



# Cost Assessment: J2EE to Jakarta EE Enterprise Migration



# Contents

Guide Updated: **March 2025**

<b>Introduction</b>	<b>3</b>
<b>Benefits of Migrating to Jakarta EE</b>	<b>3</b>
<b>Challenges in Migrating to Jakarta EE</b>	<b>4</b>
<b>Comprehensive Cost Analysis of J2EE to Jakarta EE Migration</b>	<b>4</b>
Legacy System Maintenance vs Migration Costs	5
Legacy System Costs:	5
Migration Investment Categories:	5
Scale-Based Cost Variations	5
Small-Scale Systems	5
Medium-Scale Systems	6
Large-Scale Systems	6
Cost Optimization Strategies	6
Return on Investment Considerations	6
<b>Tools and Resources for Migration</b>	<b>7</b>
Migration Tools	7
<b>Conclusion</b>	<b>7</b>

## Introduction

Many organizations rely on enterprise applications built on Java EE to support their daily operations. As Oracle moved Java EE to the Eclipse Foundation, these organizations will inevitably migrate to Jakarta EE to modernize their applications and benefit from the latest advancements. This migration is driven by the need to enhance performance, improve security, and ensure long-term support and stability. However, before embarking on this journey, it's essential to understand the costs involved and weigh them against the potential benefits. This report aims to provide a comprehensive cost assessment of J2EE to Jakarta EE enterprise migration, considering various factors that can influence the overall expenses.

Organizational culture plays a vital role in the success of any modernization initiative. Creating an environment that welcomes change while maintaining stability is essential for a smooth transition. This involves securing buy-in from all stakeholders, from executives to end-users, and fostering a collaborative approach to address challenges and ensure a successful migration.

## Benefits of Migrating to Jakarta EE

Migrating to Jakarta EE offers numerous benefits that can positively impact an organization's operational efficiency, agility, and competitiveness. These benefits often outweigh the costs associated with the migration process, making it a worthwhile investment for organizations seeking to modernize their enterprise applications.

Here are some key benefits of migrating to Jakarta EE:

- **Enhanced Features and Performance:** Jakarta EE provides access to the latest features, updates, and improvements in enterprise Java technology, leading to enhanced application performance, scalability, and reliability.
- **Improved Security:** Jakarta EE runtimes incorporate the latest security enhancements and address known vulnerabilities in older Java EE versions, improving the overall security of enterprise applications and reducing the risk of security breaches.
- **Cloud-Native Capabilities:** Jakarta EE is designed to support cloud-native development and deployment, enabling organizations to realize the benefits of cloud computing, such as scalability, elasticity, and cost-effectiveness.
- **Long-Term Support and Stability:** Jakarta EE is an open-source platform with a vibrant community and long-term viability from the Eclipse Foundation, ensuring ongoing maintenance, updates, and stability for enterprise applications.
- **Open and Collaborative Environment:** Jakarta EE fosters a more open and collaborative development environment compared to Java EE, encouraging community involvement and contributions, which can lead to faster innovation and broader support.
- **Microservices Support:** While microservices are a popular architectural style, they might not be suitable for every application. Jakarta EE provides the flexibility to support both

monolithic and microservices architectures effectively, allowing organizations to choose the approach that best aligns with their needs and goals.

## Challenges in Migrating to Jakarta EE

While migrating to Jakarta EE offers significant benefits, organizations need to be aware of the challenges involved in the migration process. Understanding these challenges can help organizations plan effectively, allocate resources appropriately, and mitigate potential risks.

Here are some common challenges in migrating from J2EE to Jakarta EE:

- **Namespace Changes:** One of the most significant changes in Jakarta EE is the shift from the `javax.*` to `jakarta.*` namespace. This change affects numerous libraries and frameworks, requiring code refactoring and updates to ensure compatibility.
- **Dependency Management:** Managing dependencies can be challenging during the migration process, as some libraries may still be using the old `javax.*` namespace, while others have migrated to `jakarta.*`. This can lead to conflicts and require careful dependency management to avoid issues.
- **Compatibility Issues:** Ensuring compatibility between different versions of Jakarta EE and with existing libraries and frameworks can be complex. Organizations need to thoroughly test their applications to identify and address any compatibility issues that may arise during or after the migration.
- **Technology Compatibility Kit (TCK):** The Technology Compatibility Kit (TCK) is a suite of tests used to verify the compatibility of Jakarta EE implementations. Organizations should adopt Jakarta EE runtimes that pass all the TCK tests for maximum compatibility and reliability.

## Comprehensive Cost Analysis of J2EE to Jakarta EE Migration

The transition from J2EE to Jakarta EE represents a significant investment for organizations, encompassing both direct migration costs and the opportunity costs of maintaining legacy systems. This analysis examines the full spectrum of expenses organizations should consider when planning their migration strategy.

