



JAKARTA EE DEVELOPER SURVEY 2018



By Mike Milinkovich, Eclipse Foundation Executive Director

For many years, Java EE has been the de facto platform for enterprise systems—offering reliable multi-vendor standards, ubiquitous in datacenters across every industry, and delivering stability and scalability on the back-end of the world’s mission critical applications. Java EE provides essential middleware capabilities that have continually evolved over the years, often driven by feedback from the wider community. The additions of the Web Profile and CDI are prime examples of this. But the pace of innovation necessary to modernize enterprise systems for cloud-centric use cases requires a new governance model—a focus on faster release cycles, and supporting an open source, community-driven evolution of the platform. In order to accelerate this innovation process for a cloud-native world, Java EE technologies are being moved to the Eclipse Foundation where they will evolve under the Jakarta EE brand.

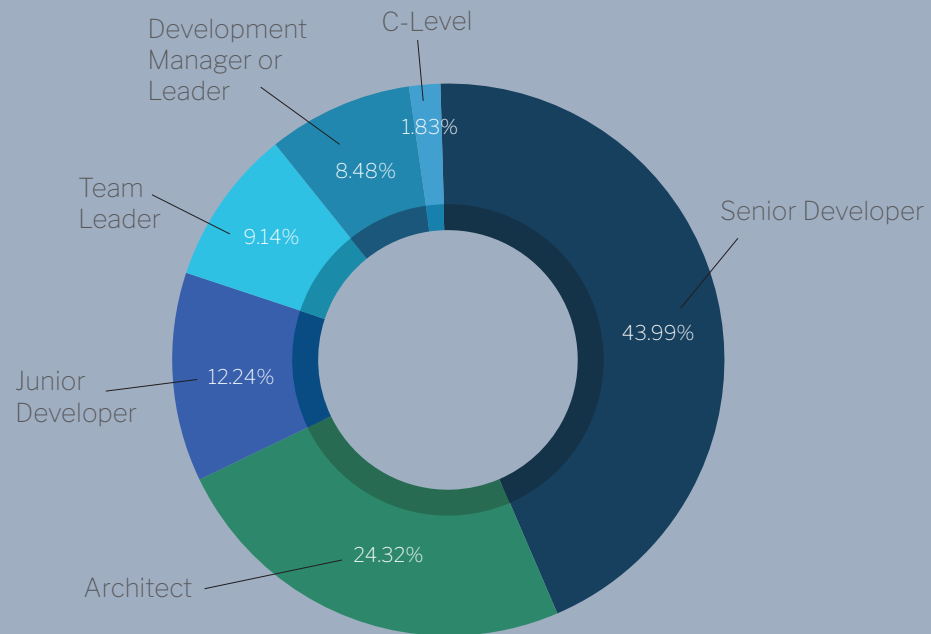
Jakarta EE is the future of cloud-native Java. Jakarta EE’s mission is more frequent releases, lowered barriers to participation, and putting the community back into the platform. It will allow and encourage everyone to participate in an open process that more accurately reflects the needs of the wider community. As it evolves we expect Java innovation from open source communities like Eclipse MicroProfile, that extend the Jakarta EE platform, to be quickly adopted into new versions of the platform to help developers create portable cloud native applications. Jakarta EE represents the best way to drive cloud-native, mission critical applications and build upon the decades of experience of real world deployments and developers.

Community feedback will be a vital ongoing instrument for the evolution of Jakarta EE. This survey is a useful benchmark on how Jakarta EE is being used today, and where the community would like to see its capabilities evolve tomorrow.

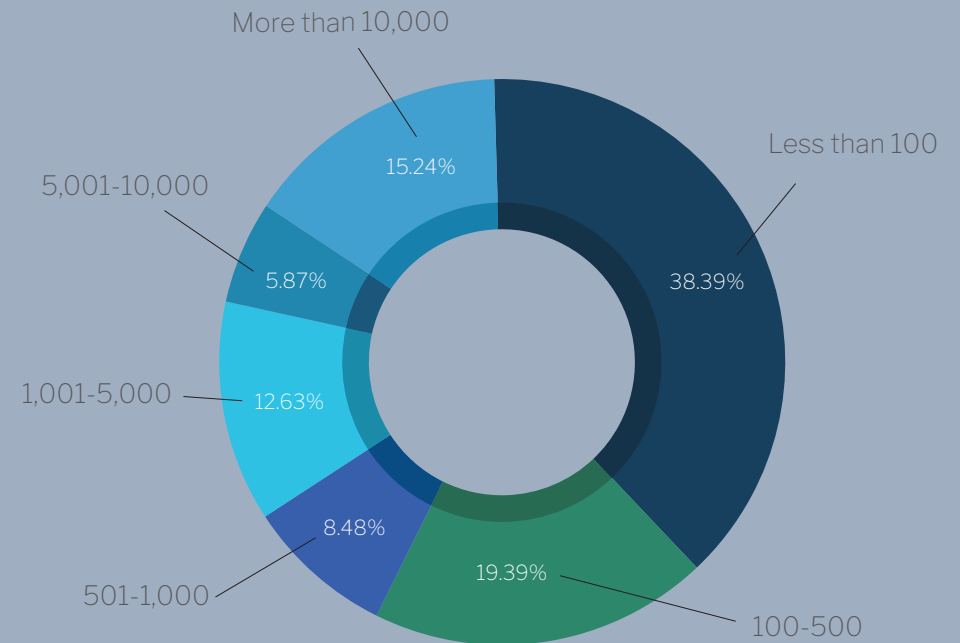
A handwritten signature in black ink, appearing to read 'Mike Milinkovich'.

ABOUT THE RESPONDENTS

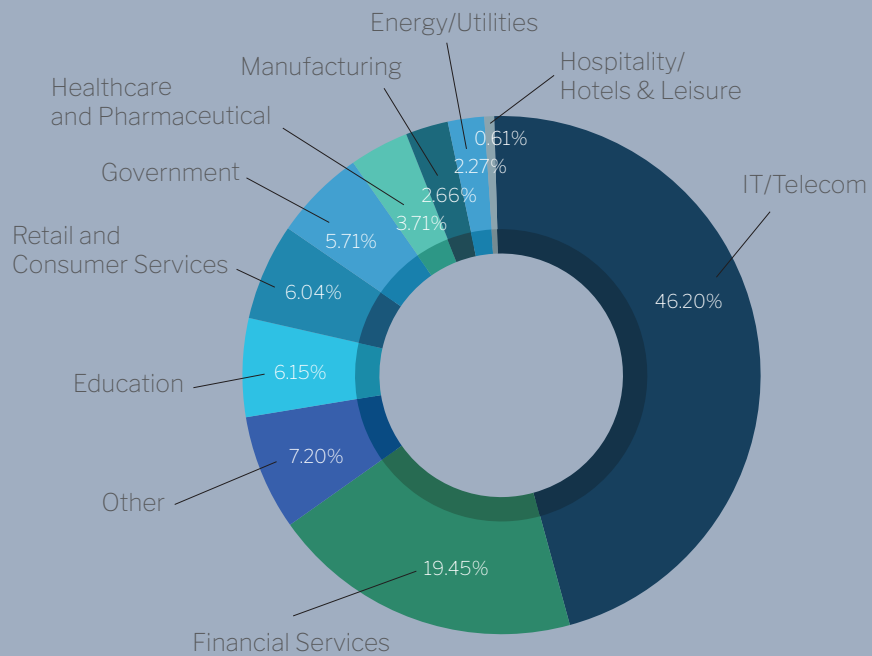
WHICH BEST DESCRIBES
YOUR ROLE?



