TOC

Geopriv	J. Winterbottom
Internet-Draft	Andrew Corporation
Intended status: Standards Track	H. Tschofenig
Expires: August 23, 2008	Nokia Siemens Networks
	M. Thomson
	Andrew Corporation
	February 20, 2008

HELD Protocol Context Management Extensions
draft-winterbottom-geopriv-held-context-02.txt

Status of this Memo

Abstract

This document describes a protocol extension for the HTTP Enabled Location Delivery (HELD) protocol. It allows a Target to manage their location information on a Location Information Server (LIS) through the application of constraints invoked by accessing a location URI. Constraints described in this memo restrict how often location can be accessed through a location URI, how long the URI is valid for, and the type of location information returned when a location URI is accessed. Extension points are also provided.

Table of Contents

1. Introduction

TOC

The HTTP Enabled Location Delivery (HELD) protocol specification [I-D.ietf-geopriv-http-location-delivery] (Barnes, M., Winterbottom, J., Thomson, M., and B. Stark, “HTTP Enabled Location Delivery (HELD),” August 2009.) provides a set of features that can be used by a Target to retrieve location information from a Location Information Server (LIS). The basic HELD specification does this in a more or less stateless manner, and when a location URI is retrieved the Target has no way of controlling how the URI is used; a Location Recipient in pocession of the location URI can get the Target's location until the URI expires. This basic mechanism may be reasonable in a limited set of applications, but is unacceptable in a broader range of applications. This position is highlighted in [I-D.ietf-geopriv-lbyr-requirements] (Marshall, R., “Requirements for a Location-by-Reference Mechanism,” November 2009.) which describes requirements for constraints relating to location URIs. This specification provides support for these requirements in HELD.

2. Terminology

TOC

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).
This document reuses the terms Target, as defined in [RFC3693] (Cuellar, J., Morris, J., Mulligan, D., Peterson, J., and J. Polk, “Geopriv Requirements,” February 2004.).
This document uses the term Location Information Server (LIS) as the node in the access network that provides location information about a Target. This term is also used in [I-D.ietf-geopriv-l7-lcp-ps] (Tschofenig, H. and H. Schulzrinne, “GEOPRIV Layer 7 Location Configuration Protocol; Problem Statement and Requirements,” July 2009.).

3. What is a Context?

TOC

A Location URI points to a LIS that is able to provide the location of a specific Target. The LIS is able to map the URI to the location of the Target inside its administrative domain. We call this mapping a "context". In the basic HELD specification the context is implicitly created with the request for a location URI in the locationRequest message. The Target has no control of the mapping from the URI to the Target's location. This specification provides a degree of control to the Target, allowing it to specify rules to the LIS on how a context should map a URI to location information.
A context expires when it reaches a certain age, at which time the mapping between the URI and the Target's location ceases. In the basic HELD specification the exiry time of the context is determined by the LIS when the Target requests a location URI. By allowing the Target to specify and change the life time of a context the Target is able to create URIs for limited periods, or to terminate URIs for which it no longer wishes its location to be returned. This specification provides explicit support for this functionality.

TOC

4. Constraints

Constraints restrict the ability of a Location Recipient to resolve a location URI to location information. The constraints are selected by the Target and they are provided to the LIS that maintains them along with the context. A LIS, understanding this specification, receives constraints provided by the Target, and returns a set of URIs influenced by the constraints.

A single Target may want to place different contraints on different references and hence may have multiple contexts on the LIS. The constraints describe what actions the LIS MUST take when a URI associated with the context is accessed. This document describes three basic constraints that a Target can use in combination for the same context. Once set, these rules remain in force of the life of the context.

4.1. Limited Use URIs

TOC

A limited use URI can only be accessed a fixed number of times to yield the location of the Target. Each time the URI is used to provide the location of the Target one usage is consumed. Once the limit is reached the URI no longer yields the location of the Target and the URI is deemed spent.

By setting the usage limit to 1, the Target is able to create a one-time-URI permitting a Location Recipient to obtain the Target's location only once. Setting the usage limit to something higher than 1 creates functionality analogous to a metro-ticket, where a Location Recipient in possession of the URI can access the Target's location many more times, but not exceeding the imposed limit.

