App Idea Signal Sources: Technical Specification and Research Report

Executive Summary

1.1 Mission Statement and Strategic Context

This document serves as the definitive Technical Specification (PRD) for the construction of an automated market intelligence engine. Designed for a solo mobile app developer with a background in Ads Product Management, the system's primary objective is to programmatically identify high-signal, underserved opportunities in "unsexy" professional verticals. The core philosophy driving this architecture is the rejection of intuition-based development in favor of **Signal-Triangulation**.

By synthesizing data from three distinct, orthogonal vectors—**Quantitative Search Arbitrage** (ASO), **Qualitative Workflow Friction** (Forums), and **Regulatory Compulsion** (Government Gazettes)—the system eliminates the "market risk" typically associated with indie app development. It targets problems that users are either legally compelled to solve (Compliance) or are actively struggling to solve with inadequate tools (Friction).

The scope is rigorously defined: professional verticals (HVAC, Logistics, specialized clinics) across English, Spanish, Turkish, and French markets. The technical feasibility is paramount; selected sources must support reliable automation via API or headless browser interaction (Playwright), ensuring the resulting "Coding Agent" can operate autonomously with high uptime and low maintenance.

1.2 The Signal Triangulation Architecture

The system logic relies on a weighted scoring model that cross-references data streams. A valid "App Idea" is defined not by a single data point, but by the convergence of signals.

Signal Category	Primary Indicator	Data Source Types	Psychological Driver
Category A: Arbitrage	High Search Volume / Low App Quality	ASO Tools (AppTweak, MobileAction)	**Intent:** "I am looking for a solution, but the market offers nothing."

Category B: Friction	Keywords: "Excel," "Manual," "Workaround"	Forums (Reddit, Ekşi Sözlük, Forocoches)	**Pain:** "The current method is broken, and I am wasting time."
Category C: Compulsion	Keywords: "Mandatory," "Regulation," "Deadline"	Gazettes (BOE, EUR-Lex, Official Journals)	**Fear/Obligation:** "I must solve this or face legal penalties."
Category D: Sentiment	1-Star Reviews on Enterprise Software	B2B Reviews (G2, Capterra)	**Frustration:** "I hate the mobile experience of my mandatory desktop tool."

The system architecture is divided into three functional layers: **The Harvester** (Data Ingestion), **The Refiner** (NLP & Normalization), and **The Synthesizer** (Logic & Scoring). This report provides the granular blueprints for the *Harvester* layer, detailing every endpoint, authentication flow, and data schema required to feed the downstream logic.

---

2. Core Signal Category A: ASO Search Arbitrage

App Store Optimization (ASO) tools provide the quantitative validation of demand. Unlike web SEO, app store search volume indicates high-intent behavior: users are actively looking for software to install. The system focuses on "Search Arbitrage"—identifying keywords where user demand (Volume) significantly outstrips supply quality (Ratings/Competitiveness).

2.1 Primary Source: AppTweak

AppTweak is selected as the primary ASO intelligence source due to its high-fidelity data on "Search Popularity" (Apple's direct volume metric) and extensive historical tracking.[1] While the cost is significant, the integrity of the volume data is critical for the "Arbitrage" signal.

2.1.1 Access Method & Authentication

- **Access Type:** REST API (JSON).
- **Authentication:** API Key passed in the HTTP Header X-Apptweak-Key.
- **Budget & Constraints:** The "Small (API)" plan is priced at approximately $166/month, providing 250,000 credits.[1]
- **Credit Consumption Model:** Not all endpoints cost the same. A simple metadata

request might cost 1 credit, while a historical keyword ranking report over 90 days can cost hundreds. The system must prioritize "Current Data" snapshots to conserve budget, only requesting "Historical Data" for high-probability candidates.[3]

2.1.2 Data Schema & Technical Implementation

The critical endpoint for this system is GET /ios/keywords/metrics.json. This endpoint allows the agent to query specific terms identified in the Friction/Compulsion phases to validate search volume.

Working Code Sample (Python/Requests):

Python

```
import requests
import json
import time

class AppTweakHarvester:
    def __init__(self, api_key):
        self.base_url = "https://api.apptweak.com"
        self.headers = {"X-Apptweak-Key": api_key}

    def get_keyword_metrics(self, term, country='es', language='es'):
        """
        Fetches volume, difficulty, and competition metrics for a specific keyword.
        Cost: ~5-10 credits per call.
        """
        endpoint = f"/ios/keywords/metrics.json"
        url = self.base_url + endpoint
        params = {
            "term": term,
            "country": country,
            "language": language
        }

        try:
            response = requests.get(url, headers=self.headers, params=params)
            response.raise_for_status()
            return response.json()
        except requests.exceptions.HTTPError as e:
```

```
            if response.status_code == 429:
                print("Rate limit hit. Sleeping...")
                time.sleep(60)
                return self.get_keyword_metrics(term, country, language)
            print(f"AppTweak Error: {e}")
            return None

# Usage Example
harvester = AppTweakHarvester("YOUR_API_KEY")
data = harvester.get_keyword_metrics("control horario")
```

JSON Response Schema Analysis:
The API returns a nested JSON object. The coding agent must extract specific fields to calculate the SNR (Signal-to-Noise Ratio).

