**финальный комплект для Голиба** (всё проверено и собрано), чтобы он **вшил в ИИ** и сдал Phase 1.2–1.3 без дыр. Копируй как есть.

# 0) Что именно встроить (итог по Phase 1.2–1.3)

* Twitter **Search + Timeline** (приоритет: **Tier A → Search → Tier B**) с бэкфиллом.
* Антибот/фильтры + **regex** из п.1.
* Конструктор **Search**-запросов (п.2) + минус-слова.
* **Tier A/B** CSV, хот-апдейт, CELEB\_OVERRIDE (п.3).
* Политика деградации/ретраев/алертов (п.4).
* Контракт /ingest/social\_alert **→ Алекс** (п.5).
* Метрики/логгинг (п.6).
* LLM-цепочка (Aegis→DeepSeek→MythoMax) (п.7).
* **Chroma KB**: запись, метаданные, снапшоты (п.8).
* Docker + Compose + .env + config.json (п.9).
* Набор **curl-тестов** (п.10).

## 1) Regex-пакет (PCRE/Python OK)

**Позитивные триггеры**

(?i)\b(just\s+launched|new\s+(token|coin)|created\s+(on|at)\s+(raydium|jupiter)|new\s+pool|pool\s+created|listing\s+(soon|today)|live\s+now|go\s+live)\b

(?i)\b(pump(ing)?|moon(ing)?|spike|break(out)?|parabolic|send\s+it|ath|0\sto\s+100)\b

(?i)\b(raydium|jupiter|orca|pump\.fun|amm|dex|lp|liquidity|mint(ed)?|jito\s+mev|sniper|bot)\b

(?i)\b(og\s+(token|coin)|original\s+(token|coin)|first\s+(token|coin)|the\s+only\s+official)\b

**Исключения (шум)**

(?i)\b(airdrop|pre[-\s]?sale|whitelist|wl\s+spot|give\s?away|free\s+mint|bounty|invite\s+link|casino|forex|copytrade|signals?\s+group)\b

**Solana mint/base58 (32–44)**

\b[1-9A-HJ-NP-Za-km-z]{32,44}\b

**DEX-баннер/реклама**

(?i)\b(promoted\s+by|sponsored|ad|paid\s+banner|promote)\b.\*\b(dex|raydium|jupiter|pump\.fun)\b

**Смена имени (эвристика)**

(?i)\b(previous(ly)?\s+known\s+as|aka|renamed\s+from)\b

Логика: **name\_change\_count ≤ 2**, иначе → spam\_flag=true.

## 2) Twitter Search — готовые шаблоны

**Поддержать синтаксис:** OR, AND, (), кавычки "...", #tag, @handle, from:, url:, has:links, -слово, lang:  
**Лимит строки:** ≤ **480** символов  
**Минус-слова (единый список):**  
airdrop,presale,whitelist,wl,giveaway,free mint,bounty,casino,forex,signals group,copytrade

**2.1 Broad (recall):**

(pepe OR doge OR "cat ceo" OR "elon meme" OR "trump coin" OR #memecoin)

(pump OR moon OR "new pool" OR "just launched" OR "created on raydium" OR "go live" OR pump.fun)

(#solana OR raydium OR jupiter OR orca)

-airdrop -presale -whitelist -giveaway -bounty -copytrade -"signals group"

lang:en

**2.2 Tight (precision):**

("just launched" OR "new pool" OR "created on raydium" OR "listing today" OR "live now")

(#solana OR raydium OR jupiter)

(pepe OR doge OR cat OR frog)

(from:tierA\_trader OR from:cryptoOG OR @elonmusk)

-airdrop -presale -whitelist -giveaway

**2.3 Locales (по необходимости):**

("только что запустили" OR "новый пул" OR "листинг сегодня") (#солана OR raydium OR jupiter) -airdrop -раздача lang:ru

("listado hoy" OR "nuevo pool" OR "acaba de lanzar") (#solana OR raydium OR jupiter) -sorteo -airdrop lang:es

## 3) Tier A/B — CSV + кэш

**Файлы:** data/tierA.csv, data/tierB.csv  
**Заголовок:**

user\_id,handle,tier,weight,celeb,notes,last\_seen

**Пример:**

44196397,@elonmusk,A,1.0,1,celeb override,2025-09-05T10:00:00Z

12345678,@tierA\_trader,A,0.9,0,alpha caller,2025-09-04T21:15:00Z

87654321,@memelord99,B,0.7,0,meme curator,2025-09-04T21:15:00Z

* user\_id резолвим при загрузке → кэш data/tw\_id\_cache.json.
* Хот-апдейт: **каждые 6 часов** + ручной /admin/reload\_tiers.
* **CELEB\_OVERRIDE=true** для Tier A с celeb=1.

