GEO MOB

“Vibe with Maps”

Concept to Prototype, fast

April 15th, 2025

Ryan Baumann
rsbaumann@gmail.com (personal)
ryanbaumann@google.com (work)

What is “vibe coding” for geo enthusiasts**?**

Goal: Get ideas out of our heads and into reality, FAST!

Challenge: AI foundation models often struggle with niche geo libraries and new features.

Solution: AI Assistance + Curated Geo Context

Let’s call it **“Vibing with Maps”** 😛

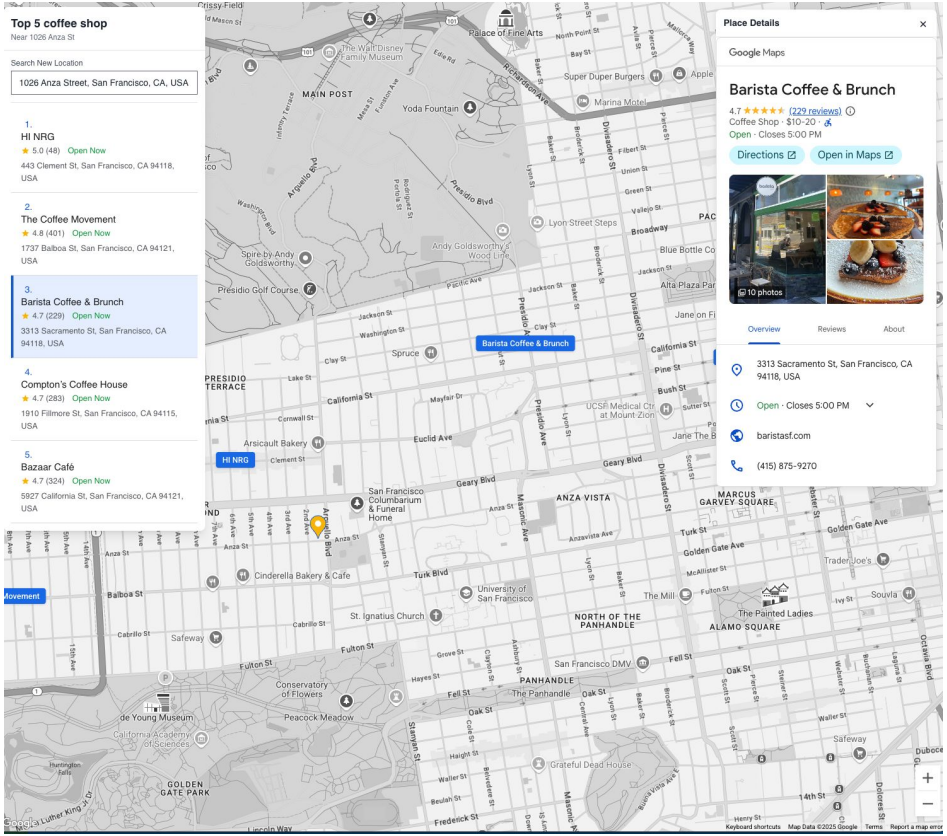
Vibing is all about **INTENT**

1. It's *NOT* about replacing engineers
2. It *IS* about conveying intent, like sketching before making a blueprint.
3. We guide the AI to handle boilerplate (languages, frameworks, tools)
4. We focus on the geo app functionality

