

Core Working Group Agenda (18-Apr-2019)

Attendees:

Old Business

- Call for Fuji F2F topics (*Reminder*)
 - Currently proposed
 - High Availability
 - 12 factor apps
 - Service Orchestration
 - Registration/deregistration
 - When and when not to use SMA
 - Instrumentation
 - We have correlation ID tracking in our logs, but what's next?
 - Reporting capabilities as determined by deployment footprint
 - Architecture Principles
 - Explicit discussion of architectural guidelines to move toward
 - Assembly of resources to support guidelines and educate new team members
 - Persistence implementations
 - Decouple from mono-repo? How to do this?
 - *Added topic RE: data lifetimes (core-data vs. core-metadata)*
 - Store and forward capabilities??
 - Collecting events at the edge, connectivity is lost. When restored, push everything stored in the meantime.
 - *Agreed to change correlation logging level from Info to Trace for the time being*
 - *PR #1250 should be merged by the time of this meeting*
- Core-command PUTs
 - *OperatingState / AdminState*
 - *Add issues to convert these endpoints to correct REST PUT requests.*
 - *Project Kanban board <https://github.com/edgexfoundry/edgex-go/projects/9>*

New Business

- CBOR
 - *Review input from Anthony on ingestion from core-data RE: no event ID (issue #1216)*
 - *See design docs from last year on concurrent approach*
 - *Will app-functions be updating the timestamp in the same manner as export-distro?*
- Update on removal of DeviceProfile from Device (core-contracts issue #27)
 - Core-contracts issue #68
 - Edgex-go issue #1254
- Core-command part deux
 - When device profile is added, we save the associated CoreCommands in a separate collection
 - See mongo/metadata.go:: AddDeviceProfile()
 - These commands are never updated if the profile changes
 - The commands are not associated directly to their parent device
 - The commands aren't updated when a device profile changes
 - The commands can only be updated by a 3rd party via a direct PUT to core-metadata/api/v1/command
 - What service does this?
 - If we were to address the following
 - Add device linkage to the CoreCommands saved in this separate collection
 - Facilitate their being kept in sync internal to core services when device profile change occurs
 - The we could
 - Allow the core-command service to lookup all commands for a given device
 - This would eliminate the current path to call core-command which calls core-metadata
 - Core-metadata returns the large Device document from which commands must be parsed and returned to caller.
 - Remember the discussion from last week about giving core-command its own DBClient...?