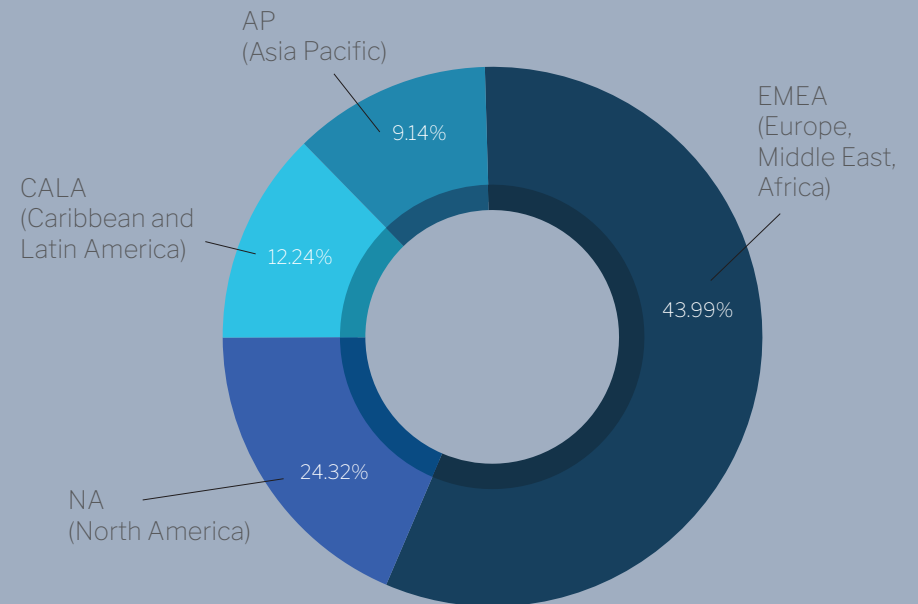
HOW MANY EMPLOYEES WORK
IN YOUR ORGANIZATION?



WHAT INDUSTRY DO YOU WORK IN?



WHAT REGION ARE YOU PERSONALLY LOCATED IN?



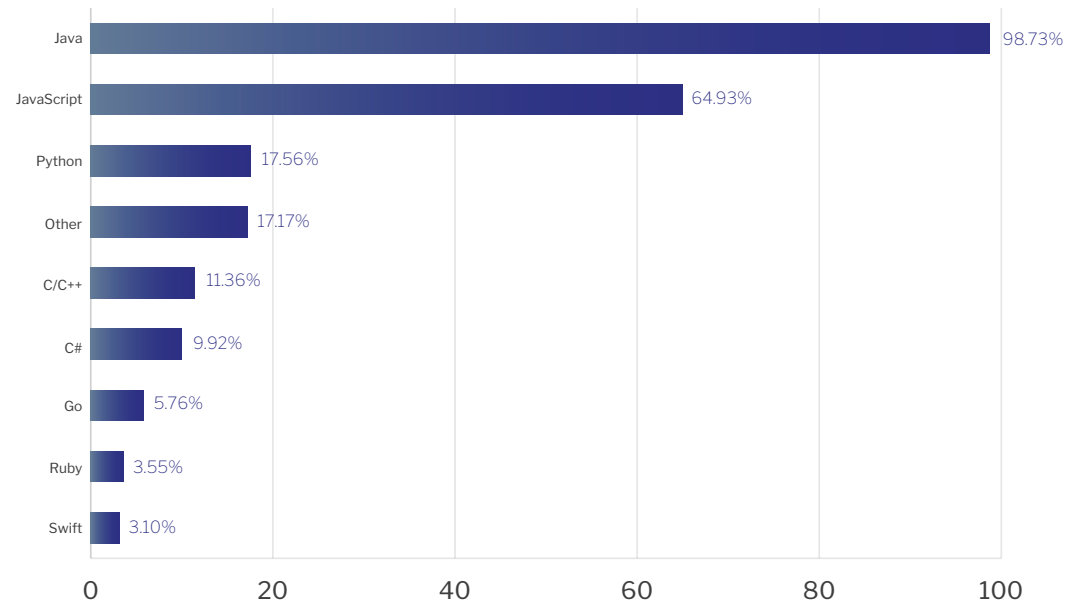


01

JAVA EE IS UBIQUITOUS IN PRODUCTION

Java is consistently either first or second in every reputable programming language ranking on an annual basis. Naturally the participants in this survey had a heavy Java orientation (the survey sought the community's feedback specifically). It is noteworthy that we are in a truly polyglot world—most enterprises today have developers working in multiple languages. But Java is still the dominant language in back-end enterprise systems.

WHICH PROGRAMMING LANGUAGES DO YOU USE AT WORK? SELECT ALL THAT APPLY.

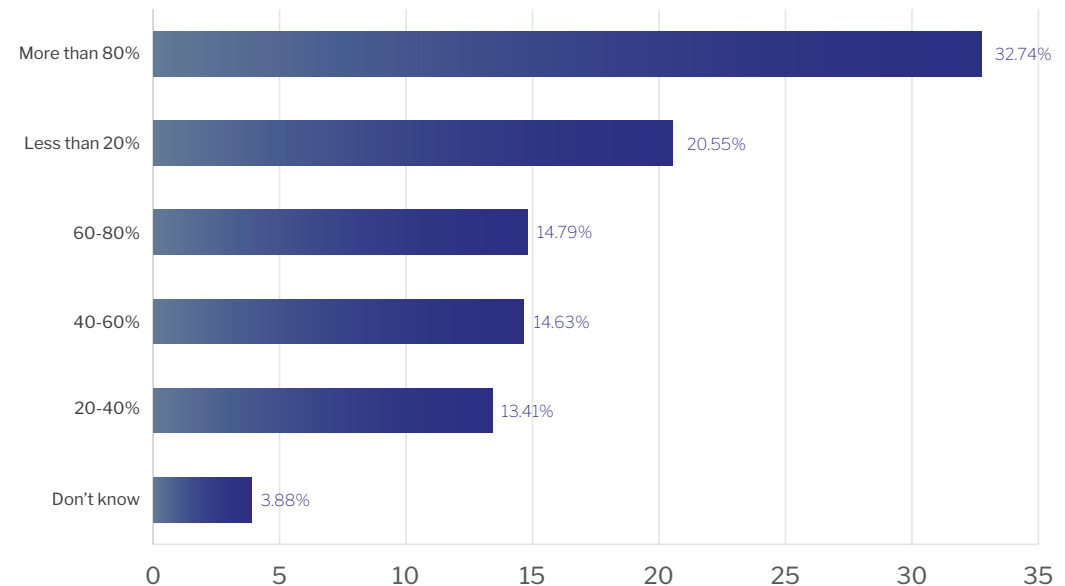


01 JAVA EE IS UBIQUITOUS IN PRODUCTION

There is no doubt that most applications today are being constructed by teams of developers employing multiple languages. IT organizations have become especially adept at mixing and matching modules written in different languages.