JSON

```
{
  "content": {
    "term": "control horario",
    "volume": 45,              // 0-100 scale. Apple Search Popularity.
    "difficulty": 22,          // 0-100 scale. Estimated ranking difficulty.
    "competition_score": 15,   // 0-100 scale. Density of competing apps.
    "results":
  }
}
```

2.1.3 Signal Extraction Logic

The coding agent must apply the following heuristic to the AppTweak data:

1. **Volume Validation:** volume must be > 30. Below this, the market is too niche to support a sustainable business, even for a solo developer.
2. **The "Ghost" Gap:** Calculate the Quality_Gap.
    - Iterate through the top 5 apps in the results array.
    - Calculate mean_rating.
    - **Signal:** If volume > 40 AND mean_rating < 3.5, this is a **Class A Opportunity**. It means users are searching often but finding poor solutions.
3. **Efficiency Check:** The system should *not* crawl the dictionary. It should only query AppTweak with keywords generated from Category B (Friction) and Category C (Compliance) to validate them.[2]

2.2 Secondary Source: ASODesk

ASODesk offers a competitive alternative, particularly for European and Russian markets (which often align with data structures in Turkey). Its "Traffic Score" and "Keyword Boost" features provide different perspectives on volume.[4]

2.2.1 Access Method & Authentication

- **Access:** REST API.
- **Auth:** Token-based.
- **Pricing:** The "Startup" plan ($33.2/month) offers 5,000 credits, making it a cost-effective backup to AppTweak for bulk validation tasks.[4]
- **Key Differentiator:** ASODesk allows checking "Daily Impressions" estimates, which can be more tangible than Apple's 0-100 index for financial modeling.[6]

2.2.2 Data Schema & Integration

The coding agent should utilize ASODesk specifically for **Keyword Auto-Suggestions**. This helps expand the seed keywords found in forums into a broader semantic core.

Endpoint: GET /v1/keywords/auto-suggestions
Logic: If "Excel HVAC" is found on Reddit, feed "HVAC" to ASODesk to find "HVAC load calculator," "HVAC invoice," etc..7

2.3 Secondary Source: MobileAction

MobileAction is critical for one specific metric: **Search Ads Intelligence**.

2.3.1 The "Pay-to-Play" Heuristic

MobileAction provides visibility into which apps are *buying* ads for specific keywords.[8]

- **Signal Logic:** If a keyword has High Search Volume and High Ad Spend (lots of bidding) but Low Organic App Quality, it indicates a "broken" organic market. Incumbents are surviving only by purchasing users, not by retention. A better product could disrupt this by ranking organically.
- **Access:** REST API with token parameter..[9]
- **Budget:** Custom enterprise pricing often applies, but startup tiers are available. The coding agent should use this sparingly for "Final Validation" of an idea before development starts.

2.4 Secondary Source: Appfigures

Appfigures is excellent for global rank tracking and category analysis.

2.4.1 Access Method

- **Access:** REST API via https://api.appfigures.com/v2/.

- **Auth:** OAuth 1.0a or Basic Auth with Client Key.[10]
- **Rate Limits:** 1,000 requests/day on the free tier (for tracked apps), but public data requires the "Public Data Add-on".[11]
- **Use Case:** Monitor the "Business" and "Productivity" categories in target countries (Spain, Turkey, France) for "New Releases". A sudden spike in new apps in a specific sub-niche often correlates with a new regulation (Category C).[12]

---

3. Core Signal Category B: Workflow Friction & "Manual Silos"

This category focuses on the qualitative detection of pain. We are looking for the "exhaust fumes" of inefficient workflows: complaints, requests for templates, and discussions about manual data entry.

3.1 Primary Source: Reddit (Global/English)

Reddit is the "front page of the internet" and a goldmine for unfiltered user feedback. Despite recent API pricing changes, it remains the most structured source for forum data.

3.1.1 Access Method & Authentication

- **Method:** PRAW (Python Reddit API Wrapper).
- **Auth:** OAuth2 (client_id, client_secret, user_agent).
- **Budget/Limits:** The free tier allows for 60 requests/minute, which is sufficient for a research agent if queries are paced correctly. Commercial use policies must be adhered to, but "internal research" generally falls under acceptable use if data isn't resold.[13]

3.1.2 Data Schema & Target Subreddits

The coding agent must monitor specific "unsexy" subreddits where professionals gather.

- **Targets:** r/supplychain, r/HVAC, r/construction, r/smallbusiness, r/excel, r/nursing, r/truckers.
- **Search Patterns:** The agent should search for "problem-solution" clusters.

