Cloud Engineering Supports Spatiotemporal Data Analysis

Jack McNulty, Junior, Computer Science

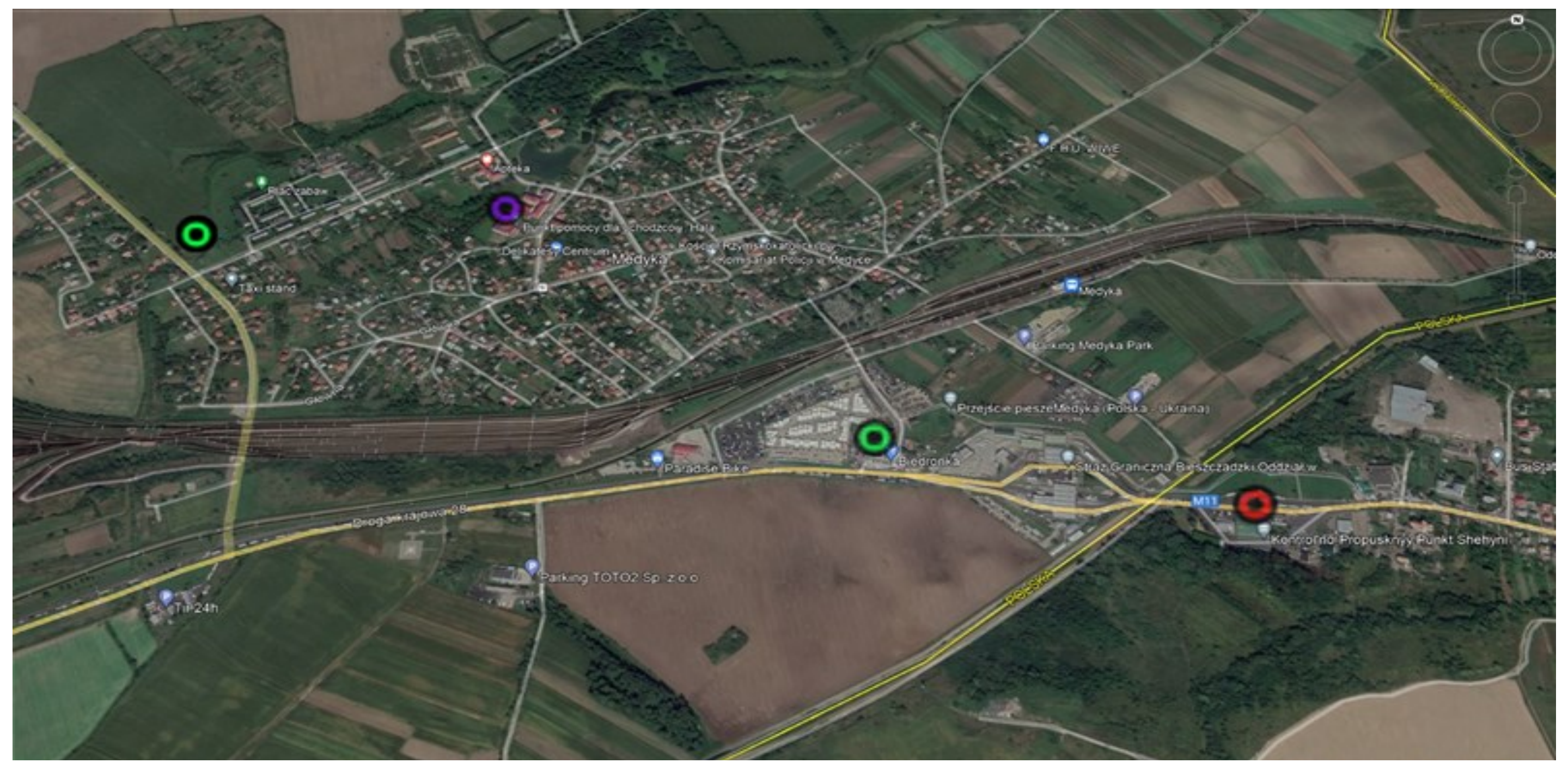
Jeffrey Oltmanns, MSSE, AF Lt. Col.

