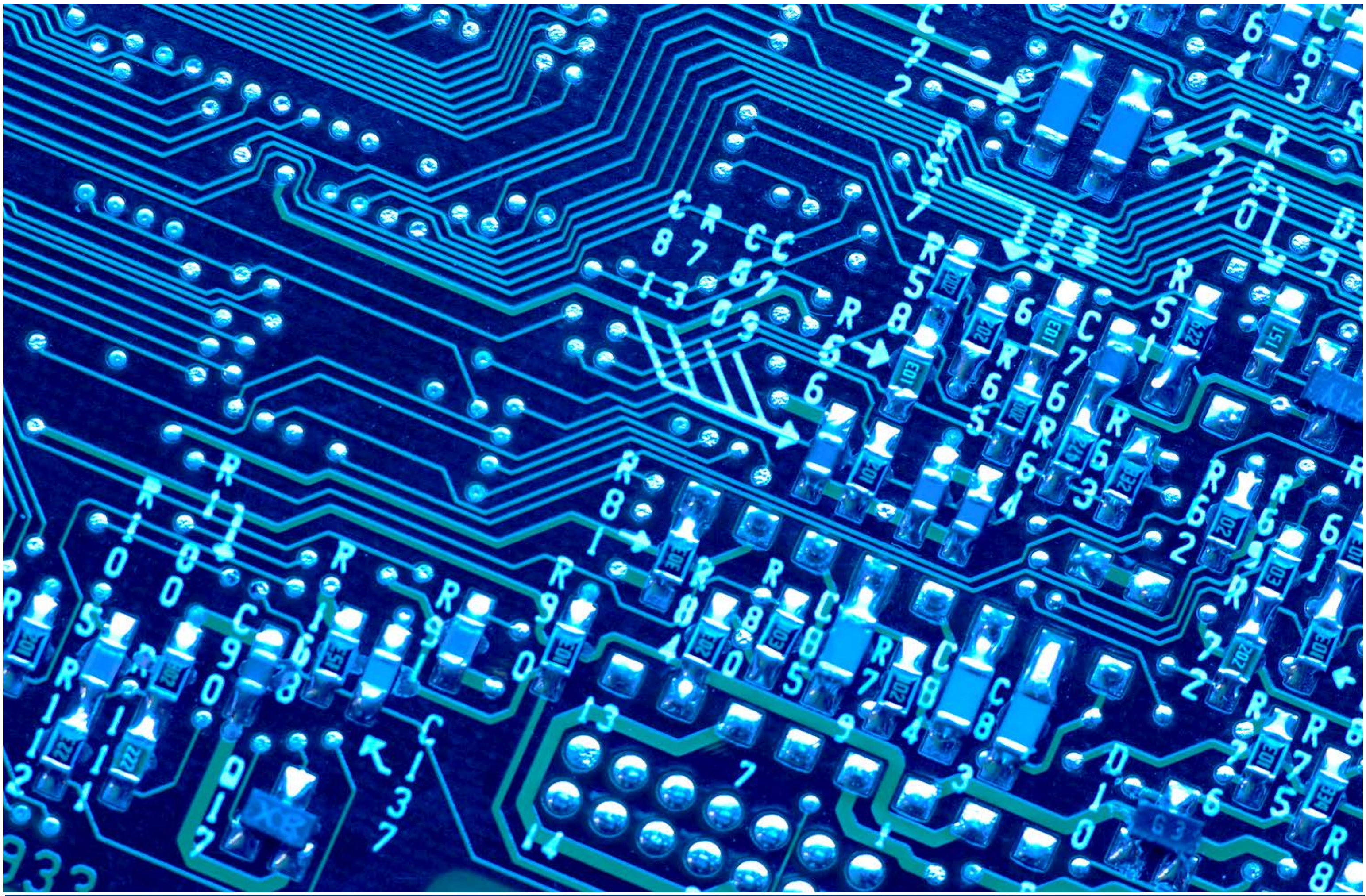

Next Generation First Responder Integration Handbook

Part 2: Engineering Design

Version 3.0 – *August 2018*

Science and Technology Directorate

Homeland Security
Science and Technology

NEXT GENERATION FIRST RESPONDER
PROTECTED, CONNECTED & FULLY AWARE

Disclaimer of Liability

The Next Generation First Responder (NGFR) Integration Handbook (hereinafter the "Handbook") is provided by the Department of Homeland Security (DHS) "as is" with no warranty of any kind, either expressed or implied, including, but not limited to, any warranty of merchantability or fitness for a particular purpose. The Handbook is intended to provide guidance for implementing specific technologies, and does not contain or infer any official requirements, policies, or procedures, nor does it supersede any existing official emergency operations planning guidance or requirements documents. As a condition of the use of the Handbook, the recipient agrees that in no event shall the United States Government or its contractors or subcontractors be liable for any damages, including but not limited to, direct, indirect, special or consequential damages, arising out of, resulting from, or in any way connected to the Handbook or the use of information from the Handbook for any purpose.

DHS does not endorse any commercial product or service referenced in the Handbook, either explicitly or implicitly. Any reference herein to any specific commercial products, processes, or services by service mark, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the United States Government or DHS. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or DHS, and shall not be used for advertising or product endorsement purposes.

Table of Contents

Figures

I. Overview

A. Program Background

The Department of Homeland Security (DHS) Science and Technology Directorate (S&T) launched the Next Generation First Responder (NGFR) Apex program in January 2015 to develop and integrate next-generation technologies to expand first responder mission effectiveness and safety. The NGFR Apex program develops, adapts and integrates cutting-edge technologies using open standards, increasing competition in the first responder technology marketplace and giving responders more options to build the systems they need for their mission and budget.

The NGFR Apex program seeks to help first responders become better protected, connected and fully aware, as described below and in Figure 1:

1. **Protected** – Responders need to be protected against the multiple hazards they encounter in their duties, including protection against projectiles, sharp objects, fire, pathogens, hazardous chemicals, explosions and physical attack.
2. **Connected** – Responders need to be connected with other responders, with incident commanders (IC), and with local, regional, state and federal command centers in order to provide information to and/or receive information from those various entities.
3. **Fully Aware** – Responders and their leadership need to be fully aware of the threats, activities and environment in which they are operating. Responders and their leadership need to be aware of the location of all resources, including both personnel and units.

Figure 1: Goals of the NGFR Apex program

One key component of the NGFR Apex program is that it is both modular—meaning that responders can select different components that will easily integrate via open standards and interfaces—and scalable—meaning that responders can build a large and complex system or a small and streamlined system, depending on their mission needs and budget. To achieve these requirements, the NGFR Apex program developed an architectural model and defined integration standards to ensure that each piece of the system is "swappable."

B. Purpose

This section of the NGFR Integration Handbook provides specific engineering design guidance to assist industry in developing and prototyping hardware and software solutions that fulfill

NGFR Apex program capability gaps. Solutions will be validated and tested by industry vendors, first responders and other stakeholders.

C. Scope

This section identifies the architecture and design of information systems, software subsystems, and hardware or software devices that will need to integrate to Responder SmartHub architecture (Part 3, Appendix J). The design principles, data flows, processing concepts and interface standards will assist industry in developing products that meet this guidance. Wherever possible, this document describes existing standards and practices, and avoids proposing the creation of new information systems or standards.

The NGFR Apex program and modules that comprise the Responder SmartHub architecture are based on operational and technical requirements from the first responder community. These requirements cover the capabilities and functionality responders need to perform their missions to become better protected, connected and fully aware.

II. Responder SmartHub High-Level Specifications

The high-level specifications below provide general guidance in designing and developing prototype solutions for responders. The requirements from which the Handbook was developed, as identified through the Project Responder 4 initiative, are provided in Part 3, Appendix J.

The five main modules of the SmartHub architecture were described in Part 1 of this Handbook. The high-level component architecture and module-to-module connections for Responder SmartHub is provided in Figure 2 as a reminder of that architecture.

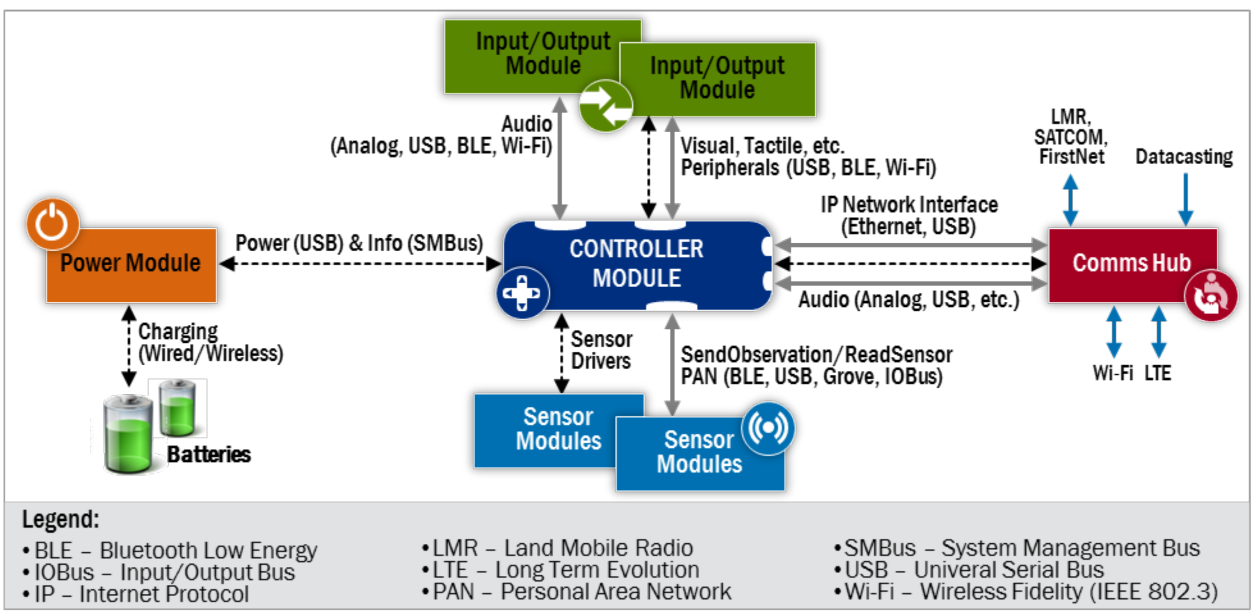

Figure 2: Responder SmartHub Module-to-Module Connection Diagram

A. Control and Information Integration

Systems must be able to interface with each other to receive and exchange information. The Responder SmartHub shall receive information from the responder, process it locally and send the processed information to the appropriate destination to include other responders, a public safety agency or incident command centers.

B. Communications

Communications must bridge voice and data across disparate pathways (e.g., voice over Land Mobile Radio (LMR) to cellular). In addition to integrating with LMR, a responder or agency must be able to identify and prioritize critical communications over routine communications. For example, emergency communications shall be transmitted using the fastest and most reliable pathways, while lower priority data can be transmitted using alternate pathways that may use a store-and-forward process to transmit the information. In the event of a loss of connectivity, information shall be cached locally until the required network regains connectivity. As part of the communications prioritization and caching capabilities, the Responder SmartHub shall automatically re-connect to a network when available and control the transmission of cached sensor data as a lower priority than the current sensor data. The Responder SmartHub shall encrypt all data communications. The Responder SmartHub will allow a responder or agency to configure the various network settings to allow the responder to connect to different, multiple networks, and configure those connections to the Responder SmartHub based upon roles and permissions. This process enables public safety agencies to set the business rules for how information is routed to and from various communication systems.

