Senti-water demo #2

Problem description

- Creating data processing system to visualize big water reservoirs in Poland
- We use recent satellite photos for this purpose

Deliverables:

- Database of big water reservoirs with spatio-temporal information
- Web application for browsing water reservoir data

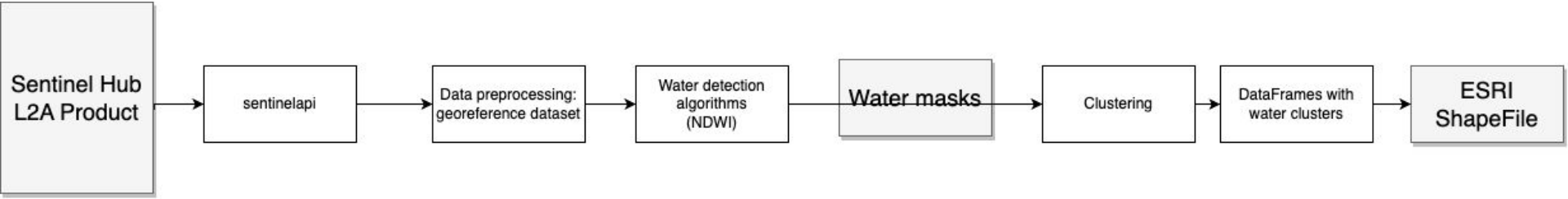
Satellite data

We use Sentinel-2 satellites data to create water reservoirs database

Satellite data products from these satellites are publicly available. We are able to retrieve images for Poland every 5-6 days.

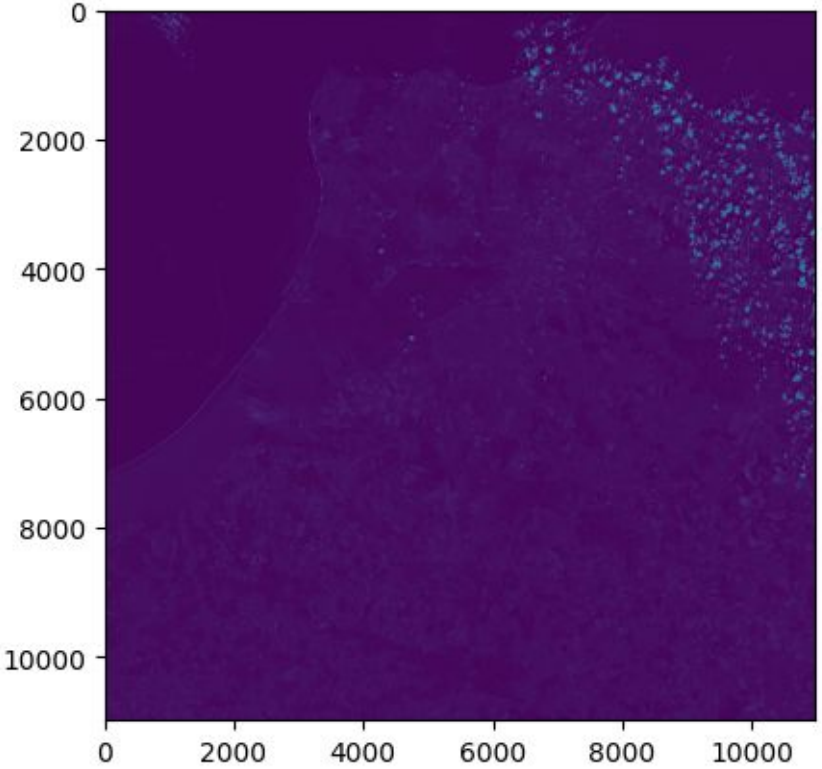
What have we done during Sprint 0?

- Research on satellite data analysis, data preprocessing, Sentinel API, gaining knowledge in multi-band image processing
- **First version of data processing pipeline that**
  - Downloads image from the satellite at a specific place and time
  - Show data in a notebook
  - Calculate the mask of water using NDWI technique
  - Show mask of water on internally created RGB satellite image

