GRANTRADAR — PRODUCT REQUIREMENTS DOCUMENT

AI-Powered Grant Intelligence Platform for Research Labs

Product Overview

Problem: Research labs miss critical deadlines because grants are discovered too late. Manual checking is reactive and inefficient. By the time a lab finds a perfect grant, they have only days to apply.

Solution: GrantRadar monitors 60+ sources in real-time and alerts labs within minutes of new grant postings. AI predicts when grants will be posted based on historical patterns, giving labs advance notice to prepare applications.

Core Features (MVP)

1. Real-Time Discovery • Continuous monitoring (15-min intervals) • NIH/NSF/Grants.gov webhooks + polling • Adaptive scraping of 50+ foundations • Alert labs within 5 minutes of posting	**2. Predictive Intelligence** • Analyze 5+ years of posting patterns • Predict grant release dates (±3 days) • Alert labs 1-2 weeks before expected • "Prepare now" mode for draft writing
3. Instant Matching & Alerts • Match new grants within 60 seconds • Email + SMS + Slack notifications • Match score with reasoning • Live dashboard updates (WebSocket)	**4. Smart Profile & Learning** • 5-minute onboarding + ORCID import • Continuous learning from interactions • Success prediction scoring • Competitor tracking (peer lab grants)

Data Sources & API Integration

Federal Grant Sources (Primary)

Source	Coverage	API/Method	Update Freq	Historical
NIH RePORTER	$42B/year, 50k+ active	REST API (free, no auth)	Weekly	1985-present
NSF Award Search	$8B/year, 11k+ active	REST API (free, no auth)	Daily	1989-present
Grants.gov	All 26 federal agencies	REST API + RSS feeds	Real-time	2002-present
DOD SBIR/STTR	$4B/year	Web scraping (structured)	Quarterly	2015-present

Foundation Sources (50+ tracked)

Top 20 Priority Foundations:

- Chan Zuckerberg Initiative ($1B+/year) - API available for some programs, otherwise scrape
- Bill & Melinda Gates Foundation ($6B+/year) - Structured HTML scraping
- Simons Foundation ($500M+/year) - RSS feed + web scraping
- Howard Hughes Medical Institute ($800M+/year) - Structured scraping
- Wellcome Trust ($1.5B+/year) - REST API available
- Alfred P. Sloan Foundation, Templeton Foundation, Arnold Ventures, Open Philanthropy, Moore Foundation, Kavli Foundation, Schmidt Futures, Carnegie Corporation, Rockefeller Foundation, Ford Foundation (15 more tracked)

Real-Time Tracking Workarounds

Challenge: Not all sources update in real-time. NIH RePORTER updates weekly, some foundations update monthly, DOD SBIR posts quarterly. We need multi-source triangulation to maintain <5 minute alerting.

Multi-Source Triangulation Strategy

Problem	Workaround Strategy
NIH RePORTER *Weekly updates only*	Primary: Grants.gov RSS feed catches NIH postings in real-time (NIH posts to both simultaneously) Secondary: Direct scraping of grants.nih.gov/funding-opportunities (HTML changes tracked hourly) Tertiary: Email monitoring - NIH sends listserv emails immediately. Set up dedicated inbox, parse emails via API **Result: Detect NIH grants within 15 minutes despite API lag**
Foundation Sites	Primary: RSS feeds where available (CZI, Simons, Arnold Ventures have feeds)

Monthly updates, irregular	Secondary: HTML change detection - hash entire opportunities page, re-scrape when hash changes (every 30 min) Tertiary: Social media monitoring - foundations announce on Twitter/LinkedIn before website. Use Twitter API v2 to monitor @CZI_Science, @gatesfoundation Quaternary: Newsletter parsing - subscribe to all foundation newsletters, parse incoming emails for grant announcements **Result: Detect foundation grants within 1 hour of announcement**
DOD SBIR/STTR *Quarterly only*	Primary: SBIR.gov has daily updates for all agencies (not just DOD) - use this instead Secondary: Pre-solicitation notices on SAM.gov (beta.sam.gov API) - agencies post intent 30-60 days early Tertiary: Agency-specific portals (Navy: navysbir.com, Air Force: afwerx.com) often post before consolidated site **Result: Detect SBIR opportunities 30+ days before official posting via pre-solicitations**
European Sources *ERC, Horizon, etc.*	Primary: Horizon Europe Portal (ec.europa.eu/info/funding-tenders/opportunities/portal) - has RSS feeds Secondary: ERC (erc.europa.eu) updates quarterly but announces via press releases first - scrape press page **Result: Track international opportunities for US labs with EU collaborators**

Change Detection Architecture

HTML Fingerprinting: For each source URL, compute SHA-256 hash of filtered HTML (remove timestamps, session IDs, ads). Store hash in Redis. Poll every 15-30 min. If hash changes → trigger full scrape + LLM extraction. Avoids expensive scraping when nothing changed.

