Considus

TECHNICAL WHITE PAPER

Catchlight for iPhone

What AI features cost you in a notes app

Where a model runs decides what an AI feature costs you. On-device, confidential server compute and ordinary servers compared, with the questions to ask any notes app.

Applies to Catchlight 1.0

Introduction

Every notes app I look at has grown a sparkle icon in the last eighteen months, and mine has not.

That is the sort of gap a reviewer notices, and the polite reading is that a small studio has not got round to it yet. The honest answer is that it is a decision, it was written down before it became fashionable to have one, and this paper is the reasoning rather than the excuse.

It is also a genuinely interesting question, which is why it is a paper and not a line on a features page. "Does this app use AI?" turns out to be close to meaningless. What matters is where the model runs, what it is allowed to read, what leaves the device, and what is kept afterwards. Those four answers vary enormously between products that describe themselves identically, and almost nobody publishes them clearly.

So this is written to be useful whichever app you use, including the ones that do it well. Section 6 scores Catchlight on the same scale and section 7 says where our position costs you something real, because it does.

What this covers, and what it doesn't

This covers the three places a model can run, what each means for the confidentiality of a note, the questions worth asking of any app that offers these features, and where Catchlight stands.

It does not evaluate whether the features are any good. Summarisation, smart sorting and natural-language search are useful to a lot of people and I am not going to pretend otherwise from behind a product that has none of them.

It does not name and shame specific apps. Their architectures change faster than a versioned document can track, and a paper that got one wrong would deserve everything it got. The questions in section 5 are the durable part, and you can put

them to any app yourself.

It is not a claim that Catchlight is more private *because* it lacks these features. That is a tempting sentence and it is not quite true. An app that runs a model strictly on-device, with no network call, can be exactly as confidential as one that runs no model at all. The difference is narrower and section 6 is careful about it.

1 Three places a model can run

Almost every AI feature in a notes app is one of three architectures, and the difference between them is the whole subject.

On the device. The model ships with the app or the operating system and runs on your phone. Your note is read into memory on hardware you own and no part of it crosses the network. This is the strongest position and it is genuinely achievable now, which was not true three years ago.

On the maker's servers. Your note is sent to a machine the company runs, processed there, and the answer comes back. Everything then depends on their retention policy, their access controls, and their ability to keep both under legal and commercial pressure.

On a third party's servers. Your note is sent to somebody else entirely, usually a model provider, often via the app maker. Now two organisations have handled your writing, and the second one's policies were not written with your notes app in mind.

There is a fourth arrangement that deserves naming separately, because it is designed specifically to close the gap between the first and the second.

Confidential server compute. Apple's Private Cloud Compute is the prominent example. The design intent is that a request goes to a server the company cannot inspect, running signed and publicly inspectable code, retaining nothing after the answer is returned. It is a serious piece of engineering and considerably better than an ordinary server.

It is also still a server. The property "the data never left your device" is simply not true of it, and the property that replaces it is "the data left your device under conditions the maker has designed and published and invited scrutiny of". That is a real and meaningful improvement. It is a different claim, and the two get conflated constantly in marketing.

2 What each one actually costs you

The useful way to compare them is to ask what has to go wrong before somebody reads your note.

ARCHITECTURE	WHAT HAS TO FAIL
On the device	Your phone is compromised, or the app is dishonest about running locally
Confidential server compute	The published design is wrong, its guarantees are not what they appear, or the attestation is not doing what it claims
The maker's servers	Retention outlasts the request, an internal tool is misused, a breach occurs, or an order arrives
A third party's servers	Any of the above, at either company

Each row down that table adds parties and adds time. The bottom row is not disgraceful and plenty of good software lives there. It is simply a different bet, and it is one you should get to make knowingly rather than discover in a settings screen.

The retention question is the one people underrate. A request that is processed and immediately forgotten is a very different thing from one that is logged for abuse monitoring, kept for thirty days, and used to improve a model. All three are common, all three can be described as "we process your data to provide the service", and only one of them means your note stops existing elsewhere when the answer comes back.