Working Code Sample (Python/PRAW):

Python

```
import praw
import pandas as pd
```

```python
class RedditFrictionHunter:
    def __init__(self, client_id, client_secret, user_agent):
        self.reddit = praw.Reddit(
            client_id=client_id,
            client_secret=client_secret,
            user_agent=user_agent
        )
        self.friction_keywords = [
            "spreadsheet", "excel template", "manual entry",
            "paperwork", "writing down", "clipboard", "google sheets"
        ]

    def scan_subreddit(self, sub_name, limit=100):
        subreddit = self.reddit.subreddit(sub_name)
        opportunities =

        # Construct a query for friction keywords
        query = " OR ".join(self.friction_keywords)

        for post in subreddit.search(query, sort='new', time_filter='year', limit=limit):
            # SNR Filter: Only consider posts with engagement
            if post.num_comments > 5:
                opportunities.append({
                    "title": post.title,
                    "text": post.selftext[:500], # First 500 chars for context
                    "score": post.score,
                    "url": post.url,
                    "comments": post.num_comments
                })
        return pd.DataFrame(opportunities)

# Usage
hunter = RedditFrictionHunter("ID", "SECRET", "ResearchBot/1.0")
df = hunter.scan_subreddit("HVAC")
```

3.1.3 Signal Extraction Logic

The extraction logic must differentiate between *sharing* a solution and *begging* for one.

- **Positive Signal (Demand):** "Does anyone have a spreadsheet for..." / "I'm tired of manually entering..." / "Is there an app for..."
- **Negative Signal (Supply):** "I made a spreadsheet..." / "Here is a free template..." (Unless the comments are "This is great, but I wish it was an app", in which case it flips back to

Positive).
- **Regex Pattern:** (looking for|need|requesting).*(template|spreadsheet|software).[15]

3.2 Source: Stack Exchange API

Stack Exchange (specifically specific communities like "Software Recommendations" or "Workplace") allows for precise querying of technical hurdles.

- **Access:** REST API.
- **Auth:** Key required for higher quotas (10,000/day).
- **Endpoint:** /search/advanced
- **Query:** q="no library for" OR "manual workaround".
- **Logic:** Look for questions where the accepted answer is "You have to do this manually" or "There is no tool for this." This defines a "Greenfield" opportunity.[16]

3.3 Source: Ekşi Sözlük (Turkey)

Ekşi Sözlük is a hypertext dictionary and the primary forum for Turkish socio-political and professional discourse. It does not offer a public API, necessitating HTML scraping.

3.3.1 Access Method (Scraping)

- **Method:** Playwright (Python) or httpx with robust User-Agent rotation.
- **Feasibility:** Moderate. Cloudflare protection is present but often passable with "Stealth" plugins.
- **Structure:** Entries (content) are nested under Titles (topics). URL structure: https://eksisozluk.com/{topic-slug}--{topic-id}.[17]

3.3.2 Signal Logic & Keywords

- **Keywords (TR):** "amk uygulaması" (damn app - generic complaint), "çöküyor" (crashing), "manuel giriş" (manual entry), "excel tablosu" (excel table).
- **Heuristic:** Users use Ekşi to vent. Search for bank apps, government gateways ("e-devlet"), or professional tools. A title like "gümrük müşavirliği" (customs consultancy) containing entries complaining about paperwork ("evrak işi") indicates a Type B signal.
- **Library:** Unofficial libraries like eksisozluk-scraper on GitHub provide guidance on DOM structure, but a custom Playwright implementation is recommended for stability.[19]

3.4 Source: Forocoches (Spain)

Forocoches is a massive, semi-closed Spanish forum. While registration is invite-only, reading is often open or accessible via cache. It is the hub for "Autónomos" (freelancers) and tradesmen.

3.4.1 Access Method

- **Method:** Playwright with stealth-plugin.

- **Constraint:** Aggressive Cloudflare and IP blocking. Low frequency crawling is essential.
- **Target Sections:** "Empleo" (Employment) and "Informática".[21]

3.4.2 Signal Logic

- **Keywords (ES):** "gestoría" (agency), "factura" (invoice), "trimestre" (quarterly tax filing), "hoja de cálculo" (spreadsheet).
- **Context:** Spanish freelancers often discuss how to dodge "Hacienda" (Tax Agency) complexities or manage clients. Threads with high page counts asking "how do you organize X" are prime targets.

3.5 Source: Meneame (Spain)

Meneame is a Digg-style news aggregator. It is open-source (code on GitHub), which makes its API structure transparent.

3.5.1 Access Method & Schema

- **Access:** API (historically open) or Scraping.
- **Base URL:** https://meneame.net/api/list.php (Legacy) or via analyzing the open-source repo Meneame/meneame.net to find current internal endpoints.[23]
- **Signal:** News stories about *new laws* (Category C triggers) often appear here first. The *comments* section (Category B) will contain the immediate reaction: "This is going to be a nightmare to manage." Scrape the comments for "cómo se hace" (how do you do this).[25]

---

4. Core Signal Category C: Regulatory & Compliance Triggers

This is the most powerful signal category. "Compulsion" creates non-negotiable markets. If a government mandates "Digital Waybills" for transport, every truck driver *must* have an app.

4.1 Primary Source: BOE (Boletín Oficial del Estado - Spain)

The Spanish Official Gazette offers a pristine Open Data API, making it the highest-feasibility source in this category.

4.1.1 Access Method & Authentication

- **Access:** REST API (Open Data).
- **Auth:** None required for public access.
- **Documentation:** https://www.boe.es/datosabiertos/api/api.php.[26]
- **Formats:** XML and JSON.

