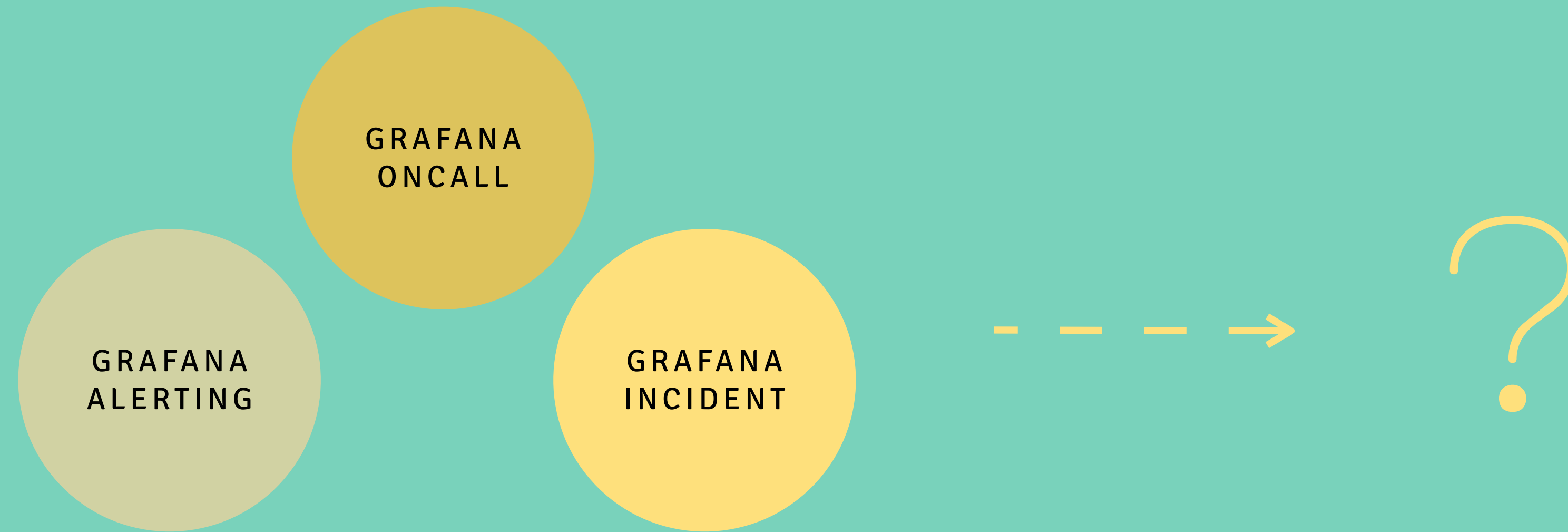


gops

unification project

UNIFYING THE USER EXPERIENCE FOR THREE SEPARATELY BUILT PRODUCTS
THAT ARE PART OF ONE WORKFLOW

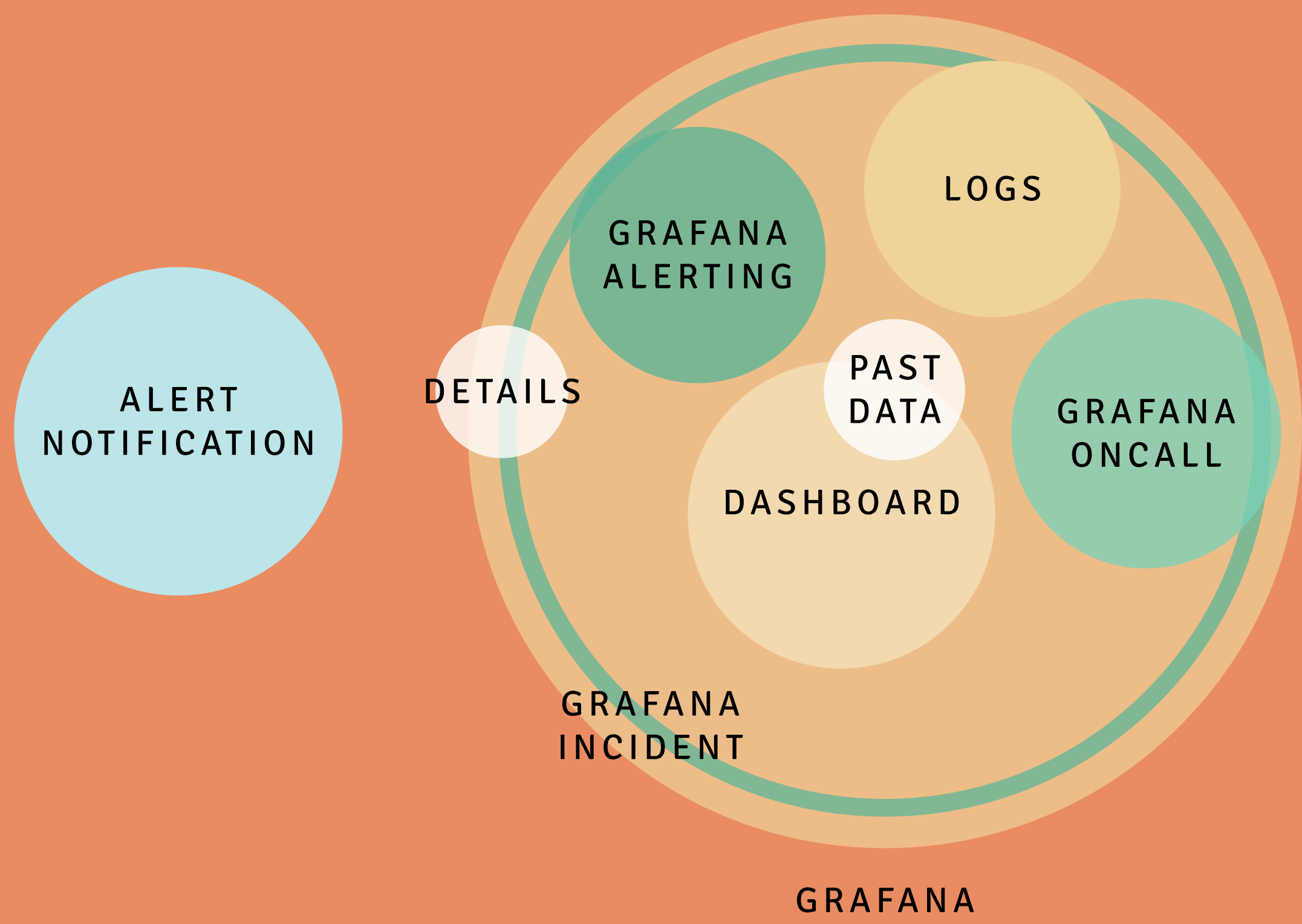


problem

THE BACKSTORY AND LEARNINGS THAT LED TO MY SOLUTION IDEAS

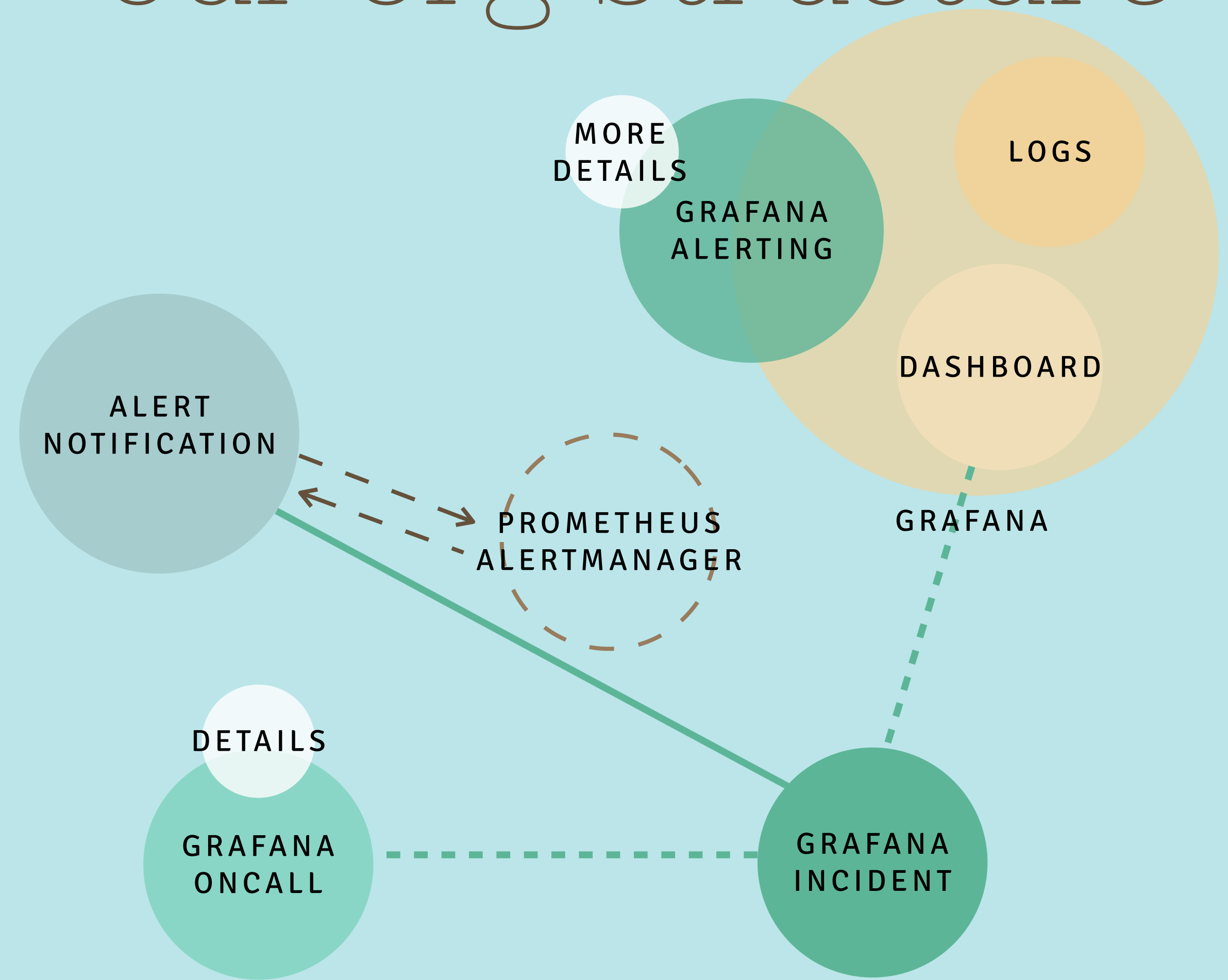
USER'S MENTAL MODEL

one tool, one unified workflow



REALITY

silos that match our org structure



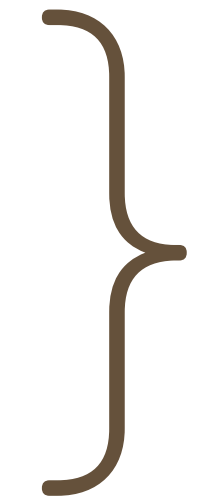
WHY THE SEPARATION?



Created 2016
Not expanded
on since



Created 2020
independently
of Grafana
Alerting



**Completely
rebuilt 2021**
due to tech debt
and customers
complaining about
“two Alerting tools”



Acquired 2021
as a big-tent
incident response
management tool



Created 2022
making incident
communication
and continuous
learning easier and
more efficient

OTHER TOOLS AND COMPETITORS

throwing data over
the fence is normal



“We don’t have control over what happens in the other tool and which tools users want to combine so we cannot optimize workflows”

BIG TENT VS 100% GRAFANA

do we need all
those fences
for the 100%
Grafana
workflow?

OBVIOUS MISSED POTENTIAL

why can't I?

- products are missing some obvious functionalities that connect IRM practices to long-standing Grafana features. Users routinely complain about them!
- while engineers can get caught up in the technical limitations of “why it works this way”, users might not care
- Even Grafanistas outside of the GOps bubble don't understand most of these concepts. How would external users understand them?

→ Declare incident
from a dashboard

→ Create alert rule
from Explore

→ Page the right engineer on-call
without setting up an OnCall alert group

→ opportunity to simplify
and be more consistent

TOIL AND INFORMATION OVERLOAD

why must I?

- users have to set up three products separately. This can for example mean installing a ChatOps integration 3 times.
- we don't provide instructions or guidance for setup and the general workings of the system
- users have to learn 3 separate interfaces and understand where each functionality lives, even if there are similar or overlapping concepts that appear across products
- we throw a lot of terms and options at the user without optimizing for their use cases and needs

Unify and merge
integrations

opportunity to simplify the
information architecture

Persona-based, unified
views of interfaces



challenges

that I have experienced in my work

PEOPLE

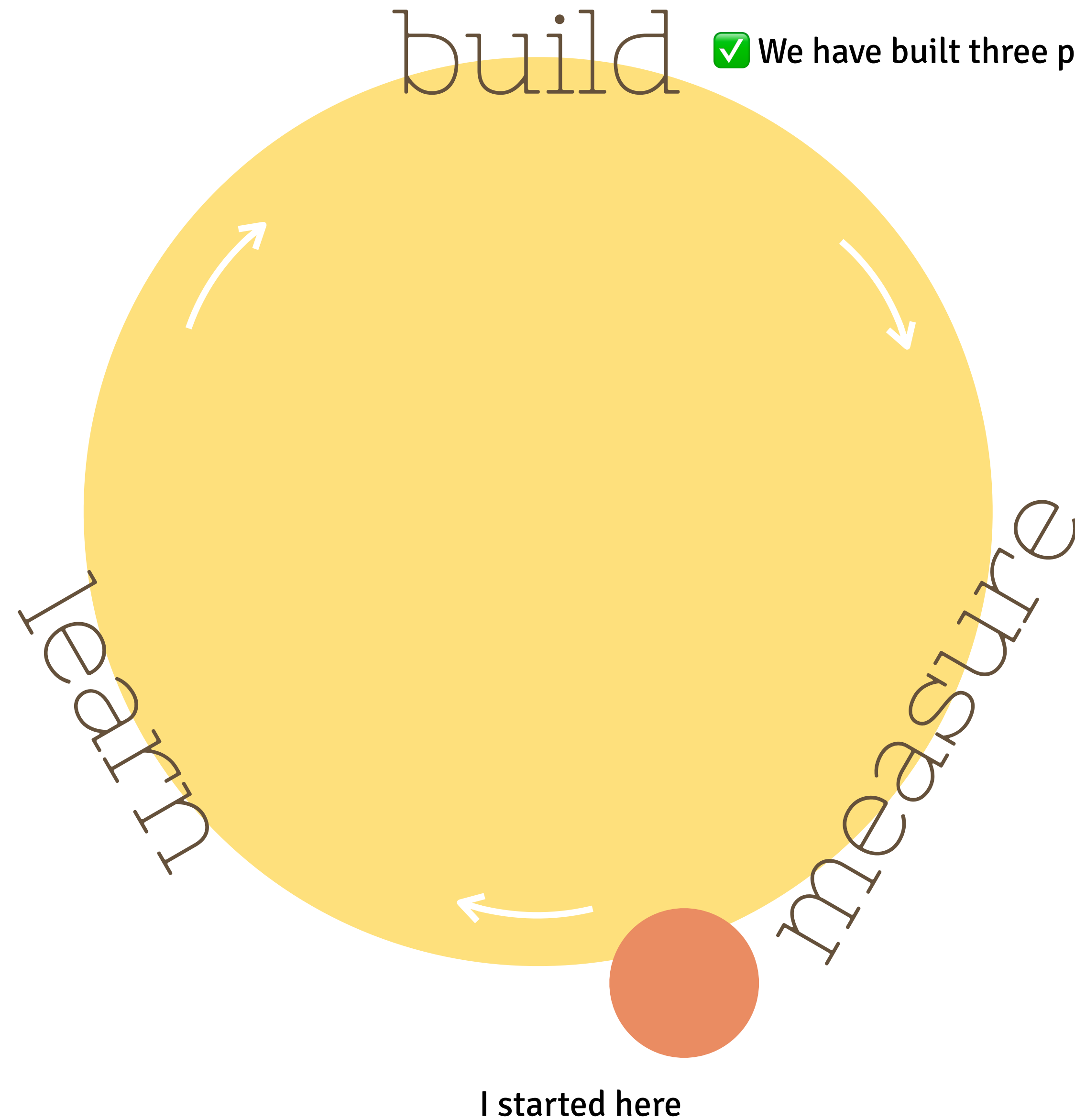
- Stakeholders in timezones from Singapore to California
- Everyone is opinionated and biased in favor of their product
- The rest of Grafana Labs have no clue what we are building or how these tools work. Even Grafanista engineers on-call struggle to use our tools

PRIORITIES

- Big technical differences between the Ops products that made it hard to unify them
- Customer escalations can get in the way of solving UX debt
- Engineers are caught up in the technical workings of our products, while users are probably not going to understand them and are focused on just accomplishing their tasks

BUSINESS

- Biggest customers have not even adopted Alerting, let alone Grafana's other IRM products
- The key customer that the unified product would address wasn't clear in the beginning of the project



✓ We have built three products

- 🤔 When I started, we weren't measuring. I had no answers to these questions:
- What are we measuring?
 - How can the team access the data?
 - How often are we reviewing it?
 - How data-driven are our decisions?

Now Marc is on it! 🚀

I FURTHER EXPLORED
THE PROBLEM

stakeholder interviews
15 stakeholders from 4 of 5
Ops products

key customer workshop
with Product and management
stakeholders to identify the target
audience for the GOps products

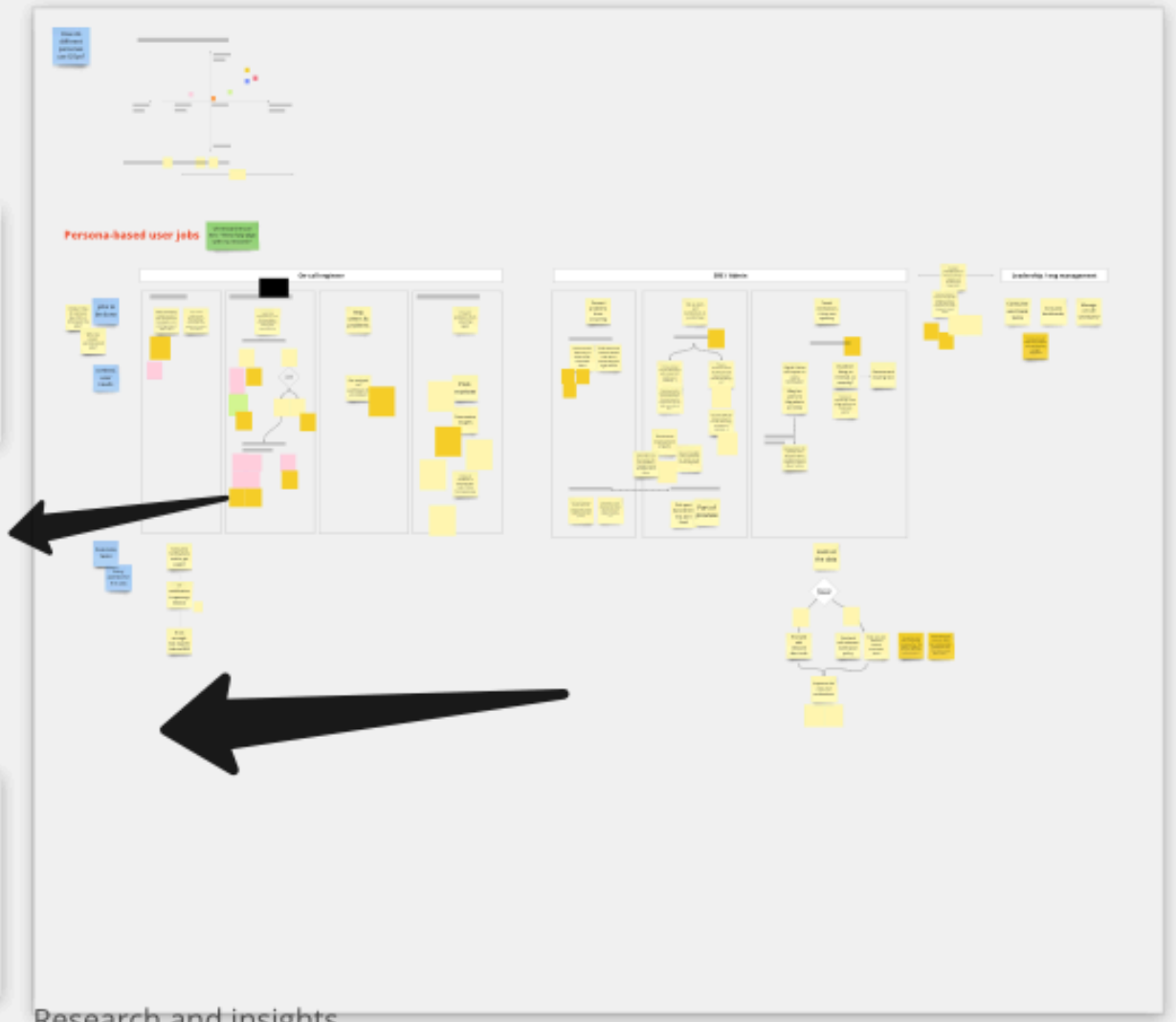
sensemaking
7 Grafanistas from different roles,
many of them not directly involved
in building GOps

alert creation usability tests
8 usability studies, 1:1 sessions with
external users, both OSS and Cloud

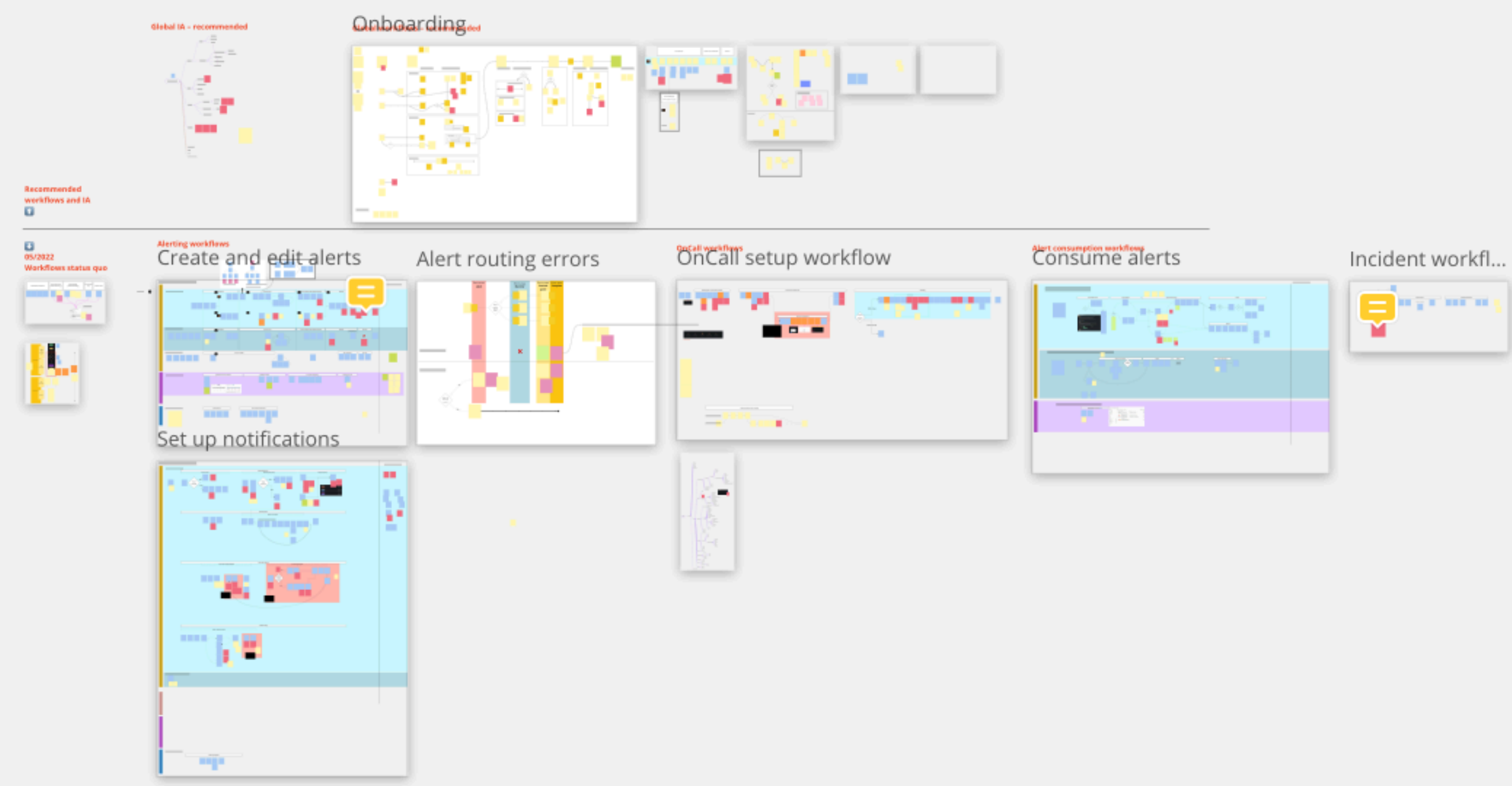
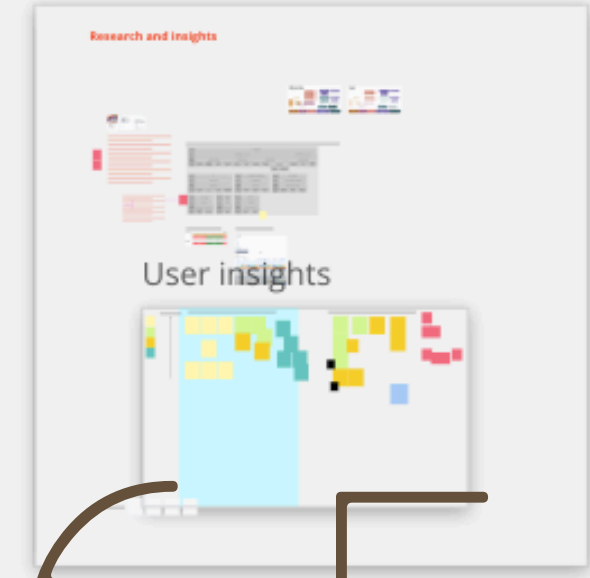




