[COMPANY LOGO]

Application Intelligence Report

Comprehensive Analysis and Migration Assessment

Repository: https://github.com/end-of-game/openshift-voting-app

Analysis Date: July 18, 2025

*Generated by Application Intelligence Platform*

Table of Contents

Executive Summary........................................... 3

Application Overview........................................ 4

Component Analysis.......................................... 5

Architecture Assessment..................................... 8

Security Analysis........................................... 10

Git History Analysis........................................ 12

Recommendations............................................. 13

Appendices.................................................. 15

Executive Summary

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Total Components | 3 |
| Programming Languages | python, java, nodejs |
| Containerization Status | 3 containerized |
| Data Sources | 0 |
| Security Findings | 0 |
| Git Commits | 1 |
| Architecture Style | microservices |

Application Overview

This report presents a comprehensive analysis of the application repository. The analysis identified 3 components using 3 different programming languages. The application demonstrates a microservices architecture pattern.

Key Findings

• 📦 3 application components identified

• 🔧 3 programming languages detected: python, java, nodejs

• 🐳 3 components are containerized

• 💾 0 data sources identified

• 🔒 0 security findings require attention

Detailed Analysis

Component Analysis

The analysis identified 3 components across the application:

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Language** | **Type** | **Packaging** |
| vote | python | Unknown | docker |
| worker | java | Unknown | docker |
| result | nodejs | Unknown | docker |

Component: vote

• Language: python

• Runtime: python

• Build Tool: unknown

• Packaging: docker

• Exposed Ports: 8080

• Base Images: python:3.9-slim

Component: worker

• Language: java

• Runtime: java

• Build Tool: unknown

• Packaging: docker

• Base Images: maven:3.5-jdk-8-alpine, openjdk:8-jre

**Notes:**

• Multiple base images detected: maven:3.5-jdk-8-alpine, openjdk:8-jre. This may indicate multi-stage builds or alternative build strategies.

Component: result

• Language: nodejs

• Runtime: nodejs

• Build Tool: unknown

• Packaging: docker

• Exposed Ports: 8080

• Base Images: node:10-slim

Architecture Analysis

Architecture Style: microservices (Confidence: ConfidenceLevel.HIGH)

Reasoning: Multiple components with independent deployment characteristics

**Evidence:**

• Found 3 components

• Multiple deployable components detected

• 3 containerized components

• Multiple deployment configurations

Security Analysis

Security analysis identified 2 findings with 3 base image risks.

**Key Security Findings:**

• Unknown: The 'result' component uses 'node:10-slim' as its base image. Node.js 10 is past End-of-Life (EOL) and contains numerous unpatched vulnerabilities, posing a severe security risk. (Severity: CRITICAL)

• Unknown: The 'worker' component uses 'maven:3.5-jdk-8-alpine' as its base image. Both Maven 3.5 and OpenJDK 8 contain known vulnerabilities and are considered outdated, exposing the application to potential exploits. (Severity: HIGH)

Git History Analysis

• Total Commits: 1

• Active Contributors: 0

• Recent Activity: inactive

• Code Stability: high

Recommendations

🔴 High Priority Recommendations

• 🔒 Security: 2 critical/high severity vulnerabilities found. Prioritize security remediation.

🟢 Low Priority Recommendations

• 📊 Development Activity: Low recent activity detected. Consider reviewing development processes and team capacity.

• 🐳 Base Images: 3 base images have known risks. Update to more recent versions.

Appendices

Appendix A: Technical Details

This analysis was generated using the Application Intelligence Platform, which performs comprehensive analysis of application repositories including code structure, infrastructure configuration, and security assessment.

Appendix B: Analysis Methodology

• Component Discovery: Automated scanning of source code and configuration files

• Language Detection: Analysis of file extensions, build configurations, and base images

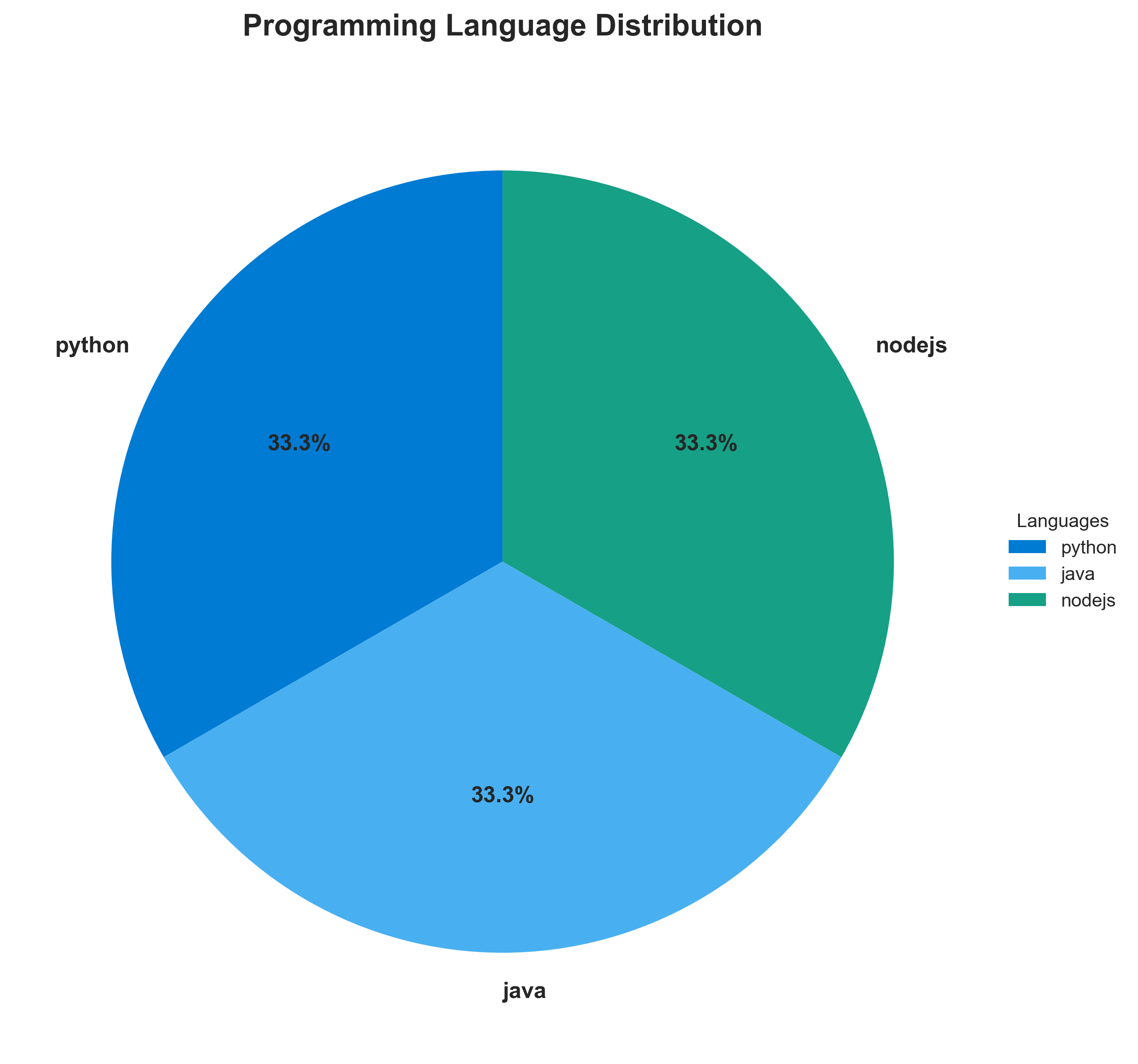
• Architecture Assessment: Evaluation of deployment patterns and component relationships

