Information Model for LMAP

draft-ietf-lmap-information-model-02
and proposed changes for 03

IETF 91, Honolulu, November 2014

Trevor Burbridge, BT

Motivation

- Overall Purpose
  - Guide standardisation of one or more control and reporting protocols
  - Enable high-level interoperability between protocols
  - Clarify MA information and functionality
- Structure
  - Assist protocols in breaking information down into separate messages that can be delivered at different times over different protocols
  - Aid readability

Information Model Sections

Pre-Configuration	Minimal set of information necessary for an MA to securely contact an initial Controller
Configuration	Information configured by the Controller pertaining to Controller communication or general MA settings such as MA and Group ID
Instruction	Configuration by the Controller of what Measurement Tasks to perform, when to perform them, and where/when to report the results
Logging	Information transmitted back to the Controller with configuration or instruction errors and general failure notices
Capability & Status	Information available to be fetched by the Controller such as the Measurement Tasks supported by the MA or interface configuration
Reporting	Information sent to the Collector regarding the Measurement Task results including MA context and Task Configuration

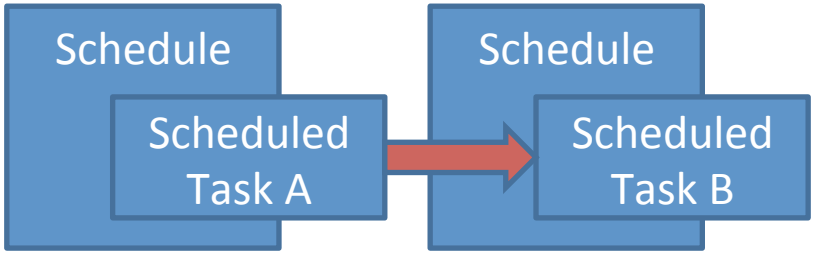
Recap: Changes from 01 to 02

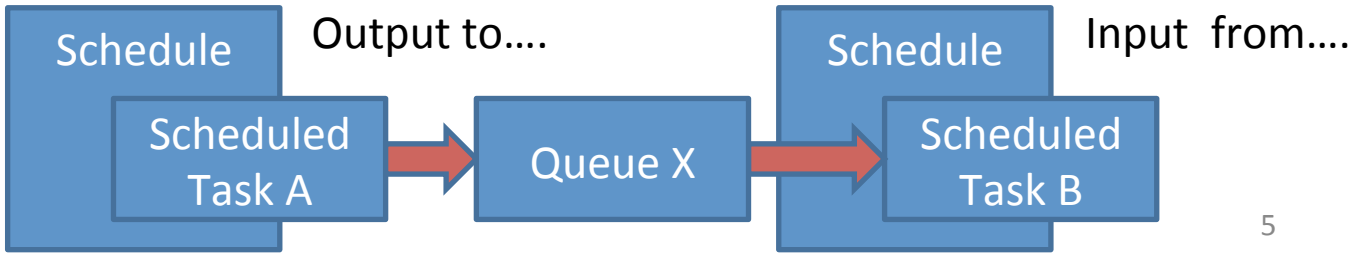
- Various small corrections and improvements
- Role added to Task Configuration
- Removed separate Pre-Configuration Information (became subset of Configuration)
- Tried to restructure task 'datasets' to separate 'Channels' and 'Downstream Tasks' to aid clarity
  - But clearly failed!

On to proposed changes discussed during/since Dublin…..

Channels & Downstream Tasks

- Agreed Channel becomes Task Configuration Option
- Terminology: 'Destination Task' instead of 'Downstream Task'
- Correction: Scheduled Tasks need to output to other Scheduled Tasks (and not just Task Configurations)
- Discussed: Explicit concept of 'Data Queue' – Rejected but explain possible queue implementation in draft
  - As aid to clarity and allow data handling parameters in future





Task Configurations vs. Scheduled Tasks

- Accepted: Allow parameters to be set specifically for a Scheduled Task as well as within Task Configuration
  - Allows general Task Configuration to be specialised within a Schedule
  - Flexibility and reduced data transfer
  - No arbitrary split
  - Need to agree:
    - Union of parameters within Task Configuration and Scheduled Task
      - Alternative is to over-ride
    - Behaviour is test dependent (e.g. take first, last or both or throw error)

Generalisation of Task Configuration and Report Information

- Task role is “just another” task parameter
  - As is Channel
  - Can leave them out of the Information Model as they are defined per Task
- Implies registry reference will be specific to role if different roles have different parameters
  - Follow whatever the Registry work comes up with
- Cross-traffic is “just another” task reporting column
  - Leaves format to individual tasks (can select which counters etc.)

Reporting

- Inclusion of Task Configuration and Schedule information is dependent on the Reporting Task (Configuration)
- Reporting, Management and other Tasks also need a registry (private, local)
- What happens to data if Reporting Task is suppressed is not defined
  - Let it build up or throw away. Tail/head drop. Memory limit etc.

Timing

- Agreed to remove ISO8601 compliance for datetimes (just leave RFC 3339)
- Agreed to remove calendar defaults but allow wildcard '*'
  - Otherwise could default to every second
- Rejected need for negative enumerations and day from end-of-month
  - E.g. test all days except Sat and Sun, test on last day of the month
- Agreed 'startup' timing executes on MA restart
  - Not on loading a new schedule – have immediate timing support if required
- Agreed what happens to two schedules start at exactly same time?
  - run in parallel

Persistence

- Data persistence is device dependent (not protocol dependent)
  - Controller can check through ‘Last Instruction” status
- Agreed to re-instate separate Pre-Configuration Information
  - Since device will revert to pre-configuration on factory reset (and clear Configuration and Instruction)

Other

- Agreed to keep JSON example
  - Currently useful but maybe superseded by forthcoming protocol documents
- Need to standardise condition codes?
  - Hierarchy would be useful
  - Follow/reference other industry examples – what are they?
- Correct Framework draft reference to Normative
- + all the other straightforward suggestions and corrections from the 02 review