

OGC Vector Tiles Pilot  
*GeoPackage 1.2 Vector Tiles Extensions Engineering  
Report*

# Table of Contents

1. Summary	4
1.1. Requirements & Research Motivation	4
1.2. Prior-After Comparison	5
1.3. Recommendations for Future Work	5
1.4. Document contributor contact points	5
1.5. Foreword	6
2. References	7
3. Terms and definitions	8
3.1. Abbreviated terms	8
4. Overview	9
5. Pilot Results	10
5.1. Pilot Architecture	10
5.2. Providers	10
5.2.1. Compusult	11
5.2.2. CubeWerx	13
5.2.3. Ecere	13
5.3. Clients	14
5.3.1. Compusult	14
5.3.2. Image Matters Client #1 - GeoPackage.js	18
5.3.3. Image Matters Client #2 - Mobile Client	20
5.3.4. Ecere	23
5.4. Technology Integration Experiments	25
5.5. Denied, Degraded, Intermittent, or Limited (DDIL) Environments	25
6. Vector Tiles Extensions (Informative)	27
6.1. Vector Tiles Extension	27
6.1.1. Extension Title	27
6.1.2. Introduction	27
6.1.3. Extension Author	27
6.1.4. Extension Name or Template	27
6.1.5. Extension Type	27
6.1.6. Applicability	27
6.1.7. Scope	28
6.1.8. Specification	28
6.2. Mapbox Vector Tiles Extension	29
6.2.1. Extension Title	29
6.2.2. Introduction	29
6.2.3. Extension Author	29
6.2.4. Extension Name or Template	29

6.2.5. Extension Type	29
6.2.6. Applicability	29
6.2.7. Scope	29
6.2.8. Specification	30
6.3. GeoJSON Vector Tiles Extension	30
6.3.1. Extension Title	30
6.3.2. Introduction	30
6.3.3. Extension Author	30
6.3.4. Extension Name or Template	30
6.3.5. Extension Type	30
6.3.6. Applicability	30
6.3.7. Scope	31
6.3.8. Specification	31
6.4. OWS Context Extension	31
6.4.1. Extension Title	31
6.4.2. Introduction	31
6.4.3. Extension Author	31
6.4.4. Extension Name or Template	31
6.4.5. Extension Type	31
6.4.6. Applicability	32
6.4.7. Scope	32
6.4.8. Specification	32
6.5. Vector Tiles Attributes Extension	39
6.5.1. gpkg_extensions	39
6.5.2. gpkgext_relations	39
6.5.3. User-defined Mapping Table	40
7. Requirements (Normative)	41
7.1. Vector Tiles Extension	41
7.1.1. gpkg_contents	41
7.1.2. gpkg_extensions	41
7.1.3. gpkgext_vt_layers	42
7.1.4. gpkgext_vt_fields	43
7.2. Mapbox Vector Tiles Extension	44
7.2.1. gpkg_extensions	44
7.2.2. User Defined Tiles Tables	44
7.3. GeoJSON Vector Tiles Extension	44
7.3.1. gpkg_extensions	45
7.3.2. User Defined Tiles Tables	45
7.4. OWS Context Extension	45
7.5. Vector Tiles Attributes Extension	45
7.5.1. gpkgext_relations	46

8. Discussion .....	47
8.1. Mapbox vs. GeoJSON Vector Tiles.....	47
8.2. Tile Compression .....	47
8.3. Attributes .....	47
8.3.1. Implications .....	47
8.3.2. Design Decisions .....	48
8.4. Coordinate Reference Systems of Tiles .....	48
8.5. Styling and Symbology .....	48
8.5.1. Decoupling Data and Styles .....	49
8.5.2. Style Encoding .....	49
8.5.3. Cardinality of Styles .....	49
8.6. Well-Known Scale Sets .....	49
8.7. Layer Extents.....	49
Appendix A: Revision History .....	50
Appendix B: Bibliography .....	51

Publication Date: 2019-02-15

Approval Date: 2019-01-22

Submission Date: 2018-11-08

Reference number of this document: OGC 18-074

Reference URL for this document: <http://www.opengis.net/doc/PER/vtp-VTPD005>

Category: Public Engineering Report

Editor: Jeff Yutzler

Title: OGC Vector Tiles Pilot: GeoPackage 1.2 Vector Tiles Extensions Engineering Report

---

## **OGC Engineering Report**

### **COPYRIGHT**

Copyright © 2019 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

### **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. However, the discussions in this document could very well lead to the definition of an OGC Standard.

## 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 NONINFRINGEMENT 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.

# Chapter 1. Summary

Tiled feature data, colloquially referred to as 'vector tiles', can be used to optimize the delivery of vector data over the web. This data may subsequently be used to support visualization (particularly through maps) as well as limited analysis activities. One goal of the OGC Vector Tiles Pilot was to define candidate extensions to existing OGC standards as a way to advance the use of vector tiles technology as part of the OGC baseline. This Engineering Report (ER) describes a set of possible extensions to GeoPackage 1.2 that documents the mechanism to store and retrieve vector tiles in a GeoPackage. These extensions work together to enable a GeoPackage to act as a container format that can support visualization and analysis activities, even in a Denied, Degraded, Intermittent, or Limited Bandwidth (DDIL) environment.

The GeoPackage Vector Tiles extensions define the rules and requirements for encoding vector tiles in a GeoPackage data store. There are five draft extensions:

1. The Vector Tiles Extension provides vector tiles support through the GeoPackage `tiles` option.
2. The Mapbox Vector Tiles Extension allows the content of a tile Binary Large Object (BLOB) to be a Mapbox Vector Tile as per version 2.1 of the Mapbox Vector Tile (MVT) specification [1].
3. The GeoJSON Vector Tiles Extension allows the content of each tile BLOB to be a [GeoJSON](https://tools.ietf.org/html/rfc7946) [https://tools.ietf.org/html/rfc7946] file.
4. The OGC Web Services (OWS) Context Extension provides a way to store information describing a list of geospatial resources, including but not limited to maps, their layers, and the styles of those layers.
5. The Vector Tiles Attributes Extension allows attribute information for each feature to be stored in relational tables for more convenient querying.

To support vector tiles, a minimum of at least two extensions is required. The first extension enables vector tiles support. However, to be usable, an encoding format must be declared via either the second or third extension. The other extensions are purely optional.

These extensions, like all GeoPackage extensions, are intended to be transparent and to not interfere with GeoPackage-compliant, but non-supporting, software packages.

## 1.1. Requirements & Research Motivation

The research presented in this ER has been motivated by the increasing adoption of vector tiling within the geospatial industry.

The ER addresses deliverable D005 of the Vector Tiles Pilot. The Vector Tiles Pilot Call for Participation (CFP) outlines the deliverable as follows:

D005: GeoPackage 1.2 Vector Tiles Extension Engineering Report – An extension to GeoPackage 1.2 written as a draft OGC standard that describes the mechanism to store and retrieve vector tiles in GeoPackage.



## 1.2. Prior-After Comparison

Before the work performed in this Pilot, there was no interoperable mechanism for sharing vector tiles information in a portable container suitable for use in a DDIL environment. There were three specifications that were inputs to the work in this ER.

1. [MBTiles](https://www.mapbox.com/help/define-mbtiles/) [https://www.mapbox.com/help/define-mbtiles/] is a format for storing tilesets, including but not limited to vector tiles. MBTiles is an open specification, but is fully controlled by Mapbox Inc. MBTiles was an influence on defining the OGC's [GeoPackage Encoding Standard](https://www.geopackage.org/spec120/) [https://www.geopackage.org/spec120/].
2. Image Matters produced a [community extension for GeoPackage](https://github.com/jyutzler/geopackage-vector-tiles/blob/master/spec/1_vector_tiles.adoc) [https://github.com/jyutzler/geopackage-vector-tiles/blob/master/spec/1\_vector\_tiles.adoc], but this extension had not been proven through working code.
3. The GeoPackage Standards Working Group (SWG), in collaboration with participants from the GeoPackage Related Tables Interoperability Experiment, has developed the [GeoPackage Related Tables Extension](https://www.geopackage.org/18-000.html) [https://www.geopackage.org/18-000.html]. This extension allows for a many-to-many relationship between base data and related data, though the working draft does not address vector tiles.

## 1.3. Recommendations for Future Work

1. When this project was initiated, the term "vector tiles" was used throughout. However, as the project progressed, the participants agreed that the term "tiled feature data" was more appropriate, particularly as a data type for GeoPackage. This is consistent with the term "tiled gridded coverage data" which is in OGC 17-066, a recently adopted OGC GeoPackage Extension. In the future, it would be appropriate to migrate references to "vector tiles" should be migrated to "tiled feature data". The exception is Mapbox Vector Tiles which is a specific technology solution.
2. There are a number of possibilities for future work pertaining to attributes. This Pilot did not take a close look at the potential for using the available attribute information to support analysis operations. It is not clear at this time what is possible given the way that geometry and feature information is spread across multiple tiles. There is currently no agreed upon way to associate the different layers embedded within a single MVT tile set with their own individual attributes table.
3. GeoPackage does not currently have an interoperable mechanism for styling. This was looked at during the Pilot, but a consensus was not reached. Future work to address this gap would allow styling rules to be shared and maps to display consistently in different GeoPackage clients.

## 1.4. Document contributor contact points

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

### Contacts

<b>Name</b>	<b>Organization</b>
Jeff Yutzler	Image Matters LLC
Gobe Hobona	OGC
Carl Reed	Carl Reed and Associates
Alia Robinson	Image Matters LLC
Adam Parsons	Compusult Ltd.

## **1.5. Foreword**

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.

# Chapter 2. References

The following normative documents are referenced in this document.

*NOTE: Only normative standards are referenced here, e.g. OGC, ISO or other SDO standards. All other references are listed in the bibliography. Example:*

- **OGC: OGC 12-128r14, OGC GeoPackage Encoding Standard, Version 1.2.0, 2018** [<https://www.geopackage.org/spec120/>]
- **OGC: OGC 06-121r9, OGC® Web Services Common Standard, 2010** [[https://portal.opengeospatial.org/files/?artifact\\_id=38867&version=2](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2)]
- **OGC: OGC 12-080r2, OGC OWS Context Conceptual Model, 2013** [[https://portal.opengeospatial.org/files/?artifact\\_id=55182](https://portal.opengeospatial.org/files/?artifact_id=55182)]
- **IETF: RFC-7946 The GeoJSON Format, 2016** [<https://tools.ietf.org/html/rfc7946>]
- **IETF: RFC-1951 DEFLATE Compressed Data Format Specification version 1.3, 1996** [<https://tools.ietf.org/html/rfc1951>]

# Chapter 3. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2) [https://portal.opengeospatial.org/files/?artifact\_id=38867&version=2] shall apply. In addition, the following terms and definitions apply.

- Extended GeoPackage

a GeoPackage that contains any additional data elements (tables or columns) or SQL constructs (data types, indexes, constraints or triggers) that are not specified in this encoding standard.

- GeoPackage file

a platform-independent SQLite database file that contains GeoPackage data and metadata tables with specified definitions, integrity assertions, format limitations and content constraints.

- tile

a rectangular pictorial representation of geographic data, often part of a set of such elements, covering a spatially contiguous extent and sharing similar information content and graphical styling, which can be uniquely defined by a pair of indices for the column and row along with an identifier for the tile matrix.

## 3.1. Abbreviated terms

- BLOB Binary Large Object
- DDIL Denied, Degraded, Intermittent, or Limited
- GPKG GeoPackage
- KML Keyhole Markup Language
- MVT Mapbox Vector Tiles
- OWS OGC Web Services
- RTE Related Tables Extension
- SRS Spatial Reference System
- SWG Standards Working Group

# Chapter 4. Overview

This Engineering Report contains the following sections that describe the way in which Vector Tiles support in GeoPackage can be implemented through a set of extensions:

- Section 5 presents a summary of the GeoPackage-related accomplishments during this Pilot, including any relevant Technology Integration Experiments (TIEs).
- Section 6 presents an informative specification for the proposed Vector Tiles Extensions.
  - Section 6a presents the Vector Tiles Extension. This extension provides vector tiles support through the GeoPackage `tiles` option. Instead of PNG or JPG files, each tile BLOB is a Vector Tile.
  - Section 6b presents the Mapbox Vector Tiles Extension. This extension allows the content of a tile BLOB to be a Mapbox Vector Tile as per the [Mapbox Vector Tiles \(MVT\) specification](https://www.mapbox.com/vector-tiles/specification/) [https://www.mapbox.com/vector-tiles/specification/] [version 2.1](https://github.com/mapbox/vector-tile-spec/tree/master/2.1) [https://github.com/mapbox/vector-tile-spec/tree/master/2.1].
  - Section 6c presents the GeoJSON Vector Tiles Extension. This extension allows the content of a tile BLOB to be a GeoJSON file.
  - Section 6d presents the OWS Context Extension. This extension provides a way to store information describing a list of geospatial resources, including but not limited to maps, their layers, and the styles of those layers.
  - Section 6e presents the Vector Tiles Attributes Extension. This extension allows attribute information for each feature to be stored in relational tables for more convenient querying. It uses the GeoPackage `attributes` option in conjunction with the emerging GeoPackage Related Tables Extension.
- Section 7 presents the extensions in Section 6 as normative requirements that could be the basis for an OGC standard.
- Section 8 describes discussion topics, critical topics that came up during the Pilot, that the GeoPackage Standards Working Group would want to consider before adopting these extensions as a standard.
- Annex A presents the revision history of this document.
- Annex B contains a Bibliography.

# Chapter 5. Pilot Results

## 5.1. Pilot Architecture

The Pilot is designed to be delivered within a short timescale in a phased approach, Phase 1 consists of delivery of draft components and ERs and Phase 2 requires delivery of final components and ERs. The overall goal of the Pilot is to integrate Vector Tiles into existing OGC standards to enable their usage through OGC compliant architectures. This is done through profiling and providing extensions to existing standards. This section contains a short description of each of the delivered components, ERs and an overall architecture.

