**WEB SECURITY LAB05**

**MÔN HỌC: Lập Trình Web (Thí Nghiệm) (CO3050)**

**GIẢNG VIÊN HƯỚNG DẪN: Nguyễn Hữu Hiếu**

**SINH VIÊN THỰC HIỆN: Lô Hoàng Bảo – 2252066**

**1. SOAP Web Service**

1. **Định nghĩa:** SOAP (Simple Object Access Protocol) là một giao thức dựa trên XML để trao đổi thông điệp giữa các ứng dụng qua HTTP, SMTP… SOAP định nghĩa chặt chẽ cách đóng gói (envelope), định dạng dữ liệu (XML Schema) và cơ chế xử lý lỗi (fault).
2. **Ưu điểm**

* **Chuẩn mực, chặt chẽ**: SOAP có WSDL (Web Services Description Language) để mô tả đầy đủ interface, kiểu dữ liệu, endpoint, giúp tự động sinh client/server dễ dàng.
* **Độc lập nền tảng và ngôn ngữ**: Dựa trên XML, SOAP có thể được triển khai trên bất cứ nền tảng nào, ngôn ngữ nào.
* **Hỗ trợ WS**: Cho phép bảo mật đầu-cuối (end-to-end), giao dịch phân tán, đảm bảo tin nhắn đến đích, v.v.
* **Hỗ trợ nhiều giao thức**: Không chỉ HTTP, mà còn SMTP, JMS…

1. **Nhược điểm**

* **Cồng kềnh, overhead cao**: Thông điệp XML lớn, nhiều phần header, envelope, làm tăng băng thông và độ trễ.
* **Phức tạp**: Cấu hình WSDL, WS-\* khá rắc rối, yêu cầu hiểu sâu về XML Schema, SOAP binding.
* **Khó mở rộng nhẹ nhàng**: Việc thêm chức năng mới hoặc versioning có thể phải thay đổi WSDL, client cần cập nhật lại.

**2. RESTful Web Service**

1. **Định nghĩa**: REST (Representational State Transfer) là một kiến trúc (architecture style) cho hệ phân tán, dựa trên các nguyên tắc của HTTP. RESTful Web Service sử dụng các HTTP method (GET, POST, PUT, DELETE…) để thao tác tài nguyên (resources) được định danh bởi URI, và trao đổi dữ liệu thường ở định dạng JSON hoặc XML.
2. **Ưu điểm**

* **Đơn giản, nhẹ**: Dựa trên HTTP, không cần tiêu chuẩn phức tạp như WSDL/SOAP. Dữ liệu thường JSON, gọn, dễ đọc/ghi.
* **Khả năng mở rộng cao**: Thiết kế tài nguyên rõ ràng, dễ thêm endpoint mới, versioning linh hoạt (ví dụ /v1/..., /v2/...).
* **Tận dụng caching của HTTP**: GET responses có thể được cache, giảm tải server.
* **Phổ biến, hỗ trợ rộng**: Hầu hết các framework hiện nay (Express, Spring Boot, Django REST Framework…) đều hỗ trợ sẵn.

1. **Nhược điểm**

* **Không chuẩn hoá mô tả dịch vụ**: Không có chuẩn như WSDL, phải tự thiết kế API docs (Swagger/OpenAPI).
* **Bảo mật và giao dịch**: REST dựa trên HTTPS/TLS để bảo mật; thiếu các tiêu chuẩn WS-\* cho giao dịch phức tạp hoặc đảm bảo “đúng một lần” (exactly-once).

**Không phù hợp với tác vụ cần trao đổi tin nhắn phức tạp**: Ví dụ cần routing, reliable messaging, transaction phân tán…