration │ │ Interface │ │  
│ └─────────┬───────┘ └─────────┬───────┘ └─────────┬───────┘ │  
│ │ │ │ │  
│ └──────────────────────┼──────────────────────┘ │  
│ │ │  
│ ┌─────────────────────────────────▼─────────────────────────────────────────┐ │  
│ │ Azure Bot Service (Entry Point) │ │  
│ │ ┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐ │ │  
│ │ │ Authentication │ │ RBAC │ │ Rate Limiting │ │ │  
│ │ │ & Identity │ │ Authorization │ │ & Throttling │ │ │  
│ │ └─────────────────┘ └─────────────────┘ └─────────────────┘ │ │  
│ └─────────────────────────────────┬─────────────────────────────────────────┘ │  
│ │ │  
│ ┌─────────────────────────────────▼─────────────────────────────────────────┐ │  
│ │ Azure Functions (Agent Runtime) │ │  
│ │ ┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐ │ │  
│ │ │ Main Agent │ │ Diagnostic │ │ Resolution │ │ │  
│ │ │ Orchestrator │ │ Agent │ │ Agent │ │ │  
│ │ └─────────────────┘ └─────────────────┘ └─────────────────┘ │ │  
│ └─────────────────────────────────┬─────────────────────────────────────────┘ │  
│ │ │  
│ ┌─────────────────────────────────▼─────────────────────────────────────────┐ │  
│ │ Microsoft Semantic Kernel │ │  
│ │ ┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐ │ │  
│ │ │ Plugin │ │ Plugin │ │ Plugin │ │ │  
│ │ │ Orchestration │ │ Management │ │ Execution │ │ │  
│ │ └─────────────────┘ └─────────────────┘ └─────────────────┘ │ │  
│ └─────────────────────────────────┬─────────────────────────────────────────┘ │  
│ │ │  
│ ┌─────────────────────────────────▼─────────────────────────────────────────┐ │  
│ │ Azure OpenAI Service │ │  
│ │ ┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐ │ │  
│ │ │ GPT-4 Turbo │ │ Embeddings │ │ Content │ │ │  
│ │ │ Deployment │ │ Model │ │ Safety │ │ │  
│ │ └─────────────────┘ └─────────────────┘ └─────────────────┘ │ │  
│ └─────────────────────────────────┬─────────────────────────────────────────┘ │  
│ │ │  
│ ┌─────────────────────────────────▼─────────────────────────────────────────┐ │  
│ │ Azure Native Services & Plugins │ │  
│ │ ┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐ │ │  
│ │ │ Azure Monitor │ │ Network │ │ VM Operations │ │ │  
│ │ │ Diagnostics │ │ Watcher │ │ Management │ │ │  
│ │ └─────────────────┘ └─────────────────┘ └─────────────────┘ │ │  
│ │ ┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐ │ │  
│ │ │ Resource │ │ Compute │ │ Network │ │ │  
│ │ │ Graph │ │ Management │ │ Management │ │ │  
│ │ └─────────────────┘ └─────────────────┘ └─────────────────┘ │ │  
│ └─────────────────────────────────┬─────────────────────────────────────────┘ │  
│ │ │  
│ ┌─────────────────────────────────▼─────────────────────────────────────────┐ │  
│ │ Data Sources & Storage │ │  
│ │ ┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐ │ │  
│ │ │ Cosmos DB │ │ Key Vault │ │ Application │ │ │  
│ │ │ (Memory) │ │ (Secrets) │ │ Insights │ │ │  
│ │ └─────────────────┘ └─────────────────┘ └─────────────────┘ │ │  
│ └─────────────────────────────────┬─────────────────────────────────────────┘ │  
│ │ │  
│ ┌─────────────────────────────────▼─────────────────────────────────────────┐ │  
│ │ Safety & Governance │ │  
│ │ ┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐ │ │  
│ │ │ Content │ │ Input │ │ Action │ │ │  
│ │ │ Safety │ │ Validation │ │ Authorization │ │ │  
│ │ └─────────────────┘ └─────────────────┘ └─────────────────┘ │ │  
│ │ ┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐ │ │  
│ │ │ PII │ │ Audit │ │ Compliance │ │ │  
│ │ │ Detection │ │ Logging │ │ Monitoring │ │ │  
│ │ └─────────────────┘ └─────────────────┘ └─────────────────┘ │ │  
│ └─────────────────────────────────────────────────────────────────────────┘ │  
│ │  
└─────────────────────────────────────────────────────────────────────────────────┘

## Core Components

### 1. Agent Orchestration Layer

**Main Agent (Orchestrator)** - **Role**: Primary coordinator and conversation manager - **Responsibilities**: - Route requests to specialized agents - Manage conversation context and memory - Handle escalation and fallback scenarios - Ensure consistent user experience

**Diagnostic Agent** - **Role**: RDP connectivity troubleshooting specialist - **Responsibilities**: - Execute comprehensive diagnostic sequences - Analyze VM health, network configuration, and service status - Generate detailed diagnostic reports - Recommend specific remediation actions

**Resolution Agent** - **Role**: Automated fix execution and validation - **Responsibilities**: - Execute approved remediation actions - Validate resolution effectiveness - Provide step-by-step guidance for manual fixes - Ensure safety through confirmation requirements

### 2. AI and Machine Learning

**Azure OpenAI Service** - **Model**: GPT-4 Turbo (gpt-4-1106-preview) - **Configuration**: - Temperature: 0.7 (balanced creativity) - Max Tokens: 1000 - Top P: 0.9 - Frequency Penalty: 0.5 - Presence Penalty: 0.5

**Microsoft Semantic Kernel** - **Purpose**: Agent orchestration and plugin management - **Features**: - Function calling and plugin execution - Memory management and context preservation - Safety guardrails and input validation - Multi-agent coordination

### 3. Data Sources and Tools

**Azure Resource Graph** - VM metadata and configuration - Resource relationships and dependencies - Compliance and policy status

**Azure Monitor & Log Analytics** - VM performance metrics - System and application logs - Guest diagnostics and health status

**Azure Network Watcher** - Network connectivity tests - Route analysis and troubleshooting - Packet capture and analysis

**Azure VM Guest Diagnostics** - Windows Event Logs - Performance counters - Service status and configuration - Firewall rules and security settings

### 4. Infrastructure Components

**Azure Bot Service** - Multi-channel support (Teams, Web Chat, Portal) - Authentication and authorization - Rate limiting and throttling - Message routing and delivery

**Azure Functions** - Serverless agent runtime - Event-driven execution - Auto-scaling capabilities - Cost-effective processing

**Azure Cosmos DB** - Conversation memory storage - Contextual information persistence - Global distribution and availability - Automatic scaling

**Azure Key Vault** - Secret and credential management - Certificate storage and rotation - Access policy enforcement - Audit logging

### 5. Safety and Security

**Azure Content Safety** - Harmful content detection - Prompt injection prevention - Content filtering and moderation

**Input Validation** - PII detection and masking - Malicious input filtering - Syntax and format validation

**Action Authorization** - RBAC-based permission checks - Dangerous operation confirmation - Audit trail generation

## Data Flow

### 1. Request Processing

1. User submits RDP issue via Portal/Support/Bot interface
2. Azure Bot Service authenticates and authorizes user
3. Request routed to Azure Functions agent runtime
4. Main Agent orchestrates the troubleshooting process

### 2. Diagnostic Phase

1. Diagnostic Agent extracts VM information from request
2. Executes comprehensive diagnostic sequence:
   * VM health and status check
   * Network Security Group validation
   * Firewall configuration analysis
   * RDP service status verification
   * Network connectivity testing
   * Authentication log analysis
3. Results analyzed and root causes identified
4. Recommendations generated and presented to user

### 3. Resolution Phase

1. User confirms recommended actions
2. Resolution Agent validates permissions and safety
3. Actions executed with confirmation requirements
4. Results validated through post-resolution testing
5. Success/failure reported with next steps

### 4. Memory and Learning

1. Conversation history stored in Cosmos DB
2. Diagnostic patterns and outcomes logged
3. Resolution effectiveness tracked
4. System performance and accuracy monitored

## Security Architecture

### Authentication and Authorization

* **Azure AD Integration**: Single sign-on with enterprise identity
* **RBAC Permissions**: Granular access control for Azure resources
* **Managed Identity**: Secure service-to-service authentication
* **Multi-Factor Authentication**: Enhanced security for privileged operations

### Data Protection

* **Encryption at Rest**: All data encrypted using Azure Key Vault
* **Encryption in Transit**: TLS 1.2+ for all communications
* **PII Detection**: Automatic identification and masking of sensitive data
* **Data Residency**: Compliance with regional data requirements

### Audit and Compliance

* **Comprehensive Logging**: All actions logged with timestamps and context
* **Audit Trail**: Immutable record of all system activities
* **Compliance Monitoring**: Continuous assessment against security standards
* **Incident Response**: Automated alerting and escalation procedures

