

The enterprise evaluation pack.

Business. Engineering. Security. Operations. Procurement.

1. One review. Every team at the table.
2. A partnership should have something worth proving.
3. Know which interface you are evaluating.
4. A contract an engineer can inspect.
5. Authority follows the actor and the operation.
6. Make acceptance reproducible.
7. Define the day after integration.
8. Separate controls from assurance.
9. Define every data boundary.
10. Agree on responsibility before activation.
11. A launch decision needs evidence.

Source review · no release certification or access grant.

Missing operational, independent and contractual evidence remains explicit.

One review. Every team at the table.

Give every team a reason to engage and a clear decision to make. Start with the opportunity, then explore the product, integration and operating model.

peyeli.com/en/docs/enterprise-review

Start with the outcome your team wants to build.

A corporation can evaluate distribution, merchant software, record exchange, accounting, institutional services or a financial rail. These relationships do not require the same technical access or financial authority. Start with the outcome and the records exchanged; only then select an integration and its approval gates.

Team	Review	Decision output
Business / strategy	Use case, product boundaries, customer journey and capability status.	Defined relationship, value hypothesis and exclusions.
Engineering / architecture	API surface, payment contract, permissions, failure recovery and acceptance matrix.	Approved interface set and integration design.
Security / privacy	Source controls, data lifecycle, assurance evidence and access grant.	Evidence-backed assessment and unresolved risks.
Finance / operations	Reconciliation, exceptions, support ownership and recovery.	Settlement responsibilities and tested operating procedure.
Legal / procurement	Contracting entities, commercial governance, data terms and exit.	Reviewed agreement; no implied program approval.

Move through four recorded decisions.

Discovery records the use case and exclusions. Design review identifies exact interfaces, data and responsibilities. A bounded sandbox proves the acceptance matrix. A production decision requires environment-specific technical, operational, commercial and financial approvals. A positive decision in one workstream cannot approve another.

- Each decision names its owner, evidence, open items and expiry or re-review condition.
- For non-financial partners, explicitly exclude fund movement and provider administration.
- Public reading needs no NDA. Restricted assessment material and scoped test access are separate decisions.

Start a substantive partnership conversation.

Use this pack to move from a promising idea to a specific integration proposal. Your teams can explore the product and architecture, inspect selected application contracts and permissions, run the synthetic integration kit, and agree on acceptance scenarios. The evidence register gives the review a shared record. Independent penetration-test reports, certifications, signed data terms, live provider acceptance and contractual service levels remain review items to obtain before final vendor approval.

Ready for discovery and technical scoping; not completed vendor approval. Use the evidence register to coordinate the remaining reports, agreements and acceptance decisions.

A partnership should have something worth proving.

Seven starting points for a serious collaboration. Choose the customer problem, shape each party's contribution and define the evidence that would justify a next step.

peyeli.com/en/docs/business-cases

Start with the work your customer needs done.

Peyeli's proposition connects everyday commerce with records people can understand and teams can operate. Your opportunity may sit at checkout, after settlement, inside another software platform or in the way merchants discover a useful workflow. The cases below are collaboration hypotheses, ready to be shaped with your team.

Pilot measures are proposed definitions, not achieved results or committed service levels. Agree eligibility, targets, sample size, baseline and approvals before testing. The downloadable integration kit runs locally with synthetic data.

Collections providers · Connect checkout to a usable record.

A merchant needs more than a payment response: the result has to remain connected to the sale, receipt and eventual settlement. Evaluate a collection relationship that carries a stable reference through that journey, including the moments when the outcome is uncertain.

- Peyeli contribution: payment-intent records, exact asset/amount conventions, receipt context and reconciliation workflows. Provider contribution: approved merchant delegation, authenticated outcomes, lookup and settlement evidence.
- Pilot: one approved collection workflow and one asset; exercise success, rejection, duplicate submission and timeout recovery with synthetic identities and authorized test operations.
- Measure: verified outcomes / eligible submitted intents; duplicate financial effects; time from uncertain submission to authoritative resolution. Report pending cases separately and define the observation window before testing.

Banks and settlement teams · Make the difference explainable.

The useful question after a day of trading is whether the recorded activity agrees with the statement—and who resolves the difference. Evaluate a relationship around statement evidence, explicit matching and a shared exception process.

