
RST
ebook
How to create a
location-based
startup
without breaking
the bank?

Executive summary

Location-based service (LBS) development creates many and diverse business opportunities across different market segments. Whether you want to build an on-demand service, an indoor navigation system, inventory management software or simply offer a personalized loyalty program for your new retail customers – you will have to rely on a geodata-powered solution.

In fact, businesses are becoming increasingly creative in how they employ LBS to grow and expand.

This means that businesses looking to develop location-aware apps are making a wise move, however, they also have to be prepared to face a saturated market.

This ebook concentrates on helping you optimize the cost of building such an application. We are sharing some of the tips and tricks we learned throughout many years of cooperation with Trans.eu, one of the leading logistics platforms operating in Europe and Asia, as well as building numerous location-based startups.

In Chapter 1, we discuss the main components of location-based services. As you read through it, you will understand what decisions must be made as you pick the tech stack for your application.

A location-aware software often requires a map provider. **In Chapter 2** we guide you through some of the available options for ready-to-use APIs, as well as their open-source alternative.

Chapter 3 also looks at the available map providers, offering an overview of the potential costs associated with the usage of paid APIs. These may be staggering, so we explain in what cases it may be worth to use an open-source geographic data.

As a long-term official AWS partner, in **Chapter 4**, we outline a list of Amazon Web Services that location-based startups and enterprises will certainly find useful as they build their cloud-based solutions.

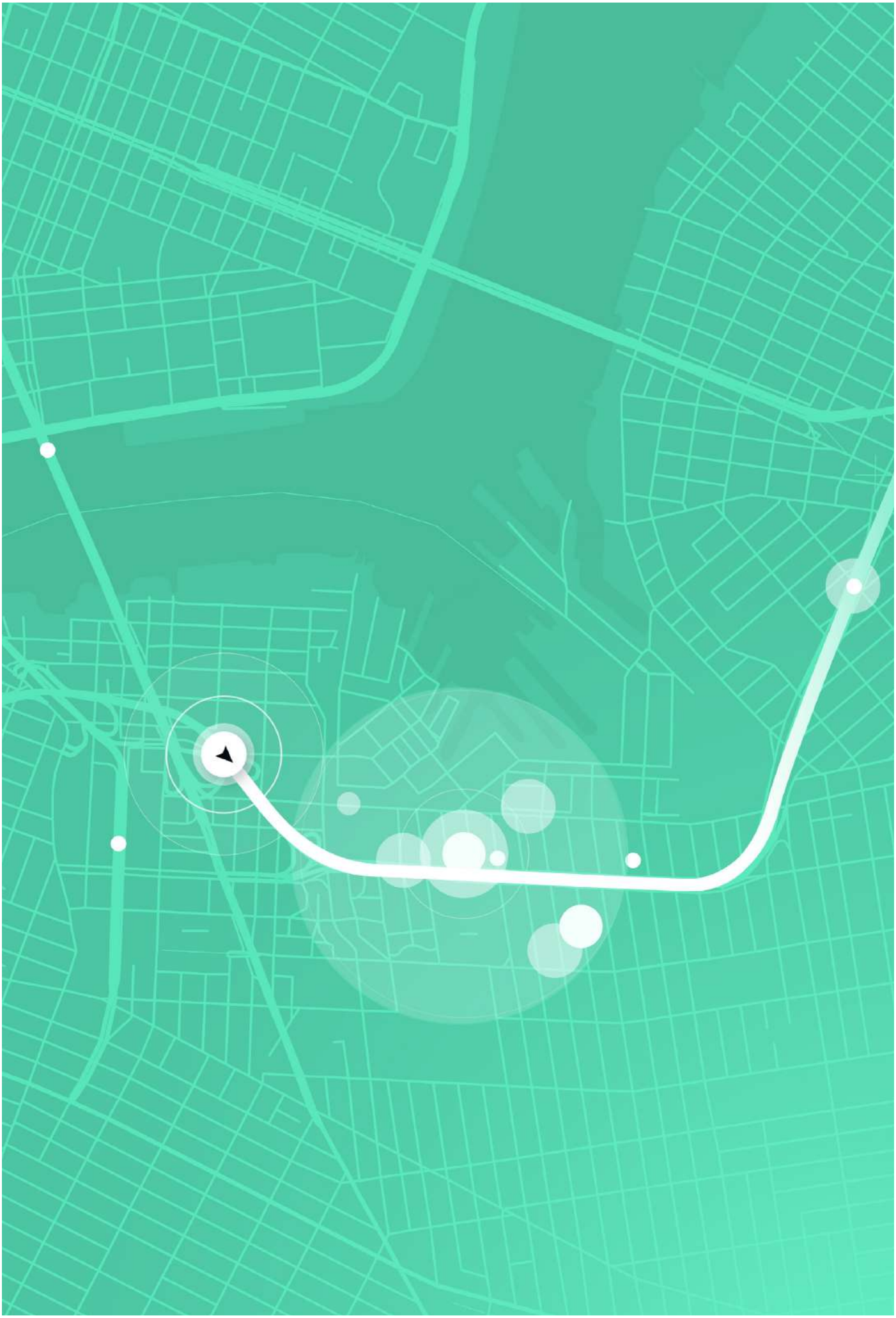
Location-based service development has its unique risks and challenges.
In Chapter 5 we discuss some of the most common ones and the ways to overcome them as you start your new location-aware project.

In Chapter 6, we explain how we built a cost-effective solution for Trans.eu. We managed to save over 400k USD in monthly map APIs usage by opting for building a custom open-source solution powered with open source map data. It may be a good option for businesses looking to launch a scalable platform aiming for high traffic volumes.

Finally, Chapter 7 provides you with an overview on the available open-source software, which can be of help in cutting the initial development costs. When adopted, discussed tools can provide you with advanced functionalities out-of-the-box without having to pay large bills early on.

We hope you find this ebook useful. If you have any questions about location-based service development, contact us directly at **offer@rst.software**. We'll be happy to discuss your new project and see whether our expertise can help you on the location-aware product development journey.

Table of Contents



Chapter 1

Location-based app development

the complete guide

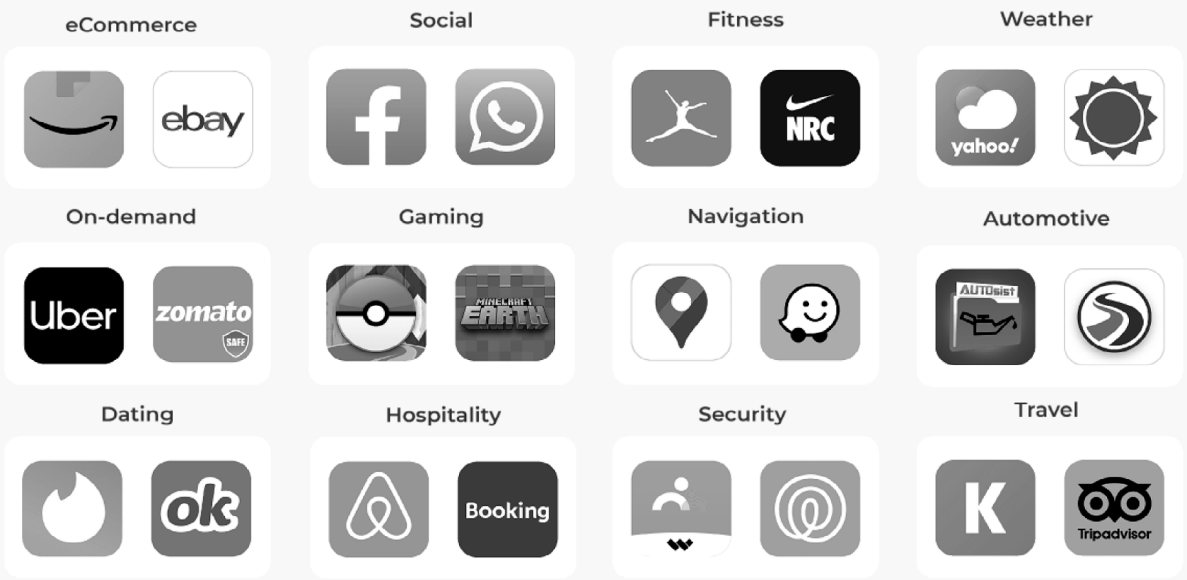
Most of today's mobile apps rely on location based services - those that require real-time location data to function. Travel, transportation and mobility industries obviously use such services heavily, but they are also becoming increasingly disruptive in retail, proximity-based marketing, healthcare, mobile workforce management, asset tracking, fraud prevention and even public safety.

According to AlliedMarketResearch, the global market size for this segment is expected to amount to $318.64 billion in 2030. A quick comparison to 2020 data, when it was worth only $36.35 billion, suggests that entrepreneurs who consider building a location-based startup may just be seizing the right opportunity. In this article, we explain everything you need to know about building location based startups or bringing your existing technology up to current standards.

Industries that rely on location based services

Without a doubt, location based services are making our lives more comfortable and efficient in a myriad of ways. Consumers can now track their delivery or check how far away their Uber driver is from the pickup location. Travelers can easily locate nearby hotels or attractions, runners can analyze the speed of training routines, social media and dating app users can meet and connect with new people across the entire globe, while gaming apps (such as Pokemon Go) can offer even more immersive experiences.

Here are some examples of popular location-based apps:



Types of location based technologies

Location based services provide Geographic Information Systems (GIS) and spatial data through different outdoor and indoor technologies. The purpose of your new application will largely determine the type of technology you have to choose. Let's take a quick look at the different choices available.

Outdoor geolocation technologies

- **GPS**

Global Positioning System is a network of about 30 satellite transmitters orbiting the Earth. Their radio signals can be intercepted by any device with a GPS tracker to determine its location and timing from these satellites. It's the most common geolocation technology that works on every mobile device which by default has a built-in GPS module. It's a quick way to get accurate location details. On the downside, it quickly drains the receiving device's battery.

- **A-GPS**

It stands for **Assisted GPS**, which is a combination of signals received from satellites and cell tower data. It's also a popular geolocation technology thanks to high speed and lower power consumption than traditional GPS. However, it may be more expensive than other options as network providers may charge an extra fee for cell tower data access.

- **Cell ID**

Cell ID may come into play when the GPS module is disabled. In such a case, a mobile device will rely on the radio signals of the operator provided by cell towers. The nearest mobile station will capture the device's coordinates to determine its exact location. This technology isn't as accurate as GPS.

- **Wi-Fi**

In location-based app development, **Wi-Fi** geolocation works either by using Received Signal Intensity Indication or online maps. The Wi-Fi network is used to determine location or track objects, although the results will be approximate.
As the Wi-Fi network covers a relatively small area, it is not as precise as the methods mentioned above, but generally works more effectively than Cell ID. This technology may be useful for entrepreneurs who want to build a location-based application for an urban area.

Indoor geolocation technologies

The global COVID-19 pandemic brought the momentum for real-time location systems (RTLSs) which provide location services based on indoor positioning technologies. They came in particularly handy for the industries who needed to track their goods or workforce in stores in the age of social distancing and allowed businesses to better manage their supply chains and sell products more effectively.

Beacons

Bluetooth Low Energy (BLE) technologies are particularly handy when it comes to indicating location inside buildings. They detect the signal from a smartphone and analyze it to determine the user's current location. Such devices are called beacons. They can be attached to pretty much anything throughout a building.

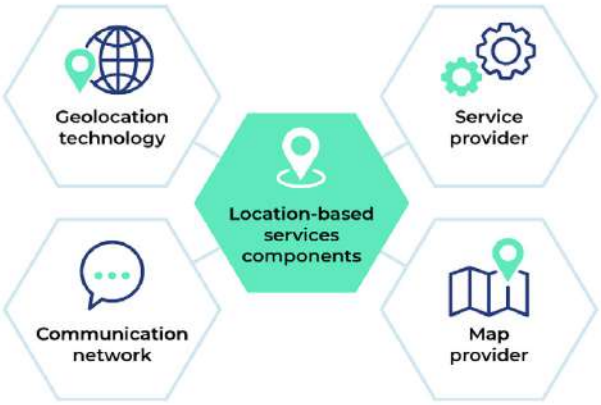
Geofencing

A geofence is a virtual "border". With GIS data, geofencing determines the time at which a person enters a specific virtual perimeter. The geofencing service triggers once a smartphone crosses that virtual border. When a user enters the geofenced area, their device will intercept communication via push notifications. This technology is used predominantly in retail and marketing.

How do location-based services work?

Building a location based service application is a rather complex process, because it involves interconnecting several components, features and players. Location based services are always delivered to users through mobile devices or any device with ability to use their components, as listed below. These include mobile phones, but also tablets, personal navigation devices, desktop computers and others. Also, they always require the user to share their live location with the service provider.

Location-based services: main components



Most location based services require the following components:

Geolocation Technology

It's required to indicate the geographic position of a mobile device, both outdoor and indoor, so GPS, Cell-ID, Bluetooth, or others.

Map Provider

Service providers obtain mapping data from the relevant maintaining authorities or business and industry partners. For information on how to choose the right map provider, refer to this article.

Service Provider

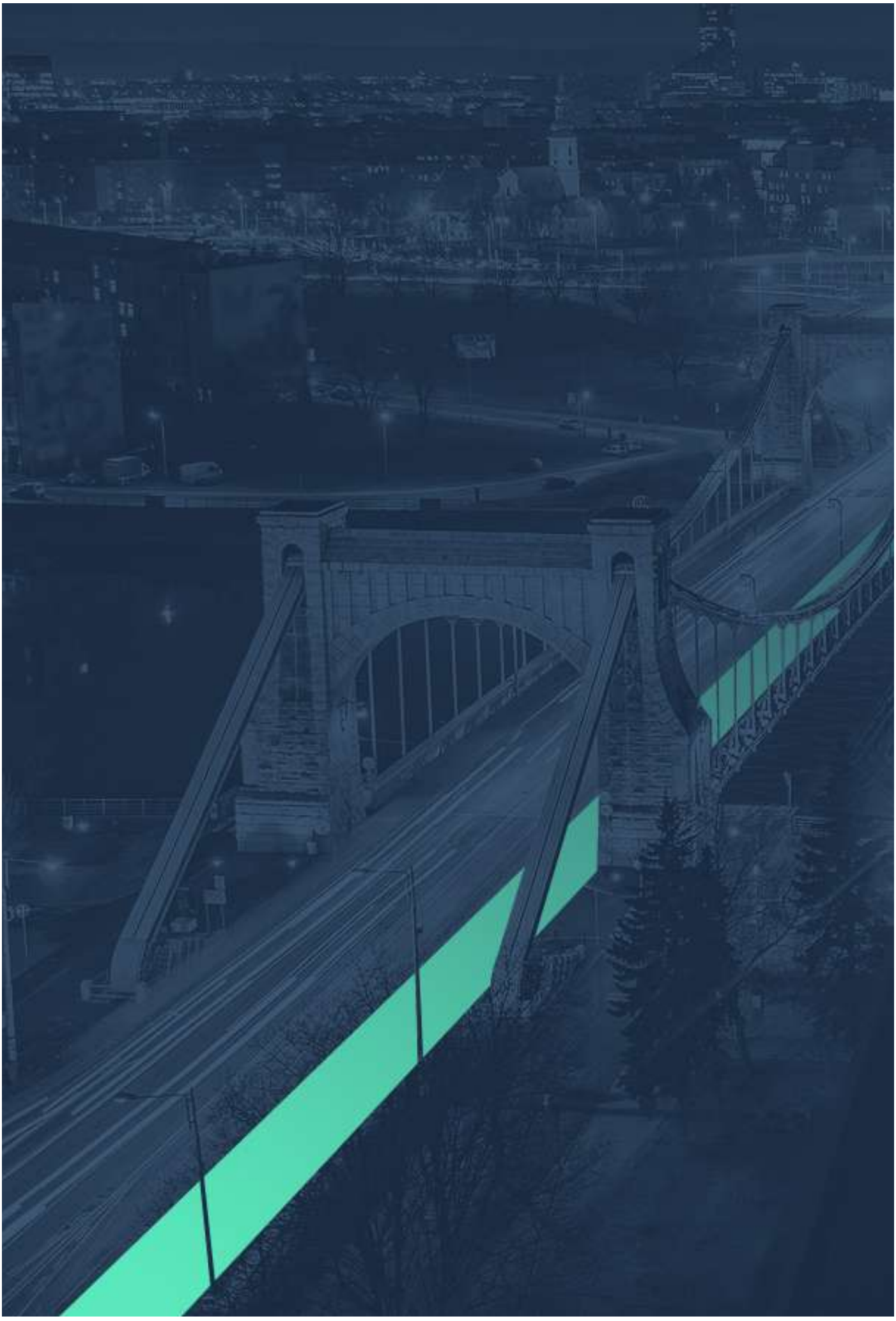
The service provider will administer the relevant software (e.g. GIS or the open-source version QGIS) and provide the necessary components to respond to user queries. The service provider models data for geolocation purposes or to indicate the best routes through relevant APIs.

Communication Network

The wireless network that facilitates the transfer of data between user (through mobile device) and server (service provider). In this day and age, we use wireless internet to achieve this, predominantly through 3G or 4G with 5G reaching 43% in market share in Feb 2022 and 6G undergoing heavy R&D.

