Kindred Grove — Project Status

A cinematic Shopify storefront, a reviewed engineering foundation, and a staged plan for large-scale commerce.

Updated: Oct 8, 2026

1. Executive overview

Kindred Grove is a premium pantry storefront built on Shopify with Liquid, CSS and vanilla JavaScript Web Components. The design combines a cinematic opening, warm cream and terracotta colors, arched product imagery and clear shopping interactions. The project combines a distinctive frontend and the engineering practices needed to evolve it: configuration ownership, privacy controls, defensive request handling, automated checks, independent review and reproducible release evidence.

The connected custom domain now serves the published Shopify storefront with seven native content pages: Our Story, Recipes, FAQ, Contact, Shipping & Returns, Find Your Pantry (quiz) and Wholesale. Shopify's development-store policy still requires the shared visitor password. The current store cannot process real transactions or be converted/transferred as a commercial client store; commercial eligibility is a separate launch dependency. The complete source and refreshed public documentation have been published on the project's review branch in pull request #8. Main still requires a separate approving review. A GitHub source update does not publish a Shopify theme.

The future program targets 10,000, 100,000 and ultimately 1,000,000 or more simultaneous visitors, with a proposed 99% availability objective. This document explains the architecture, implementation stages, ownership and validation needed to reach those goals. Capacity and operational milestones are planned work; the verified storefront and source-publication results are listed separately below.

Project source and public documentation are available through the review branch. Start with the documentation index.

2. Frontend experience and customer journey

The homepage is built around a film-led introduction. As visitors scroll, the left-side eyebrow, heading, emphasized wording and supporting sentence change with the film's chapters. The transition works in both directions, so scrolling back restores the corresponding scene and copy. Visitors can pause motion, and a reduced-motion preference retains a usable static presentation.

The header hides during downward travel to leave more room for the content, then returns when the visitor scrolls upward. Navigation also returns when focus, a menu or a dialog requires it. Its layout and hero offsets are coordinated to avoid an empty band or obscured controls. The

mobile layout has readable typography and a usable navigation fallback when JavaScript is disabled.

Product and collection pages use native Shopify catalog records. The interface includes product imagery, variants where available, product cards, quick view, pantry interactions, a cart drawer and a full cart page. The cart supports actual add, quantity change and removal. Prices come from Shopify catalog data. Native contact, newsletter, wholesale and checkout controls are present; the current development-store policy limits real transactions. No real order or customer-information submission was used during verification.

The optional pantry quiz asks five questions, computes a result and links to a real collection. Its answers exist only while that page/component is active; reload resets the experience. Canonical Find Your Pantry (quiz) and Wholesale pages are now created and verified alongside Our Story, Recipes, FAQ, Contact and Shipping & Returns. The existing Contact page identity was preserved, and navigation connects the seven pages. English and Arabic locale source is present; publishing languages, currency availability and complete RTL review remain separate merchant/QA tasks.

3. Architecture: frontend, backend and ownership

Shopify operates the storefront rendering platform, catalog, inventory, CDN, native cart and hosted checkout. The repository owns the Liquid composition, styles, browser components, configuration, tests and engineering documentation. There is no separately operated Kindred Grove database or active custom commerce server to resize.

Liquid renders the initial page and native links. Web Components enhance shopping and navigation while preserving appropriate native/static fallbacks. Public assets use Shopify’s delivery layer. Cart, consent, account and checkout state remain visitor-specific and must never enter a shared public cache. Merchant content stays in Shopify settings, locales and structured catalog/metaobject data.

The former wholesale draft-order Worker is retired in source: it returns HTTP 410 without reading submitted data, calling a service or creating an Admin API side effect. The legacy seed tool validates offline reference data and rejects remote apply operations. This reduces the exposed backend surface and avoids an extra service on the buying path.

