Create / distribute tiled map

Taro Matsuzawa

Georepublic Japan

Notes (1/2)

- This presentation use some links to external resources
  - Please download the presentation from the link below

Notes (2/2)

- This presentation is written in Markdown
  - You can edit this presentation with any text editor
  - You can convert this presentation to PDF or PowerPoint, please see the README.md
  - You can download the presentation of auto generate versions from the link below
    - PDF
    - PowerPoint
    - HTML

Self introduction

- GIS Engineer at Georepublic Japan
  - Programming: Python, JavaScript, TypeScript, Ruby etc.
  - UNIX and Linux guru
  - GIS skill: Data processing, Tiled based Map
- Community
  - Director of OSGeo.JP
  - Director of OpenStreetMap Foundation Japan
  - Sub president of Japan Unix Society
  - UNOpenGIS/7 volunteer
- Contact: taro@georepublic.co.jp / @smellman on Twitter

Today's agenda

- System setup
- What is tiled map?
- Introduction of software and data in this presentation
- How to create your own tiled map
- How to design your own tiled map
- How to distribute your own tiled map

System setup

- This presentation requires Linux based OS.
  - Use Raspberry Pi 4.

System setup - Connect to jump host

- Connect to SSID "vectortiles"
- Launch Terminal
  - Windows: Use PowerShell
  - Mac: Use Terminal.app
- Connect to Raspberry Pi with SSH

```
ssh portal@j2213.local
```

System setup - Connect to Your Raspberry Pi

```
make <YOUR HOST NAME>
```

e.g.

```
make m321
```

System setup - Install software

```
sudo apt install -y git make
git clone https://github.com/smellman/jica_scripts.git
cd jica_scripts/system
sudo HOME=$HOME USER=$USER make install
```

What is tiled map?

Tile technology

- Provide map image or data over the internet.
  - Map images are separated as tiles.
  - Zoom Level 0 = World
  - Each zoom level doubles in the dimensions.
  - Too many tiles use "Web Mercator" projection.

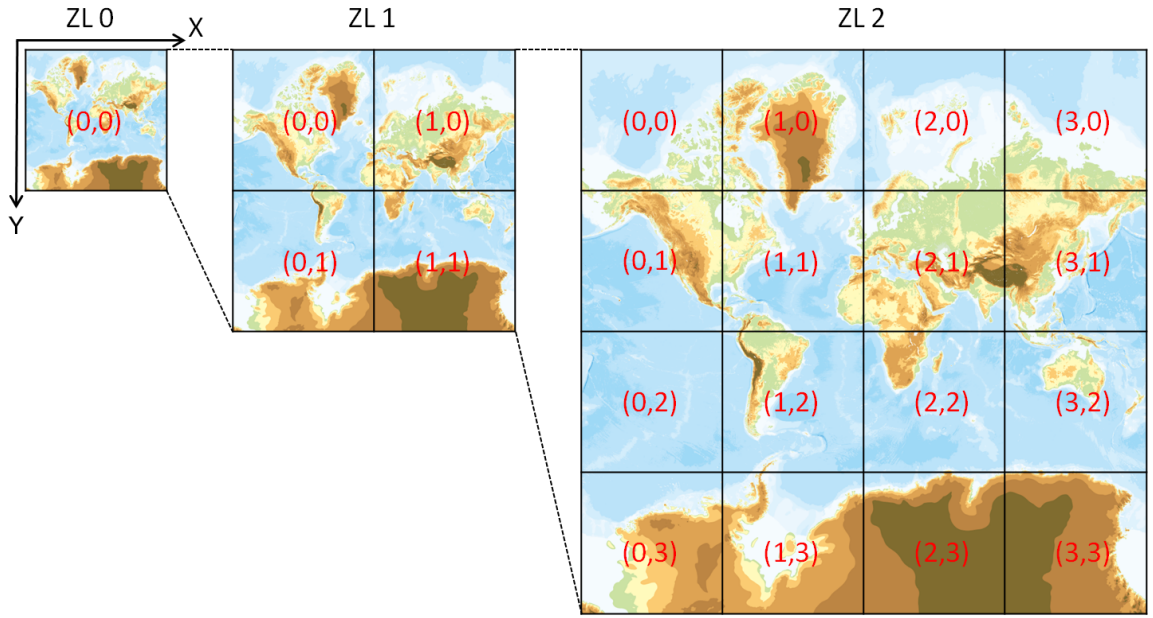
https://a.tile.openstreetmap.org/0/0/0.png

Useful to web

- Structure of tile is useful for web.
  - Enable to scroll map smoothly.
  - Enable to zoom up and zoom down map smoothly.
  - HTTP GET request.
- Tile become known for Google Maps.
  - Tile has existed from the late 1990s.

Zoom

- Zoom level 0 : 1 file
- Zoom level 1 : 2 x 2 = 4 files
- Zoom level 2 : 4 x 4 = 16 files
- ...
- Zoom level 18 : 2^18 x 2^18 = 262,144 x 262,144 = 68,719,476,736 files



