

# Open Geospatial Consortium

Publication Date: 2015-08-19

Approval Date: 2015-07-13

Posted Date: 2015-06-25

Reference number of this document: OGC 15-068r2

Reference URL for this document: <http://www.opengis.net/doc/PER/tb11-geopackaging>

Category: Public Engineering Report

Editor(s): Gobe Hobona; Roger Brackin

## Testbed-11 GeoPackaging Engineering Report

Copyright © 2015 Open Geospatial Consortium.

To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

### Warning

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements.

Document type:	OGC® Engineering Report
Document subtype:	NA
Document stage:	Approved for public release
Document language:	English

## License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable

<b>Contents</b>	<b>Page</b>
1	Introduction..... 1
1.1	Scope ..... 1
1.2	Background ..... 1
1.3	Document contributor contact points ..... 2
1.4	Future work ..... 2
1.5	Forward ..... 2
2	References..... 2
3	Terms and definitions ..... 3
4	Conventions ..... 4
4.1	Abbreviated terms ..... 4
4.2	GeoPackage versus geopackage..... 4
5	Previous Work ..... 5
6	Methodology..... 5
7	Use Case..... 5
8	Implementation ..... 8
8.1	Component Descriptions ..... 8
8.1.1	Mobile Client 1: Luciad GeoPackage Synchronization Client..... 8
8.1.2	Mobile Client 2: Compusult GeoPackage Synchronization Client..... 10
8.1.3	Envitia GeoPackaging WPS ..... 10
8.1.4	CubeWerx Transactional WFS ..... 11
8.2	Creating geopackages on the WPS..... 11
8.3	Logging Changes..... 14
8.4	Sending Changes Back from the Client to the WPS ..... 18
9	Discussion..... 23
10	Conclusions..... 24
10.1	Recommendations ..... 24
10.1.1	Standardize how binary content is carried in OWS Context documents..... 24
10.1.2	Extend GeoPackage to Standardize Tables for Supporting Synchronization ..... 24
10.1.3	Develop a Best Practice for a GeoPackaging Profile of WPS..... 25
10.2	Revision history..... 26

<b>Figures</b>	Page
<b>Figure 1. Sequence diagram describing the geopackage synchronization .....</b>	<b>8</b>
<b>Figure 2. A screenshot of Mobile Client 1 implemented using LuciadMobile.....</b>	<b>9</b>
<b>Figure 3. A screenshot of Mobile Client 2 implemented using GO Mobile.....</b>	<b>10</b>
<b>Figure 4. A screenshot of the Envia Spatial Data Manager component in Envia Discovery .....</b>	<b>11</b>
<b>Figure 5. An example to illustrate synchronization of a single change .....</b>	<b>16</b>

<b>Tables</b>	Page
<b>Figure 1. Sequence diagram describing the geopackage synchronization .....</b>	<b>8</b>
<b>Figure 2. A screenshot of Mobile Client 1 implemented using LuciadMobile.....</b>	<b>9</b>
<b>Figure 3. A screenshot of Mobile Client 2 implemented using GO Mobile.....</b>	<b>10</b>
<b>Figure 4. A screenshot of the Envia Spatial Data Manager component in Envia Discovery .....</b>	<b>11</b>
<b>Figure 5. An example to illustrate synchronization of a single change .....</b>	<b>16</b>

## Abstract

Mobile location based service applications and users have an increasing need for access to geospatial data from any place in the world, including locations with limited or intermittent connectivity to communications networks. Maintaining consistency between copies of the same data held by different mobile devices can be a significant challenge when connectivity is limited or intermittent. This OGC Engineering Report describes the work carried out in OGC Testbed-11<sup>1</sup> in relation to the creation and synchronization of SQLite databases that conform to the OGC GeoPackage standard<sup>2</sup>. This Engineering Report describes an approach for the use of various standards to achieve such synchronization. The document also presents the results and lessons learnt from the experimentation conducted in the Testbed.

## Business Value

This OGC Engineering Report describes approaches that could improve interoperability by enhancing the ability of different geospatial software products to exchange and synchronize geospatial data. The recommendations contained in this Engineering Report are important to achieving interoperability in location-based technologies where mobile applications are often required to operate under conditions of limited or intermittent connectivity to communications networks.

## Keywords

ogcdocs, ogc document, testbed-11, geopackage, mobile, synchronization, versioning, ows context, sqlite, mobile, location services

---

<sup>1</sup> URL to TB 11 description is <http://www.opengeospatial.org/projects/initiatives/testbed11>

<sup>2</sup> URL to GeoPackage standard is <http://www.opengeospatial.org/standards/geopackage>



---

# Testbed-11 GeoPackaging Engineering Report

## 1 Introduction

The Urban Climate Resilience (UCR) thread of OGC Testbed-11 sought to respond to the need to make climate information and related data readily available for the public and government decision makers to prepare for changes in the Earth's climate. The GeoPackage activity in the UCR thread set out to explore geopackage synchronization while also advancing the storage of data into geopackages. The activity also set out to identify requirements for standardizing support for synchronization. Testbed participants successfully designed and implemented an approach to demonstrate the creation and synchronization of geopackages. The implementation of the approach has provided lessons and a potential schema for a GeoPackage extension to support synchronization. The participants have also identified an approach for documenting metadata that references the source of a dataset that has been packaged. These and related aspects of the UCR thread are the subject of this report.

### 1.1 Scope

The goal of this OGC Engineering Report is to describe the work carried out in the OGC Testbed-11 for the creation and synchronization of SQLite databases that conform to the OGC GeoPackage standard. Data synchronization can be defined as the process of establishing consistency of data between different systems and maintenance of that data over time. This document describes an approach for the packaging and synchronization of such databases. The document also presents the results and lessons learnt from the experimentation conducted in OGC Testbed 11.

The content of this document is applicable to initiatives involving mobile computing and other technologies that operate with intermittent or limited connection to communications networks.

### 1.2 Background

In recent years, the increasing popularity in location based services has led to a greater need for standardization of the mechanisms used to store geospatial data on mobile devices. The OGC responded to this need with the development of the GeoPackage standard [1]. The target database for GeoPackage is SQLite, thereby making a GeoPackage data package useable on mobile operating systems such as Android and Apple iOS.

Since approval of GeoPackage as an OGC standard, testbed activities relating to GeoPackage have progressed from standardizing geospatial data storage to exploring how geopackages could be generated using OGC web services such as the Web Processing Service (WPS). Testbed-11 builds upon the achievements of recent OGC testbeds by exploring how synchronization of geopackages can be achieved.

