

FULL TECHNICAL DILIGENCE

Project Larkspur

Technical due diligence report on a hosted invoicing and payments platform

Sample report. The code reviewed is a real, public, source-available invoicing codebase. The company, people, deal, and every fact marked "Reported by the Company" are fictional. This sample shows the format and depth of a Ferenda Partners report. It is not an assessment of that project or its maintainers.

Prepared for	The Buyer (a search fund)
Target	The Company, ~\$2.1M ARR, ~14,000 paying accounts
Stage	Confirmatory diligence, after a signed LOI
Date	October 30, 2026
Prepared by	Ferenda Partners

1 Executive summary

The software is better than most at this deal size. It runs on a current framework, ships every few days, and carries a real test suite. The risks sit around the code, not in it. One person writes nearly all of it, holds the copyright to it, and owns the accounts it runs on.

OUR VIEW

Proceed, with three closing conditions

None of the findings should stop the deal. Four are high severity, and three of those four can be fixed before closing with paperwork and account transfers. The fourth, key-person risk, needs a transition plan and a hire, and should shape how the price is paid.

12

findings: 4 high, 7 medium, 1 informational

90%

of commits in the past 12 months by one developer

\$79–146k

one-time cost to fix, year one

93

releases in the past 12 months

What we recommend for the purchase agreement

1. **Closing condition: IP assignment.** The founder assigns all copyright in the codebase to the Company (F-01).
2. **Closing condition: account transfers.** Every production account moves into the Company's name, with the Buyer holding admin access and MFA on each (F-02, F-11).
3. **Closing condition: payments platform.** Stripe confirms in writing how the Connect platform account moves from the founder's earlier LLC to the Company (F-03).
4. **Founder transition agreement.** Twelve months of defined knowledge transfer. Consider tying part of the consideration to its milestones through a holdback or earnout (F-04, F-05).
5. **Price and escrow.** Use the \$79–146k remediation range in negotiation. Ask deal counsel about a specific indemnity for the open-source license question in F-06.

Findings at a glance

ID	FINDING	SEVERITY	SOURCE	COST TO FIX
F-01	Copyright in the code is held by the founder personally	HIGH	● Code	\$3–6k
F-02	Production accounts are in the founder's personal name	HIGH	● Reported	\$8–15k
F-03	Stripe Connect platform belongs to a different legal entity	HIGH	● Code + reported	\$10–25k
F-04	One developer writes 90% of the code	HIGH	● Code	\$15–25k
F-05	Deployment and operations are undocumented	MEDIUM	● Reported	\$10–15k
F-06	A GPL-2.0 dependency ships inside an ELv2 product	MEDIUM	● Code	\$6–12k
F-07	The source-available license limits the moat	INFO	● Code	n/a
F-08	78 known advisories in front-end dependencies	MEDIUM	● Code	\$4–8k
F-09	Tests and static analysis don't guard releases	MEDIUM	● Code	\$3–6k
F-10	Backups have never been restored	MEDIUM	● Reported	\$2–4k
F-11	Admin access lacks MFA; no security test since 2023	MEDIUM	● Reported	\$15–25k
F-12	25 payment gateways and many integrations for one maintainer	MEDIUM	● Code	\$3–5k
Total one-time cost to fix				\$79–146k

Costs are one-time estimates of engineering and legal time at typical market rates. They exclude the ongoing cost of a senior engineer hire (F-04), which we estimate at \$165–195k a year fully loaded.

2 Scope and method

What we reviewed

- **The backend codebase** at the release branch as of October 2026, with its full git history back to 2018.
- **Dependencies** in `composer.lock` (244 PHP packages) and `package-lock.json` (496 JavaScript packages).
- **Build and CI configuration**, including GitHub Actions workflows, Dependabot settings, and static analysis config.
- **Interviews** with the founder (two sessions, 3.5 hours in total) and one contract developer (1 hour).
- **A screen-share walkthrough** of the cloud console, DNS, payment-processor dashboards, and app store accounts.

