OGC Vector Tiles Pilot

Summary Engineering Report

Table of Contents

Publication Date: 2019-02-15

Approval Date: 2018-12-13

Submission Date: 2018-11-21

Reference number of this document: OGC 18-086

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

Category: Public Engineering Report

Editor: Sam Meek

Title: OGC Vector Tiles Pilot: Summary Engineering Report

OGC Engineering Report

COPYRIGHT

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

This OGC Engineering Report (ER) provides the summary findings resulting from completion of the OGC Vector Tiles Pilot (VTP or Pilot). The requirements for the Pilot were generated from a combination of sponsor input and analysis of typical use cases for tiling of vector feature data across the OGC Standards Baseline and related standards. The driving use case for this activity was the visualization of feature data on a client. The three main scenarios considered were consumption of tiled feature data by a web client, a desktop client and a mobile client. As a standards body, the OGC already has standards that fit these use cases. These are; Web Map Tile Service 1.0 (WMTS) for a web client, and GeoPackage 1.2 for a mobile client. Web Feature Service (WFS) 3.0 is suitable for a desktop client and has an in-built method to support tiling, but not specifically for tiled feature data such as that explored in the VTP. One of the purposes of the Pilot was to produce demonstration implementations to support tiled feature data using WFS 3.0, WMTS 1.0 and GeoPackage 1.2 that can be validated by Technology Integration Experiments (TIEs). The draft extension to these standards helped define a draft Conceptual Model for tiled feature data in support of visualization. The Conceptual Model formally captures the requirements for component implementations and rationalizes them into a model documented in the Unified Modeling Language (UML).

The ER provides an overview of each of the components, their implementation decisions and the challenges faced. The components are presented as *draft extensions* to existing standards. The WFS standard is currently in a major revision cycle and is transitioning away from services to a resource-oriented architecture. This transition has implications for access to tiled feature data. This offers options of access to pre-rendered tiles, or to tiles created using WFS 3.0 query functionality. The current WMTS standard only offers access to the pre-rendered tiles and much of the work is therefore about defining and supporting tiled feature data as a media type. The OGC GeoPackage standard is more complex as it attempts to ship all of the tiled feature data in a self-contained package aimed at environments that have Denied, Degraded, Intermittent or Limited (DDIL) bandwidth. DDIL is an important use case for GeoPackage as most normal web services do not function without connectivity. The military, first responders and other groups who work in challenging operational environments require a capability to ship, store and distribute geospatial data in an efficient, modern manner. The combination of GeoPackage and tiled feature data offers the means to supply detailed geospatial data in a portable fashion to satisfy many DDIL use cases. GeoPackage also offers the majority of the future work as it attempts to store information such as styling and attribution separately to the geometries to take advantage of a *relational database* structure.

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 than the colloquial term of "vector tiles". This engineering report therefore interchangeably uses both "tiled feature data" and "vector tiles" to refer to the approach of tiling vector feature data.

1.1. Requirements & Research Motivation

The requirements for the Pilot were to address the issues identified from previous OGC testbeds and propose draft standards to conceptualize tiled feature data and extend WFS, WMTS and GeoPackage to support tiled feature data based on the Mapbox Vector Tile (MVT) specification [1].

This ER directly addresses requirement D011 from the VTP Call For Participation (CFP). The requirement states that: “D011: Summary Engineering Report - A report that summarizes the initiative including outputs, lessons learned and recommendations”.

1.2. Prior-After Comparison

Prior to the pilot, vector tiles in the OGC was utilized in testbeds, but there was not an overall, agreed upon framework to implement against. The Pilot has now produced a Draft Standard of the Tiled Feature Data Conceptual Model [2] and tiling extensions for WMTS [3], WFS [4] and GeoPackage [5] as well as this summary report that grounds the findings in an on-going OGC context.

1.3. Recommendations for Future Work

Although comprehensive, requirements for future work have been generated from this Pilot. These recommendations are to:

- Address the outstanding technical issues with tiled feature data including:
  - Agree on a way to request primitive geometry types via a layer listing.
  - Implement methods to remove unwanted polygon edges, either through an extra border or artificial segments.
  - Disambiguate the meaning of the *ID* field in Mapbox Vector Tiles to enable follow through from original features to tiled features.
  - Support a variety of coordinate reference systems.
- Understand and address the issues with styling as there is currently no agreed method of completing this task.
- Investigate the possibility of tiling other data among OGC standards. It has been noted that an architecture such as that adopted by WFS 3.0 could support converged services. A name suggested for this service is Web Object Service (WOS).

1.4. Document contributor contact points

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

Contacts

Name	Organization
Sam Meek	Helyx SIS
Eleanor Ogden	Helyx SIS
Andrea Aime	GeoSolutions
Jerome St. Louis	Ecere

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.

- OGC: OGC 12-080r2, OGC OWS Context Conceptual Model 1.0 Standard, 2014 [https://portal.opengeospatial.org/files/?artifact_id=55182]
- OGC: OGC 12-128r15, OGC GeoPackage 1.2.1 Standard, 2018 [https://www.geopackage.org/spec120/index.html]
- OGC: OGC 07-057r7, OGC® OpenGIS Web Map Tile Service Implementation Standard, 2010 [http://portal.opengeospatial.org/files/?artifact_id=35326]
- OGC: OGC 17-069, OGC® Web Feature Service 3.0: Part 1 - Core Candidate Standard, 2018 [https://rawgit.com/opengeospatial/WFS_FES/master/docs/17-069.html]

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] shall apply. In addition, the following terms and definitions apply.

- Tile

  ```
  A tessellated representation of geographic data, often part of a set of such
  elements, covering a spatially contiguous extent which can be uniquely defined by a
  pair of indices for the column and row along with an identifier for the tile matrix
  (adapted from OGC 07-057r7)
  ```

- Tile Set

  ```
  a definition of how tiles are organized. It contains a definition of the geographic
  extent and geographic location as well as a coordinate reference system
  ```

3.1. Abbreviated terms

- OGC Open Geospatial Consortium
- WFS Web Feature Service
- WMS Web Map Service
- WMTS Web Map Tile Service
- VT Vector Tiles, Vector Tiling, Vectiles

Chapter 4. Overview

This document summarizes and presents the findings from the OGC Vector Tiles Pilot. The findings from the demonstrations and accompanying ERs within this report seek to act as draft OGC standards and start to formalize the process of using tiled feature data in OGC.

This document is organized as follows:

Section 5 provides an introduction to Vector Tiling and the OGC Pilot programs.

Section 6 gives an overview of the Vector Tiles Pilot including strategies, advantages and challenges.

Section 7 presents the abstract requirements for the Pilot, the proposed architecture and a description of the components that form the demonstration aspect of the Pilot.

Section 8 is an overview of the Conceptual Model for Tiled Feature Data to ground the implementation decisions taken.

Section 9 provides a summary of each of the components in turn.

Section 10 summarizes and rationalizes discussions for the Pilot from the component implementers.

Section 11 lists the Pilot recommendations.

Section 12 concludes the document.

Chapter 5. Introduction

This ER is the summary document for the OGC VTP endeavor that was completed towards the end of the year 2018. This ER has several objectives that include the following:

- Report on the work carried out in Phase 1 and Phase 2 of the Pilot.
- Provide an architectural overview of the Pilot.
- Present the findings of the interoperability experiments.
- Collate and rationalize the recommendations from the other ERs in the Pilot and provide further overarching recommendations.
- Present ideas for work to be carried out in future OGC Innovation Program projects or independently by the Sponsors.
- Conclude and provide an overall, distributable executive summary of the Pilot.

Much of the work on Vector Tiles has been done through OGC Testbed initiatives, Testbed-13 is a notable example where many of the foundation documents for this Pilot were created. This ER contains a review of each of these documents and provides a concise section detailing the motivating principles.

5.1. OGC Pilots and other initiatives

The OGC has three main organizational programs within its remit; the Standards Program, the Outreach and Communication Program and the Innovation Program. The Innovation Program defines several initiative types that enable the evolution or definition of draft standards through practical applications at increasing Technology Readiness Levels (TRLs). Figure 1 shows the initiatives within the innovation program, these are briefly described as follows:

- Testbed - several months long with proof of concept demonstrated with low-to-mid TRL components and Technology Integration Experiments (TIE), components are usually supported by an ER.
- Interoperability Experiment - IEs contain slightly higher levels of TRL components, unlike Testbeds, the experiments are facilitated, not led by the OGC.
- Pilots - A pilot is often a rapid turnaround mid-to-higher TRL experiment containing a few participants set out to work with a small group of technology components set around a real-world common theme.
- Plugfests - these are very short time-frame, COTS based interoperability experiments designed to validate interoperability between components implementing the same standards.
- Operational systems - these are outside of the OGC initiatives and are real-world components working in an operational capacity.

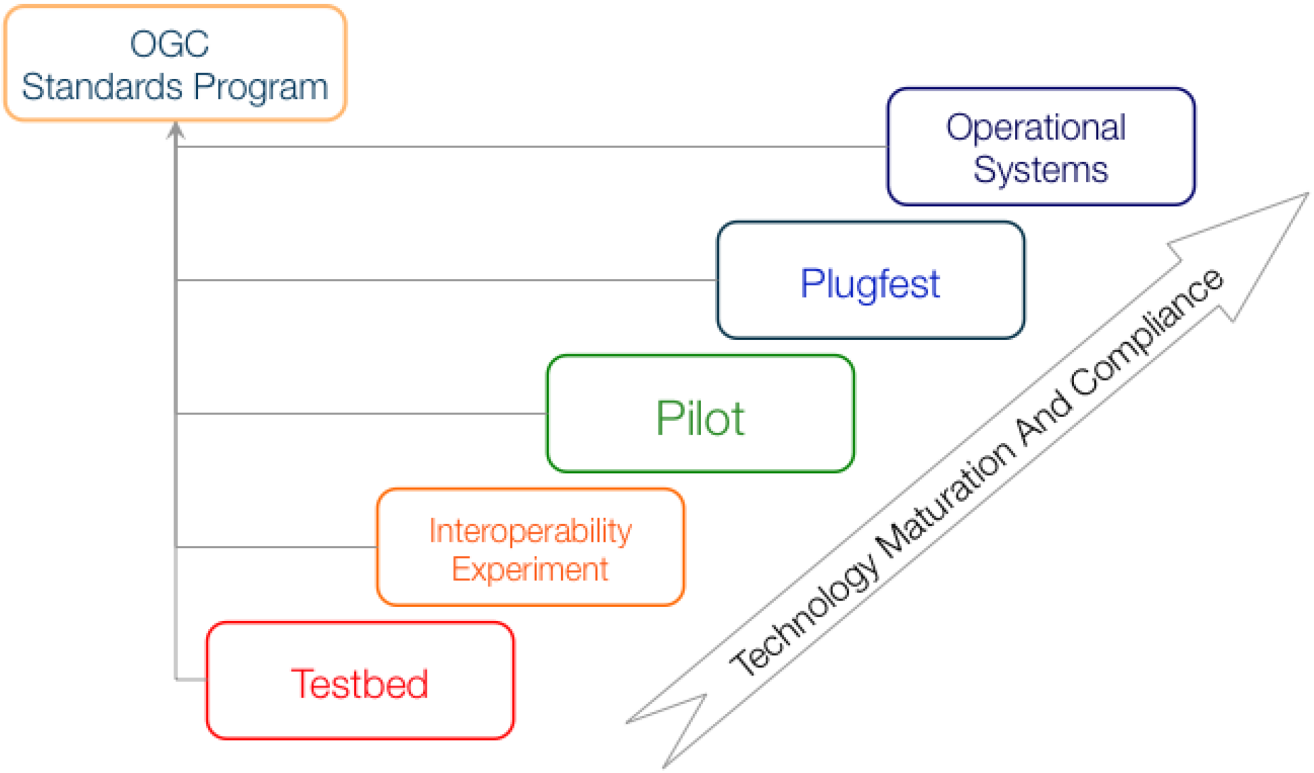

Figure 1. OGC Innovation Program activities.

The OGC also runs a number of hackathons each year. The hackathons are organized to bring groups of developers together, for a number of days, to focus on a specific problem. Hackathons also take place formally or informally during OGC testbeds and other initiatives as a delivery framework to rapidly produce results.

As mentioned previously, the work described in this ER is a summary of an OGC Pilot, therefore success criteria is based upon utilizing COTS components in an operational context.

Chapter 6. Pilot overview

The purpose of the Vector Tiles Pilot (VTP) initiative was to continue and conclude the formalization process of using tiled feature data in the OGC via production of a set of demonstration components and corresponding ERs presented as draft OGC standards or extensions to existing standards. There are several versions of tiling approaches that are mentioned in the following sections. The tiling strategy for the VTP utilizes the Mapbox Vector Tile (MVT) specification [1]. MVT uses Google Protocol Buffers for encoding content as well as some other common vector tiles formats including GeoJSON. This Pilot is the next phase in exploration of tiled feature data within the OGC and builds on work from previous OGC Testbeds.