https://maps.gsi.go.jp/help/image/tileNum.png

GET Request

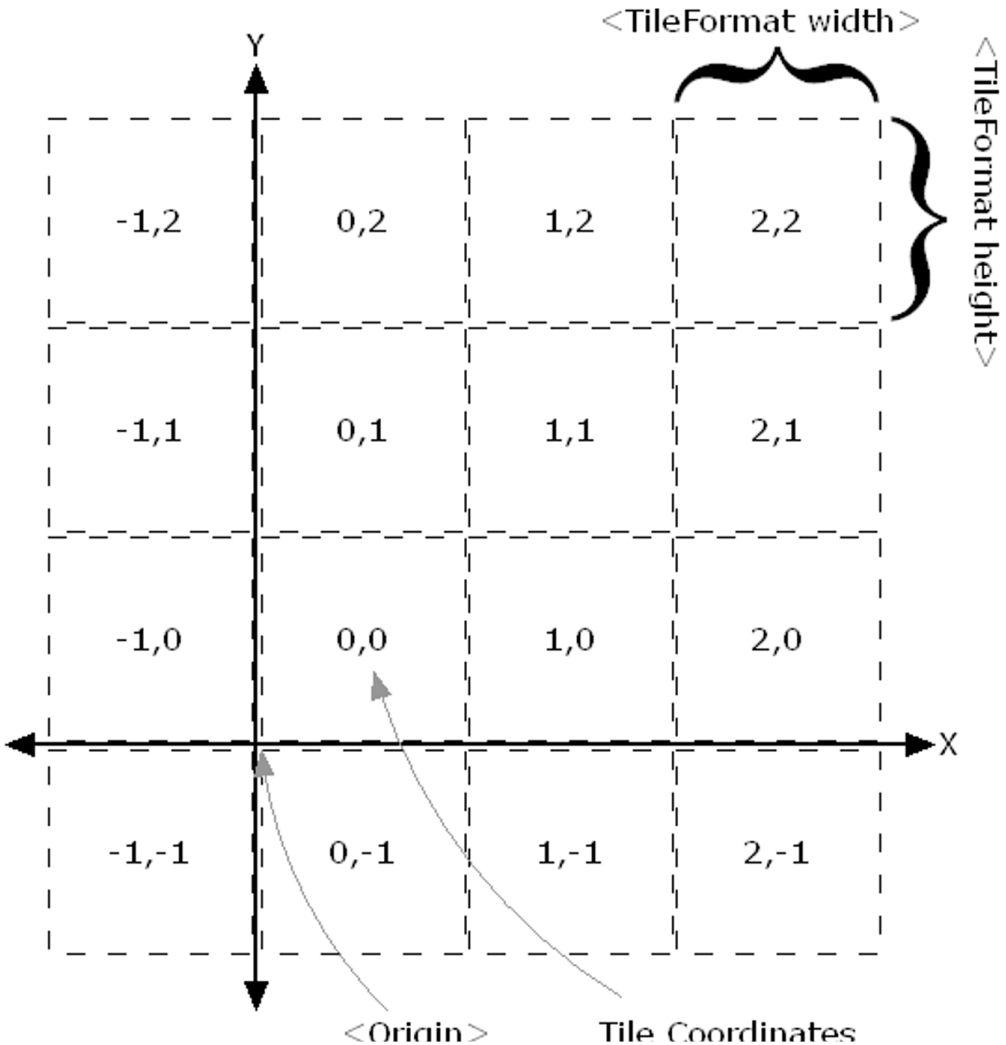
- Many services use REST API(GET Request).
  - https://.../Z/X/Y.Format
  - Z: Zoom Level
  - X: X coordinate
  - Y: Y coordinate
  - Format:
    - Raster image format(png, jpg, webp)
    - Vector data format(pbf, mvt)

GET Request example

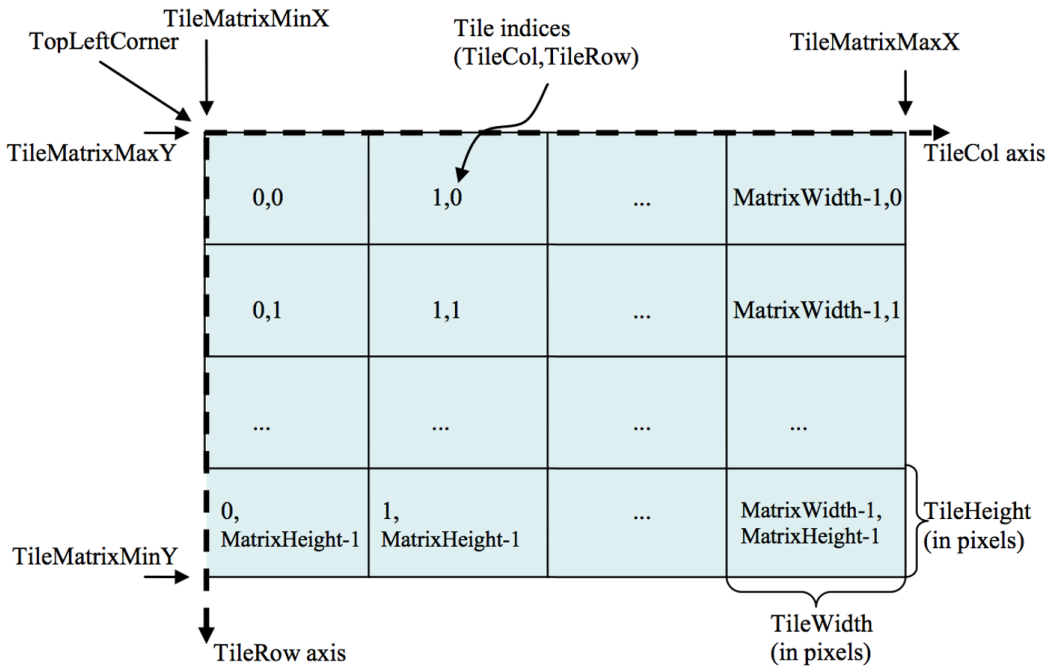
- https://a.tile.openstreetmap.org/3/2/4.png
  - Zoom = 3, X = 2, Y = 4, format = png
  - X and Y coordinates start with 0.

Specification

- Two tile service specifications are popular.
  - Tile Map Service(TMS)
  - Web Map Tile Service(WMTS)
- TMS is simpler than WMTS.
- TMS's X Y coordinate is started from bottom left.
  - Same as Cartesian coordinate system.
- WMTS's X Y coordinate is started from top left.
  - Same as Coordinate system of 2D computer graphics.


<TileFormat width>
<TileFormat height>
Y
X
-1,2
0,2
1,2
2,2
-1,1
0,1
1,1
2,1
-1,0
0,0
1,0
2,0
-1,-1
0,-1
1,-1
2,-1
<Origin>
Tile Coordinates