Personas and jobs



Research and insights



26,5

HOURS SPENT IN INTERVIEWS AND SENSEMAKING SESSIONS

1247

OF STICKY NOTES

68

OF SUMMARY SLIDES

alert creation journey map



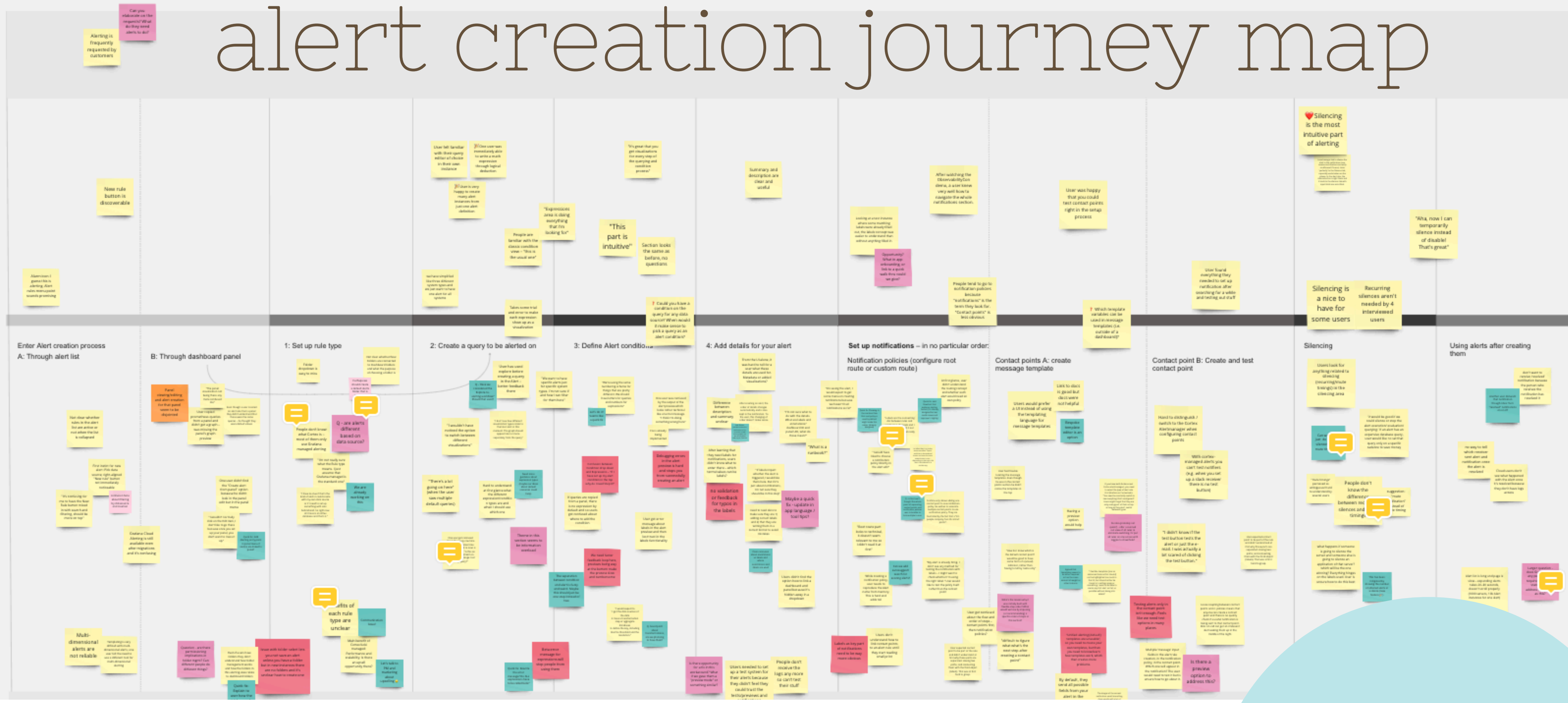
Alerting journey

Gains
General, not tied to a particular point in the journey

Gains

Pains

Pains
General, not tied to a particular point in the journey



KEY FINDINGS

1

not so many happy points, many pain points

2

main user need: guidance

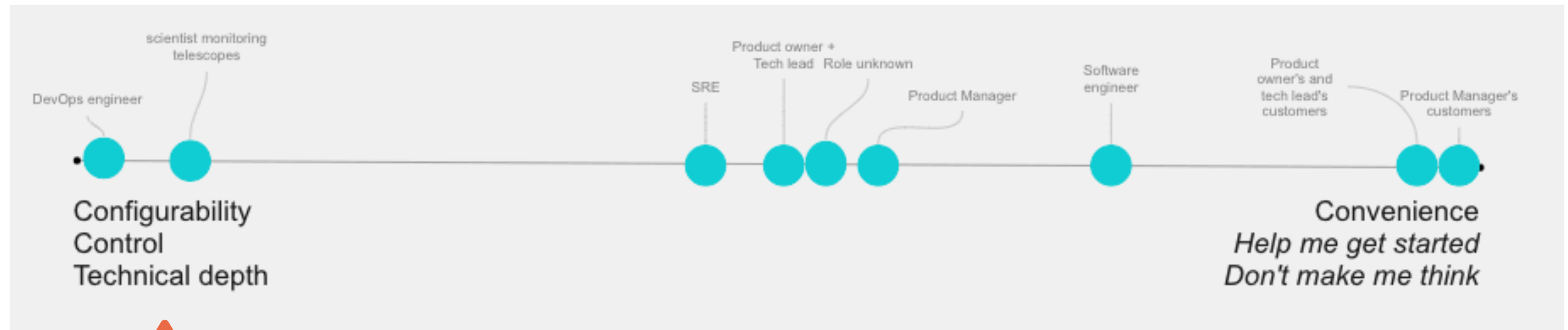
3

users want a low entry barrier
"I want to be able to create my first alert as fast as possible"

journey map in Miro

outcomes slide deck in Google Drive

DOES THIS REMIND YOU OF SOMETHING?



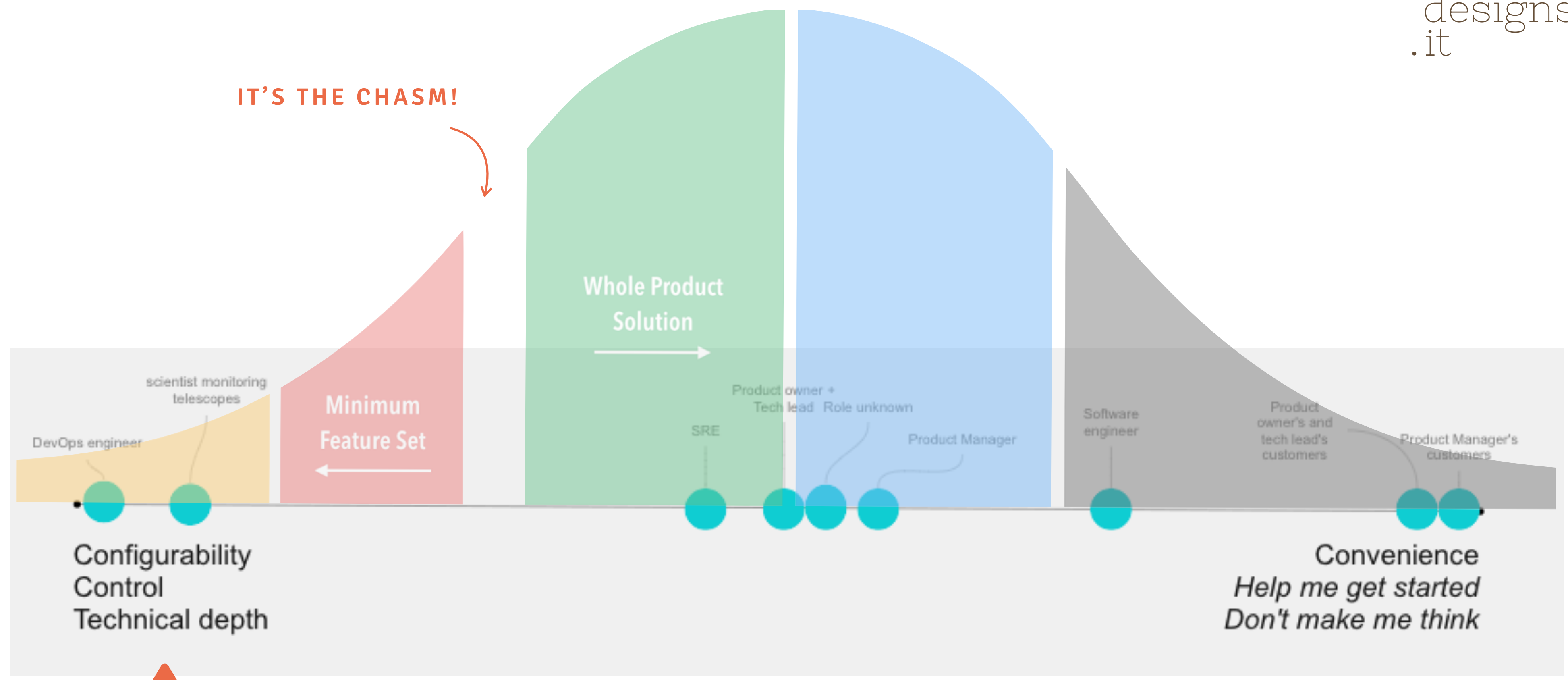
what users
want

OUT OF THEIR
ALERTING PRODUCT
EXPERIENCE

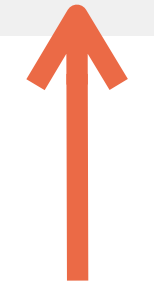
what we're
offering

what users
want

OUT OF THEIR
ALERTING PRODUCT
EXPERIENCE



what we're
offering



“maybe 10,
15min tops”

TIME A NEW USER WOULD SPEND TRYING TO
FIGURE OUT GRAFANA ALERTING*

MAIN THEMES

user confidence

- Make sure people trust the system 100%
- People won't trust something they don't understand
- Documentation, error feedback, ease of use need to be great to accomplish this

less is more

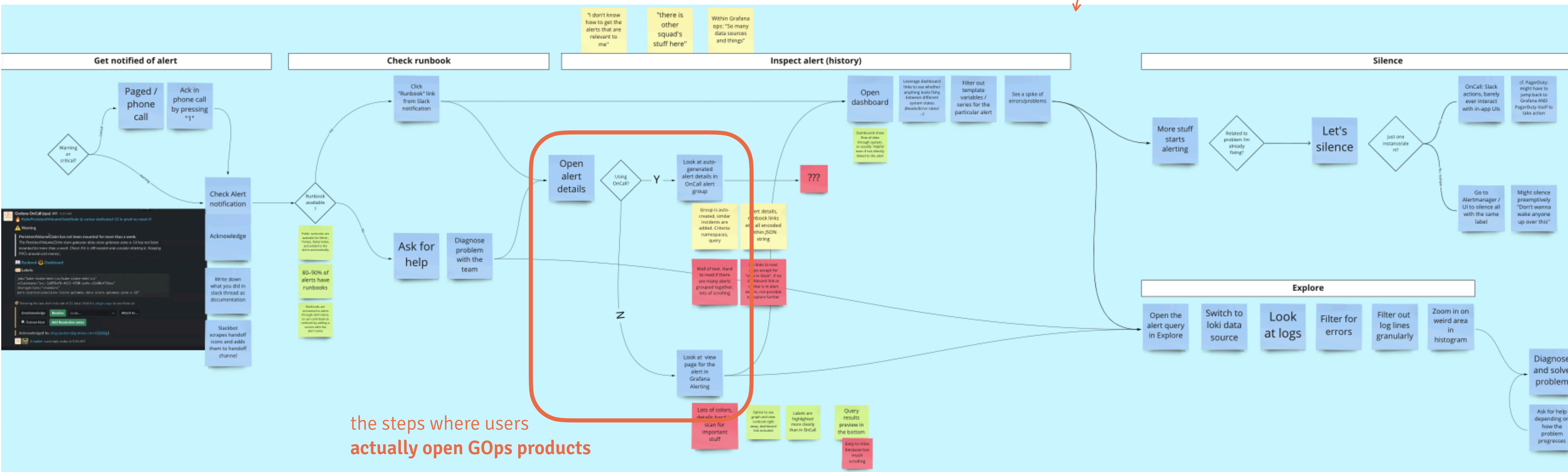
- almost all of the users got lost in details during alert creation and didn't successfully complete the task within the research session
- “there's a lot going on here”
- one user didn't even find the “New alert rule” button
- “I want to start from useful basic alerts, advanced usage can be figured out later” – a SRE during user research

MAIN THEMES

workflow ≠ interface

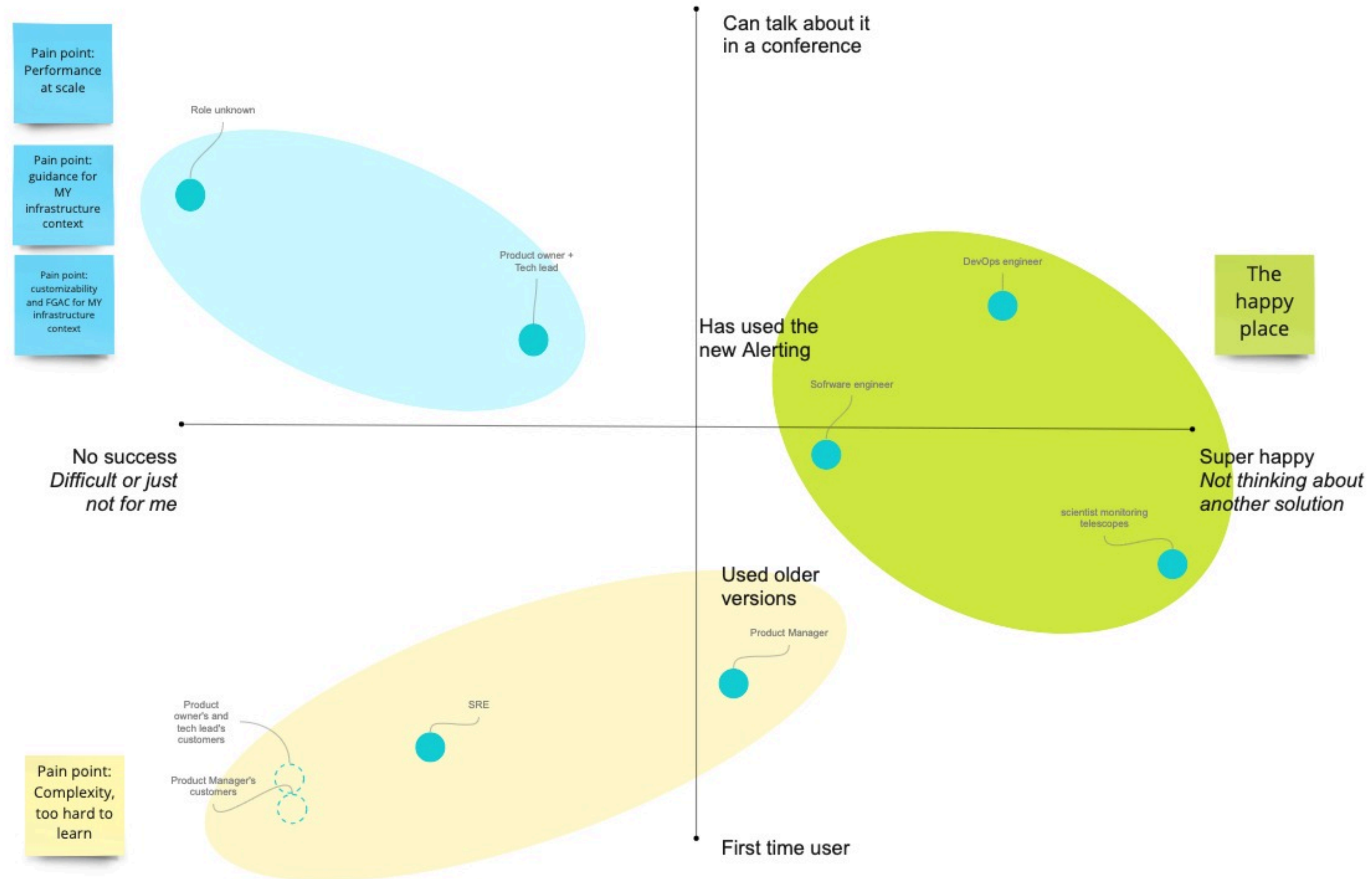
- The way users interact with and investigate alerts does not align with our tooling
- Users only access bits and pieces of Grafana when they consume and manage alerts
- We don't help users figure out in which order to do things

when consuming alerts, users barely access the GOps UI



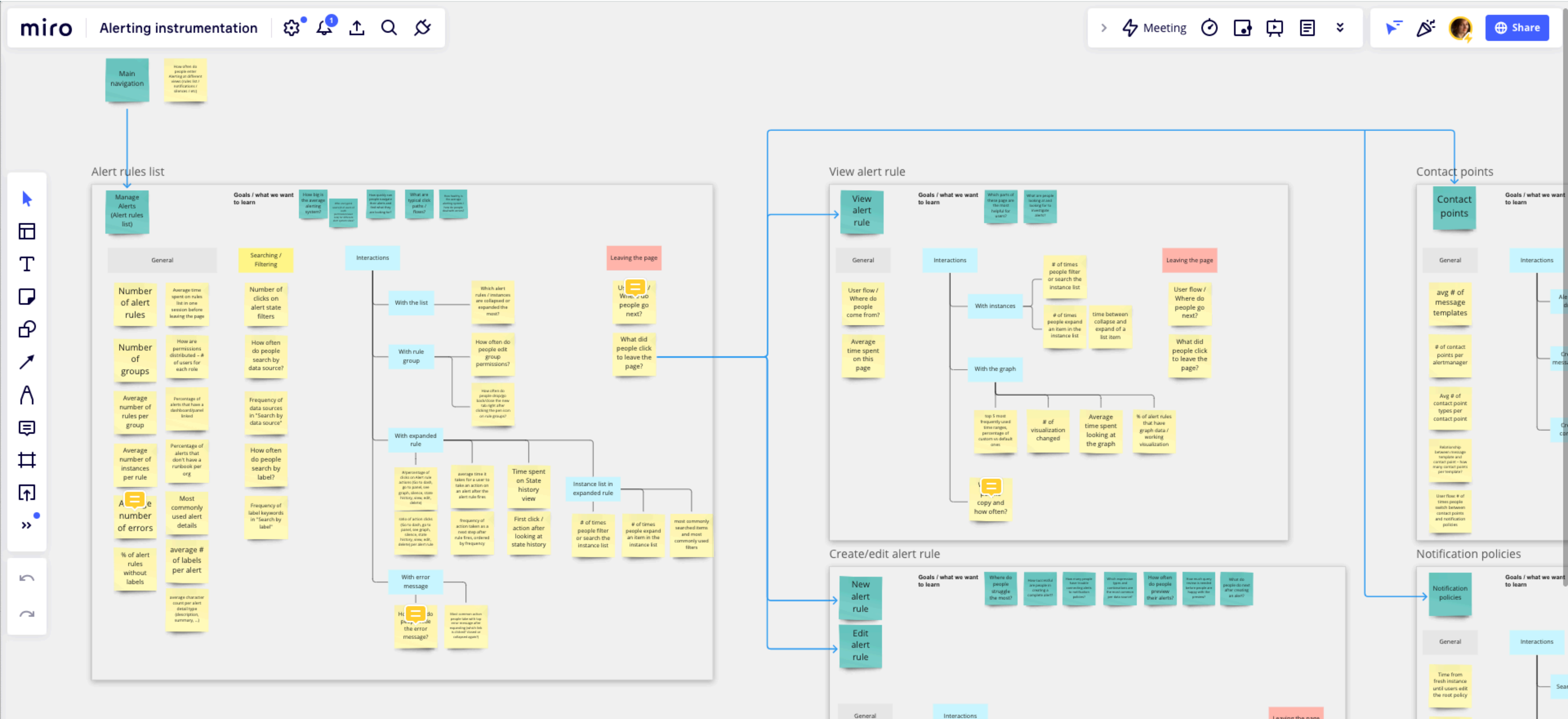
multiple personas with varying needs

HAPPINESS OF TEST USERS VS EXPERIENCE WITH GRAFANA ALERTING



I PREPARED A STRATEGY TO START COLLECTING MORE USAGE DATA

Mapping out metrics to collect for each page and view, I provided the team with a summary of which data could be the most useful and should be collected.



insights

we know the
weakest
points

setup is
hard
even for Ops experts

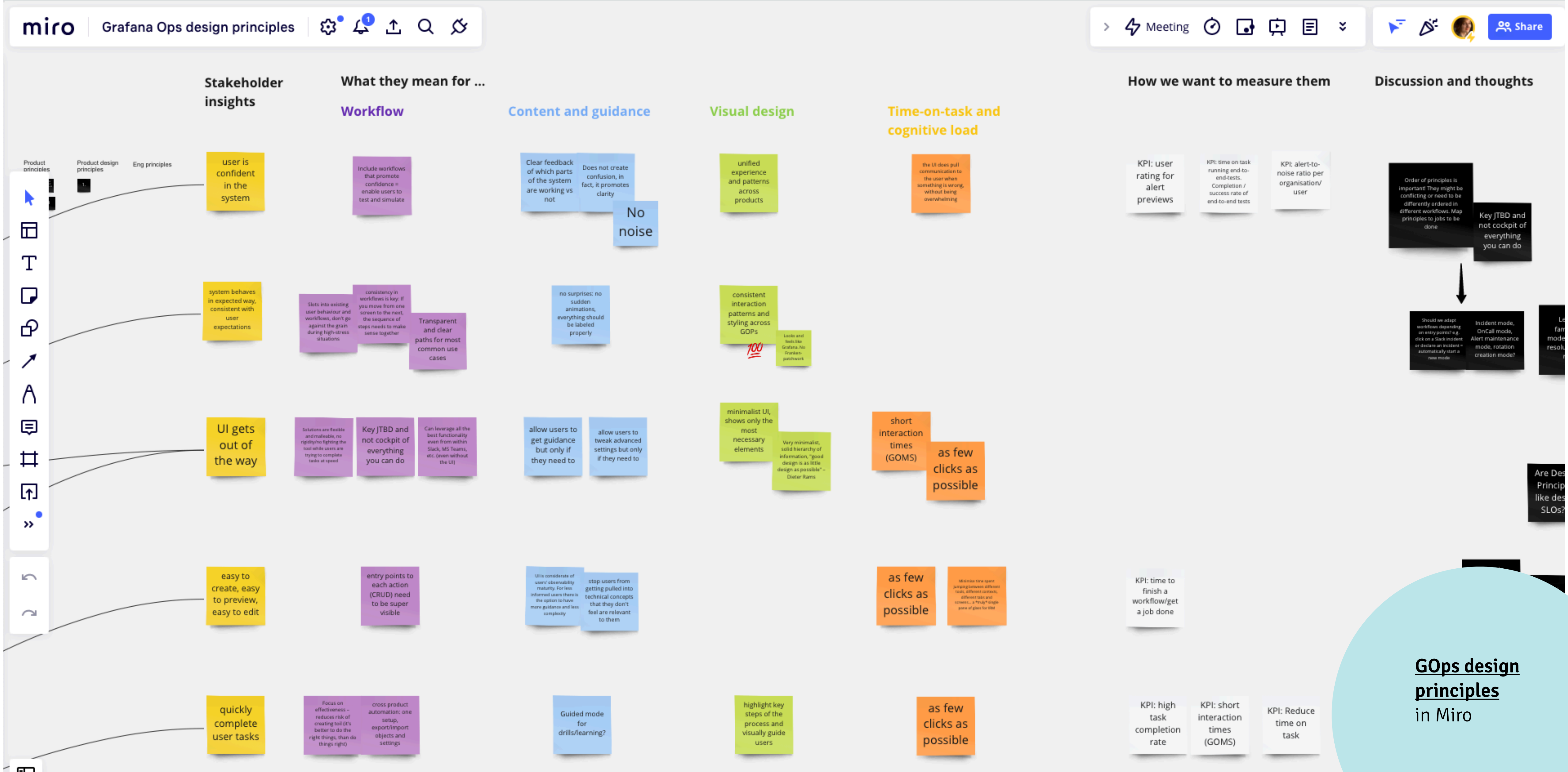
fuzzy understanding
of value propositions
every
stakeholder
said something
else

shared idea
of UX
requirements

stakeholders mentioned the same
characteristics that a successful UX
should have

the team
is biased
on average, stakeholders were
only expert in 1 of 3 tools

I FACILITATED A WORKSHOP TURNING STAKEHOLDER INSIGHTS INTO DESIGN PRINCIPLES



How does 100-user scale fit the path here?

