zango

How to build AI agents for risk & compliance

Agenda

01 Core concepts

02 The anatomy of a good AI use case

03 Building an agent

04 Implementation Case Study

About us

Ashi Bajwa

Regulatory AI Architect

zango

Sam Green

Senior Policy and Partnerships Lead

zango

Justine Wootton

UK Compliance Director

capitalontap

Tim Tyler

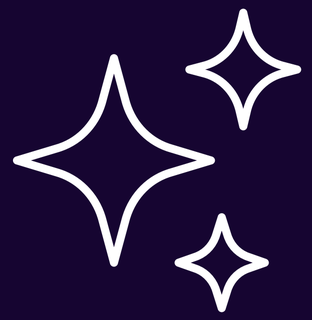
Vice President

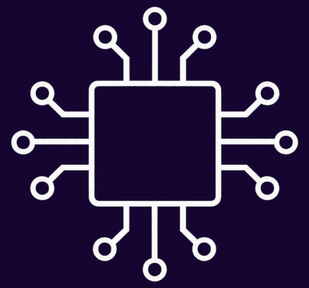
ICA INTERNATIONAL COMPLIANCE ASSOCIATION

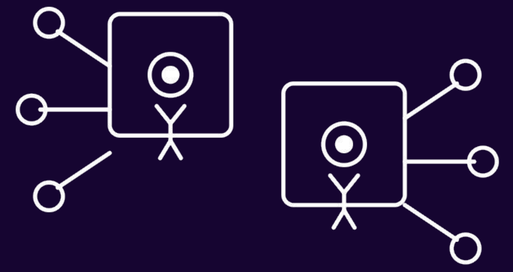
Poll 1

What brings you here today?

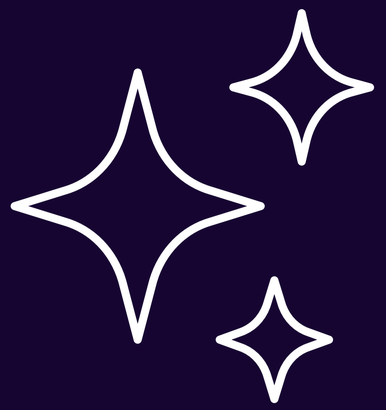
Core concepts





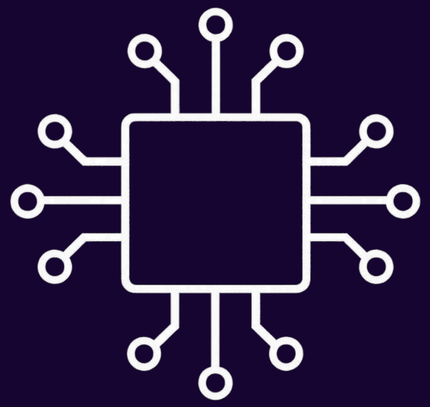


What is an LLM chatbot?



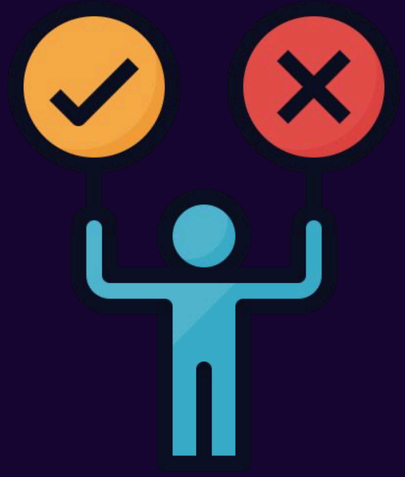
An AI system trained on large amounts of text that learns to predict what words should come next.

What is an agent?

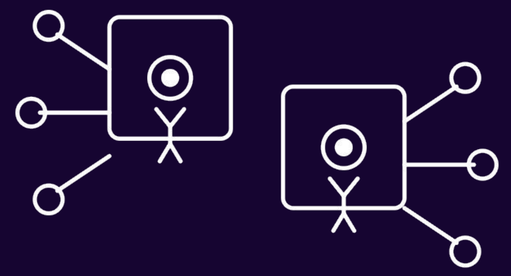


An AI system that takes in information, decides what to do, and carries out tasks autonomously.

What makes an agent



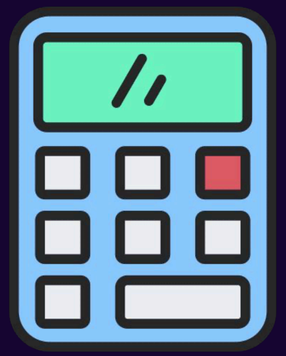
What is an agentic workflow?



A set of connected AI agents working together to complete a process end-to-end.

