Context

Two customers of Sikoia are interested in using company data to support their lending businesses. The countries they want to cover are: **United Kingdom**, **Germany** and **The Netherlands**. In order to source the data, Sikoia chooses two third-party providers: **Third Party A** and **Third Party B**.

Both providers specialise in integrating with company registries across different jurisdictions and provide a standardised view of company information. Third Party A is a new, cool start up with plenty funding and a modern tech stack. Third Party B is an old, legacy dinosaur that has been in the business for over 20 years (rumour has it that the last time they updated their stack Bush was still president). Unfortunately, however, they also have long-term partnerships with some registries that gives them access to some data that other companies cannot access.

Your task will be to use these third-party providers to come up with a Sikoia view of a company and allow our customers to retrieve company data and build automated workflows to make lending decisions.

Third Party A

Third Party A currently only supports United Kingdom and Germany. They offer a GET endpoint with the following structure:

URL: *https://interview-df854r23.sikoia.com*
Endpoint: */v1/company/{jurisdiction_code}/{company_number}*
Example: */v1/company/uk/1111*

where:

- The endpoint returns company data for the company requested.
- The endpoint does not require any authentication.
- The only two supported jurisdictions are: 'uk' and 'de'.
- For 'uk', the only two supported companies are those with numbers: '1111' and '2222'.
- For 'de', the only two supported companies are those with numbers: '3333' and '4444'.
- Any request outside of these boundaries will throw an error.

They also gave you access to their documentation at:

https://interview-df854r23.sikoia.com/docs

Third Party B

Third Party B currently only supports Germany and The Netherlands. They don't really have any documentation, and the only information that you have been able to gather comes from an email from a not-very-responsive customer success associate who's not very good at technology (he's still trying to figure out how to record a voice message on WhatsApp). They offer a GET endpoint with the following structure:

URL: *https://interview-df854r23.sikoia.com*
Endpoint: */v1/company-data?{jurisdictionCode}&{companyNumber}*
Example: */v1/company-data?jurisdictionCode=de&companyNumber=3333*

where:

- The endpoint returns company data for the company requested.
- The endpoint does not require any authentication.
- The only two supported jurisdictions are: ‘de’ and ‘nl’.
- For ‘de’, the only two supported companies are those with numbers: ‘3333’ and ‘4444’.
- For ‘nl’, the only two supported companies are those with numbers: ‘5555’ and ‘6666’.
- Any request outside of these boundaries will throw an error.

Your Task

Your task will entail building the following endpoint.

Endpoint: Company Data
The first one is a GET endpoint with the following structure:

URL: doesn’t really matter
Endpoint: */v1/company/{jurisdiction_code}/{company_number}*
Example: */v1/company/uk/1111*

where:

- The endpoint will return company data for the company requested.
- The endpoint will not need to require any authentication.
- You will have to support ‘uk’, ‘de’ and ‘nl’.
- For ‘uk’, you will need to support companies with numbers: ‘1111’ and ‘2222’.
- For ‘de’, you will need to support companies with numbers: ‘3333’ and ‘4444’.
- For ‘nl’, you will need to support companies with numbers: ‘5555’ and ‘6666’.
- Any request outside of these boundaries will need to throw an error.

You will have to come up with the best possible Sikoia view for a company based on the data received from the two third-party providers. We leave it entirely up to you to figure out their strengths and flows by playing with their API and decide how to use them to deliver the data based on your findings. However, your endpoint will have to return one unique model across all companies and jurisdictions.

Notes

You can write your solution in the programming language you are most familiar with. The solution should ideally run locally without too many dependencies, but you can give details on how to install it in your README file. Please keep in mind that we will attempt to install it and test it locally.

There are no further requirements on the solution, but we do expect good engineering practices to be applied (although we leave it to you to choose what those should be).

Questions are not only acceptable but welcomed. Please also feel free to point out any ambiguity or unclarity with respect to this task.

You will need to upload your solution to a public GitHub repository and send us the link.
Have fun!