- Peyeli contribution: recorded payment references, per-asset reconciliation and exception context. Partner contribution: authorized statement access, account identity, settlement formats and resolution ownership.

- Pilot: one authorized statement format and account scope; include matched, short, over, unmatched and duplicate entries.
- Measure: matched eligible entries / eligible entries; unresolved exception count and age; reviewer minutes per statement. Keep totals separate by asset and compare against an agreed manual baseline.

Remittance partners · Keep the customer journey connected.

A hosted transfer journey crosses product and provider boundaries. Explore how a customer can move into an approved provider experience while Peyeli retains a clear reference and an accurate account of the result.

- Peyeli contribution: proposed journey entry, reference continuity and outcome display. Partner contribution: approved corridor, customer eligibility, identity requirements, disclosures, hosted flow and authoritative status.
- Pilot: an explicitly approved test corridor and platform; verify hosted handoff, return, abandoned journey and authoritative status recovery. A return screen alone cannot establish completion.
- Measure: successful hosted handoffs / eligible starts; authoritative status retrieval / submitted test transfers; time to resolve a disconnected return. Identify abandoned and provider-pending cases separately.

Issuers and sponsors · Define the program before the interface.

A useful card relationship begins with the intended cardholder, funding model and operating responsibilities. Evaluate how a proposed wallet experience could fit an approved program, then test the events that determine what the customer sees.

- Peyeli contribution: simulated wallet/card experience and proposed record handling. Partner contribution: program eligibility, issuance authority, funding rules, authorization/clearing evidence and dispute requirements.
- Pilot: an issuer-authorized test program; exercise authorization, rejection, expiry, clearing, reversal and refund. Keep simulated balances separate from program funds.
- Measure: correctly classified lifecycle events / eligible events; unresolved authorization-to-clearing differences; duplicate effects. Program acceptance and launch authority require the issuer's recorded decision.

Accounting and software platforms · Carry meaning across systems.

A record exchange is valuable when the receiving team can use the result without rebuilding its meaning. Evaluate a narrow, consented exchange with explicit object mapping, retry behavior and an exit path.

- Peyeli contribution: selected business records, asset conventions and proposed mapping/recovery. Partner contribution: authorized object definitions, permission model, destination rules and test access.
- Pilot: one agreed object type and one authorized organization; test mapping, repeated exchange, destination rejection, revocation and disconnect. Agree read/write direction before test access.
- Measure: accepted mapped records / eligible records; duplicate destination objects; reviewer minutes per exception; revocation effectiveness. Compare with an agreed manual export/import baseline.

Distribution and enterprise partners · Build an adoption proposition.

The opportunity may be a better merchant workflow rather than a financial rail. Explore a product-distribution or enterprise relationship around a specific customer group, a useful first task and a support model that can be tested.

- **Peyeli contribution:** product walkthrough, proposed onboarding and business-record workflows. **Partner contribution:** target-segment knowledge, an approved distribution channel, participant recruitment and support ownership.
- **Pilot:** an agreed participant cohort completing a defined non-financial task in a synthetic environment. Any actual customer onboarding requires its own reviewed data terms and access scope.
- **Measure:** completed first tasks / eligible invited participants; median task time among completions; help requests per participant; recorded reasons for abandonment. Recruitment, exclusions and observation dates belong in the report.

Institutions and ecosystem partners · Test a shared operating need.

An institutional collaboration should begin with the people it serves and the operating problem it can address. Explore record organization, business education or a defined service workflow with a named sponsor and an agreed method of evaluation.

- **Peyeli contribution:** product context, synthetic records and proposed workflow design. **Partner contribution:** program purpose, participant criteria, domain expertise and independent evaluation ownership.
- **Pilot:** one defined non-financial workflow with synthetic records; agree accessibility needs, participant consent where applicable and the intended report audience.
- **Measure:** completed eligible tasks / attempted eligible tasks; record accuracy under a pre-agreed rubric; assistance required; qualitative feedback with sample size. Keep projected benefits distinct from observed outcomes.

Give the pilot a decision, not just a demo.

A good pilot ends with an answer to a specific question. Use the charter below to make that question measurable, keep responsibilities clear and give every reviewer the same result to assess.

