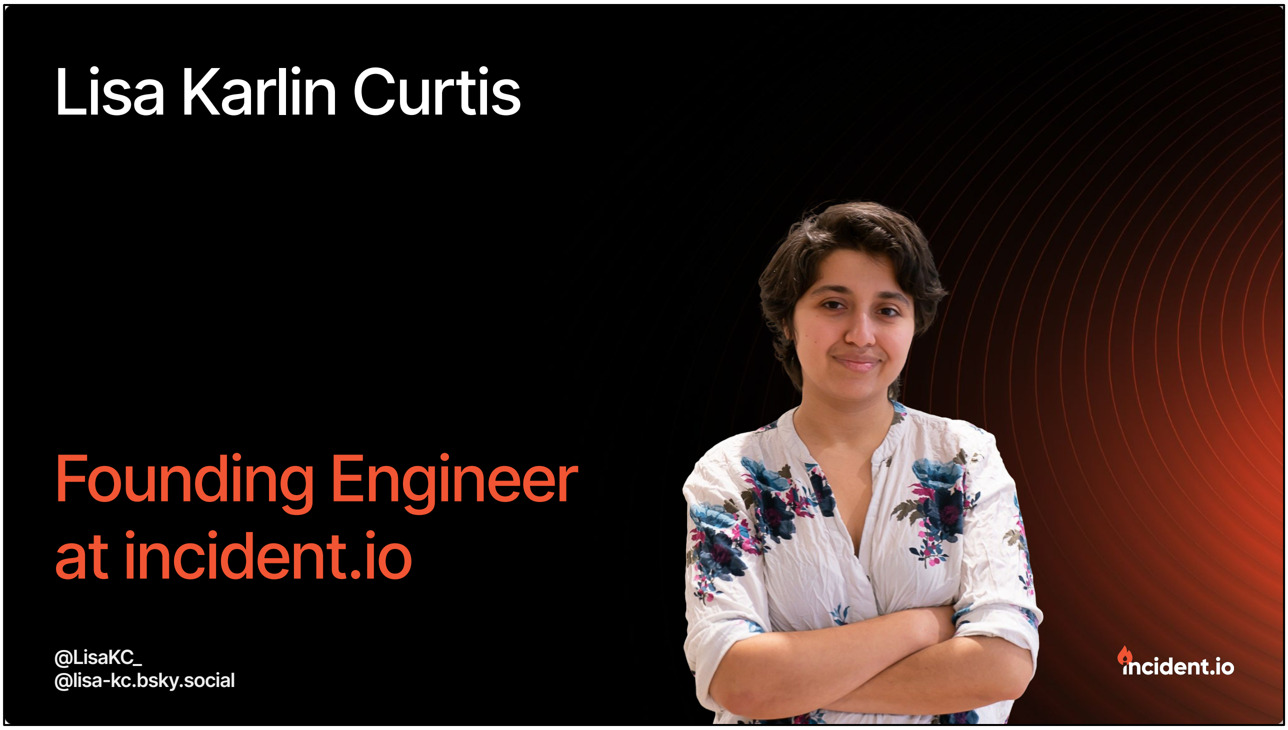
Why you're doing service catalogs wrong

Lisa Karlin Curtis
@LisaKC_
@lisa-kc.bsky.social

incident.io



Lisa Karlin Curtis

Founding Engineer at incident.io

@LisaKC_
@lisa-kc.bsky.social

incident.io

First: a quick intro. I'm Lisa Karlin Curtis, one of the founding engineers at incident.io. We are an incident management platform - if you're interested do come and find me after the talk.

I've spent the last 2 years working on our Catalog, and through that time, I've talked to a lot of customers about their experiences building and maintaining their own catalogs.

Why you’re doing service catalogs wrong

Lisa Karlin Curtis
@LisaKC_
@lisa-kc.bsky.social

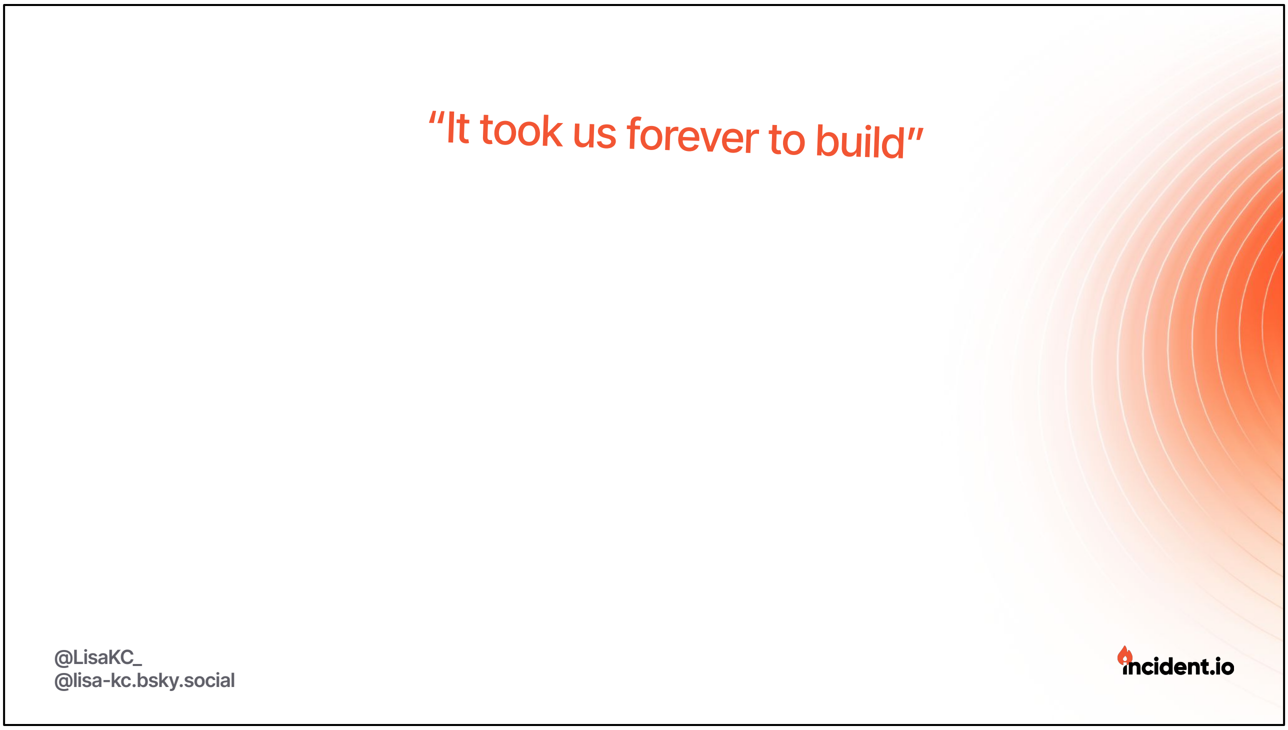
incident.io

Catalogs promise a lot of things.

At my previous role, we’d implemented a service catalog and it hadn’t been a great experience. We’d spent a huge amount of effort on it, and ended up with something that was **ok** but didn’t deliver nearly enough value to justify the effort.

Pretty much everyone agrees that service catalogs are useful, so when we started talking to customers I was waiting to find out not just all the amazing things that they’d done, but finally understand how we’d got it so wrong.

And don’t get me wrong, I heard some amazing stories about the cool things that people are now doing with their Catalogs. But I also heard the same things over and over again which felt hauntingly familiar:

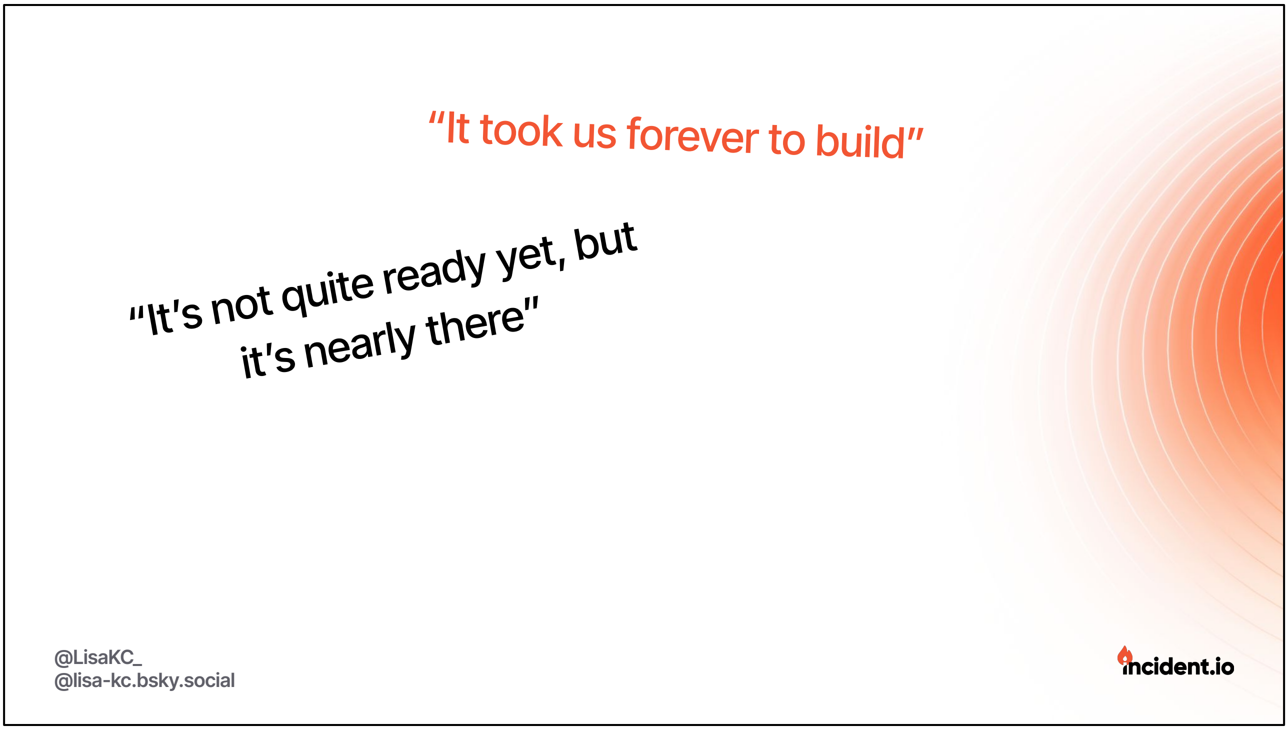


“It took us forever to build”

@LisaKC_
@lisa-kc.bsky.social

incident.io

- It took us forever to build it



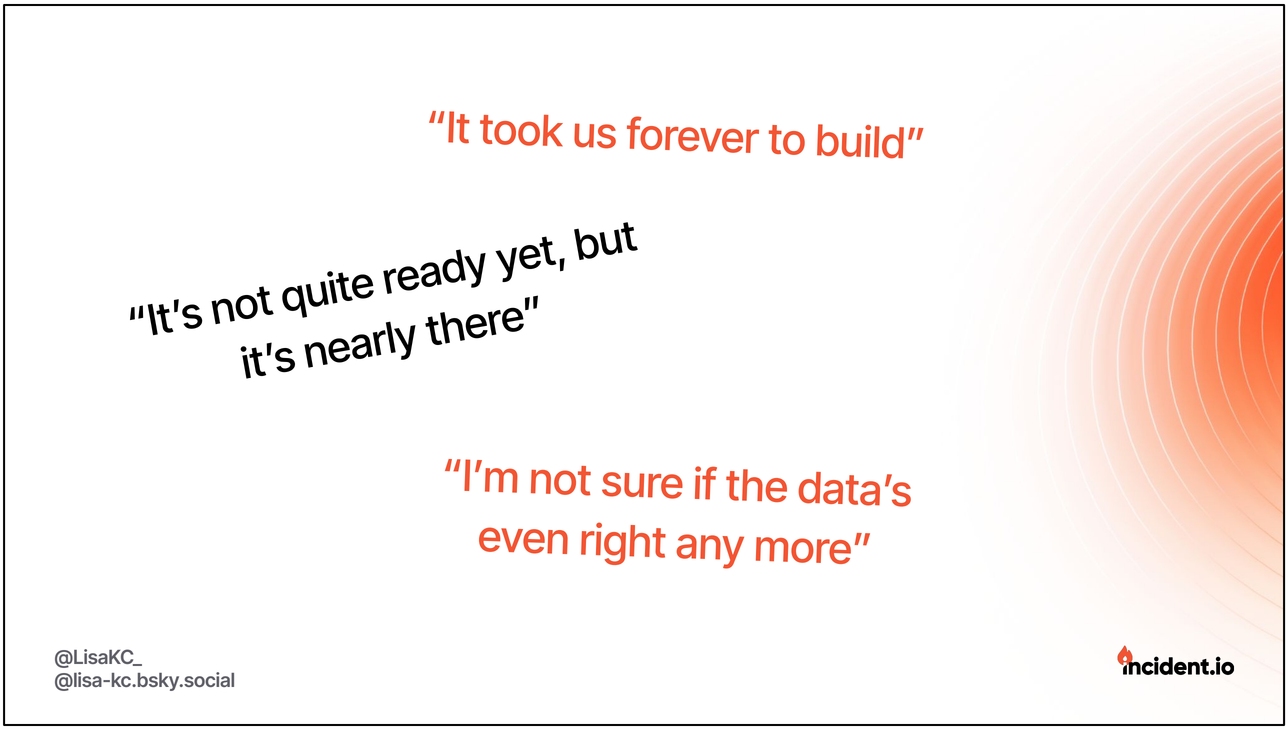
"It took us forever to build"

"It's not quite ready yet, but it's nearly there"

@LisaKC_
@lisa-kc.bsky.social

incident.io

- It's not quite ready yet, but it's nearly there



"It took us forever to build"

"It's not quite ready yet, but it's nearly there"

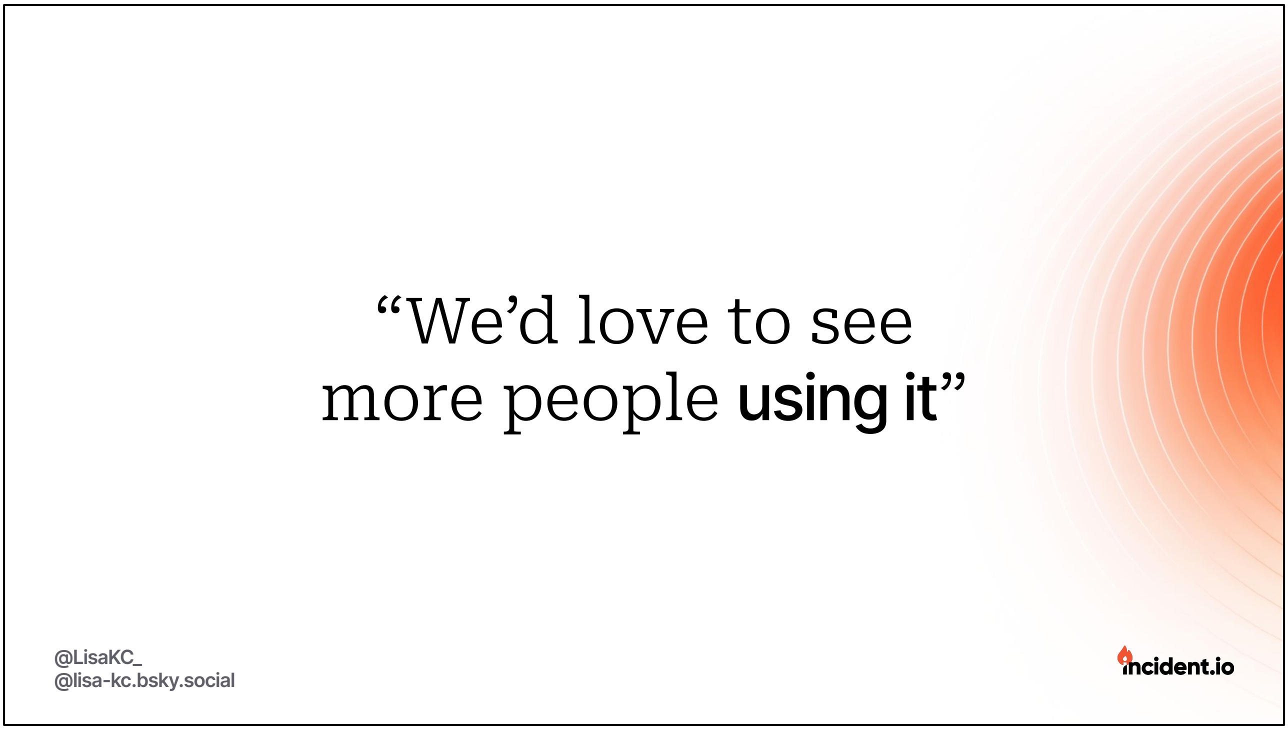
"I'm not sure if the data's even right any more"

@LisaKC_
@lisa-kc.bsky.social

incident.io

- I'm not sure if the data's even right any more

And almost everyone said



“We’d love to see more people **using it**”

@LisaKC_
@lisa-kc.bsky.social

incident.io

We’d love to see more people using it

So I started to wonder, maybe my experience in my last role wasn’t so unique after all. And maybe there is a better way.

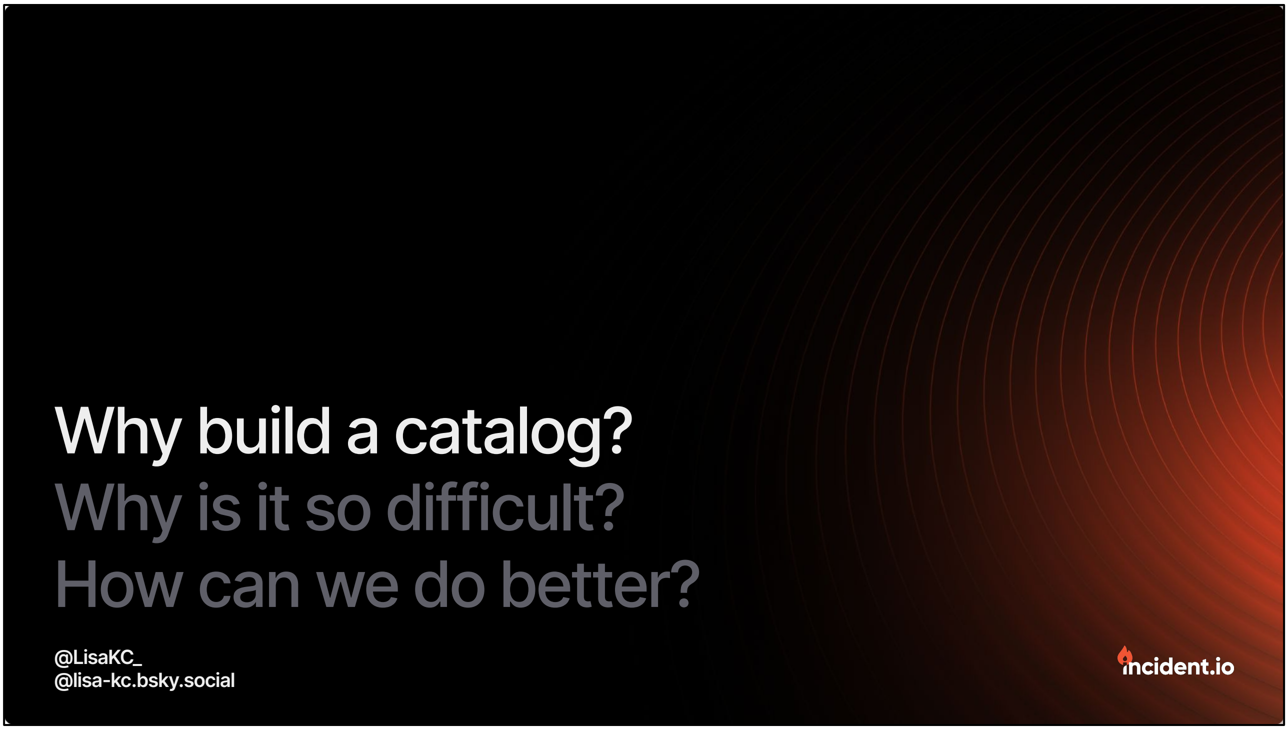
Why build a catalog?
Why is it so difficult?
How can we do better?

