Casebook for Newsroom AI Prototypes and Governance

REUTERS/Siphiwe Sibeko

Thomson Reuters Foundation

Patrick J McGovern FOUNDATION

REUTERS/Hannah McKay

Overview

This casebook documents what six small and mid-size newsrooms built, tested and learned during their strategic artificial intelligence (AI) adoption journey.

This includes three newsrooms in Latin America, and three in East Africa:

- Revista Anfibia (Argentina)
- Chequeado (Argentina)
- La Diaria (Uruguay)
- Lake Region Bulletin (Kenya)
- Radio Domus (Kenya)
- Orkonerei FM (Tanzania)

All six took part in Advancing Ethical and Strategic AI Adoption in Newsrooms, a programme run by the Thomson Reuters Foundation (TRF) in partnership with the Patrick J. McGovern Foundation. The partnership equips newsrooms to integrate AI responsibly into their editorial and business operations. The programme combines in-person training and mentoring on two tracks, (1) AI solutions and (2) policies that govern them.

The cases lay out what problems these newsrooms were trying to solve, their AI solutions, and guardrails they put in place for the ethical and responsible adoption of the technology. The objective of this casebook is to help other newsrooms weigh what AI could do for their team, and potentially build on what the six newsrooms already found, rather than starting from scratch.

How to use this casebook

PART I is organised by theme, opening with a featured case that walks through:

- editorial or operational problems each newsroom wanted AI to solve
- resulting AI solutions and how they were built
- results (including what did not work)
- insights from real use
- cost in both money and staff time
- shorter spotlight sections cover additional examples under the same theme.

PART II synthesises how all six newsrooms are governing AI, from the risks they identified to the guardrails they built.

PART III retraces how seven of the prototypes were built, one page per build.

PART IV synthesises the key takeaways from the casebook.

PART V includes a glossary of key terms

How we built this casebook

This casebook is built from interviews conducted by TRF with the six newsrooms in June 2026, and their written AI policies. Where attribution could breach a newsroom's privacy, examples are anonymised.

Interviews were conducted on Microsoft Teams and transcribed automatically. AI tools were also used to help extract, cross-check and draft the casebook. Every claim was verified against the source interviews and documents, and all content was reviewed, edited and approved by the TRF team.

What the cases show

A pattern holds across the six newsrooms. AI earned its place on narrow, repetitive tasks, like turning a weather bulletin into an article or reading thousands of comments. The affordable route was almost never building from scratch, but configuring an existing tool. And every tool that stayed useful had both a human checking its output and a named person responsible for keeping it running. While Part IV includes the full synthesis, Part I shows how these lessons were learned, including what didn't work, what each solution cost, and what each newsroom would do differently.

REUTERS/Temilade Adelaja

Part I: AI Solutions

01 **PART 1 • THEME 1**

Staff productivity and newsroom assistants

Orkonerei FM **TANZANIA**

Questions this section helps answer: Can a small team build an AI assistant without code? What does it cost? Where do humans stay in charge?

FEATURED CASE

OrkoBot, a co-pilot for a six-person station

PROBLEM: Orkonerei FM is a community radio station run by six journalists, which operates 24 hours a day. In the newsroom's own framing, the aim of the tool was "to minimize the workload of staff." They identified two challenges, (1) Social media demanded a constant presence, but content creation sat with a single social media manager. And (2) the Friday meeting to build the coming week's workplan (programmes, topics, who to interview, what to ask) "was taking a lot of time."

SOLUTION: A custom agent created inside ChatGPT, with no code on a paid subscription, which the newsroom named "OrkoBot".

The build started with hands-on AI training, where the team learned how to create an agent like this. They then discussed as a group what they needed, chose ChatGPT, built the agent, and fed it the station's documents, which covered the mission and vision as well as the tasks required. OrkoBot operates at two points in the workflow, running from idea to air.

- **Before reporting:** helps to refine a story's topic, angle, programme plan, interviewees, and the format, whether vox pop, package or narrative.

- **After reporting:** supports script drafting, followed by human editing and fact-checking, and then broadcast, website or social media.

In addition to OrkoBot, the station is also using Opus Clip to cut long videos into short TikTok and Instagram clips, and NotebookLM, for research and content development.

RESULTS: Three to five posts are produced per week, mostly by the social media manager, according to the station. Over the course of the project the station's following grew significantly, with TikTok growing more than sixfold from 2,000 to over 12,000 followers; Facebook and Instagram roughly two and a half times to 6,092 and 785; and a YouTube channel passed 500 subscribers in its first nine months since launch.

Scripts come out "easy, good and interactive", speeding up arranging news after fieldwork, and helping avoid re-covering stories already done, according to Orkonerei FM.

INSIGHTS: Prompting was the main challenge as it required the team to "write every detail in the prompt." Discovering that AI could make small mistakes, such as rearranging a script incorrectly prompted the station's mandatory human review rule. "AI supports our work, but editorial decisions, verification remain with journalists and editors."

COST: It took roughly three months to fully operationalise. The subscriptions cost around USD 1,431 for three 12-month licences (USD 650 for ChatGPT, USD 531 for NotebookLM, USD 250 for Opus Clip)

REPLICABILITY: This is a low-barrier no-code agent on a consumer subscription, grounded in the station's own mission and task documents. The critical input is training, agent creation, and prompt-writing. One caution: the agent runs on a third-party platform, and the station's own policy bars confidential sources and personal data from external AI tools without a risk assessment and editorial approval.

02 **PART 1 • THEME 2**

Verification and fact-checking

Lake Region Bulletin KENYA, Radio Domus KENYA

Questions this section helps answer: Can non-technical editors configure a fact-checking agent? How do you contain hallucinations? What does it cost to run?

FEATURED CASE

Lake Region Bulletin: a fact-checking agent built by non-technical editors

PROBLEM: Lake Region Bulletin (LRB) is a small digital newsroom in Kisumu, western Kenya, founded in 2021 to cover community stories dropped by a shrinking mainstream media. It runs with a small staff of eight to 10, and a network of around 25 independent contributors. It runs a fact-checking desk, and the manual work of collecting questionable content, running it through verification tools and producing debunks was slow and labour-intensive. The newsroom wanted to make this work easier for the existing team without having to hire.

SOLUTION: LRB built a custom agent on top of ChatGPT. The newsroom had used generative AI informally before, for basic image and text editing, but describes itself as initially wary of the technology. Developing a tool from scratch was too technical and expensive (cloud hosting, maintenance, monitoring) for their scale. Two editors worked on configuring the agent through natural-language instructions after learning they could "code using words." They explained to the agent who LRB is and what it wants to achieve, fed it instructions on how to fact-check plus reference resources, and uploaded several of LRB's own published fact-checks so it learns the house format. The agent traces

whether a photo or video has been published before and where. Most of the mis/disinformation they see is recycled, such as a Nigerian photo passed off as a Kenyan event, so tracing originals lets them debunk. Findings are fed back in and the tool drafts a fact-check story in LRB's format, with a human reviewing the output before anything is published. When outputs failed the "human test", they iterated on the prompt rather than rebuilding, adding resources and more specific instructions.

RESULTS: The agent reduced fact-checking time, freeing journalists for other tasks, and is now part of the regular workflow. All AI outputs are reviewed by human editors before getting published due to AI making mistakes or hallucinations, which is a persistent problem in Large Language Models (LLMs). In one illustrative case, the tool said Barack Obama is the current US president. Many websites block AI access, making timely verified information inaccessible. Using imprecise prompts adds to the problem, leading the tool into surfacing incorrect information from the internet.

INSIGHTS: The agent is part of the regular fact-checking workflow, but access is deliberately restricted to the two editors who run it, with wider onboarding only once the newsroom determines it's safe for others to use it. According to LRB, the desk covers on average two fact-check stories a week, plus general verification of figures in other reporting, including development, climate and science stories.

COST: The main cost is the paid subscription, around USD 25–30 a month. They tested packages from the smallest tier upward and found a modest one sufficient.

REPLICABILITY: This is a prompt-configured agent on a consumer AI subscription, built by non-technical editors and trained on the newsroom's own published stories. No coding, minimal budget. LRB's advice to peers is to define a specific task first and research which tool is strong for that task. Free tiers can be limiting and they keep changing. Newsrooms adopting this solution must use their in-house fact-checking formats, and tune it to their local context depending on the nature of the information that needs to be fact-checked. LRB's agent for example is tuned to recycled foreign content re-captioned for Kenya.

SPOTLIGHT

Radio Domus (Kenya): DigiKundi, fact-checking inside community WhatsApp groups

Unverified claims circulate through the community WhatsApp groups Radio Domus belongs to, and checking them takes local language and context. Unlike LRB's agent, DigiKundi relies on trained people before software. Instead of building a tool, the station trained the admins already active in the WhatsApp groups on fact-checking. They also set up a dedicated desk of six trained journalists, which has produced 13 fact-checked stories so far. Each story is logged in a growing archive that records the rumour, source links, verified response and date. This makes every check a reference material for the next one. The plan is to make the archive public and plug it into the station's in-development moderation tool, Safeguard (theme 8), so that past checks can surface when moderating audience comments.

