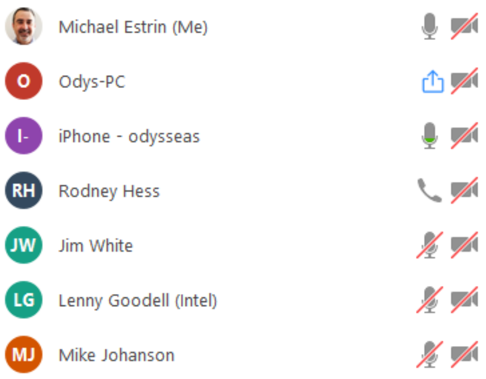
System Management Working Group – December 6, 2019

Attendees



Michael Estrin (Me)

Odys-PC

iPhone - odysseas

Rodney Hess

Jim White

Lenny Goodell (Intel)

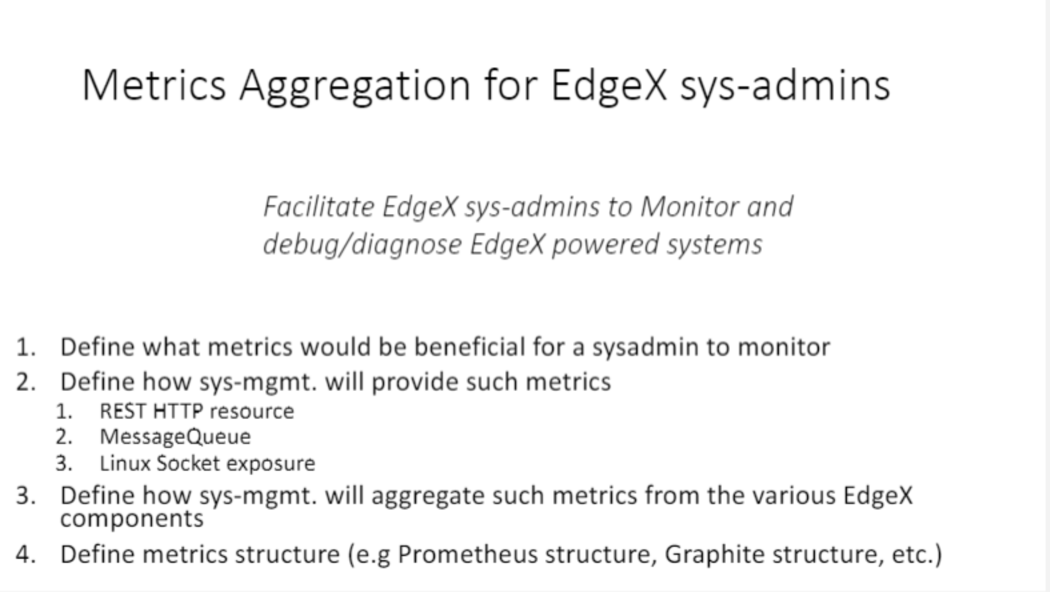
Mike Johanson

Old Business

1. Geneva Deliverables Review
    a. System Management (https://github.com/orgs/edgexfoundry/projects/29)
        i. Quick review of board. Ian's issue pending his response. Moved go-mod-core-contracts issue into committed.
    b. CLI (https://github.com/edgexfoundry-holding/edgex-cli/issues) (Alex, VMware)
        i. Quick review of CLI board; not sure what was added or what status of effort is.
    c. Open Horizon (Joe, IBM)
        i. OH will want one or more repos in holding. Waiting for Joe to give me a list so I can request them. Joe's team is working on use cases which can then be broken down into GitHub issues (within the new holding repositories) which can then be tracked and worked. Expect these to be completed and work to commence around the end of January.
        ii. Question around how OH will manage containers within EdgeX system instance (Rodney).
2. CLI move from edgexfoundry-holding to edgexfoundry -- pushed to next meeting; Alex missing from call.
    a. Progress towards 0.1.0 deliverable (prerequisite to making request)
    b. Review of related discussions in QA and Certification working groups
3. Request to add a system management client in go-mod-core-contracts – we've committed to doing this for Geneva today.
    a. Issue created (https://github.com/edgexfoundry/edgex-go/issues/2173)