The architecture of the pilot is designed to cover the three main server client relationships that are common in OGC use cases as illustrated in [Figure 1](#):

- Desktop Client → Web Feature Service (WFS) 3.0
- Web Client → Web Map Tile Service (WMTS) 1.0
- Mobile Client → GeoPackage 1.2

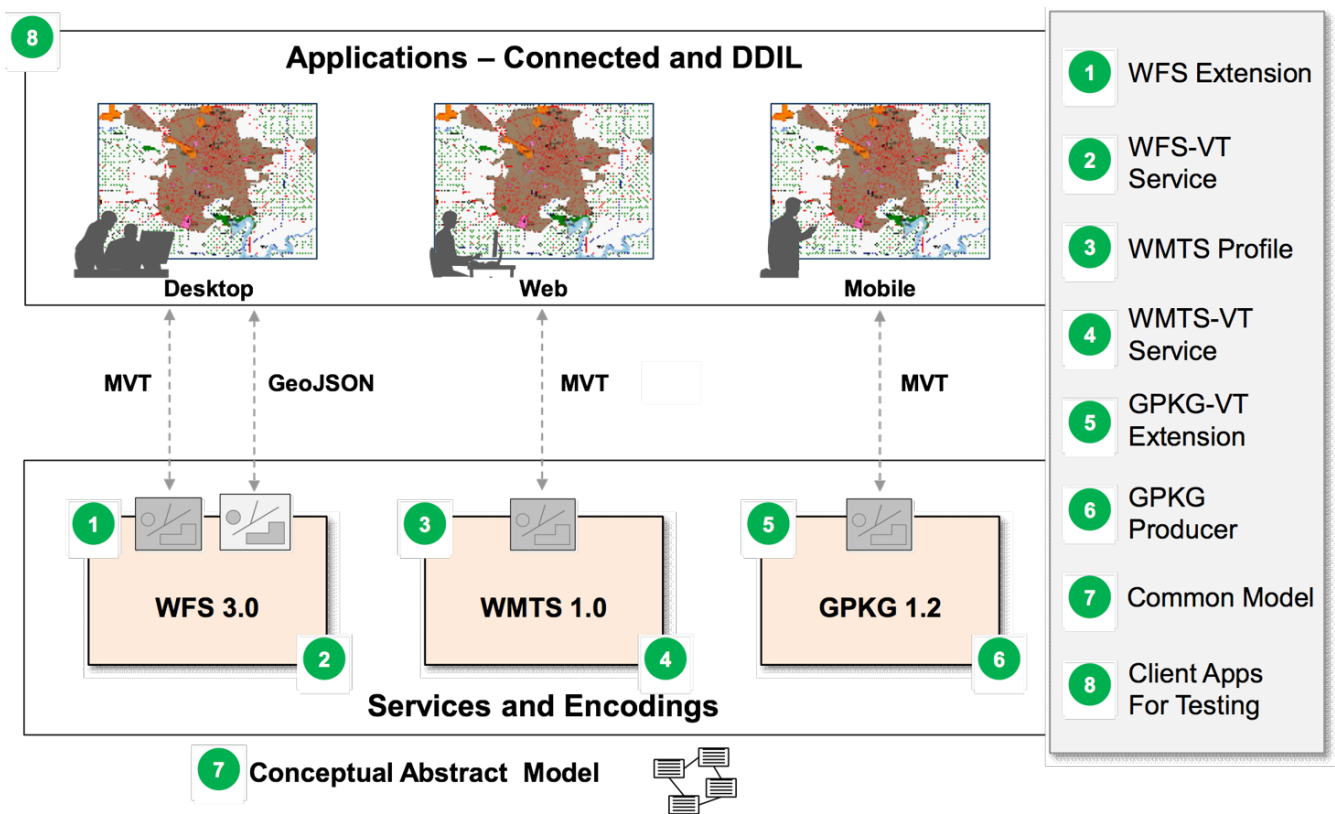


Figure 1. Vector Tiles Pilot Architecture

This architecture attempts to address vector tiles in each of the client server relationships to simultaneously enable vector tiles across the relevant suite of OGC standards. This approach provides implementers with guidance for vector tiles no matter their OGC use case.

## 5.2. Providers

During this Pilot, three GeoPackage Vector Tiles providers were produced.

### 5.2.1. Compusult

The Compusult GeoPackage Producer runs as a web-browser based application, as well as being accessible via an OGC Web Processing Service (WPS) instance. The GeoPackage Producer supports producing Mapbox Vector Tile and GeoJSON tile based GeoPackages in a user selected projection system and an uploaded vector source(Shapefile(s),GeoDatabase,SqliteDB etc.). The producer has the ability to convert all feature types into a single Mapbox Vector Tile or produce a single Vector Tile layer for each feature type. Furthermore, the producer also has the ability to embed feature attributes into the Mapbox Vector Tile or to use a [GeoPackage Related Tables Extension](https://www.geopackage.org/18-000.html) [https://www.geopackage.org/18-000.html] to produce attributes tables and appropriate mappings for optimal storage. Processed features are converted to [Mapbox Vector Tiles \(MVT\) specification](https://www.mapbox.com/vector-tiles/specification/) [https://www.mapbox.com/vector-tiles/specification/] [version 2.1](https://github.com/mapbox/vector-tile-spec/tree/master/2.1) [https://github.com/mapbox/vector-tile-spec/tree/master/2.1] by using the open-source Java library [Mapbox Vector Tile Java](https://github.com/wdtinc/mapbox-vector-tile-java) [https://github.com/wdtinc/mapbox-vector-tile-java]. Feature types of Polygon and Point geometries have their bounds clipped using a buffer to ensure clients can render freely without having to worry about artificial line segments from tile bounds. Automatic layer order for drawing purposes is detected by examining feature type size, bounds and type. The result of the Vector Tile Extension is a GeoPackage that is compliant to specifications of the National System for Geospatial Intelligence (NSG).

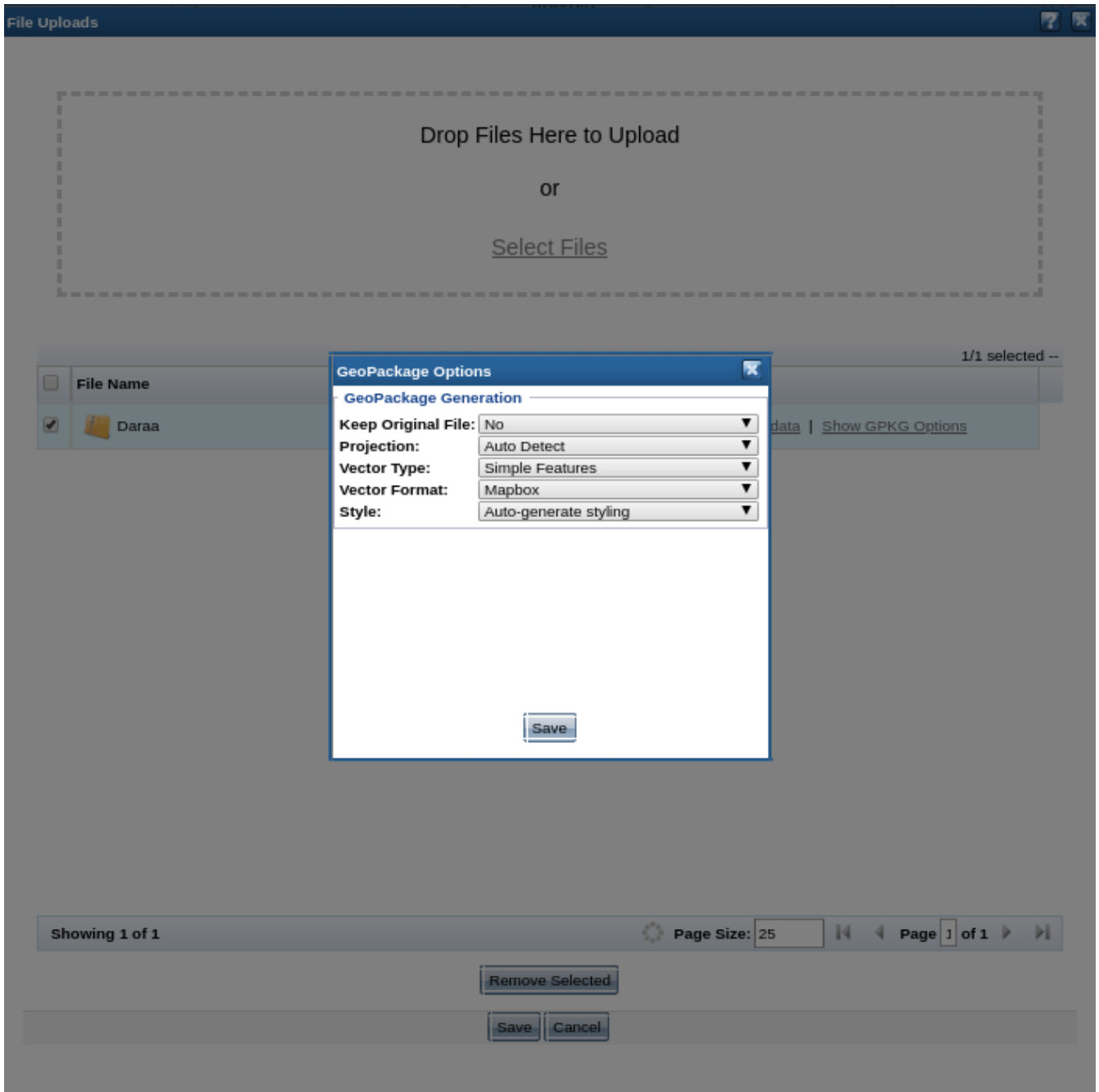


Figure 2. Compusult GeoPackage Producer Settings



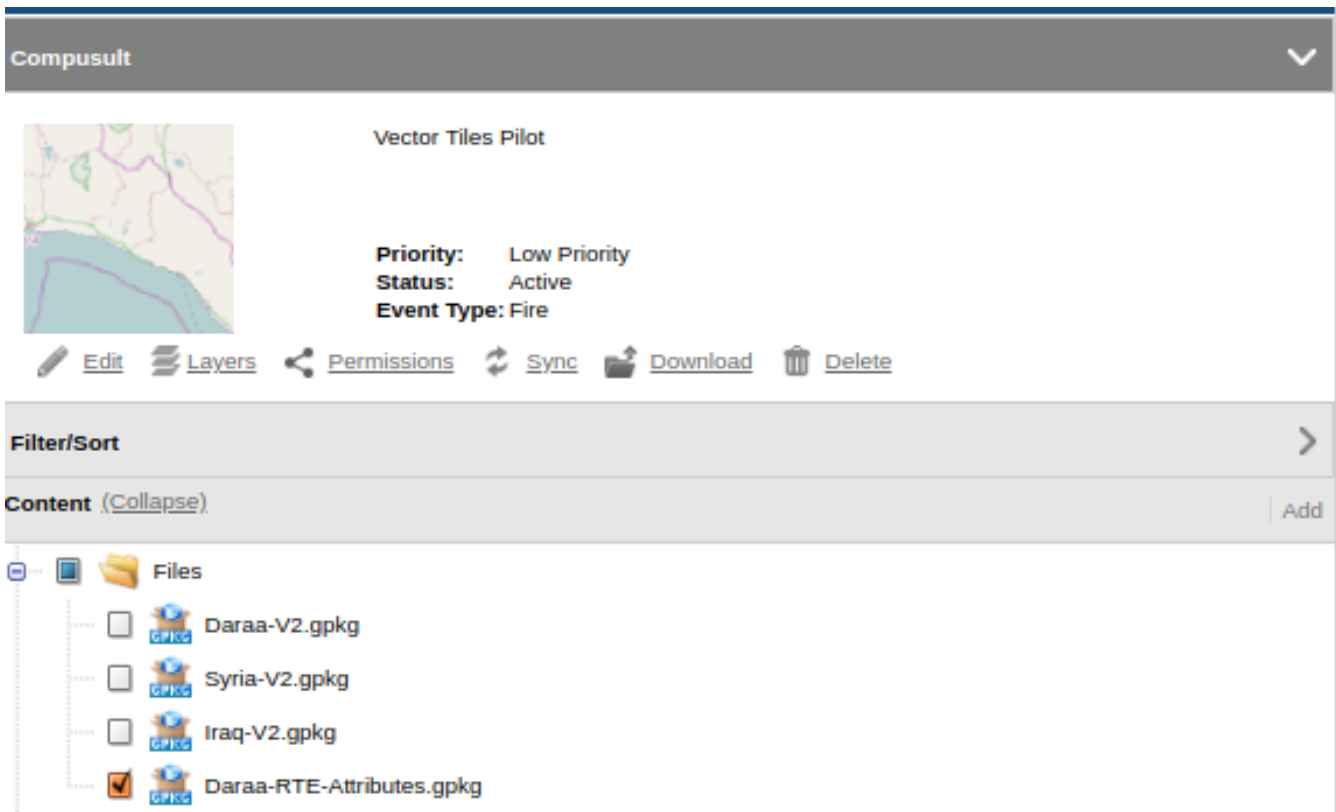


Figure 3. Compusult GeoPackage Producer Result

### 5.2.2. CubeWerx

The CubeWerx GeoPackage producer is integrated into the tile-management system for the CubeWerx implementation of the OGC WMTS standard and can generate tiles in different projections and formats from source data of many different formats registered with the CubeSTOR database. The Vector Tiles Pilot added the ability for the underlying tiling system to manage MapBox and GeoJSON vector tiles for the WMTS interface and to write them out into GeoPackages according to the draft Vector Tile Extension Specification. Deflate compression for the vector-tile blobs is also supported and can make the individual tiles a fraction of their uncompressed size.

As recommended by the MVT specification, the CubeWerx-provided MVT tiles clip lines and polygons to the edge of a buffer zone around each tile. By default, we use a buffer zone of 2% of the linear vector-tile size (so a 1024x1024 vector tile would have a buffer zone that's 20 vector-tile pixels thick), but that percentage is configurable.

### 5.2.3. Ecere

Functionality to produce GeoPackages was added to Ecere's GNOSIS Cartographer GIS desktop application. Multiple data layers can be selected to export into a GeoPackage. The option of tiling vector data using the new vector tiles extensions is provided. A number of options are available, such as whether to embed attributes information within the individual tiles themselves, or store them within an attributes table in the GeoPackage, which is normally done for storing non-tiled vector data. The advantage of the attributes table approach is to avoid repeating the same information for features which occur in many different tiles. This approach also provides the ability to perform SQL queries, potentially together with the R-tree extension for spatial indexing. A choice of tiling schemes is available, including the following:

- tiling schemes based on WMTS Well Known Scale Sets, such as Google Maps Compatible
- the GNOSIS Global Grid which has the benefit of featuring fewer tiles closer to the pole than at the equator [1: note that the draft OGC Tile matrix set description cannot accurately describe this tiling scheme, and it is hoped that a future OGC-wide tiling scheme description standard will accommodate this flexibility]

A selection of supported encodings is provided, including MVT, GeoJSON, GeoECON, GML, and GNOSIS Map Tiles. The user can select whether multiple layers should be encoded as a single MVT set (with multiple layers in each encoded tile), or as a separate tile set per layer. The ability to define and include styles within the GeoPackage, which can be based in part on the attributes associated with the data, was also implemented. Multiple style encodings can be included (GeoPackages produced have also contained Mapbox GL styles, GNOSIS Cascading Map Style Sheets, SLD/SE and GeoCSS).