Our theses for key setups

Why is this setup important to consider?

User jobs, needs, gains related to setup

End goal / value
How do the users get there?



KEY FINDING

there are multiple key customers:

1. Grafana Cloud + OpsGenie and/or 3rd party alerts
2. OSS Alerting + OnCall

Other less important setups

key customer workshop

[Marc] How do personas relate to this?
[Marc] Ideal customer already has teams who are on-call and running their incidents
Different teams within the same company might have different maturity levels

	High	Low
User needs related to Observability maturity	<ul style="list-style-type: none"> Manage thousands of alert rules Multi-dimensional alerting Alerts-as-code Integrate alerts from 3rd party data sources 	<ul style="list-style-type: none"> Small number of alerts, mostly using threshold conditions No multi-dimensional alerting!
Observability maturity evaluation criteria	<ul style="list-style-type: none"> Complex labeling and routing in Alertmanager Looking for ways to reduce on-call time or improve alerting workflow Regular post-mortems 	<ul style="list-style-type: none"> Just want to get working alert notifications Not familiar with alerting workflow, no post-mortems, no incident response, no runbooks, no SLAs/SLOs
	<ul style="list-style-type: none"> Time using Grafana / 3rd party alerting tool Do they use runbooks? 	<ul style="list-style-type: none"> Number and complexity of alert rules Do they use SLOs/SLIs?

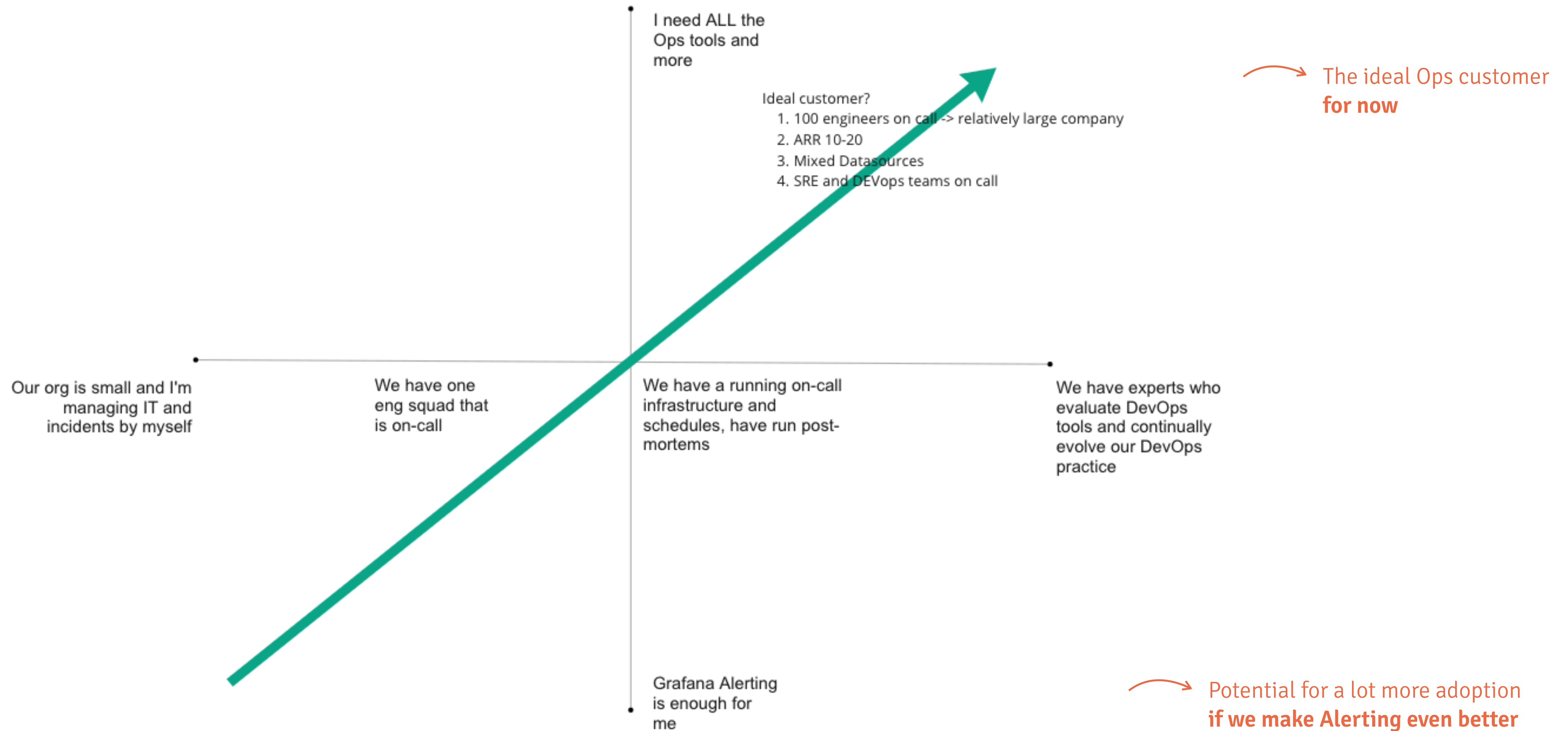
Cloud or on-prem?



read more in Miro

multiple personas with varying needs

NEED FOR OPS TOOLS VS O11Y MATURITY: A THESIS*



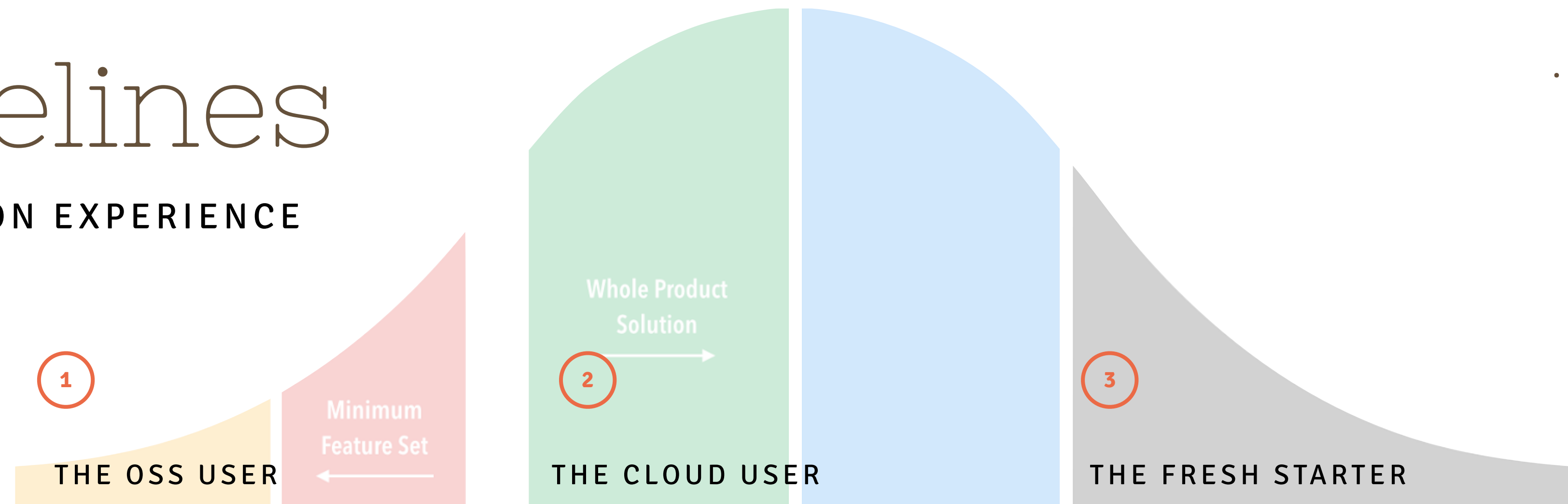
* The reality is probably more of a scatterplot

bigger org ≠
more expertise

“WE’RE AT THE VERY BEGINNING OF IMPLEMENTING
SLIS AND SLOS AT WELLS FARGO” *

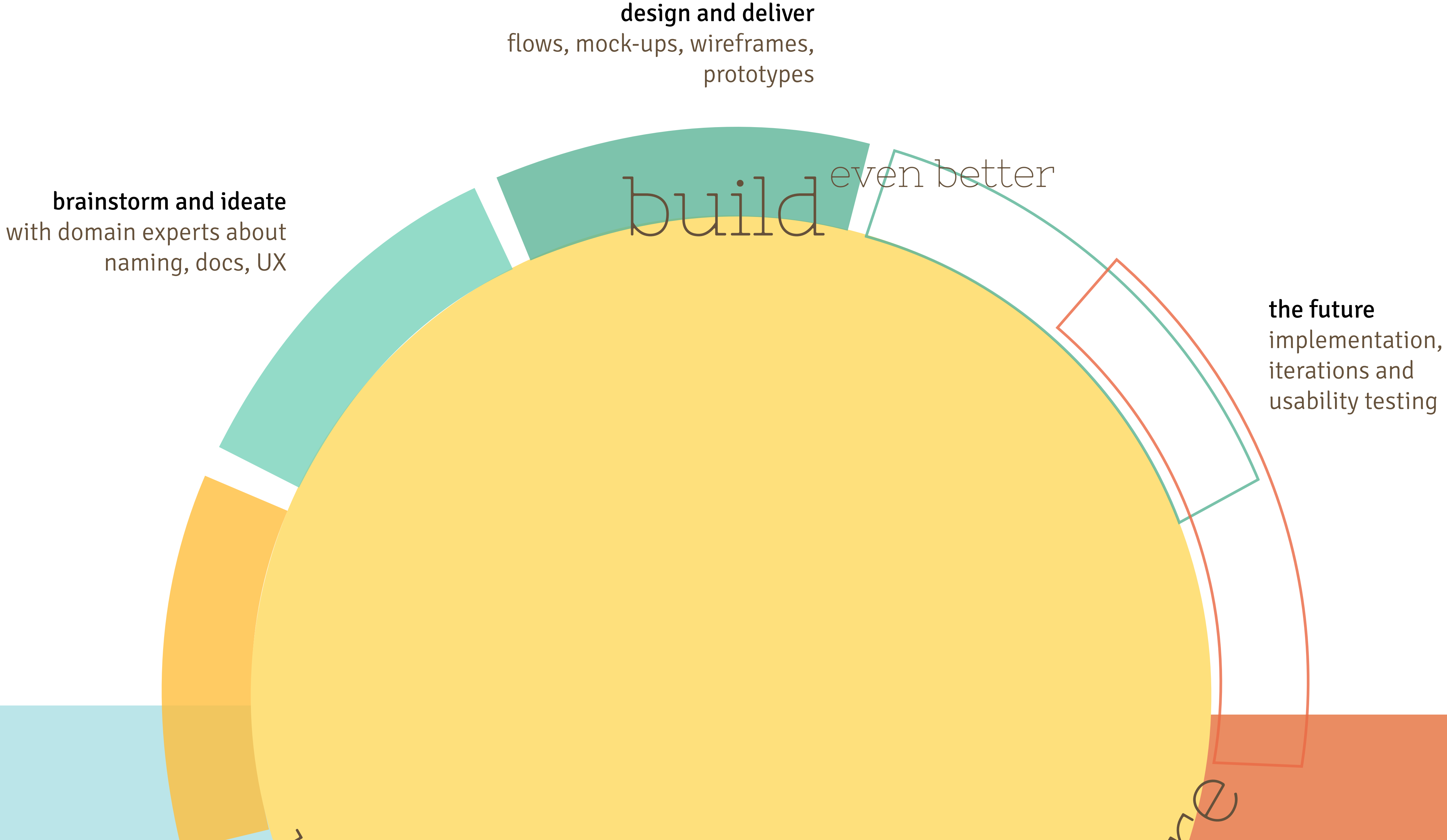
user baselines

NEEDS VARY DEPENDING ON EXPERIENCE



Experience with Grafana tools	✓	✓	little to none
DevOps maturity	✓	some	little to none
Prometheus expertise	✓	some	none
Tooling needs	“I want the best tool for each part of the job. I mix-and-match and put in the work to get there”	“I want Grafana to do as much as possible for me”	“I just want a simple alert that I can set up without reading a guidebook”
Expected complexity	show me ALL the settings	simple at first, willing to learn more	simple and quick

WITH ALL THIS KNOWLEDGE,
I DESIGNED SOLUTIONS



brainstorm and ideate

I generated ideas with cross-functional stakeholders and then polished the diamond using UX expertise.

↪ BRAINSTORMING AND WIREFRAMING WITH GILLES

↪ GETTING UX FEEDBACK

↪ PROTOTYPING

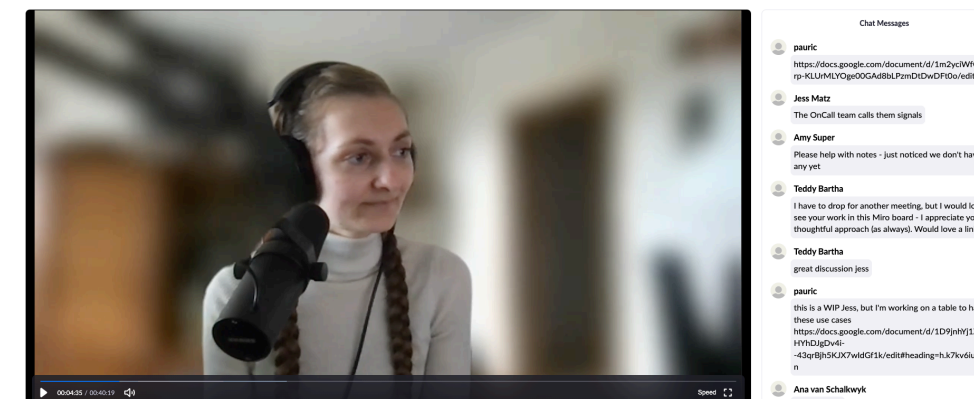
Persona-based user jobs

On-call engineer

Alerts/alert grou...

Detail view

Research and insights



Unified lists for alerts and incidents	Jessic...	25 min	I want to reduce list complexity in GOPs by optimizing for user personas and jobs. I still have nuts to crack. Please help me come up with good solutions :)
--	-----------	--------	--

Problems that I am trying to solve:

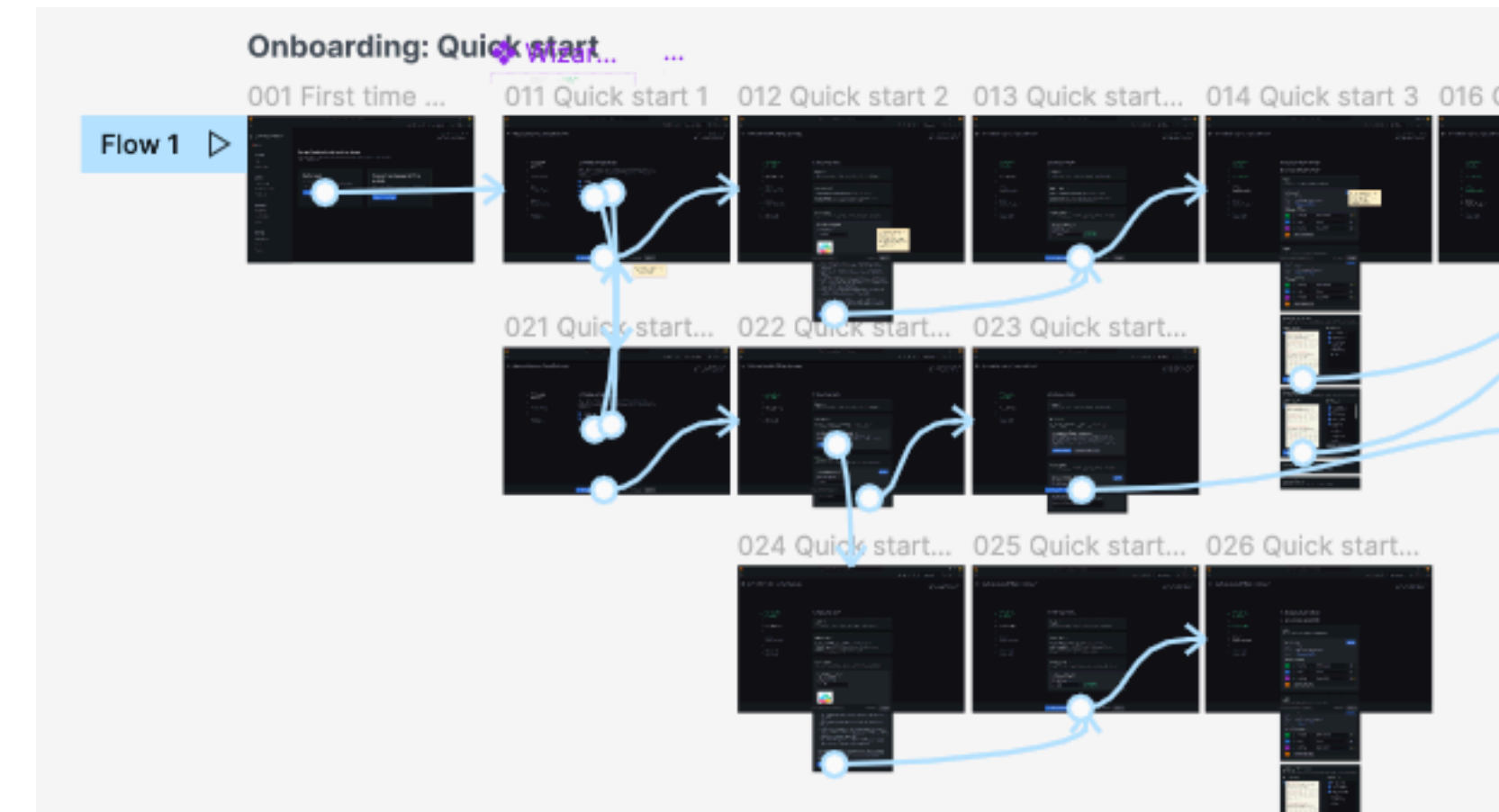
- 1. View modes?
 - How to switch between them?
 - How to mitigate potential problems?
- 2. Displaying filtering and grouping
 - How to visualize label grouping?
 - How to highlight active filters? Should there be grouping by filters other than labels?
 - Should we allow users to group by whatever they want? How to figure this out?
- 3. Distinguishing between similar but different element types
 - OnCall alert group vs Alerting alert rule vs alert instance
 - Will users care?
- 4. Implications for information architecture
 - Could we have one alert list that is not tied to product buckets?

