S3ntiment

A protocol for provable, honest feedback

ABSTRACT

Every survey tool holds the connection between an answer and the person who gave it, and then promises to look after it. S3ntiment never creates it. No link table, no master key, nothing to leak and nothing to hand over: you cannot be compelled to produce a key you do not hold. What that buys immediately is a liability position that holds by construction rather than by policy. What it buys over time is harder to prove and worth more — answers that report what people think rather than what they are willing to say.

CONTENTS

1 The claim

Every survey tool in common use holds the connection between an answer and the person who gave it: a respondent ID, an email field, an IP address. It then secures that connection as well as it can, and asks the respondent to trust that it will. That connection is what leaks in a breach, what a court order asks to be produced, and what a data protection authority asks about first.

S3ntiment does not create it.

In the Netherlands, a *stempas* arrives in the post before every election. It has your name on it. You take it to the polling station, they check it against the register, they cross you off, and they hand you a ballot. That ballot is identical to every other one in the station — no serial number, no barcode, no mark that connects it to you. You mark it, fold it, drop it in the box. When it is counted, it is just a mark on paper. The link between you and your vote never existed in the ballot itself. That is structural anonymity. Not a promise — a property of the design.

S3ntiment matches that, and then goes one step further. Your invitation gets you in, the way a *stempas* does: scarce, one-time use, provably yours. But the answer you give is never stored whole in one place. Think of it as tearing up the ballot and placing its parts in different bins. Each bin makes its own tally — a partial count that means nothing on its own. Then the tallies are combined into a result. No bin ever held a complete answer. No party ever saw the pieces meet. The Dutch ballot is anonymous because it carries no identifying mark. These answers are anonymous because they are never assembled in one place at all.

The claim is narrower and stronger than "we take your privacy seriously":

> The architecture is built so that no party can read a response in the context that could de-anonymise it — not the organiser, not the platform, not anyone. That is not a policy. It is what the cryptography is designed to guarantee.

"Cannot read" needs a precise reading, because the plain one is wrong. An individual response is not a secret object nobody may ever see. You can always read and retrieve your own answer, and open-ended text may come back in aggregate form. What the architecture makes structurally impossible is reading a response in the context that links it to its author — the identifying context a court order, an insider, or a platform operator could otherwise exploit. The property is the absence of that context, not the secrecy of the text.

That claim describes the architecture. The current deployment does not yet fully match it: the encryption infrastructure is provisioned under an account held by the application operator (§8.1). Section 8 states the full position, and a reader evaluating the claims below should read it.

Two things follow, and they are separated deliberately, because they rest on different kinds of evidence. The first is immediate and deductive: no stored link means no identifying record to leak, nothing to hand over under compulsion, and

a data-minimisation position that holds by construction rather than by policy. It follows from the architecture, and what it needs is implementation verification, not a result. That case is §2 and §3, and it is the reason to adopt.

The second is longer-run and empirical: measurement conditions that let an organisation see what people think rather than what they are willing to say. We expect it, and the reasoning is set out in §7 — but it is an extension of existing research rather than a finding already in hand, and it depends on trust accumulating inside a pool over time. It is upside, not a premise. Nothing in §2 and §3 depends on it.

2 What this removes

The acceptance vehicle is data protection, not cryptography. An HR department, a regulator, or a union does not adopt "a protocol." It adopts software that happens to be built on one, and it adopts it because of what its own counsel finds when they look. Three questions decide that review.

What personal data do you process? The defensible statement is not the absolute "none," which a DPA would and should contest. The organiser knows who it invited, invitations are scarce, derived identities are pseudonymous, and under GDPR pseudonymous data is still personal data where re-identification is reasonably likely by any means. The honest data-map answer is narrower and stronger: **no party holds data that identifies a respondent to it.** Invitees are known to the organiser by necessity, since it invited them. Everything after that sits under keys that no party can reverse to an identity. Identifiability is minimised to the point where no party holds a working link.

What are respondents' rights? Access, correction, deletion — buttons, not procedures. You own your record and can revise or erase it directly.

What is our liability? No respondent data the institution can leak by accident, no risk of violating a respondent's privacy, no compliance exposure. Where a breach notification is owed, there is no identifying record to report, so the notification is procedural and the process is followed.

A lawyer or DPO will know the underlying property by a second name: legal-process resistance. There is no link table and no master key, because the protocol never stored the connection it would take to identify a respondent. There is nothing to hand over under a court order, a subpoena, or a disclosure demand. You cannot be compelled to produce a key you do not hold. This is a property of the architecture; §8.1 qualifies what it means in today's deployment.

