

# Achieving unicorn scale

Using Team Topologies at cinch

Toli Apostolidis – Principal Practice Engineer

Andy Norton – Head of Engineering Practice



**What do we mean by  ?**





# Who are we?

**Why are we here?**

**Achieving unicorn scale**  
Using Team Topologies at cinch

The approach is based on Conway's Law, the Theory of Constraints, and the concept of cognitive load.

**Disclaimer:** We don't get any referral fees from Team Topologies.

**What do we mean by ?**

**From spike to sustain in 2 years**  
NOVEMBER 2019 → JANUARY 2022

**Who are we?**

**Why are we here?**

**What going from start-up to scale-up looks like**

**Spoiler alert:** We don't really talk about failures, and what went wrong. (please ask us at the end!)

**Disclaimer:** We don't get any referral fees from Team Topologies.

**Spike. Scale. Sustain.**  
 Weeks Months Years

**Spike.**

**"Oh yeah, the ones with Rylan in the advert"**

**Start lean, it's temporary**

**What are you optimising for?**

**In 2019 cinch was 20 people over 4 teams**

**In 2022 cinch is over 400 people over 20 teams**

**The walking skeleton**

**"We need an MVP"**

**"Organisations design systems that mirror their own communication structure"**

**Expectation vs Reality**

**Every complex system has a constraint. Focusing on anything else is waste.**

**Scale.**

**What are you optimising for?**

**Add teams to the pile.**

**Or... scale horizontally**

**Definitely not the Spotify model!**

**Cognitive load**  
"The amount of mental effort being used in the working memory"

**Team Cognitive Load**

**100% capacity**

**X-as-a-service**  
Facilitating Collaboration

**Focus on setting cars**

**Always learning the good bits**

**Intrinsic**

**Space**

**Stream-aligned teams**

**Platform teams**

**Becoming knowledge stewards**

**Technology Blueprint**

**Exemplars**

**Sustain.**

**To sustain means to support.**

**Focus on the gaps**

**What are you optimising for?**

**Observability is the ability to understand how a software system is behaving internally by evaluating its outputs.**

**Working on (the virtual) walls**

**In summary**

**Minimal viable everything.**

**Feedback is a gift**

**Change when you need to**

**So, how did we scale ?**

**Thank you.**

**From spike to sustain in 2 years**  
NOVEMBER 2019 → JANUARY 2022

**Start lean, it's temporary**

**What are you optimising for?**

**In 2019 cinch was 20 people over 4 teams**

**In 2022 cinch is over 400 people over 20 teams**

**The walking skeleton**

**"We need an MVP"**

**"Organisations design systems that mirror their own communication structure"**

**Expectation vs Reality**

**Every complex system has a constraint. Focusing on anything else is waste.**

**Scale.**

**What are you optimising for?**

**Add teams to the pile.**

**Or... scale horizontally**

**Definitely not the Spotify model!**

**Cognitive load**  
"The amount of mental effort being used in the working memory"

**Team Cognitive Load**

**100% capacity**

**X-as-a-service**  
Facilitating Collaboration

**Focus on setting cars**

**Always learning the good bits**

**Intrinsic**

**Space**

**Stream-aligned teams**

**Platform teams**

**Becoming knowledge stewards**

**Technology Blueprint**

**Exemplars**

**Sustain.**

**To sustain means to support.**

**Focus on the gaps**

**What are you optimising for?**

**Observability is the ability to understand how a software system is behaving internally by evaluating its outputs.**

**Working on (the virtual) walls**

**In summary**

**Minimal viable everything.**

**Feedback is a gift**

**Change when you need to**

**So, how did we scale ?**

**Thank you.**

**From spike to sustain in 2 years**  
NOVEMBER 2019 → JANUARY 2022

**Start lean, it's temporary**

**What are you optimising for?**

**In 2019 cinch was 20 people over 4 teams**

**In 2022 cinch is over 400 people over 20 teams**

**The walking skeleton**

**"We need an MVP"**

**"Organisations design systems that mirror their own communication structure"**

**Expectation vs Reality**

**Every complex system has a constraint. Focusing on anything else is waste.**

**Scale.**

**What are you optimising for?**

**Add teams to the pile.**

**Or... scale horizontally**

**Definitely not the Spotify model!**

**Cognitive load**  
"The amount of mental effort being used in the working memory"

**Team Cognitive Load**

**100% capacity**

**X-as-a-service**  
Facilitating Collaboration

**Focus on setting cars**

**Always learning the good bits**

**Intrinsic**

**Space**

**Stream-aligned teams**

**Platform teams**

**Becoming knowledge stewards**

**Technology Blueprint**

**Exemplars**

**Sustain.**

**To sustain means to support.**

**Focus on the gaps**

**What are you optimising for?**

**Observability is the ability to understand how a software system is behaving internally by evaluating its outputs.**

**Working on (the virtual) walls**

**In summary**

**Minimal viable everything.**

**Feedback is a gift**

**Change when you need to**

**So, how did we scale ?**

**Thank you.**

**What going from start-up to scale-up up looks like**

**What you'll need to be deliberate about**

**Why managing cognitive load is key**

Spoiler alert

**We don't really talk about  
failures, and what went wrong.**

**(please ask us at the end!)**

## Disclaimer

**We don't get any referral fees  
from Team Topologies.**



cinch.co.uk

Find a car ▾ Car finance Part exchange How cinch works ▾ News & reviews ▾ My profile 

# Buying a car online - just cinch it.

Buy, finance and part exchange. Free home delivery or collection. 14-day money-back guarantee.

Select make ▾ | Select model ▾ | Search all 11,281 cars

Not sure where to start? Try our [Help Me Choose tool](#)



Chat

Check out our 12,909 reviews

  
Trustpilot

**“Oh yeah, the ones  
with Rylan in the  
advert”**

# From spike to sustain in 2 years

NOVEMBER

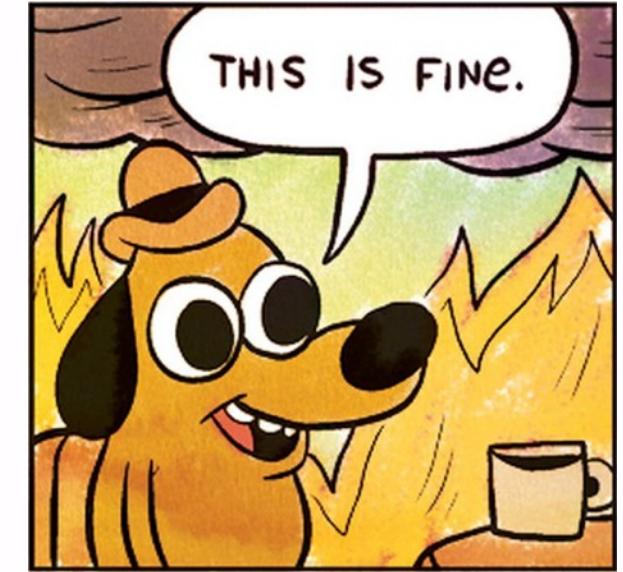
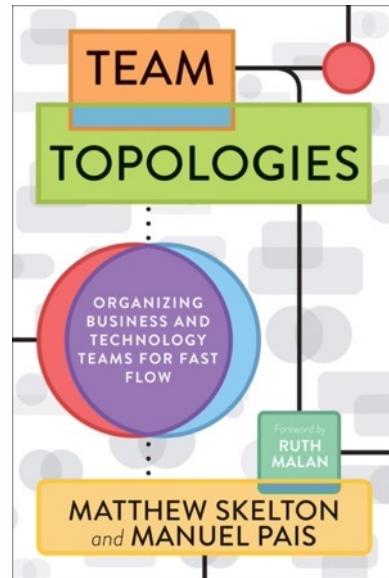
2019



JANUARY

2022

# The 'starting a start-up in 2019' starter-pack



# 2019

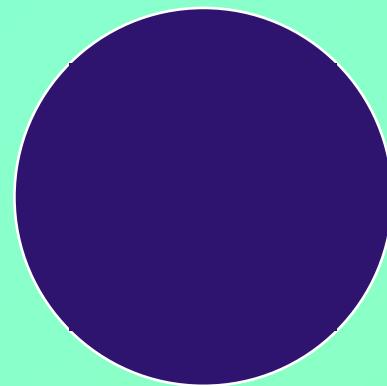
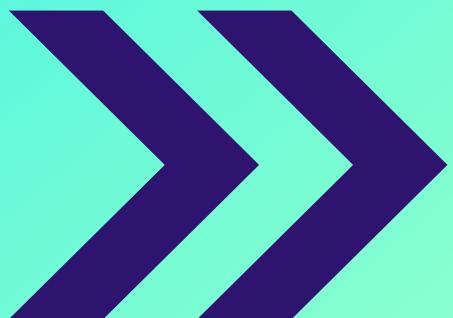
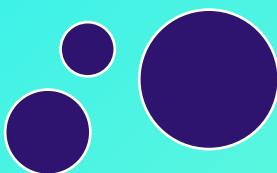


In 2019 cinch was 20  
people over 4 teams

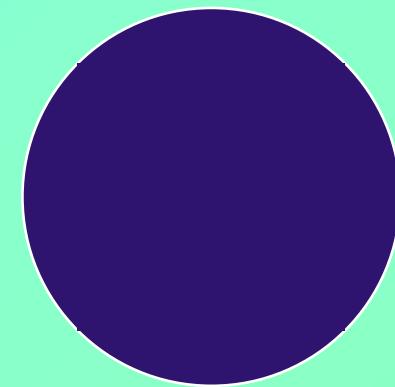
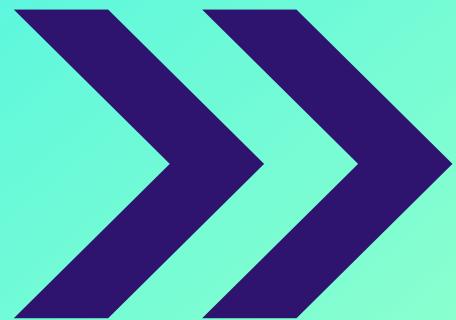
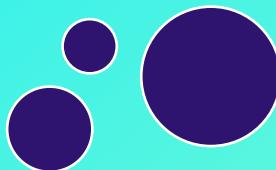
2022 is over 400  
In ~~2019~~ cinch ~~was~~ 20  
people over ~~4~~<sup>20</sup> teams



# Spike. Scale. Sustain.



# Spike. Scale. Sustain.



**Weeks**

**Months**

**Years**

# Spike.



**'What is the simplest thing we  
can program that will convince  
us we are on the right track?'**

- Ward Cunningham

**“It's a spike, it's not  
going into production.”**



**Start lean,  
it's temporary**

**What are you  
optimising for?**



**Spike**



**Scale**



**Sustain**



**Weeks**

**Months**

**Years**

**Time**



**Spike**



**Scale**



**Sustain**

Getting stuff live

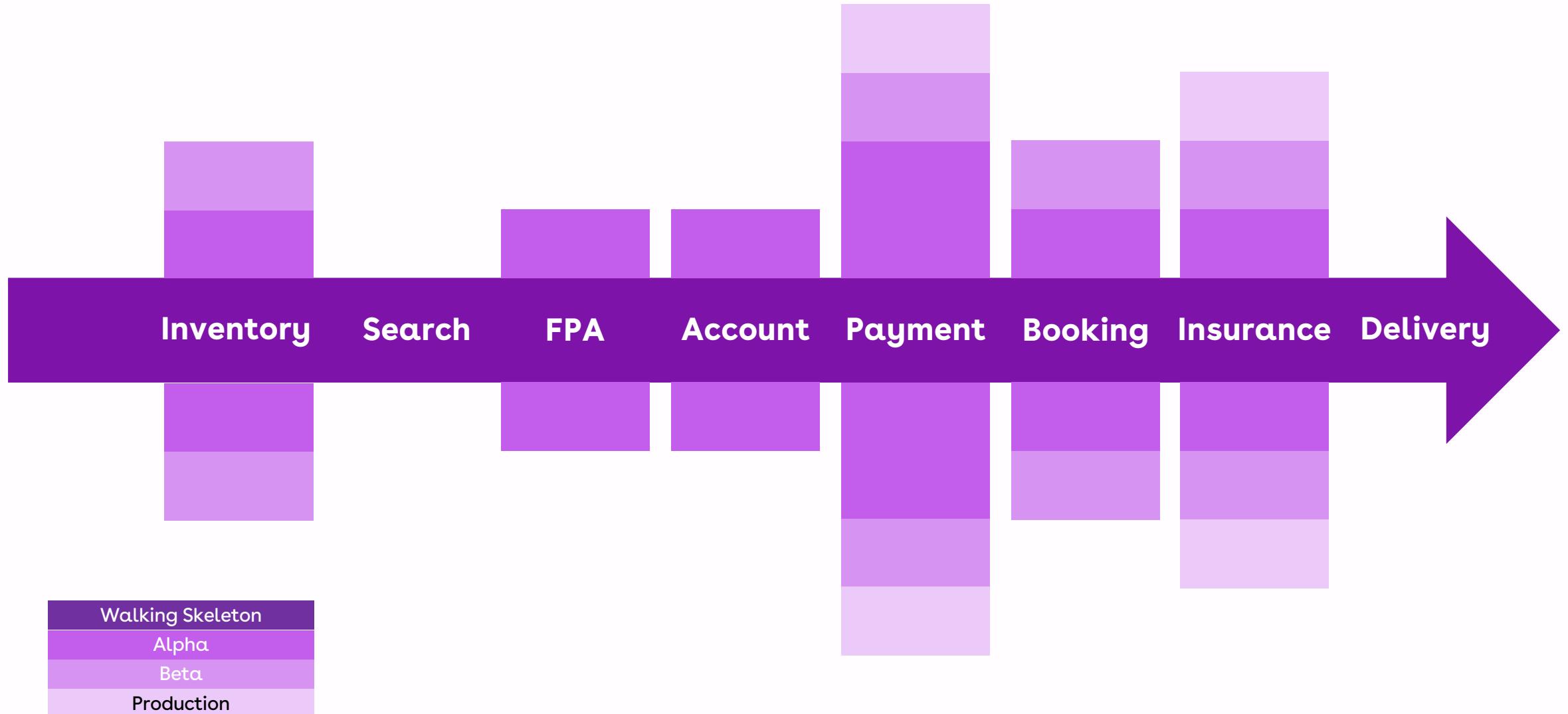
**Weeks**

**Months**

**Years**

**Time**

# The walking skeleton



**"We need an MVP"**

**"We need an MVP"**

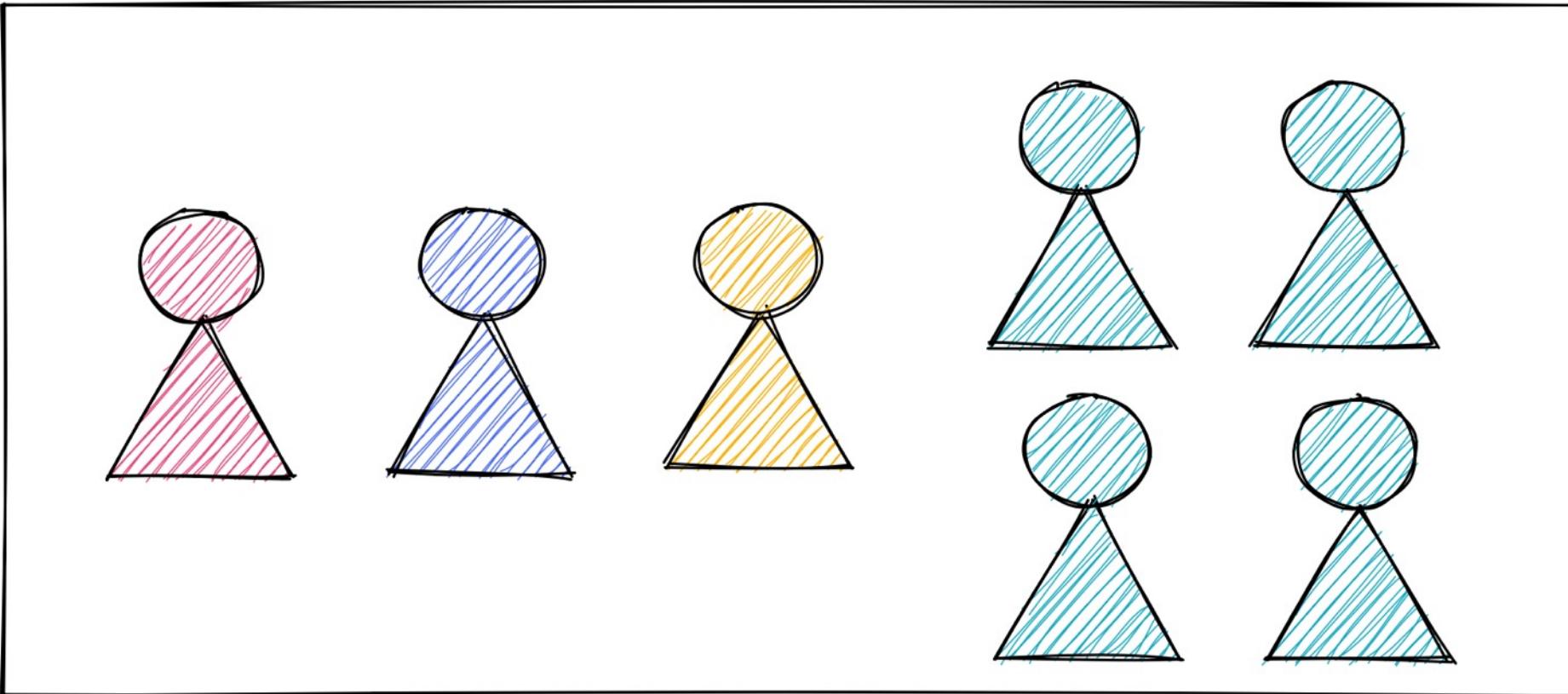
**was actually**

**"We need an MVO"**

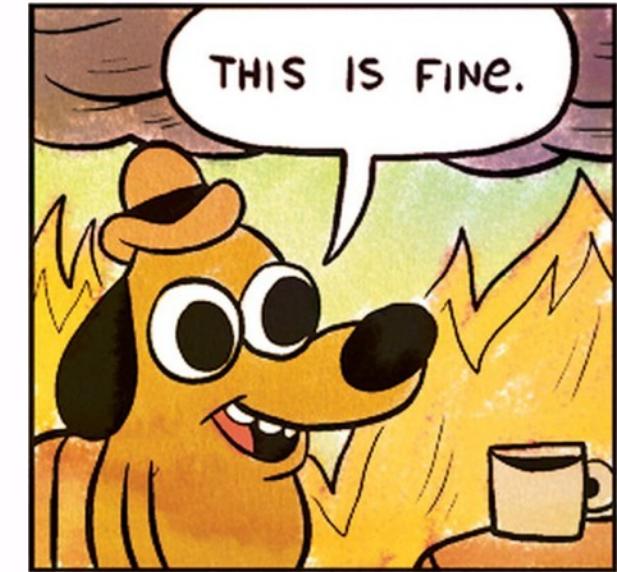
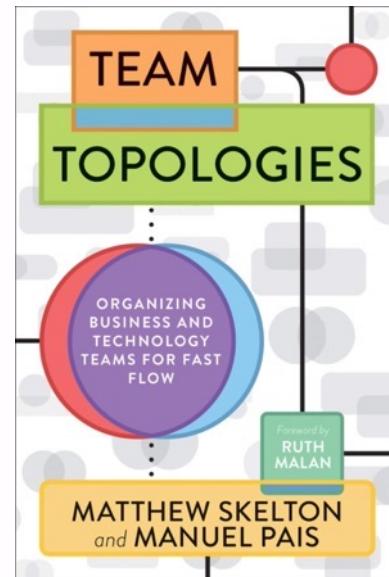
THE UNIT OF  
DELIVERY  
IS THE TEAM



# Yes, cross-functional teams

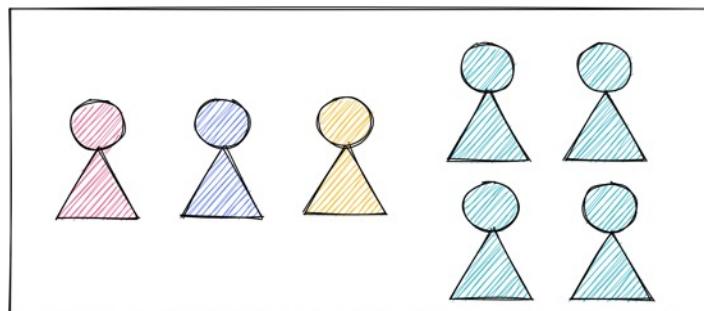
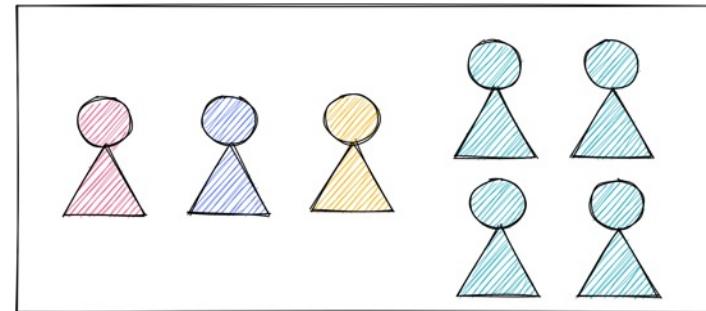
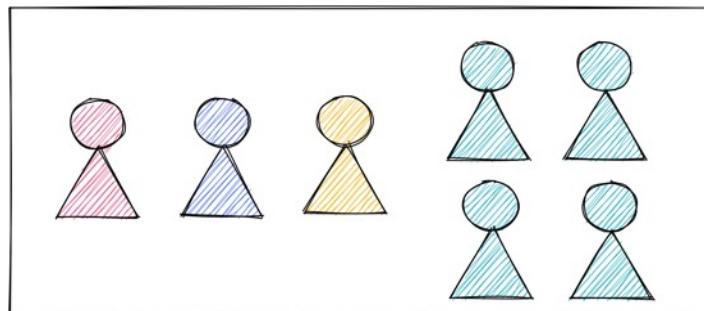
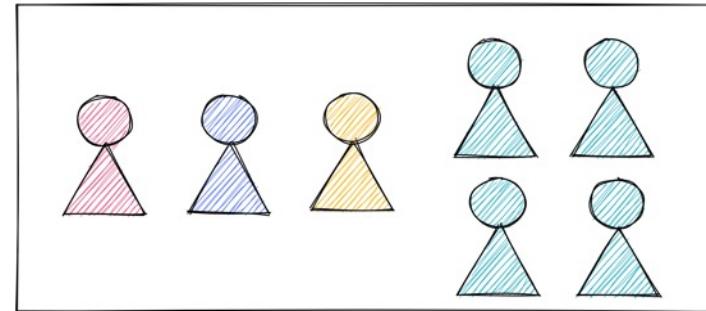
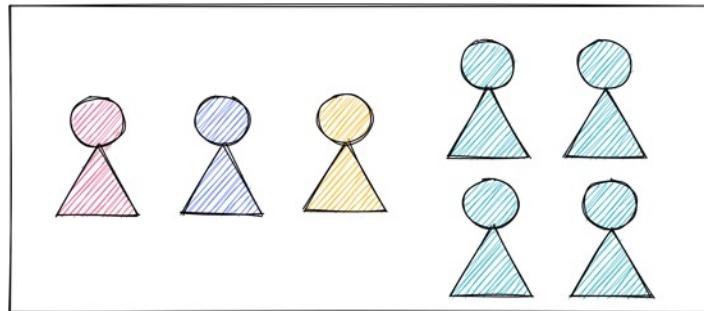


