

## Schedule / Schedule and “Polling Event” Breakup

Documented cross working group decision – 12/3/2018

Today, Schedule and Schedule Events are elements of Metadata. Schedules define a point or range of time such as a start or stop time or CRON string for something to happen. Schedules themselves do not do anything nor do they refer to anything directly such as an operation. They just define a schedule or some time or time range for something desired to happen (either in a single instance or on a reoccurring basis). Schedule Events link a Schedule (per above, a time definition) to some action on some service – such as clean up of the database in the Core Data service. Schedule Events are what drive some schedule engine to do work at the scheduled time (that is at the time specified in the Schedule in the Schedule Event).

The Schedule/Schedule Event domain items are managed by CRUD APIs on Metadata and the data is stored in the Metadata document store. They are then used by:

- a) The Scheduler micro service to trigger actions across other services (notably cleanup of data in Core Data today)
- b) The Device Services use them to trigger sensor reading of the devices that the Device Services manage.

It should be noted for the reader that both Device Services and Scheduler micro service have an engine which use these Schedule/Schedule events. This is because the Device Services must operate their polling operations with little or no latency. Device Services cannot always afford a remote REST call from Scheduler to trigger the collection of data. Further, because of loss of connectivity, a Device Service cannot rely on the Scheduler’s remote call to trigger sensor data collection (or other sensor related activity like keeping a sensor awake). The Scheduler service operates across all of EdgeX and can trigger any action on any service (to include Device Services) but in a RESTful/non-deterministic way.

It was decided during today’s Device Service working group call that these two concepts/uses of Schedule/Schedule Events were not in alignment.

Device Services need something that is simpler to drive Device Services request for new data from the sensors. For lack of better terms – this is a *reading poll schedule*. It is possible that the polling time could be dynamic – allowing for Device Services to throttle forward or pull back on the number of readings collected based on sensed changed, available resources, etc. How these new reading poll schedules will be stored in Metadata (example: as part of a device profile, as separate domain concepts associated to devices, etc.) and what is the required API around them is to be determined. The Device Services WG under direction of Steve Osselton will explore the design/implementation of a new “reading pole schedule”. Ideally (and the goal), this work will be done in time for the Edinburgh release, but will depend on other refactoring efforts and resource availability.

The Scheduler service will continue to use Schedule/Schedule Event as defined today. However, the CRUD API and database (document store) will be moved from Metadata micro service to the Scheduler micro service – in order to provide better separation of concerns and allow the Scheduler service to have independence (loose coupling) to Metadata. Core Working Group (Trevor Conn with Eric Cotter)

will begin work to make this separation. Some changes to the Schedule/Schedule Event may also occur but in context of the EdgeX scheduler needs.

At a time when the Device Service has implemented the reading poll schedule and no longer needs the Schedule/Schedule Event in Metadata, and when these domain concepts and APIs are implemented in the current and existing Scheduler service, the APIs, data initialization, and code associated to Schedule / Schedule Event will be removed from Metadata.