Our topic is In-Browser Raster Data Processing. In order to understand this topic, first I have to talk about raster data and web assembly.

**Raster Data**

Data from the satellite that analyzes the geographical features of earth that is called spatial data. And this raster data is a type of spatial data. The representation of raster data is multi-dimensional arrays.

**Web Assembly or wasm**

It is a byte code instruction format. That is, first there will be a program code in the language that we need for eg: C++ or Python. Then we compile it by using Web Assembly. The output will be the byte code. Which can be run by linking together with Javascript and HTML in a modern browser. Since, it is a low-level code, it achieves near native execution speed. Polyglot refers to the multiple programming language that we can choose to execute.

**Why? Or Motivation**

The traditional method to process raster data is with the use of server side rendering. That means, whenever from a browser if we are to request for some operation, it will go to the server were the operation code is hosted and fetch data from different server then process it from the hosted server and come back to the client side with the response.

There are two areas where we can reduce the time if we use wasm.

1. In Wasm it is client-side rendering so for processing there is no need to go to external server only fetching the data from the database server.
2. The low-level code with wasm achieves a near native execution speed as mentioned earlier.

With these two we can save time or make the browser response faster.

**Implementation**

For implementation we need a data loader to load the raster data properly, then to perform the operations webtensor is required or the raster data processing library. And finally we will need a user interface to visualize the output of those operations with raster data.

While implementing data loader we can consider these parameters. Coordinate Reference System Type, Spatial Extend, Band, Chunk Size.

**User Interface**

If we have enough time we will be building a nice interface with map plots and display aggregation results while clicking on the map. Otherwise we will just provide a dropdown for options and a submit button to view maybe histograms and other aggregated results.

**Tools**

We are planning to use Emscripten for the raster data processing library and if its difficult to implement with C++ we are thinking of switching to python with the pyodide compiler.

GDAL library is used for reading and writing raster data we can use this or libraries built on top of this library for our implementations.

Xarray is a library to manage multi dimensional arrays with labels. That can also be used for processing raster data.

**Benchmarking**

For performance measurement we are thinking of implementing the same program without wasm and compare the execution speed.