The Future Was Rehearsed. Nobody Took Notes.

An analysis of Apple’s 1987 Knowledge Navigator, modern coding agents, Siri AI, the DMA dispute, and why the trusted agent layer now needs public rules.

DRAFT PREVIEW DENNIS HEDEGREEN ANALYSE DRAFT-LOCAL
HTTPS://HEDEGREENRESEARCH.COM/ARTICLES/THE-FUTURE-WAS-REHEARSED-NOBODY-TOOK-NOTES/

In 1987, Apple showed a future computer that still feels ahead of us.

I was not alive when Knowledge Navigator was released. It came out three years before I was born. So this is not nostalgia for me. I am not looking back at a childhood computer dream. I am looking at an interface fossil and realizing that it still points forward.

That is the strange part.

Some people saw Knowledge Navigator in a room in 1987. They watched a professor speak to a digital assistant that managed messages, calendar events, journal articles, university networks, maps, simulations, calls, printing, and delegated communication.[1]

Then they lived through the web, search engines, laptops, smartphones, apps, cloud computing, Siri, GitHub, modern IDEs, ChatGPT, and now AI agents.

Only now does the shape begin to make full sense.

Maybe the vision was not wrong.

Maybe it was waiting for the missing infrastructure.

This was never just a chatbot

The most misleading way to understand Knowledge Navigator is to call it an early Siri.

Yes, the assistant talks. Yes, there is a human-like face. Yes, the professor speaks naturally to the computer.

But speech is the least interesting part.

The transcript is not about a voice assistant answering trivia. It is about workflow.

The professor starts with messages. Then calendar. Then old lecture notes. Then unread journal articles. Then fuzzy memory search: “Flemson or something.” The assistant resolves it to John Fleming. Then it pulls research. Then maps. Then university networks. Then Brazil. Then thirty years of data. Then a colleague calls back. Then two researchers combine models and maps. Then they adjust a logging rate and inspect the result. Then the assistant prints the article. Then it handles a call while the professor is away.[1]

That is not a chatbot.

That is a workflow architecture wearing a human face.

It is a personal assistant, a communications center, a research interface, a scheduler, a simulation layer, and a delegated presence. Those are not separate features. They are one interface model.

The assistant is valuable because it sits between the person and the world: files, people, time, institutions, tools, networks, archives, and future action.

That is the agent layer.

The room laughed because it was impossible

The first showing matters.

The EDUCOM keynote transcript places the video inside a serious future-of-education argument. After the Knowledge Navigator sequence, the transcript records applause. Sculley then pulled the audience back from product fantasy into long-horizon reality: he described technologies incubating in universities and company laboratories, and said it was not unrealistic to expect them to appear in commercial products early in the 21st century.[2]

That moment is perfect.

They understood both things at once: the desire and the impossibility.

That was not a product launch.

It was a future object.

It did not go viral. That phrase belongs to a later internet.

It travelled the old way: keynotes, Macworld, trade press, employee meetings, press rooms, tapes, memory, myth. Computerworld later described Apple's five-minute Knowledge Navigator video as "stunning" and noted the idea of scanning electronic archives for articles, maps, and charts. Macworld called the assistant a "talking-head agent" that could greet the user, remember appointments, and fetch database information.[4][5]

The room could applaud because it wanted the future.

The old nerds remembered because it was not wrong.

Siri was not the final act

I am not writing this as an Apple loyalist. From the outside, Apple has often looked slow in AI. Siri has spent years being more joke than revolution. Compared with ChatGPT, Claude, Gemini, and modern coding agents, Apple's public AI story has often looked controlled, delayed, and underwhelming.

But that may also be the wrong frame.

Maybe Apple was never trying to win the chatbot race in the same way. Maybe the model was never the product. Maybe the interaction layer was.

Look backward and Apple's last forty years can be read less as a random series of devices and more as a long, uneven construction of the conditions needed for a personal agent.

The Macintosh made the graphical interface personal. The iPod made navigation simple and intimate. The iPhone put the computer, camera, location, sensors, apps, messages, and identity in the pocket. The App Store created a controlled action ecosystem. iCloud and continuity connected the user across devices. Secure Enclave and privacy branding built a trust story. Shortcuts and App Intents began turning apps into callable actions. Apple Intelligence and Siri AI now try to place a personal agent on top of that machine.[7][8]

That does not mean there was a secret perfect masterplan from 1987 to now. That would be mythology.

