

System Management WG Meeting: 6/25/19

Attendees: Akram, Michael, Trevor (Dell), Rodney, Dave (Beechwoods), Joe (IBM), Doug (ControlBEAM), 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

- Open Horizons sub-project
 - Next meeting again next week July 1st.
- Fuji Work
 - Set Configuration – updated design diagrams and implementation status by Akram A. (copies in Wiki)
- Design review callouts as follows:
 - Do we anticipate any “interactions” with the work underway with the Golang-Reflection-based updates underway by Michael, to Config? → i.e. PR #1461 (in progress) “*Orthogonal concerns (Override Configuration Using Environment Variables)*”.
 - Point an environment variable to an alternate address; SMA activity (in the area of making updates to Config) will come *afterwards* (i.e. in follow-up fashion) and update the values that were seeded earlier on. Note: The Golang-Reflection-based updates underway will not be modifying the *container* contents, though.
 - What about updates to Registry setting for, say... “Port”? (Registry as the source of truth?). Making changes to Registry is, essentially, changing the underlying functionality... → What conflicts are possible, especially if dealing with Non-writable setting, which necessitate the restart of services?
 - Modify this step: SMA calls to *service* which calls to registry (to instead be *directly* from SMA to Registry).
 - Regarding current understanding of *WatchForChanges()*, see the following function `listenForChanges()`
 - Akram to follow up with Trevor (Re Implementation, and also sequencing of some steps.)
 - Does Config update apply to a given service only for the *duration* of the service’s uptime. Or does it apply for the entire *lifetime* of service?
 - Codify the knowledge of whether we’re dealing with a Writeable or non-Writeable setting (to update).
 - Not using SMA to update? Instead, lean on Consul?
 - Update each service? No.
 - Show-and-tell Demo by Akram (Done).
- Metric collection by Executor – design
 - Reviewed the design.
 - No review callouts were made.
- To be addressed as part of this work
- Metric collection by Executor – working
- Start/stop/restart all done by the executor to include stop/restart of SMA
 - Executor tracks completion and returns results
 - Deferred to reviewing in the next WG meeting.
- Safe shutdown (per last meeting)
 - Let the service decide what is part of “shutdown” operation. Each service should provide a “shutdown” function – that does the exit, but service can override and

- have a way to prevent immediate shutdown or wait for something to finish before it is shutdown. Need to work out the details of what is returned from this method.
- Make a note of the call to the function name *Destruct()*, in each service, which is responsible for clean-up activity.
 - Causing a service to stop executing; what are the associated side-effects? (See “main.go”). Releasing resource back to the system. Other use cases that would leave system in an indeterminate state? (None that the community is aware of).
- *WatchForChanges()*. It’s a blocking function - something would need to shut it down at the end.
 - No need to have the SMA call the function *WatchForChanges()* on each service. Evidently, each service (in its “init.go”) calls that function (as part of another function, named *listenForConfigChanges()*) through a Registry client, like so: *registryClient.WatchForChanges(registryUpdates, registryErrors, &WritableInfo{}, internal.WritableKey)*
 - Make a note, too, of *updateChannel ()*, for writable settings (only).
 - SMA Translation layer (stretch)
 - Pick one protocol to start (LWM2M)
 - Future release options
 - Handling multiple key sets
 - Caching of endpoints and key/value configs
 - Noted and annotated in sequence diagram (named “(Jun-25) External Entity Requests SMA to Set Configuration”)

New Business

- None