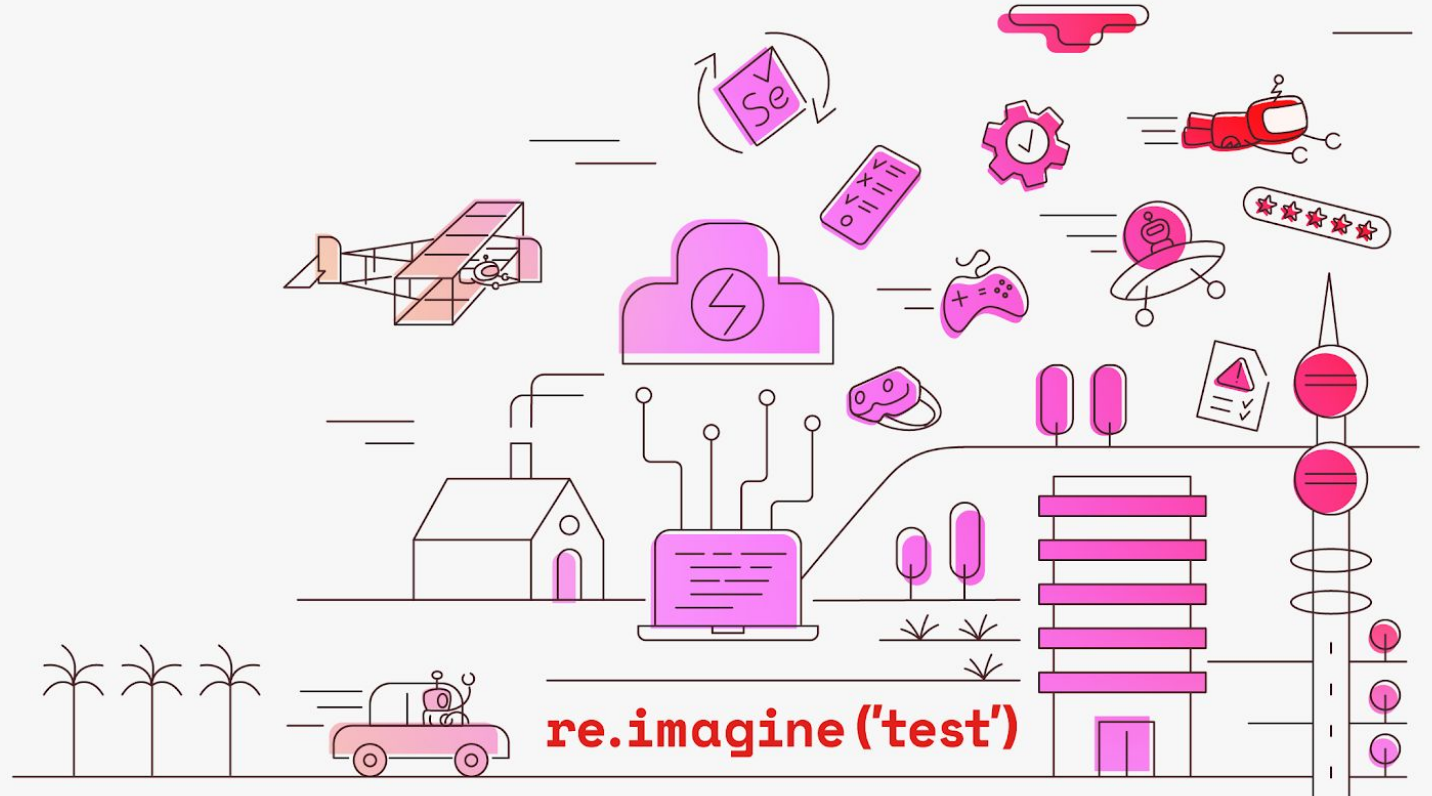