Agree before testing	Record in the charter
Customer and hypothesis	Named sponsor, target participant group, problem, proposed benefit and decision the pilot should inform.
Scope and contribution	Exact workflow, interface/version, environment, asset where relevant, data exchanged, exclusions and each party's delivery owner.
Measurements	Metric definition and unit, denominator, baseline method, agreed target, sample size, observation dates, missing data and failure cases.

Agree before testing	Record in the charter
Commercial assumptions	Implementation effort, operating/support effort, proposed pricing basis and cost owner. Model benefits from measured results; label estimates and never present them as realized savings.
Decision and exit	Required technical, security, legal and provider approvals; evidence owner; go/no-go criteria; access expiry; export/deletion procedure; reviewers and next decision date.

Know which interface you are evaluating.

Choose an interface for your use case, inspect the selected application contracts and run the downloadable synthetic integration kit.

peyeli.com/en/docs/api-surface

Four distinct surfaces

An existing HTTP route is not automatically a supported partner product. The application-session routes below support Peyeli's own authorized product journeys. A server-to-server partner contract needs separate authentication, scope, version, support and acceptance decisions before credentials can be shared.

Surface	Current purpose	Partner boundary
Public documentation	Read authored guides and synthetic examples without an account.	No record access or execution.
Application-session API	Verified user and current organization permissions.	Only an approved tenant/user workflow; no developer keys promised.
Provider protocols	Approved adapters, authenticated incoming evidence and consented callbacks.	Program-specific credentials and agreements; no arbitrary caller.
Privileged operations	Administration, scheduling, diagnostics and platform governance.	Not a partner integration surface; no default access.

Selected application contract catalog

Paths use placeholders, not valid tenant or resource identifiers. All examples are read-only documentation. This selected catalog does not authorize a call or enumerate privileged endpoints.

Method / path	Operation	Success
GET /api/v1/orgs/{org}/payment-intents	payment.view; scoped list.	200 { data: IntentView[] }
POST /api/v1/orgs/{org}/payment-intents	payment.create; idempotent creation.	201 IntentView
GET /api/v1/orgs/{org}/payment-intents/{pi}	payment.view; scope checked on the record.	200 IntentView + attempts + receipt
POST /api/v1/orgs/{org}/payment-intents/{pi}/attempts	payment.collect; provider capability checks.	201 { attemptId, providerReference }

Method / path	Operation	Success
POST /api/v1/orgs/{org}/payment-intents/{pi}/cancel	payment.cancel; unresolved attempts block cancellation.	200 IntentView
GET /api/v1/orgs/{org}/invoices	invoice.view; status filter.	200 { data, canCredit }
GET /api/v1/orgs/{org}/reconciliation/{day}	exception.view; business-day evidence.	200 { day, totals, pairs, lastImport }
GET /api/v1/orgs/{org}/exports/payments.csv	export.csv; plan entitlement may also apply.	200 text/csv attachment

Scope by domain, not the whole route tree.

The source also contains application families for catalog/orders, customers/invoices, expenses/bank files, subscriptions, accounting, consumer planning and approved financial-provider journeys. A partner may only need one. Select records and operations explicitly; the existence of a domain does not establish an external API or production availability.

Relationship	Likely review scope
Enterprise merchant / institution	Orders, invoices, customers, records, authorized exports and branch/user responsibilities.
Software / accounting / analytics	Consented records, mapping, read/write distinctions, synchronization and disconnect.
Distribution / ecosystem	Audience, onboarding, branding, support handoff and attribution; no fund movement by default.
Financial institution / provider	Only the approved collection, settlement, remittance, custody or issuance program.
Consumer product integration	User-owned records and specifically consented exchange; no merchant role inheritance.

Before a supported external integration

The review must approve a precise endpoint set, target environment, authentication mechanism, operation/tenant restrictions, retry and rate limits, schema version and deprecation process. This pack contains no published universal rate limit, official partner SDK, partner OAuth scope catalog, outbound webhook subscription contract or self-service API-key onboarding. A versioned route prefix alone does not provide those guarantees.

The downloadable JSON contract snapshot covers selected data shapes for evaluation. It is not OpenAPI,

an executable collection, a grant or a production compatibility commitment.

