

Goals and Requirements

- Some high-level definitions:
 - A “Specification” is a collection of package definitions. (In the future, these may possibly be arranged in JPMS modules.) Specifications must integrate with one or more Profiles (as described below) but can optionally be used independently of Jakarta EE.
 - A “Profile” is a collection of Specifications standardized in a Profile Specification. The “Full Profile” is unique in that it defines the Jakarta EE Platform. Profiles do not have to be arranged in unique subsets. I.e. a Specification may appear in more than one Profile.
 - Profiles define what vendors must provide in order to claim compatibility and therefore what things the brand can be associated with. In other words, to be certified as compatible, independent implementations must implement one or more Profiles and must be certified for the Profiles to which they claim compatibility..
 - Each Profile must be defined by a separate Specification.
 - Each Profile may optionally add a TCK that tests the interactions between its Specifications.
 - Any implementation of a Profile may be offered by a vendor in Modules so that if a developer wants to only deploy a portion of the Profile they can do so
- There are three separate, but related aspects to this process:
 - The development of the specifications themselves;
 - The process of ratifying and promoting official specifications; and
 - Certification of implementations of ratified Specifications and Profiles
- **Specification Artifacts:** Each specification must deliver three artifacts:
 - Specification
 - Reference Implementation
 - Technology Compatibility Kit
- **Lifecycle:** Each specification needs to have a lifecycle. Each phase explicitly named. Be able to clearly communicate what state each specification is in.
- **Compatibility:** All stakeholders in Jakarta EE have reiterated their desire to retain a strong notion of compatibility, tied to a logo-based certification process. Given the standalone and profile goals above, it probably makes sense to tie certification to profiles. E.g. a vendor would certify that their independent implementation would be “Jakarta EE Web Profile” certified.
 - Does the process support any notion of certification of any implementation of any (Jakarta EE) specification?
 - Does the process only support certification of implementations of profiles? Just some profiles? If only a subset, how is that subset selected?
- **Bootstrapping:** We need to bring over the existing JCP-based specifications, including at least some of the patent rights which were granted under the JCP process. It is

essential to do this in a way that enables independent implementations of any Jakarta EE specification to be created without having to also become a licensee of Java EE.

- **Open Source:** The specification process should be based on open source best practices of openness, transparency, and meritocracy.
 - Look for inspiration at MicroProfile, MVC, CDI, and Bean Validation, and the new open source process at OASIS.
- **Backwards Compatible** with Java EE at the Full Profile level.
- **Internally Consistent:** Create a platform which is internally consistent.
- **Standalone:** It should be a recommendation (but not a requirement) that an implementation of each specification run on its own on top of Java SE, or with as small a list of additional prerequisites as possible. Do not preclude subsetting. Optional dependencies are a helpful approach to ensure

The Jakarta EE Full Profile (a.k.a. the Jakarta EE Platform) is backwards compatible with the Java EE Platform and will continue to include all of the specs provided by Oracle when Jakarta EE launched.

- Note: the above is a goal. Obviously it will take resources to accomplish this, and we cannot state at the moment when this could be delivered.
- Question: In the future, does it still make sense to require that the Platform include every spec developed under the Jakarta EE brand? I think I can imagine scenarios where we want to add new specs which are *not* included in Platform.
 -
- Note that the above significantly changes the role of the specifications. Rather than defining a single large platform, we are defining a collection of specifications which are included in a collection of profiles.
 - If we no longer have a single large server definition as the explicit goal, then perhaps it would be possible to consider having different profiles releasing on different timetables. Does it really make sense to insist that the Microservice Profile and the Full Profile have to release on exactly the same day? What are the implications of that idea?
- **Full Profile:** All new versions of Jakarta EE specification must provide a Full Profile which defines the definition of the full platform. Additional profiles may also be defined. Independent implementations must be compatible with at least one profile.
- **Profiles:** We need to have a well-defined and relatively lightweight process by which new profiles can be defined.
 - A Profile is simply a collection of specifications which are combined to produce a runtime, and version managed together.
 - A Profile may have an additional RI and TCK beyond just those of its constituent parts.
 - Strawman: A Profile may be created with the support of at least three Jakarta EE Strategic Members. (The reason to tie it to them is that ensures that at least there are some vendors that want to support this profile in their products.)

- **Ratification/Promotion:** Specification development produces some collection of artifacts. What is the process to take that collection of artifacts, ratify them, and promote them to official status?
- **Java SE Currency:** Jakarta EE needs to have a stated position on how current it will stay with Java SE releases.
 - Related: When will Jakarta EE embrace Java 9 and Jigsaw modularity?
- **Release Cadence:** Jakarta EE needs to have a stated position on how frequently releases of the specification are done. I think it is perfectly fine to say “we don’t know yet” up until we ship Jakarta EE 9 in (say) Q2 2019.
- **Certification of Compatibility:** Certification of independent implementations as meeting compatibility requirements will be made transparently by the Specification Committee. (E.g. the materials provided to the committee demonstrating compatibility, and the decision of the committee will be part of the public record.)