Masterclass: success patterns fast flow and Team Topologies

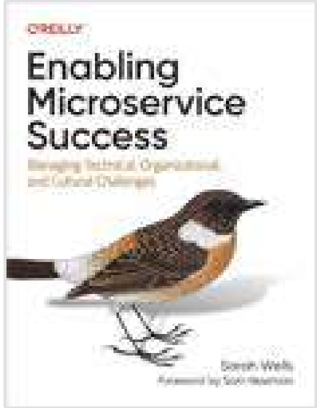
Domain-Driven Design Europe - 29/5/24

conflux

About the facilitators

Sarah Wells
Associate Principal at Conflux

Technology consultant and author



O'REILLY

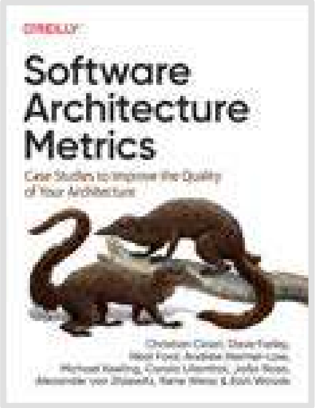
Enabling Microservice Success

Managing Technical, Organizational, and Cultural Challenges

Sarah Wells
Foreword by Sam Newman

João Rosa
Associate Principal at Conflux

Co-author of *Software Architecture Metrics*



O'REILLY

Software Architecture Metrics

Case Studies to Improve the Quality of Your Architecture

Christian Ciceri, Dave Farley, Neal Ford, Andrew Harmel-Law, Michael Keeling, Carola Lilienthal, João Rosa, Alexander von Zitzewitz, Rene Weiss & Eoin Woods

Masterclass - outline

Duration	Details
90 minutes	Part 1: The importance of the evolution of team interactions and team boundaries
	Break (30 mins)
90 minutes	Part 2: Success with platforms and platform thinking for fast flow
	Lunch (13:00 - 14:00)
90 minutes	Part 3: Finding good boundaries for flow using Independent Service Heuristics
	Break (30 mins)
90 minutes	Part 4: Skills paths and aptitudes for fast flow + wrap-up

Objective of this masterclass

Accelerate the adoption and evolution of fast flow and Team Topologies within your organization through deep insights into the key ideas and techniques that work well “on the ground”. Put into practice the ideas and techniques through guided exercises and discussions.

Results of this masterclass

Outcomes

Deep insights into fast flow and Team Topologies.

Confidence in explaining the nuances of fast flow and Team Topologies.

Gain advanced awareness of platform dynamics for fast flow.

Activities

Map some fast flow skills pathways for your organization.

Develop the skills to facilitate an Independent Service Heuristics session within your organization.

Use the Team Topologies modeling shapes to explore organization dynamics.

Hands-on experience of tools and techniques, guided by a Conflux practitioner.

Part 1 - The evolution of team interactions and team boundaries

conflux

Outline of part 1

- We begin Part 1 by a group critique of some team interaction diagrams based on different industry contexts to help embed the learning.
- We then revisit Chapters 7 and 8 of the Team Topologies (TT) book and the implications of the team interaction modes, and how Team APIs can help cross-team communication.
- We finally use the open-access TT team modeling shapes to become familiar with thinking in terms of the evolution of teams and interactions, not simply a target design.

Rules and guidelines for Team Interaction Modeling

teamtopologies.com

Team Interaction Modeling with Team Topologies - Team Topologies

Team Topologies provides a combination of principles and practices that can help structure and evolve an organization for effective collaboration, autonomy, delivery focus, and product alignment. Using the Team Topologies shapes and diagramming principl...

https://teamtopologies.com/tim

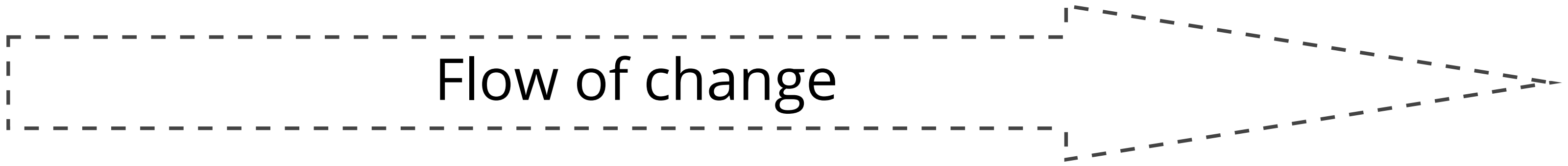
When using the team shapes to create your own diagrams, there are a number of constraints that should be applied:

1. There is always an implied flow of change from left to right in the diagram (with apologies to people more familiar with a right-to-left flow!).
2. A key aspect of Stream-aligned teams is that they have end-to-end responsibility for a flow of change to the live services/systems, with no hand-offs to other teams. There should therefore be no other team between a Stream-aligned team and their customers/users (on the right of the diagram).
3. Team shapes should be solid to represent their long-lived nature.
4. Interaction Mode shapes should be 50% transparent to represent the interaction's short-lived nature.
5. Stream-aligned teams should generally never provide an X-as-a-Service directly. Instead, data or services from the Stream-aligned team should be made available ‘as a Service’ via a platform.
6. If an X-as-a-Service or Collaboration interaction crosses over multiple teams, it may be appropriate to use a black asterisk, ’*’, to clarify which teams are interacting.

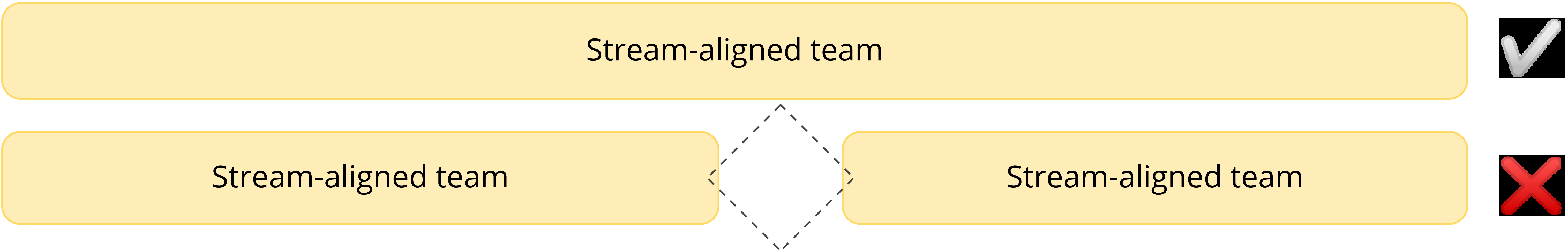
conflux

Rules and guidelines for Team Interaction Modeling - 1 & 2

1 - There is always an implied flow of change from left to right in the diagram (with apologies to people more familiar with a right-to-left flow!).

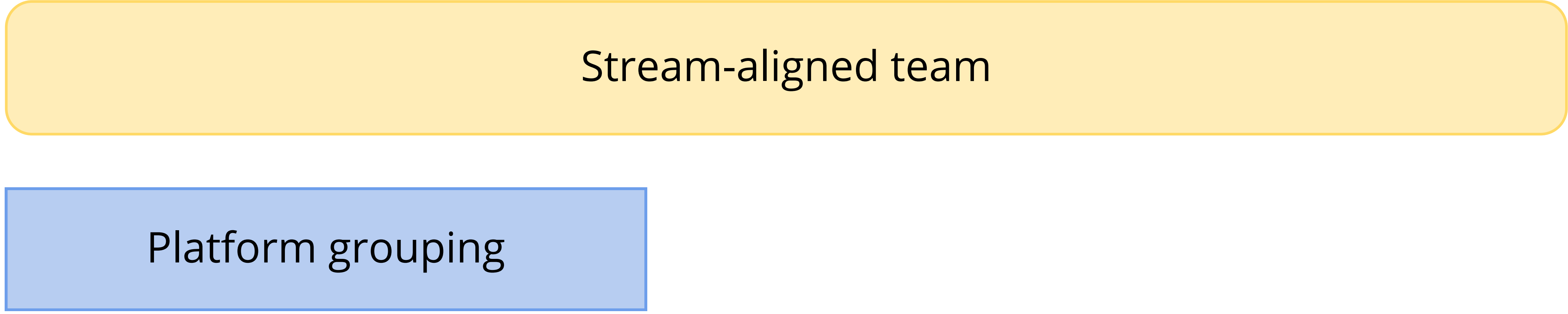


2 - A key aspect of Stream-aligned teams is that they have end-to-end responsibility for a flow of change to the live services/systems, with no hand-offs to other teams. There should therefore be no other team between a Stream-aligned team and their customers/users (on the right of the diagram).

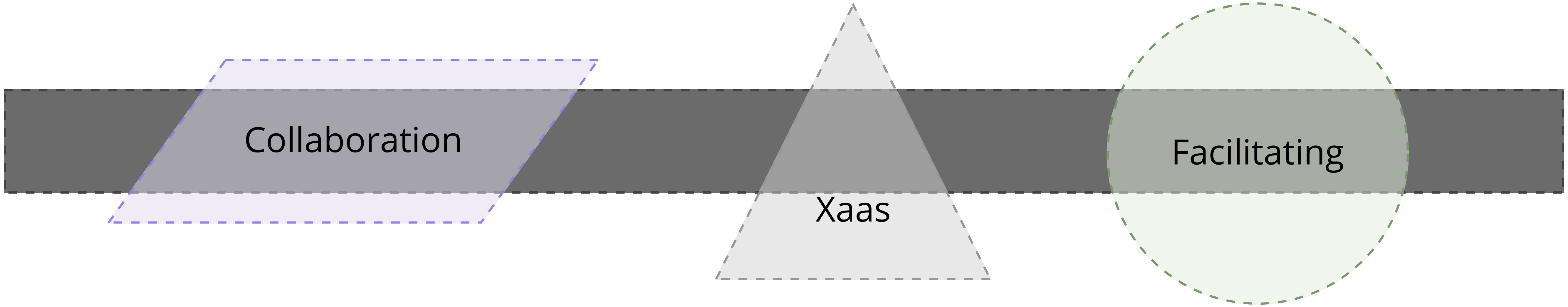


Rules and guidelines for Team Interaction Modeling - 3 & 4

3 - Team shapes should be solid to represent their long-lived nature.

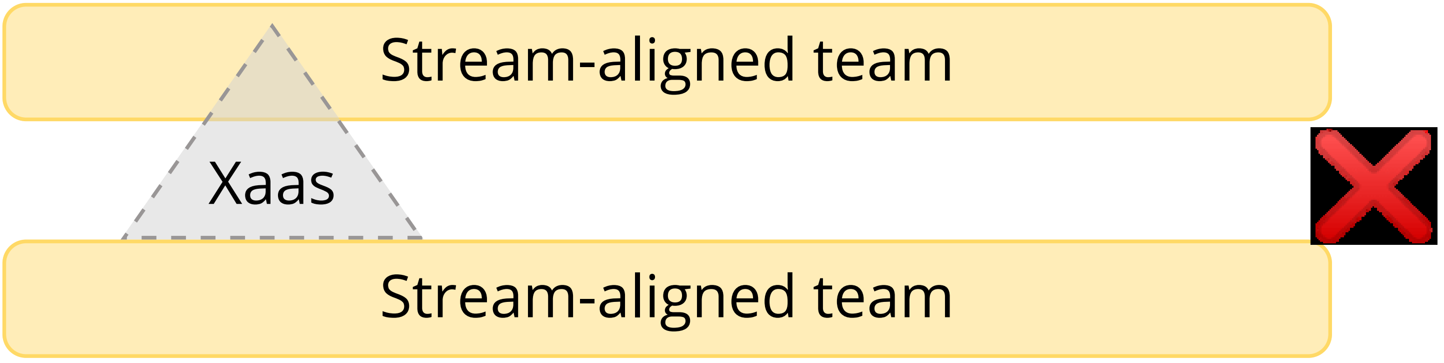


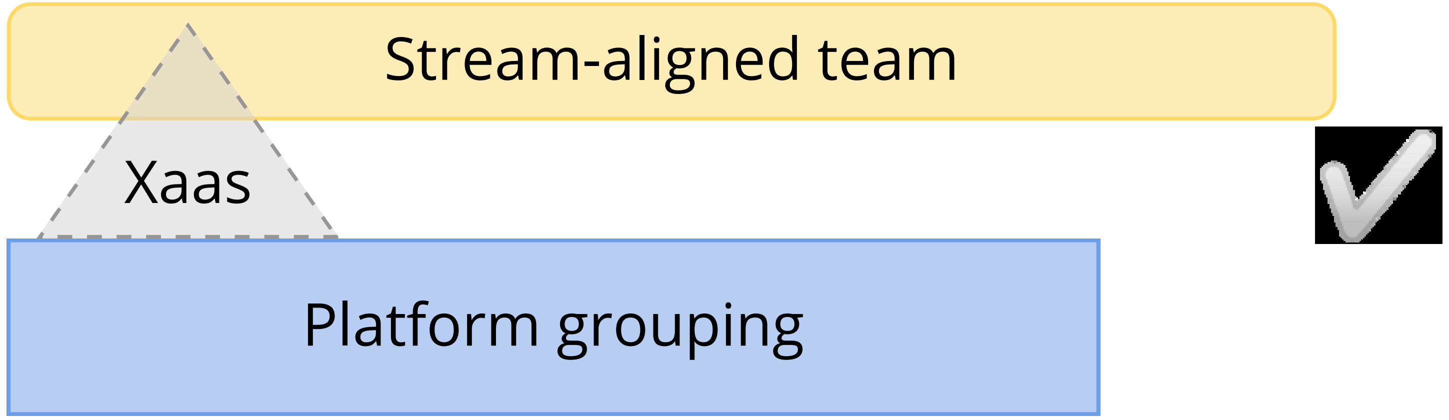
4 - Interaction Mode shapes should be 50% transparent to represent the interaction's short-lived nature.



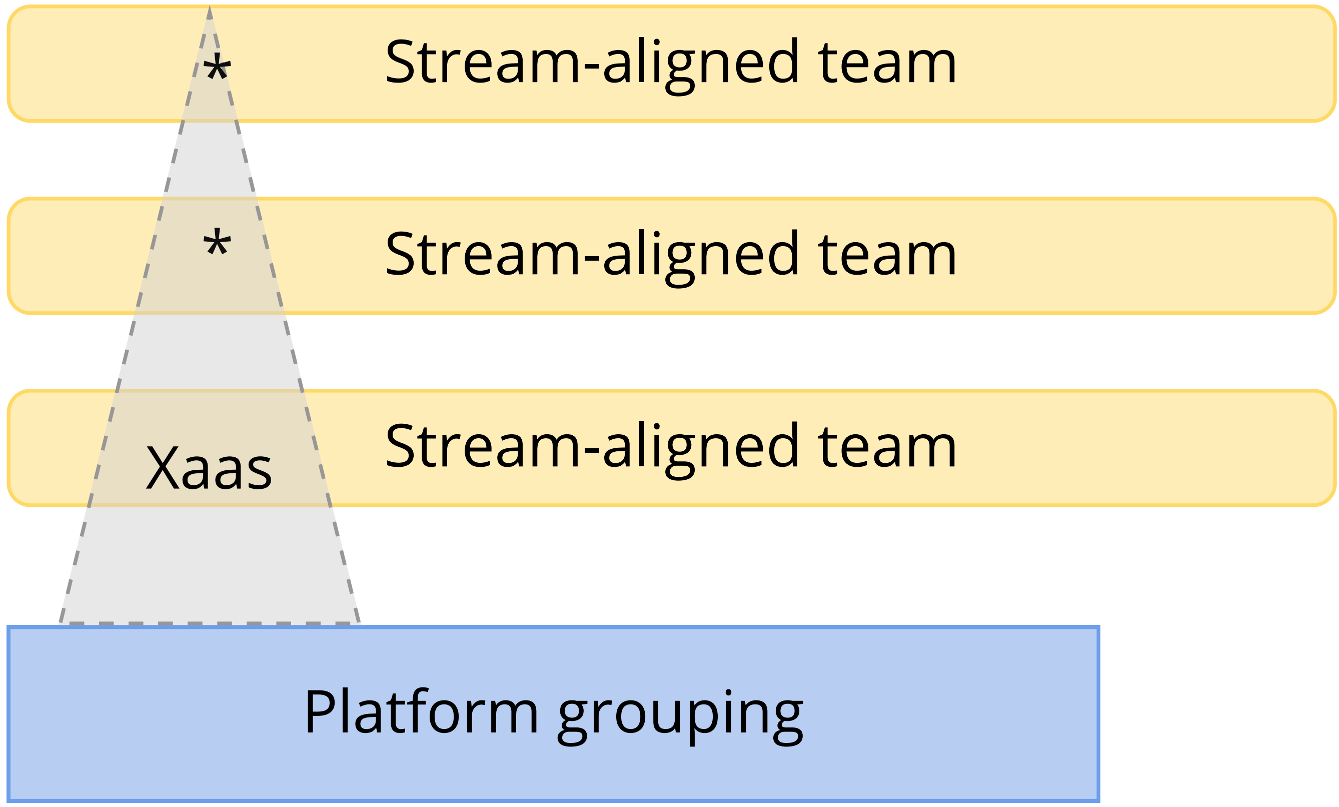
Rules and guidelines for Team Interaction Modeling - 5 & 6

5 - Stream-aligned teams should generally never provide an X-as-a-Service directly. Instead, data or services from the Stream-aligned team should be made available ‘as a Service’ via a platform.



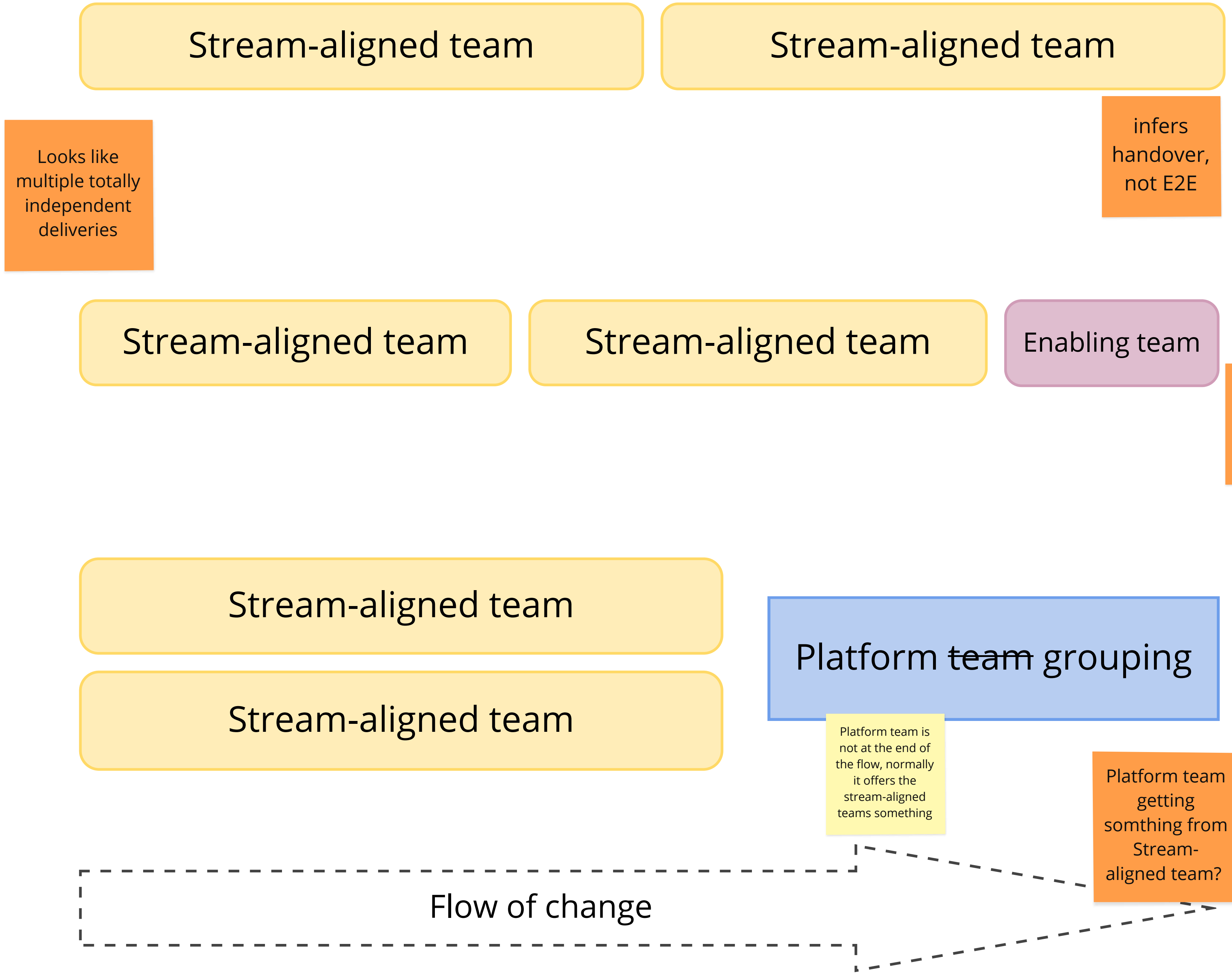


6 - If an X-as-a-Service or Collaboration interaction crosses over multiple teams, it may be appropriate to use a black asterisk, ’*’, to clarify which teams are interacting.



conflux

Team interaction diagrams - bad example 1



Looks like multiple totally independent deliveries

infers handover, not E2E

enabling team is not enabling other teams

Platform team is not at the end of the flow, normally it offers the stream-aligned teams something

Platform team getting somthing from Stream-aligned team?

Information gets Lost
No ownership

Not end to end responsibility

platform not enabling in the flow

Cannot handover to platform team, platform, others teams use platform

handovers to other teams in flow of change.
enabling team in flow of change, they should be facilitating.

Not E2E

not E2E

enabling team acting as a stream aligned team

Is the Stream aligned team delivering to the enabling team?

Position of platform team

Platform team should not be in the end of the flow

Enabling team in a flow not with SAT

Plateform team doesn't interact with a handover

Hand over from a stream-aligned team to another stream-aligned team

hand-over, imlying not E2E

not entire responsable

Interactions are not defined

Platform team not XAAS to SAT(s)

Enabling team doesn't "enable" any team

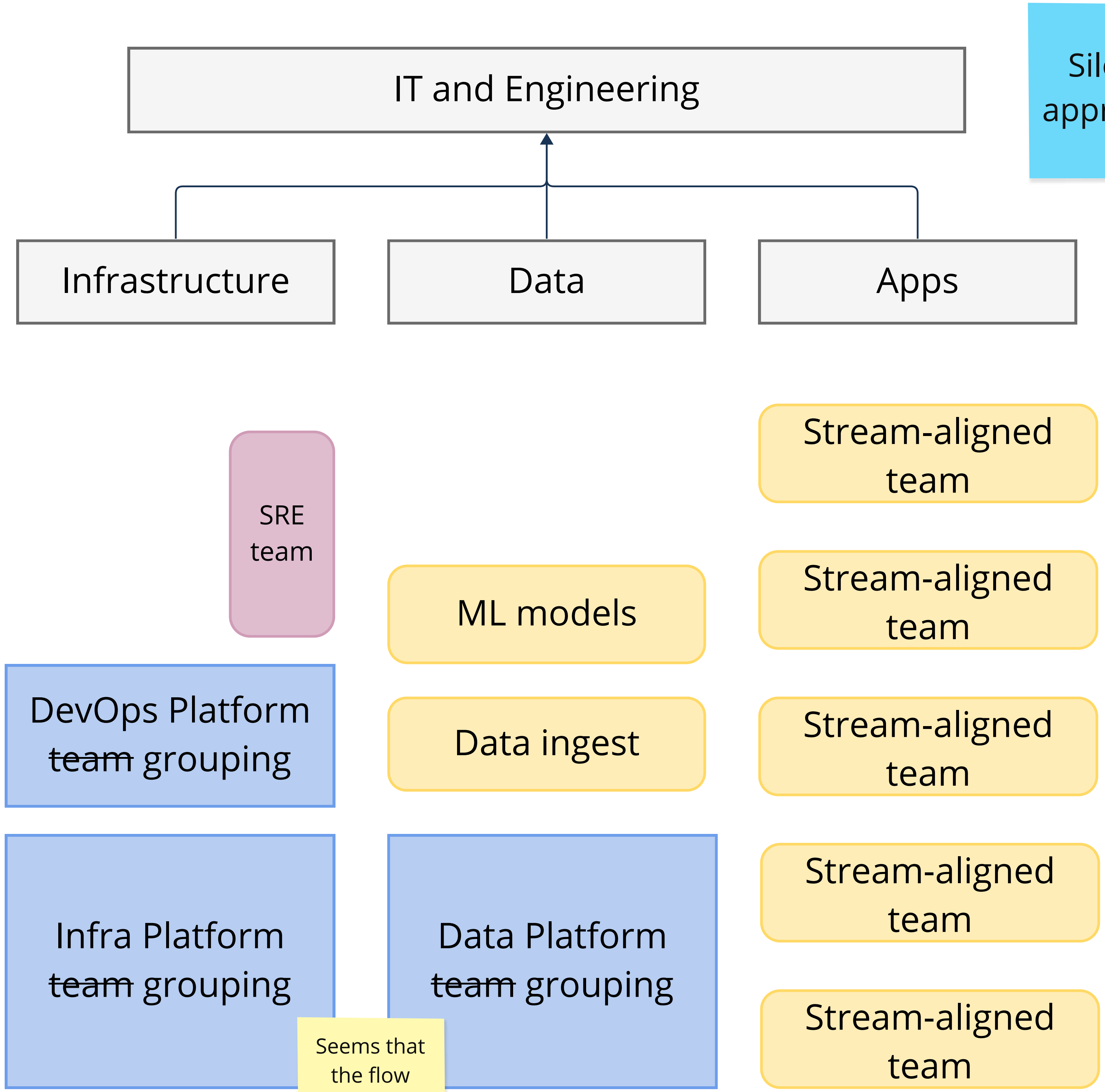
SATs handing over

Changes should be made from platform grouping

Platform interaction

conflux

Team interaction diagrams - bad example 2



Seems that the flow started with P-Teams

Siloed approach

Data Apps and Infrastructure seperated

No end-to-end flow defined

siloed

no flow

no interactions

Silos , Dsiconnect from Business Outcomes

Silo thinking

machine learning not a Valuestream activity

a topolgogy organisation baseds on the org structure

missing interactions

Flow interruptus

No interactions / teams seperated

Data Ingest not a Valuestream activity

Missing interaction patterns

The relations between teams are not clear

👍 1

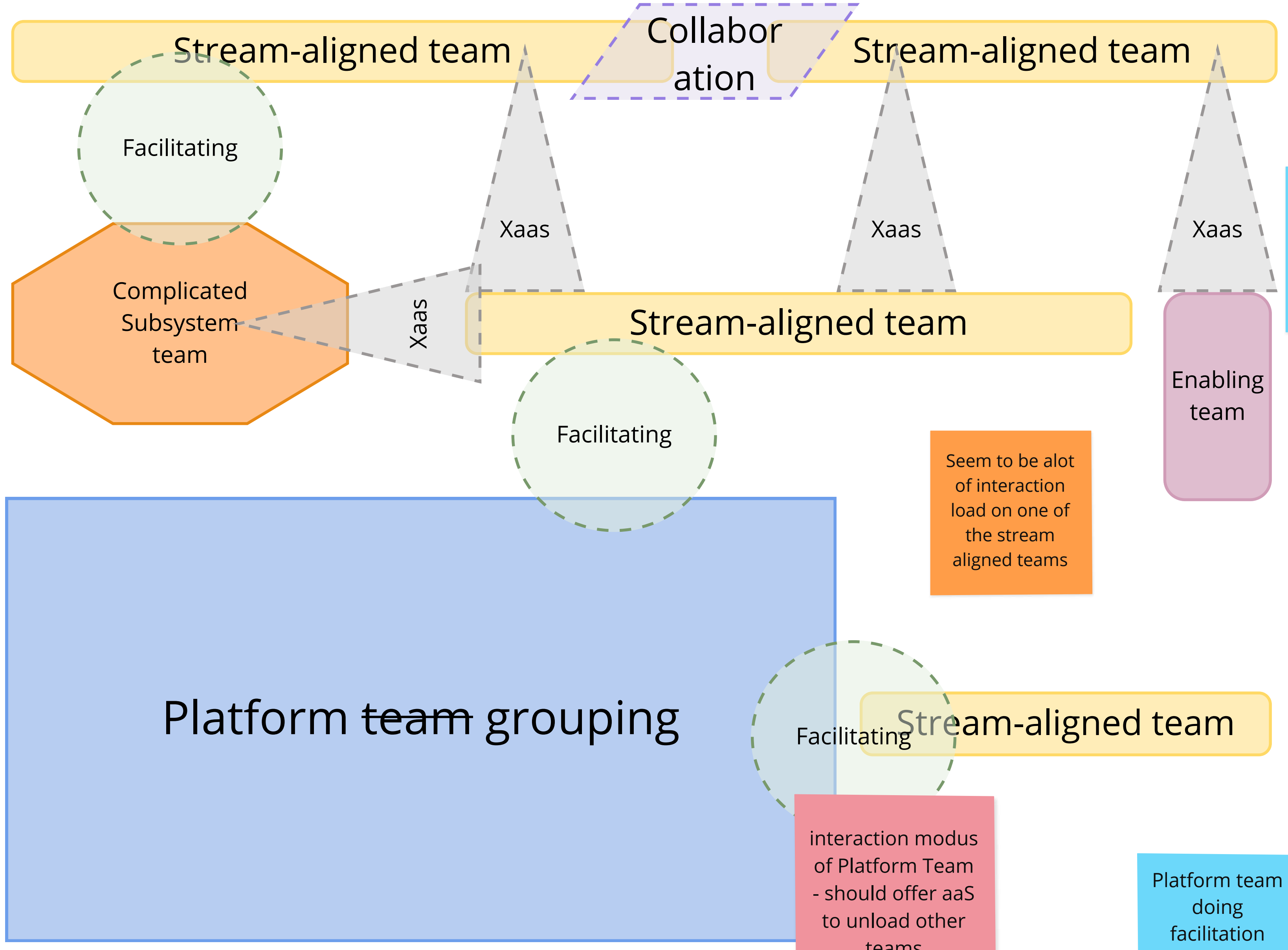
Not clear what and to whom the P-Teams are servicing SAT

no E2E due / functional handovers

Everything is silo. Not flowing

No flow of change, platform teams not supportiong others, not clear what platform are supporting, all separate services

Team interaction diagrams - bad example 3



Stream-aligned team

Collaboration

Stream-aligned team

Facilitating

Xaas

Xaas

Xaas

Complicated Subsystem team

Xaas

Stream-aligned team

Enabling team

Facilitating

Platform ~~team~~ grouping

Facilitating

Stream-aligned team

Seem to be alot of interaction load on one of the stream aligned teams

interaction modus of Platform Team - should offer aaS to unload other teams

Platform team doing facilitation instead of XaaS

Stream aligned team depending on another stream aligned team

Collaboration as a hand-over in the flow

stream aligned team handover + colloboration??

stream aligned team sproviding xaas directly

stream aligned teams not E2E

facilitations done by platform teams instead of providing services

subsytem team has both facilitating and XaaS modes

real life example :-)

enabling team not aligned with stream aligned team,..also looks handover

Platform team should be XaaS

worng interaction modes,

Enabling team disconnected from streams

Enabling team with Xaas interaction?

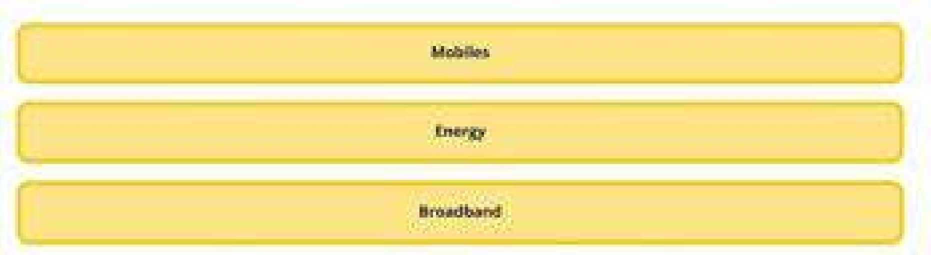
Enabling team providing XaaS

Xaas team ?

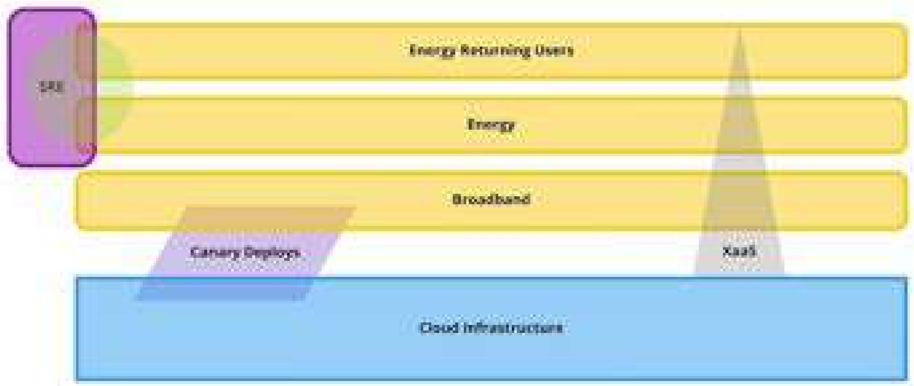
Platform grouping should provide Xaas for other team

Service provided backward (XaaS from stream aligned team to CST)

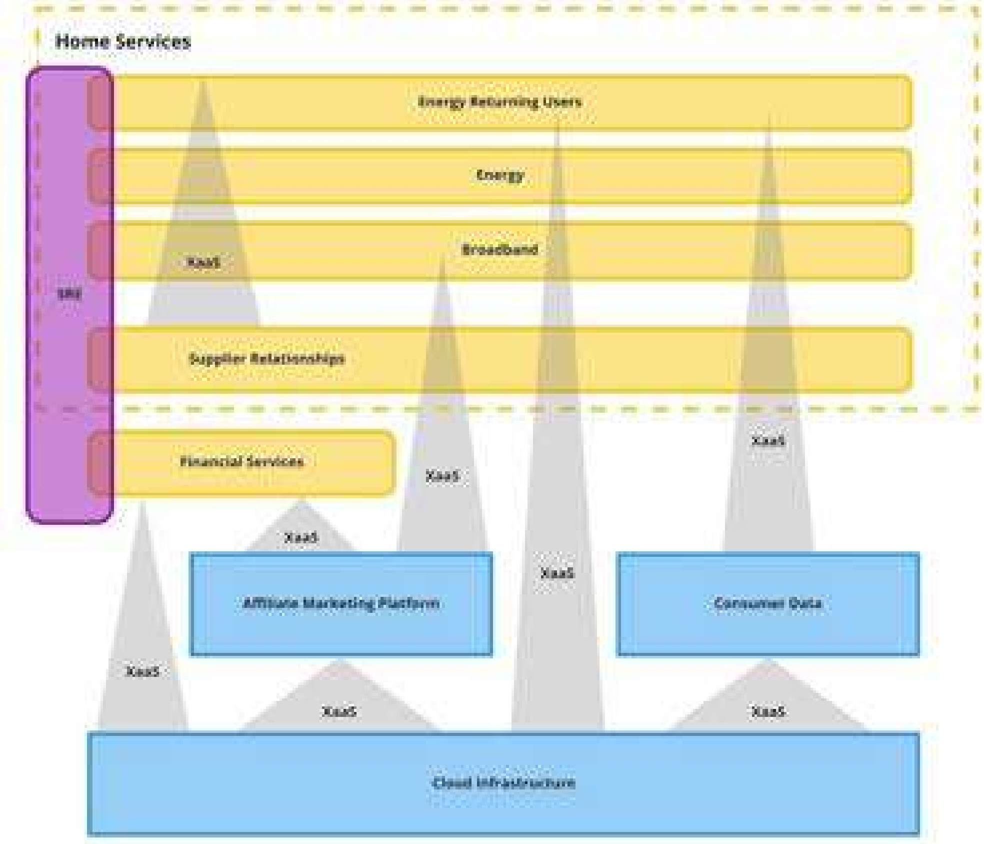
Team interaction diagrams - 1: Uswitch



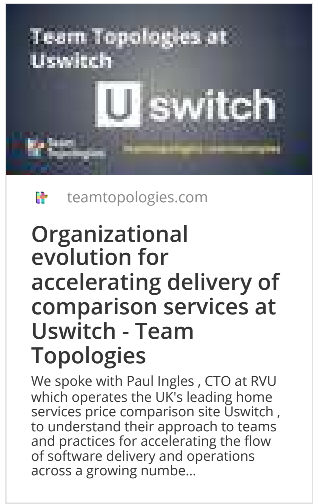
1



2



3


Team Topologies at Uswitch

teamtopologies.com

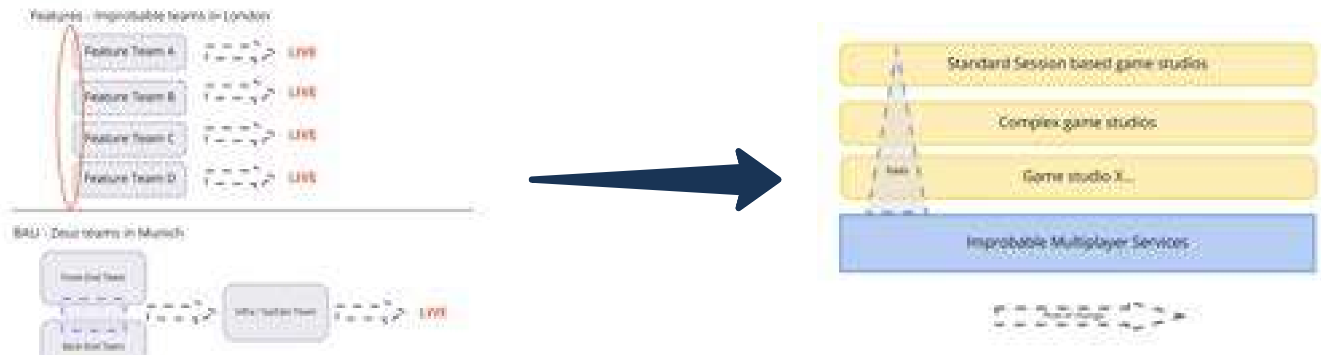
Organizational evolution for accelerating delivery of comparison services at Uswitch - Team Topologies

We spoke with Paul Ingles , CTO at RVU which operates the UK's leading home services price comparison site Uswitch , to understand their approach to teams and practices for accelerating the flow of software delivery and operations across a growing numbe...

