THE CONTENT ENGINE

SUBAGENT 2

Interest Content

Automatically detect trending topics in your market, score them for relevance, and generate point-of-view content that inserts your brand into the conversations your buyers are already having — within 24–48 hours of a trend spike.

A step-by-step technical implementation guide.

Part 2 of 4 • The Content Engine Series

What This Subagent Does

The Interest subagent monitors your market's conversations in real time and generates point-of-view (POV) content when relevant topics spike. The goal is not to summarize what's being said — it's to add a take that positions your brand as a voice worth following.

Awareness content (Subagent 1) answers questions your buyers are already asking. Interest content is different: it inserts your brand into conversations your buyers are already having, with a perspective that makes them stop scrolling.

The Workflow at a Glance

1. RSS feeds, Google Trends, and audience intelligence tools detect trending topics
2. Claude scores each topic for ICP relevance and urgency
3. Claude writes POV content: LinkedIn posts, Twitter/X threads, and short-form commentary
4. n8n queues high-urgency content for publishing within 24–48 hours
5. Lower-urgency evergreen takes route to the content calendar for scheduling

Why Speed Matters

POV content has a shelf life. A take on an industry trend published 2 days after the spike gets 5–10x more engagement than the same take published 2 weeks later. This subagent is designed for velocity — detect, write, review, publish in under 48 hours.

What You Need Before Starting

Required Accounts & API Keys

This subagent uses a combination of tools for trend detection. You don't need all of them — pick the combination that fits your budget and market. The guide covers multiple options.

Tool	Purpose	Notes
n8n	Workflow orchestration	Same instance as Subagent 1. Self-hosted (free) or n8n Cloud.
Claude API	Content generation + scoring	Anthropic API key. Sonnet recommended for speed.
RSS Feeds	Publication monitoring	Free. n8n has a built-in RSS Feed Trigger node.

Google Trends	Topic spike detection	Free via SerpAPI or unofficial API.
SparkToro	Audience intelligence	Paid. Identifies what your ICP reads, follows, and shares.
Feedly	Industry feed aggregation	Free tier available. API access on Pro plan.
Google Sheets	Content calendar	Where drafts land for review. Same sheet as Subagent 1.
Slack (optional)	Urgency notifications	Alerts when high-urgency POV content is queued.

Prerequisite: Define Your Brand POV

Before this subagent can write compelling takes, you need to articulate the 2–3 core beliefs your brand holds. These are not taglines — they're the principles that inform your perspective on every industry conversation.

Examples of Brand POV Statements

- "Automation without strategy is just faster failure."
- "Infrastructure beats tactics. Systems compound; campaigns don't."
- "The best marketing doesn't feel like marketing — it feels like help."
- "Compliance should be a product feature, not a department."

Write yours down. Claude will reference these in every POV prompt to ensure your content has a consistent editorial voice.

Step 1: Build the Trend Detection Layer

This step sets up multiple “listening posts” that detect what’s trending in your market. You’ll combine 2–3 of these sources depending on your budget and market.

1A. RSS Feed Monitoring (Free, Essential)

RSS is the backbone of this subagent. You’ll subscribe to 10–15 publications, blogs, and newsletters your ICP actually reads — then have n8n monitor them for new content daily.

How to Find the Right Feeds

1. **Ask your sales team:** "What do our prospects mention reading?" Note the publications.
2. **Check SparkToro (if available):** Enter your ICP’s job title or a competitor’s URL. SparkToro shows exactly which publications, podcasts, and social accounts your audience follows.
3. **Search manually:** Google "[your industry] blog" or "[your category] newsletter" and subscribe to the top 10–15 results.
4. **Find the RSS URL:** Most blogs have an RSS feed at /feed, /rss, or /feed.xml. If you can’t find it, use a tool like RSS.app to generate one from any webpage.

n8n RSS Feed Trigger

n8n’s RSS Feed Trigger node polls a feed URL at a set interval and outputs new items since the last check.

```
Node: RSS Feed Trigger

Settings:
  Feed URL: https://example-publication.com/feed
  Poll Times: Every 6 hours (or daily)

Output per item:
{
  "title": "Why SaaS Companies Are Rethinking Their Sales Model",
  "link": "https://example-publication.com/article/...",
  "contentSnippet": "First 200 characters of article...",
  "pubDate": "2025-01-15T08:00:00Z"
}
```

Create one RSS Feed Trigger per publication, or use a single trigger with a Code node that loops through an array of feed URLs. The second approach is cleaner for 10+ feeds.