How we reviewed it

QUESTION	METHOD
Who wrote the code, and who owns it	Git history by author and year; license and copyright headers; contributor license terms
How big and how healthy is it	Line counts by area (<code>scc</code>); test suite and CI review; static analysis config; release history
What does it depend on	License inventory of every locked package; known-vulnerability scan (<code>osv-scanner</code>) on both release and development branches
Are secrets exposed	Secret scan of the current tree (<code>gitLeaks</code>), with every hit triaged by hand
Who controls production	Founder interview and screen-share of each account's ownership and admin settings

Limits of this review

- **Other repositories.** The React admin interface and the mobile and desktop apps live in separate repositories. We did not review them. A full engagement would.
- **Git history secrets.** We scanned the current code for secrets, not every past commit. A key committed and later deleted would not appear.
- **No code execution.** We read and scanned the code. We did not run the test suite or load-test the hosted service.
- **Not legal advice.** License findings flag questions for deal counsel. They are not legal conclusions.

Reading the findings. Each finding is marked by its source. ● **Code** means we observed it directly in the repository. ● **Reported** means the Company told us or showed us on screen, and we did not verify it independently.

3 The system as it is

The Company sells hosted invoicing, billing, and payments software to small service businesses. The same codebase also ships as a self-hosted product, with paid license keys that unlock white-label features.

Architecture

The backend is a single Laravel 12 application on PHP 8.2 to 8.5, backed by MySQL or MariaDB, with Elasticsearch for search. Background work runs on Laravel queues. A separate React single-page app provides the admin interface, and Livewire renders the client portal. The application talks to 25 payment gateways, two banking-data providers, QuickBooks, four email providers, and S3-compatible storage.

One codebase serves both the hosted service and the self-hosted product. Around 200 files branch on whether the code runs hosted, self-hosted, or under a paid license. That keeps the two products in step, but it means every change has to be thought through twice.

Codebase at a glance

MEASURE	VALUE	WHAT IT TELLS US
Application code	~236k lines	PHP in <code>app/</code> , after removing 187k lines of static data tables. Large for the revenue, as is common in mature invoicing products.
Tests	652 files, ~178k lines	A serious suite by any standard for a company this size.
Framework	Laravel 12	Current major version.
History	31,000+ commits	Eight years of work by 300+ contributors.
Release pace	93 in 12 months	Roughly two releases a week. Fast, and almost all from one person.
Dependencies	244 PHP / 496 JS	PHP packages are current and free of known advisories. JavaScript packages lag (F-08).

Strengths

- **Current platform.** Laravel 12 on PHP 8.2 through 8.5 means no forced upgrade project in the first year.
- **Real test coverage.** The suite runs in 8 parallel shards across 4 PHP versions, so the team tests against the PHP versions customers actually run.
- **Steady delivery.** Releases ship about twice a week and the product keeps adding features, including recent e-invoicing work for European markets.
- **Clean contributor terms.** Outside contributions go through a contributor license agreement, which protects the Company's right to license the code as it does.
- **Clean dependencies and few secrets.** None of the 244 PHP packages has a known vulnerability, and only one of 429 secret-scanner hits is a real credential (F-11).

4 Findings

Ownership and access

F-01 Copyright in the code is held by the founder personally

HIGH ● Code

FOUND	The <code>LICENSE</code> file names an individual, not the Company, as the copyright holder, and has since 2019. The contributor license agreement grants rights to the project, so contributions may also flow to the founder rather than the Company.
WHY IT MATTERS	Buying the Company may not buy the code. Without an assignment, the founder could keep the core IP after closing.
RECOMMEND	Make a written IP assignment from the founder to the Company a closing condition. Have counsel confirm which entity the CLA names.
COST	\$3–6k in legal fees.

F-02 Production accounts are in the founder's personal name

HIGH ● Reported

FOUND	The cloud account was opened in 2019 with the founder's personal email and is billed to a personal card. The primary domain sits in the founder's personal registrar account, and DNS runs through a Cloudflare account under the same email. The Apple and Google developer accounts are individual accounts in the founder's name.
WHY IT MATTERS	Whoever holds these accounts can take the product offline. Moving individual app store accounts to an organization takes weeks, and both stores review the transfer.
RECOMMEND	Move every account into the Company's name before closing, with the Buyer as an admin. Start the app store transfers now, because they take the longest.
COST	\$8–15k, about 40–60 hours of engineering and admin time.

