Architect’s meeting, January 26, 2021, 10am MST

Attendance:

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

Topics

Completed business

- Holding repository clean up
- Semantic versioning with modules – (decision was to tag ‘em with the release number that they go with.) ADR updated and Lenny/Ernesto worked to complete this
- Docker naming changes – being worked by DevOps

In Work Group for review right now

- Core/High - Ensure that service location data is pulled from trusted source (i.e. not Consul) (Tony's ADR)
  - Covered in Security WG

High priority (priorities per previous meetings)

- We need to revisit CBOR binary support vs simple JSON / binary encoding (per Core WG meeting of 1/7/21). How to handle binary data in V2
  - Is CBOR still the right way?
  - Simplicity versus performance
  - We should have new requirements/use cases to change this
  - Jim to find the objectors to CBOR before we cover and get any suggestions/requirements for non-CBOR (there are none)
  - Lenny – votes for leaving as is for V2 (use of CBOR as done for V1)
    - App services will unmarshal the CBOR and deal with the event (and does not re-encode in CBOR)
  - Message bus vs REST – not germane to the discussion of CBOR v JSON bin encoding
  - Heart of the discussion simplicity of JSON bin encoding vs complicity of CBOR but less efficient than CBOR
    - JSON encoders take care of all the back and forth marshalling/unmarshalling
Iain did some performance testing (in C device services)
- CBOR v Json encoding
- On 1MB binary data
- 33% add to size of the payload (JSON) v. 1 for 1 size on CBOR (1MB)
- CBOR -.5 millisecond per event; JSON 20 milliseconds
  - Could maybe do some optimization to improve this
  - Base 64 encode step is already a millisecond

**Decision: for Ireland – no change**
- For Jakarta (or stretch for Ireland) – allow DS to have switch to use CBOR for everything or JSON sent through the bus (or REST channel)
- Perhaps we need some testing in Go SDK and DS; even full workflow test to see if there are significant gains across
- Could we run this through Modbus scalability tests to see results

- Client keys in configuration; use the service keys as the constants for the map keys (per Core WG meeting of 1/21/21)
  - https://github.com/edgexfoundry/edgex-go/issues/2424
  - Decision:
    - Use service keys in configuration.toml files
    - Service keys in core-contracts/clients/constants.go
    - Both issues above – should be addressed by this decision
    - Add to agenda in Core WG to make sure everyone is aware/ add to cross cutting concerns because this hits everyone.

- V2 API - should we add security foundation added to that (per some of earlier V2 API designs via Dell and Bryon N)?
  - Adding token to authenticate a micro service call (is this in scope for Ireland)
  - May not be needed unless all services are distributed
  - We need to explore alternatives to provide secure / locked out service to service communications
  - Not a core competency of EdgeX?
  - Need more input from Bryon and Security

- Use of CLI to generate new application or device service
  - Per Core WG and TSC meetings (1/20/21), there is a big desire to add a feature to EdgeX tooling to more easily generate a new device or application service (using a command line tool and some template found in the SDK)
  - Question is whether this should live in edgex-cli or is this a separate tool?
  - CLI for replacing curl scripts; this doesn’t see to fall in line with objectives of the current CLI.
  - Should be “templates” in examples for DS and AS
    - The copy for the starter of a new service
  - Tool would then pull down the template, rename things, adjust the mod file, on a copy
  - But where does this tool belong? In an SDK, CLI or separate tool.
  - **Decision: work template first; with some documentation and possible script second – then look at tool automation after that.**
Medium Priority

- Address how to get device resource info (for app services and Kuiper)
  - Probably not ADR worthy
  - Either provide Lenny's convenience APIs or tool to dig out the device resource information in the (cached) profiles
  - How/when to invalidate the cache if we use the profile-digging approach
- Keep commit history from beginning to end (don't squash them until PR approved)
- Standardizing units of measure
- Declarative Kong applicability
  - Allowing us to drop Progress DB
  - But can you configure groups/users ACL
  - Only supports JWT users

Low Priority

- 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?
- (must be done before V2 is done) - Naming scheme changes for config.Clients (key name change)
  - Use consistent name that all other services use for core data
  - Consistency in the naming vs changing all the names to use service name as part of key
  - Related to system management hard coded list of services.
  - Separate issue in arch meeting – high once report back
  - Other naming issues (secret store vs secret service)
  - Opportunity to make all config/naming consistent
  - Jim take resp – get WG leads – try to prioritize this survey
- Revisit combine core services at least at all executables in one image
  - Release would be easier but image would be bigger with more complex compose files
- Digital twin (and LWM2M) applicability
- Time series database support and applicability
  - Ian Johnson has an example of app service to InfluxDB export (snap in the store)