### 1.3 Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Name	Organization
Gobe Hobona PhD.	Envitia Ltd.
Roger Brackin	Envitia Ltd.
Chris Clark	Compusult
Robin Houtmeyers	Luciad
Peter Verlinden	Luciad
Ingo Simonis	Open Geospatial Consortium
Dave Wesloh	NGA
Adel Bolbol PhD.	Envitia Ltd.

### 1.4 Future work

No improvements to this document are planned.

### 1.5 Forward

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

## 2 References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.



1. OGC 12-128r11, OGC Geopackage Encoding Standard - With Corregendum
2. OGC 06-121r3, OpenGIS® Web Services Common Standard
3. OGC 12-119r1, OWS-9: OGC Mobile Apps: Definition, Requirements, and Information Architecture
4. OGC 14-058r1, Testbed-10 GeoPackage Engineering Report
5. OGC 12-080r2, OGC OWS Context Conceptual Model

### **3 Terms and definitions**

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Specification [2] and in OpenGIS® Abstract Specification shall apply. In addition, the following terms and definitions apply.

#### **3.1**

##### **change-only geopackage**

An instance of a geopackage that contains only the changes made since the data was checked out of the primary database.

#### **3.2**

##### **geopackage (all lowercase)**

An instance of a database that conforms to the GeoPackage standard.

#### **3.3**

##### **GeoPackage**

The OGC standard for a platform-independent SQLite database file that may contain vector geospatial features, tile matrix sets of raster maps at various scales and metadata.

#### **3.4**

##### **handheld mobile computing device**

A small, portable computing device, typically having a display screen with touch input and/or a miniature keyboard and weighing less than 2 pounds (0.91 kg). A handheld computing device has an operating system (OS), and can run various types of application software, known as apps. Definition adapted from OGC 12-119r1 [3].

#### **3.5**

##### **interoperability**

Capability to communicate, execute programs or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units.

#### **3.6**

##### **mobile application**

Mobile applications (mobile apps) are software products developed for mobile devices which may include handheld devices, such as smart phones, or wearable devices, such as

smart watches. These applications can be pre-installed on the devices during manufacture, downloaded by customers from various mobile software distribution platforms, or delivered as web applications using server-side or client-side processing.

### **3.7**

#### **wearable mobile computing device**

A mobile computer that is worn on the body and forms part of one's clothing.

## **4 Conventions**

### **4.1 Abbreviated terms**

UCR	Urban Climate Resilience
ER	Engineering Report
GML	Geography Markup Language
GSS	GeoSynchronization Service
IETF	Internet Engineering Task Force
MIME	Multi-Purpose Internet Mail Extensions
NSG	National System for Geospatial Intelligence
OGC	Open Geospatial Consortium
SDI	Spatial Data Infrastructure
URL	Uniform Resource Locator
WFS	Web Feature Service
WPS	Web Processing Service

### **4.2 GeoPackage versus geopackage**

This Engineering Report uses the term “GeoPackage” to refer to the standard and the term “geopackage” to refer to an instance of a database that conforms to the standard. This convention is intended to make it easier for readers of this report and does not reflect an official OGC convention.

## 5 Previous Work

OGC Testbed (Testbed-10) experimented with using a Web Processing Service (WPS) to create a geopackage and then developed a GeoSynchronization Service (GSS) to keep the data within the geopackage up-to-date [4].

Experimentation within Testbed-10 examined the creation of geopackages as a special case of synchronization. This approach is the user case where moving an entire dataset from one system to another the process effectively resembles a synchronization activity where one system does not have any data.

The Testbed participants discovered, however, that the approach could be problematic because synchronization requires more than just data packaging and transfer. The participants concluded that treating synchronization and creation of geopackages as separate functions was the best approach to take. This lesson was carried into the GeoPackage activities in OGC Testbed-11.

OGC Testbed-10 activity resulted in additional observations relating to the GeoPackage standard, such as the need for a GeoPackage Multi-purpose Internet Mail Extension (MIME) type and the need for standardizing where bounding boxes of raster tiles are specified in geopackages. These observations have also informed the activities of OGC Testbed-11.

## 6 Methodology

To achieve the objectives for the GeoPackage activity, the experimentation conducted in Testbed-11 followed the following process:

1. Review data sources
2. Identify data and services to support the use case
3. Configure individual components to support synchronization
4. Assess interoperability and identify lessons to learn

## 7 Use Case

Testbed-11 implemented a use case to explore synchronization of geopackages. The use case was designed with situations of limited or intermittent connectivity in mind. In such situations, a snapshot of an offline database is required. A scenario where field workers may download data before going out into the field is used. Once in the field, they work on local copies of the data. If two field workers meet in the field, data needs to be synchronized between the two using a direct device-to-device connection. Once back home, field workers can synchronize their local copies with the enterprise system. The use case applied in this scenario is as follows:

Actors:

- Mobile Client 1
- Mobile Client 2
- GeoPackaging WPS (includes synchronization capabilities)
- Transactional Web Feature Service (WFS)

Phase 1 Basic Steps:

1. Mobile Client 1 sends a request to the GeoPackaging WPS to package data from the WFS.
2. GeoPackaging WPS sends a request to the referenced WFS for the initial version of the data
3. WFS returns a response, containing GML-encoded data, back to the GeoPackaging WPS
4. GeoPackaging WPS returns a response, containing GeoPackage-encoded data, back to the Mobile Client 1

Phase 2 Basic Steps:

1. Mobile Client 2 sends a request to the GeoPackaging WPS to package data from the WFS.
2. GeoPackaging WPS sends a request to the referenced WFS for the initial version of the data
3. WFS returns a response, containing GML-encoded data, back to the GeoPackaging WPS
4. GeoPackaging WPS returns a response, containing GeoPackage-encoded data, back to the Mobile Client 2

Phase 3 Basic Steps (Peer-to-Peer Synchronization):

1. Mobile Client 1 updates the locally stored data
2. Mobile Client 2 sends a request for changed data to Mobile Client 1
3. Mobile Client 1 returns a change-only geopackage to Mobile Client 2

4. Mobile Client 2 updates their locally stored data

Phase 4 Basic Steps (Peer-to-Enterprise Synchronization):

