

# There and back again

Or: "How not to get wrecked by your data"



**Nick Cannariato** (aka. *@birdcar*)

Co-founder @ Yetto

# The problems

- Support work is time sensitive, often measured in minutes or seconds.
- Support Professionals need both customer and company data in *most* support interactions.
- Support leadership (and leadership in general) needs both quantitative and qualitative data on the support team and their current level of operation.
- This data needs to be easy to get, secure to access, and preferably self-maintaining.
- **You do not have resources to spare.**

# Goals

- Help you understand your options, their tradeoffs, and their pitfalls.
- Help you communicate with your engineering resources about exactly what you need.
- Keep you from making painful mistakes and increasing toil.
- Rant a lil bit (this is a Yetto talk, after all).

# There

What to put in the help desk, how to get it there, and where to put it

What data is important to put in the tool?

Data from your sources of truth

a source of truth is a canonical depiction of reality inside your company

– **Brian Levine** (*@balevine*): "Building and Maintaining Sources of Truth"



## How do I get it into the tool?

- Time-based syncing.
- Event-based syncing.

## Time based syncing: Benefits

- Requires less (but not *no*) up-front engineering involvement.
- No seriously, *that's it*.

## Time based syncing: Issues

- Your team *cannot* trust the data in the tool is up to date when they're working tickets.
- The job will become become unmaintainable and unmanageable over time.



## Event based syncing: Benefits

- Customer data in the tool is up to date in real time (or close to it).
- Updates happen fast and are easier to troubleshoot when they go wrong.
- It's easy to add additional data later (after the initial engineering effort).
- No time is wasted getting a list of long dormant users (if your tool tracks customers as users).

## Event based syncing: Issues

- It requires more involvement and coordination with engineering up front.
- If updates fail intermittently, it can erode trust in the data.
- Adding data later doesn't automatically update existing records (you might have to run one-time jobs to backfill stuff later).
- Rate limiting.

# Verdict

-  Time-based syncing.
-  Event-based syncing.









Where do I put the data?

# Zendesk

Type	Field	Can be used to filter tickets?	Can be used in automations?
User	Tags	✓	✓
User	Custom Fields	?	?
Organization	Tags	✓	✓
Organization	Custom Fields	●	?
Ticket	Tags	✓	✓
Ticket	Custom Fields	✓	?
UI	Sidebar App	●	●



# Help Scout

Type	Field	Can be used to filter tickets?	Can be used in automations?
Customer	Customer Properties		
Conversation	Tags		
Conversation	Custom Properties		
UI	Sidebar App		

# Yetto

- Customer Connections.
  - We make a request to your app/admin tool when you open the conversation and/or a conversation event occurs.
  - You send us back the data you want for that conversation.
  - You can optionally also tell us how you want that data displayed in the sidebar.
  - No app needed, no syncing required.

+ Add Labels

## ASSIGNEES

No one has been assigned yet.

+ Assign to...







## CUSTOMER CONNECTION

Paid customer

true

10/19/2023, 3:53 PM

# Yetto

Type	Field	Can be used to filter tickets?	Can be used in automations?
Conversation	Label		
Data	Customer Connection		
Data	Switch		

# Verdict

- *If your help desk requires you to update a Customer object*
  - Use tags wherever possible.
  - Use custom fields only when necessary.
  - Use sidebar apps for stuff you can't securely store in the help desk.
- *If your help desk asks your app/admin tool for data.*
  - Work with engineering to set it up securely.
  - Get data from more than one source of truth in that request.

# Back

How do you get data out of the tool?

A mournful rant word on your help desk's built-in data exploration tool and its consequences

I just put all this data in my help desk, I can just build dashboards there now, right?

– Me, circa 2018

**NO**



# 100%

My level of certainty that you'll regret this.

## So, what *should* we do instead?

- Tell your organization you need a database for support data.
- Pipe all your help desk data into that database.
- Connect that database to whatever data exploration tool your organization already uses (e.g. Looker, Tableau, etc.).


# Why should I do that?

- Support data only makes sense in context, not in a silo.
- You don't have to know in advance what context is important.
- You don't have to duplicate every piece of customer data you might want into your help desk (your security team will thank you).
- You are no longer limited by what kind of reporting your help desk says you need.
- Data exploration will always be a secondary priority of your help desk.
- A purpose-built data exploration tool will enable you to be more accurate and tell better stories with data.

# Zendesk

- The mother of all time-based sync jobs (😓).
- A third party data-pipeline tool.
- Triggers/Webhooks? Kind of? But you should use the data pipeline tool (trust me).

## Zendesk: Setting this up

- Research the tool that has the connectors you need.
- Get your organization to pay for it.
- Configure it to connect to Zendesk and your newfangled support database.
-  Done.

# Help Scout

- Webhooks.
- A third party data-pipeline tool.

# Help Scout: Considerations

- Webhooks
  - You have complete control over how you save data **and** what data you save.
  - Cleaning up data or aggregating data in your database later is totally possible.
  - The ongoing cost is *way* cheaper, even if the initial cost is higher (i.e. engineering time).
  - Having complete control means having to deal with problems.
- A third party data-pipeline tool.
  - Someone else has to maintain it.
  - It grabs *everything*.
  - The cost might not matter to you.
  - You might not care about how much data it generates.

# Help Scout: Setting this up

1. Have engineering build an endpoint to catch Webhooks from Help Scout.
  - They'll need the webhook docs for Help Scout.
  - Tell them you want to subscribe to every event.
  - You can suggest functions (e.g. AWS Lambda, GCP Cloud Functions) to them as the place this endpoint could live.
  - They'll save the data you want from the webhooks in your newfangled support database.
2. Take the URL they give you, and put it in HelpScout (**Manage>Apps**) subscribed to every event.
  - You may not care about every event now/it may not apply to you, but send it anyways since the endpoint supports it.
  - You can always decide you care later if you have it in the database, you *can't* decide to add it to past events later.



# Yetto

- Webhooks.

# Yetto: Setting this up

1. Have engineering build an endpoint to catch Webhooks from Yetto.
  - They'll need the webhook docs for Yetto.
  - Tell them you want to subscribe to every event.
  - You can suggest functions (e.g. AWS Lambda, GCP Cloud Functions) to them as the place this endpoint could live.
  - They'll save the data you want from the webhooks in your newfangled support database.
2. Take the URL they give you, and put it in the webhook settings for your Yetto inbox, subscribed to every event.
  - You may not care about every event now/it may not apply to you, but send it anyways since the endpoint supports it.
  - You can always decide you care later if you have it in the database, you *can't* decide to add it to past events later.

# Verdict

- You should use the data exploration tool that your company uses for everything else, not the help desk's built-in tool.
- You need a database your company owns for support data.
- If your tool allows you to easily send webhooks for everything, you should send webhooks.
- Third party data-pipeline tools are awesome, but the tradeoff is less control and increased long-term storage costs.

## TL;DR

- There's no way to do this well without *some* engineering work.
- You **need** a support database outside your help desk.
- Data you need to understand belongs in a data tool, not trapped in a help desk.
- Event based updates are always better than time based updates (Webhooks, app events, etc.).

Q & A