Not setting a usage limit provides similar semantics to the URI in the base HELD specification, enabling a Location Recipient to continually obtain the Target's location until the URI expires due to age.

When a HELD URI is assigned to a context, the limit is the number of times that the URI can be accessed before the LIS returns an error. In the case of SIP or pres URIs it is the number of NOTIFY messages that are sent prior to the LIS returning an error. Where a context supports SIP, pres, and HELD URIs it is the combination of URI accesses and NOTIFY messages that constitutes the usage value, each time the Target's location is provided constitutes a usage.

4.2. Snapshot URIs

TOC

A snapshot URI points to the location of the Target at a specific point in time, and no matter how many times the URI is accessed it will always yield the same location. This is useful if, for example, the Target does not want to be tracked. In this specification the location snapshot to which a snapshot URIs points is captured when the context is created on the LIS.

4.3. Location Type URIs

TOC

A location type URI controls the form of location that can be accessed; This may be geodetic, civic, or both.

5. Protocol Details

TOC

This specification introduces three new HELD messages, create context (<createContext>), update context (<updateContext>), and context response (<contextResponse>). A LIS that does not understand this specification is expected to return a HELD *unsupportedMessage* error code in a HELD error message. A LIS that does understand this specification returns errors associated with context operations in a HELD error message. New error codes relating to failed context operations are defined in this specification.
The specification assumes that the LIS was discovered as part of the general HELD LIS discovery process. All messages are sent using the application/held+xml MIME type as defined in [I-D.ietf-geopriv-http-location-delivery] (Barnes, M., Winterbottom, J., Thomson, M., and B. Stark, “HTTP Enabled Location Delivery (HELD),” August 2009.).

5.1. Create Context Message

TOC

The Target creates a context on the LIS using a create context message. The basic create context message supports the constraints described in Section 4 (Constraints) and consist of three attributes and one element described below:

*uses: an optional attribute instructing the LIS on how many times a URI may yield the location of the Target. This is a positive integer, and has a default value of *unlimited*. The LIS SHOULD support the Target specifying up to at least 100 uses.

*snapshot: an optional attribute instructing the LIS to take a snapshot of the Target's location for use with the context. This a boolean value and has a default of *false* meaning that a snapshot is not taken, and the Target's location is determined each time the URI is accessed.

*locationType: an optional attribute instructing the LIS on the form of location that the URI MUST return. This is an enumeration and may have a value of *geodetic*, *civic*, or *any*. If unspecified by the Target the LIS will use a value of *any*. If the Target specifies a location type that the LIS cannot provide, then the LIS MUST fail the context creation.

*lifeTime: is a mandatory element that defines the maximum period in seconds that the LIS should keep the context for. The LIS MAY create the context with a shorter life time than was requested, but the life time MUST NOT be longer than was requested.

```
<createContext
     xmlns="urn:ietf:params:xml:ns:geopriv:held:context"
     uses="10"
     snapshot="false"
     locationType="any">
  <lifeTime>7200</lifeTime>
</createContext>
```

Figure 1: createContext Example

Figure 1 (createContext Example) shows a create context message defining a context which:

*may be accessed 10 times

*will determine the location of the Target each time it is accessed

*will return the location in either geodetic or civic form depending on the request to the URI

*will be valid for 2 hours from the time of context creation

TOC

5.2. Update Context Message

A Target can change the life time of a context using an update context message. As stated in Section 4 (Constraints) the three attributes used in the context creation, uses, snapshot, and locationType cannot be changed once a context is created.
Since the Target may have more than one context on the LIS, the Target needs to identify the context to be updated. It does this by including a context identifier that is provided to it by the LIS when the context is created.

```
<updateContext
    xmlns="urn:ietf:params:xml:ns:geopriv:held:context"
    id="uhvuhdbnuiehudbnvcujevuijeijcvij3">
  <lifeTime>3600</lifeTime>
</updateContext>
```

Figure 2: updateContext Life Time Change Example

When a Target includes a life time element in an update context message, the LIS needs to calculate a new context expiry time. The LIS MUST do this by adding the new life time value to the current time on the LIS. This mechanism means the Target can terminate a context at any time. It does this by updating the context with a life time of 0, which results in the LIS setting the context expiry time to the present. The LIS MAY also terminate a context if the life time value is set to less than 10 seconds.

```
<updateContext
    xmlns="urn:ietf:params:xml:ns:geopriv:held:context"
    id="uhvuhdbnuiehudbnvcujevuijeijcvij3">
  <lifeTime>0</lifeTime>
</updateContext>
```

