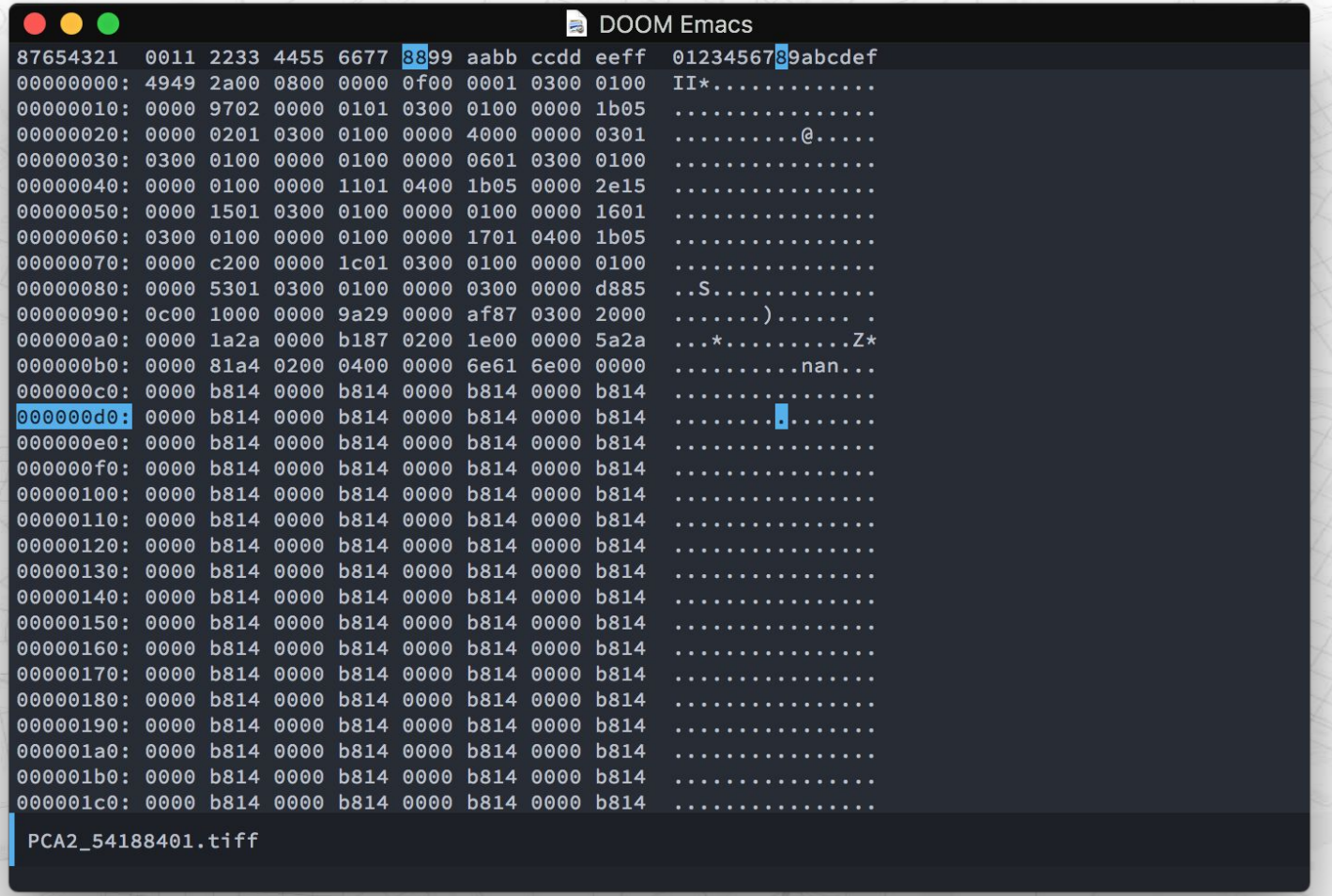
COG

CLOUD OPTIMIZED GEOTIFF

Enabling Efficient Cloud Workflows


DOOM Emacs
87654321 0011 2233 4455 6677 8899 aabb ccdd eeff 0123456789abcdef
00000000: 4949 2a00 0800 0000 0f00 0001 0300 0100 II*.............
00000010: 0000 9702 0000 0101 0300 0100 0000 1b05 ....
00000020: 0000 0201 0300 0100 0000 4000 0000 0301 ....@.....
00000030: 0300 0100 0000 0100 0000 0601 0300 0100 ....
00000040: 0000 0100 0000 1101 0400 1b05 0000 2e15 ....
00000050: 0000 1501 0300 0100 0000 0100 0000 1601 ....
00000060: 0300 0100 0000 0100 0000 1701 0400 1b05 ....
00000070: 0000 c200 0000 1c01 0300 0100 0000 0100 ....
00000080: 0000 5301 0300 0100 0000 0300 0000 d885 ..S.............
00000090: 0c00 1000 0000 9a29 0000 af87 0300 2000 ....)...... .
000000a0: 0000 1a2a 0000 b187 0200 1e00 0000 5a2a ...*..........Z*
000000b0: 0000 81a4 0200 0400 0000 6e61 6e00 0000 ....nan...
000000c0: 0000 b814 0000 b814 0000 b814 0000 b814 ....
000000d0: 0000 b814 0000 b814 0000 b814 0000 b814 ....
000000e0: 0000 b814 0000 b814 0000 b814 0000 b814 ....
000000f0: 0000 b814 0000 b814 0000 b814 0000 b814 ....
00000100: 0000 b814 0000 b814 0000 b814 0000 b814 ....
00000110: 0000 b814 0000 b814 0000 b814 0000 b814 ....
00000120: 0000 b814 0000 b814 0000 b814 0000 b814 ....
00000130: 0000 b814 0000 b814 0000 b814 0000 b814 ....
00000140: 0000 b814 0000 b814 0000 b814 0000 b814 ....
00000150: 0000 b814 0000 b814 0000 b814 0000 b814 ....
00000160: 0000 b814 0000 b814 0000 b814 0000 b814 ....
00000170: 0000 b814 0000 b814 0000 b814 0000 b814 ....
00000180: 0000 b814 0000 b814 0000 b814 0000 b814 ....
00000190: 0000 b814 0000 b814 0000 b814 0000 b814 ....
000001a0: 0000 b814 0000 b814 0000 b814 0000 b814 ....
000001b0: 0000 b814 0000 b814 0000 b814 0000 b814 ....
000001c0: 0000 b814 0000 b814 0000 b814 0000 b814 ....
PCA2_54188401.tiff

Eugene Cheipesh

echeipesh@azavea.com
GitHub: @echeipesh
www.azavea.com

Cloud Native Workflows

“Hey, let's not download the whole file every time.”

Cloud Native

- Network is between storage and computation
- Storage is chunked and probably Key/Value
- Computation must scale
- Coordination is limited and risky

COG

CLOUD OPTIMIZED GEOTIFF

Cloud Optimized GeoTiff

- Defines an internal structure of GeoTiffs that is optimized for streaming reading from blob storage like S3 or Google Cloud Storage.
- That is, it's best utilized for requests that use **HTTP GET Range Requests**

Cloud optimized GeoTIFF

Definition

A cloud optimized GeoTIFF is a regular GeoTIFF file, aimed at being hosted on a HTTP file server, whose internal organization is friendly for consumption by clients issuing HTTP GET range request ("bytes: start_offset-end_offset" HTTP header).

It contains at its beginning the metadata of the full resolution imagery, followed by the optional presence of overview metadata, and finally the imagery itself. To make it friendly with streaming and progressive rendering, we recommand starting with the imagery of the smallest overview and finishing with the imagery of the full resolution level.

More formally, the structure of such a file is:

- TIFF / BigTIFF signature
- IFD (Image File Directory) of full resolution image
- Values of TIFF tags that don't fit inline in the IFD directory, such as TileOffsets?, TileByteCounts? and GeoTIFF keys
- Optional: IFD (Image File Directory) of first overview (typically subsampled by a factor of 2), followed by the values of its tags that don't fit inline
- Optional: IFD (Image File Directory) of second overview (typically subsampled by a factor of 4), followed by the values of its tags that don't fit inline
- ...
- Optional: IFD (Image File Directory) of last overview (typically subsampled by a factor of 2^N), followed by the values of its tags that don't fit inline
- Optional: tile content of last overview level
- ...
- Optional: tile content of first overview level
- Tile content of full resolution image.