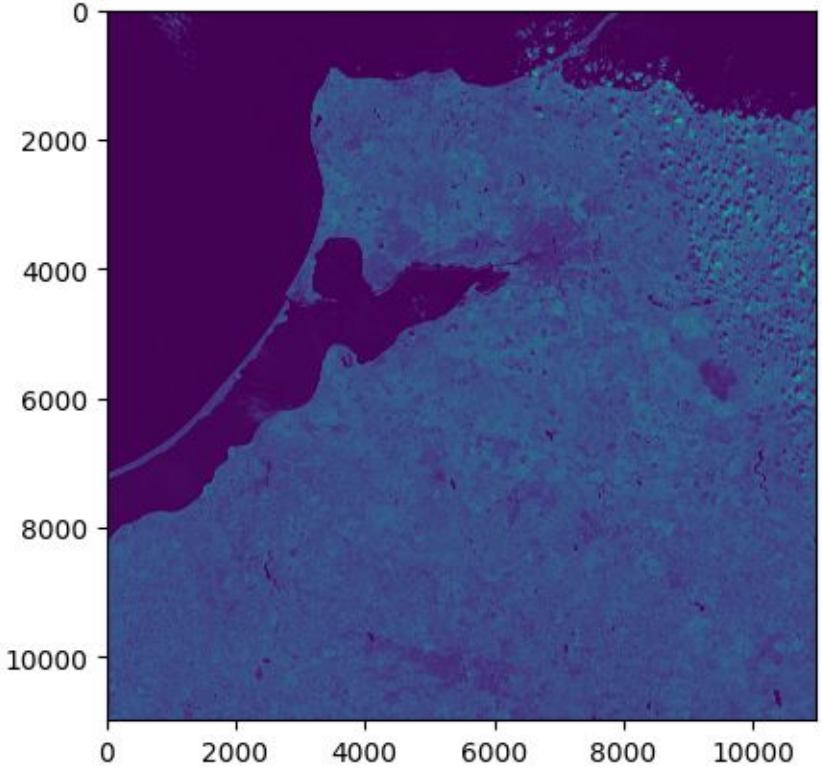


What have we done during Sprint 0? - preprocessing

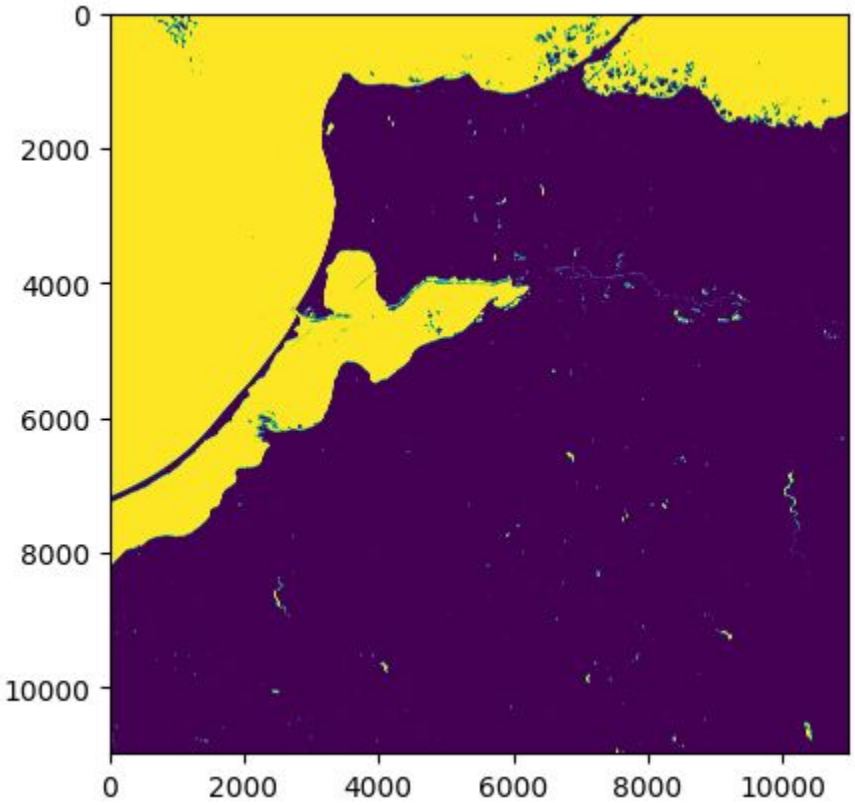
BAND 03 (GREEN)



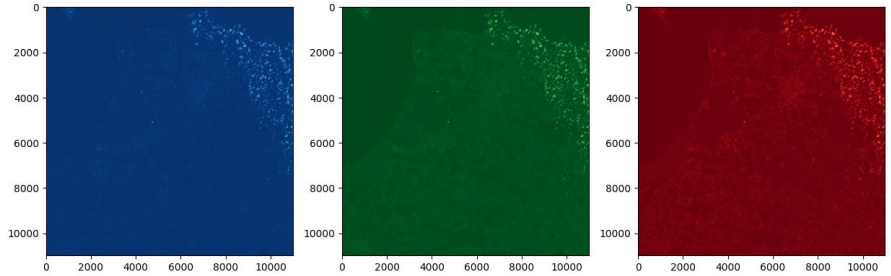
BAND 08 (NIR)



