Zeno

OCTOBER 2026

Understanding agentic AI

How it works and what it means for legal practice

Dzima Basenka

Product Manager at Zeno



What makes AI agentic?

“Agentic AI” is becoming one of the more important shifts in how AI systems work, but the term is still used very loosely. Everyone seems to use it slightly differently, so for lawyers the useful question is not whether a product calls itself agentic, but what it can actually do that earlier AI tools could not.

From answering questions to pursuing outcomes

Most AI products started in roughly the same place: you asked a question and an LLM produced an answer. That was already useful, but the interaction still depended heavily on the user knowing what to ask, breaking larger work into smaller steps, bringing the right information into each step, and deciding what should happen next.

A fixed workflow can automate more of that process, but its steps are predefined. If information is missing, the situation changes, or the workflow encounters something unexpected, it usually cannot decide how to adapt.

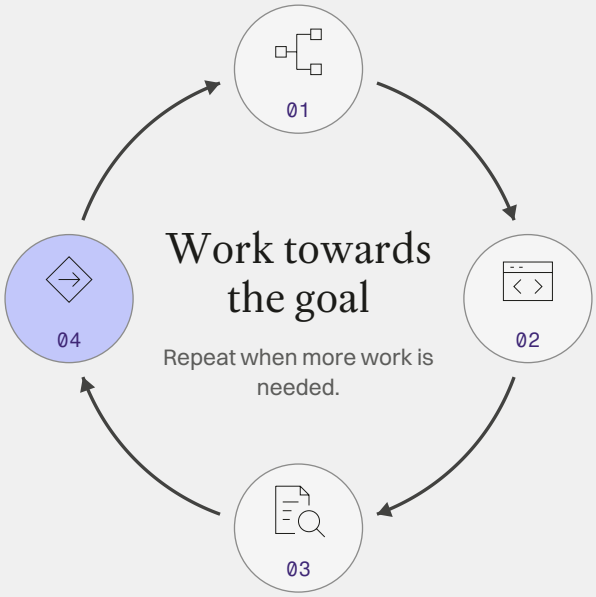
I think the simplest way to describe the next shift is that an AI system becomes agentic when it can reason about what needs to be done, orchestrate the available capabilities, and execute work towards an intended outcome.

The important difference is not simply that the answer is better. The system can interpret the intention behind a request, decide which steps are needed, choose which capabilities to use, adapt when something gets in the way, and continue until it believes the goal has been achieved. That moves AI from answering individual questions towards handling larger pieces of work.

FIGURE 01 / CONCEPTUAL MODEL

How an agent moves work forward

The next action depends on what the previous action revealed.



01 **Choose a next step**

Consider the objective, instructions and information available.

02 **Use a tool**

Search, read or edit through a capability the system can access.

03 **Examine the result**

Interpret what the tool returned and what remains unresolved.

04 **Decide what happens next**

- **Continue** if more work is needed.
- **Ask** for missing information, then resume.
- **Return** completed work for review, with any limitations.
- **Stop** and explain any blocker or limit.

Figure 1. The agentic work cycle. Each result informs the next action. The agent continues, requests missing information, returns completed work for review or stops when it cannot proceed.

What actually makes a system agentic?

I find it useful to think about four traits. They are not a formal industry definition, and different products will support them to different degrees, but together they describe the shift quite well.

1. Reasoning

Reasoning starts with understanding the request, the result the user wants and the gap between them. The agent makes and follows a plan, instead of a sequence of steps the user has to describe. If the user is planning a trip, it may check a calendar, search, compare two options, draft an itinerary and check the draft, even though the user never listed those steps.

Orchestration connects that reasoning to action by deciding which available capability to use, in which order and for what purpose. An agent can change the steps while continuing towards the same outcome. This makes it more important to help the agent understand what makes a good result for the user in that specific situation.

2. Interaction

Reasoning involves many decision points that can lead to different results. If an important detail is missing and the outcome depends on it, the agent should ask. A useful agent should be able to recognise ambiguity or an assumption and have tools to clarify it. It should be able to make its process visible and bring decisions back to the user when needed.

Responsibility for the result remains with the user. Consequential decisions must remain reviewable and controllable by the person accountable for the result. This is critical in domains such as law or finance.

3. Adaptability

“Make this short” sounds clear until two people interpret it differently. One person wants three lines, another wants a list. There is no universal good result, so a useful agent retains the user’s recurring constraints: how they write, their budget and what they refuse to compromise on. The more context it can carry forward, the less each interaction feels like starting from zero.

Adaptability means retaining individual preferences and understanding the broader work. A generic language model may know a great deal about a subject without understanding how a particular task should progress. Every domain has its own structure, concepts, definitions and stages, as well as specific ways in which a user or firm works.