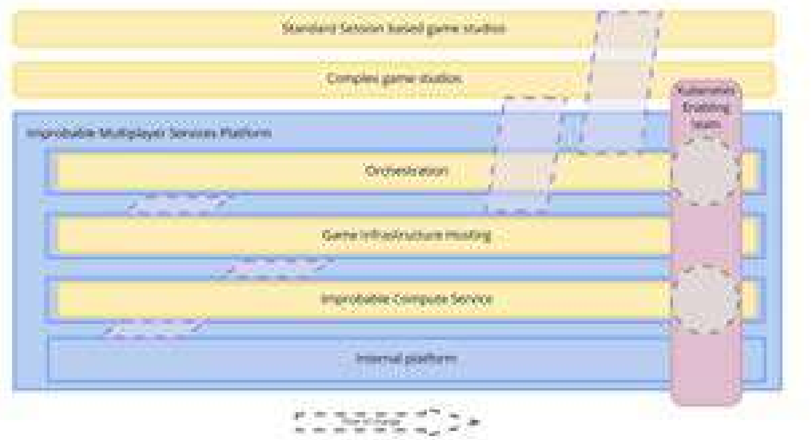
https://teamtopologies.com/industry-examples/organizational-evolution-accelerating-delivery-of-comparison-services-uswitch

conflux

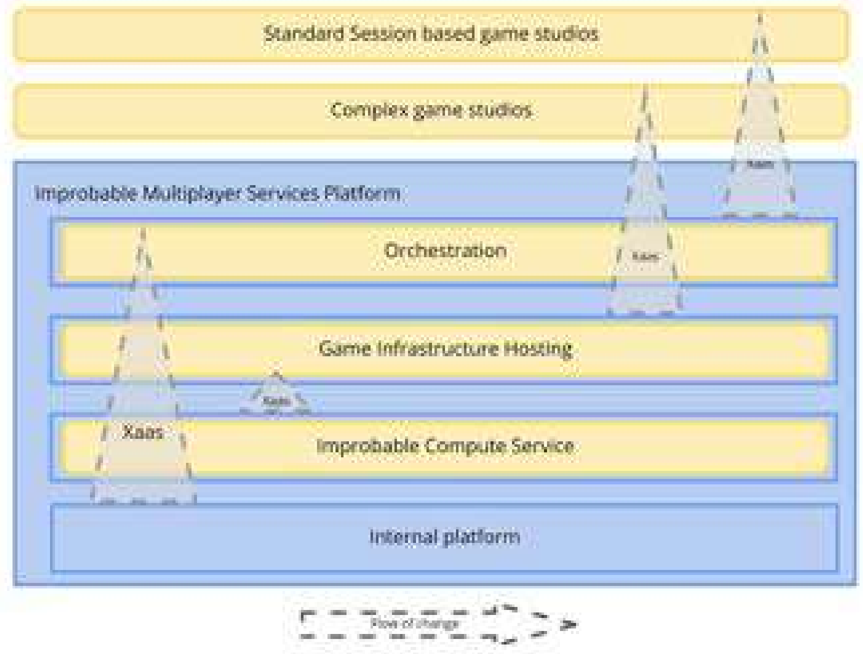
Team interaction diagrams - 2: Improbable (VR)



1


Standard Session based game studios
Complex game studios
Kubernetes Enabling team
Improbable Multiplayer Services Platform
Orchestration
Game Infrastructure Hosting
Improbable Compute Service
Internal platform
Flow of change

2


Standard Session based game studios
Complex game studios
Xaas
Improbable Multiplayer Services Platform
Orchestration
Xaas
Game Infrastructure Hosting
Xaas
Xaas
Improbable Compute Service
Internal platform
Flow of change

3

Virtual Worlds: using Team Topologies at Improbable

IMPROBABLE

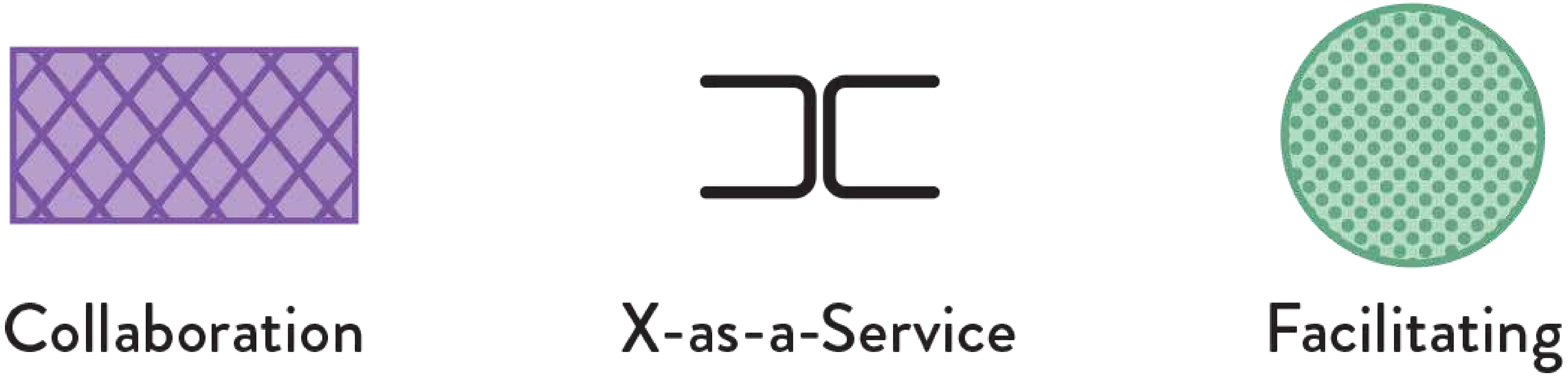
teamtopologies.com

Virtual Worlds: using Team Topologies at Improbable to transform teams, technology, reliability, and customer satisfaction - Team Topologies

Founded in 2012, Improbable is a British technology company, dedicated to solving the challenges of building rich virtual worlds and pioneering the path to the metaverse. ... In 2020 Improbable acquired Munich-based video games company Zeuz, a managed h...

https://teamtopologies.com/industry-examples/virtual-worlds-using-team-topologies-at-improbable-to-transform-teams-technology-reliability-and-customer-satisfaction

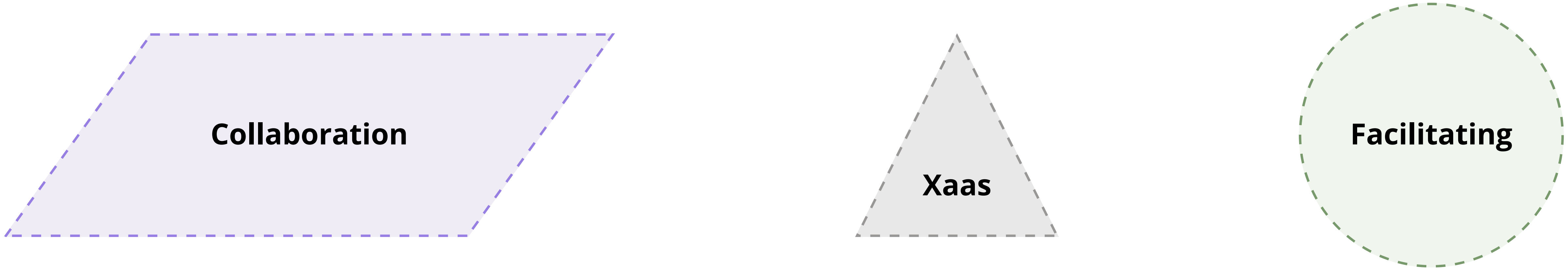
Team interaction modes



Taken from *Team Topologies* (2019). Figure 7.2

Team interaction modes 2

Using the digital TT team modeling shapes



https://shapes.teamtopologies.com/

Purpose of the team interaction modes - Collaboration

Collaboration

Co-creation, co-learning

Teams working for a finite period of time

Teams needs each others capabilties to collaborate to make a new service

not e2e - collaboration/ knowledge sharing for a period of time

comes at a cost (synchronisation / dependency mgt)

co-creation

Need to align schedule with each other

High colloboration, accelerated learning mode
e.g. mobiing on new service, use of new technology

Collaboration - suggested points

Agreeing the interface

Finding the right boundary for flow

Purpose of the team interaction modes - XaaS

XaaS, is about consuming a service without any collaboration needed.

unload Stream Aligned teams

allow faster flow

consume service with minimal interaction

👍 1

Simplify way of working for common/"commodity" capabilities

Provide self-service to others. Onboard others to the Platform or service

Xaas

Unblock Stream Aligned.
Inject Knowledge.
Lower cognitive load

Enable stream aligned teams to use the service/capability in an efficient way e.g reduce cognitive load

Limit the number of people thinking about specific implementation details

Minimal interaction

Reduce cognitive load

better way to collaborate on a extended time frame

Escapsulate the permission for management of specfic resources

Provide service for other team. Therefore, they don't need to care about the detail

reduce cognitive load of Stream Aligned Team

XaaS - suggested points

Minimise hand-offs

Limit team cognitive load in the teams using the service

Allow teams to self-serve

Purpose of the team interaction modes - Facilitating

Reducing cognitive load

Catalyse where needed

provide skills

Not full-time collaboration

Facilitation had end date

Helping to grow and learn and be independent

Facilitating

upskill, learning

make yourself redundant?

Temporary / upskill

Bridging a knowledge gap

help teams with new capabilities bridging a knowledge gab

Teaching

Cross learning to other domains

knowledge sharing for a shorter period of time

Facilitating - suggested points

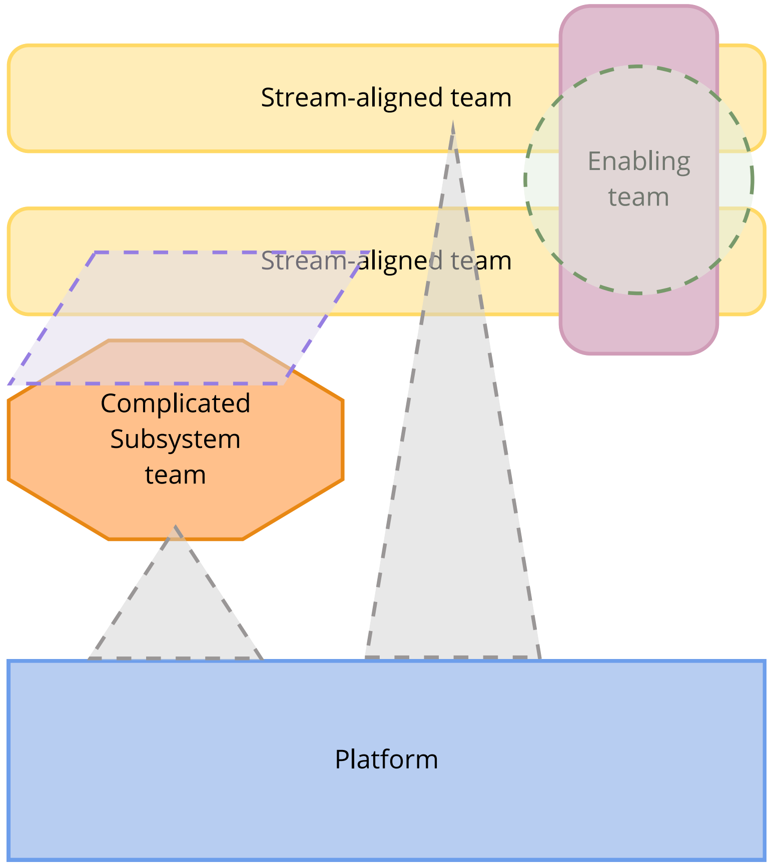
Skill up teams

Identify gaps in knowledge across teams

Act as a multiplier

Implications of the team interaction modes

Think about: ways of working, evolution over time, team mindsets for success



Principles

Collaboration and Facilitating are brief (temporary) interactions

The goal typically is XaaS

Careful to not introduce new bottlenecks because of building up dependencies

Some persons of Enabling team can be also in the Platform Team?

enabling team facilitating mode - defined period

Clear goal, success criteria ie: why will teams want to move to XaaS

how do we move subsystem team to XaaS

focus on flow mindset of continuous improvement Willing to adapt to change

We should (must?) train the teams on interactions

Consider size of the organisation vs value/cost of XaaS / platform

Location of team members

Investigate certain if tasks are really different or need to be different

Enabling: Mindset more teaching, not doing for you

Clarity about team communication

clearly target the goal & area you want to improve

Teams should be ready for their preferred interaction mode

Change collabaration of Compicate subsystem team into Xaas

Merge compicated subsystem team into Platform Grouping or create new platgorm grouping for them

learning curve (as an organistaion) drives evolution in interaction

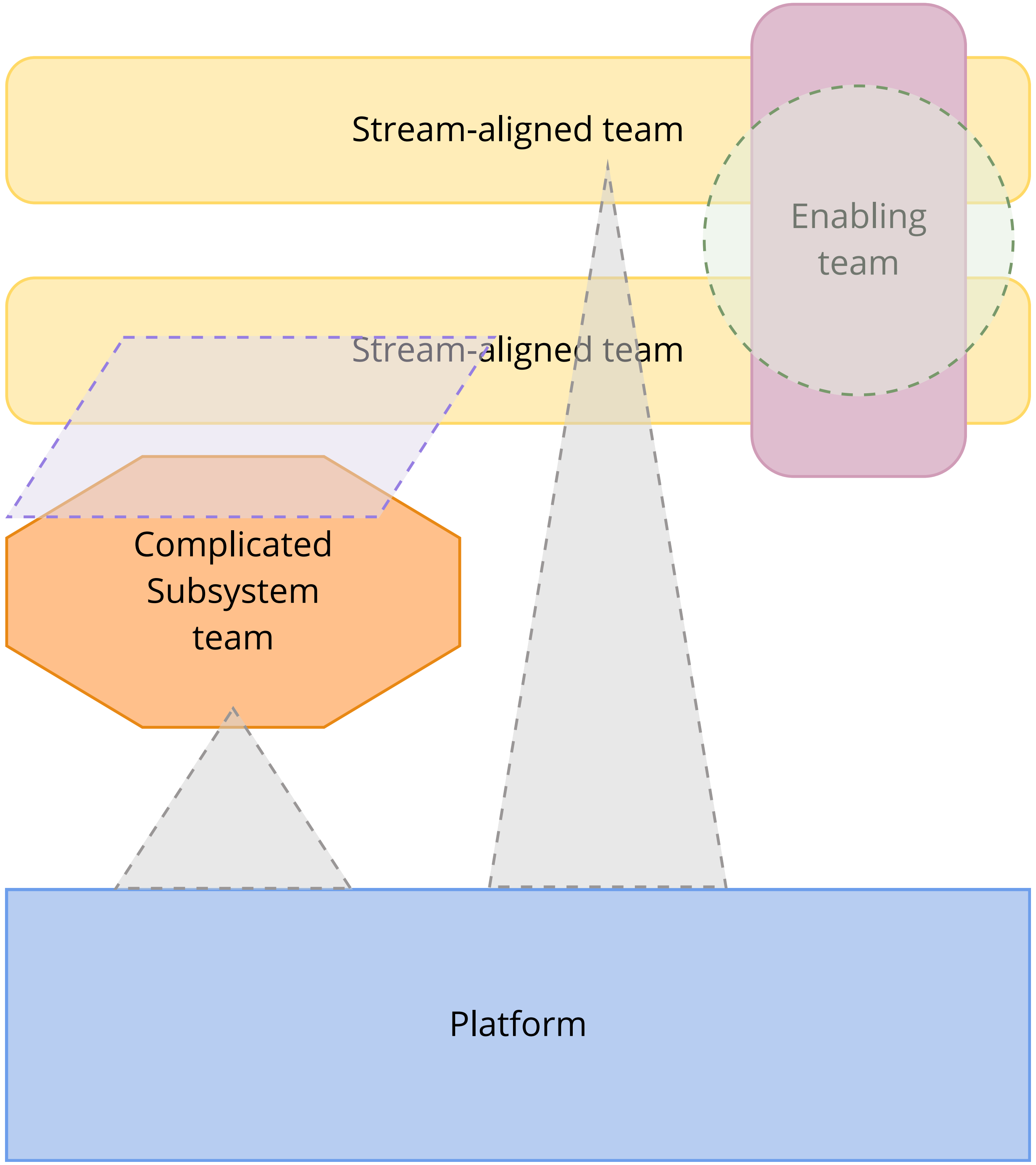
Enabling: Pull not push

clarity on interactions across teams

Expectation management about interaction duration

How to move from current collaboration to Xaas

Implications of the team interaction modes



All TT diagrams are a snapshot in time, not a fixed end state

XaaS *must* be preceded by Collaboration

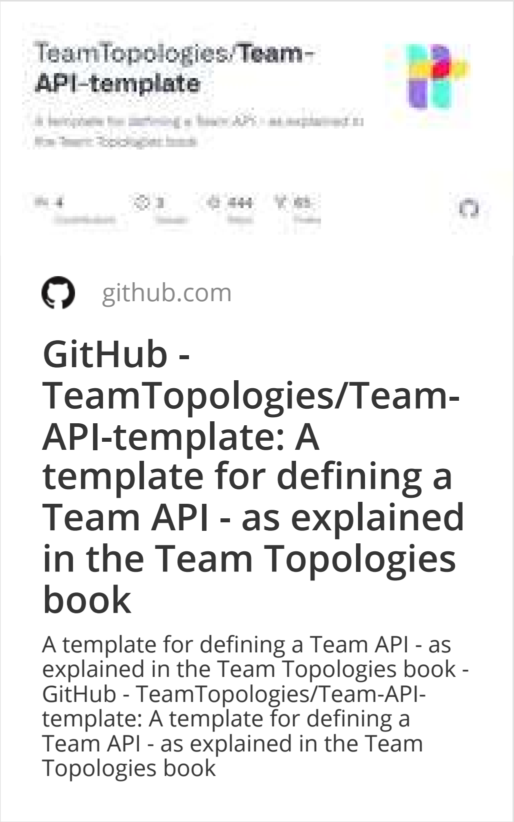
We need to ask "what is the purpose of the Collaboration / Facilitating interaction?"

... and "how long will this interaction last?"

conflux

Team API - overview

"a Team API ... is a description and specification for how to interact ... with the team."



TeamTopologies/Team-API-template

github.com

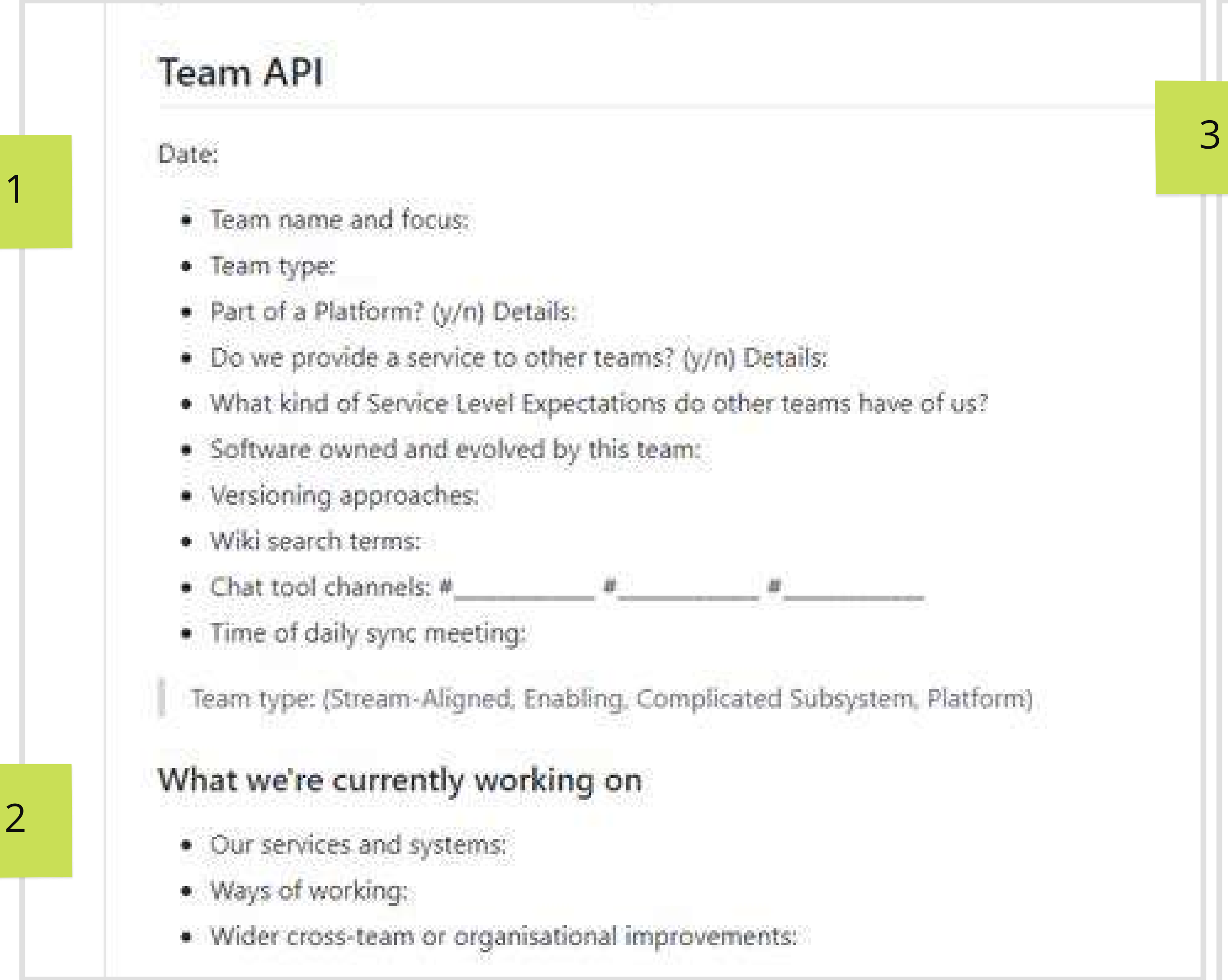
GitHub - TeamTopologies/Team-API-template: A template for defining a Team API - as explained in the Team Topologies book

A template for defining a Team API - as explained in the Team Topologies book - GitHub - TeamTopologies/Team-API-template: A template for defining a Team API - as explained in the Team Topologies book

https://github.com/TeamTopologies/Team-API-template

1

2



Team API

Date:

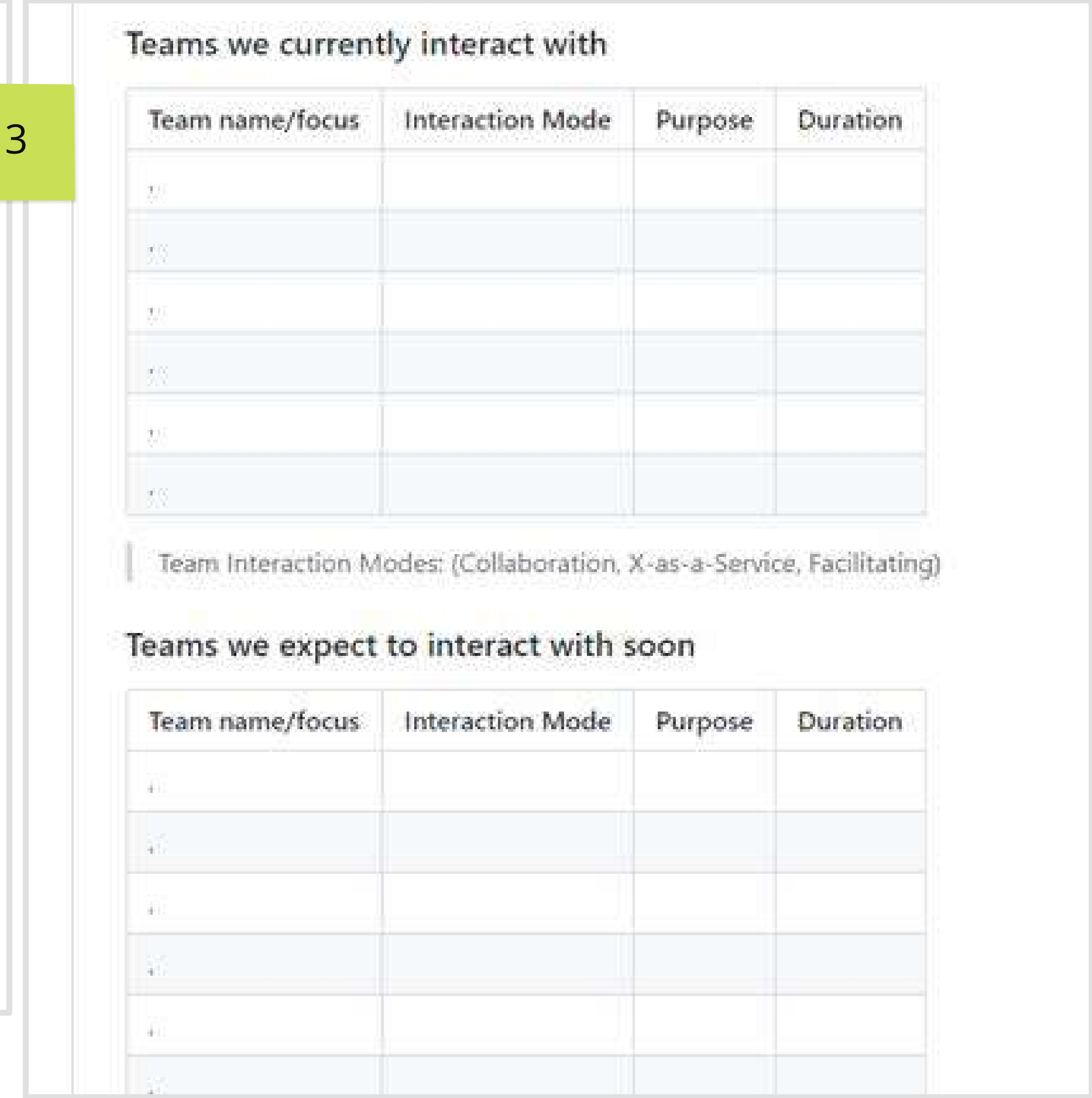
- Team name and focus:
- Team type:
- Part of a Platform? (y/n) Details:
- Do we provide a service to other teams? (y/n) Details:
- What kind of Service Level Expectations do other teams have of us?
- Software owned and evolved by this team:
- Versioning approaches:
- Wiki search terms:
- Chat tool channels: #__________ #__________ #__________
- Time of daily sync meeting:

> Team type: (Stream-Aligned, Enabling, Complicated Subsystem, Platform)

What we're currently working on

- Our services and systems:
- Ways of working:
- Wider cross-team or organisational improvements:

3



Teams we currently interact with

Team name/focus	Interaction Mode	Purpose	Duration

> Team Interaction Modes: (Collaboration, X-as-a-Service, Facilitating)

Teams we expect to interact with soon

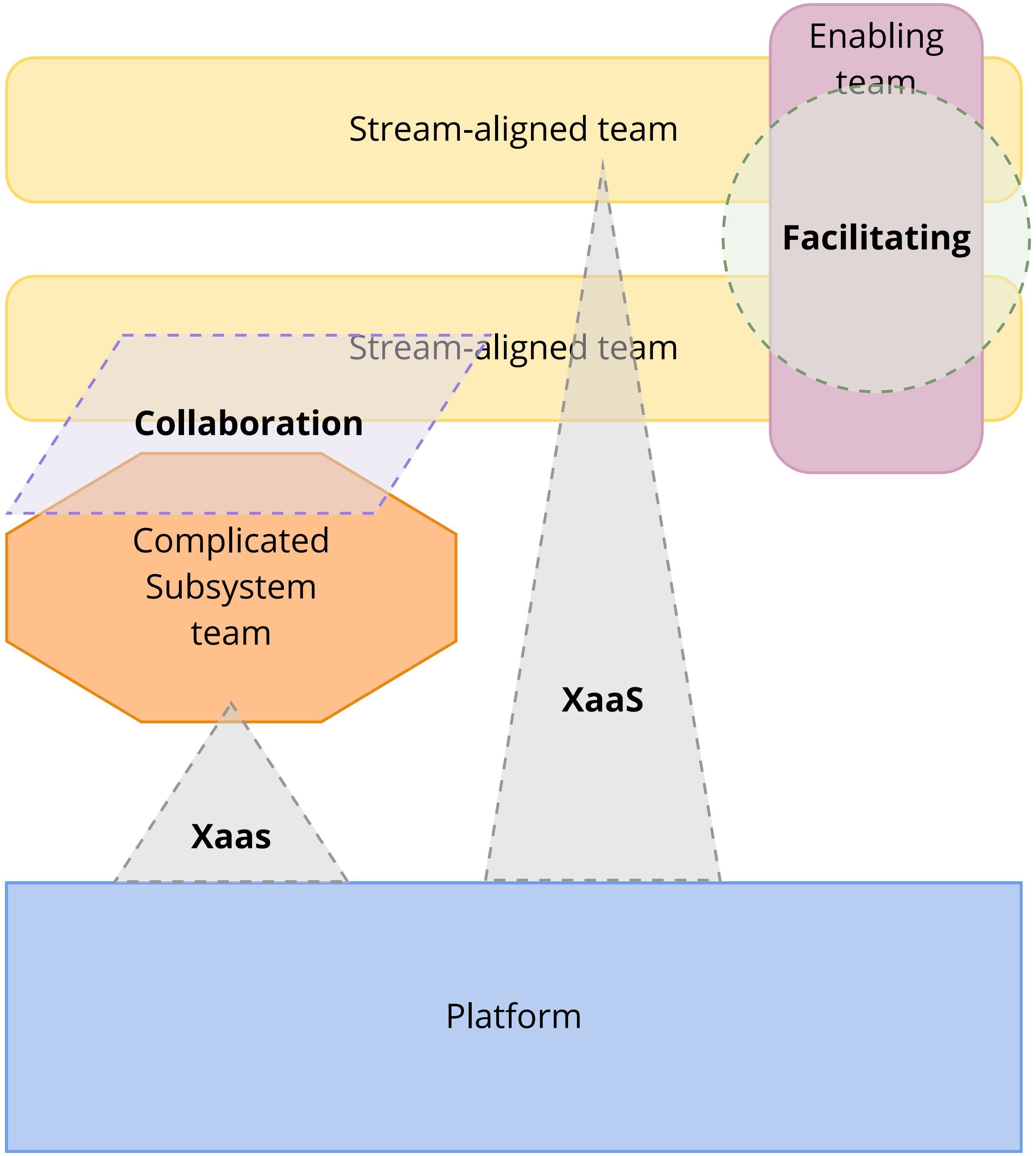
Team name/focus	Interaction Mode	Purpose	Duration

Focus on the intent of these questions

Ignore the specific format of this tool

This is not about documentation

Organizational sensing: awkward interactions - overview

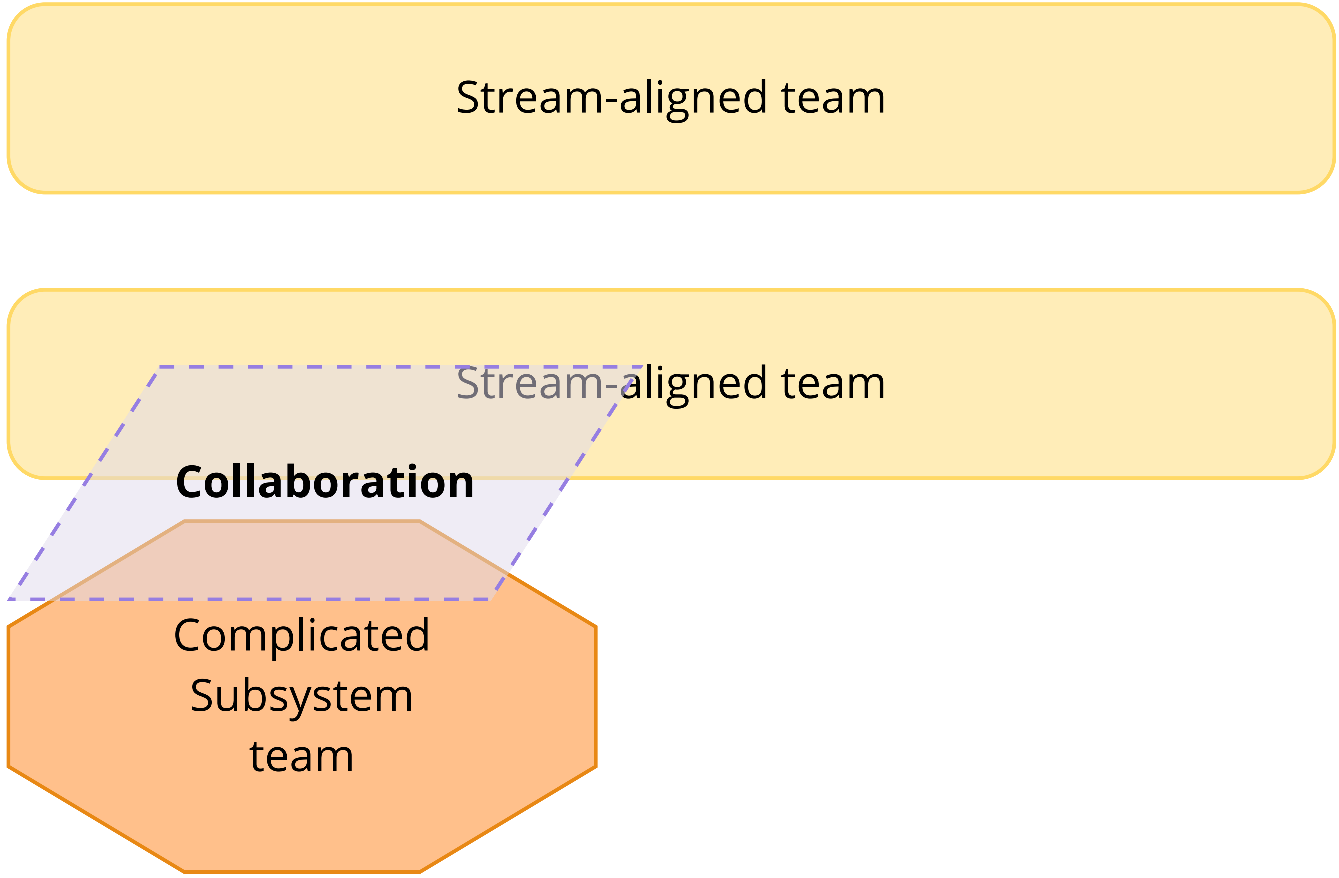


How could you detect that there is something wrong with a current team interaction?

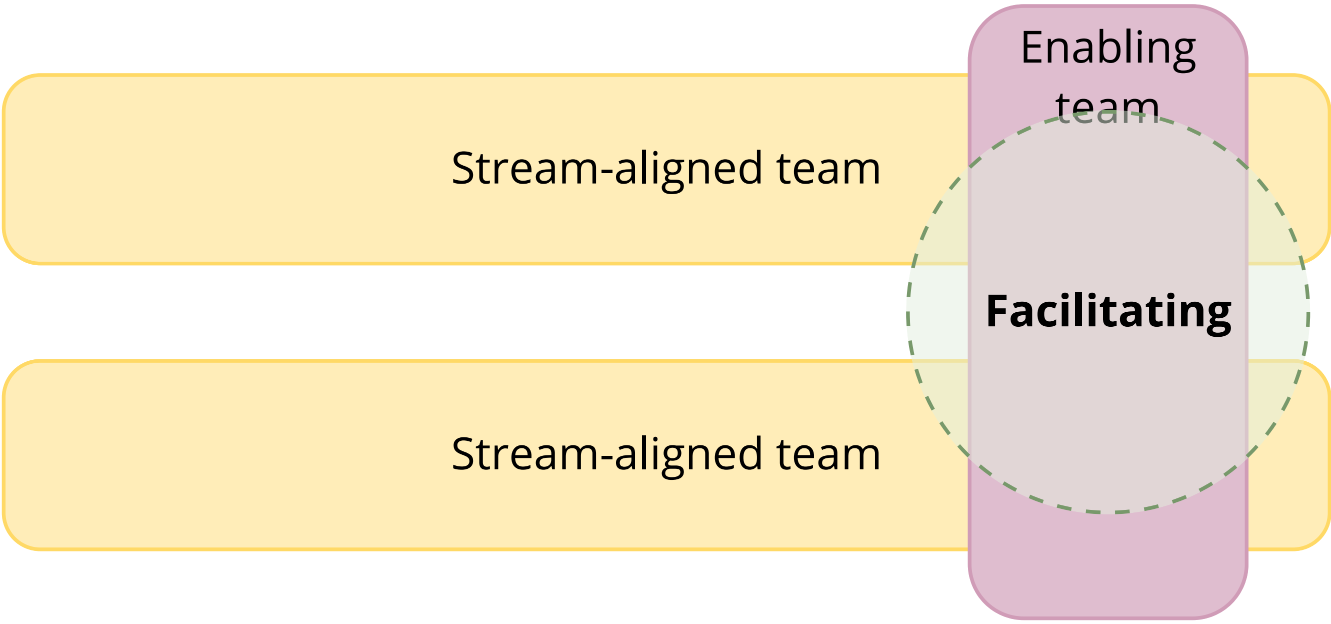
Derive some heuristics (clues) for using team interaction modes for organizational sensing

conflux

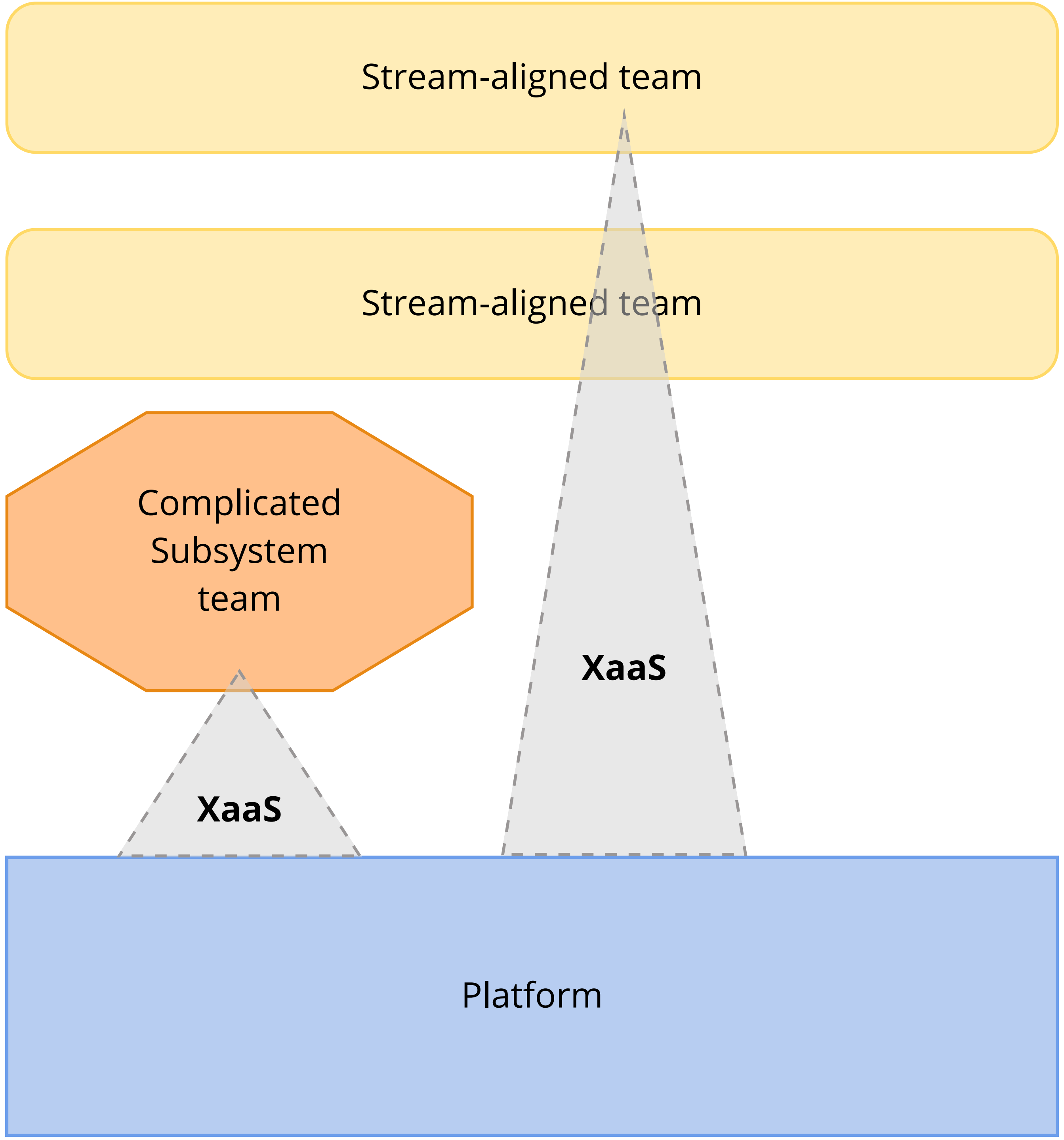
Organizational sensing: awkward interactions 2