## Scalability and Performance

### Horizontal Scaling

* **Azure Functions**: Automatic scaling based on demand
* **Cosmos DB**: Global distribution with automatic partitioning
* **Bot Service**: Multi-instance deployment for high availability

### Performance Optimization

* **Caching**: Intelligent caching of diagnostic results and configurations
* **Async Processing**: Non-blocking operations for improved responsiveness
* **Connection Pooling**: Efficient resource utilization
* **Load Balancing**: Even distribution of requests across instances

### Monitoring and Observability

* **Application Insights**: Real-time performance monitoring
* **Custom Metrics**: Business-specific KPIs and success rates
* **Alerting**: Proactive notification of issues and anomalies
* **Dashboards**: Comprehensive visibility into system health

## Deployment Architecture

### Infrastructure as Code

* **Terraform**: Complete infrastructure provisioning and management
* **Modular Design**: Reusable components for different environments
* **Version Control**: All infrastructure changes tracked and auditable
* **Environment Parity**: Consistent deployments across dev/staging/prod

### CI/CD Pipeline

* **Azure DevOps**: Automated build, test, and deployment
* **Quality Gates**: Automated testing and validation
* **Blue-Green Deployment**: Zero-downtime deployments
* **Rollback Capability**: Quick recovery from deployment issues

This architecture provides a robust, scalable, and secure foundation for automated Azure VM RDP troubleshooting while maintaining the highest standards of safety, governance, and user experience.

# Customer Flow to Resolution from entry (Portal/Support) to fix verification and closure

# Customer Flow to Resolution - Azure VM RDP Troubleshooting

## End-to-End Customer Journey

This document outlines the complete customer flow from initial issue report to resolution verification and case closure for Azure VM RDP connectivity problems.

## Entry Points

### 1. Azure Portal Integration

* **Access**: Direct integration within Azure Portal support section
* **Authentication**: Azure AD SSO with existing portal credentials
* **Context**: Automatic VM context detection and resource information
* **Advantages**: Seamless experience, no additional authentication required

### 2. Azure Support Ticket System

* **Access**: Traditional support ticket creation with AI bot option
* **Authentication**: Support ticket authentication flow
* **Context**: Manual VM specification with resource details
* **Advantages**: Familiar process, formal ticket tracking

### 3. Dedicated Web Chat Interface

* **Access**: Standalone web chat widget on support pages
* **Authentication**: Optional Azure AD or guest access
* **Context**: Manual resource specification
* **Advantages**: Quick access, no portal navigation required

## Detailed Customer Flow

### Phase 1: Initial Contact and Authentication

Customer Action → System Response → Next Step

1. **Customer Initiates Contact**
   * Selects entry point (Portal/Support/Web Chat)
   * System presents welcome message and authentication options
   * Customer authenticates using Azure AD or support credentials
2. **Identity Verification and Authorization**
   * System validates customer identity and subscription access
   * RBAC permissions checked for target VM resources
   * Customer context established with available permissions
3. **Issue Description Collection**
   * System prompts for VM details and RDP connectivity issue description
   * Customer provides:
     + VM name and resource group
     + Specific error messages or symptoms
     + Time when issue started
     + Any recent changes made

### Phase 2: Intelligent Issue Classification

Input Processing → Classification → Routing Decision

1. **Natural Language Processing**
   * Customer input analyzed using Azure OpenAI GPT-4
   * Intent classification: RDP troubleshooting vs. other issues
   * VM information extracted and validated
2. **Context Enhancement**
   * System retrieves VM metadata from Azure Resource Graph
   * Recent changes and events analyzed
   * Customer’s support history reviewed for patterns
3. **Routing Decision**
   * Issue classified as RDP-specific or general support
   * Appropriate agent (Diagnostic vs. General Support) selected
   * Conversation context initialized with relevant information

### Phase 3: Diagnostic Analysis

Diagnostic Execution → Analysis → Recommendation Generation

1. **Comprehensive Diagnostic Sequence**

* **Step 1: VM Health Assessment**
  + VM power state and provisioning status checked
  + Resource utilization and performance metrics analyzed
  + Recent events and alerts reviewed
* **Step 2: Network Security Analysis**
  + Network Security Group rules validated for RDP access (port 3389)
  + Subnet and virtual network configuration verified
  + Route table and network routing checked
* **Step 3: VM-Level Configuration**
  + Windows Firewall status and rules analyzed
  + RDP service status and configuration verified
  + Guest diagnostics and system logs examined
* **Step 4: Network Connectivity Testing**
  + End-to-end connectivity tests performed
  + Latency and packet loss measurements taken
  + Network path analysis completed
* **Step 5: Authentication and Security**
  + Authentication logs analyzed for failed attempts
  + Account lockout and password policy status checked
  + Recent security events reviewed

1. **Root Cause Analysis**
   * Diagnostic results correlated and analyzed
   * Most likely causes identified with confidence levels
   * Impact assessment and business risk evaluation
2. **Recommendation Generation**
   * Specific remediation actions proposed
   * Risk assessment for each recommended action
   * Estimated resolution time and effort provided
   * Alternative solutions offered where applicable

### Phase 4: Customer Interaction and Confirmation

Recommendation Presentation → Customer Review → Confirmation

1. **Diagnostic Results Presentation**
   * Clear, non-technical summary of findings
   * Visual representation of issues and their relationships
   * Detailed technical information available on request
2. **Remediation Plan Discussion**
   * Step-by-step action plan presented
   * Safety measures and rollback procedures explained
   * Customer approval required for automated actions
3. **Risk and Impact Communication**
   * Potential risks clearly communicated
   * Business impact assessment provided
   * Alternative approaches discussed if needed

### Phase 5: Resolution Execution

Action Authorization → Execution → Validation

1. **Pre-Action Safety Checks**
   * Customer permissions re-verified
   * Safety guardrails validated
   * Backup and rollback procedures prepared
2. **Automated Resolution (where appropriate)**
   * NSG rule creation/updates
   * VM start/restart operations
   * Firewall configuration changes
   * Service restart commands
3. **Manual Guidance (where required)**
   * Step-by-step instructions provided
   * Screenshots and command examples given
   * Real-time support during manual steps
4. **Resolution Validation**
   * Post-action diagnostics executed
   * RDP connectivity tested
   * Performance metrics verified
   * Customer confirmation requested

### Phase 6: Verification and Testing

Connectivity Testing → Performance Validation → Customer Confirmation

1. **Automated Connectivity Tests**
   * RDP port accessibility verified
   * Network latency and stability tested
   * Authentication flow validated
2. **Customer Testing**
   * Customer attempts RDP connection
   * System monitors connection success/failure
   * Additional troubleshooting if needed
3. **Performance Validation**
   * VM performance metrics checked
   * Resource utilization normalized
   * No adverse effects detected

### Phase 7: Documentation and Closure

Resolution Documentation → Knowledge Capture → Case Closure

1. **Resolution Documentation**
   * Complete resolution steps documented
   * Root cause analysis recorded
   * Time-to-resolution tracked
2. **Knowledge Base Update**
   * New patterns and solutions added
   * Diagnostic sequences refined
   * Customer feedback incorporated
3. **Case Closure Process**
   * Customer satisfaction survey
   * Follow-up scheduling if needed
   * Case marked as resolved

## Exception Handling and Escalation

### Automated Escalation Triggers

* **Complex Issues**: Multiple root causes or unusual symptoms
* **High-Risk Operations**: Potential for data loss or service disruption
* **Permission Issues**: Insufficient customer permissions for required actions
* **System Failures**: Diagnostic tools or services unavailable

### Escalation Process

1. **Intelligent Routing**: Issues routed to appropriate human specialist
2. **Context Transfer**: Complete diagnostic context and conversation history provided
3. **Seamless Handoff**: Customer experience maintained during transition
4. **Follow-up**: AI bot remains available for additional assistance

### Fallback Scenarios

* **Service Outages**: Graceful degradation with basic diagnostic capabilities
* **Authentication Failures**: Alternative authentication methods offered
* **Network Issues**: Offline diagnostic tools and guidance provided

## Success Metrics and KPIs

### Resolution Effectiveness

* **First-Call Resolution Rate**: Percentage of issues resolved without escalation
* **Time to Resolution**: Average time from issue report to resolution
* **Customer Satisfaction**: Post-resolution survey scores
* **Resolution Accuracy**: Percentage of successful resolutions

### System Performance

* **Response Time**: Time from customer input to system response
* **Diagnostic Accuracy**: Percentage of correct root cause identifications
* **Automation Rate**: Percentage of issues resolved without human intervention
* **Escalation Rate**: Percentage of issues requiring human support

### Customer Experience

* **Engagement Quality**: Conversation flow and clarity metrics
* **Information Accuracy**: Correctness of provided information
* **User Interface Usability**: Ease of use and navigation
* **Accessibility**: Support for different access methods and devices

## Continuous Improvement

### Learning and Adaptation

* **Pattern Recognition**: Identification of common issues and solutions
* **Model Refinement**: Regular updates to diagnostic algorithms
* **Feedback Integration**: Customer feedback incorporated into improvements
* **Performance Optimization**: Ongoing system performance enhancements

### Knowledge Management

* **Solution Database**: Continuously updated repository of solutions
* **Best Practices**: Documentation of proven troubleshooting approaches
* **Training Materials**: Resources for both customers and support staff
* **Community Knowledge**: Integration of community-contributed solutions

This customer flow ensures a comprehensive, efficient, and satisfying experience for customers experiencing Azure VM RDP connectivity issues while maintaining the highest standards of safety, accuracy, and service quality.

