MINDTWIN

The renewal you lose is the one nobody saw coming.

A shared cognitive memory for Customer Success — it remembers every account across Slack, email, and calls, and tells you when a renewal is quietly going sideways.

Product overview · MindTwin for Customer Success teams · mindtwin.me

THE PROBLEM

Account memory walks out the door

Customer Success runs on context — and that context is scattered across Slack, Gmail, and Gong, locked in one CSM's head. When they leave, go on PTO, or hand off an account, it's gone.

1

A CSM leaves

The new rep spends a week reconstructing the account from scratch — on a renewal that may already be at risk.

2

Signals scatter

A pricing complaint on a call, a quiet champion, a recurring bug — no one connects them until the customer is gone.

3

RAG isn't enough

Search-over-Slack returns documents. It doesn't notice that what the customer says today contradicts what they said in March.

Result: preventable churn, slow ramp, and renewals decided by whoever happened to remember.

A REAL ACCOUNT STORY

Meet Acme.

$180k	30 days	Just left
annual contract	to renewal	the CSM who owned it

“Maria is picking up Acme cold, with the renewal weeks away. ***Here's what happens with MindTwin*** *versus what happens today.”*

The next three pages follow one account through the product — everything shown is actual output from a live workspace.

ACT 1 · KNOWLEDGE CONTINUITY

“Catch me up on Acme.”

One question. Months of Slack, email, and Gong assembled into a dated, cited brief in seconds — the new rep is productive on day one, not week two.

- ✓ **Slack** — #cs-renewals threads
- ✓ **Gmail** — customer email
- ✓ **Gong** — call transcripts

Notice the redacted email address in source [4] — PII never enters the memory vault.

ACTUAL OUTPUT

You: catch me up on the Acme account — where does the renewal stand?

The Acme account renewal is currently in jeopardy. As of June 20, 2026, the renewal is considered at risk due to unresponsiveness from both Dave, after a conversation about pricing, and the executive sponsor [1]. The renewal is due in 30 days from July 5, 2026, which makes the deadline early August [2]. Additionally, the current handler of the account is leaving their position this Friday, which adds urgency [2][3].

Sources

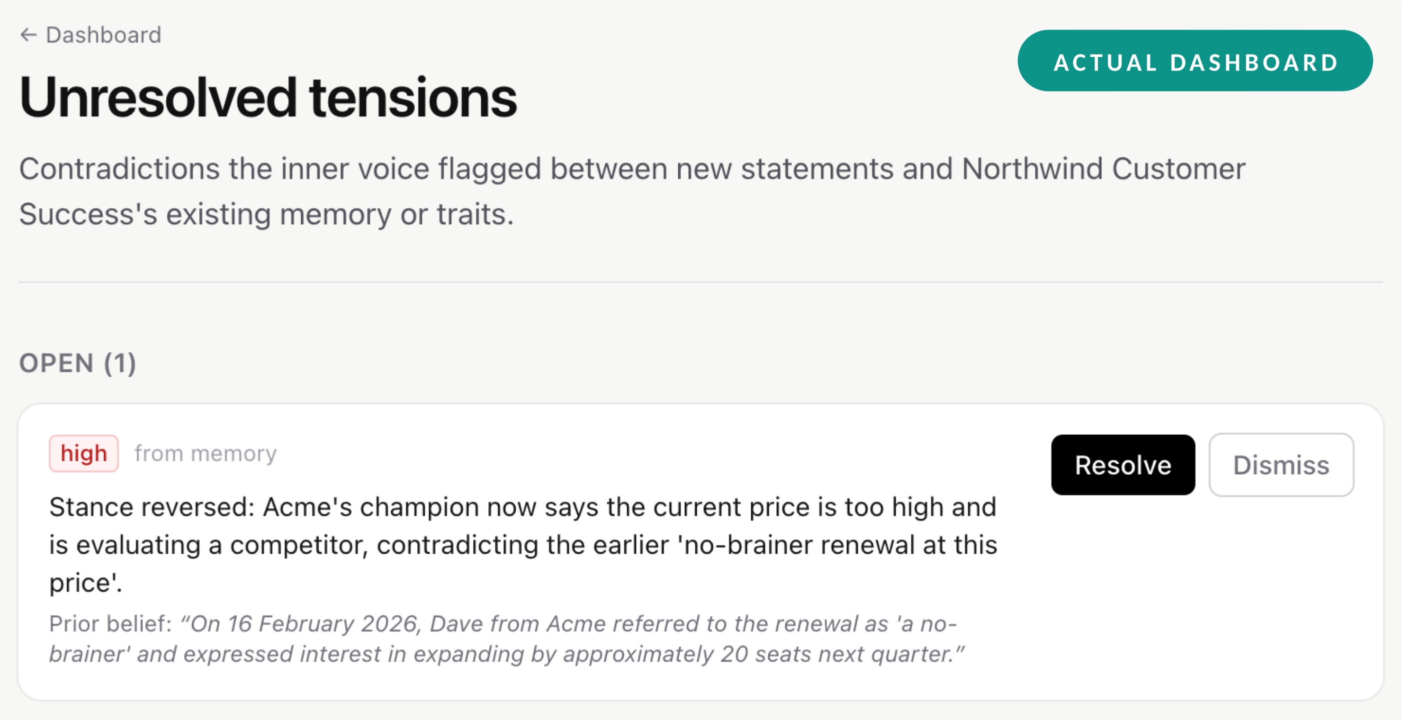
[1] Slack — As of 20 June 2026, the Acme renewal is considered at risk...
[2] Slack — The renewal for Acme is due in 30 days from 5 July 2026...
[3] Slack — Acme renews in 30 days and I'm leaving Friday...
[4] Email — Re: Acme renewal · From: [REDACTED:EMAIL] · Confirming...

