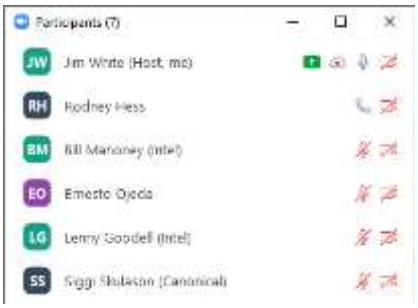
Monthly Architect's Meeting, April 19, 2021, 10am PST

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.
    - SMA needs the list of services
    - UI needs the list of services
    - Secret bootstrapping needs to provide tokens to services
    - Considerations
        - What if Consul is not in use (or should we even allow this)?
        - What if app or device services are added at runtime (post bootstrapping)?
        - What if some services are runtime enabled or disabled (notifications as an example) – especially on a resource constrained host?
        - There are some chicken/egg scenarios; security may need the list in order to provide each service with the proper token before they come up and register with the registry service

- Discussion
  - Perhaps the enable/disable consideration is not important because the security services would need to generate a token regardless of when it is started (via enable/disable)
  - Alternatives
    - A static file that is volume mounted to all services (therefore could be dynamically created at time of initial bootstrap) – or common file that services have access to when not using Docker (Snaps or native services)
      - This would solve the Consul/no-Consul issue
      - This would work for UI as long as that file is accessible to the UI at startup
      - For dev/test – where should it live and/or who creates it? Ans: Core metadata
      - Put in folder that gets mounted as shared volume
      - For production level deployments – override this file to list the services they are going to spin up (i.e. static)
      - The file may be more than just a list of services
        - API proxy would need service key and URI to service
      - Have to be careful of how to update (avoid multiple updates simultaneously) – needs to be an ATOMic operation
      - Have to make sure it does not have errors – creating all services to fail to read/use
      - Have all service (all that need service list) monitor that file
      - Distributed deployment – another issue since the file would not be able to be shared across hosts without using another distributed file system technology
    - Global configuration that everyone reads from Consul
      - But have everyone read a config file if Consul is not there
      - Consul makes it a lot easier and addresses issues – use a file as backup (but then it is static)
      - Ugly issue of chicken or egg – Consul must be secured but its not up to provide list to security bootstrapping.
    - Env vars: list of services that every service gets (list of service keys)
      - Becomes a little unmanageable (when you make a change you have to make changes to lots of services)
  - Regardless of alternative selected, dynamic service list support (add/remove service) creates problems.
  - If we say no, you can't dynamically up/down new services that is very limiting
  -
- Decision:

        - We need ADR (or something) to document options, ideas, issues, consequences (Jim to start)
        - We need to canvas the field of adopters for their opinions on static vs dynamic needs (Jim to address)

Other issues did not get addressed due to lack of time

- Standardizing units of measure
    - Per last meeting: Jim to see if there are other related standards (like the ANSI standard) and see what other IoT/Edge projects are doing. He'll provide more info at the next monthly architect's meeting.
    - See J White email providing considerations and options
- Declarative Kong applicability
    - Allowing us to drop Progress DB
    - But can you configure groups/users ACL
    - Only supports JWT users
- 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

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