## 4) Деградация / ошибки / алерты

* Backoff: **base 2s, max 120s, retries 5**.
* **10 минут** даунтайма: **деградированный режим** — Search OFF, Timeline Tier A с интервалом ×4; складываем задания в очередь и реплеим.
* Допустимая потеря цикла: **≤ 5%**.
* **Алерты**: Telegram ops\_social\_alerts + email (JSON-краткий), уровни WARN/ERROR, rate ≤ 1/мин.

## 5) Контракт к Алексу /ingest/social\_alert

* **Stage:** https://alex.stage/api/ingest/social\_alert
* **Prod:** https://alex.prod/api/ingest/social\_alert
* **Auth:** Authorization: Bearer INTERNAL\_TOKEN
* **Идемпотентность:** alert\_id в теле + заголовок X-Idempotency-Key: <alert\_id>
* **Лимиты:** body ≤ **256 KB**, ≤ **5 req/sec**, SLA ≤ **1.5s**

**Required:**  
alert\_id, created\_at, profile(twitter|pump|find|listings), focus(keyword|token|event), confidence(0..1), zscore, sources[]({url,handle,tier,ts}), summary, decision\_hint(BUY|WATCH|NO)

**Optional:**  
token\_guess{name,mint?}, og, clones, dex\_banner, name\_changes, sentiment(-1..1), media[], trace\_id, debug

**Пример тела:**

{

"alert\_id": "sa\_2025-09-06\_12-31-07\_001",

"created\_at": "2025-09-06T12:31:07Z",

"profile": "twitter",

"focus": "keyword",

"confidence": 0.82,

"zscore": 3.1,

"summary": "Tier A упомянул DOGE2025; всплеск упоминаний; TG подтверждает.",

"sources": [

{"url":"https://twitter.com/...","handle":"@tierA\_trader","tier":"A","ts":"2025-09-06T12:29:40Z"}

],

"token\_guess": {"name":"DOGE2025"},

"og": false,

"clones": 3,

"dex\_banner": true,

"name\_changes": 1,

"sentiment": 0.44,

"decision\_hint": "WATCH",

"trace\_id": "trc\_f92a"

}

## 6) Метрики / логи

* GET /api/golib/health
* Счётчики: pulled, accepted, alerts\_sent, forwarded\_ok, forwarded\_fail
* Тайминги: fetch\_ms, filter\_ms, ai\_ms, forward\_ms
* GET /metrics (Prometheus)
* Логи: **JSONL**, ключи: ts, trace\_id, alert\_id, request\_id, source\_kind(search|timeline), tier, stage, \*\_ms, status

**Пример лог-строки:**

{"ts":"2025-09-06T12:31:08Z","trace\_id":"trc\_f92a","stage":"forward","alert\_id":"sa\_2025-09-06\_12-31-07\_001","forward\_ms":143,"status":"accepted"}

## 7) LLM-цепочка

* Порядок: **Aegis → DeepSeek → MythoMax** (авто-fallback).
* Таймауты: **6s / 8s / 10s**.
* Пороги: has\_token=0.70, will\_token=0.60.
* Параметры: max\_tokens=1024, temperature=0.2, max\_calls\_per\_event=3, daily\_budget\_usd=3.

## 8) Chroma KB

* Запись: **все события** c флагом status: accepted|rejected.
* Метаданные (vector metadatas):  
  id,hash,author,created\_at,tickers[],zscore,memeability,decision\_hint,confidence,tier,og,dex\_banner,name\_changes,source,urls[]
* Ретенция: **90 дней** или **≤ 5M векторов** (TTL + LRU).
* Dev-поиск (фильтры): created\_at range, tier∈{A,B}, zscore≥, memeability≥, tickers[], og, dex\_banner.
* Экспорт снапшота: **ежедневно 00:00 UTC** → data/kb\_snapshots/YYYY-MM-DD/ (Parquet + JSONL).

