[COMPANY LOGO]

Application Intelligence Report

Comprehensive Analysis and Migration Assessment

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

Analysis Date: July 17, 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 | 8 |
| Programming Languages | java, nodejs, python |
| Containerization Status | 5 containerized |
| Data Sources | 2 |
| Security Findings | 0 |
| Git Commits | 1 |
| Architecture Style | microservices |

Application Overview

This report presents a comprehensive analysis of the application repository. The analysis identified 8 components using 3 different programming languages. The application demonstrates a microservices architecture pattern.

Key Findings

• 📦 8 application components identified

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

• 🐳 5 components are containerized

• 💾 2 data sources identified

• 🔒 0 security findings require attention

Detailed Analysis

Component Analysis

The analysis identified 8 components across the application:

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Language** | **Type** | **Packaging** |
| vote | python | Unknown | docker |
| vote-s2i | python | Unknown | wheel |
| postgresql-ephemeral | unknown | Unknown | docker |
| worker-s2i | java | Unknown | jar |
| redis-ephemeral | unknown | Unknown | docker |
| result-s2i | nodejs | Unknown | npm |
| 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: vote-s2i

• Language: python

• Runtime: python

• Build Tool: unknown

• Packaging: wheel

Component: postgresql-ephemeral

• Language: unknown

• Runtime: unknown

• Build Tool: unknown

• Packaging: docker

Component: worker-s2i

• Language: java

• Runtime: java

• Build Tool: unknown

• Packaging: jar

Component: redis-ephemeral

• Language: unknown

• Runtime: unknown

• Build Tool: unknown

• Packaging: docker

Component: result-s2i

• Language: nodejs

• Runtime: nodejs

• Build Tool: unknown

• Packaging: npm

Component: worker

• Language: java

• Runtime: java

• Build Tool: unknown

• Packaging: docker

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

**Notes:**

• Multiple base images detected: openjdk:8-jre, maven:3.5-jdk-8-alpine. 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 8 components

• Multiple deployable components detected

• 5 containerized components

• Multiple deployment configurations

Security Analysis

Security analysis identified 25 findings with 3 base image risks.

**Key Security Findings:**

• Unknown: Pattern detected: eval\( (Severity: CRITICAL)

• Unknown: Pattern detected: eval\( (Severity: CRITICAL)

• Unknown: Pattern detected: eval\( (Severity: CRITICAL)

• Unknown: Pattern detected: eval\( (Severity: CRITICAL)

• Unknown: Pattern detected: eval\( (Severity: CRITICAL)

Git History Analysis

• Total Commits: 1

• Active Contributors: 0

• Recent Activity: inactive

• Code Stability: high

Recommendations

🔴 High Priority Recommendations

• 🔒 Security: 25 critical/high severity vulnerabilities found. Prioritize security remediation.

🟢 Low Priority Recommendations

• 🔍 Component Analysis: 2 components have unknown languages. Review build configurations and source code structure.

• 📊 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

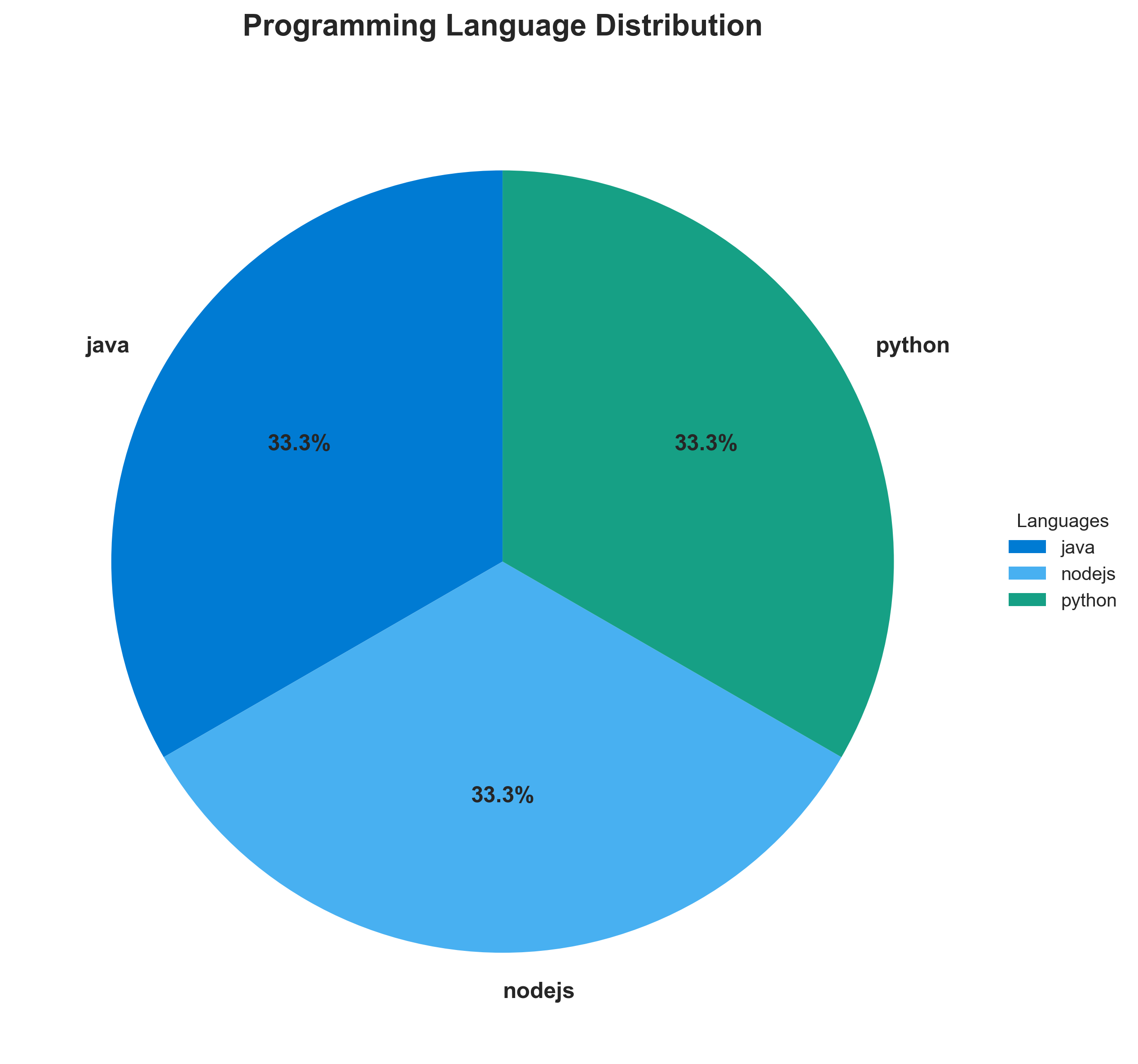
• Architecture Assessment: Evaluation of deployment patterns and component relationships

• 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:** This pie chart displays the distribution of programming languages across 8 application components. The analysis identified 3 different programming languages in use.

**📊 Key Insights:** The application uses 3 programming languages (java, nodejs, python), representing a balanced technology stack with java as the primary language.

