Fly GACA — خط معالجة حداثة المدوّنة — مواصفة التصميم

القسم operations-it-06 · النوع spec · الحالة active · المالك Founder · آخر تحديث 19-08-2026

الحالة: مبنيّ جزئيًّا · **يرتبط بـ:** البنود الشاملة في `roadmap.md` (حداثة AIP/AIRAC)، المرحلة الثانية للابتلاع، المرحلة العاشرة · **أُعِدّ في:** 30-05-2026 · **أُعيد تأسيسه على المنظومة الحالية:** 19-08-2026 **الوثيقة المرافقة:** `strategy-competitive-teardown.md`

يسمّي التفكيك التنافسي الحداثة بوصفها ضعفًا بنيويًّا لا يستطيع مشرف واحد الإمساك به بشكل موثوق، ويصف أول استشهاد ملغى بأنه اللحظة التي يتصدّع فيها عرض الثقة لدينا. وتحوّل هذه المواصفة تلك المسؤولية إلى **ضمان مرئيّ ومؤتمت ومختوم** — الموضع الوحيد الذي سيهاجمه منافس مموّل، مُغلَقًا قبل أن يتمكّن منه.

يتغيّر AIP في كل **دورة AIRAC مدّتها 28 يومًا**؛ أمّا أجزاء GACAR فتُعدَّل بشكل غير منتظم. واليوم، يجري تحديث كليهما شبه يدوي. وتجعل هذه المواصفة التقادم *قابلًا للكشف ومؤرَّخًا ومحدودًا باتفاقية مستوى خدمة (SLA).*

> خط المعالجة **للبيانات العامة فقط**. يعمل على الخادم الافتراضي الخاص (VPS) الأوروبي (`setup/setup-vps.md`، وحدود PDPL في `hosting-facts.md`) أو داخل CI — ولا يعمل أبدًا على Cloud SQL. يبتلع المدوّنة العامة ويقارن فروقها ويعيد بناء الفهارس؛ ولا يمسّ البيانات الشخصية أبدًا. وتُنشَر مخرجات العمل إلى حاوية المدوّنة؛ ولا تعبر أي بيانات مستخدم الحدود.

1. ساعتا الحداثة

فئة المصدر	الإيقاع	القيد
AIP (المطارات، الخرائط، AIS)	كل دورة AIRAC مدّتها 28 يومًا	يجب إظهار تاريخ السريان + "ليس للاستخدام التشغيلي"
أجزاء GACAR	تعديلات غير منتظمة	يجب تتبّع رقم/تاريخ التعديل وإعادة الابتلاع عند التغيير
الرف المرجعي (الكتيّبات، مراجع الإيكاو ICAO، FAA)	نادر	فرق منخفض الأولوية

يحمل كل أثر عمل تحت `public/data/` وفهرس RAG (`public/data/rag-chunks.json`) **بيانات المصدرية الوصفية** بحيث تكون الحداثة خاصيّة من خصائص البيانات، لا معرفة شفهية.

2. بيانات المصدرية الوصفية (تُضاف إلى كل عنصر في المدوّنة)

```
{
  "source_id": "gacar-part-61",
  "source_class": "gacar",          // gacar | aip | reference
  "source_url": "https://gaca.gov.sa/...",   // النسخة الرسمية المعتمدة
  "fetched_at": "2026-05-30T00:00:00Z",
  "effective": { "airac": "2026-06", "from": "2026-06-11" }, // aip فقط
  "amendment": "Amdt 3 (2025-11)",   // gacar فقط
  "content_hash": "sha256:…",        // يقود كشف التغيير
  "verified_cycle": "2026-05"        // الحداثة CI/آخر دورة أكّد فيها إنسان
}
```

واثنان من هذه موجودان أصلًا بصورة منشورة: فـ `public/data/sources.json` هو بيان المصادر الرسمية (مع `fingerprint` لكل مصدر = إشارة التغيير، إضافة إلى مرتكز AIRAC)، و `public/data/source-status.json` هو لقطة الحداثة العامة (`generated`، `lastChecked`، `lastUpdated`، دورة AIRAC الحالية والتالية، وإمكانية بلوغ كل مصدر). ويقرأ `src/calc/library/changeTracking.ts` كليهما، وهو ما تعرضه صفحة `/updates`.

