ALLIED BizTech SOLUTIONS

FIELD GUIDE · 2026

Rebuild Intercom yourself.
The plan we follow.

For support teams of 30 to 150 agents who want to own their chat widget, inbox, knowledge base and AI answers, and stop paying per seat and per resolution.

- → Is it worth it, at your size
- → What to rebuild and what to drop
- → The architecture, in three layers
- → Migration and go-live checklists
- → The data model
- → A week-by-week build plan
- → AI answers that hold up
- → Where teams get stuck

Intercom charges three ways. *A rebuild charges one.*

Intercom bills per seat (everyone in the inbox), per resolution (Fin, its AI agent) and per proactive-support contact. Each grows on its own. A rebuild replaces all three with a flat cloud bill and AI usage at your provider's token price.

TEAM	INTERCOM A YEAR	REBUILT: YEAR ONE / AFTER	SAVED IN YEAR ONE
10 agents · Essential, no Fin	~$4,680	~$21,900 / $2,000	Stay on Intercom
30 agents · Advanced + Fin	~$62,000	~$24,700 / $4,800	~$37,300
75 agents · Expert + Fin Pro	~$160,000+	~$52,000 / $14,000	~$108,000
150 agents · Proactive + Fin Premium	~$340,000+	~$72,000 / $34,000	~$268,000

Intercom list prices, May 2026. "Rebuilt" uses Anthropic Claude for AI answers, AWS for the chat, inbox and knowledge base, and our fixed build price: $19,900 up to 50 agents, $38,000 from 50 to 150. If your own team builds it, put your engineering cost in that place.

THE RULE OF THUMB

Below about 30 agents, stay on Intercom: the build does not pay back. **At 30 agents, the rebuild saves ~$37K in year one and ~$57K a year after.** From 75 agents, year one alone saves more than $100K.

Who you need

- **A senior full-stack engineer** at home with Postgres, WebSockets and an LLM API. Our team ships this in 5–7 weeks; one engineer working between other priorities should plan for longer.
- **Your support lead**, two hours a week: the workflows that matter, the five pilot agents, the go-live call.
- **An owner after launch**: hosting, monitoring, the monthly model and knowledge-base review, and on-call.

02 · SCOPE

Rebuild the 14 features in daily use. *Drop the rest.*

Intercom ships hundreds of features. A support team uses about 14 every day. Those are the project; the long tail is where rebuilds stall.

REBUILD ONE FOR ONE

- **Chat widget** on your domain: drop-in script, your brand, mobile-ready
- **Agent inbox**: priority queues, auto-routing, team workspaces
- **AI answers** (the Fin replacement): drafted from your knowledge base, with citations
- **Knowledge base**, public and private, with versions
- **Macros and saved replies**, per team, with variables
- **Customer sidebar** from your CRM or database
- **SLA tracking** and breach alerts by plan, priority and customer tier
- **Reporting**: CSAT, first response time, resolution time, agent performance
- **Email-to-conversation** and **automation rules**: auto-tag, auto-assign, auto-respond

REBUILD, SIMPLER

- **Series campaigns** become a simple workflow editor, same outcomes
- **Surveys**: Typeform directly, or a phase-two module
- **Phone and WhatsApp**: Twilio on your own account

DROP ON PURPOSE

- **The visual workflow builder's long tail**: most teams run 8–12 real workflows
- **Product tours**: Userflow, Pendo or Appcues do this better
- **ML-targeted proactive messages**: rule-based triggers (page views, cart value) instead
- **The 300+ integration directory**: rebuild your top five
- **Academy and community**: record training for your own workflow

ADD WHAT PER-SEAT PRICING DISCOURAGES

- Audit logs, custom routing, separate brands in one inbox
- New agents at no seat fee

Three layers. *All standard parts.*

None of this is exotic: a chat widget is HTML and a WebSocket, an inbox is an app on a customer database, and AI answers are search plus a model call. Pick well-documented parts; each has a managed option and an open-source one.

