



Beyond the Chat Window

Frameworks for AI-Enabled App Discovery,
Rendering, & Composition

Bob Myers

Chief Product Officer

[interop.io](https://www.interop.io)



OSFF Toronto | April 14, 2026

Two Worlds

86%

of enterprise AI spend is
chat-based copilots
Menlo Ventures, 2025

20–30

applications on a typical
Capital Markets desktop

The AI lives in one place. The work happens in another. Users move between them, copying context back and forth. The AI has no awareness of what the user is actually doing on their desktop.

*Does the chat window absorb the platform? Does the platform absorb the AI?
Or does something new emerge from the combination?*

The Building Blocks

Four open standards defining how AI integrates with the financial desktop



MCP

Model Context Protocol

Universal connector
between
AI models and tools/data



MCP-Apps

Interactive Responses

Tool calls that trigger
interactive HTML apps
in the conversation



FDC3

FINOS Standard

Defines what apps exist,
what
they do, what context they
handle



A2UI

Agent-to-UI Protocol

Open protocol for agents
to
dynamically generate UIs

Pattern 1:

"The Blender"

Pull to chat

User-initiated context retrieval from applications



Pattern 1: Pull to Chat

The Cracks

1 Trust

Must validate data. Open the source app.

2 Cognitive Overload

AI generates fast, humans consume slow. No progressive disclosure.

3 Action Gap

Summary is read-only. Leave chat to act.

Pattern 2:

"Universal Remote"

Navigate to source

1. Express intent

2. Discover apps

3. Open with context

4. Work in app

Chat UI

Show me Client A's TSLA position
and latest research

Searching for the right
applications...

User expresses intent in natural language. The LLM interprets what apps and context are needed.

Navigate to source

1. Express intent

2. Discover apps

3. Open with context

4. Work in app

Chat UI

Show me Client A's TSLA position and research

Found 4 apps matching your request...

MCP server

FDC3 desktop agent

Orchestrates apps in its domain. Connected to app directory for discovery.

Application directory

Market data app ✓

Portfolio analysis app ✓

Trade order app ✓

News & research app ✓

Compliance tool

Settlement tracker

```
{ "appId": "portfolio-analysis", "title": "Portfolio Analysis App", "description":  
"View and analyze client portfolio positions, P&L, and allocation", "categories":  
["portfolio", "analytics"], "interop": { "intents": { "listensFor": { "ViewPortfolio":  
{ "contexts": ["fdc3.portfolio"] }, "ViewPositions": { "contexts": ["fdc3.position"] }  
} }, "userChannels": { ... }, "appChannels": [ ... ] } }
```

LLM discovers apps via MCP tools that wrap the FDC3 desktop agent. Each app in the directory declares its intents and context types. Click an app to see its definition.