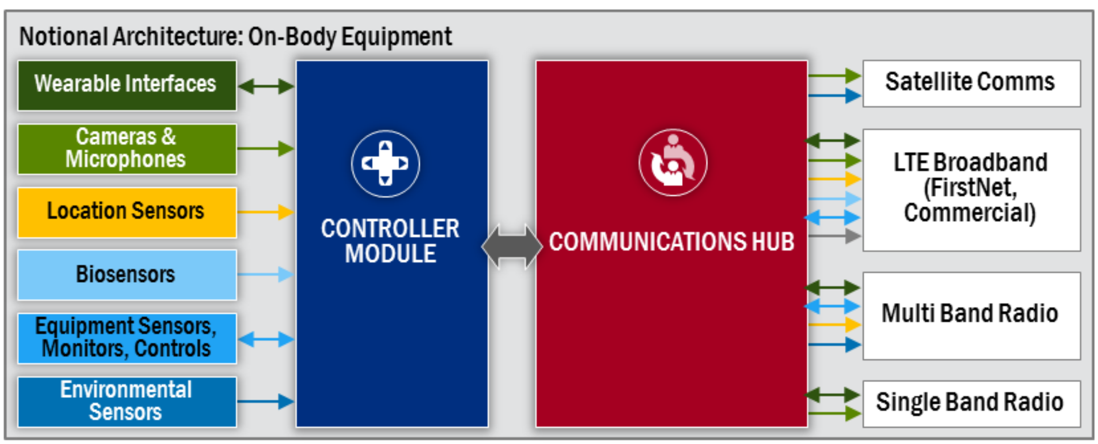

Figure 3: Responder Communications Hub Architecture

C. Sensor Integration and Management

The Responder SmartHub must integrate a variety of on-body and off-body sensors via wired and wireless connections. On-body sensors include Global Navigation Satellite System (GNSS) receivers and/or other geolocation sensor technology to track latitude, longitude and altitude coordinates, and video (camera) sensors with optional infrared sensitivity that can be worn or handheld and that can capture imagery geo-references. On-body sensors also include physiological sensors that measure heart rate, respiration and activity; environmental sensors that measure temperature, humidity and air quality; and geolocation sensors.

D. User Input/Output Interface

The Responder SmartHub module must provide a smart input/output interface, such as touchscreen or voice command, to facilitate input of data, visual output of information, control of applications, and manipulation of data and images. This interface could include speech recognition via headset/microphone, a forearm display/touchscreen or a hand gesture interpretation glove. Output devices include a smartphone touchscreen display, a forearm display or a heads-up-display. This hands-free interface provides the responders with the ability to use their hands in their mission to rescue victims.

E. Power

The Responder SmartHub will require a separate power source. Individual modules shall have internal power sources for short-term operation and be able to recharge from an external high-capacity power source (power module) for long-term operations. The power module should have rechargeable/replaceable batteries and be capable of providing power to all connected modules for a full 12-hour shift and also recharge any wireless modules.

F. Information Management

The Responder SmartHub must be able to receive and disseminate multiple types of information exchanges from responders, public safety agencies and command centers. These include the following:

1. Emergency Situation Tasking Information

Capability to receive detailed and complete messages from radio calls, computer aided dispatch and other information from public safety access points (PSAP) or IC containing the location, data, descriptions and other information regarding the emergency situation.

2. Audio/Video Information

Capability to receive emergency alerts via video/audio files containing the 9-1-1 call and/or other information. It should also have the capability to download video files stored on a server for viewing on a mobile device.

3. Location/Geospatial Information

Capability to receive dispatch information containing the incident location in text form, which is information for the responder's geospatial information system (GIS) that places the location of the event on the responder's GIS display. Other geo-located data transmitted to the Responder SmartHub or stored locally will include other responders, fire hydrants, hazards, alarms, etc.

4. Sensor Observation Information

Capability to accept any sensor device and any sensor data consistent with the standards of this handbook.

5. Alert Information

Capability to generate and receive alert information that meet the criteria and/or business rules for initiation of an alert. This alert information shall be presented to the user visually, aurally and/or haptically. The Responder SmartHub shall support local and remote detection of significant information events, as well as configurable methods of alert delivery (e.g., visual, auditory, haptic).

6. Multi-Level Information Prioritization and Persistence

Capability to manage and prioritize information to and from the responder at all levels: within the Responder SmartHub, IC and agency level. All information to and from the responders shall be logged and recorded for analysis and review.

Guidance regarding public safety data is also contained in National Fire Protection Agency (NFPA) 950 - Standard for Data Development and Exchange for the Fire Service, and NFPA 951 - Guide to Building and Utilizing Digital Information.

G. Standardized Module Hardware Connectors

The standard hardware connectivity among modules will be limited to connectors currently in use by consumer electronics, including the Universal Serial Bus (USB), USB-C, High Definition Multimedia Interface (HDMI), mini-HDMI and mini-phone connectors. Manufacturer-specific connectors, such as Apple iPhone 6 Lightning connector, may be used to provide connectivity for specific devices.

H. Personal Profile

The Responder SmartHub must have the capability to allow users or system administrators (based upon roles and permissions) to create personal settings and preferences, the ability to create specific role-based permissions, and the ability to transfer these persistent profiles from one Responder SmartHub controller to another. User profiles shall be centrally managed by the public safety agency. Roles are expected to be unit, agency and jurisdiction-specific, and user profiles for responders will align with the role or roles to which they can be assigned.

I. Form Factors

Responder SmartHub modules shall conform to a number of standard physical form factors (NFPA has specific guidance on physical devices for Responders; see NFPA 1221, NFPA 1802, NFPA 1859, NFPA 1977 and NFPA 1982) to enhance interoperability with responder clothing, equipment and interchangeability between products. Size, weight, power and form factor constraints will be dependent on responder equipment requirements and usability studies. The final format is the responsibility of the solution providers.

J. Cybersecurity

Rather than try to identify specific requirements for the NGFR On-Body ensemble of equipment, links are provided (below) to help agencies assess their cybersecurity requirements as applicable to the types of equipment they will plan to deploy. It is understood that different agencies will have different levels of cybersecurity implemented in their agency networks. There are tools for assessing an agency’s level of cybersecurity effectiveness included in the following references:

United States Computer Emergency Response Team (US-CERT), Critical Infrastructure Cyber Community Voluntary Program, https://www.us-cert.gov/ccubedvp

US-CERT, Cybersecurity Framework, https://www.us-cert.gov/ccubedvp/cybersecurity-framework

US-CERT, Resources for State, Local, Tribal, and Territorial (SLTT) Governments, https://www.us-cert.gov/ccubedvp/sltt

National Institute of Standards and Technology (NIST), Cybersecurity Framework, https://www.nist.gov/cyberframework

DHS, https://www.dhs.gov/publication/csd-mobile-device-security-study

DHS, https://www.dhs.gov/publication/mobile-device-security

DHS, https://www.dhs.gov/publication/csd-mobile-app-security-study-first-responders

a. User Identity Management

Mobile identity management involves defining and managing roles and access privileges of individual users of devices and networked systems, and the circumstances in which users are granted (or denied) those base privileges and escalated permissions. The primary objective of identity management is to verify and enforce one identity per individual. Once that digital identity has been established, it must be maintained, modified and monitored throughout each user's access session. Identity management grants contextual access to the right device and system resources to properly authenticated users. Any system's user identity management system must be able to define users and their identification attributes, and to securely store or share this data to other system components when necessary.

A challenge for some agencies is the use of shared devices, where a device is not attached to an individual but is shared among several individuals. Any User Identity Management solution will have to account for establishing user identity for shared devices.

b. Device Identity Management

Device identity management involves assigning Unique Identifiers (UID) with associated metadata to sensors, devices and objects, enabling them to connect and communicate with assurance to other system entities over the network. In conjunction with user identity management, these items are a requirement to manage connections between users, devices and other system components. Mobile device registration, or enrollment, is the first phase of system management. The system shall enable secure communications with the Mobile Device Management (MDM) server using specific information for the user and his/her device that is established prior to the enrollment process. The enrollment service shall verify that only authenticated users and their assigned devices can access and be managed by the system.

The enrollment process should include the following steps:

- Discovery of the enrollment endpoint: This step provides the enrollment endpoint configuration settings.

- Certificate installation: This step handles user authentication, certificate generation and certificate installation. The installed certificates will be used in the future to manage client/server mutual authentication and secure communications.
- Device provisioning of approved apps.

c. Data and Communication Security

Information security (INFOSEC) and communication security (COMSEC) govern how data and communications containing valuable information should be stored, transmitted and used. These security functions are designed to mitigate the risk of disclosure of sensitive data on a device and in the system, and to mitigate the risk of unauthorized access, whether through interception or compromise, to plain text or encrypted communication payloads.

SmartHub data shall be encrypted to the (AES) 256 level when stored on-body and when sent off-body. Encryption of data ensures data read by unauthorized users retains a level of security by obfuscating the data. It helps ensure the integrity of data – the assurance that the data has not been changed or tampered with.

d. Physical Security

Physical security for mobile devices consists of analyses and recommendations to reduce and/or mitigate risks due to physical break-ins, loss of a device or theft of a device, and to plan for the consequences of loss or theft. It is the responsibility of the authorized users of the devices to secure and protect the devices and authorization factors for the devices while they are officially in their possession (i.e., assigned to them).

Equipment providers and agencies shall ensure physical security through use of one or more of the following: tamper prevention, keeping devices up-to-date and in operational condition, securely wiping data, closing and removing access to debugging capabilities (e.g., USB or serial debugging ports) once placed in operational capacity, continual monitoring and policing of access to wireless networks, and developing procedures to report suspicious activity if a device is lost or stolen.

III. Encoding, Interfaces and Protocols

A. Data Encoding

Information encodings define the content of messages by which system components exchange information. This encoding may include:

- Geographic Markup Languages (GML);
- Observations and Measurements (Open Geospatial Consortium (OGC) Observations and Measurements v2.0 also published as ISO/DIS 19156);
- Sensor Markup Language (SensorML);
- Extensible Markup Language (XML);
- Open Geospatial Consortium Web Service (OWS) Context;

- Catalog Service for the Web (CSW) Catalog Record;
- JavaScript Object Notation (JSON);
- Geographic JavaScript Object Notation (GeoJSON);
- Sensor Networks: Sensor Network Reference Architecture (SNRA);
- International Organization for Standardization (ISO) 8601;
- Emergency Data Exchange Language (EDXL) standards; and
- National Information Exchange Model (NIEM).

For SmartHub, the recommended data encoding for sensor data is JSON. For enterprise system-to-system encoding of data, the recommended data encoding is EDXL Distribution Element (DE). For alerting, the recommended encoding is to use EDXL Common Alerting Protocol (CAP).

