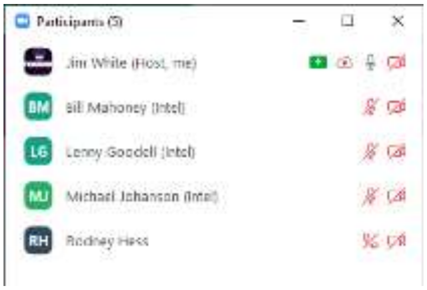
Architect’s meeting, August 20, noon CDT

Attendance:



Participants (5)

BM Bill Mahoney (Intel)

LG Lenny Goodell (Intel)

MJ Michael Johanson (Intel)

RH Rodney Hess

Jim White (Host, me)

Some attendees may have joined after the call started when this record was captured.

Topics

Old

- PR Template for conventional commits is now in place for all repositories for all PRs but without TSC approval. It doesn’t appear to be affecting any problem. We need to finalize the shape of this and officially approve the template by the TSC.
    - Mike, ~~Lisa~~ and Tony to provide improved template around conventional commits and any recommended policy for TSC approval. Meeting still forthcoming.
        - Could add scope in order to understand the domain of a PR (especially for edgex-go)
        - **Mike and Tony to discuss before next TSC.**
    - Stemming from the PR Template conversation was decision to test some automatic dependency checking tools.
        - Mike has been testing Dependabot with app services for some time and it is working well.
        - Decision – request Dependabot use for all repositories to get TSC approval at the next TSC meeting (Aug 26th). **Mike to provide overview for TSC members**.

On Hold Pending Updates

- Design metadata about the “gateway” or host platform (identity, location, …)
    - Being worked in Core WG
    - Need some identifier in Event/Reading
    - Need a unique id for each EdgeX instance
        - New field “Tags” added to Event/Reading
        - Any service could add to that “Tags”
        - Optionally – make available a UUID through Metadata
        - Is there an industry standard – can we align to any existing standard (Jim to research)
            - should the metadata align to a specific industry standard for IOT devices? So as to take advantage of …https://www.project-haystack.org/doc/Structure
        - Tags == Labels

            - Not key-value pair
            - Should be array of k/v (Maps) or JSON object
        - Tag to just to event (not needed to readings)
    - ADR to be created after research
    - Which issue to tackle this in - TBD

New

- Security WG - high priority review of ADR for bootstrapping in OCI containers requested.
    - **Architect's asked to review prior to next Architect's meeting (Sep 21)**
    - Bryon to cover highlights at that meeting
- How do we review/remove artifact (docker images in Docker Hub, snaps, etc.)?
    - New Docker policy will remove any image that is not pulled or pushed in 6 months.
    - This doesn't really effect EdgeX since every container is pulled once based on last few months metrics report. But should we clean up some of these old containers?
    - Discussion:
        - We are exposing ourselves to issues when the (very) old containers have security other vulnerabilities
        - Are we limiting this policy to release artifacts (ADR 0010)? Documents for example?
        - But should we still retain images that are old – like Delhi, Barcelona, etc.? Are we "pulling the rug out from under" someone?
            - We should retain the latest of old images
    - Decision:
        - Allow all images to remain, but tag everything appropriately so that the "latest" dot release will be pulled on request for a specific release. For example, when requesting 1.2 (Geneva), you should get 1.2.1 (not 1.2.0).
        - Update the Docker Compose files so that the latest of the applicable dot release is obtain and not pin it to a specific image. In other words, the Compose file would request 1.2 image and not 1.2.0 or 1.2.1 for Geneva images.
        - Use an X.Y tagging schema (versus release name schema) so that it applies equally well to device services, app services, as well as the core services (out of edgex-go).
        - This will actually help with ContentTrust (DCT) concerns. It will help to enforce newest container will always be pulled.
        - **Lenny and Jim to work for Hanoi release**
    - As for what to do with old (very old) images, let's check what are other projects doing in this case (Kong, Consul, Vault, …)?
        - Also consult with community and adopters; what do they expect from us? Accenture, ThunderSoft, …
        - **Jim to take this research and poll of adopters**
- Extract of Device Service requirements to ADR legacy - what are all the pieces that need to be moved there?
    - Iain has a document out there and believes it is complete
    - **Tony to review by DS meeting on Monday**
- How should we apply semantic versioning to modules? When do we update the minor and major versions of modules? (comes from the Hanoi planning meetings)

    - Decision
    - Release (and tag) them with each EdgeX release (major)
    - Enforce backward compatibility within a major release
    - **Scope this work for Ireland as it will impact DevOps (Jim to get with Ernesto)**
- How do address module and component version release needs for examples (per Slack exchange with Luis Obando). go.mod in the examples helps - or at least some documentation on dependencies.
    - Importantly, how do we let developers know what versions of things go with EdgeX. The version policy and LTS documents may lay it out, but do developers know where to find this.
    - Decision: add something to the READMEs – top level – documents.
        - Add links to "edgex versioning policy" also LTS policy link
        - Do this for the SDKs, App Srv, edgex-go, and (as a stretch goal) to each DS
        - Also add links in getting started guides (in edgex-docs)
        - **Jim to take lead on this work**
- Deferred to next meeting
    - Per the Hanoi planning conference - we need to better define "bound checking" so that a design (and eventual implementation) can be brought forth to meet the requirements
        - Currently considering limiting the number of operations that can be performed on a service (like a device service) over a period of time or setting the max request size (that lends to DoS attacks)
        - Can the solution be more globally applied?

On hold Pending More Info/WG work

- Is order of event/readings being sent by a single device service important? Are there async operations in any service that could change the order of events as they are sent from a DS to core to application services (with REST, 0MQ or MQTT infrastructure)? What do customers desire here? Is maintained order important? What is the current state of the system and can we diagram/document that?
    - Jim to do some research first. Findings: there are places in DS, Core and Application Services where messages can get out of order. If order is something that should be an option built in, it will require much work.
- Incorporation of Vertical Solution WG adopter presentation feedback
    - Jim to collect and present after all 5 presentations
    - Adopters to be invited to F2F
- EdgeX UI - it is for dev/test right now. Would we ever want to have a UI for production? Under what constraints?
    - Being worked through Core WG