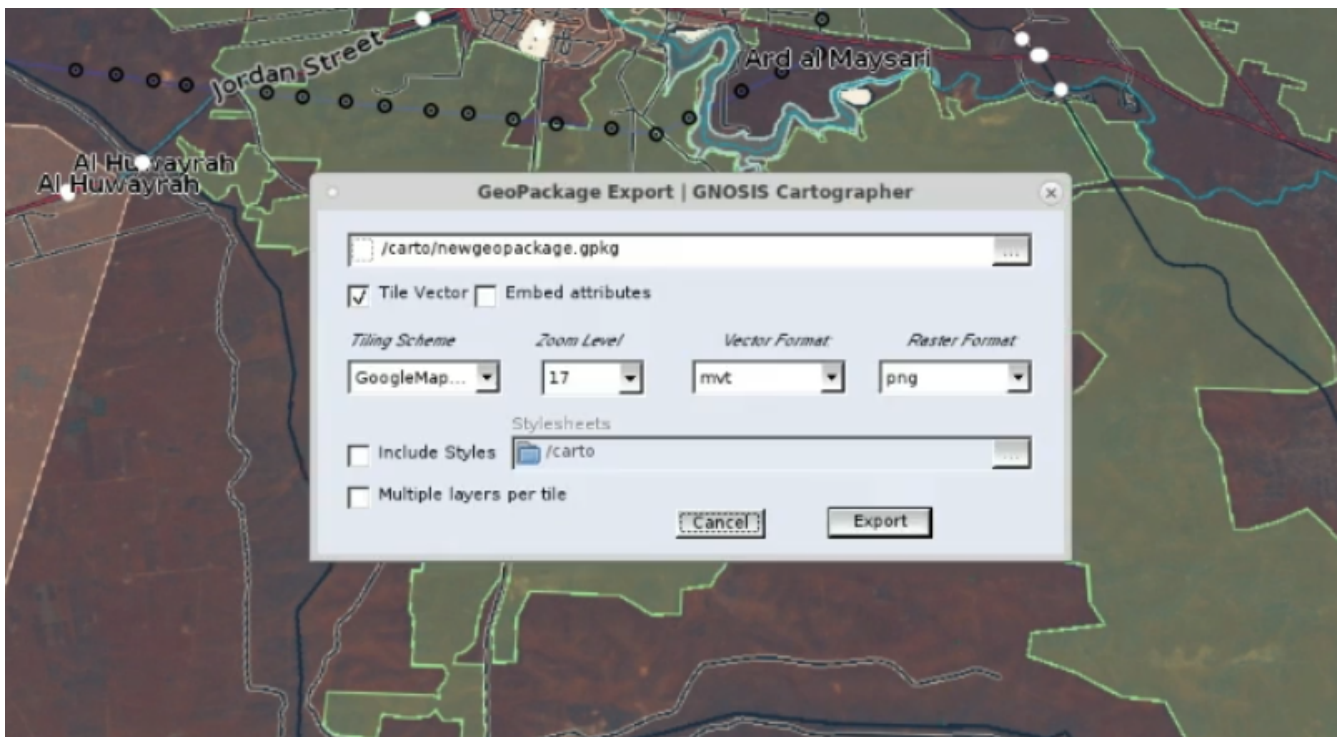


Figure 4. Ecere GeoPackage Producer

## 5.3. Clients

During this Pilot, four client applications were produced.

### 5.3.1. Compusult

The Compusult GeoPackage client runs as a web-browser based application, as well as a desktop/android based application. The GeoPackage client supports both Mapbox Vector Tile and GeoJSON tile formats. The client application allows users to select and enable/disable specific WMTS layers as well as overlay any specified base map for visualization. Styles are randomly generated for each layer. The screenshots below show the GeoPackage client rendering different GeoPackage providers. The client also allows for native GetFeatureInfo requests to be sent to a supporting server. GetFeatureInfo is one of the operations offered by Web Map Service (WMS) and WMTS compliant services. Web-browser based applications used a GeoPackage to produce internal

WMS/WMTS services that were then rendered by the browser.

