Open Geospatial Consortium

Publication Date: 2016-09-19

Approval Date: 2016-09-19

Posted Date: 2016-09-19

Reference number of this document: OGC 16-092

Reference URL for this document: http://www.opengis.net/doc/PER/IMIS IOT ER

Category: Draft Engineering Report

Editor: Roger Brackin

Incident Management Information Sharing (IMIS) Internet of Things (IoT) Extension Engineering Report

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

Contents

Page

Figures

Page

Tables

Page

Abstract

The Incident Management Information Sharing (IMIS) Internet of Things (IoT) Pilot established the following objectives:

- Apply Open Geospatial Consortium (OGC) principles and practices for collaborative development to existing standards and technology in order to prototype an 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.
- Development of profiles and extensions of existing Sensor Web Enablement (SWE) and other distributed computing standards to provide a basis for future IMIS sensor and observation interoperability.
- Prototype capabilities documented in engineering reports and demonstrated in a realistic incident management scenario.

These principles continued through the IoT Pilot Extension, with additional objectives of:

- Integration into the existing Next Generation First Responder (NGFR) Apex development program process as part of Spiral 1.
- Steps to begin the integration of existing incident management infrastructure, e.g., pulling in National Institute of Emergency Management (NIEM) message feeds.
- Demonstration and experimentation in a ‘realistic’ incident environment using two physically separate sites–an incident site within an active first responder training facility (Fairfax County Lorton site), and a command center (DHS S&T Vermont Avenue facility).

The initial Pilot activity has been documented in three OGC public engineering reports. The present report describes and documents the additional activities and innovations undertaken in the Extension.

Business Value

The IMIS IoT Pilot Extension continued the work of the IMIS IoT Pilot to develop, test and demonstrate the use of networked sensor technologies in a realistic test scenario developed in collaboration with DHS and first responder stakeholders. The Pilot Extension demonstrated an IoT approach to sensor use for incident management. Prototype capabilities included ad hoc, nearly automatic deployment, discovery and remote access to sensor information feeds, as well as derivation of actionable information in common formats for use in Computer Aided Dispatch, Emergency Operations Centers, Geographic Information systems and mobile devices.

Keywords

IMIS, IoT, OGC, Pilot, DHS, S-Hub, Sensor Things API, SOS, Datacasting

1 Introduction

1.1 Scope

This report describes the technical development, testing and demonstrations undertaken within the Department of Homeland Security (DHS) Internet of Things (IoT) Pilot Extension, integrated into the DHS Science and Technology Directorate (S&T) Next Generation First Responder (NGFR) APEX Program Spiral 1 development activity. As such, this document describes follow-on activities to the IMIS IoT Pilot which is itself covered by three earlier Engineering Reports (See Section 2 References).

The extension activities were carried out in close collaboration with the NGFR program and affiliated performers, and this experiment was part of Spiral 1 of that program. The extension work consisted of further development of IoT sensing and observation exploitation capabilities, integration with capabilities developed by the other NGFR performers, and demonstration of a new incident scenario involving an explosion and building collapse.

1.2 Document contributor contact points

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

Name	Organization (Alphabetical)
Don McGarry	Ardent
Mike Botts	Botts Innovative
Chris Clark	Compusult
Roger Brackin	Envitia (Editor)
Marcus Alzona	Noblis
Lewis Leinenweber	OGC
Steve Liang	Sensor Up/University of Calgary
Coral Bliss Taylor	Sensor Up
Mark O’Brien	SpectraRep
Joshua Lieberman	Tumbling Walls (Technical Lead)

1.2.1 Revision history

Date	Release	Editor	Primary clauses modified	Description
8/11/2016	0.5	Brackin, Roger	All	Complete draft

8/15/2016	0.6	Lieberman, Josh	All	Editorial updates
8/30/2016	0.7	Many	All	DHS S&T review and response

1.3 Future work

A description of identified areas requiring future work are provided in Section 8.

1.4 Forward

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 (OGC) 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 in this document.

1. OGC 15-118 IMIS Profile Recommendations for OGC Web Services
2. OGC 16-093 IMIS IoT Protocol Mapping Engineering Report (ER)
3. OGC 16-014r1 IMIS IoT Pilot Architecture ER
4. OGC 16-121 SWE-IoT Reference Model
5. OGC 06-121r3 Web Services Common Specification
6. 16-121 IMIS IOT Reference Architecture Draft (In preparation).

3 Terms and Definitions

The following primary terms and acronyms are used in this document:

Internet of Things (IoT)

Methodology for exploiting sensor information via the Internet and Web.

Catalogue Web Map Service (CatWMS)

An OGC Compliant Web Map Service rendering maps of catalogued sensors within the IMIS IoT Pilot. (Source: IMIS IoT Pilot)

Catalogue Service for the Web (CSW)

OGC Catalogue Service Specification. (Source: http://www.opengeospatial.org/standards/csw)

Datacasting

Transmission of data via broadcast channels (e.g., television).

Department of Homeland Security (DHS)

U.S. Government institution responsible for the first responder.

First Responder Extensible Sensor Hub (FRESH)

A capability integrating standard first responder information encoded as NIEM into a web service environment. (Source: DHS)

HAZMAT

Hazardous Material.

Hub Catalogue (HubCat)

Sensor Hub Catalogue used within IMIS IoT to catalogue sensor hubs available in a given operation. (Source: IMIS IoT Pilot)

Incident Management Information Sharing (IMIS)

Technology pilot undertaken by the OGC on behalf of the DHS. (Source: http://www.opengeospatial.org/pressroom/pressreleases/2211)

Inertial Measurement Unit (IMU)

Sensor used to measure inertial changes and determine direction of movement.

Message Queue Telemetry Transport (MQTT)

Machine-to-machine (M2M)/IoT connectivity protocol. (Source: MQTT.orghttp://mqtt.org/)

National Information Exchange Model (NIEM)

Framework, data model and set of XML Schema documents for information message exchanges.

Next Generation First Responder (NGFR)

DHS research program looking at first responder capability.

Public Safety Cloud (PSCloud)

A private computing and storage cloud supporting public safety. (Source: DHS)

Pan-Tilt-Zoom (PTZ) Camera

A camera, often in a fixed location which supports remote control of Pan Tilt Zoom. (Source: Well known concept)

Sensor Hub (S-Hub)

ASensor Observation Services (SOS or STA compliant Web Server that publishes data from locally networked / connected sensors. (Source: IMIS IoT Pilot)

Scanning Long-Term Evolution (LTE) Emission Detector (SLED)

UAV-mounted cellphone activity sensor.

Sensor Observation Service (SOS)

An OGC standard for delivering sensor information. (Source: OGC http://www.opengeospatial.org/standards/sos)

Sensor Planning Service (SPS)

An OGC standard used to task sensors. (Source: OGC http://www.opengeospatial.org/standards/sps)

SensorThings API (STA)

OGC standard for delivering sensor information. (Source: OGC)

Unmanned Aerial System (UAS)

Unmanned drone used to support camera or other sensor. (Source: UAVS.org https://www.uavs.org/index.php?page=what_is)

Web Map Service (WMS)

Web service delivering rendered map portrayals. (Source: http://www.opengeospatial.org/standards/wms)

Web Registry Processing Service (WRPS)

Web service harvesting content from other service capabilities documents and populating a Catalogue Service for the Web (CSW) Electronic Business Registry Information Model (CSW-ebRIM) compliant registry service. (Source: IMIS IoT Pilot)

Web Feature Service (WFS)

Web service delivering geographic features. (Source: OGC)

4 Conventions

4.1 UML notation

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

5 Experimentation and Demonstration

5.1 Context

The initial IMIS IoT Experiment/Demonstration focused on an outdoor incident around a road traffic accident with consequential damage to infrastructure, chemical release, etc. This involved the significant use of street-scale geospatial data and mapping. Responders to the incident dealt with the consequences of traffic chaos, powerline damage and chemical dispersal over a wide scale.

The IoT Pilot Extension scenario was based on the response to a collapsed building event. This had a much narrower geographic spread, with the scenario nominally limited to an area within the Fairfax County emergency training site at Lorton, Virginia, and focused on responders evaluating the exterior and interior of the collapsed building.

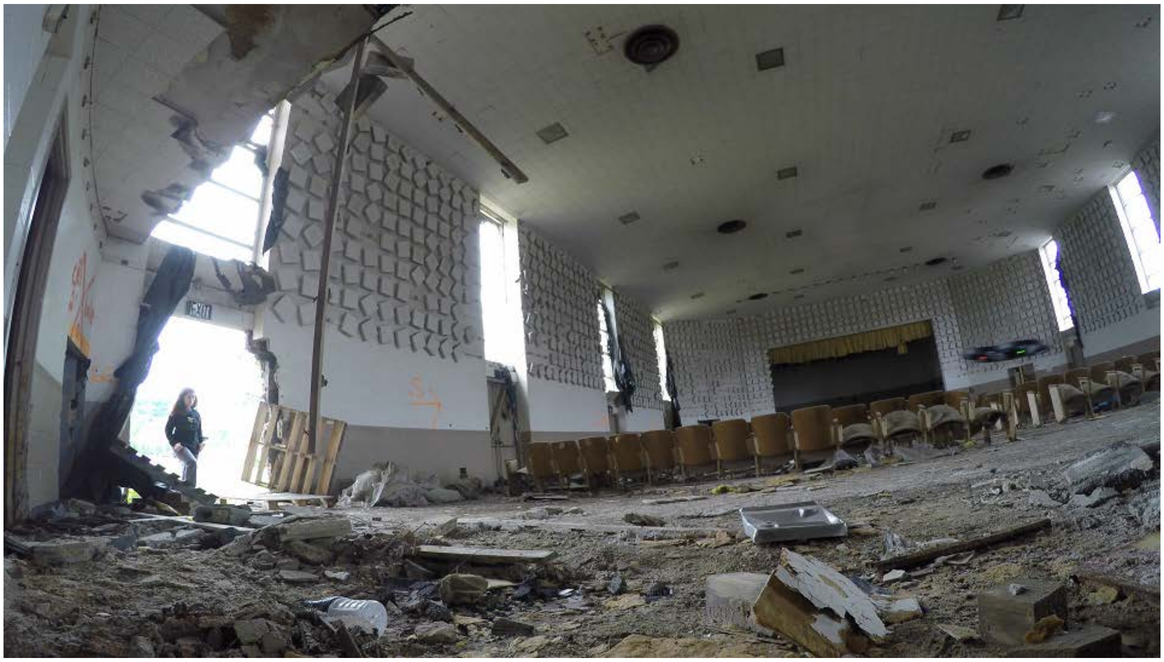


Figure 1 - *Pictures of Lorton Site/S&T headquarters*

5.1.1 Goals of the Trials/Demonstration

The demonstration had three primary goals:

1. To show that information from sensors connected via a range of standards and interfaces can be integrated and shared through common interfaces. The experiment demonstrated how the information produced from existing and new sensor sources can be federated and processed via a common information infrastructure and made available to all users across a variety of applications and platforms.

2. To show that this could be achieved in a ‘realistic’ environment, i.e., an event in the field, by deploying a scene-specific Field Sensor Hub (S-Hub) to serve as the main communication, tasking and data management hub for all on-scene sensors.

3. To provide connectivity and selective real-time streaming of sensor data to a central command center via an S-Hub deployed in the Public Safety Cloud (PSCloud), and to show how a range of information could be accessed broadly and efficiently in this manner.

5.2 Environment

The scenario was implemented across two sites, with participants deployed at the Lorton site simulating the incident occurrence and response, while those stationed at the DHS Vermont Avenue site (the location of the actual demonstration) simulated the incident command center. The latter site doubled as a secondary incident location for sensors in some of the experiments. The two sites were connected using a number of voice, video and data communications paths, none of which depended on existing DHS communications infrastructure.

Communication at the Lorton Facility consisted of a local Wi-Fi network and cellular LTE gateway established with a number of ‘Field S-Hubs’ through which local sensors received commands and transmitted observations. These hubs were implementations of the S-Hub concept developed for the IMIS IoT Pilot / NGFR Spiral 0 activity. In addition, ‘Cloud S-Hubs’ were established and deployed in the DHS PSCloud environment to implement the S-Hub concept. Cloud S-Hubs served to cache or mediate information from Field Hubs to lighten the load on responder worn or deployed devices. In addition to Wi-Fi and cellular LTE, a Datacasting capability was used to demonstrate additional communications approaches. Over the various communication channels, information such as location/orientation, responder heartrate, measured gas concentrations and high-definition (HD) video was transmitted from the Lorton site to the simulated command center at Vermont Avenue.

5.3 Demonstration Scenario

5.3.1 Overall Scenario

An incident is reported at the Lorton site and logged by dispatchers at the control center. An explosion has resulted in the collapse of a building. There may be trapped / or injured individuals, as well as exposure to hazardous materials and vapors.

5.3.2 Pre-arrival Preparation

As part of logging the event, the incident dispatcher entry automatically establishes an event ‘website,’ which is the focus of information related to the incident. This allows all information and sensors available to be identified in a catalogue for use by responders and incident coordinators. Certain standard information views can be created automatically.

5.3.3 Dispatching Resources

Responders are dispatched, an S-Hub is established at the incident scene, and unmanned aerial systems (UAS) or drones are deployed by responders to reconnoiter the affected building and surroundings. This gives command center operators an overall view of the situation. The site is deemed stable enough to work in, although there are concerns around possible gas leaks.

5.3.4 Preparation for Entry

First responders, as part of standard kit, are equipped with video cameras, external environment sensors (temperature, gas, chemical, etc.), and wearable physiological sensors measuring heart rate, breathing rate, etc.. The responder can also receive or initiate alerts based on their own kit readings or other information. To interfere as little as possible with a responder’s tasks, each wears a smartwatch on which they can receive alerts. Each responder kit connects automatically to the incident sensor network upon activation to send and receive pertinent information.

Indoor drones also plug-and-play with respect to the incident sensor network, and provide a view ahead of the responders into the collapsed building space. These provide live video feeds include and may also include onboard environmental sensors. In addition to their smartwatches, responders carry smartphones and/or tablets on which they can view the drone video.

5.3.5 Drones and Responders Progress into the Building

Drones enter the building followed by responders. The drone video can be monitored at the command center and allow the responders to be directed to specific areas. While the drones are equipped for autonomous operation, manual control is used due to the difficult interior conditions of the site (discussed later).

5.3.6 Responders and Sensor Systems Collaborate in the Building

A key goal of this scenario and the initial Pilot activity has been to demonstrate that every sensor, whether on drones, on responders, deployed in situ, model-derived or “humans-as-sensors,” can contribute to a globally accessible pool of incident information. Sensors’ observations are transmitted via field S-Hubs available at the incident scene to the field incident command post, to the Command Center and on to other participating organizations. Reliable field communications is always an issue, however, and so there is always a need to work around bandwidth or connectivity issues to ensure that sensor observations are preserved both locally and globally. During an incident response, therefore, field S-Hubs cache and serve sensor data on-scene and upload the data whenever possible to cloud-based S-Hubs. Responders then access the cloud whenever possible to minimize bandwidth demands on the field sensor network. Other innovative communications techniques (e.g., Datacasting) are also employed to ensure the best possible information availability during the response.

5.3.7 On-going Background Monitoring

Sensor information is automatically processed into a number of formats to ensure it can be exploited easily and to make critical information easier to recognize. Standardized observation data formats in particular allow application of geospatial analysis techniques, such as geofencing, that can alert busy responders and commanders to dangerous situations.