B. Interfaces

1. Machine to Machine

The Responder SmartHub will need to communicate via the following machine to machine (M2M) interfaces:

- Agency computer aided design (CAD)/situational awareness (SA)/GIS systems;
- Agency communications systems;
- Agency data systems;
- Agency audio/video systems;
- Sensors; and
- Public safety cloud (if available).

2. Human-Computer Interface

The Responder SmartHub vendors are expected to provide user interfaces that employ evolving technology (e.g., heads up display (HUD), capacitive touch, voice recognition) and meet human systems interface (HSI) best practices.

Detailed descriptions of the interfaces are provided later in this document.

C. Web Services

1. Open Geospatial Consortium

This section identifies the Open Geospatial Consortium (OGC) Web service standards that handle data types, standards and other geospatial information sources. These standards represent services and protocols that may be applicable in operational contexts, which use or process information described in the Information – Models and Encodings Section. As Web services, these standards typically rely in turn on fundamental web standards such as Hypertext Transfer Protocol (HTTP). Below is a representative list of standards; however, additional standards may be identified as necessary to realize a given functional capability:

- OpenGIS ® Web Map Service (WMS);
- OpenGIS ® Web Feature Service (WFS);
- Catalog Service for the Web (CSW);
- Web Processing Service (WPS);

- Sensor Observation Service (SOS);
- Sensor Things Application Program Interface (STAPI); and
- Sensor Notification Service (SNS).

D. Communication Protocols

This section identifies communications layer protocols that provide message handling, queuing, mesh networking, device discovery and other capabilities, particularly in support of the local networks involving inexpensive, low-power sensors. Protocols are typically defined and implemented in layers, so that choice of protocol in one layer (e.g., Bluetooth low energy (BLE) versus Long Term Evolution (LTE)) does not constrain choices in other layers (e.g., HTTP versus message queuing telemetry transport (MQTT)). A critical vertical interface occurs between protocols that support Internet Protocol (IP) packet transmission with transmission control protocol (TCP) or user datagram protocol (UDP) signaling, and protocols that operate on top of the IP protocol such as HTTP. A critical horizontal interface occurs between local Internet of Things (IoT) protocols that do not support IP packets (e.g., Constrained Application Protocol (CoAP), Data Distribution Services (DDS), +/- BLE) and those that do. A representative selection of protocol standards is listed below, but additional standards may be identified as necessary to realize required functionality:

- HTTP;
- TCP/IP;
- IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN);
- BLE;
- ZigBee;
- Extensible Messaging and Presence Protocol (XMPP);
- MQTT;
- CoAP; and
- DDS.

IV. Engineering Design

This section describes the technologies, practices and solutions that provide the functionality of each module, interactions among them, and interactions between each module and system and subsystem at the IC, PSAP and agency level.

A. Controller Module

As indicated in Part 1 of this Handbook, the Controller Module is the central component of Responder SmartHub and supports routing, persisting and processing data, interacts with the other core Responder SmartHub modules, and mediates their power requirements. The module supports standard data services and applications, and manages the federation and synchronization of data with other personal, field and cloud sensor hubs involved in an incident response. To perform these functions as a wearable device, the Controller Module maintains and uses the wearer's personal profile information to customize the Responder SmartHub experience and identifies the source or subject of sensor information being transmitted to others. The Controller Module is expected to provide location information for the responder and to provide that location information to other responders. The module may be equipped with limited communications capabilities (e.g., Wi-Fi, LMR, Bluetooth, Long Term Evolution (LTE), etc.), or those may be all contained within the communications hub.

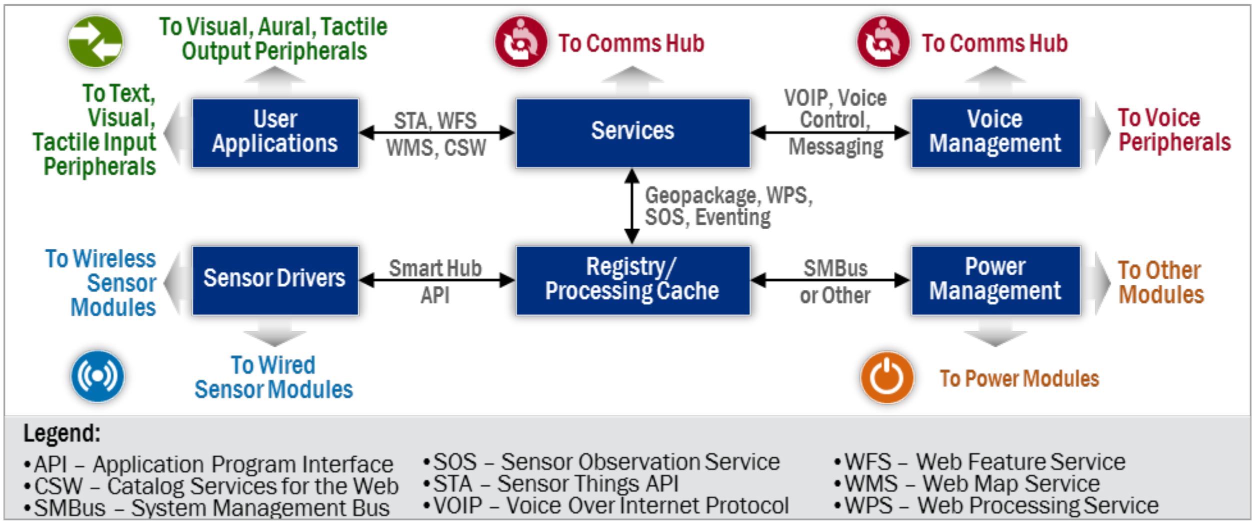

Figure 4: Controller Functionality

1. Sensor Management-

The controller houses the sensor hub application/service that interfaces with other sensors and provides a discoverable, consistent, open standards-compliant web interface.

Discoverable means that the sensor hub is available for other systems to access. Sensor hubs exist as both field hubs (software located on a Responder SmartHub Controller) and regional/cloud hubs (software located centrally for an entire agency). Sensor hubs can be synchronized for information redundancy, bandwidth mitigation and persistence of information. A sensor hub provides a flexible way to deliver information captured from the responder to be delivered to the individual responder and to all authorized users and systems, independent of their specific implementation architecture. This means any responder can obtain information from other responders or other deployed sensors, thus increasing situational awareness. A sensor hub deployed on the responder in specialty equipment, or in other equipment such as a mobile phone or tablet, connects to the central infrastructure and provides a consistent interface to deliver information to all responders. Responders will, upon donning their Responder SmartHub equipment, enable the sensor hub, and it will register with the incident management infrastructure. From then on, the responder is a sensor platform running a sensor hub service capable of delivering information to a range of authorized users.

The sensor hub should be provided in the form of an application or service running on a Responder SmartHub controller. Alternately, it could be an application or service running on a sensor platform and serving other sensors, or a separate module managing a large number of sensors. Sensor hubs should be arranged in a hierarchical form, with local sensor hubs carried by the responder and regional sensor hubs located at the IC, agency or even public safety cloud level, and managing the data from multiple local sensor hubs.

A sensor hub is expected to interface to sensors via a number of proprietary interfaces and delivers data via a number of OGC/IoT compliant services. The current mapping of sensor hub conceptual interfaces to open standards is shown in Figure 5.

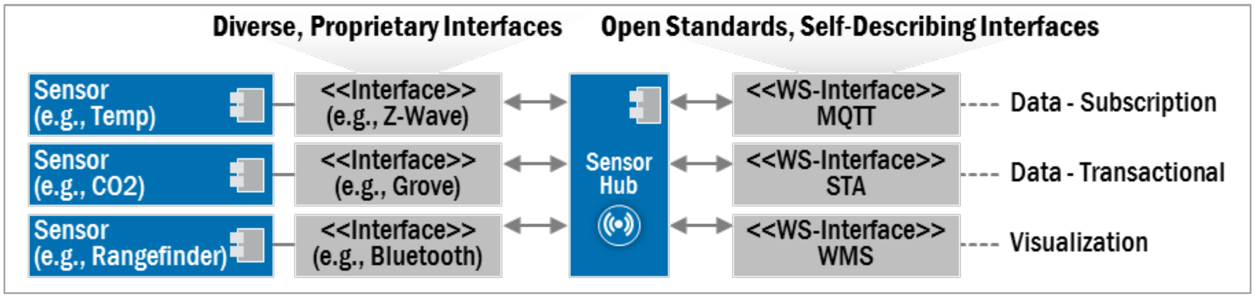

Figure 5: Sensor Hub Implementation

Sensor hubs have been tested and demonstrated in experimentation using several standards. The web service interfaces supported are:

- STAPI 1.0 (mandatory);
- MQTT 1.0 (mandatory); and
- WMS (optional).

STAPI offers the opportunity for clients of the sensor hub to query the object of interest, the observations and the observed properties, as well as the type of sensor. This offers a very general access model. In addition to transactional standards, sensor hub supports subscription-based interfaces, which provide immediate updates based on either changes in value or values exceeding a threshold. Within the sensor hub, the standard used message-based communication is MQTT, which has a close relationship with STAPI.

Note two modes of operation are possible, and a sensor hub instance would be able to handle both:

- A sensor hub includes specific interfaces to existing sensor protocols (Z-Wave, Grove etc.). It is therefore an ‘adapter’ that standardizes the sensors and typically offers a read-only web service interface.
- Sensor systems are themselves modified to be able to interact with the sensor hub via the STAPI interface. They, as a STAPI client, can write data into the sensor hub, which provides capability such as information caching, etc.

STAPI offers the opportunity for clients of the sensor hub to query the object of interest, the observations and the observed properties, as well as the type of sensor. This offers a very general access model as shown in Figure 6.

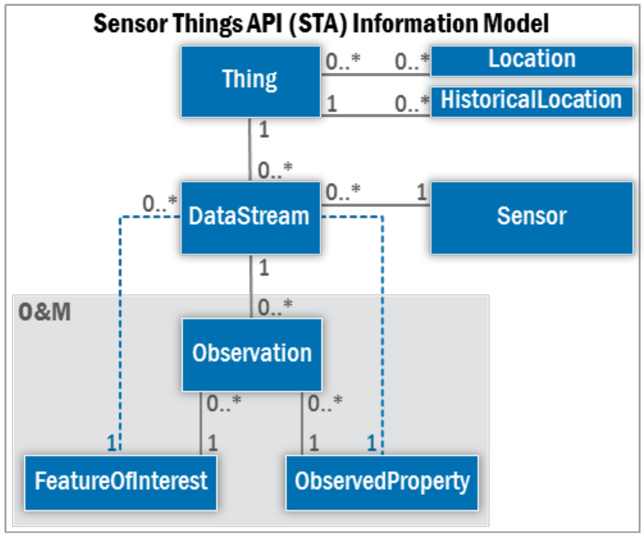

Figure 6: STAPI Information Model

a. Sensor: Registration Process

A key capability of the sensor hub is sensor discoverability, including on-body and off-body sensors. For this to work, a responder's sensor hub must be registered with a sensor hub catalog. A key element of an effective NGFR architecture is an awareness by all users of the deployed human resources so they can be effectively used and protected. Critical in this process is the registration of the systems deployed on a responder and information about their identity. For equipment deployed on a responder, each responder will have a unique identifier. This identifier will be entered and can be used to configure equipment deployed on a responder. A controller identifier format needs to be defined, but the primary goal is to identify the responder device on which the sensor hub is deployed.

