

Escaping Unicorn Culture

Alex Cross
Group Head of DevOps
Endava



Alex Cross

GROUP HEAD OF DEVOPS, ENDAVA

-
-
-
-
-
-

Alex Cross is Group Head of DevOps at Endava. Over the last decade he has helped some of the biggest names in multiple industries adopt a DevOps approach to IT, and today he directs Endava's own DevOps-oriented delivery capability. Under the hood he's a keen cloud architect and tinkerer, and there's a Philosophy PhD in there somewhere too.

Alex.Cross@Endava.com



Agenda

1. FANTASTIC ENGINEERS AND WHERE TO FIND THEM
2. BLOB-SHAPED PEOPLE
3. ROOTS OF UNICORN CULTURE
4. CURSE OF THE UNICORN
5. ESCAPING UNICORN CULTURE
6. Q&A

1

Fantastic Engineers and Where To Find Them



What's a Unicorn?



What's a Unicorn?

- ▶ Hyper-talented
- ▶ Interdisciplinary skillset
- ▶ Senior
- ▶ Loves a challenge
- ▶ Always super busy
- ▶ Never admits defeat
- ▶ Rare



Hunting Unicorns

- ▶ DevOps Engineer
- ▶ DevSecOps Engineer
- ▶ Cloud Engineer
- ▶ SRE
- ▶ Cloud Security Engineer
- ▶ DevOps Architect
- ▶ DevOps Manager

...but also

- ▶ Full Stack Developer
- ▶ Data Engineer

...and so on.





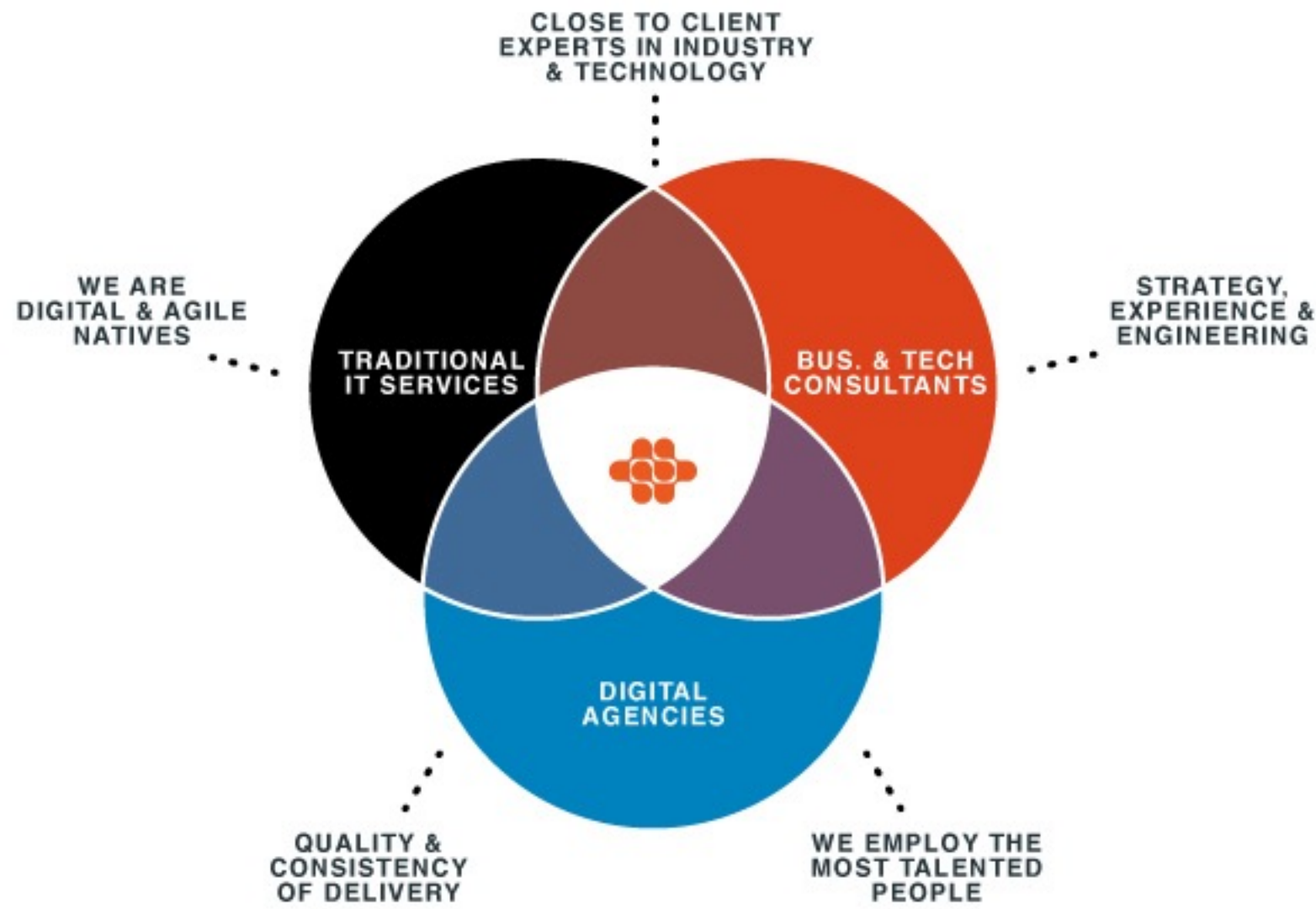
REIMAGINING THE RELATIONSHIP BETWEEN PEOPLE & TECHNOLOGY

We accelerate our clients' ability to take advantage of new business models and market opportunities by ideating and delivering dynamic platforms and intelligent digital experiences that fuel the rapid, ongoing transformation of their businesses.

By leveraging next-generation technologies, our agile, multi-disciplinary teams provide a combination of Product & Technology Strategies, Intelligent Experiences, and World Class Engineering to help our clients become more engaging, responsive, and efficient.



We Are A Pure Play Next-Gen Technology Company



We Are a Global Technology Company

WE WORK WITH SOME OF THE WORLD'S LEADING COMPANIES IN:

- TECHNOLOGY
- TELECOMMUNICATIONS
- MEDIA
- RETAIL
- HEALTHCARE
- MOBILITY
- CONSUMER GOODS

World Class Engineering:

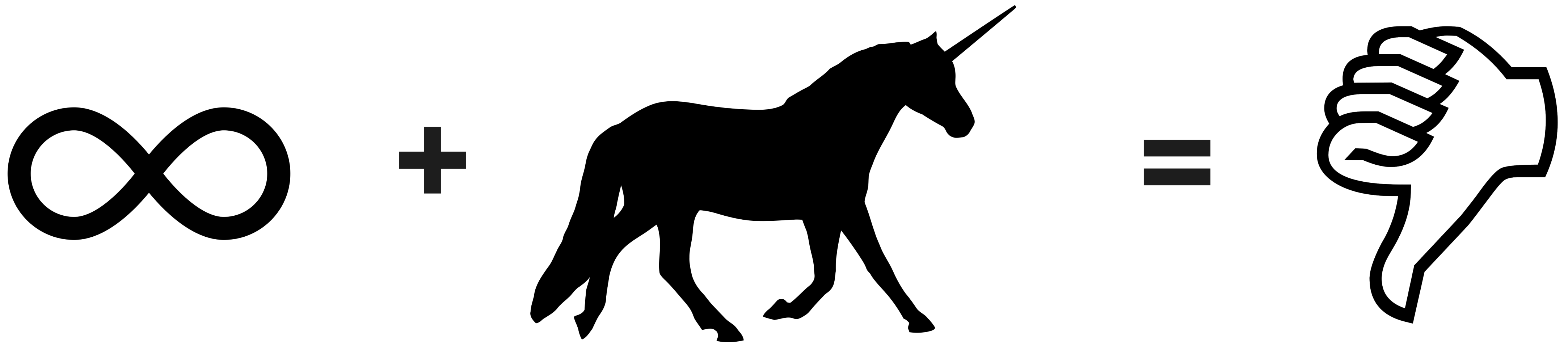
- Nearshore offices with top talent
- Challenge culture and strong values
- Agile at heart
- Focused on innovation
- Knowledge sharing mindset
- 1 in 3 new hires referred by existing staff

Don't build your DevOps capability on Unicorns.

- You're setting yourself up to fail
- Even if you can find them, you can't rely on them
- They are useful, but you don't need them

But then: why are organisations so easily drawn into unicorn hunts?

- It's a function of **culture**.