None of the above waits on adoption curves, behavioural change, or a research result. It is what the absence of a stored link entails, available on day one, and it is verifiable by reading the code and the key custody rather than by measuring anyone.

3 What existing tools do not do today

Existing survey tools fail on one structural fact: they hold the connection, and then secure it, which is a promise. The industry's breach record is the evidence. We do not create the link, so we do not need to secure it better. This is a difference in kind, not in effort.

Anti-fraud is the sharpest case. Mainstream tools prevent duplicate submissions by storing IP addresses, so their anti-fraud requires collecting the very personal data their privacy policy promises to protect. Ours requires none: participation is provable without anyone knowing who participated (§4). And where most platforms hand the survey creator the decryption keys, the standard here is higher: not even the survey creator can read a response in a context that identifies who wrote it.

The closest comparable makes the differentiator concrete. BlockSurvey is a private SurveyMonkey: responses are encrypted client-side, but the survey creator holds the decryption keys and can read everything. S3ntiment is for the case that even that design does not serve. To be precise: BlockSurvey holds creator keys today because that is its design, and it could in principle adopt the threshold pattern. The point is not that a competitor could never do this. It is that no current tool operates on the standard that even the creator cannot tie a response to a person.

4 How it works

A pool is a panel. Respondents join once and answer many surveys over time, changing their answers as their views change, under governance shared by the pool's operators — the working definition of a research panel. A standalone survey is just a pool with one survey. An employer running a continuous employee barometer is operating a panel under another name. A trade association, a union, a regulator, and a professional body each hold a known cohort and have no way to run it as a panel their members trust.

Four mechanics, each answering one way a survey can lie or leak. The protocol draft carries the full cryptographic detail. This is the shape.

1. No stored link. An invitation mints an anonymous identity by a one-way, deterministic derivation. You hold a durable anchor — a seed phrase, a hardware key, or an account secured for you. One standard route is a threshold oblivious PRF over a phone or email: a method where a network of nodes helps derive your identity without any single node learning the input. The anchor is kept independently of any app. From it, the protocol derives a fresh, unlinkable identity per pool. The same anchor always yields the same identity for a given pool, so you can return to your answers, but nothing stored or on-chain can reverse an identity back to the anchor, or to the person behind it. The app only ever holds the derived leaf, never the anchor. There is no link table and no master key. Derivation is gated behind an unused invitation, so membership cannot be tested by guessing addresses.

Identity is per-pool: the same anchor derives a different, unlinkable identity in every pool, so one individual carries a different key in each, and cross-pool correlation is impossible by construction. This is a privacy property chosen over cross-pool portability. A single portable identity across the network would be a network-effect win at a privacy cost, and we do not take it.

2. No identifying context. No party can assemble the context that would amount to any person's picture. Individual answers exist, and you can read and revise your own, but they never come together in a way that links an answer to its author. For numeric and rating answers the protection goes further: the value itself is split across the nodes of a neutral network and held inside hardware-isolated enclaves — isolated hardware that runs code a server operator cannot inspect. Any single fragment is noise. Only partial totals leave the secure environment, summed to a result that no single party ever saw whole. An answer to "how do you rate management" is stored as numbers that only mean something together, and those numbers never meet in one place.

3. Access by cryptography, not by administrators. Decryption is gated by conditions read directly by code inside an attestable enclave. The conditions: pool membership for the standard survey, the governing Safe — a multi-signature wallet the pool controls — plus a signer check for the privileged variant that contains scoring. There is no master key, no administrator account, no back door. The pool's constants are compiled into the enclave program, so the program's identifier is the authorisation. A pool's members can only ever run the programs built for that pool.

4. One participation, provably. Sybil resistance — preventing one person from submitting many times — comes from invitation scarcity plus a one-time on-chain nullifier burn: a record that proves an invitation was used, without revealing which one. Each invitation can be used once. Participation is provable without anyone, including the organiser, knowing who participated.

Ownership. Each respondent's record is owned by their own DID — a self-controlled identifier — inside an owned collection held under the pool's key, with the platform holding nothing on the write path. This is enforced cryptographically rather than honoured by the API, and it is what makes the respondent rights in §2 buttons rather than procedures.

What we deliberately do not claim: resistance to malware on your own device, or to an operator correlating network traffic. The trust assumptions are your own custody of your anchor, the neutral networks holding, and the enclave holding (§5).