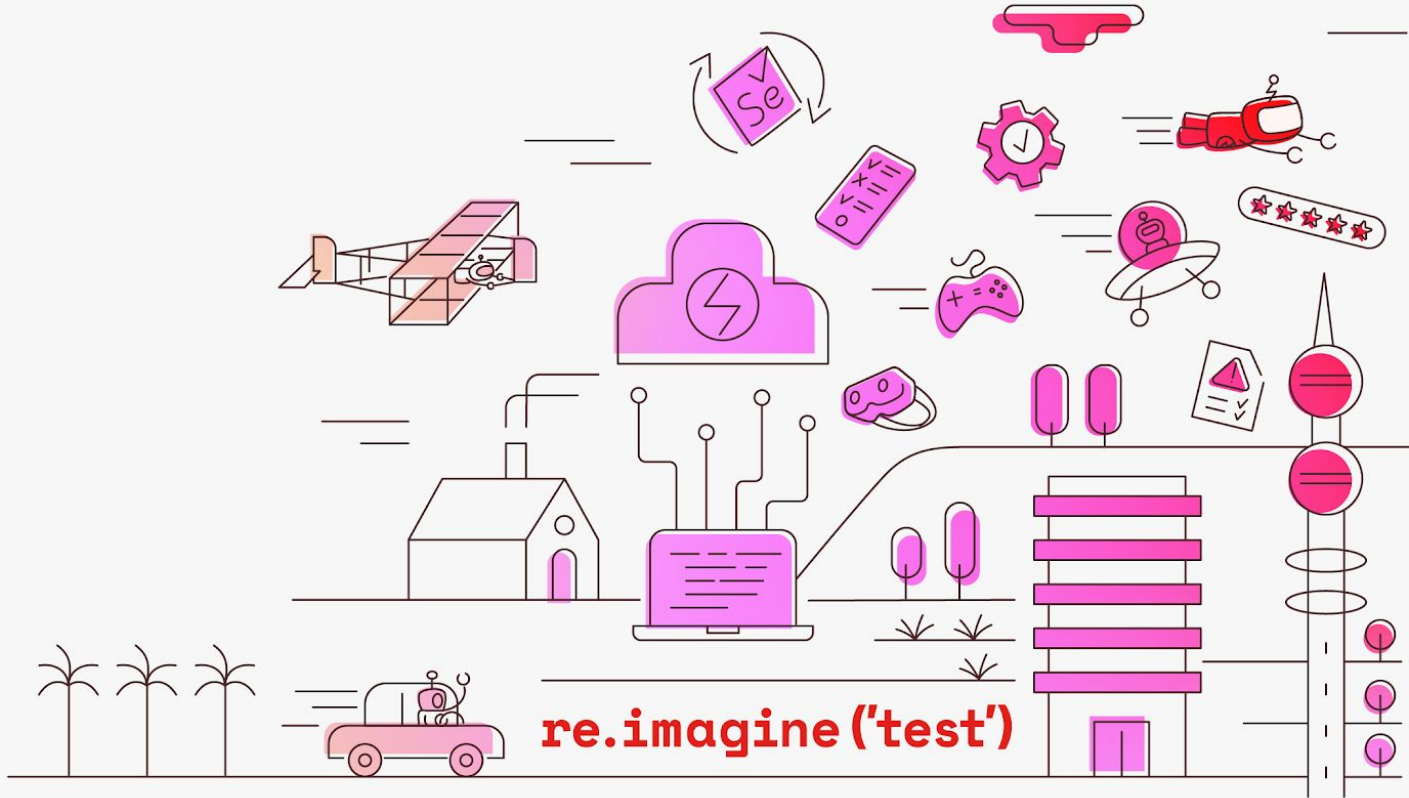