@LisaKC_
@lisa-kc.bsky.social

incident.io

Today, we’ll walk through

1. Why build a Catalog
2. What makes building a Catalog so hard
3. How can we build it better

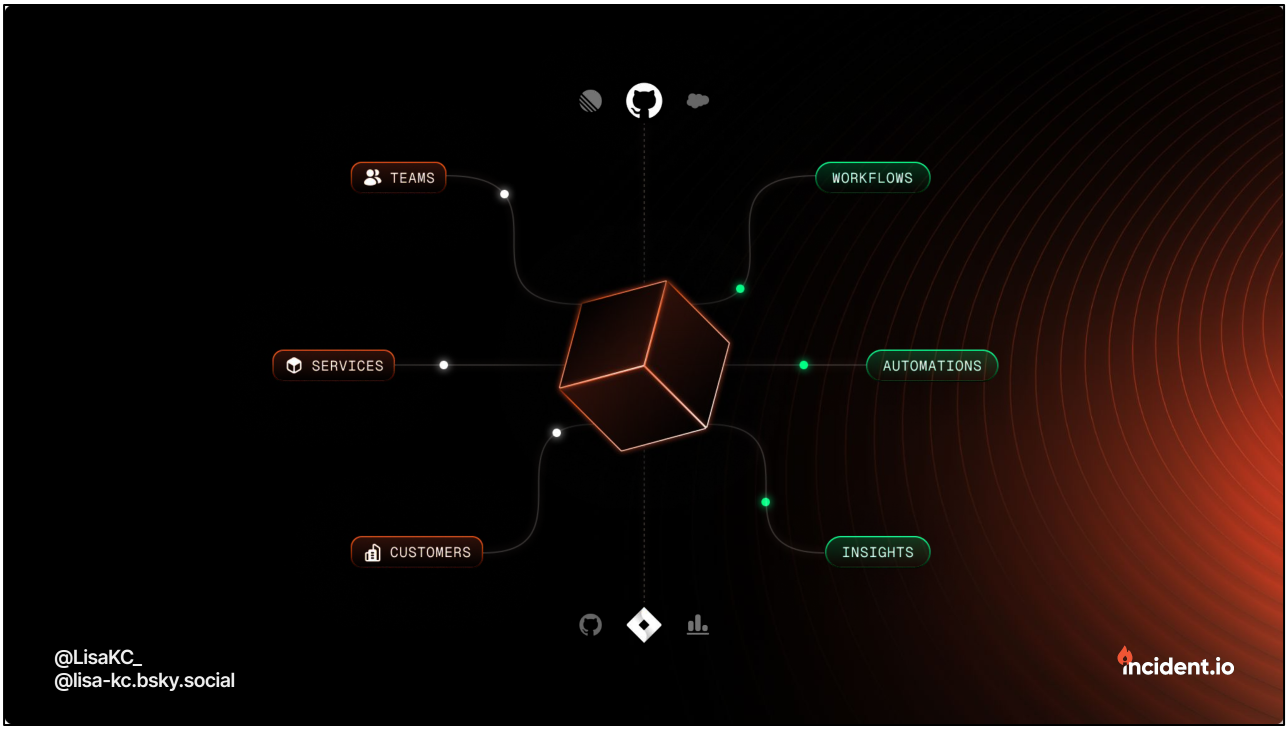


Why build a catalog?
Why is it so difficult?
How can we do better?

@LisaKC_
@lisa-kc.bsky.social

incident.io

…



TEAMS

WORKFLOWS

SERVICES

AUTOMATIONS

CUSTOMERS

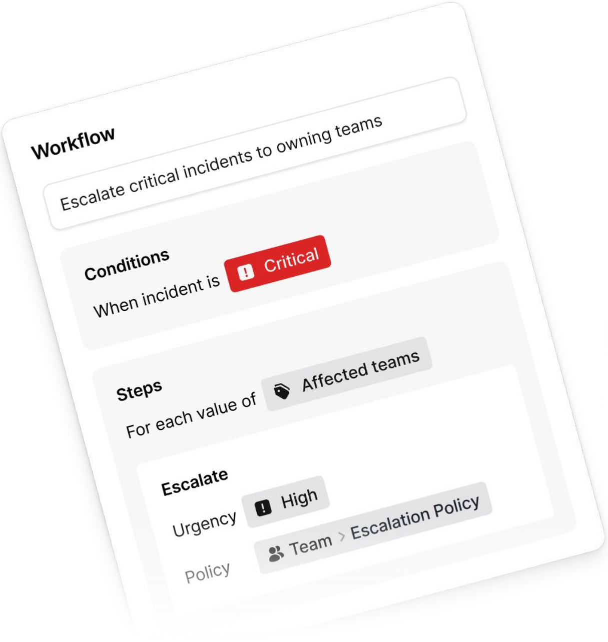
INSIGHTS

@LisaKC_
@lisa-kc.bsky.social

incident.io

At incident, we talk about the catalog as a **connected map** of everything in your organisation - when we were building ours, we joked about the tagline `connecting everything everywhere all at once`. We didn't, because we didn't want to get sued, but I stand by it as an explanation of what a great catalog is.

And really it's a database, it might store your services, teams, infra information catalog. But ultimately, it's a list of things, and connections between those things.



Workflow

Escalate critical incidents to owning teams

Conditions

When incident is ! Critical

Steps

For each value of Affected teams

Escalate

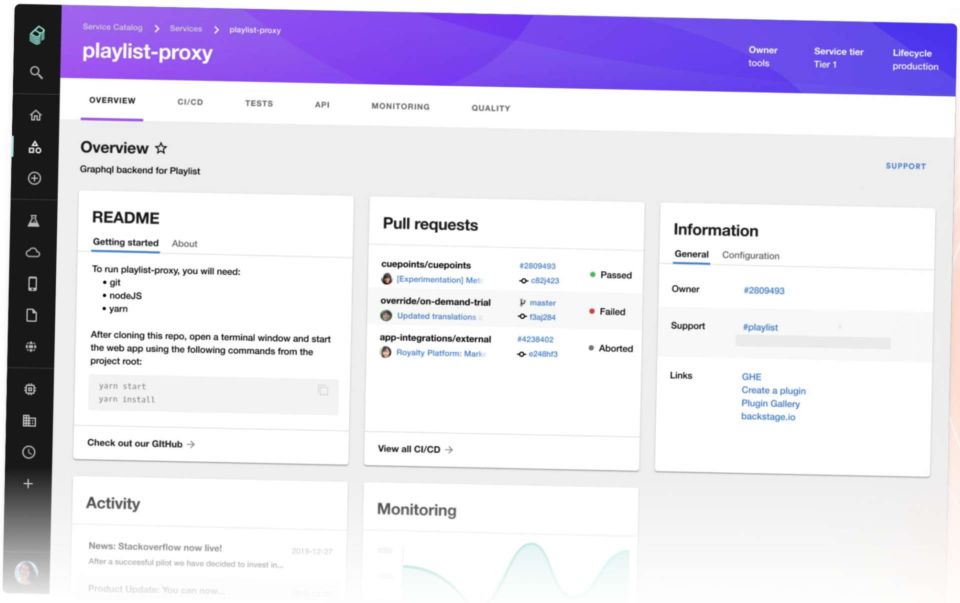
Urgency ! High

Policy Team › Escalation Policy

@LisaKC_
@lisa-kc.bsky.social

incident.io

You can use a catalog to decide which team to page when something goes wrong



Service Catalog > Services > playlist-proxy

playlist-proxy

Owner
tools

Service tier
Tier 1

Lifecycle
production

OVERVIEW CI/CD TESTS API MONITORING QUALITY

Overview ☆

Graphql backend for Playlist

SUPPORT

README

Getting started About

To run playlist-proxy, you will need:
- git
- nodeJS
- yarn

After cloning this repo, open a terminal window and start the web app using the following commands from the project root:

```
yarn start
yarn install
```

Check out our GitHub →

Pull requests

cuepoints/cuepoints [Experimentation] Met	#2809493 c82j423	● Passed
override/on-demand-trial Updated translations	master f3aj284	● Failed
app-integrations/external Royalty Platform: Mark	#4238402 e248hf3	● Aborted

View all CI/CD →

Information

General Configuration

Owner	#2809493
Support	#playlist
Links	GHE Create a plugin Plugin Gallery backstage.io

Activity

News: Stackoverflow now live! 2019-12-27
After a successful pilot we have decided to invest in...

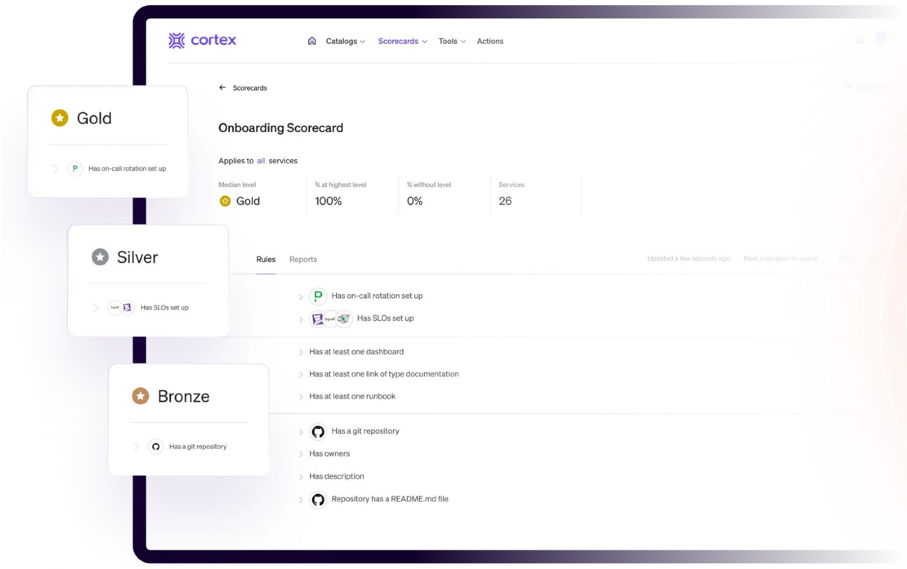
Product Update: You can now...

Monitoring

@LisaKC_
@lisa-kc.bsky.social

incident.io

You can use it to give you a birds eye view of a service, with useful links like github pull requests, incidents, monitoring dashboards etc into a single user interface.



Gold

Has on-call rotation set up

Silver

Has SLOs set up

Bronze

Has a git repository

← Scorecards

Onboarding Scorecard

Applies to all services

Median level	% at highest level	% without level	Services
Gold	100%	0%	26

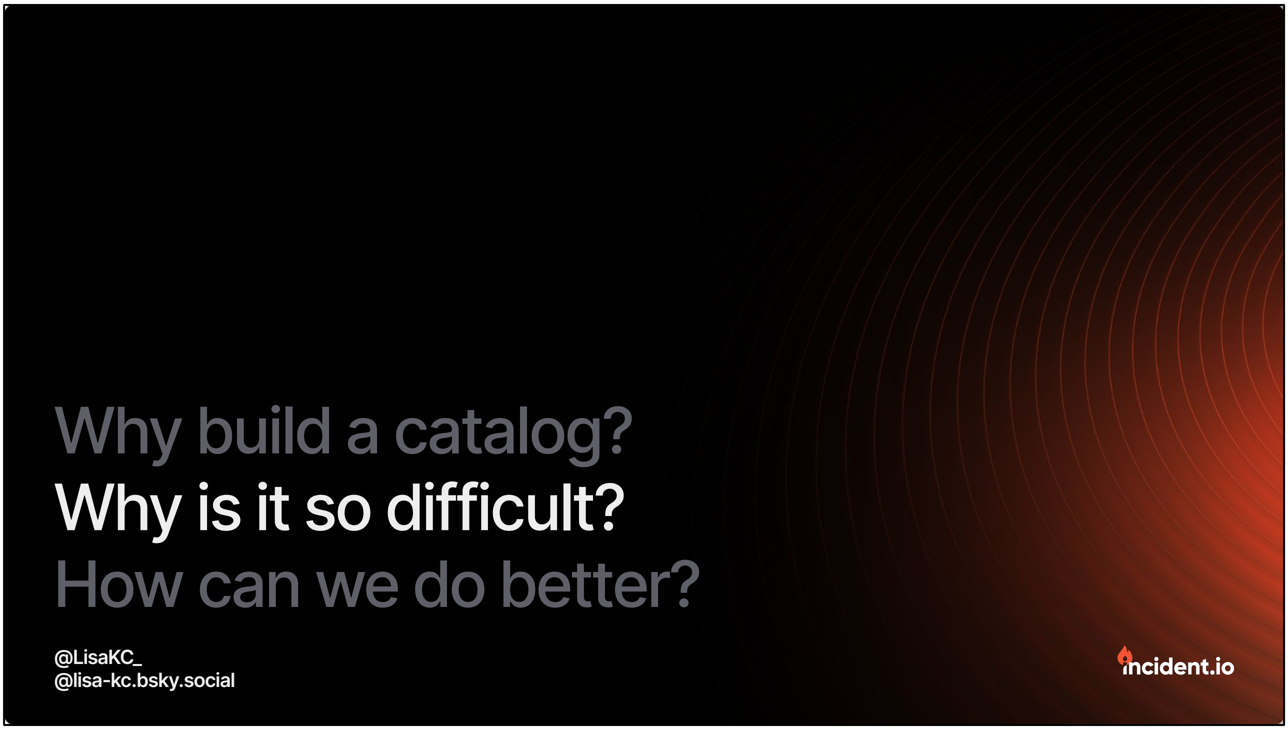
Updated a few seconds ago Next evaluation in queue

- Has on-call rotation set up
- Has SLOs set up
- Has at least one dashboard
- Has at least one link of type documentation
- Has at least one runbook
- Has a git repository
- Has owners
- Has description
- Repository has a README.md file

@LisaKC_
@lisa-kc.bsky.social

incident.io

Or help avoid the pre-audit scramble - collecting data as-you-go throughout the year you can make your annual audits much less painful.



Why build a catalog?

Why is it so difficult?

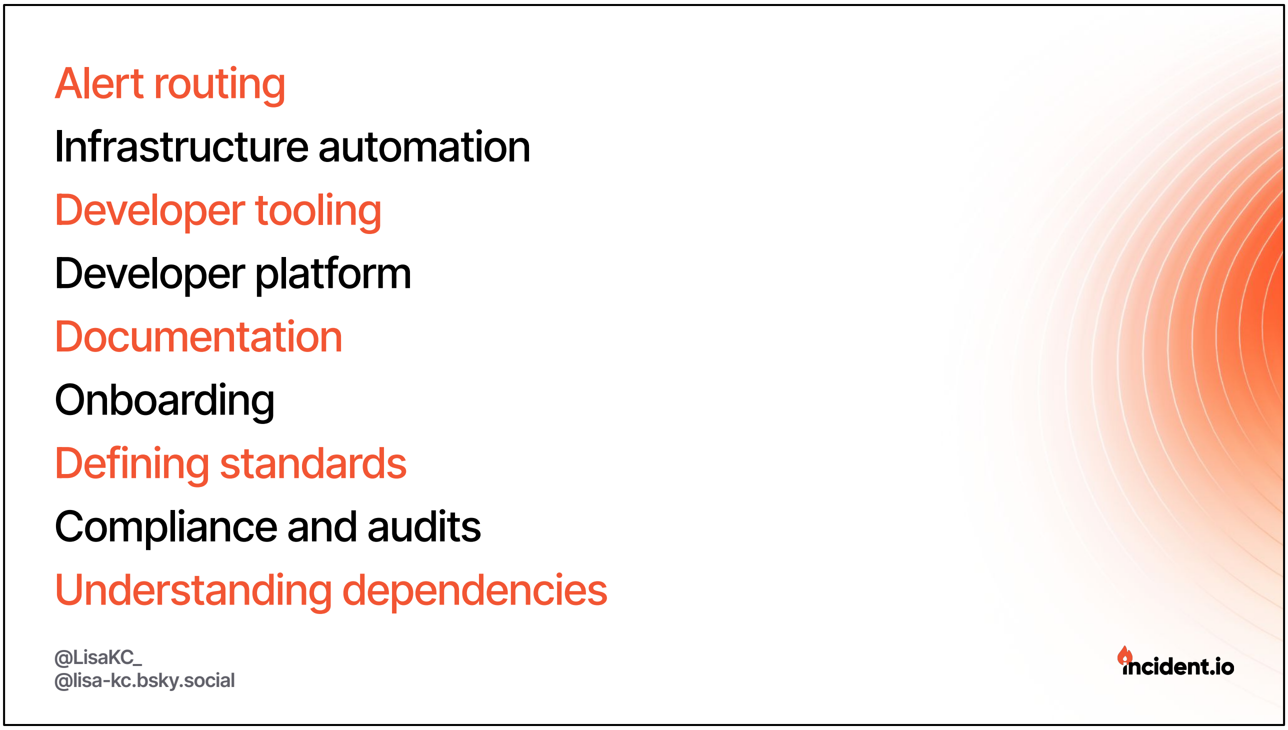
How can we do better?

@LisaKC_
@lisa-kc.bsky.social

incident.io

So, what makes building a catalog so difficult?

To explain this, I’m going to tell a story. This is really a combination of a number of stories that I’ve been told about what implementing a Catalog feels like, and how it can go quite wrong.