RESEARCH QUESTIONS

1.How is multi-variate unstructured data useful to humanitarian efforts?

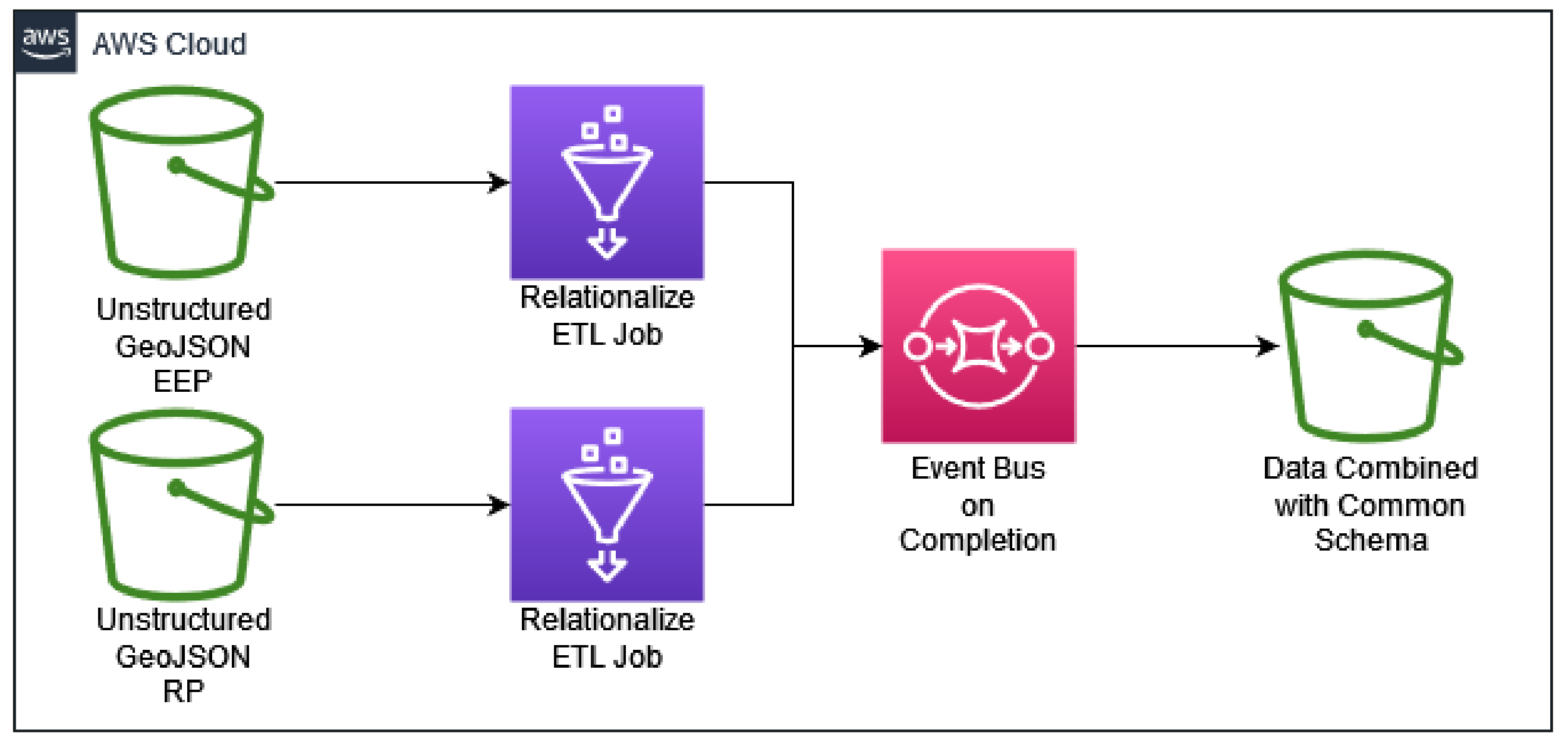
2.Do cloud-based solutions exist to engineer, visualize, and assess big data?