Integrating contract tests in a continuous integration pipeline

Julian Vargas | Engineering Manager API Testing

May 3, 2022

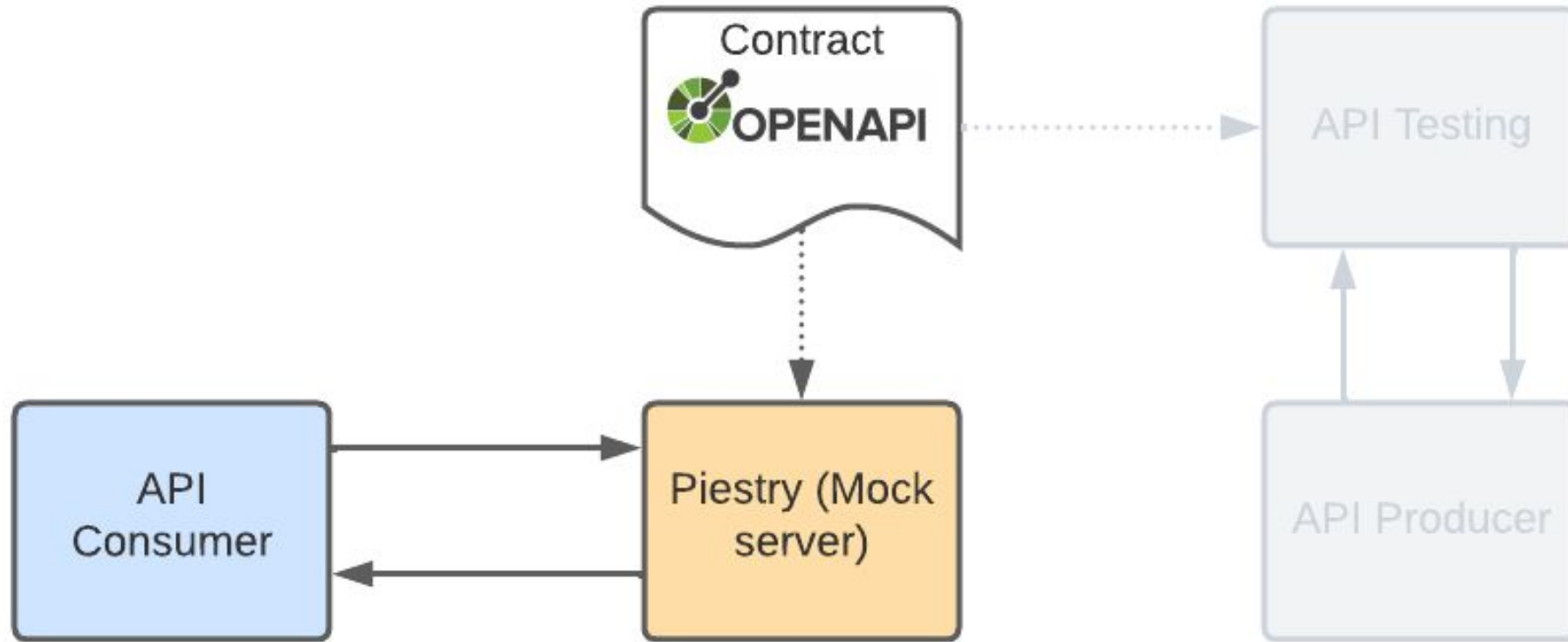




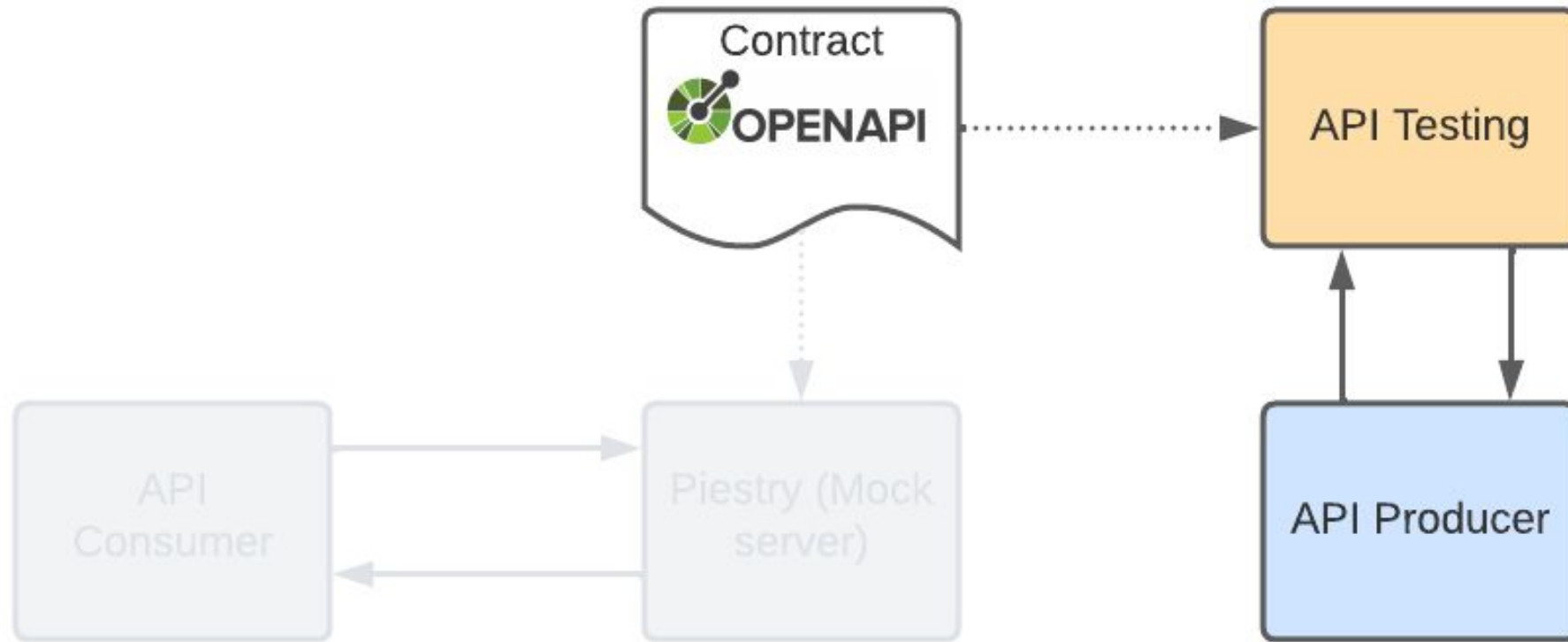
Overview - Contract Testing



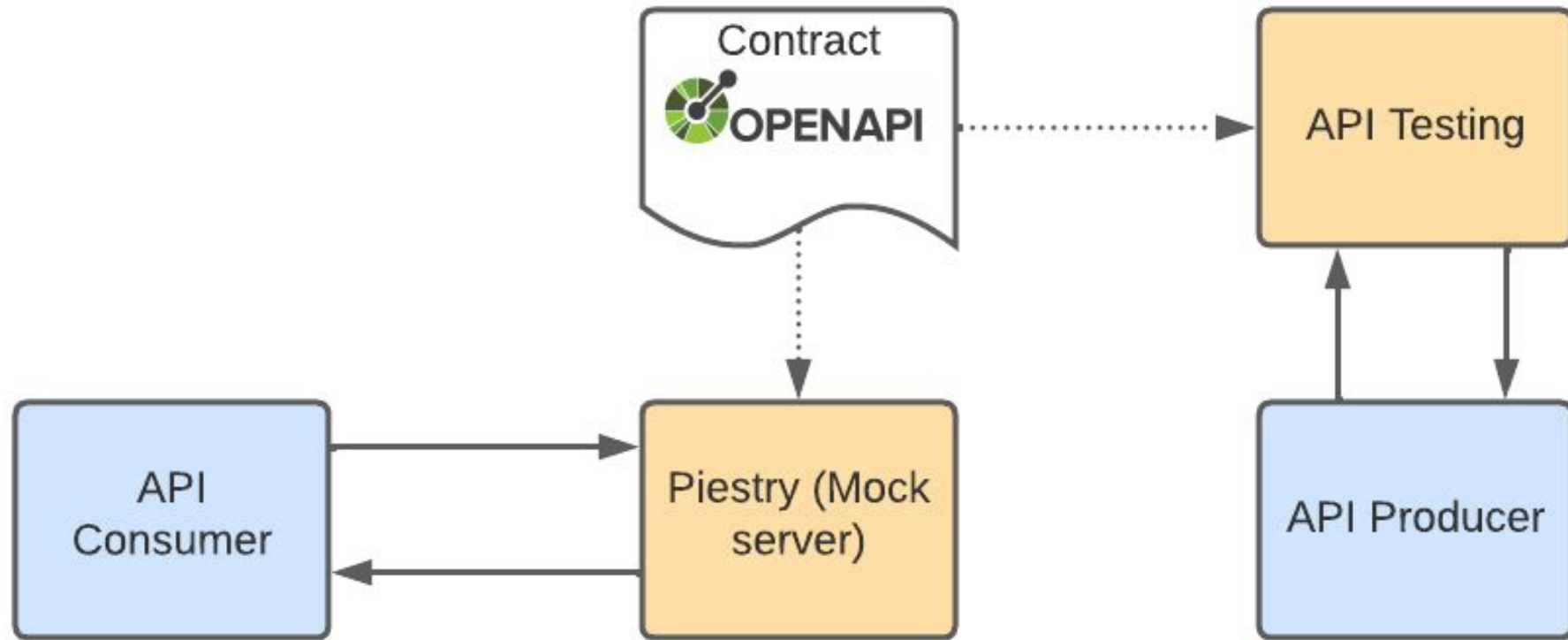
Implementing the consumer side



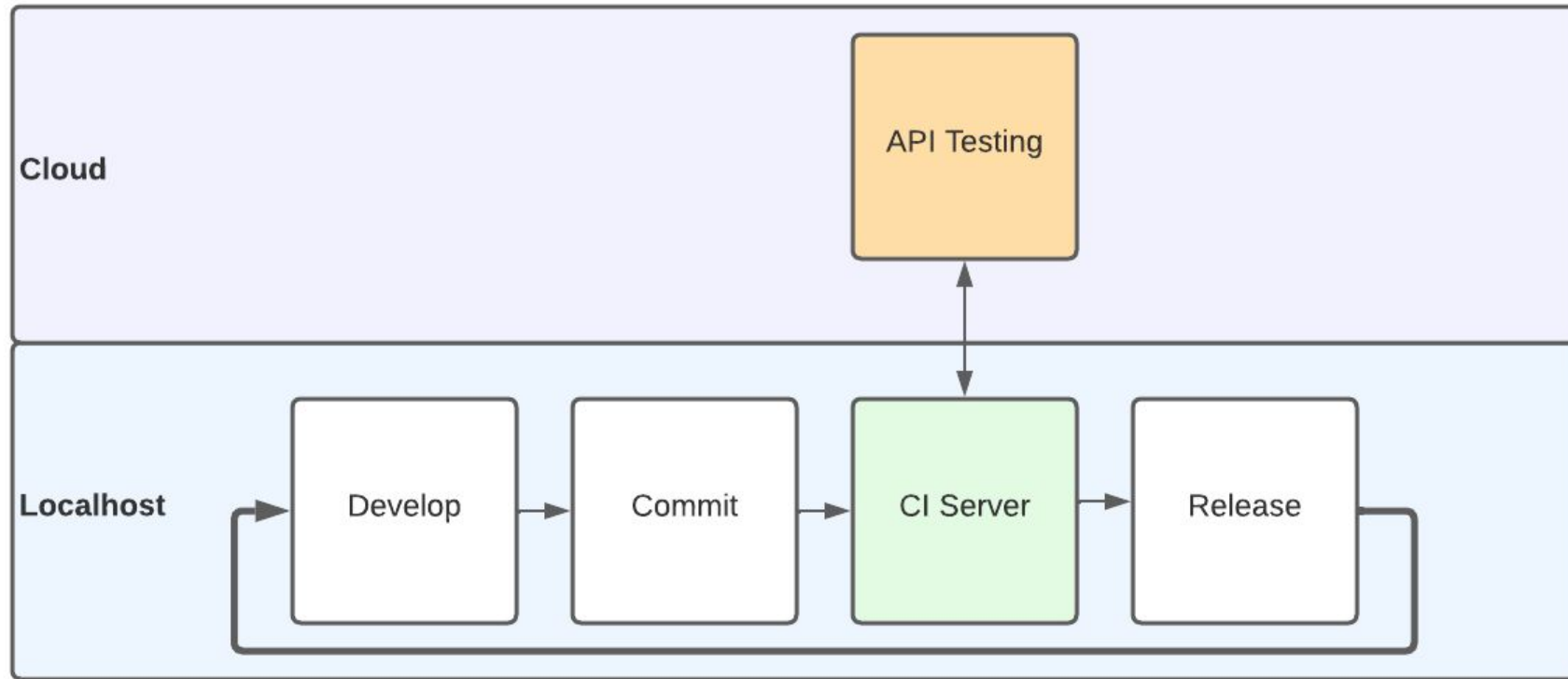
Implementing the producer side



Overview - Contract Testing



Overview - Continuous Integration