Alert routing

Infrastructure automation

Developer tooling

Developer platform

Documentation

Onboarding

Defining standards

Compliance and audits

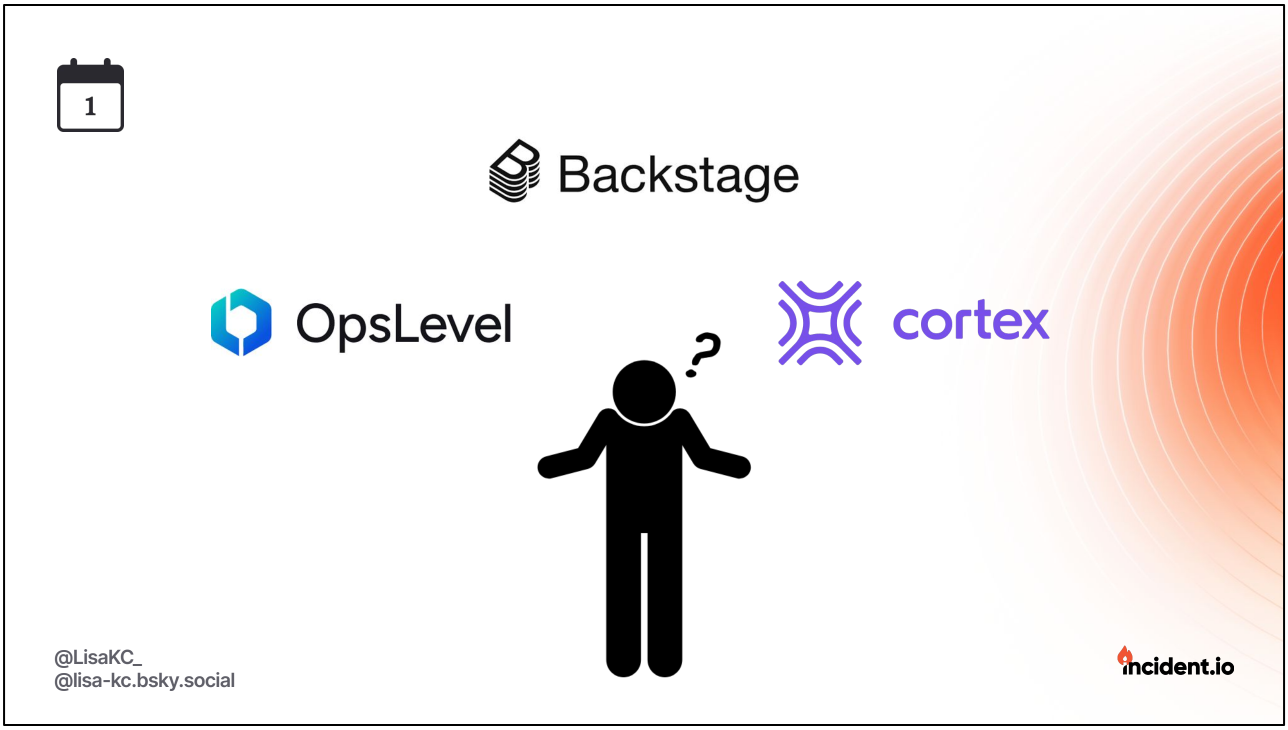
Understanding dependencies

@LisaKC_
@lisa-kc.bsky.social

incident.io

The story starts with lofty ambitions, and big ideas. You've gone to a tech conference and listened to someone like me yap on about how great a service catalog is.

You go back, and you talk to engineering managers and tech leads from across the org, making a list of their pain points which you think a Catalog can solve. You put together a business case, and it becomes the next big-ticket item on your roadmap. How long will it take? I don't know - a couple months?



1

Backstage

OpsLevel

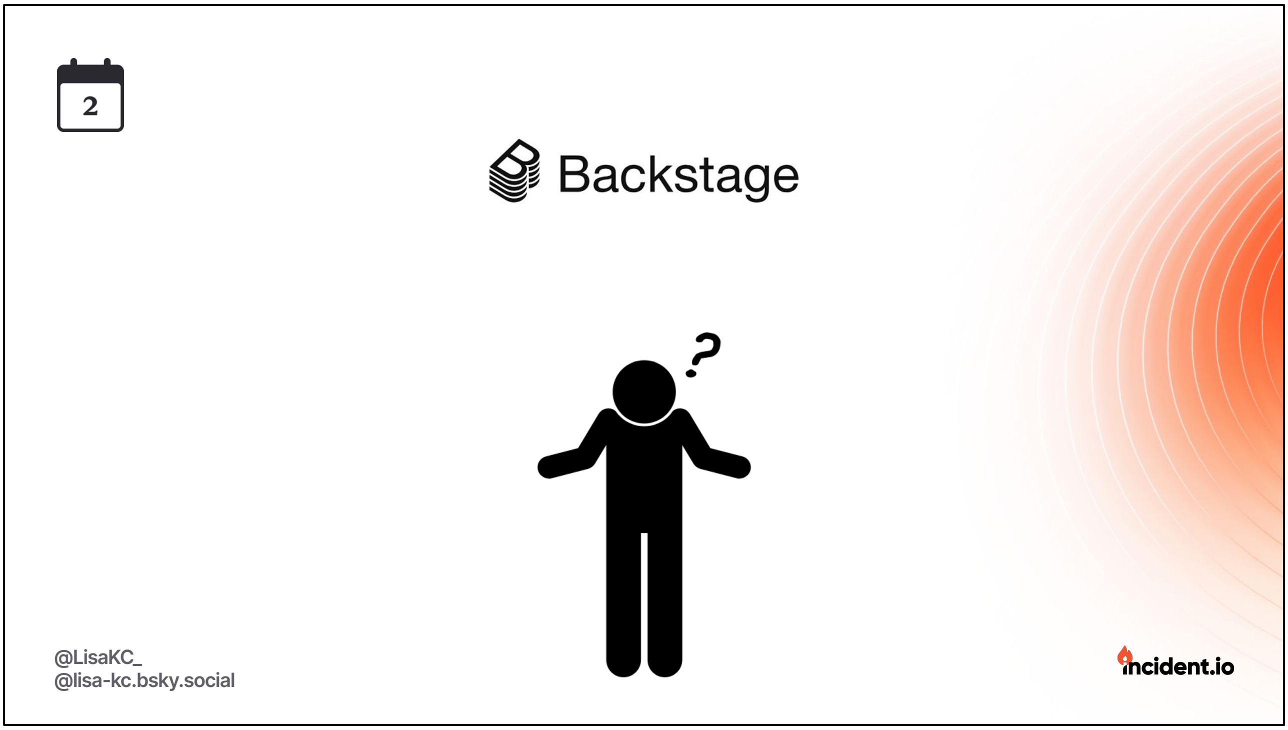
cortex

?

@LisaKC_
@lisa-kc.bsky.social

incident.io

You start with the classic build vs buy decision, writing out a list of use-cases and weighing up the pros and cons of various tools. You share a document for other engineers to read, but don't get much traction.



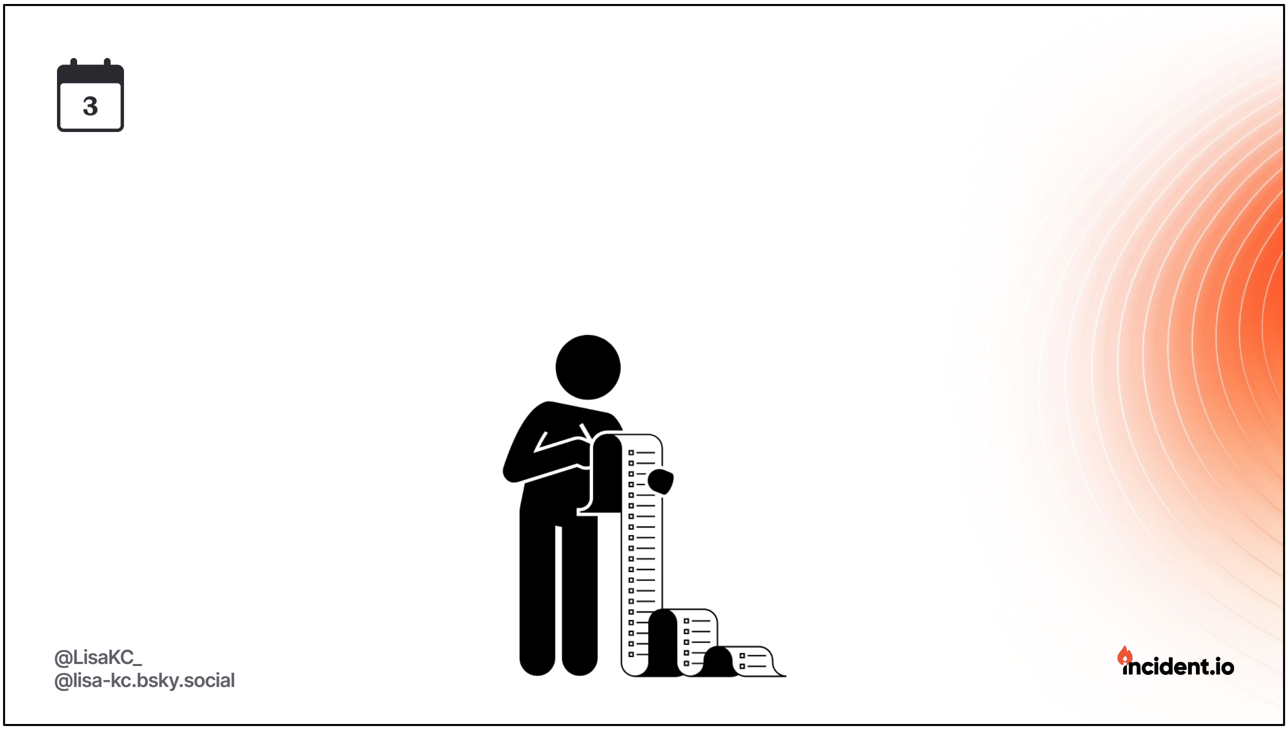
2

Backstage

@LisaKC_
@lisa-kc.bsky.social

incident.io

After a few opinions from a couple of key managers, and eventually decide on using a paid tool so you can get set up nice and fast. We can’t talk forever, let’s get building.



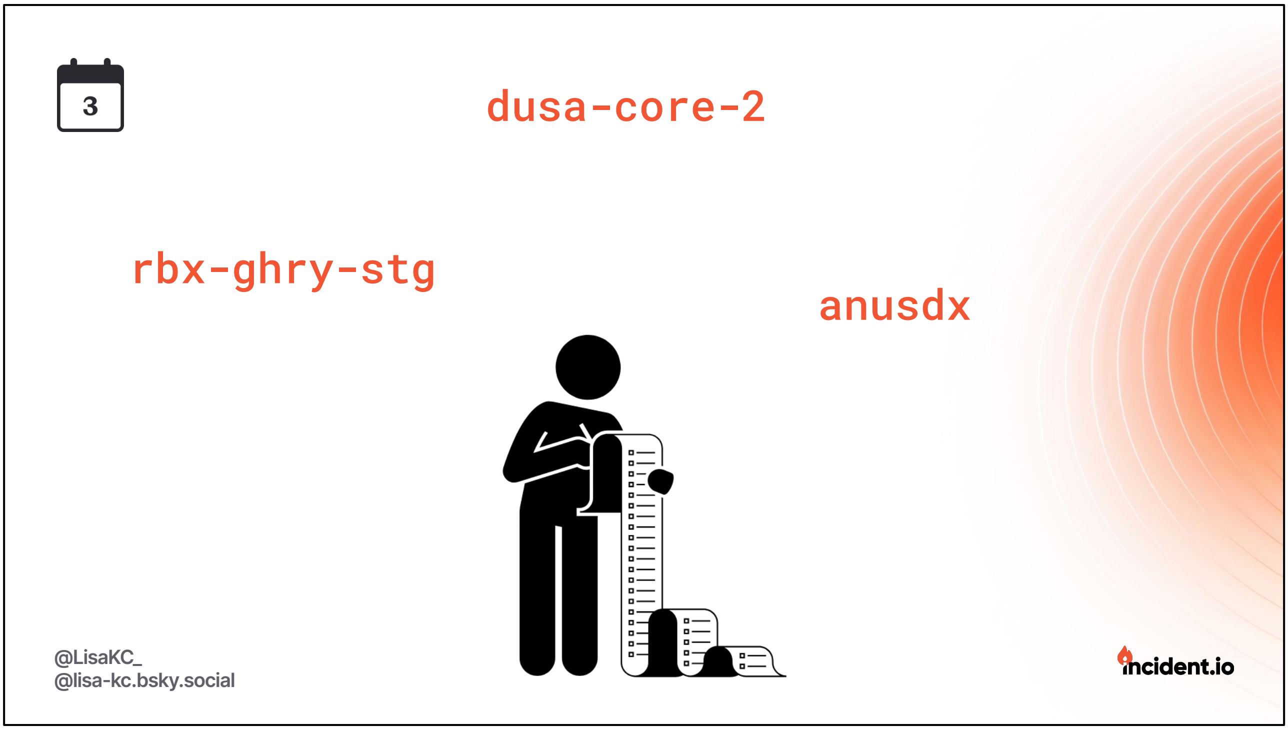
3

@LisaKC_
@lisa-kc.bsky.social

incident.io

So you open your laptop on day one and go ‘hey, do we have a list of services’? And you go and talk to some tech leads from other teams and pretty quickly conclude that no, we do not have a list of services.

And you start making a list. You write a list of all the services your team owns - well, the ones you can remember. You realise that you can pull a list of pods from your various kubernetes clusters, and work backwards from there. The names of the pods are …



3

dusa-core-2

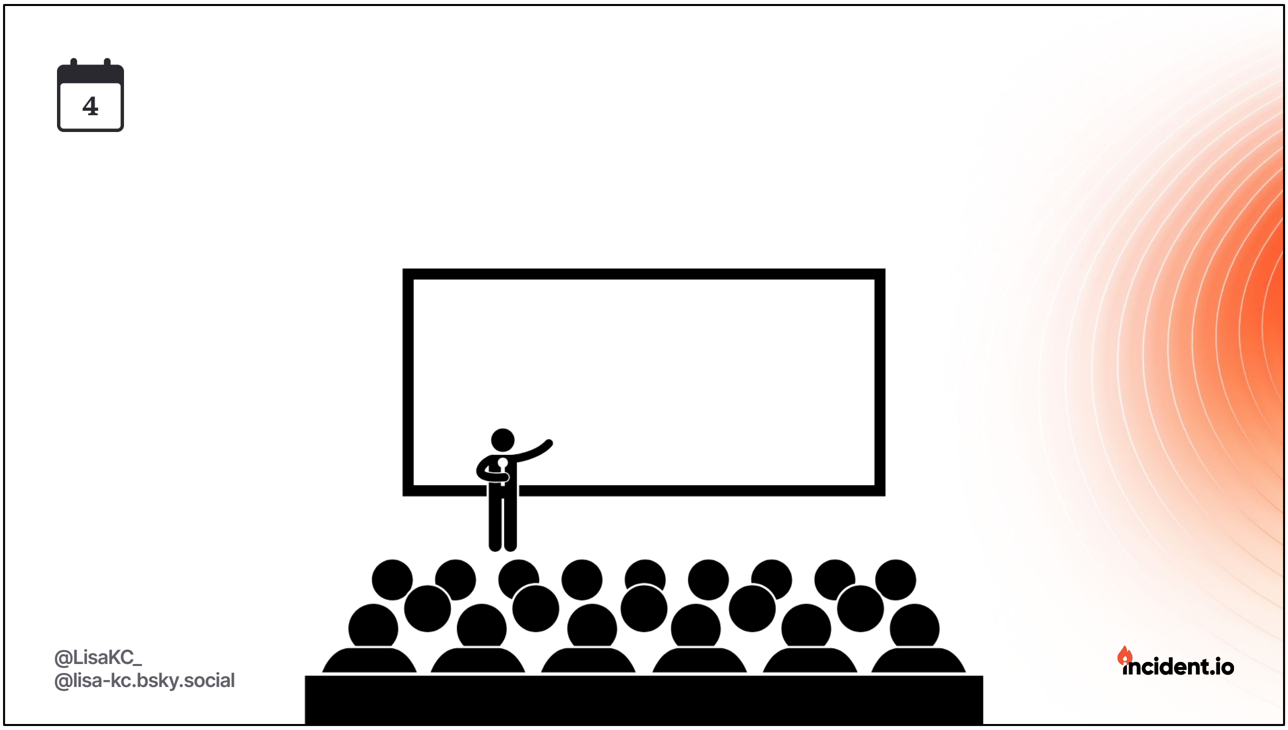
rbx-ghry-stg

anusdx

@LisaKC_
@lisa-kc.bsky.social

incident.io

Not as helpful as you’d hoped. Ultimately, you realise that you’re undertaking a huge information gathering exercise from across the organisation.

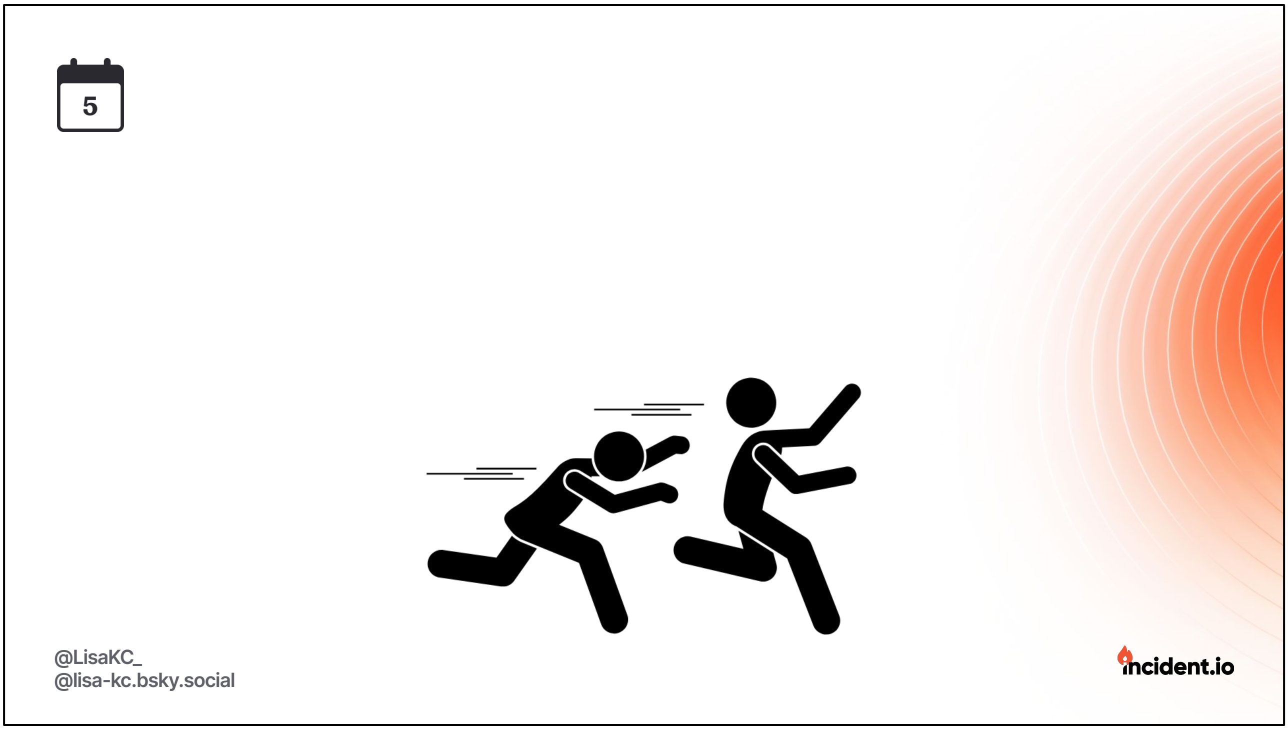


4

@LisaKC_
@lisa-kc.bsky.social

incident.io

You go to your monthly tech lead meeting and present the plan for building out your service catalog. You ask each of them to take a look at the list of kubernetes pods, map them to a ‘service’ in a big spreadsheet, answer a couple of questions about them, and then add anything that might be missing. Sounds simple

