



Integrating Claude Tools into Zeke: Use Case Proposals

Use Case 1: Autonomous Multi-Source Research Briefs

Scenario: A user asks a broad question or needs a brief on a topic without providing specific links (e.g., “What are the latest AI trends in customer service?”). Instead of the user manually finding and pasting multiple sources, Claude (integrated into Zeke) acts as an autonomous research analyst. It will **search, ingest, and synthesize** information from across media to deliver a concise, cited brief. This aligns with Zeke’s mission to “*sift across sources you’d only reach if you had a full-time researcher*” ¹, compressing what might be 10 hours of research into minutes ².

- **How it works:** Using Claude’s tool-calling capabilities, the agent can break down the task into steps. First, Claude could leverage a search tool (Anthropic provides a built-in web search function ³) or a custom internal search to find relevant podcasts, articles, or videos on the query. Next, for each promising source, Claude invokes Zeke’s content ingestion tool (exposed via the Claude Custom Tools SDK) to retrieve transcripts or text and highlight key points. This custom tool might call Zeke’s pipeline to “*turn long podcasts, papers, videos, and posts into a verified brief with receipts*” ². Claude will decide when to call these tools based on the query and intermediate results (the SDK allows Claude to assess if a tool is needed and formulate a call ⁴). After gathering facts from multiple sources, Claude analyzes the tool results and compiles a synthesized brief ⁵, ensuring that each insight is grounded in a source (with snippet and link). The outcome is a role-aware research summary that surfaces “*what matters you’d never have time to find*” on your own ¹ – complete with the evidence to back it.
- **Tools & Integration:** Key tools to expose would include: a `web_search` tool (or an internal database query) to discover content, an `ingest_content` tool to fetch/transcribe and summarize a given URL, and perhaps an `extract_insights` tool that returns structured findings (claims, metrics, timestamps). The Anthropic Custom Tools SDK supports defining such custom functions for Claude ⁶. For example, an `ingest_content` tool could accept a URL and return a list of key points with source attributions, using Zeke’s existing ingestion engine under the hood. All these would be client-hosted tools (executing on Zeke’s servers via API calls) that Claude can invoke autonomously. By configuring allowed tools and multi-turn reasoning (e.g. permitting Claude to make several tool calls in a session), we let it iterate through search results and information gathering in a controlled way.
- **Considerations:** Implementing this requires handling potentially large data. Rather than stuffing full transcripts into Claude’s prompt, the ingestion tool can pre-process and condense content into brief snippets or embeddings. Claude can then work with these snippets within its context limit. Caching or storing summaries of popular sources is useful to avoid repeated heavy processing. We must also ensure **citation fidelity** – the tool should return not just text but the reference (source URL/timestamp) so Claude can present “*every claim timestamped or quoted*” as Zeke promises ⁷. Another

consideration is trust and relevance filtering: Claude's reasoning plus Zeke's heat/novelty scoring can help prioritize which sources or claims to include, so the brief isn't cluttered with low-value info. Overall, this use case would greatly enhance user experience: the user receives a comprehensive, proof-backed answer to their research question with a single query, essentially having Claude perform the multi-source research process end-to-end on their behalf.

Use Case 2: Automated Playbook and Content Generation from Insights

Scenario: A user or team member needs to go from insights to execution rapidly – for example, generating a product playbook, marketing campaign outline, or technical experiment plan based on the latest research findings. Currently, Zeke's platform supports “*Apply*” and “*Create*” phases (e.g. one-click generation of PRDs, plans, or content using loaded goals and SOPs) ⁸ ⁹. By integrating Claude with custom tools, this process can become even more autonomous and interactive. Claude can take a high-level goal from the user and then **gather data, draft a tailored plan, and produce ready-to-use content** with minimal human guidance.

- **How it works:** Claude's reasoning ability means it can handle structured task workflows. Suppose a user says, “*We're launching Feature X – draft an experiment plan using our growth SOP and any relevant case studies*”. Claude could respond by calling a `fetch_SOP` tool to retrieve the team's internal goals or SOP document relevant to growth (ensuring the solution aligns with company policy). It could also call an internal knowledge base search (or even the multi-source research from Use Case 1) to find recent case studies or benchmarks for Feature X. Once Claude has the necessary context – the internal guidelines and external insights – it uses this information to compose the playbook. The Custom Tools SDK might include a template-filling tool like `generate_playbook` that accepts inputs (objectives, key insights, constraints) and returns a structured draft (e.g. an outline with sections for strategy, timeline, metrics). In practice, Claude itself could draft the content, but having a tool enforce structure (such as ensuring an *experiment plan* contains sections A/B/C) can improve consistency. Claude would call these tools in sequence, then integrate the results: for instance, embedding cited facts from the case studies into the plan and adhering to the retrieved SOP rules. The final output is a polished playbook or content piece that the user can immediately utilize, created in one continuous Claude session. This fulfills Zeke's promise to “*turn insights into ready-to-use plans, prompts, briefs, and content*” ¹⁰ without the user navigating multiple steps.