## Legacy System Maintenance vs Migration Costs

Organizations must weigh the ongoing costs of maintaining legacy J2EE systems against the investment required for Jakarta EE migration:

### Legacy System Costs:

- Maintaining outdated hardware with diminishing vendor support and increasing compatibility issues
- Extended development cycles due to architectural limitations and complex codebases
- Heightened security vulnerabilities requiring constant patching and monitoring
- Premium rates for developers with legacy expertise
- Growing compliance gaps as regulatory requirements evolve, particularly concerning DORA and CRA metrics

### Migration Investment Categories:

- **Development and Implementation**
  - Code refactoring to accommodate namespace changes (javax.\* to jakarta.\*)
  - Dependency updates and conflict resolution
  - Infrastructure modernization including server updates and middleware adjustments
  - Development team training and potential certification costs
- **Quality Assurance**
  - Comprehensive testing environments and tools
  - Performance validation across different load scenarios
  - Security assessment and compliance verification
  - Integration testing with existing systems
- **Operational Impact**
  - Planned downtime management and associated revenue effects
  - Temporary parallel system operation costs
  - Additional infrastructure during transition periods
  - Staff overtime and contractor expenses

## Scale-Based Cost Variations

Migration costs scale with system complexity and organizational size:

### Small-Scale Systems

- Shorter development cycles with focused teams
- Limited infrastructure modifications
- Streamlined testing requirements
- Typical cost drivers: basic refactoring and essential training

## Medium-Scale Systems

- Moderate complexity in code updates
- Selected infrastructure modernization
- Expanded testing scope
- Typical cost drivers: team expansion and infrastructure updates

## Large-Scale Systems

- Extended development timelines
- Significant infrastructure investments
- Comprehensive testing protocols
- Typical cost drivers: complex integrations and extensive security measures

## Cost Optimization Strategies

Organizations can optimize migration costs through:

- **Phased Implementation**
  - Prioritizing critical components
  - Distributing costs across multiple budget cycles
  - Allowing for incremental learning and adjustment
- **Tool Utilization**
  - Eclipse Transformer for automated code conversion
  - OpenRewrite for systematic refactoring
  - Modern IDEs supporting efficient development
  - Frameworks like Spring and Hibernate for simplified implementation
- **Resource Management**
  - Strategic timing of infrastructure updates
  - Efficient allocation of development expertise
  - Balanced use of internal and external resources
  - Knowledge transfer protocols to minimize long-term support costs

## Return on Investment Considerations

While migration costs are substantial, organizations should evaluate them against long-term benefits:

- Reduced maintenance expenses through modern architecture
- Enhanced security features decreasing incident response costs
- Improved performance leading to better resource utilization
- Access to broader talent pools with current technology skills
- Simplified compliance management reducing regulatory overhead

The total cost of migration varies significantly based on system scale, complexity, and organizational readiness. Small systems might complete migration within modest budgets, while large, complex systems require substantial investment across multiple years. Organizations should conduct thorough assessments of their specific circumstances to develop accurate cost projections and implementation timelines.

## Tools and Resources for Migration

A wide range of tools and resources are available to support organizations in their J2EE to Jakarta EE migration journey. These tools can help automate certain tasks, reduce manual effort, and potentially lower the overall migration cost.

Here are some examples of tools that can be helpful during the migration process:

### Migration Tools:

- **Eclipse Transformer:** This tool can automate code refactoring, dependency updates, and other migration tasks, reducing manual effort and potential errors.
- **OpenRewrite:** OpenRewrite is another automated refactoring tool that can help with migrating code and dependencies to Jakarta EE.
- **Payara Upgrade Advisor:** Payara's Upgrade Advisor tool is a Maven utility that will scan your application code and give you a report of changes to expect with an upgrade to Jakarta EE.

## Conclusion

The migration from J2EE to Jakarta EE represents an important transformation that balances technical debt against future opportunities. While the comprehensive cost analysis demonstrates significant upfront investment requirements, organizations must consider this migration as an essential step toward technological modernization rather than an optional upgrade.

The decision to migrate involves careful consideration of multiple factors:

- Current maintenance costs and technical limitations of legacy systems
- Available internal technical expertise and resource capacity
- Business-critical application dependencies
- Regulatory compliance requirements
- Long-term strategic technology goals

A successful migration strategy requires a balanced approach that:

- Prioritizes applications based on business value and technical urgency
- Creates a realistic timeline that accounts for resource constraints
- Establishes clear success metrics aligned with business objectives
- Maintains operational stability throughout the transition
- Builds in flexibility to adapt to changing requirements

Organizations should recognize that while the migration process presents challenges, the cost of inaction - through increased maintenance expenses, security vulnerabilities, and competitive disadvantage - often exceeds the investment required for modernization.

For organizations seeking to accelerate their migration journey while minimizing risk, Payara's Migration Accelerator service provides specialized expertise and proven methodologies. This service helps organizations navigate complex migration challenges, optimize costs, and ensure business continuity throughout the transition process. Payara's team can help organizations achieve their modernization goals efficiently and effectively by combining deep Jakarta EE expertise with practical migration experience.

The path to Jakarta EE modernization is well-defined, and with proper planning and support, organizations can successfully navigate this essential transformation while maximizing their return on investment.



## Interested in Payara? Try Before You Buy



The banner features two laptops. The left laptop displays the Payara Enterprise interface, with the 'payara® ENTERPRISE' logo above it. The right laptop displays the Payara Cloud interface, with the 'payara cloud' logo above it. A central orange button with the text 'FREE TRIAL' is positioned between the two laptops. Below each laptop is an orange button: 'PAYARA SERVER FREE TRIAL' on the left and 'PAYARA CLOUD FREE TRIAL' on the right. The background is split into orange and dark blue sections with decorative elements like fish and clouds.



**[sales@payara.fish](mailto:sales@payara.fish)**



**UK: +44 800 538 5490**  
**Intl: +1 888 239 8941**



**[www.payara.fish](http://www.payara.fish)**

Payara Services Ltd 2025 All Rights Reserved. Registered in England and Wales; Registration Number 09998946  
Registered Office: Malvern Hills Science Park, Geraldine Road, Malvern, United Kingdom, WR14 3SZ