Unspecified points

- size of tile: 256 or 512 pixels are typical however
- compression methods allowed. typically, no compression, DEFLATE or LZW can be used for lossless, or JPEG for lossy. (note that DEFLATE while more efficient than LZW can

How to generate it with GDAL

Given an input dataset in tif with already generated internal or external overviews, a cloud optimized GeoTIFF can be generated with:

https://trac.osgeo.org/gdal/wiki/CloudOptimizedGeoTIFF

Cloud Optimized GeoTIFF

An imagery format for cloud-native geospatial processing

Introduction | Why COG? | How it works | Get Started | Implementations

About

A Cloud Optimized GeoTIFF (COG) is a regular GeoTIFF file, aimed at being hosted on a HTTP file server, with an internal organization that enables more efficient workflows on the cloud. It does this by leveraging the ability of clients issuing HTTP GET range requests to ask for just the parts of a file they need.

Learn More

http://www.cogeo.org/

GeoTiff

"A GeoTIFF file is a TIFF 6.0 file, and inherits the file structure as described in the corresponding portion of the TIFF spec. All GeoTIFF specific information is encoded in several additional reserved TIFF tags, and contains no private Image File Directories (IFD's), binary structures or other private information invisible to standard TIFF readers."

http://geotiff.maptools.org/spec/geotiff2.4.html#2.4

TIFF™

Revision 6.0

Final — June 3, 1992

TIFF

- Tagged Image File Format
- Binary image is separated out into “**segments**”
- Contains metadata in a sequence of “**tags**”
- Can contain metadata for **multiple images** in different “image file directories” (IDFs)

GeoTiff Custom Tags

- GeoTiff adds custom tag GeoKeys
  - non-TIFF keys for CRS map space etc.
- This is how we know where the pixels are on the surface of the earth (the geospatial bit).

http://geotiff.maptools.org/spec/geotiff6.html#6.1

GeoTiff Streaming Read

- **Step 1**: Fetch the header
- **Step 2**: Decide what segments you want
- **Step 3**: Read the segments

Internet Engineering Task Force (IETF)
Request for Comments: 7233
Obsoletes: 2616
Category: Standards Track
ISSN: 2070-1721

R. Fielding, Ed.
Adobe
Y. Lafon, Ed.
W3C
J. Reschke, Ed.
greenbytes

June 2014

Hypertext Transfer Protocol (HTTP/1.1): Range Requests

Abstract

The Hypertext Transfer Protocol (HTTP) is a stateless application-level protocol for distributed, collaborative, hypertext information systems. This document defines range requests and the rules for constructing and combining responses to those requests.

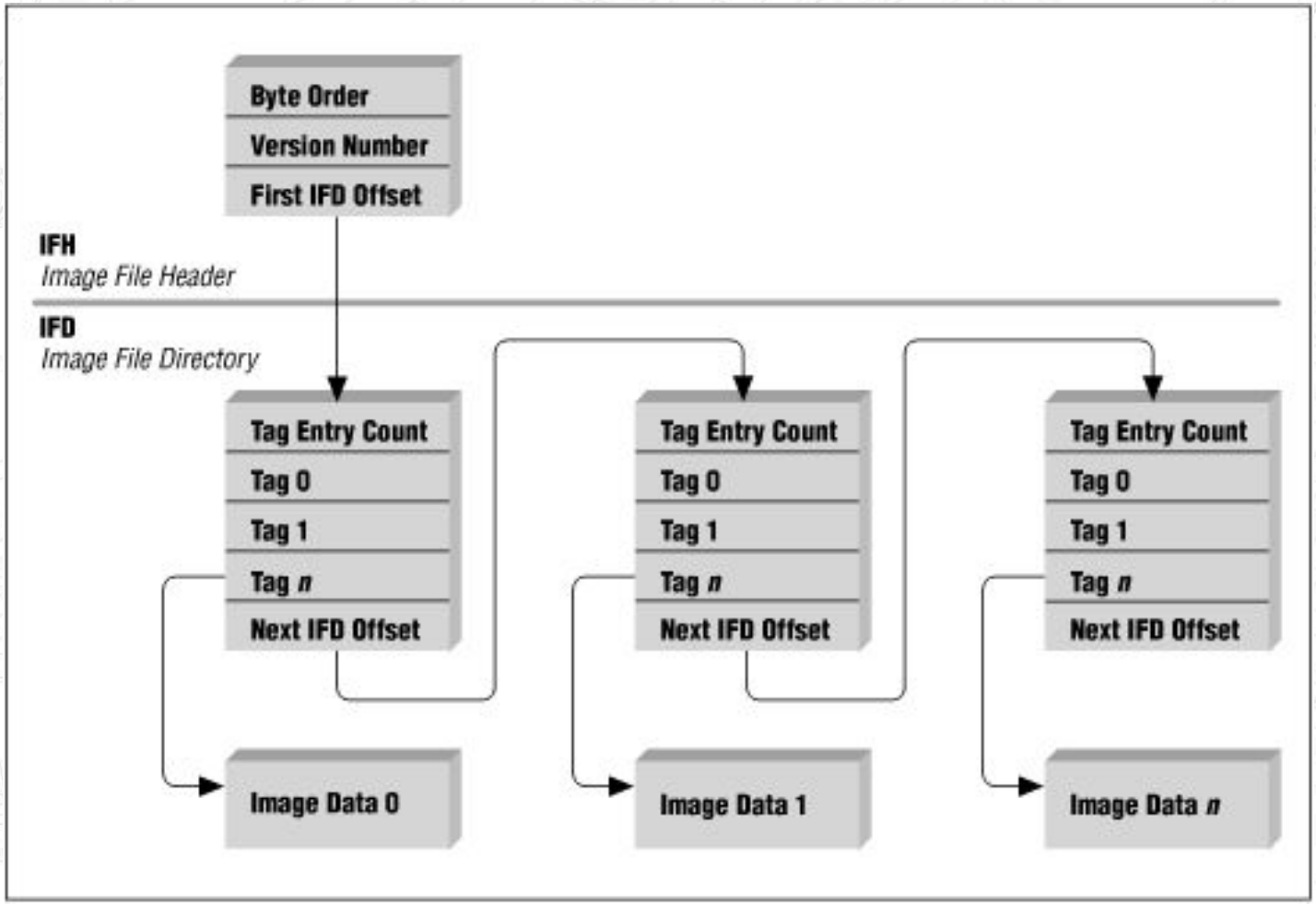
Status of This Memo

This is an Internet Standards Track document.

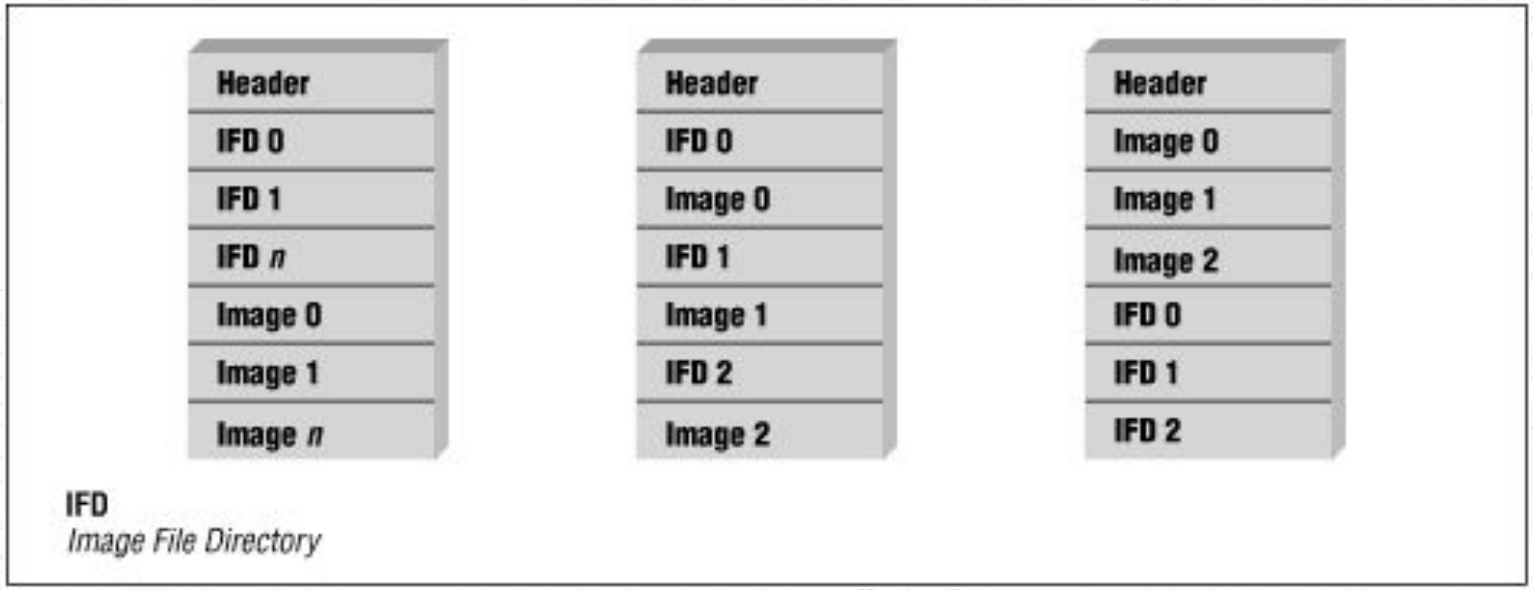
This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7233.