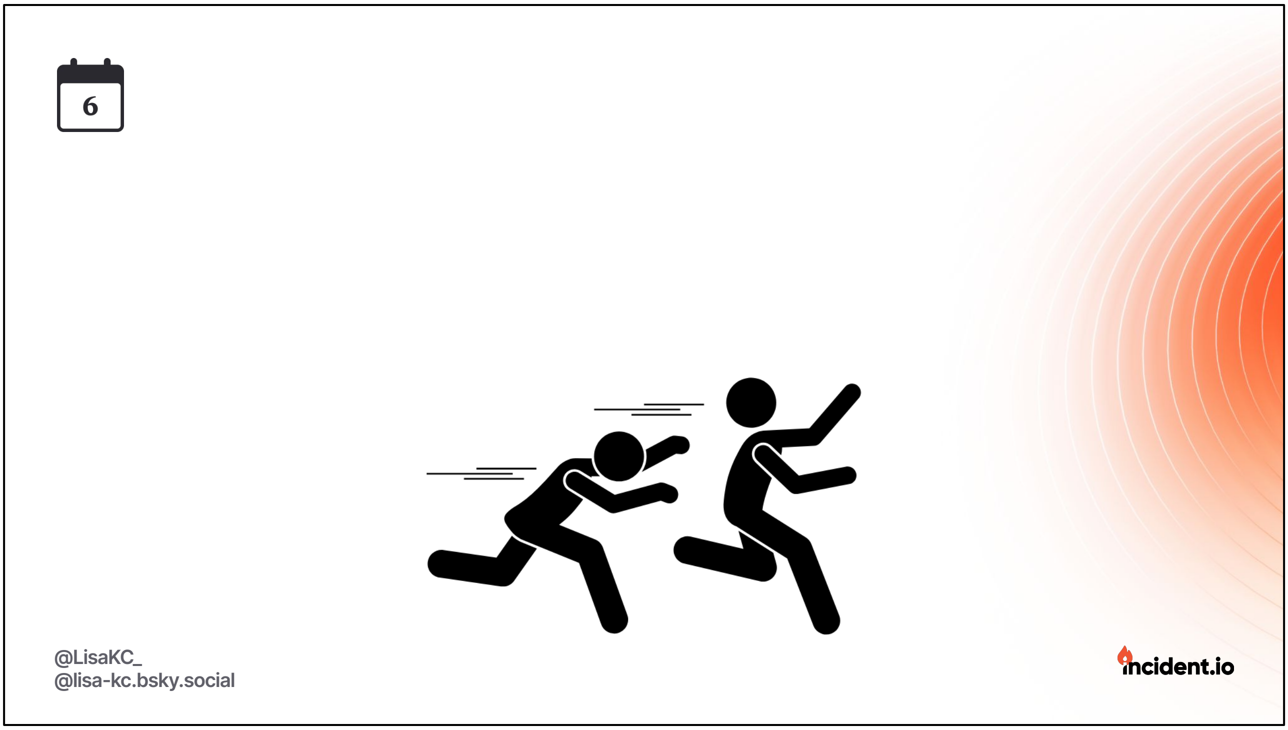


5

@LisaKC_
@lisa-kc.bsky.social

incident.io

You’ve had responses from some of your tech leads. But you’ve also discovered that this just isn’t a priority for many of them. They’ve got incidents to handle, shipping deadlines to meet, angry customers breathing down their neck. This isn’t something that they’re going to be assessed on, or held accountable for. They aren’t crying out for a Catalog - many aren’t really sure what value it will provide when it’s done. And you don’t have the authority, or the social capital, to change their mind.



6

@LisaKC_
@lisa-kc.bsky.social

incident.io

You end up spending your days chasing the busiest people in the engineering org, and burning all your social capital in the process.

7

@LisaKC_
@lisa-kc.bsky.social

incident.io

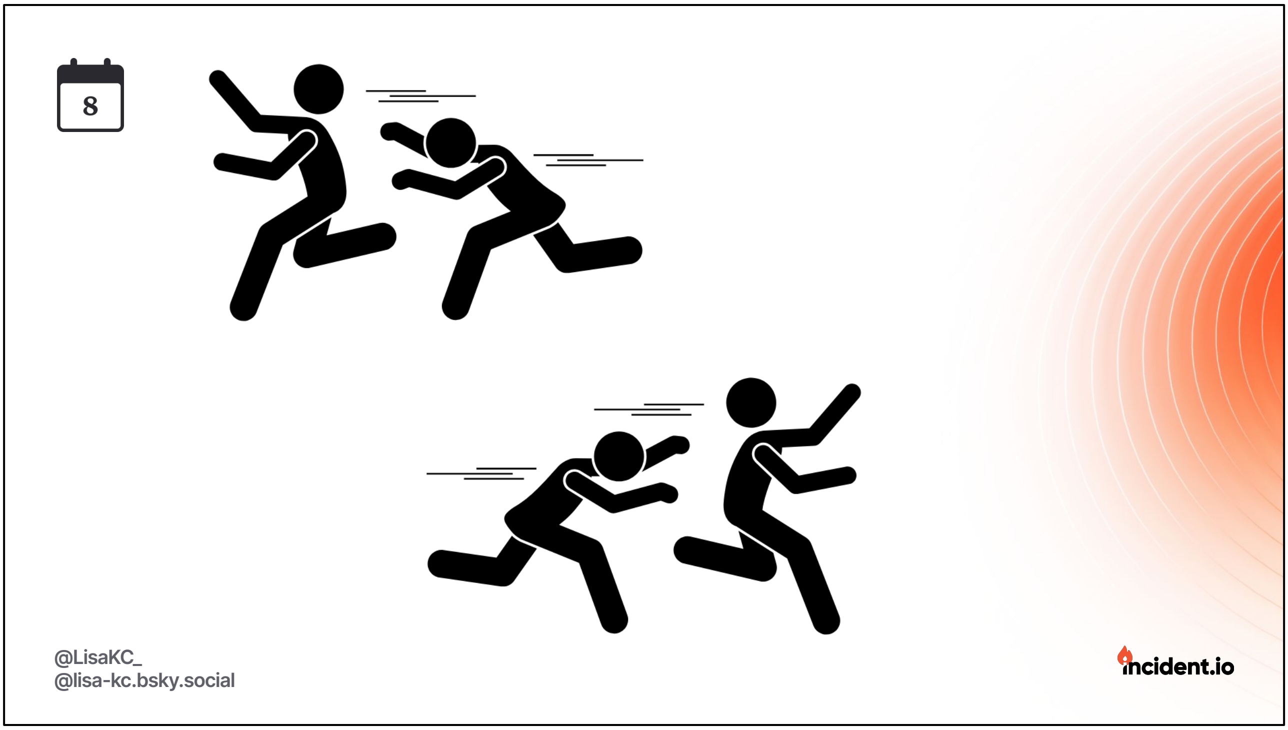
As you are chasing the rest of the org, people begin to meaningfully engage with what you're building and why - often for the first time. Turns out, it's not until you ask people to actually **do** something that they start really wanting to understand why they're doing it.

By now, you are spending most of your energy persuading people that this is a worthwhile thing for them to contribute to, and that you've chosen the right tools, approach etc.

You also probably find that you made some bad assumptions in your initial scoping: now people are really engaging, you start to see what the most valuable information you could gather actually is.

You realise that to get buy-in, you need to change your approach: ask for some different information, maybe ask people to store it in a different way. Everyone loves a good yaml versus JSON argument after all.

You know this is for the best: you'll get a better outcome this way, but it doesn't feel great at the time.

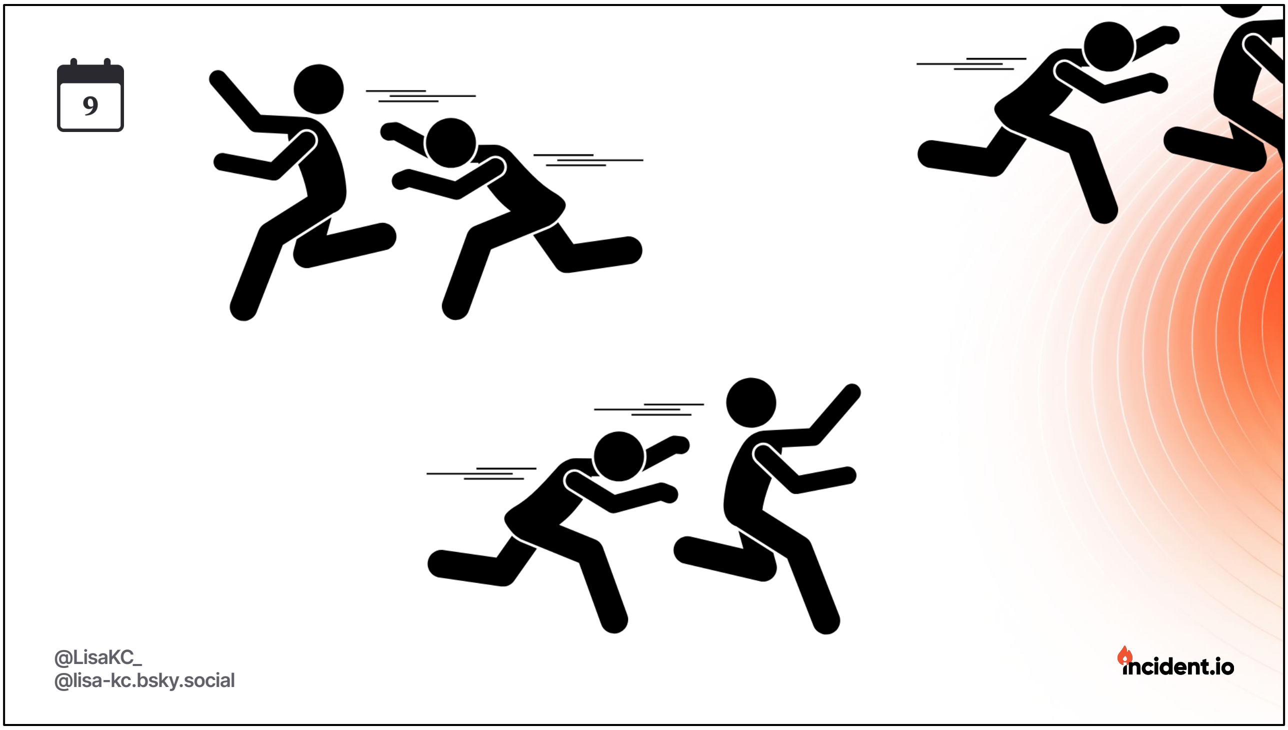


8

@LisaKC_
@lisa-kc.bsky.social

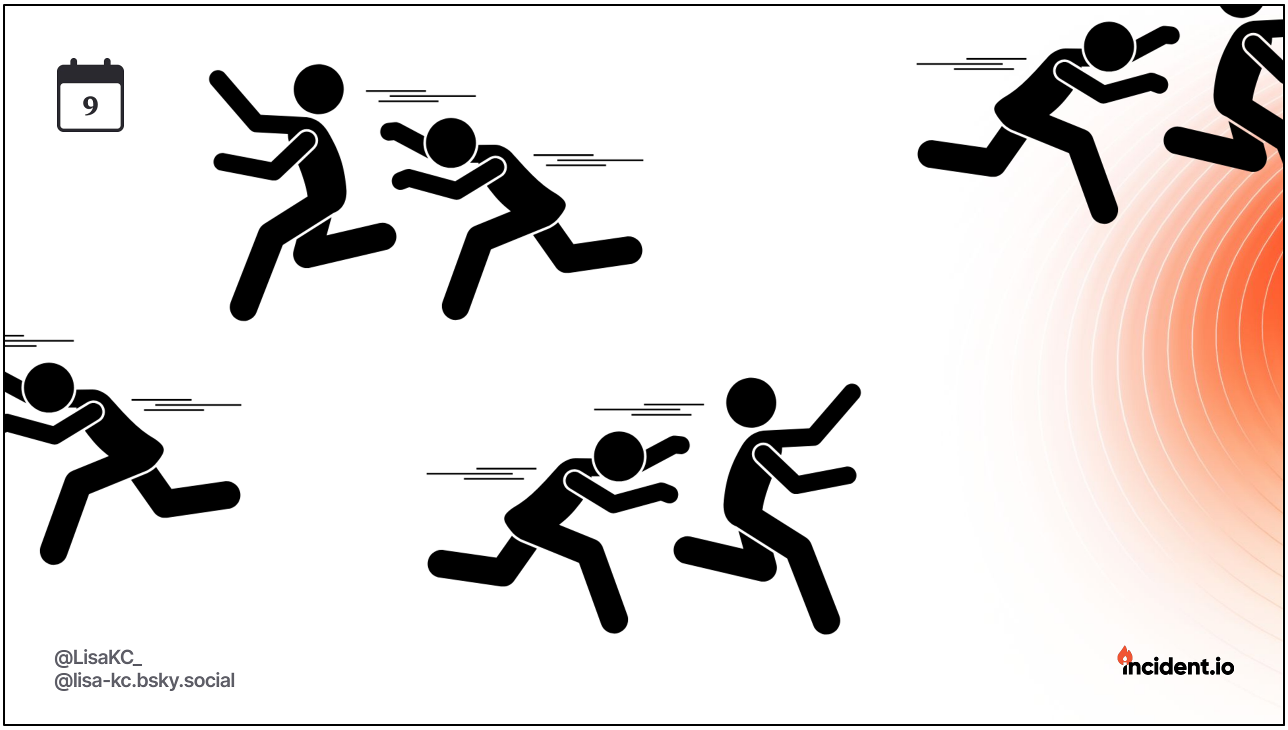
incident.io

You now find yourself going back to all those tech leads you'd spoken to and doing a second pass - asking for more information and then going back into the cycle of chasing.



@LisaKC_
@lisa-kc.bsky.social

Week 9, and … you’re still chasing …

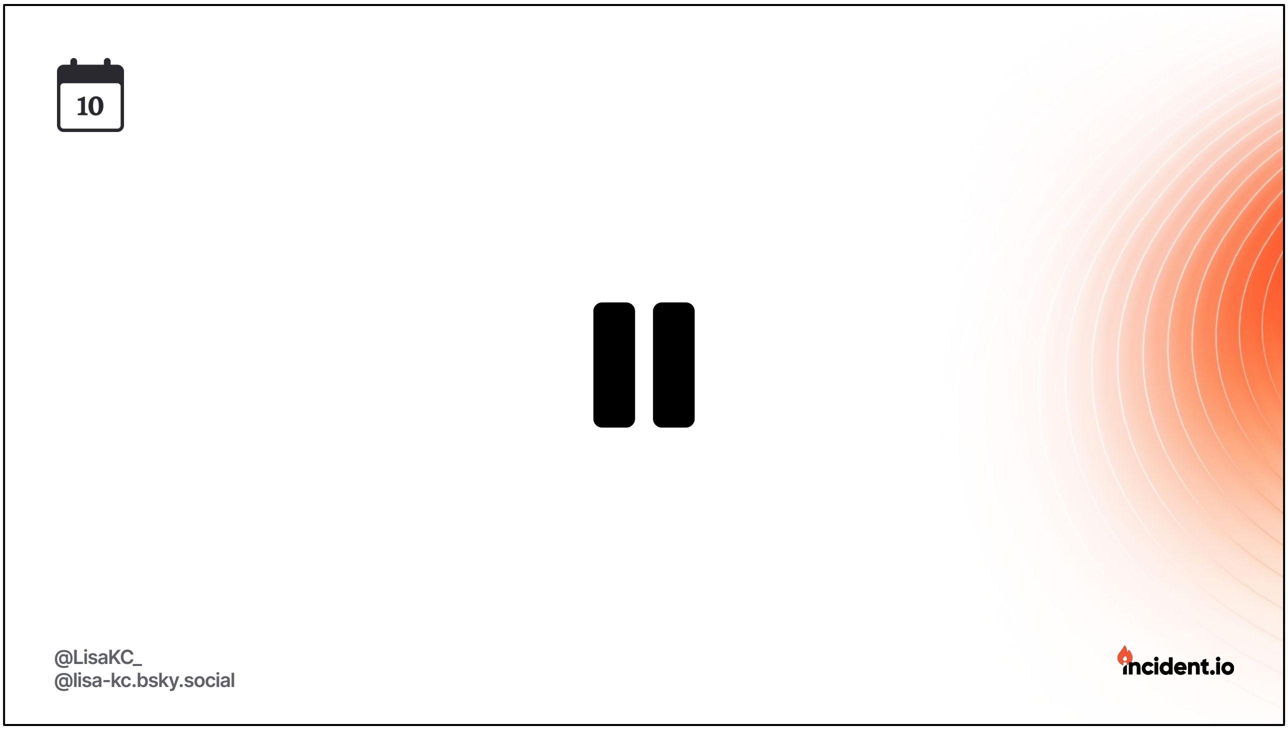


@LisaKC_
@lisa-kc.bsky.social

… and chasing

You start escalating to managers to get engagement from the last few holdouts. You do get what you need from them, but at a cost: their first experience with the Catalog is an actively hostile interaction, where they're being forced to spend time away from their top priorities.

At this point, you've used pretty much all your social capital that you had, and then some. You're in social capital debt.



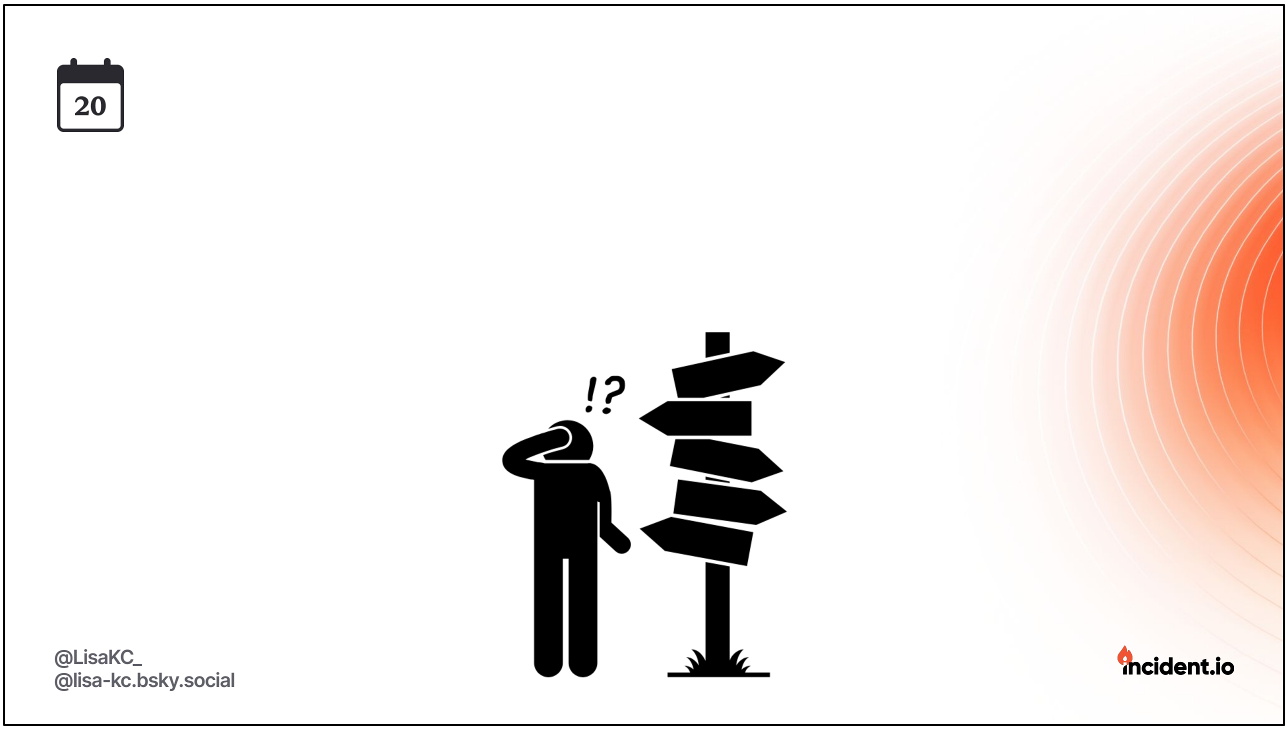
10

@LisaKC_
@lisa-kc.bsky.social

incident.io

You have a major security incident - it’s a near miss, luckily, but word comes down correctly that there’s a new P1 project that your team needs to pick up. We’ll have to pause the Catalog work - it can wait while we shore up out infrastructure.

