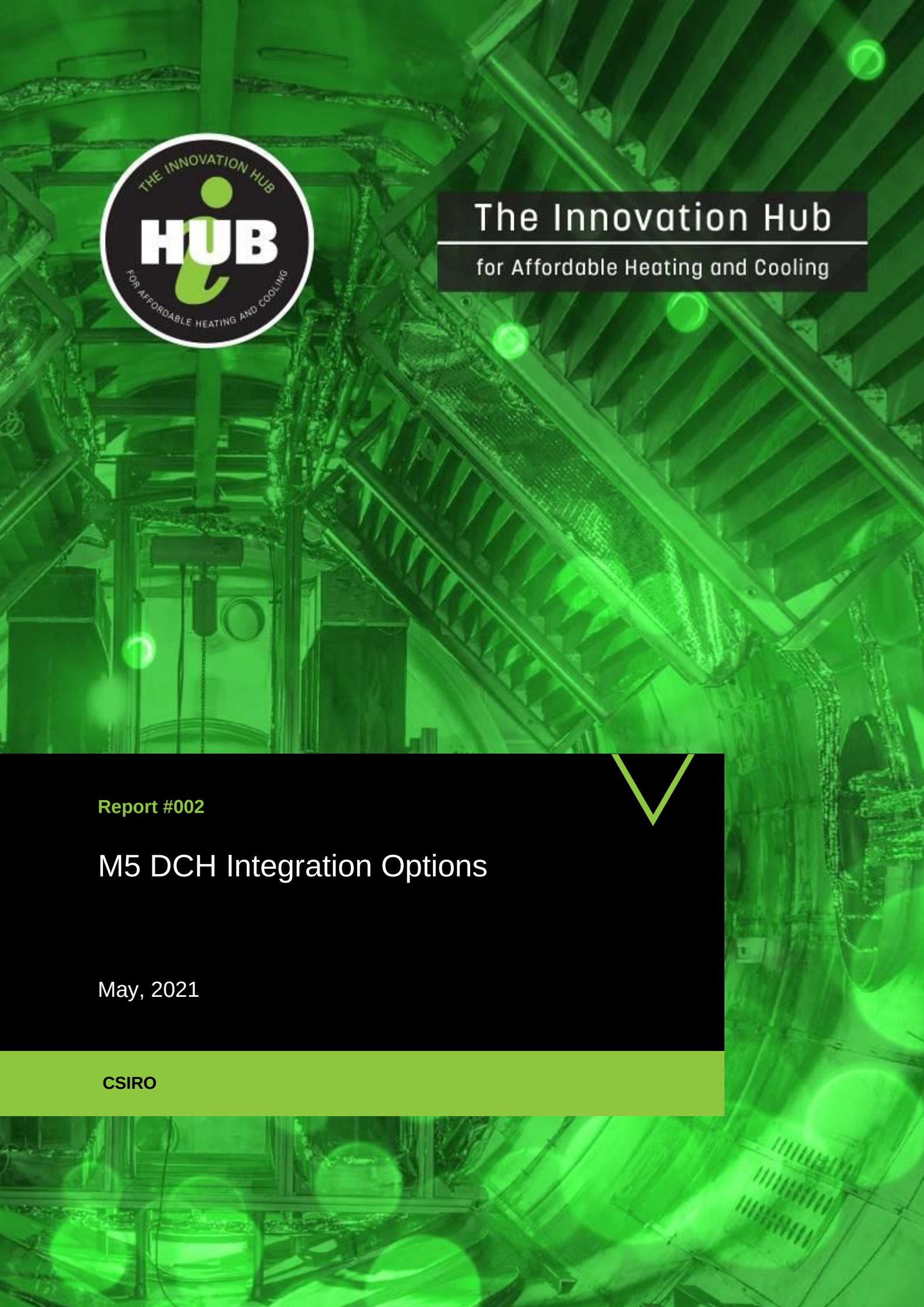


The Innovation Hub

for Affordable Heating and Cooling

Report #002

M5 DCH Integration Options

May, 2021

CSIRO

About i-Hub

The Innovation Hub for Affordable Heating and Cooling (i-Hub) is an initiative led by the Australian Institute of Refrigeration, Air Conditioning and Heating (AIRAH) in conjunction with CSIRO, Queensland University of Technology (QUT), the University of Melbourne and the University of Wollongong and supported by Australian Renewable Energy Agency (ARENA) to facilitate the heating, ventilation, air conditioning and refrigeration (HVAC&R) industry's transition to a low emissions future, stimulate jobs growth, and showcase HVAC&R innovation in buildings.

