



Fulfilling the Vision for Open Source, Cloud Native Java

Leveraging the strength
of a united community to
advance enterprise Java

As Java industry leaders, developers,
and enterprises join forces to evolve
Jakarta EE for enterprise workloads in
the cloud, we explore what cloud native
Java really means, why it matters so
much to so many people, and where it's
headed.

Contents

03	Executive summary
04	Charting a course for cloud native Java
07	Java is more important than ever
09	The time is right for fully open source enterprise Java
12	Developers can focus on resolving customers' business challenges
17	Software vendors have new opportunities to build their brand
19	Enterprises must evolve key strategic assets — their Java EE applications
20	The road ahead — part 1
23	The road ahead — part 2
24	Engage in cloud native Java
26	Become an Eclipse Foundation member today
28	Join us today

▶ Executive summary



More than two decades after it was developed, Java™ is still the most popular programming language in the world¹. According to industry analyst firm IDC, 90 percent of Fortune 500 companies rely on Java applications². Features such as stability, portability, and scalability have played a big part in securing Java's dominant position. Java EE has been the de facto platform for enterprise workloads for years, providing reliable, multivendor standards for the world's mission-critical applications and datacenters across all industry sectors.

Today, Java EE is finding new relevance as Jakarta™ EE at the Eclipse Foundation. A diverse and strongly committed community of software vendors, developers, and enterprises have united

in the Jakarta EE Working Group to deliver an open source enterprise Java platform under a transparent, vendor-neutral process. As the cloud native paradigm becomes an increasingly important pillar for IT modernization and business transformation initiatives, a vision for cloud native Java is emerging to enable enterprises' next phase of evolution. Support for containers, microservices, and multi-cloud portability top the priority list.

It's time for everyone who has a stake in enterprise Java to get involved in the Jakarta EE community. There are huge opportunities to influence and contribute to the first truly open source, cloud native Java specification for enterprises and to benefit from processes that serve everyone.

1. <https://www.tiobe.com/tiobe-index/>

2. IDC

▶ Charting a course for cloud native Java

With the contribution of Java EE technologies to the Eclipse Foundation and the creation of the Jakarta EE Working Group, leading Java software vendors, developers, and enterprises have united in an unprecedented way. And Jakarta EE community members have an unprecedented opportunity.

The Jakarta EE community will chart a new course for a cloud native Java future. This well-governed and vendor-neutral open source ecosystem will

shape the way enterprise Java evolves in its cloud native form.

A community committed to success

Leading software vendors, developers, and enterprises are strongly committed to the long-term success of cloud native Java and the Jakarta EE community.



Mark Little
VP Engineering, Red Hat

“The Jakarta EE community is the best way to have a true, vendor-neutral meritocracy based on collaborative efforts in a vendor-neutral foundation for defining what Enterprise Java means in the cloud for the next decade.”



Reza Rahman
Principal Program Manager for Java on Azure, Microsoft

“It’s very important that everybody focuses on making Jakarta EE successful for the long-term benefit of the Java ecosystem.”



Josh Juneau
Java Developer

“The Jakarta EE community is very important for the evolution of the platform. With more eyes and ears on the platform giving feedback and providing input, the more solid and agile the platform will be moving forward.”



Sebastian Daschner
Lead Java Developer Advocate, IBM

“There’s a saying about agreeing on standards and competing on implementations. When we have multiple vendors trying to improve their implementations, it helps everybody.”

A new baseline for evolution

Jakarta EE 8 provides a new baseline for evolving enterprise Java technologies to support mission-critical applications and workloads in the cloud. And the enterprise developer community is strongly voicing its thoughts on requirements and priorities for evolution.

In March 2019, nearly 1,800 people around the world participated in the 2019 Jakarta EE Developer Survey. Their responses were clear and consistent.

The enterprise marketplace is looking for Jakarta EE to support containers, microservices, and multi-cloud portability.