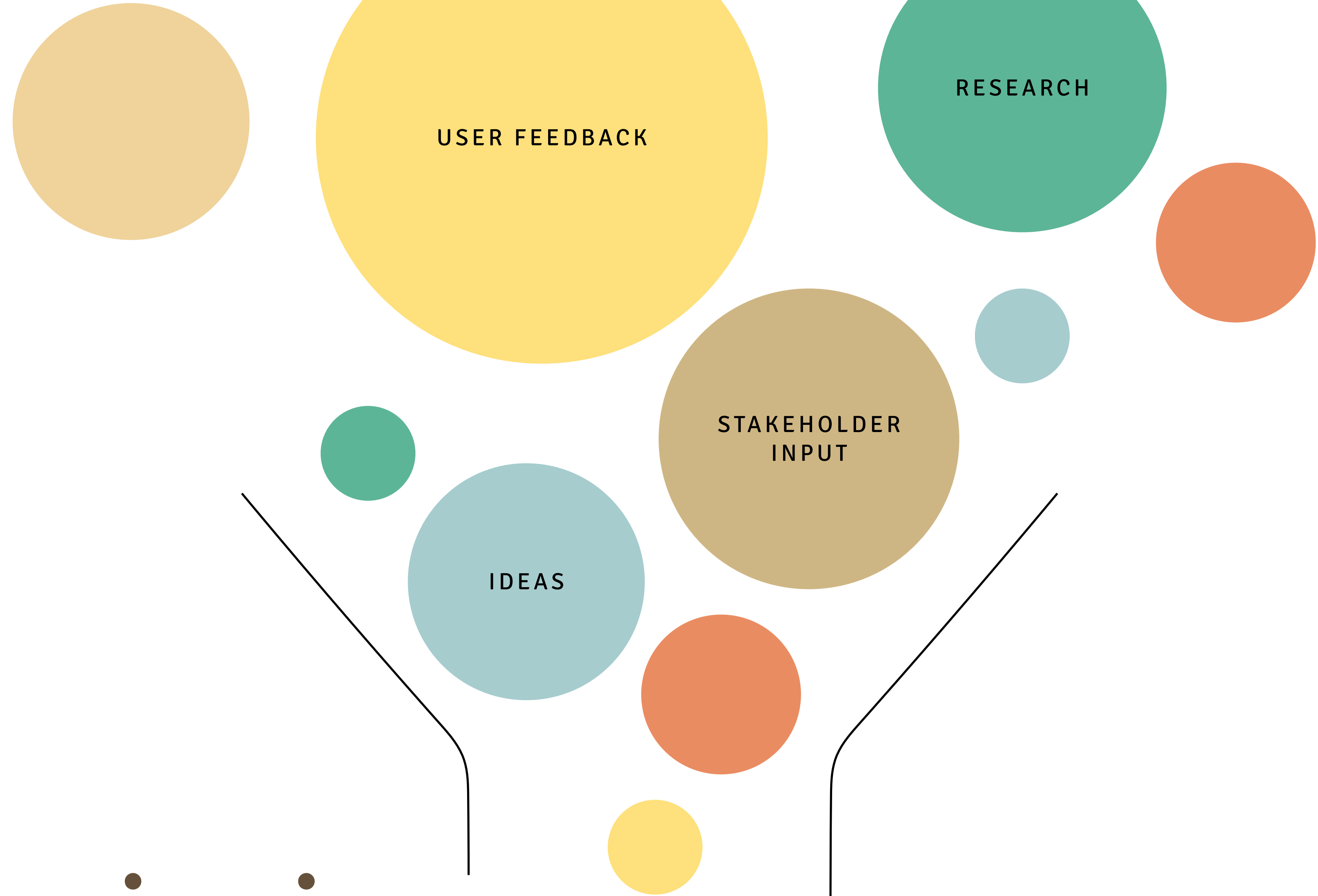
What kind of feedback am I looking for:

- Ideas and good practices

Discussion – use to show what has been discussed

- [Amy] Unified list - this makes sense to me as a way of simplifying. But I'm getting tripped up on the difference between folders and groups - is this actually different?
 - [Jess & Rob] architecturally this is actually different
 - But we all agree that users don't actually care - they are just trying to get their job done.
- [Luke] Folders - most customers use this as a way to organize resources so we probably shouldn't move away from them. They are also used for access control.
- Looking at the github-style filtered list
 - Tricky to be clear about an alert rule vs. group vs. instance
- [Luke] Like the wireframe approach - this is a good starting point for explorations. If we have ways to filter we could hide some things and make it more of a search





vision of a unified GOps

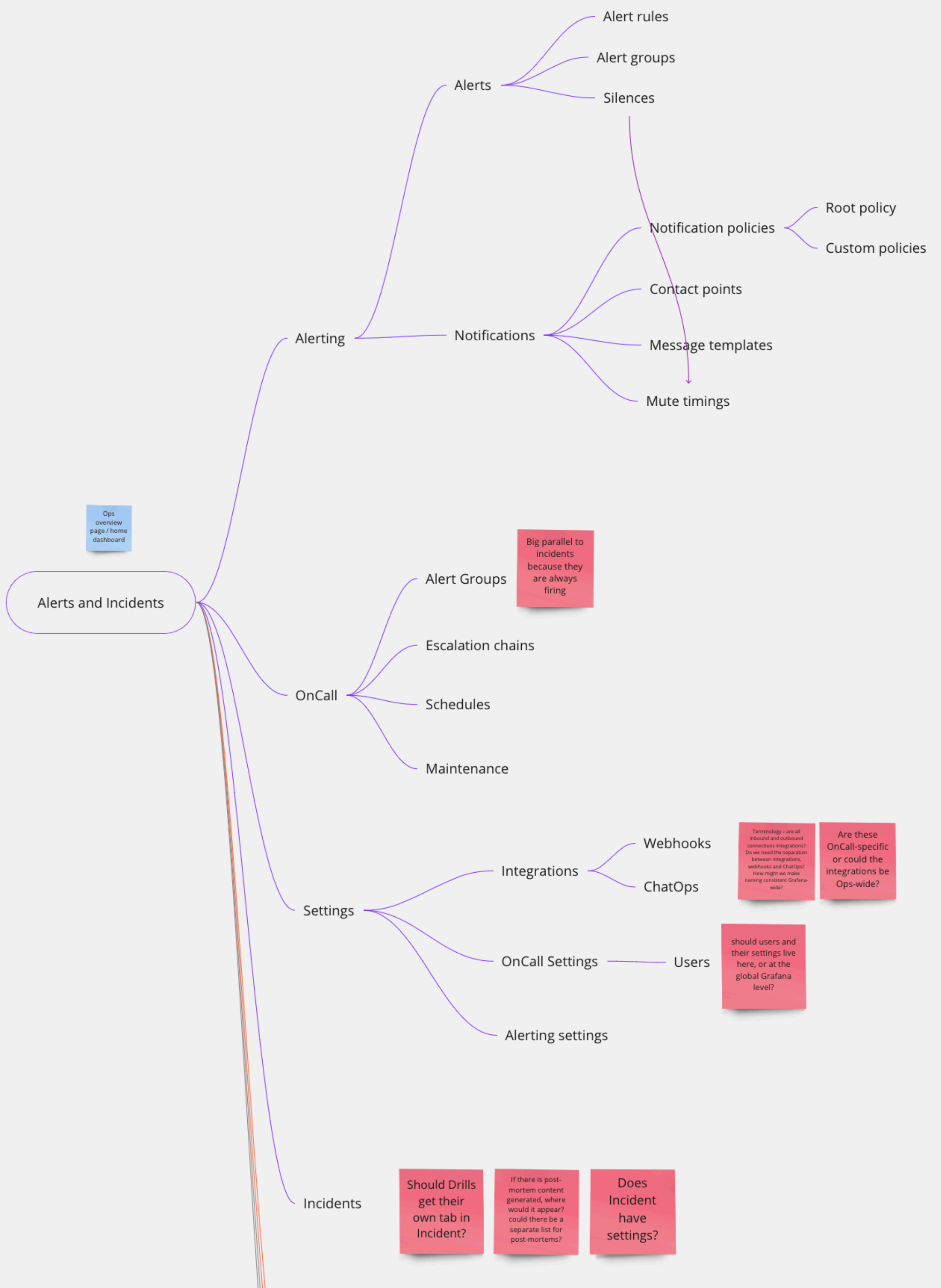
BIG HAIRY AUDACIOUS UX CHANGES I AM PROPOSING

PROBLEM

Separation of GOps tools into plugins is caused by our org structure and tech reasons. It has little to do with users and their workflows and introduces user toil, e.g. when installing the same Slack integration 3 times.

SOLUTION

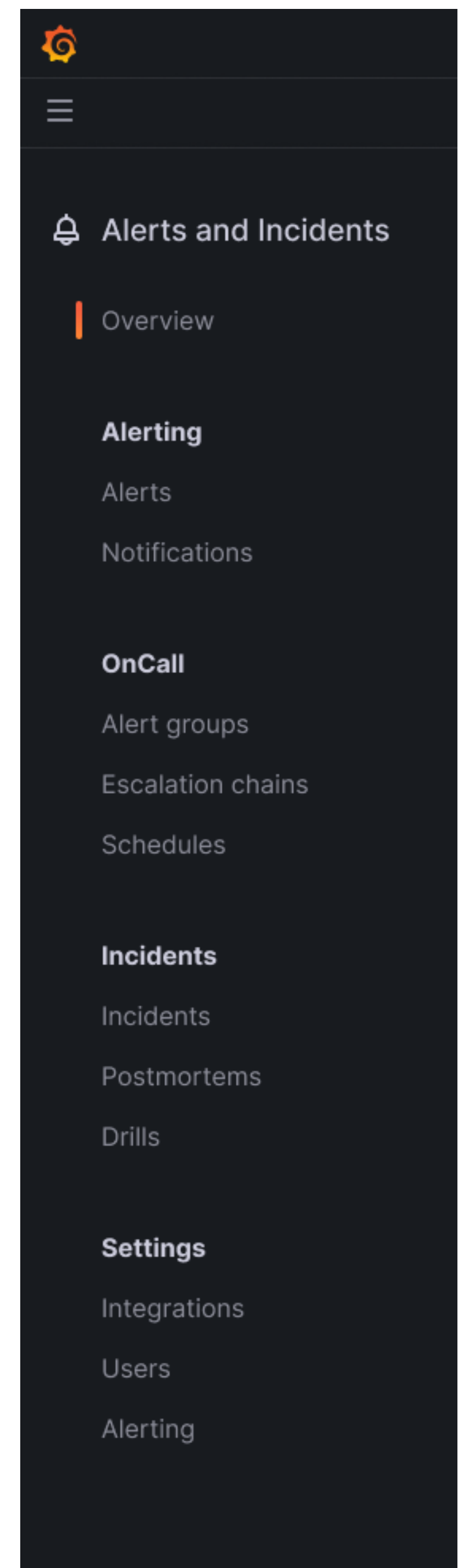
changes in information
architecture

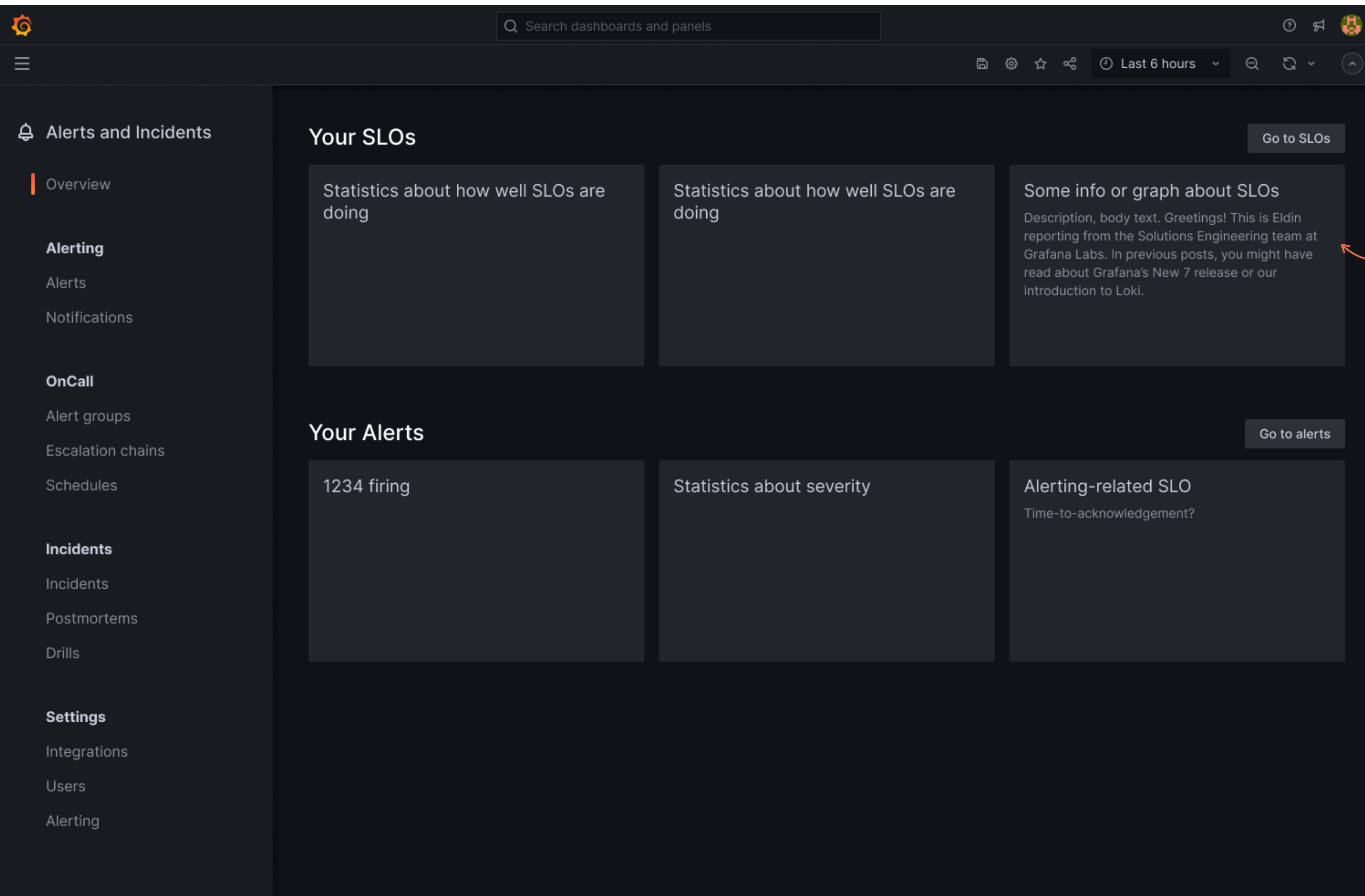


Heavily reduce amount of L1 navigation items.

Introduce a unified settings area where options can change depending on the products you have installed.

One place to integrate them all





very early proposal –
needs full-fledged UX project

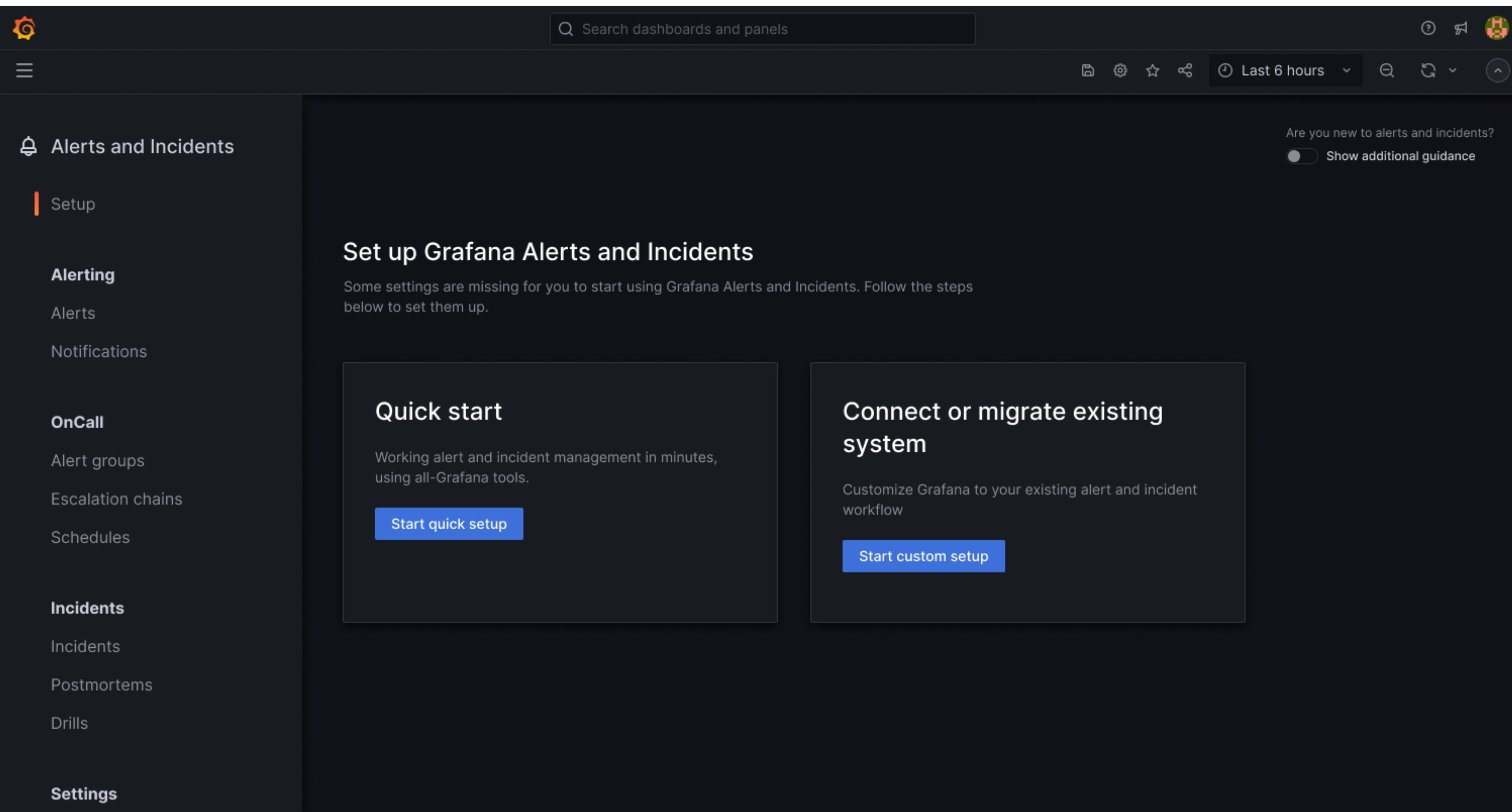
Overview dashboard that both shows the most important data of this ops system and gives quick entry points and upsell opportunities for multiple Ops products.

PROBLEM

We only have 10–15 mins to help users onboard to our tools.
We're not providing a tailored experience that is mindful of user needs.

SOLUTION

forked onboarding



Additional guidance option
for first-time users

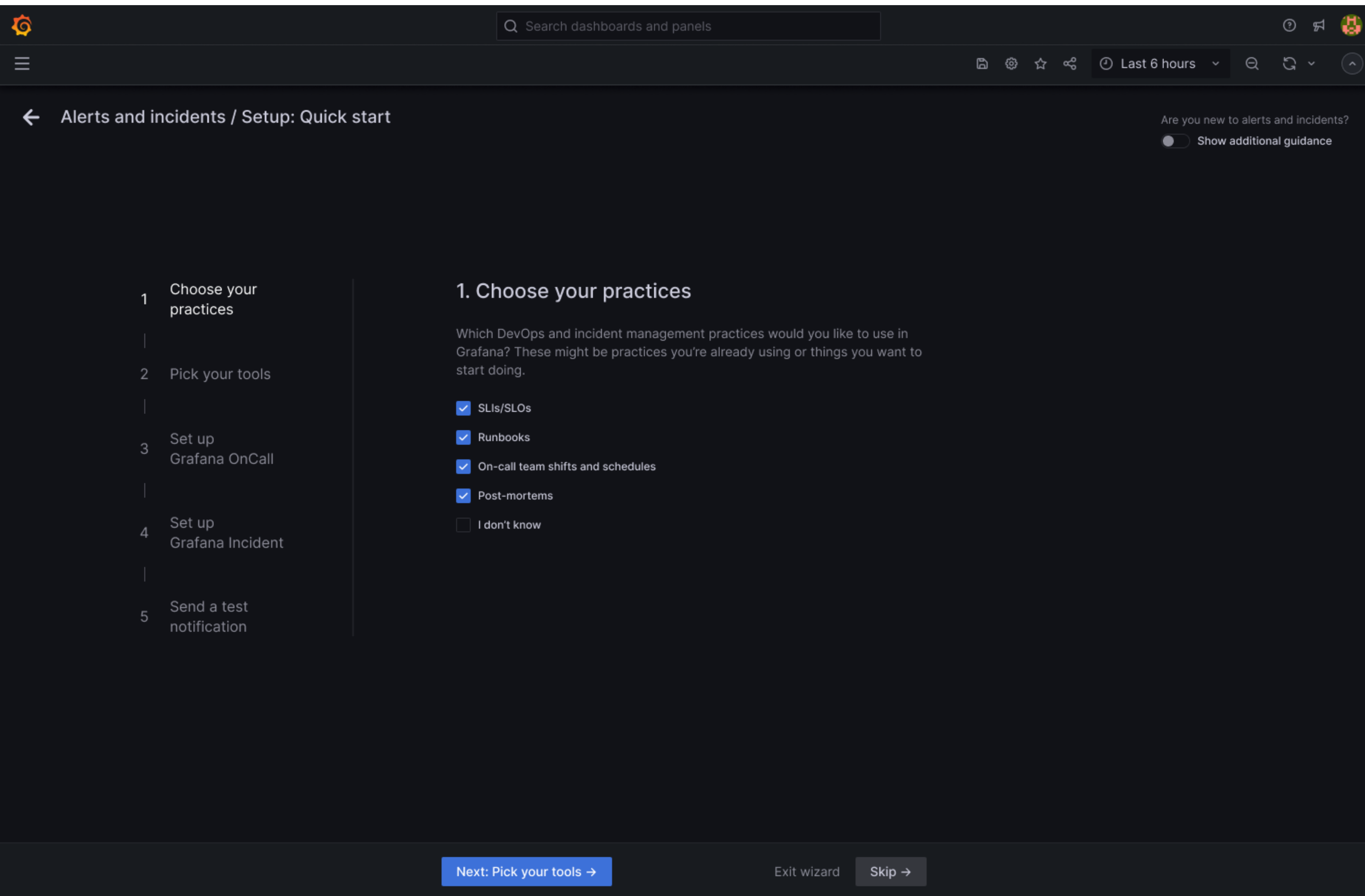
Users who are opening the Alerts and Incidents are for the first time get to start their setup flow right away, choosing between two options catered to key personas.



Priority is quick setup.
We automate away all the settings we can



Priority is migration.
We give custom settings and API instructions as needed



The wizard starts by asking users what they want to use in Grafana. Based on the user selection we install the matching products.

- ✓ Choose your practices
- ✓ Pick your tools
- 3 Set up Grafana OnCall
- 4 Set up Grafana Incident
- 5 Send a test notification

3. Set up Grafana OnCall

Users
34 users found in [Grafana users management](#).

jeshka (you) Save

Role: Admin
E-Mail: jessica.matz@grafana.com
Phone: [Add phone number](#)

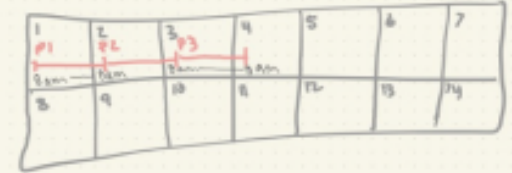
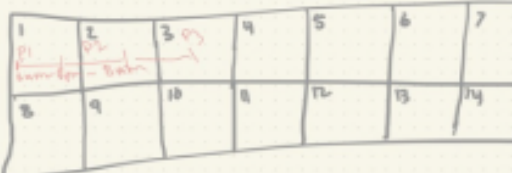
Notification timings

1	Notify by	Slack mentions	🗑️ ⚠️
2	Wait	15 min	🗑️
3	Notify by	Phone call 📞	🗑️ ⚠️
4	Add notification step		

Do we need a per-user Slack integration? How do OnCall Slack integrations work?

Create your first rotation
You will be able to further customize your rotation after finishing this setup.

Choose a pattern

- **24x7 Rotation**
Each person oncall for 24 hour shifts rotating each day.

- **Follow the sun Rotation**
Use timezones to your advantage - 12 hour shifts each day.


Choose users

- Max Mustermann
- Jasa Zelmanovic
- Raphael Batyrbaev
- Vaughn Gunnell
- Matvey Kukuy
- Gilles de Mey
- Rob Whelan
- Peter Holmberg
- Mat Ryer

Select all Deselect all

Create rotation

Escalation chain ✓
Grafana OnCall will use an escalation chain preset.

Some questions remain that the team will pick up

Basic OnCall settings like user notification timings and an initial rotation. We use an escalation chain

Alyssa contributed the rotation pattern picker scribble that still needs to be translated to a proper mock-up

- ✓ Choose your practices
- ✓ Pick your tools
- ✓ Set up Grafana OnCall
- 4 Set up Grafana Incident
- 5 Send a test notification

4. Set up Grafana Incident

Contact point ✓

Grafana Incident will send notifications to MySlackOrg.slack.com. It will automatically

- create a new Slack channel per incident with the prefix "incident-"
- post important updates and announcements to the channel
- invite active participants to the channel

@ Customize

Change contact point

Google Workspace

Grafana Incident uses Google Workspaces to automatically create a Google Meet meeting or Google Drive document for every incident.

1. Click the "Install Google Workspace" button.
2. Once you're redirected to authenticate with Google, enter your Google credentials associated with the account you wish to connect to Grafana Incident.
3. Once you're done, return back to Grafana Incident.

Install Google Workspace

GitHub

Connect GitHub repositories with Grafana Incident.

1. Click the "Install GitHub" button
2. Once you're redirected to authenticate with GitHub, enter your GitHub credentials associated with the account you wish to connect to Grafana Incident.
3. Once you're done, return back to Grafana Incident.

Install GitHub



We offer
**presents and defaults
wherever we can**

The flow is heavily inspired by Vaughn's amazing Incident onboarding flow. However, the setup for each product is shortened to a minimum to save time.

The screenshot shows the Prometheus Alerts and Incidents setup wizard. At the top, there is a search bar for dashboards and panels, and a navigation bar with icons for home, settings, favorites, and share. The current step is '5. Send a test notification'. A sidebar on the left shows the progress: 'Choose your practices' (checked), 'Pick your tools' (checked), and 'Send a test notification' (current step, indicated by a '3'). The main content area shows a 'Provisioned alert rule' named 'AlertmanagerKubeJobProblem' with a green 'Notification sent' status. Below this, there is a question 'Are you happy with your notification?' and three buttons: 'Yes, finish setup' (highlighted in blue), 'No, customize notification', and 'I didn't receive a notification'. At the bottom, there are two buttons: 'Finish setup →' and 'Exit wizard'.

