



# Serverless infrastructure to manage vector and tiff data: .pbf and COG

1. Principales objetivos
2. Arquitectura definida
3. Conclusiones

# Principales objetivos

- Consumo de datos vectoriales
  - Consumo de datos masivo: > 500.000 features
  - Incorporación de datos de OSM
  - Requisitos tecnológicos
    - Infraestructura backend mínima
      - Sin base de datos
      - Sin servidor de mapas
    - Consumo de datos estáticos por lotes

# Principales objetivos

- Consumo de datos raster
  - Origen de imágenes satelitales en formato .tiff
  - Requisitos tecnológicos
    - Infraestructura backend mínima
      - Sin servidor de mapas
      - Publicación dinámica de datos

# Arquitectura definida

- Tecnología del servidor
  - Arquitectura STAC
    - Define la forma en la que los datos se exponen
    - Implementación de un catálogo de datos estático
    - Estructura de carpetas que da soporte a la información espacial
    - Fichero .json que publica y permite la obtención de información

# Arquitectura definida

- Tecnología del servidor
  - Arquitectura STAC

It is simply a set of files on a web server that link to one another in a way that can be crawled. A static catalog can only really be crawled by search engines and active catalogs

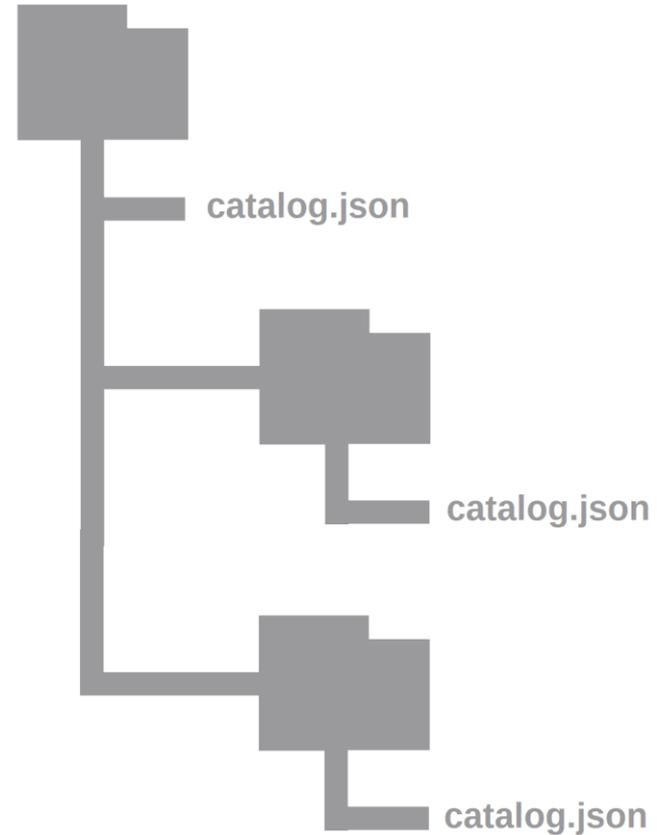
Expose as much asset metadata online as possible, so the static catalog offers a very lower barrier to entry for anyone with geospatial assets to make their data searchable

# Arquitectura definida

- Tecnología del servidor
  - Arquitectura STAC

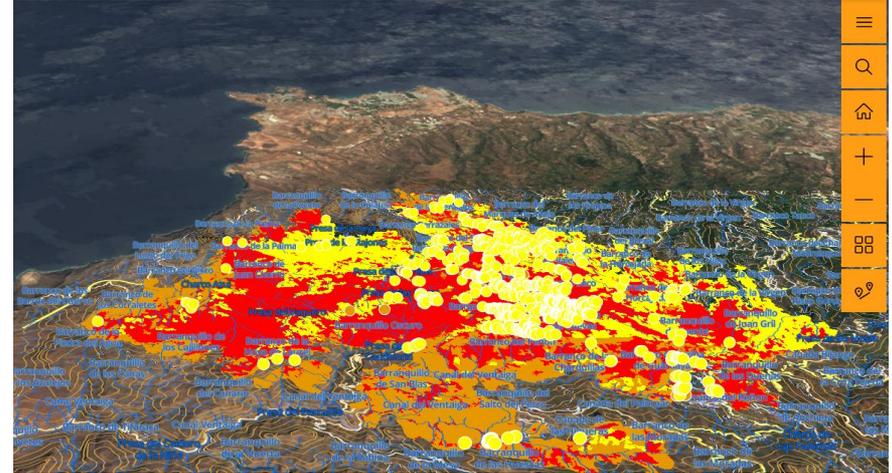
catalog.json:

```
{
  ...
  "links": [
    {"rel": "self", "href": "..."},
    {"rel": "child", "href": "..."},
    {"rel": "parent", "href": "..."},
    {"rel": "root", "href": "..."},
  ]
  ...
}
```



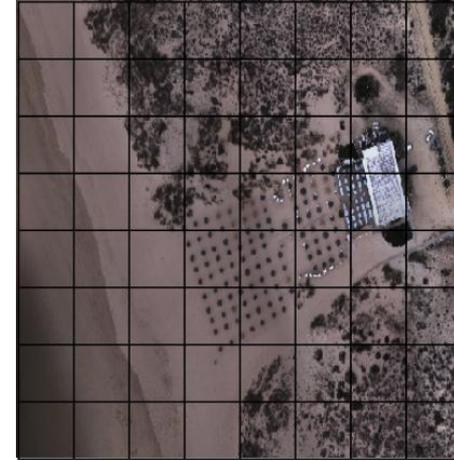
# Arquitectura definida

- Tecnología del servidor
  - Generación de información vectorial
    - Tiles vectoriales .pbf
    - Generación de tiles:
      - GeoServer + GWC
      - MapServer
      - GDAL
      - PostGIS: ST\_AsMVT + scripts
      - Tippecanoe
    - Origen de información
    - Grado de actualización de la información
    - Estilos de la información espacial
      - maputnik



# Arquitectura definida

- Tecnología del servidor
  - Generación de información raster
    - Cloud Optimized Geotiff
    - Consumo de imagen en streaming
    - Integrado en las peticiones http
    - Acceso a la zona de interés sin acceder al global del fichero raster
    - No existe tiempo de análisis, procesado y devolución de la zona requerida
    - Se reconoce como tipo de imagen en los navegadores



# Arquitectura definida

- Tecnología del servidor
  - Generación de información raster
    - 1ª Opción: proxy para tiles
      - Rdnt.io
      - Despliegue en lambda
    - 2ª Opción: geotiff.io
      - Integración con react.js
      - Acceso directo desde el cliente



# Conclusiones

- Entorno productivo haciendo uso de arquitectura serverless
  - Arquitectura STAC – catálogo estático – http server
  - .pbf para datos vectoriales
    - GDAL
    - Tippecanoe
  - COG para datos raster
    - Rdnt.io
    - Geotiff.io
- Ausencia de servidores para procesamiento de la información solicitada
- Procesado y renderizado en el navegador
- Escalabilidad intrínseca a un servicio de ficheros

# Gracias por su atención

[jenriquesoriano@guadaltel.com](mailto:jenriquesoriano@guadaltel.com)