Differential Scraping: When change detected, use LLM to compare old vs new HTML: 'What grants were added or changed?' This avoids re-processing entire catalog. LLM responds: 'New grant: R01-AI-123456, title changed for R21-HL-789012'. Only process deltas.

Email Parsing Pipeline: Set up grantradar@yourdomain.com. Subscribe to: NIH Guide (list.nih.gov), NSF funding alerts, all foundation newsletters. Use Gmail API or IMAP to pull emails. LLM extracts structured data: 'From: grants@nih.gov, Subject contains PA-25-123 → parse as new grant announcement'. 90% of foundations announce via email before website.

Social Media Monitoring: Twitter API v2: Monitor @NSF, @NIH, @CZI_Science, @gatesfoundation, @wellcometrust. Set up keyword alerts: 'funding opportunity', 'RFP', 'call for proposals', 'now open'. When match found, extract URL → trigger immediate scrape. Average lead time: 30-120 minutes before RSS/website.

Redundancy & Failover System

Multi-Path Verification: Every grant must be confirmed by 2+ sources before alerting users. Example: NIH R01 appears in (1) Grants.gov RSS, (2) NIH Guide email, (3) grants.nih.gov scrape. If only 1 source shows it → flag for manual review (could be parsing error). If 2+ sources confirm → send alert immediately.

Source Priority Ranking: Tier 1 (most reliable): Official APIs (NSF, Grants.gov). Tier 2: RSS feeds, official emails. Tier 3: Web scraping. Tier 4: Social media. Use Tier 1 as ground truth. If Tier 3/4 detects something not in Tier 1/2 within 6 hours → alert human reviewer before sending to users.

Latency Monitoring: Track time delta: actual_post_time (from agency) vs our_detection_time. Goal: median <5 min, 95th percentile <30 min. If detection latency >2 hours for any source → alert on-call engineer. Dashboard shows: 'NIH: 3 min avg, NSF: 8 min avg, CZI: 45 min avg'.

Backup Scraping: If primary method fails (API down, rate limit hit, parsing error), automatically fallback to alternative. Example: Grants.gov API down → switch to RSS → if RSS fails → scrape HTML directly. Never miss a grant due to single point of failure.

Implementation Priority

Week 1 MVP: Grants.gov RSS (covers 80% of federal), NSF API, basic NIH scraping. Get to 80% coverage fast.

Week 2-3: Add email parsing (NIH Guide, foundation newsletters), HTML change detection, top 10 foundation scrapers.

Week 4+: Social media monitoring, pre-solicitation tracking (SAM.gov), European sources, multi-path verification.

API Implementation Details

NIH RePORTER API: Endpoint: https://api.reporter.nih.gov/v2/projects/search | Rate limit: 3 req/sec | Fields: project_num, title, abstract, pi_name, org_name, award_amount, fiscal_year, project_start, project_end | Historical: 40+ years, 3M+ grants

NSF API: Endpoint: https://www.research.gov/awardapi-service/v1/awards.json | Rate limit: None (pagination: 25/page) | Fields: id, title, abstract, pi, institution, amount, start_date, expiration_date | Historical: 1989-present, 500k+ grants

Grants.gov API: Endpoint: https://api.grants.gov/v1/api/search2 | Webhooks: Available via RSS | Fields: opportunity_id, title, description, eligibility, deadline, funding_amount, agency | Real-time: New postings appear within minutes

Wellcome Trust API: Endpoint: https://api.wellcome.org/grants | Auth: API key required (free) | Fields: grant_id, title, description, funding_orgs, amount_awarded, award_date | Historical: 2005-present

Grant Prediction Methodology

Historical Data Collection (Backtesting)

Data Requirements: Minimum 5 years of posting history per grant program. Collect: posting date, deadline, funding cycle, fiscal year, agency schedule changes, congressional budget timing.

Step 1 - Historical Scraping: Pull all NIH funding opportunity announcements (FOAs) 2019-2024 from RePORTER. Extract: FOA number, posting date, parent announcement. Build database of ~2,000 recurring programs with 5-10 postings each.

