Technical Specification Group Services and System Aspects **TSGS#16(02)0220**
Meeting #16, Marco Island, Florida, 10-13 June 2002

SIG#10 Meeting LIF/SIG#10(02)-044
14-16 May 2002
Helsinki, Finland

TITLE – LIF TS 101 Specification Status update, for use in 3GPP TS 23.271 (Release 5)

SOURCE – LIF SIG (Standards Influencing Group)

The information contained in the attached contribution (the "Documentation") is provided for the purpose of defining, developing and promoting interoperable location services solutions that are open, simple and secure. The information contained in the Documentation is not binding on the Contributor. The Contributor reserves the right to add, amend and/or withdraw any of the information contained in the Documentation.

The Contributor's copyright in the Documentation has been licensed to Location Interoperability Forum Ltd. (the "Company") under the LIF Intellectual Property Agreement entered into by the Contributor with the Company. The Company licenses use of that copyright to members of Location Interoperability Forum ("LIF") who have entered a LIF Intellectual Property Agreement with the Company. That copyright licence is subject to the terms of that agreement.

Use of the Documentation is governed by the terms of the LIF Intellectual Property Agreement and the rules of LIF. The use or implementation of the Documentation may require use of materials covered by intellectual property rights owned by persons who have not authorised such use. Neither the Company, LIF or any of its members gives any assurance or otherwise with respect to the infringement, enforceability, existence or validity of any intellectual property rights in connection with the Documentation.

The Documentation is offered on an "as is" basis. No representation, warranty or condition, express or implied, statutory or otherwise, as to condition, quality, satisfactory quality, performance or fitness for purpose is given or assumed by the Company, LIF or any of its members in respect of the use or implementation of the Documentation and all those representations, warranties and conditions are excluded except to the extent that such exclusion is prohibited by law.

Dear 3GPP TSG SA Colleagues,

The Location Inter-operability Forum (LIF) appreciates your continued interest in our activity. We are delighted to have our MLP specification (LIF TS101) in your LCS Stage 2 Specification (3GPP TS23.271) as a reference.
We would like to update our progress in the specification work of LIF TS101.

Our target is to finalize and publish the version 3.0.0 of LIF TS101 which ensured to be fully conforming to 3GPP TS23.271 Release 5.
At the point in time of the closure of our LIF-SIG#10 meeting in Helsinki, Finland, LIF-SIG completed all editorial clean-ups of the TS101 to create its version 2.3.0. The time table towards the public release of version 3.0.0 is as follows;

- At the closure of LIF-SIG#10 (May 16, 2002) :
  Version 2.3.0 to enter for internal review period
  LIF-SIG attaches the current internal draft for your review
- June 3, 2002:
  Version 3.0.0 approval for the official public release

LIF SIG hopes the above schedule still can fit for your approval process of TS23.271 Release 5 (version 5.2.0) to be approved in your SA#16 Meeting for June 10-14.

Please feel free to ask if you have any questions.

Yours Respectfully,

Dan Greening
LIF SIG Chairperson

LIF DELIVERABLE COPYRIGHT NOTICE AND DISCLAIMER

© COPYRIGHT LOCATION INTEROPERABILITY FORUM LTD. 2001

Implementation of all or part of this LIF Deliverable may require licenses under or in respect of intellectual property rights including, without limitation, patent rights of a third party (who may or may not be a LIF Member). Neither Location Interoperability Forum Ltd. nor Location Interoperability Forum nor any of its members is responsible, and shall not be held responsible, in any manner for identifying or failing to identify any or all such third party intellectual property rights.

Mobile Location Protocol Specification

Abstract

The purpose of this specification is to define a simple and secure access method that allows Internet applications to query location information from a wireless network, irrespective of its underlying air interface technologies and positioning methods.

This specification covers the core of a Mobile Location Protocol that can be used by a location-based application to request MS location information from a location server (GMLC/MPC or other entity in the wireless network).

This specification has been prepared by LIF to provide a simple and secure API (Application Programmer's Interface) to the location server, but that also could be used for other kinds of location servers and entities in the wireless network.

The API is based on existing and well-known Internet technologies as HTTP, SSL/TLS and XML, in order to facilitate the development of location-based applications.

Contents

1 Revision History

1.0	23-Jan-2001	Sanjiv Bhatt, Motorola	Motorola, Nokia, Ericsson contribution to LIF
1.1	26-Jan-2001	Sanjiv Bhatt, Motorola	Updated after review in MLP adhoc committee in LIF #2 meeting
1.1.1	5-Nov-2001	Sanjiv Bhatt, Motorola	Updated after SIG#6 meeting
1.1.2	17-Nov-2001	Sanjiv Bhatt, Motorola	Updated after SIG#7 meeting
2.0.0	20-Nov-2001	Sanjiv Bhatt, Motorola	Final version (public release)
2.1.0	10-Mar-2002	Sanjiv Bhatt, Motorola	Updated after SIG#8 meeting
2.2.0	02-Apr-2002	Sanjiv Bhatt, Motorola	Updated after SIG#9 meeting
2.2.1	16-Apr-2002	Sanjiv Bhatt, Motorola	Updated before public review
2.3.0	15-May-2002	Sanjiv Bhatt, Motorola	Updated after SIG#10 meeting

2 Introduction

The Mobile Location Protocol (MLP) is an application-level protocol for getting the position of mobile stations (mobile phones, wireless personal digital assistants, etc.) independent of underlying network technology. The MLP serves as the interface between a Location Server and a Location Services (LCS) Client. This specification defines the core set of operations that a Location Server should be able to perform.

2.1 Abbreviations

ANSI	American National Standards Institute
DTD	Document Type Definition
GMLC	Gateway Mobile Location Center
GMT	Greenwich Mean Time
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
LCS	Location Services
MLC	Mobile Location Center
MLP	Mobile Location Protocol
MPC	Mobile Positioning Center
MS	Mobile Station
MSID	Mobile Station Identifier
SSL	Secure Socket Layer
TLS	Transport Layer Security
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTM	Universal Transverse Mercator
WGS	World Geodetic System
XML	Extensible Markup Language

2.2 Notational Conventions and Generic Grammar

The following rules are used throughout this specification to describe basic parsing constructs. ANSI X3.4-1986 defines the US-ASCII coded character set, see ref. □[5]

CR	= <US-ASCII CR, carriage return (13)>
LF	= <US-ASCII LF, linefeed (10)>
SP	= <US-ASCII SP, space (32)>

A set of characters enclosed in brackets ([]) is a one-character expression that matches any of the characters in that set. E.g., "[lcs]" matches either an "l", "c", or "s". A range of characters is indicated with a dash. E.g., "[a-z]" matches any lower-case letter.

The one-character expression can be followed by an interval operator, for example [a-zA-Z]{min,max} in which case the one-character expression is repeated at least min and at most max times. E.g., "[a-zA-Z]{2,4}" matches for example the strings "at", "Good", and "biG".

DTD Syntax Notation

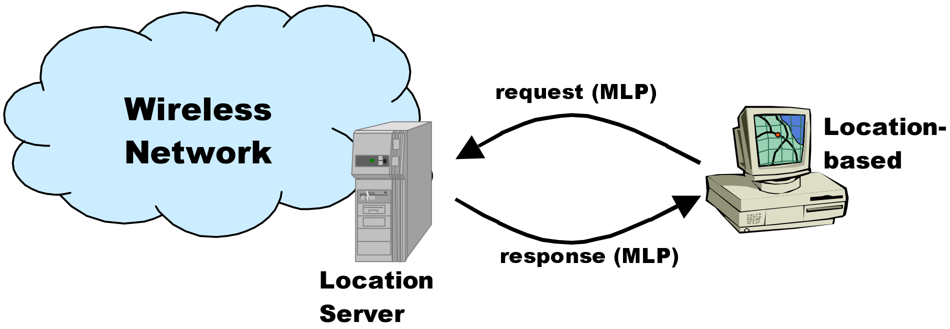
The table below describes the special characters and separators used in the DTDs defining the different services.

Character	Meaning
+	One or more occurrence
*	Zero or more occurrences
?	Optional
()	A group of expressions to be matched together
|	OR...as in, "this or that"
,	Strictly ordered. Like an AND

3 General

3.1 Overview

The Mobile Location Protocol (MLP) is an application-level protocol for querying the position of mobile stations independent of underlying network technology. The MLP serves as the interface between a Location Server and a location-based application.



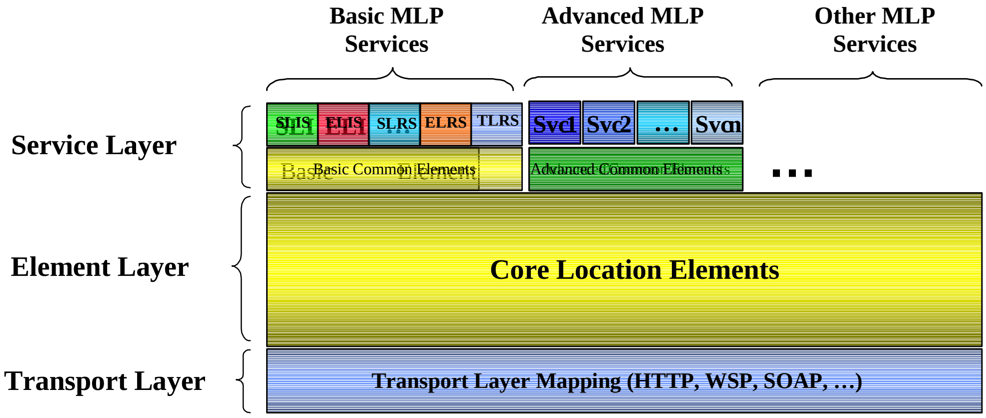
Possible realizations of a Location Server are the GMLC, which is the location server defined in GSM and UMTS, and the MPC, which is defined in ANSI standards. Since the location server should be seen as a logical entity, other implementations are possible.

In the most scenarios (except where explicitly mentioned) an LCS client initiates the dialogue by sending a query to the location server and the server responds to the query.

3.2 MLP structure

In our heterogeneous world, different devices may support different means of communication. A ubiquitous protocol for location services should support different transport mechanisms.

In MLP, the transport protocol is separated from the XML content. The following diagram shows a layered view of MLP.



On the lowest level, the transport protocol defines how XML content is transported. Possible MLP transport protocols include HTTP, WSP, SOAP and others.

The Element Layer defines all common elements used by the services in the service layer. Currently MLP defines the following set of DTDs making up the element layer of MLP:

MLP_ID.DTD	Identify Element Definitions
MLP_FUNC.DTD	Function Element Definitions
MLP_LOC.DTD	Location Element Definitions
MLP_SHAPE.DTD	Shape Element Definitions
MLP_QoP.DTD	Quality of Position Element Definitions
MLP_GSM_NET.DTD	GSM Network Parameters Element Definitions
MLP_CTXT	Context Element Definitions

The Service Layer defines the actual services offered by the MLP framework. Basic MLP Services are based on location services defined by 3GPP, and are defined by this specification. The "Advanced MLP Services" and "Other MLP Services" are additional services that either will be specified in other specifications or are specified by other fora that conform to the MLP framework.

Note: The boxes representing services in the Service Layer may contain more than one message. e.g. SLIS (Standard Location Immediate Service) consists of **slir** (Standard Location Immediate Request), **slia** (Standard Location Immediate Answer) and **slirep** (Standard Location Immediate Report) Standard Location Immediate Report messages. Messages for each service are listed in the table below

The Service Layer is divided into two sub-layers. The topmost defines the services mentioned in the previous paragraph. The lower sub-layer holds common elements which are specific for that group of services. If an element is common to more than one group of services then that element is defined in the element layer. The present specification specifies no element sub-layer.

There are a number of different possible types of location services. Each implementation of location server can select which services it wants/needs to support. The services are described in the table below.

Service	*Description*
Standard Location Immediate Service	This is a standard query service with support for a large set of parameters. This service is used when a (single) location response is required immediately (within a set time) or the request may be served by several asynchronous location responses (until a predefined timeout limit is reached). This service consists of the following messages: • Standard Location Immediate Request • Standard Location Immediate Answer • Standard Location Immediate Report
Emergency Location Immediate Service	This is a service used especially for querying of the location of a mobile subscriber that has initiated an emergency call. The response to this service is required immediately (within a set time). This service consists of the following messages: • Emergency Location Immediate Request • Emergency Location Immediate Answer
Standard Location Reporting Service	This is a service that is used when a mobile subscriber wants an LCS Client to receive the MS location. The position is sent to the LCS Client from the location server. Which application and its address are specified by MS or defined in the location server. This service consists of the following message: • Standard Location Report
Emergency Location Reporting Service	This is a service that is used when the wireless network automatically initiates the positioning at an emergency call. The position and related data is then sent to the emergency application from the location server. Which application and its address are defined in the location server. This service consists of the following message: • Emergency Location Report
Triggered Location Reporting Service	This is a service used when the mobile subscriber's location should be reported at a specific time interval or on the occurrence of a specific event. This service consists of the following messages: • Triggered Location Reporting Request • Triggered Location Reporting Answer • Triggered Location Report • Triggered Location Reporting Stop Request • Triggered Location Reporting Stop Answer

3.3 MLP extension mechanism

The MLP specification has been designed with extensibility in mind. Examples of design principles employed to achieve this include:

- Separate DTDs for definitions that are common to all messages, e.g. client address and shapes, so they can be re-used.
- Message extension mechanism allowing the addition of new messages (specific for the HTTP mapping). This mechanism works by specifying an entity parameter, '%extension;', referring to an extension DTD. The extension DTD MUST contain another entity parameter, '%extension.message', containing the definition of the extension as a string together with the actual parameters being added
- Parameter extension mechanism allowing the addition of additional parameters to existing messages. This mechanism works by specifying an entity parameter, '%extension;', referring to an extension DTD. The extension DTD MUST contain another entity parameter, '%extension.param', containing the definition of the extension as a string together with the actual messages being added.

Each extension parameters should have a vendor specific prefix in order to guarantee their uniqueness.

In order to use the extension, the extension DTD has to be explicitly referenced in the XML document.

The Location Server may ignore any extension that is not recognized and process the message as if the extension is not available.

Example 1: Message extension

```
<!—truckco_MLP_extension -->

<!ENTITY      % extension.~~–~~message    "| truckco_message">

<!ELEMENT     truckco_message          (truckco_data)>
<!ATTLIST     truckco_message
              ver CDATA                #FIXED "x.y.z">
```

```
<?xml version = "1.0" ?>
<!DOCTYPE svc_init SYSTEM "MLP_~~–~~SVC_INIT~~_221.DTD~~_230.DTD " [
  <!ENTITY % extension SYSTEM
      "http://www.truckco.com/truckco_MLP_extension.dtd">
  %extension;
]>
<svc_init ver=~~"2.2.1"~~"2.3.0">
  <hdr ver=~~"2.2.1"~~"2.3.0">
    ...
  </hdr>
  <truckco_message ver="x.y.z">
    <truckco_data>
      ...
    ~~    ~~</truckco_data>

  </truckco_message>
</svc_init>
```

Example 2: Parameter extension

```
<!–truckco_MLP_extension -->

<!ENTITY      % extension.~~–~~param       "truckco_extension">

<!ELEMENT     t~~T~~rucko_extension        (#PCDATA)>
```

```
<?xml version = "1.0" ?>
<!DOCTYPE svc_init SYSTEM "MLP_SVC_INIT~~_221.DTD~~_230.DTD" [
  <!ENTITY % extension SYSTEM
      "http://www.truckco.com/truckco_MLP_extension.dtd">
  %extension;
]>
<svc_init ver=~~"2.2.1"~~"2.3.0">
  <hdr ver=~~"2.2.1"~~"2.3.0">
    ...
  </hdr>
  <slir ver=~~"2.2.1"~~"2.3.0">
    ...
    ~~  ~~<truckco_extension>
    ~~  ~~  ...
    ~~  ~~</truckco_extension>
  </slir>
</svc_init>
```

4 Mobile Location Service Definitions

4.1 Transport Protocol Layer Definitions

MLP can be implemented using various transport mechanism as stated in section 3.2. The following mappings are specified for MLP:

Mapping	Section
HTTP	Appendix B - HTTP Mapping

4.2 Element Layer Definitions

4.2.1 Identity Element Definitions

```
<!-- MLP_ID -->

<!ELEMENT   msid                                            (#PCDATA)>
<!ATTLIST   msid
            type (MSISDN | IMSI | IMEI | MIN | MDN |        "MSISDN"
            EME_MSID | ASID | OPE_ID | IPV4 | IPV6 |
            SESSID)
            enc (ASC | ~~B64 | ~~CRP)                       "ASC">
<!ELEMENT   msid_range                                      (start_msid, stop_msid)>
<!ELEMENT   msids                                           (((msid, codeword?, session?)
                                                            | (msid_range, codeword*))+)>
<!ELEMENT   codeword                                        (#PCDATA)>
<!ELEMENT   esrd                                            (#PCDATA)>
<!ATTLIST   esrd
            type (NA)                                       "NA">
<!ELEMENT   esrk                                            (#PCDATA)>
<!ATTLIST   esrk
            type (NA)                                       "NA">
<!ELEMENT   session                                         (#PCDATA)>
<!ATTLIST   session
            type (APN | DIAL)                               #REQUIRED>
<!ELEMENT   start_msid                                      (msid)>
<!ELEMENT   stop_msid                                       (msid)>
```

Note: The type attributes of the msid elements that form the start_msid and stop_msid elements must be the same.