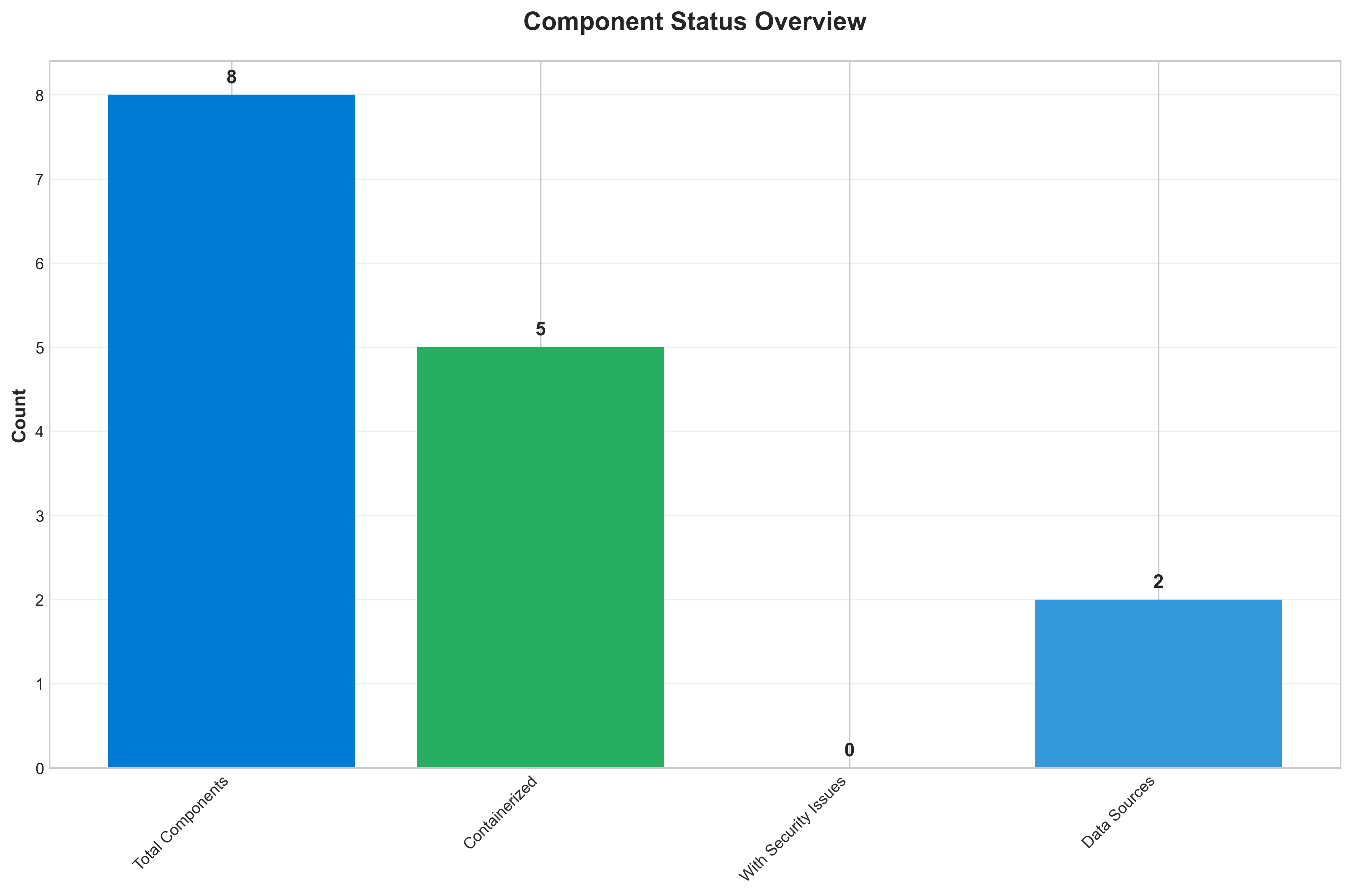
**📊 Business Impact:** Moderate language diversity provides good balance between flexibility and maintainability. The current mix allows for technology optimization while keeping complexity manageable.

**📊 Recommendations:** Monitor language usage patterns to prevent unnecessary sprawl. Document architectural decisions for language choices. Ensure cross-training opportunities for development teams.

**📊 Technical Details:** Languages detected: java, nodejs, python. Component distribution varies by language, with java representing the largest portion of the codebase.

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

Component Status Overview



**📊 Context:** This bar chart provides an overview of component status across 8 application components, showing containerization progress, security posture, and data source connections.

**📊 Key Insights:** Containerization status: good. Moderate containerization rate (5/8 components) shows ongoing modernization efforts. Security status: excellent. No security vulnerabilities detected, indicating good security posture. Data integration: 2 external data sources detected.

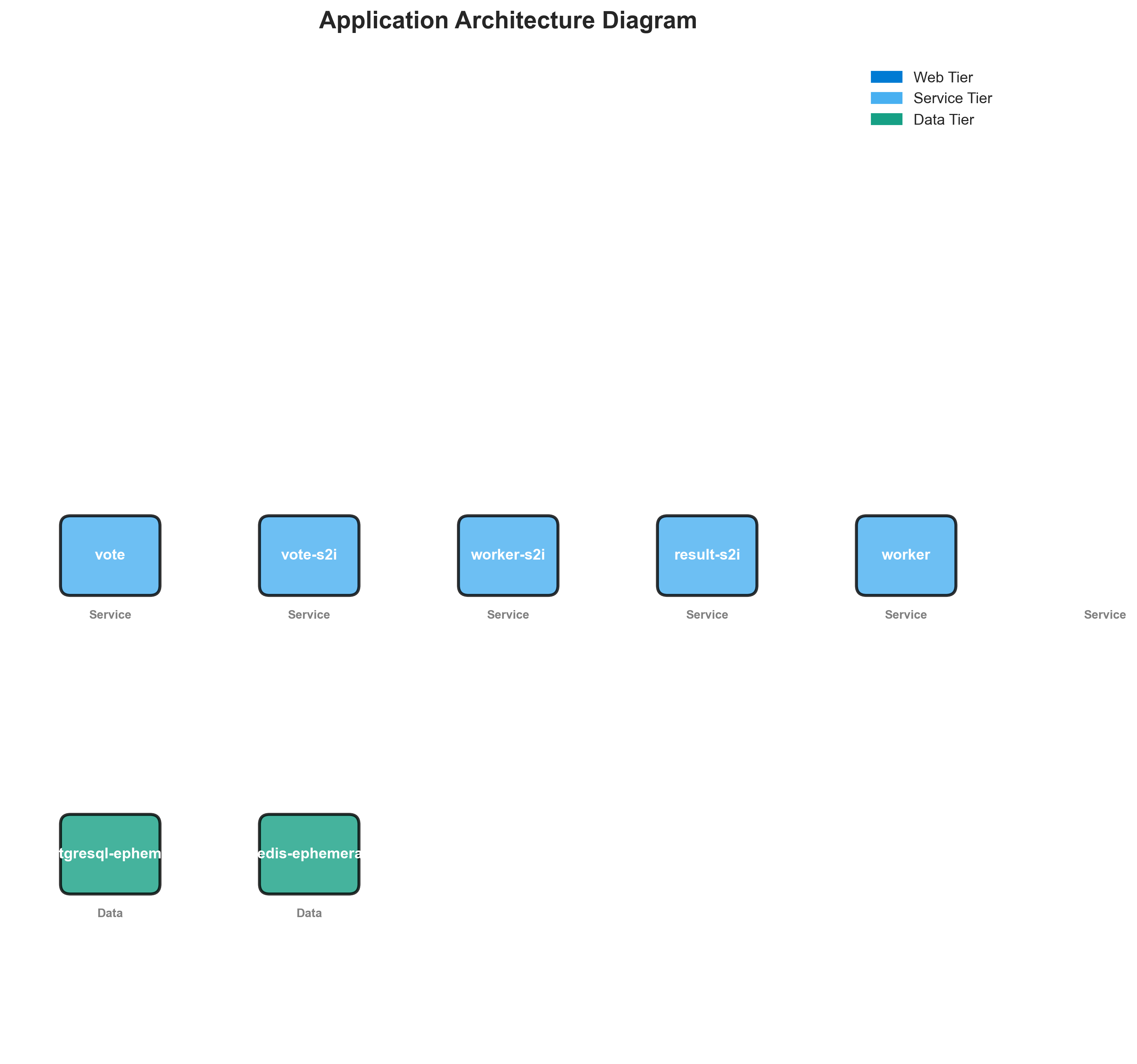
**📊 Business Impact:** Mixed technical status requires balanced approach to address containerization and security improvements.