Figure 3: updateContext Termination Example

TOC

5.3. Context Response Message

The LIS informs the Target about the outcome of context operations through the context response message. The LIS MUST always send a context response message to a Target in response to a create context or update context message when the outcome was successful. The context response message contains a code attribute indicating the performed operation, and the other attributes and elements indicating the state of the context.
The code attribute is an enumerated type and has one of the following values:

*_created_: The context was successfully created.

*_destroyed_: The context was destroyed.

*_updated_: The context was successfully updated.

The following list details the other attributes that may be returned in a context response message.

id: The identifier allocated to the context by the LIS. This identifier is unique in the scope of the LIS. The Target MUST keep this secret and MUST included it in all update requests. The LIS MUST return an id in all context response messages.

uses: The number of times that the context will yield the Target's location. The LIS MAY report either the original value, or the number of remaining uses. The LIS MUST report this value for all responses pertaining to a known and valid context. This value MAY be ommitted when indicating that a context has been destroyed.

snapshot: The value of the snapshot attribute in the context. The LIS MUST report this value for all responses pertaining to a known and valid context. This value MAY be ommitted when indicating that a context has been destroyed.

locationType: The type of location information that can be acquired through URIs addressing the context. The LIS MUST report this value for all responses pertaining to a known and valid context. This value MAY be omitted when indicating that a context has been destroyed.

expiry: The time at which the context will expire. After this time, all location URIs that reference this context no longer work. The LIS MUST report this value for all responses pertaining to a known context. This attribute MUST be provided even when a code value of _destroyed_ is included in the context repsonse message.

In addition to the above attributes, the LIS also provides a set of URIs that can used to access the Target's location with the surety that the context constraints will be applied. A URI set is returned whenever a context is successfully created on the LIS, and this set remains unchanged for the lifetime of the context. A context response message sent in reply to the create context message in Figure 1 (createContext Example) might look like Figure 4 (contextResponse Example).

```
<contextResponse
          xmlns="urn:ietf:params:xml:ns:geopriv:held:context"
          code="created"
          id="uhvuhdbnuiehudbnvcujevuijeijcvij4"
          uses="10"
          snapshot="false"
          locationType="any"
          expires="2007-11-01T13:30:00">
    <locationUriSet>
      <locationURI>
         held://lis.example.com:9768/357yc6s64ceyoiuy5ax3o4
      </locationURI>
      <locationURI>
         sips:357yc6s64ceyoiuy5ax3o4@lis.example.com:9769
      </locationURI>
    </locationUriSet>
</contextResponse>
```

Figure 4: contextResponse Example

5.4. Context Errors

TOC

When the LIS unable to perform the requested context operation it need to inform the Target of this. It does this using a held error message. New codes are defined for context operation errors:

- *badContextMessage*: The LIS was unable to understand the content of the message. In general this will apply to context messages containing extensions that the LIS does not understand.

- *unknownContext*: The LIS was unable to find the context.

*updateContextFailed: The LIS was unable to updated the requested context.

*createContextFailed: The LIS was unable to created the requested context.

A Target implementing this specification MUST accept a any HELD error message as a valid response to a create context or update context message as a LIS may not understand context messages. A LIS that does understand context messages is expected to return the error codes above unde the prescribed circumstances.

```
<error xmlns="urn:ietf:params:xml:ns:geopriv:held"
       code="createContextFailed"
       message="locationType of Civic is not supported"/>
```

Figure 5: Example Error Message

5.5. Location URI and Context Identifier Generation Rules

TOC

A primary aim of this specification is to provide a Target a means to cancel a location URI so that it can no longer be used to provide its location. To achieve this, a location URI generated as part of a context creation needs to be unique with in the scope of the LIS, and identify only that context. If the Target destroys a context and subsequently creates a new one, URIs associated the new context MUST be different from those generated for the previous context. [I-D.ietf-geopriv-http-location-delivery] (Barnes, M., Winterbottom, J., Thomson, M., and B. Stark, “HTTP Enabled Location Delivery (HELD),” August 2009.) and [I-D.ietf-geopriv-lbyr-requirements] (Marshall, R., “Requirements for a Location-by-Reference Mechanism,” November 2009.) provide guidance on the creation and desired characteristcs of a location URI.
The context identifier provided by the LIS to the Target in the context response message MUST be unique and MUST be different from the identifier provided in any location URI, and it MUST NOT be feasible to determine the context-ID from the location URI. This constraint ensures that possession of a location URI does not automatically provide access and control over the internals of the context. It MAY be feasible to determined the location URI knowing the context-ID however.

