

```
<na:time-period>
  <time dtstart="19970105T083000"
    timestart="2200"
    timeend="0800"
    byweekday="MO,TU,WE,TH,FR"
    dtend="19991230T183000"/>
</na:time-period>
</conditions>
<actions>
  <priority>5</priority>
<na:route>

<na:recipient>sip:answering-machine@home.foo-bar.com
  </na:recipient>
  </na:route>
</actions>
<transformations/>
</rule>
</ruleset>
```

4.4.2.5 Namespace

This document uses the NENA URN namespace "urn:na:policy-v1".

4.5 LoST

LoST is the protocol that is used for two functions: call routing and location validation.

- Call routing: LoST is used by the ECRF as the protocol to route all emergency calls both to¹⁰ and within the ESInet.
- Location validation: LoST is used by the LVF as the protocol to validate location information for every call origination end device prior to any potential use for emergency call routing.

Each LoST message is an XML-based document. The root element within each LoST message has the same name as the LoST message name and contains attributes and other elements. In section

¹⁰ LoST must be used within an ESInet to route calls. It is recommended that originating networks also use LoST to route calls to the entry ESRP, but they may use appropriate local functions provided calls are routed to the same ESRP as would the use of LoST to the ECRF.

4.5.1 and its sub-sections, XML attributes are denoted by “attributeName” and XML elements by “<elementName>” (e.g., sourceId and <displayName>).

In the following sections, there is text that explains how LoST works. The normative reference that defines the protocol is RFC5222 [61]. The text in this section that defines LoST protocol operations should be considered informative, and any discrepancies are resolved by RFC5222 text. The text below does contain limitations and specific application of LoST operations that are normative. A future edition of this document will remove some of the informative text and highlight the normative text.

4.5.1 Emergency Call Routing using LoST

All SIP-based emergency calls pass location information either by value (PIDF-LO) or by reference (Location URI) plus a "Service URN" to an Emergency Services Routing Proxy (ESRP) to support routing of emergency calls. The ESRP passes the Service URN and location information¹¹ via the LoST interface to an Emergency Call Routing Function (ECRF), which determines the next hop in routing a call to the requested service. The ECRF performs the mapping of the call's location information and requested Service URN to a “PSAP URI” by querying its data and then returning the URI provided. Using the returned URI and other information (time-of-day, PSAP state, etc.), the ESRP then applies policy from a Policy-based Routing Function (PRF) to determine the appropriate routing URI. This URI is the address for the "next hop" in the call's routing path that could be an ESRP URI (intermediate hop), a PSAP URI (final hop), or even a call-taker (see section 5.3 for a more detailed functional explanation of the i3 ECRF).

A single emergency call can be routed by one or more ESRPs within the ESInet, resulting in use of the LoST interface once per hop as well as once by the terminating PSAP.

Note that the term “PSAP URI” is used within the LoST protocol definition to refer to the URI returned from the service URN urn:service:sos. In NG9-1-1, the URI returned may not be that of a PSAP, but instead may route to an ESRP.

4.5.1.1 LoST Call Routing Messages

The LoST interface message used to query for the next hop within the ESInet is the <findService> message. The LoST interface message used to return the result of processing a <findService> request message is the <findServiceResponse> message. The ECRF receiving the <findService> message translates the Service URN and location information in the message into a next-hop URI, which is returned in the <findServiceResponse> message to the querying entity. If the ECRF cannot

¹¹ If an element using LoST receives location by reference, it must dereference the URI to obtain the value prior to querying the LoST server. The LoST server does not accept location by reference.

successfully process a <findService> message, it returns an <error> message. The following three sections describe these messages.

4.5.1.1.1 LoST <findService> Request Message

A querying entity (e.g., ESRP, VoIP-based endpoint, Legacy Network Gateway, Legacy PSAP Gateway, PSAP) uses the <findService> message to retrieve one or more contact URIs from an ECRF given a Service URN and a location. This message contains elements and attributes specified in Table 4-1. Note the "Name" column contains the actual <findService> message's attribute and element names as defined by the LoST protocol.

Table 4-1 – LoST <findService> Message Attributes and Elements

Name	Condition	Purpose
xmlns	Mandatory	This attribute specifies the LoST protocol's XML namespace.
<location>	Mandatory	This element contains either civic address- or geodetic coordinates-based location information.
recursive	Optional	This attribute indicates a preference for a recursive or iterative query.
<service>	Mandatory	This element contains the URN of the requested service.
<path>	Conditional	This element indicates the path the message has taken through ESRPs within the ESInet.
serviceBoundary	Optional	This attribute indicates how the service boundary should be returned to the requestor.
validateLocation	Conditional	This attribute indicates whether the civic address location should be validated.

The LoST <findService> message attributes and elements specified in Table 4-1 are described in greater detail below.

- xmlns Attribute

This required attribute must specify the LoST protocol XML namespace and is coded as follows.

```
xmlns="urn:ietf:params:xml:ns:lost1"
```

- <location> Element

This required element carries the location information used to query for routing information and has the format specified in [61]. The location information can be in the form of a civic address or geodetic coordinates. The civic address-based location information format is specified in RF4119 [6] updated by RFC5139 [76] and RFC5491 [75]. The geodetic coordinates-based location information format is specified in [75] and the supported



geographic shapes are point, polygon, circle, ellipse, and arc band. See Section 8.2 in [61] for examples of civic and geodetic-2d location information encodings.

There must be one and only one <location> element. Although the LoST protocol permits multiple <location> elements with one per unique location profile based on the same baseline location profile in a single LoST <findService> message, i3 limits the number of <location> elements to exactly one. For maximum client/server interoperability, there should be only one <location> element based on a baseline location profile in a <findService> message sent to an i3 ECRF. See Section 12 in [61] for more information about baseline and derived location profiles.

The "location" element contains many elements and attributes, some of which are described in Table 4-2.

- recursive Attribute

LoST servers can operate in recursive mode or iterative mode if a mapping is not found based on the coding of this attribute.

- The use of recursion by the ECRF initiates a query on behalf of the requestor that propagates through other ECRFs to an authoritative ECRF that returns the PSAP URI back through the intervening ECRFs to the requesting ECRF.
- The use of iteration by the ECRF simply returns a domain name of the next ECRF to contact.

This optional attribute is coded “true” to indicate recursive mode or “false” or not coded to indicate iterative mode.

The ECRF may operate in a recursive mode or an iterative mode, depending on local implementation.

- <service> Element

This required element identifies the service requested by the client. Valid service names are specified in [58] and must be "sos" or one of its sub-services for ECRFs and LVFs used by originating networks or devices. For internal ECRFs used by entities within the ESInet to route calls, the <service> element may be a service URN beginning “urn:nena”.

- <path> Element

This conditional element contains <via> elements indicating the ECRFs (LoST servers) that have handled the <findService> request as a recursive query. This element is used by ECRFs to detect a recursive query routing "loop" during recursive query processing. See Section 6 in [61] for detailed information about the <path> element.

The order of <via> elements within the <path> element is significant. The first <via> element always indicates the ECRF that received the initial <findService> message query from the requesting ESRP. The last <via> element indicates the ECRF that sent the <findService> request to the current ECRF. All <via> elements indicate the path from the initiating ECRF to the current ECRF.

The originating ESRP that sends the <findService> message to the initial ECRF does not include this element in the message; i.e., it is an error for the <path> element to exist within the <findService> message sent by any element except an ECRF.

When an ECRF receives a <findService> message, it appends its own domain name as a new last <via> element to the <path> element before forwarding the <findService> message to another ECRF or returning a <findServiceResponse> message (which contains the <path> element).

- serviceBoundary Attribute

A requesting entity can obtain the boundary of the jurisdiction or service area handled by the requested service. This is most useful for mobile devices that use geodetic coordinates since they can track their location. When they leave the jurisdictional area, they can send another <findService> request to determine the proper jurisdiction for their new location.

This optional attribute indicates whether a service boundary value or reference is preferred in the <findServiceResponse> message. The query originator can express a preference for a value or a reference using this attribute, but the ECRF makes the final decision as to whether to return a reference, a value, or even nothing.

This attribute is coded "value" to indicate the preference for returning the service boundary as a value or is omitted or coded "reference" to indicate the preference for returning the service boundary as a reference. The <serviceBoundary> element returns the service boundary "value" and the <serviceBoundaryReference> element returns the "reference".

Note that returning the service boundary as a reference passes less data in a message, using less network bandwidth, but requires later dereferencing via a LoST <getServiceBoundary> message to obtain the value, thus later using more server time and increasing call delay. Returning the service boundary as a value passes more data in a message, using more network bandwidth, but does not require later dereferencing, thus saving server time and minimizing call delay. In addition, a service boundary may require many data points to accurately identify the boundary of a jurisdiction or service area, possibly making the service boundary dataset very large.

According to [61], a LoST server may decide, based on local policy, to return the service boundary as a value or a reference, or even not to return the service boundary information by omitting both the <serviceBoundary> and <serviceBoundaryReference> elements in the <findServiceResponse> message. This means the requesting entity must handle a returned value, a returned reference, or nothing regardless of the "value" or "reference" coding or the omission of the serviceBoundary attribute in the <findService> message. ECRFs should return a service boundary if the request included the attribute.

- validateLocation Attribute

Location validation is the validation of civic address-based location information against an authoritative GIS database containing only valid civic addresses obtained from 9-1-1 Authorities.

Location validation is performed by the i3 LVF. Normally, an i3 ECRF does not perform location validation because i3 requires location information to be validated before it is passed in SIP call signaling to an ESRP; hence, an ESRP will not normally request location validation of an ECRF.

This optional attribute indicates whether location validation should be performed and is currently conditioned on the <location> element containing a civic address; i.e., it is an error to request location validation for a geodetic coordinates-based location in RFC5222. This may be changed in a future edition to allow validation of a geodetic location.

The validateLocation attribute is coded "true" to request location validation or is omitted or coded "false" to request no location validation. For i3 emergency call routing, this attribute normally will be omitted.

The attributes and elements of the <location> element given in Table 4-1 above are specified in Table 4-2 below along with a short description of their purpose. Note only the two-dimensional (2D) geoshapes—Point, Polygon, Circle, Ellipse, and Arcband, are supported for geodetic coordinates-based locations.

Table 4-2 – LoST <location> Element Attributes and Elements

Name	Condition	Purpose
Profile	Mandatory	This attribute defines the profile of the location information; i.e., the nature of the location information (civic or geodetic).
Id	Mandatory	This attribute defines an id uniquely identifying the <location> element within the <findService> message.
Xmlns	Conditional	This attribute specifies an XML namespace appropriate to the location profile.
<Point>	Conditional	This element defines a "point" geodetic shape-based location.
<Polygon>	Conditional	This element defines a "polygon" geodetic shape-based location.
<Circle>	Conditional	This element defines a "circle" geodetic shape-based location.
<Ellipse>	Conditional	This element defines an "ellipse" geodetic shape-based location.
<Arcband>	Conditional	This element defines an "arcband" geodetic shape-based location.
<civicAddress>	Conditional	This element defines a civic address-based location.

The LoST <location> element attributes and elements specified in above are described in greater detail below.

- profile Attribute

This required attribute specifies the nature of the location information contained within the <location> element and, therefore, how the information is encoded and should be interpreted.

This attribute is coded "civic" for a civic address-based location profile and "geodetic-2d" for a geodetic coordinates, shape-based location profile.

The "civic" and "geodetic-2d" profiles are baseline profiles defined by section 12 in [61]. In order to obtain maximum interoperability for emergency call routing, the ESRP and ECRF should use only "baseline" profiles for location information encoding.

- id Attribute

This required attribute uniquely identifies its <location> element within the <findService> message. If multiple <location> elements were to be present within the message, this attribute must have a unique value for each <location> element. However, i3 limits the query to only have one <location> element.

When the ECRF determines a route, it indicates which <location> element was successfully used to determine the route by copying the value of this attribute to the id attribute of the <locationUsed> element in the <findServiceResponse> message; thus permitting the requesting entity to identify the <location> element successfully used by the ECRF.

This attribute can be coded with any value. Since LoST permits only profiles based on a single baseline profile in a <findService> request and i3 permits only baseline profiles in the request, there will be only one <location> element, which makes this attribute somewhat superfluous. Notwithstanding, LoST requires it.

- xmlns Attribute

This attribute specifies an XML namespace that defines the markup language for the specified location profile. It will specify the PIDF-LO civic address XML namespace that defines the elements and their attributes for civic address-based location information, or the PIDF-LO geodetic shapes XML namespace that defines the elements (described below) and their attributes used for geodetic coordinates-based location information.

When the profile attribute is coded "civic", this attribute must be coded for the PIDF-LO civic address (see [76]) namespace. For example:

```
xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"
```

When the profile attribute is coded "geodetic-2d", this required attribute must be multiply-coded with the namespaces for generic GML shapes and specific PIDF-LO geodetic shapes (see sections 4 and 5 of [75] and [100]). The geoShapes namespace defines a subset of the GML namespace shapes in a manner appropriate to PIDF-LO, but does not redefine all shapes or attributes; hence the need to reference the GML namespace as well.

The example below shows XML namespace prefixes of "gml" and "gs". Since both namespaces define mutually named shapes, the appropriate geographic and geoshape element names would be qualified with the appropriate prefixes (e.g., <gml:Point> and <gml:pos>).

```
xmlns:gml="http://www.opengis.net/gml"
```

```
xmlns:gs="http://www.opengis.net/pidflo/1.0" "
```

Typically, the xmlns would not appear in the <location> element, but rather would appear in the location profile element (e.g., <civic address>). If an xmlns for a location profile is found in the <location> element, it must declare a prefix.

- <Point> Element

This conditional element specifies point shape-based, geodetic coordinates location information (e.g., <gml:Point>). Use of this element is described in section 12.2 of [61] and the element is described in section 5.2.1 of [75] and in [100]. <Point> is part of the <http://www.opengis.net/gml> namespace.

This attribute is conditioned on the profile attribute coded "geodetic-2d"; i.e., it is an error to specify this element when the profile attribute is not coded "geodetic-2d".

- <Polygon> Element

This conditional element specifies polygon shape-based, geodetic coordinates location information (e.g., <gml:Polygon>). Use of this element is described in section 12.2 of [61] and the element is described in section 5.2.2 of [75] and in [100]. <Polygon> is part of the <http://www.opengis.net/gml> namespace.

This attribute is conditioned on the profile attribute coded "geodetic-2d"; i.e., it is an error to specify this element when the profile attribute is not coded "geodetic-2d".

- <Circle> Element

This conditional element specifies circle shape-based, geodetic coordinates location information (e.g., <gs:Circle>). Use of this element is described in section 12.2 of [61] and the element is described in section 5.2.3 of [75] and in [100]. <Circle> is part of the <http://www.opengis.net/pidflo/1.0> namespace

This attribute is conditioned on the profile attribute coded "geodetic-2d"; i.e., it is an error to specify this element when the profile attribute is not coded "geodetic-2d".

- <Ellipse> Element

This conditional element specifies ellipse shape-based, geodetic coordinates location information (e.g., <gs:Ellipse>). Use of this element is described in section 12.2 of [61] and the element is described in section 5.2.4 of [75] and in [100]. <Ellipse> is part of the <http://www.opengis.net/pidflo/1.0> namespace

This attribute is conditioned on the profile attribute coded "geodetic-2d"; i.e., it is an error to specify this element when the profile attribute is not coded "geodetic-2d".

- <Arcband> Element

This conditional element specifies arcband shape-based, geodetic coordinates location information (e.g., <gs:Arcband>). Use of this element is described in section 12.2 of [61] and the element is described in section 5.2.5 of [75] and in [100]. <Arcband> is part of the <http://www.opengis.net/pidflo/1.0> namespace.

This attribute is conditioned on the profile attribute coded "geodetic-2d"; i.e., it is an error to specify this element when the profile attribute is not coded "geodetic-2d".

- <civicAddress> Element

This conditional element specifies civic address-based location information. Section 12.3 of [61] describes use of this element and [6] and [76] describe the element and its attributes.

<civicAddress> is part of the

urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr namespace.

Table 4-3 gives a short description of many child elements used to specify civic address information. Note that the LoST request does not include a PIDF-LO, but rather has some of the same elements as the PIDF-LO. The requestor copies those elements from the PIDF-LO to the LoST request.

This attribute is conditioned on the profile attribute coded "civic"; i.e., it is an error to specify this element when the profile attribute is not coded "civic".

Table 4-3 PIDF <civicAddress> Element Attributes and Elements

Name	Description	Example
<country>	2-letter ISO code	US
<A1>	national subdivision (e.g., state)	NY
<A2>	county, parish	King’s County
<A3>	city, township	New York
<A4>	city division, borough	Manhattan
<A5>	neighborhood	Morningside Heights
<A6> ¹²	street name (deprecated)	
<RD>	primary road name	Broadway
<PRD>	leading street direction	N
<POD>	trailing street suffix	SW
<STS>	street suffix	Ave

¹² RD must be used instead of A6. ESInet elements should accept A6 and treat as RD. If both are present and they are not the same value, it should be treated as an error.

Name	Description	Example
<HNO>	house number	123
<HNS>	house number suffix	A, 1/2
<LMK>	Landmark or vanity address	Columbia University
<LOC>	additional location info	South Wing
<NAM>	name (residence or office occupant)	Town Barber Shop
<PC>	postal or ZIP code	10027-0401
<BLD>	building (structure)	Low Library
<UNIT>	unit (apartment, suite)	Apt 42
<FLR>	floor	4
<ROOM>	room	450F
<PLC>	type of place	office
<PCN>	postal community name	Leonia
<ADDCODE>	additional code	132030000003
<SEAT>	Seat (desk, workstation, cubicle)	WS 181
<RDSEC>	road section	14
<RDBR>	branch road name	Lane 7
<RDSUBBR>	sub-branch road name	Alley 8
<PRM>	Road name pre-modifier	Old
<POM>	Road name post-modifier	Service

4.5.1.1.2 LoST <findServiceResponse> Message

When the i3 ECRF successfully processes a LoST <findService> message, it returns a LoST <findServiceResponse> message containing the "next hop" ESRP or final PSAP URI. If the ECRF cannot successfully process a LoST <findService> message, it returns a LoST <errors> message indicating the nature of the error (see section 4.5.1.1.3) or a LoST <redirect> message indicating the ECRF that can process the <findService> message (see section 4.5.1.1.4). Table 4-4 specifies the elements and attributes of the <findServiceResponse> message.

Table 4-4 – LoST <findServiceResponse> Message Attributes and Elements

Name	Condition	Purpose
xmlns	Mandatory	This attribute specifies the LoST protocol's XML namespace.
<path>	Mandatory	This element indicates the ECRF(s) (LoST servers) that handled the request.
<locationUsed>	Optional	This element identifies the location used by the ECRF to determine the service URI.
<mapping>	Mandatory	This element identifies a service region and its associated service URIs.

The elements and attributes that make up the <findServiceResponse> message are described below:

- **xmlns Attribute**

This required attribute specifies the LoST protocol XML namespace and should be coded as specified by section 17.4 in [61] (shown below).

```
xmlns="urn:ietf:params:xml:ns:lost1"
```

- **path**

This element contains <via> elements indicating the ECRF(s) that handled the <findService> request. See section 6 in [61] for detailed information about the <path> element.

The order of <via> elements within the "path" element is significant. The first <via> element always indicates the ECRF (LoST server) that received the initial <findService> message query from the requesting entity.

For a recursive query, the last <via> element indicates the authoritative ECRF and any intervening <via> elements between the first and last <via> elements indicate the path from the initiating ECRF to the authoritative ECRF.

For an iterative query, there are <via> elements indicating the ECRFs that were contacted during processing of the <findService> request.

- **locationUsed**

This optional element identifies the <location> element within the <findService> message used to successfully determine the service URI.

The value of this element is a copy of the value from the id attribute of the <location> element successfully processed by the ECRF.

- **mapping**

This required element returns the service information to the requesting entity when the ECRF successfully processed the <findService> message.

The "mapping" element contains many elements and attributes described in Table 4-5

Table 4-5 LoST <mapping> Element Attributes and Elements

Element/Attribute	Condition	Purpose
source	Mandatory	Identifies the authoritative generator of the mapping
sourceId	Mandatory	Identifies a particular mapping
lastUpdated	Mandatory	Describes when a mapping identified by the source and sourceId was last updated
expires	Mandatory	Identifies the absolute time when the mapping becomes invalid
displayName	Optional	Describes a human readable display name, e.g., the name of the PSAP serving the location (may be repeated)
service	Mandatory	Identifies the service for which the mapping applies
serviceBoundary	Optional	Identifies the area where the URI returned would be valid
serviceBoundaryReference	Optional	Identifies the reference that can be used to access the service boundary for which the URI returned is valid
serviceNumber	Optional	Provides the emergency services dial string that is appropriate for the location provided in the query
uri	Conditional ¹³	Contains the appropriate contact URI for the requested service. May be repeated when multiple protocols are accepted at the destination. Not intended to support multiple destinations.
locationValidation	Optional	Indicates which elements of the civic location were “valid” and used for mapping, which elements were “invalid” and which elements were “unchecked”

The attributes and elements that make up the LoST "mapping" element specified in Table 4-5 above are described below:

¹³ The ECRF includes one or more URIs in a <findServiceResponse> message if one can be determined. Absence of a URI indicates a mapping exists, but no URI is provided in that mapping. This should not occur.

- **source Attribute**

This element identifies the authoritative generator of the mapping (the LoST server that generated the mapping). LoST servers are identified by U-NAPTR/DDDS application unique strings, in the form of DNS name (e.g., lostserver.notreal.com).
- **sourceId Attribute**

This element identifies a particular mapping at the LoST server and is unique among all the mappings maintained by the LoST server.
- **lastUpdated Attribute**

This element describes the date and time when this specific instance of mapping was updated. The date and time is represented in UTC format.
- **expires Attribute**

This element describes the date and time when a particular mapping becomes obsolete. The date and time are described using a timezoned XML type datetime. This element may optionally contain the values of “NO-CACHE” indicating that the mapping should not be cached and “NO-EXPIRATION” indicating that the mapping has no expiration instead of the date and time.
- **<displayName> Element**

The display name is a text string that provides an indication of the serving agency(ies) for the location provided in the query. This information might be useful to PSAPs that query an ECRF. This capability could be used to provide English Language Translation (ELT)-type information that PSAPs receive from ALI databases today.
- **<service>**

The <service> element identifies the service for which this mapping is valid. The ECRF is required to support the "sos" service. Support for other services will depend on local implementation.
- **<serviceBoundary>**

The <serviceBoundary> element identifies the geographical area where the returned mapping is valid. The intent of this parameter is to allow a mobile endpoint to realize that it is moved out of the area where a stored mapping is valid and trigger it to query for a new valid mapping. This element may be supported by the ECRF depending on local implementation.
- **<serviceBoundaryReference>**

The <serviceBoundaryReference> element identifies a reference that could be used to access the service boundary for the requested mapping. This parameter may be supported by the ECRF depending on local implementation.

- <serviceNumber>
The <serviceNumber> element contains the emergency services number appropriate for the location provided in the query. This allows a foreign end device to recognize a dialed emergency number.
- Uniform Resource Identifier (<uri>)
The URI specifies either the address of the PSAP or the ESRP that is appropriate for the location sent in the query message. The decision of whether to send the PSAP URI or the ESRP URI is based on
 - a) whether the query is made by the end user, VSP Routing Proxy, i3 PSAP, or the ESRP (which would be determined by the credentials presented in the establishment of a TLS connection to the ECRF) and/or
 - b) the service urn presented in the query.
- <locationValidation>
The <locationValidation> element identifies which elements of the received civic address were “valid” and used for mapping, which were “invalid” and which were unchecked. Since the ECRF is not responsible for performing validation, this parameter may not be returned, subject to local implementations.

4.5.1.1.3 LoST <errors> Message

If the ECRF cannot successfully process a <findService> message, it returns the <errors> message instead of the <findServiceResponse> message. The <errors> message contains information indicating the nature and source of the error.

Table 4-6 – LoST <errors> Message Attributes and Elements

Name	Condition	Purpose
xmlns	Mandatory	This attribute specifies the LoST protocol's XML namespace.
source	Mandatory	This attribute specifies the source of the error.

Name	Condition	Purpose
<badRequest> <forbidden> <internalError> <locationProfileUnrecognized> <locationInvalid> <SRSInvalid> <loop> <notFound> <serverError> <serverTimeout> <serviceNotImplemented>	Mandatory	These elements specify error types.

The LoST <errors> message attributes and elements specified in Table 4-6 are described in greater detail below.

- xmlns Attribute

This required attribute must specify the LoST protocol XML namespace and is coded as follows.

```
xmlns="urn:ietf:params:xml:ns:lost1"
```

- source Attribute

This required attribute identifies the source of the error, which will be in the form of a DNS name (e.g., ecrf.example.com).

The following LoST <errors> message child elements describe the types of errors encountered or detected by the ECRF. They give the requesting entity a limited set of "error types", each of which is likely to be handled in a particular manner by the requesting entity regardless of the nature of the actual error (see message attribute below). One or more "error type" elements can be returned in the <errors> message. See section 13.1 of [61] for an explanation of each error type.

- <badRequest> Element

This element indicates the ECRF could not parse or otherwise understand the request sent by the requesting entity (e.g., the XML is malformed).

- <forbidden> Element

This element indicates an ECRF refused to send an answer. This generally only occurs for recursive queries, namely, if the client tried to contact the authoritative server and was refused.

- <internalError> Element
This element indicates the ECRF could not satisfy a request due to a bad configuration or some other operational and non-LoST protocol-related reason.
- <locationProfileUnrecognized> Element
This element indicates the ECRF did not recognize the value of the profile attribute sent with the <findService> request; i.e., it was not coded with "civic" or "geodetic-2d".
- <locationInvalid> Element
This element indicates the ECRF determined the geodetic or civic location is invalid (e.g., geodetic latitude or longitude value is outside the acceptable range).
- <SRSInvalid> Element
This element indicates the ECRF does not recognize the spatial reference system (SRS) specified in the <location> element or it does not match the SRS specified in the profile attribute (e.g., profile="geodetic-2d" and <civicAddress> element present).
- <loop> Element
This element indicates an ECRF detected a loop during a recursive query; i.e., an ECRF finds the "next hop" URL is already in a <via> element within the <path> element of the <findService> request.
- <notFound> Element
This element indicates the ECRF could not find an answer to the query.
- <serverError> Element
This element indicates the ECRF received a response from another ECRF for a recursive query but could not parse or understand the response.
- <serverTimeout> Element
This element indicates the ECRF timed out waiting for a response (e.g., another ECRF for a recursive query, the SIF server, etc.).
- <serviceNotImplemented> Element
This element indicates the ECRF detected the requested service URN is not implemented and it found no substitute for it.

Each of the preceding "error type" elements can have the following attributes.

Table 4-7 – LoST "Error Type" Element Attributes

Name	Condition	Purpose
message	Optional	This attribute specifies additional human-readable information about an error.

Name	Condition	Purpose
xml:lang	Conditional	This attribute specifies the language in which the message attribute's value is written.

The LoST <errors> message "error type" element's attributes specified in Table 4-7 are described in greater detail below.

- message Attribute
This optional attribute specifies human-readable text indicating a more particular or specific reason for the error (e.g., message="LoST server encountered a software bug.").
- xml:lang Attribute
This conditional attribute specifies the language in which the message text is written (e.g., xml:lang="en" indicates English). This attribute is conditioned on the message attribute; i.e., this attribute should not be present if the message attribute is not present. Further, if the message attribute is present, this attribute should be present so the text of a message can be properly displayed, logged and/or interpreted.

4.5.1.1.4 LoST <redirect> Message

If the ECRF cannot or should not handle a <findService> message for any reason (e.g., failover, etc.) but does know the ECRF that can, it returns the <redirect> message to the requesting entity instead of the <findServiceResponse> or <errors> message. This message returns information indicating the source of and reason for the redirection and the URL of the ECRF to which the requesting entity should redirect its <findService> message.

Table 4-8 – LoST <redirect> Message Attributes and Elements

Name	Condition	Purpose
xmlns	Mandatory	This attribute specifies the LoST protocol's XML namespace.
target	Mandatory	This attribute specifies the target of the redirection.
source	Mandatory	This attribute specifies the source of the redirection.
message	Optional	This attribute specifies additional human-readable information about the redirection.
xml:lang	Conditional	This attribute specifies the language in which the message attribute's value is written.

The LoST <redirect> message attributes and elements specified in Table 4-8 are described in greater detail below.

- **xmlns Attribute**

This required attribute must specify the LoST protocol XML namespace and is coded as follows.

```
xmlns="urn:ietf:params:xml:ns:lost1"
```

- **target Attribute**

This required attribute identifies the target of the redirection, i.e., the domain name of the ECRF to which the requesting entity should send its <findService> message.

- **source Attribute**

This required attribute identifies the source of the redirection, which will be in the form of a DNS name (e.g., ecrf.example.com).

- **message Attribute**

This optional attribute specifies human-readable text indicating a more particular or specific reason for the redirection (e.g., message="LoST server has temporarily failed over to another system.").

