

Open Geospatial Consortium

Publication Date: 2016-08-20

Approval Date: 2016-08-20

Posted Date: 2016-08-30

Reference number of this document: OGC-16-093

Reference URL for this document: <http://www.opengis.net/doc/PER/iot-protocol-mapping>

Category: Draft Engineering Report

Editor(s): Steve Liang (SensorUp / University of Calgary) and Tania Khalafbeigi (SensorUp / University of Calgary)

Incident Management Information Sharing Internet of Things Protocol Mapping Engineering Report

Copyright © 2016 Open Geospatial Consortium.
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>

The research in this presentation was conducted under contract with the U.S. Department of Homeland Security (DHS) Science and Technology Directorate (S&T), contract # HSHQDC-13-C-00119. The opinions contained herein are those of the contractors and do not necessarily reflect those of DHS S&T.

Warning

This document is not an OGC Standard and may not be referred to as an OGC Standard. This document presents a discussion of technology issues considered in an

Document type:	OGC® Engineering Report
Document subtype:	NA
Document stage:	Approved for public release
Document language:	English

initiative of the OGC Interoperability Program. The content is subject to change without notice. The content of this document does not represent an official position of the OGC.

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished receives the Intellectual Property subject to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Contents Page

Table of Contents

Open Geospatial Consortium.....	i
Incident Management Information Sharing Internet of Things Protocol Mapping Engineering Report	i
Abstract	v
Business Value.....	v
Incident Management Information Sharing (IMIS) Internet of Things Sensor Hub Protocol Mapping Engineering Report	1
1 Introduction.....	1
1.1 Scope	1
1.2 Document contributor contact points	1
1.3 Revision history.....	2
1.4 Future work	2
1.5 Foreword	2
2 References.....	2
OGC 06-121r3, <i>OGC® Web Services Common Standard</i>	2
3 Terms and definitions	3
4 Conventions	3
4.1 Abbreviated terms	3
4.2 UML notation	4
5 An IoT Reference Model	4
5.1 IoT Protocol Layers and Segments	4
5.2 Where Does the Protocol Mapping Happen?	7
6 IMIS IoT Practices of Protocol Mapping.....	7
6.1 SensorUp Smart Shirt (Hexoskin).....	8
6.1.1 Data Model.....	8
6.1.2 Sequence Diagram	8
6.1.3 Example Request/Response	9
6.2 SensorUp mBed SDK - Temperature Sensor (ZigBee to Ethernet).....	10
6.2.1 Data Model.....	10
6.2.2 Sequence Diagram	11
6.2.3 Example Request/Response	11
6.3 SensorUp Linkit ONE Software Development Kit - Environmental Sensor ...	12
6.3.1 Data Model.....	13
6.3.2 Sequence Diagram	13
6.3.3 Example Request/Response	13
6.4 SensorUp SensorThings Wearable Cam	14

6.4.1	Data Model.....	15
6.4.2	Sequence Diagram	15
6.4.3	Example Request/Response	15
6.5	SensorUp SensorTag SDK	17
6.5.1	Data Model.....	17
6.5.2	Sequence Diagram	18
6.5.3	Example Request/Response	18
6.6	Open Sensor Hub (OSH)-enabled Unmanned Aerial System (UAS)	19
7	Summary and Conclusions	20

Abstract

The IMIS Internet of Things (IoT) Pilot established the following objectives:

- Apply OGC principles and practices for collaborative development to existing standards and technology in order to prototype an Internet of Things (IoT) approach to sensor use for incident management.
- Employ an agile methodology for collaborative development of system designs, specifications, software and hardware components of an IoT-inspired IMIS sensor capability
- Develop a distributed computing architecture and design practices to integrate OGC Sensor Web Enablement (SWE) standards with IoT principles and technologies to improve both the agility and consistency of sensor networks.
- Document the Pilot achievements and lessons learned in engineering reports and demonstrate them in a realistic incident management scenario

This engineering report details Pilot experiences in connecting a variety of local communications protocols and message formats supported by low-cost sensor devices with OGC SWE Web services published globally over IP networks. It describes the Sensor Hub approach taken to support these connections and the mappings from one protocol to another required to develop integrated SWE-IoT networks.

The Incident Management Information Sharing Internet of Things (IoT) Pilot Protocol Mapping Engineering Report established the following objectives:

Business Value

The IMIS IoT Pilot aimed to develop, test and demonstrate the use of networked sensor technologies in a realistic emergency incident management scenario developed in collaboration with DHS S&T sponsors and first responder stakeholders. The Pilot demonstrated an Internet of Things (IoT) and standards-based approach to sensor use for incident management in order to establish a roadmap for industry adoption of sensor device and information sharing products that both work together and vigorously compete to provide first responders with new capabilities.

Prototype capabilities introducing new opportunities for industry and government business value included ad hoc, nearly automatic local sensor deployment combined with global discovery and access to sensor information, alerts, and visualizations. They also included location tracking and georeferencing of sensor observations as well as publication in common formats for wide use in computer aided dispatch, emergency operations center, and geographic information systems, as well as on a range of mobile devices. The scalability of the capabilities developed in the Pilot provide opportunities for industry to develop a wide range of products that work together across platforms and technologies. They also provide opportunities for each public safety organization to procure, deploy, and maintain valuable capabilities that are sized appropriately for their size, role, and budget.

Incident Management Information Sharing Internet of Things Protocol Mapping Engineering Report

Incident Management Information Sharing (IMIS) Internet of Things Sensor Hub Protocol Mapping Engineering Report

1 Introduction

1.1 Scope

This engineering report describes various mechanisms implemented in Sensor Hub (S-Hub) components to mediate between heterogeneous Internet of Things (IoT) sensing devices and Open Geospatial Consortium (OGC) standards-based information services such as SensorThings API (STA) and Sensor Observation Services (SOS). The report describes IoT sensor and S-Hub implementations for the OGC Incident Management Information Sharing (IMIS) Internet of Things (IoT) Pilot project (<http://www.opengeospatial.org/projects/initiatives/imisiot>). The Pilot Architecture Engineering Report (OGC 16-014r2) presents an overall description of the project and a general architecture; the Pilot Profile Recommendations Engineering Report (OGC 15-118) describes the adaptation of OGC services to IoT sensor deployment. This document focuses on the interactions between IoT sensors and OGC sensor web services (*i.e.*, STA or SOS). The interactions consist of mappings or conversions from the local protocols used by IoT sensor devices to global OGC Web services and vice versa. In this engineering report, the term ‘protocol mapping’ represents these interactions, often carried out by specific Sensor Hub components (S-Hubs). The protocol mapping can be further divided into two layers: (1) the interfaces and communication protocols and (2) the messages exchanged between the interfaces on top of those protocols. The implementations of these two layer components are detailed for various sensors deployed as part of the IMIS IoT pilot.

