0. Title:

Geofront: A Browser-based geo-computation platform in the era of cloud-native geodata.

Introduction

Context

```
SUMMARY:
- define context: web gis applications

3 entwined trends within this context:
- Cloud native movement
- Thick client movement
- Denichifying geospatial tools

COMMENTS
Which term to use?
- Web GIS
- Web application
- browser-based application
- web tool
- GIS tool
```

The geospatial web form an indispensable component of the modern geospatial software landscape. For the average person, an interactive \ac{giss} web application is often their first and only exposure to a \acs{gis}, be it a web mapping service, a navigation system, or a pandemic outbreak dashboard. A web application is cross-platform by nature, and offers ease of accessibility, since no installment or app-store interaction is required to run or update the app: As soon as it can be found, it can be used. The ability to share a full application with a link, or to embed it within the larger context of a webpage is also not trivial. Together, these aspects have made the browser a popular host for many important geographical applications, especially when accessibility is vital.

Three major trends are present in or approaching the field of web \ac{GISS}.

```
Mention Gaming PC streaming?: https://shadow.tech/en-gb/ ? related works?
- spotify streaming?
```

1

First and foremost, a significant development within the Open Geospatial Consortium (OGC) is the recent attention around "Cloud Native geospatial". This movement envisions a future of geodata as large, singular, statically hosted files, accompanied with cloud-based geo-computation tools similar to the Google Earth Engine. This architecture would

- make use of near-infinite resources in the shape of supercomputers and cloud-storage.
- finding large-scale patterns
- This will be easier and cheaper for data providers,
- and allows for cloud-hosted geodata to be more easily be consumed by cloud-compute tools.
- This really is a 'new era'

The cloud-native initiative aims to radically simplify geodata storehouses to just static file servers. All processing and analysis of this geodata can then be performed by separate cloud-based web services. For web GIS, this would offer direct data streaming options.

```
SECOND TREND

% A clear trend within these geo-web applications, is a push for increasingly
complex applications.
%   - Such as the ninja cityjson web viewer
%     - Proofs that the web can not only be used as viewer or end-user
application,
%       but also for geodata analysis and editing.
%     - Geocomputation
%     - Both for end-user tools, and tools used by developers.
%   - Also, Modern web tools such as webassembly allow us to run native geo-
analysis applications directly in a web browser
%     - Leading to tools such as cj-val, a 3D city validator.

% TREND : HEAVY CLIENT-SIDE FOR INTERACTIVITY
% client-side heavy web applications

% One of the biggest benefits of this setup is \emph{Interactivity}.
% If logic runs in the client, it can more quickly react to behavior of the user.
% A full client-server roundtrip containing lots of geodata will in many
situations take longer.

A second trend can be found
In recent years, a trend towards increasingly complex and demanding web
applications is noticeable.
-> not only end-users, editing, other examples of geo-computation clients, etc...
browser-based geo-computation: GeoTIFF, GeoTrellis, ModelLab.

These web app increasingly rely on 'fat clients' and 'thin servers', instead of
the traditional: 'fat server', 'thin client' paradigm.
This trend can be attributed performance upgrades of the javascript runtime, and
to new browser features such as WebAssembly.
```

2

At the same time, a trend within web applications in general shows an ever growing preference for so called 'fat clients' and 'thin servers', as opposed to 'thin clients' and 'fat servers' \cite{panidi_hybrid_2015}. This means that certain responsibilities, such as navigation, increasingly are being performed by the client rather than the server. As a consequence, applications are more quick to react on a user's input, and web-applications

Geo-computation as a phrase is used to group all the various ways geodata is transformed, such as reprojection, translating between datatypes, or boolean operations.

The trend can be attributed to new browser features such as WebAssembly, and performance upgrades of the javascript runtime (V8, JIT compiler, etc.).

```
... Blurring the lines between browser-based application, and desktop / native
application.
```

This trend is visible within web GIS towards more complex clients and simpler servers . Within the field of geo-informatics, the move towards client-side heavy applications also caused academic and commercial interest for the prospect of \emph{client-side geo-computation}. --> Whereas before the client was mainly use to visualize end results (leaflet, Celsium), now applications arise allowing users to edit (ninja) and validate (cjval) geodata, directly from within the browser. EXTREME EXAMPLES: GEOTIFF.

```
third trend:

This is why we see that as soon as geodata is sufficiently findable, accessible,
interoperable, and reusable, the accessibility concerns of geodata start to move
more towards geodata _processing_ and _processing applications_.

```

3

The third recognizable trend within the field of web-GIS is a push towards more accessible tools. It is a direct result of the first two trends. The more accessible geodata becomes, the more accessible the *means to use* geodata must become in order to keep up. After all, FAIR geodata means little if the means to access, compute, or visualize geodata are not accessible. Chris Holmes already envisions how cloud native will lead to expand the geodata community, and expand geodata users. "(cloud-native geospatial) is what gets geospatial out of its niche" \cite{podcast}. We also see this happening with certain web applications like GeoTiff & ModelLab. Modellab in particular states: "*Widespread access to frequent, high-resolution Earth observation imagery has created the need for innovative tools [...] that will help individuals and organizations to effectively access, analyze, edit, and visualize remotely sensed data in transformative new ways without years of specialized training or ongoing investments in proprietary software and technology infrastructure.*". \cite{modellab}.