1. Mobile Client 1 updates the locally stored data
2. Mobile Client 1 sends a request to the GeoPackaging WPS to upload changes into a transactional WFS
3. GeoPackaging WPS reads the geopackage received from Mobile Client 1
4. For each change registered in the geopackage received from Mobile Client 1, the GeoPackaging WPS sends transaction requests to the transactional WFS
5. The transactional WFS returns the result of each transaction to the GeoPackaging WFS
6. GeoPackaging WPS returns the results of the transactions to Mobile Client 1
7. Mobile Client 2 retrieves the updated data

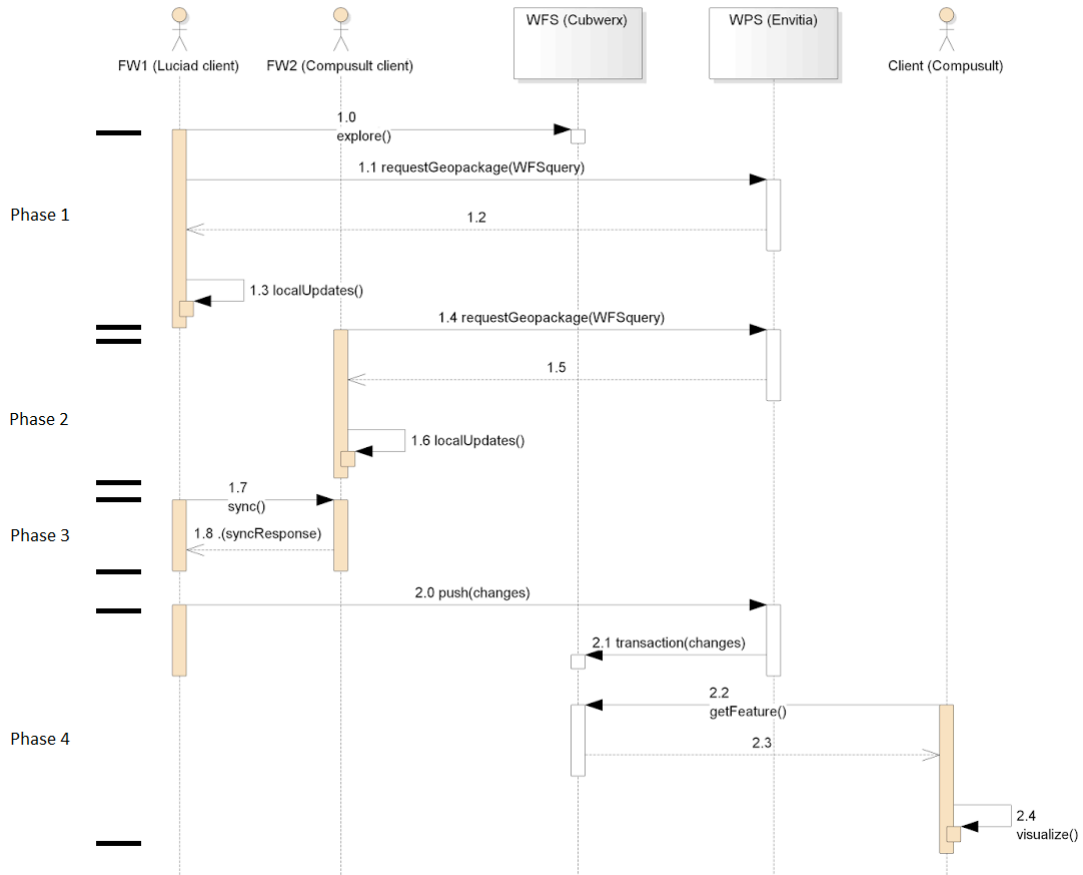


Figure 1. Sequence diagram describing the geopackage synchronization

## 8 Implementation

This section describes the implementation issues encountered.

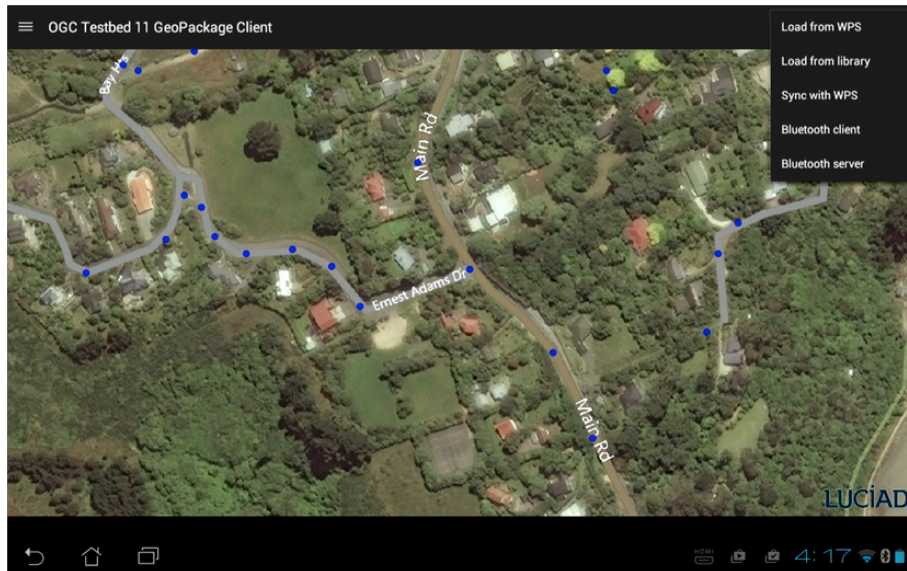
### 8.1 Component Descriptions

The following components were used in this aspect of the testbed.

#### 8.1.1 Mobile Client 1: Luciad GeoPackage Synchronization Client

Mobile Client 1 was implemented using the Luciad GeoPackage synchronization client which offered integration and testing of the GeoPackage production and synchronization services. The client also offered integration and testing of the implementation of the synchronization use case. The client is based on Luciad’s Commercial Off the Shelf (COTS) product LuciadMobile, which is targeted towards mobile devices running the

Android operating system. A screenshot of the client implement for Testbed 11 is shown in Figure 2.



**Figure 2. A screenshot of Mobile Client 1 implemented using LuciadMobile**

The product offers a set of capabilities that were used directly in Testbed 11, for example a 2D geospatial map with map interaction controllers, support for the visualization of raster data (images & elevation) and vector data served by a variety of data sources, as well as support for OGC GeoPackage. Additional functionality was implemented to address the GeoPackage synchronization requirements in the Urban Client Resilience (UCR) thread<sup>3</sup>:

- Support for OWS Context [5] to describe data sources.
- Client interface to connect with Envia's GeoPackaging WPS, to request GeoPackage data based on a data source description in OWS Context.
- Client interface to connect with Envia's GeoPackaging WPS, to push back any local GeoPackage data changes to the server.
- Client interface to synchronize local GeoPackage data with another mobile device over Bluetooth. This relied on an Android Bluetooth communication library provided by Compuconsult.
- Implementation of the relevant use case.

