CHƯƠNG 1: GIỚI THIỆU & BÀI TOÁN NGHIỆP VỤ

Tài liệu này được biên soạn cho dự án **service-proxy** (SDK_Signning) nhằm mục đích bàn giao cho đội ngũ phát triển và vận hành (Dev/Ops).

Trang 1: Tổng quan dự án

1.1 Định nghĩa

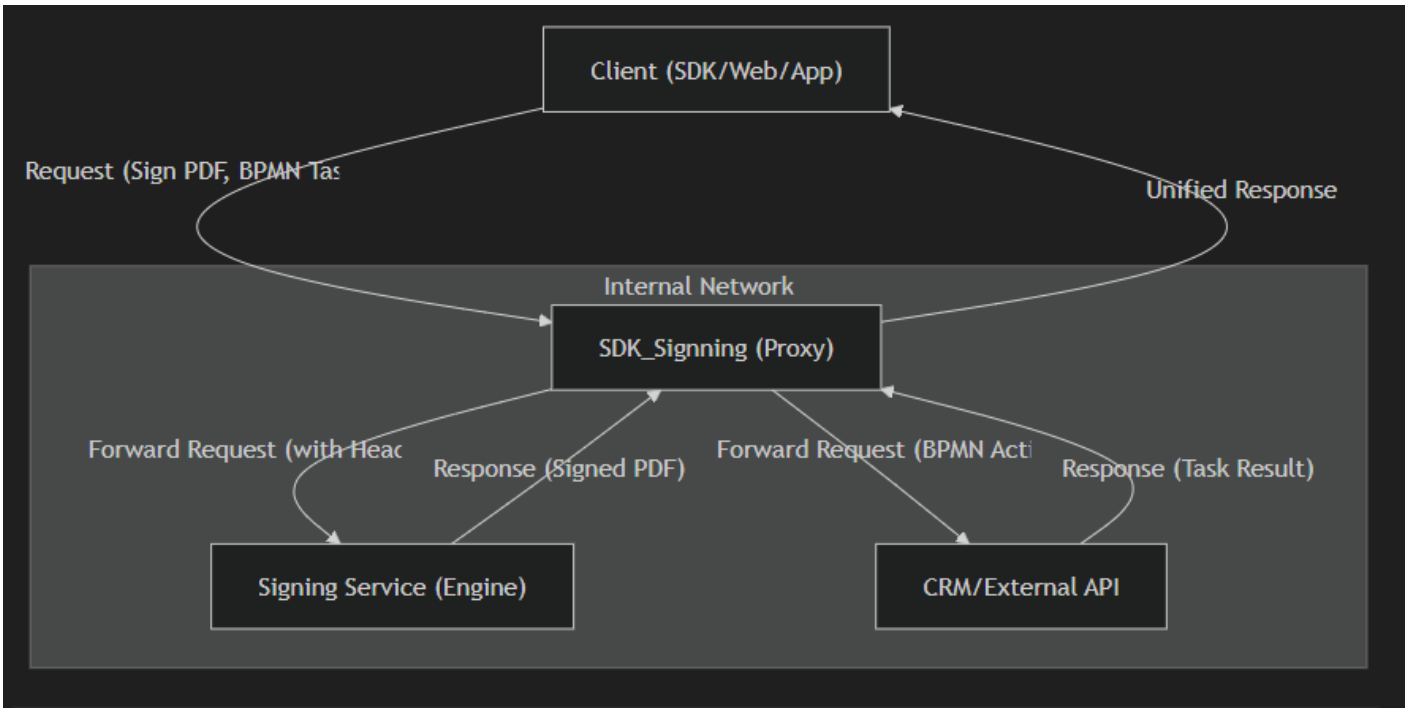
`service-proxy` là một thành phần trung gian (Middleware/API Gateway) được phát triển trên nền tảng **Spring Boot 3.3** và **Java 17**. Dự án này đóng vai trò là "cửa ngõ" (Gateway) duy nhất để các ứng dụng Client (như Mobile App, Web Frontend, hoặc các tích hợp bên thứ ba) tương tác với hệ thống ký số nội bộ và các quy trình nghiệp vụ (BPMN).

1.2 Vai trò của Proxy trong hệ thống

Dự án không trực tiếp thực hiện các thuật toán ký số hay lưu trữ cơ sở dữ liệu quy trình. Thay vào đó, nó giải quyết các vấn đề về:

- **Tập trung hóa (Centralization)**: Một đầu mối duy nhất cho mọi yêu cầu ký số.
- **Bảo mật (Security Barrier)**: Ngăn chặn truy cập trực tiếp từ Internet vào các cụm Cluster ký số nội bộ nhạy cảm.
- **Khả năng phục hồi (Resiliency)**: Tự động xử lý lỗi mạng thông qua cơ chế Retry thông minh.

1.3 Sơ đồ luồng dữ liệu (Data Flow)



Trang 2: Bài toán nghiệp vụ & Giải pháp kỹ thuật

Dự án ra đời nhằm giải quyết 4 thách thức lớn trong quá trình vận hành hệ thống ký số cũ:

2.1 Thách thức 1: Phân mảnh backend (Fragmented APIs)

- **Bài toán**: Client phải ghi nhớ và kết nối đồng thời nhiều địa chỉ Server khác nhau (Signing Engine, CRM, BPMN Server). Việc thay đổi IP/Domain của một Server yêu cầu cập nhật lại toàn bộ ứng dụng đầu cuối.
- **Giải pháp**: Proxy cung cấp các Endpoint thống nhất (`/api/sign/**`, `/api/proxy/**`). Các cấu hình Backend được quản lý tập trung trong file `application.yml`.

2.2 Thách thức 2: Độ trễ và mất kết nối mạng (Resiliency)

- **Bài toán**: Các thao tác ký PDF là tác vụ nặng, yêu cầu thời gian xử lý và băng thông. Trong môi trường mạng không ổn định, các yêu cầu thường bị thất bại (Timeout).
- **Giải pháp**: Tích hợp `Spring Retry`. Proxy sẽ tự động thử lại (tối đa 3 lần) với khoảng thời gian chờ (Backoff) trước khi trả về lỗi cho Client. Điều này giúp tăng tỷ lệ ký thành công mà người dùng không cần thao tác lại.

2.3 Thách thức 3: Bất tương thích tiêu chuẩn Header (Header Mapping)

- **Bài toán**: Ứng dụng Client hiện tại sử dụng Header `Token-Signing` để lưu trữ mã định danh, nhưng các backend chuẩn REST thường yêu cầu Bearer Token trong Header `Authorization`.
- **Giải pháp**: Proxy thực hiện ánh xạ tự động:

```
if (headers.containsKey("Token-Signing")) {
    headers.set(HttpHeaders.AUTHORIZATION, headers.getFirst("Token-Signing"));
```

```
}
```

Quá trình này diễn ra minh bạch, giúp Client không cần sửa mã nguồn cũ.

2.4 Thách thức 4: Phân vùng mạng và Bảo vệ hạ tầng

- **Bài toán**: Signing Engine chứa các chứng thư số và khóa mật mã (PKCS#11, HSM), không được phép công khai ra ngoài Internet.
- **Giải pháp**: Proxy được đặt trong phân vùng kiểm soát (DMZ), chỉ mở các cổng API cần thiết. Nó đóng vai trò là "người gác cổng", lọc các Header dư thừa (như `host`, `content-length` của Client) trước khi gửi vào mạng nội bộ để tránh xung đột giao thức.