Organizational sensing: awkward interactions 3

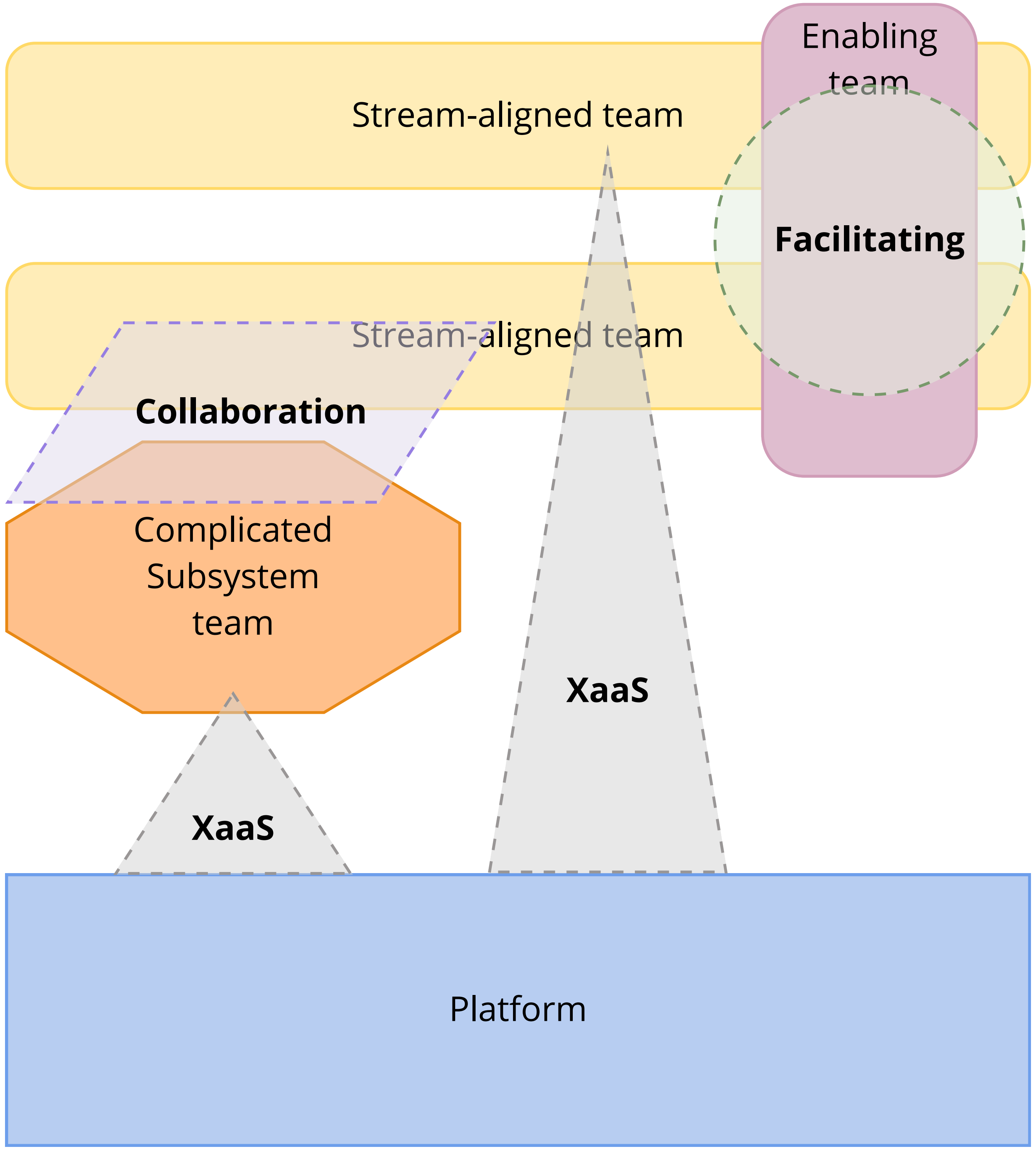


Organizational sensing: awkward interactions 4



conflux

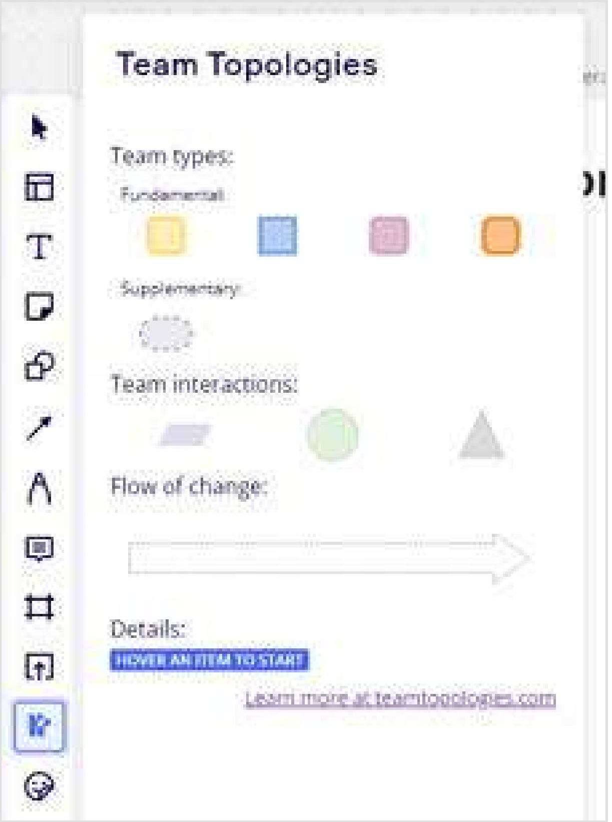
Organizational sensing: heuristics



PROBLEMS TO LOOK FOR

HEURISTICS TO RECOGNIZE AWKWARD INTERACTIONS

Evolution of teams and interactions



Team Topologies

Team types:

Fundamental:

Supplementary:

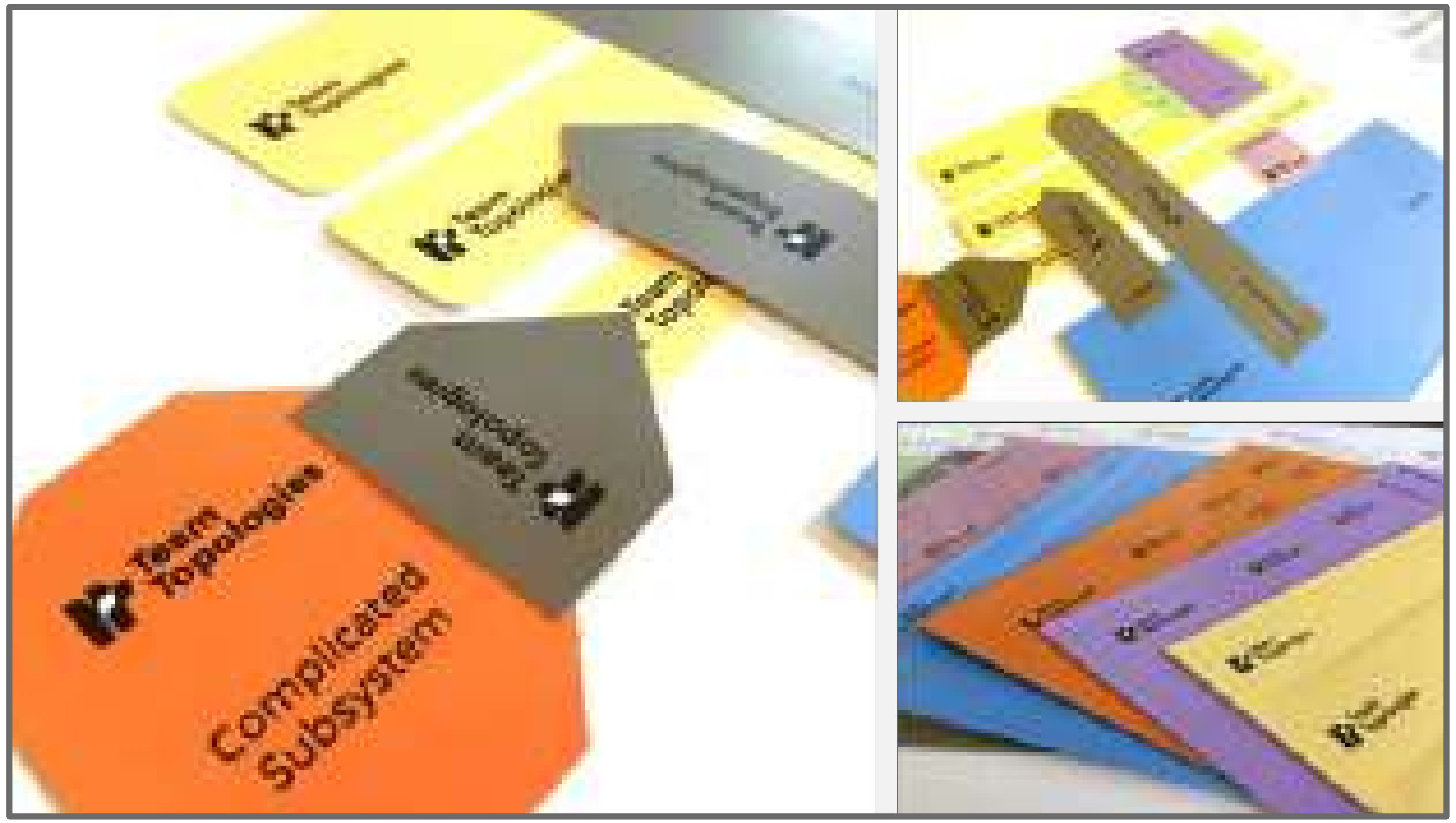
Team interactions:

Flow of change:

Details:

HOVER AN ITEM TO START

Learn more at teamtopologies.com



https://agilestationery.com/products/modeling-shapes-for-team-types-and-team-interactions

teamtopologies.com

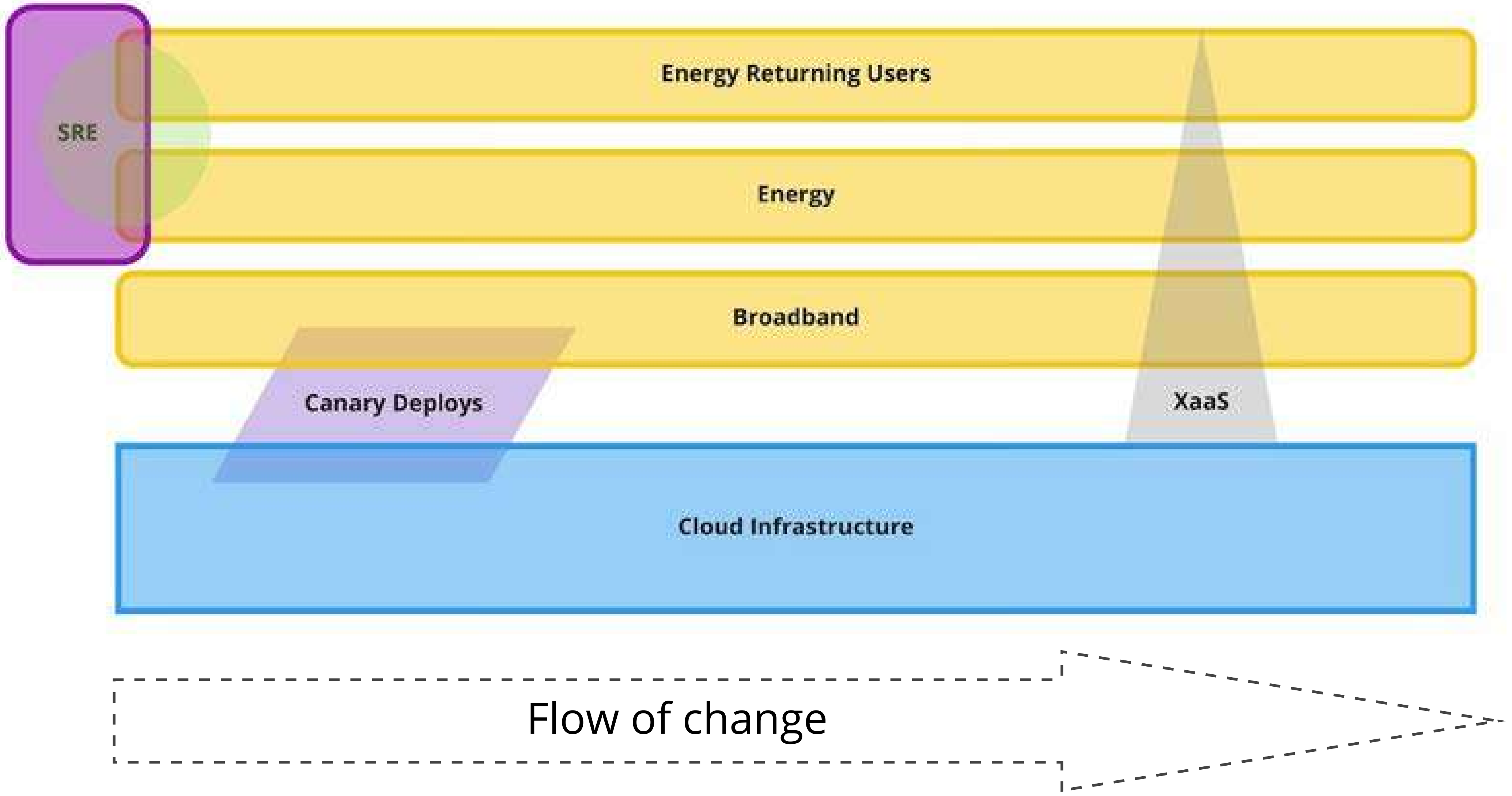
Team Interaction Modeling with Team Topologies - Team Topologies

Team Topologies provides a combination of principles and practices that can help structure and evolve an organization for effective collaboration, autonomy, delivery focus, and product alignment. Using the Team Topologies shapes and diagramming principl...

https://teamtopologies.com/tim

TT team modeling shapes - key points

Online



✓ No handover

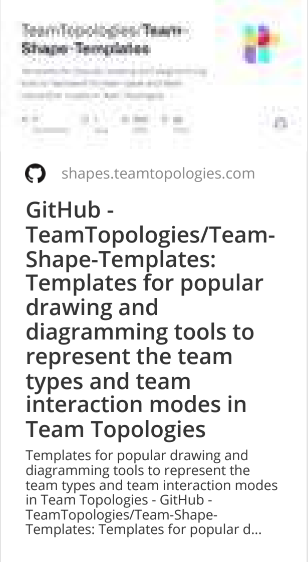
✓ Flow arrow

✓ Correct interaction modes

...

https://teamtopologies.com/industry-examples/organizational-evolution-accelerating-delivery-of-comparison-services-uswitch

💡 This shows both team and system architecture


TeamTopologies/Team-Shape-Templates

shapes.teamtopologies.com

GitHub - TeamTopologies/Team-Shape-Templates: Templates for popular drawing and diagramming tools to represent the team types and team interaction modes in Team Topologies

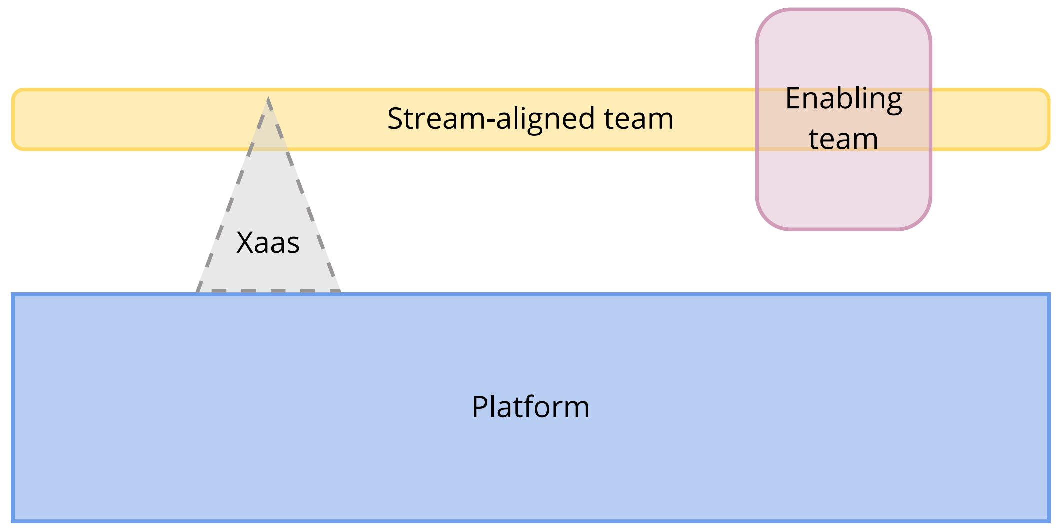
Templates for popular drawing and diagramming tools to represent the team types and team interaction modes in Team Topologies - GitHub - TeamTopologies/Team-Shape-Templates: Templates for popular d...

http://shapes.teamtopologies.com/

Evolution scenario 1 - meet compliance requirements

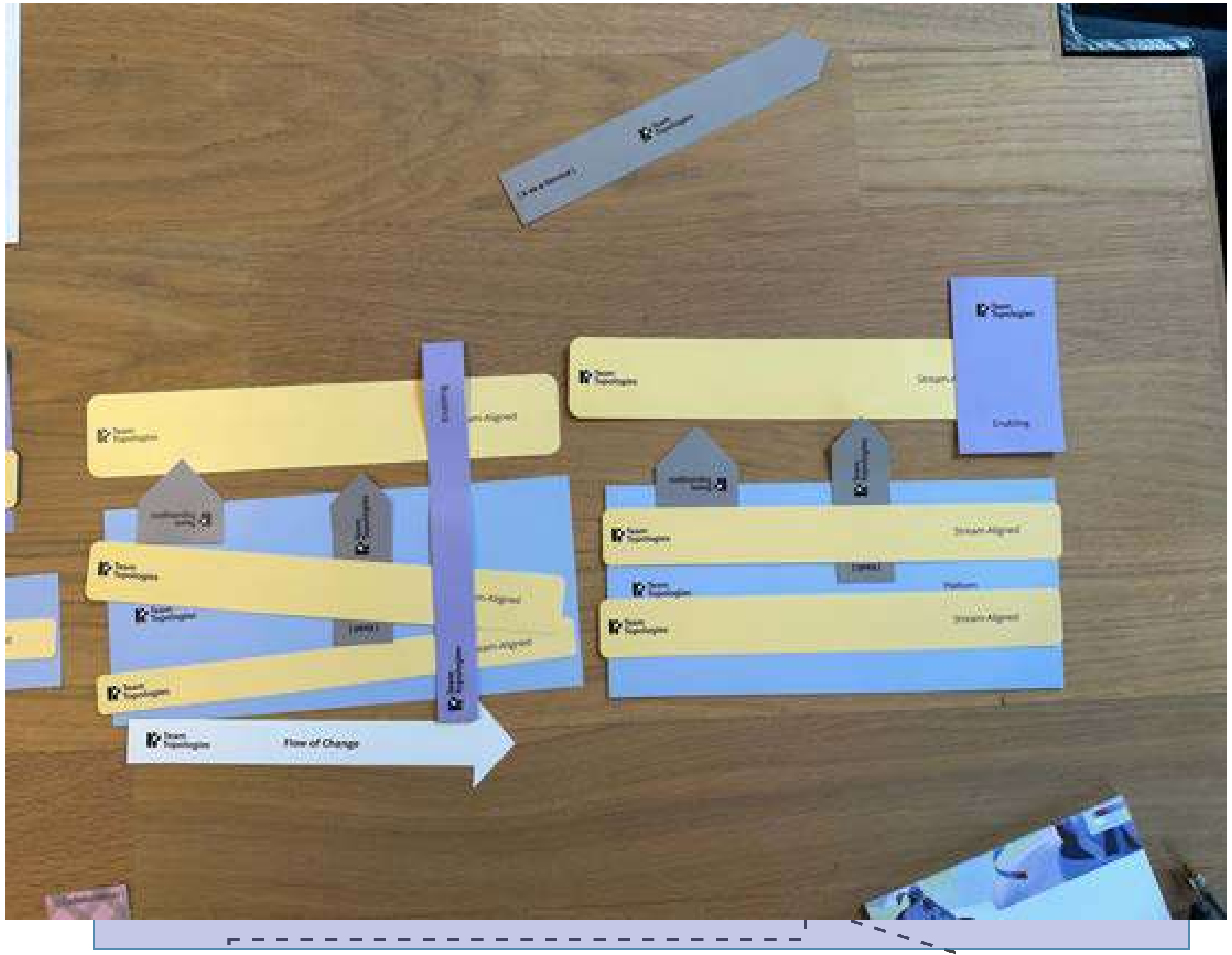
Group 5

Stage 1 - minimal compliance requirements



Flow of change

Stage 2 - compliance needed for customer-facing services and underlying data



Flow of change

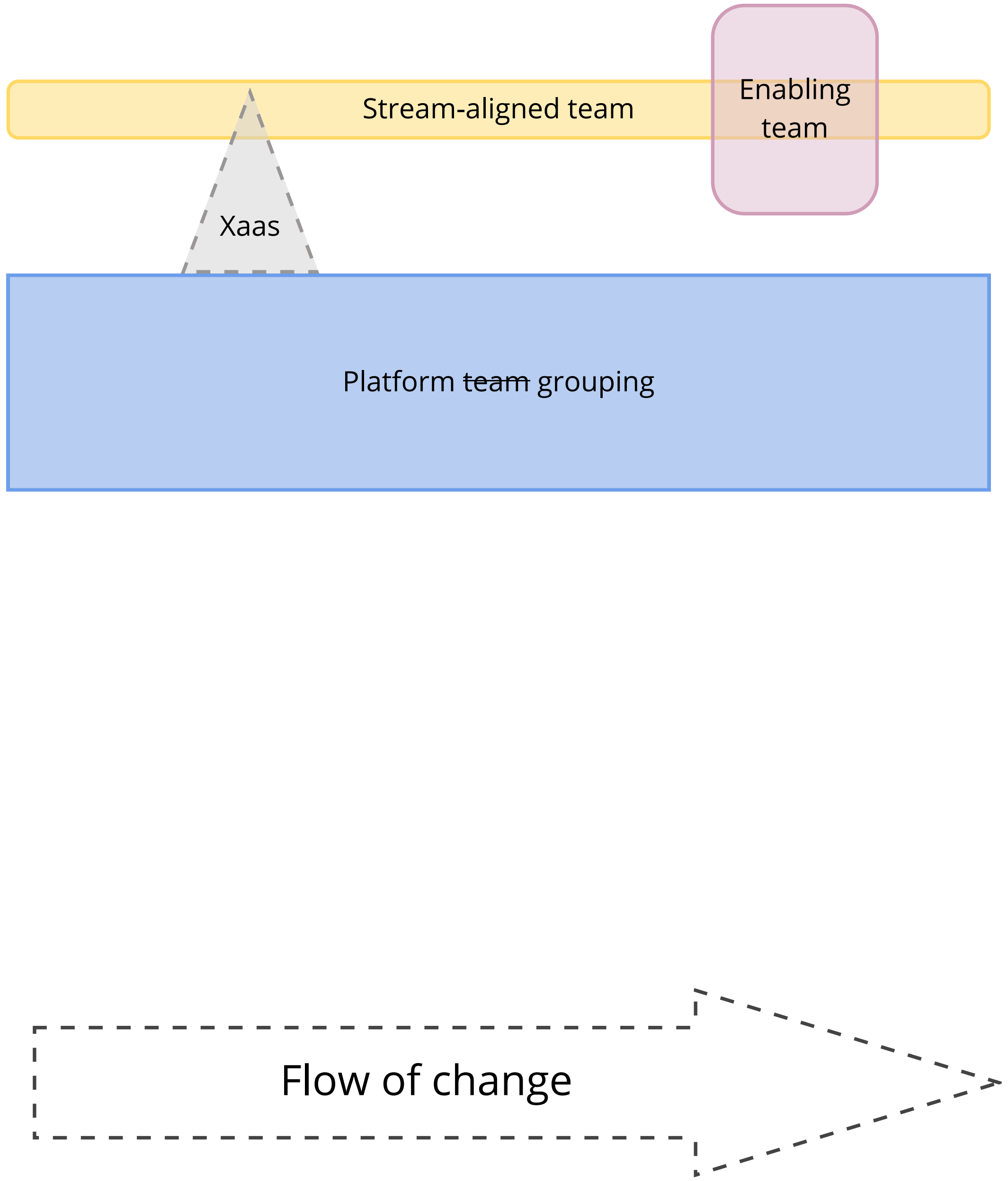
1 - Urgency, implement compliance quickly

2- Maturity, offers compliance validation as a service

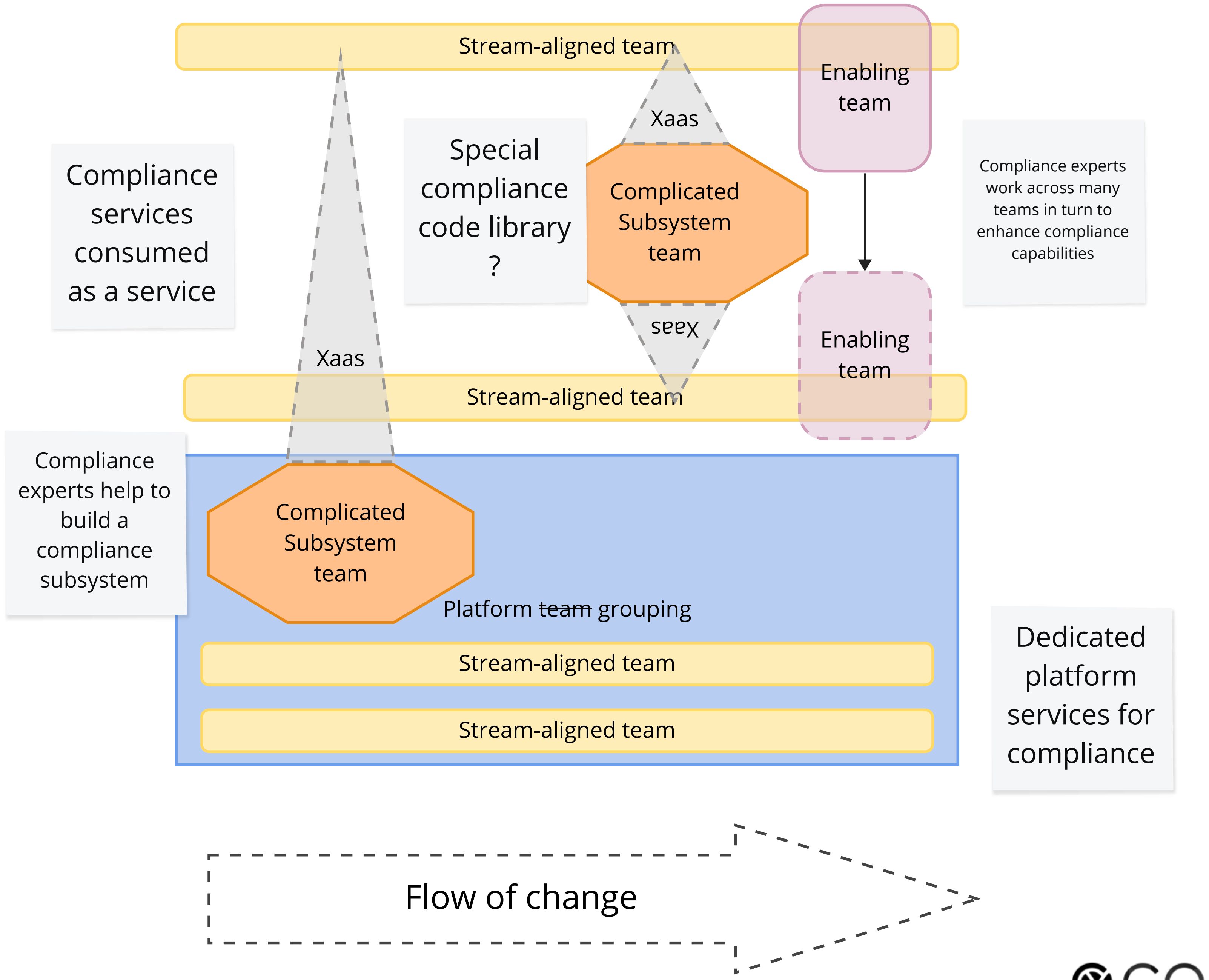
conflux

Evolution scenario 1 - compliance requirements - expert

Stage 1 - minimal compliance requirements



Stage 2 - compliance needed for customer-facing services and underlying data



Compliance services consumed as a service

Special compliance code library ?

Compliance experts work across many teams in turn to enhance compliance capabilities

Compliance experts help to build a compliance subsystem

Dedicated platform services for compliance

Part 2 - Platform thinking for fast flow

conflux

Outline of part 2

- We begin by reviewing the purpose of a platform from a TT perspective and establish some useful metrics and attitudes for platform success.
- We take a deep dive into the implications of “fractal” (self-similar, nested) platforms as defined by TT.
- We then explore some advanced patterns for platforms around multiple parallel product/service offerings, sharing or “harvesting” proven solutions, horizon scanning using Core Domain Charts, data consumption (touching on techniques like Data Mesh) and product composition.
- We finish Part 2 by looking at an emerging approach that we call Desynchronous that helps organizations to scale without the slow-downs associated with large internal platforms of the past.

The purpose of a platform + attitudes for success

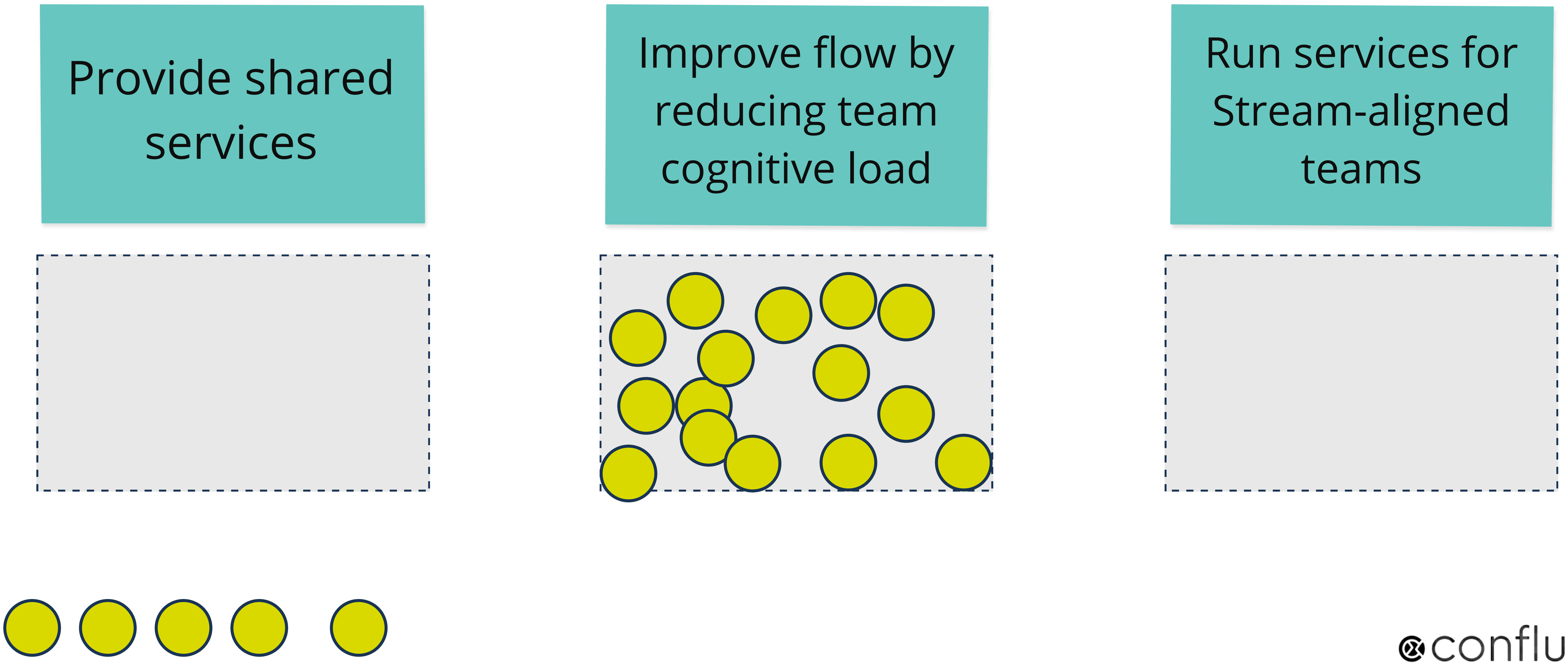
"The purpose of a platform ... is to enable stream-aligned teams to deliver work with substantial autonomy."

Team Topologies (2019), page 92

The purpose of a platform

How does a TT platform actually enable Stream-aligned teams to deliver with substantial autonomy?

Choose the *best* answer



Provide shared services

Improve flow by reducing team cognitive load

Run services for Stream-aligned teams

Metrics for a TT platform

What metrics should a flow-centric platform track in order to be true to its mission?

> ***"The purpose of a platform ... is to enable stream-aligned teams to deliver work with substantial autonomy."***

DORA

DORA 4KM

average cycle time

"custoemer satisfaction" by the Stream Aligned team

NPS on used services

Amount of consumers per service

Dev happines

cycle time

effort score on using the platform

Ease of work score

Ease of use of their services (time / knowledge)

Number of uses of their services

Number of times a team couldn't consume the service in autonomy (nb of tickets opened on a period ?)

Nbr of service consumers

Nr of stream aligned teams onboarded on the platfrom compared tto all

SLO's

Cycle time of the request from stream-align team

None. Introduce diagnostic metrics as needed in response to actual problems.

Backlog size (> not fit for purpose)

cycle time in Stream Aligned team

Vol of interactions with platform team

Flow ditribution Features, Defects, Risks and Debts

cycle time

flow metrics, cycle, lead time

ease of use

Metrics for a TT platform - recommended

Flow	Adoption	UX	Technical
Flow in teams that use the platform	Adoption patterns and barriers	User Experience of "customer" teams: UX, DevEx, NPS, etc.	Reliability, SLO/SLA, response time, etc.
[Flow in teams internal to the platform]			

Mindset for a TT platform

Serving the users/customers of the platform

Product management mindset

Enabling flow

Thinnest Viable Platform - example

A TVP is the smallest set of APIs, documentation, and tools needed to accelerate the teams developing modern software services and systems.

ABC Corp

We use the Serverless Framework to simplify access to AWS serverless services. Use these AWS services via Serverless to build apps for ABC Corp:

- AWS DynamoDB
- AWS Lambda
- AWS S3
- AWS SQS

Use these events to trigger Lambda function execution:

- Application Load Balancer
- DynamoDB
- HTTP API
- S3
- Schedule
- SQS

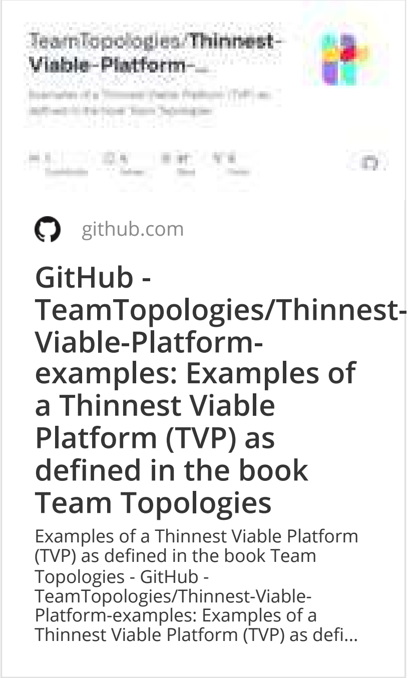
Use these services to monitor Lambda function execution:

- AWS CloudWatch

Use our credentials setup tool, credible, to set up your credentials for the Serverless framework, including all AWS IAM roles and access keys:

- [Link to credible tool]

The platform is literally a wiki page with curated recommendations



TeamTopologies/Thinnest-Viable-Platform-...

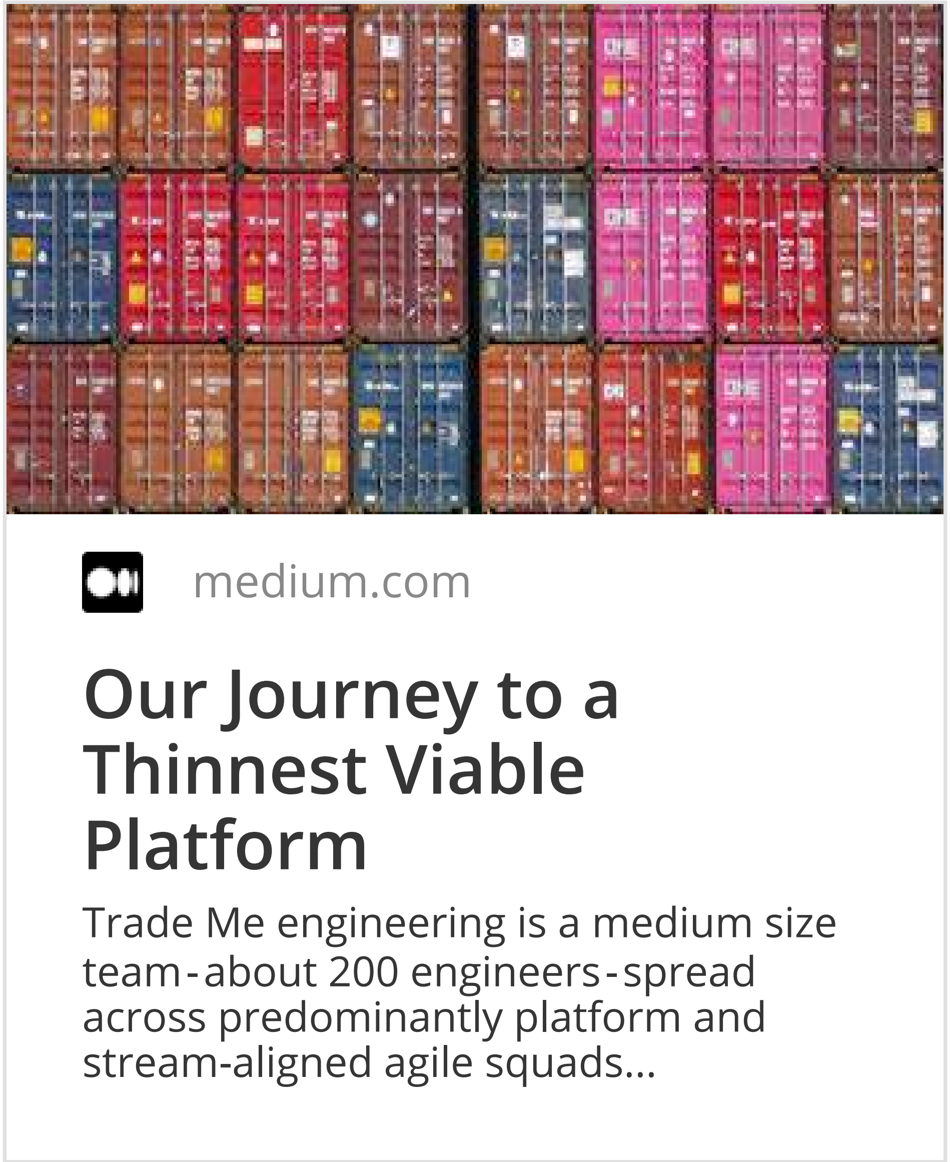
github.com

GitHub - TeamTopologies/Thinnest-Viable-Platform-examples: Examples of a Thinnest Viable Platform (TVP) as defined in the book Team Topologies

Examples of a Thinnest Viable Platform (TVP) as defined in the book Team Topologies - GitHub - TeamTopologies/Thinnest-Viable-Platform-examples: Examples of a Thinnest Viable Platform (TVP) as defi...

https://github.com/TeamTopologies/Thinnest-Viable-Platform-examples

TVP example at Trade Me



medium.com

Our Journey to a Thinnest Viable Platform

Trade Me engineering is a medium size team-about 200 engineers-spread across predominantly platform and stream-aligned agile squads...

https://medium.com/trade-me/our-journey-to-a-thinnest-viable-platform-ca3e57986eb9

"It started with a series of wiki pages highlighting the characteristics of a production-ready application and the definitive list of must-haves we expect applications to have to fulfil our stream-aligned teams' needs. We used user story-mapping to identify the Musts.

Subsequently, it evolved into a templated infrastructure-as-code project with almost fully automated provisioning pipelines."

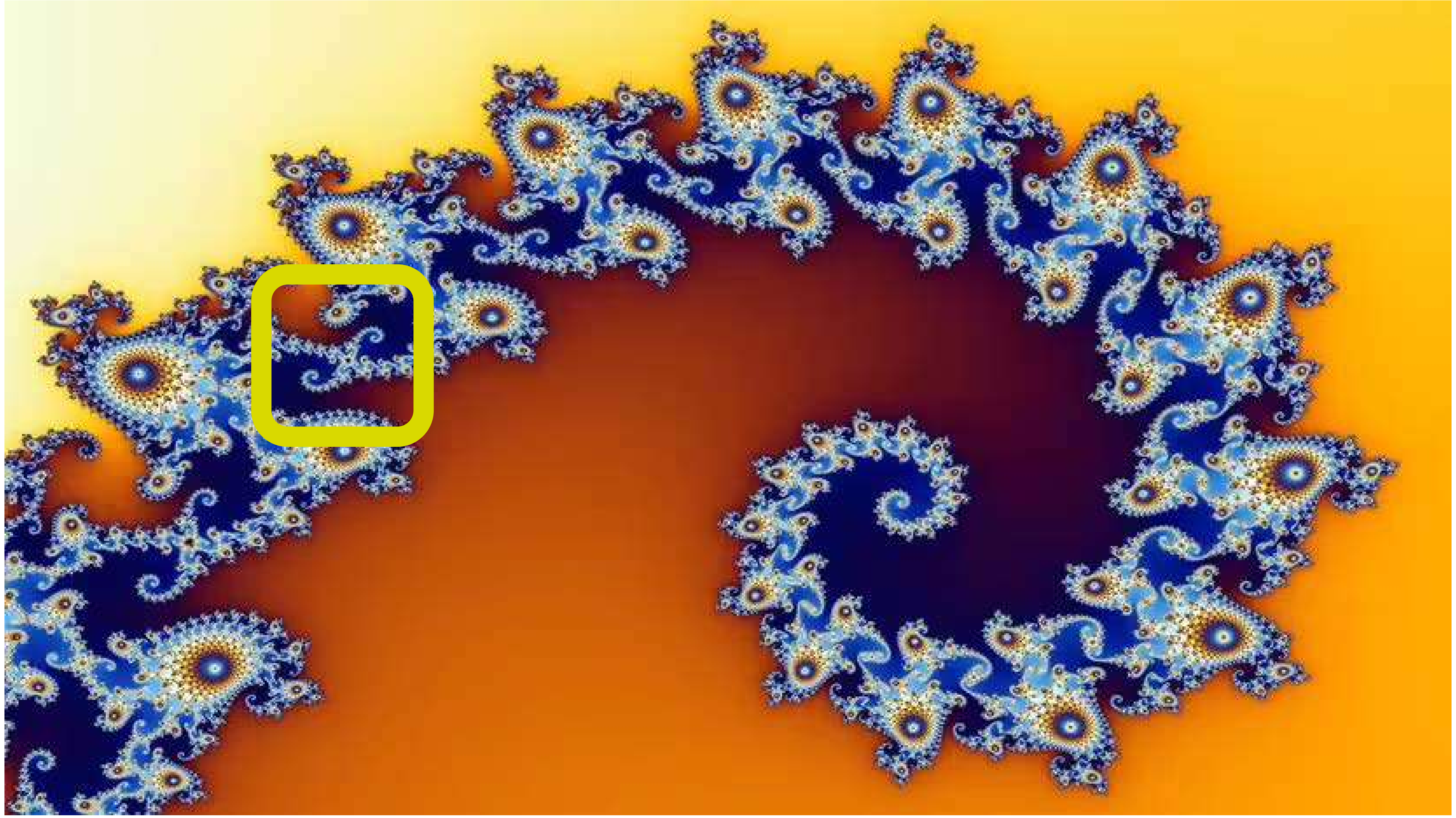
"Our main measures of success (MoS) are:

- Reducing developers' cognitive load (qualitative MoS)
- Time to First Hello World (TTFHW)"

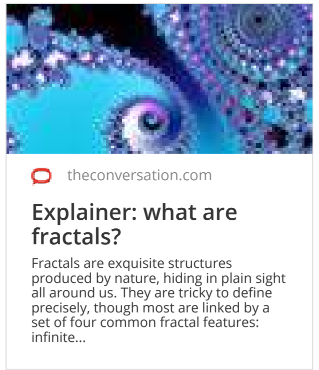
"The intention behind this is to keep the platform as simple as possible to cater to one of its primary purposes: reducing developer cognitive load."