1.2 Document contributor contact points

All questions regarding this document should be directed to the editors or the contributors:

Name	Organization
Steve Liang	SensorUp / University of Calgary
Tania Khalafbeigi	SensorUp / University of Calgary
Josh Lieberman	Tumbling Walls

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
2015-11-05	0.1	Steve Liang, Tania Khalefb eigi	All	Document initialized
2016-06-02	0.9	Steve Liang, Tania Khalefb eigi	All	First complete version
2016-08-10	1.0	Joshua Lieberman	All	Clean up and editorial

1.4 Future work

This ER is intended to protocol mapping / conversion practices in support of OGC sensor web standards. As these practices mature, future specifications and standards should take them into account in order to improve SWE-IoT interoperability.

1.5 Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

2 References

The following documents are referenced directly or indirectly in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC 06-121r3, *OGC® Web Services Common Standard*

OGC 06-042, *OGC® Web Map Service (WMS)*

OGC 07-006r1, *OGC® Catalogue Services*

OGC 08-094r1, *OGC® SWE Common Data Model*

OGC 09-001, *OGC® SWE Service Model*

OGC 09-025r2, *OGC® Web Feature Service (WFS)*

OGC 10-025r1, *OGC® Observations and Measurements (O&M) - XML Implementation*
OGC 12-000, *OGC® Sensor Model Language (SensorML)*
OGC 12-006, *OGC® Sensor Observation Service (SOS)*
OGC 14-065, *OGC® Web Processing Service (WPS)*
OGC 15-078r3, *OGC Sensor Things API (STA)*
ITU-T Y.2060, *Overview of the Internet of things, June 2012*
ISO/IEC IS 29182-3, Information technology — Sensor Networks: Sensor Network
Reference Architecture (SNRA) — Part 3: Reference Architecture Views

NOTE This OWS Common Standard contains a list of normative references that are also applicable to this Implementation Standard.

3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r3] shall apply.

4 Conventions

4.1 Abbreviated terms

API	Application Program Interface
AVL	Automated Vehicle Location
AWS	Amazon Web Services
CSW	Catalogue Service for the Web
EML	Event Pattern Markup Language
ER	Engineering Report
GML	Geography Markup Language
IMIS	Incident Management Information Sharing
IoT	Internet of Things
JSON	Java Script Object Notation
KVP	Key-Value Pair
MQTT	Message Queue Telemetry Transport
O&M	Observation & Measurements
OSH	OpenSensorHub
OWS	OGC Web Services
POX	Plain Old XML
PTZ	Pan-Tilt-Zoom

SAS	Sensor Alert Service
SES	Sensor Event Service
SensorML	Sensor Model Language
SLD	Styled Layer Descriptor
SOS	Sensor Observation Service
STA	Sensor Things API
SWE	Sensor Web Enablement
UAS	Unmanned Aerial Sensor
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UUID	Universally Unique Identifier
WEPS	Web Event Processing Service
WFS	Web Feature Service
WMS	Web Map Service
WPS	Web Processing Service
XML	Extensible Markup Language

4.2 UML notation

Most diagrams that appear in this ER are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 06-121r3].

5 An IoT Reference Model

5.1 IoT Protocol Layers and Segments

It is important to establish a reference model first, and then we can introduce the different IoT devices implemented in this pilot and how their different protocols are mapped to the OGC IoT architecture.

In this engineering report, we adopt the International Telecommunications Union Standardization Sector (ITU-T) IoT reference model. Figure 1 shows the ITU-T reference model for the IoT. The reference model consists of four layers, namely Device Layer, Network Layer, Service Support and Application Support Layer, and Application Layer. In the context of the Pilot, Sensor Hub Services are the Service Support and Application Support Layer. Detailed definition of the Sensor Hub Service can be found in the IMIS IoT Pilot ER [REF].

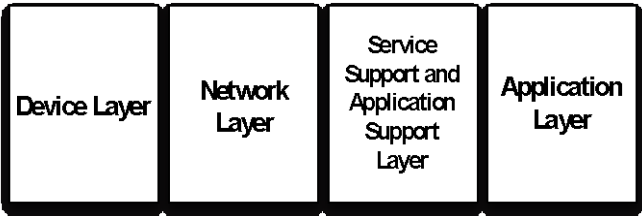


Figure 1 - IoT Reference Model (adapted from Rec. ITU-T Y.2060)

Each layer of the above four layers can be further divided into sub-layers. Figure 2 shows the sub-layers of the four major IoT layers from an OGC perspective.

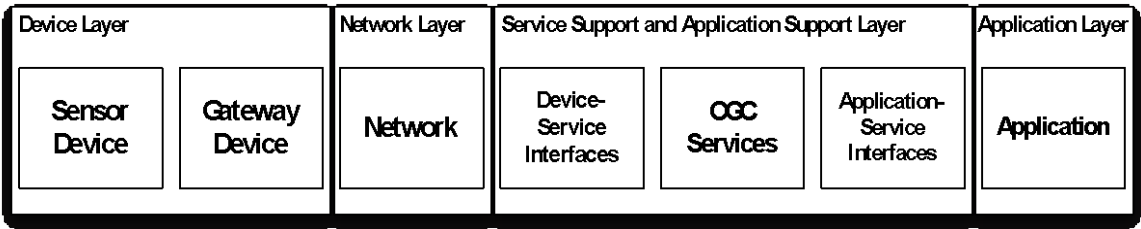


Figure 2 - IoT Reference Model and the Sub-layers

According to the ITU-T definition, an IoT device is able to communicate, and optionally can sense, actuate, store and process data [ITU-T Y2060]. We can further decompose the device layer into two sub-layers: (1) sensor device¹ and (2) gateway device. Sensor devices perform observations to detect or measure information related to the surrounding environment. IoT sensor devices are specifically able to transmit and/or receive information across a sensor network. In some cases, due to technology, size, cost, or other constraints, sensor devices are only able to form local sensor networks (LSN) or personal area networks (PAN) communicating with each other, or even simpler point-to-point connections, and require gateway devices to connect them with the wide area networks..