<sup>3</sup> [https://portal.opengeospatial.org/files/?artifact\\_id=60889](https://portal.opengeospatial.org/files/?artifact_id=60889)

### 8.1.2 Mobile Client 2: CompuSULT GeoPackage Synchronization Client

Mobile Client 2 was implemented using the CompuSULT GO Mobile client to integrate and test the GeoPackage production and synchronization services. GO Mobile is a mobile application for devices such as the iPhone, iPad, Android phones and Android-based tablets. The client offers users the ability to access products through geospatial web services. It also provides users with the option of sharing information with others through the mobile device, as well as enabling officials to use their devices for Situational Awareness and Common Operating Picture functions. A screenshot of the client implement for Testbed 11 is shown in Figure 3.



**Figure 3. A screenshot of Mobile Client 2 implemented using GO Mobile**

A number of capabilities of the product were demonstrated in the testbed, for example:

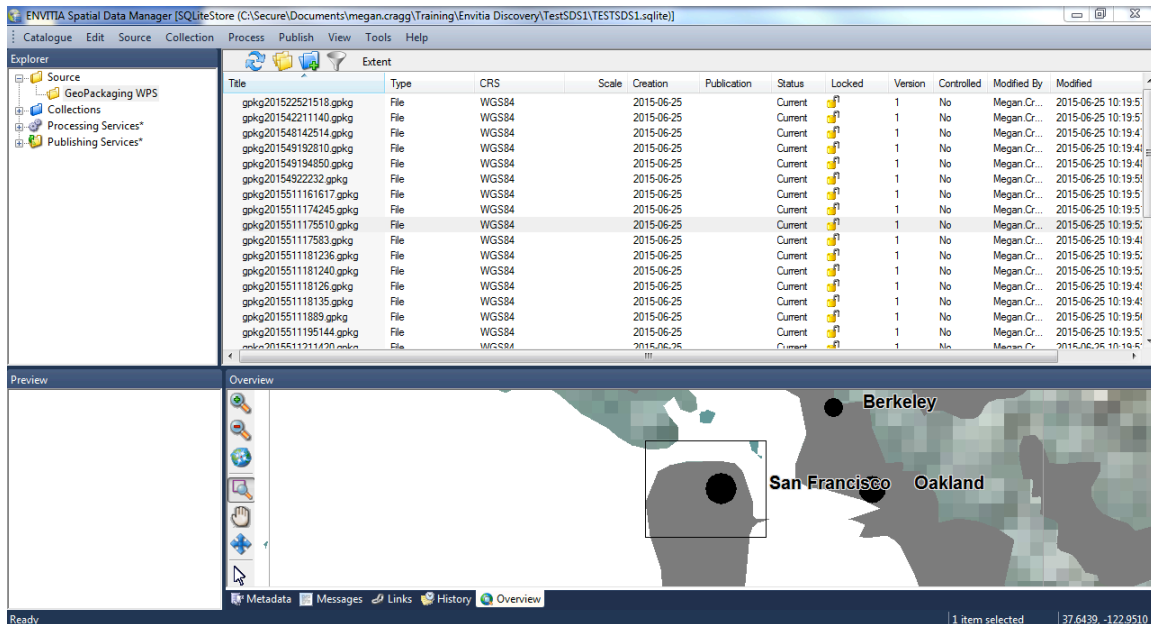
- Viewing WFS features and metadata
- Viewing GeoPackage features and metadata
- Editing GeoPackage feature metadata
- Peer to Peer synchronization of geopackages

### 8.1.3 Envitia GeoPackaging WPS

The GeoPackaging Web Processing Service (WPS) requirement was implemented using Envitia MapLink Pro, a visualization toolkit that can also be deployed in a web server to offer a WPS. Upon receipt of an OWS Context document, the WPS creates a geopackage that contains the data referenced in the OWS Context document. On the server-side each



generated geopackage can be persisted to allow for access at a later point in time. The Envitia Spatial Data Manager shown in Figure 4, which is part of Envitia Discovery, is used to monitor and manage geopackages persisted in the WPS. The GeoPackaging WPS was extended to offer a synchronization process as described later in this report. The interface to the synchronization process was extended to allow for ingest of requests that contain an OWS Context document embedded within a request rather than the document referenced from a remote server, as had been implemented in the previous testbed.



**Figure 4.** A screenshot of the Envitia Spatial Data Manager component in Envitia Discovery

To support synchronization, the GeoPackaging WPS was extended to allow for processing geopackages received as base64 binary embedded within an OWS Context document as text. The service was also extended to enable the automated creation and storage of sample ISO 19115 metadata within a geopackage. The service was configured to allow for responses encoded as geopackages (i.e. with the MIME type of application/vnd.ogc.gpkg).

#### 8.1.4 CubeWerx Transactional WFS

The transactional WFS was implemented using the CubeWerx CubeSERV WFS. The product supports transactional operations such as INSERT, UPDATE, DELETE and LOCK operations. As well as transactional operations, the service also supports query operations.

## 8.2 Creating geopackages on the WPS

The following example request triggers a GeoPackaging process. The request returns a hyperlink to a geopackage that has been generated and stored on the server. The request

uses the wps:Output element with the asReference attribute set to "true" in order to produce the hyperlinked output.

```

<wps:Execute service="WPS" version="1.0.0"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
  <ows:Identifier>GeoPackaging</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>OWSContext</ows:Identifier>
      <ows:Title>OWS Context document</ows:Title>
      <wps:Data>
        <wps:ComplexData mimeType="text/xml">
          <feed xmlns="http://www.w3.org/2005/Atom"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:georss="http://www.georss.org/georss"
xmlns:gml="http://www.opengis.net/gml"
xmlns:owc="http://www.opengis.net/owc/1.0"
xml:lang="en">
            <category
scheme="http://www.opengis.net/spec/owc/specReference"
term="http://www.opengis.net/spec/owc/1.0/req/atom"
label="This file is compliant with version 1.0 of OGC Context"/>
              <title>OGC Testbed Sample Requests</title>
              <id>http://www.opengeospatial.org</id>
              <author>
                <name>OGC</name>
              </author>
              <updated>2012-02-03T12:00:30Z</updated>
              <dc:publisher>OGC</dc:publisher>
              <entry>
                <id>http://127.0.0.1:8080/geoserver/tb11/ows/tb11/edges</id>
                <title>tb11_edges</title>
                <dc:publisher>Envitia</dc:publisher>
                <updated>2012-02-03T12:00:30Z</updated>
                <owc:offering
code="http://www.opengis.net/spec/owc-atom/1.0/req/wfs">
                  <owc:operation method="GET"
code="GetFeature"
href="http://127.0.0.1:8080/geoserver/tb11/ows?service=WFS&
version=1.1.0&request=GetFeature&
typeName=tb11:edges&maxFeatures=100"/>
                </owc:offering>
              </entry>
            </feed>
          </wps:ComplexData>
        </wps:Data>
      </wps:Input>
    </wps:DataInputs>
  </wps:Execute>

