1.1Các phương thức kết nối bên thứ 3 là **:**

**REST (Representational State Transfer):** Đây là kiến trúc được sử dụng rộng rãi nhất cho API. Cách tiếp cận máy khách / máy chủ tách biệt giao diện người dùng API API khỏi phần phụ trợ của nó là điều làm cho phương pháp này linh hoạt.

REST không stateless và điều này có nghĩa là API không lưu trữ bất kỳ dữ liệu nào giữa các yêu cầu. Các API được xây dựng với REST được gọi là RESTful APIs và cả hai đều có thể giao tiếp trực tiếp hoặc sử dụng cổng API và bộ cân bằng tải cho điều đó.

**SOAP (Simple Object Access Protocol):** SOAP thường được ghép nối với XML để tạo API web và hỗ trợ nhiều giao thức như HTTP, SMTP, TCP và các giao thức khác. So với REST, SOAP có cấu trúc và tiêu chuẩn hóa hơn.

**RPC (Remote Procedural Call):** RPC là một cách đơn giản để gửi tham số và nhận kết quả. Mặc dù REST APIs được sử dụng để trao đổi dữ liệu và tài nguyên, RPC phù hợp hơn cho các quy trình và hành động. RPC tương thích với JSON và XML để mã hóa và tùy thuộc vào ngôn ngữ, các API sẽ được gọi là JSON-RPC hoặc XML-RPC

#### **1.2** **Basic authentication**

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| * Đơn giản, được hầu hết các trình duyệt, webserver (nginx, apache,...) hỗ trợ * **Dễ dàng kết hợp với các phương pháp khác** | * **Username/password dễ bị lộ** * **Không thể logout** * **Không thân thiện với người dùng** |

**Session-based Authentication**

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| * Thông tin được giấu kín * Dung lượng truyền tải nhỏ * Không cần tác động client * Fully-controlled session: cho phép hệ thống quản trị tất cả hđ liên quan đến đăng nhập | * **Chiếm nhiều bộ nhớ** * **Khó scale: do sinh ra token** * **Phụ thuộc domain** * **CSRF: tấn cộng giả mạo** |

## **Token-based Authentication**

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| * **Stateless:** phục vụ cho việc scale ứng dụng theo chiều ngang do không sinh ra token * **Phù hợp với nhiều loại client:** * **Chống CSRF: chống tấn cộng giả mạo** * **Không bị giới hạn bởi domain:** | * **Khó quản lý đăng xuất** * **Phức tạp phần client** * **Thông tin dễ lộ** * **Dung lượng truyền tải lớn** |

**3.1 Opera PMS giao tiếp với bên thứ 3 bằng giao thức SOAP (Simple Object Access Protocol)**

**3.2** Chỗ cần lưu ý thì chắc là vấn đề bảo mật, tốc độ đường truyền, cấu hình mạng

4.

Đề nghị chia dự án ra thành các phase nhỏ. Xác định độ ưu tiên của công việc dựa trên mức độ cần thiết và mức độ impact. Đưa ra timeline cụ thể cho từng công việc. Nếu là tích hợp API thì có thể yêu cầu đối tác cung cấp schema request và response của các API trước, bên mình ráp thô và ngược lại, ko cần phải đợi đến khi hoàn thành api. Quan trọng là phải rút ra được phần việc nào là quan trọng nhất, phần nào có thể làm sau.

5.

Theo em thì phương thức check-in trong thời gian tới sẽ phổ biến là Mobile check- in

Vì: Tiết kiệm thời gian làm thủ tục để nhận phòng

Đặt trước được phòng, lựa chọn được phòng mong muốn

Check được chất lượng của khách sạn thông qua đánh giá

Lựa chọn được mức giá hoặc hình thức thanh toán mong muốn

2.1

2.2