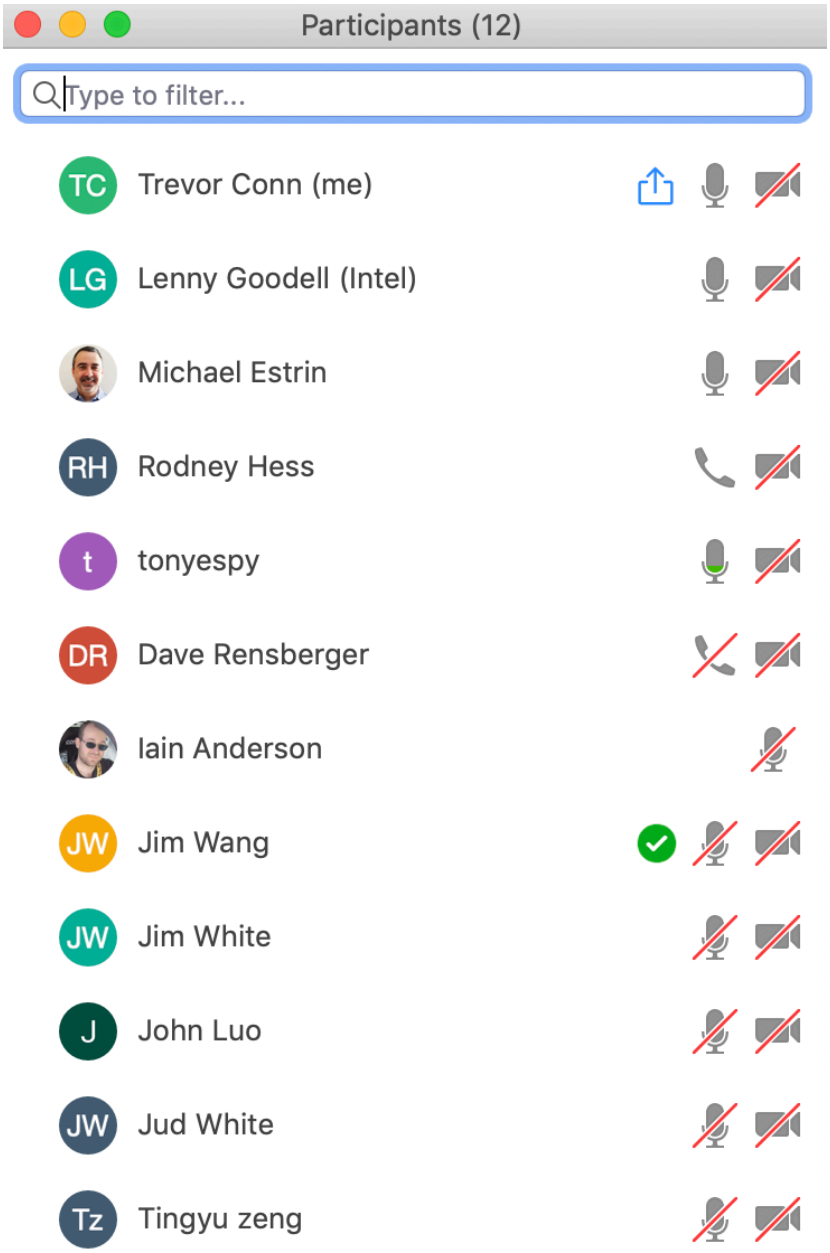
Core Working Group Meeting Notes (5-Dec-2019)

Attendees:



Old Business

- Proposal update: Common service bootstrap / DI container provided via Go module

    - Last meeting, we agreed to review once a holding repo was created and an example of integration into edgex-go was available for walkthrough.
    - Holding repo has been created
        - https://github.com/edgexfoundry-holding/go-mod-bootstrap
        - AFAIK integration example is not available yet due to holidays and other time commitments
    - Community to articulate extent of integration example required to give approval for moving the holding repo into main.
        - *Proposal to do a single service integration w/bootstrap module. Give folks some time to review, thumbs up/down.*
    - Anything to add?
        - *Trevor action item to start email vote thread for LF requirements to move holding repo.*
- Registry/Configuration proposal
    - Pending updates on the proposal from yesterday's Architecture meeting
    - *Once those points are included, we'll hold a vote to approve*
        - *Tee up for next week?*
    - *Two modules for this refactoring instead of just one (go-mod-configuration)*
        - *Go-mod-registry continues to exist, slimmed down.*
    - *Create holding repo for go-mod-configuration*
        - *Subsequent vote once holding is populated*
        - *Move the holding repo via LF request into main*
        - *Refactor services to use new module*
            - *Dependency → go-mod-bootstrap*
    - *Lenny action item start email thread for vote, Trevor to +1*
- Update on OpenAPI 3.0 (Swagger) docs for Geneva
    - https://github.com/tsconn23/edgex-geneva-api
    - First pass complete for all core/support services.
    - I have a few additional changes that need to be made based on discussions from Michael Estrin and I
        - Use of 207 Multi-Response
        - Support for operation agnostic /bulk endpoint
        - Leverage PATCH for partial updates
    - Would like some feedback on this.
        - *Jim to send notice to the WG chairs to review the above.*

        - *Need preliminary round of comments as soon as possible.*
        - *Deadline of review by Dec 15.*
        - *Using core-metadata as a working draft, still need to incorporate the three ideas above. And will also use to incorporate any agreed-upon feedback before proliferation to full suite of docs.*
- Value Descriptor changes
    - Summary from last meeting
    - Reading inheritance
        - BaseReading basis of SimpleReading, BinaryReading
        - OK with specialized reading types shown above
        - Removed superfluous properties (uomLabel, formatting)
    - Open question
        - How far do we go with validation?
            - *Today in device services, assertions are implemented but min/max is not*
        - Current validation resides only in core-data. Only validates the name of a value descriptor is valid.
            - *Lenny: App-Services would need a simple API or cache so as to not load the device profile in type validation is necessary*
            - *Discussion around where does the source of truth for value types lives?*
                - *Proposed: Core-metadata*
                - *Do we enable caching in consumer services?*
                - *Does metadata have the responsibility to message out deltas?*
                - *Proposal: Additional module that handles this communication and reconciliation of this information.*
            - *What about the role of VDs in UI / formatting?*
            - *Do we table this discussion and punt the implementation to Hanoi?*
            - *What level of usage for VDs by the UI?*
            - *GENEVA: Support for treating VDs as a response model that is returned from core-metadata as opposed to core-data.*

        - *UI will have to change the path to access this information from core-data to core-metadata*
        - *Spec out this interaction in the next few weeks to verify no backward incompatibility with this forward direction.*
    - Email discussion prior to Thanksgiving
        - Do we enforce validation only on the DS side?
        - Downside to that is that any source could push an event and there would be no server-side validation.
        - If we opt for server-side validation, then any service that could ingest an event (core/app-services) should perform that function.

New Business

- Dynamic device provisioning
    - Trevor has a few different proposed workflows to share
- Github issue triage for Geneva
    - This may be an agenda item for the meeting on 12-Dec. Putting it here as a placeholder.
- Any new business?
    - *MEstrin – Can we discuss elimination of core-data? Use case is a REST Device Service, providing mechanism for injecting data into the system via REST API. Isn't this the same thing as core-data's capabilities?*
        - *REST Device Service still depends on core-data's API for ingestion (Lenny).*
        - *If persistence is optional, repurpose core-data as support-data perhaps.*