← Alerts and incidents / Setup: Quick start

Are you new to alerts and incidents?
 Show additional guidance

- ✓ Choose your practices
- ✓ Pick your tools
- 3 Send a test notification

5. Send a test notification

Make sure your notifications are working as expected.

Provisioned alert rule
AlertmanagerKubeJobProblem
✓ Notification sent

Are you happy with your notification?

- Yes, finish setup
- No, customize notification
- I didn't receive a notification

Finish setup →

Exit wizard

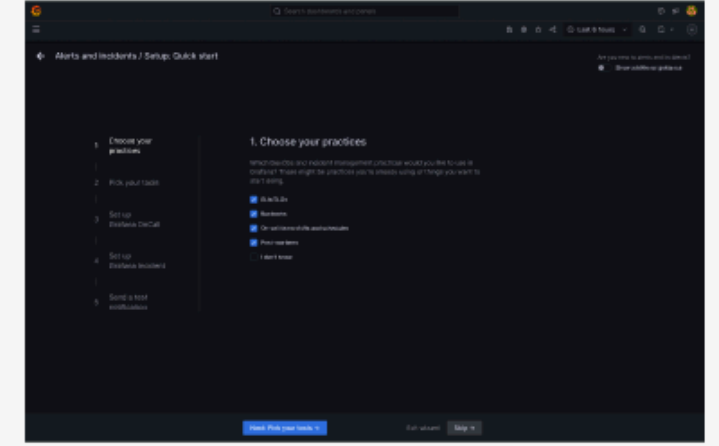
In the end, we send a test notification. If users don't like the default notification, they get the option to customize it right from the get-go.

: Quick start

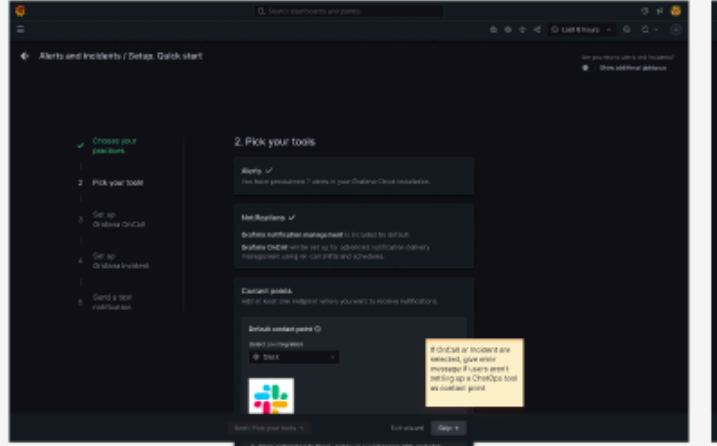
Wizard nav item

Instance

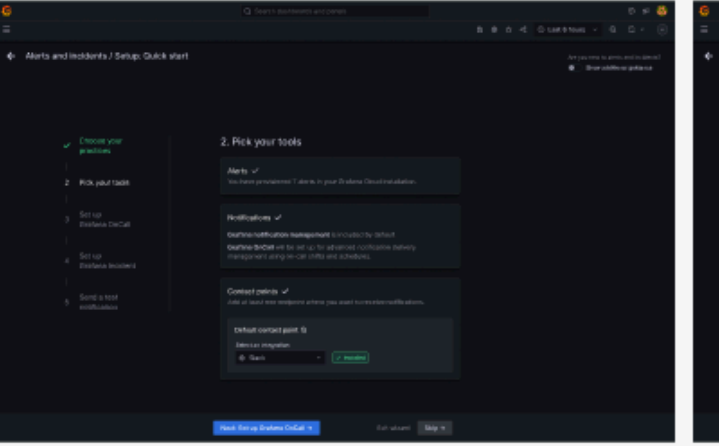
011 Quick start 1



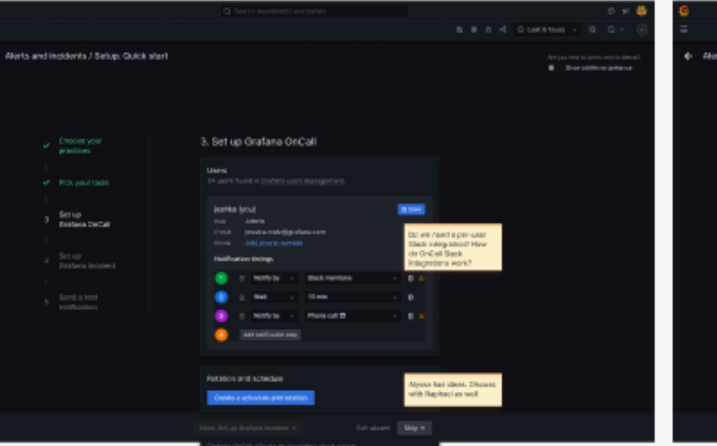
012 Quick start 2



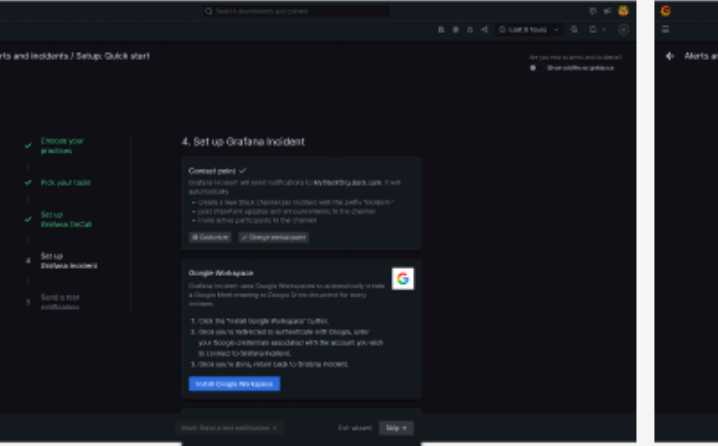
013 Quick start 2.2



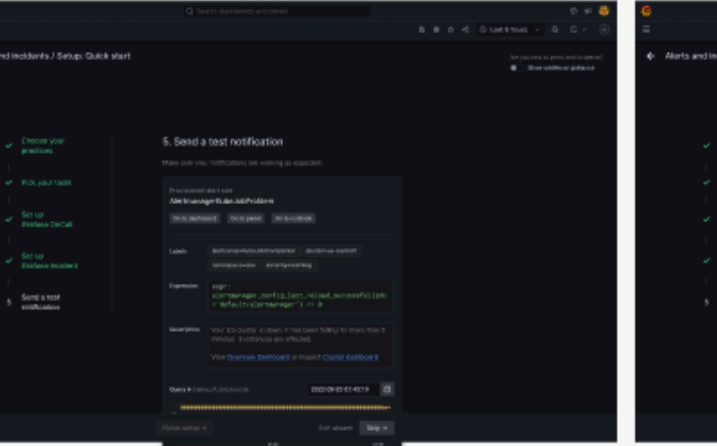
014 Quick start 3



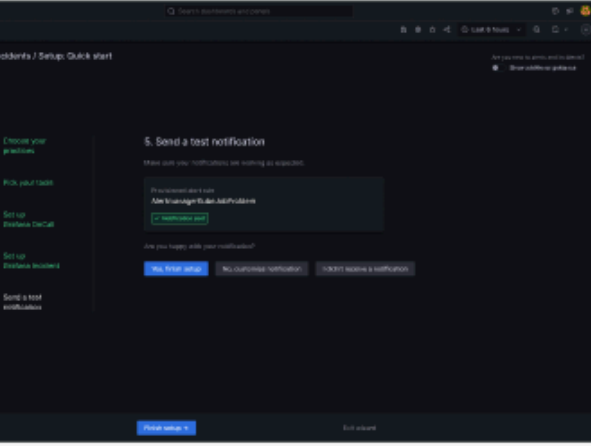
016 Quick start 4



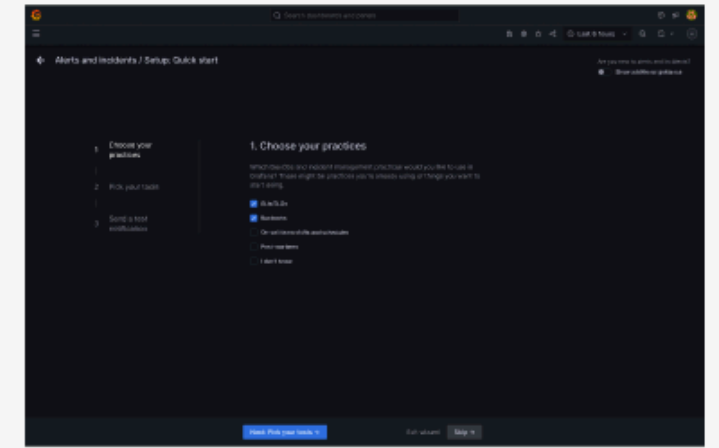
017 Quick start 5



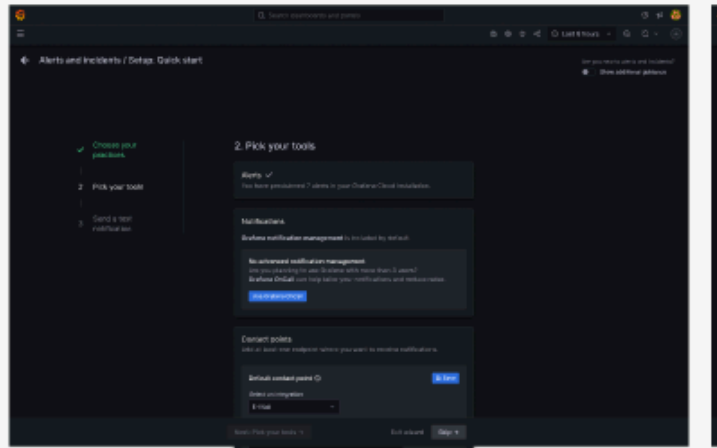
018 Quick start 5.2



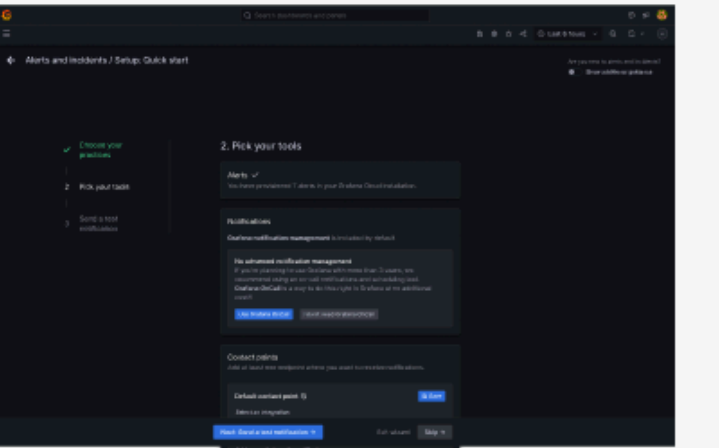
021 Quick start 1.2



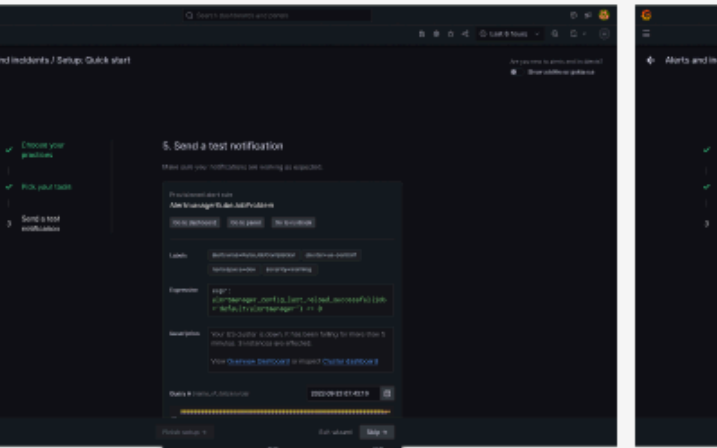
022 Quick start 2B



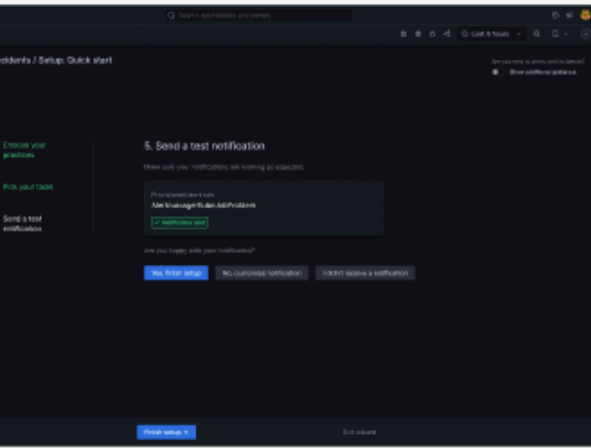
023 Quick start 2.2



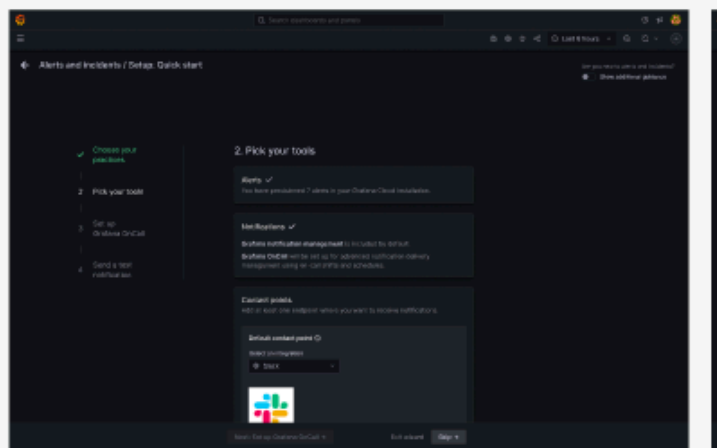
027 Quick start 4



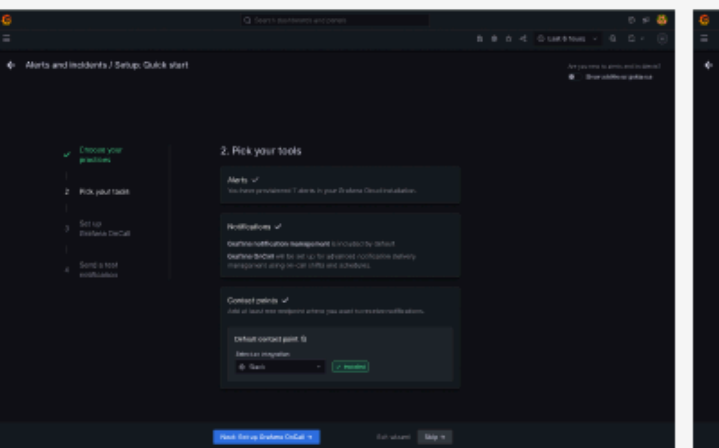
028 Quick start 4.2



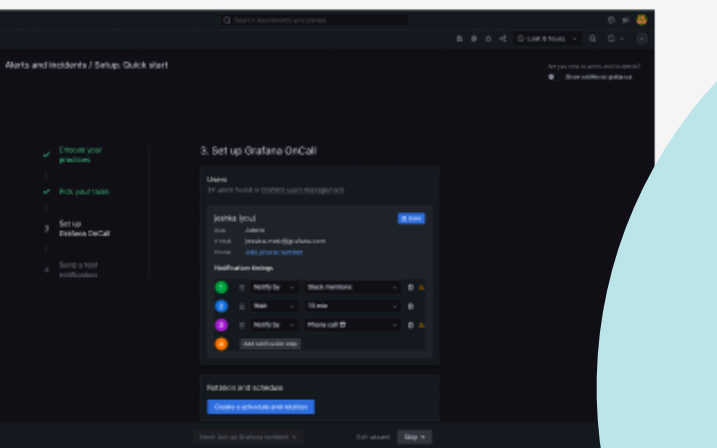
024 Quick start 2.3



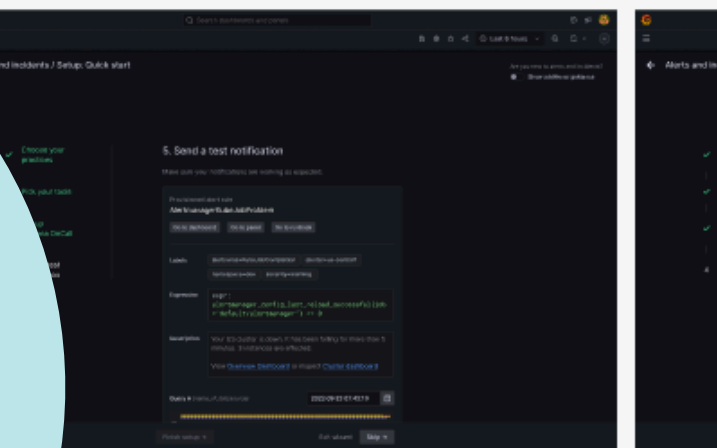
025 Quick start 2.4



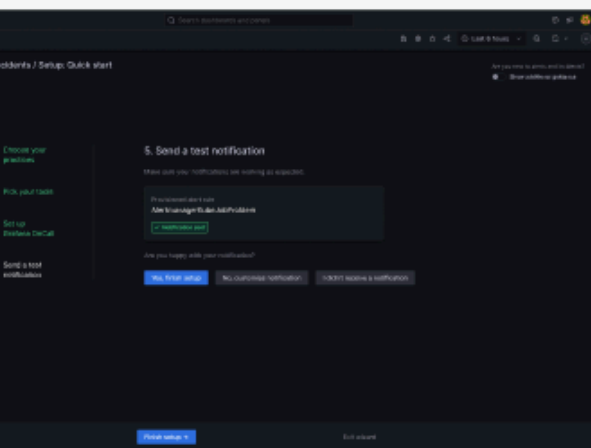
026 Quick start 3



029 Quick start 4.3



030 Quick start 4.4

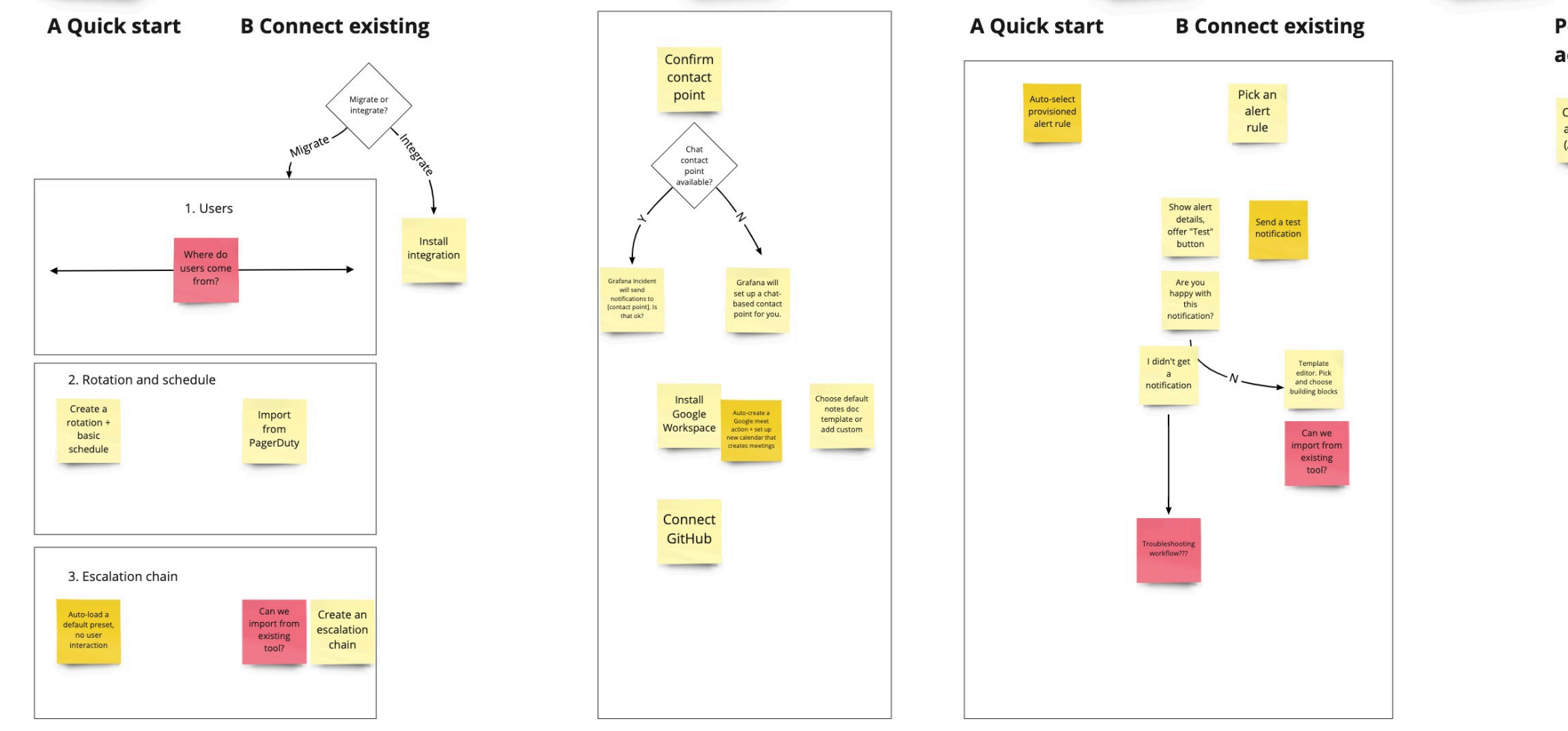
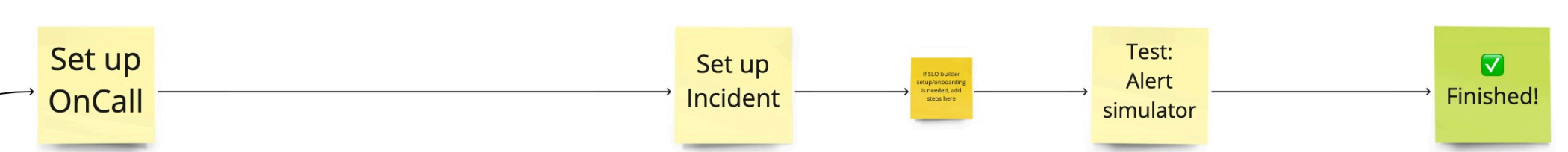
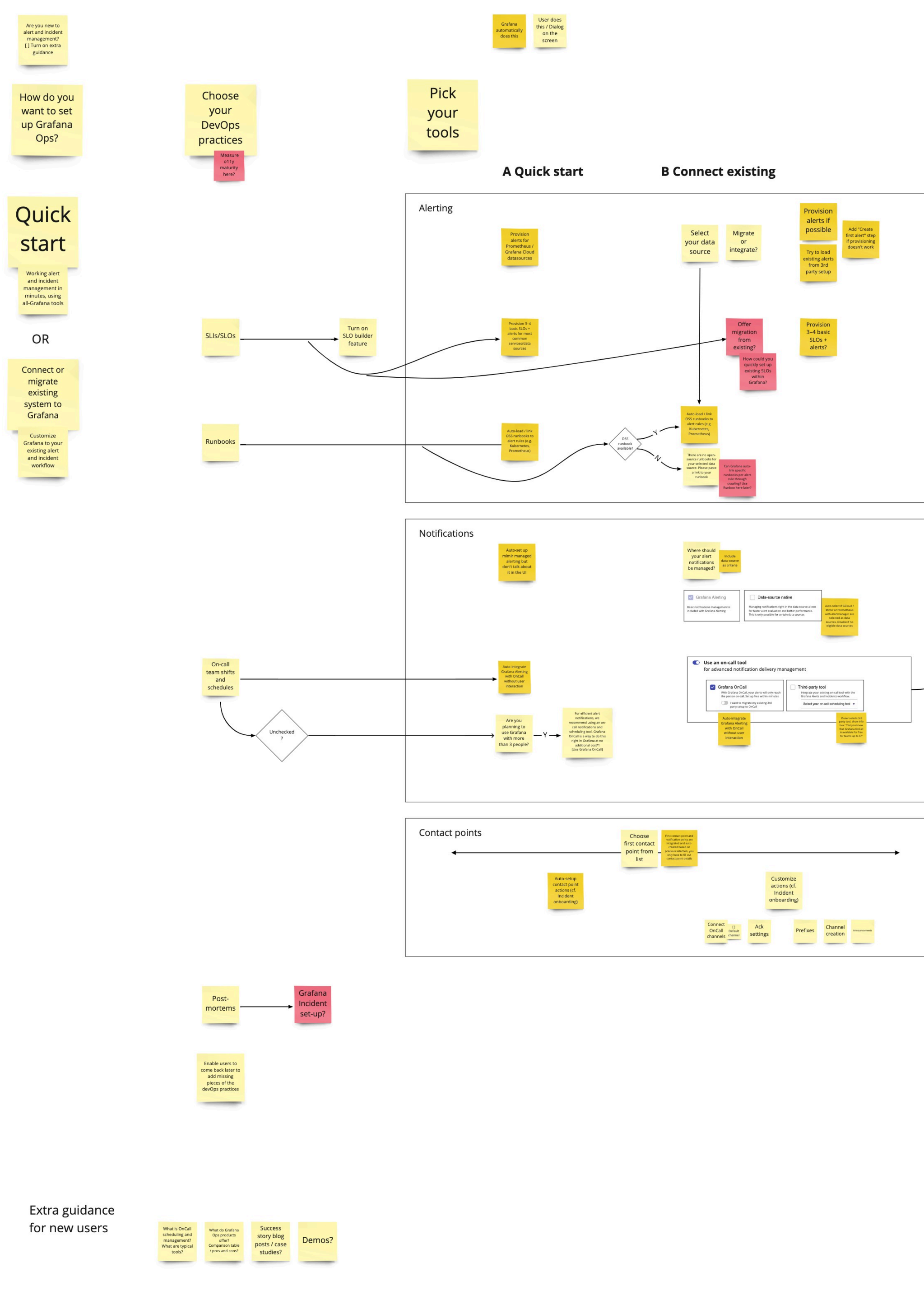