Where free text lives, and who can read it. Numeric splitting invites the wrong inference here, so this is worth stating. Free text cannot be split into meaningless fragments the way a number can. It sits encrypted, and is released only under the same enclave-gated conditions: in aggregate form, or to your own account, and never in a context that identifies its author. Its protection rests on access control and the absence of identifying context, not on fragmentation.

The small-pool problem, stated as a non-claim. Within a pool, answers across surveys are linkable. That is the panel model, and it creates a statistical

rather than cryptographic re-identification risk. In a pool of nine, recomputing an aggregate after one person changes an answer, or publishing a sparsely populated cross-tab, can reveal an individual without anyone decrypting anything. The cryptography prevents decryption-based identification. It does not by itself prevent statistical identification. Minimum reporting thresholds, cell-size floors, and query budgets are load-bearing. Whether the protocol enforces them or leaves them to the operator is an open design decision, and we do not claim the protocol already enforces them. Across pools, linking is impossible by construction.

5 Sovereignty, and the dependency we will say out loud

The data layer is sovereign. The code is open source, the organiser owns its keys and its data, and a pool could take its keys and walk away. The walk-away test passes on code and on data.

The operational layer is not yet sovereign. The protocol runs on two neutral networks, and both are companies. This is the one place a conservative institution's continuity review will push hardest, and the honest answer is stronger than a clean one. If Nillion goes away, the data goes with it. The cryptographic layer sits behind interfaces, and the substrate could in principle be replaced — but none of that replacement has been exercised yet. This is a direction, not a portability claim already proven.

Why two networks at all? Nillion handles everything related to the survey answers — storage and compute. Lit handles access management: who can read a survey's questions as a respondent, who can read the results as an organiser. Splitting answer handling from access control is what keeps any single neutral party from seeing both the ciphertext and the code that unlocks it.

Cross-border data transfer is moot here. What sits on Nillion's and Lit's nodes is encrypted fragments, not readable data. A node operator compelled under a surveillance order can hand over what it holds, but what it holds is noise. No single fragment identifies anyone, and no single party can reassemble the pieces. The Schrems II question — whether personal data crosses a border to a jurisdiction with weaker protections — does not bite, because what crosses is ciphertext, not personal data.

The enclave is not a black box, and we do not treat it as one. The access checks and decryption run inside a hardware-isolated enclave that produces attestation — a proof of what program is running — so a pool can verify which program it is trusting before it trusts it, and the pool's constants are compiled in. But enclaves have a documented history of side-channel attacks, and a compromised enclave could read plaintext during computation. The mitigation is attestation, open source, and the compiled-in pool constants. The residual risk is named, not hidden. The trust assumption is that the enclave vendor and hardware attestation hold, and this is a live area we do not claim to have retired (§8.4).

Why now. The enabling conditions are met, and only recently. Applied cryptography — threshold OPRF, MPC secret sharing, enclave-executed conditions

— is available as products rather than research, and neutral networks such as Nillion and Lit provide a substrate where independent parties each do part of the work, which is what makes the claim checkable rather than promised. We call these enablers rather than a settled foundation deliberately. Both networks are young companies, and their being the current instance of a role is exactly why the dependency is a live question (§8.2). The substrate enabled the design. It is not yet what makes it durable.

6 Why there is no token

The protocol is token-neutral. It issues membership, not tokens, and carries no tokenomics. That is what lets an HR department or a public body adopt it as software — no volatility, no "is this a security" conversation — and it is why the review in §2 is a data protection review rather than a treasury one. Boring on purpose is the selling point, and it is priced and licensed like software.

The anonymous-key layer that underlies the protocol is what would make cryptographic compensation meaningful for a panel operator: a panel could pay respondents without ever learning who they are. A token economy is something a panel operator can strap on top of that layer. The protocol permits the layer and does not require it. The institutional reader is never asked to engage with the crypto-materiality question.

7 The horizon: better conditions, and better data

Everything to this point holds on day one and follows from the architecture. What follows is the longer argument. It is the reason the category is interesting rather than merely safer, but it is an extension of existing research rather than a finding in hand, and an institution can adopt on §2 alone and treat this as upside.

Organisations decide on what people say, not on what they think. The strategic plan "validated" by a survey, the engagement survey showing everything is fine, the regulatory filing built on responses nobody believed — these are not frauds. They measured speech accurately and were wrong about belief, because the conditions for honesty were absent. Nine people tell the manager "good plan." Privately, eight of ten expect it to fail. Both groups were honest — the first about speech, the second about belief.

