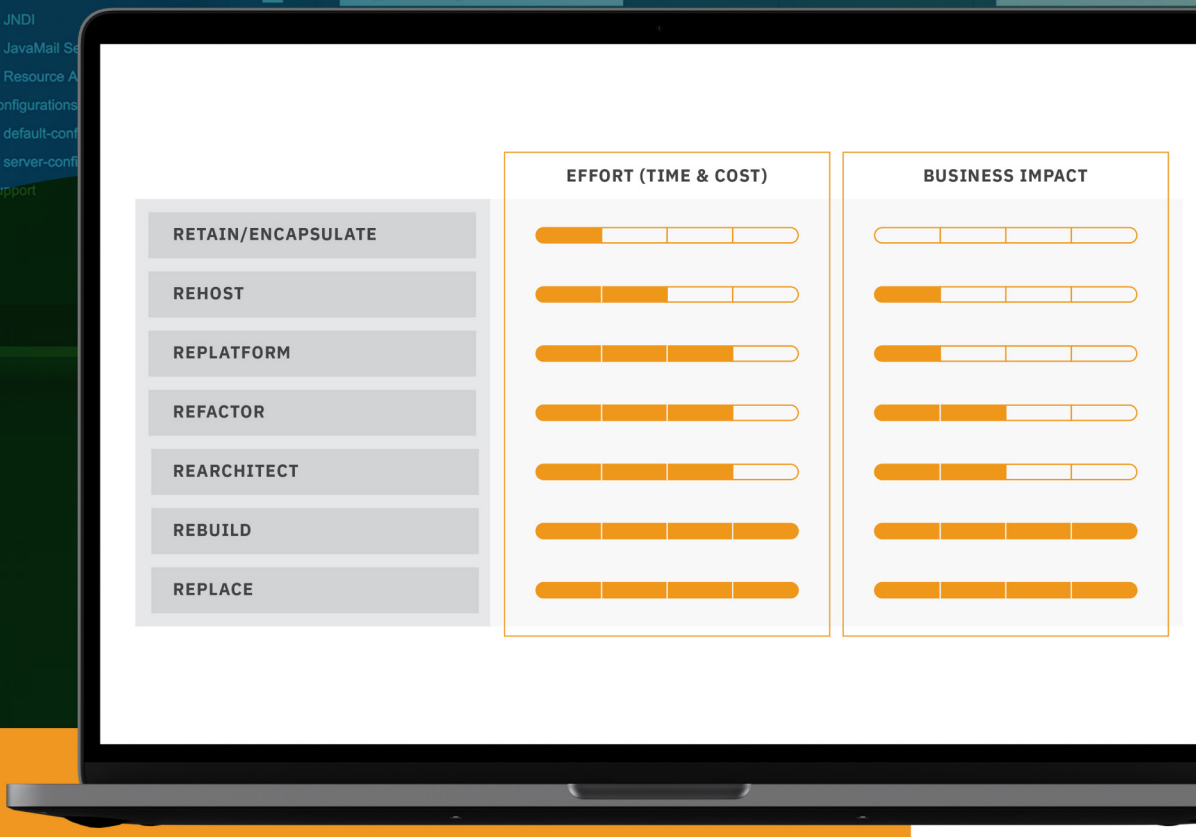
A Business Guide to Legacy Application Modernization For The Cloud Era



The Payara® Platform - Production-Ready, Cloud Native and Aggressively Compatible.

User Guide

Contents

A Business Guide to Legacy Application Modernization for the Cloud Era

Digital transformation, largely enabled by the commoditization of massive compute resources by cloud computing vendors has changed the way applications are developed, deployed and maintained. It has allowed for much faster application development iterations by organisations to meet ever changing and increasingly complex customer expectations.

The customised digital experiences pioneered by the likes of Amazon and Netflix has created a defacto standard of customer experience that all organisations aspire to in order to acquire and maintain customers. The COVID-19 pandemic also created a paradigm shift to more digital economic activities across the globe. Not all enterprise applications are positioned to take advantage of the massive benefits offered by the modern digital transformation paradigm.

Some organisations are saddled with legacy applications that will need to be modernised before they can take advantage of the benefits of the digital transformation brought about by the cloud. This paper looks at key guidelines on modernising such applications for the cloud native era.

