

ADOBE FIREFLY SERVICES · FORWARD DEPLOYED ENGINEER — ASSESSMENT, IN ONE PLACE

FDE Social Content Agentic Automation & Analytics

A campaign brief goes in. An organized folder of on-spec, checked, localized social creatives comes out — every product, every market, every aspect ratio — from as few generative calls as the brief actually requires. Then two scheduling engines turn evidence into a dated, per-channel campaign: stills **and Veo-rendered video**, every deliverable measured against brand, legal and spec rules on the rendered pixels.

Everything in this document is real output from the working application — the screenshots were taken from live runs, not mockups.

EVERYTHING, ONE PLACE

The live page (this document, plus the video walkthrough) dougmkay.info/assessment

Video walkthrough of the app dougmkay.info/fde/instructions-final.mp4

Source code - GitHub repository github.com/DougMckay2389/creative-automation-pipeline

App download (9.7 MB, works out of the box) dougmkay.info/fde/creative-automation-app.zip

Full portfolio dougmkay.info

THE RUN IN NUMBERS

18 CREATIVES PER RUN	<=2 GENERATIVE CALLS	3x3 MARKETS X RATIOS	8 COMPLIANCE RULES	0 KEYS NEEDED TO RUN
--------------------------------	-----------------------------------	--------------------------------	------------------------------	--------------------------------

Two products, three markets (en-US, ja-JP, de-DE), three aspect ratios. One product ships an approved photo — it is reused, never regenerated. The other is generated once at master resolution; every deliverable is composed locally from that one call. The automated test suite runs offline with no API key and no pytest.

WHAT THE ASSESSMENT ASKED FOR

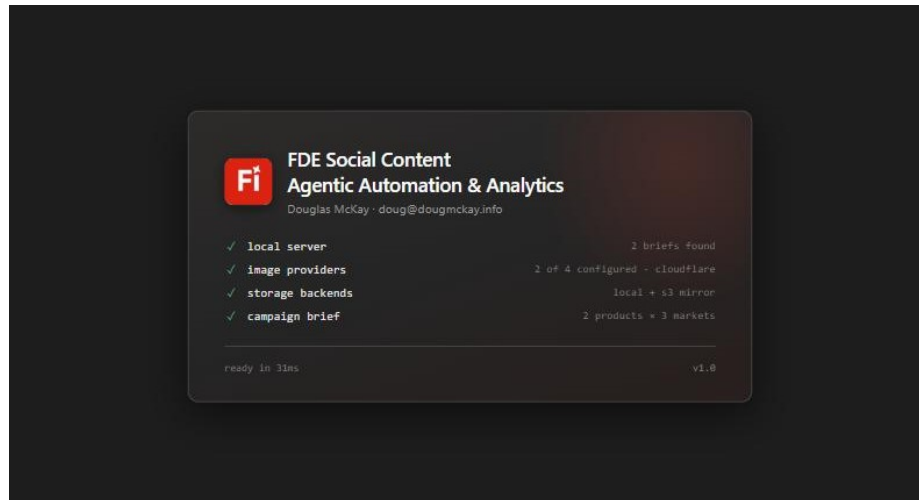
The take-home pipeline — creative-automation PoC: brief to localized, checked social creatives, with reuse-first cost logic, brand/legal gates and S3 mirroring. Covered by this document, the live page and the repository.

HOW TO USE IT — STEP BY STEP

Run **start.bat** (Windows) or **start.sh** (macOS/Linux). It checks three dependencies, starts a local server and opens the tool as its own application window. No key is needed — the offline provider renders real pixels deterministically; add a key and the same brief runs against a real model.

1 · Launch — a real startup sequence

The splash lists what is actually initialising — server, providers, storage, brief. Each line ticks when its request returns and turns red with the reason if it fails, so “none configured — offline renderer only” is said out loud instead of discovered mid-run.



Four real checks — 2 briefs found, 2 of 4 providers configured, local + S3 mirror, 2 products x 3 markets.

2 · The brief — form and YAML, same file

Products, markets with locale-aware regions and audiences, aspect-ratio presets, prohibited terms. The form is a view over the YAML — every edit regenerates the file, parsed server-side by the same PyYAML the pipeline imports. Each product carries a three-position switch: **photo as shot** (0 calls), **photo + new surface** (the approved shot goes to the model as a reference image — only the scene changes), or **generate product**.