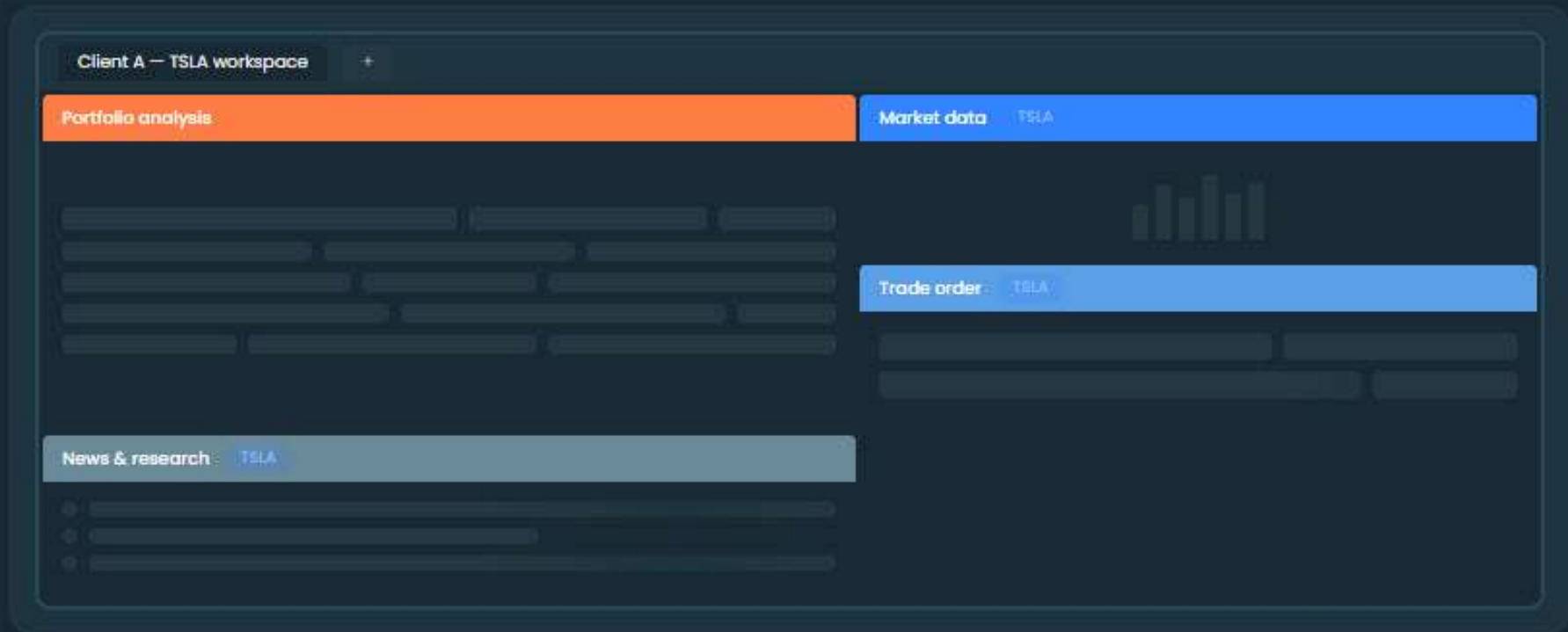
Navigate to source

1. Express intent

2. Discover apps

3. Open with context

4. Work in app



FDC3 desktop agent opens a workspace with all four apps tiled. Each app receives the right context — Client A for portfolio, TSLA for market data, orders, and research.

Navigate to source

1. Express intent

2. Discover apps

3. Open with context

4. Work in app

The screenshot displays a workspace titled "Client A — TSLA workspace". It features several panels: "Portfolio analysis" (orange header), "Market data TSLA" (blue header) containing a bar chart, "Trade order TSLA" (blue header) with input fields and a "Submit" button, and "News & research TSLA" (grey header) with a list of news items. A table with 3 columns and 6 rows is visible in the portfolio analysis section.

Client A — TSLA workspace

Portfolio analysis

Market data TSLA

Trade order TSLA

Submit

News & research TSLA

- TSLA Q1 earnings beat estimates
- Analyst upgrades target to \$285
- New factory production ramp

User works in the apps — portfolio data for Client A, TSLA charts, order entry, and research are all live and contextually linked.

Pattern 2: Navigate to Source

"Universal Remote"

Why This Matters



Cheap

Data may never touch the LLM



Safe

No hallucination: source app is system of record



Private

No data privacy concerns



Trusted

User sees the real app, not a summary

Experience lands in the source application, not the chat.

Pattern 3:

Surface an app

1. App returned

2. Live in chat

3. The problem

4. User chooses

5. In workspace

The screenshot shows a chat window titled "Chat UI". A user message asks, "Show me live TSLA pricing and Greeks". The assistant responds with, "Here's a live pricing view for TSLA via MCP App:". Below this is a rich widget titled "TSLA — Live pricing" with an "MCP App" label. The widget displays a price of \$274.82, an increase of +3.41 (+1.26%), and a bar chart. Below the chart are buttons for Delta (0.72), Gamma (0.04), Theta (-0.18), Vega (0.31), and IV (42.3%). At the bottom of the chat is a "Message..." input field and a send button.

The LLM returns a rich MCP App — not just text. An interactive live pricing widget with Greeks appears directly in the chat.

