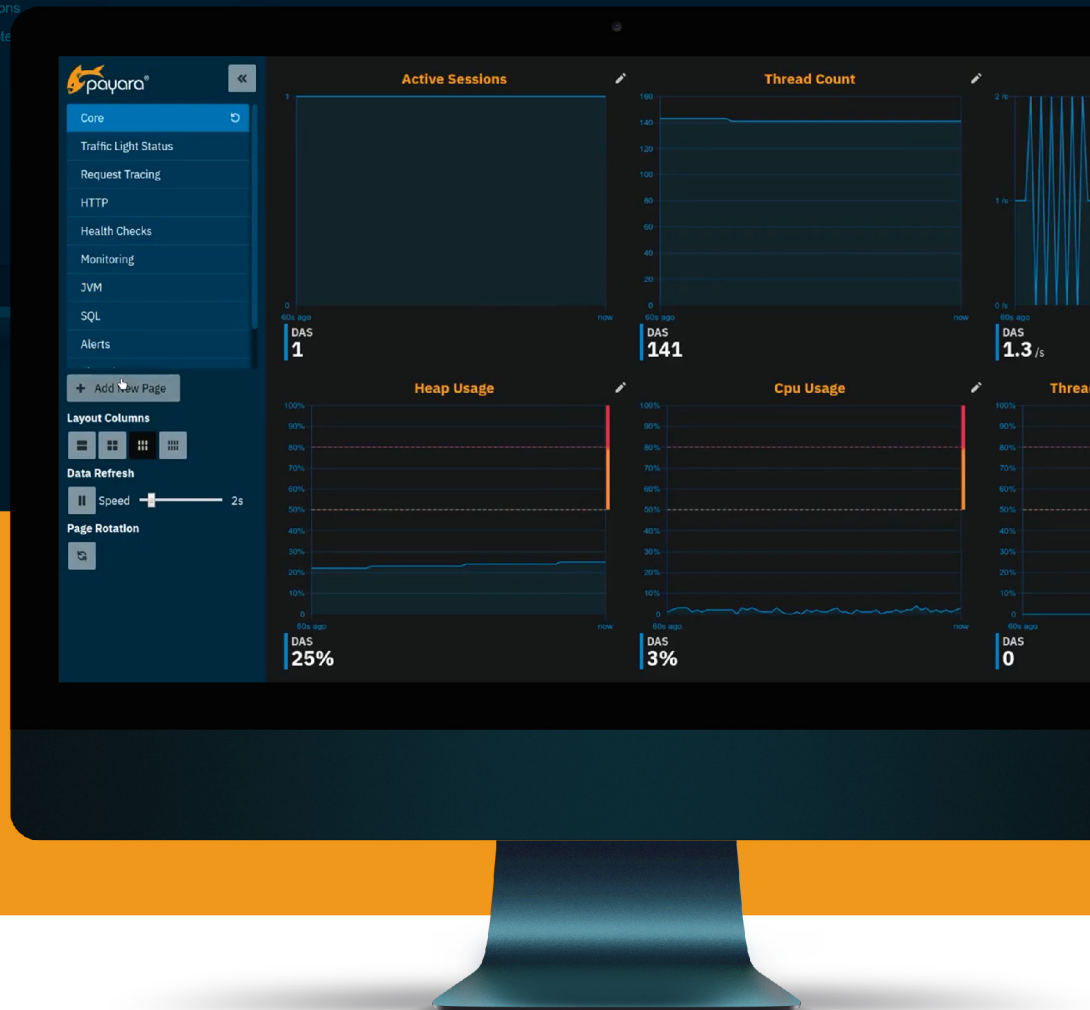




A Business Guide to Enterprise Development Options on the Java Platform



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

User Guide

Contents

The Java Platform	1
The Java Platform	2
What is an Enterprise Application?	2
Components of an Enterprise Application	3
What is an Enterprise Development Platform?	3
Development Platform Vs Framework	3
Features of a Good Development Platform	4
Accessible	4
Well Documented	4
Momentum	4
Licence	4
Tooling	5
Commercial Backing	5
Longevity	5
Java Enterprise Development Platforms	6
Jakarta EE (Formerly Java EE)	6
Spring Framework	6
Grails	7
Play	7
Others	7
Summary	8

The Java Programming Language has come a long way since its start in 1995. Originally called Oak, Java was ushered to the public limelight by then Sun Microsystems in May 1995, touting the revolutionary “Write Once Run Anywhere” capability of the language. Originally developed for creating applications for household devices like handheld entertainment controllers, the Java Platform today has evolved to become a very capable, powerful platform for developing almost any imaginable application.