Download, run, and inspect the integration kit.

The downloadable evaluation kit contains OpenAPI 3.1 for five selected operations, a chained Postman collection, a JavaScript SDK with TypeScript declarations, and a dependency-free synthetic sandbox. Run `node example.mjs` for an in-process scenario or `node server.mjs` for a loopback HTTP server. No account, API key, provider connection or production server is supplied.

The sandbox binds only 127.0.0.1:3061, holds synthetic records in memory, and makes no provider requests. It demonstrates exact integer money strings, idempotent replay/conflict, read/list, attempt submission and cancellation. Submitted attempts stay processing: no financial confirmation, settled balance or receipt is fabricated. This local simulator does not validate production authentication, tenant isolation, durable recovery, financial rails or provider acceptance. Date filters are synthetic UTC, distinct from the documented application predicate.

```
node example.mjs
# Or start the local HTTP mock, then import partner-postman.json:
node server.mjs
```

The executable specification targets the supplied simulator. Separately authorized application sessions and provider-hosted sandbox approval remain distinct. Use synthetic fixtures only; never expose or tunnel the local server.

A contract an engineer can inspect.

Build a payment-intent workflow you can inspect: create the request, read its state and understand how retries and errors behave.

peyeli.com/en/docs/payment-api

Create a payment intent

The selected POST contract requires Content-Type application/json, a verified application session, payment.create authority and a valid Idempotency-Key. The key is scoped to the organization and an operation containing the caller's identity. Creation returns 201 and the IntentView object; it does not collect money.

Field	Type / requirement	Meaning
amount.amountMinor	Required string; canonical positive integer.	No decimal, exponent, sign, leading zero or whitespace; not a float.
amount.assetId	Required string; existing environment-compatible asset.	Physical cash is excluded from payment intents.
orderReference	Required nonblank string; maximum 120 characters before trimming.	Merchant order reference; trimmed before storage.
description	Optional string; maximum 500 characters.	Omission becomes null in the response.
expiresInMinutes	Optional integer 1–10080; omitted or null defaults to 30.	Requested intent expiry, not a provider execution deadline.

```
{
  "amount": { "amountMinor": "250000", "assetId": "synthetic_asset_placeholder" },
  "orderReference": "evaluation-order-0001",
  "description": "Synthetic evaluation order",
  "expiresInMinutes": 30
}
// Shape example only. Placeholder IDs are not valid for execution.
```

Read a list or one record

The list uses status, q, limit, from and to. Default limit is 50; maximum is 200 without a date range and 1000 with a range. A range requires both valid YYYY-MM-DD dates with from ≤ to. q is at most 120 characters. Results are newest-created first. No cursor or complete-history export guarantee is defined by this list contract.

Dates are inclusive calendar days in America/Port-au-Prince. The current predicate includes records created before the end-day boundary when creation or confirmation is on/after the start-day boundary. Confirmation has no separate upper bound: an older record confirmed after the requested end date can therefore appear. This is not interchangeable with the reconciliation view, which groups payments by confirmation day and statement evidence.

IntentView fields	Representation
id, amount, orderReference	Identifier, { amountMinor, assetId }, merchant reference.
description, providerConnectionId, confirmedAt	Nullable fields; timestamps use ISO strings.
status, settlementStatus, environment	Separate payment state, settlement state and environment.
expiresAt, createdAt, createdBy	ISO timestamps and creating user identifier.
Detail-only attempts and receipt	Attempt history; receipt object or null. List adds provider/providerLabel instead.

Submission is not confirmation.

Starting an attempt requires providerConnectionId, payment.collect authority and its own idempotency key. The 201 response contains attemptId and providerReference. Durable work later submits to the provider; do not treat this response as payment success. Cancellation requires JSON, an idempotency key and payment.cancel authority. An unresolved attempt blocks cancellation.

- Intent states: requires_payment, processing, requires_review, succeeded, failed, canceled, expired.
- A repeated identical key/body replays the stored status/body and adds idempotent-replayed: true. A changed body with the same key returns idempotency_conflict (409).
- The selected common JSON wrapper returns cache-control: no-store and x-request-id. Core failures contain error.code, error.message and error.requestId; unknown failures return a generic internal_error (500).

Choose the recovery from the code.