The screenshot displays the Adobe Firefly interface, specifically the 'Brief' tab. At the top, there are tabs for 'Brief', 'Viral Engine', 'Internal Engine', 'Pipeline', 'Results', and 'Assets'. A 'Run pipeline' button and a GitHub icon are in the top right.

CAMPAIGN BRIEF

- CAMPAIGN BRIEF:** campaigns/aurora-spring.yaml
- IMAGE PROVIDER:** AI keys
- IMAGE MODEL:** Nano Banana 2 - Google, via Gateway — default
- VIDEO MODEL:** Veo 3.1 Fast (cheaper, quicker) - Google, via Gateway — default
- ASSET STORAGE:** AWS S3
- GENERATION:** ☐ REGENERATE EVERY RUN

SURFACE PROMPT what each product sits on — edit, then press Run

- HydraGlow Serum: volcanic black rock with soft water droplets and warm light (new surface)
- Velvet Matte Lip: brushed slate stone (in prompt)

BRIEF [campaigns/aurora-spring.yaml] **Form** **YAML**

CAMPAIGN

- ID:** aurora-spring-2026
- Name:** Aurora Spring Refresh
- Brand:** Aurora
- Default message:** Your skin, wide awake.

Markets without their own line fall back to this.

PRODUCTS — choose what happens to each one: keep the photo, keep it and rebuild the scene around it, or generate the product outright

HydraGlow Serum

Photo as shot **Photo + new surface** Generate product

The approved photograph goes to the model as a reference. The surface prompt describes the new scene; the product prompt names what must not change. One paid call.

SURFACE PROMPT — WHAT IT SITS ON

volcanic black rock with soft water droplets and warm light

PRODUCT PROMPT — THE OBJECT ITSELF

a frosted glass serum dropper bottle with a matte white cap

Generate now makes one image, saves it to your assets, and points this product at it

SOURCE IMAGE

campaigns/assets/hydra-glow-serum.png

Upload image... Choose existing...

Velvet Matte Lip

Photo as shot Photo + new surface **Generate product**

No source image, so the product is generated from both prompts. One paid call.

SURFACE PROMPT — WHAT IT SITS ON

brushed slate stone

PRODUCT PROMPT — THE OBJECT ITSELF

a slim matte lipstick bullet, cap off, angled upright

Generate now makes one image, saves it to your assets, and points this product at it

SOURCE IMAGE

campaigns/assets/velvet-matte-lip.png

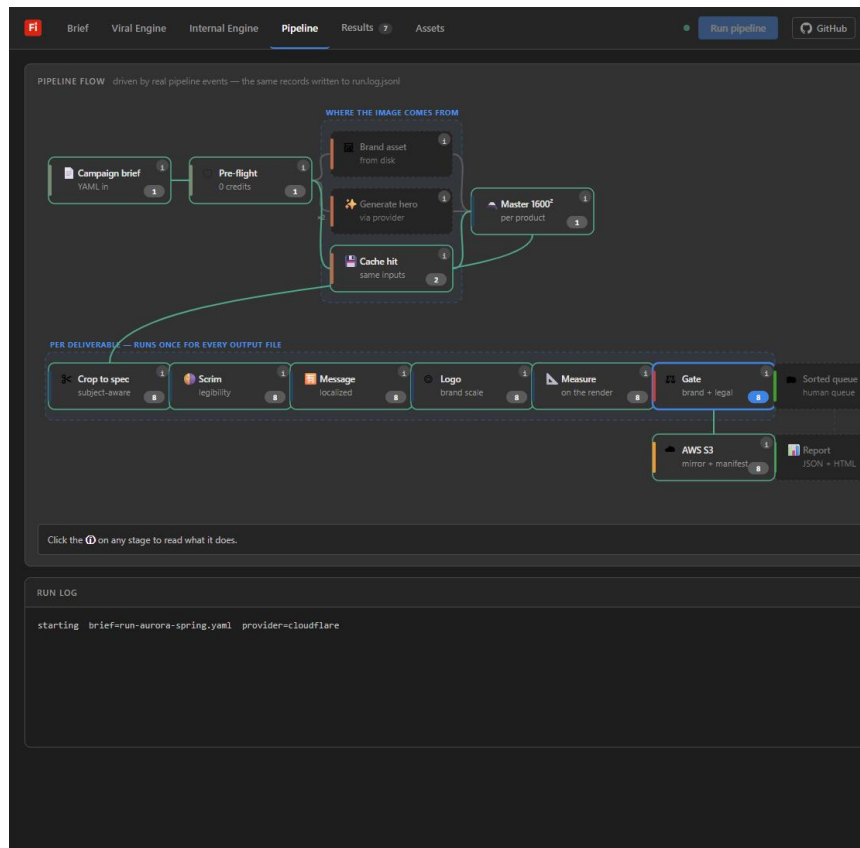
Upload image... Choose existing...