Fractal platforms

Fractal: self-similar at different "zoom levels"



This partial view of the Mandelbrot set, possibly the world's most famous fractal, shows step four of a zoom sequence: The central endpoint of the "seahorse tail" is also a Misiurewicz point.
WOLFGANG BEYER/(CC BY-SA 3.0)
https://science.howstuffworks.com/math-concepts/fractals.htm



theconversation.com

Explainer: what are fractals?

Fractals are exquisite structures produced by nature, hiding in plain sight all around us. They are tricky to define precisely, though most are linked by a set of four common fractal features: infinite...

https://theconversation.com/explainer-what-are-fractals-10865

conflux

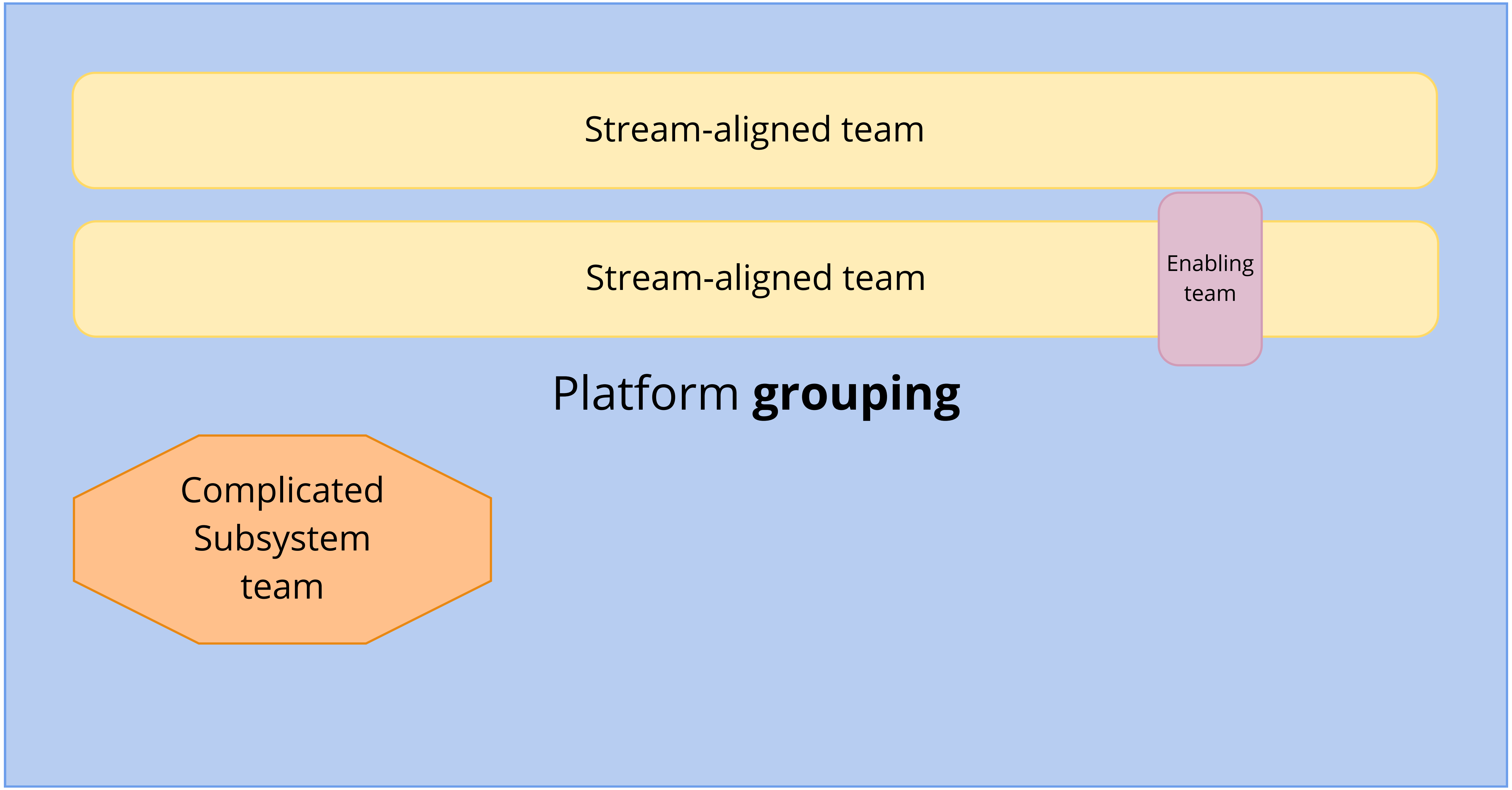
Fractal platforms - platform groupings

Fractal platforms: self-similar at different "zoom levels"

Platform ~~team~~ **grouping**

A platform acts as a "container" for other teams

Fractal platforms - platform groupings 2



Stream-aligned team

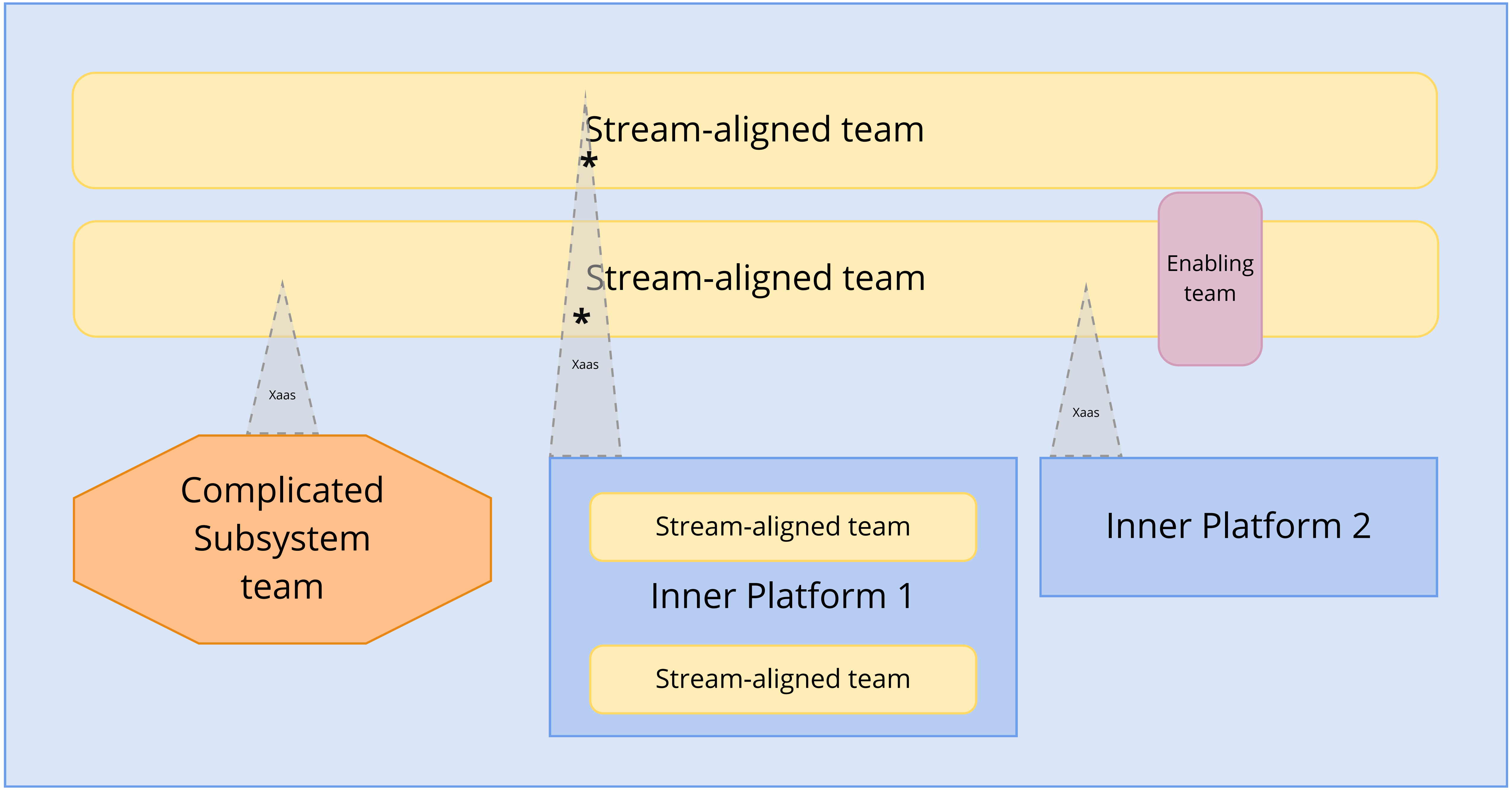
Stream-aligned team

Enabling team

Platform **grouping**

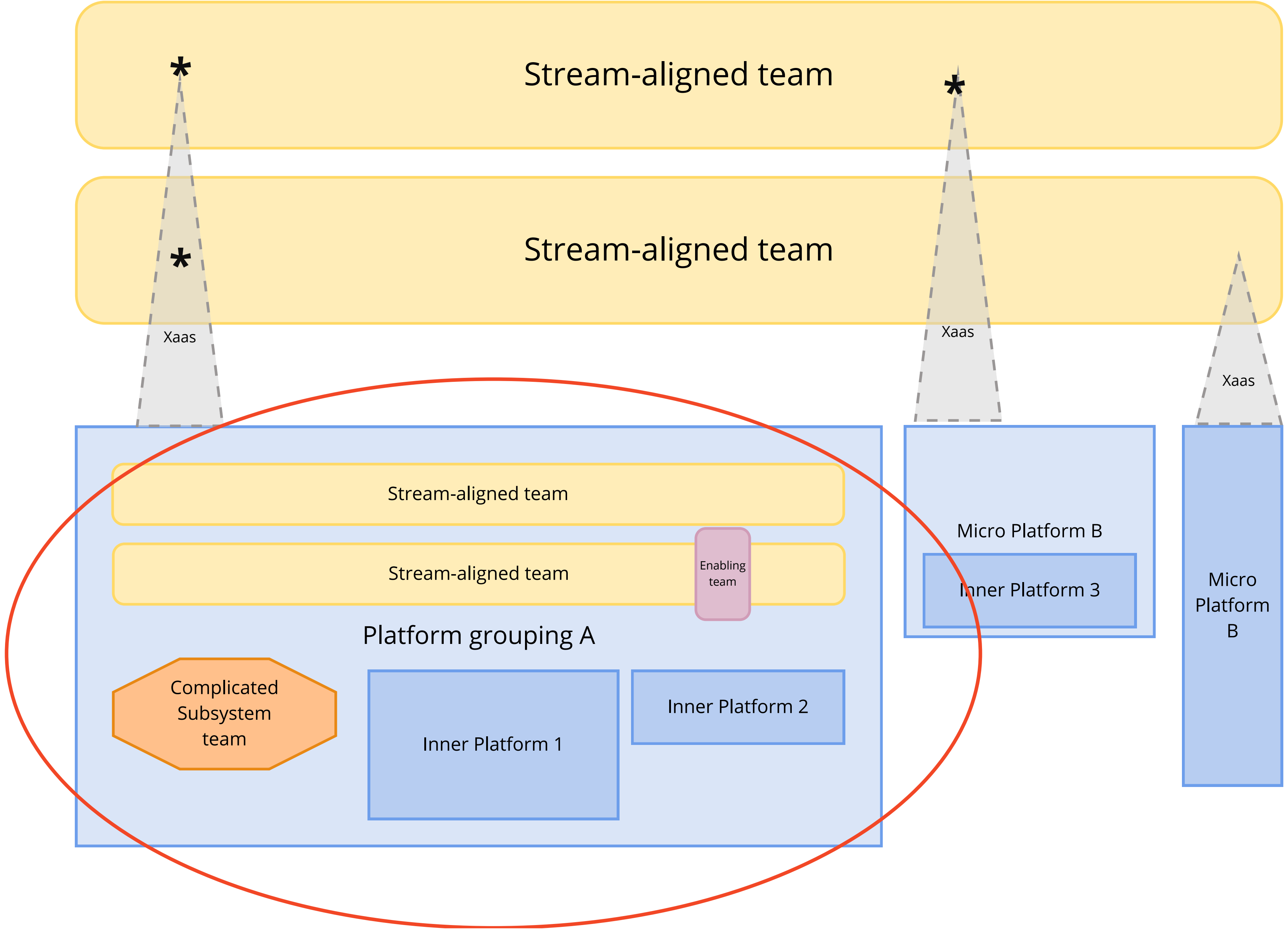
Complicated Subsystem team

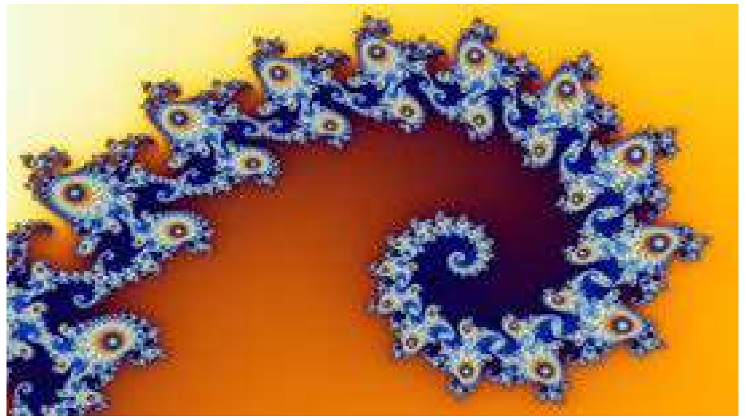
Fractal platforms - platform groupings 3



conflux

Fractal platforms - platform groupings 4

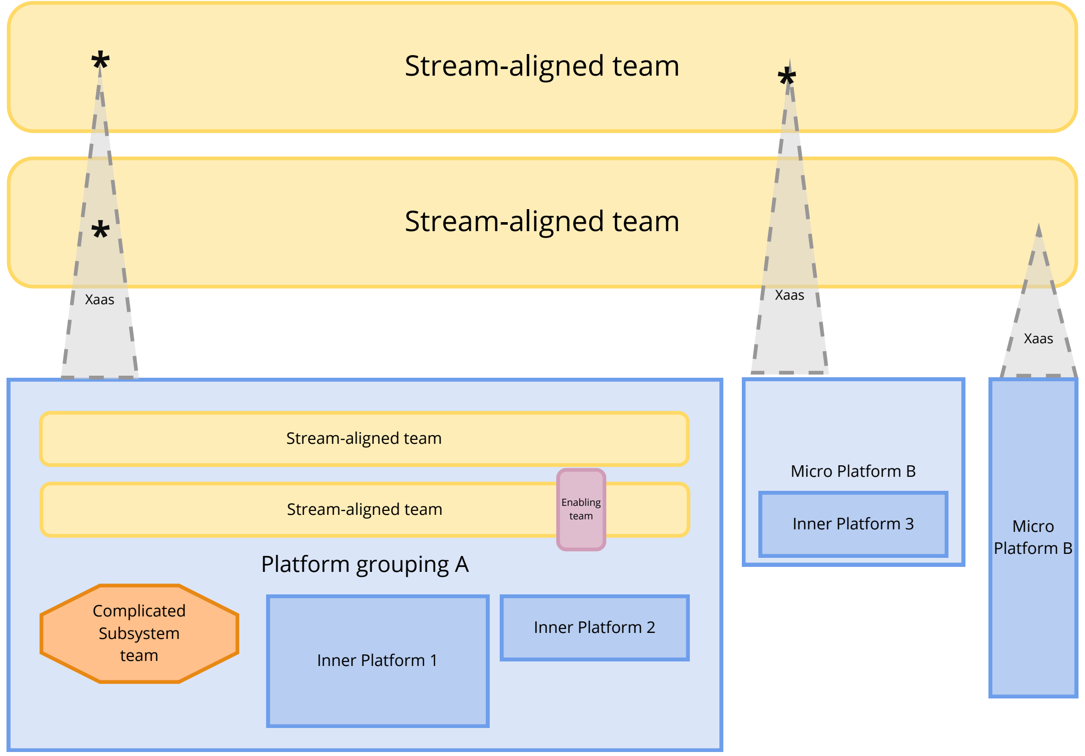




The organization itself is like a fractal

Fractal platforms - implications

What are the implications for platform product management of fractal platforms?
Think about: users/customers, perspectives in different teams, multiple platforms, etc.



Customer persona(s) per platform

Each platform needs a coherent focus

Multiple purposes might need multiple platforms

Platform discipline: aim to replace with SaaS/cloud option

Platform discipline: advertise platform capabilities and advocate for use

An ecosystem of providers 🌱

Fractal platforms - real example - Improbable

Virtual Worlds: using Team Topologies at Improbable

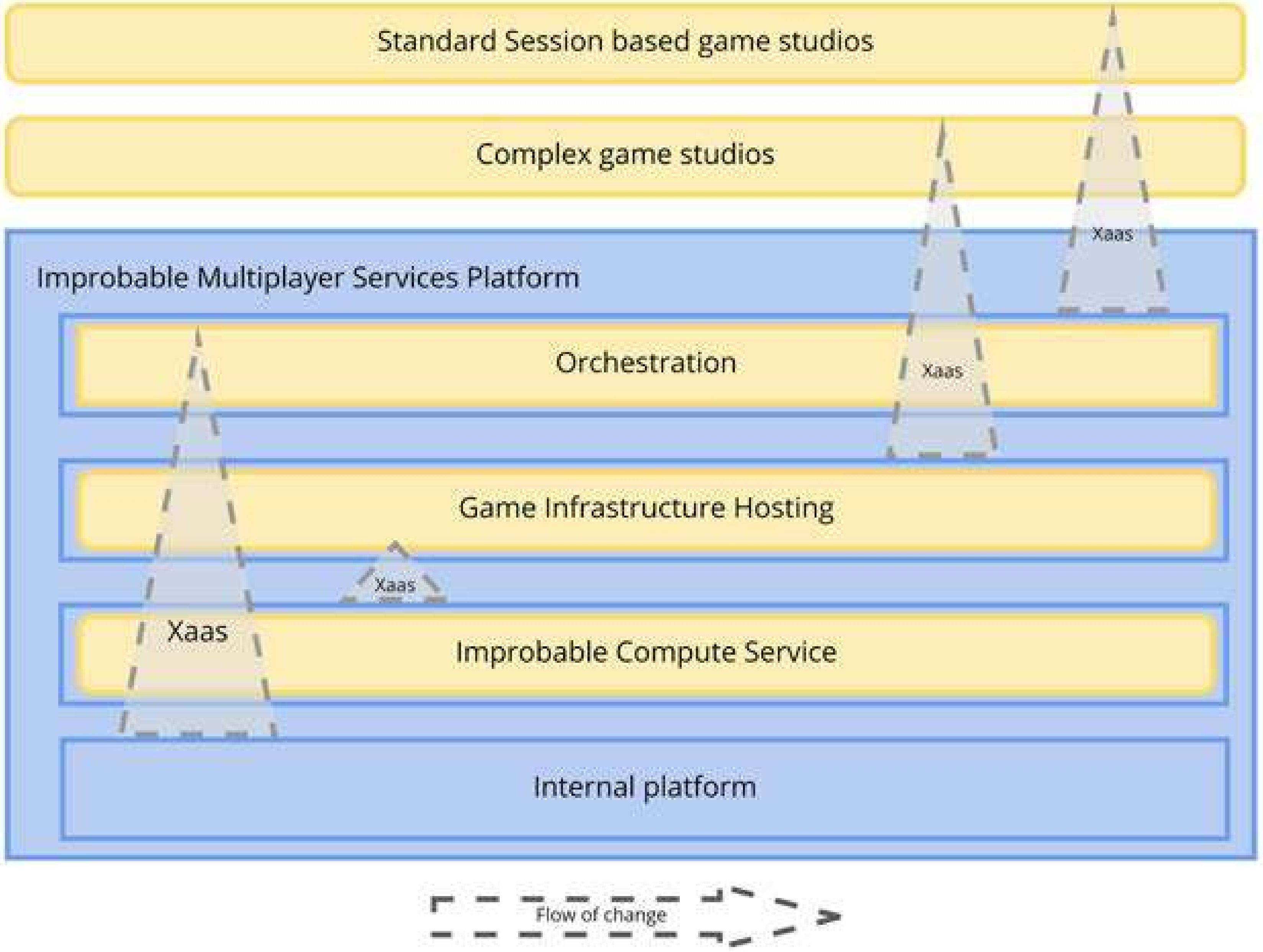
IMPROBABLE

Team Topologies teamtopologies.com/examples

teamtopologies.com

Virtual Worlds: using Team Topologies at Improbable to transform teams, technology, reliability, and customer satisfaction - Team Topologies

Founded in 2012, Improbable is a British technology company, dedicated to solving the challenges of building rich virtual worlds and pioneering the path to the metaverse. ... In 2020 Improbable acquired Munich-based video games company Zeuz, a managed h...



Standard Session based game studios
Complex game studios
Improbable Multiplayer Services Platform
Xaas
Orchestration
Xaas
Game Infrastructure Hosting
Xaas
Xaas
Improbable Compute Service
Internal platform
Flow of change

https://teamtopologies.com/industry-examples/virtual-worlds-using-team-topologies-at-improbable-to-transform-teams-technology-reliability-and-customer-satisfaction

Advanced patterns for platforms

Parallel services

Harvesting

Horizon scanning

Composite

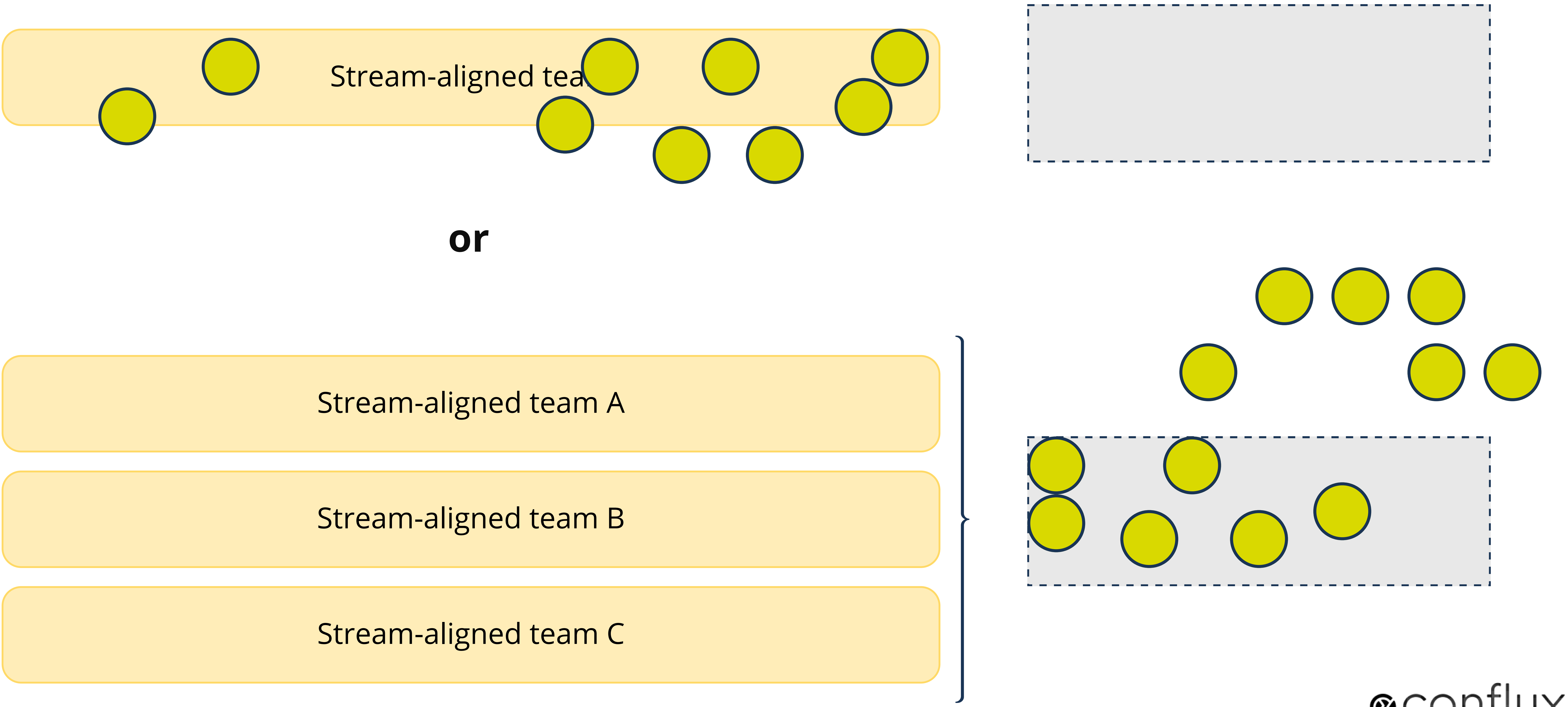
Patterns based on parallel product or service offerings

"Duplication is waste"

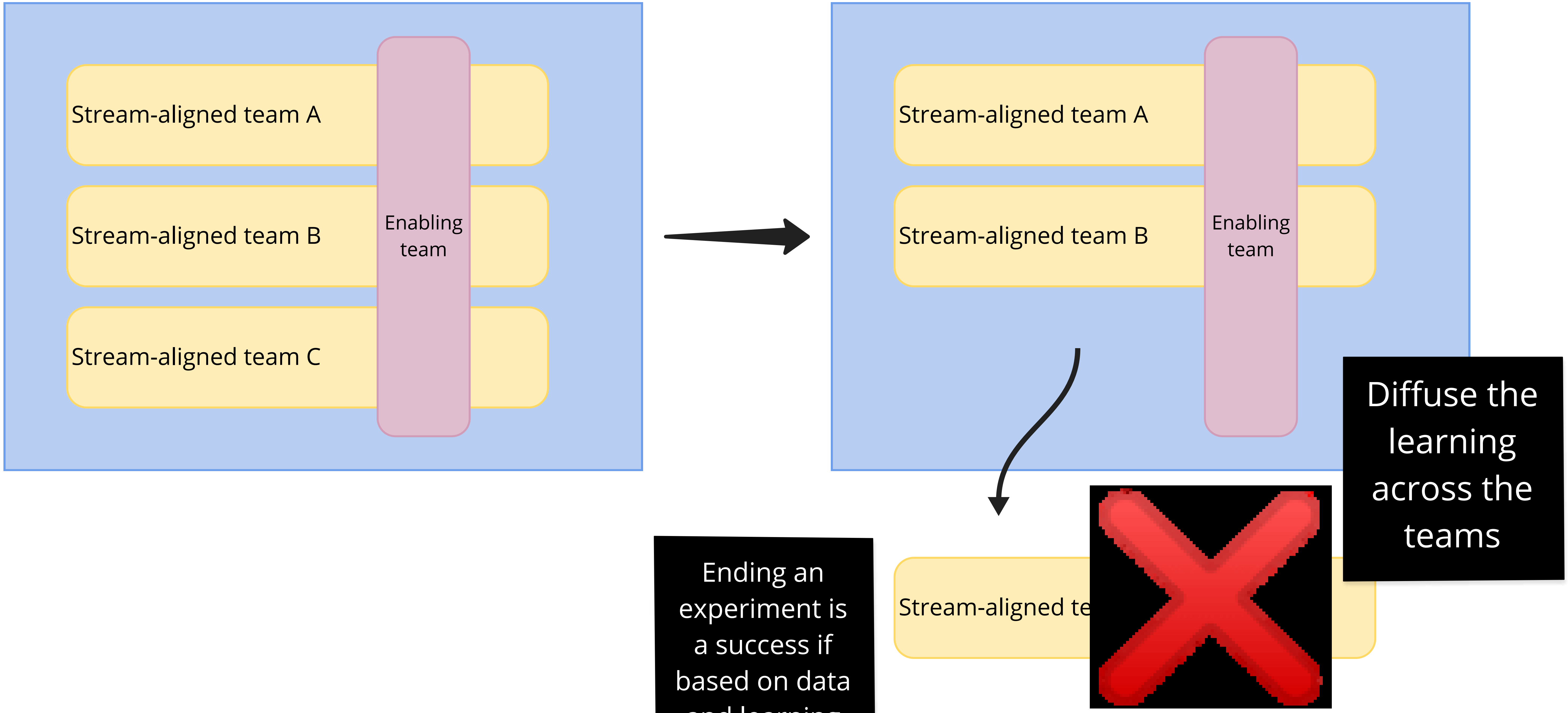
IT departments everywhere, failing hard at navigating fast flow

An early-stage "project" is simply a bet or a gamble

Which set of options would you bet on? The challenge and landscape are both changing rapidly



Parallel service offerings for rapid discovery and learning



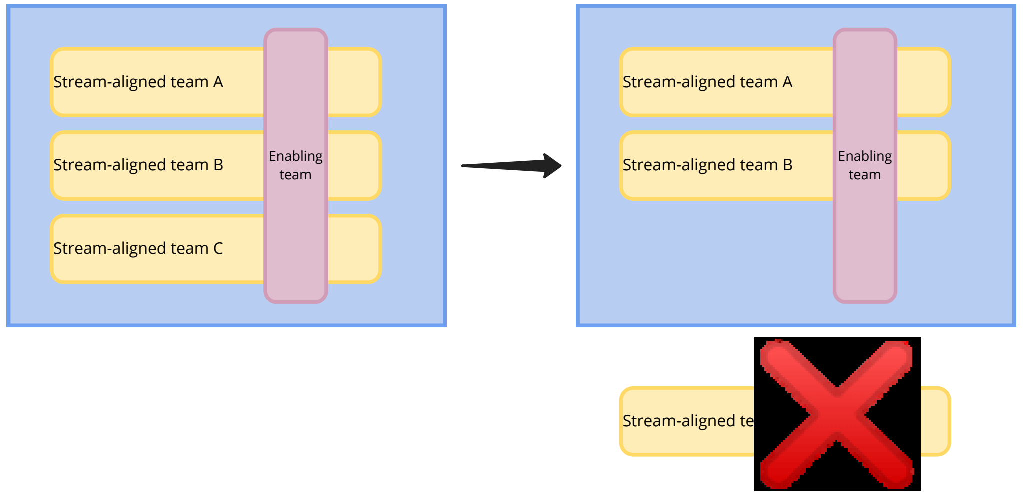
Diffuse the learning across the teams

Ending an experiment is a success if based on data and learning

conflux

Guidelines for parallel service offerings

Duplicate a sticky if you disagree with someone's choice of position!



Probably bad 👎 ⟷ 👍 Probably good

Make the teams compete for a prize 🏆

Diffuse the learning across the teams 🔀

Reward Team C for ending early 🎖️

Make the teams compete for a prize 🏆

Make the teams compete for a prize 🏆

Make an example of Team C as a "failure" 🤡

Maximum of ~ 20 parallel efforts per "problem" 2️⃣0️⃣

Reward Team A and Team B as "successful" ☑️

Reward Team C for ending early 🎖️

Make the teams compete for a prize 🏆

Diffuse the learning across the teams 🔀

Deciding to cancel one of the approaches early - based on data and learning - is a success 🧪

Make the teams compete for a prize 🏆

Maximum o 3 parallel efforts pe "problem' 3️⃣

Reward Team C for ending early 🎖️

Reward Team C for sharing their learning 🎓

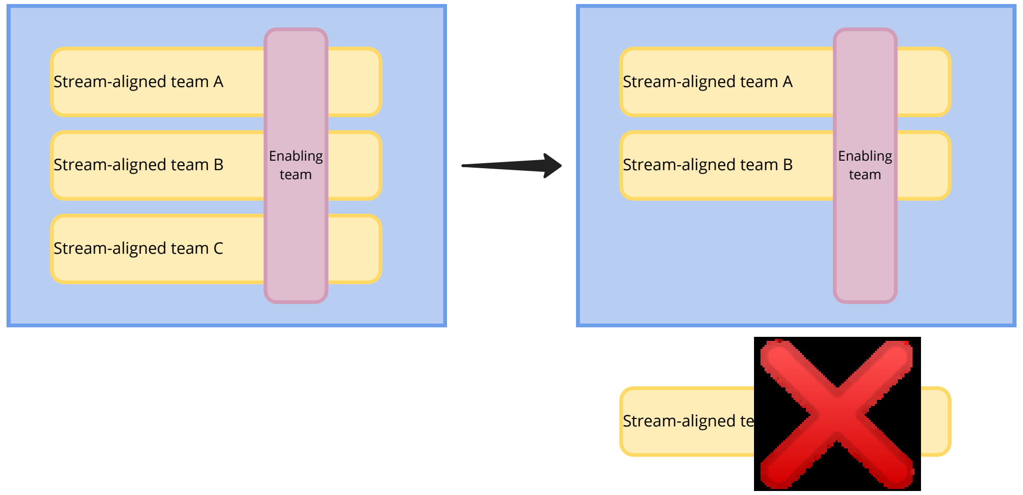
Make the teams compete for a prize 🏆

Diffuse the learning across the teams 🔀

conflux

Guidelines for parallel service offerings - expert view

Duplicate a sticky if you disagree with someone's choice of position!



Probably bad 👎 ⟷ 👍 Probably good

Maximum of ~ 20 parallel efforts per "problem" 2️⃣0️⃣

Make an example of Team C as a "failure" 🤡

Make the teams compete for a prize 🏆

Reward Team C for ending early 🎖️

Reward Team A and Team B as "successful" ✔️

Deciding to cancel one of the approaches early - based on data and learning - is a success 🧪

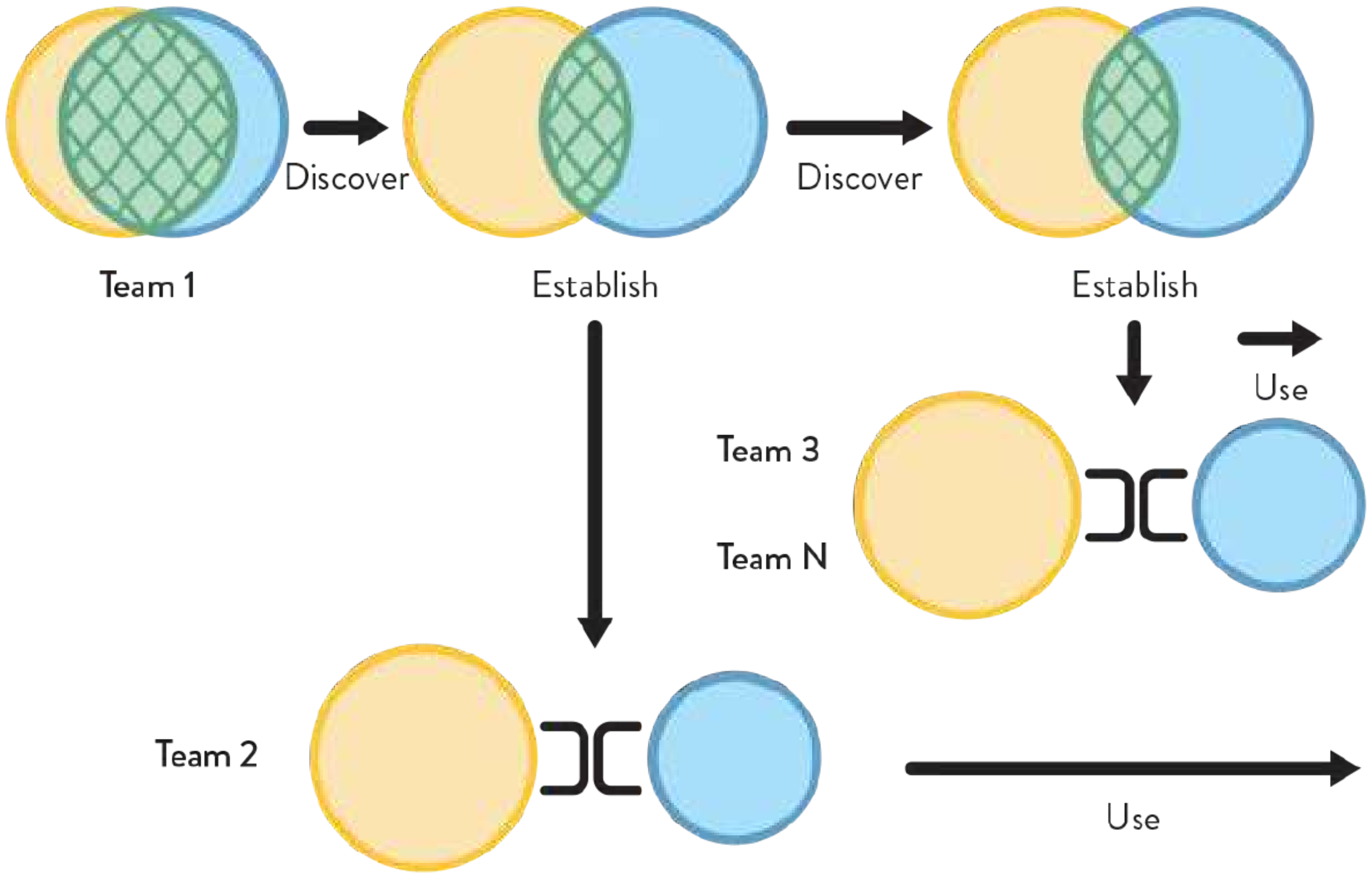
Maximum of ~ 3 parallel efforts per "problem" 3️⃣