Multi-Feed Approach with a Code Node

```
// n8n Code Node — runs after a Schedule Trigger (daily)
// Fetches multiple RSS feeds and combines results

const feeds = [
  'https://publication-one.com/feed',
  'https://publication-two.com/rss',
  'https://industry-blog.com/feed.xml',
  // Add 10-15 feeds here
];

const Parser = require('rss-parser');
const parser = new Parser();

const allItems = [];
for (const url of feeds) {
  try {
    const feed = await parser.parseURL(url);
    feed.items.forEach(item => {
      allItems.push({
        title: item.title,
        link: item.link,
        snippet: item.contentSnippet?.substring(0, 300) || '',
        pubDate: item.pubDate,
        source: feed.title || url
      });
    });
  } catch (e) {
    // Skip feeds that fail — don’t block the workflow
  }
}

// Filter to last 24 hours only
const cutoff = new Date(Date.now() - 24 * 60 * 60 * 1000);
const recent = allItems.filter(i => new Date(i.pubDate) > cutoff);

return recent.map(item => ({ json: item }));
```

Note on RSS in n8n

n8n’s built-in RSS node uses the rss-parser npm package internally. If you’re self-hosting, it’s already available. If you use the Code node approach above, you may need to install rss-parser in your n8n environment. Alternatively, use n8n’s HTTP Request node to fetch each feed URL and parse the XML manually — RSS is just XML.

1B. Google Trends Monitoring (Free)

Google Trends reveals when search interest for a topic is spiking relative to its baseline. This catches trends that publications haven't written about yet — giving you a first-mover advantage.

Option 1: SerpAPI Google Trends Endpoint

```
GET https://serpapi.com/search.json

Query Parameters:
  engine: google_trends
  q: YOUR_KEYWORD
  data_type: TIMESERIES
  date: today 1-m    (last 30 days)
  api_key: {{$credentials.serpApi}}

Response (simplified):
{
  "interest_over_time": {
    "timeline_data": [
      { "date": "Jan 8 – 14, 2025", "values": [{ "value": "73" }] },
      { "date": "Jan 15 – 21, 2025", "values": [{ "value": "100" }] }
    ]
  }
}
```

A value of 100 means peak interest in the time range. Look for keywords that jumped 30%+ in the last week compared to their 30-day average — that's a spike worth writing about.

Option 2: Google Trends RSS (Completely Free)

Google Trends offers its own RSS feed for trending searches. This is less precise than the API but costs nothing and requires no account.

```
Feed URL (US, all categories):
https://trends.google.com/trending/rss?geo=US

Feed URL (filtered by category):
https://trends.google.com/trending/rss?geo=US&category=b
  (b = business, t = technology, e = entertainment, etc.)
```

Add this as another RSS Feed Trigger in n8n. The output will be broad — that's fine. Claude will filter for relevance in Step 2.

1C. SparkToro Audience Intelligence (Optional, Paid)

SparkToro shows you exactly what your ICP reads, follows, shares, and watches. It's the highest-signal source for trend detection because it's filtered to your audience — not the general population.