5.3.8 Hazardous Gas Fumes Alarm Propagation

First responders entering the collapsed building carry wireless air quality sensors that provide the responder and the sensor network with readings on hazards such as natural gas fumes. During investigation of the collapsed building, one of these sensors registers an increase in the level of gas. As the concentration level hits a configured threshold value, an alert is generated by the corresponding S-Hub and a notification is sent to all responders' smartwatches and the command center. The responders are directed to withdraw immediately until the gas level issue is rectified, but leave the gas sensor within the building to provide confirmation of safe conditions once the gas supply to the building can be cut off. While this particular notification is generated automatically based on sensor readings, an emergency notification can also be broadcast by any responder who separately discerns a hazard or safety issue.

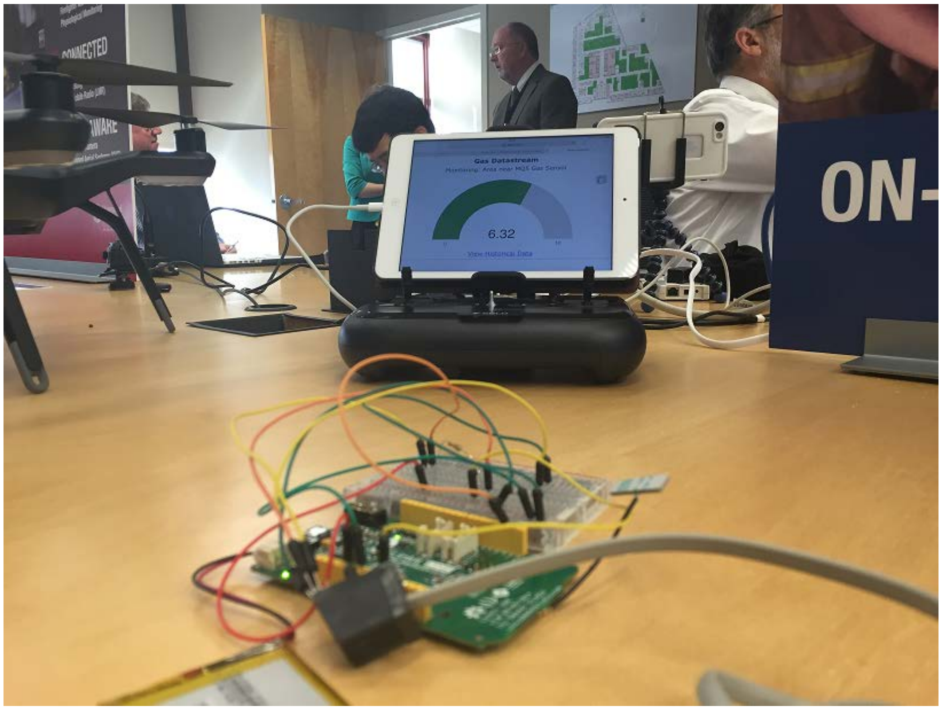


Figure 2 – Gas Sensor and Visualization on Tablet

5.3.9 Command Center Support to Responders

The command center monitors a range of information relating to the event as it unfolds. The command center has two primary means of visual engagement:

- The overall situational display (display wall). This visually conveys the overall situation, as well as specific critical or actionable information.

- Individual workstation displays. These are allocated to specific functions and manned by specialists dealing with an element of the response, for example overall coordination, utilities analysis, victim triage, etc.

Displays provide views of a number of types of information:

- General location/numbers of responders (implemented for outside locations) and their health state. Triggers indicate surpassing safe threshold levels of heart-rate, breathing rate or other health indicator in any responder.

- Environmental status of the site (structural stability, air quality, electrical hazard, etc.).

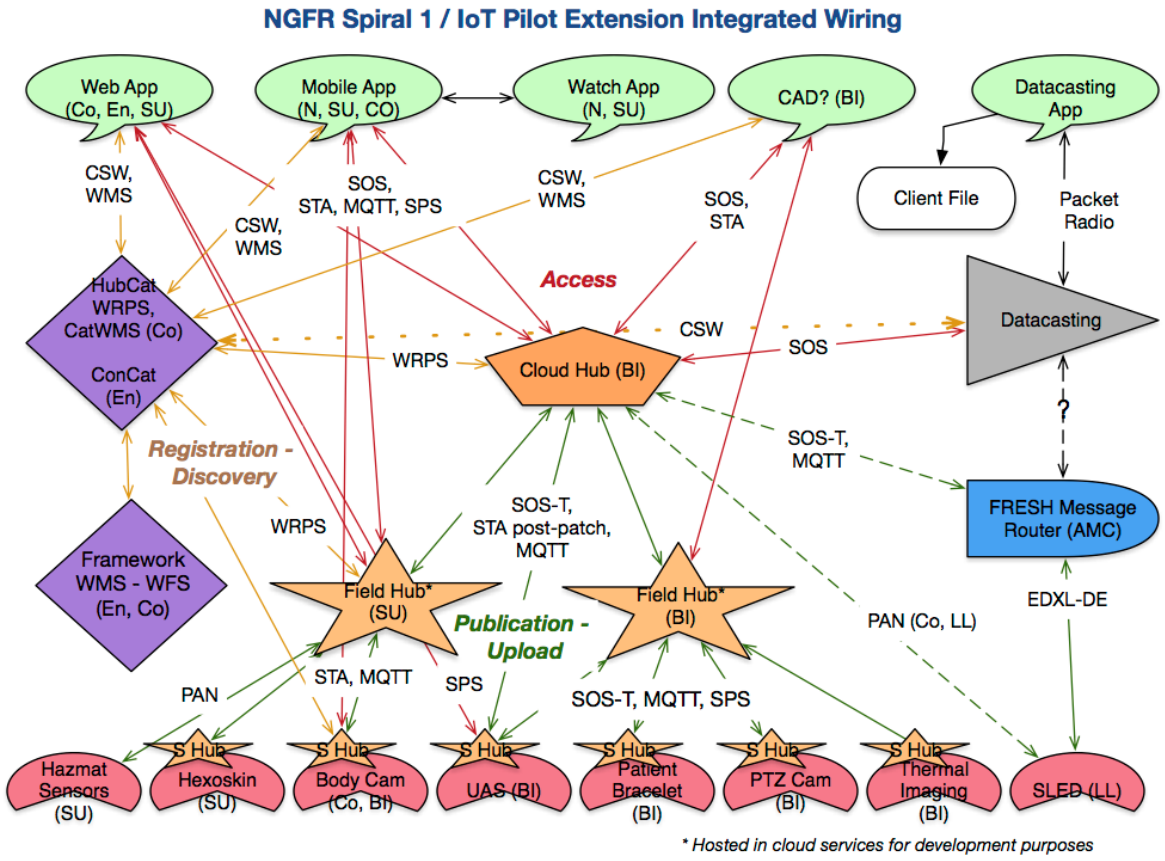
- Situation overview as video or still imagery (UAS, responder bodycam, PTZ camera, etc.).

- Existing framework data, maps and plans of site, critical site features such as power locations, entry and exit routes, etc.

6 Architecture

6.1 Overall Architecture

The overall component interaction architecture of the IoT Extension is shown below.



* Hosted in cloud services for development purposes

Figure 3 – NGFR IoT Extension Architecture

The Pilot Extension sensor network consisted of components of a number of different types, in most cases provided by more than one vendor. The key component types are:

- Catalogues (provided by Compusult and Envitia);
- Datacasting (provided by SpectraRep);
- Field S-Hubs (provided by Botts Innovative, Compusult and Sensor Up);
- Cloud S-Hubs (provided by Botts Innovative, Compusult and Sensor Up);
- Common Operating Picture Clients (provided by Compusult, Envitia);
- Mobile/Wrist Based Clients (provided by Sensor Up and Noblis);
- Sensor devices (provided by Botts Innovative, Compusult and Sensor Up);
- WiFi LAN + LTE WAN @ VTA (provided by Compusult & Tumbling Walls);

- Wireless LAN – Lorton (Provided by Botts Innovative (T-Mobile), Noblis (Verizon, T-Mobile)); and
- Routers for emergency services tracking (via NIEM) (provided by Ardent).

7 Technology Advances

7.1 Sensor Hubs in the PSCloud

One goal of this Pilot has been to establish, test, and demonstrate the ability and benefits of establishing S-Hubs within a cloud environment. For NGFR, the PSCloud was set up using Amazon Web Services. S-Hubs deployed within the PSCloud environment served as a globally accessible source of both real-time and archived observations during response and for post-emergency evaluation.

For the Pilot Extension, several S-Hubs were deployed on the PSCloud, each implementing one or more SWE and IoT standards. Since most interactions with cloud S-Hubs were through transactional services, basic OGC services' capabilities without specific local sensor drivers were sufficient.

For those cloud S-Hubs supporting Transactional Sensor Observation Service services (SOS-T), sensors can be added using a registerSensor() request. Real-time observations can then be streamed from the newly registered sensor and stored on a cloud S-Hub. Observation data persistence is an optional capability in Open Sensor Hub (OSH) implementations of S-Hub. These observations are then available in real-time or as archived data to any SOS client. Cloud S-Hubs were provided by three different participants: Botts Innovative, Compusult and Sensor Up

7.1.1 Cloud Hub 1 – Field Sensor Caching

The sensor observations that were uploaded to, and then accessed from, the Botts Innovative OSH-based cloud S-Hub included:

- Video and location data from two Garmin VIRB XEs;[1]
- Chest-strap heartbeat data also from the Garmin VIRB XEs;
- Video and location/orientation data from four Raspberry Pi [2]GeoCams;

[1] Garmin VIRB XE is a rugged camera intended for high performance sports activity such as Skiing, Cycling etc.

[2] Rasberry Pi is a small computer platform which can support a sensor hub.

- Video and location/orientation data from an Android phone and tablet;
- Remote locations from a TruPulse Laser Rangefinder;
- Video and location/orientation data from an Android-attached FLIR infra-red camera;
- Health data from an Angel Sensor wristband;
- Video and PTZ data from a Dahua HD video camera; and
- Video and location/orientation data from a 3DR Solo quadcopter.

Several of these video sources were successfully accessed in real-time by the SpectraRep datacasting team and incorporated into the PBS broadcast pipeline. This capability is described under Section 7.12.1 (communications infrastructure).

7.2 Cloud Hub 2 – Interfacing to DHS Message Feeds

One of the goals of the Pilot Extension was to integrate with the First Responder Extensible Sensor Hub (FRESH) Router implemented by Ardent, a message router service deployed in PSCloud to share first Responder and emergency management data as NIEM messages. The FRESH router provides a simple RESTful interface that uses the OASIS EDXL Distribution Element (DE) to handle messages. The following operations are available in the interface:

Http Method	**Endpoint**	**Description**
GET	<server address>/api/DE	Gets all DE. Returns DE distribution id, sender id, datetime sent, and lookup id.
GET	<server address>/api/DE/<lookup id>	Get DE with lookup id. Returns DE XML.
POST	<server address>/api/DE	Creates new DE. Message body contains DE XML, returns lookup id.

Http Method	Endpoint	Description
PUT	<server address>/api/DE	Updates DE with lookup id. Message body contains DE XML, returns success (200).
DELETE	<server address>/api/DE	Deletes all DE. Returns success (200).
DELETE	<server address>/api/DE/<lookup id>	Deletes DE with lookup id. Returns success (200).

The FRESH Router provides two ways to receive DE Messages:

- RESTful Interface
    - Using the interface provided above, users can query for the DE Messages.
- Message Federation
    - Endpoints can be manually configured to have DE Messages federated to them.

The FRESH router is not directly compliant with OGC SWE standards (SOS/STA), so it was connected via a custom adapter to a cloud S-Hub provided by Compusult. This driver implements the FRESH RESTful interface because there is no current means to configure the router dynamically for message federation. S-Hubs are activated and de-activated dynamically, so it would not have been practical to manually configure the router endpoint each time an S-Hub was activated. The RESTful interface also does not currently provide any way to perform custom queries, so the S-Hub would query the FRESH router on a schedule, and then compare the current result to the previous to identify any changes. This process unavoidably introduces a delay equal to the polling schedule and will definitely not scale well.

The S-Hub driver connected to the FRESH router also converts the data it receives so that it can be published as a SensorThings API (STA) service. The FRESH router provides both Cursor on Target (CoT) and NIEM messages, so the STA service provides a separate data-stream for each message type. The driver is able to pull out the date and location from each message type and map them directly into the STA response. There is no clear mapping currently established for the rest of the COT or NIEM message content, so the entire XMLformatted message is actually returned in the observation result. This approach is not compatible with Sensor Web clients per se, but would allow any clients that supported both STA and these message formats to digest them.

7.3 Sensor Hub as a Local Field Node

7.3.1 Lorton Deployed Sensor Hubs

Another goal of the Pilot Extension was to establish and test one or more field S-Hubs that anchor to local sensor networks at the incident scene. Field S-Hubs serve two main functions: persist and provide observations from sensors deployed in the field for the on-scene responders, and upload those same observations to cloud S-Hubs for access by the Command Center whenever there is adequate network connectivity from the field to do so.

For the Pilot Extension, an OSH instance was deployed on a laptop to serve as a field S-Hub and associated local Wi-Fi access point supporting the ingestion of and access to sensor observations. The field S-Hub also supported sensor tasking. While the deployed S-Hub used a rather ad hoc arrangement of cellular modems for WAN connectivity, future field S-Hubs will likely include robust multiplex remote connectivity capabilities, as well as a larger range of local area protocols (Wi-Fi, BlueTooth LE, Z-Wave, LTE, ZigBee, LinkITOne, etc.).

The local OSH-based field S-Hub successfully supported a range of sensors, including Garmin VIRB cameras, GeoCams, health monitors, Android phones, laser rangefinders, FLIR cameras and camera-equipped UAVs. Much of the data from these sensors were also uploaded to cloud S-Hubs from the incident scene via cellular LTE connections (phones and modems), including several video streams.

The biggest challenges with field S-Hub deployment involved network communications, both locally in the field and from the field to the PSCloud. Challenges included:

- Providing adequate range of Wi-Fi to sensors in the field;
- Powering local networks in the field away from power connections;
- Maintaining adequate LTE data bandwidth and throughput, particularly in competition with nearby highway usage that varied enormously with time of day; and
- Cellular data plans suited to response operations.

Other challenges involved maintaining enough battery capacity to support the different sensors in the field, as in adequately sized batteries, sufficient spare batteries and an effective means of recharging batteries as needed. The deployment of the OSH-based field

S-Hub itself was successful. Almost all system challenges centered around the communications and power issues discussed above.

7.3.2 Vermont Avenue Deployed Sensor Hubs

Although the deployment of sensors and S-Hubs at the Command Center lacked some degrees of realism, it did provide a way to show how sensors at multiple locations could deliver information into the entire network and deliver information to all responders. This also allowed a more vivid illustration of gas detection and alert generation (by way of a butane lighter) for the demonstration attendees, really showing the sensor functionality. Compusult and Sensor Up each deployed sensors at the Vermont facility, as well as the ability to deliver an alert simultaneously to Command Center personnel and responders at the Lorton site.

7.4 New Sensors and Platforms

7.4.1 IMIS IoT Initial Pilot

During the initial IMIS IoT Pilot activity, a wide range of sensors were integrated. These included:

- Fixed infrastructure sensors, weather stations, building alarm sensors, etc.;
- Unmanned Aerial Systems (UAS) with cameras;
- Real-time streaming video cameras (drop cams, body cams, etc.);
- Laser rangefinder (tagging remote locations);
- Plume models; and
- Responder deployed sensors (biometrics).