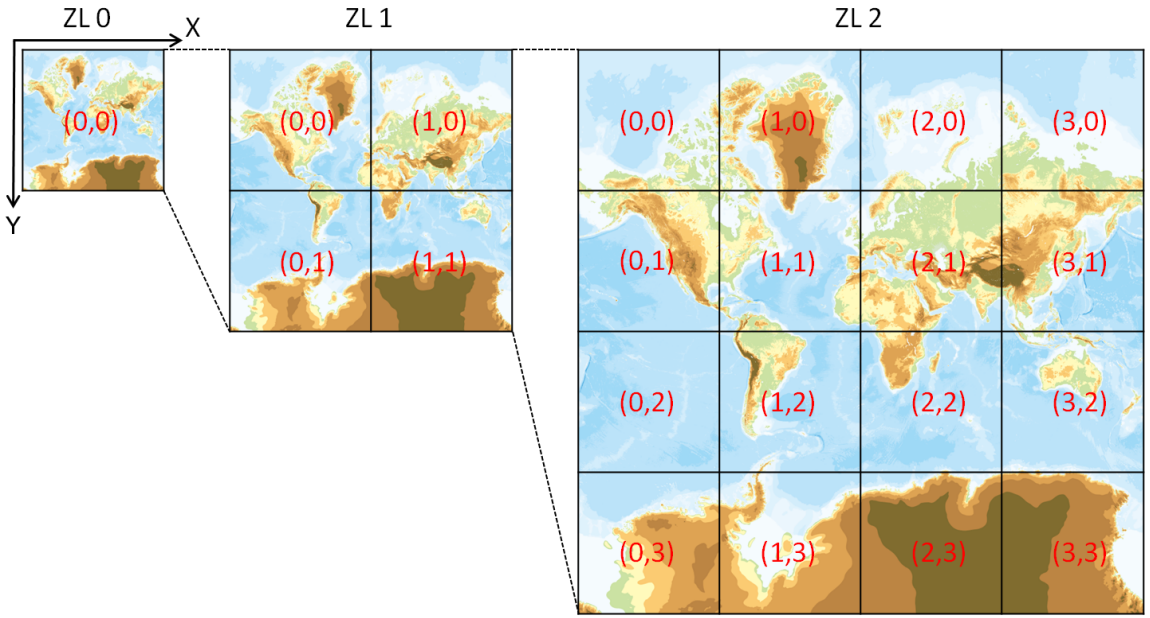
TMS


TopLeftCorner
TileMatrixMinX
Tile indices
(TileCol,TileRow)
TileMatrixMaxX
TileMatrixMaxY
TileCol axis
0,0
1,0
...
MatrixWidth-1,0
0,1
1,1
...
MatrixWidth-1,1
...
...
...
...
0,
MatrixHeight-1
1,
MatrixHeight-1
...
MatrixWidth-1,
MatrixHeight-1
TileHeight
(in pixels)
TileMatrixMinY
TileRow axis
TileWidth
(in pixels)

WMTS

The Y coordinate flipped

- OpenStreetMap use TMS like protocol but Y coordinate is numbered from top.
  - OpenStreetMap call "Slippy Map".
  - We call xyz tile.
    - {z}/{x}/{y}.png
    - Also we call zxy tile.



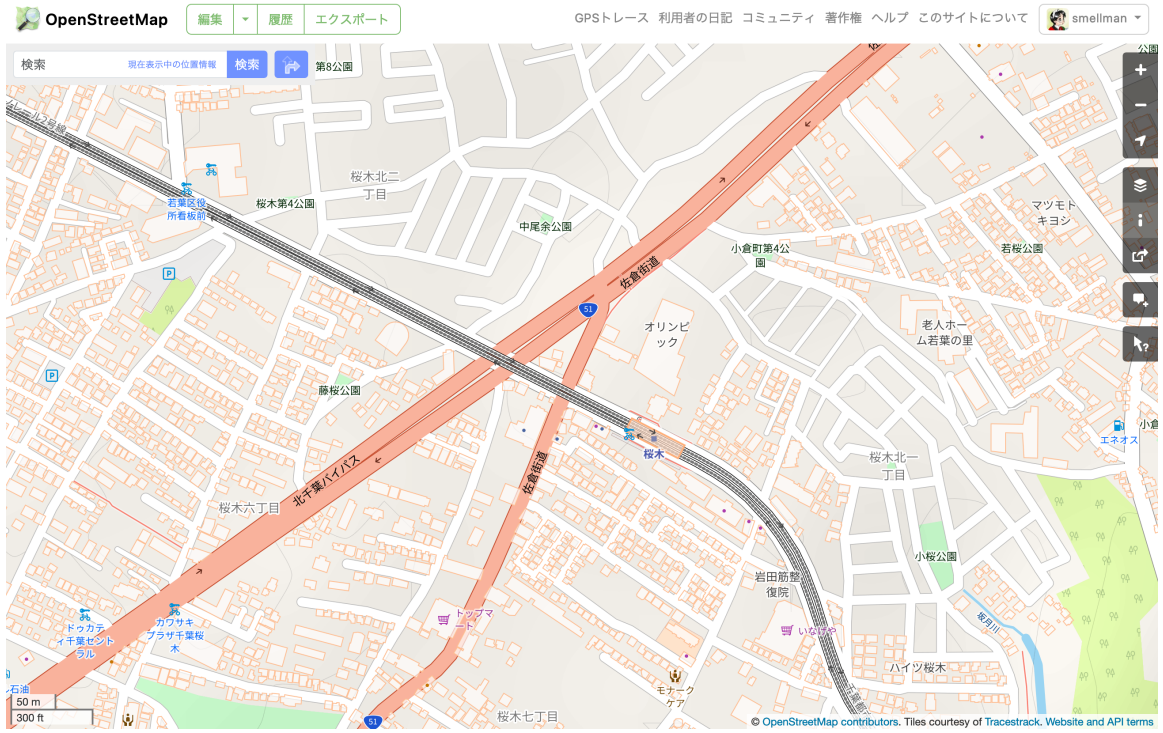
https://maps.gsi.go.jp/help/image/tileNum.png

XYZ tile

- De facto standard of tiled map.
  - Web Mercator projection
  - Y coordinate flipped TMS
  - Provide REST API
    - {z}/{x}/{y}.{format}
  - Anyone provide "Specification"
- Too many libraries support XYZ tile.
  - Leaflet, OpenLayers, Maplibre GL JS, Google Maps API etc.

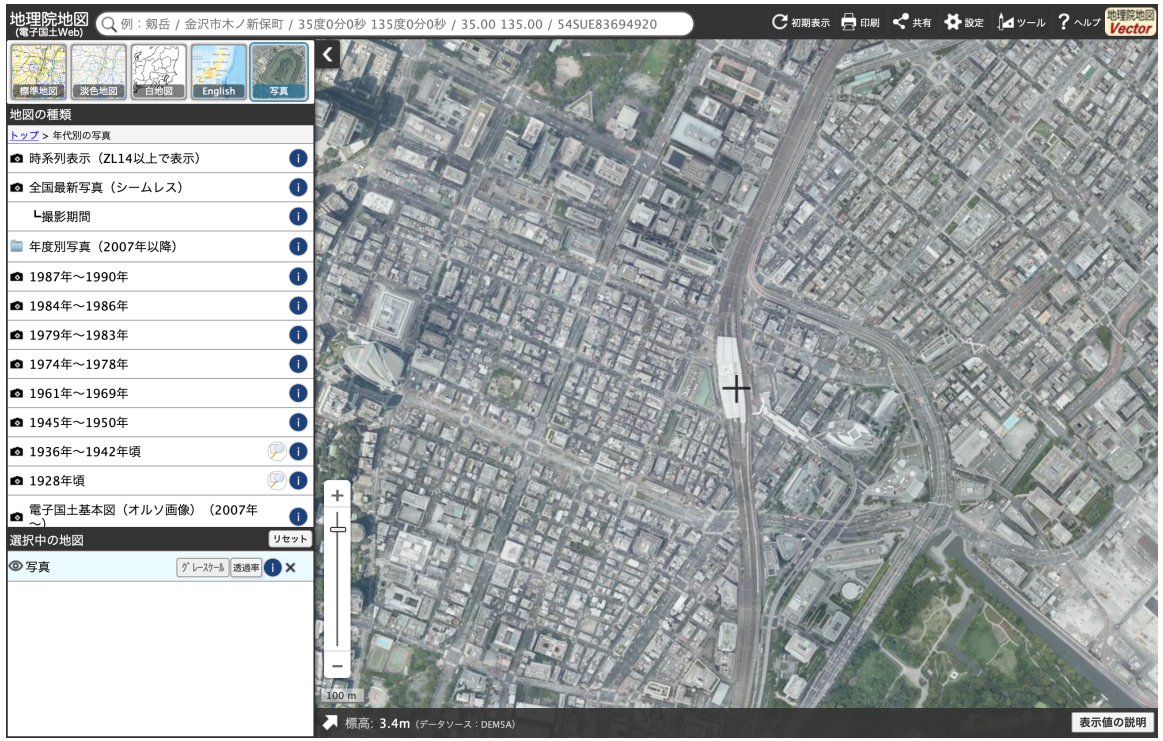
Raster tile (1/3)

