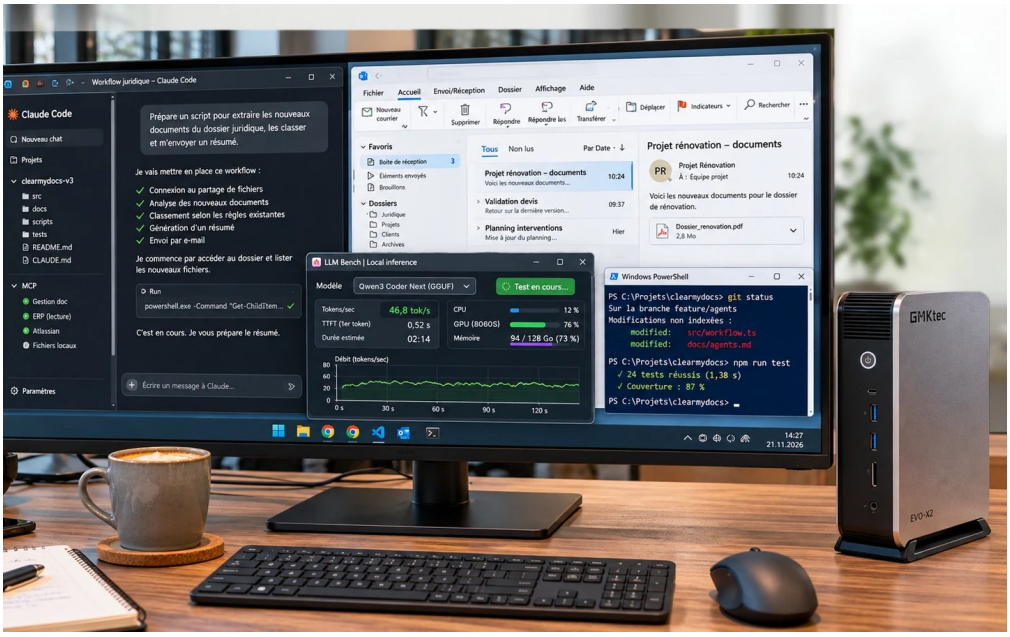
LOGICIELS ET IA

AI is starting to use our software

Connecteurs, pilotage de l'écran, outils créés à la volée : ce que changent les annonces de novembre, et pourquoi les règles métier doivent rester dans le logiciel.

Publié le 21 novembre 2026 — Imprimé le 27 novembre 2026



Published on 21 November 2026 · Karbonic

AI assistants no longer just answer questions: they open applications, work in desktop software and prepare operations. What remains to be settled is where the rules, the data and the controls live.

Within a few days, Anthropic announced several developments that, taken separately, might look like simple extra features. Together, they are beginning to change the way we work with IT.

On 10 November, Claude gained the ability to open and use applications directly in its own environment. On the 20th, the assistant's use of desktop applications was extended, in beta, to the Team and Enterprise plans.

What interests us is not so much being able to ask an AI to click a button. It is what happens when, gradually, you no longer need to know the interface of each piece of software to do your job.

Three ways of using software

Take a fairly ordinary situation: an employee has to find a job, complete the report and prepare the invoicing. Today, that may mean going through email, a document management system, a time-tracking application and an ERP.

An assistant can now step in in three ways.

The first is to use a connector. With MCP, the Model Context Protocol, an application exposes some of its functions to an assistant. The assistant does not need to know where the "Create job" button is: it calls the corresponding function directly. Anthropic is now going further with MCP Apps: applications can also display their own interface inside Claude. Adobe, Atlassian or HubSpot, among others, open in this way within the conversation. You ask for a change, see the result in the application and take back control without switching window. Changes are confirmed before they are carried out.

The second is to use the computer directly. When there is no connector, Claude can work in a piece of software through its graphical interface: seeing the screen, clicking, typing, navigating. This is particularly interesting for older applications, specialised business software or internal tools without an API. It does not turn twenty-year-old software into a modern application. Automation through the screen remains slower and more fragile than a direct integration. But it makes feasible operations that previously required specific development, sometimes hard to justify.

The third is to build the tool that is missing. With Claude Sheets, Whiteboards and Data, Anthropic makes it possible to build spreadsheets, workspaces and dashboards in the assistant's own environment. They are not equivalents of Excel or of large management systems, and that is not their purpose. But for a tracking table, a one-off analysis or a small internal application, you can now build something directly from the need expressed.

So three activities that used to be separate are starting to come together: using software, connecting it to other software and adding functions to it.

In the United States, the big ERP vendors join in

On 27 October, Oracle and OpenAI announced Oracle AI Workspace, which brings OpenAI's models and Oracle's applications together in a single interface, with plugins in ChatGPT. The idea is to run approved business processes from this shared interface, while keeping the company's governance, approvals and controls. It is still a product direction, with no announced availability date.

The signal is nonetheless interesting. For years, ERP vendors have developed their own interfaces, dashboards, portals and mobile apps. Tomorrow, part of the work could go through an interface that no longer belongs to them.

And it is not only Oracle. On 12 November, Anthropic presented Claude for Manufacturing: 17 skills and four connectors aimed at industry. One of the examples consists of reading a customer request, comparing the costs of similar operations and preparing the quote. The quote remains a draft, which a person approves before it is sent.

This is much more concrete than a demonstration in which an AI writes a text. It touches the real operations of the company.