```

```

<wps:ResponseForm>
  <wps:ResponseDocument storeExecuteResponse="false">
    <wps:Output asReference="true">
      <ows:Identifier>GeoPackage</ows:Identifier>
    </wps:Output>
  </wps:ResponseDocument>
</wps:ResponseForm>
</wps:Execute>

```

An alternative approach is to return the geopackage as the actual response (i.e. with the MIME type of application/vnd.ogc.gpkg). This alternative approach uses the wps:RawDataOutput element as illustrated below:

```

<wps:Execute service="WPS" version="1.0.0"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
  <ows:Identifier>GeoPackaging</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>OWSContext</ows:Identifier>
      <ows:Title>OWS Context document</ows:Title>
      <wps:Data>
        <wps:ComplexData mimeType="text/xml">
          <feed xmlns="http://www.w3.org/2005/Atom"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:georss="http://www.georss.org/georss"
xmlns:gml="http://www.opengis.net/gml"
xmlns:owc="http://www.opengis.net/owc/1.0"
xml:lang="en">
            <category
scheme="http://www.opengis.net/spec/owc/specReference"
term="http://www.opengis.net/spec/owc/1.0/req/atom"
label="This file is compliant with version 1.0 of OGC Context"/>
              <title>OGC Testbed Sample Requests</title>
              <id>http://www.opengeospatial.org</id>
              <author>
                <name>OGC</name>
              </author>
              <updated>2012-02-03T12:00:30Z</updated>
              <dc:publisher>OGC</dc:publisher>
            </entry>

          <id>http://127.0.0.1:8080/geoserver/tb11/ows/tb11/edges</id>
            <title>tb11_edges</title>
            <dc:publisher>Envitia</dc:publisher>
            <updated>2012-02-03T12:00:30Z</updated>
            <owc:offering
code="http://www.opengis.net/spec/owc-atom/1.0/req/wfs">

```

```

                <owc:operation method="GET"
code="GetFeature"
href="http://127.0.0.1:8080/geoserver/tb11/ows?service=WFS&
version=1.1.0&request=GetFeature&
typeName=tb11:edges&maxFeatures=100"/>
                </owc:offering>
            </entry>
        </feed>
    </wps:ComplexData>
</wps>Data>
</wps:Input>
</wps>DataInputs>
<wps:ResponseForm>
    <wps:RawDataOutput mimeType="application/vnd.ogc.gpkg">
        <ows:Identifier>GeoPackage</ows:Identifier>
    </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

### 8.3 Logging Changes

The following model is used by the WPS for logging changes and keeping track of revisions.

```

CREATE TABLE "gpkg_change_log" (
    "change_id" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    "table_name" TEXT NOT NULL,
    "feature_uuid" TEXT NOT NULL,
    "change_type" TEXT NOT NULL,
    "change_date" DATETIME NOT NULL DEFAULT (strftime('%Y-%m-%dT%H:%M:%fZ', 'now')),
    "revision_uuid" TEXT
);

CREATE TABLE "gpkg_feature_uuids" (
    "table_name" TEXT NOT NULL,
    "feature_id" INTEGER NOT NULL,
    "feature_uuid" TEXT NOT NULL PRIMARY KEY
);

CREATE TABLE "gpkg_revisions" (
    "revision_uuid" TEXT NOT NULL,
    "parent_revision_uuid" TEXT
);

```

The following is an example insert statement for logging an update to a feature. Within the scenario and use case described in Section 7, the change would be logged by the

client application. When an existing feature is modified the change type is set to 'UPDATE' and when a new feature is created the change type is set to 'INSERT'.

```
INSERT INTO gpkg_change_log (change_id, table_name, feature_uuid,
change_type, change_date) VALUES (0, 'moz__maputo__osm__edgesedgesType',
'dacb87d3-2f7f-4841-a092-80fdb369210', 'UPDATE', '2015-03-12
10:01:00');

INSERT INTO gpkg_feature_uuids (table_name, feature_id, feature_uuid)
VALUES ('moz__maputo__osm__edgesedgesType', 0, 'dacb87d3-2f7f-4841-
a092-80fdb369210');
```

In the previous example `gpkg_change_log.revision_uuid` is left 'null' until synchronization occurs. When a geopackage is first created it is given an initial `revision_uuid` with no `parent_revision_uuid`, as illustrated in the following SQL statement.

