

Combining Team Topologies with Context Maps

INNOQ



MICHAEL PLÖD

FELLOW



259



433



78



72



25

Michael Plöd

Fellow at INNOQ

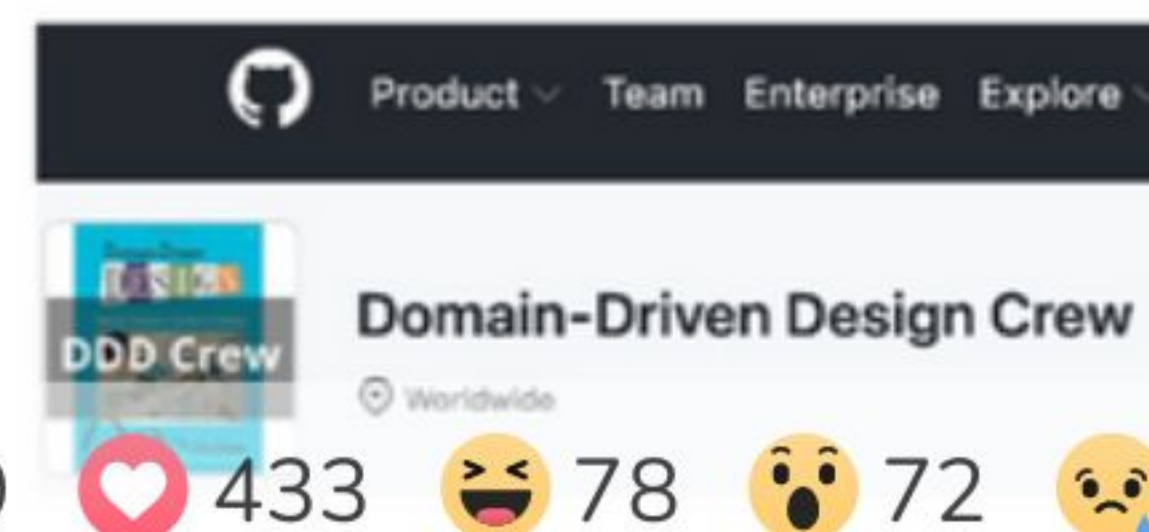
Mastodon (or Twitter): @bitboss@mastodon.social

LinkedIn: <https://www.linkedin.com/in/michael-ploed/>

Current consulting topics:

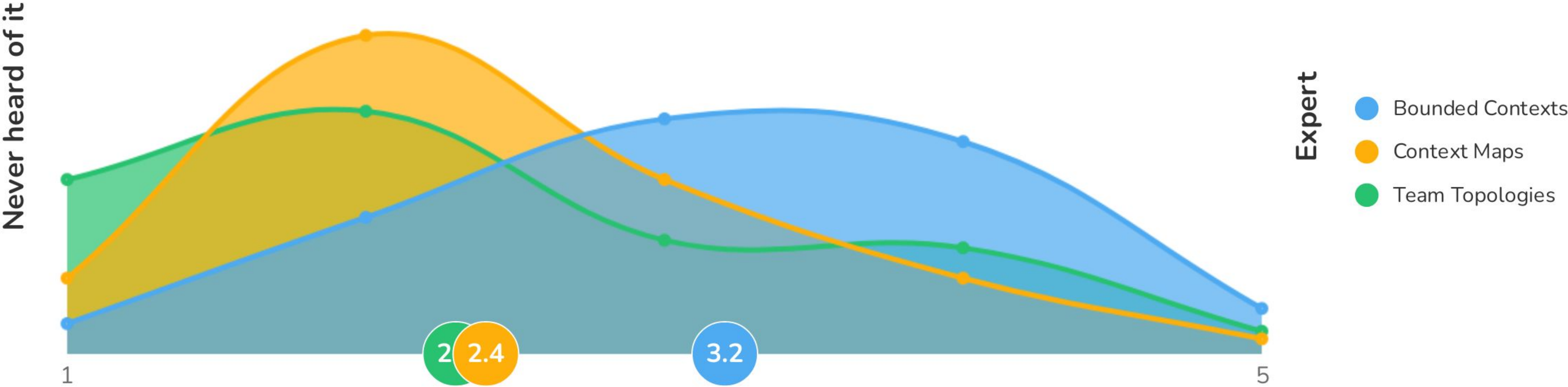
- Domain-Driven Design
- Team Topologies
- Transformation from IT Delivery to digital product orgs

Regular speaker at (inter-)national conferences and author of a book + various articles

