# Kashi - Procurement / Security-Buyer Readiness Memo

*Deep research memo tailored to "Kashi - Progress & Project Overview (2026-04-21)"*

| | |
|---|---|
| **Purpose** | Turn Kashi's existing infrastructure instincts into a buyer-facing security and procurement story that can survive enterprise review. |
| **Audience** | Kashi core team; pilot sponsors; CISO / IT / procurement reviewers; partner-facing deck writers. |
| **Bottom line** | Kashi already has credible security primitives. The gap is not mainly engineering; it is control documentation, evidence packaging, and disciplined buyer language. |

Prepared: 21 April 2026 | Prepared for internal Kashi project use

**Decision line: Security review is not a side-show after the CEO pitch. For Kashi, the enterprise sale becomes real only when the project can answer a CAIQ/SIG-style questionnaire, explain exactly where data lives, show how tenants are isolated, state what the model boundary is, and prove how deletion, retention, incident response, and key management work in practice [1][2][3][4][5][6].**

## Executive summary

- Kashi already contains multiple security-relevant primitives in the current progress deck: Supabase-backed Postgres + Auth + Row-Level Security, a multi-tenant schema, planned evidence-vault encryption, role-scoped views, auditability language, four-tier retention, and a no-live-LLM production detection path [K1].
- Those primitives are useful, but they do not yet equal a buyer-grade assurance story. Procurement and security teams want control narratives, evidence, scoping boundaries, deletion semantics, incident response commitments, and a clean subprocessor / data-flow map - not just implementation notes [1][2][3][4].
- Kashi's strongest near-term sales move is not to claim certification too early. It is to package what already exists into a disciplined Security & Assurance Pack and to close the few gaps that would otherwise stop a pilot in security review.
- The most important immediate design decision is architectural: if Kashi wants a Japan-residency story, the regulated meeting data path must be kept inside a region-selected data layer (for example, Supabase Tokyo) and minimized on US-centric app-delivery infrastructure such as Vercel [3][7][8].
- The biggest blockers today are: no formal incident response summary, no written shared-responsibility matrix, no published subprocessor / region map, no tenant-isolation evidence pack, no documented key lifecycle, and no precise model/data boundary memo.
- Bottom line: Kashi is closer to procurement-ready than it may look, but only if the team stops presenting security as “we use RLS + encryption” and starts presenting it as “here is our control model, evidence set, and residual-risk statement.”

### Readiness snapshot

| Control area | What Kashi can credibly say now | Main gap to close | Sales risk if unanswered | Status |
|---|---|---|---|---|
| Data residency | Supabase supports a primary region per project, including Tokyo as a specific region; Kashi can choose a Japan-based database region instead of generic APAC defaults [3]. | A buyer-facing region map is missing, and Vercel's DPA says its primary processing facilities are in the United States, so Kashi must clearly limit what sensitive data flows through Vercel [7][8]. | High | A |
| Assurance path | Inherited controls exist through Supabase SOC 2 Type 2 and Vercel SOC 2 Type 2 / ISO 27001-certified posture [4][7]. | Vendor assurance is not Kashi assurance; there is no Kashi SOC 2 / ISO 27001 attestation or internal control matrix yet [4][6][13]. | High | A |
| Tenant isolation | The deck already claims org-level RLS and database-layer separation [K1]. Postgres / Supabase RLS is a strong mechanism when correctly implemented [5][9]. | Need policy-test evidence, service-role restrictions, negative tests, and eventual pen-test tenant-escape coverage. | High | A |
| Retention / deletion | The current deck already defines four-tier retention and a legal-hold concept [K1]. Supabase documents backup retention, PITR, and irreversible deletion including backups stored in S3 [10]. | Need customer-admin controls, deletion SLAs, backup caveats, export semantics, and legal-hold workflow documentation. | Medium | A |
| Incident response | There is auditability language in the product story [K1]. | There is no concise incident response summary, severity matrix, or customer notification procedure; this is a classic security-review blocker [11]. | High | R |
| Key management | The evidence-vault concept is strong: client-side key generation, encrypted | Need a full key lifecycle: generation, recovery, rotation, | High | R |

| | | | | |
|---|---|---|---|---|
| | snippets, server incapable of decryption by design [K1]. | revocation, lost-key handling, user offboarding, and explicit escrow / no-escrow policy [12]. | | |
| Model / data boundary | Kashi can say the live detection path is deterministic and does not use Claude in production today [K1]. Anthropic states commercial-product data is not used for training unless the customer opts into the Development Partner Program [14]. | Need a formal memo documenting exactly when text touches models, when it does not, what retention terms apply, and how optional AI features are segregated [14][15][16]. | Medium | **A/G** |
| Audit export | Supabase and Vercel both provide audit/logging capabilities [7][17][18]. | Supabase platform audit logs have notable limits: no dashboard export, no platform-audit log drain, retention varies by plan [17]. Kashi needs exportable app-layer audit logs. | Medium | **A** |