A separate checkout-extension folder is retained as an unsupported historical implementation. Its React API/manifest combination needs migration, and its certification/shipping/returns claims are not substantiated. It is excluded from the delivered checkout feature set and from the root build evidence. A future extension requires an authorized app project, supported API, verified entitlements, approved claims and its own validation.

4. Codebase structure and maintainability

Theme layout and templates define document and route composition. Sections and blocks provide merchant-editable surfaces; snippets hold reusable markup, configuration transport and

structured-data helpers. Assets contain the unbundled components and styles. Config and locale files own settings and maintained interface text. Scripts contain validated tooling, CI helpers and the capacity model; tests contain genuine automated regression coverage.

Changeable values have named owners. Browser routes, messages, limits and timing policy are rendered through validated configuration. Test targets and runner defaults have a central Node configuration boundary. Tool versions and scanner checksums are pinned in a maintained manifest. Secrets are supplied through approved local/platform storage and are excluded from source, reports and client documents.

Components cancel superseded optional reads, clean up lifecycle work and validate responses before rendering. This structure lets another developer locate a setting, understand its consumers and extend a component without inventing a second configuration default. The architecture and configuration map explain these boundaries.

5. Layered security controls

The implemented theme uses same-origin request validation, accepted URL schemes/hosts, contextual escaping and non-executable configuration transport. Responses have configured size and time bounds; redirects and malformed data are rejected where the shared transport requires it. API content does not become executable code merely because the server returns a legacy MIME type.

Cart mutations are serialized. If a response fails after Shopify may have accepted a write, the theme does not blindly repeat that write and create duplicate additions. It reconciles by reading actual cart state and communicates an uncertain outcome. Tests cover hostile inputs, stale results, failed reads and ambiguous writes, alongside real cart behavior.

The security process includes Gitleaks source/history/staged scans, Semgrep Community Edition, the installed edit hook, configuration regressions, Theme Check and independent review. Password fixtures bind navigation and effective form submission to the intended storefront before entering a credential; password-entry recordings are disabled. Private exports, credentials, recovery records and local connection settings are not part of the public Git tree.

Future operational hardening includes verified least-privilege admin/app roles, MFA and recovery ownership, vendor access reviews, dependency response, incident exercises and server-side controls for any new backend. The current meta CSP has platform compatibility constraints, including inline script allowances; it is one defense layer, not a strict response-header security claim. The security register records scope and responsibilities.

6. Privacy, consent and data lifecycle

Shopify’s native banner remains the visitor-facing consent interface. Optional theme processing requires both the visitor’s explicit purpose consent and the matching Shopify allowed-processing

result. Unknown or unavailable permission denies optional processing. Withdrawal clears theme-owned optional keys where storage is available.

Feature-flag visitor identifiers/assignments require analytics consent. Recently viewed product handles require preferences consent and are bounded to 12 records and 30 days. Quiz answers and announcement dismissal remain in page memory. Optional theme telemetry loaders and signup forwarding have been removed. The theme does not automatically grant consent or edit Shopify's consent cookies.

Actual browser checks exercised unknown consent, accept, partial preferences, withdrawal and reload. Separate tests emulated Global Privacy Control and Do Not Track before page load and confirmed theme analytics/marketing/sharing stayed denied. Native banner settings were saved and reloaded across all 299 listed regions. These checks concern named theme controls; Shopify Network Intelligence remains enabled, and app/pixel/hosted-account processing needs its own inventory.

Before collecting real customer information, the merchant needs a truthful privacy notice, controller/contact identity, vendor and transfer register, retention schedule, request intake and deletion/exception process. A rights-request exercise should trace a test case across Shopify and actual vendors, recording the response clock, actions and closure. The privacy document maps current data and future operating procedures.

7. US legal and commercial preparation

The US requirements register covers state privacy applicability, sale/sharing opt-outs and universal signals, sensitive/consumer-health information, truthful notices, children's privacy, commercial email, advertising/reviews, shipment promises, food labeling, accessibility, tax and breach-response triggers. The actual legal entity, served states, revenue/data volumes, product facts and processing purposes determine which duties apply.