New Business

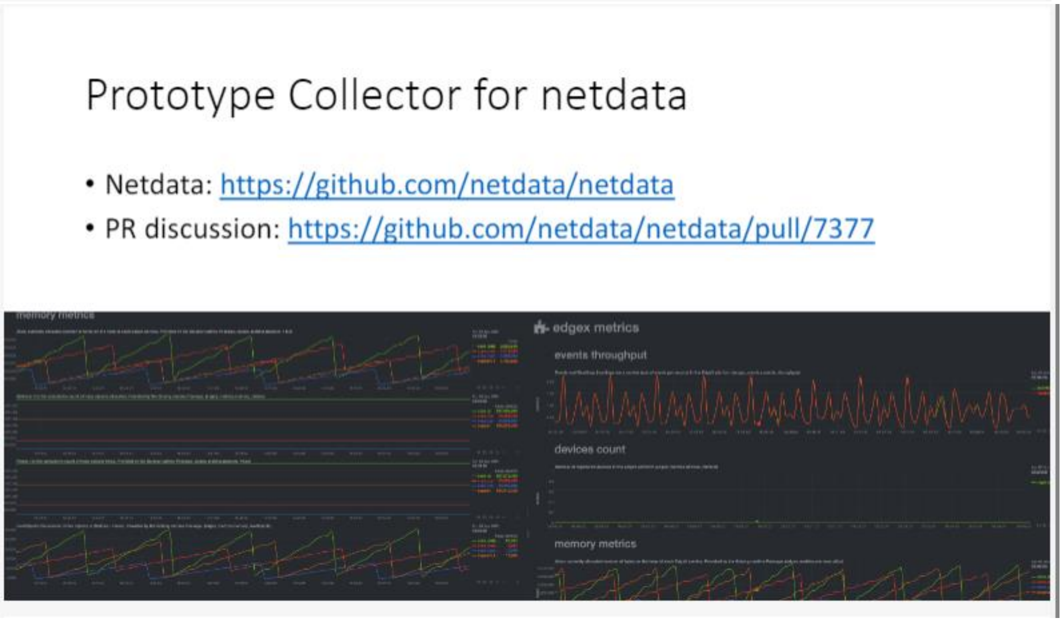
1. Support for additional system-wide metrics (Odysseas Lamtzidis)
   a. Presentation:



Metrics Aggregation for EdgeX sys-admins

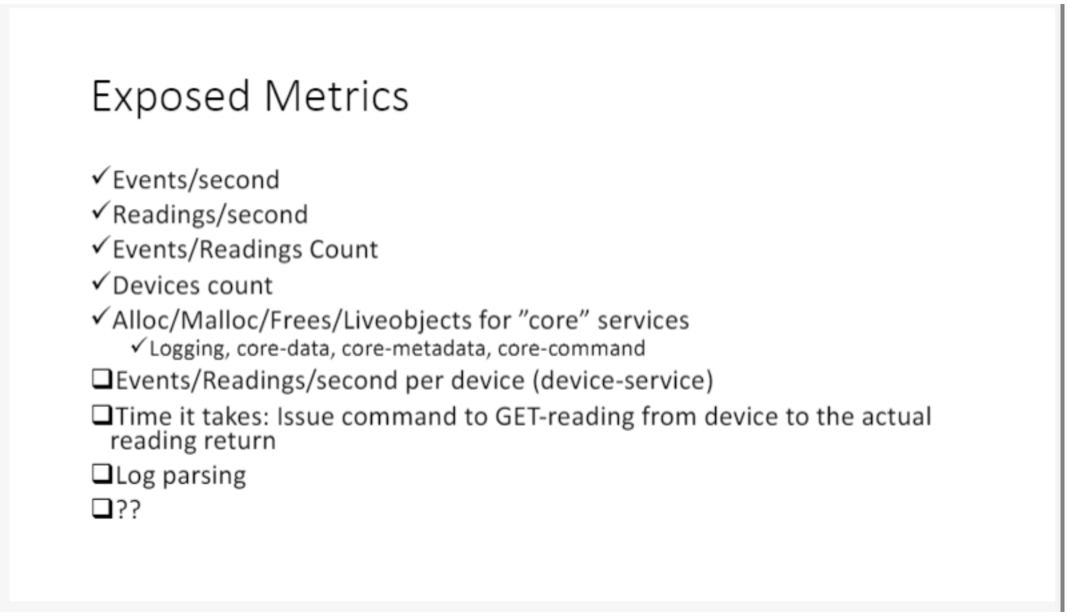
Facilitate EdgeX sys-admins to Monitor and debug/diagnose EdgeX powered systems