4.2.2 Function Element Definitions

```
<!-- MLP_FUNC -->
~~<!ELEMENT   add_info~~                         ~~(#PCDATA)>~~
<!ELEMENT   eme_event                            (eme_pos+)>
<!ATTLIST   eme_event
            eme_trigger (EME_ORG |               #REQUIRED>
            EME_REL)
<!ELEMENT   tlrr_event                           ( ms_action)>
<!ELEMENT   ms_action                            EMPTY>
<!ATTLIST   ms_action
            type (MS_AVAIL)                      #REQUIRED>
<!ELEMENT   interval                             (#PCDATA)>
<!ELEMENT   loc_type                             EMPTY>
<!ATTLIST   loc_type
            type (CURRENT | LAST                 "CURRENT">
            |CURRENT_OR_LAST | INITIAL)
~~<!ELEMENT   poserr~~                           ~~(result, add_info?, time)>~~
<!ELEMENT   prio                                 EMPTY>
<!ATTLIST   prio
            type (NORMAL | HIGH)                 "NORMAL">
```

```
<!ELEMENT   pushaddr                          (url, id?, pwd?)>
<!ELEMENT   req_id                            (#PCDATA)>
~~<!ELEMENT   result                            (#PCDATA)>~~
~~<!ATTLIST   result~~
~~            resid CDATA                       #REQUIRED>~~
<!ELEMENT   start_time                        (#PCDATA)>
<!ATTLIST   start_time
            utc_off CDATA                     "0000">
<!ELEMENT   stop_time                         (#PCDATA)>
<!ATTLIST   stop_time
            utc_off CDATA                     "0000">
~~<!ELEMENT   time                              (#PCDATA)>~~
~~<!ATTLIST   time~~
~~            utc_off CDATA                     "0000">~~
<!ELEMENT   url                               (#PCDATA)>
<!ELEMENT   time_remaining                    (#PCDATA)>
```

4.2.3 Location Element Definitions

```
<!-- MLP_LOC -->
<!ELEMENT   pos                               (msid, (pd | poserr), gsm_net_param?)>
<!ELEMENT   eme_pos                           (msid, (pd | poserr), esrd?, esrk?)>
<!ELEMENT   trl_pos                           (msid, (pd | poserr))>
<!ATTLIST   trl_pos
            trl_trigger (PERIODIC |           #REQUIRED>
            MS_AVAIL)
<!ELEMENT   pd                                (time, shape, (alt, alt_acc?)?, speed?,
                                              direction?, lev_conf?)>
<!ELEMENT   poserr                            (result, add_info?, time)>
<!ELEMENT   add_info                          (#PCDATA)>
<!ELEMENT   result                            (#PCDATA)>
<!ATTLIST   result
            resid CDATA                       #REQUIRED>
<!ELEMENT   time                              (#PCDATA)>
<!ATTLIST   time
            utc_off CDATA                     "0000">
<!ELEMENT   alt                               (#PCDATA)>
<!ELEMENT   alt_acc                           (#PCDATA)>
<!ELEMENT   direction                         (#PCDATA)>
<!ELEMENT   speed                             (#PCDATA)>
<!ELEMENT   lev_conf                          (#PCDATA)>
<!ELEMENT   geo_info                          (CoordinateReferenceSystem)>
<!ELEMENT   CoordinateReferenceSystem         (Identifier)>
<!ELEMENT   Identifier                        (code, codeSpace, edition)>
<!ELEMENT   code                              (#PCDATA)>
<!ELEMENT   codeSpace                         (#PCDATA)>
<!ELEMENT   edition                           (#PCDATA)>
```

Examples of geo_info encoding.

The encoding for WGS84 is:

```
<CoordinateReferenceSystem>
    <Identifier>
        <code>4326</code>
        <codeSpace>EPSG</codeSpace>
        <edition>6.1~~0~~</edition>
     </Identifier>
</CoordinateReferenceSystem>
```

The encoding for the Transverse Mercator coordinate system based on the OSGB1936 is:

```
<CoordinateReferenceSystem>
    <Identifier>
        <code>27700</code>
        <codeSpace>EPSG</codeSpace>
        <edition>6.1~~0~~</edition>
     </Identifier>
```

```
</CoordinateReferenceSystem>
```

Note that the GML V2.1.1 Implementation Specification is limited to use of only well-known CRSs, so this XML is currently abbreviated by a single attribute name and value:

```
srsName="http://www.opengis.net/gml/srs/epsg.xml#4326"
```

4.2.4 Shape Element Definitions

```
<!-- MLP_SHAPE -->
<!ELEMENT   shape                 (Point | LineString | Polygon | Box | CircularArea |
                                  CircularArcArea | EllipticalArea | GeometryCollection |
                                  MultiLineString | MultiPoint | MultiPolygon)>
<!ELEMENT   distanceUnit          (#PCDATA)>
<!ELEMENT   angularUnit           (#PCDATA)>
<!ELEMENT   angle                 (#PCDATA)>
<!ELEMENT   coord                 (X, Y?, Z?)>
<!ELEMENT   X                     (#PCDATA)>
<!ELEMENT   Y                     (#PCDATA)>
<!ELEMENT   Z                     (#PCDATA)>
<!ELEMENT   Point                 (coord)>
<!ATTLIST   Point
            gid ID                #IMPLIED
            srsName CDATA         #IMPLIED>
<!ELEMENT   LineString            (coord, coord+)>
<!ATTLIST   LineString
            gid ID                #IMPLIED
            srsName CDATA         #IMPLIED>
<!ELEMENT   Box                   (coord, coord)>
<!ATTLIST   Box
            gid ID                #IMPLIED
            srsName CDATA         #IMPLIED>
<!ELEMENT   LinearRing            (coord, coord, coord, coord*)>
<!ATTLIST   LinearRing
            gid ID                #IMPLIED
            srsName CDATA         #IMPLIED>
<!ELEMENT   Polygon~~t~~              (outerBoundaryIs, innerBoundaryIs*)>
<!ATTLIST   Polygon
            gid ID                #IMPLIED
            srsName CDATA         #IMPLIED>
<!ELEMENT   outerBoundaryIs       (LinearRing)>
<!ELEMENT   innerBoundaryIs       (LinearRing)>
<!ELEMENT   CircularArcArea       (coord, inRadius, outRadius, startAngle, stopAngle,
                                  angularUnit?, distanceUnit?)>
<!ATTLIST   CircularArcArea
            gid ID                #IMPLIED
            srsName CDATA         #IMPLIED>
<!ELEMENT   CircularArea          (coord, radius, distanceUnit?)>
<!ATTLIST   CircularArea
            gid ID                #IMPLIED
            srsName CDATA         #IMPLIED>
<!ELEMENT   EllipticalArea        (coord, angle, semiMajor, semiMinor, angularUnit,
                                  distanceUnit?)>
<!ATTLIST   EllipticalArea
            gid ID                #IMPLIED
            srsName CDATA         #IMPLIED>
<!ELEMENT   inRadius              (#PCDATA)>
<!ELEMENT   outRadius             (#PCDATA)>
<!ELEMENT   radius                (#PCDATA)>
<!ELEMENT   semiMajor             (#PCDATA)>
<!ELEMENT   semiMinor             (#PCDATA)>
<!ELEMENT   startAngle            (#PCDATA)>
<!ELEMENT   stopAngle             (#PCDATA)>
<!ELEMENT   GeometryCollection    (shape+)>
<!ATTLIST   GeometryCollection
            gid ID                #IMPLIED
            srsName CDATA         #IMPLIED>
~~<!ELEMENT   GeometryCollection    (shape+)>~~
~~<!ATTLIST   GeometryCollection~~
~~            gid ID                #IMPLIED~~
~~            srsName CDATA         #IMPLIED~~
```

```
<!ELEMENT   MultiLineString          (LineString+)>
<!ATTLIST   MultiLineString
            gid ID                   #IMPLIED
            srsName CDATA            #IMPLIED>
<!ELEMENT   MultiPoint               (Point+)>
<!ATTLIST   MultiPoint
            gid ID                   #IMPLIED
            srsName CDATA            #IMPLIED>
<!ELEMENT   MultiPolygon             ((Polygon| Box | CircularArea | CircularArcArea |
                                     EllipticalArea~~Ellipse~~)+)>
<!ATTLIST   MultiPolygon
            gid ID                   #IMPLIED
            srsName CDATA            #IMPLIED>
~~<!ELEMENT   startAngle               (angle)>~~
~~<!ELEMENT   stopAngle                (angle)>~~
```

4.2.5 Quality of Position Element Definitions

```
<!-- MLP_QoP -->
<!ELEMENT   eqop                            (resp_req,~~?,~~ resp_timer?, (ll_acc | hor_acc)?,
                                            alt_acc?, max_loc_age?)>
<!ELEMENT   qop                             ((ll_acc | hor_acc)?, alt_acc?)>
<!ELEMENT   ll_acc                          (#PCDATA)>
<!ELEMENT   hor_acc                         (#PCDATA)>
<!ELEMENT   max_loc_age                     (#PCDATA)>
<!ELEMENT   resp_req                        EMPTY>
<!ATTLIST   resp_req
            type (NO_DELAY | LOW_DELAY |    "DELAY_TOL">
            DELAY_TOL)
<!ELEMENT   resp_timer                      (#PCDATA)>
```

4.2.6 Network Parameters Element Definitions

```
<!-- MLP_GSM_NET -->
<!ELEMENT   gsm_net_param                   (cgi?, neid?, nmr?, ta?, lmsi?)>
<!ELEMENT   cgi                             (mcc, mnc, lac, cellid)>
<!ELEMENT   neid                            (vmscid | vlrid | (vmscid, vlrid))>
<!ELEMENT   vmscid                          (cc?, ndc?, vmscno)>
<!ELEMENT   vlrid                           (cc?, ndc?, vlrno)>
<!ELEMENT   nmr                             (#PCDATA)>
<!ELEMENT   mcc                             (#PCDATA)>
<!ELEMENT   mnc                             (#PCDATA)>
<!ELEMENT   ndc                             (#PCDATA)>
<!ELEMENT   cc                              (#PCDATA)>
<!ELEMENT   vmscno                          (#PCDATA)>
<!ELEMENT   vlrno                           (#PCDATA)>
<!ELEMENT   lac                             (#PCDATA)>
<!ELEMENT   cellid                          (#PCDATA)>
<!ELEMENT   ta                              (#PCDATA)>
<!ELEMENT   lmsi                            (#PCDATA)>
```

Note: The above table corresponds to GSM specific network element identifiers and network parameters. This information may be considered operator sensitive

4.2.7 Context Element Definitions

```
<!-- MLP_CTXT -->
<!ELEMENT   client                          (id, pwd?, serviceid?,
                                            ~~servicetype~~requestmode?)>
<!ELEMENT   sessionid                       (#PCDATA)>
```

```
<!ELEMENT   id                              (#PCDATA)>
<!ELEMENT   requestor                       (id, serviceid?)>
<!ELEMENT   pwd                             (#PCDATA)>
<!ELEMENT   serviceid                       (#PCDATA)>
<!ELEMENT   requestmode                     EMPTY>
<!ATTLIST   requestmode
            type (ACTIVE | PASSIVE)         "PASSIVE">
<!ELEMENT   subclient                       (id, pwd?, serviceid?)>
<!ATTLIST   subclient
            last_client (YES | NO)          "NO">
```

4.3 Service Layer Definitions

Each message may have two main parts, namely a context or header part and a body part. The body part consists of the request/answer and is described in sections 4.3.2 - 4.3.7. The context or header part consists of the information that identifies the client as defined in section 4.3.1.

4.3.1 Header Components

The **subclient** elements, if present, identify the ASPs, resellers and portals in the chain of service providers between the network and the end-user. The distinction between **client** and **subclient** elements is that the **client** element identifies the provider of the service that the Location Server has the initial relationship with, whereas the **subclient** elements identify the chain of other service providers up to the end-user. The final service provider in the chain is identified as such (last_client="YES"). On the other hand **requestor** is indicating the initiator of the location request, so in this context besides an ASP it could also be an MS subscriber who is asking the position of another target MS. The identity of the **requestor** may be an MSISDN or any other identifier identifying the initiator of the location request.

The **sessionid** element is used to represent the current session between the LCS Client and the Location Server. It may be used to replace the **id** and **pwd** elements, used in the context by the LCS Client to "login" to the Location Server, for the transactions that make up a session. For the first transaction of the session the LCS Client will need to "login" as usual. The Location Server may optionally return the **sessionid** in the response to this first transaction. If the Location Server does not return a **sessionid** the LCS Client will need to continue to "login" for subsequent transactions. The LCS Client can opt to ignore the **sessionid** if desired and continue to "login" for subsequent transactions.

The Location Server will decide the policy to be used to determine how the **sessionid** will be created and maintained. For example, the Location Server may determine the session as being just the transactions pertaining to a single service/MSID combination – this being restrictive and hence secure whilst still being useable, or the Location Server may allow the session to apply to a number of transactions between the Location Server and LCS Client. The Location Server may also allow the **sessionid** to be used for a particular period of time. The Location Server may also decide to return a different **sessionid** on each response, which the LCS Client will then use on the next transaction of the session.

The **sessionid** cannot be used instead of the **req_id** as this latter id refers to a set of reports that have been requested to be delivered from the Location Server to the LCS Client and do not form part of an existing LCS Client to Location Server connection. These reports are delivered by the Location Server "logging in" to the LCS Client for each one and the use of a **sessionid**, here would allow the security of the LCS Client to be breached.

4.3.1.1 Context DTD

```
<!-- MLP_HDR -->

<!ELEMENT    hdr                   ((client | sessionid | (client , sessionid)), subclient*,
                                   requestor?)>
<!ATTLIST    hdr
             ver CDATA             #FIXED ~~"2.2.1"~~"2.3.0">
```

Example 1: ASP as Initiator

```
<hdr ver=~~"2.2.1"~~"2.3.0">
  <client>
    <id>theasp</id>
    <pwd>thepwd</pwd>
    <serviceid>0005</serviceid>
    <requestmode type="PASSIVE"/>
  </client>
  <subclient last_client="YES">
    <id>thelastasp</id>
    <serviceid>0007</serviceid>
  </subclient>
  <requestor>
    <id>theoriginalasp</id>
    <serviceid>0003</serviceid>
  </requestor>
</hdr>
```

Example 2: MS as Initiator

```
<hdr ver=~~"2.2.1"~~"2.3.0">
  <client>
    <id>theasp</id>
    <pwd>thepwd</pwd>
    <serviceid>0005</serviceid>
    <requestmode type="ACTIVE"/>
  </client>
  <requestor>
    <id>461018765710</id>
  </requestor>
</hdr>
```

4.3.2 Standard Location Immediate Service

This is a standard service for requesting the location of one or more Mobile Subscribers. The service is used when a location response is required immediately (within a set time).

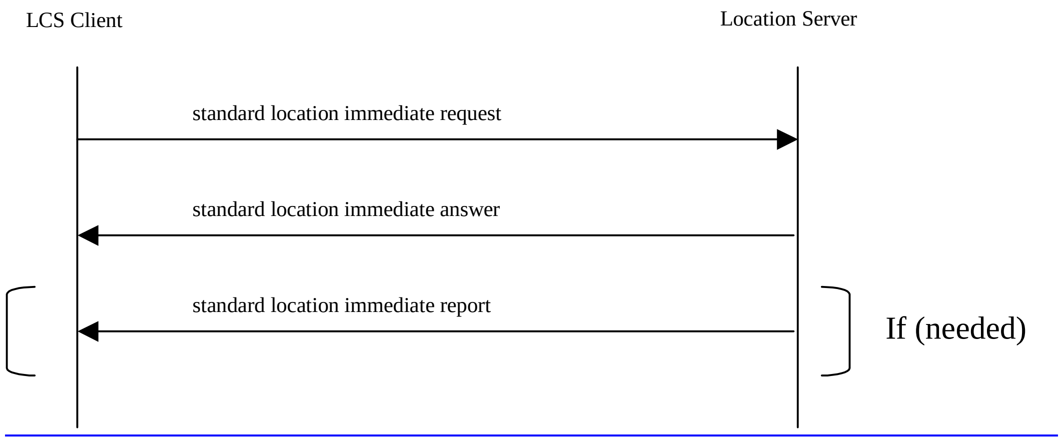
When a lot of positioning reports are requested, it may take an unacceptably long time to get the all responses from the network. If the Location Server supports it the LCS Client can define how to receive the location responses, either at a time with the response of the request, or individually using one or more connections initiated by the Location Server.

The extended service supports a number of different formats for describing the location of the mobile subscriber. It has also support for requesting a certain Quality of Service, Type of location and priority.