## 1. Why this matters to procurement, IT, and security buyers

Kashi's current deck is strong on institutional accountability and governance posture. That helps the CEO conversation. It does not by itself close an enterprise sale. In practice, a serious pilot or purchase will usually trigger some version of vendor security review, IT architecture review, procurement due diligence, and data protection review. Those reviewers are not grading “whether the idea sounds responsible”; they are grading whether the vendor can state where data resides, how access is bounded, how isolation is enforced, what happens in an incident, how deletion works, and which subprocessors or models ever touch customer data. [1][2][3][4]

This is exactly why cloud-assurance frameworks exist. The Cloud Security Alliance's Cloud Controls Matrix v4.1 is a cloud-security control framework with 207 controls across 17 domains, and its CAIQ companion is literally a set of yes/no questions for assessing security controls. Shared Assessments positions SIG Lite as a broad, high-level due-diligence questionnaire before a deeper review. In other words: procurement and security teams already have templates for how they will interrogate Kashi; Kashi should prepare the answers instead of waiting to be surprised by them. [1][2]

| **Practical implication for Kashi** |
|---|
| • Do not frame security as “we use Supabase + Vercel + encryption.” |
| • Frame security as “here is our scoped control model, here is our evidence, here is what the customer controls, here is what we control, and here are the residual limits.” |
| • That shift turns prototype infrastructure into something a buyer can sign off on. |

## 2. What the current Kashi materials already give you

The current progress deck already contains more buyer-relevant material than many early products. It explicitly states that Kashi uses Supabase for Postgres, Auth, Row-Level Security, and Storage; claims a multi-tenant schema with organization-level RLS; describes a four-tier retention model; emphasizes RBAC, k-anonymity, and audit trails; and says the production detection path does not use Claude live. It also outlines a v2 evidence vault in which client-side WebCrypto generates user keys and the server stores ciphertext it cannot decrypt [K1]. [K1]

- Those are not just engineering trivia. To a buyer, they map directly to core control questions: access control, tenant separation, data minimization, retention, model boundary, key custody, and auditability.
- The problem is that the current deck presents these as implementation details. Security buyers need them rewritten as control statements and evidence statements.

- Example: “Supabase + RLS” should become “tenant separation is enforced at the database-policy layer, not only in app code; every customer-scoped table is protected by organization-bound row policies; privileged paths are tightly limited and audited.”
- Example: “Claude is not used in live detection” should become “the production detection path is non-generative; no transcript text is sent to an external model provider in the live detector; any optional AI-assisted feature is segregated and separately governed.”

## 3. Data residency, processor geography, and subprocessor mapping

This is usually the first serious buyer question: where does our data live, and where can it travel? Supabase states that each project is deployed to one primary region. Its current region list includes Tokyo (`ap-northeast-1`) as a specific region, while the generic APAC region defaults to Southeast Asia (Singapore). That is good news for Kashi: a Japan-residency story is possible - but only if the project is explicitly pinned to the Tokyo region rather than left on a generic APAC default [3]. [3]

The catch is the app-delivery layer. Vercel's DPA says its primary processing facilities are in the United States and that customer data may be transferred and processed anywhere Vercel or its subprocessors operate. The same DPA also makes clear that customers themselves determine what customer data they route through Vercel, and Vercel operates on a shared responsibility model [7][8]. For Kashi this means the cleanest posture is architectural discipline: treat Supabase (or another region-pinned data plane) as the regulated content store, and minimize what sensitive meeting artifacts ever pass through or are persisted by Vercel. [7][8]

- What Kashi can credibly say now: “Customer meeting data is stored in a customer-selected primary database region. For Japan-focused pilots, Kashi can deploy the primary data store in Tokyo.”
- What Kashi should not say yet: “All Kashi processing is Japan-only” or “No customer data ever leaves Japan” unless the team has verified every subprocessor, log path, support path, and optional service against that claim.
- What the buyer packet needs: a one-page data-flow and subprocessor geography map covering primary storage, backups, logs, support access, optional AI services, and disaster-recovery flows.
- What should change in the architecture if strict buyers appear: keep transcripts and evidence snippets out of Vercel logs by design; confine sensitive payloads to the data layer; avoid using app-delivery services as the system of record for regulated content.