- **Tools & Integration:** To enable this use case, we'd expose tools like `get_user_SOP(goal_type)` to let Claude pull in user-specific context (e.g. the user's loaded Growth SOP template), and an `internal_search` tool to query any company knowledge bases or previous Zeke briefs for related material (for example, find if the team has a playbook for similar launches). Additionally, a content creation helper such as `create_document(template, data)` could ensure the output follows a specific format or house style (this might interface with Zeke's “*no AI slop*” guardrails, such as enforcing source attributions and tone ¹¹). Because the Claude SDK allows in-process function calls, Claude can interact with these internal APIs securely – e.g. fetching data from Notion or a database – and incorporate the results into its reasoning ⁶. Throughout, Claude's chain-of-thought would decide which tools to use and when: it might first pull the SOP, then realize it needs external evidence and perform a search or use a stored brief, and finally assemble the document. Notably, Claude can use the **code execution tool** if needed (Anthropic supports a Python execution tool ¹²),

which could be useful for any data crunching step (imagine computing a quick statistic or parsing a dataset as part of the plan).

- **Considerations:** **Data integration and permissions** are key here – Claude should only access a user's or team's SOPs and playbook library within their workspace. This means implementing proper authentication in the tool calls and possibly scopes for Claude's access. The content generation itself must respect the user's voice and formatting; we can leverage Claude's *output style* configurations or fine-tuned prompts to maintain on-brand tone ¹¹. Another consideration is how interactive to make this: Claude could draft the playbook then allow the user to refine it via chat (with Claude able to call an editing tool to apply changes). In terms of feasibility, this use case builds directly on Zeke's Phase 2/3 roadmap – “one-click Apply” to get plans, and turning findings into cited outputs ¹³ – so the underlying capability (plan templates, content generation) exists. Claude's tool-use simply supercharges it by automating the decision and assembly process. This would **enhance team efficiency**: internal members could delegate routine plan-writing or content drafting to the AI agent, trusting that it will pull from the latest research and adhere to company standards. By reducing the manual work from insight gathering to deliverable creation, the team can iterate faster on strategy and focus on execution.

Use Case 3: Automated Trend Monitoring and Intelligence Briefings

Scenario: The internal team (or a power user) wants to stay on top of fast-moving trends, competitor updates, or research breakthroughs relevant to their domain. Rather than manually checking dozens of sources or waiting for weekly meetings, Claude can act as an **autonomous scout**, continuously performing structured monitoring tasks and delivering concise intelligence reports. For example, the Zeke team could deploy Claude to produce a “*weekly AI market brief*” for the product and sales teams, or a user could ask, “*What new developments this month might impact our KPI?*” This addresses the common pain point of information overload – “*Everyone’s posting ‘breakthroughs’. Which ones move my KPI?*” ¹⁴. Claude would ensure the signal is separated from the noise, surfacing just the insights that matter (the “*two minutes that matter*” in a 10-hour deluge ¹⁴).

- **How it works:** Using Claude's ability to autonomously execute workflows, we can set up a scheduled or on-demand agent that calls various tools to gather and filter intel. For trend monitoring, Claude might use a combination of the `web_search` or a custom `news_feed` tool to scan a predefined list of sources: e.g. top industry blogs, competitor websites, research hubs, or even Twitter/Reddit (via API). The Custom Tools SDK would allow integration with external APIs or RSS feeds so Claude can fetch the latest content regularly. Once new items are found, Claude invokes Zeke's analysis tools on each: transcribing key sections of a competitor's 2-hour product launch webinar, summarizing a new research paper's findings, and extracting any metrics or claims. Crucially, Claude can compare these against the team's context — for instance, referencing the “*Trend & Research Graph*” (if Zeke has one built) or known benchmarks. If a tool `query_trends_db` is available (linking to an internal database of trends Zeke has been tracking), Claude could pull quantitative signals (e.g. “papers on topic X are up 180% QoQ” as mentioned in Zeke's vision ¹⁵). After gathering all this, Claude produces a structured brief for the team: e.g. **Weekly Briefing** with sections for Market Moves, Notable Research, Competitor Highlights, each bullet paired with a citation or timestamp. This directly realizes the “*weekly role-based briefings for execs, product, sales, research*” that Zeke aspired to offer ¹⁶, now delivered autonomously by the Claude agent.