4.1.2 Data Schema & Extraction Logic

The coding agent should target the **Legislación Consolidada** or daily **Sumario**.

Endpoint: https://boe.es/datosabiertos/api/boe
Parameters: fecha (YYYYMMDD).
Python Implementation Logic:

Python

```
import requests
import xmltodict

def scan_boe_summary(date_str):
    url = "https://boe.es/datosabiertos/api/boe"
    params = {'fecha': date_str}

    response = requests.get(url, params=params)
    if response.status_code == 200:
        data = xmltodict.parse(response.content)
        # Navigate to General Provisions (Disposiciones Generales)
        disposiciones = data.get('sumario', {}).get('diario', {}).get('seccion',)

        # Iterate and filter
        # Logic to extract 'titulo' and 'texto'
        # Check for keywords: 'Registro', 'Digital', 'Obligatorio', 'Sede Electrónica'
```

4.1.3 Idea Generation Heuristic

- **Trigger Word:** "Entrada en vigor" (Entry into force).
- **Scenario:** A law is published today but enters into force in 6 months. This is the **Goldilocks Window** for development.
- **Keywords:** "Reglamento" (Regulation), "Real Decreto" (Royal Decree), "Transposición" (Transposition of EU law).

4.2 Primary Source: EUR-Lex (European Union)

EUR-Lex covers EU Directives that eventually trickle down to national laws (Spain, France).

4.2.1 Access Method

- **Method: Cellar API** (REST) or **SPARQL** endpoint. The Cellar API is preferred for structured metadata retrieval.[27]
- **Auth:** Registration required for webservice access.

- **Endpoints:** Allows complex filtering by sector (e.g., Sector 3 = Legal Acts) and date.

4.2.2 SPARQL Query Strategy

The coding agent should utilize SPARQL to find "Directives" with approaching deadlines.

Code snippet

```
PREFIX cdm: <http://publications.europa.eu/ontology/cdm#>
SELECT?work?title?date WHERE {
 ?work cdm:work_has_resource-type
<http://publications.europa.eu/resource/authority/resource-type/DIR>.
 ?work cdm:work_date_document?date.
 ?work cdm:work_has_title?title.
  FILTER (lang(?title) = "en")
  FILTER (?date > "2025-01-01"^^xsd:date)
}
LIMIT 100
```

- **Logic:** Identify Directives related to "Reporting," "Tracking," or "Compliance." These are upstream signals for future national apps.[29]

4.3 Source: Chorus Pro & PISTE (France)

Chorus Pro is the mandatory e-invoicing gateway for the French public sector.

4.3.1 Access Method

- **Platform: PISTE** (Plateforme d'Intermédiation des Services pour la Transformation de l'État).
- **Auth:** OAuth 2.0 Client Credentials flow. Requires creating a "Technical Account".[30]
- **Documentation:** https://communaute.chorus-pro.gouv.fr/documentation/specifications-externes/.[32]

4.3.2 Signal Extraction

- **Target:** The API documentation itself is the signal. Changes in the API specs (e.g., "New mandatory field for 2026 B2B e-reporting") serve as the trigger.
- **Scraping:** Monitor the "Release Notes" page for PDF updates or changelogs.
- **Opportunity:** French B2B e-invoicing becomes mandatory for all businesses starting 2026. A simple mobile "invoice generator that connects to Chorus Pro" for small plumbers/artisans is a high-value target.

4.4 Source: Resmi Gazete (Turkey)

The Turkish Official Gazette.

4.4.1 Access Method

- **Method:** Scraping resmigazete.gov.tr.
- **Library:** mevzuat-gov-scraper (GitHub) is a Python-based parser for Turkish legislation.[33]
- **Implementation:** Use BeautifulSoup to parse daily lists.
- **Keywords (TR):** "Tebliğ" (Communiqué), "Vergi Usul Kanunu" (Tax Procedure Law), "Zorunluluk" (Obligation).
- **Logic:** Look for changes in "E-Fatura" (E-Invoice) limits. As limits drop, smaller businesses are forced to adopt digital tools, creating demand for mobile-first solutions over complex desktop ERPs.[34]

---

5. Core Signal Category D: B2B Sentiment Gaps

This category applies the "Unbundling" thesis. We look for successful B2B SaaS products that have neglected their mobile experience.

5.1 Primary Sources: G2 & Capterra

These platforms host millions of B2B reviews.

5.1.1 Critical Feasibility Note

Both platforms use enterprise-grade anti-scraping (DataDome, Cloudflare). **Direct scraping with simple requests will fail immediately.**

- **Solution 1 (Recommended):** Use a SERP API (like DataForSEO or SerpApi) to scrape the Google Cache or Search Results of G2/Capterra pages. This offloads the proxy management.[36]
- **Solution 2 (Advanced):** Playwright with stealth-plugin and residential proxies.