3 Where the operating system already reads your notes

This is the part that surprises people, and it applies whether or not your notes app has any AI features at all.

On a modern iPhone, system-level assistant features have API access to Apple's own Notes and Reminders. That access is Apple's to grant and it is documented, and it is not a scandal. But it means "my notes app has no AI" and "no model can read my notes" are separate statements, and the first does not get you the second if your notes are in Apple Notes.

For a UK reader there is a sharper version of this. Notes and Reminders are two of the ten iCloud categories that lost end-to-end encryption when Advanced Data Protection was withdrawn here, which paper 4 covers at length. So the system-level access question and the key-holding question stack on the same data.

An app that stores its own content, encrypted, outside the system's notes database is not subject to that access. That is a real architectural difference and it is worth understanding before you choose where to type.

4 Why Catchlight has none of this

The Strategic Roadmap rules it out, and the wording is unambiguous: AI summarisation, suggestions and smart sorting are **not planned at any horizon**. Two reasons are recorded and both are honest.

It contradicts the simplicity principle. Catchlight is built around one idea, the Take, which becomes a note or a task or a reminder depending on what you give it. A feature set that suggests, sorts and summarises pulls hard against an app whose whole argument is that you should not have to decide up front. Every product that adds intelligence to a simple tool has to answer what happens when the intelligence is wrong, and the answers are all complexity.

The alternatives each cost something the product will not spend. A server breaks the zero-knowledge property that the rest of this programme is built on, and it is not a small break. It means an account, infrastructure, a key somewhere, and a company that can be compelled. On-device inference avoids all of that and costs battery and storage instead, on a phone the user did not buy for this.

There is a third reason that is not in the roadmap and should be said anyway. **I do not think I could do it well.** A small studio adding a model to a notes app in 2026 is adding the least differentiated part of the product and the part most likely to embarrass it. The apps doing this properly have teams on it.

5 Questions to ask any notes app

None of these need expertise and all of them have a checkable answer.

Where does the model run? If the answer is not immediate and specific, that is itself informative. "On-device where possible" is a common phrasing and it means some requests leave.

What can I turn off, and what happens if I do? An AI feature that cannot be disabled is a decision the app has made for you.

Does it work in airplane mode? The most direct test in this paper. Turn the network off and use the feature. If it works, the model is local. If it fails, it is not. This is test 3 in How to tell whether a notes app is actually private, applied to one feature rather than the whole app.

What is kept, and for how long? Look for a number. A privacy policy that describes processing without ever giving a retention period is describing an intention rather than a limit.

Is my content used for training? And separately, is it used to *improve the service*, which is a phrase that sometimes means the same thing and sometimes does not.

Who else sees it? If a third-party model provider is involved, their policy applies to your writing too, and you agreed to it by using an app rather than by reading it.

6 Catchlight on the same questions

QUESTION	CATCHLIGHT
Where does the model run?	**There is no model.** No summarisation, no suggestions, no smart sorting, none planned at any horizon
What leaves the device?	**One network request in the whole app**, and only when you share a web link into it, to fetch that link's title and preview image. Nothing else the app does. What you ask iOS to do is a separate question and section 7 answers it
Does it work in airplane mode?	**Entirely.** Capture, search, edit and delete all work with no network. Sync is a file operation in your own folder
Is anything used for training?	**No.** There is nothing to train on: your content is encrypted before it leaves the device under a key the studio does not hold
Can the OS assistant read your Takes?	**No.** Takes live in Catchlight's own encrypted store, not in the system notes database
What does the studio see?	**Nothing.** No account, no server, no analytics, no crash reporting with content in it
What is kept, and for how long?	**180 days, and that is the only retention period in the product.** The studio keeps nothing, because there is no server to keep it on. In your own folder a deleted Take leaves a tombstone, a short record saying it was deleted, and those are pruned after 180 days. It sits inside the encrypted manifest, so your provider cannot read it and neither can I. Section 5 tells you to look for a number, so here is mine

The honest qualifier. An app running a strictly on-device model with no network call could answer these questions nearly as well as this. Having no model is not automatically more private than having a local one. What it does mean is that there is no configuration to get wrong, no feature that might quietly change architecture in a future release, and nothing to audit.

