



Kanton Basel-Stadt

Erfahrungen Basel-Stadt

Erfahrungen von Cloudformaten in Basel-Stadt

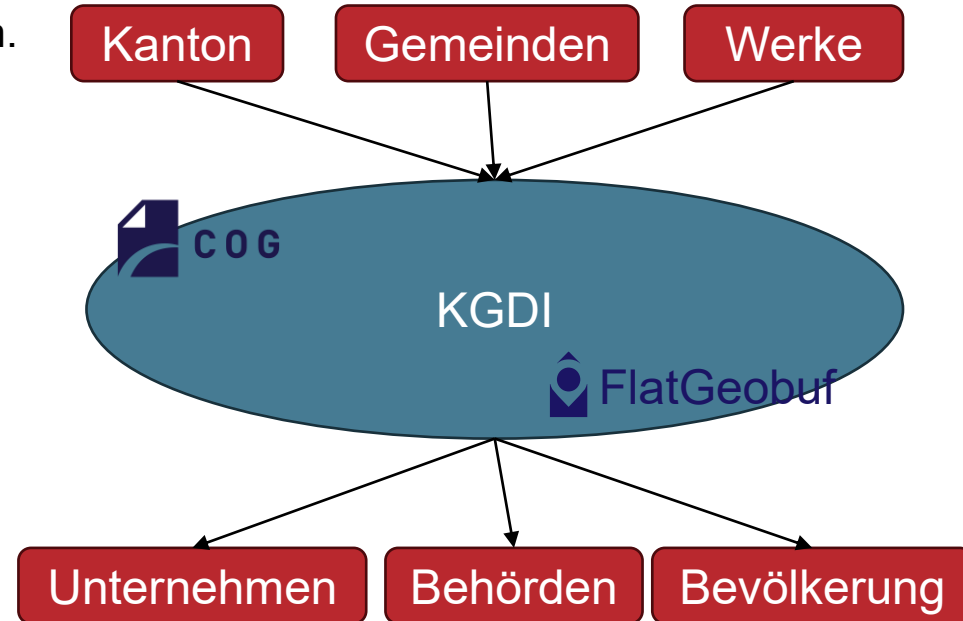
Loris Hubler 03.09.2026

Inhalt der Präsentation

- Potential von Cloudformaten in KGDI
- Einsatz von Cloudformaten in Basel-Stadt
 - Rasterdaten
 - Vektordaten
- Erfahrungen und Herausforderungen
- Zukunftspotential

Potential von Cloudformaten in KGDI

- Die KGDI ist die zentrale Drehscheibe für den Austausch kantonalen Geodaten. Daten könnten auch als Cloud Formate zur Verfügung gestellt werden.
- Cloud-Formate ermöglichen die direkte Nutzung von Daten über das Web – ohne vorgängigen Download oder klassische Schnittstelle.
- Einbinden direkt in Workflow oder Schnittstelle.



Cloudformate in Basel-Stadt

- FlatGeobuf als weiteres Format.
 - Shop und STAC
- Rasterdaten als Cloud-optimiert GeoTIFF:
 - Shop
 - Ist rückwärts kompatibel!
- **Eigentlich stellen wir unsere Geodaten noch nicht so bereit, wie sie heute sinnvoll genutzt werden könnten.**

Produkt: Zonenplan Stadt Basel

Format

☐ Autodesk dxg (Version 2000) ▼

☐ Esri Shapefile

☐ File-Geodatabase

☒ FlatGeobuf

☐ Geopackage

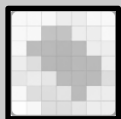
☐ GeoJSON (Koordinatensystem WGS84)

Einsatz von Cloudformaten intern

- Cloud Formate dienen als Grundlage für das Bereitstellen von bestehenden Diensten (WMS).
- Macht ETLs bei Dateiaustausch schneller durch performantes Schreiben und Lesen.