The objective of i-Hub is to support the broader HVAC&R industry with knowledge dissemination, skills-development and capacity-building. By facilitating a collaborative approach to innovation, i-Hub brings together leading universities, researchers, consultants, building owners and equipment manufacturers to create a connected research and development community in Australia.

This Project received funding from ARENA as part of ARENA's Advancing Renewables Program.
The views expressed herein are not necessarily the views of the Australian Government, and the Australian Government does not accept responsibility for any information or advice contained herein.

Primary Project Partner

Project title: CSIRO Senaps data platform demonstration and development of the Data Clearing House

This project will oversee development of the Data Clearing House (DCH), a cloud based building data management and application enablement platform. The DCH connects Internet of Things (IoT) systems from buildings and supports complex data analytics. The DCH will underpin the development of applications that improve renewable energy integration in buildings and unlock new opportunities for delivering Buildings to Grid (B2G) services.

This project will investigate features of the CSIRO Senaps data platform and their suitability for the DCH. It will combine these findings with results from the DCH2 Switch data platform subproject to develop the Data Clearing House.

Lead organisation
CSIRO

Project commencement date
July, 2019

Completion date
June 2022

Date published
19 May 2021

Contact name
Will Borghei

Email
Will.borghei@csiro.au

Project website

1 Overview

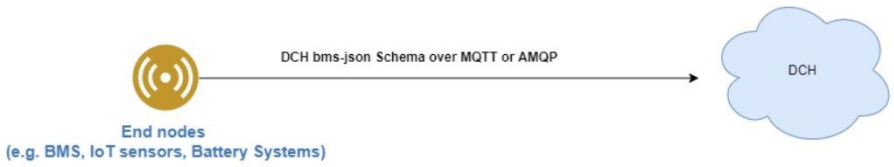
Data Clearing House (DCH) supports ingestion of data from heterogeneous sources. Currently DCH provides three options for communicating with buildings. This document provides all necessary information to assist you with integrating your on-site hardware with DCH.

2 Interfacing Options

2.1 Direct Communication over MQTT and AMQP

2.1.1 BMS-JSON SCHEMA (MQTT OR AMQP)

This is the recommended method to have bidirectional communications between DCH and the end-nodes. If your device is capable of supporting custom JSON packets (to use the DCH bms-json schema) over MQTT or AMQP, your integration would be with relative ease. Your device may be a Building Management System (BMS), sensors, Battery Management System or an already available gateway device.



Information about bms-json schema can be found in Appendices of this document.

In order to connect with the MQTT broker, the following information is required:

- **MQTT Hostname:** senaps.io
- **MQTT Username:** Set by the client during DCH Onboarding Process
- **MQTT Password:** Set by the client during DCH Onboarding Process
- **Port:** 8883
- **MQTT Topic Structure:** Refer to appendices (provided by DCH)
- **MQTT payload Schema:** Refer to appendices (provided by DCH)
- **MQTT Broker TLS:** User Login Over SSL (CA Signed Certificates), version 1.2 preferred

In order to connect with DCH over AMQP, the following information is required:

- **AMQP Hostname:** senaps.io
- **AMQP Username:** DCH Onboarding Process (set by client)
- **AMQP Password:** DCH Onboarding Process (set by client)
- **Port:** 5671
- **AMQP payload Schema:** Refer to appendices (provided by DCH)
- **AMQP TLS:** User Login Over SSL (CA Signed Certificates), version 1.2 preferred

2.1.2 DIRECT COMMUNICATION OVER HTTP: REST API

Information about the DCH REST API can be found at https://senaps.io/api-docs/.