7.4.2 Enhanced Sensors for IOT Extension

For the Pilot Extension, the same set of sensor information types was gathered (with notable additions) with more integrated sets of sensors being tested, for example, a set of sensors integrated into a bodycam for a responder to wear. Additional video sensors were introduced, creating in some cases bandwidth issues from high-definition video stream. Table 1 shows the sensors deployed and/or tested in the Pilot Extension. In many cases, similar products and capabilities were provided by different participant companies; the overlap of capabilities allowed both testing of sensor interchangeability and opportunities to pool and compare experiences.

Table 1 – Sensors Deployed in IoT Extension

Environment

Technology	Communications[1]/Sensor Connection[2] Protocols	Observed Properties	Feature of Interest	Integrator
Aeotec MultiSensor 6	Z-Wave[1]	Motion, Temperature, Light, Humidity, Vibration, UV	Deployment Environment	Compusult
Fibaro Flood Sensor	Z-Wave[1]	Flood, Tamper, Temperature, Tilt	Deployment Environment	Compusult
Garmin Tempe	ANT+[1]	Temperature	Responder Environment	Compusult
Grove - CO2 Sensor	Grove[2]	CO2	Deployment Environment	Compusult
Grove - Gas Sensor (MQ5)	Grove[2]	Gas (H2, LPG, CH4, CO, Alcohol)	Deployment Environment	Compusult
Grove - Light Sensor	Grove[2]	Light	Deployment Environment	Compusult, Sensor Up
Grove - Sound Sensor	Grove[2]	Sound	Deployment Environment	Compusult
Grove - Temperature	Grove[2]	Temperature,	Deployment	Compusult,

Technology	Communications[1]/Sensor Connection[2] Protocols	Observed Properties	Feature of Interest	Integrator
and Humidity Sensor		Humidity	Environment	Sensor Up
Grove - UV Sensor	Grove[2]	UV	Deployment Environment	Compusult
Anemometer		Wind Speed	Deployment Environment	Sensor Up
Wind Vane		Wind Direction	Deployment Environment	Sensor Up

Physiology

Technology	Communications[1]/Sensor Connection[2] Protocols	Observed Properties	Feature of Interest	Integrator
Garmin Forerunner 235	ANT+[1]	Heart Rate	Responder	Compusult
Garmin HRM-Run	ANT+[1]	Heart Rate	Responder	Compusult, Botts Innovative
HEXOSKIN Smart Shirt	Bluetooth Smart[1]	Heart Rate, Breathing Rate	Responder	Sensor Up

Visual

Technology	Communications[1]/Sensor Connection[2]/Video Format[3] Protocols	Observed Properties	Feature of Interest	Integrator
Garmin VIRB XE	WIFI[1], RTSP[1], H264[3], mDNS	Video (H264), Location	Responder Environment	Botts Innovative, Compusult, Sensor Up
FLIR ONE	MJPEG[3]	Thermal Imagery (MJPEG)	Responder Environment	Compusult, Botts Innovative
Parrot Jumping Night Drone	WIFI[1], mDNS, MJPEG[3]	Video (MJPEG)	Drone Environment	Compusult
3DR Solo Drone		Video (H264), location, view field	Drone Environment	Botts Innovative

Location and Other

Technology	Sensor Connection[2] Protocols	Observed Properties	Feature of Interest	Integrator
GlobalSat BU-353-S4	USB[2]	Location	Deployment Location	Compusult
Android smartphones (e.g., Nexus 5)		Location, Orientation, Battery Level, Video	Responder	Sensor Up

Technology	Sensor Connection[2] Protocols	Observed Properties	Feature of Interest	Integrator
Android smartphones (e.g., Nexus 5)		Location, Orientation, Video, Laser Rangefinder, FLIR	Responder	Botts Innovative
iPhone		Location, Battery Level	Responder	Sensor Up
FRESH Router		Incident Event Details	Incident	Ardent, Compusult

7.4.2.1 Garmin VIRB (Botts Innovative, Compusult, Sensor Up)

The Garmin VIRB camera device comprises an off-the-shelf video camera that includes GPS location and acceleration, but not geo-orientation. The VIRB also supports some ANT+ sensors, such as heart rate monitors and external temperature sensors. This device has primarily been designed to record HD video onto a local SD card, but does support a real-time stream of lower-resolution video and still imagery.

The VIRB supports a wide variety of accessories for attaching the camera to people and things. Real-time streaming data is delivered through a Wi-Fi network connection using a JSON-based protocol. This protocol is proprietary and requires its own S-Hub software driver to make the data accessible through SWE standard interfaces.

VIRB units were supported in this Pilot by Botts Innovative, Compusult and SensorUp S-Hubs. SensorUp’s implementation (tutorial and source code) can be viewed at https://bitbucket.org/geosensorweblab/sta-virb-xe. Garmin also provides a VIRB network services API and the Compusult S-Hub provided support for this.

A number of issues were identified with VIRB:

- No geomagnetic sensors to provide view compass direction.
- Temporal resolution of navigation data is not designed for real-time application, so it is not suitable to support geolocation of imagery from a moving platform.

- Currently, there is no way to synchronize sensor values with the frames in the live stream video. Garmin was contacted and they stated: "Each piece of data is made available as quickly as it can be. Sensors are almost instantaneous, but the video encoding takes time, as does the transfer of the video stream. Since network timing can be so unreliable and the data sources are completely independent, we would most likely have to embed the sensor data or timing information into the video stream. While technically possible, the VIRB firmware doesn't have this capability. For now, we don't have any plans to add additional metadata for synchronization of live data and video."

- Only one client can be connected to the live video stream; therefore, an S-Hub is required to make the video available to more than one viewer.

7.4.2.2 Grove Sensors

Grove is a sensor hardware connector designed for fast prototyping, so that no soldering nor breadboard configuration is needed for hardware prototyping. Seeedstudio, the Grove provider, offers sample code for each module, as well as detailed specifications and implementation guides. Hundreds of sensors are currently available. A Grove S-Hub driver was written by Compusult to allow for the plug and play autoconfiguration of supported sensors. The driver supports all of the Grove sensors in the table above.

The following issues were identified with the Grove Sensors:

- They are not inherently plug and play, so the driver needs to manually match up the sensor to the port.

- Some sensors randomly cause the base shield firmware to crash. New sensor data is then unavailable until reboot.

- Each new sensor type requires a driver update to maintain the effective "plug and play" capability.

- Some sensors return raw “analog” values instead of specific measurements. For example, the gas sensor just indicates a conductance value that does not directly correspond to a concentration in the air and needs to be processed to provide actionable information.

7.4.2.3 Z-Wave Sensors

Z-Wave is a wireless communication protocol designed for home automation. It uses a source-routed mesh network architecture and supports secure communications. An S-Hub driver was written to work with RaZberry, an add on for Raspberry Pi that turns it into a fully operational Z-Wave gateway.

Rasberry Pi firmware/software provides a way to retrieve in JSON either the entire Z-Wave stack, or changes to the stack since a moment in time, using an HTTP GET request. The S-Hub driver was written to parse the Z-Wave stack, process any changes, and then update its sensor and observation data accordingly.

The following issues were identified with the Z-Wave sensors:

- The Z-Wave protocol is well defined; however, there still exist different interpretations of the protocol, so it takes some driver mediation magic to work with similar sensors from different vendors.

- The default configuration for most sensors is to conserve battery life by updating sensor readings infrequently; however, this can be adjusted for most sensors as needed.

7.4.2.4 Android Phone as a sensor platform

Android phones or tablets provide a range of valuable sensors on a computing platform that provides its own connectivity to a local or global network through Bluetooth, Wi-Fi and LTE connectivity. They can also be used, conveniently, for voice and text communications.

Sensors normally included in Android phones or tablets include: HD video/imagery camera, GPS and geo-orientation (acceleration and geomagnetic). The Android platform can support other sensors through Bluetooth, USB or Wi-Fi connections (e.g., laser rangefinder, IR camera, portable weather station, health monitors, etc.). Android platforms natively support Java applications and thus support software such as Botts Innovative Open Sensor Hub onboard.

There were no significant issues with using Android-based sensors, except that phone processing capabilities may get overwhelmed with supporting a large variety of sensing tasks. It may be perfectly appropriate to work with multiple Android devices to assure capacity and reliability.

7.4.2.5 Raspberry Pi "GeoCam"

The importance of being able to geolocate and georectify video and still imagery is becoming more and more recognized. It is important not only to know where the camera is, but also to recognize what direction it is looking and what actual location the camera is imaging. The Raspberry Pi (RPi) provides an excellent platform for building both prototype and operational sensor systems, including geospatially aware HD video cameras.

As part of the Pilot Extension, Botts Innovative provided four "GeoCams" that include an on-board OSH S-Hub and provide HD video, GPS location and geospatial orientation sensor outputs in a compact package that can run on attached batteries for up to 18 hours. They include a screw attachment for GoPro and Garmin mounting accessories. GeoCams function as independent (local) S-Hubs and can also stream observations using SOS-T through a Wi-Fi or ethernet connection to a field or cloud S-Hub.

The GeoCams are configured to run the OSH software automatically and register with other S-Hubs whenever they are turned on. The present GeoCams are prototypes and assembled by hand. They can also be modified individually at this point for particular needs with additional sensors or LTE communication through an onboad cellular modem (e.g., FONA). They may serve as model designs for future COTS products.

7.4.2.6 Angel Sensor Wristband

The Angel Wristband is designed to monitor the wearer's heart-rate, body temperature, movement and oxygen concentration. The device communicates with other devices through Bluetooth. The Angel Wristbands deployed in the Pilot Extension were integrated with OSH-based S-Hubs. There were substantial issues with this device in terms of sensor consistency. The measurements from the sensor were sporadic at best with heartbeat only being monitored when the person/patient was perfectly still. The O_2 sensors never reported any values to the S-Hub or through the Angel Sensor API.

7.4.2.7 Apple Watch and Android Watch

The Apple Watch does carry health sensors, but in the Pilot Extension was used as an output device. The iPhone / Apple Watch sensor alert app developed by Noblis acts as a STA client, connecting to a STA service to monitor sensor readings for alert conditions. The current app added the capability to select between different S-Hubs, including Botts cloud S-Hub, SensorUp field S-Hub and Compusult field S-Hub)

Samsung Watch / Android Wear devices were used as STA / MQTT clients and as heartrate monitors, connecting to SensorUp S-Hubs.

7.4.2.8 3DR Solo Drone Video

The 3DR Solo quadcopter system provides an open hardware / software platform for supporting additional functionality. For the Pilot, software modifications were made to the Solo operating system to transmit gimbal settings, GPS location, geospatial orientation and HD video via Wi-Fi into an OSH-based S-Hub ground station. The streaming imagery could then be geo-rectified on demand allowing one to view the current camera view, as well as its map footprint within a Cesium3D-based browser client. The client application also included a window with traditional aircraft-type controls for altitude, attitude, direction and speed. Real-time geo-rectification of UAS video and still imagery greatly increases the value of UAV observations for providing situation awareness remotely, as well as connecting observations to other georeferenced information.

Issues with Solo Drone included:

- The 3DR Solo is designed principally to fly outdoors under GPS control. It provides a wealth of functionality for stabilized, constrained and even autonomous flight plans as long as a GPS fix is available. It does not, however, have an alternate means of indoor stabilization or navigation (e.g., optical flow sensors). Minus a GPS link and inside a building with an abundance of wind crosscurrents from broken windows (the case at Lorton), it proved extremely challenging to control. It is likely that reliable UAS operations would best be served by different craft that are specialized for indoor and outdoor conditions.

- The FAA has been evaluating regulations for commercial use of drone aircraft within the U.S. Understanding and obeying the rules of piloting drones for anything other than recreational use has been extremely difficult. This ambiguity made outdoor flights at the Lorton Facility problematic. The FAA has recently released new guidelines/rules for drone flight, which are set to take effect in August 2016. These new rulings should make the licensing requirements for recreational, commercial and research drone flight both simpler and more transparent. Combined with a now smaller restricted DC air space, this should allow outdoor flight at Lorton in the future and increase the opportunities to demonstrate the value of UAS operations in response activities.

7.5 Sensor Plug and Play

Sensor Plug and Play is achieved in the S-Hub architecture via the use of drivers. The purpose of the driver is to communicate with each sensor's often local and/or proprietary protocols so as to make its data available using open SWE and IoT standards.

In each of the S-Hubs used in the Pilot activities, drivers were set up before deployment, or uploaded via an administrative interface to allow for the addition of new sensors and sensor types. Each S-Hub developer provided some sort of administrative interface to configure sensors.

The typical process for adding a sensor to an S-Hub is shown below:

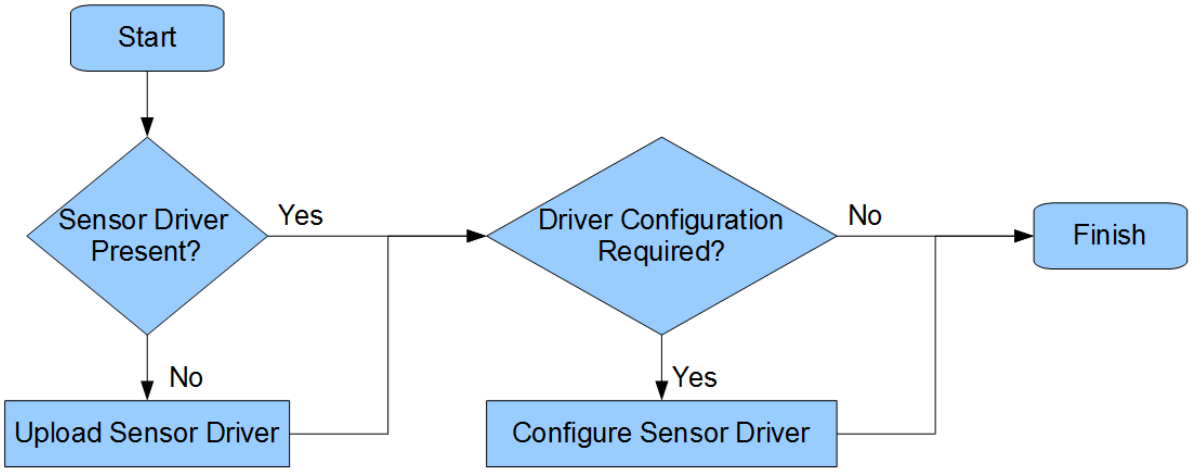
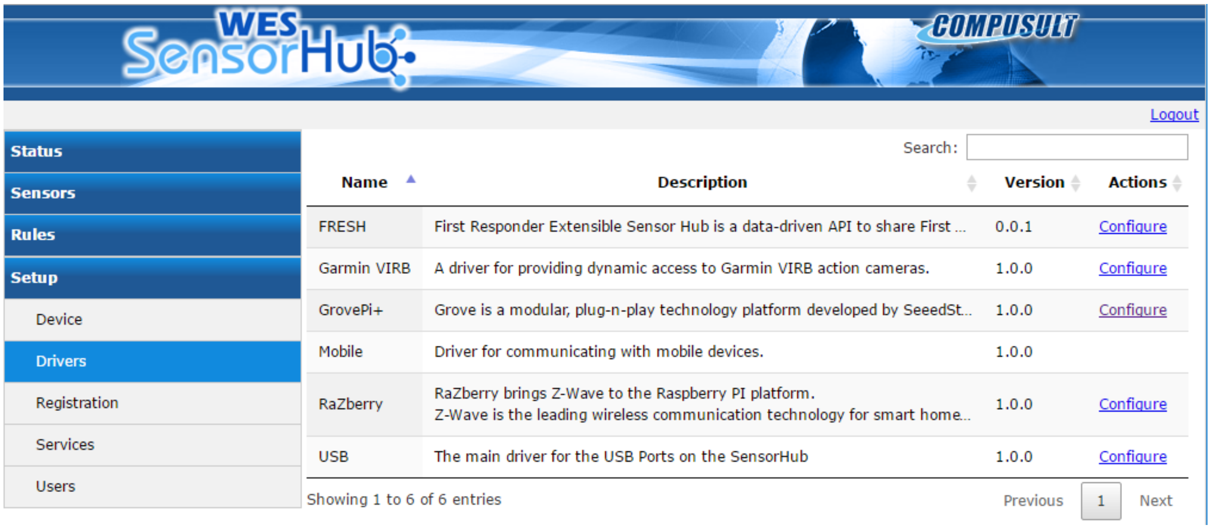


Figure 4 – Sensor Configuration Process

Of course, if a particular sensor technology or implementation (as sensor interfaces are typically vendor specific) is not yet supported, it is typically necessary to develop a new driver. Once that driver is built and configured, it is a simple task to add a new sensor (either via simple configuration files or through an easy-to-use Graphical User Interface (GUI).