**📊 Recommendations:** Increase containerization from 5 to 8 components; Review 2 data source connections for optimization

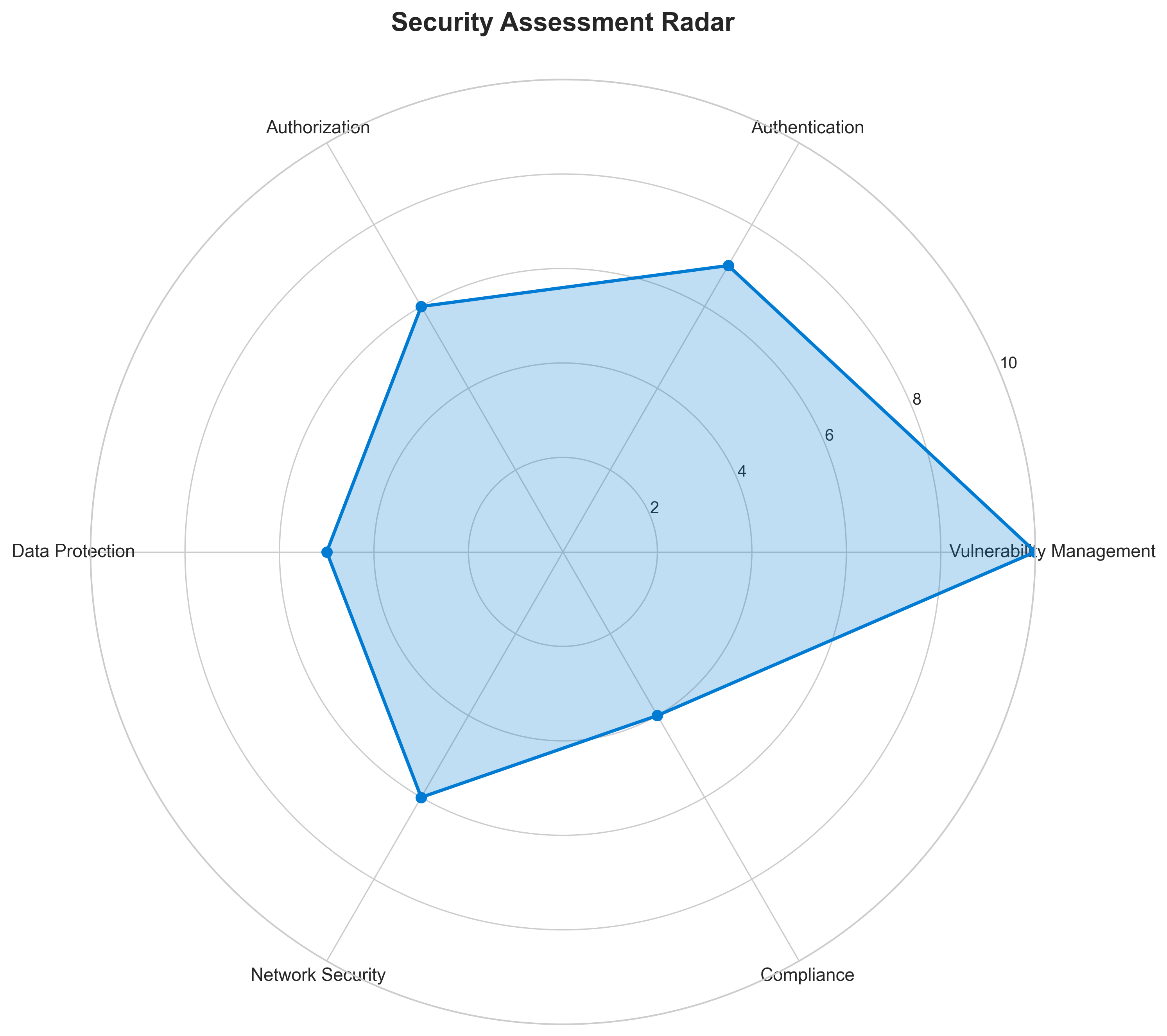
**📊 Technical Details:** Metrics: 5 containerized components, 0 security findings, 2 data sources. Containerization rate: 62.5%.

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

Application Architecture



Security Assessment Radar



**📊 Context:** This radar chart visualizes the application's security posture across six key dimensions: Vulnerability Management, Authentication, Authorization, Data Protection, Network Security, and Compliance.

**📊 Key Insights:** Weak security posture detected. Weak security posture with significant vulnerabilities (25 findings). Immediate security improvements required. Severity distribution: 1 High, 0 Medium, 0 Low severity issues.

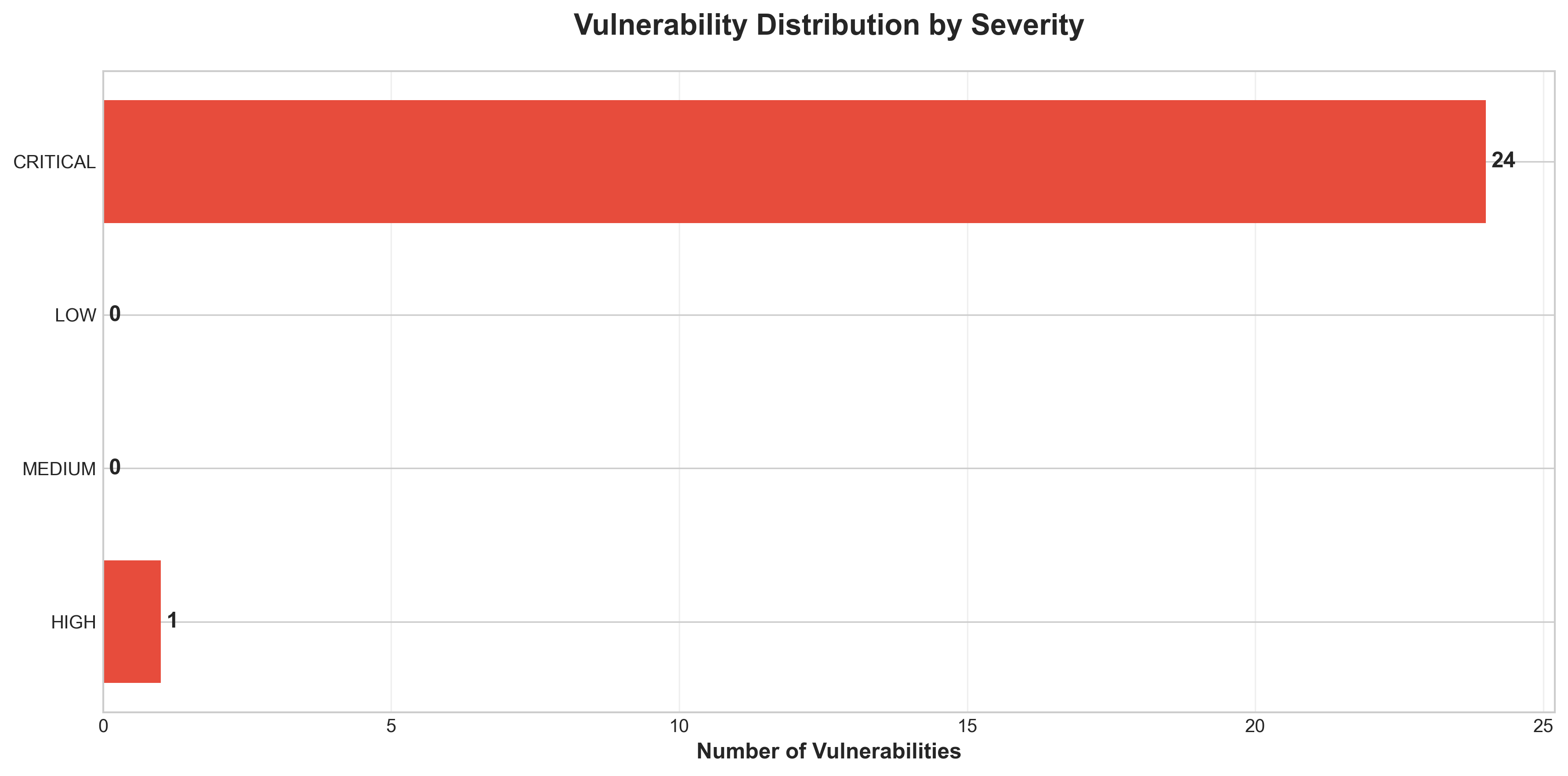
**📊 Business Impact:** Critical security risks detected (1 high severity issues). This poses immediate risk to data integrity and system availability. Potential for security breaches and compliance violations.

