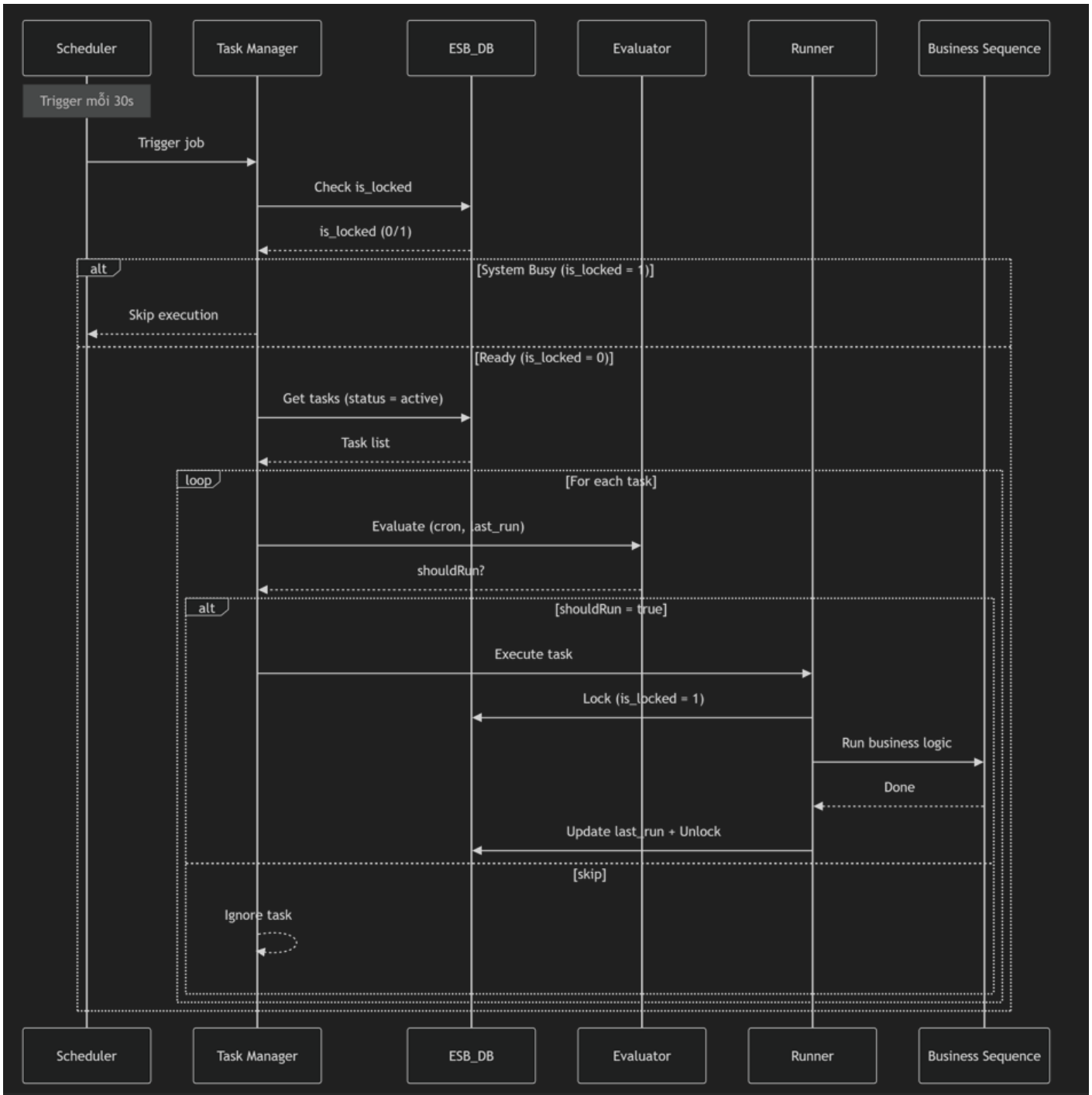


TRANG 1: Sơ đồ luồng hoạt động tổng quát (Sequence Diagram)

Hệ thống đồng bộ của LGSP không chạy theo lịch cứng mà chạy theo cơ chế "Quét và Đánh giá" (Scan & Evaluate) dựa trên cấu hình linh hoạt trong Database.



Sau khi Task kích hoạt, hệ thống sẽ đi qua một chuỗi các Sequence có nhiệm vụ chuyên biệt để đảm bảo tính an toàn và linh hoạt.