Using Amazon Web Services and Etegent Technologies' NTellus Earth View, we can layer multiple datasets in one view. Polish reception points are places that are caring for Ukrainian refugees. Border crossing points are ones that refugees will cross to leave the country to safety. We can then draw conclusions on which route they took to get there.

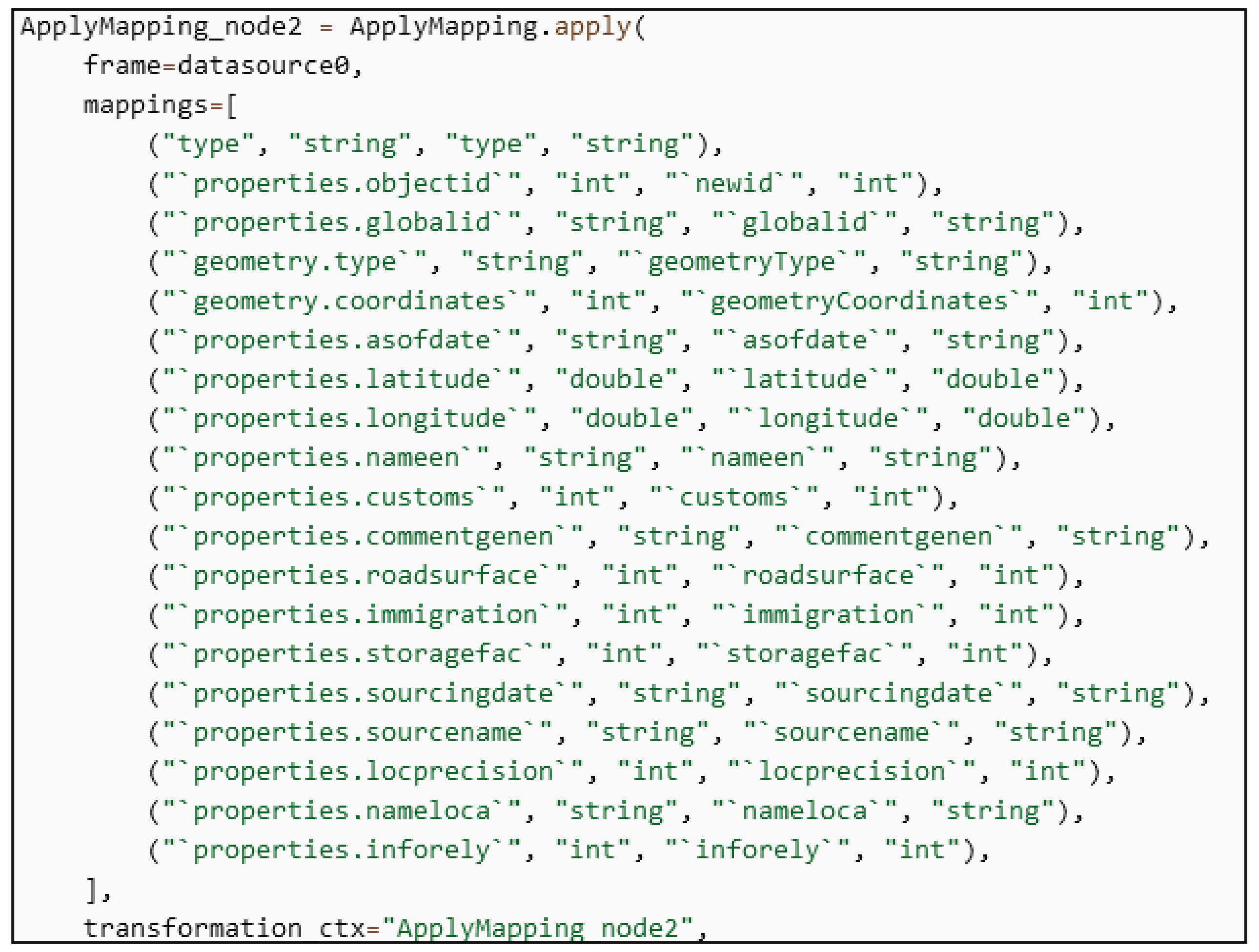


Red points are border crossings, purple points are reception points in Ukraine. The data is provided by Humanitarian Data Exchange (HDE). Basemap provided by Google Earth.