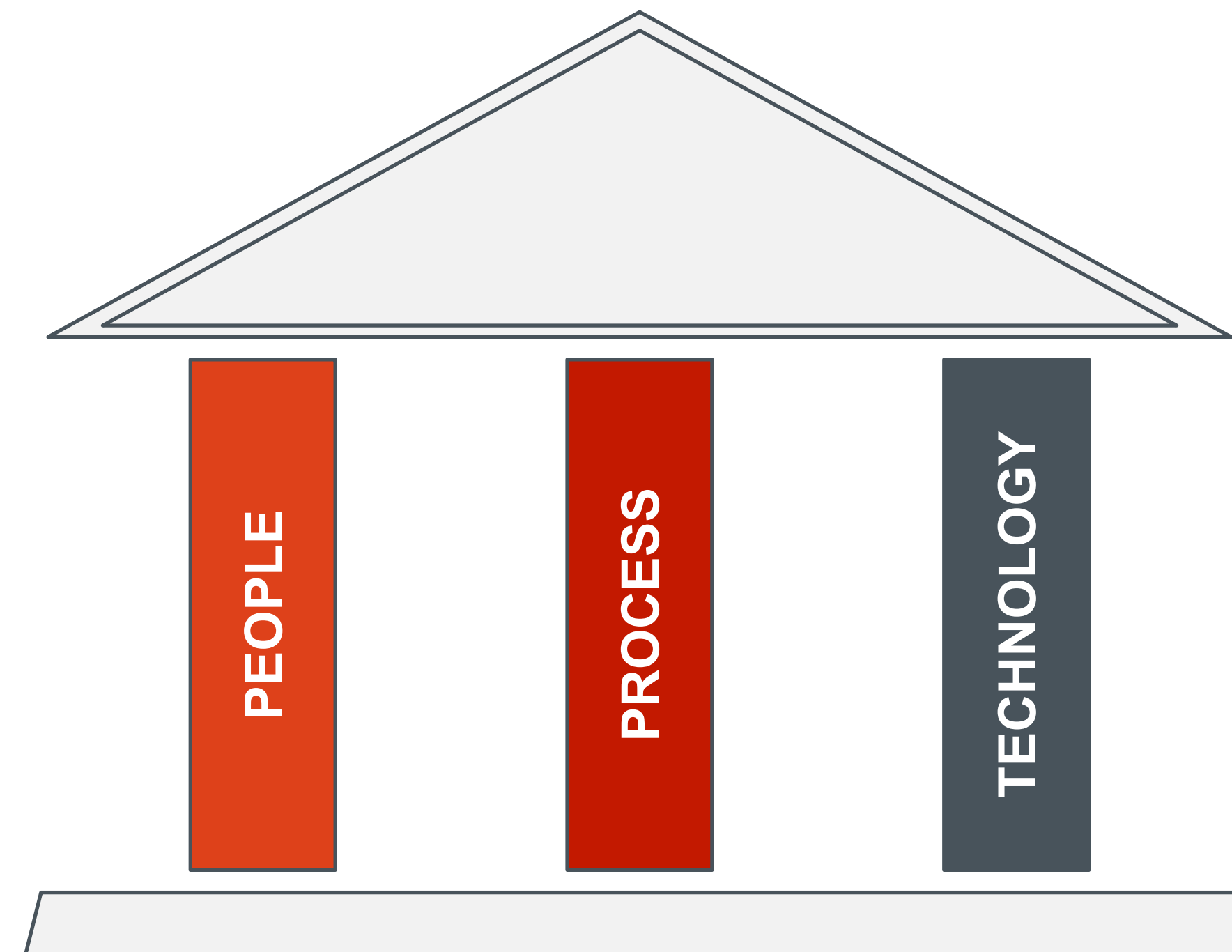
2

Blob-Shaped People

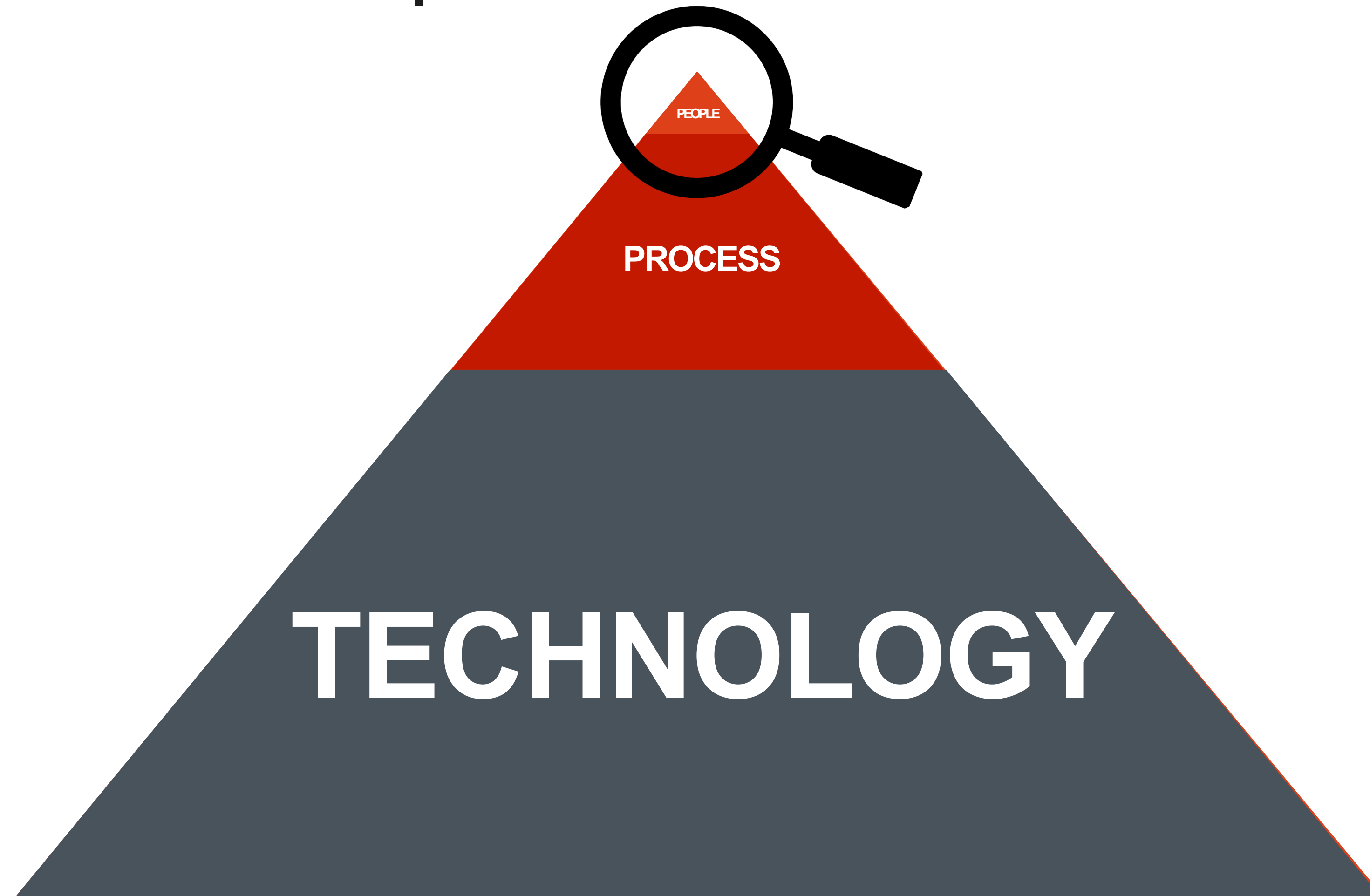


Operating Models

- DevOps & DevSecOps are names for classes of operating model
- The goal: maximise the flow of IT change without compromising on stability, security or operability.
- An operating model addresses various people, process, and technology concerns

