We cannot *see anything*

What on-device privacy guarantees, what it leaves open, and what it costs the lab

MITWA LAB, MUMBAI

IN PLAIN WORDS

Bloom runs on your iPhone, and what you say to it and what it remembers about you are never sent to us. That is all the title means. Some things do leave the phone: if you speak instead of typing, your words can go to Apple's speech service, purchase checks go out, and the app's App Store label lists usage, advertising and diagnostic data. Because we hold no copy of your conversations, there is no database of ours that could be stolen to expose everyone at once. That does not protect a phone someone else can open, a backup of it, or a flaw in the phone's own software. And because we cannot see what Bloom says to anyone, all we know about its mistakes in conversation is what we tested before release and what people tell us. We show how little those can prove.

ABSTRACT

An on-device companion is private in a particular sense: what a person says and what the companion remembers never reach a system its maker controls. We call this privacy by architecture and ask what it guarantees, what it leaves open and what it costs the lab. This is a theory and literature note and reports no new measurement of Bloom. In a model where phones are compromised independently, removing the maker's store lowers each person's chance of exposure by up to the store's breach probability and makes the exposure of many users together exponentially unlikely in their number, where a central store leaves it as likely as one breach. The model leaves out common causes such as an operating-system flaw. The guarantee covers content only, and not the phone, its backups, the platform, speech sent for transcription, purchase checks or the data declared on Bloom's App Store label. On cost, voluntary reports can bound a failure rate from below and never from above, the expected number of failures before the lab holds k reports is k/ρ at reporting rate ρ if the release lasts that long, and a clean test of n items can certify at 95% confidence that expected failures stay below one for only about $n/3$ uses. By that measure Bloom's 400 memory checks cover 134 memory claims, and, without a known floor on the rate at which people report, nothing seen after release will add to them.

01 Introduction

The release notes for Bloom on the App Store end with the line "And we still can't see anything" (App Store, 2026). This note takes that sentence as its title. What the lab cannot see is what a person says to Bloom and what Bloom remembers, because neither is ever sent to Mitwa. The title does not claim that nothing leaves the phone, or that Mitwa receives no data of any kind: voice-to-text can go to Apple's speech service, purchase checks leave the phone, and the app's public privacy label declares usage, advertising and diagnostic data (Section 3).

The care is owed to what a companion hears. An analysis of the privacy policies of six US developers of frontier models, published in 2025, found that all six appear to use people's chats to train their models by default, and that some keep those chats indefinitely (King et al., 2025). On 19 May 2025 the Italian data protection authority announced a fine of 5 million euros on the company behind the companion Replika, having found that as at 2 February 2023 it had identified no legal basis for its processing; an appeal was pending at the announcement (Garante per la protezione dei dati personali, 2025).

Bloom takes the other route: the model, the memory and every reply stay on the phone. Our earlier notes met that fact twice, and both times it cut against us. Once an invented memory ships we will not observe it unless the person tells us (Mitwa Lab, 2026a), and a

behaviour damaged by compression stays invisible unless someone tells us (Mitwa Lab, 2026b). This is a theory and literature note. It contains no new measurement of Bloom: every Bloom number comes from the two earlier notes, and every other number is derived or labelled as an illustration. The contributions are these.

- Definitions of privacy by architecture and by policy, a table placing each kind of data a companion handles, and a proposition on what removing the maker's store changes (Sections 3 to 5).
- Three propositions on the cost to the lab: what reports can show, what noticing a failure costs, and what a clean test can certify (Section 6).
- The strongest opposing designs, our limits, and what a pre-release evaluation must report (Sections 7 to 9).

Bloom is a companion and makes no claim to be a therapist.

02 Related work

Privacy by architecture. Spiekermann and Cranor separate two ways of engineering privacy. Privacy by policy rests on notice and choice over data a company holds. Privacy by architecture minimises the collection of identifiable data and favours storage and processing on the client, and they judge that it generally protects people more reliably (Spiekermann and Cranor, 2009). Local-first software makes the same move for applications in general: the copy on a person's own device is the primary one, so no central database holds everybody's data (Kleppmann et al., 2019).

Evidence without observation. After n trials with no failure, the upper limit of the 95% interval for the rate is about $3/n$ (Eypasch et al., 1995), the rule our earlier notes relied on. After release, drug safety depends on voluntary reports, which capture little: across 37 studies of adverse drug reactions, the median share that went unreported was 94%, with an interquartile range of 82% to 98% (Hazell and Shakir, 2006). Automatic reporting shows what is lost without it: error reports from beta releases of one operating system led to more than 5,000 bug fixes after static analysis and model checking had found over 100,000, and such a bug was 4.5 to 5.1 times likelier to be fixed than one a person reported (Glerum et al., 2009).

