**The Gans Report**

Or how I learned to love APIs

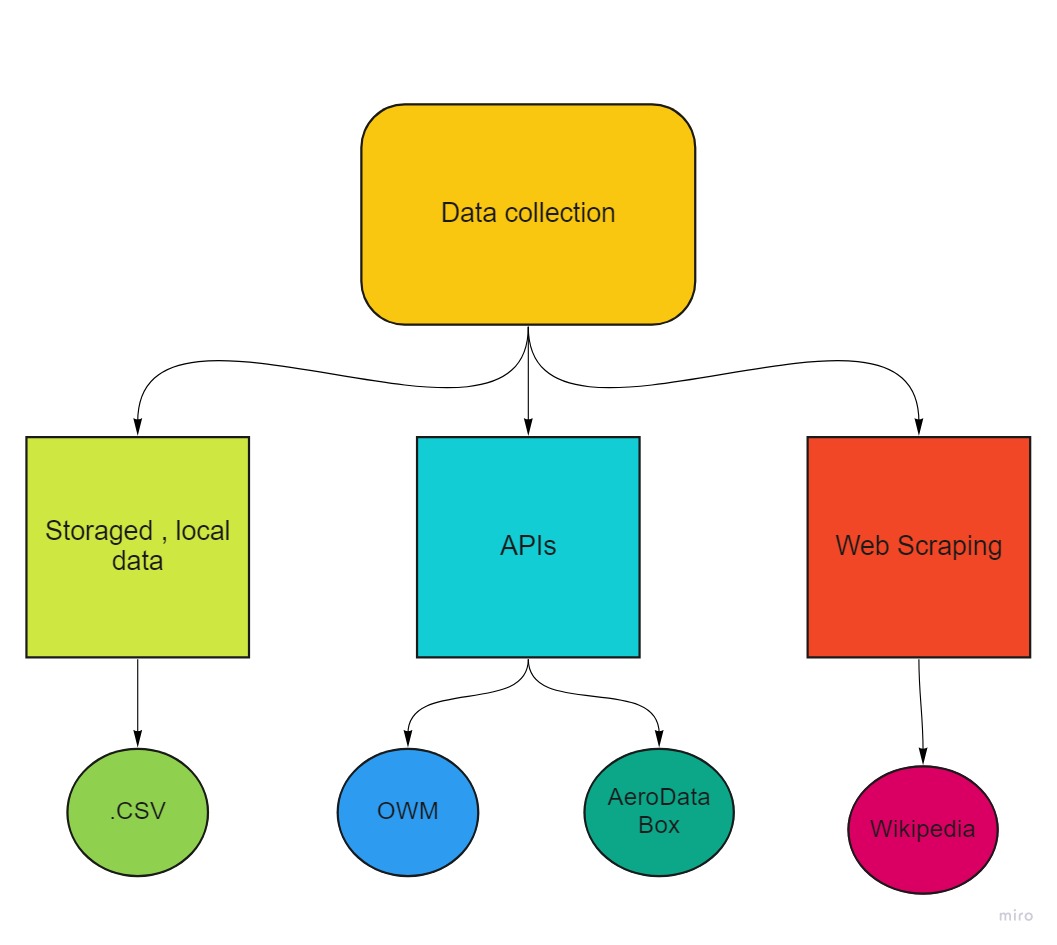
Today I want to talk about the first steps of Data Engineering. We want to build a database for our fictional company name Gans. Gans is a car and scooter sharing company and their main target group are tourists. They want to expand into the German market, and we need to build a database to help the company with their business decisions.

At first, we have to decide which data is needed for that. Normally as a junior Data Engineer someone higher up would tell us which data is desired. But for our own test project we decide to get following data for German cities:

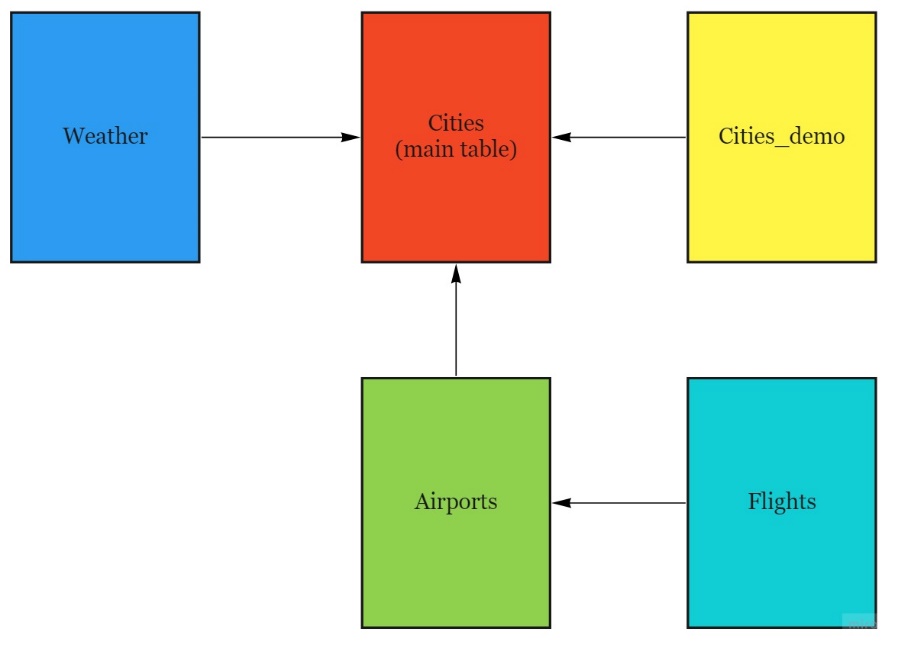
* Geographical data
* Demographical data
* Weather data
* Airport data
* Flight data for those Airports

We chose 27 German cities with an airport nearby, which have a regular commercial flight scheduled. Now that we know what we want, we must figure out how to get this data.

