Core Working Group Agenda (30-May-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        - *Malini's feedback was that English docs weren't hard to consume but that content was outdated.*
    - *Michael Hall*
    - *Slack ??*

New Business

- Review High-level Fuji deliverables
    - No sense of priority /sequencing on these yet, for the most part
- Deliverables list
    - Move to Go 1.12 (possible 1.13)
        - Change service Dockerfiles
        - Update Snap scripts
    - No planned feature changes to Export services at all for this release
        - Bug fixes if necessary
        - Targeting deprecation after Edinburgh support ends
            - When is this BTW? Is Edinburgh LTS?
    - Core / supporting services
        - Upgrade Consul to version 1.5
        - Complete Device / DeviceProfile rework
            - https://github.com/edgexfoundry/edgex-go/projects/10
        - Value descriptor creation only in core (out of DS)
            - Remove responsibility from device service for creating value descriptors
            - Refine handling/consistency of value descriptors per the following issue
                - https://github.com/edgexfoundry/edgex-go/issues/1071
        - Change event/readings to use nanoseconds in all timestamp fields

  - Device provisioning support (blacklists, etc)
    - https://github.com/edgexfoundry/edgex-go/issues/826
    - Need more definition from the Device Service team as to what is required in the core services for this effort.
- Provide watchers / callbacks for device metadata changes (stretch)
  - Have yet to obtain concrete definition of need
  - This won’t happen unless it’s well scoped and we have an actual use case.
- Move API documentation from RAML to Swagger
- Metrics on Service Performance
  - Need definition of metrics to be captured from QA/Test so desired counters can be implemented
- Improve unit test coverage
  - Codecov.io stats are being published now
  - Trevor working on a proposed pattern for refactoring existing functionality (and new features going forward) to make logic more testable.
- Rules Engine
  - Still no firm direction
  - Would someone like to prototype JSONLogic or Lua scripting?
- Question – Should security services be folded into edgex-go?
  - No viewpoint yet
  - Haven’t quantified whether there are code reuse/organizational benefits
  - Positives
    - Would ease Jenkins mgmt. (fewer jobs, branches, etc)
    - Could leverage code coverage metrics in edgex-go
  - Negatives
    - Mixed ownership – PRs and code overlap

- Things we will not be focusing on
  - High availability
  - Plugins
    - This restricts our ability to embrace other providers within EdgeX
    - Our hands are tied due to lack of support from Go Core Language Team