Apple's products can be read as the slow construction of a trusted human-computer interface.

If that reading is right, then Siri AI is not just Apple catching up to chatbots.

It is Apple trying to ship the product Siri was never strong enough to become.

Siri is not the final act.

Siri may have been the placeholder name.

Codex shipped one narrow Navigator before Apple shipped the broad one

This is where it gets uncomfortable for Apple.

Because in one narrow domain, software engineering, the Knowledge Navigator pattern is already becoming real somewhere else.

Codex is not Knowledge Navigator. It does not manage your whole life. It does not coordinate your lunch, your mother, your lecture, your maps, your journal reading, and your colleague's simulation.

But structurally, Codex is close to one piece of the old vision.

It has files. Tools. Context. Tasks. Time. Parallel work. A supervised agentic workspace. It does not merely answer. It acts inside a working environment. OpenAI describes the Codex app as a "command center for agentic coding," with built-in worktrees and cloud environments where agents can work in parallel across projects.[6]

More importantly, software gives agents something ordinary consumer computing often lacks: tight feedback loops.

A coding agent can inspect files, edit code, run a terminal command, read an error, run tests, compare a diff, and try again. It has a compiler. It has logs. It has failing tests. It has a repository state. It has a boundary around the work.

That is why coding agents matter. They do not only show that AI can write text. They show what happens when a model is placed inside an environment where action can be tested.

That is the bridge to Apple.

If Apple wants to ship the consumer version of the Navigator, the operating system has to become a safe action environment. Ordinary apps must become more like callable libraries. The agent needs permissions, state, context, feedback, and boundaries. App Intents and Shortcuts are not side features in that world. They are part of the missing infrastructure.

The old Apple video was not predicting a chatbot that knows facts.

It was predicting a computer that could move through a task with you.

Codex does that for code. It is domain-specific, technical, and still imperfect. But it has already shipped a working version of the pattern: an agent that operates across files, tools, instructions, tests, and project context.

The consumer agent problem is harder because human life does not compile.

Code gives agents a judge.

Human life does not.

A compiler can reject a mistake before the program ships. A test suite can fail. A diff can be reviewed. A build can break safely inside a sandbox.

But email, money, relationships, photos, location, reputation, family, institutions, and work do not have a clean test suite.

A failed build is one thing.

A mistakenly sent message, leaked photo, wrong payment, deleted memory, or damaged relationship is another.

That is why the consumer Navigator cannot be governed by trial and error.

The system cannot wait for the error message after the damage is done.

When life does not compile, governance has to move upstream.

The agent’s powers must be scoped before action. Delegation must be visible. High-risk execution must require confirmation. Logs must exist. Permissions must be revocable. Platform owners cannot keep secret privileges for their own assistants and call it privacy.

That is the bridge from Codex to the DMA fight.

This is not really about Siri.

It is about who gets to control the trusted agent layer when the operating system becomes an action environment.

The DMA fight is not really about Siri

If Siri AI becomes a Knowledge Navigator-like layer on iPhone, it is not just another app. It becomes the trusted interface between the user and the device. It can see context, answer from the screen, search across personal information, connect to apps, and take action.[7][8]

That is power.

Apple's argument is understandable: if deep agent access is opened carelessly, privacy and security risks become severe. A malicious or compromised assistant with access to personal files, messages, photos, apps, and actions is not just a bad chatbot. It is a dangerous operator inside a person's digital life.

Apple says it designed a "Trusted System Agent" as an intermediary that would let virtual assistants access Siri AI-like capabilities safely on EU devices, and says the European Commission rejected its proposals.[9]

But the EU's concern is also understandable: Apple cannot be allowed to give Siri privileged access to the operating system and call that privacy if no other assistant can compete through a safe equivalent route.

The counterpoint matters. AP reported the Commission's position as: the DMA does not prohibit Apple from introducing new products, the decision not to launch was Apple's, and exemptions were not available.[10]

Both concerns can be true.

This is not a simple fight between innovation and regulation.

It is a fight over who controls the trusted agent layer.

The bad version of openness is obvious: every assistant gets raw access to everything.

The bad version of platform control is also obvious: Apple gives Siri secret superpowers and calls it user protection.

Neither is good enough.

The navigator needs a constitution

The missing architecture is not a slogan.

It should look more like a constitution for delegated digital power.

Article I: Delegated powers must be visible. The user must know what the agent is allowed to do.

