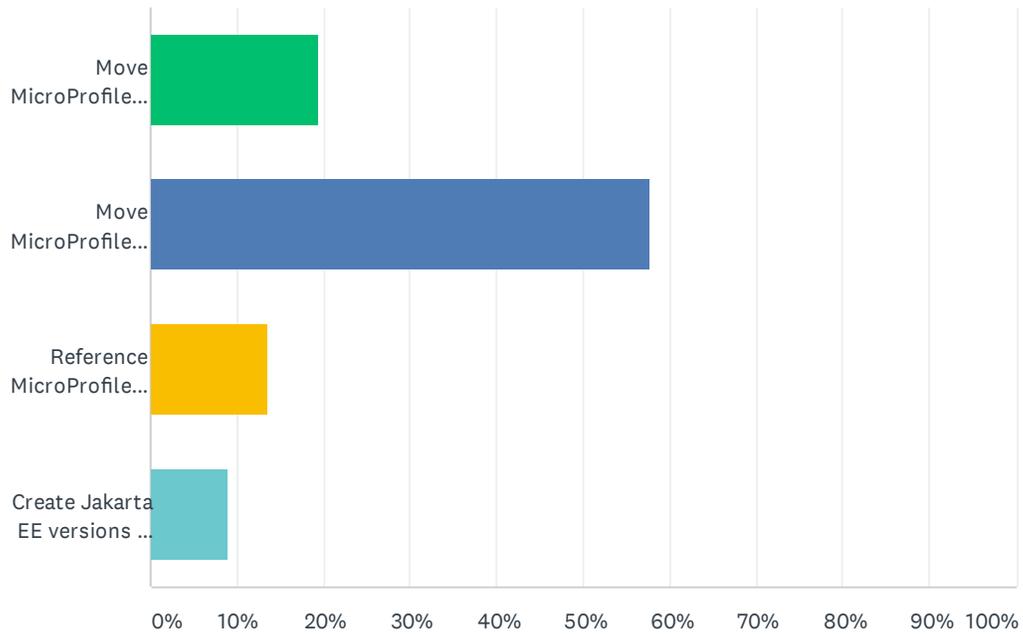


Q1 How do you best prefer Jakarta EE to consume MicroProfile specifications?

Answered: 220 Skipped: 0



ANSWER CHOICES	RESPONSES	
Move MicroProfile specifications to Jakarta EE without changing namespaces.	19.55%	43
Move MicroProfile specifications to Jakarta EE including the namespace.	57.73%	127
Reference MicroProfile specifications in Jakarta EE and not move MicroProfile specifications.	13.64%	30
Create Jakarta EE versions of MicroProfile specifications.	9.09%	20
TOTAL		220