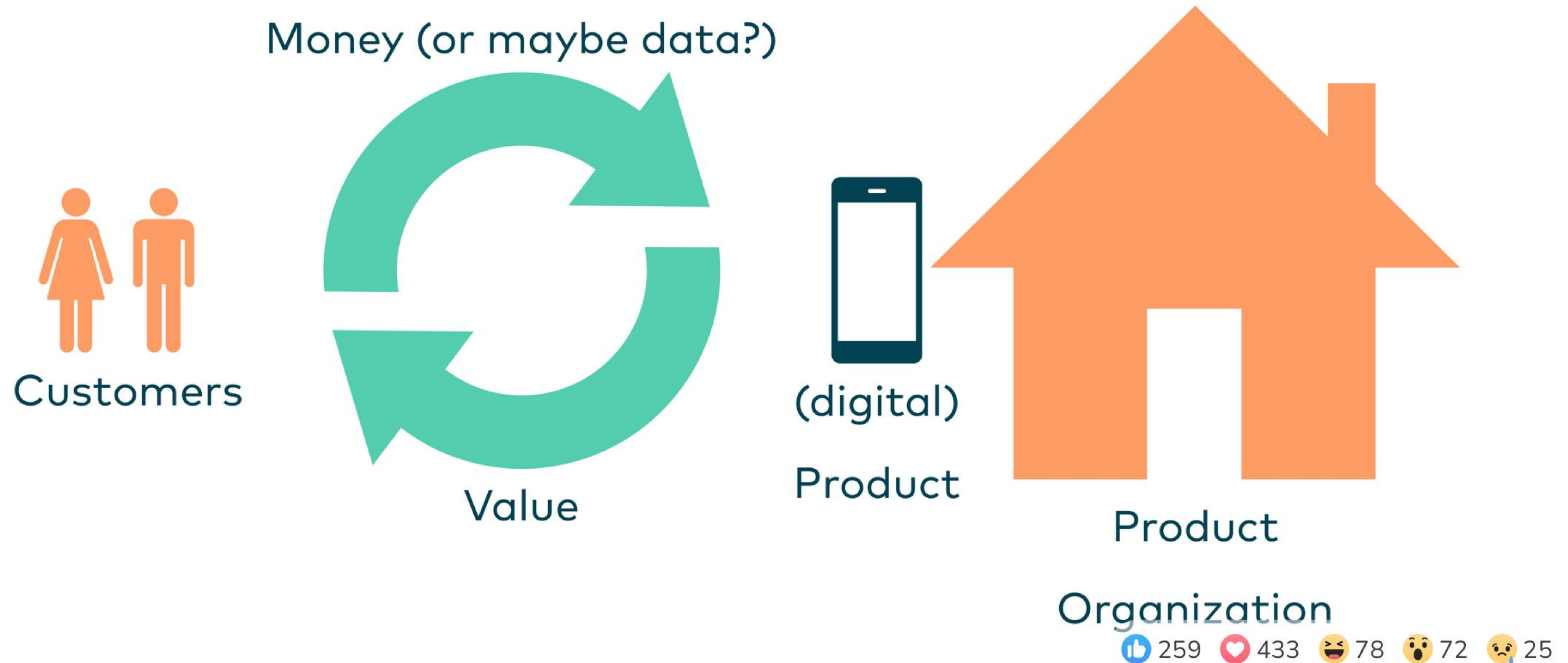


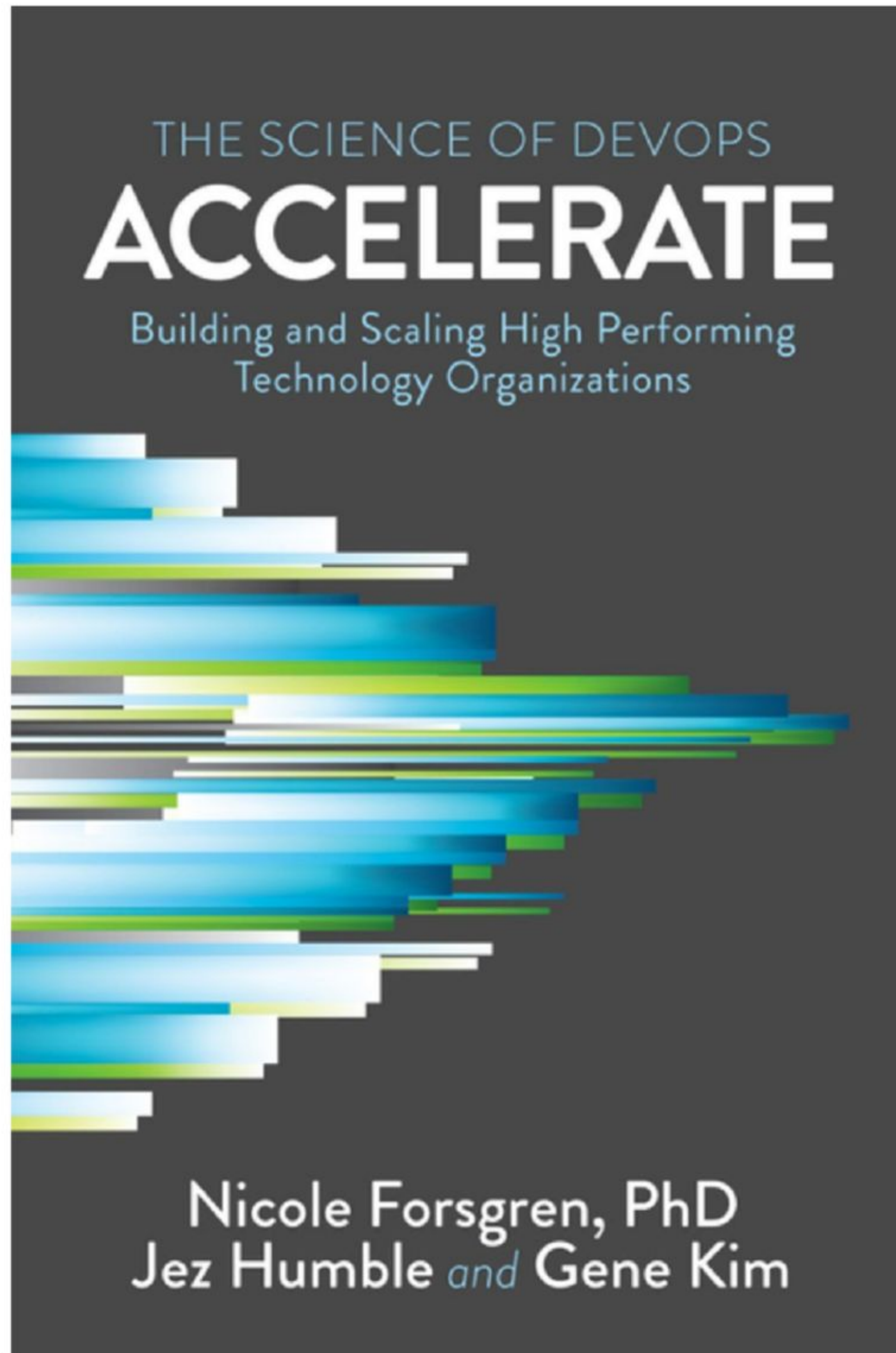
259 433 78 72 25

Rate your knowledge



How can we maximize the value exchange with the customer in a continuous fashion at high velocity?



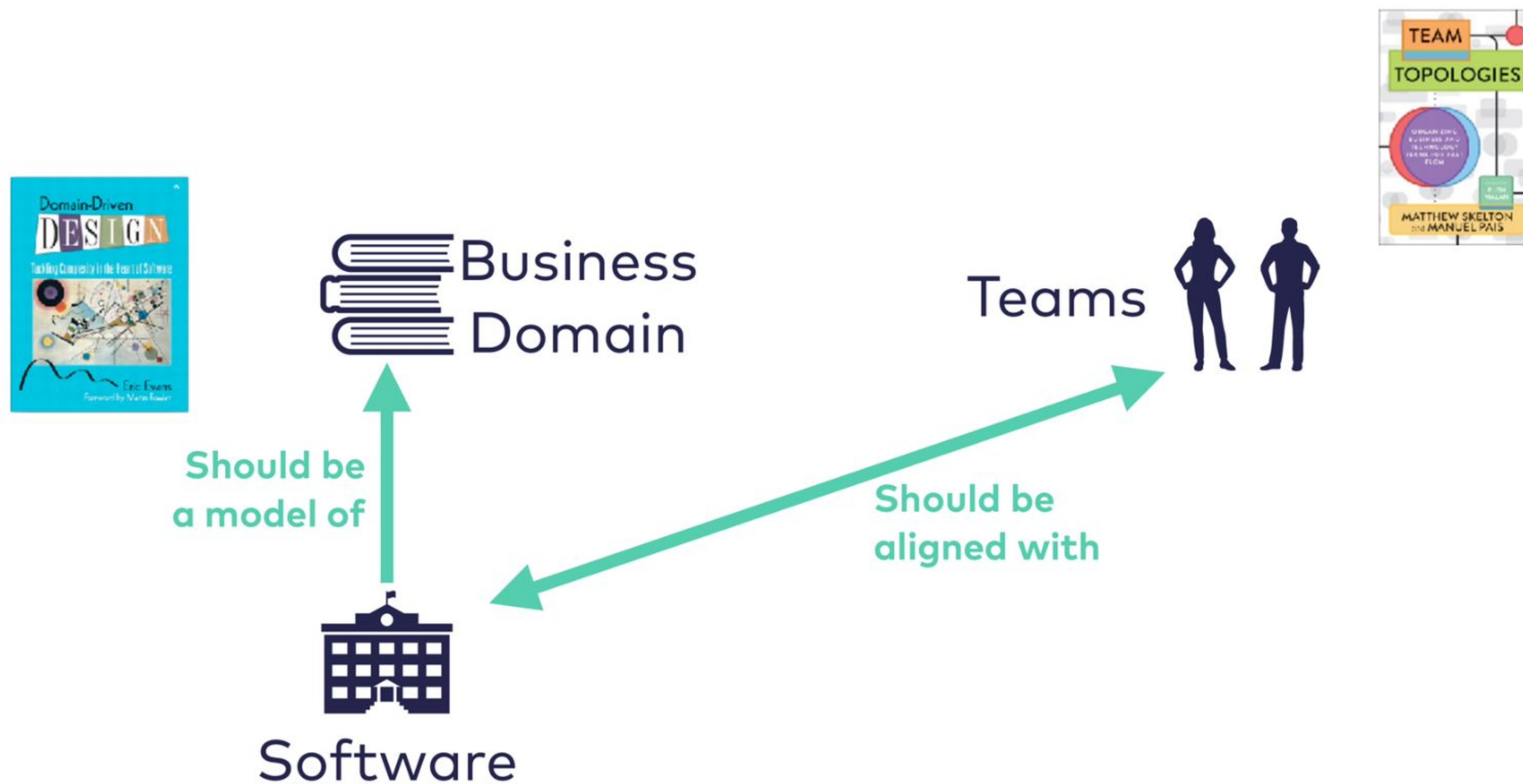


"A loosely coupled software architecture and org structure to match" is a key predictor of:

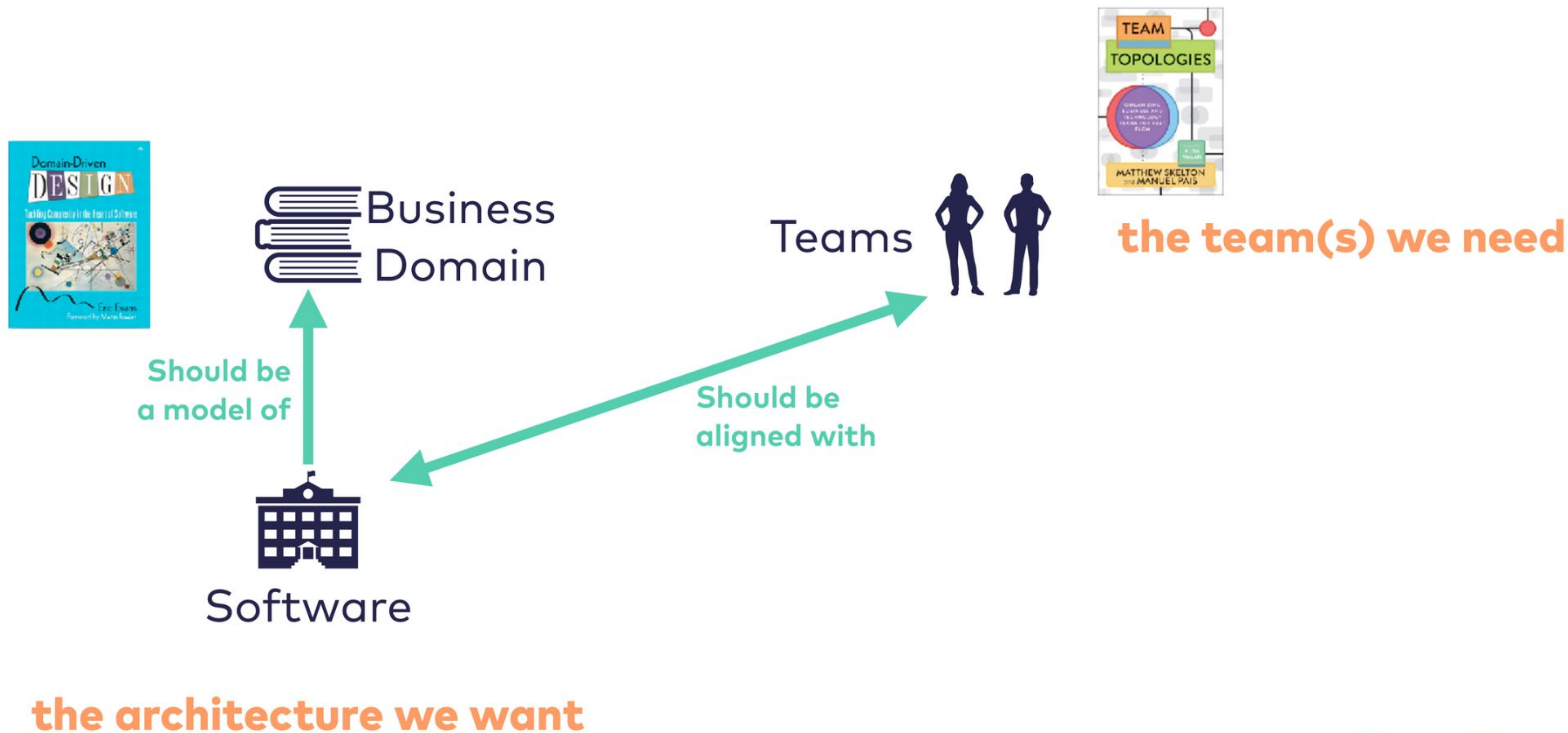
- Continuous Delivery Performance
- Ability to scale organization and increase performance linearly



Some basic ideas...



