



Cloudoptimierte Formate

Cloudoptimierte Formate für Kacheln und multidimensionale Rasterdaten

Dr. Marco Hugentobler
Sourcepole AG, Zürich
www.sourcepole.ch
<https://twitter.com/sourcepole>





Cloudoptimierte Formate

- › **Klassisch: Web-Service (WMS,WFS,WMTS,...)**
- › **Cloud: Server liefert nur Datei/en aus**
- › **HTTP range request: es muss nicht das ganze File gelesen werden**
- › **Ein grosses File besser als sehr viele kleine Files (z.B. bei S3-Speicher)**
- › **Cloudoptimiertes Format für Tiles: PMTiles**
- › **Cloudoptimiertes Format für Raster: Zarr**



- › Von Protomaps entwickelt
- › Speichert Z/Y/X Tiles (Raster und Vektor)
- › Ähnliches Grundkonzept wie MBTiles
- › Aktuell Version 3
- › <https://github.com/protomaps/PMTiles/blob/main/spec/v3/spec.md>

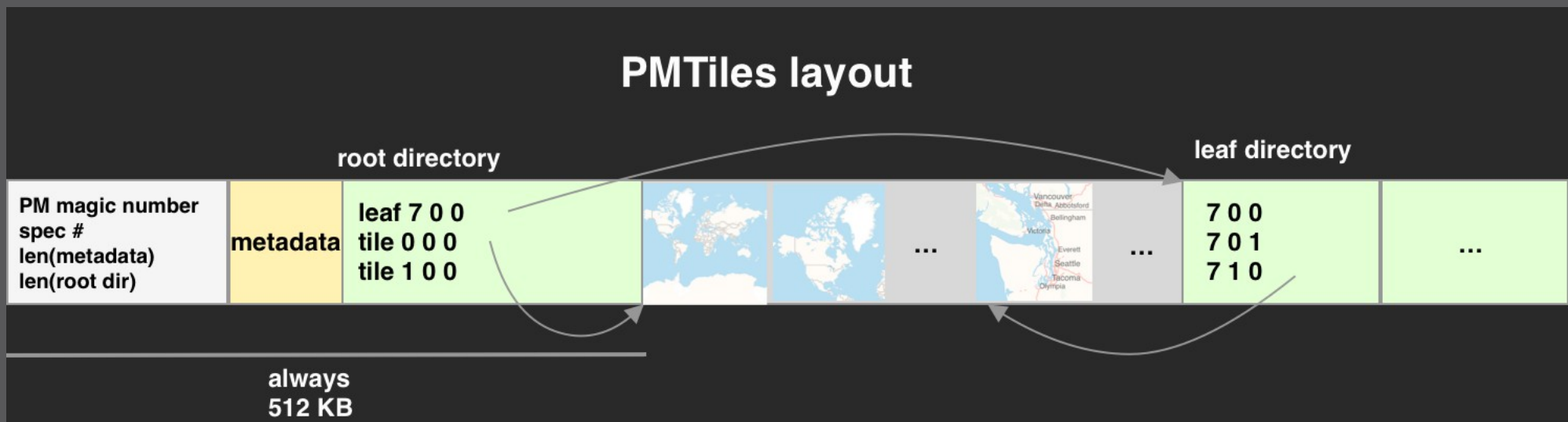


PMTiles Format

- › **MBTiles eignet sich nicht gut für die Cloud:**
 - › Sqlite DB-file
 - › Tabellen 'metadata' und 'tiles'
- › **PMTiles Aufbau:**
 - › Header (127 bytes)
 - › Root directory (Header + Root < 16384 bytes)
 - › JSON metadata
 - › Leaf directories
 - › Tile data



PMTiles Format





PMTiles erzeugen

- › **Aus MBTiles mit dem binary 'pmtiles'**
 - › pmtiles convert input.mbtiles output.pmtiles
 - › Pmtiles upload output.pmtiles s3://xxx
- › **Tippecanoe**
- › **Mit Python**