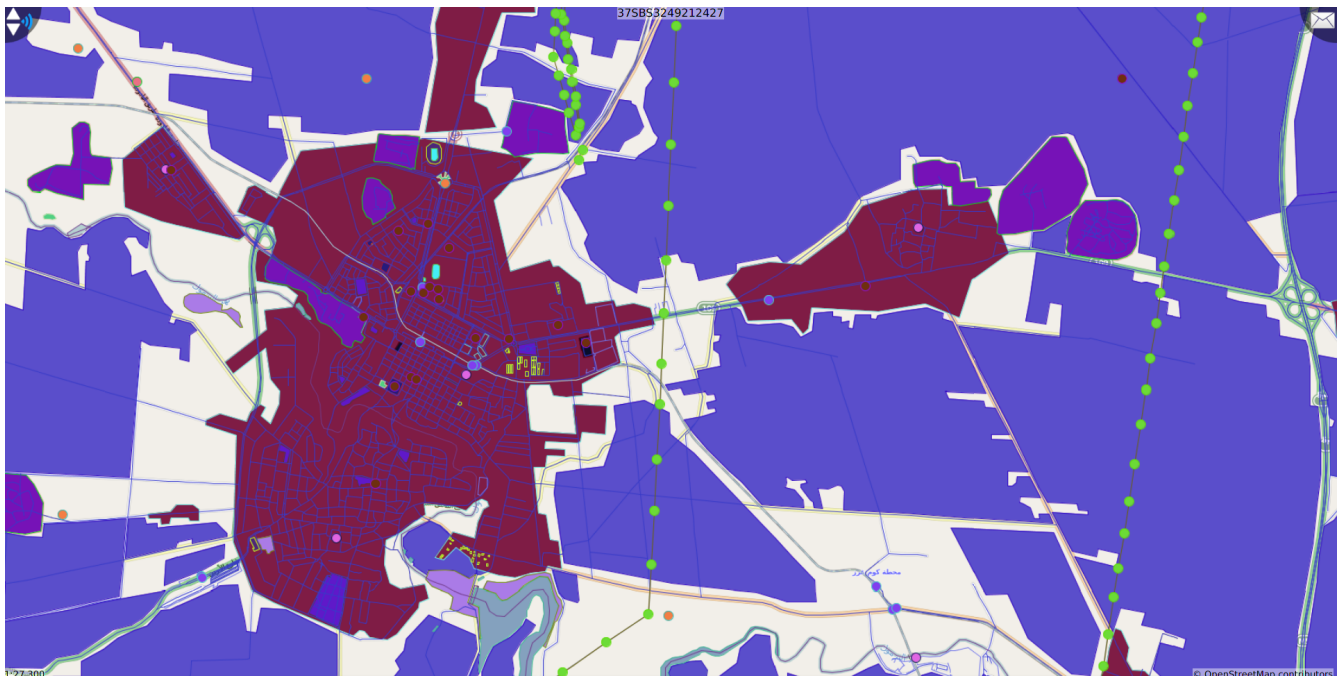


Figure 5. CubeWerx Vector Tiles Server Rendered on the CompuSult Client

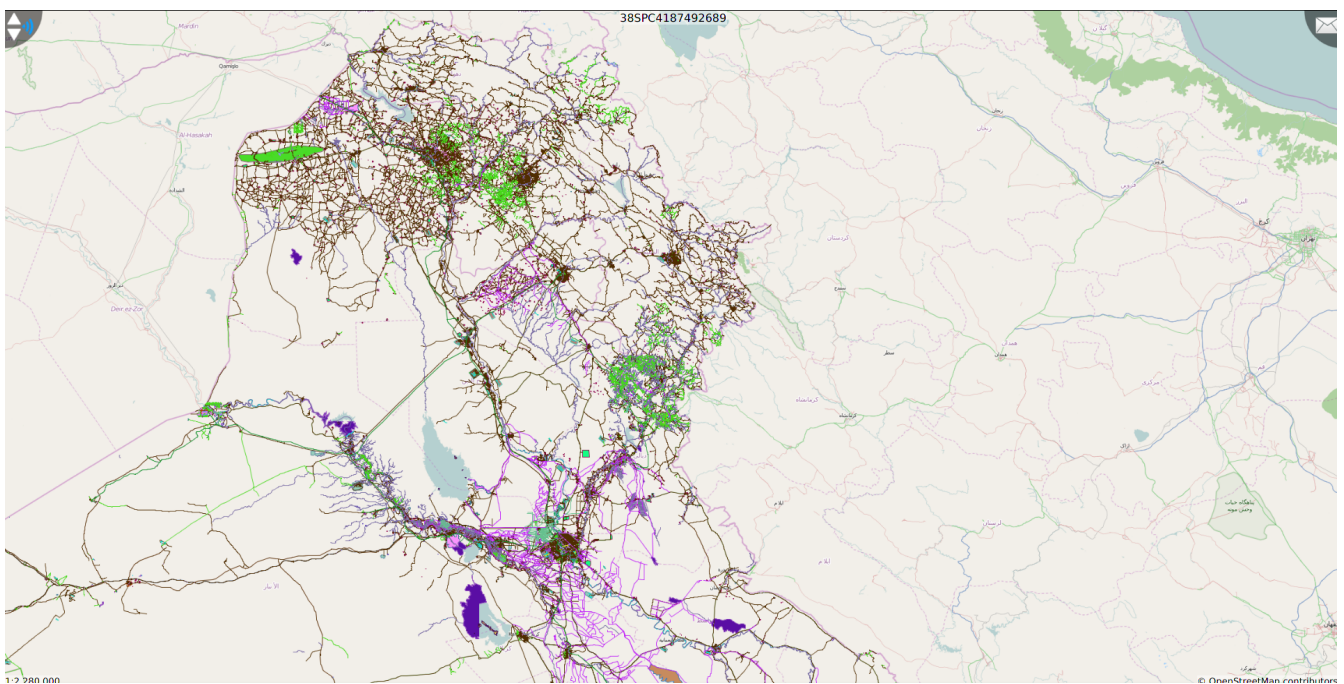


Figure 6. GeoSolutions Vector Tiles Rendered on the CompuSult Client

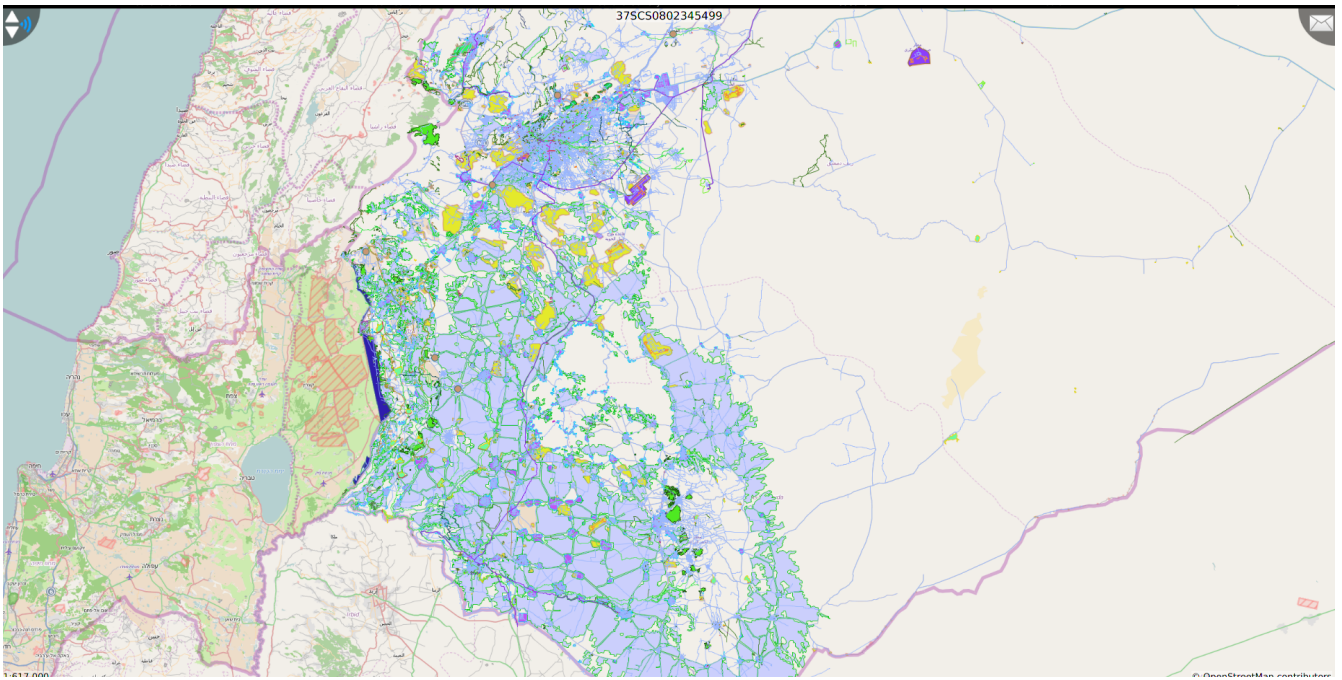


Figure 7. Ecere Vector Tiles Rendered on the Compusult Client

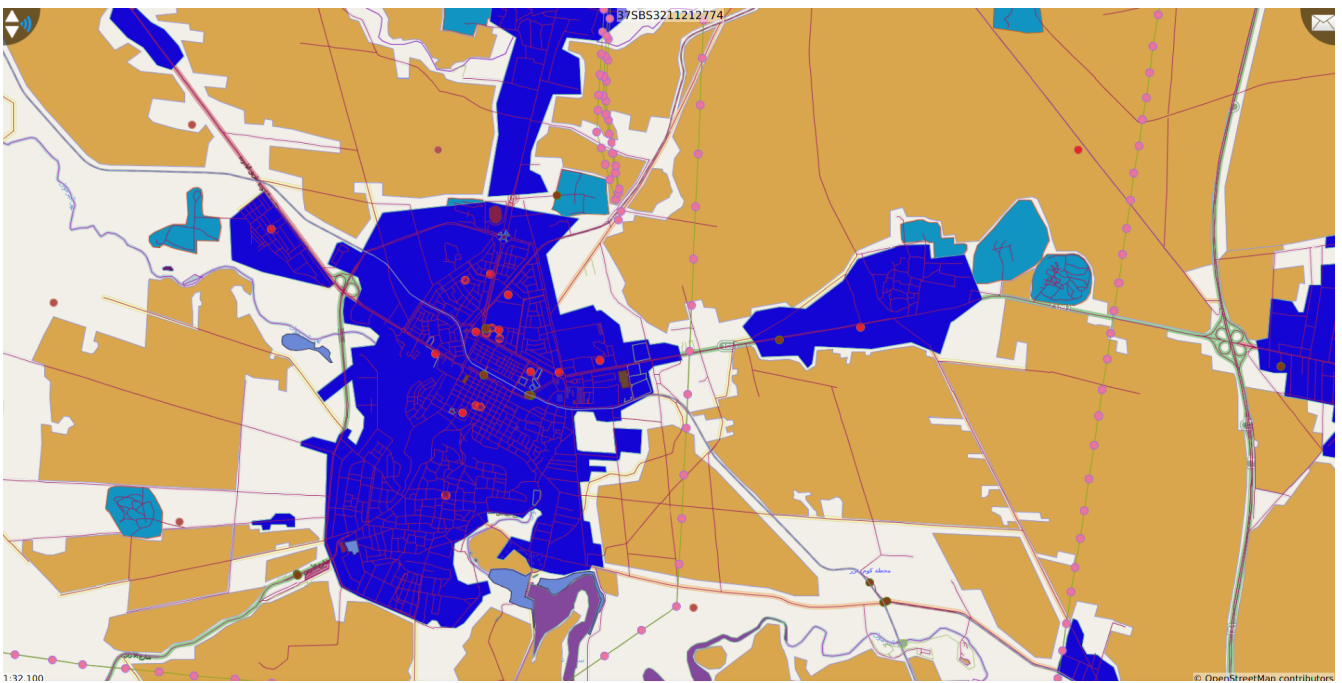


Figure 8. Feature Info Response Compusult Client

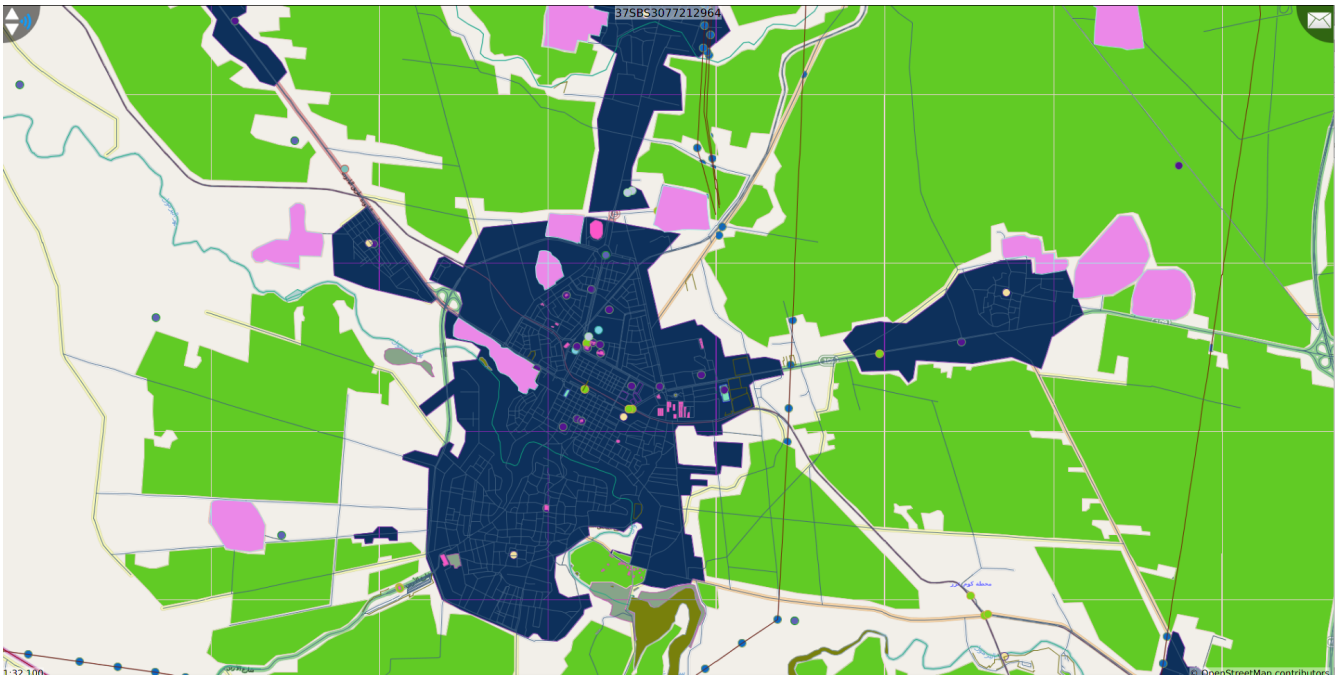


Figure 9. CubeWerx Vector Tiles Server Rendered on the Compusult Client

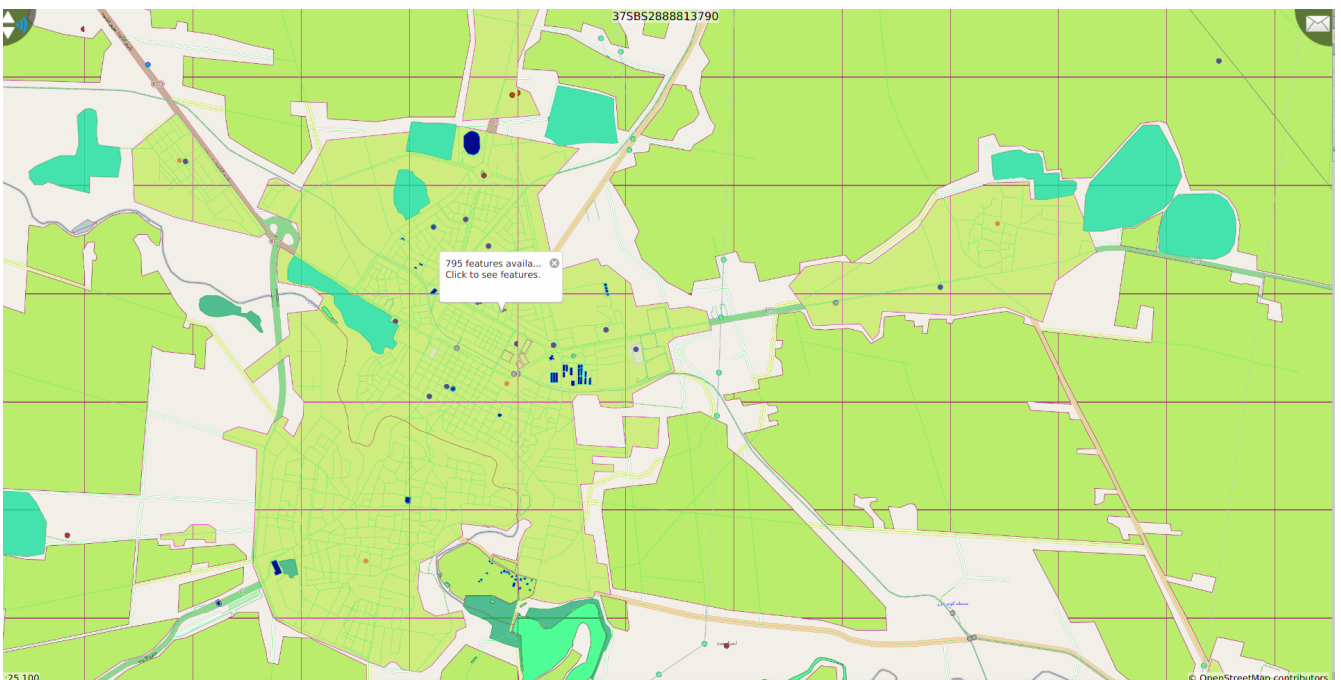


Figure 10. GeoSolutions Vector Tiles Rendered on the Compusult Client

### Minor Issues Noted

- GeoPackage web-based services displayed with all feature types as individual layers resulted in slow rendering times on a web-based client due to the number of requests and browser request limitations.
- Services with MVT tiles that have polygons that are not buffered will result in displaying intermittent tile boundaries when trying to render polygon strokes. The same issue occurs when rendering points with images that cross tile boundaries.
- GeoPackages with multiple layers did not specify the correct drawing order to properly layer all feature types with no stylesheet provided.

## Recommendations

- Specification of a media-type (formerly known as a MIME-type) is required in the data-model to ensure clients do not have to inspect tile data to determine the encoding type of the content provided.
- Clients should not have to depend on sibling tiles to perform proper rendering operations.

### 5.3.2. Image Matters Client #1 - GeoPackage.js

GeoPackage.js is an in-browser GeoPackage client published by the National Geospatial-intelligence Agency (NGA). This client is capable of displaying tile tables with image data and feature tables with vector data from a GeoPackage. Image Matters has extended this client to add support for vector tiles encoded using the Mapbox Vector Tile Specification. This extension uses the [Leaflet](https://leafletjs.com/) [https://leafletjs.com/] [VectorGrid](#) library to render these tiles. This implementation extends [VectorGrid.ProtoBuf](#) from this library to retrieve Protobufs from a GeoPackage rather than from the web. When a user imports a GeoPackage, the user tables are listed in the right pane and may be individually toggled on and off in the displayed map on the left. Expanding the “details” of a table allows a user to view the table metadata, tile bounds, feature attributes, and layers within a vector tile table.

