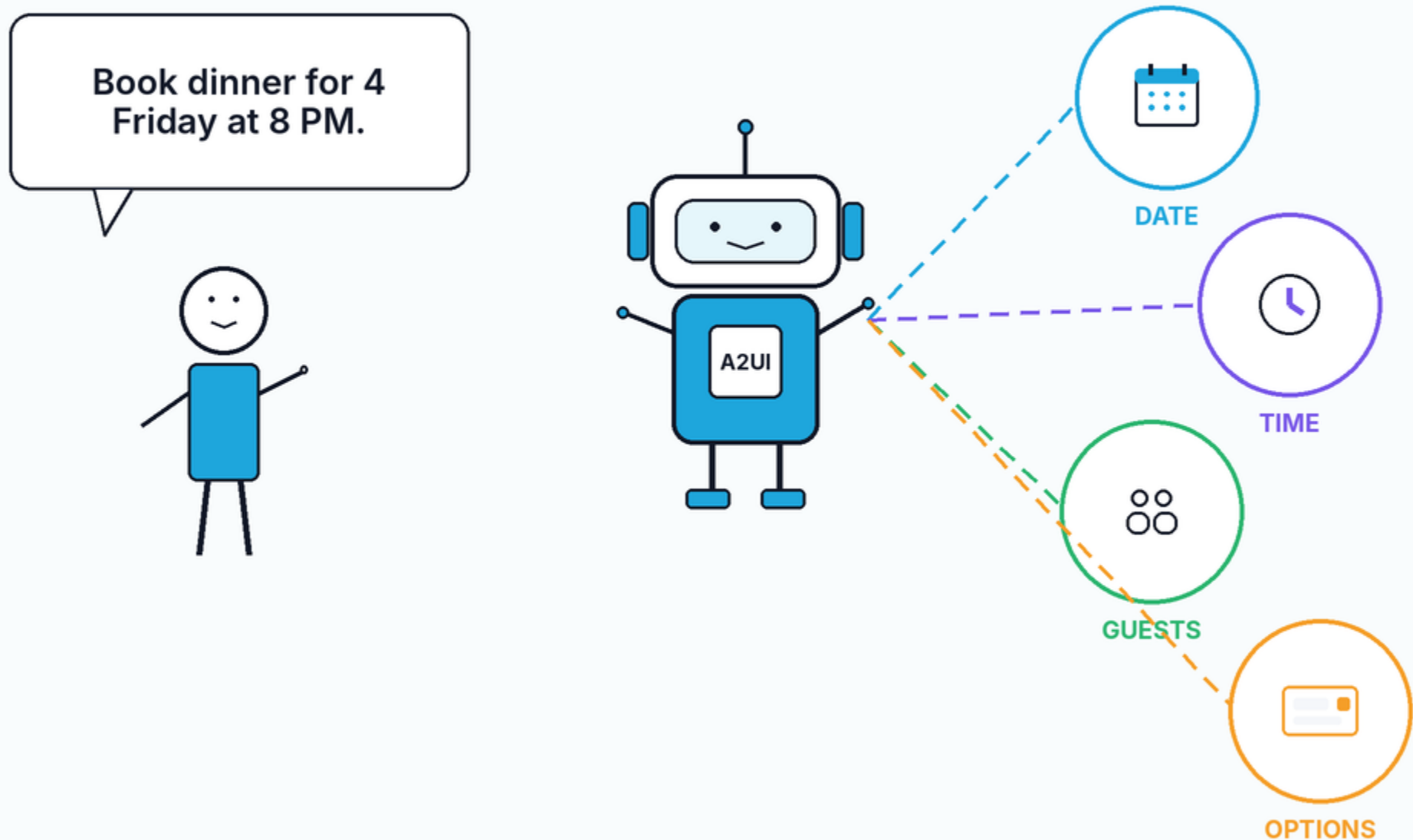


HOW DO AI AGENTS SHOW YOU THE RIGHT INTERFACE?

Meet A2UI — a shared language for agents to describe interactive UI.



MCP connects agents to tools. A2A connects agents to agents. A2UI helps agents present interfaces to people.

CHAT IS GREAT — UNTIL THE TASK NEEDS CONTROLS

Some tasks are easier to complete with fields, buttons and choices.

CHAT-ONLY

"What date?" → "What time?" →
"How many people?" → "Which
area?"



Friday. 8 PM. 4
people.

AGENT-GENERATED UI

The agent can show the useful
controls together.