We start by defining and looking at the characteristics of legacy applications, what legacy application modernisation and its benefits are and explore the various options to realising legacy application modernisation for the cloud native era. By the end of this paper, you will have a clear framework to assess the need for and undertake application modernisation for your organisation to take advantage of the benefits of the cloud native era.

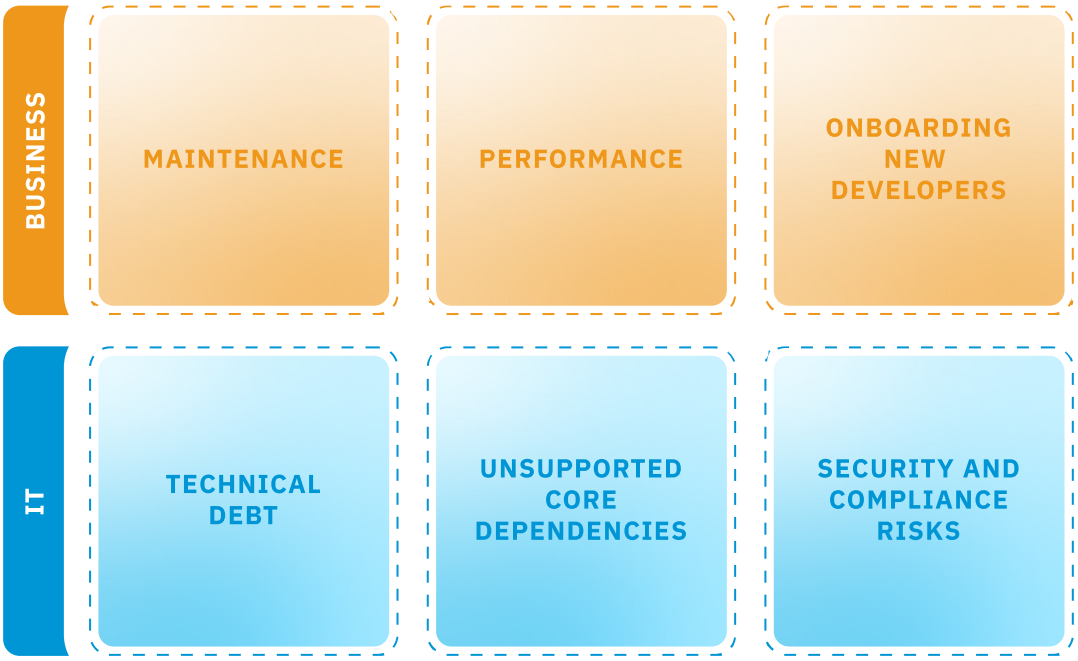
What Is a Legacy Application?

A legacy application can be defined as one that was developed with older technologies using outdated development paradigms. Gartner defines it as “an information system that may be based on outdated technologies, but is critical to day-to-day operations” of an organisation. A legacy application then, is one that is currently being used by an organisation but is in need of modernisation within the context of contemporary application development and deployment paradigms.

A legacy application will generally be working and delivering business value as it was intended, but at a much more diminished rate relative to the rate of change in customer expectations. For instance, a forty year old insurance policy and claims management system written in COBOL can still be used to manage insurance products, but will be almost impossible or expensive to use to deliver other customer experiences like self service, mobile and SMS integration among others.

A legacy application doesn't have to necessarily be one written in an older programming language. An application written in a contemporary language like Java can be considered legacy if the underlying technology, like J2EE is no longer available and/or supported. In effect, an application can be classified as legacy when it is unable to take advantage of modern digital transformations enabled by the cloud to deliver exceptional customer experiences at reasonable costs to the organisation.

Identifying an application as legacy and as such, needing modernisation isn't always so straightforward in every organisational context, however. The following points will help you assess the need for modernisation of an application. The more an application ticks these checkmarks, the greater the likelihood of it being considered legacy.



Legacy Application Evaluation

An application can be classified as legacy by a critical analysis of the overall value it brings to the business. This value analysis can be done in two thematic areas of business and IT.

Business

From the business perspective, is the application a good fit, delivering business value and agile enough in its ability to meet ever changing customer expectations brought about by digital transformation? The following points should be analysed from the business perspective.

