System Management Working Group Meeting Notes – 4/17/18

* Dell Technologies resource being added 5/7; will provide bandwidth to help start sys management work.
* We need to take the current scoping/requirements document and use it to drive out a design first. All agreed and all felt it would be better to take the scope document and extend it to be a design doc.
* Given the current scope document, some topics for consideration/discussion as we prepare for design…
	+ On the sys management agent’s request to stop each micro service, would happened in Consul? Would the agent deregister each micro service? Would the micro service deregister itself? Would the availability check eventually just determine the service no longer available? Consensus answer: each micro service should deregister itself on a call to stop.
	+ When order is important on start, the agent is dependent on the 3rd party “plugin” built to some abstraction to make sure order is maintained/handled. That 3rd party plugin could be Docker-Compose interface, Snappy interface, command line script interface, etc. The call to the 3rd party plugin could always be a no-op depending on implementation under the covers.
	+ Would a micro service need to restart itself? No, the service agent would orchestrate that with some management plugin (Docker, Snappy, wrapped OS script, etc.). The service should not be in charge of restarting itself.
	+ We need to think about and document the various start/stop use cases. For example, would more than one 3rd party management system need to start/stop EdgeX micro services. Multiple may need to get sys management data/metrics from the sys management agent, but would multiple need to send down potentially conflicting commands (ex: Pulse IoT Center and Msft Azure IoT Hub both trying to start or stop the services at the same time).
	+ The micro service API to address start, stops, restarts is actually simple (stop only) as the other calls would be handled by the agent.
	+ Would there be one agent or multiple (one for each 3rd party sys management connected)? Opinion is that there would probably only be one and use a model like the export-services in EdgeX to get sys management data to multiple destinations.
	+ Who comes up first, registry (Consul) or system management agent? Consensus was registry.
	+ Would micro services return answers to metric/alert callbacks directly or always go through the system management agent? Consensus: respond always back to the agent and then requester.
	+ Would the micro service ever need to push data or would it always and only respond to pull requests from the agent. Consensus: why not do both.
	+ Given always responding (push or pull) to agent, what if the micro service tries to send or respond to agent and the agent is down or has moved? Consensus: agent should never move, but if it was down, then the system as a whole is toast and we need to think about system wide remediation (ex: reboot, etc.).
* Salim stepping down from chair in June. Will have to look at leadership options.