Request: BBOX Mean


Byte Order
Version Number
First IFD Offset
IFH
Image File Header
IFD
Image File Directory
Tag Entry Count
Tag 0
Tag 1
Tag n
Next IFD Offset
Image Data 0
Tag Entry Count
Tag 0
Tag 1
Tag n
Next IFD Offset
Image Data 1
Tag Entry Count
Tag 0
Tag 1
Tag n
Next IFD Offset
Image Data n

GeoTIFF: All of the above

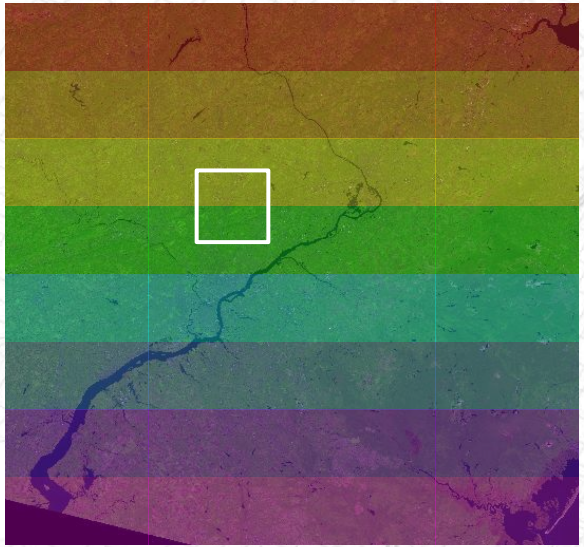


COG: IFDs First - Rule #1

TIFF Segments

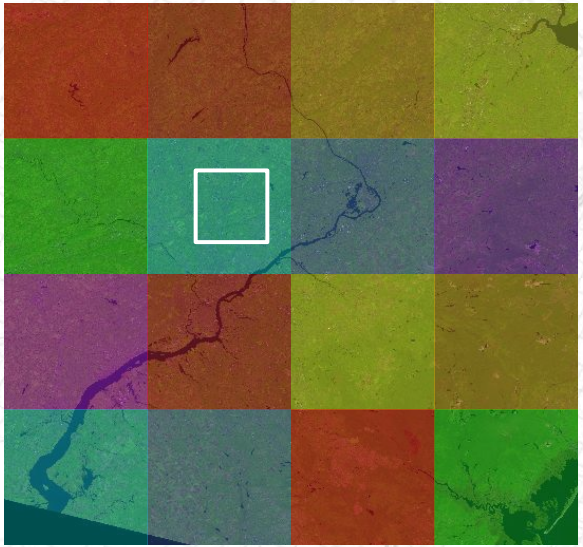
- Chunks of an image are stored at different locations. This allows for parts of the image to be **decompressed** and loaded separately.
- There are two methods for segment layout: **Striped** and **Tiled**

TIFF: Strip Segments



COG: Strip Segments - Bad

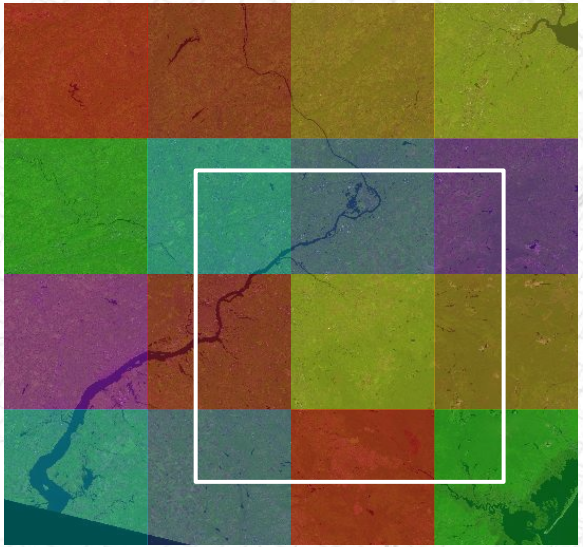
TIFF: Tile Segments



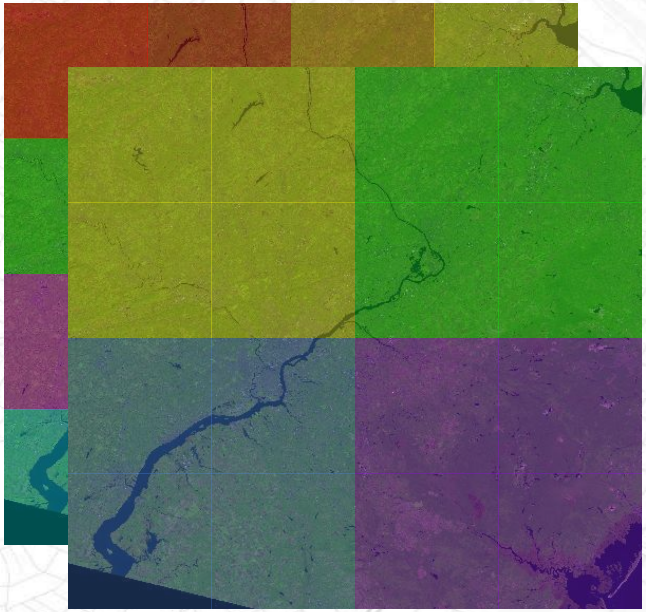
COG: Use Tiled Segments - Rule #2

Request: BBOX Mean at 60m

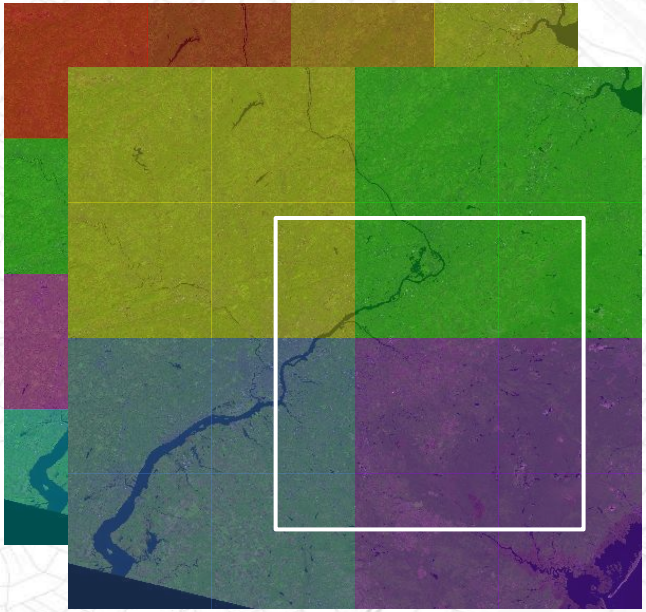
Request: BBOX Mean at 60m



GeoTIFF: Has Overviews



COG: Provide Overviews - Rule #3



COG Rules

- Put IFDs first
- Sort IFDs by resolution
- Use Tiled segments
- Include overviews (in file)

COG Rules Continued ?