Some basic ideas...



"Sociotechnical Architecture is about taking an holistic co-design approach to technical and organizational systems, given the inherent impact they have on each other."

Eduardo Da Silva

<https://esilva.net>



 259  433  78  72  25

**„Team assignments
are the first draft
of the
architecture“**



Michael Nygard

Author of „Release It“

**There are two boundaries to this
and we should align them**

**Team
Boundaries**

**Software
Boundaries**

„Good fences make good neighbors“

Robert Frost

 259  433  78  72  25

NEW YORK TIMES BESTSELLER

"Provocative and fascinating." —MALCOLM GLADWELL

Daniel H. Pink

author of *A Whole New Mind*

DRIVE

The Surprising Truth
About What Motivates Us

Autonomy

Mastery

Purpose

👍 259 ❤️ 433 😂 78 😱 72 🙄 25



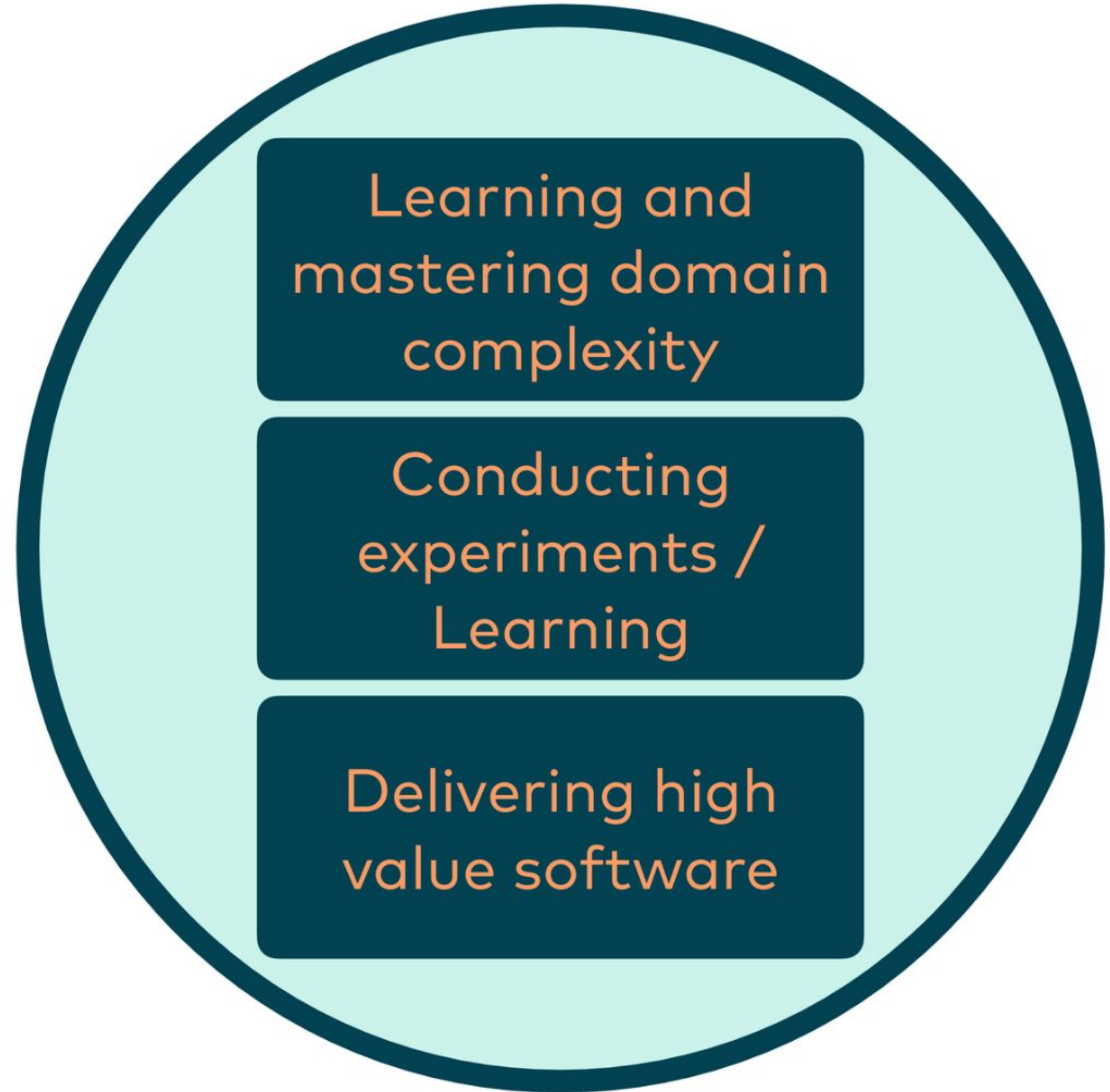
**We need good boundaries in which teams
can achieve**

Autonomy - Mastery - Purpose

A Bounded Context is a boundary for a model expressed in a consistent language tailored around a specific purpose

