

# Web Maps Research Summary

2022-03-28



<b>Background</b>	<b>2</b>
<b>Research</b>	<b>2</b>
Static analysis of HTTP Archive dataset	2
Survey	3
Expert Interviews	4
Browser and Map Implementer Interviews	4
<b>Potential Solutions</b>	<b>5</b>
Start with MapML	5
Start with Model	6
Iterative approach	6
<b>Recommendation</b>	<b>6</b>

# Web Maps Research Summary

2022-03-28



## Background

According to the static analysis of the HTTP Archive, 16% of web pages use maps. This is significant compared to usage of other elements such as the [picture](#) element, [video](#) element, or a [youtube](#) embed, coming in at 5.9%, 4.1%, and 10.2%, respectively. Of those, ~22% use complex maps (multiple pointers, popups, polygons/polylines). The three most important web map features noted in our survey are, in order, *add points*, *add polylines and/or polygons*, and *change the map style*.

Of the 16% of web pages that use maps, ~22% use complex maps (multiple pointers, popups, polygons/polylines). We also interviewed map experts and surveyed web developers. The pain points were around performance, accessibility, and authoring (more details in appendix). Based on this we think there is a need for standardized mapping features on the web.

## Research

### Static analysis of HTTP Archive dataset

Bocoup analyzed the HTTP Archive dataset (7,9 million web pages) for web maps usage, including Google Maps, Mapbox, and Leaflet usage as APIs, embeds and static maps.

Key findings:

- Around 16% of all pages use web maps.
- Of those, between 19.5% and 25.8% (the range accounts for possible overlap) of the maps found use plotted geographic features such as markers, pop-ups, and polylines/polygons.

These data points suggest that improvements made to the web platform should take into account both simple and complex use cases. To support this assertion a broad feature set is included in our detailed roadmap and will continue to be defined through additional research (e.g. MDN survey to find pain points).

Full report: [Final: Web Maps for HTML-Research Document-Feb 2022](#)

# Web Maps Research Summary



2022-03-28

## Survey

We held a survey for web developers and map library authors. It was shared on Twitter by @bocoup, @moz hacks, @NRCan, and @opengeospatial, on Slack IIF/Map CG, A11Y, NewsNerdery, and NACIS. The survey received 78 responses.

### Key findings:

- Pain points:
  - Implementing interactions is the most difficult step when making maps.
  - Around half of the respondents say there are not enough beginner-friendly mapping solutions available.
- Features: The 3 most important web map features are, in order:
  - Add points
  - Add polylines and/or polygons
  - Change the map style
- Accessibility:
  - 84% of respondents don't have the necessary tools to make a11y compliant maps.
- Performance:
  - Web maps are slow to load
  - Lag when panning and zooming
  - WebGL-based maps are power intensive
- Where can the browser do more heavy lifting?
  - Panning and zooming
  - Interactions
  - Accessibility
  - Understand CRS
  - Ability to apply CSS to map features

Quote from one respondent who is a map framework author:

“The performance is as good as it can be with script-driven parsing and rendering. The underlying map rendering engine (Leaflet) is further not capable of rendering vector tiles, so advanced rendering use cases aren't possible with our custom elements (e.g. using OffscreenCanvas). Further,

# Web Maps Research Summary



2022-03-28

since there are no map feature elements in HTML, in order to apply CSS to the rendered output, map feature content must be transformed to SVG via script, which tends to also perform poorly. It would be ideal to simply target map feature content with CSS selectors and thereby apply cartographic rules using the CSS standard.”

Takeaways: Lead with accessibility and performance.

## Expert Interviews

We interviewed 6 experienced web map makers and 1 non-sighted user of web maps. Key findings:

- There is a need for a11y compliant maps that none of the mapping frameworks are doing an adequate job addressing
- It should be easier to change the map projection or switch between map projections on the fly
- Map creators need to be able to implement custom styles on the map
- The barrier to entry for creating maps on the web is high
- Integrating maps into larger applications can be difficult
- Creating and setting up private maps is difficult

Takeaways: Prioritize performance shortly after accessibility.

## Browser and Map Implementer Interviews

During the research process, standards stakeholders suggested reusing parts of the <model> Element, specifically the privacy preserving `interactive` property. Based on this we recommend a third initial implementation area **focused on privacy**.

Takeaways: Increase buy-in from map frameworks, and take an iterative approach that uses research and polyfills, and drives fixes issues in web frameworks as part of the process.

# Web Maps Research Summary

2022-03-28



## Potential Solutions

The following outlines three directions for implementing web maps that were raised in our interviews: iterative build starting with new capabilities, extending the `<model>` element, and starting with MapML.

To begin we will review the implications of each by outlining high-level pros and cons.

### Start with MapML

If we start with MapML, we'd have a new mime type, media type, or doctype and new root element `<mapml>` ([#structure](#)) and a set of elements

- **Copies from HTML:** `<head>`, `<title>`, `<meta>`, `<body>`, `<link>`, `<input>`, `<datalist>`, `<label>`, `<select>`, `<option>`, `<image>`, `<base>`
- **New Elements:** `<tile>`, `<extent>`, `<feature>`, `<featurecaption>`, `<properties>`, `<geometry>`, `<coordinates>`

### Start with Model

If we use `<model>` for web maps, then

- Use wholesale
- Use parts e.g. interactive through the below iterative approach

### Iterative approach

If we start with an iterative approach focusing on new capabilities, then we would conduct further research in target areas, fix web frameworks, identify gaps to standardize, polyfill them, and standardize them in pieces, and then standardize high level Elements as a final step.

An iterative and incremental approach affords more opportunities for input and review, and provides map library and framework authors, web developers, and users access to features and APIs sooner, which can garner widespread and tangible support for new mapping Elements, and a new high-level map Element. This approach also makes space for implementers to evaluate and prioritize more

# Web Maps Research Summary



2022-03-28

effectively, and implement using existing workflows (e.g. [The Chromium Project: Launching Features](#)).

## Recommendation

We recommend the iterative approach, outlined in the Maps for HTML Roadmap March 2022.