Step 2 - Pattern Recognition: Group by grant mechanism (R01, R21, K99, etc.). Identify posting patterns: (a) Fixed schedule - R01 typically 3x/year: Jan, May, Sept; (b) Fiscal calendar - many agencies post in Q4 (Jul-Sep); (c) Day-of-week patterns - Wednesdays most common; (d) Lead time - average 60-90 days between posting and deadline.

Step 3 - LLM Analysis: Feed historical data to Claude with prompt: 'Analyze posting patterns. Identify: (1) Regular vs irregular programs, (2) Seasonal patterns, (3) Dependencies (budget cycles, leadership changes), (4) Anomalies (COVID delays, policy shifts).' Output: Structured JSON with confidence scores.

Step 4 - Prediction Model: Train simple time-series model per grant program. Features: days_since_last_post, month, quarter, fiscal_year_day, weekday, historical_variance. Target: days_until_next_post. Model: LightGBM or simple moving average. Accuracy target: 85% within ±3 days for regular programs.

Real-Time Prediction Updates

Continuous Learning: Every time a predicted grant is posted, log: predicted_date, actual_date, error_days. Retrain model monthly with new data. Adjust confidence scores based on recent accuracy.

Confidence Tiers: High (>85% accuracy): Programs with 5+ years consistent pattern. Alert 14 days before. Medium (70-85%): Some variance. Alert 7 days before. Low (<70%): Irregular programs. No prediction alerts, only react when posted.

Alert Strategy: 14 days before: 'NIH R01 expected Jan 15 (±3 days, 90% confidence). Start drafting now.' When posted: 'NIH R01 just posted! We predicted this 12 days ago. Your draft should be ready.'

Validation & Testing

Backtesting Approach: Train on 2019-2022 data. Test predictions on 2023-2024 actual postings. Measure: mean absolute error (MAE), accuracy within ±3/7/14 days, false positive rate (predicted but never posted).

Success Criteria: 85% of high-confidence predictions within ±3 days. 70% of medium-confidence within ±7 days. <5% false positives. Users find predictions valuable (measured by 'prepare now' alert engagement >40%).

MVP Validation: Start with top 20 most predictable programs (NIH R01, NSF CAREER, etc.). Manual validation for first 3 months. Collect user feedback: 'Was this prediction helpful?' Expand to more programs as accuracy improves.

Agentic Architecture & Backend Workflows

The platform runs on 5 autonomous AI agents that operate 24/7 without human intervention. Each agent has specific responsibilities and communicates through a shared event bus.

Agent	Responsibilities & Workflows
Discovery Agent *Real-time monitoring*	• Continuous Polling: Check NIH/NSF/Grants.gov every 15 minutes. Compare checksums to detect new postings instantly. • Webhook Listeners: Subscribe to RSS feeds, API webhooks where available. Receive push notifications for new grants (NIH uses this). • Adaptive Scraping: LLM-powered HTML parsing every 30 minutes for foundations. Auto-adjusts if page structure changes. Retries with variations. • Speed Target: New grant detected → processed → matched → alert sent in <5 minutes end-to-end.
Prediction Agent *Forecasts grant postings*	• Historical Analysis: Analyze 5+ years of posting dates for every grant program (e.g., NIH R01 typically posts 3rd Wednesday of Jan/May/Sept). • Pattern Recognition: LLM identifies patterns - day of week, month, quarter, federal fiscal calendar, agency-specific schedules. • Prediction Window: Forecast posting date ±3 days with 85% confidence. Alert labs 7-14 days before predicted date. • Preparation Mode: Send "Start drafting now" alerts with previous year's RFP, eligibility criteria, and suggested outline.
Curation Agent *Validates & enriches data*	• Quality Check: LLM scores each grant 0-100 for completeness (has deadline? amount? description?). Flags low-quality for manual review. • Data Enrichment: If deadline missing, agent visits grant URL and extracts it. Fills eligibility, focus areas, requirements automatically. • Deduplication: Embeddings + LLM to find same grant from multiple sources. Merges entries, keeps most complete version. • Auto-Categorization: Assigns research fields, grant types, career stages, keywords. Creates vector embeddings for semantic search.
Matching Agent *Connects grants to labs*	• Profile Building: Analyzes lab's publications (PubMed API), past grants (NIH RePORTER), and onboarding answers. Creates semantic profile embedding. • Semantic Matching: Compares grant embedding to lab profile (cosine similarity). LLM evaluates fit: research alignment, eligibility, strategic value. • Success Prediction: Finds similar labs that applied to similar grants (historical data). Calculates win