We will need

1 An active SauceLabs account

2 SauceConnect

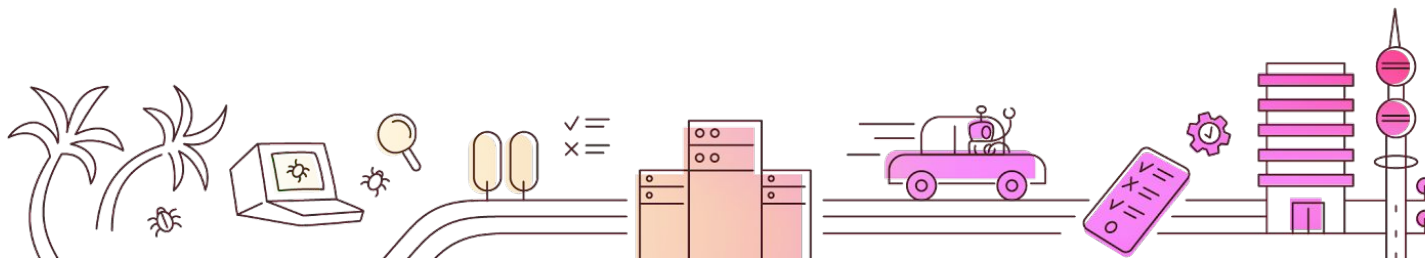
We need to expose our localhost to the cloud.

3 Deno

The example application will be written in Typescript running on Deno

4 Git SCM

For version control and for setting up our CI Server

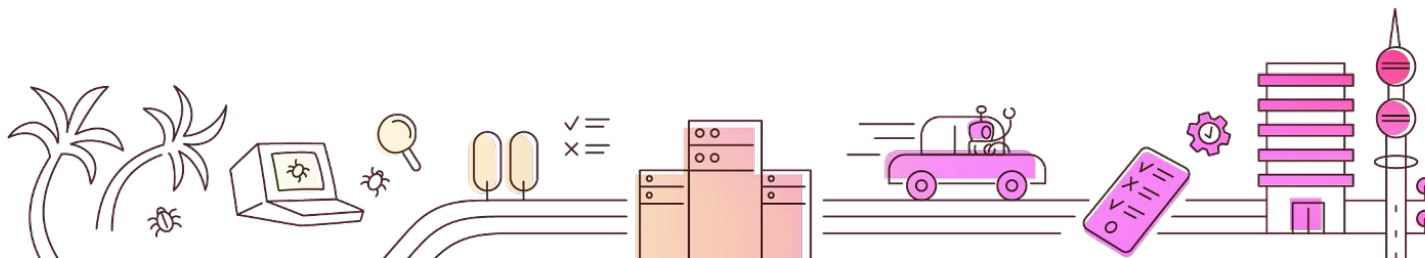


User story

As a user, I want to be able to know which is the best sauce to use for a given food so that I can enjoy my food in the best possible way.

Acceptance criteria

- command-line application that will receive a food name and will print in the terminal the name of the best sauce for that given food. For example `$ sauce_for 'fries'` will print **“ketchup”**
- This application will consume an HTTP API with one endpoint `/sauce` with a *mandatory query* parameter `?_for` that will contain the name of the food.
- The API will return a JSON document with *only one property* `sauce_name` containing the name of the best sauce for the given food, for example `{sauce_name: "ketchup"}`.
- The **Server** will expose the endpoint `http://localhost:8000/sauce/` and for any GET request received it will use the query string parameter `_for=` to read the food and respond with the best sauce.



Folder structure

```
git clone git@github.com:julianvargasalvarez/sauce-recipe.git
```