+ product

Provider and model selection up top (Veo 3.1 Fast for video), per-product resolver switch below.

3 · Run — a flow canvas driven by real events

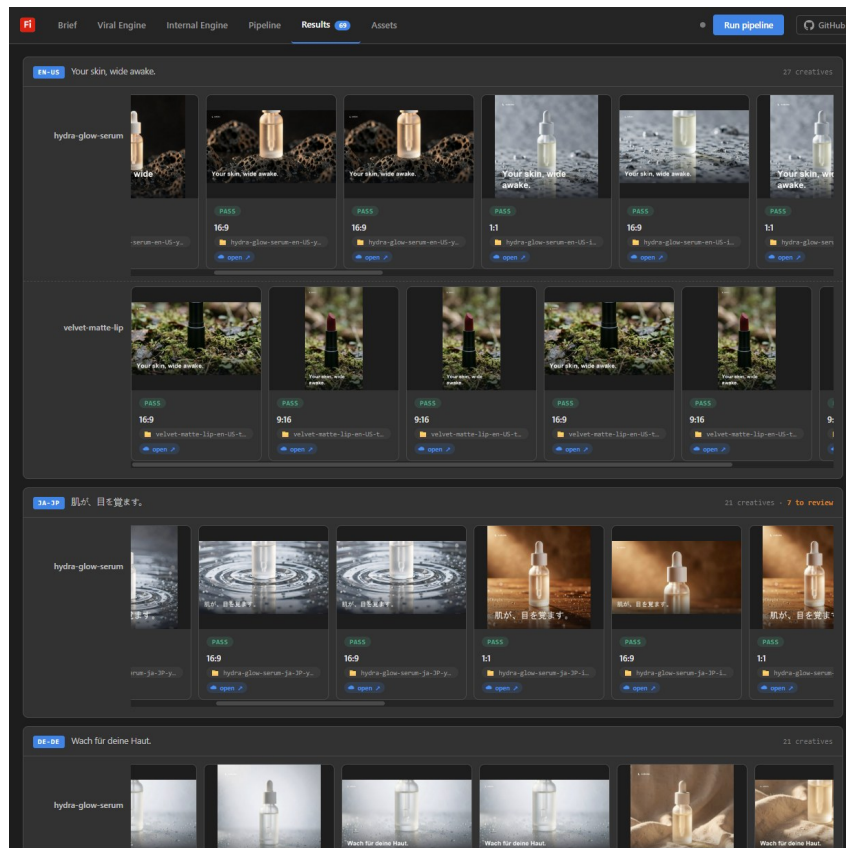
The node graph is fed by the pipeline's own event log — the same records written to `run.log.jsonl` — so nothing on screen is on a timer and a stage that did not happen cannot light up. Note the cost decision on the left: brand asset from disk, then cache, then generate — in that order.



Mid-run: 2 cache hits, zero new generative calls, deliverables moving through crop, scrim, message, logo, measure and gate, everything mirrored to S3.

4 · Results — a review queue, not an auto-approval

One row per market, each deliverable carrying a verdict and a compliance score. The stage strip shows what every compose stage did to the selected creative — and every number on it is the number the code ran on. Nothing is auto-approved and nothing fails open: blockers and majors route to named desks.



69 creatives across all three markets — en-US, ja-JP (7 routed to review) and de-DE — each tile carrying its verdict, ratio and a link to where it lives on disk.

5 · Viral Engine — cost before commitment

Pick a brief, set duration and images/videos per day, tick products. The arithmetic updates as you type, before the button that spends anything: posts per market, total posts, and the worst-case generative call count — one master per channel, every date and crop composed locally from it.