These are core mappings, not a claim that every endpoint produces every code. A transient transport failure does not prove that an operation was rejected.

Code / HTTP	Evaluation action
invalid_request / 400	Correct the field or content type; review the selected contract.
unauthenticated / 401; forbidden / 403; not_found / 404	Check verified session and current scope. A scoped 404 does not establish global nonexistence.
idempotency_conflict / 409; invalid_transition / 409	Inspect the original request and current state; do not bypass with a new charge.

Code / HTTP	Evaluation action
asset_mismatch, environment_mismatch, capability_disabled, quote_expired / 422	Resolve asset/environment/grant/quote prerequisites before resubmission.
unbalanced_journal / 422	Escalate the financial-record failure; do not invent a balancing movement.
rate_limited / 429; provider_unavailable / 503; internal_error / 500	Preserve request identity; use agreed lookup/backoff and inspect whether submission occurred.

Authority follows the actor and the operation.

Map each actor to the work they can perform, with concrete role examples your engineers and security reviewers can inspect together.

peyeli.com/en/docs/identity-permissions

Identity does not carry every permission.

The reviewed application flow verifies the cookie session with the authentication service and reads organization membership for each request. Membership must be active. Invited or revoked membership cannot authorize an operation. A role string sent by a browser is not the authority source.

The current Business roles are owner, admin, manager, cashier and viewer. They are application roles, not partner OAuth scopes. Role authority does not bypass provider capability, environment, asset or applicable plan-entitlement checks.

Selected current operation matrix

All means all authorized records within the organization; own means records created by that member. Neither permits access to another organization. This matrix documents selected operations, not every product role.

Operation	Owner / admin	Manager	Cashier	Viewer
payment.view / invoice.view	All	All	Own	All
payment.create	Allowed	Allowed	Allowed	Denied
payment.collect / payment.cancel	All	All	Own	Denied
payment.refund	All	All	Denied	Denied
exception.view / export.csv	All	All	Denied	All
settlement.import / exception.confirm_late	All	Denied	Denied	Denied
staff.manage / connection.manage	All	Denied	Denied	Denied

A corporate identity integration is separate.

This guide does not establish enterprise SSO, SCIM provisioning, partner service accounts, delegated OAuth or a corporate MFA policy for the target deployment. Agree on authentication, provisioning, offboarding, session policy and audit evidence before treating any of these as supported. Begin with named users and the narrowest approved workflow.

- Prove that a viewer cannot mutate and a cashier cannot read or collect another member's sale.
- Prove revocation against an existing session, not only a newly signed-in user.
- Never share an owner login, privileged token or production provider credential for evaluation.

Make acceptance reproducible.

Prove the behavior that matters to your integration, from successful requests to duplicates, uncertainty, revocation and exit.

peyeli.com/en/docs/acceptance-matrix

Record the conditions before the result.

Each test record needs a case identifier, interface/version, environment, release identifier, named actor and scope, synthetic inputs, expected outcome, observed result, redacted correlation references, reviewer and date. Mark a case not applicable with a reason. Fixtures, local browser tests and approved provider observations are different evidence classes.

Minimum scenario matrix

Apply financial cases only where the approved relationship executes or records those operations. Software, distribution and data partners still need identity, data-scope, compatibility and exit tests.

Case	Expected observation	Evidence
A01 · Authorized success	Exactly one verified business effect; receipt/journal where applicable.	Request, authoritative outcome and persisted result.
A02 · Duplicate request	Same key/body replays; changed body conflicts.	Response comparison and effect count.
A03 · Duplicate event	No second financial effect.	Event receipt and resulting movement count.
A04 · Timeout after submit	Unknown result remains blocked until lookup/evidence resolves it.	Original reference, lookup and final resolution.
A05 · Invalid evidence	Invalid authentication or mismatched amount/asset/account cannot confirm.	Denied transition and redacted reason.
A06 · Late / reordered outcome	Transition follows current state; exceptional confirmation is explicitly authorized.	Timeline and authorized resolution.
A07 · Reversal / refund	Original history retained; correcting action is separately authorized.	Original and correcting references.
A08 · Settlement variance	Short, over, unmatched and duplicate evidence remain explicit.	Statement, per-asset totals and exception disposition.