FRIDAY • 8:00 PM



GUESTS: 4

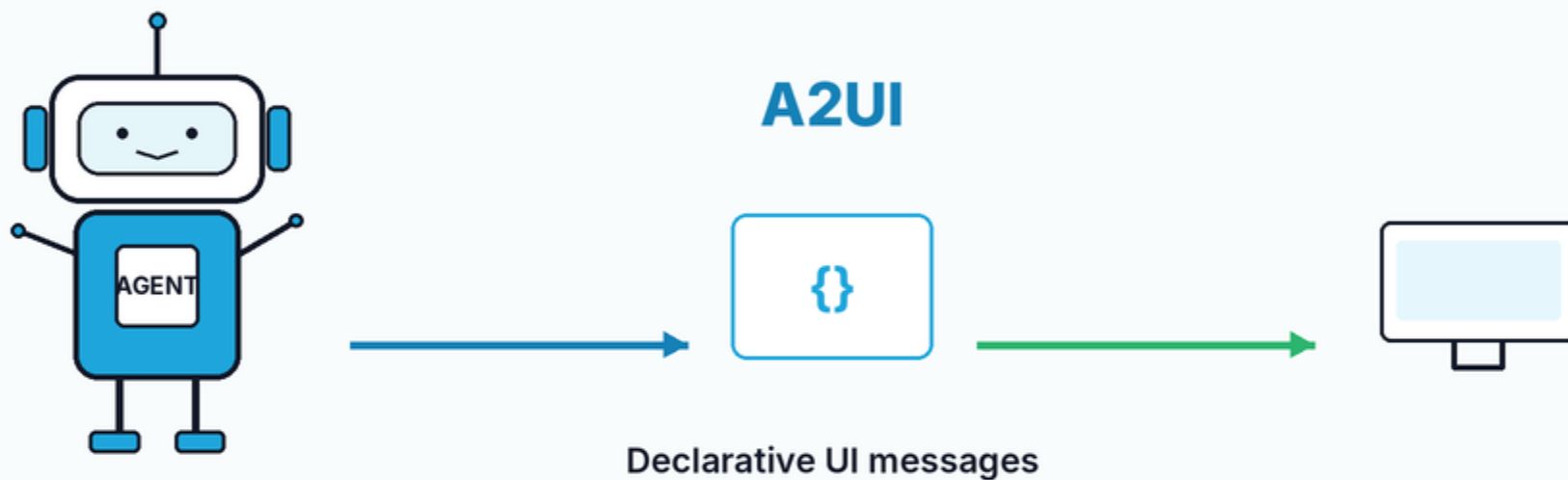


SHOW OPTIONS

The right interface can remove unnecessary back-and-forth.

A2UI = AGENT TO UI

The agent describes UI intent. The client renders it with its own trusted components.



NOT A SCREENSHOT

A description of components, data and actions.

A2UI standardizes the UI conversation — while the host app keeps control of rendering.

WHAT EXACTLY IS A2UI?

A JSON-based, streaming protocol for agent-driven interfaces.

**An agent sends declarative UI messages.
A client validates them and renders native components.**

1

CREATE

Create a UI surface

2

UPDATE UI

Add or change
components

3

UPDATE DATA

Refresh the data
model

4

REMOVE

Delete a surface

A2UI describes structure and state changes — not the agent's private reasoning.

THE AGENT DESCRIBES. THE APP RENDERS.

This separation is the core design idea.

AGENT — WHAT

Chooses the useful UI: fields, cards, actions, labels and data.



HOST APP — HOW

Maps that intent to its own design system, layout, styling and accessibility.



The agent can request a Button. The app decides what that Button looks like.

WHY NOT JUST SEND HTML OR JAVASCRIPT?

Generated executable code crosses a much larger trust boundary.

EXECUTABLE CODE

Powerful — but the client must trust code, dependencies and behavior.

A red icon representing executable code, consisting of the symbols `</>` inside a rounded square.

BIGGER ATTACK SURFACE

DECLARATIVE UI

The agent asks for approved components. The host validates before rendering.

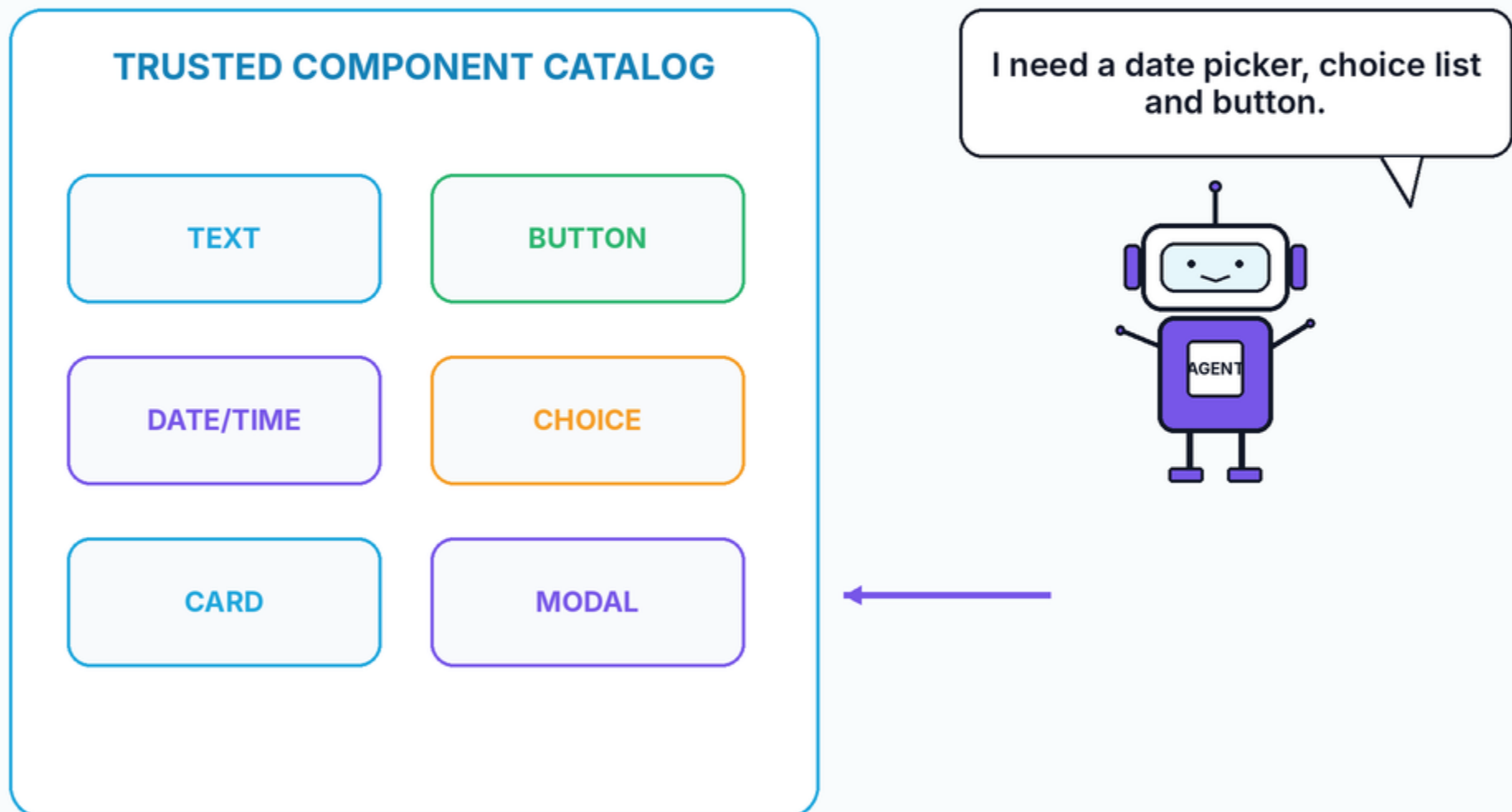
A green icon representing a declarative UI component, consisting of the symbols `{ }` inside a rounded square.