03 Definitions

A *channel* is any path by which information about a person can leave the phone: a network request, a copy of the phone's storage, or the person's own act of sharing. For a party A, which may be Mitwa, the platform or anyone else, the *view* V_A is everything that arrives over any channel at systems A controls. Write X_c for a person's data of class c and X_{-c} for all their other data. Class c is private from A *by architecture* when

$$V_A = g(X_{-c}, \xi) \tag{1}$$

for some function g and randomness ξ independent of (X_c, X_{-c}), which amounts to V_A being independent of X_c given X_{-c}. The view is then built without the class, and nothing A later decides, suffers or is ordered to do can reveal it from A's own systems, because A never held it; a later build of the software could still change what leaves the phone (Section 5). Independence from X_c alone would not be enough, since ξ could be X_c masked by X_{-c}, from which g recovers X_c. The class is private from A *by policy* when V_A does depend on X_c and what protects the data is the conduct of A: access rules, retention limits, a promise. Definition (1) does not say that A learns nothing about the person, since data outside the class, such as the hour at which an app is opened, can be informative in its own right.

Table 1 places for Bloom each class of data a companion handles. *Conversation content*, typed or spoken, is what the person says and what the companion replies; *memory* is what the companion keeps about the person. Mitwa states that neither is sent to it, and that the memory and every reply stay on the phone. Spoken content, as audio, can go to Apple's speech service if voice-to-text is used. Even towards Mitwa the statement gives (1) only on an assumption this note cannot verify, that nothing else in Mitwa's view, the usage data below included, is computed from what is said or remembered; a count of messages would break (1) with no content sent. By itself the statement establishes the weaker property that no copy of the content reaches Mitwa. Bloom is bought inside the app through the App Store (App Store, 2026), and purchase checks leave the phone.

For *usage and diagnostic signals* our source is the public record. On 12 November 2026 the privacy label on the App Store page of the Mitwa AI app, which carries Bloom, declared as data that may be collected and linked to the person usage data, with advertising data the one type under it, for the developer's advertising or marketing, analytics and product personalisation, and as data not linked to them diagnostics, for analytics and app functionality. The page says Apple has not verified the label (App Store, 2026). Apple gives which advertisements a person has seen as an example of advertising data, and counts data as collected when it leaves the device where the developer can reach it for longer than a request takes;

the label declares nothing in Apple's category of user content (Apple, n.d.-a). This note does not know what the signals contain or who receives them.

WHAT	STATUS	DETAIL AND BASIS
Typed words, replies, memories	Not sent to Mitwa	Never sent to Mitwa, as content or training data; memory and replies stay on the phone (Mitwa's statement).
Spoken words	Can reach Apple	Voice-to-text, if used, can send the audio, and so what was said, to Apple; never sent to Mitwa (Mitwa's statement).
Purchase checks	Leave	What they carry is not known to this note.
Usage, advertising, diagnostics	Declared	On the store label as data that may be collected. Content unknown to this note.
The phone and its backups	Not covered	Whoever can open the phone or a backup can read it.
The operating system	Not covered	Keyboard, microphone, storage and network are the platform's.
What a person shares	Not covered	Anything copied elsewhere, a report to us included.
Later versions	Not covered	The property belongs to one build.

Table 1. What stays on the phone, what leaves it and what the architecture does not cover, for Bloom. The label is as read on 12 November 2026 (App Store, 2026).

04 What the architecture guarantees

4.1 Nothing to take, hand over or reuse

If (1) holds, no event on Mitwa's side can expose from Mitwa's systems what a person types or says to Bloom. Neither a breach of our systems nor an employee nor a request from anyone for a person's conversations can, since our systems hold nothing to produce; a change of owner or of policy could reach them only through a later build that sends what is on the phone, which Section 5 takes up. Reuse is excluded in the same way: Bloom sends no training data to Mitwa, so nothing a person says to it can enter a later model of ours. A policy can promise each of these; under (1) each is a fact about what the software sends, which can be tested. When a cloud service says it does not log user data, researchers usually cannot confirm it (Apple, 2024), whereas someone examining what the app sends from a phone they control can look for a channel that breaks (1), though no such examination can prove that none exists. That binds us too.

