[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 | java, nodejs, python |
| 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

• WORD-GEN 3 application components identified

• WORD-GEN 3 programming languages detected: java, nodejs, python

• WORD-GEN 3 components are containerized

• WORD-GEN 0 data sources identified

• WORD-GEN 0 security findings require attention

Detailed Analysis

Component Analysis

The analysis identified 3 components across the application:

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

Component: worker

• Language: java

• Runtime: java

• Build Tool: unknown

• Packaging: docker

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

**Notes:**

• Alternative C# implementation found at 'worker/src/src/Worker/Program.cs' but does not appear to be the primary build target defined in the Dockerfile or OpenShift manifests. Primary implementation is Java.

• 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

Component: vote

• Language: python

• Runtime: python

• Build Tool: unknown

• Packaging: docker

• Exposed Ports: 8080

• Base Images: python:3.9-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 3 findings with 3 base image risks.

**Key Security Findings:**

• Unknown: Maven 3.5 with JDK 8 contains vulnerabilities (Severity: HIGH)

• Unknown: OpenJDK 8 contains known vulnerabilities (Severity: HIGH)

• Unknown: Node.js 10 is EOL and contains numerous vulnerabilities (Severity: HIGH)

Git History Analysis

• Total Commits: 1

• Active Contributors: 0

• Recent Activity: inactive

• Code Stability: high

Recommendations

ERROR High Priority Recommendations

• SECURITY Security: 3 critical/high severity vulnerabilities found. Prioritize security remediation.

• BUSINESS Business Criticality: Assessment is inferred from technical indicators. Validate with business stakeholders.

WORD-GEN Low Priority Recommendations

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

• BASE-IMAGES 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

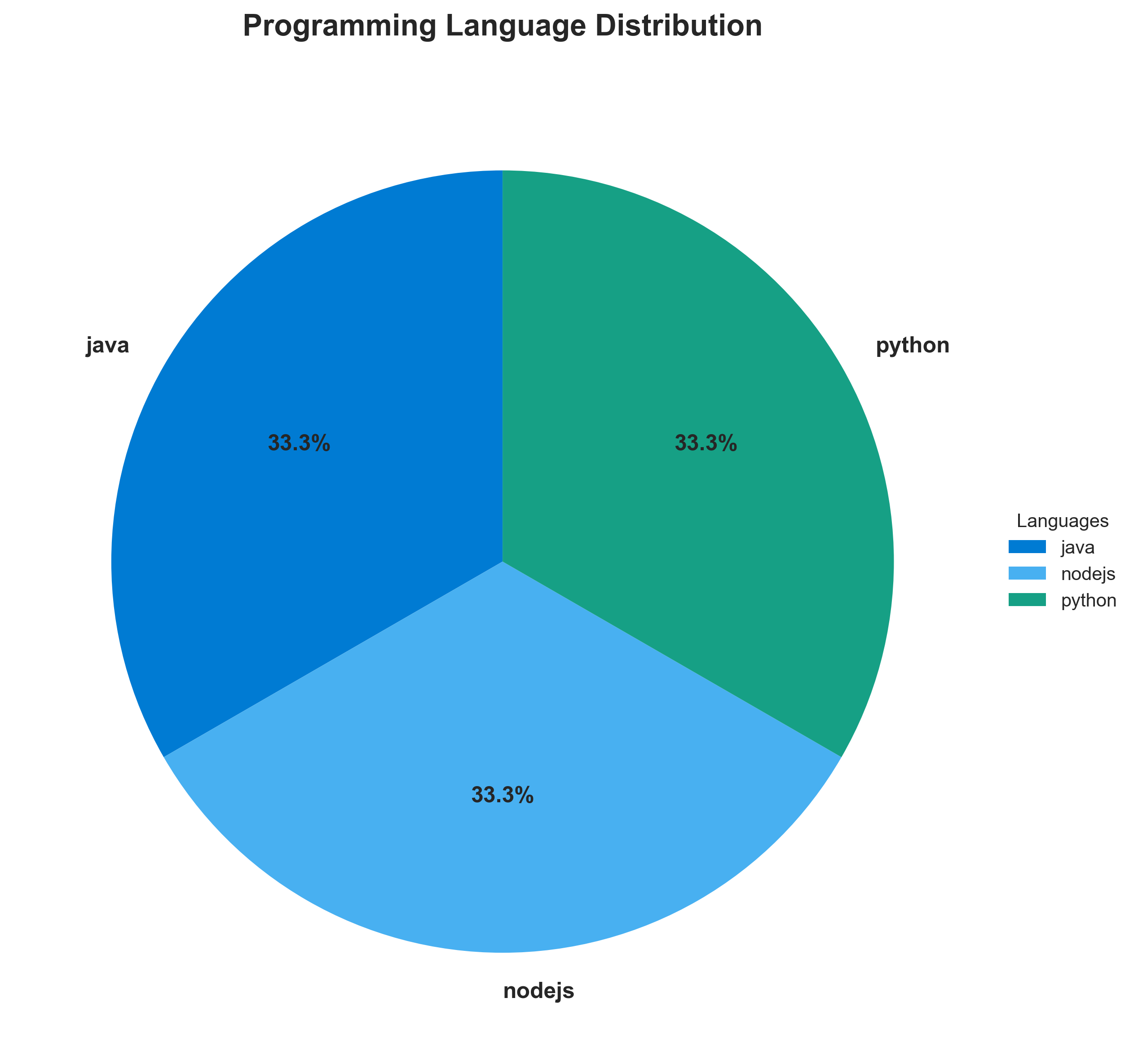
• Architecture Assessment: Evaluation of deployment patterns and component relationships

• Security Analysis: Scanning for common vulnerabilities and configuration issues

