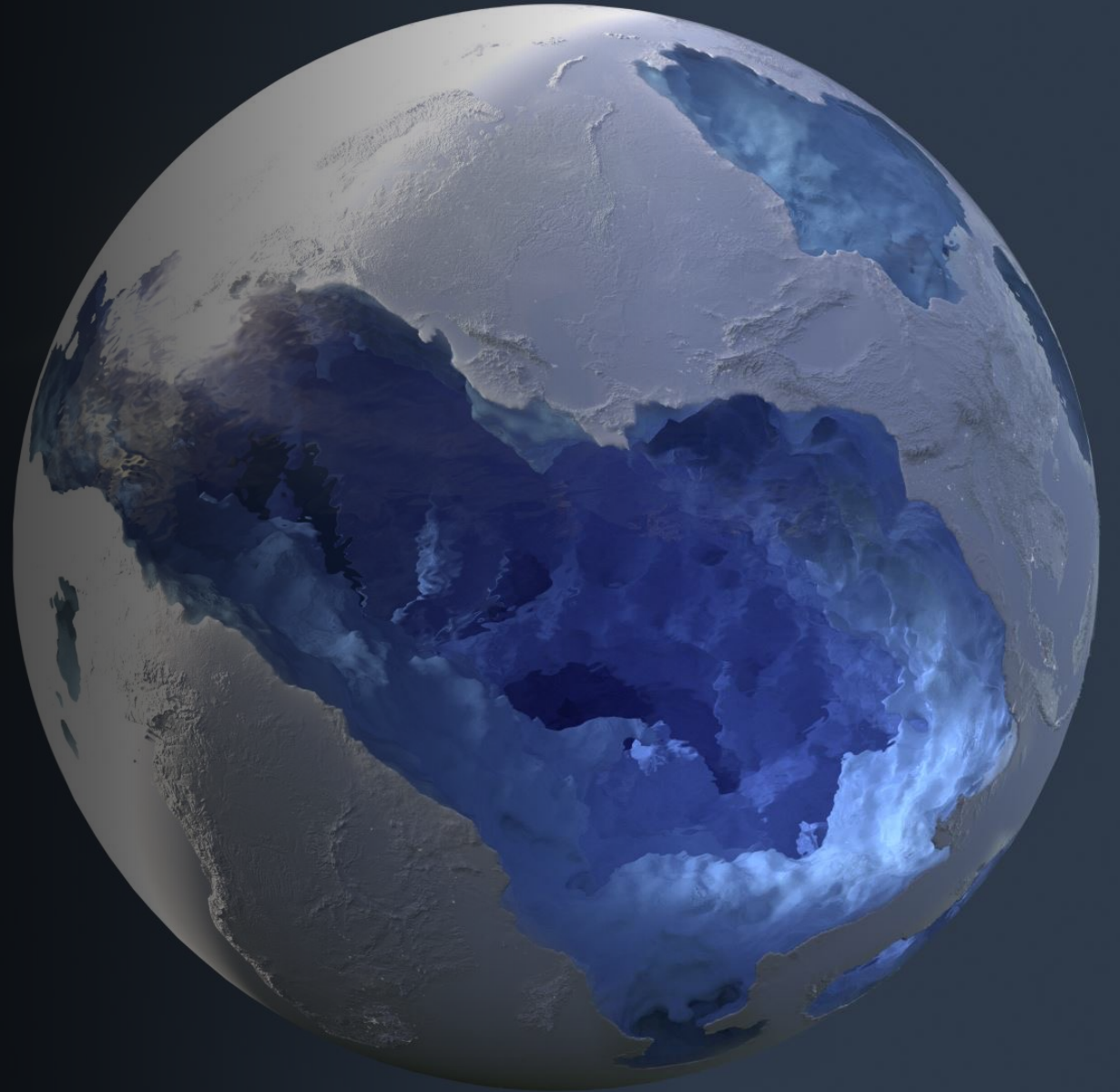# COMTiles: a case study of a cloud optimized tile archive format for deploying planet-scale tilesets in the cloud