Q2 Please share any additional thoughts

Answered: 53 Skipped: 167

#	RESPONSES	DATE
1	Microprofile should evolve APIs that eventually get absorbed by Jakarta EE. Microprofile applications should eventually be able to run with pure Jakarta EE APIs	4/4/2021 9:51 AM
2	Moving MP specs into JEE including namespace will make clear where the longer term specs are maintained. Also, for MP users it's a very easy migration path (you could even provide something similar to angular schematics to update).	4/1/2021 1:56 AM
3	I think no matter which of these options is chosen there is going to be an effect on either end users, or developers. Therefore, I would rather make the large upfront breaking changes all at once and merge the two into the same namespace. Then, have consistency going forward.	3/31/2021 11:14 AM
4	Microprofile and EE have different goal, lifecycle and design so can be neat to keep them fully split. In other words: if you reference MP in EE you must reference all vendors of any lib running on EE to be fair. A merge would mean breaking changes which would mean the abandon of EE for most of my customers still using it.	3/31/2021 9:43 AM
5	Why not have a single JakartaEE umbrella project, with a small core and separate project/modules that can have their separate release cycle and a status of "stableness". So if there is a new spec that wants to be done, it just makes another project under the umbrella, with a maturity of incubating. Then it moves to active, then stable with retrocompatibility promises. There can be "profiles" that keep together many specs, and also these can be labeled as "stable" or not. So current Microprofile would be a JakartaEE profile that is not yet "stable", but "active".	3/31/2021 3:31 AM
6	I would see the movement from org.eclipse.microprofile to the jakarta namespace as a sign of maturity (and success) for MicroProfile.	3/30/2021 8:08 PM
7	Option A2 has fewer cons and is more end user friendly. Any other approach may be causal for end users changing their development efforts for options more consistent in the market. There are some other good options for microservices building. Make separate effort at MicroProfile group and Jakarta EE group will slow the development of both. Join efforts. Don't make separate efforts, please.	3/29/2021 11:56 AM
8	Ear deployment is the major drawback in my experience. The mp spec seems focused on war deployments, and vendors do not feel ear support is required. The way some spec like mp-config are required during deployment because some other subsystems may be depending on it (eg: configuration for mp-health, mp-jwt, mp-metrics, ...), and the mixing of classpath-accessible classes/cdi-accessible beans, makes it particularly hard for vendors to easily support ear deployments. As such, I think an integration of the mp spec within jakarta ee, and with ear deployment considered, is required. I don't care much if it becomes a jakarta spec or stays a microprofile spec, but I expect ear deployments to be fully covered if it makes it to the jakarta spec.	3/29/2021 2:41 AM
9	Oblige Oracle and all major player of environment of former specification to adhere completely to such specifications	3/28/2021 11:02 AM
10	The industry world do not move so quickly to benefit from a faster release of MicroProfile, and even NOW most developers don't know the difference between JavaEE, JakartaEE, Microprofile... I prefer a more clear, albeit slow, approach, mostly for the mass of developers that will meet JakartaEE.	3/28/2021 10:40 AM
11	Jakarta EE and MicroProfile should be one specification.	3/27/2021 7:20 PM
12	I believe this would be best decided by the MicroProfile Working Group and what they think would be of best interest of the general Jakarta EE community and other working groups	3/27/2021 8:29 AM
13	In the past versions, the MP specs updated very frequently and introduced some breaking APIs when upgrading to new versions, I do not think Jakarta EE can include the current MP	3/27/2021 1:21 AM

Your Input Needed to Determine Path for Jakarta EE/MicroProfile Alignment

directly. Some mature API or spec that is required in Jakarta EE can be considered to move to Jakarta EE(renamed it to Jakarta naming). As I stated before, I would like use MP as an extension of Jakarta EE specs and a sandbox place to embrace all cool things around the world. Like the purpose of Spring Experimental project. Jakarta EE itself should contain the standards that not changed breakingly in the future 5 -10 years, it is the base of the Jakarta EE ecosystem.