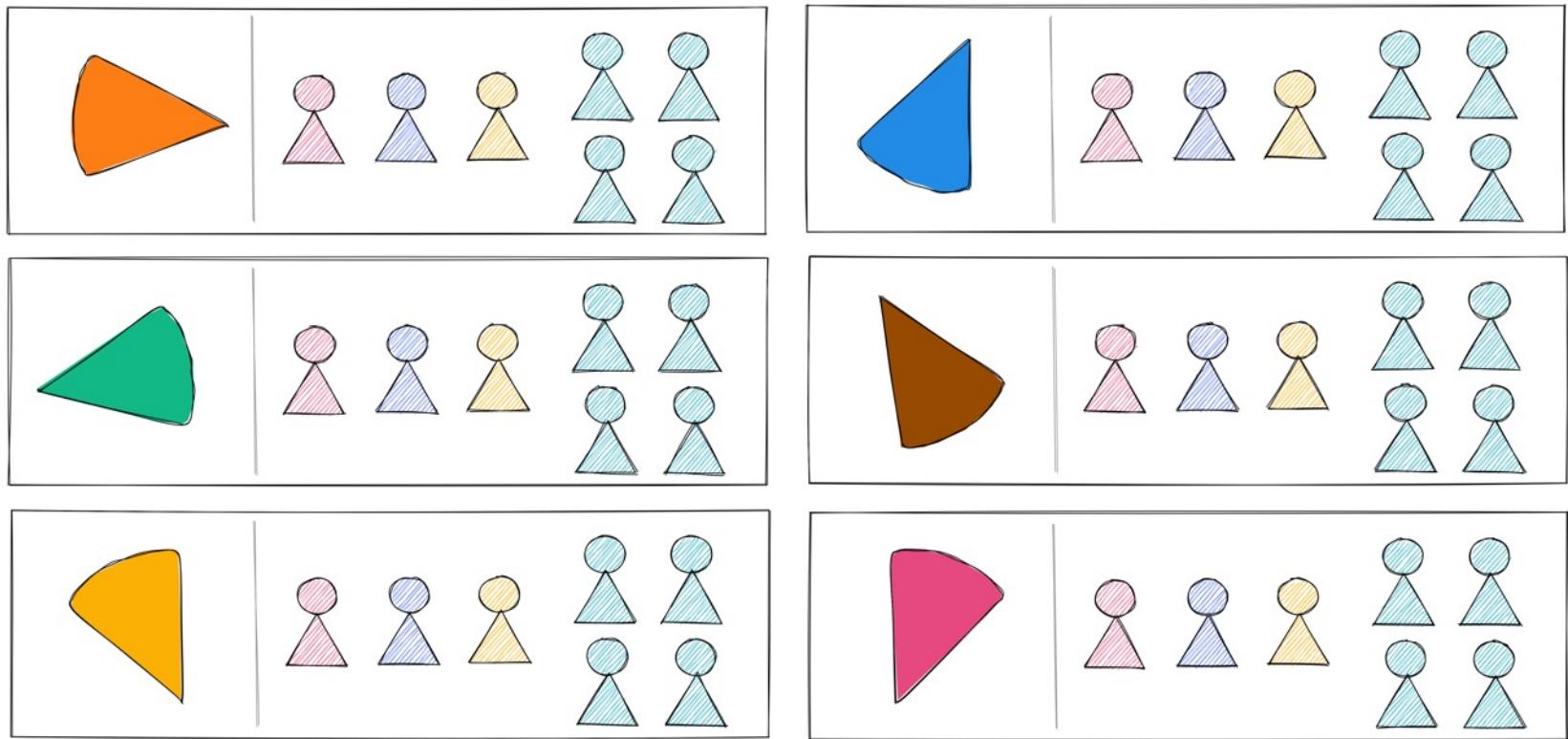
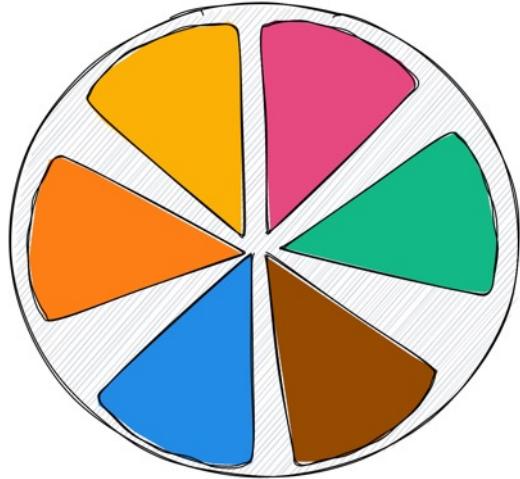
# The 'starting a start-up in 2019' starter-pack



**Still not the Spotify  
model!**

# “Viable” was a handful.





**Slice up the domain.**



**Spike**



**Scale**



**Sustain**

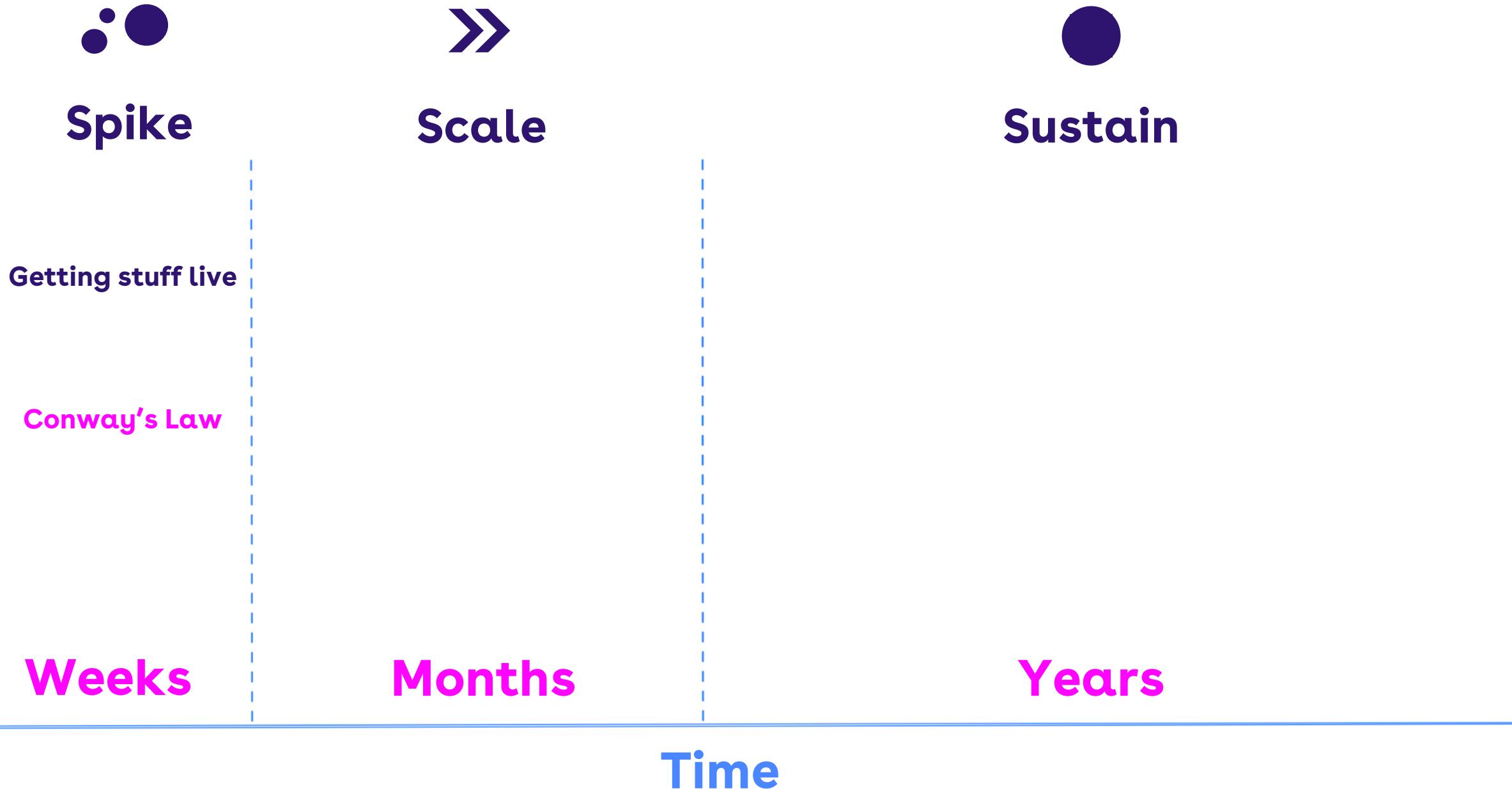
Getting stuff live

**Weeks**

**Months**

**Years**

**Time**



**“Organisations design  
systems that mirror their  
own communication  
structure”**

- Conway's Law

# Expectation



# Reality





**Spike**

**Scale**

**Sustain**

**Getting stuff live**

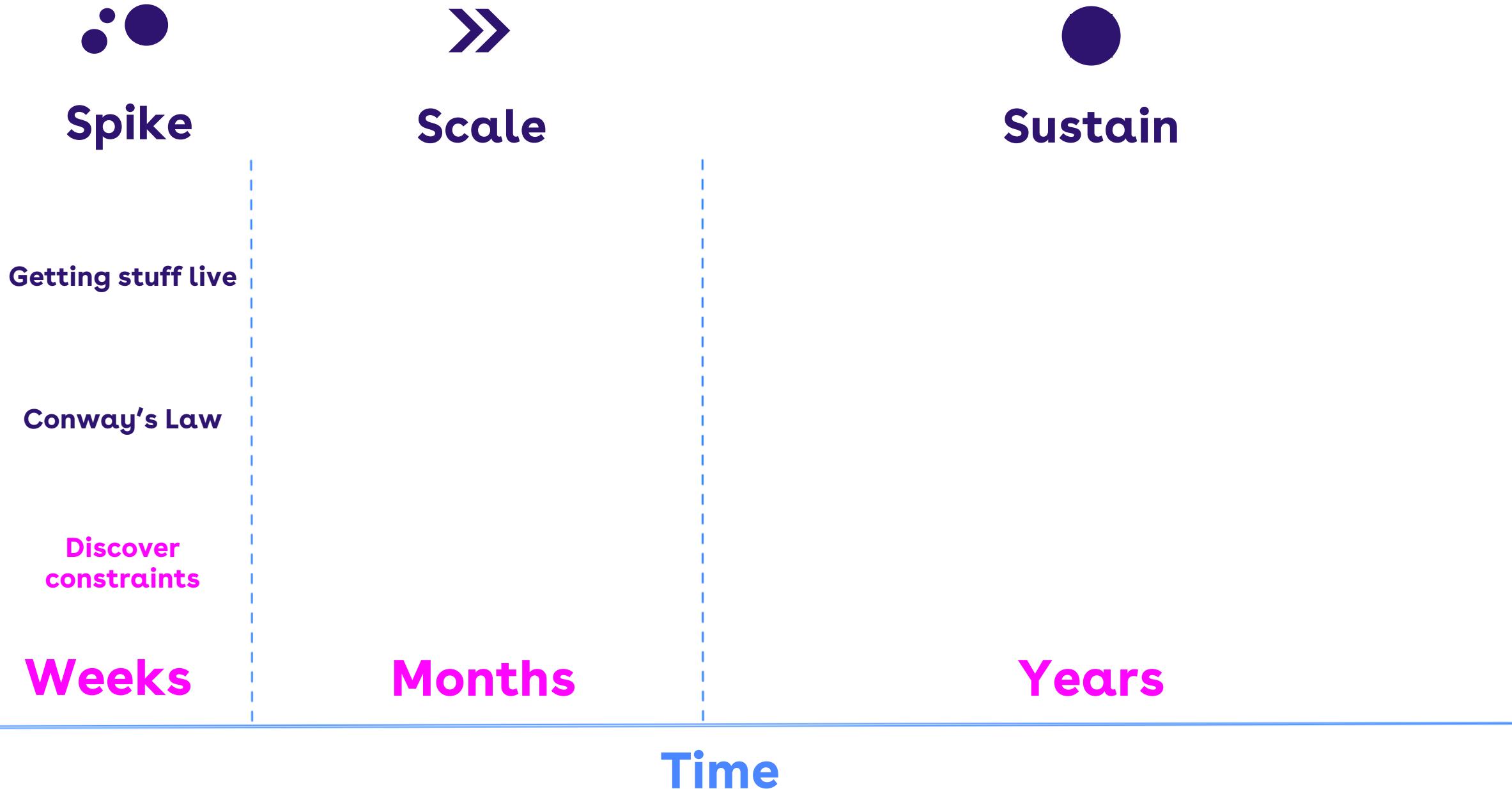
**Conway's Law**

**Weeks**

**Months**

**Years**

**Time**



**Every complex system  
has a constraint.**

**Focusing on anything  
else is waste.**



**Spike**

**Scale**

**Sustain**

Getting stuff live

Conway's Law

Discover  
constraints

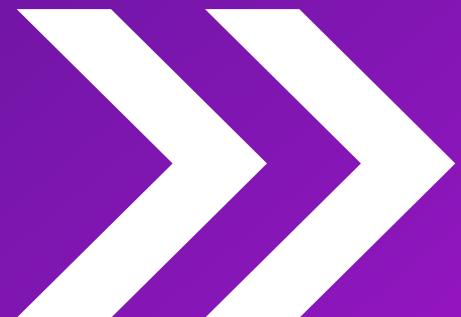
**Weeks**

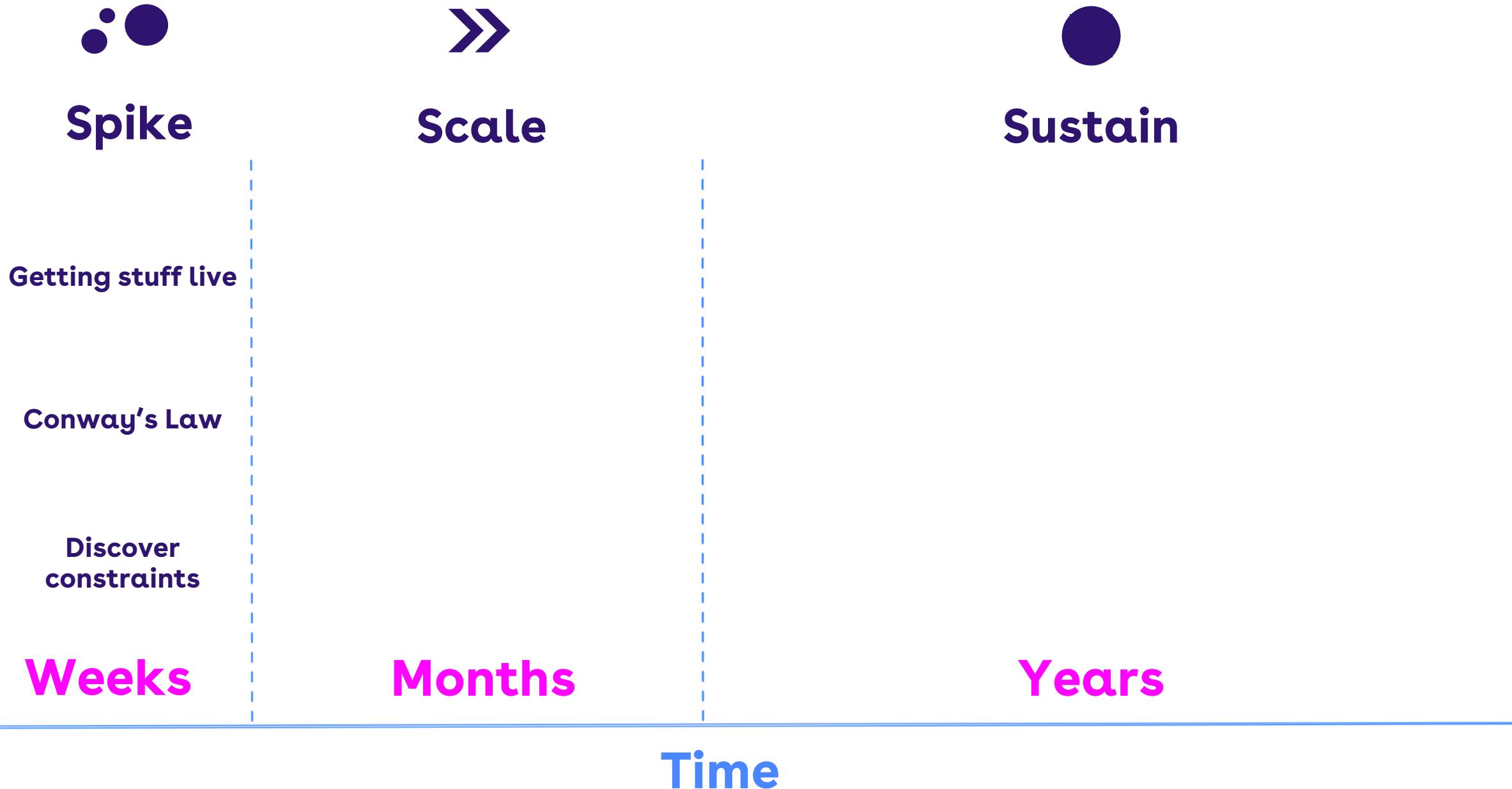
**Months**

**Years**

**Time**

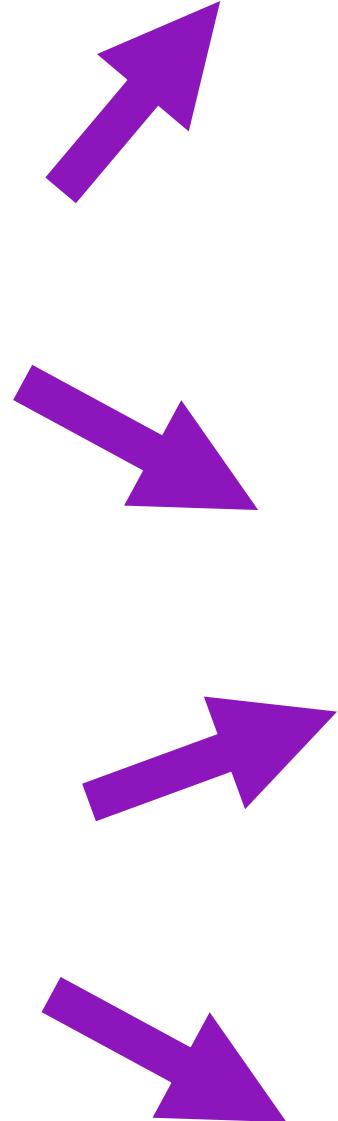
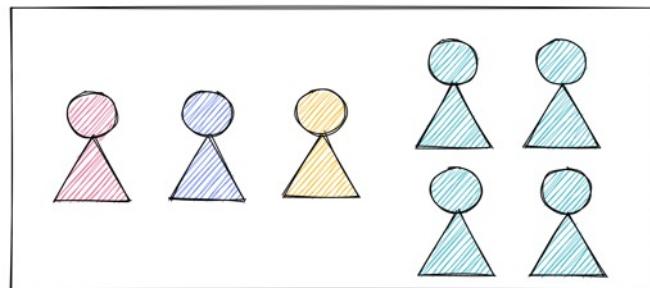
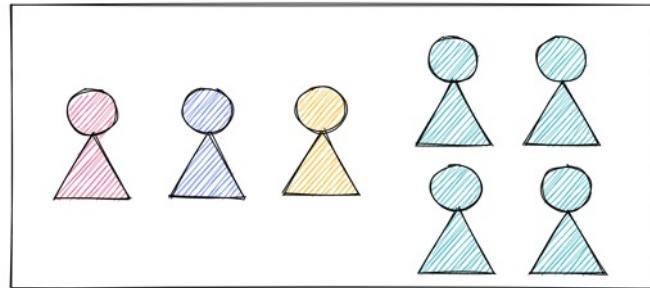
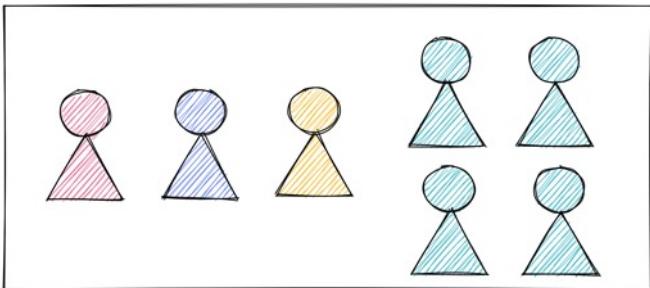
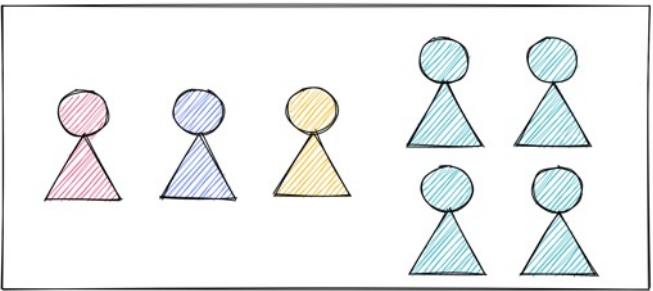
# Scale.