A context identifier is generated by a LIS to uniquely identify a context. It MUST NOT be feasible for a third-party to easily determine a context identifier by knowing the identity of the Target. This implies that internal correlation (using a hash-table or similar) is the only method that the LIS can use to associate a context id with a particular Target.

6. XML Schema

TOC

```
<?xml version="1.0"?>
<xs:schema
    targetNamespace="urn:ietf:params:xml:ns:geopriv:held:context"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:heldCx="urn:ietf:params:xml:ns:geopriv:held:context"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
    elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:simpleType name="locationType">
    <xs:restriction base="xs:token">
      <xs:enumeration value="any"/>
      <xs:enumeration value="civic"/>
      <xs:enumeration value="geodetic"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="codeType">
    <xs:restriction base="xs:token">
      <xs:enumeration value="created"/>
      <xs:enumeration value="updated"/>
      <xs:enumeration value="destroyed"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="useType">
    <xs:union memberTypes="xs:positiveInteger">
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="unlimited"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:union>
  </xs:simpleType>

  <xs:complexType name="createContextMsg">
    <xs:complexContent>
      <xs:restriction base="xs:anyType">
        <xs:sequence>
          <xs:element name="lifeTime" type="xs:nonNegativeInteger "
                  minOccurs="1" maxOccurs="1"/>
          <xs:any namespace="##other" processContents="lax"
                  minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="uses" type="heldCx:useType"
                      use="optional" default="unlimited"/>
        <xs:attribute name="snapshot" type="xs:boolean"
                      use="optional" default="false"/>
        <xs:attribute name="locationType" type="heldCx:locationType"
```

```
                     use="optional" default="any"/>
       <xs:anyAttribute namespace="##any" processContents="lax"/>
     </xs:restriction>
   </xs:complexContent>
 </xs:complexType>

 <xs:complexType name="uriSetType">
      <xs:complexContent>
       <xs:restriction base="xs:anyType">
         <xs:sequence>
           <xs:element name="locationURI" type="xs:anyURI"
                       minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
       </xs:restriction>
      </xs:complexContent>
  </xs:complexType>

 <xs:complexType name="contextResponseMsg">
   <xs:complexContent>
     <xs:restriction base="xs:anyType">
       <xs:sequence>
         <xs:element name="locationUriSet" type="heldCx:uriSetType"
                 minOccurs="1" maxOccurs="1"/>
         <xs:any namespace="##other" processContents="lax"
                 minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attribute name="id" type="xs:token"
                     use="required"/>
       <xs:attribute name="expires" type="xs:dateTime"
                     use="required"/>
       <xs:attribute name="uses" type="xs:positiveInteger"
                     use="optional"/>
       <xs:attribute name="snapshot" type="xs:boolean"
                     use="optional"/>
       <xs:attribute name="locationType" type="heldCx:locationType"
                     use="optional"/>
       <xs:attribute name="code" type="heldCx:codeType"
                     use="required"/>
       <xs:anyAttribute namespace="##any" processContents="lax"/>
     </xs:restriction>
   </xs:complexContent>
 </xs:complexType>

<xs:complexType name="updateContextMsg">
   <xs:complexContent>
     <xs:restriction base="xs:anyType">
       <xs:sequence>
         <xs:element name="lifeTime" type="xs:nonNegativeInteger "
                 minOccurs="0" maxOccurs="1"/>
```

```
              <xs:any namespace="##other" processContents="lax"
                      minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:token"
                          use="required"/>
            <xs:anyAttribute namespace="##any" processContents="lax"/>
          </xs:restriction>
        </xs:complexContent>
      </xs:complexType>

      <xs:element name="createContext" type="heldCx:createContextMsg"/>
      <xs:element name="updateContext" type="heldCx:updateContextMsg"/>
      <xs:element name="contextResponse" type="heldCx:contextResponseMsg"/>

    </xs:schema>
```