LAYER 1 · CUSTOMERS SEE

Widget and help centre

- Chat widget (`embed.js` on your domain)
- Help centre: search and browse
- Help articles with screencasts
- Proactive messages (rules plus the model)
- Email-to-conversation routing

LAYER 2 · AGENTS USE

The inbox

- Queues, priority and assignment
- Macros and saved replies
- Customer sidebar, synced from your CRM
- SLA timers and breach alerts
- AI draft panel
- Escalation routing (rules plus sentiment)

LAYER 3 · YOU RUN

The infrastructure

- **Postgres**: customers, conversations
- **WebSocket gateway** on AWS or Cloudflare
- **Meilisearch**: the knowledge-base index
- **S3**: attachments and recordings
- **Anthropic or OpenAI** on your account
- **Postmark inbound**: email to conversation

Decisions to make in week one

DECISION	OUR DEFAULT	WHY
Where it runs	Your AWS account, or Cloudflare plus your origin	Data residency stays yours; one fewer hop than a SaaS widget
Agent sign-in	Your SSO from day one	Offboarding happens where it already does
Search	Meilisearch, keyword plus vector	Serves both the help centre and AI answers
Model	One provider, behind a small adapter	Swap models without touching the inbox
CRM sync	Read-only sidebar first, write-back later	Agents get context in week three without a sync project

Compliance: on your own AWS account you inherit AWS's SOC 2, ISO 27001 and HIPAA-eligible posture (with a BAA). Write the control documentation as you build, not after.

Nine tables *carry the whole product.*

Model these before writing any screen. Keep Intercom's ID on every imported row, so the import can run twice and any thread traces back to its source.

TABLE	KEY COLUMNS
`customers`	id, external_id, email, name, company_id, plan, attributes (jsonb), created_at
`conversations`	id, external_id, customer_id, channel, status (open, snoozed, closed), priority, assignee_id, team_id, sla_policy_id, first_response_due_at, next_response_due_at, created_at, closed_at
`messages`	id, external_id, conversation_id, author_type (customer, agent, bot, system), author_id, body, attachments (jsonb), is_internal_note, created_at
`agents`, `teams`	id, external_id, name, email, role, team memberships, working hours
`assignment_log`	conversation_id, from_agent, to_agent, by (rule or person), at
`articles`	id, external_id, collection, title, body (markdown), visibility (public, private), version, updated_by, updated_at; old versions in `article_versions`
`macros`	id, team_id, title, body with variables, actions (jsonb: tag, assign, close)
`sla_policies`	id, name, applies_to (plan, priority, tier), first_response_minutes, next_response_minutes, business_hours
`ai_answers`	id, conversation_id, question, answer, citations (jsonb), model, tokens_in, tokens_out, cost, outcome (resolved, handed off), created_at

TWO RULES THAT SAVE A REBUILD

Log every AI answer with its cost and outcome: it is how you prove the saving and find the articles to fix. **Write an event for every change to a conversation** (status, assignee, priority): reporting and audit logs both read from it.

Seven weeks, *pilot included.*

This is the order we build in. Each week ends with something your support lead can try, and the last two are a pilot with five real agents.

WEEK	BUILD	DONE WHEN
1	**Export and model.** The nine tables. Export from Intercom's API into staging: contacts and companies, conversations with every part, help-centre articles, admins and teams, tags. Agent sign-in.	Counts in staging match Intercom; 50 threads spot-checked
2	**Widget and real time.** `embed.js` on your domain, the WebSocket gateway, typing indicators, attachments to S3, email-to-conversation.	The widget loads in under 100 ms and messages arrive live
3	**The inbox.** Queues, assignment rules, macros, internal notes, the customer sidebar from your CRM or database.	An agent can work a full shift without opening Intercom
4	**Knowledge base and AI answers.** Editor with versions, the Meilisearch index, the help centre, AI answers with citations and hand-off (next page).	AI answers pass your test set
5	**SLAs, routing, reporting.** SLA timers and breach alerts, escalation rules, CSAT, dashboards. Load-test the widget.	Supervisors see every number they track today
6–7	**Pilot and cutover.** Five agents for two weeks, a daily fix list, a delta export from Intercom, then the widget snippet swapped and the whole team moved.	The pilot agents choose it over Intercom

AFTER CUTOVER

Keep Intercom **read-only until your renewal date**, then cancel. Old threads stay searchable in both places while the team settles in, and nobody is in a hurry.

Answers from your articles, *or a hand-off.*

Fin's design is search over your knowledge base, a model call that drafts the answer, and a citation back to the article. Build the same thing on your own AI account and you pay tokens, not a fee per resolution. In our deployments this design reaches about 85% of Fin's resolution rate at about 15% of the cost; the quality comes from the knowledge base, which is the same one.