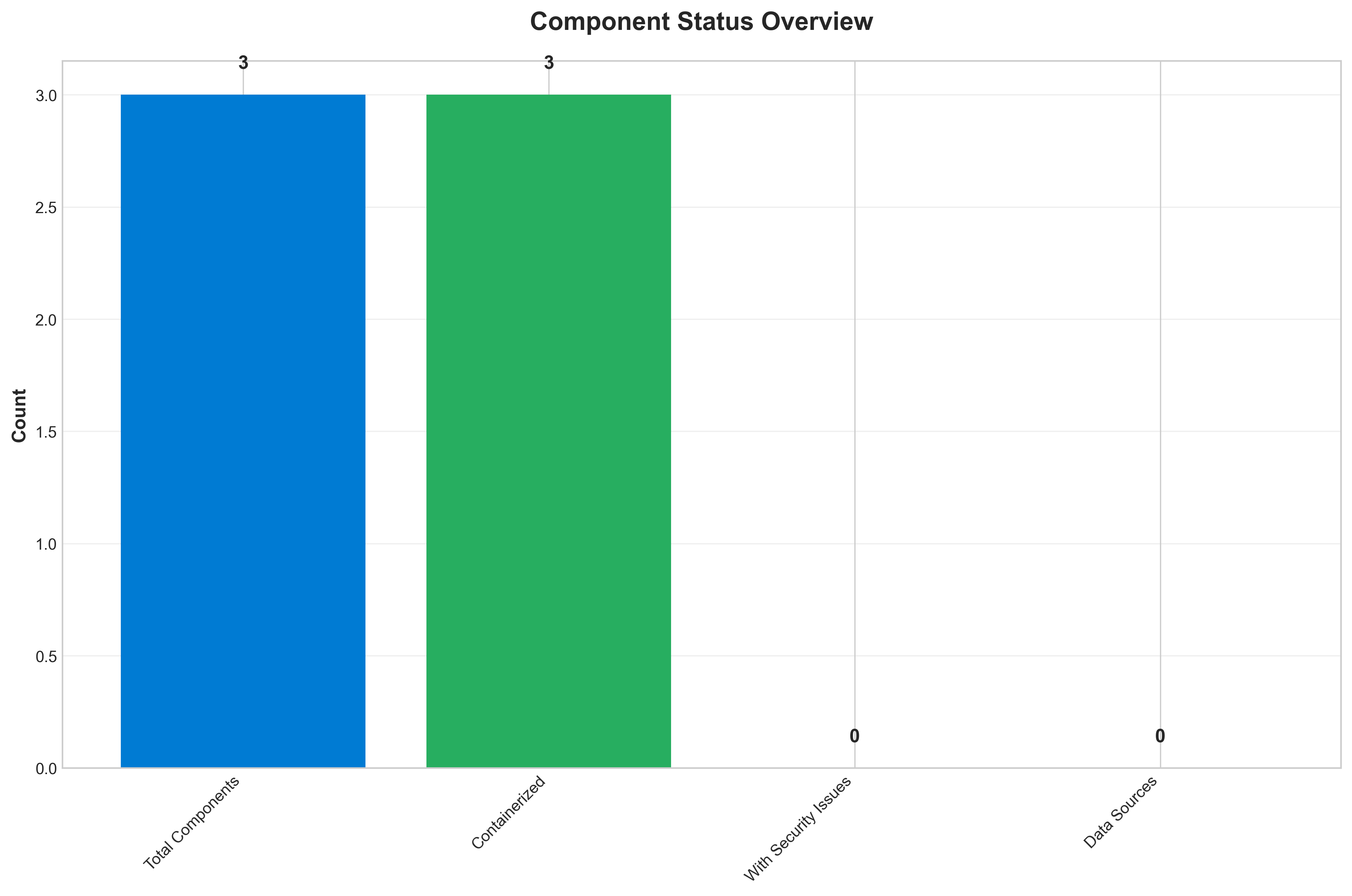
• Git History Analysis: Examination of commit patterns and development activity