• Security Analysis: Scanning for common vulnerabilities and configuration issues

• Git History Analysis: Examination of commit patterns and development activity

Charts and Visualizations

Programming Language Distribution



**📊 Context:** Shows the programming languages used across 3 components.

**📊 Key Insights:** Primary languages: python, java, nodejs. Language diversity indicates technology stack complexity.

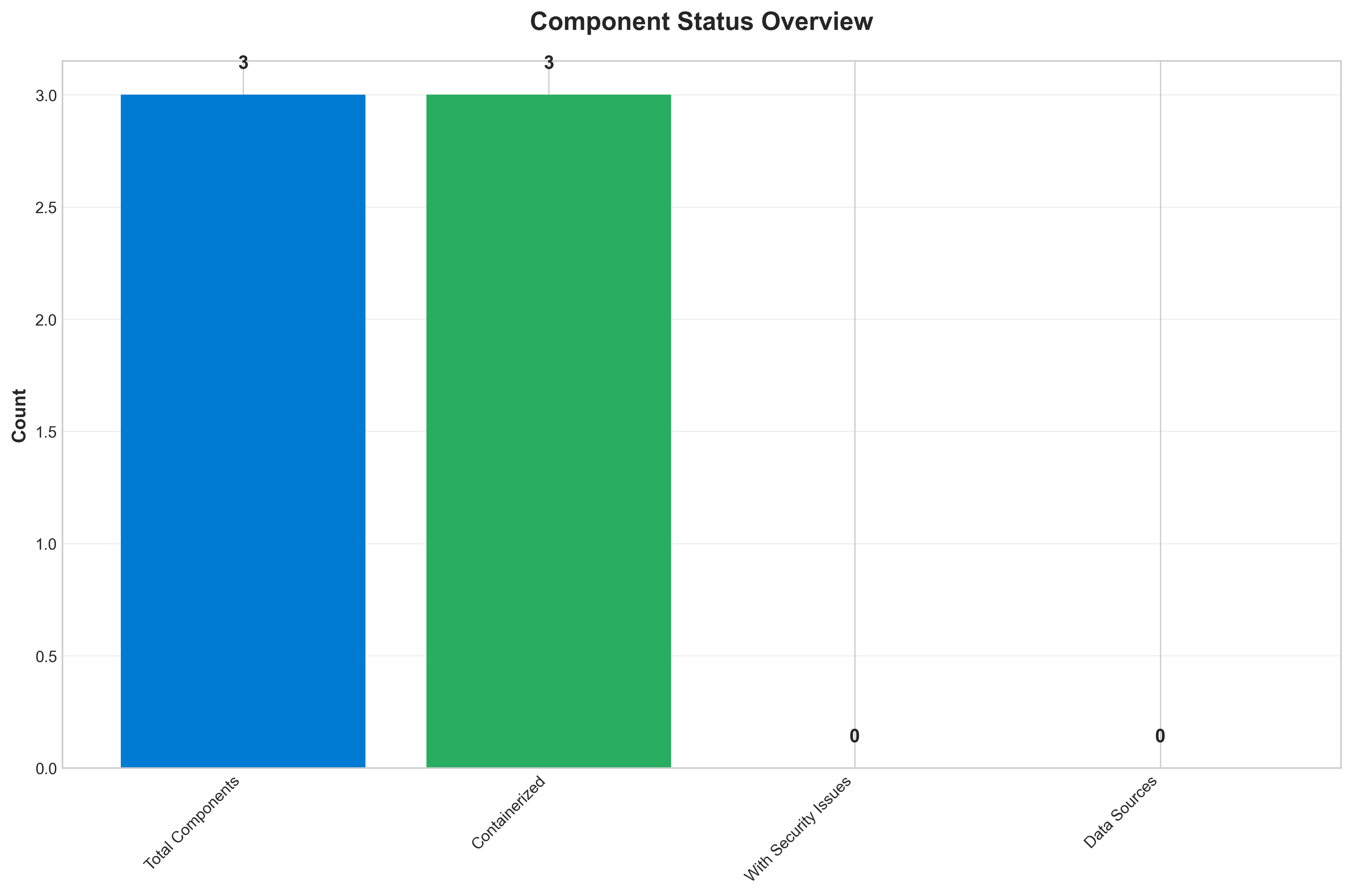
**📊 Business Impact:** Language diversity affects maintenance costs, hiring, and technical debt.

**📊 Recommendations:** Consider standardizing on fewer languages for better maintainability.

**📊 Technical Details:** Languages detected: python, java, nodejs. Analysis based on source code and build configurations.

────────────────────────────────────────────────────────────────────────────────

Component Status Overview



**📊 Context:** Shows status of 3 application components.

**📊 Key Insights:** 3 of 3 components are containerized. Fully ready for cloud deployment.

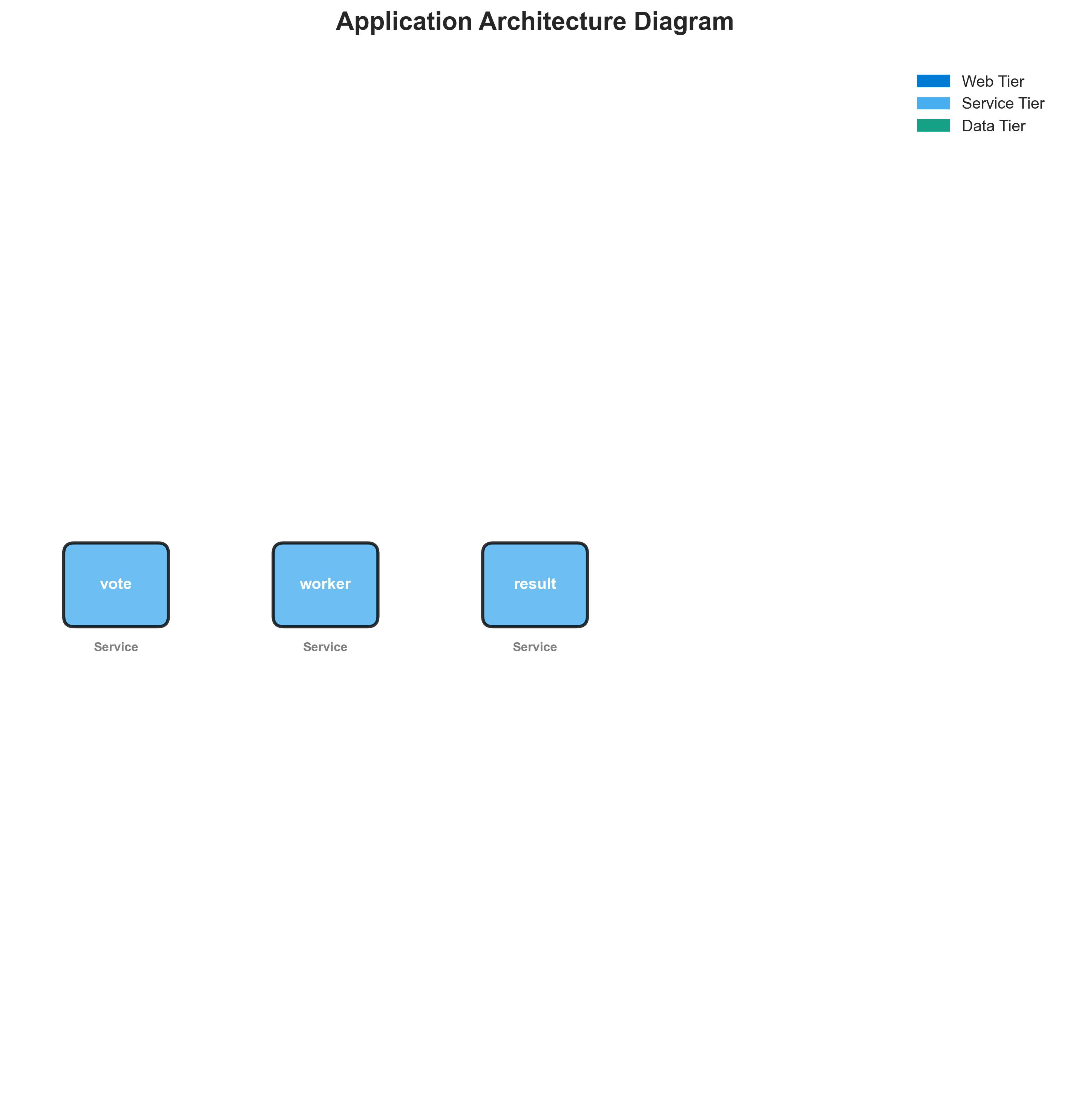
**📊 Business Impact:** Containerization status directly impacts cloud migration readiness and deployment flexibility.