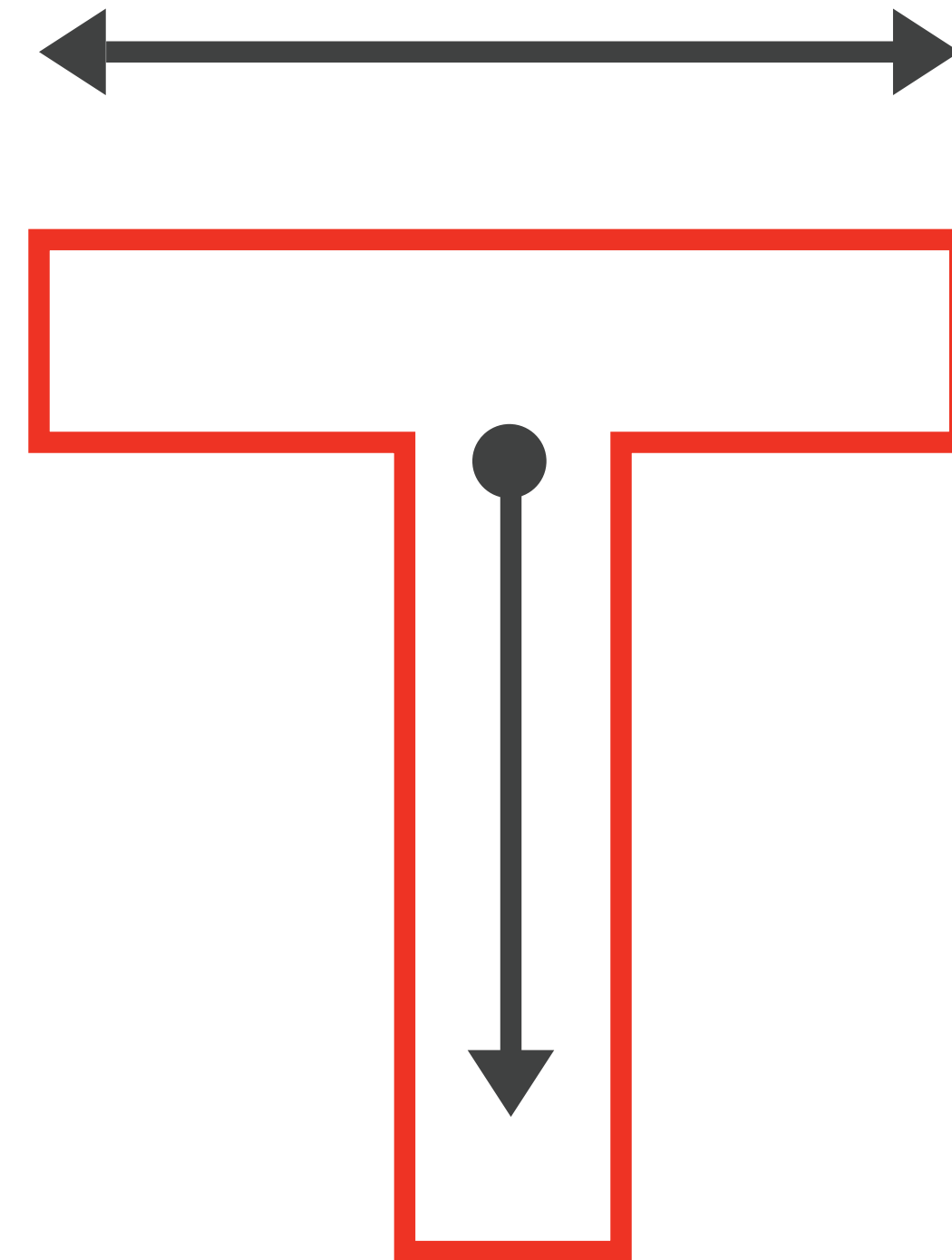


Operating Models in DevOps



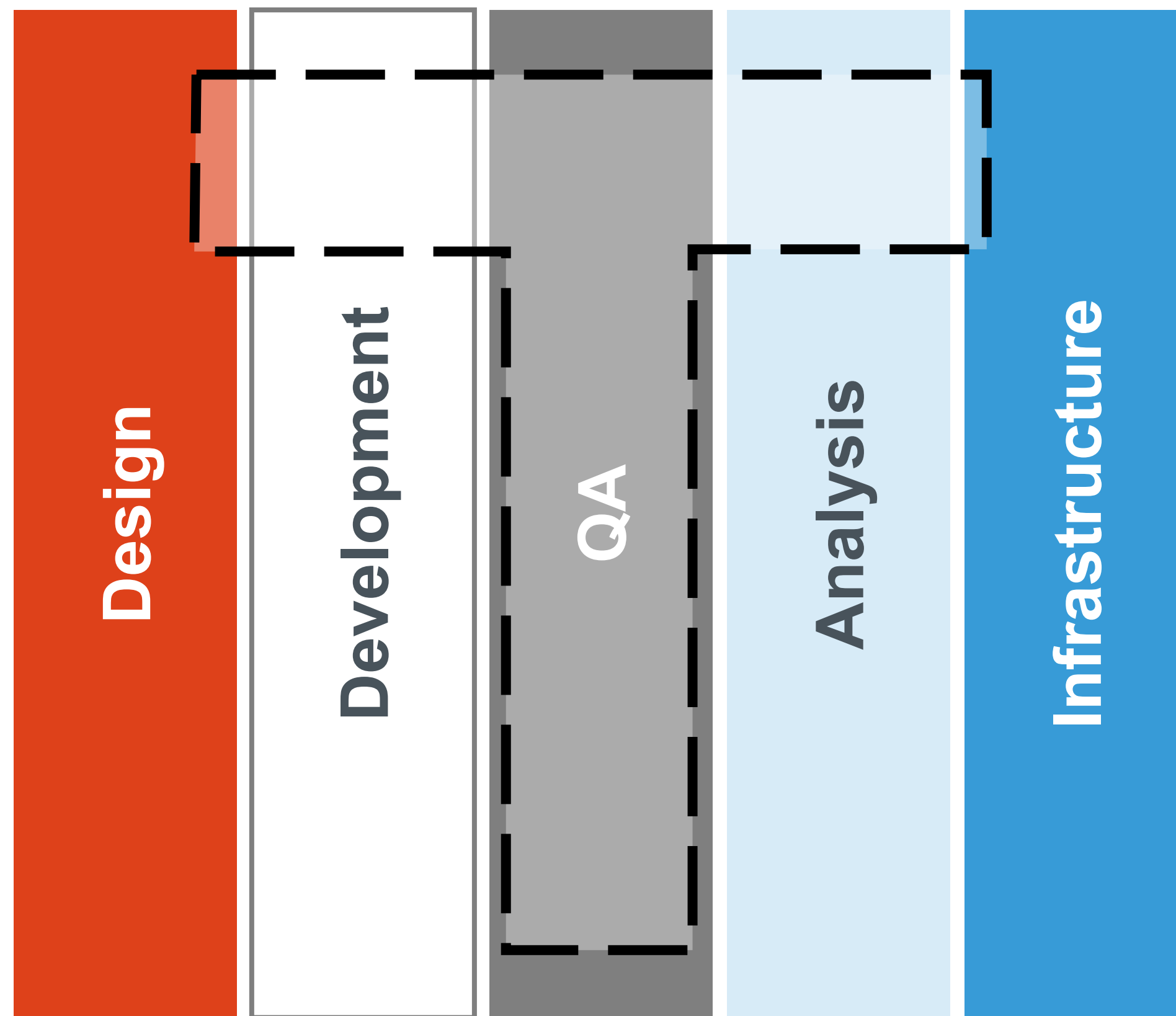
The T-Shaped Person

- One deep area of specialism
- Broader, shallower expertise in other areas

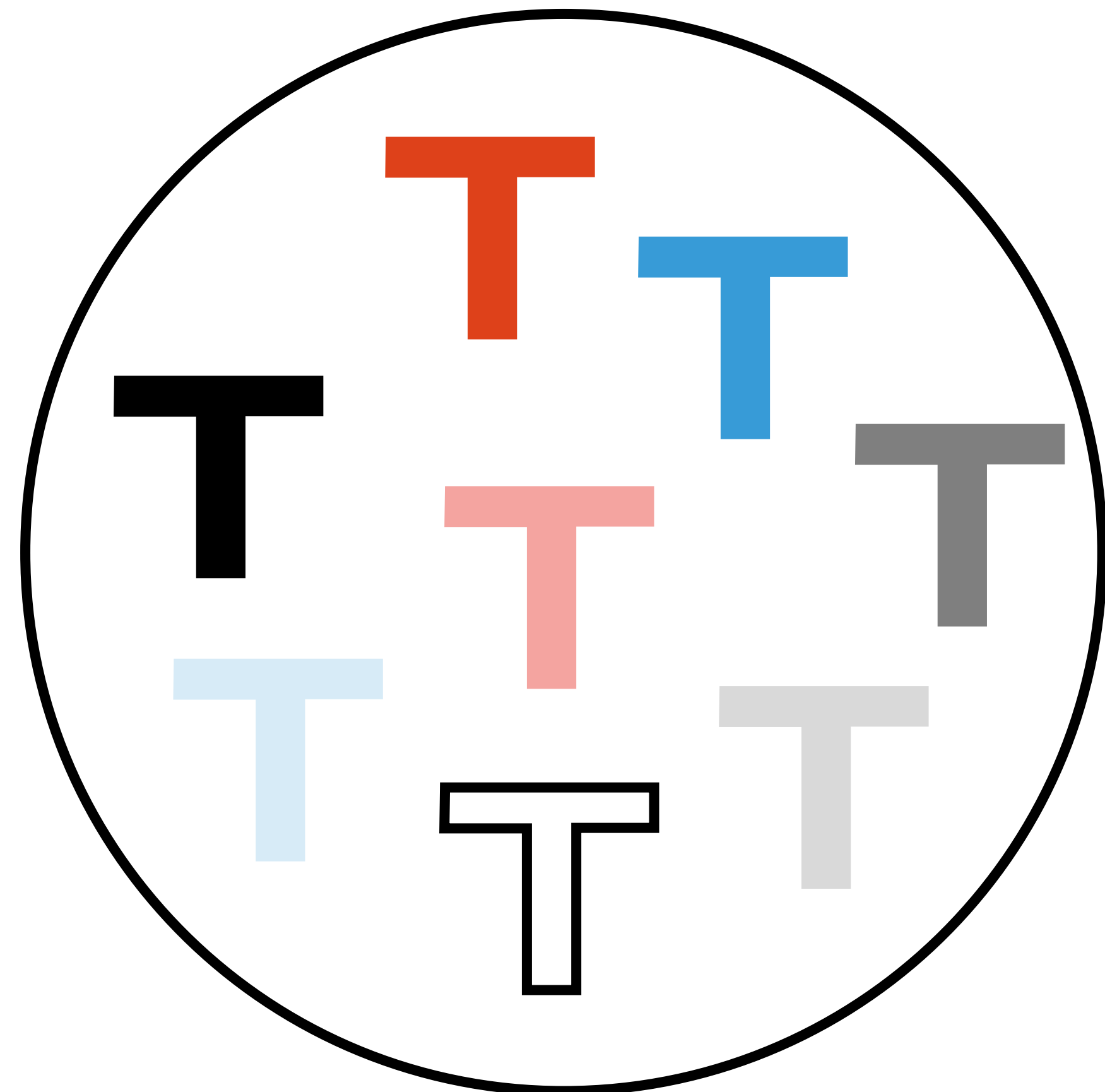


Cross-Functional Teams

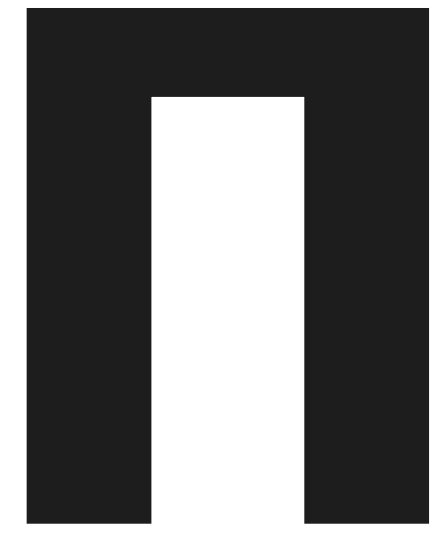
Individual



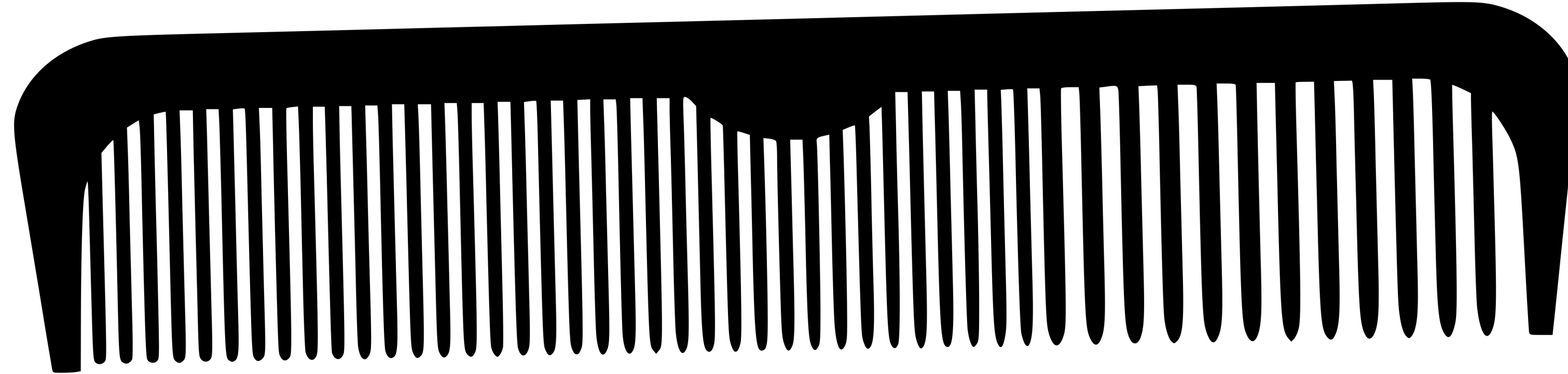
Scrum Team



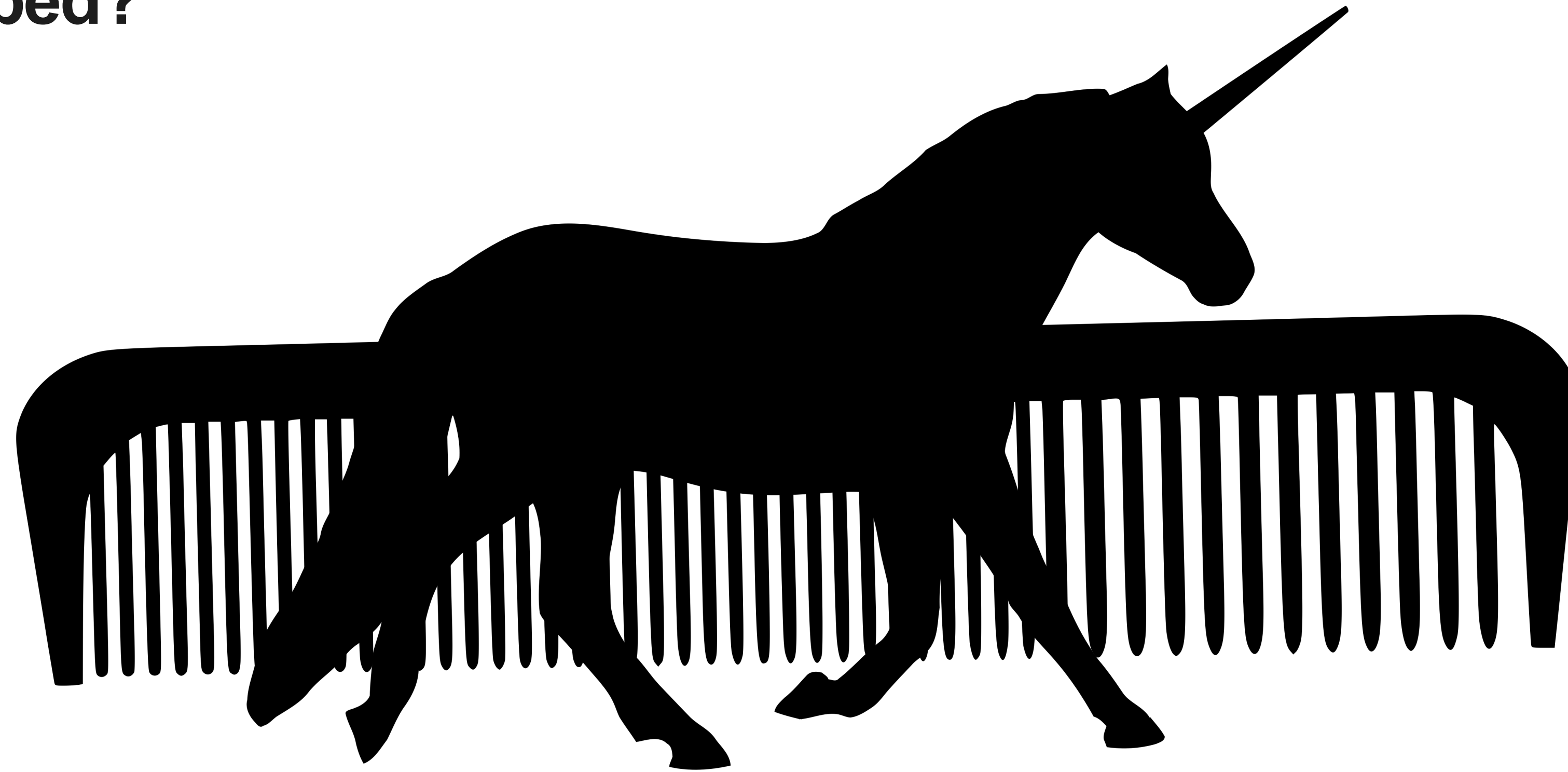
Pi-shaped



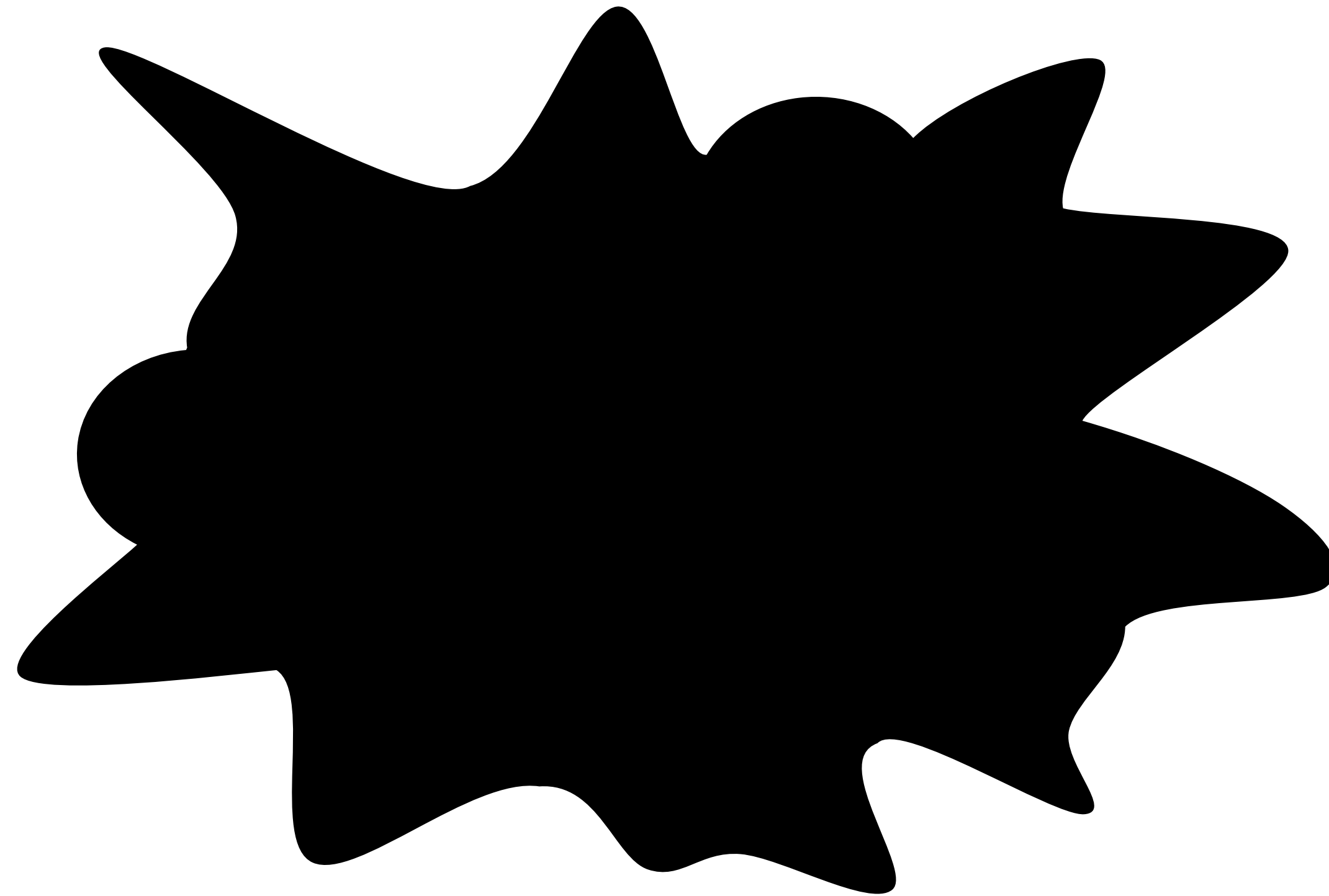
Comb-shaped?



Comb-shaped?



....**Blob-shaped.**



Why do you need all these skills in one head?