## 9) Конфиг / .env / Docker

.env

PORT=8080

INTERNAL\_TOKEN=replace\_me

DATAIMPULSE\_TOKEN=replace\_me

GROQ\_API\_KEY=replace\_me

CHROMA\_URL=http://chroma:8000

CHROMA\_COLLECTION=social\_kb

ALEX\_INGEST\_URL=https://alex.stage/api/ingest/social\_alert

TWITTER\_MODE=both

SEARCH\_BACKFILL\_HOURS=24

TIMELINE\_BACKFILL\_HOURS=48

TIMELINE\_MAX\_TWEETS=200

LANG\_FILTER=none

RETRY\_BASE\_SEC=2

RETRY\_MAX\_SEC=120

RETRY\_COUNT=5

DEGRADED\_AFTER\_MIN=10

LOSS\_TOLERANCE\_PCT=5

config.json

{

"tiers": {

"files": ["data/tierA.csv", "data/tierB.csv"],

"hot\_reload\_hours": 6,

"celeb\_override": true

},

"search": {

"max\_query\_len": 480,

"minus\_words": ["airdrop","presale","whitelist","wl","giveaway","free mint","bounty","casino","forex","signals group","copytrade"]

},

"llm": {

"order": ["aegis","deepseek","mythomax"],

"timeouts\_sec": {"aegis": 6, "deepseek": 8, "mythomax": 10},

"thresholds": {"has\_token": 0.7, "will\_token": 0.6},

"max\_tokens": 1024,

"temperature": 0.2,

"max\_calls\_per\_event": 3,

"daily\_budget\_usd": 3

},

"kb": {

"write\_policy": "all\_with\_flag",

"retention\_days": 90,

"max\_vectors": 5000000,

"metadata\_fields": ["id","hash","author","created\_at","tickers","zscore","memeability","decision\_hint","confidence","tier","og","dex\_banner","name\_changes","source","urls"]

},

"forward": {

"alex\_url\_stage": "https://alex.stage/api/ingest/social\_alert",

"alex\_url\_prod": "https://alex.prod/api/ingest/social\_alert",

"auth\_header": "Authorization",

"token": "INTERNAL\_TOKEN",

"idempotency\_header": "X-Idempotency-Key",

"rate\_limit\_rps": 5,

"max\_body\_kb": 256,

"sla\_sec": 1.5

}

}

requirements.txt

fastapi==0.112.0

uvicorn[standard]==0.30.0

httpx==0.27.0

pydantic==2.8.2

regex==2024.5.15

orjson==3.10.7

pandas==2.2.2

numpy==1.26.4

chromadb==0.5.0

tenacity==8.3.0

prometheus-client==0.20.0

Dockerfile

FROM python:3.11-slim

WORKDIR /app

COPY requirements.txt .

RUN pip install -r requirements.txt

COPY . .

ENV PYTHONUNBUFFERED=1

EXPOSE 8080

CMD ["python","-m","golib\_api.server"]

docker-compose.yml

version: "3.9"

services:

golib:

build: .

env\_file: .env

ports: ["8080:8080"]

volumes:

- ./data:/app/data

depends\_on:

- chroma

chroma:

image: chromadb/chroma:latest

ports: ["8000:8000"]

## 10) Быстрые тесты (обязательно приложить)

**Health**

curl -s localhost:8080/api/golib/health

**Reload tiers**

curl -s -X POST -H "Authorization: Bearer INTERNAL\_TOKEN" localhost:8080/admin/reload\_tiers

**Social check (demo)**

curl -s -X POST -H "Authorization: Bearer INTERNAL\_TOKEN" -H "Content-Type: application/json" \

-d '{ "request\_id":"run\_1","profile":"pump","token":{"name":"DOGE2025","mint":"D0gexxxx"}, "options":{"check\_twitter":true,"check\_telegram":true,"window\_min":30}}' \

localhost:8080/api/golib/social\_check

**Forward to Alex (stage)**

curl -s -X POST -H "Authorization: Bearer INTERNAL\_TOKEN" -H "X-Idempotency-Key: sa\_test\_001" \

-H "Content-Type: application/json" \

-d '{ "alert\_id":"sa\_test\_001","created\_at":"2025-09-06T12:31:07Z","profile":"twitter","focus":"keyword","confidence":0.82,"zscore":3.1,"summary":"test","sources":[{"url":"https://twitter.com/...","handle":"@tierA\_trader","tier":"A","ts":"2025-09-06T12:29:40Z"}],"decision\_hint":"WATCH"}' \

https://alex.stage/api/ingest/social\_alert

### Финальный контроль (что принять у Голиба)

* Архив/репозиторий с **кодом + Docker + compose + .env.example + config.json**.
* Папка data/ (Tier A/B CSV + tw\_id\_cache.json).
* Док README.md с запуском и **списком эндпоинтов**.
* Скрины/видео **E2E**: DI batch → фильтр/LLM → social\_alert → приём у Алекса (202/OK).
* metrics и health доступны.
* Демка **social\_check** по токену от Алекса (ручной вызов) работает.

Если он всё это вшивает — Алекс просто подставит токены/URL и подключит к интерфейсу.