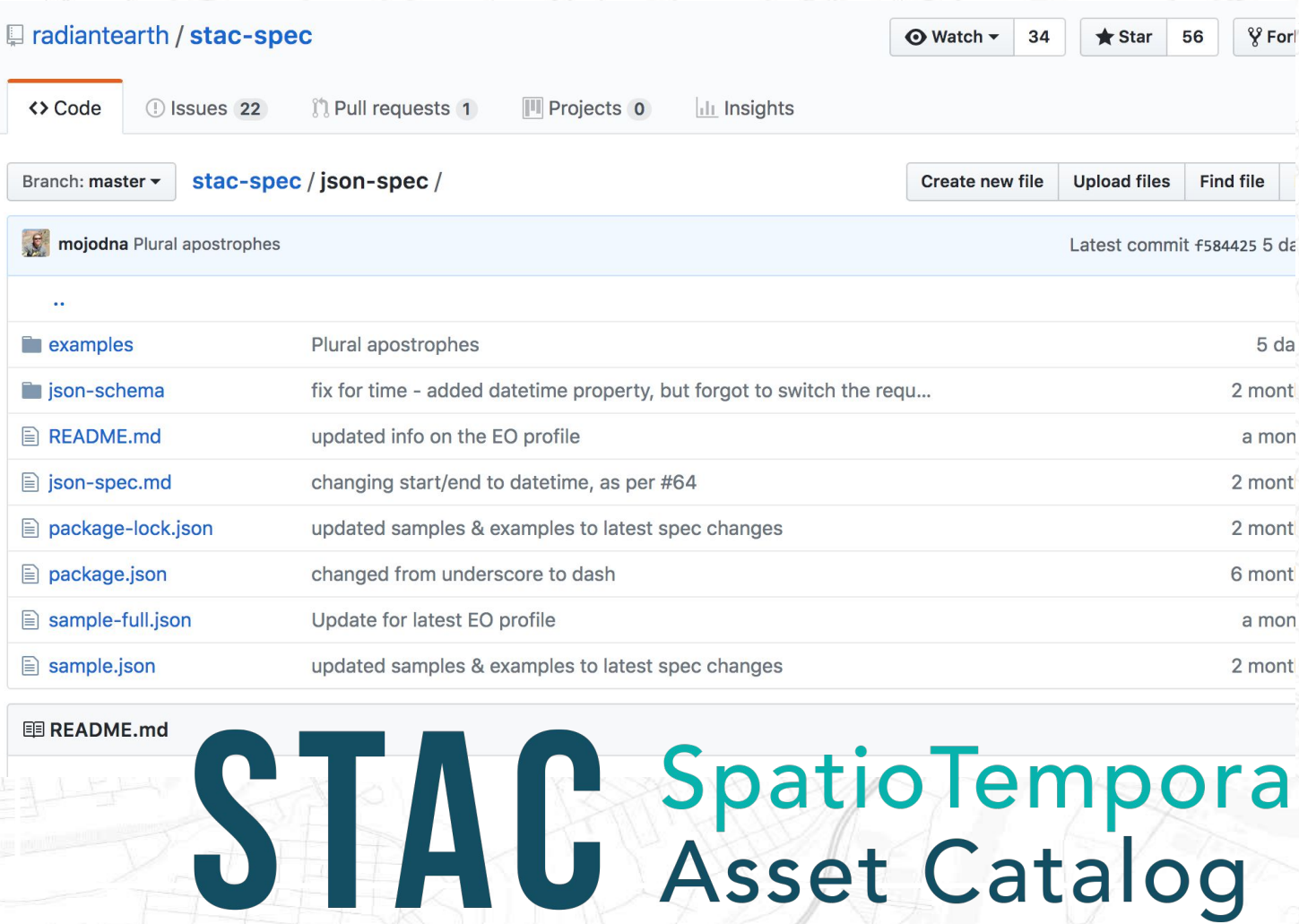
- Provenance Metadata
- Include Summary Statistics
- File naming convention
- Catalog suggestion

COG “Profiles”: Slippy Map

- Restrict CRS as EPSG:3857
- Align GeoTiff segments with slippy tile grid
- Align GeoTiff files with tile boundaries
- Serve tile endpoints directly from COG

COG World Coverage?

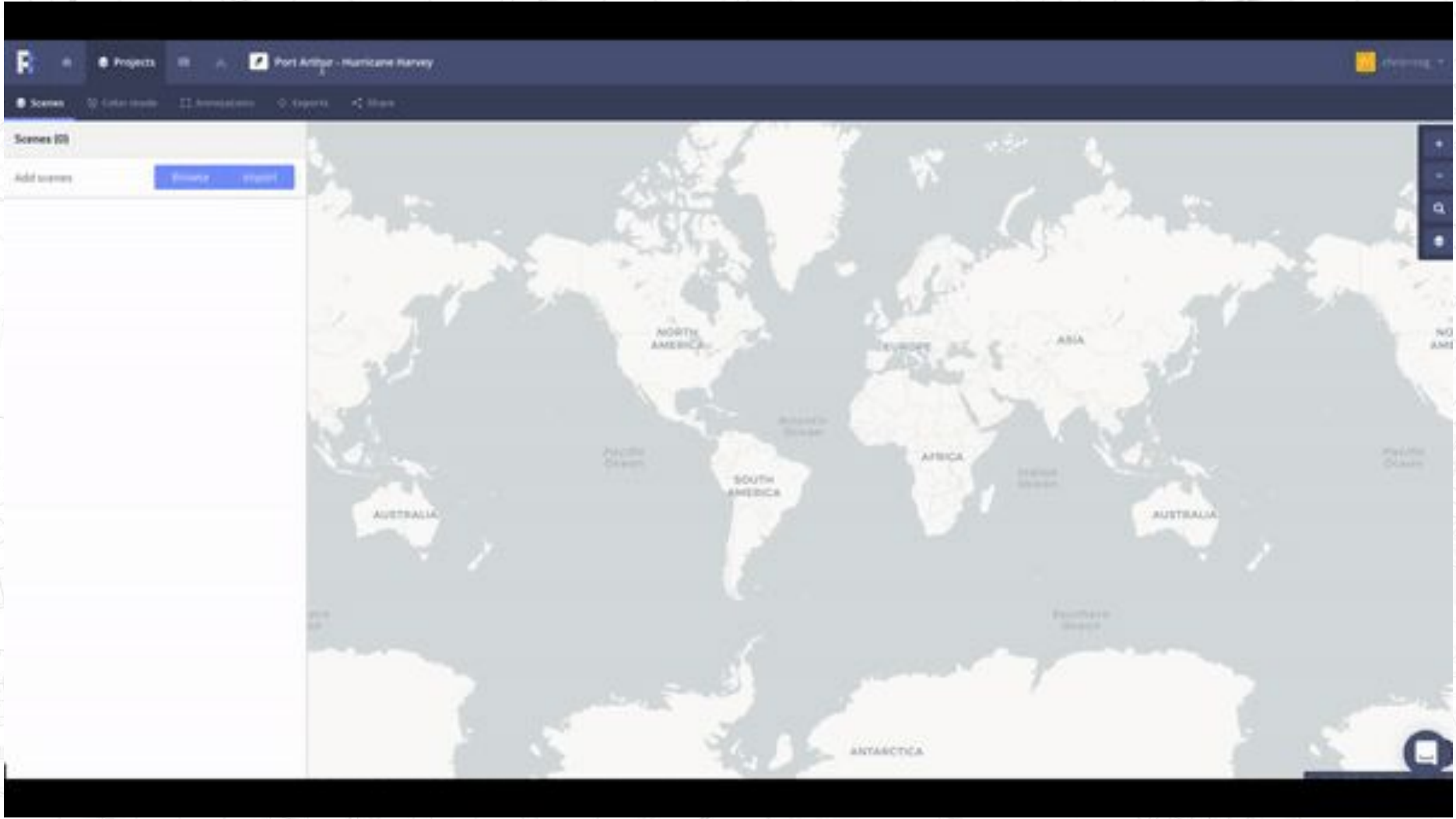
“You’re going to need more than one GeoTIFF.”