Registration and discoverability is performed either in the sensor hub or split between the sensor hub and the hub catalog. The sensor hub/hub catalog combination ensures that the sensors for all responders on-scene that are capable of registration will be registered and discoverable. The CSW Catalog Record previously referenced in this document specifies the minimum output standards that sensors should be ready for the catalog, and this becomes the minimum set of attributes that a sensor catalog should contain. The overall registration process is shown in the sequence diagram below, Figure 7. When a sensor hub boots and comes online, it sends a request to the publishing service (potentially a regional Sensor Hub, a WFS or a CSW), which then harvests the sensor hub capabilities and populates the catalog as necessary. The publishing service returns the identification (ID) of the entry (as a Universally Unique Identifier (UUID)) so that the sensor hub can update or remove the entry as its status changes.

This workflow depends on the sensor hub knowing to what catalog or publishing service it needs to connect. An alternative is an external trigger, which performs the 'add' request, which might be relevant in some circumstances.

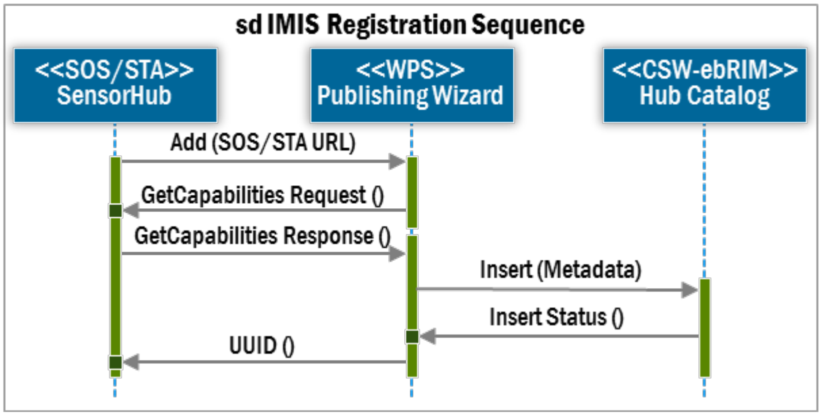

Figure 7: Sensor Hub/Hub Catalog Registration Process

b. Sensor: Update Process

The update process is initiated by the sensor hub requesting an update using the ID returned during the registration process, shown in Figure 8.

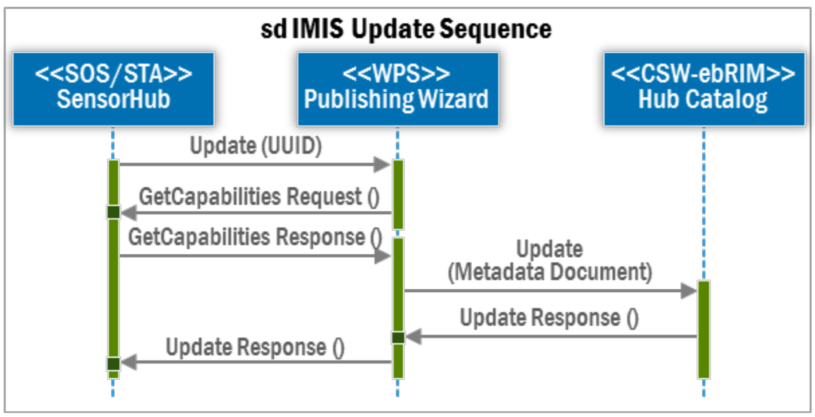

Figure 8: Sensor Update Process

c. Sensor: De-Registration Process

A similar process occurs when the sensor hub shuts down. They will initiate a de-registration process using the ID returned during registration. The result is the hub catalog will only show currently registered (and, by implication, operational) sensors, shown in Figure 9.

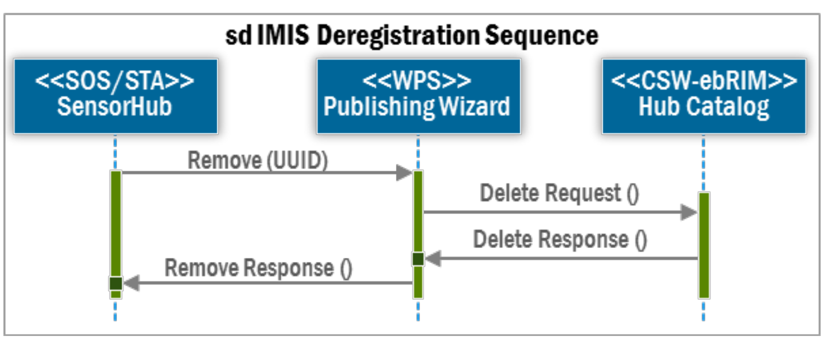

Figure 9: Sensor De-registration Process

While de-registration could remove the sensor hub from the catalog, it could potentially just mark it as “offline” or “deleted” in the catalog, along with all details of the sensor, when it was online, etc. This is a decision related to the permanence of the sensor and the need to keep records of sensor availability/use. Implementation of the catalog should poll any registered services at a configurable rate and change the status of the service from online – offline or vice versa, if required.

2. Applications

The Controller Module is expected to contain multiple applications, with each application providing a capability or group of capabilities to the first responder. These applications will primarily be provided by commercial vendors to provide functionality, such as:

a. Situational awareness;

b. Sensor hub;

c. Collaboration;

d. Messaging (Short Message Service or SMS);

e. E-mail;

f. Mapping;

g. CAD interface;

h. Hazardous Material (HAZMAT) information;

i. Medical treatment information; and

j. Sensor management.

3. Sensor Drivers

The Controller Module is expected to host the various drivers used to interface with the multiple sensors, Input/Output (I/O) devices and other modules used by first responders. Because there is no standardized sensor driver that will work with all sensors, each sensor manufacturer will have to provide a compatible driver for its associated sensor that runs on the controller’s operating system. These drivers may be installed on the controller along with the corresponding applications or bundled separately by the agency and delivered as a single driver package.

Sensor manufacturers should build libraries in commonly used programming languages such as Java, Python and C, compatible for Android, Apple iPhone Operating System (iOS), and other operating systems, so they can easily be integrated into the NGFR Architecture.

4. System Administration

The Controller Module administration functions are intended to allow authorized personnel, based upon permissions, profiles and roles to view the status of an operational sensor hub, make changes to its internal configuration, and set up and deploy a sensor hub. These administration functions are an integral part of the sensor hub. They are necessary for both the initial deployment and to allow reconfiguration as needs and priorities change. The responder should be able to access the administration functions using any network capable device, such as a laptop, tablet or phone by using any available web browser.

5. Module Status

The Controller Module should be able to present the user or administrator (based upon roles and permissions) with a high-level status of all pertinent information. The status should include, but is not limited to:

- Software version;
- Uptime/downtime statistics;
- Media Access Control (MAC) Address;
- IP address;
- Host;
- Service Uniform Resource Locator (URL);
- Storage space remaining;
- Status of all connected devices;
- Power details:
    - State of the device (i.e., plugged in, running on battery, etc.);
    - Power status of all connected devices,
    - Percent of battery remaining; and
    - Estimated operational time remaining.

6. User Management

Privileged users should be able to create and manage users and their associated permissions. Sensor hubs may operate in disconnected operations, so local user management is important. Permissions may be used to limit access to a hub, specific services or data within a service.

7. Rules Management

The Controller Module should allow a user or administrator (based upon roles and permissions) to create complex Boolean logic rules that, when matched, can trigger the hub to perform an action. Actions can include tasking devices or sending alerts by a variety of channels including email, text messages and MQTT topics. Email and text support allow for existing devices without specialized applications to receive the alerts, while MQTT delivers alerts to applications incorporating MQTT clients. MQTT topics assist by more easily identifying the content of sensor messages in an organized fashion; for example, certain data transmissions can be labeled as heart rate or humidity readings and assigned to Responder A and Responder B. It makes it easier to work with, prioritize and therefore manage. This handbook is intentionally silent on how to label or structure message topics to provide maximum flexibility in the field.

8. Driver Management

The Sensor Hub should allow a user or administrator (based upon roles and permissions) to upload and configure drivers that connect sensors and devices to the hub. Some sensors and devices may have the capability to register directly with the services running on the Sensor Hub; however, some devices may just be connected directly to the hub, and therefore the hub will be responsible for making their data available in the services.

9. Connection Management

Users or administrators (based upon roles and permissions) should be allowed to configure any external connections from the hub to other systems and hubs. Specifically, the hub should allow the user to configure to the catalog(s) with which it will be registered, allowing it to be discovered externally. The hub will also allow the user to configure to other hubs where it will push its data and prioritize the data transfer. This is particularly useful to push data from a field hub to a cloud hub.

10. Data Management

The Controller Module should allow a user or administrator (based upon roles and permissions) to view the current status of the device storage by indicating how much space is used and how much is still available. The user or administrator should be provided options for cleaning cached data older than a specified date and time, or to allow data to only be maintained for a specified period of time. The user or administrator should also be able to clear specific sensor data or types of data. The sensor hub should allow a user or administrator to prioritize the transfer of data. The user or administrator should be able to indicate the importance of specific types of data. For example, the user or administrator may want audio to take precedence over video; however, gas readings may take precedence over audio. The user or administrator should also be able to specify permitted reductions to data if they are necessary. For example, a user or administrator may want to reduce video from 30 frames per second (FPS) to 10 FPS if bandwidth is an issue, or to push sensor readings less frequently than they are captured.

11. Device Configuration

The Controller Module should allow a user or administrators (based upon roles and permissions) to modify any device configuration settings. These settings may include:

- Hostname configuration;
- Email configuration;
- MQTT configuration;
- SMS configuration;
- Fully Qualified Domain Name (FQDN)
- Date and time configuration (based upon agency time standard); and
- Default geospatial location of the device (if no GNSS is present).

The administration functions are part of the core module. They require that network capable devices are able to reach the administration web application via a web browser. The sensor hub will retain any configuration changes and write them to persistent storage.

B. Communications Hub Module

As depicted in Figure 3, the Communications Hub module (Comms Hub) is a component of Responder SmartHub, and works with the Controller Module to provide the functionality of interconnecting multiple wearable user devices (microphone, speaker, cameras, etc.) with communication devices (e.g., LMR radio, FirstNet and commercial cellular smartphones, Wireless Fidelity (Wi-Fi) and other networking devices).

The Comms Hub is intended to enable the responders to manage voice and data services with a minimum of user distraction and inputs. Among the services the Comms Hub will help the user to interconnect will be:

- Voice service: push-to-talk (PTT), "simplex" voice calls and full-duplex voice calls; and
- Data service: body-worn camera, GNSS, body-worn sensors and smart glasses.

Figure 10 provides a more detailed view of the functional components of the Comms Hub, and the following sections will provide greater detail of the interfaces and functions within the Comms Hub.

