

# MicroProfile and Jakarta EE Alignment Alternatives

## Solution Target

Going forward, Jakarta EE wishes to consume MicroProfile specification X (such as MicroProfile Configuration). There are several alternatives as to how this could be done. The following is a brief analysis of the advantages and disadvantages of each approach. At the end of the analysis, there is a survey we ask you to weigh in on. In addition to choosing the option you believe to be best, it is very valuable to provide comments justifying your preferred alternative.

## MicroProfile and Jakarta EE Context

Both Jakarta EE and MicroProfile produce specifications that are intended for and used in cloud native and microservices use cases. In particular, MicroProfile has a specific focus on meeting the needs of cloud native and microservices use cases. MicroProfile also produces comparatively faster platform releases (roughly once a quarter) while the Jakarta EE release cadence is slower (the likely long term target being approximately once a year).

Jakarta EE currently provides relatively strong guarantees for backward compatibility for all specifications. MicroProfile does not currently guarantee backwards compatibility for all specifications, but does produce production-ready specifications that have demonstrated real world adoption. This characteristic enables MicroProfile to focus on innovation in emerging areas while Jakarta EE focuses on more conservative use-cases and stability best suited to the largest enterprises. While MicroProfile specifications have on occasion needed to break backwards compatibility, this decision is made with due care for end users.

MicroProfile specifications depend on one or more Jakarta EE specifications while Jakarta EE does not currently have any dependencies on MicroProfile.

**Option A1:** Move MicroProfile specification X to Jakarta EE without changing namespace (no need to change namespace from *org.eclipse.microprofile.\** to *jakarta.\**).

Nonetheless, the Maven coordinates for MicroProfile specifications will move to Jakarta. Further evolution will take place under the Jakarta EE working group.

## Pro:

- There is no need for existing MicroProfile users to switch namespaces.
- [Reza] This gives MicroProfile due credit going forward for bringing a specification into Jakarta EE.
- No API duplication between MicroProfile and Jakarta EE.
- There is only the existing one-way dependency between MicroProfile and Jakarta EE.
- [Reza] Some users may wish for greater convergence of MicroProfile into Jakarta EE. This option satisfies this desire to some extent.

Con:

- Lack of namespace consistency for Jakarta EE users otherwise not using MicroProfile.
- This can be seen as making Jakarta EE as the only place production ready specs can be developed
- [Rudy] MicroProfile brand is linked with fast changes, innovation and breaking changes which doesn't match Jakarta EE vision.
- [Reza] It is not immediately obvious to a casual user using both Jakarta EE and MicroProfile in the same application which MicroProfile APIs belong to the MicroProfile working group and which MicroProfile specification belongs to the Jakarta EE working group. This may lead to brand confusion for some users as well as mismatched expectations with regards to characteristics such as backwards compatibility.
- [Ondro] MicroProfile can't evolve the API faster than Jakarta EE, it can only include a new version of the specification faster and even that can cause problems when MP and Jakarta EE are used together.

**Option A2:** *Move MicroProfile specification X to Jakarta EE including the namespace.* In this case, the namespaces will be changed from `org.eclipse.microprofile.*` to `jakarta.*`. Further evolution will take place under the Jakarta EE working group. No more work will be done in the MicroProfile working group to further evolve a specification once it is moved.

Pro:

- [Reza] Namespace consistency for Jakarta EE users otherwise not using MicroProfile.
- No duplication between two working groups
- [Reza] There is only the existing one-way dependency between MicroProfile and Jakarta EE.
- [Reza] It is immediately obvious to a user using both Jakarta EE and MicroProfile in the same application which specifications belong to the MicroProfile working group and which specifications belong to the Jakarta EE working group.
- [Reza] Some users may wish for greater convergence of MicroProfile into Jakarta EE. This option satisfies this desire to some extent. This may include a possible preference for the `jakarta.*` namespace, which is more generic - as opposed to the `org.eclipse.microprofile` namespace, which may imply a focus on microservices.
- [Rudy] Developers can choose if they use the classes from Jakarta EE and expect stability and backwards compatibility or the MP one (If the runtime supports it) and go for faster turnaround and innovation that might result in breaking changes in the next release.

Con:

- Existing MicroProfile users will need to switch namespaces in order to take advantage of newer versions of moved specifications. Similarly, implementers will need to put effort

towards migration, including potentially maintaining two separate work streams at least in the short term.

- This can be seen as making Jakarta as the only place production ready specs can be developed
- [Roberto] Implementations may require two development streams for each namespace if breaking changes cannot coexist
- [Roberto] Three APIs to choose from: The Jakarta API, the MP API and the Implementation API.
- [Reza] Some users may perceive this to mean only Jakarta EE is where production ready specifications are available.
- [Ondro] MicroProfile can't evolve the APIs faster than Jakarta EE, it can only include a new version of the specification faster and even that can cause problems when MP and Jakarta EE are used together.

