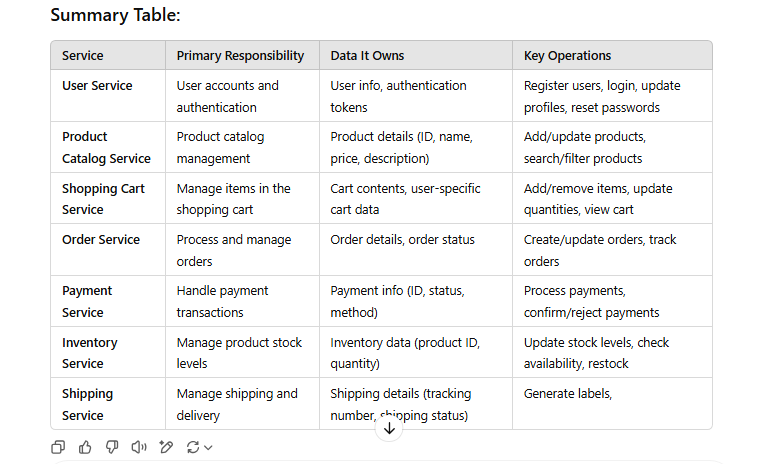
# **Detyra 1: Identifikimi i Shërbimeve**



# Detyra 2: Ndërveprimet e Shërbimeve

**Si ta bëjmë**: Ne duhet të sigurohemi që shërbimet mund të **komunikojnë** me njëri-tjetrin kur është e nevojshme. Ka dy mënyra kryesore për të komunikuar:

1. **Komunikimi Synchronized** (si të bisedosh me dikë dhe të presësh një përgjigje menjëherë):
   * Shembull: Kur vendos një porosi, **Shërbimi i Porosisë** mund t’i kërkojë **Shërbimit të Pagesave** të përpunojë pagesën. Porosia nuk mund të vazhdojë pa një përgjigje nga pagesa.
   * Mjete: API REST (si kërkesat HTTP), gRPC.
2. **Komunikimi Asinkron** (si të dërgosh një letër dhe të presësh një përgjigje më vonë):
   * Shembull: **Shërbimi i Njoftimeve** mund të dërgojë një email pasi të vendoset një porosi, por nuk ka nevojë të presë një përgjigje për të vazhduar.
   * Mjete: Rreshtat e mesazheve, autobusë ngjarjesh.

### **Si mund të duken rrugët kryesore të përdoruesve?**

Të themi se përdoruesi vendos një porosi, ja se si mund të ndodhin gjërat:

* **Përdoruesi vendos porosinë:.**
* **Kontrollo disponueshmërinë e produktit:**
* **Paga e porosisë:.**
* **Transportimi i porosisë:**
* **Njoftimi i përdoruesit:.**

### **Pikat ku mund të ndodhin problemet:**

* **Ngadalësimi i shërbimeve:**
* **Ndërprerjet e shërbimeve:**
* **Shkarkimet e ngjarjeve:**
* **Ngarkesa në databazë:.**

### **Ngarkesa në Databazë**

**Problemi:** Kur shumë përdorues kërkojnë informacion të njëjtë (si një produkt i veçantë), databaza mund të ngarkohet dhe kjo mund të ngadalësojë përgjigjet për të gjithë sistemin.

**Zgjidhjet:**

* \*\*Përdorimi i **Caching**: Siç u përmend më parë, caching mund të ruajë përkohësisht rezultatet e kërkesave të zakonshme në një memorie të shpejtë, duke zvogëluar ngarkesën e databazës.
  + P.sh., kur një përdorues kërkon informacion për një produkt, mund të ruhet një kopje në cache për një periudhë kohore të caktuar.

# Detyra 3: Diagrami i Arkitekturës