7.5.1.1 Configuring Sensors

The following image shows an example listing of configured drivers in the Compusult S-Hub.

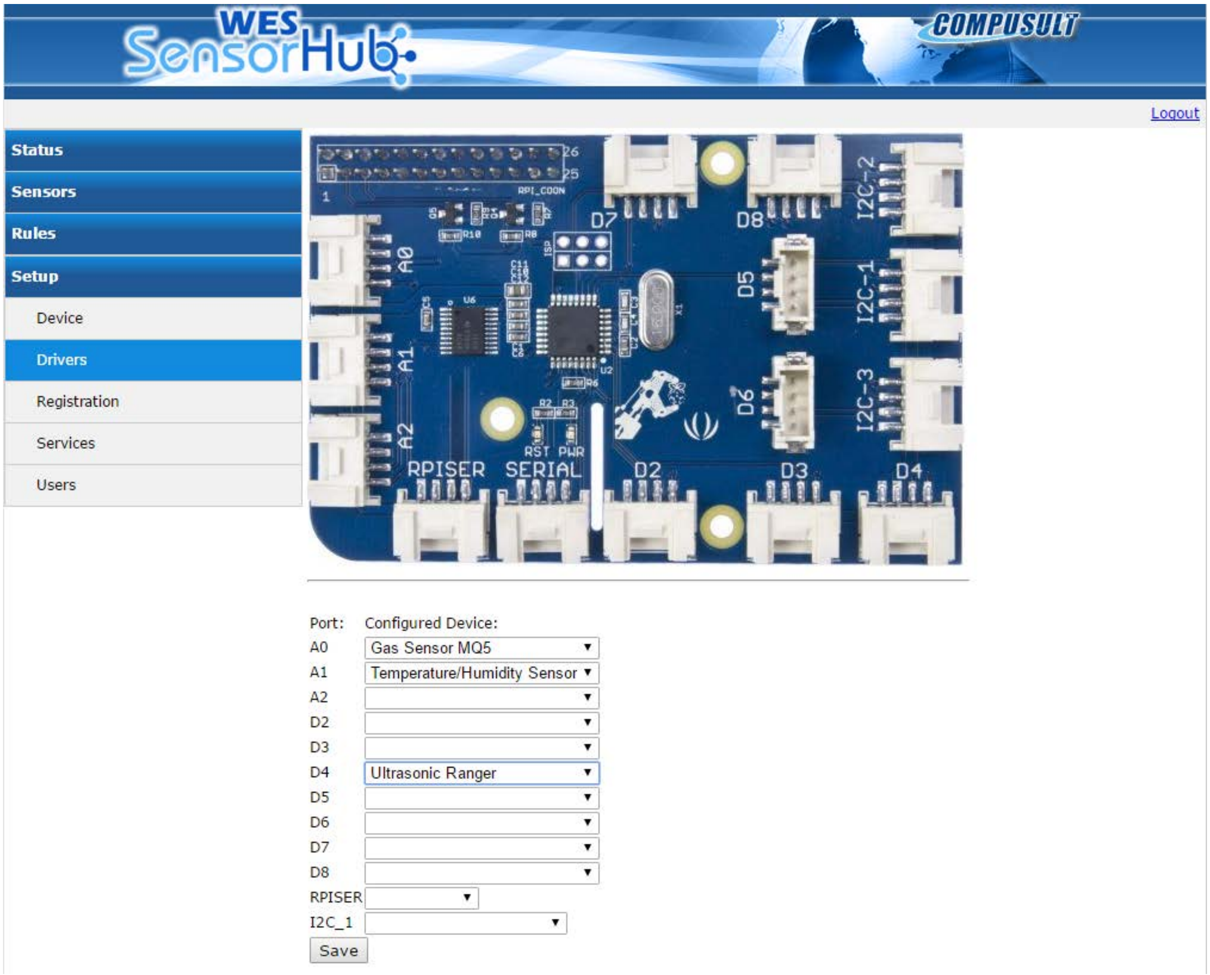


Name	Description	Version	Actions
FRESH	First Responder Extensible Sensor Hub is a data-driven API to share First ...	0.0.1	Configure
Garmin VIRB	A driver for providing dynamic access to Garmin VIRB action cameras.	1.0.0	Configure
GrovePi+	Grove is a modular, plug-n-play technology platform developed by SeeedSt...	1.0.0	Configure
Mobile	Driver for communicating with mobile devices.	1.0.0	
RaZberry	RaZberry brings Z-Wave to the Raspberry PI platform. Z-Wave is the leading wireless communication technology for smart home...	1.0.0	Configure
USB	The main driver for the USB Ports on the SensorHub	1.0.0	Configure

Showing 1 to 6 of 6 entries

Figure 2 – Configured Sensor Page

Drivers can also include interface pages that display the configuration settings for the current driver; however, this is only required if the sensors are not built on a protocol that allows for true plug and play operation. A driver that adds support for Z-Wave sensors may not require a configuration page because the device will pair and automatically appear upon startup. A driver that adds support for Grove sensors, however, will require a configuration page because the Grove specification does not provide for automatic detection of new sensors. The following is an example configuration page for Grove sensors in the Compusult S-Hub.



Port:	Configured Device:
A0	Gas Sensor MQ5
A1	Temperature/Humidity Sensor
A2	
D2	
D3	
D4	Ultrasonic Ranger
D5	
D6	
D7	
D8	
RPISER	
I2C_1	

Figure 2 – Sensor Configuration Page

Standard protocols supported by S-Hubs that allow true plug and play operation include, but are not limited to, Z-Wave, mDNS, Bluetooth and ANT+.

7.6 Sensor Hub Cataloging

During the initial Pilot, an issue was identified with how services were registered in a hub catalog (HubCat). This was documented in Reference [3]. A client registering a service through a Web Registration Processing Service (WRPS) only provides a URL to that service and only the metadata provided by the service can be harvested. Specifically, the catalog does not know which sets of services are providing the same data or are hosted by the same S-Hub.

To address this, the INSERT process of the WRPS was updated to include an optional parameter “ISO_METADATA”. This parameter allows a client to specify an additional ISO 19119 - compliant metadata document that the service harvesting process will add to its metadata, allowing the client to provide additional metadata that cannot be obtained from a service itself and also allows linkages to be specified between datasets and services. These linkages are created by taking advantage of the parameters available inside the transferOptions ISO element.

7.7 Sensor Tasking

STA tasking was implemented by Compusult as outlined in the draft Tasking Profile from the STA standards working group dated 20140807. The tasking profile was not included in the approved base STA standard, but some form of it will be included in a later extension standard.

Rules can be configured in Compusult S-Hub to activate tasks. For this pilot, a rule was created to detect a significant gas leak. When the rule matched observations, rule processing would task a red LED light connected to the S-Hub to turn on, as well as task an LCD screen to display the alert message. When the gas cleared, the rule processing would then task the LED light to turn off and the LCD screen to show “All Clear” (Rule configuration described in more detail in alerting section).

7.8 Propagation of Sensor Information into WFS/WMS Services

Present OGC SWE-IOT standards form a useful platform for working with sensor observations. The platform is quite effective at collecting and delivering observation data, but is less effective in supporting analysis and visualization of sensor information. Two standards that are widely supported in the Geospatial Information community for accessing feature and attribute information (for analysis and visualization) are the Web Feature Service (WFS) and Web Map Service (WMS). These services can be exploited by a wide range of applications. WFS specifically stores features and feature properties, some of which can be SWE-IoT features such as Feature of Interest (FoI), Observation, Result, etc. This means it is possible to map SWE service contents to WFS feature types that can be used by a wide range of GI applications for analysis and visualization.

WMS provides visualization-ready map layers that may also allow picking and interrogation of map features for their properties and other information. Agile information infrastructures are a means to define transformations between source sensor information, WFS feature types and WMS map layers. Two types of transformations were implemented in the Pilot Extension.

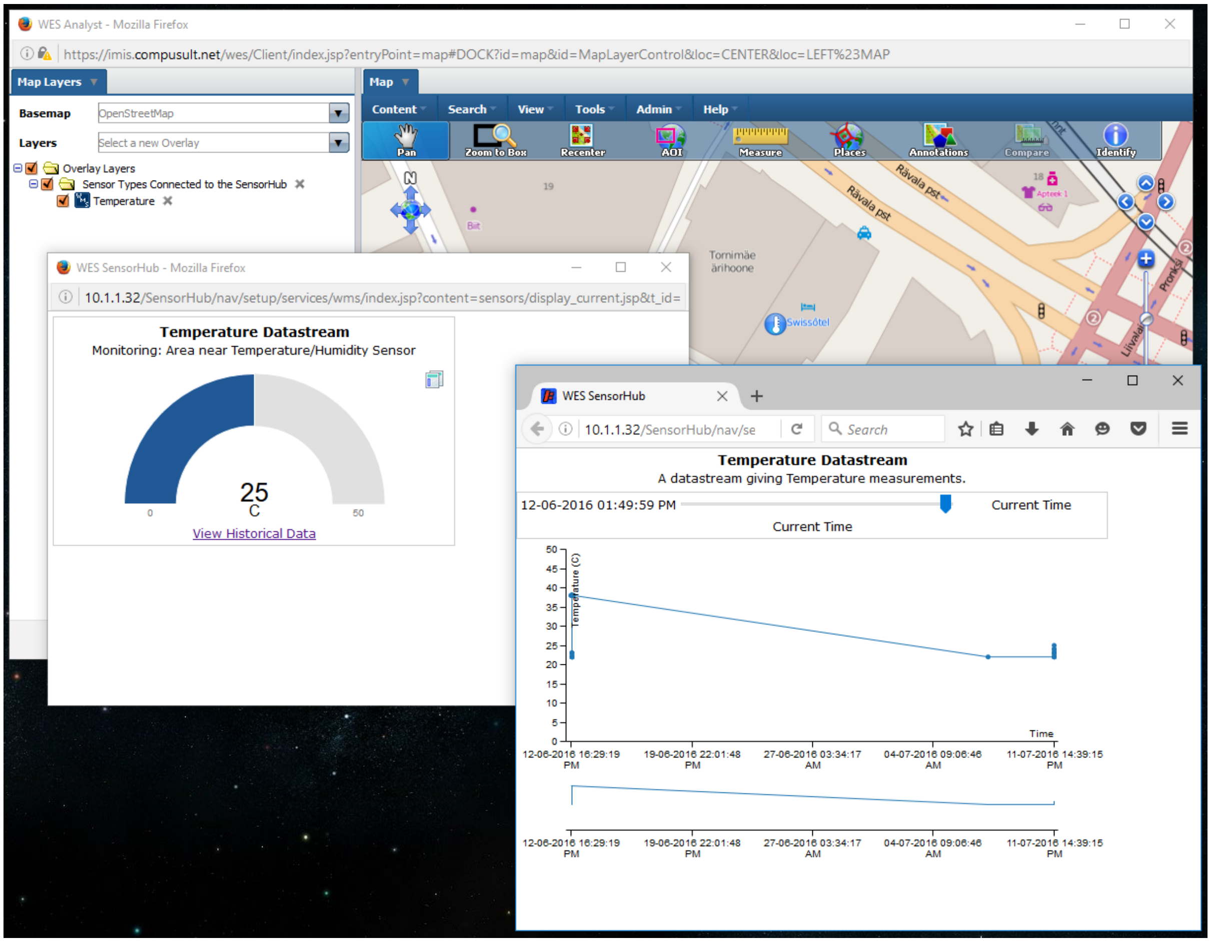
7.8.1 WMS Endpoints on Sensor Hubs

Within the Pilot Extension, Compusult implemented a visualization of the sensor information provided by its S-Hub through a WMS endpoint published parallel with a STA endpoint. The WMS was registered with an online linkage to the SensorThings service in the HubCat and vice versa.

The WMS layers grouped observations based on their ObservedProperty value in the SensorThings service, allowing users to display all current data for a given ObservedProperty on a single map layer. The data themselves were symbolized by appropriate icons on the map, e.g., a temperature gauge for temperature, raindrop for humidity, etc. A future improvement may leverage WMS dimensions to allow for temporally specific retrieval of data, as well.

The connection between S-Hub, STA and WMS content is made through the WMS GetFeatureInfo operation. This operation returns content in text/html, allowing the WMS to cue the appropriate web-based display widgets for each sensor. Display widgets then display real time and historical sensor readings. By providing a WMS alongside the STA service, we have enabled HTML-aware clients that support WMS with text/html GetFeatureInfo to view realtime sensor data without requiring them to support SWE standards directly. The WMS implementation also responds to any alerts that have been fired by the system and is able to change rendered icons to indicate that a specific sensor is in an alert state. GetFeatureInfo for an alerted sensor indicates the alert along with the typical display page.

The following image is an example of selecting the Temperature layer and opening the GetFeatureInfo for the layer.



7.8.2 STA/SOS Harvester Web Processing Service

A second approach, implemented by Envitia, is to use a standard WFS/WMS combination (here GeoServer WFS/WMS) and implement a harvesting service that transforms data served through SOS or STA into WFS feature types. This involves issuing SOS or STA new data requests and then updating the features served by the WFS. A benefit of populating the WFS is that if the service also supports a Styled Layer Descriptor (SLD) compliant WMS, WFS data can be directly visualized in the WMS using the flexible symbolization and style model defined in the SLD. Envitia developed the harvesting service described, which takes an XML definition of the harvesting to be implemented, and a mapping of SOS or STA FOI/Observations/ObservedProperties from to WFS feature types and properties. Envitia also developed SLD documents to render these FeatureTypes/Properties into appropriate map symbols.

The overall structure of this solution is shown below.

