

Scenario #8 (EIDO Subscription for Call State Update Notifications)

[**1 Scenario Description**](#_jhk1o5nzsgs1) **1**

[**2 Prerequisites**](#_l7xm3kdv4629) **2**

[**3 Detailed Steps**](#_ev8chr9r0p5j) **2**

[3.1 Subscription Establishment for all existing and new Incidents/Calls](#_wdh3vfu5m2ko) 2

[3.2 New Incident/Call is Created](#_e0jbwhoio684) 3

[3.3 MinRate Timer Expires](#_5yd51iyqc1wl) 3

[3.4 Subscription Renewal](#_3nptvr9duyhs) 4

[3.5 Subscription for Individual Incident/Call](#_guuu3jch6qap) 4

[3.6 Existing Incident/Call State is Updated](#_1y8i5yiyilni) 5

[3.7 Notifier Terminating a Subscription](#_z7wltc6827aj) 5

[3.8 Subscriber Terminating a Subscription](#_5vmnbx2m38un) 6

# 1 Scenario Description

A Subscriber interested in receiving notifications of call state changes (such as a CAD or an IDX) subscribes to a Notifier which can provide such notifications (such as a Call Handling system or an IDX acting as a data exchange hub).

The subscriber establishes a web socket connection to one or more provisioned notifiers and once established, makes a “subscribe” request of type EIDO specifying a subscription for all existing and new Incidents/Calls. Once the subscription is accepted, the Notifier will send a single notification for every existing Incidents/Calls and a single notification for every new Incident/Call. If the Subscriber needs to receive further notifications for an Incident/Call, it must send a “subscribe” request for each individual Incident/Call. These individual Incident/Call subscriptions will be torn down when the Incident is closed/Call is released.

A subscription may specify minimum and maximum rate filters which will control the rate notifications are sent.

A subscription for all new Incidents/Calls may specify a “qualFilter” which will be used to filter notifications to a specified subset (the “qualFilter” may specify a geofence or a specific Incident type(s) the subscriber is only interested in receiving).

The Subscriber must renew subscriptions before they expire else the Notifier will stop sending notifications.

# 2 Prerequisites

* Subscriber is provisioned with the URI(s) of the Notifier(s)
* Subscriber and Notifier(s) are connected to VPN

# 3 Detailed Steps

## 3.1 Subscription Establishment for all existing and new Incidents/Calls

1. Subscriber establishes a connection to a Notifier via a WebSocket per RFC 6455 specifying a WebSocket Protocol named “emergency-ent”.
2. If Secured WebSocket (wss), both parties must mutually authenticate each other using certificates traceable to the PCA.
3. Once the WebSocket is established, Subscriber sends a “subscribe” request with the following elements:
   1. “requestId”: mandatory element containing an identifier to be used to match responses with the request
   2. “requestType”: must be “EIDO”
   3. “requestSubType”: must be “new”
   4. “requestAccepts”: must be ‘application/emergency.eido+json; version=”1”’
   5. “qualFilter”: optional
   6. “qualFilterEvalPeriod”: optional if “qualFilter” is specified else must not be specified
   7. “minRate”: optional
   8. “maxRate”: optional
   9. “subscriptionId”: must not be provided as there is no existing subscription
   10. “incidentId”: must not be provided
   11. “expires”: optional
4. If subscription request is accepted Notifier will return a “subscribeResponse” with the following elements:
   1. “requestId”: must be the same as in the “subscribe” request
   2. “subscriptionId”: must be specified
   3. “qualFilterEvalPeriod”: must be specified if “subscribe” request included a “qualFilter” element
   4. “minRate”: must be specified if “subscribe” request included a “minRate” element
   5. “maxRate”: must be specified if “subscribe” request included a “maxRate” element
   6. “Expires”: must be specified
   7. “statusCode”: must be 200
   8. “statusText”: must be specified
5. Notifier then proceeds to send one or more “event” notifications containing an EIDO for every active Incident/Call, each notification contains the following elements:
   1. “subscriptionId”: must be the same as in the “subscribeResponse”
   2. “transactionId”: A GUID used to identify a notification for acknowledgement
   3. “Notification”: an array of one or more EIDOs, one for each active Incident/Call
6. For each received “event” message, Subscriber acknowledges reception with a “eventResponse” message with the following elements:
   1. “transactionId”: must be the same as in the “event” message
   2. “statusCode”: must be 200
   3. “statusText”: must be specified

## 3.2 New Incident/Call is Created

When one or more new Incident/Call is(are) created and maxRate timer allows for a notification to be sent, Notifier must send one “event” notification containing an EIDO for every new Incident/Call that has been created since the last time a notification was sent.