The service consists of the following messages:

- Standard Location Immediate Request
- Standard Location Immediate Answer
- Standard Location Immediate Report

The following message flow encapsulates this service:


LCS Client
Location Server
standard location immediate request
standard location immediate answer
standard location immediate report
If (needed)

4.3.2.1 Standard Location Immediate Request DTD

```
<!-- MLP_SLIR -->

<!ENTITY     % extension.param         "">

<!ELEMENT    slir                      ((msids | (msid, codeword?, gsm_net_param)+),
                                       eqop?, geo_info?, loc_type?, prio?, pushaddr?
                                       %extension.param;)>

<!ATTLIST    slir
             ver CDATA                 #FIXED "2.3~~0~~.0"
             res_type (SYNC | ASYNC)   "SYNC">
```

Example

```
<slir ver=~~"2.2.1"~~"2.3.0" res_type="SYNC">
  <msids>
    <msid type="IPV4">93.10.0.250</msid>
    <msid_range>
      <start_msid>
        <msid>461018765710</msid>
      </start_msid>
      <stop_msid>
        <msid>461018765712</msid>
      </stop_msid>
    </msid_range>
    <msid type="ASID">441728922342</msid>
    <msid_range>
      <start_msid>
        <msid>461018765720</msid>
      </start_msid>
      <stop_msid>
        <msid>461018765728</msid>
      </stop_msid>
    </msid_range>
  </msids>
  <eqop>
    <resp_req type="LOW_DELAY" />
    <hor_acc>1000</hor_acc>
  </eqop>
  <geo_info>
             <CoordinateReferenceSystem>
        <Identifier>
          <code>4004</c~~C~~ode>
          <codeSpace>EPSG</codeSpace>
          <edition>6.1~~0~~</edition>
         </Identifier>
    </CoordinateReferenceSystem>
  </geo_info>
  <loc_type type="CURRENT_OR_LAST" />
  <prio type="HIGH" />
</slir>
```

4.3.2.2 Standard Location Immediate Answer DTD

```
<!-- MLP_SLIA -->

<!ENTITY     % extension.param           "">

<!ELEMENT    slia                        ((pos+ | req_id | (result, add_info?))
                                         %extension.param;)>

<!ATTLIST    slia
             ver CDATA                   #FIXED "2.3~~0~~.0">
```

Example 1: Successful positioning of multiple subscribers

```
<slia ver=~~"2.2.1"~~"2.3.0" >
  <pos>
    <msid>461011334411</msid>
    <pd>
      <time utc_off="+0200">20000623134453</time>
      <shape>
```

```
                  <CircularArea srsName="www.epsg.org#4004">
             <coord>
              <X>301628.312</X>
              <Y>451533.431</Y>
             </coord>
             <radius>240</radius>
        </CircularArea>
      </shape>
    </pd>
  </pos>
  <pos>
    <msid>461018765710</msid>
    <pd>
      <time utc_off="+0300">20000623134454</time>
      <shape>
                 <CircularArea srsName="www.epsg.org#4004">
            <coord>
              <X>301228.302</X>
              <Y>865633.863</Y>
            </coord>
          <radius>570</radius>
        </CircularArea>

      </shape>
    </pd>
  </pos>
  <pos>
    <msid>461018765711</msid>
    <pd>
      <time utc_off="+0300">20000623110205</time>
      <shape>
                 <CircularArea srsName="www.epsg.org#4004">
            <coord>
              <X>781234.322</X>
              <Y>762162.823</Y>
            </coord>
            <radius>15</radius>
          </CircularArea>

      </shape>
    </pd>
  </pos>
  <pos>
    <msid>461018765712</msid>
    <poserr>
      <result resid="10">QOP NOT ATTAINABLE</result>
      <time>20000623134454</time>
    </poserr>
  </pos>
</slia>
```

Example 2: Service not supported

```
<slia ver=~~"2.2.1"~~"2.3.0" res_type= "SYNC">
  <result resid="108">SERVICE NOT SUPPORTED</result>
  <add_info>~~<!DOCTYPE slia SYSTEM "MLP_SLIA_221.DTD~~_230.DTD~~">~~'slir' is not
supported by the location server</add_info>
</slia>
```

4.3.2.3 Standard Location Immediate Report DTD

```
<!-- MLP_SLIREP -->

<!ENTITY      % extension.param          "">

<!ELEMENT     slirep                     (req_id, pos+ %extension.param;)>
```

```
<!ATTLIST    slirep
             ver CDATA                #FIXED ~~"2.2.1"~~"2.3.0">
```

Example

```
<slirep ver=~~"2.2.1"~~"2.3.0">
  <req_id>25267</req_id>
  <pos>
    <msid type="IPV6">10:A1:45::23:B7:89</msid>
    <pd>
      <time utc_off="+0300">20000813010423</time>
      <shape>
                <CircularArea srsName="www.epsg.org#4326">
           <coord>
              <X>301628.312</X>
              <Y>451533.431</Y>
            </coord>
            <radius>15</radius>
        </CircularArea>
               </shape>
    </pd>
  </pos>
  ~~<time utc_off="+0200">20000813010423</time>~~
</slirep>
```

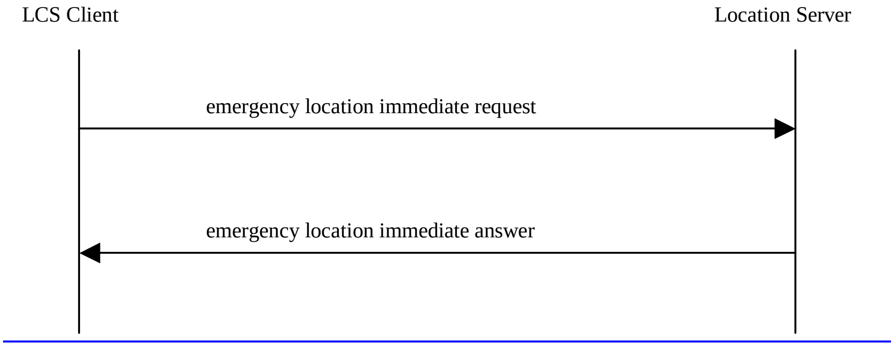
4.3.3 Emergency Location Immediate Service

The emergency location immediate service is used to retrieve the position of a mobile subscriber that is involved in an emergency call or have initiated an emergency service in some other way.

The service consists of the following messages:

- Emergency Location Immediate Request
- Emergency Location Immediate Answer

The following message flow encapsulates this service:


LCS Client
Location Server
emergency location immediate request
emergency location immediate answer

4.3.3.1 Emergency Location Immediate Request DTD

```
<!-- MLP_EME_LIR -->

<!ENTITY      % extension.param          "">

<!ELEMENT     eme_lir                    ((msids | (msid, gsm_net_param)+), qop?,
                                         geo_info?, loc_type? %extension.param;)>

<!ATTLIST     eme_lir
              ver CDATA                  #FIXED ~~"2.2.1"~~"2.3.0">
```

Example

```
<eme_lir ver=~~"2.2.1"~~"2.3.0">
  <msids>
    <msid type="EME_MSID">520002-51-431172-6-06</msid>
  </msids>
  <geo_info>
           <CoordinateReferenceSystem>
        <Identifier>
          <code>4004</c~~C~~ode>
          <codeSpace>EPSG</codeSpace>
          <edition>6.1~~0~~</edition>
         </Identifier>
     </CoordinateReferenceSystem>
  </geo_info>
  <loc_type type="CURRENT_OR_LAST" />
```

```
</eme_lir>
```

4.3.3.2 Emergency Location Immediate Answer DTD

```
<!-- MLP_EME_LIA -->

<!ENTITY      % extension.param    "">

<!ELEMENT     eme_lia              ((eme_pos+ | (result, add_info?)) %extension.param;)>
<!ATTLIST     eme_lia
              ver CDATA            #FIXED ~~"2.2.1"~~"2.3.0">
```

Example

```
<eme_lia ver=~~"2.2.1"~~"2.3.0">
  <eme_pos>
    <msid type="EME_MSID">520002-51-431172-6-06</msid>

    <pd>
      <time utc_off="+0300">20000623134453</time>
      <shape>
        <CircularArea srsName="www.epsg.org#4004">
          <coord>
            <X>N301628.312</X>
            <Y>-451533.431</Y>
          </coord>
          <radius>15</radius>
        </CircularArea>

      </shape>
    </pd>
    <esrk>7839298236</esrk>
  </eme_pos>
</eme_lia>
```

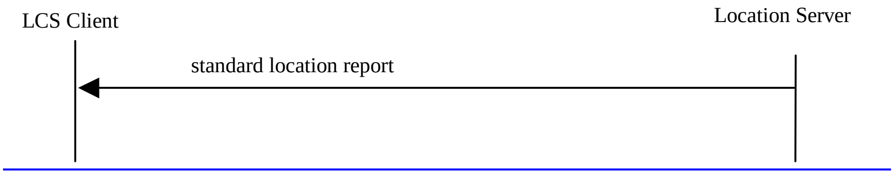
4.3.4 Standard Location Reporting Service

When a mobile subscriber wants an LCS client to receive the MS location a standard location report is generated. The LCS Client that the location report should be sent to is specified by MS or defined within the Location Server.

The service consists of the following message:

- Standard Location Report

The following message flow encapsulates this service:



4.3.4.1 Standard Location Report DTD

```
<!-- MLP_SLREP -->
<!ENTITY     % extension                       "">
<!ENTITY     % extension.param                 "">

<!ELEMENT    slrep                             (pos+ %extension.param;)>
<!ATTLIST    slrep
             ver CDATA                         #FIXED ~~"2.2.1"~~"2.3.0">
```

Example

```
<slrep ver=~~"2.2.1"~~"2.3.0">
  <pos>
    <msid>461011678298</msid>
    <pd>
      <time>20000813010423</time>
      <shape>
        <CircularArea srsName="www.epsg.org#4004">
          <coord>
            <X>301628.312</X>
            <Y>451533.431</Y>
          </coord>
          <radius>15</radius>
        </CircularArea>
      </shape>
    </pd>
  </pos>
</slrep>
```

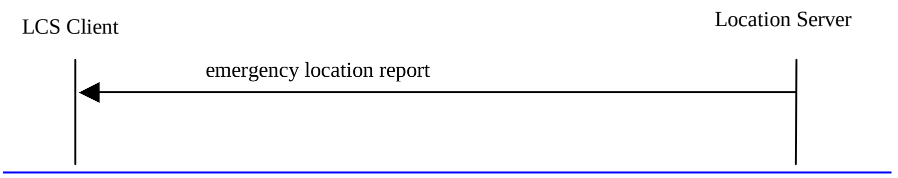
4.3.5 Emergency Location Reporting Service

If the wireless network spontaneously initiates a positioning when a user initiates or releases an emergency call, an emergency location report is generated. The application(s) that the emergency location report should be sent to is defined within the location server. Data as required geographical format and address to application is also defined within the location server.

The service consists of the following message:

- Emergency Location Report

The following message flow encapsulates this service:



4.3.5.1 Emergency Location Report DTD

```
<!-- MLP_EMEREP -->

<!ENTITY              % extension.param       "">
```

```
<!ELEMENT        emerep              (eme_event %extension.param;)>
<!ATTLIST        emerep
                 ver CDATA           #FIXED ~~"2.2.1"~~"2.3.0">
```

Example

```
<emerep ver=~~"2.2.1"~~"2.3.0">
  <eme_event eme_trigger="EME_ORG">
    <eme_pos>
      <msid>461011678298</msid>
      <pd>
        <time utc_off="+0300">20000623010003</time>
        <shape>
          <CircularArea srsName="www.epsg.org#4004">
            <coord>
              <X>301628.312</X>
              <Y>451533.431</Y>
            </coord>
            <radius>15</radius>
          </CircularArea>
        </shape>
      </pd>
    </eme_pos>
  </eme_event>
</emerep>
```

4.3.6 Triggered Location Reporting Service

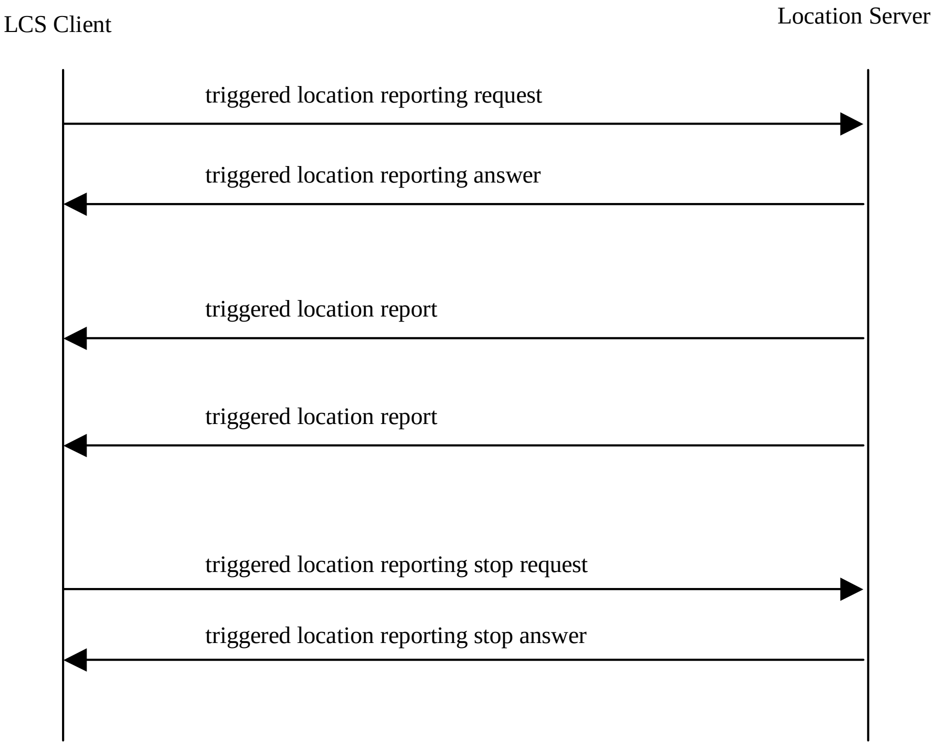
The triggered location reporting service is used when an application wants the position of the list of MS to be tracked. The triggers could be:

- The periodicity time defined by an interval
- An MS action, defined as the event "UE available" in 3GPP TS 23.271 rel. 4 [ref. 11].

The report will be triggered when one of the pre-defined MS's actions occurred or the time interval elapses. The service consists of the following messages:

- Triggered Location Reporting Request
- Triggered Location Reporting Answer
- Triggered Location Report
- Triggered Location Reporting Stop Request
- Triggered Location Reporting Stop Answer

The following message flow encapsulates this service:


LCS Client
Location Server
triggered location reporting request
triggered location reporting answer
triggered location report
triggered location report
triggered location reporting stop request
triggered location reporting stop answer

Note: *It is the intention that Triggered services will support entering or leaving an area in future releases. An area may be defined as a specified geographical area, a city or locale, a country or a network. Other triggers that may be supported are specific events not yet defined, such a subscriber being in proximity to a friend in a FriendFinder application. Other events are FFS within 3GPP and are targeted for rel.5.*

4.3.6.1 Triggered Location Reporting Request DTD

```
<!-- MLP_TLRR -->

<!ENTITY          % extension.param      "">

<!ELEMENT         tlrr                   (msids, interval?, start_time?, stop_time?,
                                         tlrr_event?, qop?, geo_info?, pushaddr?,
                                         loc_type?, prio? %extension.param;)>

<!ATTLIST         tlrr
                  ver CDATA              #FIXED ~~"2.2.1"~~"2.3.0">
```

The following rules apply to the use of ‘start_time’, ‘stop_time’, ‘interval’ and ‘tlrr_event’:

- TLRR with ‘interval’ is interpreted as a request for periodic location reports, and TLRR with ‘tlrr_event’ is interpreted as a request for a location report on the occurrence of a specific event. ‘interval’ and ‘tlrr_event’ can be combined. When neither ‘interval’ nor ‘tlrr_event’ is specified in TLRR, Location Server MUST reject the request with an error indication ‘106’ to the client.

- If no START_TIME is specified reporting starts immediately.

- If no STOP_TIME is specified the reporting will occur until explicitly canceled with ‘Triggered Location Stop Request’ or a time out occurs (depending on system configuration).

- If START_TIME is ‘older’ than current time the Location Server MUST reject the request with an error indication ‘110’ to the client.

- If STOP_TIME is ‘older’ than current time the Location Server MUST reject the request with an error indication ‘110’ to the client.

- If STOP_TIME is earlier than START_TIME the implementation MUST reject the request with an error indication ‘110’ to the client.

- If STOP_TIME is equal to START_TIME the Location Server MUST return a single location report to the client at the specified time. Any interval specified MUST be ignored.