That is a smaller claim than "no AI means more private", and it is the one I can actually defend.

7 What this position costs you

Read this with section 6, because section 6 alone reads like an advertisement.

The app does none of it, and some of it is good. Summarising a long note, finding a thing you half remember by describing it, turning a paragraph into a task list. Real conveniences, and Catchlight builds none of them. If you want them built in, this is the wrong app and that is a legitimate reason to choose another one.

Your phone can still do some of it, and I am not going to stop it. The editor is an ordinary iOS text view, so what iOS offers wherever you type works inside a Take, and that includes Writing Tools. Select a paragraph, ask for a proofread or a rewrite, and Apple handles it, sometimes on the phone and sometimes on their servers, and the phone decides which rather than you. A third-party keyboard with full access can do the same, reading the field you are typing in and sending it to its own service. **Catchlight has no AI. You can still choose to use some, and that choice is yours rather than mine.** Today the app sets no restriction, so Writing Tools behaves inside a Take as it does anywhere. Before version 1.0 that becomes a switch in Settings, three positions and **off by default**, sitting beside the Siri and Spotlight control that already works the same way. A third-party keyboard stays outside it, because blocking the keyboard you chose to install would be the same mistake in the other direction. Section 3 already drew this distinction about somebody else, that "my notes app has no AI" and "no model can read my notes" are separate statements, and it would have been convenient to leave it applying only to Apple's Notes. It applies here. The app runs no model and sends nothing to one, and the layers above it are yours to point wherever you like.

Search is literal. It matches words you type against words you wrote. It will not find "the thing about the boiler" if you wrote "heating engineer coming Tuesday". Semantic search is one of the genuinely useful applications of a local model and its absence is felt.

This is a stated intention, not a shipped guarantee. The roadmap says not planned at any horizon. Roadmaps are not contracts and I am the one who would change it. What makes the position more than a promise is the architecture underneath: with

no account and no server, a future server-side feature could not be added quietly, because it would require infrastructure that does not exist and an account you do not have. You would notice.

Nothing here protects you from your own phone. All of this concerns what leaves the device. Somebody holding your unlocked phone reads what you read, and paper 2 takes that case seriously.

8 Don't take my word for any of it

Turn off the network and use the feature, in any app, including this one. It is two minutes and it settles where the work happens more reliably than any policy document.

Read the retention line, not the processing line. Nearly every privacy policy says it processes your data to provide the service. The question is what is kept afterwards, and a policy that never gives a period is answering a different question.

Check what your operating system can already reach, separately from what your app does. If your notes are in the system notes database, the app's own position is not the whole answer.

Treat the roadmap line as the weakest thing in this paper, because it is the one claim you cannot check. Catchlight's roadmap is an internal document. The sentence in section 4 is quoted from it word for word rather than characterised, which is the most I can offer and is still only my word. What you can check is the architecture underneath it, and section 7 says why that is the part worth trusting. Adding a server-side feature would need an account you do not have and infrastructure that does not exist, so it could not arrive quietly whatever a roadmap says.

References

Standards and guidance

OWASP Mobile Application Security Verification Standard, MASVS-PRIVACY, the category covering what an app transmits about its user and what a third party receives

UK General Data Protection Regulation, Article 5, purpose limitation and storage limitation, which are the two principles a retention period actually answers to

Apple, Private Cloud Compute, the published design for confidential server compute described in section 1

Companion documents

The Catchlight encryption architecture, for why "nothing to train on" is a structural property rather than a policy

Protecting your notes without Advanced Data Protection, for the UK position on Notes and Reminders referred to in section 3

How to tell whether a notes app is actually private, for the seven tests, of which test 3 is the airplane-mode check applied here

Version history

VERSION	DATE	CHANGE
1.0	1 September 2026	First release