System Management WG Meeting: 4/2/19

Attendees: Iain, Tom (IoTech) Jim, Trevor Akram (Dell), Drasko (Mainflux), Rodney (Beechwoods), Vlado (TU Berlin), Ian (Canonical). 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

- DS C SDK and system management (Akram & Iain)
  - Issues for implementation
  - Differences in implementation (metrics)
  - Status: provide endpoints for configuration & metrics
    - Metrics present but differences
    - C doesn’t have garbage collection so not the same memory usage stats
    - What will be provided: current allocation and heap size
    - CPU usage: total time used by the service and the average usage (basic ratio based on actual time) [Go gives system load]
      - "Alloc" (total heap memory in use)
      - "Heap" (current heap size)
      - "CPU" (CPU time consumed in seconds).
      - CpuBusyAvg
        - Is the CPU time in Go for the whole system or just the service? Ex: if some other program fired up on the box, it would affect the numbers.
        - Akram/Iain – to work out the keys
          - Configuration already in place and aligned.
            - Raw JSON payload that is what the config is provided to the service
  - Documentation is to do.
  - Any consideration for generic monitoring from the Docker container? – from Vlado
    - Maybe its worthwhile to check into tools like C-advisor, tools that are modular to borrow/include for internal as well.

New Business

- Responsibilities of service discussion
  - Discussion in line with document
  - Akram’s sequence diagrams for getting metrics or config if we have agreement on responsibilities (did not get to this)
- Fuji system management scoping (did not get to this)
  - Use of registry or local manifest to drive service interaction (per previous discussion)
  - Use of executor for metrics collection
  - Allow for setting configuration (through registry/config service)
  - 1st translation layer