ALLIED BizTech SOLUTIONS

FIELD GUIDE · 2026

Rebuild Zendesk yourself.
The plan we follow.

For support teams of 50 to 200 agents who want to own their ticket inbox, help centre, SLAs and AI answers, and stop paying per seat, per resolution and per add-on.

- → 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 and SLA routing
- → Where teams get stuck

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

Zendesk bills per seat (everyone in the inbox), per resolution (Zendesk AI) and per add-on (Sunshine, advanced analytics, side conversations). 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	ZENDESK A YEAR	REBUILT: YEAR ONE / AFTER	SAVED IN YEAR ONE
10 agents · Suite Team, no AI	~$6,600	~$21,900 / $2,000	Stay on Zendesk
25 agents · Suite Growth + Lite AI	~$32,400	~$23,900 / $4,000	~$8,500
50 agents · Suite Pro + AI	~$78,000	~$25,900 / $6,000	~$52,100
150 agents · Suite Enterprise + AI	~$310,000+	~$63,700 / $23,000	~$246,300
300 agents · Enterprise Plus + Advanced AI	~$700,000+	~$90,700 / $50,000	~$609,300

Zendesk Suite list prices, May 2026, Suite tiers plus AI add-ons. "Rebuilt" uses Anthropic Claude Sonnet 4.5 for AI resolutions, AWS for the ticket platform, and our fixed build price: $19,900 up to 75 agents, $40,700 (the top of the $29,000–$40,700 scale tier) above 75. If your own team builds it, put your engineering cost in that place.

THE RULE OF THUMB

Below 25 agents, stay on Zendesk: the build does not pay back. At 25 agents, year one saves ~$8K and ~$28K a year comes back after. **At 50 agents, the rebuild saves ~$52K in year one and ~$72K a year after.** From 150 agents, year one alone saves more than $240K.

Who you need

- **A senior full-stack engineer** at home with Postgres, WebSockets, inbound email 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 SLA policies, 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.*

Zendesk Suite Enterprise has hundreds of features. A real support org uses about 14 every day. Those are the project; the long tail is where rebuilds stall.

REBUILD ONE FOR ONE

- **Unified ticket inbox**: email, chat, web form, voice and WhatsApp in one queue
- **Assignment**: round-robin or skill-based routing, team queues, status workflows
- **SLA tracking** and breach alerts: multi-policy, business-hour aware
- **Macros and saved replies**, team-shared, with variables
- **Automations**: auto-tag, auto-assign, business-hours triggers
- **AI bot and draft replies**: drafted from your knowledge base, with citations
- **Knowledge base**, public and private, with versions, localisation and SEO
- **Help centre**: customer login, ticket history, search
- **Customer sidebar**: plan, recent orders, custom fields, live from your database
- **Side conversations** and internal notes
- **Reporting**: CSAT, first response time, resolution time, volume, agent performance

REBUILD, DIFFERENTLY

- **Voice, WhatsApp and SMS**: Twilio on your own account, about 85% cheaper than Zendesk Talk
- **Workforce management**: Assembled or Tymeshift directly

DROP ON PURPOSE

- **Sunshine custom objects**: not needed for support; a phase-two module if you use them
- **The 1,400+ Marketplace apps**: your team's real list is 5–8; rebuild the top 3–5
- **Explore**: Metabase on your warehouse instead
- **Sell and Sunshine CDP**: different products, different projects

ADD WHAT PER-SEAT PRICING DISCOURAGES

- New agents at no seat fee
- AI at token price, with no fee per resolution
- SQL access to every ticket, with no API rate limits

Three layers. *All standard parts.*

None of this is exotic: a ticket is a row in Postgres, a channel is a way for that row to arrive, 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

Help centre and widgets

- Knowledge base, public and restricted
- Help articles with screencasts
- Chat widget on your domain
- Contact form and ticket submission
- Self-serve AI bot over your articles
- Status page link