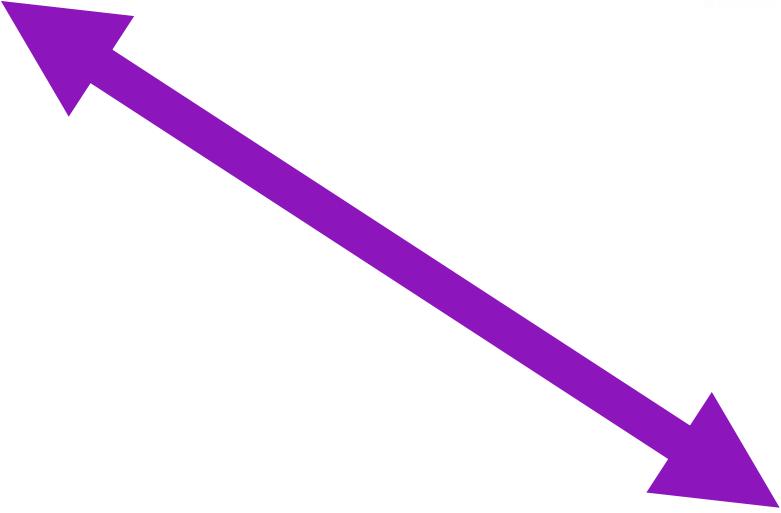
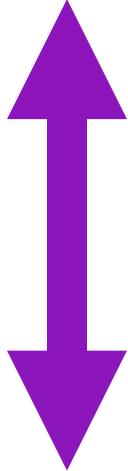
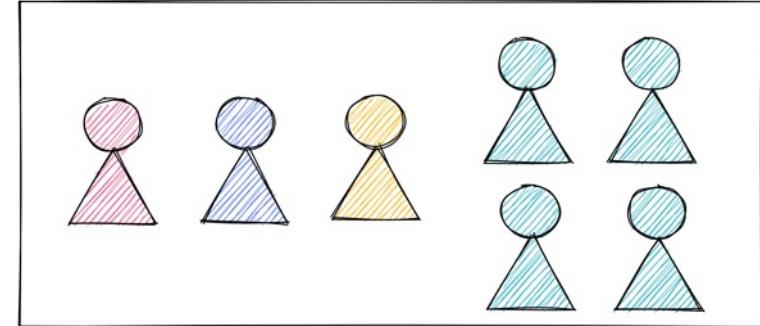
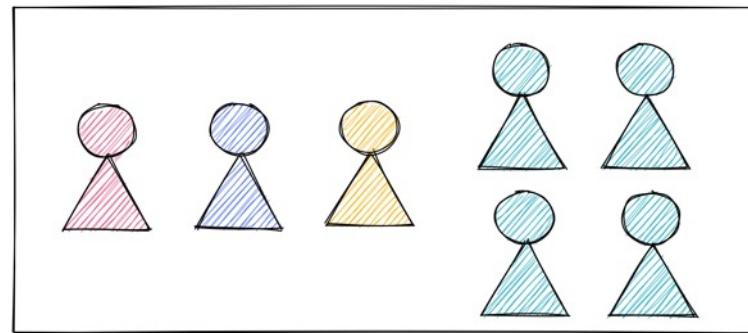
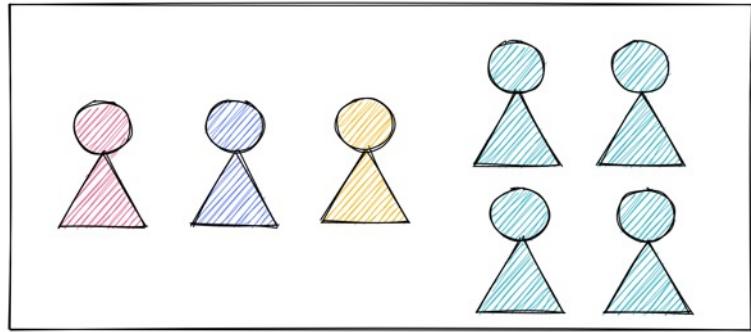
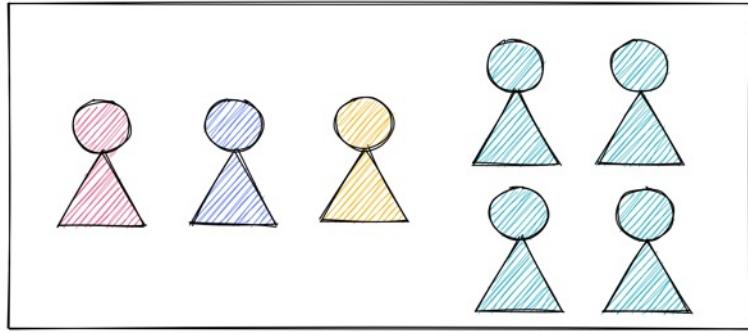




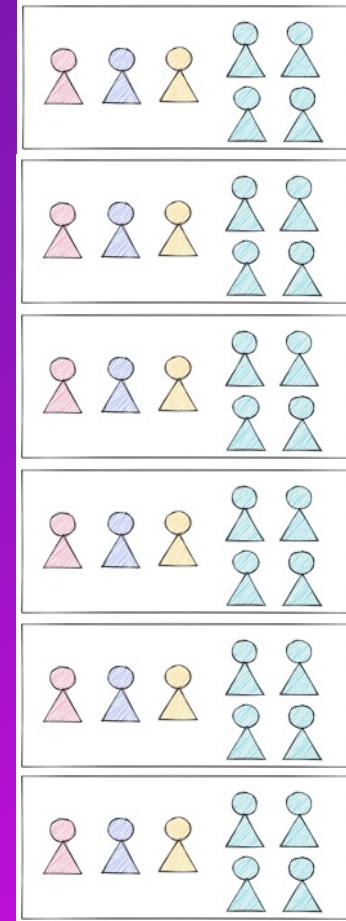
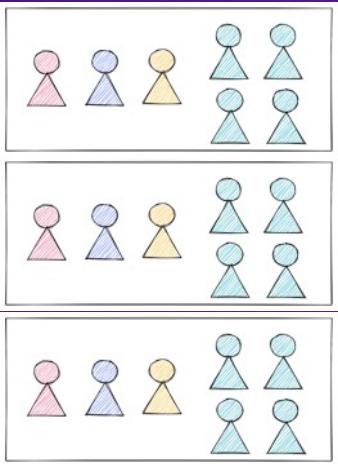
**What are you  
optimising for?**





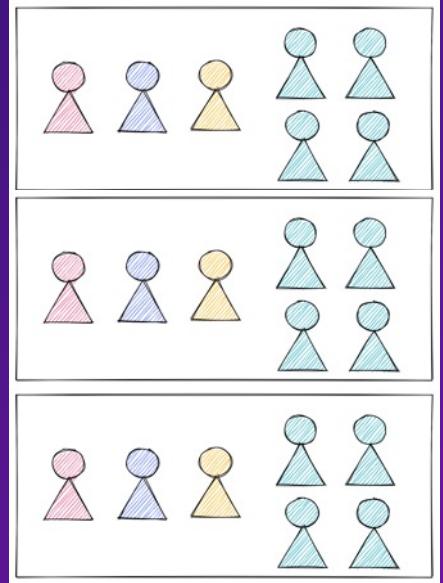


# Add teams to the pile.

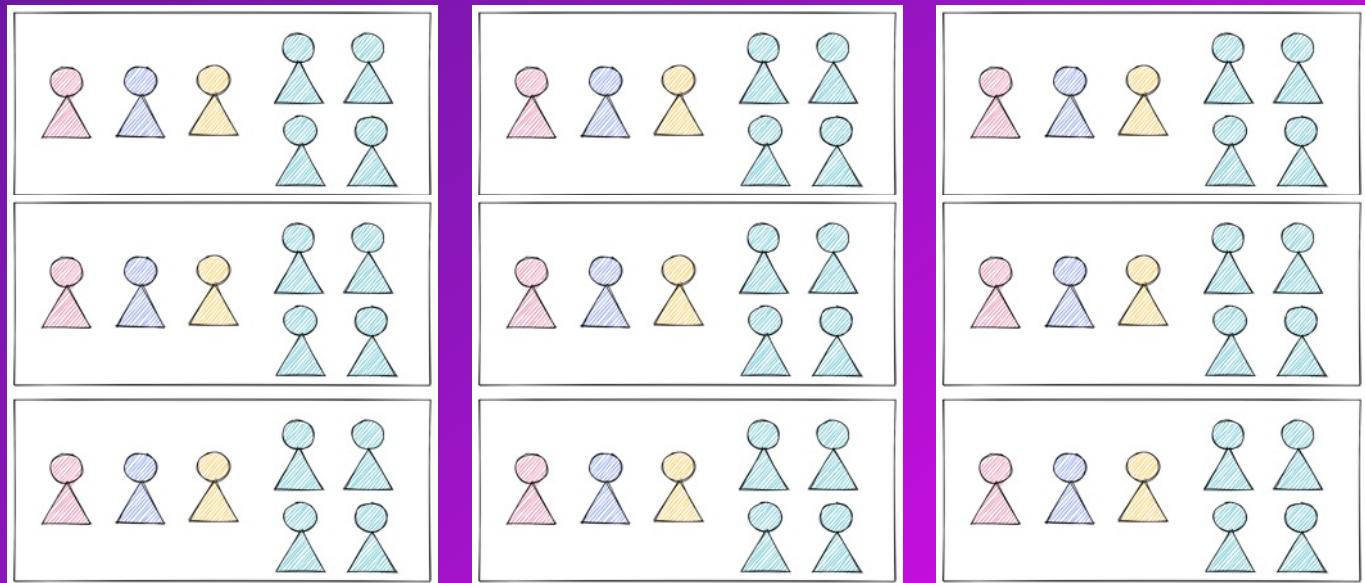




imgflip.com



Or... scale  
horizontally



**Definitely not the  
Spotify model!**





## Spike

Getting stuff live

Conway's Law

Discover constraints

**Weeks**



## Scale

Right-sized teams

Guard-rails

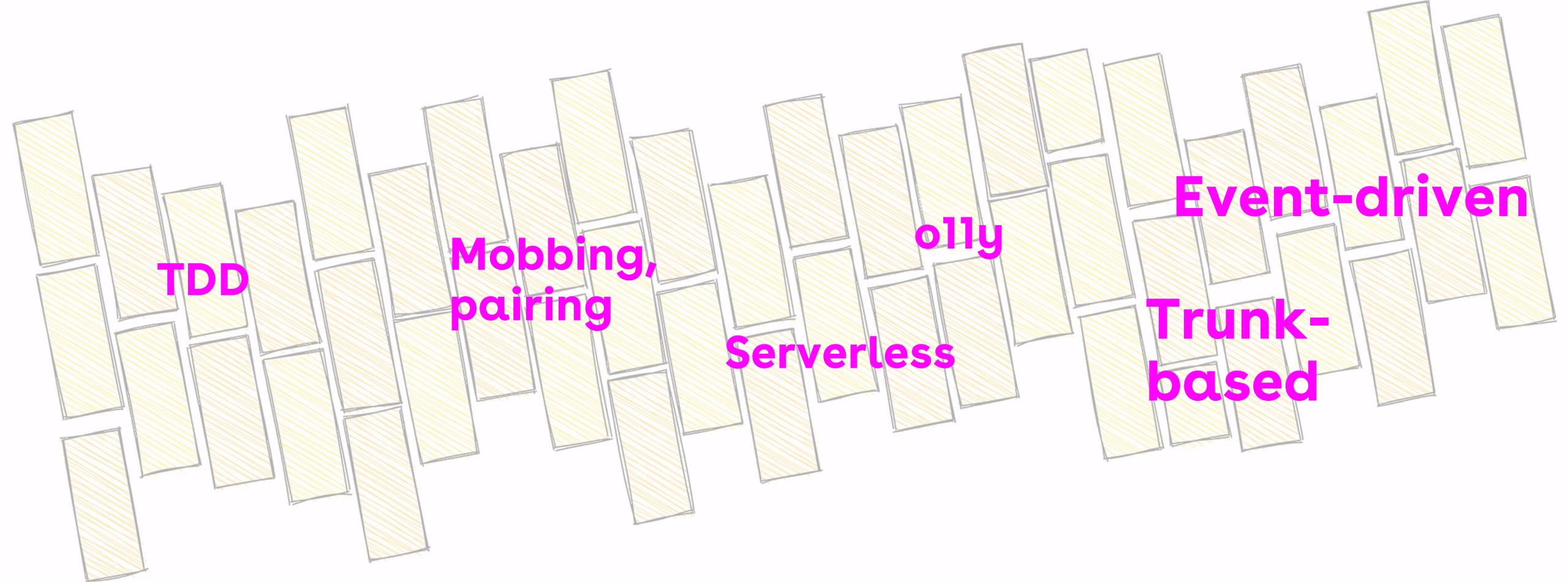
**Months**



## Sustain

**Years**

**Time**

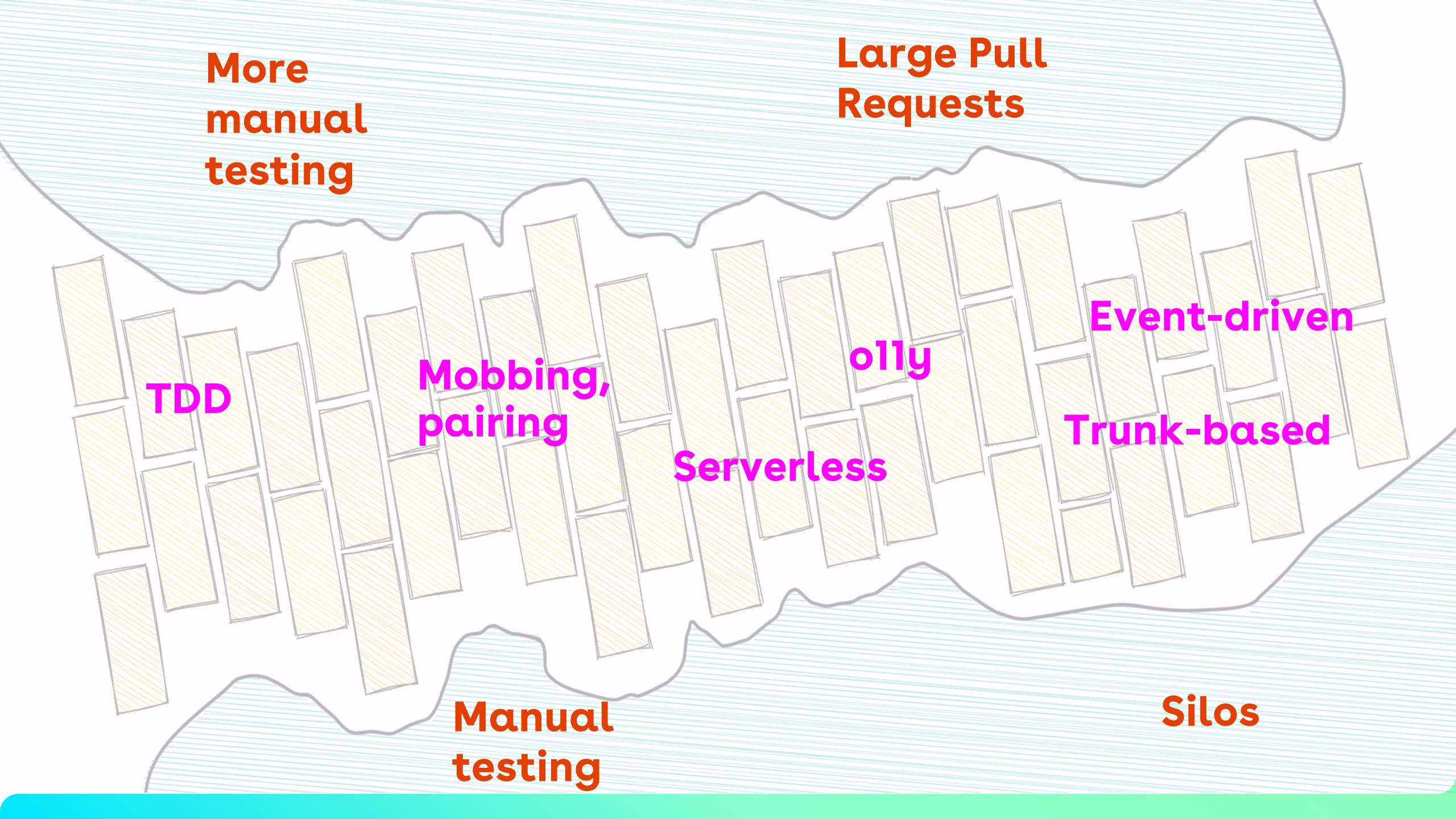


TDD

Mobbing,  
pairing

Serverless  
olly

Event-driven  
Trunk-  
based



More  
manual  
testing

TDD

Mobbing,  
pairing

Large Pull  
Requests

olly

Serverless

Event-driven  
Trunk-based

Manual  
testing

Silos





## Spike

Getting stuff live

Conway's Law

Discover constraints

**Weeks**



## Scale

Right-sized teams

Guard-rails

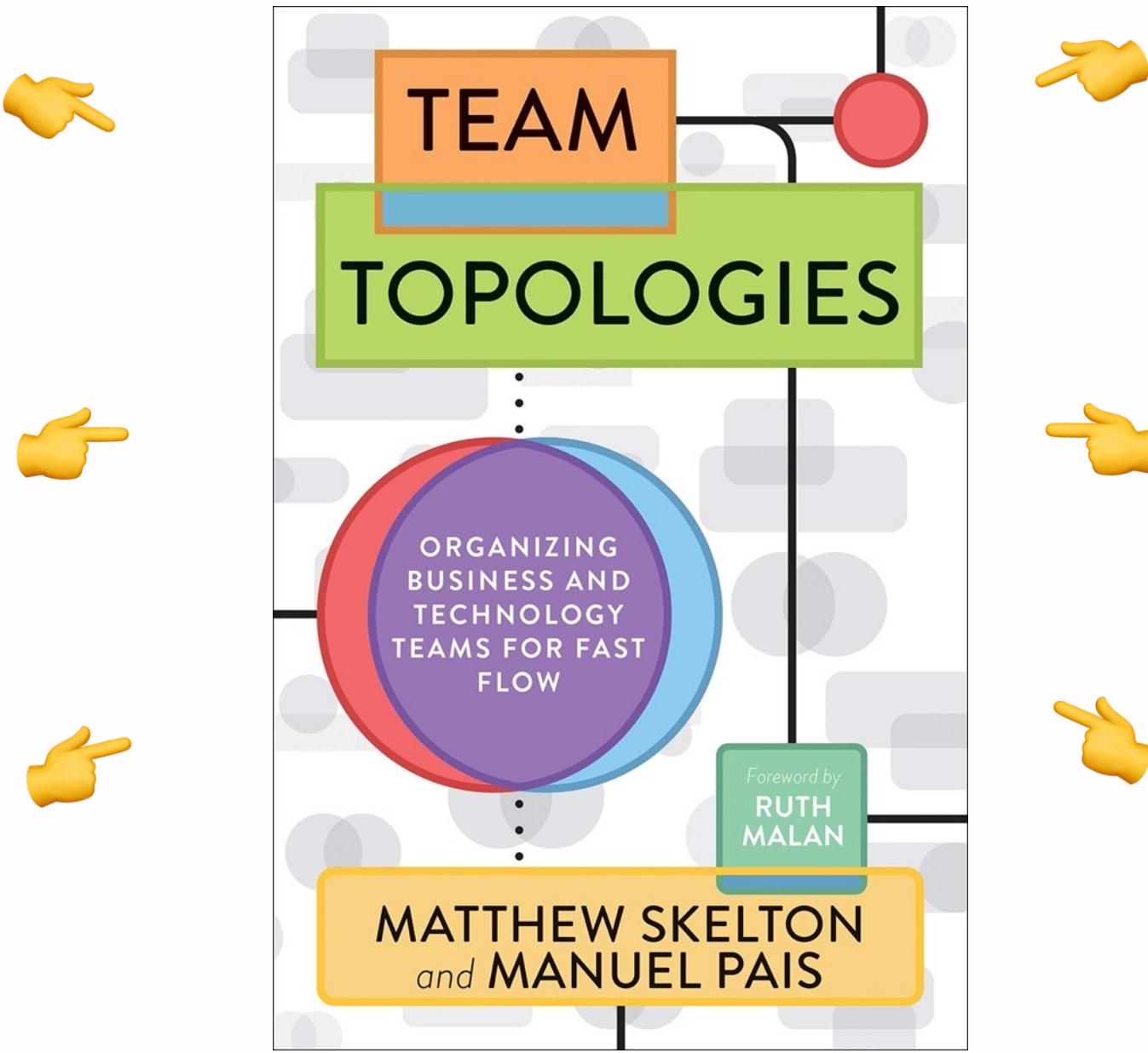
Cognitive load

**Months**



## Sustain

**Time**



## Disclaimer

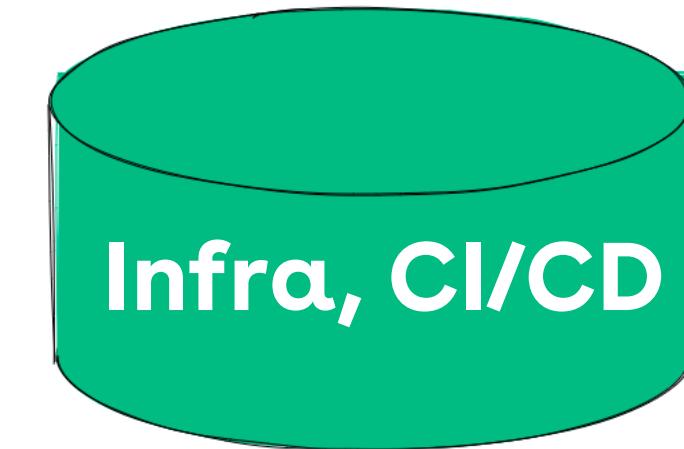
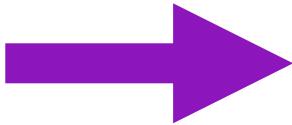
**We don't get any referral fees  
from Team Topologies.**