# Troubleshooting play tailored to Windows VM RDP failures (please hypothesize diagnostics available)

# RDP Troubleshooting Playbook - Windows VM Connectivity Issues

## Overview

This playbook provides a comprehensive framework for diagnosing and resolving Windows VM RDP connectivity issues using Azure-native diagnostic tools and services. The playbook is designed to be executed by the AI diagnostic agent but can also serve as a reference for human support engineers.

## Diagnostic Framework

### Prerequisites

* Azure subscription with appropriate permissions
* Access to Azure Resource Graph, Monitor, and Network Watcher
* VM Guest Diagnostics enabled
* Network connectivity to Azure services

### Diagnostic Sequence

The diagnostic process follows a systematic approach from basic connectivity checks to deep technical analysis:

1. VM Health Check → 2. Network Security → 3. VM Configuration →   
4. Network Connectivity → 5. Authentication → 6. Performance Analysis

## Detailed Diagnostic Procedures

### 1. VM Health and Status Check

#### 1.1 Power State Verification

**Tool**: Azure Resource Graph, Compute Management API **Query**:

Resources  
| where type == "microsoft.compute/virtualmachines"  
| where name == "{vm\_name}"  
| extend powerState = properties.extended.instanceView.statuses[1].displayStatus  
| project name, powerState, resourceGroup

**Expected Results**: - ✅ **Running**: VM is powered on and operational - ❌ **Stopped**: VM is shut down, needs to be started - ❌ **Deallocated**: VM is stopped and deallocated, needs to be started - ❌ **Unknown**: VM state unclear, requires investigation

**Resolution Actions**: - **Stopped**: Start VM using Azure portal or PowerShell - **Deallocated**: Start VM (will allocate resources first) - **Unknown**: Check Azure service health and retry

#### 1.2 Provisioning State Check

**Tool**: Azure Resource Graph **Query**:

Resources  
| where type == "microsoft.compute/virtualmachines"  
| where name == "{vm\_name}"  
| extend provisioningState = properties.provisioningState  
| project name, provisioningState, resourceGroup

**Expected Results**: - ✅ **Succeeded**: VM provisioning completed successfully - ❌ **Failed**: VM provisioning failed, requires manual intervention - ❌ **Creating/Updating**: VM is still being provisioned

### 2. Network Security Group (NSG) Analysis

#### 2.1 NSG Rule Validation for RDP

**Tool**: Azure Resource Graph, Network Management API **Query**:

Resources  
| where type == "microsoft.network/networksecuritygroups"  
| where name in ({nsg\_names})  
| extend rules = properties.securityRules  
| mvexpand rules  
| where rules.destinationPortRange == "3389" or rules.destinationPortRanges contains "3389"  
| project nsgName = name, ruleName = rules.name, access = rules.access,   
 direction = rules.direction, protocol = rules.protocol

**Critical Checks**: - **Inbound Rules**: Verify RDP (port 3389) is allowed from source - **Priority**: Check rule priority (lower numbers = higher priority) - **Source**: Validate source address prefix/cidr - **Protocol**: Confirm TCP protocol is specified

**Common Issues**: - ❌ **No RDP Rule**: NSG blocks all inbound RDP traffic - ❌ **Wrong Source**: Rule allows only specific IPs, not customer’s IP - ❌ **Wrong Port**: Rule configured for different port - ❌ **Deny Rule**: Higher priority rule explicitly denies RDP

#### 2.2 NSG Association Check

**Tool**: Azure Resource Graph **Query**:

Resources  
| where type == "microsoft.compute/virtualmachines"  
| where name == "{vm\_name}"  
| extend networkInterfaces = properties.networkProfile.networkInterfaces  
| mvexpand networkInterfaces  
| extend nicId = networkInterfaces.id  
| join kind=leftouter (  
 Resources  
 | where type == "microsoft.network/networkinterfaces"  
 | extend networkSecurityGroup = properties.networkSecurityGroup.id  
) on $left.nicId == $right.id

**Resolution Actions**: - **Missing NSG**: Create NSG with appropriate RDP rules - **Wrong NSG**: Update NSG association to correct one - **Incorrect Rules**: Modify NSG rules to allow RDP access

### 3. VM-Level Configuration Analysis

#### 3.1 Windows Firewall Status

**Tool**: Azure VM Guest Diagnostics, Run Command **PowerShell Command**:

Get-NetFirewallRule -DisplayName "\*Remote Desktop\*" |   
Select-Object DisplayName, Enabled, Direction, Action, Profile

**Expected Results**: - ✅ **Enabled**: RDP firewall rule is active - ❌ **Disabled**: RDP firewall rule is disabled - ❌ **Missing**: No RDP firewall rule exists

**Resolution Actions**: - **Enable Rule**: Enable-NetFirewallRule -DisplayName "Remote Desktop" - **Create Rule**: Create new firewall rule for RDP access - **Check Profile**: Ensure rule applies to correct network profile

#### 3.2 RDP Service Status

**Tool**: Azure VM Guest Diagnostics, Run Command **PowerShell Command**:

Get-Service -Name "TermService" |   
Select-Object Name, Status, StartType

**Expected Results**: - ✅ **Running**: RDP service is active and accepting connections - ❌ **Stopped**: RDP service is not running - ❌ **Disabled**: RDP service is disabled and won’t start

**Resolution Actions**: - **Start Service**: Start-Service -Name "TermService" - **Set Auto-Start**: Set-Service -Name "TermService" -StartupType Automatic - **Enable Service**: Set-Service -Name "TermService" -StartupType Automatic

#### 3.3 RDP Configuration

**Tool**: Azure VM Guest Diagnostics, Run Command **Registry Check**:

Get-ItemProperty -Path "HKLM:\System\CurrentControlSet\Control\Terminal Server" -Name "fDenyTSConnections"

**Expected Results**: - ✅ **0**: RDP connections are allowed - ❌ **1**: RDP connections are denied

**Resolution Actions**: - **Enable RDP**: Set-ItemProperty -Path "HKLM:\System\CurrentControlSet\Control\Terminal Server" -Name "fDenyTSConnections" -Value 0

### 4. Network Connectivity Testing

#### 4.1 End-to-End Connectivity Test

**Tool**: Azure Network Watcher Connectivity Check **API Call**:

POST https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Network/networkWatchers/{networkWatcherName}/connectivityCheck?api-version=2021-05-01  
  
{  
 "source": {  
 "resourceId": "/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Network/virtualNetworks/{vnetName}/subnets/{subnetName}"  
 },  
 "destination": {  
 "address": "{vm\_private\_ip}",  
 "port": 3389  
 }  
}

**Metrics Analyzed**: - **Reachability**: Can the destination be reached? - **Latency**: Round-trip time in milliseconds - **Hops**: Network path analysis - **Packet Loss**: Percentage of lost packets

#### 4.2 Port Accessibility Test

**Tool**: Azure VM Run Command **PowerShell Command**:

Test-NetConnection -ComputerName localhost -Port 3389 -InformationLevel Detailed

**Expected Results**: - ✅ **TcpTestSucceeded: True**: Port 3389 is accessible - ❌ **TcpTestSucceeded: False**: Port 3389 is not accessible

### 5. Authentication and Security Analysis

#### 5.1 Authentication Logs Analysis

**Tool**: Azure Monitor, Event Logs **Query**:

Event  
| where TimeGenerated > ago(24h)  
| where EventID in (4624, 4625, 4648)  
| where Computer == "{vm\_name}"  
| summarize count() by EventID, Account, SourceNetworkAddress  
| order by count\_ desc

**Event IDs**: - **4624**: Successful logon - **4625**: Failed logon attempt - **4648**: Logon attempt with explicit credentials

**Analysis Focus**: - **Failed Logons**: Patterns in failed authentication attempts - **Source IPs**: Geographic and network analysis of connection attempts - **Account Names**: Brute force attack patterns - **Time Patterns**: Unusual connection times or frequencies