**📊 Recommendations:** URGENT: Address 1 high-severity vulnerabilities immediately. Implement emergency patches and security updates. Conduct security review of affected components.

**📊 Technical Details:** Security assessment: 25 total findings across all components. Risk distribution: 1 High, 0 Medium, 0 Low. Overall security score: 1/10.

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

Vulnerability Analysis



**📊 Context:** This timeline chart shows the distribution of 25 security vulnerabilities across severity levels, providing a prioritized view of security remediation needs.

**📊 Key Insights:** CRITICAL priority security work identified. Critical security situation with 1 high-severity vulnerabilities requiring immediate attention. Vulnerability breakdown: 1 critical issues need immediate action, 0 medium issues for planned remediation, 0 low issues for routine maintenance.

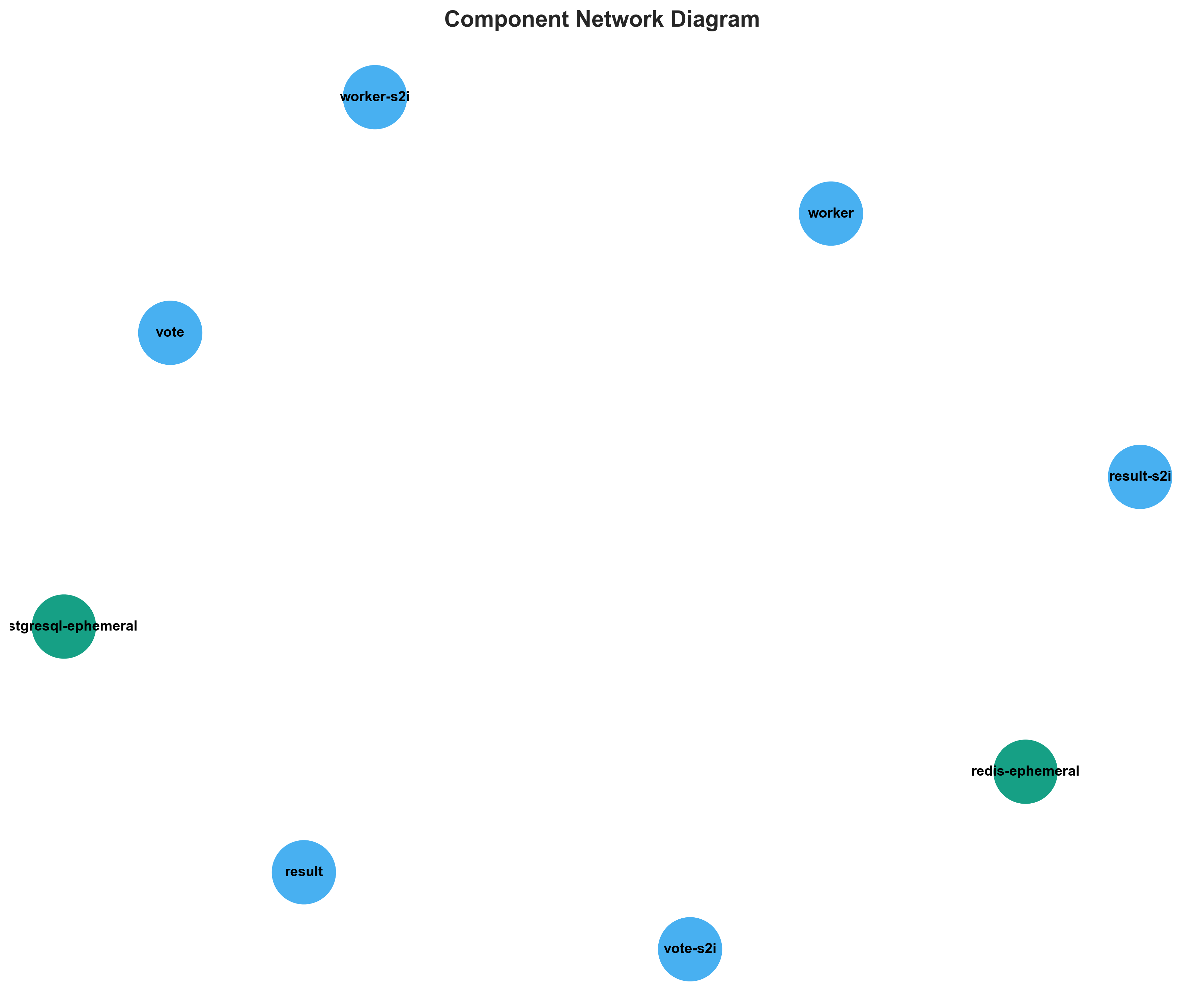
**📊 Business Impact:** HIGH BUSINESS RISK: 1 critical vulnerabilities could lead to security breaches, data loss, or system compromise. Immediate action required to prevent potential incidents.

**📊 Recommendations:** IMMEDIATE (24-48 hours): Patch 1 high-severity vulnerabilities; Implement automated vulnerability scanning in CI/CD pipeline

**📊 Technical Details:** Vulnerability timeline: 1 High-severity (immediate), 0 Medium-severity (planned), 0 Low-severity (routine). Total security debt: 25 findings.

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

Component Network Topology



**📊 Context:** This Component Network provides visual insights into the application's architecture and operational characteristics.

**📊 Key Insights:** The diagram illustrates key aspects of the application that support technical decision-making and operational planning.