prototype
in Figma

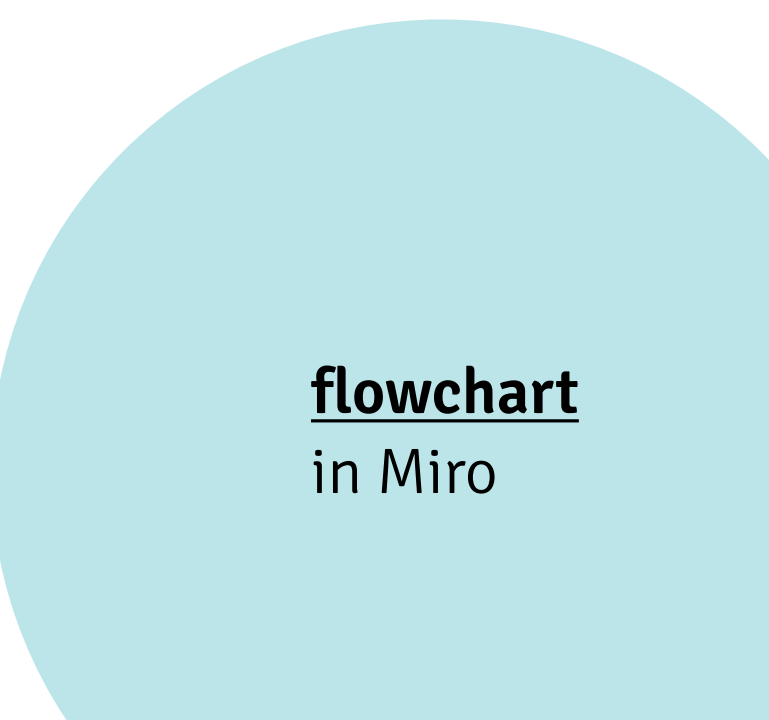
“connect existing” setup: automation is also key

AUTOMATION IDEAS

- Provision alerts for existing data if possible
- Offer migration options for top 3 existing 3rd party alerting sources?
- Provision 3–4 basic SLOs + alerts? Could existing SLOs be migrated somehow?
- Auto-link OSS runbooks to alert rules (e.g. public Kubernetes and Prometheus runbooks).
 - Allow users to paste a link to their runbook if their data source doesn't have an OSS runbook
- Import rotation and schedule from competing IRM tools (PagerDuty, Ops Genie)?
- Import escalation chain from existing tool?



No mockups or prototype yet for the second "Connect existing" setup, but the concept is done!



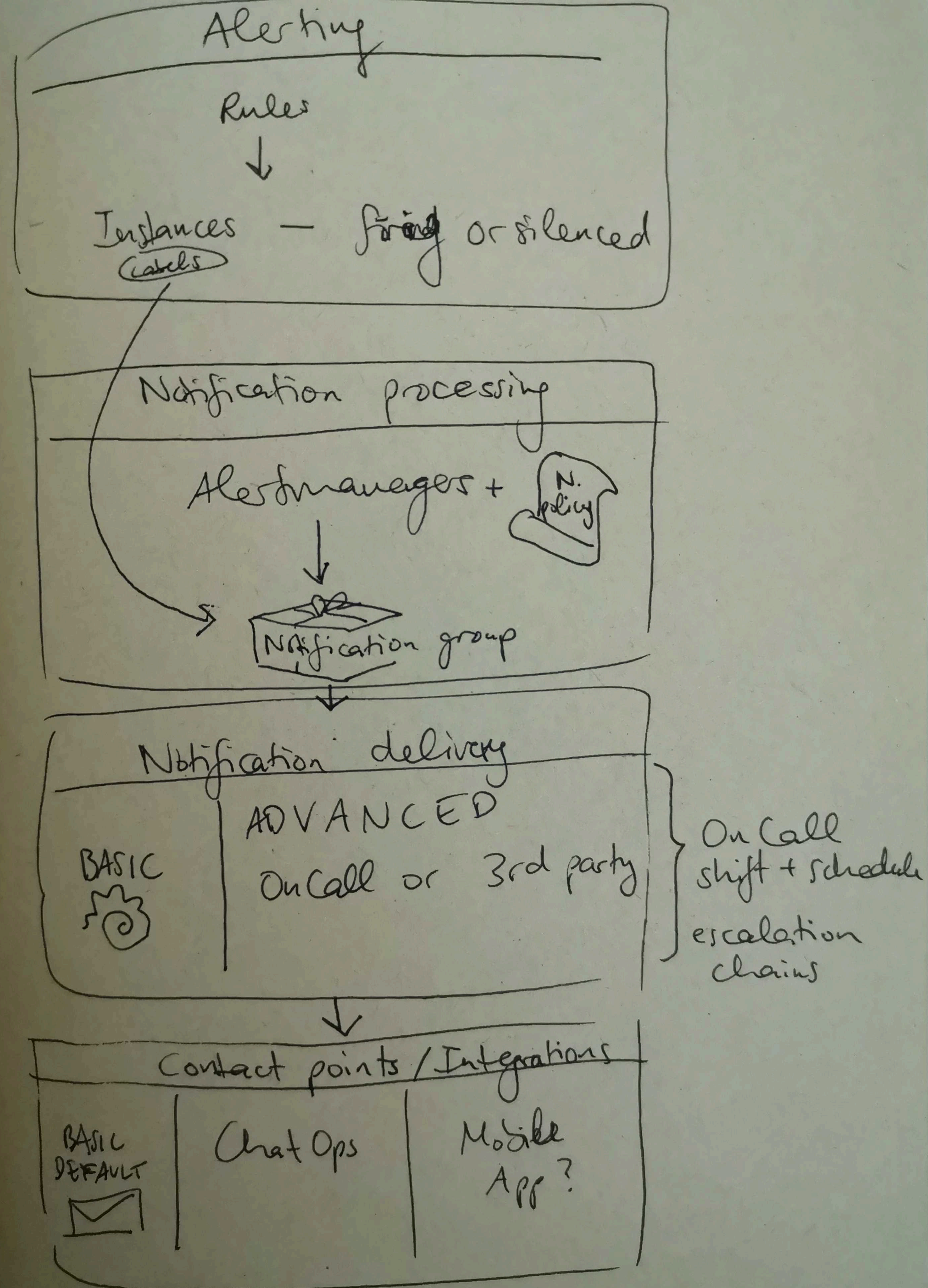
flowchart in Miro

PROBLEM

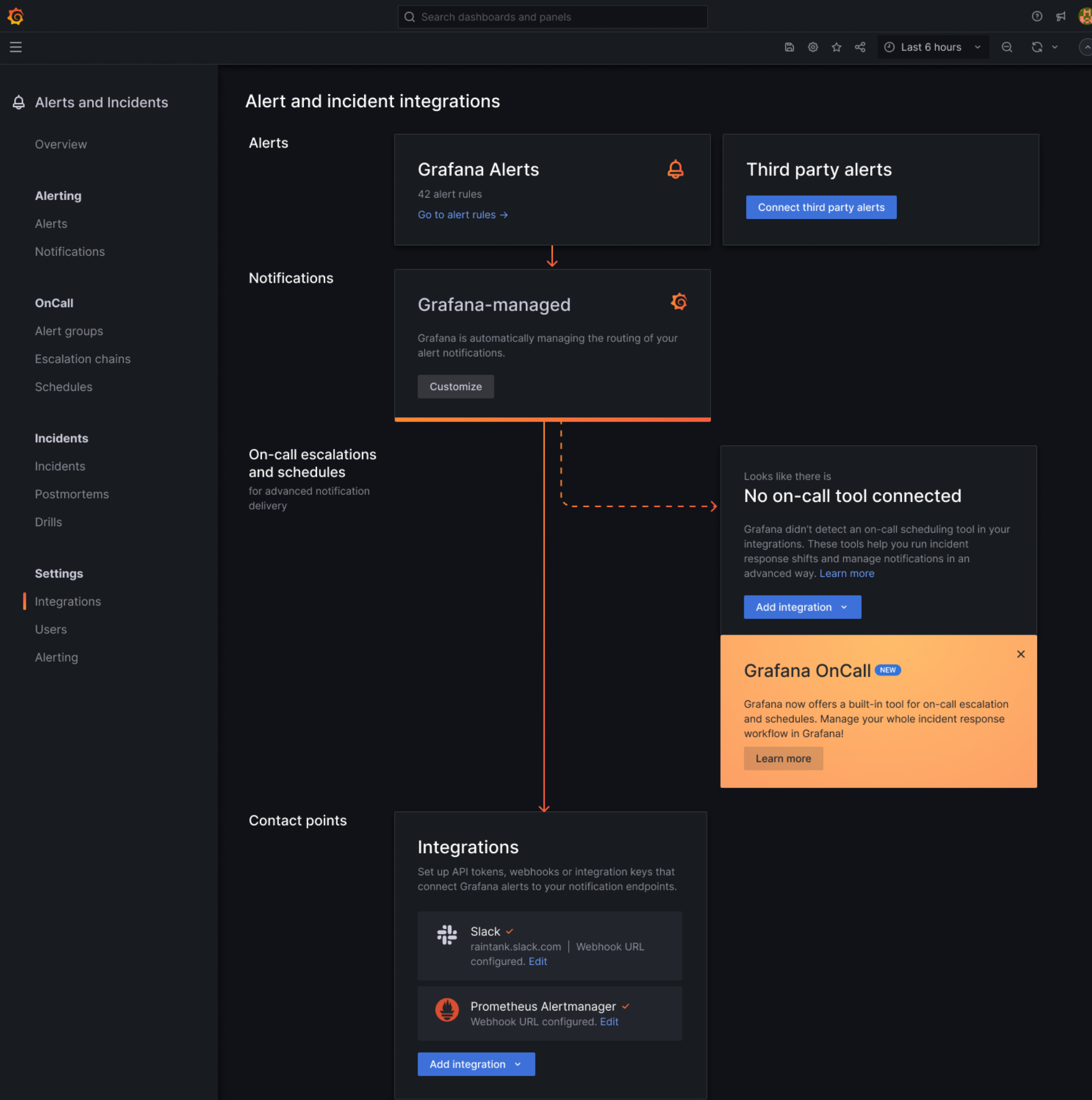
It is hard to understand how each part of the system is connected. Users struggle to understand notifications management between Alerting and OnCall.

SOLUTION

flow-based views for integrations and notifications



I tried to arrange the UI by the way alert data flows through the system. There's alerts, notifications which separated into notification processing and notification delivery, and contact points.



Left column: Grafana, right column: Third party

For integrations, you can either use built-in Grafana tooling or integrate additional things you want to use.

Grafanistas with no background knowledge were still confused and more research and iteration is needed to make sure users understand this

Alert and incident integrations

Alerts

- Grafana Alerts** (42 alert rules, Go to alert rules →)
- Third party alerts** (2 integrations: Datadog alerts, Prometheus Alerts)

Notifications

- Grafana-managed** (Grafana is automatically managing the routing of your alert notifications. Customize)

On-call escalations and schedules (for advanced notification delivery)

- PagerDuty** (Looks like you have connected. Great, your incident response workflow is all set! To change settings, please go to your third-party tool.)
- Grafana OnCall** (NEW) (Grafana now offers a built-in tool for on-call escalation and schedules. Manage your whole incident response workflow in Grafana! Learn more)

Contact points

- Integrations** (Set up API tokens, webhooks or integration keys that connect Grafana alerts to your notification endpoints. Slack, Prometheus Alertmanager)
- On-call integrations** (Set up API tokens, webhooks or integration keys that connect Grafana alerts to your notification endpoints. PagerDuty)

Flow visualization: Orange arrows show the flow from Alerts to Notifications, then to On-call escalations and schedules, and finally to Contact points. A thick orange line highlights the path from Grafana-managed to On-call integrations.

Integrations

Alerts

- Grafana Alerts** (42 alert rules, Go to alert rules →)
- Third party alerts** (2 integrations: Datadog alerts, Prometheus Alerts)

Notifications

- Grafana and Mimir-managed** (Your notification processing automatically happens in both Grafana and Mimir. Mimir supports alert management right in the data source, which is faster and more efficient in large systems. Customize)

On-call escalations and schedules (for advanced notification delivery)

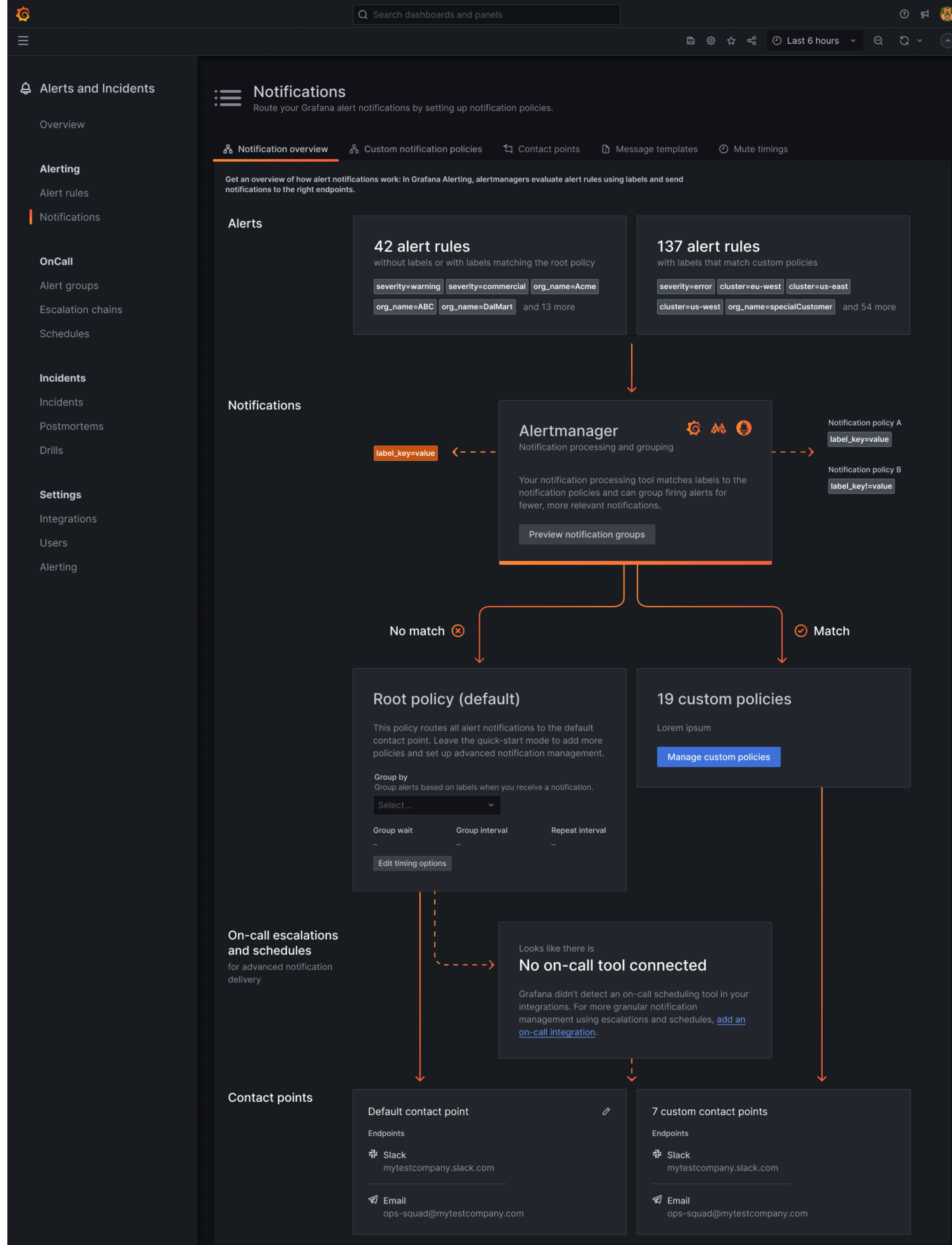
- Grafana OnCall** (You have connected. Great, your incident response workflow is all set! Manage your on-call schedules and notifications directly in Grafana. Go to Grafana OnCall)

Contact points

- Integrations** (Set up API tokens, webhooks or integration keys that connect Grafana alerts to your notification endpoints. Slack, Prometheus Alertmanager)

Flow visualization: Orange arrows show the flow from Alerts to Notifications, then to On-call escalations and schedules, and finally to Contact points. A thick orange line highlights the path from Grafana and Mimir-managed to On-call escalations and schedules.

The flow visualization and options change depending on the user's setup and integrations.



I propose to use similar visualizations on a new "Notification overview" page. The complex notification management area could become easier to understand with this.

“

I love this! The visualization is

so much easier to
understand

than everything we have right now. It's gonna be
really helpful not just for users but even our own
people to understand our system. ”

DEVIN CHEEVERS, GROUP PRODUCT MANAGER

PROBLEM

The complexity presupposes a lot of knowledge and terminology, which can be overwhelming.

Our UIs can be very dense, and could distract users from their key tasks.

SOLUTION

simplified interfaces that cater to user jobs and needs



personas



THE ENGINEER ON-CALL

- Occasional user of Grafana Ops tools, not the most versed
- Sometimes need to work around the things they don't know in a tool
- Thankful for a good runbook



THE ALERT EXPERT

- Has set up and created the alerting system or at least parts of it
- Expert user of Alerting
- Is eager to continually improve the system, but needs the right feedback to know what to change



THE LEADERSHIP STAKEHOLDER

- Mostly interested in performance and SLOs
- Might be an engineering manager who helps with scheduling on-call rotations

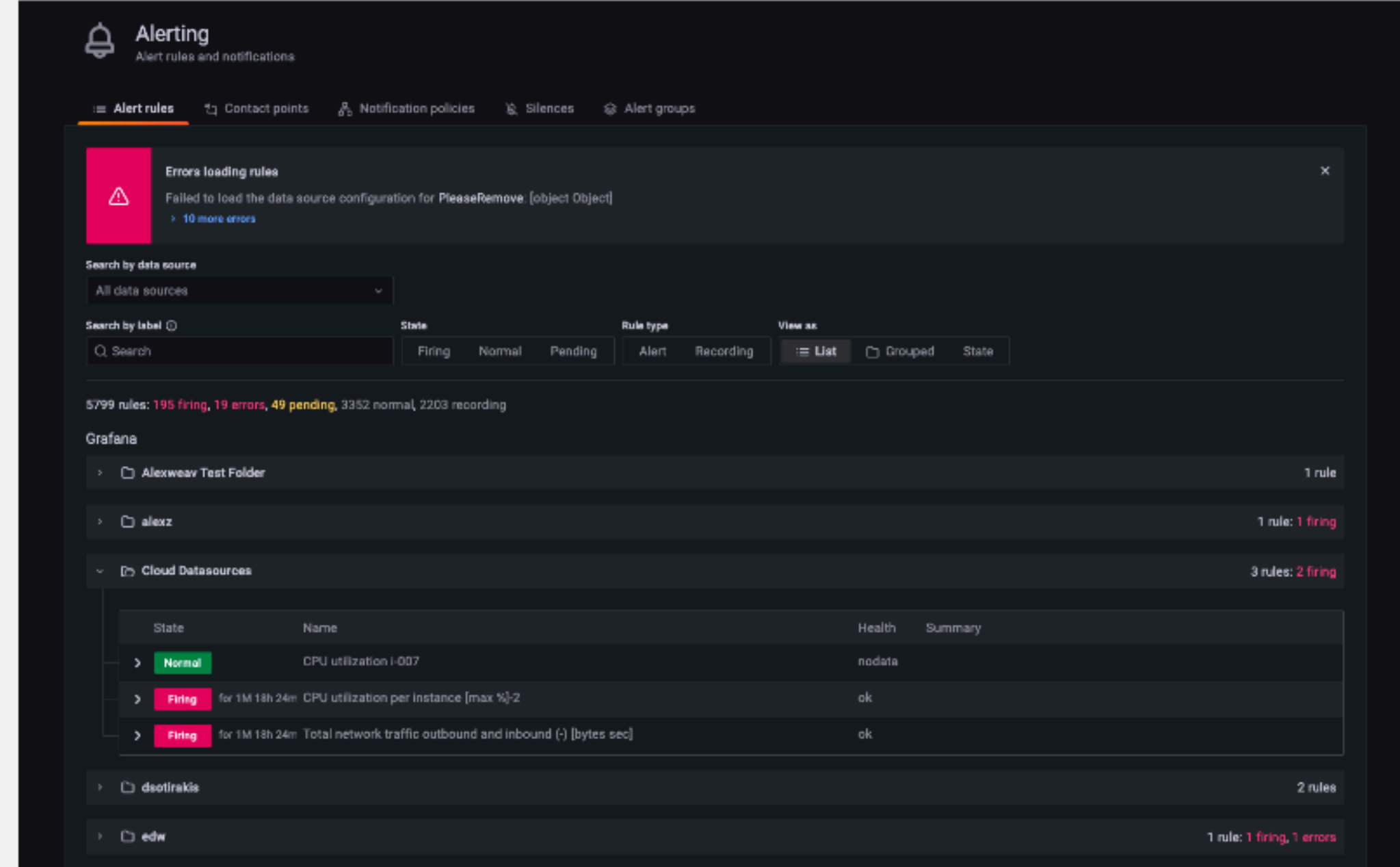
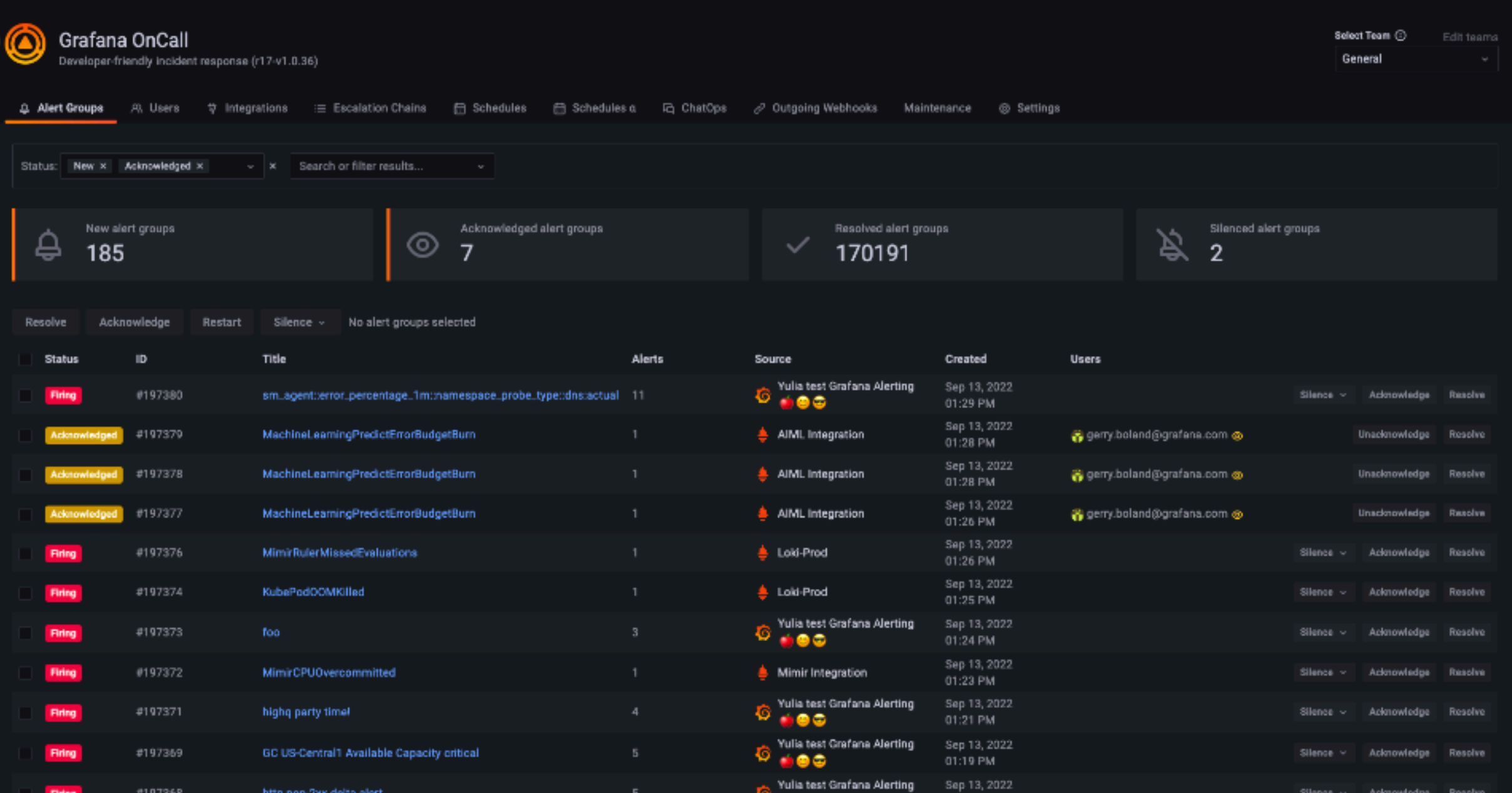


confirmed by
observability
personas
research

I have new tech fatigue.
If I can avoid it, I don't
use the OnCall UI at all,
because it's way too hard.