- **xml:lang Attribute**

This conditional attribute specifies the language in which the message text is written (e.g., xml:lang="en" indicates English). This attribute is conditioned on the message attribute; i.e., this attribute should not be present if the message attribute is not present. Further, if the message attribute is present, this attribute should be present so the text of a message can be properly displayed, logged and/or interpreted.

4.5.1.1.5 LoST Common XML Namespaces Summary

All LoST messages have root and other elements that require specification of XML namespaces for their proper interpretation. Table 4-9 shows LoST elements that require specification of the xmlns attribute to define their appropriate XML namespace. Some elements may require more than one xmlns attribute since their sub-elements contain elements defined by more than one namespace.

Table 4-9 – LoST Protocol Message Elements and xmlns Attribute Common Namespaces

Name	xmlns Attribute Value	Defines
<findService> <findServiceResponse> <errors> <redirect>	urn:ietf:params:xml:ns:lost1	LoST protocol elements

Name	xmlns Attribute Value	Defines
<location>	urn:ietf:params:xml:ns:lost1	LoST protocol elements
	urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr	Civic address elements
	http://www.opengis.net/pidflo/1.0	Geoshape elements
	http://www.opengis.net/gml	GML elements
<serviceBoundary>	urn:ietf:params:xml:ns:lost1	LoST protocol elements
	urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr	Civic address elements
	http://www.opengis.net/pidflo/1.0	Geoshape elements
	http://www.opengis.net/gml	GML elements

4.5.1.1.6 LoST srsName Attribute Common URNs Summary

GML and geoshape elements require an srsName attribute to specify a URN that defines their interpretation. Table 4-10 shows GML and geoShape elements that require specification of the srsName attribute and their possible URN value(s). Some elements may require more than one srsName attribute since their child elements contain elements defined by more than one URN.

Table 4-10 - GML and geoShape Elements and srsName Attribute Common URNs

Name	srsName Attribute Value	Defines
<gs:Point> <gs:Polygon> <gs:Circle> <gs:Ellipse> <gs:Arcband>	urn:ogc:def:crs:EPSG::4326	Two-dimensional (2D) shapes
<gs:height>	urn:ogc:def:uom:EPSG::9001	Distance Unit of Measure in meters
	urn:ogc:def:uom:EPSG::9101	Angular Unit of Measure in radians
	urn:ogc:def:uom:EPSG::9102	Angular Unit of Measure in degrees

Name	srsName Attribute Value	Defines
<gml:pos>		Latitude and Longitude in decimal degrees

4.5.1.2 Call Routing Scenarios

The following examples are preliminary. Further examples will be provided in a future edition of this document

4.5.1.2.1 Civic Address-based Call Routing LoST Interface Example Scenario

```
<?xml version="1.0" encoding="UTF-8"?>
<findService xmlns="urn:ietf:params:xml:ns:lost1"
  recursive="true" serviceBoundary="value">
  <location id="627b8bf819d0bcd4d" profile="civic">
    <civicAddress
      xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
      <country>US</country>
      <A1>OH</A1>
      <A3>Columbus</A3>
      <RD>Airport</RD>
      <STS>DR</STS>
      <HNO>2901</HNO>
      <NAM>Courtyard Marriott</NAM>
      <RM>Board Room B</RM>
      <PC>43219</PC>
    </civicAddress>
  </location>
  <service>urn:service:sos</service>
</findService>
```

A <findService> well-formed civic address query

```
<?xml version="1.0" encoding="UTF-8"?>
  <findServiceResponse xmlns="urn:ietf:params:xml:ns:lost1">
```

```
<mapping
  expires="2010-01-01T01:44:33Z"
  lastUpdated="2009-09-01T01:00:00Z"
  source="esrp.state.oh.us.example"
  sourceId="e8b05a41d8d1415b80f2cdbb96ccf109">
  <displayName xml:lang="en">
    Columbus PSAP
  </displayName>
  <service>urn:service:sos</service>
  <serviceBoundary
    profile="civic">
    <civicAddress
      xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
      <country>US</country>
      <A1>OH</A1>
      <A3>Columbus</A3>
    </civicAddress>
    </serviceBoundary>
    <uri>sip:columbus.psap@state.oh.us</uri>
    <serviceNumber>911</serviceNumber>
  </mapping>
  <path>
    <via source="ecrf.state.oh.us"/>
    <locationUsed id="627b8bf819d0bcd4d"/>
  </findServiceResponse>
```

A <findServiceResponse> Response to Well-formed query

```
<?xml version="1.0" encoding="UTF-8"?>
  <findService xmlns="urn:ietf:params:xml:ns:lost1"
    recursive="true" serviceBoundary="value">
    <location id="627b8bf819d0bcd4d" profile="civic">
      <civicAddress
        xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
```

```
<country>US</country>
<A3>Columbus</A3>
<RD>Airport</RD>
<STS>DR</STS>
<HNO>2901</HNO>
</civicAddress>
</location>
<service>urn:service:sos</service>
</findService>
```

A <findService> civic address query with partial info

```
<?xml version="1.0" encoding="UTF-8"?>
<errors xmlns="urn:ietf:params:xml:ns:lost1"
  source="ecrf.state.oh.us">
  <internalError message="notFound" xml:lang="en"/>
</errors>
```

A <error> Response to partial-formed query

This response scenario indicates an error that the server cannot find an answer to the query.

4.5.1.2.2 Geodetic Coordinates-based Call Routing LoST Interface Scenario

To be provided in a future edition of this document

4.5.2 Location Validation

“Validating” a location in NG9-1-1 means querying the Location Validation Function (Section 5.4) to determine if the location is suitable for use (specifically, if the location can be used to accurately route the call and dispatch responders). The LVF uses the same LoST interface as routing as defined above, with the validateLocation option in the <findservice> request set to true.

4.6 Event Notification

Events are communicated within and between ESInets using the SIP Subscribe/Notify mechanism RFC3265 [17]. ESInet functional elements may need to accept or generate events to outside elements using different asynchronous event notification mechanisms, which would need to be interworked to SIP Subscribe/Notify at the ESInet boundary.

NG9-1-1 events are defined by an event package which includes the name of the event, the subscription parameters, the conditions under which NOTIFYs are issued and the content of the NOTIFY, as described in RFC 3265.

4.7 Spatial Information Function Layer Replication

A SIF layer replication interface is used within the ESInet to maintain copies of layers of the master SIF that drive routing and display of maps throughout the system. A “master” SIF maintains the authoritative copy of the data. One or more copies of that data can be maintained on other services using the layer replication protocol. A change to the master SIF will be reflected in the copies nearly immediately.

The SIF layer replication is built on the OGC Web Feature Service.

4.7.1 Web Feature Service

The SIF must implement an OGC Web Feature Service (WFS) OGC04-094 [130]. As a practical matter all systems using the layer replication service must implement both a client and a server WFS.

<conformanceClass> must be “modify”

<interfaceProtocol> must be “SOAP” and may also include “REST”

<dataLanguage> must be “GML”

<schemaLanguage> must be “XML Schema”

The Data Maintenance extension (Lock, Insert, Update, Delete) must be implemented.

Note: A standard NENA schema for the WFS will be provided in a future edition of this document.

4.7.2 Atom Protocol and GeoRSS

OGC Document OGC 08-001 [131] describes loosely-coupled synchronization of geospatial databases using WFS and the Atom protocol (RFC4287 [132] and RFC5023 [133]). Essentially, the changes in the database are expressed in WFS Insert/Update/Delete actions and ATOM is used to move the edits from the master to the copy. GeoRSS (<http://www.georss.org>) is a very simple mechanism used to encode the GML in RSS feeds for use with ATOM. OGC 08-001 describes two formats for the edits: GeoRSS Simple and GML. NG9-1-1 uses GML. The “Feedback Feed” service defined in Chapter 7 is not used.

Note: OGC 08-001 is not a standard. It is a description of a pilot program. Nevertheless, the content of the document is believed to be sufficient to describe how to build interoperable implementations of the layer sync protocol. A future OGC specification or a future edition of this document will describe the protocol definitively.

4.8 CAD

To be defined in a future edition of this standard.

4.9 Discrepancy Reporting

Any time there is a database, errors or discrepancies may occur in the data. There must be a discrepancy report (DR) function to notify agencies and services (including BCF, ESRP, ECRF, Policy Store and LVF) when any discrepancy is found. The discrepancy reporting audience is anyone who is using the data and finds a problem. Some of the places discrepancies could occur include:

- The LIS needs to file a Discrepancy Report on the LVF
- The ECRF/LVF may be receiving data from another ECRF/LVF and thus will file a DR on its upstream provider
- The ECRF/LVF needs to file a DR on the GIS
- The ESRP needs to file a DR on the owner of a routing policy (PSAP, ESRP) that has a problem
- The PSAP needs to file a DR on an ESRP if a call is misrouted
- The PSAP needs to file a DR on the GIS when issues found in a map display
- Any client of an ECRF needs to file a DR on the routing data (which could be a GIS layer problem or something else)
- A PSAP or ESRP needs to file a DR on a LIS or a Service Database Provider
- A PSAP or ESRP needs to file a DR on a CIDB, or AdditionalLocationData building owner/tenant
- A BCF, ESRP or PSAP needs to file a DR on a originating network sending it a malformed call
- Any client may need to file a DR on the ESInet operator
- One PSAP needs to file a DR on another PSAP that transferred a call to it
- A data user may need to file a DR on a data owner due to rights management issues.
- A log client (logging entry or query) may need to file a DR on the log service
- Any entity may have to file a DR on another entity due to authentication issues (bad certificate, unknown entity, ...)
- An ESRP or PSAP may need to file a DR on a Border Control Function
- Any Policy Enforcement Point may need to file a DR on a Policy owner due to formatting, syntax or other errors in the policy

Next Generation 9-1-1 provides a standardized Discrepancy Reporting mechanism in the form of a web service. Each database or service agency must provide a Discrepancy Reporting web service.

A Discrepancy Report (DR) is sent by the agency reporting the discrepancy to a responding agency and will pass through several phases:

- The reporting agency creates the DR and forwards it to the responding agency
- The responding agency acknowledges the DR report and provides and estimates when it will be resolved
- The reporting agency may request a status update and receive a response
- The responding agency resolves the DR and reports its resolution to the reporting agency

All DRs must contain common data elements (a prolog) that includes:

- Time Stamp of Discrepancy Submittal
- Discrepancy Report ID
- Discrepancy reporting agency domain name
- Discrepancy reporting agent user ID
- Discrepancy reporting contact info
- Service or Instance in which the discrepancy exists
- Additional notes/comments
- Reporting Agency’s assessment of severity
- Discrepancy Service or Database specifics*

For each type of Discrepancy Report there is a specific database or service where the discrepancy originated or occurred. Within the database or service there is a defined block of data specific to the database or service that will be included in the DR and must include:

- Query that generated the discrepancy
- Full response of the query that generated the discrepancy (Message ID, Result Code, etc.)
- What the reporting agency thinks is wrong
- What the reporting agency thinks is the correct response, if available

4.9.1 Discrepancy Report

The Discrepancy Reporting web service is used by a reporting agency to initiate a Discrepancy Report and includes the following functions:

DiscrepancyReportRequest

Parameter	Condition	Description
TimeStamp	Mandatory	Timestamp of Discrepancy Report Submittal
ReportId	Mandatory	Unique (to reporting agency) ID of report
ReportingAgency	Mandatory	Domain name of agency creating the report
ReportingAgent	Optional	UserId of agent creating the report
ReportingContact	Mandatory	vCard of contact about this report
Service ¹	Conditional	Name of service or instance

		where discrepancy exist
Severity	Mandatory	Enumeration of reporting agency’s opinion of discrepancy’s severity
Comment	Optional	Text comment
Discrepancy ²	Mandatory	Database/Service-specific block

¹ Each database/service description denotes whether the “Service” parameter is required for that database/service or not, and provides an XML description of the “Discrepancy” parameter content

² In cases of routing discrepancies the PIDF-Lo would be included

The response to the Discrepancy Report includes the following;

DiscrepancyReportResponse

Parameter	Condition	Description
RespondingAgency	Mandatory	Domain name of agency responding to the report
RespondingAgent	Optional	UserId of agent responding to the report
RespondingContact	Mandatory	vCard of contact about this report
EstimatedResponseTimeStamp	Mandatory	Estimated date/time when response will be returned to reporting agency
Comment	Optional	Text comment
errorCode	Optional	Error Code

Error Codes

- 100 Okay No error
- 520 Unknown Service/Database (“not ours”)
- 521 Unauthorized Reporter
- 504 Unspecified Error

4.9.2 Status Update

A reporting agency may request a status update, the update report includes:

StatusUpdateRequest

Parameter	Condition	Description
ReportId	Mandatory	Unique (to reporting agency) ID of report
ReportingAgency	Mandatory	Domain name of agency creating the report
ReportingAgent	Optional	UserId of agent creating the report
ReportingContact	Mandatory	vCard of contact about this report
Comment	Optional	Text Comment

The status report update includes:

StatusUpdateResponse

Parameter	Condition	Description
RespondingAgency	Mandatory	Domain name of agency responding to the report
RespondingAgent	Optional	UserId of agent responding to the report
RespondingContact	Mandatory	vCard of contact about this report
EstimatedResponseTimeStamp	Mandatory	Estimated date/time when response will be returned to reporting agency
Comment	Optional	Text Comment
errorCode	Optional	Error Code

Error Codes

100 Okay No error

- 522 Unknown ReportId
- 521 Unauthorized Reporter
- 504 Unspecified Error

4.9.3 Discrepancy Resolution

The reporting agency can query for resolution to any of its outstanding reports. If any responses are available, they will be returned. A query key is passed in the request, and an updated one is returned in the response. The returned query key is used in a subsequent request.

DiscrepancyResolutionRequest is defined as:

Parameter	Condition	Description
QueryKey	Mandatory	Key value returned on previous response
ReportingAgency	Mandatory	Domain name of agency creating the report

DiscrepancyResolutionResponse is defined as:

Parameter	Condition	Description
QueryKey	Mandatory	Key value to be used on next request
ResolutionReport	Conditional	Resolution Report, if available. May be repeated
errorCode	Optional	Error Code

Error Codes

- 100 Okay No error
- 524 Bad Query Key
- 504 Unspecified Error

ResolutionReport is defined as:

Element	Type	Description
---------	------	-------------

ReportId	AN	
ReportingAgency	AgencyId	Domain name of agency creating the report
ReportingAgent	AgentId	UserId of agent creating the report
Service	Conditional	Name of service or instance
RespondingAgency	Mandatory	Domain name of agency responding to the report
RespondingAgent	Optional	UserId of agent responding to the report
RespondingContact	Mandatory	vCard of contact about this report
Timestamp	Timestamp	Date and Time of response
Comment	Optional	Text Comment
Response	Extension	Database/Service-specific response data

The following elements must be included, with the prolog, depending on the service.

4.9.4 LVF Discrepancy Report

A client of an LVF may report a discrepancy. The most common report is that the LVF claims the location sent in the PIDF is invalid, when the client believes it is valid.

LVFDiscrepancyReport is defined as:

Element	Type	Description
Location	PIDF	Location queried
Service	URN	Service URN queried
LocationValidation	LocationValidation (from RFC5222)	Validation Response
Discrepancy	Enumeration	BelievedValid,OtherReport

LVFDiscrepancyResponse is defined as:

Element	Type	Description
ValidationResponse	Enumeration	EntryAdded, NoSuchLocation, OtherResponse

4.9.5 Policy Discrepancy Report

A client of a Policy may report a discrepancy. The most common report is that the Policy Query returns an invalid Policy from the Policy Store.

PolicyDiscrepancyReport is defined as:

Element	Type	Description
policyName	Mandatory	The name of the policy
Agency	Mandatory	The agency whose policy is requested. Must be a domain name or URI that contains a domain name
RetreivePolicyResponse	Mandatory	The Response received from the Policy Retrieve Request as shown in 4.4.1

The PolicyDiscrepancyResponse is defined as:

Element	Type	Description
ValidationResponse	Enumeration	Policy Added, Policy Updated, No Such Policy, Other Response

4.9.6 LoST Discrepancy Report

4.9.7 ECRF Discrepancy Report

4.9.8 BCF Discrepancy Report

4.9.9 Log Discrepancy Report

4.9.10 PSAP Call Taker Discrepancy Report

4.9.11 Permissions Discrepancy Report

4.9.12 GIS Discrepancy Report

5 Functions

5.1 Border Control Function (BCF)

A BCF sits between external networks and the ESInet and between the ESInet and agency networks. All traffic from external networks transits a BCF.

5.1.1 Functional Description

The Border Control Function comprises several distinct elements pertaining to network edge control and SIP message handling. These include:

- Border Firewall
- Session Border Control

It is imperative that the border control function support the following security related techniques:

- Prevention
- Detection
- Reaction

Additionally, the entirety of the functional element may include aspects of the following:

- B2BUA
- Media anchoring
- Stateful Firewall

Border Firewall — This functional component of the BCF inspects ingress and egress traffic running through it. It is a dedicated appliance or software running on a computer. There are a variety of different roles a firewall can take however, the typical roles are application layer and network layer firewalls:

- 1) **Application layer** – these scan and eliminate known malware attacks from extranet and intranet sources at layer 7 before they ever reach a user’s workstation or a production server or another end point located inside the ESInet. These act as the primary layer of defense for most Internet originated malware attacks that are protocol specific.
- 2) **Network layer** — these manage access on the Internet perimeter and between network segments. Typically they do not provide active scanning at the application layer and provide access control through the use of access control lists and port based permission/denial management (UDP, TCP etc.). They also mitigate attacks on lower layer protocol layers (e.g., SYN Flooding).

Firewalls deployed on the ESInet shall meet the following specifications:

- 1) Provide both application and network layer protection and scanning.
- 2) Denial of service (DoS) detection and protection
 - a. Detection of unusual incoming IP packets that may then be blocked to protect the intended receiving user or network.
 - b. To prevent distributed denial of service (DDoS) attack, destination specific monitoring, regardless of the source address, may be necessary.
- 3) Provide a mechanism such that malware definitions and patterns can be easily and quickly updated by a national 9-1-1 CERT or other managing authority
- 4) Capability to receive and update 9-1-1 Malicious Content (NMC) filtering automatically for use by federated firewalls in protecting multiple disparate ESInets
- 5) Adhere to the default deny principle.

Please refer to NENA 04-503 [102] for more information on firewall requirements.

Session Border Control — The session border controller functional element of the BCF plays a role in VoIP services by controlling borders to resolve multiple VoIP-related problems such as Network Address Translation (NAT) or firewall traversal. Session Border Controllers (SBCs) are already being extensively used in existing VoIP service networks.

The following primary functions are related to the SBC within a BCF:

- Identification of emergency call/session and priority handling for the IP flows of emergency call/session traffic. Use of the BCF, or any other ESInet element for non-emergency calls that enter an ESInet is not described herein except for calls to an administrative number in

the PSAP. Such non-emergency calls are beyond the scope of this document.

- Conformance checking and mapping (if applicable) of priority marking based on policy for emergency calls/sessions
- Facilitate forwarding of an emergency call/session to an ESRP (and only an ESRP)
- Protection against DDoS attacks: The SBC component of the BCF shall protect against VoIP specific and general DDoS attacks on VoIP network elements.
- SIP Protocol Normalization: The SBC component of the BCF shall support SIP/SDP protocol normalization and/or repair, including adjustments of encodings to a core network profile. This may be done in order to facilitate backward compatibility with older devices that may support a deprecated version of SIP/SDP.
- NAT and NAPT Traversal: The SBC component of the BCF shall perform NAT traversal for authorized calls/sessions using SIP protocol. The SBC component must be able to recognize that a NAT or NAPT has been performed on Layer 3 but not above and correct the signaling messages for SIP.
- IPv4/IPv6 Interworking: The SBC component of the BCF shall enable interworking between networks utilizing IPv4 and networks using IPv6 through the use of dual stacks, selectable for each BCF interface. All valid IPv4 addresses and parameters shall be translated to/from the equivalent IPv6 values.
- Signaling Transport Protocol Support: The SBC component of the BCF shall support SIP over the following protocols: TCP, UDP, TLS-over-TCP, and SCTP. Protocols supported must be selectable for each BCF interface to external systems. These transport layer protocols are generated and terminated at each interface to external systems (i.e., there is no "pass-thru" of transport layer information).
- VPN Bridging or Mediation: The SBC component of the BCF shall support terminating the IP signaling received from a foreign carrier onto the ESInet address space. The SBC component of the BCF shall support Back to Back User Agent functions to enable VPN bridging if needed.
- QoS/Priority Packet Markings: The SBC component of the BCF shall be capable of populating the layer 2 and layer 3 headers/fields, based on call/session type (e.g., 9-1-1 calls) in order to facilitate priority routing of the packets.

- Call Detail Recording - The SBC component of the BCF shall be capable of producing CDRs based on call/session control information (e.g., SIP/SDP). These CDRs can be used to manage the network and for SLA auditing.
- Transcoding: The SBC component of the BCF shall optionally support transcoding. For example, the SBC component may transcode baudot tones to RFC4103 real time text. See Section 4.1.8.3.
- Encryption: The SBC component of the BCF shall support encryption (AES on TLS) for calls that are not protected entering the ESInet.

Additionally, the SBC component of the BCF performs the following functions:

Opening and closing of a pinhole (firewall)

- Triggered by signaling packets, a target IP flow is identified by "5-tuples" (i.e., source/destination IP addresses, source/destination port number and protocol identifier) and the corresponding pinhole is opened to pass through the IP flow.

Resource and admission control

- For links directly connected to the element, and optionally networks behind the element, resource availability is managed and admission control is performed for the target call/session.

IP payload processing

- Transcoding (e.g., between G.711 and G.729) and DTMF interworking.

Performance measurement

- Quality monitoring for the target IP flow in terms of determined performance parameters, such as delay, jitter and packet loss. Performance results may need to be collected for aggregated IP flows.

Media encryption and decryption

- Encryption and decryption of media streamed (e.g., IPsec).

B2BUA for UAs that do not support Replaces

- The SBC component may include a B2BUA function for 9-1-1 calls where the caller does not indicate support for the Replaces operation. See section 5.8.1.

Typically, the firewall passes traffic for inbound SIP protocol to the Session Border Controller, which acts as an Application Layer Gateway for SIP. Primary non SIP protection is accomplished by the Firewall functions of the BCF. Primary SIP protection is accomplished by the SBC component of the BCF.

5.1.2 Interface Description

The BCF supports SIP interfaces upstream and downstream per Section 4.1. BCFs must support ROHC [145]. The BCF shall support an automated interface that allows a downstream element to mark a particular source of a call as a “bad actor” (usually due to receipt of a call that appears to be part of a deliberate attack on the system) and send a message to the BCF notifying it of this marking. To facilitate this notification, the BCF shall include a “NENA-source” parameter in the Via header that it inserts in the outgoing INVITE message associated with every call. Because the SBC component of the BCF may rewrite addresses, calls must be marked by the SBC component in a way that allows the recipient to identify the BCF that processed the call. The NENA-source parameter is formatted as follows: <unique source-id>@<domain name of BCF> (e.g., a7123gc42@sbc22.example.net).

When the downstream element identifies a source as a “bad actor”, it signals the BCF which source is misbehaving by sending it a BadActorRequest that contains the sourceId from the NENA-source parameter that was included in the Via header of the incoming INVITE message. The BCF responds by returning a BadActorResponse message which indicates whether or not an error was detected in the BadActorRequest message.

Upon receiving the BadActorRequest, the SBC component of the BCF should filter out subsequent calls from that source until the attack subsides.

The bad actor request/response is a webservice operated on the domain mentioned in the parameter.

The bad actor report is a webservice operated on the domain mentioned in the parameter.

BadActorRequest

Parameter	Condition	Description
sourceId	Mandatory	sourceId from a NENA-source parameter

BadActorResponse

Parameter	Condition	Description
errorCode	Mandatory	Error Code

Error Codes

- 100 Okay No error
- 101 Already reported
- 512 No such sourceId
- 513 Unauthorized
- 504 Unspecified Error

5.1.2.1 CallSuspicion

The BCF may be able to identify calls that may be part of a deliberate attack on the system. However, under normal conditions, we allow suspicious calls in, preferring to have a bad call show up to having a good call dropped. The behavior of downstream elements (ESRPs for example) may be affected by the determination of the BCF. For this purpose, the BCF attaches a parameter to the VIA it inserts on the call. The parameter: NENA-CallSuspicion is an enumeration having the following values:

- Legit: Call appears to be legitimate
- Suspicious: Call may fit a known attack, but the BCF is unsure
- Bad: Call fits a known attack pattern and is considered fraudulent.

5.1.3 Roles and Responsibilities

The ESInet operator is responsible for the BCF at the edge of the ESInet. PSAP or other agency is responsible for a BCF between its network and the ESInet.

5.1.4 Operational Considerations

In order to withstand the kinds of attacks anticipated, BCFs at the edge of the ESInet should be provisioned with capacity, both aggregate uplink bandwidth and BCF processing capacity larger than the largest feasible DDoS attack. As of this edition, that capacity is approximately 6-8 Gigabits of mitigation.

Creation of a Public Safety Computer Emergency Response Team (CERT) is anticipated, and all BCF operators must arrange to receive alerts from the CERT and respond. It is essential that all BCF support organizations have trained staff available 24 x 7 x 365 to immediately respond to attacks and have the capability and training to be able to adjust the BCF to mitigate such attacks.

5.2 Emergency Service Routing Proxy (ESRP)

5.2.1 Functional Description

5.2.1.1 Overview

The Emergency Service Routing Proxy (ESRP) is the base routing function for emergency calls for i3. As described in NENA 08-002, ESRPs are used in several positions within the ESInet:

- The "Originating ESRP" is the first routing element inside the ESInet. It receives calls from the BCF at the edge of the ESInet
- One or more "Intermediate ESRPs" which exist at various hierarchical levels in the ESInet. For example, the Originating ESRP may be a state-level function, and an intermediate ESRP may be operated by a county agency.
- The "Terminating ESRP" is typically at the edge of a PSAP, just past the PSAP BCF.

The function of the ESRP is to route a call to the next hop. The Originating ESRP routes to the appropriate intermediate ESRPs (if they exist), intermediate ESRPs route to the next level intermediate ESRP or to the Terminating ESRP, i.e., the appropriate PSAP. The Terminating ESRP routes to a call taker or set of call takers.

ESRPs typically receive calls from upstream routing proxies. For the originating ESRP, this is typically a carrier routing proxy. For an intermediate or terminating ESRP, this is the upstream ESRP. The destination of the call on the output of the ESRP is conceptually a queue, represented by a URI. In most cases, the queue is maintained on a downstream ESRP, and is most often empty. However, when the network gets busy for any reason, it is possible for more than one downstream element to "pull" calls from the queue. The queue is most often First In First Out, but in some cases there can be out-of-order selections from the queue.

The primary input to an ESRP is a SIP message. The output is a SIP message with a Route header (possibly) rewritten, a Via header added, and in some cases, additional manipulation of the SIP messages. To do its job, the ESRP has interfaces to the ECRF for location based routing information, as well as various event notification sources to gather state, which is used by its Policy Routing Function (PRF).

For typical 1 9-1-1 calls received by an ESRP it;

1. Evaluates a policy “rule set” for the queue the call arrives on
2. Queries the location-based routing function (ECRF) with the location included with the call to determine the "normal" next hop (smaller political or network subdivision, PSAP or call taker group) URI.
3. Evaluate a policy rule set for that URI using other inputs available to it such as headers in the SIP message, time of day, PSAP state, etc.

The result of the policy rule evaluation is a URI. The ESRP forwards the call to the URI (which is a queue as above).

The ESRP may also handle calls to what used to be called “administrative lines,” meaning calls directed to a 10-digit number listed for a particular PSAP. It is recommended that such calls route through the BCF to an ESRP and be subject to the same security and policy routing as regular 9-1-1 calls. Such calls would not have a Geolocation header and the ESRP would not query an ECRF, but would use the 10-digit number to map to a PSAP URI (the same URI which the ECRF would yield), and use that URI as the “normal next hop” used to select the policy rule set to evaluate.

An ESRP is usually the “outgoing proxy server” for calls originated by the PSAP. The ESRP would route calls within the ESInet, and would route calls to destinations outside the ESInet through an appropriate gateway or SIP trunk to a PSTN or other carrier connection. Call backs to the original caller are an example of such outgoing calls to external destinations. No policy rule set evaluation is used for outgoing calls. While an ESRP could be an incoming proxy server for non-emergency calls, such use is beyond the scope of this standard.

5.2.1.2 Call Queuing

The destination of every routing decision is conceptually a queue of calls. The queue can be large or small, it can have one or many sources entering calls on a queue, it can have one or many sources

taking calls off the queue. All queues defined in this document are normally First In First Out. A queue is identified by a unique SIP URI. A queue is managed by an ESRP. A call sent to the queue URI must route to the ESRP that manages it. Calls are enqueued by forwarding them to the URI (which is usually obtained by policy rule evaluation of an upstream ESRP). Calls are dequeued by the ESRP sending the call to a downstream entity (ESRP or endpoint such as a call taker or IMR).

ESRPs may, and often will, manage multiple queues. For example, an ESRP may manage a queue that is used for normal 9-1-1 calls routed to the local ESI-net, and one or more queues for calls that are diverted to it by ESRPs from other areas which are overloaded. Each queue must have a unique URI that routes to the ESRP.