Developers are looking for Jakarta EE to support the open specifications and tools needed to create dynamic and scalable cloud native applications.

Welcome to the future of cloud native Java

With the delivery of the Jakarta EE 8 Platform, all of the pieces are in place for software vendors, developers, and enterprises to strongly engage in shaping the future of cloud native Java through the Jakarta EE community.

Jakarta EE 8:

- > Is fully compatible with Java EE 8 specifications
- > Follows the Jakarta EE Specification Process to ensure fully transparent specification development
- > Includes the same APIs and Javadoc as Java EE 8
- > Provides Jakarta EE 8 technology compatibility kits (TCKs) under open source licenses that are based on, and fully compatible with, the Java EE 8 TCKs
- > Includes a Jakarta EE 8 platform specification that describes the same platform integration requirements as the Java EE 8 platform specification
- > Provides a compatibility and branding process for demonstrating that implementations are compatible with Jakarta EE 8

We go forward from here

Continuing to evolve Jakarta EE is the best way to ensure that Java developers and software vendors can meet the modern enterprise's need for cloud-based applications that resolve key business challenges.



I work with several startups in industries ranging from automation to power grids and aviation. They use Jakarta EE for business reasons, mainly the productivity it enables, and they use clouds because they don't want to have to administer the technologies."

Adam Bien, Freelance Software Architect and Developer

▶ Java is more important than ever

Java has stood the test of time. Today, it still has a massive footprint, and it remains the dominant programming language for developing enterprise applications. In the TIOBE Index, an indicator of the popularity of programming languages, Java has never polled lower than second place³.

Our 2019 survey revealed that, while most development teams use multiple programming languages, 95% use Java.

Java's ongoing popularity and heavy usage is not surprising. Java provides the fundamental elements needed in all enterprise applications — portability, stability, and security.

These fundamentals make Java

important for almost every business. Our survey highlighted four main reasons organizations develop applications in Java:

- > Stability: 22%
- > Specifications: 21%
- > Developer availability: 19%
- > Multiple vendors with compatible implementations: 16%

Our survey also confirmed that Java is the dominant language used for applications deployed in production environments. So, it's no surprise that most companies are intent on protecting their strategic investments in Java, including retaining internal Java developer expertise.

“

*Java is not necessarily the language of the startup. But, they use Java when they grow up. When they realize they need **stability, security, and a broad-based ecosystem**, startups move from whatever they're using to Java.”*

Reza Rahman, Principal Program Manager for Java on Azure, Microsoft

3. <https://www.tiobe.com/tiobe-index/>

Cloud native Java is the next logical step

Cloud native environments are a natural evolution for building modern enterprise systems.

Software vendors, developers, and enterprises need the ability to move massive numbers of Java applications to the cloud without recoding them or having to learn another language. They also need the ability to develop cloud native applications from the ground up.

Plans for building cloud native architectures are critically important. More than 60% of our survey respondents have plans to build cloud native architectures in 2019 or 2020:

- > 32% plan to build cloud native architectures within six months
- > 16% are currently building cloud native architectures
- > 14% plan to build cloud native architectures within 12 months



*When you're looking to move your existing applications to the cloud, you don't want to have to recode them, especially if they've been working really well for a number of years. And if you're writing applications from scratch, **you're probably going to find that most of your developers have Java skills.** They may not want to learn another language, and you may not have the time or luxury of learning another language on the job."*

Mark Little, VP Engineering, Red Hat

► The time is right for fully open source enterprise Java

Jakarta EE is the only server-side, vendor-neutral open standard available. All other server-side development platforms are vendor-specific, even if they are open source. Of particular note, specification development follows the Jakarta EE Specification Process and Eclipse Development Process, which are open, community-driven successors to the Java Community Process (JCP) for Java EE. This makes for a fully open, collaborative approach to generating specifications, with every decision made collectively by the community. Combined with open source TCKs, and an open process of self-certification, Jakarta EE represents a significant shift from the past.