¹ In this pilot project, we did not include tasking scenarios, and as a result we did not include actuators.

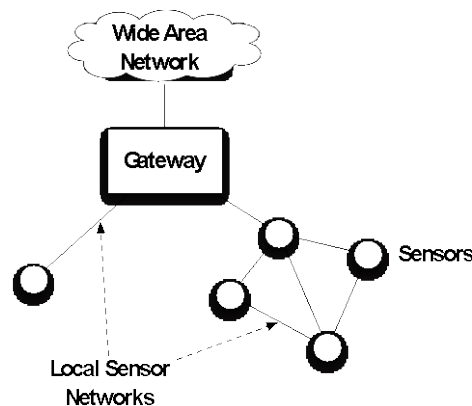


Figure 3 - Sensor Devices Form Local Sensor Networks (LSN) and Connect to Gateway Devices

A gateway role or device category also figures in an International Standards Organization (ISO)-specified IoT architecture (ISO/IEC IS 29182-3). The general role that such a gateway plays is to connect separate networks. Depending on the nature of the network stack on each side, the gateway may need to support a number of functions on multiple protocol layers.

The main function of a gateway device is to provide protocol conversion. Two different situations require protocol conversions. The first situation is when different devices use different LSN / PAN technologies, and the gateway device can provide protocol conversion capabilities between them. For example, one sensor device may use Bluetooth Low Energy (BLE) and other sensor devices may use ZigBee. A gateway device is required to convert the two different protocols so that the two groups of sensor devices are able to communicate with each other. The second situation is when the LSN / PAN and the wide area (IP) network (WAN) use different protocols / technologies. For example, the LSN may use a ZigBee technology and the WAN may use a 3G technology. A gateway device is needed in order for the sensor devices to communicate with the services in the wide area network.

The Pilot implemented particular derivatives of the ITU-T / ISO gateway concept termed Sensor Hubs (S-Hub). The S-Hub combines the protocol and/or physical mapping between networks of a gateway with a variety of additional server functions such as:

- publishing sensors and data through OGC Web services
- initiating / managing registration of sensors and services for discovery purposes,
- storing and forwarding sensor observations
- publishing and generating events and notifications
- processing sensor observations (e.g. georeferencing, calibration) to create higher-value derivative information

In the future, S-Hubs are likely to support additional functions such as security enforcement, semantic mediation, and sensor status tracking that will be important to a viable OGC SWE-IoT architecture. A complete specification of S-Hub functionality will be covered in another document, but specific protocol mapping / conversion examples from the Pilot implementations will be covered here.

5.2 Where Does the Protocol Mapping Happen?

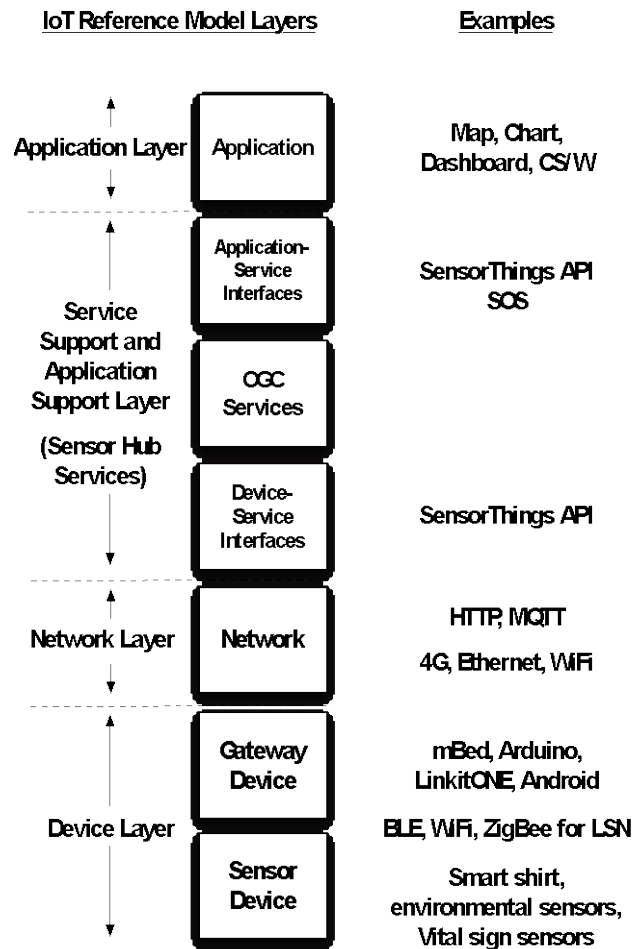


Figure 4 - The IoT Reference Model and the Technologies Experimented in the IMIS IoT Pilot

Figure 4 illustrates that mismatches between network segments can occur at any or all protocol layers in the ITU model. A gateway may therefore need to perform protocol mapping / conversion functions at any or all of these layers in order to maintain a true Internet of Things across them. Figure 4 also shows some of the protocols and technologies that played a part in the IMIS IoT Pilot, for example Catalog Service for the Web (CS/W), Message Queue Telemetry Transport (MQTT), and Bluetooth Low Energy (BLE). The diagram shows the vertical layering characteristic of an IoT distributed information system. This same stack is present in each of the horizontal segments of the system, LSN / PAN / WAN etc. but not all layers are equally well elaborated. For example, the application layer of a PAN segment is likely to be rudimentary at best, while the device details of most WAN segments can, one hopes, be safely ignored for most purposes. In the protocol mapping / conversion descriptions that follow, the main protocol or technology in each layer may be described but not necessarily each of the protocols being mapped from.