Charts and Visualizations

Programming Language Distribution



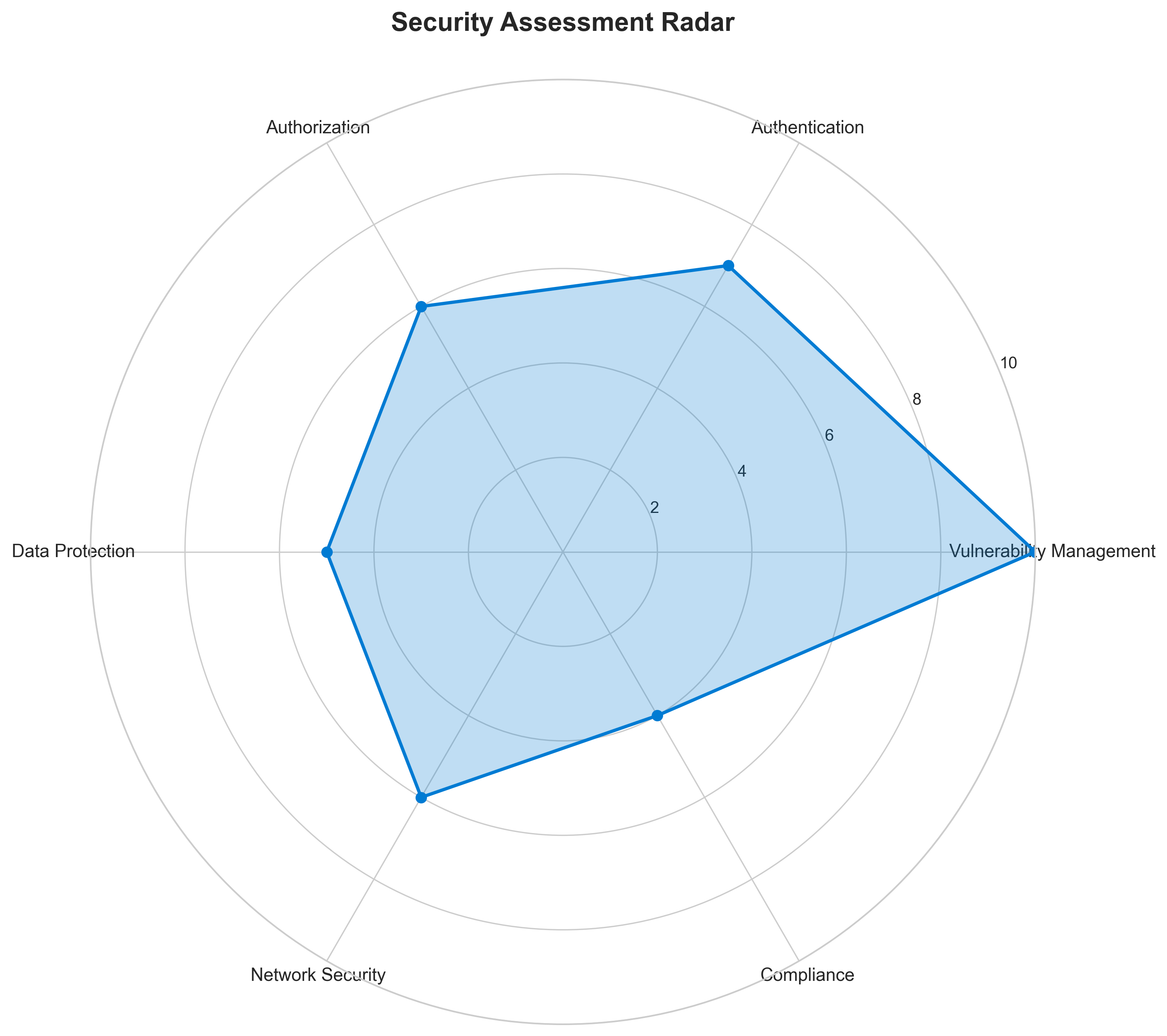
Component Status Overview



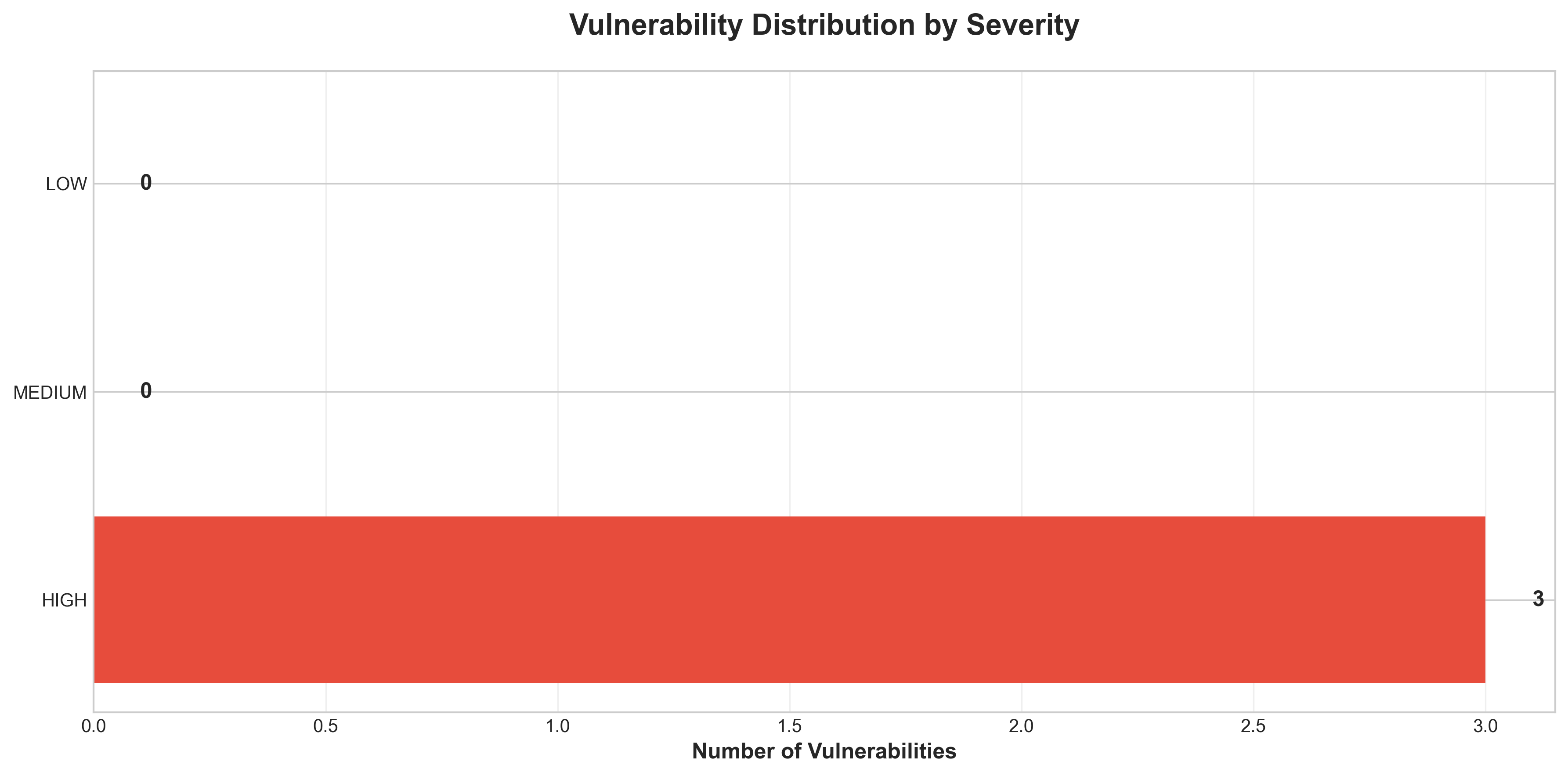