F-03 The Stripe Connect platform belongs to a different legal entity

HIGH

● Code + reported

FOUND	The code onboards customers' merchant accounts through Stripe Connect, GoCardless OAuth, and Square OAuth. The founder reports that the Stripe Connect platform account is registered to a consulting LLC that predates the Company.
WHY IT MATTERS	Each customer's Stripe account is linked to that platform. If the platform can't move cleanly to the Company, customers may have to reconnect their payment accounts, and some will churn instead. Payment processing is the stickiest part of the product.
RECOMMEND	Get Stripe's written transfer process before closing, and make it a closing condition. Check the GoCardless and Square app registrations the same way.
COST	\$10–25k in engineering and support time, plus churn risk if customers must reconnect.

Key-person risk

F-04 One developer writes 90% of the code

HIGH

● Code

FOUND	In the past 12 months, Developer A (the founder) wrote about 1,700 of 1,900 commits, or 90%. That share has risen from about 50–60% in 2020–2022. The next most active contributor wrote 31 commits in the same period. See Appendix A.
WHY IT MATTERS	The product's pace, its architecture, and its 25 payment integrations all live in one person's head. If the founder leaves after closing, releases stop and fixes slow down, and a new engineer will need months to become productive in 236k lines of code.
RECOMMEND	Sign a 12-month transition agreement with defined knowledge-transfer milestones. Hire a senior Laravel engineer in the first 90 days to shadow the founder. Consider tying part of the consideration to the transition milestones.
COST	\$15–25k for structured knowledge transfer. The hire is an ongoing cost of \$165–195k a year.

F-05 Deployment and operations are undocumented

MEDIUM

● Reported

FOUND	The founder deploys by logging into servers over SSH and running a personal script. There is no written runbook for deploys, rollbacks, queue failures, or outages. Server configuration lives outside the repository.
WHY IT MATTERS	This sharpens F-04. Even with the code in hand, no one else can ship a release or recover from an outage today.
RECOMMEND	Move the deploy into CI with a scripted rollback, and write runbooks for the five most common incidents. Make both early transition milestones.
COST	\$10–15k.

Vendors and licenses

F-06 A GPL-2.0 dependency ships inside an ELv2 product

MEDIUM

● Code

FOUND	The PDF library <code>mpdf/mpdf</code> is licensed GPL-2.0-only. It arrives through an e-invoicing package and is referenced directly by the Company's PDF and e-invoice code. The product itself is licensed under the Elastic License 2.0, which adds restrictions that the GPL does not allow.
WHY IT MATTERS	The self-hosted product is distributed to customers. Distributing GPL code inside a more restrictive license can breach the GPL, and the mismatch may need disclosure in the purchase agreement. The hosted service carries less exposure, because hosting alone is not distribution under GPL-2.0.
RECOMMEND	Ask deal counsel to assess the exposure. Engineering can remove it by moving that rendering path to <code>dompdf</code> , which the product already uses under the LGPL.
COST	\$6–12k to replace the renderer.

F-07 The source-available license limits the moat

INFO

● Code

FOUND	The full source is public under the Elastic License 2.0. Anyone may self-host it for their own use for free. Only offering it as a hosted service to others is forbidden. The license-key checks that gate white-label features are visible in the public code.
WHY IT MATTERS	Revenue depends on convenience and support, not on secrecy. A determined customer can always self-host. This is a business question rather than a defect, and it belongs in the revenue model.
RECOMMEND	In commercial diligence, test how many paying customers could self-host and what share of self-hosted installs convert to paid license keys.