	probability based on outcomes. • Strategic Insights: LLM generates personalized advice: why apply, what to emphasize, potential collaborators, timeline recommendations.
Delivery Agent *Instant multi-channel alerts*	• Instant Notifications: Push alerts via email, SMS (Twilio), Slack webhook within 60 seconds of matching. Priority queue: >90% match = immediate. • Smart Routing: LLM decides urgency level. Critical (>95% match, <14 days deadline) = SMS+Email+Slack. High (>85%) = Email+Slack. Medium = Email only. • Live Dashboard: WebSocket connection pushes updates to browser. New grants appear with animation. Real-time counter: "3 new grants today". • Personalized Content: Each alert includes match score, key reasons, deadline countdown, strategic advice. Never generic.
Orchestrator Agent *Real-time coordination*	• Event Pipeline: Grant discovered → validate (30s) → match all labs (60s) → send alerts (30s). Total latency <2 minutes for 100 labs. • Priority Queue: Critical grants jump queue. >95% match + <30 days = highest priority. Background tasks (re-indexing, analytics) run during low traffic. • Health Monitoring: Real-time dashboards for agent latency, error rates, queue depth. Auto-scale workers when queue >100 jobs. Alert on-call if any agent down >5min. • Circuit Breakers: If NIH API fails, switch to backup scraper. If Claude API slow (>10s), route to GPT-4. Graceful degradation always.

Backend Implementation Details

Real-Time Event Stream (Redis Streams): Pub/sub architecture with consumer groups. Events: grant_discovered, grant_validated, match_computed, alert_sent. Each event timestamped for latency tracking. Target: end-to-end <2 min.

Priority Job Queue (Celery + RabbitMQ): 3 priority levels: critical (>90% match), high (new grants), normal (re-indexing). Workers auto-scale 3-20 based on queue depth. Route critical jobs to fastest workers.

Vector Database (Pinecone): Real-time upserts on new grants (<100ms). Streaming similarity search across 10k+ lab profiles in <200ms. Namespace per user + global index for batch queries.

WebSocket Server (Socket.io): Persistent connections to dashboard. Push updates on: new grants, deadline reminders, prediction alerts. Reconnection with exponential backoff. Redis pub/sub for horizontal scaling.

LLM Orchestration: Claude Sonnet 4 for matching/insights (quality). GPT-4o-mini for extraction (speed). Groq Llama for categorization (throughput). Latency SLA: 95th percentile <3s. Automatic failover between providers.

User Flows

Onboarding (5 min): Sign up → Answer 8 questions → Import ORCID → Agents build profile in background → First alerts within 2 hours (or immediately if perfect match found)

Daily Engagement: Receive instant alerts (SMS/Email/Slack) → Open dashboard (live updates) → Review grant details → Save/dismiss → Dashboard shows "3 new, 5 saved, 2 deadlines this week"

Success Metrics

- Alert Latency: <5 min discovery → alert
- Daily Active Users (DAU) >40%
- Alert open rate >70% (SMS/Email)
- Match accuracy >85% (user saves)
- Prediction accuracy >80% (±3 days)
- Churn <8%/month
- NPS >55

Tech Stack

- Frontend: React + WebSocket (live updates)
- Backend: Python (FastAPI) + Socket.io
- Database: PostgreSQL + Pinecone (vector)
- Queue: Celery + RabbitMQ (priority queue)
- Events: Redis Streams (real-time)
- AI: Claude/GPT/Groq with failover
- Scraping: Playwright + Webhooks
- Notifications: Twilio (SMS) + SendGrid
- Hosting: Vercel (FE) + Railway (BE)

MVP Timeline (3 Weeks)

Week	Deliverables
Week 1	Real-time infrastructure: Redis Streams event bus. Celery + RabbitMQ priority queue. PostgreSQL + basic schema. Build polling scrapers for NIH/NSF APIs (15-min intervals). Test end-to-end latency.
Week 2	Matching + alerts: TF-IDF matching engine. Email (SendGrid) + SMS (Twilio) integration. Simple onboarding form. Manual profile creation for 5 beta users. Test: grant posted → alert sent in <5 min.

Week 3	Live dashboard: React + WebSocket for real-time updates. Grant cards with match scores. Prediction Agent v1 (basic historical pattern analysis). Landing page + Stripe. Launch with 10 paying users at $200/mo. Validate <5 min alerts.

Out of Scope (V1)

- Application writing assistance or grant proposal tools
- Collaboration features (team workspaces, shared pipelines)
- Mobile app (responsive web only)
- Integrations with institutional systems (InfoEd, Cayuse)
- Advanced analytics dashboard (heat maps, trend analysis)