

Core Working Group Agenda (16-May-2019)

Attendees:

Old Business

- CBOR
 - Progress Update
- Redis
 - Depends on discussion during DevOps call immediately preceding.

New Business

- Core-Command
 - The purpose of this service is to execute device commands via GET/PUT
 - Look at the service client in core-contracts and you'll see these are the only two capabilities
 - <https://github.com/edgexfoundry/go-mod-core-contracts/blob/master/clients/command/client.go>
 - The source of record for device/command data is core-metadata, so that is where all writable operations for these entities should be located.
 - [Proposal] to remove endpoints facilitating write operations on commands and/or devices in core-command
 - PUT /device/{id} (Used to set admin/operating State on device)
 - Already provided by the following metadata routes
 - PUT /device/{id}/adminstate/{adminState}
 - PUT /device/{id}/opstate/{opState}
 - PUT /device/name/{name} (Used to set admin/operating State on device)
 - Already provided by the following metadata routes
 - PUT /device/name/{name}/adminstate/{adminState}
 - PUT /device/name/{name}/opstate/{opState}
 -
 - PUT /device/{id} (same as above, totally redundant)
- Core-Metadata
 - As discussed previously, we are moving toward a strategy whereby commands and their relationship to a device is managed through the DeviceProfile

- See edgex-go project for more information
 - <https://github.com/edgexfoundry/edgex-go/projects/10>
- The DeviceProfile will be a template from which commands will be assigned to a device when a new device is provisioned.
- Any change to a device's commands should not be made on an individual device, but rather via the DeviceProfile. These changes will then need to cascade to all associated devices.
- [Proposal] to remove endpoints from core-metadata that facilitate writable operations to the device/command relationship
 - POST & PUT /command (adds or updates a command)
 - DELETE /command/id/{id} (deletes a command by its ID)
 - These endpoints today have no real effect because whenever we return a list of commands for a device, those commands come from the DeviceProfile and not the Commands collection.
 - Our understanding is that it is impractical to have one device out of a set that deviates from the device profile that it was provisioned under. Therefore the device profile should serve as the template for all capabilities that can be performed by a given device.
- If we are not comfortable with removing these, then I suggest for Edinburgh we return 501 – Not Implemented and document this in the RAML.
- Fuji Core Deliverables review
 - Stabilization cycle, although I'm sure stuff will come up
 - Discussed increased unit test coverage. This will require some refactoring.
 - Device auto-provisioning support (such as device blacklists). Awaiting more defined requirements.
 - Value Descriptor sync with Device Profile
 - Eliminate Device Service from Device contract type
 - <https://github.com/edgexfoundry/go-mod-core-contracts/issues/27>
 - Similar to Device Profile work above
 - Focus on the edge. No HA focus until Geneva.
- Security / Vault Integration if time permits