03 **PART 1 • THEME 3**

Content repurposing across formats

Lake Region Bulletin KENYA, Orkonerei FM TANZANIA

Questions this section helps answer: How can a text-first newsroom produce video without field production? What do AI voices get wrong locally? What does it cost?

FEATURED CASE

Lake Region Bulletin: turning published stories into video explainers

PROBLEM: LRB began as a text-led publication built on long-form stories, but its online audiences increasingly prefer video. With a staff of eight to 10 and high turnover, producing video conventionally is time-consuming, and building real video capacity would have meant a major change to how the newsroom works. The objective was to produce short video explainers of already published stories without having to go back out to the field.

SOLUTION: The tool turns a published long-form story into a video explainer of one to three minutes. Once a story is published, an editor uploads the link to the tool, which summarises the story and generates the explainer. The explainer is posted on social media with a link back to the full story, acting as a teaser that leads audiences to the website. After trials with several AI tools, LRB settled on NotebookLM as the closest fit to what it was looking for. The same two editors who run the fact-checking agent run this tool, and production is kept at a central point rather than letting everyone generate video independently, a rule written into the newsroom's AI guide.

RESULTS: A few explainers have been produced and some published. Viewership has not changed much yet, which the newsroom attributes to having only just begun publishing. The refinement list is specific to tone, accent and logos. They are training it toward a more Kenyan voice because the default voice "feels too American or European." They plan to scale up to seven videos a week, compared to one or two previously.

INSIGHTS: The explainers are teasers, and not replacements of original stories. Short video reaches audiences where they are and pulls them to the long-form content on the website. The wider lesson is economic, according to LRB. AI can help when newsrooms cannot afford to support full multimedia production. However, it cannot do everything on its own, and people still need to be involved in the process.

COST: The clearest cost signal is the plan to hire two video journalists to support it. The subscription cost USD 858.40 for two 12-month licences (USD 452.40 for NotebookLM, USD 406 for ChatGPT).

REPLICABILITY: This is a low-cost route into video for text-first newsrooms. A text story link is inputted producing the explainer with no field production, and it's run by two editors. It needs a steady flow of long-form stories worth repurposing, social channels where short videos reach audiences the website does not, and a human checking each explainer before it goes out. The real adaptation work is voice. Default AI voices did not fit LRB's Kenyan audience, and localising tone and accent remains unsolved. The expectation is to trial several tools before one fits.

SPOTLIGHT

Orkonerei FM (Tanzania): Opus Clip, from long recordings to short clips

Orkonerei FM, a six-person radio station, is expanding onto TikTok, Instagram and YouTube, where short video performs and long recordings do not. It uses Opus Clip, an off-the-shelf AI tool, to cut long videos into short clips, alongside its newsroom assistant. Newsrooms whose raw material is audio and video rather than text can repurpose the same way, with no build required.

04 **PART 1 • THEME 4**

Routine content automation

La Diaria **URUGUAY**

Questions this section helps answer: Which routine tasks are worth automating? What can a two-week build deliver? What keeps a working tool running?

FEATURED CASE

La Diaria: Núbel, a weather-alert article generator

PROBLEM: Every red, orange or yellow weather alert issued by Uruguay's meteorology institute arrives each morning as a PDF, and La Diaria's web team had to turn each one into an article. The work was purely manual. Journalists essentially reprinted the press release, with "no soul" and no added journalistic value. A bad-weather stretch could mean doing it every morning for two weeks. The web team explicitly asked for help.

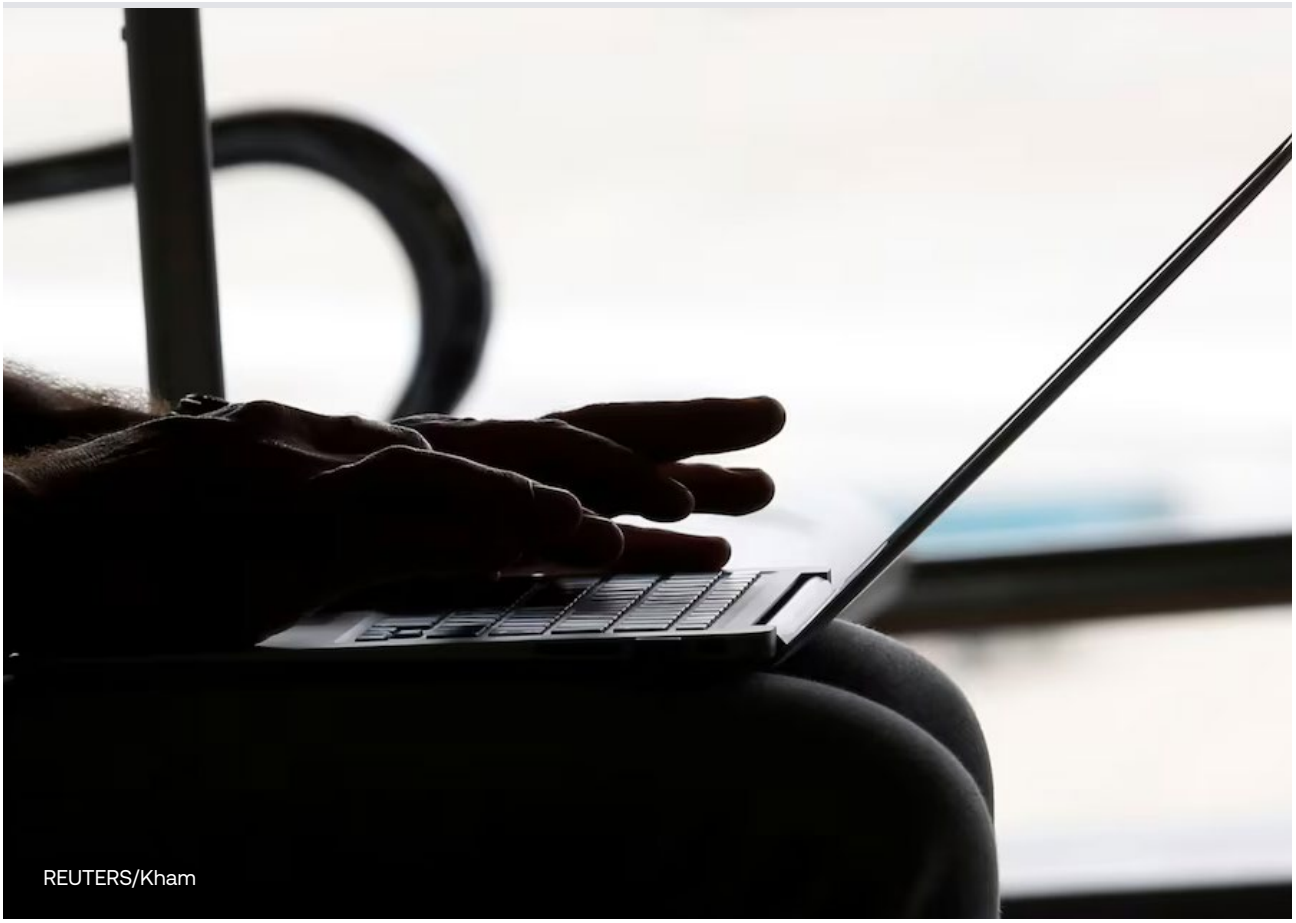
SOLUTION: Núbel takes the institute's PDF, extracts the information, and rewrites it as an article in La Diaria's house style, learned from the paper's own past forecast articles. A journalist checks every figure before publication, including wind speeds, rainfall amounts and the departments affected. The tool was built in a roughly two-week sprint using Google AI Studio, an environment for building AI prototypes. The only additional work was wrapping it in a web interface.

RESULTS: The tool cut the task by half. Producing an alert article went from about 20 minutes before Núbel to 10 minutes including supervision. By mid-2026 the tool was not running, even though the need remains.

INSIGHTS: The staff member who drove it moved to the radio section, and no meeting or routine exists to check whether the newsroom's tools still work. "We need to ensure that we maintain these tools." Without an institutional owner, even a successful build can quietly stop. Adoption was immediate because the tool answered a need the affected team had voiced themselves.

COST: A two-week sprint by the participants, with no internal IT time.

REPLICABILITY: The ingredients are a structured official bulletin in a consistent format, a set of the newsroom's own past articles to teach the style, and a named person to verify the figures on every output. La Diaria needed no internal IT and built it in two weeks in a free environment.



REUTERS/Kham

05 **PART 1 • THEME 5**

Content discoverability and search optimisation

La Diaria **URUGUAY**

Questions this section helps answer: How do you fit AI inside an existing publishing workflow? What drives adoption? What happens when a tool loses its owner?