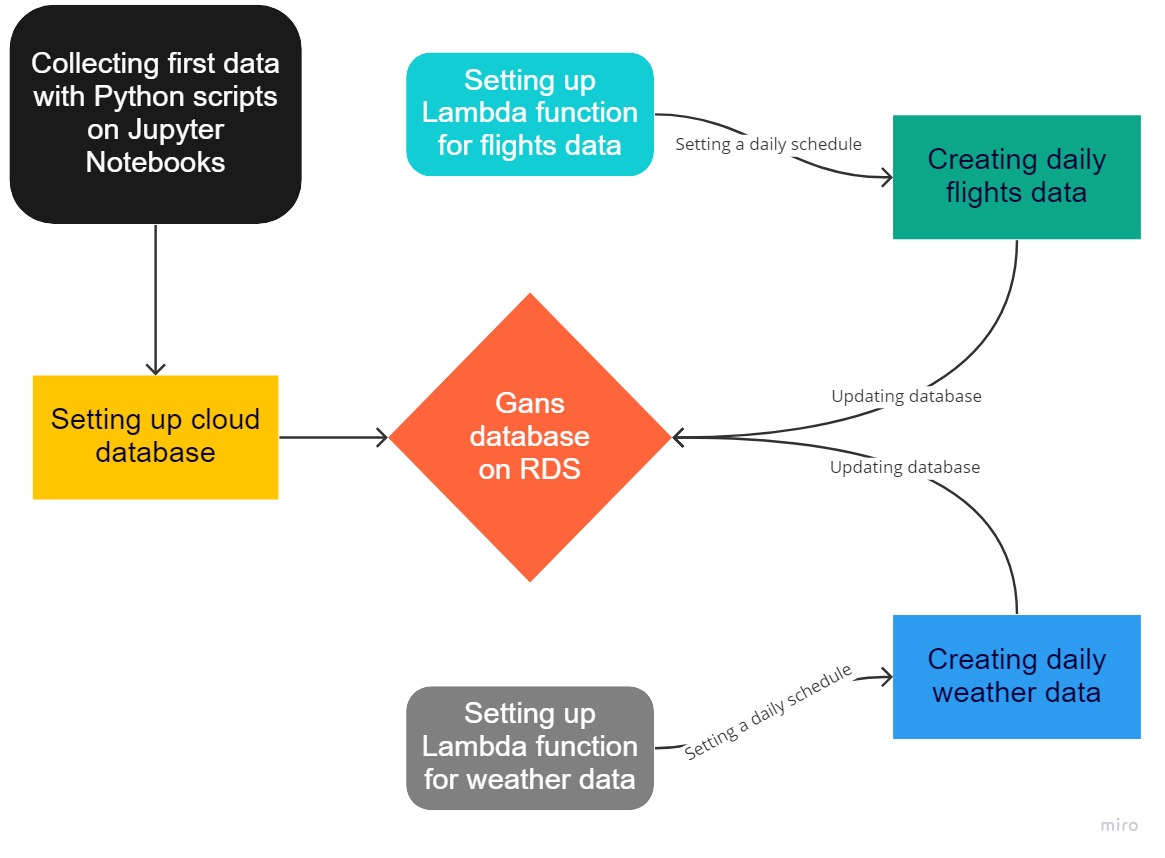
We have 3 routes to explore. For airport data we will extract it from a .csv file. For the geographical and demographical data, we will use the web scraping method. This method is often very time consuming and a grey legal area. It should be used in a last resort manner if the data is not available in another way. The weather and flights data will be retrieved through the Open Weather Map API and the AeroDataBox API. Both are available at the RapidAPI.com website. APIs are a much better way to retrieve data and should be the preferred method of choice. They can have drawbacks like not every data is available, that is available on the website or that you have to pay for (better) access.



Now that we know how to collect our data, we need to store it. We want a cloud-based solution and used the Amazon Web Services (AWS) for that. We installed a MySQL database on the Relational Database Service (RDS) on the AWS platform. We created a main table with the cities and linked the demographical data table, the airports table and the weather table with the main table and the flight data with the airport table.



After initializing our database, we need to keep it up to date. While the geographical, demographical and airport data aren’t that much in our timeframe, we want to update our flight and weather data daily. For that we are using the AWS lambda functions. With those we can execute scripts one time or repeatedly on a chosen schedule by using the EventBridge (CloudWatch Events) tool.



Conclusion: APIs and Amazon Web Services are well worth the time (and probably the money) when set up properly. Web scraping should be used cautiously and as a last resort method. And data engineering can be FUN!