Application Architecture



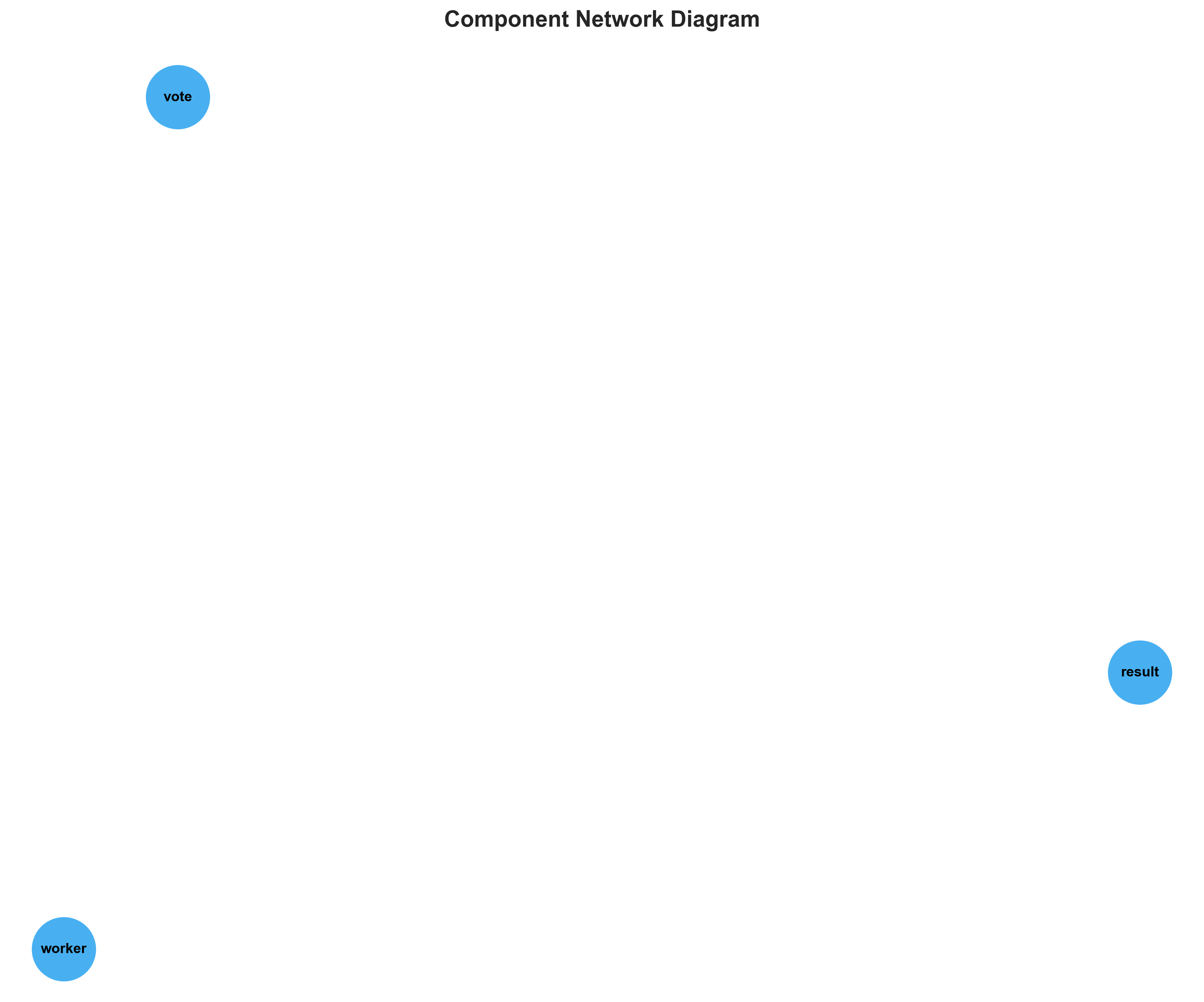
Security Assessment Radar



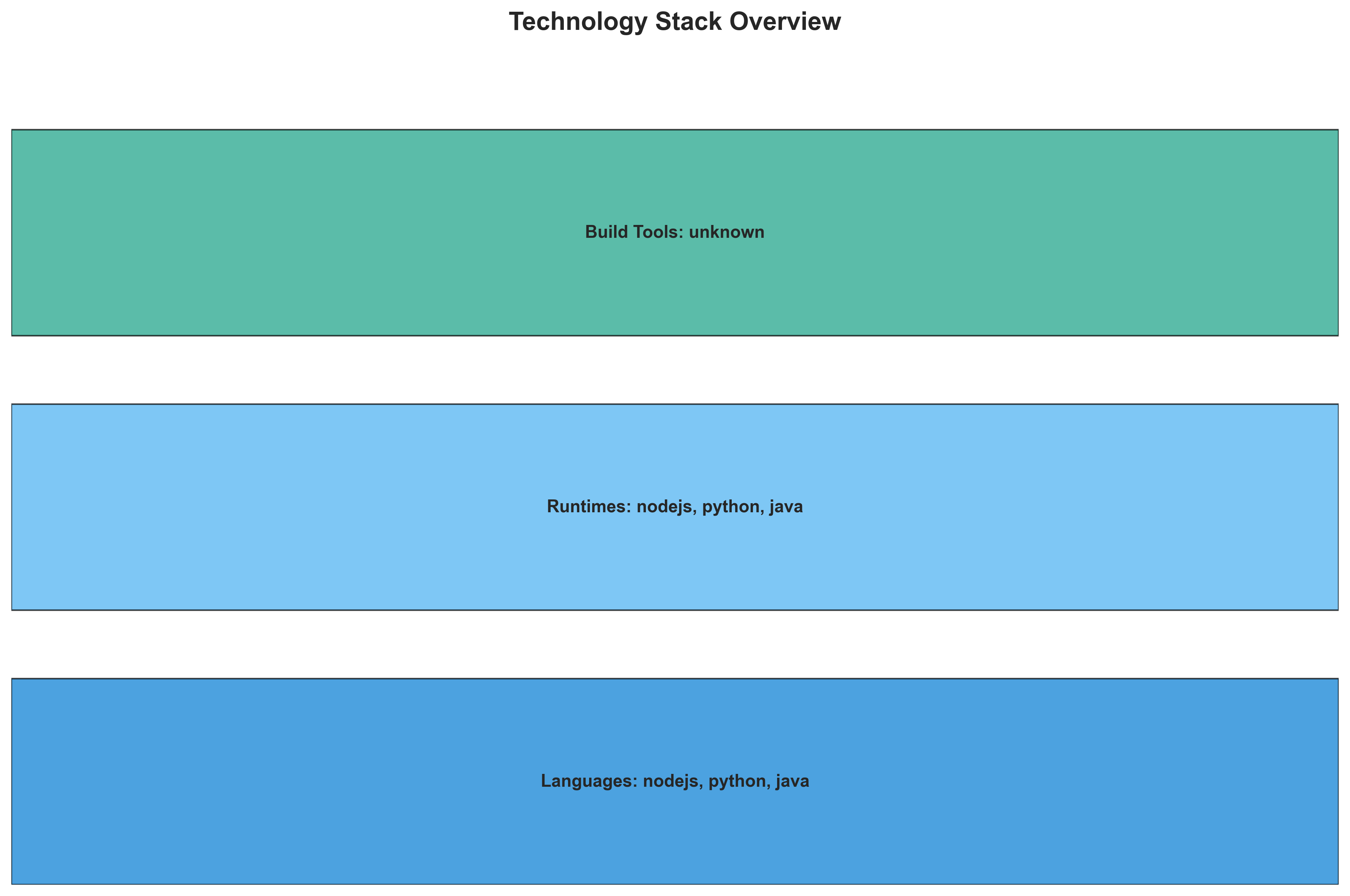
Vulnerability Analysis



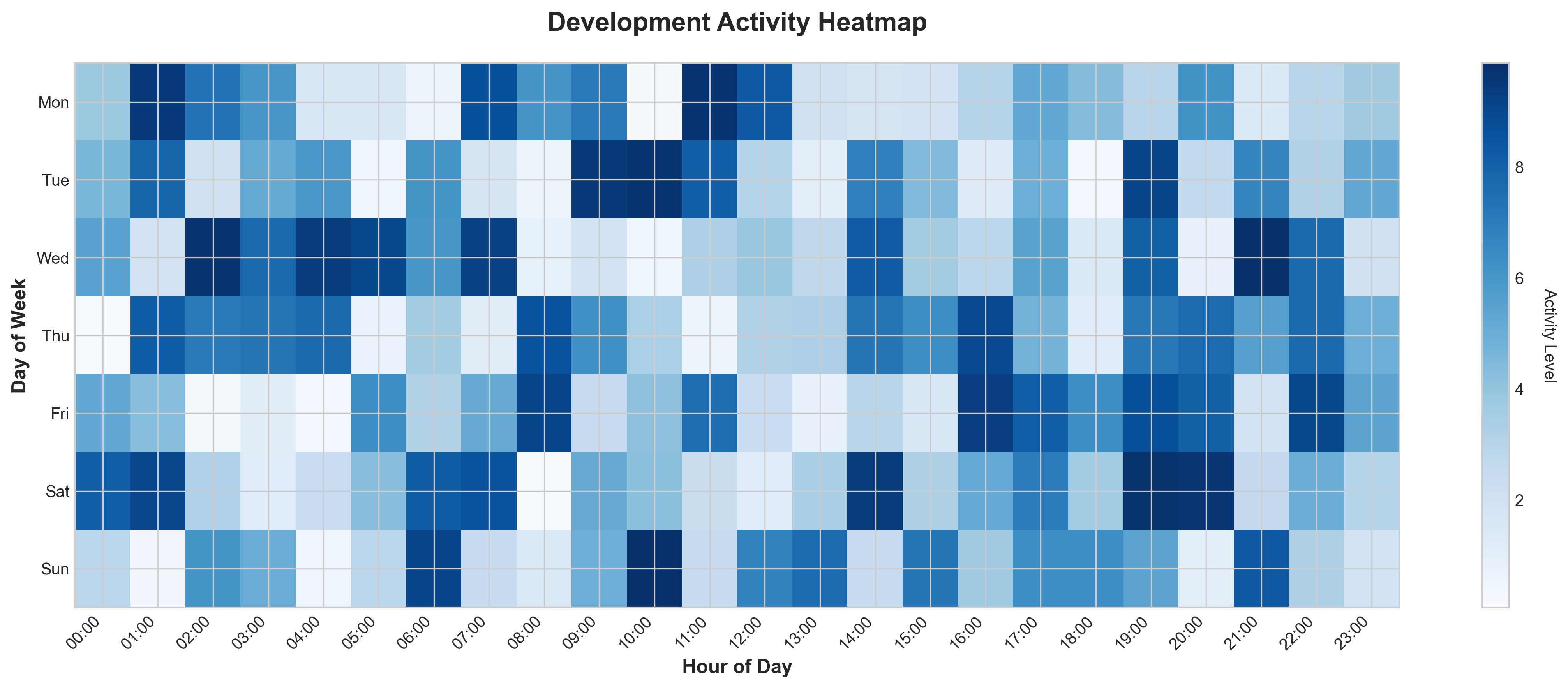