FEATURED CASE

La Diaria: TIA, a one-button title assistant that put stories back in search

PROBLEM: Journalists at La Diaria were not writing meta titles, the short titles that search engines and Google Discover display. It was an extra step nobody took at the end of a long writing day, so articles were not appearing on Google Discover or in search, keeping content confined to the existing subscriber community. Reach beyond subscribers matters commercially as subscriptions account for more than 80% of the paper's income.

SOLUTION: TIA ("Titulation by IA") adds one button inside the Content Management System (CMS) publishing flow. Before hitting publish, the journalist presses it and receives three editable meta-title suggestions. Human review is built into the design, and access is permission-controlled. The tool was calibrated with the paper's own articles paired with correct meta titles. The internal IT team did the technical work, its first AI project, across roughly three months. The tool was deliberately fitted into the existing workflow as a tiny step rather than built as a standalone product.

RESULTS: Adoption grew by word of mouth, from three initial editors to all web editors and journalists, with colleagues requesting access. The newsroom regards that organic demand as the strongest proof of success. It worked best on short, fast-turnover content like breaking news, sports, justice and crime, where reporters said it produced more concise titles in fewer words. It performs poorly on long articles, such as six-page science pieces, which the design never considered. There is no analytics evidence that it improved long-term discoverability. Impact measurement stayed informal; the users' relief at not having to invent another title was considered sufficient, since that was the original motivation.

INSIGHTS: Eight to 10 journalists used it, mostly editors across sports, environment, justice, politics and web. One use that nobody designed for involved journalists feeding the suggestions back into the article's actual headline. The harder lesson is how the tool would be used afterwards. It is not currently working, and the team now brings titles to the staff member who built it, who runs them through an early prototype manually. Like Núbel (theme 4), a working prototype needs someone checking whether it still works, needs upgrades, or has stopped because a payment lapsed. Without that, the tool depends on individuals and never becomes institutional.

COST: A monthly AI subscription of USD 190. The indirect cost was roughly three months of the internal IT team's time on top of normal duties.

REPLICABILITY: A one-button, human-supervised assist embedded and calibrated with a handful of house-style examples. It needs a CMS the newsroom can modify, some internal IT capacity, and style examples built from its own pages. It also needs to be designed for long-form content from the start.

06 **PART 1 • THEME 6**

Visual content production and design

Anfibia **ARGENTINA**

Questions this section helps answer: When is it worth hiring a developer? Where should AI imagery stop? Who are the real end users?

FEATURED CASE

Revista Anfibia: a Slack chatbot for routine visual content

PROBLEM: Every branch needs visual content for social media, from flyers and promotional images to event invitations, and all of it lands on an art department of two to three people. In addition to the magazine, Anfibia runs:

- A podcast production company
- A non-fiction theatre lab
- A large education and training area
- A membership programme.

Many requests were simple and repetitive, yet still required design skills, so orders and revision rounds piled up. Automating routine requests would free the art team and clear the backlog.

SOLUTION: A chatbot that lives in Slack and generates routine visual content on request. About five designated users, one per area, sit in a shared channel and submit image requests. The system is ChatGPT wired into Slack, with n8n (a paid workflow-automation tool) running underneath. Every output travels with the requester's brief to the art director, who must approve it before anything is used. The prototype handles only the easiest, lowest-risk content: workshop flyers, education and membership communications. It is deliberately not used for imagery attached to journalistic articles. Anfibia started with an internal needs assessment, interviewing each area's leaders about what they needed. It wrote a brief, posted a hiring call for developers on social media, interviewed three software engineers and selected one it had worked with before. The build took roughly two to three months, followed by user testing and two short developer-led training sessions for area leaders.

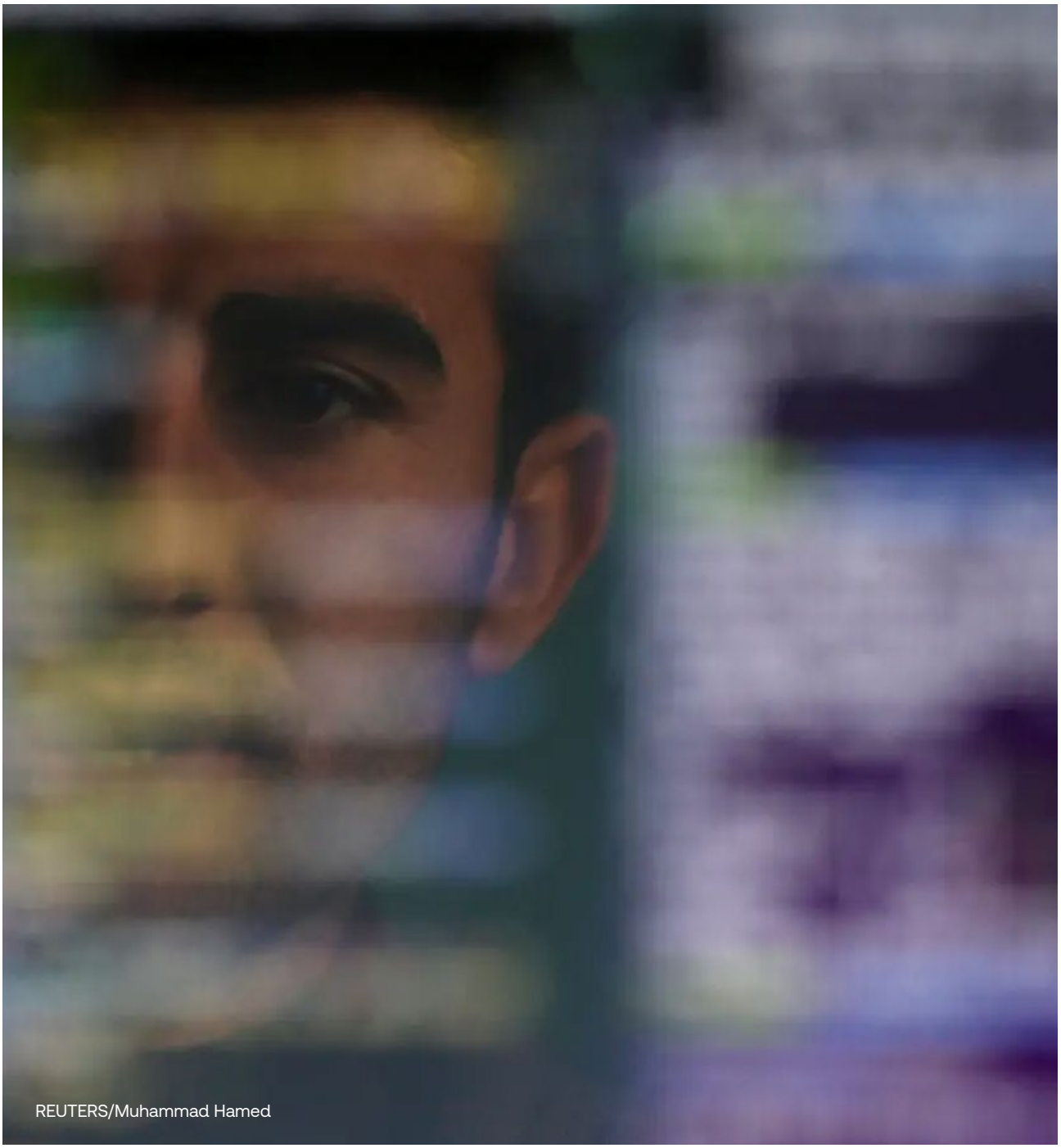
RESULTS: The tool is in active testing, producing an estimated five to seven pieces of content a week that would otherwise have gone to the art department, according to Anfibia. Not everything generated is used; some outputs contained mistakes and were discarded.

INSIGHTS: The boundary between promotional artwork and journalistic imagery is a priority for Anfibia. The year before the prototype, on a high-volume publishing day, an AI-generated Instagram gallery for an article on Patagonia and the climate included a map labelling the Falklands/Malvinas as British, published without supervision. Argentinian readers reacted furiously; the team describes it as a traumatic episode. Anfibia has since restricted AI imagery to creative and promotional artwork, never journalism visuals. The caution is also strategic. Anfibia publishes critical pieces about AI, including a five-episode podcast series, so its followers watch what it does with the technology. Careless AI content would cost Anfibia more than it would larger outlets.

The main thing the team would do differently is involve the whole social media team from the beginning. The project was designed from the art department's perspective, but the daily end users turned out to be the social media team, who joined mid-project, so the feedback that mattered most arrived late.

COST: The external developer's fee for the two-to-three-month build, the n8n licence, two or three ChatGPT seats, and the newsroom's pre-existing design-software subscriptions.

REPLICABILITY: Interview each area about its needs, write a brief, select a developer competitively, build iteratively, and have the developer train the users. Pilot on low-risk content with mandatory human sign-off, and involve the real end users from the beginning. Anfibia found its developer and free training through a tech reporter's personal network, so they budget explicitly for recruitment and training. Its multi-branch structure also creates unusually heavy flyer demand; a single-title newsroom should check it has enough requests to justify the build.