California CCPA/CPRA scope has multiple tests; its adjusted revenue threshold alone does not decide coverage. California and Oregon signal duties are assessed separately from the theme's GPC behavior. Dietary choices are not automatically medical data, but joining quiz answers to identity, purchases or inferred health conditions would need another assessment, including Washington consumer-health requirements.

A commercial email flow needs accurate identity, a real business address and an operated unsubscribe/suppression process. Delivery claims need an operational basis and a delay/cancellation/refund process. Reviews, origin stories, certifications and environmental statements need actual evidence; unverified marketing copy cannot supply that evidence. Real food products require product-specific label/allergen/nutrition and supplier review, including any exemptions and import or state obligations.

The merchant owns business facts and operating decisions. The public requirements matrix now covers US, Canadian and EU preparation. The developer implements approved controls

and provides test evidence. The legal requirements register links each topic to official sources, owners, evidence and review gates, including CPPA guidance and FTC business guidance.

8. EU legal and market preparation

EU preparation includes GDPR territorial scope, lawful purposes, transparency and rights; national ePrivacy/device-storage rules; processor contracts and international transfers; consumer contracts, prices, withdrawal and claims; food information; accessibility; VAT/imports; and relevant media/AI-transparency triggers. A blanket EU selling switch is not the implementation plan: each selected country, product and vendor must be assessed.

Before commercial launch, provide accurate trader identity/contact, total charges, taxes/shipping, payment and delivery terms, complaints handling and product-specific returns/withdrawal information. Food exceptions must be evaluated from the actual product rather than a blanket “food cannot be returned” policy. Compare-at prices do not prove the required historical reference price; retain evidence for reduction announcements and applicable unit pricing.

Online withdrawal-function measures apply from `Jun 19, 2026` through national implementation for covered contracts where a withdrawal right exists. The future design must provide an easily accessible function during the withdrawal period, a confirmation step and a durable acknowledgement carrying the submission content and date/time. This requirement is not restricted to financial services. No live commerce withdrawal feature is represented as implemented here. Directive 2023/2673, including Article 11a.

Green-transition consumer rules reached their application milestone on `Sep 27, 2026`. That milestone has passed; review environmental claims and sustainability labels against the applicable national measures before EU-facing commercial use. Mandatory food information for distance selling, emphasized allergens, quantity, nutrition, storage/use and responsible-operator/origin details must come from verified product records. Relevant language and accessibility requirements are part of the market gate.

The observed Shopify markets were United States and Canada; an EU commercial rollout has not been established. The roadmap includes merchant-specific legal/tax/product decisions and appropriate specialist review when interpretation is needed. GDPR text, European Commission consumer-rights guidance.

9. Storefront access, content and media

Shopify requires password protection on the current development store. Sharing the separate visitor password gives invited visitors access to the storefront while administrative passwords, app credentials and recovery material remain private. The password screen is generated by Shopify before the theme loads; this is a platform access restriction, not a missing frontend page or a DNS failure.

The published theme now renders native cart, contact, newsletter, wholesale and checkout controls. Theme settings cannot remove the platform-required password or enable real transactions on this store. Commercial activation requires a production-eligible store, approved merchant/product/payment settings and operated customer-data, fulfillment and support processes. End-to-end inbox delivery and payment processing remain unverified.

Media provenance is a maintained launch input. The repository contains local export/composition relationships for its images and film, but creator, source and usage-rights evidence remains unverified. The code license does not grant third-party image, music, brand or certification rights. Commercial reuse needs rights evidence and appropriate attribution/release decisions. Storefront access, media register.