Maintenance

How hard is it to maintain the application? How fast can bugs be fixed? What is the turnaround time for adding new features? Do the costs of doing these grow exponentially or linearly? How satisfied are the staff maintaining this application? Is the application living up to expectation in delivering its primary function?

Performance

How well does the system perform? Are end users of the system satisfied with its performance? Are the clerks using the system happy with how fast it allows them to get their jobs done? How well does the application scale to meet a surge in demand?

Onboarding New Developers

How fast can we onboard new developers? Are there enough developers with the skills to maintain the application? How fast can we replace existing developers if and when they leave?

IT

From the IT perspective, what are the costs associated with running and maintaining the application, how increasingly complex is every development task that needs to be carried out on it and what are the security risks associated with the current state of the application. The following pointers can help drill down on these broad factors.

Technical debt

How much technical debt does the application have? And how much is piled on with each feature or bug fix? How much does this debt impact the performance of the application? How much will it impact future development?

Unsupported Core Dependencies

Are there external core application dependencies that are no longer supported? Are there any integrations that are no longer available to the application? Can the application keep delivering its core value in the absence of these integrations? Can these integrations and external libraries be replaced? How much will it cost to replace them?

Security And Compliance Risks

How secure is the application? How much of the underlying infrastructure is no longer developed or maintained? How up to date are the core components of the application? How compliant is the application with regulatory requirements?

As you can see, some factors can intersect the business and IT perspectives. For instance, application maintenance transcends both business and IT perspectives. The more an application ticks these checkmarks or scores poorly on them, the greater the likelihood of it being classified as legacy and thus in need of modernisation. Once the state of an application is clarified, the next logical question is what is legacy application modernisation? And what benefits can it bring to an organisation that undertakes it?

Legacy Application Modernisation

After identifying an application as legacy, the next step is to modernise it. Legacy application modernization is a catch-all term that encompasses the gamut of processes involved in identifying an application as legacy (first part of this paper) to rearchitecting it to take advantage of digital transformations to deliver superior customer experiences.

Technically, legacy application modernisation, according to VMWare is "the practice of updating older software for newer computing approaches, including newer languages, frameworks and infrastructure platforms." Legacy application modernisation then, is a way of renewing an old system with cloud native capabilities to deliver exceptional customer experiences.

Advantages Of Legacy Application Modernisation

Identifying a system as legacy and accepting that it needs modernisation may not necessarily be enough to get buy-in from all decision makers. So let us explore some advantages that can be attributed to legacy application modernisation.

Costs Reduction

There are a lot of costs associated with running a legacy system. These costs can include

Specific Hardware

Some old systems were designed to run on specific hardware. Such hardware might not be readily available on the market and thus very expensive to acquire and maintain. Modernising the application could eliminate this cost driver because the modernised application can run on readily available commodity hardware.

Slower Turnarounds

The entire development cycle for a legacy application might be very slow due to the arcane processes involved in each step. Adding a new feature or fixing a bug might take a significant amount of time. Modernising the application can help it take advantage of current devops paradigms for a faster turnaround.

Dearth Of Developers

Legacy systems might be built with very outdated frameworks or programming languages that have a dearth of competent developers to carry on maintaining the system. Modernising such a system will help reduce such costs by taking advantage of readily available developers once the application is modernised to a much current infrastructure.

Performance

A legacy application might have major performance bottlenecks due to a combination of factors such as technical debt, outdated hardware, bugs and outdated runtimes. Modernising legacy applications with cloud native capabilities can allow them take advantage of the commoditization of cloud infrastructure to scale better to meet demand, become more resilient and improve uptime significantly.

Security

Modernising legacy applications can eliminate whole surfaces of security risks such as outdated libraries, runtimes and language. It can also allow for real time monitoring of the application to enable faster reactions to security threats. For instance, a cloud native modernised legacy application can employ modern continuous security scan services that will keep watch over all layers of the application such as application code, deployment containers and platform. Modernising to current versions of all application components means the application will have security patches and bug fixes for all such components.

Better Employee Experience

Oftentimes employees that work on legacy systems may not be very satisfied with their jobs because of how tedious dealing with such outdated software can be. Modernising legacy applications with cloud native capabilities can increase employee satisfaction when working on the system. This in turn can help reduce employee churn.