FOSS4G 2023 - Academic Track

Markus Tremmel

COMTiles Requirements

- Support of different coordinate reference systems (CRS)

- Minimize the transferred amount of data and the number of requests to reduce costs and latency

- Every tile in the archive can be requested with at most one additional request

- Fast decoding of the index

COMTiles Requirements

- Support of different coordinate reference systems (CRS)

- Minimize the transferred amount of data and the number of requests to reduce costs and latency

- Every tile in the archive can be requested with at most one additional request
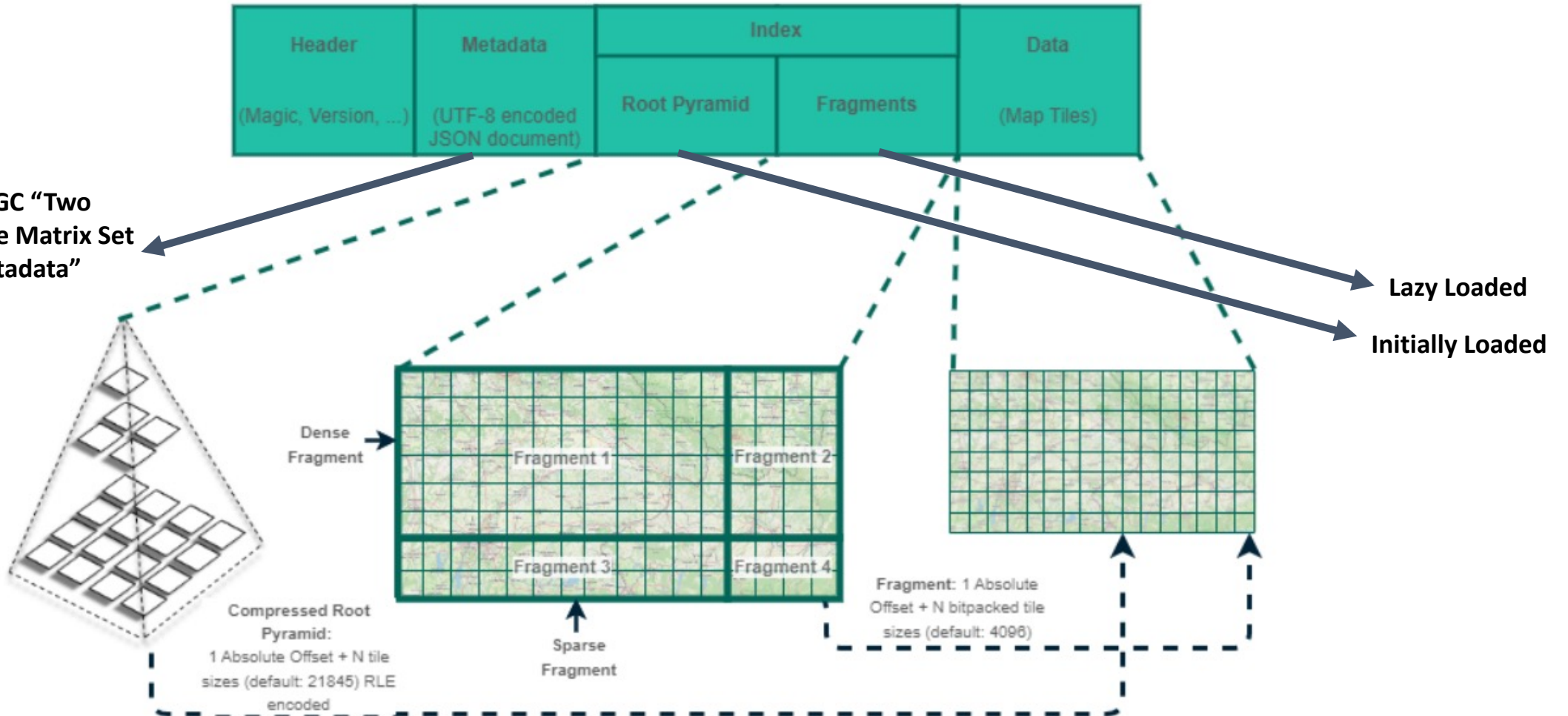
- Fast decoding of the index