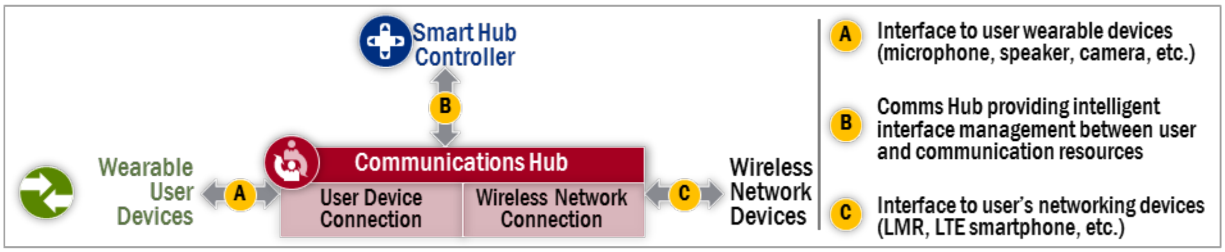

Figure 10: Detail Comms Hub Functional and Interface Diagram

The Comms Hub communicates with the Controller Module within the Responder SmartHub system to define which services will be subject to monitoring and configuration changes by the Controller Module. Some of the high-level controller functions of the Comms Hub include:

- **Communication link status monitor**: Comms Hub will monitor the status of each of the connected network devices and the associated service level capability to determine the best connection link for user voice and data services.
- **User data cache**: Provide the ability to cache user data in response to an outage of communication network resources.
- **Interface status**: Comms Hub will monitor the status of each of the available connection interfaces (A, B and C) and provide status of available wired and wireless ports to connect to the Comms Hub.
- **Resource priority**: If a conflict for interface resources arises, the Comms Hub assigns the interface resources to the service(s) having a higher priority level to ensure critical data delivery.
- **Service override**: The Responder Controller Module has the ability to override existing or pending user data traffic and instead enable designated interface(s) and communications resource(s) to carry designated priority-based voice and data traffic.

1. Voice Design

The Comms Hub provides the interfaces to carry voice traffic to and from the users. In conjunction with the Responder SmartHub Controller Module and other elements, the Comms Hub uses a variety of communication network resources, as well as a plurality of body-worn devices, to carry out voice communications. The following features provide the needed interfaces and control functions to support voice traffic within the Responder SmartHub architecture:

- **Interface A**: Comms Hub will provide the following methods to connect/pair with the user body-worn devices:
    - Bluetooth –Version 4.2 and above
    - Wi-Fi – 802.11a/b/g/n/ac
    - USB – all approved versions
    - Audio jack (3.5 mm Tip-Ring-Sleeve (TRS) jack/3.5 mm Tip-Ring-Ring-Sleeve (TRRS) jack)
    - Optional: Wired connection using standard interface protocol (e.g., Ethernet)
- **Interface B**: Comms Hub control functions for voice traffic
    - Mission-critical voice (PTT) via public safety LMR and FirstNet networks
    - Non mission-critical voice (PTT) via commercial cellular networks
    - Commercial cellular-grade voice
- **Interface C**: Comms Hub interface to network devices (non-inclusive)
    - Bluetooth – Version 4.2 and above
    - Wi-Fi – 802.11a/b/g/n/ac
    - USB – all approved versions
    - Optional: Wired connection using standard interface protocol (e.g., Ethernet)
- **Example of supported device types**:
    - User: push-to-talk microphone, speaker, ear bud with microphone
    - Network device: LMR radio (e.g., conventional, trunked, Project 25 (P25) LMR), FirstNet wireless device, commercial cellular device, satellite radio, mobile ad-hoc (meshed) digital radio

2. Data Design

The Comms Hub provides the interfaces to carry data traffic to and from the users. In conjunction with the Responder SmartHub Controller Module and other elements, the Comms Hub uses a variety of communication network resources, as well as a plurality of body-worn devices, to carry out data communications in support of situational awareness and decision-making. The following features provide the needed interfaces and control functions to support data traffic within the Responder SmartHub architecture:

- **Interface A**: Comms Hub will provide the following methods to connect/pair with the user body-worn devices**:**
    - Bluetooth – Version 4.2 and above
    - Wi-Fi – 802.11a/b/g/n/ac
    - USB – all approved versions
    - Audio jack (3.5 mm TRS jack / 3.5 mm TRRS jack)
    - Optional: Wired connection using standard interface protocol (e.g., Ethernet)
- **Interface B**: Comms Hub control functions for data and traffic
    - Mission-critical data and video via the FirstNet network
    - Non mission-critical data and video using commercial cellular networks
    - Datacasting network to distribute IP and broadcast-based data files
- **Interface C**: interface to network devices (non-inclusive)
    - Bluetooth – Version 4.2 and above
    - Wi-Fi – 802.11a/b/g/n/ac
    - USB – all approved versions
    - Optional: Wired connection using standard interface protocol (e.g., Ethernet)

- **Examples of supported device types**:
  - User devices: body-worn sensors, body-worn camera, smart glasses with display capabilities
  - Network devices: FirstNet wireless device, commercial cellular device, satellite radio, mobile ad-hoc (meshed) digital radio, datacasting receiver and dongle, agency legacy LMR/P25 radios

3. Physical Design

The Comms Hub physical attributes encompass the following (non-inclusive) features:

- **Ruggedization**: Follow NFPA 1802 guideline, Standard on Personal Portable (Hand-held) Two-Way radio Communications Devices for Use by Emergency Services Personnel in the Hazard Zone and NFPA 1800 guideline, Standard on Electronic Safety Equipment for Emergency Services.
- **Comms Hub unit**: Standalone unit, or maybe integrated as a part of an electronic device such as a smartphone.
- **Visual Display**: Display indicator to provide status information of the Comms Hub operation.
- **Emergency Button**: Provide a panic button to send an urgent message (voice and/or text message) to incident command of impending danger or hazard condition.

C. Sensor Modules

There are numerous different types of sensors developed to support responders. They use a variety of protocols and wired/wireless connections to deliver sensor data to devices such as the Controller Module. Sensors should be developed with different applications in mind, for example, some may be on-body and therefore associated with a specific first responder (example: body camera or heart rate monitor), a similar sensor may be deployed off-body at the incident site (example: drone camera), and a similar sensor may be accessed from agency networks (example: camera mounted to a building on a street corner).

1. Location Sensor Design

The location sensor is responsible for providing spatial location and orientation for the controller and any connected sensors. The location sensor will allow for tracking of personnel and location-equipped sensors, and will therefore provide real time situational awareness to those who need it. It will allow users to not only see their locations, but the locations of their peers, deployed sensors and location-equipped units (e.g., vehicles, aircraft, boats, etc.). It is possible to use the location of the various sensors to create geo-referenced alerts. For example, if a specific location-equipped sensor detects a gas leak, all the personnel in its vicinity can be instantly notified. The location sensor should run autonomously and seamlessly switch between location sources (if available) to provide the most precise location possible. The only interaction with a user should be to manually enter a location or to disable tracking, if the need arises.

d. Tracking Control

The location sensor should allow the user or administrator (based upon roles and permissions) to easily enable and disable tracking, and view or delete tracking data.

e. Manual Location Entry

The location sensor should allow the user or administrator (based upon roles and permissions) to manually enter a relative location for those instances where automated locations cannot be provided. This location should not be used for precise positioning.

f. IP Geolocation

The location sensor should automatically provide an IP geolocation to the sensor hub when network connectivity is available. This location should not be used for precise positioning.

g. GNSS

The location sensor should automatically provide a GNSS location when a signal is available. This location should include latitude, longitude, precision, timestamp and altitude.

h. Cellular Telephone Location

Location data obtained from cellular telephones shall, if so equipped, include assisted location data in addition to GNSS location data.

i. Other Location Services

The system should be able to accept location data from other location services (e.g., in-building, Wi-Fi based, Bluetooth beacon based, other beacon type based, etc.) and pass it on to the situational awareness applications. The situational awareness applications may need to de-conflict location information for a device coming from two sources if the information does not match within configurable parameters.

j. Orientation

Many sensors provide observations that are directional in nature. These include, for example, video and imaging cameras, wind direction, laser rangefinders, and acoustic detectors, just to name a few. It is important to provide an orientation suite of sensors (e.g., accelerometers, inertial momentum units and geomagnetic sensors) that provide accurate orientation for the sensors.

k. Location Message Transmission Frequency

In order to reduce bandwidth requirements while still providing the necessary location data, developers should consider the following strategies:

- Speed sensitive – transmit location messages at a frequency based upon the sensor's speed – a controller on a stationary officer directing traffic or a firefighter controlling a pumper would transmit less frequently than an officer in foot pursuit or a firefighter advancing on a fire.
- Status sensitive – transmit location messages at a frequency based upon a responder's status. A controller on a responder "out for a meal" would transmit less frequently than one on a responder assigned to a call for service.

2. Sensor Drivers

The location sensor does not require any specific sensor interface. The sensor hub driver function allows support to any sensor device interfaces that make use of the existing connection ports (USB, Bluetooth, etc.). For example, a USB GNSS that supports National Marine Electronics Association (NMEA) 0183 or a Garmin Virb that connects over Wi-Fi and provides a proprietary location interface could be used. The driver needs to know the sensor is a location provider. It is also possible for the location sensor to push its location data directly into the sensor hub by using the SensorThings service. This process would not require a sensor hub driver. The location sensor should retain the latest location, so it can be retrieved at any moment without having to wait for a new location observation to occur. The sensor hub driver facility allows for any device to act as a location provider. For example, a user's GNSS sports watch or a GNSS-enabled body camera could provide the location for the sensor hub.

3. Sensor Module: Imaging

Imaging sensors can include still imagery and video (or motion imagery). Imagery and video are a collection of thousands of simultaneous measurements with each pixel value having been influenced by something in the view at some distance from the sensor. Imagers are therefore often referred to as remote sensors. Imagers can record scenes within the spectral range visible to humans and capture scenes in other wavelengths within the electromagnetic spectrum. This can include, for example, thermal imaging, microwave detection or multi-spectral imagery including measurements in hundreds of spectral bands. It is therefore important that the imaging module not only capture the imagery itself, but also other measurements, such as the location, orientation and focal angle of the camera, as well as camera settings affecting the sensitivity of the sensor within the electromagnetic spectrum.

Imaging modules allow the responders to gain a visual awareness of the size, scope and intensity of the incident on hand, particularly for those who are not at the scene. It also allows the responder to convey to citizens the scope of the incident so that they can respond accordingly. Furthermore, imagery and video in non-visible wavelengths can provide the responder with situational awareness not available with their own eyes. An example would be thermal imaging available from cameras sensitive to infrared (IR) wavelengths. These can provide the responder with knowledge about the temperature of a fire, can determine locations of leaks of gas or liquid, and can allow one to see heat sources, including humans while in total darkness. Additionally, the video and accompanying data (location, orientation, camera settings) can be transmitted in real time via LTE or Wi-Fi, for instance, to other hubs on the agency's wide area network (or the internet) for immediate display by command and control during the incident, as shown in Figure 11.