6.1. Background and Motivating work

This section covers some of the previous work done in OGC testbeds via a short literature review with the aim of providing an overview to ground the work done in this Pilot. The main documents of interest are:

- OGC Testbed-12
  - 16-067r4 Vector Tiling Implementation ER [6]
  - 16-068r4 Vector Tiling ER [7]
- OGC Testbed-13
  - 17-041 Vector Tiles ER [8]

Vector Tiles were originally discussed in Testbed-12 and documented in the Vector Tiling ER. That document characterized tiles in terms of their definition, use cases, utility among Geographic Information System (GIS) users and route to adoption within the OGC. At that time, there was no agreed method of tiling vector data. Tiling of raster data is accomplished (broadly) by associating a pixel with a geographic area and a geographic area with a specific tile in a 1:1 relationship for a single scale. However, vector features can vary and agreed approaches have to be adopted to successfully allocate a vector feature to a particular vector tile. Additionally, raster tiles are generally scale appropriate. The approach with vector data is different, as it is likely to undergo a transformation through generalization to produce scale appropriate tiles. Styling of vector tiles is also done differently to raster equivalents. Raster tiles are made up of simple images that generally have their styling 'baked in', whereas vector data can be styled on-the-fly using a variety of techniques. Styling approaches in vector tiles offer both overhead and opportunity to produce audience appropriate tiling.

The Testbed-12 Vector Tiling ER companion is the Vector Tiling Implementation ER that documents the implementation of vector tiles in a GeoPackage format. Prior to the Vector Tiling ER, GeoPackage supported raster tiling procedures with excellent reported results. The goal in Testbed-12 was to prove the approach for vector tiles in GeoPackage to replicate the results seen with raster tiles. Tiled feature data in GeoPackage was implemented using a *vector tile pyramid concept*, where vector tiles are stored in a pyramid, much in the same way raster tiles are. There are two approaches that were considered for storing tiled data in pyramids; render based tiling and feature based tiling. Render-based tiling changes the original geometry which was considered unacceptable by the sponsors, therefore a feature based tiling approach was used with a GeoJSON encoding.

The work done on vector tiles in Testbed-13 was a comprehensive study of approaches to tiling vector data including:

- A study to evaluate the feasibility of a standardized model.
- A study on projections and moving features to assess a set of defined projections for use in tiled data and moving feature data.
- A generalized approach to styling and symbology.
- Approaches to associating attributes with features.
- Geometric considerations in tiling services.
- The implications for vector tiles in low-bandwidth environments.

The stated goal of the Testbed-13 work on vector tiles is to move closer to standardization by producing a set of draft standards and implementations to demonstrate and test the draft standards, this VTP can be seen as a continuation of the work done in Testbed-13.

6.2. Tiling

Tiling as a method of data delivery has been around since the 1960s, although in many instances, the focus was on raster-based products such as imagery. There are several terms associated with tiling that should be considered:

- Tessellation - Tiling of a plane using one or more geometric shapes with no overlaps or gaps. Tessellation can be performed with the following shapes:
  - Squares
  - Octagons
  - Equilateral triangles
  - Hexagons
- Tile - a tessellated representation of geographic data, often part of a set of such elements, covering a spatially contiguous extent.
- Tiling - a process of creating tiles.
- Tile set - a definition of how tiles are organized, with the following requirements:
  - Coordinate reference system (CRS).
  - Unit of Measurement, this may or may not form part of the CRS.
  - Extent, the area covered by the tile set.
  - Identifier.
  - Concept.
  - Scheme.
  - Origin.

6.2.1. Tiled Feature Data

Tiled feature datasets, sometimes referred to as *Vector Tiling*, *tiled vector* or rarely, *vectile*, are a product of splitting up vector feature datasets into discrete units that can be requested from a server and delivered to a client. A comparable approach is using raster data and image mosaics, where images are split up into multiple tiles of the same dimensions to make transport across networks more efficient. Raster tiling approaches were developed largely as a result of the explosion of map services being used across the Internet for mass market consumption.

Since the mid 1960s, spatially partitioned (tiled) data stores have been in use. Different names may have been used but in all cases the following deployed and widely used systems employed vector tiling as a storage model as well as (in some cases) a mechanism for enhancing rendering performance. For example:

- Canada Geographic Information System (CGIS). Deployed in the mid 1960s and used Freeman encoding of coordinates which in many ways is similar to the MVT coordinate encoding rules. It also implemented Morton Matrices and quad-trees.
- Wetlands Analytical Mapping System (WAMS). Deployed in the late 1970s and widely used in the Department of the Interior (DoI) and later known as the Army Map Service (AMS).
- Map Overlay and Statistical System (MOSS). Deployed in the late 1970s and widely used in the DoI.
- GenaMap: Deployed in the mid 1980s. Very sophisticated and technically advanced tiled data store for both topologically structured vector data and 2D/3D raster data. Used RTrees, caching, and many technologies that provided good performance even by today's standards.

These early examples had all processing done on the server, however much of the utilization of tiled data through geographic techniques now takes place on the client side. Since the 1990s, the advent and proliferation of Object-Oriented languages, such as Java, provided client access to GIS functionality such as rendering control and simple analytics.

There are many use cases for tiled feature data, both inside and outside of the OGC. Therefore tiled feature data for visualization was deemed worthy of an OGC Pilot initiative to help better understand the requirements for tiling across the OGC. Some overall, high-level use cases include; adaptation of tiled feature data to a small screen, such as on a mobile device, storage in partitioned environments and importantly for the sponsors, dissemination across degraded or low-bandwidth networks. Due to the existence of several mature and supported OGC standards, there are obvious candidates that could be extended to support these broad use cases.

Tiled feature data approaches can be designed to repeat the approach seen in the raster equivalents, except that the underlying data retains all of its attribution information. For gridded data the tiling service typically splits up large datasets, provides the user with the required tiles, and rationalizes the projection information to ensure that the tiles are reproduced in the client in the correct order and in the correct location. One of the objectives of this Pilot is to provide the same functional capabilities for vector data, except there are additional complications of correctly reproducing contained data client-side including vector specific aspects such as topological rules for contained features. Additionally, raster tiling is largely pre-rendered and often cached on the server-side which may be unfeasible in use cases demanding tiled feature data.

There are a number of OGC standards and other specifications that provide some level of tiled data including:

- Mapbox Vector Tiles - the VT specification of choice for the VTP.
- GeoJSON Vector Tiles - of secondary importance but included for interoperability purposes.
- Cesium 3D Tiles.
- Esri 13S.
- OGC CDB.
- GenaMap
- Google & Apple Maps
- Luciad

These tiling approaches contain a mixture of open and proprietary specifications, with some of them submitted to the OGC as *Community Standards*.

For this Pilot, the *Mapbox Vector Tile* (MVT) specification was selected by the Pilot sponsors for focus, and has been addressed in several prior Testbeds and Pilot initiatives from the OGC. One of the objectives of this Pilot is to formalize the use of vector tiles within the OGC through several well-defined standard interfaces. An advantage of MVT is that it supports Google Protocol Buffers which results in efficient encoding and delivery of the tiling packages. Other tiling approaches, e.g. GeoJSON vector tiles, use different encodings and are included in the VTP due to their wide adoption for interoperability purposes.

6.3. Advantages of Vector Tiles

The main advantage of vector tiles is the speed at which data can be delivered from a server to a client. Much of this speed increase over traditional methods is achieved through efficient delivery of areas of interest requested by the client. As mentioned in the review of previous work, vector tiles have several advantages over existing vector distribution technologies. The push for vector tiles has come about in part because of the explosion of geographic data available to users. This is often termed *Big Data* and does not simply refer to data volumes, but also velocity and veracity. One also has to consider technologies and techniques devised specifically to take advantage of the wealth of data available. An often cited data source that may be considered geographic *Big Data* is OpenStreetMap (OSM) [1: http://openstreetmap.org]. OSM has around 30GB of data available for the entire world and is a community driven project that anyone can contribute information to. Distribution of this data is problematic. Therefore strategies such as change-synchronization are often adopted to keep a dataset up to date server-side. Distribution of such data to clients can be challenging due to the data size. Therefore tiles are often used to provide the user with the smallest volume for their use case.

Vector tiles offer additional advantages through decoupling of peripheral aspects such as styling. Unlike most raster tiles that are pre-generated and styled according to a single schema, vector tiles have a single base dataset of features and can be styled upon request and on-the-fly. This enables feature data to be used for a multitude of clients. Client-side styling is also possible if suitable. Styling can refer to colors as well as other aspects such as point symbol size, coloring and shading.

6.4. Challenges with Vector Tiles

Although very efficient, vector tiles have had issues with cross-tile alignment of features. If cross-tile features are not aligned, then there are downstream errors in the results of topological operations such as routing, as the features are no-longer connected. Additionally, cartographic convention is broken when producing products as the output is unsightly and may dissuade audiences from using a service.

The Testbed-12 Vector Tiles ER outlined several challenges with vector tiles, these were:

- **Data Coherence:** This refers to transferring back and forth between raw features and occurs either around the edges of tiles or with particularly long or large features that cross multiple tiles. In some use cases, there is a requirement to rebuild original features from tiles. This can be accomplished by requiring vector tile data structures to carry all of the information required to rebuild a feature from a tile set.
- **Defining Multiple Levels of Detail:** As mentioned previously, provision of multiple levels of detail in raster tiles is accomplished either through usage of different imagery sets or through interpolation. Vector Tiles can be provided at multiple scales using two methods (alone or in combination):
    - Filtering - where features are omitted at different scales.
    - Generalization - where feature geometries have detail removed at smaller scales.
    - Curation** - assignment of features to levels of detail dependent upon data fidelity
- **Tile Sectioning:** There is no standard way of assigning a feature to a tile, but there are multiple approaches. One is to associate a feature with the tile that it crosses most, but this can be inefficient by requiring clients to download more data than they required. Other approaches rely on splitting features and assigning them to a tile, however there are many approaches to doing this as well.
- **Unique Feature Identification:** Through tiling operations mentioned previously, producing tiled data with unique feature identifiers is a challenge, as is linking the features back to the original features.

In addition to these, other challenges identified since include:

- **Coupling of tiled elements**. There are different schools of thought on how to store and manipulate the elements of tiled data. The main elements of interest are the geometries, the attributes and the styling. These can either be tightly coupled or loosely coupled and this will vary depending on the implementing service and the *type* of tile being returned (pre-rendered or feature based).

Chapter 7. Requirements analysis

The requirements for tiled data are based upon the three key clients for vector data, namely mobile, web and desktop and client access by the OGC standards: GeoPackage 1.2, WMTS 1.0 and WFS 3.0. The requirements for each of these types have many commonalities that are summarized in the following list:

- **Coordinate reference system** independence.
- Several methods of providing **bounding box** description consistent with OWS common.
- Provision of a **tile scheme**, i.e. the format and size of the tile.
- Flexible relationships between **layers** and tiles to support existing Vector Tiles and OGC specifications.
- Identification of **format** without prescribing a specific encoding.
- Support for **geometries** beyond the *simple feature model*.
- **Attribution** recording methods consistent across OGC standards.
- Enabling encoding of arbitrary **metadata**.
- **Tile addressing**
- Support for **3D data**.
- De-coupled **styling**.
- Consideration of access to **rendered and feature** based data.

These requirements are abstracted from specific use cases or implementations and therefore form the basis for developing a conceptual model, which in turn feeds into component deliverables. As the above list is abstract, the requirements for each individual component are likely to be a subset. For example, WMTS only supports *rendered* data and therefore should only be utilized in a use case that requires tiled data.

7.1. Pilot architecture and deliverables

The Pilot was designed to be completed within a short time frame and utilized a phased approach, Phase 1 consisted of delivery of draft components and ERs and Phase 2 required delivery of final components and ERs. The overall goal of the Pilot was to define and test approaches for vector tiles extensions to existing OGC standards. This was done through profiling and providing extensions to the existing OGC standards. This section contains a short description of each of the delivered components, ERs and an overall architecture.