Bounded Context

What we want
to achieve in a
**Bounded
Context**



NEW YORK TIMES BESTSELLER

"Provocative and fascinating." —MALCOLM GLADWELL

Daniel H. Pink

author of *A Whole New Mind*

DRIVE

The Surprising Truth
About What Motivates Us

Autonomy

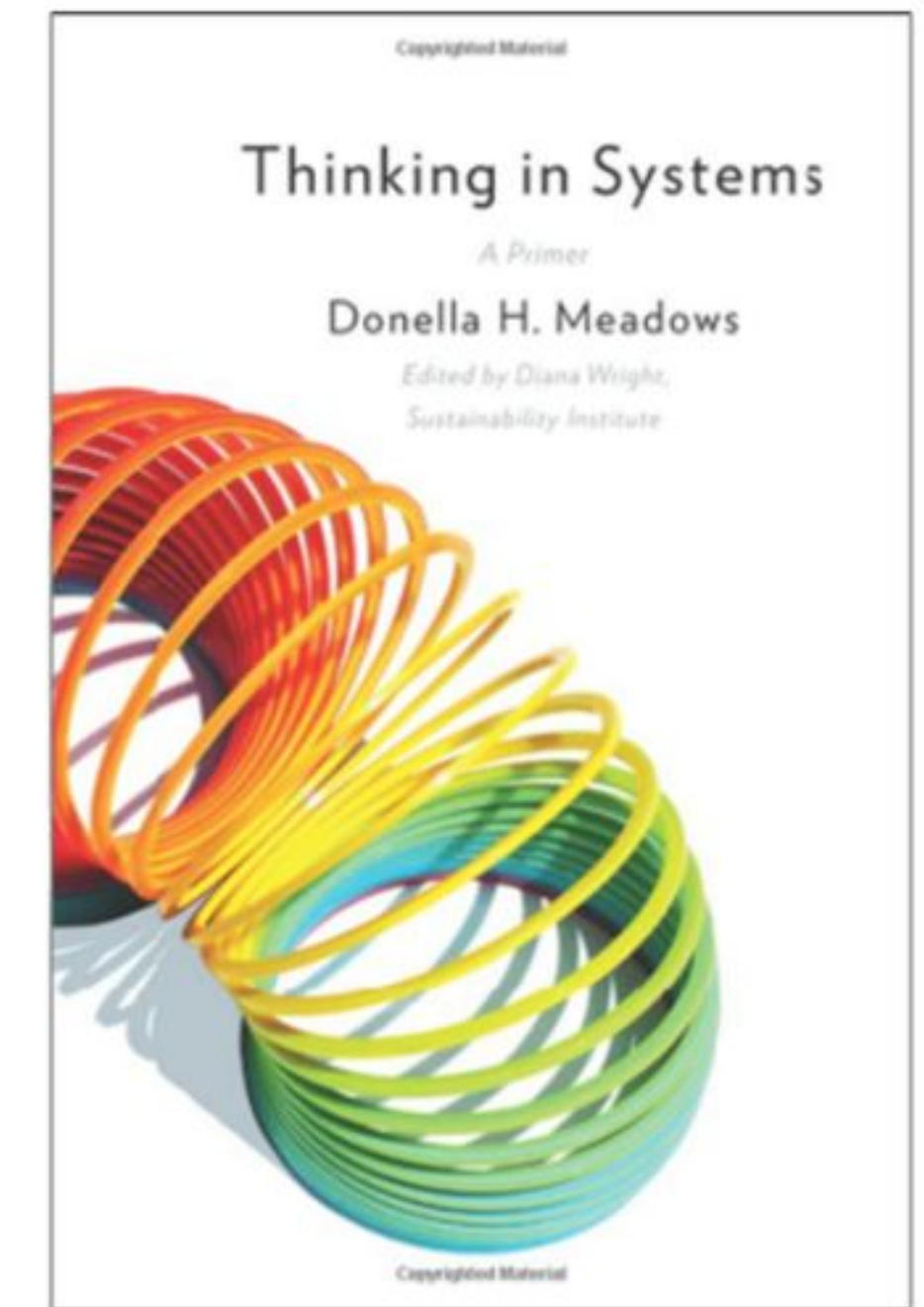
Mastery

Purpose

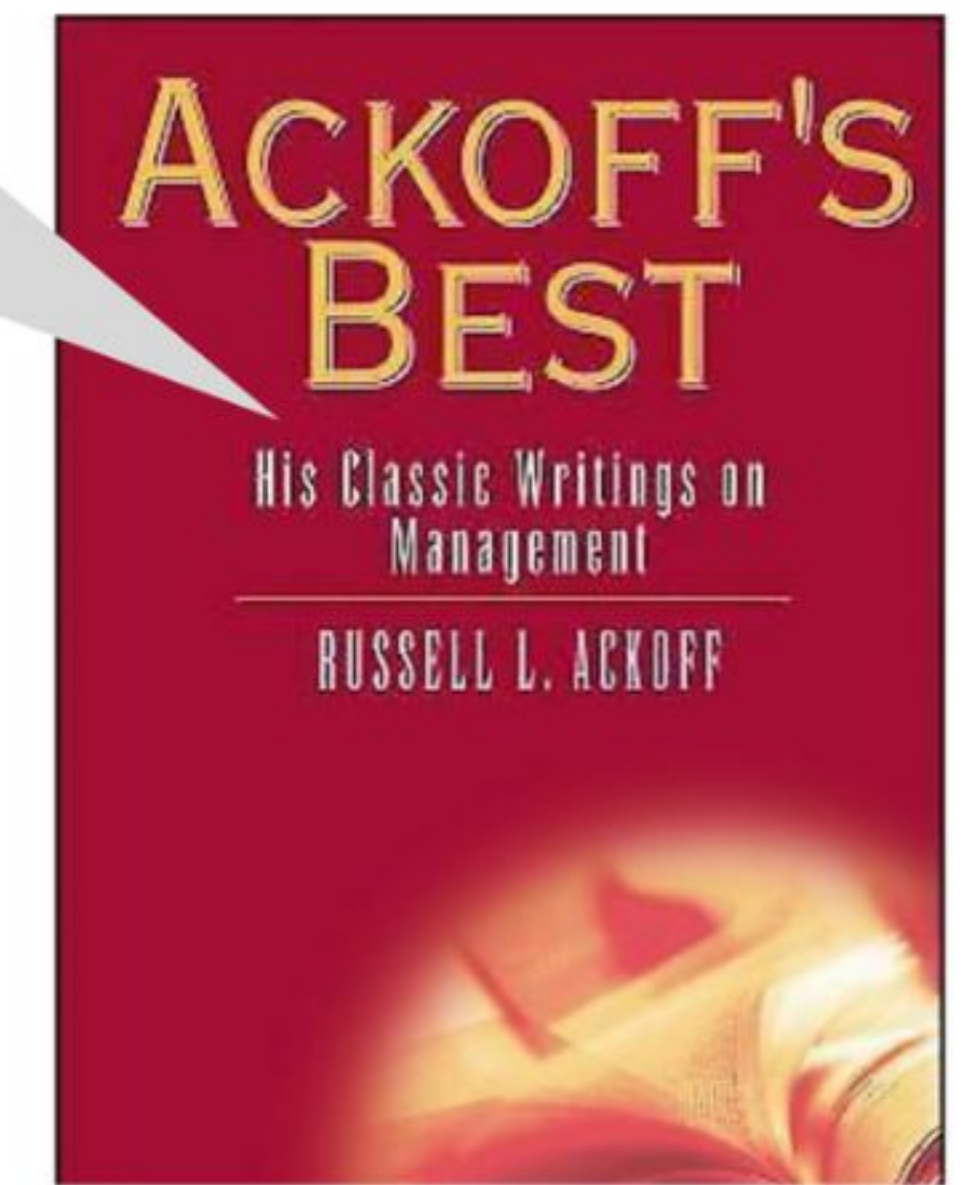
 259  433  78  72  25



Sociotechnical Architectures are a lot about Systems Thinking as well



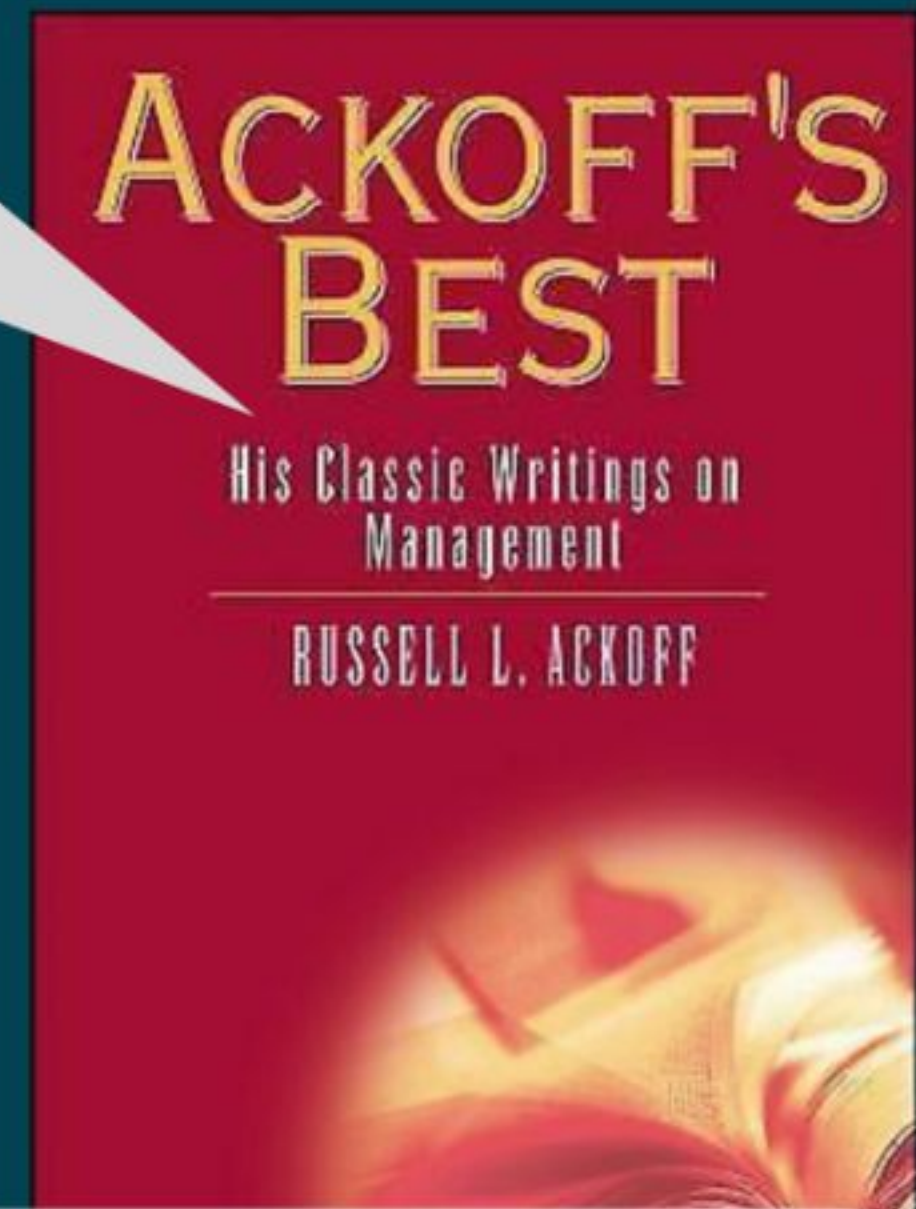
*To manage a
system effectively, you
might focus on the
interactions of the parts
rather than their behavior
taken separately*



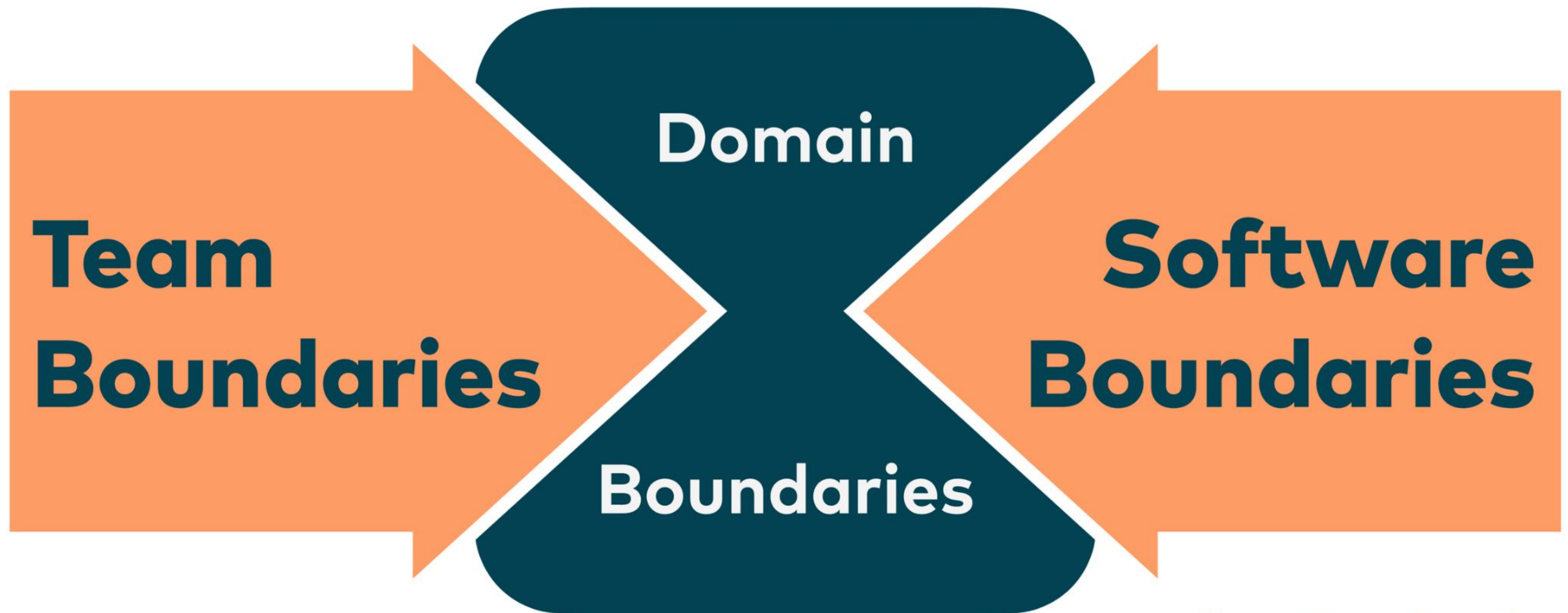
Autonomy

Isn't it about (maybe a reduction / lack of) interactions?

