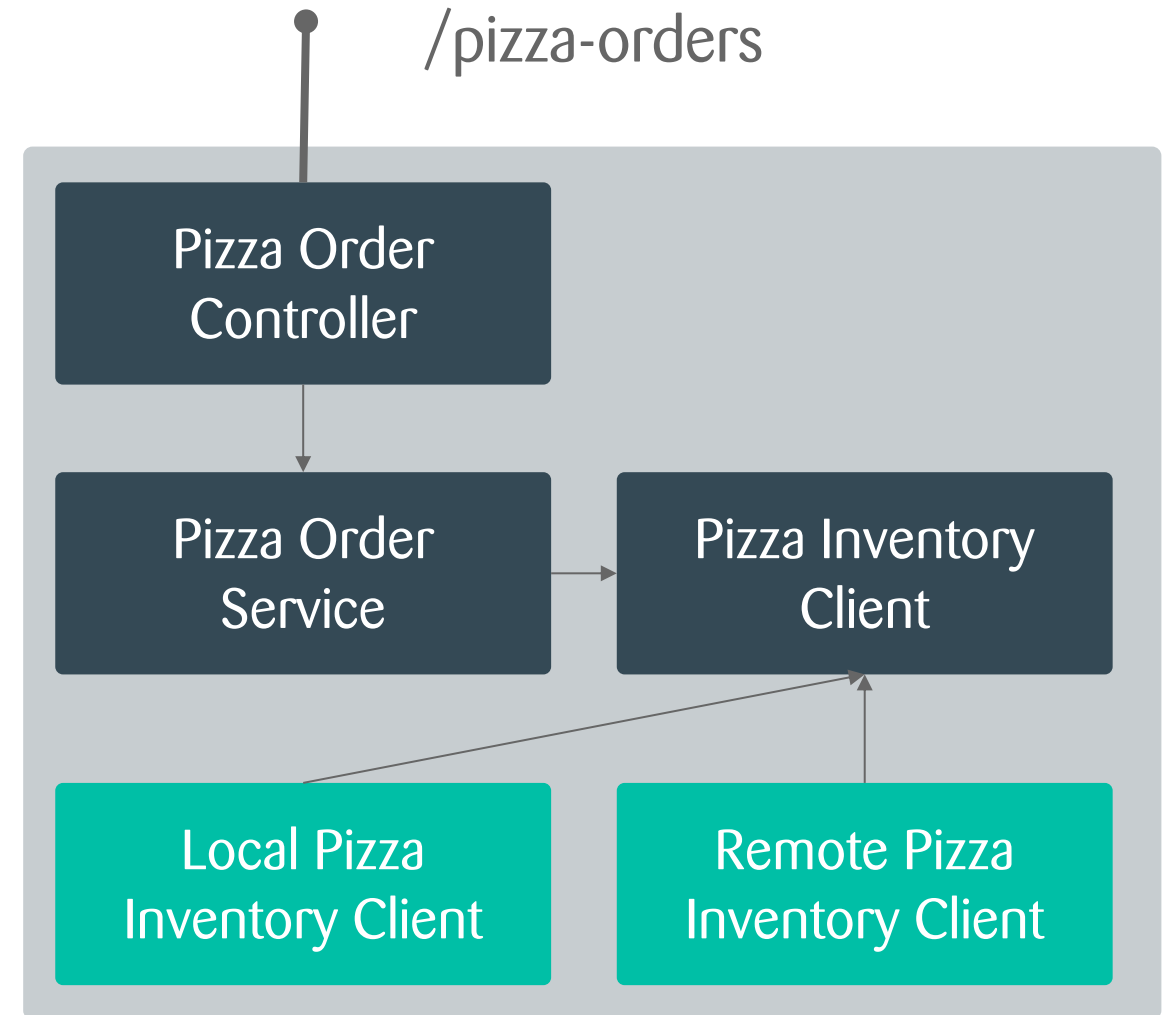# Spring Boot / Hands-On 2

- Introduce a new interface "Pizza Inventory Client" with two implementations Local- and Remote Pizza Inventory Client. Find a mechanism to load them conditionally.

- The pizza inventory client has one method *boolean isAvailable(String pizzaName)*

- The local implementation always returns true

- The remote implementation should return randomly true or false.

/pizza-orders

```
┌─────────────────────────────────────────────┐
│  ┌──────────────────┐                        │
│  │  Pizza Order     │                        │
│  │  Controller      │                        │
│  └──────────────────┘                        │
│                                              │
│  ┌──────────────────┐   ┌──────────────────┐ │
│  │  Pizza Order     │──▶│  Pizza Inventory │ │
│  │  Service         │   │  Client          │ │
│  └──────────────────┘   └──────────────────┘ │
│                                              │
│  ┌──────────────────┐   ┌──────────────────┐ │
│  │  Local Pizza     │   │  Remote Pizza    │ │
│  │  Inventory Client│   │  Inventory Client│ │
│  └──────────────────┘   └──────────────────┘ │
└─────────────────────────────────────────────┘
```

Spring Boot / Hands-On 2

- The pizza order service should inject the new pizza inventory client and in case of a new pizza order it should call the isAvailable method for each ordered pizza. And if one pizza is not available it should throw an IllegalStateException.

- In case of an IllegalArgumentException the application should return the HTTP status code 404 NOT FOUND.

- In case of an IllegalStateException the application should return the HTTP status code 400 BAD REQUEST.

/pizza-orders

Pizza Order Controller

Pizza Order Service

Pizza Inventory Client

Local Pizza Inventory Client

Remote Pizza Inventory Client