```
INSERT INTO gpkg_revisions (revision_uuid) VALUES ('41cc06c5-e954-4976-
9f9a-e55a3a2a208e');
```

When synchronization occurs, a new `revision_uuid` is generated and `parent_revision_uuid` becomes the latest `revision_uuid` from the table, as illustrated in the following SQL statement.

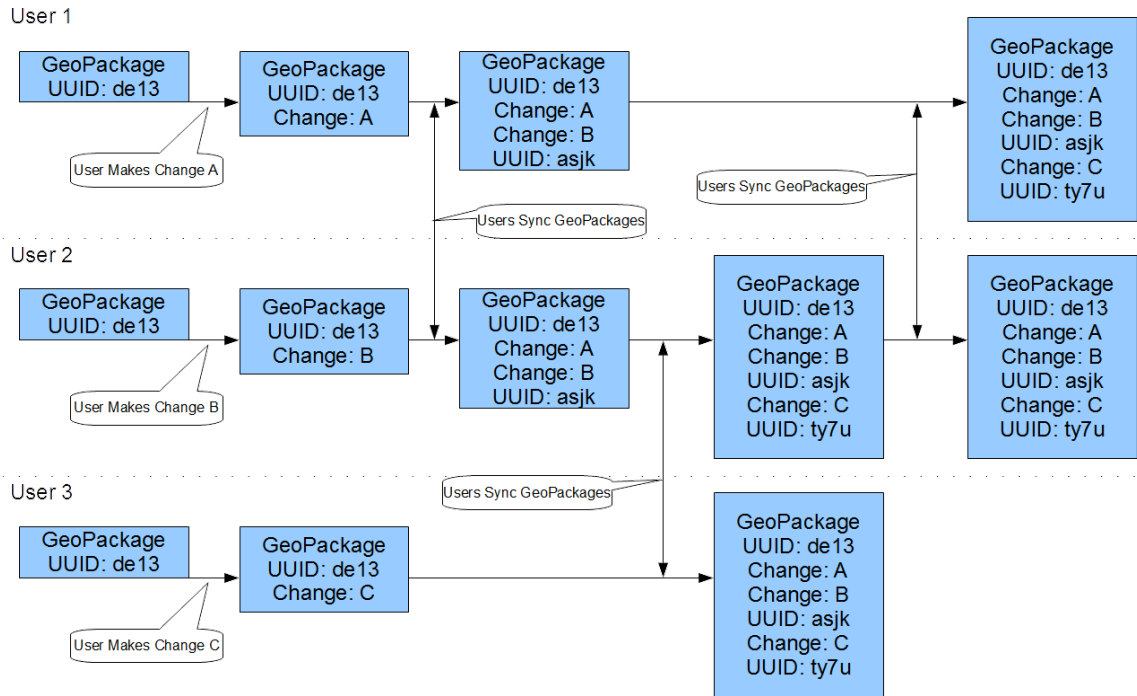
```
INSERT INTO gpkg_revisions (revision_uuid, parent_revision_uuid) VALUES
('4200197c-e2c9-4b62-91b4-8c0f516526e5', '41cc06c5-e954-4976-9f9a-
e55a3a2a208e');
```

All outstanding changes in the `gpkg_change_log` are now marked with this `revision_uuid`, since they have all been synchronized, as illustrated in the following SQL statement.

```
UPDATE gpkg_change_log SET revision_uuid = '4200197c-e2c9-4b62-91b4-
8c0f516526e5' WHERE revision_uuid IS NULL;
```

Please note that multiple changes in `gpkg_change_log` table can have the same `revision_uuid` (i.e. all the changes that need to be synchronized as part of the same revision).

The application of the change logging is further explained through the illustration in Figure 5. The illustration follows the lifecycle of a single change as it is created and synchronized between multiple devices. For ease of reading, the UUIDs have been shortened to four characters.



**Figure 5. An example to illustrate synchronization of a single change**

The following tables document how the identifiers are used to record changes and revisions.

An example of the `gpkg_revisions` table, at the time the geopackage is created, is shown in Table 1. Note that the `parent_revision_uuid` value is ‘null’.

**Table 1. `gpkg_revisions` example at creation**

revision_uuid	parent_revision_uuid
de13	

At the time of geopackage creation, the `gpkg_change_log` table does not have any entries. This is because none of the features in the geopackage have been modified at this stage. Table 2 illustrates this table at the time the geopackage is created.

**Table 2. `gpkg_change_log` table example at creation**

change_id	table_name	table_version	feature_id	feature_version	change_type	change_date	revision_uuid

When a change is made to a feature, an entry is inserted into the `gpkg_change_log` table. The `revision_uuid` is set to null at this stage, because the geopackage has not yet been synchronized. This is illustrated in Table 3.

**Table 3. `gpkg_change_log` table example at data entry**

<code>change_id</code>	<code>table_name</code>	<code>table_version</code>	<code>feature_id</code>	<code>feature_version</code>	<code>change_type</code>	<code>change_date</code>	<code>revision_uuid</code>
1	tiger_poly_landmarks	1	6	1	UPDATE	2015-03-13T14:06:00.030Z	

When the first device (held by user 1) comes into contact with the second device (held by user 2), the changes made to the geopackage in the second device are transferred to the first device. As illustrated in Table 4, a new `revision_uuid` is then issued and assigned the value of the previous `revision_uuid` as the `parent_revision_uuid`.

**Table 4. `gpkg_revisions` example at data entry**

<code>revision_uuid</code>	<code>parent_revision_uuid</code>
de13	
asjk	de13

After synchronization the `revision_uuids` in the `gpkg_change_log` table are updated to identify which revision the changes have been applied to. This is illustrated in Table 5.

**Table 5. `gpkg_change_log` table example at data entry**

<code>change_id</code>	<code>table_name</code>	<code>table_version</code>	<code>feature_id</code>	<code>feature_version</code>	<code>change_type</code>	<code>change_date</code>	<code>revision_uuid</code>
1	tiger_poly_landmark	1	6	1	UPDATE	2015-03-13T14:06:00.030Z	asjk
2	tiger_poly_landmark	2	5	1	UPDATE	2015-03-13T14:08:00.030Z	asjk

## 8.4 Sending Changes Back from the Client to the WPS

Once the changes have been logged, the client application can create a copy of the geopackage with only the changed features included. This “change-only geopackage” then needs to be sent from the Client application to the WPS. To enable the embedding of a geopackage inside the payload of an OWS Context document, the change-only geopackage was encoded as Base64 text. This encoding may be referred to as ‘base64’.

The Internet Engineering Task Force (IETF) offers the following description of base64: “The base64 encoding is designed to represent arbitrary sequences of octets in a form that allows the use of both upper- and lowercase letters but that need not be human readable....A 65-character subset of US-ASCII is used, enabling 6 bits to be represented per printable character....The encoding process represents 24-bit groups of input bits as output strings of 4 encoded characters. Proceeding from left to right, a 24-bit input group is formed by concatenating 3 8-bit input groups. These 24 bits are then treated as 4 concatenated 6-bit groups, each of which is translated into a single character in the base 64 alphabet”<sup>4</sup>.

The following is an example request to synchronize changes from a change-only geopackage to a transactional WFS.

