Inovonics Cloud

Integration Guide

Preface

Trademarks and Copyrights

- EchoStream® is a registered trademark of Inovonics Corporation

Contact Information

- Inovonics Technical Services. support@inovonics.com or 1.800.782.2709.

Document Conventions

- TBD

Revision History

Revision	Name	Description	Date
0.1	Kevin Stoner	Initial Revision	12/22/21
0.2	Kevin Stoner	Accepting Todd's Review	12/30/21
0.3	Kevin Stoner	Added Hardware Part Numbers Table	12/31/21
0.4	Kevin Stoner	Adding Cooper's Review Changes	1/12/22
0.5	Kevin Stoner	Fixing Date	1/12/22
0.6	Kevin Stoner	Updating example URL	1/12/22
0.7	Kevin Stoner	Updating for Floors and Units	10/1/22

Table of Contents

1. Introduction

1.1 Purpose

This document is intended to provide high level guidance for integrating your application with the Inovonics Cloud system.

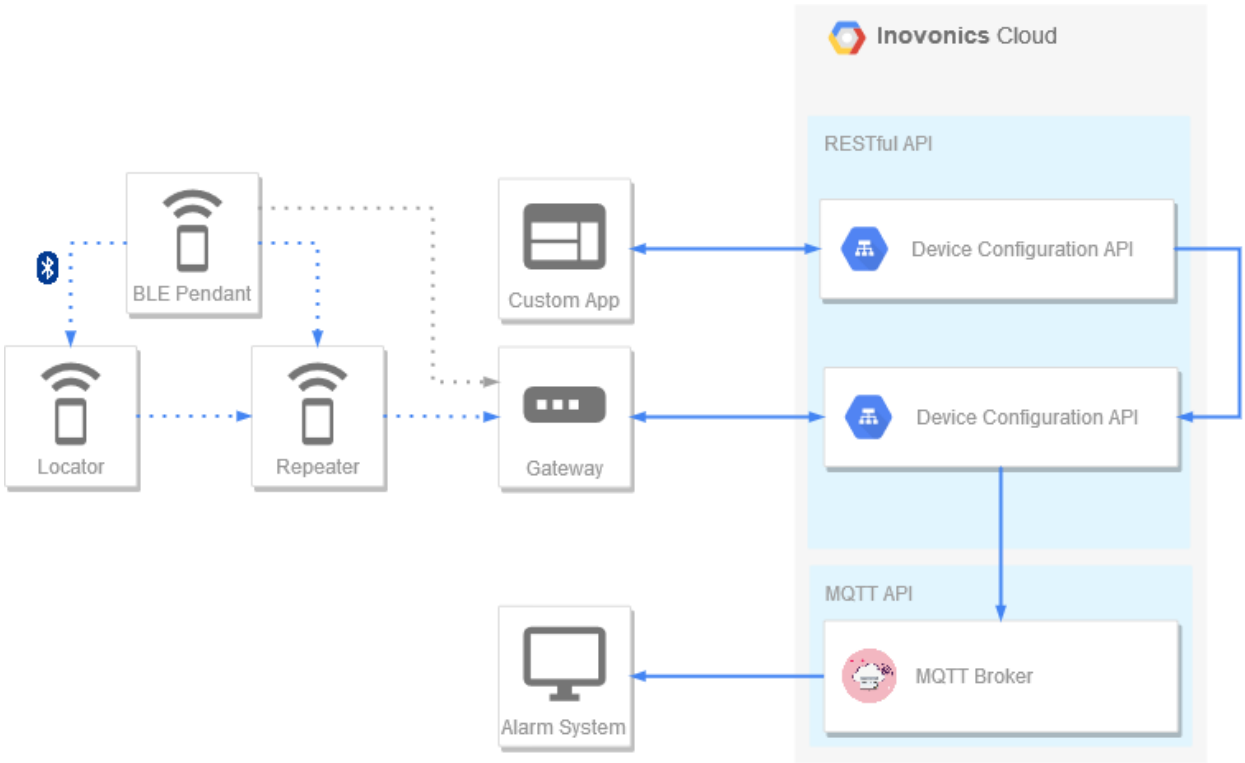
1.2 Reference Documents

The following documents are related to this project:

- TBD: Put documents here

2. System Overview

The Inovonics Cloud system is designed with the objective to allow the configuration and monitoring of Inovonics devices through the cloud. Some common uses of the system are receiving locations of alarm events, processing of fall detection alarm events and associated algorithm data, and general device health monitoring.