14	MP and Jakarta are both about standard specs and as a developer it makes little sense to separate these waters for no obvious reasons. It seems to me that right now this is more about territorial fighting rather than irreparable damage to the pace of innovation. Surely there is room for fast-paced evolution within Jakarta, just make a major release per year and an intermediate minor so that new MP versions can come abroad sooner. I've been at this industry for long enough to know that after the first couple of releases, there is little value for quarterly releases of such important things as specs (IDEs are other story, keep them coming as fast as possible)	3/26/2021 4:36 PM
15	Don't let die the project	3/26/2021 3:21 PM
16	The MicroProfile community has been adamantly against Option C (4th selection above). The MicroProfile community wants to continue to have the freedom to "move fast and break things." This goal would cause user confusion if option A2 (2nd radio above) were selected. The founding members of the MicroProfile community are staunchly opposed to becoming one working group. That excludes option A (and also A2). To maintain a semblance of clarity between the two methodologies (evolve in a rigorous compatible fashion and deprecate slowly and deliberately vs. evolve quickly and break APIs if necessary) the only choice that fits all these constraints, to me is Option B (3rd radio, above). Jakarta EE may decide to define "gateway" specifications that can form a basis for using MicroProfile in a tiered fashion but this is neither a requirement, nor should it be prohibited. Users will simply need to understand that MicroProfile specifications evolve in a different fashion -- much like they might if they were to use other non-standardized implementations. When using these APIs, adoption of new versions may require user application changes. In the immutable, containerized work-load context that MicroProfile tends to target, this may be the right procedural choice. My personal choice would be to bring the two working groups together and regularize the name-spaces. (Open A2). But I think that practically, the only pragmatic option, today is option B (3rd radio choice above)	3/26/2021 3:19 PM
17	Changing the namespace also allows more backwards incompatible changes.	3/26/2021 3:15 PM
18	I think MP should be viewed as an incubator for mature JakartaEE specifications and any specification (or subset of a specific specification) with sufficient interest and support should be pulled into jakarta.ee space and then MP can embrace and extend it as needed from that new source on their next release. This abuses MP but MP arose from an inflexible JavaEE community, JakartaEE is doomed if they fail to innovate and solidify the gains of MP since its inception. In other words, it is far more important that Jakarta and MP rapidly codify and define the rules of engagement and their relationship in a complementary fashion if either are going to get along. Business likes JakartaEE because of its stability backwards compatibility and developers like MP because of its innovation, there is room for both but not if there is a constant specter of someone taking their toybox home.	3/26/2021 2:38 PM
19	Jakarta EE vendors have demonstrated that the pain of dealing with package name changes can be mitigated by (temporarily at least) supporting both variants. That approach can be applied to Microprofile Config, as well as any other third party API/specification being considered for inclusion in Jakarta EE.	3/26/2021 2:05 PM
20	Using a different namespace makes it clear what version and expectations (backward compatibility) the user is making. Moving without a namespace is confusing (what am I using?, you can't say based on package name which is especially a problem on runtimes that support both) and just reference the specification is a great risk (no control). Creation is a duplication of work that can be avoided.	3/26/2021 1:20 PM
21	The aim of a specification should always be to make something as simple and clear as possible. The entry barriers and opportunities for error for new and inexperienced developers must be as low as possible. An inconsistent namespace or even the possibility of circular dependencies make the use simply too complicated and difficult. At the end of the day, it's all about the economic and productive development of applications.	3/15/2021 3:38 AM
22	faster jakarta ee minor releases with microprofile updates	3/14/2021 3:01 AM

Your Input Needed to Determine Path for Jakarta EE/MicroProfile Alignment