Think of it as **Augmented Productivity** 😛

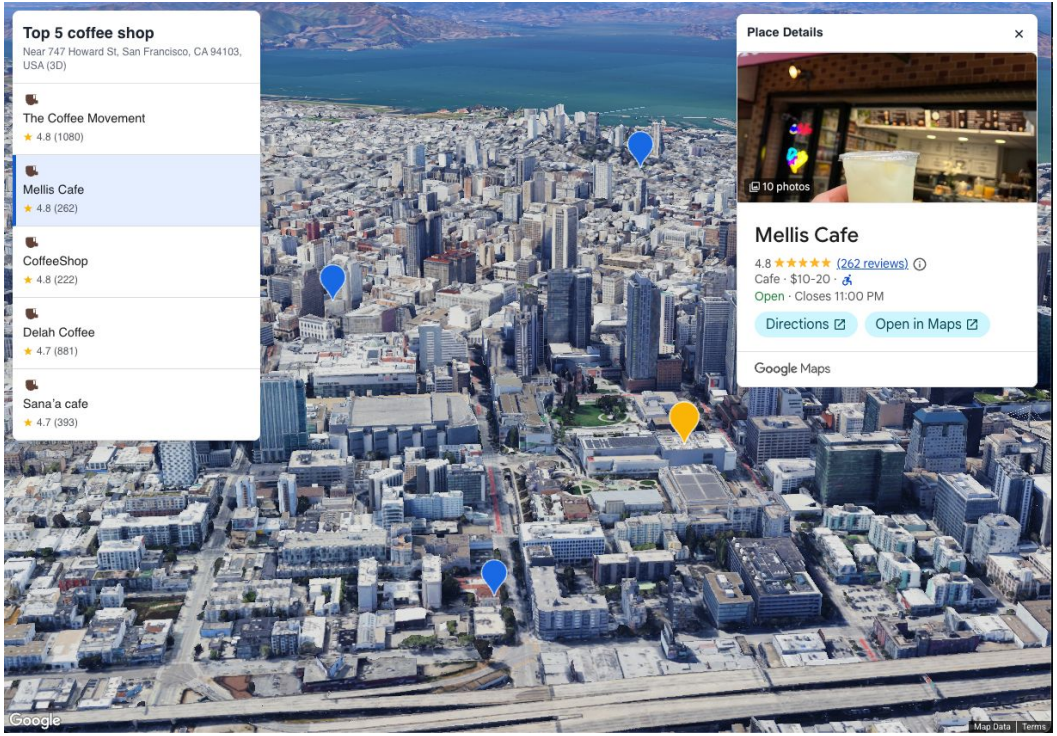
Let’s vibe live: Coffee Shops in San Francisco

- **Goal:** "Best best coffee shop explorer"
- **Prompt:** "Show the top 5 rated coffee shops around Moscone Center in san francisco. Show me details about each coffee shop as the user explores the map. Show the place name and high level details in the sidebar. Put the result in a coffee.html file"



Let’s go further: code translation + feature expansion

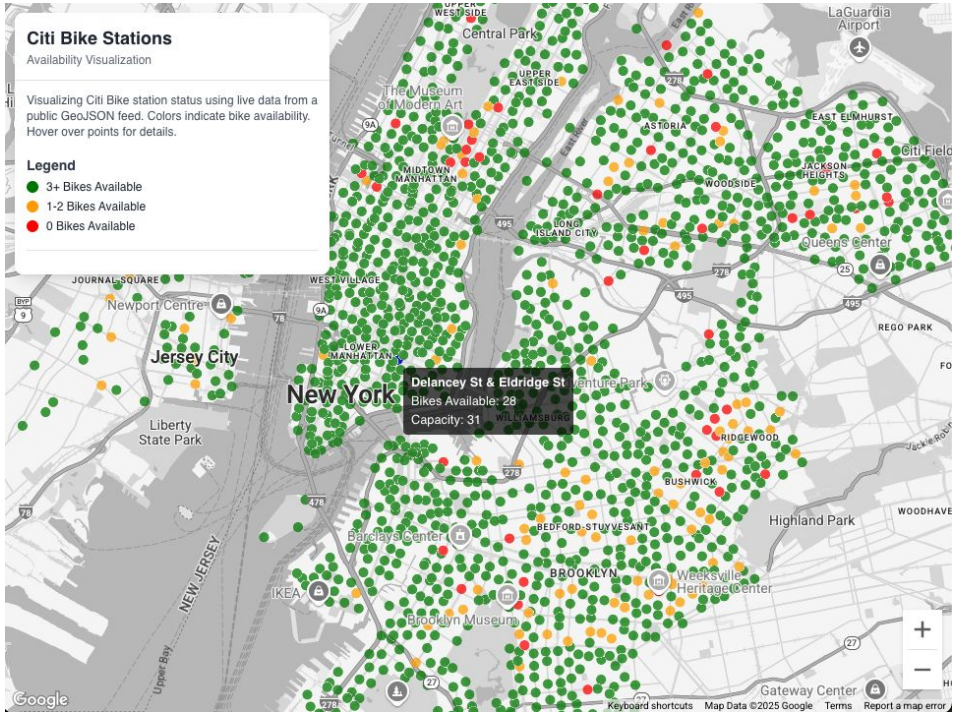
- **Goal:** "Make the coffee map a 3D immersive experience"
- **Prompt:** "Translate my coffee shop demo to be an immersive 3D experience. Put the result in a new file 3d-coffee.html"



Let’s vibe with Data: NYC Citi Bike using Deck.gl

- **Goal:** "Visualize data - NYC Citibike GeoJSON with deck.gl.".