This business guide takes a look at how to assess enterprise application development options on the Java Platform. As far reaching as Java is, we first define what is meant by Java Platform within the context of this paper, followed by business definitions of what an enterprise application and enterprise development platform/framework are. We then move on to determine some key basic requirements of a good enterprise development platform, and then proceed to look at some of the popular options for enterprise development platforms/frameworks on the Java Platform. By the end of this guide, you will have a good framework for evaluating Java application development platforms for your next project.

The Java Platform

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.

These expectations must be met with ever complex applications that can be consumed in a myriad of ways by end users. Application development entails the use of a computer programming language. Java is a statically typed, high level object-oriented programming language from the C line of languages. It is portable across operating systems as it is executed in a Java Virtual Machine or JVM. This makes it possible to develop and run Java applications on any operating system and CPU architecture on which the JVM can run.

Over time, the JVM has come to support other languages apart from Java itself. This has in turn spurred many innovations on the JVM. This development has also given rise to some confusion with regards to terminology. Java is sometimes used to refer to the language itself, other times to refer to the JVM, and yet still at other times refer to the entire ecosystem. This is also dependent on who is speaking and the context. But what exactly is the Java Platform.

The Java Platform

Within the context of this guide, the Java Platform refers to the Java Language, the JVM that executes that language and all other programming languages that the JVM can execute. It also refers to all software development frameworks that are built on top of the Java language and other JVM based languages. Effectively, we use the term Java Platform to refer to the gamut of technologies that are all eventually executed by the Java Virtual Machine. The Java Platform today supports the development of all kinds of enterprise applications. But what is an enterprise application?

What is an Enterprise Application?

Software applications come in different types and complexity. Some applications are designed to be consumed internally within an organisation. Others are designed to be consumed by billions of people across every imaginable device. Others are developed by individual hobbyists with no intention of ever making them available to anyone else. Many are developed by an organised collection of developers in an organisation.

The umbrella term application therefore, is broad and needs to be broken down when referring to applications within a business context. Organisations that primarily deal in the business of creating software and making them available for public use in return for revenue have a different definition and sometimes terminology for referring to their products. The general term used by such organisations is enterprise applications.

An enterprise software application within the context of this guide, can be defined as a non-trivial, relatively complex, multi-tier application developed by an organisation to solve a specific problem for a given target market in return for some value. Whatever the organisation and type of software provided, all enterprise applications do have some common traits that makes them non-trivial to develop.

Components of an Enterprise Application

A typical application would have three tiers - the user interface, the middle layer and the data storage. The UI is where the end users actually interact with the application. The middle layer is the “orchestration” part that handles interactions between the UI and the data warehouse. The data warehouse refers to the database management system used to store application generated data into a permanent storage mechanism. Whether an application is a monolith or microservice, these tiers generally run through the full application.

The UI could be a traditional HTML page, for applications developed for human interaction. Or it could be RESTful endpoints, for applications designed for other applications to consume. The term “user interface” has evolved to include the point at which final clients of a given application interact with that application. The middle or service layer typically contains artefacts that handle security, data warehouse transactions, interactions with other services (internally and/or externally) among others.

What is an Enterprise Development Platform?

As stated above, enterprise applications are non-trivial to develop. Getting all the requirements right is a very challenging exercise that can take time. Even though the Java Language is used for developing software applications, it is still a fairly low level tool. It does not provide all the constructs (or does but at a very low level) needed for developing applications that meet the definition outlined above. As such, further abstractions are needed on top of the base language to provide these constructs in a way that they can easily be consumed by developers creating enterprise applications.

The collection of abstractions on top of the base Java language that provide the constructs for easily fulfilling the different requirements of an enterprise application is what we call an enterprise development platform within this guide. The term development platform is sometimes used interchangeably with frameworks. But are they the same?

Development Platform Vs Framework

A framework is also an abstraction on a programming language aimed at helping software application developers create applications in a productive way. However, a framework is mostly an “end game” in the sense that you can hardly build another abstraction on top of it.

A development platform, on the other hand, is both a framework but can also have abstractions built on top of it for other niche specific cases. In the end, the differences in the terminologies don’t really matter in real life. But for the purposes of this guide, we will stick to using development platforms when referring to both. Ok so now we know what a development platform is, but what makes a good one from a business perspective?