In practice, some proxy servers may be simple RFC 3261 [12] compliant servers making simple routing decisions per RFC3264. In such cases, the queue is considered to have a length of 1 and its existence can be ignored.

The ESRP managing a queue may have policy that controls which entities may enqueue and dequeue calls to the queue. The dequeueing entity registers (DequeueRegistration) to receive calls from the queue. The ESRP would return a call from an entity not in its policy with a 404 error.

The ESRP will maintain a QueueState notifier, and track the number of calls in queue for the queues that it manages.

5.2.1.3 QueueState Event Package

QueueState is an event that indicates to an upstream entity the state of a queue. The SIP Notify mechanism described in RFC 3265 is used to report QueueState. The event includes the URI of the queue, the current queue length, allowed maximum length and a state enumeration including:

- Active: one or more entities are actively available or are currently handling calls being enqueued
- Inactive: no entity is available or actively handling calls being enqueued
- Disabled: The queue is disabled by management action and no calls may be enqueued
- Full: The queue is full and no new calls can be enqueued on it.
- Standby: the queue has one or more entities that are available to take calls, but the queue is not presently in use. When a call is enqueued, the state changes to “Active”.

QueueState need not be implemented on simple routing proxy or when queue length is 1 and only one dequeuer is permitted.

Event Package Name: nena-QueueState

Event Package Parameters: None

SUBSCRIBE Bodies: standard RFC4661 + extensions filter specification may be present

Subscription Duration Default 1 hour. 1 minute to 24 hours is reasonable.

NOTIFY Bodies: MIME type application/vnd.nena.queuestate+xml

Parameter	Condition	Description
-----------	-----------	-------------

queue	Mandatory	SIP URI of queue
queueLength	Mandatory	Integer indicating current number of calls on the queue.
maxLength	Mandatory	Integer indicating maximum length of queue
state	Mandatory	Enumeration of current queue state (e.g., Active/Inactive/Disabled)

Notifier Processing of SUBSCRIBE Requests

The Notifier (i.e., the ESRP) consults the policy (queueState) to determine if the requester is permitted to subscribe. If not, the ESRP returns 603 Decline. The ESRP determines whether the queue is one of the queues managed by the Notifier. If not, the ESRP return 488 Not Acceptable Here. If the request is acceptable, the Notifier returns 202 Accepted.

Notifier Generation of NOTIFY Requests

When state of the queue changes (call is placed on, removed from the queue, or management action/device failure changes the “state” enumeration), a new NOTIFY is generated, adhering to the filter requests.

Subscriber Processing of NOTIFY Requests: No specific action required.

Handling of Forked Requests: Forking is not expected to be used with this package.

Rate of Notification

This package is designed for relatively high frequency of notifications. The subscriber can control the rate of notifications using the filter rate control [113]. The default throttle rate is one notification per second. The default force rate is one notification per minute. The Notifier must be capable of generating NOTIFYS at the maximum busy second call rate to the maximum number of downstream dequeuing entities, plus at least 10 other subscribers.

State Agents: No special handling is required.

Race conditions exist where a dequeued call may be sent to an entity that just became congested. A call/event sent to a queue which is Inactive or Disabled, or where the current queue length is equal to or greater than the allowed maximum queue length will have an error (486 Busy Here) returned by the dequeuer. An ESRP that dequeues a call, sends it to a downstream entity and receives a 486 in return must be able to either re-enqueue the call (at the head of the line) or send it to another dequeuing entity. Note that the upstream ESRP may be configured with policy rules that will specify alternate treatment based on downstream queue state.

ESRPs normally send calls to downstream entities that indicate they are available to take calls. “Available” however, is from the downstream entities point of view. Network state may preclude an



upstream entity from sending calls downstream. Normal SIP processing would eventually result in timeouts if calls are sent to an entity that never responds because the packets never arrive. Timeouts are long however, and a more responsive mechanism is desirable to ensure that rapid response to changing network conditions route calls optimally.

If active calls are being handled, the upstream entity knows the downstream entity is connected. However, some routes are seldom used, and a mechanism must be provided that ensures the connectedness of each entity remains known.

For this purpose, we ensure relatively frequent NOTIFYs of the QueueState event. Successful completion of the NOTIFY is indication to the upstream entity that calls sent to the downstream entity should succeed. The subscription may include a “force” and/or “throttle” filter [113] to control the rate of Notification.

5.2.1.4 DequeueRegistration Event Package

DequeueRegistration is web service whereby the registering entity becomes one of the dequeuing entities, and the ESRP managing the queue will begin to send calls to it. The registration includes a value for DequeuePreference which is an integer from 1-5. When dequeuing calls, the ESRP will send calls to the highest DequeuePreference entity available to take the call when it reaches the head of the queue. If more than one entity has the same DequeuePreference, the ESRP will attempt to fairly distribute calls to the set of entities with the same DequeuePreference measured over tens of minutes.

DequeueRegistrationRequest

Parameter	Condition	Description
queue	Mandatory	SIP URI of queue
dequeuePreference	Optional	Integer from 1-5 indicating queuing preference.

DequeueRegistrationResponse

Parameter	Condition	Description
errorCode	Optional	Error Code

Error Codes

- 100 Okay No error
- 506 Bad queue
- 507 Bad dequeuePreference
- 508 Policy Violation
- 504 Unspecified Error

The ESRP will subscribe to the QueueState event for each dequeuing entity to determine its availability to take calls. Normally, a dequeuing entity is another queue maintained at the downstream entity, although the queue maintained at the terminating ESRP, which is normally the PSAP, would use call taker state rather than queue state to determine availability to dequeue calls from its upstream ESRP.

5.2.1.5 Policy Routing Function

Policy Routing refers to the determination of the next hop a call or event is forwarded to by an ESRP. The PRF evaluates two or more policy rulesets: one set determined by the queue the call arrives on, the other determined by the result of an ECRF query with the location of the caller.

The PRF in an ESRP accepts calls directed to a specific queue URI. From that URI, it extracts its own “OriginationPolicy” from its policy store for that URI and executes the ruleset. The rules normally include at least one action LoSTServiceURN(<urn>) where urn is a service urn (either urn:service:... or urn:ena:service:...). Upon encountering the LoSTServiceURN action, the PRF queries its (configured) ECRF with the location received in the call using the urn parameter in the action. The resulting URI is a variable called “NormalNextHop”. The PRF extracts a “TerminationPolicy” from its policy store associated with the domain of NormalNextHop and executes the ruleset associated with that policy. The rules normally include the action “Route”. The PRF forwards the call to the route. It would be common for the route of a 9-1-1 call intended for a PSAP in a normal state to be identical to the “NormalNextHop” URI, that is, if the ECRF query returned sip:psap1@example.com, then the TerminationPolicy ruleset for sip:psap1@example.com would have a Route(sip:psap@example.com) or a Route(NormalNextHop), which is equivalent, if the state of psap1 is nominal. If the policy store the ESRP uses does not contain a TerminationPolicy rule set for the NormalNextHop URI, the ESRP will route the call directly to that URI.

The destination of a Route action is usually the URI of a queue, but a simple proxy server can be the next hop. The PRF has access to queue state of downstream entities and can use that state in evaluating rules. Rules normally have a Route action that sends the call to a queue that is Available and not full. A Route may also be a URI that routes to an Interactive Multimedia Response system, conforming to RFC4240 [43], that plays an announcement (in the media negotiated by the caller) and potentially accepts responses via DTMF, KPML or other interaction styles.

The syntax is Route(<recipient>, <cause>), where recipient is a URI which will become the Request URI for the outgoing SIP message, and the <cause> is a value used with the Reason header associated with a History-Info header. The <cause> values are defined in a Registry which this document establishes.

Other Actions that may occur in a Termination-Policy include:

- Busy() which returns 600 Busy Everywhere to the caller
- Notify(<recipient>, <eventCode>, <urgency>, <severity>, <certainty>), which sends a NOTIFY containing a CAP message to any entity subscribing to the Normal-NextHop’s ESRPnotify event for that reason code. This may be used, for example, to advise other

entities that calls are being diverted, etc. If the <recipient> is a service urn, the CAP message is wrapped in a SIP MESSAGE and is routed via the ECRF to the proper recipients.

By using these mechanisms, the full range of call treatments can be applied to any class of call for any circumstance based on the PRF ruleset.

Rules may make use of the following variables. Several require the ESRP to use the SIP-based notification mechanism described in RFC 3265 to obtain the value of the variable.

1. ElementState, expressed as Elementstate.<domain> where <domain> is a hostname, or a URI. If a URI is specified, the Domain function is used to extract the domain from the URI. The domain must be that of a PSAP that the ESRP can subscribe to the ElementState package for.
2. QueueState (and implied “Not Reachable” state), expressed as QueueState.<queue> where <queue> is the name of a queue
3. SecurityPosture , expressed as SecurityPosture.<domain> where <domain> is a hostname, or a URI. If a URI is specified, the Domain function is used to extract the domain from the URI. The domain must be that of an agency or element that the ESRP can subscribe to the SecurityPosture package for.
4. CallSuspicion, the BCF’s opinion of the call, expressed as CallSuspicion.<suspicionLevel>. See Section 5.1.2.1.
5. Call Source (as defined in the Via headers of the INVITE), interpreted by the ESRP to ignore intra ESInet Vias, and other intermediaries. CallSource should be the ESRP’s best determination of the domain of the originating network that handled the call. If there is more than one, the last SP prior to the ESInet should be returned. If there are no originating networks, CallSource returns the domain of the caller.
6. Any header in the call INVITE message, expressed as Invite.<header name>. Even though a call may be initiated with a sip Message, Invite.<header name> is used to specify the headers
7. Any element in a body that is included in the message which is XML encoded, expressed as Body <mimetype><element tag>. If a body contains more than one part (of a multipart) with the same mimetype, only the first part with that mimetype can be used. This capability may be used to route on parameters in a CAP message.
8. The location used for routing, expressed as PIDF.<element name>
9. Any element in the Additional Data about a call or caller or location structures if available, expressed as AcallData.<element name>, AcallerData.<element name> or AlocationData.<element name>. See Sections 5.10 and 8.
10. Time of Day, expressed as TimeOfDay or DayOfWeek, where TimeOfDay is wall clock time (0000 to 2359) and DayOfWeek is Mon, Tue, Wed, Thu, Fri, Sat, Sun.
11. RequestURI (URI call was sent to ESRP with)
12. ECRF query results (Normal-NextHop).

13. The queue the call was received on (IncomingQueue)

Rules have a priority. If more than one rule yields a value for NextHop, the rule with the highest priority prevails. If more than one rule with the same priority yields a value for NextHop, the ESRP chooses randomly from the results with approximately uniform distribution.

Usually, there is a “default” rule for use when everything is in normal status. Most calls will route via this rule. For example IF True THEN Route(NormalNextHop) {10}; Other rules exist for unusual circumstances.

In congestion for typical transient overload, a specific PSAP would be delegated to take diverted calls (via a rule other than the default rule). A call is said to be diverted when it is sent to a PSAP other than the one serving the location of the caller, usually due to some failure or overload condition. A queue is established for that route, with one dequeuing PSAP. Such a diversion PSAP would be accepting calls on its normal queue as well as the diversion queue. Its rules can differentiate such calls from the queue they arrive on.

For more extensive overload, a group of PSAPs would subscribe to take calls from a designated queue. For example, all PSAPs in neighboring counties might subscribe to a low priority rule for overload for a county PSAP. Similarly, all NG9-1-1 PSAPs in a state might dequeue for a “Denial of Service Attack” queue, or interstate queues may be established that have a “ripple” effect (using priority) to spread calls out when the state queue becomes busy.

ESRPs managing a queue may become a dequeuer for one or more upstream queues. Origination rules at the ESRP can govern how such calls are handled, as the URI used to get the call to the ESRP (which could be the name of a queue maintained at the ESRP) is an input to the PRF. When handling diverted calls, no ECRF dip may be needed (and thus no termination policy ruleset is used). In such a case, the origination policy ruleset would determine NextHop. Rules can determine the priority of multiple queues feeding calls to the ESRP. PSAP ESRPs may dequeue for multiple call queues, placing them on internal queues for call takers.

5.2.1.6 ESRPnotify Event Package

The ESRP sends a Notify for this event when the PRF encounters a Notify action. It is used to inform other agencies or elements about conditions in an incoming call they may be interested in. For example, a call that contains an AdditionalCallData record may have a telematics dataset that indicates a severe injury. The ruleset may issue the ESRPnotify event to a helicopter rescue unit to inform them that their services may be needed. The ESRPnotify event is defined as follows:

Event Package Name: nena-ESRPnotify

Event Package Parameters:

Parameter	Condition	Description
Normal-NextHop	Mandatory	URI of downstream entity occurring in a Termination-Policy
ESRPEventCode	Mandatory	Enumeration of event codes.



		May occur more than once
--	--	--------------------------

SUBSCRIBE Bodies: standard RFC4661 + extensions. Filter specification may be present

Subscription Duration Default 1 hour. 1 minute to 24 hours is reasonable.

NOTIFY Bodies: MIME type application/vnd.nena.ESRProute+xml

The ESRPnotify NOTIFY contains a Common Alerting Protocol (CAP) message, possibly wrapped in an EDXL wrapper. The <area> element of the CAP message contains the location of the caller in the Geolocation header, although <area> is always location by value. The Geolocation header must also be copied to the NOTIFY headers. The CAP message is in the body of the NOTIFY, with MIME type **application/common-alerting-protocol+xml**.

A list of the parameters on the notification will be provided in a future edition of this document

Note:

Note: If the URI in the Notify action in a rule contains a service urn, then the CAP message is sent to entities whose service boundaries intersect the location of the caller where the service URN matches that in the Notify action. In such a case, a SIP Message is used, rather than a SIP NOTIFY.

The <identifier> is determined by the ESRP, and must be globally unique. The identifier in the CAP message is not the same as the Call Identifier assigned in the ESInet, but the log contains the record that relates the two.

The <sender> is the NextHop URI (i.e., the downstream entity whose rules invoked the Notify).

The <addresses> element contains the URIs of the subscribers to the event that are being notified.

An <info> element must be included. The element must contain an <event code>. The <valueName> must be “NENA-EsrpNotify”. This document defines a registry, “EsrpNotifyEventCodes” which registers values that may be used in an <event code>. The initially defined values in the registry can be found in Section 12.9. The <event category> is determined from the registry: each event code has a corresponding category

<urgency>, <severity> and <certainty> are copied from the parameters in the Notify action from the rule.

If there are Call-Info headers containing Additional Data (Call or Caller), they must be sent in the CAP message in a <parameter> element. Additional Call data has a <value name> of ADDLCALL and Additional Caller data has a <value name> of ADDLCALLR. The URI is the <value> element.

A digital signature should be included in the CAP message. The message should not be encrypted. TLS may be used on the SIP MESSAGE transmission to encrypt the message.

The CAP message may be enclosed in an EDXL wrapper. If it is, the body of the MESSAGE will contain a section **application/emergency-data-exchange-language+xml**.

Notifier Processing of SUBSCRIBE Requests

The Notifier (the ESRP) consults the policy (NotifyPermissions) for Normal-NextHop to determine if the requester is permitted to subscribe. If not permitted, the ESRP returns 603 Decline. The ESRP determines if at least one policy it uses contains a Notify action with that event code. If not, the ESRP returns a 488 Not Acceptable Here. If the request is acceptable, the ESRP returns 202 Accepted.

Notifier Generation of NOTIFY Requests

When the Notify(ESRPRouteEventCode) action is present in the rule that determines routing, send NOTIFY to any subscriber requesting that notification (based on the Normal-NextHop whose policy is being evaluated and the ESRPRouteEventCode present in the action.

Subscriber Processing of NOTIFY Requests: No specific action required.

Handling of Forked Requests: Forking is not expected to be used with this package.

Rate of Notification

A notification for each call/event handled by the ESRP could be sent. Rate controls [113] may be used to limit Notifications.

State Agents: No special handling is required.

5.2.1.7 Processing of an INVITE transaction

When the ESRP receives an INVITE transaction it first evaluates the Origination ruleset for the queue the call arrived on. If a LoSTServiceURN action is encountered it looks for the presence of a Geolocation header. If present the ESRP evaluates the header and extracts the location in the Geolocation header [10]. Each ESRP must be capable of receiving location as a value or a reference, and must be provisioned with credentials suitable to present to all LISs in its service area to be able to dereference a location reference using either SIP or HELD.

The ESRP must be able to handle calls with problems in location. This can occur if the call is originated by an element outside the ESInet, the call is to an emergency service URN, and there is no Geolocation header. This also occurs if the location contents are malformed, the LIS cannot be contacted, the LIS refuses to dereference, the LIS returns a malformed location value or the ESRP encounters another error that results in no location. In all such cases the ESRP must make a best effort to determine a suitable default location to use to route the call. The call source, IP address of the caller or other information from the INVITE may be used to determine the best possible default location. It is felt that the earlier in call processing that bad or missing location is determined, the more likely the ESRP will have information needed to get the best possible default location, and downstream entities will be in a worse position to do that.

The ESRP then queries its local (provisioned) ECRF with the location, using the service urn specified and the value of the Route header in the LoSTServiceURN action parameter. For example, the originating ESRP receiving an emergency call from outside the ESInet where there are no intermediary ESRPs in its service area (meaning the originating ESRP routes calls directly to the PSAP) may use the service "urn:nena:service.sos.psap ". The ECRF returns a URI for that service.

Calls to an administrative number do not have location and are mapped by a provisioned table in the ESRP from the called number to a URI.

The ESRP retrieves the terminating policy ruleset for the URI. The PRF evaluates the ruleset using the facts available to it such as PSAP state, time of day, queue state, information extracted from the INVITE, etc. The result is a URI of a queue. The ESRP attempts to forward the call to the URI, using the DNS to evaluate the URI into an IP address. DNS may provide alternate IP addresses to resolve the URI. Normal SIP and DNS processing is used to try these alternate IP addresses. Should no entity respond, the ESRP must provide the call with a provisioned treatment such as returning busy. Note that normally, the state of the downstream elements that would appear in the URI report their state to the ESRP and the ruleset would use that state to specify an alternate route for the call.

Calls that are received by an ESRP which originate inside the ESInet are routed per normal SIP routing mechanisms. Calls to E.164 telephone numbers not otherwise provided for in the ESRP provisioning must be routed to a provisioned gateway or SIP Trunk interconnected to the PSTN.

5.2.1.8 Processing a BYE Transaction

An ESRP processes BYEs per RFC3261.

5.2.1.9 Processing a CANCEL transaction

An ESRP processes CANCELs per RFC3261.

Note: The ESRP should have a way to notify a PSAP that a call arrived at the ESRP, but was CANCELLED before the INVITE was sent to the PSAP. This would be one case of abandoned call. This will be covered in a future edition of this standard.

5.2.1.10 Processing an OPTIONS transaction

An ESRP processes OPTIONS transactions per RFC3261. OPTIONS is often used as a “keep alive” mechanism. During periods of inactivity, the ESRP should periodically send OPTIONS towards its upstream entities and expect to see OPTIONS transactions from its downstream dequeuing entities.

5.2.2 Interface Description

5.2.2.1 Upstream Call Interface

The ESRP has an upstream SIP interface that typically faces a BCF for the originating ESRP or an upstream ESRP for an intermediate or terminating ESRP. The upstream SIP call interface for the originating ESRP must only assume the minimal methods and headers as define in Section 4.1.1 but must handle any valid SIP transaction. All other ESRPs must handle all methods and SIP headers. The ESRP must respond to the URI returned by the ECRF and/or specified in a Route action for a rule for the upstream service the ESRP receives calls from. The ESRP must assure that pager mode Instant Messages route to the same PSAP per Section 4.1.9

The upstream SIP interface is also used for calls originated inside the ESInet, where the ESRP is the outgoing proxy for a PSAP. Calls originated in the ESInet and destined for agencies within the ESInet are routed by the ESRP using normal SIP routing methods. Calls originated in the ESInet

and destined for external termination (such as call backs) are routed to gateways or SIP trunks terminated by a carrier.

The upstream interface on the originating ESRP must support UDP, TCP, and TCP/TLS and may support SCTP transports. The upstream interface on other ESRPs must implement TCP/TLS but must be capable of fall-back to UDP. SCTP support is optional. The ESRP should maintain persistent TCP and TLS connections to downstream ESRPs or UAs that it serves.

5.2.2.2 Downstream Call Interface

The ESRP downstream call interface typically faces a downstream ESRP for all but the terminating ESRP, which typically faces user agents. The downstream SIP call interface must implement all SIP methods to be able to propagate any method invoked on the upstream call interface. The downstream interface may add any headers noted in section 4.1.2 permitted by the relevant RFCs to be added by proxy servers. The INVITE transaction exiting the ESRP must include a Via header specifying the ESRP. It may include a Route header. The Request URI remains urn:service:sos (although the ESRP may not depend on that) and it replaces the top Route header with the next hop URI (this is described in -phonebcp [59]). The ESRP adds a History-Info and Reason headers per Section 4.1.7 using the cause code specified in the Route action if cause is specified (which it would be for a diverted call).

A call entering the ESInet is initially assumed to be a new Incident. Thus, the first ESRP in the path adds a Call-Info header with a purpose parameter of “nena-IncidentId” and a new Incident Tracking Identifier per Section 3.1.5. The ESRP also creates a new Call identifier (Section 3.1.4) and adds a Call-Info header with a purpose parameter of “nena-CallId”.

The downstream interface must implement TCP/TLS towards downstream elements, but must be capable of fall-back to UDP. SCTP support is optional. No ESRP may remove headers received in the upstream call interface; all headers in the upstream message must be copied to the downstream interface except as required in the relevant RFCs. The ESRP should maintain persistent TCP and TLS connections to downstream ESRPs.

The downstream SIP interface may also accept calls originating within the ESInet.

5.2.2.3 ECRF interface

The ESRP must implement a LoST interface towards a (provisioned) ECRF. The ESRP must use a TCP/TLS transport and must be provisioned with the credentials for the ECRF. The ESRP should maintain persistent TCP and TLS connections to the ECRF.

The ESRP must use the ECRF interface with the "urn:nena:service:AdditionalLocationData" service URN when the relevant ruleset specifies an element in that structure. The same location used for the location-based route is used for the AdditionalLocationData query.

5.2.2.4 LIS Dereference Interface

The ESRP must implement both SIP Presence Event Package and HELD dereference interfaces. When the ESRP receives a location (in a Geolocation header on the upstream SIP interface) it uses the LIS dereference interface to obtain a location value to use in its ECRF query. The ESRP uses its PCA issued credentials to authenticate to the LIS¹⁴. The ESRP must use TCP/TLS for the LIS Dereference Interface, with fallback to TCP (without TLS) on failure to establish a TLS connection. The ESRP should maintain persistent TCP and TLS connections to LISs that it has frequent transactions with. A suggested value for "frequent" is more than one transaction per day.

5.2.2.5 Additional Data Interfaces

The ESRP must implement https clients for the AdditionalCallData services. These services may be invoked when the ESRP receives a call with a CallInfo [12] header with a "purpose" of "emergencyCallData", "emergencyCallerData" or "emergencyPSAPdata". The resulting data structure is an input to the Policy Routing Function. The ESRP must be able to accommodate multiple additional data services and structures for the same call.

Note: Multiple CallInfo headers with "emergencyCallData" may occur when more than one originating network handles the call and/or the device itself reports data. For example, a call may have additional data provided by a wireless carrier as well as a telematics service. The call may have more than one Call-Info header with emergencyCallerData when, for example, the call is from a residence wireline telephony service where there is more than one resident. When used in a routing rule, the PRF merges multiple AdditionalCall or AdditionalCaller data. If the merge results in conflicting information, the information derived earlier encountered Call-Info headers shall take precedence over information derived from subsequent Call-Info headers.

The ESRP should only invoke the web service when the relevant ruleset specifies an input from the AdditionalCallData/AdditionalCallerData/AdditionalPSAPdata structure.

The ESRP must also be able to query the ECRF for AdditionalLocationData when the policy rules are dependent on that data.

5.2.2.6 ESRP, PSAP and Call Taker State Notification and Subscriptions

The ESRP must implement the client side of the ElementState event notification packages. The ESRP must maintain Subscriptions for this package on every downstream element it serves. These state interfaces supply inputs to the Policy Routing Function.

¹⁴ The LIS must accept credentials issued to the ESRP traceable to the PCA. If a call is diverted to an alternate PSAP, it could be any willing PSAP, anywhere. The alternate PSAP must be able to retrieve location.

The ESRP must implement the server side of the ElementState event notification package and accept Subscriptions for all upstream ESRPs it expects to receive calls from. The ESRP must promptly report changes in its state to its subscribed elements. Any change in state which affects its ability to receive calls must be reported.

5.2.2.7 Time Interface

The ESRP must implement an NTP client interface for time-of-day information. The ESRP may also provide an interface to a hardware clock. The time of day information is an input to the Policy Routing Function as well as the logging interface

5.2.2.8 Logging Interface

The ESRP must implement a logging interface per Section **Error! Reference source not found.** The ESRP must be capable of logging every transaction and every message received and sent on its call interfaces, every query to the ECRF and every state change it receives or sends. It must be capable of logging the ruleset it consulted, the rules found to be relevant to the route, and the route decision it made.

Note: The specifics of the log entries will be provided in a future edition of this document.

5.2.3 Data Structures

The ESRP maintains an ElementState structure for its own state, and an ElementState structure for every downstream element it serves.

If the ESRP manages queues, it maintains a QueueState structure for each queues it manages, including the states of the entities registered to dequeue calls from the queue, the overall queue state, the number of calls in queue, the max number of calls allowed, and the current queue state.

The ESRP constructs AdditionalCallData, AdditionalCallerData and AdditionalLocationData structures when the relevant ruleset mentions elements from these structures and, in the case of call and caller data, the upstream Call Interface receives the appropriate CallInfo header with a URI for the AdditionalCallData/AdditionalCallerData dereferencing services.

5.2.4 Policy Elements

The ESRP uses an Origination-Policy ruleset for each queue it manages. For every URI the ECRF can return for the service query the ESRP makes (Normal-NextHop), it must have access to the appropriate Termination-Policy ruleset.

The ESRProuteEvent Policy determines which entities may subscribe to the ESRProute Event (see Section 5.2.1.6).

The queueState policy determines which entities may subscribe to the queueState event

The ElementState policy determines which entities may subscribe it its ElementState event

The DequeueRegistration policy determines which entities may subscribe to the DequeueRegistration event

The takeCallsOnQueues policy determines which queues this ESRP will dequeue from (that is, which queues it will subscribe to the dequeueRegistration and queueState events for)

Note: Specific policy document structures will be specified for each of the above in a future edition of this document.

5.2.5 Provisioning

The ESRP is provisioned with:

- The queues it manages
- The queues it dequeues from
- The default locations it uses, including (potentially) one for each origination domain, and an overall default location
- The ECRF it uses
- The Logging service it uses
- Mappings from 10-digit PSAP telephone numbers to URIs (if the ESRP handles 10 digit calls on behalf of PSAPs)
- The URI of a default route PSAP that takes calls when a route cannot be determined.

5.2.6 Roles and Responsibilities

An ESRP may be operated by a State, Regional or local 9-1-1 Authority. A terminating ESRP may be operated by a PSAP. The ESRP for non-originating ESRPs must supply a ruleset for the upstream ESRP.

5.2.7 Operational Considerations

To be provided in a future edition of this standard.

5.3 Emergency Call Routing Function (ECRF)

In i3, emergency calls will be routed to the appropriate PSAP based on the location of the caller. In addition, PSAPs may utilize the same routing functionality to determine how to route emergency calls to the correct responder. The NG9-1-1 functional element responsible for providing routing information to the various querying entities is the Emergency Call Routing Function (ECRF). An ECRF provided by a 9-1-1 Authority and accessible from outside the ESInet must permit querying by an IP client/endpoint, an IP routing proxy belonging to a VSP, a Legacy Network Gateway, an Emergency Services Routing Proxy (ESRP) in a next generation Emergency Services network, or by some combination of these. An ECRF accessible inside an ESInet must permit querying from any entity inside the ESInet. ECRFs provided by other entities may have their own policies on who may query them. An origination network may use an ECRF, or a similar function within its own network, to determine an appropriate route, equivalent to what would be determined by the authoritative ECRF, to the correct ESInet for the emergency call. The ECRF must be used within

the ESInet to route calls to the correct PSAP, and by the PSAP to route calls to the correct responders.

5.3.1 Functional Description

The ECRF supports a mechanism by which location information (either civic address or geo-coordinates) and a Service URN serve as input to a mapping function that returns a URI used to route an emergency call toward the appropriate PSAP for the caller's location. Depending on the identity and credentials of the entity requesting the routing information, the response may identify the PSAP or an Emergency Services Routing Proxy (ESRP) that acts on behalf of the PSAP to provide final routing to the PSAP itself. The same database used to route a call to the correct PSAP may also be used to subsequently route the call to the correct responder, e.g., to support selective transfer capabilities. Depending on the type of routing function requested, the response may identify a secondary agency.