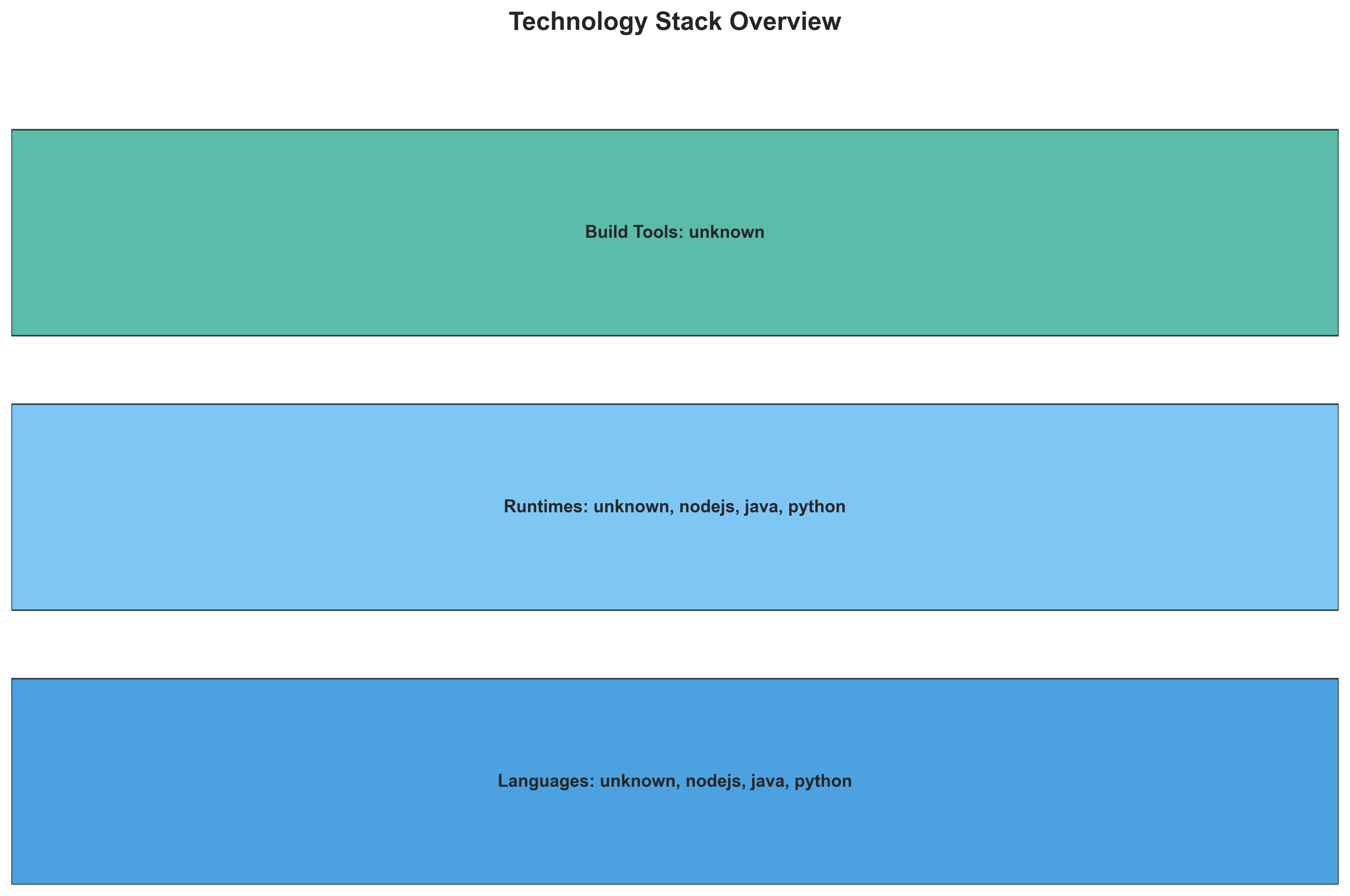
**📊 Business Impact:** Visual representation supports better understanding of system characteristics and aids in strategic planning.

**📊 Recommendations:** Review diagram insights with technical teams to identify optimization opportunities and action items.

**📊 Technical Details:** Diagram type: Component Network. Generated from comprehensive application intelligence analysis.

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

Technology Stack



**📊 Context:** This Technology Stack provides visual insights into the application's architecture and operational characteristics.

**📊 Key Insights:** The diagram illustrates key aspects of the application that support technical decision-making and operational planning.

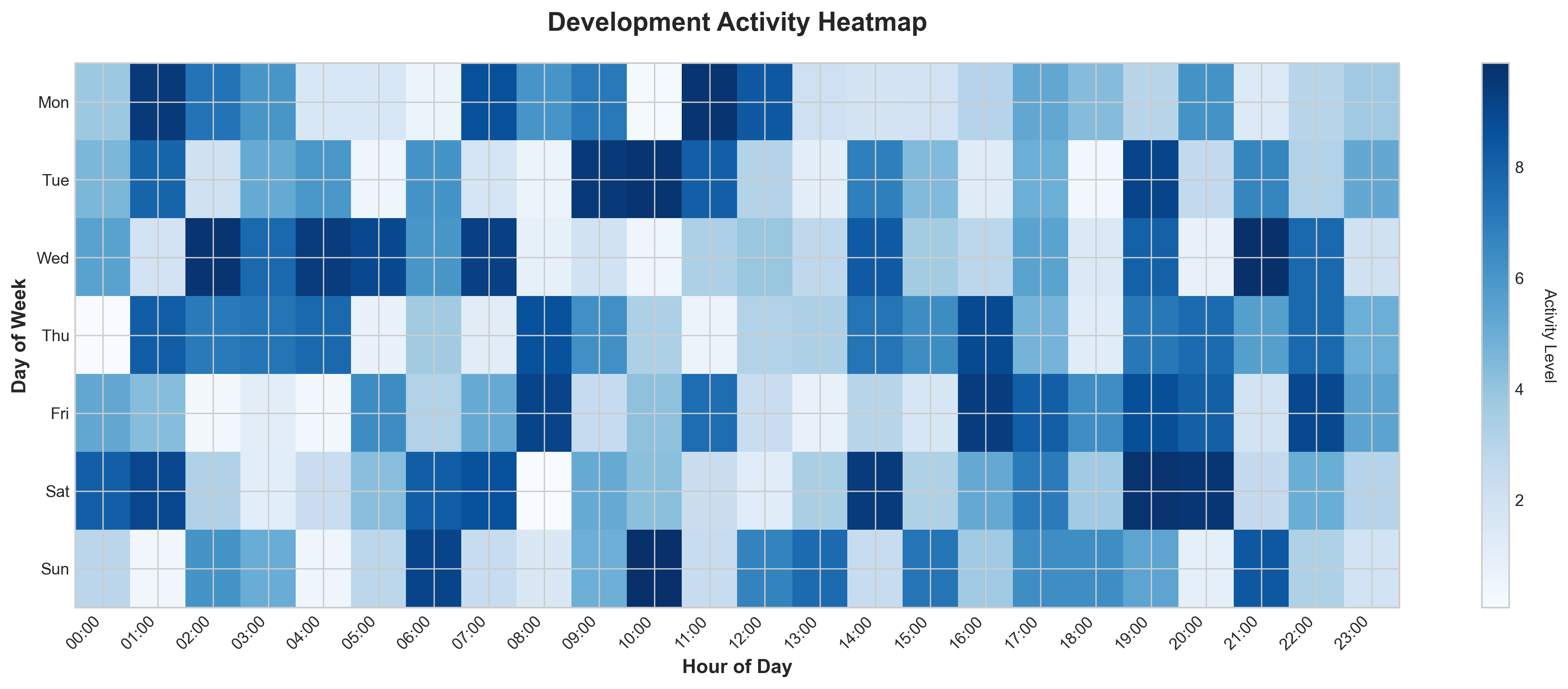
**📊 Business Impact:** Visual representation supports better understanding of system characteristics and aids in strategic planning.

**📊 Recommendations:** Review diagram insights with technical teams to identify optimization opportunities and action items.

**📊 Technical Details:** Diagram type: Technology Stack. Generated from comprehensive application intelligence analysis.

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

Development Activity Heatmap



**📊 Context:** This Development Activity provides visual insights into the application's architecture and operational characteristics.

**📊 Key Insights:** The diagram illustrates key aspects of the application that support technical decision-making and operational planning.