REST API can also be used for 3rd party app developers to interface with DCH. At the moment, DCH team can provide machine accounts and keys to utilize REST API. If you plan to use the REST API, please contact DCH to discuss further steps.

2.1.3 DIRECT COMMUNICATION OVER FTP: CSV FILES

This will be an assisted service from the DCH team for integration. DCH can use CSIRO's secure SFTP servers to receive CSV files. Information required to send data to DCH over FTP are as follows:

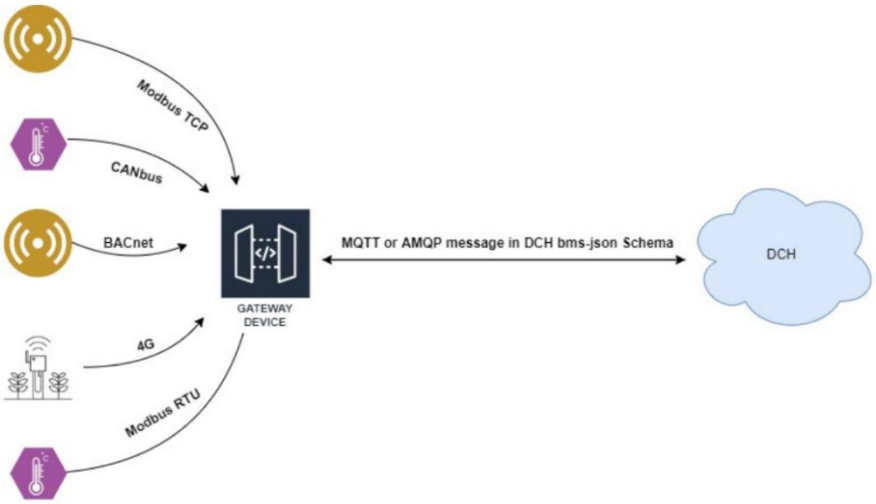
- **Server:** pftp.csiro.au
- **Port:** 22
- **Username:** Provided by DCH
- **Password:** Provided by DCH
- **Folder to use:** Provided by DCH

Additionally, DCH could configure the Data Source to collect files from a user's FTP server. This process consists of creating a FTP account for the client and setting up a pipeline to ingest the received data from the FTP server to DCH.

2.2 Communication through Gateways

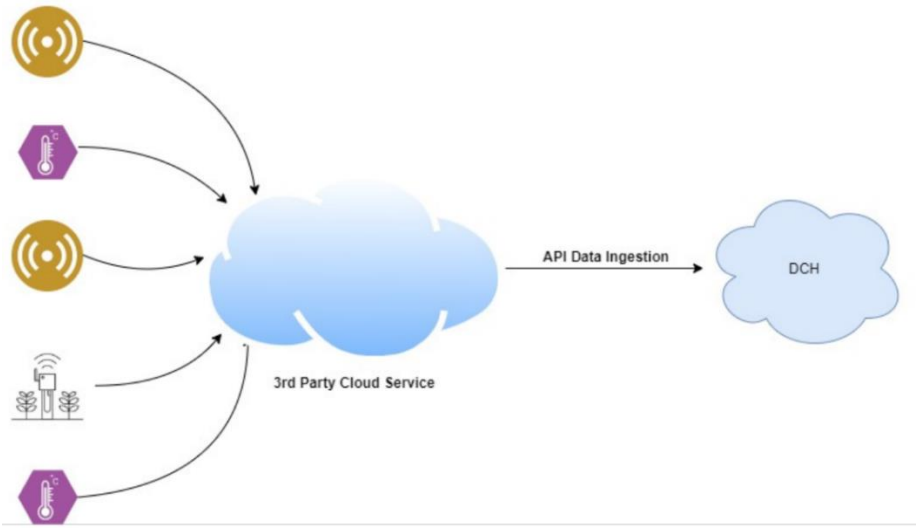
If your devices do not support the above options (MQTT, AMQP or HTTP), you may require a gateway to translate the data to the mentioned protocols. Common gateways such as Tridium JACE and devices that can run Node-Red, can translate the data to MQTT. Documentations for configuring some of these devices to communicate with DCH are available. Your devices may be Modbus RTU, Modbus TCP, BACnet, 4G, LAN, CANbus etc. and you need to ensure that your gateway is able to translate those packets and send the data using bms-json schema over MQTT or AMQP.