Article II: Access must be strictly scoped to execution. No assistant should receive the whole device when it only needs one capability.

Article III: Actions must be logged. Delegated action must leave a readable trace.

Article IV: High-risk actions require confirmation. Money, deletion, publication, security changes, identity, and human contact cannot be silent operations.

Article V: Capabilities must be revocable. Permission is not surrender.

Article VI: Platform owners cannot reserve secret agent privileges for themselves. If Siri can safely use a capability, there must be a safe governed route for other trusted agents.

Article VII: The user steers. Delegated execution must never hide the architecture so completely that the human becomes a passenger.

That is the missing middle between naïve openness and closed platform control.

Not every assistant should get everything.

But no platform should be allowed to turn privacy into a monopoly over the future interface.

The Jobs test

Knowledge Navigator was not Steve Jobs' project. That matters.

The lazy sentence is wrong:

"Steve Jobs predicted AI."

No.

Knowledge Navigator belongs to the Sculley-era Apple imagination, higher education strategy, interface design, and the lineage of personal computing dreams around knowledge, learning, and networked information.[3]

But Jobs still belongs in the article as a test, not as a prophet.

The classic Apple idea of the computer as a bicycle for the mind gives us the right question. A bicycle amplifies the human. It extends effort. It gives range without removing agency. You still steer. You still feel the terrain. You still understand that movement happened because you participated.

An AI agent can become a bicycle for the mind.

Or it can become a chauffeur for the mind.

That difference matters.

A chauffeur AI completes the task.

A bicycle AI leaves the user more capable than before.

This is not a small distinction. It is the difference between automation and agency.

Take DMARC.

A chauffeur AI says: here are the SPF, DKIM, and DMARC records. Copy and paste them.

A bicycle AI says: email was built on an older trust assumption. SMTP did not begin as a hostile global battlefield. Spam, spoofing, phishing, and domain impersonation broke that trust model. SPF, DKIM, and DMARC became layered attempts to restore identity, authentication, and accountability on top of infrastructure that was not originally designed for today's abuse patterns.

That is not history as decoration.

That is architecture debugging.

Better architecture comes from understanding both the decisions and the rejected paths.

A debugger shows what the system does.

A good AI can help debug why the system became that way.

That is the version of AI worth defending.

Not the machine that hides the world.

The machine that makes the world more readable.

Wozniak's missing warning

Wozniak built machines a person could take apart and understand.

Knowledge Navigator imagined a machine that could take the person apart and understand them.

The agent era has to keep both.

That is the missing warning.

If we only chase machines that understand us, we may lose machines that we can understand. We may gain convenience and lose agency. We may get perfect assistants and forget how the system works.

A good personal computer did not only serve the user.

It invited the user into the machine.

A good agent should do the same.

We rehearsed the future and treated it as entertainment

This is the part that annoys me.

Humanity has spent decades watching versions of this future. Science fiction gave us talking computers, artificial companions, machine judgment, automation, post-work societies, corporate capture, surveillance, synthetic labour, intelligent interfaces, and computers woven into daily command structures.

Star Trek understood something most current AI debates still avoid: the computer is not just a productivity tool. It is part of social architecture.

It sits inside a world of rules, roles, missions, trust, command, responsibility, and public purpose.

And yet now, when the agent layer is finally becoming technically plausible, the public debate often collapses into small slogans.

AI will take jobs.

The EU is blocking innovation.

Apple is behind.

Chatbots are stupid.

None of these statements are useless. Job loss matters. Regulation matters. Apple's execution matters. Chatbots are often overhyped.

But as the main frame, it is too small.

AI does not only threaten tasks. It changes the architecture around work: who can act, who can coordinate, who owns the tools, who controls access, who sees the whole system, who becomes dependent on platforms, who gets upgraded, who gets deskilled, and who is allowed to delegate.

We did not fail to imagine the future.

We failed to treat imagination as preparation.

The future interface needs public thought

Knowledge Navigator did not become real because computers learned to talk.

Talking was only the surface.

It becomes real when computers can move through files, networks, calendars, research, colleagues, institutions, tools, and actions on our behalf.

That is why the current AI debate feels too small.

This future was already rehearsed in old videos, science fiction, interface dreams, and decades of cultural imagination. Some people watched it in 1987 and waited. Some of us found it later and realized it was still ahead.

Now the infrastructure is arriving.

The question is no longer whether the computer can answer.

The question is who gets to act through it.

The future was rehearsed.

Nobody took notes.