#### 5.2 Account Status Check

**Tool**: Azure VM Run Command **PowerShell Command**:

Get-LocalUser | Where-Object {$\_.Enabled -eq $true} |   
Select-Object Name, Enabled, LastLogon, PasswordRequired

**Critical Checks**: - **Account Enabled**: User account is active - **Password Required**: Account has password set - **Account Lockout**: Account is not locked due to failed attempts - **Group Membership**: User is in appropriate groups (Remote Desktop Users)

### 6. Performance and Resource Analysis

#### 6.1 System Resource Utilization

**Tool**: Azure Monitor, Performance Counters **Query**:

Perf  
| where TimeGenerated > ago(1h)  
| where Computer == "{vm\_name}"  
| where CounterName in ("% Processor Time", "% Committed Bytes In Use", "Available MBytes")  
| summarize avg(CounterValue) by CounterName, bin(TimeGenerated, 5m)

**Thresholds**: - **CPU Usage**: > 90% for extended periods - **Memory Usage**: > 95% committed bytes - **Available Memory**: < 100 MB

#### 6.2 Network Performance

**Tool**: Azure Monitor, Network Performance Counters **Query**:

Perf  
| where TimeGenerated > ago(1h)  
| where Computer == "{vm\_name}"  
| where CounterName contains "Network"  
| summarize avg(CounterValue) by CounterName, bin(TimeGenerated, 5m)

**Metrics**: - **Bytes Total/sec**: Network throughput - **Packets/sec**: Packet rate - **Current Bandwidth**: Available bandwidth

## Common Issue Patterns and Solutions

### Pattern 1: VM Stopped or Deallocated

**Symptoms**: Connection timeout, no response to ping **Root Cause**: VM is not running **Solution**: Start the VM using Azure portal or PowerShell **Prevention**: Set up auto-start policies or alerts

### Pattern 2: NSG Blocking RDP

**Symptoms**: Connection timeout, port not accessible **Root Cause**: NSG rules deny RDP access **Solution**: Add NSG rule allowing TCP port 3389 from source IP **Prevention**: Document and standardize NSG configurations

### Pattern 3: Windows Firewall Blocking

**Symptoms**: Connection established but immediately dropped **Root Cause**: Windows Firewall blocks RDP traffic **Solution**: Enable Remote Desktop firewall rule **Prevention**: Use Group Policy to manage firewall rules

### Pattern 4: RDP Service Not Running

**Symptoms**: Connection refused, service unavailable **Root Cause**: Terminal Services service is stopped **Solution**: Start and enable Terminal Services service **Prevention**: Monitor service health and set auto-start

### Pattern 5: Authentication Issues

**Symptoms**: Connection established but authentication fails **Root Cause**: Invalid credentials, account lockout, or policy restrictions **Solution**: Reset password, unlock account, or adjust policies **Prevention**: Implement account lockout policies and monitoring

### Pattern 6: Network Connectivity Problems

**Symptoms**: Intermittent connectivity, high latency, packet loss **Root Cause**: Network routing, bandwidth, or infrastructure issues **Solution**: Optimize network configuration, increase bandwidth **Prevention**: Monitor network performance and capacity

## Automated Resolution Procedures

### Resolution 1: Start Stopped VM

Start-AzVM -ResourceGroupName "{resourceGroup}" -Name "{vmName}"

### Resolution 2: Add NSG Rule for RDP

$nsg = Get-AzNetworkSecurityGroup -ResourceGroupName "{resourceGroup}" -Name "{nsgName}"  
Add-AzNetworkSecurityRuleConfig -NetworkSecurityGroup $nsg -Name "AllowRDP" -Access Allow -Protocol Tcp -Direction Inbound -Priority 1000 -SourceAddressPrefix "\*" -SourcePortRange "\*" -DestinationAddressPrefix "\*" -DestinationPortRange "3389"  
Set-AzNetworkSecurityGroup -NetworkSecurityGroup $nsg

### Resolution 3: Enable RDP Firewall Rule

Enable-NetFirewallRule -DisplayName "Remote Desktop"

### Resolution 4: Start RDP Service

Start-Service -Name "TermService"  
Set-Service -Name "TermService" -StartupType Automatic

### Resolution 5: Enable RDP in Registry

Set-ItemProperty -Path "HKLM:\System\CurrentControlSet\Control\Terminal Server" -Name "fDenyTSConnections" -Value 0

## Validation and Testing

### Post-Resolution Validation

1. **Connectivity Test**: Verify RDP port accessibility
2. **Authentication Test**: Confirm successful login
3. **Performance Test**: Check system responsiveness
4. **Stability Test**: Monitor for recurring issues

### Success Criteria

* ✅ RDP connection established within 30 seconds
* ✅ Authentication successful with valid credentials
* ✅ Desktop loads completely without errors
* ✅ System performance within normal parameters
* ✅ No recurring connectivity issues for 24 hours

## Documentation and Knowledge Management

### Case Documentation

* **Issue Description**: Detailed problem statement
* **Diagnostic Results**: Complete diagnostic output
* **Root Cause Analysis**: Identified cause and contributing factors
* **Resolution Steps**: Actions taken to resolve the issue
* **Validation Results**: Confirmation of successful resolution
* **Lessons Learned**: Insights for future similar issues

### Knowledge Base Updates

* **New Patterns**: Previously unknown issue patterns
* **Solution Refinements**: Improved resolution procedures
* **Tool Enhancements**: Diagnostic tool improvements
* **Process Improvements**: Workflow optimizations

This troubleshooting playbook provides a comprehensive framework for systematically diagnosing and resolving Azure VM RDP connectivity issues while maintaining high standards of accuracy, efficiency, and customer satisfaction.

# Safety, security, and governance plan

# Safety, Security, and Governance Plan

## Overview

This document outlines the comprehensive safety, security, and governance framework for the Azure Agentic AI Support Bot. The plan ensures that the AI system operates within strict security boundaries while providing effective support services.

## Security Architecture

### 1. Multi-Layer Security Model

┌─────────────────────────────────────────────────────────────┐  
│ Security Layers │  
├─────────────────────────────────────────────────────────────┤  
│ Layer 1: Input Validation & Content Safety │  
│ Layer 2: Authentication & Authorization │  
│ Layer 3: Data Protection & Privacy │  
│ Layer 4: Operational Security │  
│ Layer 5: Compliance & Governance │  
└─────────────────────────────────────────────────────────────┘

### 2. Security Principles

* **Zero Trust Architecture**: Never trust, always verify
* **Principle of Least Privilege**: Minimum necessary permissions
* **Defense in Depth**: Multiple security controls
* **Privacy by Design**: Built-in privacy protection
* **Continuous Monitoring**: Real-time security oversight

## Input Validation and Content Safety

### 1. Content Safety Framework

#### Azure Content Safety Integration

* **Harmful Content Detection**: Automatic filtering of inappropriate content
* **Prompt Injection Prevention**: Detection and blocking of injection attempts
* **Content Classification**: Multi-category content analysis
* **Severity Scoring**: Risk-based content assessment

#### Input Validation Pipeline

def validate\_input(user\_input, context):  
 # Step 1: Content Safety Check  
 safety\_result = content\_safety\_client.analyze\_text(user\_input)  
   
 # Step 2: Prompt Injection Detection  
 injection\_detected = detect\_prompt\_injection(user\_input)  
   
 # Step 3: PII Detection and Masking  
 pii\_result = detect\_and\_mask\_pii(user\_input)  
   
 # Step 4: Syntax and Format Validation  
 format\_valid = validate\_format(user\_input)  
   
 return {  
 "safe": all\_checks\_passed,  
 "sanitized\_input": masked\_input,  
 "warnings": detected\_issues  
 }

### 2. Prompt Injection Prevention

#### Detection Patterns

* **Instruction Override**: “Ignore previous instructions”
* **Role Impersonation**: “You are now a different AI”
* **System Prompt Access**: “Show me your system prompt”
* **Token Manipulation**: Special character sequences
* **Context Injection**: Attempts to modify conversation context

#### Prevention Mechanisms

* **Pattern Matching**: Regex-based detection of known injection patterns
* **Semantic Analysis**: AI-powered detection of malicious intent
* **Context Isolation**: Strict separation of user input and system prompts
* **Input Sanitization**: Automatic cleaning of potentially malicious content

### 3. PII Detection and Protection

#### Detected PII Types

* **Email Addresses**: RFC-compliant email pattern matching
* **Phone Numbers**: Various international formats
* **Social Security Numbers**: US SSN format detection
* **Credit Card Numbers**: Luhn algorithm validation
* **IP Addresses**: IPv4 and IPv6 address detection
* **Azure Resource IDs**: Subscription, resource group, and resource identifiers

#### Protection Measures

* **Automatic Masking**: Real-time PII redaction
* **Encryption at Rest**: PII encrypted in storage
* **Access Logging**: All PII access events logged
* **Retention Policies**: Automatic PII data purging

