

Project overview

Problem

Users need a single source of truth for the definitions of the events they are tracking as well as what properties and rules are attached to those events. A tracking plan acts as that source of truth. Because Rudderstack lives in the warehouse, the tracking plan can also be a governance tool, validating events against the documentation and keeping the data tracked clean and defined.

Process

- 2 day workshop with product/ENG
- Sat on customer success calls to understand asks/workflows
- Defined design stories and product requirements
- Designed a beta product, released to ~10 customers
- Worked with customer success to understand beta issues
- Created an updated product plan with new features and updated UI look and feel for general release

My role

- Ran the workshop
- Early product research - mapped current workflow, ran competitive analysis, diagrammed proposed architecture
- Wrote design stories
- Designed UX/UI
- Ideated features such as:
 - Separate data catalog - tracking plan architecture
 - Easy-implement code gen
 - Advance keyword integration
 - Property bundles and nested property bundles
 - Mini data catalog in edit flow
 - Activity viewer/version control

Results

- Multiple marketing initiatives based on this product
- Multiple customers closed or upgraded based on tracking plans functionality alone and growing
- Our largest and closest to enterprise client specifically requested tracking plans enablement

Feature launch: Tracking Plans for violation management

Written by



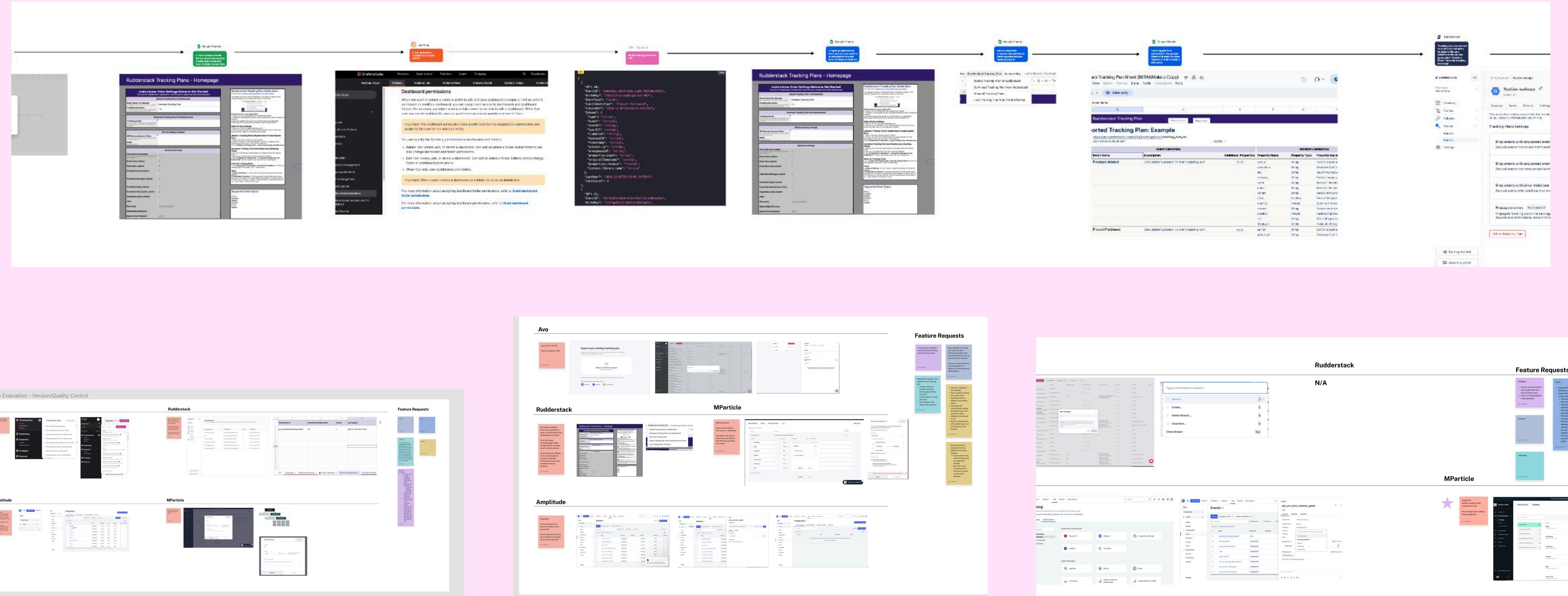
John Wessel

CTO, Mentor, & Data Consultant



Badri Veeraragavan

Director of Product



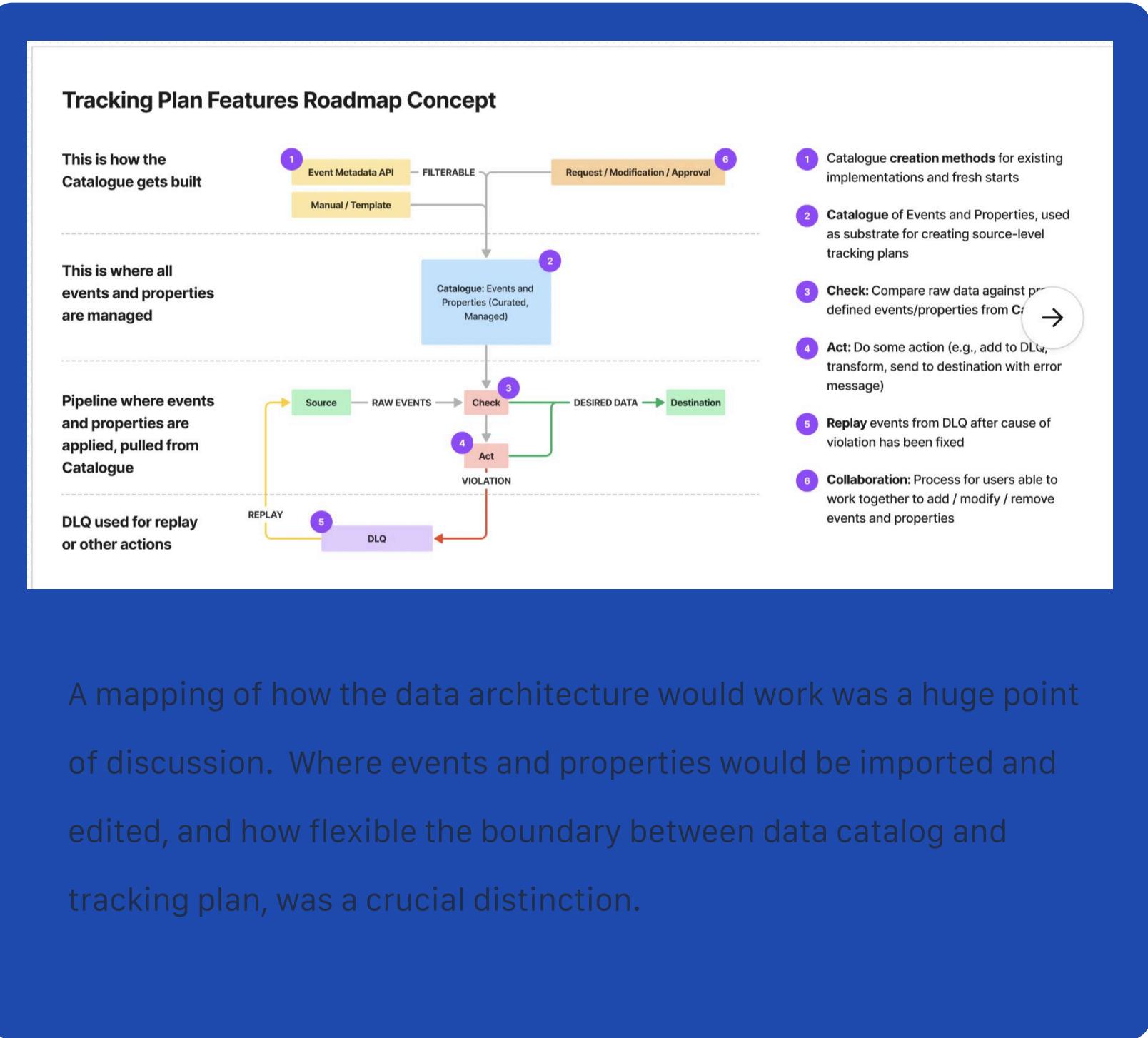
Getting Started

The first step for me was auditing our current tracking plan experience. This included mapping the workflow across all of the user touchpoints, across integrations, including those currently in rudderstack. Each box color in the diagram above represents a different webapp - meaning the experience was very disjointed. (top)