But most enterprise applications are usually built primarily using frameworks based on languages such as Java that have stood the test of time.

WHAT PERCENTAGE OF YOUR APPLICATIONS USE MULTIPLE LANGUAGES WITHIN THEIR IMPLEMENTATION (E.G. JAVA AND JAVASCRIPT)?

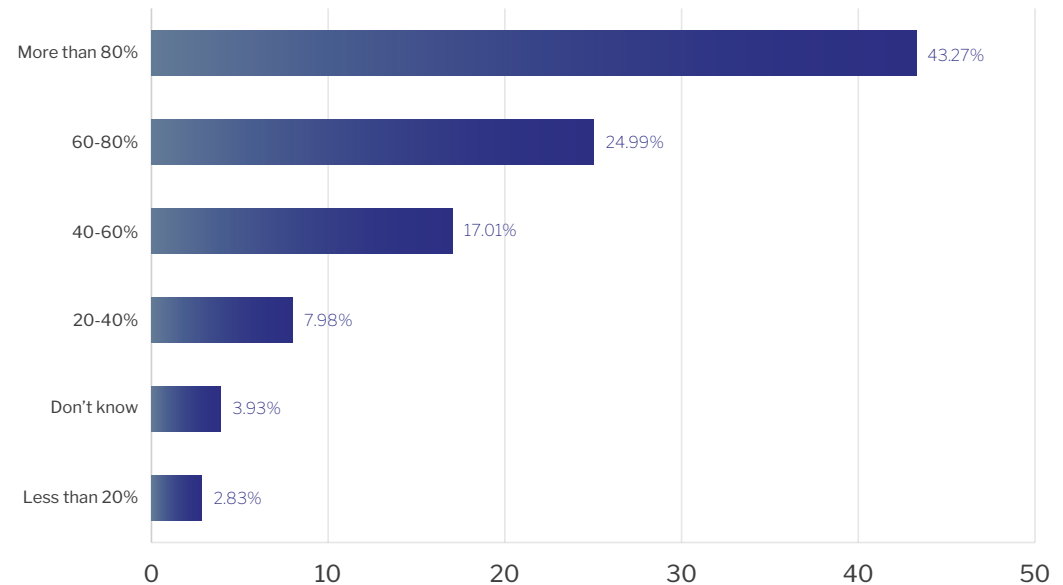


01 JAVA EE IS UBIQUITOUS IN PRODUCTION

When it comes to deploying applications in a production environment Java clearly dominates. Over two-fifths of respondents report 80 percent of their applications running in production are based on Java. Another 25 percent say 60 to 80 percent of their production applications are written in Java.

Given that level of dependence it's easy to see why most organizations are anxious to extend hard-won existing Java expertise.

WHAT PERCENTAGE OF YOUR PRODUCTION SYSTEMS WOULD YOU ESTIMATE YOUR ORGANIZATION BUILT USING JAVA?

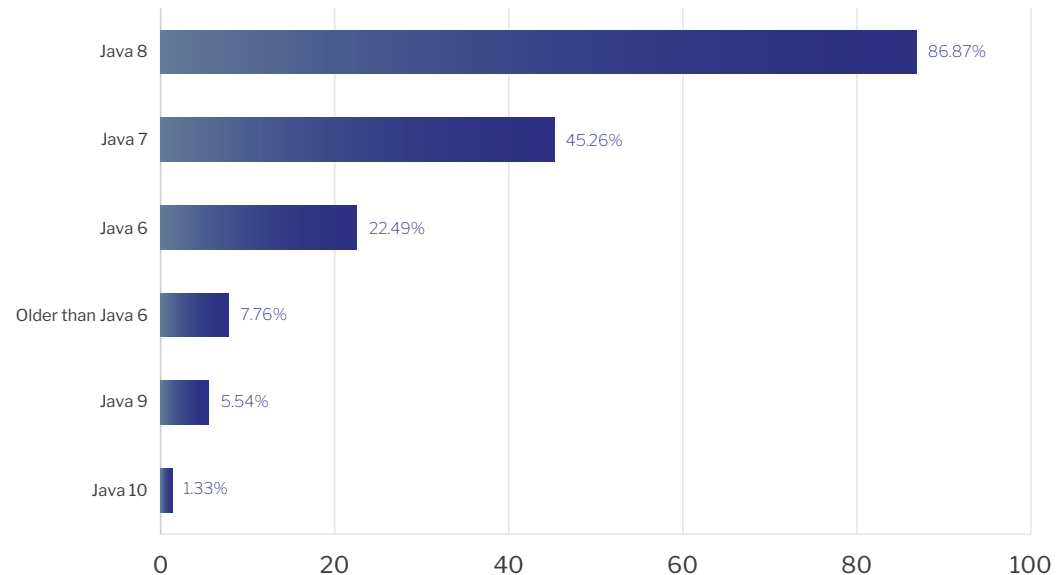


01 JAVA EE IS UBIQUITOUS IN PRODUCTION

Despite the availability of Java 9 and 10, a full 87 percent of survey respondents are running Java 8, with another 45 percent still running Java 7. Enterprise IT organizations are naturally cautious when transitioning platforms.

But a lack of migration progress also suggests there's room for improvement in terms of how future innovations manifest themselves. The Jakarta EE community is committed to not only accelerating the rate at which those innovations get delivered, but also making them simpler to consume.

WHAT VERSION OF JAVA SE IS YOUR ORGANIZATION RUNNING IN PRODUCTION? SELECT ALL THAT APPLY.

