Core Working Group Meeting Notes (19-Dec-2019)

Attendees:

Old Business

- Update on OpenAPI 3.0 (Swagger) docs for Geneva
  - https://github.com/tsconn23/edgex-geneva-api
  - Review period is closed.
  - All docs are sync'ed conceptually
  - Preparing estimate / recommendation for implementation

- - Intent to move content from the above repo into the following
    - Edgex-go PR – docs to be added to edgex-go/api/openapi
    - Edgex-docs/design/ADR PR
      - Add ADR doc to Table of Contents
      - Add doc to design/adr/core
        - Name: *0001*_API_V2_Design_Principles
      - Confirm the following understanding w/r/t naming
        - Number in file name is universally sequential
        - TOC should be referred to see what the current highest value is
        - When adding a new document, use that value +1 in doc name
- Removal of export services
  - Completed
  - We informed the edgex-ui working group (17-Dec). They need to take an action item to revise and repoint to Application Services.

New Business
- Messaging Provider Replacement (ZeroMQ)
  - We have a new resource coming on at Dell and we've assigned this work to him
  - Three new issues (go-mod-messaging repo), each related to a specific provider.
    - Redis Streams #35
    - MQTT #36
    - NATS #37 (proposed – we'll see once we get to this point)
- Two weeks ago question was asked w/r/t re-purposing core-data as "support-data". Follow up on that?
  - *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).*
      - *Clarification: This is because of the device SDK. If the SDK were revised to some other transport or storage, REST Device Service would inherit that.*
      - *Discussion related to the feasibility of REST DS being used as a pass-through (modbus → REST → App Service)*

- *Further analysis required related to persistence aspects of core-data vs querying capabilities.*
  - *If persistence is optional, repurpose core-data as support-data perhaps.*
- Any new business?
  - *Two proposals from Diana A.*
    - *Initialization of Mongo DB for developers*
      - *Two ways to do this*
        - *Init_mongo.js (developer-scripts)*
        - *Go application (docker-edgex-mongo)*
      - *Should we eliminate the javascript?*
        - *Init_mongo used in non-secure environment*
        - *Go app used in non-secure/secure environment*
    - *Should we branch developer-scripts?*
      - *Scripts in the repo only target "master"*
      - *This is because historically the intent of the scripts was to target "master" since the release artifacts are officially docker images.*
      - *If a dev wants to natively troubleshoot a previous release, there's no way to easily get to the script versioned to the earlier release.*
  - *For movement on these*
    - *Trevor to document usage requirements*
    - *Diana to use those requirements for implementation proposal, assuming the elimination of legacy javascript*
      - *In addition, recommendation for cleaning up developer-scripts folder. Can we delete older, un-used scripts?*
    - *Ensure Lisa has visibility*
    - *Will extensions to the docker-edgex-mongo executable align with the CLI capabilities? (Alex)*