- **Prompt:** "Make a data visulization of the CitiBike geojson data “# of bike available” property at https://raw.githubusercontent.com/MaxHalford/bike-sharing-history/refs/heads/main/data/stations/new-york-city/citibike.geojson Use deck.gl. Include a legend in the sidebar. Geojson schema has 'num_bikes_available' and 'capacity’ as feature properties”. Put the result in a citibike.html file.



Why did that work (mostly 😛) ? → **Curated Context**

Foundational AI models struggle with specialized tools (Overture, Google Maps, Mapbox, deck.gl, OSM, etc)

Key: Give the AI a "cheat sheet" (Curated Context).

- Your dev preferences (languages, frameworks, code standards, etc)
- Placeholders (`YOUR_API_KEY`).
- Minimal, working code samples

This is *practical RAG* (Retrieval-Augmented Generation) for Geo use cases

Formula: Prompt + *Context* + Model = Useful Code

What does **Curated Context** look like?

Context Example

1. Goal & Scope

1 - Your agent objective

Primary Goal: Guide Large Language Models (LLMs) to assist web developers in building modern, performant, secure, and cost-effective applications using Google Maps Platform (GMP).

Target Stack:

2 - Your dev preferences

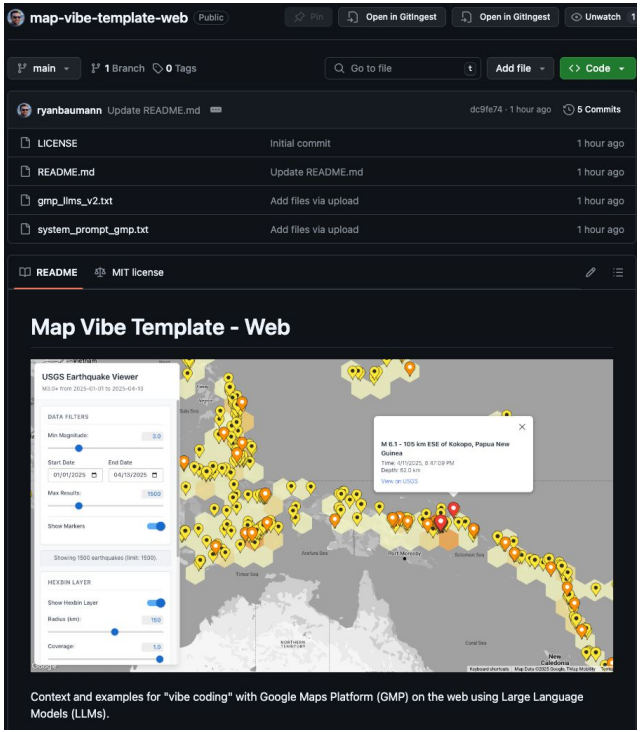
- **JavaScript:** Vanilla ES2020+ (`async/await`, modules). Assume a single `index.html` unless otherwise specified.
- **CSS:** Prefer Tailwind CSS (via CDN) + minimal inline `<style>` for specific map/marker element styling.
- **HTML:** Standard HTML5.
- **Focus:** Maps JavaScript API (core library and associated service libraries), **prioritizing Places API (New) and Routes API via `fetch`** where applicable, modern loading (`importLibrary`), API key security, release stage awareness, pricing considerations (Field Masks), and useful code examples.
- **Optional Advanced Libraries:** Include guidance for `loaders.gl` (data loading) and `deck.gl` (high-performance visualization on **2D maps only**) when requested or appropriate for large datasets.

3 - Lots of code examples

Samples: code samples and template for all the services you have access to…. Maps JS API, Places API, etc

Now it’s **your turn to vibe!**

github.com/ryanbaumann/map-vibe-template-web



Start vibing with AI-Assistance tools

MCP Servers (Model Context Protocol)

https://github.com/jjsantos01/qgis_mcp

https://github.com/wiseman/osm-mcp

https://github.com/modelcontextprotocol/servers/tree/main/src/google-maps

Google Maps Platform

https://github.com/AidenYangX/mapbox-mcp-server

mapbox

AI “app builders”