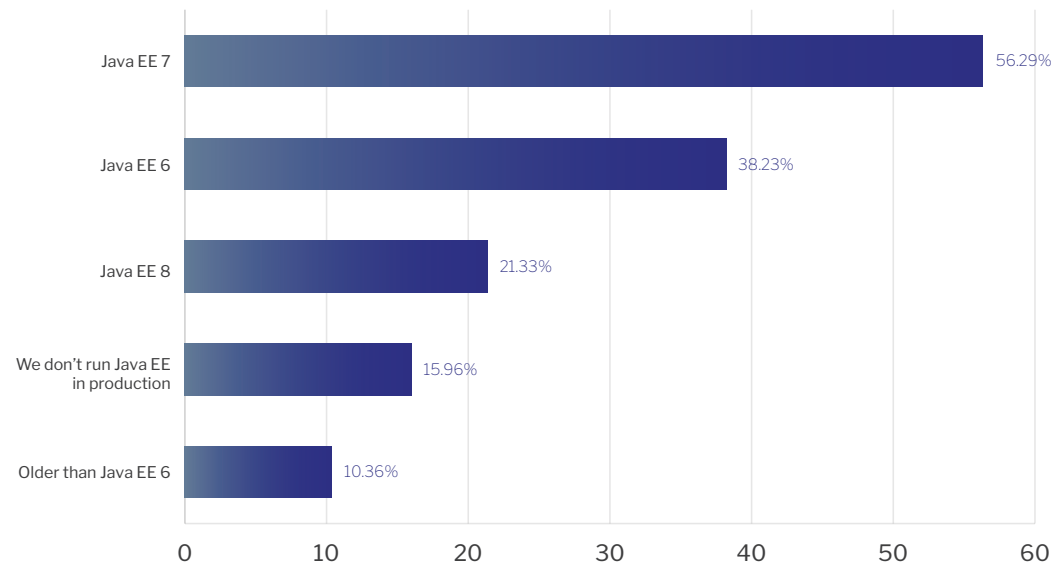


01 JAVA EE IS UBIQUITOUS IN PRODUCTION

Caution among enterprise IT organizations manifests itself when it comes to Java EE as well. Over half of respondents are still running Java EE 7, while 38 percent are still running Java EE 6. That compares to 21 percent running Java EE 8.

There is clearly a significant opportunity for many organizations to leap from an older version of Java EE to Jakarta EE, as it delivers on its promise of providing a modern cloud native Java platform.

WHAT VERSION OF JAVA EE IS YOUR ORGANIZATION RUNNING IN PRODUCTION? SELECT ALL THAT APPLY.

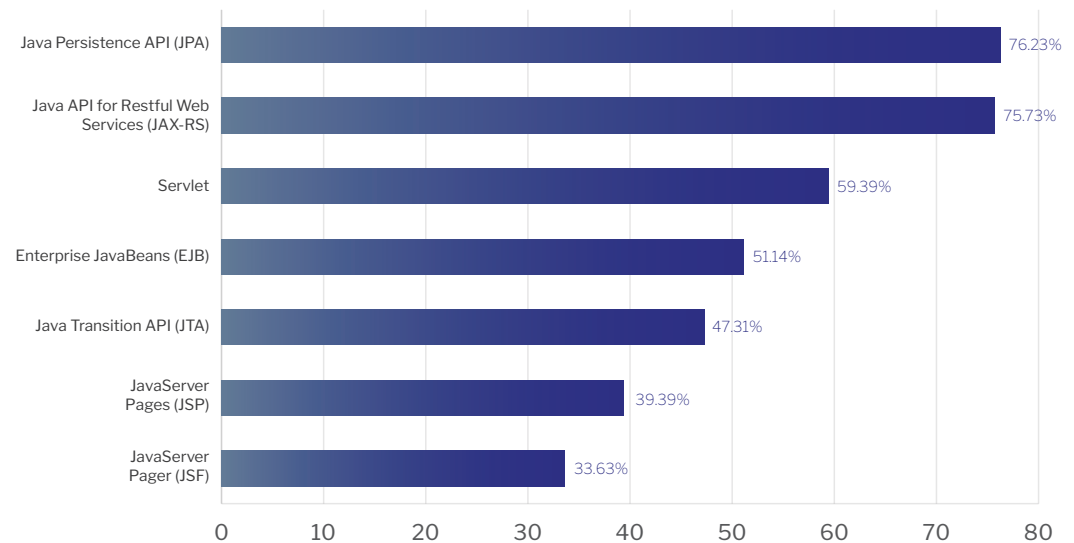


01 JAVA EE IS UBIQUITOUS IN PRODUCTION

There's no shortage of diversity when it comes to Java EE specifications. The most widely used are Java Persistence API (76%), Java API for RESTful Web Services (75%), Servlet (59%), and Enterprise JavaBeans (51%).

Clearly, the Java community puts a premium on open interfaces that provide them with architectural choices for their applications.

WHAT JAVA EE SPECIFICATIONS DO YOU USE IN YOUR APPLICATIONS? SELECT ALL THAT APPLY.



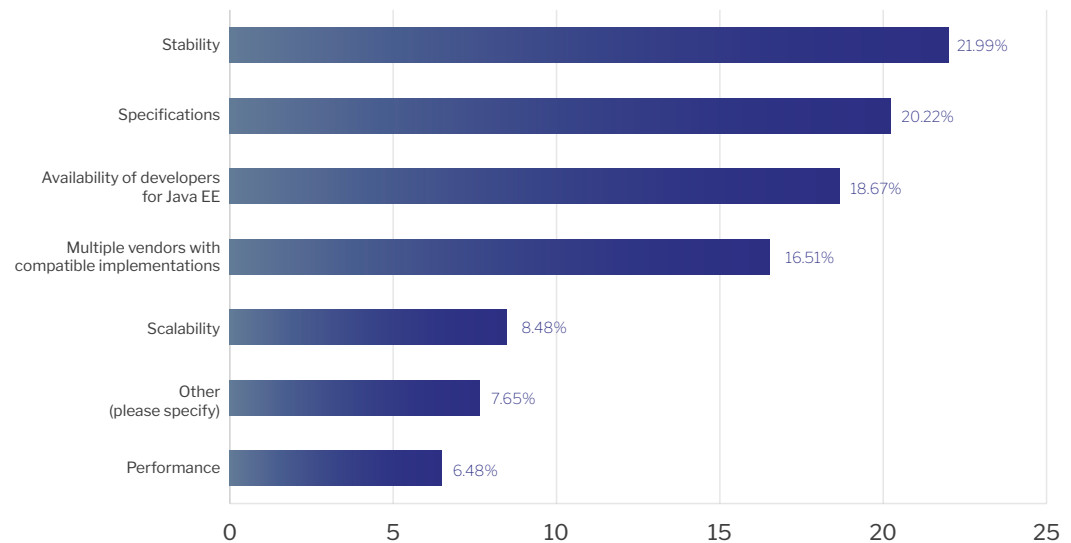
02

OPPORTUNITIES TO BUILD ON JAVA EE'S SUCCESSES

Reasons abound for why so many enterprise IT organizations rely on Java. When asked to cite the most important reasons that their organization selected Java EE the top responses were stability (22%), specifications (20%), availability of developers (18%) and multiple vendors with compatible implementations (16%).