## Authentication and Authorization

### 1. Azure AD Integration

#### Authentication Flow

User → Azure AD → Bot Service → Functions → AI Agents

#### Supported Authentication Methods

* **Single Sign-On (SSO)**: Azure AD integration
* **Multi-Factor Authentication (MFA)**: Enhanced security
* **Conditional Access**: Policy-based access control
* **Guest Access**: Limited external user support

### 2. Role-Based Access Control (RBAC)

#### Permission Levels

* **Reader**: View-only access to diagnostic information
* **Contributor**: Execute diagnostic and resolution actions
* **Owner**: Full access including configuration changes
* **Custom Roles**: Specific permission sets for specialized functions

#### Permission Matrix

| Action | Reader | Contributor | Owner |
| --- | --- | --- | --- |
| View VM Status | ✅ | ✅ | ✅ |
| Run Diagnostics | ❌ | ✅ | ✅ |
| Execute Fixes | ❌ | ✅ | ✅ |
| Modify Config | ❌ | ❌ | ✅ |
| Access Logs | ❌ | ✅ | ✅ |

### 3. Managed Identity Security

#### Service-to-Service Authentication

* **System-Assigned Identities**: Automatic identity management
* **User-Assigned Identities**: Shared identity across services
* **Key Vault Integration**: Secure credential storage
* **Automatic Rotation**: Regular credential updates

## Data Protection and Privacy

### 1. Encryption Standards

#### Data at Rest

* **Azure Key Vault**: AES-256 encryption for secrets
* **Cosmos DB**: Transparent Data Encryption (TDE)
* **Application Insights**: Encrypted log storage
* **Storage Accounts**: Service-managed encryption keys

#### Data in Transit

* **TLS 1.2+**: All external communications
* **HTTPS Only**: Web and API communications
* **Certificate Pinning**: Enhanced SSL/TLS security
* **Perfect Forward Secrecy**: Session key protection

### 2. Data Classification

#### Data Categories

* **Public**: General information, no restrictions
* **Internal**: Company-internal information
* **Confidential**: Sensitive business information
* **Restricted**: Highly sensitive data requiring special handling

#### Handling Requirements

* **Confidential Data**: Encryption required, access logging
* **Restricted Data**: Additional approval required, audit trails
* **PII Data**: Automatic masking, limited retention
* **Customer Data**: Subject to data residency requirements

### 3. Privacy Controls

#### Data Minimization

* **Purpose Limitation**: Data collected only for specific purposes
* **Retention Limits**: Automatic data deletion after defined periods
* **Access Controls**: Role-based data access restrictions
* **Consent Management**: User consent tracking and management

#### User Rights

* **Data Access**: Users can view their data
* **Data Correction**: Users can correct inaccurate data
* **Data Deletion**: Users can request data deletion
* **Data Portability**: Users can export their data

## Operational Security

### 1. Infrastructure Security

#### Network Security

* **Virtual Network**: Isolated network environment
* **Network Security Groups**: Traffic filtering and access control
* **Private Endpoints**: Secure service connectivity
* **DDoS Protection**: Automatic attack mitigation

#### Compute Security

* **Managed Identity**: Secure service authentication
* **Container Security**: Isolated execution environment
* **Runtime Protection**: Real-time threat detection
* **Vulnerability Scanning**: Regular security assessments

### 2. Monitoring and Detection

#### Security Monitoring

* **Azure Security Center**: Unified security management
* **Azure Sentinel**: Security information and event management
* **Application Insights**: Application-level monitoring
* **Custom Metrics**: Business-specific security KPIs

#### Threat Detection

* **Anomaly Detection**: Unusual behavior identification
* **Attack Pattern Recognition**: Known threat signature detection
* **Real-time Alerting**: Immediate security incident notification
* **Automated Response**: Automatic threat mitigation

### 3. Incident Response

#### Response Procedures

1. **Detection**: Automated and manual threat detection
2. **Assessment**: Impact and severity evaluation
3. **Containment**: Immediate threat isolation
4. **Eradication**: Complete threat removal
5. **Recovery**: Service restoration and validation
6. **Lessons Learned**: Process improvement and documentation

#### Escalation Matrix

| Severity | Response Time | Escalation Level |
| --- | --- | --- |
| Critical | 15 minutes | Executive Team |
| High | 1 hour | Security Team |
| Medium | 4 hours | Operations Team |
| Low | 24 hours | Support Team |

## Compliance and Governance

### 1. Regulatory Compliance

#### Standards and Frameworks

* **SOC 2 Type II**: Security, availability, and confidentiality controls
* **ISO 27001**: Information security management system
* **GDPR**: European data protection regulation compliance
* **CCPA**: California consumer privacy act compliance
* **HIPAA**: Healthcare information privacy compliance (if applicable)

#### Compliance Monitoring

* **Regular Audits**: Quarterly compliance assessments
* **Automated Checks**: Continuous compliance validation
* **Documentation**: Comprehensive compliance documentation
* **Training**: Regular staff compliance training

### 2. Governance Framework

#### Governance Structure

* **Security Council**: Executive security oversight
* **Technical Committee**: Technical security decisions
* **Compliance Officer**: Regulatory compliance management
* **Data Protection Officer**: Privacy and data protection oversight

#### Policy Management

* **Policy Development**: Formal policy creation process
* **Policy Approval**: Multi-level policy approval workflow
* **Policy Communication**: Regular policy updates and training
* **Policy Enforcement**: Automated policy compliance checking

### 3. Risk Management

#### Risk Assessment

* **Risk Identification**: Comprehensive risk catalog
* **Risk Analysis**: Impact and probability assessment
* **Risk Evaluation**: Risk tolerance and acceptance criteria
* **Risk Treatment**: Mitigation, transfer, or acceptance strategies

#### Risk Monitoring

* **Risk Dashboard**: Real-time risk visibility
* **Risk Metrics**: Quantitative risk measurements
* **Risk Reporting**: Regular risk status updates
* **Risk Reviews**: Periodic risk assessment reviews

## Audit and Logging

### 1. Comprehensive Logging

#### Log Categories

* **Authentication Events**: Login attempts, failures, and successes
* **Authorization Events**: Permission checks and access grants
* **Data Access Events**: Data retrieval and modification activities
* **System Events**: Application and infrastructure events
* **Security Events**: Security-related activities and violations

#### Log Format Standards

{  
 "timestamp": "2024-01-15T10:30:00Z",  
 "event\_type": "authentication",  
 "user\_id": "user@company.com",  
 "action": "login\_attempt",  
 "result": "success",  
 "source\_ip": "192.168.1.100",  
 "user\_agent": "Mozilla/5.0...",  
 "session\_id": "sess\_123456789",  
 "resource\_accessed": "/api/v1/diagnostics",  
 "risk\_score": 0.1  
}

### 2. Audit Trail Management

#### Audit Requirements

* **Immutable Logs**: Tamper-proof log storage
* **Long-term Retention**: Compliance-mandated retention periods
* **Secure Storage**: Encrypted audit log storage
* **Access Controls**: Restricted audit log access

#### Audit Analysis

* **Automated Analysis**: Real-time audit log analysis
* **Anomaly Detection**: Unusual pattern identification
* **Compliance Reporting**: Regular compliance reports
* **Forensic Analysis**: Detailed incident investigation capabilities

## Continuous Improvement

### 1. Security Metrics

#### Key Performance Indicators

* **Mean Time to Detection (MTTD)**: Average time to detect security incidents
* **Mean Time to Response (MTTR)**: Average time to respond to incidents
* **False Positive Rate**: Percentage of incorrect security alerts
* **Security Training Completion**: Percentage of staff completing security training
* **Vulnerability Remediation Time**: Time to fix identified vulnerabilities

#### Security Dashboards

* **Executive Dashboard**: High-level security metrics for leadership
* **Operational Dashboard**: Detailed metrics for security operations
* **Compliance Dashboard**: Regulatory compliance status
* **Risk Dashboard**: Current risk assessment and trends

### 2. Security Training and Awareness

#### Training Programs

* **General Security Awareness**: Basic security concepts for all staff
* **Technical Security Training**: Advanced training for technical staff
* **Incident Response Training**: Specific training for incident response teams
* **Compliance Training**: Regulatory compliance education

#### Awareness Activities

* **Phishing Simulations**: Regular phishing attack simulations
* **Security Newsletters**: Regular security updates and tips
* **Security Workshops**: Hands-on security training sessions
* **Security Champions**: Staff volunteers promoting security awareness

This comprehensive security and governance plan ensures that the Azure Agentic AI Support Bot operates with the highest standards of security, privacy, and compliance while maintaining effective support capabilities for Azure customers.