23	Jakarta EE should focus more on what's new than legacy. It's better to reference APIs that are out there than to duplicate a stale version which is a painful port. I would rather deal with more rapid change (methods being changed or dropped) and always have what's new than to have a platform that never changes.	3/13/2021 3:49 PM
24	use microprofile to test new ideas and when they are more solid transfer to Jakarta. in this way microprofile. * would be for test alfa/beta versions and ideas and jakarta.* for more solid implementations	2/25/2021 6:31 AM
25	the end result after migration seems the cleanest of all please don't introduce circular dependencies	2/19/2021 3:43 PM
26	keep the situation as it is right now	2/18/2021 2:20 AM
27	Microprofile should evolve independently of Jakarta EE. This way, innovation is faster.	2/18/2021 2:02 AM
28	I think this is the most consistent option for end-users and for broad adoption.	2/17/2021 5:12 PM
29	Move some MicroProfile specifications (e.g. MP Config like when its stable) to Jakarta EE including the namespace.	2/17/2021 2:46 AM
30	I think this maintain the innovation from the Microprofile and conservative from the Jakarta	2/16/2021 9:02 PM
31	I think it's more important to keep the release model consistent than the identifier.	2/15/2021 3:31 PM
32	Simplify as much as possible.	2/15/2021 2:21 PM
33	I would most like to see an option C2: create separate Jakarta EE versions of MicroProfile specifications *by taking snapshots of the corresponding MicroProfile versions and only replacing the namespace each time*. I.e. C without independent evolution.	2/13/2021 5:40 AM
34	As a Jakarta EE developer, I'd expect MP specifications to be fully integrated into Jakarta EE. So some MP specs would be subsumed by a Jakarta EE spec (for example MP JWT -> Jakarta Security and MP REST Client -> Jakarta REST), others would be moved to a new Jakarta EE spec (for example MP Config). So this rules out options 1 and 3. And since option 4 would be utterly confusing to me, this leaves us with option 2. Jakarta EE isn't just about backward compatibility, it's also about being a cohesive, well-integrated platform. So having e.g. both `jakarta.security.enterprise.authentication.mechanism.http` and `org.eclipse.microprofile.jwt` would not make sense to me. On a related note, I think the argument of "but Java SE also has non java/javax packages" doesn't make sense: organizations like W3C and IETF are industry-wide, non-Java-specific organizations. On the other hand, Eclipse MicroProfile and Jakarta EE are both EF-hosted, Java-specific projects which are developed/maintained by mostly the same organizations. (And yes, there's also the `org.sax` packages, but I'd argue that's a historical mistake we shouldn't repeat, rather than an example to follow.) Ideally, I believe MP should merge with Jakarta EE at some point, and everything should be moved into the `jakarta.*` namespace (and so, due to the backward-incompatible namespace change, the sooner the better). As a brand, I don't see how MP will survive in the long run. The name itself, "MicroProfile", doesn't make sense without referring to Jakarta EE's profiles. And as long as an MP implementation doesn't have to pass any of the Jakarta EE dependencies' TCKs, I don't see how people can take it serious as a portable platform. Finally, when you ask people what they use, they'll either say "Jakarta EE/Java EE/J2EE" or "Quarkus/Helidon" (or "Spring (Boot)/Micronaut/..."), but rarely, if ever, "Eclipse MicroProfile".	2/13/2021 4:45 AM
35	the persons in the working groups should be the same so if it's start es ah microprofile group it should be transformed into a jakarta working group. They who "invented" it should also be the ones who evolve it. Otherwise it would be like a forced adoption	2/12/2021 8:14 AM
36	Developers usually don't care very much about the package names they import, so Option A1 is viable. Option A2 would look nicer & more consistent, but lead to additional migration costs (hardly worth it). Option B sounds scary & complex because of stronger coupling between Jakarta & MP (and circular dependencies). Option C should be a no-go (duplication is the worst thing that could happen). It would be nice to have a central, official statement & comparison of the guarantees that Jakarta and MP make (with regard to prod readiness, backwards compat, and release cadence). This could address possible confusion with regard to prod readiness of MP. In the long run, I would prefer Jakarta & MP to completely merge under a common umbrella spec (as mentioned in my short blog article: https://enji.systems/2021/01/09/jee-mp-merge.html). Then we would not even have the need to decide whether / when / how specs	2/12/2021 2:31 AM