See you in a couple of months.



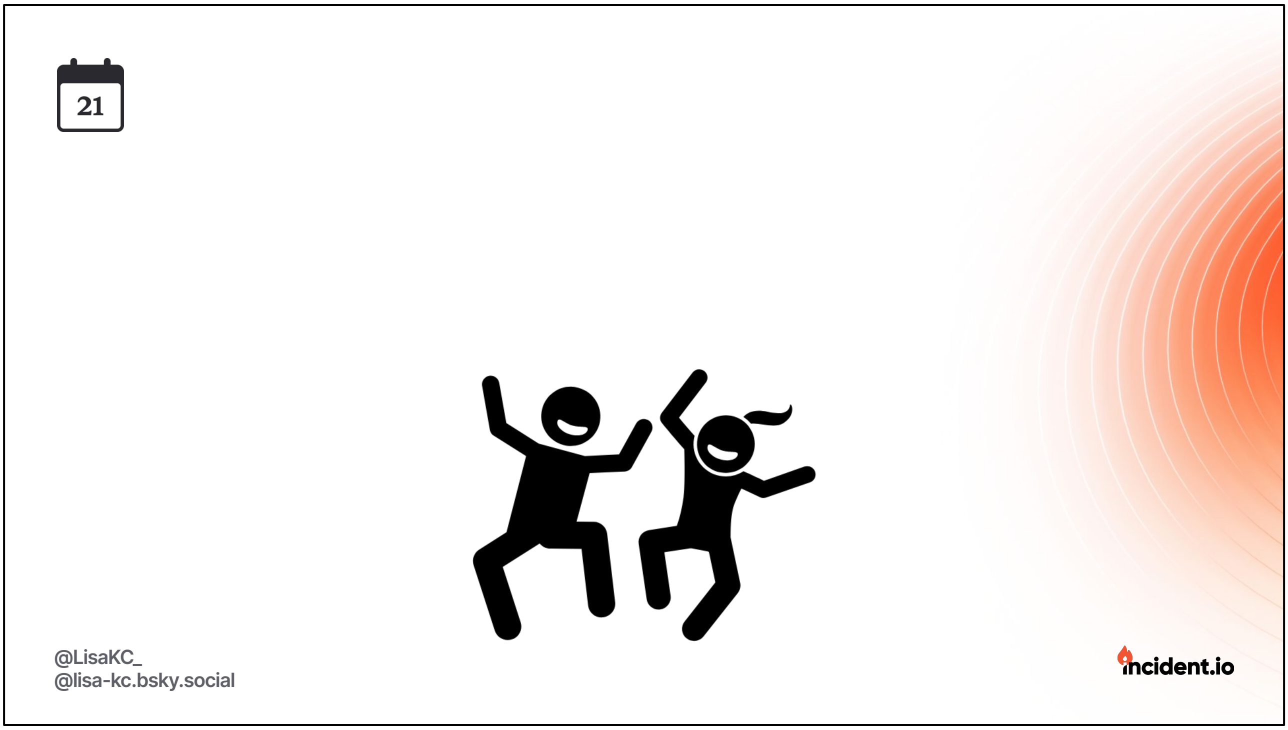
20

@LisaKC_
@lisa-kc.bsky.social

incident.io

And we’re back. You go to a company meeting where someone’s talking about the MVP of a new fraud detection service that they’ve just released. You realise that isn’t represented in your catalog at all - that team hasn’t kept their catalog up-to-date. How many other things have changed since you started gathering data?

You decide that it doesn’t matter. You need to launch this thing, so you can both start building features that use the catalog, and get out of this never ending death spiral of data-collection.



21

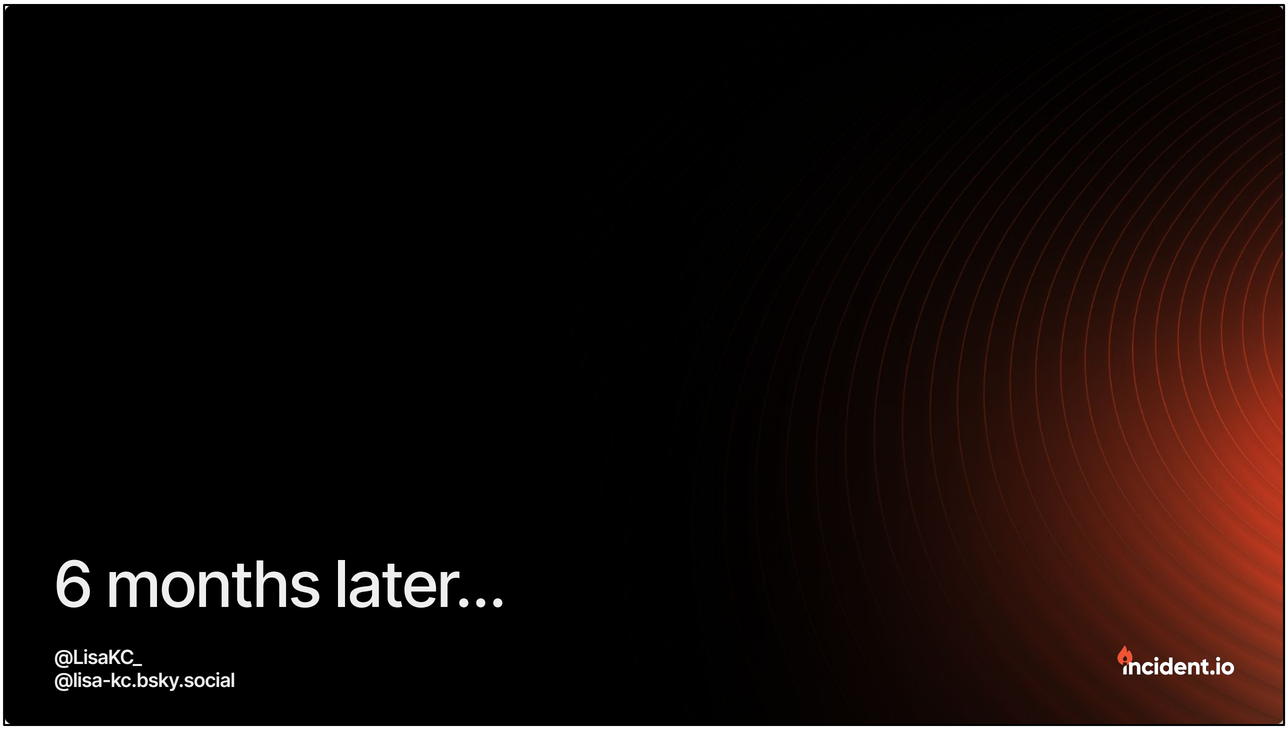
@LisaKC_
@lisa-kc.bsky.social

incident.io

You turn up one bright Monday morning and announce in Slack ‘hey, we’ve launched the catalog’. And you look around, and no one else really seems to care that much.

You run a demo with the whole of engineering, and everyone nods and says ‘ooh that’s neat’, and you tap yourselves on the back for a job well done.

But then, everyone goes back to their day-to-day as if nothing’s really happened.



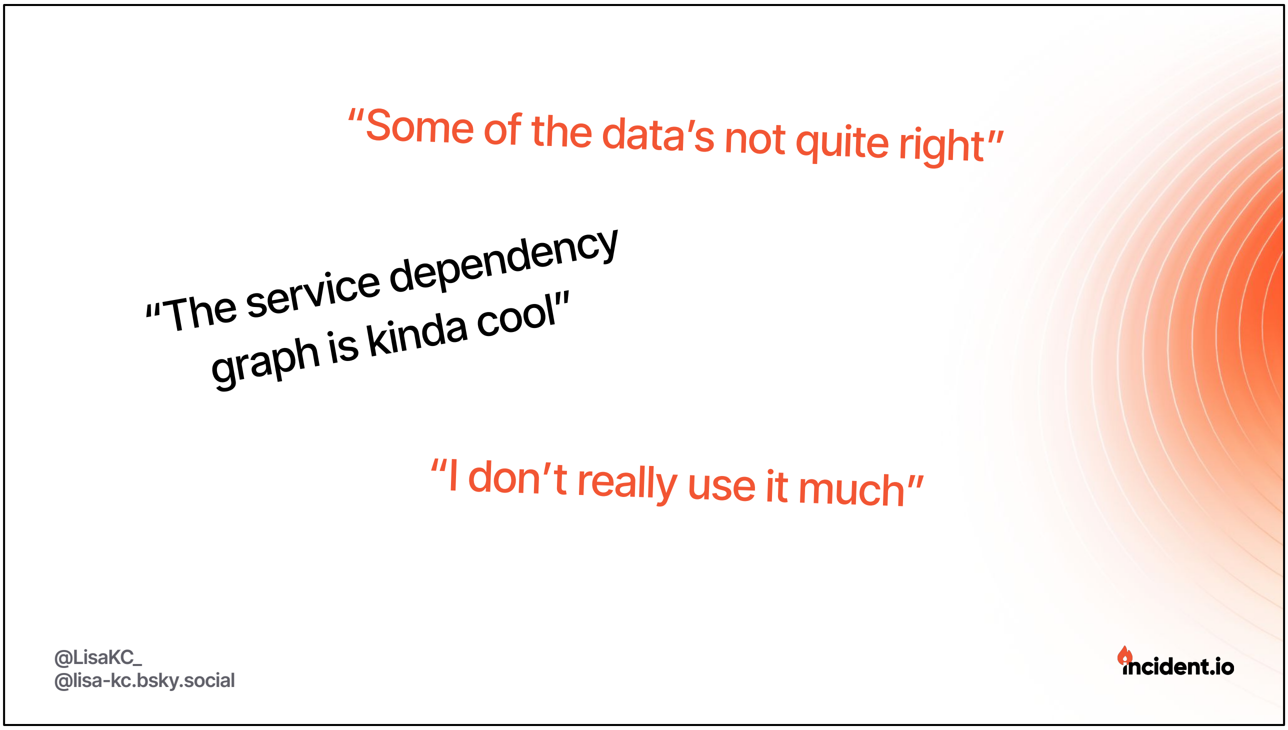
6 months later...

@LisaKC_
@lisa-kc.bsky.social

incident.io

Over the next 6 months, your team continues to try and find different things to build using the catalog. You run an internal survey 6 months later, asking engineering how often they use the catalog, and what they find most useful.

And the response is … underwhelming.



"Some of the data's not quite right"

"The service dependency graph is kinda cool"

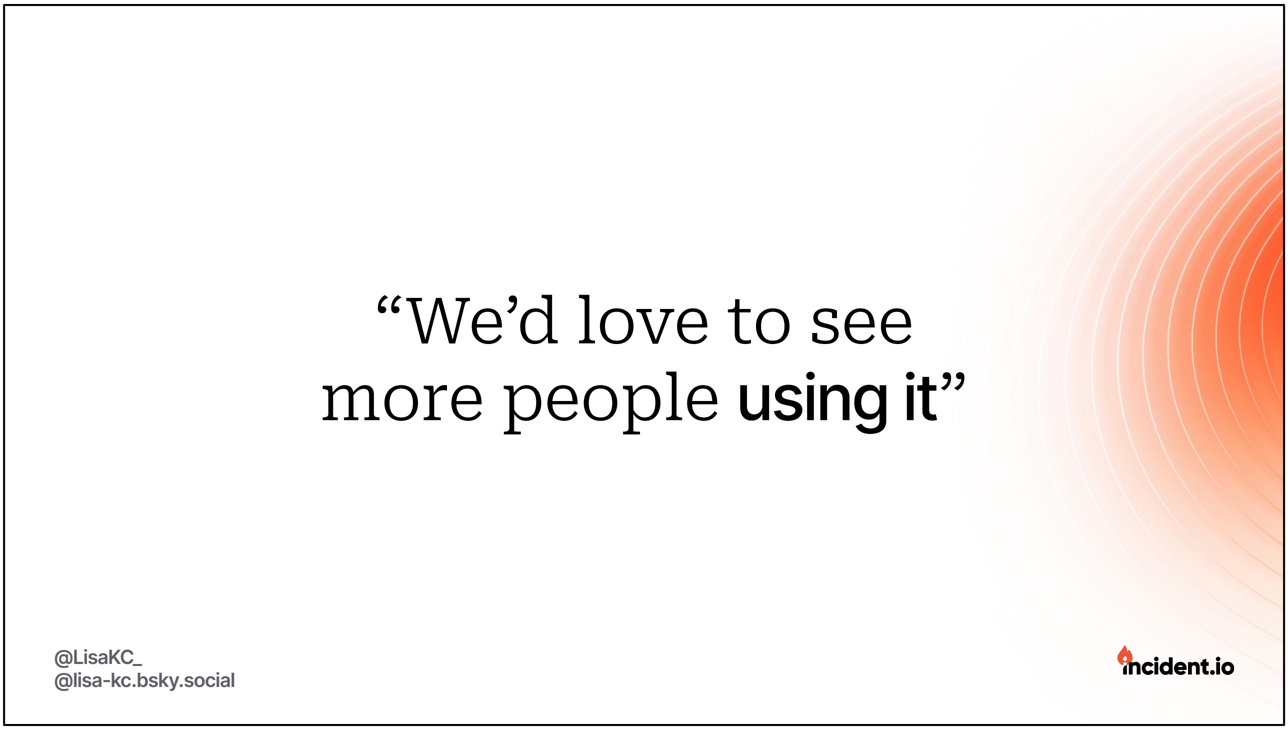
"I don't really use it much"

@LisaKC_
@lisa-kc.bsky.social

incident.io

- Some of the data's not quite right
- The service dependency graph is cool
- I don't really use it much to be honest

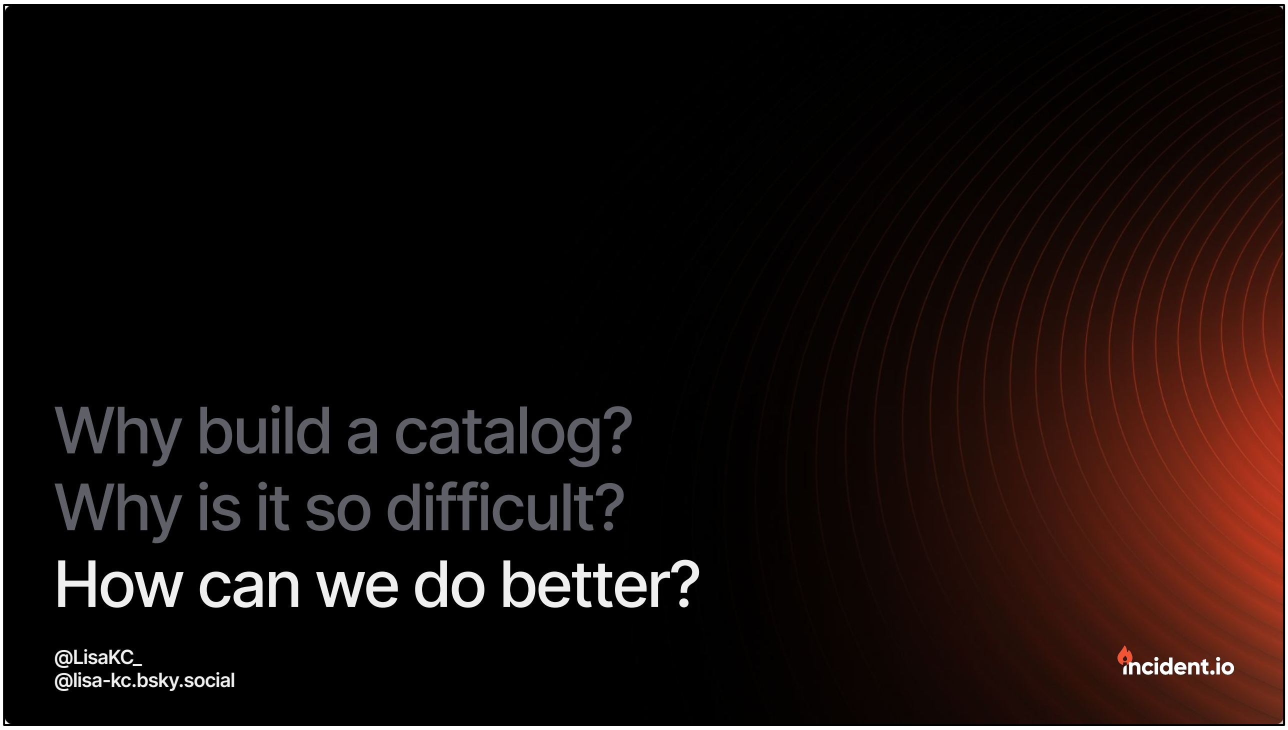
And when you reflect as a team, you conclude that



“We’d love to see more people **using it**”

@LisaKC_
@lisa-kc.bsky.social

We’d love to see more people using it



Why build a catalog?