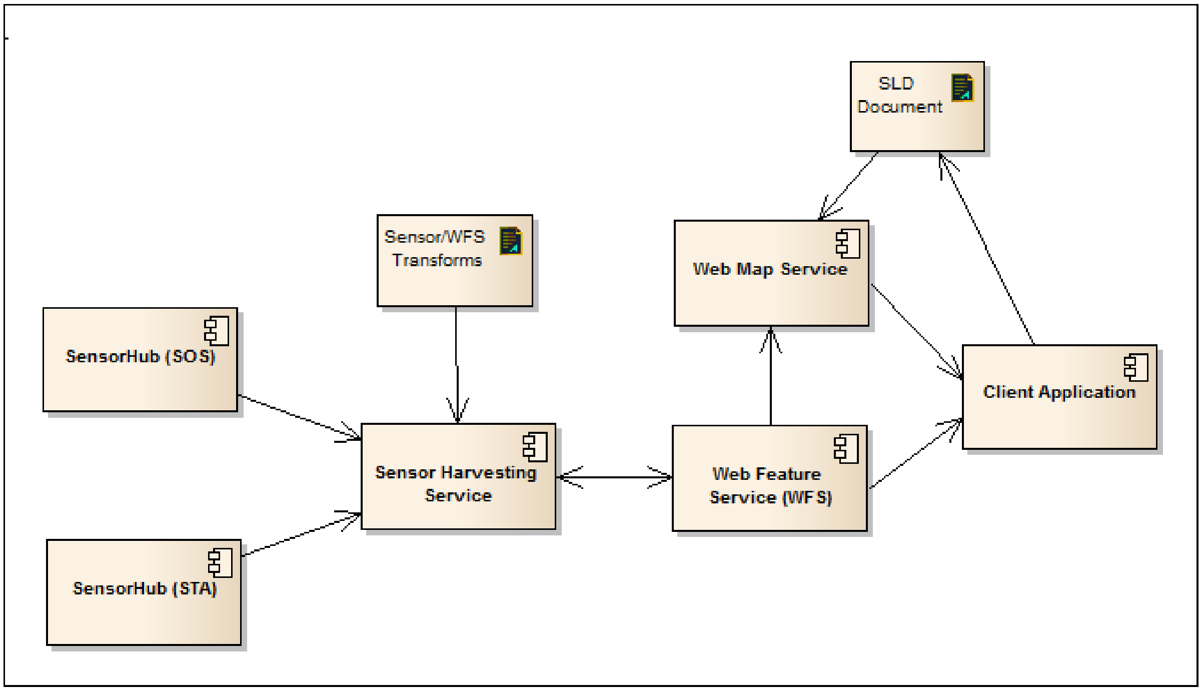
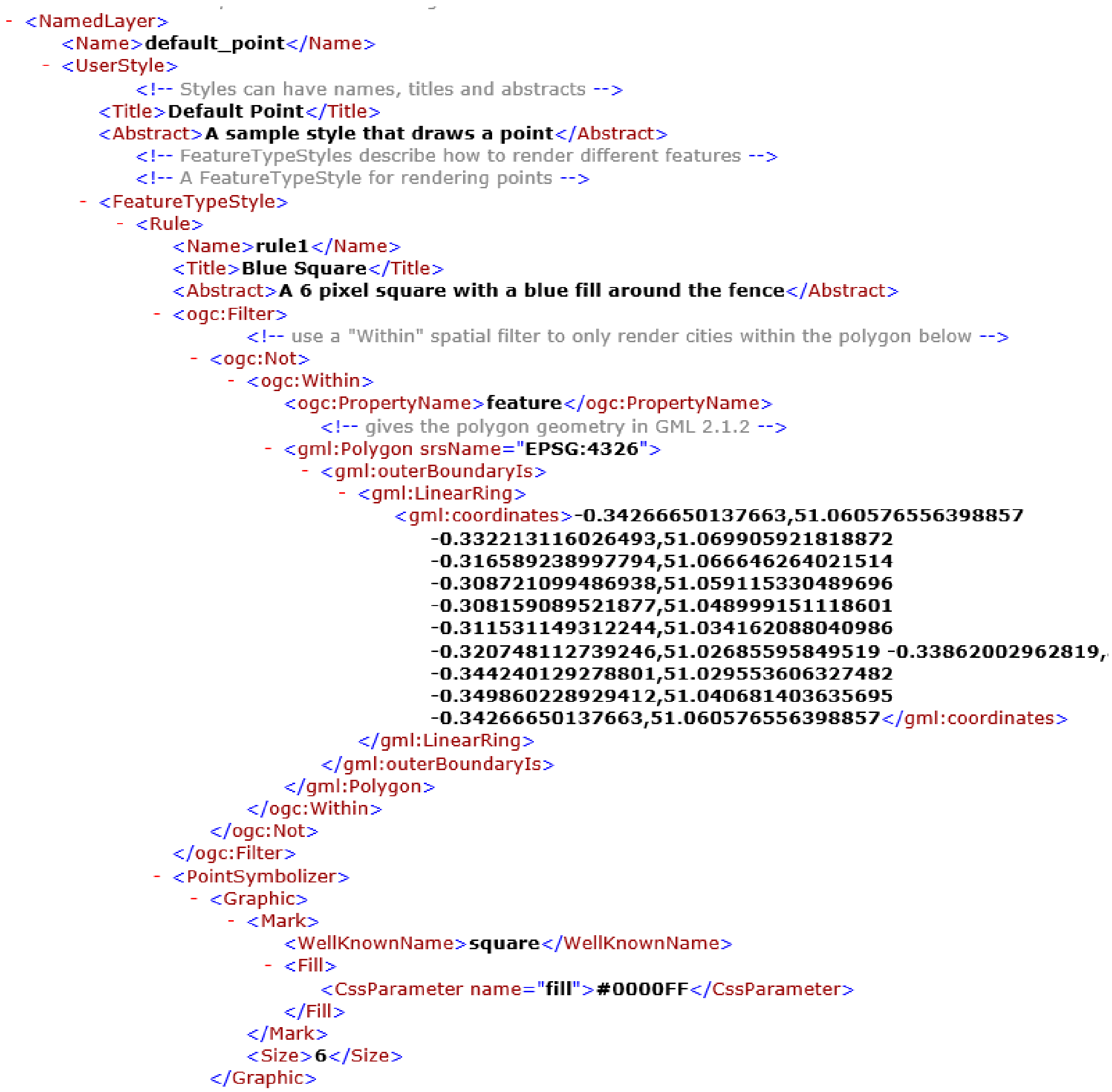


Figure 5 – S-Hub/SOS Harvesting Architecture

Mapping from STA/SOS to WFS requires some careful decisions depending on the content being mapped.

The SLD map styling language allows significant flexibility, including styling based on intersection of other features in different feature classes. This feature allows Envitia to support a geo-fence capability. An SLD can define one or more 'Geo-fence Polygons.' Any object situated within one of the polygons can then be highlighted on the map (for example, in a different color or by making it larger or flashing).



```
- <NamedLayer>
    <Name>default_point</Name>
  - <UserStyle>
        <!-- Styles can have names, titles and abstracts -->
      <Title>Default Point</Title>
      <Abstract>A sample style that draws a point</Abstract>
        <!-- FeatureTypeStyles describe how to render different features -->
        <!-- A FeatureTypeStyle for rendering points -->
    - <FeatureTypeStyle>
      - <Rule>
          <Name>rule1</Name>
          <Title>Blue Square</Title>
          <Abstract>A 6 pixel square with a blue fill around the fence</Abstract>
        - <ogc:Filter>
              <!-- use a "Within" spatial filter to only render cities within the polygon below -->
          - <ogc:Not>
            - <ogc:Within>
                <ogc:PropertyName>feature</ogc:PropertyName>
                  <!-- gives the polygon geometry in GML 2.1.2 -->
              - <gml:Polygon srsName="EPSG:4326">
                - <gml:outerBoundaryIs>
                  - <gml:LinearRing>
                      <gml:coordinates>-0.34266650137663,51.060576556398857
                        -0.332213116026493,51.069905921818872
                        -0.316589238997794,51.066646264021514
                        -0.30872109948693 8,51.059115330489696
                        -0.308159089521877,51.048999151118601
                        -0.311531149312244,51.034162088040986
                        -0.320748112739246,51.02685595849519 -0.33862002962819,
                        -0.344240129278801,51.029553606327482
                        -0.349860228929412,51.040681403635695
                        -0.34266650137663,51.060576556398857</gml:coordinates>
                    </gml:LinearRing>
                  </gml:outerBoundaryIs>
                </gml:Polygon>
              </ogc:Within>
            </ogc:Not>
          </ogc:Filter>
        - <PointSymbolizer>
          - <Graphic>
            - <Mark>
                <WellKnownName>square</WellKnownName>
              - <Fill>
                  <CssParameter name="fill">#0000FF</CssParameter>
                </Fill>
              </Mark>
              <Size>6</Size>
            </Graphic>
```

Figure 6 - SLD used to generate GeoFence

The result is visualized in the Envitia Horizon GeoPortal as shown below.

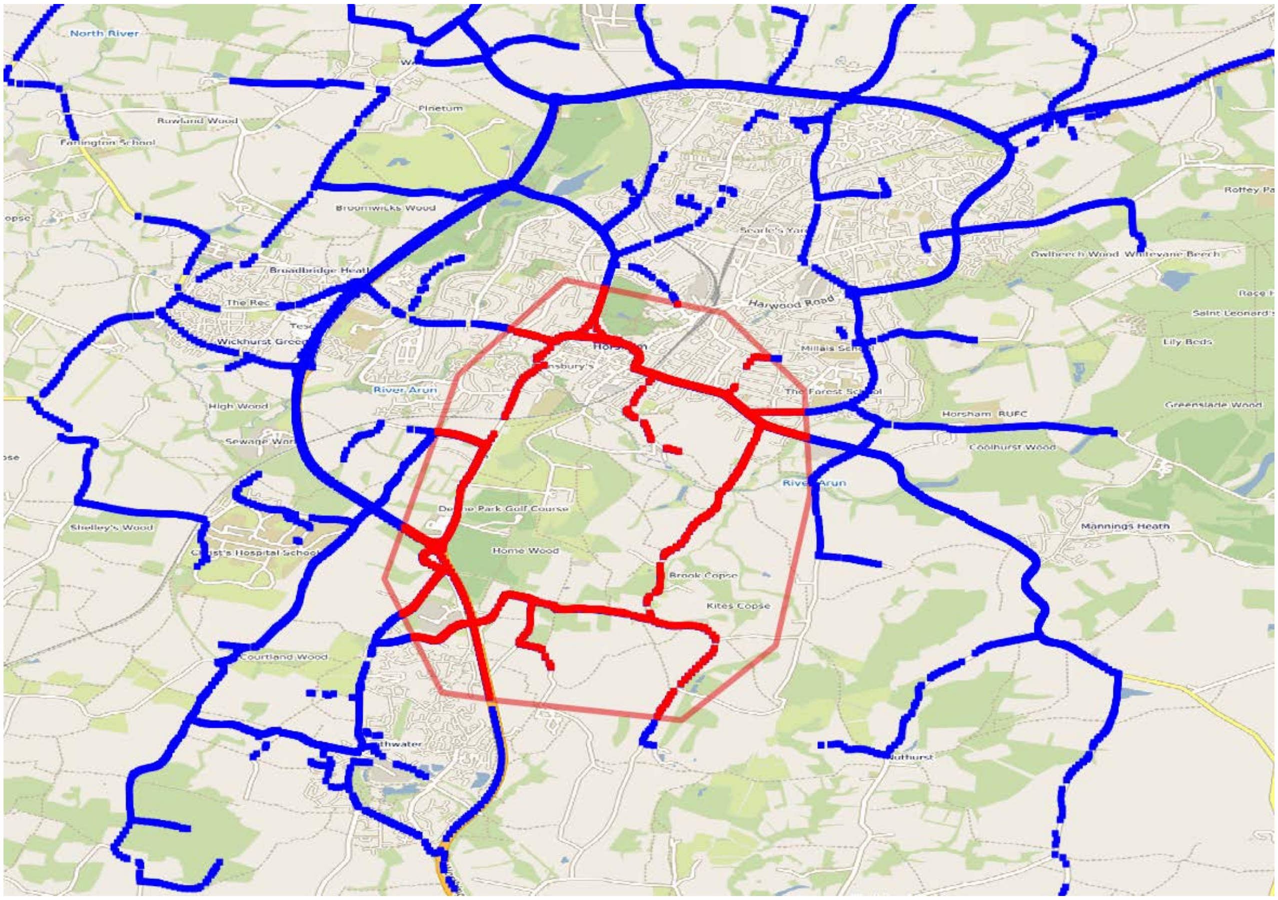


Figure 7 – GPS track points from S-Hub services via WFS, styled with SLD

7.9 Sensor Things API Service Visualization

A number of additional sensor visualization tools were used in the Pilot Extension, in addition to those demonstrated in the initial Pilot. For example, SensorUp demonstrated an extended browser based client able to visualize observations from STA instances provided by either Compusult or SensorUp. Other examples are shown in Figures 8-10.