With the cost of a larger total index size

COMTiles Layout
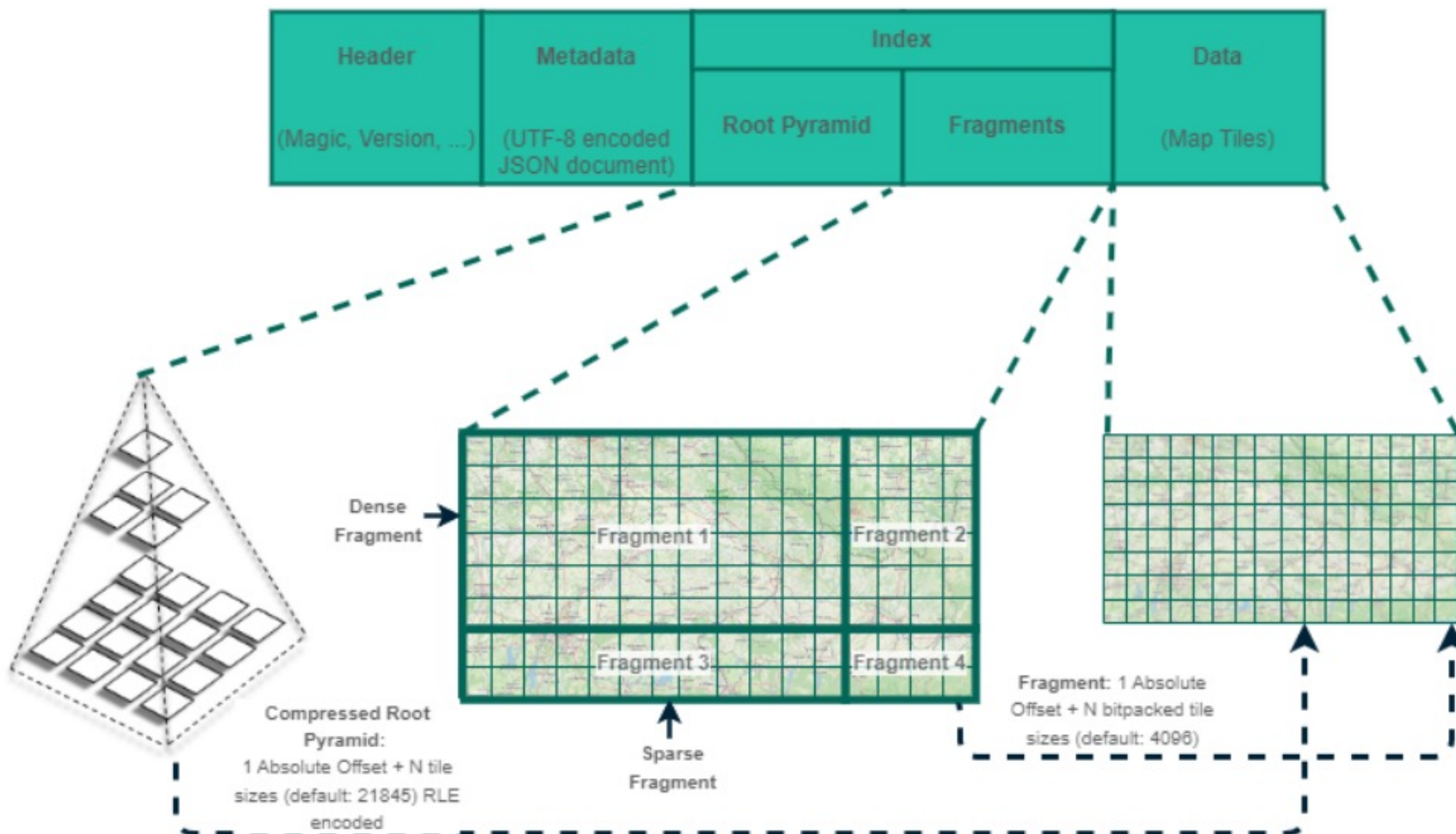
COMTiles Layout

Based on the OGC "Two Dimensional Tile Matrix Set and Tile Set Metadata" specification