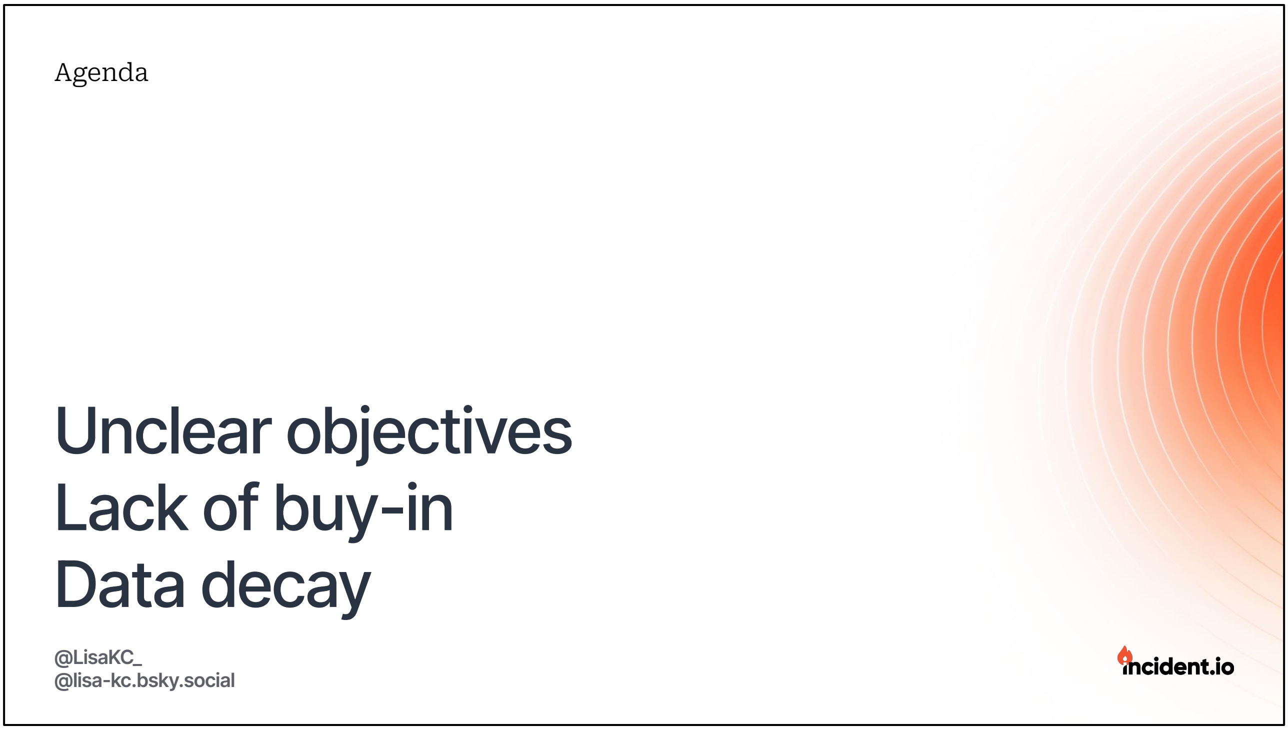
Why is it so difficult?

How can we do better?

@LisaKC_
@lisa-kc.bsky.social

incident.io

So, what went wrong?



Agenda

Unclear objectives
Lack of buy-in
Data decay

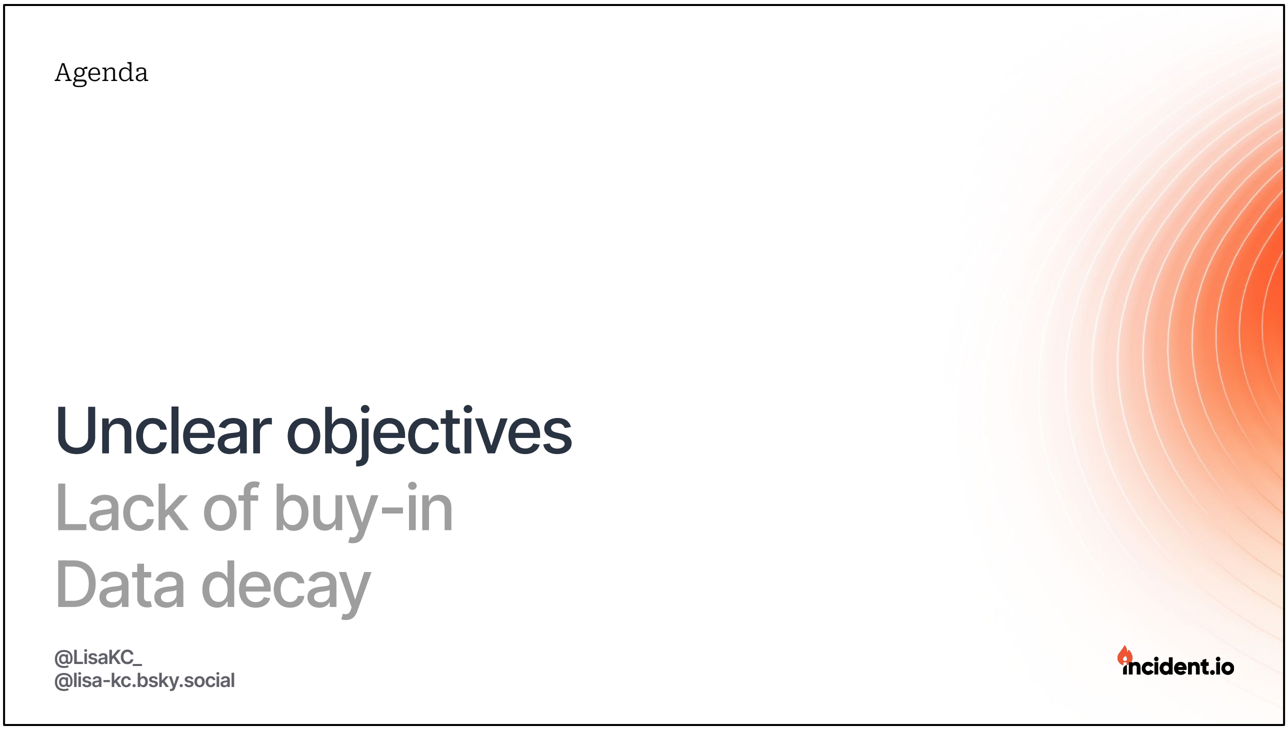
@LisaKC_
@lisa-kc.bsky.social

incident.io

We’ve ended up with three big problems:

1. We don’t really remember why we built it
2. We don’t have buy in from the org
3. The data isn’t staying up to date

Let’s break that down, and see what we can learn



Agenda

Unclear objectives

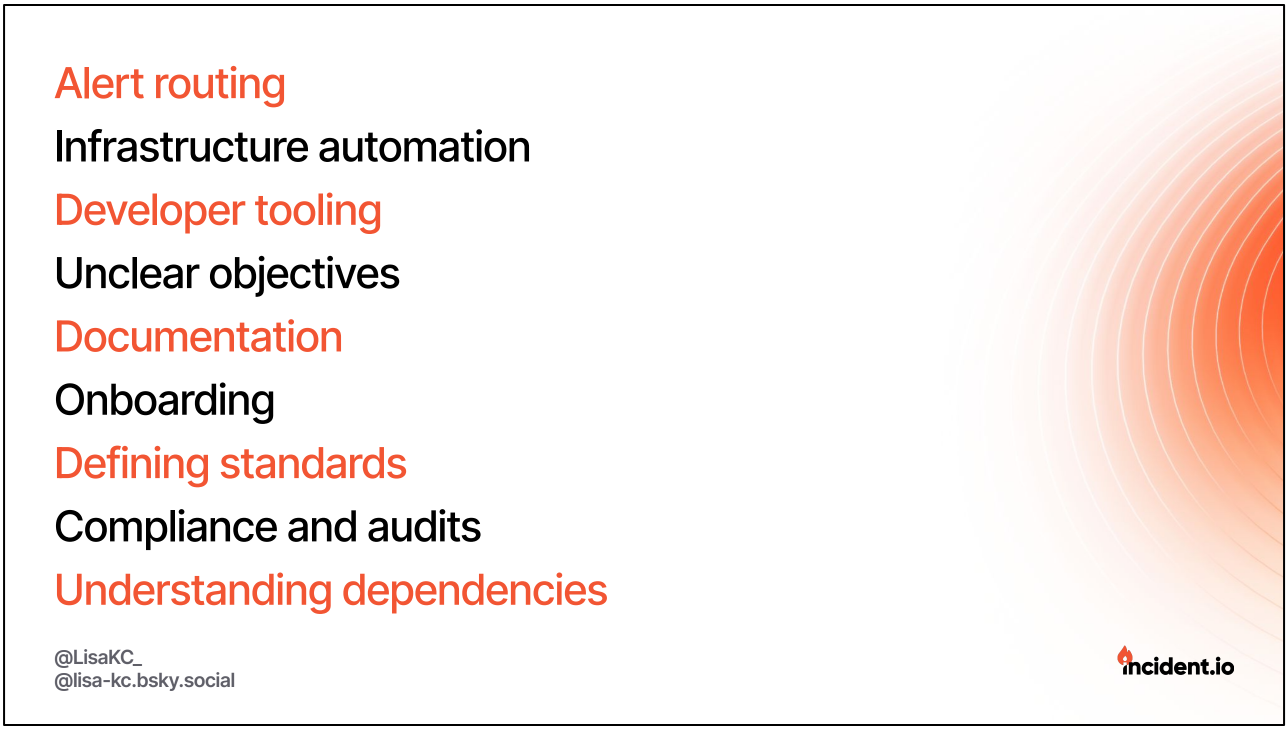
Lack of buy-in

Data decay

@LisaKC_
@lisa-kc.bsky.social

incident.io

We had unclear objectives at the start



Alert routing

Infrastructure automation

Developer tooling

Unclear objectives

Documentation

Onboarding

Defining standards

Compliance and audits

Understanding dependencies

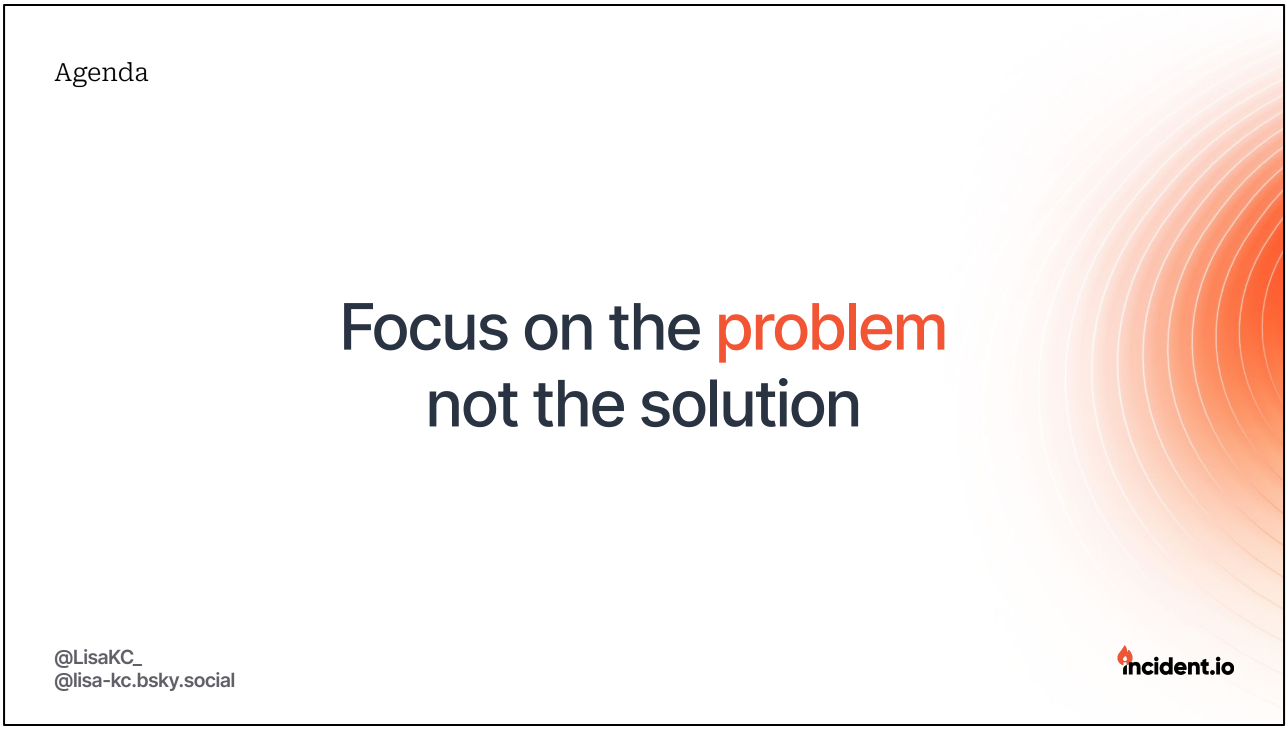
@LisaKC_
@lisa-kc.bsky.social

incident.io

Remember that laundry list of things that the catalog was going to do? Having so many different objectives makes it really difficult to deliver value on any of them. Your focus is constantly split, trying to make a catalog that can be all things to all people.

It’s an easy mistake to make. When you wrote your business case to build a catalog, perhaps no one of these problems would justify such a long piece of work. But together, it becomes a no brainer.

When we start building a catalog, we seem to forget everything we know about building products.



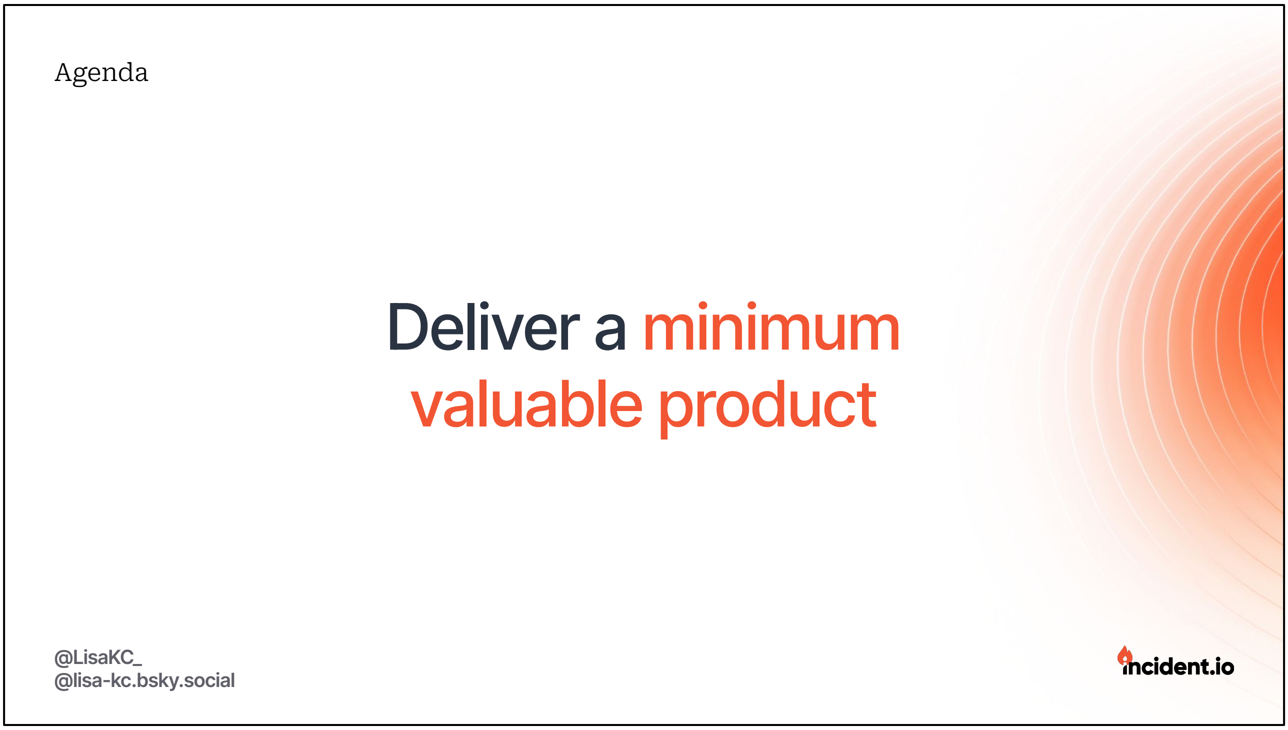
Agenda

Focus on the problem not the solution

@LisaKC_
@lisa-kc.bsky.social

incident.io

To build a good product, you have to be focussed on the problem not the solution. But with a catalog, the problem is so multi-faceted, that instead all our focus goes on the solution: how are we going to get all this data that we don’t currently have? What tool are we going to put it in?



Agenda

Deliver a minimum valuable product

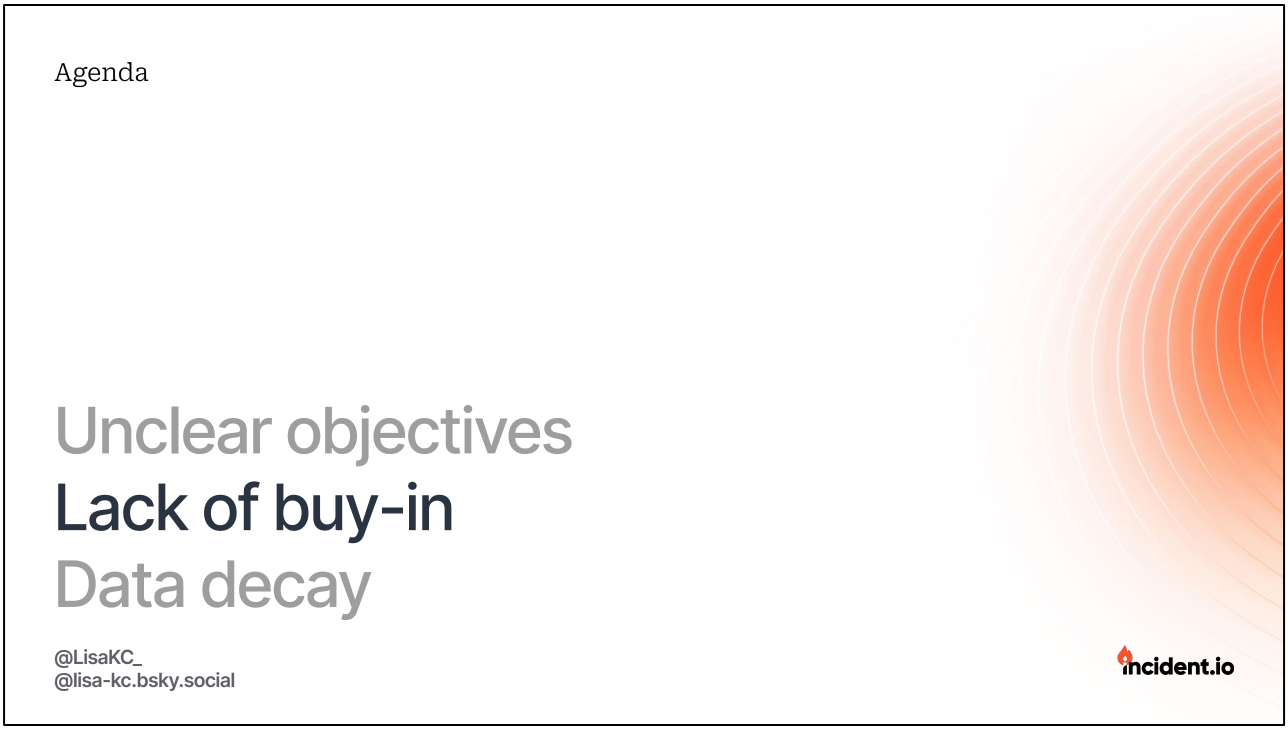
@LisaKC_
@lisa-kc.bsky.social

incident.io

And, just like building any other product, you want to deliver value incrementally: focussing on the first minimum valuable product. What's the first thing that would add value? How fast can I get there? And then we do that again, and again, and again.

And maybe the first thing you build might not be a catalog at all. Maybe it's just reorganising your `infrastructure` repo so that all the configuration relating to a service can live in a single file. Or creating a Notion database where you can put all your runbooks.

Those projects aren't failures, they're useful steps towards building out a catalog that's valuable for your team - looking for the next valuable increment.