Wie erstellen wir Cloudformate

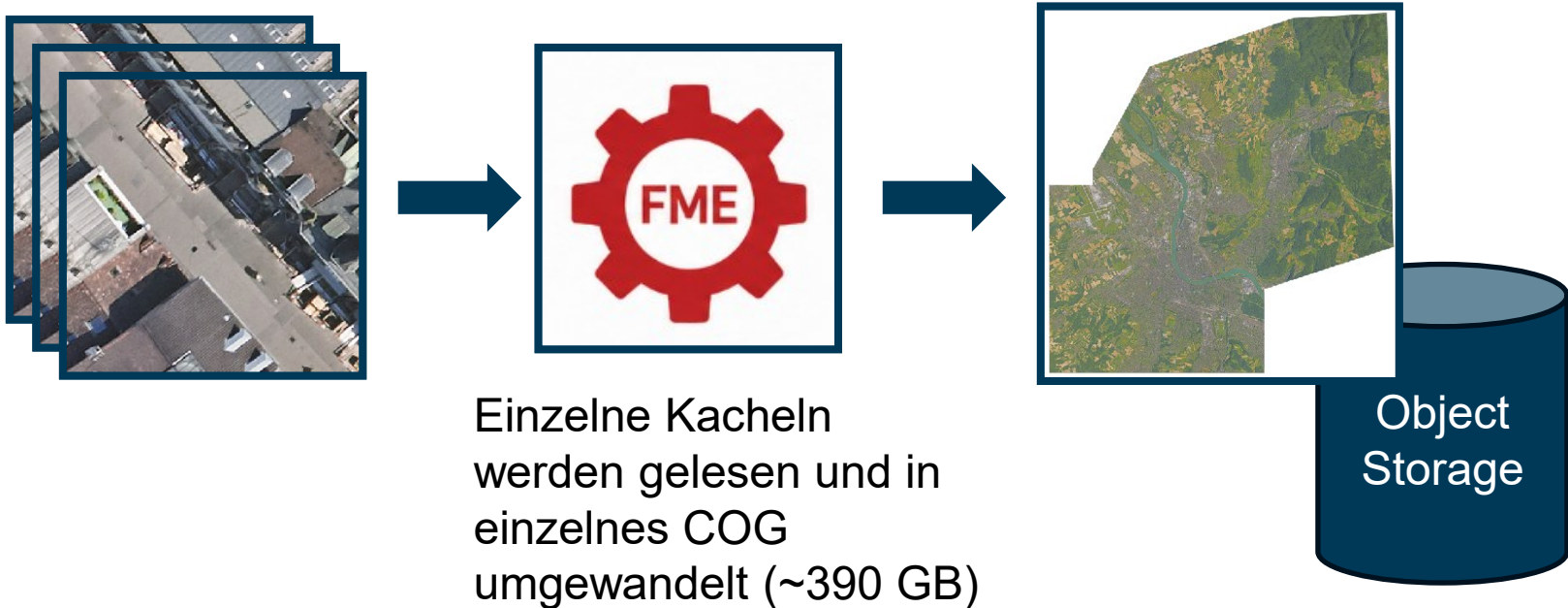


Cloud optimized GeoTIFF → GDAL / FME



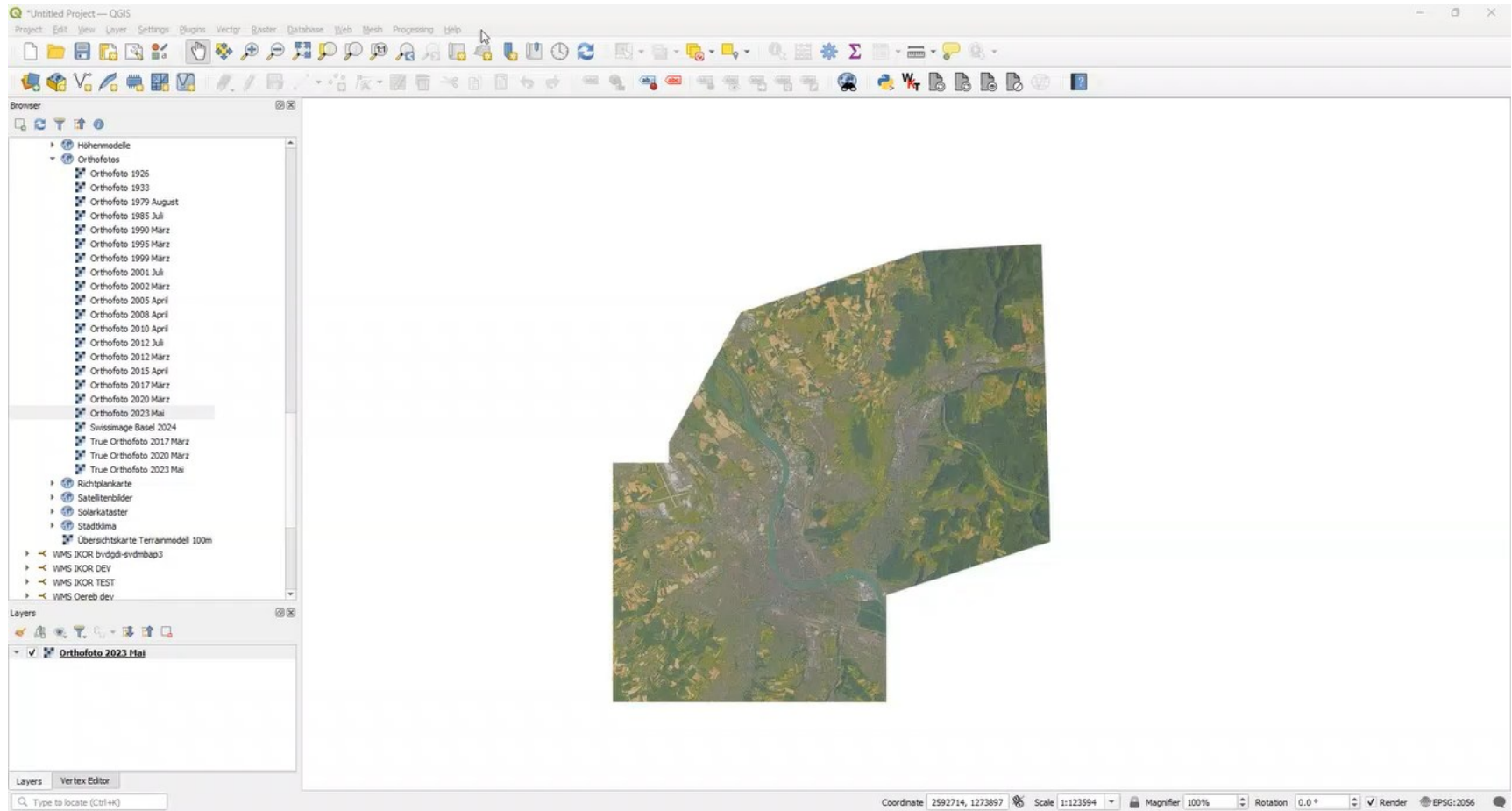
GeoParquet / FlatgeoBuf → GDAL / FME

COGs für WMS-Dienst Aufbereitung



COGs für WMS-Dienst





Vektordaten und WMS?

- Funktioniert auch für Vektordaten
- QGIS-Server mit FlatGeobuf als Quelle in Betrieb
 - Wir Speichern QGIS-Projekt zusammen mit Daten als Artefakt → Versionierung von Daten mit Darstellung
