

# LEO Activation Platform

## Kế Hoạch Triển Khai POC 12 Ngày

**Mục tiêu:** Xây dựng "Bộ não & Cơ bắp" AI-first cho LEO CDP.

**Core Tech:** FunctionGemma (AI), PostgreSQL 16 (Core), Celery (Async).

**Ràng buộc:** 12 ngày phải có Demo.

**Owner:** Product & Engineering

**Ngày:** 7/1/2026

## Bức tranh tổng thể (The Big Picture)

1. **Ingest (Thu thập):** Đồng bộ dữ liệu từ LEO CDP (ArangoDB) → Postgres.
2. **Think (Tư duy):** FunctionGemma phân tích Profile + Context → Quyết định "Làm gì" (Agent Task).
3. **Target (Nhắm mục tiêu):** Snapshot Segment (bất biến).
4. **Act (Hành động):** Dispatch tới các kênh (Zalo, Email, Push).

## Timeline Sprint (12 Ngày)

- **Phase 1: Nền tảng (Ngày 1-3)**
  - Deploy `schema.sql`, Pipeline đồng bộ dữ liệu, Cơ chế Segment Snapshot.
- **Phase 2: Bộ não AI (Ngày 4-7)**
  - Logic `agent_task`, Tích hợp FunctionGemma, Truy vết quyết định (Decision Tracing).
- **Phase 3: Cơ bắp thực thi (Ngày 8-10)**
  - Channel Dispatchers, Delivery Logs, Vòng lặp phản hồi.
- **Phase 4: Ôn định hóa (Ngày 11-12)**
  - Load Testing, UAT cơ bản, Release.

## Phase 1: Nền tảng (Ngày 1-3)

**Mục tiêu: Cấu trúc dữ liệu cứng cáp, tin cậy.**

## [Jira-01] Khởi tạo Database & Extensions

### Mô tả:

Khởi tạo PostgreSQL 16 với schema production đã cung cấp. Đảm bảo kích hoạt đầy đủ extensions (`vector`, `pgcrypto`).

### Technical Tasks:

1. Provision Postgres instance.
2. Chạy `schema.sql` (Tables: `tenant`, `cdp_profiles`, `campaign`, v.v.).
3. Kiểm tra Partitioning trên bảng `marketing_event`.
4. Kiểm tra RLS (Row Level Security) đã hoạt động chưa.

### Definition of Done (DoD):

- [ ] `\d marketing_event` hiển thị đủ 16 partitions.
- [ ] Insert vào `cdp_profiles` chỉ thành công khi có `tenant_id` hợp lệ.
- [ ] Trigger `update_timestamp()` hoạt động đúng.

## [Jira-02] Worker Đồng bộ Dữ liệu (ArangoDB → Postgres)

### Mô tả:

Xây dựng Celery worker để kéo dữ liệu profile từ LEO CDP ArangoDB và upsert vào bảng `cdp_profiles` của Activation.

### Technical Tasks:

1. Tạo `SyncProfileWorker`.
2. Map các thuộc tính từ Arango sang cột Postgres (`email`, `mobile`, `raw_attributes`).
3. Xử lý logic `ON CONFLICT (tenant_id, ext_id)`.

### Definition of Done (DoD):

- [ ] Độ trễ Sync < 200ms cho batch 100 profiles.
- [ ] Dữ liệu JSONB trong `raw_attributes` có thể query được qua GIN index.
- [ ] Không sinh ra profile trùng lặp.

## [Jira-03] Segment Snapshot Engine

### Mô tả:

Implement logic "đóng băng". Khi campaign kích hoạt, hệ thống phải ghi lại chính xác ai đang ở trong segment tại thời điểm đó.

### Technical Tasks:

1. API: `POST /snapshot/create`.
2. Logic: Query profiles theo điều kiện segment → Insert vào `segment_snapshot` → Bulk insert `segment_snapshot_member`.
3. **Ràng buộc cứng:** Kiểm tra trigger `prevent_snapshot_removal` có hoạt động không.

### Definition of Done (DoD):

- [ ] Tạo snapshot cho 10k profiles trong < 2 giây.
- [ ] Cố tình xóa data trong `segment_snapshots` phải bị DB bắn lỗi Exception.
- [ ] `snapshot_id` link đúng với `tenant_id`.

## Phase 2: Bộ não AI (Ngày 4-7)

**Mục tiêu: Text-to-Function & Truy vết Quyết định.**

## [Jira-04] FunctionGemma Model Service

### Mô tả:

Deploy FunctionGemma model (qua API wrapper) để dịch intent marketing thành các function call có cấu trúc.

### Technical Tasks:

1. Setup LLM Gateway (FunctionGemma).
2. Định nghĩa Tools/Functions Schema:
  - `send_notification(channel, template_id, params)`
  - `wait_duration(minutes)`
  - `check_condition(attribute, operator, value)`
3. Implement Prompt Template sử dụng context từ `cdp_profiles`.

### Definition of Done (DoD):

- [ ] Input: "Gửi Zalo nếu là khách VIP" → Output: JSON Function Call đúng cú pháp.
- [ ] Latency < 2s để sinh ra quyết định.