— A GRAFANA ENGINEER WHO IS OCCASIONALLY ON-CALL





jess
designs
.it

USERS ARE TRYING TO AVOID THE UI.
THEY WILL ONLY OPEN GRAFANA OPS TO

1

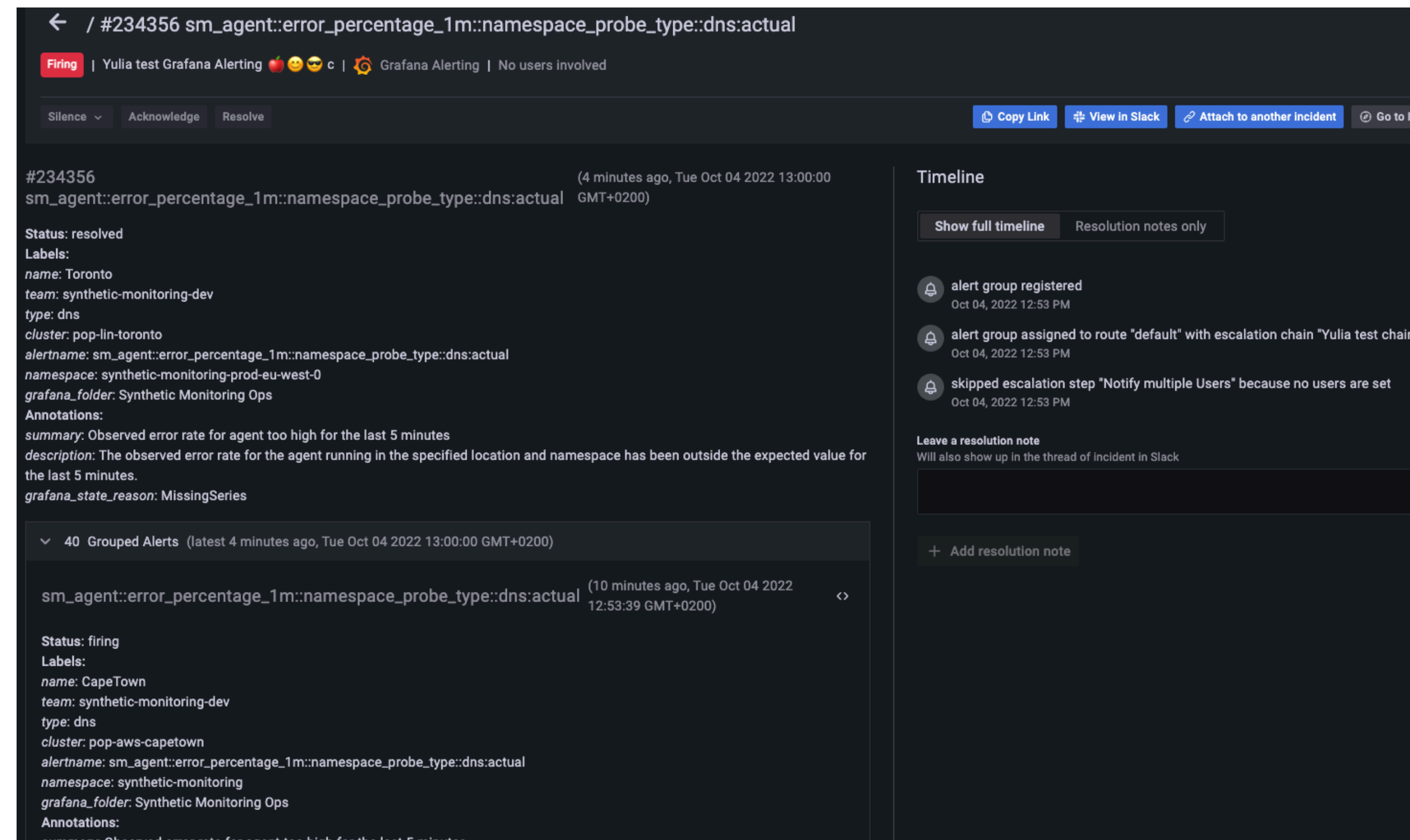
set up the system

2

manage and improve
existing alert rules

3

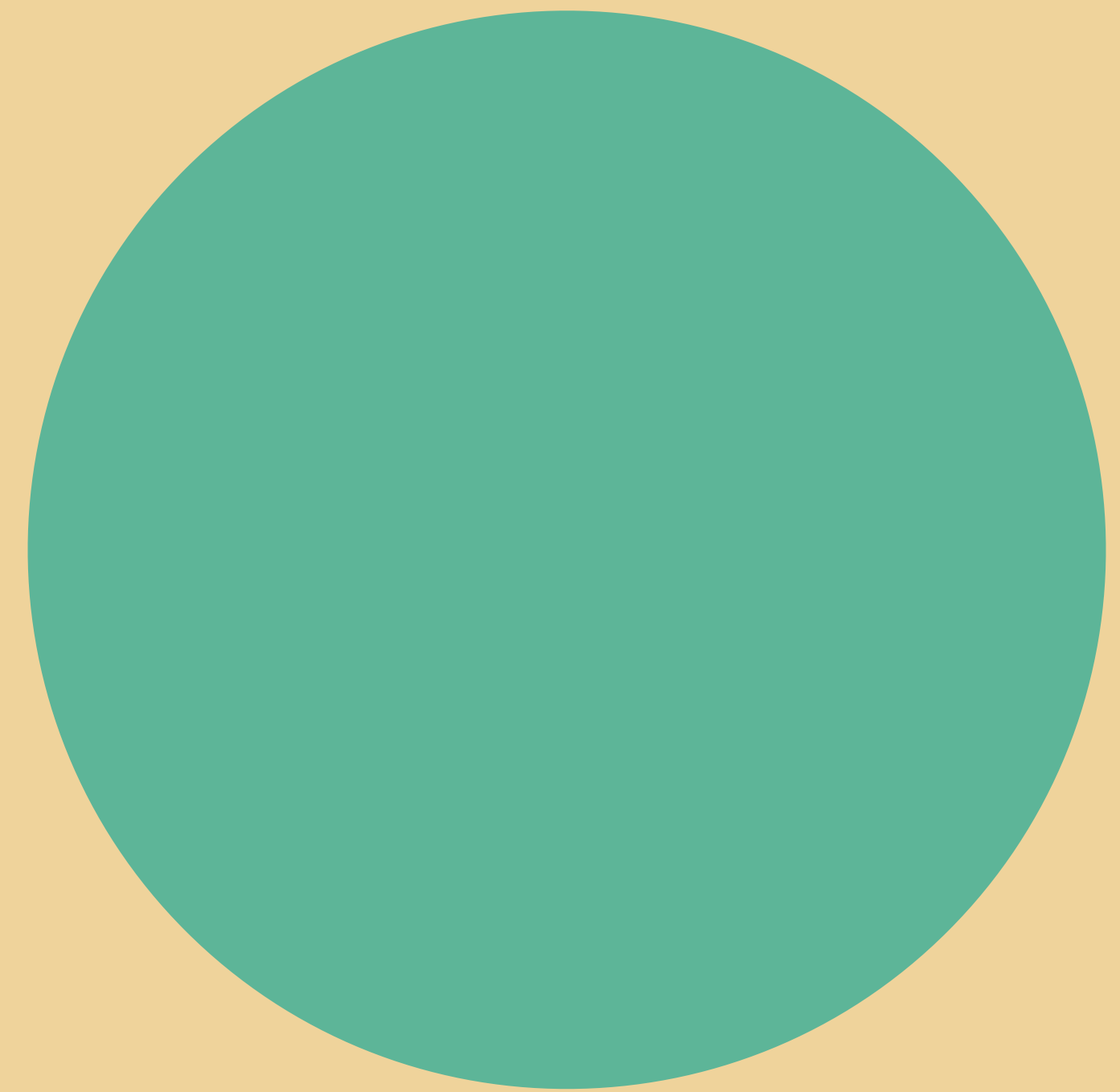
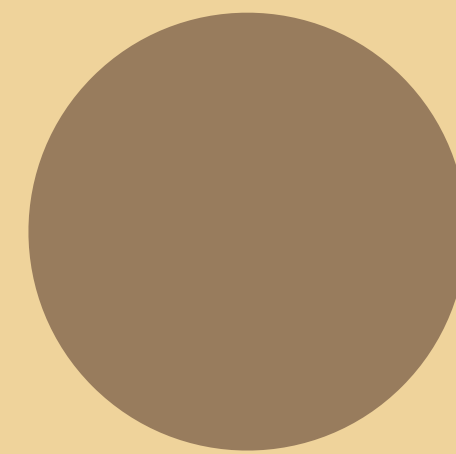
investigate alerts and incidents



GRAFANA OPS NEEDS TO

help personas do
their jobs better

BUT WHAT ARE THEIR JOBS?



MAIN INTERACTION POINT

alert lists and detail pages

USER JOBS



CURRENT SOLUTION

1 see clearly what's breaking

- UIs can be overwhelming: **information overload** and **not enough highlighting** of what's important
- having multiple different list UIs makes it hard and **slows users down in learning** our tools
- within an OnCall alert group, there is no content clarifying why these alerts are grouped together

2 get clear instructions how to fix the problem efficiently

- Grafana doesn't nudge users to add instructions for fixing problems
- from the OnCall alert details, there are no links with further actions



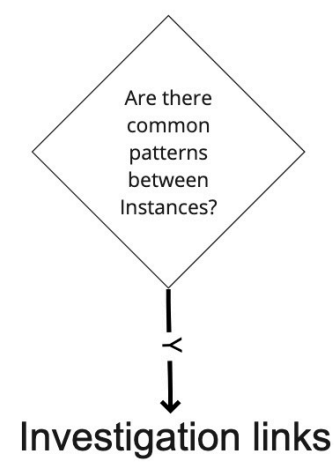
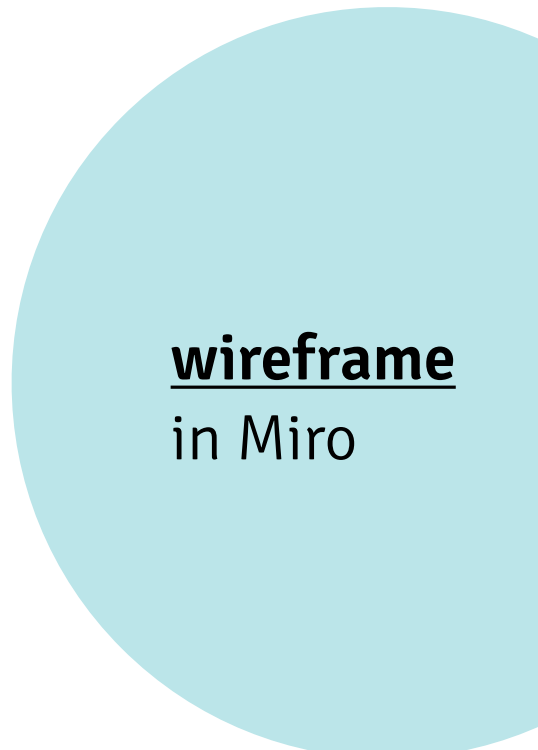
IDEA: SHARED TEMPLATE FOR DETAIL PAGES

used both for Alerting rule view and OnCall group detail view

Give more visual information: **State history** of all instances over time as well as **past incidents**

Allow for **investigating multiple instances together** if they have the same dashboard/runbook/alert rule

Adapt shown content blocks depending on the product



“I’d really like to explore how we allows customers to merge incidents, alert groups and alerts depending on what tools they’re using from us.”

– DEVIN CHEEVERS, GROUP PRODUCT MANAGER



IDEA: SHARED TEMPLATE FOR ALERT-RELATED LISTS

Shown content adapts to both used product and viewer persona through view modes.
Same design and components, different actions and content



idea produced and refined
with **Gilles de Mey**

on-call
investigator
mode

alert rule
tweaking
mode

- Focus on state and timings
- For investigating firing alerts on-call
- == OnCall alert groups

- Focus on folder and evaluation group
- For creating and improving rules
- == Alerting UI folder view



IDEA: SHARED TEMPLATE FOR ALERT-RELATED LISTS

overview

filters

list content

escalation level
indicator



Item name + metadata

Three horizontal rectangular boxes representing item names and metadata, stacked vertically.

Child count
(Instances/Signals)

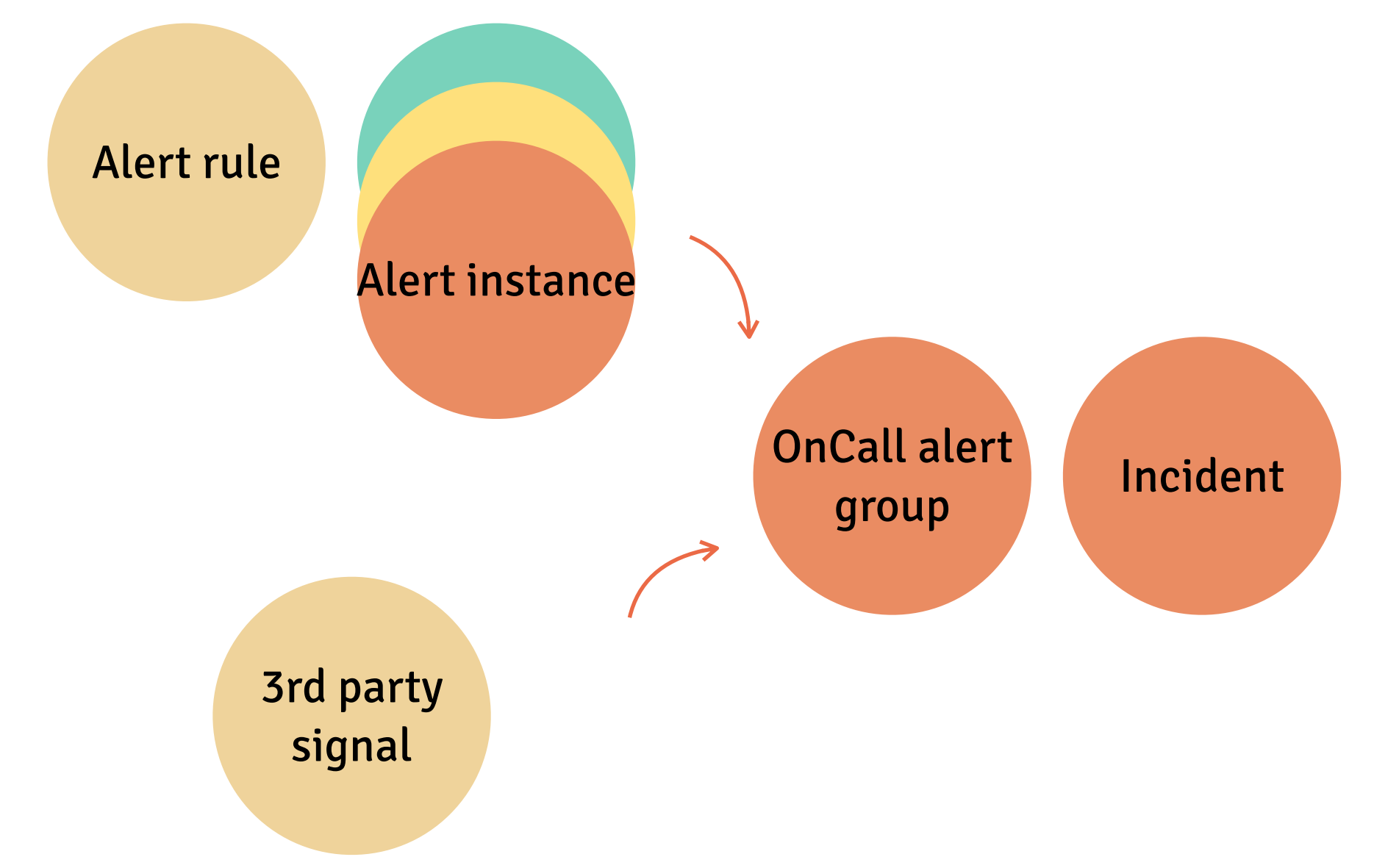
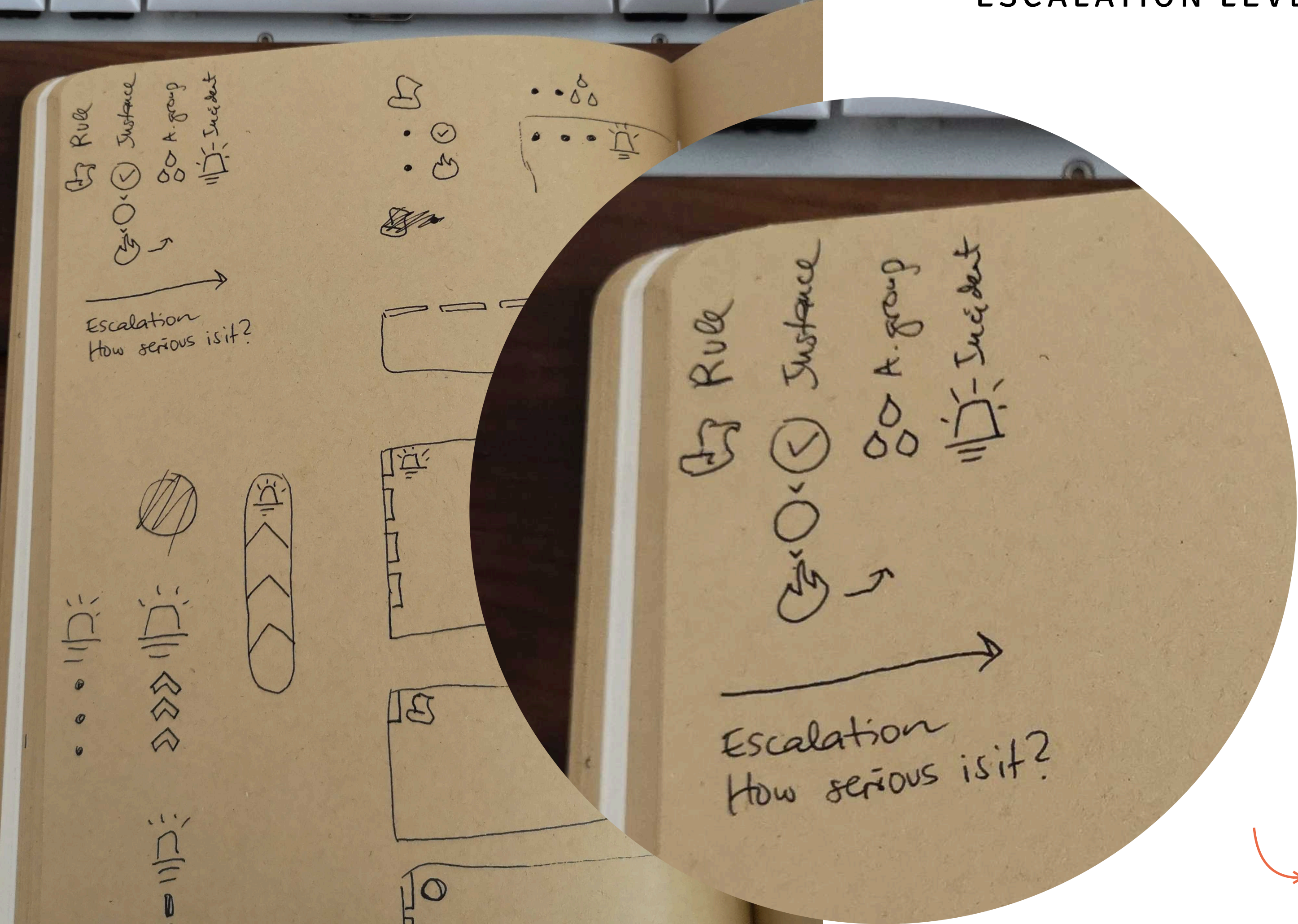
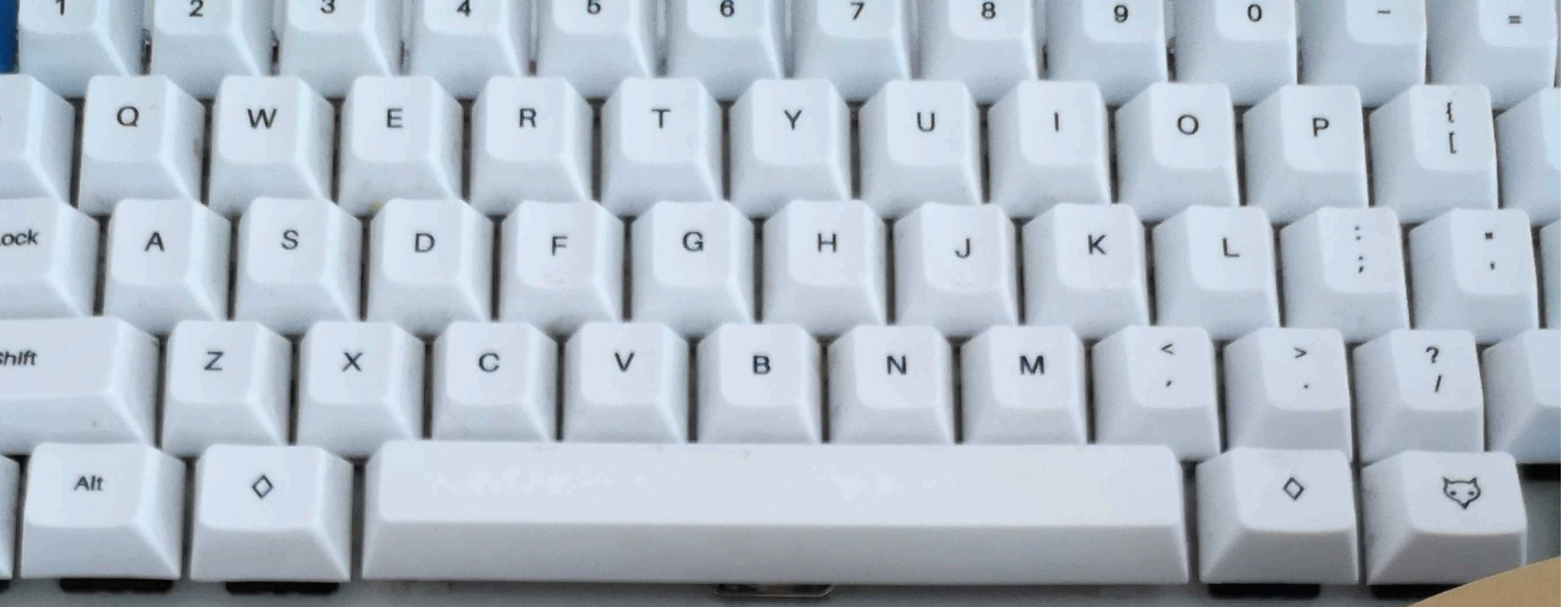
⊗ 5