With a true, vendor-neutral approach to server-side application development, software vendors, developers, and enterprises can choose the compatible implementations that make the most sense for their needs. They have a strong voice and opportunities to influence how those implementations evolve.

A common vision for cloud native Java has emerged

Enabling technologies have pushed the idea of the cloud and cloud native development into the forefront of what developers are thinking about today.

Topics related to cloud native development, microservice architectures, containers, orchestration, and service meshes are on the agenda at major conferences around the world.

While the specific definition of cloud native Java varies among community members, a common vision has emerged for industry standards and an open source stack for building, deploying, and managing containerized Jakarta EE-based microservices on Kubernetes. Java developers can build on Jakarta EE with several best-of-breed enabling technologies that have become available in the last few years.



Key enabling technologies include:

Eclipse MicroProfile®

An Eclipse Foundation project that creates new features and capabilities to address microservice architectures for Jakarta EE and non-Jakarta EE technologies. Developers can seamlessly mix and match Jakarta EE and MicroProfile APIs in the same application. This approach allows enterprises that built applications during the pre-cloud, pre-container era to take advantage of more efficient microservices.

Docker

Containers that give developers the agility to easily create different application environments, test different scenarios, and port applications from one cloud or environment to another.

Kubernetes

An open source container orchestration system for automating application deployment, scaling, and management. Kubernetes has become the de facto standard for Linux container orchestration.

Istio

An open source service mesh for connecting, monitoring, and securing microservices. With Istio, developers can create a network of deployed services with load balancing, service-to-service authentication, monitoring, and more, without requiring any changes in service code.



As the development world unites around cloud native Java, the number of Java apps running in the cloud is projected to increase significantly. By 2021, 32% of our 2019 survey respondents expect to be running more than 60% of their Java applications in the cloud.

"With the rise of Linux containers and Kubernetes, the definition of cloud native today is different to what it would have been only a few years ago. Today, you can develop and deploy applications in a way that is portable across all cloud vendors that support Linux containers, which is pretty much all of them. And Kubernetes takes care of the fault tolerance and detecting failures for new levels of reliability and availability."

Reza Rahman, Principal Program Manager for Java on Azure, Microsoft

"It's basically enterprise applications that are implemented using Java and built with a cloud native technology stack using containers, container orchestration, service meshes, and everything we have within the modern ecosystem of environments and tools"

Sebastian Daschner, Lead Java Developer Advocate, IBM

▶ Developers can focus on resolving customers' business challenges

For developers, cloud native Java brings new opportunities to move away from the challenges of working with massive, monolithic infrastructures, continuously trying to resolve nuts-and-bolts-level issues, and learning new frameworks.

Overall, developers have more time to focus on business logic and other areas of strategic importance to their customers. Their value to customers escalates from resolving programming challenges to resolving key business challenges.



"If you would like to solve problems right now, I think Jakarta EE and MicroProfile are the perfect fixes because all of the hard decisions are already made, and you can be very productive right away. It's right out of the box with no special requirements. We've used cloud native Java for larger projects and for startups, and there were no real problems."

**— Adam Bien, Freelance
Software Architect and
Developer**

A virtuous circle of collaborative development

The ability to quickly and easily leverage technologies that already exist opens the door for developers to experiment with improvements to existing implementations and push the technology forward for their own benefit and the benefit of the entire ecosystem. They have new opportunities to influence and contribute to an even stronger code base to better support enterprise applications and the next set of cloud native Java improvements.

This collaborative approach to problem-solving and technology evolution means cloud native Java is continually advancing. And it evolves in a way that reflects a broad set of interests, knowledge, and capabilities, rather than a narrow set of industry- or application-specific interests and skill sets. This breadth makes it far more likely that the technology will meet the needs of developers, their customers, and the overall ecosystem.