5.3.2 Interface Description

5.3.2.1 Routing Query Interface

The ECRF shall support a routing query interface that can be used by an endpoint, ESRP, or PSAP to request location-based routing information from the ECRF. The ECRF takes the location information and Service URN received in a routing query and maps it to the destination URI for the call. The LoST protocol supports this functional interface in NG9-1-1.

When an ECRF receives a LoST query, the ECRF determines whether an authenticated user (e.g., an ESRP) originated the query and the type of service requested (i.e., emergency services). Authentication must apply for ESRPs and i3 PSAPs that initiate queries to the ECRF. TLS is used by all ECRFs within the ESInet, and credentials issued to the entity querying that are traceable to the PCA must be accepted. Devices and carriers outside the ESInet may not have credentials, TLS is not required, and the ECRF should assume a common public identity for such queries. Based on the identity and credentials of the query originator and the service requested, the ECRF determines which URI is returned in the LoST response, which could be a URI of a PSAP or a downstream ESRP. The same database used to route a call to the correct PSAP may also be used to subsequently route the call to the correct responder, e.g., to support selective transfer capabilities.

The LoST protocol is a query/response protocol defined by [61]. The client seeking routing information sends a LoST <findService> query to the server (in this case the ECRF). The ECRF responds to the query with a response message that contains the requested information (see <findServiceResponse> in section 4.5.1.1.2), an error indication (see <errors> in section 4.5.1.1.3), or a redirect to another ECRF (see <redirect> in section 4.5.1.1.4). The LoST protocol is a flexible protocol and is defined with many options. Many of the options provided in the LoST protocol are not specifically required to support emergency call routing.

5.3.2.1.1 Routing Query

The LoST protocol specifies the following query messages:

- <findService>

- <getServiceBoundary>
- <listServices>
- <listServicesByLocation>

The <findService> message is used to retrieve one or more contact URIs given a service URN and a location. Since the primary function of the ECRF is to support the routing of emergency calls, the ECRF must be capable of receiving, processing and responding to LoST <findService> query messages containing the "sos" service or a "sos" sub-service URN. See section 4.5.1.1.1 for an explanation of the LoST <findService> message. 9-1-1 Authorities may also choose to route other sos urns to the primary PSAP.

The ECRF may also support the other LoST query types (see [61] for details related to the <getServiceBoundary>, <listServices>, and <listServicesByLocation> query messages).

5.3.2.1.2 Routing Response

The LoST protocol describes the following response messages that can be used depending on the received query:

- <findServiceResponse>
- <findServiceBoundaryResponse>
- <listServicesResponse>
- <listServicesByLocationResponse>

The only response message that the ECRF is required to support is the <findServiceResponse> message. The ECRF shall be capable of generating a LoST <findServiceResponse> message (Section 4.5.1.1.2) an <errors> message (section 4.5.1.1.3), or a <redirect> message (section 4.5.1.1.4) in response to a received <findService> message.

The <findServiceResponse> message is composed of the elements listed in Table 4-2.

Table 4-2. <findServiceResponse> Message Elements

Element	Condition	Description
source	Mandatory	Identifies the authoritative generator of the mapping
sourceId	Mandatory	Identifies a particular mapping
lastUpdated	Mandatory	Describes when a mapping identified by the source and sourceId was last updated
expires	Mandatory	Identifies the absolute time when the mapping becomes invalid

<displayName>	Optional	Describes a human readable display name, e.g., the name of the PSAP serving the location
<service>	Mandatory	Identifies the service for which the mapping applies
<serviceBoundary>	Optional	Identifies the area where the URI returned would be valid
<serviceBoundaryReference>	Optional	Identifies the reference which could be used to access the service boundary for which the URI returned is valid
<serviceNumber>	Optional	Provides the emergency services dial string that is appropriate for the location provided in the query
<uri>	Conditional ¹⁵	Contains the appropriate contact URI for the service being requested
<path>	Mandatory	Contains the Via elements indicating the LoST servers that handled the request. Used for recursive operation.
<locationUsed>	Optional	Identifies the location used to determine the URI

¹⁵ The ECRF shall include a URI in a <findServiceResponse> message if one can be determined.

<locationValidation>	Optional	Indicates which elements of the civic location were “valid” and used for mapping, which elements were “invalid” and which elements were “unchecked”
----------------------	----------	---

The elements that make up the <findServiceResponse> message are described below:

- source - This element identifies the authoritative generator of the mapping (the LoST server that generated the mapping). LoST servers are identified by U-NAPTR/DDDS application unique strings, in the form of DNS name. For example, lostserver.notreal.com.
- sourceId - This element identifies a particular mapping at the LoST server and is unique among all the mappings maintained by the LoST server.
- lastUpdated - This element describes the date and time when this specific instance of mapping was updated. The date and time is represented in UTC format.
- expires - This element describes the date and time when a particular mapping becomes obsolete. The date and time are described using a timezoned XML type datetime. This element may optionally contain the values of “NO-CACHE” indicating that the mapping should not be cached and “NO-EXPIRATION” indicating that the mapping has no expiration instead of the date and time.
- <displayName> Element - The display name is a text string that provides an indication of the serving agency(ies) for the location provided in the query. This information might be useful to PSAPs that query an ECRF. This capability could be used to provide English Language Translation (ELT)-type information that PSAPs receive from ALI databases today.
- <service> Element - The <service> element identifies the service for which this mapping is valid. The ECRF is required to support the sos service. Support for other services will depend on local implementation.
- <serviceBoundary> - The <serviceBoundary> element identifies the geographical area where the returned mapping is valid. The intent of this parameter is to allow a mobile endpoint to realize that it is moved out of the area where a stored mapping is valid and trigger it to query for a new valid mapping. This element may be supported by the ECRF depending on local implementation.
- <serviceBoundaryReference> - The <serviceBoundaryReference> element identifies a reference that could be used to access the service boundary for the requested mapping. This parameter may be supported by the ECRF depending on local implementation.
- <serviceNumber> - The <serviceNumber> element contains the emergency services number that is appropriate for the location provided in the query. This will allow a foreign end device to recognize that an emergency number is being dialed.

- Uniform Resource Identifier (<uri>) - The <uri> specifies either the address of the PSAP or the ESRP that is appropriate for the location sent in the query message. The decision of whether to send the PSAP <uri> or the ESRP <uri> is based on whether the query is made by the end user, VSP Routing Proxy, i3 PSAP, or the ESRP. In i3, the end point and VSP Routing Proxy will receive an ESRP <uri>. Only authorized ESRPs and i3 PSAPs are entitled to receive a PSAP <uri>. Lower layer authorization procedures are used to identify the query originator.
- <path> - The <path> contains via elements indicating the ECRF(s) that handled the request.
- <locationUsed> - The <locationUsed> element identifies the location used to determine the URI.
- <locationValidation> - The <locationValidation> element identifies which elements of the received civic address were “valid” and used for mapping, which were “invalid” and which were unchecked. Since the ECRF is not responsible for performing validation, this parameter may not be returned, subject to local implementations.

If the proffered location is not specified as a point (that is the location in the query is a shape) and the shape intersects more than one service boundary with a given service URN, the response is the URI of the service boundary with the greatest area of overlap (with a tie breaking policy for the case of equal area of overlap).

If more than one service boundary for the same service URN at a given location exists in the ECRF, two <mapping>s will be returned. The querier (for example, a PSAP), must have local policy to determine how to handle the call. In some cases, the ECRF can use the identity of the querier, or a distinguished Service URN to return the URI of the correct agency. This condition only occurs for queries to an ECRF from within an ESInet. External queries will only return one (PSAP) URI.

The service boundary returned from an ECRF may not be the actual service boundary of the PSAP, or even that of the ESRP that will handle an emergency call from the location in the query. Instead, it may be a simpler shape chosen to have only a few points. For example, the polygon may be the largest rectangle that completely fits in the actual boundary measured from the location in the query. The service boundary returned at a point near a service boundary may represent a portion of the agency’s service boundary near the edge where the location exited the original boundary, and may be somewhat more complex, but still an approximation of the actual boundary. As the location sent in the query gets closer and closer to the actual service boundary, the area represented by the returned service boundary may be smaller, the number of points may be somewhat larger, and the fidelity to the actual service boundary may be greater. This minimizes the network bandwidth and compute load on the device.

5.3.2.1.3 Error and Warning Messages

If the ECRF is unable to completely fulfill a request, it shall return either an error or a warning message, depending on the severity of the problem.

If no useful response can be returned for the query, the ECRF shall return a LoST <errors> message with the appropriate "error type" element(s) as described in section 4.5.1.1.3 and section 13.1 of [61].

If the ECRF is able to respond to a query in part, it shall return a <warnings> element as part of another response element as described in section 13.2 of [61] and in section 4.5.1.1.3 for the Lost <findServiceResponse> message.

In both cases, the source attribute of the "error type" and "warning type" element(s) identifies the server that originally generated the error or warning (e.g., the authoritative server). When possible, the ECRF should populate the message and xml:lang attributes of the "warning type" and "error type" elements to more specifically identify the nature of the warning or error for logging and possible later troubleshooting purposes.

5.3.2.2 Data Source Interface

The ECRF's data source is a map, specifically, a set of layers from one or more source SIFs. A SIF layer replication interface, as described in Section 4.7, is used to maintain copies of the required layers. The ECRF is provisioned with the URI and layer names of its data sources. It has layers that define the locations (state/county/municipality/street/address), as well as service boundary polygons.

A resulting location-based URI associated with a routing request may undergo further modification at an ESRP due to policies related to such things as time of day, current congestion conditions, etc. (See Section 4.2.4 for further discussion.)

5.3.2.3 Time Interface

The ECRF must implement an NTP client interface for time-of-day information. The ECRF may also provide an interface to a hardware clock. The time of day information is an input to the mapping expiration time as well as the logging interface.

5.3.3 Data Structures

5.3.3.1 Data to Support Routing Based on Civic Location Information

The ECRF must be able to provide routing information based on location information represented by a civic address. To do so, it is expected the ECRF will represent the geographic service boundary in a manner that allows the association of a given address with the service boundary it is located within. Theoretically, the ECRF maintains the civic address data as the SIF layers used to provision it, using a geocode followed by point-in-polygon algorithms to determine the service boundary the civic address is located within. The ECRF may internally compute a tabular civic address form of data representation with the associated URI resulting from the point-in-polygon operation. This would reduce the LoST query resolution for a civic address to a table lookup. However, if the provisioning data changes, the ECRF must respond immediately to the change, which may invalidate (for at least some time) the precalculated tabular data.

The ECRF shall be capable of receiving the following data elements that may be present in the civic location information received in a routing query from an NG9-1-1 element (i.e., VoIP endpoint, VSP Routing Proxy, ESRP, i3 PSAP), identifying the service boundary the civic location described by the data elements lies within, and performing a mapping to determine the associated routing data. RFC 4776 ([8]) provides a full set of parameters that may be used to describe a civic location. Specifically, RFC 5139 ([76]) lists several civic address types (CAtypes) that require support in the formal PIDF-LO definition that are not in RFC 4119 ([6]).

Table 4-3. Civic Location Data Elements

Label	Description	Type	Example
country	2-letter ISO code	alpha	US
A1	national subdivision (e.g., state)	alpha	NY
A2	county, parish	alpha	King’s County
A3	city, township	alpha	New York
A4	city division, borough	alpha	Manhattan
A5	neighborhood	alpha	Morningside Heights
A6 ¹⁶	street	alphanumeric	Broadway
PRD	leading street direction	alpha	N
POD	trailing street suffix	alpha	SW
STS	street suffix	alpha	Ave
HNO	house number	alphanumeric	123
HNS	house number suffix	alphanumeric	A, 1/2
LMK	Landmark or vanity address	alphanumeric	Columbia University
LOC	additional location info	alphanumeric	South Wing
NAM	name (residence or office occupant)	alphanumeric	Town Barber Shop
PC/ZIP	postal/ZIP code	alphanumeric	10027-0401

¹⁶ RD should be used in preference to A6. A6 must be accepted by the ECRF

BLD	building (structure)	alphanumeric	Low Library
UNIT	unit (apartment, suite)	alphanumeric	Apt 42
FLR	floor	alphanumeric	4
ROOM	room	alphanumeric	450F
PLC	type of place	alpha	
PCN	postal community name	alpha	Leonia
POBOX	post office box (P.O. box)	numeric	12345
ADDCODE	additional code	alphanumeric	132030000003
SEAT	Seat (desk, workstation, cubicle)	alphanumeric	WS 181
RD	primary road name	alphanumeric	Broadway
RDSEC	road section	alphanumeric	14
RDBR	branch road name	alphanumeric	Lane 7
RDSUBBR	sub-branch road name	alphanumeric	Alley 8
PRM	Road name pre- modifier	alphanumeric	Old
POM	Road name post- modifier	alphanumeric	Service

No individual element in a civic address stored in the ECRF shall be longer than 256 bytes.

To provide this data, the ECRF uses layer replication of one or more SIFs that cover the ECRF’s service area. The source SIF may be provided by 9-1-1 Authorities, other government agencies with GIS responsibility such as a county mapping agency and/or responders who define their own service areas. The ECRF mapping data is provided by:

Table 4-4. Civic Location Data Elements

PIDF Element	Layer Name	Geometry or
-------------------------	-------------------	------------------------

		Attribute
country	None, provisioned	None
A1	State	Name
A2	County	Name
A3	Municipality	Name
A4	City Division	Name
A5	Neighborhood	Name
A6	Street Centerline or Street Geometry	Name
PRD	Same as A6	PRD
POD	Same as A6	POD
STS	Same as A6	STS
HNO	Address Point or Parcel or sub parcel	HNO
HNS	Same as HNO	HNS
LMK	Same as HNO	LMK
LOC	Same as HNO	LOC
NAM	Same as HNO	NAM
PC/ZIP	ZIP code	Name
PCN	ZIP code	Post Office
RD	Same as A6	Name
PRM	Same as A6	PRM
POM	Same as A6	POM

5.3.3.2 Service Boundaries

Location represented by geodetic coordinates provides data that corresponds to a specific geographic location point. It is possible to represent a larger geographic area, such as a PSAP serving area as a polygon set. More than one polygon may occur in the set when the service area has holes or non-contiguous regions.

For each service urn supported by an ECRF, one or more layers will provide polygon sets associated with URIs. Two attributes are used on these polygons:

URN: The service URN this boundary is associated with

URI: The URI returned if the location is within the boundary

The ECRF computes a response to a LoST query by finding the polygon with the service URN attribute matching that provided in the LoST query containing the location, and returning the URI attribute of that polygon set.

5.3.3.3 Routing Data – URI Format

For an end-to-end IP network where the caller is an IP endpoint and the PSAP is accessed over an IP network, routing information will be in the form of a URI. The URI may identify a PSAP, or an ESRP that will forward calls to the appropriate PSAP. The source of the query determines which URI is returned. Therefore, it will be necessary to be able to associate multiple URIs with a service boundary. URI format is described in IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*. URIs can be of variable length. It is suggested that the length allowed for a URI be as compact as possible, not exceeding 1.3 K, which is the maximum size of a packet on the ESInet, less any header information.

5.3.3.4 Other Data

- ECRF Identifier - contains a LoST application unique string identifying the authoritative generator of the mapping
- ECRF mapping identifier - identifies a particular mapping and contains an opaque token that must be unique among all different mappings maintained by the authoritative source for that particular service. For example, a Universally Unique Identifier (UUID) is a suitable format.
- Date and time mapping was last updated – contains the XML data type dateTime in its timezoned form, using canonical UTC representation with the letter 'Z' as the time zone indicator.
- Date and time of mapping expiration – contains a timezoned XML type dateTime, in canonical representation. Optionally, this attribute may contain the values of 'NO-CACHE' and 'NO-EXPIRATION' instead of a dateTime value. The value 'NO-CACHE' is an indication that the mapping should not be cached. The value of 'NO-EXPIRATION' is an indication that the mapping does not expire.
- Display name – contains a description of the service using a string that is suitable for display to human users, which may be annotated with the 'xml:lang' attribute that contains a language tag to aid in the rendering of text. The display name is used as the “English Language Translation” (ELT) and must be provided for all responder URIs.
- Service identifier for which mapping is valid
- Service boundary definition – Service boundaries must be defined using exactly one of the two baseline profiles (i.e., geodetic-2d, civic), in addition to zero or more additional profiles. A location profile MUST define:
 - The token identifying it in the LoST location profile registry;

- The formal definition of the XML to be used in requests, i.e., an enumeration and definition of the XML child elements of the <location> element;
- The formal definition of the XML to be used in responses, i.e., an enumeration and definition of the XML child elements of the <serviceBoundary> element;
- The declaration of whether geodetic-2d or civic is to be used as the baseline profile. It is necessary to explicitly declare the baseline profile as future profiles may be combinations of geodetic and civic location information.

To support the delivery of service boundary information using the geodetic 2d profile in a response to a client, the ECRF must support the following location shapes:

- Point
- Polygon
- Circle
- Ellipse
- Arcband

To support civic service boundaries, each service boundary consists of the set of civic addresses that fall within the service boundary, namely all the addresses that textually match the civic address elements provided, regardless of the value of the other address elements. A location falls within the mapping's service boundary if it matches any of the service boundary elements.

Note that the provisioning interface to the ECRF is the SIF layer replication protocol, and thus always delivers a geodetic service boundary definition to the ECRF. The ECRF may compute a civic representation of the boundaries internally. A trivial example is a service boundary polygon exactly matching a state, county or municipality boundary.

- Service boundary reference definition - The identifier must be globally unique. It uniquely references a particular boundary. It could be a locally unique token and the hostname of the source of the boundary separated by an '@'
- Service number - contains a string of digits, * and # that a user on a device with a 12-key dial pad could use to reach that particular service.

5.3.4 Recursive and Iterative Query Resolution

An ECRF may receive a query for a location that is not within its internal database. For such queries, it may redirect the querier to another ECRF (iteration), or it may query the other ECRF and return the result to the querier (recursion). Which action it takes is primarily determined by a query parameter, but may be limited by provisioning and may depend on the location in the query. For example, it may allow recursive resolution for any in-state queries but insist on redirecting an out-of state query to the national forest guide, see section 5.13.

Each state should have an ECRF and/or forest guide which can resolve, by iteration or recursion, any query. The State ECRF should have boundaries for every authoritative ECRF in the state as well as

the ability to redirect out of state queries to the national forest guide. It may have knowledge of adjacent state ECRFs. Any lower level ECRF can refer or redirect any query it cannot handle to its state ECRF, which can refer or redirect to another ECRF in the state or can consult the national forest guide. It is recommended that ECRFs handle in-state queries via recursion.

All ECRFs must provide the proper <path> element as described in RFC5222.

5.3.5 Coalescing Data and Gap/Overlap Processing

ECRFs may coalesce data from several 9-1-1 Authorities. The resulting database appears to be a seamless route database for the union of the service areas of each 9-1-1 authority. Such ECRFs are provisioned to accept data from multiple SIFs.

In some local SIFs, for convenience, some area beyond the service boundary of the PSAPs the 9-1-1 Authority provides data for may be present. If so, this area must be marked with an “Informative” attribute, and the ECRF will ignore it.

When the data is coalesced, boundaries may have gaps and overlaps. The relevant 9-1-1 Authorities should endeavor to address such issues early, but despite best efforts, the ECRF may encounter a gap or overlap. The ECRF must have a provisionable threshold parameter that indicates the maximum gap/overlap that is ignored by the ECRF. This threshold is expressed in square meters. Gaps or overlaps that are smaller than this parameter must be handled by the ECRF using an algorithm of its choice. For example, it may split the gap/overlap roughly in half and consider the halves as belonging to one of the constituent source SIFs.

The ECRF must report gaps and overlaps larger than the provisioned threshold. To do so, it makes use of the GapOverlap event. All 9-1-1 Authorities who provide source GIS data to an ECRF must subscribe to its GapOverlap event. The event notifies both agencies when it receives data that shows a gap or overlap larger than the threshold. The notification includes the layer(s) where the gap/overlap occurs, whether it is a gap or an overlap, and a polygon that represents the gap or overlap area.

The response of the agencies must be updates to the data that address the gap/overlap. The ECRF will repeat the notification at least daily until it is resolved (by changing the SIF data so the gap/overlap is eliminated or at least smaller than the threshold parameter). During the period when the gap/overlap exists, notifications have been issued, and queries arrive (which could be at call time) with a location in the gap/overlap, the ECRF must resolve the query using an algorithm of its choice. For example, it may split the gap/overlap roughly in half and consider the halves as belonging to one of the constituent source SIFs.

The GapOverlap event is defined as follows:

Event Package Name: nena-GapOverlap

Event Package Parameters: none

SUBSCRIBE Bodies: standard RFC4661 + extensions filter specification may be present

Subscription Duration Default 24 hour. 1 hour to 96 hours is reasonable.

NOTIFY Bodies: MIME type application/vnd, nena.GapOverlap+xml

Parameter	Condition	Description
Agency	Mandatory	URI of Agency with gap/overlap. Will be repeated at least twice
Layer	Mandatory	Enumeration of layer where gap/overlap exists. May occur multiple times
Gap	Mandatory	Boolean, True if gap, false if overlap
Area	Mandatory	GML Polygon area of gap/overlap

Notifier Processing of SUBSCRIBE Requests

The Notifier consults the policy (NotifyPermissions) for GapOverlap to determine if the requester is permitted to subscribe; agencies allowed to provide authoritative data to the ECRF are permitted by default. If the requester is not permitted, the Notifier returns 603 Decline. Otherwise, the Notifier returns 202 Accepted.

Notifier Generation of NOTIFY Requests

When the provisioning GIS data creates a gap or overlap whose area is above the GapOverlapThreshold parameter, the Notifier generates a Notify to all subscribers. The Notifier repeats the Notification at least once per 24 hours as long as the gap/overlap remains.

Subscriber Processing of NOTIFY Requests: No specific action required.

Handling of Forked Requests: Forking is not expected to be used with this package.

Rate of Notification

Notifies normally only occur when the provisioning data changes. Throttle may be used to limit Notifications.

State Agents: No special handling is required.

5.3.6 Replicas

An ECRF is essentially a replica of a subset of the layers of one or more SIFs. The ECRF in turn, may provide a feed to other ECRFs who wish to maintain a copy of the data in an ECRF. As the ECRF is not the data owner, the source SIF must have a policy that permits the ECRF to do so, and the policy may restrict which entities the ECRF may provide replication data to. The ECRF also has a policy that defines who it will provide data to. If the ECRF provides a replica service, the interface

is the layer replication service as described in Section 4.7. In this case, the ECRF is the server side, as opposed to the client interface it must provide towards the SIF(s) it receives data from.

5.3.7 Provisioning

The ECRF is provisioned with

- a set of layers from one or more SIFs.
- the domains it may accept queries from, if its use is restricted.

To maximize the probability of getting help for any kind of emergency by foreign visitors who may have separate dial strings for different types of emergencies, the ECRF should be provisioned with every sos urn in the IANA registry¹⁷. All sos service URNs that represent services provided by the PSAP return the dial string ‘9-1-1’ and the PSAP URI. Other services available in the area would typically return a tel uri with the proper PSTN telephone number.

5.3.8 Roles and Responsibilities

The ECRF plays a critical role in the location-based routing of emergency calls. Therefore, it is crucial that the data in the ECRF be accurate and authorized. NENA therefore expects that 9-1-1 Authorities will be responsible for inputting the authoritative data for their jurisdiction in the ECRF. The data may be aggregated at a regional or state level, and the ECRF system provided at that level may be the responsibility of the associated state or regional emergency communications agency. In addition, replicas of the ECRF may be maintained by access or calling network operators. Thus the operation and maintenance of individual ECRFs may be the responsibility of the provider of the network in which they physically reside, but it is the 9-1-1 Authority that is responsible for maintaining the integrity of the source data housed within those systems. The 9-1-1 Authority will also provide input to the definition of the policy which dictates the granularity of the routing data returned by the ECRF (i.e., ESRP URIs vs. PSAP URIs), based on the identity of the query originator.

5.3.9 Operational Considerations

The NG9-1-1 architecture allows for a hierarchy of ESInets, with replicas of ECRFs at different levels of the hierarchy as well as in access/origination networks. It is expected that ECRFs that are provided as local copies to network operators will only have the layers necessary to route to the correct originating ESRP, whereas ECRFs that are inside the ESInet(s) will have all available layers and use authorization to control who has access to what information. Since it is not possible that all

¹⁷ While there is only one dialstring, 9-1-1, for emergencies in North America, all services in the sos tree should return a valid route when queried. For services the PSAP is responsible for, such as sos.police, the same URI used for urn:service:sos should be returned.

entities that need to access an ECRF will have one in their local domain, an ECRF for each 9-1-1 Authority must be accessible from the Internet¹⁸. Consideration needs to be given to the operational impacts of maintaining different levels of data in the various copies of the ECRF. In addition, tradeoffs between the aggregation of data in higher level ECRFs versus the use of Forest Guides to refer requests between ECRFs that possess different levels of ECRF data must be considered. Provisioning of data within appropriate ECRF systems for use in overload and backup routing scenarios must also be supported.

5.4 Location Validation Function

The NENA NG9-1-1 solution must properly route incoming IP packet-based emergency calls to the appropriate PSAP, as well as support the dispatch of responders to the right location. The location information used, when provided in civic form, must be proved sufficient for routing and dispatch prior to the call being placed. We refer to this as having a “valid” location for the call¹⁹. The i3 architecture defines a function called the LVF (Location Validation Function) for this purpose. The LVF is generally only used for civic location validation. Geo coordinate validation has some limited use, in extreme cases, including national boundary routing scenarios, over coastal waters, etc. The primary validation is accomplished as locations are placed in a LIS. Validation may also be done by an endpoint if it is manually configured with location, or if it retrieves location from the LIS (via a location configuration protocol [4]). Periodic re-validation of stored location is also recommended [59]²⁰. For fixed endpoints, location must be validated when the device is deployed, at each boot-up (power-cycle), and periodically, in order to reach the level of assurance required for acceptable route quality. For Nomadic devices, an LVF request must be invoked as in the fixed case, and in addition, whenever an end device changes its location. Mobile location differs in that it is expected to use only geo-coordinates (e.g., lat/lon), and therefore does not require the same level of LVF interaction and may not require any LVF interaction.

¹⁸ The Internet accessible ECRF may be a state or regional ECRF containing the local ECRF data of all 9-1-1 Authorities within the state or region

¹⁹ We note that RFC5222, which describes the LoST protocol used by the LVF validates against the service urn provided in the query, which for an outside (the ESInet) entity would be urn:service:sos. Strictly speaking, this is a call routing validation. NG9-1-1 requires validation for dispatch purposes. The LVF will validate to a level suitable for both routing and dispatch when the urn:service:sos is specified in the query.

²⁰ Short periods (days or a few weeks) allows errors that arise due to changes in underlying data the LVF uses to validate to show up sooner. However, the more often a LIS validates, the more load this places on the LIS and the LVF. A period of 30 days is recommended. LIS operators may wish to consult with the LVF operator to determine an optimal revalidation period.

5.4.1 Functional Description

The Location Validation Function (LVF) should be engineered to respond to LVF clients within a few seconds. The LVF data and interfaces are similar to those used by an ECRF representing the same geographic area(s). As a result, the LVF shares the same SIF data layer information as the ECRF, and reuses the same LoST protocol that is used by the ECRF, yet with a few additional data elements. The LVF supports an input query mechanism requiring civic location, a service URN, and a validation flag. This validation flag is an xml parameter setting, and is the main difference between a LoST query intended for an LVF and a LoST query used for routing, that is issued to an ECRF.

Messaging that is returned from an LVF contains all the same data as is returned from an ECRF query. In addition, an LVF validation query response also includes an indication of which data elements were found within the LVF itself. It's this address field matched data that enables the LVF client to determine if the civic location provided in the input is considered valid, and to what level of granularity.

Many other aspects of the LVF, its interfaces, and the data it contains are identical to the ECRF. Please refer to those sections for more detail.

5.4.2 Interface Description

The LVF supports two interfaces: a query/response interface, and a provisioning interface. Since the LVF is based on the LoST server architecture, the validation query/response interface is defined as the LoST protocol, per RFC5222 [61].

RFC5222 section 8.4.2 states that the inclusion of location validation is optional, and subject to local policy. NENA i3 requires that all LoST server implementations, deployed as an LVF, must support the inclusion of location validation information in the “findServiceResponse” message.

Local LVF policy is also responsible for determining which elements are given priority in determining which URI and which associated location data element tokens are deemed valid. Sometimes different data elements are in conflict with each other. As in the example message, the findServiceResponse message returns the Postal Code (value of 45054) as <invalid>, showing that the A1 & A3 (State & City) data elements in combination – in this case - are given preference over Postal Code that doesn't exist. Whereas the decision to prefer real data over non-existent data makes good sense, it is possible to have cases where all data elements are real, but not consistent with each other. In this case, NENA and local policy will determine which elements are used, and are shown as valid.

LVF interaction at emergency call time may be performed by a PSAP.

5.4.2.1 User Endpoint interaction

Any user endpoint (i.e., UE, device, handset, client application, etc.) that will perform a location validation directly, must implement the LVF (LoST) interface to be able to access an LVF. The endpoint must use the LVF interface with the same service URN as would be used for a routing query to the ECRF, viz "urn:service:sos", along with location information.