The architecture of the pilot was designed to cover the three main server client relationships identified above, shown in Figure 2, these include:

- Desktop Client → WFS 3.0
- Web Client → WMTS 1.0
- Mobile Client → GeoPackage 1.2

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

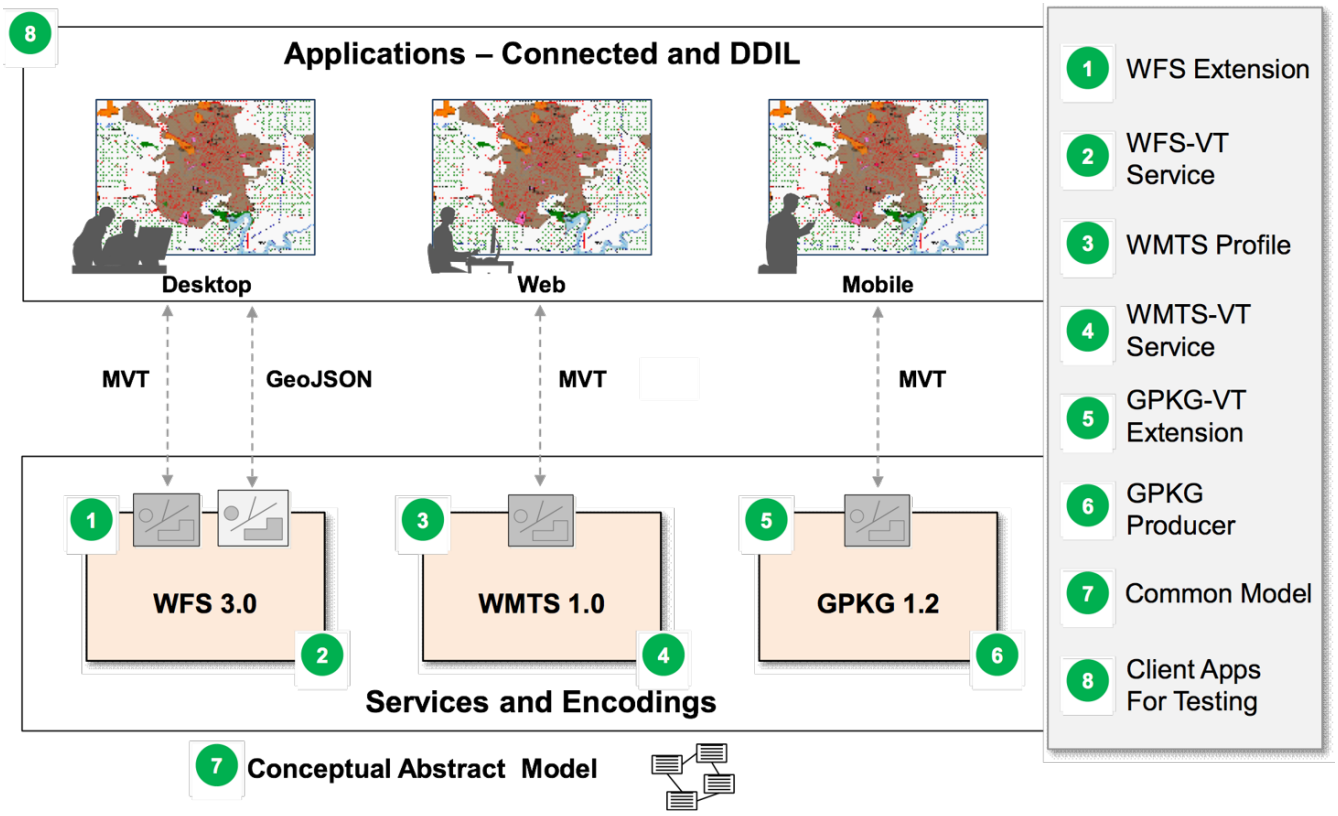

Figure 2. Vector Tiles Pilot Architecture

7.1.1. Engineering Reports

ERs perform several functions in OGC Pilots including:

- Documentation of implementations.
- Definitions of new draft standards.
- Communication of change requests.
- Call for future work.

Generally, the results of a component implementation are accompanied by an ER that captures the results.

WFS 3.0 Vector Tiles Extension ER

The draft WFS 3.0 standard is a complete re-working of previous WFS versions. Instead of working using a typical set of OGC compliant web services calls, WFS 3.0 has moved to a Representational State Transfer (REST) based architecture using OpenAPI (previously known as Swagger). As such WFS 3 is not as dependent on XML as an encoding. The purpose of this ER is to document a vector tiles extension for WFS 3.0 in order to support tiling in a recognized and standardized manner [4].

GeoPackage 1.2 Vector Tiles Extension ER

This ER describes a vector tiling extension for the GeoPackage Standard to support tiling [5]. GeoPackage is the SQLite container and the GeoPackage Encoding Standard governs the rules and requirements of content stored in a GeoPackage container. Therefore the extension consists of instructions on how to correctly configure the database.

WMTS Vector Tiles Extension ER

The WMTS Vector Tiles Extension ER formalizes the change request for vector tiles support in WMTS [2: http://ogc.standardstracker.org/show_request.cgi?id=517] and reports on the implementation of the WMTS server component [3].

Tiled Feature Data Conceptual Model ER

The outputs of this work item are presented as an ER outlining the tiling conceptual model that should be considered by the other ERs in this Pilot project [2]. The conceptual model is an abstract, technology independent construct that guides implementation.

Pilot Summary ER

This document.

7.1.2. Server Components

Server components consist of implementations that produce vector tiles via the following interfaces or formats:

- Web Feature Service 3.0 (WFS) - produces vector data where individual attributes can be queried.
- Web Map Tile Service 1.0 (WMTS) - produces tiles, traditionally as images.
- GeoPackage (GPKG) 1.2 - a self contained SQLite container.

WFS 3.0 Tile server

This work item was essentially an implementation of the work documented in the WFS 3.0 Extension ER. Utilizing the recommendations from OGC Testbed-13, the component was required to support MVT as an output encoding using Google Protocol Buffers. The component was also required to offer GeoJSON as an output. These were mandatory encodings, but it was also expected that the component supports other formats as well.

WMTS Tile server

Web Map Tile Service (WMTS) is an OGC standard that complements the existing Web Map Service (WMS) and WFS. WMTS has the ability to support vector tiles through its protocol, but does not yet have the ability to request a tiling *format*. The Change Request [3: http://ogc.standardstracker.org/show_request.cgi?id=517] outlines the requirements for support of vector tiles as an extension to WMTS. Essentially the requirement was to include a variety of internet media types, including as a minimum MVT and GeoJSON.

GeoPackage 1.2 Producer

As with the WFS and WMTS component deliverables, a working example was needed to fulfill the requirements of the sponsors. This component was to be TIE tested to ensure the adopted approach is viable. Essentially the service was to enable clients to request vector tiles as GeoPackages from a service with the correct media type, schema and encoding parameters.

7.1.3. Client components

Clients were required to perform the TIE experiments that conclude testing in the pilot. Clients could consist of simple web pages for querying and retrieving data to plugins for existing pieces of software such as QGIS or ArcGIS. Each of the server components outlined in the previous section required a client to perform TIE, these are described in the relevant server component reports.

Chapter 8. Tiled Feature Data Conceptual Model

The Tiled Feature Data Conceptual Model fulfills the tiling requirements across implementations and provides the abstract specification for vector tiles. The model is based upon a cascading set of concepts:

- Feature.
- Layer.
- Tile.
- TileSet.

This model represents a Feature as an individual vector element that has properties associated with it, a Layer as a collection of features for a particular scale that includes information about styling. The model represents a Tile as a collection of features with an extent that is organized according to the Layer and a TileSet as a collection of Tiles. Note that this representation of a TileSet covers what has historically been referred to as a TileMatrix or a Tiling Scheme. This simple model enables simple storage of the data required to build up a set of vector tiles from a set of features. The styling aspect held in the Layer class enables simple interchangeable styling options which are decoupled from the data held in the tiles.

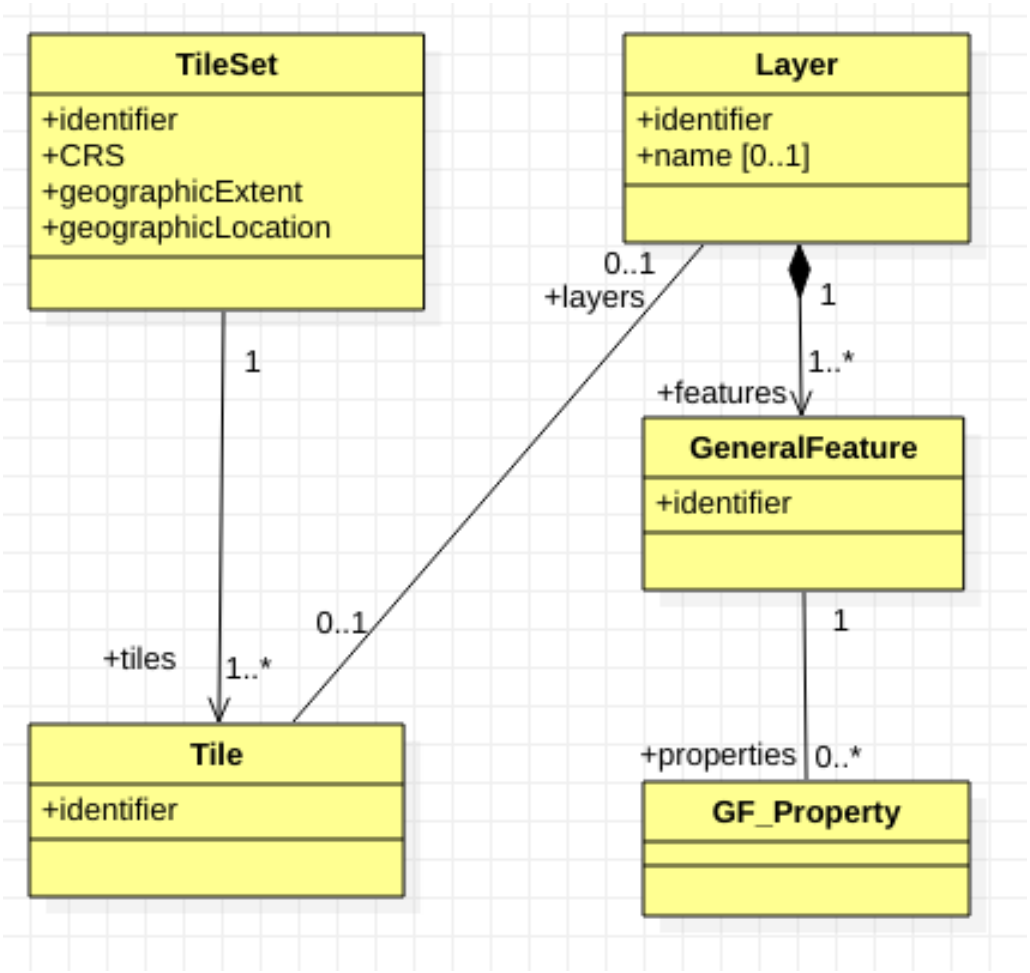

Figure 3. The Vector Tiles Conceptual Model

Vector Tiles do not specify an encoding, however Mapbox vector tiles utilize Google Protocol buffers to encode the tiles, this forms part of the conceptual model. Additionally, from the requirements, Google Protocol buffers are used to encode the vector tiles as they can include arbitrary metadata relating to the vector tiles.

Chapter 9. Summary of components

This section outlines each of the components delivered as part of this Pilot.

9.1. Vector Tile API for WFS 3.0

Web Feature Service 3.0 is revolutionary rather than evolutionary as it does not start with the same assumptions as previous versions of WFS. Instead of building upon the previous *service based* interface, the standard has migrated to a *REST* interface, a move that may be followed across the OGC suite of standards where appropriate. WFS 3.0 supports the same *operations* as the previous version, except they are now exposed as an Application Programming Interface (API) based on the OpenAPI (https://www.openapis.org/) specification.

The objective of the Vector Tiles Pilot for this component is to extend the WFS 3.0 operations to support vector tiles, notably the emphasis is on *read access* therefore the only required supported HTTP method is HTTP GET, however it is noted that servers with VT enabled WFS3.0 should also independently support HEAD and OPTIONS for each resource.

There are two types of *resources* that are supported by the WFS 3.0 Server, these are *direct access to tiles* and *access to features within the tiles*. The two different resources have different approaches and calls associated with them. The interface is based entirely upon **http verbs**, which is a major shift from utilizing service calls such as *GetCapabilities* and *DescribeFeature*. The http verbs used are:

- GET
- POST
- DELETE
- HEAD
- PUT

The resources exposed via WFS 3.0 are accessed via REST and the HTTP verbs mentioned. Architecturally, WFS 3.0 is not implicitly bound by a single interface to a single server, which was the assumption in previous versions of WFS. Instead, resources are fronted via an API and the actual mechanics of storing the resources at rest are left to the implementer to choose the type of infrastructure performing the work behind the scenes.

There are two approaches to offering vector tiles in the WFS 3.0 extension:

- Access to tiles - this approach offers pre-generated, regularly gridded tiles arranged into a pyramid of tile matrices at fixed resolutions. Direct access to these pre-generated tile sets is offered via a simple Uniform Resource Locator (URL).
- Access to features - this approach leverages the querying capabilities of WFS 3.0 to generate a set of tiles on-the-fly as a *response* like any other media type.

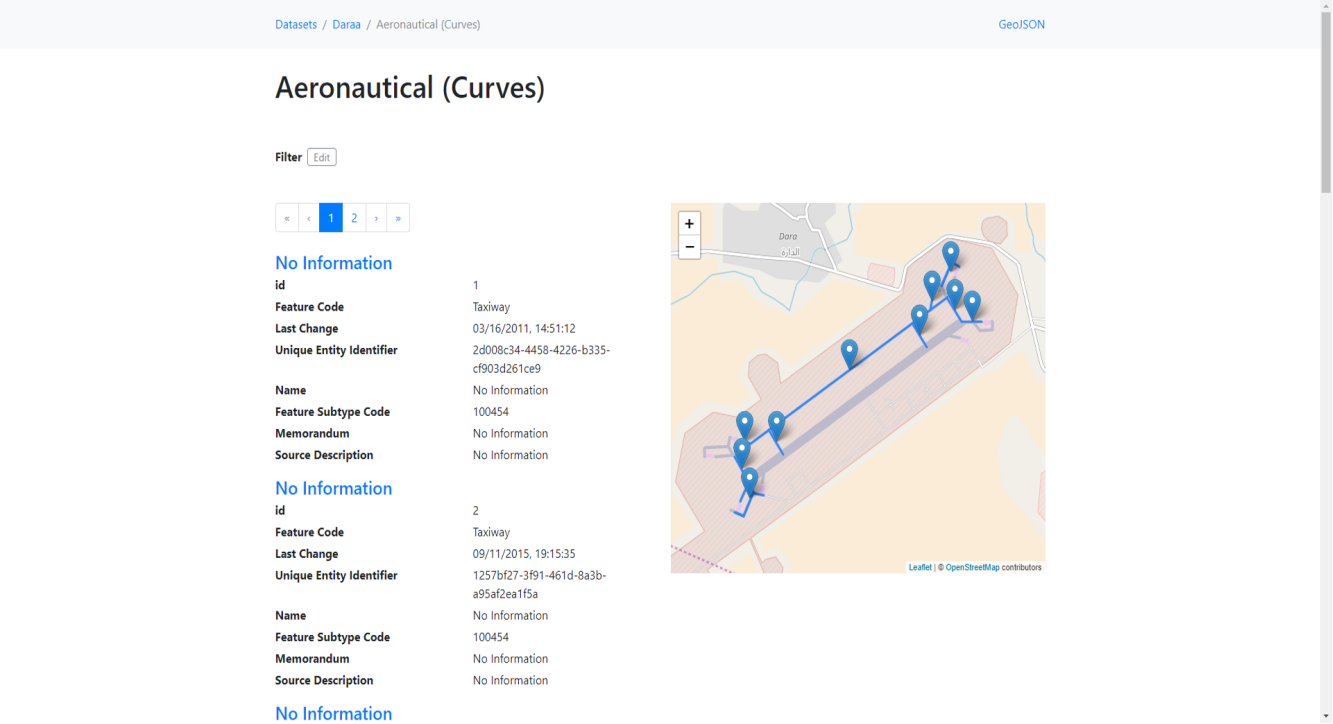

Figure 4. The WFS 3.0 server with the Vector Tiles extension.