Deliver Superior Customer Experiences

Modernising legacy applications for the cloud can enable an organisation to deliver superior customer experiences. A cloud enabled application can take advantage of the massive array of cloud services to help predict, meet and exceed customer expectations, thereby increasing customer satisfaction, reducing the customer maintenance costs and increasing overall profitability.

Reduced Compliance Risks

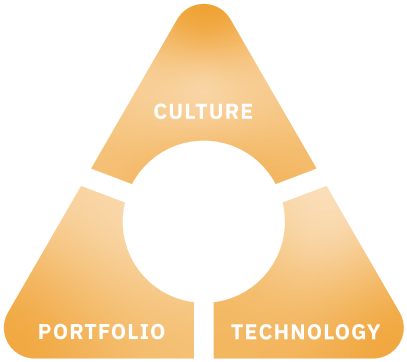
Every organisation operates within some regulatory framework. And there may be regulatory requirements that the organisation needs to meet as part of delivering its core mandate. Legacy applications can be hard to amend to keep up with changing regulations. Modernising such applications can significantly reduce the cost of regulatory compliance by making the application much more amenable to changing regulations. Being compliant to regulations reduces the risk of the organisation incurring fines for regulatory breaches.

A study by NTT Data Consulting found that 63% of surveyed banks had challenges of integrations with external systems and 78% of spendings on core deposit systems is on legacy application maintenance and compliance. Such major challenges leave little room for innovation, which can eventually lead to stagnation in quality service delivery.

These are by no means an exhaustive list of advantages that can be derived from modernising legacy applications. Every organisation can realise a lot more benefits that will be specific to their own context. Once the benefits of legacy modernisation is considered, the next step is to explore a broad framework for planning and carrying out the modernisation effort.

Modernisation Framework

Legacy application modernisation can be a very complicated and expensive exercise that must be carried out with as much planning as possible. This section of the paper offers a framework that can help you plan a path to modernising your legacy application. The framework consists of three broad thematic areas - culture, portfolio and technology.



Culture

Modernising a legacy application, depending on the size of the application, can be an exercise that impacts everyone in the organisation. As such, it is very important to start the process by aligning the company culture to the process. Getting everyone in the company to be onboard, from bottom up is critical to ensuring the success of the process. It is also important to set clear expectations before, during and after the process. This is because the modernisation effort, depending on the actual type of modernisation process chosen, could result in a significantly different software. This means current users of the software need to have their expectations calibrated to avoid major cognitive dissonance as a fallout of the modernisation effort.

Internal teams may also need restructuring as part of the modernisation process. This is important to communicate upfront and get buy-in from all concerned parties before the actual process begins. Having the people aligned with the modernisation process is the single most important part of the process and should be handled with utmost care and tact. As every organisation is the sum of its people, it is important to get their buy-in before undertaking a significant exercise such as modernisation of a legacy application, a process that can have far reaching effects for everyone.