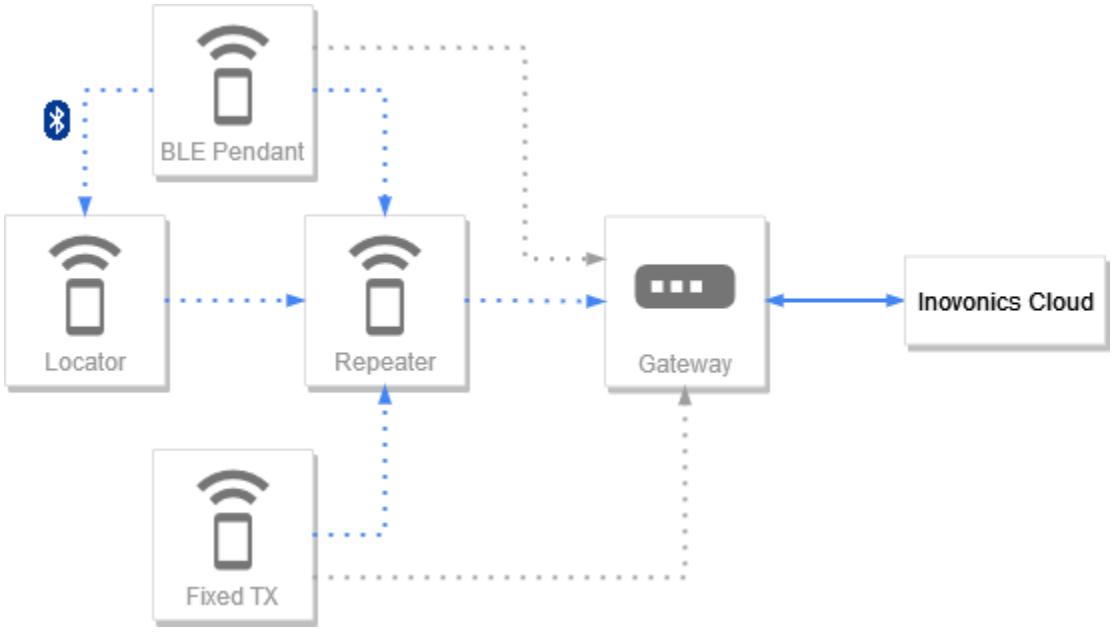


In the above diagram, a custom developed application may leverage RESTful APIs in order to configure and set up a site with devices in the Inovonics Cloud. Physical devices on each site will communicate wirelessly over the EchoStream network, which will be heard and sent

up to the Inovonics Cloud via a gateway. From there, important events such as alarm locations will be sent to the Inovonics MQTT Broker. A custom MQTT client can then subscribe to the broker and handle incoming events in the appropriate manner. Each of these components will be discussed in further detail later in this document.

3. Hardware Components

Sites are made up of a variety of Inovonics devices. These devices leverage the Inovonics EchoStream protocol in order to communicate wirelessly with high reliability at the site. The messages then are collected and pushed up to the cloud by a Gateway. In some cases an Inovonics Gateway can be added to a pre-existing EchoStream system in order to allow cloud access to the site, while leveraging the on-premise network for local/primary alarming.



3.1 Fixed Transmitters

Fixed transmitters are Inovonics EchoStream transmitting devices that are designed to remain in a given location at a site. Examples of fixed transmitters are door sensors and smoke detectors. These devices will transmit events over the EchoStream network which will be collected by the Gateway on the site.

3.2 Mobile Pendants

Mobile pendants are transmitter devices that are to be worn by facility residents and staff. Traditional mobile pendants will only transmit EchoStream messages when the wearer activates the pendant by pressing a button. Newer pendants such as the Inovonics Fall Detect Pendant allow for fall detection alarm processing, as well as location by sending

Bluetooth messages in addition to the standard EchoStream messages. This in conjunction with strategically placed locators allow the Inovonics Cloud to locate the pendant within a site.

3.3 Locators

Locators are a Bluetooth to Echostream “bridge” that can be installed throughout a site in order to locate Bluetooth equipped mobile pendants when they alarm. The locator listens for Bluetooth messages from pendants on the site, and will convert those messages to Echostream. Those messages can be used to determine the closest locator to a mobile pendant when it is activated.