5.4.2.2 LIS Interaction

The LVF may receive a location validation request from the LIS in order to assure that the location information along with a particular service URN, used in the LVF query, will be deemed “valid”, that is, that there exists an appropriate route URI (e.g., PSAP URI) to match the query. The LVF must return the same URI that the ECRF would have returned (and subsequently will return at emergency call time), based on the same inputs used for the LVF.

5.4.2.3 Provisioning Interaction

The LVF requires the same type of data as required with the ECRF, and is expected to be provisioned through an xml provisioning interface either manually or via a machine-to-machine implementation. This includes synchronization between redundant and tiered LVF elements.

5.4.3 Interface Description

Currently, the LVF supports several interfaces, including the following:

- validation query interface
- validation response interface
- provisioning interface
- time interface
- logging interface
- SIF layer replication protocol, see section 4.7.

5.4.3.1 Validation query interface:

Examples taken from Figures 5 & 6 of RFC 5222.

Example of a validation request message:

```
<?xml version="1.0" encoding="UTF-8"?>
<findService
  xmlns="urn:ietf:params:xml:ns:lost1"
  recursive="true"
  validateLocation="true"
  serviceBoundary="value">
  <location id="627b8bf819d0bad4d" profile="civic">
    <civicAddress
      xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
      <country>US</country>
      <A1>OH</A1>
      <A3>Middletown</A3>
      <RD>Main</RD>
```

```
<STS>ST</STS>
<HNO>123</HNO>
<PC>45054</PC>
</civicAddress>
</location>
<service>urn:service:sos </service>
</findService>
```

5.4.3.2 Validation response interface

The LVF, for validation, only supports the “findServiceResponse” message. In the following example of a validation response message, note the bolded elements that indicate the validation:

```
<?xml version="1.0" encoding="UTF-8"?>
<findServiceResponse xmlns="urn:ietf:params:xml:ns:lost1">
  <mapping
    expires="2010-01-01T01:44:33Z"
    lastUpdated="2009-11-01T01:00:00Z"
    source="authoritative.example"
    sourceId="4db898df52b84edfa9b6445ea8a0328e">
    <displayName xml:lang="en">
      Middleton PSAP
    </displayName>
    <service>urn:service:sos</service>
    <serviceBoundary profile="civic">
      <civicAddress
        xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
        <country>US</country>
        <A1>Ohio</A1>
        <A3>Middelton</A3>
        <PC>45054</PC>
      </civicAddress>
    </serviceBoundary>
    <uri>sip:middleton-psap@example.com</uri>
```

```
<uri>xmpp:middleton-psap@example.com</uri>
<serviceNumber>911</serviceNumber>
</mapping>
<locationValidation>
  <valid>country A1 A3 A6 STS</valid>
  <invalid>PC</invalid>
  <unchecked>HNO</unchecked>
</locationValidation>
<path>
  <via source="resolver.example"/>
  <via source="authoritative.example"/>
</path>
<locationUsed id="627b8bf819d0bad4d"/>
</findServiceResponse>
```

The basis of a validation response is the inclusion of the data element, “validateLocation” being set to “true” in the validation query. In addition to the regular default inputs being returned, the validateLocation=true attribute setting will result in a response using the xml element “findServiceResponse” containing sub-element “locationValidation”, with attributes and tokens relating to which input elements were checked and shown as valid (or invalid).

The ECRF supports the <locationValidationUnavailable> warning element when an LVF server seeks to notify a client that it cannot fulfill a location validation request. This warning allows a server to return mapping information while signaling this exception state [RFC5222, sect. 13.2].

5.4.3.3 LVF Provisioning/synchronization

The LVF provisioning interface the same as that of the ECRF and uses the SIF Layer Replication protocol defined in Section 4.7

5.4.3.4 Alternative Address Interface

The ability to have alternative addresses returned, as supported within an i2 VDB, is currently out-of-scope for this document, and is left for future consideration.

5.4.3.5 Time Interface

The LVF must implement an NTP client interface in order to maintain current, accurate time-of-day information. The time of day information is an input to the LVF validation response information, as well as each transaction to the logging interface.

5.4.3.6 Logging Interface

The LVF must implement a logging interface per section 5.12.1.1. The LVF must be capable of logging every incoming validation request along with every recursive request and all response messages. In addition, the LVF must log all provisioning and synchronization messages and actions. In addition to the requirement for logging all the same data elements currently defined for logging by the ECRF, we have additional specific data logging requirements.

5.4.3.6.1 Validation query logging

The LVF logging mechanism must be capable of logging all input data elements for a validation query, including the specific input location and service URN. All logging transactions must be stored in the form of transaction detail records, and must be made external when warranted by implementation policy. The data elements logged include the following:

- Date & Time of transaction
- Request message type
- Type of location received
- Location elements received
- Service URN received.

5.4.3.6.2 Validation response logging

The LVF logging mechanism must be capable of logging all output data elements provided in the validation response message, including the validation response status of each location element. All logging transactions must be stored in the form of transaction detail records, and must be made external when warranted by implementation policy. The data elements logged include the following:

- Date & Time of transaction
- Response message type
- Validation attributes
- Location element tokens
- “Error Code” values.

5.4.3.6.3 Provisioning/Synchronization logging

The LVF logging mechanism must be capable of logging all provisioning input and output messages from an individual provisioning client or another LVF. All logging transactions must be stored in the form of transaction detail records, and must be made external when warranted by implementation policy. The data elements logged include the following:

- Date & Time of transaction
- Transaction type (e.g., Add, Delete, Modify)
- Record information
- Response acknowledgement.

5.4.4 Data Structures

The data structures for the LVF include those defined for the ECRF. In addition to those used for the ECRF, the following LVF specific data structures are included:

Table 5-1 LVF Specific Location Data Elements

Label	Description	Type	Example
validateLocation	Xml attribute for findService elementvalidation (see notes 1 & 2)	Boolean	true
locationValidation	Xml attribute for findServiceResponse element	n/a (see note 3)	n/a
valid	Xml attribute to list those input element tokens that were successfully validated	n/a (see note 3)	A1
invalid	xml attribute to list those input element tokens that were unsuccessfully validated	n/a (see note 3)	RD
unchecked	Xml attribute to list those input element tokens that were not checked for validation (see note 3)	n/a (see notes 3 & 4)	HNO

Note 1. If the validateLocation is not included, it is treated as “false”.

Note 2. The attribute is ignored if the input contains a geodetic form of location.

Note 3. RFC5222 states only that the presence of each element token is optional, subject to local policy.

Note 4. Any input element tokens not included in the locationValidation response, belong to the “unchecked” category.

5.4.5 Roles and Responsibilities

PSAPs are directly responsible for LVF data, though a PSAP may contract data maintenance over to a third-party if they choose to. The LVF provisioning interface is the SIF layer replication protocol defined in Section 4.7.

The ECRF and the LVF are provisioned, directly or indirectly, from an authoritative SIF, using the layer replication protocol. A change in the SIF will be propagated to any ECRFs and LVFs

connected to that SIF system. Thus the ECRF and LVF do not have to be provided by, or operated by the same entity, although it will be common for them to be so connected. Indeed, it may be common for the ECRF and LVF to be collocated in the same box.

5.4.6 Operational Considerations

The placement of LVF elements in the IP-enabled network varies with implementation. Since both end devices as well as LIS elements need to validate location, it is recommended that LVF elements are within the local domain or adjacent to it. Given that NG9-1-1 elements will also need to validate civic locations that either come with an emergency call, or are conveyed over the voice path, it is also a requirement that LVF elements are reachable from within any ESInet. Finally, since it is not possible that all entities that need to access a LVF will have one in their local domain, a LVF must be accessible from the Internet²¹.

LVF elements are based on the LoST server architecture and use the LoST protocol [61]. The LVF is a logical function that may share the physical platform of an ECRF, and must share the same data for a given jurisdiction as the ECRF. The justification for shared data is rooted in the idea of consistency – expecting a similar result from the same, or matching data. The LVF is used during a provisioning process (loading data into a LIS for example), while an ECRF is in the real time call flow. Separating the functions may make more sense. The Service Level Agreements for the two functions may dictate whether they can be combined or not.

An LVF, wherever deployed, whether within an Access network, or in some other type of Origination network, needs to be able to reach out to other LVFs in case of missing data, or in the case where the requested location is outside its local jurisdiction. If the LVF doesn't know the answer, based on configuration, it will either recurse (refer) a request for validation to one or more other LVFs, or it will iterate the request to some other LVF, providing the other LVF's URL in the original LVF response.

Redundant LVF elements are recommended, similar to DNS server deployments (the LVF shares some of the same replication characteristics with DNS), by example, in order to maintain a high level of availability and transaction performance.

As with the ECRF, and given the close association between the LVF and ECRF elements, LVFs should be deployed hierarchically and with “n” number of replicas at each level of the hierarchy. The same redundancy/replica considerations apply to access/calling/origination networks that use an LVF. This level of redundancy aids in maintaining high levels of availability during unexpected system outages, scheduled maintenance windows, data backup intervals, etc.

²¹ The Internet accessible LVF may be a state or regional LVF containing the local LVF data of all PSAPs within the state or region

Similar to ECRF deployments, localized LVF elements may have limited data, sufficient to provide location validation within its defined boundaries, but must rely on other LVFs for validation of a location outside its local area.

LVFs within the ESInet will likely have considerably more data than those LVFs in origination networks, providing aggregation for many local access areas as well as PSAP jurisdictions. Even the level of data that an LVF might contain will vary depending on the hierarchy of the ESInet that it supports. An ESInet serving a local PSAP may have within its LVF, only base civic location data for its described jurisdiction, whereas a State-level or County-level LVF may aggregate all of the local PSAP data within that level of hierarchy.

5.5 Spatial Information Function

The Spatial Information Function (SIF) is the base database for NG9-1-1. Nearly all location related data is ultimately derived from the SIF. If a datum is somehow associated with location, the base data will reside in the SIF. The SIF supplies data for:

1. The ECRF/LVF
2. Map views for alternate PSAPs.

The SIF is a specialized form of a Geospatial Information System, and may be implemented on a conventional GIS with the appropriate interfaces. The SIF itself is not standardized in i3. What is standardized is a method of replicating layers from the master SIF to external databases. The ECRF and LVF provisioning interfaces use this mechanism. When calls are answered at an alternate PSAP, map views are generated from off-site replicas of layers in the SIF system, which are maintained by this interface.

5.5.1 Layers

In order to be useful, i3 standardizes certain layers in the SIF system so that interchange between SIF systems is practical. Appendix B defines the layers and the required attributes those layers must implement. The NG9-1-1 system is dependent on all SIF systems having common definitions for these layers. All attributes listed in Appendix B must be implemented as specified. The layers defined include:

- Layers with polygon features
 - State (PIDF A1)
 - County (PIDF A2)
 - Municipality (PIDF A3)
 - Division (PIDF A4)
 - Sub-Division (PIDF A5)
 - Parcels (Can be PIDF HNO and components)
 - Sub-Parcels (Can be PIDF HNO and components)
 - PSAP Service Boundary
 - Responding Agency Services Boundary – Law Enforcement, EMS, Fire, Highway Patrol, etc...
- Layers with line features

- Road Centerlines (PIDF RD and components)
- Layers with point features
 - Site / Structure Locations (address points) (PIDF HNO and components)

5.5.2 MSAG Conversion Service (MCS)

The MSAG Conversion Service provides a convenient way to provide data to, or get data from, an un-upgraded system that still uses MSAG data. The service provides conversion between PIDF-LO and MSAG data. Two functions are defined:

PIDFLOtoMSAG: which takes a PIDF-LO as described in RFC4119 updated by RFC5139 and RFC5491 and returns an MSAG address as an XML object conforming to NENA 02-010 Version 4, XML Format for Data Exchange

MSAGtoPIDFLO: which takes an MSAG address as an XML object conforming to NENA 02-010 Version 4, XML Format for Data Exchange and returns a PIDF-LO as described in RFC4119 updated by RFC5139 and RFC5491

MSAG Conversion Service is provisioned using the same mechanism as is used to provision the ECRF and LVF: layer replication from the master SIF. The layers include all of the layers to create a PIDF as described above, plus any layers needed to construct the MSAG for the local jurisdiction. These would typically include an MSAG Community Name, often includes the County ID, and for many jurisdictions where prefix/suffix and/or directionals are included in the Street Name would include a Street Name layer. Where the content of the MSAG is the same (for all addresses in the jurisdiction) as the equivalent PIDF-LO field, the layer need not be present.

MCS uses a forest guide referral mechanism identical to the ECRF. If the input address is not within the service boundary of the local MCS, it can consult a forest guide to refer the query to the appropriate MCS.

The PIDFLOtoMSAG function locates the point in the database represented by the input PIDF-LO and retrieves the MSAG layers associated with that point. It constructs an MSAG address using any MSAG layers available, and the PIDF-LO layers where MSAG and PIDF-LO are the same. The functions return Version 4 XML data exchange, but the client can convert to any other MSAG version from the XML representation.

PIDFtoMSAGRequest

Parameter	Condition	Description
pidflo	Mandatory	PIDF-LO to be converted

PIDFtoMSAGResponse

Parameter	Condition	Description
msag	Conditional	MSAG resulting from conversion

referral	Conditional	URI of another MCS
errorCode	Mandatory	Error response, see below

Either msag or referral must be present in the response

Error Codes

100 Okay No error

508 NoAddressFound: the input appears to be within the service boundary of the MCS, but no point matching the input was located

509 Unknown MCS: the input is not in the service boundary of the MCS and the local MCS could not locate an MCS who served that location.

504 Unspecified Error

The MSAGtoPIDFLO function works in the same manner, locating the point in the database the MSAG address refers to, and composing a PIDF-LO from the PIDF-LO layers.

MSAGtoPIDFRequest

Parameter	Condition	Description
msag	Mandatory	msag to be converted

MSAGtoPIDFResponse

Parameter	Condition	Description
pidflo	Conditional	PIDF-LO resulting from conversion
referral	Conditional	URI of another MCS
errorCode	Mandatory	Error response, see below

Either pidf or referral must be present in the response

Error Codes

100 Okay No error (optional to return)

508 NoAddressFound: the input appears to be within the service boundary of the MCS, but no point matching the input was located

509 Unknown MCS: the input is not in the service boundary of the MCS and the local MCS could not locate an MCS who served that location.

504 Unspecified Error

The service logs the invocation of the function, as well as the input and output objects.

5.5.3 Geocode Service (GCS)

The Geocode service provides geocoding and reverse geocoding. Two functions are defined:

Geocode: which takes a PIDF-LO as described in RFC4119 updated by RFC5139 and RFC5491 containing a civic address and returns a PIDF-LO containing a geo for the same location.

ReverseGeocode: which takes a PIDF-LO as described in RFC4119 updated by RFC5139 and RFC5491 containing a geo and returns a PIDF-LO containing a civic address for the same location.

The Geocode Service is provisioned using the same mechanism as is used to provision the ECRF and LVF: layer replication from the master SIF. The layers include all of the layers to create a PIDF-LO as described above.

Any conversion, and specifically geocoding and reverse geocoding can introduce errors. Unless the underlying SIF has very accurate polygons to represent all civic locations precisely, the conversion is complicated by the inherent uncertainty of the measurements and the “nearest” point algorithm employed. Users of these transformation services should be aware of the limitations of the geocoding and reverse geocoding mechanisms. Reverse geocode is typically less accurate than geocoding, although some error, and unquantified uncertainty is inherent in both.

The GCS uses a forest guide referral mechanism identical to the ECRF. If the input address is not within the service boundary of the local GCS, it can consult a forest guide to refer the query to the appropriate GCS.

The Geocode function locates the point in the database represented by the input PIDF-LO and retrieves the geo associated with that location. It constructs a PIDF-LO with the geo. If the PIDF-LO in the request contains more than one location, the return must contain only one result, which is the conversion of the first location in the PIDF.

GeocodeRequest

Parameter	Condition	Description
pidflo	Mandatory	PIDF-LO with civic to be converted

GeocodeResponse

Parameter	Condition	Description
pidflo	Conditional	PIDF-LO resulting from conversion
referral	Conditional	URI of another GCS
errorCode	Mandatory	Error response, see below

Either pidf or referral must be present in the response

Error Codes

100 Okay No error

508 NoAddressFound: the input appears to be within the service boundary of the GCS, but no point matching the input was located

509 Unknown MCS: the input is not in the service boundary of the GCS and the local GCS could not locate a GCS who served that location.

504 Unspecified Error

The ReverseGeocode function works in the same manner, locating the location in the database the input geo refers to, and composing a PIDF-LO from the PIDF-LO layers.

ReverseGeocodeRequest

Parameter	Condition	Description
pidflo	Mandatory	PIDF-LO with geo to be converted

ReverseGeocodeResponse

Parameter	Condition	Description
pidflo	Conditional	PIDF-LO resulting from conversion
referral	Conditional	URI of another GCS
errorCode	Mandatory	Error response, see below

Either pidflo or referral must be present in the response

Error Codes

100 Okay No error

508 NoAddressFound: the input appears to be within the service boundary of the GCS, but no point matching the input was located

509 Unknown MCS: the input is not in the service boundary of the GCS and the local GCS could not locate a GCS who served that location.

504 Unspecified Error

The service logs the invocation of the function, as well as the input and output objects.

Note: The IETF geopriv working group is considering the definition of a geocoding protocol/service. If such a standardization effort is undertaken, and if the resulting work is suitable, it will replace this NENA-only interface in a future edition of this document.

5.5.4 Operational Considerations

The SIF is not used directly in call processing, although its data is critical to achieving proper routing. For that reason, a single SIF system, with frequent backup operations is sufficient. However, since calls may be answered by other PSAPs, and the originally intended PSAP may be

unavailable, copies of the layers sufficient for display should be made available, using the layer replication mechanism defined in Section 4.7.

5.6 PSAP

A PSAP provides the following interfaces towards the ESInet

5.6.1 SIP Call interface

The PSAP must deploy the SIP call interface as defined in Section 4.1 including the multimedia capability, and the non-human-initiated call (emergency event) capability. PSAPs must recognize calls to their administrative numbers received from the ESInet (and distinguishable from normal 9-1-1 calls by the presence of the number in a sip or tel URI in the To: field and the absence of the sos service URN in a Route header). The SIP call interface may also be used to place non 9-1-1 calls (including call backs) from the PSAP using normal SIP Trunking mechanisms as specified in sipConnect V1.0 [108].

Note: while all i3 PSAPs must handle all media, a legacy PSAP behind an LPG would only handle voice media and TTY. There is no mechanism by which a caller could discover what media the PSAP supports. This will be covered in a future edition of this document.

5.6.2 LoST interface

The PSAP must provide a LoST client interface as defined in Section 4.5. The PSAP uses the ECRF and LVF to handle calls that must be dispatched and calls that must be transferred based on the actual location of the incident. The ECRF and LVF use the LoST interface.

5.6.3 LIS Interfaces

The PSAP must implement both SIP Presence Event Package and HELD dereference interfaces to any LIS function as described in Section 5.9. When the PSAP receives a location reference (in a Geolocation header on the upstream SIP interface) it uses the LIS dereference interface to obtain a location value. The PSAP must be provisioned with credentials for every LIS in its service area²². The PSAP must use TCP with either TLS or IPsec for the LIS Dereference Interface, with fallback to TCP (without TLS) on failure to establish a TLS connection when TLS is used. The PSAP should maintain persistent TCP (and TLS where used) connections to LISs that it has frequent transactions with. A suggested value for "frequent" is more than one transaction per day.

²² This document specifies that the LIS accept credentials issued to the PSAP traceable to the PCA. Notwithstanding that requirement, ESInet elements needing location, including PSAPs, must be able to be provisioned with credentials acceptable to LIS's that do not accept the PCA credential.

For HELD location URIs, specifying `responseTime = emergencyDispatch` will result in a location meeting regulated accuracy requirements. If the PSAP wishes an immediate location, it can specify a short `responseTime` (perhaps 250 ms), and get the best location quality available in that time. Location updates for location URIs using HELD may be obtained by repeating the dereference.

PSAPs receiving SIP location URIs should subscribe to the Presence event per RFC 3856 [31]. The PSAP receives an immediate location report, which may reflect the best available location at the time of the subscription. A subsequent location update is sent when more accurate location is available. By setting the expiration time of the subscription, the PSAP is able to control what updates it receives. PSAPs that wish to track the motion of a caller could use the location filter and event rate control mechanisms in `loc-filters` [103] and `rate-control` [113] to control updates.

Note that because the PSAP will not have an identity of an arbitrary device with which it could query a LIS to get the device's location, the “manual query” function in an E9-1-1 ALI has no equivalence in NG9-1-1.

5.6.4 Bridge Interface

A PSAP may deploy a bridge (as described in Section 5.7) inside the PSAP, in which case it must provide the bridge controller interfaces. PSAPs must be able to accept calls from, and utilize the features of outside bridges.

5.6.5 ElementState

The PSAP must deploy an `ElementState` notifier as described in Section 3.3.2. Note that the terminating ESRP may route to a (queue of) call taker(s). Each call taker should implement an element state notifier.

5.6.6 SIF

The PSAP may provide²³ a GIS server interface as described in Section 5.5 for the ECRF, GIS Replica, and other interfaces. The PSAP may provide the MSAG conversion service (server side) or may use an ESInet service (client side).

5.6.7 Logging Service

The PSAP may deploy a logging service (as described in Section 5.12) inside the PSAP, in which case it must provide the logging service retrieval functions. A PSAP may use a logging service in the ESInet, in which case it must deploy the logging service insert functions.

²³ The GIS system may be provided by a 9-1-1 Authority

5.6.8 Security Posture

The PSAP must provide a Security Posture notifier as described in Section 3.3.1.

5.6.9 Policy

The PSAP may provide a policy store as described in Section 4.4.1, in which case it must implement the server side of the policy retrieval functions, and may provide the server side of the policy storage function. The PSAP may provide a Policy Editor, in which case it must deploy the client side of the policy retrieval and storage functions. If the PSAP uses a policy store outside the PSAP to control functions inside the PSAP, it must deploy the client side of the policy retrieval functions.

PSAPs must provide a Termination-Policy for the queue(s) its calls are sent to.

PSAPs must provide a takeCallsOnQueues policy to determine which queues the PSAP will dequeue from (that is, which queues it will subscribe to the dequeueRegistration and queueState events for).

5.6.10 Additional Data dereference

The PSAP must deploy a dereference (HTTP Get) interface for additional data as described in Section 8.

5.6.11 Time Interface

The PSAP must implement an NTP client interface for time-of-day information. The PSAP may also provide an interface to a hardware clock.

5.6.12 Test Call

The PSAP may deploy the test call function as described in Section 11.

5.6.13 Call Diversion

A PSAP may be overloaded and be unable to answer every call by a call taker. Overload is determined by exceeding the size of the primary queue that its calls are sent to. Routing rules for the PSAP would then cause calls to receive an alternate call treatment:

- Calls can be sent a “Busy” indication
- Calls can be diverted to an Interactive Multimedia Response unit
- Calls can be diverted to one or more alternate PSAPs.

The latter is mechanized by sending the call to queues which other PSAPS dequeue from. Since the diverted-to PSAP(s) have to explicitly register to dequeue (DequeueRegistration, see Section 5.2.1.2), no calls can be sent to a PSAP that hasn’t explicitly asked for them.

PSAPs that agree to take calls from other PSAPs may require explicit management approval at the time the calls are sent. Effectively, such PSAPs are agreeing to take calls on a standby basis only, and explicit management action is required before the calls will actually be accepted.

To accomplish this, the diverted-to PSAP subscribes to the DequeueRegistration event of the diverted-from PSAP with the “Standby” parameter set to “true”. The diverted-to PSAP also

subscribes to the queueState event for the diversion queue. It may specify a filter that limits notifications to those setting queueState to “DiversionRequested”. When the queueState event notification occurs with “DiversionRequested” state, the diverted-to PSAP management would be alerted. If it agrees to accept calls, it would resubscribe to the DequeueRegistration event with Standby set to “false”, and calls would subsequently be sent to it. When the diverted-to PSAP determines that its services are no longer needed, it can reinstate the <standby>true</standby>.

5.6.14 Incidents

A new call arrives with a new Incident Tracking Identifier assigned by the first ESRP in the ESInet. The ESRP assumes each call is a new Incident. The call taker may determine that the call is actually part of another Incident, usually reported in a prior call. The PSAP must merge the IncidentTrackingID assigned by the ESRP with the actual IncidentTrackingID. It does so with the MergeIncident log record. The actual IncidentTrackingID would be part of the AdditionalPSAPData object passed to a secondary PSAP or responder and part of the INVITE if the call is transferred. When the PSAP completes processing of an Incident, it logs a ClearIncident record.

5.7 Bridging

Bridging is used in NG9-1-1 to transfer calls and conduct conferences. Bridges have a (SIP) signaling interface to create and maintain conferences and media mixing capability. Bridges must be multimedia (voice, video, text). A bridge is necessary to transfer a call because IP-based devices normally cannot mix media, and transferring always adds the new party (for example, a call taker at a secondary PSAP) to the call before the transferor (for example, the original call taker at the PSAP which initially answered the call) drops off the call. The rough transfer sequence, based on the procedures defined in RFC4579 [51], is:

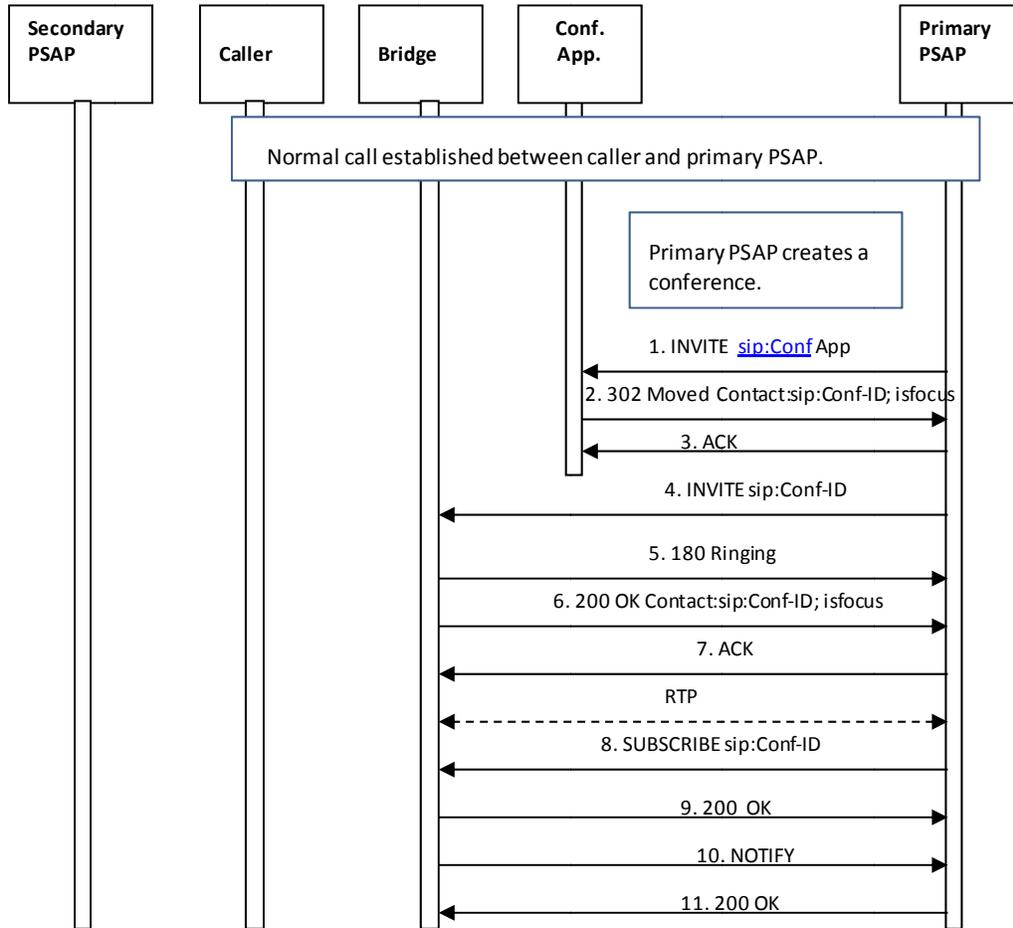
1. PSAP creates a conference on the bridge
2. PSAP REFERS the caller to the bridge
3. PSAP tears down the original PSAP-Caller leg
4. PSAP REFERS transfer target (secondary PSAP for example) to the conference
5. PSAP tears down its leg to the conference, the secondary PSAP and the caller remain
6. Secondary PSAP REFERS the caller to it
7. Secondary PSAP terminates the conference.

5.7.1 Bridge Call Flow

Conferencing procedures are documented in RFC4579. This document includes definition of an Event package that allows conference participants to manage the conference. In the message sequences below, all participants are conference aware (that is, they implement the event package). It is not necessary for the caller to be conference aware, and if it were not, its SUBSCRIBE to the conference package would not occur. It is required that the caller, or some element in the path, implement the Replaces header, see Section 5.8

5.7.1.1 Creation of a Conference Using SIP Ad-Hoc Methods

This scenario described in the call flow depicted below follows Section 5.4 of RFC4579.