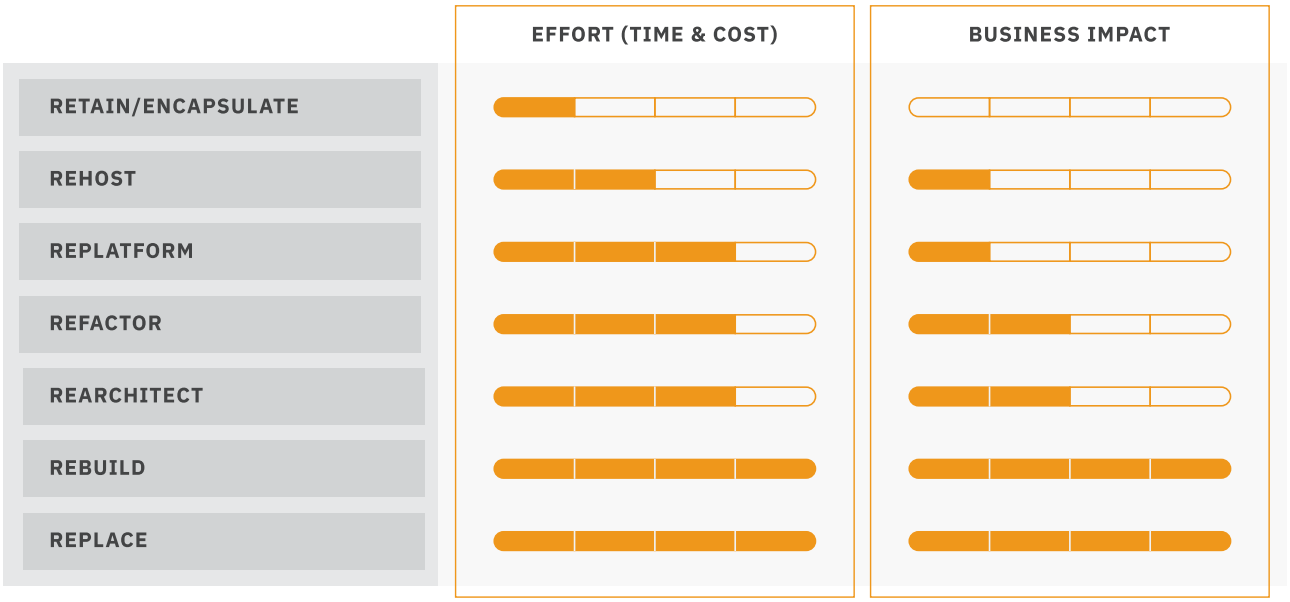
Portfolio

Legacy application modernisation is a straightforward process if your organisation has a single application under consideration. However, when there are more than one such application due for modernisation, it is important to prioritise the resources available for the process in a way that maximises the potential outcome. Legacy applications in the organisation's portfolio should be prioritised in order of greatest returns to the least and least cost to the greatest. Such a matrix will help determine which application should be modernised first.

As almost every organisation has limited resources, allocating same to a complex exercise as application modernisation should be carried out after extensive considerations of all legacy applications in the portfolio. This will help allocate resources to the project with the best chance of succeeding.

Technology

Technology is the third in the trio of the legacy modernisation process. Technology refers to the various options available for the actual modernisation effort. As every organisation and application are different, there are different ways that the modernisation effort can be carried out. Depending on the goal of the effort, a given technology choice or combination of choices can be made. There are seven broad options that can be used to modernise a legacy application. These options are not mutually exclusive and can be used in combination with each. Let us explore them, from the least complicated to the most complicated and expensive.



Retain / Encapsulate

This option entails retaining the current legacy system and exposing its core functions for consumption by greenfield microservices (or even monoliths) with cloud native features built in. This allows abstracting the core features of the legacy application behind microservices that can offer all the benefits of cloud native applications for users. This is generally the least complicated and expensive option. The legacy system can still be maintained/hosted as is, but have the application consuming the exposed core functionality from it deployed to the cloud for better scale, performance and resilience.

Rehost

As one of the cost drivers of running a legacy system is related to hardware and infrastructure, modernising a legacy application could entail rehosting it on a different hardware infrastructure. The new infrastructure could be a new physical, virtual or public cloud platform. The goal is to optimise the legacy application to take advantage of hardware for better performance and security.

Replatform

Replatform is migrating the legacy application to a latest version of its runtime without making any extensive changes to the code and its structure. Replatform is bringing the underlying frameworking, library or platform of the application up to date to take advantage of new improvements. For example, a legacy application written in Java 5 on the J2EE Platform can be migrated to Java 11 and Jakarta EE 8 with minimal changes to the code structure. A seemingly minor such update means lots of improvements for the application with all the optimizations that have gone into the JVM from Java 5 - 11. Upgrading to Jakarta EE 8 also means escaping all the security flaws in the long unsupported J2EE runtime.

Refactor

Refactoring entails optimising and restructuring the application internally, without impacting its external functionality. The goal of this type of modernisation is to improve the internal performance and maintenance of the application without causing significant downtime. Refactoring can be combined with replatform in a staggered modernisation exercise where the former follows the latter after some time.

Rearchitect

Rearchitecting means fundamentally altering the code of the application to shift it to a new architecture with the goal of exploiting the benefits of the new architecture. For example, the J2EE application mentioned above can be rearchitected into a set of cloud native microservices that together perform the same function as the current legacy application. As microservices however, the rearchitected application can take full advantage offered by the cloud to accentuate the customer and user experience of the application.