REUTERS/Muhammad Hamed

07 **PART 1 • THEME 7**

Audience listening and comment analysis

Chequeado ARGENTINA;
see also Radio Domus's Safeguard in theme 8

Questions this section helps answer: What does it take to read 10,000 comments a week? How much checking does internal-only output need? What does it cost?

FEATURED CASE

Chequeado: Nunchi, an agent that reads the comments no human could

PROBLEM: Chequeado watched metrics but was not reading the comments on its website or social media. The scale made manual reading impossible at roughly 10 Instagram posts a week, averaging a thousand comments each; around 10,000 comments a week from Instagram alone. Website comments are far fewer but "valuable" and need curation. The value in the comment section went unextracted.

SOLUTION: Nunchi listens to, organises and prioritises audience conversations and turns them into editorial insights. Comments are collected by n8n (an open-source workflow automation tool), sent to an AI agent on the OpenAI Platform for categorisation, and stored in a PostgreSQL database. The results surface as Slack alerts and a dashboard, used both to check the AI's categorisation and for broader analysis. This includes spotting how one topic is generating political debate while another draws positive engagement. Each alert summarises the signal, suggesting an editorial action such as adding a methodological clarification in the comments. It includes real comments alongside the AI's reading, which lets staff check the system against its source and learn to trust

it. An external n8n specialist set up the initial workflow while training the team to support the tool themselves. Free and self-hosted, n8n runs on Chequeado's existing cloud. The product team took it from ideation to an implemented Minimum Viable Product (MVP).

RESULTS: Initially worried about AI or producing simplistic categories for Spanish-language comments, the team received positive results from the tool. One or two pieces have been published off the back of it, correcting or clarifying something the comments flagged.

INSIGHTS: The early trap was trying to do too much by extracting too many aspects from each comment. Narrowing down to fewer dimensions helped the team achieve better progress. Another lesson is timing. Defining the right moments to involve the newsroom, balancing showing a prototype too early against too late for useful feedback.

No one is assigned to verify every categorisation; the team keeps an eye on it and reads the raw comments when editorial flags a bad output. That lighter validation is accepted because the output is internal-only. Anything audience-facing would require stricter processes. The roadmap points into using more platforms as data sources, new use cases on the accumulated database such as trends and impact measurement, adoption in other areas of the organisation, and possibly opening the tool to other newsrooms.

COST: Application Programming Interface (API) usage runs roughly USD 50–100 a month at current comment volume, expected to rise if other platforms are added. The n8n licence costs nothing self-hosted. Internal build time was roughly 10 to 15 hours a week.

REPLICABILITY: The stack is inexpensive using open-source collection, a consumer AI API, a database and Slack. Narrowing the scope is important. Over-scoped categorisation is the predictable trap, and raw comments beside AI output are what build newsroom trust. The categorisation Chequeado settled on is based on sentiment in six registers (analytical, critical, frustrated, hostile, ironic and positive), intent, and signals including support for the outlet, criticism of it, and conspiracy theories, with a reviewer field built into the interface. The build pays off mainly for newsrooms whose comment volume is beyond manual reading; volume and platform mix set both the value and the running cost.

08 **PART 1 • THEME 8**

Content moderation and community safety

Radio Domus KENYA

Questions this section helps answer: What do global moderation tools miss in local languages? Build, buy, or integrate? Which guardrails come first?

FEATURED CASE

Radio Domus: Safeguard, moderation built for languages global tools miss

PROBLEM: Radio Domus is a community radio station in Kenya with growing digital conversations around its content on Facebook and YouTube. Off-the-shelf moderation kept failing in both directions. Global tools mis-flag benign local usage, such as an audience member commenting that a show “was a bomb”, meaning it was excellent, and Facebook flagged it as terrorism. Meanwhile, clearly abusive Swahili and Sheng terms pass unflagged. Manual moderation ate limited staff time. The station wanted moderation that understands how its community actually speaks.

SOLUTION: Safeguard is a locally built, AI-assisted moderation tool with:

- keyword filtering that flags harmful or misleading content using locally relevant terms
- post control, with review, approval and escalation before anything goes public
- capture of audience feedback, complaints and engagement signals
- ethical guardrails ensuring human oversight, data protection and accountability

Its core is a keyword library the station compiled itself of over 500 terms in English and Kiswahili, covering hate speech, ethnic discrimination, incitement, profanity and election manipulation, drawn from how the community uses language. Market tools alone would not work for a community station, so Radio Domus is building Safeguard on a paid Claude subscription while integrating third-party tools such as NapoleonCat and the Meta dashboard. Two external organisations were brought in for the front end, while the team learned the backend so the skills stay in-house. The human loop lives outside the tool, in Slack, where journalists submit drafts, an editor reviews, revisions follow, and approval precedes publication. The tool alerts whoever is online for post control, and a human makes the final call on everything.

RESULTS: Safeguard is not yet public-facing. It is in phased internal adoption, tested by 11 users (eight staff, three interns) alongside existing tools. No formal impact analytics exist yet. The gains are meant to be at the workflow-level, providing structured identification of harmful content, quicker internal review, clearer escalation decisions, more consistent moderation, and categorisation of comments to track patterns. In one documented example of success, abusive Swahili and Sheng terms that a generic English-trained tool would miss were flagged in its own Facebook comments.

INSIGHTS: The operating model is simple. AI suggests, a human reviews in Slack, the decision is logged. Core protections include data minimisation, role-based access to audience data, keeping sensitive data out of open AI tools, and reviewing false positives and negatives for bias. A moderation decision log builds newsroom-specific precedent over time. Safeguard is strongest as a content moderation and audience-intelligence prototype, according to the station. The fact-check archive and revenue potential are future extensions. Next steps include completing the dashboard, expanding the keyword library with election-risk terms ahead of Kenya's 2027 election, integrating the fact-check archive, and piloting with partner community radios.

COST: Deliberately low, but the tech stack is wide, including subscriptions to Claude, ChatGPT, Slack, NapoleonCat and Trello, a Microsoft 365 trial. A single AI subscription left the social media side uncovered, so complementary tools had to be budgeted.

REPLICABILITY: The method is transferable. Newsrooms need to run their own keyword-gathering exercise in their community's languages, choose build-and-integrate rather than build-or-buy, keep the human loop in cheap common tools, and log moderation decisions to build your own precedent. External technical help is expected, and budget for several tools rather than one subscription. It's recommended to start with the one or two platforms where a newsroom's audience actually is.

REUTERS/Mike Hutchings

REUTERS/Louafi Larbi

Part II: Governance

01 Key risks the newsrooms identified

Declining audience trust tops the newsrooms' list of risks. A subscription-funded paper frames every AI decision against "the trust of the people that are trusting us with the money by month." A fact-checker concluded it was "impossible to be transparent and not be transparent on how we use AI." Community stations report audiences detecting foreign-sounding output ("this is not you guys; this English is totally foreign"). One editor's rule of thumb is that "anytime they think this is AI, there's that negative connotation." Two newsrooms learned that even disclosed, guideline-compliant AI use can draw criticism. Compliance does not guarantee acceptance.

Misinformation reaching the audience is the second near-universal risk. A hallucinating agent asserting Barack Obama is the current US president; AI-supplied statistics with no provenance; a script rearranged incorrectly without the journalist noticing; a published caption that audibly read as AI; and, at one Latin American newsroom, an AI-generated image with a politically explosive map error published without supervision. Every error that stayed out of print was caught the same way with a human reviewing before publication.

Uncontrolled individual use is another risk. "If you have each person doing it their own way, you're likely to leave room for more mistakes," noted an editor, adding that uncoordinated adoption erodes trust between colleagues. One newsroom names the pattern shadow AI, with everyone using it without transparency, and staff declining even the newsroom's own sanctioned tools rather than admitting they use AI at all.

Over-delegation to AI is another risk. "Once people understand AI can do it, they will just be watching AI do it," against the counter-principle that the machine must not decide.

External contributors submitting AI-generated pieces. More editors experiencing an increase in AI-generated submissions further jeopardising audience trust if uncaught before publication.

Tools going dormant after launch. Two of the prototypes documented here are no longer in regular use: a weather-alert generator whose champion changed roles, and a title assistant now used manually by its creator when needed. The causes differ, but the pattern remains the same. Without a named owner and a routine check on whether tools still work, even a successful tool build stops working as intended.

Governance failures as risks in themselves. Policies that go dormant, the absence of feedback channels through which staff could raise the alarm, and low editorial buy-in, with end users brought in late or the newsroom involved at the wrong moments.

02 Mitigations and guardrails

Human review before publication is non-negotiable. What varies is how it is applied through routing every output to a named approver (an art director, or an editor at the end of a Slack submission chain). A layered sequence in which the journalist reviews first, then a role-specific checker, editor for on-air content and social media manager for platforms. Internal-facing outputs are checked lightly and audience-facing ones strictly.

