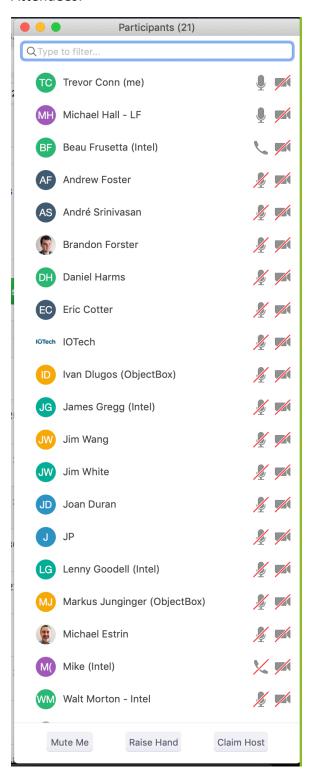
## **Core Working Group Meeting Notes (17-Jan-2019)**

## Attendees:



## **Old Business**

- Abstraction Layer for Consul (Issue 797)
  - o Pending plugin decision

## **New Business**

- Plugins
  - o 4 approaches proposed
    - Vanilla plugins
    - Optional plugins (Ian Johnson @ Canonical)
      - Default implementation in the repo
      - Plugin shim to wrap either plugin integration or default usage
        - Please clarify / explain a bit more
      - Retains Windows Support
      - Details from Lenny
        - Takes advantage of conditional compilation via \_suffix.go
        - Example
          - Consul\_windows.go
            - Calls native implementation
          - Consul\_linux.go
            - Calls out to plugin
        - If using native implementation (Windows) your choice is limited.
    - Why not use modules? (Ivan @ ObjectBox)
      - Description in #corewg-persist channel
      - Involves some code generation prior to build
        - Init.Db file has knowledge of the db provider
          - Configuration contains specification of desired provider
          - As part of make, the file is generated prior to build

- Could be done either locally or as part of build pipeline
- CI-CD Question
  - CI = one reference implementation
  - Certification = out of band certification testing process
- No plugins
- Discuss
  - Update from Beau on Windows plugin support
    - Had a call w/Microsoft dev contributors to Go project
    - Enthused to work on the outstanding GitHub issue but need to find the cycles
    - Need to sync knowledge w/r/t DLLs and compiler
    - Once it's more clear, need to size
    - Follow-up call in 2 weeks
  - Suggested by Jim W.
    - For Edinburgh Support for Redis/Mongo (as per Delhi)
    - During this dev cycle, POC for above ideas build time module integration, possibly plugin shim (per lan)
    - Allows for more time to continue conversation with Microsoft re: plugins
  - Expect POC from ObjectBox in coming weeks (will notify when ready)
  - Andre's estimate for Edinburgh March 15
- API Versioning in URL paths (Michael Hall raised this question)
  - o Do we need this?
    - /api/v1/device for example
    - Lenny / Michael Keep the version indicator but rev it for visual confirmation with integrating applications
    - Preference is to keep it Once we actually get to 2.0, we'll make a decision on supporting both versions going forward
      - E.g. multiple version request support
  - In the case of edgex-go all services are versioned together

- For this to be of use, multiple EdgeX deployments would have to exist in the same environment
- If we keep it, does that mean all routes rev to /api/v2 in our 2.0 release regardless of whether they've changed or not?
- Possibly discuss message queuing abstraction if we have time.
  - Postponed
- No Core WG Next Thursday (24-Jan) due to Application Services F2F in Boston