Along with the ability to get involved, influence technology direction, and make change happen, developers also have access to a community of highly knowledgeable and skilled collaborators who are ready and willing to help resolve problems encountered along the way.

“

It's a risky proposition if you're developing your flagship application and it's tied to a particular, vendor-specific stack. If you're working with a multivendor technology and it's an open standard, it's less risky. Because these technologies are developed by consensus, you're not tied to the decisions of a single company.”

**Reza Rahman, Principal Program Manager
for Java on Azure, Microsoft**



Developer benefits transfer to enterprise customers

Enterprises are also looking to the cloud to reduce, and in some cases eliminate, the time and cost burden of administering their own services and learning new technologies. A growing number of enterprises have recognized the efficiencies and opportunities to solve key business problems with cloud native applications.

The benefits that developers gain with cloud native Java translate directly into benefits for their enterprise customers.

Portability increases flexibility

Developers can write applications once and deploy them in many different ways and in many different environments, today and tomorrow. This is particularly important in critical areas of application development, such as ensuring the security of enterprise applications, because it means trusted and proven code can be consistently applied to any number of different containers and used across multiple enterprise applications.

Applications can be developed for a specific container architecture today, then ported to a different container architecture in the future if required. This is enabled by using cloud native application programming interfaces (APIs) that are designed to work across container architectures.

Portability also extends across clouds to enable migration from one cloud platform provider to another. This capability is essential for moving mission-critical Java enterprise applications from monolithic architecture to cloud native architectures.

Scalability increases agility

It's not hard to figure out why it's so much easier and more practical to scale highly containerized applications than monolithic, non-cloud-native applications. It no longer makes sense for developers to support a massively horizontal or vertical scalable infrastructure to ensure that all use case requirements are met in a reliable way.

Architectures that leverage orchestrated containers and microservices make it easy to optimize scalability in cloud native Java applications.

Simplicity increases speed and productivity

The programming model for Jakarta EE is built for speed and productivity.

Many of the key pieces are available to developers and ready for use. This eliminates the need for developers to search for the right capabilities and to find third-party components that must then be built and linked to applications. All of the components needed are part of the infrastructure.

This simplicity means that developers can focus on the business logic and not on the infrastructure because much of what they need is ready for use. Simplicity also extends to application maintenance and upgrades.

Automation increases efficiency

Automation is key to defining and building environments as quickly as possible. The phrase used often today is “infrastructure as code” — the ability to very easily and efficiently define what the target state should look like and what the environment that runs the software should look like. Then, it’s simply a matter of executing these elements in that technology.

With Docker containers, it’s very efficient and fast to include everything required to run the instance of the software and very fast to build the software. Similarly, with Kubernetes, an entire cluster environment with services and deployments can be defined, built, and shipped in a very efficient way.



"I used to develop applications that were monolithic. Nowadays, I've separated that logic out into several individual smaller applications that only consist of one or two tasks. It makes these applications easier to maintain and it means I can deploy them across a number of different databases and communicate from a number of different operating systems, if desired."

Josh Juneau, Java Developer

"We could build everything ourselves, and many companies have done that in the past. But to keep up with a fast-moving world, we need to get as quickly as possible from source code to deployed features in production that can deliver value to our clients."

Sebastian Daschner, Lead Java Developer Advocate, IBM

"What's really important to me is the time to first commit. How long does it take until you become productive? With Jakarta EE, you can become productive in 30 seconds or a few minutes. I can do it over and over again. And it is actually good enough for the cloud."

Adam Bien, Freelance Software Architect and Developer

► Software vendors have new opportunities to build their brand

When software vendors engage in evolving cloud native Java at the Eclipse Foundation, they get more value from their software investments than they would if they tried to go it alone.