Action area

Investigate

list content

ESCALATION LEVELS AS DISTINGUISHING CRITERIA



based on a suggestion by
Rob Whelan

The higher the escalation, the more lines. Rule = level 0, Instance = level 1, Alert group=2, Incidents=3

Alert rule

Alert rule name `label=value` `much-longer-label=1243-value-234test`

Server Monitoring / My group 3 linked past incidents

gdev-prometheus > gcloud-ops-ngalertmanager

Instances 5

Alert instances

Instance of "Alert rule name"

`label=value` `much-longer-label=1243-value-234test`

Server Monitoring / My group for 4 minutes 1 linked live incident, 3 past incidents

gdev-prometheus > gcloud-ops-ngalertmanager

Instance of "Alert rule name"

`label=value` `much-longer-label=1243-value-234test`

Server Monitoring / My group for 4 minutes

gdev-prometheus > gcloud-ops-ngalertmanager

Instance of "Alert rule name"

`label=value` `much-longer-label=1243-value-234test`

Server Monitoring / My group last firing on 01.01.2022

gdev-prometheus > gcloud-ops-ngalertmanager

+ action area missing in these draft wireframes

more details in Miro

Alert groups

#123456

Alert group name

17 labels included for 4 minutes 1 linked live incident, 3 past incidents

gdev-prometheus > gcloud-ops-ngalertmanager acknowledged by OnCall-user123

5

Silence Acknowledged Resolve Investigate

Escalate to incident button for alert groups

show labels only if they match a filter

The overlaying lines indicate the escalation level
The more preceding stages, the more serious and the higher the level

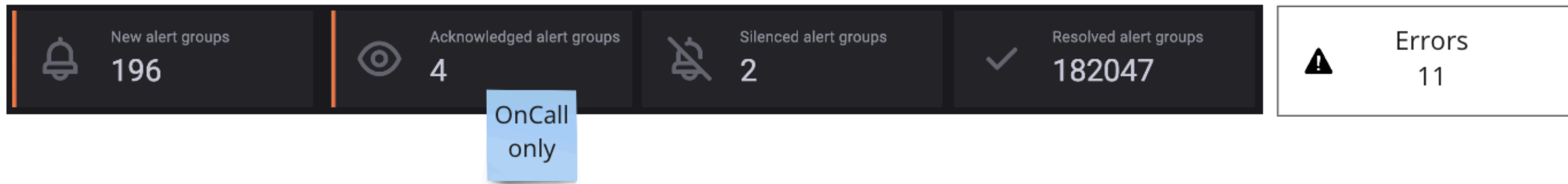
Incidents

Incident list content was not taken into account yet

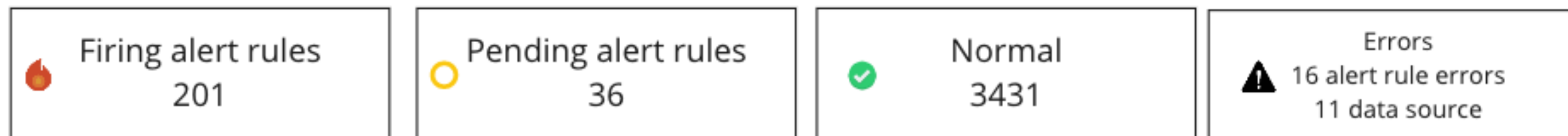
so far only explored Alerting + OnCall, next step could be connecting this with the Incident list

overview

- Use the data cards with filtering capabilities from OnCall
- Allows for a nice way to surface errors together with other summary content



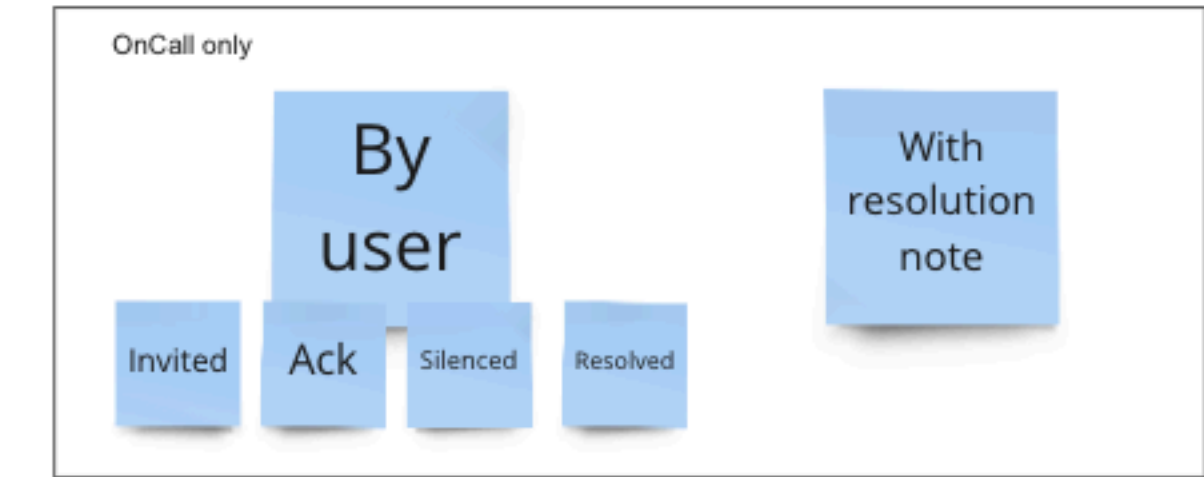
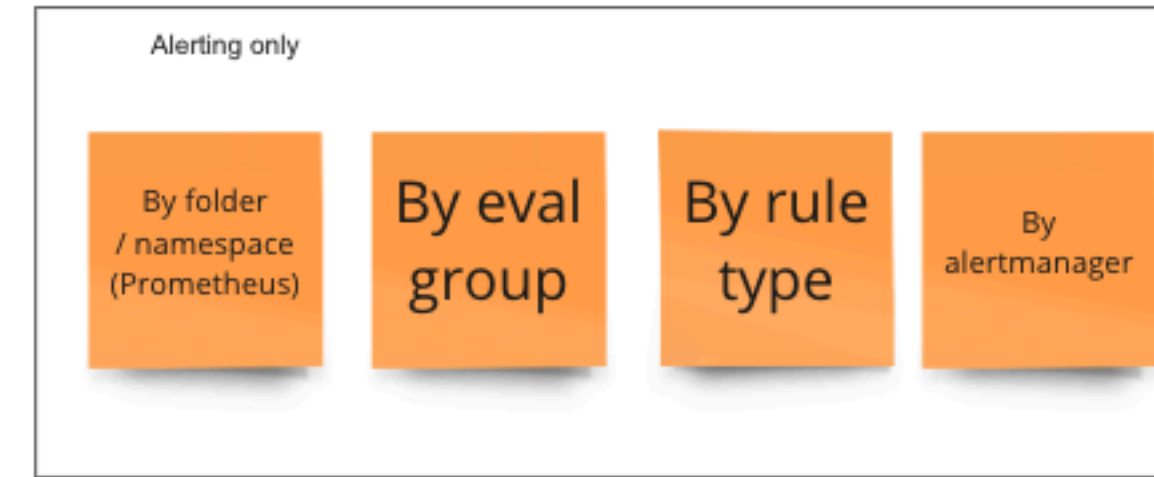
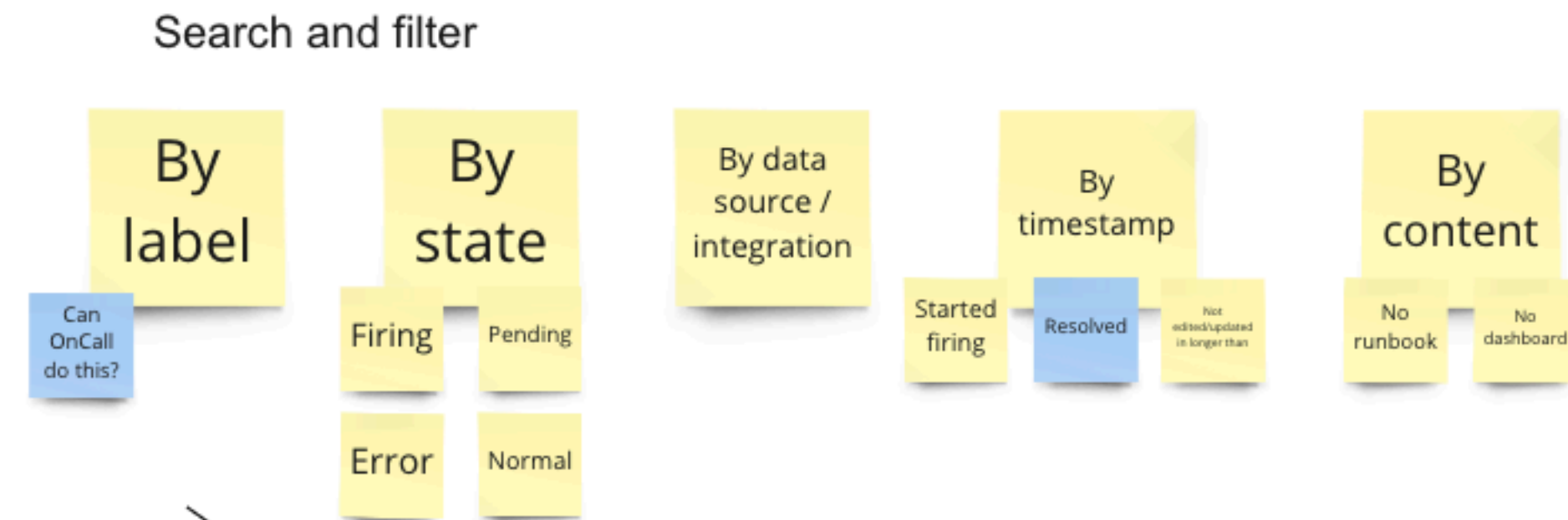
ADAPTED OVERVIEW FOR ALERT RULES LIST



filters

WHICH OPTIONS TO SHOW FOR WHICH PRODUCT?

"As an on-call engineer I mostly only care about instances"



Most needed filters per persona, rest hidden behind "Show all" link



raw exploration
in Miro

put together in collaboration
with **Gilles de Mey**

filters

- show one row of most relevant filters for every mode, with an option to show all
- list all active filters below the actual filter pickers

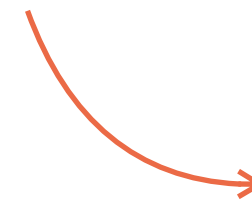
Search in **All alertmanagers** Grafana gcloud-ops-ngalertmanager Sort by Most recently firing first

Label

State

Folder

Data source [Show all filters](#)



Search in **All alertmanagers** Grafana gcloud-ops-ngalertmanager

Label

State

Folder

Data source

Evaluation group

Rule type

Timestamp

Missing content

Label State x Timestamp x [Clear all filters](#)

key=derp

#123456 **CPU usage** key=value much-longer-label=1243-value-234test

Server Monitoring / My group for 4 minutes
gdev-prometheus > gcloud-ops-ngalertmanager OnCall-user123

key=value much-longer-label=1243-value-234test

#123456 **CPU usage** key=value much-longer-label=1243-value-234test

Server Monitoring / My group for 4 minutes



[more details](#)
in Miro

filters

SOME UNRESOLVED QUESTIONS

should we have one text input where you can type in filtering shorthands for quicker usage and less clicking?

How might we show grouping by label across folders/groups /data sources?

How might we introduce filter suggestions to narrow down the search even more? How might we allow users to add criteria to their filtering based on the current results?

based on output from a **UX feedback session**

needs further explorations

other ideas and questions

RELATED TO PERSONAS AND JOBS

- auto-link OSS runbooks if available (depends on data source)
- nudge users to add runbook links during alert creation
- introduce consistent actions for both lists and detail pages

more ideas
in Miro

PROBLEM

Users expect the Ops products to already be interconnected. When they don't see the options they expect, they ask "Why can't I".

SOLUTION

spiderweb connections
between products

surface connections

The list UIs as key interaction point in Grafana Ops have a lot of potential to bring awareness to how alert rules, instances, groups and incidents are based on each other.

INCIDENTS IN ALERT LIST

The image shows a horizontal alert list item with several annotations. On the left, a yellow box labeled 'Alert groups' points to a square checkbox. Below it, another yellow box says 'show labels only if they match a filter' pointing to the label 'gdev-prometheus > gcloud-ops-ngalertmanager'. The main alert card contains: a fire icon, the ID '#123456', the title 'Alert group name', and details '17 labels included for 4 minutes 1 linked live incident, 3 past incidents acknowledged by OnCall-user123'. To the right are icons for 5 live incidents (fire) and 1 past incident (bell). Action buttons include 'Silence', 'Acknowledged', 'Resolve', 'Investigate' (with a dropdown arrow), and an incident icon. A yellow box at the top right says 'if you have runbook/dashboard etc, don't show a dropdown but link to detail page' pointing to the 'Investigate' button. A red arrow points from the incident counts to the text 'show linked live and past incidents'. Another red arrow points from the incident icon button to the text 'Escalate to Incident action'.

SHOW INCIDENT HISTORY WITH STATE HISTORY

The image shows a timeline visualization for incident history. The y-axis lists 'backend_01', 'backend_02', and 'backend_03'. The x-axis represents time. Vertical bars in red and blue indicate incident periods across the backends. A yellow box highlights a specific incident period on 'backend_01'. A red arrow points from this box to the text 'Visual highlighting of areas linked to incidents'. The text 'past incidents' is written above the timeline.

to do: audit for connecting actions

We have heard many requests for our products to allow for connecting actions. Some ideas are listed below, but there is probably more. This is a to-do for my successor to explore and identify all the things we could do.



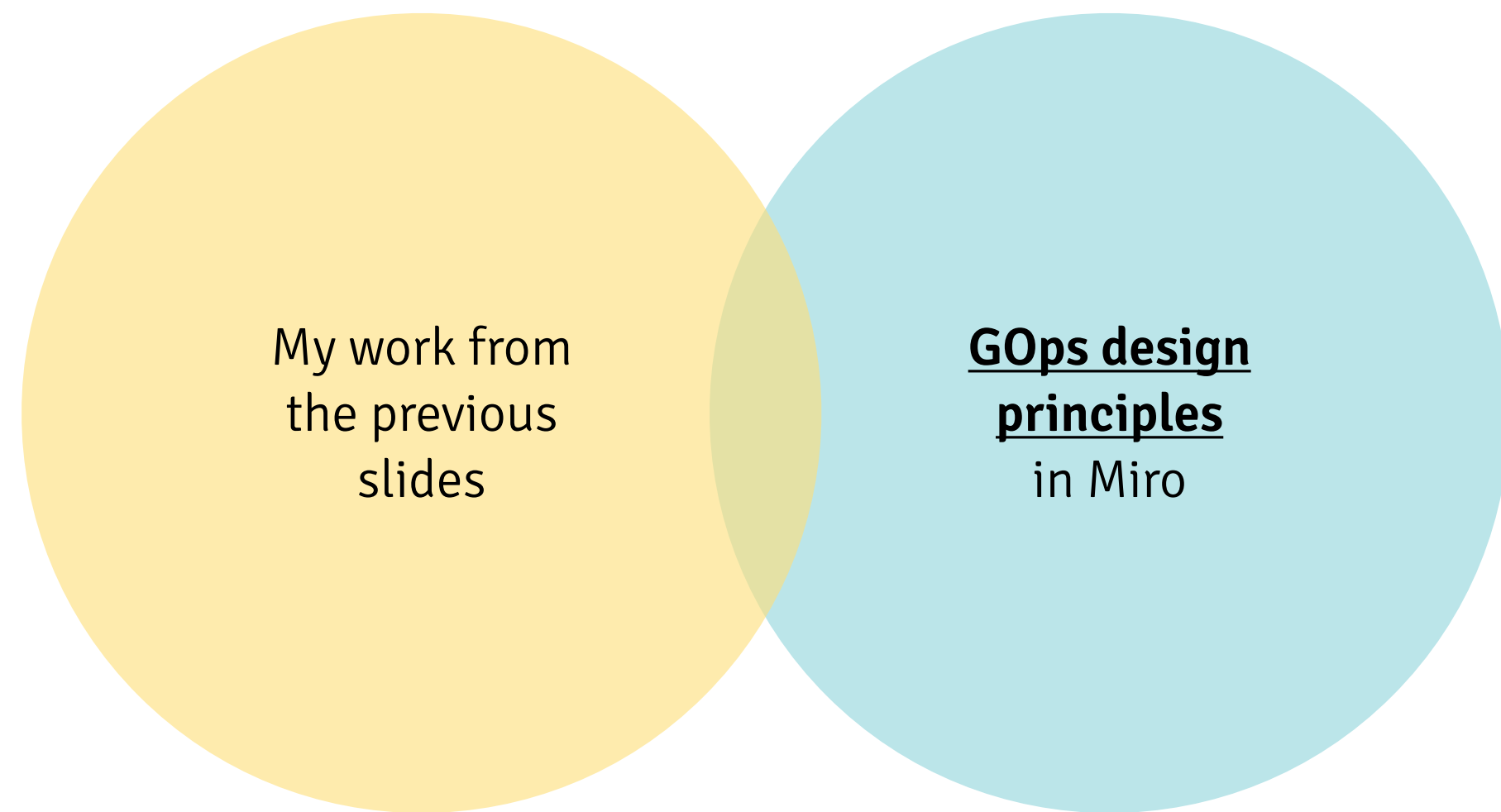
CREATE INCIDENT FROM DASH / WITHOUT ALERT GROUP

ADD OPTION TO CREATE INCIDENTS IN MESSAGE TEMPLATES

KEEP CONTEXT AND INFORMATION FROM PREVIOUS SOURCES WHEN ESCALATING TO INCIDENT

SLO, ALERT AND RUNBOOK ADJUSTMENTS AS PART OF POST-MORTEM

???



NEXT STEPS

design principles check

We have some solution ideas now, but I didn't have time to reevaluate them using our GOps design principles. This still needs to happen!

THE FUTURE

usability tests

Product-market fit validation is key.

Building solutions that don't solve customer problems is a waste of resources.

Test early and often and save on development costs that don't bring value.

PROPOSED STEPS

4-6 usability tests with real customers
per proposed solution, prioritizing adoption
workflows

AND THEN

iterate

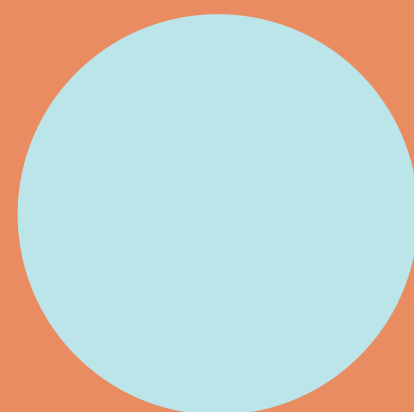
WHY THIS WORK IS VALUABLE

provide a knowledge base

My work has uncovered gaps and produced much-needed alignment and aggregation of information.

“Jess actually understands these concepts, just like an engineer!”

- GILLES DE MEY, TECH LEAD ALERTING FRONTEND



“The workflow map will be really useful even for engineers.”

- ROB WHELAN, ENGINEERING MANAGER



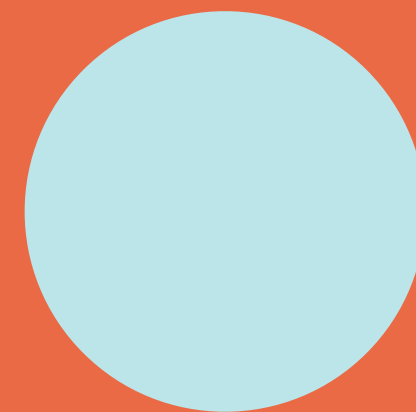
WHY THIS WORK IS VALUABLE

provide a big design vision

The team focus from the product and engineering side has been tactical. People have talked about things we should be doing and that we need unification, but it was my job in this project to define what that actually means.

“This looks awesome!”

- MARC CHIPOURAS, DIRECTOR OF
ENGINEERING GOPS

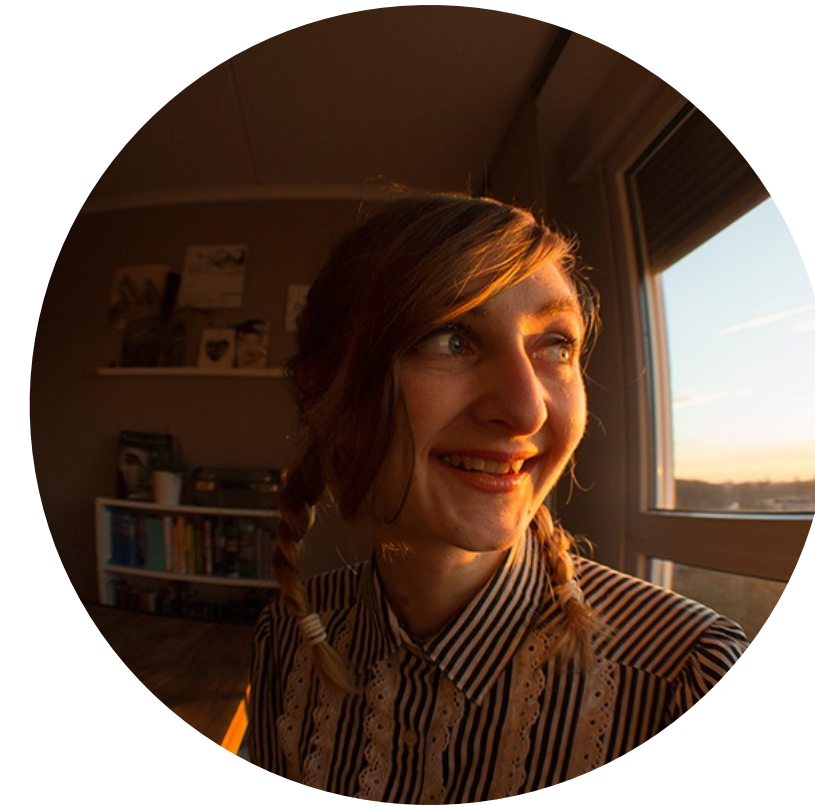


“This is exactly what our teams need to build a product that truly meets our users needs. Absolutely brilliant!”

- MARY SITZENSTATTER, UX MANAGER



THANK YOU! QUESTIONS?



Jessica Matz
hello@jessdesigns.it

Read more about me and find my
portfolio at
www.jessdesigns.it