*To manage a system effectively, you might **focus on the interactions of the parts** rather than their behavior taken separately*



Align along domain boundaries



TEAM

TOPOLOGIES

ORGANIZING
BUSINESS AND
TECHNOLOGY
TEAMS FOR FAST
FLOW

Foreword by
RUTH
MALAN

MATTHEW SKELTON
and MANUEL PAIS

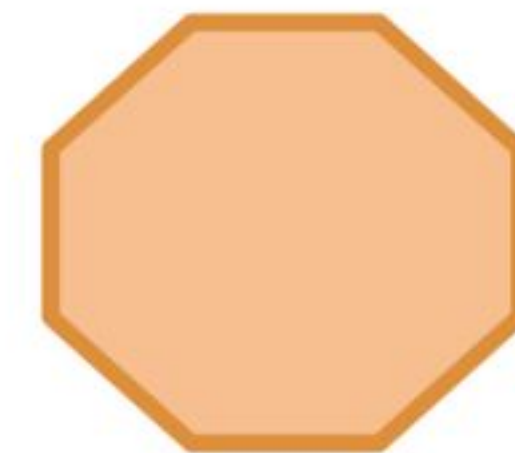
"An architect should be thinking:

Which team interaction modes
are appropriate for these two
teams?

What kind of communication do
we need between these two
parts of the system, between
these two teams?"



Fundamental Team Topologies



Complicated Subsystem



Enabling



Platform



Stream-aligned

TEAM

TOPOLOGIES

ORGANIZING
BUSINESS AND
TECHNOLOGY
TEAMS FOR FAST
FLOW

Foreword by
**RUTH
MALAN**

MATTHEW SKELTON
and **MANUEL PAIS**

Stream-aligned Team



- Tailored to a business area or organizational capability (Bounded Context)
- Is intended to create customer value quickly, safely and autonomously without having to delegate parts of the work to other teams.

Stream-aligned Team



- Tailored to a business area or organizational capability (Bounded Context)
- Is intended to create customer value quickly, safely and autonomously without having to delegate parts of the work to other teams.

Complicated Subsystem Team



- Responsible for building and maintaining a part of the system that is highly dependent on specialist expertise
- Team manages the complexity of the subsystem using specific skills and expertise that are usually difficult to find or recruit.

Stream-aligned Team



- Tailored to a business area or organizational capability (Bounded Context)
- Is intended to create customer value quickly, safely and autonomously without having to delegate parts of the work to other teams.

Complicated Subsystem Team



- Responsible for building and maintaining a part of the system that is highly dependent on specialist expertise
- Team manages the complexity of the subsystem using specific skills and expertise that are usually difficult to find or recruit.

Platform Team



- Should give stream-aligned teams the possibility to do their work with a high degree of autonomy,
- Platform provides self-service APIs, tools and services as an internal product

Stream-aligned Team



- Tailored to a business area or organizational capability (Bounded Context)
- Is intended to create customer value quickly, safely and autonomously without having to delegate parts of the work to other teams.

Complicated Subsystem Team



- Responsible for building and maintaining a part of the system that is highly dependent on specialist expertise
- Team manages the complexity of the subsystem using specific skills and expertise that are usually difficult to find or recruit.

Platform Team



- Should give stream-aligned teams the possibility to do their work with a high degree of autonomy,
- Platform provides self-service APIs, tools and services as an internal product

Enabling Team



- Work alongside the stream-aligned teams and support them in the area of knowledge building and empowerment.
- Have a strong collaborative nature and strive to understand the problems and shortcomings of the other teams
- Inhouse consulting team