Building location-based services in 2022

The choice of technologies and map providers isn't a simple task either, especially when scalability is one of our requirements. Wrong choices may lead to rising costs - both in terms of data usage and the potential bills incurred for remodeling the solutions. If you aren't sure what is the right choice for your application, contact us directly and we will be happy to advise and discuss your needs.



Chapter 2

Which map API should you choose and why?

Google Maps vs. Mapbox vs. OpenStreetMap

When embarking on a journey to build a location-based startup, one of the first things you should start thinking about is selecting the right map API provider that fits both your business needs, technical requirements and your budget.

To help with making the right decision, we've compiled a feature-based comparison of the most popular solutions on the market: Google Maps, Mapbox and OpenStreetMap.

If you're not familiar with some of them, no worries, you should have a good understanding by the time you finish this piece. And if you're completely new to location-based services, we'd recommend checking this article to learn more about the basics and relevant terminology.

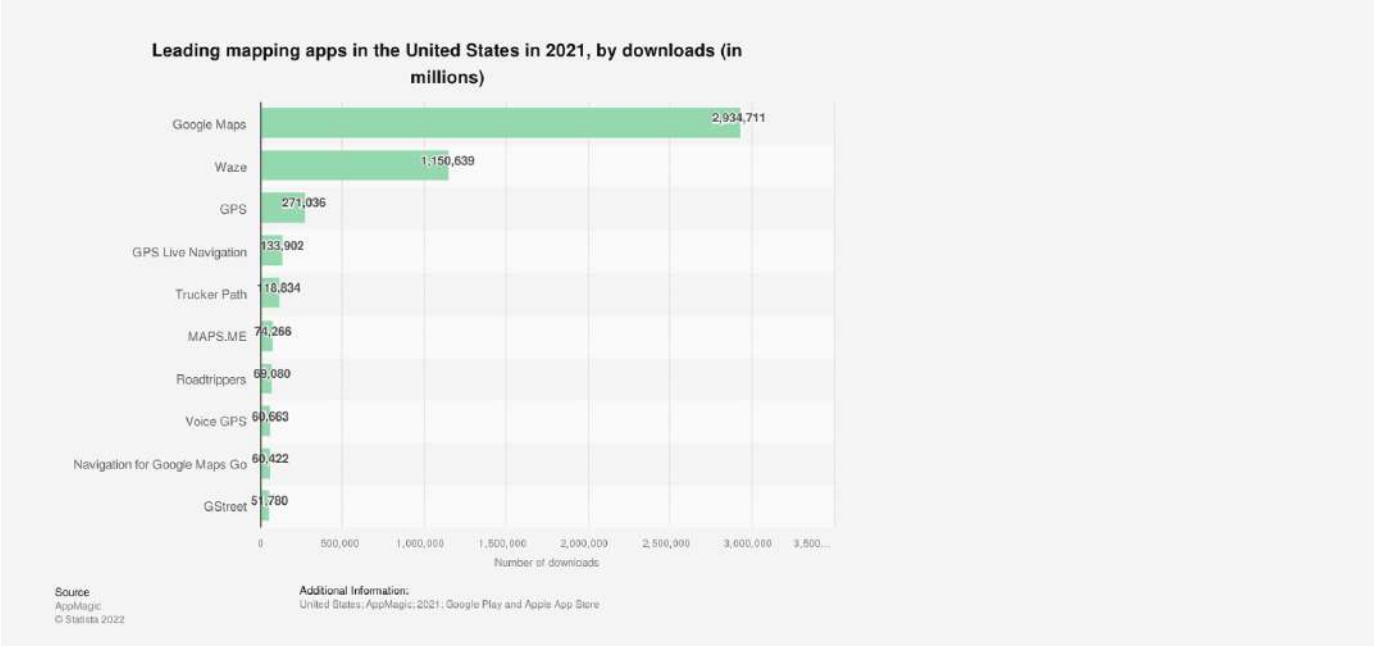
Now, let's dive straight in.

Which is the best map API?

Obvious question with a not-so-obvious answer. Each map API provider has its pros and cons, thus naming the best map API is dependent on your particular use case. To make the distinction between the selected candidates, let's look at the benefits and drawbacks of each.

Google Maps Platform review

Google Maps is most likely what comes to mind in a split-second after you start thinking about map services, and it's obviously not without a reason. If you look at the download statistics for maps application in the US in 2021, you'll notice a clear winner.



This may lead you to believe that going with Google Maps should be a no-brainer. But its consumer popularity is not that useful when it comes to business implementation.

The reality is, your users don't particularly care what map API provider you use, if it does the job they expect your application to do.

Just look at Facebook, for instance. It doesn't use Google Maps and yet its map functionality doesn't seem to suffer too much.

So in what cases should you look at Google Map API, and when should you divert your gaze?

GOOGLE MAPS API SERVICES

Maps API – offers a set of tools for creating, customizing and embedding maps on various platforms: mobile, web and cross-platform.

Street View API – offers access to real-world imagery and panoramic views.

Directions API – generates directions between various locations.

Distance Matrix API – measures travel distance and time for several points of destination.

Elevation API – generates elevation data for any point on the map.

Geocoding API – converts geographic coordinates into actual addresses.

Geolocation API – uses location data from Wi-Fi nodes and cellular towers to pinpoint the user's exact location.

Places API – an enormous list of up-to-date information about various locations.

Roads API – offers accurate GPS tracking with snap-to-road functionality.

Time Zones API – provides time zone data for any location.

PROS

- Because Google Maps are so widely used, they offer probably the largest database of both business and regular locations with constant updates and high accuracy.
- Google Maps support over 80 languages without you having to add them on your own, which can come handy if you're planning to operate in less popular areas.
- Relatively cheap for niche use cases with low traffic, currently (as of April 2022) offers 28 500 maploads per month for free.
- Numerous APIs can cover a vast variety of your business needs.

CONS

- Customization options are limited in comparison to its competitors, which might be a dealbreaker depending on your specific needs.
- Not open-source, which means you're bound to use what Google offers and adjusting it to your peculiar needs might require workarounds or won't be possible at all.
- Energy - and resource-hungry, which might be troublesome, especially if your market operates on low-spec devices.
- After Google's infamous spike in pricing back in 2018, many businesses are cautious of heavily relying on Google Maps services and potentially subjecting themselves to a sudden increase in overall usage costs.

Mapbox review

Mapbox is probably the largest B2B competitor of Google Maps and rightfully so. They managed to develop an enormous number of various map API services and tools, which I won't even dare to try and list in this article. I'll try my best to do an overview and for the complete information on Mapbox APIs – I'll direct you to their documentation.

MAPBOX API SERVICES

Maps APIs – offer a variety of tools for creation, customization and embedding of custom maps in web and mobile applications.

Navigation APIs – provide such functionalities as turn-by-turn navigation between multiple locations, route optimization, travel distance and time measurements.

Search APIs – perform worldwide forward and reverse geocoding.

Vision APIs – a set of tools for augmented reality navigation and automated driving applications that can interpret road scenes in real time.

Data APIs – allow adding a wide range of location datasets onto your maps, be it satellite imagery, aerial image, terrain data and more.

PROS

- Mapbox Studio is the first thing that comes to mind when I think of Mapbox. It's a tool that allows you to design the exact maps you need, from simply adding or tweaking a couple of elements, to creating full-blown 3D maps of real and/or fantasy worlds.
- Comprehensive documentation for developers provides all kinds of information for those who decide to use Mapbox in their project.
- Due to strict data management rules imposed by Mapbox, you have no other choice but to essentially adopt a standardized data handling approach, which will definitely be appreciated by your developers and your business in the long run.
- Offline mode and great performance provided by Mapbox's tileset architecture can be a great solution for many use cases, including those for working in areas with problematic internet connectivity or on low-spec devices.

CONS

- A steep learning curve requires higher investments in your development team in order to:
  - provide them with time needed to learn how to work with Mapbox,
  - conduct a more selective recruitment process if you're willing to hire experienced developers,
  - hire an external technology partner with extensive location-based services expertise.
- Mapbox is only partly open-sourced, which means you can encounter limitations during the development process that might require workarounds.
- Mapbox might be too complicated for smaller projects and definitely will be an overkill for a simple map embedding.
- Might not be the best solution for India- and China-based projects due to not-so-high data accuracy.

OpenStreetMap review

OpenStreetMap (OSM) was founded even before Google Maps was launched, hence it's definitely not some random map API provider that should be rejected without understanding what it can offer, although it might be much less known to a wider public.

OpenStreetMap is an open-source map provider that provides maps in their raw form, for free, with an open license and freedom to use said data in the way businesses or private users want.

OPENSTREETMAP API SERVICES

Unlike Google Maps and Mapbox, OSM doesn't provide numerous API services out-of-the-box, and requires integrating, often also open-source, 3rd-party solutions, i.e. traffic information, direction services etc. You can find a list of such components in OSM's wiki. Here's what you'll get from OpenStreetMap by default:

Editing API – provides fetching and saving of raw geodata from and to OpenStreetMap's database, which enables you to edit your map data according to your taste and requirements.

Overpass API – a read-only API that is optimized for dealing with user requests for various geodata with quick response rates.

PROS

- Completely free and open-source, which means you can tailor it specifically to your company's needs and add additional functionalities without having to deal with workarounds or simply waiting for the provider to further develop their product.
- A worldwide community constantly updates OSM's data and creates open-source solutions that enhance OpenStreetMap functionalities.
- When used for bigger projects, can save large sums of money that would otherwise rack up enormous costs with Google Maps or Mapbox. Just to give you a perspective, we've managed to avoid $250 000 monthly bill (approximated based on Google's official pricing and project's monthly number of requests) for one of our customers by not going with Google Maps and developing their own OpenStreetMap-based solution, which now costs them only $1 500 in monthly usage costs.

- Provides great performance and can be optimized even further, as shown by our experience in developing one of Europe's largest logistics platforms – Trans.eu that has over 62 000 concurrent online users on a daily-basis in 45 various countries.

CONS

- Poor documentation provides a steep learning curve, but once you master what OSM has to offer, you're capable of delivering exactly what your product needs.
- Due to fewer out-of-the-box features, requires initial capital investment in custom software development to build your own components and functionalities, or integrate with available, often also open-source, 3rd-party solutions.
- The majority of aggregated data is user-submitted, which may cause inaccuracies in less popular areas and necessitate manual correction.
- Might be an overkill for small-scale projects.

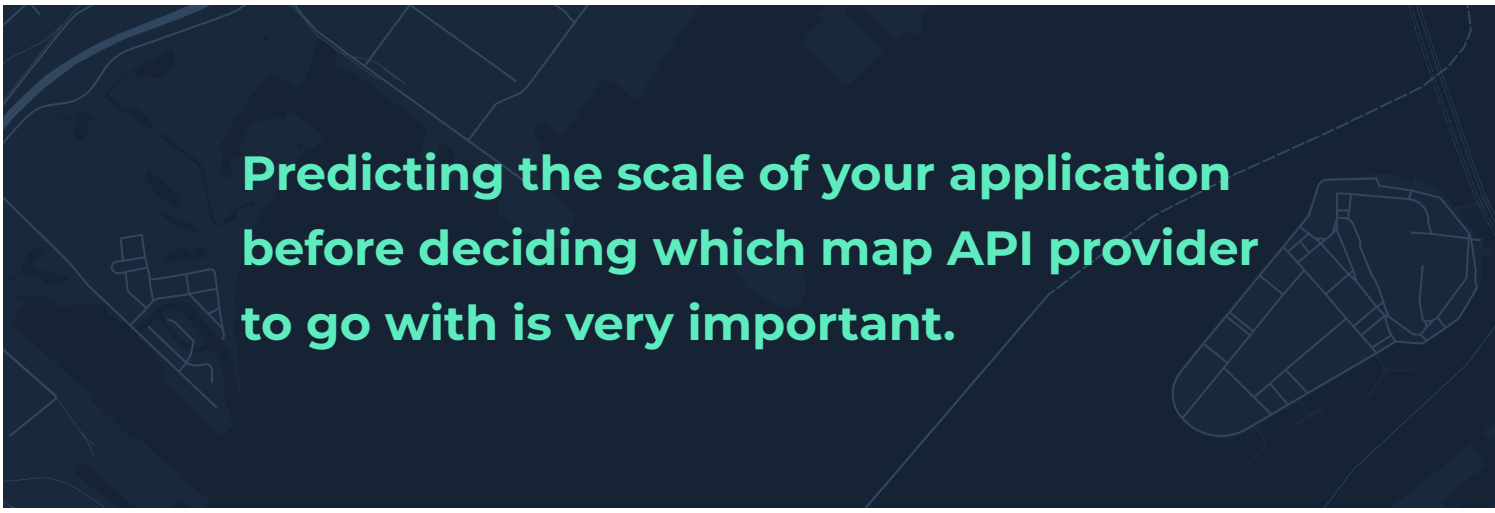
Map APIs comparison chart

	GOOGLE MAPS	MAPBOX	OPENSTREETMAP
Customization options	Limited	Very high	Unlimited
Number of out-of-the-box services	High	High	Low
Open-source code	No	Partly	Yes
Suits for small projects	Yes	No	No
Suits for medium-large projects	Yes, but very costly	Yes, but costly	Yes, but requires custom development in the beginning
Pricing model	Complicated pricing model that gets very costly with growing project's scale	Complicated pricing model that gets costly with growing project's scale	Free
Performance on low-spec devices	Medium	High	High

So, which map API should you choose?

As you can see from our comparison, each solution has its pros and cons as well as use cases. There's definitely no universal solution.

Looking at our experience with developing a number of location-based startups and large enterprise solutions, predicting the scale of your application before deciding which map API provider to go with is very important.



You do want to avoid a situation where you spend half a year developing your product based on one provider, to then double the scale of map usage and suddenly find yourself in a conundrum, trying to figure out how to decrease the unforeseen spike in monthly costs.

If you're not sure how to do this research yourself, you can drop us an email at offer@rst.software and we'll schedule a quick consultation call.

Chapter 3

GIS software development: how to cut the map usage costs with OpenStreetMap

With the location-based services market growing at a steady pace, location-aware applications are increasingly becoming a requirement across a number of industries.

Founders looking to launch their location based startups will find they have several choices to make on the journey, including the selection of a map API provider. There are several options here. We've already compared Google Maps, Mapbox and OpenStreetMap for their pros and cons in an earlier article.

In this article, we'd like to present some of the available offers solely from the pricing perspective. This is indeed an important factor, since pricing models are vastly different across the different providers and your usage bills will vary depending on the provider you choose. We'll take a look at what Google Maps and Mapbox can offer through their ready-to-use APIs and juxtapose it with an open-source alternative OpenStreetMap.

Let's start with the tech giant.

Google Maps APIs pricing

Google Maps, one of the most popular API providers, doesn't really need an introduction. It offers top data quality and it charges for it. The company dramatically increased its pricing back in 2018 - in some cases, by as much as 14 times! At the same time, it offers free tiers that may be a good option for those who look for some basic functionalities and expect rather limited traffic.

Google Maps charges for the usage per every API product. The pricing options are divided into three tiers based on the volume of requests: 0–100,000; 100,001–500,000 and 500,001+. I listed the rates for some of the most common APIs in a table further below. Google's pricing ranges are among the most expensive on the market.

Although the tech giant certainly provides unmatched quality on the geodata market (they rely mainly on cars with mapping technologies and aggregated user data), customization options for the APIs offered are rather limited - you will always have to use the default base layer of Google Maps. Mapbox may be a more attractive alternative in this department.

Mapbox APIs pricing

Mapbox was created in 2013. This solution rose in popularity specifically after Google increased its pricing for map APIs usage. Apps that are currently powered by Mapbox include Facebook, Pinterest, Shopify, Grubhub, Airbnb and many others.

Similarly to Google Maps, with Mapbox you only pay for the API usage. The provider offers solid free tiers for a limited number of requests per month as well. It offers different pricing options for individual users at four different tiers: 0-1,000, 100,1-50,000, 50,001-100,000, 100,001-200,000 and 200k+. The prices are somewhat less strenuous to the budget than Google's offering – do take a look at the table included below for common APIs (or Mapbox website) for more details.

Mapbox sources map data from OpenStreetMap, Microsoft Open Maps, Wikidata and other data vendors. The OpenStreetMap layers and features can be customized and added directly in Mapbox Studio. In fact, customization is the greatest strength of Mapbox; developers are free to customize its base layer.

OpenStreetMap pricing

OpenStreetMap was founded in the UK in 2004 when map data sources were controlled by governments and private companies, and thus were expensive. OpenStreetMap, an open source project run mainly by volunteers, was supposed to solve this problem. Unlike Google Maps and Mapbox, which provide map APIs, OpenStreetMap is solely a data pool and is available for free. Google Maps and Mapbox don't share the map data unless you pay for it. If you're able to self-host, you may want to look at the free Mapbox-like project that offers ready-to-use vector map packages.

With OpenStreetMap, location-based startups don't have to bear the map usage costs, no matter the volume of requests. However, they will have to custom develop whatever features they want to include in their tool. That will leave them with the infrastructure, development and maintenance bills. Altogether, the costs may be significant.