Figure 6: Context Management Schema

7. Security Considerations

TOC

There are several security concerns associated with the details in this specification. The first is to do with the nature of the sensitivity of any data passed from the Target to the LIS for inclusion in a context. The second is the ability of the LIS to contain the number of contexts that it will permit to exist for a given Target address. Finally, there is a threat of Targets performing DoS attacks on the LIS by trying to create large numbers of short-lived contexts that result in the LIS expending resources in message processing.

HELD [I-D.ietf-geopriv-http-location-delivery] (Barnes, M., Winterbottom, J., Thomson, M., and B. Stark, "HTTP Enabled Location Delivery (HELD)," August 2009.) mandates the use of TLS for exchanges between a Target and the LIS. This is deemed adequate to provide confidentiality to any contextual data in transit. The LIS implementation and the operator of the LIS need to take sufficient steps to ensure that active contextual data on the LIS is not readily available to anyone other than the Target. The Target MUST NOT provide any information to the LIS that it does not want the LIS to know or be able to use in some capacity associated with determination or providing of the Target's location.

It is quite conceivable that a LIS will be required to provide location to Targets residing behind a NAT; a DSL home router with 5 PCs attached is a good example this situation. In this case it is reasonable for each device to create its own context on the LIS, and for the LIS to treat each context individually even though the LIS cannot make any other distinction between the end hosts; that is, they share a common IP address/identity from the LIS perspective.

Given the constraints that can be added to a context and the way that a Target might want to manage expiry separately, a Target may use multiple contexts as a way to isolate applications from each other. For instance, a Target can create a context for each application so that it can revoke access to its location information for each without affecting other applications' access. This environment, however, opens the LIS to a type of denial of service attack through an overload of contexts. It is RECOMMENDED that an implementer of this specification include mechanisms to restrict to the maximum number of contexts that can be created on the LIS by an individual Target.

Using short-term location URIs in a carefully controlled manner may obviate the need for individual location authorization policies on the LIS. This leads to reduced LIS complexity and the amount of private information that the Target need share with the LIS. This specification provides the ability for a Target to cancel a location URI which extends the Target's ability to enforce its entitlement to privacy.

Using the mechanisms described in this memo a target can create URIs with short validity periods; this restricts how long a third-party is able to obtain the location of the Target while still allowing the Target the convenience of using a location reference.

The generation of context identifiers by the LIS is a critical component to supporting the functionality described in this memo. The LIS MUST follow the rules described in Section 5.5 (Location URI and Context Identifier Generation Rules) for generating context identifiers.

8. IANA Considerations

TOC

This document registers the schema and associated namespace with IANA.

8.1. URN Sub-Namespace Registration for urn:ietf:params:xml:ns:geopriv:held:context

TOC

This section registers a new XML namespace, urn:ietf:params:xml:ns:geopriv:held:context, as per the guidelines in [RFC3688] (Mealling, M., “The IETF XML Registry,” January 2004.).

> URI: urn:ietf:params:xml:ns:geopriv:held:context
>
> Registrant Contact: IETF, GEOPRIV working group, (geopriv@ietf.org), James Winterbottom (james.winterbottom@andrew.com).

XML:

```
           BEGIN
             <?xml version="1.0"?>
             <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
               "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
             <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
               <head>
                 <title>HELD Context Management Messages</title>
               </head>
               <body>
                 <h1>Namespace for HELD Context Management Messages</h1>
                 <h2>urn:ietf:params:xml:ns:geopriv:held:context</h2>
     [[NOTE TO IANA/RFC-EDITOR: Please update RFC URL and replace XXXX
         with the RFC number for this specification.]]
                 <p>See <a href="[[RFC URL]]">RFCXXXX</a>.</p>
               </body>
             </html>
           END
```

8.2. XML Schema Registration

TOC

This section registers an XML schema as per the guidelines in [RFC3688] (Mealling, M., “The IETF XML Registry,” January 2004.).