4.2 One breach does not expose everyone

Consider N people over a year. Under either design a person's phone is a way in: if it is compromised, that person's record is exposed. Suppose this happens to each phone independently with probability q_d. A cloud companion also keeps every record in the maker's store. Suppose the store is breached with probability q_s, independently of the phones, and that a breach exposes every record. On the device there is no such store. A person's chance of exposure under the two designs is

$$p_{\text{dev}} = q_d, \qquad p_{\text{cen}} = q_d + q_s(1 - q_d) \tag{2}$$

Proposition 1 (Breach exposure).
Under these assumptions, (i) each person's chance of exposure is lower on the device by $q_s(1 - q_d)$, and the expected number of people exposed is lower by $Nq_s(1 - q_d)$. (ii) For any share φ of users with $q_d < \varphi \le 1$, the chance that at least φN people are exposed is at least q_s with a central store, whatever N is, and on the device it is at most the bound (3).

$$\Pr\left(K_{\text{dev}} \ge \phi N\right) \le \exp\left(-N D(\phi \,\|\, q_d)\right) \tag{3}$$

Here $D(a \,\|\, b) = a \ln(a/b) + (1 - a) \ln((1 - a)/(1 - b))$. Part (ii) is the larger gain. With a central store, the exposure of everyone together is as likely as a single breach, and no growth in the number of users dilutes it. On the device the same outcome needs N separate compromises in one year, and even a share φ above q_d needs φN of them (200 in the illustration below), so while phones fail independently its probability falls exponentially with N. Take, as an illustration, 10,000 people and a 1% yearly chance both of a store breach and of any one phone being compromised. A person's chance of exposure is 1% on the device and 1.99% with a store. The chance that 2% or more of all users are exposed in the year is at least 1% with a store and below 10^{-16} on the device (Figure 1). We do not know the real probabilities for any product.

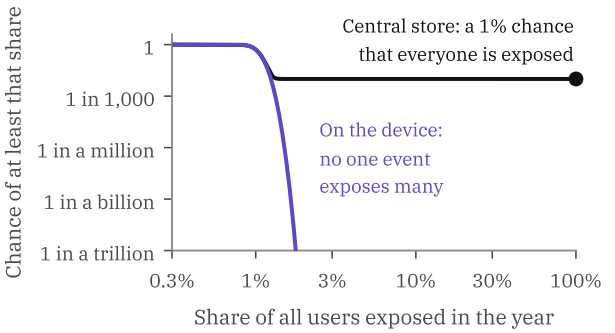


Figure 1. The chance that at least a given share of all users is exposed in a year, under the model of Proposition 1, computed exactly with the illustrative values of the text; bound (3) lies above the device curve. Both axes are logarithmic.

The breach record has this shape. Among 6,422 breach events from 2000 to 2015, those above fifty thousand records were under a tenth of the events and held more than 99% of the records exposed (Wheatley, Maillart and Sornette, 2016). The large event is the one the architecture removes from the maker. The proposition says nothing about q_d or about one cause that reaches many phones at once; Section 5 takes up what it leaves alone.

05 What it does not guarantee

Everything in Section 4 concerns Mitwa's view. Content on the phone is as safe as the phone: anyone who can open it, or any program that has compromised it, can read it, and a shared phone needs no attacker.

Apple's documentation says that third-party app data can be stored in iCloud Backup, that under the default setting Apple holds the keys to a backup, and that under the optional Advanced Data Protection setting the backup is end-to-end encrypted (Apple, 2026). For a person who backs up under the default, whatever an app allows into the backup sits with the platform, and the central store that Proposition 1 removed from the maker reappears there. Whether Bloom's conversations and memories enter a backup is a detail of how Bloom is built, which this note does not state.

The keyboard, the microphone, the storage and the network belong to the operating system, and (1) is silent about the platform's view. Apple documents two kinds of speech interface: for one it describes the person's voice being sent to Apple's servers, and for the other it says no audio is sent (Apple, n.d.-b). This note does not state which Bloom uses, so for spoken content (1) can fail with respect to Apple while holding with respect to Mitwa.

Last, (1) is a property of one build. A later build could add a channel, and could send what earlier builds left on the phone. What prevents that is the maker's conduct, so across versions the guarantee rests on policy after all, with the difference that a change is open to the examination of Section 4.1.