The screenshot shows the 'Viral Engine' interface with the following sections:

- Navigation:** Brief, **Viral Engine**, Internal Engine, Pipeline, Results, Assets. Buttons: Run pipeline, GitHub.
- Intro:** Viral Engine — evidence from other people's posts. Each channel is crawled for look-alike products in this market, and the top performers decide the prompt. Discovery is currently *apify* — look-alikes are real public posts, each carrying the link you can open and check. Channels: *TikTok, Instagram, YouTube*. Facebook is not offered — its scraper needs named pages and Facebook has no public keyword post search, so it could only ever have returned invented data beside three channels of real posts. Video is rendered from the approved still.
- 1 Campaign brief:**
 - Buttons: Use a brief in the repo, Paste / upload YAML.
 - Dropdown: campaigns/aurora-spring.yaml
- 2 Schedule:** (FINISHED READY)

DURATION (DAYS)	IMAGES / DAY	VIDEOS / DAY	STARTS
7	2	1	mm/dd/yyyy

21 posts per market per product · 63 in total
 Counts are the publishing rate **across** all 3 channels, not per channel.
 Generative calls: 9 at most — one master per channel, every date and crop composed from it.
- 3 Products:** (Select all)

☐ **HydraGlow Serum**
hydra-glow-serum
en-US North America · Women 25-40, skincare-led routine, shops OTTC and Sephora
ja-JP Japan · Women 20-35, Tokyo metro, layered skincare routine, value texture and subtlety
de-DE DACH · Women 28-45, ingredient-literate, price-aware pharmacy channel

☐ **Velvet Matte Lip**
velvet-matte-lip
en-US North America · Women 25-40, skincare-led routine, shops OTTC and Sephora
ja-JP Japan · Women 20-35, Tokyo metro, layered skincare routine, value texture and subtlety
de-DE DACH · Women 28-45, ingredient-literate, price-aware pharmacy channel

Run engine select at least one product

7 days x 2 images + 1 video/day = 21 posts per market per product, 63 total — from at most 9 generative calls.

6 · Viral Engine running — discovery you can audit

Per product, per market, per channel: crawl look-alike products, fold in our own channel history, emit one strategy, spend one generative call on the master, compose every slot locally, then render video slots from the approved still. The live log names every step and every fallback.

Viral Engine — evidence from other people's posts. Each channel is crawled for look-alike products in this market, and the top performers decide the prompt. Discovery is currently *apify* — look-alikes are real public posts, each carrying the link you can open and check. Channels: [TikTok](#), [Instagram](#), [YouTube](#). Facebook is not offered — its scraper needs named pages and Facebook has no public keyword post search, so it could only ever have returned invented data beside three channels of real posts. Video is rendered from the approved still.

1 Campaign brief

Use a brief in the repo | Paste / upload YAML

campaigns/aurora-spring.yaml

2 Schedule STAY READY

DURATION (DAYS) | IMAGES / DAY | VIDEOS / DAY | STARTS

7 | 2 | 1 | mm/dd/yyyy

21 posts per market per product · 126 in total
Counts are the publishing rate **across** all 3 channels, not per channel.
Generative calls: 18 at most — one master per channel, every date and crop composed from it.

3 Products Clear all

☒ **HydraGlow Serum**
HydraGlow Serum
en-US North America - Women 25-40, skincare-led routine, shops Ulta and Sephora
ja-JP Japan - Women 20-35, Tokyo metro, layered skincare routine, value texture and subtlety
de-DE DACH - Women 28-45, ingredient-literate, price-aware, pharmacy channel

☒ **Velvet Matte Lip**
Velvet Matte Lip
en-US North America - Women 25-40, skincare-led routine, shops Ulta and Sephora
ja-JP Japan - Women 20-35, Tokyo metro, layered skincare routine, value texture and subtlety
de-DE DACH - Women 28-45, ingredient-literate, price-aware, pharmacy channel

Running... 2 products x 3 markets x 3 channels

```
found ja-JP youtube 8 look-alikes via synthetic (cached)
strategy ja-JP 21 slots, 3 generative call(s)
discover de-DE tiktok
found de-DE tiktok 8 look-alikes via synthetic (cached)
discover de-DE instagram
found de-DE instagram 8 look-alikes via synthetic (cached)
discover de-DE youtube
found de-DE youtube 8 look-alikes via synthetic (cached)
```

2 products x 3 markets x 3 channels; the log records discovery per channel and each strategy's call count.