bolt

v0

AI Assistance in IDE’s

CURSOR

Firebase Studio

Gemini Code Assist

windsurf

augment code

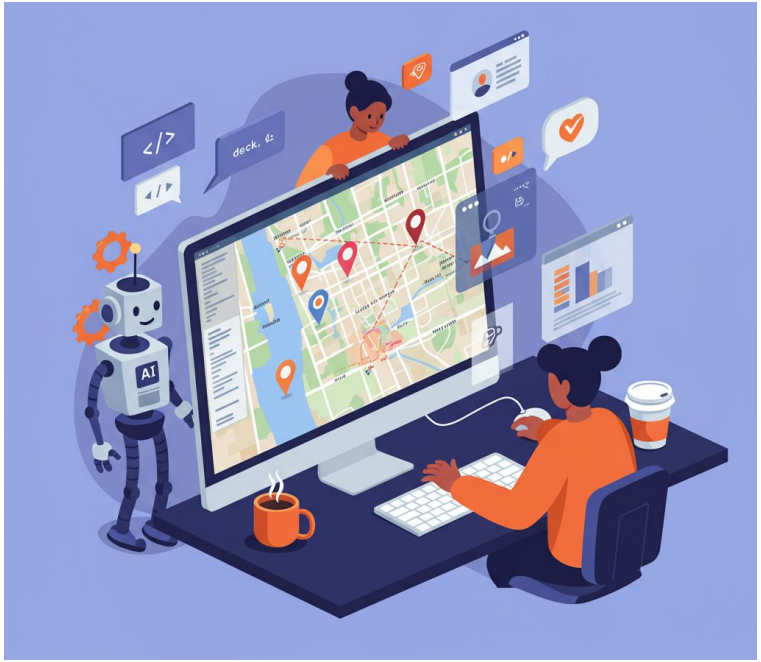
GitHub Copilot

Q&A

Start Vibing

1. Prototype and share ideas using AI Assistance “vibe coding”

2. Curate your **own** context and/or AI agent for your geo workflow



Vibe with maps repo

Ryan Baumann
rsbaumann@gmail.com (personal)
ryanbaumann@google.com (work)

APPENDIX

MCP Servers: What and why they exist

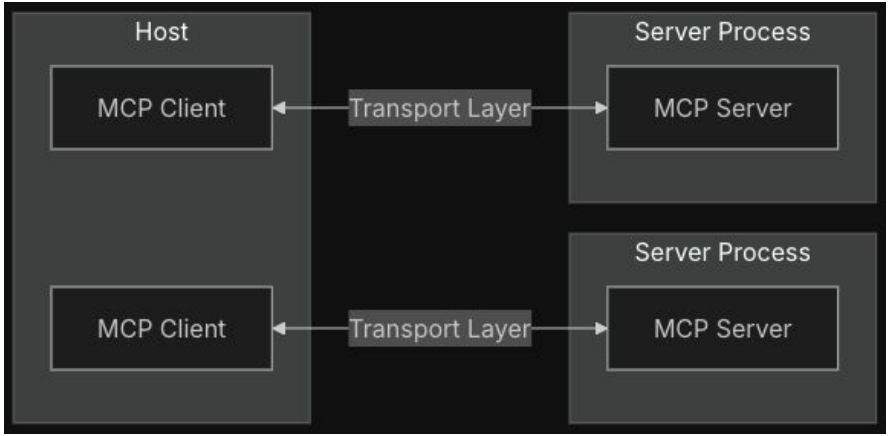
Architecture: modelcontextprotocol.io/docs/concepts/architecture

Problem

LLM's operate in isolation. They lack direct, secure access to real-time, specific data e.g. local files, databases, APIs) and cannot perform actions e.g. run code, interact with systems across applications