Diffuse the learning across the teams 🔀

Reward Team C for sharing their learning 🎓

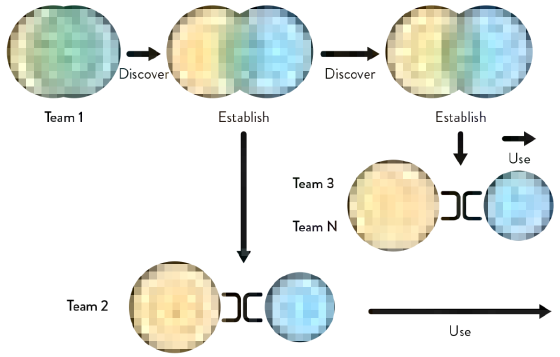
conflux

Sharing and harvesting proven solutions

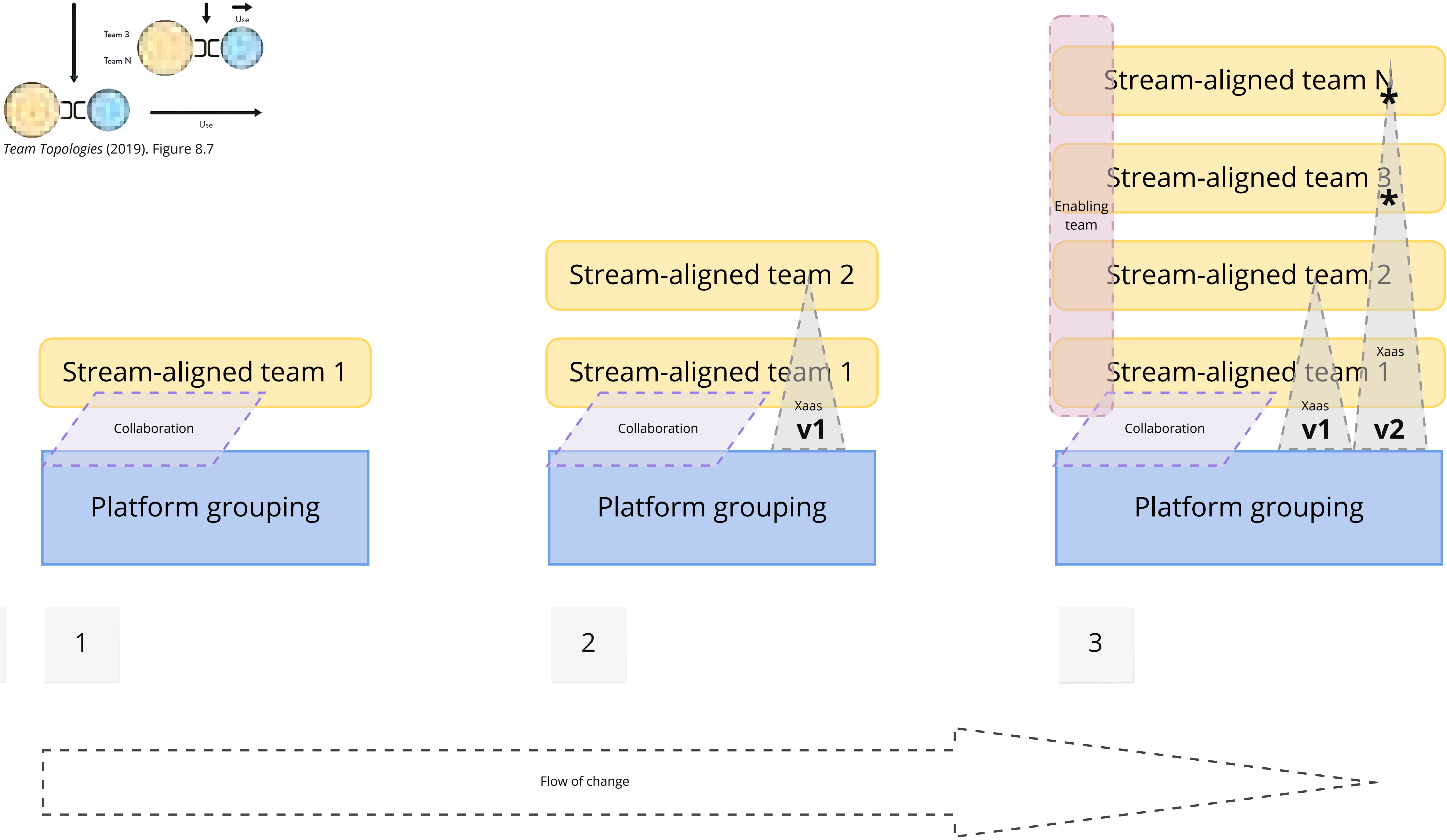


Taken from *Team Topologies* (2019). Figure 8.7

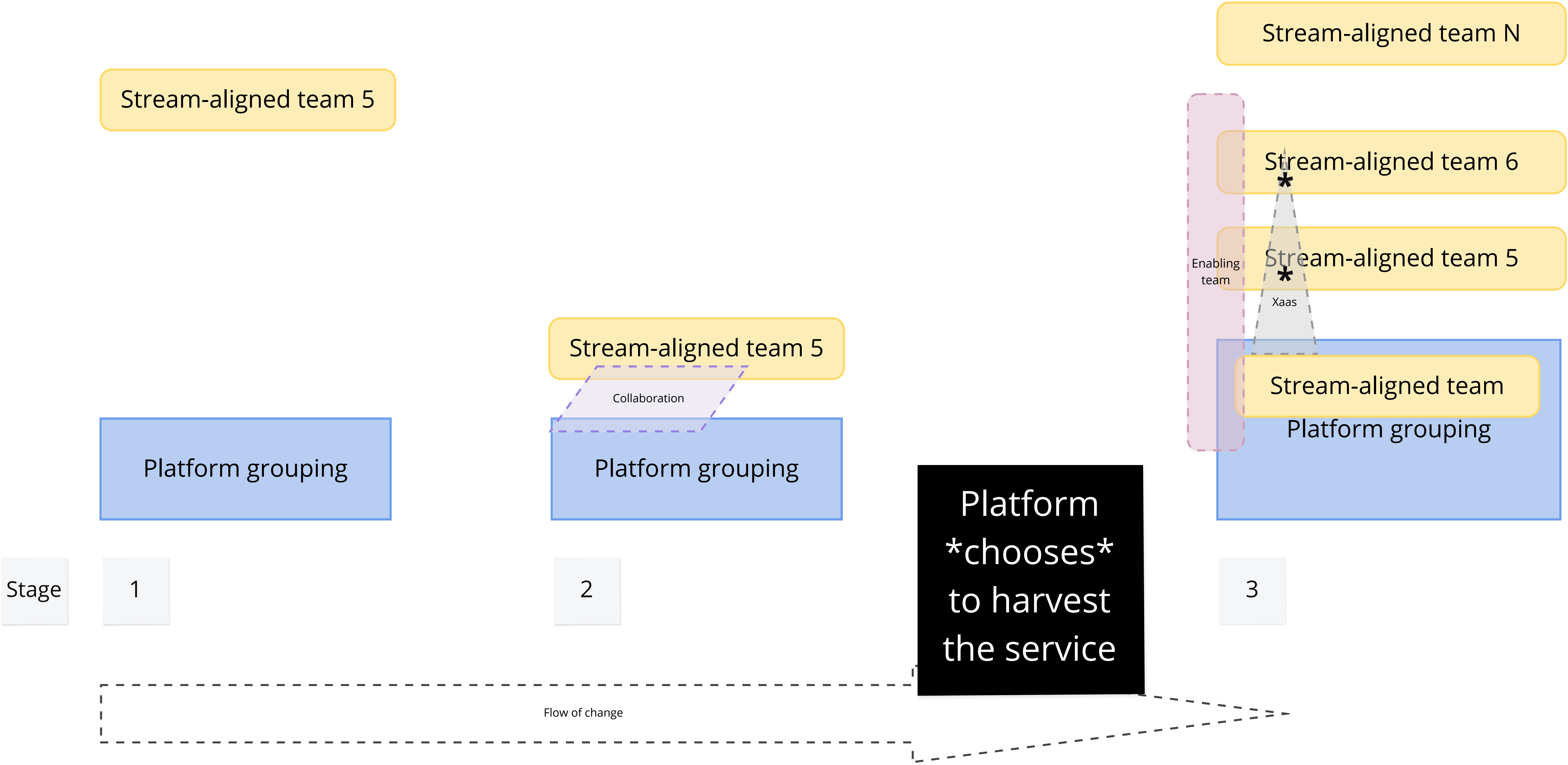
Sharing and harvesting proven solutions - TT shapes



Taken from *Team Topologies* (2019). Figure 8.7



Platforms choose what to harvest



Platform *chooses* to harvest the service

conflux

Freedom of choice in two directions for platforms

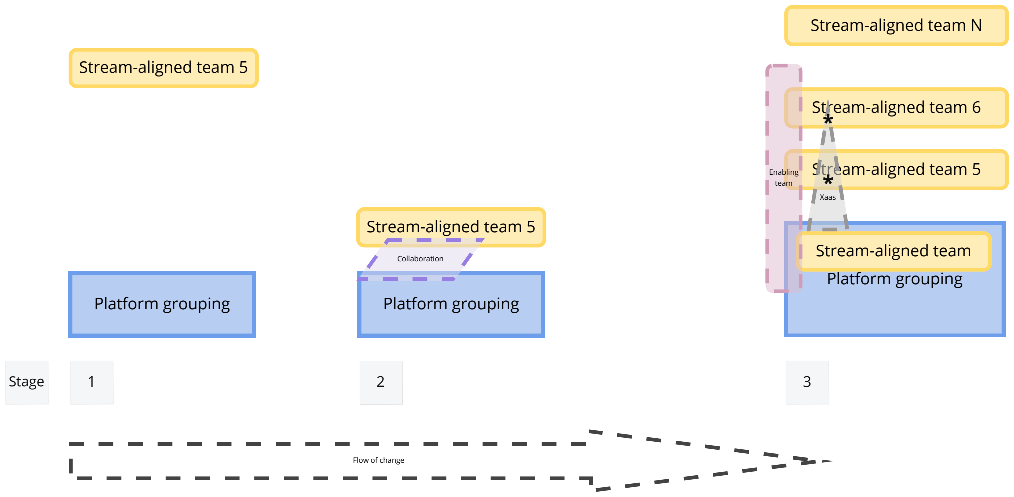
Using the platform

Steam-aligned teams *choose* to use one or more platform services

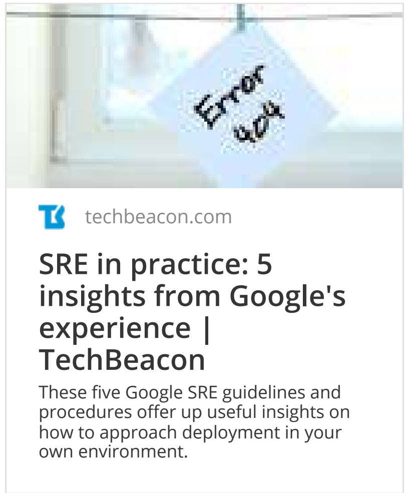
Evolving the platform

Platform *chooses* to harvest the service

What's needed for a platform to *choose* to harvest? - expert



Platform *chooses* to harvest the service



techbeacon.com

SRE in practice: 5 insights from Google's experience | TechBeacon

These five Google SRE guidelines and procedures offer up useful insights on how to approach deployment in your own environment.

https://techbeacon.com/enterprise-it/sre-practice-5-insights-googles-experience

Stream-aligned team wants to "offload" the service

Platform group wants to "adopt" the service

Business case

Cost details

SLI / SLO / SLA

User Personas

UX profiling

Vendor Relationship

Docs

Use Cases

Incident logs

Reliability profiling

Monitoring and telemetry

Deployment tests

Operational tests

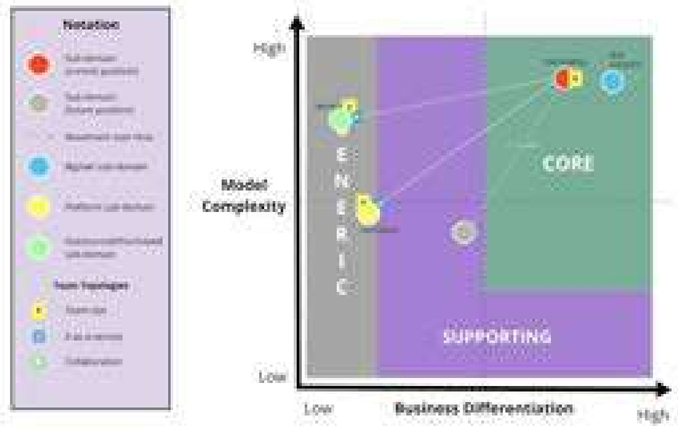
Good automated test coverage

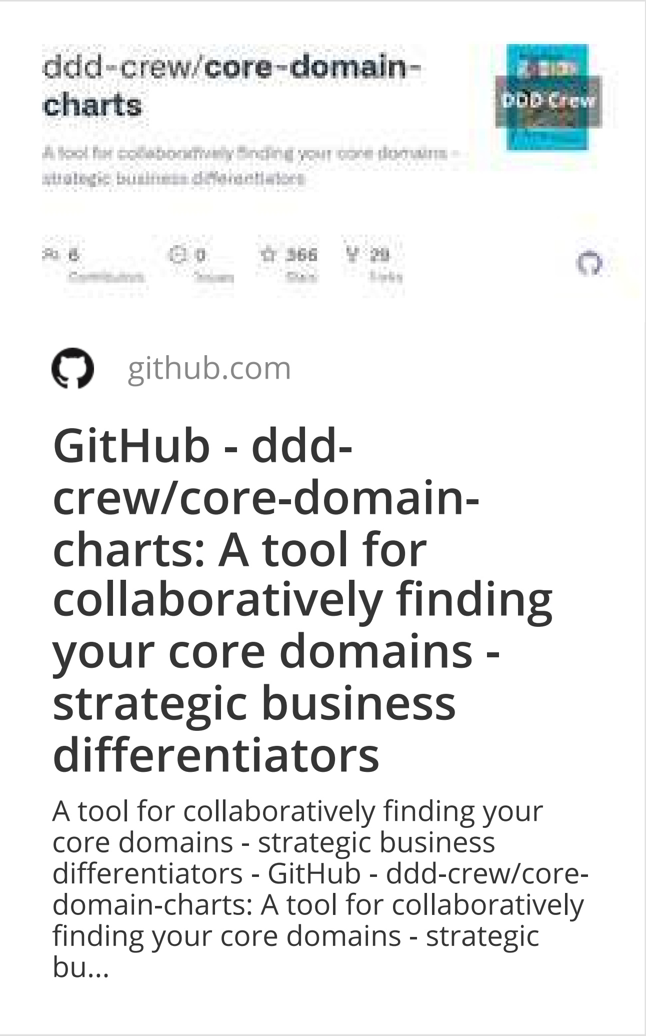
Source code

Horizon scanning using Core Domain Charts

Core Domain Charts help you to visualise the strategic importance of each (sub)domain or business capability in your architecture allowing you to make business model-aligned architectural decisions.

Core Domains are the parts of your domain where the expected ROI is greatest, and deserve the highest focus.
The true power of this technique is the conversations that it triggers, especially cross-discipline. Complexity is something that engineers can gauge whereas business differentiation is provided by product managers or business stakeholders.





ddd-crew/**core-domain-charts**

A tool for collaboratively finding your core domains - strategic business differentiators

6 Contributors · 0 Issues · 366 Stars · 29 Forks

github.com

GitHub - ddd-crew/core-domain-charts: A tool for collaboratively finding your core domains - strategic business differentiators

A tool for collaboratively finding your core domains - strategic business differentiators - GitHub - ddd-crew/core-domain-charts: A tool for collaboratively finding your core domains - strategic bu...

https://github.com/ddd-crew/core-domain-charts

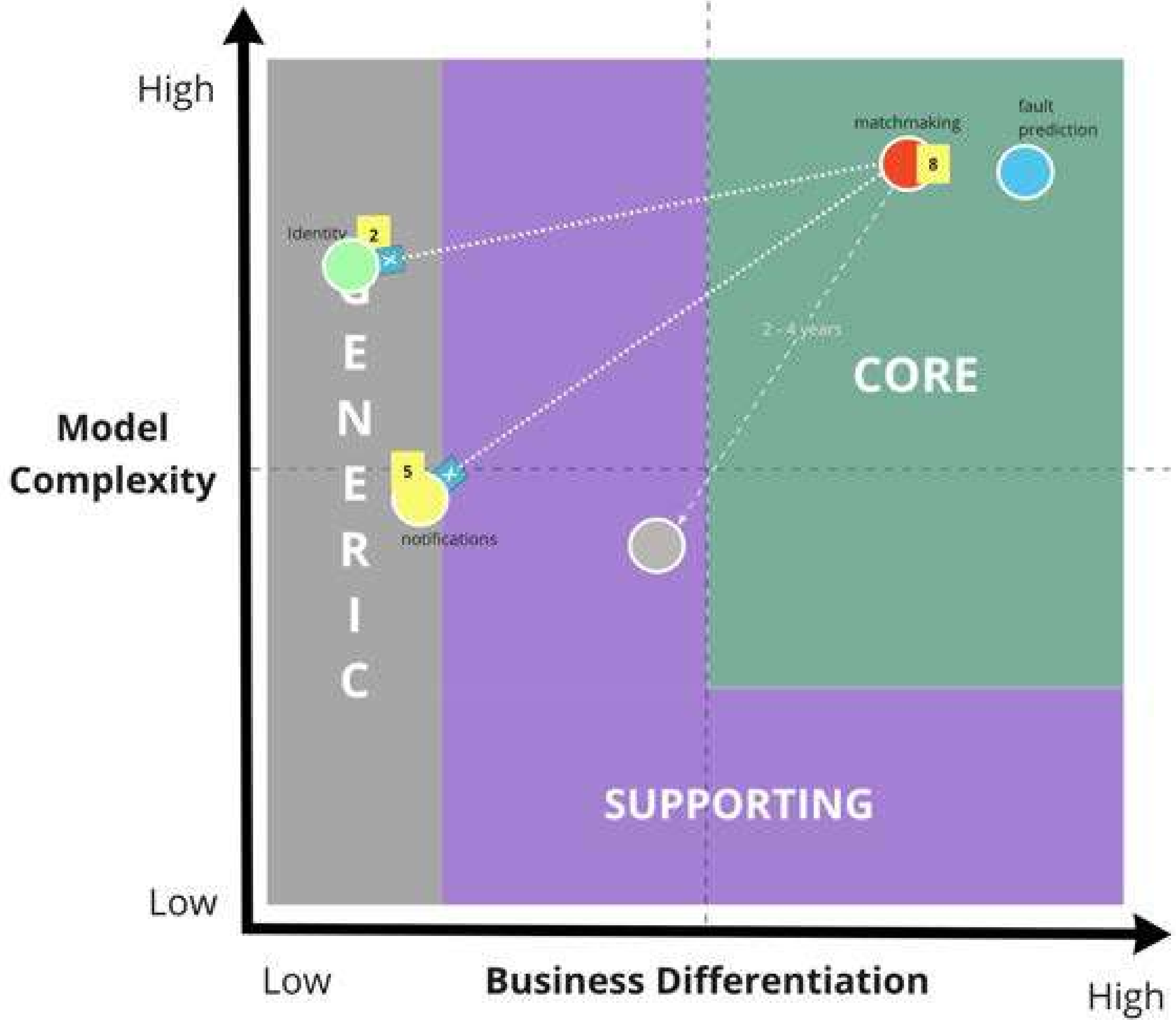
Core Domain Charts - rapid overview

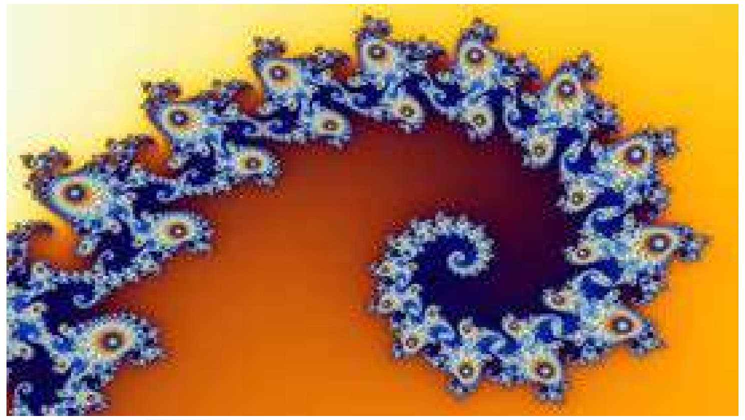
Notation

- 🔴 Sub-domain (current position)
- ⚪ Sub-domain (future position)
- - - → Movement over time
- 🔵 Big-bet sub-domain
- 🟡 Platform sub-domain
- 🟢 Outsourced/Purchased sub-domain

Team Topologies

- x Team size
- ☒ X-as-a service
- 🟩 Collaboration



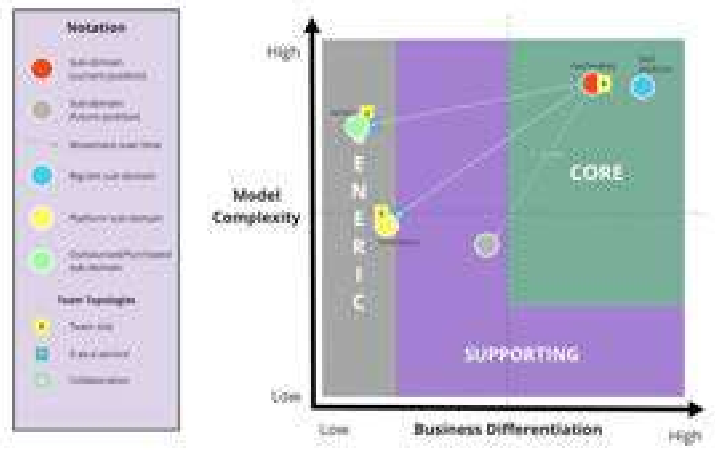


💡 Because the organization is fractal, we can use Core Domain Charts inside a platform

conflux

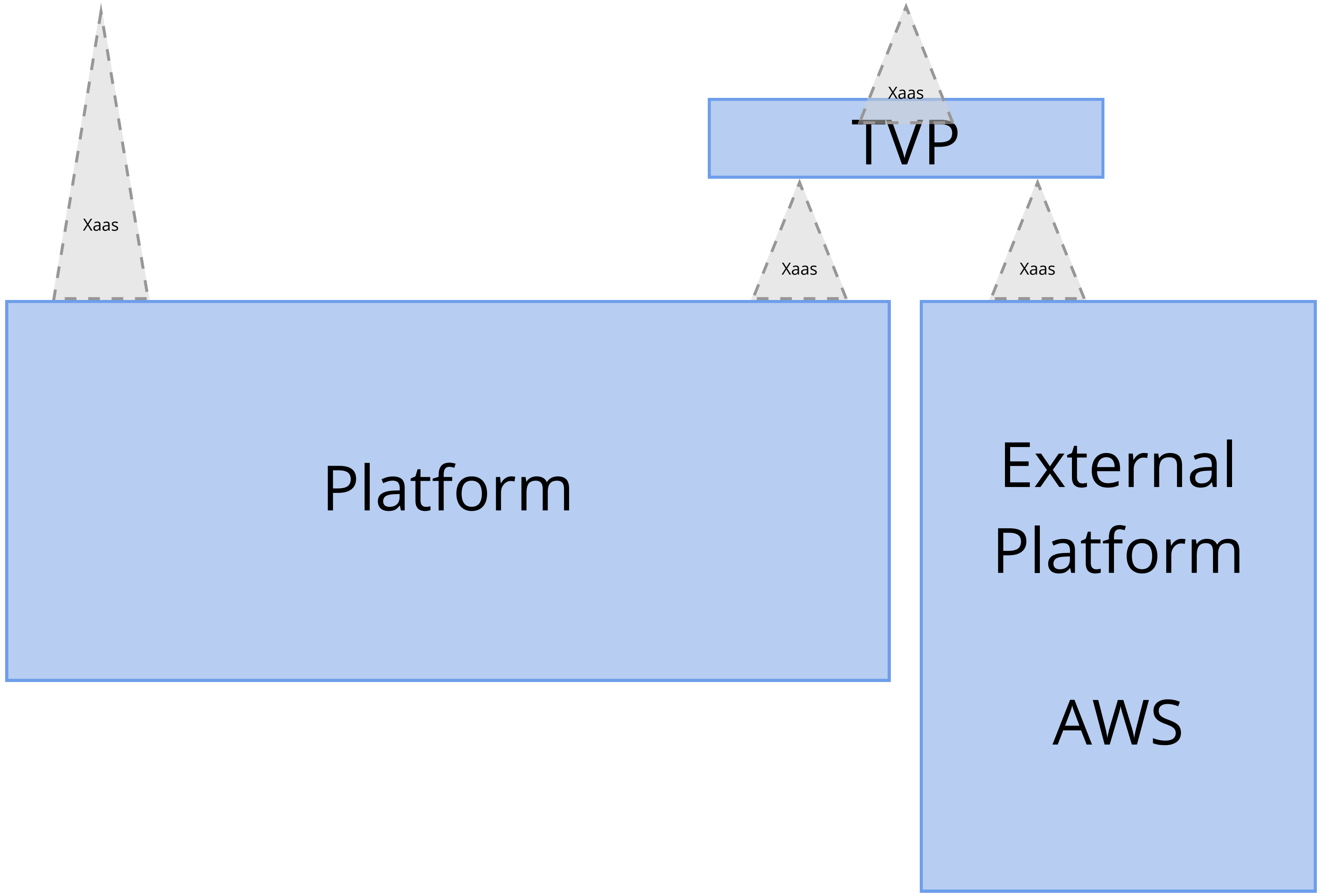
What happens if we have an existing platform already?

Core Domain Charts

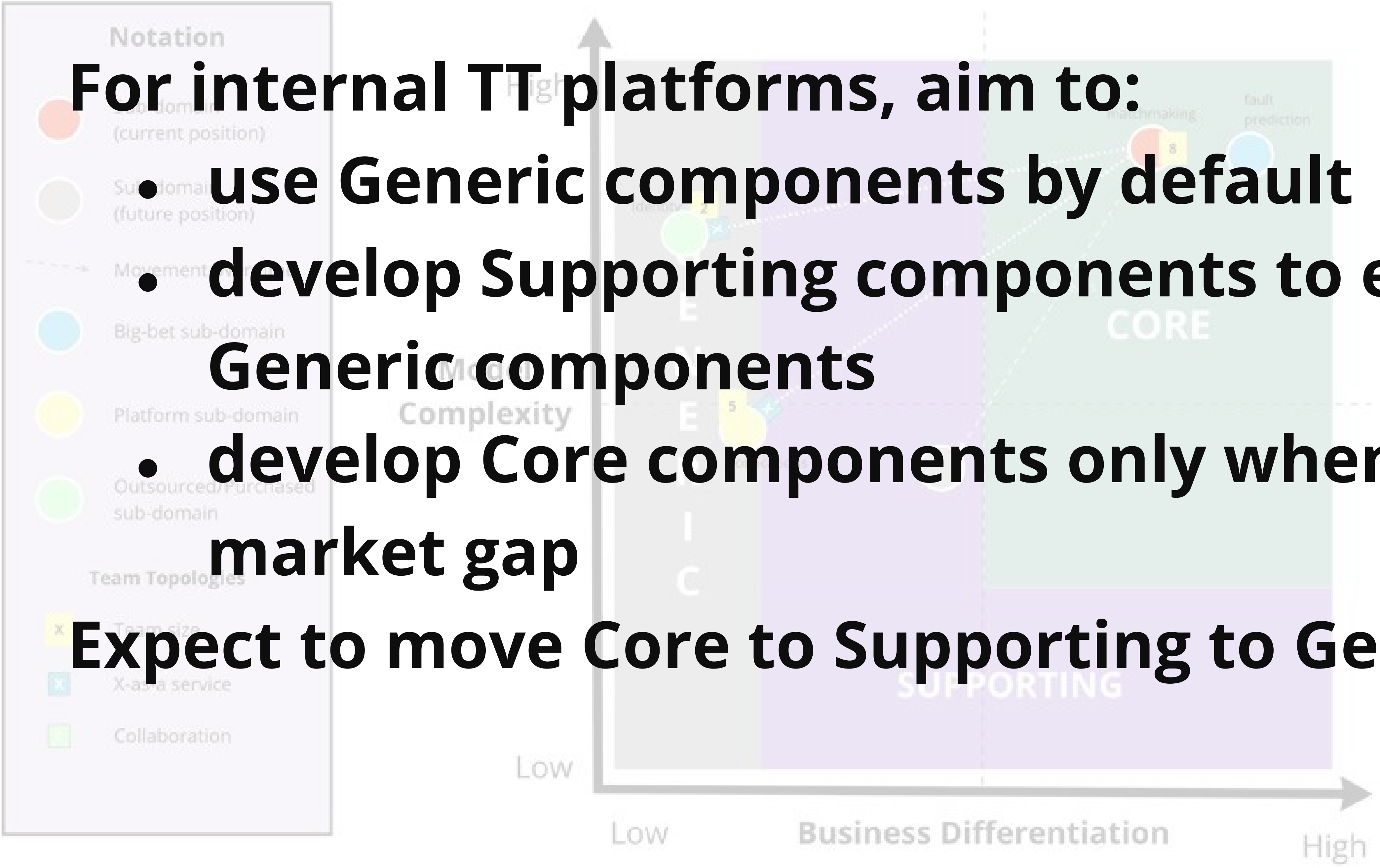


TradeMe

"TVP follows the Platform-as-Product mindset. As a product, the main customers of TVP are our developers. The main goal of this product is to improve DevEx (Developer Experience)."



Horizon Scanning for TT platforms using Core Domain Charts



For internal TT platforms, aim to:

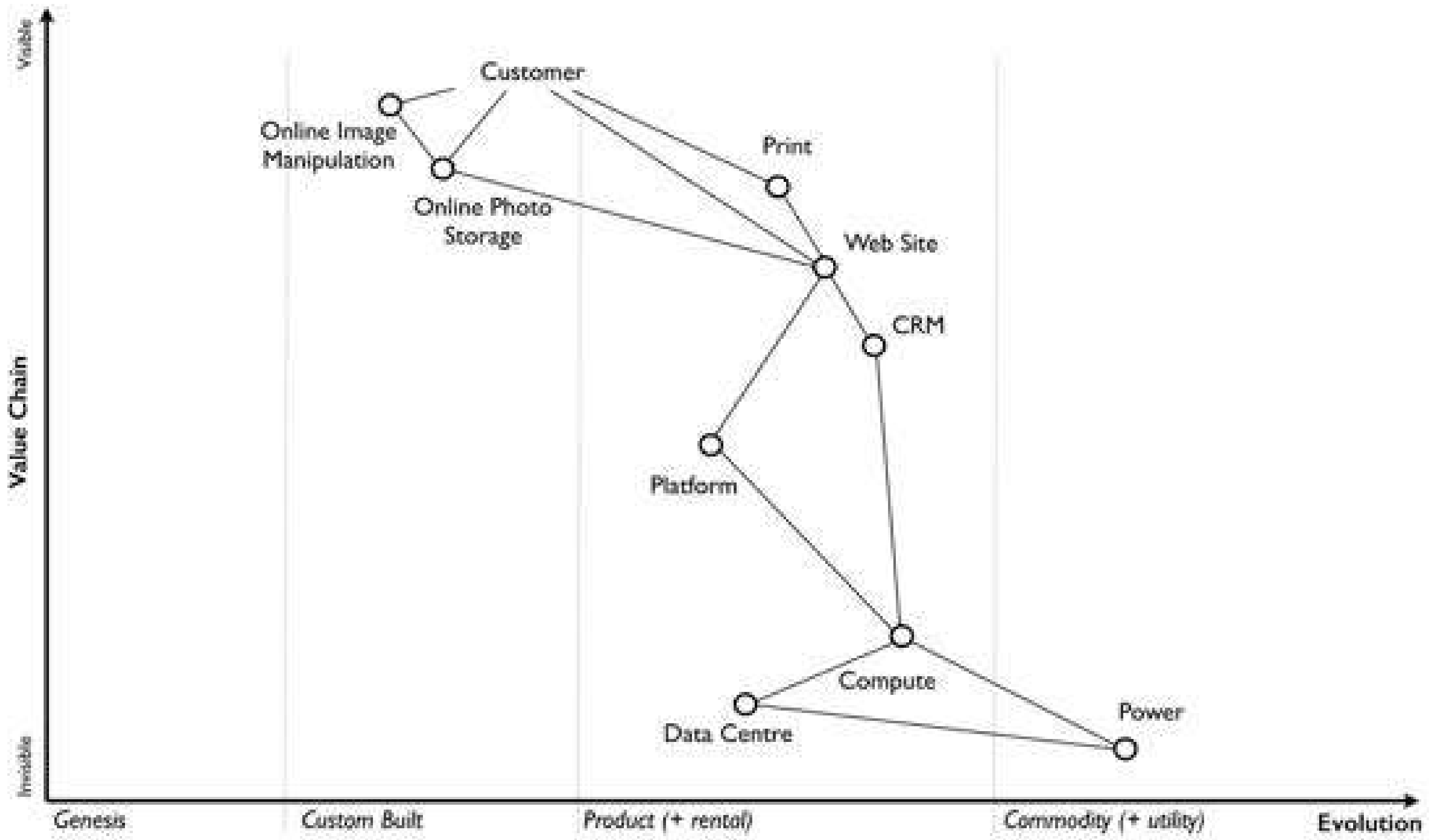
- **use Generic components by default**
- **develop Supporting components to ease usage of Generic components**
- **develop Core components only when there is a market gap**

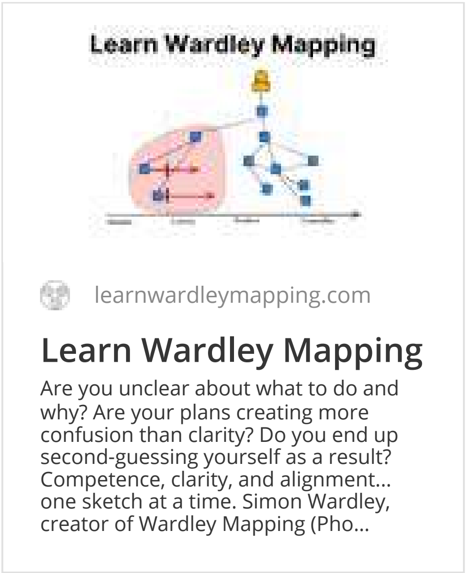
Expect to move Core to Supporting to Generic ASAP

Horizon Scanning for TT platforms - Wardley Mapping?

Q: Is this Wardley Mapping by stealth?

A: Yes, but don't worry about it 😁





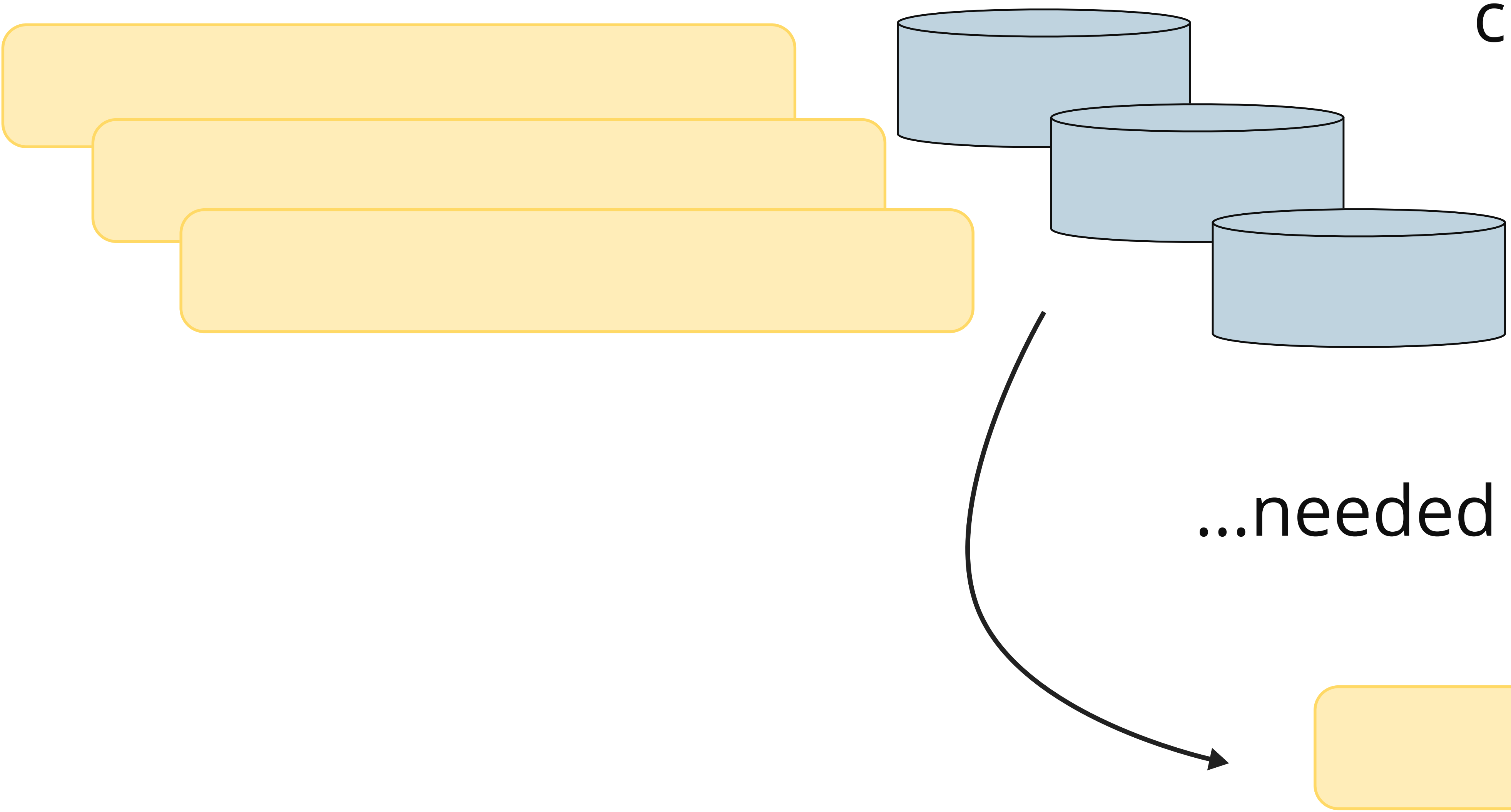
learnwardleymapping.com

Learn Wardley Mapping

Are you unclear about what to do and why? Are your plans creating more confusion than clarity? Do you end up second-guessing yourself as a result? Competence, clarity, and alignment... one sketch at a time. Simon Wardley, creator of Wardley Mapping (Pho...

https://learnwardleymapping.com/

Composite Services

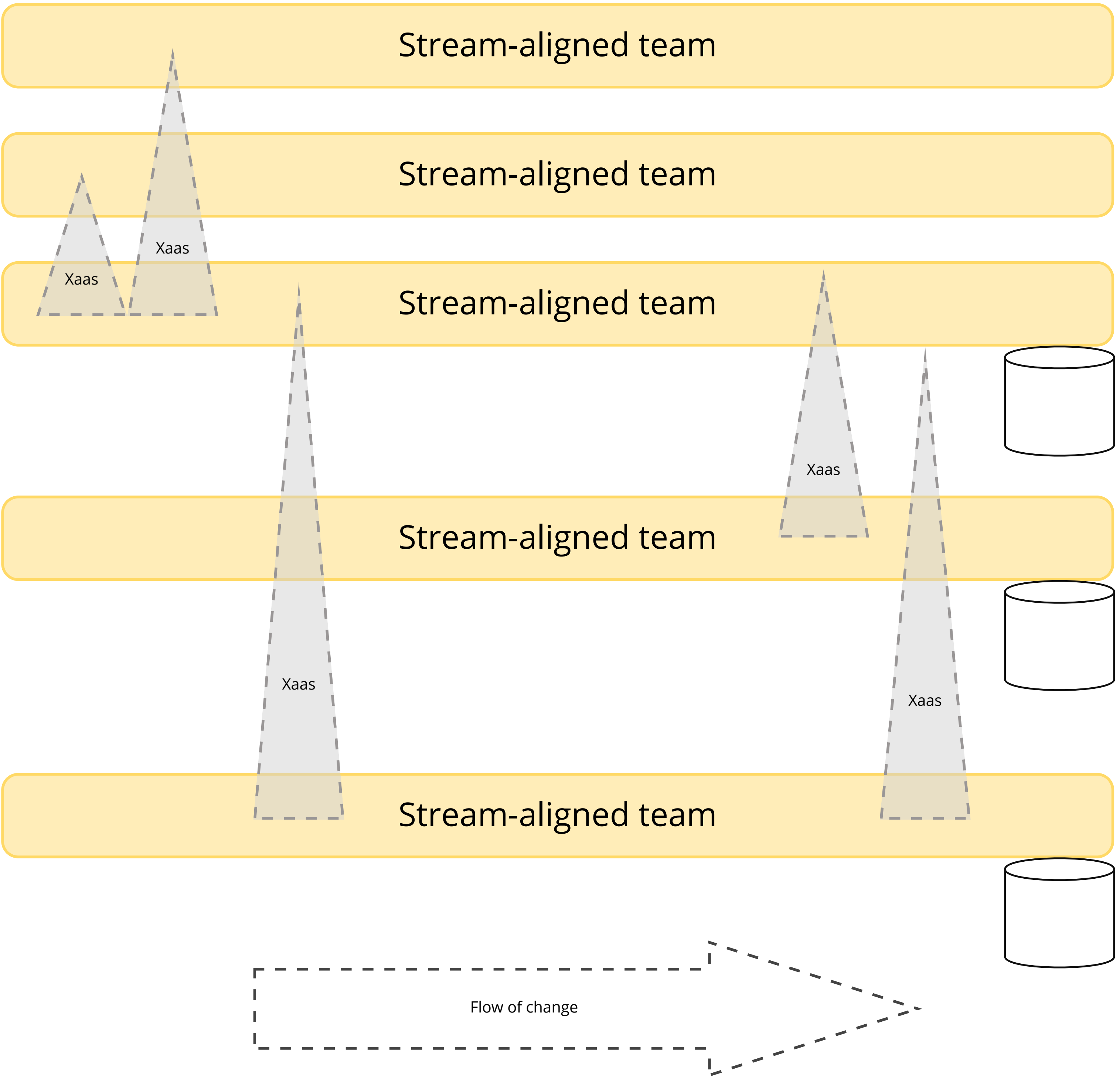


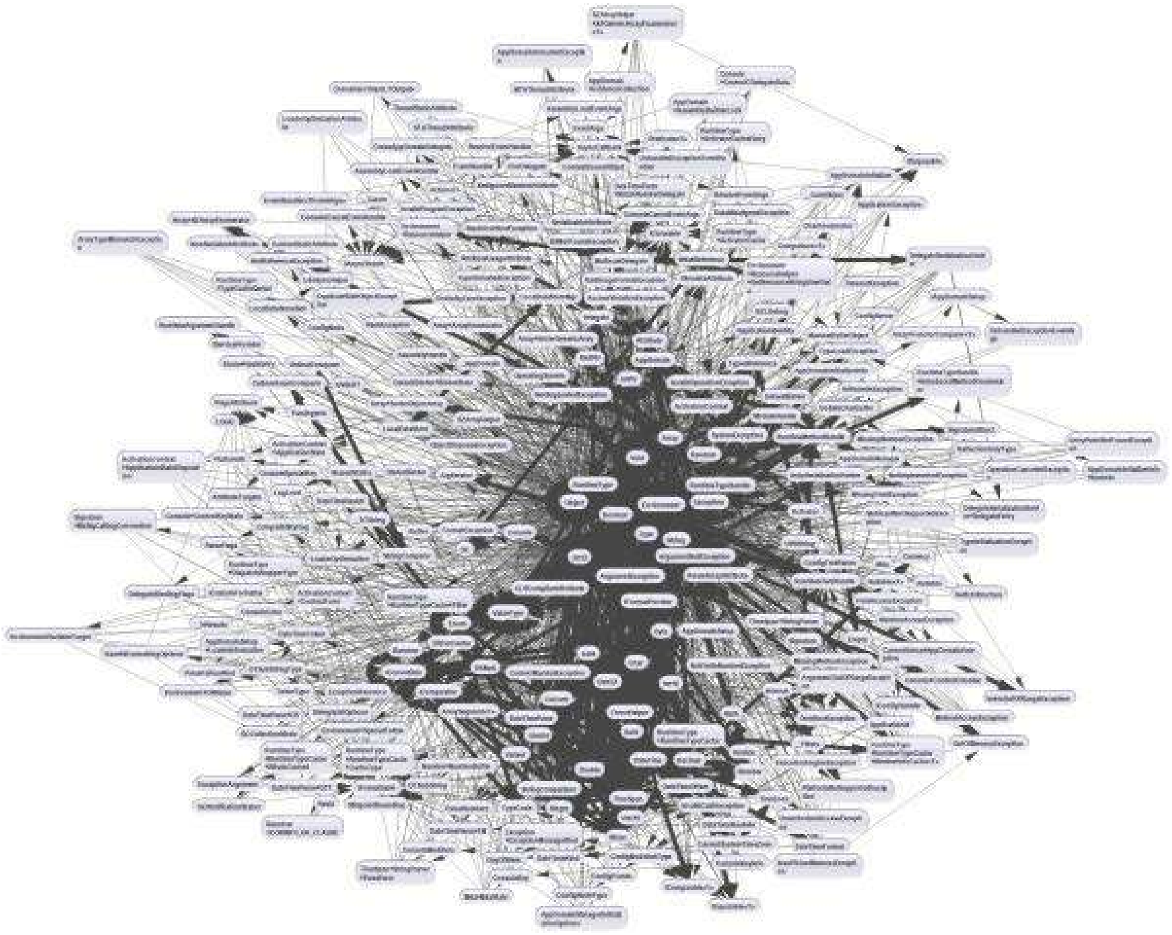
Data from several customer-facing services...