The agent needs to recognise the work behind the prompt: what kind of task this is, what information is likely to matter, what may be missing and what usually needs to happen next. At the higher end, this shifts the system from merely responding towards helping guide the work itself.

4. Autonomy

Autonomy means continuing without the user watching every step. A simple version is a scheduled task. More advanced systems can execute longer workflows, react to new information or continue until a defined condition is met. Autonomy must be bounded, though. The user decides what the agent may do alone, what needs approval and where it must stop.

It can find a hotel and draft the itinerary. It should not pay or email anyone until the user says so. Autonomy creates value when it removes unnecessary supervision without removing necessary judgement.

FIGURE 02 / THE FOUR TRAITS

What actually makes a system agentic?

What to look for in the way an AI system works.



01

Reasoning

Understand the request and the intended result, then plan and adjust the steps needed to get there.

Orchestration connects the plan to available capabilities. Explain what a good result looks like.



02

Interaction

Recognise ambiguity and missing details that could change the outcome.

Ask for clarification and make the process visible. Consequential decisions remain reviewable and controllable by the person responsible.



03

Adaptability

Retain the user's preferences and constraints while understanding how the wider work should progress.

Use the task's context to identify what matters, what is missing and what should happen next.



04

Autonomy

Continue without supervision of every step, within limits set by the user.

Agree what can proceed alone, what requires approval and where the agent must stop.

Figure 2. The four traits of an agentic system. Reasoning, interaction, adaptability and autonomy work together. They describe how a system plans work, responds to context and operates within agreed limits, with the user retaining control over consequential decisions.

The agent's ecosystem: capabilities and access

The four traits describe how an agent behaves, but how useful it becomes in practice depends heavily on the ecosystem around it. To me, that ecosystem comes down to two things: capabilities and access. Even an excellent chef with only one pan and eggs will mostly end up making boiled eggs; give the same chef more tools and ingredients, and the range of possible outcomes expands dramatically.

Capabilities define what an agent can do. That may include searching the web, reading a spreadsheet, editing a file, creating a Word document, analysing data, or producing a visualisation. The more useful capabilities it can choose from, the more ways it has to act on its reasoning and complete the work.

Access defines how far the agent can reach. That includes access to data in document management systems, cloud storage, and other internal sources, but also access to other applications. An agent that can work with Outlook, Teams, Slack or other business tools can retrieve information from them, use that information in the task, and potentially act through those systems as part of the workflow. The value is not only in knowing more; it is in being able to move across the same systems and information that the user works with.

This is why two products built on similarly capable underlying models can feel completely different in practice. One may have strong capabilities but little access outside its own interface, while another can reach authoritative sources, internal data and the tools people already use every day. The model matters, but the ecosystem of capabilities and access around it often determines how useful the agent becomes.

FIGURE 03 / CAPABILITIES AND ACCESS

The ecosystem around the agent



Capabilities

What the agent can do

Search and read

Legal sources, documents and spreadsheets.

Analyse and compare

Information from the materials available.

Create and edit

Documents, data analysis and visualisations.



Access

What the agent can reach

Information

Matter files, templates and internal data.

Connected applications

Document storage and business tools.

Permissions

Which information and actions are allowed.

Put reasoning to work towards the intended outcome

Figure 3. Capabilities and access. Capabilities provide the operations an agent can perform. Access connects it to information and applications, within the permissions set for the task. Together, they shape the work it can carry out.

Example of how a legal question gets to finished work

A law firm was advising a company on a reorganisation where late-hour and weekend support would move to an external provider. The goal was to prepare legal advice in the form of a memorandum that the lawyer could review and use with the client.

Information available

The matter contained the reorganisation plan, UWV guidance, a spreadsheet with the affected employees and their individual terms, and the firm's memorandum and separation agreement templates. This is the **context** for the task. It tells Zeno what is happening, who is affected, and what the expected work product should look like.

Around that context sits Zeno's **agentic ecosystem**. Zeno can research relevant legal sources, use legal knowledge available through its knowledge graph, work with the documents and structured data in the matter, and create Word documents through Document Studio. These capabilities and sources are what make it possible for the agent to do more than reason about the problem in isolation.

Lawyer input

The lawyer asked Zeno to prepare the legal advice as a memorandum using the available matter materials and relevant legal sources.

The important part is that the lawyer defined the outcome, rather than prescribing every step needed to reach it. Zeno then had to work out what analysis was required, which information and sources it needed, and how to get from the available context to the finished memorandum.

Process

Zeno analysed the reorganisation plan and UWV guidance, researched the relevant primary legal sources, and applied them to the facts of the matter.