- Provides "rendered image"
  - The image doesn't have any "data".
  - Focus to visualization.



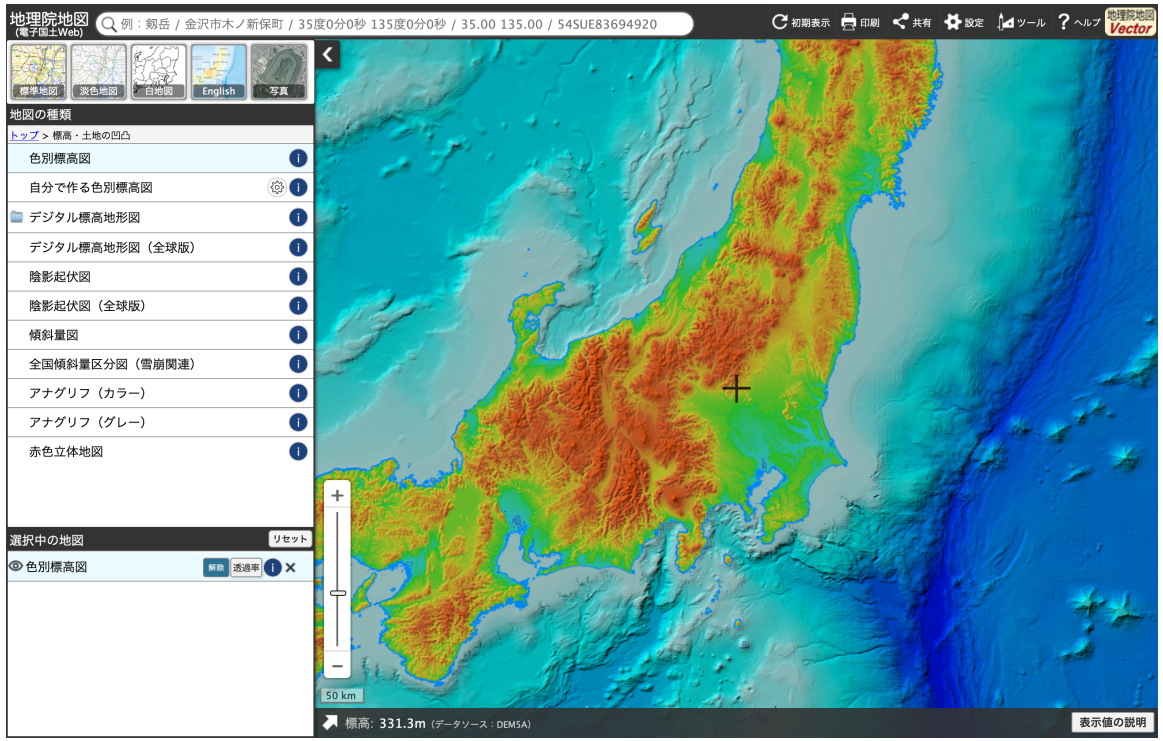
Raster tile (2/3)

- Provides "Satellite images" or "Aerial photograph"
  - Focus to photography.
  - The image doesn't have any "data" too.



Raster tile (3/3)

- Provides "data" as image.
  - Focus to data.
    - Population, Temperature, Rainfall, Elevation, etc.
  - The image has "data" as color.
    - Sample raster tiles contain the elevation value obtainable by calculating with RGB values.



Vector tile (1/2)

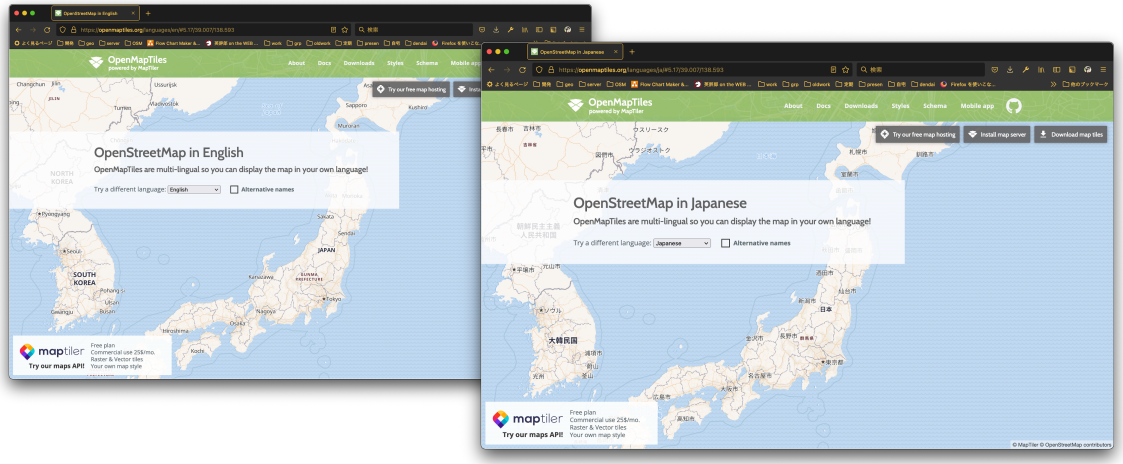
- Provides "Vector data"
  - Each tile contains "Vector data".
    - The tile like a data container.

Vector tile (2/2)

- Vector tile doesn't have a style.
  - The client renders images with style settings.
    - Easy to rotation and bearing.
    - Supports 3D rendering.
- Programmable.
  - The client can change the style dynamically.