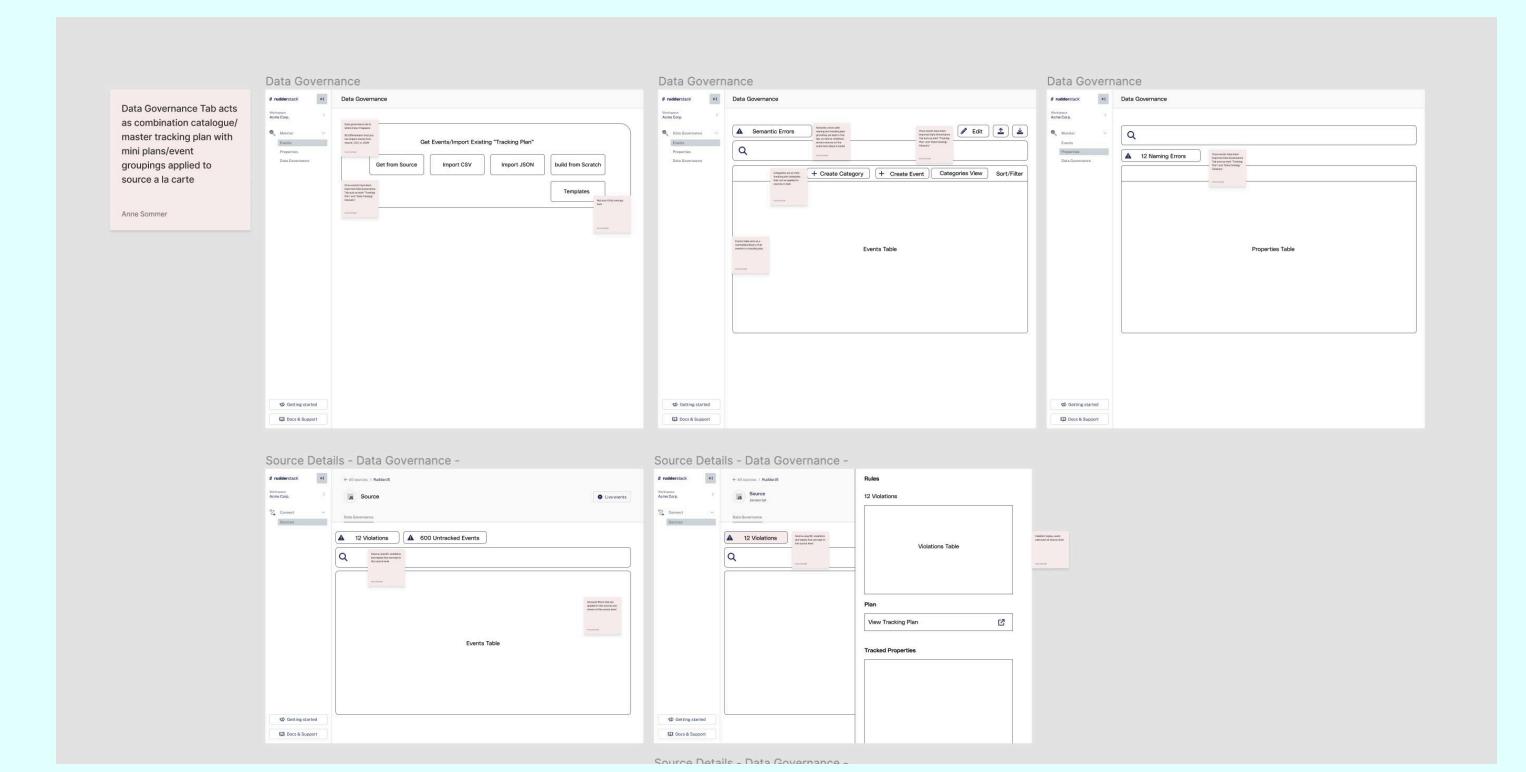
I looked at other tracking plan solutions and grouped screenshots on boards seperated by workflow stage and feature set. I added quotes from customer calls (each sticky-color is a different customer) to each feature grouping to guide thinking. (bottom)

Workshop Output

I met for a 2 day working session in New York with the head of product, engineering manager, and product manager to hash out what the MVP features were and basic architecture of the app. The major fundamental question was how the data catalog and tracking plan would bifurcate.