The truth, of course, is much closer to “all of the above.” IT organizations have myriad requirements that Java uniquely meets.

WHAT ASPECT OF JAVA EE HAS MOST MADE IT THE PLATFORM OF CHOICE FOR YOUR JAVA APPLICATIONS?

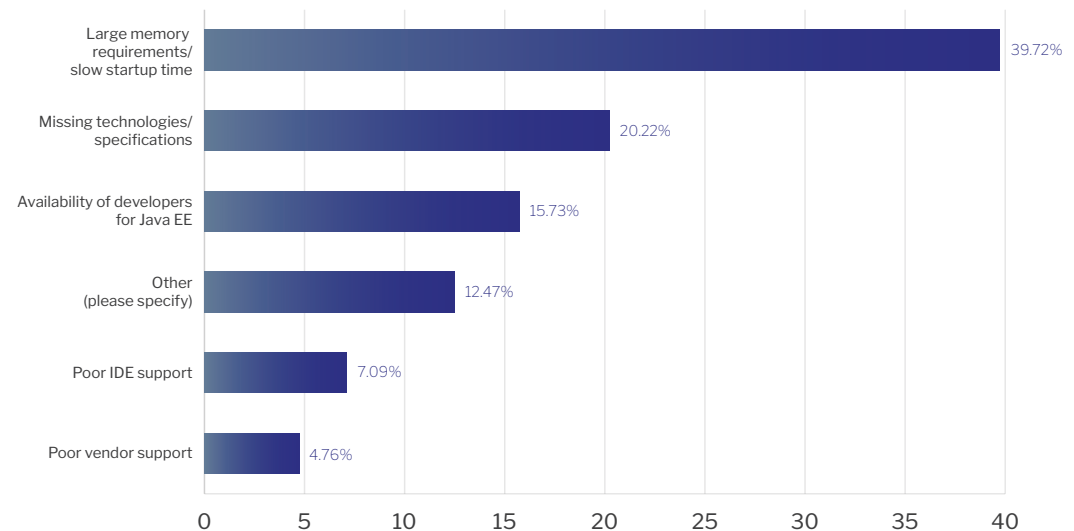


02 OPPORTUNITIES TO BUILD ON JAVA EE'S SUCCESSES

The most widely acknowledged issue when employing Java EE is large memory requirements (40%), then slow startup times (40%), followed by missing technologies and specifications (20%).

It's more than probable that a Jakarta EE community under a code meritocracy and open source governance model is capable of tapping into the combined expertise of the community, and vendors such as IBM, Oracle, Red Hat will make addressing these issues a high priority.

WHAT IS THE MOST CHALLENGING ASPECT OF WORKING WITH JAVA EE?



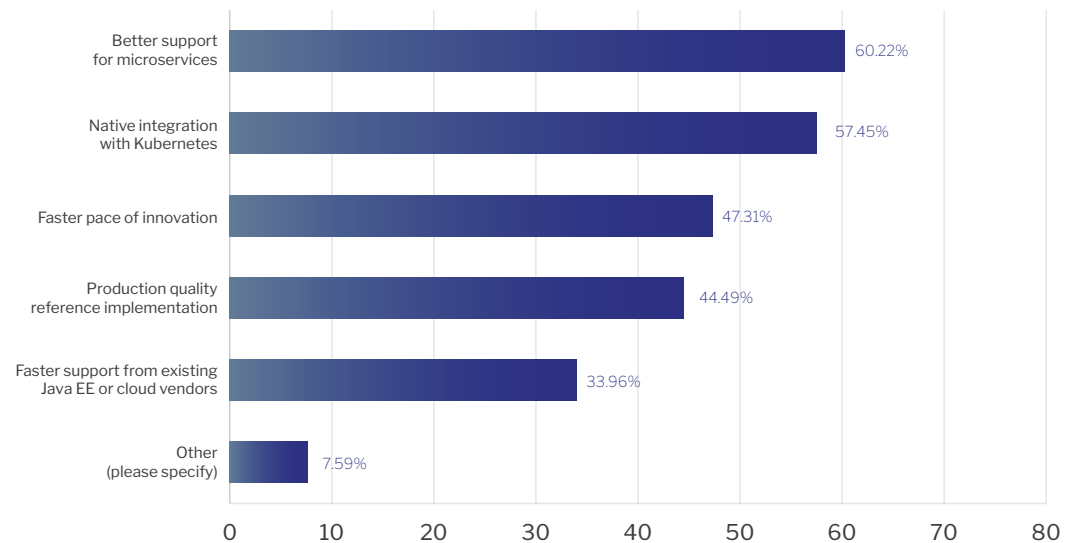
02 OPPORTUNITIES TO BUILD ON JAVA EE'S SUCCESSSES

Not surprisingly, better support for microservices is at the top of the community agenda with 61% of developers citing it as their number one wish for the Jakarta EE platform.

Other desires include integration with Kubernetes and faster rates of innovation for similar types of advances.

The Jakarta EE community leaders have already made it clear how much feedback from the community will drive their agenda, so stay tuned for further developments!

HOW CAN THE ECLIPSE FOUNDATION BEST EVOLVE JAKARTA EE TO MEET YOUR CLOUD NEEDS? SELECT ALL THAT APPLY.



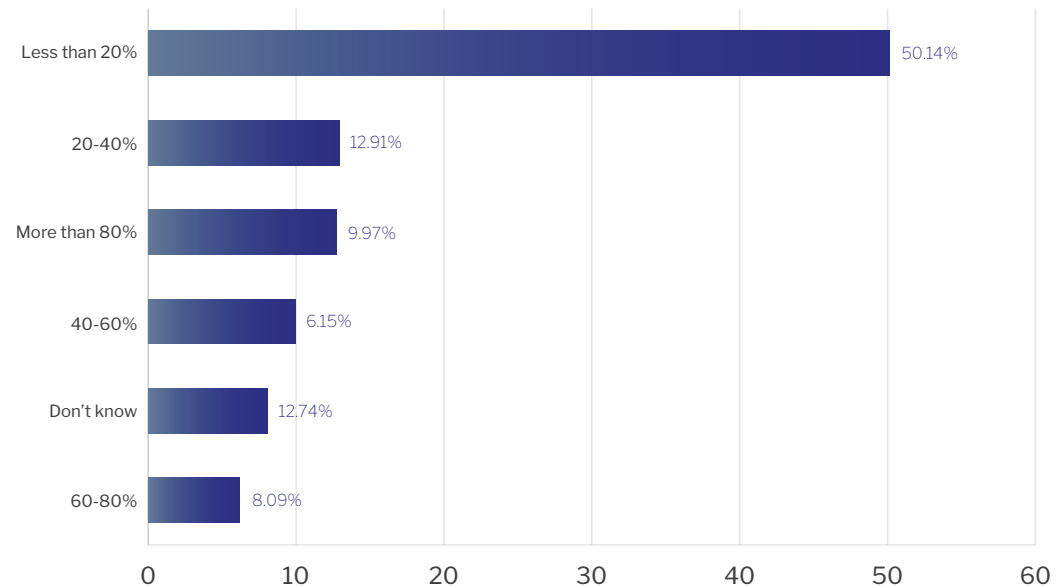
03

THE CLOUD NATIVE JAVA IMPERATIVE FOR JAKARTA EE

IT organizations are moving mission-critical applications based on Java into the cloud at a very deliberate pace. In many cases, the economics of cloud computing simply don't match the requirements of Java applications that continually run. In other instances, compliance requirements necessitate those applications run on-premises.

