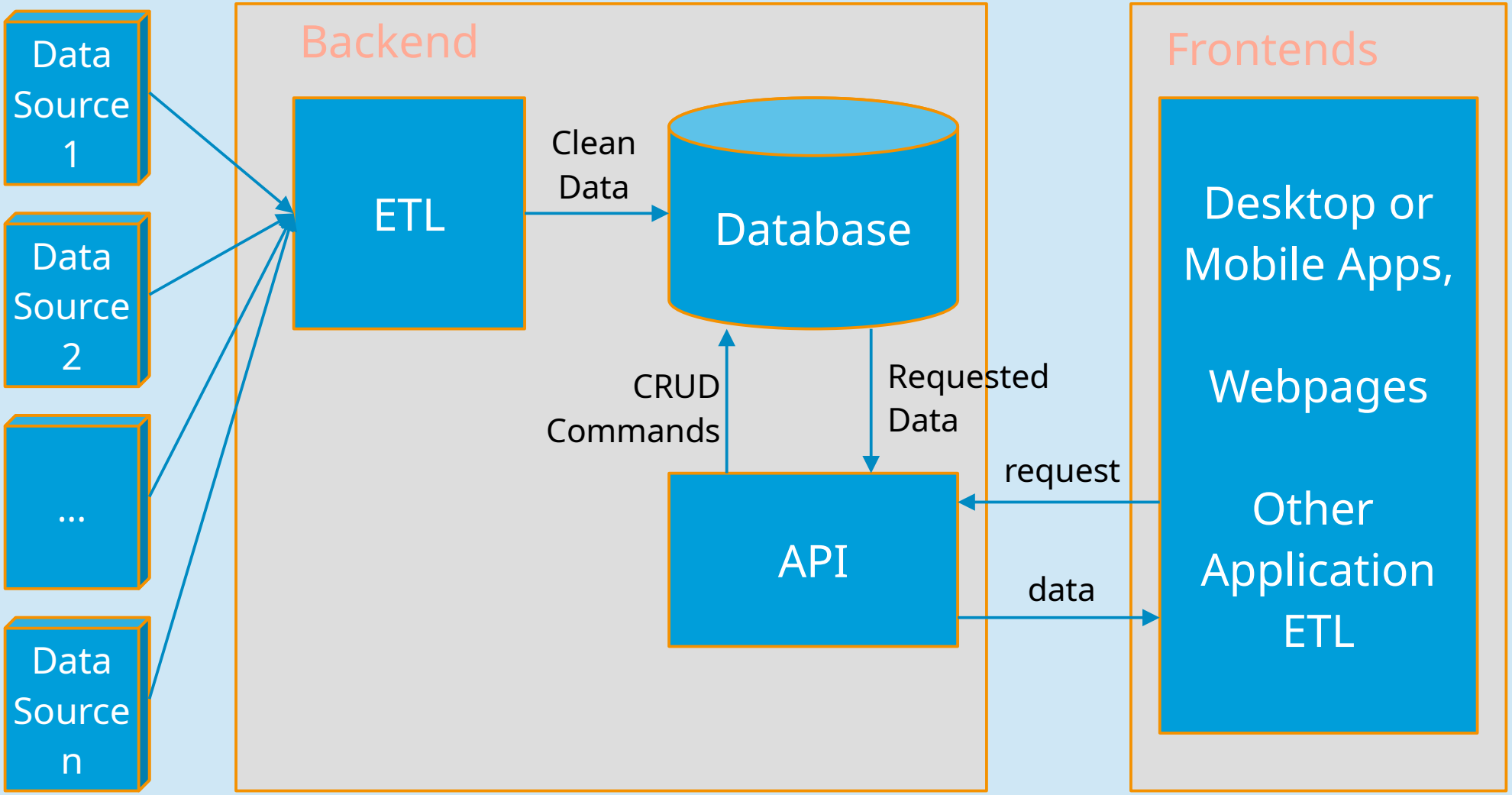
ETL + CRUD + API
Working example

Alexandre Neto

What is ETL?