3. مراحل خط المعالجة

1. **الجلب** — اسحب النسخ الرسمية المعتمدة لكل مصدر متتبَّع في `sources.json`. *(الاكتشاف/الاستخراج مجموعة وكلاء، واحد لكل فئة مصدر، تُخرج `records.json` مُنقَّحًا.)*
2. **التجزئة والفرق** — قارن `content_hash` لكل سجل ببصمته المحفوظة. دون تغيير ← تخطَّ. متغيّر ← علّم لإعادة الابتلاع. وتُفحص مصادر AIP وفق تقويم AIRAC بغضّ النظر عن التجزئة. **منشور:** أمر `npm run sync:gaca` (`scripts/sync-gaca.mjs`) تجربة جافّة افتراضيًّا، ويبلّغ عن الجديد / المتغيّر / غير المتغيّر.
3. **إعادة الابتلاع** — ادمج الفروق وأعد اشتقاق الفهارس، تزايديًّا لا كإعادة بناء كاملة. **منشور:** يدمج `npm run sync:gaca:apply` الفروق ذات البيانات الوصفية فقط ويحفظ بصمة مصدرية كل سجل؛ ويُنمِّط `npm run data:normalize` (`scripts/normalize-corpus-data.mjs`) أشكال المدوّنة؛ ويعيد `npm run parse:regulations` تجميع جدول الإحالات المرجعية؛ ويعيد `npm run build:chunks` (`scripts/build-rag-chunks.mjs`) بناء فهرس الاسترجاع `public/data/rag-chunks.json`؛ ويحدّث `npm run embeddings:upsert` مخزن pgvector في Supabase لمسار الاسترجاع الكثيف. أمّا السجلّات الحاملة لمتن (أصل PDF/eAIP خام) فما تزال تحتاج خطوة التحويل ويُبلَّغ عنها كمؤجَّلة بدل أن تُكتب.
4. **الختم** — اكتب/حدّث كتلة المصدرية وارفع `verified_cycle` في `source-status.json`.
5. **التحقق** — شغّل حزمة التقييم على المدوّنة المعاد بناؤها؛ فأي استشهاد لم يعد يُحلّ إلى قسم حيّ **يُفشِل البناء** (يرتبط ببوابة تقييم CI في المرحلة العاشرة). وتعيش التقييمات في `iflygaca/Captain-Adel` (`evals/`).
6. **النشر** — زامِن `public/data/` المُحدَّثة إلى حاوية المدوّنة (التي يقدّمها عامل الخدمة بنهج **الشبكة أولًا**، فتصل مدوّنة محدّثة إلى العملاء دون نشر تطبيق جديد)، وأصدر مراجعة Cloud Run جديدة كي تلتقط واجهة البرمجة `rag-chunks.json` المعاد بناؤه — إذ يخبزه ملف Dockerfile في الصورة فلا يحتاج BM25 إلى جلب عند البدء البارد. ووسِم المراجعة بدورة AIRAC.

```
fetch → hash/diff → [changed?] → re-ingest → stamp → eval-gate → publish
                        │no                                        ├─ corpus bucket
(network-first)
                        └──────────────────────────────→ skip      └─ Cloud Run revision
(rag-chunks.json)
                           (log "current")
```

[NOTE!] **المُجدوِل هو النصف الغائب.** فما يزال `public/data/sources.json` و `source-status.json` يسمّيان `scripts/update-sources.js` وملف `.github/workflows/update-sources.yml` بوصفهما مالكيهما؛ ولا وجود لأيٍّ منهما في `iflygaca/FlyGACA` اليوم (فالمستودع يُشحن بلا مجلّد `.github/workflows/` أصلًا)، وقد وُلِّد `source-status.json` آخر مرة في 2026-06-15 مع وسم `gacar` بأنه `unreachable`. فالنصف الخاص *بالتطبيق* من خط المعالجة حقيقي وقابل للتشغيل؛ أمّا نصف *الأتمتة* فهو وظيفة cron (على VPS أو Cloud Scheduler) لم يُعِد أحد إنشاءها منذ النقل. وتلك أعلى فجوة قيمةً في هذه المواصفة — فختم حداثة لا يُحدَّث هو بالضبط الإخفاق الذي تحذّر منه §4.