Rebuild

Rebuild entails redesigning or completely rewriting the application from scratch but still preserving its core functions. Rebuilding is different from rearchitecting in the sense that the former entails a sort of one to one rewrite while the latter entails a change in how the application building blocks are arranged. Rebuilding can be challenging because it has to do with maintaining the architecture of the application but rewriting it to take advantage of new capabilities offered by the underlying language, framework or runtime.

Replace

Replacing is the most expensive and hardest of all the modernisation options. Replacing means to eliminate the current legacy application, replacing it with a new one that considers current requirements. Fully replacing a legacy application is a very complicated and expensive exercise that can take a significant amount of time to complete. However, it is also the most effective of all the modernisation options.

Deciding which modernization effort to pick is a domain dependent decision. However, care must be taken to choose the most efficient effort for the given context. There is the temptation to jump on the bandwagon to adopt an option that might prove much more costly along the way than the cost of running the legacy application.

A Note on Microservices and Monoliths

Undoubtedly almost all legacy applications are large, complex monoliths that were written years ago. Monolithic architecture was generally the de facto architecture of the time. You might be tempted to want to replace your legacy application with microservices as that is the current and popular architecture option. However, my advice is to pick an architecture based on the business value it brings in within your organisational context.

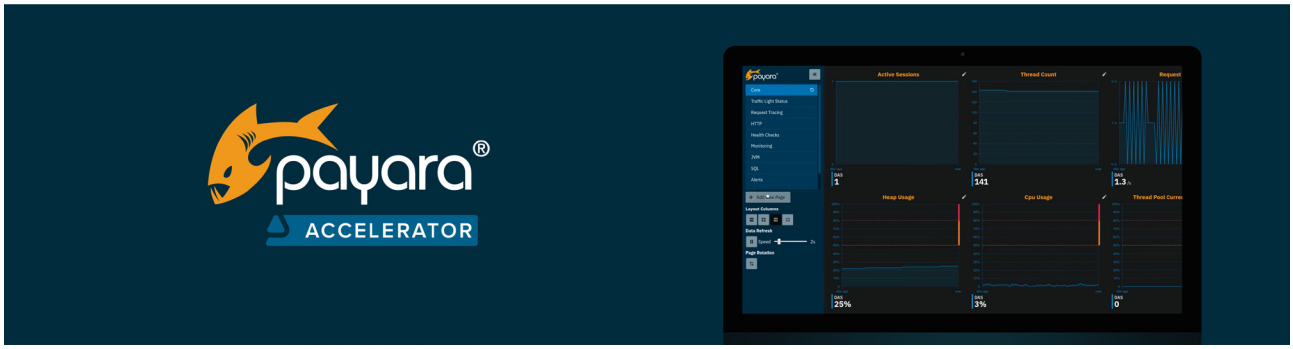
A legacy monolith application can be effectively modernised without needing to rewrite everything as microservices. Platforms like Jakarta EE allow all types of applications to take advantage of cloud native features and benefits without tying you to a given architecture. In the end, my advice as to which modernisation effort to choose is to make an assessment of the legacy application within the context of organisational goals. Doing so will clarify what modernisation costs the organisation can and is willing to incur in the exercise.

Conclusion

Legacy application modernisation is something every organisation will eventually have to contend with as it ages. The decision to breathe new life into an existing system at the lowest cost possible is one that requires a lot of contextual analysis. As discussed in this paper, legacy applications will start costing more than the revenue they bring in the older they get. This paper explored the costs associated with running legacy applications and the benefits that can be expected by modernising such applications.

The decision to modernise an application is a far reaching one that should be taken within the framework proposed in this paper, that is culture, portfolio and technology. Aligning organisational culture, prioritising legacy application portfolio and picking the best technology means of modernisation can significantly increase the chances of a successful modernisation exercise.

If you are currently considering legacy application modernisation and need technical expertise with the process, Payara Accelerator can help you optimise and accelerate the process for better profitability. Feel free to reach out to us and we can help you optimise your application stack for the cloud native era to deliver superior customer experiences.



sales@payara.fish

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

www.payara.fish