Case	Expected observation	Evidence
A09 · Cross-tenant / own-record denial	No foreign record, existence leak or unauthorized effect.	Allowed/denied actor matrix.
A10 · Revocation / disabled capability	Existing session cannot bypass current denial.	Before/after grant and denied operation.
A11 · Dependency outage / recovery	Durable work remains recoverable without blind duplicate execution.	Fault window, backlog and recovery result.
A12 · Version / data / exit	Selected client/schema compatibility, scoped export and disconnect are demonstrated.	Compatibility matrix, redacted export and revocation test.

Acceptance is a signed-off scope.

The decision states passed/failed/not-applicable cases, material defects, permitted operations and limits, responsible reviewers, evidence location and expiry. A fixture pass cannot close a provider-acceptance item; local browser success cannot close a production deployment item. Failed material cases require remediation or an explicit bounded decision from the authorized owner.

Define the day after integration.

Plan who responds, how incidents are investigated and what recovery looks like before your integration becomes an operating responsibility.

peyeli.com/en/docs/operations-support

Assign an owner on both sides.

Before a pilot, name technical and business owners, an incident contact path, escalation authority, operating hours, allowed actions and a change-notification process. The public contact link opens the existing partnership channel; it is not a guaranteed incident hotline. This guide establishes no 24/7 support promise or response-time SLA.

Situation	Required response
Unknown payment result	Preserve the attempt reference; block blind re-charge; obtain authoritative lookup.
Provider unavailable	Separate unsent work from submitted work; apply approved recovery limits.
Settlement difference	Compare statement and per-asset totals; assign and record the exception.
Suspected credential or data exposure	Use the restricted incident process; authorized owner coordinates containment and evidence.
Non-financial sync failure	Preserve source identity and checkpoint; replay only the approved scope.

Measure commitments in the target environment.

No measured throughput, latency percentile, availability target, backup recovery objective or maximum data-loss objective is published by this edition. Agree on proposed SLO/SLA, RTO/RPO, transaction volumes, backlog thresholds, limits and evidence before relying on them. Queue implementation and local tests do not establish capacity or disaster recovery.

- Exercise restoration and replay using isolated fixtures; retain measured recovery results.
- Demonstrate alert ownership, deployed instrumentation and one incident walkthrough.
- Record planned maintenance, release rollback and backward-compatibility responsibilities.

Share a redacted incident packet.

Provide the environment and release, occurrence time with timezone, permitted operation, HTTP/core code, request identity, synthetic reproduction and impact classification. Share financial/customer evidence only through the agreed restricted channel. Never place credentials, raw provider responses, customer KYC or account data in public tickets.

Separate controls from assurance.

Give security and procurement teams a concrete review path: inspect source-backed controls and request the evidence needed for their decision.

peyeli.com/en/docs/security-assurance

What can be answered from source

These statements describe the reviewed implementation. They are not an independent assessment of the deployed service. Scope the questionnaire to the selected integration and ask for operational evidence where source alone cannot answer.

Question	Source-informed answer	Additional evidence
Who authorizes requests?	Verified application identity and current membership/operation checks.	Target session policy, provisioning, revocation and MFA/SSO decision.
How is tenant scope applied?	Organization and own-record restrictions in selected domain flows.	Independent cross-tenant tests on deployed schema/configuration.
Can a browser confirm a payment?	Authoritative verified evidence is required; browser return is insufficient.	Approved provider protocol and adverse-event walkthrough.
Does public documentation expose records?	Static authored allowlist; no internal-file or customer-data reader.	Published-route/asset and deployment review.
Are secrets and encryption operations audited?	Public docs contain no credentials. This pack does not verify live key custody or rotation.	Restricted key-management, transport, storage and rotation evidence.
Independent assurance available?	No SOC 2, ISO certification, PCI attestation or penetration-test report is established here.	Dated report, scope, issuer and remediation status where available.

A restricted review has a separate boundary.

Public material can explain control design and evaluation gates. A named security team may request sanitized architecture detail, assessment summaries, subprocessor evidence, restore results and remediation tracking under agreed confidentiality and sharing terms. Credentials, raw customer data and unrestricted infrastructure access are not assessment deliverables.

Close findings before expanding access.