Is OpenStreetMap a good choice for your project? All in all, opting for a custom-developed solution based on OpenStreetMap will help you cut the costs under the right circumstances.

What are they? To arrive at the answer, you should first consider these crucial questions:

01

What are your use cases?

If you're building an application for public transport, navigation, weather visualization, geodata processing, map-based business data or any other highly-advanced solution that will involve the processing of large amounts of data, then a custom-developed solution with OpenStreetMap could allow you to significantly cut the usage costs. Building such solutions with paid map APIs is bound to incur hefty bills.

02

What traffic are you anticipating?

Both Google Maps and Mapbox offer free tiers for low map usage. If you only need to embed a simple map and expect low traffic, this is a perfect option for your business. Startups could certainly benefit from these free tiers, however, they should also consider how they plan to scale in the long run.

03

What are your custom project design needs?

What custom features do you want to incorporate in your location based application? Perhaps you want to add a custom graphic layout? You won't be able to create custom designs with Google Maps. Mapbox, on the other hand, offers a lot of different customization options. If you are pressed to include specific features or characteristics that API providers don't offer, you will have to opt for custom map development. In such cases, OpenStreetMap will be your best option.

04

What specific functionalities are you looking for?

How complex is the solution you are building? You need to list the specific features you want in your tool and check them against the features offered by API providers. Do you want to include route tracking? Will you need dynamic or non-dynamic data? Do you want to display e.g. the types of vehicles? Complex functionalities will either require customizing what is available or building it from scratch.

05

How much time do you have at hand?

This is an important question. If your project is time sensitive, you will have to consider whether you actually have time to custom develop the mapping functionalities from scratch. If not, opting for ready-to-use APIs that you can adjust to your needs may be a quicker and better option. Preparation, development and maintenance will take much longer than implementing ready-made APIs from Google Maps or Mapbox. However, you will still have to consider the map usage costs and the benefits of switching to a custom solution a few years down the road.

06

What data quality do you require?

In comparison to map providers like Google Maps, open source-based solutions like OpenStreetMap and Mapbox will somewhat compromise on the quality of data in use. Remote areas may not be properly covered by open-source geodata, for example. Is this good enough for you? The lack of systematic quality checks of data in OpenStreetMap may be a disadvantage.

07

What region are you planning to cover?

If you're focusing on a specific region, you may want to check on the data quality you intend to get for that specific area. The detailedness of map data will vary across different geographic regions. Geospatial data is prone to errors related to e.g. positional, attribute and temporal accuracy, as well as logical consistency and completeness of data. Some parts of the world aren't commonly navigated or easily accessible, and that could affect that data quality. OpenStreetMap offers lower coverage than Google Maps and somewhat lower than Mapbox (as it also uses OSM but enriches its data from other sources).

08

What's your custom development budget?

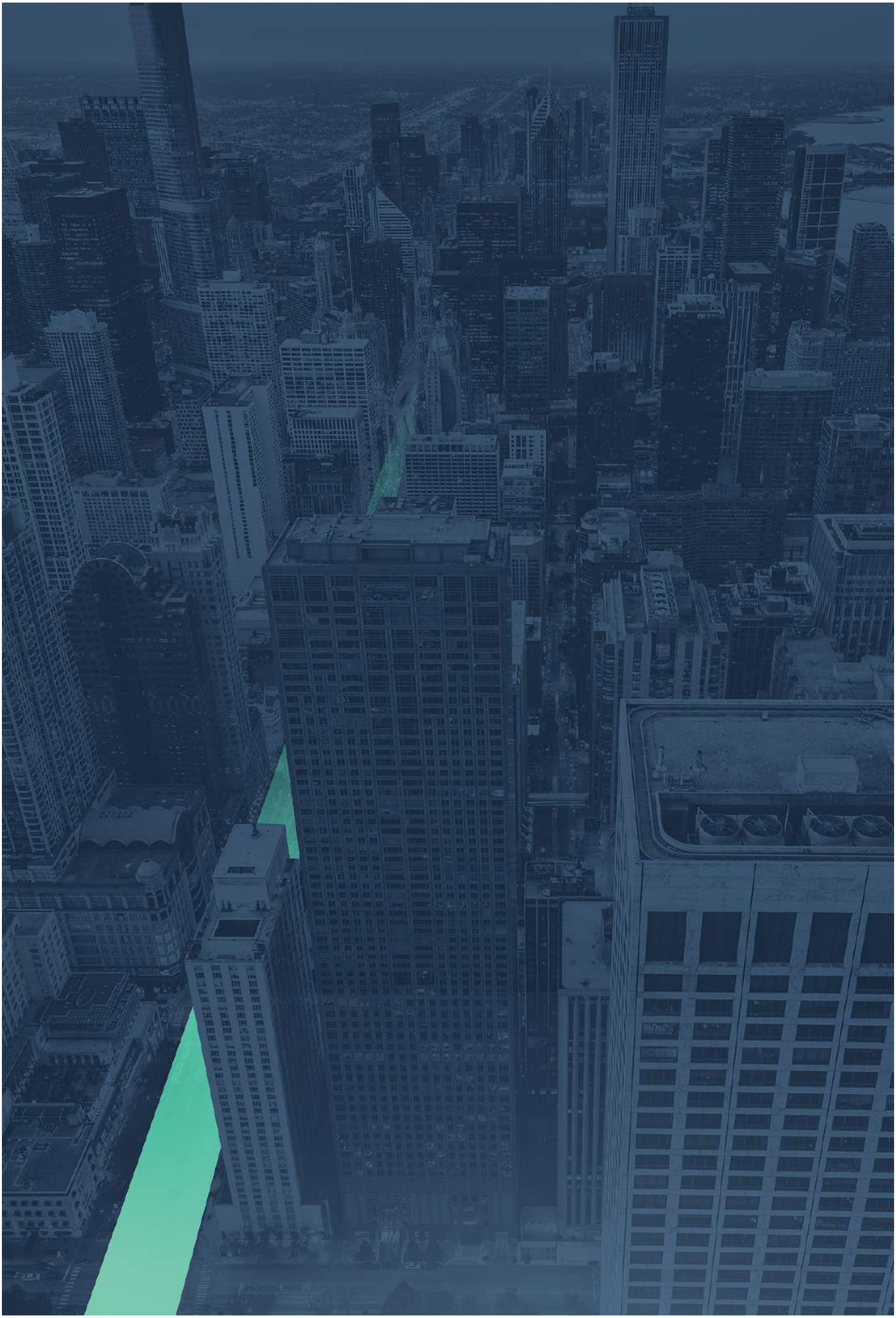
Your application traffic will be generating the map usage costs, so it's important to know the expected traffic level. If we're talking about a relatively low usage, you may be better off with what paid API providers can offer. Consider specifically the free tiers that both Google Maps and Mapbox offer; I have listed in the table right above. With a custom-developed solution, you will only have to cover the development costs once, but these can be significant if you're building a complex application.

Cutting map usage costs with OpenStreetMap

When you're considering the different map providers for your location-based application, opting for an open source solution may be a good alternative to API providers like Google Maps or Mapbox, if you're oriented at cutting the costs stemming from map usage. We built Trans.eu, our logistics platform that handles 30 million requests on a monthly basis, with OpenStreetMap.

To determine whether OpenStreetMap is the right option for your location-based tool, you should first consider the traffic levels you are expecting and planning in the long run, the use cases as well as the features and functionalities to be offered by your solution.

Smaller applications might be better off using Google Maps or Mapbox APIs. Depending on your needs, you may be better off benefiting from promotions, beneficial pricing models or free tiers. Complex solutions with advanced functionalities that require custom development can definitely benefit from OpenStreetMap, as long as you are fine with a somewhat lower data quality.



Chapter 4

Top 12 AWS services for building an Uber-like location-based startup

On-demand economy – also known as 'gig economy' or 'sharing economy' – is constantly growing and expected to generate half the sales revenue ($335bn) in the sectors with yet-prevalent traditional operating models by 2025, according to the PwC study.

With the likes of Uber, Lyft, Airbnb, and Netflix expanding further and further, more startups try to challenge the industry leaders or adapt their business models to yet-unconquered markets.

Succeeding in these endeavors requires thorough software development planning and preparation for rapid scalability, should you decide to actually have a go at battling the titans with their access to ‚unlimited' capital.

If the above is your goal – read on, as today we'll focus on developing a mobility startup, specifically one of its crucial elements – the infrastructure. For the simplicity of explanation, let's take building an Uber-like app as our example.

Cloud vs on-premise

To begin, let's lay out the foundation. Should you go for a cloud or an on-premise hosting? Both solutions, obviously, have their own pros and cons, but as we are a startup company, going for the cloud-based option will help us save the cost of setting up and running a feasible on-premise solution.

As RST is an official AWS consulting partner, I have access to our certified cloud architects and had a chance to speak with our Chief Cloud Solutions Officer – Marek Ziółkowski, whose knowledge and expertise will help me guide you through the process of understanding what is required from the infrastructure perspective to face rapid scalability and growth of your business.

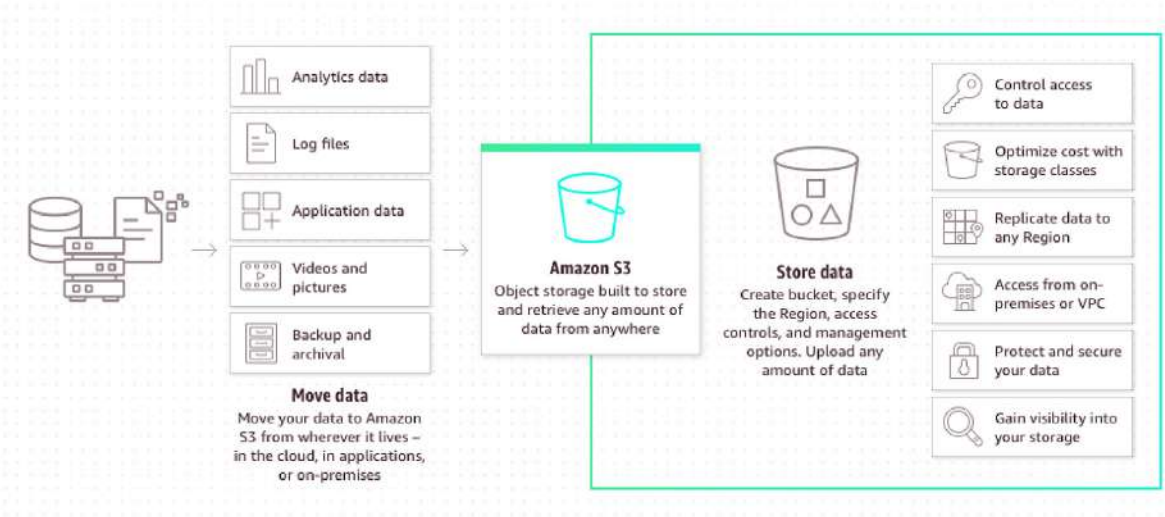
Frontend for an Uber-like app

Let's start with the frontend of your application. In most cases, it is represented by a static web or mobile screen, meaning you don't need to re-render what is being shown on the screen each time someone uses your application. All the interactive parts, eg navigating a map, will be done by embedding an external service, be it your custom-built one or via third-party APIs. This leaves us a fairly simple task to fulfil.

Amazon S3

Amazon Simple Storage Service (Amazon S3), simply put, is your cloud storage that offers industry-leading scalability, data availability, security, and performance features and should be used for hosting your static applications' frontend.

Not much to add here, to be honest with you. It's as simple as it gets. Obviously, we could dive deeper into the process of properly setting up an S3-based storage, but that would make this article too technical for the purpose of giving you an overview of what is needed for Uber-like app development, so let's leave it for later.



Compute platform for an Uber-like app

The compute part of your application is where it gets slightly complicated. First, we need someplace to host our infrastructure. There are two options.

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) is the simplest and yet broadest compute platform offered by Amazon – in other words – your virtual server. Amazon gives you full access to your EC2 instances, just as if you would have with a regular on-premise server.

Here's what it means:

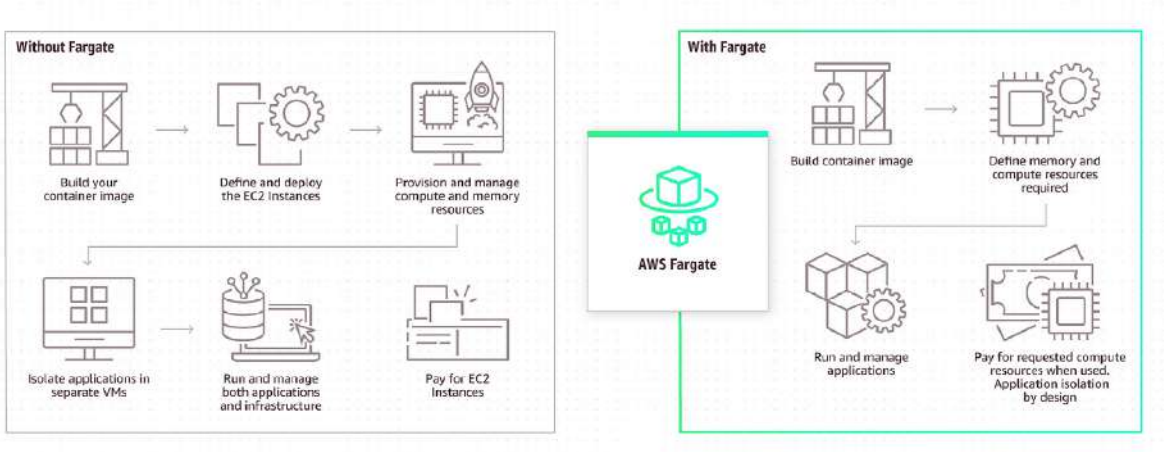
- You don't want to build a cloud-native application (which we do recommend building, by the way);
- You want to have access to the server's OS;
- You want to access the console terminal;
- You either don't plan to scale rapidly or you have access to DevOps engineers who will be able to configure everything properly;
- You plan to have a stable and predictable resource usage and want to reserve instances for a longer time-frame, which will lower your costs; or
- You don't want to vendor lock-in yourself with AWS and want to be able to migrate when required.

AWS Fargate

AWS Fargate is Amazon's serverless, pay-as-you-go compute engine that allows you to ignore an entire process of dealing with your own servers, whether virtual or not, and focus on developing your app instead.

To be completely honest with you, Fargate is the solution we usually advise to go with, unless you have a very specific case, which an Uber-like app isn't. Why?

- Fargate removes the operational overhead of scaling, patching, securing, and managing servers;
- It automatically scales your compute infrastructure according to the actual usage;
- If your user traffic suddenly peaks due to an ongoing marketing campaign, you don't need to prepare beforehand; and
- If your app usage drops at night or during working hours, you don't have to pay for the idle servers.



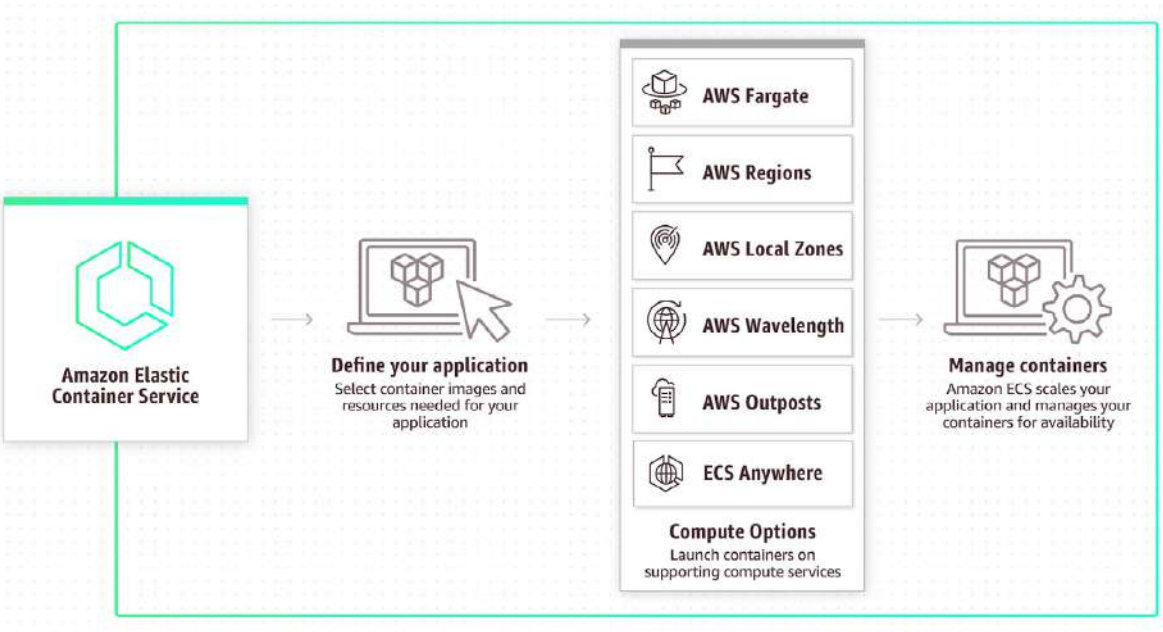
Now, let's think about containerization of our application. AWS offers two services, and it's up to you and your dev team to decide for which to go, as both solutions are equally good for the given task. Let's look at both ECS and EKS (both are compatible with AWS Fargate and EC2).