A mapping of how the data architecture would work was a huge point of discussion. Where events and properties would be imported and edited, and how flexible the boundary between data catalog and tracking plan, was a crucial distinction.



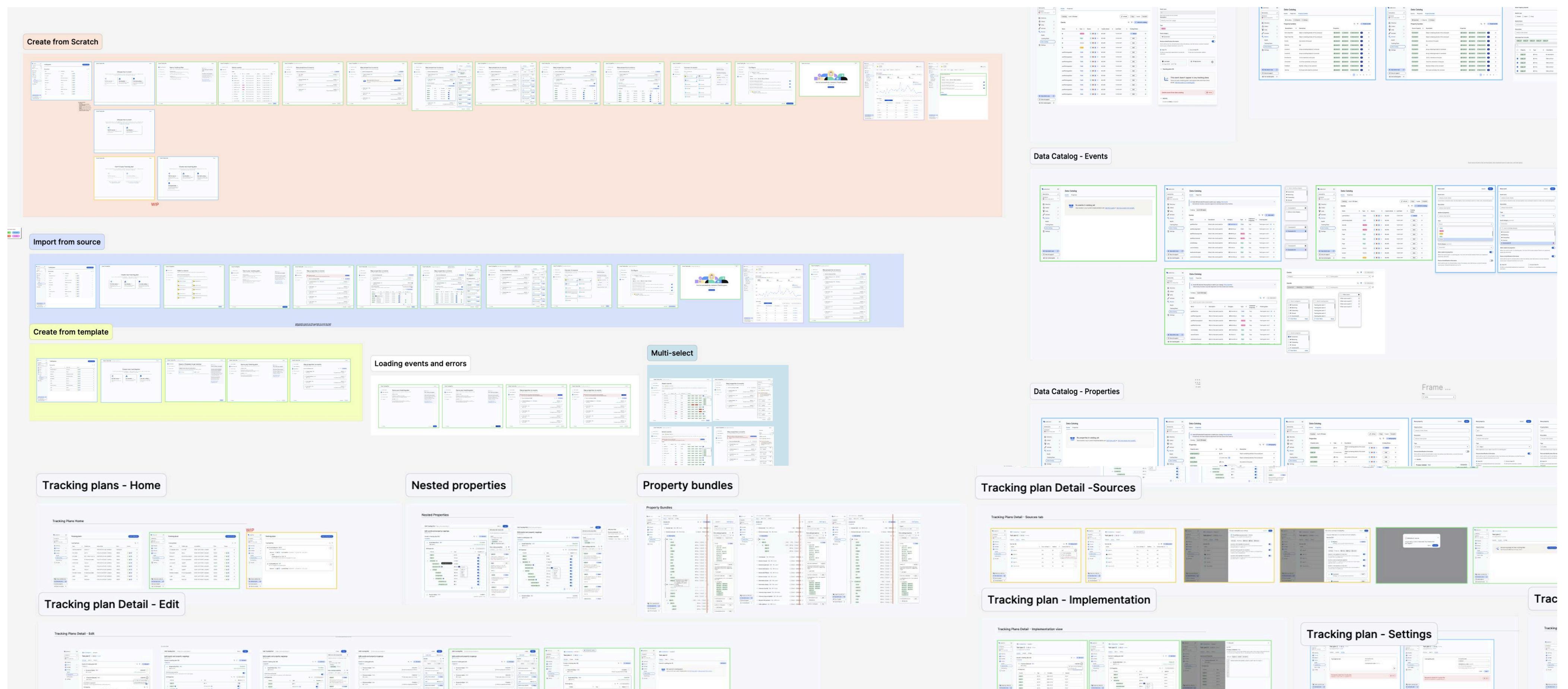
Very basic wireframes helped to sketch out thinking as it came up from each stage during these sessions and to get early feedback from engineering and product