```
<wps:Execute service="WPS" version="1.0.0"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
  <ows:Identifier>GeoSynchronize</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>OWSContext</ows:Identifier>
      <ows:Title>OWS Context document</ows:Title>
      <wps>Data>
        <wps:ComplexData mimeType="text/xml">
          <feed xmlns="http://www.w3.org/2005/Atom"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:georss="http://www.georss.org/georss"
xmlns:gml="http://www.opengis.net/gml"
xmlns:owc="http://www.opengis.net/owc/1.0"
xml:lang="en">
            <category scheme="http://www.opengis.net/spec/owc/specReference"
term="http://www.opengis.net/spec/owc/1.0/req/atom"
```

<sup>4</sup> <https://www.ietf.org/rfc/rfc2045.txt>



```

label="This file is compliant with version 1.0 of OGC Context"/>
  <title>OGC Testbed Sample Requests</title>
  <id>http://www.opengeospatial.org</id>
  <author>
    <name>OGC</name>
  </author>
  <updated>2012-02-03T12:00:30Z</updated>
  <dc:publisher>OGC</dc:publisher>

  <entry>
    <id>http://www.opengeospatial.org/testbed-11/edges</id>
    <title>edges</title>
    <dc:publisher>Envitia</dc:publisher>
    <updated>2012-02-03T12:00:30Z</updated>
    <owc:offering code="http://www.opengis.net/spec/owc/1.0/conf/atom/gpkg">
    <owc:operation method="POST" code="Transaction"
href="http://127.0.0.1:8080/geoserver/wfs"/>
      <owc:content type="application/vnd.ogc.gpkg">
U1FMaXRlIGZvcmlhdCAzAAQAAQEAQCAGAAAACgAAAewAAAAAAAAAAAAAAAAABIAAAAEAAAAAAAAAA
AAAAABAAAAAAAAAAABHUDEwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAJAC36QEAc6-
I2GvA5wGB2Tx4WOkBAx7q4jQbwOcCBLGANAB8jExMtDw8vL4EuMTA2NDY
IQP7A_YD8QPsA__SO85wDaDDgUAISFGMTIxNvDY0Njk0ODlyEyMTU0NDg
EIrBw6QEI2GvA5wG...Content Shortened for Readability...8vDY0jQbwOcCBLGANAjQbw
Njk0ODlyZXNpZGVudG1hbDU3NTU3MSoqK
iBVbmtub3duICoqKioxMTM4LjE3NTc0MzcyMDkwODU3NC41ODEwODkyNDY1MDkwMjhHUAAD5hA
QGB2Tx4WOkBAx7q4jQbwOcDkFB3J5e85wAECAAAAAGAAAGB2T
x4WOkBAx7q</owc:content>
      </owc:offering>
    </entry>
  </feed>

  </wps:ComplexData>
  </wps>Data>
</wps:Input>
</wps>DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml">
      <ows:Identifier>Result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>

```

The Testbed participants examined different approaches for directing the changes from the geopackage to a WFS. The first approach determines the namespace and feature type name from the OWS Context entry. The second approach determines the namespace and feature type name from the gpkg\_metadata table. After discussion by the Testbed participants, it was determined that the first approach would not be appropriate because it would limit updates to a single feature type per OWS Context entry (potentially leading to duplication of entries). The second approach was therefore selected for the testbed. A sample metadata document describing a feature dataset that has been packaged is shown in the following example:

```

<MD_Metadata xmlns="http://www.isotc211.org/2005/gmd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:gco="http://www.isotc211.org/2005/gco"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.isotc211.org/2005/gmd
../../../../../../../../iso/19139/20060504/gmd/gmd.xsd"
uuid="US_NGA_VPF_VMAP0" id="VMAP0">
  <fileIdentifier>
    <gco:CharacterString>2345-aa002-edges57</gco:CharacterString>
  </fileIdentifier>
  <language>
    <gco:CharacterString>eng</gco:CharacterString>
  </language>
  <hierarchyLevel>
    <MD_ScopeCode
codeList="http://www.isotc211.org/2005/....codeList.xml#MD_ScopeCode"
codeListValue="dataset"/>
  </hierarchyLevel>
  <contact>
    <CI_ResponsibleParty>
      <organisationName>
        <gco:CharacterString>OGC</gco:CharacterString>
      </organisationName>
      <role>
        <CI_RoleCode
codeList="http://www.isotc211.org/2005/...codeList.xml#CI_RoleCode"
codeListValue="publisher"/>
      </role>
    </CI_ResponsibleParty>
  </contact>
  <!-- Date the metadata was created -->
  <dateStamp>
    <gco:Date>2004-03-14</gco:Date>
  </dateStamp>
  <metadataStandardName>
    <gco:CharacterString>ISO19115</gco:CharacterString>
  </metadataStandardName>
  <metadataStandardVersion>
    <gco:CharacterString>2003/Cor.1:2006</gco:CharacterString>
  </metadataStandardVersion>
  <!-- IDENTIFICATION INFORMATION -->
  <identificationInfo>
    <MD_DataIdentification>
      <citation>
        <CI_Citation>
          <title>
            <gco:CharacterString>Maputo Road Network Edges</gco:CharacterString>
          </title>
          <date>
            <CI_Date>
              <date>
                <gco:Date>2000-09-03</gco:Date>
              </date>
            </CI_Date>
          </date>
        </CI_Citation>
      </citation>
    </MD_DataIdentification>
  </identificationInfo>

```

```

        </date>
        <dateType>
            <CI_DateTypeCode
codeList="http://www.isotc211.org/2005/....codeList.xml#CI_DateTypeCode"
codeListValue="publication"/>
        </dateType>
        </CI_Date>
    </date>
    <identifier>
        <MD_Identifier>
            <code>
                <gco:CharacterString>
{http://namespacedomain/context}moz__maputo__osm__edgesedgesType
            </gco:CharacterString>
        </code>
        </MD_Identifier>
    </identifier>
    </CI_Citation>
</citation>
<abstract>
    <gco:CharacterString>
Maputo road transportation network edges generated from Open Street Map
</gco:CharacterString>
</abstract>
<pointOfContact>
    <CI_ResponsibleParty id="ID00001">
        <role>
            <CI_RoleCode
codeList="http://www.isotc211.org/2005/....codeList.xml#CI_RoleCode"
codeListValue="publisher"/>
        </role>
    </CI_ResponsibleParty>
</pointOfContact>
<language>
    <gco:CharacterString>eng</gco:CharacterString>
</language>
<topicCategory>
    <MD_TopicCategoryCode>transportation</MD_TopicCategoryCode>
</topicCategory>
<extent>
    <EX_Extent>
        <geographicElement>
            <EX_GeographicBoundingBox>
                <westBoundLongitude>
                    <gco:Decimal>31.9999</gco:Decimal>
                </westBoundLongitude>
                <eastBoundLongitude>
                    <gco:Decimal>32.7371</gco:Decimal>
                </eastBoundLongitude>
                <southBoundLatitude>
                    <gco:Decimal>-26.3013</gco:Decimal>
                </southBoundLatitude>
                <northBoundLatitude>
                    <gco:Decimal>-25.5537</gco:Decimal>
                </northBoundLatitude>
            </EX_GeographicBoundingBox>

```