7 · The strategy — every clause carries its evidence

Each channel's prompt is assembled clause by clause, and the “Why this prompt” table says where each clause came from — a look-alike post (attributed, with its reach), the brief itself, a market trend, or the channel's own format rules. Competitor hooks are shown as reference and never used as our caption. Evidence is labelled OBSERVED or SYNTHETIC, and a failed live crawl falls back with the reason on screen instead of failing the run.

The screenshot shows the Adobe Firefly 'Viral Engine' interface. The top navigation bar includes 'Brief', 'Viral Engine', 'Internal Engine', 'Pipeline', 'Results', and 'Assets'. The 'Viral Engine' tab is active, displaying a table of prompts and their sources. The table has columns for 'FINISH', 'LIGHT', and 'FRAMING'. The 'FINISH' row shows 'dewy micro-droplets and a wet specular sheen' with a 'BRIEF' source. The 'LIGHT' row shows 'soft diffused daylight from a low side angle' with a 'TREND' source. The 'FRAMING' row shows 'tight three-quarter framing, shallow depth of field' with a 'CHANNEL' source. Below the table, there is a 'HASHTAGS' section with tags like #serum, #hydraglowserum, #skincare, and #routine. The 'Schedule' section shows a grid of video thumbnails with their respective dates, times, and scores. The thumbnails are arranged in a 2x5 grid, with the first row showing 5 thumbnails and the second row showing 4 thumbnails. Each thumbnail has a date and time in the top left corner, a 'PASS' or 'FAIL' status in the bottom left, and a score in the bottom right. The thumbnails are arranged in a grid, with the first row showing 5 thumbnails and the second row showing 4 thumbnails. Each thumbnail has a date and time in the top left corner, a 'PASS' or 'FAIL' status in the bottom left, and a score in the bottom right.

Material, finish, light and framing each cite their source; below, the dated schedule — stills and 12s Veo clips, each with its verdict and score.

8 · Internal Engine — our own history instead of the crowd

The same machinery pointed inward: nothing is crawled, and the prompt is built from what has already earned reach on our accounts. In this build that history is synthetic sample data and every row says so — a real deployment swaps one function for the platform Insights APIs named in pipeline/insights.py.

Internal Engine — evidence from our own posts. Nothing is crawled. The prompt is built from our own channel history—impressions, CTR, engagement rate, saves, comments, shares and watch-through — so it repeats what has already earned reach on our accounts rather than copying a stranger's. That history is synthetic sample data in this build and every row says so; a real deployment swaps `_engine_history()` for the platform Insights APIs named in `pipeline/insights.py`. Channels: [TikTok](#), [Instagram](#), [YouTube](#). Video is rendered from the approved still.

1 Campaign brief

Use a brief in the repo Paste / upload YAML

campaigns/aurora-spring.yaml

2 Schedule TEMPERED READY

DURATION (DAYS) IMAGES / DAY VIDEOS / DAY STARTS

7 2 1 mm/dd/yyyy

21 posts per market per product · 63 in total
Counts are the publishing rate **across** all 3 channels, not per channel.
Generative calls: 1 at most — one master per channel, every date and crop composed from it.

3 Products Select all

☐ **HydraGlow Serum**
hydra-glow-serum
en-US North America - Women 25-40, skincare-led routine, shops DTC and Sephora
ja-JP Japan - Women 20-35, Tokyo metro, layered skincare routine, value texture and subtlety
de-DE DACH - Women 28-45, ingredient-literate, price-aware, pharmacy channel

☐ **Velvet Matte Lip**
velvet-matte-lip
en-US North America - Women 25-40, skincare-led routine, shops DTC and Sephora
ja-JP Japan - Women 20-35, Tokyo metro, layered skincare routine, value texture and subtlety
de-DE DACH - Women 28-45, ingredient-literate, price-aware, pharmacy channel