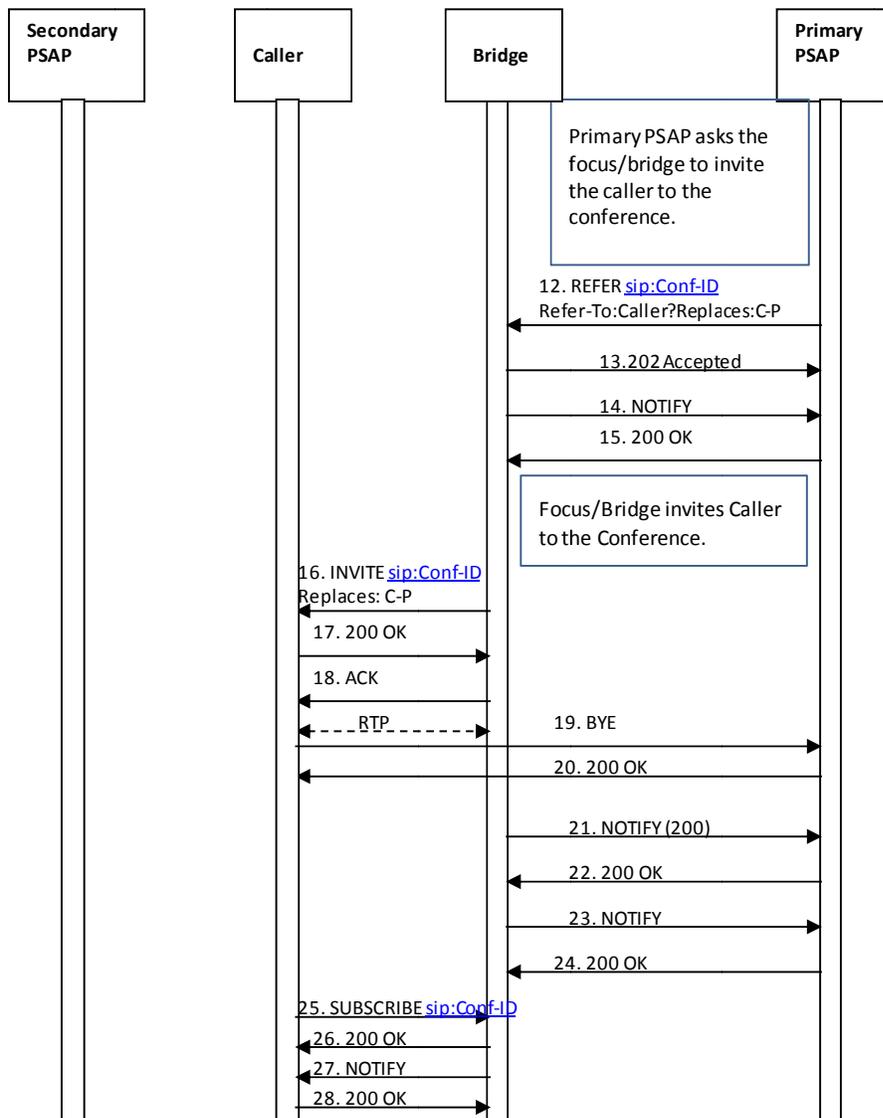
1. The Primary PSAP creates a conference by first sending an INVITE to a conference application, using a URI that is known by/provisioned at the Primary PSAP.
2. The Conference Application responds by sending a 302 Moved message which redirects the Primary PSAP to the conference bridge, and provides the Conference-ID that should be used for the conference.
3. The Primary PSAP acknowledges the receipt of the 302 Moved message.
4. The Primary PSAP generates an INVITE to establish a session with the conference bridge.²⁴

²⁴ Note that, based on RFC 4579, the messages sent in Steps 2, 3 and 4 are optional and may not be exchanged if the conference application and the media server are the same.

5. The conference bridge responds to the INVITE by returning a 180 Ringing message.
6. The conference bridge then returns a 200 OK message, and a media session is established between the Primary PSAP and the conference bridge.
7. The Primary PSAP returns an ACK message in response to the 200 OK.
8. through 11. Once the media session is established, the Primary PSAP subscribes to the conference associated with the URI obtained from the Contact header provided in the 200 OK message from the conference bridge.

5.7.1.2 Primary PSAP Asks Bridge to Invite the Caller to the Conference

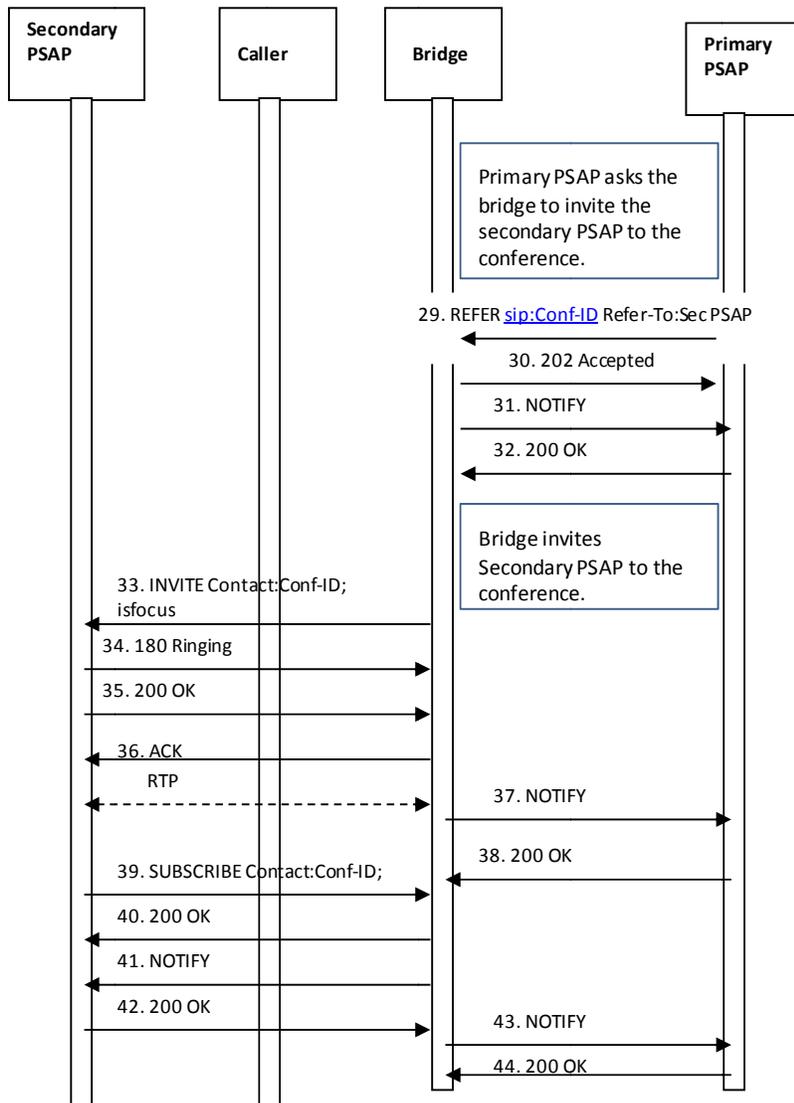
This flow is based on Section 5.10 of RFC4579.



12. After the Primary PSAP establishes the conference, it sends a REFER method to the conference bridge asking it to invite the caller to the conference. The REFER method contains an escaped Replaces header field in the URI included in the Refer-To header field.
13. The bridge returns a 202 Accepted message to the Primary PSAP.
14. The bridge then returns a NOTIFY message, indicating the subscription state of the REFER request (i.e., active).
15. The Primary PSAP returns a 200 OK in response to the NOTIFY message.
16. The bridge invites the caller to the conference by sending an INVITE method containing the Conf-ID and a Replaces header that references the leg between the caller and the Primary PSAP.
17. The caller accepts the invitation by returning a 200 OK message.
18. The bridge acknowledges receipt of the 200 OK message by returning an ACK.
A media session is established between the caller and the bridge.
19. The caller releases the connection to the Primary PSAP by sending a BYE message.
20. The Primary PSAP responds by returning a 200 OK message.
21. The bridge sends a NOTIFY message to the Primary PSAP to provide REFER processing status.
22. The Primary PSAP responds by returning a 200 OK message.
23. The bridge sends a NOTIFY message to the Primary PSAP to provide updated status associated with the conference state.
24. The Primary PSAP responds by returning a 200 OK message.
25. The caller subscribes to the conference associated with the Conference ID provided in the INVITE message from the bridge by sending a SUBSCRIBE message to the bridge. (Optional)
26. The bridge acknowledges the subscription request by sending a 200 OK message back to the caller. (Optional)
27. The bridge then returns a NOTIFY message to the caller to provide subscription status information. (Optional)
28. The caller responds by returning a 200 OK message. (Optional)

5.7.1.3 Secondary PSAP is Invited to the Conference

This flow is based on Section 5.5 of RFC4579.

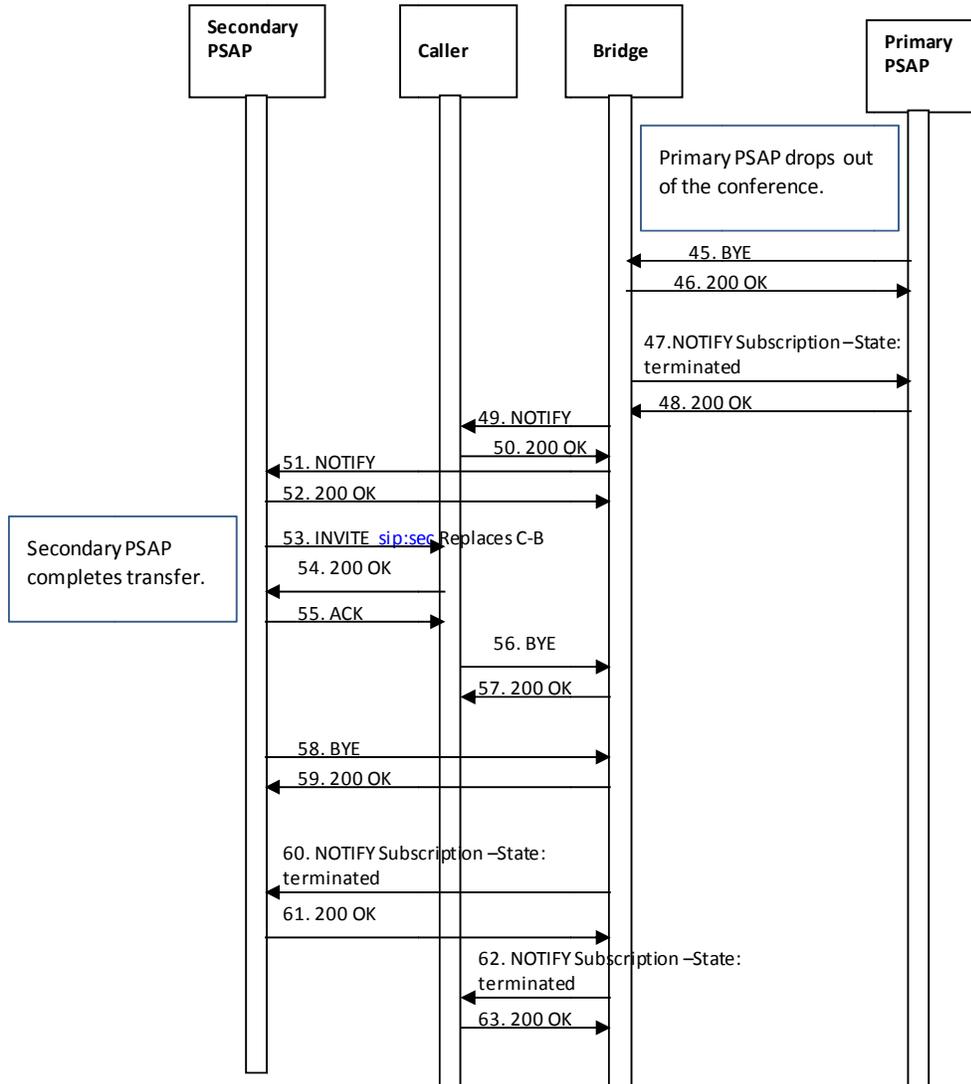


29. The Primary PSAP sends a REFER method to the conference bridge asking it to invite the Secondary PSAP to the conference. The REFER method contains the Conf-ID and a Refer-To header that contains the URI of the Secondary PSAP. The REFER method also contains an escaped Call-Info header field containing a reference URI that points to the “Additional Data Associated with a PSAP” data structure.
30. The bridge returns a 202 Accepted message to the Primary PSAP.
31. The bridge then returns a NOTIFY message, indicating that subscription state of the REFER request (i.e., active).
32. The Primary PSAP returns a 200 OK in response to the NOTIFY message.
33. The bridge invites the Secondary PSAP to the conference by sending an INVITE method containing the Conf-ID and Contact header that contains the conference URI and the isfocus feature parameter. The INVITE contains the Call-Info header field containing a reference URI that points to the “Additional Data Associated with a PSAP” data structure.

34. The Secondary PSAP UA responds by returning a 180 Ringing message to the bridge.
 35. The Secondary PSAP accepts the invitation by returning a 200 OK message.
 36. The bridge acknowledges receipt of the 200 OK message by returning an ACK.
- A media session is established between the Secondary PSAP and the bridge.*
37. The bridge returns a NOTIFY message to the Primary PSAP to provide updated status of the subscription associated with the REFER request.
 38. The Primary PSAP responds to the NOTIFY message by returning a 200 OK message.
 39. The Secondary PSAP subscribes to the conference associated with the Conf- ID provided in the INVITE message from the bridge by sending a SUBSCRIBE message to the bridge.
 40. The bridge acknowledges the subscription request by sending a 200 OK message back to the Secondary PSAP.
 41. The bridge then returns a NOTIFY message to the Secondary PSAP to provide subscription status information.
 42. The Secondary PSAP responds by returning a 200 OK message.
 43. The bridge sends a NOTIFY message to the Primary PSAP providing updated status for the subscription associated with the REFER request.
 44. The Primary PSAP responds to the NOTIFY message by returning a 200 OK message.

At this point the caller, Primary PSAP, and Secondary PSAP are all participants in the conference.

5.7.1.4 Primary PSAP Drops Out of Conference; Secondary PSAP Completes Transfer



45. Upon determining that the emergency call transfer should be completed, the Primary PSAP disconnects from the call by sending a BYE message to the bridge.
46. The conference bridge responds by returning a 200 OK message.
47. The bridge then returns a NOTIFY message indicating that the subscription to the conference has been terminated.
48. The Primary PSAP returns a 200 OK in response to the NOTIFY.
49. The bridge then returns a NOTIFY message to the caller indicating that there has been a change to the subscription state. (Optional)
50. The caller returns a 200 OK in response to the NOTIFY. (Optional)
51. The bridge returns a NOTIFY message to the Secondary PSAP indicating that there has been a change to the subscription state.
52. The Secondary PSAP returns a 200 OK in response to the NOTIFY.

53. Upon recognizing that the caller and the Secondary PSAP are the only remaining participants in the conference, the Secondary PSAP completes the transfer by sending an INVITE to the caller requesting that they replace their connection to the bridge with a direct connection to the secondary PSAP. The secondary PSAP learns the URI of the caller through the “Additional Data Associated with a PSAP” data structure
 54. The caller responds by returning a 200 OK message to the Secondary PSAP.
 55. The Secondary PSAP returns an ACK in response to the 200 OK.
 56. The caller then sends a BYE to the bridge to terminate the session.
 57. The bridge responds by sending the caller a 200 OK message.
 58. The Secondary PSAP also terminates its session with the bridge by sending a BYE message to the bridge.
 59. The bridge responds by sending a 200 OK message to the Secondary PSAP.
 60. The bridge then returns a NOTIFY message to the Secondary PSAP indicating that the subscription to the conference has been terminated.
 61. The Secondary PSAP returns a 200 OK in response to the NOTIFY message.
 62. The bridge sends a NOTIFY message to the caller indicating that the subscription to the conference has been terminated. (Optional)
 63. The caller responds with a 200 OK message. (Optional)
- At this point, the transfer is complete, and the caller and the Secondary PSAP are involved in a two-way call.*

5.7.2 Passing data to Agencies via bridging

When another PSAP is bridged to a 9-1-1 call there are separate “legs” for each participant in the bridge. The 9-1-1 call itself terminates at the bridge, with the call taker and the transfer target having separate legs into the bridge. When the transfer target receives the initial SIP transaction it is an INVITE from the bridge to a conference. It is critical that the transfer target receive (or have access to) the location of the original caller, as well as any “Additional Data” that the primary PSAP call taker may have accessed or generated during the processing of the emergency call. Caller location information received by the primary PSAP in the Geolocation header of the INVITE message, along with any additional data that the primary PSAP call taker may have obtained when the call was delivered (i.e., “Additional Data Associated with a Call” and/or “Additional Data Associated with a Caller”) or that was generated by the call taker as a result of processing the incoming emergency call, should be populated in the “Additional Data Associated with a PSAP” structure. (See Section 8 for further discussion of Additional Data structures.) When an emergency call is transferred, the primary PSAP will request that the bridge insert by embedded header, a Call-Info header with a URI that points to the “Additional Data Associated with a PSAP” data structure in the REFER method sent to the bridge. The bridge must subsequently include this Call-Info header in the INVITE it sends to the transfer target.

5.8 Transfer Involving Calling Devices that Do Not Support Replaces

As discussed in Section 5.7 of NENA 08-002, there is a problem that some devices that could originate 9-1-1 calls do not support the Replaces header. If a PSAP needs to transfer a call originated by such a device, it cannot use the standardized SIP signaling to the caller as described above. Section 5.7 of NENA 08-002 describes three solutions to this problem.

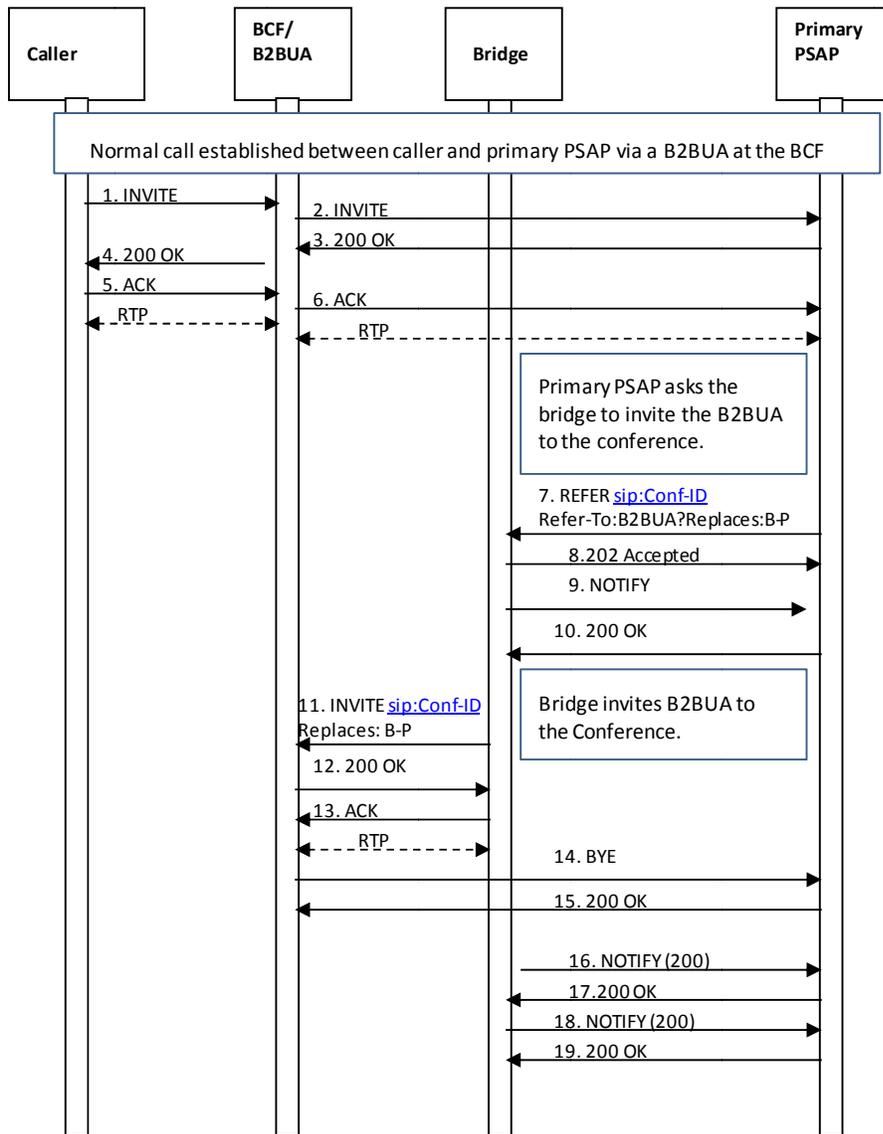
Each of these solutions is specified in more detail in the sections below.

5.8.1 B2BUA in the Border Control Function

When this solution is implemented, the BCF must include a B2BUA function as described in RFC3261. All calls are relayed through the B2BUA. The B2BUA is transparent to signaling with the following exceptions:

1. Media endpoints towards both the caller and the PSAP are rewritten to be contained within the BCF
2. The REFER method, when executed on the PSAP side to a conference bridge, causes the bridge to invite the B2BUA to the conference, and the B2BUA to respond as illustrated below. The leg between the caller and the B2BUA sees no transaction.
3. If the BCF receives an INVITE from a caller that does not include a Supported header containing the replaces option-tag it must include a Supported header containing the replaces option-tag in the INVITE forwarded to the ESInet and provide the functionality described in this section.

Note that the following flow assumes that the Primary PSAP has already created a conference using SIP Ad Hoc methods, as described in Section 5.7.1.1.



1. The caller initiates an emergency session request by sending an INVITE message to the B2BUA. The INVITE contains a Geolocation header with caller location information.
2. The B2BUA sends a corresponding INVITE message via the i3 ESInet toward the Primary PSAP. (Elements and signaling involved in routing the emergency call within the i3 ESInet are not shown in this flow.) The INVITE would contain a Supported header indicating support for Replaces.
3. The Primary PSAP responds by returning a 200 OK message to the B2BUA.
4. The B2BUA responds to the receipt of the 200 OK from the Primary PSAP by sending a 200 OK message to the caller's device.

5. The caller's device responds by sending an ACK to the B2BUA.

A media session is established between the caller and the B2BUA. Depending on the design of the ESInet, the B2BUA may cross connect media from the caller to the Primary PSAP

6. The B2BUA sends an ACK to the Primary PSAP in response to receiving an ACK from the caller's device.

A media session is established between the B2BUA and the Primary PSAP.

7. The Primary PSAP sends a REFER method to the conference bridge asking it to invite the B2BUA to the conference. The REFER method contains an escaped Replaces header field in the URI included in the Refer-To header field.

8. The bridge returns a 202 Accepted message to the Primary PSAP.

9. The bridge then returns a NOTIFY message, indicating that subscription state of the REFER request (i.e., active).

10. The Primary PSAP returns a 200 OK in response to the NOTIFY message.

11. The bridge invites the B2BUA to the conference by sending an INVITE method containing the Conf-ID and a Replaces header that references the leg between the B2BUA and the Primary PSAP.

12. The B2BUA accepts the invitation by returning a 200 OK message.

13. The bridge acknowledges receipt of the 200 OK message by returning an ACK.

A media session is established between the B2BUA and the bridge. Note that the media session between the B2BUA and the Primary PSAP still exists at this time. Note also that the media session between the caller and the B2BUA is undisturbed. As above, the B2BUA may cross connect media from the caller to the bridge

14. The B2BUA releases the connection to the Primary PSAP by sending a BYE message.

15. The Primary PSAP responds by returning a 200 OK message.

At this point, the media session between the B2BUA and the Primary PSAP is torn down.

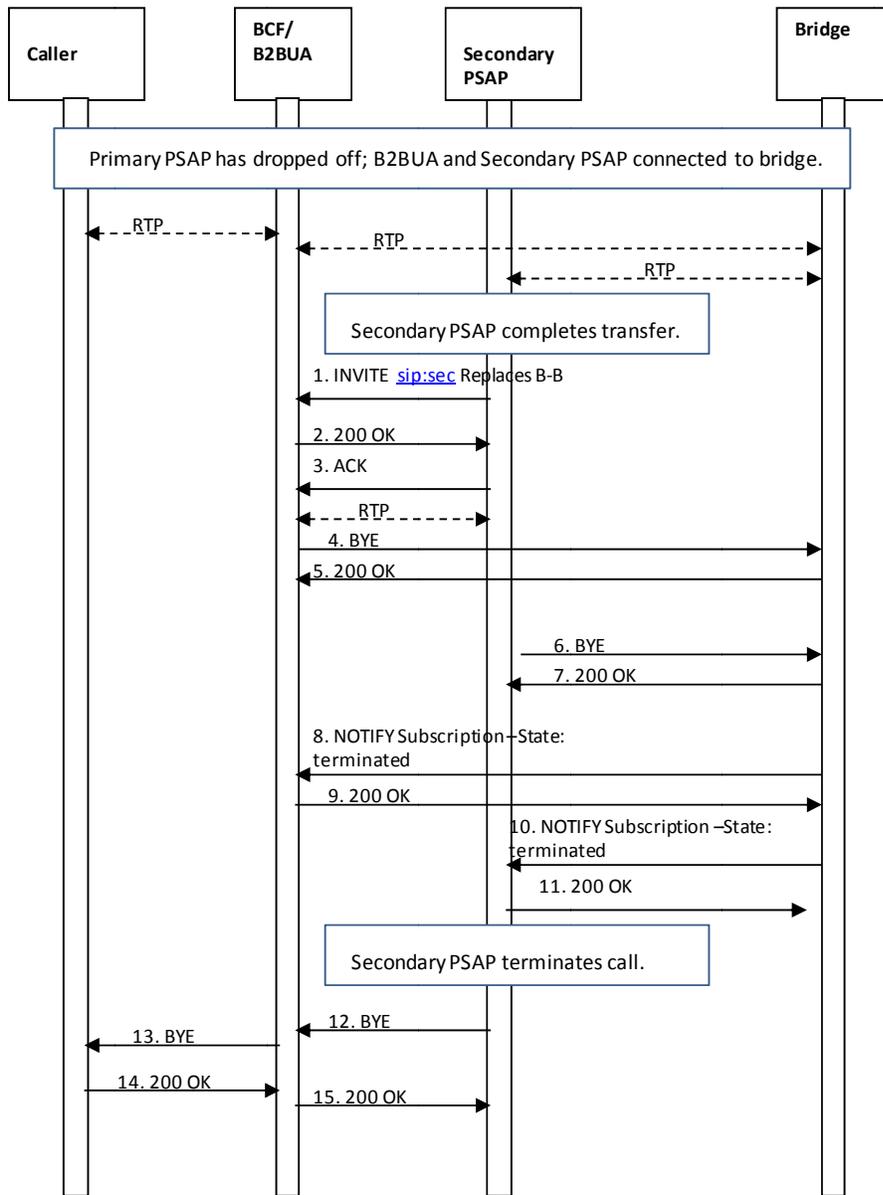
16. The bridge sends a NOTIFY message to the Primary PSAP to provide updated status of the subscription associated with the REFER request.

17. The Primary PSAP responds by returning a 200 OK message.

18. The bridge sends a NOTIFY message to the Primary PSAP to provide updated status of the subscription associated with the REFER request.

19. The Primary PSAP responds by returning a 200 OK message.

At this point, the Primary PSAP requests that the bridge add the Secondary PSAP to the conference, following the flow described in Section 5.7.1.3. Once the Primary PSAP determines that the transfer can be completed, it drops off the call, following the flow described in Section 5.7.1.4. The Secondary PSAP then completes the transfer as illustrated below. Note that the connection between the caller and the B2BUA is unaffected by the Primary PSAP disconnecting or the completion of the transfer by the Secondary PSAP. The following flow also illustrates termination of the emergency call initiated by the Secondary PSAP.



1. The Secondary PSAP completes the transfer by sending an INVITE to the B2BUA requesting that it replaces its connection to the bridge with a direct connection to the Secondary PSAP. The Secondary PSAP learns the URI of the B2BUA from the “Additional Data associated with a PSAP” data structure.
2. The B2BUA responds by returning a 200 OK message to the Secondary PSAP.
3. The Secondary PSAP returns an ACK in response to the 200 OK.

At this point, a media session is established between the B2BUA and the Secondary PSAP. The media session between the B2BUA and the bridge also still exists at this time. The B2BUA may cross connect media as per above

4. The B2BUA then sends a BYE to the bridge to terminate the session.
5. The bridge responds by sending the B2BUA a 200 OK message.
At this time the media session between the B2BUA and the bridge is torn down.
6. The Secondary PSAP also terminates its session with the bridge by sending a BYE message to the bridge.
7. The bridge responds by sending a 200 OK message to the Secondary PSAP.
At this point, the media session between the Secondary PSAP and the bridge is torn down.
8. The bridge then returns a NOTIFY message to the B2BUA indicating that the subscription to the conference has been terminated.
9. The B2BUA responds with a 200 OK message.
10. The bridge then returns a NOTIFY message to the Secondary PSAP indicating that the subscription to the conference has been terminated.
11. The Secondary PSAP responds with a 200 OK message.
At this point, the transfer is complete, and the caller and the Secondary PSAP are involved in a two-way call.
12. The Secondary PSAP determines that the call should be terminated and sends a BYE message to the B2BUA.
13. The B2BUA sends a BYE message to the caller to terminate the session.
14. The caller sends a 200 OK message to the B2BUA in response to the BYE.
15. The B2BUA sends a 200 OK to the Secondary PSAP in response to receiving the 200 OK from the caller. At this point the emergency session is terminated.