Features of a Good Development Platform

Every software development platform will describe itself as the silver bullet for software application development. However, from a business perspective, how do you pick a platform for your projects? What are the features of a good development platform? Let us explore those next in no particular order.

Accessible

A good development platform should be easily accessible to the average developer. Getting started with it should not be a ceremonial affair. The easier it is to get started with a development platform, the faster developers can get to work and start delivering business value. The community around the development platform should also be accessible and inclusive for different persons and opinions.

Well Documented

A good development platform should be well documented. For a developer, the documentation is always the single source of truth for how to use the platform. A good development platform should have adequate, updated documentation that ideally holds a fine balance between depth and breadth. A well documented platform allows developers to navigate problems with the platform much faster, thereby reducing blockers. The documentation should not necessarily be limited to text. It can be a mix of quick start guides, videos and detailed documentation for each feature of the platform. In fact the more and better organised the documentation, the better the platform.

Momentum

A good development platform should have momentum in the form of an active community. No matter how good a development platform is, without an active community of users the platform would be no good in the long run for adoption. A good platform should have an active community of users that are using and evolving the platform through the filing of bugs, creation of libraries to fill in gaps and general contributions to the platform in terms of code. Having major conferences organised around the platform is also great.

Licence

A good platform is one that is ideally licensed under permissive open licences. It should be easy to adopt, use and alter the platform without having to worry about flouting the licence of the platform. Depending on the specific platform, at a minimum, individuals and organisations should be able to use the platform without fear or excessive legal clouds hanging over their heads.

Tooling

A good development platform should be one that has adequate tooling of all sorts. Such tooling should cover the gamut of the software development process, from scaffolding to build tools to deployment tools to anything in-between. In effect, the platform should have as much tooling as possible that automates the mundane tasks of using it. The tooling could themselves be free or commercial. The critical point is to have tooling available for developers using the platform.

Commercial Backing

A reliable platform should have commercial backing. This commercial backing does not necessarily have to be directly linked to the platform. It can be in the form of selling of services, tooling or other commercial activities from which some of the proceeds are invested into the platform, directly or indirectly. No matter the lofty ideals of any development platform, it cannot endure without some form of economic activity.

Longevity

A good development platform is one that can be expected to be available for the foreseeable future. As enterprises have to support their applications for a long time, it is vital that the development platform on which the application is developed is one that will endure for a reasonable period of time.

There are other factors that can be considered on a case by case basis. But the ones listed above are the most general and permeating to consider when adopting a development platform. Within the context of the above factors, let us now take a look at the major enterprise application development options available on the Java Platform.

Java Enterprise Development Platforms

After identifying some key factors that should be considered in selecting a Java enterprise application development platform, we can now proceed to take a look at some of the major options available to you. These options are not listed in any particular order, and are discussed based on the features of a good development platform listed above.

Jakarta EE (Formerly Java EE)

The Jakarta EE (formerly Java EE) platform is arguably the most far reaching, oldest enterprise application development platform of all the options available on the Java Platform. Its history is closely tied to the history of the Java Language itself and dates back to not long after the introduction of Java. Built on the Java Standard Edition, Jakarta EE at its core is a collection of abstract specifications that forms a cohesive platform for developing multi-tier enterprise applications.

It has gone through many iterations and is now managed as a full community project by the Eclipse Foundation. Jakarta EE itself is the foundation on which other development platforms have been built. For example, its Servlet specification has been the most used web development API for a number of years now. In terms of accessibility, Jakarta EE can be bootstrapped as a normal Apache Maven project. The community is organised as a set of mailing lists for each specification and open to anyone interested in joining.

In terms of documentation, Jakarta EE with its long history has a good collection of official and community documentation that covers the full spectrum of the platform. A very large community of users across all industries, with a healthy number of conferences gives the platform a good momentum to evolve. It uses an open licence, has great tooling and has corporate backing in the form of the Eclipse Foundation. All in all, even though the Jakarta EE platform has had challenges in the past, it is currently a platform that ticks, to a significant degree, all the features of a good development platform from a business perspective.

Spring Framework

The Spring Framework is an enterprise application development platform that was originally inspired by the challenges the then Jakarta EE (then known as J2EE) faced. What started as an inversion of control container has grown to become a fully capable software development platform for all kinds of applications. Currently owned by VMWare with a significant community, Spring Framework is accessible, well documented and has a very large community and ecosystem around it. In fact Spring and Jakarta EE both influence each other in their evolutions.