How are they different

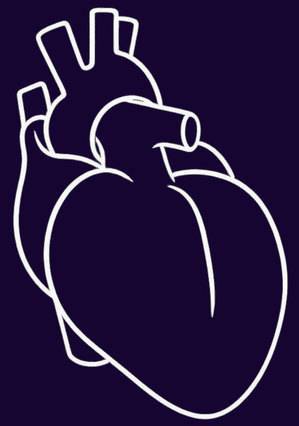
LLM Chat Tool



Agent

Agentic Workflow

The anatomy of a good AI use case



The anatomy of a good AI use case

High-volume / Repetitive

Clear source of truth

Defined output structure

Reviewable / auditable

The anatomy of a good AI use case

High-volume / repetitive

Defined output structure

Clear source of truth

Reviewable / auditable

Transaction Monitoring

Horizon Scanning

Policy Management

Marketing Compliance

The anatomy of a good AI use case

High-volume / repetitive

Defined output structure

Clear source of truth

Reviewable / auditable

Transaction Monitoring

Horizon Scanning

Policy Management

Marketing Compliance

Horizon Scanning

FCA FINANCIAL CONDUCT AUTHORITY

PSR Payment Systems Regulator

European Commission

The problem with horizon scanning

Fragmented inputs - Teams monitor multiple sources

Noise - Most updates are irrelevant increasing risk of missing change.

Interpretation bottleneck
Deciding what is relevant requires manual review

Horizon Scanning as an AI use case

- High volume and repetitive
- Clear source of truth
- Defined output structure
- Reviewable and auditable

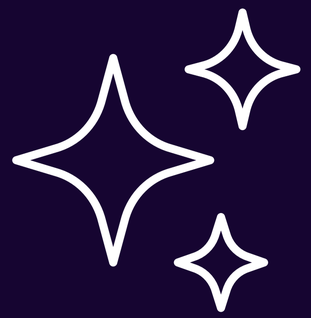
Poll 2

How is Horizon Scanning currently performed in your firm?

Lets build a horizon scanning agent

Profile ..... Sources ..... Relevance ..... Report

Workshop 1

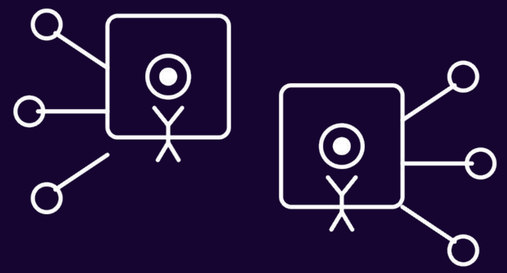


Principles for success

- **Don't overthink it** - Start simple, some of the magic is in the range of results and format included in the response.
- **Have trusted sources to compare** - Establish a baseline to validate results. Consider building a feedback loop to track what was added or removed.
- **Experiment!** - Iterate on your prompts. Each run teaches you something new about what works.

Questions?

Workshop 2



Resources

Who does this work for

- **Smaller compliance teams** without a systemised horizon scanning process.
- **Current Workflow is** manual or newsletter-driven
- **Limited Scope** - Likely to have a limited geographic scope or product range

Constraints to be aware of

- **Coverage reviews needed to ensure** no alerts missed
- **Website structures -** Different sites have different architectures.
- **Jurisdictions** - Multi-jurisdiction scanning across broader regulatory landscapes can be complex

Questions?