How to Use SparkToro for This Subagent

SparkToro does not have a public API with automated access at the time of writing. Here are two approaches to integrate it:

1. **Manual Weekly Export (recommended for most teams):** Log into SparkToro weekly. Search by your ICP's job title, a relevant hashtag, or a competitor's URL. Export the "Trending" and "Frequently Shared" data to a CSV. Upload the CSV to a Google Sheet that n8n monitors.
2. **Semi-automated with Firecrawl:** Use Firecrawl to scrape the publications and social accounts SparkToro identifies as most-followed by your ICP. Subscribe to those sources via RSS (Step 1A). This is a one-time setup that makes your RSS feed list directly informed by audience data.

> **SparkToro's Real Value Here**
>
> You don't need SparkToro running daily. Use it once to identify the 10–15 publications and 20–30 social accounts your ICP actually follows. Then monitor those sources directly via RSS and social listening. SparkToro is the research layer; RSS is the monitoring layer.

1D. Social Listening (Optional, Advanced)

For teams that want to monitor social conversations directly, you can add Twitter/X, LinkedIn, or industry Slack communities as data sources.

Platform	Method	Cost
Twitter/X	X API v2 — search recent tweets by keyword. Use the filtered stream endpoint for real-time monitoring.	Free tier: 500k tweets/month
LinkedIn	No public API for feed monitoring. Use Firecrawl to scrape company pages or hashtag feeds, or manually flag trending posts.	Firecrawl costs only
Reddit	Reddit API — monitor subreddit "hot" posts for spike detection. Different from Subagent 1 which scrapes questions.	Free with app registration

Feedly	Feedly’s API aggregates RSS, social, and news sources into topic boards. The “Trending” feature surfaces spikes.	Free tier limited; Pro for API

Start with RSS + Google Trends. Add social listening only after the core workflow is running and you need more signal density.

Step 2: Score and Filter with Claude

Raw trend data is noisy. A new article about AI from TechCrunch might be irrelevant to your cybersecurity ICP. This step sends all detected topics to Claude for relevance scoring, urgency assessment, and angle identification.

2A. The Trend Scoring Prompt

This prompt runs daily (or every 6 hours for high-velocity markets). It processes all new items from your trend detection sources.

```
System Prompt:
You are a content strategist monitoring industry trends.

Our ICP: [YOUR IDEAL CUSTOMER PROFILE]
Our product category: [YOUR CATEGORY]
Our brand POV (core beliefs):
  1. [BELIEF 1]
  2. [BELIEF 2]
  3. [BELIEF 3]

User Message:
Here are articles and topics that surfaced in the last 24 hours
from publications our ICP reads:

{{$json.allItems}}

For each item, evaluate:

1. RELEVANCE (1-5): How directly does this topic affect our ICP?
   5 = core to their daily work
   3 = tangentially related
   1 = irrelevant

2. URGENCY (1-5): How time-sensitive is a response?
   5 = breaking news, conversation happening right now
   3 = trending this week, 48-hour window
   1 = evergreen topic, no time pressure

3. ANGLE: Given our brand POV, what is our specific take?
   This should be contrarian, additive, or a reframe —
   NOT a summary of what’s already being said.
   One sentence describing the angle.

4. CONTENT TYPE: What format best delivers this take?
   Options: linkedin_post, twitter_thread, short_commentary,
   detailed_analysis, video_hook
```

```

Output JSON:
{
  "scored_topics": [
    {
      "title": "original article/topic title",
      "source": "publication name",
      "link": "url",
      "relevance": 4,
      "urgency": 5,
      "combined_score": 9,
      "our_angle": "one sentence describing our take",
      "content_type": "linkedin_post",
      "brand_pov_reference": "which of our beliefs this ties to"
    }
  ]
}

Only include topics with relevance >= 3.
Sort by combined_score descending.
Return valid JSON only.
```

2B. The API Call

```
POST https://api.anthropic.com/v1/messages

Headers:
  x-api-key: {{$credentials.anthropicApi}}
  anthropic-version: 2023-06-01
  Content-Type: application/json

Body:
{
  "model": "claude-sonnet-4-20250514",
  "max_tokens": 4096,
  "messages": [
    {
      "role": "user",
      "content": "[FULL PROMPT WITH VARIABLES]"
    }
  ]
}
```

