

Collections, Phase One

What actually shipped between 19–21 August 2026, and what to distrust about it.

Internal engineering walkthrough · ~2,500 words · prepared as a source document

Why this document exists

Between the 19th and the 21st of August, twenty-three items were marked done on the collections phase tracker. Every one of them merged, passed CI, and cleared a blocking code review. Not one of them has been looked at by anyone outside engineering.

That gap is the reason for this walkthrough. "Done" on that tracker means *merged and green*. It does not mean *confirmed correct by someone who knows what should have happened*. For a lot of this work those are the same thing. For some of it they are very much not, and this document tries to be honest about which is which.

There is a second reason. The tracker rows were written when each item was still a problem, so they describe what was *broken*, in the present tense. A row still reads "nothing today can hear one" for a capability that now exists. Reading the tracker today tells you what was wrong in August, not what is true now. So this is the version that says what is true now.

Step zero, before anything else: is the customer configuration stamped?

There is one setting that everything downstream depends on, and if it is missing then a large part of what follows does nothing at all — silently, and by design.

The deterministic email parsers — the ones that read the eviction firm's mail and the rental-assistance programmes' mail — run on every property that has an email integration. But the firm's subject-line templates are the *firm's*, and they live in source code. The things that are specific to a customer — which counsel they retain, which assistance programmes they work with, how their unit numbers are written, the street addresses in their mail that must never be mistaken for a unit number, their client id at the firm, and the landlord entity that appears as plaintiff on a court caption — those are customer configuration. They live on the property record, in a field called collections correspondents.

Absent configuration is the fail-closed state. A property with no correspondents configured produces no parses at all. Not wrong parses. No parses.

The reason it is built that way is worth understanding, because it is a genuine multi-tenancy trap: a claimed parse *replaces* the model's unit list, including when that list is empty. So one property's unit-number pattern applied to another property's mail does not merely fail to add a unit reference — it deletes a correct one. Lossy, not inert. Fail-closed is the only safe default.

The practical consequence: before you evaluate any parser output, confirm the configuration is actually stamped for the property you are looking at. The script that manages it is dry-run by default — running it without the apply flag prints the current configuration and writes nothing. That is the check. I believe it was stamped earlier, but "I believe" is exactly the standard this document is trying to avoid, so treat it as unconfirmed until you have seen the dry run print it back.

Part one: the things you can click

The collections list

This is the page that changed most visibly. It follows the shape of the renewals board: aged buckets across the top, then one table with tabs for what requires action, what is in progress, and what is resolved.

Two things landed here that are worth looking at specifically.

The first is the zero-balance ratchet. Roughly forty-three thousand six hundred dollars of already-paid debt was frozen on that page — balances that had been settled but were still being displayed and still driving the page. That had to be fixed before showing every account, because otherwise widening the view would have widened the error.

The second is the Stage column, which shipped on the morning of the 21st and replaced a column called Status. This is the one I would most like you to be sceptical about, and there is a whole section on it below.

The tenant page collections panel

On an individual tenant, there is now a collections panel with three parts: an aged strip showing where the money sits, a lifecycle rail showing where that account is in the delinquency process, and a chronology that interleaves four sources — what the system did, what an operator recorded, what the law firm sent, and what the assistance programme sent.

The reason this exists is that the collections signals extracted from email had literally zero readers. A defective demand notice, or an assistance approval, existed only as an unread email. It was being parsed and stored and then never surfaced anywhere a human would look.

Part two: the things that run where you cannot see them

This is the larger half of the work, and it is the half you cannot verify by clicking.

The auto-reply guard

The single highest-value fix in the whole phase, measured by lines of code changed.

The eviction firm sends its case correspondence through a ticketing system. Our auto-reply guard was classifying that entire leg as bulk mail and deleting it. That included a "Defective Demand — Unable to File" notice from April. In four months, not one signal had ever been extracted from that queue, because the mail was being thrown away before anything could read it.

The fix was to add the domain to an allowlist. The allowlist had one domain in it.

The email parsers

Deterministic parsers for law-firm mail and rental-assistance mail, derived from a tag taxonomy built over the real corpus. Across all fifteen thousand one hundred and twelve messages, they produce eight hundred and forty-four claims, at one hundred percent flow precision and about eighty-seven percent recall.

A hundred percent precision on that volume deserves an explanation rather than applause: these are deterministic rules, tuned to templates that a specific firm and specific programmes actually send. They are precise because they refuse anything they do not recognise. Recall is the number that has room in it, and eighty-seven percent means roughly one in eight real messages is not claimed.

The two programmes involved are Brothers Redevelopment, and the CED Project, whose assistance has a ten-thousand-dollar cap and cheques that expire after ninety days. That expiry is not trivia — a cheque that expires unspent is money that was approved and then lost.

Hearing "stop the process"

This is the capability I would most want on a podcast, because the stakes are legible.

In the corpus there are twenty recorded messages that amount to "cancel the eviction, assistance is coming". Four of them arrived *after* a writ was already at the sheriff. Until this shipped, nothing in the product could hear any of them.

It now detects seventeen of the twenty-one real stop messages. And on all four units where a stop landed after a writ had reached the sheriff, the approval queue refuses to send a demand.

Two design choices are worth knowing. First, the detector is deliberately generous about *how* the stop was phrased — a stop stated as a question ("can you accept a commitment letter and stop the eviction?") or as a condition ("we will cancel as soon as the balance is paid") both count. The reasoning is that the errors are not symmetric: a false positive means we decline to text somebody about money, and a false negative means we text a resident whose assistance cheque has already cleared, or one the sheriff is scheduling on.