Comparison with PMTiles

- COMTiles is compared to PMTiles for planet scale tilesets based on the following metrics, as they have a significant impact on user experience and cloud access charges
  - Size of downloaded data from a cloud object storage
  - Number of requests to a cloud object storage
  - Performance for decoding portions of the index

Comparison with PMTiles

- COMTiles is compared to PMTiles for planet scale tilesets based on the following metrics, as they have a significant impact on user experience and cloud access charges
  - **Size of downloaded data from a cloud object storage**
  - **Number of requests to a cloud object storage**

    Two different realistic map interactions have been automatic simulated

  - Performance for decoding portions of the index

Focus on zoom based interactions (Netek et al., 2020)

Focus on a panning based map navigation pattern

Test Case	Format	Transferred data	Requests
1	COMTiles	1.0	1.0
	PMTiles	3.29	1.13
2	COMTiles	1.0	1.10
	PMTiles	3.1	1.0

Reference: https://github.com/mactrem/com-tiles-evaluation

Comparison with PMTiles

- COMTiles is compared to PMTiles for planet scale tilesets based on the following metrics, as they have a significant impact on user experience and cloud access charges
  - Size of downloaded data from a cloud object storage
  - Number of requests to a cloud object storage
  - **Performance for decoding portions of the index**

Format	Num Entries	Decoding Performance
COMTiles fragment	4096	1
PMTiles country-scale	4096	19
PMTiles planet-scale	16384	63

Reference: https://github.com/mactrem/com-tiles-evaluation

Batching Tile Requests

- The individual tile requests can be batched to improve performance (for HTTP/1.1 requests) and in particular to reduce the cloud acces charges

- In the test (Netek et al., 2020) the number of tile requests was reduced by approximately 77%

- This enables the deployment of a planet-scale OSM tileset on a Cloudflare R2 storage and access to approximately 35 million tiles a remarkably low cost of only $1.35 per month

Reference: https://github.com/mactrem/com-tiles-evaluation
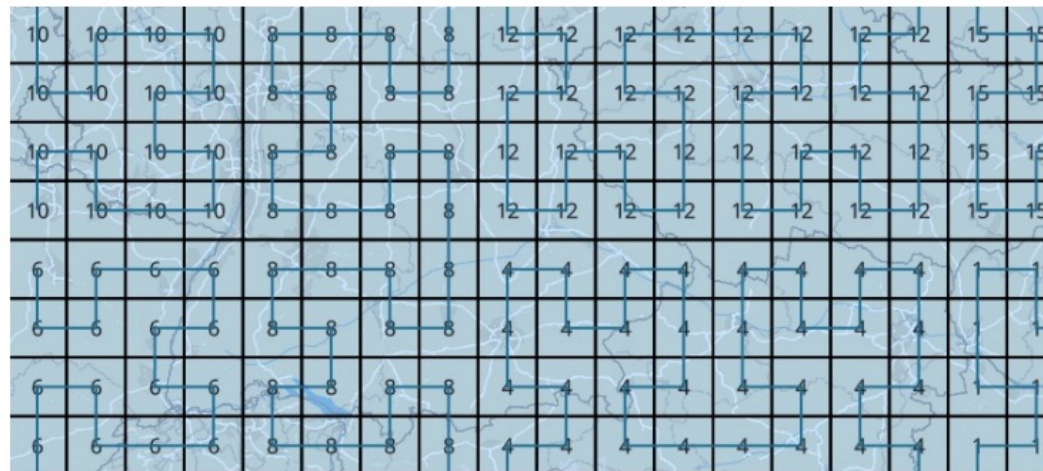
Future Work

- Future work may include further reduction of the index size while continuing to meet the requirements

- One approach could be to use a Bitvector encoding to reduce the index size of sparse tilesets since over 50% of an OSM vector tileset are not present

- This could be inspired by the implicit tiling extension of the 3DTiles spec that contains an availability section to efficiently encode sparse dataset