Grails

Grails is a Rails (Ruby web framework) inspired development platform built on the Spring Framework. It primarily uses Apache Groovy as the target development language, though you can fully develop on it with Java. Over the years, Grails has lost a lot of momentum as a first choice for Java developers.

Play

Play is a Scala based development platform for creating web applications. It originally was written in Java but was rewritten in Scala. It has a niche community of users but has lost much of its original momentum over the past few years.

Others

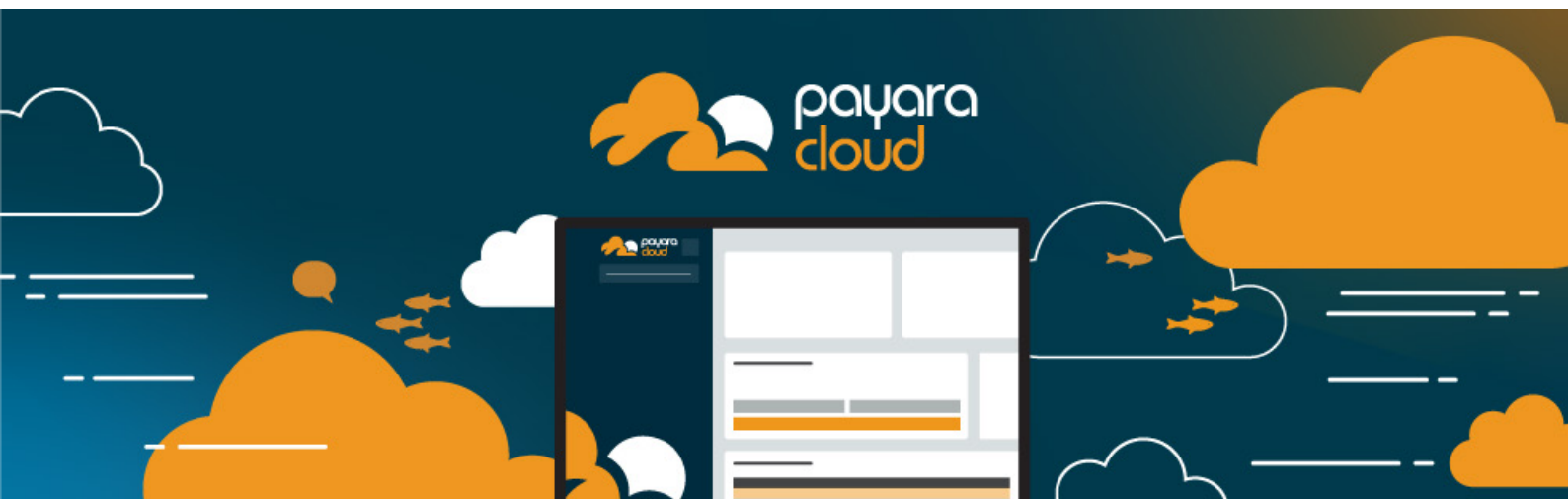
There are many other Java development frameworks in the ecosystem. However, they all differ in terms of offering an all round development platform. For instance Vaadin is a Java UI framework for developing enterprise applications. However, it provides just for the UI component of an application. As stated in the section on what an enterprise application is, within the context of guide, a development platform should be able to provide for the data, UI and middle layer.

Summary

Enterprise applications have become even more critical in a fast paced digital ecosystem spawned primarily by the commoditization of cheap cloud compute resources. Enterprises need to invest time and money in their applications. As such, it is important to pick a platform that offers the best return on investment for an organisation. In this guide, we looked at the Java Platform, defined what it is, defined what an enterprise application is and proceeded to identify the key features of a good enterprise application development platform. It is the hope of this author that next time when you are evaluating development platforms, you do so from a business perspective within the context of the outlined features and any other that is specific to your business case.

If you enjoyed this guide, please check out our other higher-level content around developing enterprise applications:

- [A Business Guide to Legacy Application Modernization For The Cloud Era](#)
- [Jakarta EE 10: What Decision Makers Need to Know](#)
- [A Business Guide To Cloud Deployment Options For Jakarta EE Applications](#)



[CLICK HERE](#)



sales@payara.fish



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



www.payara.fish

Payara Services Ltd 2023 All Rights Reserved. Registered in England and Wales; Registration Number 09998946
Registered Office: Malvern Hills Science Park, Geraldine Road, Malvern, United Kingdom, WR14 3SZ