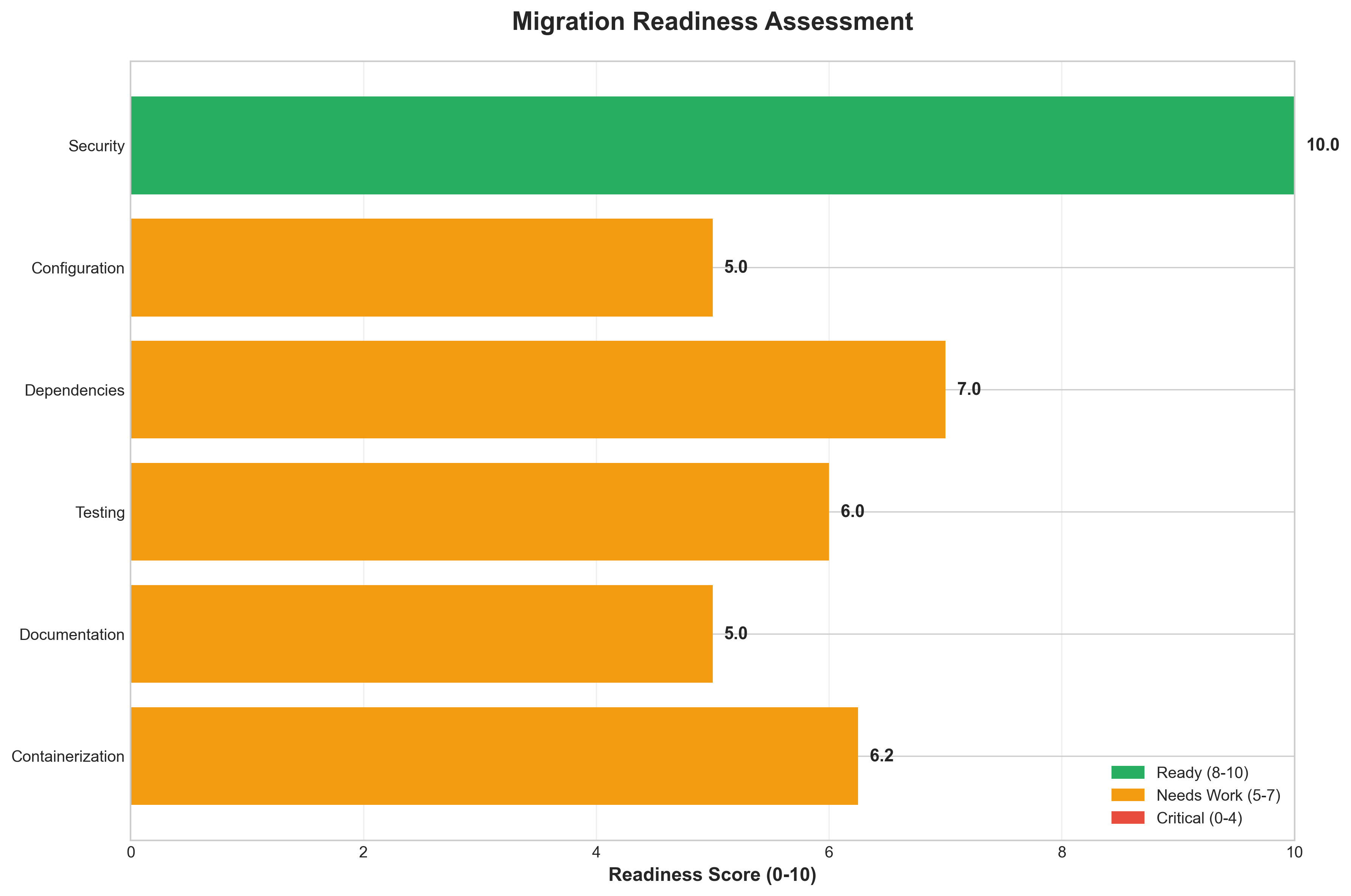
**📊 Business Impact:** Visual representation supports better understanding of system characteristics and aids in strategic planning.

**📊 Recommendations:** Review diagram insights with technical teams to identify optimization opportunities and action items.

**📊 Technical Details:** Diagram type: Development Activity. Generated from comprehensive application intelligence analysis.

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

Migration Readiness Assessment



**📊 Context:** This migration readiness assessment evaluates the application's preparedness for cloud migration across six key factors: Containerization, Documentation, Testing, Dependencies, Configuration, and Security.

**📊 Key Insights:** HIGH migration readiness. High migration readiness (score: 8.1/10). The application is well-prepared for cloud migration with minimal blockers. Factor breakdown: Containerization: Needs work; Security: Ready

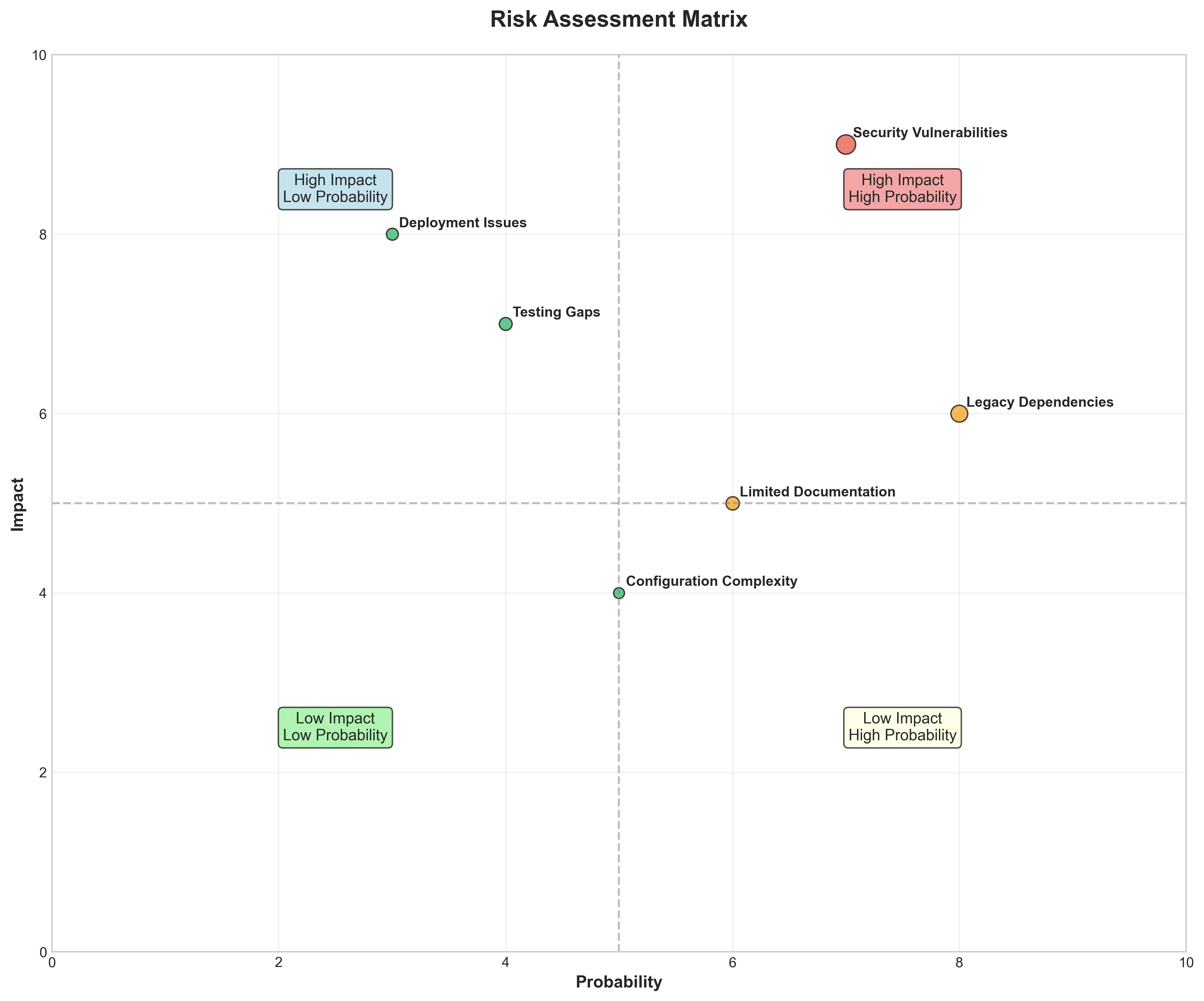
**📊 Business Impact:** POSITIVE: Application is ready for migration with minimal risk. Migration can proceed with confidence, enabling faster time-to-market and reduced infrastructure costs.