The B2BUA may act as a media relay for all media. All media packets on all negotiated media streams are relayed from one side of the B2BUA to the other.

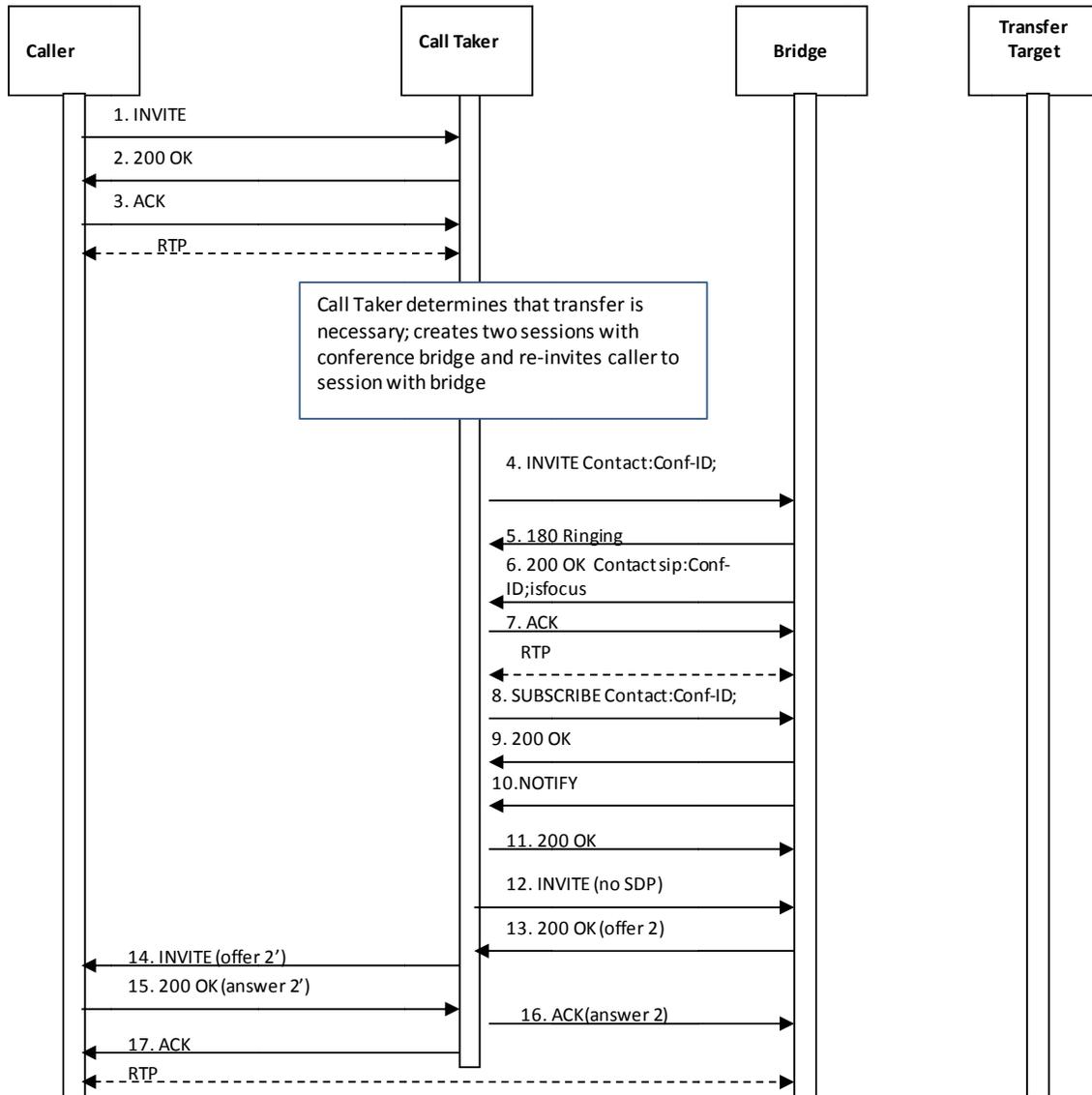
Characteristics of this solution are:

- The solution is deployed at the edge of the ESInet; the rest of the ESInet can assume Replaces works
- Media is anchored at the BCF regardless of what happens to the call
- The B2BUA is call stateful.
- The B2BUA is in the path regardless of whether the device implements Replaces or not.

5.8.2 Bridging at the PSAP Using Third Party Call Control in the Call Taker User Agent

RFC 3725 [35] describes a technique in which the initial answering UAC becomes a signaling B2BUA. If this method is chosen in an ESInet, a call taker UA receiving a call which does not contain a Supported header indicating support for Replaces must take the actions described in this section. Unlike the examples in RFC 3725, the caller has a call established with the call taker (which takes on the role of the “controller” in RFC 3725). The call sequence (based on RFC 3725 Flow IV) is described in the following subsections.

5.8.2.1 Call Taker Creates a Conference

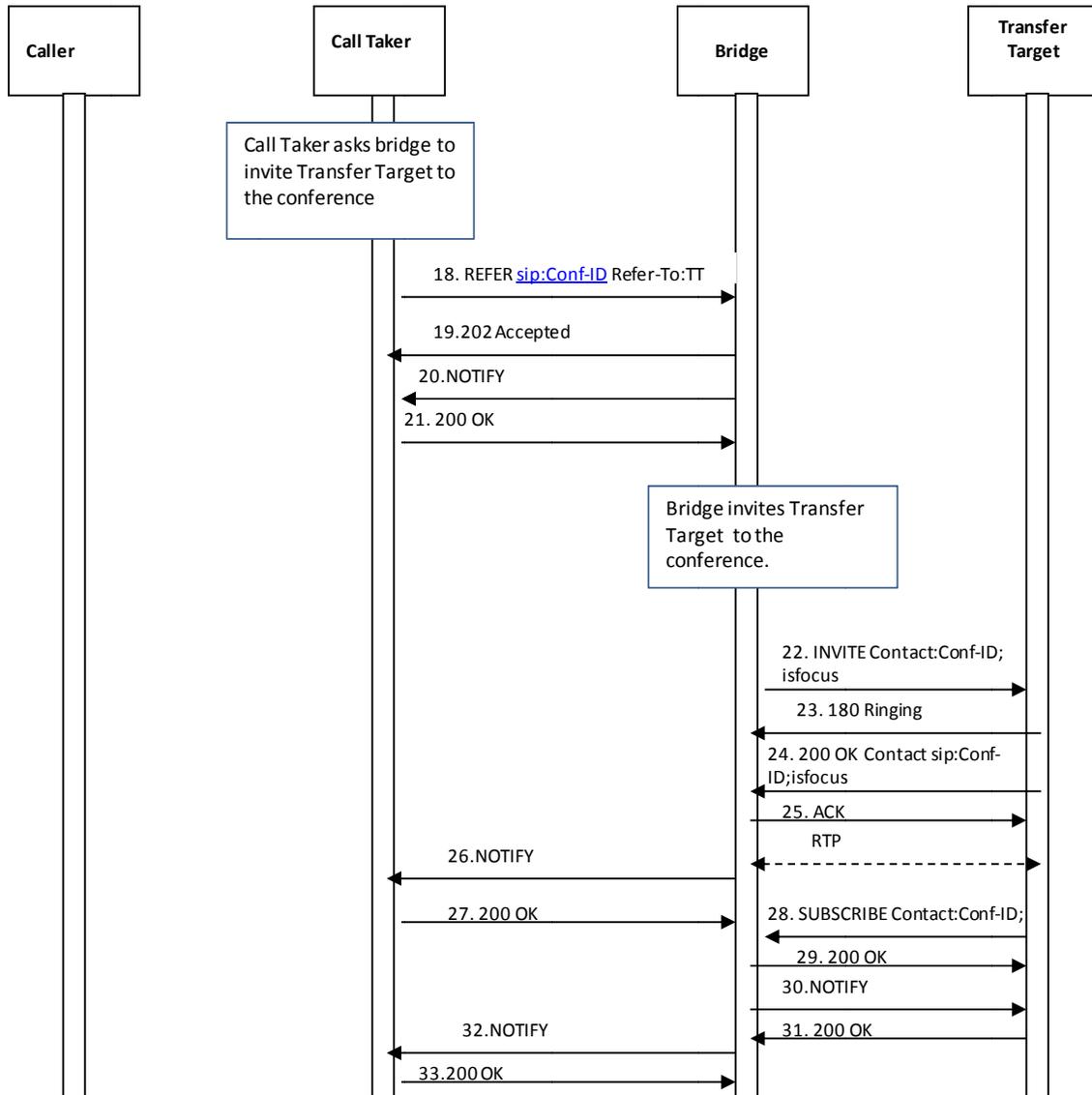


1. The caller initiates an emergency session request by sending an INVITE message via the i3 ESInet to the Primary PSAP call taker. The INVITE contains a Geolocation header with caller location information. (Elements and signaling involved in routing the emergency call within the i3 ESInet are not shown in this flow.)
2. The Primary PSAP responds by returning a 200 OK message to the caller's device.
3. The caller's device responds by sending an ACK to the Primary PSAP.
A media session is established between the caller and the Primary PSAP. The Primary PSAP determines that a transfer is necessary and uses SIP signaling to create a conference with a conference bridge, having previously received a Conference ID from a conference application (as described in Section 5.7.1.1).

4. The Primary PSAP initiates its first session with the bridge (with media) by sending it an INVITE message containing the Conf-ID.
5. The conference bridge responds to the INVITE by returning a 180 Ringing message.
6. The conference bridge then returns a 200 OK message, and a media session is established between the Primary PSAP and the conference bridge.
7. The Primary PSAP returns an ACK message in response to the 200 OK.
8. The Primary PSAP subscribes to the conference associated with the Conf-ID by sending a SUBSCRIBE message to the bridge.
9. The bridge responds by returning a 200 OK message.
10. The bridge then sends a NOTIFY message to the Primary PSAP providing the status of the subscription.
11. The Primary PSAP responds to the NOTIFY by returning 200 OK message to the bridge.
12. The Primary PSAP initiates its second session with the bridge (without media) by sending it an INVITE message with no SDP.
13. The bridge responds with a 200 OK that contains an offer (i.e., “offer 2”).
14. The Primary PSAP sends a re-INVITE to the caller’s device with the new offer.
15. The caller’s device responds by sending a 200 OK (providing an answer to the offer) to the Primary PSAP.
16. The Primary PSAP conveys the answer in an ACK sent to the bridge.
17. The Primary PSAP also sends an ACK to the caller’s device.

At this time, a media session is established directly between the caller and the bridge.

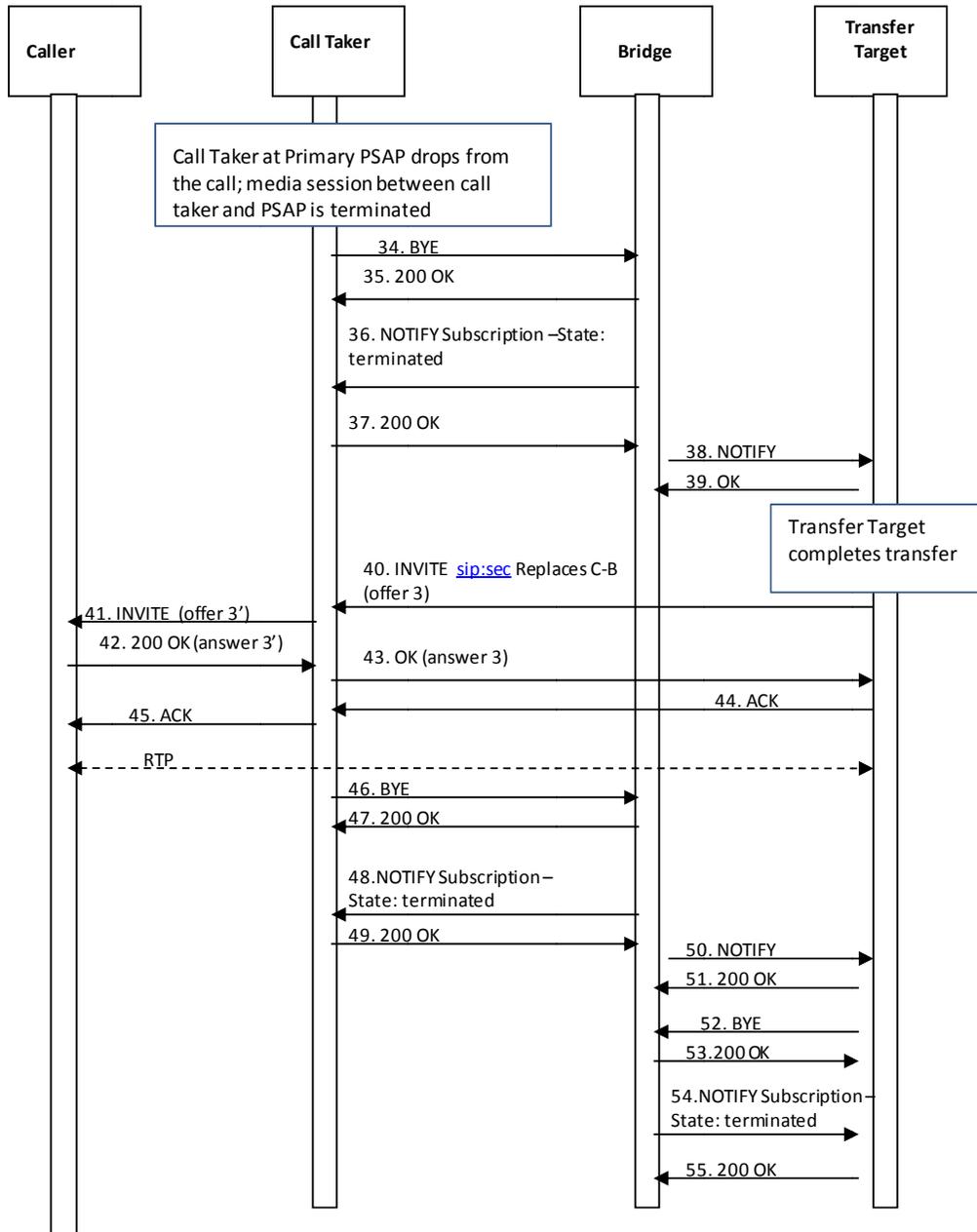
5.8.2.2 Call Taker Asks the Bridge to Invite the Transfer Target to the Conference



18. The Primary PSAP sends a REFER method to the conference bridge asking it to invite the Transfer Target (i.e., Secondary PSAP) to the conference. The REFER method contains the Conf-ID and a Refer-To header that contains the URI of the Transfer Target. The REFER method also contains an escaped Call-Info header field containing a reference URI that points to the “Additional Data Associated with a PSAP” data structure.
19. The bridge returns a 202 Accepted message to the Primary PSAP.
20. The bridge then returns a NOTIFY message to the Primary PSAP, indicating that subscription state of the REFER request (i.e., active).
21. The Primary PSAP responds by returning a 200 OK message.

22. The bridge invites the Transfer Target to the conference by sending an INVITE method containing the Conf-ID and the 'isfocus' feature parameter. The INVITE will also have the Call-Info header field containing a reference URI that points to the "Additional Data Associated with a PSAP" data structure.
23. The Transfer Target responds by returning a 180 Ringing message to the bridge.
24. The Transfer Target accepts the invitation by returning a 200 OK message.
25. The bridge acknowledges receipt of the 200 OK message by returning an ACK.
A media session is established between the Transfer Target and the bridge.
26. The bridge returns a NOTIFY message to the Primary PSAP to provide updated status of the subscription associated with the REFER request.
27. The Primary PSAP responds to the NOTIFY message by returning a 200 OK message.
28. The Transfer Target subscribes to the conference associated with the Conf- ID provided in the INVITE message from the bridge by sending a SUBSCRIBE message to the bridge.
29. The bridge acknowledges the subscription request by sending a 200 OK message back to the Transfer Target.
30. The bridge then returns a NOTIFY message to the Transfer Target to provide subscription status information.
31. The Transfer Target responds by returning a 200 OK message.
32. The bridge sends a NOTIFY message to the Primary PSAP providing updated status for the subscription associated with the REFER request.
33. The Primary PSAP responds to the NOTIFY message by returning a 200 OK message.
At this point the caller, Primary PSAP, and Transfer Target are all participants in the conference.

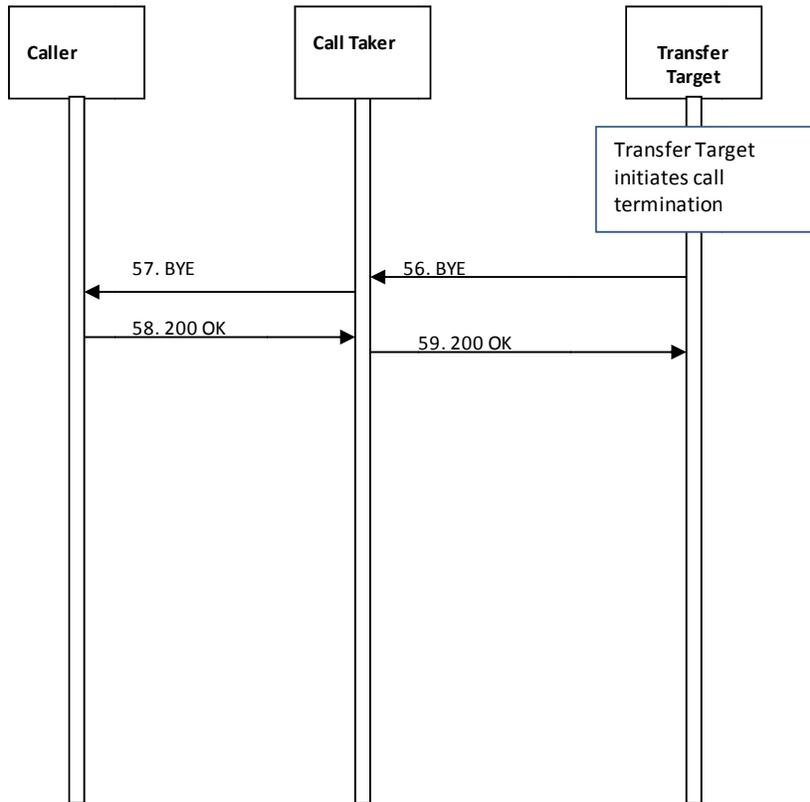
5.8.2.3 Primary PSAP Drops; Transfer Target Completes Transfer



34. The Primary PSAP initiates termination of its media session with the bridge by sending the bridge a BYE message.
35. The bridge responds by sending the Primary PSAP a 200 OK message.
At this time the media session between the Primary PSAP and the bridge is torn down.
36. The bridge sends a NOTIFY message to the Primary PSAP indicating that the subscription has been terminated.
37. The Primary PSAP responds by returning a 200 OK message.

38. The bridge sends a NOTIFY message to the Transfer Target to provide it updated status information.
39. The Transfer Target replies by returning a 200 OK message.
40. The Transfer Target completes the transfer by sending an INVITE to the Primary PSAP (acting as the B2BUA for the caller) asking it to replace its connection to the bridge (i.e., the media session between the caller and the bridge) with a direct connection to the Transfer Target (with offer 3). Note that the Transfer Target must be aware that it is the Primary PSAP that receives the INVITE.
41. The Primary PSAP sends a re-INVITE to the caller's device asking it to move the media from the bridge to the Transfer Target (with offer 3)
42. The caller's device responds by sending a 200 OK message back to the Primary PSAP (with answer 3).
43. The Primary PSAP sends a 200 OK message to the Transfer Target (with answer 3).
44. The Transfer Target acknowledges the 200 OK message by returning an ACK to the Primary PSAP.
45. The Primary PSAP acknowledges the 200 OK message by returning an ACK to the caller's device.
At this point, a media session is established directly between the caller and the Transfer Target.
46. The Primary PSAP sends a BYE to the bridge to terminate the session with the bridge.
47. The bridge responds by sending a 200 OK message to the Primary PSAP.
At this time the media session between the caller and the bridge is terminated.
48. The bridge sends the Primary PSAP a NOTIFY message indicating that the subscription has been terminated.
49. The Primary PSAP responds by sending a 200 OK message.
50. The bridge sends the Transfer Target a NOTIFY message to provide it updated information on the status of the conference.
51. The Transfer Target responds by returning a 200 OK message.
52. The Transfer Target sends a BYE to the bridge to terminate the session with the bridge.
53. The bridge responds by sending a 200 OK message to the Transfer Target.
At this point, the media session between the Transfer Target and the bridge is terminated.
54. The bridge sends the Transfer Target a NOTIFY message indicating that its subscription has been terminated.
55. The Transfer Target responds by sending a 200 OK message.

5.8.2.4 Transfer Target Terminates Session with Caller



56. The Transfer Target initiates call termination by sending the Primary PSAP a BYE message.
57. The Primary PSAP sends a BYE message to the caller's device to initiate request termination of the session.
58. The caller's device responds by returning a 200 OK message to the Primary PSAP.
59. The Primary PSAP responds by returning a 200 OK message to the Transfer Target.
At this time the media session between the caller and the Transfer Target is terminated.

In this transfer scenario, the Call Taker UA remains in the signaling path for the duration of the call. The media flows directly (via any BCF firewall of course) between the caller and the Transfer Target. Any further transfers would be accomplished in a similar manner, with the Call Taker UA accepting an INVITE with a Replaces header, and initiating a re-INVITE towards the caller to establish the correct media path.

This sequence is only necessary when the device does not implement Replaces. The Call Taker UA can notice the presence of the Supported header, and if Replaces is supported, it can just initiate a transfer using standard SIP methods, as described in Section 5.7. It could, optionally, attempt the Replaces even if a Supported header was not found, detect an error and initiate the re-INVITE as above in response.

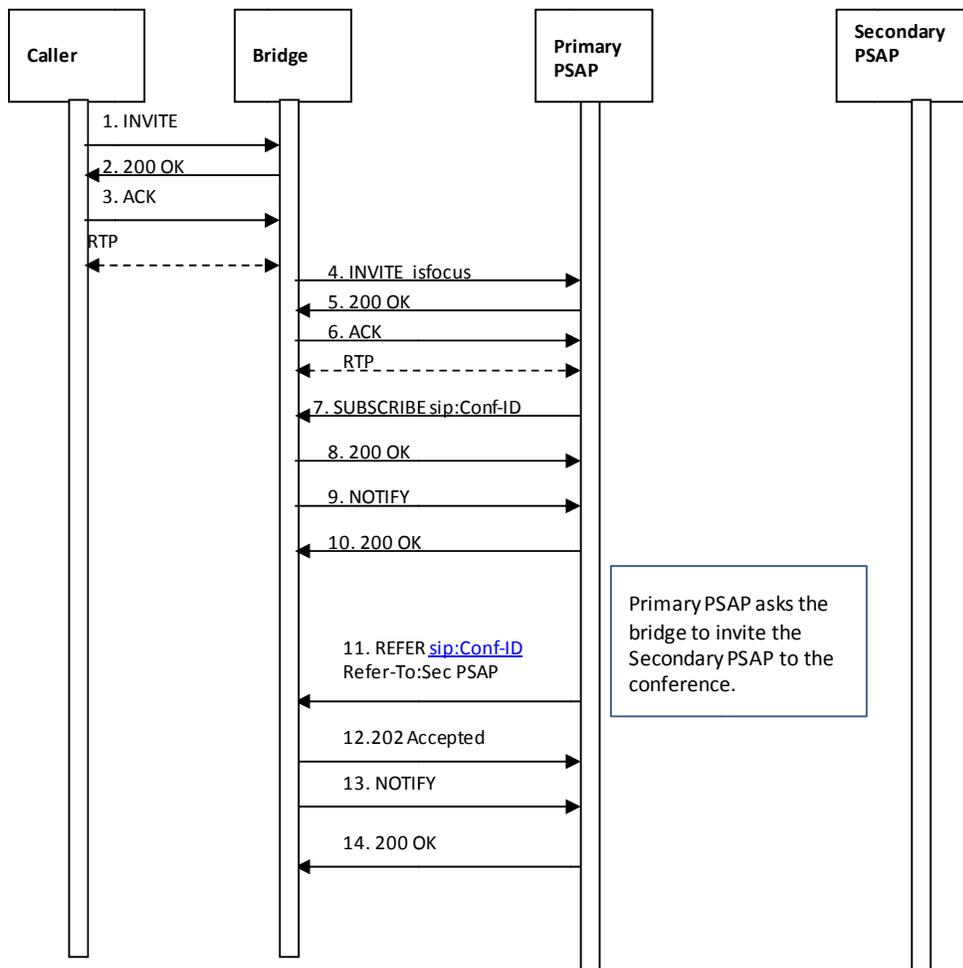
The characteristics of this solution are:
Version 1, June 14, 2011

- No additional network signaling elements in the path unless necessary
- Media goes direct between endpoints
- Caller UA receives multiple Re-INVITE messages

5.8.3 Answer all calls at a bridge

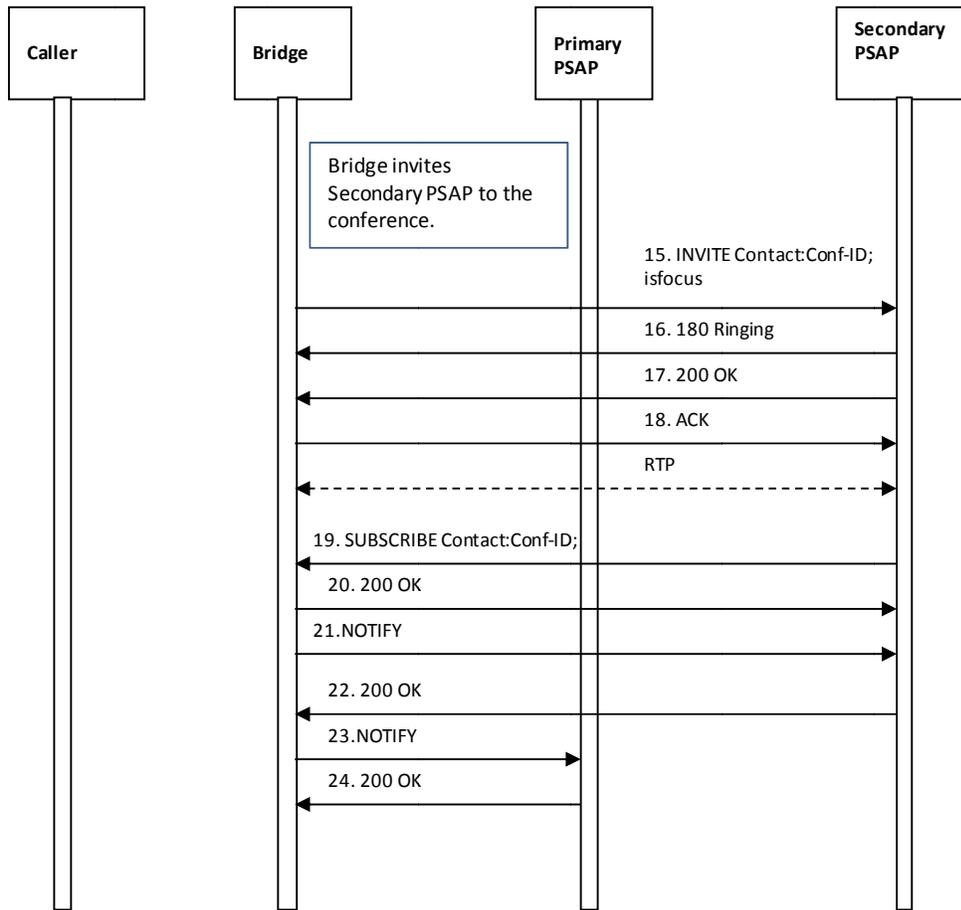
All incoming 9-1-1 calls are answered at a bridge. When the bridge receives a call for the URI specified in the last hop LoST route, the bridge creates the caller to bridge leg, and initiates an INVITE to the PSAP/Call Taker (depending on configuration and where the bridge is located: in the network or in the PSAP). The caller remains on the bridge where it was first answered. The call taker can add other parties to the bridge, other parties can add additional parties, parties can drop off the bridge, and the caller to bridge leg remains stable.