1.1. Bước 1: Quản lý Luồng (ManagerEvaluateTasksSequence)

Đây là sequence "cửa ngõ". Nhiệm vụ chính:

- **Kiểm soát concurrency:** Sử dụng `dataServiceCall` tới bảng `sync_lock`. Nếu có một tiến trình `RunTaskSequence` trước đó chưa kết thúc (vẫn đang `is_locked=1`), nó sẽ dừng ngay lập tức.
- **Phân tách dữ liệu:** Sử dụng mediator `<iterate>` để biến danh sách Task từ Database thành từng gói tin độc lập cho mỗi Task.

1.2. Bước 2: Đánh giá Cron (EvaluateSingleTaskSequence)

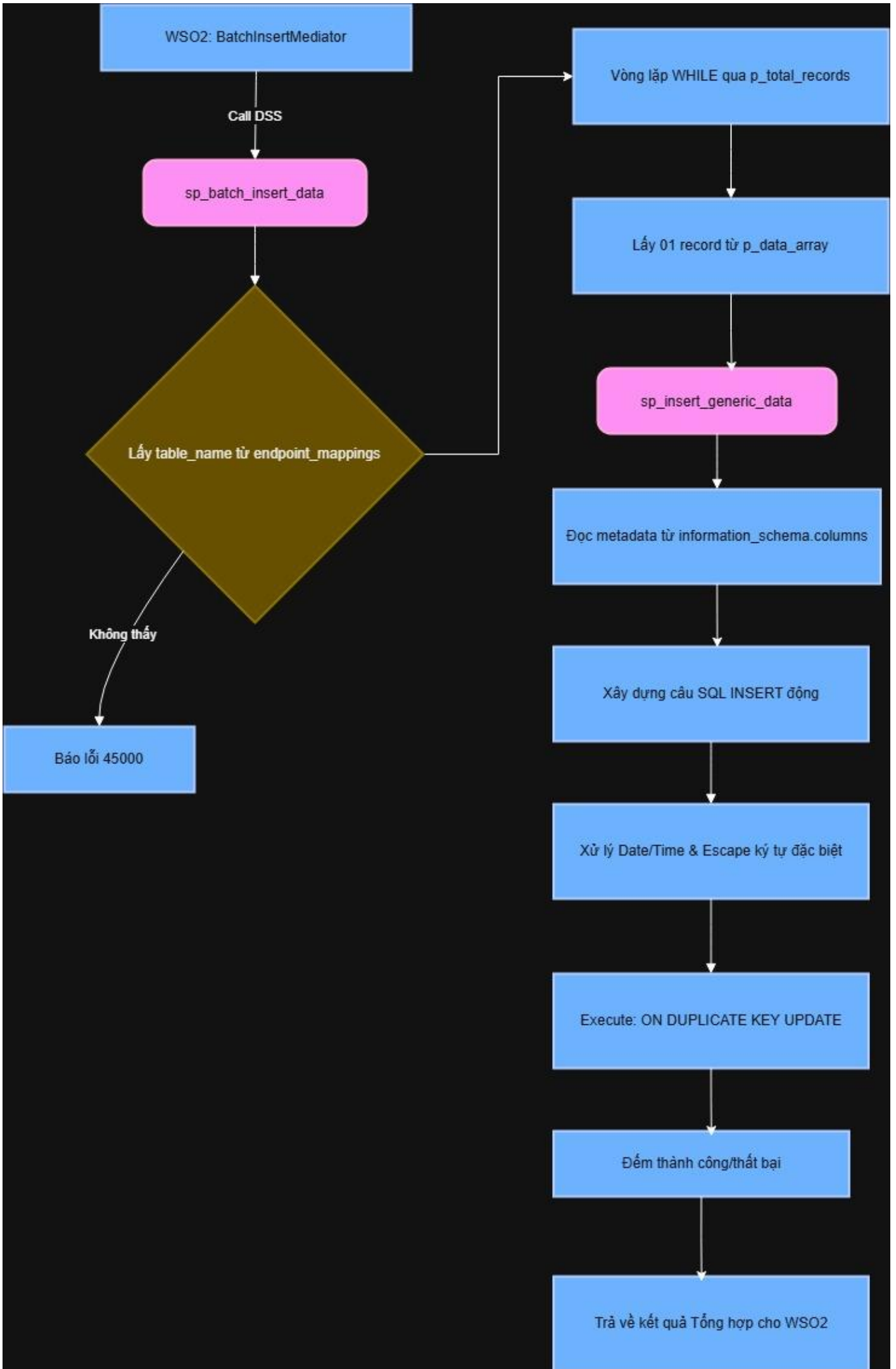
Với mỗi Task được tách ra, bộ Evaluator sẽ:

- Trích xuất `cron_expression` (ví dụ: `0 */15 * * * ?` - chạy mỗi 15 phút).
- Sử dụng **Script Mediator (JavaScript)** để tính toán thời gian chạy tiếp theo.
- **Kết quả:** Nếu thỏa mãn điều kiện thời gian, nó sẽ nạp biến `shouldRun = true` vào Message Context.

