

# Chương 2: Luồng Nghịệp vụ Chi tiết (Business Workflows)

Chương này mô tả chi tiết các trình tự xử lý và luồng dữ liệu (Data flow) của 3 nghịệp vụ cốt lõi trong hệ thống Signing Service Tân Cảng. Các biểu đồ tuần tự dưới đây giúp người đọc hình dung rõ cách các thành phần hệ thống tương tác với nhau từ lúc nhận yêu cầu đến khi trả về tài liệu đã ký.

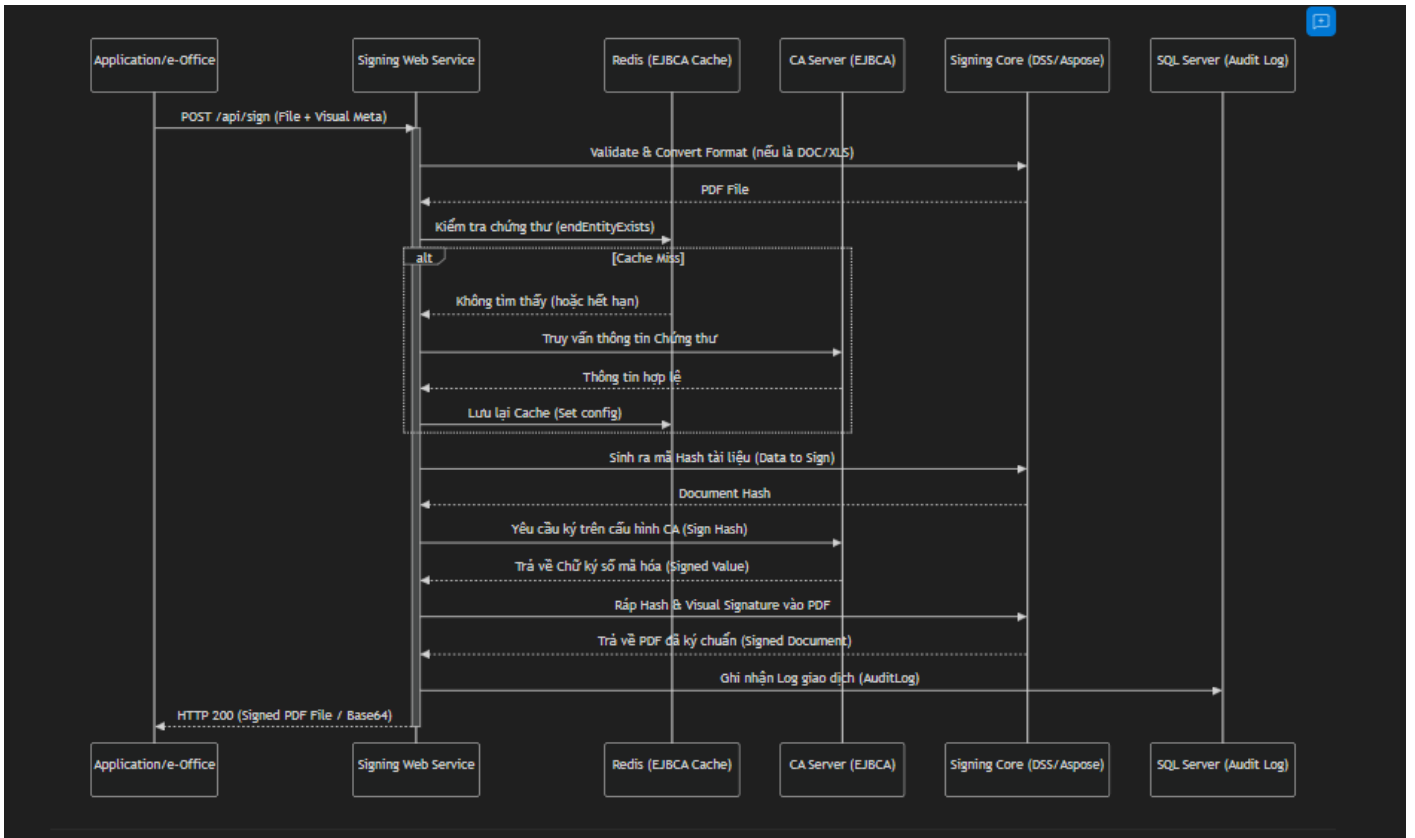
- [2.1. Luồng Ký số Tập trung \(Web Service / Server-side\)](#)
- [2.2. Luồng Ký số Local File \(Desktop App / USB Token\)](#)
- [2.3. Luồng Chuyển đổi và Đóng dấu \(Visual Only\)](#)