LAYER 2 · AGENTS USE

The workspace

- One inbox: email, chat, form
- Macros, saved replies, signatures
- SLA timers and breach alerts
- AI draft-reply panel and sentiment
- Side conversations, internal notes
- Reporting, agent performance, CSAT

LAYER 3 · YOU RUN

The infrastructure

- **Postgres**: tickets, customers
- **WebSocket gateway**: live chat
- **Postmark inbound**: email to ticket
- **Anthropic or OpenAI** on your account
- **Twilio**: voice, WhatsApp, SMS
- **S3**: attachments, recordings

Decisions to make in week one

DECISION	OUR DEFAULT	WHY
Where it runs	Your AWS account	Tickets and customer data never leave it
Agent sign-in	Your SSO from day one	Offboarding happens where it already does
Search	Full-text plus vector, one index	Serves the help centre and the AI bot alike
Model	One provider, behind a small adapter	Swap models without touching the inbox
Customer sidebar	Read live from your database or CRM	Agents get context in week three without a sync project
Agent screens	Zendesk's layout, macro syntax and SLA pills	Most teams are productive within a week

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

Fifteen tables *carry the whole product.*

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

TABLE	KEY COLUMNS
`customers`, `organizations`	id, external_id, email, name, organization_id, plan, custom_fields (jsonb), created_at
`tickets`	id, external_id, requester_id, channel, status (new, open, pending, on hold, solved, closed), priority, group_id, assignee_id, tags, custom_fields (jsonb), created_at, solved_at
`comments`	id, external_id, ticket_id, author_id, body, is_public, attachments (jsonb), side_conversation_id, created_at
`agents`, `groups`	id, external_id, name, email, role, group memberships, skills, schedule_id, max_open_tickets
`ticket_events`	ticket_id, field (status, assignee, group, priority), old_value, new_value, by (rule, agent, system), at
`macros`, `triggers`	id, external_id, group_id, title, conditions (jsonb), actions (jsonb: reply with variables, tag, assign, set status), last_used_at
`sla_policies`	id, name, position, conditions (priority, plan, organization), targets per priority in minutes (first reply, next reply, resolution), schedule_id
`schedules`	id, time_zone, weekly hours, holidays
`ticket_sla_targets`	ticket_id, policy_id, metric, due_at, paused_at, achieved_at, breached_at
`articles`	id, external_id, section, locale, title, body (markdown), visibility, version, updated_at; old versions in `article_versions`
`ai_answers`	id, ticket_id, question, answer, citations (jsonb), model, tokens_in, tokens_out, cost, outcome (resolved, handed off)

TWO RULES THAT SAVE A REBUILD

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

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 fifteen tables. Export from Zendesk's API into staging: tickets with every comment, users and organizations, help-centre articles, macros, triggers, groups and team config. Agent sign-in.	Counts in staging match Zendesk; 50 tickets spot-checked
2	**Channels.** Postmark inbound for email, the chat widget and WebSocket gateway, the web form, attachments to S3, outbound replies threaded on the original email.	An email, a chat and a form each land as a ticket in seconds
3	**The inbox.** Views and queues, round-robin and skill-based assignment, macros, internal notes, side conversations, the customer sidebar.	An agent can work a full shift without opening Zendesk
4	**Help centre and AI answers.** Editor with versions and locales, search, the help centre on your domain, the AI bot and draft replies with citations and hand-off.	AI answers pass your test set
5	**SLAs, triggers, reporting.** SLA policies on business hours, breach alerts, triggers and automations, CSAT, dashboards.	SLA timers match Zendesk's on a replayed week of tickets
6–7	**Pilot and cutover.** Five agents for two weeks, a daily fix list, a delta export from Zendesk, then support email forwarded, the widget swapped, help-centre URLs redirected and the whole team moved.	The pilot agents choose it over Zendesk

AFTER CUTOVER

Keep Zendesk **read-only until your renewal date**, then cancel. Old tickets stay searchable in both places while the team settles in. Record a 30-minute walkthrough of your own workflow; with the same inbox layout and macro syntax, most teams are productive within a week.