Agenda

Unclear objectives

Lack of buy-in

Data decay

@LisaKC_
@lisa-kc.bsky.social

incident.io

Onto problem 2: we just didn't have enough buy-in from the org

@LisaKC_
@lisa-kc.bsky.social

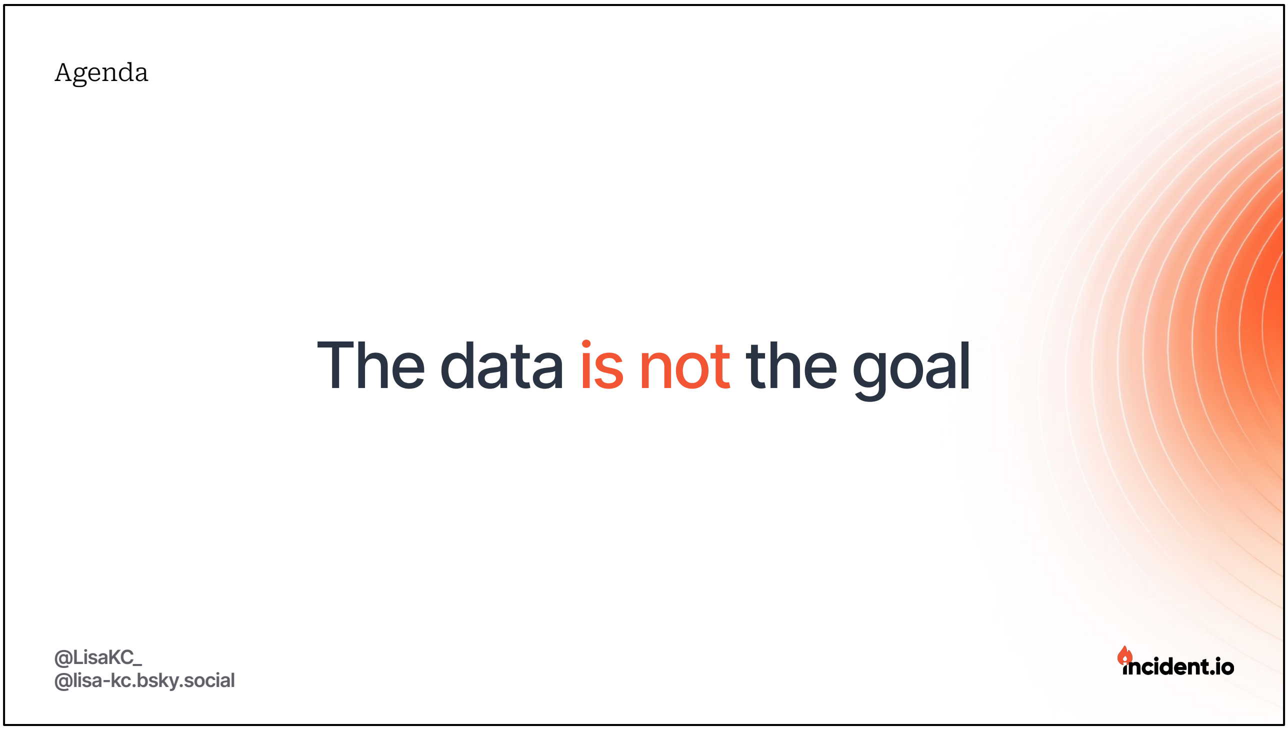
incident.io

You'll remember in our story, we spent a lot of time chasing people from around the organisation. That process used a lot of social capital, and left key people around the org feeling bored, grumpy and confused.

We were asking people to take time out of their busy schedules to do something fundamentally boring - write lists of things that they own, and fill in forms about them.

And remember, we didn't deliver any value back to those people for nearly 5 months.

Once again, focussing on incremental value will help us here. It's also much easier to get buy-in for smaller pieces of work with clear goals, and when you're delivering value back to people quickly and iteratively.



Agenda

The data is not the goal

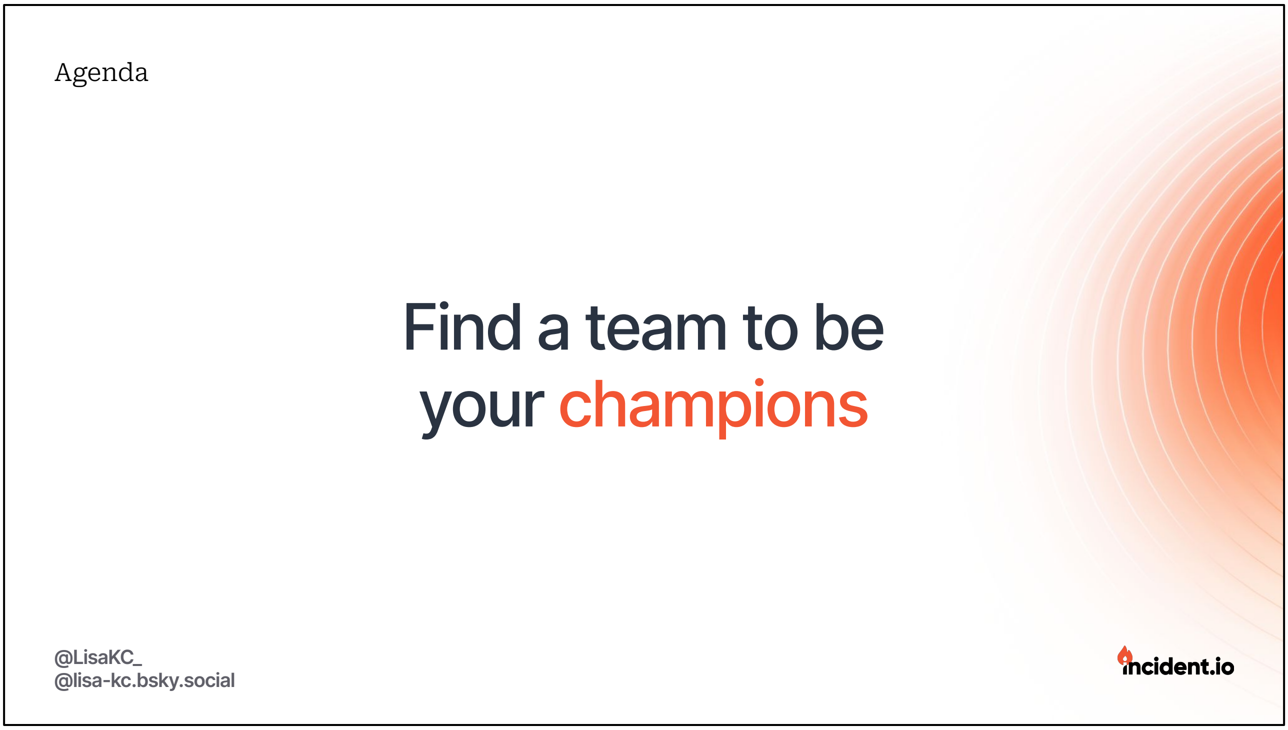
@LisaKC_
@lisa-kc.bsky.social

incident.io

The other trap that we fell into was to prioritise **breadth** over **depth**. When you're making a list … and checking it twice … it's really easy to become fixated on finishing the list. Finishing the list is really satisfying, and it feels great to have everything written down.

But we risk falling foul of the 80-20 rule.

Gathering that long tail of data is **incredibly** time consuming, uses all of our social capital and often not all that valuable.



Agenda

Find a team to be your champions

@LisaKC_
@lisa-kc.bsky.social

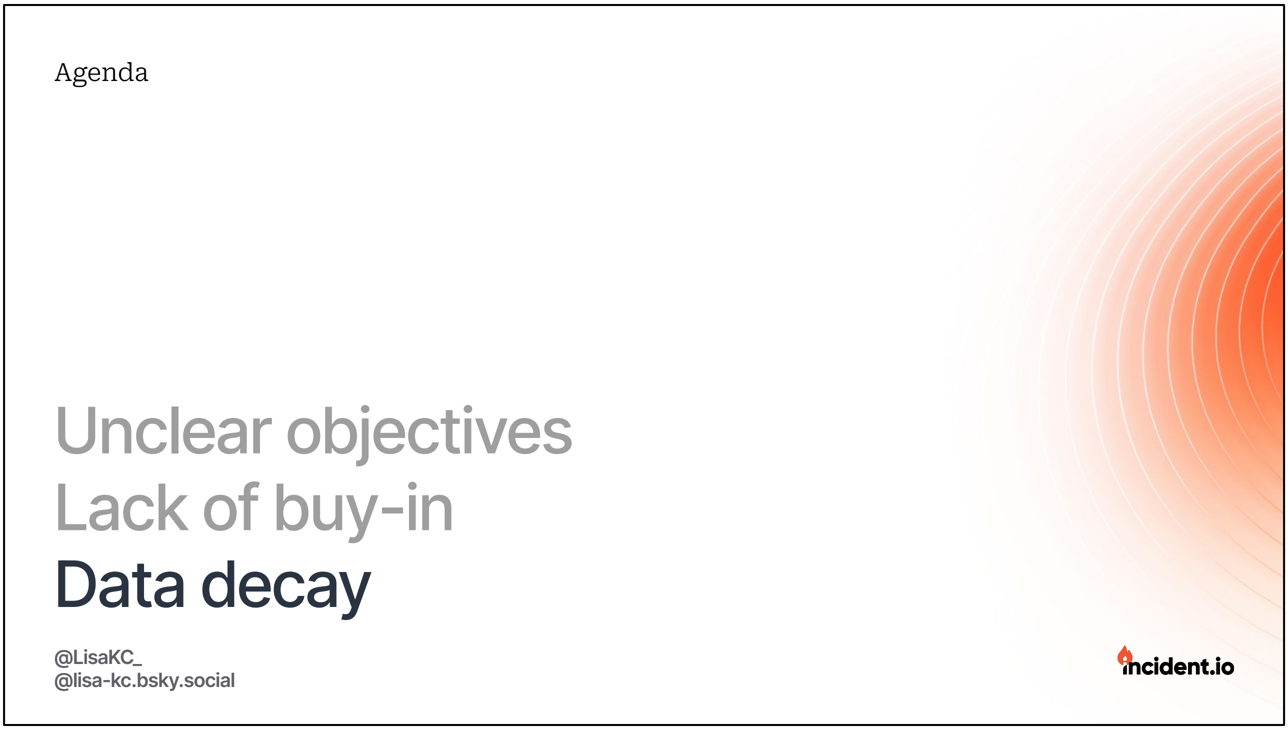
incident.io

Instead of trying to provide a thin layer of value to the whole org, we can instead focus on a single team first. That again makes our first project much more achievable: we don't have to do something for all 300 services that we run or 3000, we can just make it feel really good for the 15 services that this one team has ownership of.

We can choose a team that has a bit more space in their roadmap - maybe a team that we've worked with before and already have a pre-built relationship.

We can iterate quickly on their feedback, making them feel part of the process and delivering incremental value as we go.

Once we've got something that feels good, it's easier to get buy-in from the rest of the org. You can show them exactly what they'll get if they take the time to gather the data. **And** you can continue to deliver value despite the handful of teams that don't want to engage right now.



Agenda

Unclear objectives

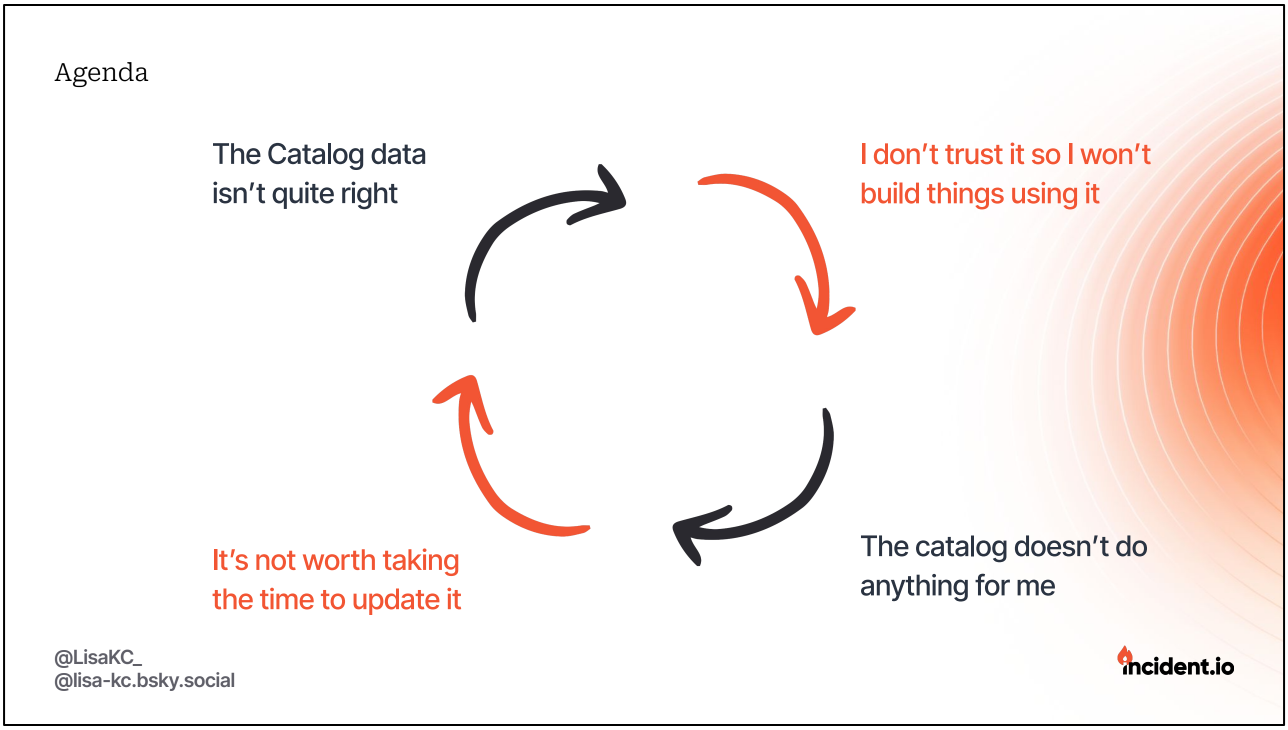
Lack of buy-in

Data decay

@LisaKC_
@lisa-kc.bsky.social

incident.io

And so we come to our final problem. The data is rotting



Agenda

The Catalog data isn’t quite right

I don’t trust it so I won’t build things using it

The catalog doesn’t do anything for me

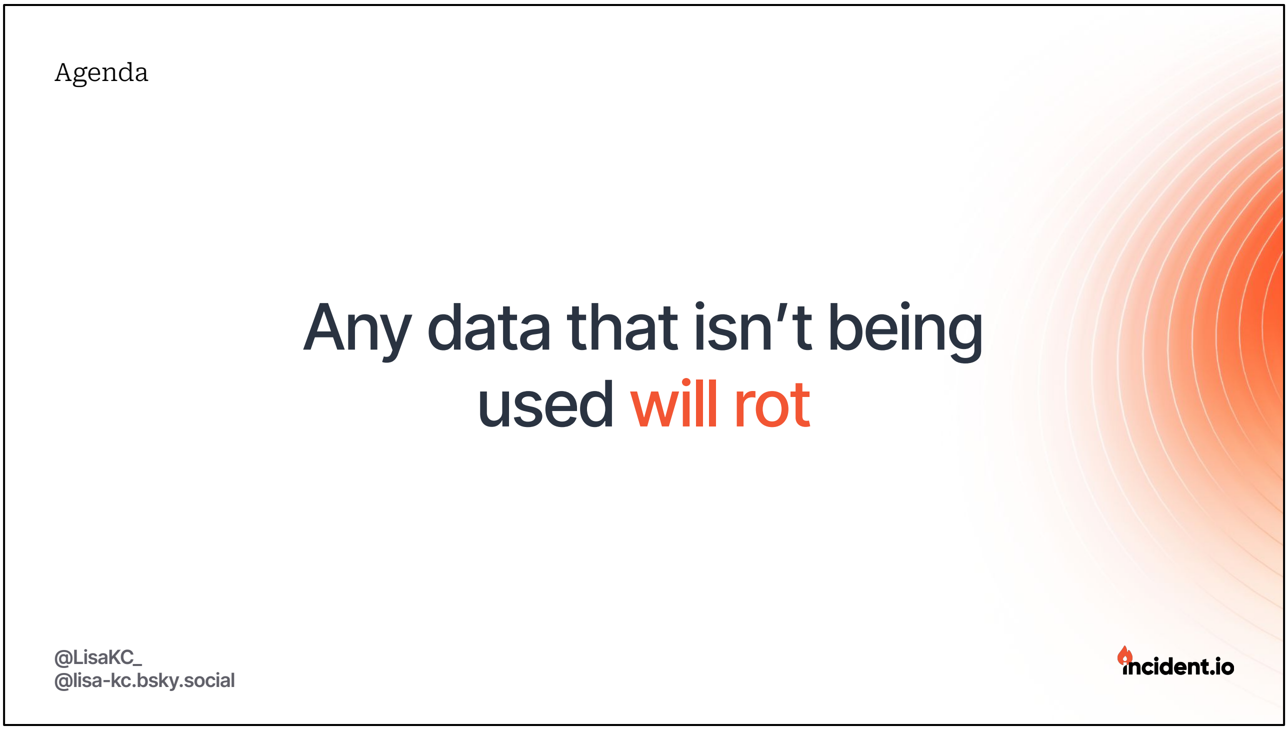
It’s not worth taking the time to update it

@LisaKC_
@lisa-kc.bsky.social

incident.io

We’ve found ourselves trapped into a negative cycle

- The catalog data isn’t quite right
- I don’t trust it so I’ll build on it
- The catalog doesn’t do anything for me
- It’s not worth updating the catalog



Agenda

Any data that isn't being used will rot

@LisaKC_
@lisa-kc.bsky.social

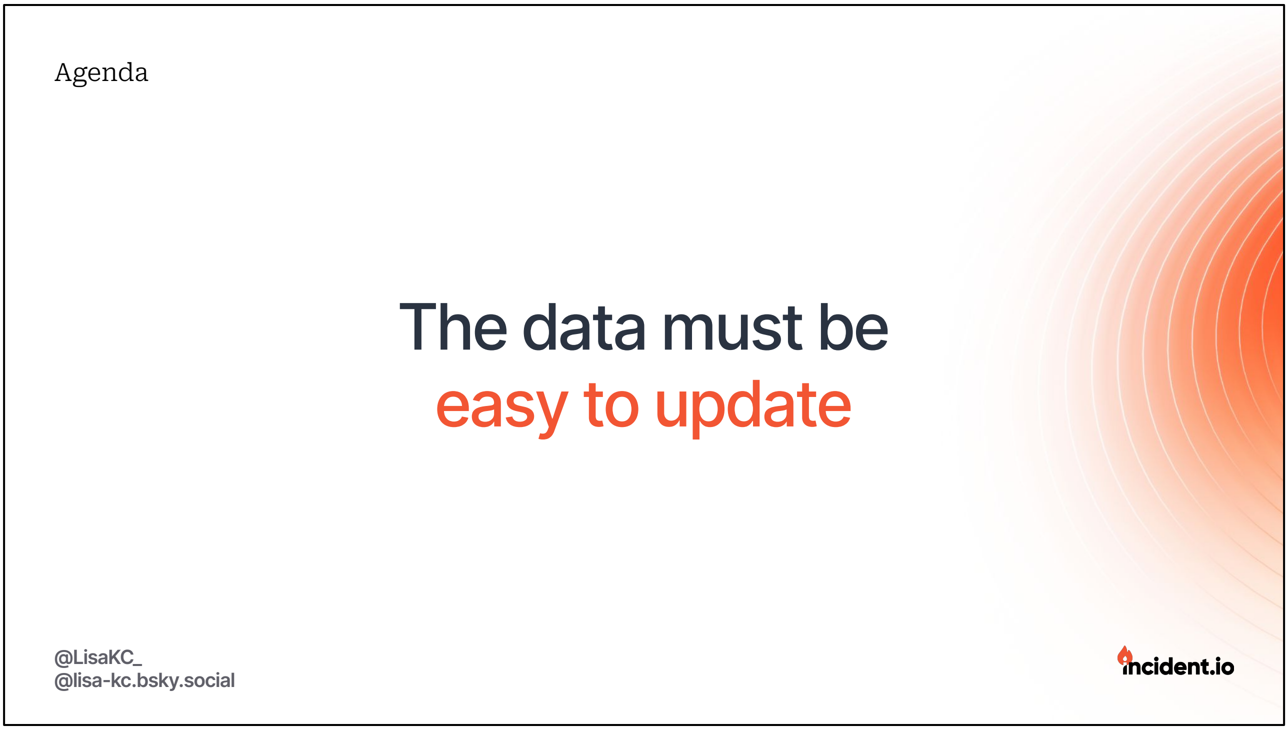
When building a catalog, you must always remember that **any data that isn't being used will rot**. Engineering organisations change quickly, meaning that the data becomes stale. And once it's stale, it's not just unhelpful but can be actively harmful.