Team Interaction Modes



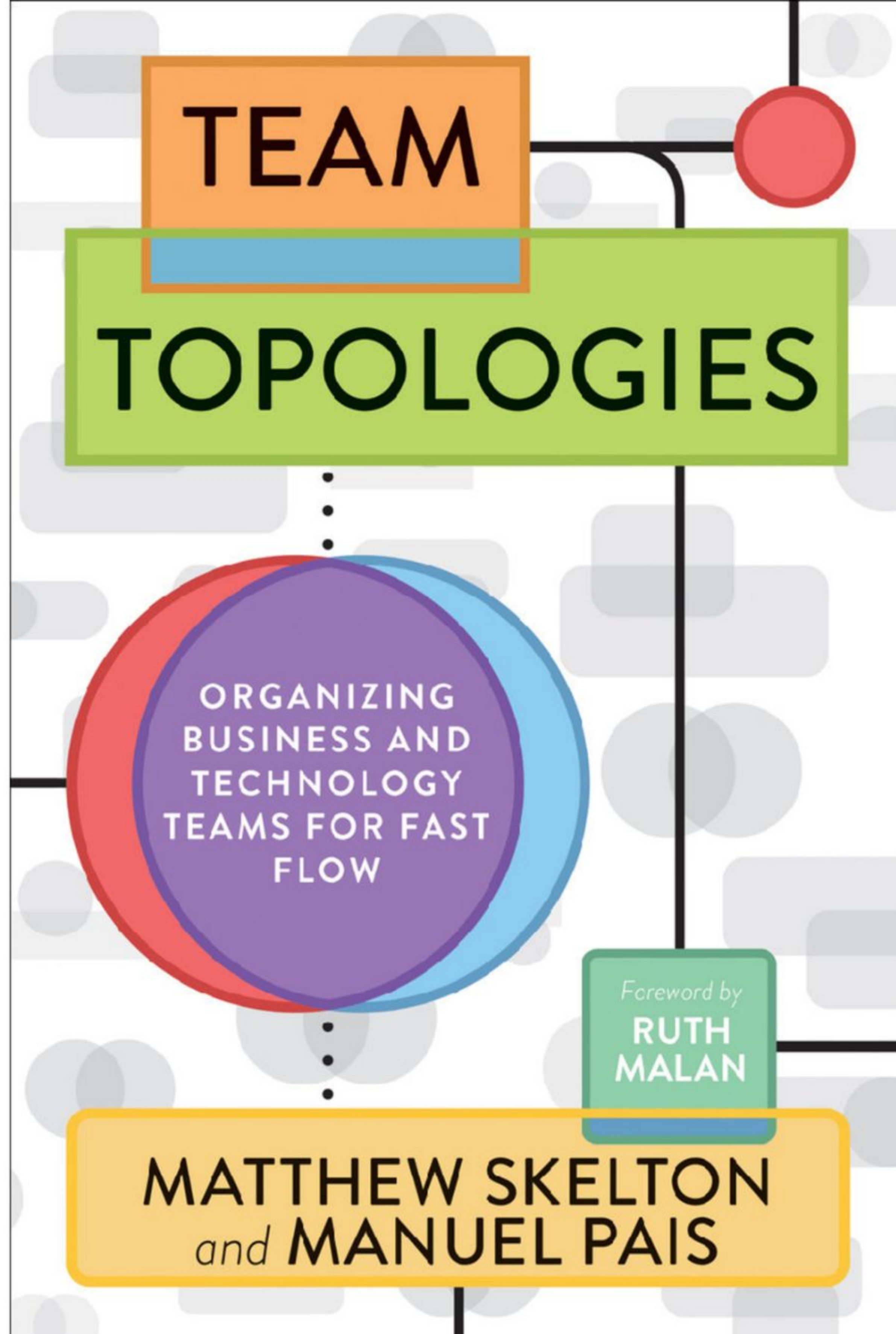
Collaboration



X-as-a-Service



Facilitating



259



433



78



72



25

Team Interaction Modes

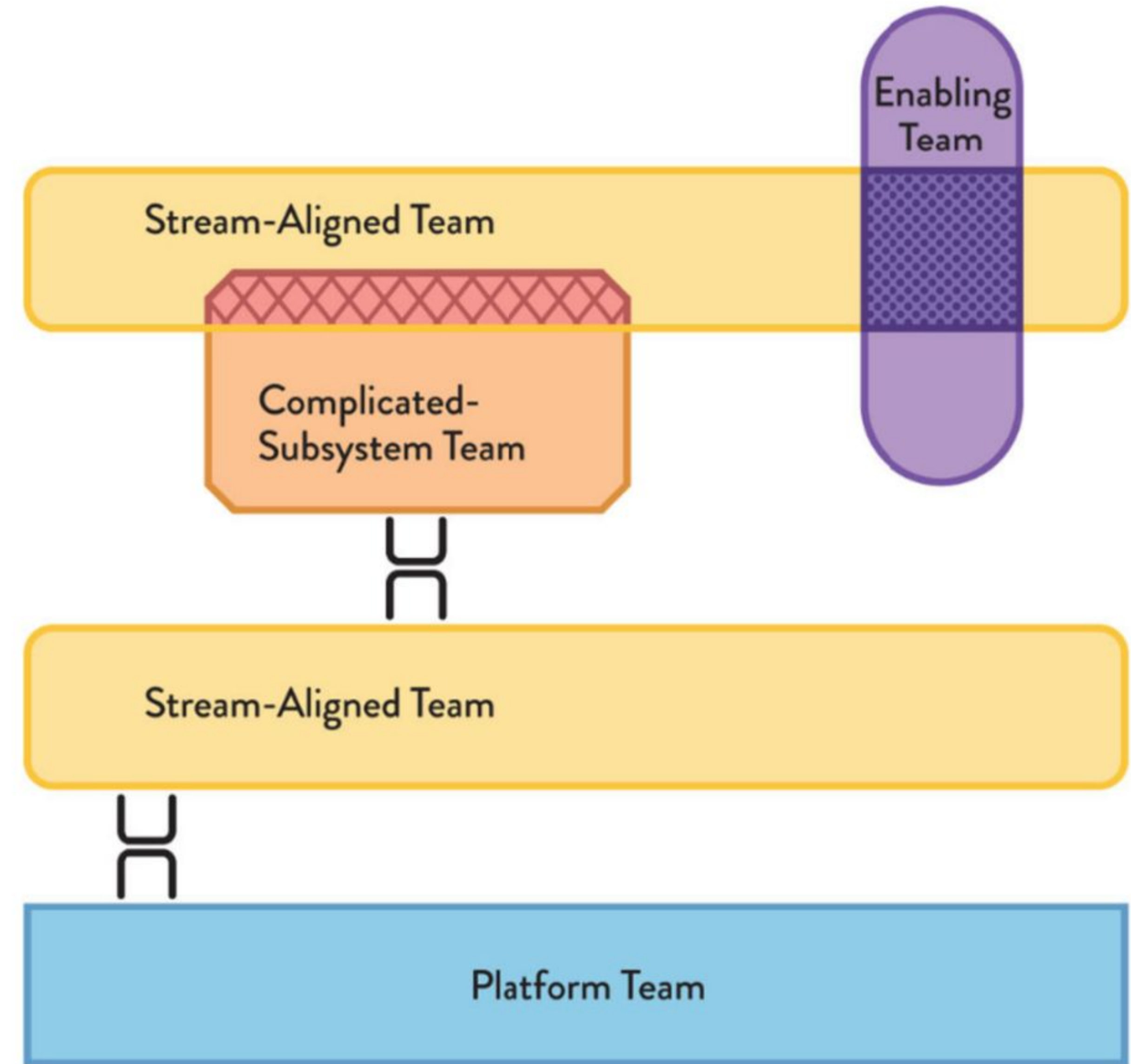
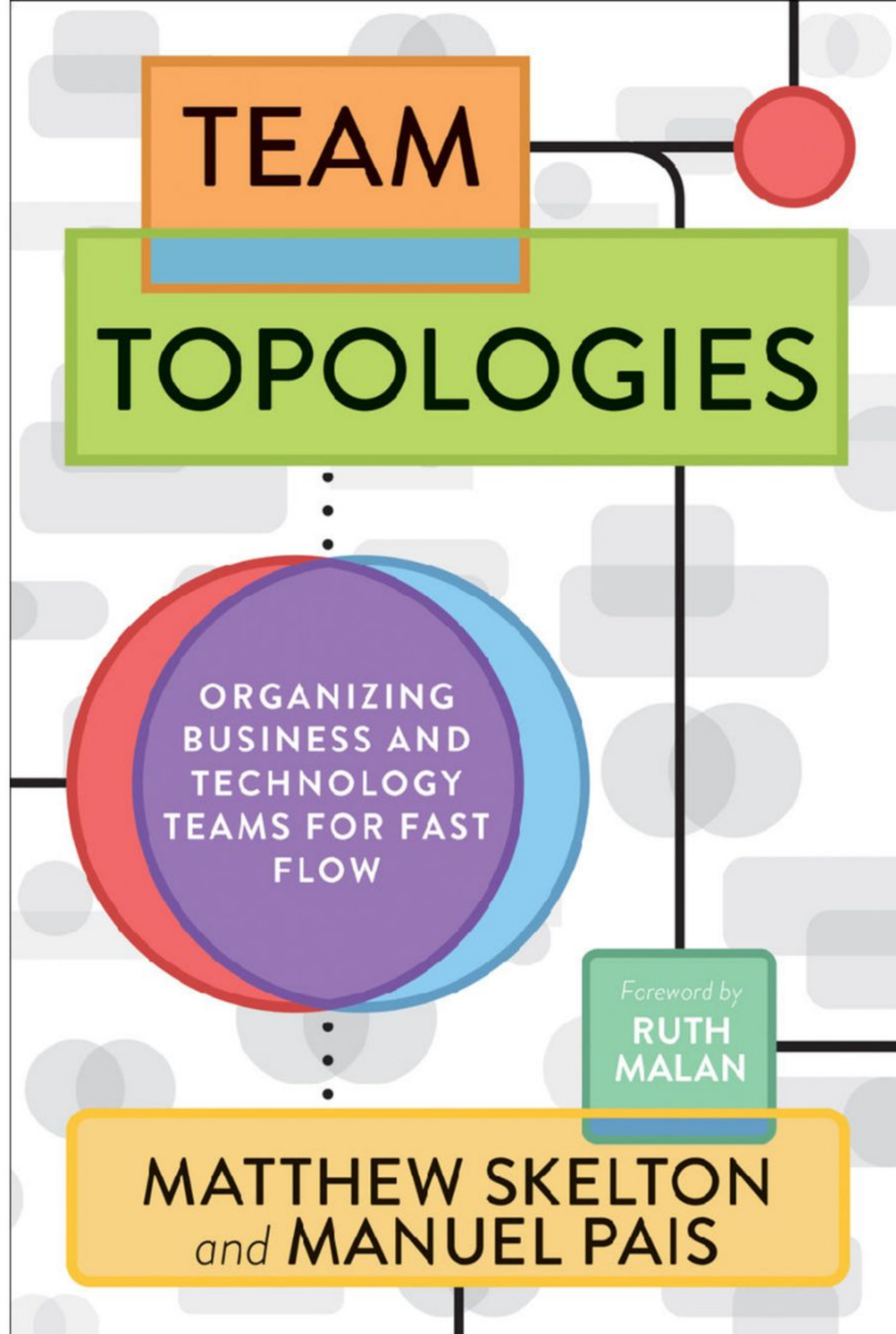
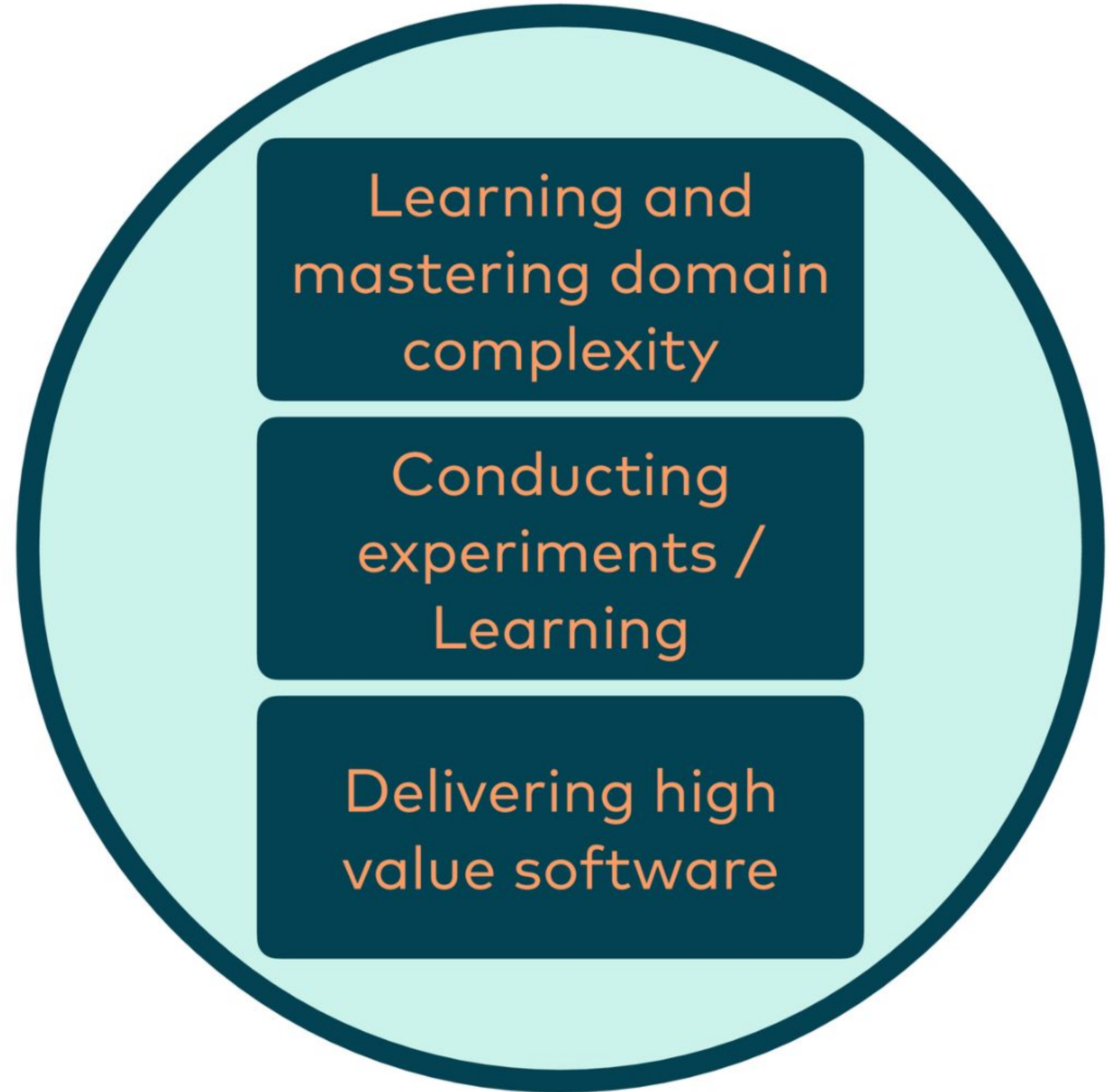


Image taken from the [Team Topologies](#) book

Mind the **COGNITIVE LOAD**

of the teams.
We need a
boundary for
this!