Now the navigator needs a constitution.

Source Notes

1. Internet Archive, “Knowledge Navigator,” Apple Inc., 1987. Used for Apple attribution, publication date, official description, and transcript sequence. archive.org/details/knowledge-navigator (https://archive.org/details/knowledge-navigator)
2. Internet Archive, “EDUCOM 1987 John Sculley Keynote Address On The Knowledge Navigator.” Used for keynote context, post-video applause marker, and Sculley’s long-horizon framing. archive.org/details/Apple-Computer-Educom-1987-John-Sculley-Keynote-Address-On-The-Knowledge-Navigator (https://archive.org/details/Apple-Computer-Educom-1987-John-Sculley-Keynote-Address-On-The-Knowledge-Navigator)
3. Bud Colligan, “How the Knowledge Navigator video came about,” Dubberly Design Office. Used for EDUCOM origin context, higher-education strategy, Sculley keynote setting, and production history. dubberly.com/articles/how-the-knowledge-navigator-video-came-about.html (https://www.dubberly.com/articles/how-the-knowledge-navigator-video-came-about.html)
4. Computerworld, August 8, 1988, Internet Archive full text. Used for contemporary trade-press reception, archive scanning/articles/maps/charts framing, and “stunning” video wording. archive.org/stream/computerworld2232unse/computerworld2232unse_djvu.txt (https://archive.org/stream/computerworld2232unse/computerworld2232unse_djvu.txt)
5. Macworld, November 1988, Internet Archive full text. Used for contemporary “talking-head agent” reception and appointment/database description. archive.org/stream/MacWorld_8811_November_1988/MacWorld_8811_November_1988_djvu.txt (https://archive.org/stream/MacWorld_8811_November_1988/MacWorld_8811_November_1988_djvu.txt)
6. OpenAI, “Codex.” Used for official “command center for agentic coding” product framing, worktrees, cloud environments, and parallel agents. openai.com/codex (https://openai.com/codex/)
7. Apple Newsroom, “Apple introduces Siri AI, a profoundly more capable and personal assistant,” June 2026. Used for personal context, onscreen awareness, actions across apps, and world-knowledge framing. apple.com/newsroom/2026/06/apple-introduces-siri-ai-a-profoundly-more-capable-and-personal-assistant (https://www.apple.com/newsroom/2026/06/apple-introduces-siri-ai-a-profoundly-more-capable-and-personal-assistant/)
8. Apple Newsroom, “Apple aids app development with new intelligence frameworks and advanced tools,” June 2026. Used for App Intents, app actions, onscreen awareness, and Xcode 27 agentic-coding workflow claims. apple.com/newsroom/2026/06/apple-aids-app-development-with-new-intelligence-frameworks-and-advanced-tools (https://www.apple.com/newsroom/2026/06/apple-aids-app-development-with-new-intelligence-frameworks-and-advanced-tools/)
9. Apple Newsroom, “Due to DMA, Siri AI delayed in EU for iOS 27 and iPadOS 27,” June 2026. Used for Apple’s DMA argument and Trusted System Agent framing. apple.com/newsroom/2026/06/due-to-

dma-siri-ai-delayed-in-eu-for-ios-27-and-ipados-27 (https://www.apple.com/newsroom/2026/06/due-to-dma-siri-ai-delayed-in-eu-for-ios-27-and-ipados-27/)

10. Associated Press, “Apple and Brussels blame each other for delaying launch of its latest AI products in Europe.” Used as the EU-side counterpoint. apnews.com/article/siri-ai-europe-apple-5d18df90b03e4e98ac528c8802e2b531 (https://apnews.com/article/siri-ai-europe-apple-5d18df90b03e4e98ac528c8802e2b531)

RELATION MEMORY

CONTINUES AI Should Be Infrastructure, Not Authority · Both pieces argue that AI must become supporting infrastructure without becoming hidden authority.

NEARBY Today's AI Is the Worst AI You Will Ever Use · Both pieces read AI through the systems and interfaces it is becoming, not only through the current product moment.

NEARBY We Connected AI to the Internet Without Noticing · Both pieces treat the next AI problem as infrastructure, permission and delegated action.

CANONICAL URL: HTTPS://HEDEGREENRESEARCH.COM/ARTICLES/THE-FUTURE-WAS-REHEARSED-NOBODY-TOOK-NOTES/
PDF VIEW GENERATED: 2026-06-09T18:00:45.761Z