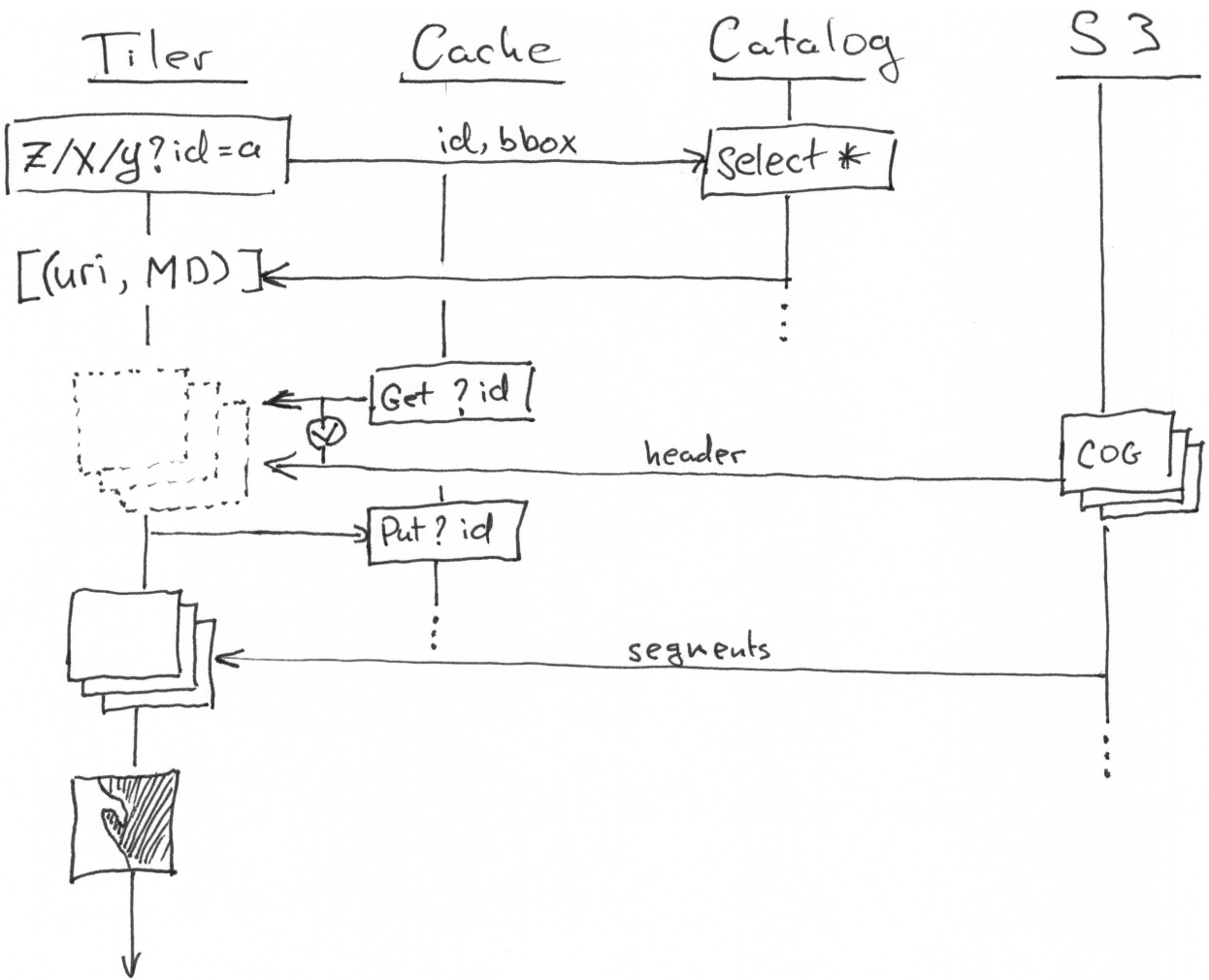

radiantearth / stac-spec
Watch 34
Star 56
For
Code
Issues 22
Pull requests 1
Projects 0
Insights
Branch: master
stac-spec / json-spec /
Create new file
Upload files
Find file
mojodna Plural apostrophes
Latest commit f584425 5 da
..
examples Plural apostrophes 5 da
json-schema fix for time - added datetime property, but forgot to switch the requ... 2 mont
README.md updated info on the EO profile a mon
json-spec.md changing start/end to datetime, as per #64 2 mont
package-lock.json updated samples & examples to latest spec changes 2 mont
package.json changed from underscore to dash 6 mont
sample-full.json Update for latest EO profile a mon
sample.json updated samples & examples to latest spec changes 2 mont
README.md
STAC
SpatioTemporal
Asset Catalog

s3://[bucket]/[Shard]_[GeoHash]_[ISO_TIME]_[ID].tiff

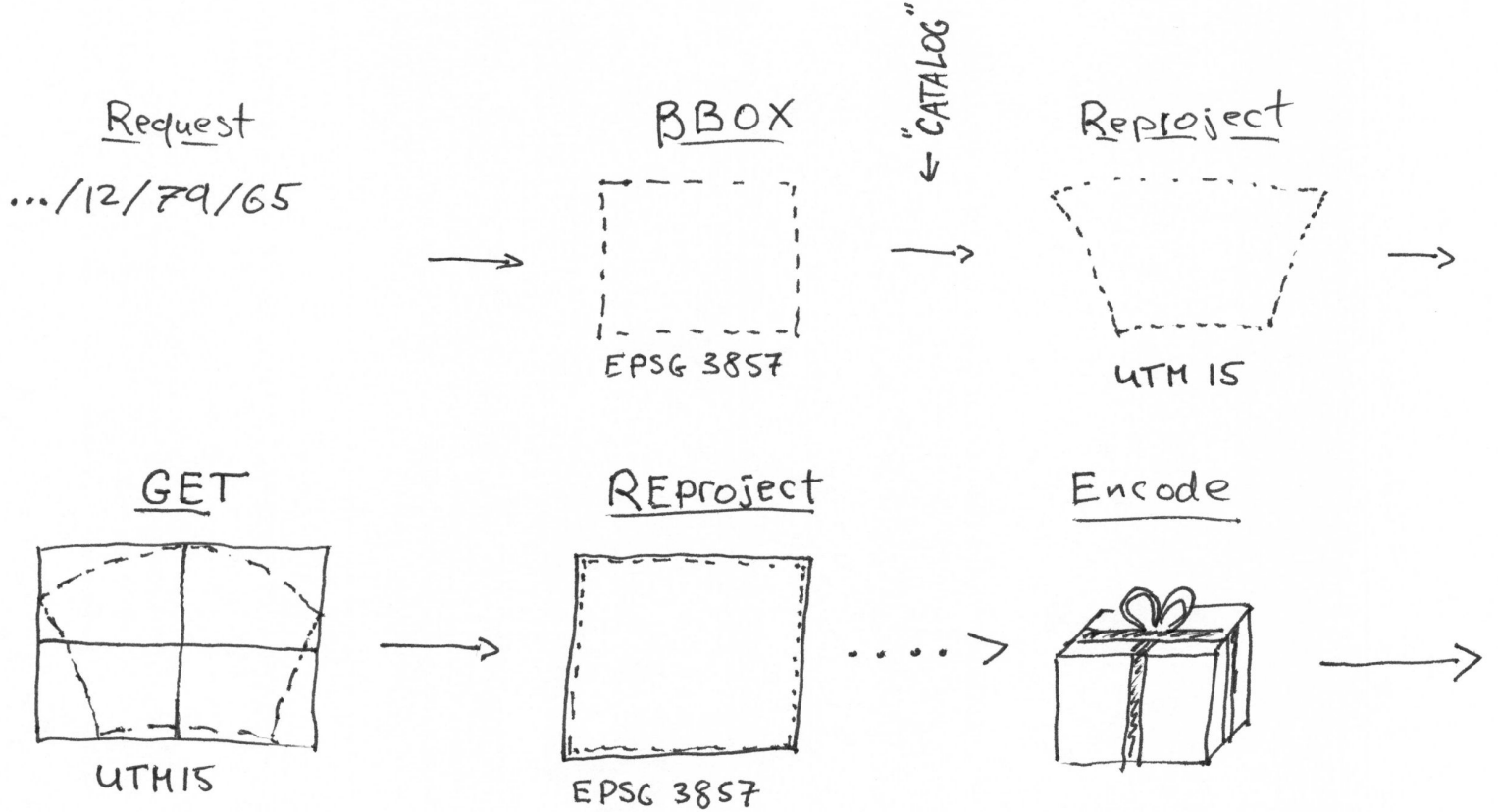
GeoTrellis & COGs


Raster
Foundry


Projects
Port Arthur - Hurricane Harvey
Scenes
Scenes (0)
Add scenes
Browse
Import
NORTH AMERICA
EUROPE
ASIA
AFRICA
SOUTH AMERICA
AUSTRALIA
AUSTRALIA
ANTARCTICA


Tiler
Cache
Catalog
S3
z/x/y?id=a
id, bbox
Select *
[(uri, MD)]
Get ?id
header
COG
Put ?id
segments