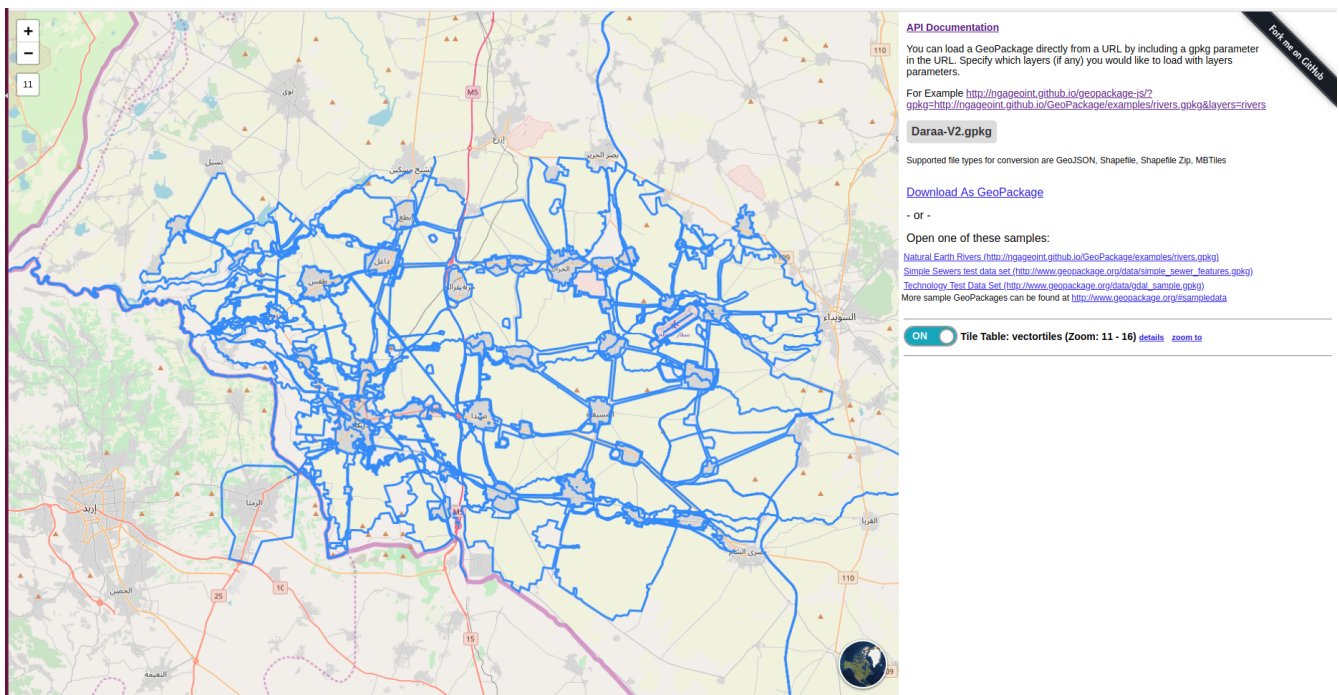


Figure 11. Vector Tiles Rendered on the Image Matters Web Client

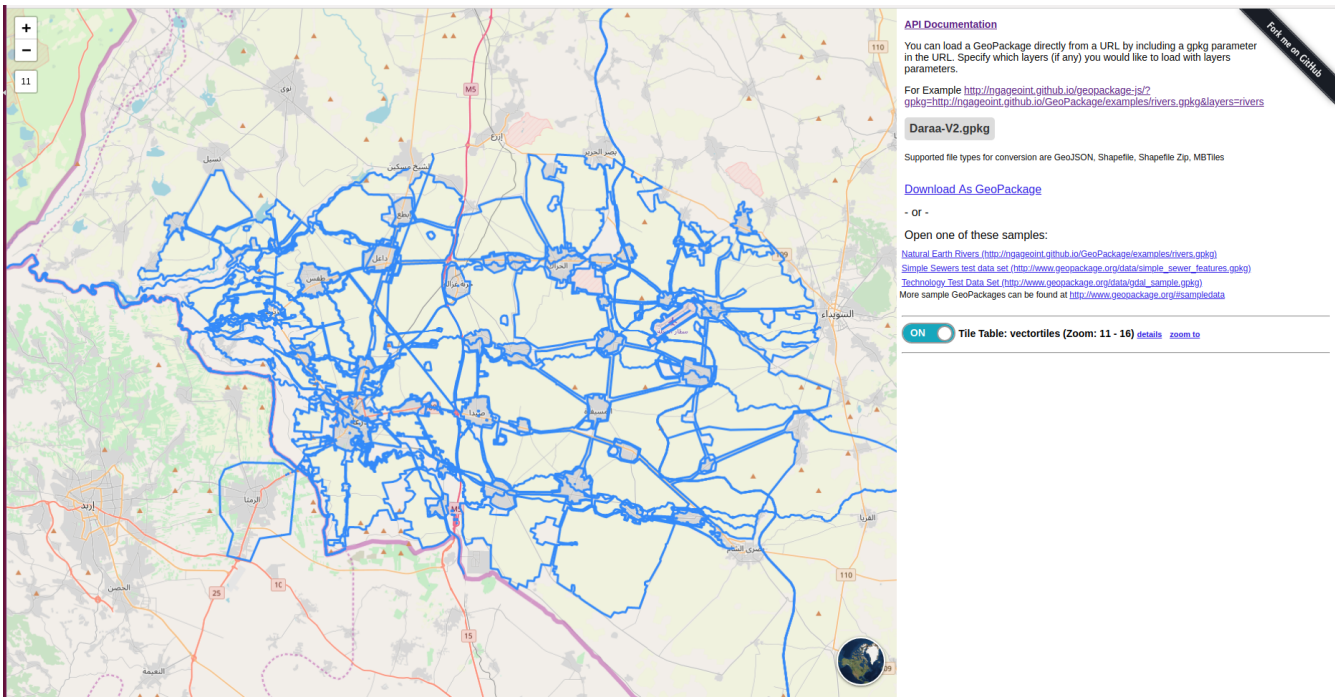


Figure 12. Vector Tiles Rendered on the Image Matters Web Client

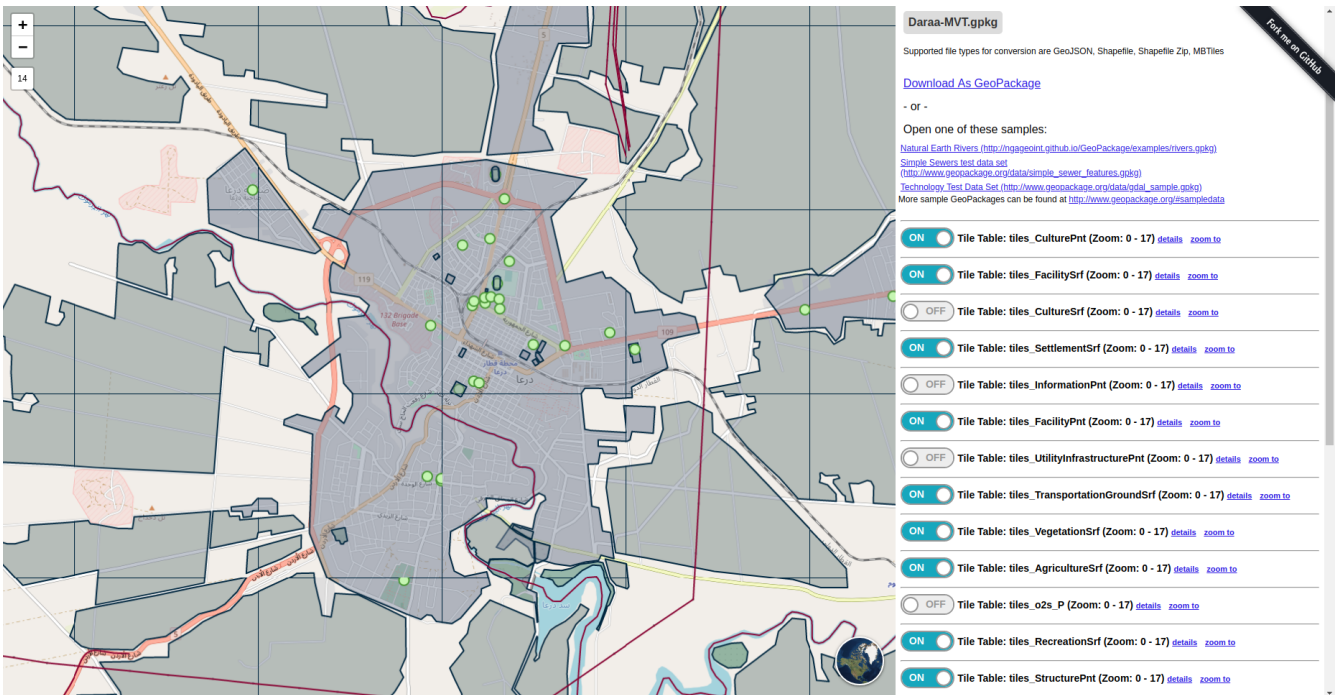


Figure 13. Ecere Vector Tiles Rendered on the Image Matters Web Client

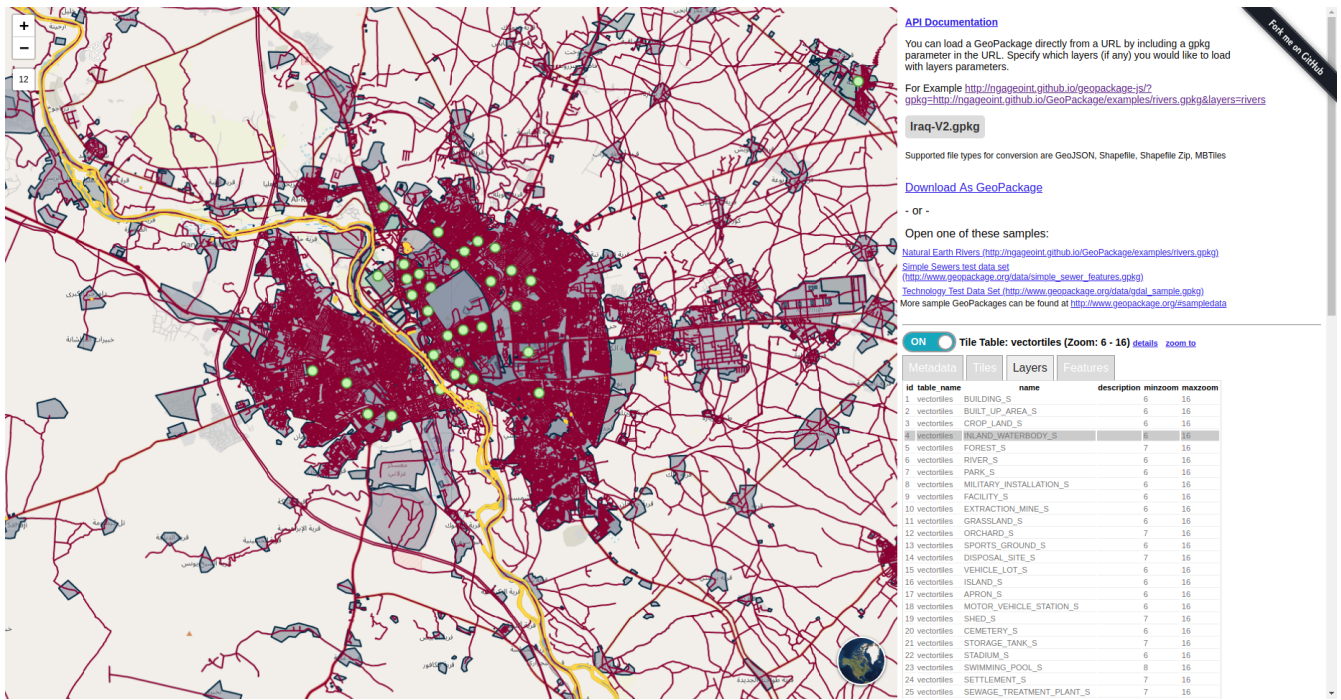


Figure 14. CompuSult Vector Tiles Rendered on the Image Matters Web Client

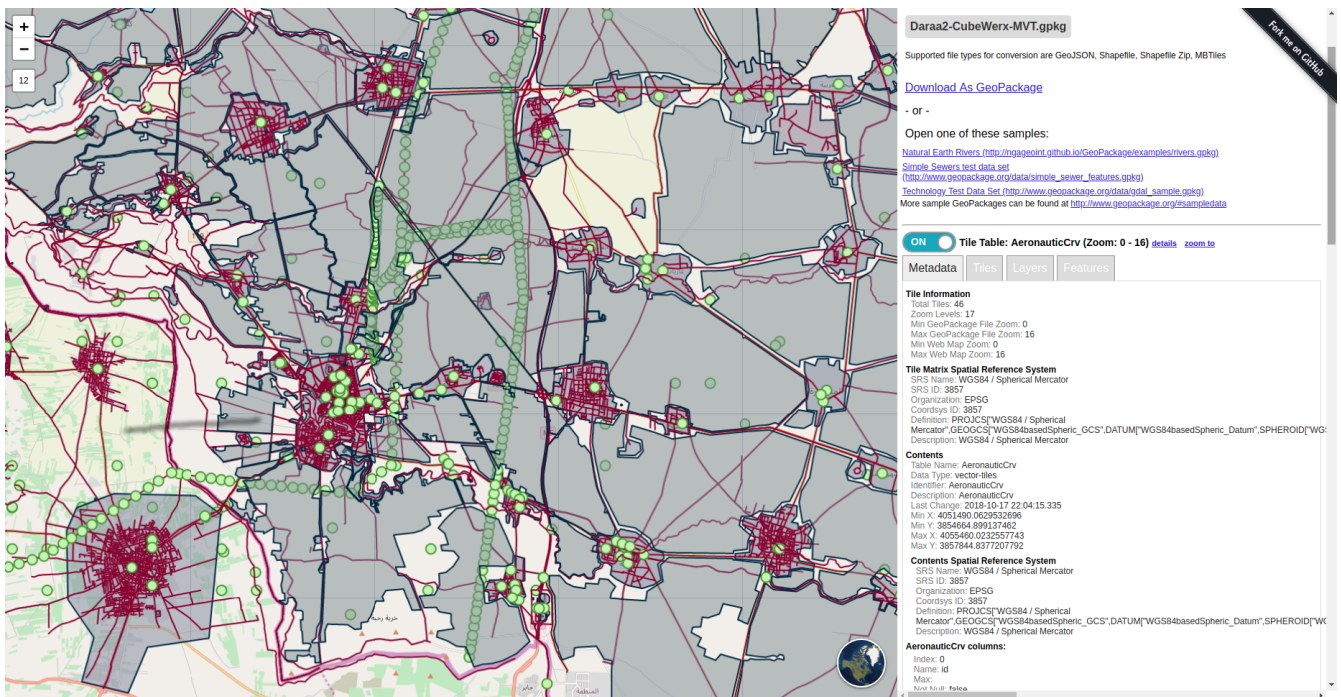


Figure 15. CubeWerx Vector Tiles Rendered on the Image Matters Web Client

### 5.3.3. Image Matters Client #2 - Mobile Client

The Vector Tile mobile client creates a tile provider that returns a rendered tile image given a column, row, and zoom level. The client first searches the specified Vector Tiles table in the GeoPackage for a Vector Tile blob at that column, row, and zoom level. The client then parses the BLOB, then looks for styling information. Styling information is included in the GeoPackage as a JSON file. The client then renders the features using that styling information, and saves an image to a tile cache. Layers are saved individually, so that they may be enabled and disabled separately. The map engine can simply display these images at the correct coordinates, which is handled automatically by the tile provider. This version of the client does not have well-specified styling rules. This is something that should be looked at in the future.



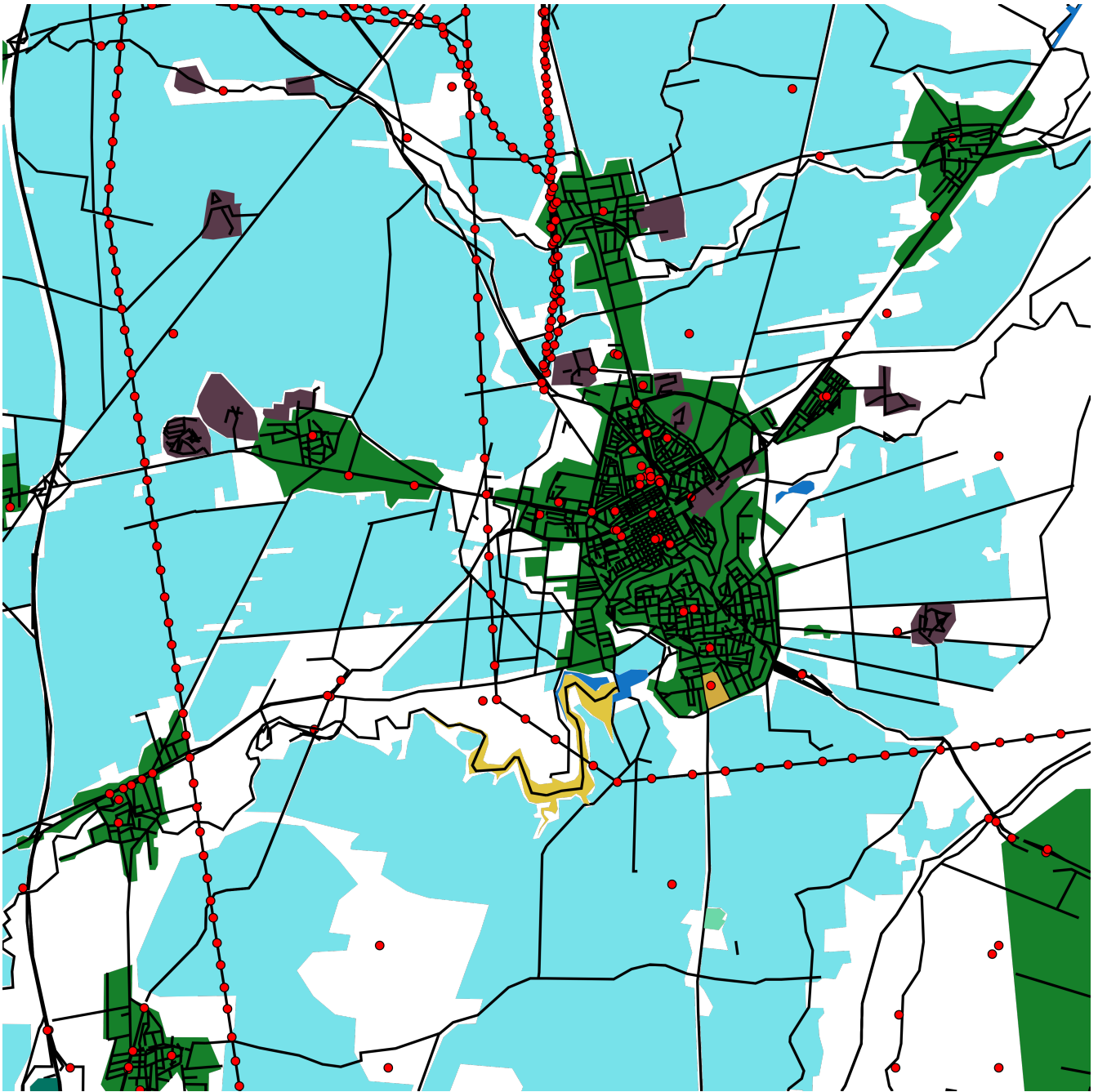
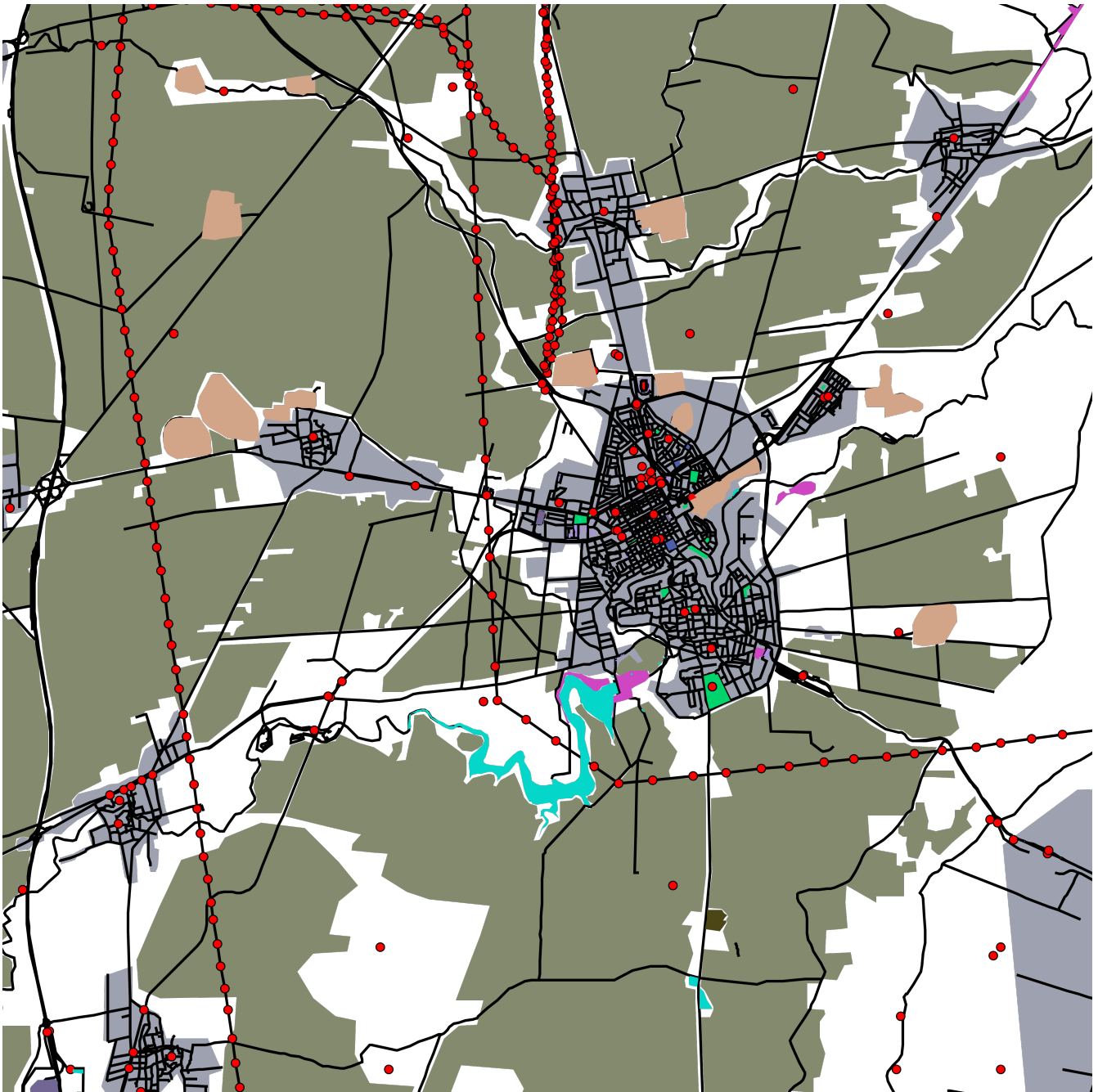