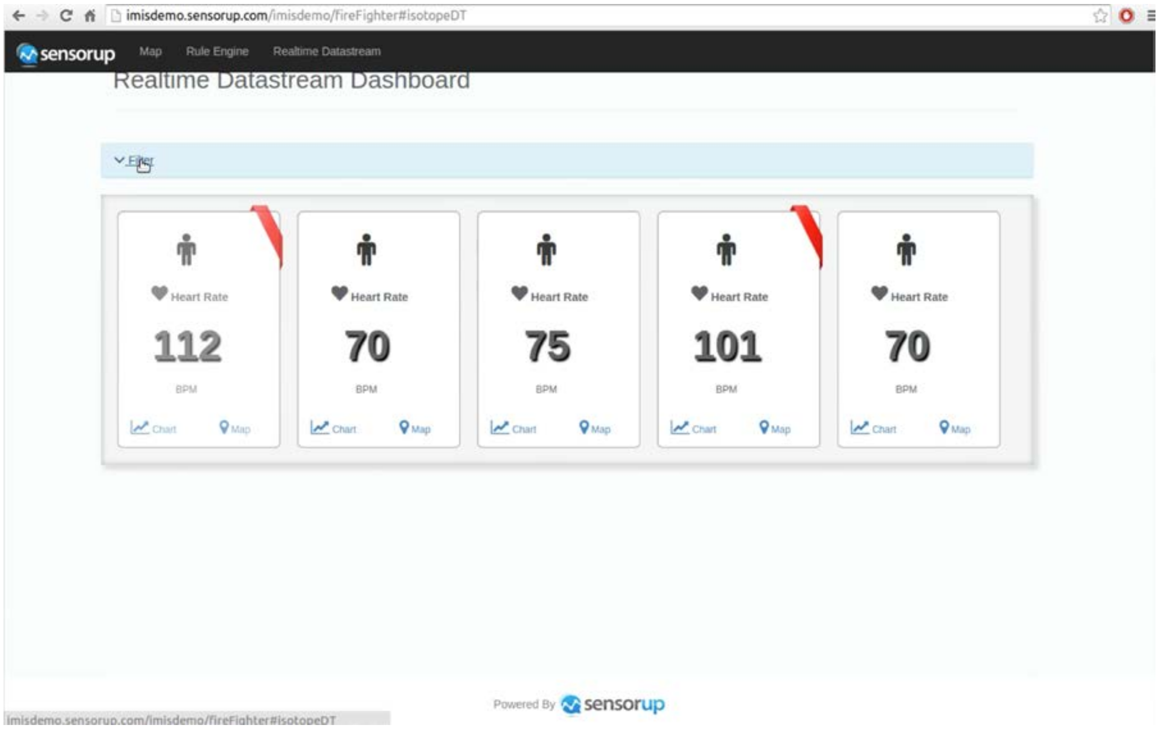


Figure 8 - Visualization of responder vital signs from multiple sensors

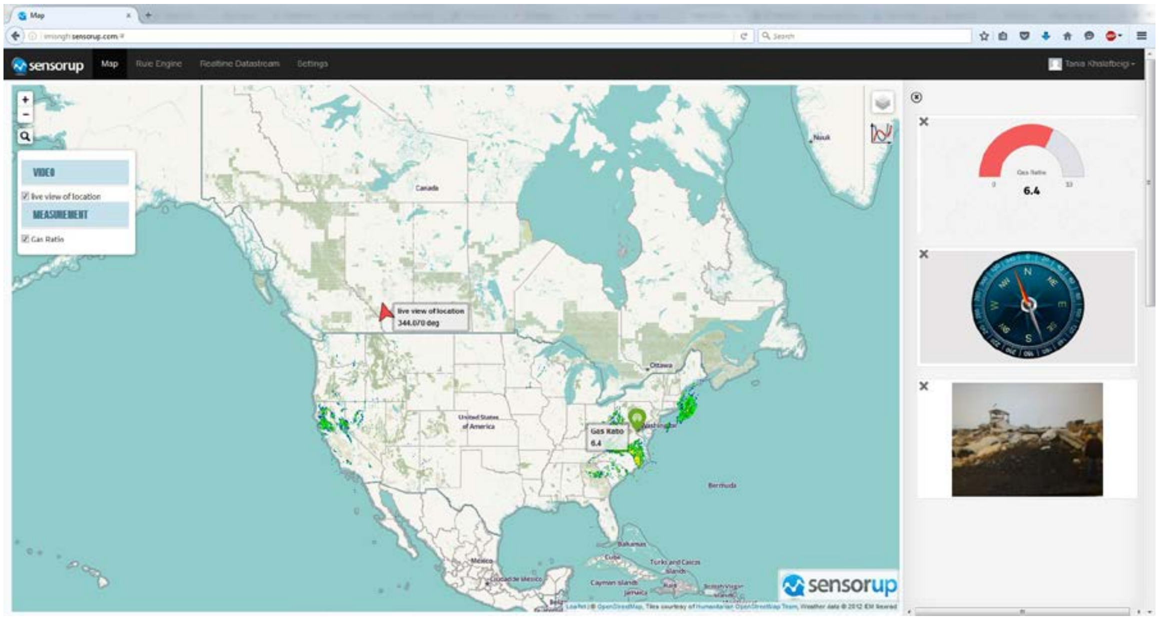
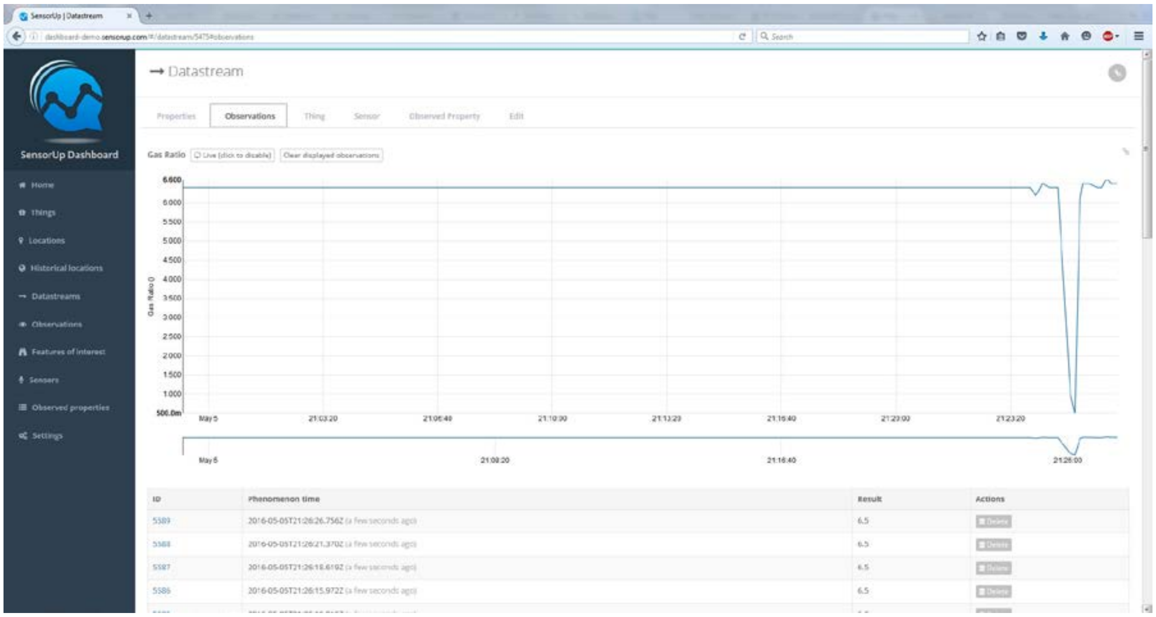


Figure 9 - Visualization of multiple sensors and types



ID	Phenomenon time	Result	Actions
5589	2016-05-05T21:26:26.756Z (a few seconds ago)	6.5	Delete
5588	2016-05-05T21:26:21.370Z (a few seconds ago)	6.5	Delete
5587	2016-05-05T21:26:18.619Z (a few seconds ago)	6.5	Delete
5586	2016-05-05T21:26:15.972Z (a few seconds ago)	6.5	Delete

Figure 10 – Chemical detector sensor time series display

7.9.1 S-Hub / Sensor Client Interoperability

A key goal of the IoT Pilot and Pilot Extension was to demonstrate how standards could contribute to sensor integration without stovepipes and attendant need for a specific application or display for each sensor system. The general case of Sensor/Display system interoperability is shown in Table 2 below.

Table 2 – Typical Interoperability of Sensor System (Clients and Servers)

	Clients	Supplier 1 Client	Supplier 2 Client	Supplier 2 Mobile Client	Supplier 3 Client	Supplier 3 Watch	Supplier 4 Desktop	Supplier 4 Mobile Client	Supplier 5 Mobile Client
Sensor Hubs									
Supplier 1	Custom	Yes		No	No	No	No	No	No
Supplier 2	Custom	No	Yes	Yes	No	No	No	No	No
Supplier 3	Custom	No	No	No	Yes	Yes	No	No	No

Clients in stovepipe systems typically only interact with their own sensor platforms and the opportunity for external clients to interact those platforms is limited. Implementers of more general clients may need to implement a different interface for almost every sensor and may not have access at all to those with proprietary interfaces. This contrasts strongly with the interoperability and interchangeability achieved in the IoT Pilot Extension, as shown in Table 3. There is even more room to improve, for example, with improvements to semantic sensor descriptions.

Table 3 – Actual interoperability of Components in IMIS IoT Experimentation

	Clients	Botts Innovative Deskktop	Compusult Desktop	Compusult Mobile	Sensor Up Desktop	Sensor Up Android Watch	Envitia Desktop	Envitia Mobile	Noblis Apple Watch
Sensor Hubs									
Botts Innovative	SOS	Yes	Yes	Yes	Pending	Pending	Yes	Yes	Pending
Compusult	SOS/STA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Sensor Up	SOS/STA MQTT	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

While there are a small number of interoperability gaps (certain clients and servers only implementing a subset of the standard), in practice most clients could read all data from both the field and cloud S-Hubs. This is a dramatically more positive interoperability level than most existing sensor systems. Note that for each of the vendor S-Hubs, many sensor types have been integrated, providing access to over 100 different types of sensors in all key clients.

A minor issue of interoperability still to be resolved is due to there being two standards for observation access: SOS and STA. There are good reasons for both to be available. For example, STA clients are very easy to implement in browser applications. Full interoperability, however, will require that S-Hubs support both standards and make all data available through both interfaces. In general, it is preferable that servers support as much versatility as possible since:

1. There are typically more clients than servers, and servers are more powerful with less operational limitations than clients.

2. Clients often already have to deal with many interfacing standards (for sources such as mapping, weather data, etc.), so minimizing client development makes adoption easier.

3. Clients often have more limitations. (Browser based clients struggle with SOS as its default format is XML, but processing clients are able to leverage the more

detailed information and the wider capabilities available from a SOS. When every client has to implement both standards to access needed data, that tends to neutralize the benefit of having both standards.

In the end, to ensure interoperability, either standard is chosen, the scope of use of the different standards is defined, or both standards are mandated at server and/or client. Botts Innovative are at present working on a level of support for STA so the above gaps (marked as Pending) are expected to vanish in the near term.

7.10 Communications Exploitation

7.10.1 Communications Infrastructure

As stated above, the scenario was implemented at two sites with two teams: one team deployed at the Lorton site and the other deployed at the simulated command center at the DHS Vermont Avenue offices. The two sites were connected using a number of voice, video and data communications paths, none of which depended on existing DHS or first responder specific communications infrastructure.

The communications infrastructure used for the IoT Pilot Extension is shown in Figure 11. It is a hybrid environment and has a piecemeal quality because planned NGFR Communications Hubs (C-Hubs) are not yet available. It was able nonetheless to demonstrate the value of multiple communications options.

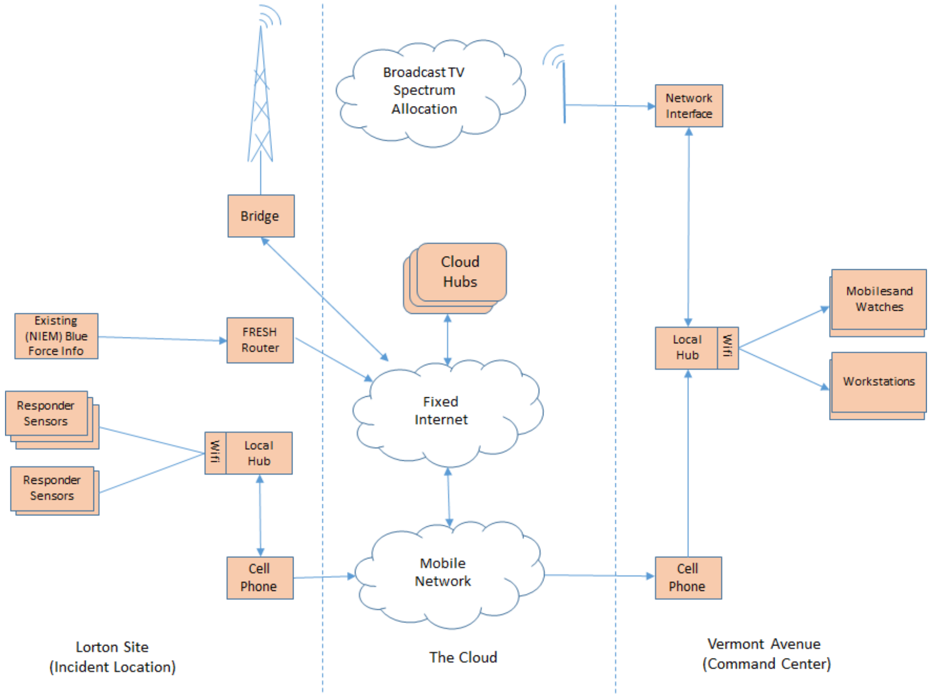


Figure 11 - Communications Infrastructure for the Pilot Extension

Connectivity at the Lorton Facility relied on a Wi-Fi network established by an OSH-based field S-Hub through which all local sensors received commands and transmitted observations. Communication between the field S-Hub and the cloud S-Hubs depended on AT&T and/or Verizon hosted LTE data networks. Over these communication channels were transmitted location/orientation, heartbeat measurements and high-definition (HD) video. In addition, transmission of real-time video streams from cloud S-Hubs to datacasting hosts used standard internet infrastructure.

7.10.2 DHS Infrastructural Comms

The initial IoT Pilot demonstration experience was that security constraints on the DHS network (on users, ports, etc.) were likely to be too severe to be useable, so the Pilot Extension demo did not attempt to leverage the DHS network. The only potential use of the DHS communications was the transmission of DHS EDXL DE messages via the FRESH router, but this was done via a public internet connection rather than within the DHS network. In an operational environment, a devoted C-Hub communications platform would presumably be available and its absence did not greatly hamper the demonstration, although in one case a data quota exceedance more or less completely disabled one mobile device–a lesson for mobile data provision to first responders. The overall takeaway is that reliable data communications are essential, and moving to identify bandwidth and infrastructure requirements (security and access control, etc.) to support incident management and emergency response is critical.

7.10.3 Deployed LANs

Multiple sensors were connected to local or field S-Hubs, which themselves connected to the public Internet using standard cellular mobile devices. Adequate bandwidth was available and two suppliers were alternately used, but there were data allowance and throughput issues that killed several of the video feeds during demonstration performances.

Sufficient bandwidth to deliver multiple sensor feeds while not locking out other important sensor data transmissions is a critical issue if every responder is going to transmit video/audio continuously. At the same time, future work should touch on a capability to prioritize transmissions so that the most critical piece of information or level of detail gets through first.

7.10.4 Command Center LAN

The command center LAN was a wireless network set up in the command center at Vermont Avenue, connected via Internet sharing to a tethered AT&T iPhone.

7.10.5 Datacasting Network

The datacasting network provided by SpectraRep supported the broadcast of video streams delivered from Botts Innovative and Compusult sensors at Lorton through cloud S-Hubs, the SpectraRep infrastructure, and National Public Radio (NPR) broadcasting facilities.

Datacasting enables existing broadcast television signals to deliver encrypted and targetable Internet Protocol (IP) data packets. Like traditional television content, it is natively one-to-many; unlike traditional television programing, datacast content can be targeted to specific recipients or groups of recipients by means of selective encryption.

Broadcast television in the United States operates on a 6MHz channel using licensed spectrum. The ATSC standard supports delivery of 19.39Mbps of MPEG transport in that 6 MHz channel. Television stations are free to allocate the MPEG transport as they like, typically dedicating a portion to a High Definition program stream and multiple lower quality standard definition streams.

When datacasting is deployed, a portion of the MPEG stream is dedicated to this service, typically 1Mbps. Equipment is installed at the TV station to encapsulate IP packets into the MPEG transport payload. Since the MPEG payload packet is the same format as all of the other MPEG packets being delivered to the transmitter, they pass through to air. Since the IP payload cannot be processed by traditional television sets, it is ignored. A special receiver connected to a computer is used to pass the received program stream to software that de-encapsulates the encrypted IP packet stream, revealing the IP payload. Datacasting software determines if that receiver is on the targeting list, and allows decryption if appropriate. Content remains encrypted and inaccessible on non-targeted receivers.

The television broadcast stream is one-way User Datagram Protocol (UDP), so file delivery is accommodated using forward error correction (FEC) and carousel ling, allowing the software to reconstruct or pick up dropped packets on a subsequent pass. The television delivery network is also natively broadcast multicast (one-to-many) so that the number of users scales infinitely. Essentially, anyone able to receive the television signal can be targeted to allow access to the datacast content without requiring any additional spectrum or resources.

Because the transmission infrastructure and licensed spectrum is already in place and operating, datacasting is very cost effective to deploy requiring only the addition of equipment to support this service. Ingest paths must be designed and implemented, content management and other datacast roles addressed, but the highly resilient television delivery network is already operating, maintained by professional engineers, has backup generators and other systems to assure reliable content delivery.

A new broadcast standard to be adopted soon in North America and South Korea supports new capabilities like direct reception on portable devices, deeper building penetration, native IP support, integration with other IP networks and more. As broadcasters move to this new transmission architecture, all of those benefits will inure to datacast content and users.

Datacasting was used in the Pilot Extension to deliver data from Lorton-based camera systems to the Command Center at Vermont Avenue. While the link from Lorton to the PSCloud S-Hub depended on moderate bandwidth mobile links, once video was received by the Datacasting Center, it could be delivered to multiple users (at both Lorton and Vermont Avenue) with no extra network bandwidth use. This applied to both video (Botts Innovative) and sensor information (Compusult).