Run engine select at least one product

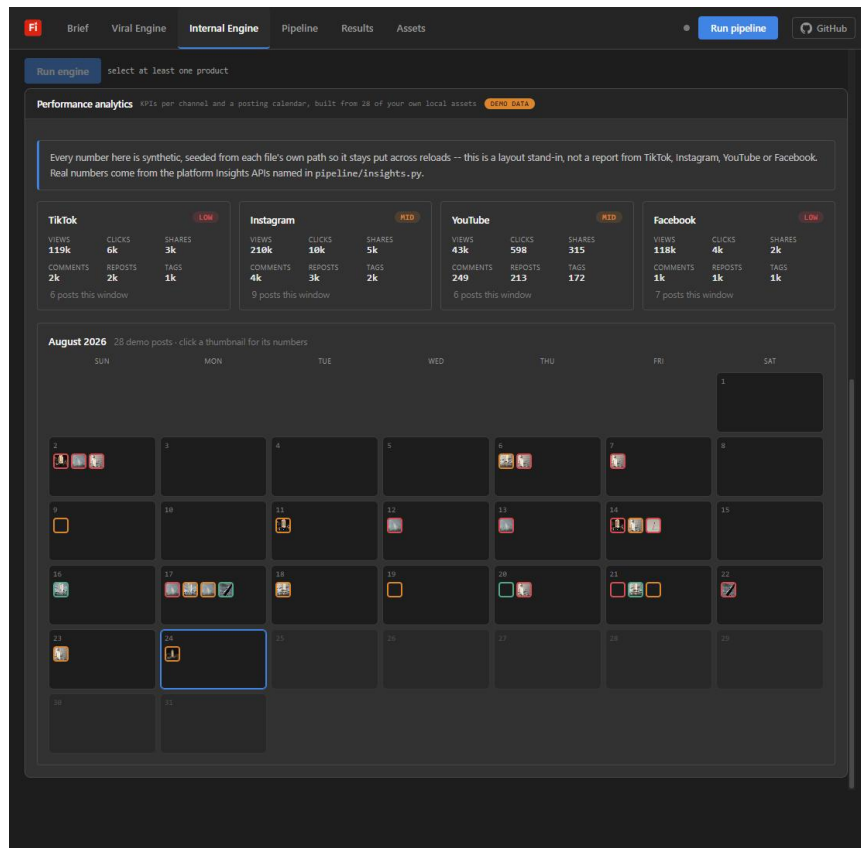
Performance analytics KPIs per channel and a posting calendar, built from 28 of your own local assets More data

Every number here is synthetic, seeded from each file's own path so it stays put across reloads — this is a layout stand-in, not a report from TikTok, Instagram, YouTube or Facebook. Real numbers come from the platform Insights APIs named in `pipeline/insights.py`.

Same brief, schedule and products flow, driven by our own channel history.

9 · Performance analytics — the loop closes

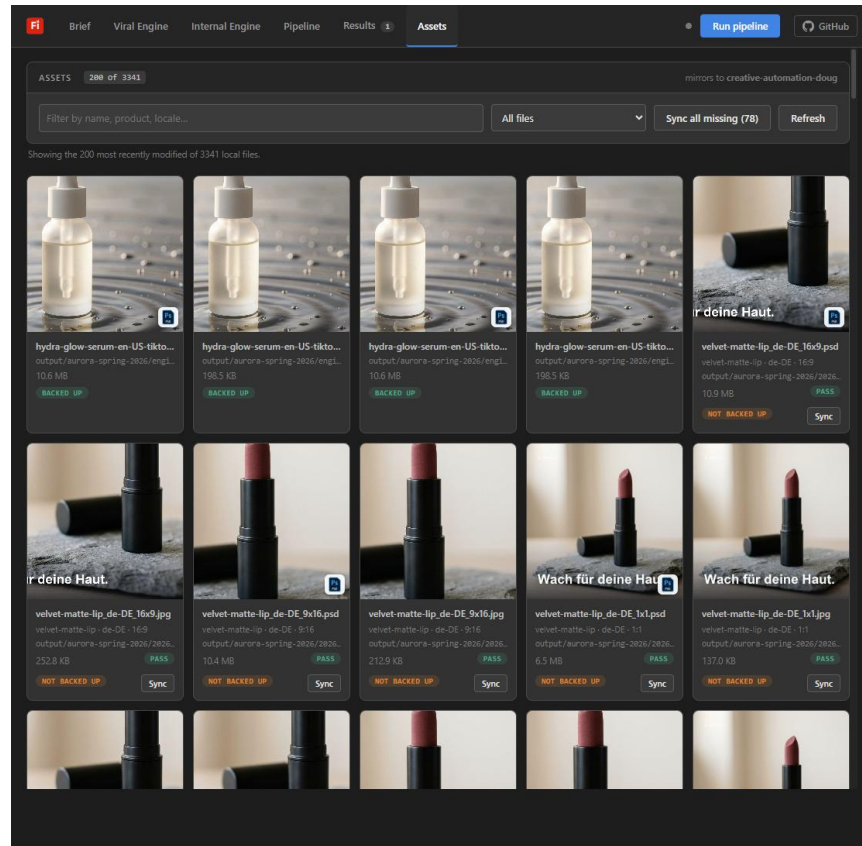
Per-channel KPI cards and a posting calendar built from the asset library. Posts are tiered high / mid / low by engagement, colour-coded and labelled. The banner does the honest work: these numbers are seeded demo data standing in for the Insights integration, and the page says so on its face.