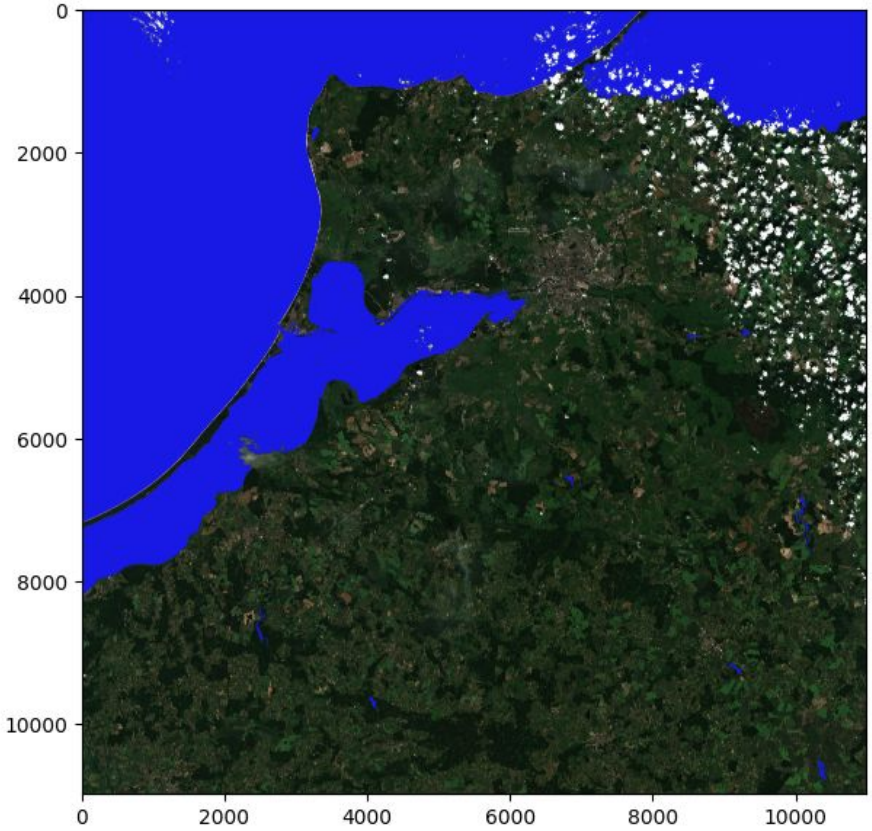
What have we done during Sprint 0? - NDWI calculation



What have we done during Sprint 0? - color photo creation



What have we done during Sprint 0? - water mask layer and cluster creation



What we planned to do in Sprint 1?

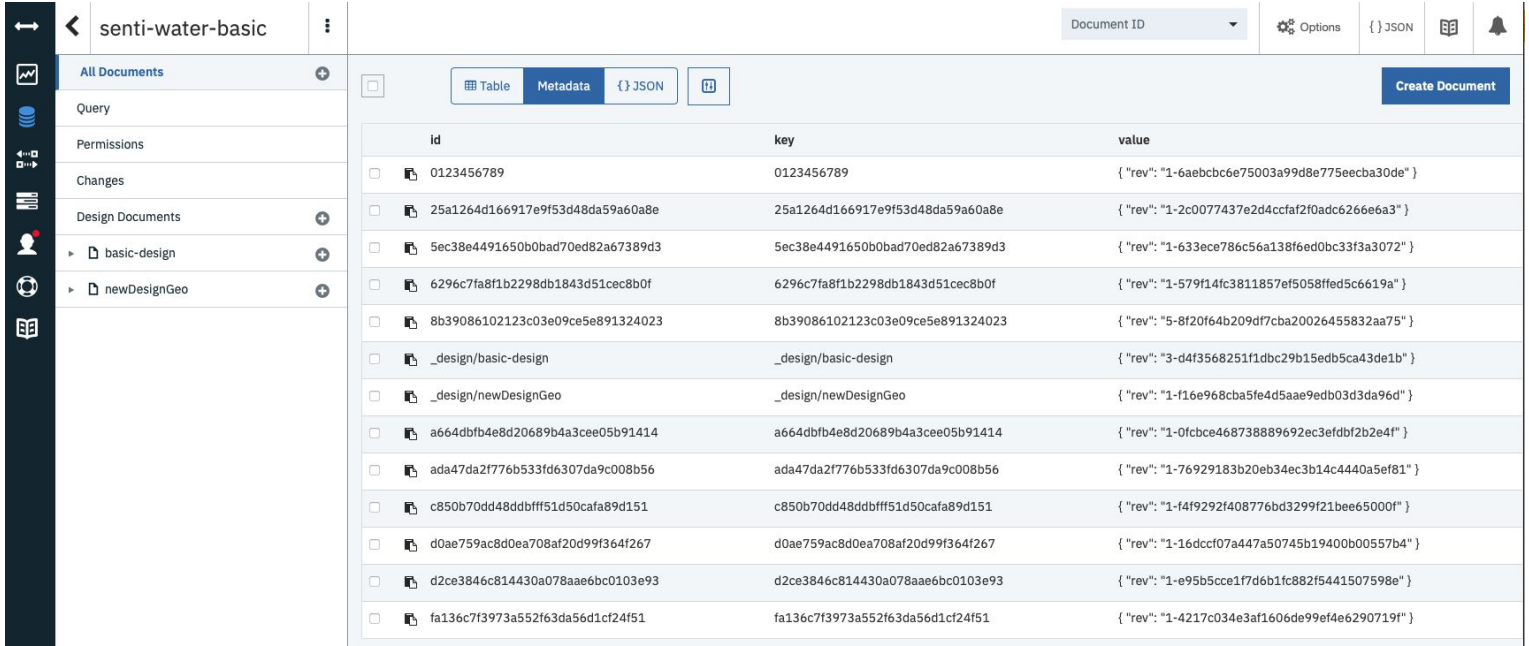
- Create a database (DB2 on Cloud + Geo extension)
- Download only the required files (+ preprocessing optimizations)
- Creating an algorithm that calculates the center of a water reservoir
- Designation and simplification of the shape of a water reservoir