...needed for a 'composite' service

conflux

Anti-pattern: a web of data requests

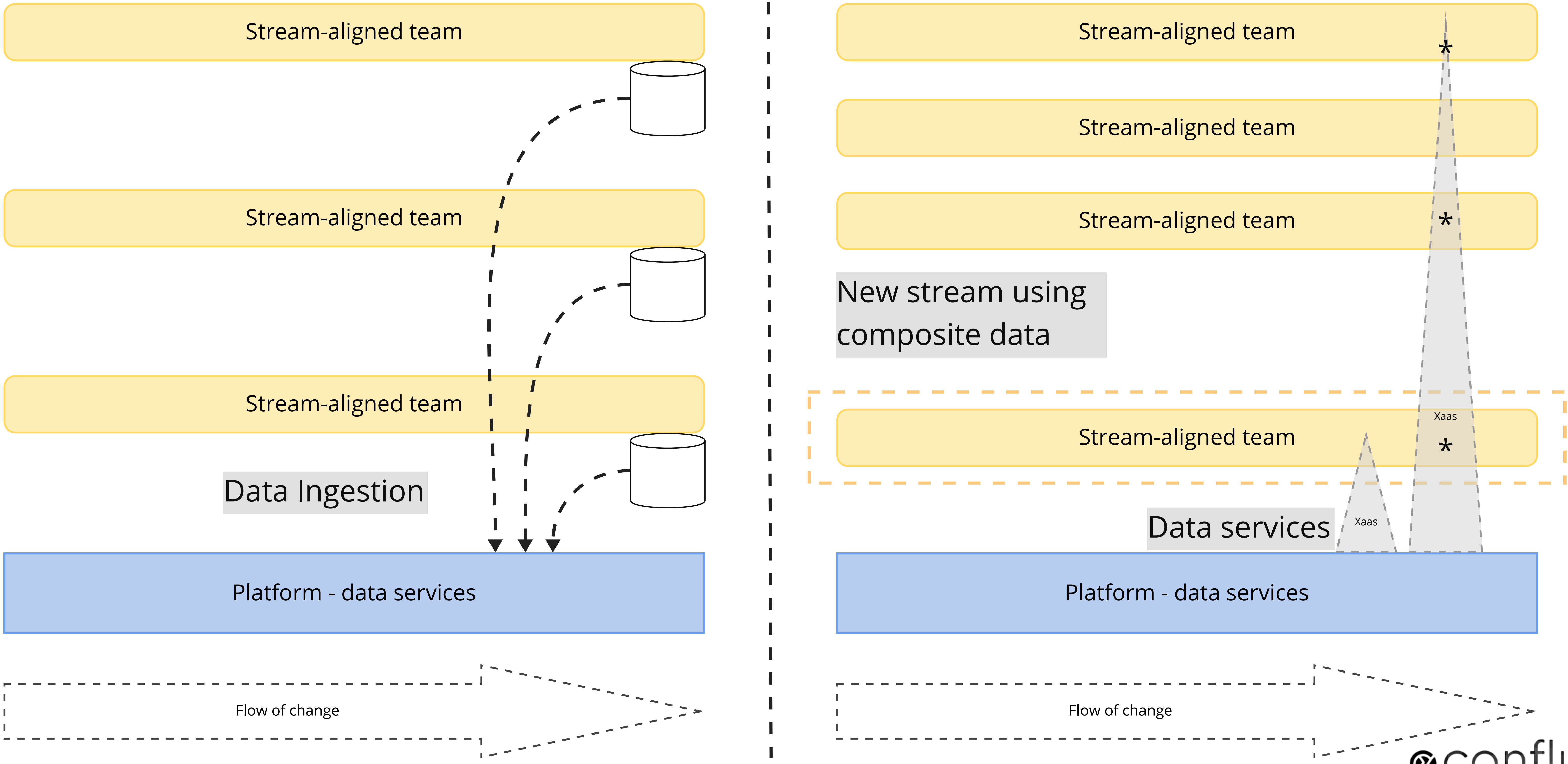




"...the fact that you're doing microservices doesn't automatically save you from that. Likely you just have a distributed big ball of mud" - Mario Fusco

https://twitter.com/mariofusco/status/1112332826861547520

Pattern: composite services via a data platform

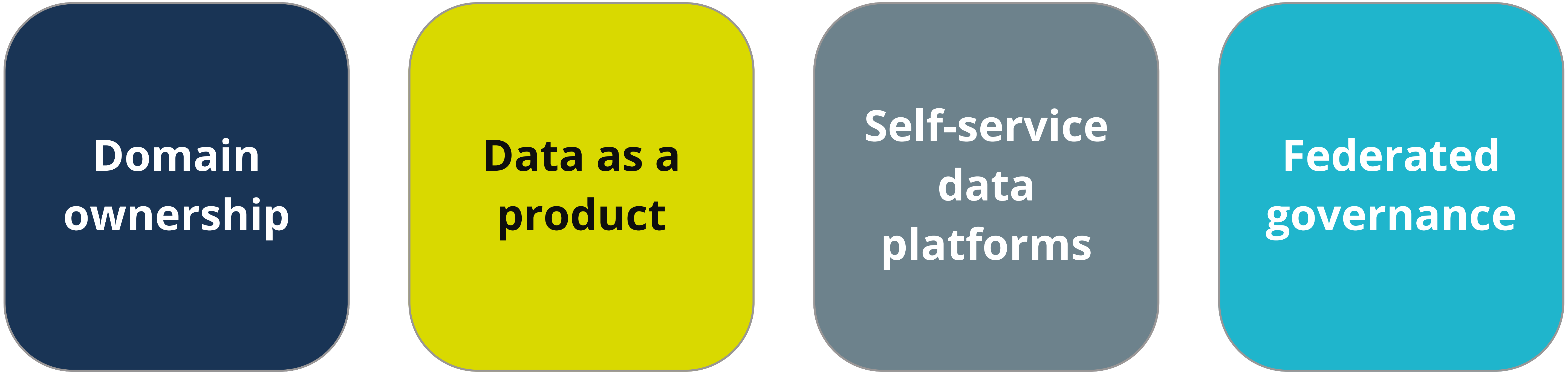


conflux

Data Mesh

Data Mesh is an analytical data architecture and operating model where data is treated as a product and owned by teams that most intimately know and consume the data.

Four principles of Data Mesh



Domain ownership

Data as a product

Self-service data platforms

Federated governance

Desynchronous

Avoid temporal dependencies

Desynchronous - key rules

Desynchronous

1. 🕑 No team is allowed to *force* or coerce another team to change an API or service or module within a particular timescale.
2. 🤝 No more than two teams may work in a synchronized way (with synchronized changes) in the same timescale.
3. 💰 The value of an underlying API or service or module increases as more APIs or services or modules depend on that thing at runtime.

Desynchronous - implications

Desynchronous

Open Systems Theory

Redundancy of parts vs redundancy of functions

Imagine each team MUST work with what's actually available already, as if internal teams were external providers, rather than creating time-based dependencies on other teams.

Part 3 - Finding good boundaries for flow using Independent Service Heuristics

conflux

Outline of part 3

- Independent Service Heuristics (ISH) as a technique for finding good boundaries for fast flow
- Three different lenses: fracture planes, user needs, micro-enterprises
- Using ISH to find good boundaries using these lenses

How can we find good boundaries for flow?

Good team and service boundaries work well for **flow**.
But how can we find good boundaries for flow?

One technique is **Independent Service Heuristics**. We look for services that could (if we wanted) be run as a separate cloud service, and then critique the candidate service against various criteria. [CC BY-SA license]

https://github.com/TeamTopologies/Independent-Service-Heuristics

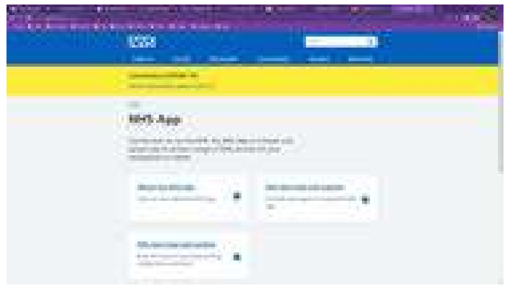
The end-goal from using Independent Service Heuristics

Loosely-coupled, independent services and teams, aligned to viable streams of value

Real-world Software-Enriched Services

Inspiration:

1. Clothes laundering https://www.laundryheap.co.uk/ - simplifies the tasks
2. Buy a car https://www.cinch.co.uk/ - delivered to your door and 14-day money back
3. Run a business https://www.odoo.com/ - an opinionated "platform" of tools
4. Try on clothes digitally https://www.zyler.com/ - saves time in stores
5. ID verification https://www.id-pal.com/ - uses features of a mobile app
6. Transcription https://rev.com/ - automated or human, but all 'As a Service'
7. Vaccination notification and booking: https://www.nhs.uk/nhs-app/ - location specific
8. Website monitoring https://uptimerobot.com/ - no installed software - just a focus on what is public
9. International money transfers https://wise.com/ - no actual bank accounts, but much of the hassle removed
10. Try on glasses virtually https://luna.io/virtual-try-on/ - no need to visit a store



The fix cheaper international account.
Available for you in United Kingdom

ISH - 3 different lenses

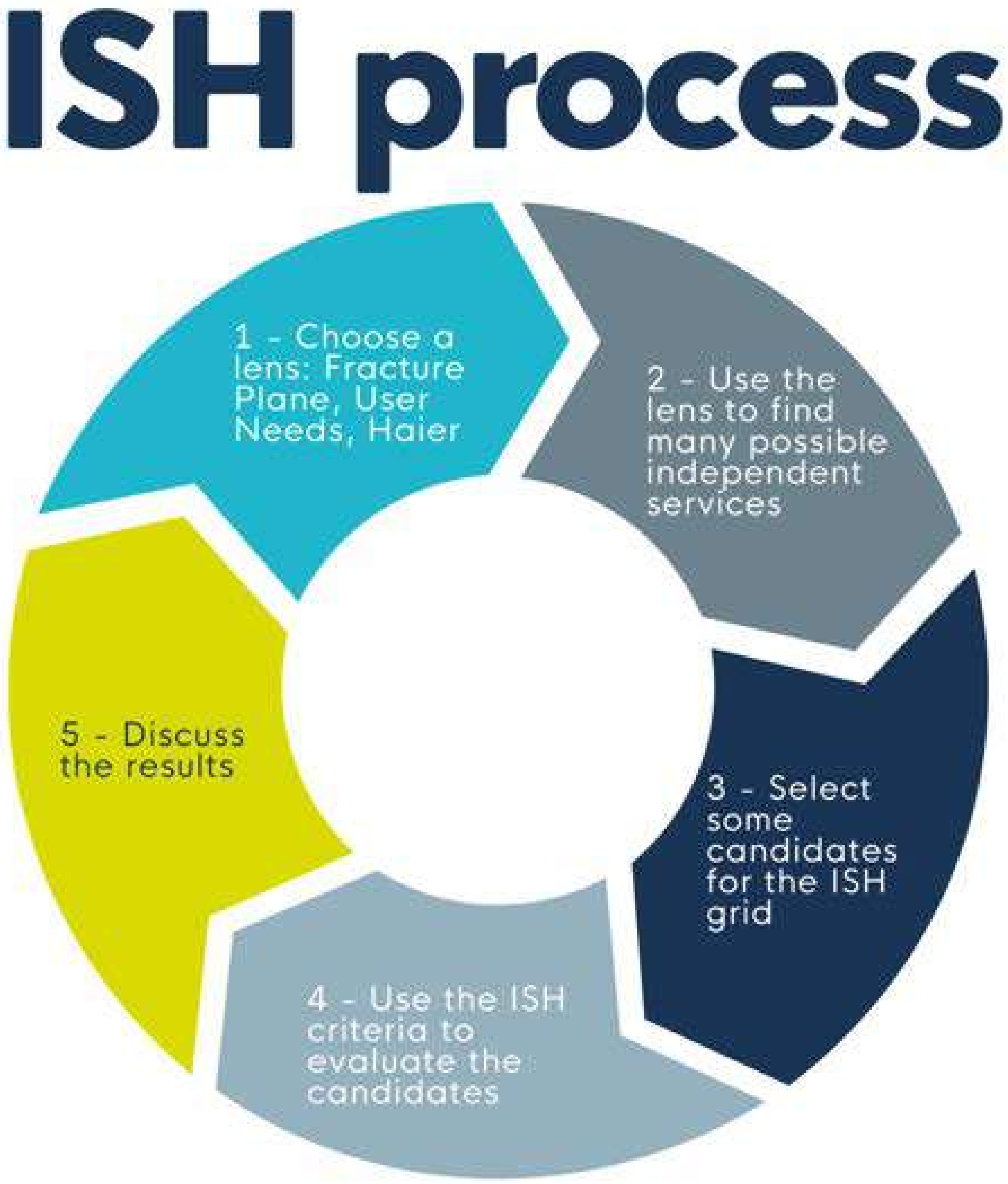
Fracture planes

User needs

Micro-enterprises

ISH process

1. Choose a lens: Fracture Plane, User Needs, Haier
2. Use the lens to find many possible independent services
3. Select some candidates for the ISH grid
4. Use the ISH criteria to evaluate the candidates
5. Discuss the results


ISH process
1 - Choose a lens: Fracture Plane, User Needs, Haier
2 - Use the lens to find many possible independent services
3 - Select some candidates for the ISH grid
4 - Use the ISH criteria to evaluate the candidates
5 - Discuss the results

ISH Checklist: 1 - As-a-Service

Could it make any logical sense to offer this thing "as a service"?

Is this thing independent enough?

Would consumers understand or value it?

Would it simplify execution?

ISH Checklist: 2 - Brand

Can you imagine this thing branded as a public cloud service?
(For example, like AvocadoOnline.com)

Would it be a compelling offering?

Would it be a viable business (or "micro-business") or service?

Could a marketing campaign be convincing?

ISH Checklist: 3 - Revenue / Customers

Could this thing be managed as a viable cloud service in terms of revenue and customers?

Would it be a viable service with a paid offering?

Would it bring recurring revenue with subscription plans?

Is there a clearly-defined customer base or segment?

ISH Checklist: 4 - Cost tracking

Could the organization currently track costs and investment in this thing separately from similar things?

Are the full costs of running this thing transparent or possible to discover?

Is this thing fairly separate, disconnected from other things in the organization?

Does the organization track this separately?

conflux

ISH Checklist: 5 - Data

Is it possible to define clearly the input data (from other sources) that this thing needs?

Is the thing fairly independent from any data sources?

Are the sources internal?

Is the input data clean (not messy)?

Is the input data provided in a self-service way?

ISH Checklist: 6 - User personas

Could this thing have a small/well-defined set of user types or customers (user personas)?

Is the thing meeting specific user needs?

Do we know (or can we easily articulate) these user types and their needs?

conflux

ISH Checklist: 7 - Teams

Could a team or set of teams effectively build and operate a service based on this thing?

Would the cognitive load (breadth of topics/context switching) be bounded to help the team focus and succeed?

Would significant infrastructure or other platform abstractions be unnecessary?

conflux

ISH Checklist: 8 - Dependencies

Would this team be able to act independently of other teams for the majority of the time to achieve their objectives?

Is this thing logically independent from other things?

Could the team "self-serve" dependencies in a nob-blocking manner from a platform?

ISH Checklist: 9 - Impact / Value

Would the scope of this thing provide a team with an impactful and engaging challenge?

Is the scope big enough to provide an impact? Would the scope be engaging for talented people?

Is there sufficient value to customers and the organization that the value would be clearly recognized?

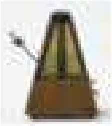
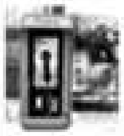
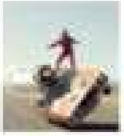
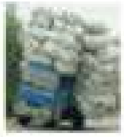
ISH Checklist: 10 - Product decisions

Would the team working on this thing be able to "own" their own product roadmap and the product direction?

Does this thing provide discrete value in a well-defined sphere of execution?

Can the team define their own roadmap based on what they discover is best for the product and its users?

Fracture Planes

Business Domain	"Order" Lead ➡️ Customer
Regulatory Compliance	
Change Cadence	
Technology	
Risk	
Performance Isolation	
User Personas	
Team Location	

techbeacon.com

How to break apart a monolith without destroying your team | TechBeacon

How will breaking apart a monolith affect the teams involved in building your software? Start with the needs of your team.

https://techbeacon.com/app-dev-testing/how-break-apart-monolith-without-destroying-your-team

Candidate streams using the fracture planes lens

Business Domain	New user registration	Collection	Delivery	Special requests	Affiliate referral	?	?
Regulatory Compliance	?						
Change Cadence	?						
Technology	Stripe payments	Credit Card data management					
Risk							
Performance Isolation							
~~User Personas~~	IGNORE FOR NOW						
Team Location	Manchester / London						

Laundry & dry cleaning with 24h delivery in Manchester

Ranked #1 by customers

https://www.laundryheap.co.uk/

Independent Service Heuristics (ISH)

Start by asking “Could this thing be run as a cloud-hosted (SaaS) service or product?”

- Ask the team to think about, and write down aspects of the business, relevant tasks, software applications, and customer journeys which could become an independent domain, service, or value stream.
- Remove duplicates and cluster similar ideas.
- Use this checklist to confirm, or discard areas of focus.

1. **Sense-check**: Could it make any logical sense to offer this thing "as a service"?
    - Is this thing independent enough?
    - Would consumers understand or value it?
    - Would it simplify execution?
2. **Brand**: Could you imagine this thing branded as a public cloud service (like *AvocadoOnline.com* 🥑)?
    - Would it be a viable business (or "micro-business") or service?
    - Would it be a compelling offering?
    - Could a marketing campaign be convincing?
3. **Revenue/Customers**: Could this thing be managed as a viable cloud service in terms of revenue and customers?
    - Would it be viable service with a paid offering?
    - Would it bring recurring revenue with subscription plans?
    - Is there a clearly-defined customer base or segment?
4. **Cost tracking**: Could the organisation currently track costs and investment in this thing separately from similar things?
    - Are the full costs of running this thing transparent or possible to discover considering infrastructure costs, data storage costs, data transfer costs, licence costs, etc.?
    - Is this thing fairly separate, disconnected from other things in the organisation?
    - Does the organisation track this separately?
5. **Data**: Is it possible to define clearly the input data (from other sources) that this thing needs?
    - Is the thing fairly independent from any data sources?
    - Are the sources internal (under our control, not external)?
    - Is the input data clean (not messy)?
    - Is the input data provided in a self-service way? Can the team consume the input data "as a service"?
6. **User Personas**: Could this thing have a small/well-defined set of user types or customers (user personas)?
    - Is the thing meeting specific user needs?
    - Do we know (or can we easily articulate) these user types and their needs?
7. **Teams**: Could a team or set of teams effectively build and operate a service based on this thing?
    - Would the cognitive load (breadth of topics/context switching) be bounded to help the team focus and succeed?
    - Would significant infrastructure or other platform abstractions be unnecessary?
8. **Dependencies**: Would this team be able to act independently of other teams for the majority of the time to achieve their objectives?
    - Is this thing logically independent from other things?
    - Could the team "self-serve" dependencies in a non-blocking manner from a platform?
9. **Impact/Value**: Would the scope of this thing provide a team with an impactful and engaging challenge?
    - Is the scope big enough to provide an impact? Would the scope be engaging for talented people?
    - Is there sufficient value to customers and the organization that the value would be clearly recognized?
10. **Product Decisions**: Would the team working on this thing be able to "own" their own product roadmap and the product direction?
    - Does this thing provide discrete value in a well-defined sphere of execution?
    - Can the team define their own roadmap based on what they discover is best for the product and its users (so that the team is not driven by the requirements and priorities of other teams)?

Answer these questions for each of the candidate streams you have identified. The more 'yes' or 'maybe' answers a possible stream has, the greater the chance that you have found a good candidate for being a separate stream of change.

https://teamtopologies.com/ish

H - evaluation matrix 1

Fracture planes

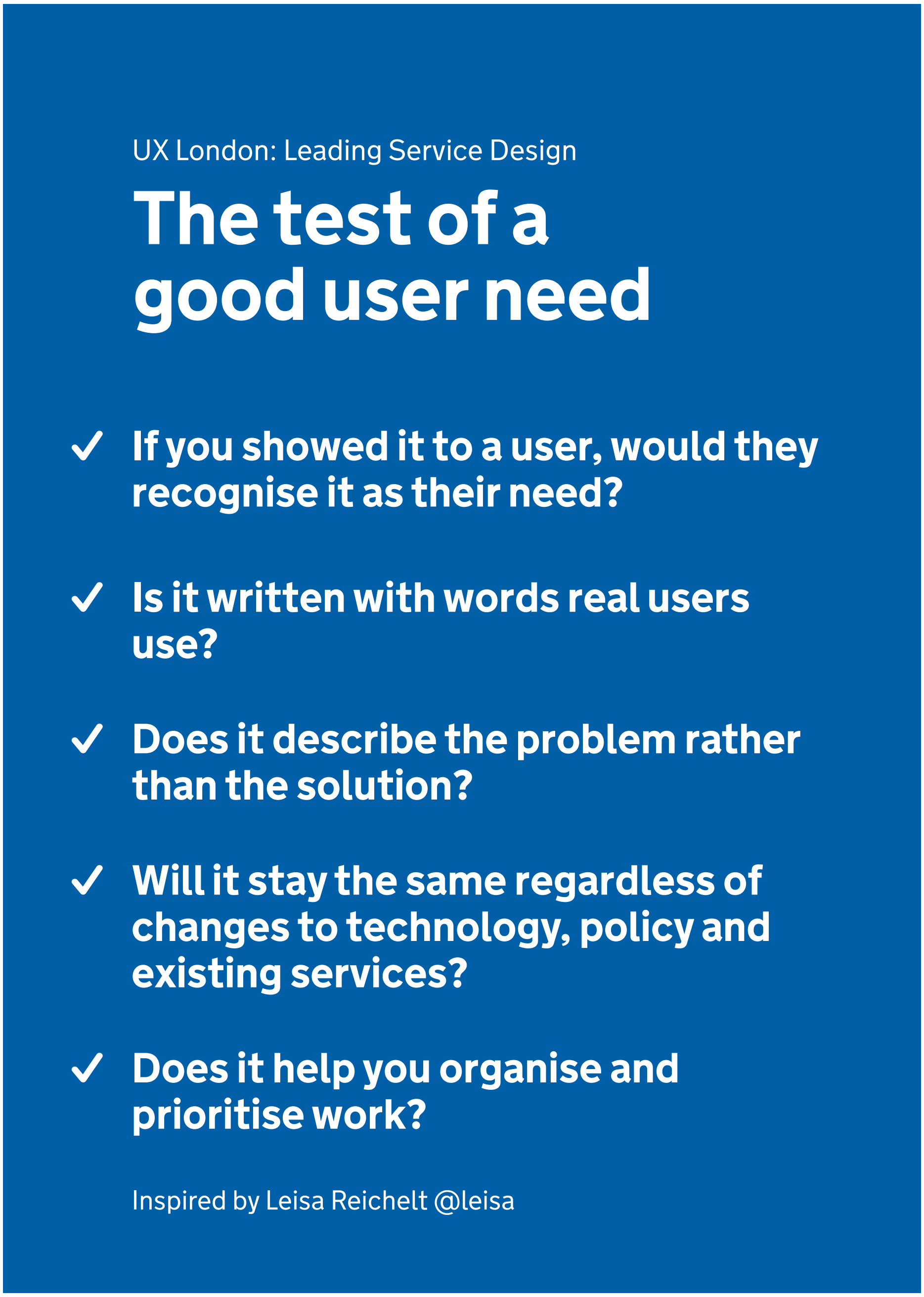
Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes Yes

Maybe Maybe Maybe Maybe Maybe Maybe Maybe Maybe Maybe Maybe Maybe Maybe Maybe Maybe Maybe Maybe Maybe

No No No No No No No No No No No No No No No No No

	Stream candidate	1 As-a-service Could it make any logical sense to offer this thing "as a service"?	2 Brand Could you imagine this thing branded as a public cloud service (like *AvocadoOnline.com* 🥑)?	3 Revenue / Customers Could this thing be managed as a viable cloud service in terms of revenue and customers?	4 Cost Tracking Could the organisation currently track costs and investment in this thing separately from similar things?	5 Data Is it possible to define clearly the input data (from other sources) that this thing needs?	6 User Personas Could this thing have a small/well-defined set of user types or customers (user personas)?	7 Teams Could a team or set of teams effectively build and operate a service based on this thing?	8 Dependencies Would this team be able to act independently of other teams for the majority of the time to achieve their objectives?	9 Impact / Value Would the scope of this thing provide a team with an impactful and engaging challenge?	10 Product Decisions Would the team working on this thing be able to "own" their own product roadmap and the product direction?
A	New user registration	Yes Yes Yes	Yes Maybe Yes	Yes Yes Yes	Maybe Yes No	Maybe Yes	No Yes	Yes Maybe Yes	No No Yes	Yes Yes	Yes
B	Collection	Maybe Yes Maybe Yes	Yes	No No Maybe	No Maybe	No No Maybe	Yes Yes	No	No	No No Maybe	No Yes
C	Affiliate referral	Maybe Maybe	Yes Yes	Yes	Yes Yes	Yes Yes	Yes	Yes Maybe	Yes	Yes	Yes
D	Delivery of cleaned clothing	Yes Yes Maybe	Yes Yes Yes	Yes Yes	Yes	Yes					
E	Stripe payments	Yes Yes	Yes Yes	Yes Yes							
F	Credit Card data management	Yes Maybe Maybe	Maybe Maybe Maybe	Yes Yes Maybe							

Warning flags? 🚩 🚩 🚩

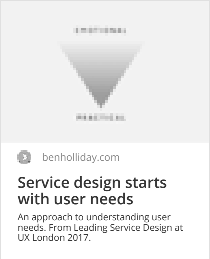


UX London: Leading Service Design

The test of a good user need

- ✓ If you showed it to a user, would they recognise it as their need?
- ✓ Is it written with words real users use?
- ✓ Does it describe the problem rather than the solution?
- ✓ Will it stay the same regardless of changes to technology, policy and existing services?
- ✓ Does it help you organise and prioritise work?

Inspired by Leisa Reichelt @leisa



benholliday.com

Service design starts with user needs

An approach to understanding user needs. From Leading Service Design at UX London 2017.

https://benholliday.com/2017/07/14/leading-service-design-user-needs/

- **explicit needs**: derived from how our users describe what they are trying to do
- **implicit needs**: those that are not expressed and that users are sometimes not aware of, but that are evident from our observation
- **created needs**: where a user has to do something because it is required by the service

needs to a family?

Candidate streams using the user needs lens

	business professional	family	care worker
EXAMPLES			
Whole Service			
Check a status		Track specific items of clothing	
Money-related		Set up a Direct Debit	
Order-related	Book and manage collections Order some more QR code labels for laundry bags		
Discover Information			Manage multiple collections for different people
Trigger an action			
…			

Laundry & dry cleaning with 24h delivery in Manchester

Ranked #1 by customers

https://www.laundryheap.co.uk/

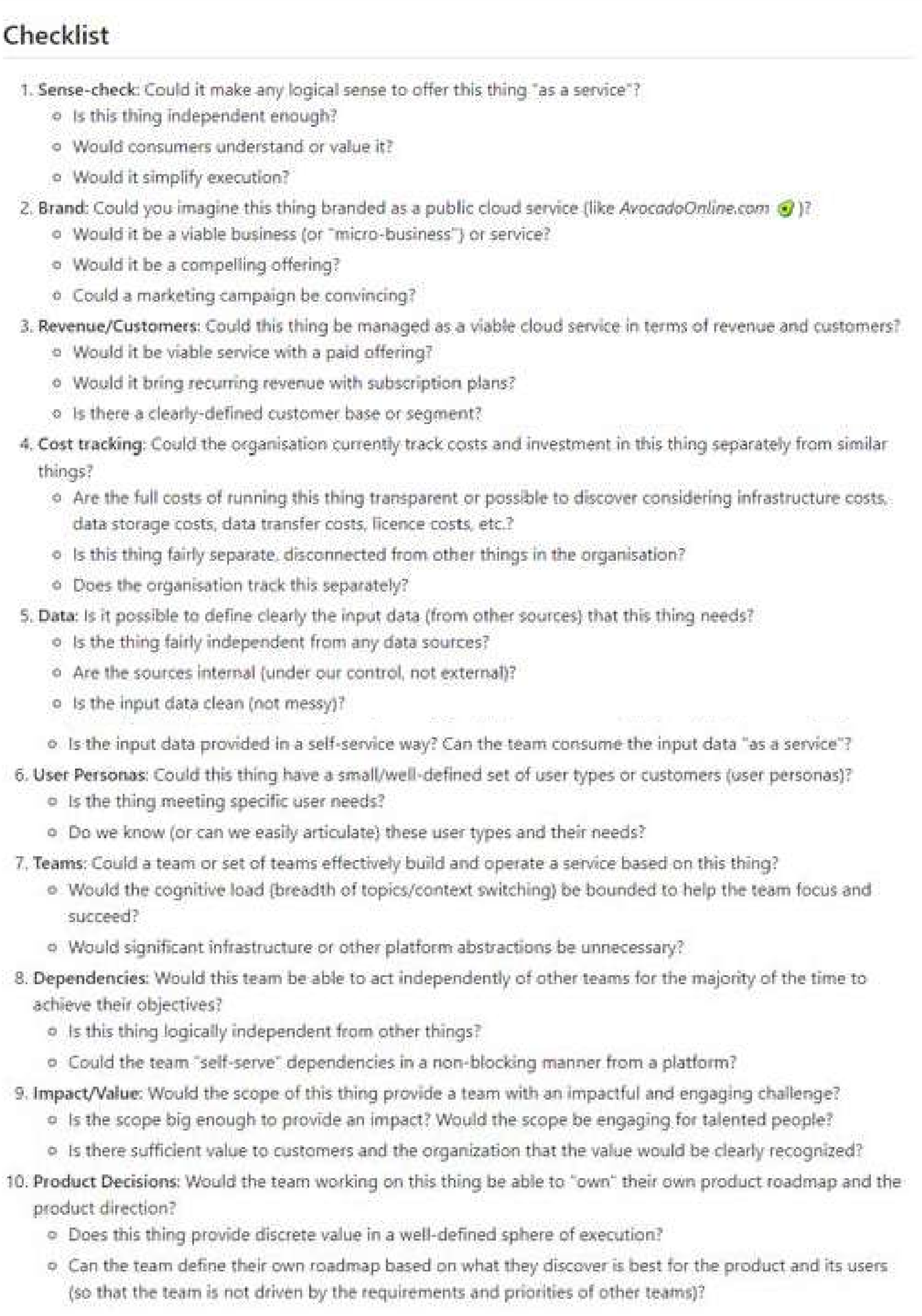
?

?

?

conflux

Independent Service Heuristics



Checklist

1. **Sense-check:** Could it make any logical sense to offer this thing "as a service"?
    - Is this thing independent enough?
    - Would consumers understand or value it?
    - Would it simplify execution?
2. **Brand:** Could you imagine this thing branded as a public cloud service (like *AvocadoOnline.com* 🥑)?
    - Would it be a viable business (or "micro-business") or service?
    - Would it be a compelling offering?
    - Could a marketing campaign be convincing?
3. **Revenue/Customers:** Could this thing be managed as a viable cloud service in terms of revenue and customers?
    - Would it be viable service with a paid offering?
    - Would it bring recurring revenue with subscription plans?
    - Is there a clearly-defined customer base or segment?
4. **Cost tracking:** Could the organisation currently track costs and investment in this thing separately from similar things?
    - Are the full costs of running this thing transparent or possible to discover considering infrastructure costs, data storage costs, data transfer costs, licence costs, etc.?
    - Is this thing fairly separate, disconnected from other things in the organisation?
    - Does the organisation track this separately?
5. **Data:** Is it possible to define clearly the input data (from other sources) that this thing needs?
    - Is the thing fairly independent from any data sources?
    - Are the sources internal (under our control, not external)?
    - Is the input data clean (not messy)?
    - Is the input data provided in a self-service way? Can the team consume the input data "as a service"?
6. **User Personas:** Could this thing have a small/well-defined set of user types or customers (user personas)?
    - Is the thing meeting specific user needs?
    - Do we know (or can we easily articulate) these user types and their needs?
7. **Teams:** Could a team or set of teams effectively build and operate a service based on this thing?
    - Would the cognitive load (breadth of topics/context switching) be bounded to help the team focus and succeed?
    - Would significant infrastructure or other platform abstractions be unnecessary?
8. **Dependencies:** Would this team be able to act independently of other teams for the majority of the time to achieve their objectives?
    - Is this thing logically independent from other things?
    - Could the team "self-serve" dependencies in a non-blocking manner from a platform?
9. **Impact/Value:** Would the scope of this thing provide a team with an impactful and engaging challenge?
    - Is the scope big enough to provide an impact? Would the scope be engaging for talented people?
    - Is there sufficient value to customers and the organization that the value would be clearly recognized?
10. **Product Decisions:** Would the team working on this thing be able to "own" their own product roadmap and the product direction?
    - Does this thing provide discrete value in a well-defined sphere of execution?
    - Can the team define their own roadmap based on what they discover is best for the product and its users (so that the team is not driven by the requirements and priorities of other teams)?

ISH - evaluation matrix 2

User needs

Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe
No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No

	Stream candidate	1 As-a-service Could it make any logical sense to offer this thing "as a service"?	2 Brand Could you imagine this thing branded as a public cloud service (like *AvocadoOnline.com* 🥑)?	3 Revenue / Customers Could this thing be managed as a viable cloud service in terms of revenue and customers?	4 Cost Tracking Could the organisation currently track costs and investment in this thing separately from similar things?	5 Data Is it possible to define clearly the input data (from other sources) that this thing needs?	6 User Personas Could this thing have a small/well-defined set of user types or customers (user personas)?	7 Teams Could a team or set of teams effectively build and operate a service based on this thing?	8 Dependencies Would this team be able to act independently of other teams for the majority of the time to achieve their objectives?	9 Impact / Value Would the scope of this thing provide a team with an impactful and engaging challenge?	10 Product Decisions Would the team working on this thing be able to "own" their own product roadmap and the product direction?
G	Set up a Direct Debit	Yes Yes Yes	Yes Maybe Yes	Yes Maybe Yes	Yes Maybe Maybe	Maybe Yes No	Yes Yes Yes	Yes Maybe Yes	No Yes Yes	Yes Yes Yes	Maybe Yes Yes
H	Book and manage collections										
I	Manage multiple collections for different people										
J											
K											
L											

Warning flags? 🚩

🚩

🚩

Selected candidates 2

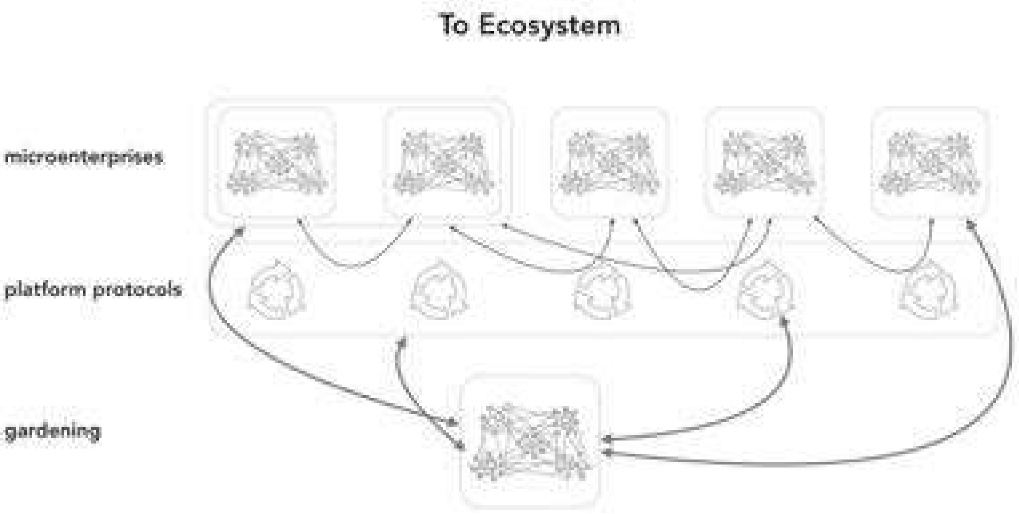
Haier Microenterprise

medium.com

Evolution of the Platform Organization: 3 Haier, Rendanheyi, and Zhang Ruimin's Vision

Haier's Chairman, Zhang Ruimin, is one of the world's most well-known management thinkers, at least in the rarefied world of business...

https://medium.com/work-futures/evolution-of-the-platform-organization-3-haier-rendanheyi-and-zhang-ruimins-vision-d8afceef7f5e



Candidate streams using micro-enterprise lens

Customer-facing

White glove service for High Net Worth clients

Underlying service via partner

Delivery and Collection

Automatic wash type sorting based on QR code

Enabling

?

Laundry care

?