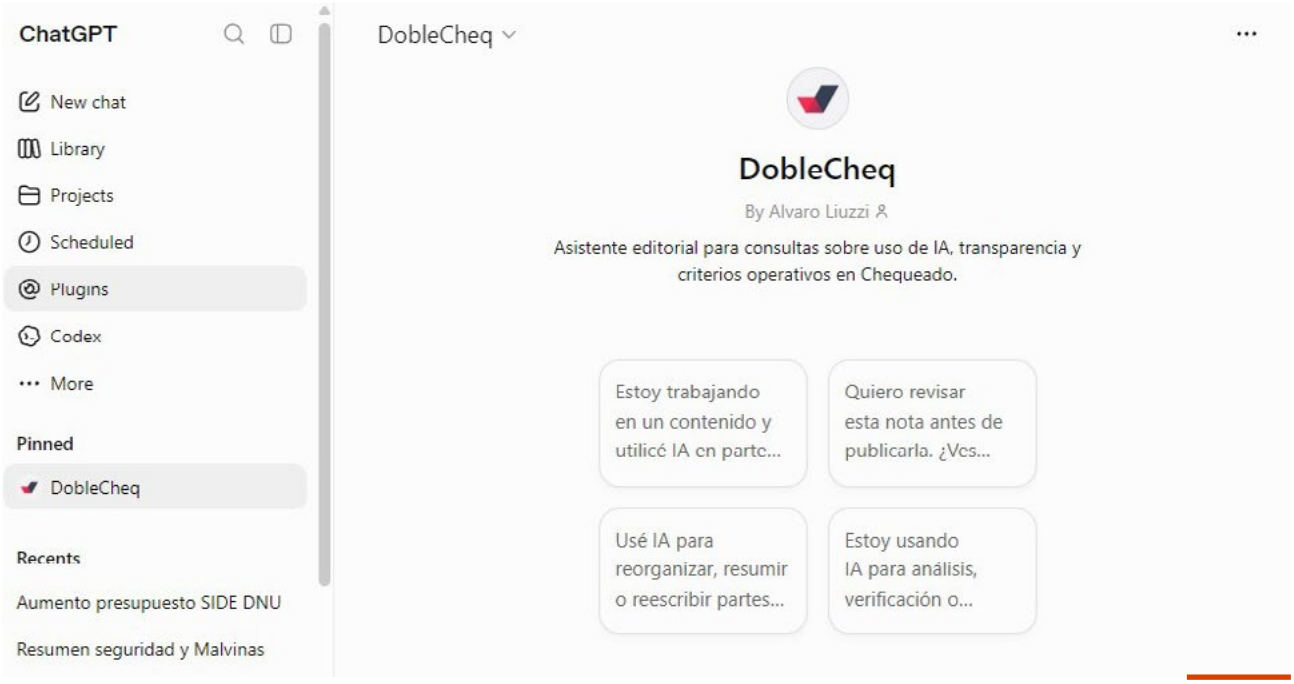
Scoping and red lines. AI can be used for promotional artwork but never to generate editorial imagery. Sensitive functions such as fact-checking centralised in one trained agent under two editors, with lower-risk uses left open to staff. Permission-controlled AI features in the publishing system. One written policy adds an approved-tools list, which an editor approves before journalists use them, reviews the list quarterly, and bars unapproved platforms from sensitive editorial material. Another pairs its whitelist with a lifecycle of proposal, risk assessment, pilot testing, conditional approval and ongoing review, under which a tool can be suspended or withdrawn. It also tiers uses by risk, from routine tasks needing no extra approval to prohibited uses touching source confidentiality or unpublished material, and for major stories journalists document which AI tools were used, for what purpose, how outputs were verified, and what disclosure was decided.

Disclosure to the audience is agreed in principle. Whether to disclose every time or some of the time, generically or naming the tool, on air or per post remains a challenge. Chequeado publishes a visible disclosure on any product where AI played a relevant role, with a ready-made formula stating that AI tools assisted the production under the outlet's editorial standards and that the team reviewed and verified all content, plus credits or watermarks on audiovisual material. Orkonerei requires disclosure when AI generated or significantly rewrote a script, summarised a bulletin, translated in a way that affects meaning, or produced synthetic voices or images; not for spelling checks, formatting or internal transcription. Radio Domus opts for disclosures such as a short tagline, a caption or credit on social media, or an editor's note voiced on air. This is extended when appropriate to publishing the prompts fed to the model.

Document architecture varies. Chequeado separates a stable public policy from internal, frequently updated protocols so the public document doesn't go stale. La Diaria tiers its documents by audience, with a public declaration of its philosophy, an internal summary, and the full policy held by an AI-use committee, alongside a misuse track covering protocols, sanctions and responsibility.

Verification rules. Any AI-supplied fact or figure must be traced to a source before use, and every number in an auto-drafted article checked by a journalist. Chequeado's published policy states that AI can never be a primary source of information. One tool carries the rule into its own interface, with a standing footer telling users to always verify the original sources.

Culture-building and internal training. Language "clinics" that feed a community's approved registers back into the tools. Physical banners stating that AI is allowed in moderation, with checklists. A log of past moderation decisions to build in-house precedent. Talking resistant staff through why the sanctioned, house-trained tool beats their personal one. Chequeado pairs its policy with a protocol of operational criteria, and staff can ask an internal AI assistant, DobleCheq, when unsure what the policy allows.



DobleCheq AI policy assistant

Data protection. Radio Domus collects only the data that moderation and feedback require. They restrict who can view, edit or export audience data, keeping sensitive data out of open AI tools and reviewing false positives and negatives for bias. Unpublished investigations, embargoed stories and internal editorial deliberations stay out of external AI systems. Chequeado prohibits entering sensitive information (such as anonymous sources, personal data or confidential documents) into external AI tools, and is the only policy to cover intellectual property, preferring tools with authorisation criteria and renouncing no rights over its own original content. Orkonerei goes furthest into a community newsroom's particulars. Confidential sources, whistleblower recordings, interviews with survivors of gender-based violence, sensitive land-rights and legal cases, and community members' personal data must not enter third-party AI platforms without a risk assessment and editorial approval. Journalists first ask where the data is stored, whether the platform trains on it, and whether a source could be exposed.

03 Shared principles for governance

AI assists, humans decide. “Journalists must be journalists, even if the AI is there.”

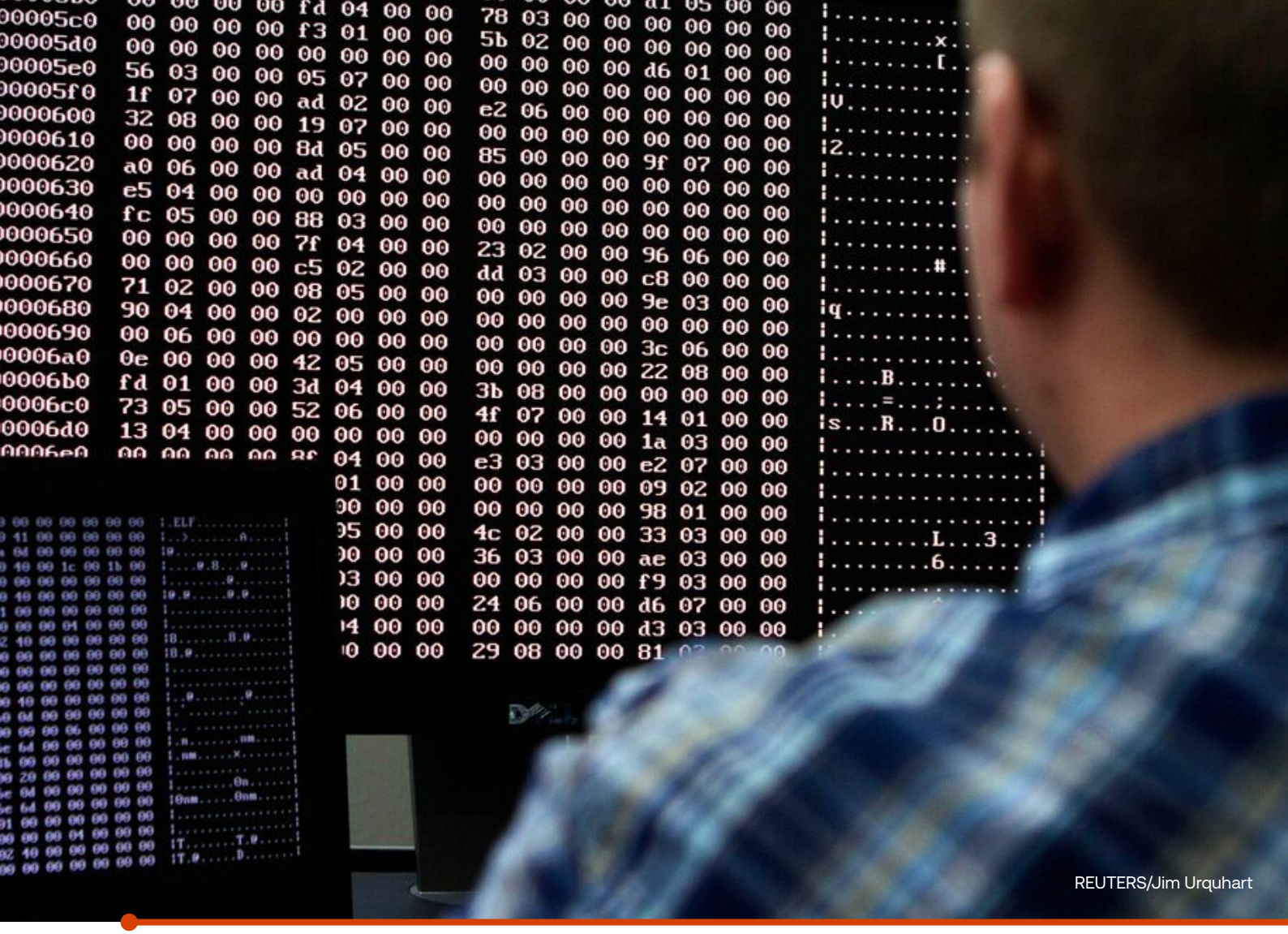
Expand capability, don’t replace people. Automation exists to free small teams for higher-value work.