- T-shape people are important for DevOps
- Cross-functional teams are important for DevOps
- So... are cross-functional people important for DevOps?

3

Roots of Unicorn Culture



Why Unicorns?

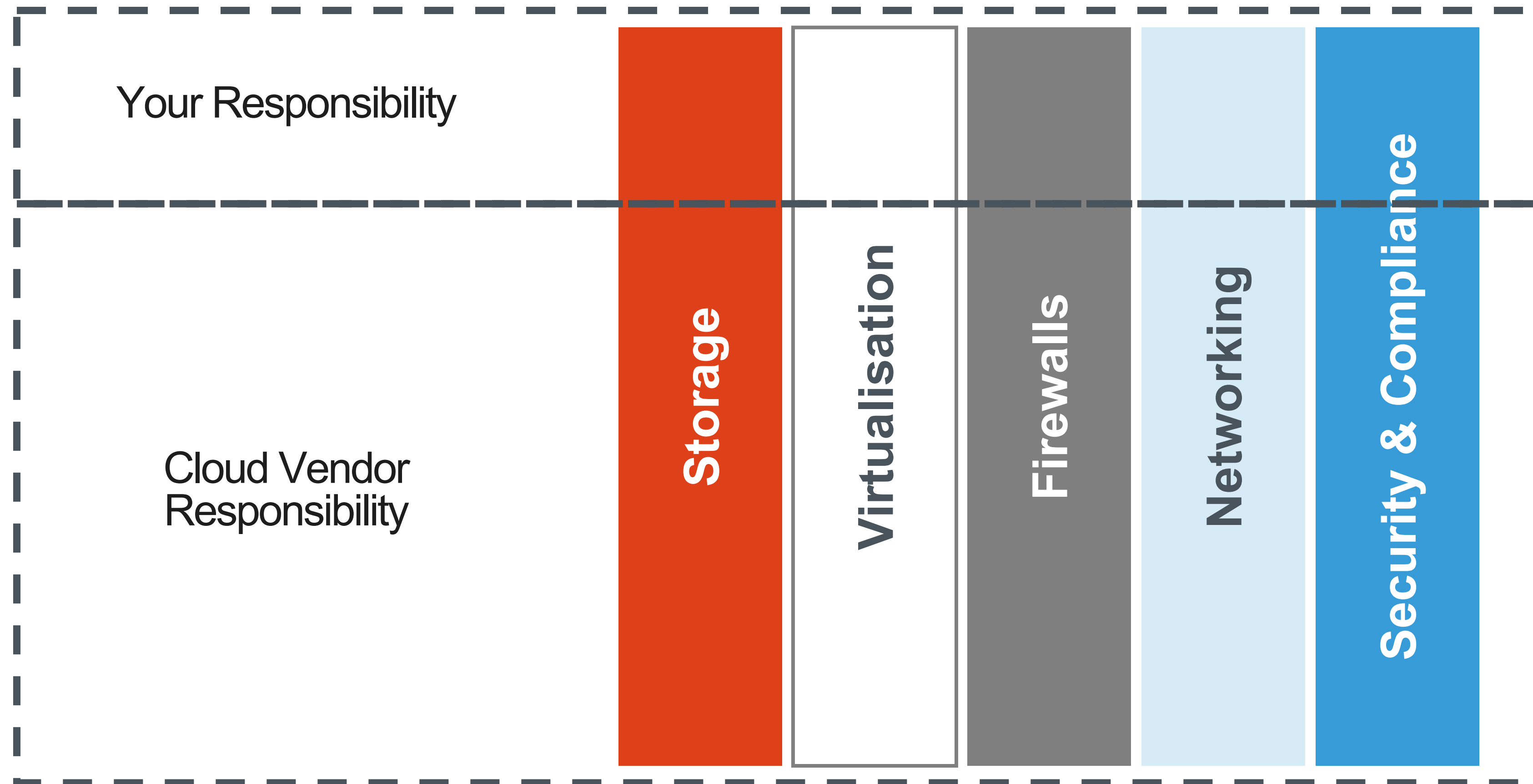
- Cost
- Industry Groupthink
- Organisational Inertia

Cost

- Why hire 10 people when you can hire a unicorn?
- Even if you buy into the mythology, there are some obvious risks
- Risk from SPOFs
- Risk of poaching by other unicorn hunters



Cloud and Cognitive Load

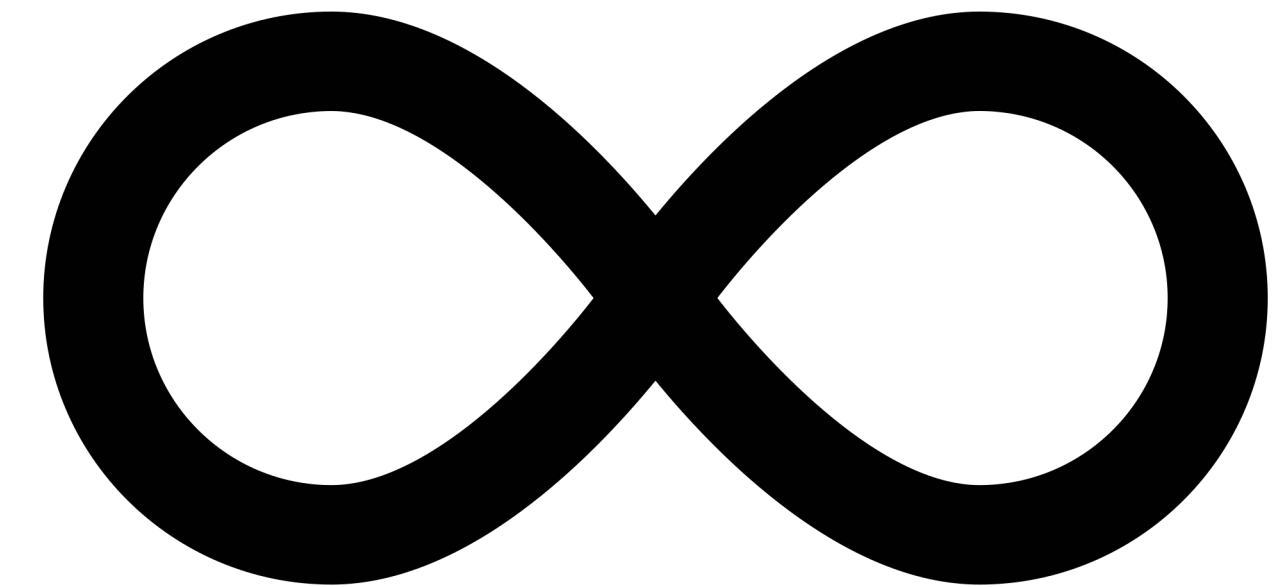


Is DevOps about saving cost?