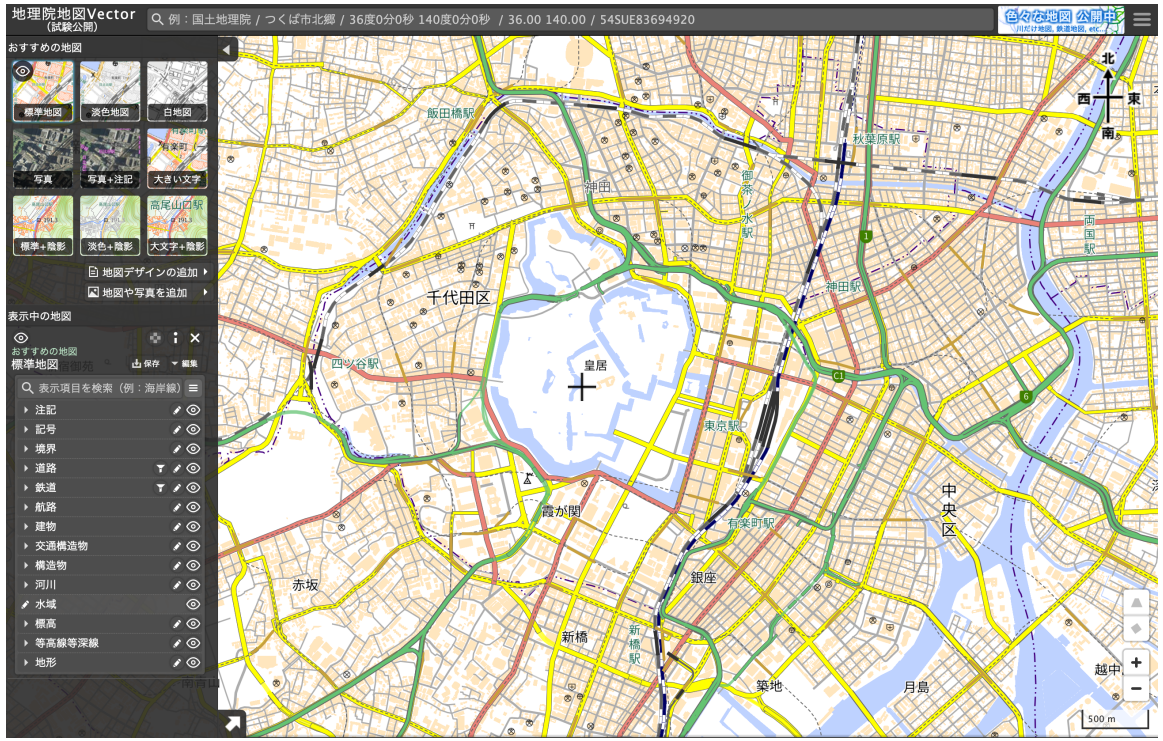
Vector tile example - Multilingual

- https://openmaptiles.org/languages/
  - Enable to change main language dynamically.



Vector tile example - Geospatial Information Authority of Japan

- https://maps.gsi.go.jp/vector/
  - GSI provides vector tile.
  - Enable to change style dynamically.



Mapbox Vector Tile

- De facto standard of vector tile.
  - Vector tile specification by Mapbox Inc.
- Specification
  - A tile encoded by Protocol Buffer format.
  - Desinged for Web Mercator projection.
  - Supports Layers and Features.

https://docs.mapbox.com/data/tilesets/guides/vector-tiles-standards/

Mapbox GL ecosystem and Style Specification

- Mapbox provides Mapbox GL JS(Web), Mapbox GL Native(Smartphone and Desktop application).
  - Mapbox provides specification of styling. https://docs.mapbox.com/mapbox-gl-js/style-spec/

Note: Mapbox GL is proprietary software

- Mapbox GL became proprietary software from end of 2020.
  - Mapbox GL JS is OpenSource software until v1.5.
  - Mapbox GL JS over v2 must require mapbox service’s token.
- MapLibre GL ecosystems are fork of mapbox OpenSource versions.
  - https://maplibre.org/
  - Highly recommend to use MapLibre GL JS now.

Tile support libraries - Javascript

- Leaflet
  - https://leafletjs.com/
  - Lightweight and easy to use.
  - Supports Mapbox Vector Tile with plugin.
- OpenLayers
  - https://openlayers.org/
  - Difficult to use but powerful.
  - Supports Mapbox Vector Tile.
- MapLibre GL JS
  - https://maplibre.org/
  - Easy to use for Mapbox Vector Tile.
  - Supports raster xyz tile too.

Tile support libraries - Android

- MapLibre GL Native
  - https://maplibre.org/
  - Easy to use for Mapbox Vector Tile.
  - Supports raster xyz tile too.
- Google Maps SDK
  - https://developers.google.com/maps/documentation/android-sdk/overview
  - Easy to use for raster xyz tile.

Tile support libraries - iOS

- MapLibre GL Native
  - https://maplibre.org/
  - Easy to use for Mapbox Vector Tile.
  - Supports raster xyz tile too.
- Mapkit
  - https://developer.apple.com/documentation/mapkit
  - Easy to use for raster xyz tile.

Desktop application

- QGIS
  - https://qgis.org/
  - Supports raster xyz tile.
  - Supports Mapbox Vector Tile.

Introduction of software and data in this presentation

Requirements

- This presentation requires Linux based OS.
- Also, you can use Raspberry Pi 4.
  - Raspberry Pi 4 is cheap and powerful.
  - Raspberry Pi 4 is ARM64/aarch64 architecture.
  - Raspberry Pi 4 is easy to use for GIS.
- My repository for this presentation supports only ARM64/aarch64 architecture.

Software - GDAL/OGR

- https://gdal.org/
- GDAL/OGR is the most popular GIS library and provides command line tools.
  - QGIS based on GDAL/OGR.
- GDAL/OGR supports many GIS data formats.
- GDAL/OGR supports raster xyz tile.

Software - Tippecanoe

- https://github.com/felt/tippecanoe/
- Build vector tilesets from large (or small) collections of GeoJSON, FlatGeoBuf or CSV features.
- Tippecanoe is the most popular vector tile builder.

Software - Charites

- Command line tool for writing Mapbox/MapLibre Vector Style Specification in YAML.
  - Organized by The United Nation Vector Tile Toolkit(UNVT).
- Charites convert Style Specification(JSON) to YAML.
  - YAML is easy to read and write for human.
  - YAML is easy to edit for beginners.
- Charites enable to dynamic serving style.

Software - editor

- `nano` is a simple text editor.
  - nano is easy to use for both beginners.
- `vim` is a powerful text editor.
  - vim is difficult to use for beginners.
  - vim is easy to use for experts.