About half of survey respondents report that less than 20 percent of their applications are running in the cloud. Obviously, migrating Java applications to the cloud is still very much a work in progress.

WHAT PERCENTAGE OF YOUR JAVA SYSTEMS ARE CURRENTLY DEPLOYED IN THE CLOUD?

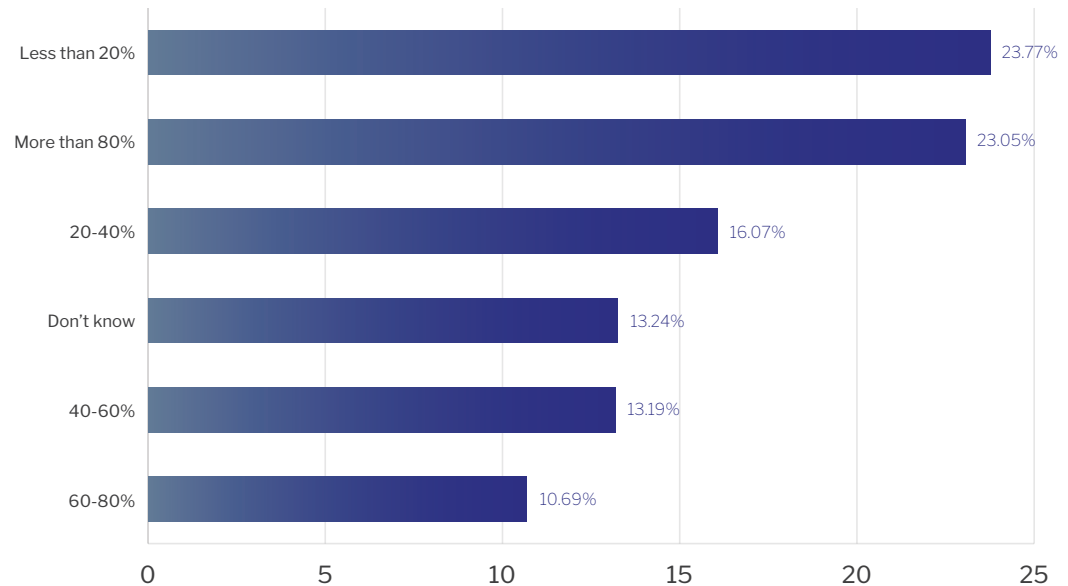


03 THE CLOUD NATIVE JAVA IMPERATIVE FOR JAKARTA EE

Two years from now the number of Java applications running in the cloud is projected to substantially increase. The survey finds almost a quarter of respondents expect to have more than 80 percent of their applications running in the cloud.

Those estimates may be low given that most of the respondents have not had a chance to evaluate a forthcoming cloud-native implementation of Jakarta EE that will be backwards compatible with Java EE.

WHAT PERCENTAGE OF YOUR JAVA SYSTEMS WILL BE RUNNING IN THE CLOUD IN TWO YEARS?

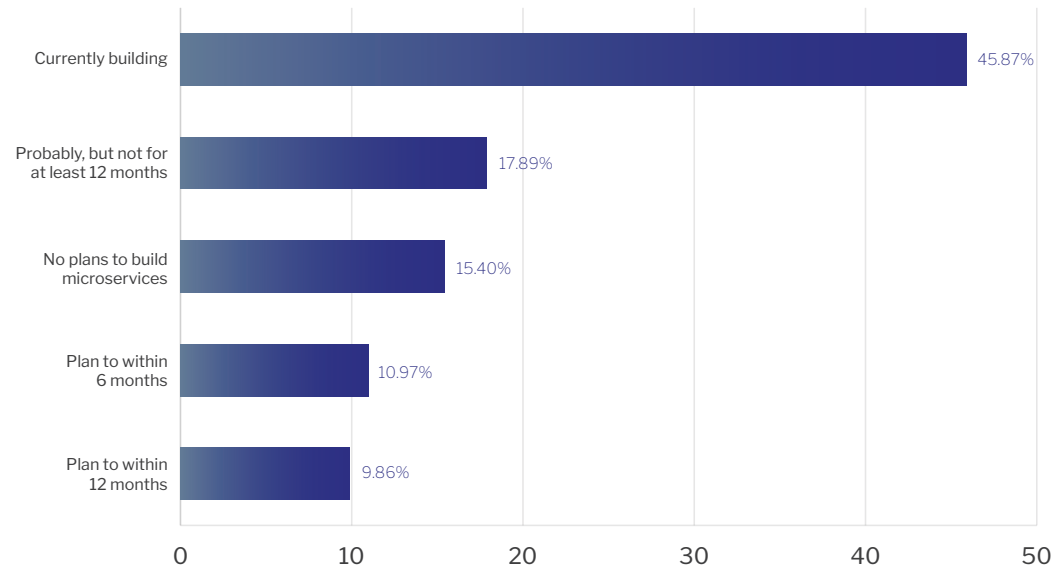


03 THE CLOUD NATIVE JAVA IMPERATIVE FOR JAKARTA EE

That fact that so many organizations (46%) are already building microservices is a testament to how many developers in the community remain at the cutting edge of IT. Another 20 percent say they plan to build microservices in the next year.

Many of those microservices are clearly destined to be deployed on a broad mix of public and private cloud computing environments.

ARE YOU CURRENTLY CLOUD, OR ARE YOU PLANNING ON, BUILDING MICROSERVICES?