- Not primarily.
- ITIL optimizes for cost: skill silos, shared at enterprise level
- DevOps optimizes for agility: cross-functional teams, as few shared people as possible
- You can save costs doing DevOps, but through local optimizations rather than big sweeping ones.
- It's better to see DevOps as an investment in making IT delivery more flexible and responsive.

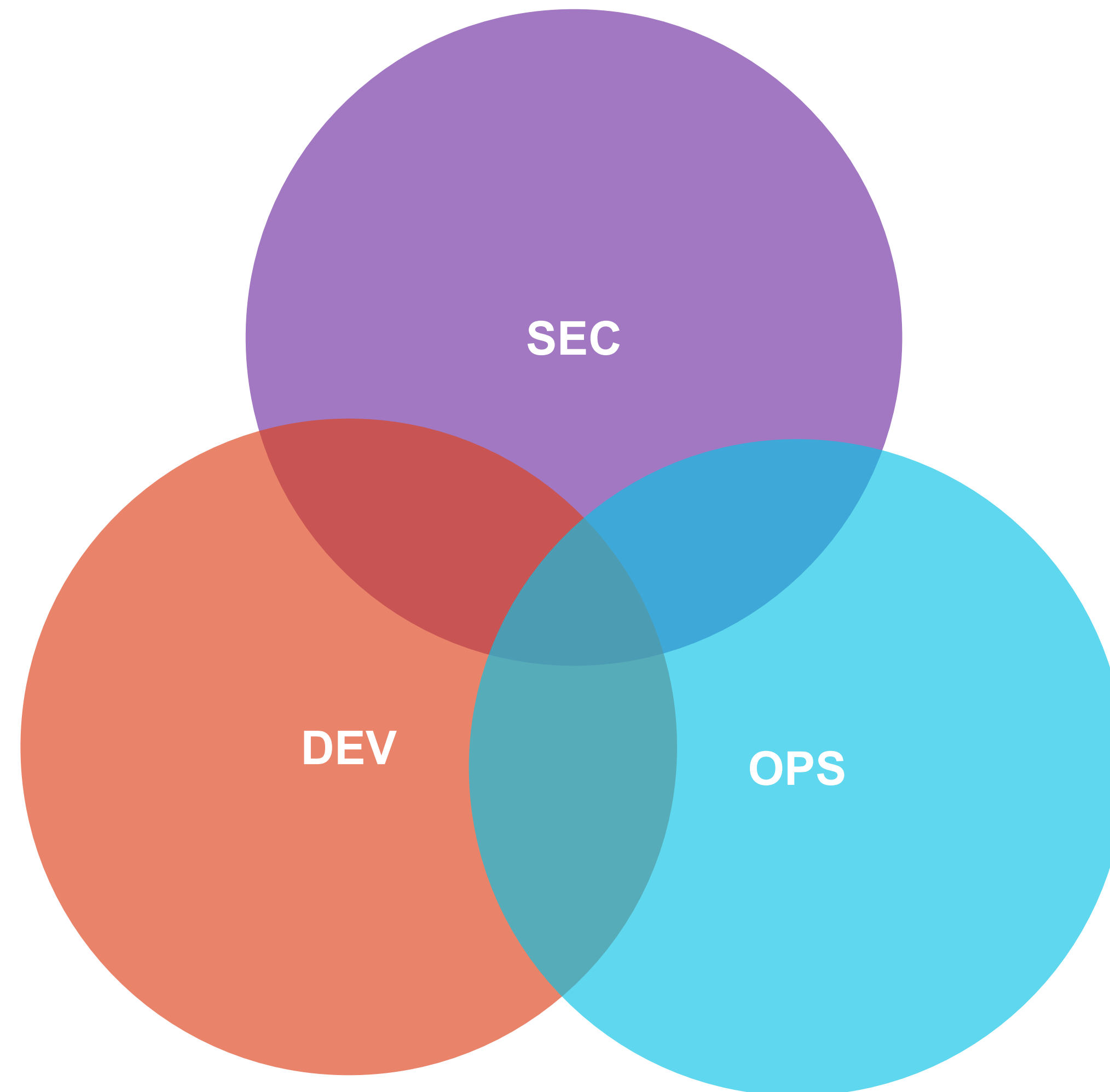
Industry Groupthink

- Everybody goes chasing unicorns, so everybody thinks they need unicorns
- The whole 'devops engineer' concept is symptomatic of this
- We've cemented the idea DevOps is something done by devops engineers.