Figure 16. Ecere Vector Tiles Rendered on the Image Matters Mobile Client



*Figure 17. Compusult Vector Tiles Rendered on the Image Matters Mobile Client*



*Figure 18. CubeWerx Vector Tiles Rendered on the Image Matters Mobile Client*

### **5.3.4. Ecere**

Support for GeoPackages and vector tiles extensions as a geospatial data source has been added to Ecere's GNOSIS software libraries. The capability is available from within Ecere's GNOSIS Cartographer GIS tool, as well as from any application built using the GNOSIS SDK (whether for desktop, web or mobile). Support for both multiple layers per tiles, or separate tile sets; attributes embedded within the tiles or using attribute tables; for all the different encodings (MVT, GeoJSON, GNOSIS Map Tiles, GML, GeoECON); support for arbitrary tiling schemes; and support for retrieving associated style sheets from within the GeoPackage were implemented.



Figure 19. Ecere Vector Tiles GeoPackage Rendered in Ecere's Visualization Client



Figure 20. Compusult Vector Tiles GeoPackage Rendered in Ecere's Visualization Client



Figure 21. CubeWerx Vector Tiles GeoPackage Rendered in Ecere's Visualization Client

## 5.4. Technology Integration Experiments

The table below presents the results of the Technology Integration Experiments (TIEs) for this Pilot.

Table 1. Technology Integration Experiments

	<b>Ecere</b>	<b>CubeWerx</b>	<b>Compusult</b>	<b>Notes</b>
Compusult	done	done	done	
Image Matters Web	done	done	done	MVT Only
Image Matters Mobile	done	done	done	MVT Only
Ecere	done	done	done	

## 5.5. Denied, Degraded, Intermittent, or Limited (DDIL) Environments

Many operational environments suffer from Denied, Degraded, Intermittent, or Limited (DDIL) communications. Traditional web services either do not function at all or function with reduced capability in these environments. The geospatial industry has been looking for ways to maintain operational capabilities despite communications limitations. This invariably means that it is important to find a way to deliver operational data to mobile and handheld devices and producing applications and creating software applications capable of using that data despite the limited resources available on those devices.

This Pilot gave participants an opportunity to investigate an approach for mitigating one of the major challenges of delivering geospatial data to mobile and handheld devices, high storage clients. While raster data is relatively convenient to work with and can be used quite readily on mobile and handheld devices, this convenience comes at a cost of large file sizes. Most mobile and handheld devices do not have the physical storage capacity to store high resolution raster data over a large area. Vector tiles address this problem by storing the data as points, lines, and polygons instead of pictures. For synthetic data, a dataset with vector tiles is often orders of magnitude smaller than equivalent data stored as raster tiles.

One of the primary advantages of GeoPackage is that it is well-suited to DDIL operations due to its self-contained structure. Operation in DDIL environments is a common GeoPackage use case, but it is impaired by the high storage requirements of conventional raster tile pyramids. All of the GeoPackage clients demonstrated in this Pilot are capable of operating in a completely disconnected mode as long as the GeoPackage is physically delivered to the device beforehand. By enhancing GeoPackage software with vector tiles capabilities, the participants have demonstrated the benefits of vector tiles in these environments.

# Chapter 6. Vector Tiles Extensions (Informative)

Five GeoPackage extensions were developed as part of this Pilot project. This section presents each one using the [GeoPackage Extension Template](http://www.geopackage.org/spec120/#extension_template) [http://www.geopackage.org/spec120/#extension\_template]. This section is informative, providing information relevant to a developer or administrator who wishes to understand how these extensions are supposed to work. Potential normative requirements are presented in [Requirements \(Normative\)](#).

## 6.1. Vector Tiles Extension

### 6.1.1. Extension Title

Vector Tiles

### 6.1.2. Introduction

The GeoPackage Vector Tiles extension defines the rules and requirements for encoding vector tiles in a GeoPackage data store.

#### WARNING

This extension does not define an encoding for vector tiles. To be usable, an additional extension such as [Mapbox Vector Tiles Extension](#) or [GeoJSON Vector Tiles Extension](#) must also be used.

This extension, like all GeoPackage extensions, is intended to be transparent and to not interfere with GeoPackage-compliant, but non-supporting, software packages.

### 6.1.3. Extension Author

Image Matters LLC, in collaboration with the participants of the OGC Vector Tiles Pilot.

### 6.1.4. Extension Name or Template

`im_vector_tiles` (If this extension is adopted by OGC, then `gpkg_vector_tiles` will be named as an alias.)

### 6.1.5. Extension Type

This extension provides new requirements dependent on GeoPackage [Clause 2.2 \(tiles\)](#) [http://www.geopackage.org/spec120/index.html#tiles].

### 6.1.6. Applicability

This extension defines an alternate way to encode feature information into a GeoPackage.

## 6.1.7. Scope

read-write

## 6.1.8. Specification

If this extension is in use, then all of the [Tiles Option](http://www.geopackage.org/guidance/getting-started.html#tiles) [http://www.geopackage.org/guidance/getting-started.html#tiles] applies. The individual tiles (`tile_data` in a tile pyramid user data table) are vector tiles.

**NOTE** Individual vector tiles MAY be deflate compressed.

There are two additional required metadata tables, `gpkgext_vt_layers` and `gpkgext_vt_fields`, that mirror the [vector\\_layers key from the JSON object from the metadata from MBTiles](https://github.com/mapbox/mbtiles-spec/blob/master/1.3/spec.md#vector_layers) [https://github.com/mapbox/mbtiles-spec/blob/master/1.3/spec.md#vector\_layers]. This allows client software to understand the feature schemas without having to open individual tiles.

### `gpkg_extensions`

To use this extension, add the following rows to this table.

Table 2. `gpkg_extensions` Table Rows

<b>table_name</b>	<b>column_name</b>	<b>extension_name</b>	<b>definition</b>	<b>scope</b>
<code>gpkgext_vt_layers</code>	NULL	<code>im_vector_tiles</code>	<i>a reference to this file</i>	<i>read-write</i>
<code>gpkgext_vt_fields</code>	NULL	<code>im_vector_tiles</code>	<i>a reference to this file</i>	<i>read-write</i>

### `gpkg_contents`

Like any other content type, add a row for each tile set, using a `data_type` of "vector-tiles".

### `gpkg_spatial_ref_sys`

Like any other content type, the Spatial Reference System (SRS) for the content to be stored must be registered in this table. See clause 1.1.2 in the core GeoPackage standard. While any valid SRS may be used, Web Mercator (EPSG:3857) maintains compatibility with MVT.

### `gpkgext_vt_layers`

The `gpkgext_vt_layers` table describes the vector layers in a vector tile set. The columns in this table are:

- `id` is a primary key
- `table_name` matches the entry in `gpkg_contents`
- `name` is the layer name
- `description` is an optional text description
- `minzoom` and `maxzoom` are the optional integer minimum and maximum zoom levels



## gpkgext\_vt\_fields

The `gpkgext_vt_fields` table describes the fields (attributes) for a vector tile layer. The columns in this table are:

- `id` is a primary key
- `layer_id` is a foreign key to `id` in `gpkgext_vt_layers`
- `name` is the field name
- `type` is either "String", "Number", or "Boolean"

## 6.2. Mapbox Vector Tiles Extension

### 6.2.1. Extension Title

Mapbox Vector Tiles

### 6.2.2. Introduction

The GeoPackage Mapbox Vector Tiles extension defines the rules and requirements for encoding vector tiles in a GeoPackage data store as Mapbox Vector Tiles. This extension is based on the [Mapbox Vector Tiles \(MVT\) specification](https://www.mapbox.com/vector-tiles/specification/) [https://www.mapbox.com/vector-tiles/specification/] [version 2.1](https://github.com/mapbox/vector-tile-spec/tree/master/2.1) [https://github.com/mapbox/vector-tile-spec/tree/master/2.1]. Note that this format uses [Google Protocol Buffers](https://github.com/google/protobuf) [https://github.com/google/protobuf] as the content encoding for each tile.

This extension, like all GeoPackage extensions, is intended to be transparent and to not interfere with GeoPackage-compliant, but non-supporting, software packages.

### 6.2.3. Extension Author

Image Matters LLC, in collaboration with the participants of the OGC Vector Tiles Pilot.

### 6.2.4. Extension Name or Template

`im_vector_tiles_mapbox` (If this extension is adopted by OGC, then `gpkg_vector_tiles_mapbox` will be named as an alias.)

### 6.2.5. Extension Type

This extension defines an encoding for the [Vector Tiles Extension](#).

### 6.2.6. Applicability

This extension defines a specific encoding for Vector Tiles in a GeoPackage.

### 6.2.7. Scope

read-write

## 6.2.8. Specification

If this extension is in use, then all of [Vector Tiles Extension](#) applies.

### User Data Tables

Like other tile-based content, the physical data is stored in [user-defined tiles tables](#) [<http://www.geopackage.org/guidance/getting-started.html#user-data-tables>]. The `tile_data` is a Google Protocol Buffer as [defined by MVT](#) [[https://github.com/mapbox/vector-tile-spec/blob/master/2.1/vector\\_tile.proto](https://github.com/mapbox/vector-tile-spec/blob/master/2.1/vector_tile.proto)].

### `gpkg_extensions`

To use this extension, add a row to this table for each tile pyramid user data table.

Table 3. `gpkg_extensions` Table Rows

table_name	column_name	extension_name	definition	scope
<i>tile pyramid user data table name</i>	<code>tile_data</code>	<code>im_vector_tiles_mapbox</code>	<i>a reference to this file</i>	<i>read-write</i>

## 6.3. GeoJSON Vector Tiles Extension

### 6.3.1. Extension Title

GeoJSON Vector Tiles

### 6.3.2. Introduction

The GeoPackage Vector Tiles extension defines the rules and requirements for encoding vector tiles in a GeoPackage data store using [The GeoJSON Format](#) [<https://tools.ietf.org/html/rfc7946>].

This extension, like all GeoPackage extensions, is intended to be transparent and to not interfere with GeoPackage-compliant, but non-supporting, software packages.

### 6.3.3. Extension Author

Image Matters LLC, in collaboration with the participants of the OGC Vector Tiles Pilot.

### 6.3.4. Extension Name or Template

`im_vector_tiles_geojson` (If this extension is adopted by OGC, then `gpkg_vector_tiles_geojson` will be named as an alias.)

### 6.3.5. Extension Type

This extension defines an encoding for the [Vector Tiles Extension](#).

### 6.3.6. Applicability

This extension defines a specific encoding for Vector Tiles in a GeoPackage.

### 6.3.7. Scope

read-write

### 6.3.8. Specification

If this extension is in use, then all of [Vector Tiles Extension](#) applies.

#### User Data Tables

Like other tile-based content, the physical data is stored in a [GeoJSON Feature Collection](#) [<https://tools.ietf.org/html/rfc7946#section-3.3>].

#### `gpkg_extensions`

To use this extension, add a row to this table for each tile pyramid user data table.

Table 4. `gpkg_extensions` Table Rows

table_name	column_name	extension_name	definition	scope
<i>tile pyramid user data table name</i>	<code>tile_data</code>	<code>im_vector_tiles_geojson</code>	<i>a reference to this file</i>	<i>read-write</i>

## 6.4. OWS Context Extension

**WARNING** | This subsection is under discussion and may change radically.

### 6.4.1. Extension Title

OWS Context

### 6.4.2. Introduction

This extension provides a mechanism for storing [OWS Context](#) [<http://owscontext.org>] content in a GeoPackage. It is aligned with the [OWS Context Conceptual Model](#) [[https://portal.opengeospatial.org/files/?artifact\\_id=55182](https://portal.opengeospatial.org/files/?artifact_id=55182)].

### 6.4.3. Extension Author

Image Matters LLC, in collaboration with the participants of the OGC Vector Tiles Pilot and the OWS Context SWG.

### 6.4.4. Extension Name or Template

`owscontext` (will become `gpkg_owscontext` if adopted by OGC)

### 6.4.5. Extension Type

New requirement dependent on [Clause 1](#) [<http://www.geopackage.org/spec/#core>].

## 6.4.6. Applicability

This extension adds an additional level of organization to existing GeoPackage data.

## 6.4.7. Scope

read-write

## 6.4.8. Specification

### gpkg\_extensions

To use this extension, add the following rows to this table as described in [gpkg\\_extensions](http://www.geopackage.org/guidance/getting-started.html#gpkg_extensions) [http://www.geopackage.org/guidance/getting-started.html#gpkg\_extensions].

Table 5. GeoPackage Extensions Table Values

table_name	column_name	extension_name	definition	scope
gpkgext_context	null	ows_context	see below	read-write
gpkgext_context_resources	null	ows_context	see below	read-write
gpkgext_context_of_ferings	null	ows_context	see below	read-write
gpkgext_context_styles	null	ows_context	see below	read-write

#### NOTE

The values in the **definition** column SHOULD refer in some human-readable way to this extension specification. If the RTE is adopted by OGC, it will gain the "gpkg\_" prefix and get a different definition permalink.