TRUSTED COMPONENTS

A2UI is declarative data, not arbitrary code.

THE COMPONENT CATALOG IS THE MENU

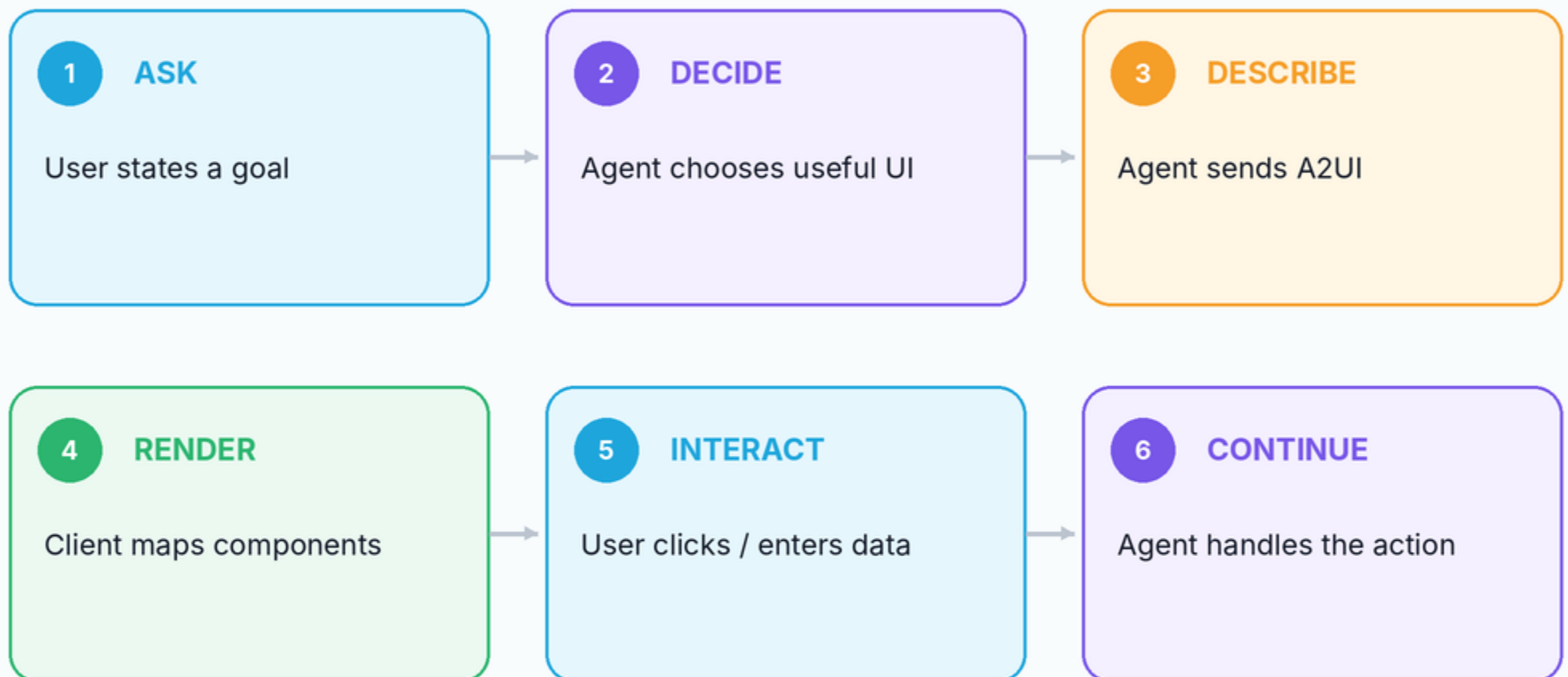
The agent can request only components the host makes available.