06 What it costs the lab

6.1 Prevention and correction

A maker can keep a failure rare by bounding its rate before release, which is prevention, or by noticing it in use and repairing it, which is correction. A product whose conversations pass through its maker's servers can do both. Bloom's conversations never reach us. We keep no logs of what is said and cannot see the model's replies in use, so for a failure of content, such as an invented memory, correction is left with one channel, a person telling us. We assume throughout that the declared signals of Section 3 carry no count of such failures; this note does not know what they contain.

6.2 What can be known after release

Let θ be the rate of some failure per interaction; for memory, per memory claim. If a pre-release test of n independent items shows no failure, the largest rate which that result leaves open at confidence $1 - \delta$ is

$$u(n) \;=\; 1 - \delta^{1/n} \;\approx\; \frac{3}{n} \qquad (\delta = 0.05) \tag{4}$$

For Bloom this is the 0.75% ceiling of the 400 memory checks (Mitwa Lab, 2026a). Note 03's 143-turn behaviour bank and 60-scenario safety suite left 2.1% and 4.9% open, but these bound the share of items on which the smaller build did worse than the larger, a loss relative to that build (Mitwa Lab, 2026b). A cloud maker that reviews a share s of M interactions a day adds sM items to its sample every day, and an on-device maker adds none from the field. If the reviewed items stay clean, and treating test and reviewed items as draws at one rate (Section 6.5 relaxes this), the cloud maker's ceiling after t days is $u(n + sMt)$, which falls roughly as $1/t$ (Figure 2). Reviewing 100 memory claims a day, an illustrative figure, would take the 0.75% ceiling to 0.088% in a month and to the 1 in 10,400 of note 01 in 308 days; at 1,000 a day it would take 31 days (note 01 took an illustrative three claims a day and a 10% yearly chance of meeting one invented memory). On the device the ceiling stays where the test left it, unless the lab runs more checks of its own.

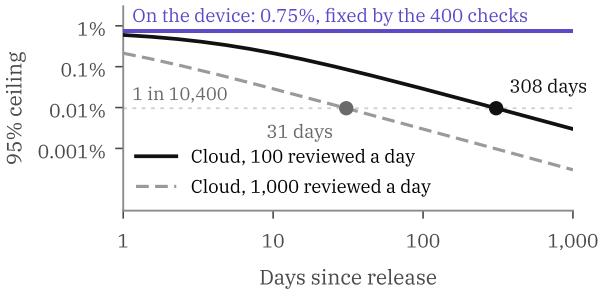


Figure 2. The 95% ceiling on a failure rate after a clean test of 400 items, from (4), if every memory claim reviewed after release is clean and drawn at the test's rate. Review volumes are illustrative; both axes are logarithmic.

6.3 What reports can and cannot show

Suppose each failure in use is reported to the lab independently with probability ρ. A report needs the person to notice the failure and to take the trouble, and for a failure that goes unnoticed, which note 01

argued is the worst kind, ρ is zero. This note has no figure for ρ or for M.

Proposition 2 (Silence is not evidence). *Let each of Mt interactions fail independently with probability θ and each failure be reported independently with probability ρ. The number of reports R(t) is binomial with probability $\theta\rho$, so it depends on θ and ρ only through their product, and none arrives with probability (5). Hence (i) for any θ, (5) is as close to 1 as desired for small enough ρ; (ii) an upper confidence bound on θ computed from the reports alone, valid at confidence $1-\delta$ for every θ and ρ in (0, 1], must be at least 1 with probability at least $1-\delta$; (iii) when Mt is known or bounded above, the reports do support a lower bound, since $\theta \geq \theta\rho$.*

$$\Pr\,(R(t)=0) \;=\; (1-\theta\rho)^{Mt} \;\approx\; e^{-\theta\rho Mt} \tag{5}$$

Reports can show that a failure exists and, when the volume of use is bounded above, that it is at least so frequent. They cannot show that it is rare. A cloud maker chose s, knows M and holds a random sample with a known denominator, while voluntary reporting samples at a rate set by the people who were failed. Silence becomes evidence only if the lab has a lower bound on the volume of use and can put a floor $\rho_{\min}$ under that rate, and then the field is worth roughly $\rho_{\min}Mt$ test items in (4).

6.4 The price of noticing

Suppose the lab acts once it holds k reports of the same failure, since one report of something it cannot inspect may not be enough. Let F_k be the number of failures that have happened when the kth report arrives, and T_k the number of days.