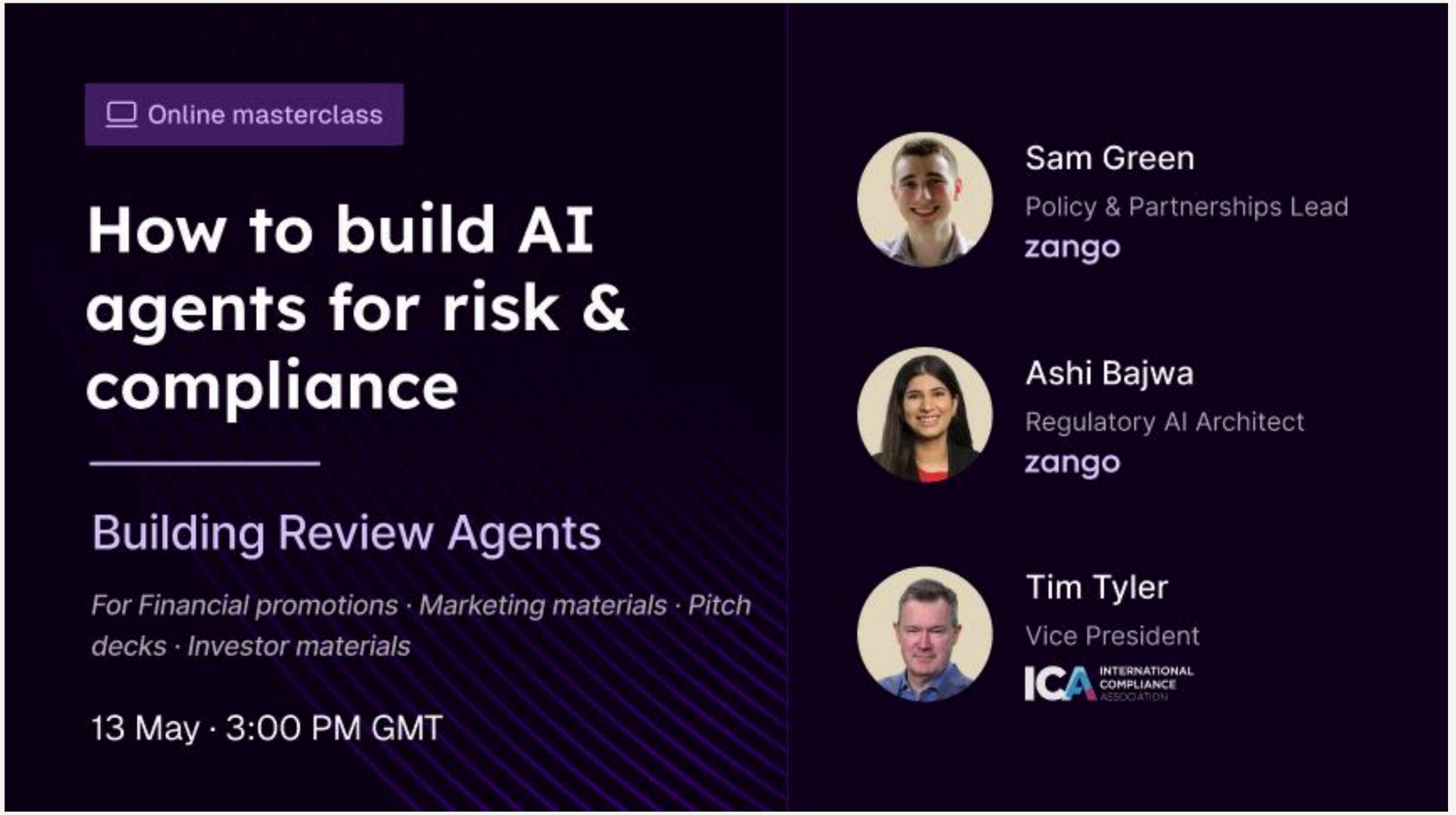
Scaling horizon scanning

- **Track multiple sources at once**
- **Highlight relevant and actionable updates**
- **Turn regulatory updates into a knowledge base**

Poll 3

What compliance use case should we build next?

Next workshop



FCA Horizon Scanner

Build an AI-powered regulatory monitoring workflow in n8n

Introduction

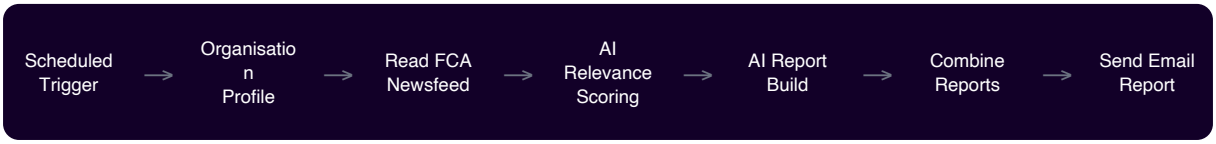
In this workshop you will build **FCA Horizon Scanner** — an n8n workflow that automatically reads the FCA's RSS news feed, uses AI to score each article's relevance to a fictional fintech company (**Meridian Pay**), generates styled HTML report cards, and emails them as a single digest.

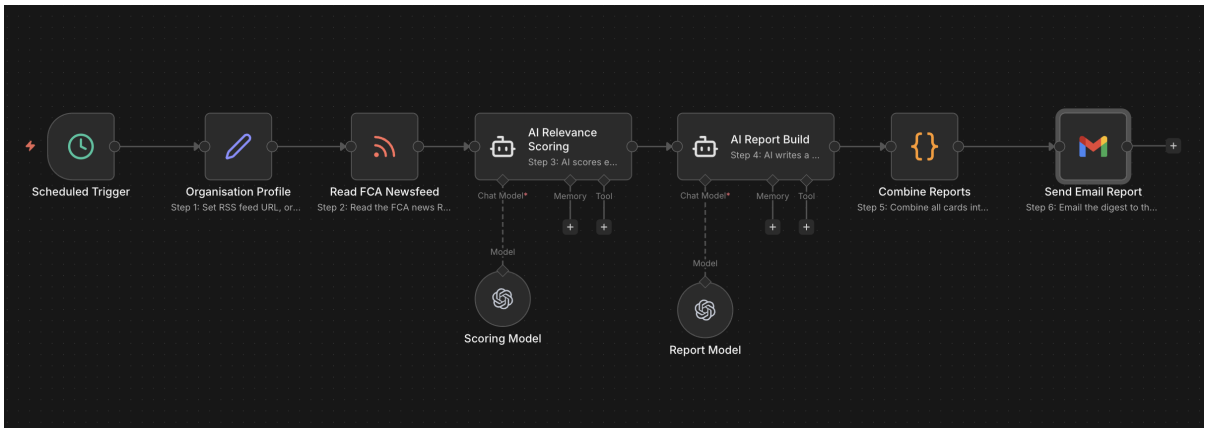
The workflow uses **7 nodes** in a straight-line pipeline — simple enough to build in under 30 minutes.

> **What you'll learn**
> RSS ingestion, AI agent prompting with structured JSON output, HTML email generation, n8n expression syntax, and the `$input.all()` aggregation pattern.