# Cognitive load

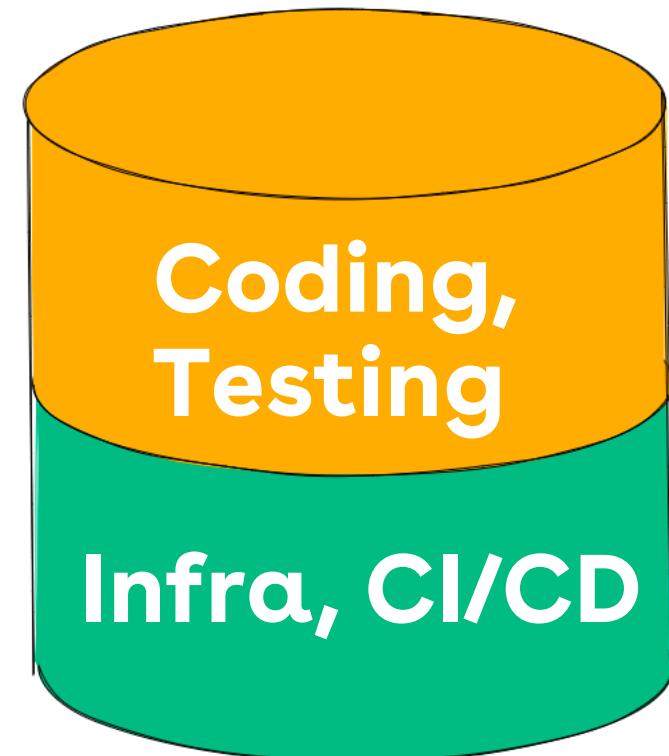
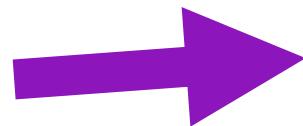
**“The amount of mental effort being used in the working memory”**

- John Sweller, Psychologist, 1988

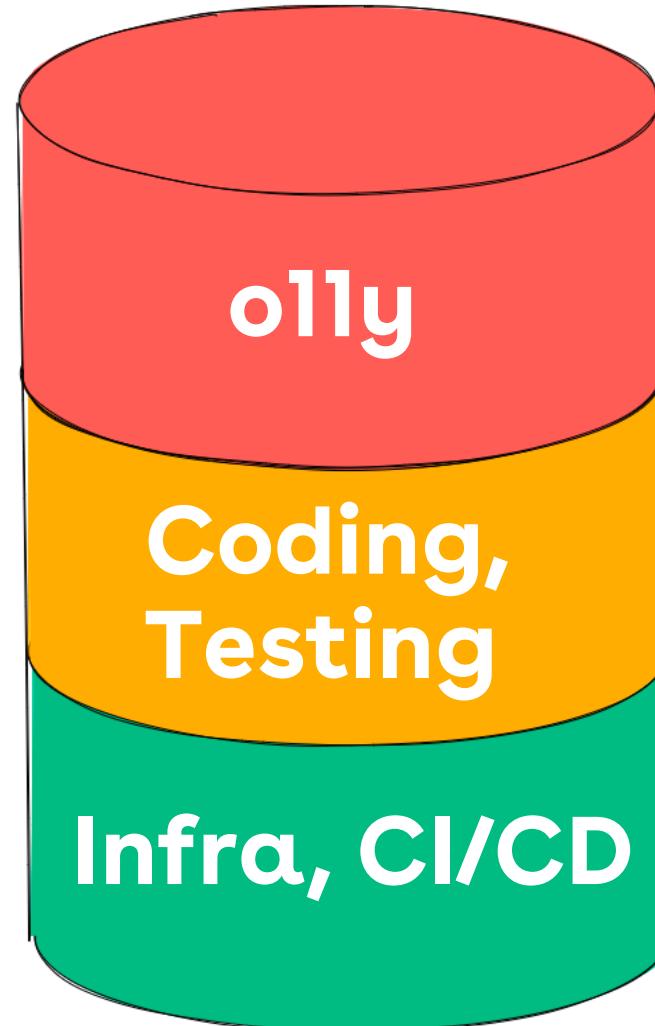
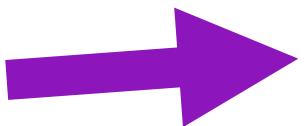
**Team  
Cognitive  
Load**



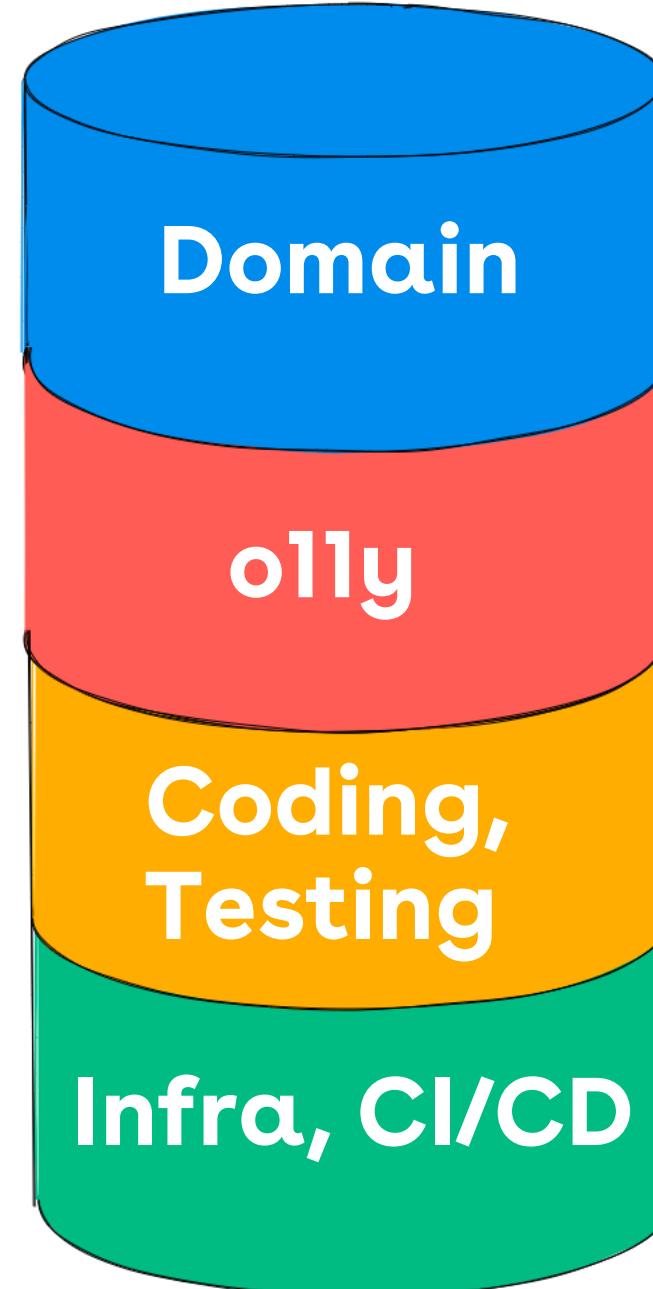
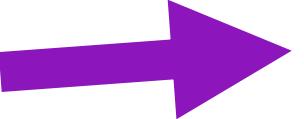
**Team  
Cognitive  
Load**



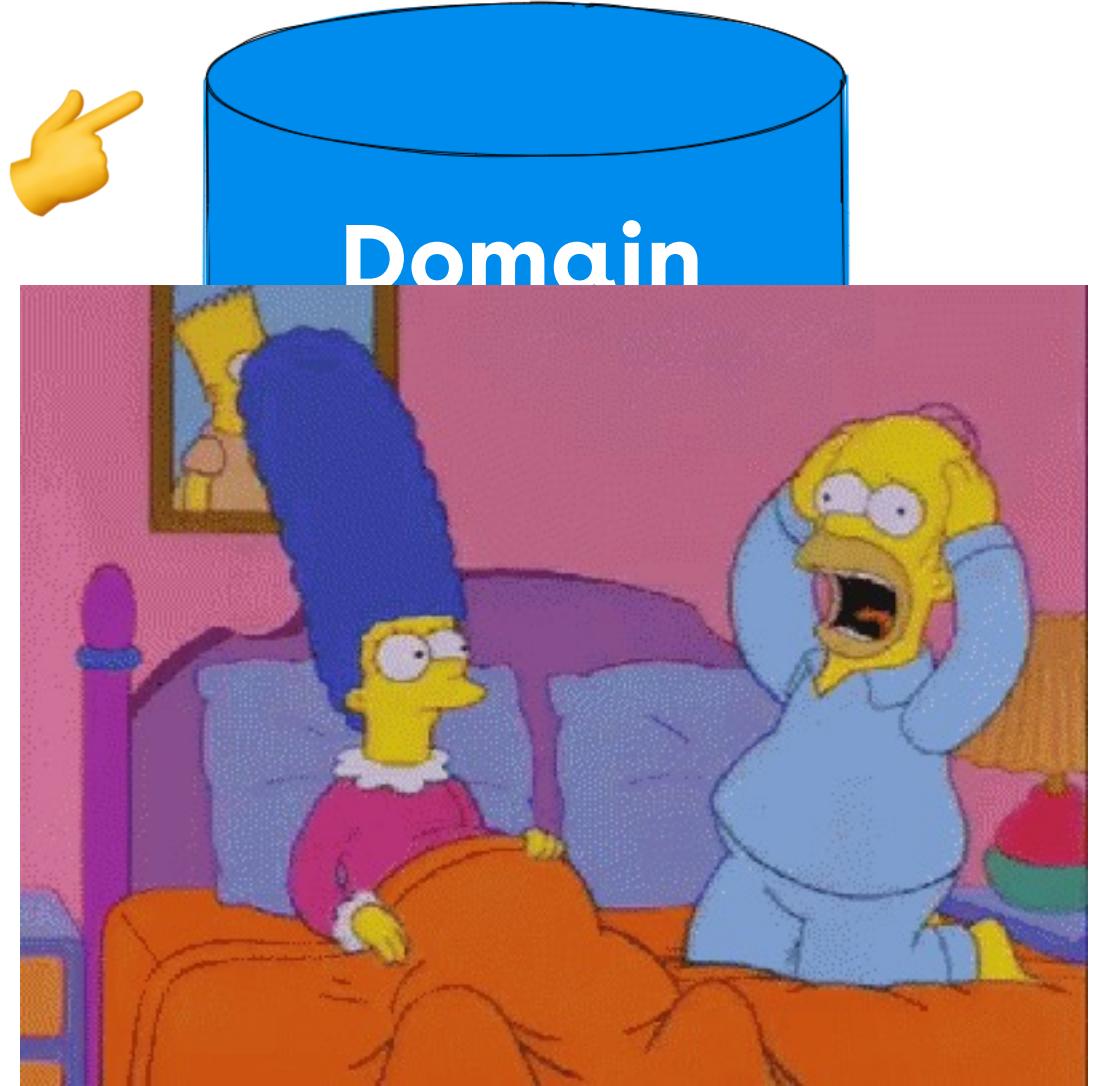
**Team  
Cognitive  
Load**



**Team  
Cognitive  
Load**



# 100% capacity

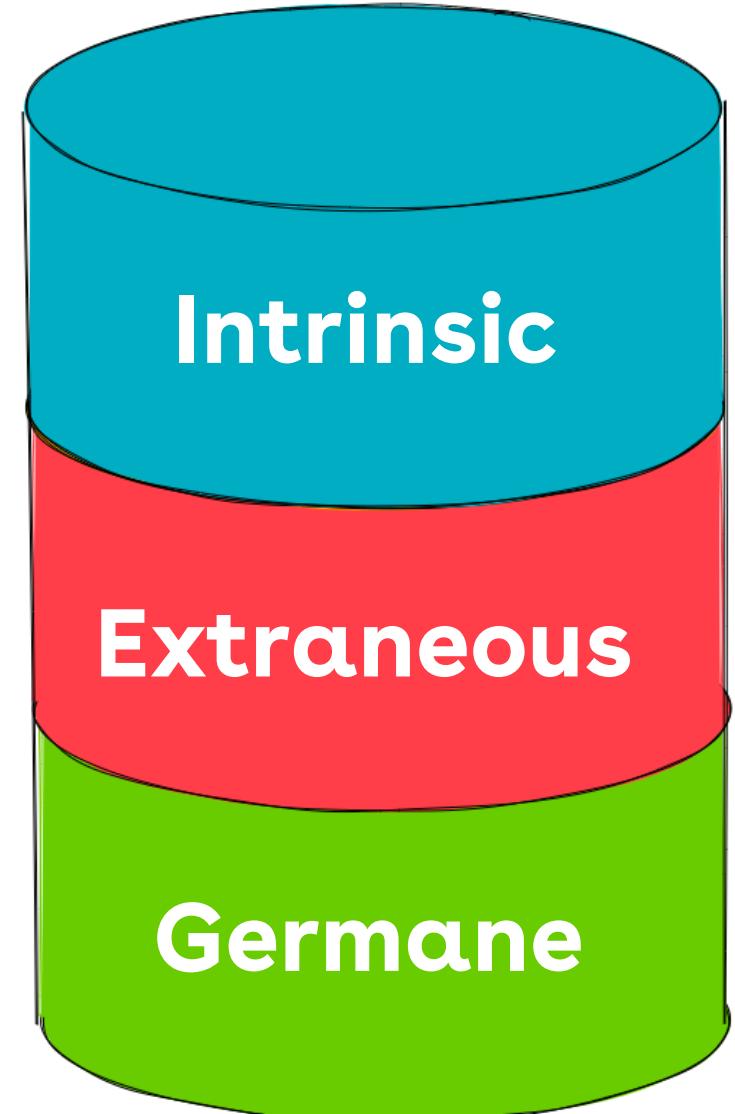


Infra, CI/CD

**What are types in Typescript?**

**“One sec, I’ll find that AWS command again.”**

**What service should offer vehicle details?**



**Focus on  
selling cars**

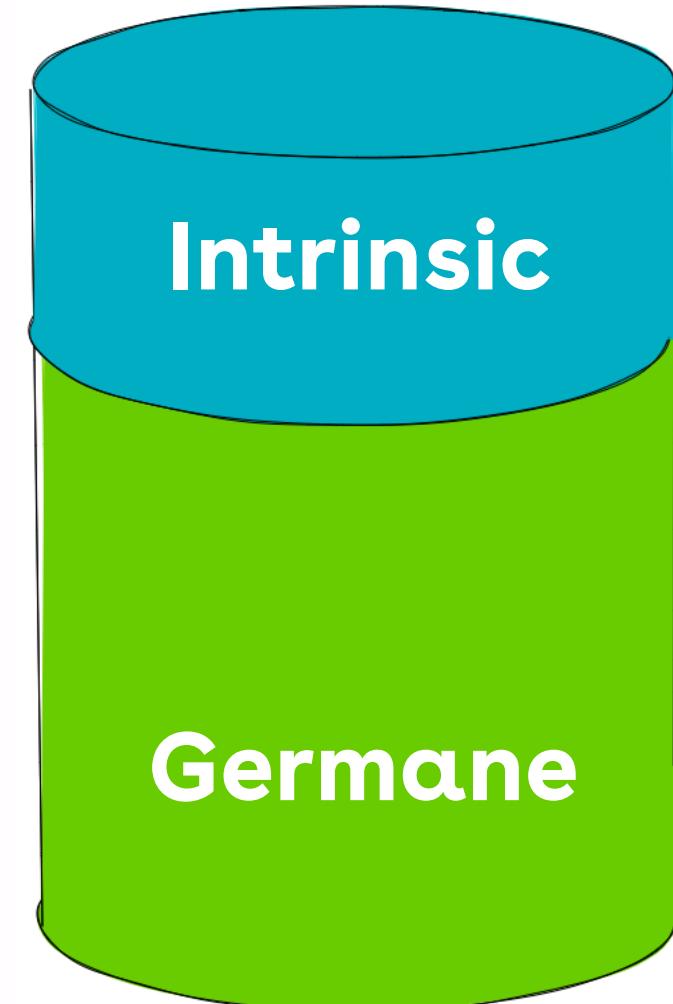
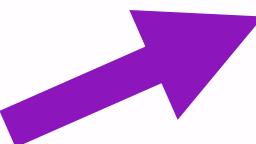


**Germane**

**Always  
learning the  
good bits**



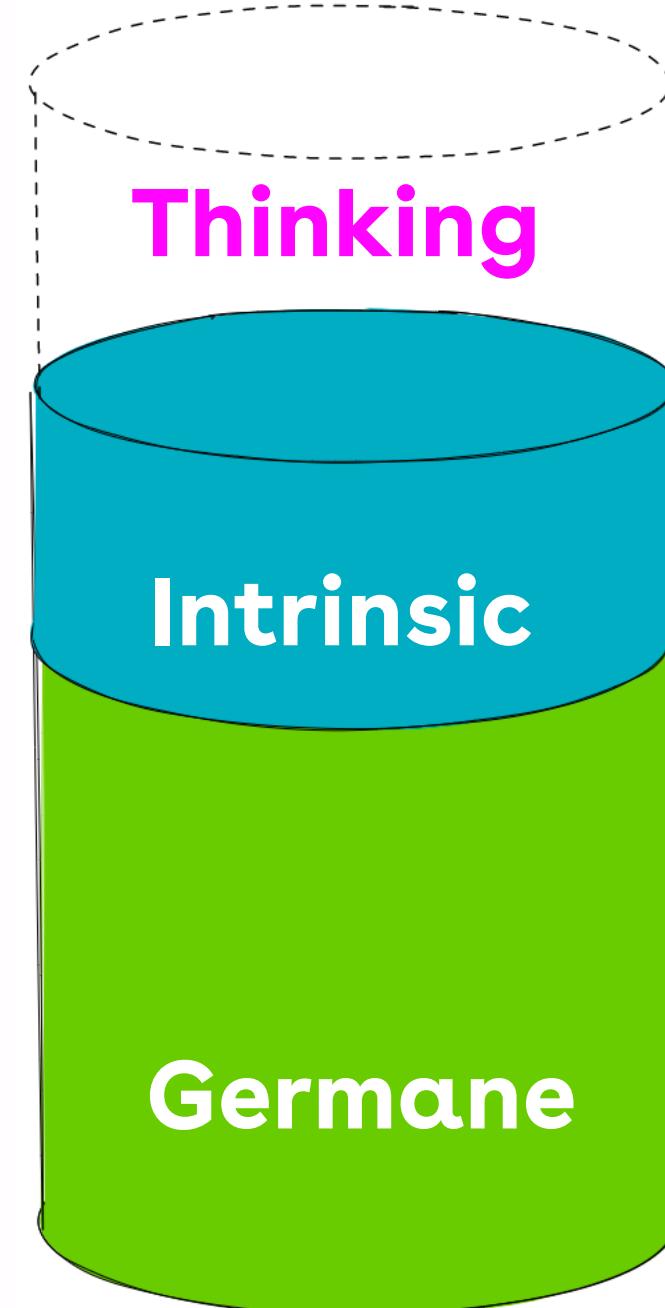
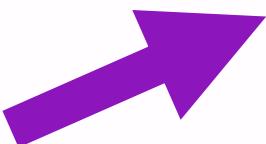
**Focus on  
selling cars**