Your Input Needed to Determine Path for Jakarta EE/MicroProfile Alignment

should move from MP to Jakarta. Both projects are just great, and strongly benefit from each other.

37	As most people involved in MP are also involved in JEE, it should not hurt them too much when "their" specs move. Start with MP Config, as it's a core infrastructure and it should be easy for other JEE specs to reference it. MP Rest Client should be completely merged into JAX-RS. Then wait until other specs will start to need more complex dependencies and decide case by case.	2/11/2021 10:53 PM
38	When both MP and Jakarta are under the same organization, it makes no sense to keep them separate. MP can be considered as an upstream release for Jakarta. Similar to what Fedora Linux is to RedHat Enterprise Linux	2/11/2021 2:17 PM
39	I think that Jakarta EE must be a main standard to itself in IT market. Dmitri,	2/11/2021 1:15 PM
40	This should be done for apis that "graduate" to Jakarta EE to be included in the platform. Not all MicroProfile apis should graduate. This is aligned with the goals of the two initiatives and makes it clear where each platform starts and ends.	2/11/2021 12:48 PM
41	This aligns with the original idea behind MicroProfile. MicroProfile is already built on a base set of Jakarta EE dependencies. When a certain specification will be moved to Jakarta EE (important: at the appropriate time!) then it should be stable enough to integrate into Jakarta EE's release cycle. After that, MicroProfile can add this specification to the base set of Jakarta EE dependencies it already depends on, and remove the MP version from the MP platform if it so desires. And then the circle is full, resulting in a specification incubated in MP, adopted in a stable way by Jakarta EE.	2/11/2021 5:15 AM
42	They belong as one, not separate. All the arguments against smell like "implementation details" to me	2/10/2021 7:44 PM
43	I think it would be better for Jakarta EE and Microprofile to have the same namespace and specification in terms of the long term sustainability.	2/10/2021 7:42 PM
44	As a relatively new user to JavaEE/JakartaEE, I can say that namespace matching is an important part of understanding if this spec belongs to the framework I am using. If for example, JakartaEE adopts any other API from other vendors, let's say apache commons, but do not change the namespace to jakarta.* then there will multiple namespaces within the same platform. And if there is no consistency, then it does not feel/come across like a platform. It's more like a collection of things that someone put together and is asking/promoting others to use it because it's more stable. I do not want to come across as rude, I just wanted to share my honest opinion! Appreciate the work you guys are doing and looking forward to contribute in even more ways!	2/10/2021 5:44 PM
45	My organization is using Jakarta EE to reduce maintenance efforts. It's a government agency. The number of developer positions that we can fill is limited by the budget drawn up by the government. That means that we don't have the option of hiring more developers. We rely on the backward compatibility guarantees of Jakarta EE to reduce the effort for application maintance. It's important to us that any alignment of Microprofile and Jakarta EE does not weaken the strengths of Jakarta EE. The strengths of Microprofile are currently irrelevant to us because we don't have access to any cloud infrastructure. We won't use any specifications that don't provide backward compatibility. The name and/or namespace of the specification should make it immediately apparent if backwards compatibility is guaranteed or not. This applies both to Microprofile and possible incubating Jakarta EE specifications. Microprofile specifications that guarantee backwards compatibility can be renamed to use the jakarta namespace. If a specification doesn't want to provide backwards compatiblity or is not yet stable enough, it should not use the jakarta namespace. Jakarta EE should not reference Microprofile specifications. That introduces instability which we don't want to have in Jakarta EE.	2/10/2021 5:34 PM
46	I believe keeping MP specifications separate and referenced from Jakarta EE will result in the best user experience (package names remain stable) and also result in the least confusion regarding the owner/maintainer of the specifications. As a developer, I am perfectly content knowing that the servers/runtimes on which my software runs support two separate sets of specifications, but I do prefer that it remains obvious to which set a particular specification belongs (argument against option A). There are, of course, exceptions. For example the MP REST Client recently discussed on the mailing list. That particular case highlights that some	2/10/2021 5:23 PM

Your Input Needed to Determine Path for Jakarta EE/MicroProfile Alignment

MP specifications may be filling in the gaps of existing Jakarta specifications, and moving the MP specification to plug the hole is the best option.

47	It should be possible to evolve Jakarta EE specifications fast so that there's not a big gap between the speed of evolution of Jakarta EE and MicroProfile. If needed, the Jakarta specification process should be amended to allow faster evolution to decrease the gap.	2/10/2021 4:32 PM
48	MicroProfile should just be an incubator for new ideas, and the it shall be Jakarta EE that turns these ideas in final specifications.	2/10/2021 4:24 PM
49	I wish there shouldn't be too group at first place. If I remember correctly MicroProfile born out of slow moving Oracle's Java EE. I was hoping when oracle announced moving Java EE to Eclipse, MicroProfile also join under same. Another reason I think is now we moved packages from javax to Jakarta, it's breaking change. Anybody wants to migrate have to put effort to change from developer perspective. I don't want any dev effort to make MicroProfile to align with Jakarta. Why just keep MicroProfile 4 based Java EE 8(Jakarta EE 8) and create MicroProfile 5 based on Jakarta EE 9. Let Jakarta EE 10 consume all MicroProfile with new namespace that way dev have one migration plan to move from old javax. I guess that will not work as both community want to evolve separately. Best option is move without changing MicroProfile.	2/10/2021 4:22 PM
50	Jakarta is feature rick, solid and can be even more. I expect consistent market characteristics such as backwards compatibility and rationale release cadence - not that one by newest technology. Cant see a point for convergence just for one or two specification.	2/10/2021 4:12 PM
51	Changing namespaces is an extremely disruptive and drastic measure. Unless you are completely breaking the existing method signatures (as in re-architecting the entire spec), namespace changes ought to be avoided at all cost in order to avoid putting burden on the end user.	2/10/2021 3:23 PM
52	Hopefully best of microprofile is integrated into Jakarta EE in order to save the platform	2/10/2021 2:40 PM
53	Especially option 1 creates a huge risk by package-splitting which derails Java Module System (Jigsaw) modularity.	2/10/2021 2:21 PM