Component Network Topology



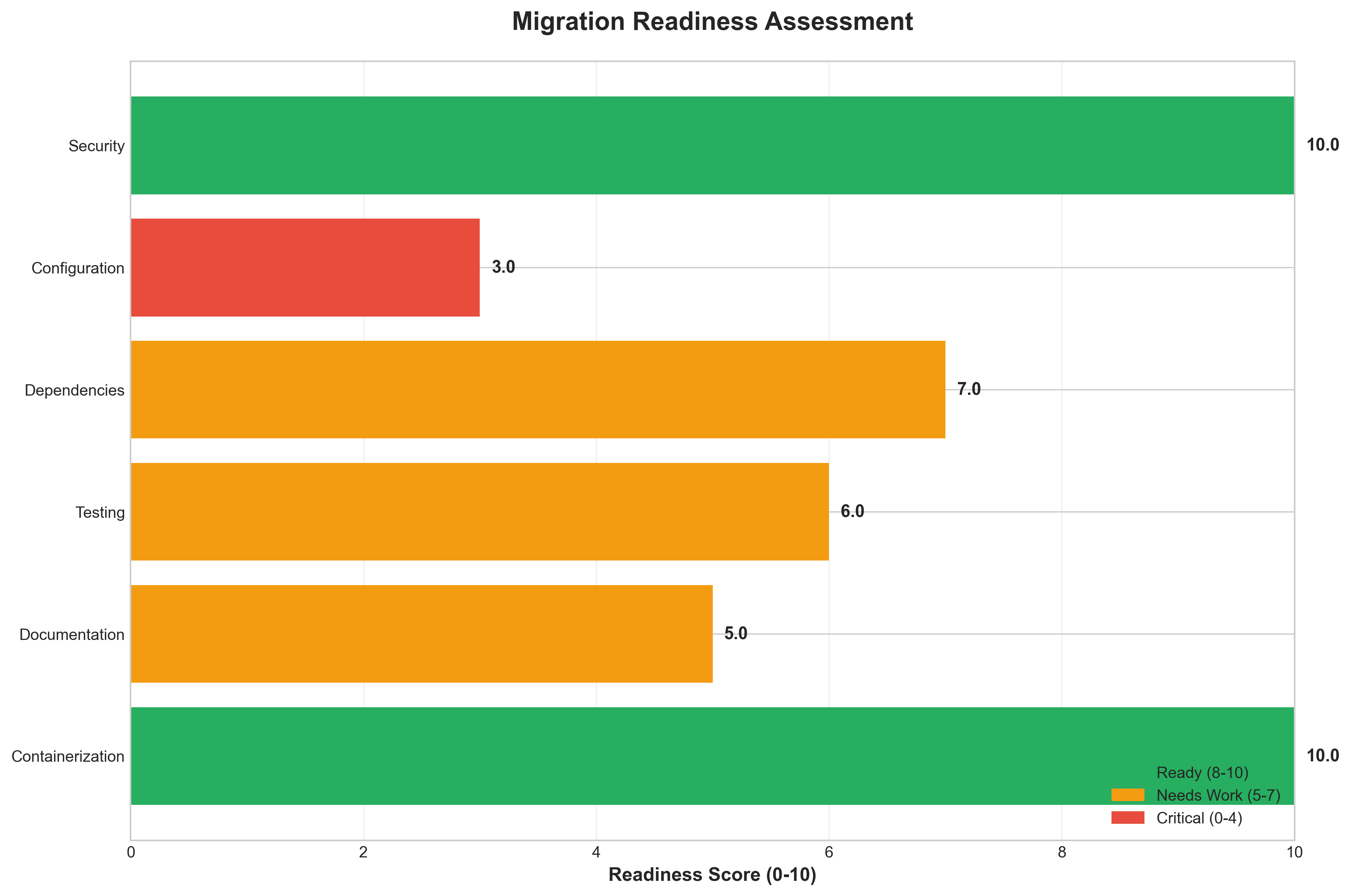
Technology Stack



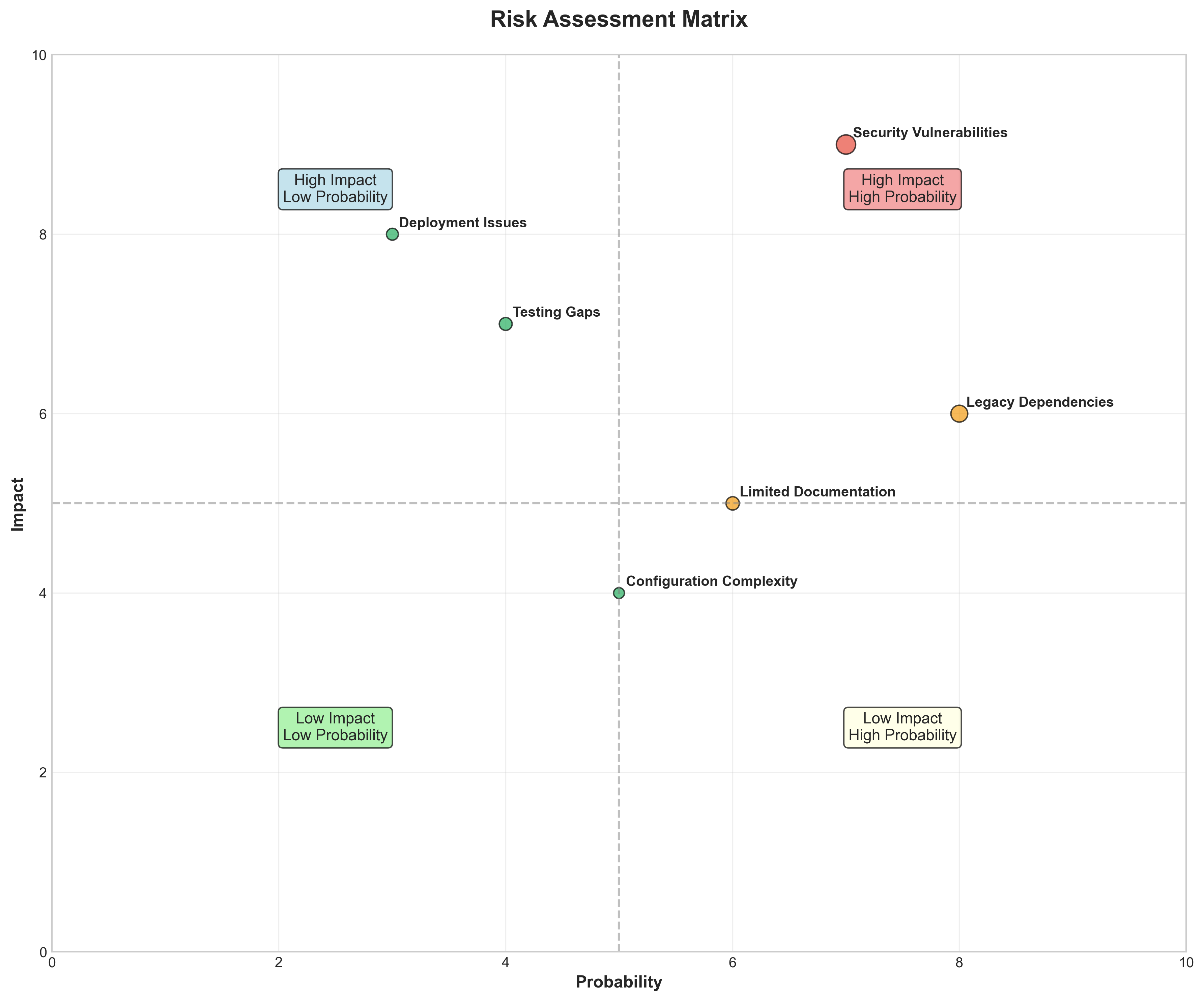
Development Activity Heatmap



