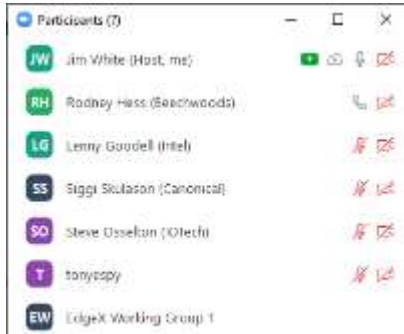


Monthly Architect's Meeting, May 17, 2021, 10am PST

Attendance:



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

High Priority

No High Priority items remain on the list

Medium Priority

- Address how to provide services, UIs, startup scripts/services, etc. with a list of system services.
 - ADR
 - Discussion
 - Some clean-up of the Context is desired. Possibly combine the first two bullets.
 - On the requirements, the requirement about ACID and changes to the list should specify that the services can't get different lists when making a request "at the same time".
 - There is a new requirement that there are actually two lists: potential list of services (for security) and the list of running services (other needs).
 - On requirement #5, get rid of "configuration" in configuration/registry.
 - Jim to clean up the ADR with these comments and those still remaining in the document
 - At a high level, it was decided that:
 - There needs to be a new (or possibly existing) service that runs a seed script. This seeding happens before all other services start. It populates (Consul or a shared volume or shared file) with list of potential services into a global config area
 - Question: where did the original list come from? A TOML file
 - Will this work with snaps when not running Consul (yes?)
 - From that point forward, (Consul or the shared volume, etc.) is the single point of consistency/authority about the list of services running. It can be queried via registry API.
 - That list of services could be added to or removed from via (Consul or the shared volume, etc.) registry API.

- Removed is different from stopped and crashed which are considered temporary states of services.
 - This solution would, it is believed, even work with system that did not run Consul (although it may not be implemented by EdgeX as part of the reference implementation).
 - Could this solution allow us to get rid of client sections in TOMLs? Probably not entirely, but it could at least reduce that configuration by removing the location information (removing a potential conflict and redundant configuration information)
 - Next steps: **Jim to update the ADR and lay out this high-level decision.** Subject to review and conclusion at the upcoming Jakarta planning meeting.
- Standardizing units of measure
 - See J White email providing considerations and options
 - Discussion:
 - No clear definitive standard
 - We should allow any standard (or no standard) to be used with EdgeX values
 - Need to optionally associate a standard to any device resource
 - If the standard provides a means to validate the unit of measure, optionally (configurable to turn it on or off) allow EdgeX to validate the unit of measure.
 - Where the validation physically occurs (at data collection time in the DS, by core data, in the app service?) is still subject to debate.
 - Decision: **Jim to draft an ADR outlining the discussion** and design decision. Some details (like where to do the validation) to be determined.
- The following topics did to not get covered in the meeting and will be pushed to the next meeting.
- How much info should be put into errors and log messages. Temporary decision of 4/22/21 was to provide enough to support debugging. Any additional formalization or rules around this?
- System Management Service future/direction
 - How to deal with dynamically adding/removing services
 - Does it reside in registry/configuration service
 - How to handle security issues
- Declarative Kong applicability
 - Allowing us to drop Progress DB
 - But can you configure groups/users ACL
 - Only supports JWT users

Low Priority

- Where should tools/scripts for migration go
 - After the architect's meeting of Jan 26, 2021, it was decided that "templates" should be created in all SDKs to allow for the easy creation of new services (removing the old samples in the SDKs). The templates will be a means for users

- to copy and create a new service with some instruction on how to rename and replace TODOs with necessary code.
- After the templates are in place, there is a decision to be made about where automation can be placed to use the templates to create new services (versus a manual copy). In the CLI, in a new tool, in a set of simple scripts?
- Documenting Project Decisions
 - Is the Wiki the best place to document project decisions (those outside of or smaller than ADRs). This was our initial take. Should we revisit?
- Revisit where/how/when we might combine service executables (combine services or combine exe in same container?)
 - Release would be easier but image would be bigger with more complex compose files
 - Per Core WG of 2/18/21 - is it at least worth exploring the combination of Core Metadata and Core Command since the two have to share so much data?
 - Core command is just a proxy service today, but reasons for having a separate service include: additional security to protect actuation; issue multiple device commands with one request (make one request and fire it to all Modbus devices or all devices under the control of one service); provide the means to limit requests down to a device so as not to overwhelm it or wake it up). These needs could also be incorporated into a combined metadata service but there are advantages to separation of concerns.
- Time series database support

Tabled for now

- Digital twin (and LWM2M) applicability
 - Being worked via liaison with DTC