**📊 Recommendations:** All components are containerized. Focus on optimizing container configurations and deployment strategies.

**📊 Technical Details:** Containerization rate: 3/3. Analysis includes Docker and orchestration configurations.

────────────────────────────────────────────────────────────────────────────────

Application Architecture



**📊 Context:** This Architecture Diagram shows application architecture insights.

**📊 Key Insights:** Analysis of 3 components reveals system characteristics.

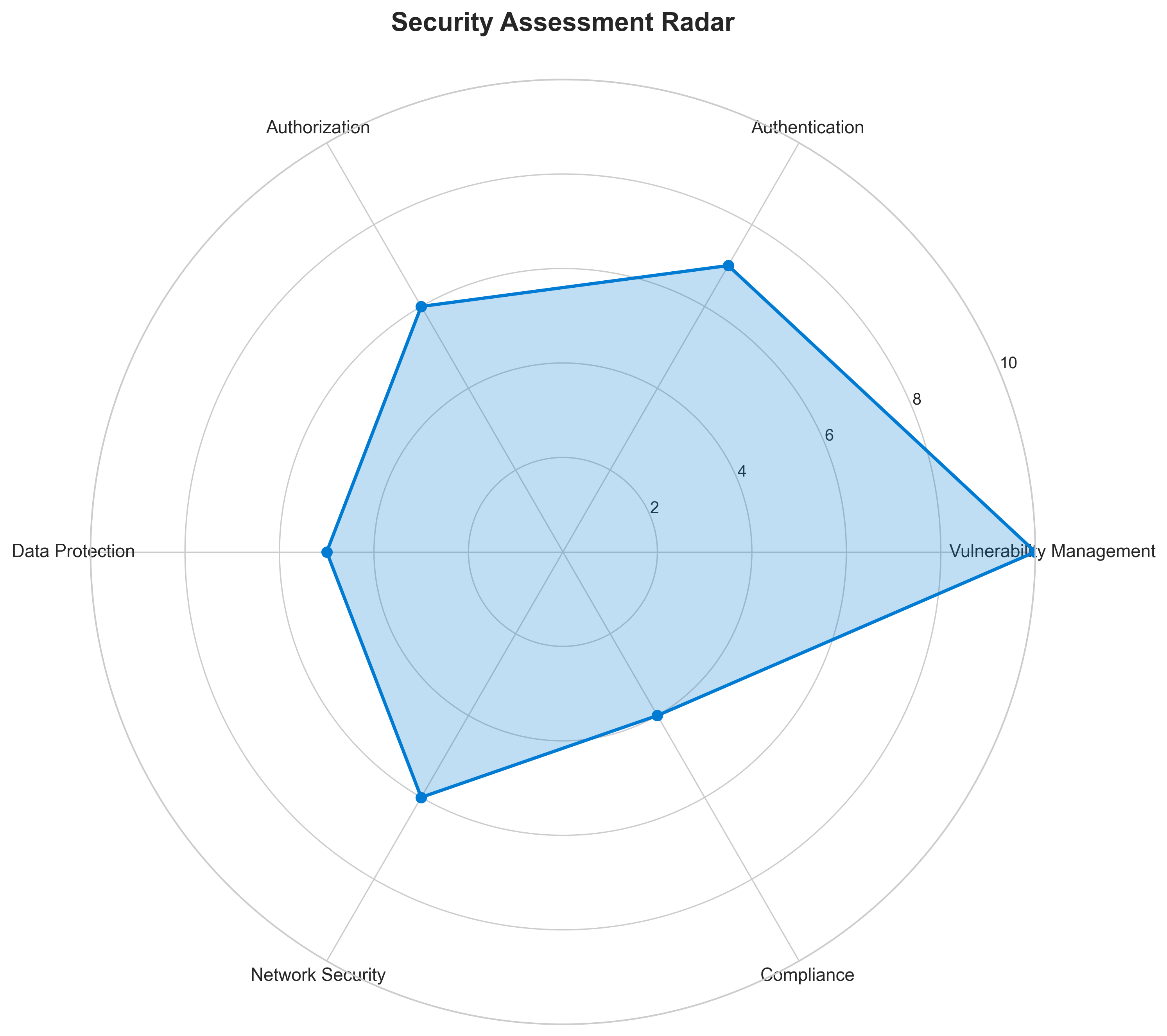
**📊 Business Impact:** Understanding system architecture supports better planning and decision-making.