### gpkgext\_context

This table describes OWS Contexts. Add a row to this table for each map or resource collection in the GeoPackage. It has the following structure:

Table 6. OWS Context Table Definition

Column Name	Column Type	Column Description	Null Allowed	Default	Key
id	INTEGER	Autoincrement primary key	no		PK
title	TEXT	A human-readable title for the OWS Context document	no		

Column Name	Column Type	Column Description	Null Allowed	Default	Key
abstract	TEXT	A human-readable description of the OWS Context document purpose and/or content	yes	"	
last_change	DATETIME	timestamp of last change to content, in ISO 8601 format	no	strftime('%Y-%m-%dT%H:%M:%fZ', 'now')	
min_x	DOUBLE	Bounding box minimum easting or longitude for the users of the context document	yes		
min_y	DOUBLE	Bounding box minimum northing or latitude for the users of the context document	yes		
max_x	DOUBLE	Bounding box maximum easting or longitude for the users of the context document	yes		
max_y	DOUBLE	Bounding box maximum northing or latitude for the users of the context document	yes		

Column Name	Column Type	Column Description	Null Allowed	Default	Key
<code>srs_id</code>	INTEGER	Spatial Reference System ID: <code>gpkg_spatial_ref_sys.srs_id</code> for the geographic extents	yes		FK
<code>author</code>	TEXT	Identifier for the author of the document	yes		
<code>publisher</code>	TEXT	Identifier for the publisher of the document	yes		
<code>creator</code>	TEXT	The tool/application used to create the context document and its properties	yes		
<code>rights</code>	TEXT	Rights which apply to the context document	yes		
<code>keywords</code>	TEXT	Comma-delimited list of keywords related to this context document	yes		
<code>metadata_id</code>	INTEGER	<code>id</code> from <code>gpkg_metadata</code>	yes		
<code>min_time</code>	DATETIME	Beginning of time interval, in ISO 8601 format	yes		
<code>max_time</code>	DATETIME	End of time interval, in ISO 8601 format	yes		

**NOTE**

The rights described apply to the Context Document itself and not to any of its contents. The intent of the temporal and spatial extents is to indicate to a GeoPackage client the expected view of the information in area in time, not to describe the referenced resources themselves.

## gpkgext\_context\_resources

This table represents an `owc:SQLResource`. Add a row to this table for each map or resource collection in the GeoPackage. It has the following structure:

Table 7. OWS Context Resources Table Definition

Column Name	Column Type	Column Description	Null Allowed	Default	Key
<code>id</code>	INTEGER	Autoincrement primary key	no		PK
<code>context_id</code>	INTEGER	<code>id</code> from <code>gpkgext_context</code>	no		FK
<code>title</code>	TEXT	A human-readable title for the OWS Context resource	no		
<code>abstract</code>	TEXT	A human-readable description of the OWS Context document purpose and/or content	yes	"	
<code>author</code>	TEXT	Identifier for the author of the document	yes		
<code>publisher</code>	TEXT	Identifier for the publisher of the document	yes		
<code>rights</code>	TEXT	Rights which apply to the context document	yes		
<code>min_x</code>	DOUBLE	Bounding box minimum easting or longitude for the users of the context document	yes		

Column Name	Column Type	Column Description	Null Allowed	Default	Key
min_y	DOUBLE	Bounding box minimum northing or latitude for the users of the context document	yes		
max_x	DOUBLE	Bounding box maximum easting or longitude for the users of the context document	yes		
max_y	DOUBLE	Bounding box maximum northing or latitude for the users of the context document	yes		
srs_id	INTEGER	Spatial Reference System ID: <code>gpkg_spatial_ref_sys.srs_id</code> for the geographic extents	yes		FK
min_time	DATETIME	Beginning of time interval, in ISO 8601 format	yes		
max_time	DATETIME	End of time interval, in ISO 8601 format	yes		
description	TEXT	A reference to a description of the Context resource in alternative format	yes		



Column Name	Column Type	Column Description	Null Allowed	Default	Key
active	BOOLEAN	This flag indicates the state of the resource within the context document. It can be interpreted by the caller as required (this may be defined in a profile or in the specific service extensions)	yes	TRUE	
keywords	TEXT	Comma-delimited list of keywords related to this context document	yes		
min_scale_denominator	DOUBLE	Minimum scale for the display of the layer	yes		
max_scale_denominator	DOUBLE	Maximum scale for the display of the layer	yes		
order	DOUBLE	The ascending order of the resource	yes		
requestURL	TEXT	Service Request URL or file URL	yes		
code	TEXT	Code identifying the type of resource	no		
layer_name	TEXT	A single layer, table, or view name	yes		
query	TEXT	The actual SQL or HTTP query	yes		

## gpkgext\_context\_offerings

This table represents an owc:Offering. Add a row to this table for each map or resource collection in the GeoPackage. It has the following structure:

Table 8. OWS Context Offerings Table Definition

Column Name	Column Type	Column Description	Null Allowed	Default	Key
id	INTEGER	Autoincrement primary key	no		PK
resource_id	INTEGER	id from gpkgext_context_resources	no		FK
code	TEXT	Code identifying the type of offering	no		
stylesheet_id	INTEGER	id from gpkgext_stylesheets	yes		FK
method	TEXT	Name of operation method request	no		
type	TEXT	Media type (formerly known as MIME-Type) of the return result	no		
contents	BLOB	Actual data (for inline data)	yes		

## gpkgext\_styles

The gpkgext\_styles table implements owc:StyleSet. Add a row to this table for each style in the GeoPackage. It has the following structure:

Table 9. Styles Table Definition

Column Name	Column Type	Column Description	Null	Default	Key
id	INTEGER	Autoincrement primary key	no		PK
style	BLOB	the actual style	no		
content_type	TEXT	Media type (formerly known as MIME-Type) of style	no		

## 6.5. Vector Tiles Attributes Extension

The candidate [GeoPackage Related Tables Extension](https://www.geopackage.org/18-000.html) [https://www.geopackage.org/18-000.html] (RTE) defines the rules and requirements for creating relationships in a GeoPackage data store between geospatial data tables and other tables that contain or reference related content such as attributes or media. The core of the Related Tables Extension is a mapping between existing table types defined by GPKG 1.2 - features, tiles and attributes. The mapping is defined by a new kind of table defined by the Related Tables Extension. The mapping table links related rows in those tables of those types by reference to their primary keys.

The "GeoPackage Extension for Related Tables" allows a GeoPackage to contain additional data that is related to geospatial (e.g., features) or attributes data. As an example, this can be used to establish a many-to-many relationship between features (e.g., points, lines, or areas) and multimedia files. By definition, the "left" side of the relationship is the "base" data and the "right" side of the relationship is the "related" data. When relating vector tiles with attributes, the vector tiles are the "base" data and the attributes are the "related" data.

### NOTE

At the time of writing, the RTE has been published in draft form (OGC 18-000) as part of an open comment period. This version of the RTE does not define a requirements class to map tiles tables with attributes tables. This section defines a requirements class that will fill this need. A note has been added to the draft RTE standard to indicate that additional requirements classes are possible. For information on using the Related Tables Extension, see the [Getting Started Guide](https://github.com/engeos/geopackage-related-tables/wiki/Getting-Started) [https://github.com/engeos/geopackage-related-tables/wiki/Getting-Started].

### 6.5.1. gpkg\_extensions

To use this extension, add the following rows to this table as described in [gpkg\\_extensions](http://www.geopackage.org/guidance/getting-started.html#gpkg_extensions) [http://www.geopackage.org/guidance/getting-started.html#gpkg\_extensions].

Table 10. *gpkg\_extensions* Table Rows

table_name	column_name	extension_name	definition	scope
<a href="#">gpkgext_relations</a>	null	<a href="#">related_tables</a>	<a href="https://github.com/engeos/geopackage-related-tables">https://github.com/engeos/geopackage-related-tables</a>	<a href="#">read-write</a>
<i>name of actual User-defined Mapping Table</i>	null	<a href="#">related_tables</a>	<a href="https://github.com/engeos/geopackage-related-tables">https://github.com/engeos/geopackage-related-tables</a>	<a href="#">read-write</a>

### NOTE

If the RTE is adopted by OGC, it will gain the "gpkg\_" prefix and get a different definition permalink.

### 6.5.2. gpkgext\_relations

This table describes extended relationships. The table requires the following columns:

Table 11. *gpkgext\_relations* Table Rows

Column	Description
<code>id</code>	primary key
<code>base_table_name</code>	Name of the vector tiles data table
<code>base_primary_column</code>	<code>id</code> (all user-defined vector tiles tables have this column)
<code>related_table_name</code>	Name of the user-defined attributes table
<code>related_primary_column</code>	Name of the primary key column in <code>related_table_name</code>
<code>relation_name</code>	<code>vector_tiles_attributes</code>
<code>mapping_table_name</code>	Name of a <a href="#">User-defined Mapping Table</a>

Add a row to this table for each vector tileset that has its own attributes.

### 6.5.3. User-defined Mapping Table

A user-defined mapping table describes the many-to-many relationships between base data (tiles) and related data (attributes). A user-defined mapping table requires at least the following columns:

Table 12. *gpkgext\_relations* Table Rows

Column	Description
<code>base_id</code>	The primary key value of the base data table (tile ID)
<code>related_id</code>	The primary key value of the related data table (attributes ID)

Add a row to this table for each feature/tile instance.

# Chapter 7. Requirements (Normative)

This section describes the normative requirements for the GeoPackage extensions described in this document. It is in the form of a draft standard that may ultimately be adopted by OGC.

## 7.1. Vector Tiles Extension

<b>Requirements Class : Vector Tiles</b>	
<a href="http://www.opengis.net/spec/gpkg-vt/1.0/vector-tiles">http://www.opengis.net/spec/gpkg-vt/1.0/vector-tiles</a>	
Target type	Token
Dependency	www.opengis.net/spec/gpkg/1.2.0/opt/tiles

### WARNING

This extension does not define an encoding for vector tiles. To be usable, an additional extension such as [Mapbox Vector Tiles Extension](#) or [GeoJSON Vector Tiles Extension](#) must also be used.

### 7.1.1. `gpkg_contents`

#### Table Values

<b>Requirement VTE1 – Vector Tiles Presence</b>	<i>/req/vector_tiles/presence</i>  A GeoPackage with at least one row in <code>gpkg_contents</code> with a <code>data_type</code> of "vector-tiles" SHALL comply with this extension.
---	---

<b>Requirement VTE2 – <code>gpkg_contents</code> Rows</b>	<i>/req/vector_tiles/gcr</i>  Each row in <code>gpkg_contents</code> with a <code>data_type</code> of "vector-tiles" SHALL be a vector tiles set as described by this extension.
---	--

### NOTE

A vector tile MAY be [deflate compressed](https://tools.ietf.org/html/rfc1951) [https://tools.ietf.org/html/rfc1951]. For maximum interoperability, a client SHOULD inspect a tile, determine whether it is compressed, and respond accordingly.

### 7.1.2. `gpkg_extensions`

#### Table Values

<b>Requirement VTE3 – gpkg_extensions Rows</b>	<i>/req/vector_tiles/ger</i>  A GeoPackage that complies with this extension SHALL contain rows in the <code>gpkg_extensions</code> table for <code>gpkgext_vt_layers</code> and <code>gpkgext_vt_fields</code> as described in <a href="#">gpkg_extensions Table Rows</a> .
--	--

<b>Requirement VTE4 – gpkg_extensions Encoding</b>	<i>/req/vector_tiles/ger-encoding</i>  For each vector tiles set, there SHALL be one and only one corresponding row in <code>gpkg_extensions</code> specifying an extension describing the encoding for the vector tiles in the <code>tile_data</code> column of the tile pyramid user data table.
--	--

### 7.1.3. `gpkgext_vt_layers`

#### Table Definition

<b>Requirement VTE5 – gpkgext_vt_layers Table</b>	<i>/req/vector_tiles/gpkgext_vt_layers/table</i>  A GeoPackage that complies with this extension SHALL contain a <code>gpkgext_vt_layers</code> table as per <a href="#">Vector Tiles Layers Table Definition</a> and <a href="#">[gpkgext_vt_layers_sql]</a> .
---	---

Table 13. Vector Tiles Layers Table Definition

Column Name	Column Type	Column Description	Null Allowed	Key	Unique
<code>id</code>	INTEGER	Autoincrement primary key	no	PK	yes
<code>table_name</code>	TEXT	Name of the table user-defined tiles table containing the vector tiles	no	FK	yes (see <a href="#">[vte4]</a> )
<code>name</code>	TEXT	Layer Name	no		yes (see <a href="#">[vte4]</a> )
<code>description</code>	TEXT	Text description	yes		
<code>minzoom</code>	INTEGER	Minimum zoom level	yes		
<code>maxzoom</code>	INTEGER	Maximum zoom level	yes		

## Table Values

<b>Requirement VTE6</b> – <b>gpkgext_vt_layers</b> <b>table reference</b>	<i>/req/vector-tiles/gpkgext_vt_layers/table_ref</i>  For each row in <b>gpkgext_vt_layers</b> , there SHALL be a table or view of the name referenced in <b>table_name</b> and that table or view SHALL have an entry in <b>gpkg_contents</b> .
<b>Requirement VTE7</b> – <b>gpkgext_relations</b> <b>Base Table</b>	<i>/req/vector-tiles/gpkgext_vt_layers/uniqueness</i>  The rows in <b>gpkgext_vt_layers</b> SHALL have a jointly unique <b>table_name</b> and <b>name</b> .

### 7.1.4. gpkgext\_vt\_fields

#### Table Definition

<b>Requirement VTE8</b> – <b>gpkgext_vt_fields</b> <b>Table</b>	<i>/req/vector_tiles/gpkgext_vt_fields/table</i>  A GeoPackage that complies with this extension SHALL contain a <b>gpkgext_vt_fields</b> table as per <a href="#">Vector Tiles Fields Table Definition</a> and <a href="#">[gpkgext_vt_fields_sql]</a> .
---	---

Table 14. Vector Tiles Fields Table Definition

Column Name	Column Type	Column Description	Null Allowed	Key
<b>id</b>	INTEGER	Autoincrement primary key	no	PK
<b>layer_id</b>	INTEGER	An <b>id</b> from <b>gpkgext_vt_layers</b>	no	FK
<b>name</b>	TEXT	Field name	no	
<b>type</b>	TEXT	"String", "Number", or "Boolean"	no	

## Table Values

<b>Requirement VTE9</b> – <b>gpkgext_vt_fields</b> <b>table reference</b>	<i>/req/vector-tiles/gpkgext_vt_fields/table_ref</i>  For each row in <b>gpkgext_vt_fields</b> , the <b>layer_id</b> SHALL have a corresponding entry in <b>gpkgext_vt_layers</b> .
---	---

## 7.2. Mapbox Vector Tiles Extension

<b>Requirements Class : Mapbox Vector Tiles</b>	
<a href="http://www.opengis.net/spec/gpkg-vt/1.0/mapbox-vector-tiles">http://www.opengis.net/spec/gpkg-vt/1.0/mapbox-vector-tiles</a>	
Target type	Token
Dependency	<a href="http://www.opengis.net/spec/gpkg/1.2.0/opt/tiles">www.opengis.net/spec/gpkg/1.2.0/opt/tiles</a>
Dependency	<a href="http://www.opengis.net/spec/gpkg-vt/1.0/vector-tiles">http://www.opengis.net/spec/gpkg-vt/1.0/vector-tiles</a>

### NOTE

At the time of writing, Mapbox Vector Tiles is not a standard. Because of this, this extension is not a candidate for becoming an OGC standard. However, it is a candidate for being a [community extension](http://www.geopackage.org/extensions.html) [http://www.geopackage.org/extensions.html] and an [OGC Best Practice](http://www.opengeospatial.org/docs/bp) [http://www.opengeospatial.org/docs/bp].