Views, clicks, shares, comments, reposts and tags per channel; the calendar shows 28 demo posts colour-coded by performance tier.

10 • Assets — everything produced, in one library

Every produced file with its verdict, size and S3 backup state — layered .psd beside every flattened creative so the copy stays editable, one-click sync for anything not yet mirrored.



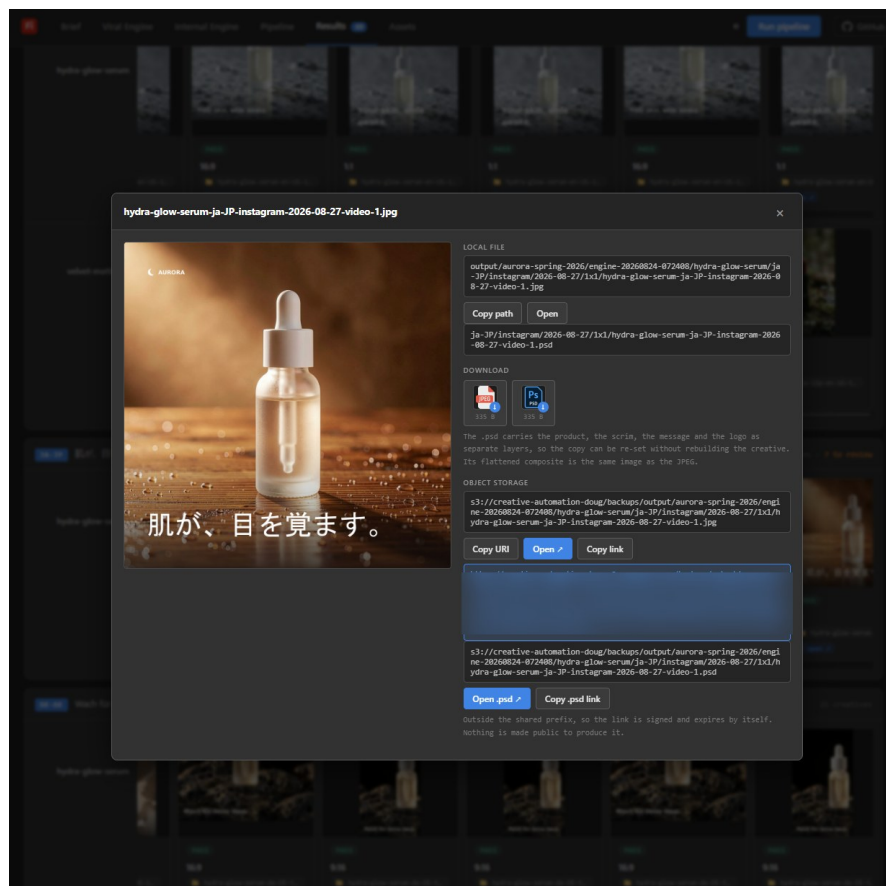
3,341 local files indexed; PASS verdicts, PSD badges, and per-file backup state against the S3 mirror.

READING THE AUDIENCE OUT OF THE BRIEF

Every market in the YAML carries more than a language code — it names who the campaign is for, written by someone who knows the market:

```
markets:
- locale: ja-JP
  region: Japan
  audience: "Women 20-35, Tokyo metro, layered skincare routine, value texture and subtlety"
  message: "肌が、目を覚ます。"
```

That one stanza steers the whole system. Discovery crawls **Japan** for look-alike products reaching **that audience**; the strategy weights warm, tactile, subdued surfaces because the audience line asks for **texture and subtlety** rather than a hard sell; and the compositor sets the message the brief's author wrote in Japanese — with the font resolver verifying it can draw **every character** before a single pixel ships, so the failure mode of localized creative (tofu boxes where the copy should be) cannot reach a deliverable. The result below is the Instagram 1:1 slot: the same approved product, re-read for a Tokyo audience.