Subject Matter Experts (SME)

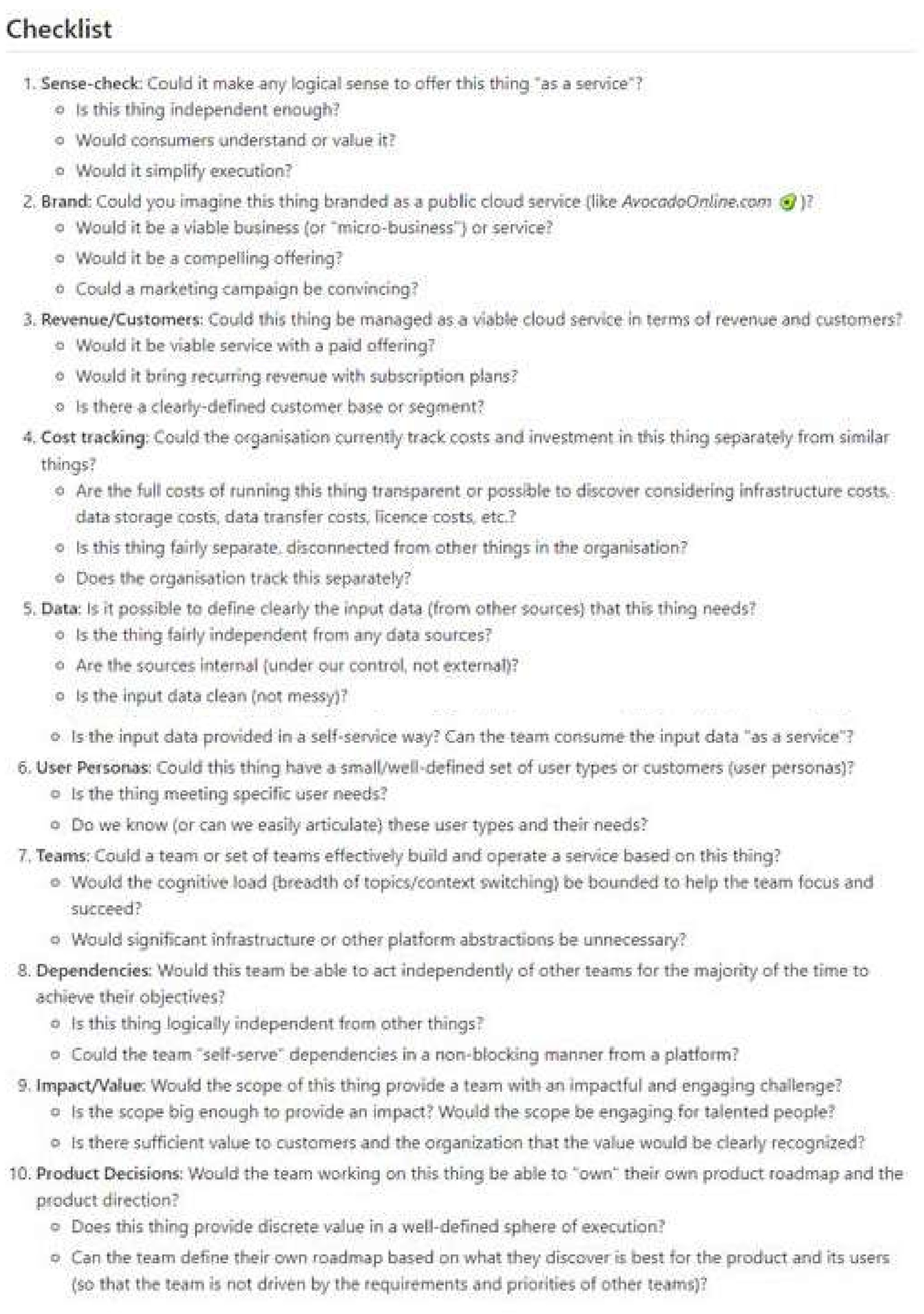
...

Laundry & dry cleaning with 24h delivery in Manchester

Ranked #1 by customers

https://www.laundryheap.co.uk/

Independent Service Heuristics



Checklist

1. **Sense-check:** Could it make any logical sense to offer this thing "as a service"?
   - Is this thing independent enough?
   - Would consumers understand or value it?
   - Would it simplify execution?
2. **Brand:** Could you imagine this thing branded as a public cloud service (like *AvocadoOnline.com* 🥑)?
   - Would it be a viable business (or "micro-business") or service?
   - Would it be a compelling offering?
   - Could a marketing campaign be convincing?
3. **Revenue/Customers:** Could this thing be managed as a viable cloud service in terms of revenue and customers?
   - Would it be viable service with a paid offering?
   - Would it bring recurring revenue with subscription plans?
   - Is there a clearly-defined customer base or segment?
4. **Cost tracking:** Could the organisation currently track costs and investment in this thing separately from similar things?
   - Are the full costs of running this thing transparent or possible to discover considering infrastructure costs, data storage costs, data transfer costs, licence costs, etc.?
   - Is this thing fairly separate, disconnected from other things in the organisation?
   - Does the organisation track this separately?
5. **Data:** Is it possible to define clearly the input data (from other sources) that this thing needs?
   - Is the thing fairly independent from any data sources?
   - Are the sources internal (under our control, not external)?
   - Is the input data clean (not messy)?
   - Is the input data provided in a self-service way? Can the team consume the input data "as a service"?
6. **User Personas:** Could this thing have a small/well-defined set of user types or customers (user personas)?
   - Is the thing meeting specific user needs?
   - Do we know (or can we easily articulate) these user types and their needs?
7. **Teams:** Could a team or set of teams effectively build and operate a service based on this thing?
   - Would the cognitive load (breadth of topics/context switching) be bounded to help the team focus and succeed?
   - Would significant infrastructure or other platform abstractions be unnecessary?
8. **Dependencies:** Would this team be able to act independently of other teams for the majority of the time to achieve their objectives?
   - Is this thing logically independent from other things?
   - Could the team "self-serve" dependencies in a non-blocking manner from a platform?
9. **Impact/Value:** Would the scope of this thing provide a team with an impactful and engaging challenge?
   - Is the scope big enough to provide an impact? Would the scope be engaging for talented people?
   - Is there sufficient value to customers and the organization that the value would be clearly recognized?
10. **Product Decisions:** Would the team working on this thing be able to "own" their own product roadmap and the product direction?
    - Does this thing provide discrete value in a well-defined sphere of execution?
    - Can the team define their own roadmap based on what they discover is best for the product and its users (so that the team is not driven by the requirements and priorities of other teams)?

ISH - evaluation matrix 3

Micro-enterprises

Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe	Maybe
No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No

	Stream candidate	1 Sense-check Could it make any logical sense to offer this thing "as a service"?	2 Brand Could you imagine this thing branded as a public cloud service (like *AvocadoOnline.com* 🥑)?	3 Revenue / Customers Could this thing be managed as a viable cloud service in terms of revenue and customers?	4 Cost Tracking Could the organisation currently track costs and investment in this thing separately from similar things?	5 Data Is it possible to define clearly the input data (from other sources) that this thing needs?	6 User Personas Could this thing have a small/well-defined set of user types or customers (user personas)?	7 Teams Could a team or set of teams effectively build and operate a service based on this thing?	8 Dependencies Would this team be able to act independently of other teams for the majority of the time to achieve their objectives?	9 Impact / Value Would the scope of this thing provide a team with an impactful and engaging challenge?	10 Product Decisions Would the team working on this thing be able to "own" their own product roadmap and the product direction?
M	White glove service for High Net Worth clients										
N											
O											
P											
Q											
R											

Warning flags? 🚩 🚩 🚩

Selected candidates 3

Selected candidate streams for further investigation

Fracture planes

User needs

Micro-enterprises

What to do next with these candidates?

Learn how to apply Independent Service Heuristics in more depth

Domain-Driven Design

Use EventStorming

Deep Dive into 1 or 2 domains

e.g. better define the data or interfaces in one area

End-to-End "walking skeleton"

Part 4 - Skill paths and aptitudes for fast flow

conflux

Outline of part 4

- We begin by exploring the skills pathways needed for typical roles in IT and software delivery when embarking on a fast flow transition.
- We then discuss challenges and opportunities around skills gaps and aptitudes and how we can use market sensing techniques to help avoid skills stagnation.
- We finish Part 4 by reviewing the success patterns covered throughout the Masterclass and answering any remaining questions.

Flows of change

Almost ALL roles and teams should be focused on either:

A flow of change 🔀

Supporting flows of change 🤲

A flow of change

- Software changes to a service or application
- Configuration changes to commercial off-the-shelf (COTS) software
- Onboarding a new employee
- Reviewing legal contracts
- Installing audio-visual equipment

Supporting flows of change

- An infrastructure platform for clouds services or applications
- A data platform for analytics
- A wiki or How-To guide
- Real-time data for decision making

What do we aim for in the context of fast flow?

Continuously 'untangle' business concepts

Minimize hands-offs

Avoid blocking dependencies

Find and adjust team and system boundaries for flow

Move some decision-making to teams

What do we aim for in the context of fast flow?

Long-lived, autonomous teams

Ownership and curation

Modern, digital product management

Consider team cognitive load

Implications of fast flow for teams and roles

- Flow unblocker
- Boundary finder
- Platform to reduce cognitive load
- Product management for holistic experiences
- …

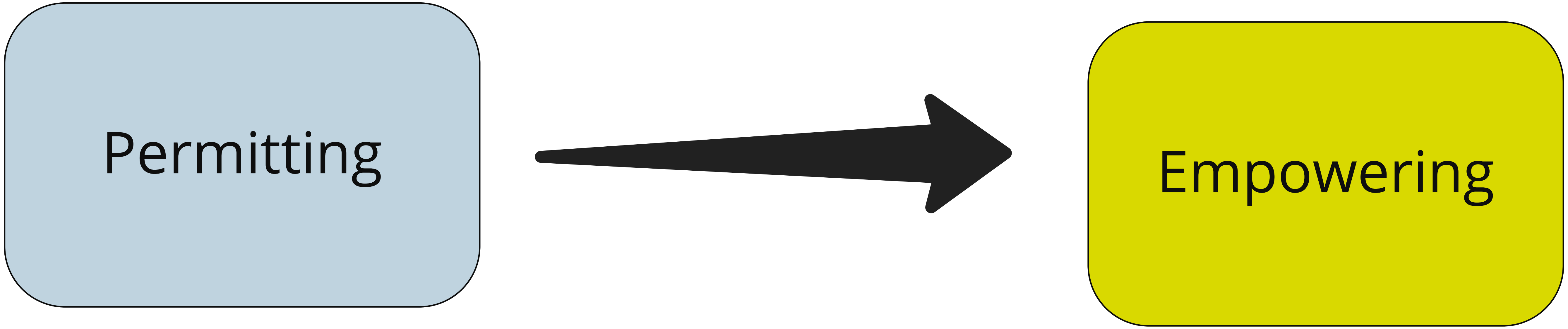
Mindset shifts for 3 example roles

Compliance expert

Architect

Manager

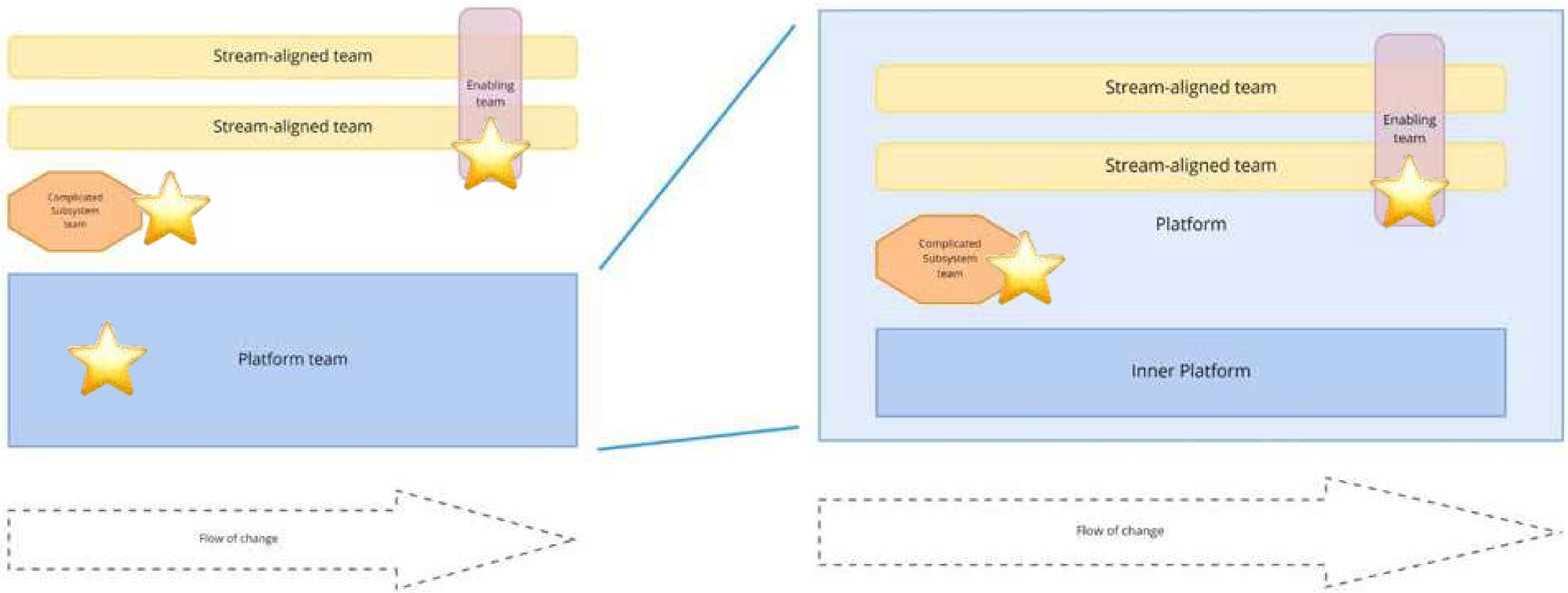
Compliance mindset shift



Permitting

Empowering

Example: compliance expert (empowering)



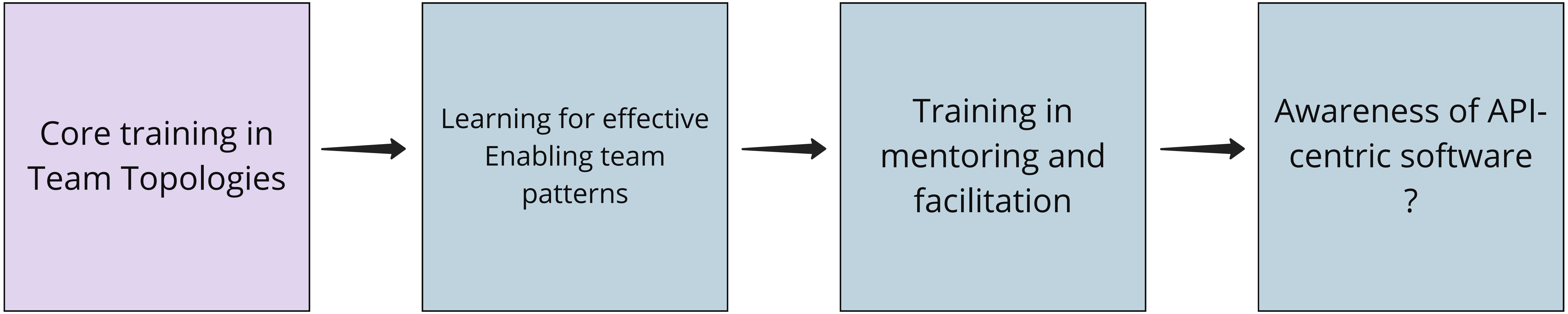
- Part of an enabling team working across many stream-aligned teams
- Part of a platform, advising on the evolution of platform services
- Part of a complicated sub-system team, building a compliance service

Mindset shift for compliance expert - expert view

Individual / team	⚠️ Working as part of a team	⚠️ Working as part of an Enabling team specifically	⚠️ Working in a facilitating way	⚠️ Understanding the importance of fast flow in general	⚠️ Understanding the importance of APIs for fast flow
Organization	⚠️ Aligning incentives and compliance responsibilities	⚠️ Building trust across departments / divisions			

➡️ Empowering other teams

Example skill path for financial compliance expert



Core training in Team Topologies

Learning for effective Enabling team patterns

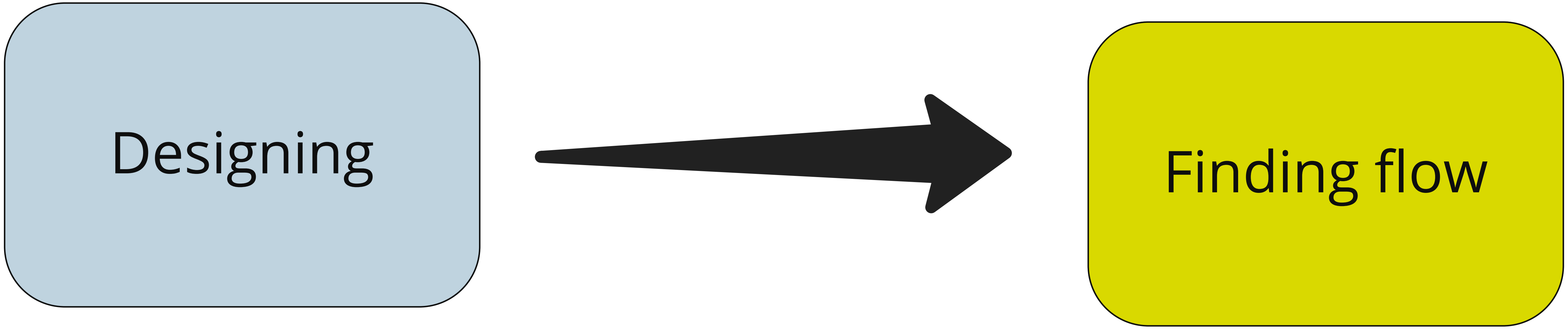
Training in mentoring and facilitation

Awareness of API-centric software
?

➡️ Empowering other teams

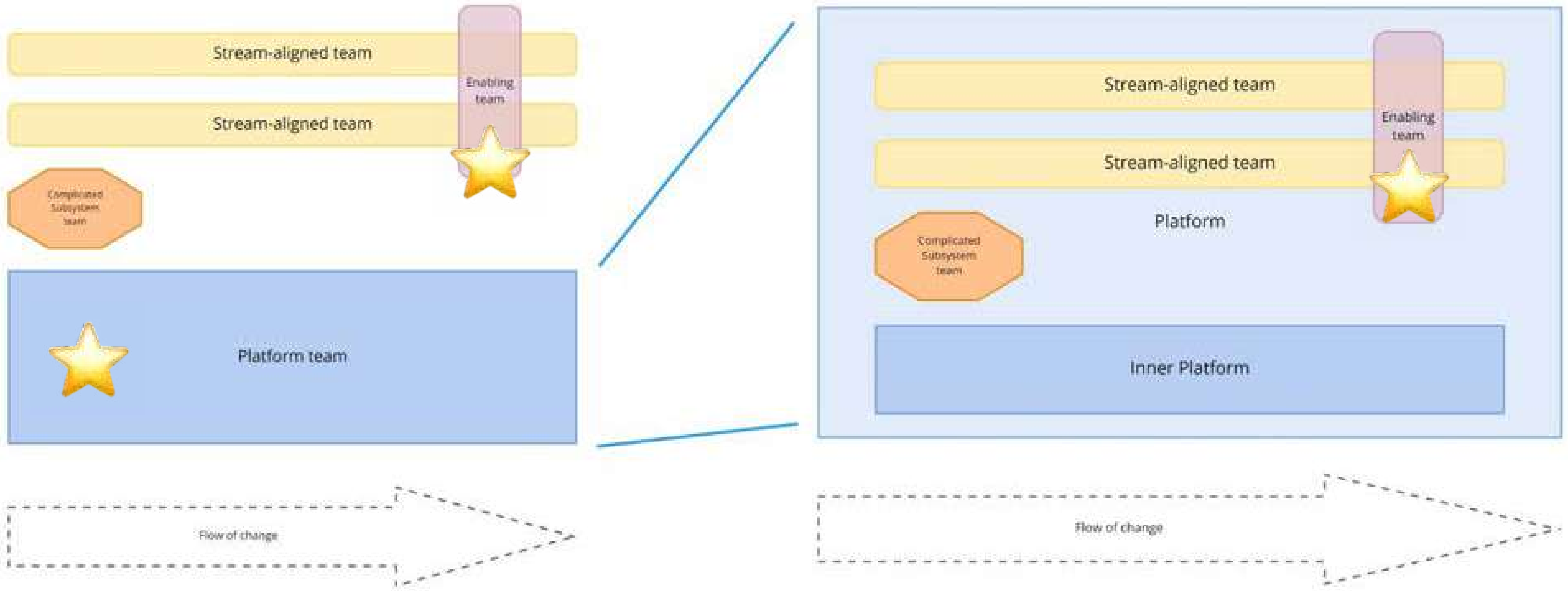
conflux

Architect mindset shift


Designing

Finding flow

Example: enterprise architect (finding flow)



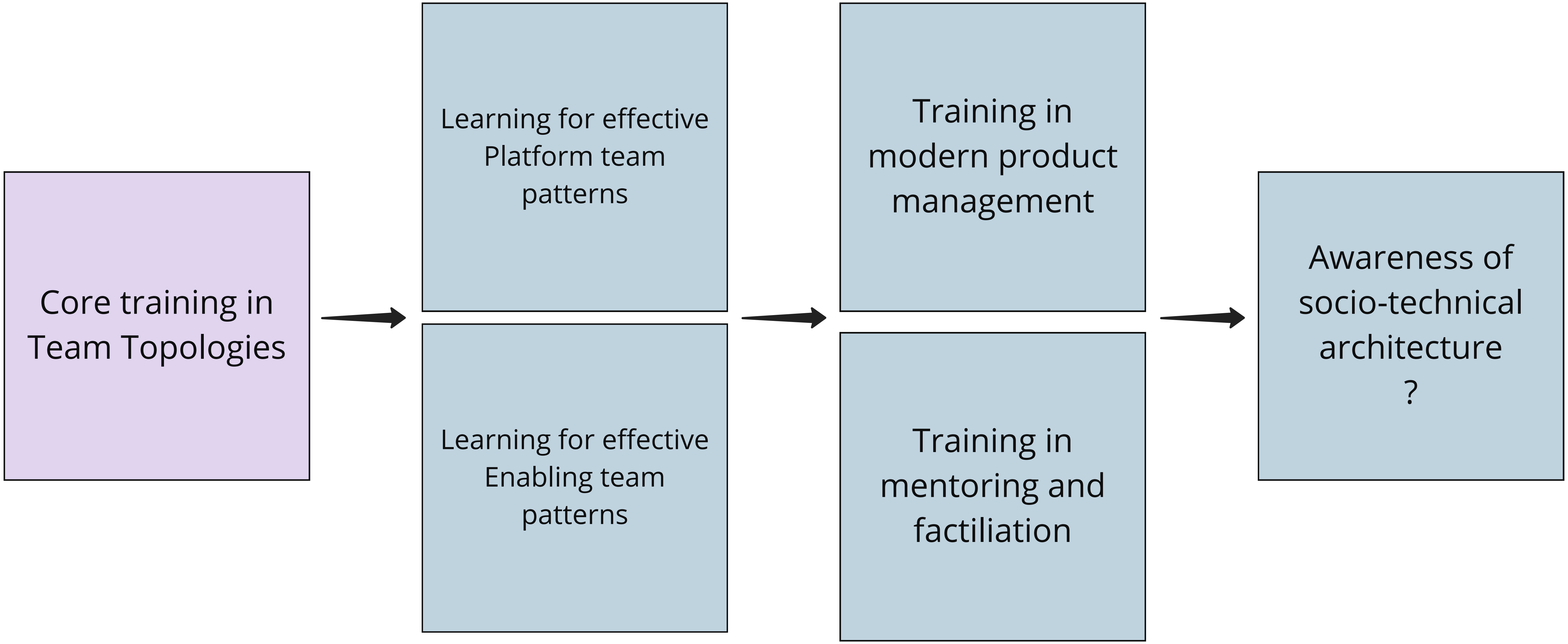
- Part of an Enabling team working across many Stream aligned teams
- Part of a Platform, guiding the platform evolution

Mindset shift for enterprise architect - expert view

Individual / team	⚠ Working as part of a team	⚠ Working as part of an Enabling team specifically	⚠ Working in a facilitating way	⚠ Working as part of a product platform	⚠ Sociotechnical architecture
Organization	⚠ Seeing architecture as sociotechnical - bridging departments / divisions	⚠ Building trust across departments / divisions			

➡ Focus on finding flow

Example skill path for enterprise architect



Core training in Team Topologies

Learning for effective Platform team patterns

Learning for effective Enabling team patterns

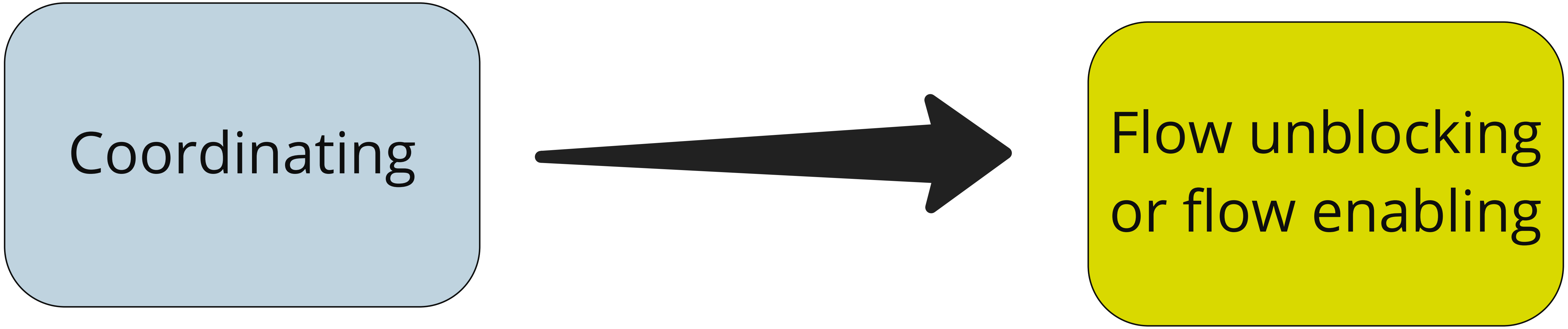
Training in modern product management

Training in mentoring and factiliation

Awareness of socio-technical architecture
?

➡️ Focus on finding flow

Manager mindset shift



Coordinating

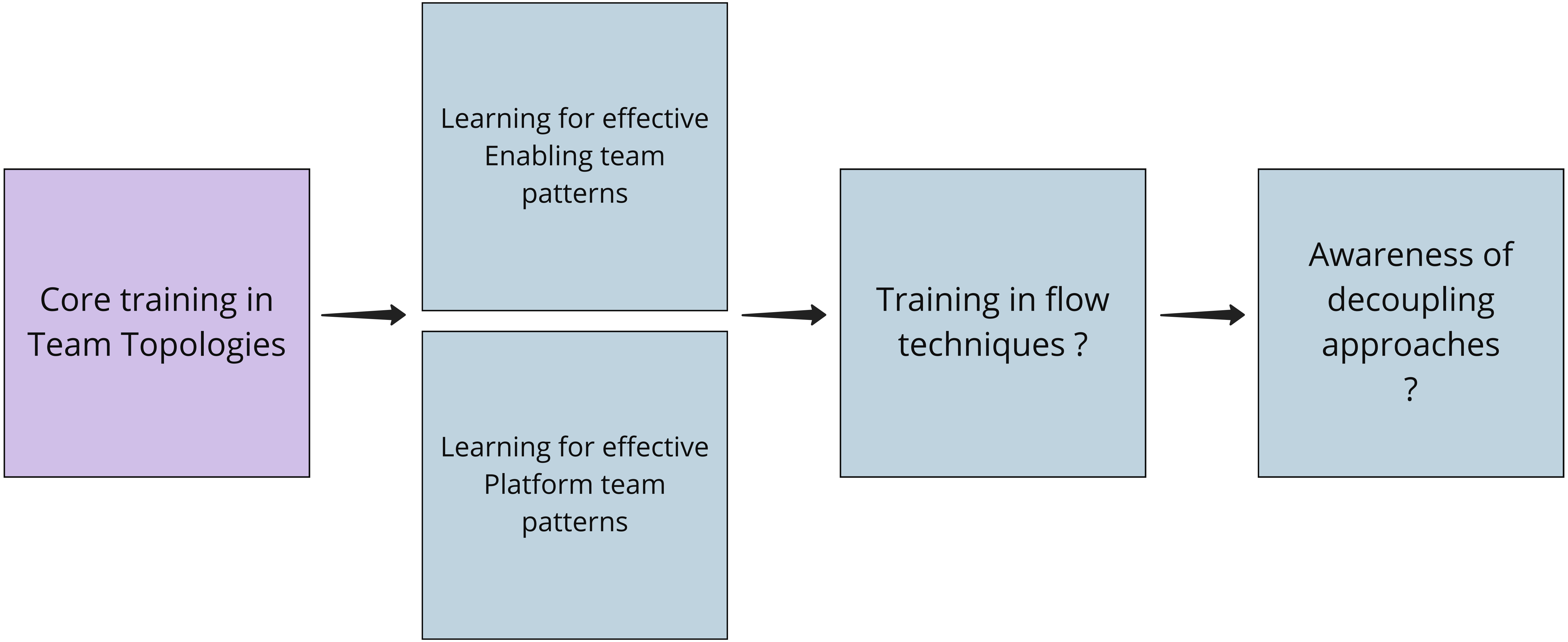
Flow unblocking or flow enabling

Mindset shift for manager - expert view

Individual / team	⚠️ Working as part of a team	⚠️ Working as part of an Enabling team specifically	⚠️ Working in a facilitating way	⚠️ Working as part of a product platform	⚠️ Flow techniques
Organization	⚠️ Seeing "management" as mostly about flow, not coordination	⚠️ Building trust across departments / divisions			

➡️ Focus on unblocking flow

Example skill path for manager



Core training in Team Topologies

Learning for effective Enabling team patterns

Learning for effective Platform team patterns

Training in flow techniques ?

Awareness of decoupling approaches ?

➡️ Focus on unblocking flow

conflux

Skills gaps as an advantage - market sensing

Use skills gaps and aptitudes as a strategic advantage to avoid stagnation

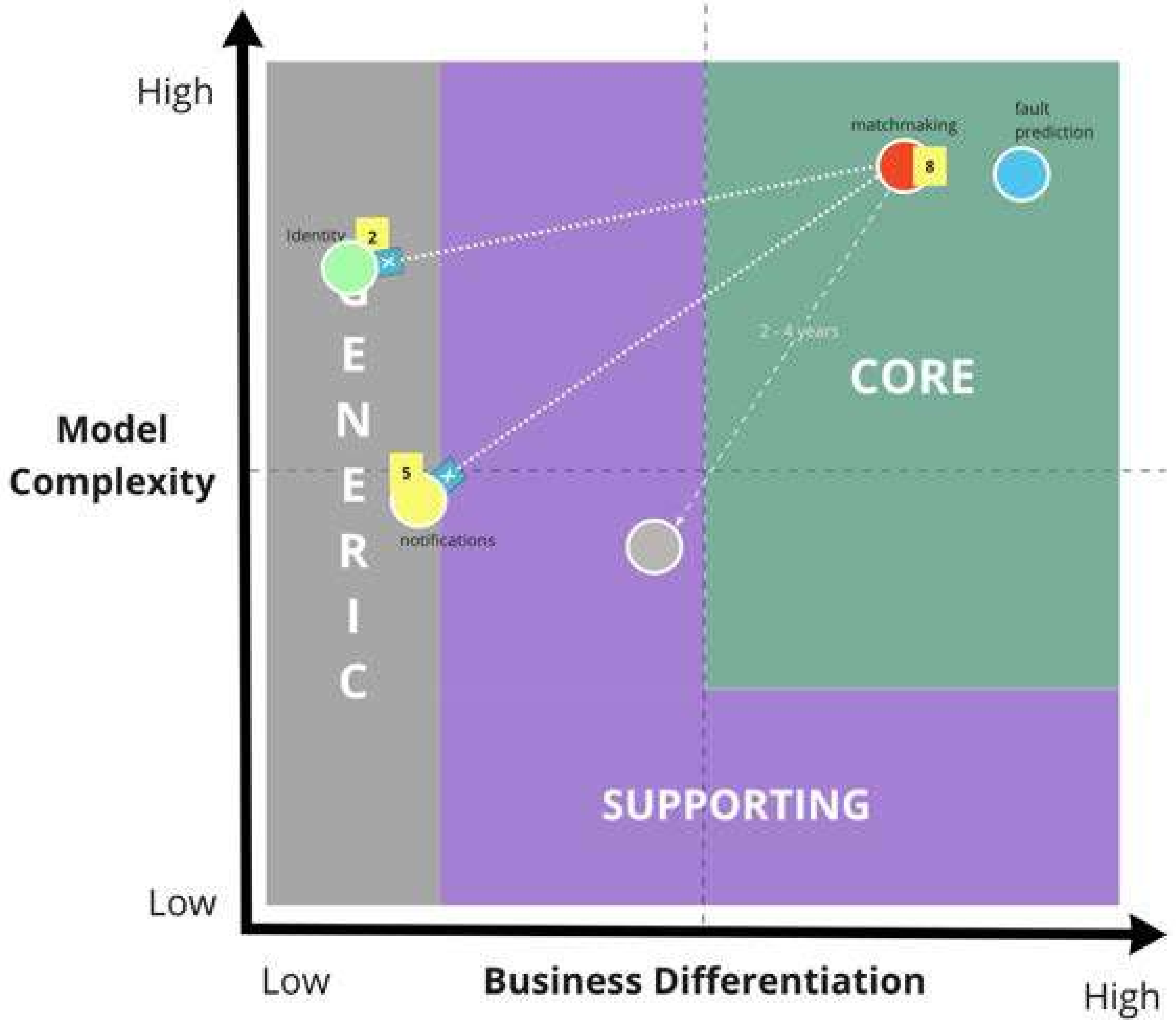
Core Domain Charts - skills

Notation

- 🔴 Sub-domain (current position)
- ● Sub-domain (future position)
- ⇢ Movement over time
- 🔵 Big-bet sub-domain
- 🟡 Platform sub-domain
- 🟢 Outsourced/Purchased sub-domain

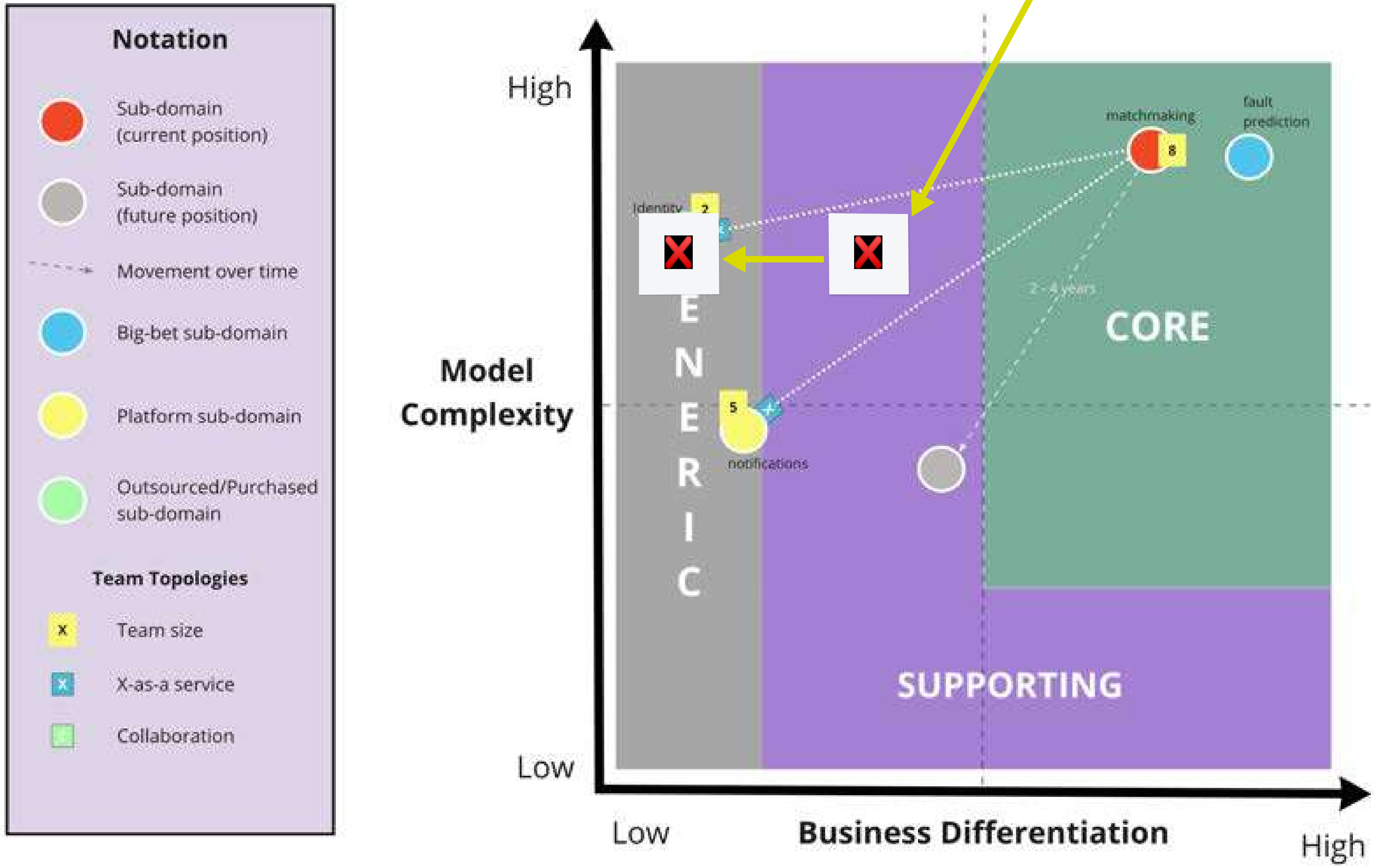
Team Topologies

- 🟨 x Team size
- 🟦 x X-as-a service
- 🟩 Collaboration



Core Domain Charts - skills adaptability

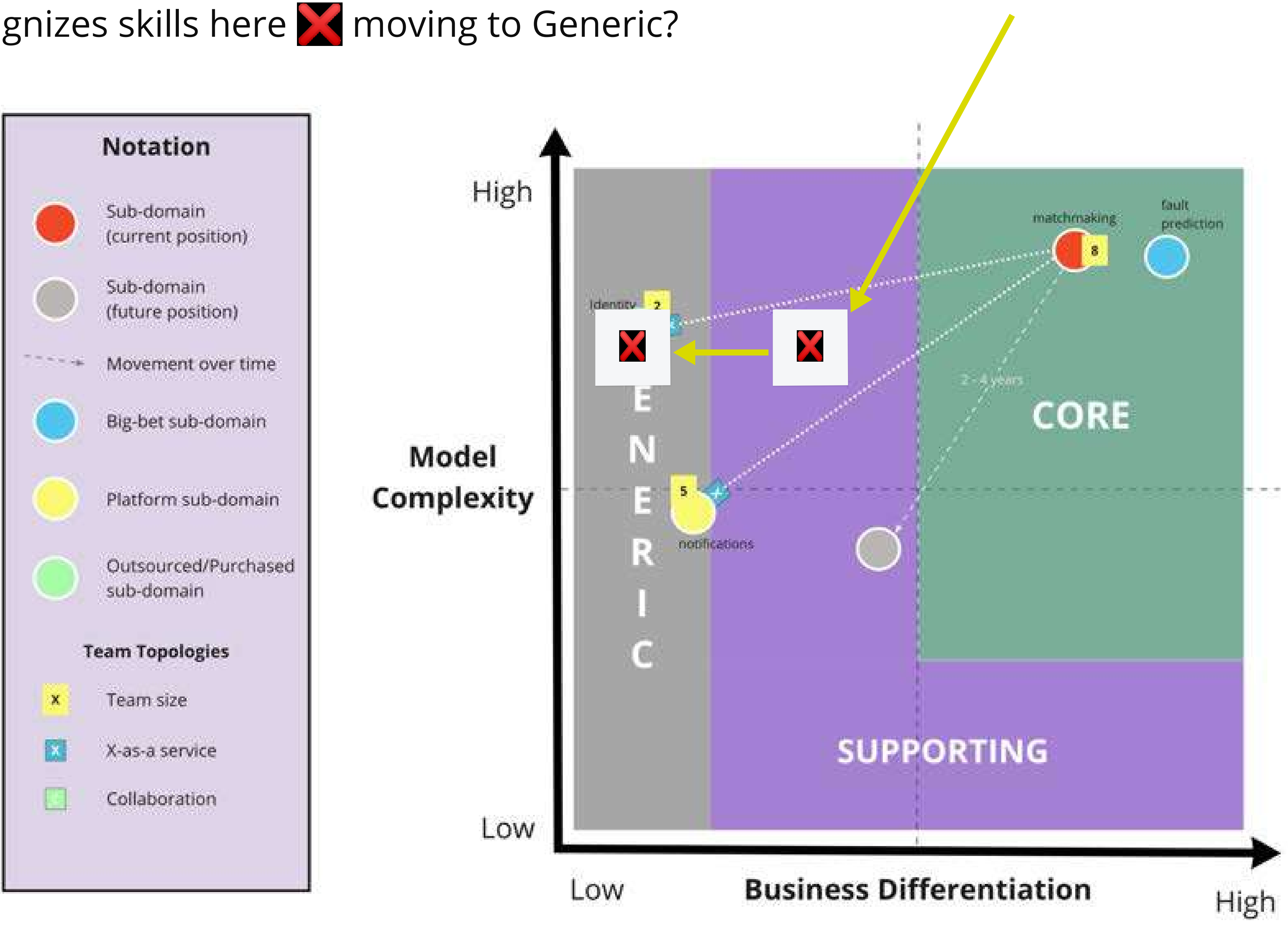
What should a smart organization do around skills and capabilities if it recognizes skills here ❌ moving to Generic?



