



# Base Maps

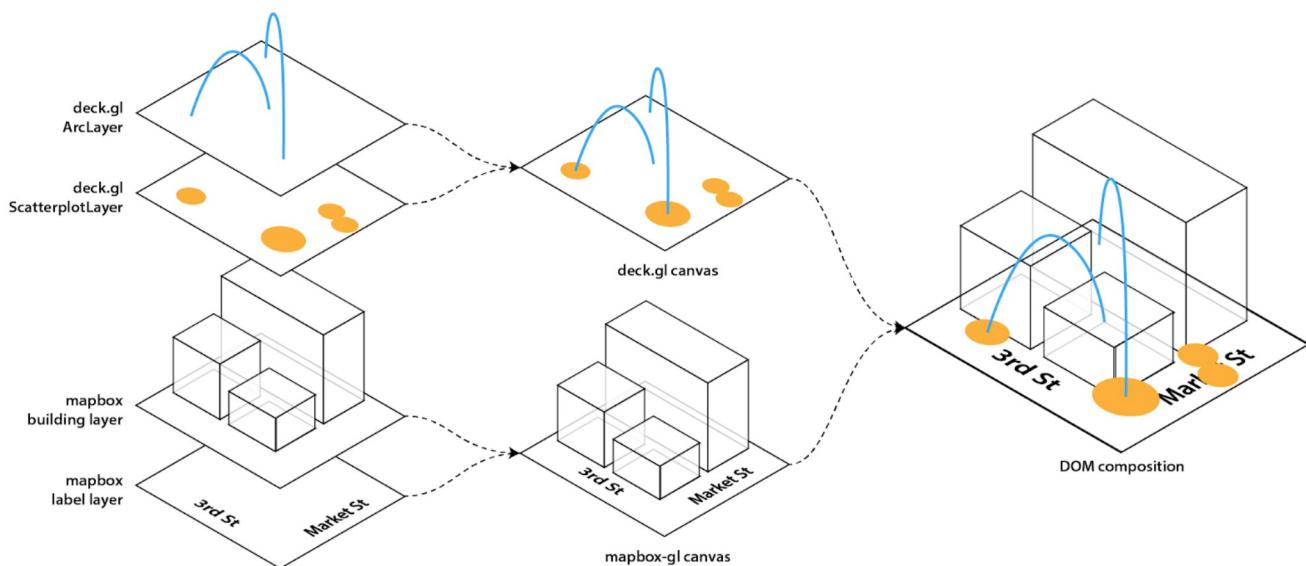
While deck.gl works independently without any map component, when visualizing geospatial datasets, a base map can offer invaluable context for understanding the overlay layers.

## Base Maps Renderers

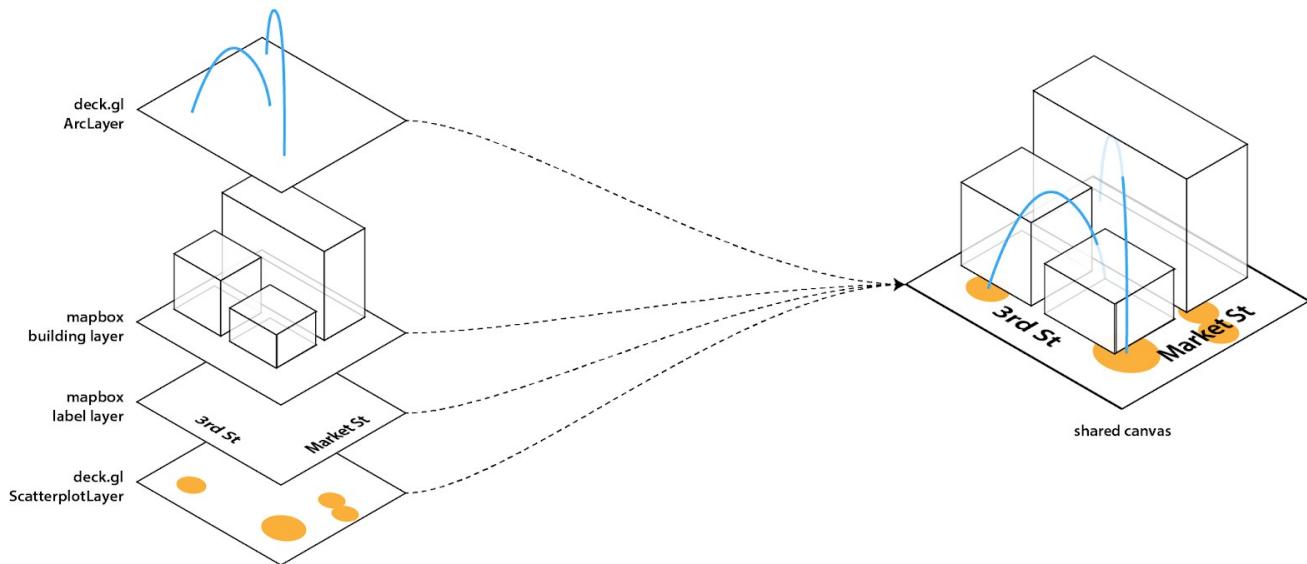
deck.gl has been designed to work in tandem with popular JavaScript base map providers. Depending on your tech stack, deck.gl's support for a particular base map solution may come with different level of compatibility and limitations.

There are two types of integration between deck.gl and a base map renderer:

- **Overlaid:** the Deck canvas is rendered over the base map as a separate DOM element. Deck's camera and the camera of the base map are synchronized so they pan/zoom together. This is the more robust option since the two libraries manage their renderings independently from each other. It is usually sufficient if the base map is 2D.



- **Interleaved:** Deck renders into the WebGL context of the base map. This allows for occlusion between deck.gl layers and the base map's labels and/or 3D features. The availability of this option depends on whether the base map solution exposes certain developer APIs, and may subject the user to bugs/limitations associated with such APIs.



Library	Pure JS	React	Overlaid	Interleaved	Docs
ArcGIS API for JavaScript	✓	✓		example	<a href="#">link</a>
Google Maps JavaScript API	✓	✓	example	example	<a href="#">link</a>
harp.gl	✓		example		
Leaflet	✓		example		
Mapbox GL JS	✓	✓	example	example	<a href="#">link</a>
MapLibre GL JS	✓	✓	example	example	<a href="#">link</a>
OpenLayers	✓		example		

It is also important to understand the difference between the JS library that renders the map and the map data provider. For example, you can use Mapbox GL JS with the Mapbox service, but also with any other service that hosts Mapbox Vector Tiles. When using a base map, be sure to follow the terms and conditions, as well as the attribution requirements of both the JS library and the data provider.

 [Edit this page](#)