**📊 Recommendations:** Review analysis results with technical teams for action planning.

**📊 Technical Details:** Diagram type: Architecture Diagram. Based on comprehensive application analysis.

────────────────────────────────────────────────────────────────────────────────

Security Assessment Radar



**📊 Context:** Security analysis identified 3 potential issues.

**📊 Key Insights:** Security posture assessment shows 3 findings requiring attention.

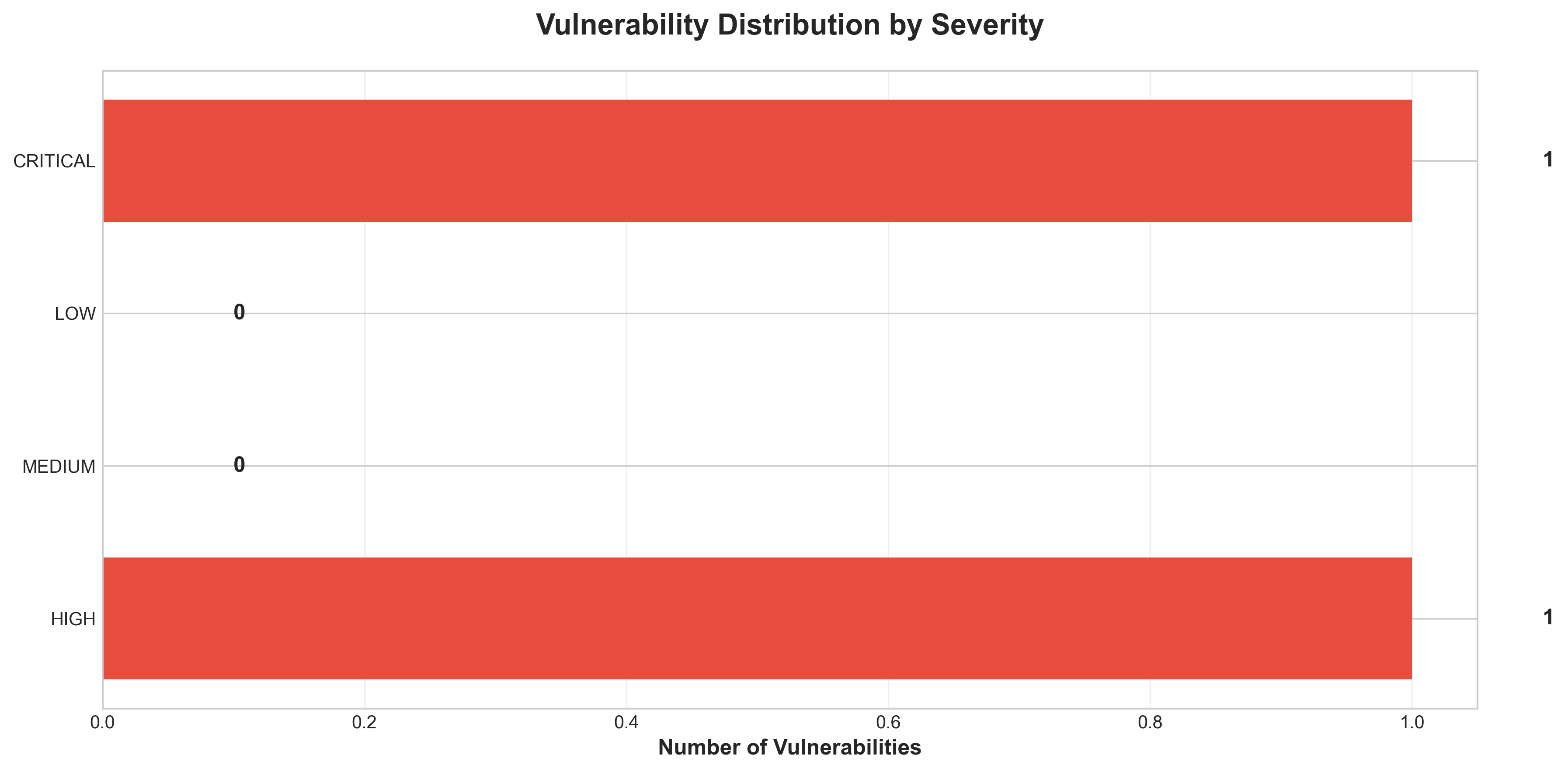
**📊 Business Impact:** Security findings represent potential risks that could impact system reliability and compliance.

**📊 Recommendations:** Address high-severity findings first, then implement security scanning in CI/CD pipeline.

**📊 Technical Details:** Findings: 3. Analysis includes base image vulnerabilities and code patterns.

────────────────────────────────────────────────────────────────────────────────

Vulnerability Analysis



**📊 Context:** This Vulnerability Timeline shows application architecture insights.

**📊 Key Insights:** Analysis of 3 components reveals system characteristics.

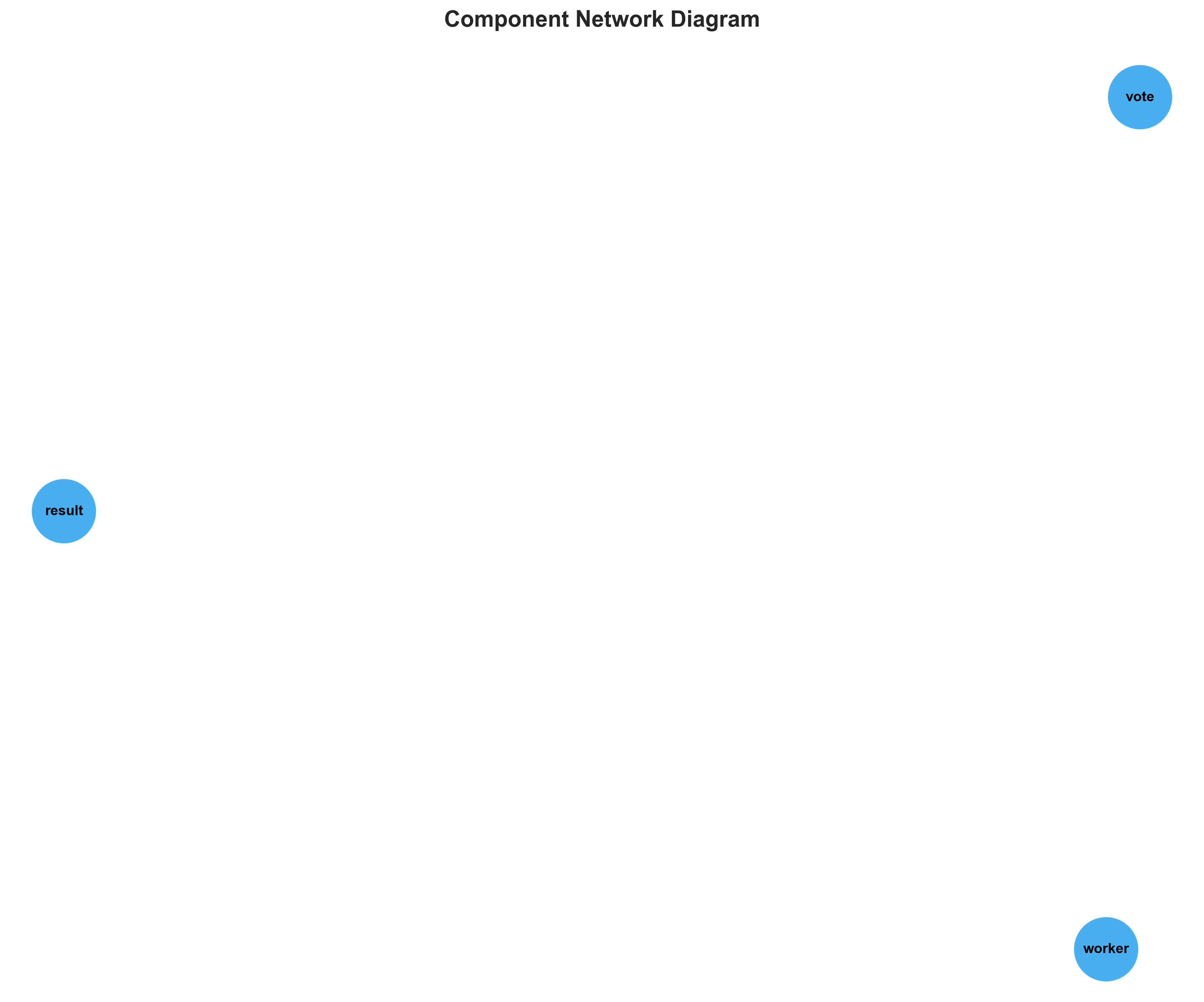
**📊 Business Impact:** Understanding system architecture supports better planning and decision-making.