# Observability & metrics

# Observability and Metrics - Azure AI Support Bot

## Overview

This document outlines the comprehensive observability framework for the Azure Agentic AI Support Bot, including monitoring, logging, metrics, alerting, and performance tracking to ensure optimal system performance and customer experience.

## Observability Architecture

### 1. Multi-Layer Monitoring

┌─────────────────────────────────────────────────────────────┐  
│ Observability Stack │  
├─────────────────────────────────────────────────────────────┤  
│ Application Layer: Custom Metrics & Business KPIs │  
│ Service Layer: Azure Monitor & Application Insights │  
│ Infrastructure Layer: Resource Health & Performance │  
│ Security Layer: Security Events & Threat Detection │  
│ User Experience Layer: Customer Satisfaction Metrics │  
└─────────────────────────────────────────────────────────────┘

### 2. Monitoring Pillars

* **Metrics**: Quantitative measurements of system performance
* **Logs**: Detailed event records for debugging and analysis
* **Traces**: Request flow tracking across distributed components
* **Alerts**: Proactive notification of issues and anomalies
* **Dashboards**: Visual representation of system health and performance

## Application Performance Monitoring

### 1. Custom Business Metrics

#### Agent Performance Metrics

# Agent Response Time  
agent\_response\_time = Histogram(  
 name="agent\_response\_time\_seconds",  
 documentation="Time taken by agents to respond",  
 labelnames=["agent\_type", "operation\_type"]  
)  
  
# Diagnostic Accuracy  
diagnostic\_accuracy = Gauge(  
 name="diagnostic\_accuracy\_percentage",  
 documentation="Percentage of correct diagnostic results"  
)  
  
# Resolution Success Rate  
resolution\_success\_rate = Gauge(  
 name="resolution\_success\_rate\_percentage",  
 documentation="Percentage of successful resolutions"  
)  
  
# Conversation Quality Score  
conversation\_quality = Histogram(  
 name="conversation\_quality\_score",  
 documentation="Quality score of conversations (0-100)"  
)

#### Customer Experience Metrics

# Time to Resolution  
time\_to\_resolution = Histogram(  
 name="time\_to\_resolution\_minutes",  
 documentation="Time from issue report to resolution",  
 buckets=[5, 15, 30, 60, 120, 300, 600]  
)  
  
# First Call Resolution Rate  
first\_call\_resolution = Counter(  
 name="first\_call\_resolutions\_total",  
 documentation="Total number of first-call resolutions",  
 labelnames=["issue\_type"]  
)  
  
# Customer Satisfaction Score  
customer\_satisfaction = Gauge(  
 name="customer\_satisfaction\_score",  
 documentation="Average customer satisfaction score (1-5)"  
)

### 2. System Performance Metrics

#### Azure Functions Performance

# Function Execution Time  
function\_execution\_time = Histogram(  
 name="function\_execution\_time\_seconds",  
 documentation="Azure Functions execution time",  
 labelnames=["function\_name", "trigger\_type"]  
)  
  
# Function Invocation Count  
function\_invocations = Counter(  
 name="function\_invocations\_total",  
 documentation="Total function invocations",  
 labelnames=["function\_name", "status"]  
)  
  
# Cold Start Frequency  
cold\_starts = Counter(  
 name="function\_cold\_starts\_total",  
 documentation="Number of function cold starts"  
)

#### Azure OpenAI Performance

# OpenAI API Response Time  
openai\_response\_time = Histogram(  
 name="openai\_response\_time\_seconds",  
 documentation="Azure OpenAI API response time",  
 labelnames=["model", "operation\_type"]  
)  
  
# Token Usage  
token\_usage = Counter(  
 name="openai\_tokens\_used\_total",  
 documentation="Total tokens consumed",  
 labelnames=["model", "token\_type"]  
)  
  
# API Error Rate  
openai\_error\_rate = Counter(  
 name="openai\_errors\_total",  
 documentation="OpenAI API errors",  
 labelnames=["error\_type", "model"]  
)

## Logging Strategy

### 1. Structured Logging

#### Log Levels and Usage

* **ERROR**: System errors, failed operations, critical issues
* **WARN**: Warning conditions, degraded performance, potential issues
* **INFO**: General information, successful operations, important events
* **DEBUG**: Detailed diagnostic information, development debugging
* **TRACE**: Very detailed diagnostic information, request tracing

#### Log Format Standards

{  
 "timestamp": "2024-01-15T10:30:00.123Z",  
 "level": "INFO",  
 "service": "azure-ai-support-bot",  
 "component": "diagnostic-agent",  
 "operation": "analyze\_rdp\_issue",  
 "trace\_id": "trace\_123456789",  
 "span\_id": "span\_987654321",  
 "user\_id": "user@company.com",  
 "conversation\_id": "conv\_abc123",  
 "vm\_name": "test-vm-01",  
 "resource\_group": "test-rg",  
 "message": "Diagnostic analysis completed successfully",  
 "duration\_ms": 1250,  
 "diagnostic\_results": {  
 "issues\_found": 2,  
 "root\_causes": ["nsg\_blocking", "firewall\_disabled"],  
 "confidence\_score": 0.95  
 },  
 "metadata": {  
 "azure\_region": "eastus",  
 "subscription\_id": "sub\_12345"  
 }  
}

### 2. Log Categories

#### Application Logs

* **Agent Operations**: Agent execution and decision-making processes
* **Diagnostic Results**: Detailed diagnostic analysis and findings
* **Resolution Actions**: Automated and manual resolution activities
* **Conversation Flow**: User interactions and conversation management

#### System Logs

* **Infrastructure Events**: Azure resource health and performance
* **Security Events**: Authentication, authorization, and security incidents
* **Performance Logs**: System performance and resource utilization
* **Error Logs**: System errors, exceptions, and failure conditions

#### Business Logs

* **Customer Interactions**: Customer engagement and satisfaction metrics
* **Issue Resolution**: Problem resolution tracking and outcomes
* **Knowledge Base**: Learning and knowledge management activities
* **Compliance**: Regulatory compliance and audit activities

## Distributed Tracing

### 1. Trace Configuration

#### Trace Headers

# Trace context propagation  
trace\_headers = {  
 "traceparent": "00-{trace\_id}-{span\_id}-01",  
 "tracestate": "azure-ai-support-bot=1"  
}

#### Span Creation

from opentelemetry import trace  
  
tracer = trace.get\_tracer(\_\_name\_\_)  
  
@tracer.start\_as\_current\_span("diagnostic\_analysis")  
def analyze\_rdp\_issue(vm\_info, user\_context):  
 with tracer.start\_as\_current\_span("vm\_health\_check") as span:  
 span.set\_attribute("vm.name", vm\_info["vm\_name"])  
 span.set\_attribute("vm.resource\_group", vm\_info["resource\_group"])  
 # VM health check logic  
 result = check\_vm\_health(vm\_info)  
 span.set\_attribute("vm.status", result["status"])  
 return result

### 2. Trace Analysis

#### Key Trace Metrics

* **Request Duration**: End-to-end request processing time
* **Service Dependencies**: External service call patterns
* **Error Propagation**: Error flow through distributed components
* **Performance Bottlenecks**: Slowest components and operations

#### Trace Visualization

* **Service Map**: Visual representation of service interactions
* **Waterfall Diagrams**: Request flow timing visualization
* **Error Analysis**: Failed request investigation
* **Performance Analysis**: Bottleneck identification

## Alerting Strategy

### 1. Alert Categories

#### Critical Alerts (Immediate Response)

# System Down Alert  
- name: "System Unavailable"  
 condition: "availability < 95%"  
 duration: "2 minutes"  
 severity: "critical"  
 escalation: "immediate"  
  
# High Error Rate Alert  
- name: "High Error Rate"  
 condition: "error\_rate > 10%"  
 duration: "5 minutes"  
 severity: "critical"  
 escalation: "immediate"

#### Warning Alerts (Response within 1 hour)

# Performance Degradation Alert  
- name: "Performance Degradation"  
 condition: "response\_time > 30s"  
 duration: "10 minutes"  
 severity: "warning"  
 escalation: "1 hour"  
  
# Resource Utilization Alert  
- name: "High Resource Utilization"  
 condition: "cpu\_usage > 80% OR memory\_usage > 85%"  
 duration: "15 minutes"  
 severity: "warning"  
 escalation: "1 hour"

#### Info Alerts (Monitor and Log)

# Unusual Pattern Alert  
- name: "Unusual Usage Pattern"  
 condition: "request\_count > 2x normal"  
 duration: "30 minutes"  
 severity: "info"  
 escalation: "monitor"

### 2. Alert Channels

#### Notification Methods

* **Email**: Detailed alert information and context
* **SMS**: Critical alerts requiring immediate attention
* **Teams/Slack**: Team collaboration and incident management
* **PagerDuty**: On-call escalation and rotation management