What we actually did in Sprint 1?

- Create a database (IBM Cloudant)
- Preprocessing optimizations (+ download only the required files)
- Creating an algorithm that calculates the center of a water reservoir (partially)
- Designation and simplification of the shape of a water reservoir (moved to Sprint 2)
- Started creating simple React application that shows our data

Cloudant DB

- Experimenting with shapefiles and DB2
- Turned out too expensive (no geo-extension with Lite Cloud plan)
- IBM Cloudant simpler in development and cheaper (enough for now)
- For now we store very basic data: Geojson with geo points from each water reservoir



Optimizations

- Preprocessing optimization:
  - Before optimization:

    pixels -> coordinates -> algorithm -> good coordinates -> good pixels

  - After optimization:

    pixels -> algorithm -> good pixels

- Downloading only required files which require less data to download (4GB vs 0,5GB)

Started simple web application

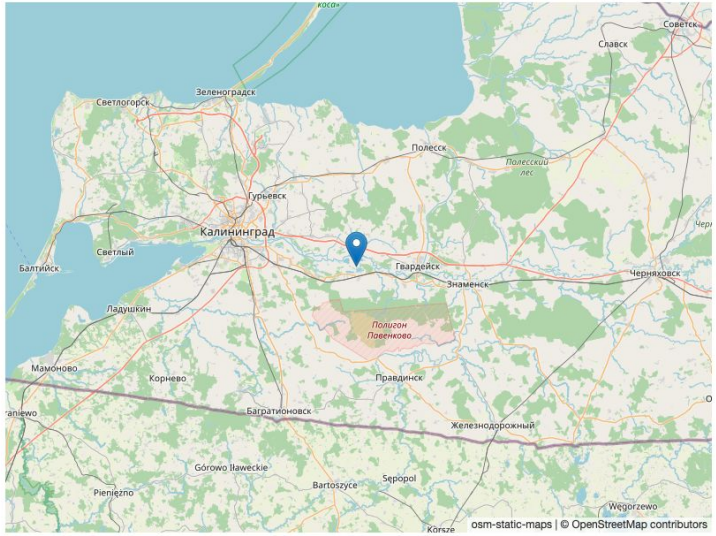
- Internal API for retrieval data from our database
- Simple page showing data and connecting it with OpenStreetMap for context

localhost:3000/waters

localhost:3000/waters

Senti Water

DB ID	X	Y
0	19.4346016390245	55.03692054159784
1	20.68322683989857	55.04657393156168
291	20.393178623331796	54.66497234272443
346	20.88672207899719	54.64531965670733
360	20.76557445909724	54.63948398872447
437	20.506506748844537	54.460579188120114
451	21.01125416944789	54.434624431229544
527	19.84764864969051	54.28918662750187
577	20.84830105622954	54.22539218509128
606	20.08823495073795	54.17781783921189
659	21.051691598285828	54.10288472820522
72	19.95032041177108	54.89525391897372



What are we going to do in Sprint 2?

- Algorithm that calculates water reservoir geometry from water mask
- Extend data stored in DB: (center of water body, geometry, surface area, timestamp, name field)
- Second water retrieval method (MNDWI)
- Further pipeline optimization (with metrics)
- Application extension:
  - Ability to edit entries (adding name to existing data)
  - Showing additional data (surface area, timestamp, name)
  - (STRETCH) Showing water mask, geometry and original satellite photo