4. الضمان المرئي للمستخدم (السطح التسويقي)

- **ختم حداثة على كل صفحة قارئ:** "GACAR Part 61 — Amdt 3, verified AIRAC 2026-05" / "AIP AD — not for operational use · effective AIRAC 2026-06." أعد استخدام منطق الدورة في أداة حاسبة AIRAC الموجودة.
- **شريط تقادم:** إذا كانت `verified_cycle` أقدم من دورة AIRAC الحالية لعنصر من فئة AIP، تُظهِر الصفحة تلقائيًّا "التحقق معلّق للدورة الحالية — تأكّد من نسخة AIP الرسمية"
- **سطر حالة عام** ("المدوّنة محدّثة اعتبارًا من AIRAC YYYY-CC") على واجهة المكتبة، إضافة إلى تغذية التغييرات `/updates` لمعرفة "ما الذي تغيّر منذ آخر اطّلاع".

هذا هو عامل التمييز: *محدّثة دائمًا، يُتحقَّق منها كل دورة، مع التاريخ على الصفحة* — ادعاء لا يستطيع مشرف يدوي واحد تقديمه بمصداقية ويمكن لمنافس عندئذٍ أن يسوّق ضدّه. وما يجعله حقيقيًّا لا تطلّعيًّا هو الفرق + الختم + بوابة التقييم المؤتمتة.

5. تكامل الكابتن عادل

- يفضّل الاسترجاع المقطع الأحدث؛ وتُستبعَد المقاطع الملغاة من `rag-chunks.json` وقت البناء، لا يُخفَّض ترتيبها وقت الاستعلام فحسب.
- يستشهد موجّه النظام أصلًا بالجزء الدقيق (`server/src/captain-adel-prompt.ts`)؛ وسّع استشهاد `ChatSource` المبنيّ في `server/src/corpus.ts` ليحمل **ختم التعديل / AIRAC** بحيث تكون الإجابة مؤرَّخة، لا مصدَّرة فحسب. وكتلة `lineage` في المقطع تحمل أصلًا حقل `effective_date` يُعلَّق عليه ذلك.
- يُبَتّ في التأصيل من جانب الخادم انطلاقًا من ثقة الاسترجاع (`REFUSE_SCORE` / `GROUNDED_SCORE`)، فتنحدر المدوّنة المتقادمة والرقيقة إلى رفض بدل إجابات خاطئة واثقة. وذلك هو الأمان عند الفشل، لا الهدف.
- تكتسب حزمة التقييم **فئة حالات تقادم**: فالإجابة التي تستشهد بمصدر ملغى تُعدّ إخفاقًا قاطعًا، إلى جانب حالات الاستشهاد/الرفض/الحقن الموجودة.

6. اتفاقية مستوى الخدمة (SLA) (الالتزام الذي يمكّننا خط المعالجة من الوفاء به)

فئة المصدر	هدف التحديث
فئة AIP	إعادة التحقق خلال **أول 72 ساعة من كل دورة AIRAC جديدة**
GACAR	إعادة الابتلاع خلال **7 أيام** من تعديل مكتشَف
المرجع	بأفضل جهد، فرق شهري

7. ما الذي يُبنى، بالترتيب

1. **استعادة وظيفة الجلب + التجزئة/الفرق المجدولة** — مشغّل cron (على VPS) أو Cloud Scheduler يشغّل `sync:gaca` ويعيد كتابة `source-status.json`. أرخص خطوة وأعلاها قيمة، وهي حاليًّا ما يعطّل كل ادعاء تحتها.
2. استكمال كتلة المصدرية عبر مخطّط المدوّنة + استكمال العناصر الموجودة بأثر رجعي.
3. ختم حداثة صفحة القارئ + شريط التقادم (مكسب مرئي للمستخدم، بجهد صغير).
4. وصل إعادة الابتلاع التزايدية (`sync:gaca:apply` ← `data:normalize` ← `build:chunks`) بمخرجات الفرق، وصولًا إلى مزامنة الحاوية ومراجعة Cloud Run.
5. حالة تقييم التقادم + بوابة CI (تقترن بعمل تقييم CI في المرحلة العاشرة).
6. سطر الحالة العام "محدّثة اعتبارًا من AIRAC" ووصل تغذية `/updates`.

تحوّل الخطوات 1–3 وحدها الضعف إلى قوّة مسوَّقة؛ وتجعل الخطوات 4–6 الضمان قائمًا بذاته.