

Independent Implementations

Or: What is a JSR supposed to enable, anyway?

Do we agree that: (?)

- Enabling multiple stand-alone, and (potentially) independent implementations is core to what the JSPA is supposed to provide for?
- A known-at-time-of-JSR-approval license needs to be offered for each the Spec, RI, and TCK ?
- The JSPA requires the JSR Spec Lead to include some minimum terms in the Spec, RI, and TCK licenses?
- The Spec Lead may offer other licenses terms to interested parties, but **MUST** offer the licenses specified in the JSR.

Do we agree that: (?)

- That (regardless of FOU controversy), the SPEC, RI, and TCK license specified in the JSR MUST enable stand-alone, independent implementations of the SPEC ?
- That this is pretty much the only point in having a JSR to begin with?
- Otherwise, we are just documenting the One-and-only-ever-allowed implementation...

So what is this doing in the JSR 336/337 TCK License?

... In addition, to be a Product, a Licensee product that implements a Java Environment Specification must: (a) have a principal purpose which is **substantially different from a stand-alone implementation** of that specification, while the value-added portion of the product operates in conjunction with the portion that implements the Java Environment Specification; (b) represent **a significant functional and value enhancement over any stand-alone implementation** of that specification; and (c) **not be marketed as a technology which replaces or substitutes for a stand-alone implementation** of that specification.

Suggestion

- Include language that specifically requires that the JSR Spec, RI, and TCK licenses **MUST** all allow independent, standalone implementation of the Spec.