10. Accessibility and inclusive interaction

WCAG 2.2 AA is the engineering target. The earlier September accessibility checks included keyboard/focus/Escape behavior, a usable direction-aware header, reduced-motion handling, mobile no-JavaScript navigation and product reflow at 320, 370, 371, 390 and 1440 pixels. Four base routes passed axe, and active Quiz/Wholesale template scans passed with zero violations. The Quiz progressbar had a visible, meaningful accessible name.

Manual screen-reader testing, actual 200%/400% browser zoom, text spacing, forced colors, complete Arabic RTL journeys and real-device mobile assistive technology remain planned checks. Earlier enlarged-root-font experiments are not counted as browser-zoom conformance. Automated scan exclusions for platform/app markup are documented and require separate review of the whole visitor experience.

Accessibility responsibilities are assessed against the merchant's business and markets, including relevant EU accessibility and US ADA guidance. The accessibility plan gives evidence and the maintained manual checklist.

11. Performance and media optimization plan

The theme renders without a frontend application bundler or runtime framework dependency. Responsive image helpers, dimensions, lazy loading and priority hints support efficient delivery, but the actual waterfall and rendered page determine the result. Optional requests are bounded/cancellable, and reduced motion avoids the film load.

The journey video is approximately 5.5 MiB on disk. Current playback fetches the complete asset as a blob for predictable seeking. The next performance phase will compare hosted range playback with whole-asset loading on real desktop/mobile profiles, measuring transferred bytes, first usable frame, LCP, CLS, long tasks and memory under cold and warm caches. One million complete downloads of the current file would be about 5.24 TiB before protocol overhead; this is planning arithmetic, not measured CDN egress or a bill.

Maintained lab budgets include LCP at most 2.5 seconds and CLS at most 0.10, plus asset/request budgets. A lab result cannot establish field INP or event-scale capacity. Each

optimization must preserve visual quality, pause/reduced-motion behavior, accessibility and the cart. App embeds and third-party scripts must be included in the measurement. Performance plan.

12. Future scale: 10k to 1M+ simultaneous visitors

The scaling design starts with a workload, not a server count. Concurrent visitors, requests per second and in-flight requests are different measurements. The maintained model currently assumes one page view every 10 seconds per active visitor, eight static requests per view, a 95% assumed static cache-hit fraction, 0.05 cart mutations and 0.01 checkout starts per view, with a three-times burst factor. These are editable scenario inputs, not observed customer behavior.

Under those inputs, 10,000 active visitors project 1,000 page navigations, 8,000 static requests, 400 static misses, 50 cart writes and 10 checkout starts per second. At 100,000 visitors the corresponding rates are 10,000, 80,000, 4,000, 500 and 100. At 1,000,000 visitors they are 100,000, 800,000, 40,000, 5,000 and 1,000 per second. The modeled burst multiplies each rate by three. A static miss is an assumed sensitivity value; it does not reveal Shopify's internal origin load.

Shopify owns the capacity of Liquid rendering, CDN, native cart and hosted checkout. Theme engineering can reduce bytes, expensive markup/app work, redundant reads and retry pressure, but cannot resize Shopify's internal workers. API documentation describing one API's limits is not a capacity guarantee for another API or the entire purchase journey.

At 10k, qualify the real route/action mix, cold/warm delivery, devices, markets and baseline latency/errors with Shopify's store-specific guidance. At 100k, validate bursts, soak, regional mix, dependency failure and any queue/back-pressure behavior. At 1M+, require a platform-approved traffic/capacity plan, representative global workload, controlled ramp/abort/recovery exercise and staffed event operations. Progress beyond one million follows the same measured bottleneck and provider-approval process.

The local loopback harness verifies model behavior and bounded test requests without contacting Shopify. Its 1,000-request successful run is labeled LOCAL_MOCK_NOT_SHOPIFY. The complete stage gates, formulas and source ownership are in scalability, alongside Shopify API limits.