sauce-recipe

├── sauce-client

│ ├── sauce-contract.yml

│ ├── sauce.test.ts

│ └── sauce.ts

├── sauce-server

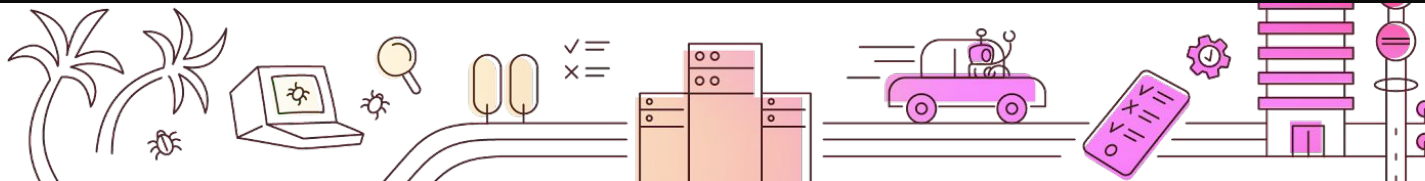
│ ├── sauce-server.ts

│ ├── sauce.test.ts

│ ├── sauce.ts

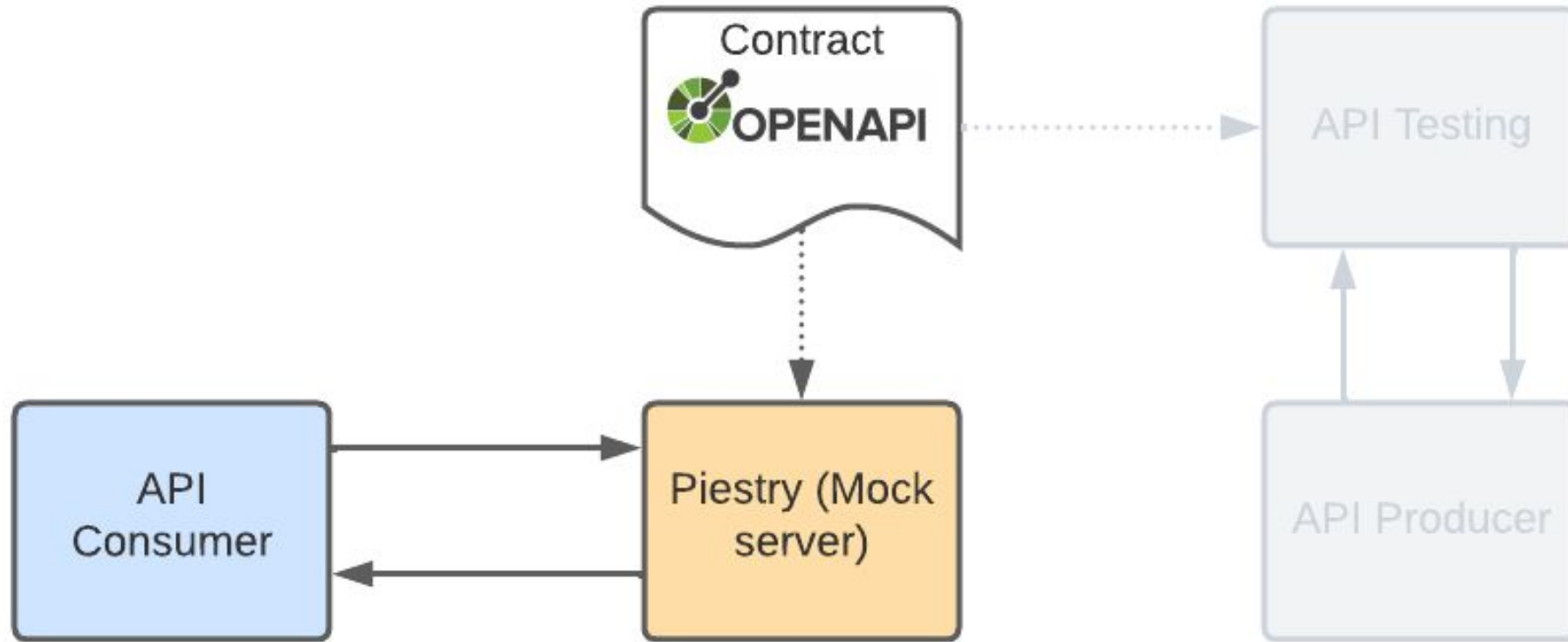
│ └── tunnel.yml

└── test-suite.sh



Let's start with the consumer side

Implementing the consumer side



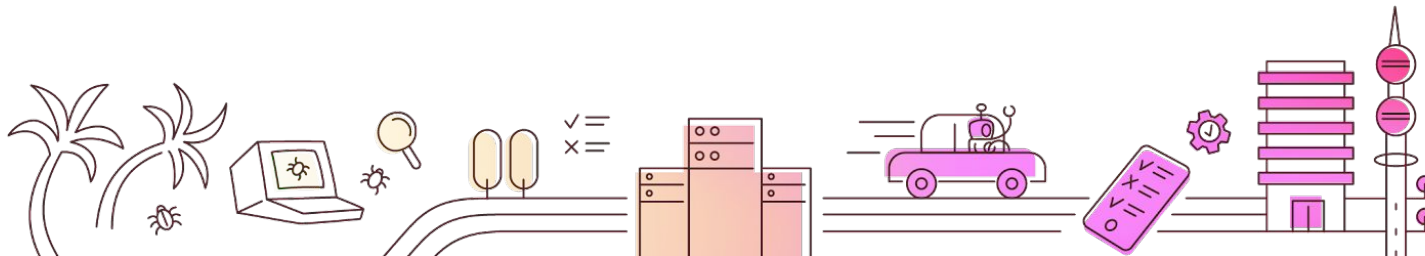
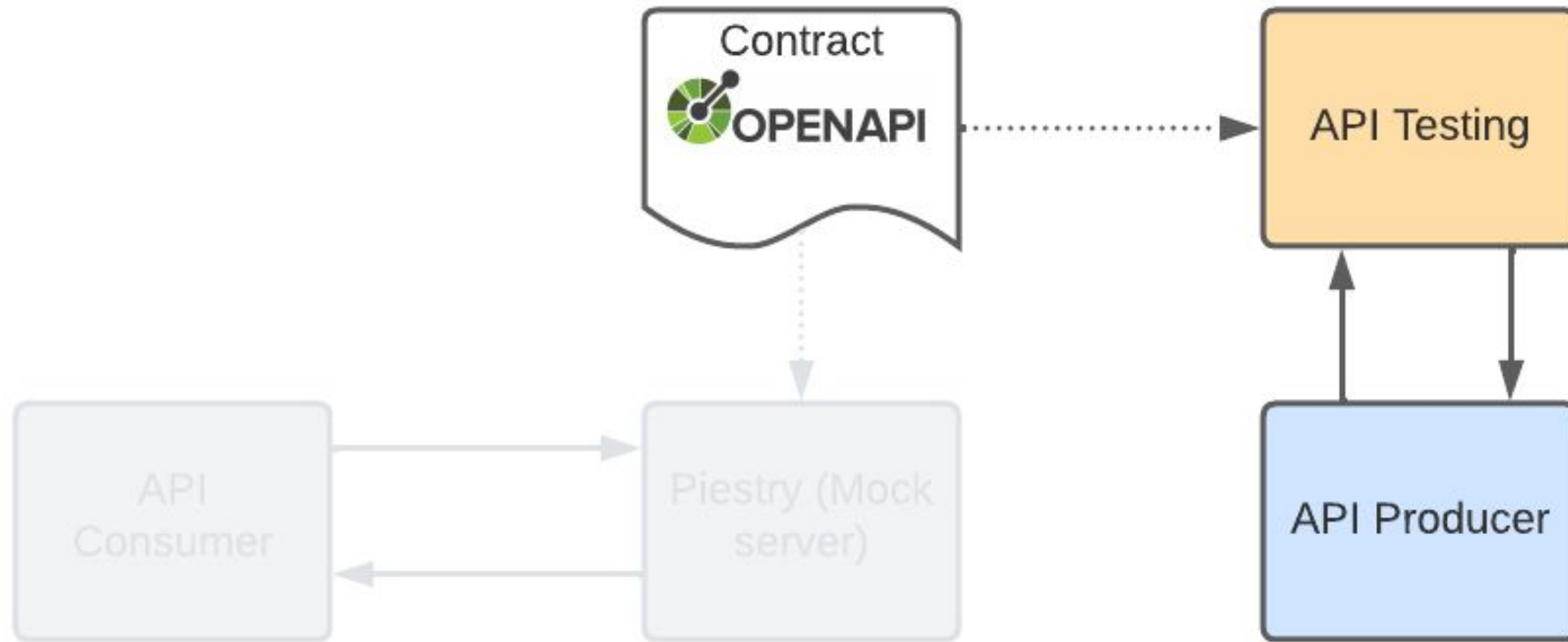
What have we accomplished so far?

- 1 We have implemented a client application emulating the server side using Piestry
- 2 We have implemented a contract from the consumer side point of view.
- 3 We have integrated contract testing from the consumer side in a Continuous Integration pipeline



Now the producer side

Implementing the producer side



What have we accomplished so far?

- 1 We have implemented the producer side
- 2 We have tested the contract from the producer side integrating SauceLabs API Testing
- 3 We have a fully integrated Contract Testing workflow within our Continuous Integration pipeline