Architecture Overview

Data flows left to right through 7 nodes:





The complete FCA Horizon Scanner workflow in the n8n editor canvas

How the loop works
The RSS node outputs multiple articles (one item each). n8n automatically loops the two AI nodes — each article gets scored and turned into an HTML card individually. The Combine Reports node then collects all cards into a single item so Gmail sends just one email.

Prerequisites

- An **n8n cloud account** (free tier works) — n8n.io
- A **Gmail account** connected as a credential in n8n (OAuth2)
- An **OpenAI API credential** configured in n8n (or the built-in free credits)

Step 1 — Organisation Profile

1

Organisation Profile — Define Your Config

Settings

This node stores all configurable values in one place so you never need to dig into other nodes to change settings.

1.1 Create a new workflow called **FCA Horizon Scanner**. A **Schedule Trigger** node is added automatically (or add one manually and set it to run weekly on Mondays at 9am).

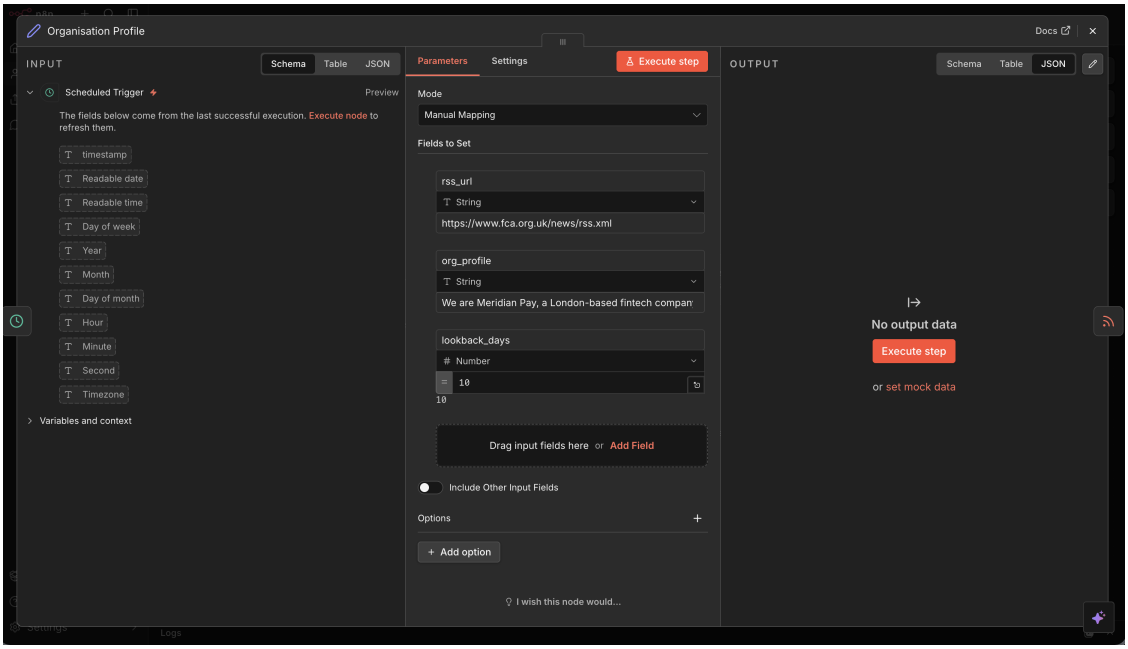
1.2 Click **+** after the trigger → search for **"Edit Fields (Set)"** → select it.

1.3 Rename to **Organisation Profile**.

1.4 Set **Mode** to **Manual Mapping**.

1.5 Add three fields using **"Add Field"**:

Field Name	Type	Value
rss_url	`String`	`https://www.fca.org.uk/news/rss.xml`
org_profile	`String`	`We are Meridian Pay, a London-based fintech company offering payment processing and e-money services across the UK and Europe. We are authorised by the FCA and hold an e-money licence. Our key compliance areas are Anti-Money Laundering (AML), payment services regulation, consumer protection, and data privacy. We operate primarily in the UK (GBR) and have passporting arrangements into the EU.`
lookback_days	`Number`	`10`



Organisation Profile node showing all three fields

> **Tip**
> For `lookback_days`, make sure the type is **Number** (not String). This field is available for future use — e.g. adding a date filter step later.