### 7.2.1. gpkg\_extensions

#### Table Values

<b>Requirement MVTE1 – gpkg_extensions Rows</b>	<i>/req/mvte/ger</i>  A GeoPackage that contains a row in the <a href="#">gpkg_extensions</a> table for <a href="#">im_vector_tiles_mapbox</a> or its alias as described in <a href="#">gpkg_extensions Table Rows</a> SHALL comply with the Mapbox Vector Tiles Extension as described by this document.
---	---

### 7.2.2. User Defined Tiles Tables

#### Table Values

<b>Requirement MVTE2 – Tile Format</b>	<i>/req/mvte/tile_format</i>  For each vector tiles set governed by this extension, <a href="#">tile_data</a> column values of the corresponding tile pyramid user data table SHALL contain Mapbox Vector Tiles as per the <a href="#">Mapbox Vector Tiles (MVT) specification</a> [https://www.mapbox.com/vector-tiles/specification/] <a href="#">version 2.1</a> [https://github.com/mapbox/vector-tile-spec/tree/master/2.1].
--	---

## 7.3. GeoJSON Vector Tiles Extension

<b>Requirements Class : GeoJSON Vector Tiles</b>	
<a href="http://www.opengis.net/spec/gpkg-vt/1.0/geojson-vector-tiles">http://www.opengis.net/spec/gpkg-vt/1.0/geojson-vector-tiles</a>	
Target type	Token



Dependency	<a href="http://www.opengis.net/spec/gpkg/1.2.0/opt/tiles">www.opengis.net/spec/gpkg/1.2.0/opt/tiles</a>
Dependency	<a href="http://www.opengis.net/spec/gpkg-vt/1.0/vector-tiles">http://www.opengis.net/spec/gpkg-vt/1.0/vector-tiles</a>

### 7.3.1. gpkg\_extensions

#### Table Values

<b>Requirement GVTE1 – gpkg_extensions Rows</b>	<p><i>/req/gvte/ger</i></p> <p>A GeoPackage that contains a row in the <b>gpkg_extensions</b> table for <b>im_vector_tiles_geojson</b> or its alias as described in <b>gpkg_extensions Table Rows</b> SHALL comply with the GeoJSON Extension as described by this document.</p>
---	--

### 7.3.2. User Defined Tiles Tables

#### Table Values

<b>Requirement GVTE2 – Tile Format</b>	<p><i>/req/gvte/tile_format</i></p> <p>For each vector tiles set governed by this extension, <b>tile_data</b> column values of the corresponding tile pyramid user data table SHALL contain GeoJSON Feature Collections as per <b>The GeoJSON Format</b> [<a href="https://tools.ietf.org/html/rfc7946">https://tools.ietf.org/html/rfc7946</a>].</p>
--	---

## 7.4. OWS Context Extension

#### NOTE

This section is intentionally left blank because it is the subject of continuing discussions in the OWS Context and GeoPackage SWGs.

## 7.5. Vector Tiles Attributes Extension

<b>Requirements Class : Vector Tiles Attributes</b>	
<a href="http://www.opengis.net/spec/gpkg-vt/1.0/geojson-vector-tiles">http://www.opengis.net/spec/gpkg-vt/1.0/geojson-vector-tiles</a>	
Target type	Token
Dependency	<a href="http://www.opengis.net/spec/gpkg/1.2.0/opt/tiles">www.opengis.net/spec/gpkg/1.2.0/opt/tiles</a>
Dependency	<a href="http://www.opengis.net/spec/gpkg/1.2.0/opt/attributes">www.opengis.net/spec/gpkg/1.2.0/opt/attributes</a>
Dependency	<a href="http://www.opengis.net/spec/gpkg-vt/1.0/vector-tiles">http://www.opengis.net/spec/gpkg-vt/1.0/vector-tiles</a>
Dependency	<a href="http://www.opengis.net/spec/gpkg-rt/1.0/table-defs">http://www.opengis.net/spec/gpkg-rt/1.0/table-defs</a>

## 7.5.1. gpkgext\_relations

### Table Values

<b>Requirement VTAE1</b> – <b>gpkgext_relations.relation_name</b>	<i>/req/vtae/ge_r/relation_name</i>  For each row in <b>gpkgext_relations</b> that has a <b>relation_name</b> of "vector_tiles_attributes", this requirements class SHALL apply.
<b>Requirement VTAE2</b> – <b>gpkgext_relations.base_table_name</b>	<i>/req/vtae/ge_r/base_table_name</i>  For each relevant row in <b>gpkgext_relations</b> , the <b>base_table_name</b> SHALL correspond to a GeoPackage tiles table or view as per <a href="http://www.geopackage.org/spec120/#r34">http://www.geopackage.org/spec120/#r34</a> .
<b>Requirement VTAE3</b> – <b>gpkgext_relations.base_table_column</b>	<i>/req/vtae/ge_r/base_table_column</i>  For each relevant row in <b>gpkgext_relations</b> , the <b>base_table_column</b> SHALL be "id".
<b>Requirement VTAE4</b> – <b>gpkgext_relations.related_table_name</b>	<i>/req/vtae/ge_r/related_table_name</i>  For each relevant row in <b>gpkgext_relations</b> , the <b>related_table_name</b> SHALL correspond to a GeoPackage attributes table or view as per <a href="http://www.geopackage.org/spec120/#r118">http://www.geopackage.org/spec120/#r118</a> .
<b>Requirement VTAE5</b> – <b>gpkgext_relations.related_table_column</b>	<i>/req/vtae/ge_r/related_table_column</i>  For each relevant row in <b>gpkgext_relations</b> , the <b>related_table_column</b> SHALL be the name of the primary key column in the corresponding related (attributes) table.

# Chapter 8. Discussion

## 8.1. Mapbox vs. GeoJSON Vector Tiles

Mapbox Vector Tiles provide clear performance benefits to either the features option or GeoJSON because of its use of protocol buffers. The benefits of GeoJSON Vector Tiles are less clear but it may prove useful to have that capability in the future because GeoJSON is emerging as an increasingly common exchange format. As per the Pilot sponsors, Mapbox Vector Tiles is the first requirement for this Pilot project. GeoJSON Vector Tiles is a secondary requirement that was not prioritized due to resource constraints. In the interest of extensibility, this ER proposes to modularize the encodings into separate extensions so that participants could support GeoJSON if desired.

## 8.2. Tile Compression

One participant produced three separate GeoPackages, one with MVT, one GeoJSON, and one JPG/PNG. The raw GeoJSON file is 7.1× the size of the MVT file, which is as expected, and the image file is about the same size as the MVT. When Zipped, the GeoJSON is 3.3× the size of the MVT and the image file is 6.1× the size of the MVT. MVT and especially GeoJSON benefit quite a lot from Deflate compression, shrinking by 90% and 96%, respectively. Surprisingly, MVT and Google Protocol Buffers do not include any kind of inherent entropy-encoding compression method. (The JPEG/PNG images are already compressed.)

The MBTiles specification allows individual tiles to be deflate compressed inside the SQLite database (see <https://github.com/mapbox/mbtiles-spec/issues/43> for more details). This capability was carried over to GeoPackage. Unfortunately it was added towards the end of the Pilot after many of the test GeoPackages were developed. Some clients were able to demonstrate the capability through duck-typing, but no interoperable solution was developed for declaring that compression is in use for a particular tile pyramid.

## 8.3. Attributes

Vector Tiles include both geometry and attribute information. The attribute information is useful for styling and to enable certain analytical operations. While it is possible to open individual vector tiles and access the attributes that way, this is not efficient and it does not use the underlying SQLite database capabilities. The participants have proposed an approach (described in [Vector Tiles Attributes Extension](#)) to separate the attributes from the geometries. This way, the ensuing GeoPackage will have the benefits of the protocol buffers-based Mapbox Vector Tiles and the flexibility of table-based attributes.

### 8.3.1. Implications

Participants have not taken a close look at the analytical operations that could potentially be supported by vector tiles. Beyond feature visualization and identification, there are many potential pitfalls. Feature information may be modified as you move up or down the tile pyramid for performance reasons. Features may be split across multiple tiles and geometries may be chopped up at tile boundaries. New techniques may be required to fully use this information to its full

potential.

This was identified as an area for future work.

### 8.3.2. Design Decisions

Two approaches were considered. One approach created an individual attributes table for each tile. This approach was rejected because it could potentially produce an extreme number of tables. In addition, features can be strewn across multiple tiles and this would make effective querying impossible.

This ER proposes using the Related Tables Extension (RTE) to manage the many-to-many relationship between the attributes and the tiles. This required adding a new requirements class since the draft RTE does not address vector tiles. At this time, it is not known what the performance characteristics of this approach are. This is a possibility for future work.

There is currently no agreed upon way to associate the different layers embedded within a single MVT tile set with their own individual attributes table. One approach that was considered was to add an 'attribute\_table\_name' column to the `gpkgext_vt_layers` table. This could be implemented as a change to the [Vector Tiles Attributes Extension](#).

## 8.4. Coordinate Reference Systems of Tiles

There is currently no mechanism for indicating the Spatial Reference System (SRS) of vector tiles. This is not an issue for MVT (the geometries are in pixel-space) or GeoJSON (which specified `urn:ogc:def:crs:OGC::CRS84`) but it could be for other tile types that were not explored during the Pilot. However, the existing GeoPackage tables are not a good candidate for storing this information. First of all, GeoPackage generally treat extents (min x, min y, max x, max y) and an SRS ID as a unit. After all, a coordinate has no meaning without an SRS defining it and forcing a client to go to a second table to retrieve the reference is somewhat of a nuisance. Second, there are two distinct sets of extents that GeoPackage clients care about. The first is the extents of the tile pyramid. This is what the client uses to calculate what tiles go where and is stored in `gpkg_tile_matrix_set`. The second is the extents of the content (tiles) in that pyramid. In the absence of something like OWS Context, clients conventionally use this to provide a default view of the tiles layer or to support a "zoom to extents" operation. After all, no one wants to see the whole world by default when the content is limited to a single city like Daraa, Syria. These (informative) extents are stored in `gpkg_contents`.

If the community determines that the tile SRS must be stored somewhere, it must be somewhere other than `gpkg_contents` or `gpkg_tile_matrix_set`.

## 8.5. Styling and Symbology

One of the requirements for this project is to portray the data stored in Vector Tiles-based GeoPackages using defined styles and symbols. GeoPackage does not currently have a standard encoding for style and symbol information but GeoPackage SWG members have identified two strong preferences for this capability. The first is to decouple the styles from the data and the second is to provide a tabular encoding for the styles. The Pilot participants have devised an

approach for styling and symbology that provides a viable but not necessarily optimal way ahead for this requirement.

### 8.5.1. Decoupling Data and Styles

Other formats such as KML have a tight coupling between the data and the style and this causes problems because the data cannot be used for more than one purpose without duplication. The GeoPackage SWG and OWS Context SWG have agreed in principle on a common approach for encoding OWS Contexts into a GeoPackage that will decouple this information. This is documented as [OWS Context Extension](#), which is a minor update to the proposed GeoPackage extension in [2]. This extension provides a way to encode maps (layers, their styles, and associated metadata) into a GeoPackage. This approach is being tested by OGC members for the first time as part of this Pilot.

### 8.5.2. Style Encoding

The second preference is for the style encoding to be tabular in form so that developers can do all of their work in the relational domain and not have to work with document parsing. However, there is not currently a conceptual model that can be implemented in a tabular way. (This is being explored as part of the OGC Testbed-14 project, but at the time of writing, that project was still in progress). For the purposes of this Pilot, the participants have agreed to use existing style encodings such as the Mapbox style encoding. The proposed approach uses the `content_type` column to define the encoding for the style. This approach is being tested by OGC members for the first time as part of this Pilot.

### 8.5.3. Cardinality of Styles

While OWS Context allows an offering to have multiple styles, the Pilot participants did not foresee a need for this. By eliminating this requirement, an offering can point to the stylesheet it uses and stylesheets can be reused by multiple offerings without a join table.

## 8.6. Well-Known Scale Sets

WMTS introduced the notion of well-known scale sets. However, this concept has not gained much traction in GeoPackage. Since each tile pyramid must be fully described in `gpkg_tile_matrix_set`, there is not a clear benefit to supporting well-known scale sets. A producer will have to populate `gpkg_tile_matrix_set` anyway and a client will have to be able to read that table anyway, so supporting well-known scale sets is a new function point with a dubious benefit.

## 8.7. Layer Extents

There is currently no mechanism for storing or retrieving the extents of an individual layer within a tileset that contains multiple layers. This could be useful. Layer extents are a thorny issue for the features option because they can be changed through edits and they are relatively expensive to calculate. (This is why the extents in `gpkg_contents` are informative.) With vector tiles, the data is not expected to be updated in real-time so having layer extents is reasonable. This would be a departure from how MBTiles handles layers.

# Appendix A: Revision History

Table 15. Revision History

<b>Date</b>	<b>Editor</b>	<b>Release</b>	<b>Primary clauses modified</b>	<b>Descriptions</b>
August 25, 2018	Jeff Yutzler	.1	all	initial version
August 31, 2018	Jeff Yutzler	.2	all	preliminary ER
September 28, 2018	Jeff Yutzler	.3	all	draft ER
October 31, 2018	Jeff Yutzler	.9	all	proposed final draft ER
November 8, 2018	Carl Reed	.91	all	final draft ER
December 23, 2018	Jeff Yutzler	.92	1	adding future work item

# Appendix B: Bibliography

1. Mapbox: Mapbox Vector Tile Specification - version 2.1, <https://www.mapbox.com/vector-tiles/specification/>, (2016).
2. Yutzler, J.: GeoPackage / OWS Context Harmonization Discussion Paper. OGC 18-037, Open Geospatial Consortium, [https://portal.opengeospatial.org/files/?artifact\\_id=79181&version=1](https://portal.opengeospatial.org/files/?artifact_id=79181&version=1) (2018).