F-12 25 payment gateways and many integrations for one maintainer MEDIUM ● Code

FOUND	The application carries 25 payment-gateway drivers, about 25k lines in total, plus integrations with two banking-data providers, QuickBooks, four email providers, and S3-compatible storage.
WHY IT MATTERS	Every one of these is an outside API that changes on its own schedule. Each change lands on the same one developer (F-04). Some gateways probably serve only a handful of customers.
RECOMMEND	Pull revenue and transaction volume by gateway. Plan to retire the low-use ones, and give each remaining integration a named owner.
COST	\$3–5k for the usage analysis and a retirement plan.

Code and architecture

F-08 78 known advisories in front-end dependencies MEDIUM ● Code

FOUND	The JavaScript lock file contains 24 packages with 78 published advisories. Most sit in build tools such as <code>vite</code> and <code>rollup</code> , which never reach users. A few ship to browsers, including <code>axios</code> , <code>pdfjs-dist</code> , and <code>form-data</code> . The development branch shows the same results. Dependabot is configured, so its updates appear not to be getting merged.
WHY IT MATTERS	The browser-side packages are real exposure in the client portal. The build tools are low risk but signal that dependency upkeep has slipped.
RECOMMEND	Upgrade the runtime packages first, then the build chain. Assign someone to merge Dependabot updates every month.
COST	\$4–8k.

F-09 Tests and static analysis don't guard releases MEDIUM ● Code

FOUND	The test suite runs only on pushes to the development branch. It doesn't run on pull requests or on the release branch that customers receive. PHPStan is configured at level 5, but its config includes a baseline file that isn't in the repository, and CI never runs it.
WHY IT MATTERS	A good test suite only protects customers if it blocks bad releases. Today a change can reach the release branch without the suite ever running against it.
RECOMMEND	Run the suite on every pull request and on the release branch, and block merges on failure. Regenerate the PHPStan baseline and add it to CI.
COST	\$3–6k.

Security and data

F-10 Backups have never been restored

MEDIUM

● Reported

FOUND	The database takes nightly managed snapshots with 7-day retention. No one has ever restored one to a clean environment. Uploaded documents in object storage have no separate backup.
WHY IT MATTERS	Invoices and payment records are the customer's books. An untested backup may fail when it's needed, and seven days is too short to catch slow data corruption.
RECOMMEND	Extend retention to 35 days, add versioning to the document storage, and restore to a clean environment every quarter.
COST	\$2–4k, plus a small increase in storage cost.

F-11 Admin access lacks MFA; no security test since 2023

MEDIUM

● Reported

FOUND	The cloud root user has no MFA, and three former contractors still have active IAM users. The last external penetration test was in 2023. In the code, <code>codecov.yml</code> contains a committed upload token for the coverage service, and a test environment file contains a sample Laravel <code>APP_KEY</code> .
WHY IT MATTERS	The Company stores payment-gateway credentials and banking-data tokens for thousands of customers. A compromised admin account exposes all of them.
RECOMMEND	Turn on MFA and remove stale users in the first week, which costs nothing. Rotate the coverage token and confirm production uses its own <code>APP_KEY</code> . Commission a penetration test of the hosted service and client portal within 90 days.
COST	\$15–25k, almost all of it the penetration test.

5 The system as it should be

Within a year, the product should ship at its current pace without depending on one person.

Target state at month 12

AREA	TODAY	TARGET
Ownership	Code and accounts held by the founder	Held by the Company, Buyer as admin
People	One developer writes 90% of commits	Two engineers who can ship any area
Releases	Manual SSH deploy by the founder	CI deploys with tested rollback and written runbooks
Quality gates	Tests run on one branch only	Tests and static analysis block every merge
Dependencies	78 open front-end advisories	No open high advisories; monthly updates
Resilience	Untested 7-day backups	35-day retention, quarterly restore tests
Security	No MFA on root; last pen test 2023	MFA everywhere; annual penetration test
Integrations	25 gateways, one maintainer	Low-use gateways retired; named owners

The first 90 days

Days 0–30

Secure and stabilize

- Confirm all closing-condition transfers are complete and tested (F-01, F-02, F-03).
- Turn on MFA everywhere and remove stale contractor access (F-11).
- Extend backups and run the first restore test (F-10).
- Start the engineer search and knowledge-transfer sessions (F-04).