Proposition 3 (The price of noticing). *If interactions arrive as a Poisson process at M a day, each fails independently with probability θ, and each failure is reported independently with probability ρ, the expected values of F_k and T_k are given by (6). The first depends neither on θ nor on M.*

$$\mathrm{E}[F_k] \;=\; \frac{k}{\rho}, \qquad \mathrm{E}[T_k] \;=\; \frac{k}{\theta\rho M} \tag{6}$$

As long as the release lasts until the lab notices, a lower failure rate, or fewer users, does not reduce the expected number of failures that reach people before it does; it spreads the same number over a longer time. Over a release of E interactions the expected number before the kth report is at most the smaller of k/ρ and θE. Under sampling the formulas hold with s in place of ρ, and s is the maker's to choose. We have no measurement of ρ for a companion. In drug safety, a field with dedicated reporting systems, the median under-reporting of 94% (interquartile range 82% to 98%) across 37 studies (Hazell and Shakir, 2006) corresponds to ρ = 6%, within about 2% to 18%. Table 2 uses that figure and a lower one. At the 1 in 10,400 of note 01, with one failure in a hundred reported, the third report is expected after 8.5 years and 300 failures; a release that lasted one year would instead meet about 35 failures in expectation and end without a third report with probability above 99%.

TRUE RATE	REPORTED	FIRST REPORT	THIRD REPORT	FAILURES BY THEN
0.75%	6%	2.2 days	6.7 days	50
0.75%	1%	13.4 days	40 days	300
1 in 10,400	6%	173 days	520 days	50
1 in 10,400	1%	2.8 years	8.5 years	300

Table 2. Expected time to the first and third report, and failures by the third, from (6), at the 0.75% ceiling and at the 1 in 10,400 of note 01, for 1,000 memory claims a day, if the release lasts that long. Volume and reporting rates are illustrative.

6.5 What the test must carry

With correction this weak, prevention has to carry the weight, and (4) says how much a test can carry.

Proposition 4 (The burden on the test). *Suppose a clean test of n independent items is the only evidence that bounds θ from above, and a release will see E interactions. (i) If the items are drawn from the situations met in use and $h < E$, a valid upper confidence bound on θ can certify at confidence $1-\delta$ that the expected number of failures in use is at most h only if the test satisfies (7), and the bound (4) does so whenever (7) holds. (ii) If test items are drawn from a distribution Q of situations and use from a distribution P, and the build fails with the same probability in a given situation in both, the failure rates θ_P in use and θ_Q in the test satisfy (8), where d_{TV} is the total variation distance, and (8) can hold with equality.*

$$n \;\geq\; \frac{\ln(1/\delta)}{-\ln(1-h/E)} \;\approx\; \frac{3E}{h} \tag{7}$$

$$\theta_P \;\leq\; \theta_Q + d_{\mathrm{TV}}(P,\,Q) \tag{8}$$

Part (i), read backwards, gives the reach of the tests we have. A clean test of n items can certify at 95% confidence that the expected number of failures stays below one for the first $1/u(n)$ uses, about $n/3$, and we say the test covers those uses. For the 400 memory

checks that is 134 memory claims, counted across every phone. For the 143-turn bank and the 60-scenario suite, read the same way, it is 48 and 20 interactions over which an expected compression loss relative to the larger build stays below one. Holding the expected count below one across a year at an illustrative 1,000 claims a day would take about 1.1 million clean checks, and note 01's per-person standard about 31,100. Both exceed any test we have run.

Part (ii) is the cost that no amount of testing buys back. The second term in (8) measures how far real use departs from what the test imagined. A cloud maker estimates it by looking at what people say to its product; we cannot see that, so for us it stays unknown. If, as an illustration, real use matched the test except that 2% of situations were of kinds the test never drew from, the ceiling from 400 clean checks would be 2.7%, and no number of further checks on the kinds it did draw from could bring it below 2%.

07 The strongest opposing view

The case against our design is that the choice it presents, between a maker who sees everything and one who sees nothing, is a false one, and two lines of work support it. The first keeps computation in a cloud and limits what the operator can see. Apple's Private Cloud Compute is a publicly described example: by its published design, requests are processed without being retained, staff have no privileged access at run time, and Apple committed to publishing the software image of every production build so that outsiders can check what runs (Apple, 2024). Such a design buys larger models than a phone runs while aiming at the property in (1). The request still leaves the phone in a form the server can read, so the guarantee rests on hardware, attestation and published code all being sound, which is more to trust than the absence of a channel. Traffic can also leak without being read: from the lengths of encrypted response tokens, researchers closely reconstructed the opening segment of 29% of one commercial assistant's responses and inferred the topic of 55% (Weiss et al., 2024). A service that retains nothing gives up the transcripts a maker would sample, though, unlike us, it sees each request while serving it and could count failures as they pass; Apple's design provides for server metrics and error logs gathered without undermining its privacy protections (Apple, 2024).