## 4. Assurance language: inherited controls versus Kashi's own controls

Supabase states that it is SOC 2 Type 2 compliant and assessed annually. Vercel states that it has a SOC 2 Type 2 attestation and ISO 27001 certification. Those are helpful inherited controls because they reduce buyer anxiety around the hosting stack [4][6][7]. But they do not make Kashi itself SOC 2 compliant or ISO 27001-certified. Supabase's own docs are explicit that running databases is a shared responsibility, and Vercel says the same about its platform [6][7][18]. [4][6][7][18]

This distinction matters because a sophisticated buyer will ask two separate questions: “What assurance do your subprocessors have?” and “What assurance do you, the vendor, have over your product-level controls and operations?” The answer to the first can already be reasonably good. The answer to the second is not yet certification; it is a clear roadmap. [6][13]

- Use vendor assurance as inherited infrastructure assurance, not as a substitute for Kashi's own program.
- State the Kashi path as: internal control set -> documented shared-responsibility matrix -> evidence pack -> pilot-ready questionnaire answers -> later formal attestation or certification when the company reaches the right revenue / customer stage.

- Anchor the internal control set first to ISO/IEC 27001 concepts because ISO 27001 is an ISMS standard for establishing, implementing, maintaining, and continually improving information security management [13].
- For buyer questionnaires, map the product to CSA CCM / CAIQ, and for AI-sensitive buyers prepare an AI-CAIQ lens for any optional model-assisted features [1][2][19].

| What not to claim |
| --- |
| • Do not imply “Supabase is SOC 2, therefore Kashi is SOC 2.” |
| • Do not imply “hosted on certified vendors” equals buyer sign-off. |
| • Do not imply ISO readiness without a documented ISMS scope, policy set, risk register, access-review cadence, incident process, and vendor-management routine. |

## 5. Tenant isolation and access control: RLS is a mechanism, not proof

The current deck says Kashi uses organization-level RLS in a multi-tenant schema and blocks cross-organization leakage at the database layer [K1]. That is directionally strong. Postgres row-security policies restrict, on a per-user basis, which rows can be returned, inserted, updated, or deleted; Supabase documents RLS as a Postgres primitive that can provide defense in depth and be combined with Auth for end-to-end user security from browser to database [5][9]. [5][9]

But a buyer will not stop at “we use RLS.” They will ask whether privileged service roles bypass RLS, how admin workflows are separated, what prevents mistaken policy changes from exposing data, how customer support accesses data, and whether anyone has tested for tenant escape. This is where Kashi needs evidence, not adjectives. [5][9]

- Required control statement: tenant isolation is enforced primarily at the database-policy layer, not only in application code.
- Required evidence: table inventory showing which tables are customer-scoped; RLS policy inventory; service-role inventory; automated negative tests proving one tenant cannot retrieve another tenant's rows; migration review procedure for policy changes.
- Required operational discipline: break-glass access policy, logged support access, and ideally a “no routine production query” stance for engineers unless procedurally approved.
- Future enterprise-hardening step: include cross-tenant isolation testing in external penetration testing and preserve the executive summary in the buyer pack.

## 6. Retention, deletion, legal hold, and admin controls

This is one of Kashi's strongest areas conceptually. The current materials already describe a four-tier model: short-lived raw data, analytics data, review-worthy events, and extended retention only for legal hold or approved case material [K1]. That is already better than most vague “we keep only what we need” language. [K1]

The next step is to make the model operational and buyer-readable. Supabase documents daily backups, PITR options with second-level restore granularity, and explicitly says that when you delete a project it permanently removes all associated data including backups stored in S3. It also documents that PITR changes backup semantics and requires downtime planning for restore events [10]. Those infrastructure facts are useful, but Kashi still needs product-layer deletion semantics on top of them. [10]

- Define deletion by data class: transcript raw text, analytics features, review-worthy events, encrypted evidence snippets, user accounts, and audit records should each have explicit rules.
- Explain customer-admin controls: who can set retention windows, who can place legal hold, who can release hold, and whether defaults can be shortened or only lengthened.