COG: Tile Request

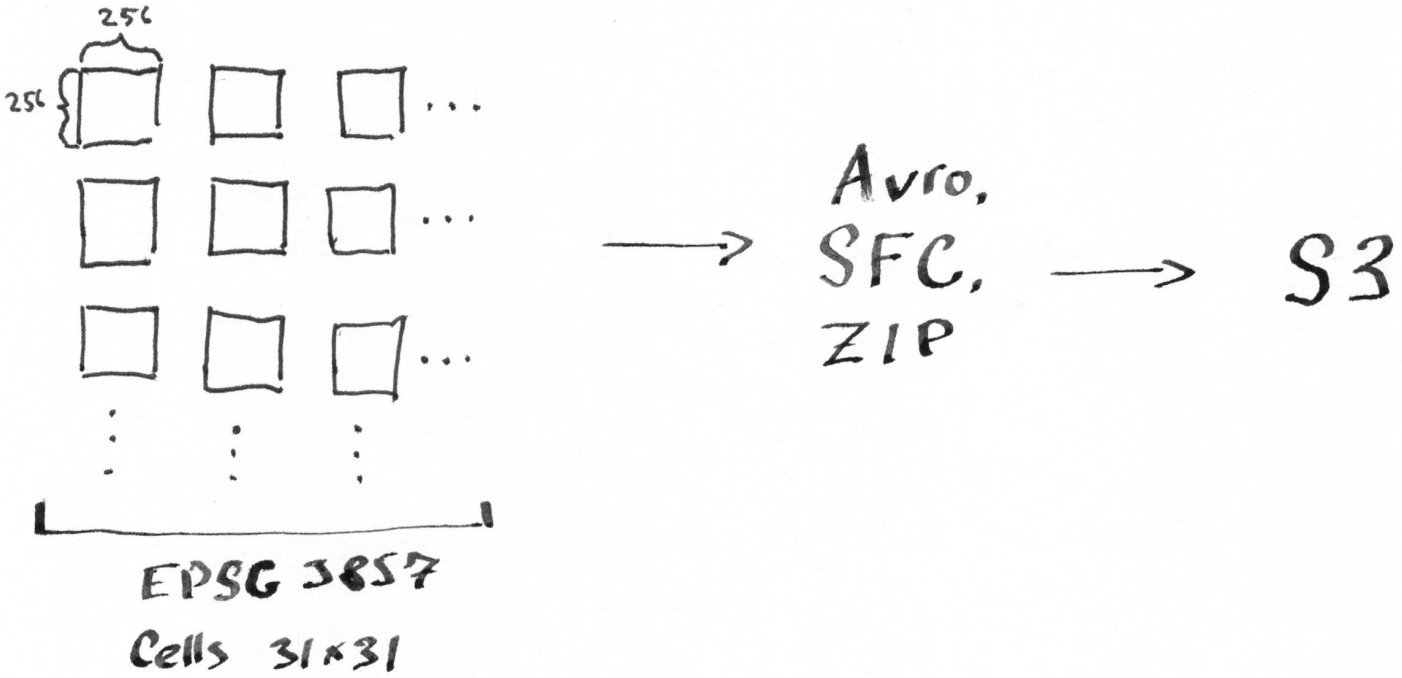


COG Tiler

- Application specific catalog
- Reprojection on read
- Memache gives responsiveness
- Raster metadata pushed into files

GeoTrellis Layers

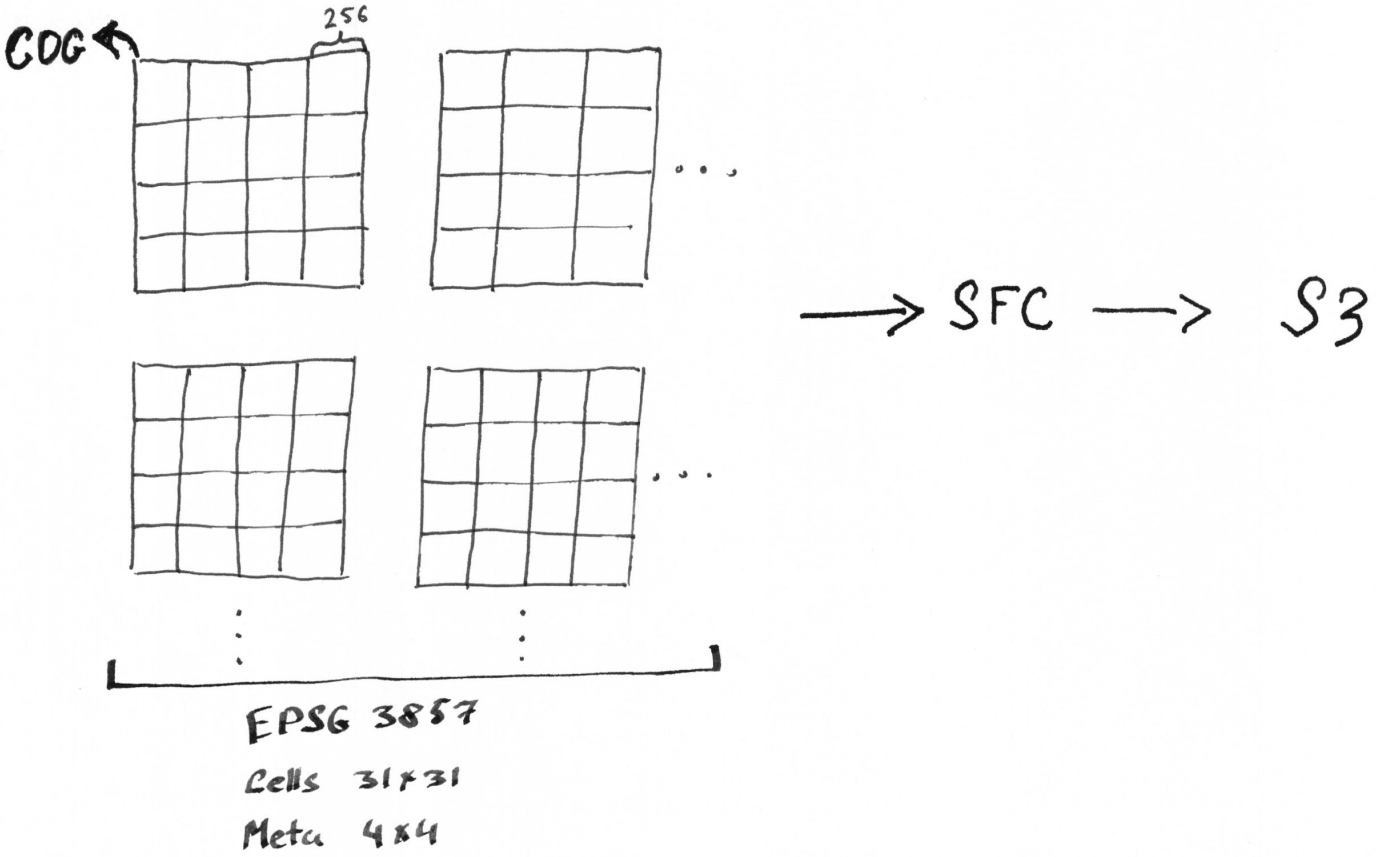
GeoTrellis: Avro Layers



GeoTrellis Avro Layers

- Many small files, one per tile
  - PUT requests get expensive
- Data duplication hazzard
  - Probably not authoritative
- Limited access
  - Requires GeoTrellis code to read