The second line learns aggregates without collecting raw data. In the local model of differential privacy each device randomises its own answer before sending it, so a collector can estimate a population rate while every single report stays deniable (Dwork and Roth, 2014), and Apple has described running such a scheme, opt-in, at the scale of hundreds of millions of users (Apple, 2017). Applied to our problem, each phone would send one randomised bit saying whether a given failure occurred in some period. Under the classic two-coin scheme, in which a phone reports its true bit with probability 3/4 and which is (ln 3)-differentially private (Dwork and Roth, 2014), the share v of people affected is estimated from N phones with variance $4\lambda(1-\lambda)/N$, where $\lambda = 1/4 + v/2$, against $v(1-v)/N$ if the bit were sent in the clear. Estimating an illustrative share of 0.75% with a standard error of half its size would take about 54,000 phones in place of about 530, a factor of about 100 whatever precision is asked for. Federated learning goes further and trains a shared model from updates computed on each device (McMahan et al., 2017); what a device shares can itself give data away, since gradients from small batches have been inverted to recover training text token for token, though that attack worked only for batches of up to eight and failed when enough of the gradient was pruned (Zhu, Liu and Han, 2019).

These designs would buy real things: an estimate of a failure rate across everyone who uses the product and, in the federated case, a model that learns from use. Each also sends something computed from a person's conversations off the phone, and each needs the phone to recognise the failure for itself. Bloom does none of this: it sends no training data to Mitwa, and we know of no channel in it that reports failures of content. Whether the trade is worth making is a decision about what the title of this note should mean, and a technical note is the wrong place to make it.

08 Limitations and broader impact

Nothing here was measured. Propositions 1 to 4 are statements about models, and every probability, reporting rate and volume attached to them is an illustration. We do not know the chance that a store or a phone is compromised or the rate at which people report failures, and we have no count of Bloom's replies or memory claims in use. We assert (1) for conversation content and memory on Mitwa's own statements together with the assumption, unverified here, that the declared signals carry nothing computed from conversation content or memory, and we report no independent audit.

The models are simple. Proposition 1 assumes independent compromises, which an operating-system flaw exploited at scale would break, and that a compromised phone exposes the same record under either design, which may flatter the

device, since a phone holding a person's whole history is a richer prize than one holding a cache. It also has a breach expose every record, where real breaches are often partial. Propositions 2 and 3 assume one reporting rate, where in practice it varies with the person and the failure. Proposition 4 assumes independent test items, which note 01 already doubted of our own checks.

The label. Mitwa's list of what is never sent to it includes an ad profile. The store label declares advertising data linked to the person, with the developer's advertising or marketing among its purposes. We cannot reconcile the two from the material we have, and we have not tried to explain the label. The same store page also describes the app with the words "nothing leaving your phone" (App Store, 2026), which is wider than the scope set in Section 1 and wider than this note supports.

The oath and Apex. The lab's site carries an oath that includes the lines "We can't see what you say" and "Your words never sit on our servers". Apex is planned as a cloud model, which sends a person's words to a server to be processed, so the oath as written will not hold for Apex. We record the conflict; resolving it belongs to the people who own the oath.

Broader impact. The property described here protects people from the maker and from a mass breach of the maker, and it leaves them alone with the product. A failure that harms someone in a conversation no one at Mitwa can see will reach us only if they say so, and Section 6 shows how weak a signal that is. A maker that chooses this design owes people evidence gathered before release in proportion to what it has given up afterwards, and a plain account of what does leave the phone. This note makes no clinical claim.

09 What a pre-release evaluation has to report

When the test is the only evidence there will ever be, its report should give at least these.