Step 2 — Read FCA Newsfeed

2 Read FCA Newsfeed — Fetch the RSS Feed

RSS

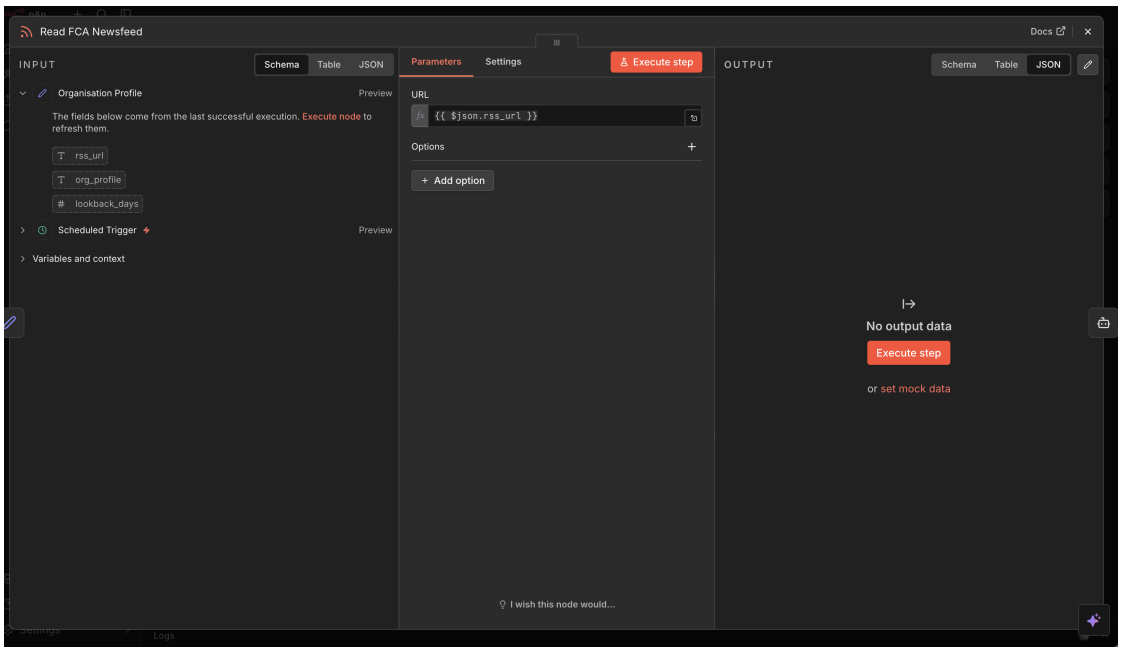
This node reads the FCA's RSS feed and outputs one item per article with fields like `title`, `link`, `pubDate`, and `content`.

2.1 Click **+** after Organisation Profile → search for **"RSS Feed Read"** → select it.

2.2 Rename to **Read FCA Newsfeed**.

2.3 In the **URL** field, click the expression toggle and enter:

```
{{ $json.rss_url }}
```



Read FCA Newsfeed — the URL field uses an expression

What comes out
This node typically outputs ~20 items (articles). Each has title, link, pubDate, content, and categories. All items flow into the next node, which processes them one at a time automatically.

Step 3 — AI Relevance Scoring

3 AI Relevance Scoring — Rate Each Article

AI Agent

This AI Agent receives each article (one at a time — n8n loops automatically) and produces a structured JSON analysis with a relevance score, summary, topics, key dates, and reasoning.

3.1 Click **+** after Read FCA Newsfeed → search for **"AI Agent"** → select it.

3.2 Rename to **AI Relevance Scoring**.

3.3 Under **Chat Model**, click **+** → select **OpenAI Chat Model**. Set the model to **gpt-4o-mini** and select your OpenAI credential.

3.4 Set **Source for Prompt** to **"Define below"**. Paste into the **Prompt** field:

```
=ORGANISATION PROFILE:
{{ $('Organisation Profile').first().json.org_profile }}

ARTICLE TITLE: {{ $json.title }}
ARTICLE DATE: {{ $json.pubDate }}
ARTICLE URL: {{ $json.link }}
ARTICLE CATEGORY: {{ $json.categories }}

ARTICLE CONTENT:
{{ $json.content }}
```