1. Define what metrics would be beneficial for a sysadmin to monitor
2. Define how sys-mgmt. will provide such metrics
   1. REST HTTP resource
   2. MessageQueue
   3. Linux Socket exposure
3. Define how sys-mgmt. will aggregate such metrics from the various EdgeX components
4. Define metrics structure (e.g Prometheus structure, Graphite structure, etc.)



Prototype Collector for netdata

- Netdata: https://github.com/netdata/netdata
- PR discussion: https://github.com/netdata/netdata/pull/7377



Exposed Metrics

✓Events/second
✓Readings/second
✓Events/Readings Count
✓Devices count
✓Alloc/Malloc/Frees/Liveobjects for ”core” services
  ✓Logging, core-data, core-metadata, core-command
□Events/Readings/second per device (device-service)
□Time it takes: Issue command to GET-reading from device to the actual reading return
□Log parsing
□??

b. He implemented a prototype collector for netdata.

c. Current system management agent metric endpoint doesn’t return metrics in a timely fashion; i.e. takes more than a second to return. Talking directly to an individual service for metrics is quicker.

d. What EdgeX-specific metrics would be useful?

    i. How much time does it take for a request to be processed?

e. Why NetData? Why not Graphite and Prometheus? (Rodney) NetData has a distributed architecture; Prometheus has exporters on each machine that push data to a central Prometheus server. We should be able to use any implementation.

f. What metrics would be useful for admins? (Rodney)

g. Metrics that system management agent provides shouldn’t include those that aren’t EdgeX-specific (i.e. memory and cpu usage) and which can be gathered using other tools/technologies. (Lenny).

h. What EdgeX-specific metrics would be useful for the system management API to support?

    i. Latency statistics – from device sensor up north and back (and vice-versa).
    ii. Throughput of events.
    iii. Indication of health – that events are being processed
    iv. Validation failures
    v. Data/events not transiting the system; being blocked somewhere and queueing up
    vi. The volume of data each edge system is producing northbound.
    vii. Security related; intrusion detection for example.

i. Leads to “instrumentation of the code” (Lenny)

j. Could we mine this from the logs? (Jim) Logs are more batch-oriented.

k. Lenny mentioned Intel’s experience with instrumenting code on prior project – Telegraph, Influx, and Graphana – that provided proactive notification of extraordinary events. Based on prior experience, don’t instrument all the things. Define your metrics first and provide an initial limited view.

l. Concern about service bloat (and introduction of latency related to capturing instrumentation). We don’t want to turn EdgeX into an enterprise-like system. (Jim) Questions around how we could implement this in a *concise and economical* way.

m. Sampling and controlling should be configurable. (Mike)

n. Leveraging local rules engine to potentially capture metrics and action based on out-of-bounds conditions; i.e. runaway device whose that can be filtered to avoid overwhelming northbound (and potentially running up a large cloud provider bill). (Jim)

o. Suggestion to check in with the Security Group. (Jim) Michael will do so.

2. Other new business?