Amazon ECS

Amazon Elastic Container Service (Amazon ECS) is Amazon's own fully-managed container orchestration service that takes care of deploying, managing and scaling of your containerized applications.

It makes sense to go for this service if:

- You're going to use other AWS services for hosting your application, like the aforementioned S3, and you don't particularly care about the vendor lock-in with AWS; and
- You don't want to worry about having to set up everything on your own and would rather utilize proven solutions.

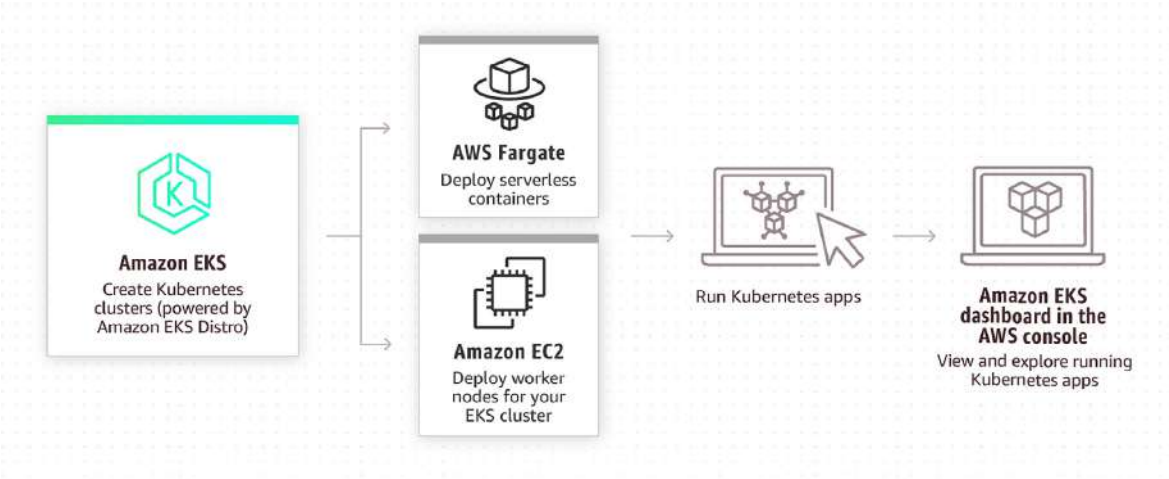


Amazon EKS

Amazon Elastic Kubernetes Service (EKS) is a managed container service that allows you to run and scale Kubernetes-based applications with your AWS infrastructure.

It's worth going for EKS if:

- You don't want to vendor lock-in with AWS and want to be able to run your app or its elements with Kubernetes on-premise or within other cloud providers (this approach is called 'multi-cloud', if you'd like to search more about it); or
- Your team already has Kubernetes competencies and you don't want to switch technologies.

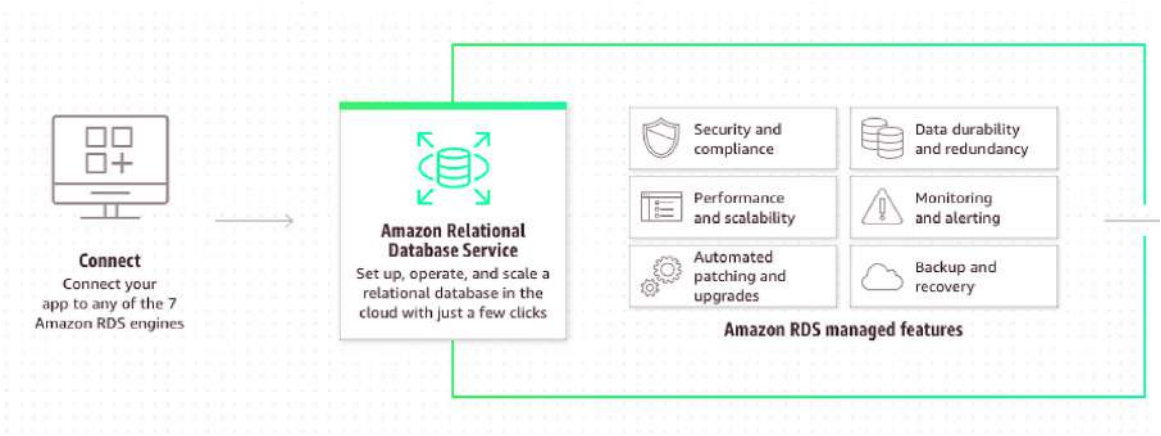


Data storage services for an Uber-like app

Okay, we've sorted out the computing piece. Now we need to figure out how to store our data, which in the case of an Uber-like application requires specific solutions compatible with location-based services.

Amazon RDS

Amazon Relational Database Service (RDS) is a collection of managed database services which we'll use to host our database. It's compatible with the seven most popular engines, and most importantly, with the one we're going to need in our case.



Amazon RDS for PostgreSQL + PostGIS

Because our location-based application needs to operate on geodata, PostGIS is an extension for PostgreSQL that enables storing and managing of various spatial information, eg polygons, geographic lines and coordinates, geometric figures etc. PostgreSQL also supports libprotobuf 1.3.0 – a library used by PostGIS to deal with map box vector tile data.

It's also worth taking into consideration it's best to create a separate database with installed PostGIS extension that will handle your map-related data alongside your regular PostgreSQL database.

Amazon S3 Glacier

Amazon S3 Glacier is a storage class designed specifically for data archiving, be it your logs, backups, or some actual data archives. The main principle is that you shouldn't need to have instant access to those files. Which, in turn, will allow you to significantly reduce your storage costs for long-term digital preservation and rarely-accessed data.

Amazon CloudFront

Amazon CloudFront is a content delivery network (CDN) that is built to reduce latency and increase the performance of your application in various regions around the world. In simple terms, it caches contents of your app and when a user in Australia runs it, instead of requesting the content within your server region. Let's say you're based in the US, the request goes to Amazon's Australian CDN and receives what the user needs in a fraction of the Australia-US latency.

This is extremely useful if you're planning to operate in multiple locations around the globe, as it not only provides better experience to the end-user but also reduces your costs.

Location-based services for an Uber-like app

Not the last and definitely not the least, building an Uber-like startup heavily relies on proper usage of location-based services. There are plenty of various third-party map providers, and building everything from scratch would be a suicide mission from any logical standpoint, so you'll need to select an option that works best with your planned location based features.

In our previous articles, we've compared the most commonly used ones, ie Google Maps, Mapbox and OpenStreetMap. If you haven't had a chance to check it, please follow the links to the map APIs comparison as well as their pricing models.

But, as we're looking at the services provided by Amazon, it's worth noting that they also have their own tools for building Uber-like apps.

Amazon Location Service

Amazon Location Service is a set of solutions that natively integrate with the rest of your AWS-based services and provide plenty of ways to generate and process location data out-of-the-box.

THAT MEANS YOU CAN USE ALS IF YOU WANT TO:

- Track your assets (which would be your fleet in our Uber-like app);
- Coordinate delivery tracking (which can also be understood in a sense of moving passengers from one location to another with push notifications once you enter the geo-fenced region of passenger's proximity etc.);
- Implement routes optimization for point-to-point navigation (perfect for offering an on-demand taxi service); or
- Use it for user engagement and geomarketing (special discount for app users at an airport).

Geospatial data is provided to Amazon from well-established global providers such as Esri and HERE, so you don't have to worry about getting good geographic coverage and geodata accuracy.

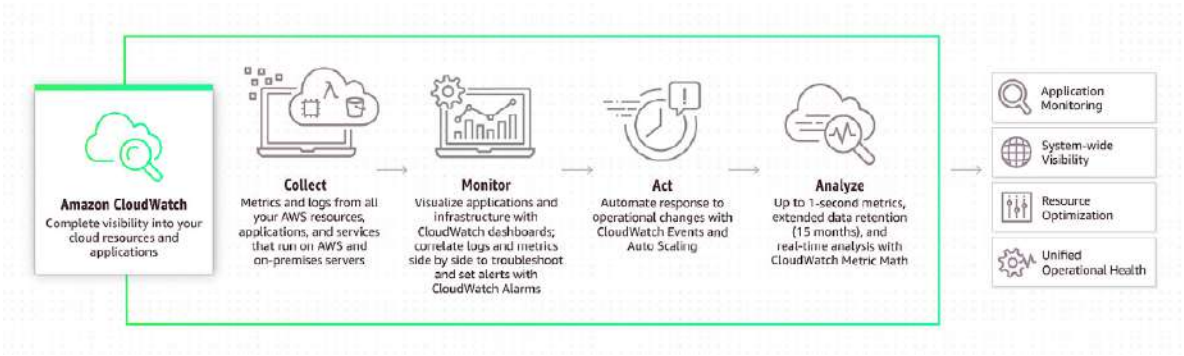
If your Uber-like application requires custom data layers or map features altogether, you can easily integrate third-party solutions or build your own location-based services using OpenStreetMap, for instance. We used the latter for one of our largest clients – Trans.eu (here's our case study) – the leading logistics platform in Europe and Asia.

Analytics tools for an Uber-like app

Your app is up and running. You want to keep tabs on both app's performance and business growth. Here's what you can get from Amazon's offer.

Amazon CloudWatch

Amazon CloudWatch is a monitoring and observability service that provides your dev team with insights into your Uber-like app's performance, whether we speak of system-wide metrics or just your infrastructure resources utilization. While you, as a business founder, might not be able to fully understand the scope of provided analytics data, it's important to have it set up from the beginning, so it's both easier to optimize your system and quickly pinpoint any existing issues.



Amazon QuickSight

Amazon QuickSight is what you might find a lot more interesting and aligned with your field of responsibility. QuickSight is one of the most popular cloud-native, serverless business intelligence (BI) tools.

As it was in the case of Fargate, QuickSight is also an auto-scaling, pay-as-you-go service that allows you to connect all

your data sources within one place for complex analysis. A customizable dashboard provides you only with the data that matters to you and integrated machine learning algorithms can detect anomalies, forecast business metrics as well as perform interactive what-if analyses.

If you store all of your data inside the suggested above RDS and S3 storages, QuickSight also provides you with native integrations and granular access control.

Sum-up of the best AWS tools for building an on-demand delivery app

As you can see, Amazon Web Services offer a plethora of tailored solutions that can easily power your Uber-like startup, from its initial software development stages to further improvements, analysis, rapid scalability and world-wide availability.

In short:

01

Amazon S3

Best for hosting your static frontend: websites, web and mobile applications.

02

Amazon EC2

The best cloud server for your backend, especially if you want to avoid vendor lock-in, but requires an experienced development team for proper setup and maintenance.

03

AWS Fargate

The best serverless solution for your cloud-native application that offers pay-as-you-go pricing model, requires no deep DevOps expertise and is best for rapid scalability due to its auto-scaling infrastructure.

04

Amazon ECS

Amazon's own containerization service, best if you're going to use more AWS-powered services in your Uber-like app.

05

Amazon EKS

Amazon's solution for Kubernetes-based containers, best if you're trying to avoid vendor lock-in or have extensive prior experience with Kubernetes.

06

Amazon RDS

Your go-to choice of database storage.

07

PostgreSQL + PostGIS

The best choice for storing your geospatial data with increased performance and computation abilities.

08

Amazon S3 Glacier

The cheapest storage option for secure data archiving.

09

Amazon CloudFront

The best CDN service for low-latency access to your app from anywhere in the world.

10

Amazon Location Service

A set of tools designed specifically for building location-based startups that offer cost-effective location-based services (LBS) out-of-the-box.

11

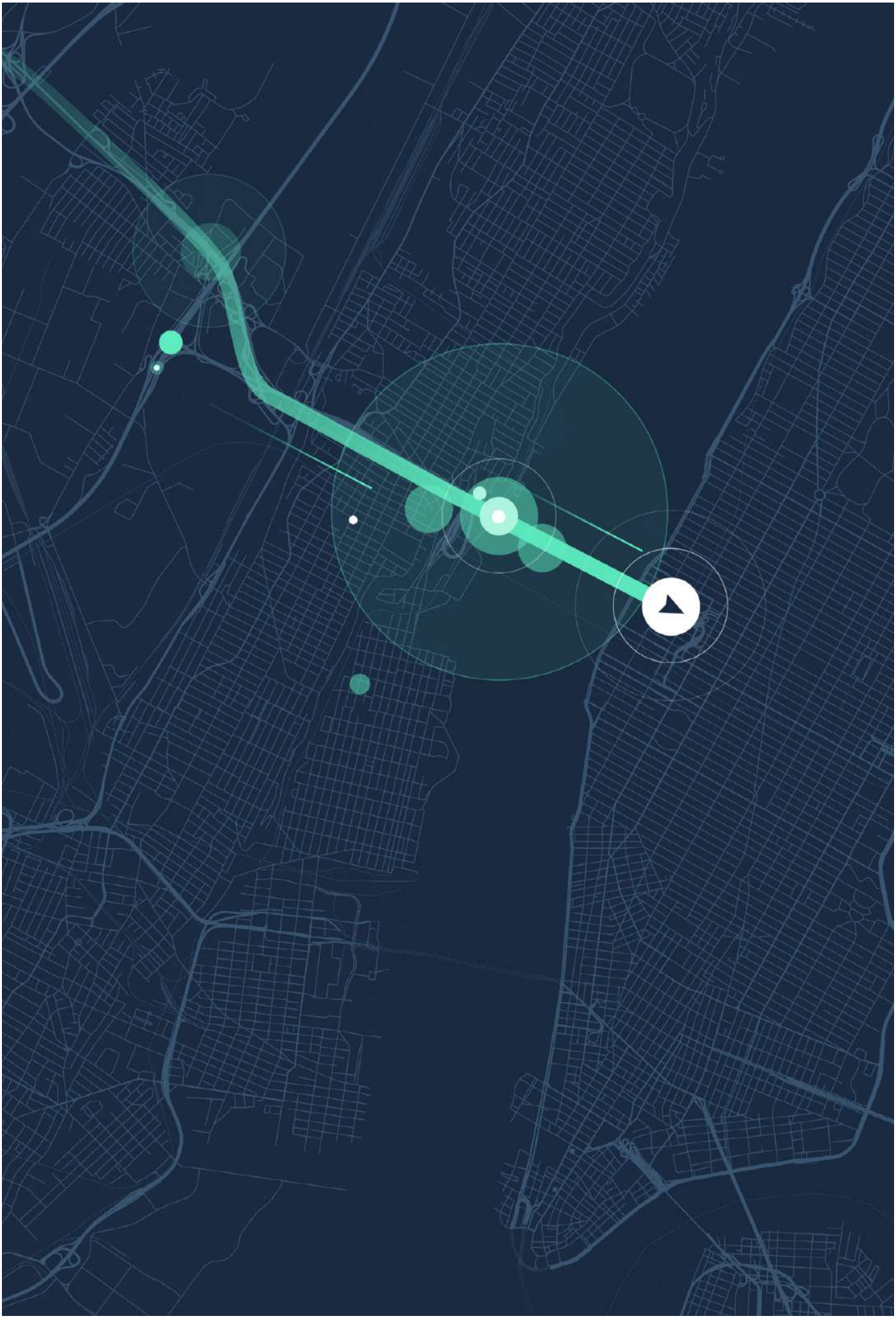
Amazon CloudWatch

The monitoring tool of your choice that provides your dev teams with crucial information about your system's performance and existing issues, should such appear; and

12

Amazon QuickSight

The best cloud-native BI solution that can enhance your decision-making processes with real-time business data and machine learning-powered analysis.



Chapter 5

6 common geolocation app development challenges and how to overcome them

The use cases of location-based applications seem to be endless. In times of on-demand economy and industry 4.0, the likes of Uber, on-demand service apps, health monitoring, fleet management solutions and even dating apps are growing in number as businesses strive to make our lives easier in increasingly innovative ways.

The location based services market continues to grow at a steady pace - it's expected to amount to 48 billion USD by 2026, growing at a Compound Annual Growth Rate of 19.4% between 2021-2026.

The downside is that geolocation app development isn't straightforward. Mobile apps that include geolocation services require features that, among other things, will allow tracking user location in real-time. In this article, I'd like to explain the issues to be considered as you embark upon building a geolocation app. Let's take a look at the most common challenges and how to overcome them.

Challenge
01

Choosing the right geolocation tech stack

For the geolocation app to function efficiently and accurately, you must ensure the right tech stack to support it. Typically, geolocation tracking apps will use at least two APIs: one to detect the user location and another to place them on a map. If you'd like to get a better understanding of the components of location based services, I'd like to direct you to an earlier article where you'll find all the information you need. Now, let's take a look at some of the geolocation tech stack issues you have to consider as you create a location based application.

Outdoor vs. indoor geolocation technologies

In order to choose the right tech stack, you first need to define your business objectives, the use cases of the geolocation app and the target audience.