Using NTellus' Distance feature, we were able to choose two points on the map and find the distance between them. We found that the distance between the reception point and the border crossing was 1.5km away. Humanitarian initiatives could use this information to more effectively search for injured refugees or plan for logistical deliveries and security. Considering it takes approximately 15 minutes for the average person to walk 1km, we could extrapolate this information and determine how long it should take to walk from border crossing to reception point.

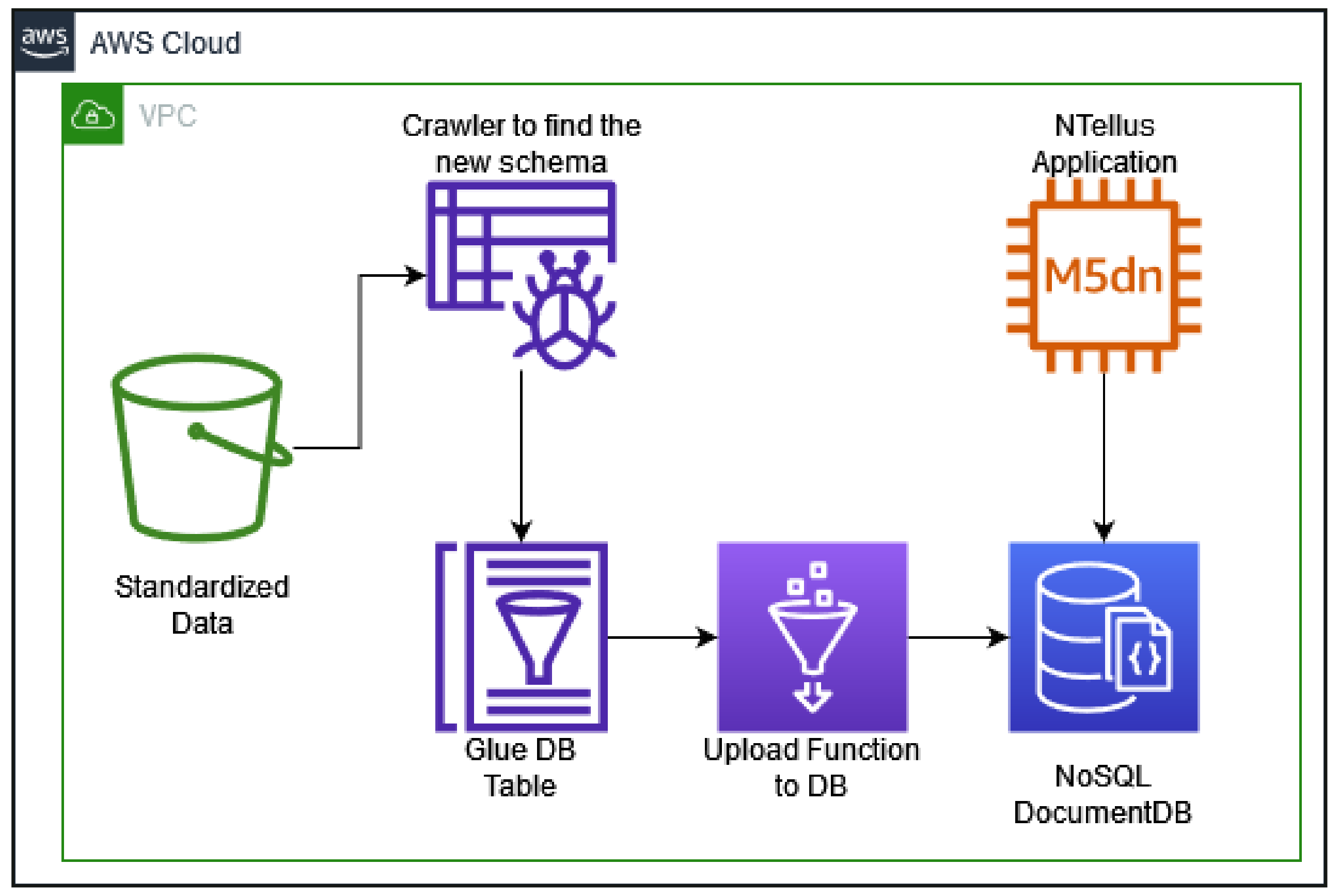


Extract, Transform, and Load (ETL) process for turning unstructured data into STAC compliant data



```
ApplyMapping_node2 = ApplyMapping.apply(
    frame=datasource0,
    mappings=[
        ("type", "string", "type", "string"),
        ("`properties.objectid`", "int", "`newid`", "int"),
        ("`properties.globalid`", "string", "`globalid`", "string"),
        ("`geometry.type`", "string", "`geometryType`", "string"),
        ("`geometry.coordinates`", "int", "`geometryCoordinates`", "int"),
        ("`properties.asofdate`", "string", "`asofdate`", "string"),
        ("`properties.latitude`", "double", "`latitude`", "double"),
        ("`properties.longitude`", "double", "`longitude`", "double"),
        ("`properties.nameen`", "string", "`nameen`", "string"),
        ("`properties.customs`", "int", "`customs`", "int"),
        ("`properties.commentgenen`", "string", "`commentgenen`", "string"),
        ("`properties.roadsurface`", "int", "`roadsurface`", "int"),
        ("`properties.immigration`", "int", "`immigration`", "int"),
        ("`properties.storagefac`", "int", "`storagefac`", "int"),
        ("`properties.sourcingdate`", "string", "`sourcingdate`", "string"),
        ("`properties.sourcename`", "string", "`sourcename`", "string"),
        ("`properties.locprecision`", "int", "`locprecision`", "int"),
        ("`properties.nameloca`", "string", "`nameloca`", "string"),
        ("`properties.inforely`", "int", "`inforely`", "int"),
    ],
    transformation_ctx="ApplyMapping_node2",