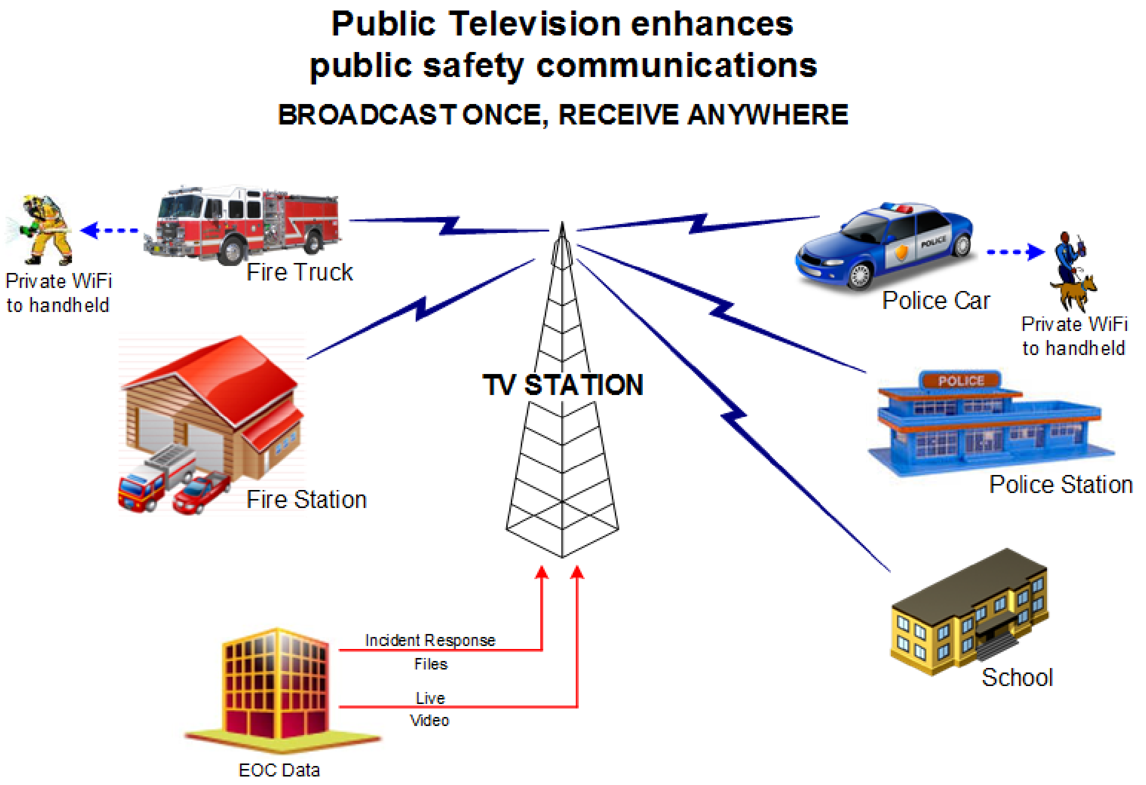


Figure 12 - Datacasting Concept for First Responders

7.10.6 Issues with Communications

Two main issues existed with the communications technologies used in the experiment. First, they were based on consumer grade mobile technology, limited in bandwidth and relatively fragile. There is also an issue of scalable bandwidth, particularly if many video streams are needed to characterize and monitor a large scale incident scene. Although datacasting does solve some of the issues, it still does not deal with the local uplink bandwidth issue. Satellite uplinks can provide relatively high bandwidth, but with high cost and latency issues.

A dedicated NGFR Comms Hub able to manage prioritized transmissions and use multiple IP networks (c.f. Google Project Fi) may be one part of the solution to first responder data communications challenges. Another part may lie in wireless network infrastructure, such as FirstNet, that is dedicated or at least prioritized for public safety operations. New transmission models such as datacasting and caching strategies such as variable level of detail may also play a role.

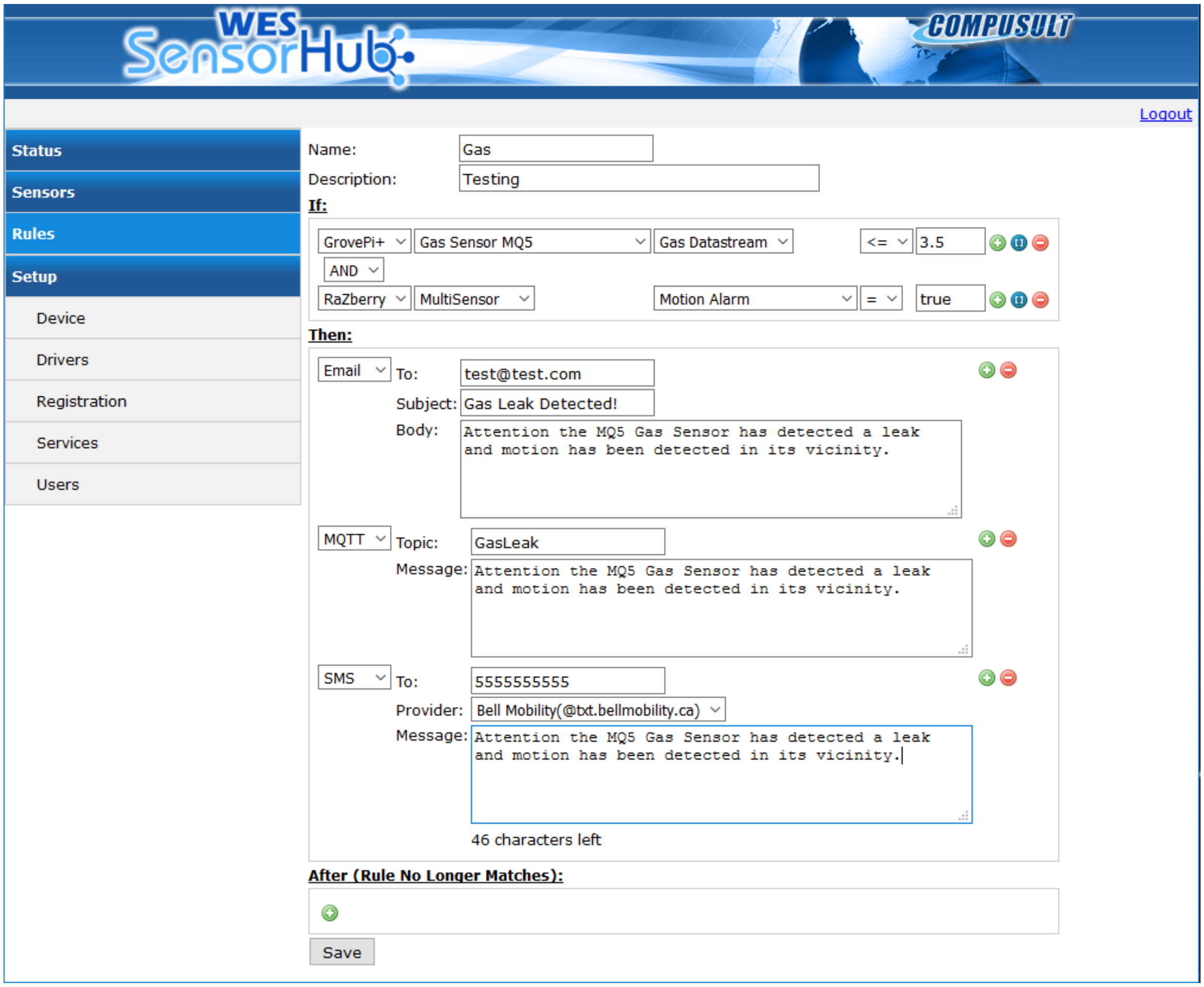
7.11 Alerting

Responders already deal with many sources and types of information while carrying out their duties and do not have time to continually monitor a flood of sensor data. Alerts and alert notifications of critical and actionable events, demonstrated effectively in the Pilot Extension, are an important means of contributing to responder awareness and not overload them. Within the Pilot Extension, alerts were generated based on several thresholds. They included a dangerous gas (simulated by an unlit butane lighter) exceeding a critical concentration limit, or a responder entering a zone geofenced as dangerous. Alerted events are usually generated according to one or more threshold criteria, but event processing can occur in a number of different ways and different points in the Sensor-to-Responder pipeline.

7.11.1 Server Side Rules

Compusult S-Hub provides an administrative User Interface to create complex boolean logic rules, that when matched can trigger the S-Hub to perform an action. Actions include tasking devices and sending alerts by a variety of channels, including email, text messages and Message Queue Telemetry Transport (MQTT) topics. Email and text support allows for existing devices without specialized applications to receive the alerts, while MQTT delivers alerts to applications incorporating MQTT clients.

For the Pilot Extension, particular rules were configured to send alerts if a gas leak was detected. A Compusult web client and mobile client used MQTT to receive alerts and show them to the user. The MQTT message was a plain text message displayed to the user. A configuration page used to define alerting rules for the Compusult S-Hub is shown in Figure 13.



Name: Gas
Description: Testing

If:

GrovePi+ | Gas Sensor MQ5 | Gas Datastream | <= | 3.5
AND
RaZberry | MultiSensor | Motion Alarm | = | true

Then:

Email To: test@test.com
Subject: Gas Leak Detected!
Body: Attention the MQ5 Gas Sensor has detected a leak and motion has been detected in its vicinity.

MQTT Topic: GasLeak
Message: Attention the MQ5 Gas Sensor has detected a leak and motion has been detected in its vicinity.

SMS To: 5555555555
Provider: Bell Mobility(@txt.bellmobility.ca)
Message: Attention the MQ5 Gas Sensor has detected a leak and motion has been detected in its vicinity.
46 characters left

After (Rule No Longer Matches):

Figure 13 – Alert configuration for the Compusult S-Hub

7.11.2 Client Side Rules

During the Pilot Extension, both SensorUp and Noblis provided smartwatch (+ smartphone) applications that allowed the user to configure simple rules that when matched against data from an STA service, an alert would appear on the smartwatch. The application by SensorUp took advantage of the Sensor Things MQTT extension to efficiently retrieve pushed alerts, while the Noblis app took a more traditional approach and polled for the data to match against rules. Polling for data, while not as efficient, allows for supporting STA services without the MQTT extension.

Compusult was the only provider of the MQ5 gas sensor used in the demonstration. Compusult's STA service did not have support for the MQTT server extension, but was able to forward data to other STA services as a client. The Compusult S-Hub therefore could be polled for its MQ5 gas sensor readings, but also forwarded them to the SensorUp STA service, which in turn pushed MQTT alerts to the SensorUp watch. As a result, it was possible to deliver MQTT notifications from the Compusult sensor to applications such as the Noblis Apple Watch application even though Compusult did not support MQTT by using the SensorUp S-Hub as a proxy.

The forwarding mechanism provided by Compusult illustrates the way in which an update of a cloud S-Hub from a field or local S-Hub could work, with clients then subscribing to the cloud MQTT service without needing to overload the field S-Hub with requests or subscriptions.

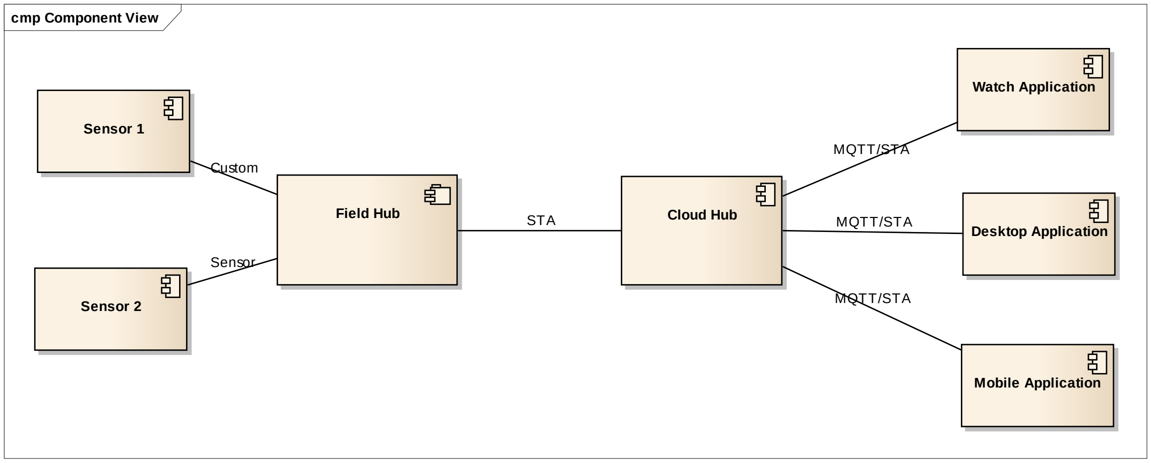


Figure 14 – Hub Update Sequence

Figure 14 shows the eventing architecture implemented by Compusult and SensorUp with Noblis, Compusult and SensorUp clients all displaying the same alert.

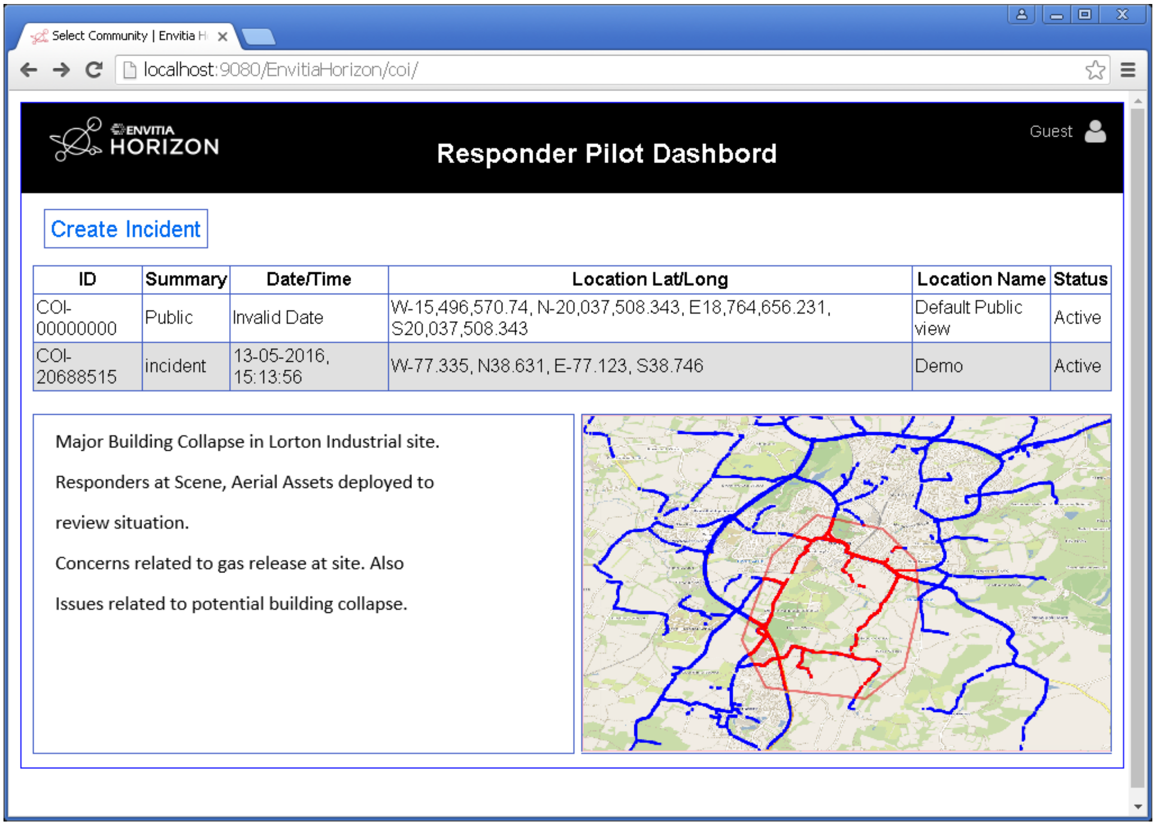
7.12 Context Catalogue and Event Workspaces

There has been a very significant growth in the volume of information that could be available from IoT sensors to contribute to first responder situational awareness. While first responders need just the right information at the right time, Command Center analysts and experts will likely benefit from being able to search and sift through as much of this information as possible, get the results in front of decision makers, and then get critical bits out to responders to reduce the time and uncertainty of the response.

While a central catalog of data and services is a powerful means for an analyst to find and access relevant information, personalized views of information saved as Context documents on an incident webpage is a mechanism by which analysts can then deliver just the right information in just the right way to responders and commanders alike. Conext views can be targeted equally well to desktop, Command Center, field and mobile users. Customized and persistent contextualized views provide a significant benefit in ensuring visibility of the most important information for any particular role amid a deluge of sensor data.

