Microservices

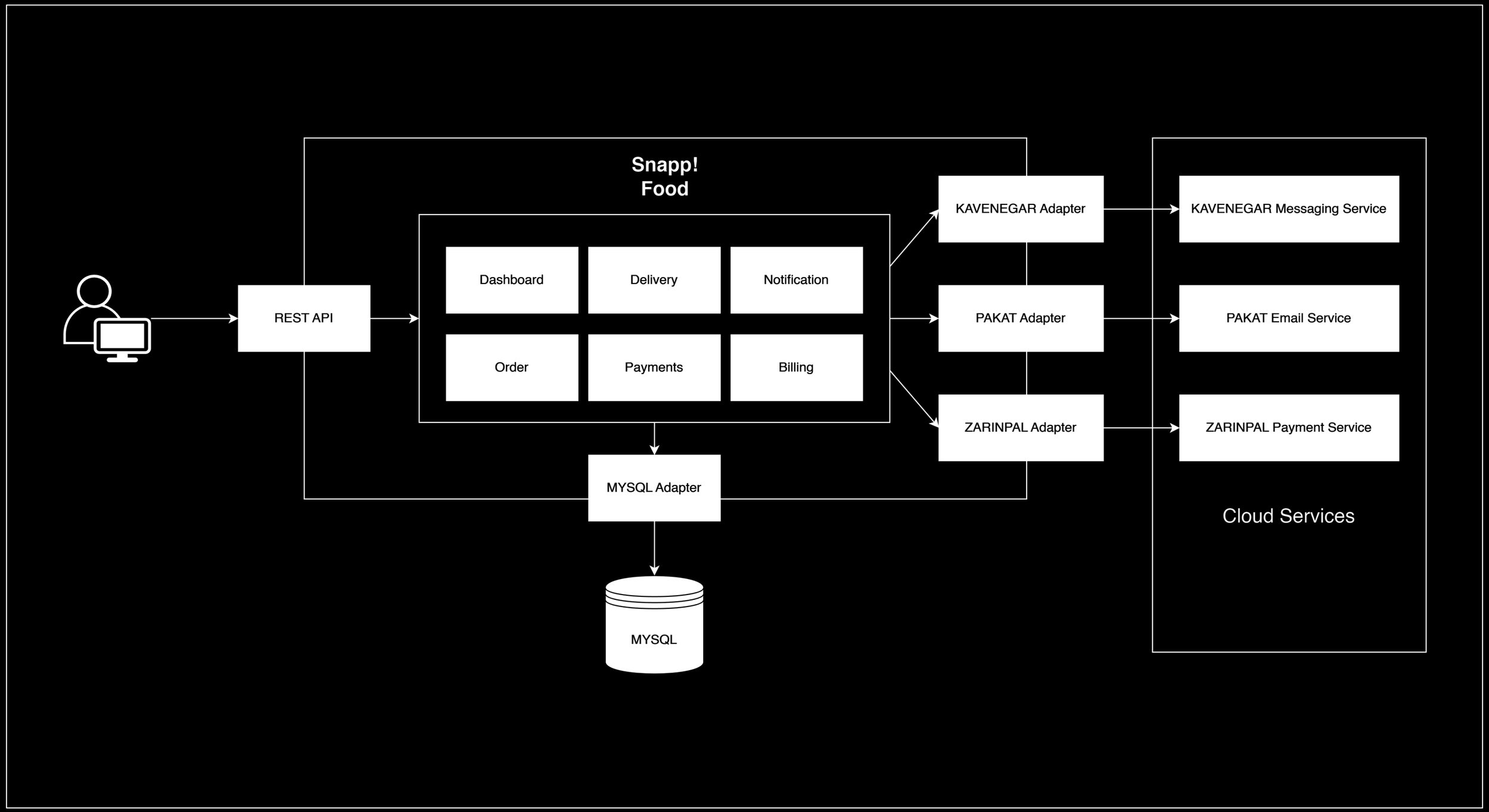
Systems Analysis & Design

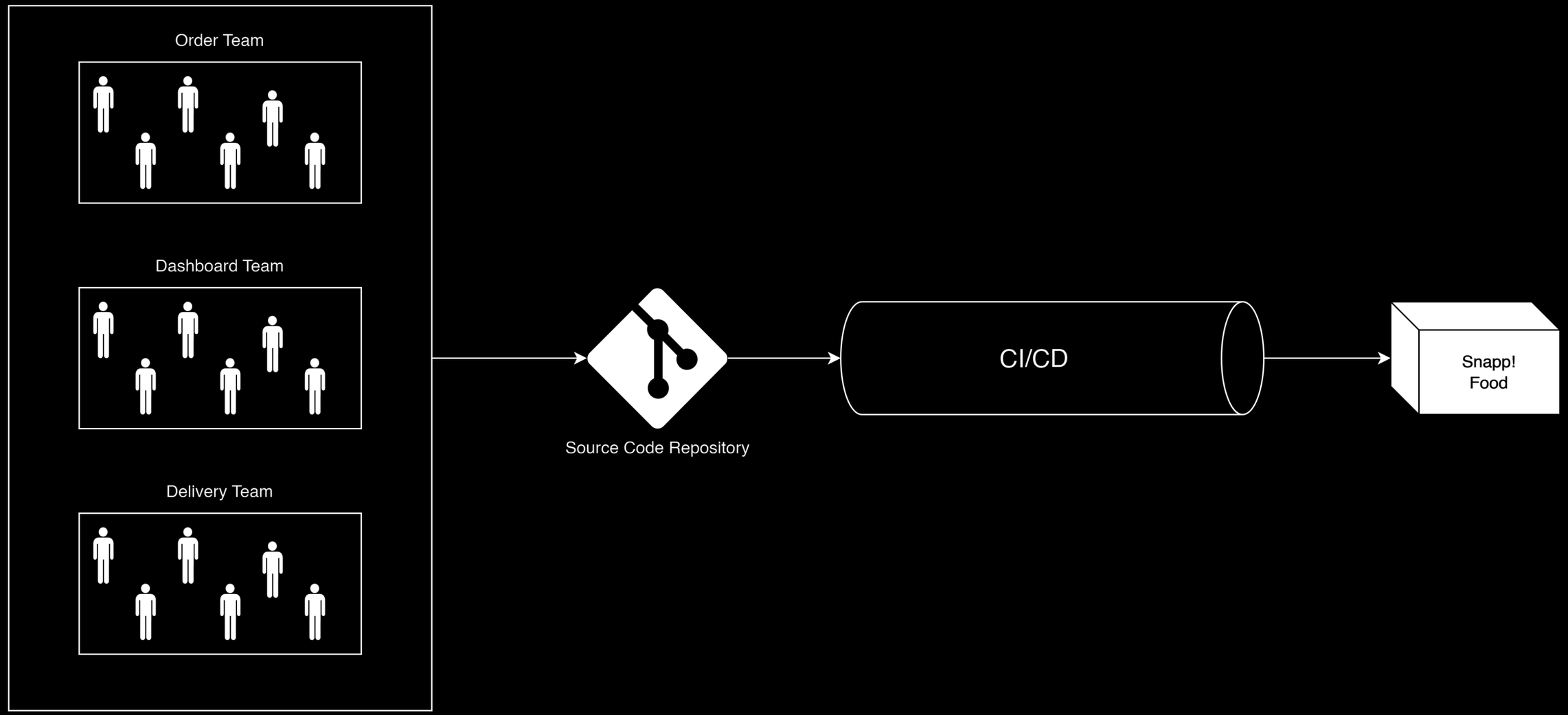
Learning Objectives

By the end of this session, you will have acquired the following information:

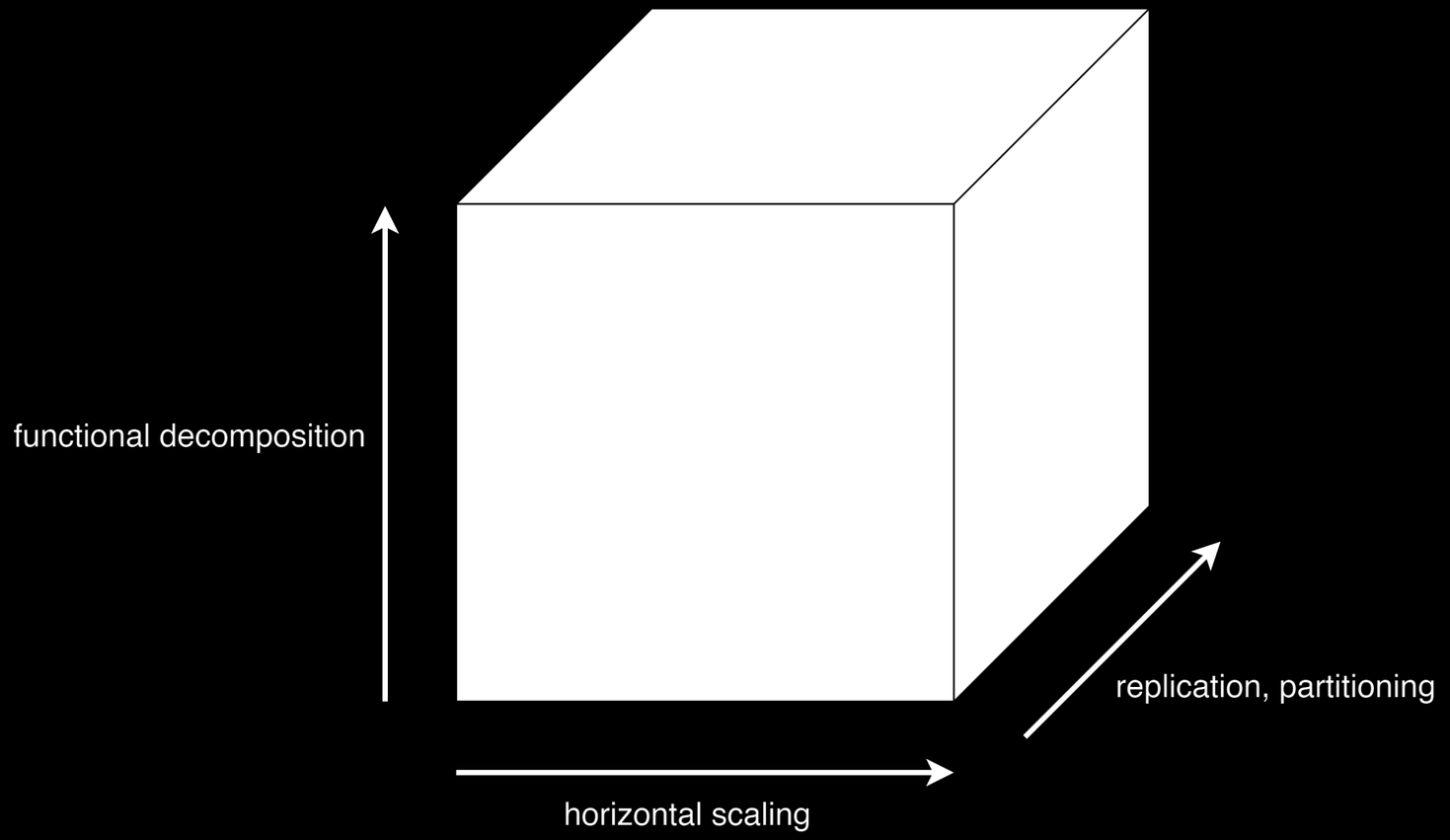
- Monolithic Architecture
- Scale Cube
- Microservice Architecture

Monolith


Snapp!
Food
REST API
Dashboard
Delivery
Notification
Order
Payments
Billing
MYSQL Adapter
MYSQL
KAVENEGAR Adapter
PAKAT Adapter
ZARINPAL Adapter
KAVENEGAR Messaging Service
PAKAT Email Service
ZARINPAL Payment Service
Cloud Services