- Achtung: Problematik beim Austausch der Datei. Datei Metadaten sind relevant und werden nicht immer geladen.

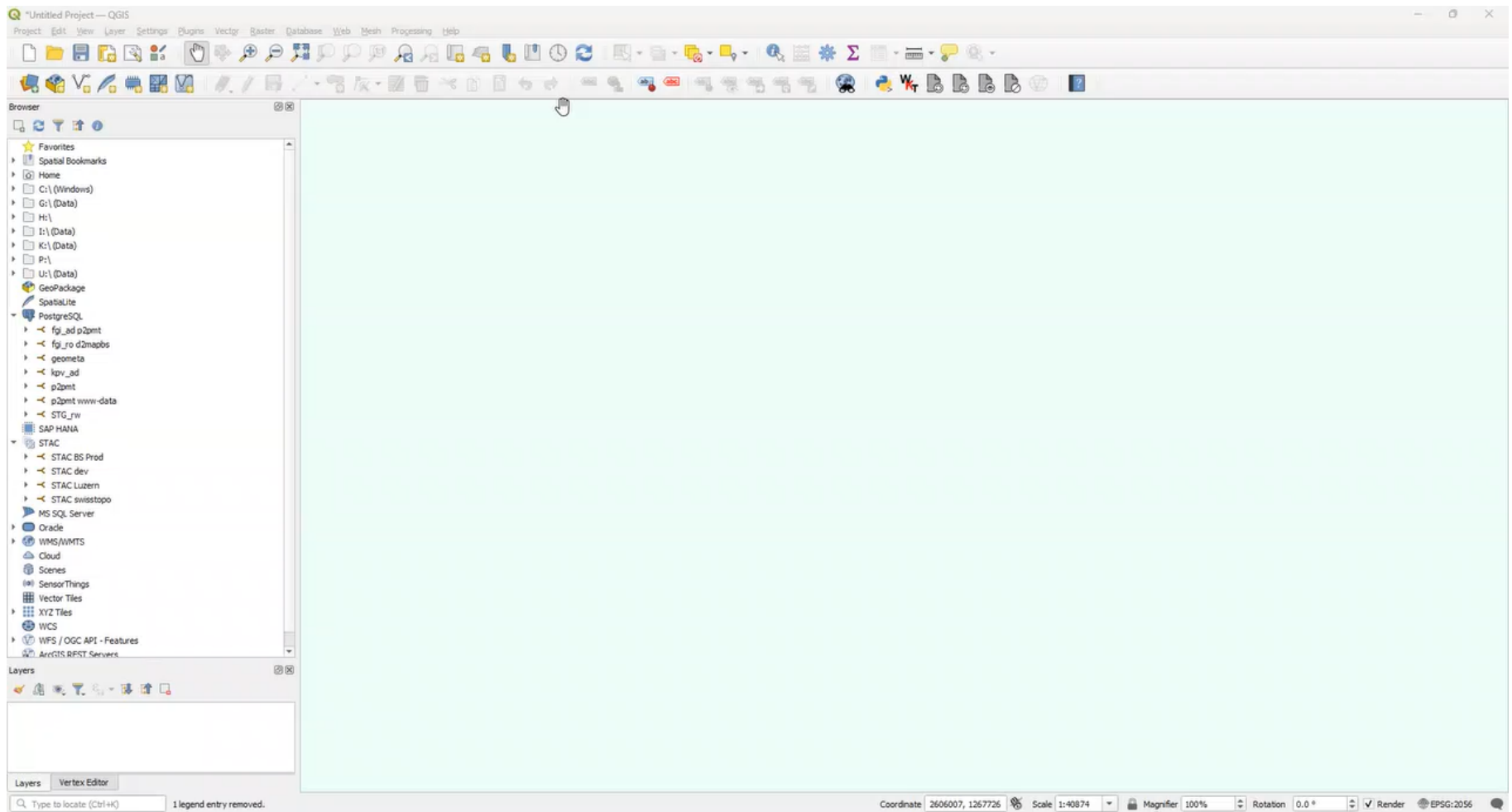
```
FlatGeobuf (native) 0.008s
Shapefile (native)  0.010s
FlatGeobuf (OGR)    0.013s
Shapefile (OGR)     0.023s
GeoPackage (OGR)    0.042s
Spatialite (OGR)    0.045s
PostGIS (native)    0.053s
GeoJSON (OGR)       0.089s
```