Example 1: TLRR for periodic location reports during a specified period

```
<tlrr ver=~~"2.2.1"~~"2.3.0">
  <msids>
    <msid>461011678298</msid>
  </msids>
  <interval>00003000</interval>
  <start_time utc_off="+0300">20011003112700</start_time>
  <stop_time utc_off="+0300">20011003152700</stop_time>
  <qop>
    <hor_acc>100</hor_acc>
  </qop>
  <geo_info>
    <CoordinateReferenceSystem>
      <Identifier>
        <code>4326</c~~C~~ode>
        <codeSpace>EPSG</codeSpace>
        <edition>6.1~~0~~</edition>
      </Identifier>
    </CoordinateReferenceSystem>
  </geo_info>
  <pushaddr>
    <url>http://location.application.com</url>
  </pushaddr>
  <loc_type type="CURRENT" />
  <prio type="HIGH" />
</tlrr>
```

Example 2: TLRR for single location report at a specified time. ‘stop_time’ is specified equal to ‘start_time’.

```
<tlrr ver=~~"2.2.1"~~"2.3.0">
  <msids>
    <msid>461011678298</msid>
  </msids>
  <interval>00003000</interval>
  <start_time utc_off="+0300">20011003112700</start_time>
  <stop_time utc_off="+0300">20011003112700</stop_time>
  <qop>
    <hor_acc>100</hor_acc>
  </qop>
  <geo_info>
    <CoordinateReferenceSystem>
      <Identifier>
        <code>4004</c~~C~~ode>
        <codeSpace>EPSG</codeSpace>
        <edition>6.1~~0~~</edition>
      </Identifier>
    </CoordinateReferenceSystem>
  </geo_info>
  <pushaddr>
    <url>http://location.application.com</url>

  </pushaddr>
  <loc_type type="CURRENT" />
  <prio type="HIGH" />
</tlrr>
```

Example 3: TLRR for a location report on the occurrence of a MS_AVAIL event after a specified time.

```
<tlrr ver=~~"2.2.1"~~"2.3.0">
  <msids>
    <msid>461011678298</msid>
  </msids>
```

```
      <start_time utc_off="+0300">20011003112700</start_time>
      <tlrr_event>
    <ms_action type="MS_AVAIL"/>
    </tlrr_event>
      <qop>
        <hor_acc>100</hor_acc>
      </qop>
      <geo_info>
        <CoordinateReferenceSystem>
          <Identifier>
            <code>4326</c~~C~~ode>
            <codeSpace>EPSG</codeSpace>
            <edition>6.1~~0~~</edition>
          </Identifier>
        </CoordinateReferenceSystem>

      </geo_info>
      <pushaddr>
        <url>http://location.application.com</url>

      </pushaddr>
      <loc_type type="CURRENT" />
      <prio type="HIGH" />
    </tlrr>
```

4.3.6.2 Triggered Location Reporting Answer DTD

```
<!-- MLP_TLRA -->

<!ENTITY     % extension.param        "">

<!ELEMENT    tlra                     ((req_id | (result, add_info?))
                                      %extension.param;)>
<!ATTLIST    tlra
             ver CDATA                #FIXED ~~"2.2.1"~~"2.3.0">
```

Example 1: TLRA if corresponding TLRR was successful

```
    <tlra ver=~~"2.2.1"~~"2.3.0">
      <req_id>25293</req_id>
    </tlra>
```

Example 2: TLRA if corresponding TLRR was in error

```
    <tlra ver="2.2~~0~~.1~~0~~">
      <result resid="4">UNKNOWN SUBSCRIBER</result>
    </tlra>
```

4.3.6.3 Triggered Location Report DTD

```
<!-- MLP_TLREP -->

<!ENTITY     % extension.param    "">

<!ELEMENT    tlrep                (req_id, trl_pos+, time_remaining?
                                  %extension.param;)>
<!ATTLIST    tlrep
             ver CDATA            #FIXED ~~"2.2.1"~~"2.3.0">
```

Example

```
<tlrep ver=~~"2.2.1"~~"2.3.0">
  <req_id>25267</req_id>
  <trl_pos trl_trigger="PERIODIC">
    <msid>461011678298</msid>
    <pd>
      <time utc_off="+0300">20000813010423</time>
      <shape>
        <CircularArea srsName="www.epsg.org#4326">
          <coord>
            <X>301628.312</X>
            <Y>451533.431</Y>
          </coord>
          <radius>15</radius>
        </CircularArea>
      </shape>
    </pd>
  </trl_pos>
  <time_remaining>00010000</time_remaining>
</tlrep>
```

4.3.6.4 Triggered Location Reporting Stop Request DTD

```
<!-- MLP_TLRSR -->

<!ENTITY          % extension.param         "">

<!ELEMENT         tlrsr                     (req_id %extension.param;)>
<!ATTLIST         tlrsr
                  ver CDATA                 #FIXED ~~"2.2.1"~~"2.3.0">
```

Example

```
<tlrsr ver=~~"2.2.1"~~"2.3.0">
  <req_id>25293</req_id>
</tlrsr>
```

4.3.6.5 Triggered Location Reporting Stop Answer DTD

```
<!-- MLP_TLRSA -->

<!ENTITY      % extension.param         "">

<!ELEMENT     tlrsa                     (req_id, result, add_info? %extension.param;)>
<!ATTLIST     tlrsa
              ver CDATA                 #FIXED ~~"2.2.1"~~"2.3.0">
```

Example

```
<tlrsa ver=~~"2.2.1"~~"2.3.0">
```

```
    <req_id>25293</req_id>
    <result resid="0">OK</result>
  </tlrsa>
```

4.3.7 General Error Message Definition

When an LCS client attempts to invoke a service not defined in this specification, the location server will return a General Error Message. Sending a general error message (GEM) is no proper solution **by itself** because it can not always be expected that the client will understand this (MLP) response message, since - by sending an invalid request - the client showed that it may not be familiar with the proper set of MLP services. So additional error indications may be described in the different transport layer mappings.

```
<!-- MLP_GEM -->

<!ELEMENT   gem                           (result, add_info?)>
<!ATTLIST   gem
            ver CDATA                     #~~fixed ~~FIXED ~~"2.2.1"~~"2.3.0">

<!ENTITY    % mlp_loc.dtd                 SYSTEM "MLP_LOC~~_221.DTD~~_230.DTD">
%mlp_loc.dtd;
```

Example

```
  <gem ver=~~"2.2.1"~~"2.3.0">
    <result resid="108">SERVICE NOT SUPPORTED</result>
    <add_info>
      The server does not support a service named ’skir’
    </add_info>
  </gem>
```

5 Elements and attributes in DTD

5.1 add_info

Description:	
A text string containing additional information about a certain result.	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	`<add_info>EVENT</add_info>`
Note: -	

5.2 alt

Description:	
The altitude of the MS in meters in respect of the ellipsoid which is used to be define the coordinates	
Type:	Element
Format:	Char String
Defined values:	[+|-] [0-9]+
Default value:	-
Example:	`<alt>1200</alt>`
Note: This element is present if altitude is possible to attain by the used positioning method.	

5.3 alt_acc

Description:	
Accuracy of altitude in meters	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<alt_acc>200</alt_acc>`
Note: -	

5.4 angle

Description:	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-
Example:	`<angle>24.30</angle>`
Note: -	

5.5 angularUnit

Description:	
The angularUnit defines the unit for any angular value used in the shape description. For example the startAngle value in the CircularArcArea will be defined by this unit. If this unit is not included in a shape definition the distance unit defined in the CRS will be used.	
Type:	Element
Format:	C~~c~~har String
Defined values:	Degrees~~, Radians~~
	Radians
Default value:	`Degrees`
Example:	`<angularUnit>Degrees</angularUnit>`
Note:.	

5.6 Box

Description:	
The Box element is used to encode extents	
Type:	Element
Format:	
Defined values:	-
Default value:	-

Example:	`<Box srsName="www.epsg.org#4004" gid="some_thing">`
	`  <coord>`
	`    <X>301628.312</X>`
	`    <Y>451533.431</Y>`
	`  </coord>`
	`  <coord>`
	`    <X>311628.312</X>`
	`    <Y>461533.431</Y>`
	`  </coord>`
	`</Box>`

Note: -	

5.6.1 gid

Description:	
The gid is of XML attribute type ID and is used for references to elements within a single XML document. It allows XML technologies such as XPointer and xref to be used..	
Type:	attribute
Format:	Char String
Defined values:	
Default value:	
Example:	`<Box srsName="www.epsg.org#4004" gid="some_thing">`
Note:.This attribute is optional and is on all shape elements	

5.6.2 srsName

Description:	
srsName is a short hand method of defining the CoordinateReferenceSystem. It is a URI datatype that contains the codeSpace and code values, which are defined the same as in the CoordinateReferenceSystem..	
Type:	attribute
Format:	Char String
Defined values:	
Default value:	`www.epsg.org/#4326`
Example:	`<Box srsName="www.epsg.org/#4326">`
Note:.This attribute is optional and is on all shape elements. If the srsName is not included the WGS84 CRS is assumed.	

~~5.6~~5.7 cc

Description:	
Specifies the country code.	
Type:	Element
Format:	Char String
Defined values:	3 digits e.g. 355 for Albania
Default value:	-
Example:	`<cc>355</cc>`
Note:	

~~5.7~~5.8 cellid

Description:	
Identifies the Cell Identity	
Type:	Element
Format:	Char String
Defined values:	0-65535
Default value:	-
Example:	`<cellid>546</cellid>`
Note:	

5.9 CircularArcArea

Description:	
An arc is defined by a point of origin with one offset angle and one uncertainty angle plus one inner radius and one uncertainty radius.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-

Example:	`<CircularArcArea srsName="www.epsg.org#4004" gid="some_thing">`
	`  <coord>`
	`    <X>301628.312</X>`
	`    <Y>451533.431</Y>`
	`  </coord>`
	`  <inRadius>280</inRadius>`
	`  <outRadius>360</outRadius>`
	`  <startAngle>5</startAngle>`
	`  <stopAngle>240</stopAngle>`
	`</CircularArcArea>`

Note:	

5.9.1 gid

See section 5.6.1.

5.9.2 srsName

see section 5.6.2.

5.10 CircularArea

Description:	
The set of points on the ellipsoid, which are at a distance from the point of origin less than or equal to “r”.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-

Example:	`<CircularArea srsName="www.epsg.org#4004" gid="some_thing">`
	`  <coord>`
	`    <X>301628.312</X>`
	`    <Y>451533.431</Y>`
	`  </coord>`
	`  <radius>240</radius>`
	`</CircularArea>`

Note:	

5.10.1 gid

See section 5.6.1.

5.10.2 srsName

See section 5.6.2.

~~5.8~~5.11 code

Description:	
This is the unique identifier for the Coordinate ReferenceSystem as used by the authority cited in codeSpace	
Type:	Element
Format:	Char String
Defined values:	
Default value:	
Example:	`<code>4326</code>`
Note: .	

~~5.9~~5.12 codespace

Description:	
The codeSpace is the authority which is responsible for the definition of the coordinate reference systems.	
Type:	Element
Format:	~~URI~~Char String
Defined values:	
Default value:	www.epsg.org/…
Example:	`<codeSpace>www.epsg.org</codeSpace>`
Note:.	

~~5.10~~5.13 codeword

Description:	
Codeword is an access code defined per MS, used to protect location information of MS against unwanted location request. Only location requests with the correct codeword of a target MS are accepted.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-
Example:	`<codeword>0918a7cb</codeword>`
Note: An error shall be returned if the number of codewords is not equal to the number of msid in an msid_range.	

~~5.11~~

~~5.12~~5.14 distanceUnit

Description:	
The distanceUnit defines the linear unit for any distance used in the shape description. For example the radius value in the CircularArea will be defined by this unit. If this unit is not included in a shape definition the distance unit defined in the CRS will be used.	
Type:	Element
Format:	C~~c~~har String
Defined values:	meter, kilometer, foot, …
Default value:	`meter`
Example:	`<distanceUnit>surveyfoot</distanceUnit>`
Note:.	

~~5.13~~5.15 direction

Description:	
Specifies the direction, in degrees, that a positioned MS is moving in.	
Type:	Element
Format:	Char String
Defined values:	`0-360`
Default value:	-
Example:	`<direction>120</direction>`
Note: This element is present if direction is possible to attain by the used positioning method.	

~~5.14~~

~~5.15~~

5.16 edition

Description:	
The edition defines which version of the CRS database defined by thie codeSpace authority is used..	
Type:	Element
Format:	C~~c~~har String
Defined values:	
Default value:	
Example:	`<edition>6.0</edition>`
Note:.	

5.17 EllipticalArea

Description:	
IA set of points on the ellipsoid, which fall within or on the boundary of an ellipse. This ellipse has a semi-major axis of length r1 oriented at angle A (0 to 180°) measured clockwise from north and a semi-minor axis of length r2.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-

Example:	`<EllipticalArea srsName="www.epsg.org#4004" gid="some_thing">`
	`  <coord>`
	`    <X>301628.312</X>`
	`    <Y>451533.431</Y>`
	`  </coord>`
	`  <angle>240</angle>`
	`  <semiMajor>150</semiMajor>`
	`  <semiMinor>275</semiMinor>`
	`  <angularUnit>degrees</angularUnit>`
	`</EllipticilArea>`

Note:	

5.17.1 gid

See section 5.6.1.

5.17.2 srsName

see section 5.6.2.

~~5.17~~5.18 eme_event

Description:	
Specifies the events that initiated the positioning of the MS at an emergency call.	
Type:	Element
Format:	-
Defined values:	-
Default value:	-
Example:	`<eme_event eme_trigger="EME_ORG">`
Note: -	

~~5.17.1~~5.18.1 eme_trigger

Description:		
Specifies the trigger that initiated the positioning of the MS at an emergency call.		
Type:	Attribute	
Format:	Char string	
Defined values:	`EME_ORG`	An emergency service user originated an emergency call
	`EME_REL`	An emergency service user released an emergency call
Default value:	-	
Example:	`<eme_event eme_trigger="EME_ORG">`	
Note: -		

~~5.18~~5.19 esrd

Description:	
This element specifies Emergency Services Routing Digits (ESRD).	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	`<esrd>761287612582</esrd>`
Note: -	

~~5.18.1~~5.19.1 type

Description:		
Defines the origin of the ESRD		
Type:	Attribute	
Format:	Char string	
Defined values:	`NA`	Indicates that the ERSD is defined as the North American ESRD (NA-ERSD). NA-ESRD is a telephone number in the North American Numbering Plan that can be used to identify a North American emergency services provider and it's associated Location Services client. The NA-ESRD also identifies the base station, cell site or sector from which a North American emergency call originates
Default value:	`NA`	
Example:	`<esrd type="NA">12345678</ersd>`	
Note: Currently only NA is specified. It is expected that other origins will be specified in the future		

~~5.19~~5.20 esrk

Description:	
This element specifies the Services Routing Key (ESRK).	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	`<esrk>928273633343</esrk>`
Note: -	

~~5.19.1~~5.20.1 type

Description:		
Defines the origin of the ESRK		
Type:	Attribute	
Format:	Char string	
Defined values:	`NA`	Indicates that the ERSK is defined as the North American ESRK (NA-ERSK). NA-ESRK is a telephone number in the North American Numbering Plan that is assigned to an emergency services call for the duration of the call. The NA-ESRK is used to identify (e.g. route to) both the emergency services provider and the switch currently serving the emergency caller. During the lifetime of an emergency services call, the NA-ESRK also identifies the calling subscriber.
Default value:	`NA`	
Example:	`<esrk type="NA">12345678</ersk>`	
Note: Currently only NA is specified. It is expected that other origins will be specified in the future		

~~5.20~~

~~5.21gid~~

~~**Description:**~~	
~~The gid is of XML id type and is use for references to elements within a single XML document. It allows XML technologies such as XPointer and xref to be used..~~	
~~**Type:**~~	~~attribute~~
~~**Format:**~~	~~ID~~
~~**Defined values:**~~	
~~**Default value:**~~	
~~**Example:**~~	~~`<gid>"2"</gid>`~~
~~**Note:**.This attribute is optional and is on all shape elements~~	

5.21 GeometryCollection

Description:	
A collection of shapes.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-

Example:	`<GeometryCollection srsName="www.epsg.org#4004" gid="some_thing">`
	`  <shape>`
	`    ...`
	`  </shape>`
	`</GeometryCollection>`

Note:	

5.21.1 gid

See section 5.6.1.

5.21.2 srsName

See section 5.6.2.

5.22 hor_acc

Description:	
Requested horizontal accuracy in meters	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<hor_acc>200</hor_acc>`
Note: -	

5.23 id

Description:	
A string defining the name of a registered user performing a location request. In an answer the string represents the name of a location server.	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	`<id>TheTruckCompany</id>`
Note: - This element is implementation specific.	

5.24 in_radius

Description:	
The inner radius is the geodesic distance (in meters) between center of the circle (that the arc is a part of) and arc closest to the center	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<in_radius>100</in_radius>`
Note: If the inner radius is 0 (zero) the area described represents a circle sector.	

5.25 interval

<table>
<tr><td colspan="3"><b>Description:</b></td></tr>
<tr><td colspan="3">Specifies the interval between two responses in case of a TLRR that indicates timer controlled, periodic responses.</td></tr>
<tr><td><b>Type:</b></td><td colspan="2">Element</td></tr>
<tr><td rowspan="6"><b>Format:</b></td><td colspan="2">Char string

The interval is expressed as <code>ddhhmmss</code> where:</td></tr>
<tr><td><b>String</b></td><td><b>Description</b></td></tr>
<tr><td><code>dd</code></td><td>Number of days between responses</td></tr>
<tr><td><code>hh</code></td><td>Number of hours between responses</td></tr>
<tr><td><code>mm</code></td><td>Number of minutes between responses</td></tr>
<tr><td><code>ss</code></td><td>Number of seconds between responses</td></tr>
<tr><td><b>Defined values:</b></td><td colspan="2">-</td></tr>
<tr><td><b>Default value:</b></td><td colspan="2">-</td></tr>
<tr><td><b>Example:</b></td><td colspan="2"><code>&lt;interval>00010000</interval&gt;</code></td></tr>
<tr><td colspan="3"><b>Note: -</b></td></tr>
</table>

5.26 lac

Description:	
Identifies the Location Area Code	
Type:	Element
Format:	Char String
Defined values:	1-65535
Default value:	-
Example:	`<lac>234</lac>`
Note:	

5.27 lev_conf

Description:	
This parameter indicates the probability in percent that the MS is located in the position area that is returned.	
Type:	Element
Format:	Char String
Defined values:	`0-100`
Default value:	-
Example:	`<lev_conf>80</lev_conf>`
Note: -	

5.28 LinearRing

Description:	
A linear ring is a closed, simple piece-wise linear path which is defined by a list of coordinates that are assumed to be connected by straight line segments.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-

Example:	`<LinearRing srsName="www.epsg.org#4004" gid="some_thing">`
	`  <coord>`
	`    <X>301628.312</X>`
	`    <Y>451533.431</Y>`
	`  </coord>`
	`  <coord>`
	`    <X>401628.312</X>`
	`    <Y>481533.431</Y>`
	`  </coord>`
	`  <coord>`
	`    <X>332628.312</X>`
	`    <Y>461533.431</Y>`
	`  </coord>`
	`  <coord>`
	`    <X>301628.312</X>`
	`    <Y>451533.431</Y>`
	`  </coord>`
	`</LinearRing>`

Note:	

5.28.1 gid

See section 5.6.1.

5.28.2 srsName

See section 5.6.2.