9.1.1. Access to tiles

Access to vector tiles directly requires extension to the existing WFS 3.0 specification and contains the following paths:

- /tilingSchemes - provides high-level metadata about tiling schemes, enables multiple collections to refer to a single tiling scheme.
- /tilingSchemes/{tilingSchemeId} - provides metadata on a single tiling scheme.
- /collections/{collectionId}/tiles/{tilingSchemeId}/{level}/{row}/{col} - allows direct access to individual tiles per collection. The order of the location and level is determined by observed practice.
- /tiles/{tilingSchemeId}/{level}/{row}/{col} - access to individual tiles with one layer per collection.

This is a simple extension designed to support an existing tile set with read-only access, as it is returning pre-generated vector tiles. It somewhat mirrors WMTS in its read-only approach and recreates a lot of the functionality.

9.1.2. Access to features

Feature access in the broad sense is the objective of WFS, therefore feature access in WFS 3.0 for vector tiles is enabled by advertising support via the capabilities call. An extension proposed for vector tiles in core WFS 3.0 is Selection of *attribution*, WFS 3.0 already supports attribution, therefore the only addition required is the selection aspect. Tile sizes are requested according to the bounding box and resolution of the target area, fixed tile sizes are more suited for direct access to tiles. These aspects are called through the following endpoints:

- .../collections/{collectionId}/items?...&propertyNames=A,C,D,F&...

- …/collections/{collectionId}/tiles/{tilingSchemeId}/{level}/{row}/{col}?propertyNames=A,C,D,F

Calling of Mapbox vector tiles as a media type can be done either by using *"MVT"* or *"application/vnd.mapbox-vector-tile*".

Unlike the *access to tiles* method this method supports querying using core WFS 3.0 capabilities. For reference, the following table describes the filtering parameters that may be implemented to support vector tiles querying:

Table 1. Query Parameters for Accessing Mapbox Vector Tile via the Feature Access Path

Parameter Name	**Description**
bbox	Only features that have a spatial geometry that intersects the bounding box shall be included in the MVT response.
time	Only features that have a temporal geometry that intersects the timestamp or time period shall be included in the MVT response.
{prop}	Only features that satisfy the specified property filter shall be included in the MVT response.
Sizing parameters	
resolution/zoomlevel/size	A parameter that, perhaps in combination with the bbox parameters, sets the size of the MVT.
CRS parameters	
bbox-crs	Asserts the CRS used to specify the bounding box CRS.
crs	Asserts a specific WFS-supported CRS transformation to be applied to the compatible geometries of the features to be included in the MVT response.
Tile attribution	
properties	Comma-separated list of feature properties to include in the MVT response.

Requesting MVT as an output is enabled simply by advertising MVT as a capability in the OpenAPI/Capabilities document, this is allowed on the client side by setting the *Accept* header in the request to the required media type.

9.1.3. WFS 3.0 servers

The pilot included WFS services from a range of vendors as described below.

GeoSolutions

GeoSolutions implemented the WFS3 tiling extension as part of the open WFS 3.0 community module.

- Implementation of "direct tile access path API": Uses Mapbox Vector Tiles and supports both GlobalCRS84Geometric and GoogleMapsCompatible tiling schemes. Results are generalized

based on current zoom and clipped by a small tile buffer. The outputs are generated on-the-fly and all attribute information is included in the tile.

- Implementation of "feature access path API": Allows encoding of Mapbox vector tiles out of the /collections/{collectionId}/items resource using the MVT format.

Interactive Instruments

The interactive instruments WFS 3.0 server can be found here: https://services.interactive-instruments.de/vtp. Two simple OpenLayers clients were created for testing; http://portele.de/daraa.html and http://portele.de/iraq-syria.html.

- Implementation of "direct tile access path API": Can use either Mapbox Vector Tile or GeoJSON outputs, noting that the MVT data is clipped by a small tile buffer unlike the GeoJSON where no clipping occurs. This implementation supports only the GoogleMapsCompatible tiling scheme in WMTS standard.

Mapbox

The Mapbox WFS 3.0 server implementation can be found here: http://ec2-54-88-75-105.compute-1.amazonaws.com/ running on an Amazon Web Service (AWS) and using a Relational Database Service (RDS). Open source libraries tilelive and tilelive-postgis are used to create and send tiles to the client. A user can generate a specific tile and have it downloaded to their machine by putting in the appropriate url with the {z}/{x}/{y}.

Ecere

Ecere implemented a WFS 3.0 service for this pilot, which can be found here: http://maps.ecere.com/hms . The implementation focused on the direct access path (tiles), but also implemented some aspects of the feature access path (items). The service can also provide vector features and tiles using a WFS 1.1 interface, currently with the understanding that both BBOX and CLIPBOX will result in geometry getting clipped. Tiles for any layers are generated on-the-fly for any supported encoding and tiling scheme. Support for Mapbox vector tiles was added during the pilot. To demonstrate the value of harmonizing map services, imagery and coverage layers are also served using the same API from the same endpoint. Testing was done using the vtvalidate [https://github.com/mapbox/vtvalidate] tool, which validates consistency with Mapbox vector tiles specifications; with OGR software; with QGIS; as well as with the Ecere, Mapbox and GIS-FCU clients.

9.1.4. WFS 3.0 clients

There are several WFS 3.0 clients from a variety of vendors.

GIS FCU

The GIS FCU WFS 3.0 client is a web application using OpenLayers v5.1.3 JS library. The client can use both Direct Access Path and Feature Access Path URL templates to access MVT or GeoJSON. The GIS FCU implementation demonstrates the MVT format against the GeoJSON Vector Tile format, using rendering based on the National Geospatial-Intelligence Agency (NGA) Digital Printing Specification, it can be found here: https://map.gis.tw/pilot/wfs.html

Mapbox

The Mapbox WFS 3.0 client simply uses HTML to render tiles taken from the Mapbox server, it can be found here: http://bl.ocks.org/briandaviddavidson/c81b06997c6bb086ed2e6152ef998fab. The technology behind the client is Mapbox-gl.js that performs the rendering on the server. Styling is performed using *paint* within the Mapbox-gl library.

GeoSolutions

Two GeoSolutions WFS 3.0 clients were built based on OpenLayers:

- Daraa map, showing a comparison between tiles provided by GeoServer in different tiling schemas and data from various server implementations, found here: http://vtp2018.s3-eu-west-1.amazonaws.com/wfs.html. Maps shown are all synchronized and updated as one, basic client side styling is available and map popups present the attributes of the vector tiles.
- GeoServer implementation of available datasets, which can be found here: http://vtp2018.s3-eu-west-1.amazonaws.com/datasets.html. Popups also contain tile attribute information. It should be noted that mis-using MVT can produce poor performance such as a lack of zoom and caching.

Ecere

The Ecere WFS 3.0 client is implemented as a capability of Ecere's GNOSIS software libraries, available from within Ecere's GNOSIS Cartographer GIS tool, as well as from any applications built using the GNOSIS SDK (whether for desktop, web or mobile). Support for accessing tiles through the WFS 3.0 REST API has been added for this pilot, with support for GNOSIS Map Tiles, GML, GeoECON, GeoJSON and Mapbox vector tiles. Visualization of Mapbox vector tiles and GeoJSON were entirely new capabilities developed during the pilot.

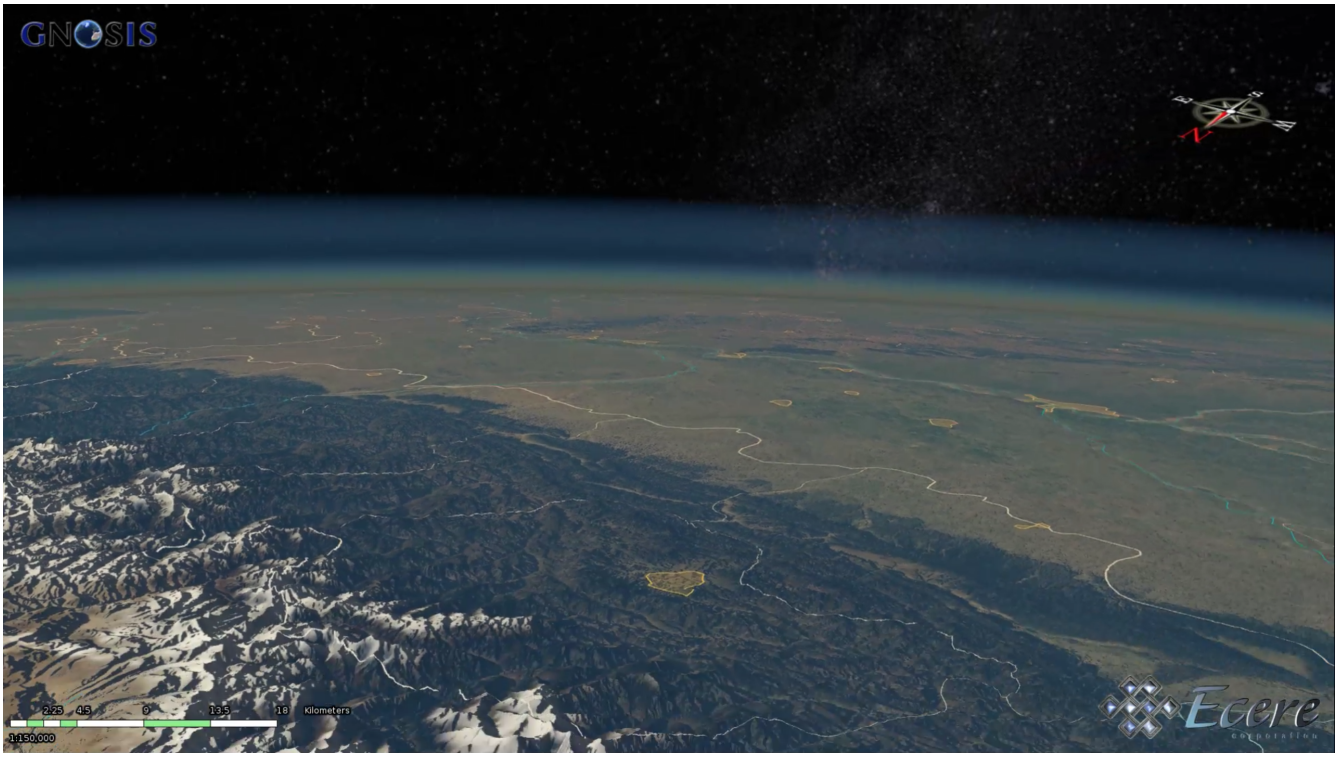

Figure 5. Vector, imagery and terrain elevation tiles served from a single endpoint by Ecere's Harmonized (WFS 3.0 style) Map Service rendered in Ecere's GNOSIS client

9.2. GeoPackage 1.2 Vector Tiles Extension

As part of the pilot effort, a Vector Tiles extension has been created for GeoPackage. Prior to this extension there was no defined interoperable, agreed upon method to store vector tiles in a portable container. The primary use case for GeoPackage is to be a portable container for use primarily in Denied, Degraded, Intermittent, or Limited Bandwidth (DDIL) networks. DDIL use cases can utilize GeoPackage and vector tiles as the size and therefore delivery of vector tiles is orders of magnitude smaller than the raster equivalents. However, initial load of the GeoPackage onto a mobile device is required prior to utilization in a disconnected environment meaning that remote updating remains problematic.

Overall, the extension is based upon the MVT specification and uses Google Protocol buffers to encode the content within each tile. The extension requires GeoPackage core, and like all extensions, will run with clients that do not support it.

The extension is defined by creating new metadata tables within a GeoPackage:

- gpkgext_vt_layers. Contains columns:
  - id - primary key
  - table_name.
  - name - the name of the individual layer.
  - description - an optional description of the tile set.
  - minzoom - the minimum zoom level.
  - maxzoon - the maximum zoom level.
- gpkgext_vt_fields. Contains columns:
  - id - primary key.
  - layer_id.
  - name - the name of the attribute.
  - type - the data type for the column, can be string, number or boolean.