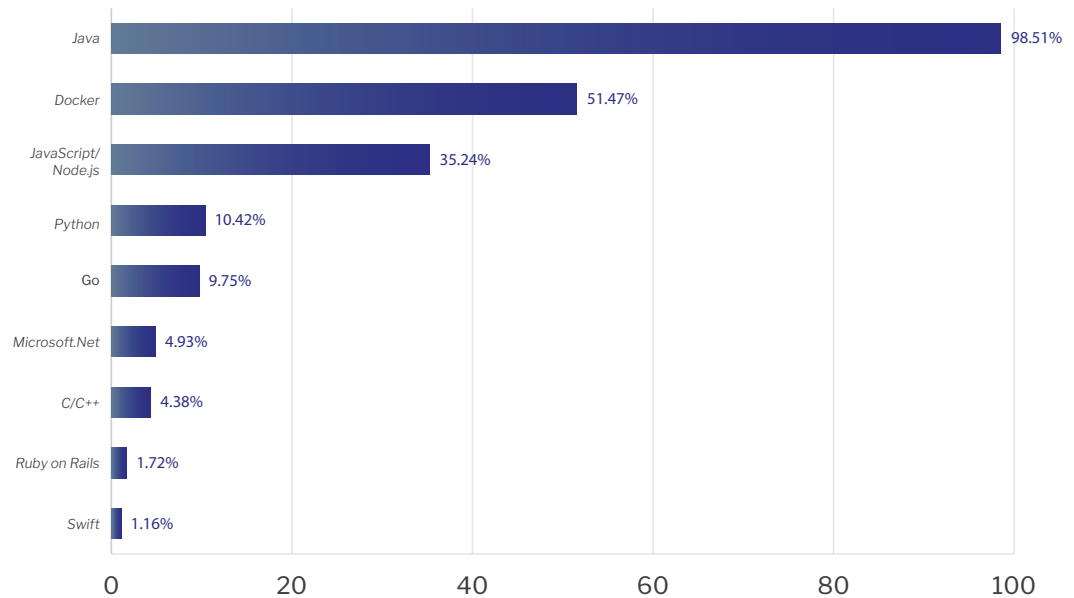


03 THE CLOUD NATIVE JAVA IMPERATIVE FOR JAKARTA EE

Organizations building microservices clearly favor Java (95%), while another 51 percent cite Docker.

The relationship between Java and Docker containers is clearly evolving. But the one thing that is clear is that Java will manifest itself in multiple run-time environments, not the least of which will include Docker containers.

IF YOU ARE BUILDING OR PLANNING ON BUILDING MICROSERVICES, WHAT LANGUAGES OR RUNTIMES DO YOU/WILL YOU USE TO BUILD MICROSERVICES? SELECT ALL THAT APPLY.

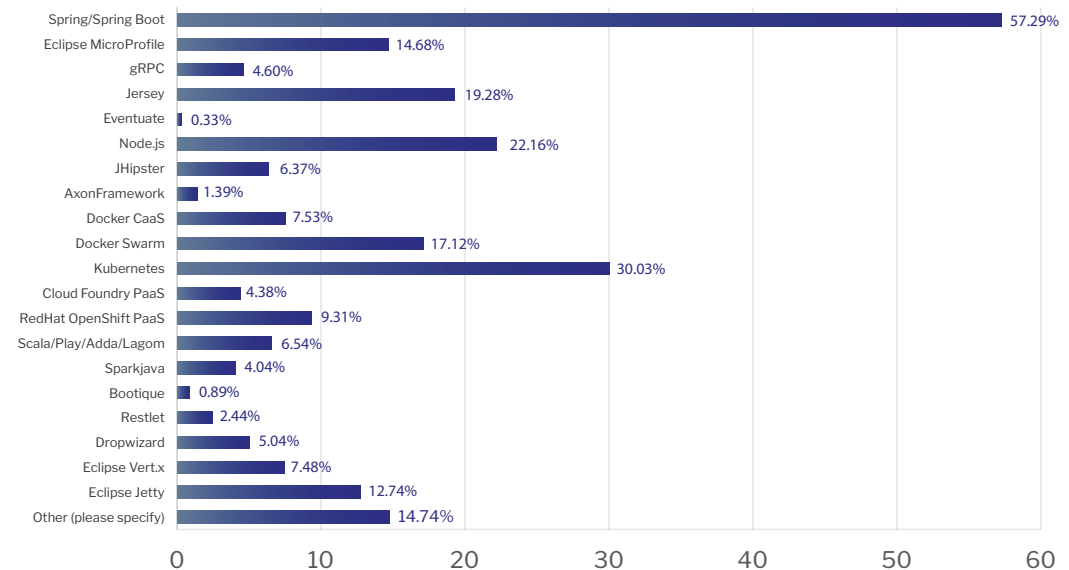


03 THE CLOUD NATIVE JAVA IMPERATIVE FOR JAKARTA EE

The Spring and Spring/Boot frameworks (57%) dominate today when it comes to building microservices using a derivative of Java. But as Jarkarta EE continues to evolve it's also more than apparent that framework such as Kubernetes (30%), Node.js (22%), Jersey (19%) and Eclipse MicroProfile (15%) are gaining traction.

Among those frameworks it's worth noting that the community-led Eclipse MicroProfile is only a year old with much more to come.

IF YOU ARE BUILDING MICROSERVICES, WHAT FRAMEWORKS ARE YOU EMPLOYING? SELECT ALL THAT APPLY.

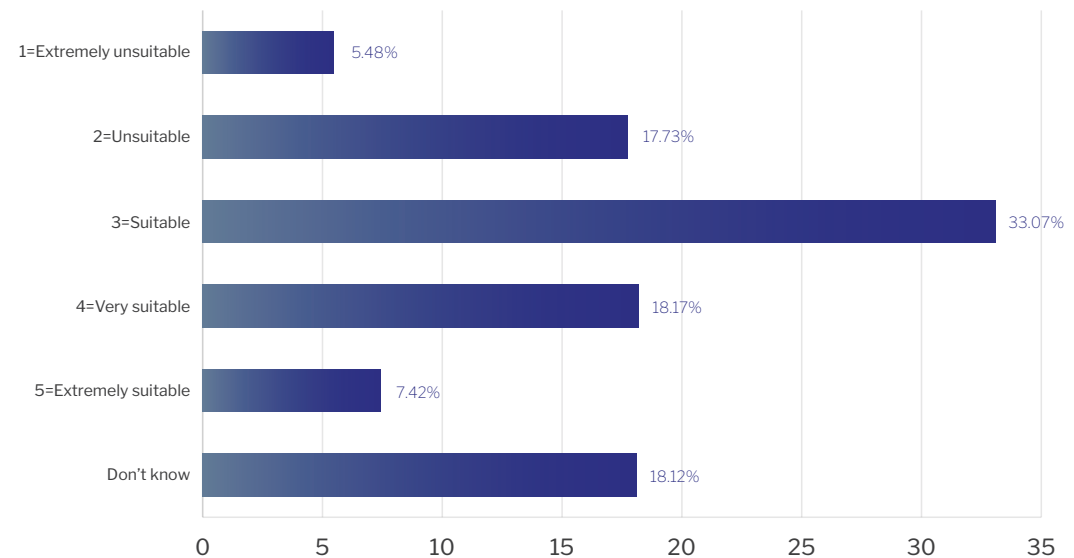


03 THE CLOUD NATIVE JAVA IMPERATIVE FOR JAKARTA EE

Well over half of respondents (58%) view Java EE in its current form as being suitable for the cloud. Of course, the community has not seen anything yet with Jakarta EE.

