

After the App

Designing for intent, agency and interfaces that appear when we need them

Niki Shu Designs · October 2026

For most of the history of digital products, we have designed around a simple assumption. If someone wants software to do something, they will go to the software. We open the banking app to pay and the calendar to arrange a meeting, and we have become remarkably good at learning each one. Agentic AI raises a different question. **What happens when the software starts coming to us?**

Imagine saying "I'm travelling to Manchester next Thursday. Find me a train after nine, somewhere decent to stay near the meeting, and make sure I can be home by six on Friday." Today that means a calendar, a train service, maps, a hotel site and a lot of copying between them. An agent starts somewhere else entirely. Not with an application, but with the intention.

We designed software around applications

The app has been an extraordinarily successful unit of design, giving functionality a boundary and a brand. It also asks a lot of us. We have to know which software owns a capability, open it and translate what we want into the steps it expects. We think "I need to be in Manchester", not "open calendar, open maps, apply four filters". The user has always performed the translation between **intent** and **interface**, and AI increasingly lets software do some of that translation instead.

The app is starting to come apart

There are already signs of this shift. Android's AppFunctions let apps expose their capabilities so that agents can discover and invoke them through natural language. OpenAI's apps in ChatGPT expose functionality while showing only the interface the moment needs, from an inline card to a fullscreen workspace. Google's A2UI lets an agent describe the interface it needs and have it rendered with trusted components from the host's own design system. Microsoft's Adaptive Cards bring approvals, inputs and actions into a conversation. The shared idea is simple. The

capability of an application is beginning to separate from the interface we traditionally used to reach it.

This probably doesn't mean everything becomes chat

It is tempting to imagine one enormous text box where software simply disappears. That seems unlikely. Language shines when intention is ambiguous. "Somewhere nice for dinner, not too formal, for six" is awkward to express through filters. But when the system finds eight restaurants, we don't want eight paragraphs. We want photographs, prices and cards we can compare. In other words, we want an interface.

The same is true elsewhere. Show me a chart when I want to see how revenue has changed. Give me a calendar when I am choosing dates. If an agent is about to spend £2,000, give me an explicit confirmation. The interesting future is not chat replacing UI. It is natural language becoming one way to express intent, while the system chooses a better representation for whatever follows. The interface hasn't disappeared. It has become situational.

The design system becomes a vocabulary

For decades designers have anticipated journeys and produced the screens to support them. An agent makes the interface itself contextual. Ask for restaurants and a map appears. Ask to edit something visually and the conversation opens into a canvas.

That needn't mean generating arbitrary pixels. A safer model is emerging in which designers create trusted components while the agent decides when to compose them, and the host keeps accessibility and conventions intact. Tomorrow the design system may also be a **vocabulary available to an agent**. The designer establishes the grammar, and the agent composes the sentence.

The difficult part begins when software can act

An AI that tells me which train to catch is providing information. One that books the train is exercising agency. One that notices a cancellation, rebooks, changes the hotel and moves tomorrow's meeting has gone further still. Now we are designing a relationship with a system that acts on our behalf.

That relationship sits on a spectrum. At one end the human controls almost everything and AI simply assists. Further along it suggests, then acts under supervision. At the far end it quietly watches conditions, acts within agreed boundaries and interrupts only when something needs attention. There is no single right position. We may happily let an agent tidy our photographs yet expect real friction before it moves a large sum of money. The question shifts from what the AI can do to **what it should be allowed to do here**.

The material we design with now includes initiative, authority, uncertainty, memory and attention.

This is where UX becomes more interesting rather than less relevant. New questions join the familiar ones. Should the system act without being asked? When should it interrupt or ask permission? What should it remember, and can an action be undone? These aren't simply technical questions, because they determine how the experience feels.

Trust becomes part of the interface

When I open a photo editor, I have a fair mental model of what it can do. A blank box asking "What would you like me to do?" makes that boundary almost invisible. Can it read my email or buy something while I am away? The simpler the interface looks, the more complex the capability behind it may be. Good agentic design makes invisible things visible when they matter, showing what the agent intends, what has already happened and what needs approval. That isn't a screen full of warnings. It is proportionate visibility. Recent Nielsen Norman Group research suggests that while AI is now widely used by UX professionals, they remain cautious about output quality and privacy.

Trust can't be written into a reassuring message. It has to emerge from the behaviour of the system.

Rules of engagement over literal journeys

UX has long relied on literal journeys, one screen after another. Agentic systems make them less predictable. One response may need a card while another needs no visible interface at all. The job becomes designing the **rules of engagement**. What may the system infer, what must it ask, what deserves attention and where does the human come back in? Nielsen Norman Group's State of UX report makes a related point, that as more interaction is mediated by AI, the surface matters less and judgement and the wider behaviour of the system matter more.

The capability becomes the unit

So do we still need apps? Almost certainly. The bank still needs the systems that make a payment, and complex creative work still deserves dedicated tools. What changes is how we reach them, and the home screen icon becomes only one doorway. The important unit may no longer be **the app** but **the capability**.

The designer's role becomes broader

If AI can produce a polished screen from a sentence, competent interface production does become easier. But production has never been the whole of UX. The harder questions are whether the right things are automated, whether people know when to trust it and whether they can step in. Those problems grow as software becomes more capable. The designer moves upstream, from screens to systems of interaction, from journeys to boundaries and from where the button sits to whether the button needs to exist at all.

After the app

The app isn't disappearing tomorrow, and the future probably isn't one enormous chat window. Something subtler is happening. We are increasingly deciding **when the machine should listen, show, suggest or act, and when it should hand control back**. The best interface may be the one that appears exactly when we need it and knows when to get out of the way.

Sources

- Android Developers, AppFunctions overview. <https://developer.android.com/ai/appfunctions>
- A2UI, a protocol for agent driven interfaces. <https://a2ui.org/>
- OpenAI Apps SDK, UI guidelines. <https://developers.openai.com/apps-sdk/concepts/design-guidelines>