A production app can use its own component catalog and design system.

HOW AN A2UI INTERACTION HAPPENS

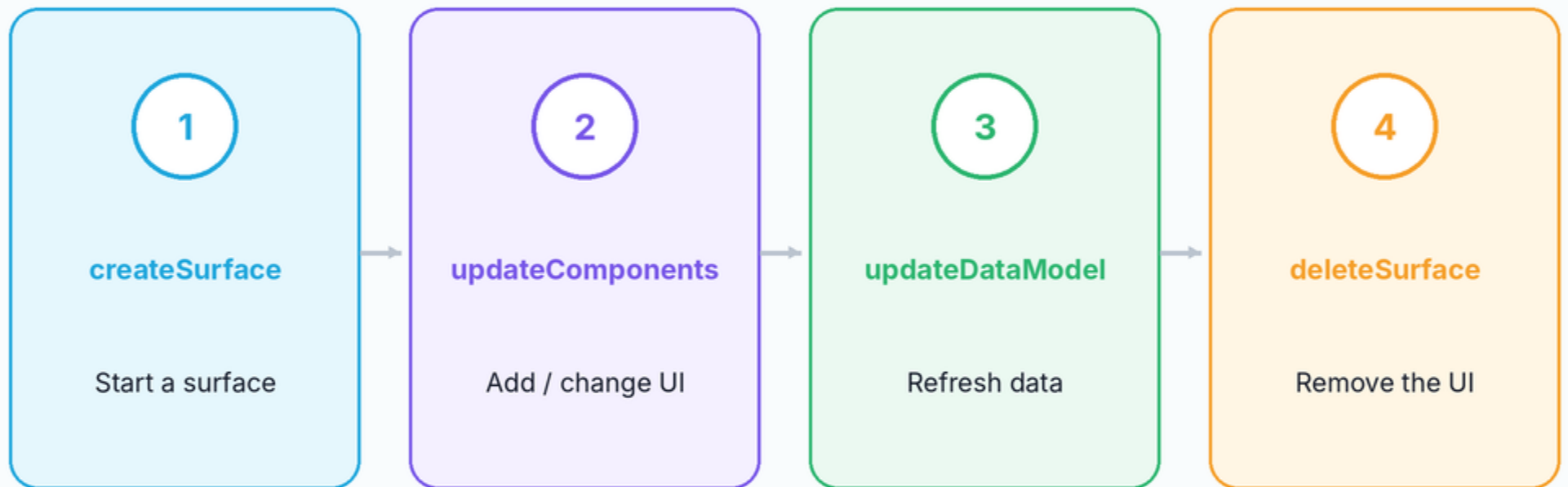
From user request to rendered UI — then back to the agent.



The interface is part of the agent loop — not just decoration.

A2UI IS A STREAM — NOT A SCREENSHOT

The UI can be created and updated message by message.



Progressive rendering = useful UI can appear before the entire task is finished.

A2UI can update a live interface as the agent learns more.