1. The notification contains the following elements:
   1. “subscriptionId”: must be the same as in the “subscribeResponse”
   2. “transactionId”: A GUID used to identify a notification for acknowledgement
   3. “Notification”: an array of one or more EIDOs, one for each new Incident/Call
2. Subscriber acknowledges reception with a “eventResponse” message with the following elements:
   1. “transactionId”: must be the same as in the “event” message
   2. “statusCode”: must be 200
   3. “statusText”: must be specified

## 3.3 MinRate Timer Expires

If a minRate value is specified and there are no new Incident/Call to report when the minRate timer expires, Notifier must send an empty “event” message where the “Notification” array is empty and Subscriber must acknowledge with an “eventResponse”

## 3.4 Subscription Renewal

If the Subscriber wishes to refresh an active subscription that is about to expire:

1. Subscriber sends a “subscribe” request with the following elements:
   1. “requestId”: mandatory element containing an identifier to be used to match responses with the request
   2. “requestType”: must be “EIDO”
   3. “requestSubType”: must be “new”
   4. “requestAccepts”: must be ‘application/emergency.eido+json; version=”1”’
   5. “qualFilter”: optional
   6. “qualFilterEvalPeriod”: optional if “qualFilter” is specified else must not be specified
   7. “minRate”: must not be provided
   8. “maxRate”: must not be provided
   9. “subscriptionId”: must be the same as in the active subscription
   10. “incidentId”: must not be provided
   11. “expires”: optional
2. If subscription renewal is accepted Notifier will return a “subscribeResponse” with the following elements:
   1. “requestId”: must be the same as in the “subscribe” request
   2. “subscriptionId”: must be the same as in the active subscription
   3. “qualFilterEvalPeriod”: must be specified if “subscribe” request included a “qualFilter” element
   4. “minRate”: must be specified if “subscribe” request included a “minRate” element
   5. “maxRate”: must be specified if “subscribe” request included a “maxRate” element
   6. “Expires”: must be specified
   7. “statusCode”: must be 200
   8. “statusText”: must be specified

## 3.5 Subscription for Individual Incident/Call

As mentioned above, a subscription for “new” Incident/Call will only provide a single notification. To receive further notifications for an individual Incident/Call:

1. Subscriber sends a “subscribe” request with the following elements:
   1. “requestId”: mandatory element containing an identifier to be used to match responses with the request
   2. “requestType”: must be “EIDO”
   3. “requestSubType”: must be “single”
   4. “requestAccepts”: must be ‘application/emergency.eido+json; version=”1”’
   5. “qualFilter”: must not be specified
   6. “qualFilterEvalPeriod”: must not be specified
   7. “minRate”: optional
   8. “maxRate”: optional
   9. “subscriptionId”: must not be provided as there is no existing subscription
   10. “incidentId”: must be provided
   11. “expires”: optional
2. If subscription renewal is accepted Notifier will return a “subscribeResponse” with the following elements:
   1. “requestId”: must be the same as in the “subscribe” request
   2. “subscriptionId”: must be specified
   3. “qualFilterEvalPeriod”: must not be specified
   4. “minRate”: must be specified if “subscribe” request included a “minRate” element
   5. “maxRate”: must be specified if “subscribe” request included a “maxRate” element
   6. “Expires”: must be specified
   7. “statusCode”: must be 200
   8. “statusText”: must be specified

## 3.6 Existing Incident/Call State is Updated

When the state of an existing Incident/Call is updated and maxRate timer allows for a notification to be sent, Notifier must send one “event” notification containing a single EIDO.

1. The notification contains the following elements:
   1. “subscriptionId”: must be the same as in the “subscribeResponse”
   2. “transactionId”: A GUID used to identify a notification for acknowledgement
   3. “Notification”: an array of one EIDO
2. Subscriber acknowledges reception with a “eventResponse” message with the following elements:
   1. “transactionId”: must be the same as in the “event” message
   2. “statusCode”: must be 200
   3. “statusText”: must be specified.

## 3.7 Notifier Terminating a Subscription

A notifier sends a “terminate” message when a subscription is terminated such as due to subscription expiration or when an incident is closed by the notifier:

1. Notifier sends a “terminate” message with the following elements:
   1. “requestId”: mandatory element containing an identifier to be used to match responses with the request
   2. “subscriptionId”: Identifier of the subscription being terminated.
2. Subscriber acknowledges reception with a “terminateResponse” message with the following elements:
   1. “requestId”: must be the same as in the “terminate” message
   2. “subscriptionId”: Identifier of the subscription being terminated
   3. “statusCode”: must be 200
   4. “statusText”: must be specified.

## 3.8 Subscriber Terminating a Subscription

If the Subscriber wishes to terminate the subscription:

1. Subscriber sends an “unsubscribe” message with the following elements:
   1. “requestId”: mandatory element containing an identifier to be used to match responses with the request
   2. “subscriptionId”: Identifier of the subscription being terminated.
2. Notifier acknowledges reception with a “unsubscribeResponse” message with the following elements:
   1. “requestId”: must be the same as in the “terminate” message
   2. “subscriptionId”: Identifier of the subscription being terminated
   3. “statusCode”: must be 200
   4. “statusText”: must be specified.