Software - make

- make is a build automation tool.
- make is easy to use for both beginners and experts.
- make is a standard tool of UNIX and Linux.
    - This presentation use make for build and deploy.

Software - nginx

- nginx is a web server.
- nginx is easy to use for both beginners and experts.
- nginx is a standard tool of UNIX and Linux.
    - This presentation use nginx for serving tiles.

Software - tileserver-gl-light

- tileserver-gl-light is a vector tile server.

Software - docker

- docker is a container platform.
- docker is easy to use for both beginners and experts.
- This presentation use docker for serving tiles or running tileserver-gl-light.

Data - Global Map

- Digital geographic information
  - Provided by International Steering Committee for Global Mapping(ISCGM).
  - Composed of 8 Data Sets
    - Vector Data (Transportation, Boundaries, Darainage, Population Centre)
    - Raster Data (Elevation, Vegetation, Land Cover, Land Use)
- Free for non-commercial use.

Global Map - archive

- Archives and website were moved into github by GSI.
  - https://github.com/globalmaps
  - https://globalmaps.github.io/
- Old website was closed
- Some countries provides global map archives at the national site.
  - All links: https://github.com/globalmaps/projectmanagement/blob/master/REPOS.md
- Some links are dead now.

Global map – format

- Vector data provide as Shapefile.
  - It provided as Geography Markup Language (GML) format.
- Raster data provide as GeoTiff file.
  - It provided as Band interleaved by line (BIL) format.

Data – Aerial photograph

- https://www.mlit.go.jp/plateau/
- In Japan, Plateau Project release too many Aerial photograph data.
  - Plateau released PointCloud, 3D data, and Aerial photograph.
  - Aerial photograph is released as GeoTiff data.
    - It is good sample to create raster tile.

Data - OpenStreetMap

- https://www.openstreetmap.org/
- OpenStreetMap is the most popular OpenData.
  - OpenStreetMap provides planet data as PBF format.
- Today's presentation use OpenStreetMap data as sample data.
  - Use small area data for easy to understand.

Data for this presentation

- Global Map Sri Lanka 1.0
  - https://github.com/globalmaps/gmlk10
- Global Map Sri Lanka 2.0
  - https://github.com/globalmaps/gmlk20
- Plateau Higashimurayama City in Tokyo GeoTIFF
  - https://www.geospatial.jp/ckan/dataset/plateau-13213-higashimurayama-shi-2020
- OpenStreetMap data
  - https://tile.opensteetmap.jp/static/planet.pmtiles

How to create your own tiled map

Raster tile processing pattern 1: Global map (One GeoTIFF file)

- Download GeoTIFF file from Global Map archive.
- Enable transparency.
- Convert GeoTIFF to XYZ tile using gdal2tiles.

How to process

```
cd ~/jica_scripts/raster_tile_gm
make fetch # Download GeoTIFF file from Global Map archive.
make transparent # Enable transparency.
make generate_tile # Convert GeoTIFF to XYZ tile using gdal2tiles.
make serve # run nginx
```

How to read Makefile

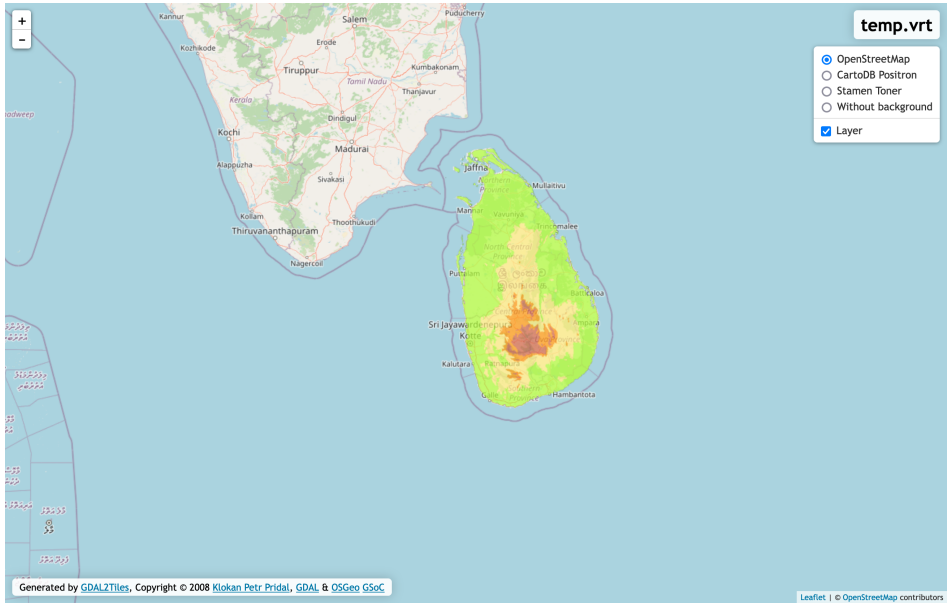
```
fetch:
	git clone https://github.com/globalmaps/gmlk10.git

transparent:
	gdalbuildvrt -srcnodata "0 0 99" el.vrt gmlk10/el.tif

generate_tile:
	gdal_translate -of vrt -expand rgba el.vrt temp.vrt
	gdal2tiles.py --xyz -s EPSG:4326 -z 0-11 temp.vrt

serve:
	docker run -p 8080:80 -v $(PWD)/temp:/usr/share/nginx/html:ro nginx
```