1. **Retrieve.** Split articles at their headings and index the passages in Meilisearch, keyword and vector together. Fetch the best few passages for each question.
2. **Draft.** Tell the model to answer only from those passages and to cite them. When nothing relevant comes back, hand the conversation to an agent instead of guessing.
3. **Test before launch.** Take 200 resolved conversations from your Intercom export, replay each customer's first message and mark the answer right, wrong or handed off. Every wrong answer points at an article to fix. Launch when your support lead signs off on the results.
4. **Measure.** Log every answer with its tokens, cost and outcome. Watch the resolution rate, the hand-off rate and the cost per answer.
5. **Review monthly.** Read the failed answers, fix the articles behind them, re-index, and check the model version.

The hand-off is the feature

A customer who gets a confident wrong answer is worse off than one who waits two minutes for a person. Make the hand-off fast and visible: the agent sees the question, the passages the model found and the draft it would have sent.

Phone, WhatsApp and Slack

Twilio on your own account covers voice and WhatsApp; Slack's API covers the Slack surfaces Intercom charges for. Add them after cutover, once the inbox is settled.

Move it, launch it, *run it.*

Migration

- ☐ Export twice: everything in week one, the changes since at cutover
- ☐ Keep Intercom's ID on every row
- ☐ Recreate teams and tags before importing conversations
- ☐ Keep thread order, authors, internal notes and assignment history
- ☐ Copy attachments to S3 and rewrite their links
- ☐ Trim macros to the ones used in the last 90 days
- ☐ Redirect every help-centre URL (301), so old links and search rankings survive
- ☐ Check counts, then read 50 threads end to end

Go-live

- ☐ The widget snippet swapped on every site and app
- ☐ Support email forwarded to the new inbound address
- ☐ Agents signing in through your SSO
- ☐ Backups proven by a restore
- ☐ Alerts on widget errors, queue depth and SLA breaches
- ☐ Rollback ready: the old snippet one deploy away

Running it

- ☐ **Always on:** uptime and error alerts, security patches
- ☐ **Weekly:** SLA and CSAT report to the support lead
- ☐ **Monthly:** AI answer review, knowledge-base fixes, model version check
- ☐ **Quarterly:** cost review, AI usage against the Intercom bill you replaced
- ☐ **Yearly:** a restore drill and an access review

Who owns it

Name one person before launch. The product is small, but it is your customers' front door: someone has to answer when the widget errors at 2 a.m. or a model update changes the answers.

Five ways a rebuild *stalls.*

1. The cheap build

A $10K freelance build gets you a chat widget. It does not get you assignment logic, SLAs, AI answers, the knowledge base or CRM sync, which are the parts Intercom actually charges for.

2. Self-hosting an open-source inbox and calling it done

Chatwoot is genuinely good. Setup, maintenance, AI integration and your own workflows are still 3–5 weeks of senior engineering, and the running of it is yours from then on.

3. Rebuilding everything

The long tail of Intercom's features is where schedules slip. Rebuild the 14 in daily use and keep a phase-two list for the rest; most of it never gets asked for.

4. Skipping the pilot

Five agents for two weeks find what a test plan misses: the macro nobody listed, the report the head of support reads every Monday, the customer tier with its own SLA.

5. Switching off Intercom on day one

Keep it read-only until renewal. It costs nothing extra and removes all the pressure from cutover week.

And one that is not a mistake: negotiating

Negotiation services typically take 15–25% off an Intercom annual commitment. Do it if the rebuild is a year away; it captures a fraction of what owning the stack does.

RATHER HAVE IT BUILT?

We build this exact plan. *Fixed price.*

The same scope, architecture and seven weeks, delivered by a senior team that has shipped 250+ products.

$19,900 fixed for 10–50 agents, $29,000–$38,000 from 50 to 150, agreed before we start

---

Live in 5–7 weeks, with a two-week pilot with five of your agents first

---

Migration in week one, conversations, customers, articles and macros with their history

---

90-day warranty, and the code is in your GitHub from day one

We can run it for $2,400 a year: hosting, monitoring, monthly model and knowledge-base updates

alliedbiz.tech/rebuild/intercom-alternative

Allied BizTech · 15+ years · ISO 9001:2015 · 250+ products shipped. Intercom is a trademark of its owner, named here for identification only; this guide is independent and not endorsed by Intercom. Prices as of May 2026.