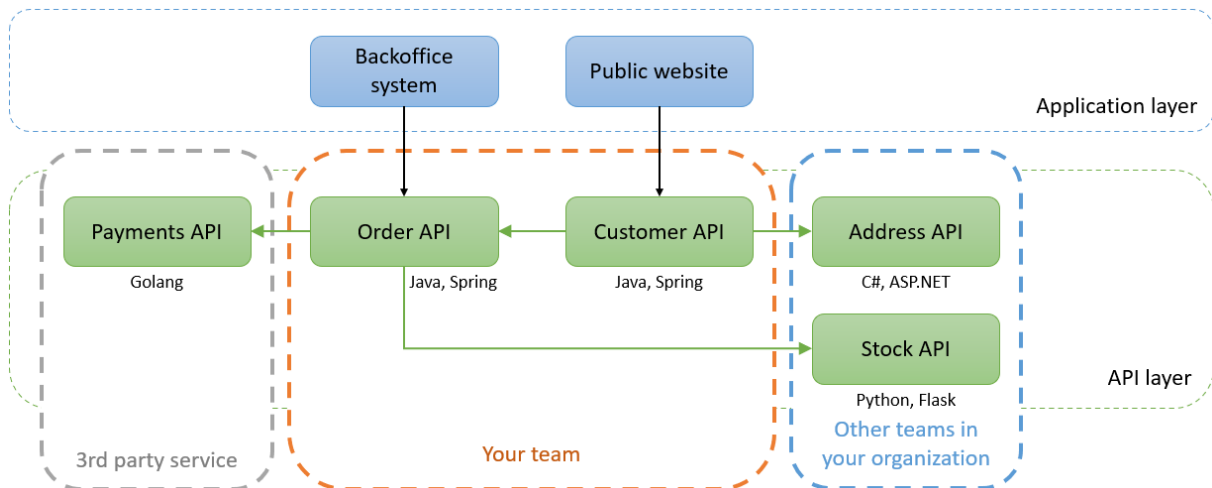# Case: developing a contract testing strategy

Your organization

The organization you're working for is a market leader in online selling and delivering of furniture.

The web application that serves the website and handles all orders that come through is built on a microservices-based architecture, part of which is shown here:

The APIs and their relations

- The customer API (built in Java using Spring Boot) is responsible for managing all data related to customers. It is consumed by the public website, where customers can see their data and their orders.
- Order data is managed by the Order API (also a Java Spring Boot service), which is consumed by the customer API so that all orders for a specific customer can be retrieved.
- Addresses are managed in a separate Address API (built in C# using ASP.NET), mainly for efficiency reasons.
- To make sure that orders can be delivered, the Order API consumes a Stock API (written in Python using Flask) to retrieve the current stock for any of the items on sale.
- To facilitate payments, a public payments service is used (think Stripe, Mollie, etc.). This service is written in Go.
- There's also a backend system (based on Node.JS) for store employees to manage orders, this directly consumes the Order API.

Your assignment

You are tasked with creating a contract testing strategy for this product and for your team. This means you'll have to address (among others) the following questions:

- Would contract testing be helpful in this situation? Why (not)?
- What type of contract testing would be most useful for each of the integrations? Why?
- What would you recommend in terms of a strategy for adoption of contract testing? Where would you start to see if it is bringing your development process any value?

Are there any other aspects that you think make a vital part of the contract testing strategy? Feel free to list those as well!

After you have compiled the answers to these questions, please prepare to briefly (~ 5 minutes) present your strategy to the other teams.