ACT 2 · THE DIFFERENTIATOR

It notices the stance reversal

MindTwin doesn't just store both statements — its inner voice flags that they contradict each other, months apart, and raises the renewal risk before anyone asks.

FEB “Renewal is a no-brainer.” **APR** Export bug keeps breaking. **MAY** Finance squeezing SaaS spend. **JUN** “Evaluating a competitor.”



The tension MindTwin raised on its own — severity, the reversal, and the dated prior belief it contradicts.

It catches the costly mistake

The new rep, not knowing the history, drafts a cheerful renewal note. MindTwin vets it against the account before it's sent.

DRAFT EMAIL TO DAVE

“Hi Dave — thrilled you're as happy as ever. Locking in your renewal at current pricing; excited to expand!”

ACTUAL OUTPUT

aligned = false

Two contradictions with the account record (severity: high):

- Dave said pricing must be revisited before he can take a renewal to his CFO — the draft locks in current pricing.
- The renewal is flagged AT RISK — the champion went quiet after the pricing conversation — while the draft assumes he's “happy as ever”.

That's the unforced error that loses renewals — caught before it left the building. Works in Slack and inside Claude (MCP).

WHY SECURITY SIGNS OFF

Safe by construction

The capability that blocks most CS-tooling deals — handling sensitive customer data — is built into the ingestion path, not bolted on.

1	**PII scrubbed on ingest**	Emails and phone numbers are redacted before anything is stored — the raw values never enter the vault, so they can never be recalled back at anyone.
2	**Crisis content handled**	If a CSM ever DMs the bot something personal in a hard moment, it surfaces a helpline — it doesn't store the text and doesn't try to be a therapist.
3	**Tenant isolation by default**	Postgres row-level security means a team can only ever see its own memory — cross-tenant access is physically impossible, not just discouraged.

UNDER THE HOOD

How it works

Connect

Slack · Gmail · Gong

Real history ingested through one safety-checked path; every turn distilled into dated, standalone facts.

→

Cognitive memory

Salience · decay · re-rank

Not flat RAG — it weights what matters, lets go of what doesn't, and detects contradictions over time.

→

Act anywhere

Slack bot · MCP

Answers, risk signals, and check_alignment — in Slack and in Claude / Claude Code.

Multi-tenant from day one · cited answers · the cognitive layer is the moat

PROVEN, NOT PROMISED

Claims you can check yourself

1,540

questions

Evaluated on the full LoCoMo long-conversation memory benchmark — public methodology, reproducible with one command.