Makefile is simple to run tasks.

```
task_name:
	command
```

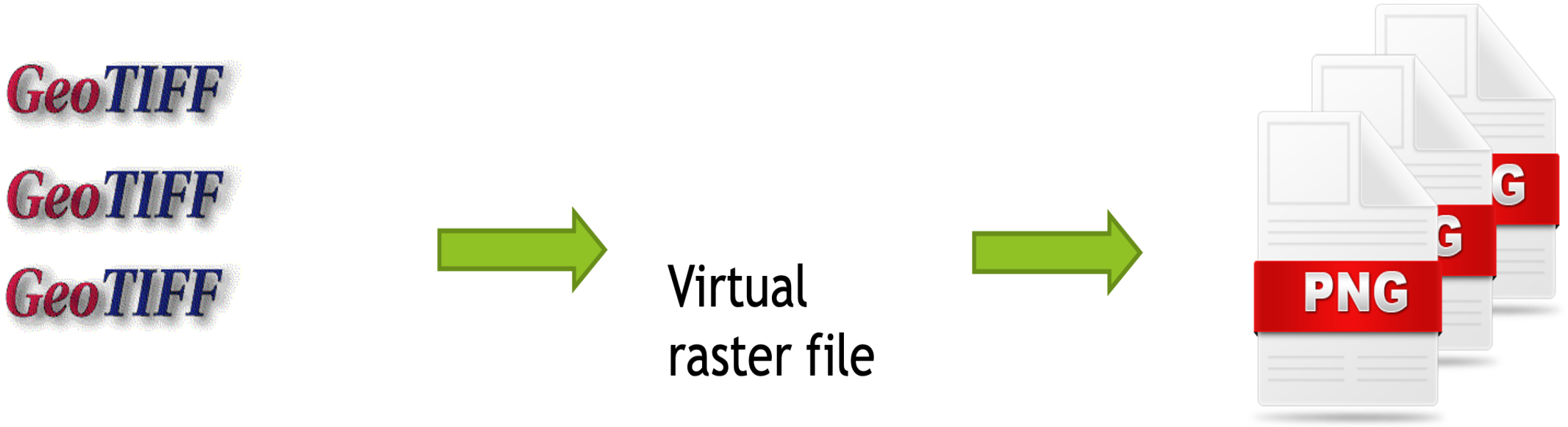
Result

Access to http://<your host>.local:8080/leaflet.html



Raster tile processing pattern 2: Plateau (Many GeoTIFF files)

- Generate VRT file from GeoTIFF files.
- Convert VRT file to XYZ tile using gdal2tiles.

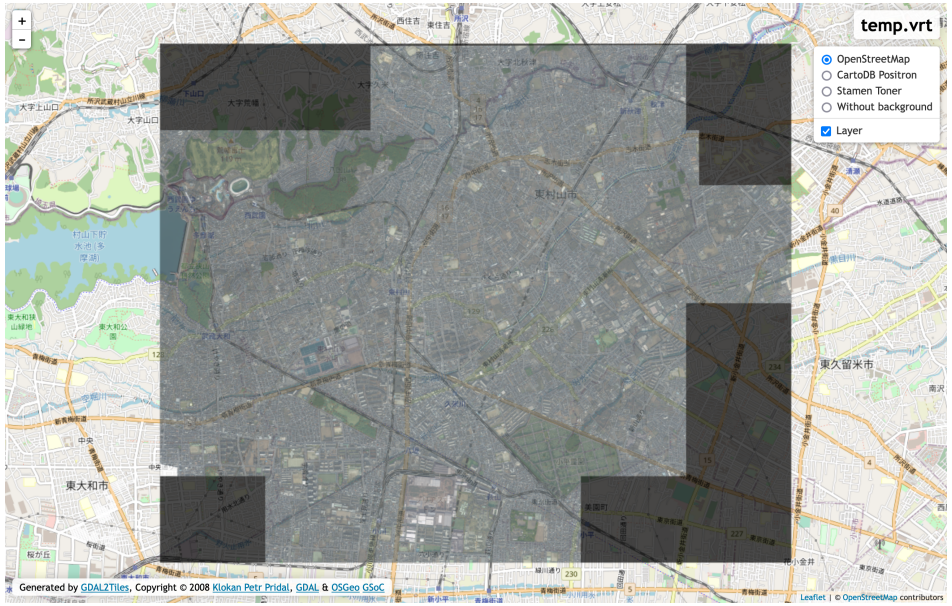


How to process

```
cd ~/jica_scripts/raster_tile_plateau
make fetch # Download GeoTIFF file from Plateau archive and unarchive
make buildvrt # Generate VRT file from GeoTIFF files.
make generate_tile # Convert VRT file to XYZ tile using gdal2tiles.
make serve # run nginx
```

Result

Access to http://<your host>.local:8080/leaflet.html



Vector tile processing pattern: Global map

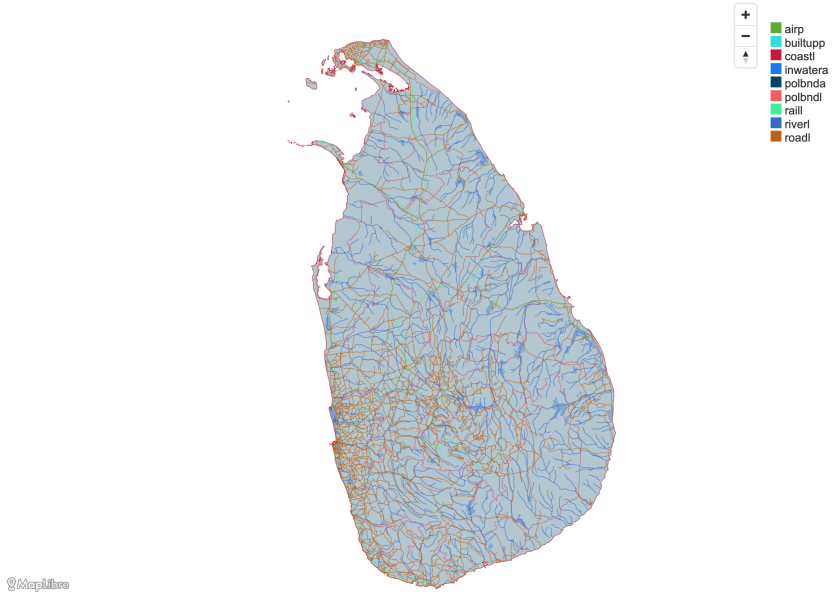
- Download Shapefile file from Global Map archive.
- Convert Shapefile to GeoJSON using ogr2ogr.
- Convert GeoJSON to Mapbox Vector Tile using tippecanoe.

How to process

```
make fetch # Download Shapefile file from Global Map archive.
make convert # Convert Shapefile to GeoJSON using ogr2ogr.
make generate # Convert GeoJSON to Mapbox Vector Tile using tippecanoe.
make tileserver-gl # run tileserver-gl-light
```

Result

Access to http://<your host>.local:8081/



Makefile (1/3)

```
fetch:
	git clone https://github.com/globalmaps/gmlk20.git

convert:
	cd gmlk20; \
	ogr2ogr airp_lka.geojson -s_srs EPSG:4326 -t_srs EPSG:4326 airp_lka.shp; \
	ogr2ogr builtupp_lka.geojson -s_srs EPSG:4326 -t_srs EPSG:4326 builtupp_lka.shp; \
	ogr2ogr coastl_lka.geojson -s_srs EPSG:4326 -t_srs EPSG:4326 coastl_lka.shp; \
	ogr2ogr inwatera_lka.geojson -s_srs EPSG:4326 -t_srs EPSG:4326 inwatera_lka.shp; \
	ogr2ogr polbnda_lka.geojson -s_srs EPSG:4326 -t_srs EPSG:4326 polbnda_lka.shp; \
	ogr2ogr polbndl_lka.geojson -s_srs EPSG:4326 -t_srs EPSG:4326 polbndl_lka.shp; \
	ogr2ogr raill_lka.geojson -s_srs EPSG:4326 -t_srs EPSG:4326 raill_lka.shp; \
	ogr2ogr riverl_lka.geojson -s_srs EPSG:4326 -t_srs EPSG:4326 riverl_lka.shp; \
	ogr2ogr roadl_lka.geojson -s_srs EPSG:4326 -t_srs EPSG:4326 roadl_lka.shp
```

ogr2ogr convert Shapefile to GeoJSON. Notes: Those Shapefiles are not included .prj file.