Jakarta EE as a cloud-native implementation of a platform capable of running modern and legacy applications is about to inject unprecedented levels of agility into the industry's most venerable platform.

HOW WOULD YOU RATE THE HISTORICAL SUITABILITY OF JAVA EE FOR CLOUD ENVIRONMENTS? (1=EXTREMELY UNSUITABLE, 5=EXTREMELY SUITABLE)



#1: JAVA: THE ONCE AND FUTURE KING OF PRODUCTION APPS

WHEN IT COMES TO BUILDING and deploying applications it's clear we live in a polyglot world. But when it comes to production environments a new survey of 1,800 developers conducted by the Eclipse Foundation makes it clear Java still reigns supreme.

Under the auspices of the Eclipse Foundation, the community surrounding Java is now pooling their collective efforts to accelerate the rate at which future innovations are made available to what will be a cloud-native platform dubbed Jakarta EE. Everything that has occurred prior to the formation of Jakarta EE will still be known as Java EE and will remain so to comply with licensing agreements with Oracle. All future Jakarta EE work will be driven by a consortium of vendors working collaboratively with a large community of open source developers.

The Eclipse Foundation survey specifically finds that nearly half of respondents (47%) say they expect that one of the primary benefits of Jakarta EE will be a much faster pace of innovation. Overall, the survey finds that 99 percent of developer respondents build applications using Java at work, compared to 65 percent that also cited JavaScript. Other programming languages being employed included Python (18%), C/C++ (10%), C# (9%). Interestingly, Go (6%), Ruby (4%) and Swift (3%) collectively represent only 13 percent. In fact, in terms of overall programming language usage at work “Other” comes in at 19 percent, ahead of Python.

Developers are increasingly comfortable working with multiple languages. A third of respondents (33%) say they are using multiple languages to construct more than 80 percent of their applications. But nearly half (46%) are using Java to construct more than 80 percent of the applications they deploy in a production environment. Another 41 percent says they are using Java to build anywhere from 40 to 80 percent of their production applications.

That dominance is also beginning to be extended as organizations begin to become more comfortable with deploying mission critical Java applications beyond the four walls of the traditional data center. Half of the survey respondents said fewer than 20 percent of their Java applications are running in the cloud today. Over 30 percent said that within the next two years they expect to be running 60 percent or more of their Java applications in the cloud.

The rate is only going to increase as developers become more familiar with Jakarta EE and more adept at building microservices. Nearly half of respondents (45%) said they are using Java to build microservices, with another 21 percent planning to join them in the coming year. The survey also finds that 15 percent of respondents have already adopted Eclipse MicroProfile, which was released in July 2017. Eclipse MicroProfile is an open source platform definition lead by IBM, Red Hat and others that optimizes Enterprise Java for a microservices architectures.

#2: OPPORTUNITIES TO BUILD ON JAVA EE'S SUCCESSES

SOMETIMES IT'S EASY to take for granted a technology that most enterprise IT organizations rely on to run their most critical applications. The survey of developers finds 43 percent of developers said more than 80 percent of the applications running in production were built using Java. Another 25 percent said 60 to 80 percent of their production applications were based on Java.

Over 80 percent of those same developers also identified Java 8 as being the version of Java SE employed; followed by 45 percent that still had some form of Java 7 running. Less than 10 percent are running either Java 9 or Java 10. But when it comes to Java EE, there's more reliance on older versions. Just over half (55%) are running Java EE 7, followed by Java EE 7 at 39 percent and Java EE 8 at 21 percent. Clearly, there's plenty of room for improvement when it comes to enticing organizations to embrace the latest and greatest Java innovations. But the survey also finds the primary reasons organizations continue to rely on Java EE are stability (22%), specifications (21%) and, not to be overlooked, the actual availability of Java developers (19%).

That success comes despite there being multiple implementations by different vendors over the years; an issue that Jakarta EE creates an opportunity to overcome once and for all.

#3: THE CLOUD NATIVE JAVA IMPERATIVE FOR JAKARTA EE

NAMING CONVENTIONS ASIDE, Jakarta EE is about defining the future of Java in the enterprise—especially when it comes to building cloud-native applications. As has been shown multiple times over, a consortium of vendors can drive innovation using a commercial-friendly open source model at a much faster and sustainable rate than any single vendor or standards body. Just as importantly, because of all the peer review resources that can be brought to bear on the code that emerges from an open source consortium, the software output is more battle-tested, reliable, secure and—not to be overlooked—backwards compatible. To this end, the Eclipse Foundation, working with its members, has established the Jakarta EE Working Group.

The Eclipse Foundation will ensure that Jakarta EE is based on a self-governing meritocracy that sets all technical agendas and plans. The Project Management Committee (PMC) for an Eclipse top-level project is responsible for among other things, reviewing and approving plans, accepting new projects, approving releases, and managing committer elections. The PMC approves new projects but does not direct the activities of the projects managed under its auspices. Each project manages their technical direction, but the PMC may elect to coordinate the activities of multiple projects to facilitate development of a specific innovation. The Jakarta EE Working Group may sponsor complementary activities

to generate and promote these plans, but ultimately it is up to the projects to agree to implement these plans or roadmaps. The founding members of the Jakarta EE working group are Fujitsu, IBM, Oracle, Lightbend, Payara Systems, Pivotal, Red Hat, Tomitribe and Webtide.

Overall, this community based and member-driven approach is a far cry from having to wait for a single vendor to determine whether your proposed contribution meets with their corporate goals for a platform.

SUMMARY

The arrival of Jakarta EE represents a renaissance for the Java community. While most developers are proficient in multiple languages, survey respondents made it clear that Java EE as it is evolving into Jakarta EE is still the platform they rely on most to build true enterprise class applications. Developers might use other platforms from time to time. But the cost of learning another language as well as they know Java EE and, by extension, now Jakarta EE, is simply too high at a time when developers are under more pressure than ever to build and deploy applications faster than ever.

The community has spoken. To fulfill its promise, Jakarta EE must deliver rapid innovation, open collaboration and broad community participation. The success of Jakarta EE will be member-driven and based on a governance model that reflects the evolving needs of all stakeholders. In the years to come, expect open source Java innovations like Eclipse MicroProfile to be adopted into the Jakarta EE stack to help developers rapidly create microservices and cloud native applications. Jakarta EE will rely on the passion, experience and engagement of the community to deliver on the future of cloud native Java.