- Document deletion guarantees carefully: app-layer delete, backup persistence, PITR caveats, deletion after contract termination, and irreversible-destruction boundaries should all be stated plainly.
- State the difference between operational retention and legal hold. Buyers care because “we retain for investigations” can quietly mutate into “we keep everything indefinitely” if not bounded.
- Add export semantics: what a customer can export before deletion, in what format, and whether evidence-vault ciphertext remains portable.

## 7. Incident response and auditability

This is currently the biggest missing procedural story. NIST SP 800-61r3 positions incident response as part of ongoing cybersecurity risk management and says the goal is to help organizations prepare for incidents, reduce their number and impact, and improve the effectiveness of detection, response, and recovery [11]. Buyers expect to hear at least a compact version of that operating model from any vendor handling sensitive workplace data. [11]

Kashi does have some auditability language already. The product story references audit trails, and Supabase platform audit logs automatically capture dashboard or API actions by organization members. However, Supabase also documents important limitations: platform audit logs are available only on Team and Enterprise plans, there is currently no dashboard export, no log drain for platform audit logs, and retention depends on the plan [17]. That means Kashi cannot rely on hosted-platform logs alone for enterprise accountability. [17]

- Kashi needs a short incident-response summary for buyers: incident categories, severity levels, internal owner roles, investigation path, evidence preservation, containment and recovery steps, and customer-notification approach.
- Kashi also needs application-layer audit logs that are exportable. Buyer trust will be much higher if drill-downs, break-glass access, retention overrides, deletion events, and evidence-vault actions can all be exported for review or SIEM ingestion.
- Do not overclaim “fully auditable” if the export path is missing. Say “auditable in product today, export/SIEM integration on the enterprise roadmap” unless the export path is already built.
- Run at least one tabletop exercise before pilot sales. The point is not ceremony; it is to discover where your notification timelines, evidence preservation, and support paths are currently ambiguous.

## 8. Key management and the evidence-vault story

This is one of Kashi's most differentiated buyer stories - if it is executed properly. The current v2 concept says WebCrypto generates an RSA-OAEP-2048 keypair in the browser, uses AES-256-GCM for snippet encryption, stores the private key client-side, and leaves the server incapable of decrypting stored evidence [K1]. That is attractive because it creates a strong answer to the fear that an employer is secretly building a readable transcript archive. [K1]

But cryptography only becomes persuasive when key management is specified. NIST SP 800-57 makes this explicit: key management is not just about choosing an algorithm; it covers guidance and best practices for managing cryptographic keying material, protecting different key types, and handling the broader lifecycle issues that arise when cryptography is used [12]. [12]

- Questions the buyer will ask: Who generates the key? Where is it stored? What happens if the employee loses the device? Can the employer escrow or recover it? How does revocation work? What happens on departure from the company? Can encrypted evidence be ported to another provider or retained after account deletion?
- Questions Kashi should answer before pilot: whether recovery phrases are mandatory or optional, whether multiple devices are supported, whether ciphertext can be re-wrapped to a new key, and whether the vendor ever has a support path into plaintext (ideally: no, absent a totally separate design).

- Buyer-facing caution: do not call it “end-to-end encryption” loosely unless the system design really deserves that term and the key lifecycle is finalized. It is safer to say “client-held decryption boundary” or “user-controlled encryption for evidence snippets” until the design is complete.

## 9. Model / data boundary documentation

For security and procurement reviewers, “AI” is a trigger word. Kashi's best current answer is actually quite strong: the production detection path is deterministic and not Claude-backed in live operation [K1]. That alone sharply reduces buyer concern around prompt leakage, training use, and probabilistic black-box claims in the core signal path. [K1]

If Kashi later adds optional AI-assisted features, the vendor terms matter. Anthropic states that for commercial products it acts as a processor on behalf of the customer, processes data under the customer's instructions, and does not use the data shared through commercial products to train models unless the customer opts into the Development Partner Program. Anthropic also says zero data retention applies only to eligible APIs and products using the customer's Commercial organization API key [14][15]. [14][15]

- This means Kashi should separate model usage into explicit lanes: no-model live detection; optional model-assisted internal authoring/testing; optional model-assisted victim-side or analyst-side features only if separately governed.
- For every optional AI lane, document: what text enters the model, whether the model is external, what retention terms apply, whether the customer can disable the lane, and whether a zero-data-retention contract path exists.
- The model boundary memo should also reconcile buyer claims with technical reality. If any future detector uses embeddings, semantic similarity, or other text-derived features, say so precisely and state where that processing runs. Buyers care less about whether there is “AI” in the abstract than whether the vendor is being exact about where it appears.