## [Jira-05] Agent Task Orchestrator

### Mô tả:

Vòng lặp cốt lõi quản lý vòng đời của một tác vụ AI, sử dụng bảng `agent_task`.

### Technical Tasks:

1. API `POST /activate/agent`.
2. Tạo record trong `agent_task` với `status = 'pending'`.
3. Gọi FunctionGemma.
4. Lưu suy luận của AI vào `reasoning_trace` (JSONB) và tóm tắt vào `reasoning_summary`.
5. Cập nhật kết quả vào `agent_task`.

### Definition of Done (DoD):

- [ ] Bảng `agent_task` lưu đầy đủ luồng suy nghĩ (thought process) của AI.
- [ ] Xử lý Retry logic nếu AI call bị lỗi (max 3 lần).
- [ ] RLS đảm bảo Agent của tenant nào chỉ thấy task của tenant đó.

## Phase 3: Cơ bắp thực thi (Ngày 8-10)

**Mục tiêu: Gửi tin chính xác (Deterministic Delivery).**

## [Jira-06] Unified Dispatcher & Delivery Log

### Mô tả:

Lớp trừu tượng (Abstraction layer) điều phối lệnh tới các channel adapter và ghi log kết quả làm **sự thật duy nhất (truth)**.

### Technical Tasks:

1. Tạo class `Dispatcher` (Factory Pattern).
2. Implement insert `delivery_log` trước và sau khi gọi external API.
3. Đảm bảo sinh `event_id` dùng hàm hash tất định (deterministic) từ `schema.sql`.

### Definition of Done (DoD):

- [ ] `delivery_log` ghi đủ: `sent_at`, `provider_response` (JSON), `delivery_status`.
- [ ] An toàn transaction: Nếu ghi log lỗi thì không được gửi tin.

## [Jira-07] Channel Adapter: Zalo OA & Email

### Mô tả:

Implement các connector cụ thể cho thị trường Việt Nam.

### Technical Tasks:

1. **Zalo Adapter:** Xử lý OA Token refresh, ZNS Template API, Rate Limiting (lỗi 429).
2. **Email Adapter:** Tích hợp SMTP/SendGrid với HTML templating.
3. **Validation:** Chuẩn hóa số điện thoại theo định dạng Zalo (84...).

### Definition of Done (DoD):

- [ ] Gửi Zalo ZNS thành công với tham số động lấy từ Profile.
- [ ] Số điện thoại rác phải trả về `delivery_status = 'failed'` (không được crash worker).

## [Jira-08] Channel Adapter: Push (Web/App) & Telegram

### Mô tả:

Các kênh thông báo thời gian thực (Real-time).

### Technical Tasks:

1. **Telegram:** Tích hợp Bot API (Lookup Chat ID từ Profile).
2. **Push:** Tích hợp Firebase (FCM) hoặc PushAlert.
3. **Queueing:** Các kênh này volume lớn, cần tách Celery queue riêng (`priority_high`).

### Definition of Done (DoD):

- [ ] Push notification ting ting trên máy < 1s sau khi dispatch.
- [ ] Telegram xử lý lỗi Markdown parsing (escape ký tự đặc biệt).

## Phase 4: Ôn định hóa (Ngày 11-12)

**Mục tiêu: Không có lỗi ngầm (No Silent Failures).**

## [Jira-09] End-to-End Traceability Test

### Mô tả:

Kiểm chứng "Luồng vàng" (Golden Path) từ Event tới Delivery Log.

### Technical Tasks:

1. Bắn thử event → Check hash trong `marketing_event`.
2. Check suy luận trong `agent_task`.
3. Check trạng thái `delivery_log`.
4. Verify tính toàn vẹn của `segment_snapshot`.

### Definition of Done (DoD):

- [ ] Một câu SQL Query join 4 bảng phải ra được toàn bộ hành trình của 1 user.
- [ ] Không có log "mồ côi" (orphaned logs).

## [Jira-10] Load Testing & Documentation

### Mô tả:

Đảm bảo hệ thống chịu được tải Demo POC.

### Technical Tasks:

1. Giả lập 5,000 events/phút.
2. Monitor CPU Postgres và hiệu năng của Partition.
3. Cập nhật `README.md` hướng dẫn API cho team Frontend.

### Definition of Done (DoD):

- [ ] Hệ thống nuốt trôi 5k events mà không bị lock bảng `delivery_log`.
- [ ] Tỉ lệ lỗi API < 1%.

## Rủi ro & Giải pháp (Mitigation)

Rủi ro	Tác động	Giải pháp
AI Ảo giác (Hallucination)	Agent gọi sai hàm	Validate chặt chẽ JSON Output bằng Schema cứng.
DB Locking	Gửi tin chậm	Partitioning giúp chia nhỏ tải insert; Monitor kỹ.
Zalo/FB Tokens chết	Gửi thất bại	Chạy Worker tự động refresh token mỗi 12h.

## Hành động ngay (Day 0)

1. **DevOps:** Dựng Postgres 16 instance.
2. **Lead Dev:** Review lại chiến lược Partitioning trong `schema.sql`.
3. **PM:** Đẩy các ticket này vào Jira Backlog.

"Code wins arguments. Ship it."