# 2.1. Luồng Ký số Tập trung (Web Service / Server-side)

Luồng Ký số Server-side được kích hoạt thông qua REST API trên **Signing Web Service**. Luồng này được sử dụng khi hệ thống e-Office tự động sinh tài liệu và cần đóng dấu đỏ công ty hoặc ký duyệt hàng loạt thông qua chứng thư lưu trên Server / HSM.

## Mô tả các bước:

1. **Application (Client)** gửi yêu cầu `POST /api/sign` kèm theo file tài liệu (PDF, DOCX) và các thông tin Metadata (Tọa độ hình ảnh, Page, Base64 ảnh chữ ký).
2. **Web Controller** (`SigningController`) tiếp nhận, kiểm tra tính hợp lệ của Request và Token (nếu có Auth).
3. Nếu tài liệu là định dạng Office (Word/Excel), gọi qua module **Converter (Aspose)** để chuyển thành định dạng chuẩn PDF.
4. Hệ thống kiểm tra Cache (Redis) đối với **EJBCA** để xác minh thư mục lưu thông tin chứng thư (End Entity).
5. Gọi API sang **EJBCA CA Server** ký hash của tài liệu và lấy dữ liệu chữ ký trả về.
6. Module **Signing Core (DSS)** ráp chữ ký và hình ảnh con dấu (Visual Signature) vào tài liệu PDF.
7. Ghi Log Audit vào Database và phản hồi kết quả trực tiếp cho Client.