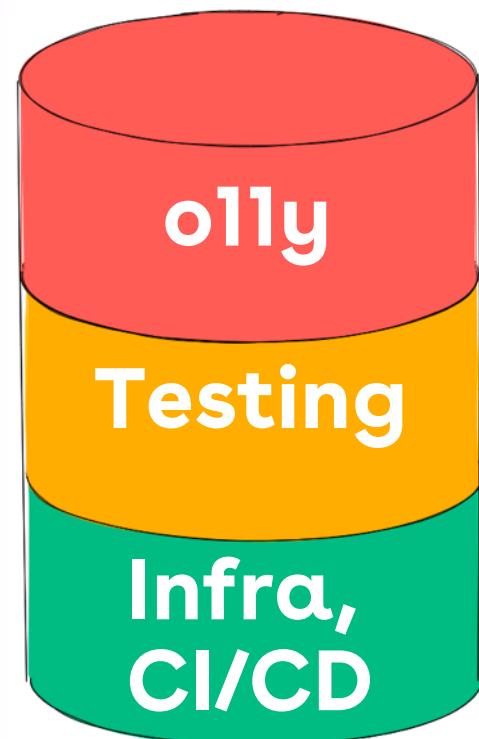
**Always  
learning the  
good bits**



**Focus on  
selling cars**



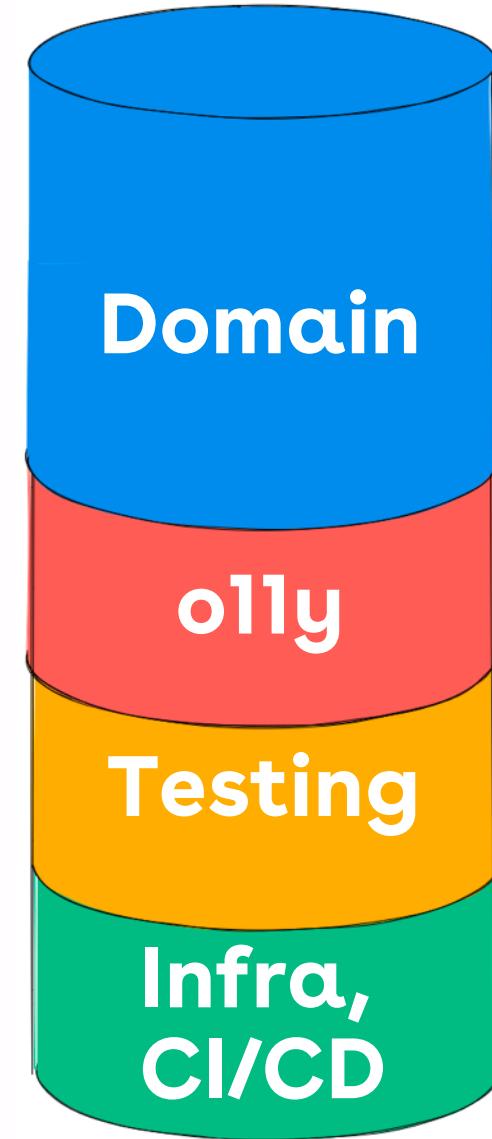
# Intrinsic



**Germane**



**Intrinsic**



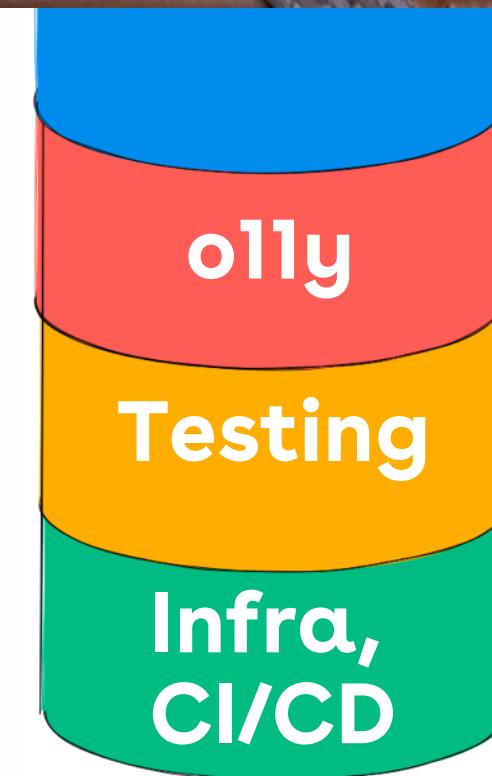
# Space

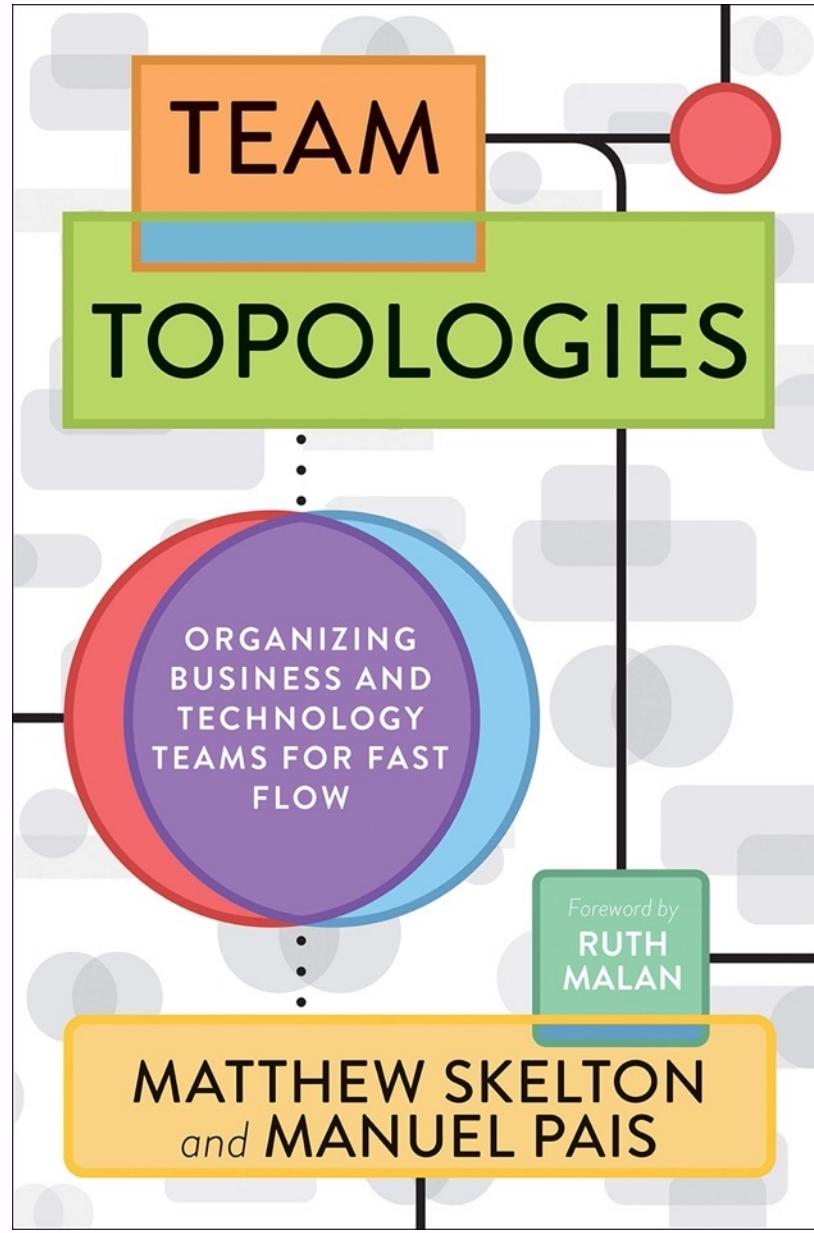


# Germane



# Intrinsic





# **Stream-aligned teams**

## **Enabling teams**

## **Platform teams**

## **Complicated sub-system teams**





**Search**

Principal  
Engineers

**Vehicle Detail**

**Orders**

**Infrastructure**

# X-as-a-service

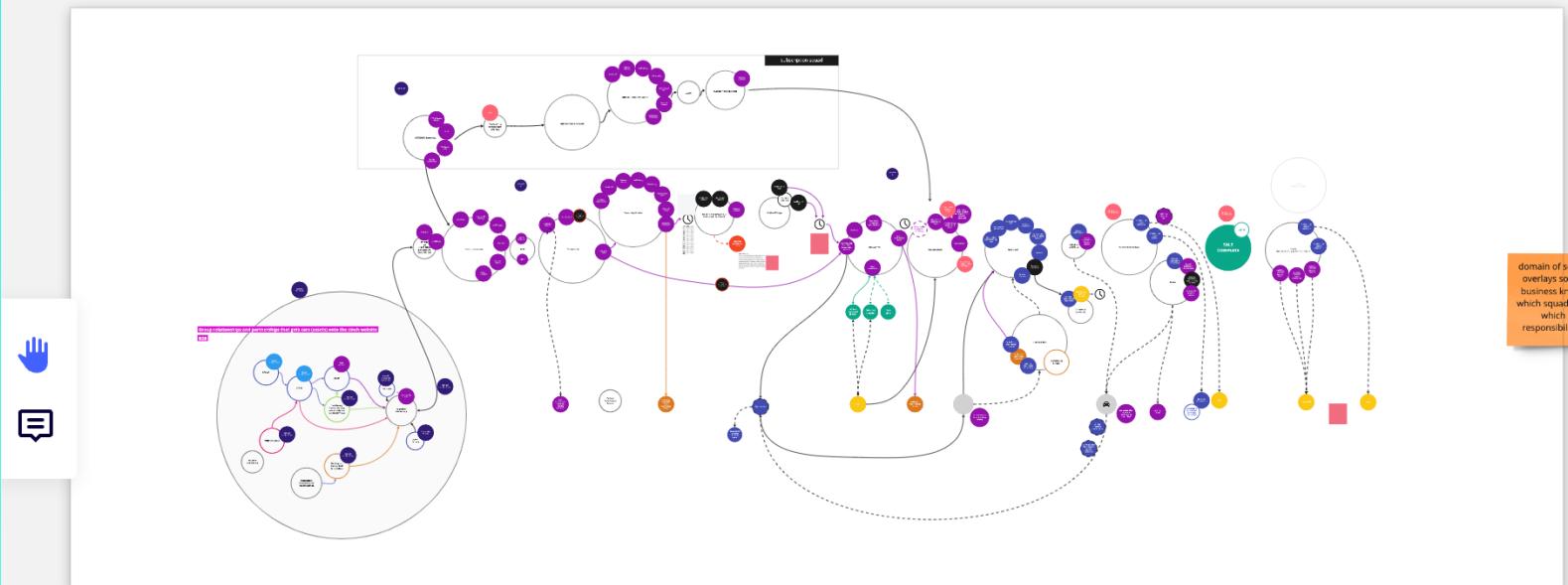
## Facilitating

## Collaboration

# service overview

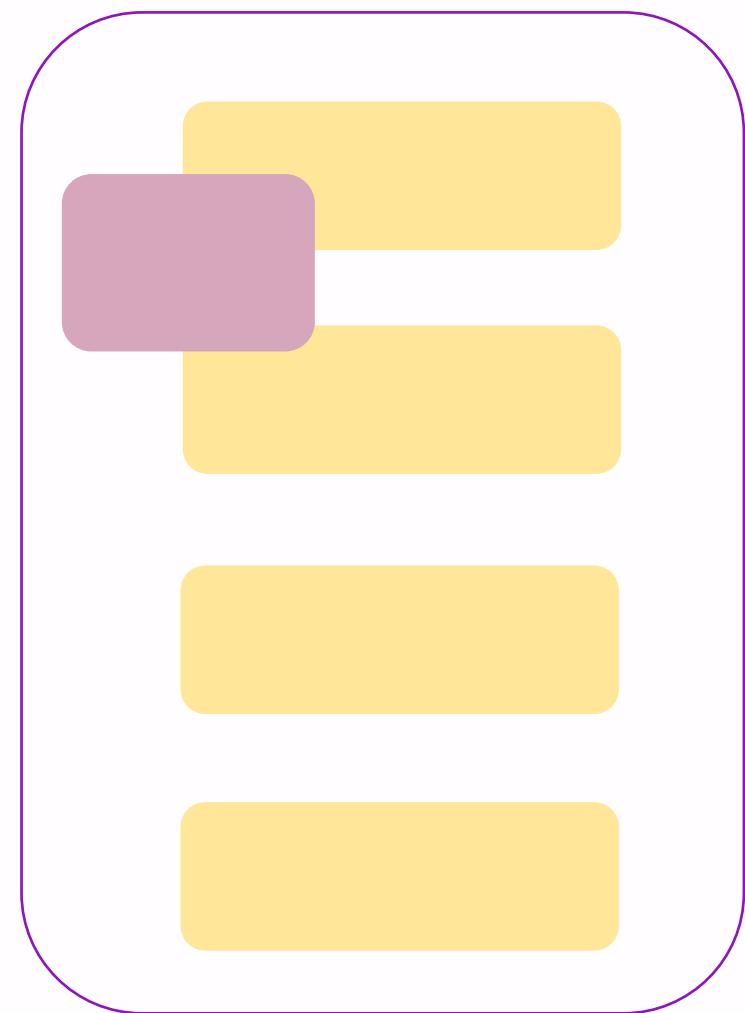
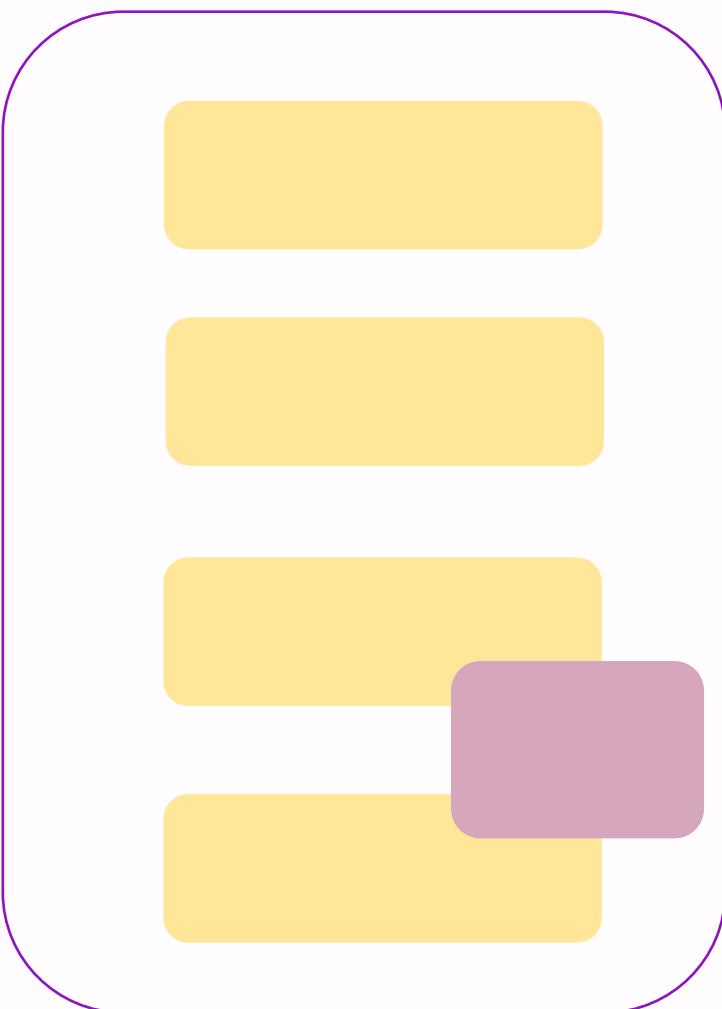
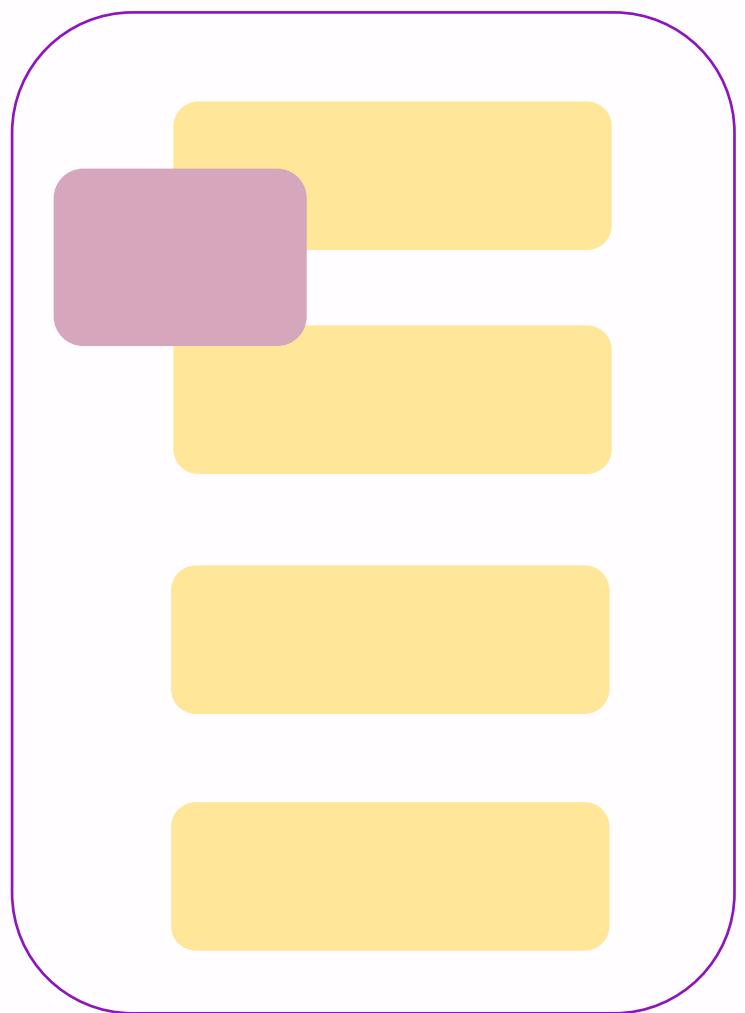
## 10,000ft view Service Blueprint (WIP)