Web GIS applications are in a good position to provide these accessible geospatial tools. The previously named advantages of cross-platform publishing & requiring no installation give web GIS a lead in accessibility compared to desktop applications in this pursuit of "denichifying" geospatial tools.

Problem:

```

```

```
SYNTHESIZE PROBLEMS FROM THESE THREE TRENDS:

- Major Opportunities due to the trends
  - Name the positive changes

- Challenge:
  - Major changes needed.
  - re-examination of the role & architecture of web GIS

- Challenges:
  - What is the role of client web GIS in the era of cloud-native, regarding geo-computation?

-
```

When viewing the trend of Cloud Native Geospatial together with the advancements of web clients and the push towards accessible geo-tools, a pattern emerges: Major opportunities are coming to the field of Web GIS tooling. Increasingly powerful options for server-side and client-side computation are, or will be, available for developers. More data & more services are becoming more open, and the lines between cloud-based, desktop and browser-based applications are getting increasingly blurry.

These developments lead to a critical re-examination of both the **architecture** and **role** of web GIS applications, and this is particularly apparent when regarding geo-computation functionalities.

When OGC web-services are substituted by statically hosted, singular geodata files, all of their current geo-computation functionalities will have to be realized elsewhere. CRS-reprojections done by a WMS are a good example of this. These can either be facilitated by other, new web-services, or can be added to the responsibilities of (web) clients. The same dichotomy can be found in web-GIS applications aiming to make geo-computation capabilities more accessible to a wider audience. Developers of these applications will have to choose between hosting / paying for cloud-based geo-computation services, or adding these often heavy computations to client applications. The choice is also not a binary one. Hybrid approaches have proven to be beneficial \cite{csg_2016}, And as such, it is very likely that both architectures could operate in a complimentary manner.

In any case, with both the possibility of cloud-native geo-computation servers and powerful clients, it becomes clear that developers of web GIS applications will have to make important decisions. However, due to the novelty of these trends, we lack sufficient knowledge and examples to make these decisions in an informed manner.

This study recognizes A particular need to investigate geo-computation at the side of the client.

Web GIS tools offering the access and visualization of geodata have been out there for a long time, and are well documented. Web tools offering geo-computation in the form of analysis or transformations are much newer, and as such more uncertain.

The same can be said about the client-side. The cloud native geospatial movement can be relied upon to make a strong case for cloud-based geo-computation, and tools currently offering geo-computation (Google Earth Engine, ModelLab) use mostly server-side geo-computation.

Browser-based geo-computation is a newer idea and less often used, but this does match the aforementioned trend of fat clients, and the purpose of the Cloud Native movement to relieve data providers from the burden of setting up and maintaining multiple web services. It could also provide situational benefits. It is cheaper to perform client-side geo-computation on the machines of end-users, instead of maintaining or otherwise paying for additional geo-computation web services. \STUDY FROM 2019. Performing calculations close to the client could also reduce web-traffic. This might be very important if the bottleneck of an application is sending data back and forth between server and client.
Finally, the prospect of Cloud Native geodata, combined with powerful geo-computation additions to accessible web clients, opens up interesting new functionalities and use-cases for browser-based GIS. \STUDY FROM 2019

The challenges of this second option are that browser-based geo-computation (BBG) might not be as performant as server side geo-computation, \STUDY FROM 2015, 2016, and that WebAssembly will have to be used to make use of industry-standard geoprocessing libraries such as Proj, Geos, GDAL, or CGAL.

Additionally, how to present the complex endeavour of geo-computation in the format of an accessible web application is unknown, and which use-cases might benefit from BBG also remains to be seen.

with the general movement moving to the cloud, it becomes even more important to look at alternatives.

- If study fails: we know that the move to the cloud is a wise one
- If study succeeds: we learn that in certain scenarios, client-based geo-computation is advantageous, and not ALL geo-computation should be done in the cloud.

this is why we study the second option.

This study

```
OVERVIEW

- Particular Goal
  - How to build a web app which fully runs in the browser
  - How to perform geo-computation with existing libraries in the browser?
  - How to make geo-computation accessible?
  - How could this use cloud-native geo types (COPC).
  - Who benefits from THIS web app?

- Larger Goals:
  - Studying the technology landscape of the browser.
  - Examine the role of web GIS in an future era of cloud computation
  - Helping the decision of choosing Cloud native geo-computation and / or Client-side geo-computation.
  - Seeking new roles for the client in an era of cloud computation

Particular goals are a proxy to give data, in order to maybe answer the larger goals

- DO NOT MAKE AN ARGUMENT ANYMORE FOR browser-based geo-computation.
- DO NOT FULLY EXPLAIN GEOFRONT

SKETCHES
```

```
this study seeks to investigate the possibility,
implementation and ramifications of the second option.
While both options are entirely possible,
```

This thesis covers the design, creation, and evaluation of a web application called "GeoFront". The application plays into the trends of "Cloud-native geospatial", "Fat-client web applications" and "The need to denichify geo-computation". GeoFront is a fully browser-based geo-computation tool focussed on point clouds. It enables users to configure geodata transformation pipelines using visual programming. It has the capability to make use of almost any rust-based geo-computation library, as well as use specific C++ libraries like CGAL and GDAL. It is similar in purpose to ModelLab or FME, but fully browser-based, and focussed on point clouds.