3.4 Repeaters

Repeaters listen for EchoStream messages and repeat them by re-transmitting the message. They are used for extending the range of the EchoStream network on the site. To meet certain standards, repeaters on some sites must run in Directed Messaging mode. For more information about this please see: TBD

3.5 Gateway

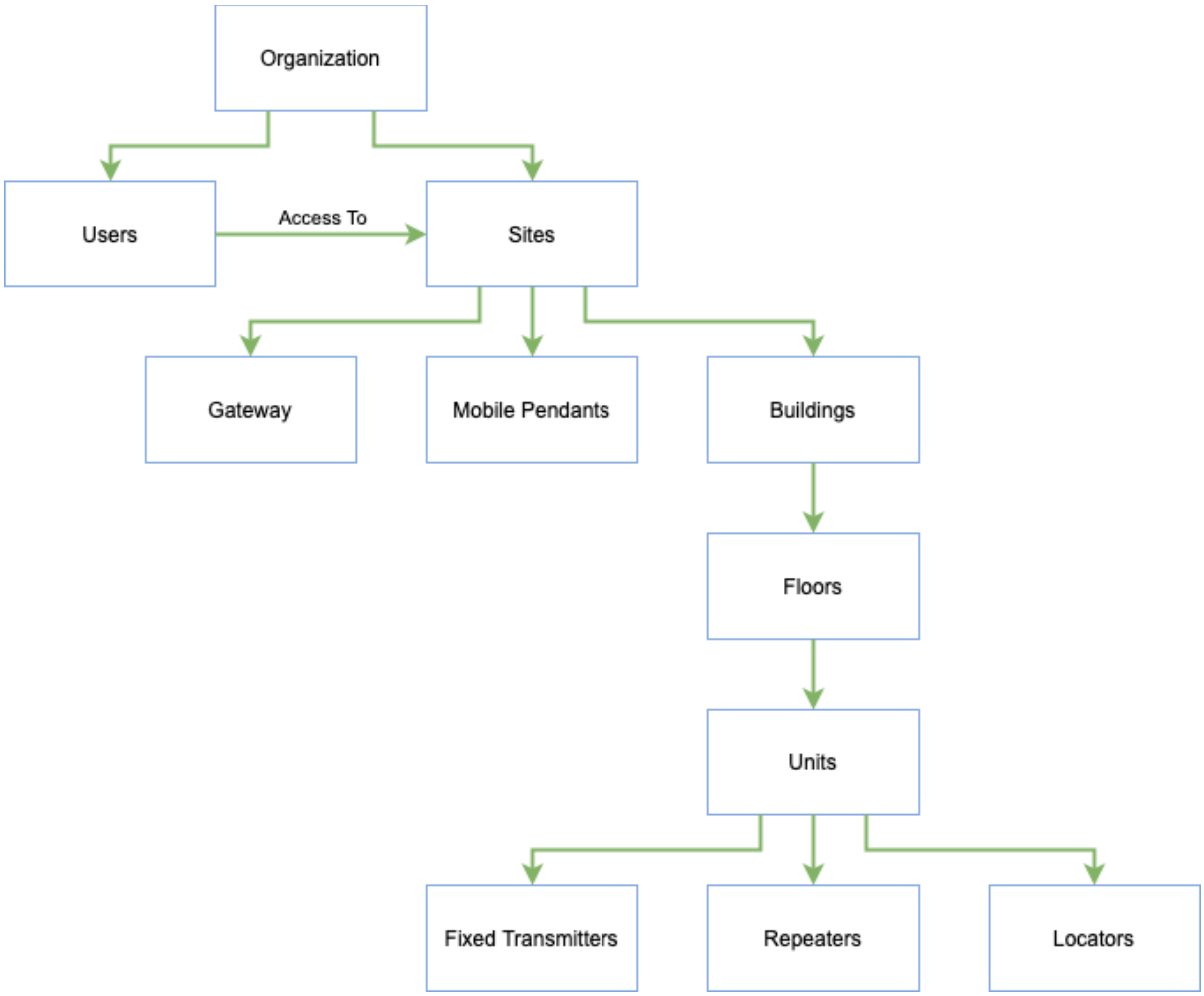
The Gateway is the connection between the site and the cloud. The Gateway will listen for, and collect EchoStream messages using the built in receiver, and will relay them up to the Inovonics Cloud for further processing and storage. In the case of network issues between the gateway and the site, the gateway is able to store messages until it is able to once again communicate with the cloud. The gateway must be added to a site in the Inovonics Cloud in order to pull site configuration information and send up EchoStream traffic from devices.

4. RESTful API

The RESTful API is used for configuring sites and administering users. Site configuration involves registering the devices that will be used on the site to the cloud. When the configuration is created or changed, it is pushed down to the gateway on the site. Configuration information is also used in order to send out more detailed information during events and alerts. The RESTful APIs also allow for the administration of users. Users can be given different permission levels and access in order to be able to view and modify sites within their organization.