#### Alert Enrichment

* **Context Information**: Relevant system state and metrics
* **Runbook Links**: Automated resolution procedures
* **Historical Data**: Previous similar incidents and resolutions
* **Escalation Path**: Clear escalation procedures and contacts

## Dashboard Design

### 1. Executive Dashboard

#### Key Performance Indicators

{  
 "dashboard\_title": "Azure AI Support Bot - Executive View",  
 "refresh\_interval": "5 minutes",  
 "widgets": [  
 {  
 "type": "metric",  
 "title": "System Availability",  
 "metric": "availability\_percentage",  
 "threshold": 99.9  
 },  
 {  
 "type": "metric",  
 "title": "Customer Satisfaction",  
 "metric": "customer\_satisfaction\_score",  
 "threshold": 4.5  
 },  
 {  
 "type": "metric",  
 "title": "Resolution Success Rate",  
 "metric": "resolution\_success\_rate",  
 "threshold": 85  
 },  
 {  
 "type": "metric",  
 "title": "Average Resolution Time",  
 "metric": "avg\_resolution\_time\_minutes",  
 "threshold": 15  
 }  
 ]  
}

### 2. Operations Dashboard

#### Detailed Metrics

* **Request Volume**: Requests per minute/hour/day
* **Response Times**: P50, P90, P95, P99 response times
* **Error Rates**: Error percentage by component
* **Resource Utilization**: CPU, memory, storage usage
* **Active Conversations**: Current active user sessions

### 3. Security Dashboard

#### Security Metrics

* **Authentication Events**: Login attempts and failures
* **Authorization Violations**: Permission denied events
* **Content Safety Flags**: Blocked content and safety violations
* **Threat Detection**: Security alerts and incidents

### 4. Business Intelligence Dashboard

#### Business Metrics

* **Issue Categories**: Most common issue types
* **Resolution Patterns**: Common resolution approaches
* **Customer Segments**: Usage patterns by customer type
* **Cost Analysis**: Resource costs and optimization opportunities

## Performance Optimization

### 1. Performance Monitoring

#### Key Performance Indicators

* **Throughput**: Requests per second
* **Latency**: Response time percentiles
* **Error Rate**: Percentage of failed requests
* **Resource Efficiency**: Cost per request

#### Performance Baselines

# Performance benchmarks  
PERFORMANCE\_TARGETS = {  
 "response\_time\_p95": 30, # seconds  
 "availability": 99.9, # percentage  
 "error\_rate": 0.1, # percentage  
 "throughput": 100, # requests per second  
 "cost\_per\_request": 0.01 # USD  
}

### 2. Performance Analysis

#### Bottleneck Identification

* **CPU Bottlenecks**: High CPU utilization patterns
* **Memory Bottlenecks**: Memory usage and garbage collection
* **Network Bottlenecks**: Network latency and bandwidth
* **Database Bottlenecks**: Query performance and connection pooling

#### Optimization Strategies

* **Caching**: Intelligent caching of frequently accessed data
* **Connection Pooling**: Efficient resource utilization
* **Async Processing**: Non-blocking operations
* **Load Balancing**: Even distribution of requests

## Capacity Planning

### 1. Resource Scaling

#### Auto-Scaling Rules

# Azure Functions Scaling  
scaling\_rules:  
 - metric: "cpu\_percentage"  
 threshold: 70  
 scale\_out\_cooldown: "5 minutes"  
 scale\_in\_cooldown: "10 minutes"  
   
 - metric: "memory\_percentage"  
 threshold: 80  
 scale\_out\_cooldown: "3 minutes"  
 scale\_in\_cooldown: "15 minutes"  
  
# Cosmos DB Scaling  
cosmos\_db\_scaling:  
 - metric: "request\_units\_consumed"  
 threshold: 80  
 auto\_scale: true  
 max\_throughput: 10000

#### Capacity Forecasting

* **Historical Analysis**: Past usage patterns and trends
* **Growth Projections**: Expected future growth and demand
* **Seasonal Patterns**: Usage variations over time
* **Event-Based Scaling**: Scaling for planned events or promotions

### 2. Cost Optimization

#### Cost Monitoring

* **Resource Costs**: Individual service costs and trends
* **Cost per Transaction**: Cost efficiency metrics
* **Waste Identification**: Underutilized resources
* **Optimization Opportunities**: Cost reduction recommendations

#### Cost Controls

* **Budget Alerts**: Spending threshold notifications
* **Resource Tagging**: Cost allocation and tracking
* **Automated Shutdown**: Non-production resource management
* **Right-Sizing**: Optimal resource sizing recommendations

## Continuous Improvement

### 1. Metrics Analysis

#### Trend Analysis

* **Performance Trends**: Long-term performance patterns
* **Usage Trends**: User behavior and adoption patterns
* **Error Trends**: Error pattern analysis and reduction
* **Cost Trends**: Cost optimization opportunities

#### Anomaly Detection

* **Statistical Analysis**: Deviation from normal patterns
* **Machine Learning**: AI-powered anomaly detection
* **Threshold-Based**: Rule-based anomaly detection
* **Contextual Analysis**: Anomaly significance assessment

### 2. Feedback Loops

#### Monitoring Feedback

* **Alert Effectiveness**: Alert accuracy and relevance
* **Dashboard Usage**: Dashboard effectiveness and adoption
* **Response Time**: Alert response and resolution times
* **User Satisfaction**: Monitoring system user feedback

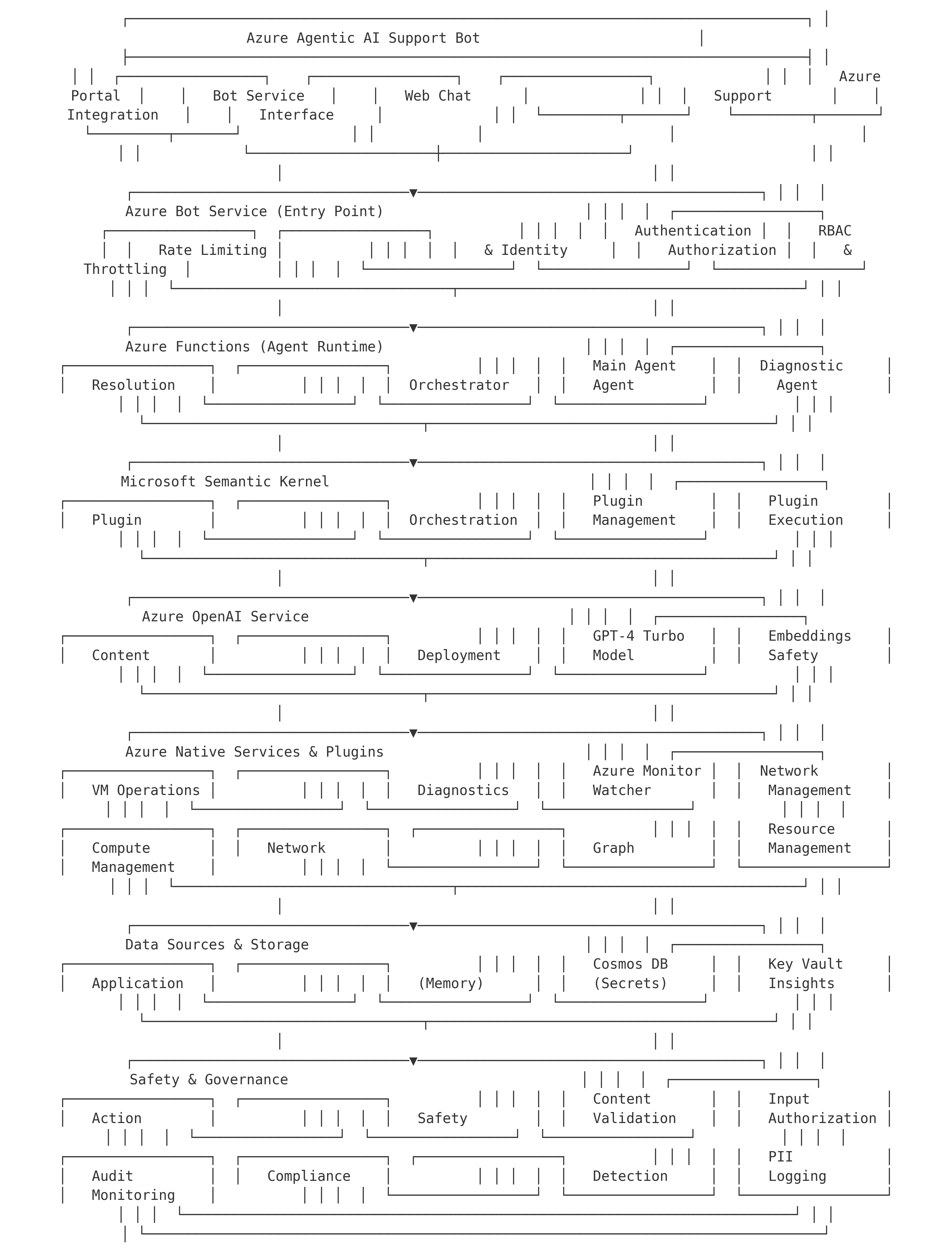
#### Continuous Optimization

* **Metric Refinement**: Improved metric definitions and calculations
* **Alert Tuning**: Optimized alert thresholds and conditions
* **Dashboard Updates**: Enhanced visualization and insights
* **Process Improvement**: Streamlined monitoring and response processes

This comprehensive observability framework ensures that the Azure Agentic AI Support Bot operates with optimal performance, reliability, and customer satisfaction while providing deep insights into system behavior and business outcomes.