Record each finding's affected operation, impact, owner, due date, remediation evidence and authorized decision. Obtain current independent review for material authorization or financial changes. This public guide supplies no vulnerability bounty, guaranteed response clock or assurance that no vulnerabilities exist.

Define every data boundary.

Follow the data your relationship needs, from collection and processing to retention, export and an orderly exit.

peyeli.com/en/docs/data-lifecycle

Map the minimum necessary exchange.

The data map names the source, purpose, recipient, authority, environment and lifecycle for each field. Synthetic evaluation data is preferred. Do not assume a provider, accounting platform or model processor receives every category just because it is part of the architecture.

Category	Typical purpose	Review boundary
Identity / membership	Authenticate and authorize the selected actor.	Minimize identity transfer; verify provisioning and offboarding.
Merchant and financial records	Orders, invoices, outcomes, receipts and reconciliation.	Organization isolation; asset/reference scope; approved recipient.
Uploaded receipts / documents	Extraction draft and human review.	Review model/provider transfer, retention and human confirmation.
Provider evidence	Authenticate outcomes and resolve differences.	Share redacted evidence only; exclude credentials.
Audit / diagnostics	Trace authorized actions and investigate failures.	Approved access, retention and redaction policy.

Retention is a reviewed policy, not a guess.

This edition does not establish a per-category retention schedule, deletion-completion SLA, hosting-residency guarantee or approved subprocessor register. A privacy review must obtain the current operational register and signed terms for the selected processing. Technical deletion of an account is not proof that financial records, audit history and backups have all been erased.

- For each category: purpose, owner, retention trigger/duration, deletion method and exceptions.
- For each processor: service, transferred fields, region, terms, subprocessors and change process.
- For backups: access, encryption evidence, expiration and tested restore/deletion implications.

Plan export and exit before onboarding.

Agree on authorized export formats, record coverage, date/time/asset semantics, reconciliation of final outstanding items, disconnection and revocation. The reviewed payment list is bounded and is not a complete export API. A CSV route exists for authorized business users; it does not establish every portability or deletion obligation. Required legal terms and deadlines need the parties' review.

Agree on responsibility before activation.

Shape the working relationship: agree on responsibilities, commercial terms and operating decisions for the partnership you want to build.

peyeli.com/en/docs/commercial-governance

A discussion draft your teams can complete.

Use this term sheet to prepare the commercial, privacy and service schedules for the selected relationship. Every entry below is open for agreement. The completed proposal becomes an input to each party's authorized legal and operational review; this public draft has no signatures or binding terms.

Schedule	Complete together
Parties and purpose	[Legal entities and authorized representatives]; [chosen business case]; [customer group]; [approved workflow and exclusions]; [term and renewal decision].
Commercial proposal	[Deliverables and each party's owner]; [implementation effort and cost owner]; [pricing unit, currency and invoicing]; [third-party fees and taxes]; [acceptance method]; [change approval].
Data and privacy proposal	[Roles and purposes]; [data categories and permitted users]; [processing/subprocessor locations and approval]; [retention and backup expiry]; [rights-request ownership]; [incident notice procedure]; [export, return and deletion evidence].
Service proposal	[Support hours, channels and named escalation owners]; [severity definitions]; [response and restoration targets to agree]; [measurement source and exclusions]; [maintenance and incident communications]; [tested recovery procedure].
Financial schedule, if applicable	[Authorized provider/program and environment]; [assets and limits]; [execution, custody and settlement owner]; [customer eligibility and disclosures]; [refund/dispute authority]; [provider approval references]. Exclude this scope for non-financial relationships.
Legal review and exit	[Confidentiality, intellectual property, liability and dispute terms for counsel review]; [termination and exit assistance]; [revocation and data disposition]; [open items and decision owners]; [controlled original executed document reference after signing].

Write the relationship charter.

Identify contracting entities, use case, eligible audience, countries, assets, environments, interface set, volumes and excluded activities. Define whether the relationship is software supply, distribution, data exchange or a

regulated financial program. The label “partner” establishes none of these automatically.

- Agree pricing, billing basis, provider fees, taxes, settlement deductions and who bears unresolved differences.
- Review customer disclosures, branding, marketing claims and authorization to represent either party.
- Define confidentiality, data terms, liability, disputes, change approval, termination and exit assistance with the responsible teams.