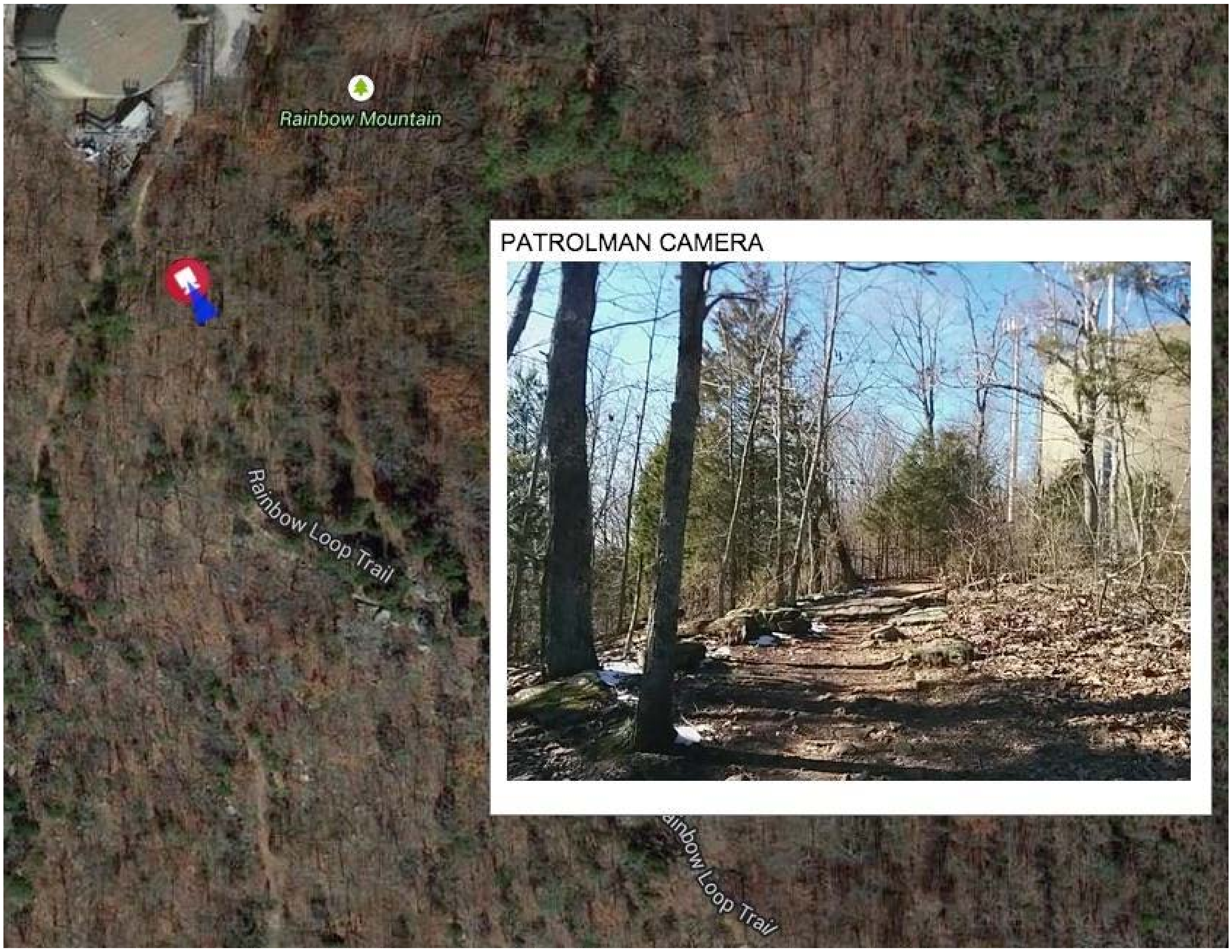

Figure 11: Body Camera Image

Imaging modules would typically be mounted permanently onto buildings or other structures, attached to mobile vehicles (e.g., dash cams or hood cams), worn by responders (i.e., bodycams), hand carried, airborne (e.g., drones and balloons), or distributed at the scene (e.g., drop cams or sticky cams). While video and imagery could be recorded for later review, the imaging module is most effective if the video or images, as well as the location, orientation and settings, can be made available to local responders and remote observers in real-time. While a responder could serve as a carrier for the imaging module (e.g., to remote viewers), the local responder could also view the video or imagery output to gain increased situational awareness. If a pan-tilt-zoom (PTZ) capability exists (on a vehicle mount, for instance), remote command and control could remotely task the camera to look at different areas of the scene.

The imaging modules should be capable of capturing video or images, location, orientation, field of view, and camera settings. The imaging module should provide accurate time tagging of all of this data using a single, synchronized and accurate clock. The imaging module should be capable of real-time broadcasting of this data to a local field hub or to remote hubs through the internet or broadcast channels (e.g., datacasting). The imaging module should be capable of supporting PTZ control by the responder where appropriate (e.g., on a vehicle mount).

D. Hybrid Module

The hybrid model (HM) refers to a Responder SmartHub Controller Module that fulfills three key roles: sensors collecting information, the sensor hub managing information from all sensors and a user interface to deliver that sensor information to the responder. Each of these roles is satisfied by the deployment of a software component. The technology used in tablets and smartphones is receiving a very high level of investment and so it is highly capable of providing the software platform for deployment of the software, sensor and input/output components for a hybrid module.

A responder's interface needs the ability to present information clearly in one of a number of consistent styles to deal with specific needs and needs to be easily configured. Both summary and detailed information is needed. The following are representations used in previous Responder SmartHub demonstrations:

- Environment sensor information – fuel gauge/highlight representations;
- On-body cameras – video windows, specific snapshots; and
- Responder and other asset locations – map or schematic (in building) display or counts of people nearby.

A responder's equipment will be configured to match their profile, and information can be delivered to each individual responder based on their identity and assigned role. The Responder SmartHub system should provide an operational view (invoked on a mobile device or tablet by clicking on an icon), which displays the key information for that responder. Information layers should also be provided so that the responder can view information needed for their role (e.g., blueprints, standpipe connections, electrical wiring layout, etc.). These views can be constructed as layers or templates and updated as necessary for a given situation. One solution would be to define such views as open standards compliant Open Geospatial Consortium Web Service Context documents.

1. Smartphone

A smartphone can play four different roles in the context of Responder SmartHub:

1) May serve as a gateway device forwarding sensor observations from sensors to a sensor hub service.
2) May play a sensor role as there are many built-in sensors on a smartphone.
3) May be a client device of the sensor hubs that allows users to visualize sensor observations or receive notifications
4) May host the sensor hub application and act as a platform for the sensor hub.

Figure 12 shows an example of a smartphone as a gateway device.

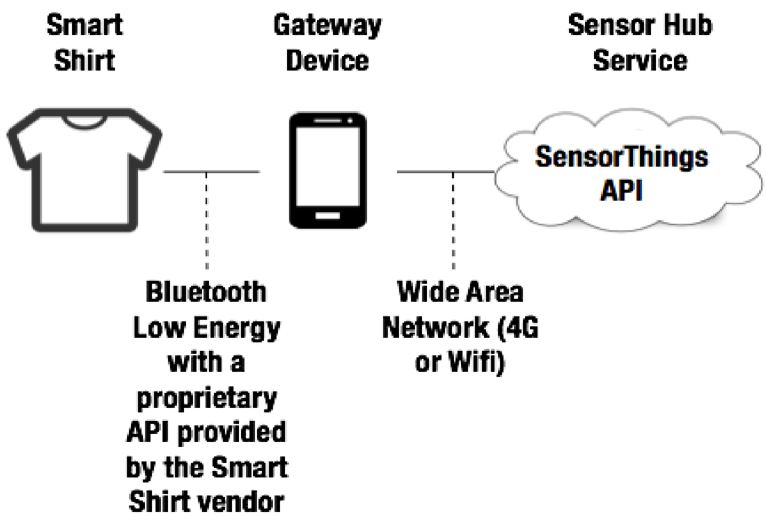

Figure 12: Smartphone as a Gateway Device

a. Smartphone as a Sensor System

A smartphone has many built-in sensors that can be useful for responders. Accessing the sensor data depends on the smartphone operating system. Android operating systems provide APIs for applications to access sensor data,[1] (e.g., accelerometer, orientations, air pressure, gyroscope, etc.) In addition to these *in-situ* sensors, a smartphone's camera can be a very useful sensor when used to broadcast real-time video to a sensor hub service. Below are details of how a smartphone can register itself as a camera sensor in a sensor hub service.

In order to be accessible in a sensor hub service, the smartphone needs to register itself to a STAPI. It can be provisioned in advance or the smartphone can register itself by sending POST requests to a STAPI. The following Unified Modeling Language (UML) summarizes the data model of a smartphone as a video camera sensor.

[1] https://developer.android.com/guide/topics/sensors/sensors_overview.html

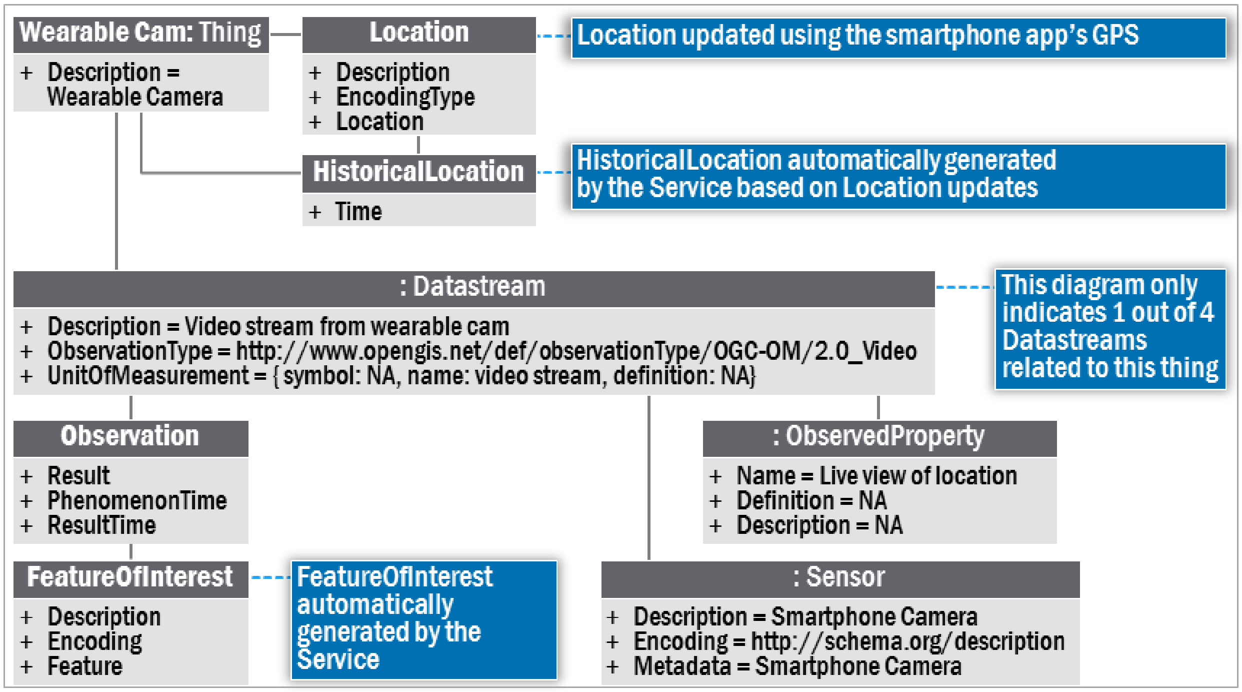

Figure 13: Data Model of a Smartphone as a Video Camera

Figure 14 is a sequential diagram showing the interactions between a smartphone as a video sensor and an OGC STAPI.