There are several Vector Tiles extensions that have been developed as part of the pilot that are discussed in the next sections. There is no *single* Vector Tiles extension to GeoPackage, instead the extensions are modularized for simple implementation and to increase interoperability with services supporting the core standard only.

9.2.1. Mapbox Vector Tiles Extension

The Mapbox Vector Tiles extension for GeoPackage defines the rules and requirements for encoding vector tiles in a GeoPackage Data Store, it is expressed as a draft standard. A key aspect to this extension is that the data are encoded using Google Protocol Buffers, which has many advantages discussed elsewhere in this document. This provides noticeable performance enhancements over other vector data formats and encodings.

9.2.2. GeoJSON Vector Tiles Extension

The GeoJSON Vector Tiles extension defines the rules and requirements for encoding GeoJSON Vector Tiles in the GeoPackage Data Store. It is not the focus of this pilot but may be utilized in future OGC initiatives. GeoJSON vector tiles are considered here as the GeoJSON format is useful for exchange with other services and vendors.

9.2.3. OWS Context Extension

The context document is a method of storing context of vector tiles as part of a GeoPackage file. OWS Context is an OGC standard that has encodings in Atom and GeoJSON. An OWS Context extension for GeoPackage is not yet adopted by OGC and remains a candidate standard. There are several requirements for this extension that are explored in the VT GeoPackage Extension ER.

9.2.4. Vector Tile Attribute Extension

Attributes can be stored as part of Vector Tiles Binary Large Objects (BLOBs) in GeoPackage, however enabling this extension allows vector tiles objects to be queried without having to open the individual tiles, as the attribution is stored as a table within the GeoPackage.

9.2.5. GeoPackage Producers

GeoPackages with Vector Tiles extensions were produced by Compusult, CubeWerx and Ecere.

9.2.6. GeoPackage Clients

There are several GeoPackage clients from a variety of vendors.

Image Matters

GeoPackage.js is an in-browser GeoPackage client currently capable of displaying tile tables with image data and feature tables with vector data from a GeoPackage. Image Matters has added support for vector tiles encoded using the MVT Specification and employed the Leaflet VectorGrid library to perform rendering. This implementation extends VectorGrid.Protobuf from this library to retrieve Protobufs from a GeoPackage rather than from the web. Metadata, tile bounds, attributes and layers within a vector tile table can all be viewed by expanding the tables details.

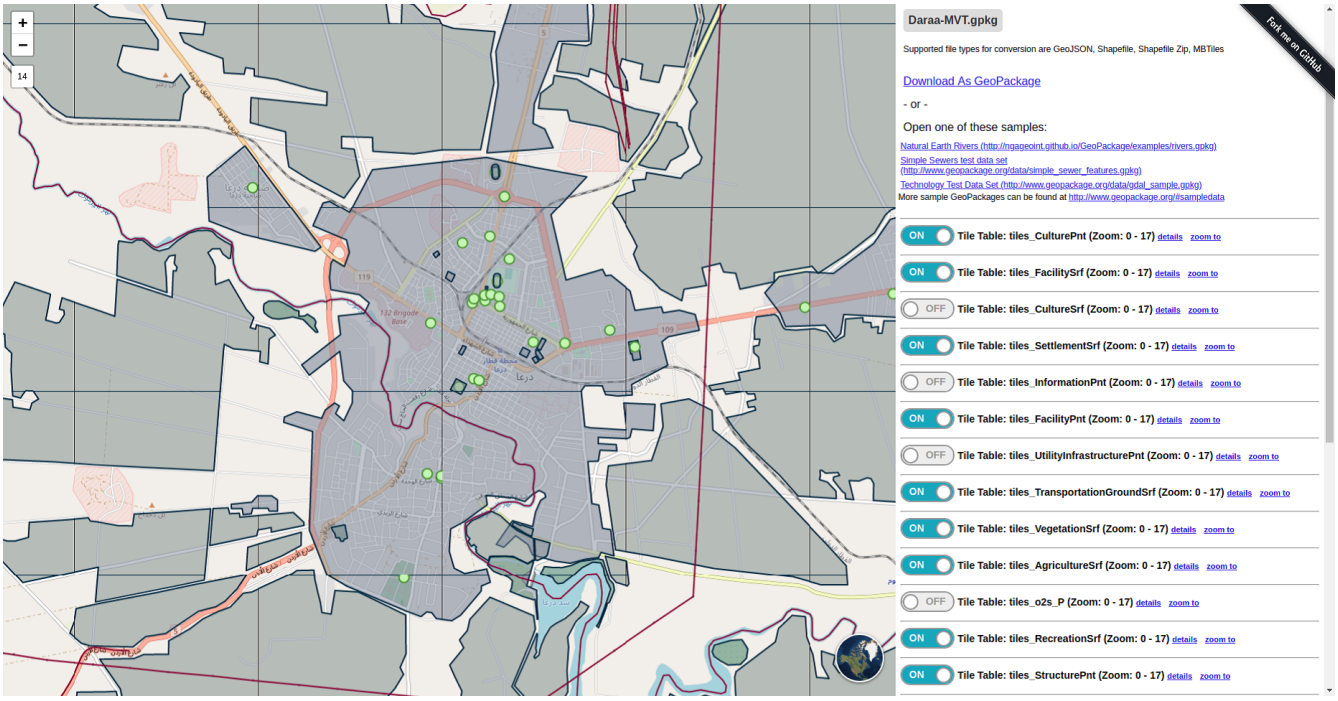

Figure 6. Ecere Vector Tiles Rendered on the Image Matters Web Client

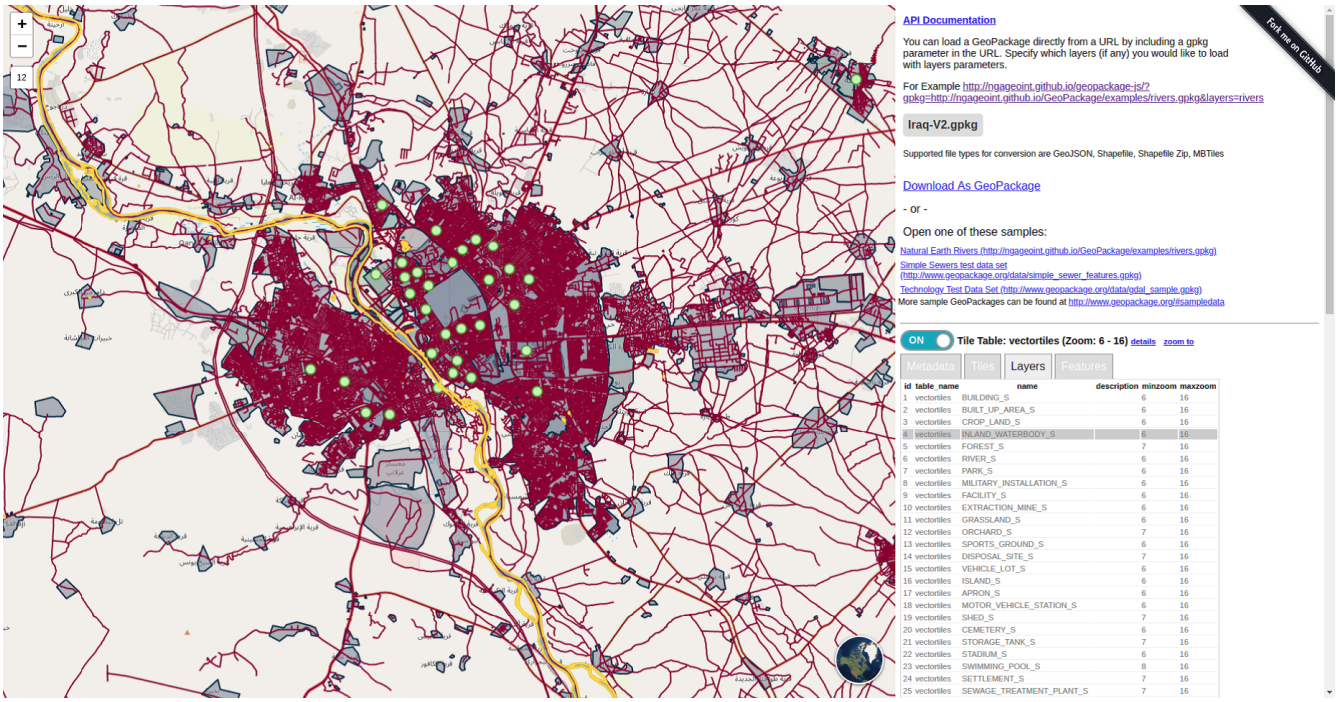

Figure 7. Compusult Vector Tiles Rendered on the Image Matters Web Client

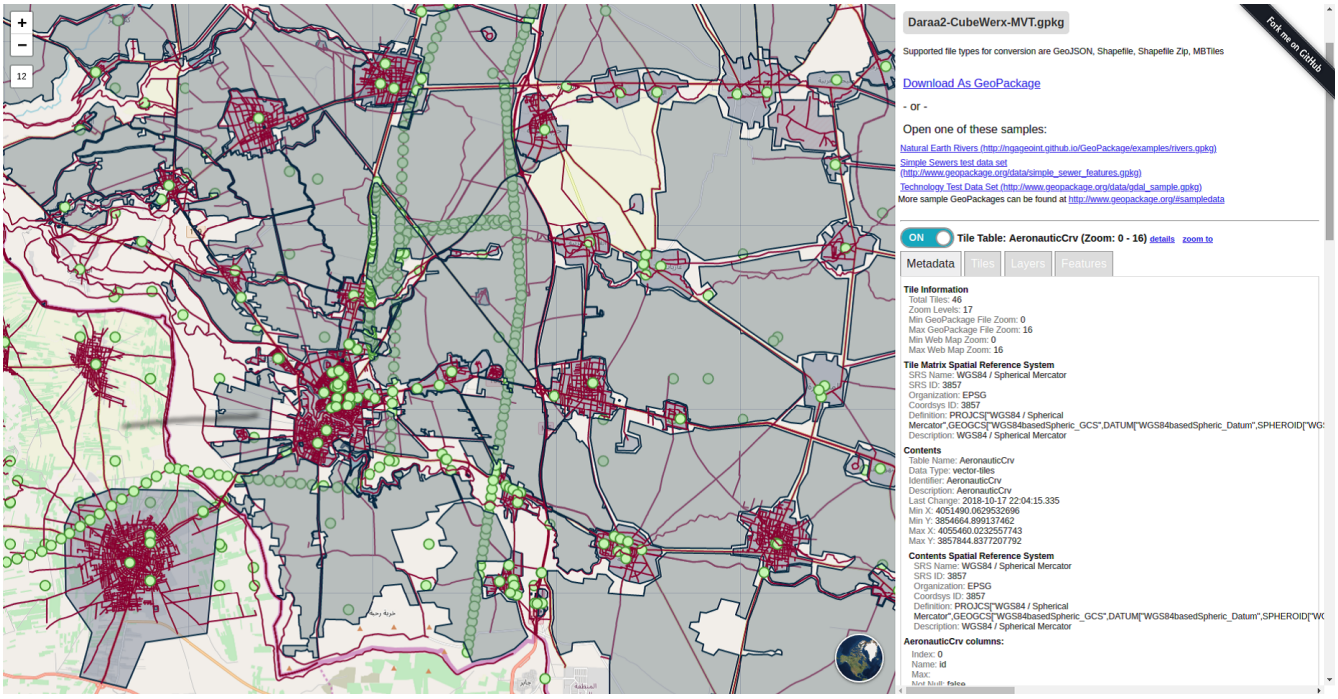

Figure 8. CubeWerx Vector Tiles Rendered on the Image Matters Web Client

Compusult

Mapbox Vector Tile and GeoJSON tile formats are both supported by the Compusult GeoPackage client which can run as both a web-based or desktop/Google Android based application. Users can also utilize specific WMTS layers and overlay base maps for visualization. It should be noted that the style of each layer is randomly generated and therefore the different GeoPackage providers were rendered differently by the GeoPackage client. During the pilot, the client ran as a web-browser based application that used a GeoPackage to produce internal WMS/WMTS services that were then rendered by the browser. Several minor issues are noted in the VT GeoPackage Extension ER, along with the recommendations to include internet media type in the data-model and avoid dependence on sibling tiles to perform rendering operations.