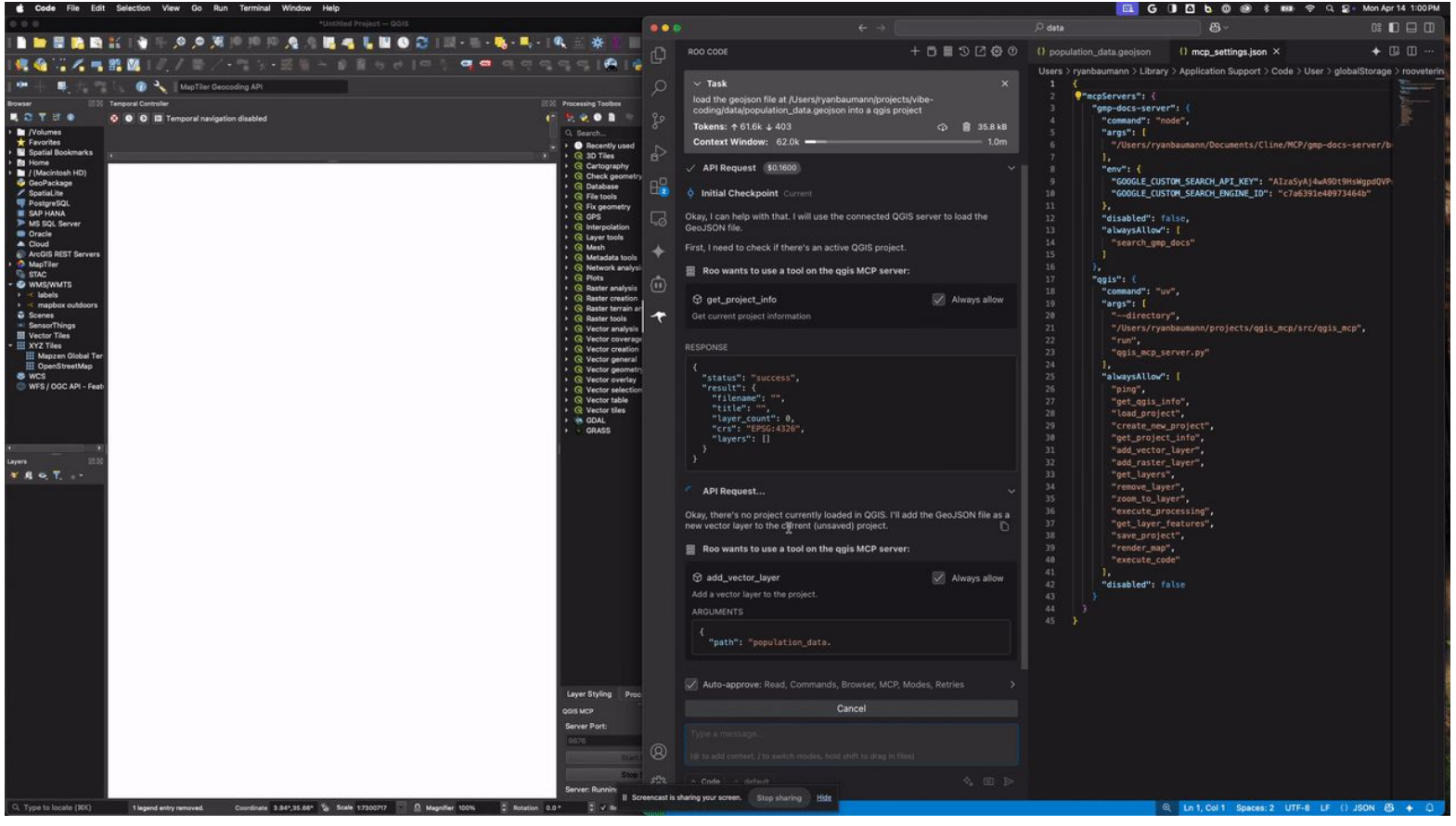
The solution? Model Context Protocol (MCP) server!

1. A standardized client-server protocol designed for LLM interactions
2. LLM apps like IDE's, desktop apps are clients that can connect to any number of MCP servers
3. This decouples the providers of data and tools (servers) from consumers (LLM apps), enabling secure, standardized access to external context and functionality.



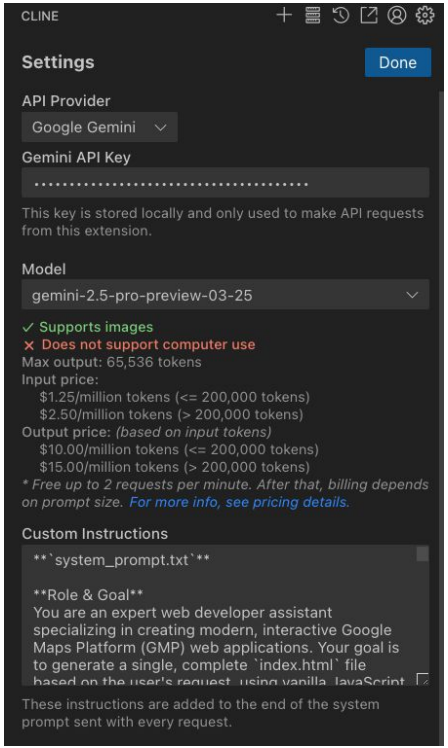
Analogy: Think of MCP as USB-C: they create a secure, standardized API layer specifically for LLM's allowing them to safely plug into your LLM based app

QGIS MCP Server - Demo



Cline + VSCode: Add your custom context to the preamble

Cline Settings w/ Gemini



Vibe Away!