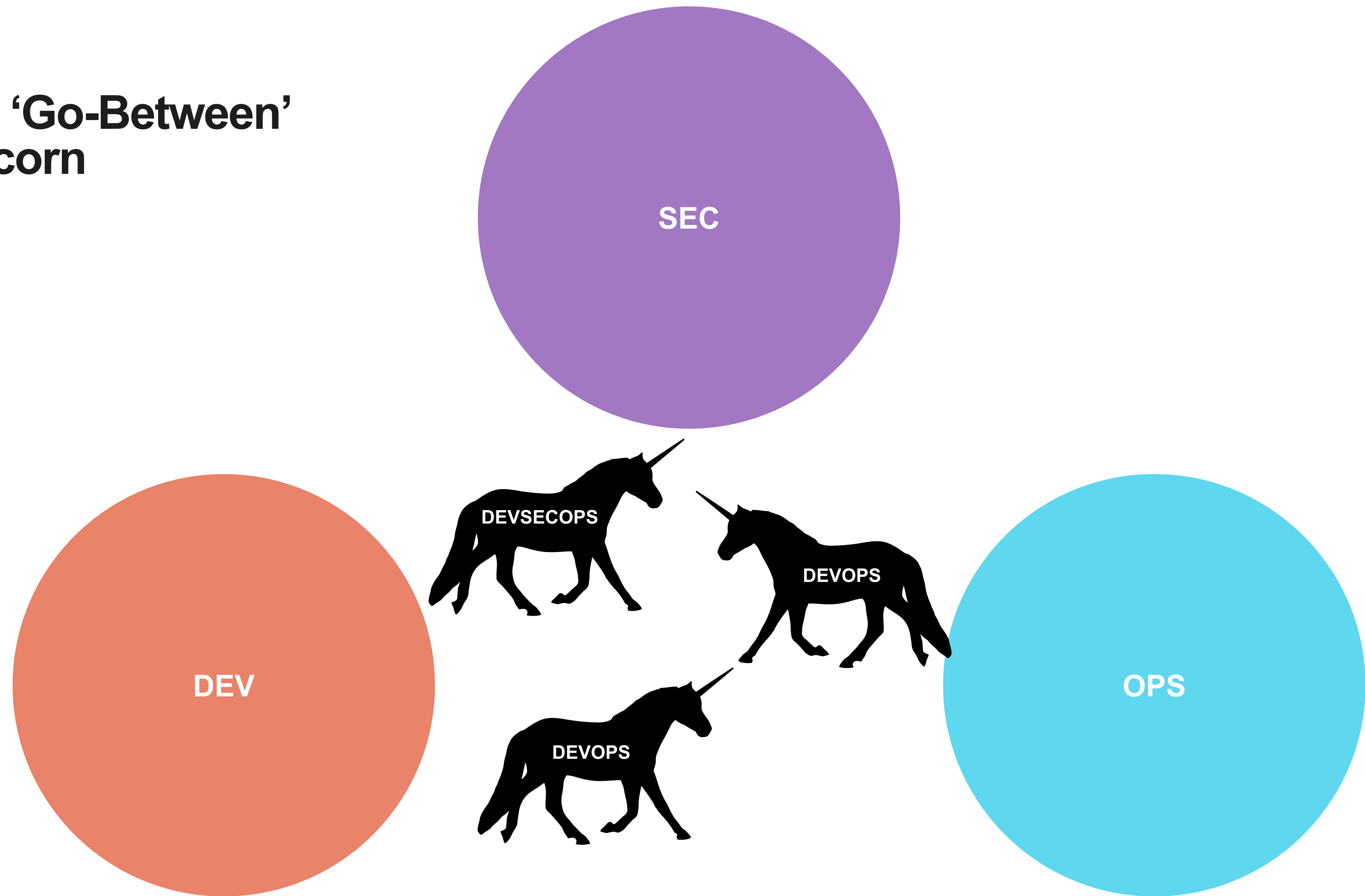


Organisational Inertia

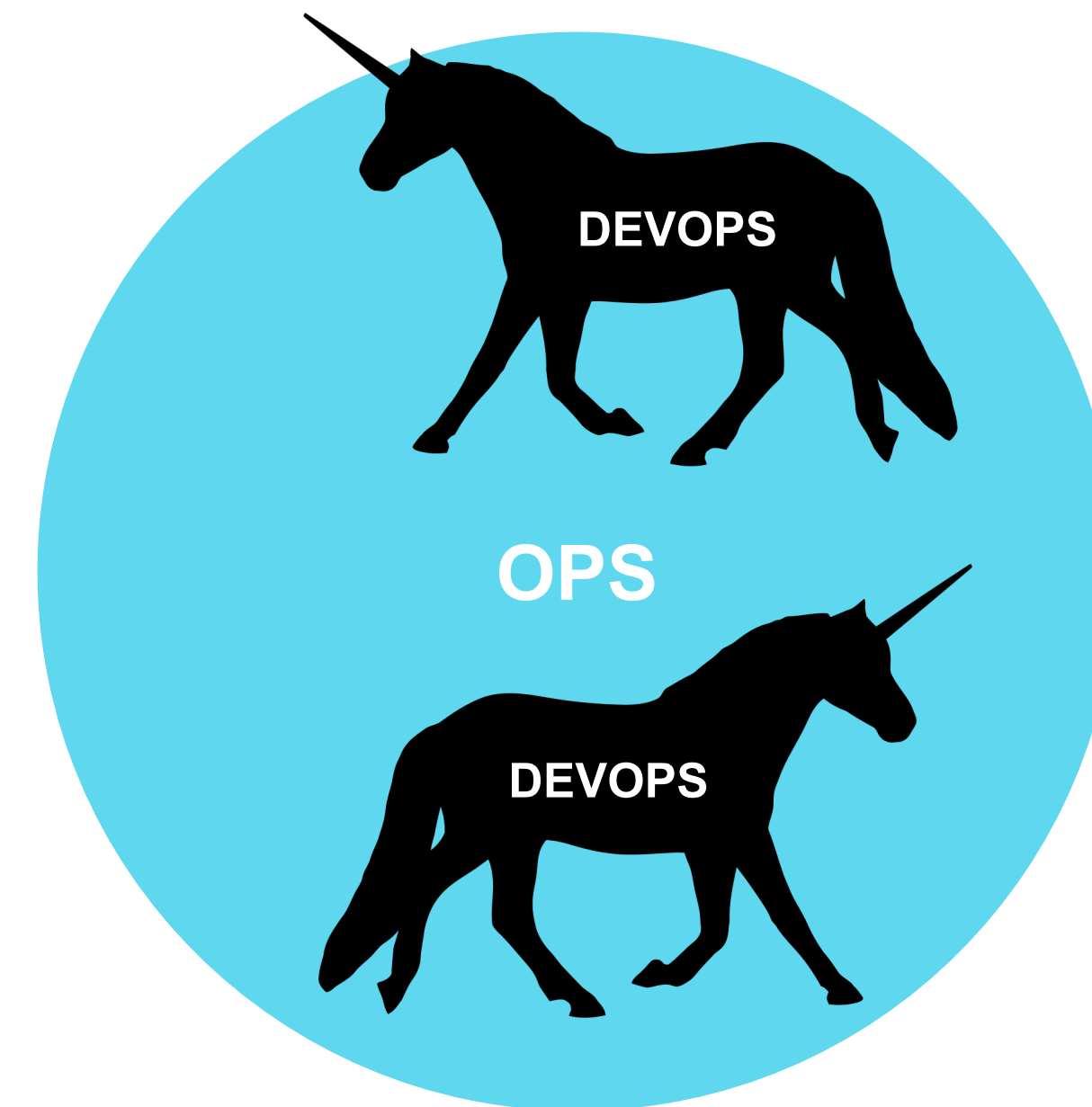
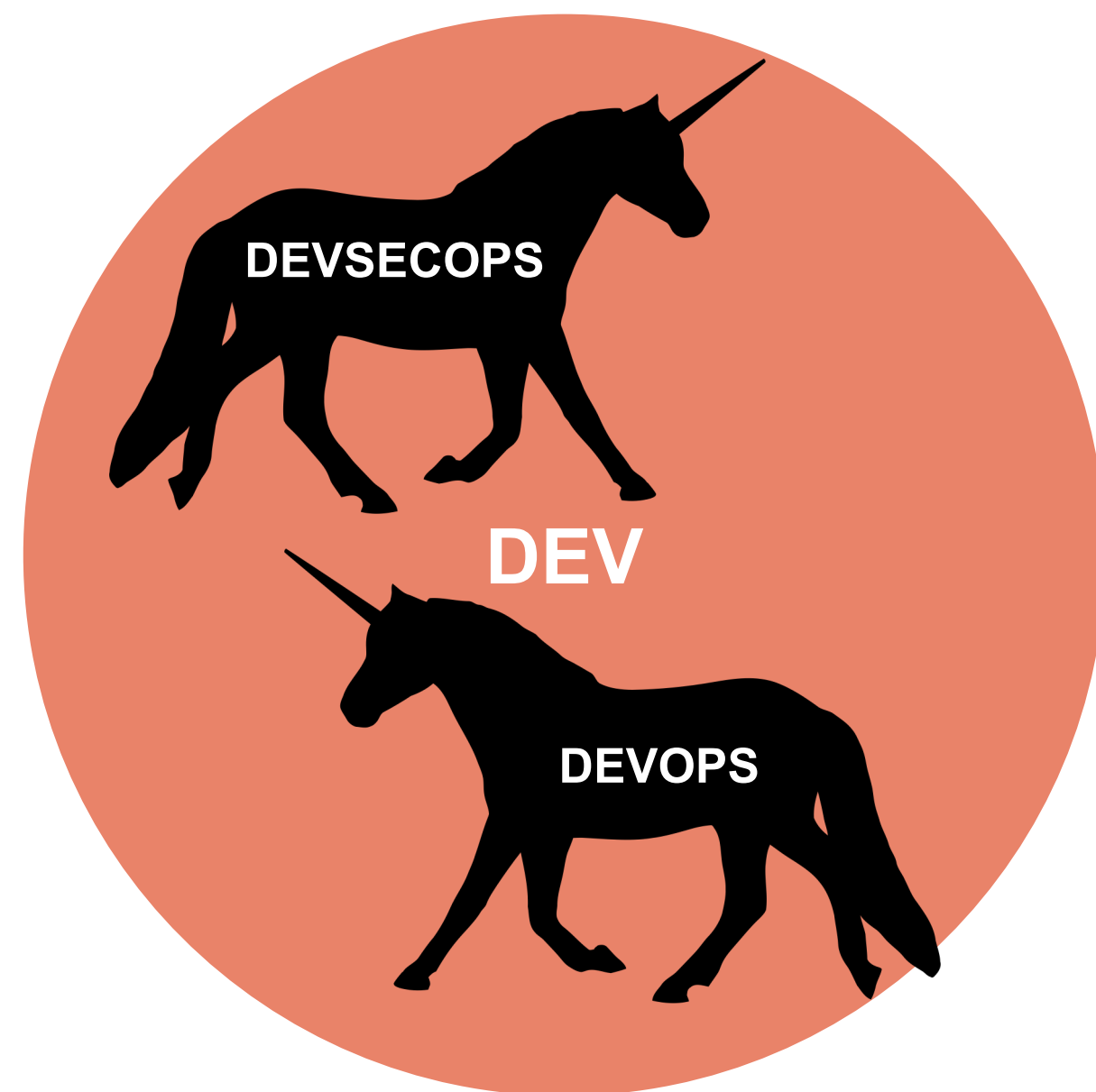
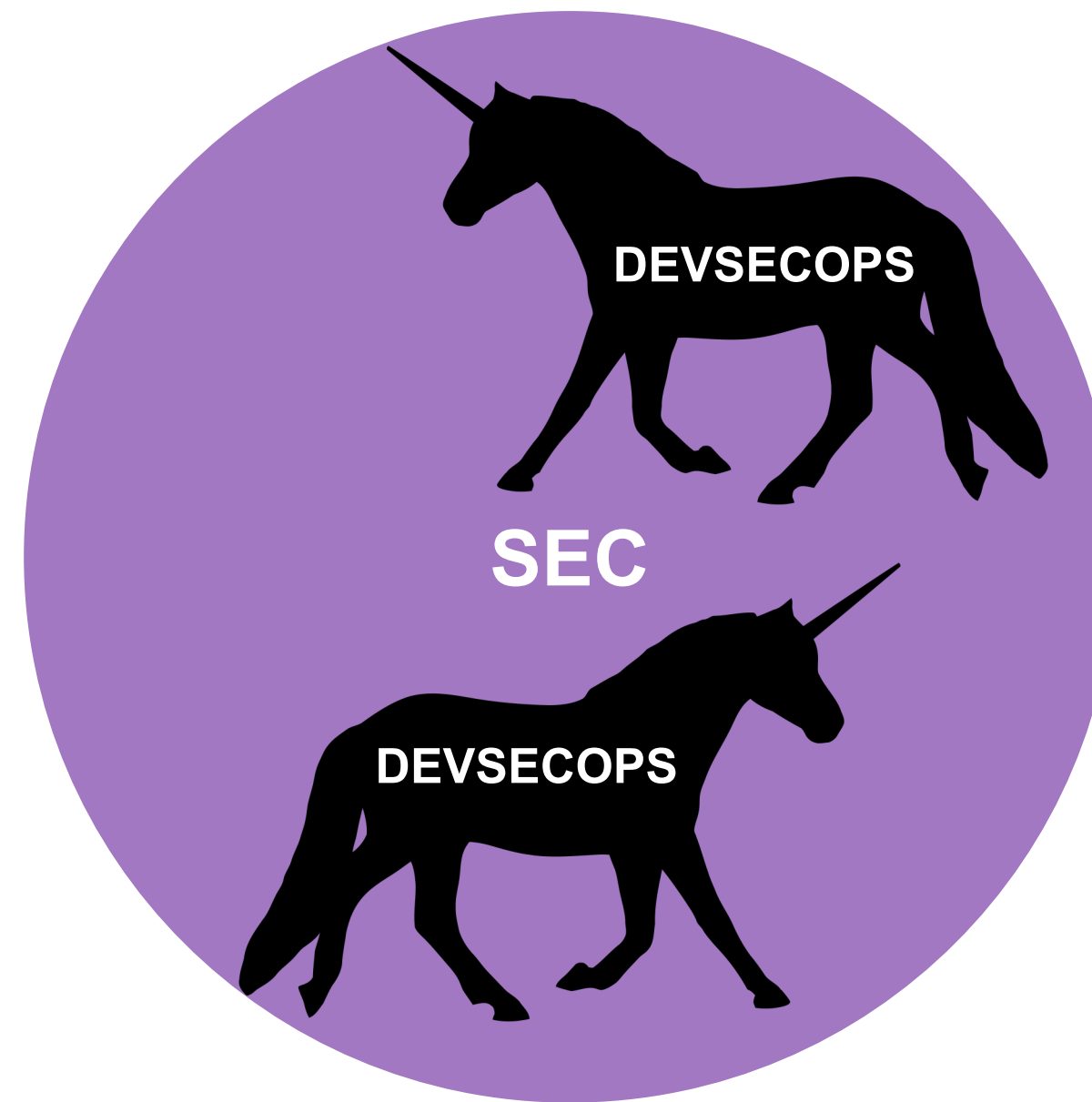
- Hiring unicorns *feels* like a move towards DevOps
- ...and it's easier than making the change you really need...
- Like getting Dev, Sec and Ops to work together.



The 'Go-Between' Unicorn



The 'Substitute for Dialog' Unicorn



4

Curse of the Unicorn



What's so bad about unicorns anyway?

- They're expensive
- They're hard to retain
- They're hard to find
- They're hard to place
- It's hard to train new ones
- It's hard to offer them career growth
- Reliance on them will block you from replicating/scaling your DevOps success
- They don't usually know as much as they seem to.
- They perpetuate a culture of firefighting and heroics.

Remember 'The Phoenix Project'?