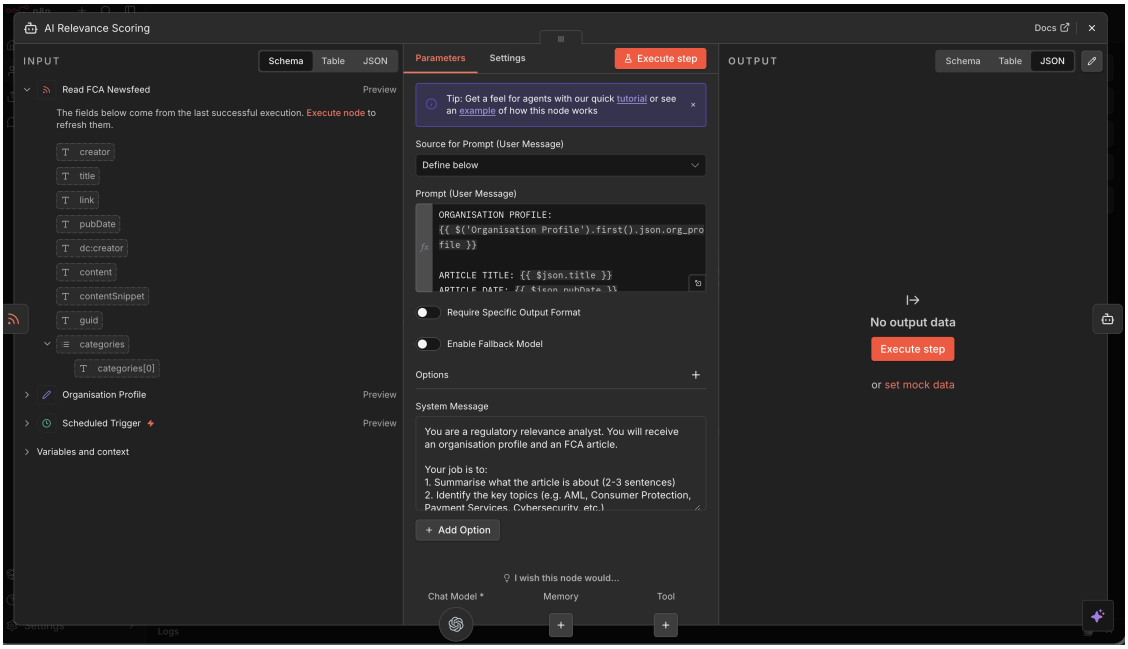
3.5 Expand **Options** → set **System Message** to:

```
You are a regulatory relevance analyst. You will receive an organisation profile and an
 FCA article.

Your job is to:
1. Summarise what the article is about (2-3 sentences)
2. Identify the key topics (e.g. AML, Consumer Protection, Payment Services, Cybersecur
ity, etc.)
3. Extract ALL dates, deadlines, and timelines from the article (exact dates, relative
timeframes like 'this autumn' or 'H1 2025', consultation periods, implementation milest
ones)
4. Score relevance to the organisation: 0 to 100
5. Decide: is_relevant true (score >= 50) or false (score < 50)
6. Explain WHY it is or is not relevant to this specific organisation

Respond with ONLY valid JSON (no markdown, no code fences):
{
  "title": "short title",
  "source_url": "copy the ARTICLE URL exactly as provided above",
  "summary": "2-3 sentence summary",
  "topics": ["topic1", "topic2"],
  "key_dates": [
    {"date": "19 September 2024", "event": "Forbearance period begins"},
    {"date": "H1 2025", "event": "New regime expected"}
  ],
  "relevance_score": 72,
  "is_relevant": true,
  "reasoning": "This is relevant because it directly impacts our e-money licence obliga
tions and consumer protection requirements..."
}

IMPORTANT: The source_url MUST be the exact ARTICLE URL from the input — copy it verbat
im, do not modify or omit it.
IMPORTANT: The key_dates array must contain EVERY date or timeframe from the article. I
f genuinely none exist, use an empty array [].
IMPORTANT: The reasoning must specifically reference the organisation's business areas
and explain the connection (or lack of connection).
```



AI Relevance Scoring — prompt with expressions

Key expressions

`{{ $('Organisation Profile').first().json.org_profile }}` — pulls the org profile

`{{ $json.title }}`, `{{ $json.link }}` — pulls fields from the current article

Automatic looping

If the RSS node outputs 20 articles, this AI node runs 20 times — once per article. No loop logic needed!

Step 4 — AI Report Build

4 AI Report Build — Generate Styled HTML Cards

AI Agent

This node takes the JSON analysis from Step 3 and transforms it into a styled HTML report card — green header for relevant articles, grey for not relevant.

4.1 Click **+** after AI Relevance Scoring → search for **"AI Agent"** → select it.

4.2 Rename to **AI Report Build**.

4.3 Connect a **Chat Model** (same setup — OpenAI gpt-4o-mini).

4.4 Set **Source for Prompt** to **"Define below"**. Paste into the **Prompt** field:

```
=ANALYSIS RESULT:
{{ $json.output }}

Write an HTML report card for this article. The analysis result above contains everythi
ng you need: title, summary, topics, key_dates, relevance_score, is_relevant, and reaso
ning. Use all of these fields.
```