With that, you can decide whether you require indoor or outdoor geolocation technology and how accurate it needs to be. When it comes to outdoor location tracking technologies, you have the following options:

GPS, which determines your current location based on signals received from satellites, with accuracy errors within the range of 0,3-5 meters, so it's pretty accurate.

A-GPS, which uses Cell ID to further improve the phone's GPS data quality, which means that it will work even in adverse weather conditions or when there is interference with the GPS signals.

Cell ID, which uses a nearby cellular network to determine physical location. The accuracy depends on the strength of the cell tower's signal. It can work well in the cities, but will be weak in rural areas.

Wi-Fi, with accuracy error of up to 25 meters.

Note that an integration of all four elements will allow you to increase precision of the signal. We advise app developers to use a combination of as many technologies as possible to maximize the accuracy of their location based data.

Indoor solutions, including beacons and geofencing, are particularly useful in personalized location-based marketing and asset management. Wi-Fi can also be used for indoor geolocation purposes.

GIS platform

Geolocation app development also requires choosing the relevant GIS, or geographic information system, provider (the base of your location based mobile app). Google Maps is among the most popular providers, but there are many other solutions that can be useful in geolocation services development, including Bing Maps, Mapbox, Tomtom and more.

These solutions offer ready-to-use APIs for maps, routes, places and other components required for building location-based apps. Choosing the relevant GIS provider for your geolocation functionality will largely depend on your business goals and processes. You may not require an advanced system for your use case, in which case you could benefit from ready-made APIs. However, geolocation mobile apps built this way may incur substantial bills (more information on this in the next point).

Challenge
02

Forgetting to consider pricing implications

As I've just mentioned, location-based app development can benefit from ready-to-use map APIs provided by Google Maps, Mapbox, TomTom or other providers. It's important to carefully consider the different pricing options offered by these providers, as the usage of different geolocation features will generate costs that vary across these API providers. For those who intend to use APIs in their geolocation app development projects and want to find out more about the pricing of Google Maps and Mapbox specifically, I've written a comparison in this article.

Developers can also consider open-source solutions for their location-based app development projects, but this is bound to incur considerable development costs. At the same time, when you use OpenStreetMap, for instance, you don't pay for the usage of location data. This is a great option for those who want to cut the location based app usage costs, as long as they are prepared to work with lower quality map data. If you'd like to review the available open-source options for your location-based technology, take a look at this article.

Alternatively, owners can cut the costs of their location-based service by opting for cloud-based geolocation apps. In our view, AWS's PostGIS + PostgreSQL is the best option for storing geospatial data in the cloud. See this article where we outline top 12 services useful in building location-based applications in the cloud.

Challenge
03

Obtaining permission to access user's location data

Geolocation apps require user permission to track their location. Today's users are well-informed and picky, so getting their permission to share their location with you is a task in itself. For instance, geolocation apps that track employees' location are often met with hesitation when it comes to location tracking. Employees want to know if they will be tracked outside the working hours, for what purposes and how this data will be stored - and that is reasonable. In general, users do not want to have their location data tracked at all times.

Unless you're building an Uber-like app, you will have to explicitly communicate why the solution requires geolocation to operate. Otherwise, users may be hesitant about sharing their information. And, if you fail to get their permission, your location-based app will not function properly. Therefore, it's crucial to think about messaging that will convince the users to opt-in. You should also think about relevant and convincing error information in cases where location tracking wasn't enabled.

A well-designed geolocation app won't have to defend its purpose. Make sure you state clearly why the permission to track users' location is required and how the data will be stored. Ideally, users should be informed about it as they create their accounts with your geolocation app or solution.

Challenge
04

Optimizing battery consumption

Businesses should always aim for the minimal battery usage - ideally, the energy required to power the screen. If the location tracking feature is permanently on or the app polls GPS data every 60 seconds (or even more frequently), it's going to drain the battery of the mobile device quickly and you will risk losing those hard-earned users.

To overcome this issue, think about the relevant action thresholds that will help you avoid unnecessary battery strain on the mobile device (we've compared the most popular map APIs and their battery consumption in one of our previous articles, amongst other things). You can, for instance, lower the frequency of polling if a user location hasn't changed for several minutes. For rapidly moving objects, you can analyze the speed to determine the required frequency of polling. Finally, benefit from automated tests to track energy consumption of specific devices.

Challenge
05

Ignoring user data privacy concerns

According to different regulations across the globe, e.g. EU's General Data Protection Regulation (GDPR), location data qualifies as personal information when it relates to an identifiable individual. Organizations delving into geolocation app development must take these legal measures into account and implement

relevant compliance measures. Check the relevant local regulations for the region you will cover with the location based app and take into account the regional measures, such as GDPR or the U.S. Electronic Communication Privacy Act or the California Consumer Privacy Act.

Geolocation app owners must implement relevant compliance mechanisms to protect the data they collect. Apply multi-factor authentication to minimize the risk of hacker attacks. Ensure all communications are encrypted. You can use SSL certificates, Network Security Configuration for Android and App Transport Security for iOS. By all means, do not store sensitive data directly on the device - if you have to keep it, encrypt it with relevant algorithms, such as AES-256 or RSA.

Challenge
06

Customer support for location aware apps

When users navigate the streets with your location based app, they rely on real-time, accurate data. Geolocation apps aren't free of risks and functionality issues may occur just like with any other mobile application. In addition to bugs or issues you could encounter in apps without geolocation features, location based apps are prone to issues with location accuracy, indoor navigation, inability to find a specific location, etc.

Whether these occur due to poor internet connection, issues with location features or bugs in custom development, you should be oriented at addressing them quickly and ensuring there's a customer support team available 24/7 to respond to user queries. Also, think about how you are going to maintain the geolocation app. Location data may be changing quickly and unexpectedly, so location based app owners should take that fact into account. A trusted custom software development company may help you in these endeavors.

Avoiding mistakes in geolocation app development

Geolocation app development process is somewhat more complex than building mobile apps without the geolocation feature. Choosing the relevant tech stack that enables location tracking is just one of them. You'll have to make several other decisions throughout the development process.

Even the most straightforward geolocation app will have its technical nuances that will largely depend on the solution's use cases. Owners must also think about data privacy and the fact that user's are generally reluctant to share their location with application providers. Finally, businesses should also remember that their geolocation apps must remain in compliance with legislation on data protection.

Chapter 6

GIS software development cost optimization:

how we saved over $420 000 on monthly map APIs usage

There are more and more use cases for location-based applications. Allied Market Research predicts this segment should reach $318.64 billion by 2030, with the projected compound annual growth rate (CAGR) of 24.3% between 2020-2030. This data shows that location-based service apps are increasingly becoming a necessity for many businesses.

Relying on ready-made APIs offered by popular map providers such as Google Maps or Mapbox can prove to be very costly for companies that build robust location-based solutions or those that scale up quickly, if not faster than expected. We learned this while working with Trans.eu – one of the leading logistics platforms in Europe and Asia.

In this article, I will explain what led us to choosing the open-source OpenStreetMap to build Trans.eu. Let's start with a little bit of background on the platform itself.

About Trans.eu

Trans.eu is web-based logistics marketplace, concurrently used by over 62 000 companies with nearly 6 mln location-based APIs requests on a daily basis. We started building it back in 2005, so it has grown and evolved to a great extent over the years.

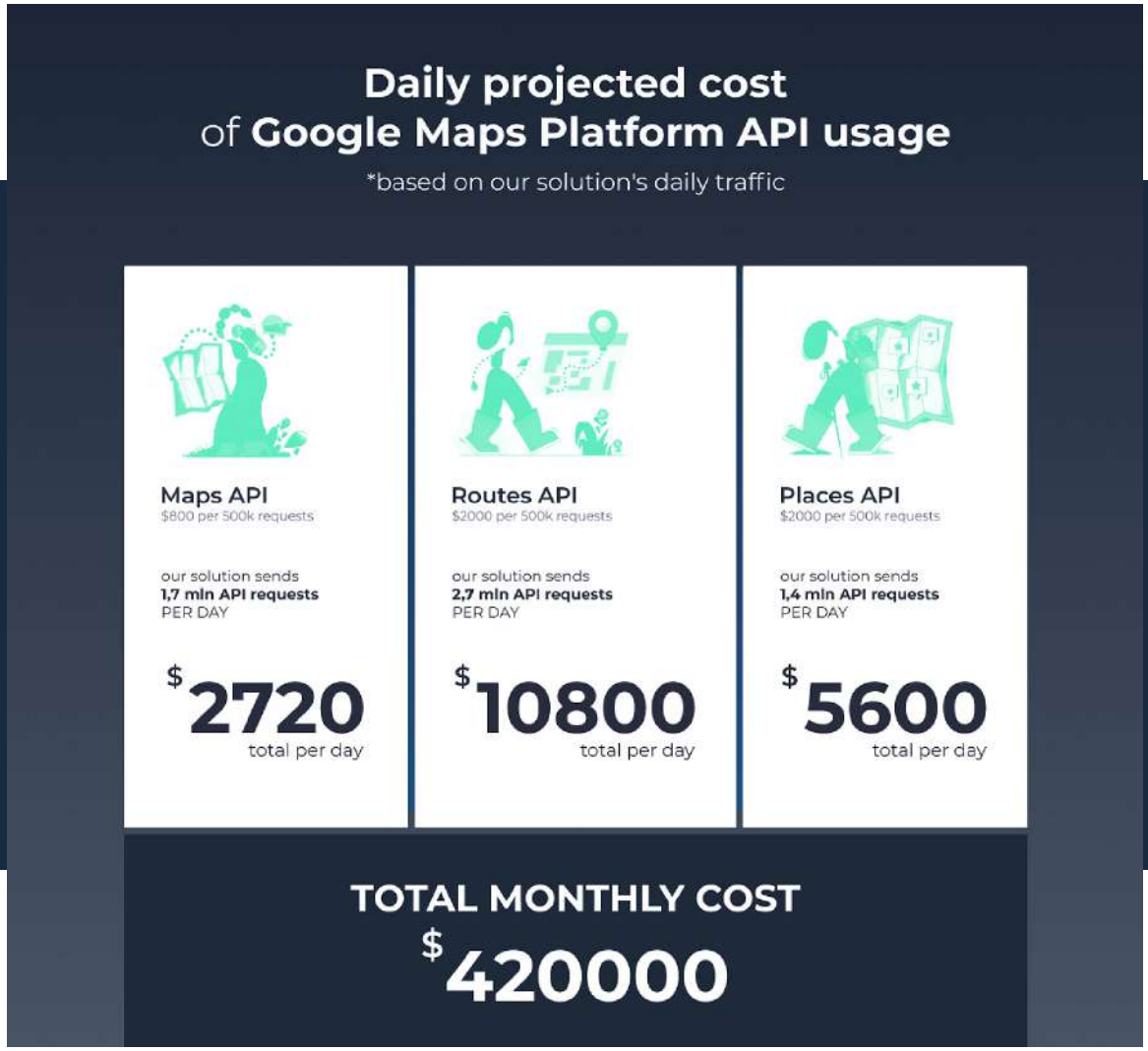
The logistics industry has somewhat different needs than, let's say, users of on-demand apps like Uber or DoorDash. Freight forwarders often navigate maps using postcodes rather than addresses. In Trans.eu, we knew we needed to give them an opportunity to search through a large database of geospatial information for specific coordinates.

We could have used Google Maps APIs to build the app, but considering what we wanted to achieve and our large user-traffic, it would have cost us at least 420 000 USD per month – stemming only from the map-related API requests. Let's take a look at what exactly contributes to this total cost.

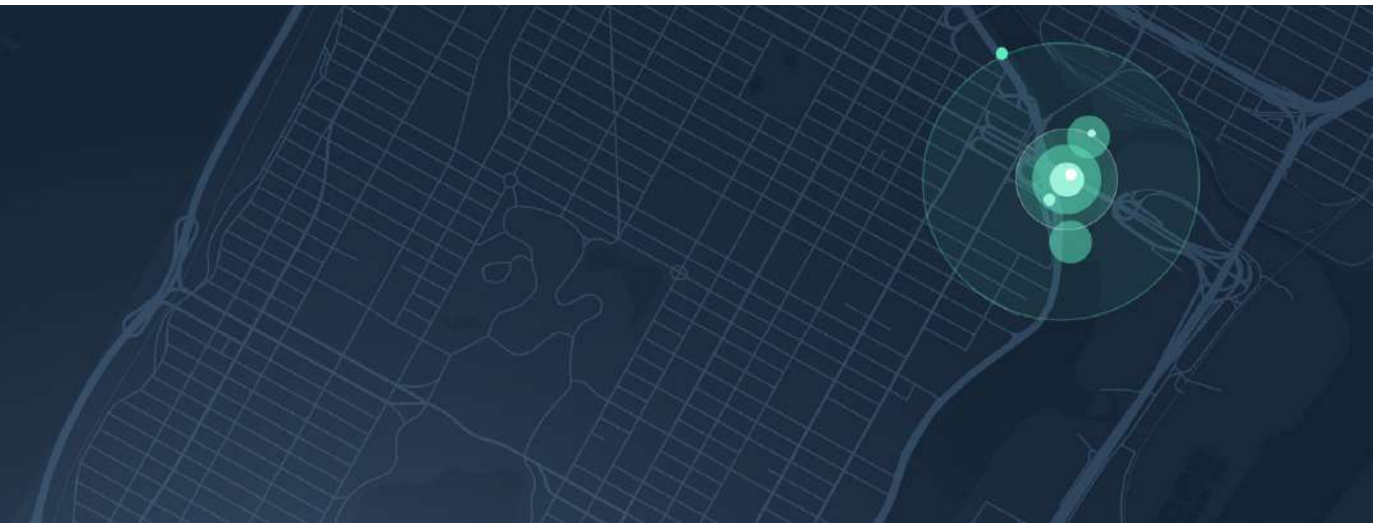
Map APIs usage pricing: Google Maps

Below, I'm presenting the usage costs for the selected APIs at 5,8 mln requests per day (the actual usage in Trans.eu). The totals were calculated using the official pricing breakdown offered by Google Maps.

We are aware that just with our daily usage, we already go beyond the scale of the offer, so we presume we could have negotiated a discount, but that would still generate hefty usage bills.



As you can see, the total cost for building a logistics platform with a large volume of traffic on ready-made APIs is immense. We found we wouldn't be able to create a sustainable business model for Trans.eu with the commercial solutions available. Opting for OpenStreetMap was the best option in our case. In fact, OpenStreetMap also gave us additional benefits.



OpenStreetMap benefits

Data consistency

This was important to us because we built the visual layer for Trans.eu with OpenStreetMap as well. We wanted to avoid a situation in which the search function works for the coordinates, but doesn't display them on the map (or the other way round: something visible on the map cannot be found in the search field). That could have happened if we used a different API provider.

Greater influence on the app behavior

As mentioned earlier, in the logistics industry, operators often search their destination through postcodes. We wanted to build a custom geocoding service for that. Users can simply type in, e.g. PL5 4RL, which is a typical postcode format in the UK, to find the relevant area.

Greater flexibility

We also wanted to leave the door open for additional data inputs. This flexibility is a great added value that allows businesses to freely expand the tools in the directions they want. We are currently working on marking warehouse entryways so that truck drivers can be directed to the exact point where they can drop off the cargo.

