

# Meeting summary — Jakarta EE Future Directions Interest Group (Feb 5, 2026)

Source: meeting transcript “Jakarta EE Future Directions - interest group meeting”

Jakarta EE Future Directions - ...

## Attendees (speaking roles observed)

- **Jan Westerkamp** (presenter)
- **Michael Redlich**
- **Steve Butler**
- **James Perkins**
- **Brian Stansberry**
- **Jared Anderson** (noted as host/recording early in the call)

## Purpose / agenda

- Share an update on **MicroProfile Config “future directions”** and the broader question of **moving MicroProfile working group content into Jakarta EE**, with emphasis on **namespace strategy**, **dependency hygiene**, and **migration mechanics**.

---

## Key discussion points

### 1) Background: why “Config” is resurfacing now

- Prior efforts to align **Jakarta EE** and **MicroProfile** (CN4J) and transition **MicroProfile Config** stalled due to asynchronous release cadences and limited cross-group bandwidth.
- Separate attempts to create **Jakarta Config** also stalled (feature-scope disagreements; later “minimum viable subset” approach didn’t stick).
- Renewed pressure: **Jakarta NoSQL** and other specs want a configuration story that avoids heavy XML/config burden and avoids depending long-term on an “outside org” spec (circular dependency concern).

### 2) The rejected IBM proposal and the central blocker: namespace

- IBM-backed proposal to move MicroProfile content to Jakarta **while keeping the `org.eclipse.microprofile.* namespace`** was rejected by the Jakarta Spec Committee.
- The group discussed the prevailing view that “moving to Jakarta should imply a **Jakarta namespace**” for long-term consistency and user experience, even if it requires a planned transition.

### 3) Two migration strategies on the table

- **Option A — “Big Bang” move of the MicroProfile working group content to Jakarta**
  - Would likely include a **namespace change** during the transition, using a migration playbook similar to Java EE → Jakarta EE (tooling like Eclipse Transformer/OpenRewrite-style automation discussed).
  - Follow-on decision: whether to keep a **separate Jakarta MicroProfile profile** vs pulling pieces (like Config) into Jakarta Core Profile. Rationale: some specs (example cited: telemetry) may require faster cadence than Jakarta’s platform rhythm.
- **Option B — Move MicroProfile Config first into a Jakarta Config spec**
  - Namespace changes are expected here.
  - Idea includes a **transition period** where a MicroProfile Config layer could be maintained atop Jakarta Config to preserve compatibility temporarily.

### 4) Structural/spec engineering issues (repo layout + automation)

- Jan called out the productivity advantage of MicroProfile’s **single-repo, multi-module** structure (spec + API + TCK aligned in one change-set).
- Jakarta specs often split into multiple repos (spec/API/TCK), creating friction for atomic changes and coordinated releases.
- Related point: MicroProfile’s **parent POM/automation** provides CI/release checks (including API-change-to-versioning discipline) that Jakarta doesn’t uniformly replicate.

### 5) A concrete technical prerequisite: split Config into CDI-dependent vs CDI-independent

- MicroProfile Config’s **dependency on CDI** prevents using it to configure CDI itself (dependency-cycle issue).
- Proposal: split into:
  - a **core, CDI-independent** config API (usable broadly, including for CDI or lower-level specs), and
  - a **CDI integration** layer.
- Open question: do the split **before** moving (on MicroProfile side for speed) or **after** moving (on Jakarta side for alignment).

### 6) Versioning / maintenance reality check

- MicroProfile Config has lagged because it was “expected to move,” leaving it pinned to older build infrastructure and older Jakarta/Java baselines.
- A MicroProfile release train was referenced (MP 7.2 planning), with Config changes framed as mostly build/parent updates (e.g., parent POM modernization) as a near-term step.

### 7) Governance + resourcing constraints

- Jan stated IBM is **not currently supporting a new transition** after the proposal failure, and that successful migration requires **compatible implementations** on both sides (MicroProfile and Jakarta).
- Process note: actual migration requires **formal votes/majorities** in both working groups.

## 8) Vendor concern: “configuring the server runtime” could become chaotic

- Brian Stansberry raised a major implementation-vendor concern: if many Jakarta component specs independently define runtime/server configuration, it risks a **fragmented configuration model** (“a mess”).
- Suggestion direction: if Jakarta adopts Config broadly, it may need an overarching **runtime configuration specification/model** to avoid spec-by-spec divergence.

## 9) Collaboration mechanics / transparency

- Discussion noted CN4J was deprecated and the group needs a reliable forum to coordinate MicroProfile–Jakarta transitions.
- There was interest in resurrecting a cross-community venue, potentially under **Jakarta Future Directions**, and inviting key stakeholders (names mentioned: Dimitri, Roberto, Emily).

## Decisions (explicit)

- **No formal decisions were made** during this session; it was a status + direction-setting discussion.

## Action items / next steps (implied)

1. **Restart broader discussion** in an appropriate cross-community forum (Jakarta Future Directions suggested).
2. Socialize and refine the two migration paths (Big Bang vs Config-first).
3. Drive consensus on:
  - **Namespace change strategy**
  - **Config split (core vs CDI integration)** and where to execute it
  - **Target Jakarta EE baseline** for the migration point (Jakarta EE 10 vs 11 referenced)
4. Identify/confirm **implementation ownership** on both sides to keep specs/TCK moving post-transition.
5. If Jakarta proceeds, define guardrails for **runtime configuration consistency** across component specs.

## Risks / watch-outs

- **Lack of vendor alignment** on scope for Jakarta Config (historically the root cause of stalls).
- **Fragmentation risk** if component specs define runtime configuration independently.
- **Resourcing risk** if major contributors do not actively sponsor the transition work.
- **Process friction** from mismatched repo structures and automation expectations between MicroProfile and Jakarta.