Within the initial IoT Pilot, Envitia demonstrated the use of the OWS Context document standard to define a series of views with a defined area of interest and set of references to specific data layers (background maps, imagery, sensor layers, annotation etc.). Each document was stored in a catalogue/registry for specific roles in specific incidents to provide each responder with rapid, easy access to critical and up-to-date information.

For the IoT Pilot Extension, Envitia created a catalog component called an ebRIM Registry Extension Package (eREP) to be loaded into an OGC Compliant CSW-ebRIM catalog in support of incident management information sharing. The IMIS eREP adds catalog record types describing an incident, providing the basic information needed for response, and including the ability to manage a set of Context views of the incident. The capability to create one of these record groupings around an incident and add Context views was implemented in the Envitia Horizon Portal applicatoin. This provided an example of basic incident creation, as well as a view of creation and management.


Select Community | Envitia H
localhost:9080/EnvitiaHorizon/coi/
ENVITIA
HORIZON
Responder Pilot Dashbord
Guest
Create Incident

ID	Summary	Date/Time	Location Lat/Long	Location Name	Status
COI-00000000	Public	Invalid Date	W-15,496,570.74, N-20,037,508.343, E18,764,656.231, S20,037,508.343	Default Public view	Active
COI-20688515	incident	13-05-2016, 15:13:56	W-77.335, N38.631, E-77.123, S38.746	Demo	Active

Major Building Collapse in Lorton Industrial site.

Responders at Scene, Aerial Assets deployed to

review situation.

Concerns related to gas release at site. Also

Issues related to potential building collapse.

Figure 15 - Envitia Incident Registry Display

One view is declared as the ‘overall scenario’ view and this is shown alongside the incident summary when the dispatcher clicks on the incident. Opening the incident then displays a set of views representing the situation. These views access the various sensor and background services displaying relevant incident information as required.

8 Future Work

The work undertaken within the IMIS IoT Pilot and Pilot Extension has progressed the capability to deliver open IoT- and SWE-based systems that can provide real benefits at manageable cost to the first responder community. There were also a number of areas identified during the course of the activities, which would benefit from further work to address specific challenges, as well as mature the technologies involved and move them closer to operational deployment.

8.1.1 Detailed Reference Architecture Development

The design work undertaken so far has led to identification of some elements of a reference architecture that would support the development of truly flexible and interoperable NGFR systems. An initial NGFR IoT Reference Model document (Ref [6]) attempts to capture this architecture. With further implementation and outreach experience, a Reference Model can serve as a key guide to development of a first responder product ecosystem. Work on the Model can best progress in parallel with development activity so that it is grounded in and informs the direction of that activity.

8.1.2 Key Component Profiling

In order to mature an OGC SWE-IoT architecture, it is necessary to provide clear, domain-focused implementation guidance for components, and particularly for interfaces. Focusing on open standards as opposed to developing domain-specific standards is key to maximizing the use of existing COTS/OSS technology. Custom standards hinder more than they help, but there is value in developing domain profiles of general standards that call out specific capabilities and qualify them, restricting or extending them within the scope of the standard. A first step in this direction was made by the OGC 15-118 Profiles ER. Advancing these and similar profiles would be a valuable next step, particularly those that more completely characterize the canonical capabilities of S-Hubs, HubCat catalogs, those involved in synchronization processes, and those supporting processing and visualization capabilities. Event handling is key to a fully aware first responder, so this also is a key area for standardization work.

8.1.3 Persistent Test Environment

At least some of the Pilot infrastructure has operated successfully over a period of approximately one year. A mixture of a cloud deployments and locally available laptops, phones and sensors has provided a useful, nearly continuously available experimentation platform.

The next phase of work should further develop and persist this NGFR IoT platform to deal with analysis, visualization, sensor tasking and collaboration technologies. The objective is to progress towards the NGFR goal of a better connected, protected and fully aware responder, but also to model the product ecosystem that could eventually be made up of commercially available products for the public safety community. In order to ensure maximum value from potential exploiter technologies, a persistent sensor platform, with previously collected sensor data and simulated and taskable sensors, would provide a low threshold to new innovation and education. An internet-based platform would mean easy access from potential technology providers maximizing the NGFR goal of having a broad set of technology options for implementation. The platform would also be useful for experiments and demonstrations focusing on usability and the human computer interface, as well as supporting engagement and promotion of what is learned.

8.1.4 Formal Sensor Characterisation

The Pilot and Pilot Extension activities implemented a wide range of deployable, wearable, drivable, fly-able, and even virtual sensors. In order to achieve an agile and interoperable sensor information environment, levels of in-domain and inter-domain interoperability were needed. Two categories of sensors will be exploited in NGFR as it evolves: official sensors and citizen sensors. Authoritative or official sensors will be those provided to authorized responders by responsible governmental organizations. A key example will be standard sensor ensembles worn by a first responder. Initial follow-on work identifies 5 key categories of wearable sensors:

- Physiological and environmental sensors;
- Audio and video recording;
- Location tracking (using GPS other technologies); and
- Power management and status.

There will also be a range of sensors attached to deployed assets, such as:

- UAS sensors (e.g., video, temperature); and
- Ad hoc in situ sensors (e.g., autonomous proximity sensors).

Sensors and devices will also be deployed by citizens, at a minimum as part of their smartphones and other devices, and may be made available to responders in emergency situations. That could be a lot of varied sensors and observations, resulting in a mess of incommensurate takes on a given situation.

The sensors and standard sensor platforms need to be clearly identified and classified so that the sensor components from whatever source are interchangeable and their outputs are convertible from one to another. Whatever the origin of a responder and their technologies (imagine a cross-state event with a different technology supplier providing sensor information for each state), the infrastructure must support common interpretation of the information. You need not just be able to read the information (i.e., temperature of 37.5 °C), but also know it is an external, skin or body temperature. Equally, it is important to understand the phenomena being measured and units of the results. A register of properties characterizing sensor or sensor platform classes allows clients to understand what they might encounter and to interpret information correctly. This is also helpful for less authoritative data, for example exploiting building sensors not provided by public safety organizations. The use of a wide range of sensors can provide valuable insight, but more so if they are well characterized. Lack of standard sensor characterizations in the Pilot really interfered with useful observation processing, such as aggregation and change detection, since human involvement was needed to understand how to interpret each sensor correctly. Further work is therefore recommended in the area of 'self-description' where sensors identify themselves to sensor and sensor property registry elements. This offers real value in semantic sensor interoperability and effective interpretation of large volumes of sensor data.

8.1.5 Identity Management and Access Control

Security will be a key element of future work on IoT technologies for incident management. Any approach to identity management and access control within the overall NGFR architecture will need careful consideration. This includes consideration of the practicality of deployment using existing identity management/access control infrastructure within DHS and its collaborators, and the potential use of technologies aimed at broader community access control, for example the SAML/XACML standards being adopted widely in commercial and more recently military environments. The implications of the chosen infrastructure on components, such as low-power sensors and low-bandwidth communications protocols, requires considerable investigation and experimentation. The architecture must also maximize the use of both open and authoritative sources (a process begun with the FRESH Router/SOS/STA integration activity) and be able to deliver to a broad community, including other government organizations and potentially non-governmental organizations (NGO) and the public in a controlled way.

8.1.6 Extended Datacasting

Datacasting was a successful capability within the Pilot Extension. As an IP-packet-based transmission system, it has the potential to be a much more widely used capability if other client applications and types of data can make use of it. One possibility still to be investigated involves connecting a datacasting receiver directly to a field S-Hub so that all local clients, mobile or otherwise, can leverage what is received.

8.1.7 Analysis and Notification

Alerting and notification played a key role in the Pilot Extension. It could play a much larger role in an NGFR architecture if it were more of an integral part of all of the included services. For example, an alerting capability could be combined with geofencing rules. In general, there is merit in continuing to evaluate more event processing and analysis options possible now that a basic NGFR IoT architecture has been developed.

8.1.8 Integrated User Interfaces

Section 7.10.1 (S-Hub / Sensor Client Interoperability) outlined the high level of interoperability achieved between different clients and S-Hubs, as well as other components. But these clients, despite being in most cases web based, are mostly independent. The Pilot Extension highlighted a need to provide a flexible way of assembling multiple views of information to meet specific requirements. One aspect of this is being able to deliver a ‘wall of information’ for overall situational assessment and briefing. This needs to be a subject for further experimentation. In the run-up to the demonstration, the need to easily configure a ‘Wall Information Display’ for critical users became clear. This requires a facility to be able to lay out a series of application pages and drill through them as required. In the event, this was done in a very limited way by simply arranging the applications on screen, but further thought is clearly needed in this area.

Another issue identified was that the diversity of client applications led to a heterogeneous user experience. An integrated client was not in scope as the focus was primarily on sensor platform issues, hence existing clients were used. While interoperability is generally served by having a choice of applications, some users noted the learning curve involved in working with what seemed like a different application for every task. Without stifling innovation, there is clearly some user experience standardization that could be done so that using a new responder app is more like getting into a new car than learning a new language.

8.1.9 Virtual and Augmented Reality

Virtual Reality (VR) and Augmented Reality (AR) offer a new paradigm for user interfaces, allowing immersion into and visual organization of massive amounts of information. AR can benefit first responders through delivery of information in context. It should not be too invasive if it can be integrated into already needed protection gear and does not interfere with vision. Critical issues such as technology compactness, reliability and operation in difficult situations (e.g., where external light level varies from very dark to extreme brightness) need to be considered. Experimentation on functional value versus cognitive loading on responders is important to assess the real value of progressing with such a technology. AR/VR can also be valuable in command center situations, allowing virtual wallboards or bird-tables to be displayed, showing the overall situation or the specific environment inside a building with information from a range of sensors. It has already been used to simulate incident scenes for operation rehearsal. Both of these techniques are rapidly evolving from a hardware and software perspective and have reached the point where on-the-ground experimentation could be fruitful.

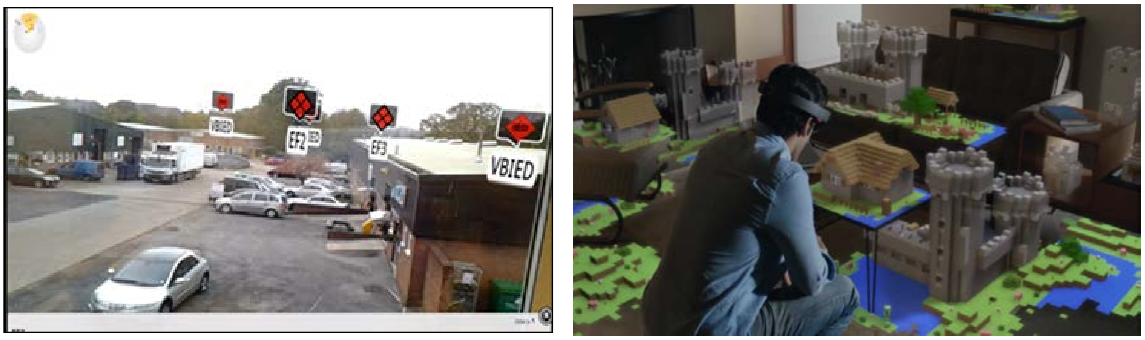


Figure 16 - AR Concept Examples (Deployed Responder and Command Center User)

Screen Shots, Copyright Envitia MapLink Pro ™ and Microsoft HoloLens™

9 Summary and Conclusions

9.1 Success of a Demonstration Across Two Sites

The Pilot Extension two-site approach (three if the simulated sensors used at Vermont Avenue are included) provided significant value in vetting both designs and implementations in a more realistic context. It provided a test of remote alerting capabilities for all responders. It demonstrated some value and also limitations of video transmission and distribution. A significant benefit was the variety of new sensor drivers added to the S-Hub implementations, and the number of sensor types integrated into the IoT infrastructure. The activity also showed how a tiered architecture of S-Hubs (field and cloud) could maximize access and minimize bandwidth issues, and cache critical information. Lastly, Datacasting was a promising element in providing additional communications capacity.

9.2 Further Demonstration of IMIS IoT Architecture Viability

The Pilot Extension demonstrated that the S-Hub / Sensor Catalogue / Sensor Services / Client Applications model is an effective mechanism to provide IoT-style agility and avoid lock-in of proprietary sensor architectures. In a number of cases it was even possible to set up ad-hoc chains of services quickly to address identified limitations (for example, using the SensorUp STA service to provide notification capabilities for the Compusult STA service). All implementations have limitations, but the open standards environment established in this work has shown to be very flexible and resilient as situations and technologies change.

Also worthy of note is that the participant organizations in both IMIS IoT and IoT Extension worked face-to-face only on three occasions in each experiment: first at the kickoff, and then for two or three days at Lorton/Vermont (set up and demonstration execution). Hence, all integration was possible via Internet-connected services, and configuration could be performed in very short order.

9.3 Working Practices

The Pilot and Pilot Extension both used agile development methodologies. This took the form of a series of relatively short development and testing sprints to implement user stories in working code and hardware. The process worked well, and the prototyping and demonstrations, we believe, went well in terms of raising the level of appreciation for agility and off-the-shelf, bring-your-own-device integration. The benefits of agile methodologies were most clear at development level in this project but other OGC activities have shown a corresponding benefit to directly and iteratively engaging a range of stakeholders throughout the lab-to-market process. The agile, learn-as-you-go approach poses real challenges in environments where a more top-down, plan-then-execute approach is the norm, but those challenges are worth addressing in order to benefit rather than suffer from the increasing pace of technological innovation.

9.4 Parallel Design and Implementation Activities

The Pilot and Pilot Extension activities together were first steps towards a commercial IMIS IoT ecosystem. Further progress depends on two parallel activities.

- An overall ‘blueprint’ generally known as a ‘Reference Architecture’ needs to be established and matured. The blueprint will guide the development of commercially products that are able work together to improve first responder connectedness and awareness in a multi-vendor, multi-organization environment. This can also provide the infrastructural context for the work being undertaken by the NGFR Apex Program to develop a design handbook for first responder on-body sensor and communications gear. The initial Reference Architecture, as implemented in the Pilot Extension activity, specifically addresses information interoperability at the incident and enterprise levels needed to support on-body gear. Both design artifacts should be developed further in a coordinated fashion.

- Regarding further implementation and operations experience, further work remains to be done on standards-based interoperability between devices, but the interoperability between devices and their users also needs improvement. Relevant activities to this end include focused trials, simulations and ‘structured play.’ The latter requires expanding and stabilizing the technical capabilities that have been developed so far in order to focus on usability aspects, for example by creating a persistent and supported exercise environment. A stable and dependable wireless remote communications capability, as well as security capabilities, would also be required.

9.5 Overall Success of the Program

The IMIS IoT Pilot and Pilot Extension evolved from one innovation proposal among many considered by the IMIS Subcommittee (IMIS SC) to persuasive demonstrations that standards-based IoT sensor systems have the potential to serve as backbones of situation awareness for first responders. Both demonstrations were well received by Dr’s Brothers and Griffin and other DHS S&T officials; the entire NGFR team received an S&T Undersecretary’s Award. A follow-on demonstration activity was well attended within the Capital Building. The ultimate success of the program will depend on whether the ideas and innovations so far developed can be translated into commercial products and operational deployments. This will involve both outreach to the first responder and incident management communities by DHS, and further evolution of a supporting body of standard specifications by OGC and other relevant organizations. Time will tell whether both community and commercial stars can become aligned favorably to the particular ecosystem described in this report, but there is no doubt that sensors in some form are in the future of incident management and response.