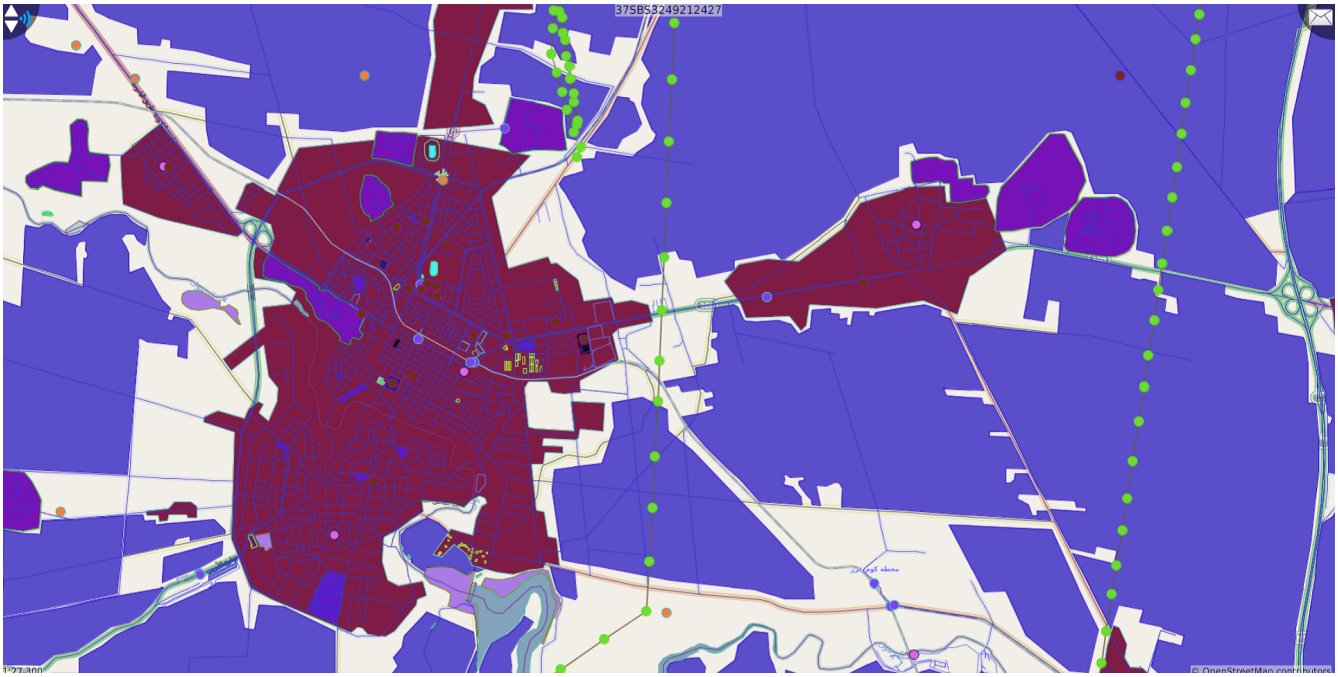

Figure 9. CubeWerx Vector Tiles Server Rendered on the Compusult Client

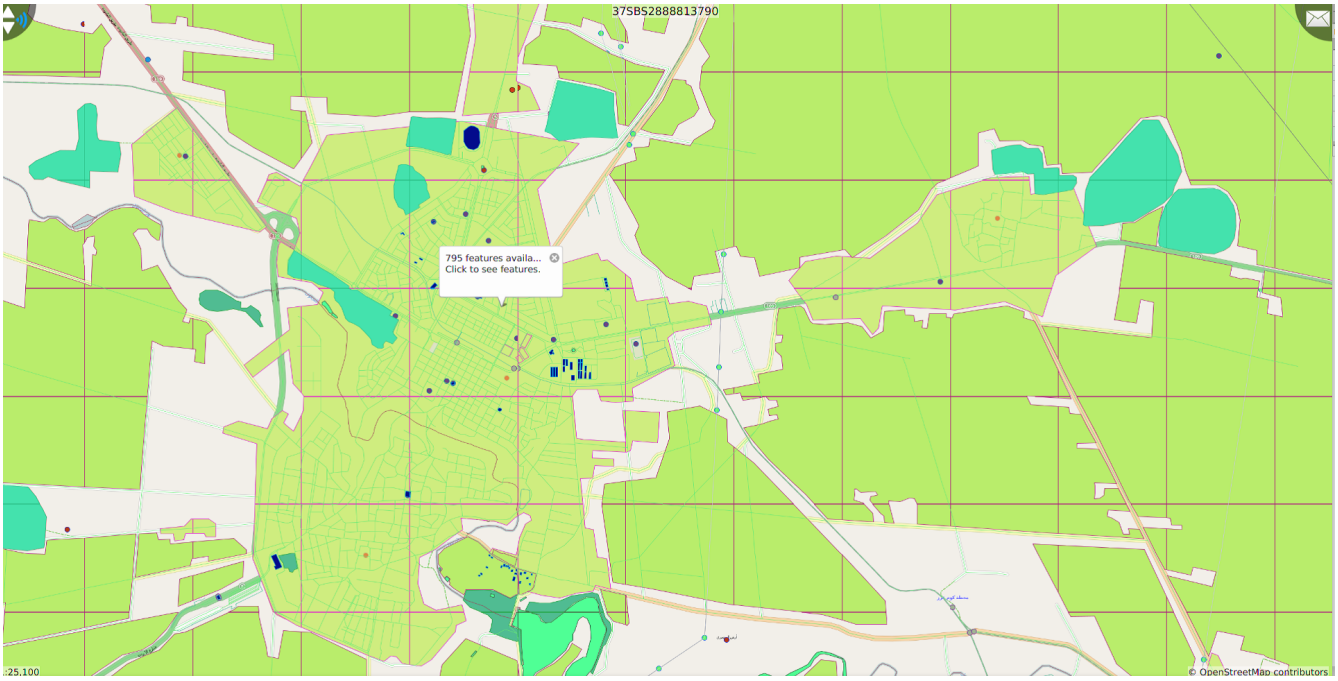

Figure 10. GeoSolutions Vector Tiles Rendered on the Compusult Client

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 11. Ecere Vector Tiles GeoPackage Rendered in Ecere's Visualization Client

Figure 12. Compusult Vector Tiles GeoPackage Rendered in Ecere's Visualization Client

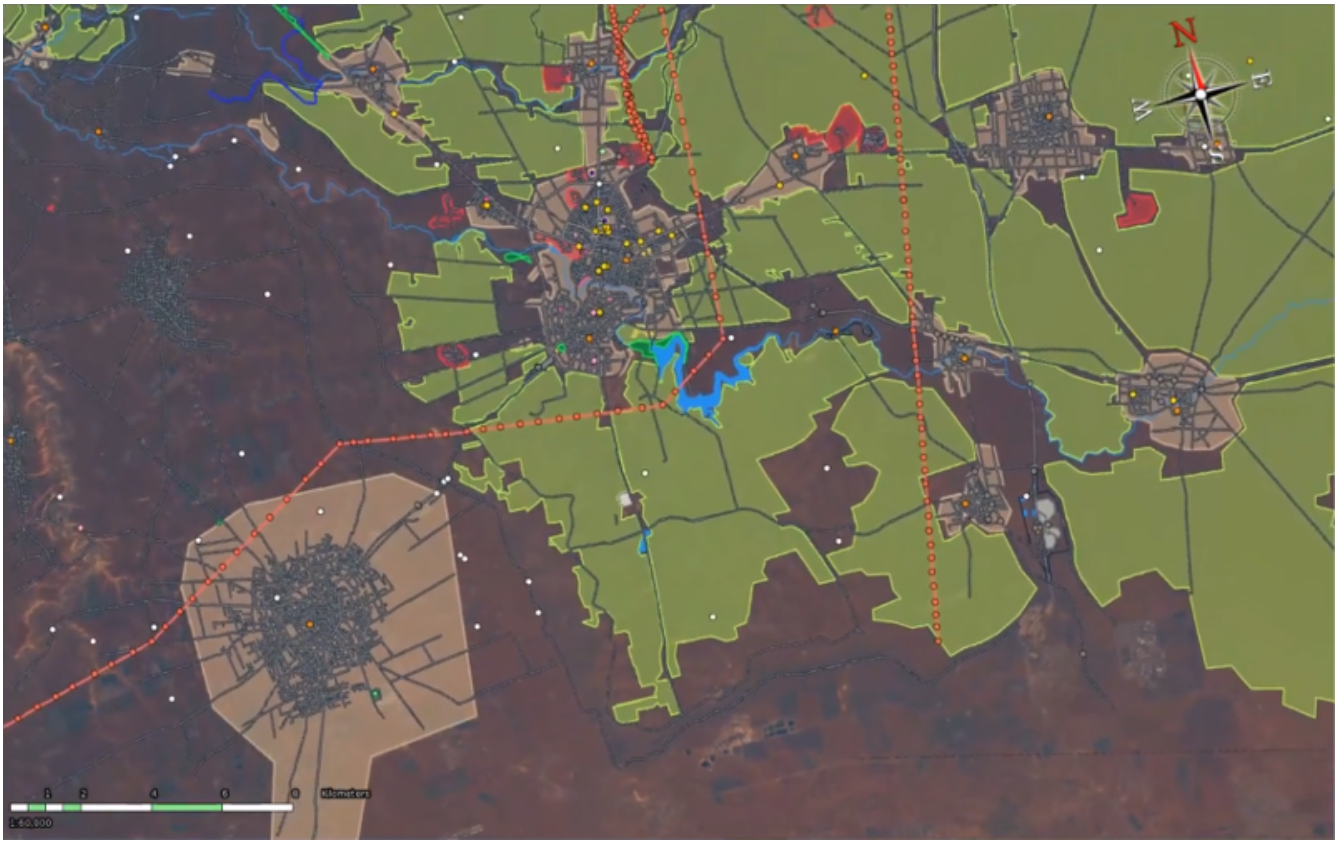

Figure 13. CubeWerx Vector Tiles GeoPackage Rendered in Ecere's Visualization Client

9.3. WMTS for Vector Tiling

WMTS is an OGC service designed to offer a fixed set of graphical images in pixel space to a client. From the perspective of clients there is therefore no opportunity to dynamically style tiling on-the-fly. One of the objectives of the WMTS aspect of the pilot is the ability to provide vector tiles as a standard output of any service. The focus of this work item is Mapbox vector tiles, however there is also a requirement to implement GeoJSON vector tiles as a separate output. Several implementations were produced, as with the other work items in the Pilot.

9.3.1. WMTS servers

CubeWerx

Deployed on an Amazon instance the CubeWerx WMTS supports a range of tile formats and the service offers a range of operations and vendor-specific operations. No changes were necessary to the WMTS specification to support MVT, however several third-party libraries had to be integrated and Mapbox vector tiles had to be generated as per the official MVT specification. The test URL for this server is located here:

https://tb14.cubewerx.com/mvtTest/

Ecere

The Ecere WMTS service [http://maps.ecere.com/wmts] offers GetCapabilities and GetTile operations along with supporting a range of tile formats and tile matrix sets. No changes were necessary to the WMTS other than adding support for MVT as an additional output format, support for which was

developed during the pilot. Testing and validation of the server was completed using *vtvalidate* (https://github.com/mapbox/vtvalidate), a Mapbox tool which performs a series of tests on an encoded tile, such as consistency with specifications and decoding layers, features, properties and geometries; with OGR software; with QGIS; as well as with the Ecere and Compusult WMTS clients.

GeoSolutions

The GeoSolutions WMTS service offers GetCapabilities, GetTile and GetFeatureInfo along with supporting application/vnd.mapbox-vector-tile tile format and a range of Tile Matrix sets. No changes were necessary to the WMTS, MVT was just included as an additional output format. The WMTS Vector Tiles Extension Engineering Report details the characteristics of the GeoServer implementation such as zoom level specific filters and rules determining the features within the tile layer.

Mapbox

The Mapbox WMTS service implemented version 1.0 of the WMTS standard. The service offers GetCapabilities and GetTile operations.

9.3.2. WMTS Clients

CubeWerx

The CubeWerx implementation integrates several third-party libraries to enable encoding using Google Protocol Buffers and generation of the Mapbox vector tiles as per the official specification. Generation of GeoJSON vector tiles was implemented in parallel to Mapbox VT generation as required by the Pilot. Access to the CubeWerx server is done via the client shown in Figure 14.

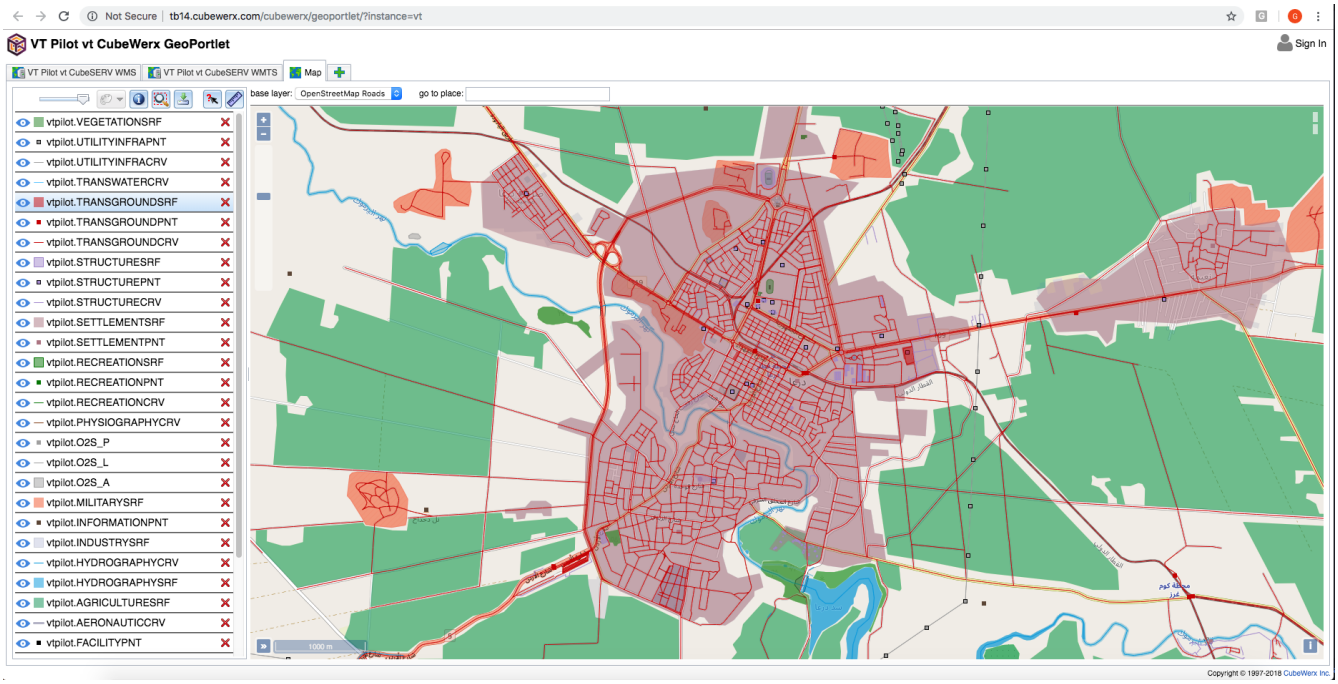

Figure 14. The WMTS Client provided by CubeWerx

Ecere

The Ecere WMTS client is implemented as a capability of Ecere's GNOSIS software libraries,

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 accessing vector tiles through WMTS has been added for this pilot, with support for GNOSIS Map Tiles, GML, GeoECON, GeoJSON and Mapbox vector tiles. Visualization of Mapbox vector tiles and GeoJSON was an entirely new capability developed during the pilot.