- For each failure tested, the number of items, the number of failures and the ceiling (4), given as a rate and as the number of uses it covers in the sense of Section 6.5, about $n/3$.
- The largest rate the maker would accept, fixed before the test, and the items it requires by (7).
- Where the test situations came from and what use they stand for, since the second term of (8) cannot be estimated afterwards; for us, that no user conversation was available.
- How heavily it sampled situations in which a failure would go unnoticed, which by Proposition 2 the field will never send back.
- Who built and who judged the test, whether anyone outside the lab did either, and whether items that share a conversation were counted as independent (Mitwa Lab, 2026a).
- The exact build tested and what leaves the phone in it, class by class as in Table 1, since an update inherits none of the old evidence; and, after release, the count of reports for each failure, as a lower bound.

10 Conclusion

Keeping a companion on the phone gives one thing with unusual firmness. What a person types or says to the companion and what it remembers are absent from the maker's systems, so nothing held on the maker's side can reach them and no breach of the maker can expose everyone together. That rests on Mitwa's statement, on an unverified assumption about the declared usage data, and on the build, since a later one could change what leaves the phone. Proposition 1 bounds mass exposure only while phones are compromised independently, so a platform flaw exploited at scale or a backup held by the platform falls outside it. Beside the guarantee, spoken words can reach Apple, and purchase checks and the label's declared data leave the phone.

The price is that we cannot see what our product says in use. Reports can tell us that a failure exists and cannot tell us that it is rare, the expected number of failures that reach people before we notice, if we notice within the release, is fixed by how often people report, and a clean test can certify at 95% confidence only about a third as many uses as it has items, which for the 400 memory checks behind note 01 is 134 memory claims. The work this leaves is larger tests built by people other than us, an independent check of what leaves the phone, a public account of what the data declared on the store label contain, and an answer to the conflict between the oath and Apex.

Disclosure. Bloom is a product of Mitwa AI, and this note was written by the lab that builds it. No user conversation is available to us, and none was used. Reproducibility. Every result in Sections 4 and 6 is closed-form and proved in Appendix A, and every number follows from the formulas and the stated inputs.

References

Apple (2017). Learning with privacy at scale. Apple Machine Learning Research, 6 December 2017. machinelearning.apple.com/research/learning-with-privacy-at-scale

Apple (2024). Private Cloud Compute: A new frontier for AI privacy in the cloud. Apple Security Research, 10 June 2024. security.apple.com/blog/private-cloud-compute/

Apple (2026). iCloud data security overview. Apple Support, published 5 January 2026. support.apple.com/en-us/102651

Apple (n.d.-a). App privacy details on the App Store. Apple Developer. Opened 12 November 2026. developer.apple.com/app-store/app-privacy-details/

Apple (n.d.-b). Asking permission to use speech recognition. Apple Developer Documentation. Opened 12 November 2026. developer.apple.com/documentation/speech/asking-permission-to-use-speech-recognition

App Store (2026). Mitwa AI. Product page, India storefront, with version history and the App Privacy label. Opened 12 November 2026. apps.apple.com/in/app/mitwa-ai/id6755034468

Dwork, C., and Roth, A. (2014). The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3–4), 211–407. doi.org/10.1561/0400000042

Eypasch, E., Lefering, R., Kum, C. K., and Troidl, H. (1995). Probability of adverse events that have not yet occurred: A statistical reminder. BMJ, 311(7005), 619–620. doi.org/10.1136/bmj.311.7005.619

Garante per la protezione dei dati personali (2025). AI: The Italian Data Protection Authority fines company behind chatbot "Replika". Press release, 19 May 2025. garanteprivacy.it/home/docweb/-/docweb-display/docweb/10132048

Glerum, K., Kinshumann, K., Greenberg, S., Aul, G., Orgovan, V., Nichols, G., Grant, D., Loihle, G., and Hunt, G. (2009). Debugging in the (very) large: Ten years of implementation and experience. In Proceedings of SOSP 2009. sigops.org/s/conferences/sosp/2009/papers/glerum-sosp09.pdf

Hazell, L., and Shakir, S. A. (2006). Under-reporting of adverse drug reactions: A systematic review. Drug Safety, 29(5), 385–396. doi.org/10.2165/00002018-200629050-00003

King, J., Klyman, K., Capstick, E., Saade, T., and Hsieh, V. (2025). User privacy and large language models: An analysis of frontier developers' privacy policies. arxiv.org/abs/2509.05382

Kleppmann, M., Wiggins, A., van Hardenberg, P., and McGranaghan, M. (2019). Local-first software: You own your data, in spite of the cloud. In Proceedings of Onward! 2019, 154–178. doi.org/10.1145/3359591.3359737

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. In Proceedings of AISTATS 2017. arxiv.org/abs/1602.05629

Mitwa Lab (2026a). Memory you can trust: The asymmetric cost of invented memories in AI companions. Technical note 01.