## 10. Recommended buyer-facing security story for Kashi

Below is the shape of the security story Kashi should tell. This is not a marketing flourish. It is a procurement-grade narrative scaffold that aligns the current project with the kinds of controls buyers are already trained to ask about.

- Kashi is designed around boundary discipline. The default product posture is not universal transcript visibility; it is role-scoped signals, auditability, and constrained access to raw material.
- Primary regulated meeting data is stored in a customer-selected primary region. For Japan-focused customers, Kashi can use a Tokyo primary region in Supabase. Sensitive content should be kept in the region-pinned data plane rather than treated as generic web-app traffic [3].
- Tenant isolation is enforced at the database-policy layer through organization-scoped access controls rather than relying only on UI filtering [5][9].
- The live production detector is non-generative and not Claude-backed. Any optional model-assisted feature is segregated, separately documented, and controlled by explicit customer settings [14][15].
- The product already differentiates raw data, analytics, review-worthy events, and legal-hold material. This should become customer-administered retention policy rather than only deck language [10].
- Kashi can evolve into a stronger confidentiality story with a user-controlled evidence vault in which the server stores ciphertext and cannot decrypt evidence snippets by default [12][K1].

| The exact sentence to use with buyers |
| --- |
| • Kashi does not ask customers to trust an invisible AI box. The production signal path is deterministic, auditable, and role-bounded. Sensitive meeting data is kept in a customer-scoped data layer, tenant isolation is enforced at the database-policy layer, retention is segmented by data class, and any optional |

model-assisted feature is separately governed and documented.

## 11. Sample questionnaire-answer bank

The table below is written to be directly reusable when a partner, pilot sponsor, or procurement reviewer asks standard security questions. It is intentionally conservative: it says what Kashi can credibly say now, while highlighting the caveats that still need to be closed.

| Likely buyer question | Recommended answer framing |
|---|---|
| Where is customer data stored? | Say: “Kashi stores regulated meeting data in a customer-selected primary data region. For Japan-focused deployments we can use a Tokyo primary region in Supabase. We separately document any subprocessor or app-delivery path that may touch customer data.” Caveat: do not imply total geography exclusivity unless every path is verified [3][7][8]. |
| How do you isolate tenants? | Say: “Tenant separation is enforced at the database-policy layer through organization-scoped row-level security, not just at the UI layer. We maintain policy controls and will provide tenant-isolation test evidence for production use.” Caveat: if service-role bypasses exist, explain how they are restricted and audited [5][9]. |
| Do you have SOC 2 / ISO 27001? | Say: “Our infrastructure providers carry strong assurance, including Supabase SOC 2 Type 2 and Vercel SOC 2 Type 2 / ISO 27001 posture. Kashi itself is not yet presenting a standalone attestation; instead we provide a documented control pack, shared-responsibility matrix, and evidence package suitable for pilot review.” Caveat: never imply inherited compliance equals Kashi certification [4][6][7][13]. |
| Can we configure retention and deletion? | Say: “Kashi is designed around distinct data classes - raw inputs, analytics, review-worthy events, and legal-hold material - with different retention logic. The customer-facing admin layer should expose those controls explicitly. We also document backup and recovery semantics, including where deletion is immediate and where backup windows still apply.” Caveat: if admin controls are not yet shipped, call them roadmap rather than present capability [10]. |
| What happens in a security incident? | Say: “Kashi maintains auditable administrative actions and is formalizing an incident response process that covers severity classification, containment, evidence preservation, customer notification, and recovery.” Caveat: if notification timelines are not yet fixed, do not improvise promises in a sales call; commit to a written incident-response summary instead [11][17]. |
| Does any model provider train on our data? | Say: “The production detection path does not use an external LLM. If any optional model-assisted capability is enabled in the future, it will be separately governed. Anthropic states that commercial-product data is not used for training unless the customer opts into the Development Partner Program.” Caveat: document whether any feature is eligible for zero data retention before you mention ZDR [14][15]. |
| Who can see raw content? | Say: “The product is designed around role-bounded visibility and auditability rather than universal transcript browsing. Any access to retained raw material should require a defined purpose, appropriate role, and auditable trail.” Caveat: be precise about whether support engineers or admins have any exception path today [K1][17]. |
| How is encryption handled? | Say: “Data-in-transit and data-at-rest protections exist at the infrastructure layer. For especially sensitive evidence, Kashi is building a user-controlled encryption boundary where evidence snippets are encrypted client-side and the server cannot decrypt them by default.” Caveat: do not oversell this until key recovery, rotation, and loss-handling are documented [7][12][K1]. |