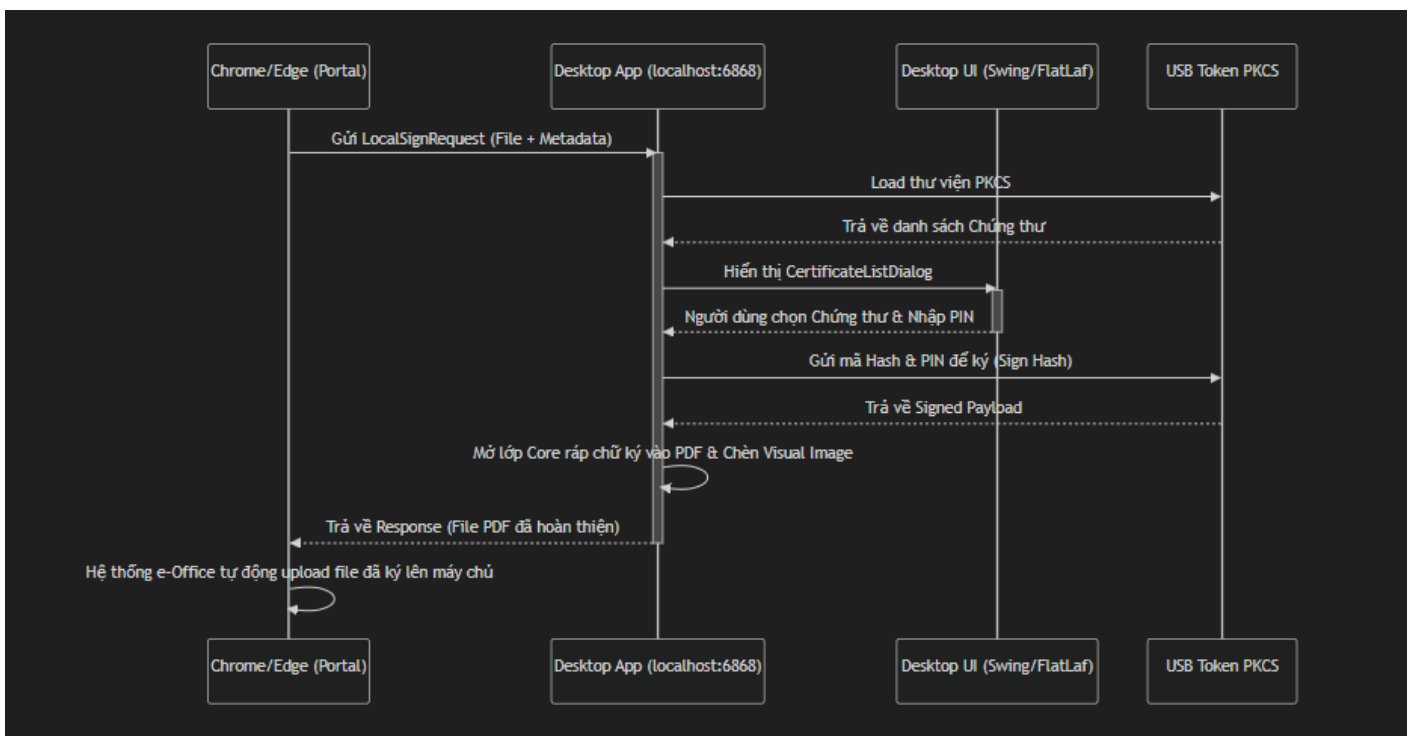


## 2.2. Luồng Ký số Local File (Desktop App / USB Token)

Luồng ký Local phục vụ cho nhu cầu nhân sự cá nhân cắm USB Token vật lý (Viettel, VNPT, FPT,...) để ký văn bản phê duyệt trên máy cá nhân.

### Mô tả các bước:

- Website/Portal nội bộ của doanh nghiệp tạo lệnh gọi **LocalSignRequest** xuống ứng dụng Desktop App đang chạy ngầm trên máy thông qua `http://localhost:6868/api/desktop/...`.
- Tọa độ** hoặc hình ảnh chữ ký được truyền theo xuống DesktopSigningController.
- Ứng dụng Desktop App khơi gọi (pop-up) giao diện **UI (TokenProfileDialog/LoadingDialog)** để lấy Danh sách chứng thư số hiện hành đang cắm vào máy.
- Yêu cầu nhập mã **PIN Token** từ người thao tác.
- Desktop App liên kết thông qua thư viện `dss-token` tạo một kết nối PKCS#11 trực tiếp vào driver USB để ký file.
- Ký thành công, Desktop App phản hồi REST API lại cho Website/Portal kết quả để Client upload file đó lên lại Server.



## 2.3. Luồng Chuyển đổi và Đóng dấu (Visual Only)

Trong một số trường hợp, người dùng chỉ cần chèn hình ảnh phê duyệt (Con dấu báo cáo, chữ ký nháy hiển thị bằng hình ảnh) mà chưa cần áp hệ thống CA mã hóa Cryptography. Hệ thống ứng dụng Endpoint `/sign-visual-only`. Thông qua đó, nếu nhận được file Office, hệ thống sẽ thực thi luồng sau:

### Mô tả các bước:

1. Tiếp nhận file định dạng `.doc` / `.docx` / `.xlsx` thông qua REST API.
2. Kiểm tra phần header nhị phân (Magic Bytes) xem file có bị đổi đuôi thủ công hay không (Ví dụ: `.pdf` đổi tên thành `.docx`). Nếu phát hiện là PDF nguyên gốc, hệ thống by-pass quy trình convert để tiết kiệm tài nguyên và bảo toàn file.
3. Nếu đúng là file Office, sử dụng **Aspose Engine (Aspose.Words / Aspose.Cells)** để convert từ cấu trúc DOM của tài liệu sang PDF chuẩn.
4. Đọc metadata (`image_info`) truyền xuống (tọa độ X, Y, số thứ tự trang cài đặt). Tính toán vị trí Absolute (Tuyệt đối) hoặc Reference (Tương đối) trên PDF.
5. Ráp chèn Hình Ảnh vào trang, lưu kết quả xuất file và trả về Client (Không qua khâu tương tác DSS CA/EJBCA).