2C. Route by Urgency

After parsing Claude’s response, use an n8n IF node to split topics into two paths based on their urgency score:

- **Urgency 4–5 (publish within 24–48 hours):** Route to immediate content generation (Step 3) and send a Slack notification so the team knows time-sensitive content is queued.
- **Urgency 1–3 (schedule for later):** Route to the content calendar with a suggested publish date. These are evergreen POV pieces that can be published anytime.

```
// n8n Code Node — Parse and split by urgency
const response = $input.first().json;
const text = response.content[0].text;
const clean = text.replace(/```json|```/g, '').trim();
const parsed = JSON.parse(clean);

return parsed.scored_topics.map(topic => ({
  json: {
    ...topic,
    is_urgent: topic.urgency >= 4
  }
}));
```

Step 3: Generate POV Content

For each qualifying topic, Claude generates content in the format identified during scoring. The key difference from Subagent 1: this content must have a strong, specific take — not just answer a question.

3A. LinkedIn Post Prompt (Primary Format)

LinkedIn posts are the highest-leverage format for B2B POV content. Under 200 words, designed to stop the scroll and drive comments.

```
System Prompt:
You are a thought leader in [YOUR INDUSTRY].
You write LinkedIn posts that are direct, opinionated, and add
a perspective that isn’t being said by everyone else.

Brand POV: [YOUR CORE BELIEFS]

User Message:
Trending topic: "{{$json.title}}"
Source: {{$json.source}}
Summary: {{$json.snippet}}
Our angle: {{$json.our_angle}}
Brand POV to tie back to: {{$json.brand_pov_reference}}

Write a LinkedIn post (under 200 words) that:

1. Opens with a hook that challenges conventional thinking
   (pattern interrupt, bold claim, or counterintuitive observation)
2. Delivers 2-3 sentences of substance — your actual take,
   backed by reasoning or a specific example
3. Does NOT summarize the trending article — adds to the conversation
4. Ties back to our brand’s core belief naturally
5. Ends with a question that invites debate or personal experience

Formatting rules:
- Short paragraphs (1-2 sentences max)
- Use line breaks between paragraphs for readability
- No hashtags
- No emojis
- Write like a practitioner sharing a conviction, not a brand posting content

Output JSON:
{
  "hook": "first line of the post (standalone)",
  "full_post": "the complete post with line breaks",
  "word_count": 187,
  "closing_question": "the comment-driving question"
}
```

3B. Twitter/X Thread Prompt

For topics that benefit from a longer-form breakdown or when your audience is active on X. Threads of 3–5 posts work best for B2B.

```
User Message:
Trending topic: "{{$json.title}}"
Our angle: {{$json.our_angle}}

Write a Twitter/X thread (3-5 tweets) that:

1. Tweet 1: Hook — bold claim or observation (under 280 chars)
2. Tweets 2-4: Build the argument with one point per tweet.
   Each tweet should stand alone but build on the previous one.
3. Final tweet: Restate the core insight and ask for responses.

Rules:
- Each tweet under 280 characters
- No hashtags, no emojis
- Conversational tone — sounds like a person, not a brand

Output JSON:
{
  "tweets": ["tweet 1", "tweet 2", "tweet 3", ...],
  "thread_length": 4
}
```

3C. Short Commentary Prompt

For lower-urgency topics or when you want to comment on someone else's post rather than write original content. This is the fastest content type — 2–3 sentences that add a take.

```
User Message:
An influencer in our space posted this: "{{$json.snippet}}"
Our angle: {{$json.our_angle}}

Write a 2-3 sentence comment that:
1. Acknowledges their point without restating it
2. Adds a specific, actionable insight they didn't cover
3. Sounds like a peer contributing, not a brand marketing
```

```
Output JSON:
{
  "comment": "the 2-3 sentence response",
  "context": "where to post this (comment on their post, quote tweet, etc.)"
}
```