Example: Capacity planning for data centers --> cloud

Core Domain Charts - skills adaptability - expert opinion

What should a smart organization do around skills and capabilities if it recognizes skills here ❌ moving to Generic?



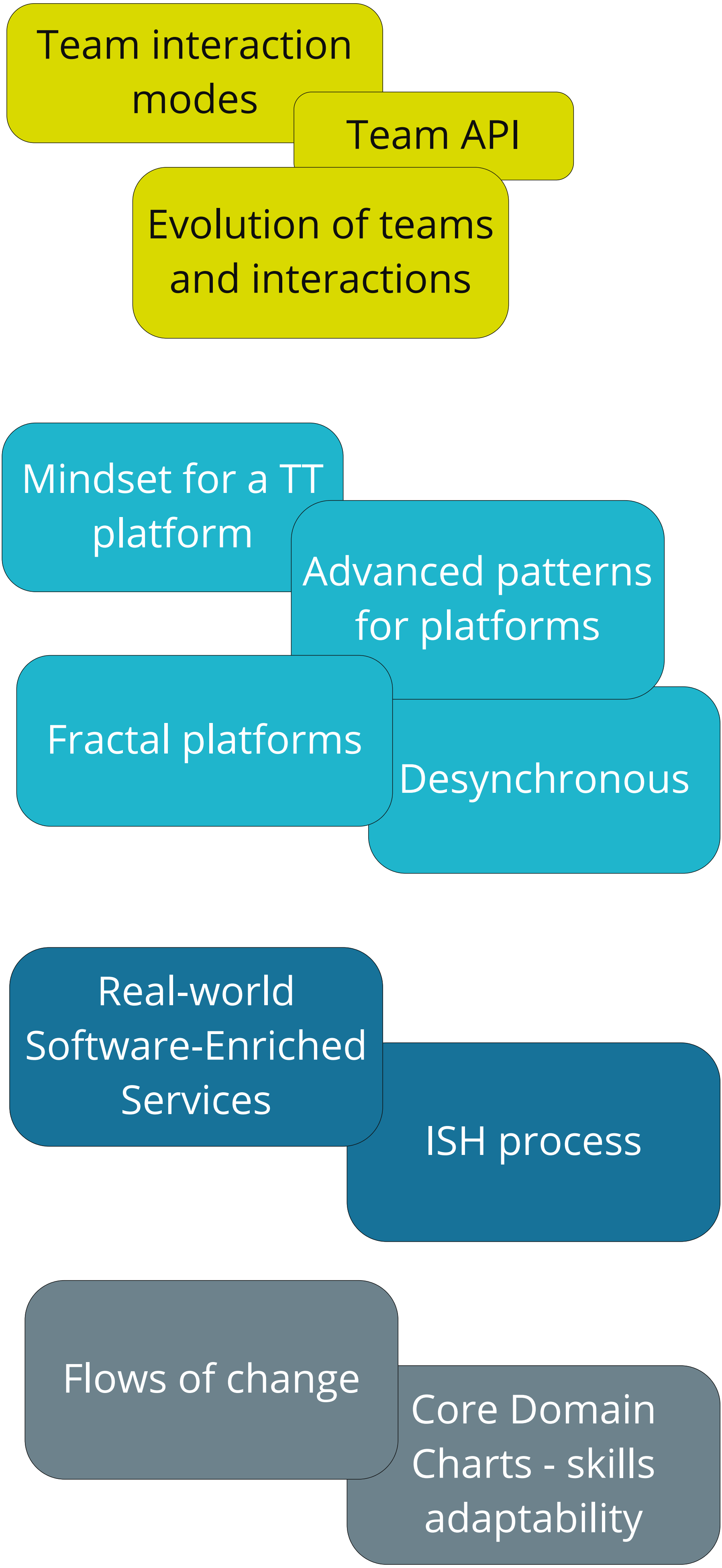
Prefer external Generic to internal Generic

Cross-train people ASAP

Adaptability is perhaps the most important skill

Move from "this is who I am" to "this is what I do now"

HR: Make job descriptions and titles more generic: "engineer", "manager", "coach", etc.



Team interaction modes

Team API

Evolution of teams and interactions

Mindset for a TT platform

Advanced patterns for platforms

Fractal platforms

Desynchronous

Real-world Software-Enriched Services

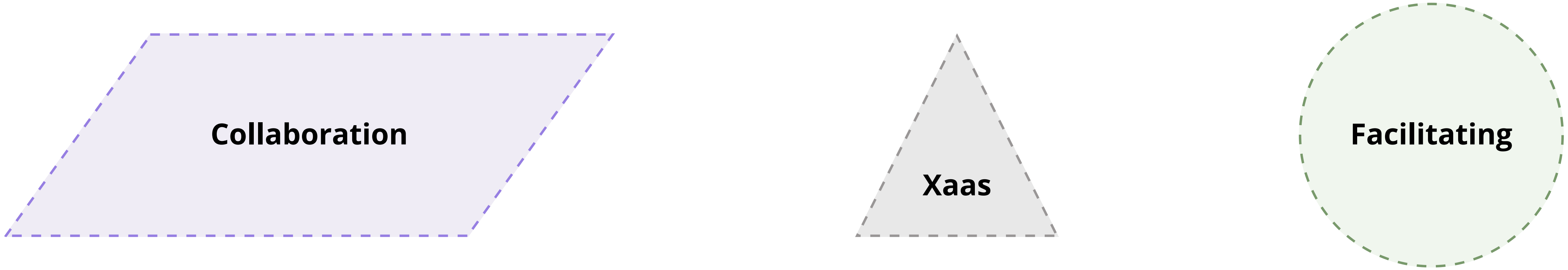
ISH process

Flows of change

Core Domain Charts - skills adaptability

Team interaction modes 2

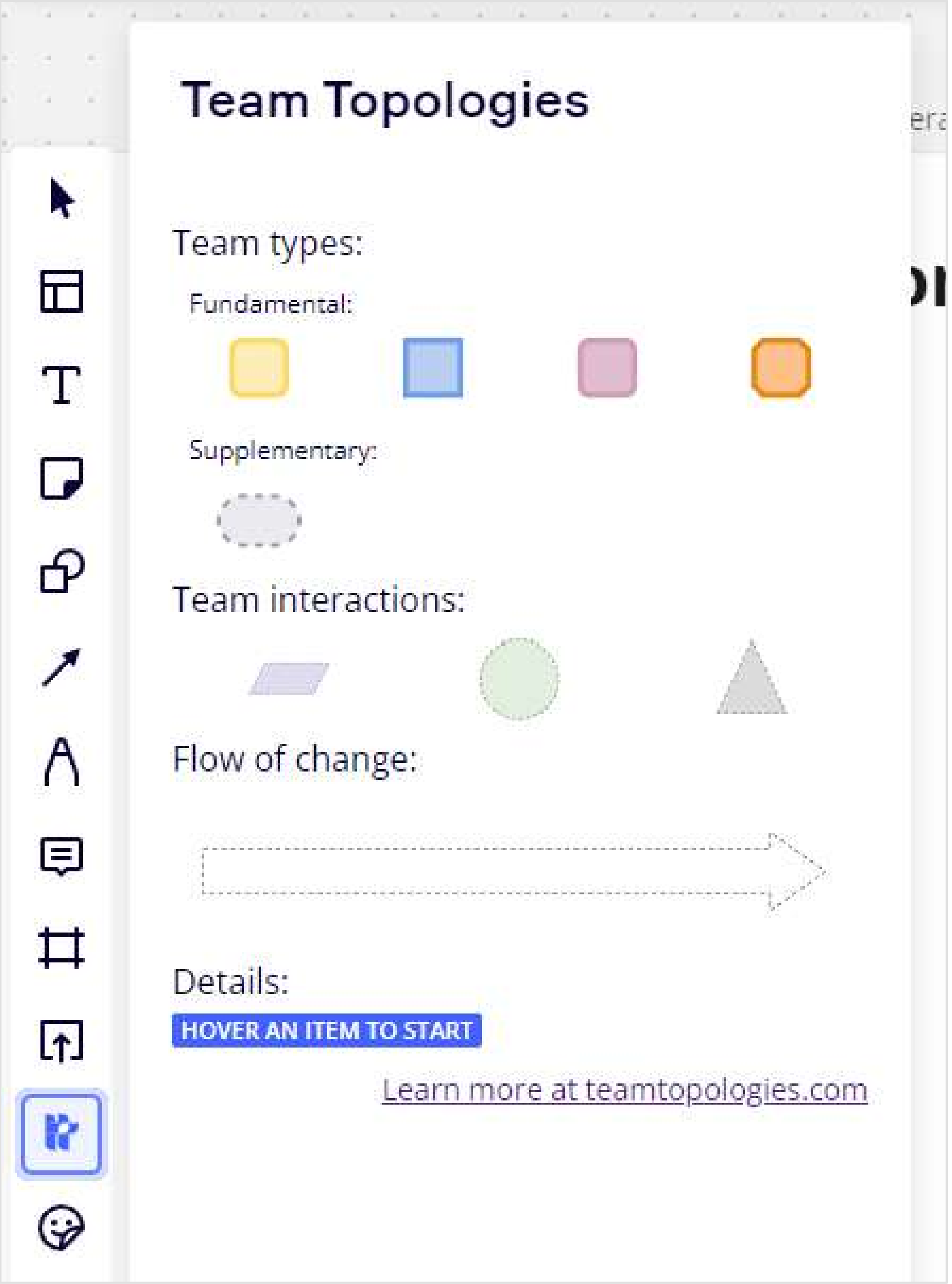
Using the digital TT team modeling shapes



https://shapes.teamtopologies.com/

conflux

Evolution of teams and interactions



Team Topologies

Team types:

Fundamental:

Supplementary:

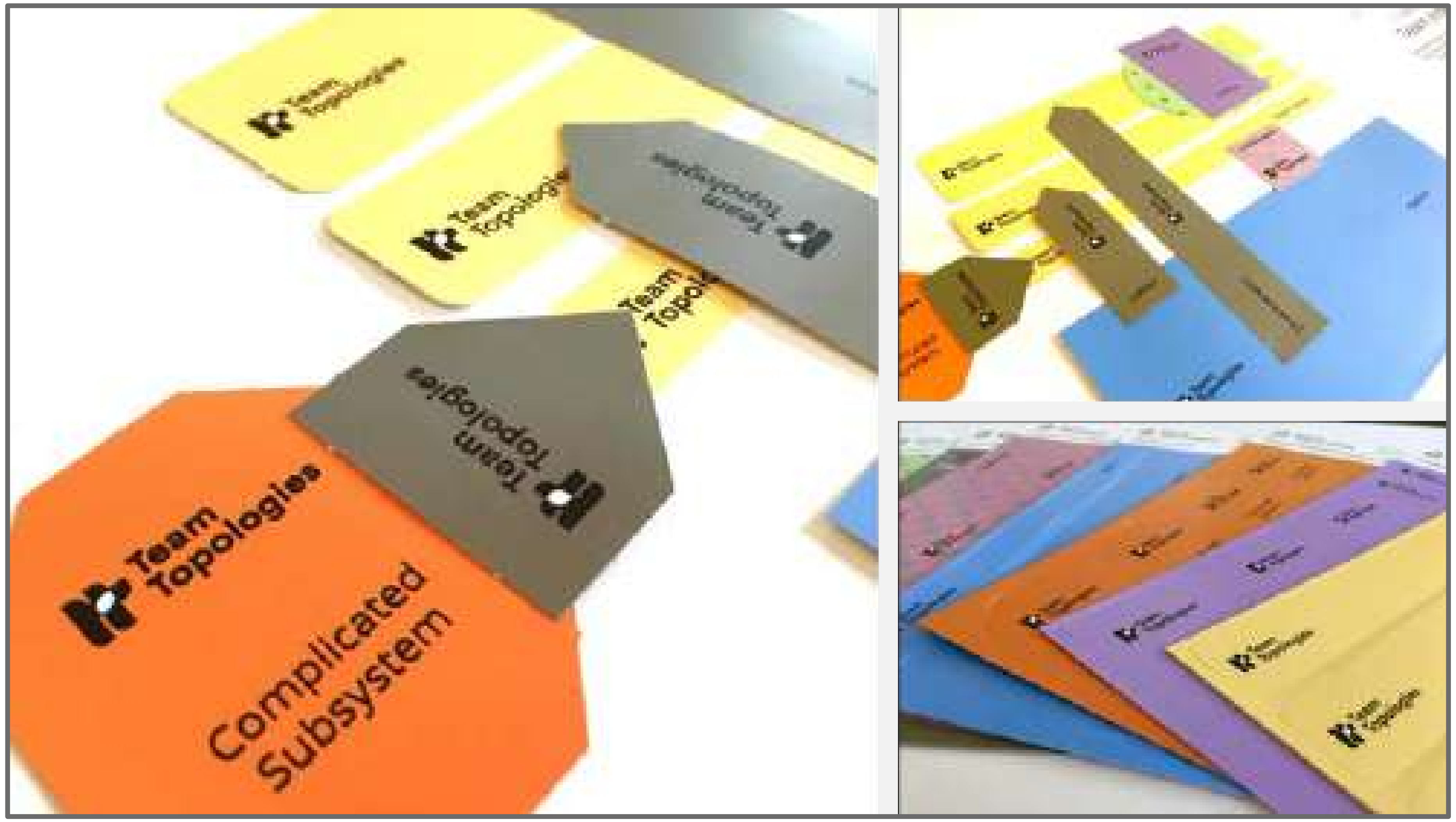
Team interactions:

Flow of change:

Details:

HOVER AN ITEM TO START

Learn more at teamtopologies.com



https://agilestationery.com/products/modeling-shapes-for-team-types-and-team-interactions

Team API - overview

TeamTopologies/Team-API-template

A template for defining a Team API - as explained in the Team Topologies book

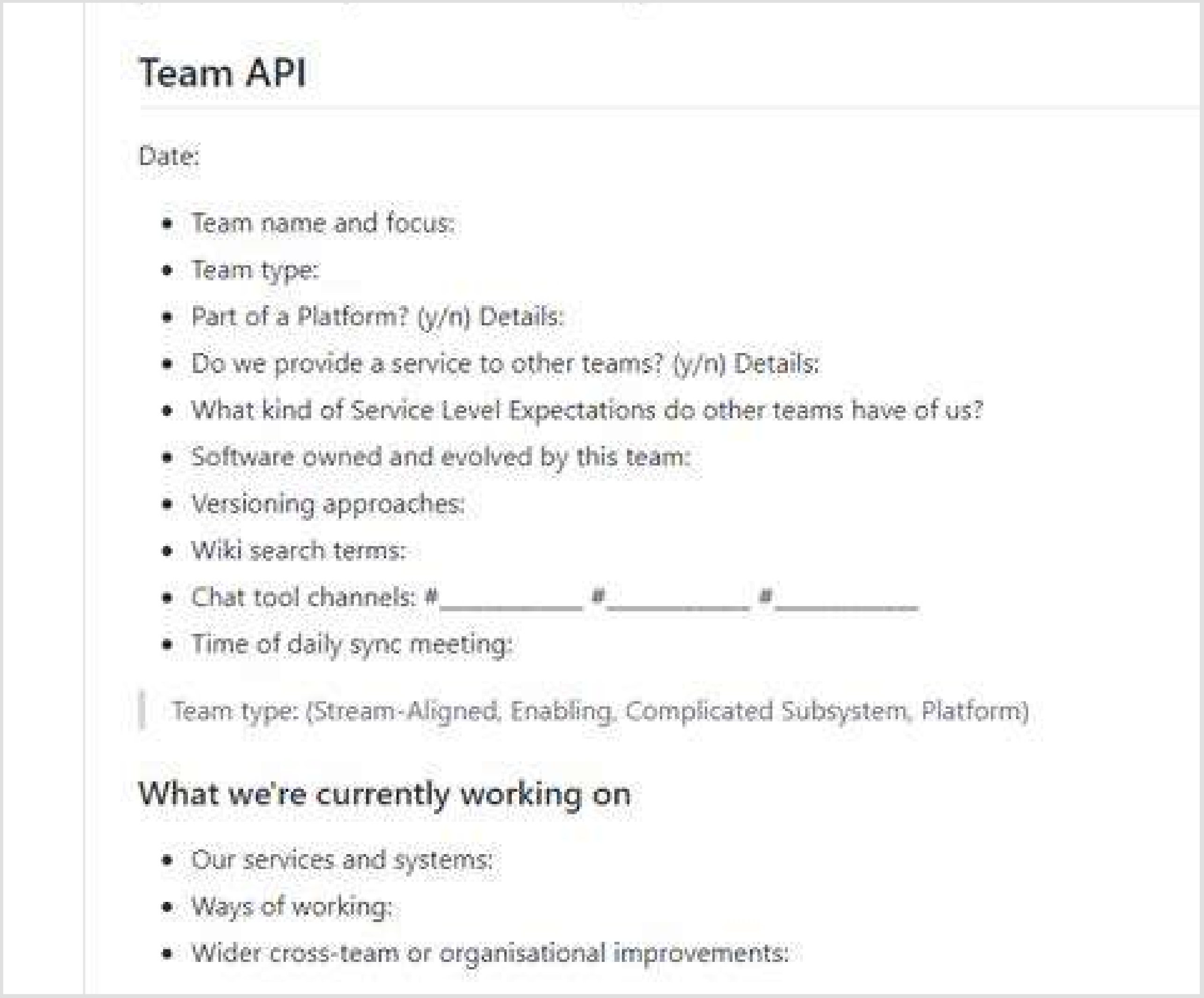
4 3 444 65

github.com

GitHub - TeamTopologies/Team-API-template: A template for defining a Team API - as explained in the Team Topologies book

A template for defining a Team API - as explained in the Team Topologies book - GitHub - TeamTopologies/Team-API-template: A template for defining a Team API - as explained in the Team Topologies book

https://github.com/TeamTopologies/Team-API-template



Team API

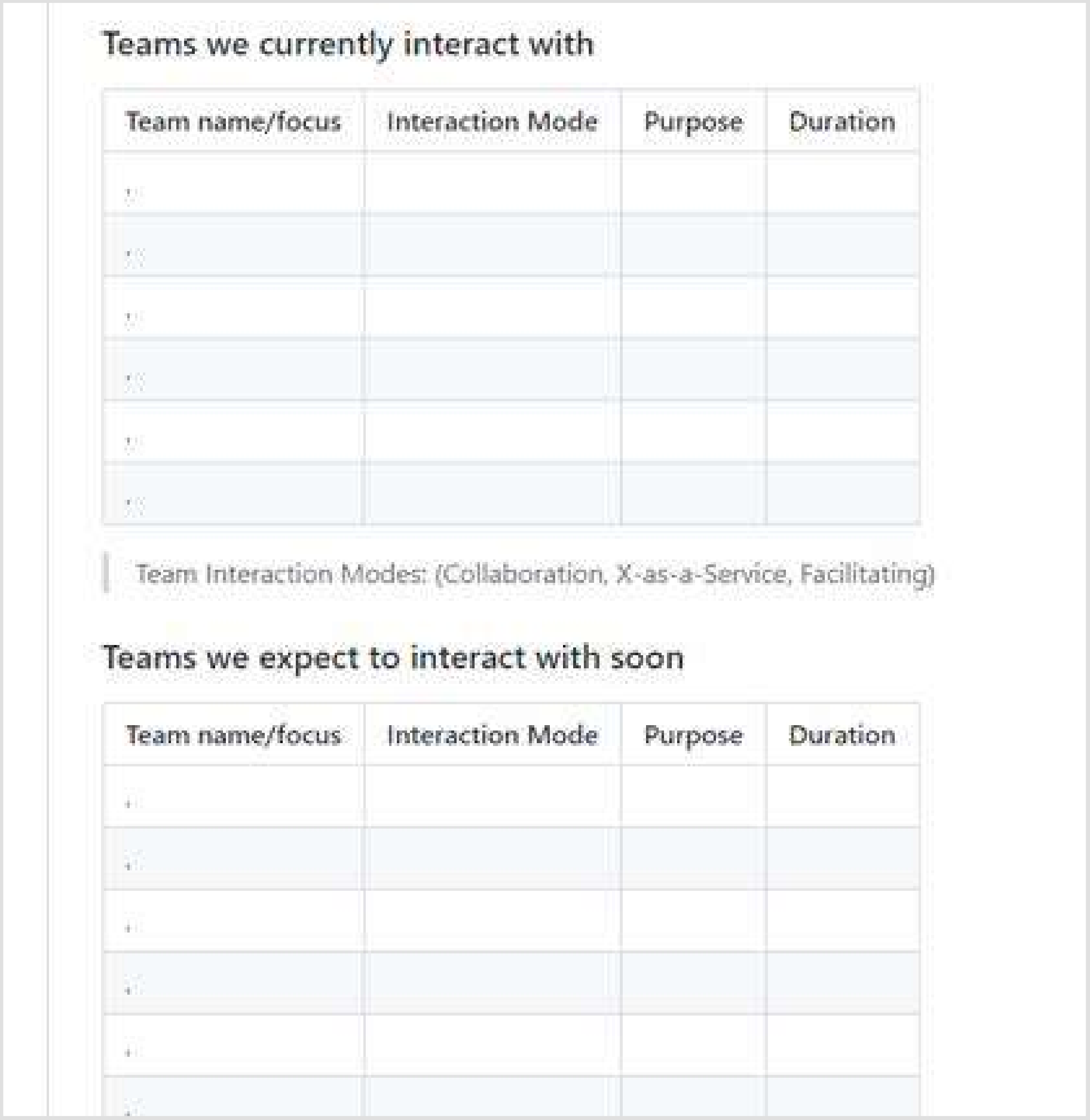
Date:

- Team name and focus:
- Team type:
- Part of a Platform? (y/n) Details:
- Do we provide a service to other teams? (y/n) Details:
- What kind of Service Level Expectations do other teams have of us?
- Software owned and evolved by this team:
- Versioning approaches:
- Wiki search terms:
- Chat tool channels: #__________ #__________ #__________
- Time of daily sync meeting:

> Team type: (Stream-Aligned, Enabling, Complicated Subsystem, Platform)

What we're currently working on

- Our services and systems:
- Ways of working:
- Wider cross-team or organisational improvements:



Teams we currently interact with

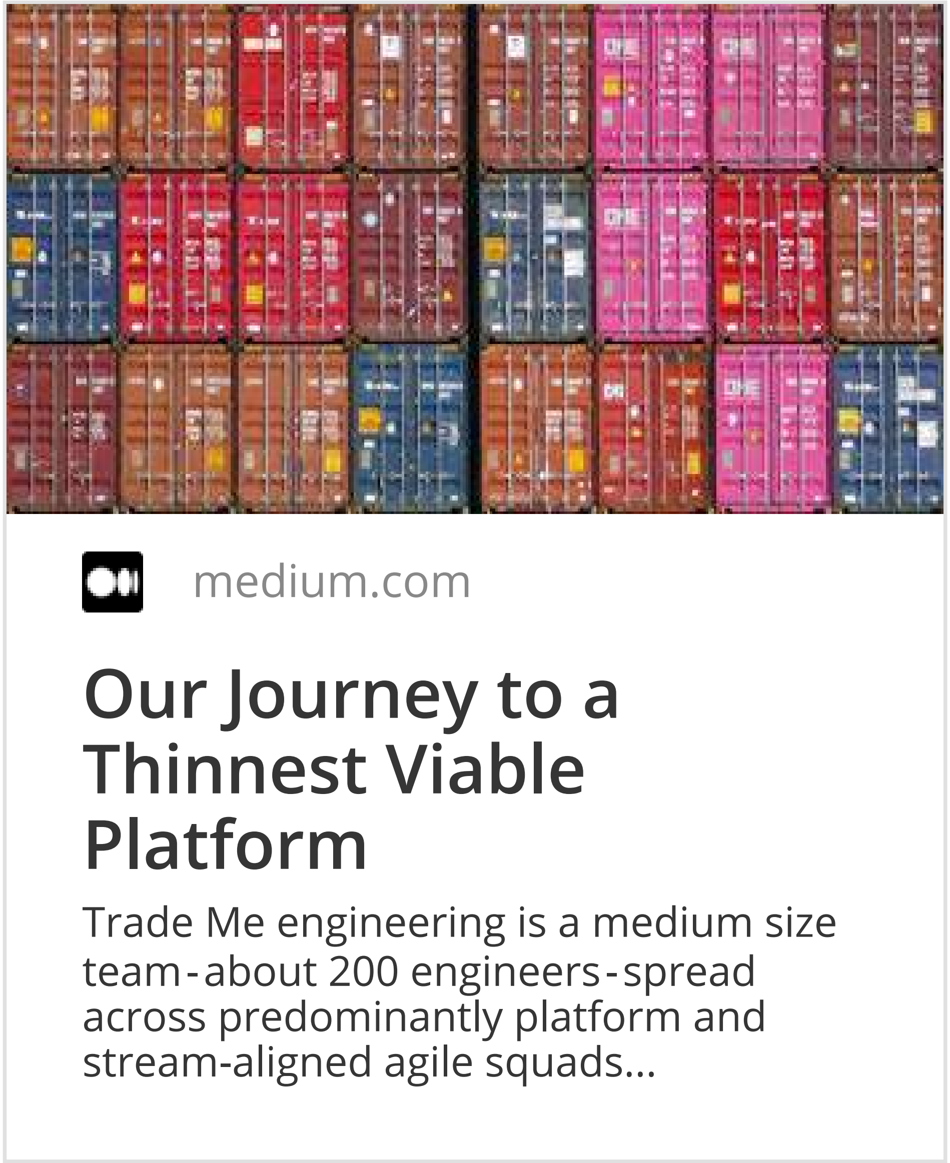
Team name/focus	Interaction Mode	Purpose	Duration

> Team Interaction Modes: (Collaboration, X-as-a-Service, Facilitating)

Teams we expect to interact with soon

Team name/focus	Interaction Mode	Purpose	Duration

TVP example at Trade Me



medium.com

Our Journey to a Thinnest Viable Platform

Trade Me engineering is a medium size team - about 200 engineers - spread across predominantly platform and stream-aligned agile squads...

https://medium.com/trade-me/our-journey-to-a-thinnest-viable-platform-ca3e57986eb9

"It started with a series of wiki pages highlighting the characteristics of a production-ready application and the definitive list of must-haves we expect applications to have to fulfil our stream-aligned teams' needs. We used user story-mapping to identify the Musts.

Subsequently, it evolved into a templated infrastructure-as-code project with almost fully automated provisioning pipelines."

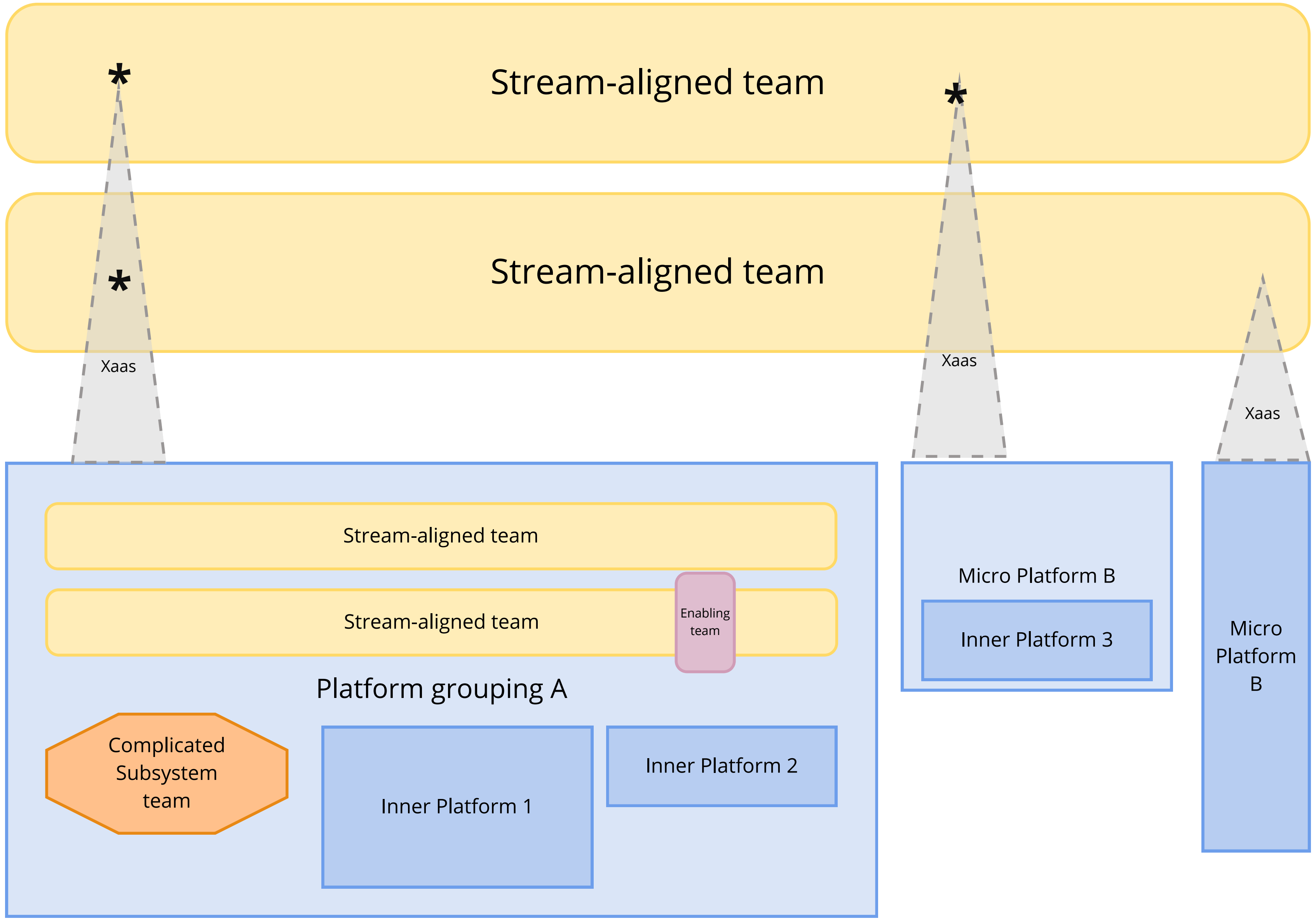
"Our main measures of success (MoS) are:

- Reducing developers' cognitive load (qualitative MoS)
- Time to First Hello World (TTFHW)"

"The intention behind this is to keep the platform as simple as possible to cater to one of its primary purposes: reducing developer cognitive load."

trademe

Fractal platforms - platform groupings 4



Advanced patterns for platforms

Parallel services

Harvesting

Horizon scanning

Composite

Desynchronous

Avoid temporal dependencies

Real-world Software-Enriched Services

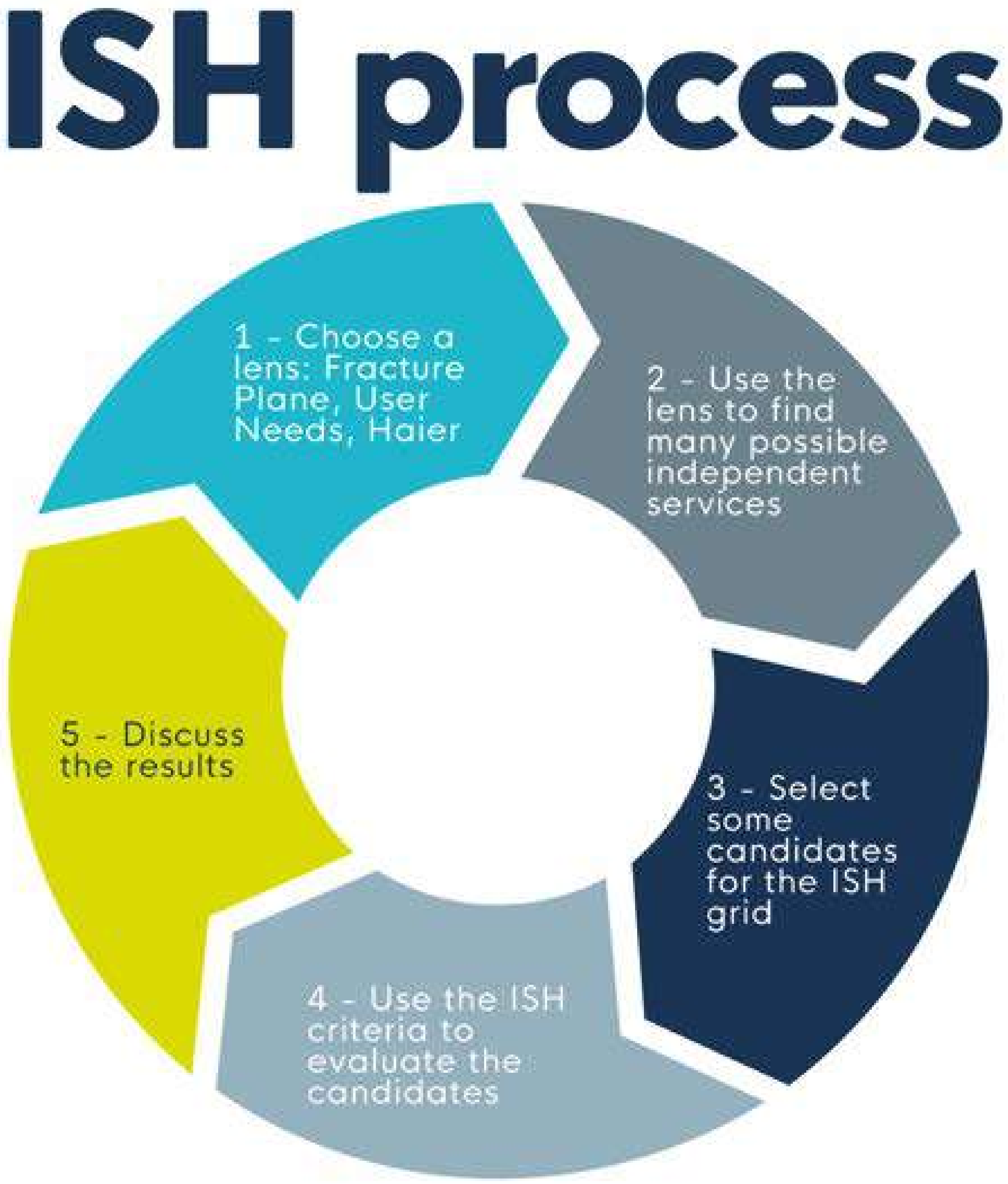
1. Clothes laundering https://laundrapp.com/en/ - simplifies the tasks
2. Buy a car https://www.cinch.co.uk/ - delivered to your door and 14-day money back
3. Run a business https://www.odoo.com/ - an opinionated "platform" of tools
4. Try on clothes digitally https://www.zyler.com/ - saves time in stores
5. ID verification https://www.id-pal.com/ - uses features of a mobile app
6. Transcription https://rev.com/ - automated or human, but all 'As a Service'
7. Vaccination notification and booking: https://www.nhs.uk/nhs-app/ - location specific
8. Website monitoring https://uptimerobot.com/ - no installed software - just a focus on what is public
9. International money transfers https://wise.com/ - no actual bank accounts, but much of the hassle removed
10. Try on glasses virtually https://luna.io/virtual-try-on/ - no need to visit a store

Convert Audio & Video To Text
Trusted by 170,000+ customers across all industries

The fx cheaper international account.
Available for you in United Kingdom

ISH process

1. Choose a lens: Fracture Plane, User Needs, Haier
2. Use the lens to find many possible independent services
3. Select some candidates for the ISH grid
4. Use the ISH criteria to evaluate the candidates
5. Discuss the results


ISH process
1 - Choose a lens: Fracture Plane, User Needs, Haier
2 - Use the lens to find many possible independent services
3 - Select some candidates for the ISH grid
4 - Use the ISH criteria to evaluate the candidates
5 - Discuss the results

Flows of change

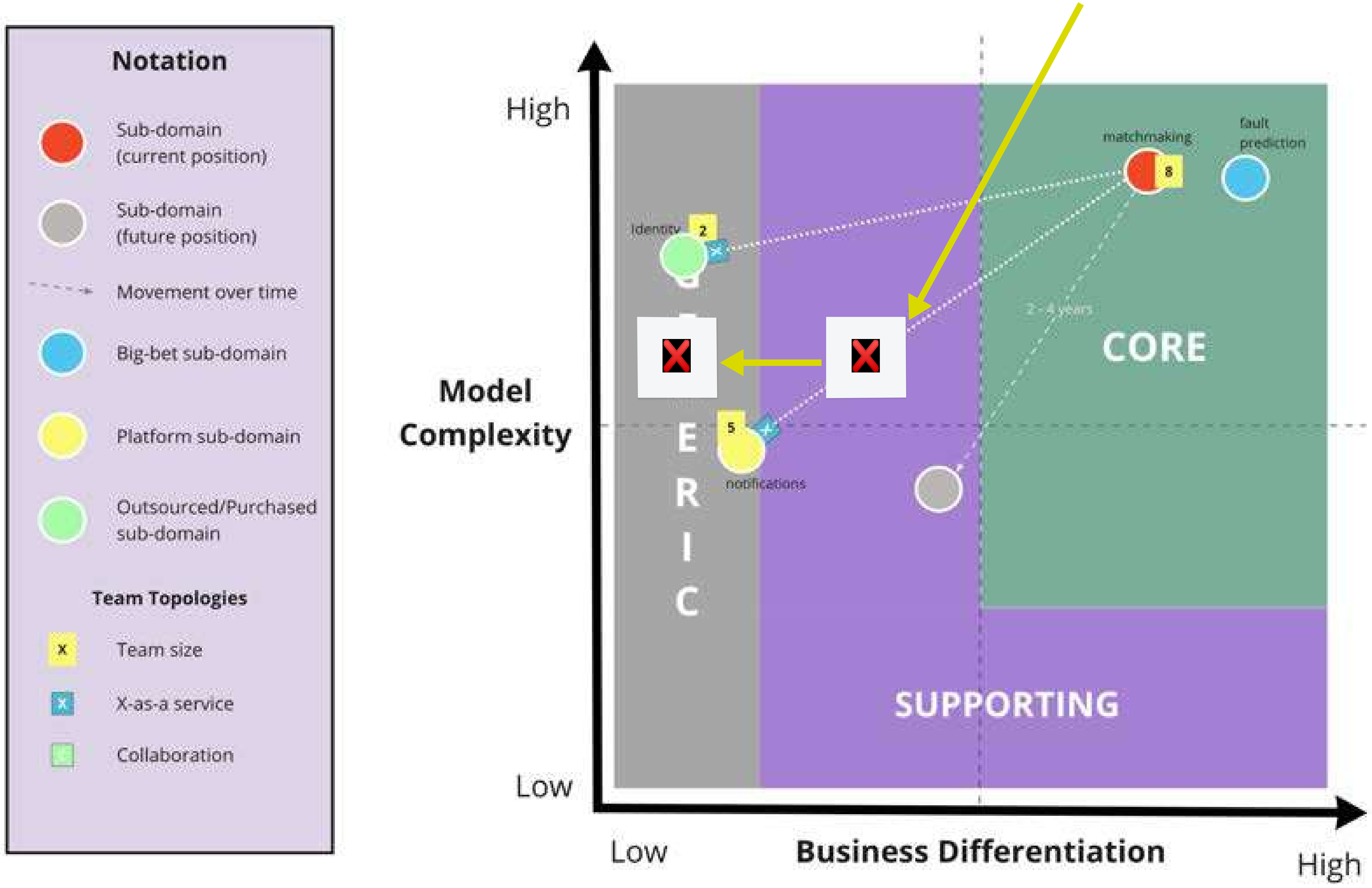
Almost ALL roles and teams should be focused on either:

A flow of change 🔀

Supporting flows of change 🤲

Core Domain Charts - skills adaptability

What should a smart organization do if it recognizes skills here ❌ moving to Generic?



Core Domain Charts - skills adaptability

Extreme Modelling Patterns - Alberto Brandolini - DDD Eur...
YouTube

https://www.youtube.com/watch?v=jv1-ohCWbE0

Example of Composite pattern

Human centred system design - Trond Hjorteland
YouTube

https://www.youtube.com/watch?v=KSWyIZevgHc

Open systems and related to environment