URI: urn:ietf:params:xml:schema:geopriv:held:context

Registrant Contact: IETF, GEOPRIV working group, (geopriv@ietf.org), James Winterbottom (james.winterbottom@andrew.com).

Schema: The XML for this schema can be found as the entirety of Figure 6 (Context Management Schema) of this document.

8.3. New HELD Error Code Registration

TOC

Reference: RFC-XXXX (i.e., this document)requires the following new HELD error codes to be added the HELD error code respository defined in [I-D.ietf-geopriv-http-location-delivery] (Barnes, M., Winterbottom, J., Thomson, M., and B. Stark, “HTTP Enabled Location Delivery (HELD),” August 2009.).

> Error code: badContextMessage
>
> Error code: unknownContext
>
> Error code: updateContextFailed
>
> Error code: createContextFailed

9. Acknowledgements

TOC

Thanks to Adam Muhlbauer and Neil Justusson for their comments on the pre-version of this draft.
Thanks also to Tim Zelinski and Michael Diponio , who pointed out a problems while implementing an early revision of this specification.

Appendix A. Context Extensions

TOC

A context contains specific information about a Target and is stored on the LIS. As with other protocols it is necessary to consider extensibility. When defining context data extensions it is necessary to consider how they will be used; this includes not only how to provide the information from the Target to the LIS, but also acceptance and error indications from the LIS back to the Target. For example, a context may be created with several extensions included, how does the LIS indicate that extensions 1 and 3 were successful but that extension 2 had a problem in its formatting? Guidelines for designing context extensions that provide functionality are described below.
Two basic types of context data extension are envisioned. The first consist of data provided by the Target to be consumed by the LIS; for example information pertaining to PIDF-LO construction, usage-rules, and authorization policies. The second type of data consists of a two way exchange between the Target and the LIS; for example exchanging location determination capabilities. Extensibility to the context scheme is to allow additional elements to be added to the context easily. The general idea is shown in Figure 7 (Create Context with Extensions).

```
  <hc:createContext
          xmlns:hc="urn:ietf:params:xml:ns:geopriv:held:context">
    <lifeTime>7200</lifeTime>
    <ex1:extension-1
          xmlns:ex1="urn:ietf:params:xml:ns:geopriv:held:ex1">
      <ex1:value>7200</ex1:value>
    </ex1:extension-1>
    <extension-2 xmlns="urn:ietf:params:xml:ns:geopriv:held:ex2"/>
    <extension-3 xmlns="urn:ietf:params:xml:ns:geopriv:held:ex3"/>
    .
    .
    .
    <extension-N xmlns="urn:ietf:params:xml:ns:geopriv:held:exN"/>
</hc:createContext>
```

Figure 7: Create Context with Extensions

When defining a context data extension it is necessary to ensure that the LIS can provide an adequate response to the Target indicating acceptance or rejection of the data provided. This may be an explicit OK or FAIL message within the extension namespace, it may be an attribute associated with part of a larger data exchange, or it may result in the LIS failing to create the context at all. Regardless, it is mandatory for a context data extension to provide an indication of success or failure.

```
  <hc:contextResponse
          xmlns:hc="urn:ietf:params:xml:ns:geopriv:held:context"
          code="created"
          id="uhvuhdbnuiehudbnvcujevuijeijcvij"
          uses="unlimited"
          snapshot="false"
          locType="any"
          expires="2007-08-01T13:00:00">
    <locationUriSet>
      <locationURI>
         held//ls.example.com:9768/357yc6s64ceyoiuy5ax3o
      </locationURI>
      <locationURI>
         sips:357yc6s64ceyoiuy5ax3o@ls.example.com:9769
      </locationURI>
    </locationUriSet>
    <ex1:extension-1 xmlns:ex1="urn:ietf:params:xml:ns:geopriv:held:ex1"
                  ex1:response="OK"/>
    <ex2:extension-2 xmlns:ex2="urn:ietf:params:xml:ns:geopriv:held:ex2"
                  ex2:response="OK"/>

    <ex3:extension-3
         xmlns:ex3="urn:ietf:params:xml:ns:geopriv:held:ex3">
      <datum-3>data</datum-3>
      <stuff>guff in here for extension</stuff>
    </ex3:extension-3>
</hc:contextRresponse>
```

Figure 8: LIS response to createContext

When defining information to be included in a context data extension consideration should be given to how that data can be removed from the context. In some cases it may be necessary to void the context on the LIS in order to remove information, but this SHOULD be treated as a last resort and not used as the primary mechanism for removing data from the context.