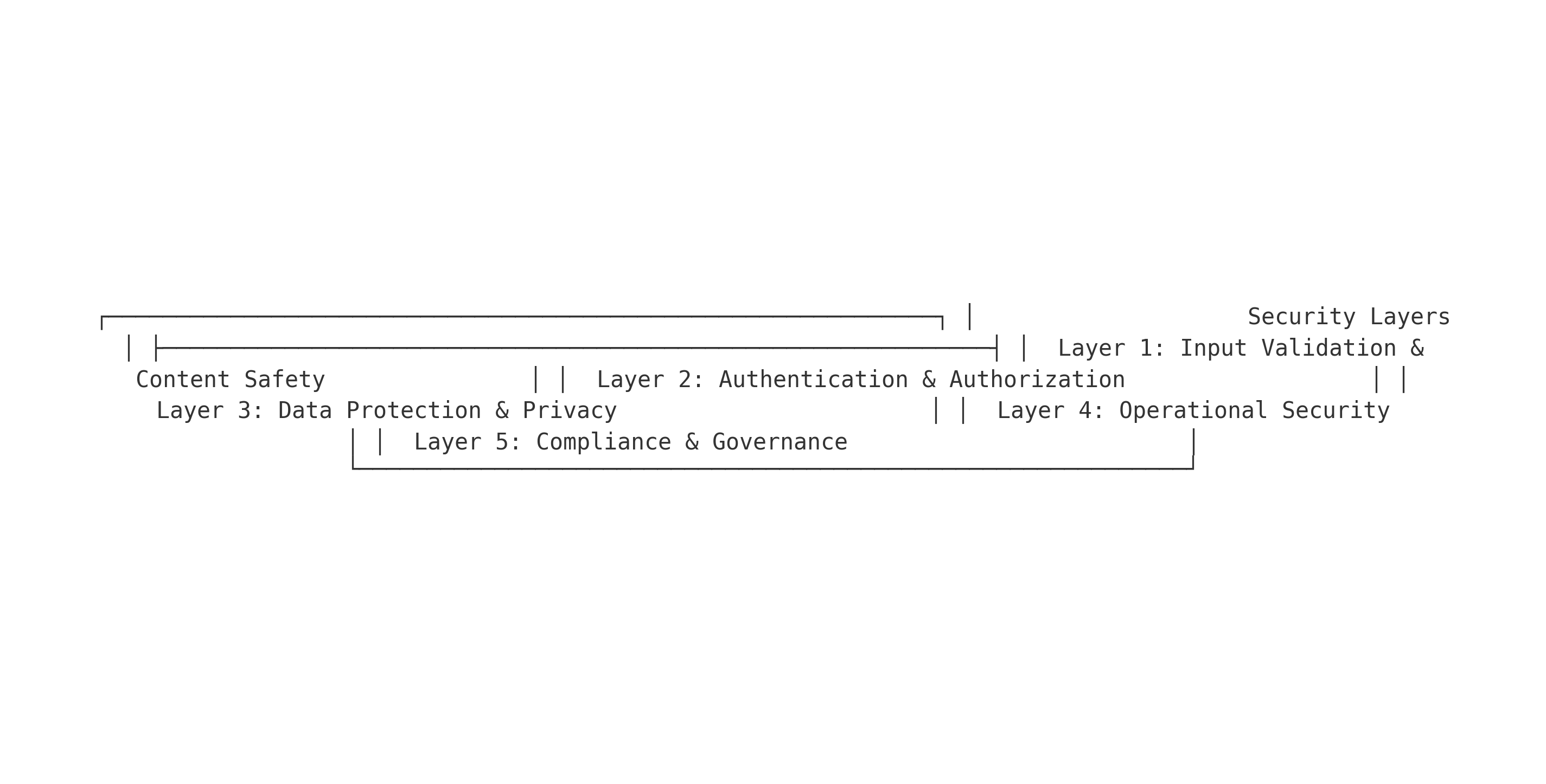
Rendered Architecture Diagrams

Architecture



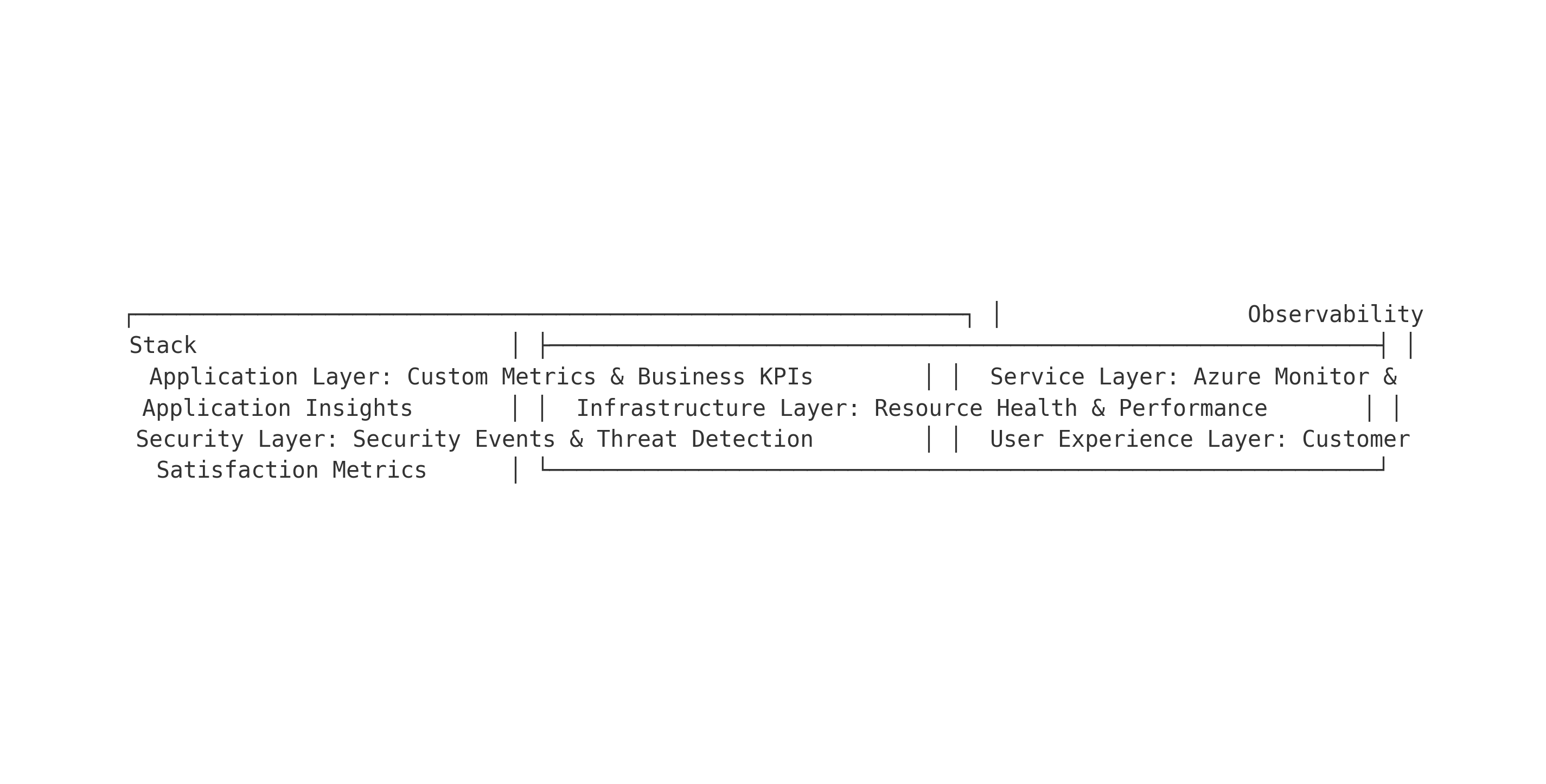
Auto-rendered from Markdown ASCII diagram for professional visualization.

Security Governance



Auto-rendered from Markdown ASCII diagram for professional visualization.

Observability



Auto-rendered from Markdown ASCII diagram for professional visualization.