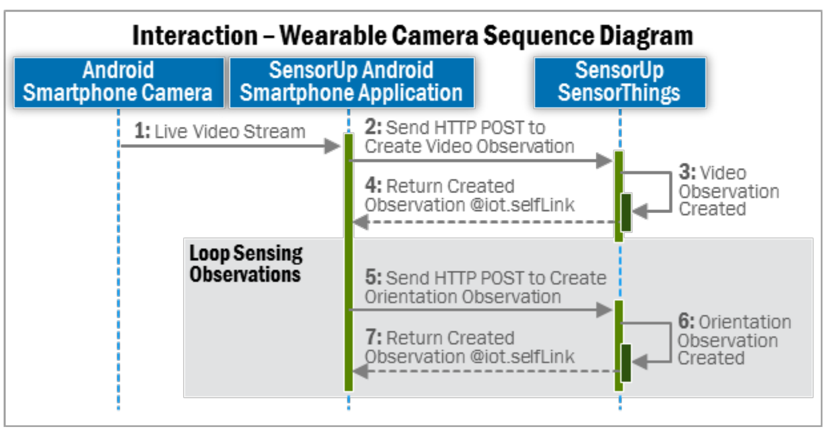

Figure 14: Interactions between Smartphones and OGC STAPI

Based on this diagram, the example below shows JSON requests of a smartphone registering itself to an OGC STAPI.

Example - Request/Response for Adding a Video Camera Video Stream to OGC STAPI

Example Request

POST /Things HTTP1.1

Host: example.org/v1.0

Content-Type: application/json

```
{
 "description": " Wearable Camera",
 "Locations": [
   {
     "description": "GNSS Location",
     "encodingType": "application/vnd.geo+json",
     "location": {
       "type": "Feature",
       "geometry": {
         "type": "Point",
         "coordinates": [
           10,
           10
         ]
       }
     }
   }
 ],
 "Datastreams": [
   {
     "description": " Video stream from wearable cam ",
     "unitOfMeasurement": {
       "name": " video stream",
       "symbol": null,
       "definition": null
     },
     "observationType": "http://www.opengis.net/def/observationType/OGC-
OM/2.0/OM_Video",
     "ObservedProperty": {
       "name": " Live view of location",
       "definition": null,
       "description": null
```

```
      },
      "Sensor": {
        "description": " Smartphone Camera",
        "encodingType": "http://schema.org/description",
        "metadata": " Smartphone Camera"
      },
      "Observations": [
        {
          "result": " http://example.org/video"
        }
      ]
    }
  ]
}
```

Example Response

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

b. Smartphone as a Client Device for Sensor Hub Services

A smartphone can also be a client device for users to consume sensor observations from sensor hub services. The interactions and request/response between a smartphone client and an OGC STAPI are similar to any desktop-based client.

Finally, a smartphone can also act as a sensor hub to receive sensor observations from sensor devices.

2. Smartwatch

A smartwatch is a wearable, consumer device. Capabilities depend on the specific hardware device; however, they may include:

- Input: Movement, GNSS, heart-rate, I/O (button, dial, touchscreen, force-touch)
- Output: Display, haptic
- Network: Bluetooth, Wi-Fi

3. Other Application Functionality

Power management, device security and provisioning are important considerations when deciding to use a hybrid module. While very computationally powerful, today's mobile devices are not designed for power requirements that responders need. Responders' work shifts are often eight hours or more; however, very few currently available commercial smartphones can run a GNSS-intensive application for eight hours straight without overheating or running out of power. Developers looking to use a hybrid module approach need to be cognizant of this limitation and provide the appropriate optimization or backup mechanisms to better support a responder's mission. Providing a way to allow the user or agency to configure the hybrid device to poll certain information on a periodic-basis is very desirable. A responder on foot may not require their GNSS to constantly provide updates, as they typically have not moved very far since the last update (if at all). Allowing the user to configure their device to only get GNSS position once every minute or two could greatly extend battery life, while still providing adequate responder positioning. Other sensors, such as heartbeat sensors, may provide their own power. However, if such a type of sensor (continuously updating) requires power from the hybrid module, power consumption needs to be considered and managed.

Mobile devices are not as secure as today's commercially available laptops or computer systems, because they are physically more accessible and signal encryption takes computer processing time and battery power. While a majority of the currently available mobile devices support some type of device encryption, not all encryption is equal, nor is it enabled initially. It becomes beholden on the user to enable encryption to better secure the data on the device. Security needs to be enabled both on the device and signal levels. All communications should be encrypted to as high a level as possible. Additionally, a hybrid device should require strong passwords or secure access mechanisms. Device encryption does no good if a bad actor can access the device through a simple pattern swipe.

Given the chaotic nature of larger scale events, a straightforward provisioning process should be considered for hybrid devices. A new responder showing up to a large-scale incident needs to be able to quickly and securely identify themselves, be granted the right level of access to the appropriate systems, and set up their device in the context of the incident (i.e., configure the correct networks, get information from the correct systems, connect to known field sensors, etc.). Proper authentication and authorization vetting of responders on-scene is an important part of incident safety and security.

E. Interface Descriptions

1. Controller Module-Comms Hub Module Interface

The Controller Module (hosting the sensor hub service) to Comms Hub communications will need to support a variety of different interfaces. Depending on how the sensors associated with the controller are connected, the Comms Hub may only have to support STAPI and MQTT (and optionally WMS) communications to and from the controller. Alternatively, the Comms Hub may also have to support the sensor driver interfaces to the Controller Module. Sensor driver interface support is largely dependent on the module to which the sensors are connected. If the Controller Module supports Bluetooth independent of the Comms Hub, for example, then Bluetooth sensors can connect directly to their Sensor Drivers. This is the same for USB, if the Controller Module supports USB independent of the Comms Hub. Otherwise, the sensors will need to connect their respective sensor drivers through the Comms Hub, which will require the Comms Hub to support the sensor driver interfaces to the controller.

The same will hold true for the Controller Module to I/O interfaces. If the I/O module is connected directly to the Controller Module, the Comms Hub will not have to support the Controller Module - I/O interfaces. If, however, the I/O module connects to the Controller Module through the Comms Hub, then the Comms Hub will need to support the appropriate interfaces to the I/O and Controller Module.

2. Controller Module-Sensor Interface

The Controller Module and sensors primarily communicate via a sensor driver interface. While the sensor to sensor driver interface is specific to the type of sensor, the Controller Module and sensor driver interface is more generalized. This section describes the Controller Module – sensor driver interface in more detail.

a. General Capabilities

Each sensor driver shall support the following capabilities:

- **Auto Discovery** – when possible, the driver should automatically detect devices.
- **Observing** – the driver should make sensor data available to other services on the Controller Module.
- **Save State** – the driver should maintain the device configuration through power cycles.
- **Upgradeable** – the driver should support software updates via an admin interface.

While these capabilities are optional and used as applicable:

- **Configuration** – when applicable, the driver should support a configuration page to allow a user administrator (based upon roles and permissions) to adjust device parameters and add/remove devices.
- **Tasking** – when applicable, the driver should support tasking capabilities for the device.
- **Display** – when applicable, the driver should provide display pages to allow a user to view the observations.

b. Data Interfaces

The UML diagram shown in Figure 15 describes the various interfaces that comprise a generalized sensor driver interface.

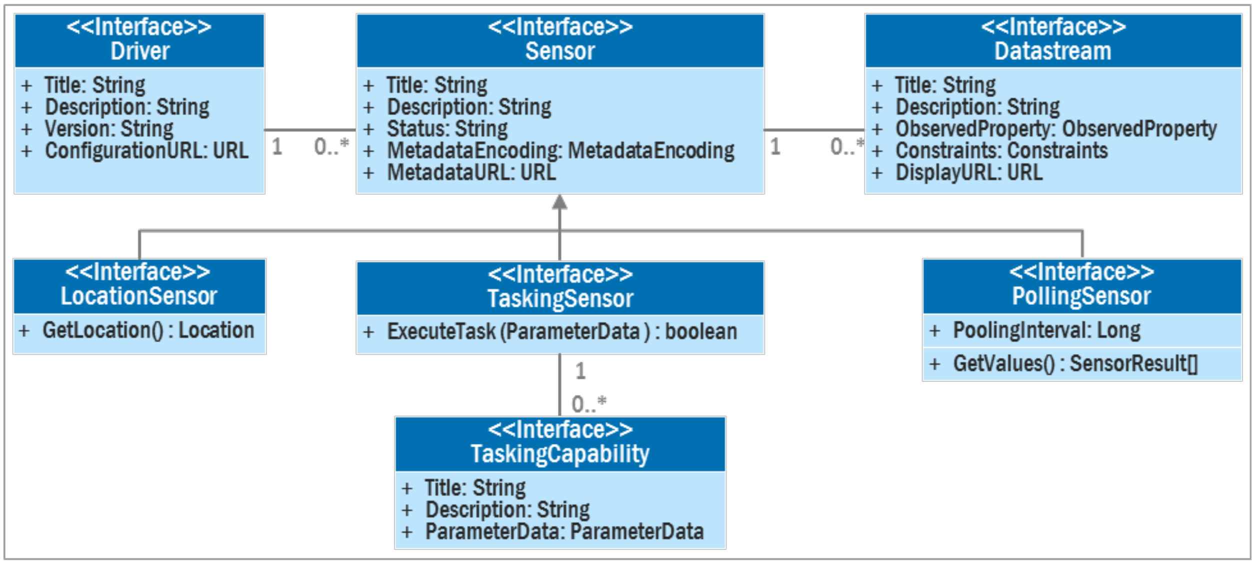

Figure 15: Sensor Driver Interface UML Diagram

c. Driver Interface

The driver interface is the main interface for a sensor driver. It should minimally provide:

- Title
  - Human readable title for the driver.
- Description
  - Human readable description for the driver.
- Version
  - The current version of the driver. For example, 1.0.0.
- Configuration URL

  The entry URL for the driver configuration page. For example, http://sensorhub.compusult.net/SensorHub/Configuration/virb_config.jsp.

d. Sensor Interface

A driver may contain zero or more sensors because you may have a driver that currently does not have devices attached. The sensor interface should minimally provide:

- Title
  - Human readable title for the sensor.
- Description
  - Human readable description for the sensor.
- Status
  - Current status of the sensor. For example, ONLINE or OFFLINE.
- Metadata Encoding

- The encoding of the sensor metadata. For example, portable document format (PDF) or SensorML.