Quelle:
<https://mapserver.org/input/vector/flatgeobuf.html>

Allgemeine Erfahrungen (Positiv)

- Cloud Optimized GeoTIFF
- Geometrien bei GeoParquet/FlatGeobuf sind Well-Known Binary (WKB)
- GDAL unterstützt Formate (/vsi3/ oder /vsicurl/).
- Schnelles lesen und schreiben → Gut für Austausch.
- Viele Vektor Features speichern (benötigt wenig Speicher)
- Cloud Optimized GeoTiff ist OGC-Standard
- GeoParquet und FlatGeobuf sind auf dem Weg zu OGC-Standard





Allgemeine Erfahrungen (Negativ/Herausforderungen)

- Einige Formate wie GeoParquet sind noch sehr neu und benötigt Treiber, welche teilweise installiert werden müssen.
- Noch wenig Know-how vorhanden
- In einem KGDI-Setup eher Zusatz zu bestehende Infrastruktur: Unsere Nutzer benötigen aktuell einen WMS.
- Wir stellen die Cloudformate noch nicht so zur Verfügung, dass sie sinnvoll genutzt werden können (ZIP)



Wo sind Cloudformate ungeeignet

- Datenerfassung → Datenbank
- Dynamische Daten → Datenbank
- Lesen von einzelnen Zeilen → Datenbank

Zukünftiges Potential und Ausblick

- PMTiles für Vektortiles
- Einfachere Skalierbarkeit (Einfacher eine Datei zur Verfügung stellen wie ein Dienst)
- Ein Datensatz, verschiedene Anwendungen: Derselbe Datenbestand kann Grundlage für Download, Visualisierung, Analyse und Weiterverarbeitung → Einfachere Infrastruktur
- IT-Projekte mit wenig Infrastruktur
- Direkte Einbindung von Cloud Formaten in WebGIS.
- Direkte Analyse von dezentralen Daten mit SQL (z.B. mit DuckDB)

Habt ihr Erfahrung mit Cloud Formaten

Fragen

- Hat jemand schon Cloudformate benutzt?
- Hat jemand Cloudformate im produktiven Einsatz?

Weitere Inputs

- Möchte jemand weitere Erfahrungen teilen?



Weitere führende Links:

- Cloud-Optimized Geospatial Formats Guide <https://guide.cloudnativegeo.org/>
- An Exploration of 'Cloud-Native Vector': <https://cholmes.medium.com/an-overview-of-cloud-native-vector-c223845638e0>
- <https://spatialworkflow.io/geoparquet-columnar-storage-explained>
- <https://anitagraser.com/2024/11/23/geoparquet-in-qgis-smaller-faster-files-for-the-win/>