Mitwa Lab (2026b). A companion in 2 GB: What a smaller model buys on a phone. Technical note 03.

Spiekermann, S., and Cranor, L. F. (2009). Engineering privacy. IEEE Transactions on Software Engineering, 35(1), 67–82. doi.org/10.1109/TSE.2008.88

Weiss, R., Ayzenshteyn, D., Amit, G., and Mirsky, Y. (2024). What was your prompt? A remote keylogging attack on AI assistants. arxiv.org/abs/2403.09751

Wheatley, S., Maillart, T., and Sornette, D. (2016). The extreme risk of personal data breaches and the erosion of privacy. arxiv.org/abs/1505.07684

Zhu, L., Liu, Z., and Han, S. (2019). Deep leakage from gradients. In Advances in Neural Information Processing Systems 32 (NeurIPS 2019). arxiv.org/abs/1906.08935

A Appendix A: Proofs

Proposition 1.

(i) On the device a person is exposed exactly when their phone is compromised. With a store they escape only if neither the phone nor the store fails, with probability $(1 - q_d)(1 - q_s)$, which gives (2); expected counts follow by adding over N people. (ii) A breach of the store exposes all N, so the chance that at least φN are exposed is at least q_s. On the device K is binomial with N trials and probability q_d. For any $\lambda > 0$, Markov's inequality gives $\Pr(K \geq \varphi N) \leq e^{-\lambda\varphi N}(1 - q_d + q_d e^{\lambda})^N$, and for $q_d < \varphi < 1$ the right side is smallest at $e^{\lambda} = \varphi(1 - q_d)/[q_d(1 - \varphi)]$, where it equals (3). At $\varphi = 1$, (3) is ${q_d}^N$, the exact chance that every phone is compromised. The exponential decay rests on independence across phones; a common cause that compromises many phones at once lies outside the model.

Proposition 2.

Each interaction yields a report with probability $c = \theta\rho$, independently, which gives the binomial law and (5). (i) As ρ falls to 0 with θ fixed, (5) rises to 1. (ii) The pair (θ, ρ) and the pair $(1, c)$ give the same law. A bound U that is valid at $(1, c)$ satisfies $\Pr(U \geq 1) \geq 1 - \delta$ under that law, hence under (θ, ρ), and this holds even when Mt is known. (iii) $\rho \leq 1$, so $\theta \geq c$, and c can be bounded from below from the reports when Mt is known. If only an upper bound on Mt is known, the bound computed with it remains valid, since the chance of at least r reports grows with the number of interactions.

Proposition 3.

The number of failures up to and including the one that produces a report is geometric with mean $1/\rho$, and F_k is a sum of k independent such counts, so its mean is k/ρ. An interaction yields a report with probability $\theta\rho$, so the kth report arrives after $k/(\theta\rho)$ interactions on average, which for a Poisson process at M a day is $k/(\theta\rho M)$ days; with exactly M a day it holds to within one day. Over a release of E interactions, the failures met before the kth report or the end of the release are at most F_k, whose mean is k/ρ, and at most the failures in the whole release, whose mean is θE.

Proposition 4.

(i) A rate above $u(n)$ would give a clean test with probability below δ, so $\theta \leq u(n)$ at confidence $1 - \delta$. The expected number of failures in E uses is then at most $u(n)E$, which is at most h when $\delta^{1/n} \geq 1 - h/E$. Taking logarithms gives (7), which is defined because $h < E$, and $\ln(1/0.05) \approx 3$. Conversely, if a valid bound that is not randomised takes the value a after a clean test, then for every $\theta > a$ the clean result, which has probability $(1 - \theta)^n$, leaves θ uncovered, so $(1 - \theta)^n \leq \delta$ and $a \geq u(n)$; certifying h then needs $u(n)E \leq h$, which is (7). Since the items are drawn from use, θ is the rate in use. (ii) Let $f(x)$ be the probability of failure in situation x, with values in $[0, 1]$. Then $\theta_P - \theta_Q$ is the sum over x of $f(x)[P(x) - Q(x)]$, which is at most the sum of $P(x) - Q(x)$ over the situations where P exceeds Q. That sum is $d_{\mathrm{TV}}(P, Q)$, taken as the largest difference in probability that P and Q give to any set of situations, which is half their L1 distance. Equality holds when the build fails exactly in those situations.