3D. Detailed Analysis Prompt

For high-relevance, lower-urgency topics that deserve a longer treatment. These become blog posts or newsletter essays that establish deeper authority.

```
User Message:
Trending topic: "{{$json.title}}"
Source: {{$json.source}}
Our angle: {{$json.our_angle}}

Write a 500-word analysis that:

1. States the trend and what most people are saying about it (2-3 sentences)
2. Identifies what’s being missed or misunderstood (your thesis)
3. Supports your thesis with 2-3 specific points, examples, or data
4. Concludes with what this means for practitioners (actionable takeaway)
5. Bridges naturally to your product category without being promotional

Include a headline (under 80 chars) and a one-sentence teaser.

Output JSON:
{
  "headline": "...",
  "teaser": "one sentence summary",
  "body": "full 500-word analysis in markdown",
  "word_count": 512
}
```

Step 4: Route to Calendar and Notify

After content is generated, it needs to land in the right place based on urgency. High-urgency content triggers immediate notifications; everything else goes to the content calendar for scheduling.

4A. Google Sheets Schema

Extend the content calendar from Subagent 1 or create a separate tab. Here are the columns specific to Interest content:

Column	Type	Description
Date Created	Date	Auto-populated by n8n
Trending Topic	Text	The original article/topic title
Source	Text	Publication or platform where it was detected
Source URL	URL	Link to the original article/post
Relevance Score	Number	1–5 from Claude’s scoring
Urgency Score	Number	1–5 from Claude’s scoring
Combined Score	Number	Relevance + Urgency
Our Angle	Text	One-sentence description of our take
Content Type	Dropdown	LinkedIn Post / Twitter Thread / Commentary / Analysis
Content Draft	Long text	The generated content
Brand POV Used	Text	Which core belief this ties to
Status	Dropdown	Draft / In Review / Approved / Published / Expired
Publish By	Date	Deadline based on urgency: urgent = today+2, other = flexible
Assigned To	Text	Who’s reviewing this piece

4B. n8n Routing Logic

```
Field Mapping (n8n Google Sheets Node):

Date Created    →  {{ $now.toISODate() }}
Trending Topic  →  {{ $json.title }}
```

```
Source          →  {{ $json.source }}
Source URL      →  {{ $json.link }}
Relevance Score →  {{ $json.relevance }}
Urgency Score   →  {{ $json.urgency }}
Combined Score  →  {{ $json.combined_score }}
Our Angle       →  {{ $json.our_angle }}
Content Type    →  {{ $json.content_type }}
Content Draft   →  {{ $json.full_post || $json.body || $json.comment }}
Brand POV Used  →  {{ $json.brand_pov_reference }}
Status          →  Draft
Publish By      →  {{ $json.is_urgent ? $now.plus(2, 'days').toISODate() : '' }}
```

4C. Slack Notification (Urgent Content)

For urgency 4–5 topics, send a Slack message so the team can review and publish fast. Use n8n’s Slack node after the IF node that filters for urgent content.

```
Slack Message Template:

🔥 *Urgent POV content queued*

*Topic:* {{$json.title}}
*Source:* {{$json.source}}
*Our angle:* {{$json.our_angle}}
*Format:* {{$json.content_type}}
*Publish by:* {{$json.publish_by}}

Draft is in the content calendar. Review and publish within 48 hrs.
```

Step 5: The Complete n8n Workflow

Here is the full workflow assembled end-to-end. This runs as a separate workflow from Subagent 1, on a daily schedule.