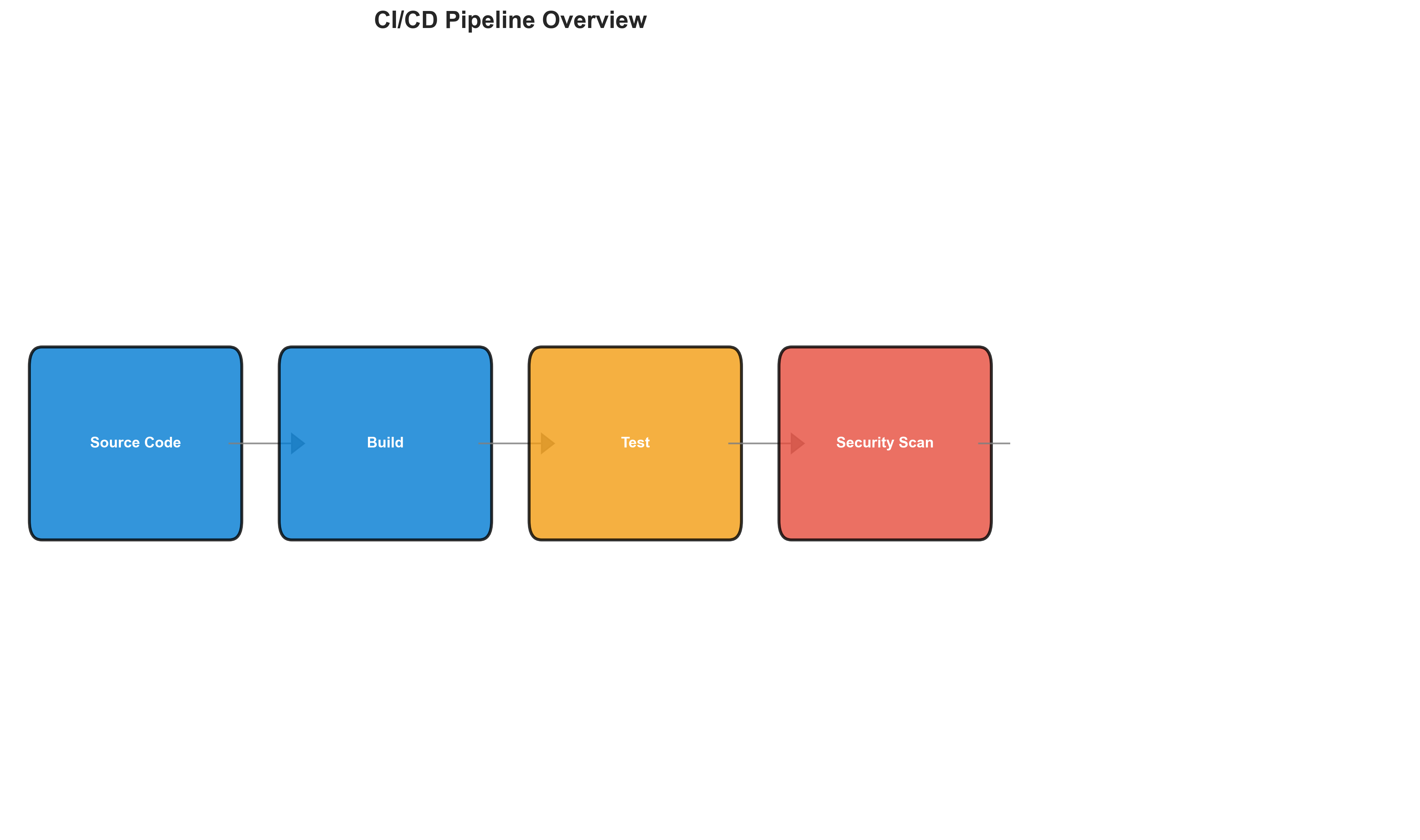
Migration Readiness Assessment



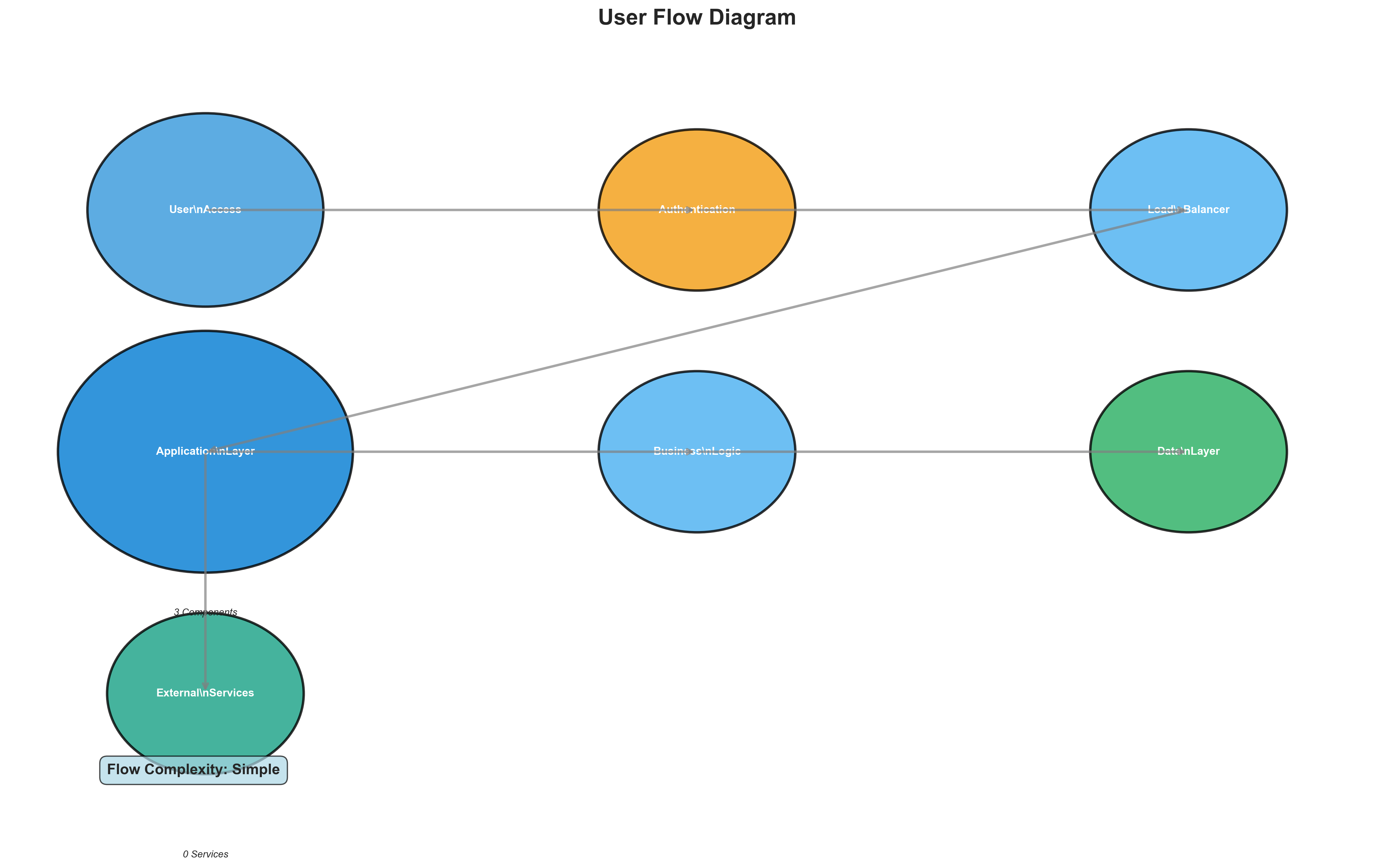
Risk Assessment Matrix



CI/CD Pipeline Overview