Appendix B. HELD Compliance to IETF Location Configuration Protocol Location Reference Requirements

TOC

This section describes how HELD and this specification comply to the LCP location reference requirements stipulated in [I-D.ietf-geopriv-lbyr-requirements] (Marshall, R., “Requirements for a Location-by-Reference Mechanism,” November 2009.).

High-level requirements for a location configuration protocol.

> **C1.** Location URI support - LCP: The configuration protocol MUST support a location reference in URI form.
>
> COMPLY. HELD only provides location references in URI form.
>
> **C2.** Location URI expiration: The LCP MUST support the ability to specify to the server, the length of time that a location URI will be valid.
>
> COMPLY. HELD with the context management extensions described in this document provide the Target the ability to specify expiry times for location URIs.
>
> **C3.** Location URI cancellation: The LCP MUST support the ability to request a cancellation of a specific location URI.
>
> COMPLY. HELD with the context management extensions described in this document provide the Target the ability to void location URIs when required.
>
> **C4.** Random Generated: The location URI MUST be hard to guess, i.e., it MUST contain a cryptographically random component.
>
> COMPLY. The HELD specification and this document provide specific guidance on the security surrounding location URI generation.
>
> **C5.** Identity Protection - LCP: The location URI MUST NOT contain any information that identifies the user, device or address of record within the URI form.
>
> COMPLY. The HELD specification and this document provide specific guidance on the anonymity of the Target with regards to the generation of location URIs.
>
> **C6.** Reuse flag default: The LCP MUST support the default condition of a requested location URI being repeatedly reused.
>
> COMPLY. HELD with the context management extensions described in this document provide the Target the ability to specify how many times a location URI may yield the location of Target.
>
> **C7.** One-time-use: The LCP MUST support the ability for the client to request a 'one-time-use' location URI (e.g., via a reuse flag setting).

> COMPLY. HELD with the context management extensions described in this document provide the Target the ability to specify how many times a location URI may yield the location of Target. This value may be set to 1 to create a one-time URI.

10. Normative References

TOC

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC3693]	Cuellar, J., Morris, J., Mulligan, D., Peterson, J., and J. Polk, "Geopriv Requirements," RFC 3693, February 2004 (TXT).
[RFC3688]	Mealling, M., "The IETF XML Registry," BCP 81, RFC 3688, January 2004 (TXT).
[I-D.ietf-geopriv-http-location-delivery]	Barnes, M., Winterbottom, J., Thomson, M., and B. Stark, "HTTP Enabled Location Delivery (HELD)," draft-ietf-geopriv-http-location-delivery-16 (work in progress), August 2009 (TXT).
[I-D.ietf-geopriv-l7-lcp-ps]	Tschofenig, H. and H. Schulzrinne, "GEOPRIV Layer 7 Location Configuration Protocol; Problem Statement and Requirements," draft-ietf-geopriv-l7-lcp-ps-10 (work in progress), July 2009 (TXT).
[I-D.ietf-geopriv-lbyr-requirements]	Marshall, R., "Requirements for a Location-by-Reference Mechanism," draft-ietf-geopriv-lbyr-requirements-09 (work in progress), November 2009 (TXT).

Authors' Addresses

TOC

	James Winterbottom
	Andrew Corporation
	PO Box U40
	University of Wollongong, NSW 2500
	AU
Phone:	+61 242 212938
Email:	james.winterbottom@andrew.com
URI:	http://www.andrew.com/products/geometrix
	Hannes Tschofenig

Nokia Siemens Networks
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany
Phone: +49 89 636 40390
Email: Hannes.Tschofenig@nsn.com
URI: http://www.tschofenig.com

Martin Thomson
Andrew Corporation
PO Box U40
University of Wollongong, NSW 2500
AU
Phone: +61 242 212915
Email: martin.thomson@andrew.com
URI: http://www.andrew.com/products/geometrix

Full Copyright Statement

TOC

This document and the information contained herein are provided on an “AS IS” basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.