System Management WG Meeting: 4/16/19

Attendees: Jim, Akram, Trevor (Dell), Doug Migliori (ControlBeam), Iain (IoTech), Yongli (Microsoft), Rodney (Beechwoods), Vlado (TU Berlin), Michael (LF), Lenny (Intel). Attendees that may have joined after the start of the meeting may not have been captured and listed.

Discussion and action items as a result of meeting in RED

Old Business

- Documentation work ongoing
- System Management Metrics – keys and metrics (Go v C)
  - Akram and Iain discussion
  - Metrics (and Keys) will be different in Go versus C services around memory and CPU.
    - No issues from the community on this.
- Responsibility Discussion
  - Decisions per last meeting
    - Registry/Config service is optional (as it is today). The registry flag determines whether it is in place or not (and to use it when it is).
    - The registry/config interface/contract (done by module) could be implemented by any number of providers.
      - Providers: Consul, SMA, file system could be providers of this interface; but it is still optional.
    - SMA should implement the simple registry service.
    - Registration order of precedence:
      - 1st – Consul
        - It would know what services are up or not
      - 2nd – SMA (thus the reason for it to implement the registry contract)
        - If Consul is not there, the SMA would understand what services are up or not
      - 3rd – if Consul is not running, and SMA is not running, the services are own their own and would have no central place to know what is running (registry flag would be off)
    - ON HOLD use case –the SMA is up but Consul (registration/config services) is not. Can't think of a use case that meets this need.
      - Perhaps if you are looking at running as few of services as possible, you would eliminate Consul, but not SMA. Is this the use case that requires the order above? SMA could be more attractive than running Consul with a simple/lightweight registry.
    - Deregistration of the service in now a need (all services must deregistration when they go down in graceful manner).
      - In the case of a crash and the SMA is serving as the registry, the SMA should try to restart crashed services after configurable timeout.
      - Use the ping interval and a configurable number of retries to determine when to try to restart.

        - For each request of what services are running, the SMA must go to the registry service for what is up and running (no caching). In the case that it is the registry service, it goes local – and it may want to persist the state of the services.
        - SMA is never the bootstrapper – in other words the first thing to start EdgeX. It would be something like a script, Docker Compose, Kubernetes, sysd, etc.
            - After the bootstrapping, the SMA/executors can start or stop or restart services
            - The SMA (via executor) could request to restart or stop any service – including the SMA itself.
    - Continuing the discussion
        - Stop and Restart with SMA
            - Stopping the SMA – executor can kill the SMA.
            - The executor would be in charge of returning results to the SMA about the stop/restart processes.
            - Corner cases – does SMA get restarted last or first by executor?

New Business

- Fuji system management scoping
    - In
        - Implementation of SMA as lightweight registry per above
        - Implementation of list of services for SMA per above
        - Implementation of stop/start/restart rules per above
        - Executor used for collection of all metrics
        - Allow for setting of configuration (now that we have read and write config)
            - How to implement – through service?
            - How does this impact the need for restarting of services (if we are using Consul)?
            - This is a tough problem to design/implement. But Gets eventually beget the need for Puts; thus the reason for it on the release roadmap.
        - SMA translation layer (stretch)
            - Should the SMA offer the metrics and config info through some other protocol (LWM2M, etc.?)
            - Should the start/stop and set config be allowed?
        - Suggestion from Iain: SMA to store configuration – lightweight configuration store (in replacement of Consul or whatever provider we have for reg/config).
            - Lenny: Watching for changes would be toughest part of this
            - Iain: Simplifies config updates/puts when it is the config server
            - Trevor: What we are trying to solve is service resiliency – not necessarily providing an alternate to config management.
        - Will bring this last option up at F2F. I am inclined not to do this and certainly not for Fuji, but worthy of other inputs.
    - Out
        - Storing metrics collected locally
        - Callbacks (alert clients of changes to config)

- SMA translation to other protocols beyond 1st
- Actuation based on metric change
  - Rules engine for control plane data
- Consideration to use QoS and blockchain to prioritize resource usage by certain services

- Other opens