This is where **reasoning** comes into play. The agent decides what needs to be analysed, which capabilities to use, and how the different pieces of information fit together.

During the work, Zeno found that several facts were still missing before the advice could be completed reliably. Instead of making assumptions, it stopped and suggested the open questions the lawyer should clarify with the client. This is the **interaction** part of the agentic process. The agent continues where it can, but brings the lawyer back in when missing information could materially change the result.

Once the answers came back, Zeno continued the analysis and completed the memorandum using the firm's template. This illustrates **adaptability** to the firm and the work, and **autonomy** within the task: it continues where it can and asks when needed. Researching legal sources, using legal knowledge and producing the Word document depend on the ecosystem around it.

Result

The lawyer received the completed legal advice in the firm's memorandum format.

The same matter context then supported the next task. The spreadsheet already contained the affected employees and the relevant terms for each person, while the matter contained the separation agreement template. Zeno used that context to produce the individual agreements for all affected employees, with each document reflecting the relevant person and terms from the spreadsheet.

This is where the different pieces come together. **Context** gives the agent the facts of the matter. The **ecosystem** gives it the capabilities and access needed to work with those facts. The **agentic behaviour** lets it reason towards the goal, surface gaps when something is missing, involve the lawyer when needed, and continue the work once the missing context is available.

Where agentic workflows can go wrong

As systems gain more freedom to choose their own steps, sources to use, they also become more sensitive to the quality of the goal and context they receive.

One common way it can go wrong is a request that describes a mechanical action without explaining why the work is happening. Imagine a deed containing information about five parties. Asking an AI system to “create five different documents, one for each party” specifies the operation, but not the underlying purpose or constraint. A better request would explain that each party needs its own document and must not see the transaction details of the others. That gives the system enough context to reason about the desired result rather than simply perform a transformation.

The same issue appears when relevant context is missing. An agent may otherwise make an assumption that should have been clarified, which is why visibility and steerability matter so much in agentic workflows.

A lawyer should be able to inspect the sources being used, understand the direction of the work, notice material uncertainty and intervene before an incorrect assumption becomes embedded in the final result. Agentic systems become more useful when they can act independently within the parts of the work that are clear while making consequential uncertainty visible.

How to work with an agentic system

The way people prompt AI also needs to change slightly. With earlier systems, users often learned to specify exactly what the model should do step by step, but an agentic system benefits more from understanding the purpose of the work.

STAGE / KEY ACTION	IN PRACTICE
Before delegating **Define the goal and a good result.**	Start with the **intent** behind the task and the **goal** that needs to be achieved. Then provide the relevant **context**: the situation, parties, matter materials, jurisdiction, constraints, firm practices or anything else that affects how the work should be done. When it matters, explain what a **good result** looks like as well. The objective is not to make every prompt long, but to give the agent enough information to make good decisions while working.
While the work is happening **Resolve uncertainty and steer the work.**	Pay attention to clarification questions that affect the outcome. If the system surfaces an assumption that could change the legal conclusion, resolve it; if the direction is wrong, steer it; if the system needs information or access that should not be used, stop it. Interactive reasoning should not mean forcing the user to approve every harmless step. It should bring the human in when the human decision actually matters.
Before relying on the result **Check the conclusions, sources and edits.**	The final responsibility still sits with the lawyer, so legal conclusions need review and citations need to be checked against primary sources and matter documents. Areas of uncertainty deserve more attention than routine parts of the work. In Zeno, citations, statement confidence, and track changes support this review process in different ways. Citations make the underlying sources inspectable, statement confidence can highlight where support may need additional verification, and track changes makes edits visible instead of silently replacing the original text. The mechanisms will differ between products, but the principle is broader: as AI performs more of the work, review becomes more important rather than less.

The broader shift

Agentic AI is not simply a smarter chatbot; the larger change is a move from answer generation to goal-directed work.

The strongest systems combine reasoning, interaction, adaptability and autonomy, while their practical value depends heavily on the ecosystem around them: the capabilities they have, the data they can reach, and the external systems they can work with.

A highly capable agent with no access to the relevant matter, no authoritative legal sources and no way to produce a usable work product will still be limited. Give the same system better tools, richer context, stronger knowledge and clearer boundaries, and the range of work it can handle expands significantly.

For lawyers, I do not think the interesting future is handing judgement over to AI, but being able to delegate increasingly substantial pieces of work while keeping control over the decisions that require professional judgement. To me, that is the useful meaning of “agentic” in legal work.



About Dzima Basenka

Dzima Basenka is Product Manager at Zeno. In this report, he explains the product principles behind agentic AI and how they shape the way AI-based systems operate in legal practice.

Engineering Note 02 · October 2026 · More from Zeno