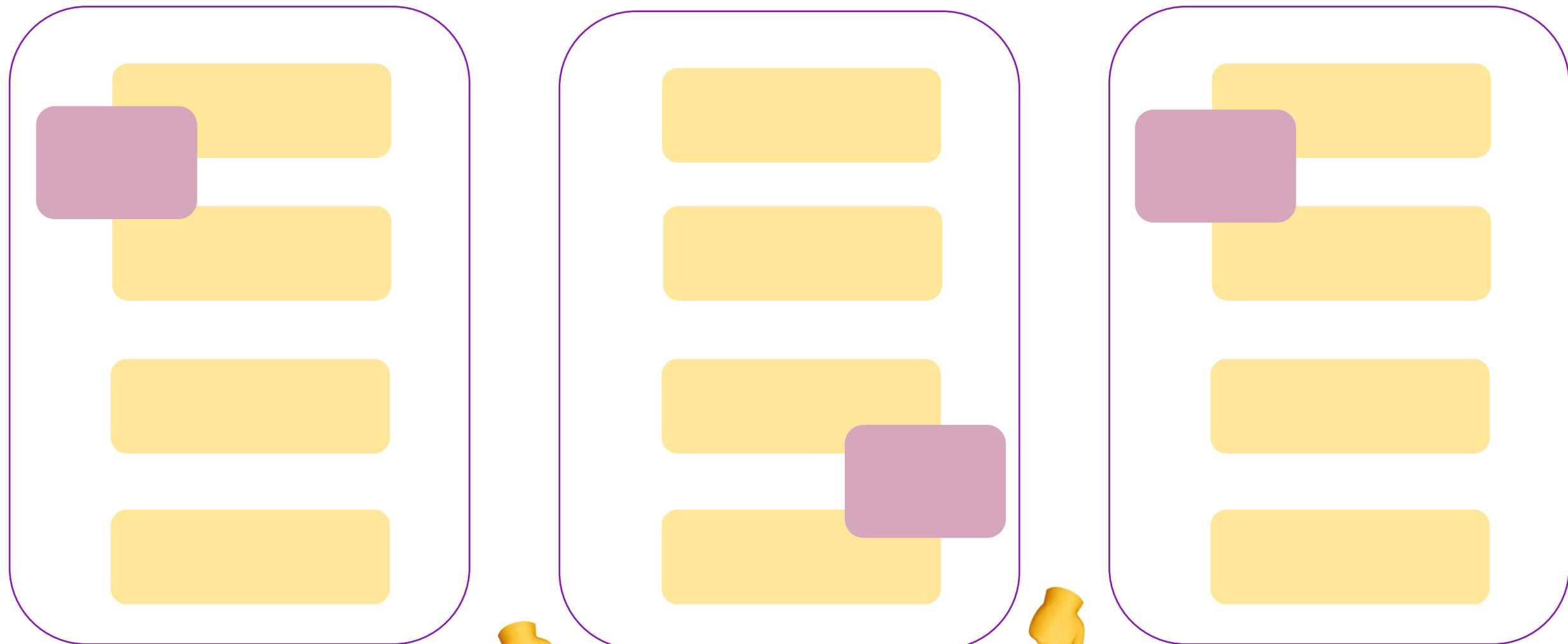
High level end to end

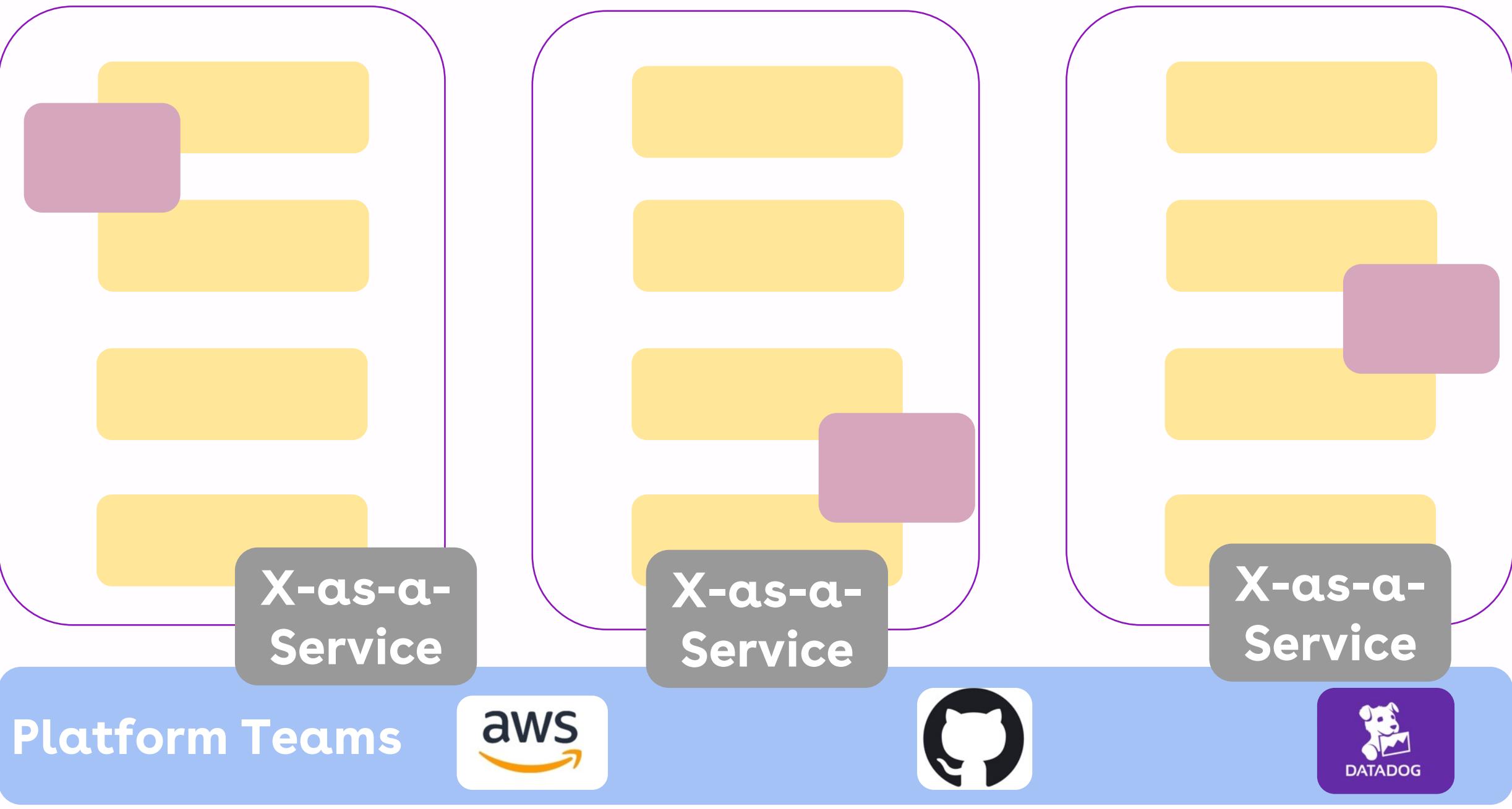


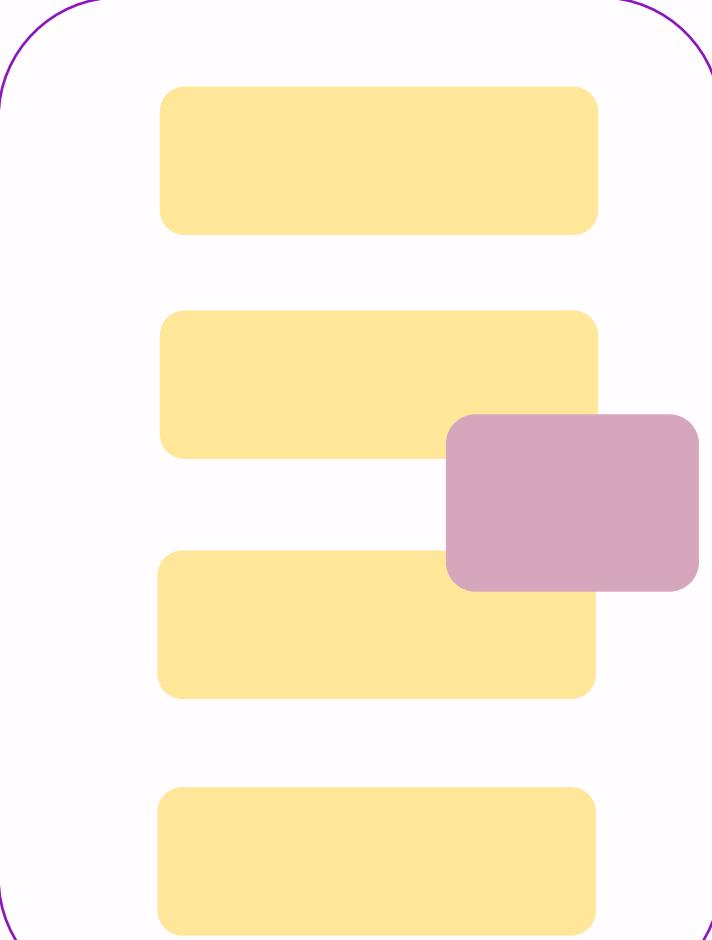
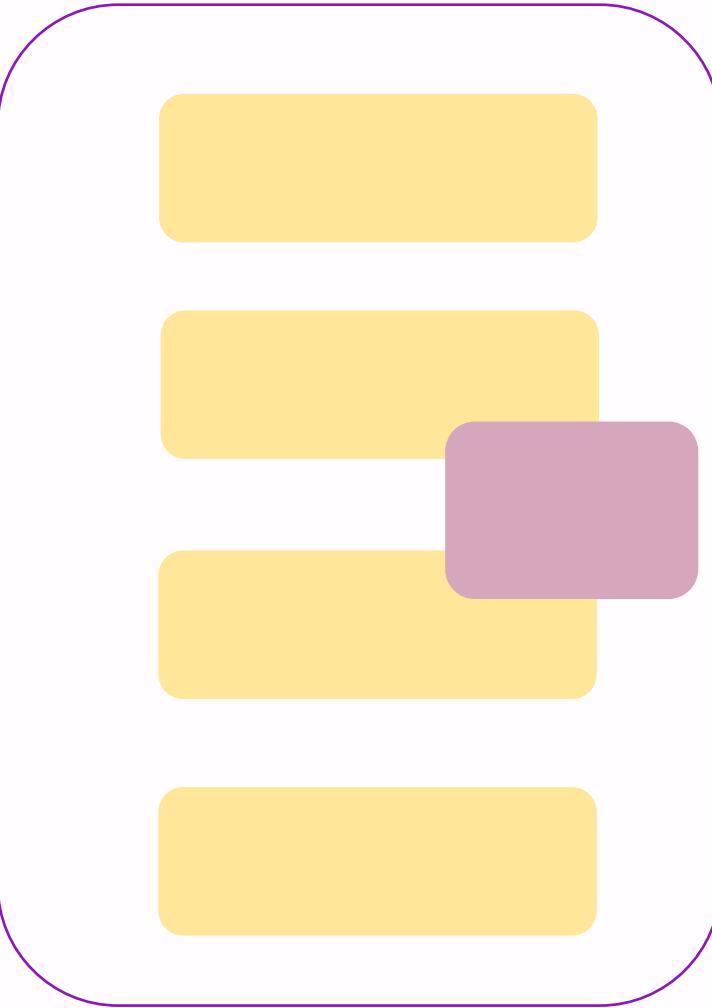
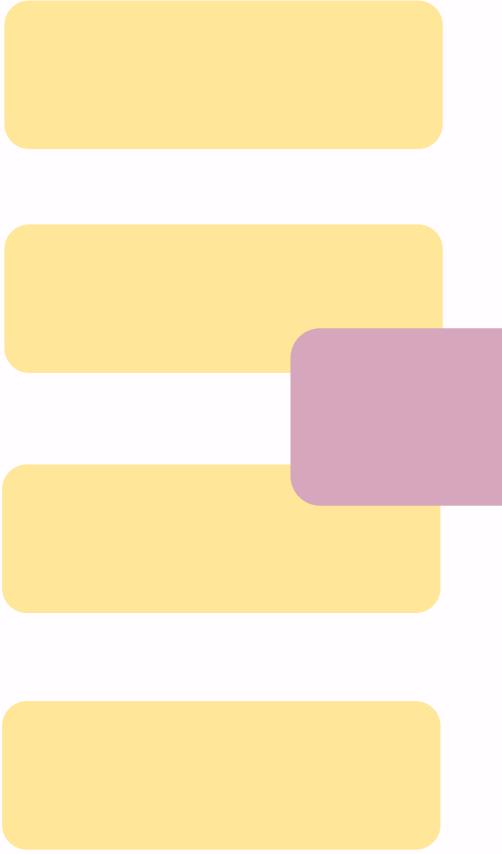
take a closer look below

X-as-a-Service







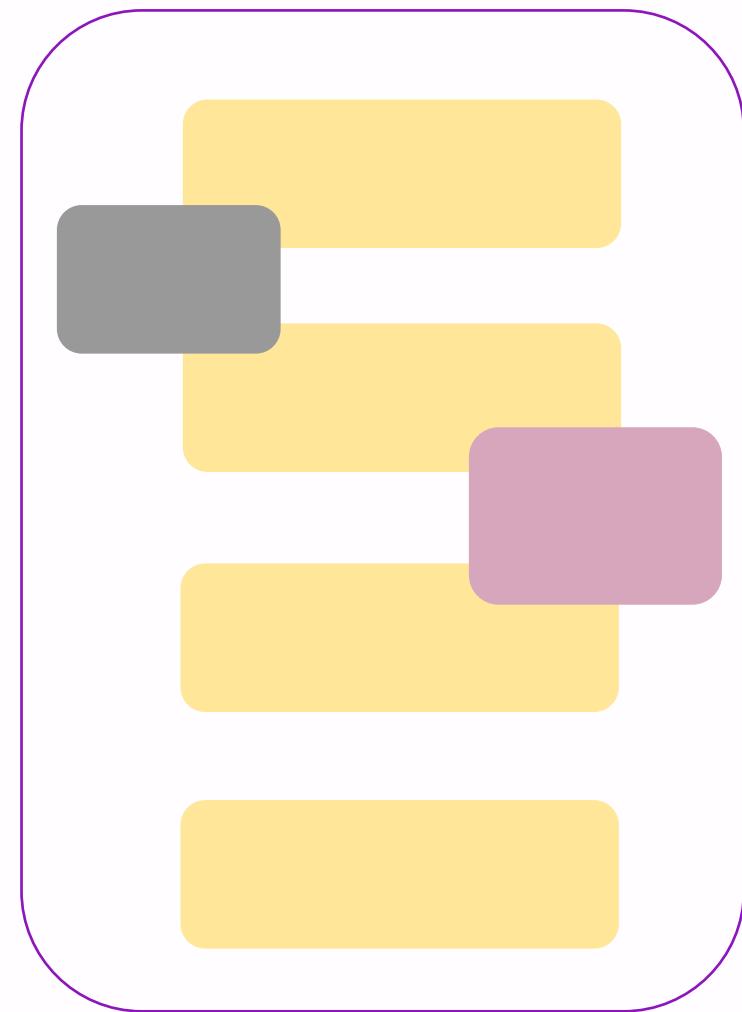
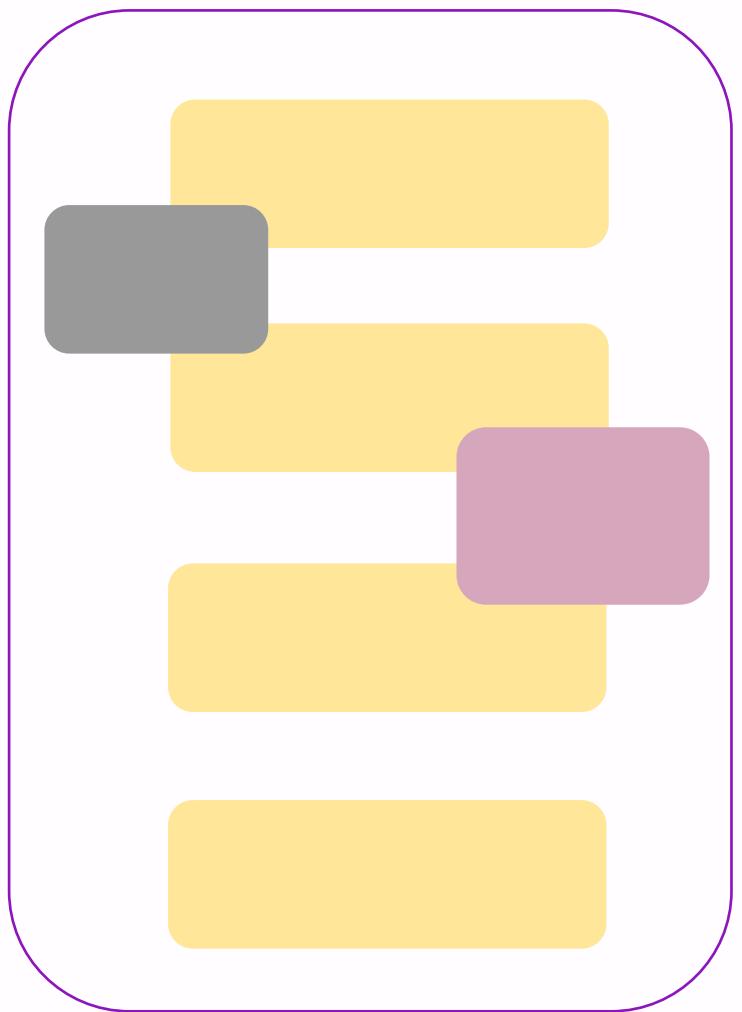
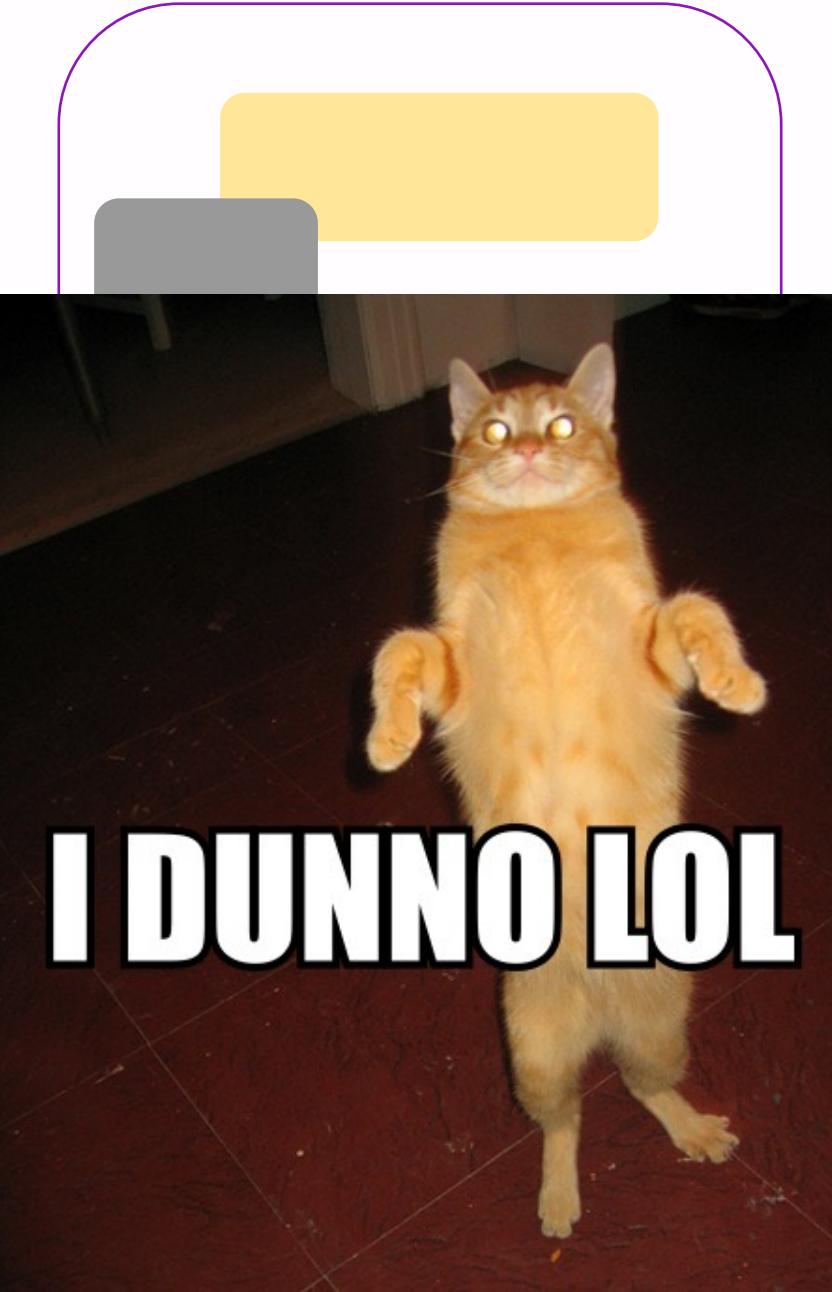


## Working Groups

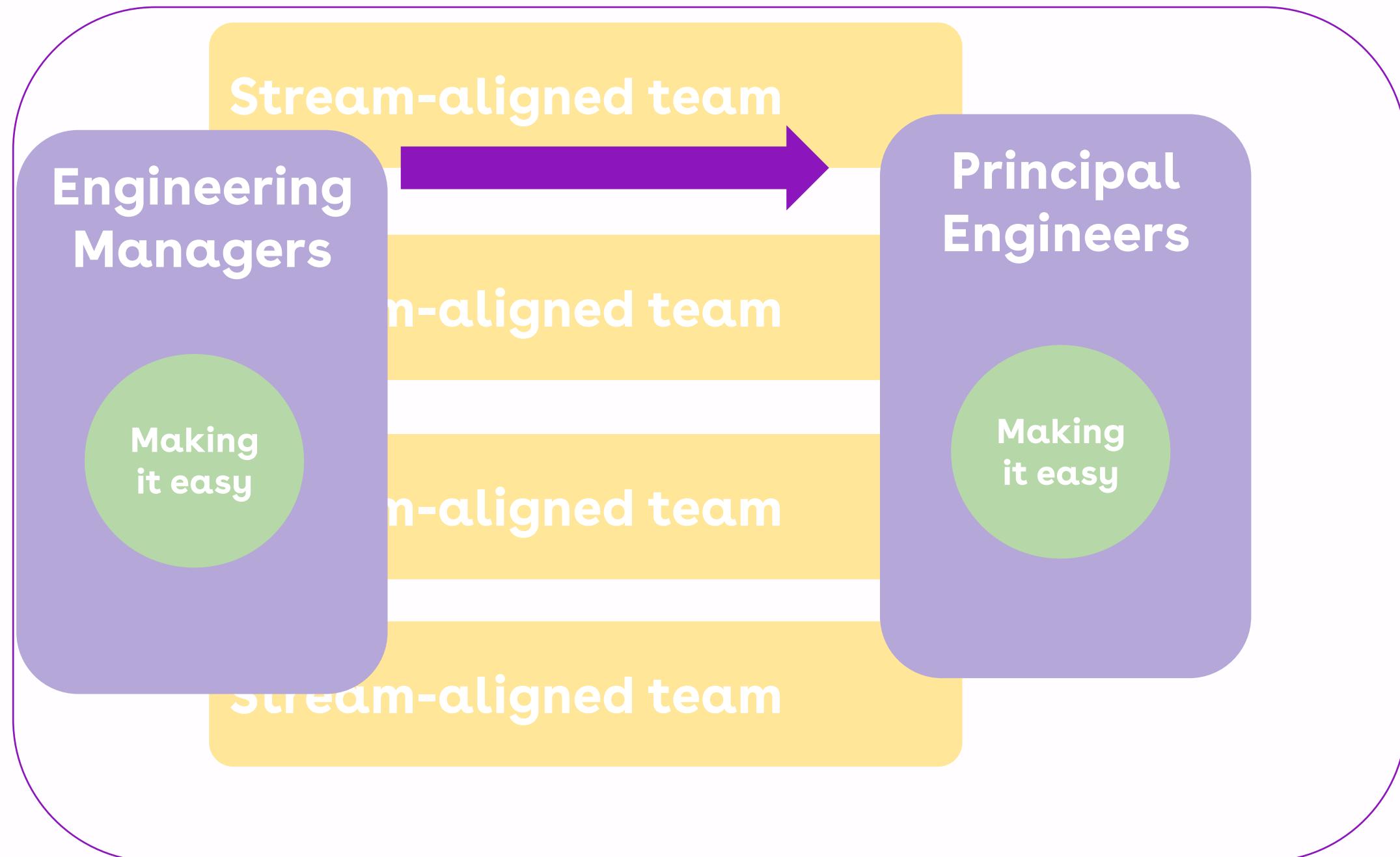
Make it  
easy

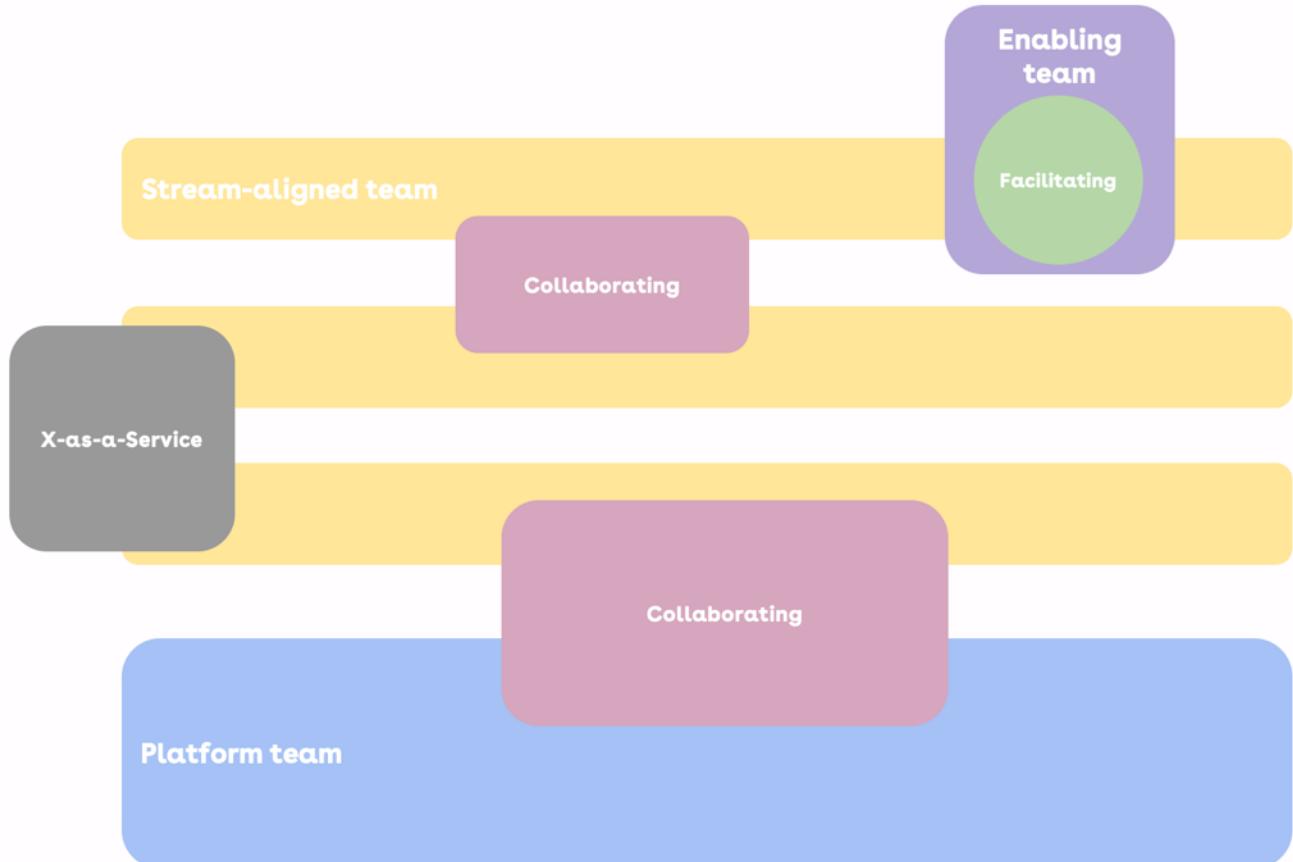
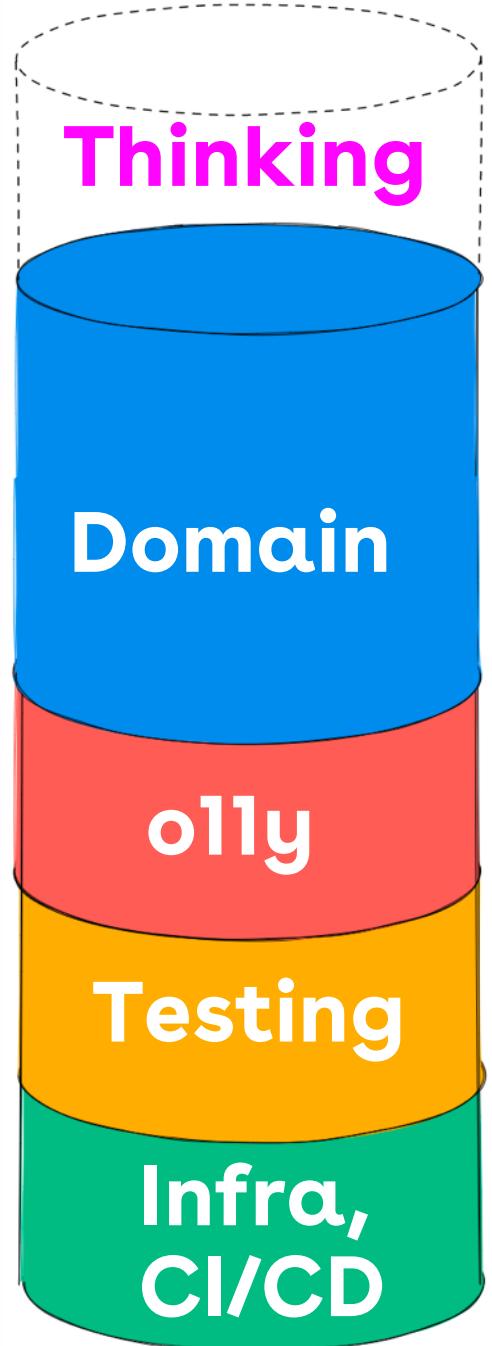
## Working Groups