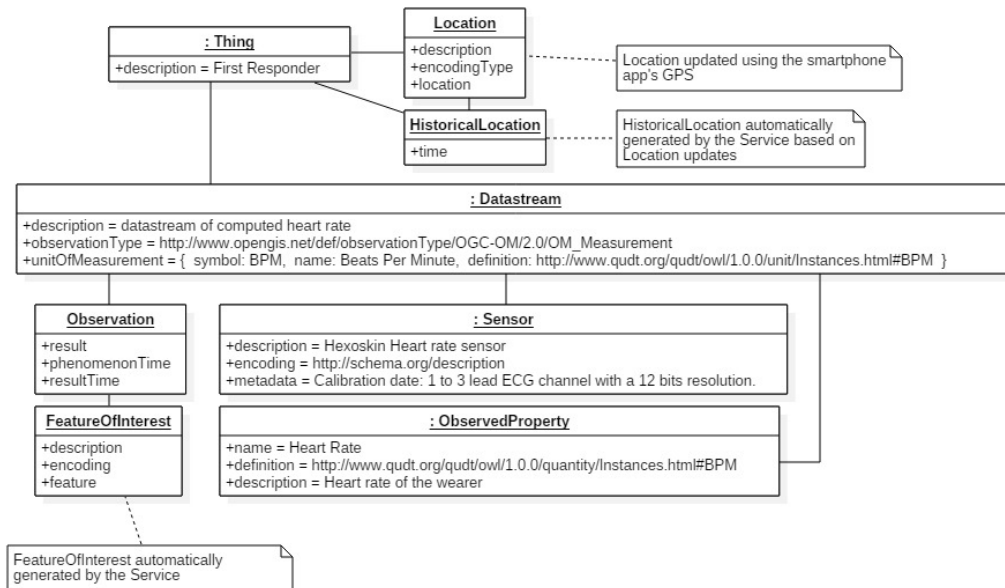
6 IMIS IoT Practices of Protocol Mapping

This section provides the implementation details of the protocol mapping implemented in the Pilot.

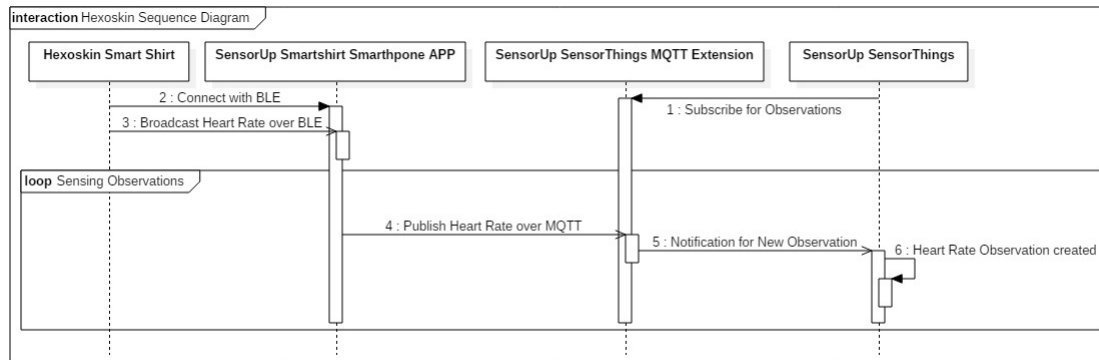
6.1 SensorUp Smart Shirt (Hexoskin)

	Protocol Used	Notes
Device Layer	Hexoskin Smart Shirt + BLE	http://www.hexoskin.com/
Gateway Layer	Android Smartphone	
Network Layer	HTTP/MQTT over 3G/4G or WiFi	
Device-Service Interfaces Layer	SensorThings API CREATE-UPDATE-DELETE	
Application-Service Interfaces Layer	SensorThings API and MQTT Extension	

6.1.1 Data Model



6.1.2 Sequence Diagram



6.1.3 Example Request/Response

Example 1 - Request/Response for Create Thing and Related Entities for Hexoskin Smart Shirt

```

POST /Things HTTP1.1
Host: example.org/v1.0
Content-Type: application/json

{
  "description": "First Responder",
  "Locations": [
    {
      "description": "GPS Location",
      "encodingType": "application/vnd.geo+json",
      "location": {
        "type": "Feature",
        "geometry": {
          "type": "Point",
          "coordinates": [-114.0708,51.0486]
        }
      }
    }
  ],
  "Datastreams": [
    {
      "description": "datastream of computed heart rate",
      "unitOfMeasurement": {
        "name": " Beats Per Minute",
        "symbol": "BPM",
        "definition": "http://www.qudt.org/qudt/owl/1.0.0/unit/Instances.html#BPM"
      },
      "observationType": "http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement",
      "ObservedProperty": {
        "name": "Heart Rate",
        "definition": "http://www.qudt.org/qudt/owl/1.0.0/quantity/Instances.html#BPM",
        "description": "Heart rate of the wearer"
      },
      "Sensor": {
        "description": "Hexoskin Heart rate sensor",
        "encodingType": "http://schema.org/description",
        "metadata": "Calibration date: 1 to 3 lead ECG channel with a 12 bits resolution."
      }
    }
  ]
}

```

Example Response

```
{
  "@iot.selfLink": "http://example.org/v1.0/Things(example_id)",
  "Datastreams@iot.navigationLink": "
http://example.org/v1.0/Things(example_id)/Datastreams",
  "@iot.id": example_id,
  "description": "First Responder",
  "Locations@iot.navigationLink": "
http://example.org/v1.0/Things(example_id)/Locations",
  "properties": {},
  "HistoricalLocations@iot.navigationLink":
"http://example.org/v1.0/Things(example_id)/HistoricalLocations"
}
```

Example 2 - MQTT Publish for Creating Hexoskin Heart Rate Observation

MQTT Publish

Topic: v1.0/Datastreams({example_id})/Observations

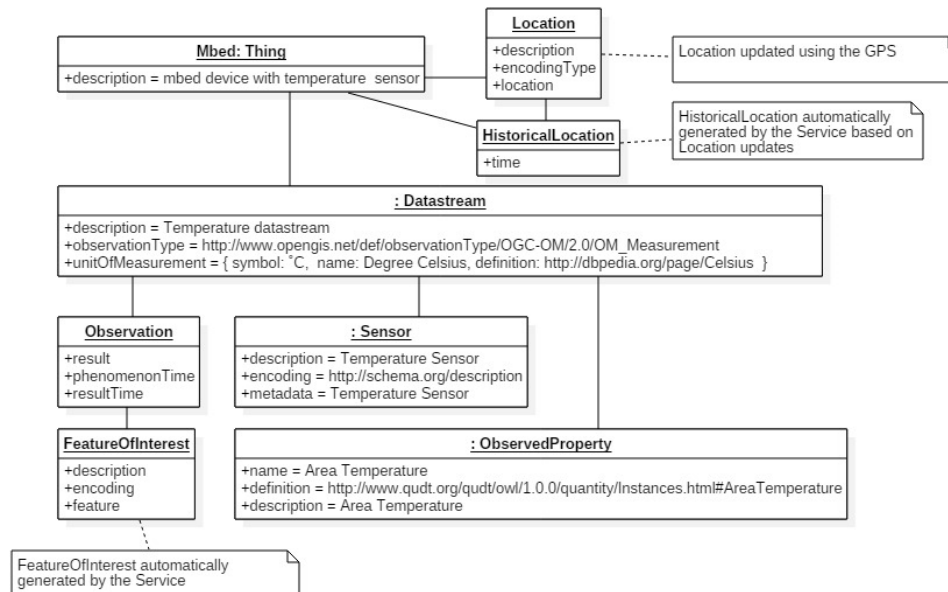
Message:

```
{
  "result": 70,
  "phenonmenonTime": "2015-02-05T17:00:00Z"
}
```

