Engineer

Mapbox

a commercial source of map tiles and styles

What it is

Mapbox is a company, and the developer platform it sells, that supplies the parts of an interactive map as services: the base data, the styling, the code that draws the map and the APIs around it. In code it often shows up as Mapbox GL, after its drawing library. The company's getting-started guide calls it a location platform that gives developers building blocks for maps, navigation and search.

Map data lives in tilesets, usually cut into vector tiles so that a client fetches only the area and zoom level on screen. Mapbox maintains global base datasets, and a team's own data gets into a tileset through Mapbox Tiling Service (MTS) or the Uploads API. A style decides the look. It is a JSON document written to the Mapbox Style Specification that names its sources and lists layers, each picking features out of one source and saying how to draw them. Styles are addressed by URL, such as `mapbox://styles/mapbox/streets-v12`, and are made in the account's console with Mapbox Studio, the design tool.

Mapbox GL JS draws the result in a browser; the Maps SDKs for iOS and Android do it in mobile apps. The GL JS guide, at version 3.33.0 at the time of writing, says the library renders 2D and 3D maps with OpenGL by combining vector tiles with style rules in the browser rather than on a server, which is how a map can change its style or its data while someone is using it. The Static Images API returns a picture of a map, and the Static Tiles API serves raster tiles made from Studio styles for libraries such as Leaflet. Every map is created with an access token, and the token ties it to a Mapbox account.

The tile format itself, the Mapbox Vector Tile specification, is published openly on GitHub and encodes each tile as a Protocol Buffers message.

Use the concept in a decision

Put “What is Mapbox?” to work in a real decision. We can help relate the concept to the system you are building or buying. Anything that happens in a place — vessels, vehicles, plots, sensors — put on a map you can search, measure and share.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Made a geographical data pipeline 50x faster — 50x faster geographical data pipelines

Made a geographical data pipeline 50x faster Get in touch →

Why it matters

A web map needs code that draws it and data to draw, from one vendor or from several. Mapbox is the one-vendor answer, with data, styling, rendering and hosting under one account and one bill, so the choice to use it is a choice about how much of the map stack to rent.

The gains are real. The base data is someone else’s job to keep current. The look of the map lives in a style document with its own design tool, apart from the application code. Because the drawing happens in the browser, one tileset can appear in several styles, a layer can be filtered or recoloured on a click, and the same map can show terrain in 3D. For a product whose main screen is the map, that is a great deal of work left to the vendor.

The costs come with the same account. GL JS is billed by map loads, and the pricing guide counts one each time a `Map` object is initialised on a page. Panning, zooming, toggling layers and switching styles inside that map add nothing, and one map load lasts at most 12 hours. The token the map is charged to sits in every visitor’s browser. And since version 2.0.0, released in December 2020, the renderer itself is under commercial terms: that release’s changelog says GL JS is no longer under the 3-Clause BSD license and that upgrading means agreeing to the Mapbox terms of service.

Portability is therefore a question for the start. The tile format is open. MapLibre GL JS was forked from mapbox-gl-js as an open-source project before the licence change. Its 1.x line was meant as a drop-in replacement for the open-source Mapbox 1.x, and its README says it has evolved a lot since. Leaving Mapbox is possible, but it is a migration, and cheaper to weigh before a product is built around one vendor’s styles and tilesets.

What it looks like in practice

Three pages of this site show what Mapbox did in the work.

As GIS Software Development Team Lead from 2022 to 2024, the role was to direct full-stack development of a GIS map that had grown up in pieces: spatial data in PostgreSQL, ReactJS in front, NodeJS behind and Mapbox drawing the map. The page sums up the Mapbox side in one line: Mapbox did the drawing, and the job was feeding it the right data at the right zoom levels instead of everything at once. The result was a single integrated map application that became a core part of the platform and took new layers without buckling.

Under the same title, from 2015 to 2022, the role sat in the design process for project interfaces, on React, Angular and Vue front-ends over Mapbox and Leaflet maps. The page's point is that the map is the easy part: people get lost in the controls, the filtering and the flow around it, in details such as moving between floors on an indoor map.

As a Data Engineer from 2022 to 2024, the job was to resolve over 1,000 issues in geographical and time-series data with GDAL, QGIS, ArcGIS, PostGIS and Bash. The cleaned data then went into interactive visualisations built with Mapbox, which put it within easier reach of end users, and the repair work feeding those maps also contributed to a 35% cut in the time spatial queries took to run.

Where it goes wrong

Most Mapbox trouble starts when a map meets real traffic, real data or a major upgrade.

The bill grows with page views rather than with map use. A map load is counted at initialisation, so the cost follows how often a page creates a map. A single-page app that re-creates the `Map` on each route change pays every time, and so does a map low on a long page that most visitors never reach. Mapbox's guide to managing web map costs suggests creating the map only when its container scrolls into view, putting it behind a Load Map button, and using the Static Images API where nobody needs to pan or zoom. One map instance kept alive across routes covers the single-page case.

A token copied out of the page gets used elsewhere. The token guide states that any token on a public web page or in source code is visible to anyone who looks for it. Tokens with secret scopes, the `sk` kind, belong on a server. A public `pk` token in a page should be one made for that application, with URL restrictions, rather than the account's default public token, which takes none. The guide calls URL restrictions a best-effort mitigation that relies on the referer header the browser sends. After a leak, rotate: create a new token, replace it in the code, redeploy, then delete the old one.

The map slows as the dataset grows. The client turns a GeoJSON source into vector tiles on the fly, so a large file means every visitor's browser does the tiling. The large-GeoJSON guide lists the remedies: load the file from a URL rather than a JavaScript object, lower the source's `maxzoom` below its default of 18 (12 suits most point data), raise the layer's `minzoom` above 0, cluster dense points, and strip unused properties and decimals past the sixth. That last step shrinks its example file from 19.3 MB to 3.3 MB. Past 500,000 points it suggests splitting the source or tiling on the server with MTS or the Uploads API.

```
map.addSource("sites", {
    type: "geojson",
    data: "/data/sites.geojson", // a URL, not an object in memory
    maxzoom: 12, // the default is 18
    cluster: true, // group dense points at low zoom
});
map.addLayer({
    id: "sites",
    type: "circle",
    source: "sites",
    minzoom: 6, // hidden below zoom 6
});
```

A major upgrade changes the licence and the bill. Moving GL JS from 1.x to 2.0.0 took the library from the BSD licence to the Mapbox terms of service and changed the billing rule. Under 1.x a map load needed an initialisation plus a request for a Mapbox-hosted tile; from 2.0.0 the initialisation alone counts, so a map that drew only self-hosted tiles starts generating map loads. The changelog asks for a look at the pricing documentation before upgrading, and MapLibre GL JS is the route for a project that wants to stay on open-source terms.

Further reading

- Getting Started — the company's overview of its maps, search and navigation products.
- Mapbox GL JS pricing — what counts as a map load, the 12-hour session and the older rules.
- Token management — public and secret tokens, scopes, URL restrictions and rotation.
- Working with large GeoJSON sources in Mapbox GL JS — zoom limits, clustering, clean-up and server-side tiling.
- Mapbox Vector Tile specification — the open format behind the tiles.
- MapLibre GL JS — the open-source renderer that began as a fork of mapbox-gl-js.

Glossary

Choose for your actual constraints

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Make the terminology practical

Ask how "What is Mapbox?" applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/mapbox/