Days 31–60

Make releases repeatable

- Move the deploy into CI with a scripted rollback, and write the first runbooks (F-05).
- Run tests on every pull request and restore static analysis (F-09).
- Upgrade the browser-facing JavaScript packages (F-08).
- Get counsel's view on the GPL dependency and act on it (F-06).

Days 61–90

Spread the knowledge

- The new engineer ships a release and runs an incident solo (F-04).
- Complete the penetration test and fix its high findings (F-11).
- Publish the gateway retirement plan (F-12) and review the transition milestones.

A Evidence

A.1 Commit concentration by year

YEAR	COMMITTS	AUTHORS	TOP AUTHOR	TOP AUTHOR SHARE
2020	~2,200	12	~1,300	58%
2021	~5,300	24	~2,600	48%
2022	~3,700	17	~2,300	62%
2023	~3,700	28	~3,000	79%
2024	~3,100	29	~2,000	64%
2025	~1,800	19	~1,600	91%
2026 (to Oct)	~1,600	34	~1,500	89%

Non-merge commits on the release branch, rounded to the nearest hundred. The top author is the same person in every year.

A.2 Dependency licenses

LICENSE	PHP PACKAGES	NOTES
MIT, BSD, Apache-2.0, ISC	234	Permissive. No obligations beyond attribution.
LGPL (2.1 and 3.0)	4	<code>dompdf</code> and its font and SVG libraries, <code>smalot/pdfparser</code> . Fine as unmodified libraries.
GPL-2.0-only	1	<code>mpdf/mpdf</code> , through an e-invoicing visualizer. See F-06.
Dual BSD / GPL	2	<code>nette/schema</code> and <code>nette/Utils</code> . Usable under BSD.
Proprietary	1	The Authorize.net SDK. Counsel should confirm its terms allow redistribution in the self-hosted product.
OSL-3.0	1	<code>apimatic/jsonmapper</code> , through the Square SDK. A copyleft license that also covers network use. Fine while unmodified, but counsel should confirm.
Elastic-2.0	1	The Company's own e-invoice package, pinned to a development branch rather than a release.

A.3 Known vulnerabilities

ECOSYSTEM	PACKAGES	VULNERABLE	ADVISORIES	SHIPS TO USERS
PHP (Composer)	244	0	0	n/a
JavaScript (npm)	496	24	78	axios, pdfjs-dist, form-data, follow-redirects, immutable

Scanned with osv-scanner against the OSV database on October 30, 2026. Results were identical on the release and development branches.

A.4 Secret scan triage

The scanner raised 429 hits in the current code. We reviewed every one. One is a real credential, and one is a sample key worth checking.

WHERE	COUNT	WHAT IT WAS
API documentation	178	Example <code>curl</code> calls and Slack URLs with placeholder values
Tests and seed data	176	Test fixtures and demo data
Application code	41	Internal gateway type identifiers and sample webhook payloads in comments
Database migrations	26	The same gateway type identifiers
Compiled front-end files	6	Variable names in minified JavaScript
<code>.env.dusk.example</code>	1	A sample Laravel <code>APP_KEY</code> . Harmless unless production reuses it. See F-11.
<code>codecov.yml</code>	1	A real upload token for the coverage service. Low impact, but it should be rotated. See F-11.

B Open items

These documents were requested and not yet received. None changes our view, but each should be checked before closing.

ITEM	WHY WE ASKED	RELATED
The CLA text and the entity it names	Confirms who holds rights to outside contributions	F-01
Stripe's written platform transfer process	Confirms customers won't have to reconnect	F-03
Contracts with the two contract developers	Confirms their work was assigned to the Company	F-01, F-04
Twelve months of cloud invoices	Confirms hosting cost and finds untracked services	F-02
Revenue and volume by payment gateway	Sizes the gateway retirement plan	F-12
The 2023 penetration test report	Shows whether its findings were fixed	F-11

About this report. Ferenda Partners provides technical assessments only, not legal, tax, financial, or investment advice. Findings marked "Reported" rely on information from the Company that we did not independently verify. This sample uses real public code with a fictional company and deal.