4.1 Organization Hierarchy

The organization hierarchy as pictured below demonstrates the relationship of different objects within the Inovonics Cloud. When creating new sites it is recommended to follow this pattern in order to avoid any issues with compatibility.



4.1.1 Organizations

At the top most level, everything belongs to an organization. An organization can be considered a partner of Inovonics. Organizations are created by Inovonics as well as the first few users within the organization, to get the partner started. From there it is on the partner to create the subsequent users and sites.

4.1.2 Users

Users are members of the organization that need to create, view, or modify sites. Users are NOT the residents of the site. Every user within an organization requires a unique email address, even if the user is a service account user. There are three different permission levels users can be assigned: Administrator, Technician, and Viewer. Administrators are allowed to create and modify sites, as well as administer other users within the organization. Technicians are allowed to create and modify sites but cannot administer users. Viewers can only view sites, but cannot make any changes otherwise. Besides assigning roles, Administrators can optionally restrict access for other users to certain sites over certain timeframes.

4.1.3 Sites

Sites are a specific place that an EchoStream network of Inovonics devices are installed (e.g. a hospital). Sites may contain many devices, with the only restriction being that each site must only have one gateway.

4.1.4 Buildings

Buildings can be considered a type of folder to organize the site. A smaller site may only have one building, but larger sites may have multiple buildings. While it is not necessary to leverage buildings, floors and units, it is recommended to leverage them as it helps to better define the location of each device. Mobile pendants should not be tied to a building as they may move freely across the site.

4.1.5 Floors

Floors represent a specific floor within a building. A building may have one or more floors within it. Floors may have a floor plan attached to them in order to indicate where units and devices are on the floor. Floors have an ordering that specifies how the floors stack up from lowest to highest. The ordering may be used in order to improve accuracy in location determination.

4.1.6 Units

Units represent specific room (or area) within a floor. A unit may contain multiple devices within it that are tied to that specific room/area. Units can be placed on a floor plan using annotations to define where the unit is within a floor.

4.1.7 Devices

The types of devices on each site are the same as defined in the *Hardware Components* section of this document. With the exception of the gateway, devices must be registered by entering the TXID of the device. This is the number found usually on a sticker on the outside casing of the device. Sites must have a unique set of TXIDs on them, however it is possible to use the same TXID on two separate sites.

4.2 REST API Requirements

The URL for making API calls will be provided by Inovonics once the Organization is set up for a partner. All API requests require that a user is authenticated via OAuth 2 and that the connection is secured by TLS. Users making requests must have the correct access and permissions to the resource they are requesting. For instance a Viewer may not make a request to rename a site. Upon authenticating with the API, the user will be provided a Bearer token that can be used in subsequent requests to verify their authentication.

4.3 Request Format

With a few exceptions, every API request (and response) will be in JSON format. In order to make a request, it is required to specify the content-type of the body (usually "application/json") in the headers of the request. There are a few API requests that accept no body as well as an import API that accepts spreadsheets. More information about these requests can be found in the API documentation. The authentication (bearer) token should also be placed in the header of each request. Every API endpoint in the Inovonics Cloud ends in a forward slash ("/") character, if the character is left off an error will be returned. In general, HTTP verbs are used to perform the associated commands. To add a new site, the user would send a POST request. To remove the site, the user would send a DELETE request. To modify the site, the user would send a PUT request. To retrieve information about a site, the user would send a GET request.

Example Request to Add Site:

Request:

```
POST https://security-
api.inovonics.com/v1/organizations/<organization_id>/sites/
```

Headers:

```
Authentication: Bearer <access key>
Content-Type: application/json
```

Body:

```
{
   "name": "a new site",
   "code": "ans",
   "timezone": "US_EASTERN"
}
```

4.4 Response Format

The response of a request also follows standard HTTP protocol. For example if a resource does not exist, a 404 NOT FOUND response will be returned. Other common error codes returned are 409 CONFLICT for attempting to create something that already exists (e.g. a transmitter with a duplicate TXID), 403 FORBIDDEN, 401 UNAUTHORIZED, and 400

INVALID DATA for attempting to make a poorly formatted request. For valid requests, a 200, 201 or 204 may be returned. A 201 indicates that something new was created in the case of a POST request, and a 204 indicates that there is no return body in the response. Please refer to the API documentation for a complete list of request and response bodies and codes.