By conducting this research, this thesis aims to gain partial insight into the possibilities and limitations of the broader topic of browser-based geo-computation. The thesis seeks to understand how web GISS might profit from from the opportunities presented by these trends. It wishes to equip developers and decision-makers with the data, tools and ideas necessary to navigate the coming era of cloud-native geospatial data and accessible geo-computation.

Research Question:

```
% GOAL: - explore new use-cases of the web for interactive geo-processing.
    %   - explore the 'fitness' of the web for many different geo-computation use
cases.
% 2. Study to which end different needs can be fulfilled / different use-case
requirements can be met
This study seeks to investigate browser-based transformation of geodata,
as well as how browser-based GISS might profit from from the new opportunities.
This study intends to discover if contemporary web technologies can facilitate a
full-scale client-side geo-computation tool by seeking an answer to the following
question:
This study investigates browser-based transformation of geodata,
as well as how
```

"How to design and implement an accessible geo-computation web application by only using browser-based technologies?"

Sub Questions

```
To answer main research question
- we need insight into "accessible"
- We need insight into browser-based technologies
```

The main research question is subdivided into the following components:

- Which browser-based technologies do we need for effective geo-computation, and how must they be applied?

- What features are required for a browser-based geo-computation client?
- How can browser-based geo-computation make use of existing geo-computation libraries?
- In which use-cases can GeoFront be regarded as accessible?

Not Include

statements on the broader topic of browser-based geo-computation

- We will save this for the discussion part of this thesis.

Related work

```
...
[JF]: This writes itself as soon as I get to it
      We do need to add some info from the introduction, in order to make the
intro smaller, but still makes us able to refer to it during methodology or
implementation
...
```

Methodology:

How are we going to conduct this study?

Design Choices

This chapter explains the various choices which make up geoFront, and where these decisions came from.

Browser based / client-side

Reduce web trafic

Geofront

% Accessibility % - Shareability % - Reproducibility

% Interactivity % - Visual programming % - Visual feedback & insight % - Sliders

% # Some Introduction

% Web applications have many benefits over native applications % - cross-platform % - easy to distribute % - findable, accessible % - ...

% For these reasons the web has become a popular target for many applications, including geoweb applications. % - Tools like Leaflet and Celcium are widely used.

% A clear trend within these geo-web applications, is a push for increasingly complex applications. % - Such as the ninja cityjson web viewer % - Proofs that the web can not only be used as viewer or end-user application, % but also for geodata analysis and editing. % - Both for end-user tools, and tools used by developers. % - Also, Modern web tools such as webassembly allow us to run native geo-analysis applications directly in a web browser % - Leading to tools such as cj-val, a 3D city validator.

% This thesis seeks to push this trend further, % and explores to what end the capabilities of the contemorary web can facilitate the needs of geo-informatics in general, % and geodata processing in particular.

% We introduce GeoFront, a web based, open source, geodata processing and visualization tool. % It enables users to configure geodata pipelines using visual programming. % Users are able to make use of native geo-computation tools such as CGAL, visualize and debug both end results and in-between results.

% The input and output of the functions exposed by these libraries

% the aspect of interactivity is

% - The web gives us great \emph{Accessibility}, and making it client-side promises us great \emph{Interactivity}

% Browser based \ac{geo-computation}, hereby known as \ac{bbg}, ...

% 'interactive geo-processing' % - Insight % - Reproducibility

Web App

Geofront is designed as a client-side web application. On one level, the choice for native application or browser-based application is not very significant, as the line between these two types of applications continues to blur. Tools like Electron bring a

However, as opposed to a native application as an experiment on accessibility.

\cite{kulawiak_analysis_2019, panidi_hybrid_2015, hamilton_client-side_2014}. Interactive geospatial data manipulation and online geospatial data processing techniques where described as "highly valuable trends in evolution of the Web mapping and Web GIS" (\cite{panidi_hybrid_2015}).

Use Cases

Four use-cases are envisioned for browser-based geo-computation in an era of cloud-native geospatial data and tools.

1 Educational

```
- Web Demo
```

2 Academic

```
- Open Science
- Jupyter notebook
- Reproducibility
```

3 End user app

```
- Exposiign
- Retrieval, Processing, visualization in a browser app
```

4 Cloud Compute Prototyping

```
- Geoflow
- Sampling
- Trying different parameters
```

Implementation

```
...
```

Discussion

```
...
```

Conclusion

```
...
```