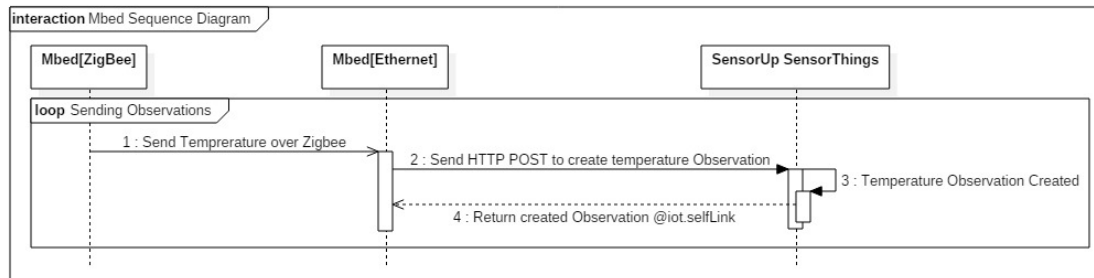
6.2 SensorUp mBed SDK - Temperature Sensor (ZigBee to Ethernet)

	Protocol Used	Notes
Device Layer	mBed Application Board + ZigBee	https://www.mbed.com/en/
Gateway Layer	mBed Application Board	
Network Layer	HTTP/MQTT over Ethernet	
Device-Service Interfaces Layer	SensorThings API CREATE-UPDATE-DELETE	
Application-Service Interfaces Layer	SensorThings API and MQTT Extension	

6.2.1 Data Model



6.2.2 Sequence Diagram



6.2.3 Example Request/Response

Example 3 - Request/Response for Create Thing and Related Entities for mBed

```

POST /Things HTTP1.1
Host: example.org/v1.0
Content-Type: application/json

{
  "description": " mbed device with temperature  sensor ",
  "Locations": [
    {
      "description": "GPS Location",
      "encodingType": "application/vnd.geo+json",
      "location": {
        "type": "Feature",
        "geometry": {
          "type": "Point",
          "coordinates": [10,10]
        }
      }
    }
  ]
}
    
```

```
    },
    "Datastreams": [
      {
        "description": " Temperature datastream ",
        "unitOfMeasurement": {
          "name": "Degree Celsius",
          "symbol": "°C",
          "definition": "http://dbpedia.org/page/Celsius"
        },
        "observationType": "http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement",
        "ObservedProperty": {
          "name": " Area Temperature ",
          "definition":
"http://www.qudt.org/qudt/owl/1.0.0/quantity/Instances.html#AreaTemperature",
          "description": " Area Temperature "
        },
        "Sensor": {
          "description": " Temperature Sensor",
          "encodingType": "http://schema.org/description",
          "metadata": " Temperature Sensor"
        }
      }
    ]
  }
}
```

Example Response

```
{
  "@iot.selfLink": "http://example.org/v1.0/Things(1753459)",
  "Datastreams@iot.navigationLink": "
http://example.org/v1.0/Things(12345)/Datastreams",
  "@iot.id": 12345,
  "description": "mbed device with temperature sensor",
  "Locations@iot.navigationLink": " http://example.org/v1.0/Things(12345)/Locations",
  "properties": {},
  "HistoricalLocations@iot.navigationLink":
"http://example.org/v1.0/Things(12345)/HistoricalLocations"
}
```

Example 4 - Request/Response for Create Temperature Observation for mBed

POST /Datastreams({id})/Observations HTTP1.1

Host: example.org/v1.0

Content-Type: application/json

```
{
  "result": 25.6,
  "phenomenonTime": "2015-02-05T17:00:00Z",
}
```

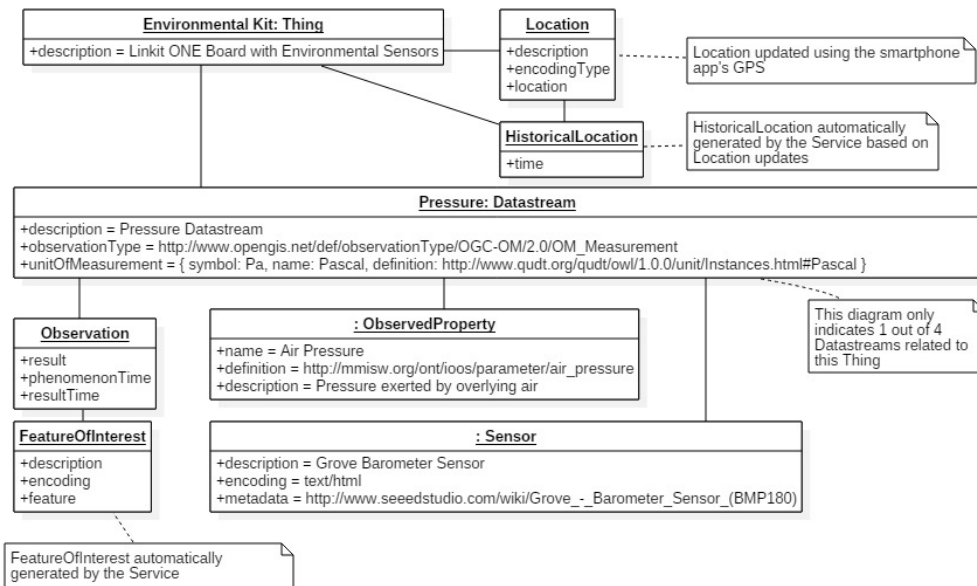
Example Response

```
{
  "@iot.selfLink": "http://example.org/v1.0/Observations(1753459)",
  "Datastream@iot.navigationLink": "
http://example.org/v1.0/Observations(12345)/Datastream",
  "@iot.id": 12345,
  "result": 25.6,
  "phenomenonTime": "2015-02-05T17:00:00Z",
  "resultTime": null,
  "FeatureOfInterest@iot.navigationLink":
"http://example.org/v1.0/Observations(12345)/FeatureOfInterest"
}
```