4.5 Expand **Options** → set **System Message** to:

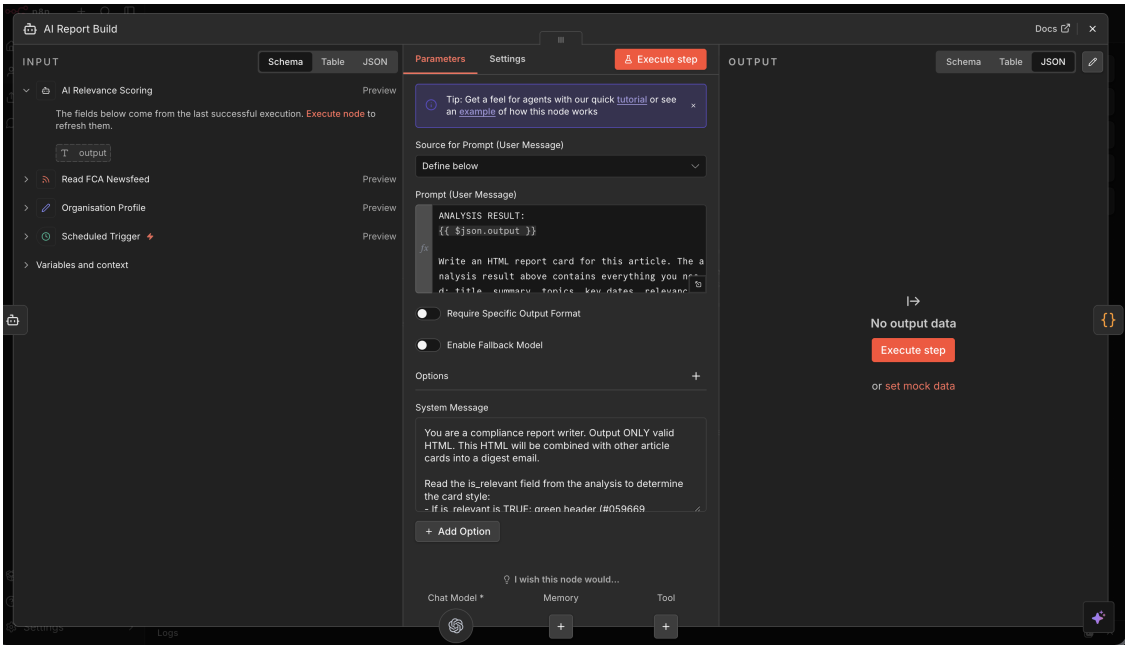
```
You are a compliance report writer. Output ONLY valid HTML. This HTML will be combined
with other article cards into a digest email.

Read the is_relevant field from the analysis to determine the card style:
- If is_relevant is TRUE: green header (#059669 background)
- If is_relevant is FALSE: grey header (#6b7280 background)

Use this exact HTML structure:

<div style="font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif; max
-width: 680px; margin: 0 auto 24px; color: #1a1a1a; line-height: 1.6; border: 1px solid
 #e5e7eb; border-radius: 8px; overflow: hidden;">
  <div style="background: [#059669 if relevant, #6b7280 if not]; color: white; padding:
 14px 20px;">
    <h2 style="margin:0; font-size:18px;">[Article Title]</h2>
    <p style="margin:4px 0 0; opacity:0.85; font-size:13px;">Published [date] — Score:
[score]/100</p>
  </div>
  <div style="padding: 16px 20px;">
    <p style="margin:0 0 12px;"><strong>[RELEVANT or NOT RELEVANT]</strong> — [reasonin
g from analysis, 1-2 sentences explaining WHY]</p>
    <table style="width:100%; border-collapse:collapse; font-size:13px; margin-bottom:1
2px;">
      <tr style="background:#f9fafb;"><th style="text-align:left; padding:6px 8px; bord
er-bottom:1px solid #e5e7eb;">Date</th><th style="text-align:left; padding:6px 8px; bor
der-bottom:1px solid #e5e7eb;">Event</th></tr>
      [One <tr> per item in key_dates. If empty, one row: 'No specific dates' / '']
    </table>
    <p style="margin:0; font-size:13px; color:#6b7280;">Topics: [comma-separated] | <a
href="[source_url]" style="color:#1e3a5f;">Read full article</a></p>
  </div>
</div>

RULES:
- Output ONLY the HTML above, nothing else
- No markdown, no code fences
- Inline styles only
- Green header = relevant, grey header = not relevant
- Keep it compact — this is a card, not a full report
```



AI Report Build — converts JSON analysis into styled HTML cards

Why two AI nodes?
Splitting analysis (structured JSON) from presentation (HTML) makes each task simpler and more reliable. The scorer focuses on accuracy; the writer focuses on formatting.

Step 5 — Combine Reports

5 Combine Reports — Merge All Cards into One

Code

Up to this point, each article flows through as a separate item. This Code node collects *all* the HTML cards and joins them into a single string so Gmail sends just one email (not one per article).

5.1 Click **+** after AI Report Build → search for **"Code"** → select it.

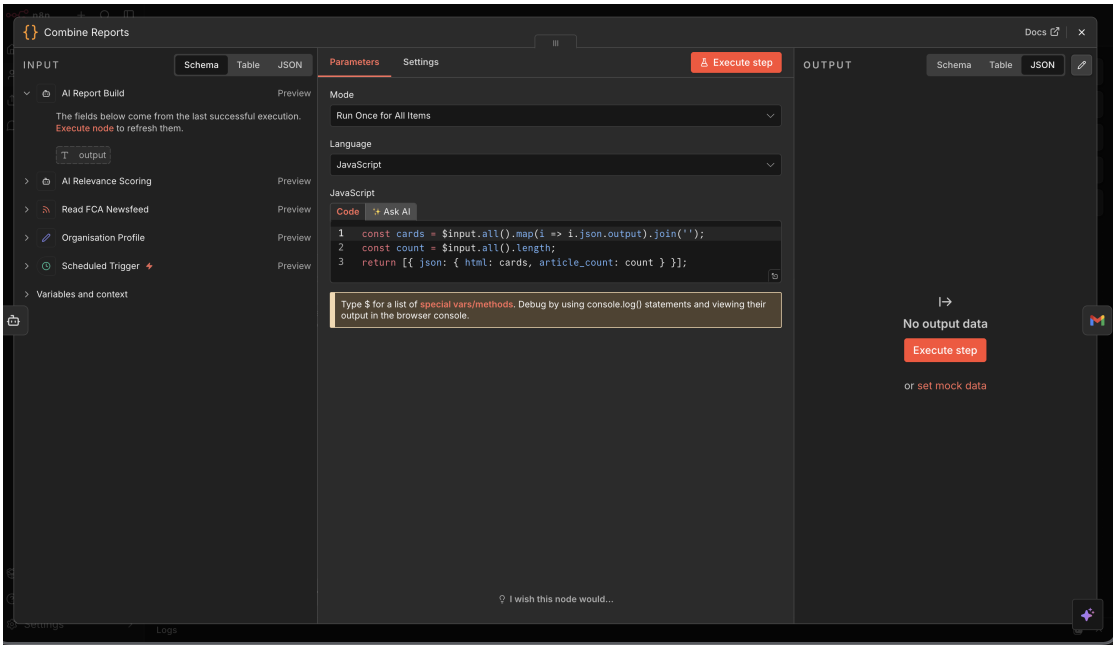
5.2 Rename to **Combine Reports**.