- Brent, the guy who heroically cements himself as an indispensable bottleneck
- He's a unicorn
- They meant: beware of depending on people like Brent
- We heard: go hire a bunch of people like Brent



5

Escaping Unicorn Culture



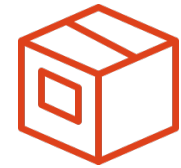
Oh no! what do we do?

- Admit you have a problem 😊

Then...

- **Architect your teams as you would architect a system.**

Architecting Teams



COHESIVE

Make sure teams have a cohesive, preferably quite narrow function or purpose, and that most of their effort contributes directly to fulfilling that purpose.



RESILIENT

Within the team, avoid single points of failure – a single individual who uniquely performs some critical function.

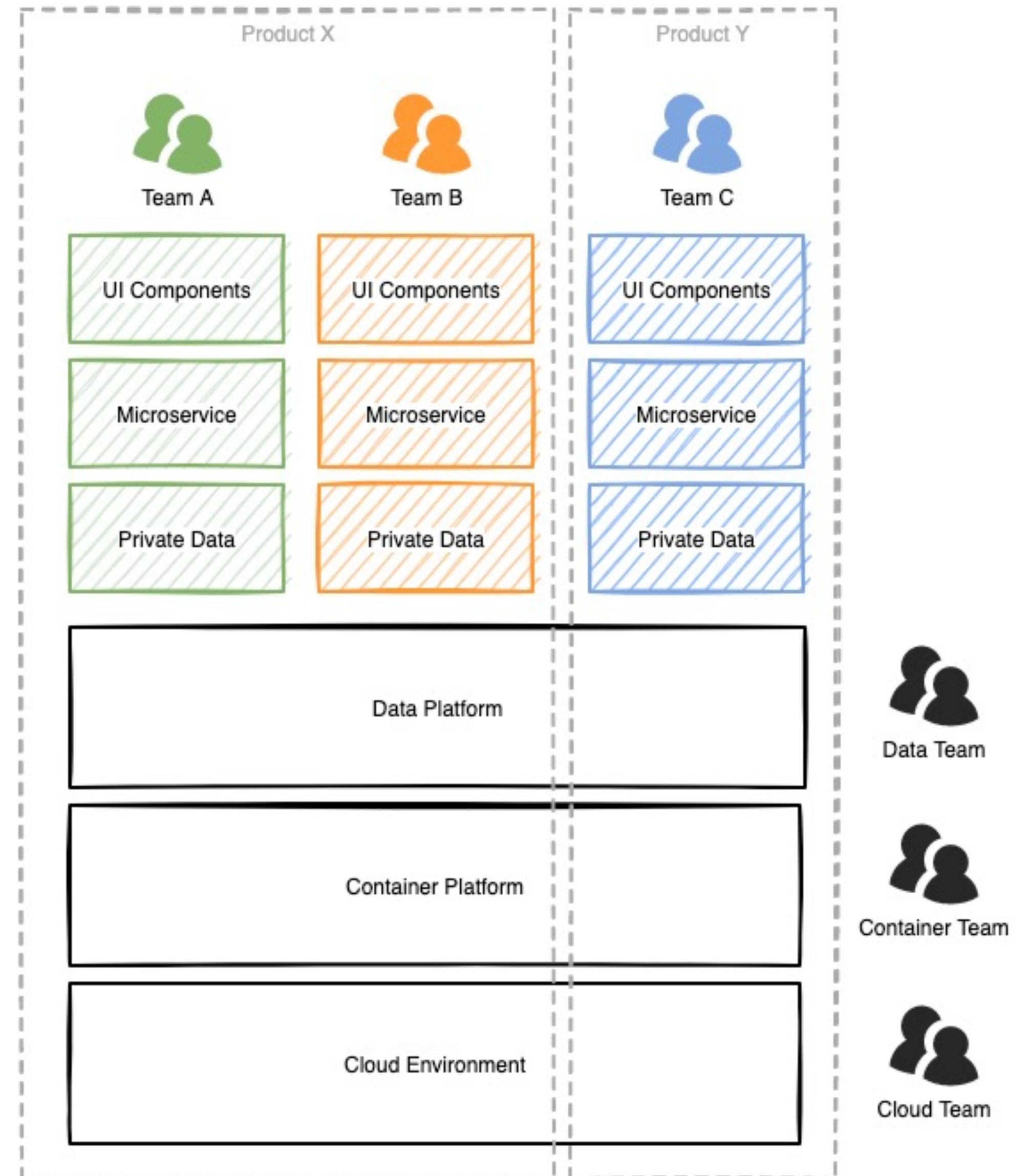


LOOSELY COUPLED

Plan for loose coupling between teams – but don't isolate them.

The Platform Problem

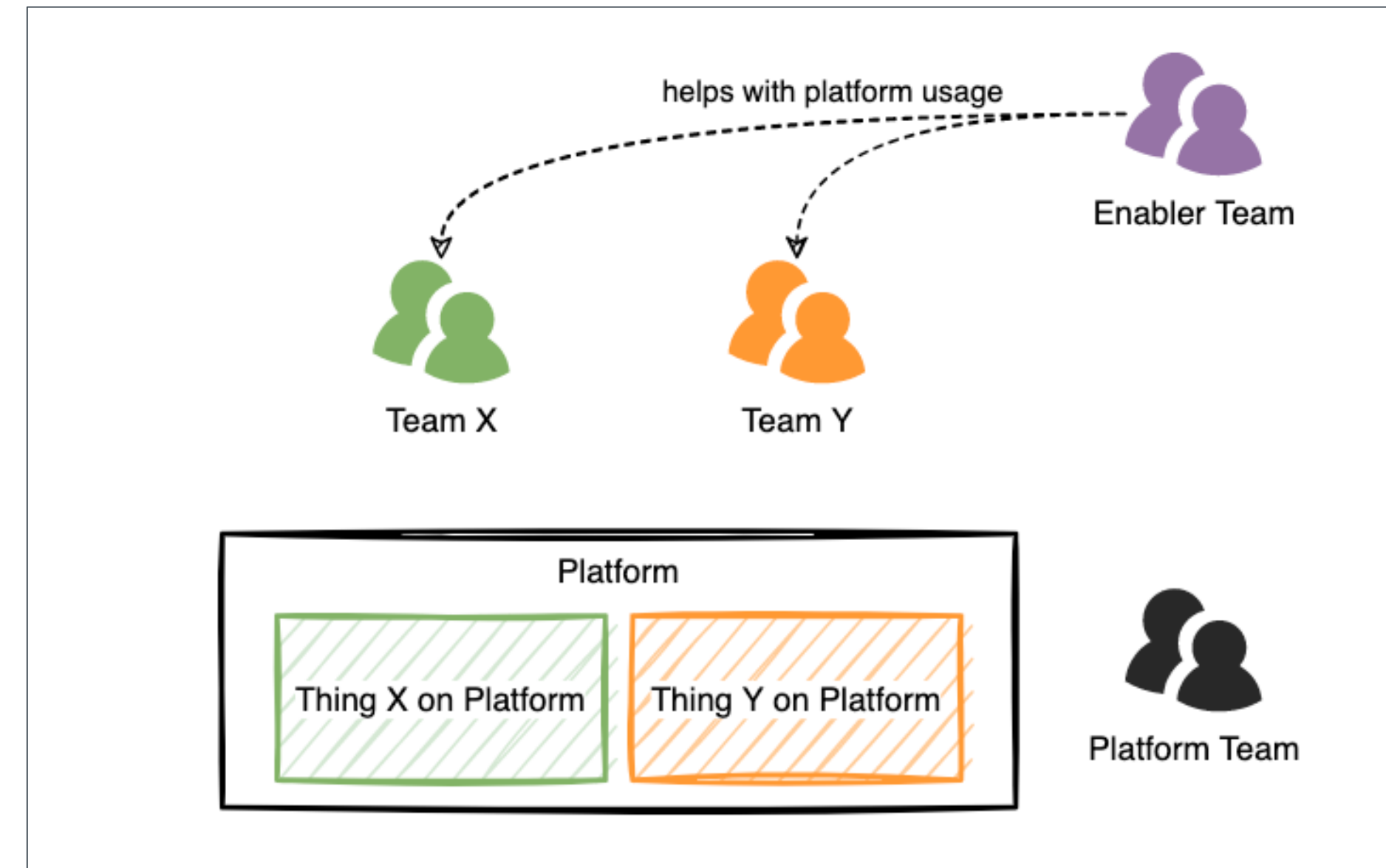
- For modern architectures, decoupling teams is easier to do at the application layer, which is one reason why microservices are so popular
- But it's often quite difficult when we get down to infrastructure and data
- It's especially hard with 'platform' components that logically ought to be shared by multiple teams.



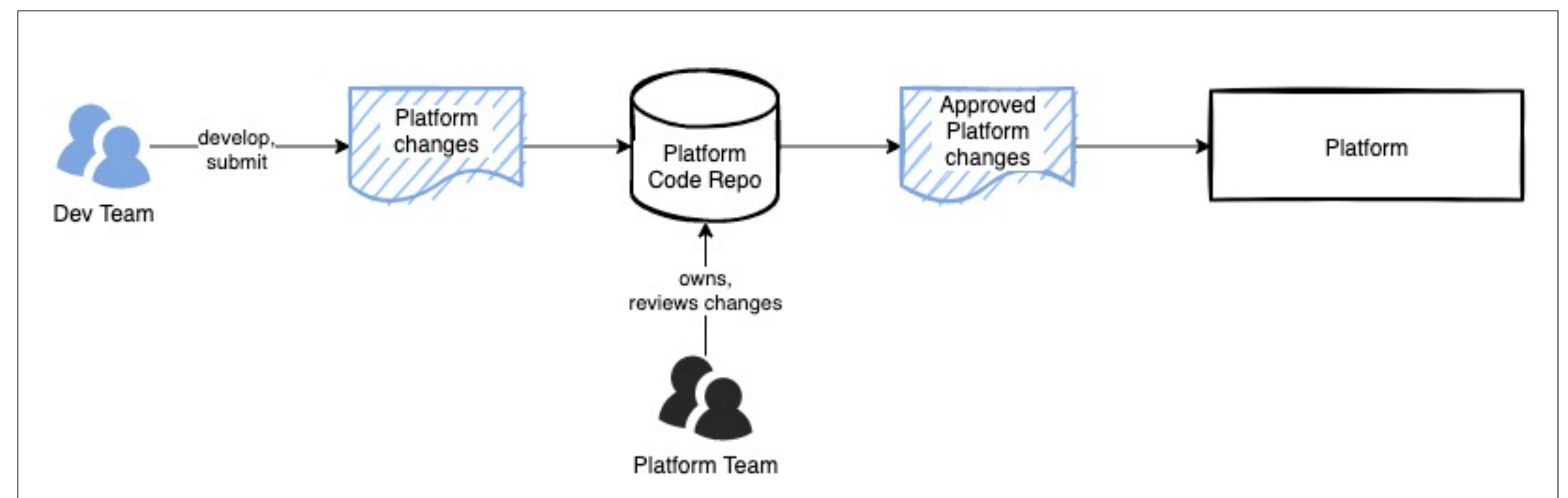
Platform Teams

- ▶ Follow a self-serve paradigm
- ▶ Encapsulate cohesive platform capabilities
- ▶ Create speed gains, but also reduces cognitive load on other teams
- ▶ Can be more tailored to the specific needs of an organisation or teams within it
- ▶ Platforms can be domain-aligned!

