

What would Certification Process of Device Service Look Like?

Q's

- Is this for new or existing (replacement) device service? Is it commercial vs. open source?
 - We would assume the process remains the same but there may be some points of departure. See below.
- Are the tests only testing what is common to the DS SDK? Are there other things that are to be tested – like how data is deposited to Core Data?
 - We would assume that it does include how it impacts and behaves with other services, but this may come in phases (crawl, walk, run).
- Do we need to modify the current DS SDK black box tests to be more parameter driven – especially with regard to Command services?
 - Yes, again, over time.
- How do we test for and explore “unusual behavior” – like how a DS might send data to Export or something like Security that we did not expect (or maybe want)?
 - We won't be able to do much at first (other than manual inspection) but over time this is something we'll want to add more tests for.

Assumptions

- It is assumed that the device service tests would be the same for all device services. Tests would not be unique to a device service.
- We are certifying to only a specific platform at first.
- Guidelines on documentation review that is subjective need to be written to make this more comfortable to submitters; making it less subjective.

Process

1. Self-assessment is done by the submitter.
2. Device Service submitter sends the Certification WG the following:
 - a. device service (containerized),
 - b. device (or simulator) used to test the device service
 - c. device profile(s) – and any documentation necessary to explain the profile(s)
 - d. documentation as per below.
 - i. Specifically, the protocol details should be included in the documentation – for example in a Modbus device service, the submitter would indicate whether using Modbus RTU vs TCP/IP
 - e. self-assessment results/report
 - f. Guidance on how much data is going to be captured by default
3. Perform black box tests that we already have (API check)
 - a. DS SDK black box/API tests
 - b. Device Service specific tests (for device services that we already have – if any are written, which to date they are not)
4. Inspect or test that the service behaves well with regard to rest of EdgeX (some of this could be done via running existing other black box tests; most will have to be done by manual inspection today)

- a. Appropriately populates Core Data with readings/events
 - b. Appropriately populates Value Descriptors
 - c. Appropriately populates metadata (Devices, Device Services, Profiles, ...)
 - d. Appropriately receives and interacts with Command service (responding as expected)
 - e. Appropriately gets its configuration through the config/registry service
 - f. Uses the Logging micro service to log errors, warnings, information and debug statements.
5. Perform “stability” exercise and report results. Leave the device service running for a certain extended period of time (time period to be determine) and examine:
 - a. How much event/reading data did it report (did it seem to match expectations provided by the submitter?)
 - b. How long does the service stay up – did it run out of resources or otherwise crash?
 - c. Did it impact other services negatively (like causing any of them to fail due to data volume or size issues)?
6. Performance evaluation and metric collection
 - a. Report on CPU, Memory and other resource usage
 - b. Report on time to come up
 - c. Report on storage usage
7. Documentation review (subjective review)
 - a. Information on any operations, configuration, additional functionality, etc. that differ from the reference implementation (example – providing configuration defaults and options that differ from the reference implementation service).
8. After each phase of above, any failure, test observation or concern can be addressed with the submitter – allowing the submitter to address issues and re-iterate the entire process or a section as required
 - a. Are the failures in line with submitters self-assessment documentation and exceptions noted? Are these reasonable and does EdgeX need to revamp tests to accommodate or at least grant an exception on the basis of what is discussed.
 - b. Cert WG privately reviews and makes a determination of the results and failures – allowing the submitter re-iterate through the process