- **E**TL stands for:
  - **E**xtract – Extract data from local or external sources
  - **T**ransform – Cleanse the data to improve data quality and establish consistency
  - **L**oad – Save the resulting data either locally or in a Database
- It’s data integration process that combines data from multiple data sources into a single, consistent data store that is loaded into a data warehouse or other target system.

ETL + CRUD + API


Backend
Frontends
Data Source 1
Data Source 2
...
Data Source n
ETL
Clean Data
Database
CRUD Commands
Requested Data
request
API
data
Desktop or Mobile Apps,
Webpages
Other Application ETL

What is an API?

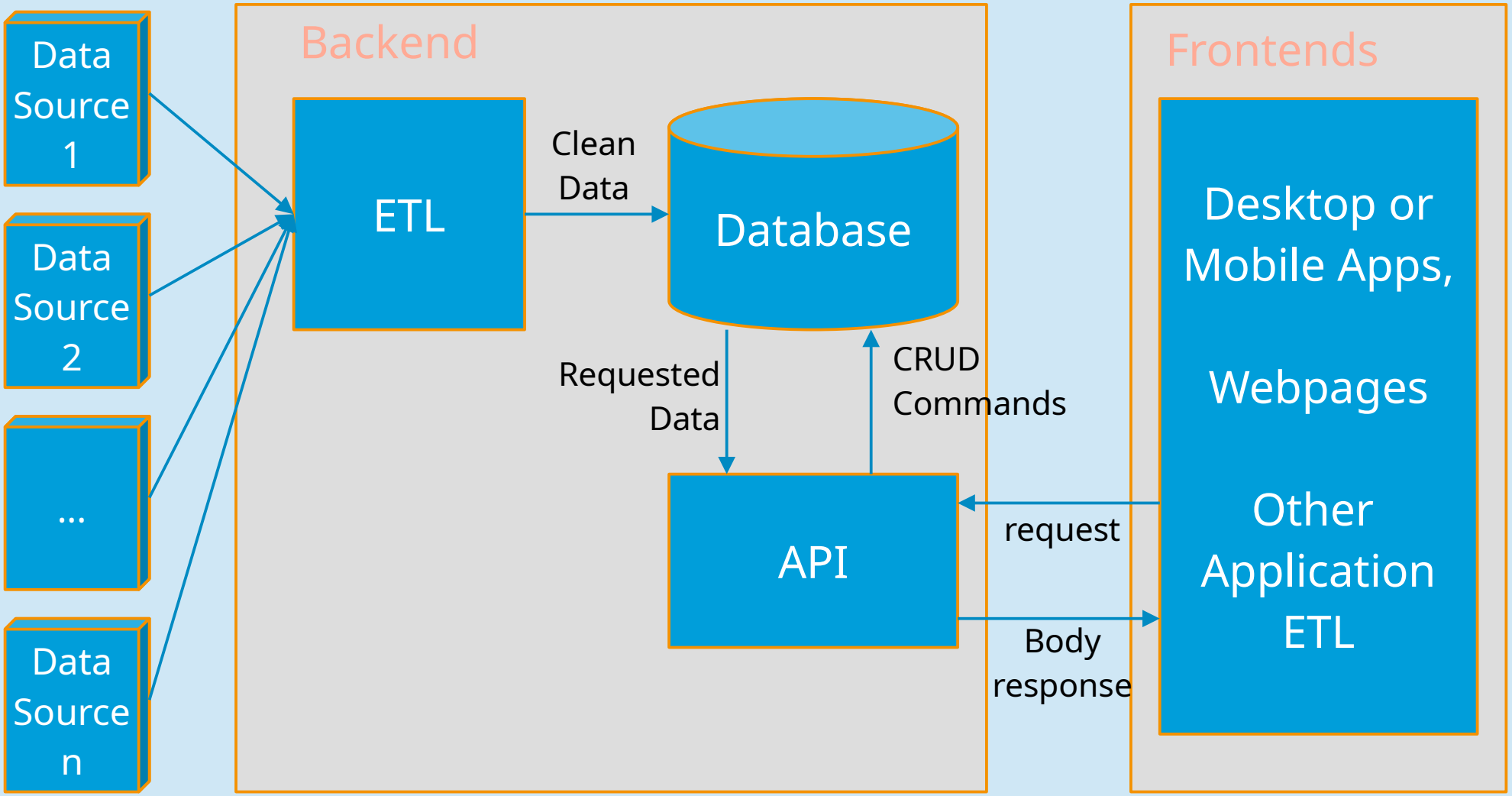
- **API** stands for **A**pplication **P**rogramming **I**nterface.
- It is a set of defined rules that enable different applications to communicate with each other.
- In building applications, an API simplifies programming by abstracting the underlying implementation and only exposing objects or actions the developer needs.
- It also provides means for others (e.g. third party applications) to use your data, without exposing the database directly.