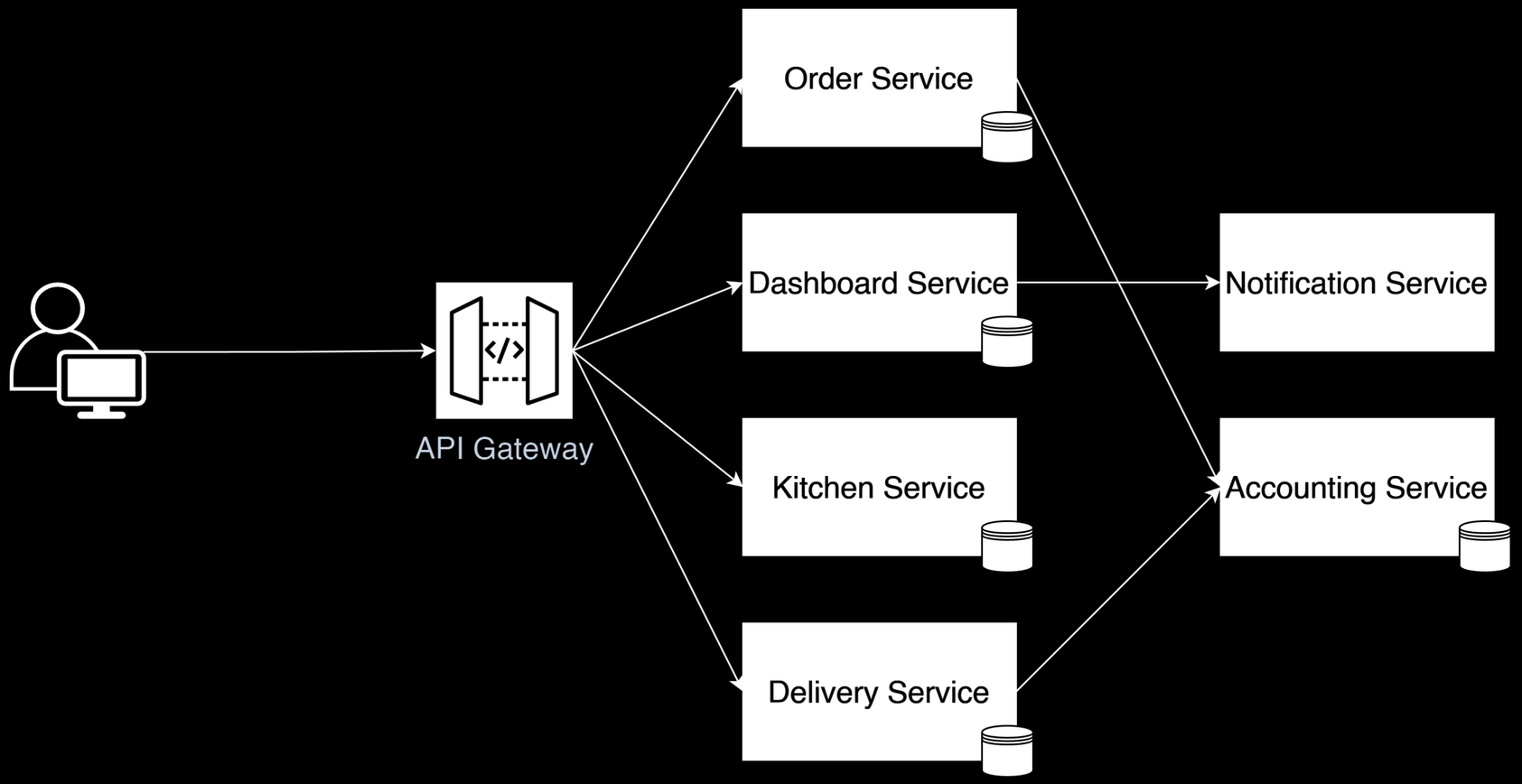

Order Team
Dashboard Team
Delivery Team
Source Code Repository
CI/CD
Snapp!
Food

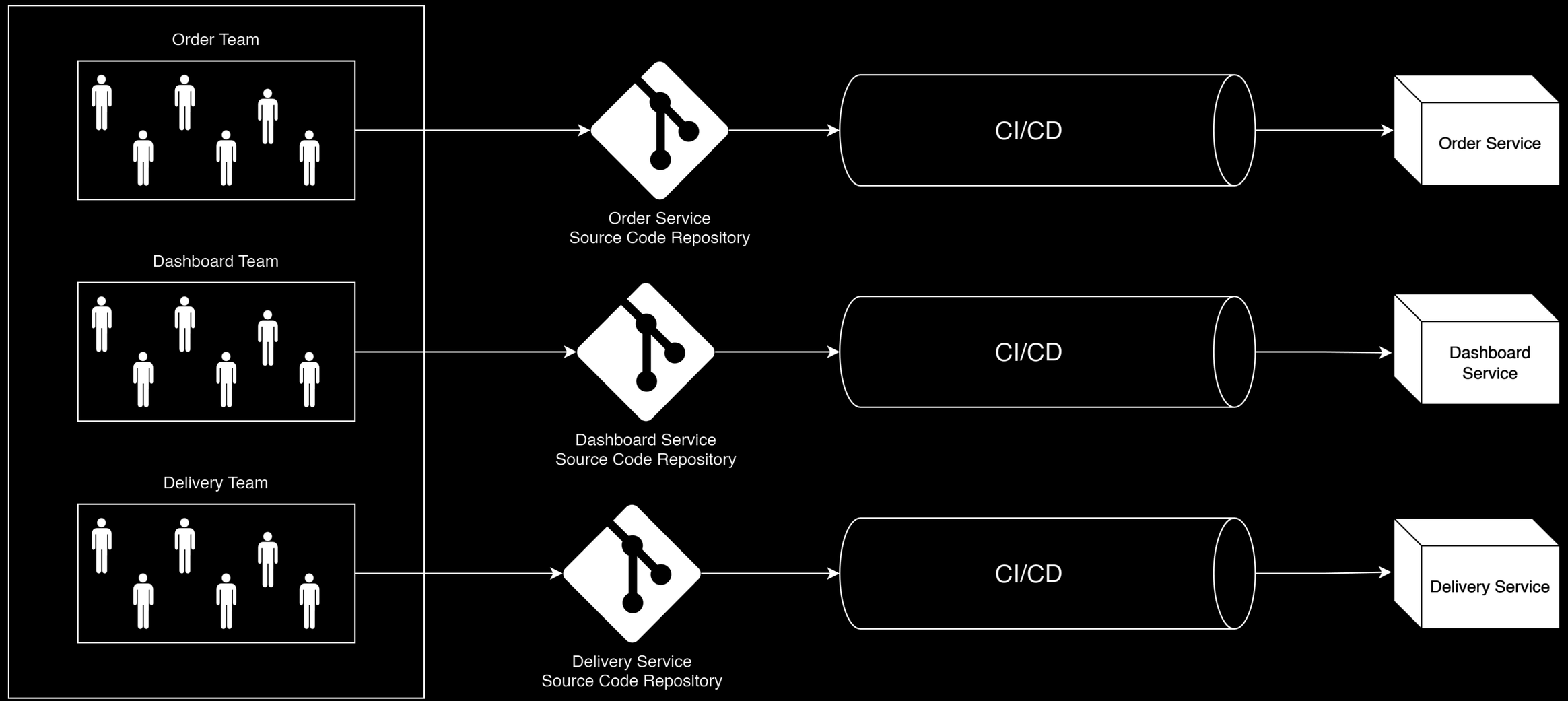
Scale Cube


functional decomposition
replication, partitioning
horizontal scaling

Microservices

- Microservice architecture develops an application as a suite of small services.
- Each service runs in its own process and communicates with lightweight mechanisms.
- Services are built around business capabilities and can be independently deployed.
- Minimal centralized management is required.
- Services can use different programming languages and data storage technologies.


Order Service
Dashboard Service
Notification Service
API Gateway
Kitchen Service
Accounting Service
Delivery Service