13. Conditional backend expansion

Retain the native Shopify architecture while it meets the product and measured operating requirements. A custom service should solve a specific capability or measured bottleneck. Headless rendering is an option when experience and operational evidence justify it; it introduces responsibility for rendering, cache keys, sessions, API traffic and deployments, and does not remove checkout/provider limits.

A justified asynchronous service should sit outside checkout where possible. Its design needs authenticated/authorized callers, least-privilege credentials, bounded input, per-principal and global admission limits, durable idempotency, bounded queues, retry/dead-letter policy, truthful rejection when full, privacy retention, monitoring and a named operator. Autoscaling must consider backlog age, database connections and downstream quotas, not only CPU.

Regional storage, database topology, failover and backup decisions follow actual data classification, recovery targets and workload. A new database, cache, queue or hosting provider is not automatically necessary because a projected visitor number is large. Each additional dependency needs a cost, security, legal and failure-mode review before activation.

14. Proposed 99% availability objective

The operating proposal is 99% good eligible observations over a rolling 30-day window, measured separately for browsing, cart and checkout navigation. Browsing must reach real content in the intended market; an HTTP 200 password, challenge or error page is not success. A cart check must establish the expected mutation/read/cleanup outcome in an isolated test context. Checkout navigation measures reaching a genuine checkout without entering customer/payment data; it does not measure successful payment authorization.

Missing samples are unknown. Keep an end-to-end result that includes dependency failures, with separate diagnostic attribution to theme, Shopify or another provider. Record sampling interval, coverage, regions, revision and latency policy; a laptop test does not provide continuous monitoring.

A continuous-time 99% objective over 30 days permits 432 minutes, or 7 hours 12 minutes, of unavailability. A request-count error budget is 1% of eligible observations and is a different denominator. Proposed alerts should use burn-rate windows and coverage checks, with named recipients and response hours. Repeated budget burn should pause nonessential releases and prioritize repair or tested restoration.

The monitoring implementation, alert delivery and sustained measurements are future milestones. Select any independent monitor only after confirming free entitlement, acceptable traffic, privacy and maintenance. See operations and Google SRE guidance.

15. Recovery, incidents and continuity

The proposed theme recovery objective is restoration within 30 minutes, with the last reviewed artifact as the theme recovery point. This needs a timed rehearsal on an unpublished target. Theme snapshots do not restore orders, customers, payment records or catalog state; those require their own Shopify/merchant recovery arrangements.

Before a write, capture native target/role, remote snapshot, reviewed file hashes, intended changes and current stage. After upload, inspect structured errors, download the result and compare exact hashes. If power or transport fails after a mutation begins, inspect the real

remote state before deciding whether anything needs replaying. Preserve unexplained drift and reconcile it locally.

Incident handling needs a primary/backup owner, severity, affected paths, timeline, containment, provider escalation, customer communication decision and recovery proof. Personal-data incidents additionally require an awareness timestamp, affected data/people, risk and jurisdiction assessment, and evaluation of applicable notification deadlines. Preserve only necessary evidence in private records. No regulator/customer messaging is automated by the theme.

16. Testing and delivered evidence

The earlier September development artifact met all six Phase 2 criteria after independent scoped review. Local security/configuration tests passed 95/95. The integrated browser suite passed 41 cases, failed none, skipped five evidence-based cases and had no flaky results. Separate alternate Quiz/Wholesale functional tests and active axe tests each passed 2/2. All 150 local, frozen and downloaded development-theme files matched exact hashes.

The earlier publication passed 95 security/configuration tests and scanned 39 source/tooling files. The current publication passed 102/102 security, configuration and public-document tests; Theme Check reported zero errors and two existing orphan-snippet warnings. Semgrep Community Edition applied 23 applicable rules across 40 tracked source/tooling files with zero findings or errors. Source/history secret scans and the installed edit hook passed; all 266 public files matched the scanned snapshot. Local theme packaging passed, independent fresh-context review returned PASS, and all 154 shipping-theme hashes were unchanged by this documentation update. The theme is unbundled; standalone type-check, JS/CSS lint and bundler-build scripts are absent. The separate legacy extension is not counted as built or tested.

