

How Eloquens® AI automated auto-renewal opt-outs

How one Eloquens® AI workflow absorbed ~70 cancellation emails a day, recovered close to a full-time role, and paid for itself in under two working days.

~70/day

Requests handled

10 min

Response time

~6 hrs/day

Labor removed

< 2 days

Payback

Subject: Please cancel auto-renewal on my subscription

3:47 AM EST

Hi Support Team,

I'd like to opt out of the auto-renewal on our current subscription number 1234567. We don't plan to renew when the term ends, so please make sure we don't be charged when it expires in September.

Can you send me a written confirmation that auto-renewal has been turned off?

Thanks,
Sarah

The why: business and operational need

Opting out of auto-renewal is a routine, low-complexity request. At scale, though, the volume turns it into a genuine operational load. The requests arrive as ordinary emails (roughly 70 every working day) and each one carries the same chain of manual steps.

- Read the email and work out what the customer is actually asking for.
- Identify the correct subscription from whatever the customer wrote.
- Confirm the subscription is eligible to be opted out.
- Make the change in the main portal.
- Log the request in a tracking spreadsheet.
- Update the matching deal in the CRM.
- Write back to the customer with a clear confirmation.

Done by a person, that is about five minutes per request, and with around 70 a day, close to six hours of staff time every working day spent on repetitive processing.

It is also work that is easy to get wrong. A misread subscription number, a missed eligibility deadline, or an inconsistent reply each create downstream problems. And because every request waits for someone to be available, a customer can wait hours for what should be an instant confirmation.

The goal: handle the request reliably from end to end (accept it as a normal email, apply the eligibility rules the same way every time, record it correctly, and respond fast) without a person touching each one. That is exactly the job Eloquens® AI was put to work on.

The how: the process, and the topology

A customer emails a dedicated cancellation inbox in their own words. Eloquens® AI reads the message and first decides whether it is a request it can act on at all. For a valid request, it extracts the subscription number or license key, no matter how the customer phrased it.

- **Multiple items in one email:** it processes the first, then asks the customer to send the rest as separate emails, one identifier per email.
- **No usable identifier:** it never guesses. It replies asking the customer to resend with a valid subscription number or license key.

Eloquens® AI then calls a purpose-built integration that does the actual work against the account portal. That integration authenticates, retrieves the subscription, and applies the opt-out policy identically every time:

Condition	Action taken
Already opted out	No change made; customer told it was already done.
Eligible and >30 days to expiry	Opt-out is processed.
Fewer than 30 days to expiry	Request declined; customer told opt-outs must be requested at least 30 days before expiry.
No subscription / license found	Customer asked to double-check and resend.
Anything else unexpected	Held for human follow-up rather than guessed at; customer told it is being looked into.

When an opt-out is completed, the action is written to the tracking spreadsheet and the subscription's auto-renewal status is updated in the CRM, so records stay consistent across systems. Once the integration returns its result, Eloquens® AI replies in plain language with no technical detail, and where the opt-out succeeded with no reason given, it asks why, opening the door to a win-back conversation.

The design deliberately splits two responsibilities: Eloquens® AI handles interpretation and communication (the parts that benefit from understanding free-text email), while the purpose-built integration handles the rules and the actual changes (the parts that must be exact and identical every time).

Why Eloquens® AI

The whole workflow hinges on the one thing rules engines and rigid forms can't do, make sense of a real customer email. That is where Eloquens® AI earns its place:

- **Reads intent, not templates.** It pulls the right subscription number or license key out of free-text email however the customer phrased it.
- **Knows when not to act.** If no valid identifier is present, it asks the customer to resend rather than guessing; no wrong subscription ever gets touched.
- **Replies like a person.** Plain-language confirmations with no technical detail, and a natural “why?” prompt that turns a cancellation into a win-back opening.
- **Orchestrates the rest.** It calls the purpose-built integration for the exact, rule-bound work and hands anything uncertain to a human, so speed never comes at the cost of correctness.

AI-powered automated reply:

3:47 AM EST

Subject: RE: Please cancel auto-renewal on my subscription

Hi Sarah,

Thank you for reaching out. Your auto-renewal opt-out request has been processed successfully.

Your subscription will remain active until the end of the current term. After that date, it will not renew automatically, and no charges will be applied to your account.

If you decide to renew in the future, you can do so at any time.

We'd also love to hear what led to your decision — your feedback helps us improve.

Thank you for being a valued customer. If you have any questions, we're happy to help.

Best regards,

The Support Team

Time saved & ROI: translated into people

The clearest way to picture the impact is to convert it into human labor. Here is the arithmetic, step by step.

The daily labor that disappears

Requests per working day	~70
Manual handling time per request	~5 minutes
Daily manual effort ($70 \times 5 \text{ min} = 350 \text{ min}$)	≈ 5.8 hours/day

That is close to six hours of work entirely removed every single working day, not just sped up, because no person processes a request anymore.

What that means in headcount

Assuming a productive workday of about 7.5 hours, the saving maps onto people like this:

Daily effort removed	≈ 5.8 hours
Productive hours per person per day	≈ 7.5 hours
Headcount freed up ($5.8 \div 7.5$)	≈ 0.8 FTE
Per year ($\approx 250 \text{ working days} \times 5.8 \text{ hrs}$)	≈ 1,450 hours

In plain terms: the better part of one full-time person, who would otherwise do nothing but process cancellations, is freed up entirely, roughly 1,450 hours a year handed back to higher-value work.

Payback on the build

The build was a one-time effort of about 10 hours, including setting up the custom integration behind Eloquens® AI. Against a saving of nearly 6 hours a day:

One-time build effort	≈ 10 hours
Daily saving	≈ 5.8 hours
Break-even (10 ÷ 5.8)	< 2 working days

A one-time build of 10 hours now saves 29 hours every week, the equivalent of getting back almost a full workweek of one person's time, every week.

Could this be you? Sizing it to your business

The economics scale with volume. Around 70 opt-out requests a day points to a sizable installed base with predictable, recurring renewal churn, the kind of book where a steady trickle of routine admin quietly consumes a person's entire week. If your team recognizes any of the following, the same pattern likely applies to you:

- A subscription or license base large enough to generate a daily stream of the same routine request (renewals, cancellations, address changes, and simple status updates).
- Requests that arrive as free-text email and currently wait for a human to be available.
- A clear, rule-based eligibility check that should be applied identically every time (e.g. a notice-period deadline).
- Two or three systems that must stay in sync, a portal, a tracking sheet, and a CRM.

As a rule of thumb: every ~70 daily requests at ~5 minutes each is roughly 0.8 of a full-time role. At ~35/day it is about half a role; at ~140/day it is a role and a half. The break-even stays measured in days, not months, because the build cost is fixed while the saving compounds with every request.

Built to fail safe. An uncertain or failed request is always handed to a person rather than acted on incorrectly, while the customer is simply told the matter is being looked into. Speed without that guardrail would not be worth much; here you get both.

See what six hours a day looks like back on your calendar.

If routine email requests are quietly consuming a role's worth of time on your team, put Eloquens® AI to work on them free for 30 days, no commitment.

Start your trial at eloquens.ai.