```

Using the Apache Spark Library, I was able to relationalize the data and change it from a non-standardized format, to comply with Spatiotemporal Asset Catalogue (STAC). Relationalizing the data makes it so that it is no longer stored in a nested format. As seen in the above code, I was able to rename and recast the data types for each field to put it in a form that works with our tools hosted in the Commercial Innovation Center (CIC). Doing this allows the data to be hosted in a non-relational database that can be accessed by various tools that support NoSQL queries and visualize the data. The primary purpose of this was to ensure that the latitude and longitude fields were of type double, as they were initially presented in type string.



Process for accessing the STAC compliant data through NTellus Earth View

After retrieving the dataset from the HDE, AWS Glue and Apache Spark was used to normalize and change the structure of the data. An AWS DocumentDB was developed to serve as an endpoint from the Glue service. Network connections were then made between the NTellus application and the database to read entries as a layer on Earth View. AWS serves as a level of fusion to bring all sources together in one environment. Without AWS, the analysis would cost an end user over 10 times more money, and the user would have to manage all of the infrastructure on their own, rather than AWS managed services.

Using AI/ML within the AWS Cloud, we could host advanced classification tools, such as the SaaS provider HEAVY.AI. This service would be able to classify refugee movement and density on its own, and provide those insights to humanitarian aid efforts. HEAVY.AI could also use other Synthetic Aperture Radar imagery and datasets to create more comprehensive models and provide better insights to routes taken. You can also use open source intelligence to find stronger insights on the data, as seen in the image of the street view below provided by Google.

Street view of Polish recreation center housing Ukrainian Refugees

Polish Refugee Center for fleeing Ukrainians in Medyka, Poland from Google Maps