The earlier five browser skips were a single-variant catalog item, absent canonical Quiz/Wholesale pages, no test article and a Shopify-hosted account redirect. Subsequent verification created the canonical pages and recorded 55 browser passes plus a separate native-search/mobile-accessibility pass; three catalog-dependent cases remained skipped. Seven pages were exercised at desktop and mobile widths, including 1440, 390 and 320 pixels. The last native theme release matched 154 local, frozen and downloaded shipping files. Real checks covered cart 0→1→2→0, native consent changes, signal emulation, header/video, keyboard/focus, reduced motion, no-JavaScript navigation and narrow reflow. The public testing guide and defect log preserve that scope.

17. GitHub publication and release governance

The source update includes the reviewed frontend, native Liquid/backend boundaries, retired Worker, offline tooling, meaningful CI tests, configuration and public documentation. All three tracked READMEs were refreshed. Dedicated backend-contract, project-requirements, storefront-access and root security-reporting documents were added; the architecture, privacy,

legal, scalability, operations, roadmap, release and historical decision records were reconciled. Credentials, private project records, client exports and investigation artifacts were excluded.

The candidate workflow source uses pinned tooling, separates offline checks from manual credentialed preview tests and blocks theme writes until portable drift/parity automation is ready. GitHub main requires one approving review. The previous default-branch production workflow can write live and its environment has no reviewer rules; the replacement must be reviewed and merged before it is relied on as the default policy. No protection was weakened. The storefront had already been published through its native release process; this documentation refresh changed no shipping-theme file and performed no additional Shopify deployment.

The release guide explains staged publication, secret checks, immutable artifacts, remote drift, interrupted operations and rollback. The earlier September publication also passed both hosted checks on source commit 078aae7. GitHub's Security and config regression and Liquid linting checks both passed on the current published source commit 861efa33b2aba41ed18122a23e442359dc468b28 in PR #8. Gitleaks remains disabled by inactivity as a hosted workflow, while the local source/history/staged scans passed. Required review remains outstanding; no hosted browser or deployment run was invoked.

18. Ten-phase delivery roadmap

Phase 1 — Connected subdomain and service boundary: completed for the existing live theme, including native domain/TLS and isolated product/cart/checkout-navigation evidence without an order.

Phase 2 — Published, functional Shopify storefront: completed within the reviewed theme scope, with seven native content pages and shared-password access. Native commerce remains subject to development-store restrictions; catalog/content provenance and end-to-end delivery require merchant evidence.

Phase 3 — US/Canada/EU applicability and governance: use the expanded requirements register to resolve actual merchant, product, vendor, privacy, rights, food, tax, claims and market facts; implement and exercise each triggered operating control.

Phase 4 — Measured performance: establish representative desktop/mobile cold/warm baselines; compare video delivery; optimize measured bottlenecks with visual/accessibility/cart regression checks.

Phase 5 — Capacity qualification: move from configurable projections and local workload simulation to platform-approved 10k, 100k and 1M+ validation stages, each with workload, latency/error, provider and abort/recovery evidence.

Phase 6 — Availability operation: implement privacy-safe measurement, missing-data reporting, named on-call ownership, alert delivery and error-budget policy for the 99% objective.

Phase 7 — Recovery rehearsal: time an unpublished-theme restoration, verify exact hashes and actual shopping behavior, and test interruption-safe continuation against the proposed recovery targets.

Phase 8 — Reproducible delivery: finish portable reviewed-artifact automation, verify hosted checks and required statuses, and prove target/drift/parity protections before any write workflow is enabled.

Phase 9 — Independent integrated audit: freeze one candidate, review source/data/configuration, run relevant tests and manual checks, reconcile all documents and resolve release-blocking findings.