Workflow Architecture

```
Schedule Trigger (Daily, or every 6 hours)
  │
  ├── Code Node: Fetch all RSS feeds (10–15 publications)
  ├── HTTP Request: SerpAPI Google Trends (per keyword)
  ├── (Optional) Google Sheets: Pull SparkToro export data
  │
  └── Merge Node (combine all trend signals)
      │
      └── HTTP Request: Claude API (scoring + filtering)
          │
          └── Code Node: Parse JSON, add urgency flag
              │
              ├── IF Node: urgency >= 4 (URGENT PATH)
              │   │
              │   └── HTTP Request: Claude API (POV content gen)
              │       │
              │       ├── Google Sheets: Append Row
              │       └── Slack: Urgent notification
              │
              └── ELSE: urgency < 4 (SCHEDULED PATH)
                  │
                  └── HTTP Request: Claude API (POV content gen)
                      │
                      └── Google Sheets: Append Row
```

Node Count and Estimated Build Time

Metric	Value
Total n8n nodes	12–16 (depending on source count and notification setup)
Build time (first time)	2–3 hours (plus 1–2 hours finding RSS feeds)

Build time (experienced)	45–60 minutes
Daily run time	3–8 minutes (API dependent)
Output per day	2–5 scored topics, 1–3 content pieces
Weekly output	7–15 POV content pieces across formats

Step 6: Human-in-the-Loop Review

POV content carries more brand risk than awareness content. A bad take can damage credibility. Every piece needs a human pass, especially for urgent content.

1. **Take check.** Is this actually a contrarian or additive perspective? Or is Claude just restating the consensus with slightly different words? If the take could be attributed to any company in your space, it’s not a take — rewrite it.
2. **Accuracy check.** Verify any claims about the trending topic. Claude may mischaracterize the original article or trend. Click the source link and confirm the facts.
3. **Tone check.** POV content should be confident, not combative. You want “respectfully disagree” energy, not “everyone else is an idiot” energy. Edit anything that reads as arrogant rather than authoritative.
4. **Bridge check.** Does the tie-back to your brand feel natural? If you stripped out the brand connection, would the post still be worth reading? That’s the test.
5. **Timing check.** For urgent content, is the conversation still active? Check the source and social media. If the trend has already faded, downgrade to scheduled or skip entirely.

Time Investment

POV content requires less editing time per piece (they’re shorter) but more judgment per piece (the take matters more). Plan for 5–10 minutes per LinkedIn post, 10–15 minutes per thread or analysis piece. With 1–3 pieces per day, that’s 15–45 minutes of daily editorial work.

Troubleshooting Common Issues

Problem	Solution
Claude's takes are generic	Your brand POV statements are too vague. Rewrite them to be specific and opinionated. Instead of "we believe in efficiency," try "automation without strategy just creates faster mistakes."
Too many irrelevant topics passing	Tighten your ICP description in the scoring prompt. Add 2–3 examples of irrelevant topics and explicitly tell Claude to filter those out.
RSS feeds return stale content	Some feeds only update weekly. Replace slow feeds with more active ones. Check pubDate filtering — make sure the 24-hour cutoff is working.
Google Trends data is too broad	Use more specific keyword phrases. "AI in healthcare" is too broad; "clinical AI documentation automation" is better. Match your actual ICP's search language.
Content sounds like commentary, not a take	Add this line to your prompt: "Your post should be quotable. If someone screenshotted one sentence from this post, would it get reshared? If not, make it bolder."
Urgency scoring is always low	Your market may not have daily breaking news. Adjust the schedule to every 2–3 days instead of daily. Or lower the urgency threshold from 4 to 3 for the "fast path."
Duplicate topics across days	Add a deduplication Code node that hashes topic titles and checks against a Google Sheet of previously scored topics. Same approach as Subagent 1.

What's Next

This subagent keeps your brand visible and relevant by inserting your perspective into the conversations your buyers are already having. Combined with Subagent 1 (Awareness), you now have both the top and middle of your content funnel automated.

In Subagent 3 (Desire Content), you'll build a competitive intelligence engine that scrapes competitor websites — pricing pages, feature lists, case studies — and generates comparison content and ROI calculators that do the evaluation work for your prospect before they ever talk to sales.

Up next: Subagent 3 — Desire Content (Competitive Intelligence → Comparison Assets)