5.1.2 Extraction Logic

- **Target:** SaaS products in Logistics, Construction, Field Services.
- **Filter:**
  1. Find Product X.
  2. Check "Overall Rating" (Must be > 4.0 - Product is good).
  3. Check "Mobile App Rating" (Must be < 3.0 - Mobile experience is bad).
  4. Text Mine Reviews: "useless app," "crash," "can't work offline," "iPad."
- **Result:** Building a 3rd party mobile client (if API exists) or a competitor focused *only* on

the mobile workflow.[37]

5.2 Source: Trustpilot

Trustpilot is often used for service-based businesses (freight forwarders, clinics).

5.2.1 Access Method

- **Method:** Public API (Rate limited) or Scraping.
- **Library:** trustpilot-scraper (Python).[39]
- **Logic:** Monitor the "Business Services" category. Look for 1-star reviews on otherwise reputable companies specifically complaining about the *digital interface* (e.g., "Great shipping service, but their tracking portal is from 1990"). This indicates a need for a wrapper app or a modern competitor.

---

6. Cross-Source Strategy: The "Opportunity Score" Algorithm

The coding agent must not simply dump data; it must rank it. The "Opportunity Score" (0-100) is the synthesizer of the system.

6.1 The Scoring Formula

$$Score = (W_C \times S_C) + (W_B \times S_B) + (W_A \times S_A) + (W_D \times S_D)$$
Where:

- **W (Weight):** The strategic importance of the category.
- **S (Signal Strength):** Normalized value (0.0 to 1.0).

6.2 Weighting Configuration

1. **W_C (Regulatory - Weight 40%): Compulsion is King.** If a law requires X, the market for X is guaranteed.
   - $S_C = 1.0$ if a specific deadline is found in BOE/EUR-Lex.
2. **W_B (Friction - Weight 30%): Validation.** People are already feeling the pain.
   - S_B scales with the number of Reddit/Forum threads discussing "spreadsheets" for this topic.
3. **W_A (ASO - Weight 20%): Intent.**
   - S_A is high if Search Volume > 40 and Difficulty < 30.
4. **W_D (Sentiment - Weight 10%): Gap.**
   - S_D is high if competitors have < 3.0 star ratings.

6.3 Operational Workflow (The "Loop")

The agent should run in a specific sequence to maximize efficiency and minimize cost.

1. **Phase 1: The Law (Daily):** Scrape BOE, Resmi Gazete, EUR-Lex. Extract entities (e.g., "Digital Tachograph").
2. **Phase 2: The Friction (Weekly):** Take entities from Phase 1 and search Reddit/Forocoches/Ekşi. Look for "How to track [Entity]".
3. **Phase 3: The Validation (Ad-Hoc):** If Phase 2 yields hits, query AppTweak API for "[Entity] App".
4. **Phase 4: The Report:** If Score > 75, generate an alert: "High Probability Opportunity: [Entity] Tracker."

---

7. Recommended Tech Stack

To build this system effectively, the following technology stack is mandated for the coding agent.

Component	Technology	Justification
Language	**Python 3.11+**	Dominant ecosystem for scraping (Playwright, Scrapy) and NLP (Spacy).
Browser Automation	**Playwright**	Superior to Selenium for handling modern JS frameworks (React/Angular) and evading anti-bot detection.[41]
HTTP Client	**httpx (Async)**	Essential for high-throughput scraping without blocking.
Reddit API	**PRAW**	The industry standard, robust wrapper for Reddit API.[42]

XML Parsing	**xmltodict**	Simplifies parsing complex BOE/EUR-Lex XML dumps into Python dictionaries.[43]
NLP Engine	**Spacy**	es_core_news_lg, fr_core_news_lg, tr_core_news_tr. Runs locally, fast entity extraction, no recurring API costs.
Database	**PostgreSQL**	Relational integrity for "Laws" and "Apps", with JSONB support for unstructured "Forum Threads".
Job Queue	**Redis + Celery**	Manage scraping tasks, handle rate limits, and retries gracefully.
Deployment	**Docker**	Containerization ensures the scraper environment (browser binaries) is consistent across dev and prod.

---

8. Conclusion

This Technical Specification defines a rigorous, deterministic system for market research. By pivoting from "idea generation" to "signal extraction," it leverages the inherent inefficiencies of professional markets. The "unsexy" nature of the target verticals (HVAC, Compliance, Logistics) ensures lower competition, while the triangulation of **Regulatory Mandates** (Compulsion) with **Forum Complaints** (Friction) guarantees genuine user need.

The coding agent should proceed by implementing the **BOE** and **Reddit** modules first, as they offer the highest data accessibility and lowest barrier to entry, establishing a baseline flow of "Mandate -> Friction" signals before integrating the more costly and complex ASO and B2B Review modules.