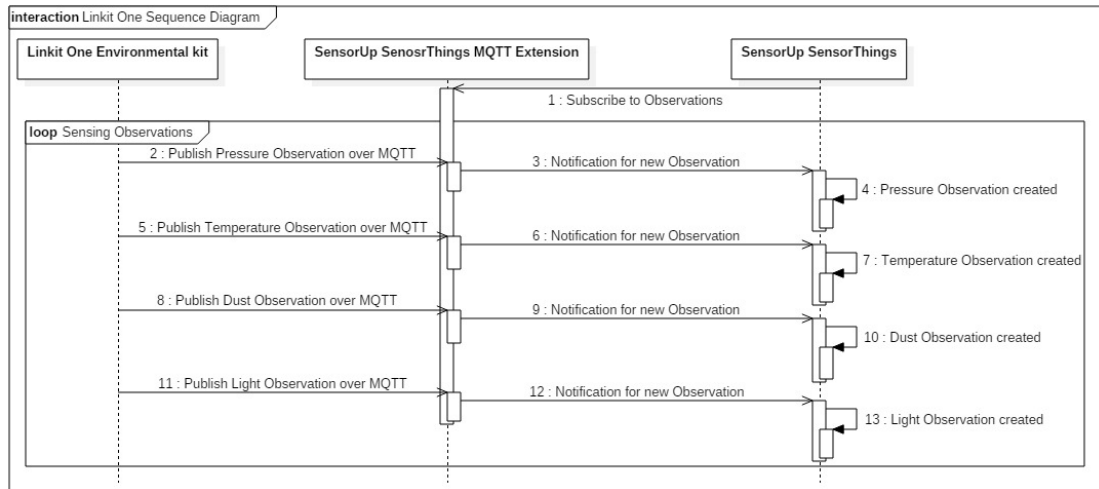
6.3 SensorUp Linkit ONE Software Development Kit - Environmental Sensor

	Protocol Used	Notes
Device Layer	Linkit ONE Board + Grove Sensors	Grove System
Gateway Layer	Linkit ONE Board	Linkit ONE
Network Layer	HTTP/MQTT over WiFi	
Device-Service Interfaces Layer	SensorThings API CREATE-UPDATE-DELETE	
Application-Service Interfaces Layer	SensorThings API and MQTT Extension	

6.3.1 Data Model



6.3.2 Sequence Diagram



6.3.3 Example Request/Response

Example 5 - Request/Response for Create Thing and Related Entities for Linkit ONE Environmental Kit²

```
POST /Things HTTP1.1
Host: example.org/v1.0
Content-Type: application/json

{
  "description": " Linkit ONE Board with Environmental Sensors ",
  "Locations": [
    {
      "description": "GPS Location",
      "encodingType": "application/vnd.geo+json",
      "location": {
        "type": "Feature",
        "geometry": {
          "type": "Point",
          "coordinates": [10,10]
        }
      }
    }
  ],
  "Datastreams": [
    {
      "description": " Pressure Datastream ",
      "unitOfMeasurement": {
        "name": "Pascal",
        "symbol": "Pa",
        "definition": " http://www.qudt.org/qudt/owl/1.0.0/unit/Instances.html#Pascal "
      },
      "observationType": "http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement",
      "ObservedProperty": {
        "name": " Air Pressure ",
        "definition": " http://mmisw.org/ont/ioos/parameter/air_pressure",

```

² The example only contains one out of four Datastreams for Linkit ONE Environmental Kit.

```

    "description": " Pressure exerted by overlying air"
  },
  "Sensor": {
    "description": " Grove Barometer Sensor ",
    "encodingType": " text/html ",
    "metadata": " http://www.seeedstudio.com/wiki/Grove_-_Barometer_Sensor_(BMP180) "
  }
}
]
}

```

Example Response

```

{
  "@iot.selfLink": "http://example.org/v1.0/Things(1753459)",
  "Datastreams@iot.navigationLink": "
http://example.org/v1.0/Things(12345)/Datastreams",
  "@iot.id": 12345,
  "description": "Linkit ONE Board with Environmental Sensors ",
  "Locations@iot.navigationLink": " http://example.org/v1.0/Things(12345)/Locations",
  "properties": {},
  "HistoricalLocations@iot.navigationLink":
"http://example.org/v1.0/Things(12345)/HistoricalLocations"
}

```

Example 6 – MQTT Publish for Creating Linkit ONE Temperature Observation

MQTT Publish

Topic: v1.0/Datastreams({id})/Observations

Message:

```

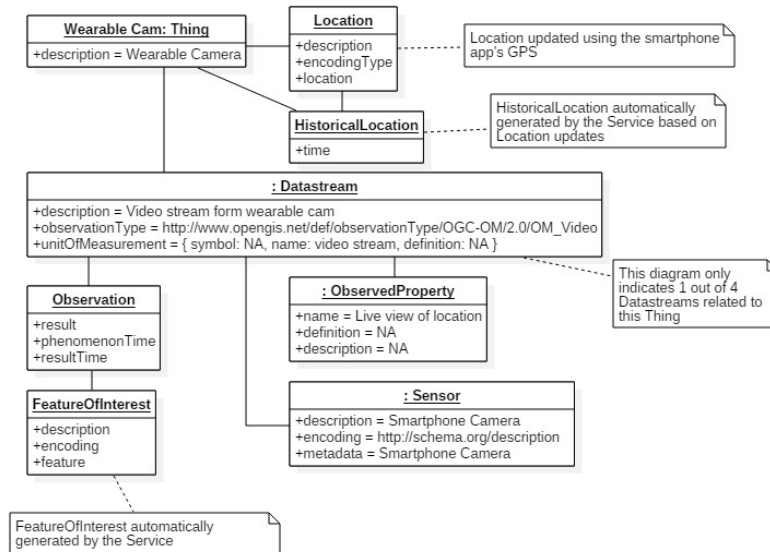
{
  "result": 23.1,
  "phenomenonTime": "2015-02-05T17:00:00Z"
}