5.29 LineString

Description:	
A line string is a piece-wise linear path which is defined by a list of coordinates that are assumed to be connected by straight line segments.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-

Example:	`<LineString srsName="www.epsg.org#4004" gid="some_thing">`
	`  <coord>`
	`    <X>301628.312</X>`
	`    <Y>451533.431</Y>`
	`  </coord>`
	`  <coord>`
	`    <X>401628.312</X>`
	`    <Y>481533.431</Y>`
	`  </coord>`
	`  <coord>`
	`    <X>332628.312</X>`
	`    <Y>461533.431</Y>`
	`  </coord>`
	`</LineString>`

Note:	

5.29.1 gid

See section 5.6.1.

5.29.2 srsName

See section 5.6.2.

~~5.28~~5.30 ll_acc

Description:	
Longitude and latitude accuracy in seconds.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-
Example:	`<ll_acc>7.5</ll_acc>`
Note: -	

~~5.29~~5.31 lmsi

Description:	
A local identity allocated by the VLR to a given subscriber for internal management of data in the VLR as defined in 29.002	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-
Example:	`<lmsi>2344512344565</lmsi>`
Note: -	

~~5.30~~5.32 loc_type

Description:	
Defines the type of location requested.	
Type:	Element
Format:	Void~~-~~
Defined values:	-
Default value:	-
Example:	`<loc_type type="INITIAL" />`
Note: -	

~~5.30.1~~5.32.1 type

Description:		
Defines the type of location requested		
Type:	Attribute	
Format:	Char string	
Defined values:	CURRENT	After a location attempt has successfully delivered a location estimate, the location estimate is known as the current location at that point in time.
	LAST	The current location estimate is generally stored in the network until replaced by a later location estimate and is known as the last known location. The last known location may be distinct from the initial location., i.e. more recent.
	CURRENT_OR_LAST	If a location attempt has successfully delivered, the current location is returned. Otherwise the last known location stored in the network is returned.
	INITIAL	In an originating emergency call, the location estimate at the commencement of the call set-up is known as the initial location.
Default value:	CURRENT	
Example:	`<loc_type type="INITIAL" />`	
Note: -		

~~5.31~~5.33 max_loc_age

Description:	
This states the maximum allowable age in seconds of a location sent ~~in~~ as a response to a location request. This location information may have been cached somewhere in the system from a previous location update.	
Type:	Element
Format:	Char string ~~representing seconds~~
Defined values:	Maximum number of seconds (must be >= 0)
Default value:	Implementation specific.
Example:	`<max_loc_age>3600</max_loc_age>`
Note: -	

~~5.32~~5.34 mcc

Description:	
Specifies the mobile country code (MCC).	
Type:	Element
Format:	Char String
Defined values:	3 digits, e.g. 234 for the UK
Default value:	-
Example:	`<mcc>234</mcc>`
Note:	

~~5.33~~5.35 mnc

Description:	
Specifies the mobile network code.	
Type:	Element
Format:	Char string
Defined values:	Up to 3 digits e.g. 15 for Vodafone
Default value:	-
Example:	`<mnc>215</mnc>`
Note: -	

~~5.34~~5.36 ms_action

Description:	
Specifies the trigger that initiated the positioning of the MS.	
Type:	Element
Format:	Void~~-~~
Defined values:	-
Default value:	-
Example:	`<ms_action type="MS_AVAIL" />`
Note: -	

~~5.34.1~~5.36.1 type

Description:		
Specifies the trigger that initiated the positioning of the MS.		
Type:	Attribute	
Format:	Char string	
Defined values:	`MS_AVAIL`	The positioning is triggered by the MS available notification when the MS regains radio connection with the network if the connection was previously lost. For more information refer to 3GPP TS 23.271 rel. 4.
Default value:	-	
Example:	`<ms_action type="MS_AVAIL" />`	
Note: -		

~~5.35~~5.37 msid

Description:	
This element represents an identifier of a mobile subscriber	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	`<msid>460703057640</msid>`
Note: -	

~~5.35.1~~5.37.1 type

Description:		
Type of identifier for the mobile subscriber		
Type:	Attribute	
Format:	Char string	
Defined values:	`MSISDN`	Mobile Station International ISDN Number
	`IMSI`	International Mobile Subscriber Identity
	`IMEI`	International Mobile station Equipment Identity
	`MIN`	Mobile Identification Number
	`MDN`	Mobile Directory Number
	`EME_MSID`	Emergency MSID
	`ASID`	Anonymous Subscriber Identity
	`IPV4`	Mobile station IP address (Version 4)
	`OPE_ID`	Operator specific Identity
	`IPV6`	Mobile station IP address (Version 6)
	`SESSID`	Session identifier relating to the user, which may be anonymous
Default value:	`MSISDN`	
Example:	`<msid type="IMSI">`	
Note: -		

~~5.35.2~~5.37.2 enc

Description:		
Type of encoding for MSID identifier for the mobile subscriber		
Type:	Attribute	
Format:	Char string	
Defined values:	ASC	Normal textual format
	~~B64~~	~~Base 64 encoding~~
	CRP	Encrypted format: In some countries the Network Operator (where is placed the Location Server) isn't allowed to send to a LCS the private information of an MS like MSISDN. The Network Operator can send out to LCS the Encrypted MSID, since only the Network Operator is the only entity able to decode this information, the LCS will be never able to break the privacy of the MS.
Default value:	ASC	
Example:	`<msid type="IMSI" enc="`~~B64~~`ASC">`	
Note: -		

5.38 MultiLineString

Description:	
A collection of line strings.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-

Example:	`<MultiLineString srsName="www.epsg.org#4004" gid="some_thing">`
	`  <LineString>`
	`    ...`
	`  </LineString>`
	`</MultiLineString>`

Note:	

5.38.1 gid

See section 5.6.1.

5.38.2 srsName

~~5.36~~see section 5.6.2.

5.39 MultiPoint

Description:	
A collection of points.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-

Example:	`<MultiPoint srsName="www.epsg.org#4004" gid="some_thing">`
	`  <Point>`
	`    ...`
	`  </Point>`
	`</MultiPoint>`

Note:	

5.39.1 gid

See section 5.6.1.

5.39.2 srsName

See section 5.6.2.

5.40 MultiPolygons

Description:	
A collection of polygons.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-

Example:	`<MultiPolygon srsName="www.epsg.org#4004" gid="some_thing">`
	`  <Polygon>`
	`    ...`
	`  </Polygon>`
	`</MultiPolygon>`

Note:	

5.40.1 gid

See section 5.6.1.

5.40.2 srsName

see section 5.6.2.

5.41 ndc

Description:	
Specifies the network destination code.	
Type:	Element
Format:	Char string
Defined values:	Up to 4 digits e.g. 7785 for Vodafone
Default value:	-
Example:	`<ndc>215</ndc>`
Note: -	

~~5.37~~5.42 nmr

Description:	
Network specific measurement result for the target MS.	
Type:	Element
Format:	Char string
Defined values:	`For examples see relevant standards documents. (GSM 04.08 – rel.98 section 10.5.2.20)`
Default value:	-
Example:	
Note: This element remains to be defined**.**	

~~5.38~~5.43 radius

Description:	
The uncertainty radius is the radius (in distanceUnit) of the uncertainty; this is the geodesic distance between the arc and the position point.	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<radius>850</radius>`
Note: -	

5.44 Point

Description:	
A geographic 2D coordinate	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-

Example:	`<Point srsName="www.epsg.org#4004" gid="some_thing">`
	`  <coord>`
	`    <X>301628.312</X>`
	`    <Y>451533.431</Y>`
	`  </coord>`
	`</Point>`

Note:	

5.44.1 gid

See section 5.6.1.

5.44.2 srsName

See section 5.6.2.

5.45 Polygon

Description:	
A connected surface. Any pair of points in the polygon can be connected to one another by a path. The boundary of the Polygon is a set of LinearRings. We distinguish the outer (exterior) boundary and the inner (interior) boundaries; the LinearRings of the interior boundary cannot cross one another and cannot be contained within one another.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-

Example:	`<Polygon srsName="www.epsg.org#4004" gid="some_thing">`
	`  <outerBoundaryIs>`
	`    ...`
	`  </outerBoundaryIs >`
	`</Polygon>`

Note:	

5.45.1 gid

See section 5.6.1.

5.45.2 srsName

See section 5.6.2.

~~5.39~~5.46 prio

Description:	
Defines the priority of a location request	
Type:	Element
Format:	Void~~-~~
Defined values:	-
Default value:	-
Example:	`<prio />`
Note: -	

~~5.39.1~~5.46.1 type

Description:		
Defines the priority of a location request		
Type:	Attribute	
Format:	Char string	
Defined values:	`NORMAL`	The request is handled with normal priority
	`HIGH`	The request is handled with high priority
Default value:	`NORMAL`	
Example:	`<prio type="HIGH" />`	
Note: -		

~~5.40~~5.47 pwd

Description:	
The password for the registered user performing a location request. In an answer the string represents the password for a location server.	
Type:	Element
Format:	Char string
Defined values:	
Default value:	-
Example:	`<pwd>the5pwd</pwd>`
Note: -	

~~5.41~~5.48 outRadius

Description:	
The radius of a circle furthest away from the position in a CircularArcArea. The value is in the distanceUnit	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<outRadius>120</outRadius>`
Note: -	

~~5.42~~5.49 req_id

Description:	
Unique identification of a request	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	`<req_id>435.23.01</req_id>`
Note: -	

~~5.43~~5.50 resp_req

Description:	
This attribute represents response time requirement.	
Type:	Element
Format:	~~Char string~~Void
Defined values:	-
Default value:	-
Example:	`<resp_req type="NO_DELAY" />`
Note: -	

~~5.43.1~~5.50.1 type

Description:		
This attribute represents response time requirement		
Type:	Attribute	
Format:	Char String	
Defined values:	`NO_DELAY`	No delay: The server should immediately return any location estimate that it currently has.
	`LOW_DELAY`	Low delay: Fulfilment of the response time requirement takes precedence over fulfilment of the accuracy requirement.
	`DELAY_TOL`	Delay tolerant: Fulfilment of the accuracy requirement takes precedence over fulfilment of the response time requirement.
Default value:	`DELAY_TOL`	
Example:	`<resp_req />`	
Note: - The interpretation of these parameters is defined in 3GPP specifications 22.071 and 29.002. The use of this element together with resp_timer is for further study.		

~~5.44~~5.51 resp_timer

Description:	
Defines a timer for the response time within which the current location should be obtained and returned to the LCS Client.	
Type:	Element
Format:	Char String ~~The time is expressed as mmss where:~~ ~~mm~~ ~~Minutes~~ ~~ss~~ ~~Seconds~~
Defined values:	`Maximum number of seconds (must be >= 0)`~~-~~
Default value:	The default value is defined in the location server and will be implementation specific
Example:	`<resp_timer>`~~0010~~`45</resp_timer>`
Note: - The use of this element together with resp_req is for further study	

~~5.45~~5.52 result

Description:	
A text string indicating the result of the request or an individual positioning	
Type:	Element
Format:	Char string
Defined values:	See chapter 6.1
Default value:	-
Example:	`<result resid=0>OK</result>`
Note: -	

~~5.45.1~~5.52.1 resid

Description:	
This attribute represents a numeric representation of a result message	
Type:	Attribute
Format:	Char String
Defined values:	[0-9]+ See chapter 6.1
Default value:	-
Example:	`<result resid=0>OK</result>`
Note: -	

~~5.46~~5.53 semiMajor

Description:	
Specifies the length of the semi-major axis of an ellipse in meters.	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<semiMajor>560</semiMajor>`
Note: -	

~~5.47~~5.54 semiMinor

Description:	
Specifies the length of the semi-minor axis of an ellipse in meters.	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<semiMinor>560</semiMinor>`
Note: -	

~~5.48~~5.55 serviceid

Description:	
Specifies an id that is used by an entity to identify the service or application that is accessing the network.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-
Example:	`<serviceid>0005</serviceid>`
Note:	

~~5.49srsName~~

~~**Description:**~~	
~~srsName is a short hand method of defining the CoordinateReferenceSystem. It is a URI datatype that contains the codeSpace and code values, which are defined the same as in the CoordinateReferenceSystem..~~	
~~**Type:**~~	~~attribute~~
~~**Format:**~~	~~char string~~
~~**Defined values:**~~	
~~**Default value:**~~	~~`www.epsg.org/#4326`~~
~~**Example:**~~	~~`<srsName>"www.epsg.org/#4326"</srsName>`~~
~~**Note:**.This attribute is optional and is on all shape elements. If the srsName is not included the WGS84 CRS is assumed.~~	

~~5.50~~5.56 requestmode

Description:	
Defines the type of the service that has been requested by the ASP.	
Type:	Element
Format:	~~-~~Void
Defined values:	-
Default value:	-
Example:	`<requestmode />`
Note:	

~~5.50.1~~5.56.1 type

Description:		
Defines the type of the service that has been requested by the ASP		
Type:	Attribute	
Format:	Char string	
Defined values:	`PASSIVE`	The service is one that is not directly initiated by the user.
	`ACTIVE`	The service is one that the user is initiating personally.
	`SESSION`	The Service is one that has an established session with the user
Default value:	`PASSIVE`	
Example:	`<requestmode type="ACTIVE" />`	
Note: The default value is set to PASSIVE, as this is likely to be the one that is most restrictively defined by the user.		

~~5.51~~5.57 session

Description:	
This element should be presented in location request when the LCS Client is making has an active session with the User Equipment, this will be either the number called by the UE or the APN on which the UE established the session.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-
Example:	`<session>447073100177</session>`
Note: This information may be required for privacy validation of the location request by the VMSC, SGSN or MSC server	

~~5.51.1~~5.57.1 type

Description:		
Defines the type of the session that is established between the User Equipment and LCS Client		
Type:	Attribute	
Format:	Char string	
Defined values:	`APN`	Access Point Name.
	`DIAL`	The number dialed by the user to access the LCS client.
Default value:	-	
Example:	`<session type="DIAL" />`	
Note:		

~~5.52~~5.58 sessionid

Description:	
Specifies an id that can be used by an entity to support privacy ~~To~~ mechanisms, a sessionid may replace the need to use an ID and PWD to use the location services. In a request when a client and sessionid are present together the session id may indicate the number dialed by the end user to access the service or the APN through which the original session was established that initiated the service. In an answer it indicates the sessionid that the entity can use on subsequent requests. In this case ~~T~~the s~~S~~ession~~-~~id could be ~~is~~ a generated alphanumeric string and can be time-limited.	
Type:	Element
Format:	Char String
Defined values:	-
Default value:	-
Example:	`<sessionid>34eg6.876.76h4</sessionid>`
Note:	

~~5.53~~5.59 speed

Description:	
The speed of the MS in m/s.	
Type:	Element
Format:	Char String
Defined values:	[0-9]+
Default value:	-
Example:	`<speed>23</speed>`
Note: This element is present if speed is possible to attain by the used positioning method.	

~~5.54~~5.60

~~5.55~~5.61 start_time

Description:		
This element defines the absolute start time in a range of times.		
Type:	Element	
Format:	Char String The time is expressed as yyyyMMddhhmmss where:	
	String	**Description**
	yyyy	Year
	MM	Month
	dd	Day
	hh	Hours
	mm	Minutes
	ss	Seconds
Defined values:	-	
Default value:	-	
Example:	`<start_time>20010630142810</start_time>`	
Note: -		

~~5.55.1~~5.61.1 utc_off

Description:	
Specifies the UTC offset in hours and minutes. Positive values indicate time zones east of Greenwich.	
Type:	Attribute
Format:	Char string
Defined values:	[+|-]0000-1400
Default value:	-
Example:	`<start_time utc_off="+0200">20000813010423</start_time>`
Note: utc_off is specified as ’HHMM’, where ’HH’ can range between 0-14 and ’MM’ between ’0-59’. All other values shall result in error 105, ’Format error’.	