Deep-Dive Documentation

1. Category A: ASO Search Arbitrage

1.1 Source: AppTweak

Overview: AppTweak is the market leader for ASO intelligence, offering the most accurate "Search Popularity" data derived directly from Apple's volume index.
Access Method: REST API.
Auth: API Key (X-Apptweak-Key header).
Budget: The "Small API" plan costs ~$166/month. This is the single largest expense in the stack. To mitigate this, the system uses AppTweak only as a validation step, not a discovery step.
Data Schema:

- **Request:** GET /ios/keywords/metrics.json?term={keyword}&country={cc}
- Response: (See Executive Summary for JSON).
  Signal Extraction:
- **Volume:** Focus on keywords with volume between 30 and 60. >60 implies dominance by giants (e.g., "Instagram"). 30-60 is the "Niche Sweet Spot."
- Competition: The competition_score is a compound metric. High score = Bad.
  Heuristic:
- If Volume > 30 AND Difficulty < 40 AND Top 3 Apps Rating < 3.5: **GREEN LIGHT**.
- This indicates users are searching, but the existing apps are failing them (crashes, bugs, outdated UI).

1.2 Source: MobileAction

Overview: MobileAction excels in Ad Intelligence. It tells us where the money is flowing.
Access Method: REST API.
Auth: Token (?token=YOUR_API_KEY).
Budget: Pricing is opaque/enterprise. Start with "Startup" plans if available, or use the limited free tier manually for spot-checking.
Signal Extraction:

- **Metric:** Search Ads Popularity.
- **Logic:** Look for "Keyword Gaps". If a keyword has High Search Ads Popularity (expensive CPC) but Low Organic Difficulty, it means advertisers are desperate. They *have* to buy ads because their product isn't good enough to rank organically. A high-quality indie app can steal this traffic for free via SEO.

1.3 Source: ASODesk

Overview: Strong in European/Russian markets, often offering better pricing/data for Turkey/Eastern Europe.
Access Method: REST API.
Auth: API Token.
Budget: Startup plan ~$33/mo.4
Data Schema:

- **Endpoint:** /v1/keywords/analytics
- Response: Includes daily_impressions—a more tangible metric than Apple's 0-100 index. Signal Extraction: Use ASODesk to cross-reference volume data from AppTweak. If both tools show rising volume for a niche term (e.g., "e-fatura" in Turkey), the signal is validated.

---

2. Category B: Workflow Friction

2.1 Source: Reddit

Overview: The primary engine for discovering user pain.
Access Method: PRAW.
Auth: OAuth2.
Code Sample (Advanced Friction Detection):

Python

```python
def analyze_friction_sentiment(subreddit, keywords):
    """
    Scans a subreddit for keywords and calculates a 'Frustration Score'.
    """
    for submission in subreddit.search(keywords, limit=50):
        # Calculate text ratio: Longer posts often indicate deeper pain
        text_len = len(submission.selftext)

        # Check for 'Help' flair if applicable
        flair = submission.link_flair_text

        # Keyword density of 'manual' terms
        manual_count = submission.selftext.lower().count("manual")
        excel_count = submission.selftext.lower().count("excel")

        if (manual_count + excel_count) > 2:
            print(f"HIGH FRICTION: {submission.title} (Score: {submission.score})")
```

Heuristic: Prioritize threads where the OP (Original Poster) describes a complex workflow involving >3 steps (e.g., "I take a photo, email it to myself, copy to Excel, then PDF it"). This entire chain can be one app.

2.2 Source: Ekşi Sözlük (Turkey)

Overview: Turkish "Sour Dictionary". Users define concepts, often cynically.
Access Method: Scraping (playwright or cloudscraper).
Heuristic:

- **Target:** Titles of banks, government gateways ("e-devlet", "uyap"), and ISP mobile apps.
- **Signal:** Look for "rezalet" (disgrace), "açılmıyor" (won't open), "giriş yapılmıyor" (can't login).
- **Opportunity:** If the official "Istanbul Kart" app has 50 pages of complaints about loading times, a lightweight wrapper or alternative interface (if API exists) is a winner.

2.3 Source: Forocoches (Spain)

Overview: The "Bar" of the Spanish internet. High noise, but high signal in specific sub-forums.
Access Method: Playwright (Headless).
Target Sub-forums: "Empleo" (Jobs), "Emprendedores" (Entrepreneurs).
Signal Logic: Search for "Gestoría" (Administrative Agency). Small business owners frequent Forocoches to ask how to file taxes or manage invoices without paying a Gestoría 50€/month. An app that automates this for 5€/month has a built-in market.

2.4 Source: Meneame (Spain)

Overview: News aggregator.
Access Method: API / Scraping.
Heuristic:

- **Trigger:** News article: "New Freelancer Quota System Approved."
- **Action:** Scrape the comments.
- **Signal:** Comments saying "I have no idea how to calculate my new quota." -> **App Idea:** "Cuota Autónomos Calculator."

---

3. Category C: Regulatory & Compliance

3.1 Source: BOE (Spain)

Overview: The source of truth for Spanish law.
Access Method: Open Data API.
Auth: None.
Implementation:

The xmltodict library is essential here. The BOE API returns a complex XML tree. The agent must parse <text> nodes within <disposicion> elements.
Regex Targets:

- \b(registro|libro|cuaderno) digital\b (Digital register/book/notebook).
- \b(obligación|deber) de conservar\b (Obligation to keep/store).
- \b(plazo|fecha) de adaptación\b (Adaptation deadline).