OpenStreetMap downsides

Are there any downsides to using OpenStreetMap? There's no point in beating about the bush here, of course there are:

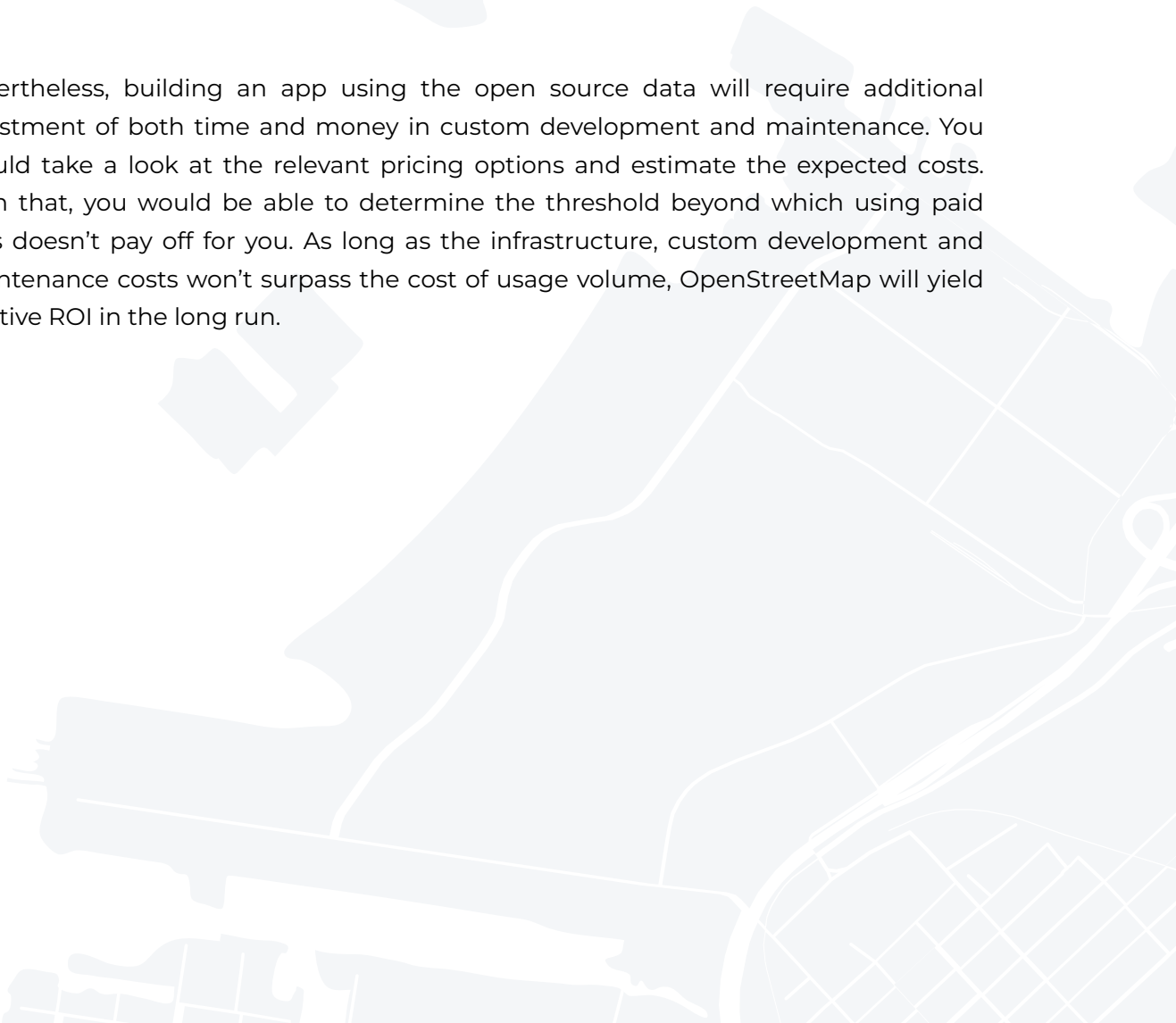
- ✕ It's an open-source solution, which means that errors can occur. Sometimes data can be missing when somebody deletes something by accident, or the data may be slightly inaccurate. This requires companies that use OpenStreetMap to monitor the data for quality. That can be outsourced, but generates additional costs.
- ✕ We secured ourselves against such errors by implementing a custom-built mechanism for updates (OpenStreetMap releases an update package daily for implementation by users), but that took us a while. It requires ongoing maintenance support and generates additional bills.

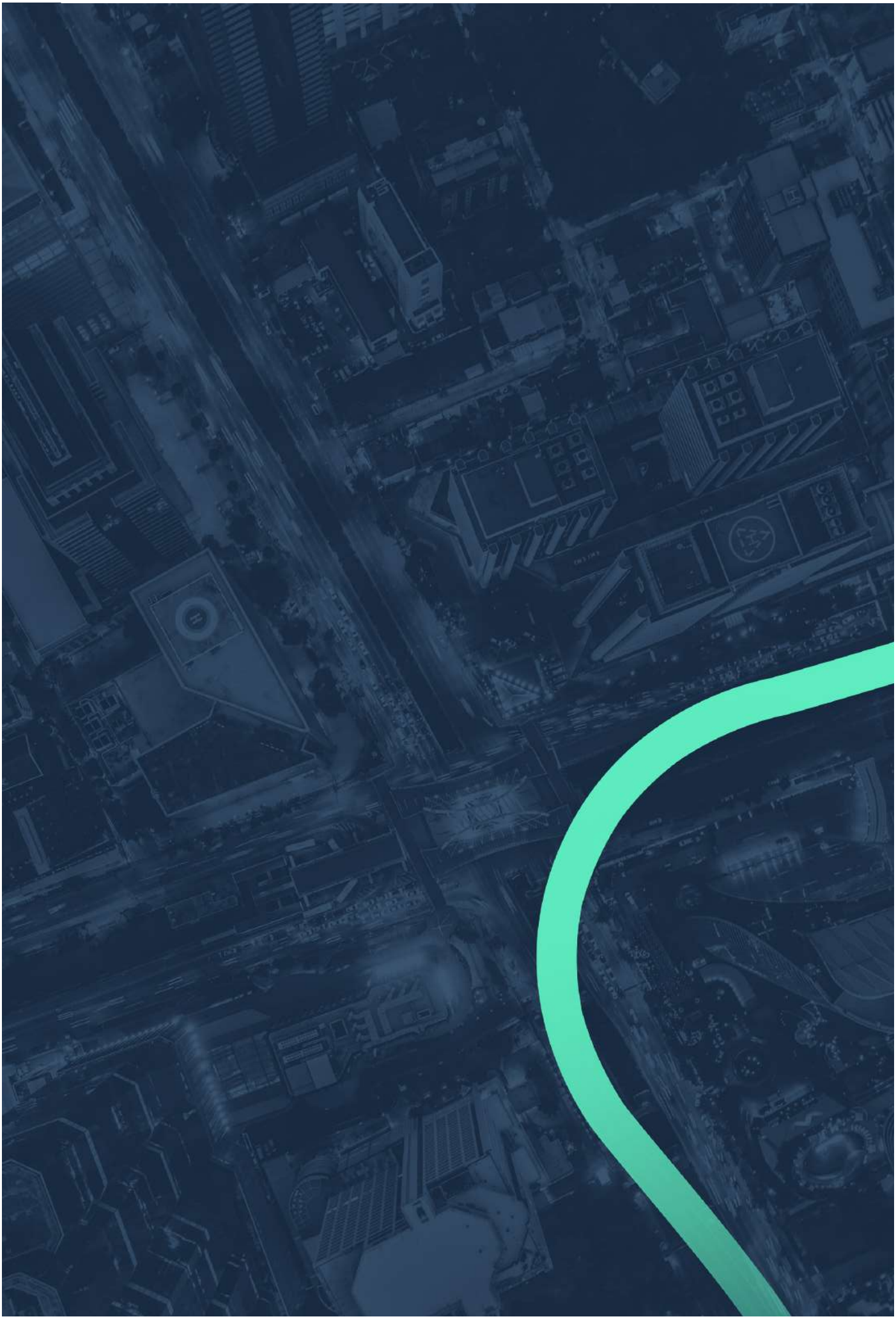
> We had to custom-build Trans.eu around OpenStreetMap from scratch, which took us a long time and cost us a lot of money as well. However, weighing the pros and cons, we decided that eliminating the monthly usage bills afterwards will be more profitable for us in the long run.

Cutting map usage costs with OpenStreetMap

What are the conclusions here? OpenStreetMap is a good map data source and it can help you eliminate the bills for map APIs usage.

Nevertheless, building an app using the open source data will require additional investment of both time and money in custom development and maintenance. You should take a look at the relevant pricing options and estimate the expected costs. With that, you would be able to determine the threshold beyond which using paid APIs doesn't pay off for you. As long as the infrastructure, custom development and maintenance costs won't surpass the cost of usage volume, OpenStreetMap will yield positive ROI in the long run.





Chapter 7

Top 10 open-source tools for location-based startup development

Developing a location-based app requires a plethora of tools to cover various areas of operational significance: ranging from website and app analytics, through mapping and GIS solutions, to process automation and back-office management. You can cover all of the above using a rich offer of available SaaS software, or you can decide to save your budget on recurring monthly payments and invest in open source instead.

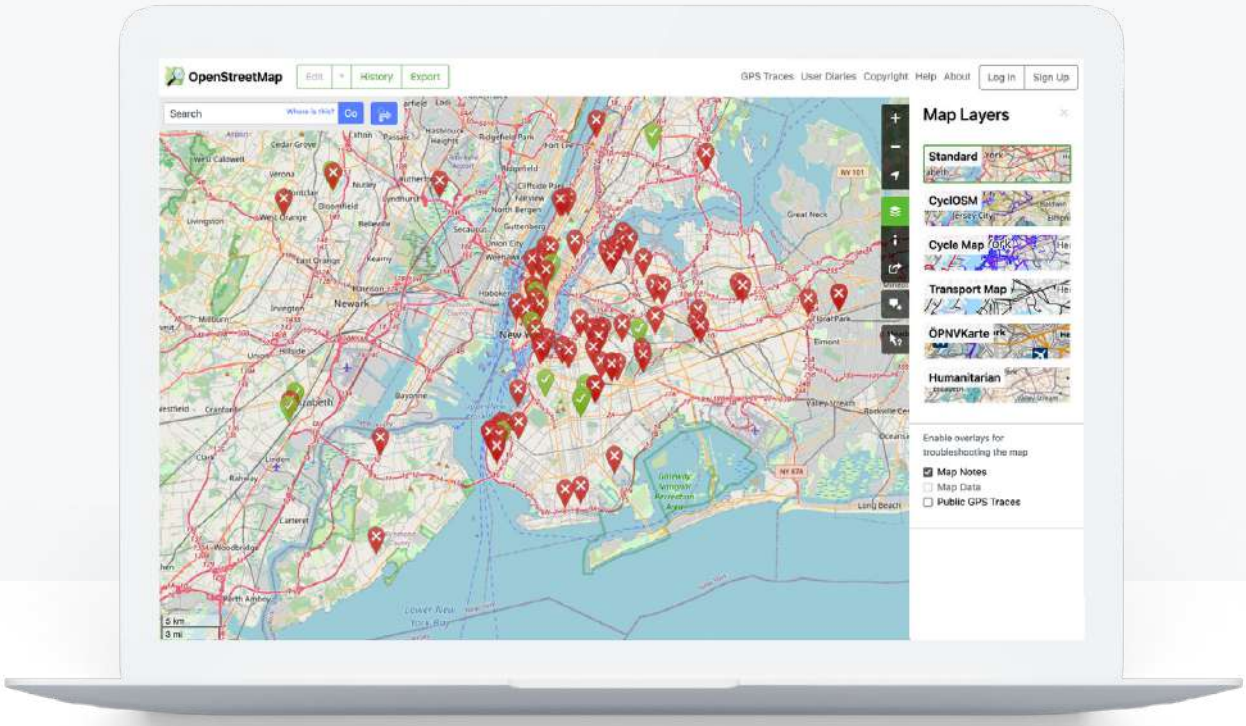
The latter can often fully cover your needs without the involvement in custom development, although there are also a number of situations where that will be unavoidable.

The pros and cons of going open source is a vast topic in itself and requires a deep analysis of your specific situation, which we can cover in the future, should you be interested in us doing so.

Today, though, I'll tell you about the top 10 open-source tools you definitely should consider when building your own location-based app.

OpenStreetMap

OpenStreetMap – raw location data source of your choice



OpenStreetMap. Something me and my colleagues already wrote about when we were comparing the best map APIs, both from features perspective and pricing (psst, follow the links, if you'd like to learn more).

OSM is the world's biggest community-driven geographic database. The project started in 2004 (before Google Maps) and was focused on mapping the UK. Nearly two decades later, it serves as a source of geodata not only to its community, but also to some of the most well-known companies in the world, eg Apple, Facebook, Amazon, Uber, Snapchat, Mapbox and more.

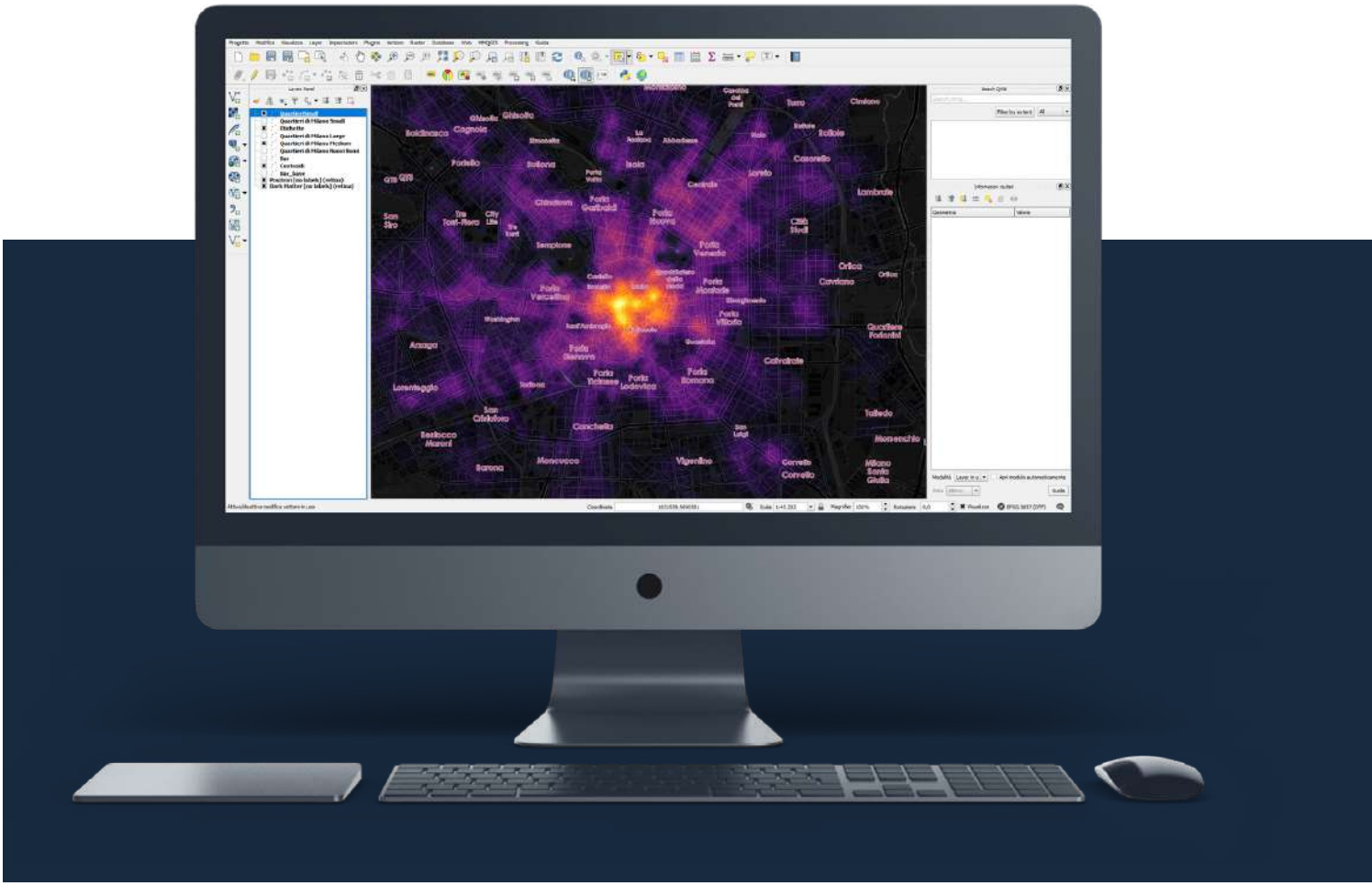
At the first glance, it may seem like OpenStreetMap doesn't provide much – only raw geodata. No familiar map interface, no navigation, no geocoding. And while that is something a regular user needs, a location-based product is worthless without that data. You can develop your own UI, geocoding service or even custom data visualizations with relative ease, but there's no way you're going out there to map the entire world, especially if you're just starting out.

While, yes, you can use other map API providers, their pricing can take a toll on your available capital, especially if you scale rapidly. So all in all, OSM is definitely a must-to-consider.

QGIS – perfect for geospatial data processing

QGIS – yet another world's leading software on our list. As the name implies, it's a geographic information system (GIS) – an application that enables viewing, editing, printing, and analysis of geospatial data. The data itself can come from any source, including aforementioned OpenStreetMap, both from .osm XML files or via appropriate APIs, dxf MapInfo, PostGIS and other industry-standard formats.

Being an open-source software solution, it means you also gain access to an international community of users who enhance QGIS's functionalities through 3rd-party plugins.