~~5.56~~5.62 stop_time

<table>
<tr><th colspan="2">Description:</th></tr>
<tr><td colspan="2">This element defines the absolute stop time in a range of times.</td></tr>
<tr><td><b>Type:</b></td><td>Element</td></tr>
<tr><td><b>Format:</b></td><td>Char String

The time is expressed as yyyyMMddhhmmss where:
<table>
<tr><th>String</th><th>Description</th></tr>
<tr><td><code>yyyy</code></td><td>Year</td></tr>
<tr><td><code>MM</code></td><td>Month</td></tr>
<tr><td><code>dd</code></td><td>Day</td></tr>
<tr><td><code>hh</code></td><td>Hours</td></tr>
<tr><td><code>mm</code></td><td>Minutes</td></tr>
<tr><td><code>ss</code></td><td>Seconds</td></tr>
</table></td></tr>
<tr><td><b>Defined values:</b></td><td>-</td></tr>
<tr><td><b>Default value:</b></td><td>-</td></tr>
<tr><td><b>Example:</b></td><td><stop_time>20010630142810</stop_time></td></tr>
<tr><td colspan="2"><b>Note: -</b></td></tr>
</table>

~~5.56.1~~5.62.1 utc_off

<table>
<tr><th colspan="2">Description:</th></tr>
<tr><td colspan="2">Specifies the UTC offset in hours and minutes. Positive values indicate time zones east of Greenwich.</td></tr>
<tr><td><b>Type:</b></td><td>Attribute</td></tr>
<tr><td><b>Format:</b></td><td>Char string</td></tr>
<tr><td><b>Defined values:</b></td><td>[+|-]0000-1400</td></tr>
<tr><td><b>Default value:</b></td><td>-</td></tr>
<tr><td><b>Example:</b></td><td><stop_time utc_off="+0200">20000813010423</stop_time></td></tr>
<tr><td colspan="2"><b>Note:</b> utc_off is specified as ’HHMM’, where ’HH’ can range between 0-14 and ’MM’ between ’0-59’. All other values shall result in error 105, ’Format error’.</td></tr>
</table>

~~5.57~~5.63 subclient

<table>
<tr><th colspan="2">Description:</th></tr>
<tr><td colspan="2">Identifies the ASPs, resellers and portals in the chain of service providers between the network and the end-user</td></tr>
<tr><td><b>Type:</b></td><td>Element</td></tr>
<tr><td><b>Format:</b></td><td>-</td></tr>
<tr><td><b>Defined values:</b></td><td>-</td></tr>
<tr><td><b>Default value:</b></td><td>-</td></tr>
<tr><td><b>Example:</b></td><td><subclient last_client="NO">
  <id>TheASP</id>
  <serviceid>0006</serviceid>
</subclient></td></tr>
<tr><td colspan="2"><b>Note:</b> -</td></tr>
</table>

~~5.57.1~~5.63.1 last_client

Description:		
Identifies whether the SUBCLIENT is the last one in the chain or not		
Type:	Attribute	
Format:	Char String	
Defined values:	YES	This is the last client – the one that the end-user is actually communicating with
	NO	This is not the last client
Default value:	`NO`	
Example:	`<subclient last_client="YES">`	
Note: -		

~~5.58~~5.64 ta

Description:	
This Radio Access Network element that can arguably be used to offer enhanced positioning.	
Type:	Element
Format:	Char string
Defined values:	`0-62`
Default value:	`0`
Example:	`<ta>3</ta>`
Note: Further Information regarding this element can be found in the relevant GSM Specifications	

~~5.59~~5.65 time

Description:		
In a location answer this element indicates the time when the positioning was performed.		
Type:	Element	
Format:	Char String The time is expressed as `yyyyMMddhhmmss` where:	
	String	**Description**
	`yyyy`	Year
	`MM`	Month
	`dd`	Day
	`hh`	Hours
	`mm`	Minutes
	`ss`	Seconds
Defined values:	-	
Default value:	-	
Example:	`<time>20010630142810</time>`	
Note: -		

~~5.59.1~~5.65.1 utc_off

Description:	
Specifies the UTC offset in hours and minutes. Positive values indicate time zones east of Greenwich.	
Type:	Attribute
Format:	Char string
Defined values:	[+|-]0000-1400
Default value:	-
Example:	`<time utc_off="+0200">20000813010423</time>`
Note: utc_off is specifed as 'HHMM', where 'HH' can range between 0-14 and 'MM' between '0-59'. All other values shall result in error 105, 'Format error'.	

~~5.60~~5.66 time_remaining

<table>
<tr><th colspan="2">Description:</th></tr>
<tr><td colspan="2">Defines the time remaining until the location server terminates the current triggered location service. The time for which the service is valid is either specified by the client using start time and stop time, or is a network operator specific default value where no s stop time is defined or where the stop time exceeds the allowed value by the location server involved.</td></tr>
<tr><td><b>Type:</b></td><td>Element</td></tr>
<tr><td><b>Format:</b></td><td>Char String

The time is expressed as ddhhmmss where:
<table>
<tr><th>String</th><th>Description</th></tr>
<tr><td>dd</td><td>Day</td></tr>
<tr><td>hh</td><td>Hours</td></tr>
<tr><td>mm</td><td>Minutes</td></tr>
<tr><td>ss</td><td>Seconds</td></tr>
</table></td></tr>
<tr><td><b>Defined values:</b></td><td>-</td></tr>
<tr><td><b>Default value:</b></td><td>The default value is defined in the location server</td></tr>
<tr><td><b>Example:</b></td><td><time_remaining>00010000</time_remaining></td></tr>
<tr><td colspan="2"><b>Note: -</b></td></tr>
</table>

~~5.61~~5.67 trl_pos

Description:	
Specifies the position of the MS at a triggered location report.	
Type:	Element
Format:	-
Defined values:	-
Default value:	-

Example:	`<tlr_pos trl_trigger="PERIODIC">`
	`    <msid>4711</msid>`
	`    <poserr>`
	`        <result resid=1>SYSTEM FAILURE</result>`
	`        <time utc_off="0100">20011127104532</time>`
	`    </poserr>`
	`</trl_pos>`

Note: -	

~~5.61.1~~5.67.1 trl_trigger

Description:		
Specifies the trigger that initiated the positioning of the MS at a triggered location report.		
Type:	Attribute	
Format:	Char string	
Defined values:	PERIODIC	The positioning is triggered when the periodical timer expired
	MS_AVAIL	The positioning is triggered by the MS presence notification
Default value:	-	
Example:	`<tlr_pos trl_trigger="PERIODIC">`	
Note: -		

~~5.62~~5.68 url

Description:	
Specifies the location to which a response to a TLRR or an asynchronous SLIR should be sent to	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	`<url>http://location.client.com/Response/</url>`
Note: - URL is part of pushaddr element which may also contain id and pwd. These elements are used by the LCS Client to inform the Location Server what credentials to use when 'pushing' a location report to the LCS Client in case of an asynchronous service.	

~~5.63~~5.69 vlrno

Description:	
Uniquely specifies a VLR within a network.	
Type:	Element
Format:	Char String
Defined values:	In GSM this is the Global Title address. The Global Title is in the same format as an E.164 number.
Default value:	-
Example:	<vlrno>1541154871</vlrno>
Note:	

~~5.64~~5.70 vmscno

Description:	
Uniquely specifies a VMSC within a network.	
Type:	Element
Format:	Char String
Defined values:	In GSM this is the Global Title address. The Global Title is in the same format as an E.164 number.
Default value:	-
Example:	<vmscno>1541154871</vmscno>
Note:	

~~5.65~~5.71 X

Description:	
The first ordinate in a coordinate system	
Type:	Element
Format:	Char string
Defined values:	~~(+|-)[0-9+.0-9+]~~-
Default value:	-
Example:	<X>33498.23</X>
Note: -	

~~5.66~~5.72 Y

Description:	
Second ordinate in a coordinate.system. This is optional if it is a linear coordinate system.	
Type:	Element
Format:	Char string
Defined values:	-
Default value:	-
Example:	<Y>33498.23</Y>
Note: -	

~~5.67~~5.73 Z

Description:	
third ordinate in a coordinate.system. This is optional if it is a 2D coordinate system.	
Type:	Element
Format:	Char string
Defined values:	~~(+|-)[0-9+.0-9+]~~-
Default value:	-
Example:	`<Z>33498.23</Z>`
Note: -	

~~5.68~~5.74 Service attributes

~~5.68.1~~5.74.1 res_type

Description:		
Defines a response type at the Standard Location Immediate Service. This attribute applies to the Standard Immediate Location Request message.		
Type:	Attribute	
Format:	Char string	
Defined values:	SYNC	An LCS Client requests to receive the location response in one response
	ASYNC	An LCS Client request to receive the location responses one by one using some connections initiated by the LCS Server
Default value:	SYNC	
Example:	`<slir ver=`~~"2.2.1"~~`"2.3.0" res_type="SYNC">`	
Note: -		

~~5.68.2~~5.74.2 ver

Description:	
Defines the version of the location protocol. This attribute is valid for ALL messages	
Type:	Element
Format:	Char string
Defined values:	[0-9].[0-9].[0-9]
Default value:	-
Example:	`<slia ver=`~~`"2.2.1"`~~`"2.3.0">`
Note: -	

6 Result codes

6.1 Result codes

This table defines the result codes that indicate the result of the request or individual positioning. The error codes are divided in ranges:

0	-	99	Location server specific errors
100	-	199	Request specific errors
200	-	299	Network specific errors
300	-	499	Reserved for future use
500	-	599	Vendor specific errors

Note: For privacy reasons it might be needed to not report certain specific errors. In this case it is up to the implementation or configuration of the location server which errors will be reported.

Resid	Slogan	Description
0	OK	No error occurred while processing the request.
1	SYSTEM FAILURE	The request can not be handled because of a general problem in the location server or the underlying network.
2	UNSPECIFIED ERROR	An unspecified error used in case none of the other errors applies. This can also be used in case privacy issues prevent certain errors from being presented
3	UNAUTHORIZED APPLICATION	The requesting location-based application is not allowed to access the location server or a wrong password has been supplied.
4	UNKNOWN SUBSCRIBER	Unknown subscriber. The user is unknown, i.e. no such subscription exists.
5	ABSENT SUBSCRIBER	Absent subscriber. The user is currently not reachable.
6	POSITION METHOD FAILURE	Position method failure. The location service failed to obtain the user's position.
101	CONGESTION IN LOCATION SERVER	The request can not be handled due to congestion in the location server.
102	CONGESTION IN MOBILE NETWORK	The request can not be handled due to congestion in the mobile network.
103	UNSUPPORTED VERSION	The Location server does not support the indicated protocol version.
104	TOO MANY POSITION ITEMS	Too many position items have been specified in the request.
105	FORMAT ERROR	A protocol element in the request has invalid format. The invalid element is indicated in ADD_INFO.
106	SYNTAX ERROR	The position request has invalid syntax. Details may be indicated in ADD_INFO.
107	PROTOCOL ELEMENT NOT SUPPORTED	A protocol element specified in the position request is not supported by the Location Server. The element is indicated in ADD_INFO.
108	SERVICE NOT SUPPORTED	The requested service is not supported in the Location Server. The service is indicated in ADD_INFO.
109	ELEMENT ATTRIBUTE NOT SUPPORTED	A protocol element attribute is not supported in the Location Server. The attribute is indicated in ADD_INFO.
110	INVALID PROTOCOL ELEMENT VALUE	A protocol element in the request has an invalid value. The element is indicated in ADD_INFO.
201	QOP NOT ATTAINABLE	The requested QoP cannot be provided.

202	POSITIONING NOT ALLOWED~~NOT IN PRIVACY EXCEPTION LIST~~	The subscriber does not allow the application to position him/her for whatever reason (privacy settings in location server, LCS privacy class).~~The requesting application is not included in the privacy exception list of the MS.~~
~~203~~	~~CALL TO USER NOT SETUP~~	~~The requesting application has not a call set up to an MS that only allows call related location requests.~~
204	DISALLOWED BY LOCAL REGULATIONS	The location request is disallowed by local regulatory requirements.
207	MISCONFIGURATION OF LOCATION SERVER	The location server is not completely configured to be able to calculate a position.
500 - 599		Vendor specific errors

7 References

References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific:

- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

7.1 References (Normative)

[1] Hypertext Transfer Protocol –HTTP/1.1
RFC 2616, June 1999
Available at http://www.ietf.org

[2] The TLS Protocol Version 1.0
RFC 2246, January 1999
Available at http://www.ietf.org

[3] Extensible Markup Language (XML) 1.0
W3C Recommendation: REC-xml-20001006
Available at http://www.w3c.org

[4] Internet Assigned Numbers Authority (IANA)
http://www.iana.org/

[5] US-ASCII. Coded Character Set - 7-Bit American Standard Code for Information Interchange. Standard ANSI X3.4-1986, ANSI, 1986.

7.2 References (Informative)

[6] GSM 02.71: "Digital cellular telecommunications system (Phase 2+); Location Services (LCS); Service description; Stage 1".

[7] GSM 03.71: "Digital cellular telecommunications system (Phase 2+); Location Services (LCS); Functional description; Stage 2".

[8] GSM 09.02: "Digital cellular telecommunications system (Phase 2+); Mobile Application Part (MAP) specification".

[9] 3GPP TS 22.071: "Location Services (LCS); Service description, Stage 1".

[10] 3GPP TS 23.171: "Functional stage 2 description of location services in UMTS"

[11] 3GPP TS 23.271: "Functional stage 2 description of LCS"

[12] 3GPP TS 23.032: " Universal Geographical Area Description (GAD)"

[13] 3GPP TS 29.002: "Digital cellular telecommunications system (Phase 2+); Mobile Application Part (MAP) specification".

[14] 3GPP TS 29.198-6 "Open Service Access (OSA) Application Programming Interface (API); Part 6: Mobility"

[15] Parlay API 2.1 Mobility Interfaces v1.1.1.
Available on the Parlay web-site at http://www.parlay.org

[16] ITU-T E.164: "The international public telecommunication numbering plan

[17] TR-45 J-STD-036 "Enhanced Wireless 9-1-1 Phase 2 Document"

[18] IS-41D: " Cellular Radiotelecommunications Intersystem Operations", June 1997

[19] OpenGIS© Consortium Abstract Specification Topic 2: 01-063R2 at the public OGC document repository http://www.opengis.org/techno/abstract/02-102.pdf.

[20] OpenGIS© Consortium Recommendation Paper 01-014r5: Recommended Definition Data for Coordinate Reference Systems and Coordinate Transformations available at http://www.opengis.org/techno/discussions/01-014r5.pdf

[21] OpenGIS© Consortium Impementation Specification: Geography Markup Language V 2.0 available at http://www.opengis.net/gml/01-029/GML2.html

[22] OpenGIS© Consortium Abstract Specification Topic 1 Feature Geometry : 010101 at the public document repository http://www.opengis.org/techno/abstract/01-101.pdf.

8 Appendix A (informative): Adaptation to 3GPP LCS

8.1 Version mapping between 3GPP TS23.271 and this specification

The following table shows the version number of this specification (LIF TS101) fully conforming to a certain version of 3GPP TS23.271, i.e. the version of this specification for the correct reference in a certain version of the 3GPP specification.

3GPP TS23.271 version number	Conforming version number of LIF TS101
~~5.2.0~~Release 5	~~3.0.0~~Version 3

Note: In case there are versions not appearing in this table, it should be interpreted that such update did not affect the other specification. That is, the version number not appearing in the table should apply to the conformance mapping for the closest smaller version number in the table.

8.2 The terminology mapping table with 3GPP LCS Specifications

The following is a list of the terms in MLP used differently from the ones defined for 3GPP:

Term		Notes
MLP	3GPP	
Location Server	LCS Server	
MS (Mobile Station)	UE	
MSID (Mobile Station Identifier)	Identification of the target UE	
MPC (Mobile Positioning Centre)		There is no term applicable to 3GPP.

8.3 The corresponding terms used for the location procedures in 3GPP LCS Definition

The following is a list of terms defined in MLP corresponding to the 3GPP LCS definition in TS23.271 for the location procedures.

Location procedures defined in 3GPP(23.271)		Services defined in MLP
Circuit Switched Mobile Terminating Location Request CS-MT-LR	LCS Service Request	Standard Location Immediate Request
	LCS Service Response	Standard Location Immediate Answer
CS-MT-LR without HLR Query - applicable to North America Emergency Calls only	LCS Service Request	Emergency Location Immediate Request
	LCS Service Response	Emergency Location Immediate Answer

Packet Switched Mobile Terminating Location Request PS-MT-LR	LCS Service Request	Standard Location Immediate Request
	LCS Service Response	Standard Location Immediate Answer
Network Induced Location Request NI-LR	Location Information	Emergency Location Report
Packet Switched Network Induced Location Request PS-NI-LR	Location Information	Emergency Location Report
Mobile Terminating Deferred Location Request	LCS Service Request	Triggered Location Reporting Request
	LCS Service Response (Provide Subscriber Location ack)	Triggered Location Reporting Answer
	LCS Service Response (Subscriber Location Report)	Triggered Location Report
Combined Periodical/Deferred Mobile Terminating Location Request	LCS Service Request	Triggered Location Reporting Request
	LCS Service Response (Provide Subscriber Location ack)	Triggered Location Reporting Answer
	LCS Service Response (Subscriber Location Report)	Triggered Location Report
Cancellation of a Deferred Location Request	LCS Cancel Service Request	Triggered Location Reporting Stop Request
	LCS Cancel Service Response	Triggered Location Reporting Stop Answer
Mobile Originating Location Request, Circuit Switched CS-MO-LR	Location Information	Standard Location Report
Mobile Originating Location Request, Packet Switched PS-MO-LR	Location Information	Standard Location Report

~~9~~8.4 Error Mapping(informative)

The following list provides a mapping between the errors defined for LCS in MAP (see [13]) and MLP (see section 6)

MAP error	MLP resid
Unknown subscriber	4
Unidentified Subscriber	4
Absent Subscriber	5
System failure	1
Facility Not Supported	6

MAP error	MLP resid
Unexpected Data Value	1
Data missing	1
Unauthorised LCS Client with detailed reason	3
Position method failure with detailed reason.	6
Illegal Subscriber	2
Illegal Equipment	2
Unauthorized requesting network	2

9 Appendix B - HTTP Mapping

This section describes how to use MLP over the HTTP transport mechanism using "HTTP/1.1".

HTTP is a request/response protocol involving a server and a client. In the context of MLP, the client is referred to as the LCS Client and the server is the Location Server (GMLC/MPC). For more information about HTTP, refer to http://www.w3.org and ref [1].

The Location Server should provide two socket ports for operation, one for encryption with SSL/TLS and one without. The reason for having one insecure port is that encryption can consume resources, and if the client is in a secure domain there might not be a need for encryption. Applications residing in an insecure domain, i.e. on the Internet, may use the secure port to ensure the security and privacy of the location information.

For further information about SSL/TLS see ref [2].

Two port numbers have been selected and proposed as standard ports for location servers implementing MLP. These ports are registered with IANA (Internet Assigned Numbers Authority, see ref [4]). The two port numbers are:

- 9211 for secure SSL/TLS transfers
- 9210 for insecure transfers

A Location Server can choose to introduce any other socket based or HTTP transparent technology for secure transfers. Any such technology should be provided over a different port than the two mentioned above.

Note: IANA has ~~not yet~~ approved the use of these ports.