Transparency as identity, not compliance. Disclosure framed as who they are rather than a rule. Honesty about AI use as a duty to a paying or watching audience, including the disarming formulation that anyone claiming in 2026 not to use AI “is lying to you.” The real obligation is to say how you check it and own the mistakes. Chequeado makes owning them concrete. Errors derived from AI use are corrected with the same visibility the original had.

Sound like yourself. Identity preservation as a governance matters. Growing “without losing the soul,” output in the community’s own language and register, and house format as the thing a tool must learn before it is trusted.

Policies are living documents designed for revision. They are published as a starting point or guide under continuous, ongoing review, and a draft following national regulation.

A duty to educate and protect the audience about AI. Newsrooms should help their audience understand AI, treating AI literacy as part of news literacy. Their disclosures are written to teach as well as to declare.



REUTERS/Jim Urquhart

Part III. How the prototypes were built

01 How Orkonerei FM built OrkoBot

WHAT IT IS. A custom agent, a tailored AI assistant configured for specific tasks without coding, was built inside ChatGPT by a six-journalist community radio station in Tanzania. It helps plan the weekly programme schedule, structure and rearrange scripts, and produce social media captions.

STEP 1: Get trained first. The build followed hands-on AI training; the builder didn't have the knowledge to create an agent before. Two skills matter most: creating a custom agent, and detailed prompt-writing.

STEP 2: Define the problem as a team. The whole team discussed what they needed.

STEP 3: Choose the platform. ChatGPT, on a paid subscription.

STEP 4: Create the agent and ground it in your own documents. They fed it the station's mission, vision and task descriptions so it can understand their mission and vision and the tasks needed.

STEP 5: Test, then open to everyone. Tested first, then shared with all team members.

STEP 6: Fix it into the workflow with human checks. The documented chain starts with a story idea, refined with OrkoBot for topic, angle, programme plan, interviewees and format. This is followed by field reporting, AI-supported script drafting, human editing and fact-checking, and then broadcast, website or social media. The review rule is written into the station's policy. The journalist reviews first, then a role-specific checker.

BUDGET AND TIMELINE. Roughly three months from training to team use. Costs cover the AI and internet subscriptions. The three AI licenses together cost USD 1,431 for 12 months.

WHAT TO EXPECT. The challenge is prompting, not the technology itself. Results at Orkonerei show three to five posts a day and marked follower growth over the project, with the newsroom's own careful caveat that AI workflows "combined with newsroom efforts" contributed, alongside a YouTube channel launch.

CHECK BEFORE YOU COPY.

Which languages the agent must work in, and how well it performs in yours.

02 How Lake Region Bulletin built its fact-checking agent

WHAT IT IS. A custom agent built on top of ChatGPT by two editors, neither with a technical background, at a small digital newsroom in Kisumu, Kenya. It traces whether suspect photos and videos have been published before, helps locate originals, and drafts fact-check stories in the newsroom's own house format. A human reviews everything before publication.

STEP 1: Get trained. The team describes itself as initially wary of AI, with no strong background in it. Hands-on training helped them learn they could "code using words," configuring an agent purely through natural-language instructions.

STEP 2: Define one specific task. Make the fact-checking desk's manual work faster. Most mis/disinformation they see is recycled, such as a Nigerian photo passed off as a Kenyan event, so the task was tracing where content first appeared.

STEP 3: Choose a custom agent over building from nothing. Building a new tool was judged too technical and too expensive (cloud hosting, maintenance, monitoring) for their scale. A configured agent on a paid consumer subscription was affordable and fast.

STEP 4: Configure it in three steps. Explain to the agent who the newsroom is and what it wants to achieve. Feed it with instructions on how to fact-check, plus reference resources. Upload several of your own published fact-checks so it learns the house format.

STEP 5: Run it with a human supervision. The agent traces provenance; the editors debunk; findings are fed back in and the tool drafts a fact-check story in house format. A human reviews before anything is published. No AI-caused mistake has ever gone to press.

STEP 6: Iterate the prompt, not the tool. When outputs failed the "human test," they added more resources and more specific instructions to narrow what it searches for.

STEP 7: Restrict access until you trust it. Only the two editors use it, "to regulate its access for proper management," with wider onboarding only "once convinced it is safe."

TIME AND MONEY. The main cost is the paid subscription, around USD 25–30 a month, found sufficient after testing tiers from the smallest upward with unlimited usage. A secondary cost is supporting staff home Wi-Fi, since the team works remotely.

WHAT TO EXPECT. Persistent hallucinations (in one case the tool asserted Barack Obama is the current US president), which is why the human stays in the loop. Some websites block AI access, so information is sometimes unreachable. Imprecise prompts send the tool "fishing" wrong information. Free tiers are limiting, and the tools keep changing. The desk now averages two fact-check stories a week, plus general verification of figures in other reporting.

CHECK BEFORE YOU COPY.

You must supply your own house-format fact-checks, and your own local threat picture. LRB's agent is tuned to recycled foreign content re-captioned for Kenya; yours needs tuning to whatever circulates where you are.

03 How Chequeado built Nunchi, its audience-comment agent

WHAT IT IS. Nunchi collects audience comments, categorises them with AI, and surfaces the results as Slack alerts and a dashboard, turning conversations the newsroom could never read manually into editorial insights. Built from ideation to prototype to an implemented MVP. Stack: n8n for collection, the OpenAI Platform for analysis, PostgreSQL for the database, Claude Code in development, integrated with Slack.

STEP 1: Start from a clear problem, with numbers. Chequeado watched metrics but was not reading its comments of roughly 10 Instagram posts a week averaging a thousand comments each, around 10,000 a week from Instagram alone.

STEP 2: Buy the expertise, keep the capability. An external n8n specialist set up the initial workflow, deliberately skipping the learning curve while training the team, so they can support the tool themselves.

STEP 3: Check what is free before you pay. They expected to pay for an n8n licence, then discovered it is open source and free if self-hosted, which reduced operating costs and freed that budget.

STEP 4: Narrow the scope until it works. They first tried to extract many facets per comment and stalled. Cutting to far fewer dimensions unblocked progress. Over-scoping the categorisation is the predictable trap.

STEP 5: Involve the newsroom and show the raw material. Three or four one-hour UX sessions with the wider team. The key decision is to include the real comments inside each Slack alert, so the team can check the AI against the source and trust the system. Their own hindsight lesson is to define the right moments to involve the newsroom, balancing showing a prototype too early against too late for useful feedback.

STEP 6: Calibrate checking to risk. No one is assigned to verify every categorisation; the team keeps an eye on it and reads the raw comments when

editorial flags a bad output. That lighter validation is acceptable because the output is internal-only. Audience-facing output would require stricter processes.

TIME AND MONEY. API usage runs roughly USD 50–100 a month at current volume, expected to rise if other platforms are added. The n8n licence costs nothing self-hosted. Build time was around 10 to 15 hours a week.

WHAT TO EXPECT. Spanish-language comments worked far better than initial thoughts. One or two pieces have been published based on insights the comments surfaced. The roadmap includes more platforms, new use cases on the accumulated database, adoption beyond the newsroom, and possibly opening the tool to other newsrooms as a sustainability model.

CHECK BEFORE YOU COPY.

Your comment volume and platform mix set both the value and the cost. The categorisation they settled on focused on sentiment in six registers, intent, and signals such as support, criticism of the outlet, and conspiracy theories.

04 How Radio Domus is building Safeguard

Safeguard is a prototype in refinement, not a finished product; this guide covers the path so far.

