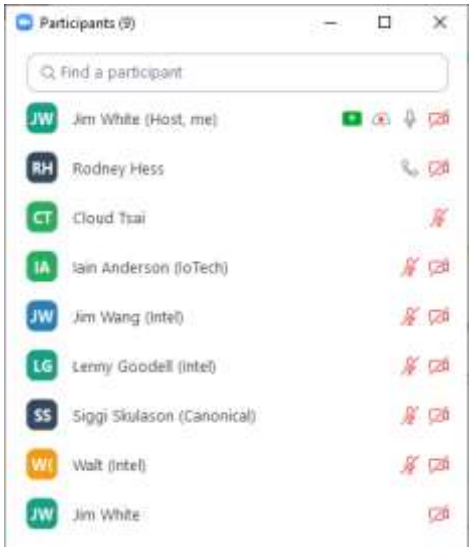
EdgeX Foundry Core Working Group

*(plus system management and UI)*

# 2/18/21

Attendees:




Others may have been in attendance after roll was captured.

## Agenda

Old

- Bugs and issues
- V2 API update
- Ireland Timeline
- Add source issue

New

- Device resource via metadata API
-

## Old Business

Current Issues/Bugs/PRs

- https://docs.google.com/presentation/d/1EZTf1Ap8wX0Bq0jrV0AaFNgpc9LaMjQOwvHtnXGbsEU/edit#slide=id.gbbda15f58d_0_0

V2 API update - update from Cloud

- Completed core command APIs (core services are done with exception of CBOR support - target next week)
- Updated DS SDK Go to use new the core APIs (haven't run through full tests yet)
    - Applying to virtual and modbus DS next in order to test
- Supporting services are WIP (notifications/scheduling)

EdgeX Foundry Core Working Group

*(plus system management and UI)*

Ireland Timeline

- Most V2 API done by April (- sys mgmt)
    - CBOR work
    - TAF
    - Clients
- System management
- Integration / use of V2 APIs and any issues - May & June
    - Consumption points
        - Core data to metadata
        - Metadata to notifications
        - Scheduler to core data
- Documentation - May/June
- What's left???
    - Message bus - haven't made many steps yet in this area (DS to AS with Core tag on); may not be a heavy lift
    - What is the primary implementation for message bus (TBD - TSC)
        - Redis (kuiper doesn't have the password)
            - Doesn't require broker but in secure mode the password issue
        - MQTT
            - Secured MQTTS a stretch for Ireland

Other Ireland work

- Should we add 'source' to the Event DTO and Model last week?
    - Discussed the idea of adding "source" on event per Lenny suggestion. This could be useful to understand, at the Event level, what led to the event - but at the cost of another field (which might be redundant to the resource name when just the resource is requested).
      Decision was made to wait for Tony/Cloud/Iain input next week.
    - Clarification - add SourceName and not Source
    - IOTech is adding CommandName to V1 for customer need - so this seems to be in line with this direction
    - **Decision**: Yes - add to Event
    - AutoEvent has Resource and this would be changed to SourceName as well.
    - In coreCommand section of the device profile - we have a "name" field but should also a sourceName.
        - Allow "name" to be arbitrary and have "sourceName" that identifies which "deviceCommand" or "deviceResource" is being referred to. When there is both a deviceCommand and a deviceResource, the deviceCommand takes precedence over deviceResource of the same name.
        - Make name optional. When not provided, it will default to sourceName.
        - The sourceName would be required.

  - There was further discussion around whether all the redundancy in Event and Reading is a good thing. Could we remove device profile and device name fields in reading (and just use those in Event).
  - Issue was whether anyone would need to query Readings and need the profile or device name when they just pulled Readings independently.
  - Are there other places where Readings become divorced from Events and we need to consider the consequences of duplicating fields in each? If so, would it be better just to have Readings have a reference to its "owning" Event?.
  - **This issue will be taken up in the monthly architect's meeting.**
- Allow DS to have switch to use CBOR for everything or JSON sent through the bus (or REST channel) - on hold as a stretch goal for Ireland. Lenny and Jim to work out what test for this might look like.
- Consul update to 1.8.1. New lib approval needed - https://github.com/hashicorp/go-hclog
  - Attribution PR to merge once lib approved; fixes attribution check issue
- Board cleaned up - closed a lot of already completed issues

Kubernetes - nothing new to report
UI - nothing new to report
CLI - nothing new to report

## New Business

- Per Monthly Architect's meeting on Tuesday
  - As we know the data we need resides in the `DeviceResource` within `DeviceProfile` . More specifically it is the `PropertyValue` within the `DeviceResource` that is needed. Both the App Services and Kuiper Rules Engine need this information. Neither need to know anything about a DeviceProfile. They only need to know about `properties` of a `DeviceResource` for which a `Reading` was received.
  - Rather than forcing both App Services and Kuiper to have knowledge of DeviceProfiles, where to find the resource properties within and code to process them, I am proposing a new simple Core Metadata end point `/deviceresource/properties/{profileName}/{resourceName}` which returns the `PropertyValue` DTO for the specified `resourceName` from the profile specified by `profileName`. This keeps the knowledge in Core Metadata of how and where `Resource Properties` are stored within a `DeviceProfile`. Then App Services and Kuiper Rules Engine only have knowledge to the data they actually need. Both will cache the data rather than looking it up each time.
  - As we discussed in todays' meeting, there is no need to invalidate the cache as only adds of device resources will be allowed. The properties for any newly added resources will be looked up the first time a `Reading` is received that uses them.

- OR
- Just get/cache the profiles and dig out the device resources internal to the app services
- Additionally core command does this today with regard to core commands and profiles. Should there at least be symmetry between these types of requests?
- As another option, we could put the convenience methods in the client (allowing the app service the means to get what it wants but don't add it to metadata directly)?
  - This option was considered but rejected because the client is Go-only and would require any other language such as C to have ot implement it on its own.
- **Decision**: create a convenience method on metadata for getting device resources. Specifically provide a GET device resource method (other PUT, POST, DELETE methods are not needed at this time and would be added only when a use case is defined) that has the following structure:
  - /deviceresource/profile/{profileName}/resource/{resourceName}
- It was also decided that we may want to offer similar convenience for commands for the command service to use. But there was a discussion about whether it was just for core commands or should include device commands. This decision was deferred to the next meeting to allow everyone to explore the needs before making a decision. **[Jim to add this to the agenda for the next meeting.**
- **Decision**: As a related but independent part of the discussion, it was also decided that PropertyValue be changed to ResourceProperties in the device profile. Also, the structure of the profile should be changed to remove the intermediate "values".
- Finally, due to all of the issues about what Metadata should offer with regard to device resources/commands/etc, it re-raised the issue about whether Core Metadata and Core Command should be combined. This is a topic on the architect's monthly list. **[Jim will elevate it in the discussion list.]**

- Board updates