~~9.1.1~~9.1 Location Services using HTTP

An LCS Client requests a Location Service by issuing an HTTP POST request towards the Location Server. For more information about HTTP POST, see ref. [1]. The request line syntax is shown below.

Request-line: `POST` *`SP`* `host` *`SP`* `HTTP/1.1` *`CRLF`*

The request must include the entity-header Content-length field as part of the request. The message body of the request should include the XML formatted request and should have the length specified by the LCS Client in the Content-length field.

If the request is a deferred request (triggered or periodic) the result is delivered to the client through an HTTP POST operation issued by the Location Server. This implies that the client must be able to receive HTTP POST requests and be able to give a valid response.

All Location Services are invoked by sending a request using HTTP POST to a certain URI. An example of an URI is shown below.

http://host:port/LocationQueryService/

The response to the invocation of a Location Service is returned using an HTTP response.

If the LCS client requests standard location of asynchronous mode, triggered or periodic reporting of location, the Location Server will return the answer by performing an HTTP POST operation towards the client. The client must specify the URI that the answer should be posted to. This is done in the service request or by having it in the LCS client profile that can be stored in the Location Server.

The answer will be included in the message body and the Content-length entity will be set to the length of the answer.

When an LCS client attempts to invoke a service request that is not defined in this specification, the Location Server shall return a General Error Message (GEM) in a HTTP '404' error reponse:

Status-Line:~~ =~~ `HTTP`~~`-Version`~~`/1.1` *SP* ~~`Status-Code`~~`404` *SP* `Not Found`~~`Reason-Phrase`~~ *CRLF*

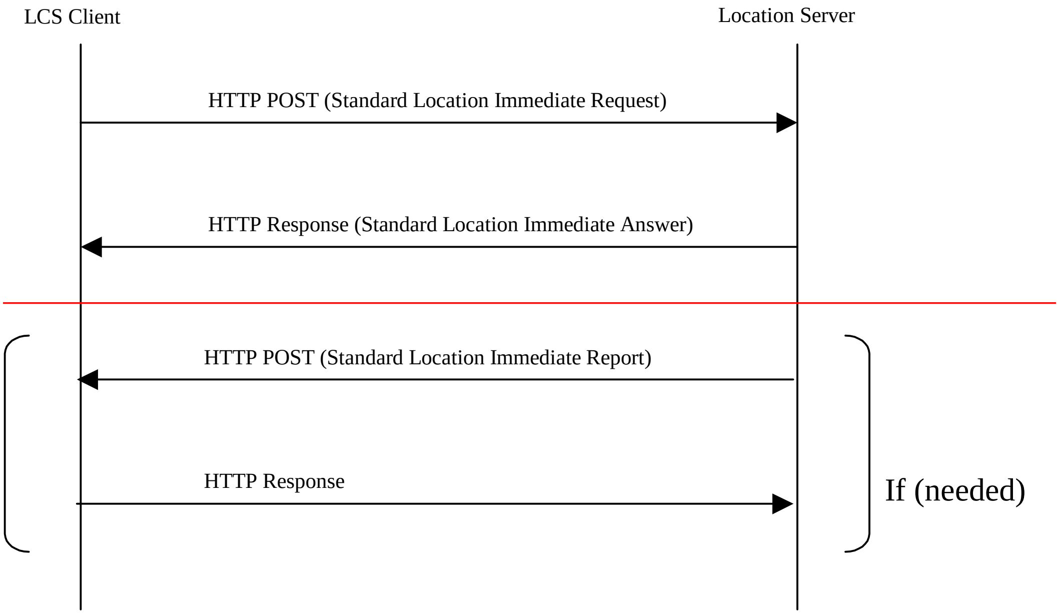
~~The precise status code used is left for implementation of the Location Server~~.

~~9.1.2Message Sequence Diagrams~~

~~9.1.2.1Standard Location Immediate Service~~

~~The service consists of the following messages:~~
~~□Standard Location Immediate Request~~
~~□Standard Location Immediate Answer~~
~~□Standard Location Immediate Report~~

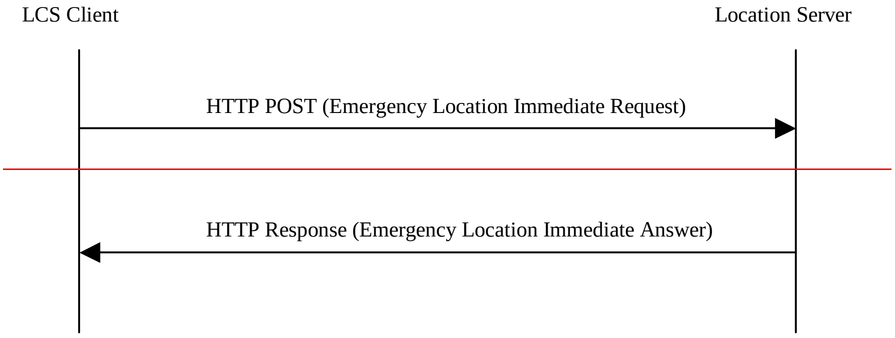
~~The following HTTP message flow encapsulates this service:~~


LCS Client
Location Server
HTTP POST (Standard Location Immediate Request)
HTTP Response (Standard Location Immediate Answer)
HTTP POST (Standard Location Immediate Report)
HTTP Response
If (needed)

~~9.1.2.2Emergency Location Immediate Service~~

~~The service consists of the following messages:~~
- ~~Emergency Location Immediate Request~~
- ~~Emergency Location Immediate Answer~~

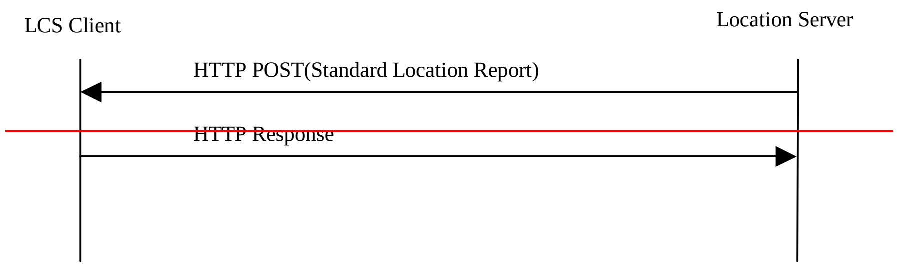
~~The following HTTP message flow encapsulates this service:~~


LCS Client
Location Server
HTTP POST (Emergency Location Immediate Request)
HTTP Response (Emergency Location Immediate Answer)

~~9.1.2.3Standard Location Report Service~~

~~The service consists of the following message:~~
- ~~Standard Location Report~~

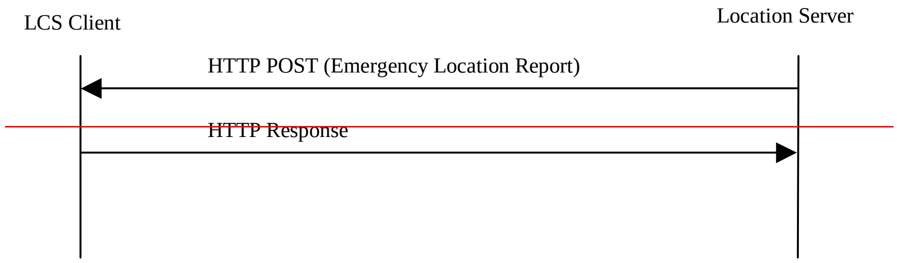
~~The following HTTP message flow encapsulates this service:~~



~~9.1.2.4Emergency Location Report Service~~

~~The service consists of the following message:~~
~~□Emergency Location Report~~

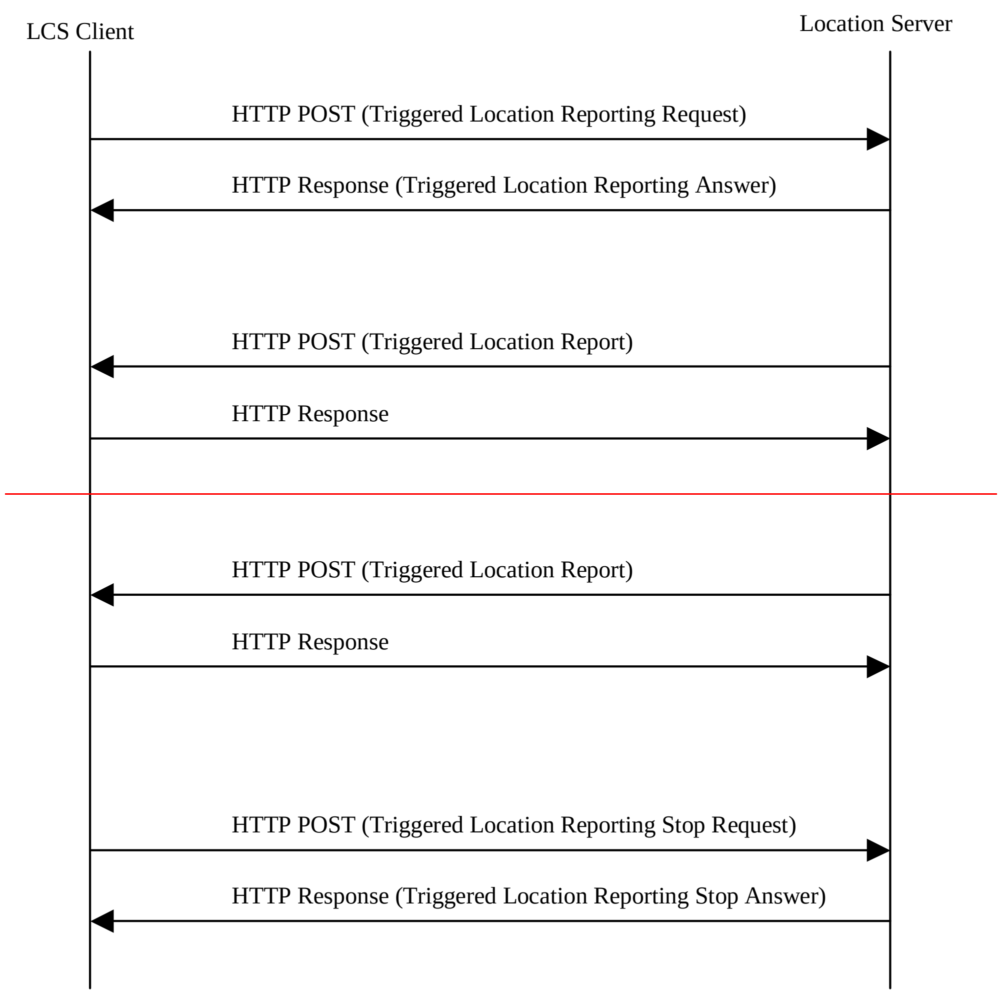
~~The following HTTP message flow encapsulates this service:~~



~~9.1.2.5Triggered Location Report Service~~

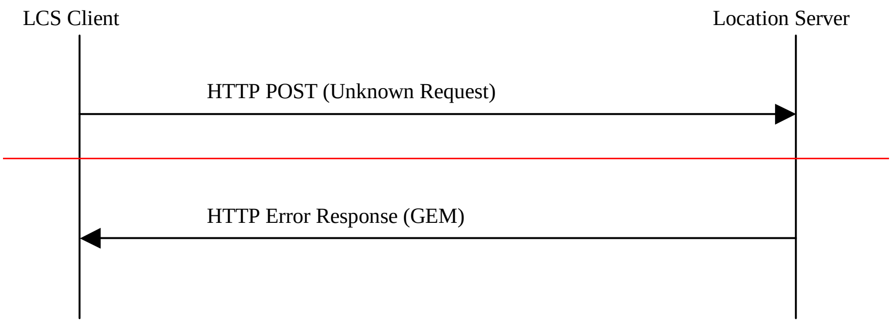
~~The service consists of the following messages:~~
~~□Triggered Location Reporting Request~~
~~□Triggered Location Reporting Answer~~
~~□Triggered Location Report~~
~~□Triggered Location Reporting Stop Request~~
~~□Triggered Location Reporting Stop Answer~~

~~The following HTTP message flow encapsulates this service:~~


LCS Client
Location Server
HTTP POST (Triggered Location Reporting Request)
HTTP Response (Triggered Location Reporting Answer)
HTTP POST (Triggered Location Report)
HTTP Response
HTTP POST (Triggered Location Report)
HTTP Response
HTTP POST (Triggered Location Reporting Stop Request)
HTTP Response (Triggered Location Reporting Stop Answer)

~~9.1.2.6General Error Message~~

~~The following message flow encapsulates this service:~~



~~9.1.3~~9.2 Request and Response Encapsulation

A request and a response consist of a header part and a body part so to be able to make a location request with a single XML document the header and the body are encapsulated in the same service initiation DTD.~~In the previous section the context header was introduced.~~ The context header holds the authentication and authorization data pertinent to a particular location request. The body part is described in the sections 4.3.2~~0~~ - 4.3.6~~1.5.5~~.~~ To be able to make a location request with a single XML document the header and the body are encapsulated in the same service initiation DTD.~~

~~9.1.3.1~~9.2.1 Service Initiation DTD

```
<!-- MLP_SVC_INIT -->

<!ENTITY    % extension.message          "">

<!ELEMENT   svc_init                     (hdr, (slir | eme_lir | tlrr | tlrsr
                                         %extension.message;))>
<!ATTLIST   svc_init
            ver CDATA                    #FIXED ~~"2.2.1"~~"2.3.0">

<!ENTITY    % mlp_ctxt.dtd               SYSTEM "MLP_CTXT~~_221.DTD~~_230.DTD">
%mlp_ctxt.dtd;
<!ENTITY    % mlp_id.dtd                 SYSTEM "MLP_ID~~_221.DTD~~_230.DTD">
%mlp_id.dtd;
<!ENTITY    % mlp_func.dtd               SYSTEM "MLP_FUNC~~_221.DTD~~_230.DTD">
%mlp_func.dtd;
<!ENTITY    % mlp_qop.dtd                SYSTEM "MLP_QoP~~_221.DTD~~_230.DTD">
%mlp_qop.dtd;
<!ENTITY    % mlp_loc.dtd                SYSTEM "MLP_LOC~~_221.DTD~~_230.DTD">
%mlp_loc.dtd;
<!ENTITY    % mlp_shape.dtd              SYSTEM "MLP_SHAPE~~_221.DTD~~_230.DTD">
%mlp_shape.dtd;
<!ENTITY    % mlp_gsm_net_param.dtd      SYSTEM "MLP_GSM_NET~~_221.DTD~~_230.DTD">
%mlp_gsm_net_param.dtd;


<!ENTITY    % mlp_hdr.dtd                SYSTEM "MLP_HDR~~_221.DTD~~_230.DTD">
%mlp_hdr.dtd;
<!ENTITY    % mlp_slir.dtd               SYSTEM "MLP_SLIR~~_221.DTD~~_230.DTD">
%mlp_slir.dtd;
<!ENTITY    % mlp_eme_lir.dtd            SYSTEM "MLP_EME_LIR~~_221.DTD~~_230.DTD">
%mlp_eme_lir.dtd;
<!ENTITY    % mlp_tlrr.dtd               SYSTEM "MLP_TLRR~~_221.DTD~~_230.DTD">
%mlp_tlrr.dtd;
<!ENTITY    % mlp_tlrsr.dtd              SYSTEM "MLP_TLRSR~~_221.DTD~~_230.DTD">
%mlp_tlrsr.dtd;
```

Example

```
<?xml version="1.0" ?>
<!DOCTYPE svc_init SYSTEM "MLP_SVC_INIT~~_221.DTD~~_230.DTD">
<svc_init ver=~~"2.2.1"~~"2.3.0">
  <hdr ver=~~"2.2.1"~~"2.3.0">
    ...
  </hdr>
  <slir>
    ...
  </slir>
</svc_init>
```