Configuration manuals for some gateway devices are provided by DCH.



2.3 Communication through 3rd Party Cloud Data Ingestion with API

If your data is already available on a 3rd party cloud server and the provider offers API communications, DCH could ingest the data directly.



To do this, the 3rd party needs to provide DCH with credentials to establish two-way communications between DCH and the end-nodes. Be aware, if the provider does not support two-way API communications, DCH can only READ the data on the cloud and will not be able to WRITE to the devices. Some platforms such as LoRaWAN TTN, Yurika and Wattwatchers have already been integrated with DCH now and further 3rd party services will be integrated in the near future. Third party data useful for energy applications (e.g. AEMO NEM public data, BoM 30-minute weather observations) are also available in DCH.

3 Appendices

Appendix A: bms-json schema valid topics

MQTT Topic formatting:

The MQTT topic will be structured as follows:

dch/<siteName>/<buildingName>/<gatewayType>/<gatewayName>

siteName: Site name as defined in the json schema.

buildingName: Building name as defined in the json schema.

gatewayType: Must be one of 'tridium', 'edgex', 'easyio', 'mango', 'nexcom', 'ewon', 'generic'

gatewayName: as defined in the json schema.

IMPORTANT: Each field described above may use URL legal characters: alphanumerics, dash, underscore, but NOT periods.

This is because the DCH message broker uses periods as its topic separator at this time.

Examples:

dch/Newcastle/NewcastleEnergyCentreB001/tridium/DCH_2470Newcastle_JACE01

dch/Newcastle/NewcastleEnergyCentreB001/tridium/EMS_2470Newcastle_EMSJ01

dch/BlackMountain/SynergyB801/tridium/DCH_1020BlackMountain_JACE01

dch/Clayton/ClaytonB307/tridium/DCH_6040Clayton_JACE0

Appendix B: minimal bms-json schema example

```
{
  "$schema": "http://csiro.au/dch/bms-json/schema-draft-06.json",
  "point": {
    "pointName": "CS.EC.OFF.AHU 02.HW VALVE",
    "currentValue": 0,
    "parentName": "AHU02"
  }
}
```