- > **Influence** how the technology evolves to help ensure they can deliver the cloud native capabilities their enterprise customers need
- > **Incorporate** the results of other vendors' pooled investments and development efforts in their own commercial offerings
- > **Increase** their long-term relevance and value in the marketplace with technology that evolves based on broader decision making rather than narrow, vendor-specific interests
- > **Expand** their developers' knowledge and expertise through contributions to Jakarta EE projects
- > **Ensure** their developers are aware of Jakarta EE technology developments, to the ultimate benefit of customers
- > **Leverage** Jakarta EE Working Group marketing services to attract new developers and users to Jakarta EE to increase adoption of their Jakarta EE projects and grow market share for their commercial products

The path to true thought leadership lies in community involvement

The Jakarta EE community gives software vendors access to a global pool of experts who are committed to ensuring cloud native Java remains at the forefront of enterprise cloud software development. Access to this deep pool of knowledge and understanding allows software vendors to extend their own knowledge and bring their customers best practices in using cloud native technologies to build enterprise applications for a modern world.

Software vendors that take a leading role in evolving cloud native Java strengthen their thought leadership role and credibility on two fronts:

- > **Customers:** By increasing their customers' understanding and knowledge of how best-of-breed, cloud-native technologies can be used to streamline operations, increase efficiency, and accelerate business transformation.
- > **Industry:** By reinforcing their reputation as a community-oriented, driving force for continuous improvement and evolution of open source, cloud native technologies.



Together, software vendors are stronger

With the undeniable relevance of Java and the need to migrate so many existing Java applications to cloud native environments, it's essential for software vendors to align around the Jakarta EE community.

Many leading software vendors have already recognized that a united approach is the only way to avoid industry fragmentation and build a cloud native Java ecosystem for the long term. The Jakarta EE community is constantly growing and is backed by world-leading companies in the Java industry, including:



► Enterprises must evolve key strategic assets — their Java EE applications

Enterprises in a variety of industries — particularly financial services and insurance — have relied on applications written in Java for years. They have large teams of Java developers working to solve critical business problems. And, like developers and software vendors around the world, they've recognized the many benefits of migrating to cloud native Java.

The Jakarta EE community is a new and unique opportunity for these enterprises to have a real voice in shaping the cloud native Java specifications process and to play a role in defining Jakarta EE strategic themes and priorities to better meet their requirements.

All of the benefits that developers and software vendors gain with the move to cloud native Java apply equally to enterprises. As active members of the

Jakarta EE Working Group, enterprises can:

- > Play a role in defining Jakarta EE strategic themes and priorities
- > Ensure the quality and sustainability of technologies that are critical to their businesses and customers
- > Gain insights into the cloud native Java technology roadmap
- > Collaborate with other enterprises through professionally managed development initiatives
- > Protect their strategic investments in Java EE
- > Provide an appealing career path for their in-house developers

The Eclipse Foundation recognizes the high value that enterprises have to offer and has developed a Jakarta EE member profile designed specifically for enterprise members.

► The road ahead: Short term priorities for cloud native Java

With the high number of Java EE applications in use today, full compatibility between Java EE 8 and Jakarta EE 8 is extremely important. Compatibility dramatically simplifies migration requirements. It also dramatically reduces the risk that major issues or roadblocks will be encountered as applications are migrated. Developers, software vendors, and enterprises can be confident their existing applications will run on Jakarta EE 8 implementations.

In the short term, it is crucial for the Jakarta EE Working Group to explore Java EE capabilities that are not required in cloud implementations. It is equally important to add cloud-specific capabilities that are not provided in Java EE. These initiatives must include adaptations to:

- > Evolve some aspects of older Java EE code to make it more flexible and cloud-adaptable

- > Provide greater ability to develop applications in containerized environments

The open, community-based process ensures these adaptations are implemented in a way that benefits the majority of members. Members determine the technical priorities for the short term and the longer term. They also determine how those priorities are executed and implemented over time.