Proposed responsibility map

This map is a discussion framework, not an executed RACI or agreement. Replace each role with a named owner in the selected relationship. For non-financial integrations, mark financial roles not applicable and exclude their permissions.

Work	Peyeli review role	Counterparty review role
Product and application records	Software design and scoped record behavior.	Use case, authorized users and integration mapping.
Financial execution / custody	Approved adapter and evidence handling within agreed scope.	Authorized provider/program owner under the agreement.
Compliance and customer eligibility	Required software gates and agreed disclosures.	Named legal/compliance authority; no delegation assumed.
Incident and settlement operations	Application investigation and exception records.	Provider/source evidence, contacts and authorized resolution.
Data, audit and exit	Scoped access/export and agreed technical controls.	Data rights, review, retention requirements and offboarding.

Commercial agreement is not technical acceptance.

Keep separate approval records for commercial terms, security/privacy, interface acceptance, operational readiness and any financial program. No public fee schedule, liability limit, exclusivity term, live corridor, customer eligibility or service commitment is created by this documentation. These must be explicitly agreed for the selected relationship.

A launch decision needs evidence.

Make the next approval concrete. Assemble the evidence, name the decision owners and track what remains before a scoped launch.

peyli.com/en/docs/evidence-launch

What exists in this pack, and what remains open

Available means supplied in this documentation; source-informed means inspected implementation, not deployed acceptance. Requested means evidence not established by this edition. No requested artifact should be represented as complete in a corporation's questionnaire.

Artifact	Status	Closing evidence
Product, architecture and selected contracts	Available / source-informed.	Target release and approved integration scope.
Role examples and adverse test matrix	Available / source-informed; proposed acceptance cases.	Observed deployed results and named review.
Partner API, SDK and event delivery contract	Not established as a supported external product.	Versioned contract, auth/scopes, limits, SDK/event acceptance if required.
Security assessment / certification	Requested; none established here.	Current independent report, scope and remediation evidence.
Privacy and subprocessor register	Requested.	Approved data map, register, retention and signed terms.
Provider / issuer / bank acceptance	Program-specific; not closed by source tests.	Authorized account/program and observed sandbox/live approval.
Capacity, resilience and support	Requested; no metrics or SLA established here.	Load evidence, restore drill, incident exercise and service terms.
Commercial / financial responsibility	Proposed framework; agreement outstanding.	Named entities, signed scope and responsibility allocation.

Approve the exact pilot, not the whole platform.

The launch record identifies release/environment, approved interfaces, named organizations/users, data categories, financial operations if any, assets/regions, limits, monitoring, support, rollback, expiry and owners. Required evidence must be available and reviewed; any accepted exception must be specific, bounded and approved by the responsible owner.

- Technical acceptance and compatible deployed schema/client/API.
- Security/privacy decisions and verified provisioning/revocation.
- Operational contacts, observability, recovery and settlement owners.
- Commercial terms and separate provider/financial approval where applicable.
- A bounded pilot plan with stop conditions and exit procedure.

How to send this to a corporation.

Send the enterprise review link with a one-page statement of the proposed relationship, the team contacts and the decision requested. Ask teams to review their workstream and return missing evidence requests. Use this pack to earn a scoped evaluation meeting; do not label it completed enterprise due diligence or a promise of production readiness.

Evidence has a collection and review workflow.

Blank bilingual worksheets now define the security assessment/remediation dossier, operational baseline/restore/incident exercises, agreement review schedules and provider/program acceptance dossier. These are usable review artifacts, not completed audits or signed contracts. Original confidential records belong in an approved controlled evidence store; no public route imports them.

The private technical checker requires exact partner/integration/environment/release scope, reviewed authority, issue/expiry dates, artifact hashes and approval references. It rejects missing, draft, expired, changed and wrong-scope records. A complete result remains pending independent release review: the checker cannot authenticate signatures, prove assessor independence, validate provider authority or approve deployment. Financial scope needs real provider acceptance; non-financial scope needs an approved exclusion of fund movement.

No independent security report, executed agreement or live provider acceptance is established by these new templates. Local checks and loopback measurements are explicitly limited technical evidence, not production assurance.