Figure 15. The WMTS Client provided by Ecere

GeoSolutions

Figure 16 shows the WMTS client for vector tiles, in reality, it is six clients in one that show vector tiles rendering and styling in several different formats. The clients shown are:

- OpenLayers.
- Leaflet.
- Mapbox GL.

A key advantage to using vector tiles is the styling aspect, the clients also show a mixture of VT styles that include:

- Mapbox native.
- OpenLayers native.
- Leaflet native.
- Tangram styling by Mapzen.

Overall, this client shows the flexibility of vector tiles in a variety of different formats and styles.

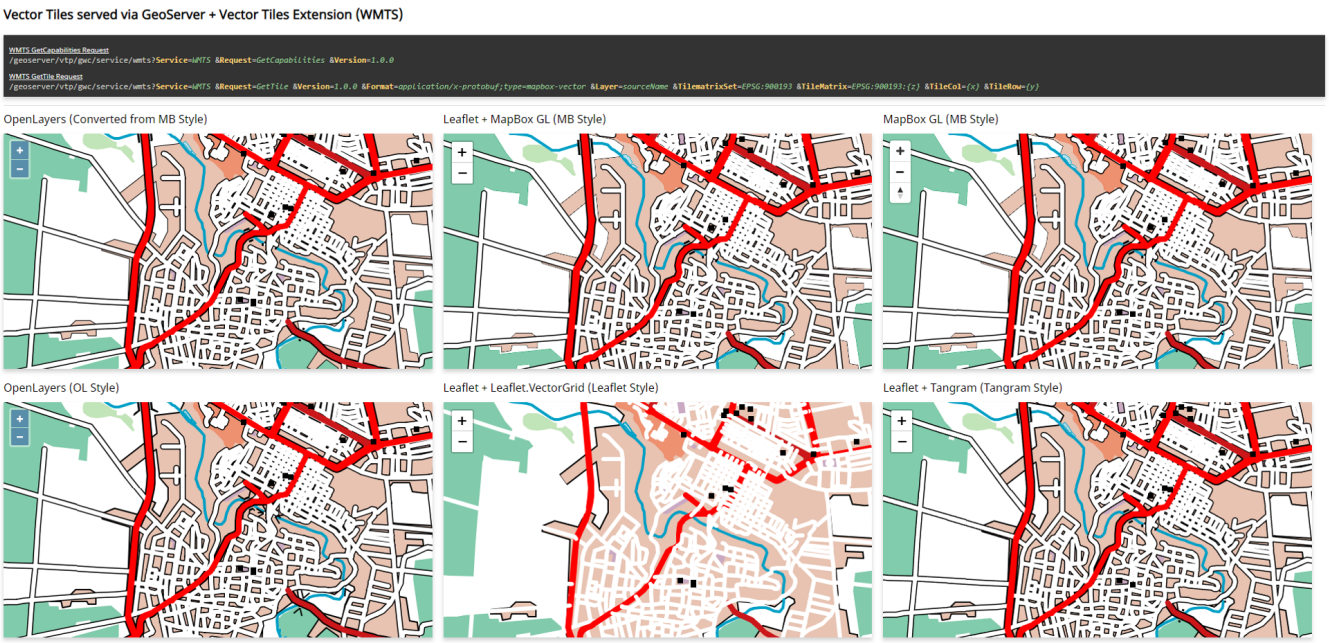

Figure 16. The WMTS Client provided by Geosolutions

Compusult

Mapbox Vector Tile and GeoJSON tile formats are both supported by the Compusult WMTS client which can run as both a web-based or desktop/Google Android based application. Users can also utilize specific WMTS layers and overlay base maps for visualization. It should be noted that the style of each layer is randomly generated and therefore the different GeoPackage providers were rendered differently by the WMTS client. Several minor issues are noted in the WMTS Vector Tiles Extension Engineering Report, along with the recommendations to; avoid dependence on sibling tiles to perform rendering operations and consider including a WMTS extension which would allow users to filter MVT layers with a multi-layer composition.

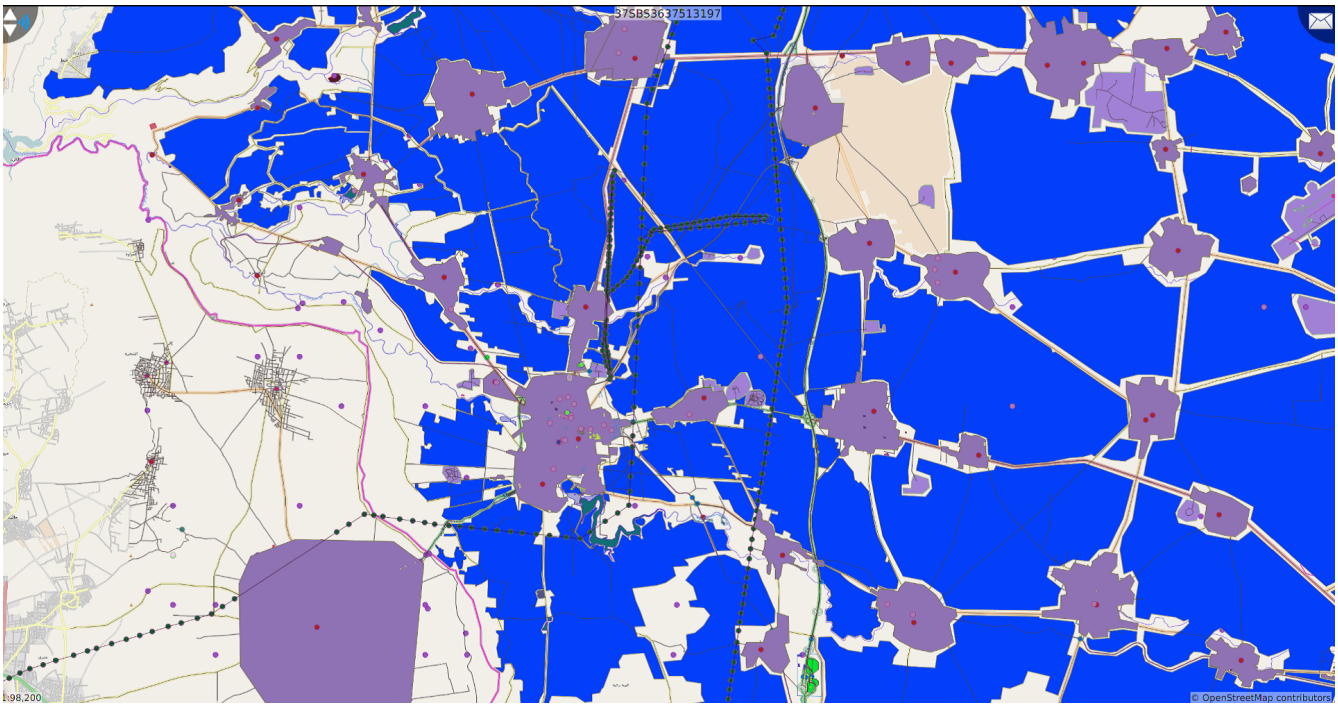

Figure 17. CubeWerx Vector Tiles Server Rendered on the Compusult Client

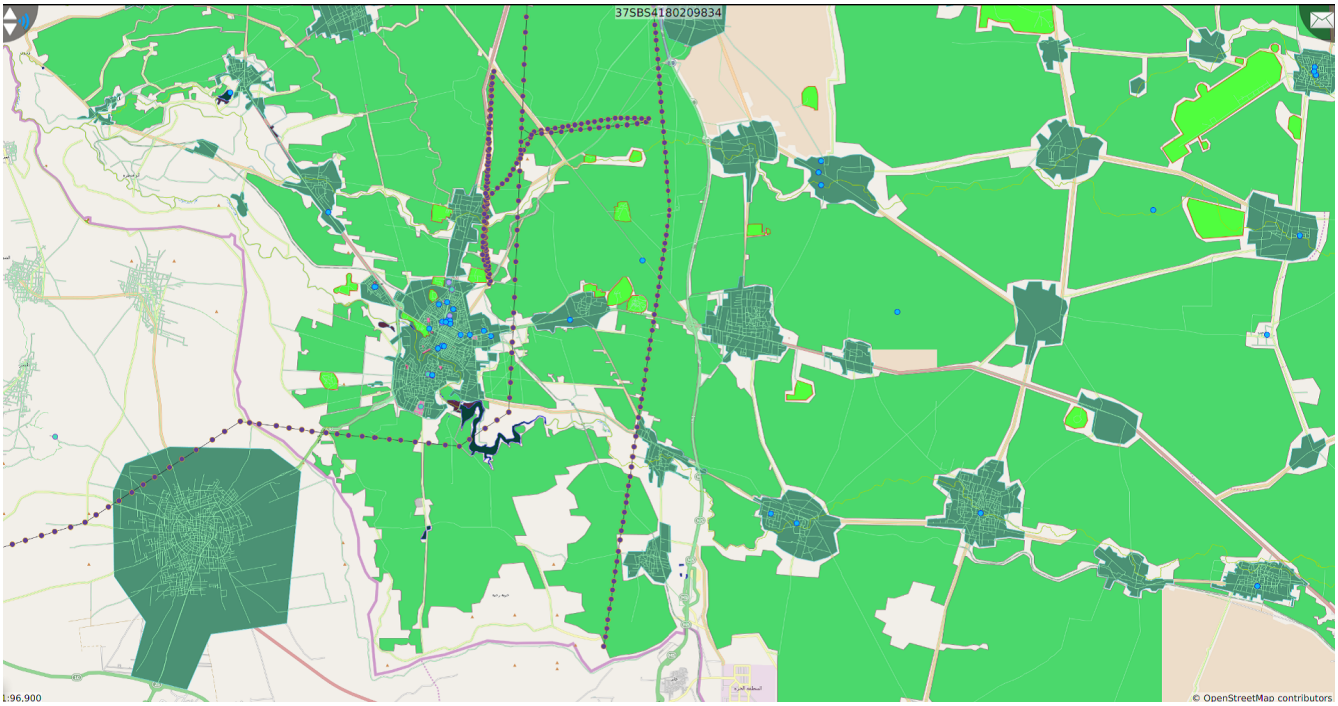

Figure 18. GeoSolutions Vector Tiles Rendered on the Compusult Client

Mapbox

Mapbox are the vendors and originators of the MVT standard. As with the other implementations, no changes are required to the standard to support MVT as an output. Mapbox have also produced a WMTS client.

Challenges with Mapbox Vector Tiles in WMTS

Several challenges specific to WMTS were encountered during the implementation phase, these include:

- Support of multiple geometry types within a single tile set.
- Simplification and clipping of geometries which was a consideration of Testbed-12 Vector Tiles Engineering Reports.
- Making specific VT parameterization configurable.
- Generation of tiles on a low-level, with the flexibility to support any source and parameter set.
- Generation of WMTS Tile Matrix sets.