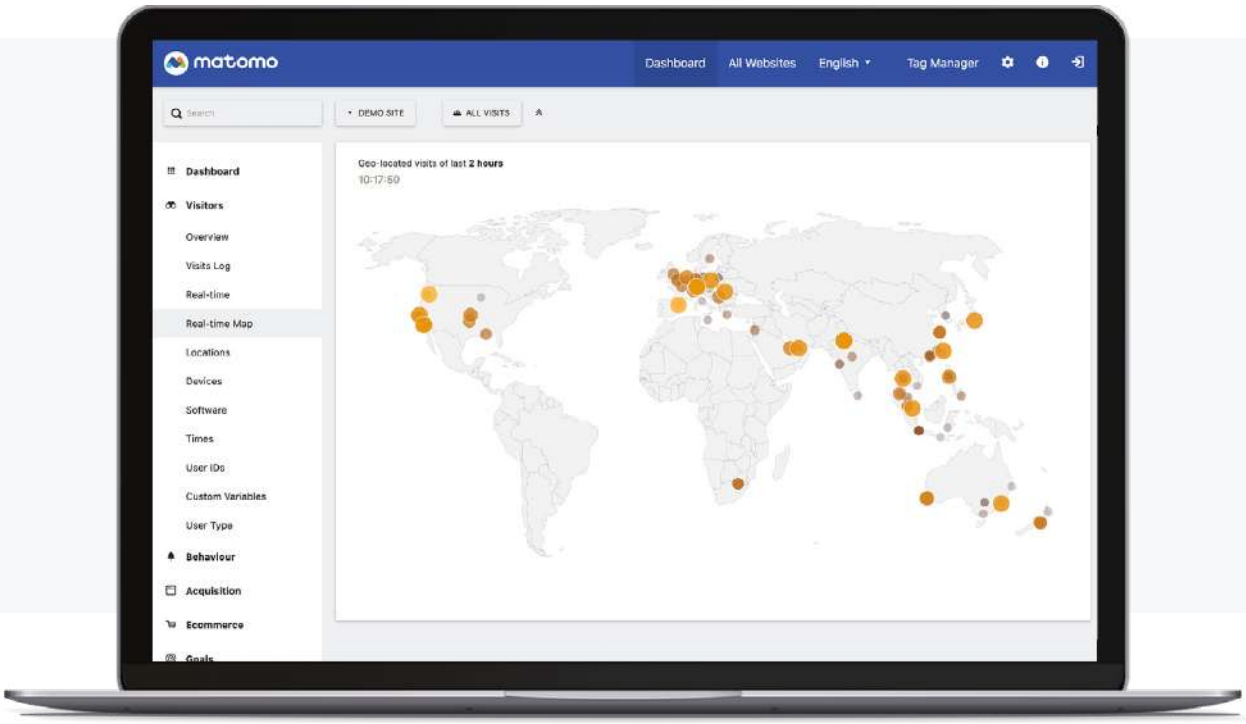
That might not even be needed, considering QGIS out-of-the-box supports:

- **raster layers processing;**
- **vector layers processing;**
- **mesh layers processing;**
- **can georeference raster images;**
- **can perform geoprocessing in the likes of the tools found in ArcGIS;**
- **can geocode;**
- **and more.**

It can also display multiple data layers, which comes in handy when you want to compare or mash together information from different sources. Last but not least, you're free to modify and develop custom modules, should such a need arise, without having to rely only on those features paid GIS solutions offer.

Matomo gives your location-based apps complete control of target audience data

Matomo, formerly known as Piwik, is a privacy-driven open-source web analytics platform. You are in full control over the gathered data and can even host it on-premise without using Matomo cloud services.

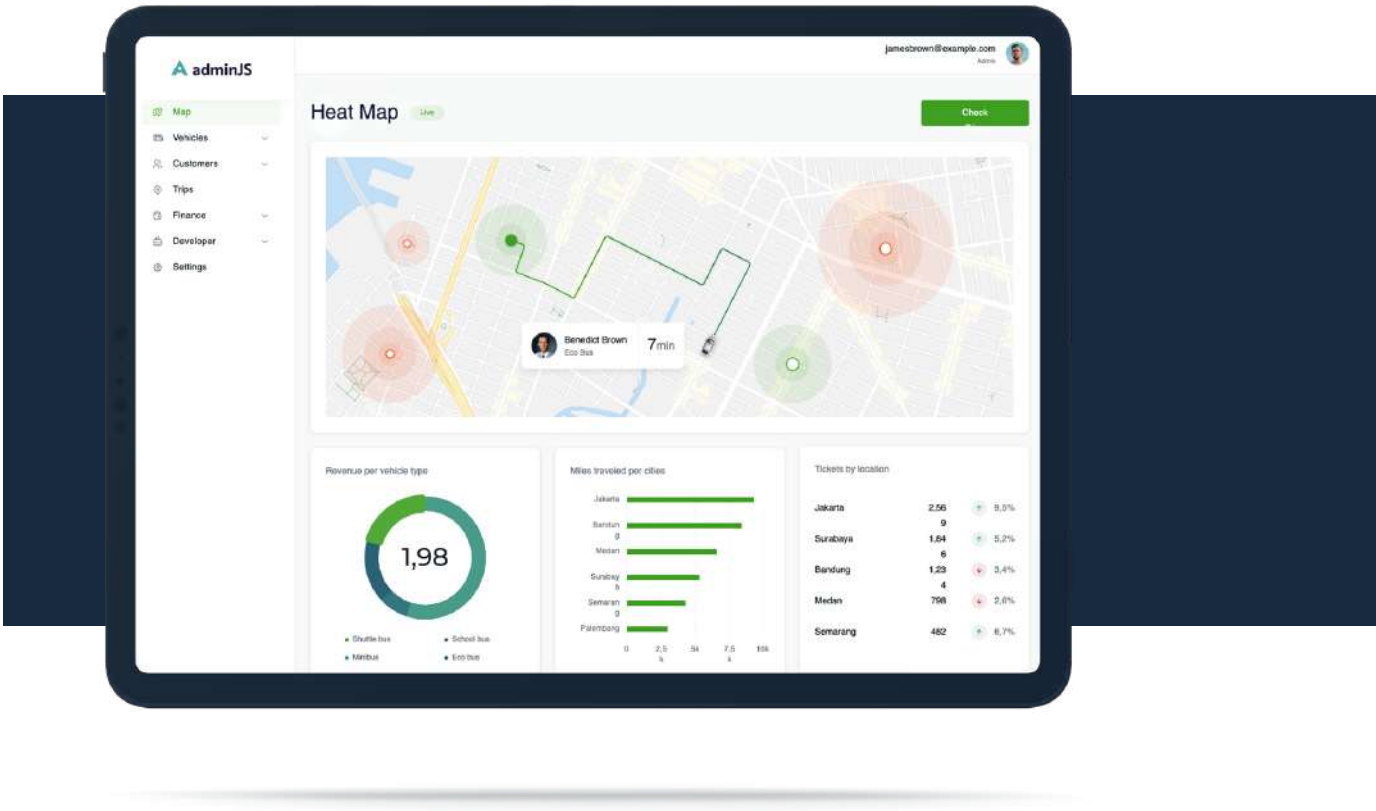


What's important to notice is that, contrary to your most-probable initial impression, Matomo doesn't compete with Google Analytics only. It also offers heat mapping and sessions recording capabilities like the ones you can find in Hotjar or its alternatives (the full comparison tables can be found here and here respectively). If you decide to switch from GA, Matomo supports historical data import, so you won't have to sacrifice what has already been collected.

And once again, as it goes with open source, you are free to modify and extend Matomo's functionalities in accordance with your business needs.

AdminJS speeds up your location-based app development process

Every location-based app requires an admin panel. AdminJS is the world's most popular auto-generated admin panel for Node.js applications. Unlike some of its competitors, it is a database-agnostic solution, which means it doesn't impose any particular database schema in order to work. It serves as a plugin that uses various ORM/ODM adapters to connect to whatever database you are using.

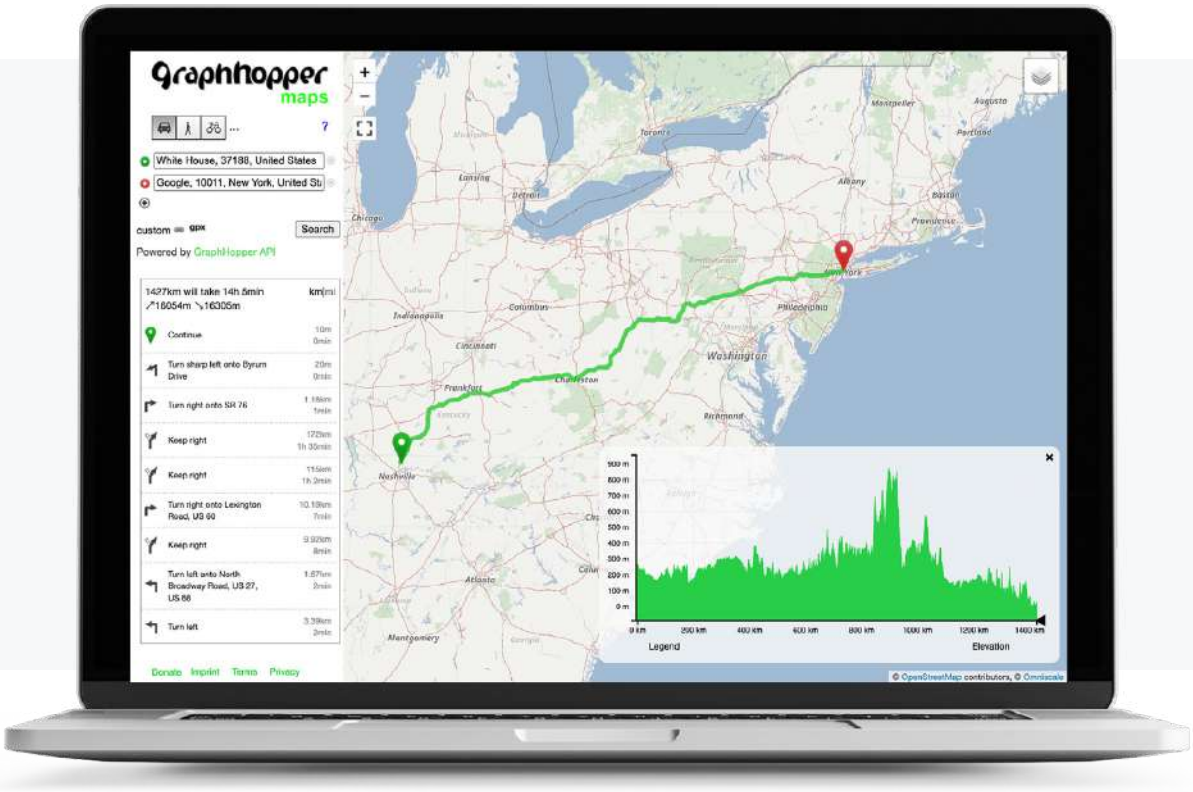


Installed in a matter of hours, it provides you and your team with CRUD operations and can be customized to operate as a back-office system even for your non-technical employees.

We've used AdminJS to develop a mobility startup – Bussr (click to open the case study) – and integrated the required location-based services within its administration panel, which provided their Operations Department with the ability to assign daily bus routes directly on the map, instead of manually uploading .CSV files with geo coordinates into the database.

Customizability of AdminJS offers companies freedom to develop their own modules that support their particular needs, be it for e-commerce, FoodTech or location-based products.

GraphHopper – crucial component of many location based apps



GraphHopper is one of those libraries that make open source as powerful as it is. GH is a fast and memory-efficient routing engine that allows you to calculate the time, distance and turn-by-turn instructions, amongst other attributes, for a route between two or more geographical points.

Based on OpenStreetMap and Shuttle Radar Topography Mission data (you can also connect your own sources if the above are not useful), GraphHopper can also use different routing algorithms, like Dijkstra, A* and its bidirectional versions. More so, to optimize the speed of their routing engine for long routing paths (continental size) and avoid heuristic approaches, GraphHopper uses contraction hierarchies by default.

Use cases for this can be found in on-demand applications, most delivery applications, geolocation tracking apps, public transport platforms, taxi apps or logistics companies.

CAMUNDA

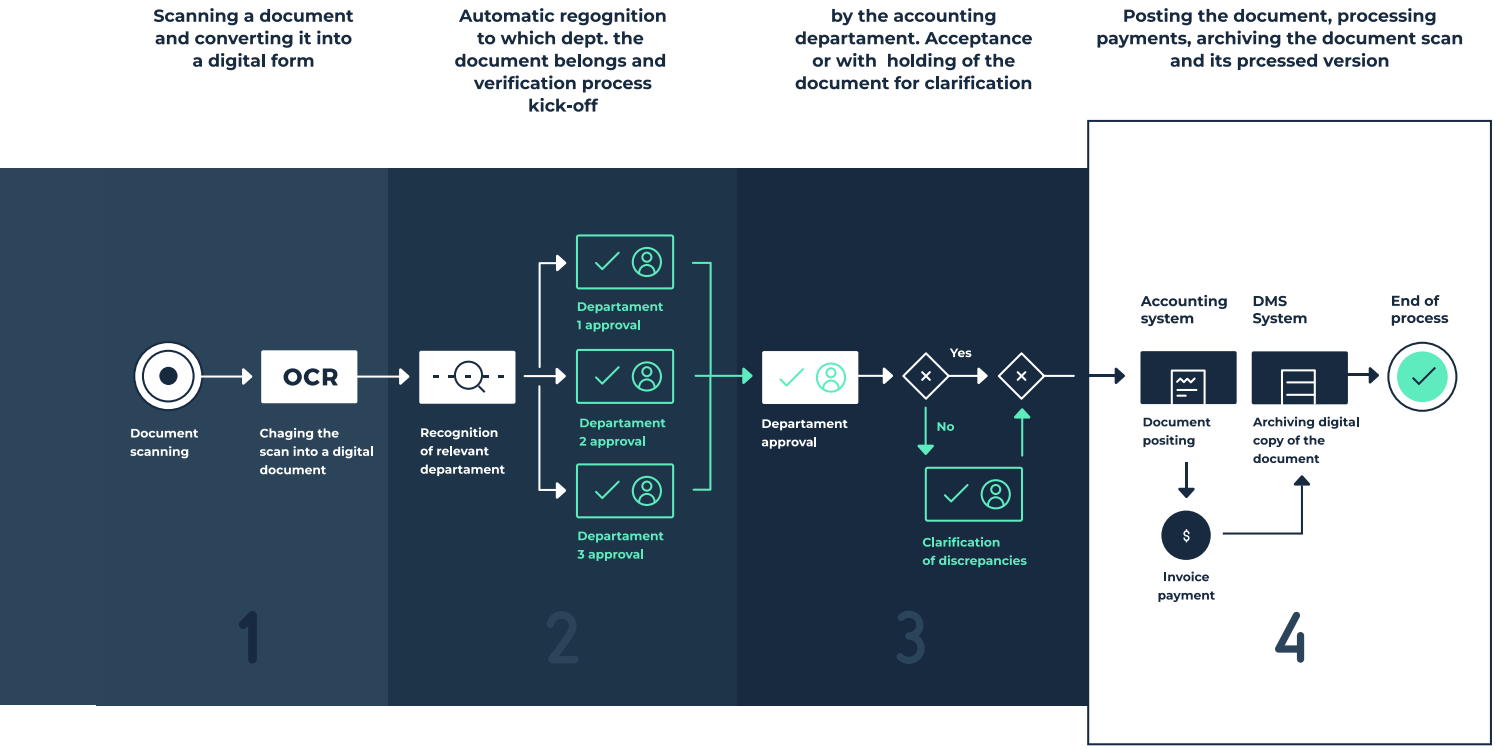
Camunda – delegating tedious tasks to machines

Camunda is an open-source workflow and decision automation platform that utilizes both BPMN and DMN standard-compliant workflow engines.

To put it simply, process automation allows you to automate (obviously) all the tedious processes that take too much time to execute manually and can be performed without human intervention.

Document processing is a good example. We've recently completed the development of Camunda-based PoC (you can view the case study here) for Alphabet – one of the largest car fleet management services, operating in 31 countries with a fleet of over 720 000 cars.

HERE'S THE PROCESS AUTOMATED BY THE PROTOTYPE:



As you can see, it eliminates a variety of manual tasks and allows your employees to focus on other, more important tasks.

It might be an overkill at the very beginning of your startup journey, but as your business grows, Camunda becomes more and more attention-worthy.

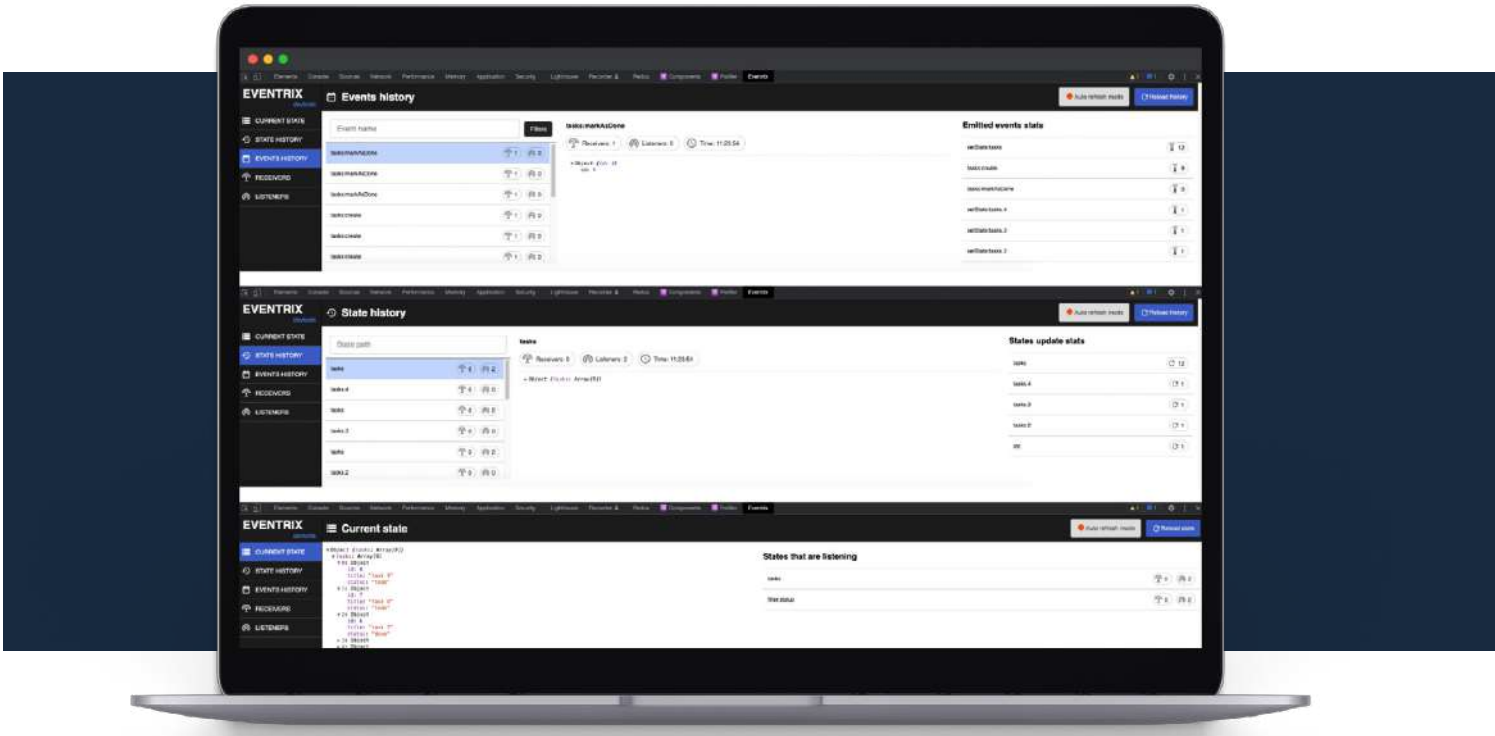
Cesium enables 3D-capabilities within your location-based application

Cesium in an open-source JavaScript library for 3D geospatial visualization. It is designed to create interactive web and mobile apps using 3D globes and maps with robust interoperability and support for massive datasets in mind.