Second, a resident cannot trigger it. Only the counterparties — the firm, the programmes, the operator — can.

Payment plans

AppFolio's payment-plan column was being read off the wire and thrown away. A resident on an active arrangement was invisible to us and fully dunnable.

The subtlety that caught this: the column is money in one report and a flag in another. Reading only one of them would have graded every real payment plan as absent — which is to say, the naive implementation would have looked like it worked and been wrong about every single case.

Marking someone as being on a payment plan now actually stops the cadence, which it did not do before.

Category drift

A ten-thousand-dollar rental-assistance payment confirmation was being filed as a vendor cost. Filtering on the delinquency and eviction categories was losing about a quarter of collections traffic.

There is a decision embedded here that I think is the most encouraging thing in the phase. A benchmarked prompt upgrade was prepared, and it improved the category score substantially on clean gold data. When it was measured against *real* mail, it was a regression. It was deliberately not shipped. The measurement was allowed to overrule the expectation.

Part three: the safety work

Three items were not features at all.

A cross-tenant path was closed on the AppFolio trusted-sender gate. The gate that authorises notice-to-vacate and countersignature side-effects was trusting one customer's mail domain globally, and unit-number-only matching could land a notice on the wrong property's resident.

A drift guard now makes it impossible to compile a customer's identifiers into shared source. It covers three thousand one hundred and two files and nine machine identifiers, and it starts from a dated allowlist that can only shrink. It found the exact hole it exists to close, which is the part I like: a comment-stripper that made five hundred and forty-six lines of one file invisible while reporting them as inspected.

And real resident names were scrubbed out of source. That one kept widening. The final sweep found thirty-two distinct real people across seven directories, including directories that earlier passes had reported clean or never looked at. The sharpest find was a test fixture holding two siblings from one real household along with two real phone numbers, written surname-first, which is precisely why three previous sweeps had missed it.

Part four: what "done" does not mean

This is the section to pay attention to.

The Stage column is 82.9 percent right

Stage is derived. It is not typed in by anybody. It comes from the same lifecycle graph the tenant page draws, which was a deliberate choice — one derivation, not two that can disagree.

It was measured against seventy-six real units reconstructed from the corpus, and it agrees on sixty-three of them. That is eighty-two point nine percent.

Of the thirteen disagreements: six are a declared limitation, where the product has no source for that state yet and says so on screen. Seven are an office-channel blind spot — things that happened by phone or in person and never touched a system we read.

Three of those seven run in the dangerous direction. In those three cases the product shows an active legal posture for an account whose assistance had actually cleared off-channel. That is the failure mode that matters, it is named openly in the pull request, and it is the specific thing I would like you to look for when you go through real accounts.

There is also an honesty detail on that page: three filter options can never match anything, and each is labelled with the reason. "Demand ready — no source yet." "Evicted — no source yet." "Fee eligible — no rows land here." Those are not bugs. They are the vocabulary being complete while the plumbing is not, stated rather than hidden.

The attachments are pointers to data we do not have

Four of the highest-value emails — the delinquency report, the eviction filing report, the results of cases, and the writs at the sheriff — carry their per-resident detail inside PDF and spreadsheet attachments. We captured the filenames. We did not capture the contents. So those four are currently pointers to data we cannot read.

Thirty-two phone numbers still match the corpus

The name scrub measured what remains: of eight hundred and seventy-eight phone numbers in source, thirty-two match numbers in the mail corpus. That was reported as the next worklist rather than quietly fixed, because separating company lines from personal ones needs case-by-case judgement.

Nothing writes a stage automatically yet

This is the honest headline. The architecture decision that permits email signals to move a collections stage was accepted on the 21st. Accepted is not built.

Specifically: automatic stage transitions are only permitted on a case that a human has explicitly linked to a tenancy, and that link did not exist anywhere in the system. So the first piece of work in that sequence is not the write at all — it is the link. That is in review now. The automatic stop-more write, the release proposals that a person confirms individually, and the alert for a case nobody has linked, all come after it.

The design has a shape worth stating plainly, because it is what makes the whole thing safe. Anything that makes the product do *less* to a resident can happen automatically. Anything that removes a restraint, or moves an account further up the legal ladder, is a proposal that a human confirms one at a time. And an unlinked case writes nothing but still raises the alarm — deliberately, because the message most likely to arrive on a case nobody has linked yet is the assistance approval that arrives after a writ. Gating the alarm on the link would have made the worst case the silent one.

Part five: what I would actually test

Six things, in the order I would do them.

One. Confirm the customer configuration is stamped, with the dry run. Everything downstream is inert without it, and inert looks identical to working-but-quiet.

Two. Open the collections list and find an account whose stage you personally know. Check whether the stage is right. This is the eighty-three percent number made concrete, and you are the only available instrument for it.

Three. Specifically hunt for the risky-direction case: an account showing an active legal posture where you know assistance actually came through. Three of those exist in the measured set. If you find one, that is not a surprise — it is the known failure reproducing, and worth confirming it looks the way we think it does.

Four. Open a tenant with law-firm history and read the chronology. The question is not whether it renders, but whether the story it tells matches what you remember happening.

Five. Check a resident you know is on a payment plan, and confirm they are not being chased.

Six. Look at the three "no source yet" filter options and decide whether that honesty is the right call, or whether an option that can never match should not be shown at all. That is a product judgement, not an engineering one, and it has not been made by anybody yet.

What is left

Two rows remain open in Phase One.

The demand builder, which is Fede's, has an open pull request with merge conflicts.

And the signals-to-stage write path, which is the four-piece sequence described above, with the first piece in review.

Everything else in Phase One is merged and running. Whether it is *right* is the question this document exists to hand to you.