Multi-tenant PaaS Model

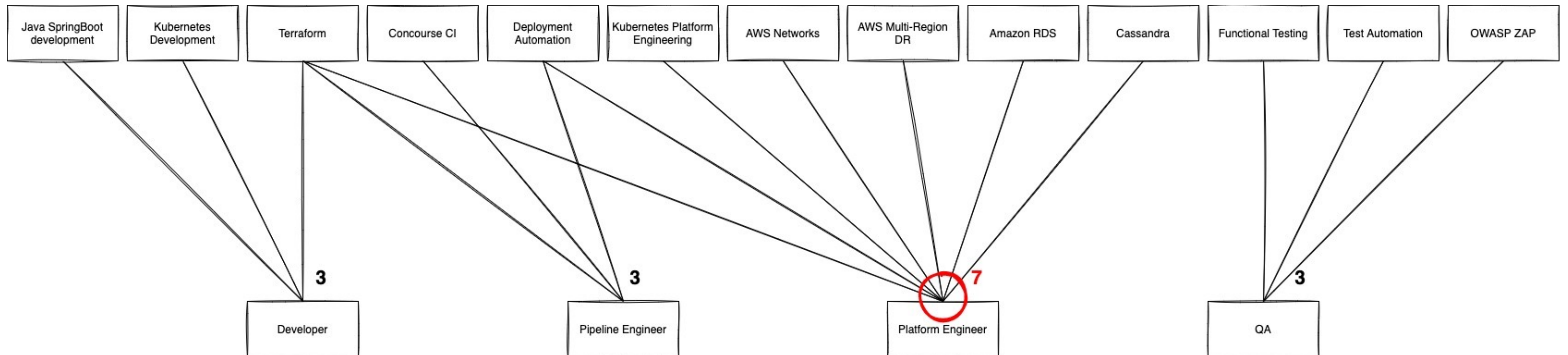


PR Model



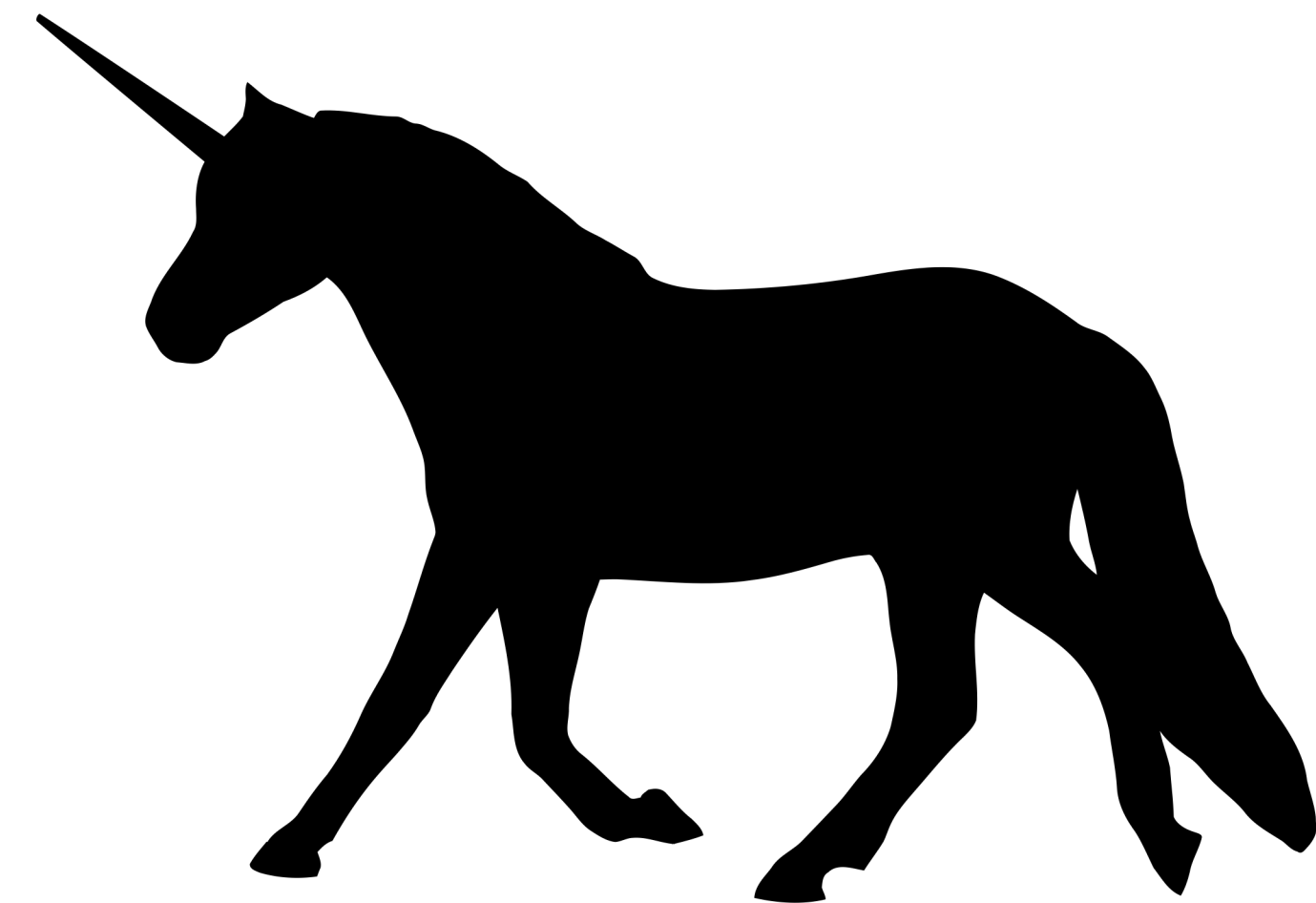
Role Mapping

- Break out a list of cohesive, granular roles you need in your team
- Map granular roles 1..* to broader roles in the team
- Think about it like RBAC
- Split any blob-shaped team roles that emerge between multiple people



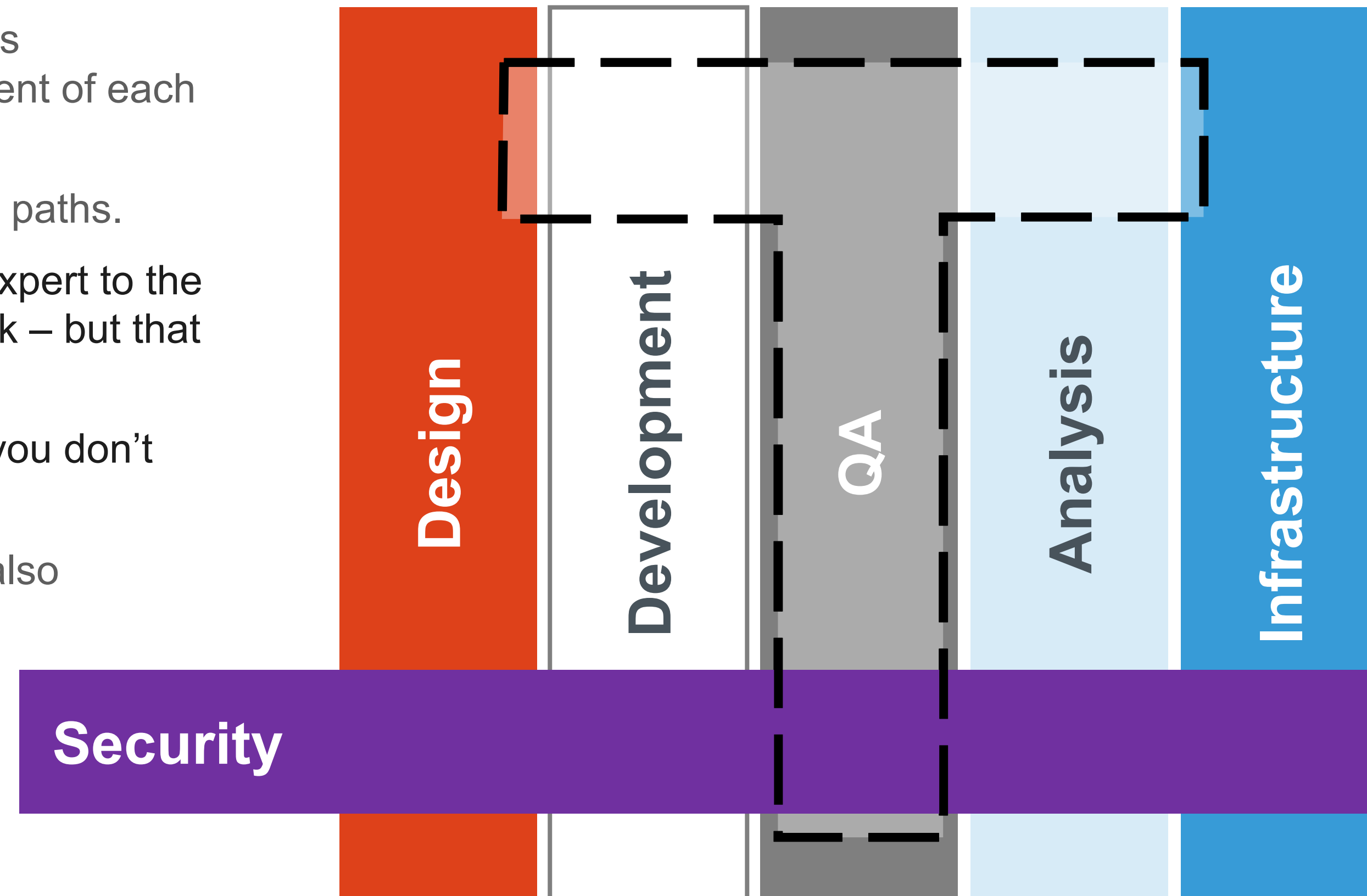
What about our existing unicorns?

- Use them carefully, you need to defuse the negative impact
- They shine in exceptional circumstances, so may perform well in more short-lived roles
- For a more stable, strategic role in the organisation, apply the same role-shaping ideas to unicorns as everyone else...
- ...but prepare for turbulence.



And security specifically?

- Scalability comes from recognising that security is interdisciplinary, so it's best imagined as a segment of each role rather than a separate role.
- You need role-specific security training and skills paths.
- You could augment this by attaching a security expert to the team or business domain area for faster feedback – but that role can easily become a unicorn.
- Security Champions are a better bet – provided you don't overburden them.
- Carving out some sensible platform capabilities also simplifies the security load on dev teams.



6

Q&A