**📊 Recommendations:** Increase containerization from 5 to 8 components; Conduct migration pilot with least complex component; Establish rollback procedures and monitoring

**📊 Technical Details:** Migration readiness scores: Overall 8.1/10, Containerization 6.2/10, Security 10.0/10. 5/8 components containerized, 0 security issues.

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

Risk Assessment Matrix



**📊 Context:** This Risk Assessment Matrix provides visual insights into the application's architecture and operational characteristics.

**📊 Key Insights:** The diagram illustrates key aspects of the application that support technical decision-making and operational planning.

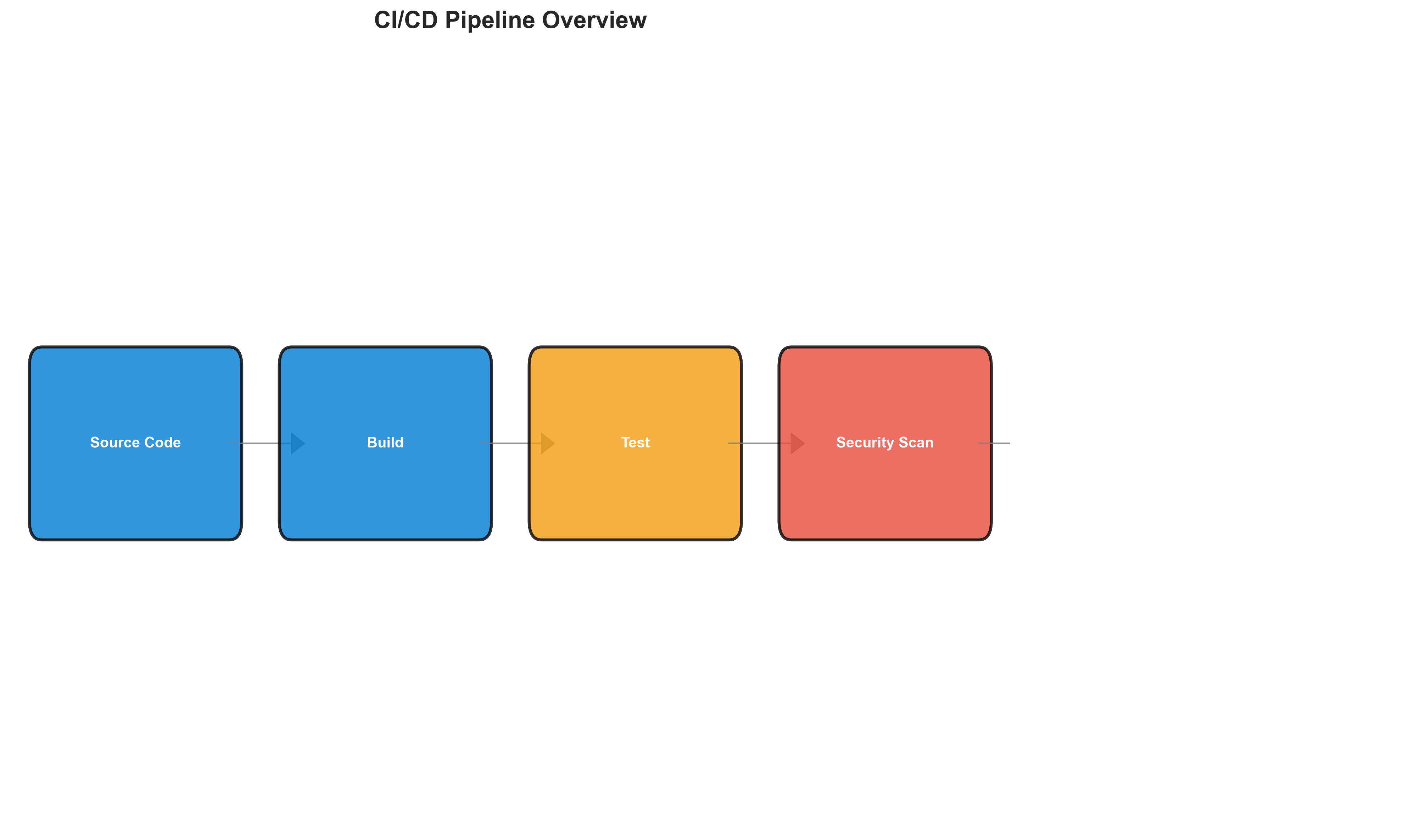
**📊 Business Impact:** Visual representation supports better understanding of system characteristics and aids in strategic planning.

**📊 Recommendations:** Review diagram insights with technical teams to identify optimization opportunities and action items.

**📊 Technical Details:** Diagram type: Risk Assessment Matrix. Generated from comprehensive application intelligence analysis.

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

CI/CD Pipeline Overview



**📊 Context:** This Cicd Pipeline provides visual insights into the application's architecture and operational characteristics.

**📊 Key Insights:** The diagram illustrates key aspects of the application that support technical decision-making and operational planning.

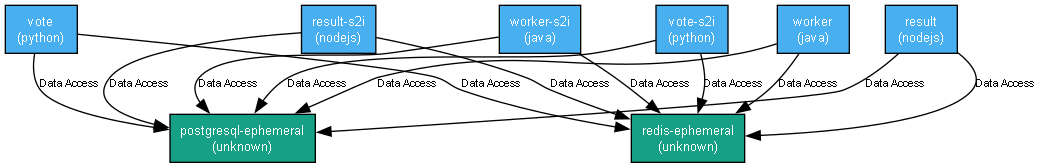
**📊 Business Impact:** Visual representation supports better understanding of system characteristics and aids in strategic planning.

**📊 Recommendations:** Review diagram insights with technical teams to identify optimization opportunities and action items.

**📊 Technical Details:** Diagram type: Cicd Pipeline. Generated from comprehensive application intelligence analysis.

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

Component Relationships (Graphviz)



**📊 Context:** This graph diagram maps the detailed relationships between 8 application components, showing dependencies, communication patterns, and integration points.

**📊 Key Insights:** Moderate coupling between components. Good balance between integration and independence. Moderate system complexity with 8 components. Monitor growth patterns.

**📊 Business Impact:** Balanced component relationships support both integration and maintainability. Continue monitoring coupling trends.

**📊 Recommendations:** Review component boundaries and reduce unnecessary dependencies. Implement interface-based communication. Consider event-driven patterns.

**📊 Technical Details:** Component relationship graph: 8 components, estimated 16 relationships, 57.1% coupling density.

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

Security Flow Diagram (Mermaid)



**📊 Context:** This Security Flow Diagram provides visual insights into the application's architecture and operational characteristics.

**📊 Key Insights:** The diagram illustrates key aspects of the application that support technical decision-making and operational planning.