1.3. Bước 3: Bộ định tuyến động (RunTaskSequence)

Sequence này đóng vai trò là "Cầu giao điện" tổng. Nó thực hiện:

- **Khóa hệ thống:** Gọi `lockSync` để ngăn các Task khác nhảy vào chiếm dụng tài nguyên.
- **Điều hướng (Switch Mediator):** Dựa vào tên Sequence lưu trong database (`sequence_name`), nó sẽ điều hướng tới logic nghiệp vụ thật sự:
 - `MultiEndpointSyncSequence`: Đồng bộ hóa đa điểm tiêu chuẩn.
 - `DongBoDVLTSquence`: Đồng bộ dữ liệu dịch vụ lưu trữ (Sử dụng `BatchInsertMediator` để nạp dữ liệu lớn).
 - `DongboVBDenEdocSequence`: Chuyên trách đồng bộ Văn bản điện tử.
- **Hoàn tất:** Sau khi nghiệp vụ chạy xong, nó cập nhật `last_run` vào DB và thực hiện `unlockSync`.



1.4.2. Phân tích Procedure bộ điều phối:

sp_batch_insert_data

Nhiệm vụ chính: **Phân rã mảng dữ liệu (Array Parsing)**.

- **Tham số đầu vào:** `p_endpoint_path` (để xác định bảng đích), `p_data_array` (chuỗi JSON khổng lồ chứa hàng trăm bản ghi), `p_total_records`.
- **Cơ chế chịu lỗi (Fault Tolerance):** Procedure sử dụng `DECLARE CONTINUE HANDLER FOR SQLEXCEPTION`. Điều này cực kỳ quan trọng: Nếu trong 100 bản ghi có 1 bản ghi bị lỗi dữ liệu, Procedure sẽ **không dừng lại** mà chỉ tăng biến `v_error_count`, sau đó tiếp tục xử lý bản ghi tiếp theo.
- **Kết quả trả về:** Một tập bản ghi (ResultSet) chứa: `status` (success, partial, error), `message` chi tiết và `record_count` thực tế đã nạp thành công.

1.4.3. Phân tích Procedure thực thi động:

sp_insert_generic_data

Đây là chìa khóa của sự linh hoạt. Nó thực hiện **Tự động nhận diện cấu trúc bảng (Table Discovery)**.

- **Bước bóc tách dữ liệu:** Procedure kiểm tra xem JSON có bọc trong thẻ `$.data` hay không để lấy đúng nội dung cần thiết.
- **Bước ánh xạ cột (Dynamic Mapping):**
 - Nó truy cập vào `information_schema.columns` để xem bảng đích có những cột nào, kiểu dữ liệu là gì.
 - Nếu cột trong JSON không có trong Database -> Tự động bỏ qua (Giúp hệ thống không bị chết khi đối tác thêm trường dữ liệu mới).
- **Xử lý kiểu dữ liệu:**
 - Tự động nhận diện các cột `DATE`, `DATETIME`, `TIMESTAMP` để sử dụng hàm `STR_TO_DATE`.
 - Tự động thay thế (`REPLACE`) các ký tự đặc biệt như dấu nháy kép, dấu gạch chéo để chống lỗi cú pháp SQL.
- **Cơ chế Upsert:** Sử dụng cú pháp `ON DUPLICATE KEY UPDATE updated_at = NOW()`.
 - Nếu bản ghi đã tồn tại (trùng Primary Key): Cập nhật thông tin mới nhất.
 - Nếu chưa có: Chèn mới.
 - Điều này giúp dữ liệu trong kho địa phương luôn đồng bộ với trục Quốc gia mà không gây trùng lặp.

1.4.4. Các bảng cấu hình phụ thuộc

Để 2 Procedure này hoạt động, bạn cần cấu hình bảng mapping

```
CREATE TABLE endpoint_mappings (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  endpoint_path VARCHAR(200) NOT NULL, -- Đường dẫn API (Vd: /api/v1/don-vi)  
  table_name VARCHAR(100) NOT NULL, -- Tên bảng trong Database  
  status TINYINT DEFAULT 1 -- 1: Kích hoạt  
);
```

Phiên bản #3

Được tạo 2026-04-02 08:36:14 UTC bởi Nam Đặng

Được cập nhật 2026-04-02 09:25:28 UTC bởi Nam Đặng