Surface an app

1. App returned

2. Live in chat

3. The problem

4. User chooses

5. In workspace

The screenshot shows a chat interface with a dark theme. At the top, the title "Chat UI" is displayed. The chat history includes a user message: "Show me live TSLA pricing and Greeks". The assistant responds with: "Here's a live pricing view for TSLA:". Below this is a widget titled "TSLA - Live pricing" with a red "MCP App" label in the top right corner. The widget displays the current price as "\$275.10" in large white text, followed by "+3.69 (+1.36%)" in green. A small bar chart is visible at the bottom of the widget. Below the widget, the user asks: "What's the 30-day trend?". The assistant responds with a blue bubble containing the text: "TSLA is up 12% over 30 days with increasing momentum...". At the bottom of the chat, there is a text input field with the placeholder "Message..." and a red send button with a white arrow.

The app is live and interactive inside the chat. The user continues the conversation while the pricing widget updates.

Surface an app

1. App returned

2. Live in chat

3. The problem

4. User chooses

5. In workspace

Chat UI

TSLA is up 12% over 30 days...

What about AAPL exposure?

Your AAPL position is 340 shares at \$186.42...

Compare the two


TSLA +12% vs AAPL +3.2% over 30d...

Pull up the risk report

Running VaR analysis across your portfolio...

Message...



 The live pricing app has scrolled away. The user can't see real-time updates, can't interact with it, and has no easy way to return to it. Live data in a scrolling chat is fundamentally ephemeral.

The problem: as the conversation continues, the MCP App scrolls out of view. Live data is lost. There's no way to pin it or keep it visible.

Surface an app

1. App returned

2. Live in chat

3. The problem

4. User chooses

5. In workspace

Chat UI

Show me live TSLA pricing and Greeks

Here's a live pricing view for TSLA:

TSLA — Live pricing

MCP App

This app has live data. Where would you like it?

Keep in chat

Open in workspace

Message...



The user chooses where to surface the app. The chat offers an option: keep it inline or open it in the workspace. For live data, workspace is the clear choice.

Surface an app

1. App returned

2. Live in chat

3. The problem

4. User chooses

5. In workspace

The screenshot displays a chat interface on the left and a persistent workspace on the right. The chat UI shows a conversation where the user asks for Tesla pricing and Greeks, then asks about AAPL exposure, and finally compares the two. The workspace, titled 'TSLA workspace', shows 'TSLA - Live pricing' with a price of \$275.10, a change of +3.69 (+1.36%), and a bar chart. Below the chart are five data points: Delta 0.72, Gamma 0.04, Theta -0.18, Vega 0.31, and IV 42.3%. The workspace is labeled as 'Persistent'.

Chat UI

Show me live TSLA pricing and Greeks

Opened TSLA pricing in your workspace.

What about AAPL exposure?

Your AAPL position is 340 shares at \$186.42...

Compare the two

TSLA +12% vs AAPL +3.2% over 30d...

Message... →

TSLA workspace +

TSLA — Live pricing Persistent

\$275.10 +3.69 (+1.36%)

Delta 0.72 Gamma 0.04 Theta -0.18 Vega 0.31 IV 42.3%

The MCP App persists in the workspace. The conversation continues freely in chat while live pricing stays visible, interactive, and always up to date.

Three Patterns, One Framework

	Pull to Chat	Navigate to Source	Surface an App
WHEN	Need data from multiple apps blended together	App exists, user needs to get there with context	Need a purpose-built interface for the task
HOW	MCP	MCP + FDC3	MCP server returns interactive UI (MCP App)...in the future A2UI
GOOD FOR	Summaries, research prep, pre-call briefings	App discovery, context-aware launch, trade execution	Data visualisations, forms, interactive tools
LANDS IN	Chat window	Source application	Chat today; workspace on AI-centric desktop

The Generative UI Spectrum

From pre-built interfaces to fully AI-generated experiences

TODAY



Predefined UI

MCP Apps today. Tool call returns a known, pre-built interface. The developer built it in advance.

Maps, charts, forms

EMERGING



Declarative UI

AI describes what it wants. A rendering engine builds from a schema. Structured instructions, not raw HTML.

Schema-driven rendering

FRONTIER



Fully Generative UI

AI generates the entire interface from scratch. HTML, CSS, interactions. The A2UI frontier.

Dynamic dashboards, custom tools

The Desktop Is Changing

The answer is not simply chatbots bolted onto existing applications.

The operating model of the desktop itself is changing, reflecting the way we are using AI to support our work.

The chat interface does not disappear, but it becomes one of several ways the AI participates in the user's work — and not always the primary one.