A strong focus on the basics

Each member of the Jakarta EE community has unique requirements for scalability, application complexity, business functions, and other capabilities. The key will be to prioritize the next steps for Jakarta EE evolution by carefully considering the capabilities that are necessary right now, and those that should be part of a longer-term plan.



The challenge is not every company has the same problems and companies sometimes try to solve problems that affect only a few customers. It's called cargo cult programming and it's like trying to solve non-existent problems or doing something without knowing why you're actually doing it."

— Adam Bien, Freelance Software Architect and Developer

"I think the first priority is to meet the needs of the majority of developers, which is really just maintaining the stability of the APIs, which we've had for a while, as well as moving capabilities that do meet the requirements of the majority into Jakarta EE."

Reza Rahman, Principal Program Manager for Java on Azure, Microsoft

"In the short term, I see compatibility as very important to enable people to migrate applications with ideally zero effort — basically making it fully binary compatible."

Sebastian Daschner, Lead Java Developer Advocate, IBM

Top Jakarta EE community priorities

According to our 2019 survey, microservices is the leading architecture for implementing Java in the cloud.

- > Microservices: 43%
- > Hybrid: 20%
- > Monolith: 13%

Not surprisingly, the top two priorities for the Jakarta EE community are related to the preferred architecture for cloud native implementations:

- > Better support for microservices: 61%
- > Native integration with Kubernetes: 61%



Microservices and Kubernetes are on my priority list. As far as working with Kubernetes, that's just a given if you have microservices. You need the ability to orchestrate these services together in an easy way."

— Josh Juneau, Java Developer

Jakarta EE 8 compatibility certification is underway

The certification of the Eclipse GlassFish 5.1 application server as a Jakarta EE 8-compatible implementation is a major milestone that demonstrates the power of diverse, community-driven collaboration.

Eclipse GlassFish is a complete application server that includes implementations of all required and optional Jakarta EE APIs, and passes all Jakarta EE TCKs. GlassFish also includes a complete administration console, clustering support, and other developer- and production-focused tools and features.

The GlassFish compatibility certification is just one of many expected certifications. Major software vendors who are members of the Jakarta EE Working Group have also announced they intend to certify their Java EE 8 compatible implementations as Jakarta EE 8 compatible.

► The road ahead: A transparent and standards-based future

Over time, Jakarta EE will diverge from Java EE to become even more cloud-centric. This divergence is necessary to ensure that cloud native Java remains relevant, can effectively compete with other cloud native programming languages, and becomes the natural choice for ground-up cloud application development.

The process for delivering Jakarta EE specifications is fully transparent and follows the Jakarta EE Specification Process (JESP). The JESP is based on the

Eclipse Foundation Specification Process (EFSP), and is tailored for Jakarta EE.

While there's a huge need for Jakarta EE community members to agree on Jakarta EE standards and specifications, it will take multiple parties to improve the way specifications are implemented. Understanding and maintaining the technologies server-side developers need will be key to meeting their requirements and broadening the Jakarta EE ecosystem.



Divergence from Java EE will allow Jakarta EE to move much more fluently and evolve much more quickly. Even though Jakarta EE is a standard, it's going to have to evolve at a fast pace and separate itself from the Java EE platform to become a true cloud native platform."

— Josh Juneau, Java Developer

► Engage in cloud native Java

It's time for developers, software vendors, and enterprises everywhere to join the Jakarta EE community and engage in turning the huge potential for cloud native Java into reality.

Success drives success. The more companies and people that commit to cloud native Java and contribute to its evolution, the more robust and relevant the community and the technology will become.

Developers and their employers can't rely on closed systems with agendas and priorities they are unable to influence to deliver what they need. A "by developers, for developers" approach that is aligned with the core values of open source software development is the best way for community members to remain at the forefront of:

- > The latest advances in cloud native Java programming and technologies
- > Best practices in cloud native Java application development
- > Innovative approaches to problem solving
- > New architectures for developing more agile, efficient, and cost-effective enterprise applications for single and multi-cloud environments

Contribute to a vibrant Jakarta EE community

The ultimate Jakarta EE community is a vibrant mix of people who understand the existing, mature technology set as well as cloud native development experts, and those who have never developed in Java before. This broad-based and highly engaged community will create unique and important opportunities to evolve a very established technology with new and exciting capabilities.