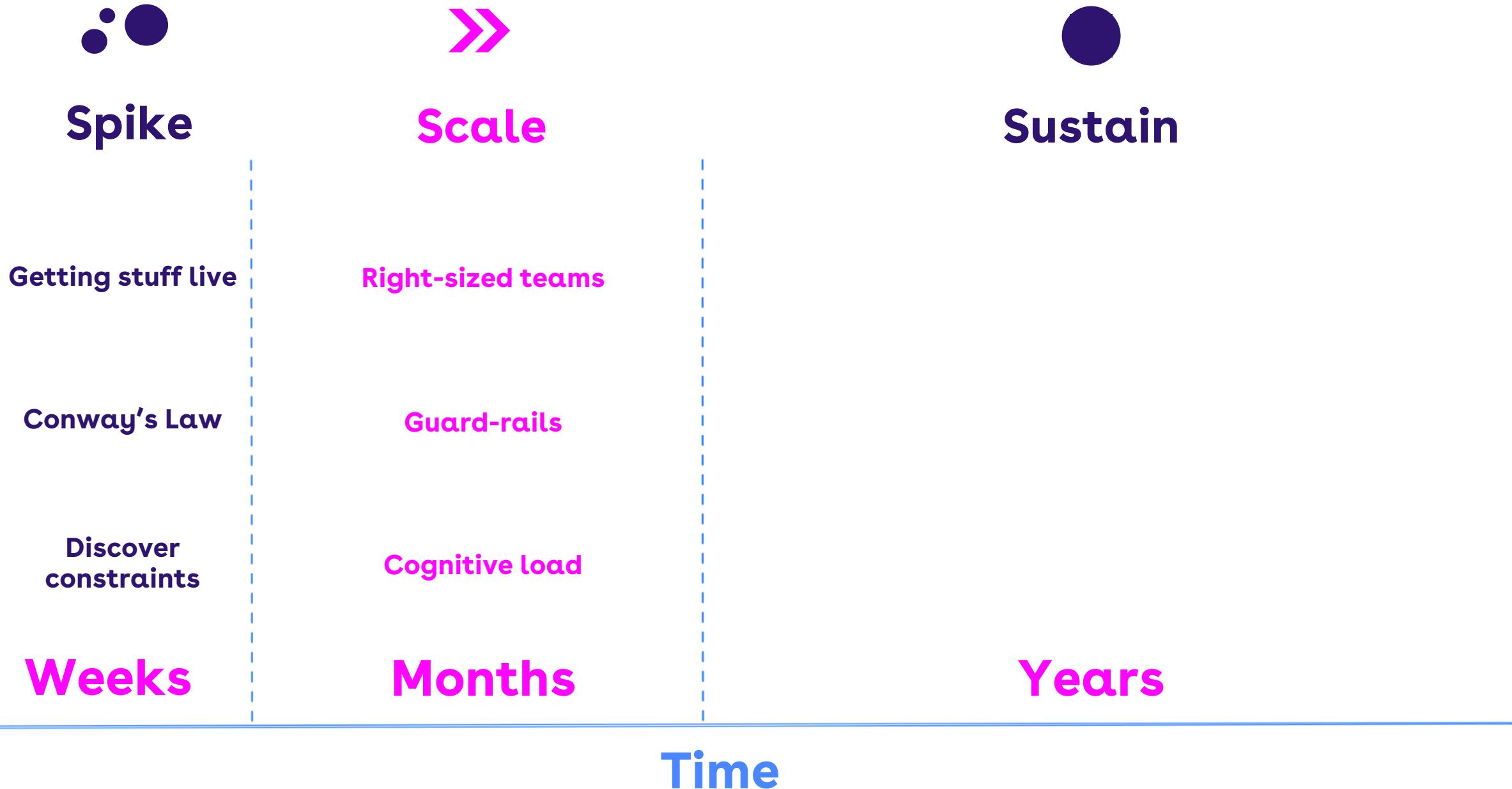
Make it  
easy



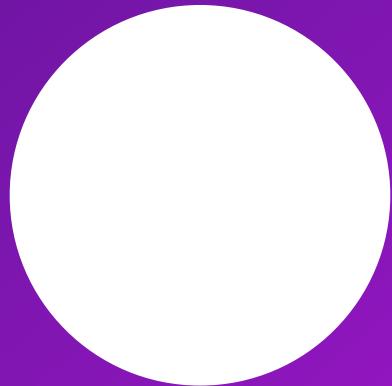








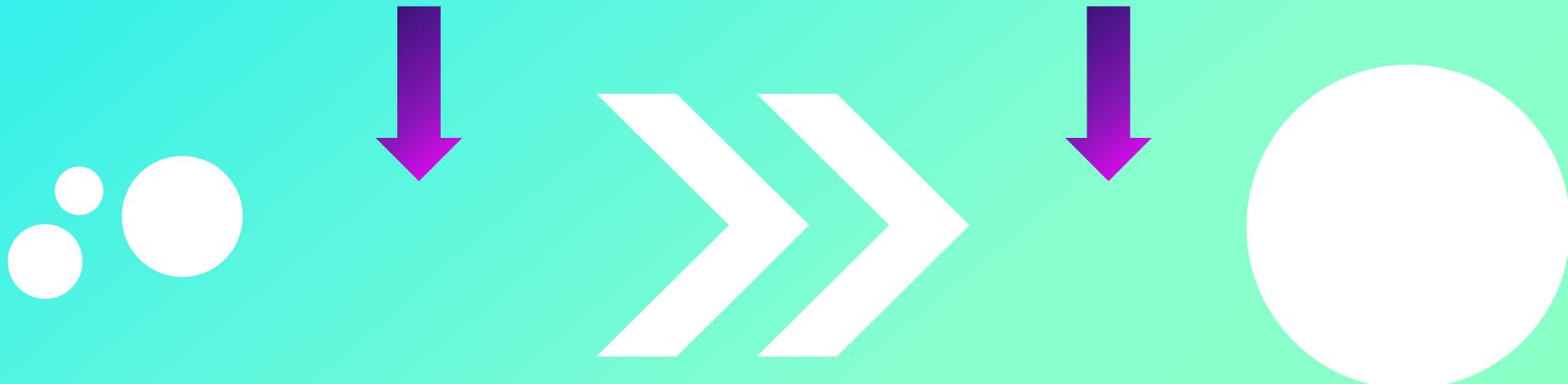
# Sustain.



To sustain means to  
support.

**What are you  
optimising for?**

# Focus on the gaps





## Spike

Getting stuff live

Conway's Law

Discover constraints

## Scale

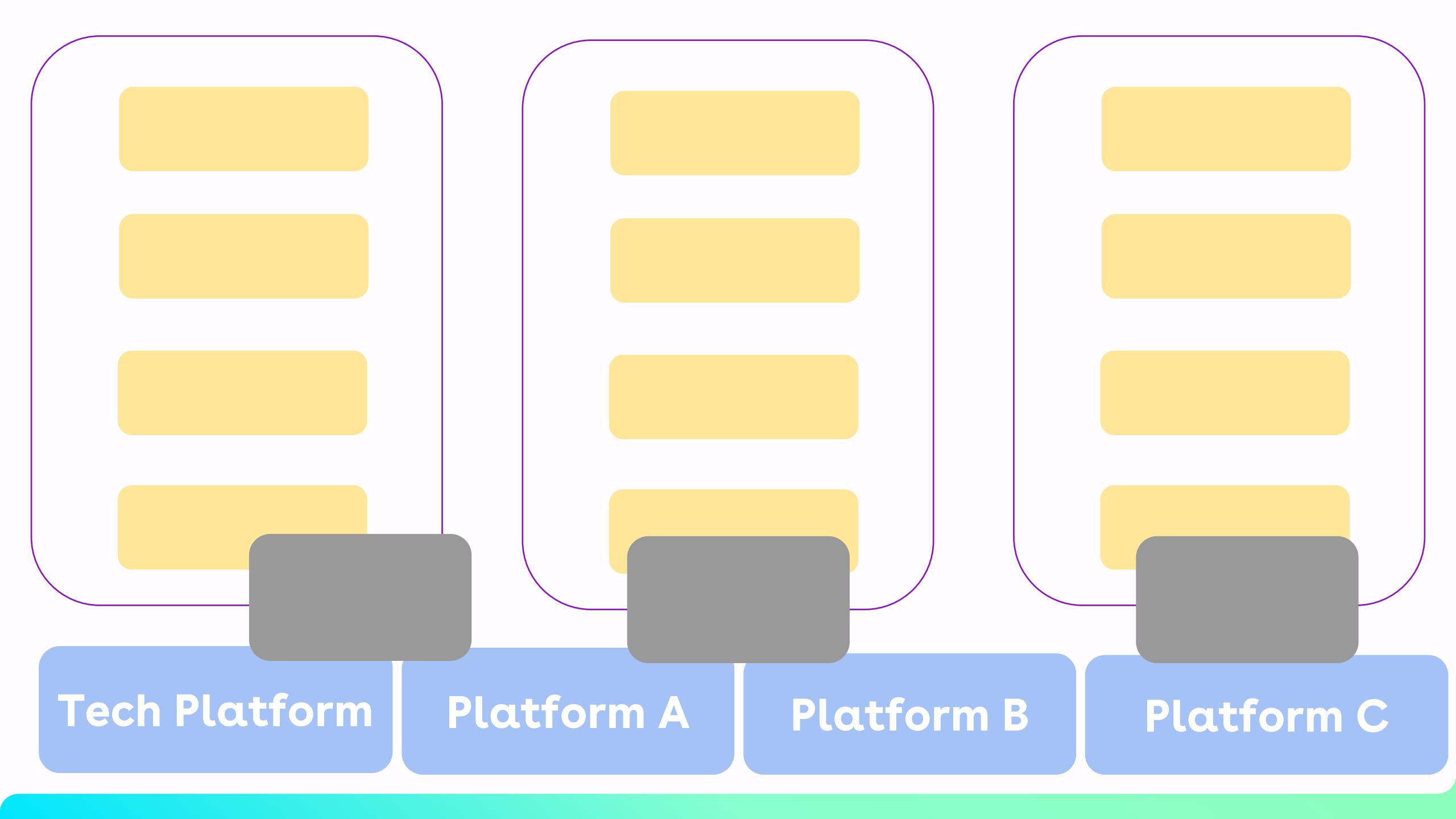
Right-sized teams

Guard-rails

Cognitive load

## Sustain

Better ways of working



# Platform flavours

Infrastructure

Design System

Orders

Vehicle Data

# Becoming knowledge stewards

**Practice** → **How we get better at what we do**

**Experience** → **Learning from those who've been there**

**People** → **Cross-organisation collaboration**

**Domain** → **Understanding the bigger picture**

**Technology principles help to define what good looks like, pointing teams in the direction of practices and ways of working we want to support or actively discourage.**

# Technology Blueprint

The image displays two side-by-side screenshots of the Cinch Technology Blueprint website, showing different sections of the platform.

**Left Screenshot: Observability principles**

- Header:** cinch Technology Blueprint, Guidance, Patterns, Examples, Technology radar.
- Sidebar:** Guidance, Principles, Technology, Organisational, **Observability** (selected), CI/CD, Architecture, Decision records, Cloud, Event Driven Architecture, Patterns, Examples, Technology radar.
- Section:**

## Observability principles

The core goal of cinch is to operate a car buying platform that takes the faff out of finding, buying, and owning a used car. Below are some guiding principles and practices for good observability.

### Observability at cinch

Change in software is a constant. A software system that doesn't change will inevitably suffer. At cinch, we embrace change but we also strive to build software with observability at its core.

### Why do we need observability?

At cinch, we build event-driven distributed software systems. This does provide us with resilient software. It also introduces complexity. Understanding the what, when and why something happens in the system is hard. We provide value to customers as they use our software systems. So we would like to answer the question:

  - How are our software doing when exercised by real users?

We can extrapolate a mental model of what *may* happen by reading the codebase. This is rarely efficient or even enough. This is where observable software systems come in handy. With observability, we can understand how our software is working. And we do this by evaluating its outputs.

### What do observable systems provide?

Insights into the workings of our software are in the form of telemetry data. We can build a mental model that maps what is happening within the software. With this we can understand both the *what* is going on (and *when*) but crucially, the *why*. With observable systems we are able to answer the following questions by querying the telemetry outputs of our systems:

  - What is going wrong?
  - Why is it going wrong?
  - How well is the system performing against our business objectives?

### How do we achieve observability at cinch?

At cinch, we use [Datadog](#) as our observability platform. The vehicle to more observable systems is to have good telemetry data. Coupled with good squad engineering practices and ceremonies.

For more practical observability guidance please refer to [The Observability Journey](#) on Confluence

**Right Screenshot: Consumer Contract Tests for Http Requests**

- Header:** cinch Technology Blueprint, Guidance, Patterns, Examples, Technology radar.
- Sidebar:** Guidance, Principles, Architecture, Cloud, Event Driven Architecture, Patterns, Examples, **Blueprints** (selected), Customer API authentication, Optimise for awareness, Evolve meaningful monitors and dashboards, Glossary, Contract testing, Consumer HTTP test (selected), Provider HTTP test, Consumer event test, Provider event test, Templates, Backend component, Technology radar.
- Section:**

## Consumer Contract Tests for Http Requests

### Scenario

A backend-for-frontend component, serving a mobile app, makes a POST request to the cinch-brew-order-component to create an Order.

### Acceptance Criteria

GIVEN the Customer has selected what they want to drink  
WHEN a valid request is made  
THEN the Order is created

### Example Code

```
// imports and mocks have been omitted
const orderClient = async (url: string) => {
  return axios.get(`${url}/orders`, { headers: { Accept: 'application/json' }, validateStatus: false });
};

const provider = new Pact({
  consumer: 'cinch-brew-app-bff-component', // the consuming participant
  provider: 'cinch-brew-order-component', // the providing participant
  port: 8091,
  pactfileWriteMode: 'merge',
  log: path.resolve(process.cwd(), 'logs', 'pact.log'),
  dir: path.resolve(process.cwd(), 'pacts'), // where to write the pact file to
  logLevel: 'debug',
});

beforeAll(async () => await provider.setup());

afterAll(() => provider.finalize());

 afterEach(() => provider.verify());

describe('Orders HTTP API Consumer', () => {
  it('can create an Order', async () => {
    const interaction = {
      state: undefined, // corresponds to a provider's stateHandler; defining null here, uponReceiving: 'a request to create an Order', // a description of the test
      uponReceiving: 'a request to create an Order',
      given: 'the customer wants to buy a beer',
      when: 'the provider receives the request',
      then: 'the provider creates an order'
    };
    provider.addInteraction(interaction);
  });
});
```

# Exemplars

github.com

README.md

## cinch-brew

Cinch brew is a fictional software system demonstrating good observability. The system is designed with common cinch architectural patterns in mind - including backend and frontend.

### An observability blueprint

An observability blueprint serves as the educational center for good observability patterns and practices used at cinch. Its purpose is to offer guidance by way of working examples of common scenarios you will come across whilst developing components.

### What this is not

- A set of rules you must follow
- A resource to simply copy & paste
- A guide for engineering best practices
- Datadog documentation

### Observability Tooling

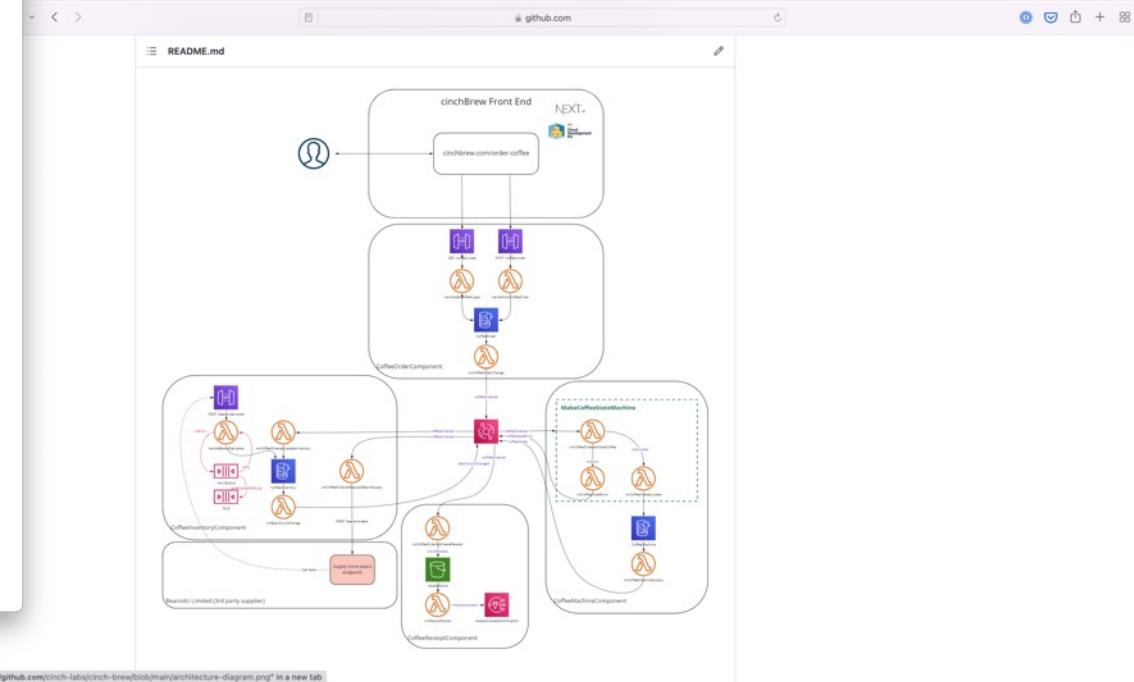
- Frontend
- Backend
- CI and Tests

### The cinch brew architecture

The cinch brew architecture is based around common architectural patterns used by many components at cinch. It utilizes technologies and AWS services you may already be familiar with.

The system is made up of multiple components using both synchronous and asynchronous methods of communication. It also features communication with a 3rd party system, a state machine, document storage and queues, all common scenarios you may come across.

cinchBrew Front End







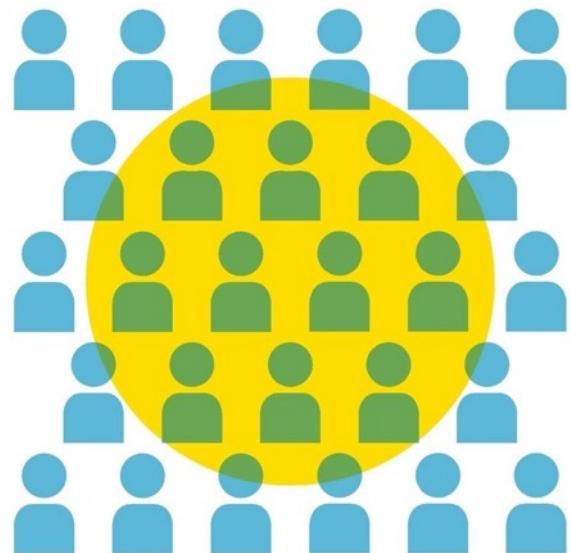
Communities are  
**powerful things**

**“Groups of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly”**

- Wenger-Trayner

## BUILDING SUCCESSFUL COMMUNITIES OF PRACTICE

Discover how connecting people makes better organisations



Emily Webber



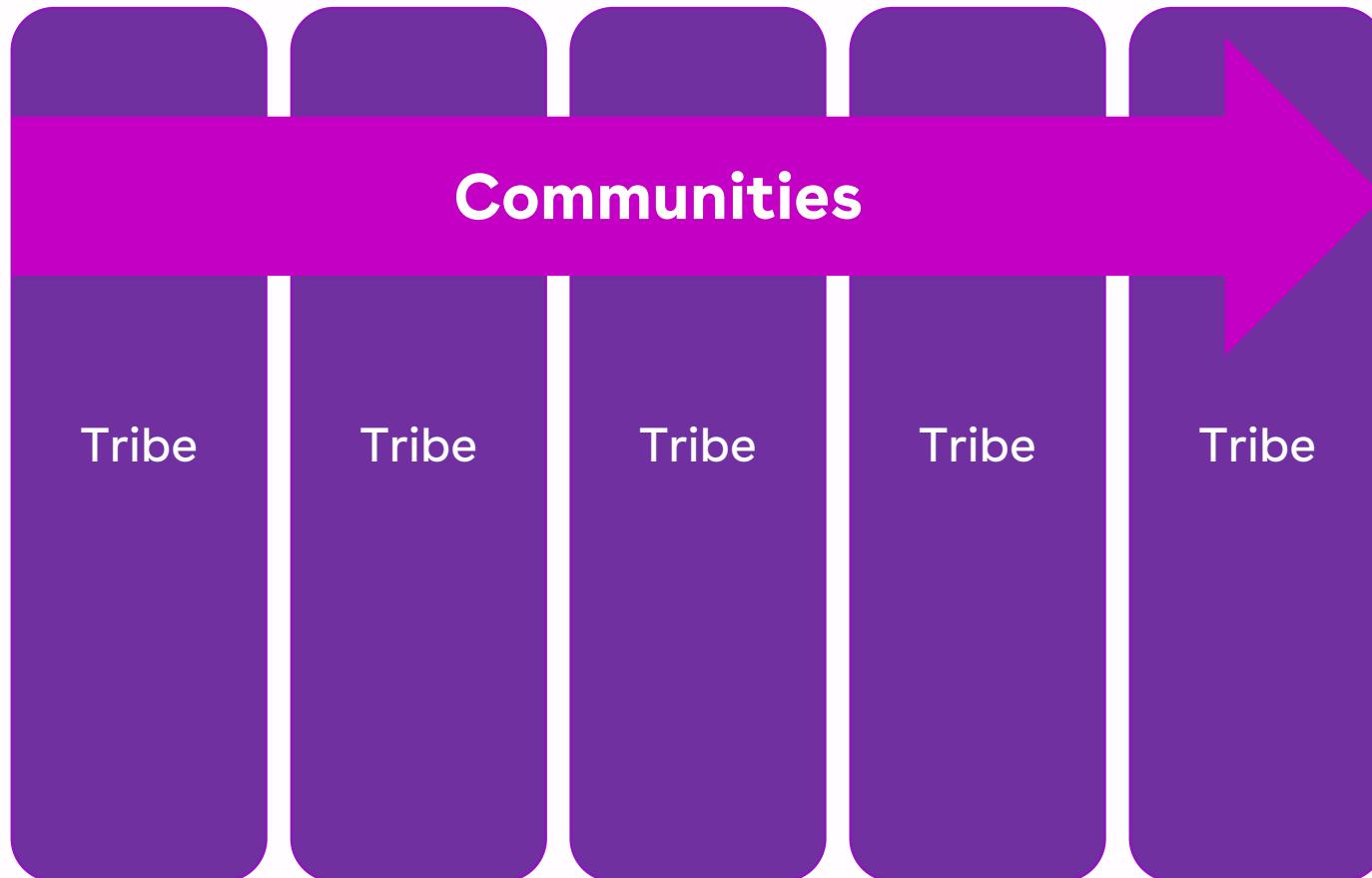
A GUIDE TO MANAGING KNOWLEDGE

# CULTIVATING COMMUNITIES OF PRACTICE

ETIENNE WENGER  
RICHARD McDERMOTT  
WILLIAM SNYDER



HARVARD BUSINESS SCHOOL PRESS





Communities

Silos

**Meg's Mumbles - My Start At Cinch**

In General Topics 0 min 12 46 19 0 Follow

I thought I'd add some insight to starting from Square 1 and my first week at cinch.