THE UI CAN CHANGE AS NEW DATA ARRIVES

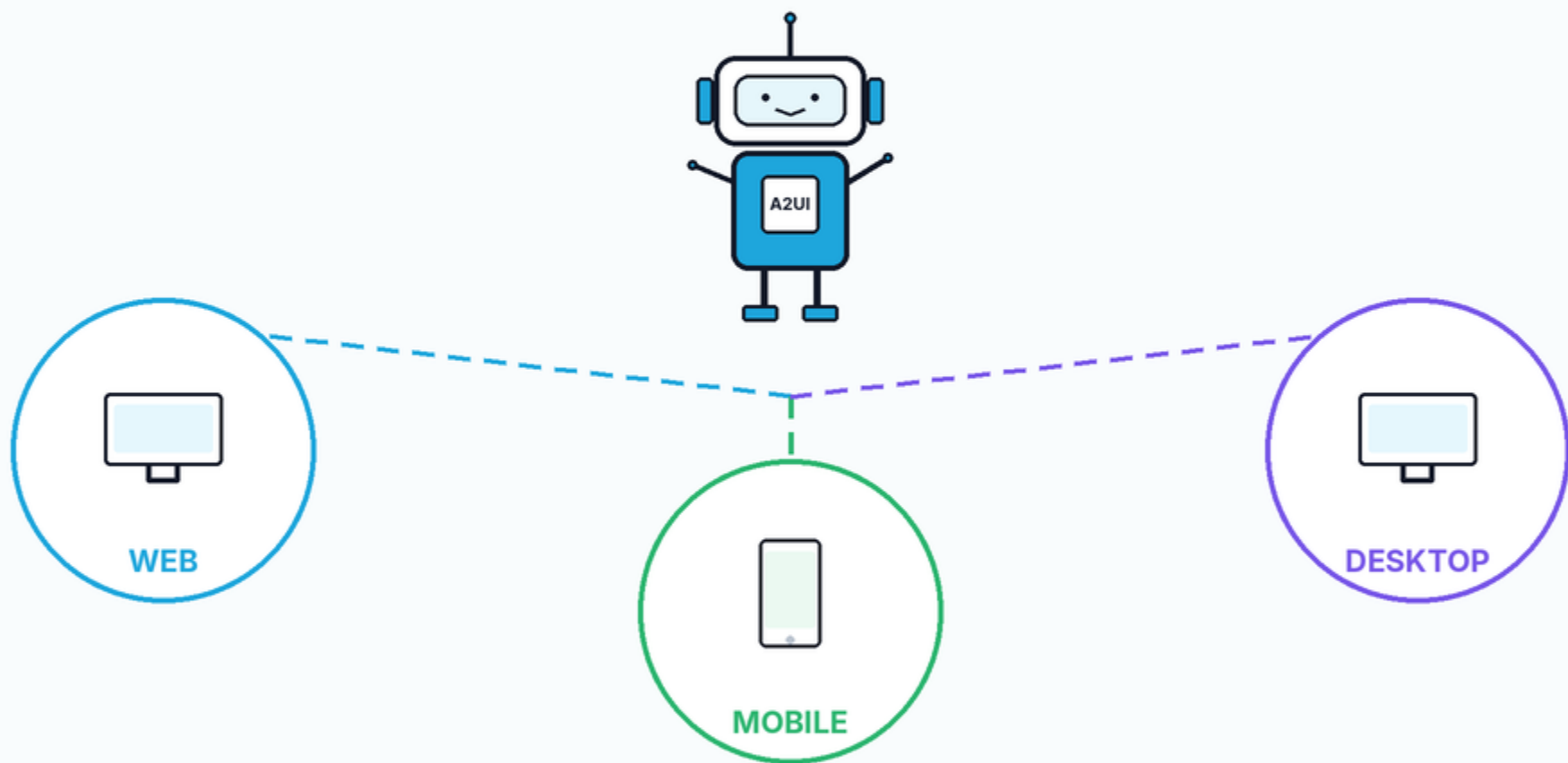
A long task does not need to regenerate the whole screen from scratch.



Streaming UI is useful when availability, progress or results change over time.

ONE UI INTENT, MANY PLATFORMS

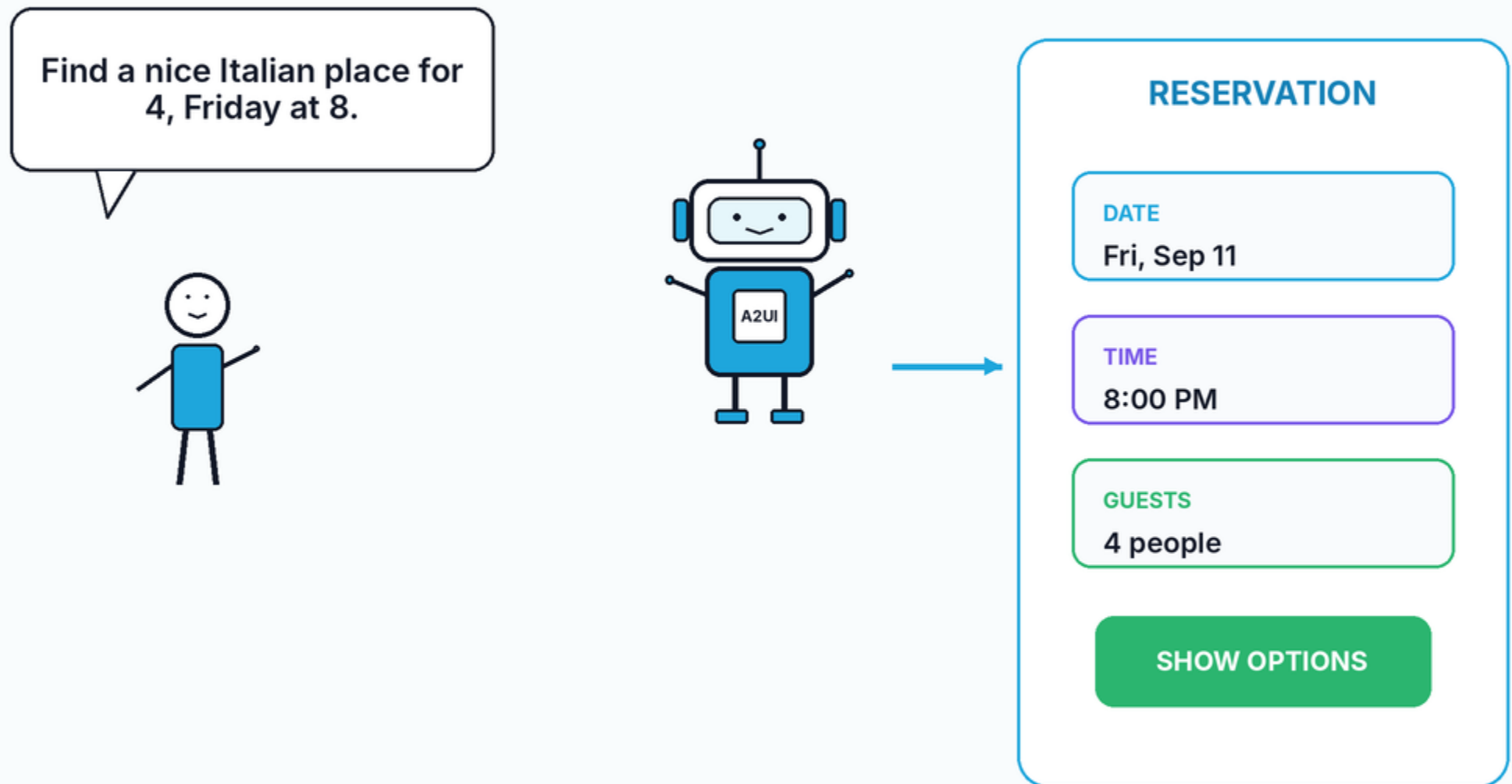
The same agent intent can be rendered using each platform's native components.