- **Tools & Integration:** For this use case, we'd utilize a mix of **server-based tools** (like Anthropic's own `web_fetch` or `web_search` to retrieve online data ¹⁷) and **custom data tools**. Custom tools might include `get_latest_content(source_list)` which encapsulates pulling new items from specified sources or APIs, and `analyze_content` (similar to the ingestion tool from Use Case 1, but possibly optimized for brevity if we only need a one-paragraph summary per item). Additionally, an internal `compare_insights` tool could help Claude cross-reference new information with existing knowledge – for example, flagging if a competitor's announced feature overlaps with our product, or if a new paper's results contradict last week's findings. While Claude could do comparisons via pure prompt reasoning, a tool that queries a knowledge graph or database of past insights could provide a factual basis for those comparisons. All these tools would be orchestrated by Claude's reasoning: it might loop through sources, accumulate findings, then call a `summarize_brief` tool or simply draft the report itself. Because the SDK supports **streaming and iterative prompts**, Claude can handle dozens of pieces of content in a single session by processing them one by one and keeping summary notes in memory (thanks to its large context window).
- **Considerations:** One challenge is deciding **how often and how broadly to fetch data**. Implementation could involve a scheduled trigger (outside Claude) that periodically asks Claude to run the monitoring routine. We must ensure the agent remains focused on relevant content – this can be achieved by curating the source list or using filters (for instance, instructing the `news_feed` tool to only pull articles tagged with certain keywords). Data integration with Zeke's systems (like a trend database or user profile) would allow personalization: the agent should know what topics or competitors matter to the user/team. Another consideration is **accuracy and verification**: since this agent might operate with less direct oversight, it's critical to use Zeke's citation and flagging mechanisms (e.g. if something is unverified, Claude should note uncertainty rather than confidently assert it ¹⁸). The good news is that Claude's tool use inherently promotes verification – by design, it "interacts with external services or APIs" to get real data ⁶ instead of guessing, and it will include those references in the output. Finally, delivering the briefing could integrate with existing workflows: for example, posting the brief to a Slack channel or email via an outgoing webhook tool. This use case, if implemented, means **team members spend zero time** gathering intel yet stay informed. It turns the avalanche of content into a manageable, tailored trickle of insights – fulfilling Zeke's value of connecting *research to outcomes* quickly and transparently.

Each of the above use cases demonstrates how Anthropic's Claude, enhanced with custom tool integrations, could **perform complex research and work tasks autonomously** in the Zeke platform. By giving Claude access to Zeke's core functions and external data sources, we combine Claude's reasoning and language strengths with Zeke's rich data processing. This synergy can greatly enhance user experience and team efficiency: users get deeper insights and finished products with less effort, and internal teams can automate labor-intensive research workflows. Importantly, these ideas build on Zeke's existing workflows (ingest → analyze → apply → create ¹⁹), making them technically feasible. The Claude Custom Tools SDK provides the mechanism to implement this safely – Claude will invoke only the tools we allow, in a controlled manner, and we can monitor its tool use turns ⁴ to ensure reliability. With thoughtful implementation (ensuring data is accessible, up-to-date, and secure), integrating Claude in this way could turn Zeke into an even more powerful "easy button" for research and action, delivering on the promise of effortless, AI-augmented problem solving.

[1](#) [2](#) [7](#) [8](#) [9](#) [10](#) [11](#) [13](#) [14](#) [15](#) [16](#) [18](#) [19](#) **Exec Overview**

<https://www.notion.so/26b6eef21d7a80ccb697f13b65ccf34f>

[3](#) [4](#) [5](#) [12](#) [17](#) **Tool use with Claude - Anthropic**

<https://docs.anthropic.com/en/docs/agents-and-tools/tool-use/overview>

[6](#) **Custom Tools - Anthropic**

<https://docs.anthropic.com/en/docs/clause-code/sdk/custom-tools>