5.8.3.1 Call Established Between Caller and Primary PSAP Via Bridge; Primary PSAP Asks Bridge to Invite the Secondary PSAP to the Conference



1. The caller initiates an emergency session request by sending an INVITE message into the i3 ESInet. The INVITE contains a Geolocation header with caller location information. (Elements and signaling involved in routing the emergency call within the i3 ESInet are not shown in this flow.) The call is routed using i3 mechanisms, and the URI of the target Primary PSAP is determined. The call is delivered to a bridge in the i3 ESInet.
2. Upon receiving the INVITE from the caller, the bridge responds by returning a 200 OK to the caller.
3. The caller returns an ACK in response to the 200 OK from the bridge.
A media session is established between the caller and the bridge.
4. Upon receiving the call at the bridge, the bridge initiates a call to the Primary PSAP by sending an INVITE message. The INVITE message generated by the bridge must include a Geolocation header that contains the caller location information received in the Geolocation header of the INVITE message from the caller, as well as any Call-Info headers that were received in the incoming INVITE message.
5. The Primary PSAP responds by returning a 200 OK message to the bridge.
6. The bridge responds by sending an ACK to the Primary PSAP.
A media session is established between the bridge and the Primary PSAP.
7. Once the media session is established, the Primary PSAP sends a SUBSCRIBE message to the bridge to subscribe to the conference associated with the Conf-ID identified when the conference was initially established with the bridge.
8. The bridge responds to the SUBSCRIBE message by returning a 200 OK message to the Primary PSAP.
9. The bridge then returns a NOTIFY message to the Primary PSAP to provide it with status information regarding the conference.
10. The Primary PSAP responds to the NOTIFY message by returning a 200 OK message.
11. The Primary PSAP sends a REFER method to the bridge asking it to invite the Secondary PSAP to the conference. The REFER method contains the Conf-ID and a Refer-To header that contains the URI of the Secondary PSAP. The REFER method also contains an escaped Call-Info header field containing a reference URI that points to the “Additional Data Associated with a PSAP” data structure.
12. The bridge returns a 202 Accepted message to the Primary PSAP.
13. The bridge then returns a NOTIFY message, indicating that subscription state of the REFER request (i.e., active).
14. The Primary PSAP returns a 200 OK in response to the NOTIFY message.

5.8.3.2 Bridge Invites the Secondary PSAP to the Conference

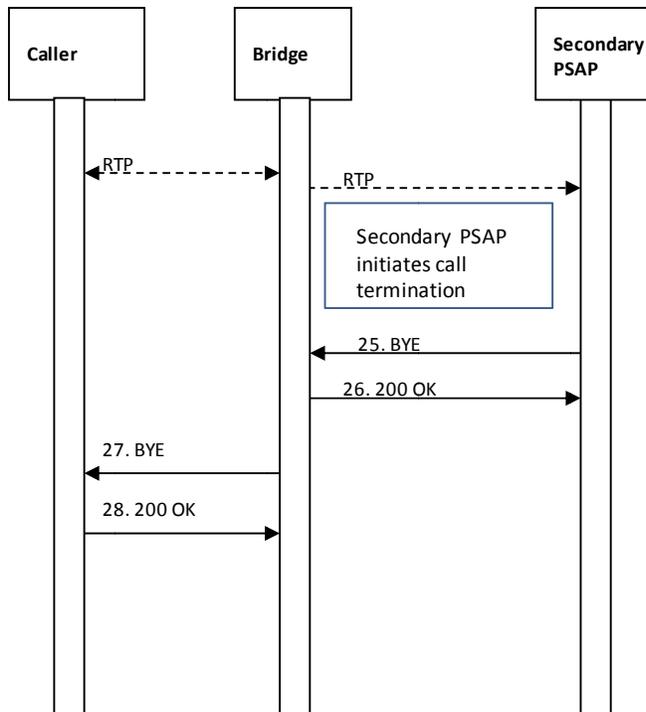


15. The bridge invites the Secondary PSAP to the conference by sending an INVITE method containing the Conf-ID and the isfocus feature parameter. The INVITE also contains a Call-Info header containing a reference URI that points to the “Additional Data Associated with a PSAP” data structure.
16. The Secondary PSAP UA responds by returning a 180 Ringing message to the bridge.
17. The Secondary PSAP accepts the invitation by returning a 200 OK message.
18. The bridge acknowledges receipt of the 200 OK message by returning an ACK.
A media session is established between the Secondary PSAP and the bridge.
19. The Secondary PSAP subscribes to the conference associated with the Conf-ID provided in the INVITE message from the bridge by sending a SUBSCRIBE message to the bridge.
20. The bridge acknowledges the subscription request by sending a 200 OK message back to the Secondary PSAP.
21. The bridge then returns a NOTIFY message to the Secondary PSAP to provide subscription status information.

22. The Secondary PSAP responds by returning a 200 OK message.
23. The bridge sends a NOTIFY message to the Primary PSAP providing updated status for the subscription associated with the REFER request.
24. The Primary PSAP responds to the NOTIFY message by returning a 200 OK message.
At this point the caller, Primary PSAP, and Secondary PSAP are all participants in the conference.

5.8.3.3 Secondary PSAP Terminates the Call

When the Primary PSAP determines that it can drop from the bridge, it will follow the flow described in Section 5.7.1.4. When the Secondary PSAP determines that the call should be terminated, it will follow the flow illustrated below.



25. Secondary PSAP initiates call termination by sending a BYE message to the bridge.
26. The bridge responds by returning a 200 OK message.
At this point, the session between the bridge and the Secondary PSAP is torn down.
27. The bridge sends a BYE message to the caller's device.
28. The caller's device responds by returning a 200 message to the bridge.
At this point, the session between the caller and the bridge is torn down.

The characteristics of this solution are:

- Media is anchored at the bridge regardless of what happens to the call.

- The bridge is always in the path regardless of whether the device implements Replaces or not.
- The original bridge is always in the path whether the Primary PSAP subsequently transfers the call or not. Receipt of the call on the bridge must trigger dial out of the call to the Primary PSAP/call taker.
- The bridge must populate the (original) caller location information received in the Geolocation header of the incoming INVITE message in the Geolocation header of the outgoing INVITE message to the Primary PSAP.
- The bridge must populate any Call-Info headers received in the incoming INVITE message in the outgoing INVITE message to the Primary PSAP.
- Termination of the Secondary PSAP leg causes the bridge to (automatically) terminate the leg to the caller.
- Note that call taker's system behaves differently in this scenario in that the initial call is received with an 'isfocus' feature parameter; call taker need not establish a bridge if it determines that a transfer is necessary

5.8.4 Recommendations

BCFs should support option 1. This is the most likely scenario for most networks and has no impact or dependency on other elements. PSAP CPE may support option 2, which has no impact or dependency on other elements. PSAP CPE may support option 3 if the bridge support is available. Bridges may support Option 3. ESIInet designers must decide which mechanism will be used on their network and all appropriate elements must support that mechanism. Consideration must be given to how calls will be transferred to or accepted from ESIInets making different choices. Only ONE mechanism should be enabled. Other methods are acceptable provided that they do not assume/require support of Replaces by calling devices. Selection of a method to handle the lack of Replaces implementations in calling devices must take into account how overall system reliability goals are to be met, and specifically, how failures of various elements in the solution affect call reliability.

5.9 Location Information Server (LIS)

A Location Information Server supplies location, in the form of a PIDF-LO (location by value) or a location URI (location by reference). The LIS also provides a “dereference” service for a location URI it supplies: given the URI, the LIS provides the location value as a PIDF-LO. A LIS may be a database, or may be a protocol interworking function to an access network specific protocol.

In NG9-1-1, the LIS supplies location (by value or reference) to the endpoint, or proxy operating on behalf of (OBO) the endpoint. The ESIInet is not directly involved in that transaction: the resulting PIDF-LO or location URI must appear in the initial SIP message in a Geolocation header. If the LIS supplies location by reference, it must also provide dereferencing service for that location URI. Elements in the ESIInet, including the ESRP and PSAP may dereference a location URI as part of processing a call.

If the LIS supplies location by reference, it must support HELD [9] and/or SIP Presence Event Package [31]. The SIP Presence Subscribe/Notify mechanism can control repeated dereferencing,

especially when tracking of the caller is needed. However, HELD is acceptable on any location URI. LISs supporting SIP must support location filters [103] and event rate control [113].

LISs queried by Legacy Network Gateways during the processing of a wireline emergency call would typically use HELD with the identity extension [104] using a telephone number as the identity and supply location by value in return.

LISs queried by Legacy Network Gateways during the processing of wireless emergency calls are usually protocol interwork functions between SIP or HELD and the legacy network's location determination subsystem. Typically they would supply location by reference.

If the broadband network supports true mobility, it should supply location by reference. If the broadband network is a fixed network like a cable modem network or DSL, location by value is preferred, but location by reference is acceptable.

A LIS must validate locations prior to entering them in to the LIS using the LVF (Section 5.4).

A LIS must accept credentials traceable to the PCA for authenticating queries for a location dereference. Since calls may be diverted to any available PSAP, the LIS cannot rely on any other credential source to authorize location dereferencing.

When location is provided by reference there is a need for the reference to be valid at least for the length of the call. Whether the reference should remain valid for some time beyond the duration of the call is a topic for future study as are the privacy considerations of such access.

5.10 Call Information Database (CIDB)

A call that passes through an origination network or service provider of any kind must have a Call info header with a URI that resolves to an AdditionalCall Data structure. The database that dereferences this URI is a Call Information Database. There is a minimum amount of information listed as Mandatory in NENA 71-001 that mirrors the information currently provided by all origination networks in the ALI.

Important Note: The version of 71-001 that was in effect as of the release of this document requires the <list of fields> as mandatory elements. Within a CIDB, these elements are optional, and a future edition of 71-001 will correct this and only <list of fields> are the minimum fields that must be provided.

All origination networks and service providers (where a service provider here is a 3rd party in the path of a call which is not the originating network presenting the call to the ESInet) are required to provide at least this minimum set of information which must be populated in a CIDB. The CIDB is queried with the URI obtained from the Call-Info header with a purpose of emergencyCallData, and returns the Additional Call Data structure NENA 71-001 Section 8.1 [105]. The query is an HTTPS GET with the URI obtained from the Call-Info header. The return is the XML data structure as defined in NENA 71-001. It is important that ALL service providers handling the call add a Call-Info header and supply a CIDB to dereference it. The transaction to dereference the Additional Call Data URI must be protected with TLS. The dereferencing entity, which may be an ESRP, PSAP or responding agency uses its credentials (traceable to the PCA for NG9-1-1 entities). The service

provider can use any credential, as long as the domain listed in the URI is the domain of the SubjectAltName in the cert.

Call Information Database servers are not required to be able to serve a query more than 5 minutes after an emergency call is terminated.

Devices such as telematics equipped vehicles and medical monitoring devices that can place emergency calls should have the capability to respond to a CIDB query, which includes the reference to the device data (telematics, health monitoring, ...). A service provider (such as a telematics service provider) may provide the CIDB instead of the device. Other devices may also provide a CIDB for use in an emergency call.

The CIDB could be provided by the origination network or service provider. For service providers and origination networks that only provide the minimal data called for in NENA 71-001, the CIDB could be provided by a third party. Extension of SIP to allow the data to be included by value in the signaling is for future study.

5.11 Interactive Media Response system (IMR)

The IMR is similar to an Interactive Voice Response (IVR) unit, but handles audio, video and text media. It may be used to answer calls when the PSAP is receiving more calls than it has call takers to answer them. It offers interaction with the caller (“Press 1 if this about the car crash on Fourth and Main, Press 2 if this is about some other problem”).

IMRs must implement RFC4240 [43], and VXML V2.0 [134]. VXML <audio> tags must specify multiple MIME types with appropriate types for the media. Synthesis scripts must render text for text media. The IMR must implement at least the codecs listed in Section 4.1.8

The syntax for specifying a URI to route to a specific VXML script is defined in RFC4240.

Calls may be queued within the IMR waiting for available call takers. The queue of calls must be a queue as defined in 5.2.1.2 and maintain the specified queueState and DequeRegistration events so that PSAP management can monitor and control the queue as it does all other queues.

IMRs must interpret an IM, RTT or other text received consisting the digits 0-9, ‘#’ or ‘*’ immediately following a prompt for input as equivalent to DTMF key presses.

5.12 Logging service

The logging service in NG9-1-1 is a standardized functional element used by all elements in an ESInet to log all significant events; logging is not restricted to events within a PSAP. All significant steps in processing a call are logged. NG9-1-1 defines an external logging service interface so that the logging function can be provided in the ESInet. Logging includes external events, internal events, media and messages.

5.12.1 Interfaces

The log service is primarily a web service. In addition to the web service interface, logging services that record media provide an RTSP (RFC2326 [135]) interface to play back the media. The web service includes the following functions.

5.12.1.1 LogEvent

LogEvent logs an event into the logging service. The LogEvent includes parameters:

Parameter	Type	Description
timestamp	String	A timestamp as defined in Section 3.2
agencyOrElement	String	agencyID or hostname of an element which logged the event
agent	String	The agentId (Section 3.1.1) of an agent at the agency listed in the agentOrElement tag, see Section 3.1.2
callId	String	The call identifier of a call, see Section 3.1.4
incidentId	String	The Incident Tracking Identifier associated with the call, see Section 3.1.5
eventType	Enumeration	Type of log record

Each EventType contains additional data specific to the EventType.

The following EventTypes are defined in this document

CallProcess: Each element which is not call stateful, but handles a call logs the fact that it saw the call pass through by logging a CallProcess event. There are no parameters to “Call Process”

StartCall/EndCall: Each element which is call stateful logs the beginning and end of its processing of a call with Start Call and End Call events. StartCall includes a copy of the headers in the INVITE message, encoded in <header> tags. EndCall includes the response code that ended the call (200 OK in the case of a successful call), encoded in a <responseCode> tag.

Note: it may be desirable to log other messages that are part of the INVITE transaction, such as the ACK. This will be covered in a future edition of this document.

TransferCall: When a call is transferred, the transfer is logged by the transferor (the PSAP which had the call prior to transferring it. The transfer target URI is logged in a <transferTarget> tag.

Route: Proxy servers that make routing decisions (ESRPs or other SIP proxy servers in the path of the call) log the route it selected with the Route EventType. The URI where it decided to send the call (encoded in a <uri> tag, plus a text string <reason> for choosing that

route are included in the LogEvent. For ESRPs, the name of the rule is included in a <rule> tag.

Media: Media is the log of call media (voice, video and interactive text). The media event includes a text string <udp> tag that contains an RFC2327 Session Description Protocol [55] description of the media. The SDP must include SDES keys if the RTP stream is protected with SRTP. Each independent stream must include an RFC4574 [136] label to identify each stream and the label must be logged with a <mediaLabel> tag. More than one Media event can occur for a call. Recorded media streams include integral time reference data within the stream.

EndMedia. EndMedia causes the logging service to terminate recording of media. The EndMedia event includes one or more <mediaLabel> tags which must match the SDP labels in the corresponding Media event. More than one EndMedia (with different <mediaLabel>s) may occur for a call.

Message: An SIP Message (Instant Message) is logged with a Message log event. The text of the message is included as a <text> parameter.

AdditionalAgency: When an agency becomes aware that another agency may be involved, in any way, with a call, it must log an AdditionalAgency event. The AdditionalAgency event includes an <agency> tag which is an Agency Identifier (see Section 3.1.1). Among other uses, this event is used by PSAP management to query all logging services that may have records about a call or incident.

Note: a mechanism to discover the logger associated with an agency will be provided in a future edition of this document

MergeIncident: at some point in processing, an agency may determine that a call marked with an IncidentId may in fact be part of another, previously determined Incident. When it is determined that two IncidentIds have been assigned for the same real world Incident, the Ids are merged with MergeIncident. The MergeIncident record contains the IncidentId of the incorrectly assigned incident in the <incidentId> tag in the header of the log record, and the Incident Id of the actual Incident in an <actualIncident> tag. Note that other agencies may not know that the Incidents are being merged, and therefore could log events against the originally assigned IncidentId.

ClearIncident: When an agency finishes its handling of an Incident, it logs a ClearIncident record. Other agencies may still be processing the Incident.

ECRFquery: any element that queries the ECRF and the ECRF itself generate an ECRFquery LogEvent. The LogEvent includes the PIDs-LO (and only the Location Object) using the RFC4119 tags and the service URN in a <service-urn> tag.

ECRFresponse: Both the elements that query the ECRF and the ECRF generate the ECRFresponse. The entire response is logged using the LoST tags.

This document creates a registry for LogEvents. See Section 12.11.

LogEvent function assigns a globally unique LogIdentifier to each LogEvent and returns the LogIdentifier in its response. The form of a LogIdentifier is a URI consisting of the string “_LI_”, a unique string, the “@” character and the domain name of the logging service. The unique string must be between 10 and 35 characters long and unique to the logging service. An example LogIdentifier is LI_0013344556677-231@logger.state.pa.us. The domain specified must be the domain of the logging service to which the appropriate RetrieveLogEvent can be sent.

5.12.1.2 RetrieveLogEvent

To retrieve a logged event from the logging service, RetrieveLogEvent will return the log record for all events. The request to RetrieveLogEvent includes a <logIdentifier> parameter, as returned by the original LogEvent.

When the event is a Media event, the returned event from RetrieveLogEvent will not have the SDP parameter, but will instead have an <rtsp> parameter that must be an RTSP URL. The RTSP URL can be used to play back the media stream(s).

An <errorCode> is also returned from RetrieveLogEvent that can include:

Error Codes

100	Okay	No error
517	No such logIdentifier	
504	Unspecified Error	

Policy controls who can retrieve logged events from the logging service. The policy of the element/agency which logged the event governs.

5.12.1.3 ListEventsByCallId

Returns a list of LogIdentifiers that have a specified Call Identifier. The request includes the <callIdentifier>. The response includes zero or more <logIdentifier>(s). An <errorCode> is also returned that can include:

100	Okay	No error
518	No such callIdentifier	
504	Unspecified Error	

5.12.1.4 ListEventsByIncidentId

Returns a list of LogEvents that have a specified Incident Tracking Identifier. The request includes the <incidentIdentifier>. The response includes zero or more <logIdentifier>(s). An <errorCode> is also returned that can include:

100	Okay	No error
519	No such incidentIdentifier	
504	Unspecified Error	

5.12.1.5 ListCallsbyIncidentId

Returns a list of Call Identifiers associated with a specific Incident Tracking Identifier. The request includes the <incidentIdentifier>. The response includes zero or more <callIdentifier>(s). An <errorCode> is also returned that can include:

- 100 Okay No error
- 519 No such incidentIdentifier
- 504 Unspecified Error

5.12.1.6 List IncidentsByDateRange

Returns a list of Incident Tracking Identifiers occurring within a time/date range. The request includes a <startTime> timestamp and an <endTime> timestamp. The response includes zero or more <incidentIdentifier>(s). An <errorCode> is also returned that can include:

- 100 Okay No error
- 519 Bad Timestamp
- 520 EndTime occurs before StartTime
- 504 Unspecified Error

5.12.1.7 ListIncidentsByLocation

Returns a list of Incidents that occurred within a specified geographic region. The request includes a GML shape in a <areaOfInterest> tag. The response includes zero or more <incidentIdentifier>(s). An <errorCode> is also returned that can include:

- 100 Okay No error
- 521 Bad Geoshape
- 504 Unspecified Error

5.12.1.8 ListIncidentsByDateAndLocation

A combination of ListIncidentsbyDateRange and ListIncidentsByLocation, the request includes a <startTime>, <endTime> and <areaOfInterest>. The response includes zero or more <incidentIdentifier>(s).). An <errorCode> is also returned that can include:

- 100 Okay No error
- 519 Bad Timestamp
- 520 EndTime occurs before StartTime
- 521 Bad Geoshape
- 504 Unspecified Error

5.12.1.9 ListCallsByDateRange

Returns a list of Call Identifiers occurring within a time/date range. The request includes a <startTime> timestamp and an <endTime> timestamp. The response includes zero or more <callIdentifier>(s). An <errorCode> is also returned that can include:

100	Okay	No error
519	Bad Timestamp	
520	EndTime occurs before StartTime	
504	Unspecified Error	

5.12.1.10 ListAgenciesByCallId

Returns a list of agencies that recorded AdditionalAgency events about a call. The request includes a <callIdentifier>. The response includes zero or more <agencyIdentifier>(s). An <errorCode> is also returned that can include:

100	Okay	No error
518	No such callIdentifier	
504	Unspecified Error	

5.12.1.11 ListAgenciesByIncidentId

Returns a list of agencies that recorded AdditionalAgency events about an Incident. The request includes an <incidentIdentifier>. The response includes zero or more <agencyIdentifier>(s). An <errorCode> is also returned that can include:

100	Okay	No error
519	No such incidentIdentifier	
504	Unspecified Error	

5.12.2 Instant Recall Recorder

The ability to quickly review current or recent emergency communications content must be provided. The Logging service's Web Service interface supports this capability with the query, retrieval and streaming media functions described in section 5.12. This interface supports recall of all defined media types. A client application may use these functions to retrieve media for display or playback. The client is expected to impose any additional limitations required by local policy, such as limiting recall to communications the user has handled, to specific communications types, and/or limiting the time period from which recent communications can be recalled. The client is also responsible for providing functionality that allows the user to navigate within and between recalled communications. Access to media for instant recall is subject to the same security restraints as all log records. The PSAP may impose additional constraints on which agents may access media.

5.12.3 Roles and Responsibilities

Any agency including a PSAP may run its own logging service. The ESInet may have one or more logging services. All agencies and NG9-1-1 functional elements must have access to a conformant logging service and log all relevant events in it. Media is recorded by the entity answering the call, and by any bridge in the path. Recording of media at the BCF can be substituted for recording of media at the endpoints if the BCF is always in the path of all media.

5.12.4 Operational Considerations

To be supplied in a future edition of this standard.

5.13 Forest Guide

The ECRF and LVF infrastructure make use of Forest Guides as defined in RFC5582 [60]. A server that does not answer the query can refer to a Forest Guide to determine the response.

5.13.1 Functional Description

The following definitions are adapted from those in RFC5582 used with permission of the authors:

- authoritative ECRF/LVF: A LoST server that can provide the authoritative answer to a particular set of queries, e.g., covering a set of civic labels or a particular region described by a geometric shape. An authoritative ECRF/LVF may redirect or forward a query to another authoritative ECRF/LVF within the tree.
- child: An ECRF/LVF which is authoritative for a subregion of another authoritative ECRF/LVF. A child can in turn be parent for another authoritative ECRF/LVF.
- (tree node) cluster: A node cluster is a group of ECRFs that all share the same mapping information and return the same results for queries. Clusters provide redundancy and share query load. Clusters are fully-meshed, i.e., they all exchange updates with each other.
- coverage region: The coverage region of an authoritative ECRF/LVF is the geographic region within which the ECRF/LVF is able to authoritatively answer mapping queries. Coverage regions are generally, but not necessarily, contiguous and may be represented as either a subset of a civic address or a geometric object.
- forest guide (FG): A forest guide has knowledge of the coverage region of trees for a particular top-level service.
- parent: A LoST server that covers the region of all of its children. A LoST server without a parent is a root authoritative ECRF/LVF.
- tree: A self-contained hierarchy of authoritative mapping servers for a particular service. Each tree exports its coverage region to the forest guide.

Given a query to an area outside its coverage area, an ECRF/LVF may have the coverage regions of other ECRF/LVFs to which it could refer a query, or it would refer to a Forest Guide. In NG9-1-1, each state is a tree, with local ECRF/LVFs as the children. The top of the tree is a state ECRF/LVF. There is a national forest guide that has knowledge of the state trees. The national forest guide exchanges mappings with other national forest guides. A state mapping, exported to the national

forest guide is the civic state element, and a polygon representing the state boundary (or more precisely, the union of the coverage regions of all PSAPs in the state).

5.13.2 Interface Description

The national forest guide maintains a LoST interface, as described in Section 4.5, for query resolution. It also maintains a LoST-sync interface defined in draft-ietf-lost-sync [112] for updating its coverage regions. The LoST-sync interface is used for both state ECRF/LVF interfaces and other national forest guides. The national forest guide only serves urn:service:sos, urn:nena:service:sos and urn:nena:service:responder. It may be able to refer to other forest guides for services other than these.

5.13.3 Data Structures

The Forest Guide has a civic data structure (PIDF-LO down to the A2 level) and a GML polygon (set) representing the state coverage region. It also maintains mappings for other countries in a similar manner (civic A1 level, plus a polygon set for the country coverage region).

5.13.4 Roles and Responsibilities

The Forest Guide must be managed nationally (agency not yet identified) and may evolve to an entity more representative of all public safety agencies. State ECRF and LVF operators are responsible to arrange for their mappings to be provisioned in the national forest guide. The national forest guide operator will maintain well known contact information so that other national forest guides can arrange to exchange their coverage regions and mappings.

5.13.5 Operational Considerations

While the national forest guide is only authoritative for the service urns listed above, it may refer other queries to other forest guides if it knows the forest guide for that service. The forest guide idea is specifically designed so that there is no global “root” forest guide. This means that the national forest guide will have to develop policies for its own operation when deciding what is an authoritative forest guide for another country or area. Specifically, it can be expected to have to deal with disputed territory, where more than one national forest guide claims they are authoritative for the same area.

5.14 DNS

All elements identified by hostnames must have corresponding Domain Name Service (DNS) records STD13 [106] in the global public DNS. All elements connected to the ESInet must have local DNS resolvers to translate hostnames they receive to IP addresses. Since the ESInet must continue to work in the face of disasters, DNS servers must be highly redundant, and resolvers must be able to use cached records even if they have expired if they lose connections to authoritative DNS servers to resolve names.

A domain that has SIP elements within the domain must have an SRV record RFC2782 [107] for a SIP service for the domain, and any of its subdomain which may appear in a URI.

5.15 Agency Locator

To be provided in a future edition of this document.

5.16 Policy Store

5.16.1 Functional Description

A policy store holds policies created by an agency and used by a functional element such as an ESRP. The policy store is a simple repository; it does not manipulate the policy.

5.16.2 Interface Description

A policy store implements the policy storage and retrieval functions defined in section 4.4.1. Policy store replicas can be maintained by having one policy store retrieve policies from another policy store and subsequently accept requests to retrieve such policies. Replicas normally do not allow a policy store operation for a policy that they replicate. There is always one (possibly redundant) authoritative policy store for a given policy.

5.16.3 Roles and Responsibilities

Any agency may operate a policy store. While it is permissible for an element to contain a policy store that it uses, it normally is not authoritative, but rather a replica of the policy, and the element must have a mechanism to not use the internally stored replica, but rather retrieve the policy from the authoritative source if provisioned to do so.

5.17 Time Server

The ESInet must provide an NTP service for time-of-day information. The service may have a hardware clock, or may be synchronized to another NTP time service provided that there are sufficient backups so that if the ESInet is isolated from its time source, it can provide local time. Time accuracy must be within 1 ms of true time. Agencies may have their own time server, which may have a hardware clock if it is more accurate than syncing the server to the ESInet time server.

5.18 Origination Networks and Devices

A device, network or service provider presenting calls to an ESInet must support the following interfaces. How the origination network, device or service arranges its emergency calling services to meet this standard is beyond the scope of this document.

5.18.1 SIP Call Interface

The origination network must present calls to the ESInet meeting the ESInet SIP interface specified in Section 4.1. All calls must be signaled with SIP, must contain a geolocation header, except if they are calls to an administrative number, and must be routed by the ECRF, or an equivalent function that produces the same result, using the location contained in, or referenced by the Geolocation header.

5.18.2 Location by Reference

Origination networks that are also access networks must also provide a Location Information Server function (that is, location dereference, and location validation if applicable) meeting the requirements of section 5.9 if they supply location by reference.

5.18.3 Call Information Database

Origination networks and devices presenting calls to ESInets must provide a Call Information Database interface meeting the requirements of section 5.10.

6 Security

6.1 Identity

Each agency and each agent in an agency are issued credentials that allow them to be identified to all services in the ESInet. An agency identifier is a globally unique domain name (such as erie.psap.ny.us), which appears in the SubjectAltName of an X.509 certificate issued to the agency. The agency assigns identities to an agent. The identity for an agent is a string containing a userpart which is unique to the agent within the agency, an “@” and the domain name of the agency. For example: nancy@erier.psap.ny.us. This string appears in the SubjectAltName of an X.509 certificate issued to the agent. See Identifiers in Section 3.1

For PSAPs and 9-1-1 Authorities, the root Certificate Authority for agent and agency certificates is the PSAP Credentialing Agency. The certificate can be issued directly by the PCA, or the PCA can issue a certificate to an agency which, in turn, issues certificates to other agencies or agents. It is recommended that a state PCA be created for each state, with the national PCA signing the state PCA certificate, and the state PCA signing 9-1-1 Authority and PSAP certificates. 9-1-1 Authorities or PSAPs may sign the certificate of their agents.

Operating a CA requires creation of, and strict adherence to a Certificate Policy and Practice Statement (CP/CPS) CP/CPS includes strict specifications for vetting: who gets a certificate, under what conditions they get a certificate, and what proof of identity is needed before a certificate can be issued. If an agency cannot reasonably control its certificate issuing mechanisms it should contract to an entity which can provide strong controls and strict adherence to a suitable CP/CPS. NENA foresees that other agencies such as police, fire and EMS agencies will need a similar Public Key Infrastructure (PKI), and it may be that, for an example, a county level agency provides the Certificate Authority for all agents in the county.

The identities, and the credentials, must be presented to gain access to ALL services and data in the ESInet.

6.2 PSAP Credentialing Agency

NENA will contract to operate the PCA. The PCA CP/CPS must be in conformance with the minimum standards to be provided in a future edition of this document. Any agency or agent may obtain a certificate from the PCA directly. As this is a similar function to the VESA in i2, it is expected that the VESA and the PCA are the same entity.