**📊 Recommendations:** Review analysis results with technical teams for action planning.

**📊 Technical Details:** Diagram type: Vulnerability Timeline. Based on comprehensive application analysis.

────────────────────────────────────────────────────────────────────────────────

Component Network Topology



**📊 Context:** Shows status of 3 application components.

**📊 Key Insights:** 3 of 3 components are containerized. Fully ready for cloud deployment.

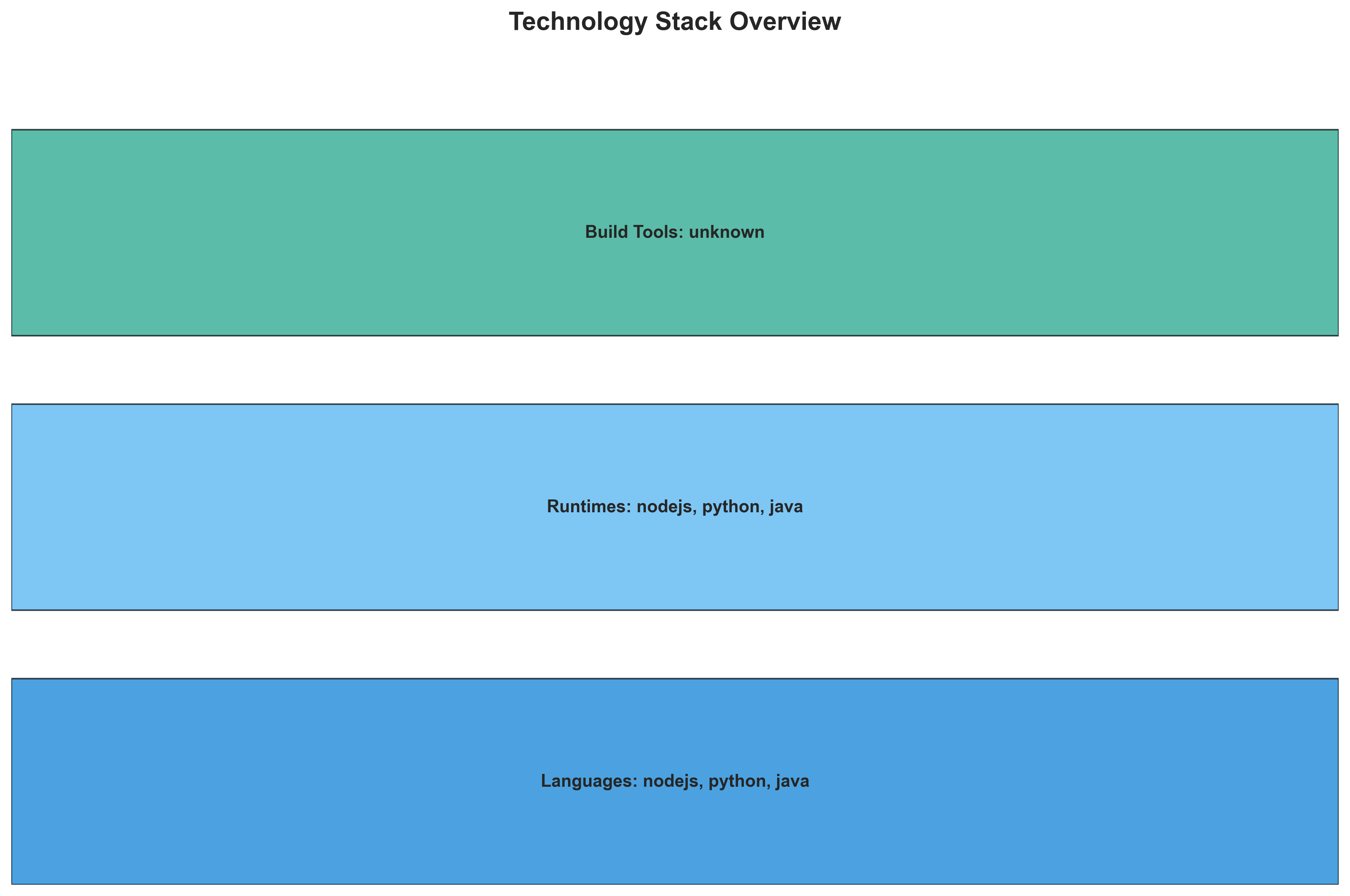
**📊 Business Impact:** Containerization status directly impacts cloud migration readiness and deployment flexibility.

**📊 Recommendations:** All components are containerized. Focus on optimizing container configurations and deployment strategies.

**📊 Technical Details:** Containerization rate: 3/3. Analysis includes Docker and orchestration configurations.

────────────────────────────────────────────────────────────────────────────────

Technology Stack



**📊 Context:** This Technology Stack shows application architecture insights.

**📊 Key Insights:** Analysis of 3 components reveals system characteristics.

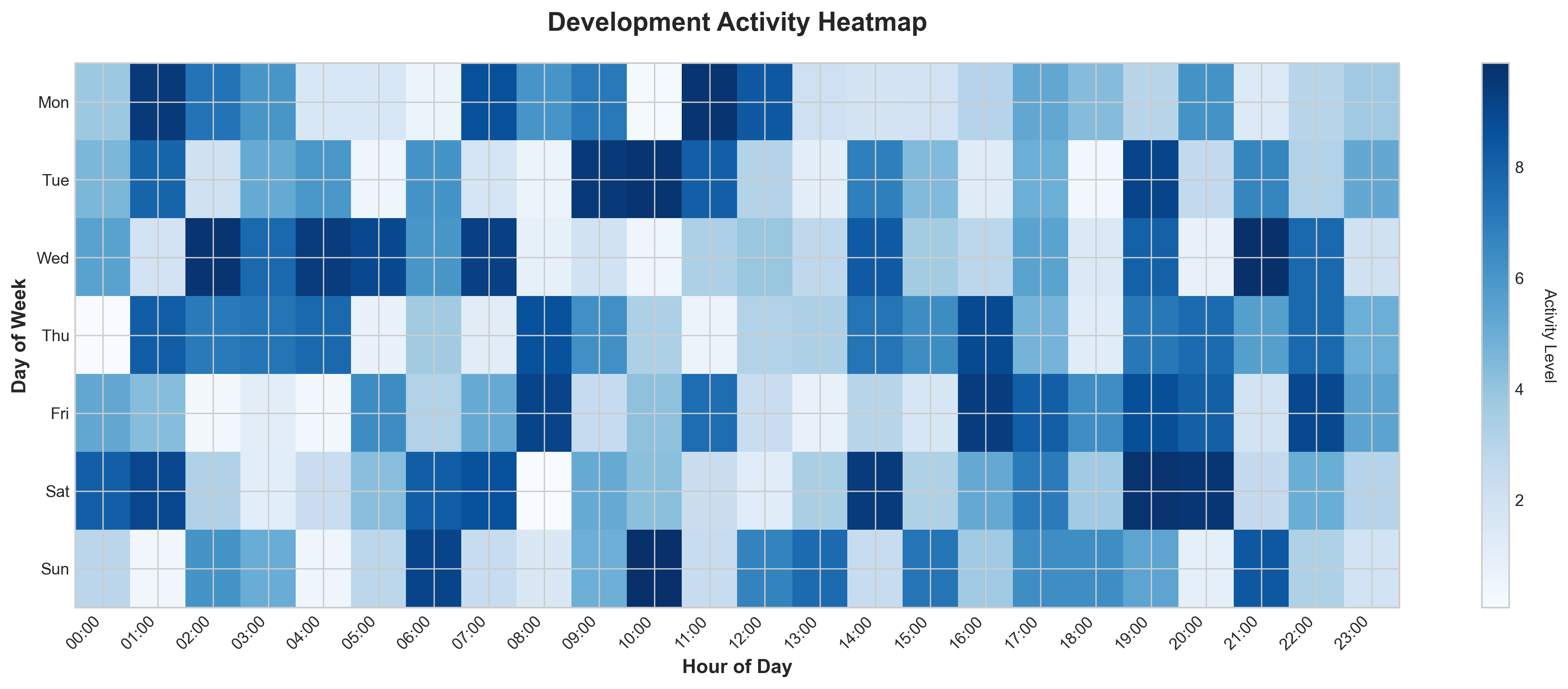
**📊 Business Impact:** Understanding system architecture supports better planning and decision-making.