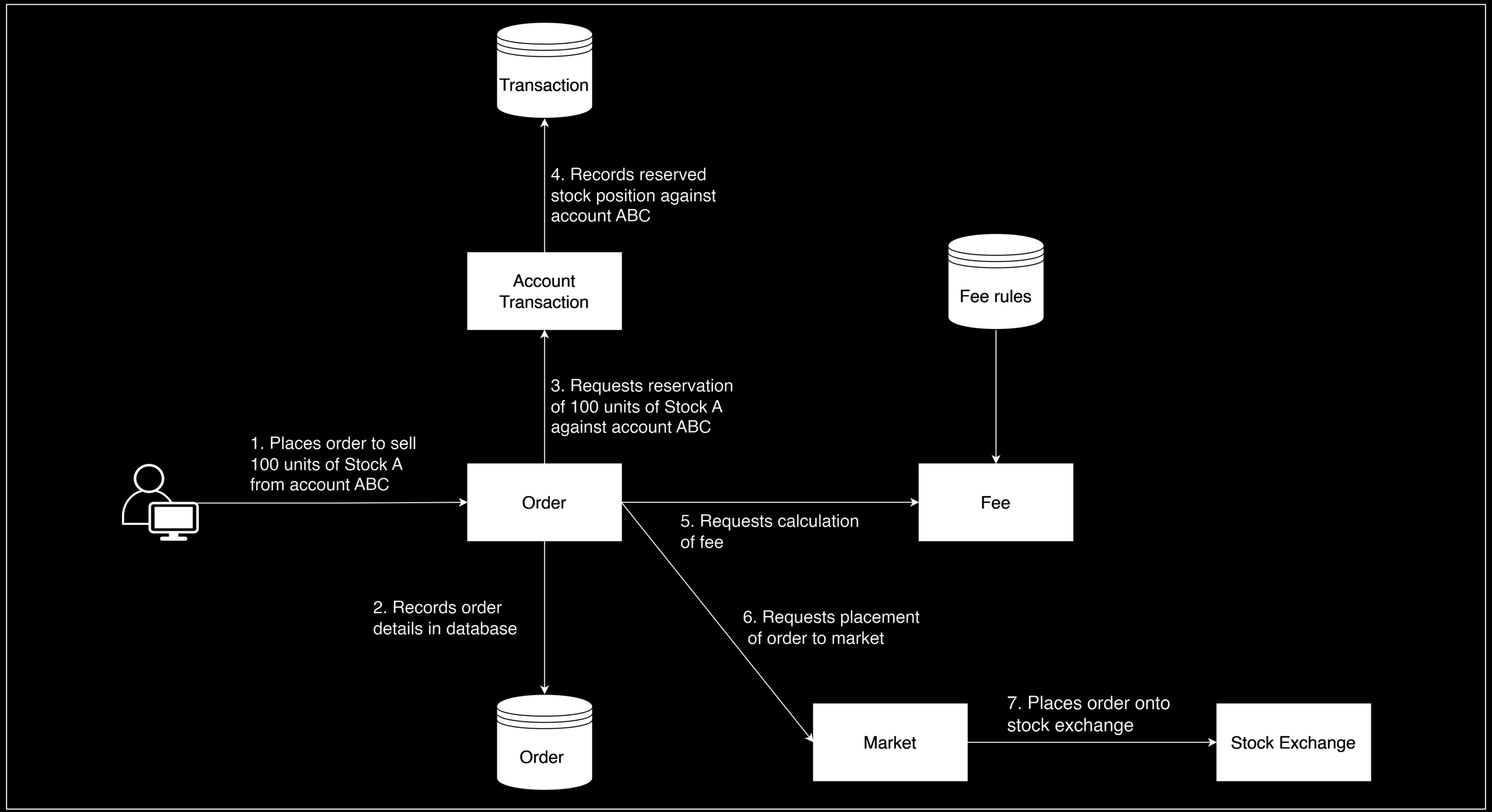

Order Team
Dashboard Team
Delivery Team
Order Service
Source Code Repository
Dashboard Service
Source Code Repository
Delivery Service
Source Code Repository
CI/CD
CI/CD
CI/CD
Order Service
Dashboard
Service
Delivery Service

Order Service APIs

- GET /orders
- POST /orders
- POST /orders/8/pay
- POST /orders/8/cancel
- GET /orders/8
- PUT /orders/8
- PATCH /orders/8
- DELETE /orders/8

```yaml
/orders/{order_id}:
  parameters:
    - in: path
      name: order_id
      required: true
      schema:
        type: string
        format: uuid
  get:
    summary: Returns the details of a specific order
    operationId: getOrder
    responses:
      '200':
        description: OK
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/GetOrderSchema'
      '404':
        $ref: '#/components/responses/NotFound'
```

Illustrative Example


Transaction
4. Records reserved stock position against account ABC
Account Transaction
Fee rules
3. Requests reservation of 100 units of Stock A against account ABC
1. Places order to sell 100 units of Stock A from account ABC
Order
Fee
5. Requests calculation of fee
2. Records order details in database
6. Requests placement of order to market
Order
Market
7. Places order onto stock exchange
Stock Exchange

Further Resources

- Designing and running microservices
- How to break a Monolith into Microservices