Spec Committee Meeting minutes January 16th, 2019

Attendees (present in **bold**):
Kenji Kazumura - Fujitsu, **Michael DeNicola**
Dan Bandera - IBM, **Kevin Sutter,** Alasdair Nottingham
Bill Shannon - Oracle, **Ed Bratt**, Dmitry Kornilov
Steve Millidge - Payara, **Arjan Tijms**
Scott Stark - Red Hat, Mark Little
David Blevins - Tomitribe, **Richard Monson-Haefel**
Ivar Grimstad - PMC Representative
Alex Theedom - Participant Member
Werner Keil - Committer Member

Eclipse Foundation: **Wayne Beaton, Tanja Obradovic**

- Past business / action items
    - Meeting min Jan 9th - approved for publishing
    - Alex moved, second from Ivar, no objections
- Jakarta EE Town Hall call for the community
    - We'll use Jakarta EE Working Group timeslot, Jan 30th
        - Review of 2018
        - Look ahead
    - Panel participation
        - Mike M., Kevin, David, Dmitry, Ivar, Dan?
        - Heiko Rupp, MC
    - Agenda
        - Mike M. will do a presentation addressing review of 2018 and next steps
        - Ivar / Tanja will prepare How to get involved and share with the Community
        - Q&A from the audience
- TCK process - David Blevins
    - TCK project nightly builds - once jakarta.ee/project pages are available it will be linked from there
        - Templates for project pages still in works
    - Java Batch TCK process - Kevin provided quick summary
        - The Java Batch TCK uses both JUnit and Test NG annotations. You can use either one. There is a thin utility layer (SPI) that can be used to detect a "job completed" event. Otherwise, the default mechanism is polling. I mentioned on the call that the TCK was then "thrown over the wall" to Oracle for integrating with the overall CTS bucket. Although that was a valid statement, I now understand that we also had to some glorified find/replace transformations on the tests to make them ready for CTS. For example, the CTS bucket couldn't handle JUnit's Assert library, so that usage had to be scrubbed before delivering to CTS. But, the actual integration of the TCK bucket into CTS was done by Oracle. As far as tracking challenges with the Java Batch TCK, this was done via github Issues.

  - CDI/Bean Validation TCK process - Red Hat
    - To be presented on next call on Jan 23rd
  - Refer to email “Understanding the Current Specification Process” and Understanding Java EE TCK Process
  - Understanding MicroProfile TCK Process - Kevin Sutter, please review for discussion in the New Year
  - Completion of the TCK process no later than **Jan 30th 2019**
  - The topic of distribution of TCK via Maven Central was discussed
    - Compiled ZIP files can also be distributed via EF downloads server.
    - Some of these TCKs are quite large and are produced several times a year, so Sonatype may not be happy with us.
    - May consider using EF hosted Nexus instance
    - Do we need consumers to explicitly agree to any terms to get access to the TCK for a Ratified Final Specification? (e.g. some sort of click through)
      - “We gotta have an audit trail” - Dan Bandera

Assumption: Jakarta EE 8 == Java EE 8 ie. no additional specifications, API, behavioral changes or new profiles.

Jakarta EE 8 release

Will consist of the following:

- **Specification** (documents and APIs) for the full platform including full and web profiles
- **TCKs** (source and binaries); part of the TCKs are documents on how to use TCKs
  - the Jakarta EE 8 will not attempt refactoring of the current monolith of the Java EE 8 TCK
  - Specifications that already have a separate TCK will remain as is
- **A Compatible Implementation of the full and web profiles,** at least one, available under open source license;

Required for the release:

- **Jakarta EE Specification Process** needs to be completed
- **Jakarta EE TCK process** needs to be defined
- Means to publish Final Specifications and related TCKs
- Branding process and logo usage guidelines

  - EF will be further working on identifying the work on the projects that can be done at this point

- Jakarta EE Spec process and Jakarta EE NoSQL project
  - based on EFSP v1.0, but not necessarily the same
    - Example: Jakarta EE Specification Process v1.0
    - EFSP v1.1 bugzilla link
  - We will use Jakarta EE NoSQL to test EF Spec Process and do customization if necessary to create Jakarta EE Spec Process.By going through the process with Jakarta EE NoSQL, we will capture notes and evolve a draft of the Jakarta EE Specification Process.
  - Jakarta EE NoSQL provisioning is next step; we are **waiting** on :

    - Board approval of the new IP policy (DONE)
    - Agreements need to be in place (MCCA and WGPA) for project lead/primary committer (Tomitribe)
    - Agreements for other committers (being pursued by the EF Membership Team via separate channels).
    - Werner - update on Jakarta EE NoSQL
      - Nothing to add. We're **waiting** on agreements to be signed. Hopefully we'll have what we need in 2019Q1. There is effectively zero chance that we will be able to bootstrap NoSQL in 2018.
  - The equivalent of a JCP maintenance review needs to be addressed (additional notes Nov 28th)
    - Please refer to Maintenance Reviews email thread.
  - We need a "Dials and knobs" document (possibly an FAQ) that details how a Working Group might extend the EFSP.
    - Suggestion to describe what is immutable.
- Jakarta EE Working Group calls
  - Next call February 2019
  - WG Agenda file
- Jakarta EE road map - on hold till all agreements are in place, however we should start planning
  - Platform Project team is looking into this
- Java EE Specification Documents contribution from Oracle -> this is still being discussed, however for now the answer is NO
  - No news are expected in next little while
  - Bill Shannon reported that the best case scenario is ⅓ specs will be contributed and ⅔ will not be (EJB and Servlet will not be contributed); criteria for deciding what specification can be contributed is not clear yet