But first, an introduction to me.

I'm Megan. Or Meg. Or MJ. I don't really mind. I began my journey with cinch last week (21/03/2022) after graduating from a physics degree and then a first job which unfortunately was not a good fit - it happens. My specialities in my physics degree delved into astro and space physics. From a young age I have been really intrigued by the workings of outside of Earth and so decided a physics degree was the thing for me. And it was amazing. I fell further in love with the planets that spin alongside us and how things came to be. But I fell in love with something new too. Coding.

Nasa Image of Jupiter with its polar aurora visible

Through lab experiments and analysing the data received and moving into my dissertation on Jupiter's ionosphere (which I will hopefully go into depths of if anyone is interested), I found a passion for making things work through a computer and so came to a crossroads in my choices after completing my Bachelors degree. Continue being an astrophysicist or become a software engineer. I think you can predict which I

**Autonomy of Schedule**

0 2 min 4 34 Edit 9 0 Follow

I have what you would call in [Team-Topologies-speak](#), a number of **streams**. Squads are called **squads**. Squads are the unit of software delivery at cinch. As we have established other team types:

- Groups
- Communities (of Practice or Interest)
- Cubs (☞ The DevOps Handbook Book Club)

**not why**

It about the purpose of all team types, but the 'how' of working outside of your squad. If you belong to a squad or a tribe, that is your core membership. You have made a promise to the people in your squad or tribe, and they take priority over other tribes.

It means that you can't participate in things outside the squad. In fact, we encourage you to do so, but it is vital that you:

- Things visible
- Communicate with your core membership - your squad

Katie Raby  
Software Engineer Apr 4, 2022 (12:56)

**axe-con 2022 Digital Accessibility Conference**

In Events 0 8 min 0 22 6 0 Follow

**What is axe-con?**

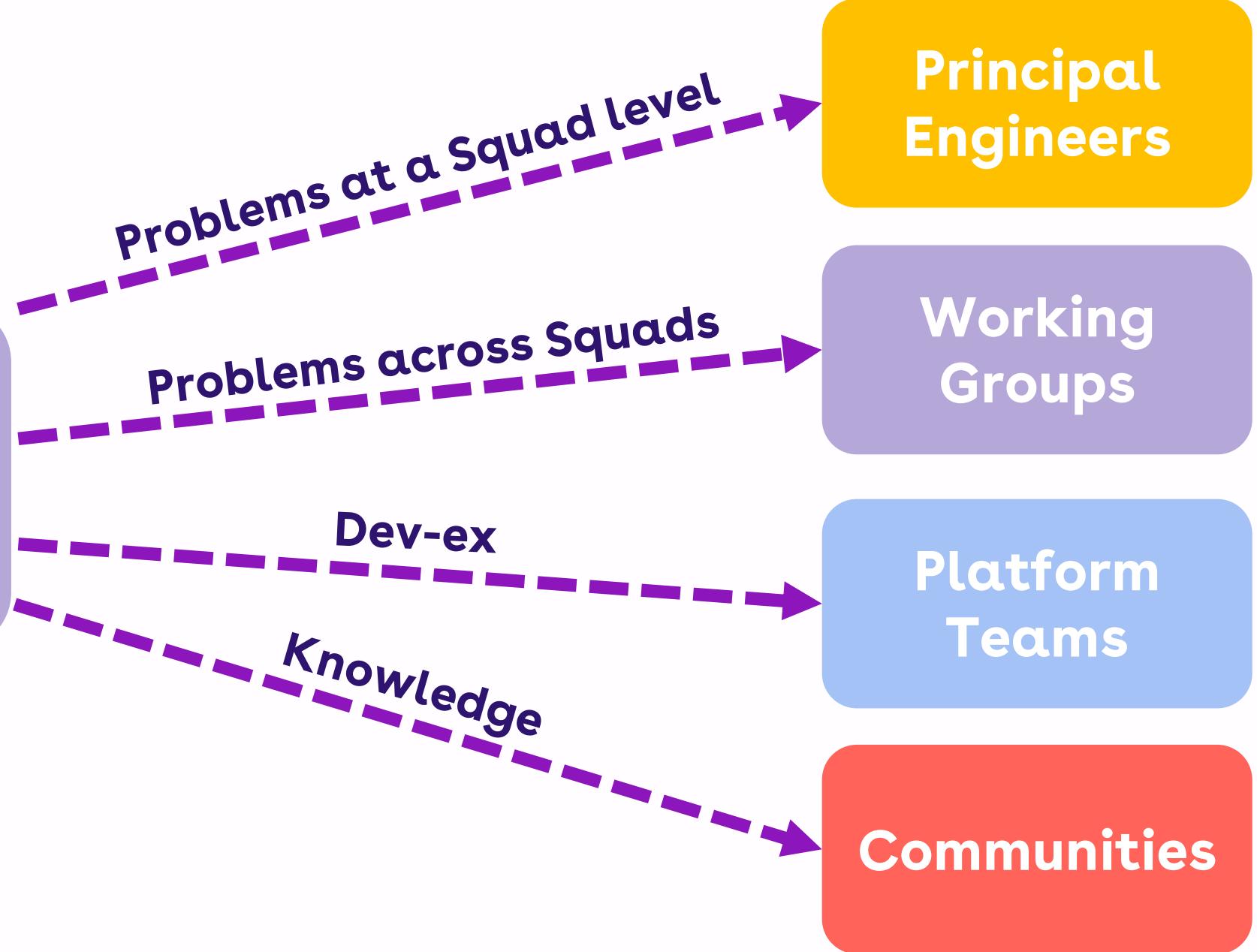
axe-con is a free, open and inclusive digital accessibility (a11y for short) conference held online. The conference focuses around building, testing, and maintaining accessible experiences. axe-con is suitable for developers, designers, business users, and accessibility professionals of all experience levels.

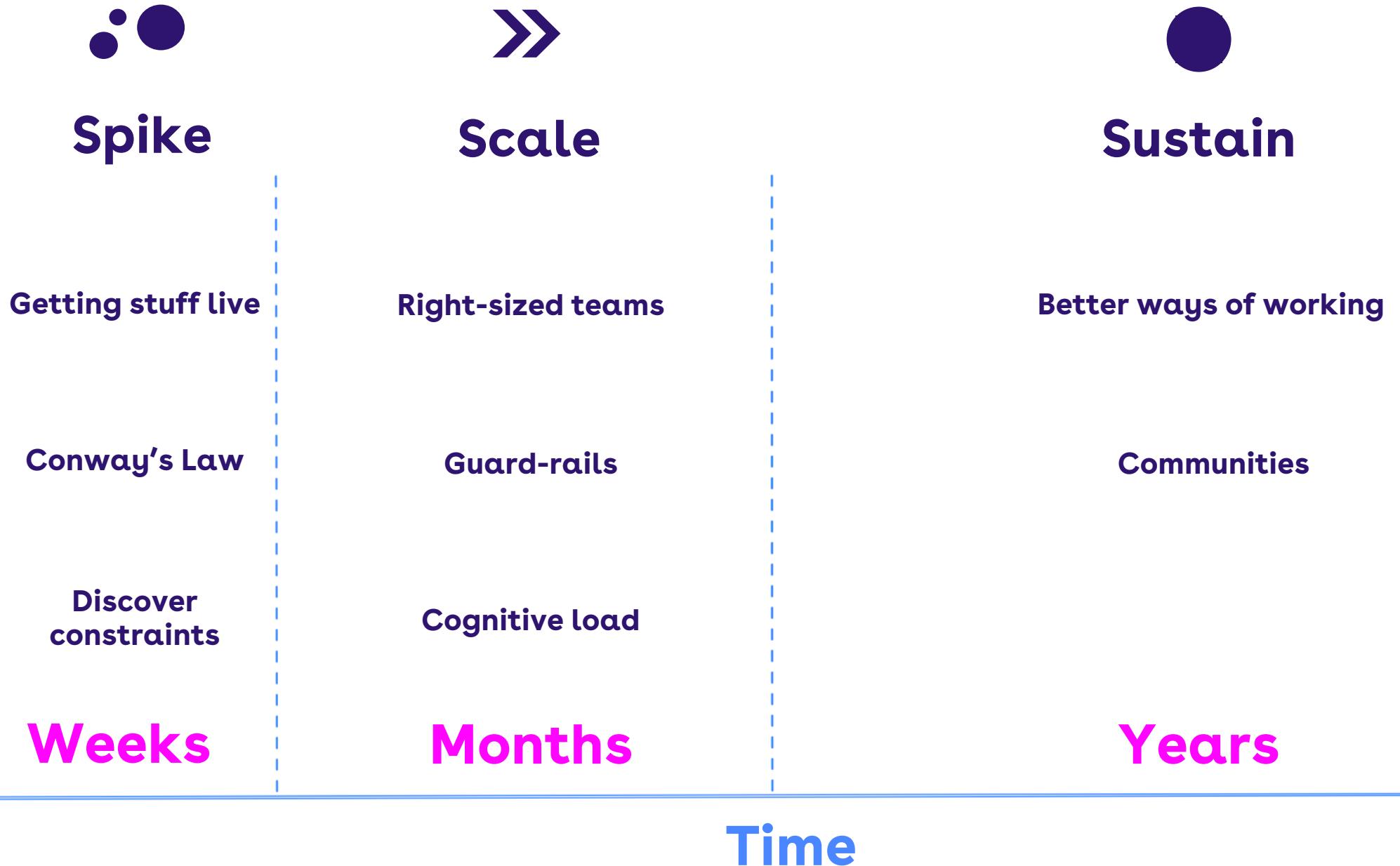
We live-streamed the axe-con sessions in the Manchester office (with lots of snacks - shout out to the sour cream & onion Pringles 😊). We connected with attendees in a11y at cinch, and had fun debating the topics we were watching in.

**Takeaways from a few talks we watched**

remove semantic meaning

Style Sheets (CSS) is what's used to give websites life and to make them look better, to quote a phrase from a legendary Marvel movie:







## Spike

Getting stuff live

Conway's Law

Discover constraints

**Weeks**

## Scale

Right-sized teams

Guard-rails

Cognitive load

**Months**

## Sustain

Better ways of working

Communities

Observability

**Years**

Time

**Observability is the ability to understand how a software system is behaving internally by evaluating its outputs.**

# Working on the (virtual) walls

The image displays two side-by-side screenshots of cloud monitoring dashboards, illustrating how different teams use monitoring tools for their operations.

**Orders Squad - Motherboard** (Left Dashboard):

- Are orders looking healthy?** (23 widgets)
- Is our part exchange service healthy?** (12 widgets)
- What changes have been released?** (6 widgets)
- Do we have any incidents?** (1 widget)
- Are we seeing errors?** (5 widgets)
- Are my APIs healthy?** (12 widgets)

**Search Squad Standup Board** (Right Dashboard):

- Search squads Links:** A grid of links to various tools:
  - Jira
  - Repos
  - Miro
  - Dashboards
  - Docs
  - AWS
  - APM - Search
  - APM - Rec
  - CloudTrail
  - RUM
  - Sonar
  - Logs
- Health at a Glance:** A summary section with the following data:
  - Monitor statuses:** Shows 35 monitors in an **OK** status.
  - Alerted monitors:** No matching entries found.
  - LCP average:** A chart showing LCP average over time.
  - Unique user crashes:** A table of unique user crash data.



## Spike

Getting stuff live

Conway's Law

Discover constraints

**Weeks**

## Scale

Right-sized teams

Guard-rails

Cognitive load

**Months**

## Sustain

Better ways of working

Communities

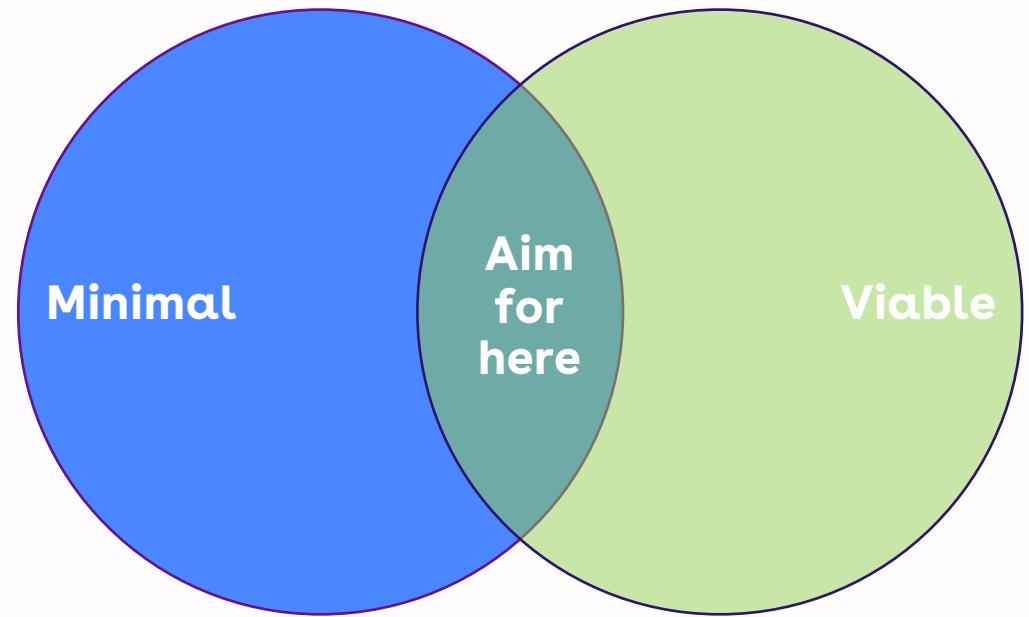
Observability

**Years**

Time

# In summary

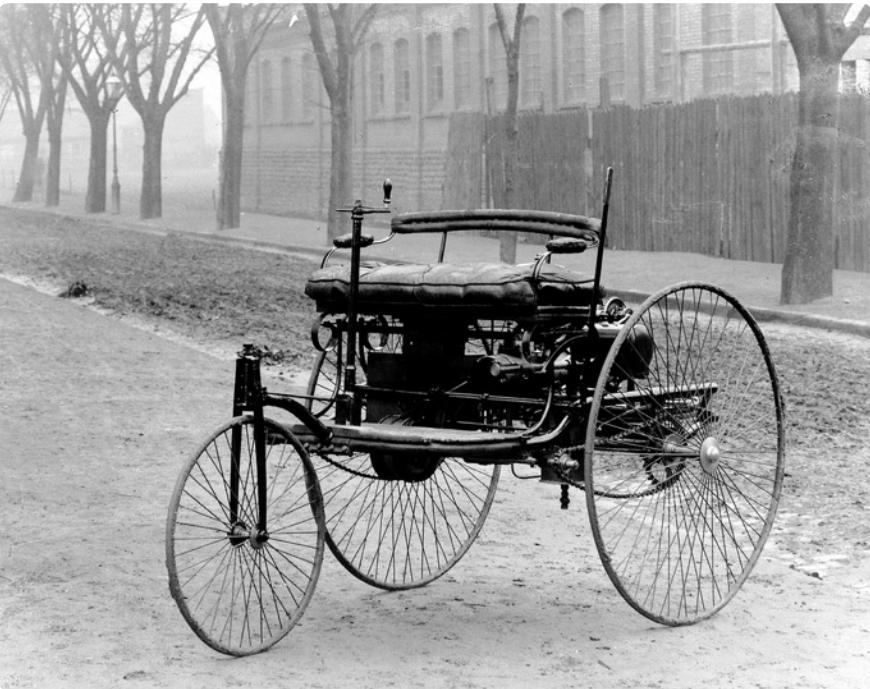
# Minimal viable everything.

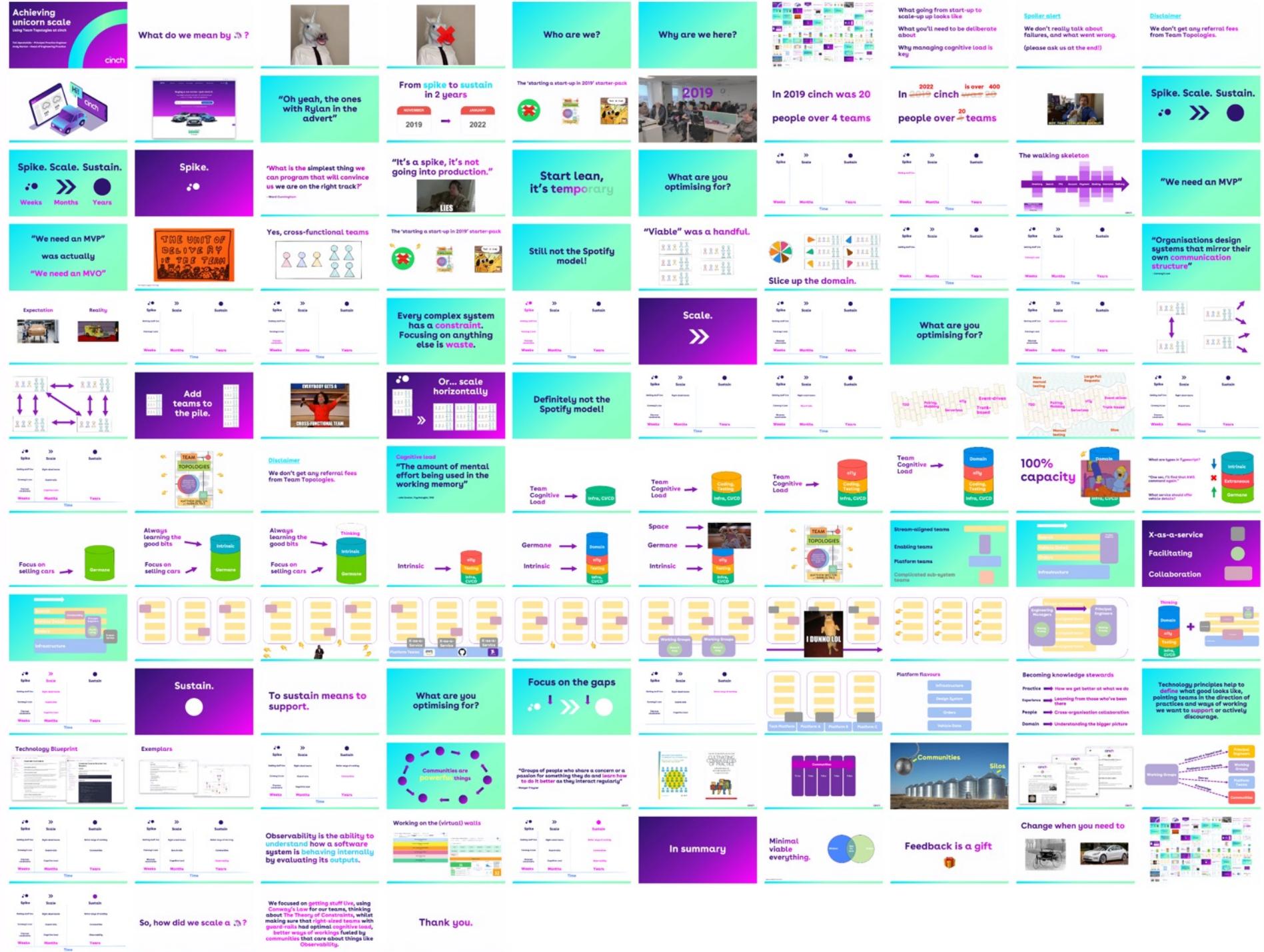


# Feedback is a gift



# Change when you need to







## Spike

Getting stuff live

Conway's Law

Discover constraints

**Weeks**

## Scale

Right-sized teams

Guard-rails

Cognitive load

**Months**

## Sustain

Better ways of working

Communities

Observability

**Years**

Time

**So, how did we scale a 🦄 ?**

We focused on getting stuff live, using Conway's Law for our teams, thinking about The Theory of Constraints, whilst making sure that right-sized teams with guard-rails had optimal cognitive load, better ways of workings fueled by communities that care about things like Observability.

**Thank you.**