0

false flags

Safety gate across 5,882 emotionally varied conversation turns: zero crisis false-positives, zero recall cost.

174

automated tests

Including a mandatory cross-tenant isolation suite that gates every change.

MCP

live today

The memory tools — recall, tensions, check_alignment — already work inside Claude and Claude Code.

Benchmark results, harness, and methodology ship with the product — rerun them on your own machine.

DEFENSIBILITY

Why not just a copilot over Slack?

Anyone can do retrieval. The churn signal depends on capabilities generic copilots don't have.

Capability	Glean / Mem0 / generic copilot	MindTwin
Search across Slack/email/calls	Yes	Yes
Notices a stance reversal over time	No	Inner voice — flags it as risk
Vets an outgoing message before you send	No	check_alignment
PII scrub + crisis handling built in	No	Safety pipeline
Lets go of beliefs proven wrong	No	Processing-aware decay

THE ECONOMICS

It pays for itself on one saved renewal

Illustrative, 10-CSM team. Two value pools, both tied to numbers a CS leader already tracks.

~$360k

retained ARR / yr

churn caught earlier

~$116k

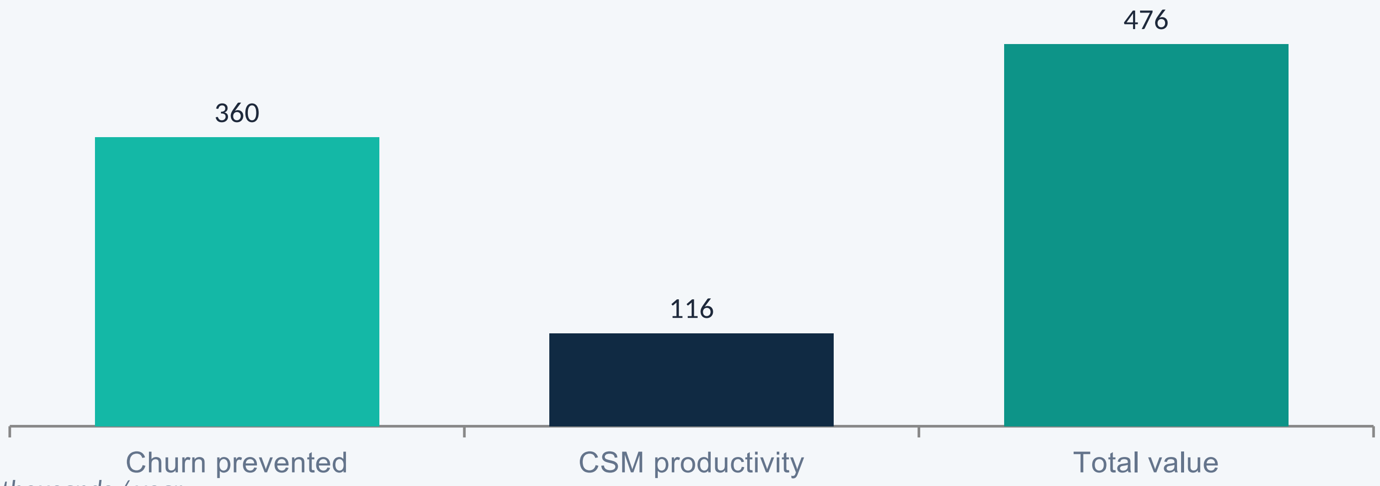
recovered capacity / yr

~4 hrs/wk per CSM

$180k

one saved Acme renewal

dwarfs the subscription



$ thousands / year

Prove it on your data.

A 6-week paid pilot with a success metric agreed up front — e.g. surface ≥3 at-risk renewals the team would have missed.

THE ASK

Run a paid pilot on one at-risk segment.

Six weeks · fixed fee, scoped on one call · one success metric agreed up front (e.g. surface ≥3 at-risk renewals you'd have missed) · you keep the findings either way.

1 Connect

Your real Slack, Gmail, and Gong for one pod of accounts.

2 Measure

Baseline week 0, the agreed metric at week 6.

3 Decide

Convert on the result — value proven on your own data.