WHAT IT IS. An AI-assisted moderation tool built by a Kenyan community radio station, with four parts: keyword filtering of harmful content using locally relevant terms, post control with review and escalation, audience feedback capture, and ethical guardrails. It is tested internally by 11 users (eight staff, three interns) alongside existing tools, and is not yet public-facing.

STEP 1: Start from a local failure of global tools. Facebook flagged an audience comment calling a show “a bomb” (meaning excellent) as terrorism. Meanwhile genuinely abusive Swahili and Sheng terms pass unflagged. That gap, in both directions, is the problem Safeguard exists to close.

STEP 2: Build a keyword library from how your community actually speaks. Over 500 terms across English and Kiswahili covering hate speech, ethnic discrimination, incitement, profanity and election manipulation, fed into the AI agent to contextualise moderation. Expansion planned ahead of Kenya's 2027 election. A starter dataset exists and is shareable.

STEP 3: Choose build-and-integrate, not build-or-buy. Market tools alone would not work for a community station, so they're building Safeguard with a paid Claude subscription while integrating third-party tools (NapoleonCat, the Meta dashboard) and feeding local knowledge into them.

STEP 4: Get help for what you cannot do, but learn the backend. The second phase onboarded two external organisations for the front end, judged too complex to do alone, while the team deliberately learned the backend so the skills stay in-house.

STEP 5: Put the human loop in cheap, common tools. The human-in-the-loop mechanism lives in Slack, not inside Safeguard where journalists submit drafts, an editor reviews for accuracy and balance, revisions follow, approval precedes publication. The tool alerts whoever is online for post control. A human makes the final call on everything.

STEP 6: Write the ethics before the automation. The operating model revolves around AI suggesting, human reviewing, and decision logging. Core protections include data minimisation, role-based access to audience data, no sensitive data in open AI tools, and review of false positives and negatives. The team says building these guardrails first deliberately slowed automation.

STEP 7: Scope to your highest-volume platforms and test with a lean team. Facebook and YouTube first; Instagram, X and TikTok deferred. Interns support testing and iteration because everyone multitasks.

TIME AND MONEY. Deliberately low cost, but the stack is wide. Subscriptions to Claude, ChatGPT, Slack, NapoleonCat and Trello, a Microsoft 365 trial, and staff training.

WHAT TO EXPECT. They underestimated the technical expertise required. One AI subscription alone left the social media side uncovered, so complementary tools had to be factored in. The dashboard is unfinished, no impact analytics exist

yet. Early observed gains are at the workflow-level, with structured identification of harmful content, quicker review, clearer escalation, more consistent decisions.

CHECK BEFORE YOU COPY.

The keyword library does not transfer: you must run your own gathering exercise in your own community's languages. Expect to need external technical help for the front and backend, and budget for a small stack of tools, not one subscription.

05 How La Diaria built Núbel, its weather-alert article generator

WHAT IT IS. Núbel takes the weather alerts Uruguay's meteorology institute issues as PDFs each morning and rewrites them as publishable articles in La Diaria's house style. A journalist checks every figure before publication. Built in a two-week sprint, with no internal IT involvement.

STEP 1: Start from a task nobody wants. The web team had to turn every red, orange or yellow alert into an article manually, essentially reprinting a press release with "no soul" and no added journalistic value. A bad-weather stretch meant doing it every morning for two weeks. The team asked for help, which made adoption immediate later.

STEP 2: Use a free prototyping environment. The tool was built in Google AI Studio, an environment for building AI prototypes, in a roughly two-week sprint. The only additional work was wrapping it in a web interface journalists could use.

STEP 3: Teach it your style with your own articles. Núbel learned the house style from the paper's own past forecast articles, so its output reads like La Diaria, not like a press release.

STEP 4: Keep a human on the figures. A journalist verifies every number before publication, including wind speeds, rainfall amounts, the departments affected. The tool drafts; it does not publish.

STEP 5: Name an owner before the sprint ends. The tool cut the task from about 20 minutes to 10, and adoption was immediate. By mid-2026 it was nonetheless not running. The staff member who drove it changed sections, and no routine existed to check whether tools still work.

TIME AND MONEY. A two-week sprint by the participants, a free build environment, no internal IT time.

WHAT TO EXPECT. Fast results on any structured official bulletin that arrives in a consistent format. The risk is not the build but the afterlife. Without an institutional owner, even a successful tool quietly stops.

CHECK BEFORE YOU COPY.

You need a bulletin with a stable format, your own past articles to teach the style, and a named person to verify figures on every output.

06 How La Diaria built TIA, its one-button title assistant

WHAT IT IS. TIA ("Titulation by IA") is a button inside La Diaria's CMS. Before hitting publish, a journalist presses it and receives three editable meta-title suggestions, the short titles search engines and Google Discover display. Built by the internal IT team as its first AI project.

STEP 1: Find the friction point. Journalists were not writing meta titles. It was an extra step nobody took at the end of a long writing day, so articles were not appearing in search or on Google Discover. With subscriptions over 80% of income, reach beyond subscribers matters commercially.

STEP 2: Fit the tool into the workflow, not beside it. TIA was deliberately built as a tiny step inside the existing publishing flow rather than a standalone product. One button, three suggestions, all editable. Human review is the design, not an add-on.

STEP 3: Calibrate with house examples. The tool was calibrated with the paper's own articles paired with correct meta titles.

STEP 4: Give it to the whole IT team. The internal IT team did the technical work across roughly three months on top of normal duties, its first AI project.

STEP 5: Control access, then let demand spread it. Access is permission-controlled. Adoption grew by word of mouth from three initial editors to all web editors and journalists, with colleagues requesting access, the newsroom's strongest proof of success.

TIME AND MONEY. A monthly AI subscription, plus roughly three months of the internal IT team's time.

WHAT TO EXPECT. Strongest on short, fast-turnover content such as breaking news and sports, but poor on long articles the design never considered. Journalists also fed the suggestions back into actual headlines, a use nobody planned. Like Núbel, the tool later stopped working for lack of an owner.

CHECK BEFORE YOU COPY.

You need a CMS you can modify, some internal IT capacity, and house-style examples from your own pages. Design it for your long-form content from the start.

07 How Anfibia built its Slack chatbot for visual content

WHAT IT IS. A chatbot living in Slack that generates routine visual content on request: workshop flyers, education and membership communications. About five designated users, one per area, submit requests in a shared channel, and every output goes to the art director for approval. ChatGPT wired into Slack, with n8n underneath.

STEP 1: Run a needs assessment first. Anfibia interviewed each area's leaders about what they needed. The problem that surfaced: every branch of a multi-arm organisation needs social visuals, and all of it landed on an art department of two to three people.

STEP 2: Write a brief, then hire competitively. The team wrote a specification, posted a hiring call for developers on social media, interviewed three software engineers and selected one it had worked with before.

STEP 3: Build iteratively, then have the developer train the users. The build took roughly two to three months with feedback from the newsroom, followed by user testing and two short developer-led training sessions for area leaders.

STEP 4: Pilot on the lowest-risk content only. The chatbot handles the easiest requests and is deliberately not used for imagery attached to journalistic articles. Anfibia learned that boundary the hard way. An AI-generated gallery once labelled the Falklands/Malvinas as British on a map, published without supervision, and the audience's reaction was furious.

STEP 5: Keep the approval loop mandatory. Every output travels with the requester's brief to the art director, who must approve before anything is used. Faulty outputs are simply discarded.

WHAT THEY WOULD DO DIFFERENTLY: involve the real end users from day one. The project was designed from the art department's perspective, but the daily users turned out to be the social media team, who joined mid-project. The feedback that mattered most arrived late.