If the data isn't accurate, our negative cycle is almost inevitable: once people lose trust in the data, it's hard to win it back.

The only way to keep data from going stale is to **use it**. And no, displaying data on a web page does not count as using it.

If your catalog contains the data about which services are deploying in which clusters, it should **decide** which services are deployed in which clusters - so it can't rot.

If it stores a list of which Teams own which Services, it should decide who gets paged when there's an alert against that service. If the wrong person gets paged, I promise they'll fix the catalog data very quickly.



Agenda

The data must be easy to update

@LisaKC_
@lisa-kc.bsky.social

incident.io

Well, as long as they *can* change the data.

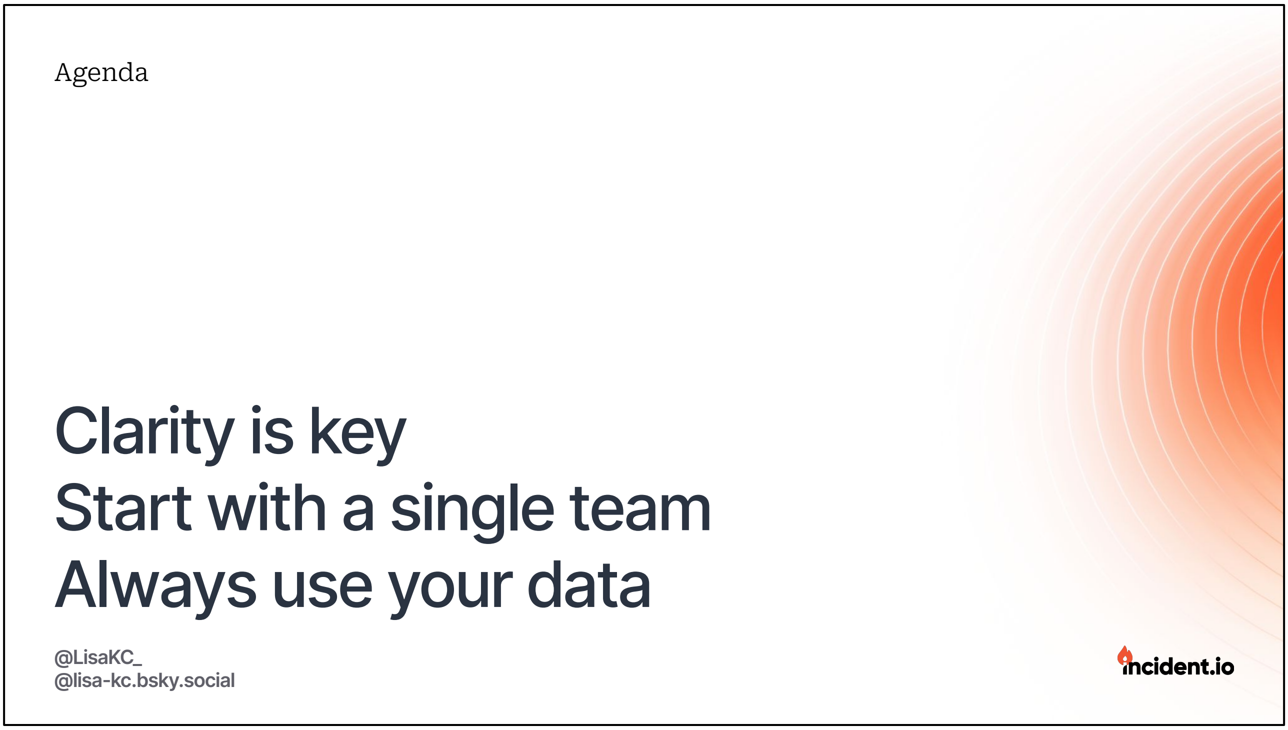
If someone can’t change the data easily or safely, then the data will inevitably rot, and fast.

The last thing you want is to become the gatekeepers of the underlying data, receiving endless requests to make small changes, which will eventually dry up as people just … stop bothering.

That means, everyone needs to know where the data is, and have the confidence that they change it.

A classic example of this going wrong is having an overloaded IT or PeopleOps team who are responsible for managing team membership, leaving the ‘on paper’ org chart to drift further and further from reality.

When you get this wrong, and the data becomes stale, engineers, we’re a good bunch of problem solvers, and we’ll find another ways to solve our problem (like adding a hacky redirect into your paging software to redirect alerts to the right team. Yes I’ve seen it happen, and yes it was a bad idea).



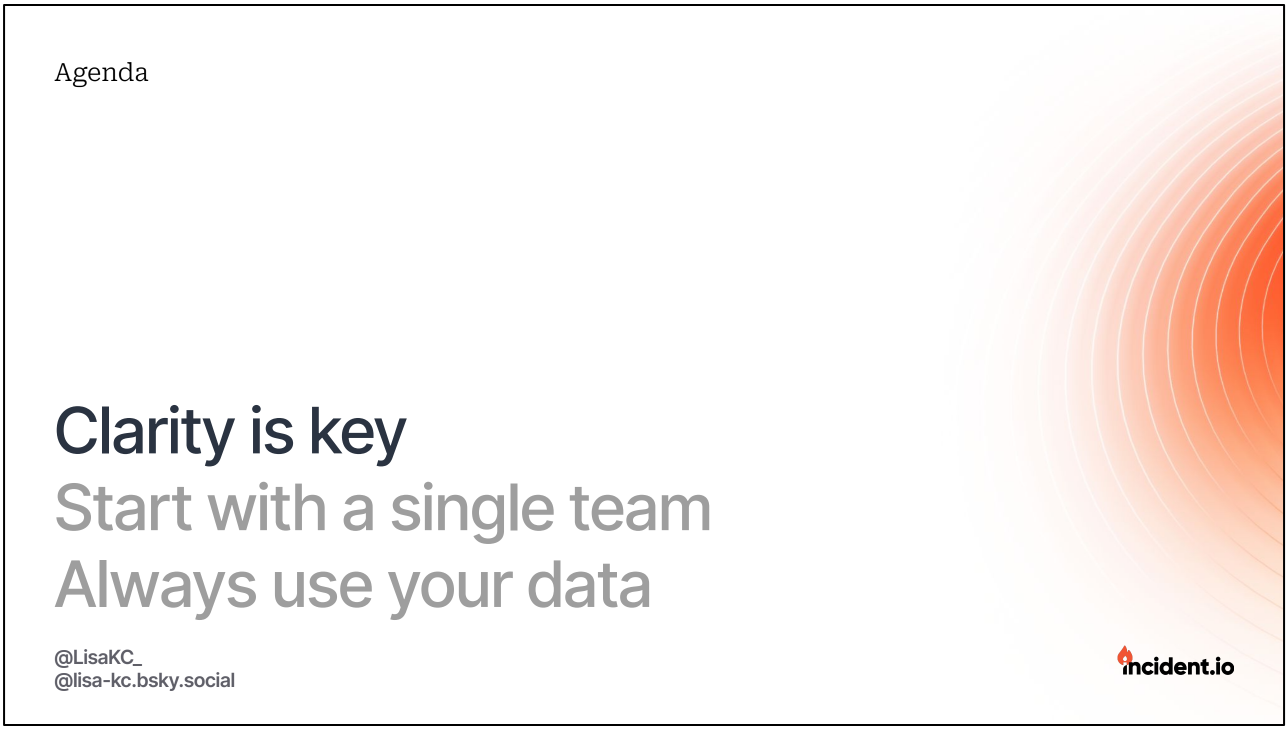
Agenda

Clarity is key
Start with a single team
Always use your data

@LisaKC_
@lisa-kc.bsky.social

incident.io

So, let’s recap our key takeaways



Agenda

Clarity is key

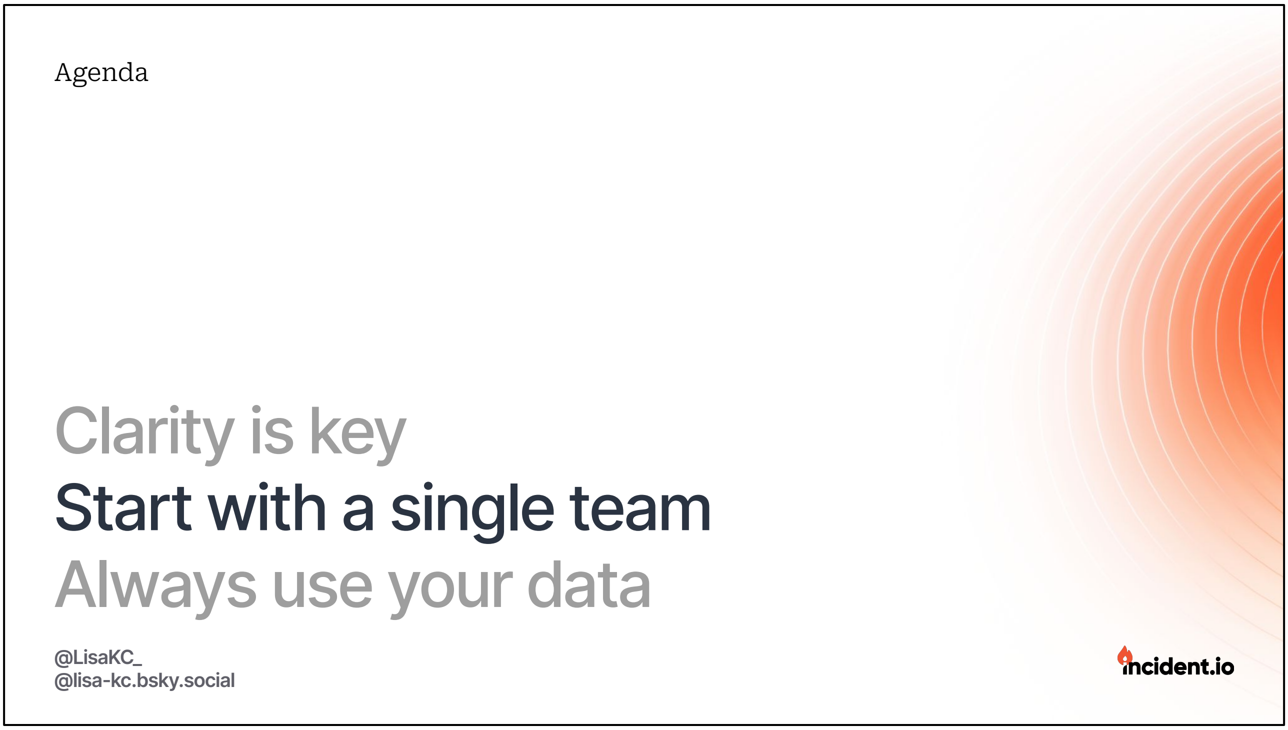
Start with a single team

Always use your data

@LisaKC_
@lisa-kc.bsky.social

incident.io

First off, we realised that clarity is key. With so many objectives floating around, it's easy to lose sight of the bigger picture. But by focusing on incremental value and tackling problems one step at a time, we can deliver value to our team step by step.



Agenda

Clarity is key

Start with a single team

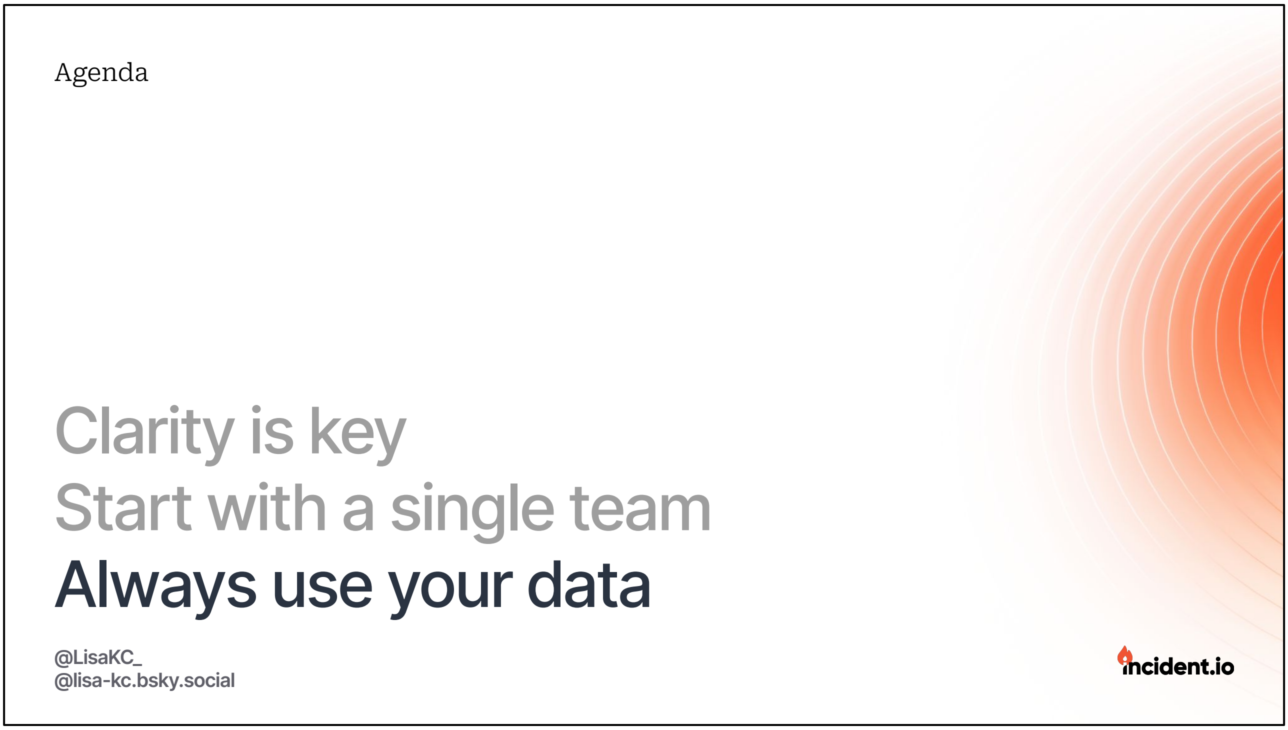
Always use your data

@LisaKC_
@lisa-kc.bsky.social

incident.io

Start small, with a single team and deliver quick wins, to preserve your social capital.

Prioritise depth over breadth – it's better to make one team love you than have every team just-about-remember that you exist



Agenda

Clarity is key

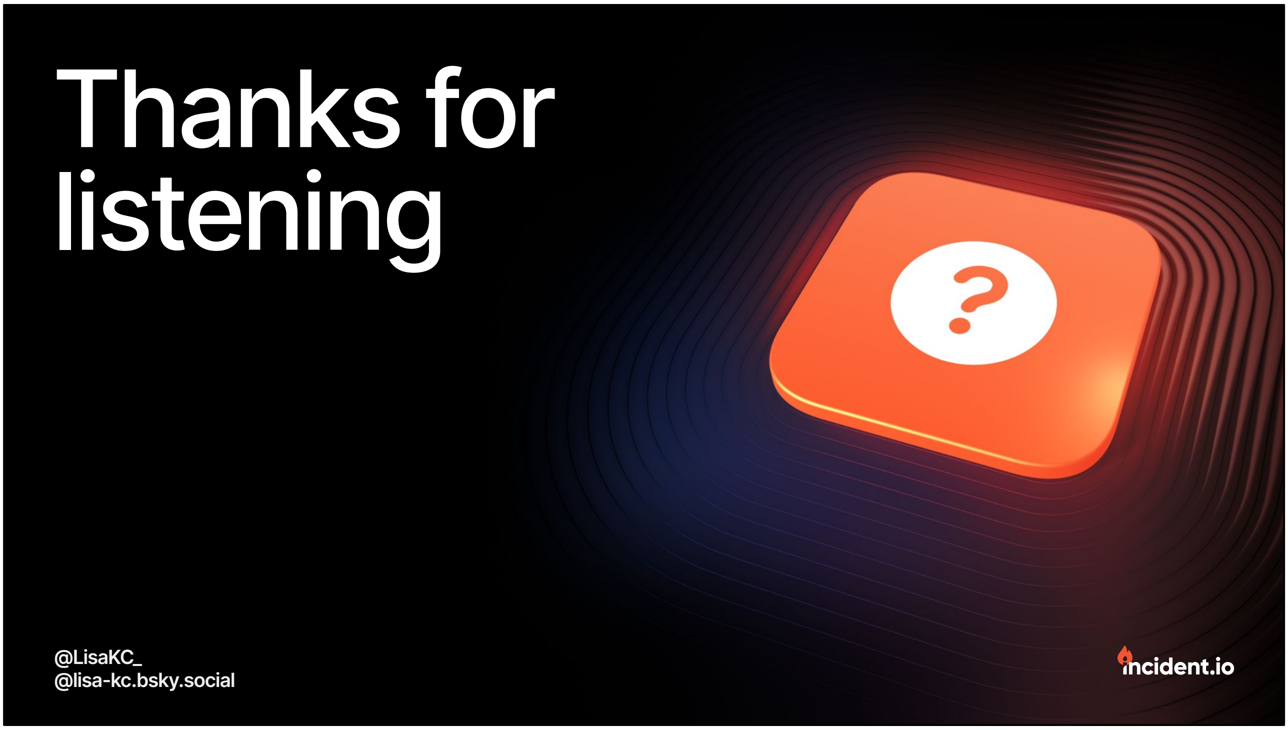
Start with a single team

Always use your data

@LisaKC_
@lisa-kc.bsky.social

incident.io

Collect data which you can use, and then continue to use it so it doesn’t rot. Make it easy for everyone to update the source data to keep it fresh.



Thanks for listening

@LisaKC_
@lisa-kc.bsky.social

incident.io

Thanks so much for listening. If you have any questions or just want to chat about catalogs, incidents, on-call or engineering in general, please come and find me on the conference floor.