**Option B:** *Reference MicroProfile specification X in Jakarta EE and not move MicroProfile specifications.* Jakarta EE will not duplicate any referenced specifications and MicroProfile specifications will only be evolved under the MicroProfile working group.

Pro:

- MP Spec X can be released as a standalone spec/API that is usable as is, in Jakarta EE X, and MP 4+
- [Reza] No API duplication between MicroProfile and Jakarta EE.
- [Reza] No migration effort is needed for any users or implementors, while Jakarta EE can still use the specification.
- [Reza] Some users may wish for MicroProfile and Jakarta EE to remain as separate as possible. This option satisfies this desire to some extent.

Con:

- This introduces circular dependencies between Jakarta EE and MP. e.g. MP 4.0 aligns with Jakarta EE 8. If Jakarta EE 8 Spec X relies on specs in Config 1.4 MP 3.3 because Jakarta EE 8 released before MP 3.3. A microservice wants to use both Jakarta EE 8 and MP 4.0. Which Config it will end up with? If Config 2.0 was loaded, do we know Jakarta EE 8 Spec X function ok as it was not being tested when it was released? This might make it impossible to implement both the latest MP version and the latest Jakarta EE version in the same product.

The following is one possible example illustrating version dependency mismatches.

MicroProfile m2 aligns with Jakarta EE j1, while Jakarta EE j1 aligns with MicroProfile m1. MicroProfile Configuration c2 is included in MicroProfile m2. Jakarta Persistence p1

relies on MicroProfile Configuration c1 in MicroProfile m1 because Jakarta EE j1 was released before MicroProfile m2. An application wants to use both Jakarta EE j1 and MicroProfile m2 together. Which MicroProfile Configuration version will the application end up with? If MicroProfile Configuration c2 in MicroProfile m2 was loaded, Jakarta Persistence p1 may not work with MicroProfile m2 as expected when tested and released via the Jakarta EE j1 compatibility test kit/TCK).

- It may not be possible to support the latest MP and Jakarta EE in the same product
- [Rudy]The Pull model voted by MicroProfile says that it will not adjust any of his procedures for another group. It is up to the other parties to adjust to or fork the MicroProfile specifications.
- The referenced MicroProfile specification may wish to break backwards compatibility at some point in its evolution while Jakarta EE does not. Additional efforts will need to be made to address such mismatches.
- [Reza] Lack of namespace consistency for Jakarta EE users otherwise not using MicroProfile.
- [Reza] It is not immediately obvious to a user using both Jakarta EE and MicroProfile in the same application which MicroProfile specifications are referenced by Jakarta EE and which are not (and as a result have different expectations with regards to characteristics such as backwards compatibility).
- [Reza] If Jakarta EE integration specific changes are required in MicroProfile specifications, it will require coordination across working groups in a timely fashion with regards to dependencies, release cadence and features.

**Option C:** Create Jakarta EE versions of MicroProfile specifications. In this case, Jakarta EE and MicroProfile will develop similar features in parallel.

Pro:

- MP Spec X evolves as seen fit across the two independent platforms
- [Rudy]In line with the view of the MicroProfile Pull model that MP will not change any of his procedures for other parties.
- [Ondro] It would always be possible to implement the latest version of MP and Jakarta EE in the same product

Con:

- This will introduce a lot of confusion between two versions of X and end users will have difficulties to work out which one to use. Sometimes contradictory behaviour might be introduced as they evolve differently.
- More effort needed to maintain the alignment or reduce conflicts.
- Duplication of effort and resources across Jakarta EE and MicroProfile. The duplication of effort will likely also extend to implementations.

- MicroProfile and Jakarta EE will very likely be seen as directly competing efforts, leading to further confusion.

[Tomas] **Option D:** Create a new Jakarta EE specification X based on *MicroProfile specification* X. Further evolution will take place under the Jakarta EE working group. No more work will be done in the MicroProfile working group to further evolve a specification once it is moved. The specification would be re-visited to align design and architecture with Jakarta EE.

Pro:

- Namespace consistency for Jakarta EE users otherwise not using MicroProfile.
- No duplication between two working groups
- There is only the existing one-way dependency between MicroProfile and Jakarta EE.
- There is a possibility to have older versions of MP supported on newer version of Jakarta
- The new specifications are consistent with other Jakarta specifications
- Designed to last (as opposed to MP designed for change)

Cons:

- As for other options that change namespace (can be mitigated if aligned with change from javax to jakarta in MP)

### **Voice Your Opinion!**

- Survey link to be included later, ideally collected by a truly independent party such as the Eclipse Foundation, InfoQ or DZone.