- Metadata URL
  - The URL to retrieve the sensor metadata.

i. LocationSensor Interface

LocationSensor is an extension to Sensor and indicates a device can provide a location. This is important because this location can be used to make other devices "smarter." It should minimally provide:

- getLocation
  - A function that returns the latest location of the device.

ii. TaskingSensor Interface

TaskingSensor is an extension to the Sensor interface and indicates a device can be tasked. It should minimally provide:

- executeTask(ParameterData)
  - A function that takes defined tasking parameter data and executes the specified task.

a) TaskingCapability Interface

A TaskingSensor may contain zero or more tasking capabilities. The device may be in a state where it currently cannot be tasked and therefore may provide no capabilities. It should minimally provide:

- Title
  - Human readable title for the capability.
- Description
  - Human readable description for the capability.
- Parameter Data
  - An object that provides the acceptable parameters, if they are required and their definition (i.e., unit of measure, data type, permitted values, etc.).

iii. PollingSensor Interface

PollingSensor is an extension to Sensor and indicates a device needs to be polled for its data. It should minimally provide:

- PollingInterval
  - How often the device should be polled for values.
- getValues()
  - A function that returns the current SensorResults for a device. The SensorResult contains the datastream and its value.

iv. Datastream Interface

A Sensor may contain zero or more datastreams. The device may be in a state where it currently cannot provide data and therefore provides no datastreams. It should minimally provide:

- Title

    - Human readable title for the datastream.
- Description
    - Human readable description for the datastream.
- Observed Property
    - The property the current device observes. For example, speed, heart rate, etc. These values need to come from a defined source.
- Constraints
    - Constraints on the data of the observed property. For example, heart rate will be >= 0 beats per minute (bpm) and <= 220 bpm.
- Display URL
    - The entry URL for the observation display page. For example, http://sensorhub.compusult.net/SensorHub/Display/virb_display.jsp.

3. Controller Module-Input/Output Interface

The Controller Module to the I/O Interface provides several key capabilities. A user or administrator (based upon roles and permissions) needs to be able to view sensor information, register the Controller Module, and perform various system administration duties for the Controller Module and various attached sensor drivers. A Controller Module should support I/O access via any network connected I/O device, such as a laptop, smartphone, tablet, etc. Responders need key situational awareness information, but already have a high cognitive load and so cannot deal with irrelevant information. User interfaces therefore need to be clear and recognizable. It must be possible, though, to provide customized information to responders reacting to specific situations; in other words, the information presentation must be agile and focused on the needs of the responder.

e. Administration

A user or administrator with elevated privileges (based upon roles and permissions) needs to be able to view the status of a Controller Module and change its configuration.

f. General Capabilities

The following general capabilities are required as part of the administrative functions of a sensor hub:

- Controller Status;
- User Management;
- Rules Management;
- Driver Management;
- Connection Management;
- Data Management; and
- Device Configuration.

g. Controller Module Status

The Controller Module should provide the I/O module with a high-level status of the controller. The status information should include, but is not limited to:

- Software Version;
- Up Time;
- MAC Address;
- IP Address;
- Host;
- Service URLs;
- Power Details:
  - State of the device (i.e., plugged in, running on battery, etc.);
  - Percent of battery remaining; and
  - Estimated operational time remaining; and
- Storage Space Remaining.

h. User Management

The Controller Module should allow I/O devices to access, view and manipulate the users and their permissions within the controller. A privileged user or administrator (based upon roles and permissions) should be allowed to create or manage users and their associated permissions. Controller Modules need to operate in disconnected operations, so local user management is important. Permissions should be used to limit access to a Controller Module to specific services or data within a service as agency policy dictates.

i. Rules Management

The controller should allow a user or administrator (based upon roles and permissions) to create complex Boolean logic rules, that when matched can trigger the controller to perform an action. Actions can include tasking devices or sending alerts by a variety of channels, including email, text messages and MQTT topics. Email and text support allow for existing devices without specialized applications to receive the alerts, while MQTT delivers alerts to applications incorporating MQTT clients.

j. Driver Management

The Controller Module should allow a user or administrator (based upon roles and permissions) to upload and configure drivers that connect the sensor or devices to the Controller Module. Some sensors and devices may have the capability to register directly with the services running on the Controller Module; however, some devices may just be connected directly to the Controller Module, and therefore it will be responsible for making their data available in the services. This process may require manual configuration.

k. Connection Management

The Controller Module should allow a user or administrator (based upon roles and permissions) to configure any external connections from the Controller Module to other systems or controllers. Specifically, the Controller Module should allow the user or administrator (based upon roles and permissions) to configure what catalog(s) it will register itself to, allowing it to be discovered externally. The Controller Module will also allow the user or administrator (based upon roles and permissions) to configure to which other Controller Modules it will push its data and how to prioritize the data transfer. It is particularly useful to push data from a Controller sensor hub to a cloud sensor hub.

l. Data Management

The Controller Module should allow a user or administrator (based upon roles and permissions) to view the current status of the device storage by indicating how much space is used and how much is still available. The user or administrator (based upon roles and permissions) should be provided with options for cleaning cached data older than a specified date and time, or to allow data to only be maintained for a specified period of time. The user or administrator (based upon roles and permissions) should also be able to clear specific sensor or types of data.

The Controller Module should also allow a user or administrator (based upon roles and permissions) to prioritize the transfer of data. The user or administrator (based upon roles and permissions) should be able to indicate the importance of specific types of data. For example, the user or administrator (based upon roles and permissions) may want audio to take precedence over video; however, gas readings may take precedence over audio. The user or administrator (based upon roles and permissions) should also be able to specify permitted reductions to data if they are necessary. For example, a user or administrator (based upon roles and permissions) may want to reduce video from 30 FPS to 10 FPS if bandwidth is an issue, or to push sensor readings less frequently than they are captured.

m. Device Configuration

The Controller Module should allow a user to modify any device configuration settings. These settings may include:

- Hostname configuration;
- Email configuration;
- MQTT configuration;
- SMS configuration;
- Date and time configuration; and
- Default geospatial location of the device (if no GNSS is present).

n. View Information

Controller Modules provide a variety of information from their associated sensors (e.g. location, single readings or continuous readings (data streams).) While this information may be useful in and of itself, often a responder will want that information displayed in the larger context of their mission. This requires the ability to aggregate sensor information and display it on a map, in a table, etc. Consequently, the Controller Module needs to provide sensor information to the I/O device in a meaningful and easily recognizable format. If the I/O device cannot interpret the information, it may not be able to display that information in a meaningful way to the user. The sensor drivers and Controller Module producers should work towards common representations of various types of sensor information, so that information can be displayed in a meaningful fashion.

o. Register Sensor Hub Services

One important aspect of the Controller Module ecosystem is the ability to discover sensor hub services with which the responder can communicate. By registering with a sensor hub catalog with a unique identifier, Controller Module sensor hub services can be distinguished from each other and allow discoverability of the available sensor hub services. Discoverability is dependent on a Controller Module knowing how to communicate with a sensor hub catalog or its associated publishing service. Once a sensor hub service on a Controller Module has been configured to communicate to one of these services, the sensor hub service is able to add itself to the connected service, which then retrieves the sensor hub service capabilities, adds the capabilities to the catalog and returns a unique identifier for the sensor hub service to use in later updates.

4. Controller Module-Power Module Interface

The Controller Module is expected to have an application and driver to communicate with the power module. This application is expected to provide information to the first responder regarding the status of the power module, the status of any connected batteries and the status of any connected devices. Additional specifications regarding the Controller Module-Power Module interface are provided in Part 3. Section IV.E.4 of this handbook.

F. Application Patterns

Application patterns provide design templates for Controller Module applications through which a Responder SmartHub user interacts with actionable information. The basic applications expected to be included in the Controller Module are (this is not an exhaustive list):

- Messaging (SMS, e-mail);
- CAD interface to receive dispatch information, and send status updates or additional information to PSAP systems;
- Camera/voice recording and display/playback;
- Voice-to-text for messaging and application commands;
- Text-to-speech for incoming messages and alerts;
- Map display, including layer filtering/selection and position display;
- Communications system management configuration, status, display, operation;
- Off-body sensor system management, configuration, status, data display;
- Responder physiological sensor system management, configuration, status, data display;
- Alerting system management, configuration, display;
- Web browser for access to enterprise network and internet;
- Responder logon, identification, credentialing; and
- Agency database query and response.

V. Acronyms

Acronym	Definition
6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
AES	Advanced Encryption Standard
BLE	Bluetooth Low Energy
BPM	Beats Per Minute
CAD	Computer Aided Dispatch
CAP	Common Alerting Protocol
CoAP	Constrained Application Protocol
COMSEC	Communications Security
CSW	Catalog Service for the Web
DDS	Data Distribution Services
DE	Distribution Element
DHS	Department of Homeland Security
EXDL	Emergency Data Exchange Language
FPS	Frames Per Second
FQDN	Fully Qualified Domain Name
GeoJSON	Geographic JavaScript Object Notation
GIS	Geospatial Information System
GML	Geographical Markup Language
GNSS	Global Navigation Satellite System
HAZMAT	Hazardous Material
HDMI	High Definition Multimedia Interface
HM	Hybrid Module
HSI	Human Systems Interface
HTTP	Hypertext Transfer Protocol
HUD	Heads Up Display
I/O	Input/Output
IC	Incident Commander
INFOSEC	Information Security
iOS	iPhone Operating System

Acronym	Definition
IoT	Internet of Things
IP	Internet Protocol
IR	Infra-red
JSON	JavaScript Object Notation
LMR	Land Mobile Radio
LTE	Long-Term Evolution
M2M	Machine to Machine
MAC	Media Access Control
MDM	Mobile Device Manager
MQTT	Message Queuing Telemetry Transport
NFPA	National Fire Protection Association
NGFR	Next Generation First Responder
NIEM	National Information Exchange Model
NIST	National Institute of Standards and Technology
NMEA	National Marine Electronics Association
OGC	Open Geospatial Consortium
OWS	Open Geospatial Consortium Web Service
P25	Project 25
PDF	Portable Document Format
PSAP	Public Safety Access Point
PTT	Push To Talk
PTZ	Pan-Tilt-Zoom
S&T	Science and Technology Directorate
SA	Situational Awareness
SensorML	Sensor Markup Language
SLTT	State, Local, Tribal and Territorial
SMS	Short Message Service
SNRA	Sensor Network Reference Architecture
SNS	Sensor Notification Service
SOS	Sensor Observation Service

Acronym	Definition
STAPI	Sensor Things API
TCP	Transmission Control Protocol
TRRS	Tip-Ring-Ring-Sleeve
TRS	Tip-Ring-Sleeve
UID	User Identification
UML	Universal Markup Language
URL	Universal Resource Language
USB	Universal Serial Bus
US_CERT	United States Computer Emergency Response Team
UUID	Universally Unique Identifier
VAC	Volts Alternating Current
WFS	Extensible Messaging and Presence Protocol Web Processing Service Web Map ServiceWireless FidelityWeb Feature Service
Wi-Fi	Trademark for 802.11 Standards
WMS	Web Map Service
WPS	Web Processing Service
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol

1