In China, the assistant already does the shopping

The Chinese approach is interesting to compare. On 15 January 2026, Alibaba opened public testing, in China, of an evolution of its Qwen assistant that makes it possible to order a meal, organise a trip or book a service without leaving the conversation. It relies on its own ecosystem: Taobao for orders, Fliggy for travel, Alipay for payment.

The user says what they want to achieve, the assistant uses the services needed and asks for explicit confirmation before paying.

The difference from the American announcements is revealing. At Oracle, the assistant works with the company's management applications. At Alibaba, it is placed between the consumer and everyday services. Two very different markets, one and the same evolution: the assistant becomes the intermediary between the user and the applications.

Carrying out an operation, or carrying it out properly

This is where our experience of software development makes us more attentive.

Let us return to our job to be invoiced. An assistant can find the client, identify the work done, suggest a duration and prepare the entry. That is already a worthwhile gain. But you still need to know whether the rate is right, whether the hours are billable, which VAT regime applies and whether the job can be closed.

These rules are not interface details: they are part of the company's business. You can ask the assistant to apply them. You can also leave them in the ERP and allow the assistant to call functions that respect them. The two approaches do not offer the same guarantees.

There is also a less spectacular point in the demonstrations: verification. If the assistant prepares fifty entries that have to be checked line by line, has any time really been saved? Not necessarily. If, on the other hand, it gathers the documents, prepares the entries and

flags the three cases that fall outside the usual rules, the work changes in nature.

This is, incidentally, an interesting aspect of Claude Data: data teams can certify certain queries and definitions used in dashboards. That avoids two analyses giving different results because the AI did not use the same definition of revenue or margin.

The point is therefore not only to have more operations carried out, but to make their results reliable enough to be used.

A new layer in company IT

Anthropic also offers, on request, to host its agents in the company's own infrastructure, on site or in its cloud. The service then runs at the company, with the database that stores the agents, sessions and their history. The commands and files handled by the agent run in isolated environments that the company controls.

Mind the distinction: hosting the agent does not mean running the model locally. The server still sends the conversation to an inference provider: Anthropic by default, but also Amazon, Google, Microsoft or a gateway of the company's own.

Components that until now were bundled in a single service are nonetheless starting to be separated. The company keeps its software, its data and the history of what its agents do. The model provides understanding and reasoning, and becomes one supplier among others.

The documentation is, incidentally, clear about the trade-off: an agent's shell is a real shell, with the rights and network access of the environment in which it runs, and isolation is the responsibility of whoever hosts it. Should an agent allowed to view an invoice also be able to modify, forward or cancel it? These are classic IT security questions, in a new environment.

This architecture could make it possible to change model without rebuilding every integration, even though the openness of MCP does not by itself guarantee independence from a provider. The portability of data, tools and rules also has to be planned.

What we do with it in practice

We are developing an ERP for SMEs, and we had to settle these questions very early on.

The application has two entry points for AI. The first is a built-in assistant, connected to the API of the model the company chooses, which acts in the context of the logged-in user and never has more rights than that user. The second is an MCP server, which lets an outside assistant such as Claude drive the application. Both go through the same rights filter: there is only one authorisation gate in the code. That has, incidentally, become our rule: each of our software products, from document management to email sorting and the page editor, is developed from the outset with its own MCP server.

An action requested by the AI follows the same route as a click on the corresponding button, with the same validations, and never touches the database directly. Business rules therefore apply without exception: a quote without a client cannot move to the "sent" status, even if the assistant asks for it. Our first trials showed that this refusal had to be reported back to it explicitly; otherwise it thinks it has succeeded. A deletion goes to the recycle bin, and nothing disappears without trace. For sensitive operations, such as posting a batch of invoices, we add a step in which the assistant first presents what it is going to do and waits for explicit confirmation.

The development itself follows the same logic. The ERP is built by a team of AI agents spread across several machines, working in small batches, each within its own scope. Each batch passes automated checks, including static analysis of the code, then an independent judge before it is integrated. Judges also play the role of an employer, a department head and an employee to check that the screens suit each of them.

Part of this work can run locally. We are testing a mini PC with 128 GB of unified memory, running Fedora with llama.cpp and Qwen3 Coder Next, for the judges and small tasks. The aim is not to do everything locally, but to find out what can be, and to be able to process sensitive data without it leaving the company.

This is also what makes the question of architecture so concrete for us: the model can change, while the rules and the data stay in the application.

What now?

For a long time, developing a feature also meant developing its interface: screens, forms, menus, controls, navigation. Some of these elements will remain

indispensable. But why impose a series of forms on someone who could simply explain what they want to do?

The ERP would keep the data, the transactions, the business rules and the controls. The assistant would understand the request, gather the information and prepare the operations.

This does not remove the need for well-designed software. It makes its architecture even more important: clearly defined functions, data accessible in a controlled way, verifiable operations.

We are not there yet. Some functions are in beta, the connectors do not cover every application and control through the screen can still be improved. But the change is already concrete enough to be worth paying attention to.

We have spent decades teaching users how software works. We are beginning to be able to explain to software what users want to do.

Sources

- The New Stack: Anthropic wants Claude to be the front end for your entire enterprise app stack
- Anthropic: Claude release notes
- Anthropic: Claude for Manufacturing
- Anthropic: self-hosted Managed Agents
- Oracle: press release of 27 October 2026
- Alibaba: Qwen App