Sử dụng kiến trúc Microservices

Lý do:

* Ngân hàng thường phục vụ lượng lớn người dùng (hàng trăm nghìn đến hàng triệu). Microservices cho phép mở rộng từng chức năng (chuyển khoản, thanh toán) mà không ảnh hưởng toàn hệ thống. Ví dụ: Nếu giao dịch tăng đột biến, chỉ cần thêm server cho Transaction Service.
* Mỗi dịch vụ có thể áp dụng bảo mật riêng (OAuth cho xác thực, mã hóa dữ liệu cho giao dịch), giảm rủi ro nếu một phần bị tấn công.
* Dễ tích hợp với hệ thống bên ngoài (thanh toán liên ngân hàng, ví điện tử) hoặc thêm tính năng mới.
* Ví dụ thực tế: Các ngân hàng số lớn như Revolut, Monzo, hoặc ứng dụng như PayPal đều dùng Microservices để xử lý tải lớn và đảm bảo dịch vụ liên tục.

1. Các thành phần (Components)

* Authentication Service (Dịch vụ xác thực)
  + Use Case: Đăng ký, Đăng nhập, Đăng xuất, Xác nhận giao dịch, Thay đổi mật khẩu
  + Chức năng: Quản lý đăng ký người dùng, xác thực (username/password, OTP), cấp token, xác nhận giao dịch qua OTP, quản lý khôi phục mật khẩu
* Account Service (Dịch vụ tài khoản)
  + Use Case: Đăng ký, Xem số dư tài khoản, Quản lý thông tin cá nhân, Mở tài khoản tiết kiệm, Tất toán tài khoản tiết kiệm
  + Chức năng: Quản lý đóng/mở tài khoản, tài khoản tiết kiệm, Quản lý thông tin tài khoản, số dư, cập nhật thông tin cá nhân
* Transaction Service
  + Use Case: Chuyển tiền, Xem lịch sử giao dịch, Rút tiền ATM.
  + Chức năng: Xử lý các giao dịch tài chính (chuyển khoản, nạp tiền, rút tiền), lưu và truy xuất lịch sử giao dịch.
* Payment Service (Dịch vụ thanh toán)
  + Use Case: Quét QR, Nạp tiền điện thoại.
  + Chức năng: Xử lý thanh toán qua mã QR (liên kết với Transaction Service để trừ tiền).
* Loan Service
  + Use Case: Đăng ký vay online, Xem trạng thái đơn vay, Thanh toán khoản vay
  + Chức năng: Quản lý đơn vay, kiểm tra điều kiện vay, gửi yêu cầu phê duyệt.
* Savings Service
  + Use Case: Mở tài khoản tiết kiệm (liên kết với Account Service).
  + Chức năng: Quản lý tài khoản tiết kiệm, tính lãi suất.
* Admin Service
  + Use Case: Quản lý tài khoản, Xem báo cáo hệ thống
  + Chức năng: Quản lý tài khoản người dùng, kiểm soát hệ thống, tạo báo cáo (dành cho admin).
* Notification Service
  + Chức năng: Gửi thông báo (email, SMS, push) khi giao dịch hoàn tất, xác nhận OTP, v.v.
* API Gateway
  + Chức năng: Điểm vào duy nhất cho các yêu cầu từ client, định tuyến đến dịch vụ phù hợp.
* Database Cluster
  + Chức năng: Lưu trữ dữ liệu riêng cho từng dịch vụ (hoặc dùng cơ sở dữ liệu chung với phân vùng).

2. Kết nối giữa các thành phần (Connections)

* Frontend Client 🡪 API Gateway:
  + Kết nối: HTTPS với REST API.
  + Mô tả: Người dùng gửi yêu cầu (như "Chuyển tiền") qua app/web, API Gateway định tuyến đến dịch vụ phù hợp.
* API Gateway 🡪 Các service:
  + Kết nối: gRPC
  + Mô tả: API Gateway gửi yêu cầu đến Authentication Service (đăng nhập), Transaction Service (chuyển tiền), v.v.
* Authentication Service 🡪 Transaction Service
  + Kết nối: gRPC.
  + Mô tả: Transaction Service gọi Authentication Service để xác nhận OTP trước khi thực hiện giao dịch.
* Transaction Service 🡪 Account Service:
  + Kết nối: gRPC.
  + Mô tả: Transaction Service kiểm tra số dư từ Account Service trước khi chuyển tiền hoặc rút tiền.
* Transaction Service 🡪 Payment Service:
  + Kết nối: REST/gRPC.
  + Mô tả: Payment Service gọi Transaction Service để trừ tiền khi quét QR.
* Transaction Service 🡪 Notification Service:
  + Kết nối: Kafka
  + Mô tả: Transaction Service gửi sự kiện (giao dịch hoàn tất) để Notification Service gửi thông báo.
* Loan Service 🡪 Account Service
  + Kết nối: gRPC.
  + Mô tả: Loan Service kiểm tra thông tin tài khoản để đánh giá điều kiện vay.
* Savings Service 🡪 Account Service:
  + Kết nối: gRPC.
  + Mô tả: Savings Service cập nhật tài khoản chính khi mở tài khoản tiết kiệm.
* Admin Service 🡪 Account Service:
  + Kết nối: gRPC.
  + Mô tả: Admin Service gọi Account Service để quản lý tài khoản người dùng.
* Các service 🡪 Database Cluster:
  + Kết nối: SQL/NoSQL qua driver (JDBC, MongoDB client).
  + Mô tả: Mỗi dịch vụ truy cập cơ sở dữ liệu riêng hoặc phân vùng theo chức năng.

3. Thuộc tính của các thành phần