```

        </geographicElement>
    </EX_Extent>
</extent>
</MD_DataIdentification>
</identificationInfo>
<distributionInfo>
    <MD_Distribution>
        <distributor>
            <MD_Distributor>
                <distributorContact>
                    <CI_ResponsibleParty id="ID00010">
                        <organisationName>
                            <gco:CharacterString>OGC</gco:CharacterString>
                        </organisationName>
                        <role>
                            <CI_RoleCode
codeList="http://www.isotc211.org/2005/...codeList.xml#CI_RoleCode"
codeListValue="publisher"/>
                            </role>
                        </CI_ResponsibleParty>
                    </distributorContact>
                    <distributorTransferOptions>
                        <MD_DigitalTransferOptions>
                            <onLine>
                                <CI_OnlineResource>
                                    <linkage>
<URL>http://127.0.0.1:8080/geoserver/tb11/ows?service=WFS&
version=1.0.0&request=GetFeature&type=tb11:edges</URL>
                                    </linkage>
                                    <function>
                                        <CI_OnLineFunctionCode
codeList="http://www.isotc211.org/2005/...codeList.xml#CI_OnLineFunctionCode"
codeListValue="download"/>
                                        </function>
                                    </CI_OnlineResource>
                                </onLine>
                            </MD_DigitalTransferOptions>
                        </distributorTransferOptions>
                    </MD_Distributor>
                </distributor>
            </MD_Distribution>
        </distributionInfo>
    </MD_Metadata>

```

An explanation of the relevant information in the metadata is shown below.

Name	XPath	Role in gpkg_metadata
feature type identifier	MD_Metadata/identificationInfo/MD_DataIdentification/citation/CI_Citation	The namespace-qualified identifier of the feature type that

	/identifier/MD_Identifier/code	the feature dataset came from.
west bounding coordinate	MD_Metadata/identificationInfo/MD_DataIdentification/extent/EX_Extent/geographicElement/EX_GeographicBoundingBox/westBoundLongitude	The longitude furthest west.
south bounding coordinate	MD_Metadata/identificationInfo/MD_DataIdentification/extent/EX_Extent/geographicElement/EX_GeographicBoundingBox/southBoundLatitude	The latitude furthest South.
east bounding coordinate	MD_Metadata/identificationInfo/MD_DataIdentification/extent/EX_Extent/geographicElement/EX_GeographicBoundingBox/eastBoundLongitude	The longitude furthest East.
north bounding coordinate	MD_Metadata/identificationInfo/MD_DataIdentification/extent/EX_Extent/geographicElement/EX_GeographicBoundingBox/northBoundLatitude	The latitude furthest North.
resource linkage	MD_Metadata/distributionInfo/MD_Distribution/distributor/MD_Distributor/distributorTransferOptions/MD_DigitalTransferOptions/onLine/CI_OnlineResource/linkage/URL	The URL from which the feature dataset can be downloaded. Notice that in this case the URL is a GetFeature KVP request

## 9 Discussion

This Testbed activity set out to explore synchronization while also advancing the packaging of data into geopackages. The successful implementation of a GeoPackaging WPS that is configured through an OWS Context document supported the assertion that OWS Context documents are appropriate for triggering processes to package geospatial data. However, the activity also highlighted the need for standardizing how binary content, such as a geopackage itself, can be carried as a payload in an OWS Context document. The testbed successfully demonstrated how this could be done using base64 and thus an important lesson identified in this thread is that there is a need to standardize how binary content is carried in OWS Context documents.

The testbed also set out to identify requirements for standardizing support for synchronization. The successful implementation of software components to demonstrate a synchronization use case supported the identification of a table schema to support the synchronization. The testbed designed and implemented the proposed tables that include

a table for logging changes and a set of associated tables to uniquely identify revisions. The approach was successfully applied to synchronization between a mobile device and an enterprise server (WFS), as well as being applied between two different mobile devices. An important lesson identified in this aspect of the testbed was that an extension to the GeoPackage standard that specifies tables for logging changes and identifying revisions is required to standardize synchronization support in geopackages.

In addition to identifying tables for recording changes, the testbed also considered and identified an approach for documenting metadata that references the source of a dataset that has been packaged. The components were configured to support such metadata and the testbed demonstrated how these enabled the use case described in this report.

## **10 Conclusions**

This Engineering Report describes the work carried out in the UCR thread in OGC Testbed-11 to advance the packaging and synchronization of geospatial data into SQLite databases that conform to the OGC GeoPackage standard. Also presented are the results and lessons identified from the experimentation conducted by the testbed. The UCR Thread participants conclude that web services based on the WPS standard can be configured to produce geopackages according to specifications of resources provided by OWS Context documents. Further, base64-serialized geopackages can be successfully sent wrapped in an OWS context entries. Finally, the GeoPackage standard can be extended to support synchronization by introducing tables to log changes and to uniquely identify revisions.

### **10.1 Recommendations**

This engineering report makes the following recommendations:

#### **10.1.1 Standardize how binary content is carried in OWS Context documents**

Testbed-11 demonstrated how a geopackage serialized in base64 text could be sent wrapped in an OWS Context entry. The approach was also observed to be applicable to other binary content such as imagery (e.g. GeoTIFF, JPEG2000 etc.) by specifying the MIME type of the binary content. As the current version of the OWS Context standard does not specify an approach for carrying binary content within context documents, this engineering report recommends that future versions of OWS Context standardize the use of base64 for wrapping binary content inside such documents.

#### **10.1.2 Extend GeoPackage to Standardize Tables for Supporting Synchronization**

Testbed-11 demonstrated how synchronization between geopackages can be achieved. The TB also demonstrated how synchronization between a geopackage and an enterprise server could also be achieved. The successful implementation and demonstration of this capability shows that extension of the GeoPackage standard using the tables described in

Section 8 can offer a standardized approach for synchronization involving geopackages. Therefore the development of a synchronization extension of the GeoPackage standard is recommended.

### **10.1.3 Develop a Best Practice for a GeoPackaging Profile of WPS**

Recent testbeds, including Testbed-11, have generated a wealth of evidence supporting the use of WPS for creating geopackages. As a result of this experience and the increasing popularity of the GeoPackage standard, the OGC should develop a Best Practice for a GeoPackaging Profile of WPS.

Annex A

**10.2 Revision history**

<b>Date</b>	<b>Release</b>	<b>Editor</b>	<b>Primary clauses modified</b>	<b>Description</b>
8/5/15		Carl Reed	Various	Preparation for publication