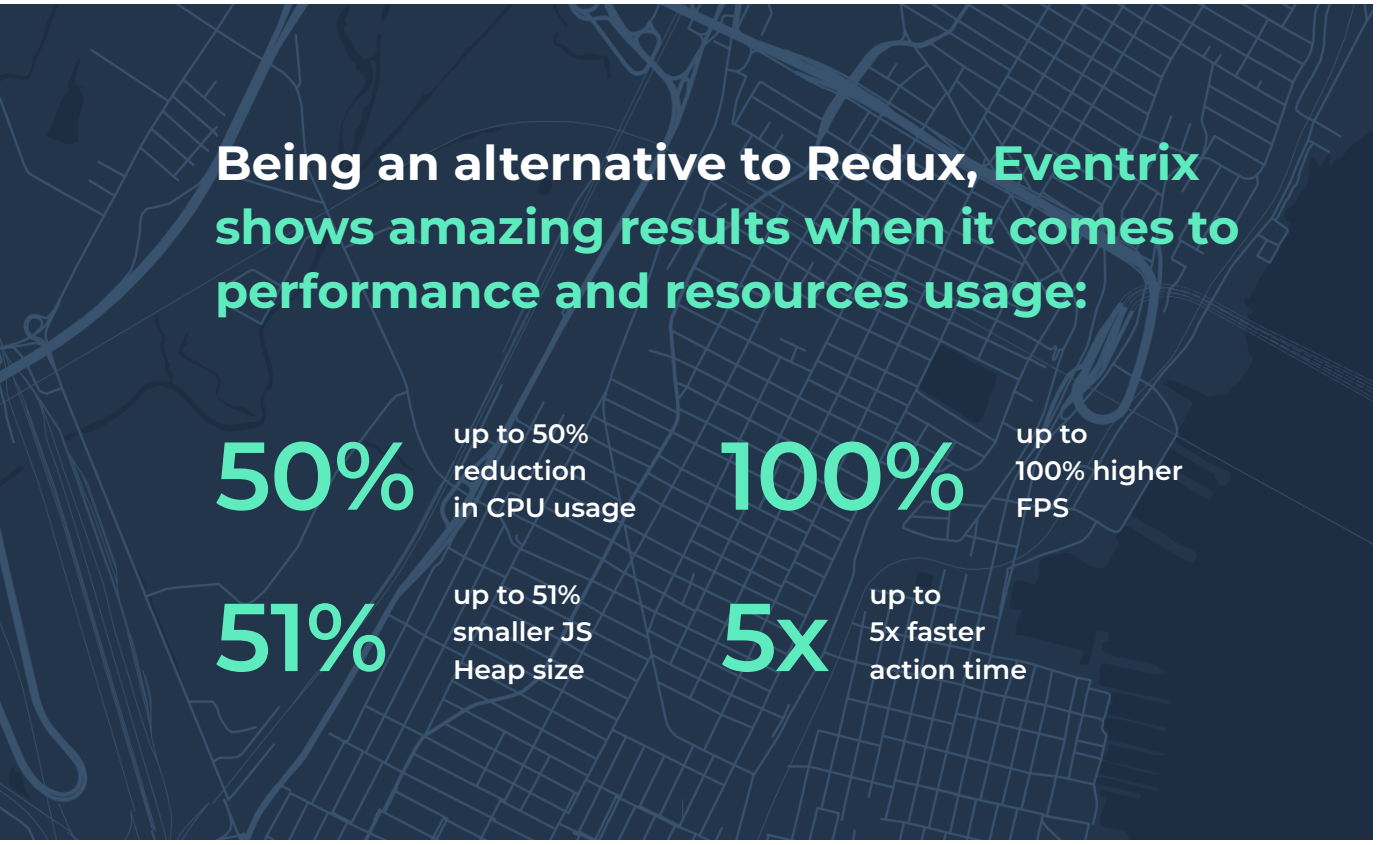
Using a high-precision WGS84 globe, you can visualize and analyse data streamed from a source of your choice, including their commercial Cesium ion platform that transforms your terrain, imagery, and other location data into optimized, ready-to-stream 3D Tiles content.

Eventrix powered by RST

Eventrix enhances performance of your location-based applications



Eventrix is our own open-source, predictable and highly efficient state management library for React applications. Developed for our internal needs when working with our enterprise partners at Trans.eu – one of the largest logistics companies in Europe and Asia – it solves the issue of sharing information between various components of our applications as well as their direct communication.

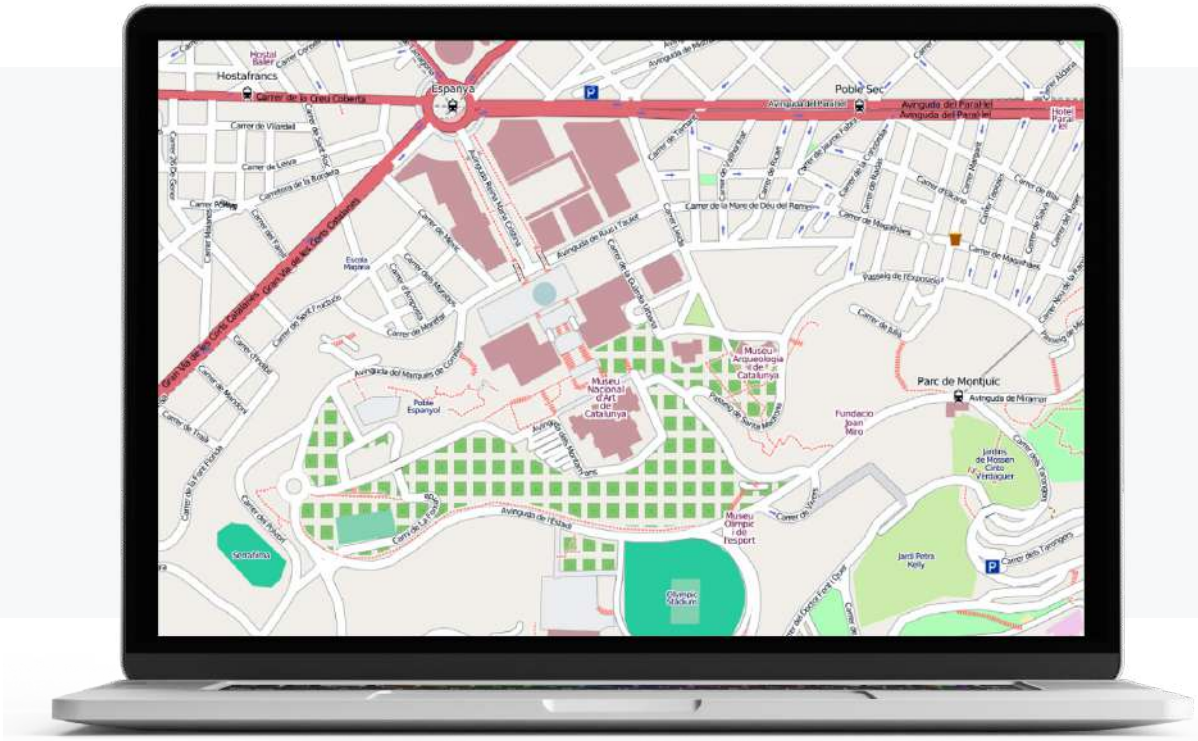


Eventrix powers the Trans logistics platform and handles over **62,000 concurrent daily users** with more than **13,000 state changes per second** and **300 messages sent also every second** (check out our case study here).

As you can see, it can withstand enormous load and thus is perfect not only for small projects, but also enterprise-scale location-based apps.

mapnik

Mapnik – toolkit for mapping apps

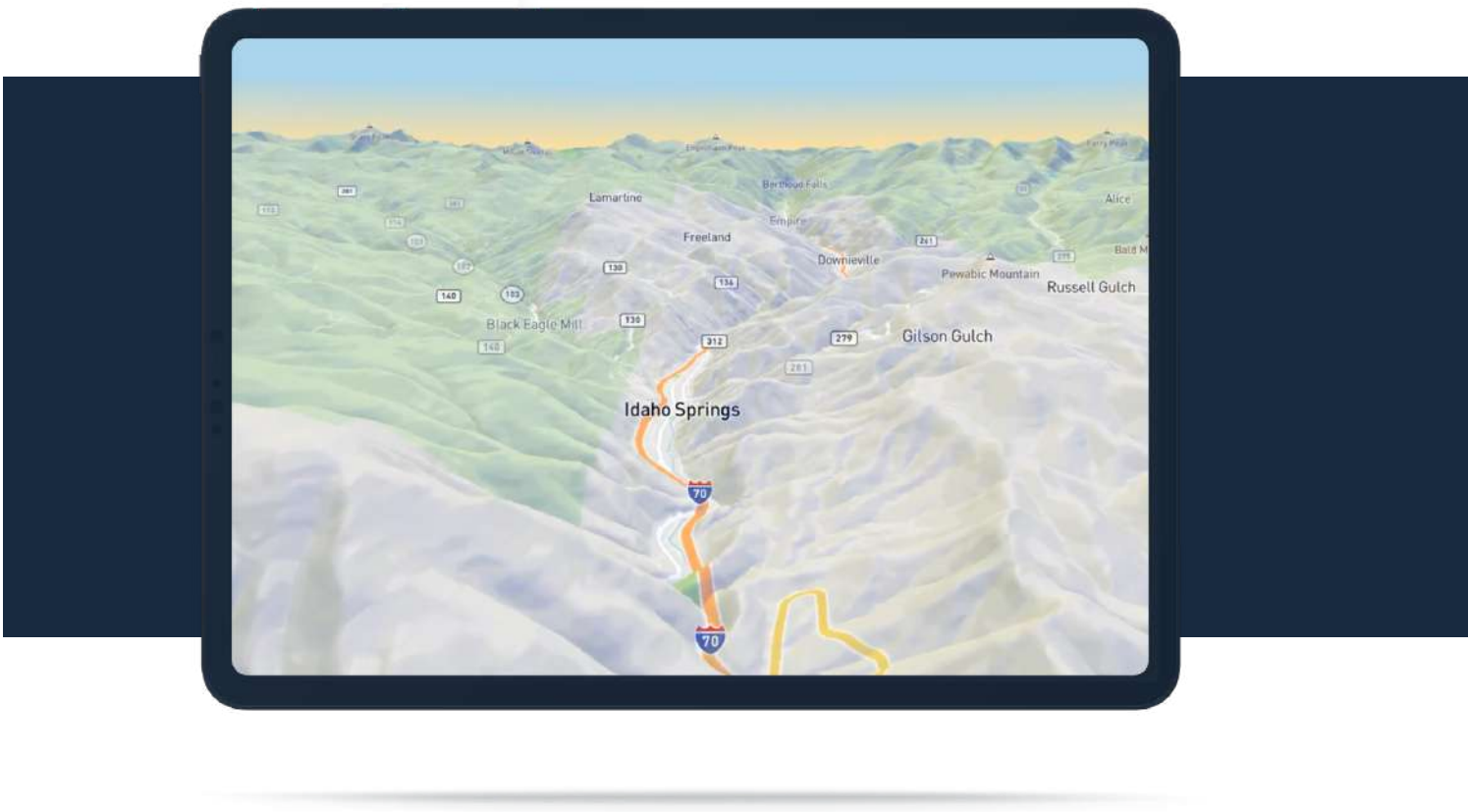


Mapnik is an open-source mapping toolkit for geospatial visualization and processing. To put it simply, it is a collection of geographic data like maps, layers, features, and geometries. Its OS-agnostic approach means it can be implemented on any server environment.

Using community-developed plugins, Mapnik supports industry-standard formats to read both vector and raster datasets as well as custom Shapefile, PostGIS and GeoTIFF formats. Last but not least, it can render OSM data into custom maps.

Which it does even for the OpenStreetMap.org default layer. Mapnik is also used by other location services, eg Mapbox, CartoDB or MapQuest.

Mapbox GL powers interactive mapping apps



And to end (I didn't mean to use homophones, I swear) today's list, let's take a look at a phenomenal JavaScript library for creating interactive and highly-customizable maps in the browser using vector map tiles and WebGL – Mapbox GL.

Developed by Mapbox, Mapbox GL is a client-side tool that allows you to build a mobile or web app using modern mapping technologies.

Use cases include, but are not limited to:

- visualizing and animating geodata in both 2D and 3D;
- adding a georeferenced images to your maps;
- processing of live real-time geodata (eg fitness trackers, travel apps, traffic jams notification system, weather apps and more);
- dynamically displaying and styling custom data on a given map; or
- programmatically adding custom markers and pop-ups to your maps.

The variety of examples is huge, so I recommend you to have a look at their official examples page. Mapbox GL can definitely become of use during your location based app development process, especially if you decide to provide your users with uniquely styled maps.

Recap of tools for geolocation app development

To conclude, here's a short cheat sheet of what's good for what.

OpenStreetMap

Offers the largest source of raw geographic data.

QGIS

One of the best, if not the best, open-source GIS solutions available on the market and can be an alternative to paid products in the likes of ArcGIS.

Matomo

One of the best on-premise privacy-driven analytics solution.

AdminJS

An amazing tool for building a Node.js-powered internal tools: be it a basic admin panel or a features-packed back-office system.

GraphHopper

A route optimization engine, must have for most location-based products that require advanced routing algorithms.

Camunda

A fantastic tool for automating your processes and eliminating manual labour that don't particularly required any human intervention and can be performed much faster and with greater results by a machine.

Cesium

What you're looking for in you need to visualize your 3D geospatial data within your web or mobile applications.

Eventrix

An enterprise-scale-ready, resources-efficient state management and global state centralization library for your application's frontend.

Mapnik

An open-source software solution that powers such giants mapping apps as OpenStreetMap and Mapbox.

Mapbox GL

Speaking of Mapbox, Mapbox GL is the ultimate tool for creating interactive maps with unlimited customization options.

We hope you find this list useful! If you're looking for custom recommendations regarding the specific business needs of your geolocation app or an enterprise solution – feel free to drop us a line at offer@rst.software, and we'll do our best to help you out.

About the authors

Andrzej Lewandowski

Certified AWS Solution Architect

Advocate of DevOps culture. Passionate programmer. Believes that solving programming problems is the meaning of life. Loves identifying new technologies and “proof of concept” solutions.

Wojciech Tomaszewski

Location-based Solutions Team Leader

Expert in geospatial development starting from data sourcing and database design to styling and generating maps. Thinks that visualizing data can be quite fun.

Marek Ziółkowski

Chief Solutions Officer

Alumni of the Faculty of Electronics at Wrocław University of Technology and is focused on everything that is innovative in technology, especially when it comes to cloud solutions.

Ross Krawczyk

Senior Business Generalist

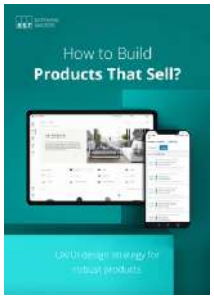
Senior business generalist with a mindset focused on getting things done in the shortest possible time with the best possible quality.

Magdalena Jackiewicz

Senior Content Writer

Content writer with several years of experience in the IT sector. Driven by finding the simplestpossible way of explaining complex issues.

More ebooks



How to build products that sell?

Product Design

Transitioning from a monolith to microservices

Cloud & DevOps

Want to build a location-based startup?

Learn more about our geospatial development services.

LOCATION-BASED SERVICES

→

Or simply drop us an email at hi@rst.software

RST

www.rst.software