The involvement of major software vendors and leading Java developers in the Jakarta EE community confirms that cloud native Java is considered to be a high-value technology. And events such as JakartaOne Livestream that are dedicated to furthering Jakarta EE and related technologies demonstrate the broad community engagement that is already underway.



“

I would encourage everybody to get involved, whether they can contribute code or just contribute ideas. Ideas are in many ways as important as contributing a single line of code.”

— Mark Little, VP Engineering, Red Hat



“

What I've said a lot is, just get involved. Even if developers cannot or do not want to actively contribute to the way standards are formed and shaped in the future, it helps just to have a look at the Eclipse Foundation, at the Jakarta EE platform, and be informed about what might be out there in the future. And it helps us to receive feedback about how the technology is being used in projects.”

— Sebastian Daschner, Lead Java Developer Advocate, IBM



► Become an Eclipse Foundation member today

The Eclipse Foundation is a not-for-profit organization supported by more than 275 member organizations who value the Foundation's unique Working Group governance model, open innovation processes, and community-building events.

The Foundation provides a global community of individuals and organizations with a mature, scalable, and business-friendly environment for

open source software collaboration and innovation. It is home to Jakarta EE, MicroProfile, the Eclipse IDE, and more than 350 open source projects including runtimes, tools, and frameworks for a wide range of technology domains including the Internet of Things, edge computing, automotive, geospatial, systems engineering, and many others.

Make your mark in the Jakarta EE Working Group

The Jakarta EE Working Group is the home of open, cloud native Java innovation. Within the Eclipse Foundation's unique Working Group governance model, the world's Java ecosystem leaders are collaborating on advancing enterprise Java technologies to support the migration of mission-critical applications and workloads to the cloud.

Jakarta EE Working Group members:

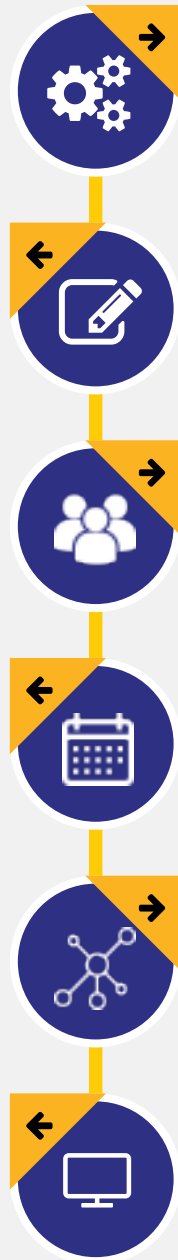
- > Play a key role in defining Jakarta EE strategic themes and priorities
- > Shape the definition and evolution of the specifications process
- > Gain insights to the technology roadmap
- > Help protect their past strategic investments in Java EE

Get the broader benefits of an Eclipse Foundation membership

Take advantage of marketing programs that target the Eclipse developer community, including exposure for commercial products on Eclipse Marketplace Client

Receive discounts on sponsorship and registration fees for Eclipse Foundation events

Add the Eclipse Foundation member logo to your site and have your logo included on our members' page and in other promotional material



Join and participate in Eclipse Foundation Working Groups, industry collaborations that develop new open source software platforms

Participate in the Eclipse Board of Directors and guide the strategy of the Eclipse Foundation (based on established guidelines)

Network with other member organizations through Foundation-run events and one-on-one introductions by Foundation staff



Join us today

For more information about Jakarta EE, and about joining the Jakarta EE Working Group, please visit:

<https://jakarta.ee/membership>