Portable intent does not mean identical pixels. Each client can render natively.

REAL EXAMPLE: BOOK A RESTAURANT

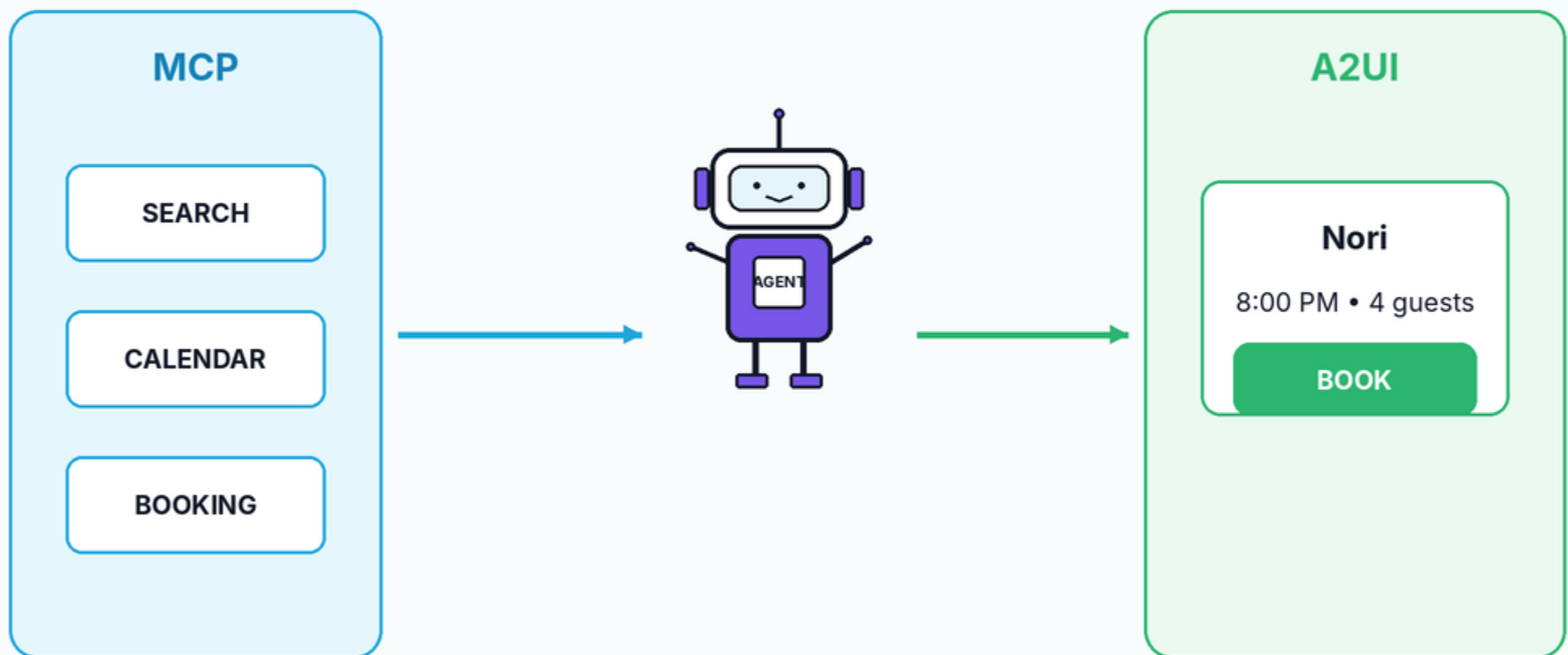
One user goal can become a small interactive workflow.



Instead of asking five follow-up questions, the agent can present the useful controls at once.