Example Response to Add Site:

Response:

```
201 Created
```

Body:

NOTE: Some fields have been left out of this example to keep it concise.

```
{
   "address": "",
   "code": "ans",
   "name": "a new site",
   …
   "site_id": "<site_id>",
   "timezone": "US_EASTERN"
}
```

4.5 Referencing Objects

Upon creation, all objects such as users, sites, devices, buildings, floors and units will generate a unique ID which is returned in the response. The ID is formatted as a GUID and should be used to reference each object in subsequent API requests. This ID should not be confused with the TXID of a device which is a separate ID used to tie a physical device to its cloud object representation.

4.6 Enum APIs

There are a few APIs that exist for the sole purpose of providing values that can be entered into other APIs. For example, a developer might call the transmitter model API to find what values they can use for the model of a new transmitter device.

4.7 Import/Export APIs

As an alternative to configuring a site with individual APIs, a user may leverage the import/export APIs to import in an Excel spreadsheet containing an entire site's configuration. In order to do this, the site must be created first using a POST command. An export spreadsheet may then be created by doing a GET on `/sites/<site_id>/export`. Once changes have been manually applied to the spreadsheet, it can be re-imported with a PUT on `/sites/<site_id>/import/`.

4.8 Report APIs

A few report APIs exist for the purpose of pulling historical data or snapshots in time. These reports should not be leveraged for real time event handling. Instead the MQTT API should be used to respond to device events.

4.9 Creating Relationships

All objects such as devices, buildings, floors and units must be added to the site initially with a POST request. After adding the object to the site, further relationships can be created by performing PUT requests between two objects. For example when adding locators, each locator must first be added directly to the site (POST request on `/sites/<site_id>/transmitters/`). From there in order to add a locator to a unit, the relationship can be created between a device and unit within the same site (PUT request on `/sites/<site_id>/units/<unit_id>/transmitters/<transmitter_id>/`).

5. MQTT API

In order to receive location and fall information for alarms as well as other device notifications, the Inovonics MQTT API can be leveraged. Clients that are subscribed to the MQTT broker will receive events and notifications as the cloud processes them without having to poll the RESTful API for information.

5.1 MQTT Integration Setup

MQTT clients do not use the same authentication as the RESTful API, the authorization credentials (username/password) must be configured by creating an MQTT integration. There are two ways that MQTT integrations can be set up to allow a client to listen to MQTT traffic: at the site level, or at an organization level. MQTT integrations at the site level will allow for a unique set of credentials to be assigned to each site. These credentials can be used to connect and subscribe to MQTT traffic for a specific site. Alternatively an organization level integration creates a single set of credentials for the entire organization that is able to listen to traffic for all sites within the organization. Only one type of integration may be used at a time per organization, so either each site must have an individual integration or the organization itself must have one integration, but not both. Each type of integration can be set up with RESTful APIs (see REST API documentation). There are

unique API calls for creating an organization level integration and site level integrations respectively.

5.2 Connecting to the MQTT Broker

The MQTT broker URL will be provided by Inovonics upon requesting to do an MQTT integration. The broker allows for both TLS encrypted (port 8883) and unencrypted (1883) connections however it is highly recommended to use an encrypted connection for production sites. The Inovonics broker has a CA signed certificate that is used for encrypted connections. Currently the Inovonics MQTT Broker uses MQTT version 5.0. In order to connect the username and password configured in the MQTT integration will need to be used. The client ID for the connection must be globally unique so it is recommended to use a random string of characters for the client ID. In the future there may be further restrictions on the client IDs that can be used. When connecting it is also recommended to set the "Clean Session" flag to False. This will allow a client that disconnects, to reconnect and receive any messages that it missed. Currently the Inovonics MQTT broker will store unsent messages for up to 1 day.

5.3 MQTT Subscriptions

MQTT Subscriptions can be made after connecting with the client. In general Inovonics recommends subscribing to MQTT topics with a QOS of 2 in order to prevent missing or duplicate messages. Currently there is a common "status" topic that can be leveraged for determining the health of the cloud. Every authenticated MQTT user automatically has access to this topic. The rest of the topics are restricted by MQTT username. This prevents an MQTT integration for one site from listening to traffic from another site. All MQTT payloads are in JSON format.

Appendix A: Inovonics Hardware Part Numbers

Hardware Component	Part Numbers	Application		
		Mobile Duress	Senior Living	TapWatch
Gateway	EN7295	■		
	EN7380		■	
	EN7580			■
Repeater	EN5040	■	■	■
	EN5040-T	■	■	■
	EN5040-20T		■	
Locator	EN5060	■	■	
BLE-Enabled Pendants	EN2221S-60		■	
	EN2222S-60		■	
	EN2224	■		
	EN2233D	■		
	EN2233S	■		
	EN2235D	■		
	EN2235S	■		
	EN2236D	■		
	EN2238D	■		