Answer from your articles. *Route against the clock.*

Zendesk AI is search over your knowledge base, a model call that drafts the answer, and a citation back to the article. Build the same on your own AI account and you pay tokens, not a fee per resolution. In our deployments it reaches about 85% of Zendesk AI'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, keyword and vector together. Fetch the best few for each question.
2. **Draft or hand off.** The model answers only from those passages and cites them. When nothing relevant comes back, the ticket goes to an agent with the passages and the draft attached.
3. **Test before launch.** Replay the first message of 200 solved tickets from your export; mark each answer right, wrong or handed off. Every wrong answer points at an article to fix.
4. **Review monthly.** Read the failed answers, fix the articles, re-index, check the model version.

The SLA clock

Match each new ticket to the first policy whose conditions fit (priority, plan, organization) and store a due time for each target: first reply, next reply, resolution. The clock runs only inside the policy's schedule, in its time zone, and skips holidays. Pending pauses it. Recompute the due time from the event log on every change, and fire the breach warning before the deadline, not at it.

Test the edges: a ticket at 17:55 on a Friday, a holiday, a clock change, a priority raised halfway through.

Routing that holds

Assign in one database transaction, locking the ticket row, so two agents never take the same ticket. Match skills first, then pick the online agent with the fewest open tickets under their cap. Move tickets at risk of breach to the front of the queue, and log every assignment for the report.

Move it, launch it, *run it.*

Migration

- ☐ Export twice: everything in week one, the changes since at cutover
- ☐ Keep Zendesk's ID on every row
- ☐ Recreate groups, tags and custom fields before importing tickets
- ☐ Keep comment order, authors, public and private notes, and assignment history
- ☐ Copy attachments to S3 and rewrite their links
- ☐ Trim your 50–200 macros and triggers 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 tickets end to end

Go-live

- ☐ Support email forwarded to the new inbound address
- ☐ The widget and web forms swapped on every site and app
- ☐ Agents signing in through your SSO
- ☐ SLA policies and business hours checked against Zendesk's
- ☐ Backups proven by a restore
- ☐ Alerts on inbound email failures, queue depth and SLA breaches
- ☐ Rollback ready: email forwarding one change 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 Zendesk 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 inbound email stops at 2 a.m. or a model update changes the answers.

Six ways a rebuild *stalls.*

1. The cheap build

A $15K freelance build gets you a ticket form. Routing, SLAs, AI, the knowledge base and audit logs stay on you, and those are the parts Zendesk actually charges for.

2. Self-hosting Chatwoot or FreeScout and calling it done

Chatwoot is genuinely good. Setup, 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. Switching brands, not the model

Freshdesk and Help Scout are 15–35% cheaper per seat today; their pricing tracks Zendesk's once you add the AI features your team will end up needing. Intercom has comparable seat economics and charges per resolution. Service Cloud usually costs more than Suite Enterprise at the same scale.

4. Rebuilding everything

Sunshine, Explore and the Marketplace are where schedules slip. Rebuild the 14 features in daily use and keep a phase-two list; most of it never gets asked for.

5. Skipping the pilot

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

6. Switching off Zendesk on day one

Keep it read-only until renewal. It costs nothing extra and takes the pressure off cutover week.

And one that is not a mistake: negotiating

Negotiation services typically take 15–25% off a Zendesk renewal. Do it if the rebuild is a year away; it captures less than a quarter 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 25–75 agents, $29,000–$40,700 from 75 to 200, agreed before we start

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

Migration in week one: tickets, customers, articles, macros and triggers, with full thread and assignment 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 knowledge-base and model updates

alliedbiz.tech/rebuild/zendesk-alternative

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