Core Working Group Agenda (21-June-2018)

Attendees:

Old Business

- Readiness of export connectors (Janko / Mainflux)
    - Status of ToDo items below from last week
    - Needs more testing / verification – Andy / Keith?
        - Requesting tools / accounts for public cloud vendors to enable testing
        - Approval has been obtained from the Board.
    - Platforms
        - Google Iot
        - Azure Iot
        - Janko to make full list and submit to Keith
            - Send follow-up email to Janko/Keith RE: funding and setup for account creation.
    - Code will ship. May need dot release if testing finds issues.
- Structured log format
    - *I personally haven't had time to follow-up on this (Trevor)*
    - *Post-California*
    - Need to flesh out
        - Eric / Trevor – Document need and criteria (possibly targeting distributed environment)
            - Most likely Eric
        - Tony – Document need and criteria (targeting smaller, constrained environment)
        - How do we implement a solution that works in both of these domains?
        - Rodney to follow up with more info based on known solutions
    - Human readable vs queryable
        - One pager requirements to summarize need
            - Perhaps one page for each
                - System mgmt.
                - Support
    - Suggested improvements
        - Specify log level threshold
            - See go log pkg and their approach to formatting
        - Tony to submit issue RE: local export log generation
            - Also log level threshold

New Business

- California branch cutting
    - Glide.lock will be checked into branch for build dependencies.
    - Version will be incremented to 0.6.0
    - New docker-compose file.
    - Dot release (July)
        - support-notifications (possibly scheduler)
        - Consul upgrade
- Binary Serialization
    - CBOR / Protobuf – decision is TBD
    - Content negotiation for all services…??
    - Abstraction needed for serialization mechanism
    - Requirements discussion
        - Speed of encoding, protobuf is better
        - Schema vs schemaless
            - Schema (protobuf) ramifications for build pipeline
            - Protobuf requires a recompile to be extended, for example security. CBOR has fields for security.
        - Hitachi (Stella Yu)
            - Conversion to binary format from device
            - Send through EdgeX hopefully avoiding cost of serialize/deserialize
            - Export data in binary form
            - Jim: Is encoding the real performance hit or is it a question of REST vs message bus?
            - Hitachi would prefer transport more efficient than REST (advocate of gRPC)
            - Does EdgeX need to understand the data in order to do actuation?
                - Stella: No. Store and forward for now.
            - Performance is important but size of message more so. Base64 encoding in JSON is very large, which in turn slows performance.
            - Favors protobuf due to existing Hitachi eco-system, but nothing against CBOR.
            - Jim asked for numbers around performance/size requirements
                - Stella to follow up with Janine
                - What is maximum size for reading?
            - Reading value is currently a string. If this supported binary, she could embed protobuf representation.

        - Value = binary
        - Valuetype = string (e.g. “protobuf”)
        - Extend device-virtual to execute this path
- Code Quality Pipeline
    - Progress Update w/gometalinter
        - https://github.com/alecthomas/gometalinter
    - Can be run as git-hook
    - Does not support multi-language obviously
        - I’m working toward our reference implementation at the moment
        - Other repos (those who support the other languages) can make their own decisions about tools
        - Paid platforms will get us multi-language support, but may not be a concern right now.
    - Can be integrated with CI
    - Can be run in a dev’s local environment
    - Any follow-up from below item from last week?
        - *Does LF have a recommendation based on other projects?*
            - *Identify key Go-based projects and do a survey (Brett / Jeremy)*
- Dockerfiles creating containers “From Scratch”
    - Empty file system means there’s no way to shell into a running container
        - Shouldn’t this be required from a support perspective?
    - Had a situation with Cloud whereby we needed to get into the edgex-support-logging container to review logs written to file system and couldn’t.
    - Tony
        - Dev environments give shell access but not prod
    - Jim
        - Sees this as a bug
    - If no shell in prod deployment, need some way to extract the logs from the support-logging service via API.
    - What kinds of enhancements does the support-logging service need?