## 12. Immediate deliverables: the Security & Assurance Pack

This is the most pragmatic next step. Kashi does not need a perfect enterprise program before it can talk to serious buyers. It does need a coherent packet that converts the current architecture into procurement-ready evidence. The first version should be concise enough to read quickly and rigorous enough to survive follow-up questions.

- 1. Executive security overview (2 pages): architecture boundaries, live detector path, major controls, customer-admin model, key open risks.
- 2. Shared-responsibility matrix: what Supabase controls, what Vercel controls, what Anthropic would control if optional model-assisted features are enabled, and what Kashi itself controls [6][7][14].
- 3. Data-flow and subprocessor map: storage region, backup region, log path, support access, AI path, exports, deletion path.
- 4. Retention and deletion policy: one page with data classes, default periods, hold logic, backup caveats, deletion triggers.
- 5. Incident response summary: severity levels, escalation roles, evidence handling, communication path, customer notification commitment.
- 6. Tenant-isolation evidence note: RLS design summary, privileged-path controls, negative-test summary, future pen-test scope.
- 7. Model/data boundary memo: exactly what does and does not touch external models.
- 8. CAIQ/SIG-lite draft answers: even if incomplete, start the workbook now so the team sees every question that will arrive anyway [1][2].

## 13. 30 / 60 / 90 day execution plan

| Window | Primary objective | Concrete deliverables | Owner lens |
|---|---|---|---|
| **30d** | Make the security story coherent | Region/subprocessor map; shared-responsibility matrix; model/data boundary memo; do-not-say list for sales; draft incident response summary. | Founders + eng |
| **60d** | Make the controls evidenced | Tenant-isolation test pack; exportable app audit log design; retention/admin-control spec; key-management lifecycle decision note; first CAIQ/SIG-lite draft. | Eng + product + legal ops |
| **90d** | Make the pilot package buyer-ready | Security & Assurance Pack v1; tabletop exercise notes; security FAQ for partners; subprocessor page; customer-ready DPA/security appendix draft. | Cross-functional |

## 14. Final judgment

Kashi does not need to become a finished enterprise security program overnight. It does need to stop treating buyer trust as something that will automatically fall out of technical correctness. Procurement and security reviewers buy control clarity. Right now Kashi has enough raw material to look serious: region-pinnable storage, database-policy isolation, a bounded production detection path, auditability intent, retention segmentation, and a compelling future confidentiality model via user-controlled evidence encryption. That is a strong starting position. [3][4][5][7][10][12][14]

The near-term task is therefore not mainly to add more features. It is to package the existing architecture into a disciplined assurance story and to close the few genuine blockers: incident response, key lifecycle, exportable audit evidence, deletion semantics, and explicit geography/model boundaries. If Kashi does that, the project stops looking like a clever governance demo and starts looking like a pilotable enterprise system.

## References

[1] Cloud Security Alliance, “Cloud Controls Matrix and CAIQ v4.1,” released Jan. 27, 2026. Notes: 207 controls across 17 domains; CAIQ provides yes/no assessment questions. https://cloudsecurityalliance.org/artifacts/cloud-controls-matrix-v4-1

[2] Shared Assessments, “What is a SIG LITE questionnaire?” and SIG overview pages. Notes: SIG Lite is a broad, high-level due-diligence questionnaire that can precede deeper review. https://sharedassessments.org/about-sig/

[3] Supabase Docs, “Available regions.” Notes: each project has one primary region; generic APAC defaults to Singapore; specific regions include Tokyo (`ap-northeast-1`). https://supabase.com/docs/guides/platform/regions

[4] Supabase Docs, “SOC 2 Compliance and Supabase.” Notes: Supabase states it is SOC 2 Type 2 compliant and assessed annually. https://supabase.com/docs/guides/security/soc-2-compliance