~~9.1.3.2~~9.2.2 Service Result DTD

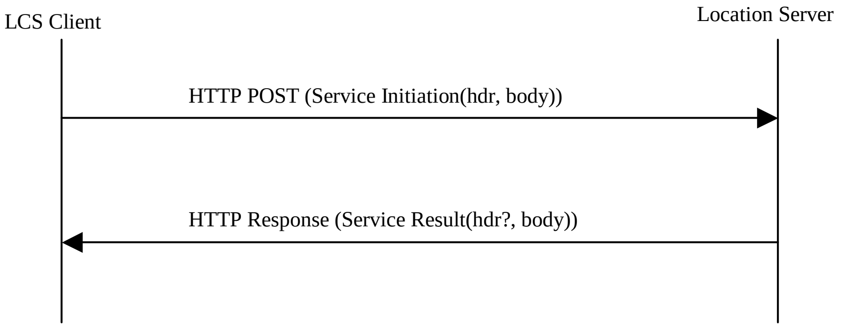
```
<!-- MLP_SVC_RESULT -->

<!ENTITY    % extension.message         "">
<!ELEMENT   svc_result                  (hdr?, (slia | slirep | slrep | eme_lia | emerep |
                                        tlra | tlrep | tlrsa %extension.message;))>

<!ATTLIST   svc_result
            ver CDATA                   #FIXED ~~"2.2.1"~~"2.3.0">

<!ENTITY    % mlp_ctxt.dtd              SYSTEM "MLP_CTXT~~_221.DTD~~_230.DTD">
%mlp_ctxt.dtd;
<!ENTITY    % mlp_id.dtd                SYSTEM "MLP_ID~~_221.DTD~~_230.DTD">
%mlp_id.dtd;
<!ENTITY    % mlp_func.dtd              SYSTEM "MLP_FUNC~~_221.DTD~~_230.DTD">
%mlp_func.dtd;
<!ENTITY    % mlp_qop.dtd               SYSTEM "MLP_QoP~~_221.DTD~~_230.DTD">
%mlp_qop.dtd;
<!ENTITY    % mlp_loc.dtd               SYSTEM "MLP_LOC~~_221.DTD~~_230.DTD">
%mlp_loc.dtd;
<!ENTITY    % mlp_shape.dtd             SYSTEM "MLP_SHAPE~~_221.DTD~~_230.DTD">
%mlp_shape.dtd;
<!ENTITY    % mlp_gsm_net_param.dtd     SYSTEM "MLP_GSM_NET~~_221.DTD~~_230.DTD">
%mlp_gsm_net_param.dtd;


<!ENTITY    % mlp_hdr.dtd               SYSTEM "MLP_HDR~~_221.DTD~~_230.DTD">
%mlp_hdr.dtd;
<!ENTITY    % mlp_slia.dtd              SYSTEM "MLP_SLIA~~_221.DTD~~_230.DTD">
%mlp_slia.dtd;
<!ENTITY    % mlp_slirep.dtd            SYSTEM "MLP_SLIREP~~_221.DTD~~_230.DTD">
%mlp_slirep.dtd;
<!ENTITY    % mlp_slrep.dtd             SYSTEM "MLP_SLREP~~_221.DTD~~_230.DTD">
%mlp_slrep.dtd;
<!ENTITY    % mlp_eme_lia.dtd           SYSTEM "MLP_EME_LIA~~_221.DTD~~_230.DTD">
%mlp_eme_lia.dtd;
<!ENTITY    % mlp_emerep.dtd            SYSTEM "MLP_EMEREP~~_221.DTD~~_230.DTD">
%mlp_eme~~_~~rep.dtd;
<!ENTITY    % mlp_tlra.dtd              SYSTEM "MLP_TLRA~~_221.DTD~~_230.DTD">
%mlp_tlra.dtd;
<!ENTITY    % mlp_tlrep.dtd             SYSTEM "MLP_TLREP~~_221.DTD~~_230.DTD">
%mlp_tlrep.dtd;
<!ENTITY    % mlp_tlrsa.dtd             SYSTEM "MLP_TLRSA~~_221.DTD~~_230.DTD">
%mlp_tlrsa.dtd;
```

Example

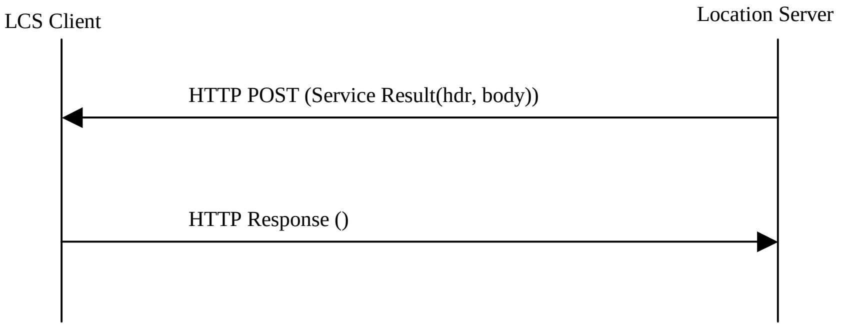
```
<?xml version="1.0" ?>
<!DOCTYPE svc_init SYSTEM "MLP_SVC_RESULT~~_221.DTD~~_230.DTD">
<svc_result ver=~~"2.2.1"~~"2.3.0">
  <slia>
    ...
  </slia>
</svc_result>
```

9.2.3 Message Sequence Diagram

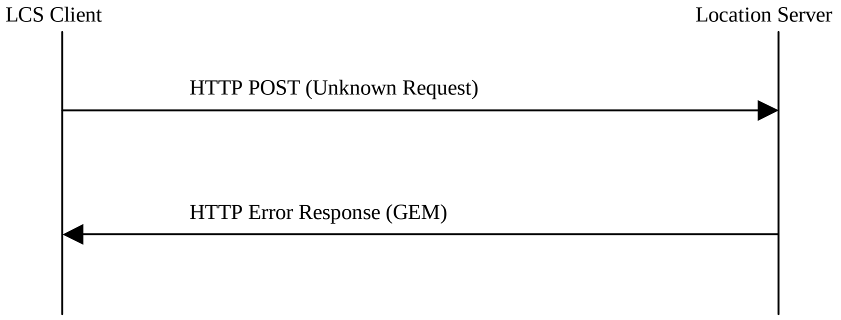
The following HTTP sequence is used for all the defined service requests/responses in MLP.


LCS Client
Location Server
HTTP POST (Service Initiation(hdr, body))
HTTP Response (Service Result(hdr?, body))

The following HTTP sequence diagram is used for all defined reports in MLP.



The following HTTP sequence diagram is used in the case of a General Error Message.



10 Appendix C: Geographic Information

10.1 Coordinate Reference systems (Informative)

The study of determining the relative positions on or close to the surface of the earth is a complex science, referred to as geodesy. A complete definition of Coordinate Reference systems is not within the scope of this standard. This section includes a brief overview of the subject. For more details see the OpenGIS© Consortium Abstract Specification Topic 2 [19.

10.1.1 The Geoid, ellipsoids and datums

The Geoid is a physically realizable surface defined by the set of points with equal gravity potential approximately at the Mean Sea Level. While this surface is measurable it is not easy to define mathematically. In order to use known mathematics, the Geoid is approximated by an ellipsoid (spheroid).

There are many ellipsoids, each defined to best approximate some part of the Geoid. These ellipsoids are defined by an ellipse that is rotated about the major axis. There are many methods for defining an ellipse, the most common used in Geodesy the length of the semi-major axis and the flattening. This defines a mathematical ellipsoid for calculations. it does not provide enough information to locate the ellipsoid with respect to the Geoid or other ellipsoids. To locate the ellipsoid in space a datum is defined. Some of the common ellipsoids are WGS84, Bessel1841, Clark 1866.

A datum is the ellipsoid with it's position in space. The position is defined by the origin and orientation of the ellipsoid with respect to the Geoid. Different datums locate latitude, longitude at different positions in space. For example ellipsoids Samboja, CH1903 and Stockholm are each based on Bessel1841, the National Geodetic Network and World Geodetic System 1984 are based on WGS84.

10.1.2 Coordinate systems

A coordinate system is the link between the datum and the coordinate values. It defines all of the information about the axes system that defines the values. The names of the axes, their units (formats), the order of ordinates ((Easting, Northing) versus (Northing, Easting)) and the angle between the axes are defined by the coordinate system.

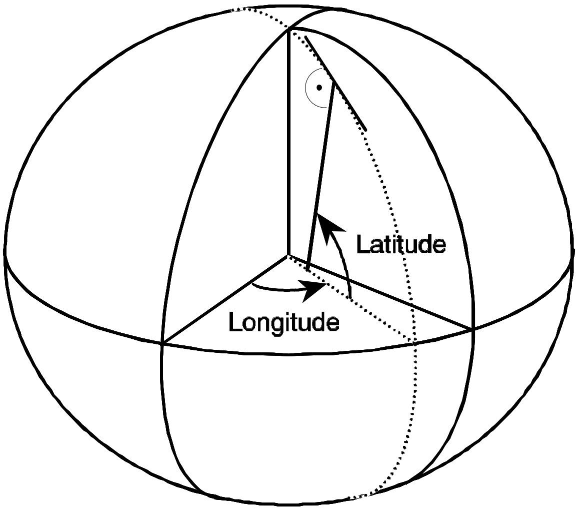
10.1.2.1 Cartesian coordinate systems

A Cartesian coordinate system is defined by values of (**x,y,(z)**). x is the distance from the x-axis, y is the distance from the y-axis, z the distance from the z-axis.The axis are orthogonal to each other. The unit used for x, y, z are a distance unit, such as meter. These coordinate systems are used for flat 'planar' descriptions of points. In general they are used over small areas where a projection method has been used to minimize distortions of the geography in the area.

10.1.2.2 Ellipsoid coordinates

More global geographic calculations need to take the surface of the earth into account. So we need a second coordinate system that describes each position relative to other points and lines on the earth's surface.

Each point can then be described as set of values (longitude, latitude) or (longitude, latitude, altitude) giving a point on the ellipsoid or relative to the ellipsoid we choose to describe the earth. The longitude tells us how far east we have to move on the equator from the null-meridian, the latitude tells us how far north to move from the equator and the altitude tells us how far above the ellipsoid to go to finally reach the location. Negative values direct us to go in the opposite direction.



10.1.3 Coordinate Reference Systems

The two coordinate reference systems relevant to this protocol are Geographic 2D Coordinate Reference Systems and Projected Coordinate Reference Systems.

Geographic 2D Coordinate Reference Systems describe locations on the ellipsoid. They are used for large national or continental geodetic networks. In particular GPS uses the Geographic 2D Coordinate Reference System WGS84. This uses the World Geodetic System 1984 based on the WGS84 ellipsoid. The coordinate axes have units of decimal degrees (or DMSH) with ordinate order (Northing, Easting). This Coordinate Reference System is the default for all basic MLP service requests and responses. A GMLC is only required to support WGS84. The GMLC geographies that are defined with altitude are modeled in this protocol as geographies in a Geographic 2D CRS with a separate altitude element, not as a Geographic 3D CRS. The geographies are planar and carrying a constant z value is not desirable.

There are several ways to convert ellipsoid coordinates to 2 dimensional cartesian coordinates. These are called projection methods. Each method is designed to minimize some type of distortion in the mapping for the ellipsoid to the 2D Cartesian coordinate system.

Projected Coordinate Reference Systems are used for map display, to allow Cartesian mathematics and for Advanced Location Services.

10.2 Coordinate Reference System Transformations (Informative)

A transformation is used to define a point in one CRS into the appropriate values in a second CRS. When the datums are the same, the transformation can frequently be defined by equations. A transformation from one datum to another is usually done with a least squares approximation. Transformation equations are available in from several places, transformation services are also available.

10.3 Methodology for defining CRSs and transformations in this protocol (Normative)

The MLP protocol defines the CRS by citing an authority and the unique reference identifier for the CRS defined by this authority. This leaves the definition of many CRS used over the world to be defined by a group of geodesy experts. This methodology is used by the OpenGIS© Consortium and the ISO TC 211 working group for well-known CRS. The encoding used is from the OpenGIS© Consortium Recommendation Paper 01-014r5: Recommended Definition Data for Coordinate Reference Systems and Coordinate Transformations [20].

The MLP protocol ~~will~~ may use the {EPSG} authority as an example. Support of other authority is for further study. This database is defined by a Microsoft Access database which can be found at www.epsg.org. An xml version of this database will be available at http://www.opengis.net/gml/srs/epsg.xml in the future.

The default WGS84 CRS is defined to be 4326 by the EPSG authority. Other examples are 326xx define the UTM xx N zones.

Coordinate Reference System transformation are done by an advance Location Service request. The implementation of this service is determined by the provider.

10.4 Supported coordinate systems and datum

All MLP implementations must support at least the WGS84 Coordinate Reference System.

10.5 Shapes representing a geographical position

There are a number of shapes used to represent a geographic area that describes where a mobile subscriber is located. There are additional shapes that are required for advanced MLP services. The standards bodies for geographic data for advanced MLP services such as routing, geocoding, coordinate conversion, and map display are the Location Interoperability Forum, the OpenGIS© Consortium and the ISO TC211 working group. The current public XML specification defining geography from these groups is GML V211 [21]. These two groups work together and are working towards a GML V3 with additional geometry and topology types. The geometry required for the MLP is the GMLV211 with additional polygon types with boundaries that contain circles, ellipses or circular arcs. GML V3 will define the linear curves segments to allow the these polygons to be defined. These boundaries will be defined as special cases of polygons, using the given interpolation methods. The following geographies are defined in this protocol. The relevant OGC Abstract Specification is Topic 1 [22].

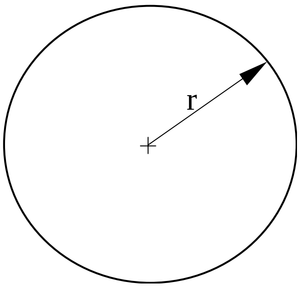
10.5.1 Ellipsoid point

This a point on the ellipsoid and is modeled as a point in a Geographic 2D Coordinate Reference Systems.

10.5.2 Ellipsoid point with uncertainty circle

An ellipsoid point with uncertainty circle is characterized by the coordinates of an ellipsoid point (the origin) and a radius, “r”. It describes the set of points on the ellipsoid, which are at a distance from the point of origin less than or equal to “r”. This shape can be used to indicate points on the Earth surface, or near the Earth surface. This shape is a special case of a polygon with no interior boundaries.

The typical use of this shape is to indicate a point when its position is known only with a limited accuracy.



10.5.3 Ellipsoid point with uncertainty ellipse

The shape of an "ellipsoid point with uncertainty ellipse" is characterized by the following:

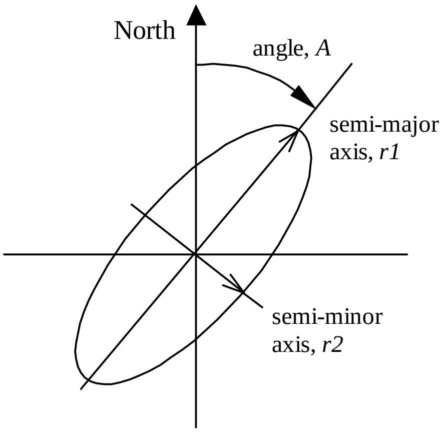
- The coordinates of an ellipsoid point (the origin)

- The distances r1 and r2
- The angle of orientation A

It describes formally the set of points on the ellipsoid, which fall within or on the boundary of an ellipse. This ellipse has a semi-major axis of length r1 oriented at angle A (0 to 180°) measured clockwise from north and a semi-minor axis of length r2. The distances being the geodesic distance over the ellipsoid, i.e., the minimum length of a path staying on the ellipsoid and joining the two points, as shown in figure below.

As for the ellipsoid point, this can be used to indicate points on the Earth's surface, or near the Earth's surface, of same latitude and longitude. This shape is a special case of a polygon with no interior boundaries.

The typical use of this shape is to indicate a point when its position is known only with a limited accuracy, but the geometrical contributions to uncertainty can be quantified.

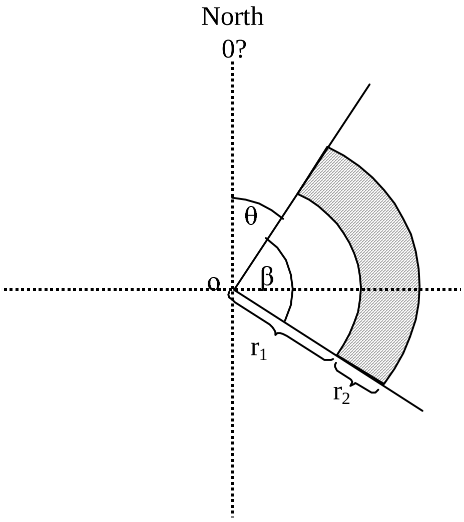


10.5.4 Ellipsoid point with uncertainty arc

The shape of an "ellipsoid point with uncertainty arc" is characterized by the following:

- The coordinates of an ellipsoid point (the origin)
- The inner radius(r) and uncertainty radius(r),
- The offset angle (θ) and included angle (β)

An arc is defined by a point of origin with one offset angle and one uncertainty angle plus one inner radius and one uncertainty radius. In this case the striped area describes the actual arc area. The smaller arc defines the inner radius(r) and the difference between inner and the outer arc defines the uncertainty radius(r). This shape is a special case of a polygon with no interior boundaries.



10.5.5 Polygon

A **Polygon** is a connected surface. Any pair of points in the polygon can be connected to one another by a path. The boundary of the Polygon is a set of LinearRings. We distinguish the outer (exterior) boundary and the inner (interior) boundaries; the LinearRings of the interior boundary cannot cross one another and cannot be contained within one another. There must be at most one exterior boundary and zero or more interior boundary elements. The ordering of LinearRings and whether they form clockwise or anti-clockwise paths is not important. The minimum number of points allowed in a LinearRing is 3.

A **LinearRing** is a closed, simple piece-wise linear path which is defined by a list of coordinates that are assumed to be connected by straight line segments. The last coordinate must be coincident with the first coordinate and at least four coordinates are required (the three to define a ring plus the fourth duplicated one). This geometry is only used in the construction of a Polygon.

For basic MLP services polygons are the number of interior bondaries MUST be 0. Also to conform to 3GPP TS 23032 the maximum number of points allowed in an exterior boundary is 15. The points shall be connected in the order that they are given.

The described area is situated to the right of the exterior boundaries and left of the interior boundaries with the downward direction being toward the Earth’s center and the forward direction being from a point to the next.

NOTE: This definition does not permit connecting lines greater than roughly 20 000 km. If such a need arises, the polygon can be described by adding an intermediate point.

Computation of geodesic lines is not simple. Approximations leading to a maximum distance between the computed line and the geodesic line of less than 3 meters are acceptable.

10.5.6 LineString

A **LineString** is a piece-wise linear path defined by a list of coordinates that are assumed to be connected by straight line segments. A closed path is indicated by having coincident first and last coordinates. At least two coordinates are required.

10.5.7 Box

The **Box** element is used to encode extents. Each <Box> element encloses a sequence of two <coord> elements containing exactly two coordinate tuples; the first of these is constructed from the minimum values measured along all axes, and the second is constructed from the maximum values measured along all axes

10.5.8 Geometries Collections

These are geometry objects that contain 2 or more primative geometry objects. These collections can either be homogenous, a set of points, or heterogeneous, a point, circularArea and a LineString.

Geometry collections are not valid for the basic MLP services.