**📊 Recommendations:** Review analysis results with technical teams for action planning.

**📊 Technical Details:** Diagram type: Technology Stack. Based on comprehensive application analysis.

────────────────────────────────────────────────────────────────────────────────

Development Activity Heatmap



**📊 Context:** This Development Activity shows application architecture insights.

**📊 Key Insights:** Analysis of 3 components reveals system characteristics.

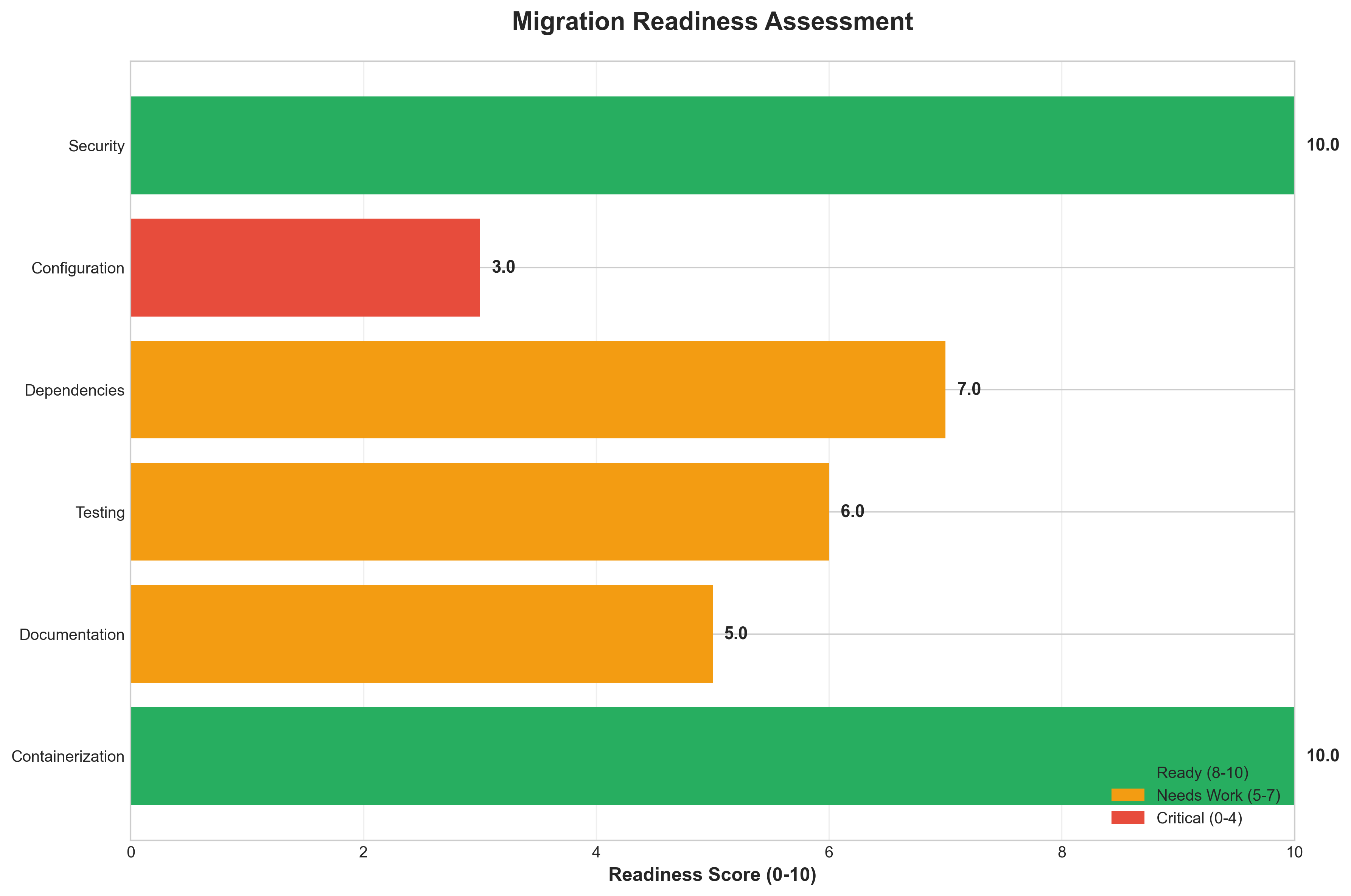
**📊 Business Impact:** Understanding system architecture supports better planning and decision-making.

**📊 Recommendations:** Review analysis results with technical teams for action planning.

**📊 Technical Details:** Diagram type: Development Activity. Based on comprehensive application analysis.

────────────────────────────────────────────────────────────────────────────────

Migration Readiness Assessment



**📊 Context:** This Migration Readiness shows application architecture insights.

**📊 Key Insights:** Analysis of 3 components reveals system characteristics.

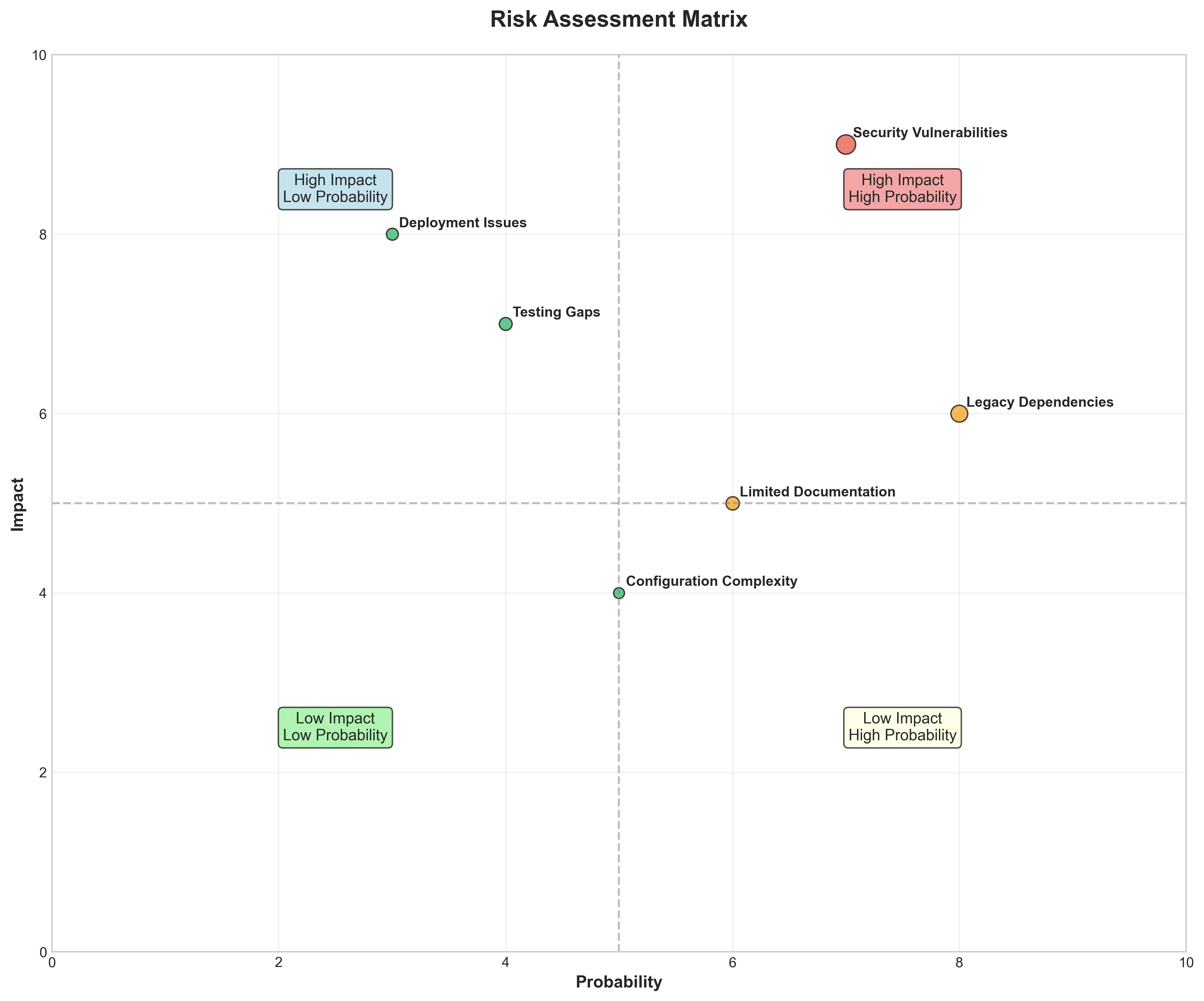
**📊 Business Impact:** Understanding system architecture supports better planning and decision-making.