3.2 Source: EUR-Lex (EU)

Overview: EU Directives.
Access Method: Cellar API (REST).
Query Strategy:

- Focus on **Directives** (must be transposed) vs **Regulations** (apply immediately).
- **Keyword:** "Digitalisation," "Reporting," "Sustainability" (ESG is huge right now).
- **Signal:** The "CSRD" (Corporate Sustainability Reporting Directive) forces SMEs to report carbon. This is a massive app opportunity for "SME Carbon Trackers."

3.3 Source: Chorus Pro (France)

Overview: French e-invoicing.
Access Method: Documentation Scraping.
Signal Logic: The timeline for "Factur-X" implementation is the roadmap. If the mandate for "Medium Enterprises" kicks in Jan 2026, the app must launch Jan 2025 to catch the "early anxiety" wave.

3.4 Source: Resmi Gazete (Turkey)

Overview: Turkish Official Gazette.
Access Method: Scraping.
Signal Logic: "Enflasyon Muhasebesi" (Inflation Accounting). Recent changes in Turkey require inflation adjustment in accounting. This is complex math that small businesses struggle with. An "Inflation Adjustment Calculator" app based on the Resmi Gazete formulas is a valid Category C idea.

---

4. Category D: B2B Sentiment Gaps

4.1 Sources: G2, Capterra

Overview: Reviews for enterprise software.
Access Method: Google SERP Scraping (Bypass anti-bot).
Query: site:capterra.com "Product Name" "reviews" "mobile app"
Signal:

- If the snippet says "The mobile app is clunky" or "No Android version," capture this.
- **Synthesis:** If "Product X" has 10,000 users but a 2-star mobile app, create a "Mobile

Companion for X" (using their API).

4.2 Source: Trustpilot

Overview: Service reviews.
Access Method: Scraping/API.
Signal: Look for logistics companies (e.g., DHL, local couriers). If reviews say "I never know where my package is," build a "Universal Package Tracker" that scrapes that specific courier's tracking page better than their own app does.

---

5. Cross-Source Strategy

5.1 The Verification Loop

1. **Trigger:** BOE publishes a law on "Digital Time Tracking for Remote Workers."
2. **Verification 1 (Friction):** Search Reddit/Forocoches for "how to track hours remote." If users are sharing Excel sheets, Friction is confirmed.
3. **Verification 2 (Arbitrage):** Search AppTweak for "time tracker." If results are generic, but "compliance time tracker Spain" has 0 results, Arbitrage is confirmed.
4. **Action:** Build "CumpleHorario" - a simple app specifically for Spanish remote work law compliance.

5.2 The "Unbundling" Loop

1. **Trigger:** G2 reviews for a "Logistics ERP" complain that the "driver app is slow."
2. **Verification (Friction):** Search trucker forums for "ERP_Name app sucks."
3. **Action:** Build a lightweight driver app that connects to that ERP's API (or simply replaces the driver workflow).

---

6. Recommended Tech Stack

- **Scraping:** Python + Playwright (Handles JS/Anti-bot best).
- **API Client:** httpx (Async support for faster scraping).
- **NLP:** spacy (Fast, runs locally, good multi-language support).
- **Database:** PostgreSQL (JSONB columns for flexible schema storage).
- **Queue:** Redis (manage scraping jobs).
- **Scheduling:** Airflow or simple Cron.

This PRD provides the necessary blueprint for a coding agent to construct a high-fidelity signal discovery engine. By rigorously filtering for Regulatory Compulsion and Workflow Friction, the system minimizes market risk and targets genuine, urgent user needs.

Works cited

1. Apptweak Pricing: Cost and Pricing plans - SaaSworthy, accessed January 5, 2026, https://www.saasworthy.com/product/apptweak/pricing
2. AppTweak Overview: Features, Pricing, and Plans | Blog, accessed January 5, 2026, https://www.strataigize.com/blog/apptweak-overview-features-pricing-and-plans
3. API Data Pricing - Overview - AppTweak, accessed January 5, 2026, https://developers.apptweak.com/reference/pricing-table
4. API Plans | Asodesk, accessed January 5, 2026, https://asodesk.com/api-pricing
5. ASOdesk Review: Marketing Tool Analysis & Guide, accessed January 5, 2026, https://www.marketingmonk.so/products/asodesk
6. API Overview - Help Center, accessed January 5, 2026, https://help.asodesk.com/api-overview
7. App Store & Google Play Intelligence API - ASOdesk, accessed January 5, 2026, https://asodesk.com/api
8. API Solutions - MobileAction, accessed January 5, 2026, https://www.mobileaction.co/api-solutions/
9. Introduction - API Documentation | MobileAction, accessed January 5, 2026, https://docs.mobileaction.co/guide/introduction
10. Appfigures API, accessed January 5, 2026, https://docs.appfigures.com/
11. Appfigures API: Access, Limits, and Add-ons, accessed January 5, 2026, https://help.appfigures.com/en/article/appfigures-api-access-limits-and-add-ons-1seiibo/
12. Appfigures Pricing & Plans | ASO tools, App Intelligence, and Analytics, accessed January 5, 2026, https://appfigures.com/platform/pricing
13. Reddit API Pricing: What It Costs and Better Alternative, accessed January 5, 2026, https://data365.co/blog/reddit-api-pricing
14. How to get audience insights for free using the Reddit API, accessed January 5, 2026, https://www.philmorton.co/how-to-get-audience-insights-for-free-using-the-reddit-api/
15. How to make a custom Reddit notification system with Python, accessed January 5, 2026, https://www.freecodecamp.org/news/make-a-custom-reddit-notification-system-with-python-4dd560667b35/
16. What is the Trustpilot API request limit? - Stack Overflow, accessed January 5, 2026, https://stackoverflow.com/questions/69313397/what-is-the-trustpilot-api-request-limit
17. alkinkasap/eksiautoentry: automated script for adding entries to ekşi ..., accessed January 5, 2026, https://github.com/alkinkasap/eksiautoentry
18. uysalibov/eksi_scraper: asynchronous Ekşi Sözlük scraper - GitHub, accessed January 5, 2026, https://github.com/uysalibov/eksi_scraper