TEAM

TOPOLOGIES

ORGANIZING
BUSINESS AND
TECHNOLOGY
TEAMS FOR FAST
FLOW

Foreword by
RUTH
MALAN

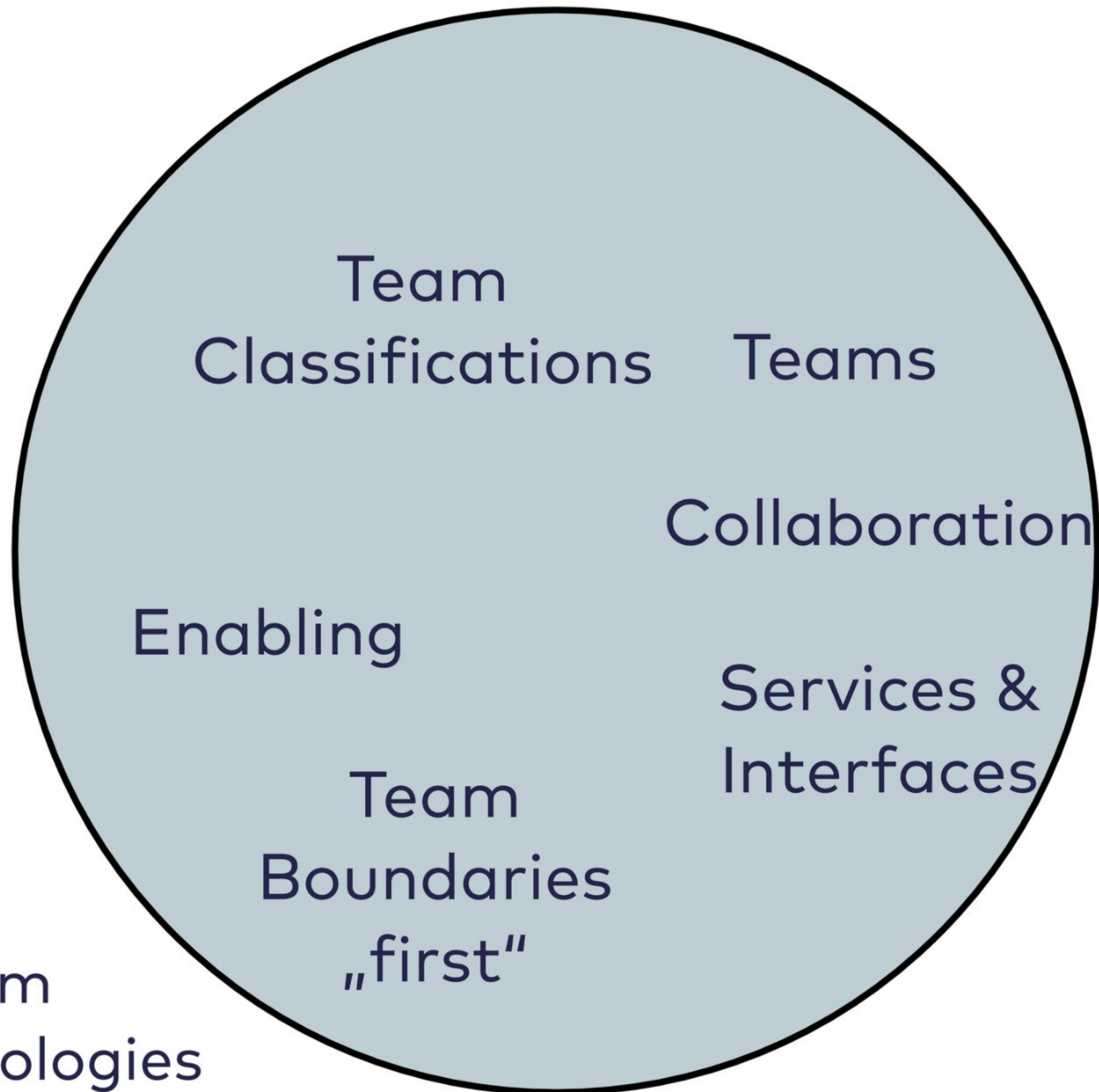
MATTHEW SKELTON
and MANUEL PAIS

The
Bounded Context
(as a fracture plan)
is a

team first
boundary



Team
Topologies

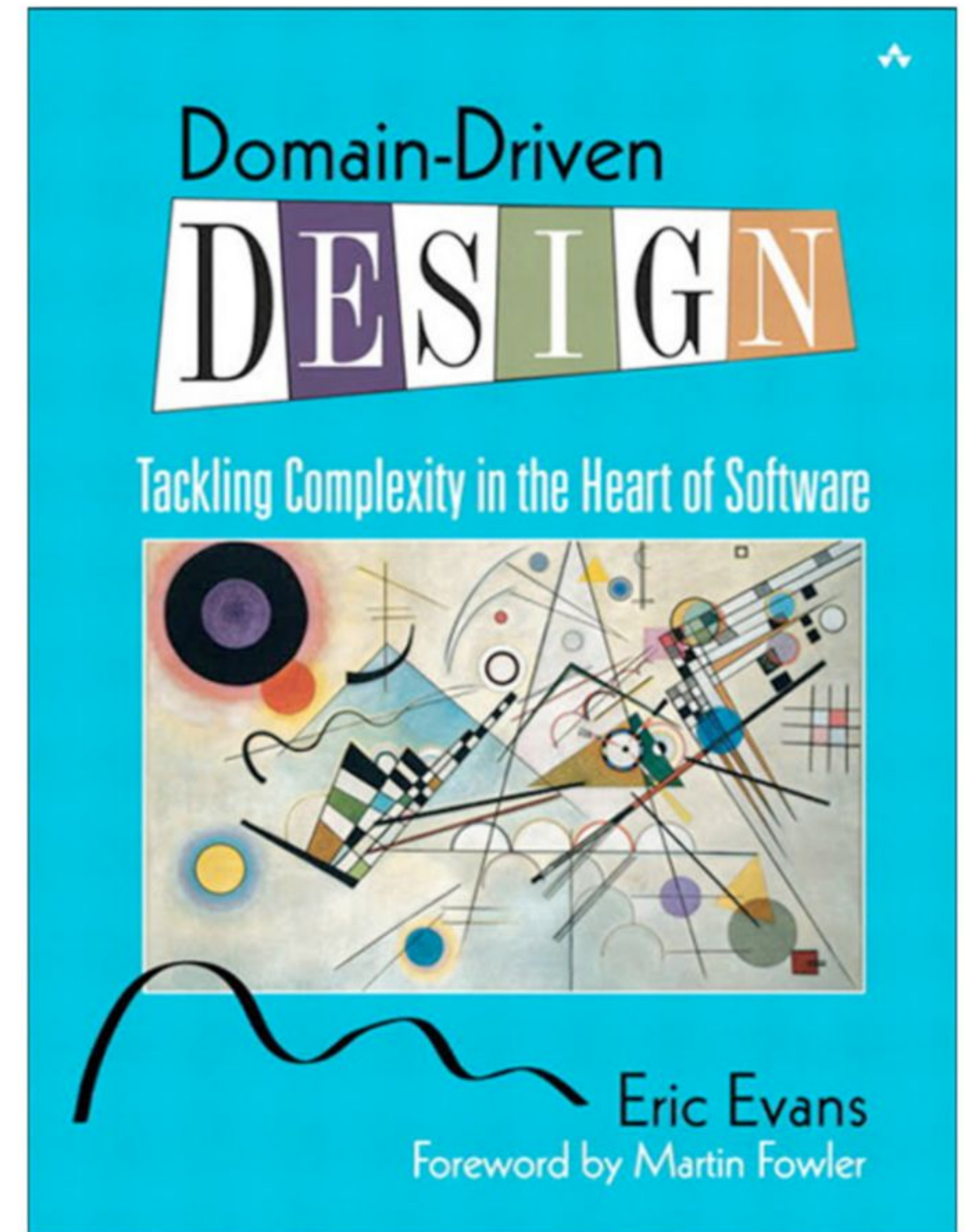


Where do you see limits of Team Topologies in terms of having a holistic overview?