```

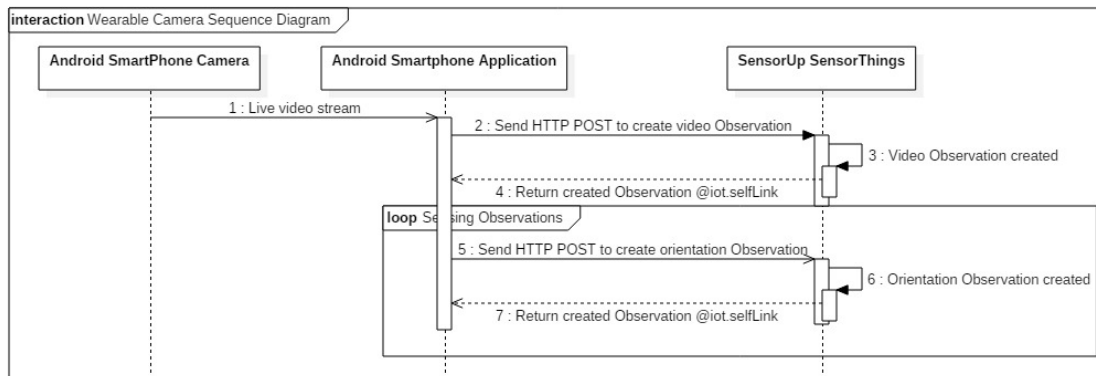
6.4 SensorUp SensorThings Wearable Cam

	Protocol Used	Notes
Device Layer	Android Smartphone	
Gateway Layer	Same as Above	
Network Layer	HTTP over WiFi	
Device-Service Interfaces Layer	SensorThings API CREATE-UPDATE-DELETE	
Application-Service Interfaces Layer	SensorThings API and MQTT Extension	

6.4.1 Data Model



6.4.2 Sequence Diagram



6.4.3 Example Request/Response

Example 7 - Request/Response for Adding Wearable Camera Video Stream to SensorThings

```

POST /Things HTTP/1.1
Host: example.org/v1.0
Content-Type: application/json

{
  "description": "Wearable Camera",
  "Locations": [
    {
      "description": "GPS Location",
      "encodingType": "application/vnd.geo+json",
      "location": {
        "type": "Feature",
        "geometry": {

```

```

        "type": "Point",
        "coordinates": [10,10]
      }
    }
  ],
  "Datastreams": [
    {
      "description": " Video stream form wearable cam ",
      "unitOfMeasurement": {
        "name": " video stream",
        "symbol": "NA",
        "definition": "NA"
      },
      "observationType": "http://www.opengis.net/def/observationType/OGC-
OM/2.0/OM_Video",
      "ObservedProperty": {
        "name": " Live view of location",
        "definition": "NA",
        "description": " NA "
      },
      "Sensor": {
        "description": " Smartphone Camera",
        "encodingType": "http://schema.org/description",
        "metadata": " Smartphone Camera"
      },
      "Observations": [{
        "result": " http://example.org/video"
      }]
    }
  ]
}

```

Example Response

```

{
  "@iot.selfLink": "http://example.org/v1.0/Things(1753459)",
  "Datastreams@iot.navigationLink": "
http://example.org/v1.0/Things(12345)/Datastreams",
  "@iot.id": 12345,
  "description": " Wearable Camera",
  "Locations@iot.navigationLink": " http://example.org/v1.0/Things(12345)/Locations",
  "properties": {},
  "HistoricalLocations@iot.navigationLink":
"http://example.org/v1.0/Things(12345)/HistoricalLocations"
}

```

Example 8 - Request/Response for Create Orientation Observation for Wearable Camera

POST /Datastreams({id})/Observations HTTP1.1

Host: example.org/v1.0

Content-Type: application/json

```

{
  "result": 83.38215637207031,
  "phenomenonTime": "2015-02-05T17:00:00Z",
}

```

Example Response

```

{
  "@iot.selfLink": "http://example.org/v1.0/Observations(1753459)",
  "Datastream@iot.navigationLink": "
http://example.org/v1.0/Observations(12345)/Datastream",
  "@iot.id": 12345,
  "result": 83.38215637207031,
  "phenomenonTime": "2015-02-05T17:00:00Z",
}

```

```

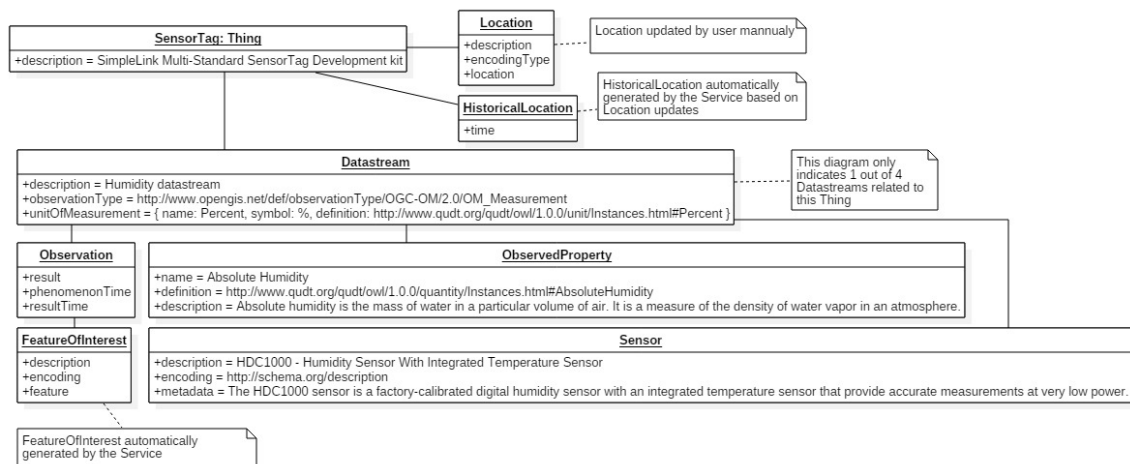
"resultTime": null,
"FeatureOfInterest@iot.navigationLink":
"http://example.org/v1.0/Observations(12345)/FeatureOfInterest"
}

