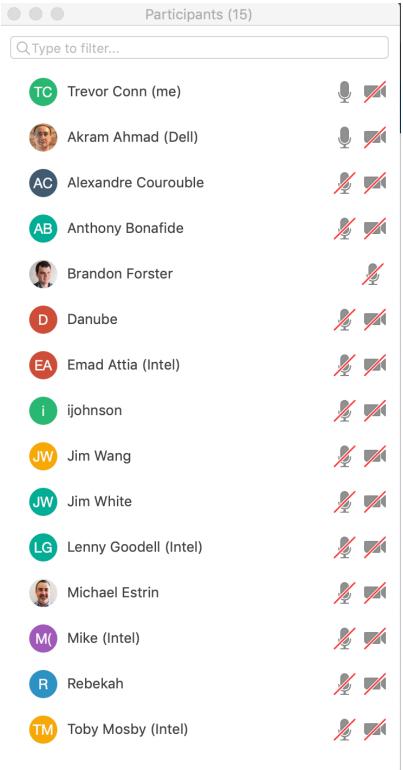
Core Working Group Meeting Notes (6-Jun-2019)

Attendees:



Old Business

- CapEx (Keith brought this up on TSC call 22-May)
 - I was unclear of the items to which this might apply
 - Tooling expenses, outside resources / services
 - Prep for Stage 3 project in LF
 - Deadline TBD (end of June)
 - Translation services
 - Good idea but keeping it updated may not be sustainable
 - Malini's feedback was that English docs weren't hard to consume but that content was outdated.
 - o Michael Hall
 - o Slack ??
 - Other tools for code quality and testing are being allocated by those specific working groups...?
 - Remove this, got it covered.
- Fuji deliverables
 - o Go 1.12 in Dockerfiles
 - https://github.com/edgexfoundry/edgex-go/issues/1401
 - Issue for Snap?
 - Snap will piggyback on the above issue.
 - Value descriptor creation only in core
 - https://github.com/edgexfoundry/edgex-go/issues/1071
 - On Dell's plate
 - Event/Reading timestamps in nanoseconds
 - No work necessary in core-contracts
 - https://github.com/edgexfoundry/edgex-go/issues/1404
 - On Dell's plate
 - Question RE: change in fidelity and how this is used from the UI
 - Does this entail a minor version rev? I get we're planning to do that anyway for Fuji (at least)
 - REST request backward compatibility How is the UI providing the value? If in milliseconds, comparison with nanoseconds will differ. Does the service need to pad it?
 - Or versioned API?
 - Events are also being returned to the caller as nanoseconds. How does this affect the caller's interpretation of that data?
 - Reach out to the UI team for input on this scenario (Trevor)
 - Decided we'll be dealing with this for Fuji but we need a "freeze exception" process for late changes (Tony)
 - Device provisioning support
 - Requirements definition owned by Device Service group. Will be a regular agenda item in that meeting

- Tony provided email brain dump of legacy implementation in the Java services.
- Right now, being tracked from the Core side via issue #826 but we may replace that with new issues (and possibly a project board) once requirements become more clear.
- Move API documentation from RAML to Swagger
 - IOTech QA/Test team has agreed to take this on. Removing from Core list.
- o Rules Engine
 - Check in with Lenny RE: Hashicorp plugins
 - Provided examples are simple, actual usage a bit more complicated
 - Working on a possible usage pattern for presentation next week.
 - Choice of RPC / GRPC
- Unit testing
 - I have the refactoring done for AddNewDevice to support my idea of increased test coverage, but I do not have the actual tests fully written yet.

New Business

- Was there any discussion on TSC call yesterday RE: LTS for Edinburgh?
 - No, but it doesn't sound like the view has changed that this should not be LTS
 - Still thinking Fuji earliest candidate for LTS
- Folding security services into edgex-go
 - Sounds like this is approved and we will be doing this
 - Understand that there will be work to take the existing services and refactor them to meet the internal patterns within edgex-go
 - Discussed possibility of improper route formats, do we fix as part of the move or not?
 - TBD this as we get into analysis
 - DevOps
 - Will cause the edgex-go build pipeline to take longer
 - Will affect unit test numbers
 - Legacy repos/verify/merge jobs no longer needed (on master)
 - What is the timing?
 - Security services have been branched for Edinburgh
 - How does this affect our versioning
 - o If Edinburgh is LTS, how does this affect security services?
 - Images for security services will be published with same name, different version from edgex-go
 - Security-client
 - https://github.com/edgexfoundry-holding/go-mod-core-security
 - If we're bringing the security services into the edgex-go repo, then should this module actually be part of go-mod-core-contracts?
 - Could easily go here

- https://github.com/edgexfoundry/go-mod-corecontracts/tree/master/clients
- Actually, what I'm hearing is that security is a separate concern like registry or messaging – so it should be akin to go-mod-security
- If there's a security service that proxies Vault and gets moved into edgex-go, then I'm fine calling it go-mod-core-security
- Need to follow up with a Security Working Group agenda item to unpack this so we can name and organize accordingly.
- Application services (Lenny)
 - New services coming do they each need to have their own repo or should it be in a app-services mono repo?
 - Mono repo can't be versioned each service independently.