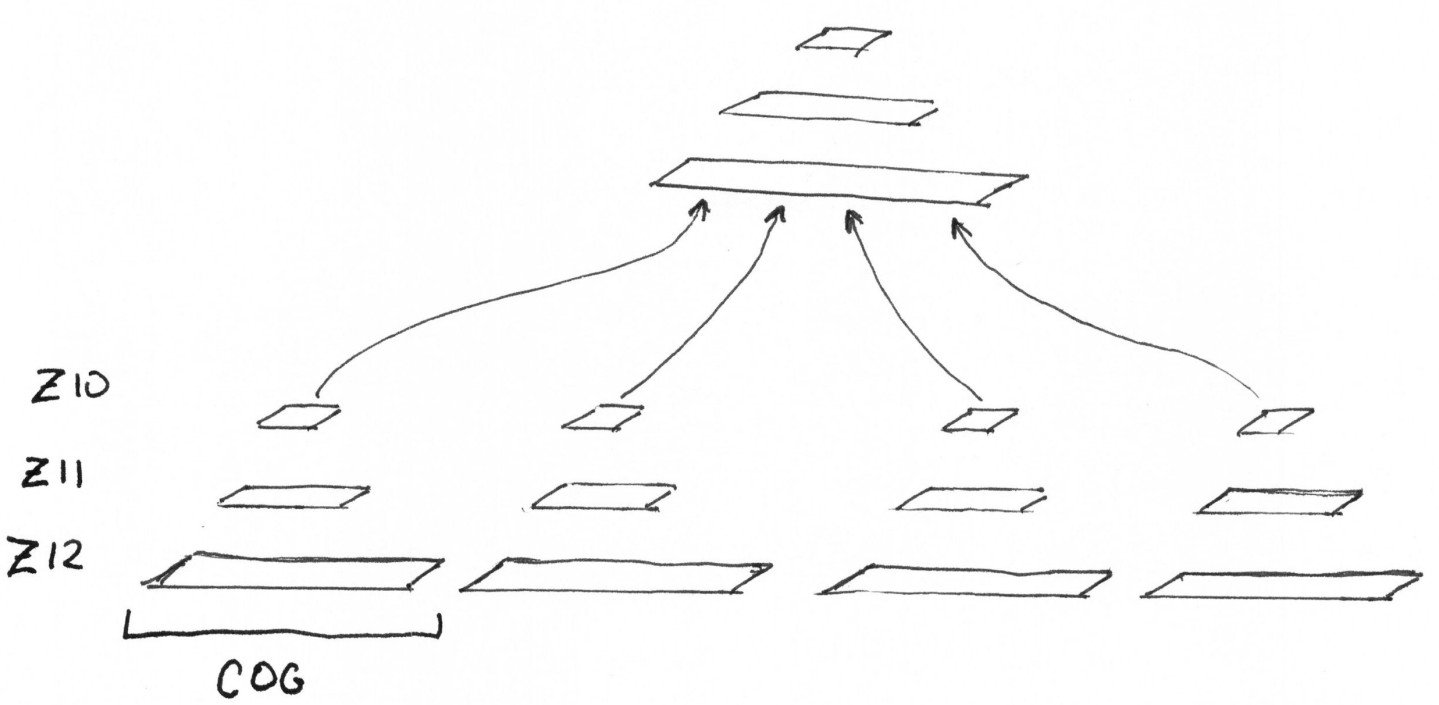
GeoTrellis: COG Layers

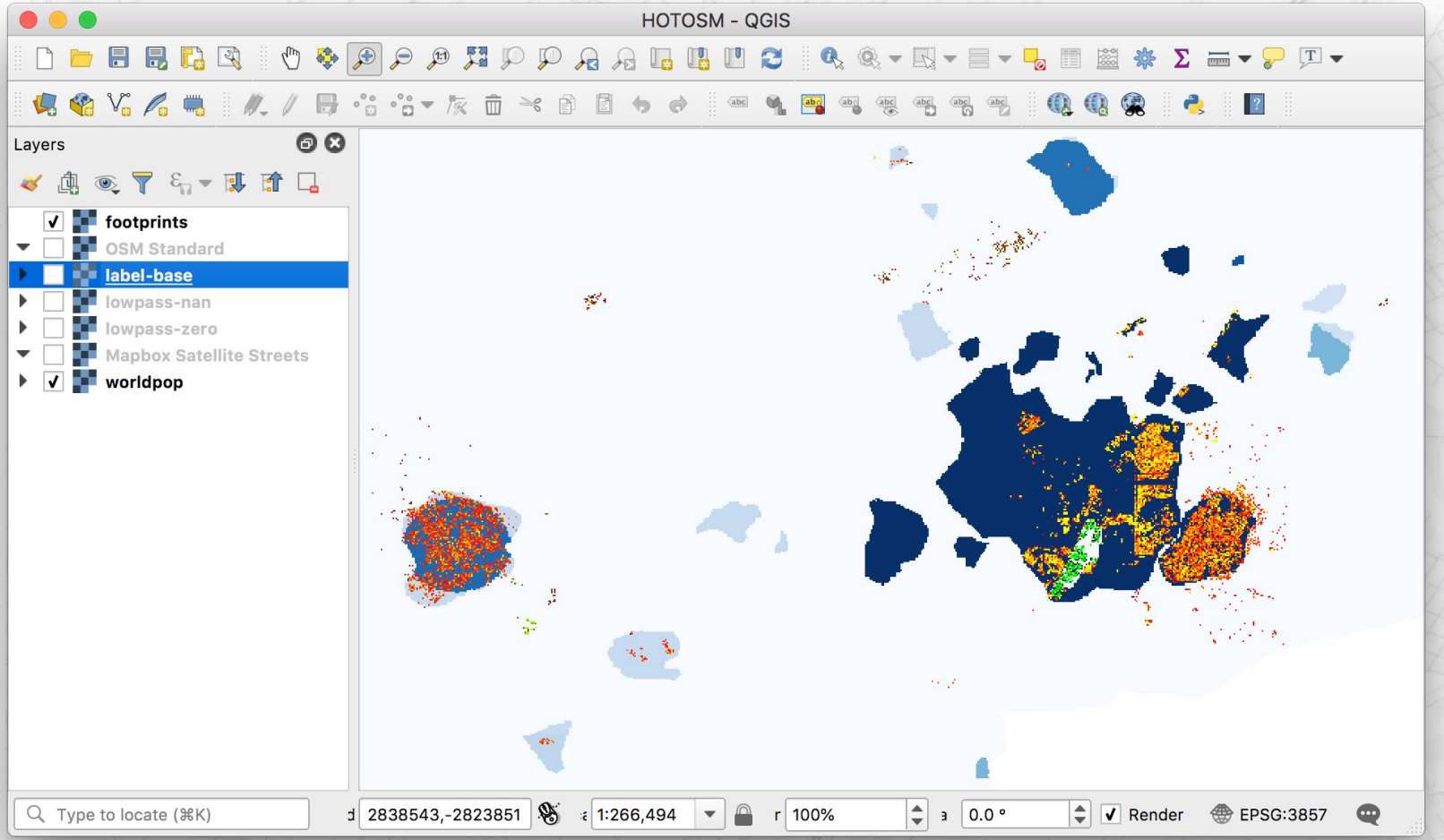


GeoTrellis Strucuted COG

- Base zoom tiles are in base GeoTiff layer
- Introduces “**DATE_TIME**” tag
- Includes **VRT** for layer files
- GeoTiff **segments match** tile size and layout
- Additional zoom levels stored in **overviews**
- **Pyramid** up when overviews “run out”

GeoTrellis: COG Pyramid




HOTOSM - QGIS
Layers
footprints
OSM Standard
label-base
lowpass-nan
lowpass-zero
Mapbox Satellite Streets
worldpop
Type to locate (⌘K)
2838543,-2823851
1:266,494
100%
0.0 °
Render
EPSG:3857

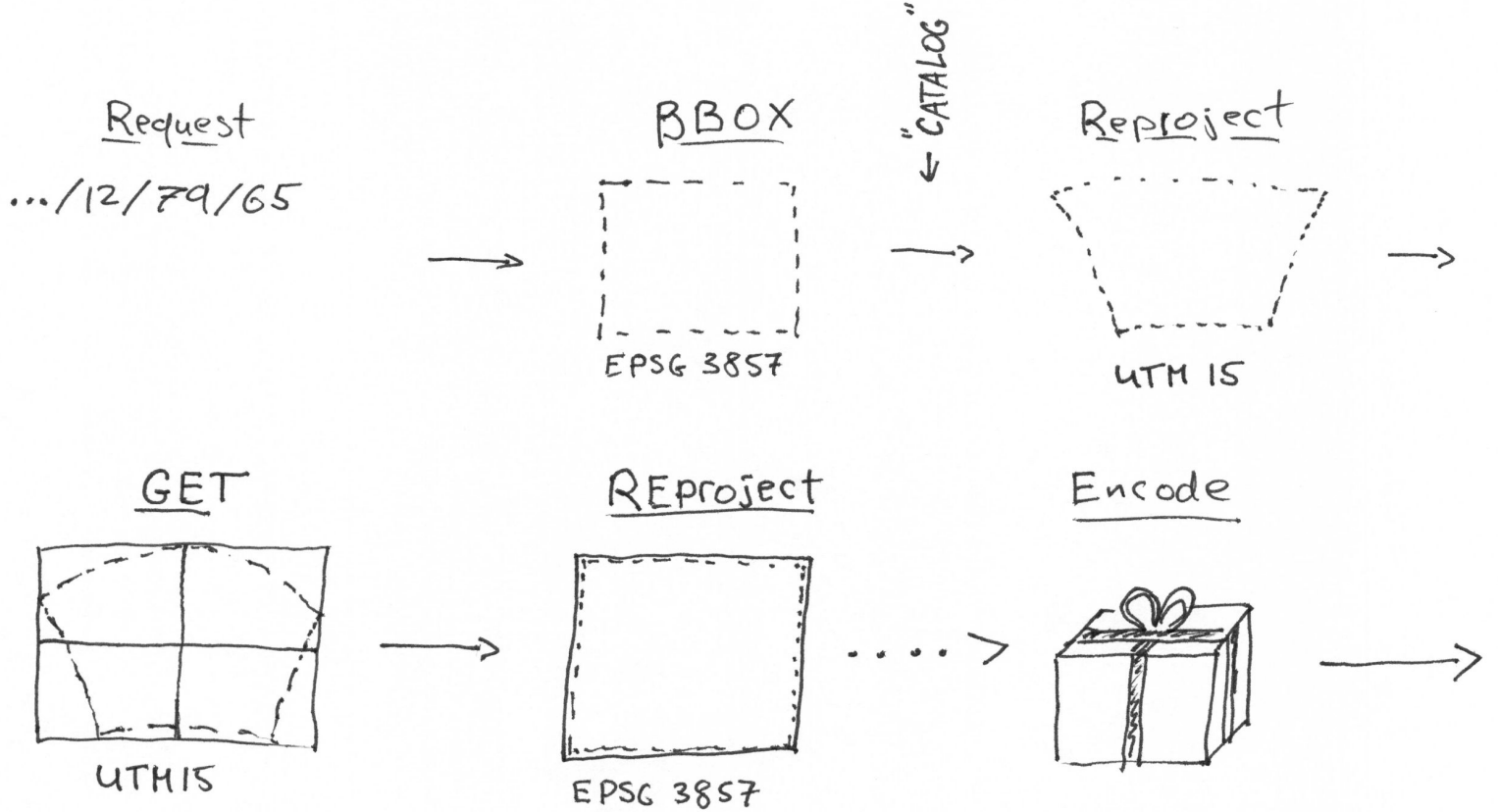
Unstructured COGs

Cloud Native thinking

GeoTrellis Unstructured COG

- Lots of good raster sources are already COG
- They don’t have a strict layout though
- Lets “inline” ingest process into the batch job
- Slower but reduces data duplication