https://www.ibm.com/topics/api

How does an API works

1) The requesting application makes a call to the API endpoint, aka a **request**.
2) After receiving a valid request, the API makes a call to the external program or web server (*e.g.* Database).
3) The server sends a response to the API with the requested information.
4) The API transfers the data (**body response**) to the initial requesting application.

ETL + CRUD + API


Backend
Frontends
Data Source 1
Data Source 2
...
Data Source n
ETL
Clean Data
Database
Requested Data
CRUD Commands
API
request
Body response
Desktop or Mobile Apps,
Webpages
Other Application ETL

API protocols

- **SOAP** (**S**imple **O**bjects **A**ccess **P**rotocol). It is an API protocol which employs XML to enable API communication. It is the oldest API protocol in use, emerging in 1998. SOAP uses XML files to transfer data between web services.
- **REST** (**Re**presentational **S**tate **T**ransfer) REST protocols overcome SOAP's dependency on XML by supporting data transmission in multiple formats such as JSON (most prominent), HTML, Python, plain text as well as media files.
- **GraphQL:** GraphQL stands for Graph Query Language, and so like database query languages like SQL, GraphQL essentially queries data from the server. One request can fetch data from several datasources or tables.
- **XML-RPC** (XML-Remote Procedure Call): The XML-RPC protocol relies on a specific XML format to transfer data
- **JSON-RPC**: Like XML-RPC, JSON-RPC is a remote procedure call, but JSON (JavaScript Object Notation) is used instead of XML to transfer the data.

- Many others ...

REST API requests

This request is made using a **Uniform Resource Identifier** (**URI**), which includes a request **verb**, **headers**, and sometimes, a **request body**.

- The **method** (**verb**) is the type of request you send to the server. The four main resource methods that are associated with REST APIs are:
  - **GET**: This method allows for the server to find the data you requested and sends it back to you.
  - **PUT/PATCH**: If you perform the 'PUT' or a 'PATCH' request, then the server will update an entry in the database.
  - **POST**: This method permits the server to create a new entry in the database.
  - **DELETE**: This method allows the server to delete an entry in the database.
  - CRUD = (CREATE, READ, UPDATE, DELETE) = (POST, GET, PUT/PATCH, DELETE)

REST API requests

- **API headers** are like an extra source of information for each API call you make. Their job is to represent the meta-data associated with an API request and response (*e.g.* authentication headers).
- **API Request Body** is data sent by the client to your API. In methods like **PUT, PATCH** and **POST**, the client needs to send data do the API (*e.g.* adding new rows to the database).

REST API response

An API response consists of the following parts

- **HTTP status code**: a 3-digit value that indicates the outcome of the operation.
- **HTTP headers** provide additional information about the response.
- **Response Body** describes or contains the result of the requested action.

HTTP status codes

Category	Numeric range	Description
Informational	100-199	1xx codes indicate some type of acknowledgement. You'll rarely see HTTP status codes in the 1xx range.
Successful	200-299	2xx codes indicate success.
Redirects	300-399	3xx codes indicate some type of redirection. You'll rarely see HTTP status codes in the 3xx range.
Client errors	400-499	4xx codes indicate an error due to a problem with the request.
Server errors	500-599	5xx codes indicate an error due to a problem with the server.

APIs

Huge list of Public APIs

- https://github.com/public-apis/public-apis

Interesting APIs for geospatial

- https://nominatim.org/
- https://openrouteservice.org
- https://overpass-turbo.eu/