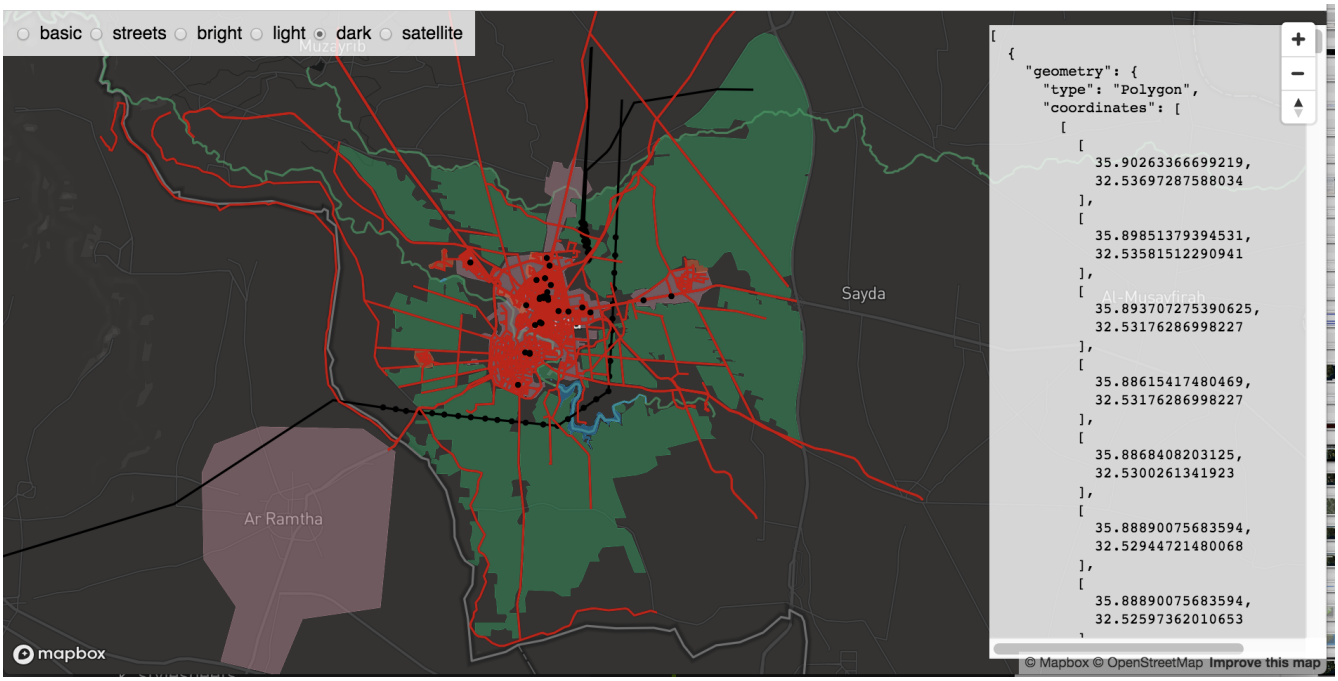

Figure 19. Vector Tiles Rendered on the Mapbox Client

Chapter 10. Discussion

This section revisits and summarizes the points of discussion and next steps from across the work completed as part of the Vector Tiles Pilot. Conceptually, decisions have been made to *bind* the conceptual model to enable straightforward implementation, many of these discussions were concerned with scoping of the model. Ultimately, decisions were made to enable a viable implementation, the major aspects for consideration were:

- Visualization. Vector Tiles implementations often contain their own visualization specifications, for example MVT.
- Feature properties. Encoding properties varies across specifications.
- Naming. Standards outline different terms for a tile scheme, similar terms should be named _"TilingScheme".
- Tile shape. Most schemes use a rectangle, however, this is not mandatory as inclusion of other shapes are theoretically possible

WFS 3.0 is an emerging standard within the OGC, there are several points of discussion raised regarding the implementation and exposure of vector tiles using this methodology. One of the points that was raised in several use cases is the methodology for requesting a tile with features from multiple collections into a single response tile. Currently, WFS 3.0 does not define a mechanism for allowing simultaneous access to an enumerated set of feature collections into a single request. Extension of the standard has been proposed to address this shortcoming through an extra path called /items for requesting an entire feature set and a second path called /collections to limit the scope of the request and limit the returned features to enumerated collections.

WMTS is the least complicated to extend to support vector tiles due to the extensible nature of the standard. No changes were necessary to the WMTS specification to support MVT and in most cases it was simply included as an additional output format offered by the WMTS. The main points of discussion for implementation are concerned with the media type (i.e. MIME type) for Mapbox vector tiles and usage of the *Format* parameter in the WMTS call. Essentially, the Internet Assigned Numbers Authority (IANA) has designated *application/vnd.mapbox-vector-tile* as the correct format for the media type, therefore this is what should be used to refer to the format throughout the standards.

GeoPackage 1.2 extension enables vector tiles to be made available to mobile clients utilizing the standard. GeoPackage is based upon an SQLite database and therefore takes advantage of related tables to store information. Much of the discussion in GeoPackage is concerned with how to best utilize the table structure of a database to best serve tiles.

10.1. Tile Compression

Three separate GeoPackages were produced:

- MVT.
- GeoJSON.
- JPG/PNG.

It should be noted that MVT and GeoJSON benefit greatly from compression shrinking by 90% and 96% respectively. The capability to Deflate compress individual tiles inside the SQLite database was added to GeoPackage near the end of the Pilot, unfortunately after the development of many of the test GeoPackages.

10.2. Attributes

Storing attributes for vector tiles in GeoPackage has been subject to debate in the community. Two main methods have been identified:

- Storing an individual table for each tile.
- Using the Related Tables extension (RTE).

The power of relational databases is the ability to relate tables to de-duplicate storage requirements, therefore there is a preference for the RTE rather than the verbose method of storing a table per tile. Additionally, performing queries across tiles can be problematic as overlapping features would appear in some form in multiple tables. The RTE is the GeoPackage method of managing many-to-many relationships, a typical use case for vector tiles and features that all have attributes. Currently, the RTE does not support requirements specific to vector tiles, that is, the tiles acting as the *base* data and the attributes as the *relationship*, therefore two tables are proposed for the RTE:

- gpkg_extensions - provides the name of the relationship mapping table.
- gpkg_relationships - provides a description of the extended relationships, i.e. the tile to attribute mapping.

10.3. Styling and Symbology

An advantage of vector tiles is the ability to de-couple elements, this has already been discussed with respect to tiling and attributes. Another requirement is to de-couple styling from tiling, there are two main proposals for accomplishing this in GeoPackage:

- De-couple the styles from the data (i.e. store them separately). This requires minor changes to the OWS Context standard discussed earlier.
- Store the styles via tabular encoding. Tabular encoding of styles has the advantage of direct access to the required information without having to parse documents. This is being explored in OGC Testbed-14 and potentially in a further phase of the VTP.

10.4. Vector Tile format differences

Across the implementations, participants noted that Mapbox Vector Tiles offer a significant performance increase over other tiling approaches due to the mandatory inclusion of Google Protocol Buffers as the encoding format. GeoJSON vector tiles do not experience a performance increase over other formats but are an emerging format for exchange and were implemented for interoperability reasons.

10.5. DDIL Considerations

As mentioned previously, DDIL networks are a concern in many information communities. Traditionally, OGC services assumed consistent and stable internet connectivity. Delivery of data in DDIL environments requires assumptions regarding connectivity be readdressed. There are several strategies that can be adopted to compensate for DDIL environments that include:

- Pre-loading or pre-caching.
- Delivery of updates through physical means.
- An ad-hoc network.
- Making best use of networks that are available.

If a network is available, then techniques such as compression can be used to deliver the data required. Mapbox vector tiles are well suited to these use cases due to the implementation of Google Protocol Buffers that provide optimal compression for delivery across a network. Another strategy used to take advantage of intermittent networks is the use of asynchronous services, where the server processes the request in the background and delivers the result when notified of client connectivity. Possible extensions to each of the standards considered are outlined in the individual ERs. Another method considered is the *publish/subscribe* method (pub/sub) that utilizes polling to synchronize changes across a network when required. This method is suited to more frequent updates and forgiving DDIL environments.

Overall, some elements of the VTP have shown that the technologies described are suited to these use cases, but there is more work to be done in explicitly extending the standards to DDIL environments.

10.6. Convergence of OGC Services

Throughout the Pilot project there were discussions regarding the remit of the different OGC services and tiled data. Essentially, tiled feature data and supporting products are built from features that have attributes, much like any other vector data. Therefore, much of the exercise of the Pilot was to produce tiled feature data using the three OGC standards of interest. The standards used are not comparable, for example GeoPackage 1.2 is essentially a container and WFS 3.0 defines access protocols for feature data on a server. WMTS provides access to tiled matrix datasets, but in contrast to WFS 3.0, these tile matrices are pre-rendered. The WFS 3.0 extension proposed in this Pilot also has the ability to access pre-rendered tiles. This utilization of OGC services begs the question as to whether WFS 3.0 can fulfill all of the service requirements outlined in this Pilot; data for mobile, data for desktop and data for the web. The mobile data could be created by replacing the *GeoPackage producer* with WFS 3.0 with tiled data encoded with Google Protocol Buffers in a GeoPackage format.

This discussion may have further consequences for OGC services such as whether WFS 3.0 can be utilized to serve tiled data in terms of their features. The question is also whether WFS 3.0 can be extended to serve other *files* as well? This has obvious consequences for WCS, WMS and WMTS as their functionality could be re-produced using a WFS 3.0 style architecture based upon OpenAPI.

As OpenAPI is an interface description, it does not mandate what happens on the server, which can

be treated as a *black box*. For example, there is no required "GetCapabilities" call that would be exposed via a REST endpoint as OpenAPI is concerned with the exposure of *resources*, there is no one interface to one server relationship implied or mandated. Therefore, there is no reason that geoprocessing cannot be covered by an extended version of the WFS 3.0 interface. In a strict interpretation of geoprocessing, this is *already* happening under the current WFS 3.0 specification because of the operations that can be performed, for example, filtering and applying styles to vector tiles requires some level of geoprocessing to be happening server side to deal with the request. One suggestion for a converged capability is the Web Object Service (WOS), although it remains to be seen whether a converged service will garner the required response from the Technical Committee. The reason for the lack of support for a converged capability is that modularization is a key principle of OGC Service Architecture that makes it possible to build a variety of solutions due to the ‘separation of concerns’.

Testbed-14 contains a study on the implications of technologies such as Docker, Kubernetes and Cloud Foundry being utilized in the standards work of the OGC. Essentially these technologies have the ability to dynamically allocate resources using container technology to deliver results. A REST based interface could be put in front of such an architecture for efficient and dynamic processing and data delivery. The implications of this on tiled data and the wider OGC should be explored in future initiatives.

Chapter 11. Recommendations

This section contains the recommendations summarized from the individual ERs in the VTP as well as overall recommendations in relation to on-going discussions in the OGC.

- Some of the technical issues with MVT requests remain and should be addressed in future Pilot or Testbed initiatives:
  - MVT tile sets with multiple layers are inefficient to discover and query as every tile needs to be requested to produce a layer listing with primitive geometry type.
  - Avoiding unwanted edges of polygons can be done using two different methods; an extra border or using artificial segments. These should be offered as part of the MVT request.
  - Feature ID should be carried from original features to tiled features, this requirement is currently ambiguous.
  - Tiling schemes should support a multitude of CRSs.
- Investigate and present the concept of a Web Object Service to the Technical Committee to discuss whether resource or service convergence is determined, wanted or to be resisted. A requirement gathering exercise is needed across the relevant DWGs and SWGs, which is likely to be most of them.
- Investigate standardization of styling and symbology to include:
  - Standardizing the styles applied to vector tiles for specific instances e.g. default styling for grass could be green and water blue.
  - Investigating a set of values for a style document initially using the Mapbox Style specification [9] and then expanding the investigation to include the OGC Style Layer Descriptor (SLD) standard.
  - Investigating the idea of including multiple features into a single vector tile.

Chapter 12. Conclusion

This report provides an overview of the Vector Tiles Pilot endeavor with emphasis on the findings of the component implementations and the conceptual model. Although self-contained, the Vector Tiles Pilot has provided opportunities for future work in this space. Overall, vector tiling is an important concept and is an efficient method of transporting large volumes of data.

Appendix A: Revision History

Table 2. Revision History

Date	Editor	Release	Primary clauses modified	Descriptions
2018-08-22	S. Meek	.1	all	initial version
2018-10-18	E. Ogden	.2	various	update
2018-10-30	A. Aime	.3	various	update
2018-11-05	J. St Louis	.4	various	update

Appendix B: Bibliography

1. Mapbox: Mapbox Vector Tile Specification - version 2.1, https://www.mapbox.com/vector-tiles/specification/, (2016).
2. Ingensand, J., Maia, K.: OGC Vector Tiles Pilot: Tiled Feature Data Conceptual Model Engineering Report. OGC 18-076,Open Geospatial Consortium, https://docs.opengeospatial.org/per/18-076.html (2019).
3. Vretanos, P.A.: OGC Vector Tiles Pilot: WMTS Vector Tiles Extension Engineering Report. OGC 18-083,Open Geospatial Consortium, https://docs.opengeospatial.org/per/18-083.html (2019).
4. Vretanos, P.A.: OGC Vector Tiles Pilot: WFS 3.0 Vector Tiles Extension Engineering Report. OGC 18-078,Open Geospatial Consortium, https://docs.opengeospatial.org/per/18-078.html (2019).
5. Yutzler, J.: OGC Vector Tiles Pilot: GeoPackage 1.2 Vector Tiles Extensions Engineering Report. OGC 18-074,Open Geospatial Consortium, https://docs.opengeospatial.org/per/18-074.html (2019).
6. Balog, D., Houtmeyers, R.: OGC Testbed-12: Vector Tiling Implementation Engineering Report. OGC 16-067r4,Open Geospatial Consortium, http://docs.opengeospatial.org/per/16-067r4.html (2017).
7. Balog, D., Houtmeyers, R.: OGC Testbed-12: Vector Tiling Engineering Report. OGC 16-068r4,Open Geospatial Consortium, http://docs.opengeospatial.org/per/16-068r4.html (2017).
8. Cavazzi, S.: OGC Testbed-13: Vector Tiles Engineering Report. OGC 17-041,Open Geospatial Consortium, http://docs.opengeospatial.org/per/17-041.html (2018).
9. Mapbox: Mapbox Style Specification, https://docs.mapbox.com/mapbox-gl-js/style-spec/.