A2UI + MCP: SHOW + DO

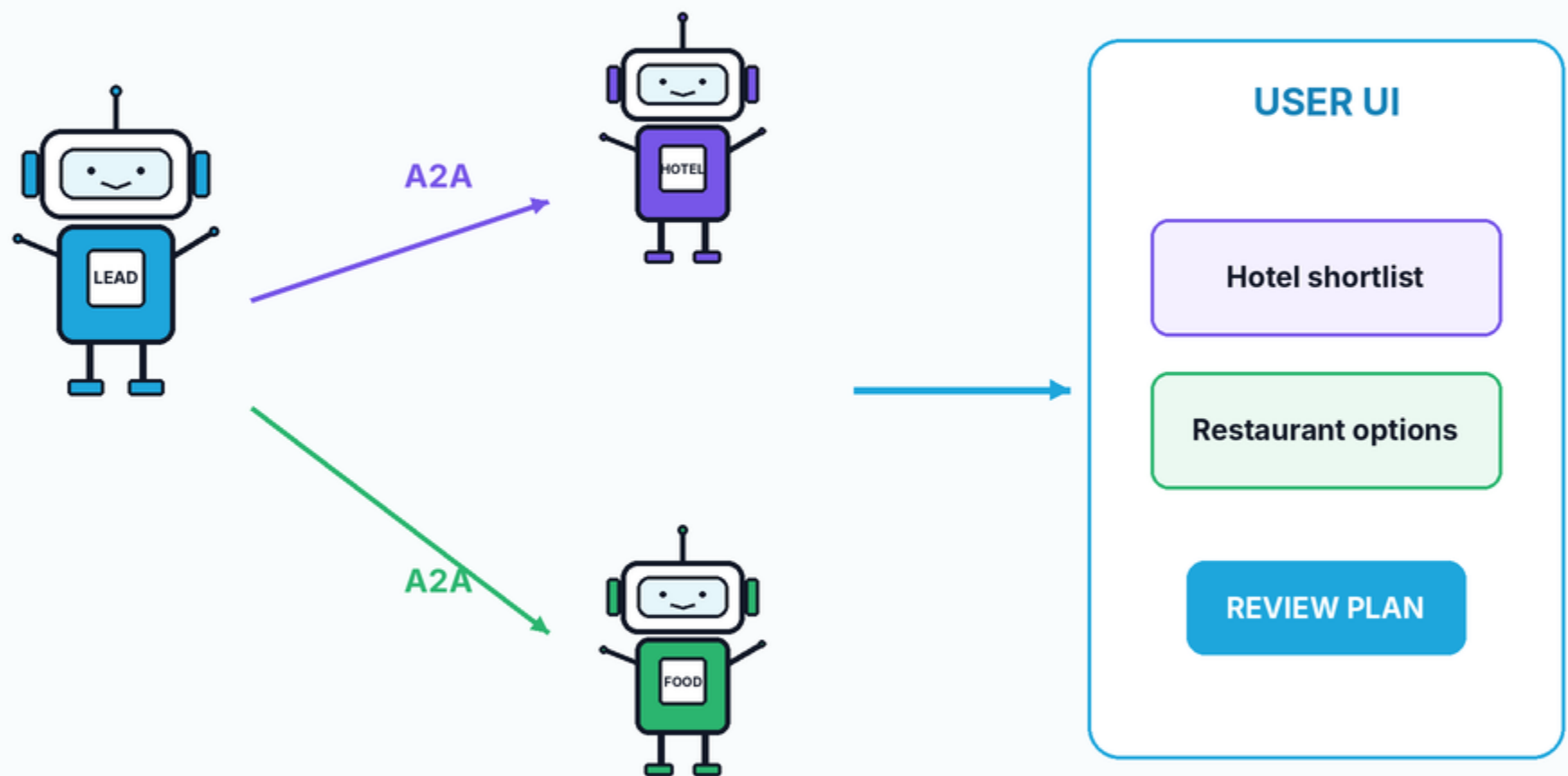
MCP connects the agent to capabilities. A2UI presents the useful interface.



MCP gives the agent access. A2UI gives the user an interface to choose, review and confirm.

A2UI + A2A: COLLABORATE + PRESENT

A user-facing agent can ask specialists for work, then surface the result as UI.



A2A can coordinate specialists. A2UI can present their combined work to the person.

A2UI VS MCP APPS: TWO UI STRATEGIES

They can coexist, but they solve different presentation needs.

A2UI

Declarative UI descriptions rendered by the host using trusted/native components.



PORTABLE • NATIVE-FEELING

MCP APPS

Richer embedded web experiences when the task needs a custom interactive application.



CUSTOM • EMBEDDED EXPERIENCE

Use the simplest UI mechanism that matches the interaction you actually need.

A STANDARD UI PROTOCOL DOES NOT CREATE AUTOMATIC TRUST

The client still decides what components, data and actions are allowed.