Appendix C: full bms-json schema example


```
{
  "definitions": {},
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "http://csiro.au/dch/bms-json/schema-draft-06.json",
  "type": "object",
  "title": "DCH Building Management System JSON schema.",
  "required": [
    "point"
  ],
  "properties": {
    "gatewaySoftwareVersion": {
      "$id": "#/properties/gatewaySoftwareVersion",
      "type": "string",
      "title": "The BMS Gateway software version name",
      "default": "",
      "pattern": "^[a-zA-Z0-9\\-\\_\\.]+$$"
    },
    "gatewayModel": {
      "$id": "#/properties/gatewayModel",
      "type": "string",
      "title": "The BMS Gateway manufacturer and model",
      "default": "",
      "examples": [
        "Tridium Jace"
      ],
      "pattern": "^[a-zA-Z0-9\\-\\_\\. ]+$"
    },
    "gatewayName": {
      "$id": "#/properties/gatewayName",
      "type": "string",
      "title": "The BMS Gateway name",
      "default": "",
      "examples": [
        "DCH_2470Newcastle_JACE01"
      ],
      "pattern": "^[a-zA-Z0-9\\-\\_\\.]+$"
    },
    "siteName": {
      "$id": "#/properties/siteName",
      "type": "string",
      "title": "The Site name",
      "default": "",
      "examples": [
        "NewcastleEnergyCenter"
      ],
      "pattern": "^[a-zA-Z0-9\\-\\_\\.]+$"
    },
    "buildingName": {
      "$id": "#/properties/buildingName",
      "type": "string",
      "title": "The Building name",
      "default": "",
      "examples": [
        "NewcastleEnergyCenterB001"
      ],
      "pattern": "^[a-zA-Z0-9\\-\\_\\.]+$"
    },
    "point": {
      "$id": "#/properties/point",
      "type": "object",
      "title": "The Point Schema",
      "additionalProperties": false,
      "required": [
        "pointName",
        "currentValue",
        "parentName"
      ],
      "properties": {
        "pointName": {
          "$id": "#/properties/point/properties/pointName",
          "type": "string",
          "title": "The Point name",
          "default": "",
          "examples": [
            "CS.EC.OFF.AHU06.POWER"
          ],
          "pattern": "^[\\x20-\\x7e]+$"
        },
```

```
"timestamp": {
  "$id": "#/properties/point/properties/timestamp",
  "type": "string",
  "format": "date-time",
  "title": "The Timestamp in format defined by RFC 3339 Section 5.6 ",
  "default": "",
  "examples": [
    "2020-02-26T13:03:18.998+11:00",
    "2020-02-26T13:03:18.998Z"
  ]
},
"currentValue": {
  "$id": "#/properties/point/properties/currentValue",
  "type": [
    "string",
    "number",
    "boolean"
  ],
  "title": "The Current value of the point",
  "default": 0.0,
  "examples": [
    1203.3746,
    "AUTO",
    true
  ]
},
"status": {
  "$id": "#/properties/point/properties/status",
  "type": "string",
  "title": "The Status Schema",
  "default": "",
  "examples": [
    "{ok}"
  ],
  "pattern": "^[a-zA-Z0-9\\-\\_\\{\\}\\(\\), @]+$"
},
"ord": {
  "$id": "#/properties/point/properties/ord",
  "type": "string",
  "title": "The Ord Schema",
  "default": "",
  "examples": [
    "station:|h:af00"
  ],
  "pattern": "^[a-zA-Z0-9\\-\\_\\{\\}\\(\\),:\\|]+$"
},
"parentName": {
  "$id": "#/properties/point/properties/parentName",
  "type": "string",
  "title": "The Parent Name",
  "default": "",
  "examples": [
    "AHU06"
  ],
  "pattern": "^[\\x20-\\x7e]+$"
},
"facets": {
  "$id": "#/properties/point/properties/facets",
  "type": "object",
  "title": "The Facets Schema",
  "patternProperties": {
    "^[a-zA-Z0-9\\-\\_\\.]+$": {
      "type": [
        "string",
        "number"
      ]
    }
  },
  "additionalProperties": true
},
"bacnetProperties": {
  "$id": "#/properties/point/properties/bacnetProperties",
  "type": "object",
  "title": "BacNET properties associated with the Point",
  "patternProperties": {
    "^[a-zA-Z0-9\\-\\_\\.]+$": {
      "type": [
        "string",
        "number",
        "boolean"
      ]
    }
```

```
      },
      "additionalProperties": false
    },
    "modbusProperties": {
      "$id": "#/properties/point/properties/modbusProperties",
      "type": "object",
      "title": "MODBUS properties associated with the Point",
      "patternProperties": {
        "^[a-zA-Z0-9\\-\\_\\.]+$": {
          "type": [
            "string",
            "number",
            "boolean"
          ]
        }
      },
      "additionalProperties": false
    },
    "proxyExtProperties": {
      "$id": "#/properties/point/properties/proxyExtProperties",
      "type": "object",
      "title": "Properties derived from Tridium protocol drivers",
      "patternProperties": {
        "^[a-zA-Z0-9\\-\\_\\.]+$": {
          "type": [
            "string",
            "number",
            "boolean"
          ]
        }
      },
      "additionalProperties": false
    },
    "tags": {
      "$id": "#/properties/point/properties/tags",
      "type": "object",
      "title": "key-value tags associated with the Point",
      "patternProperties": {
        "^[a-z]+:[a-zA-Z0-9\\-\\_\\.]+$": {
          "type": [
            "string",
            "number",
            "boolean"
          ]
        }
      },
      "additionalProperties": false
    }
  }
 }
}
```

4 Acknowledgements

The author acknowledges the following project team members who have contributed to this report: Akram Hameed, Mark Peristy.