5.3 Set **Mode** to **Run Once for All Items** (critical — this is what aggregates everything).

5.4 Set **Language** to **JavaScript**.

5.5 Paste this code:

```
const cards = $input.all().map(i => i.json.output).join('');
const count = $input.all().length;
return [{ json: { html: cards, article_count: count } }];
```



Combine Reports — Code node set to "Run Once for All Items"

Why is this needed?

Without this node, Gmail receives 20 separate items and sends 20 separate emails. The `$input.all()` call collects all items, `.map()` extracts the HTML from each, and `.join('')` concatenates them into one string. The node returns a single item — so Gmail fires exactly once.

Step 6 — Send Email Report

6 Send Email Report — Gmail

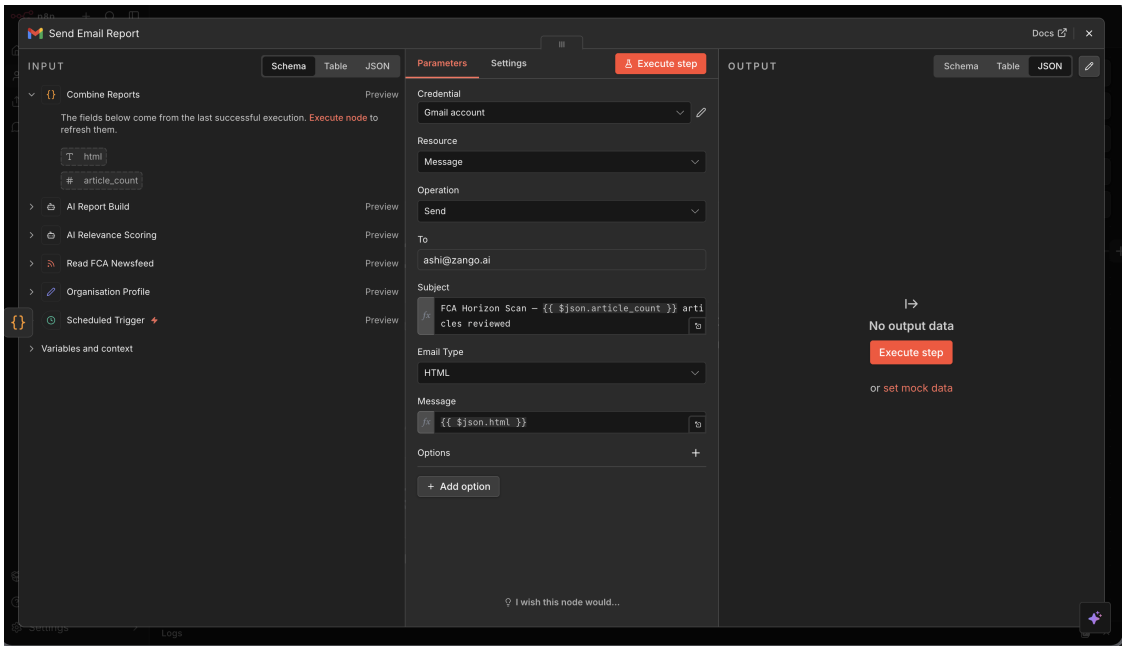
Email

The final node sends the combined HTML as a single digest email.

- **6.1** Click **+** after Combine Reports → search for **"Gmail"** → select it.
- **6.2** Rename to **Send Email Report**.
- **6.3** Select your **Gmail credential**.
- **6.4** Set **Operation** to **Send**.
- **6.5** Configure the email fields:

Field	Value
To	`your-email@example.com`
Subject	`=FCA Horizon Scan — {{ $json.article_count }} articles reviewed`

Field	Value
Email Type	`HTML`
Message	`={{ $json.html }}`



Send Email Report — Gmail node with expressions

Important

Set **Email Type** to **HTML**. Both the Subject and Message fields must start with = to enable expression mode.

Test & Run

Execute the Workflow

Test

1 Click the **Execute Workflow** button (bottom of canvas).

2 Watch data flow through each node — green ticks mean success. Item counts appear above each node.

3 Check that **Combine Reports** shows **1 item** output (not 20). This confirms aggregation is working.

4 Check your inbox for a single **"FCA Horizon Scan"** digest email with all article cards.

Testing individual nodes

Right-click any node and choose **"Execute this node only"** to test it in isolation.