**📊 Recommendations:** Review analysis results with technical teams for action planning.

**📊 Technical Details:** Diagram type: Migration Readiness. Based on comprehensive application analysis.

────────────────────────────────────────────────────────────────────────────────

Risk Assessment Matrix



**📊 Context:** This Risk Assessment Matrix shows application architecture insights.

**📊 Key Insights:** Analysis of 3 components reveals system characteristics.

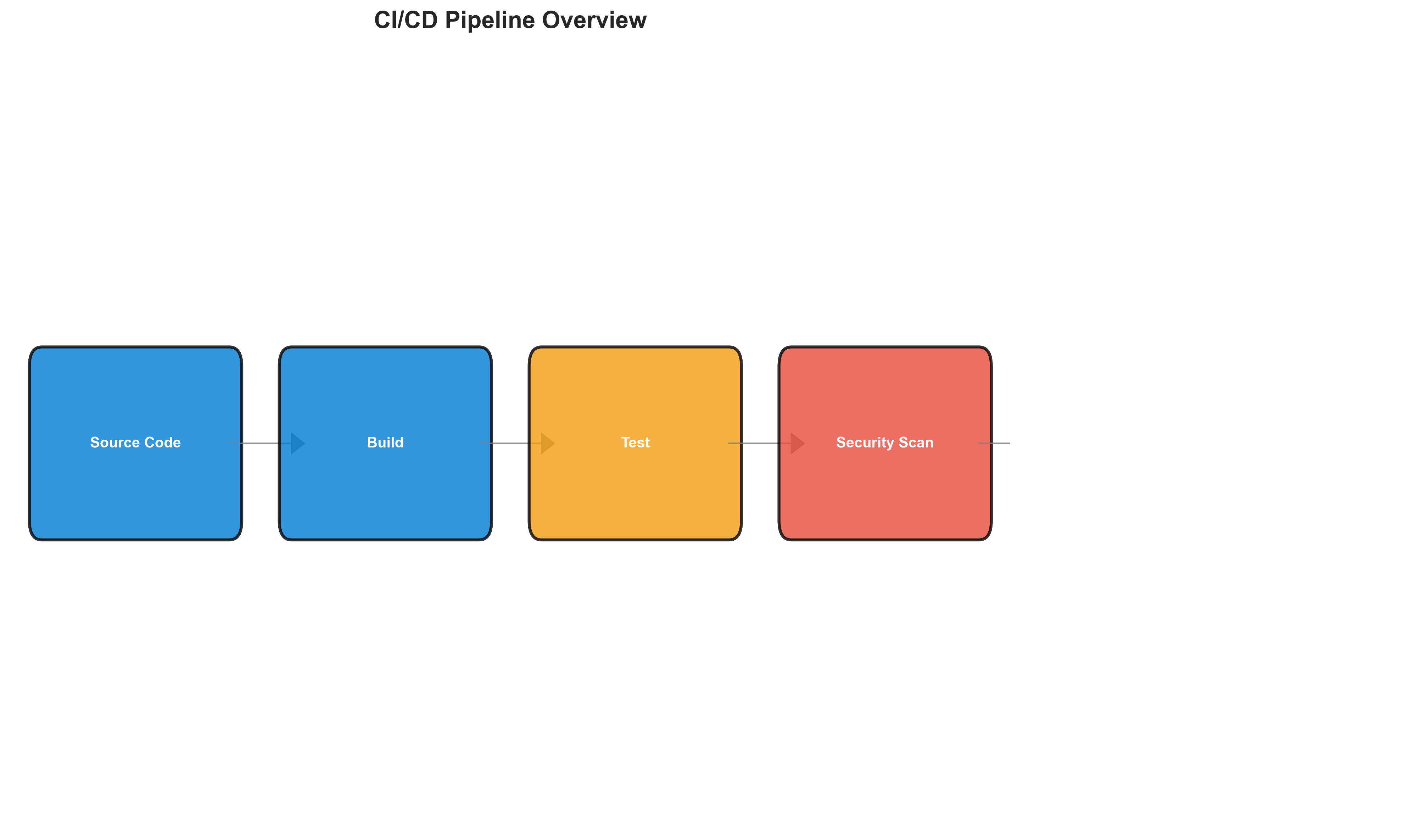
**📊 Business Impact:** Understanding system architecture supports better planning and decision-making.

**📊 Recommendations:** Review analysis results with technical teams for action planning.

**📊 Technical Details:** Diagram type: Risk Assessment Matrix. Based on comprehensive application analysis.

────────────────────────────────────────────────────────────────────────────────

CI/CD Pipeline Overview



**📊 Context:** This Cicd Pipeline shows application architecture insights.

**📊 Key Insights:** Analysis of 3 components reveals system characteristics.

**📊 Business Impact:** Understanding system architecture supports better planning and decision-making.

**📊 Recommendations:** Review analysis results with technical teams for action planning.

**📊 Technical Details:** Diagram type: Cicd Pipeline. Based on comprehensive application analysis.

────────────────────────────────────────────────────────────────────────────────

Component Relationships (Graphviz)