PMTiles Benutzen

- › **Webviewer zum testen:**
<https://protomaps.github.io/PMTiles/>
- › **Von JavaScript: OpenLayers, Leaflet, MapLibre GL JS**
- › **Martin (Tile Server)**
- › **Libraries für Go und Python**
- › **Direkt aus QGIS leider nicht möglich (mbtiles geht)**



PMTiles Vor- und Nachteile

› Vorteile:

- › In S3-Speicher hochladen effizient weil ein grosses File anstatt viele kleine
- › Client muss zuerst nur den Header laden, nicht die ganze Datei

› Nachteile:

- › Nur EPSG:3857
- › Fixes Tile-Schema, keine tile matrix sets wie WMTS
- › HTTP range request nicht überall unterstützt
- › Keine Kompression bei HTTP range requests
- › Kein update einzelner Kacheln möglich
- › Client muss im Minimum 512 KB herunterladen (V3 nur noch 16KB)



Zarr Format

- › OGC community standard
- › Ursprünglich kein GeofORMAT. CRS kann in Attribut gespeichert werden (z.B. GDAL)
- › V2, V3 und GeoZarr ist in Arbeit
- › <https://zarr.readthedocs.io/en/stable/>
- › Format für mehrdimensionale Raster
- › Hierarchisch, aufgeteilt in chunks, Inhalt komprimiert
- › Kann von Python aus gut benutzt werden



Zarr Format

- › **Gruppen für hierarchische Gliederung**
- › **Eine Gruppe kann Arrays und Untergruppen haben**
- › **Array hat Metadaten**
- › **Array kann beliebig viele Dimensionen haben**
- › **Array ist in Chunks aufgeteilt**
- › **Komprimierung für Array-Daten**
- › **Verschiedene Speicherformate (Filesystem, zip, S3, ...) und Kompressionen**



› Mit Python

› Mit GDAL

- › gdal_translate

- › gdalmdimtranslate (Multidimensional data model) : `gdalmdimtranslate -of Zarr infile.nc outfile.zarr -co ARRAY:COMPRESS=GZIP`



› Mit GDAL

- › <https://gdal.org/drivers/raster/zarr.html>
- › `gdalinfo file.zarr`
- › `gdalmdiminfo file.zarr`
- › Netzwerkdaten mit `vsicurl`, `vsis3`, ...

› Mit Python

› Mit anderen Programmiersprachen (C, C++, Rust, JS, Java)

› In QGIS via GDAL



Benutzung mit Python

<https://zarr.readthedocs.io/en/stable/tutorial.html>

```
import zarr
store = zarr.DirectoryStore('data/group.zarr')
root_grp = zarr.group(store, overwrite=True)
sub_grp = root_grp.create_group('foo')
a = sub_grp.create_dataset('variable1', shape=(20, 20), chunks=(10, 10))
a[:] = 42
a.attrs['comment'] = 'answer to life, the universe and everything'
sub_grp2 = sub_grp.create_group('bar')
b = sub_grp2.create_dataset('variable2', shape=(30,30),
chunks=(10,10))
b[:] = 1
```



Benutzung in QGIS mit GDAL

The screenshot shows the QGIS interface with a dialog box titled "Hinzuzufügende Elemente wählen | group". The dialog box is open over a map area that is mostly black with a red rectangle. The "Layer" panel on the left shows a group containing two layers: "/foo/variable1" and "/foo/bar/variable2". The "Browser" panel at the bottom left shows "Favoriten" and "Räumliche Lesezeichne".

The dialog box contains the following elements:

- Path: </home/marco/geodaten/zarr/data/group.zarr>
- Suche... (Search input field)
- Element list:
 - /foo/bar/variable2
 - /foo/variable1
- Buttons: "Alle wählen" and "Alle abwählen"
- Options section:
 - Layer hinzufügen
 - Abbrechen



Vor- und Nachteile

- › Flexible Struktur, Vor- und Nachteil
- › Noch keine einheitliche Angabe CRS / Georeferenzierung, kommt in GeoZarr
- › Verwendung aus dem Desktop-GIS mit GDAL (z.B. `/vsicurl/http://localhost/group.zarr`)



**Vielen Dank für die
Aufmerksamkeit**



Marco Hugentobler
marco.hugentobler@sourcepole.ch