[5] Supabase Docs, “Row Level Security.” Notes: RLS is a Postgres primitive that can provide defense in depth and be combined with Supabase Auth for browser-to-database security. https://supabase.com/docs/guides/database/postgres/row-level-security

[6] Supabase Docs, “Shared Responsibility Model.” Notes: running databases is a shared responsibility between customer and Supabase. https://supabase.com/docs/guides/deployment/shared-responsibility-model

[7] Vercel Security / DPA pages. Notes: Vercel states SOC 2 Type 2 attestation and ISO 27001 certification; DPA states primary processing facilities are in the United States, customer controls what data enters the service, and customers can delete/suppress customer data on demand. https://vercel.com/security ; https://vercel.com/legal/dpa

[8] Vercel Docs / DPA. Notes: Vercel frames security through a shared-responsibility model. https://vercel.com/docs/security/shared-responsibility ; https://vercel.com/legal/dpa

[9] PostgreSQL Documentation, “Row Security Policies.” Notes: row-security policies restrict, on a per-user basis, which rows can be returned, inserted, updated, or deleted. https://www.postgresql.org/docs/current/ddl-rowsecurity.html

[10] Supabase Docs, “Database Backups.” Notes: daily backups, PITR options, restore workflow, and project deletion removing associated data including backups stored in S3. https://supabase.com/docs/guides/platform/backups

[11] NIST SP 800-61 Rev. 3, “Incident Response Recommendations and Considerations for Cybersecurity Risk Management.” Notes: incident response should be integrated into cyber risk management to prepare for incidents, reduce number/impact, and improve detection/response/recovery. https://csrc.nist.gov/pubs/sp/800/61/r3/final

[12] NIST SP 800-57 Part 1 Rev. 5, “Recommendation for Key Management: Part 1 - General.” Notes: key management covers broader lifecycle guidance and protection requirements for cryptographic keying material. https://csrc.nist.gov/pubs/sp/800/57/pt1/r5/final

[13] ISO, “ISO/IEC 27001:2022 - Information security management systems.” Notes: ISO 27001 is the best-known ISMS standard and defines requirements for establishing, implementing, maintaining, and continually improving an ISMS. https://www.iso.org/standard/27001

[14] Anthropic Privacy Center, “Does Anthropic act as a Data Processor or Controller?” Notes: for commercial products Anthropic says it acts as a processor on behalf of the customer and does not train on customer data unless the customer opts into the Development Partner Program. https://privacy.claude.com/en/articles/9267385-does-anthropic-act-as-a-data-processor-or-controller

[15] Anthropic Privacy Center, “I have a zero data retention agreement with Anthropic. What products does it apply to?” Notes: ZDR applies only to eligible APIs and products using the customer’s Commercial organization API key. https://privacy.claude.com/en/articles/8956058-i-have-a-zero-data-retention-agreement-with-anthropic-what-products-does-it-apply-to

[16] Cloud Security Alliance, “AI-CAIQ” and related AI Controls Matrix materials. Notes: AI-CAIQ is intended to help organizations identify gaps, mitigate AI-related risks, and demonstrate accountability in line with CSA AI controls. https://cloudsecurityalliance.org/artifacts/ai-consensus-assessments-initiative-questionnaire-ai-caiq

[17] Supabase Docs, “Platform Audit Logs.” Notes: platform audit logs capture dashboard/API actions but currently have no dashboard export, no platform audit-log drain, and retention depends on plan. https://supabase.com/docs/guides/security/platform-audit-logs

[18] Vercel Docs, “Vercel security overview” and shared responsibility materials. Notes: Vercel describes shared responsibility, encryption, compliance measures, firewalling, and audit logs. https://vercel.com/docs/security ; https://vercel.com/docs/security/shared-responsibility

[19] Cloud Security Alliance, “AI Controls Matrix / STAR for AI” materials. Notes: AICM and AI-CAIQ provide an AI-focused assurance layer that maps to broader standards and can be used for third-party AI assessments. https://cloudsecurityalliance.org/artifacts/ai-controls-matrix ; https://cloudsecurityalliance.org/star/ai

[K1] Internal project source: “Kashi - Progress & Project Overview (2026-04-21)” PDF provided by the user. Used for current-architecture claims such as Supabase + RLS, multi-tenant schema, four-tier retention, auditability language, no-live-LLM production detection, and the v2 evidence-vault concept.