**📊 Context:** Shows status of 3 application components.

**📊 Key Insights:** 3 of 3 components are containerized. Fully ready for cloud deployment.

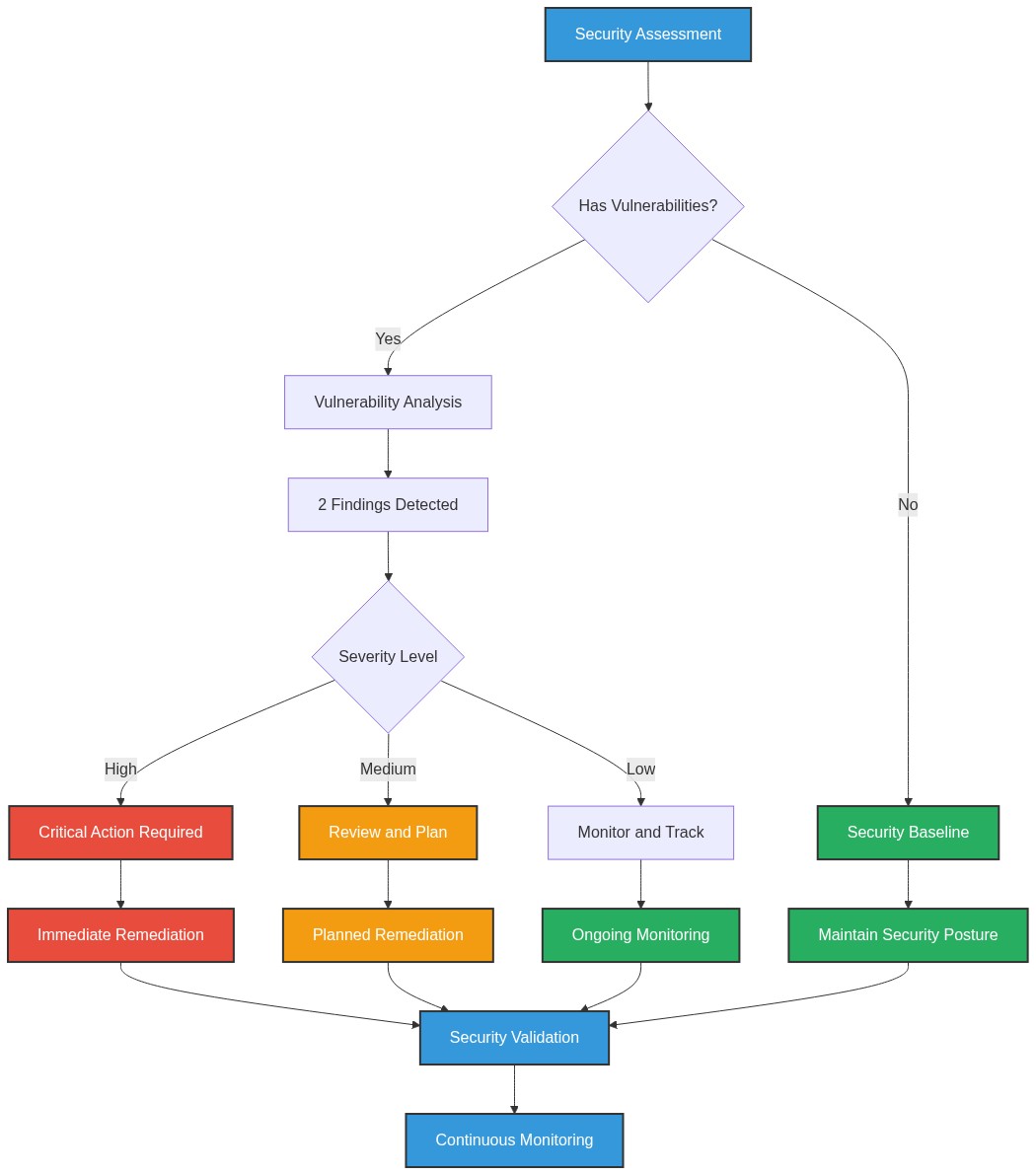
**📊 Business Impact:** Containerization status directly impacts cloud migration readiness and deployment flexibility.

**📊 Recommendations:** All components are containerized. Focus on optimizing container configurations and deployment strategies.

**📊 Technical Details:** Containerization rate: 3/3. Analysis includes Docker and orchestration configurations.

────────────────────────────────────────────────────────────────────────────────

Security Flow Diagram (Mermaid)



**📊 Context:** Security analysis identified 3 potential issues.

**📊 Key Insights:** Security posture assessment shows 3 findings requiring attention.

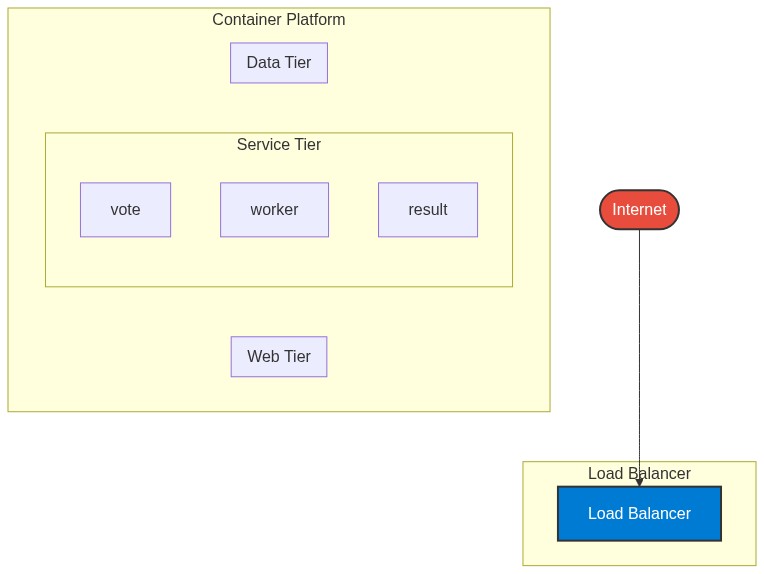
**📊 Business Impact:** Security findings represent potential risks that could impact system reliability and compliance.

**📊 Recommendations:** Address high-severity findings first, then implement security scanning in CI/CD pipeline.

**📊 Technical Details:** Findings: 3. Analysis includes base image vulnerabilities and code patterns.

────────────────────────────────────────────────────────────────────────────────

Deployment Architecture (Mermaid)



**📊 Context:** This Deployment Architecture shows application architecture insights.

**📊 Key Insights:** Analysis of 3 components reveals system characteristics.

**📊 Business Impact:** Understanding system architecture supports better planning and decision-making.

**📊 Recommendations:** Review analysis results with technical teams for action planning.

**📊 Technical Details:** Diagram type: Deployment Architecture. Based on comprehensive application analysis.

────────────────────────────────────────────────────────────────────────────────

Risk Assessment Flow (Mermaid)



**📊 Context:** This Risk Assessment Flow shows application architecture insights.

**📊 Key Insights:** Analysis of 3 components reveals system characteristics.

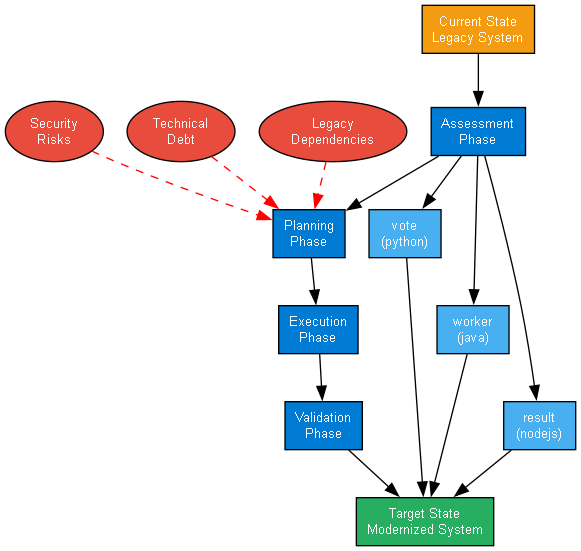
**📊 Business Impact:** Understanding system architecture supports better planning and decision-making.

**📊 Recommendations:** Review analysis results with technical teams for action planning.

**📊 Technical Details:** Diagram type: Risk Assessment Flow. Based on comprehensive application analysis.

────────────────────────────────────────────────────────────────────────────────

Migration Strategy (Graphviz)



**📊 Context:** This Migration Strategy Diagram shows application architecture insights.

**📊 Key Insights:** Analysis of 3 components reveals system characteristics.

**📊 Business Impact:** Understanding system architecture supports better planning and decision-making.

**📊 Recommendations:** Review analysis results with technical teams for action planning.

**📊 Technical Details:** Diagram type: Migration Strategy Diagram. Based on comprehensive application analysis.

────────────────────────────────────────────────────────────────────────────────