Native Japanese copy set on a warm, textural scene — and the detail view carries the layered .psd (copy stays editable) and the S3 backup beside the flattened JPEG, with a signed, expiring share link.

WHAT THE TWO ENGINES MEASURE

Viral Engine — external evidence

Each channel is crawled for look-alike posts in the target market (TikTok, Instagram, YouTube — via hosted Apify actors, with a deterministic synthetic floor). From each post the engine keeps only what a caption and its counters can honestly give:

REACH	views per post, used to weight which look-alikes matter
ENGAGEMENT	engagement rate per post (likes + comments + shares over views)
FORMAT	image vs video share among top performers — decides what leads the channel
RATIO	which aspect ratio leads (9:16 vs 1:1 vs 16:9) and the produce order
CADENCE	posting rhythm of the winners, folded into the schedule
HOOKS	wording of top captions — attributed reference, never used as our copy
TRENDS	fastest-moving market terms, week-over-week velocity, virality score

A scraped caption cannot describe its own art direction, so look-alikes decide **format, ratio priority, cadence and hook style** — the surface treatment comes from our own measured history unless a caption literally names a material. Every strategy clause cites its source and whether it was OBSERVED or SYNTHETIC.

Internal Engine — our own history

Nothing is crawled. The strategy is built from our own per-channel history — so it repeats what earned reach on our accounts rather than copying a stranger's:

IMPRESSIONS	delivered reach per post and per placement
CTR	click-through rate per creative treatment
ENGAGEMENT	engagement rate by placement — our best 9:16 vs 1:1 vs 16:9
SAVES/SHARES	the compounding signals, kept separate from likes
WATCH-THROUGH	video completion, which decides whether video leads at all

The **Performance analytics** section closes the loop the brief's fifth business goal asks for — per-channel KPI cards (views, clicks, shares, comments, reposts, tags) and a posting calendar tiered high/mid/low by engagement. **Said on-screen, not in a footnote:** in this build the channel history and analytics numbers are synthetic and labelled DEMO DATA; pipeline/insights.py names the real platform Insights API a production deployment would call for each network.

KEY DESIGN DECISIONS

1 · Generate once, compose everywhere

One master per product that lacks an asset; every deliverable is cropped, scrimmed, localized and branded locally — 18 files from at most 2 calls. At Firefly Services' documented 4 requests/minute, a wasted call is a wasted minute.

2 · Everything vendor-shaped sits behind an adapter

Image providers (offline mock, Cloudflare Workers AI, Gemini, Firefly Services v3 async), storage (local + hand-rolled SigV4 S3 mirror, no boto3) and discovery (Apify, Playwright, synthetic) are swappable by flag, not refactor.

3 · Compliance is measured on the render

Logo presence, clearspace, palette distance, message legibility and delivered pixels are checked on the composed file, not the YAML. Nothing is auto-approved; nothing fails open — if a rule raises, the asset routes to review.

4 · Localization is a font problem first

The brief carries native copy per market; the failure mode that ships is tofu. The font resolver verifies the face can draw every character in the string (ja-JP ships in Japanese) and raises rather than rendering empty boxes.

5 · Reproducibility is a compliance property

Seeds derive from the variant id — the same brief regenerates the same pixels. Verified against the live endpoint: same seed, byte-identical images.

6 · Honesty over polish

Synthetic data says SYNTHETIC on screen. A failed scraper falls back and prints why. A resurfaced product counts as a paid call. The README leads with assumptions and limitations — most of them are where a real engagement starts.

Douglas McKay · doug@roifirm.com · (949) 667-2732 · linkedin.com/in/doug-mckay-721b143b · dougmackay.info

The Adobe Firefly icon appears in the app because this is an exercise submitted to Adobe for a Firefly Services role. Adobe Firefly and the Firefly mark are trademarks of Adobe Inc.; this project is not affiliated with, sponsored by or endorsed by Adobe.