```

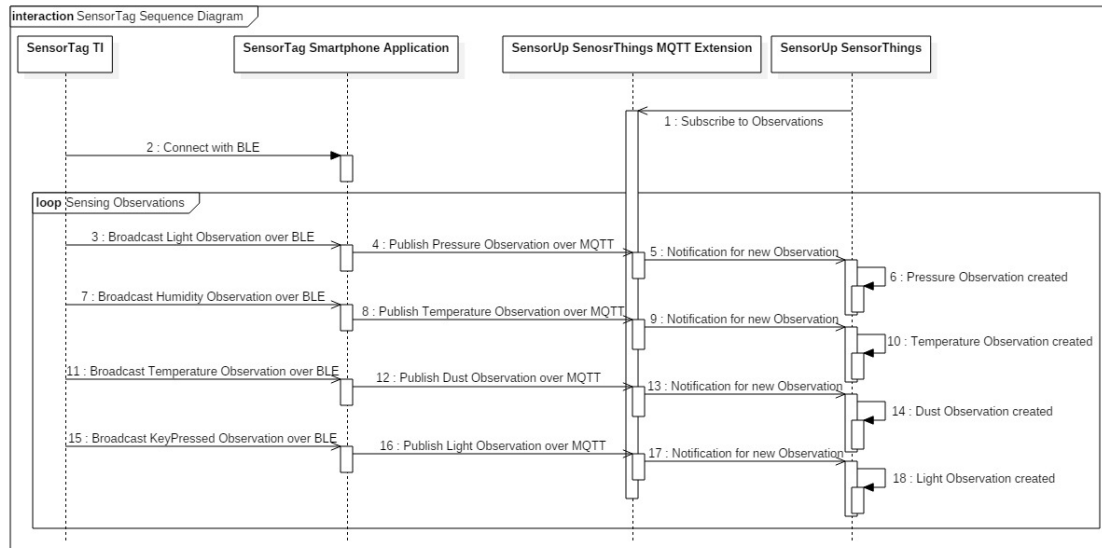
6.5 SensorUp SensorTag SDK

	Protocol Used	Notes
Device Layer	TI SensorTag	SensorTag
Gateway Layer	BLE	
Network Layer	MQTT over WiFi or 3G/4G	
Device-Service Interfaces Layer	SensorThings API MQTT CREATE	
Application-Service Interfaces Layer	SensorThings API and MQTT Extension	

6.5.1 Data Model



6.5.2 Sequence Diagram



6.5.3 Example Request/Response

Example 9 - Request/Response for Create Thing and Related Entities for SensorTag TI³

```

POST /Things HTTP1.1
Host: example.org/v1.0
Content-Type: application/json

{
  "description": " SimpleLink Multi-Standard SensorTag Development kit",
  "Locations": [
    {
      "description": "Manual Location Added",
      "encodingType": "application/vnd.geo+json",
      "location": {
        "type": "Feature",
        "geometry": {
          "type": "Point",
          "coordinates": [10,10]
        }
      }
    }
  ],
  "Datastreams": [
    {
      "description": "Humidity datastream",
      "unitOfMeasurement": {
        "name": "Percent",
        "symbol": "%",
        "definition": " http://www.qudt.org/qudt/owl/1.0.0/unit/Instances.html#Percent "
      },
      "observationType": "http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement",
      "ObservedProperty": {
        "name": " Absolute Humidity",

```

³ The example only contains one out of four Datastreams for SensorTag TI.

```

      "definition":
        "http://www.qudt.org/qudt/owl/1.0.0/quantity/Instances.html#AbsoluteHumidity",
        "description": " Absolute humidity is the mass of water in a particular volume
of air. It is a measure of the density of water vapor in an atmosphere."
      },
      "Sensor": {
        "description": " HDC1000 - Humidity Sensor With Integrated Temperature Sensor",
        "encodingType": " http://schema.org/description",
        "metadata": "The HDC1000 sensor is a factory-calibrated digital humidity sensor
with an integrated temperature sensor that provide accurate measurements at very low
power."
      }
    }
  ]
}

```

Example Response

```

{
  "@iot.selfLink": "http://example.org/v1.0/Things(1753459)",
  "Datastreams@iot.navigationLink": "
http://example.org/v1.0/Things(12345)/Datastreams",
  "@iot.id": 12345,
  "description": " SimpleLink Multi-Standard SensorTag Development kit",
  "Locations@iot.navigationLink": " http://example.org/v1.0/Things(12345)/Locations",
  "properties": {},
  "HistoricalLocations@iot.navigationLink":
"http://example.org/v1.0/Things(12345)/HistoricalLocations"
}

```

Example 9 - MQTT Publish for Creating SensorTag Humidity Observation

MQTT Publish

Topic: v1.0/Datastreams({id})/Observations

Message:

```

{
  "result": 45,
  "phenomenonTime": "2015-02-05T17:00:00Z"
}

```

6.6 Open Sensor Hub (OSH)-enabled Unmanned Aerial System (UAS)

	3DR Solo UAV + GoPro camera -> Linux Ground Control System	Customized ArduPilot controller software, connected by UDP MavLink telemetry
	Open Sensor Hub	Connected via Wi-Fi to UAV and controller

Network Layer	Wi-Fi to LTE Cellular	Tethered cellphone and cellular modem router
Device-Service Interfaces Layer	SOS Transaction	H.264 MP4 video encoding
Application-Service Interfaces Layer	SOS & JS browser client	

	Protocol Used	Notes
Device Layer	3DR Solo UAV + GoPro camera -> Linux Ground Control System	Customized ArduPilot controller software, connected by UDP MavLink telemetry
Gateway Layer	Open Sensor Hub	Connected via Wi-Fi to UAV and controller
Network Layer	Wi-Fi to LTE Cellular	Tethered cellphone and cellular modem router
Device-Service Interfaces Layer	SOS Transaction	H.264 MP4 video encoding
Application-Service Interfaces Layer	SOS & JS browser client	

	Protocol Used	Notes
Device Layer	3DR Solo UAV + GoPro camera -> Linux Ground Control System	Customized ArduPilot controller software, connected by UDP MavLink telemetry
Gateway Layer	Open Sensor Hub	Connected via Wi-Fi to UAV and controller
Network Layer	Wi-Fi to LTE Cellular	Tethered cellphone and cellular modem router
Device-Service Interfaces Layer	SOS Transaction	H.264 MP4 video encoding
Application-Service Interfaces Layer	SOS & JS browser client	

7 Summary and Conclusions

The IMIS IoT pilot project provided an opportunity to test the OGC IoT standards using real-world scenarios defined by first responders. It demonstrated the maturity of the OGC standards and the ‘state-of-practical’. This report describes and begins to systematize the protocol conversions and other adaptations that were undertaken in the pilot in order to develop a scalable information system architecture combining the best features of the Internet of Things and OGC Sensor Web

Enablement. It is clear from the work of aligning those features both vertically within the protocol stack and horizontally across segments of target sensor networks that further work is needed. Specifications of both the critical protocol conversions and the capabilities of an S-Hub component to implement them will be important next steps in the process of maturing this hybrid architecture as well as supporting further implementations in emergency response and other domains.