TRUSTED CATALOG

Only approved components



VALIDATE DATA

Reject malformed or unsafe values



ACTION APPROVAL

Confirm consequential actions



STATE CHECKS

Do not render stale choices



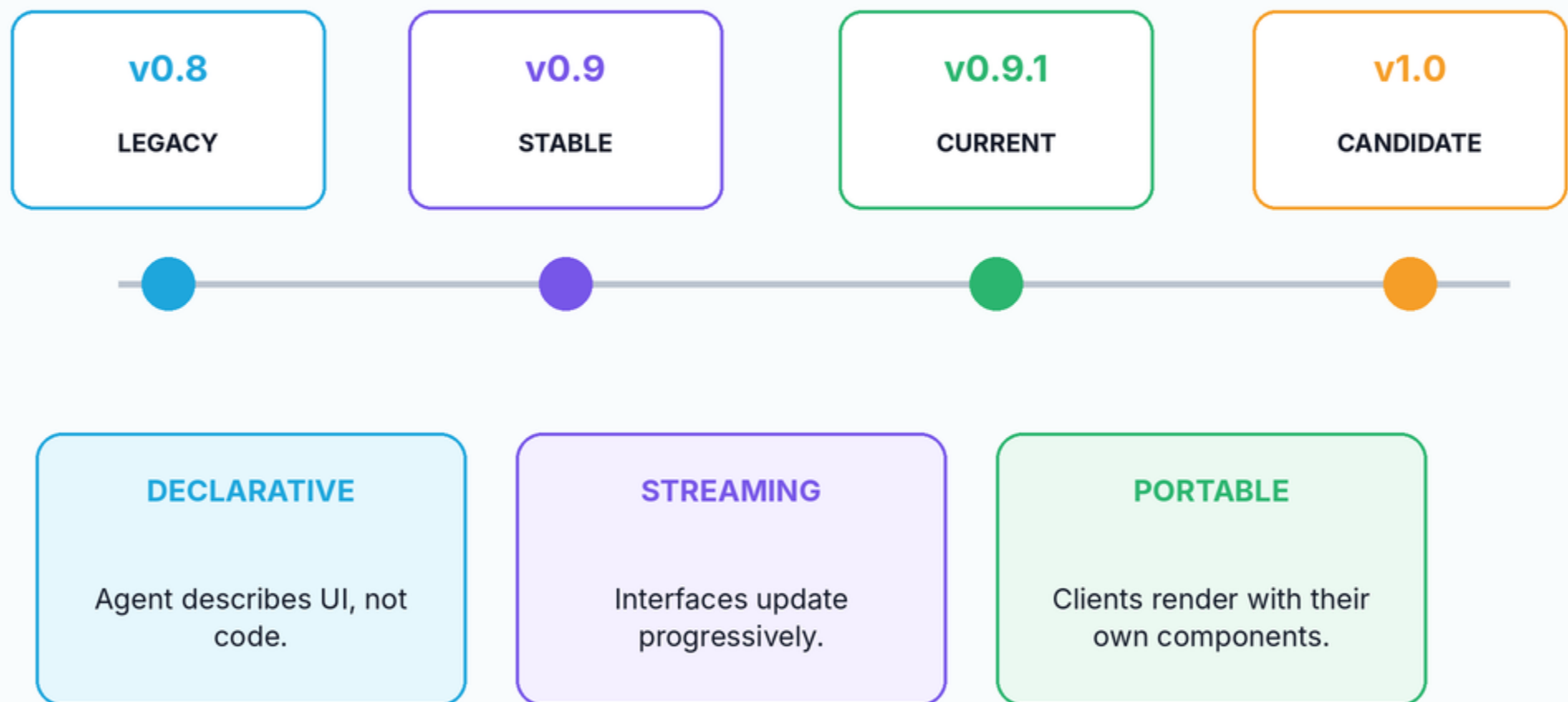
PERMISSIONS

Limit what the agent may do

Declarative UI reduces one class of risk — it does not remove product, permission or safety checks.

A2UI IN 2026: FROM EXPERIMENT TO STANDARD

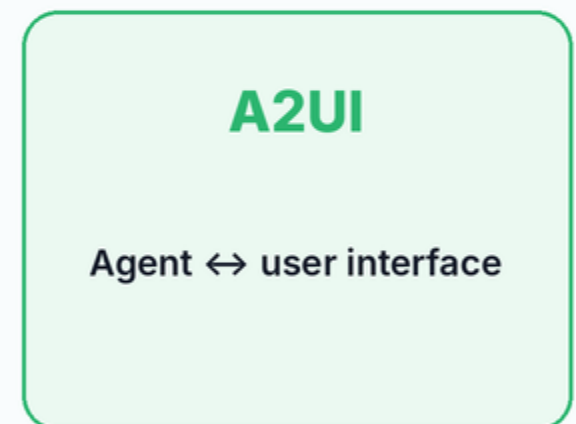
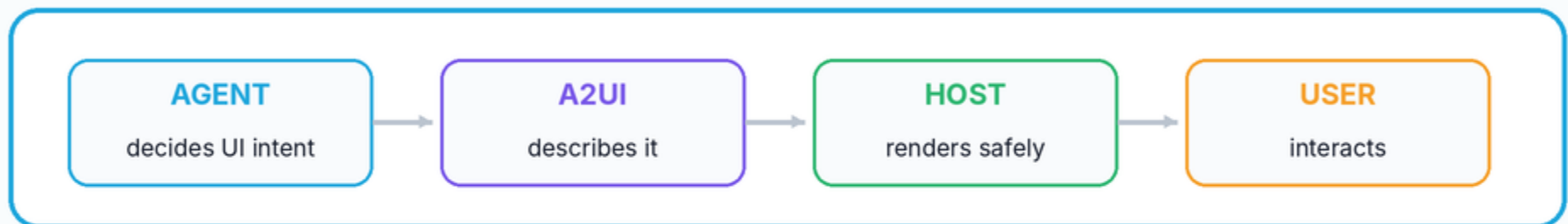
The protocol is evolving toward a stable, interoperable UI layer for agents.



A2UI is becoming part of the broader interoperability stack around agents.

DAY 18 TAKEAWAY

A2UI gives agents a shared way to communicate interactive UI intent.



AGENTS ARE MOVING FROM "CHAT WITH ME" TO "WORK WITH ME."

NEXT: DAY 19 — AG-UI: streaming agent state and events to the frontend.

What should an agent show you instead of another paragraph?