COG: Tile Request (Again)



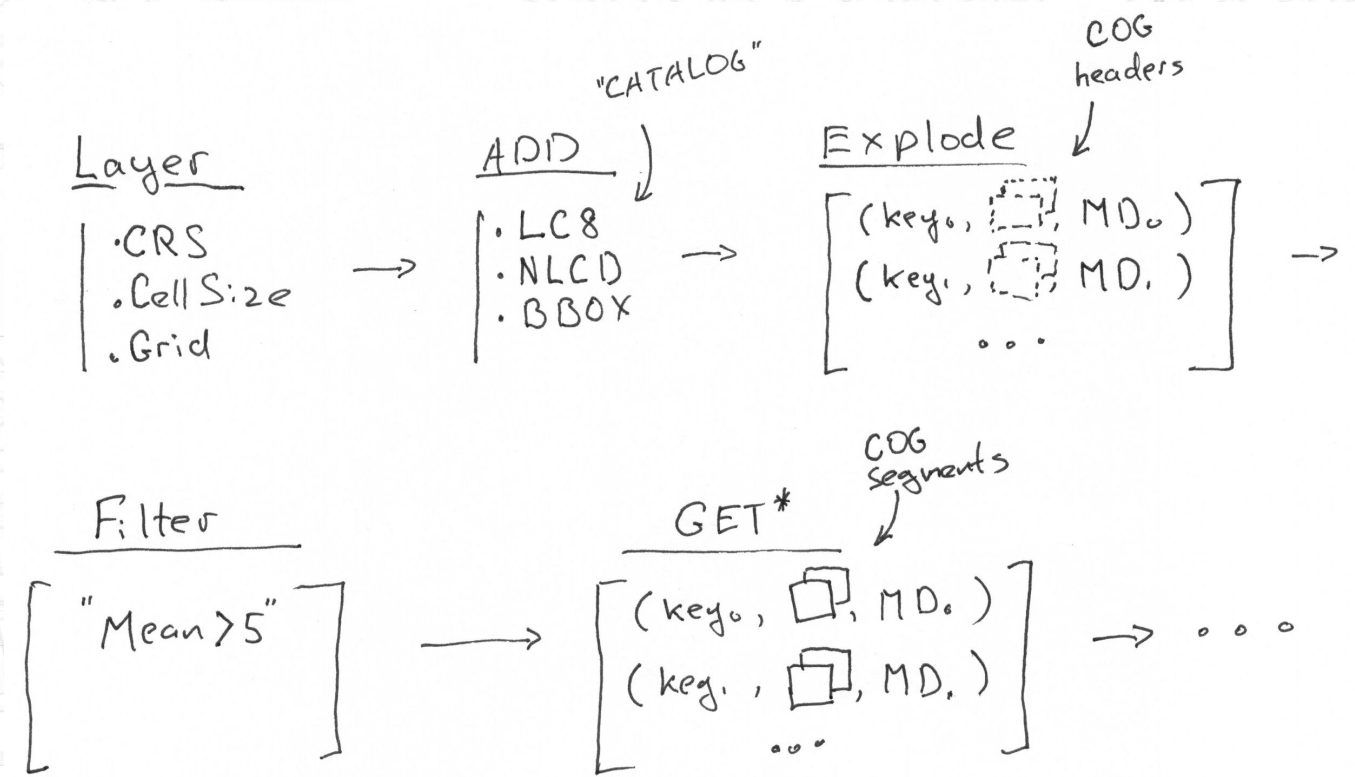
GeoTrellis Unstructured COG

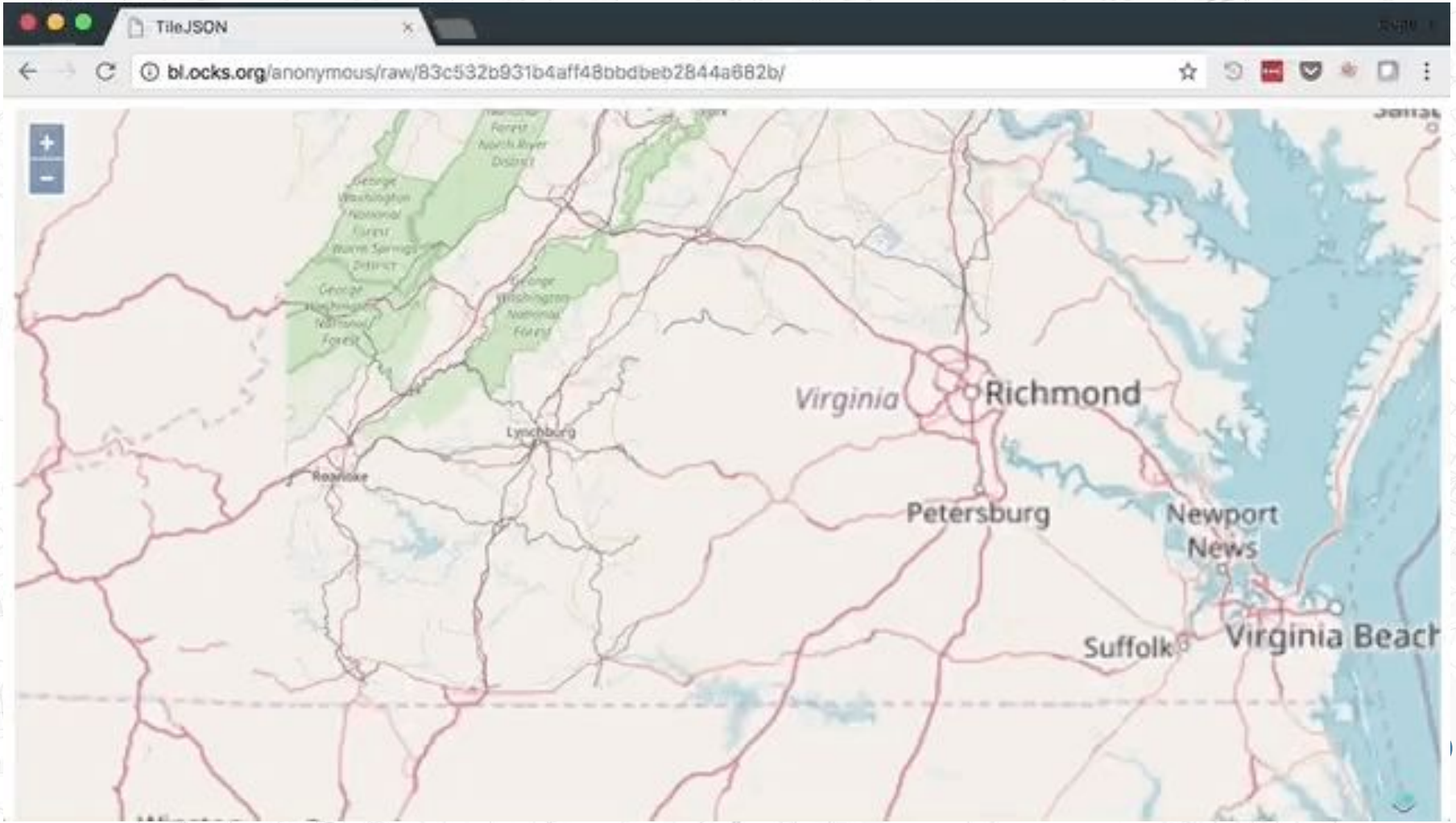
- Requires mechanism to query for raster names
  - WFS, STAC, PostgreSQL, Scan
- COG provies “gradual” wins over GeoTiff
- Reproject on read to match workflow

Coregistration Workflow

- Working with **multiple** raster layers with differing:
  - CRS, Resolution, Extent
- Must pick the working/target layout
- We can “push-down” this process into IO

Coregisteration Workflow




TileJSON
bl.ocks.org/anonymous/raw/83c532b931b4aff48bbdbeb2844a682b/
Virginia
Richmond
Lynchburg
Roanoke
Petersburg
Newport
News
Suffolk
Virginia Beach

COGs for GeoTrellis

- Treating GeoTiff as a Key/Value tile store
  - With built in notion level of detail
- Turns an ingest workflow into query workflow
  - Use the metadata to plan the read
- Opens up the results
  - QGIS, GeoServer, GDAL

COGs in General

- GeoJSON of raster formats
- Gradated spec
- Set of best practices
- Driven by implementations

THANKS

Q: Is this a good idea ?

Q: So is this a standard ?

Q: Why not use MRF ?