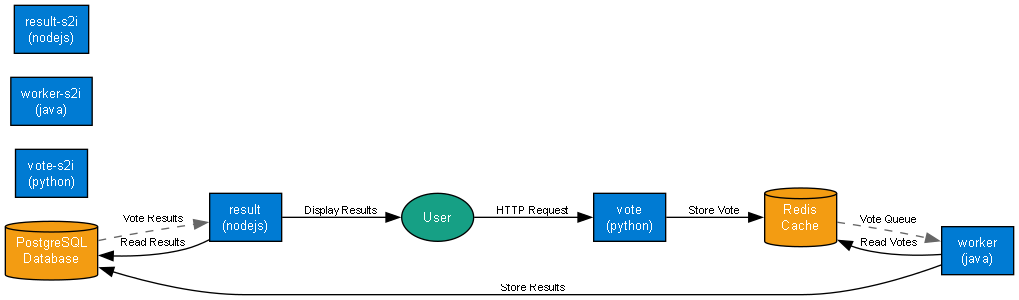
**📊 Business Impact:** Visual representation supports better understanding of system characteristics and aids in strategic planning.

**📊 Recommendations:** Review diagram insights with technical teams to identify optimization opportunities and action items.

**📊 Technical Details:** Diagram type: Security Flow Diagram. Generated from comprehensive application intelligence analysis.

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

Data Flow Diagram (Graphviz)



**📊 Context:** This data flow diagram illustrates how information moves through the application, showing 8 components and their interactions with 2 data sources.

**📊 Key Insights:** MODERATE data flow complexity. Moderate data flow complexity with 2 data sources. Standard data management practices should suffice. Good data-to-component ratio. Each data source serves multiple components efficiently.

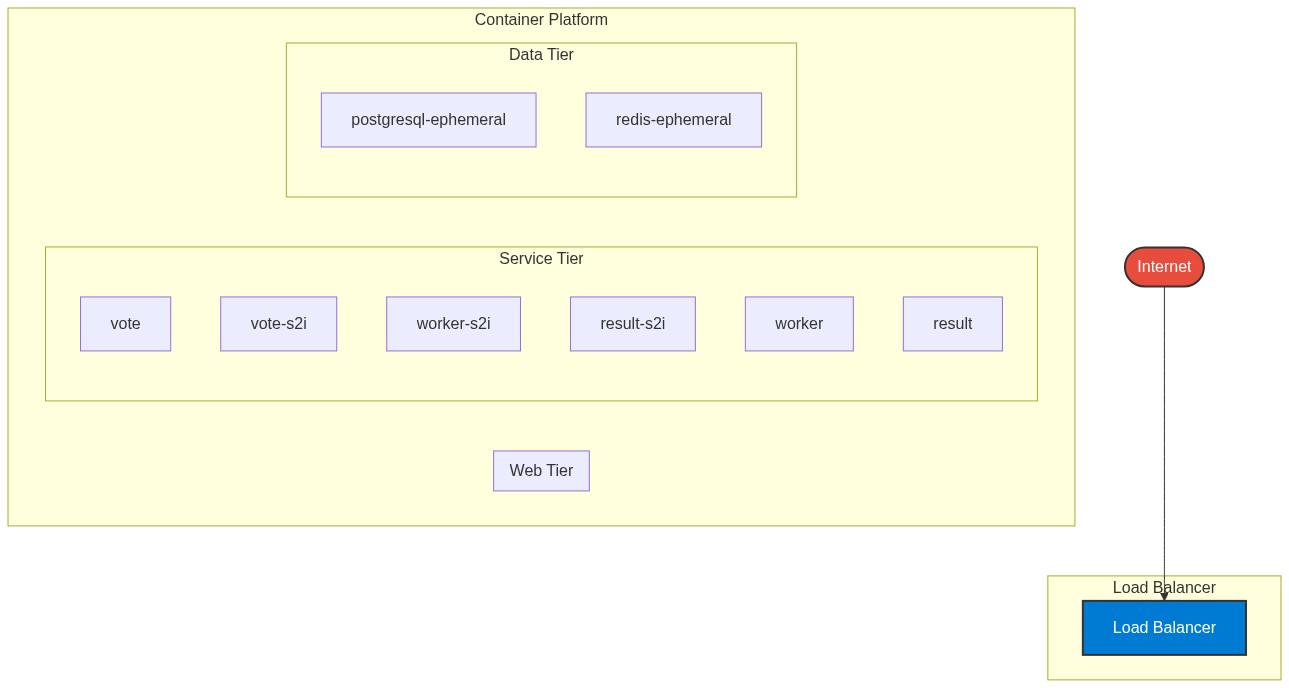
**📊 Business Impact:** Manageable data flows with standard complexity. Monitor performance and implement appropriate caching strategies.

**📊 Recommendations:** Monitor data flow performance and establish SLAs

**📊 Technical Details:** Data flow: 8 components, 2 data sources, moderate complexity. Data-to-component ratio: 2/8.

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

Deployment Architecture (Mermaid)



**📊 Context:** This Deployment Architecture provides visual insights into the application's architecture and operational characteristics.

**📊 Key Insights:** The diagram illustrates key aspects of the application that support technical decision-making and operational planning.

**📊 Business Impact:** Visual representation supports better understanding of system characteristics and aids in strategic planning.

**📊 Recommendations:** Review diagram insights with technical teams to identify optimization opportunities and action items.

**📊 Technical Details:** Diagram type: Deployment Architecture. Generated from comprehensive application intelligence analysis.

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

Risk Assessment Flow (Mermaid)



**📊 Context:** This Risk Assessment Flow provides visual insights into the application's architecture and operational characteristics.

**📊 Key Insights:** The diagram illustrates key aspects of the application that support technical decision-making and operational planning.

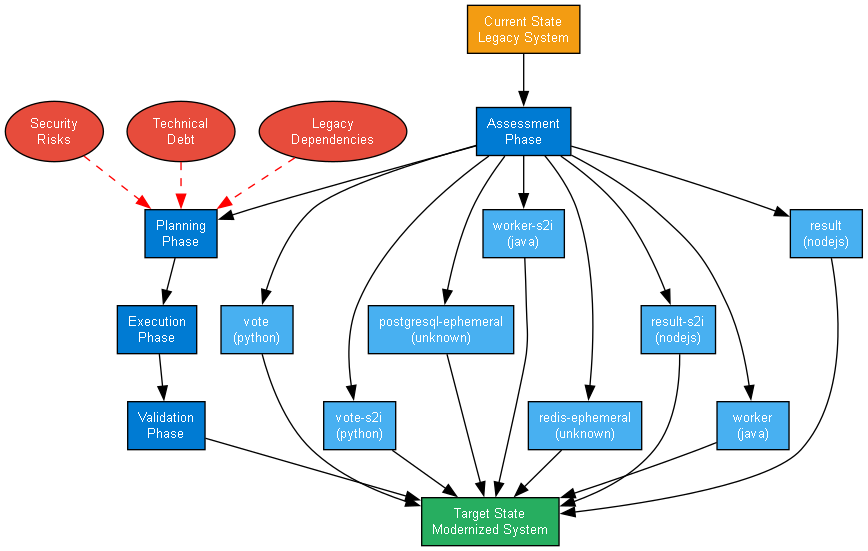
**📊 Business Impact:** Visual representation supports better understanding of system characteristics and aids in strategic planning.

**📊 Recommendations:** Review diagram insights with technical teams to identify optimization opportunities and action items.

**📊 Technical Details:** Diagram type: Risk Assessment Flow. Generated from comprehensive application intelligence analysis.

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

Migration Strategy (Graphviz)



**📊 Context:** This Migration Strategy Diagram provides visual insights into the application's architecture and operational characteristics.

**📊 Key Insights:** The diagram illustrates key aspects of the application that support technical decision-making and operational planning.

**📊 Business Impact:** Visual representation supports better understanding of system characteristics and aids in strategic planning.

**📊 Recommendations:** Review diagram insights with technical teams to identify optimization opportunities and action items.

**📊 Technical Details:** Diagram type: Migration Strategy Diagram. Generated from comprehensive application intelligence analysis.

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