Phase 10 — Controlled release and maintained handoff: approve a concrete release only after its gates, owners, rollback and operating dependencies are ready. Maintain source, client documentation, credentials recovery and incident learning as the project evolves.

The current source/documentation handoff prepares these future milestones without claiming their implementation is complete. Full build plan, roadmap.

19. Cost, ownership and practical launch decisions

This delivery used existing platform access and free/local tools. No paid monitoring, AI review, media generation, new hosting, plan upgrade or external load test was activated. Standard GitHub-hosted runners for public repositories have a documented free boundary; optional integrations and future retained infrastructure need separate entitlement/cost review.

Before commercial activation, name the merchant/controller, product/label owner, privacy-request owner, incident operator, release approver and backup contact. Supply real business identity and support details; confirm target markets, fulfillment, taxes, payments, vendors, media rights and policy evidence. Assign dates to future milestones only when dependencies and resources are known.

For clients, the value of this foundation is a polished storefront combined with visible engineering decisions, reproducible evidence and an actionable growth path. Developers can start from the source and configuration map; merchants can use the merchant guide; reviewers can trace each future promise to its acceptance gate and accountable owner.

20. Backend contracts and integration admission

Shopify owns rendering, catalog, inventory, cart, forms, privacy preferences and hosted checkout. The theme owns accessible composition, safe browser transport and honest errors. Reconcile authoritative state before repeating a cart mutation with an uncertain outcome. Exclude carts, account data, passwords, consent and checkout from shared caches. The backend service contracts map responsibility and evidence.

Native contact, newsletter and wholesale controls need a controlled delivery test and a named inbox/support owner before customer use. Hosted accounts, payment authorization, tax and fulfillment need separate real-flow qualification on an eligible production setup. Adding a custom database or API is conditional on an approved capability or measured gap.

Future integrations need per-operation authorization, minimal data/retention, bounded concurrency/queues, durable idempotency and safe retries. Webhooks need original-byte signature verification, replay protection, durable acceptance and dead-letter recovery. Operators own privacy-safe monitoring, secrets, migrations, backups, escalation and tested rollback. Checkout dependencies need justification and platform approval; paid activation needs prior cost authorization.

21. Canadian market and payment preparation

Canada is an observed Shopify market. Assess PIPEDA and provincial privacy laws against actual business and processing facts; assign a privacy contact and purpose/consent, access, retention, safeguards and breach procedures. Commercial messages need a CASL applicability and consent/identification/unsubscribe assessment, evidence and an operated suppression process.

Verify each Canadian food SKU's label, ingredients/allergens, bilingual, quantity, origin, storage and supplier/import requirements. Assess provincial consumer/tax duties against sales/fulfillment facts. Keep cards in approved hosted checkout; platform certification does not discharge merchant/acquirer or app PCI duties. Sources: OPC/PIPEDA, Government of Canada CASL, CFIA industry labeling and PCI merchant resources.

22. Qualification, ownership and continued maintenance

The project requirements matrix assigns responsible roles and exit evidence across storefront, backend, business identity, products, commerce eligibility, support, privacy, security, accessibility, performance, scale, availability, recovery, delivery and costs. Name primary/backup owners before launch; source code does not establish merchant identity, approved claims, working inboxes or response coverage.

The 10k, 100k and 1M+ stages need representative workloads, ramp/burst/soak and dependency-failure evidence, provider guidance and abort/recovery criteria. The 99% objective needs owned per-path measurements, coverage, alert delivery and a recorded rolling window. Local workload exercises and browser sessions do not close these operating gates.

Proposed maintenance: review advisories, tests, app/privacy changes, drift, artifact identity and rollback before releases; permissions, observation gaps and defects monthly; market/legal applicability, vendors, retention and product claims quarterly. Rehearse restoration before major campaigns and after architecture changes. Update recovery checkpoints around remote writes; inspect remote state after interruption. Update the client document in place and verify a temporary PDF before replacing its canonical export.