The mechanism is well documented under two names. Organisational silence: people learn quickly which views are safe to express. Pluralistic ignorance: everyone privately disagrees, and everyone believes they are alone.

The empirical claim is that proof changes this where a promise cannot. Randomised-response methods hand the respondent a provable deniability mechanism for sensitive questions — you flip a coin, in effect, and answer one of two questions depending on the outcome, so even you cannot be sure which answer you gave. Under them, reported rates of stigmatised behaviour rise sharply and data quality improves across the board, because respondents stop hedging once they no longer need to protect themselves.

The extension is load-bearing, so it is stated plainly. That literature measures disclosure of stigmatised behaviour: drug use, undeclared income, infidelity. Carrying it to opinion honesty in a workplace survey extends the finding rather than citing it. We expect the extension to hold. The mechanism is the same — people hedge when honesty is costly — the research points that way, and it matches what anyone who has sat through an engagement survey already suspects.

But the expectation carries a condition, and the condition is time. A guarantee only changes what someone writes once they trust it, and trust in an unfamiliar guarantee accumulates through use. It does not arrive with the first invitation. The honest position is not that one survey will demonstrate the effect. It is that a pool should become more candid across successive waves as the guarantee becomes familiar and its consequences are seen to hold. It is one more reason the panel shape matters: a one-shot survey cannot show the effect even where the effect is real.

What would count against it, so the claim stays falsifiable: a flat candour trend across several waves in a pool where the guarantee has been explained, exercised, and not breached. An unremarkable first wave is the expected starting point, not evidence (§8.6).

The legibility constraint is the deepest objection here. The property is cryptographic, but the honesty gain depends on the respondent believing it. An employee who does not understand threshold cryptography — a method where multiple parties each hold a fragment, and no single party can reconstruct the whole — or enclaves is, from where they sit, trusting a promise again. A better-sourced promise, but a promise. Randomised response works partly because its deniability is legible in about one sentence: even you can't tell which of my two answers is real. Ours is not. A guarantee a respondent cannot understand is, to that respondent, not yet a guarantee. This is a design constraint, not a marketing footnote. Legibility is part of the product, it governs how fast the trend above can appear, and it is unsolved.

8 What is true today

The architecture is built so that no party can read in an identifying context. Several pieces of that architecture are not yet switched on. What follows is the state of each, and the direction.

1. **Encryption infrastructure sits under an application-operator-held account.** The Nillion builder key (nilDB storage, nilCC compute, delegations) and the enclave account are provisioned under an account held by the application operator today. This is the single fact that most qualifies both the headline claim (§1) and the legal-process position (§2). Direction: hand the account to the pool's own Safe. No modes, only who owns what.

2. **Dependency on two neutral networks.** Both are companies; both could change terms or pivot. Nillion migrated from Cosmos to Ethereum in early 2026. The interfaces that would replace them are not yet exercised (§5). The why-now argument claims the architecture is design-settled, not substrate-settled.

3. **Small-n statistical re-identification is not yet guarded by the protocol** (§4). Minimum reporting thresholds, cell-size floors, and query budgets are load-bearing, but they are not yet hardened protocol guards. Whether they should be protocol-enforced or operator-managed is an open decision. Until it is resolved, this is a stated non-claim, not a hidden one.

4. **The enclave threat model is a live trust assumption** (§5). Attestation via Blacklight is planned but not yet exercised. Direction: document the attestation story and the compromise response before institutional deployment.

5. **Organiser-side invitation supply.** The nullifier prevents duplicate participation. It does not prevent an organiser minting extra invitations. That is accepted today, since the organiser is the customer and controls its own cohort, but it is a governance decision to state rather than assume.

6. **The empirical expectation is not yet evidenced in our own data.** The RR-to-opinion-honesty extension (§7) and the small-cell behaviour (§8.3) are the two things live deployment must bear out before the horizon argument is fully leaned on. Because the first depends on trust accumulating inside a pool, the evidence for it is a candour trend across waves rather than a single result, which means the earliest deployments will not settle it either way. This is pre- or early deployment: the measurement is outstanding, not completed. §2 does not depend on this item.

The OPRF route for securing an anchor adds a threshold-network trust assumption: security rests on a threshold across the network's nodes plus a TEE-held share — a share kept inside trusted hardware — and on there being no single coordinator. That is materially weaker than a high-entropy seed, and it is the explicit reason the anchor model offers seed and hardware as first-class alternatives (§4).