Problem statement

Imagine a new online shopping website called 'ShopNow.' They want to create a modern and reliable platform where customers can buy products from different sellers. To make this happen, they plan to build a strong and flexible system using a microservices approach.
The website will have three main parts: one for managing user accounts, one for handling product information, and one for processing customer orders.

The 'ShopNow' team aims to design an easy-to-use and consistent system. They want to follow best practices in how they set up the different parts of the website. They also want to make sure that customer information is safe and protected.

Your objective is to design a scalable, modular, and well-structured microservices-based architecture for this e-commerce platform. The platform should follow RESTful API principles, ensuring clear resource naming conventions and coherent endpoint designs.

Specifically, your task includes:

1. **Microservices Architecture:**
   - Design each microservice to focus on its designated responsibilities, adhering to a loosely coupled architecture for better maintainability.
   - Ensure the microservices can communicate efficiently and securely with each other.
2. **RESTful API Design:**
   - Create the API endpoints for each service with intuitive and meaningful resource naming.
   - Define clear request and response formats, following RESTful principles.
3. **Software Design Patterns:**
   - Mention appropriate software design patterns to promote code reusability and maintainability.
4. **Error Handling:**
   - Design robust error handling mechanisms and provide informative error messages to assist users and developers.
5. **Data Storage:**
   - Choose appropriate data storage mechanisms for each service (e.g., SQL, NoSQL) to handle user accounts, product information, and order data.
6. **Authentication and Authorization**
   - Explain how to implement secure authentication and authorization mechanisms.

Non-functional requirements

- 100+ million transactions per day
- Response time <= 300ms
- 99,99% availability
- Auto-scaling
- DRP (multi-region)
- Duration: The Product Owner is requiring this component in 4 sprints