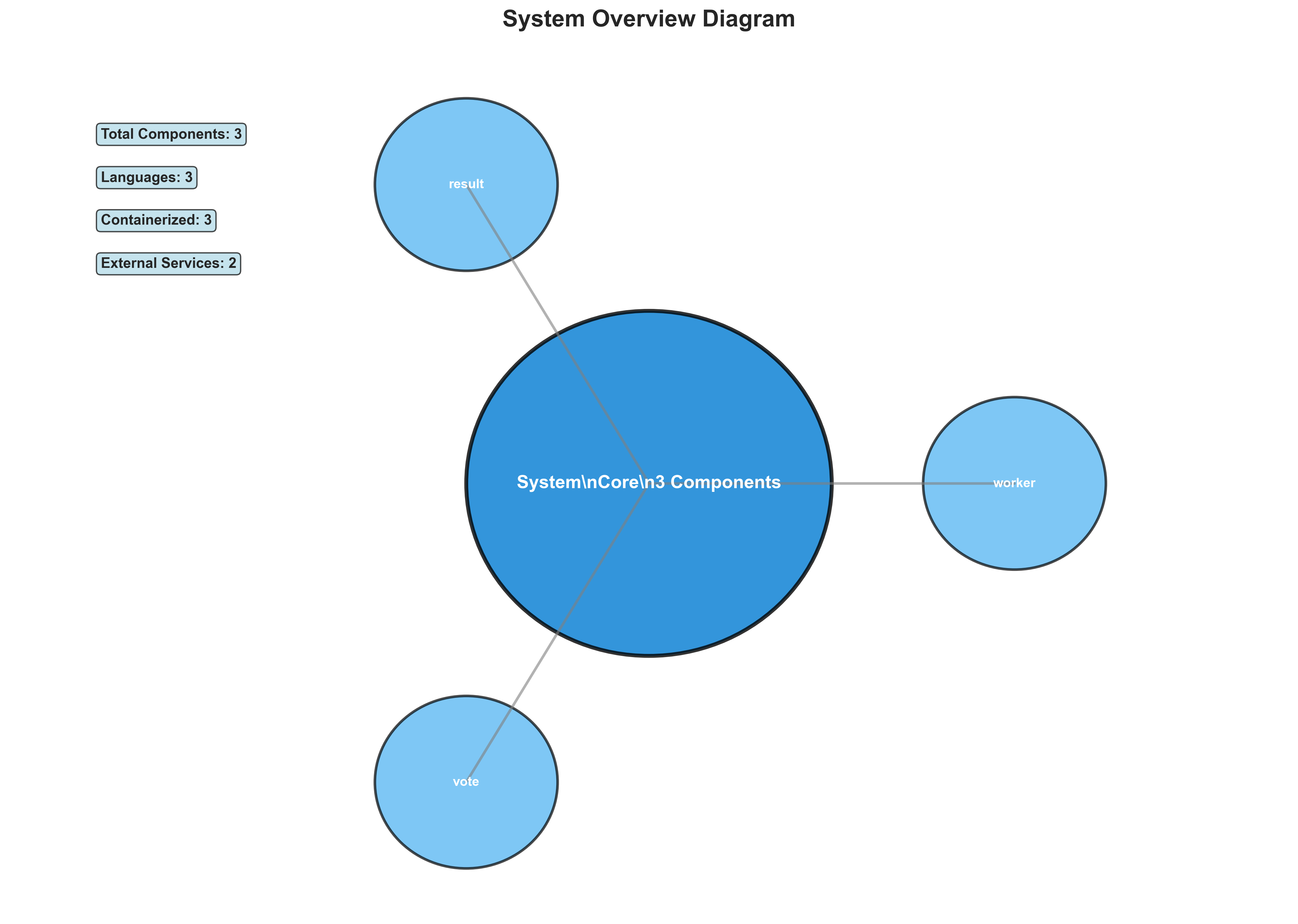
User Flow Diagram



High-Level Architecture Overview



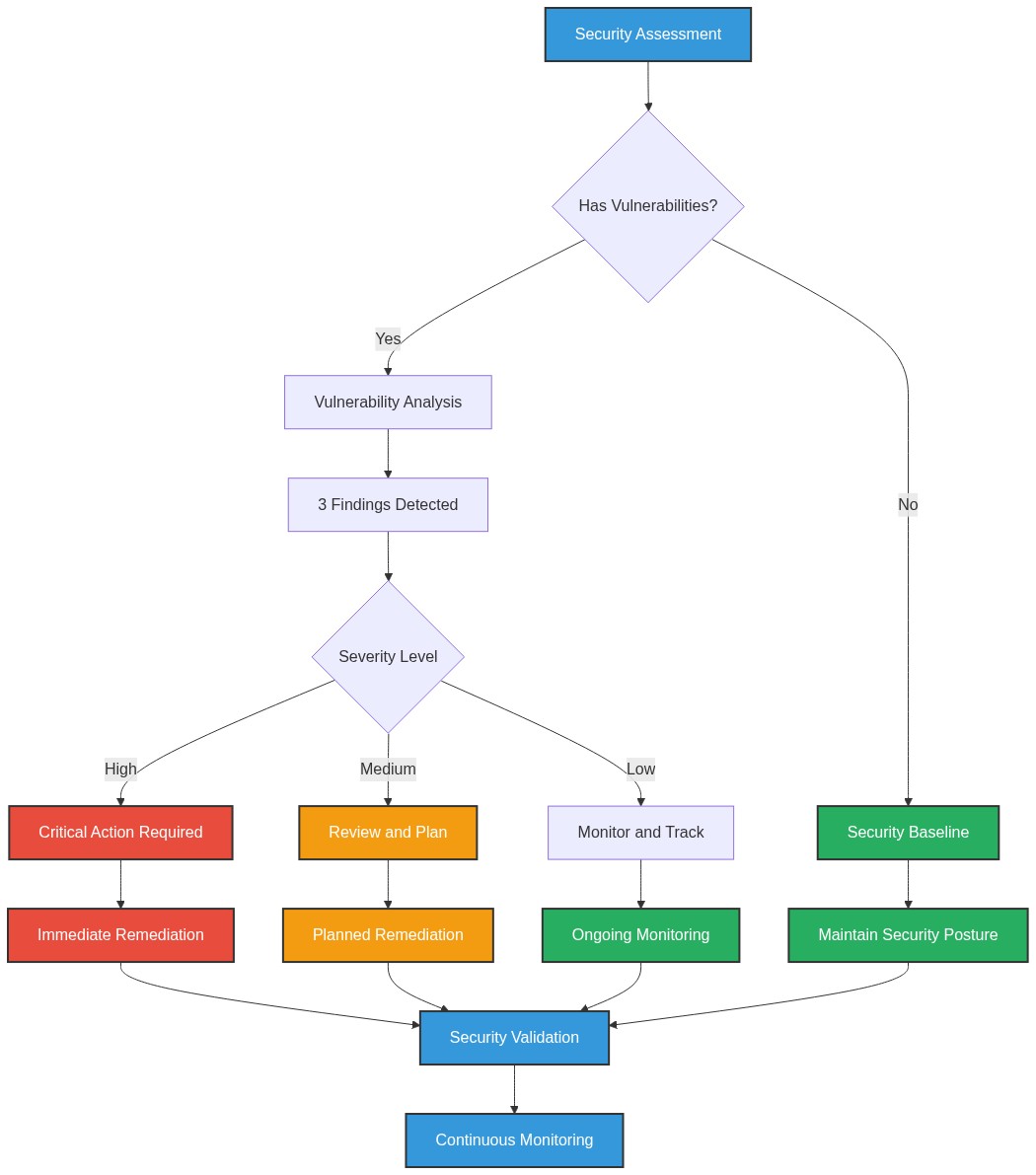
System Overview Diagram



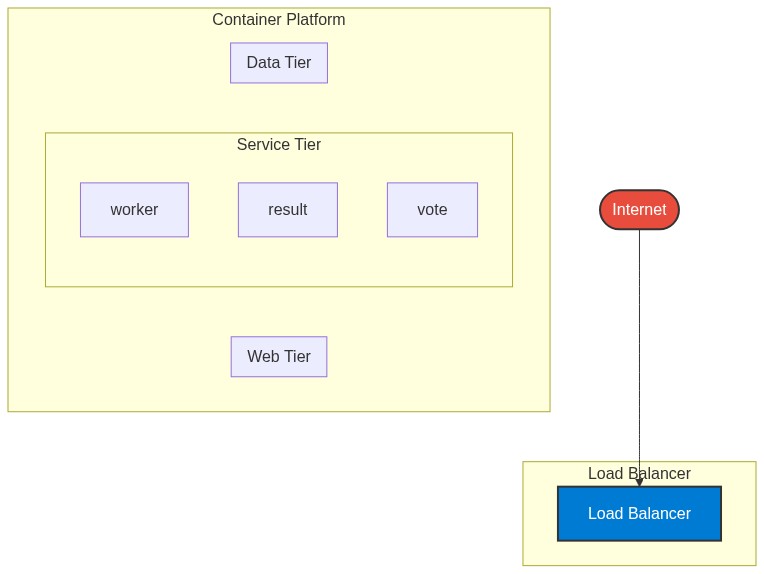
Component Relationships (Graphviz)



Security Flow Diagram (Mermaid)



Deployment Architecture (Mermaid)



Risk Assessment Flow (Mermaid)



Migration Strategy (Graphviz)