UX Output

The result of the early design initiative was a massive new chunk of the product, including:

- Data Catalog (Events, Properties import/edit/creation)
- Tracking Plan Creation and edit flows
 - Separate import, template, and create from scratch flows
- Edit and connect tracking plan

Workshop to beta design delivery took about 9 weeks - from late October to the New Year



The screenshot shows the Rudderstack Tracking Plans interface. On the left, there's a sidebar with navigation links like 'Acme & Co.', 'Workspace', 'Directory', 'Connect', 'Enhance', 'Monitor', 'Health', 'Tracking Plans' (which is selected), 'Data Catalog', and 'Settings'. The main area is titled 'Test-plan-2' and shows 'Events in tracking plan (45)'. It lists two events: 'Products Added' (Track, Conversion, 10 properties) and 'Products Selected' (Track, Conversion, 5 properties). Below these, there's a section for 'Allow Unplanned properties' and a table for 'Properties'.

The screenshot shows the 'Edit Tracking Plan' interface for 'Test-plan-ecommerce 2'. The title bar includes 'Edit Tracking Plan | Test-plan-ecommerce 2', 'Cancel', and 'Save' buttons. The main area is titled 'Edit events and property mappings' and shows 'Events in tracking plan (45)'. It lists five 'Products Added' events, each with a 'Track' button and a note about 'Unplanned properties not allowed'. To the right, there are sections for 'Add event' and 'Add property' with dropdown menus for 'String' and 'Added'.

Beta Experience

Version one of the UI was a challenge because it had to be flexible enough to allow for extensive mappings of events to properties, but clear enough that a user could easily read and edit. I was also constantly aware that what we were competing with was code editors in the form of JSON-based tracking plans, and Excel.

I wanted to make it easy for users to be able to browse both their events in their tracking plan as well as their data catalog at the same time so that they could edit mappings easily. I was inspired by newer webapps like Linear and Notion and their snappy, responsive interfaces. I wanted users to be able to see everything dynamically to select what they need from a menu at right because users would be editing hundreds of events.

v1 Experience -Updated UI, Property Bundles

Once the beta went out to users and people started importing their own tracking plans in, it became clear our interface could be a lot more compressed and lightweight. I refined the UI to accommodate more events in the viewport and worked with another designer to update the look of our header, which then guided a design system update to be developed.

Property bundles and array and object properties were a huge point of discussion. Design and engineering ended up going around in circles on edge cases (such as a user wanting to add a bundle to a bundle within a nested array) that distracted from what the best design would be for the majority of users. We ended up with a solution that accommodated everyone.

Advanced Features

Advanced features, such as a detailed version control and activity log, code generator for implementing the tracking plan, and JSON-based rules called “advanced keywords” were determined as “nice to haves.” Because I had such a solid understanding of what the product was trying to achieve, I was able to spin these features up relatively quickly, from requirements --> design. I relied heavily on ChatGPT to help me generate advanced keyword examples and understand the functionality to ensure our UI-based solution would be as extensible as code.

The screenshot displays several panels from the Rudderstack interface, illustrating the implementation of tracking plans:

- Tracking Plans Detail - Implementation view:** Shows the main navigation bar and a sidebar with options like Directory, Connect, Enhance, Monitor, Health, and Tracking Plans.
- All Tracking Plans / Test-plan-2:** Overview of the tracking plan, showing sources (Acme & Co., Acme-corp-prod), events (Events (2)), and properties (5 Properties).
- Test-plan-2 / Events (2): Products Added:** Configuration for tracking products added, with options for Track and Conversion.
- Test-plan-2 / Events (2): Products Selected:** Configuration for tracking products selected, with options for Track and Conversion.
- All Tracking Plans / Test-plan-2 / View event code:** A modal showing the JSON event code for tracking products selected:

```
*JSON CODE PLACEHOLDER*
{
  "body": {
    "XML": 0,
    "FORM": {
      "data": [
        {"user_data": {"external_id": "}}
```
- All Tracking Plans / Test-plan-2 / Activity:** A timeline showing tracking plan updates, catalog updates, and events affected, with entries for Feb 24, 2024 and Feb 23, 2024.