Makefile (2/3)

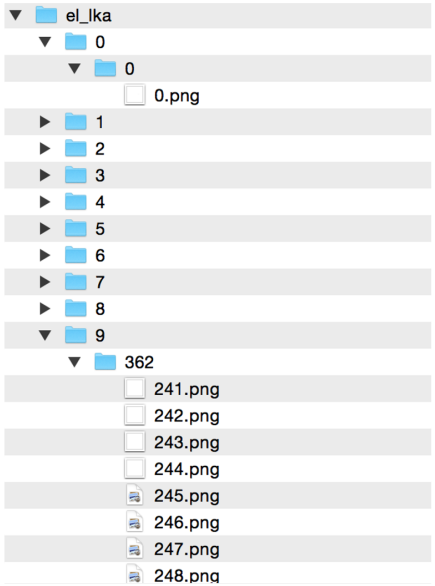
```
generate:
	tippecanoe -o lka.pmtiles \
	  -L airp:gmlk20/airp_lka.geojson \
	  -L builtupp:gmlk20/builtupp_lka.geojson \
	  -L coastl:gmlk20/coastl_lka.geojson \
	  -L inwatera:gmlk20/inwatera_lka.geojson \
	  -L polbnda:gmlk20/polbnda_lka.geojson \
	  -L polbndl:gmlk20/polbndl_lka.geojson \
	  -L raill:gmlk20/raill_lka.geojson \
	  -L riverl:gmlk20/riverl_lka.geojson \
	  -L roadl:gmlk20/roadl_lka.geojson
	tippecanoe -o lka.mbtiles \
	  -L airp:gmlk20/airp_lka.geojson \
	  -L builtupp:gmlk20/builtupp_lka.geojson \
	  -L coastl:gmlk20/coastl_lka.geojson \
	  -L inwatera:gmlk20/inwatera_lka.geojson \
	  -L polbnda:gmlk20/polbnda_lka.geojson \
	  -L polbndl:gmlk20/polbndl_lka.geojson \
	  -L raill:gmlk20/raill_lka.geojson \
	  -L riverl:gmlk20/riverl_lka.geojson \
	  -L roadl:gmlk20/roadl_lka.geojson
```

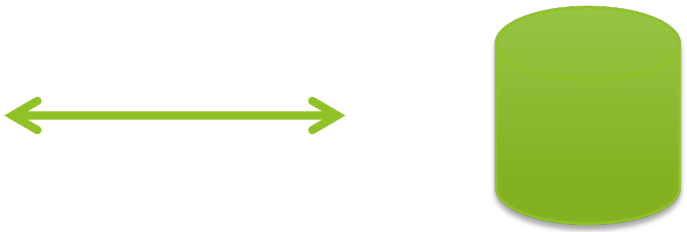
2 outputs

- tippecanoe runs 2 times and generate 2 outputs.
  - .mbtiles file
    - SQLite database file.
    - Contains vector tile.
  - .pmtiles file
    - "Cloud Native" format.
    - You can host .pmtiles as static file.

MBTiles - SQLite database

- MBTiles is container of tile.
  - MBTiles is single file database(SQLite).
  - TMS schema.





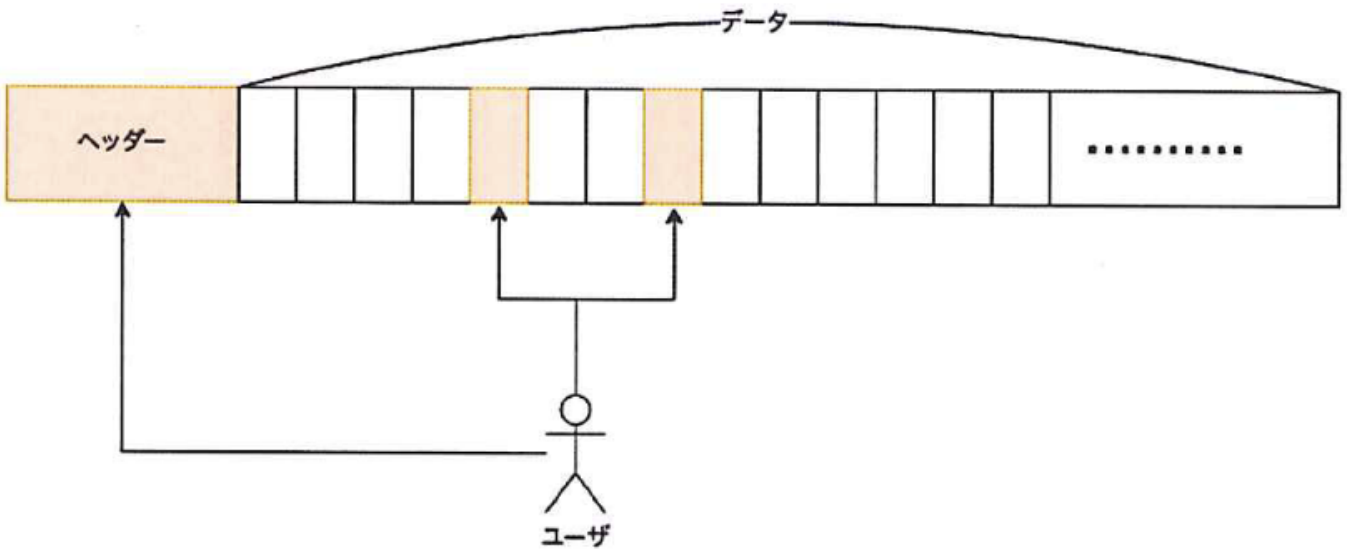
X	Y	Z	blob
0	0	0	(binary)
...	...	...	...
241	362	9	(binary)

Makefile (3/3)

```
tileserver-gl:
	docker run --rm -it -v $(PWD):/data -p 8080:80 \
	  maptiler/tileserver-gl-light \
	  -p 80 --file /data/lka.mbtiles
```

PMTiles - Cloud Native format

- PMTiles is similar to MBTiles.
  - "Cloud Native" format.
  - You can easy to convert mbtiles to pmtiles using `pmtiles` command.



https://smellman.github.io/pmtiles-example/

How to design your own tiled map

How to distribute your own tiled map

copyright