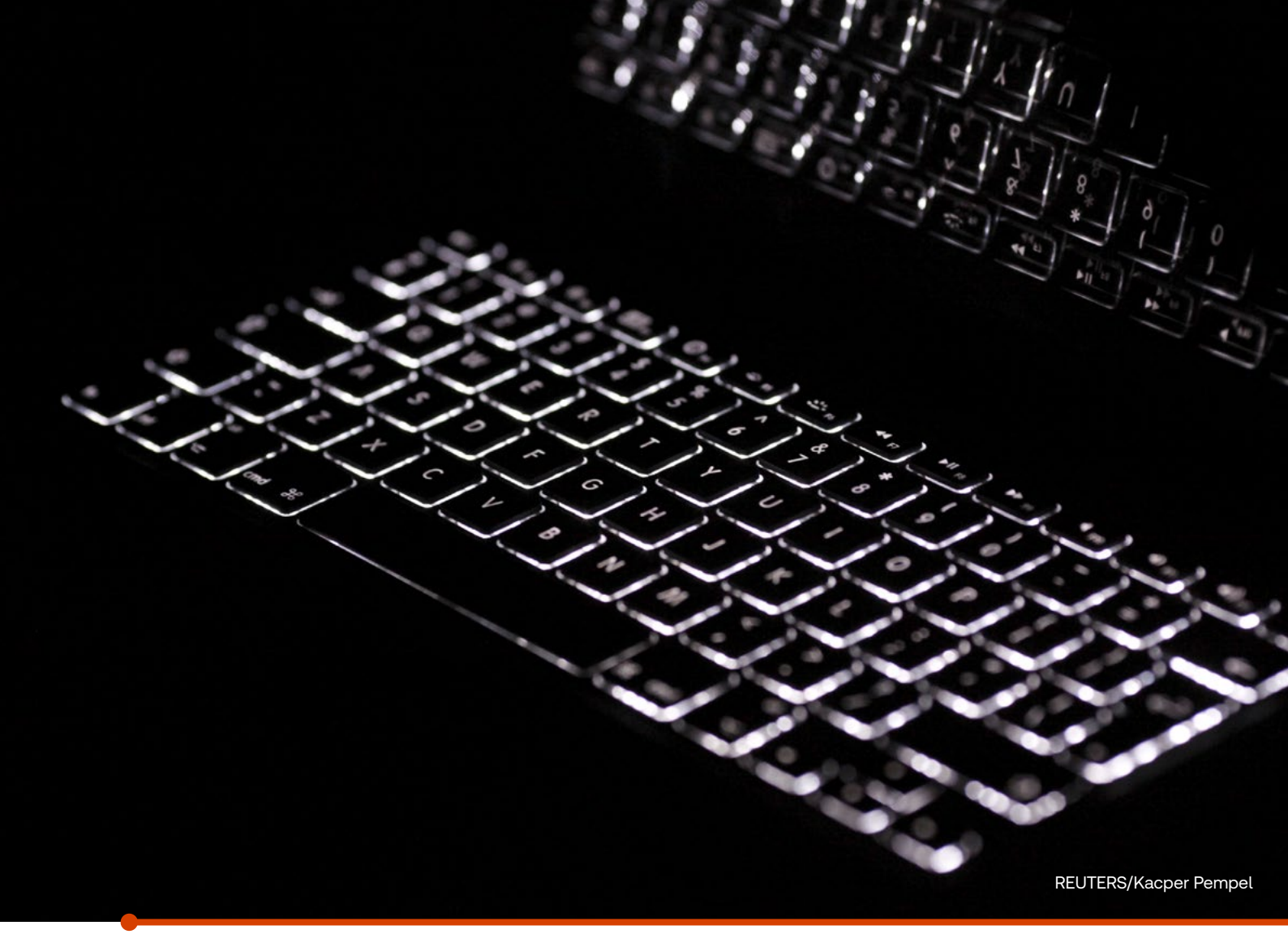
TIME AND MONEY. The external developer's fee for the two-to-three-month build, an n8n licence, two or three ChatGPT seats, plus existing design-software subscriptions.

WHAT TO EXPECT. An estimated five to seven pieces of content a week that would otherwise have gone to the art department, with the tool still in testing. Not everything generated gets used.

CHECK BEFORE YOU COPY.

Recruitment is the real variable. Anfibia found its developer through a tech reporter's personal network. Make sure to budget for hiring and training. Check you have enough routine visual demand to justify the build; Anfibia's many branches generate unusually heavy flyer traffic.

REUTERS/Mike Hutchings



REUTERS/Kacper Pempel

Part IV. Key takeaways

The pattern across the six newsrooms is consistent. AI earned its place on tasks that were repetitive, structured and already well understood, including:

- Turning a weather bulletin into an article
- Drafting a meta title
- Cutting a long recording into clips
- Tracing where a photo first appeared
- Reading comments no human could

Each solved a problem the team had named itself, and every tool had a human checking the output before it reached their respective audiences.

On build versus buy, the cohort mostly did neither. Building something new was judged too technical and too expensive at this scale, and off-the-shelf products missed local languages and formats. The working middle ground was configuring existing tools.

For example, some set up custom agents in plain language on consumer subscriptions, while others built on a free prototyping environment, deployed open-source automation, or added a button to an existing CMS.

Where a build needed skills the newsroom lacked, they hired help but learned the backend so the capability stayed in-house.

The common pitfalls were rarely technical. Prototypes over-scoped and stalled until the task was narrowed. End users joined too late. Working tools went quiet because no one owned them. Without a named owner and a routine check, even a successful build stops. And these were early builds, so the time spent training, prompting, testing and discarding was part of the cost.

The governance essentials run across every policy in this casebook. This includes:

- Human review before anything is published
- Red lines on what AI must not touch
- Disclosure to the audience
- Sensitive data kept out of external tools

For a small or mid-size newsroom, the starting point the cohort suggests is modest. Pick one specific task nobody wants, train first, configure rather than code, keep a human on every output, and name the person who will keep the tool alive.

REUTERS/Thomas Mukoya

Part V. Glossary

Agent / custom agent — A tailored AI assistant configured for a specific task, typically built on top of a general-purpose model such as ChatGPT or Claude by feeding it instructions and reference documents, rather than by writing code.

API (Application Programming Interface) — A way for one piece of software to send requests to another, commonly used to pass data to an AI model for processing.

ChatGPT — A consumer AI product from OpenAI, often used as the base model for a custom agent built through plain-language instructions rather than code.

Claude — An AI model from Anthropic, sometimes used as an alternative or complement to ChatGPT.

Claude Code — A coding-focused AI tool used to support the technical side of an AI build.

CMS (Content Management System) — The publishing software a newsroom uses to write, edit and publish articles.

Data minimisation — A privacy principle of collecting only the data a task strictly requires, often paired with role-based access controls limiting who can view, edit or export it.

Disclosure — A statement to the audience that AI was used in producing a piece of content. Newsrooms vary on whether disclosure happens every time or selectively, generically or by naming the tool, and on air, in text, or via caption or watermark.

False positive / false negative — In moderation, a false positive is benign content wrongly flagged as harmful, such as a comment saying a show “was a bomb” (meaning excellent) being flagged as terrorism; a false negative is genuinely harmful content that passes unflagged, such as abusive slang terms a generic tool doesn’t recognise.

Google AI Studio — A free environment for building and testing AI prototypes without first standing up cloud infrastructure.

Google Discover / meta title — Google Discover is a feed that surfaces articles to users based on search-engine signals, including an article’s meta title, a short title distinct from the on-page headline that search engines display.

Guardrails — Rules, restrictions or technical controls built into an AI workflow to limit risk, such as barring AI from generating editorial imagery or requiring human sign-off before publication.

Hallucination — When an AI model generates false or fabricated information presented as fact, a persistent problem in large language models.

House style / house format — A newsroom's established tone, structure and conventions. Tools are often trained on an outlet's own past articles so their output reads as the newsroom's own rather than generic AI text.

Human-in-the-loop — A design principle in which a human must review, approve or intervene in an AI-driven process before it takes effect, rather than the AI acting autonomously.

Keyword library — A list of terms, compiled from how a specific community actually speaks, used to flag content for moderation or fact-checking.

LLM (Large Language Model) — A type of AI model trained on large volumes of text to generate human-like language, prone to occasional hallucination.

Meta dashboard — Facebook/Instagram's own analytics and moderation interface, sometimes integrated alongside a custom moderation tool rather than replaced by it.

MVP (Minimum Viable Product) — A basic, working version of a tool built to test a concept before investing in a fuller build.

n8n — A workflow-automation tool used to connect other software together, such as linking a messaging platform to an AI model and a database. Free if self-hosted, or available as a paid licence.

NapoleonCat — A third-party social media management tool.

NotebookLM — A Google AI tool used for research, summarisation and content generation, including converting written text into narrated video or audio.

OpenAI Platform — The API service (distinct from the consumer ChatGPT app) used to send data to an AI model for processing.

Opus Clip — An off-the-shelf AI tool that cuts long video or audio recordings into short clips suitable for social platforms like TikTok and Instagram.

Package — A broadcast journalism term for a fully produced, narrated report combining voice, sound and interviews.

PostgreSQL — An open-source database system used to store structured data.

Prompt / prompt-writing — The instructions a user gives an AI model to produce a desired output, frequently identified as the main operational challenge in running a custom agent.

Risk assessment — A formal check of the risks of a given AI use — such as data exposure or source confidentiality — required before certain tools or data types are approved.

Sentiment register — A category of emotional tone used to classify audience comments.

Shadow AI — The informal, undisclosed use of AI tools by individual staff members outside of any newsroom-sanctioned process.

Sheng — Urban Swahili-English slang, which global moderation tools tend to miss.

Slack — A workplace messaging platform frequently used as the human-review interface in an AI workflow, rather than building review functions inside the AI tool itself.

Trello — A task-management tool.

Vox pop — Short for vox populi; brief, informal audience or street interviews used as a story format.

Whitelist / approved-tools list — A list of AI tools formally approved for staff use, typically reviewed periodically and paired with rules on what data or tasks are off-limits to unapproved platforms.

Thomson Reuters Foundation