19. Various scraping functions for eksisozluk.com Written in python, accessed January 5, 2026, https://github.com/mcagridurgut/EksiSozlukScraper
20. Ekşi Sözlük scraper : r/LinuxTurkey - Reddit, accessed January 5, 2026, https://www.reddit.com/r/LinuxTurkey/comments/1oo5fmk/ek%C5%9Fi_s%C3%B6zl%C3%BCk_scraper/
21. TheBronx/shurscript: Script para forocoches - GitHub, accessed January 5, 2026, https://github.com/TheBronx/shurscript
22. Too Lazy to Click: How I Scraped a 400-Page Forum Thread with ..., accessed January 5, 2026, https://dev.to/hexcreator/too-lazy-to-click-how-i-scraped-a-400-page-forum-thread-with-python-1ol5
23. Menéame - GitHub, accessed January 5, 2026, https://github.com/Meneame/meneame.net
24. velorek1/meneameterm: Meneame.net en la terminal linux - GitHub, accessed January 5, 2026, https://github.com/velorek1/meneameterm
25. Meneame - Menéame · GitHub, accessed January 5, 2026, https://github.com/Meneame
26. Datos abiertos (OpenData) - BOE.es, accessed January 5, 2026, https://www.boe.es/datosabiertos/api/api.php
27. EUR-Lex Data Extraction using Web Services, accessed January 5, 2026, https://eur-lex.europa.eu/content/tools/webservices/DataExtractionUsingWebServices.pdf
28. Cellar data - Publications Office of the EU - European Union, accessed January 5, 2026, https://op.europa.eu/en/web/cellar/cellar-data
29. An API for European Union law, accessed January 5, 2026, http://api.epdb.eu/
30. PISTE presentation – Communauté Chorus Pro, accessed January 5, 2026, https://communaute.chorus-pro.gouv.fr/documentation/presentation-de-piste/
31. Getting Started - Chorus Pro API Documentation, accessed January 5, 2026, https://cpro-docs.choruspay.fr/fr/getting-started
32. External specifications - Communauté Chorus Pro, accessed January 5, 2026, https://communaute.chorus-pro.gouv.fr/documentation/specifications-externes/
33. muhammetakkurtt/mevzuat-gov-scraper: Web scraping ... - GitHub, accessed January 5, 2026, https://github.com/muhammetakkurtt/mevzuat-gov-scraper
34. Resmî Gazete, accessed January 5, 2026, https://www.turmob.org.tr/arsiv/mbs/resmigazete/32840-5.pdf
35. API'ler (UYGULAMA PROGRAMLAMA ARAYÜZLERİ) - Haber Ege, accessed January 5, 2026, https://www.haberege.com.tr/yazar/zafer-ozcivan/apiler-uygulama-programlama-arayuzleri-1078-kose-yazisi
36. G2 Scraping Teardown - ScrapeOps, accessed January 5, 2026, https://scrapeops.io/websites/g2/
37. G2 Product Info Scraper API | Free Trial - Piloterr, accessed January 5, 2026, https://www.piloterr.com/library/g2-product-info
38. How to Scrape Capterra to Get Reviews Data - Octoparse, accessed January 5, 2026, https://www.octoparse.com/blog/how-to-scrape-capterra

39. irfanalidv/trustpilot_scraper: A Python library for scraping Trustpilot ..., accessed January 5, 2026, https://github.com/irfanalidv/trustpilot_scraper
40. How to Scrape Trustpilot.com Reviews and Company Data - Scrapfly, accessed January 5, 2026, https://scrapfly.io/blog/posts/how-to-scrape-trustpilot-com-reviews
41. Web Scraping with Playwright and Python: A Developer's Guide, accessed January 5, 2026, https://dev.to/alex_aslam/web-scraping-with-playwright-and-python-a-developers-guide-3i48
42. I created a simple script that gets Reddit posts with certain keywords ..., accessed January 5, 2026, https://www.reddit.com/r/learnpython/comments/flfdkd/i_created_a_simple_script_that_gets_reddit_posts/
43. API para el acceso a la colección de Legislación Consolidada, accessed January 5, 2026, https://www.boe.es/datosabiertos/documentos/APIconsolidada.pdf