No tech info, no business info	When org is very large (we have 190 teams)	People might just copy it, without thinking through their own situation/challenges
Cross team alignment in Lager organisations	Support-Teams	Informal exchanges afe important for creative processes.
It does not conclude anything on the business domain	Team topologies only tough how teams collaborate towards a greater goal (-> the streams)....	Different culture and thinking
Difficult to apply on bin and lang grown organizations	May not include all scenarios or patterns	QA teams
Duplications across teams might make it difficult to implement cross-cutting concerns. At least that's one of our problems.	Who can be a supermaster to organize cognitive load across the teams	No mention of constraints beyond cognitive load.
It's very hard to get an holistic overview. It's quite easy to miss some facts out..	Team too small	„gärtchendenken“ vs. „one company spirit“
Forcing reducing of collaboration (maybe should be cross enablement), may cause teams to pursue local goals rather than enterprise goals	A think a distinction has to be made, communicating while planning vs. Communicating while executing the plan.	Some are much better at seeing the big pic, others at seeing the tree but cutting the forest into sections, that can cooperate well anyway
New Platform as SelfService can be a big challenge	knowledge transfer, coffe talks etc.	When Team composition/boundary is defined by conflicting incentives
If systems/teams are setup for throughout, then autonomy is not THE goal, thefore collaboration needs to be planned at scale	"Short time" issue squads	TT does not talk about depth of dependency trees between stream aligned teams and associated trade offs.
Conway's Law?	No forces between teams explicitly represented How models of one team percolate to other teams	Teams split all over the world -> time zone
Different Project priorities in Teams by Management.	Depends a lot on company culture, touches Management, HR and easily gets highly political	Enabling Team = where does it Stop?
Dependency Ladder	Humans general fear of change	Team Topologie Chance continously
	unknown technologie and tech depth	Who decides the team structures?
		It's difficult to make everybody happy

**Strategic Domain
Driven Design**
also has a technique
which can be used
to visualize
sociotechnical
relationships:

CONTEXT MAPS



Context Maps in Domain Driven Design address relationships between Bounded Contexts and teams. They start „bounded context first“.



259



433



78



72



25

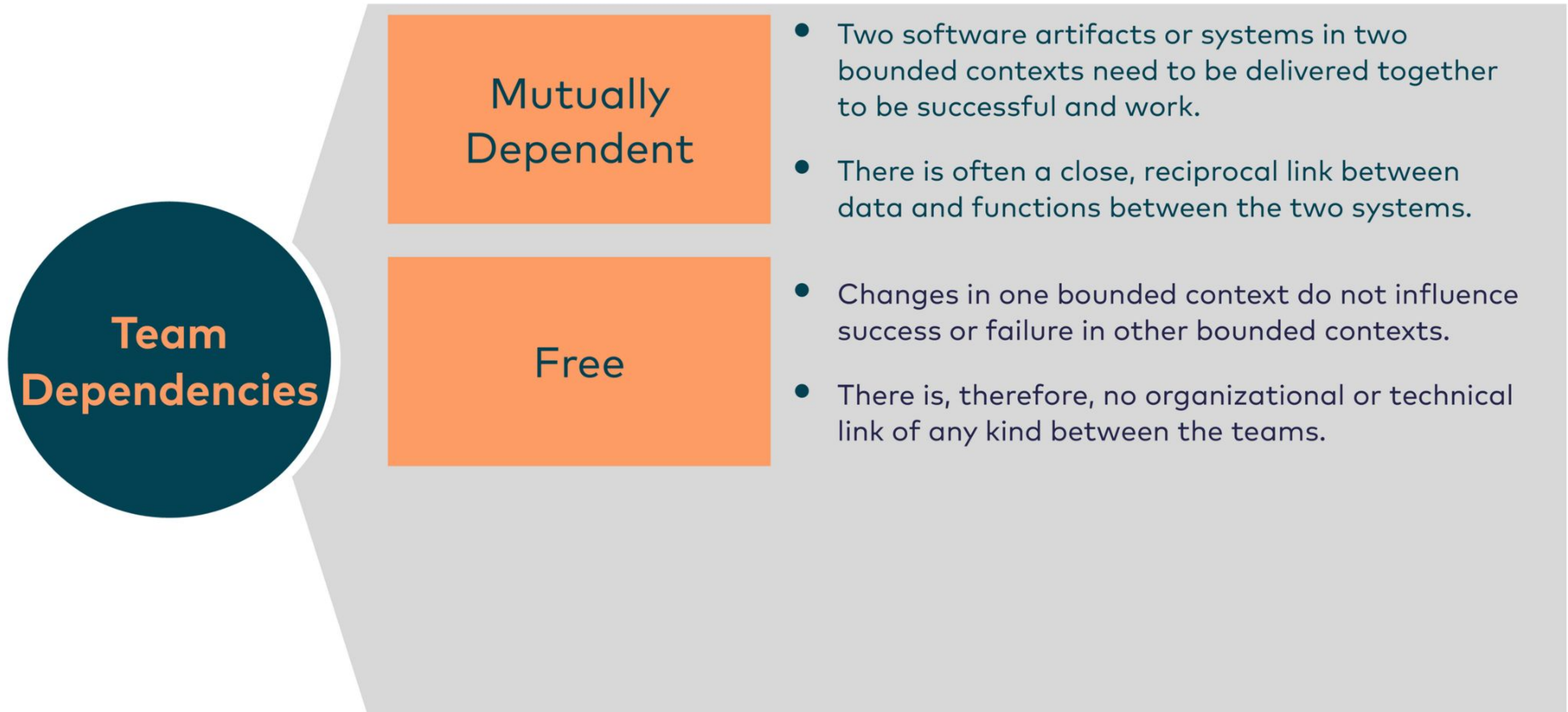
Dependencies between teams



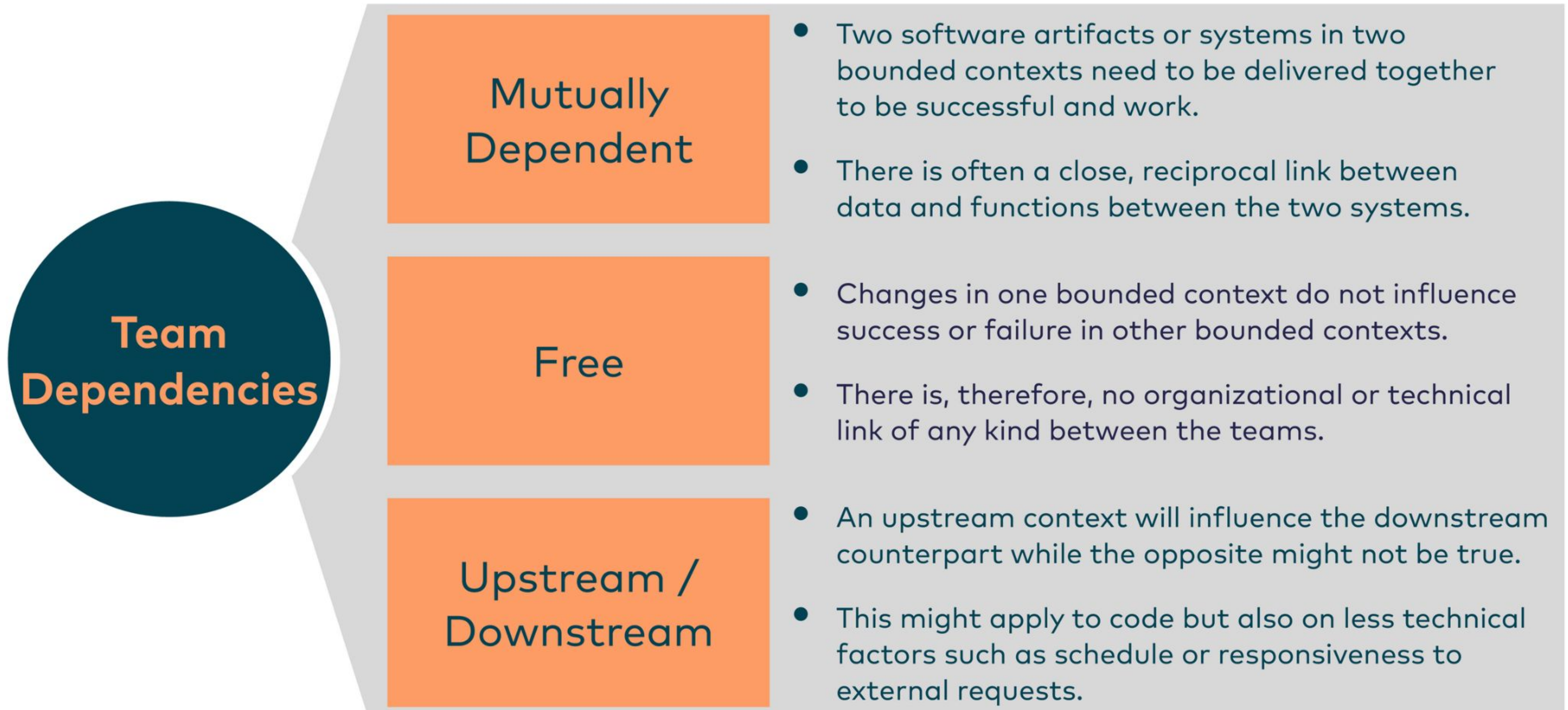
Mutually Dependent

- Two software artifacts or systems in two bounded contexts need to be delivered together to be successful and work.
- There is often a close, reciprocal link between data and functions between the two systems.

Dependencies between teams



Dependencies between teams



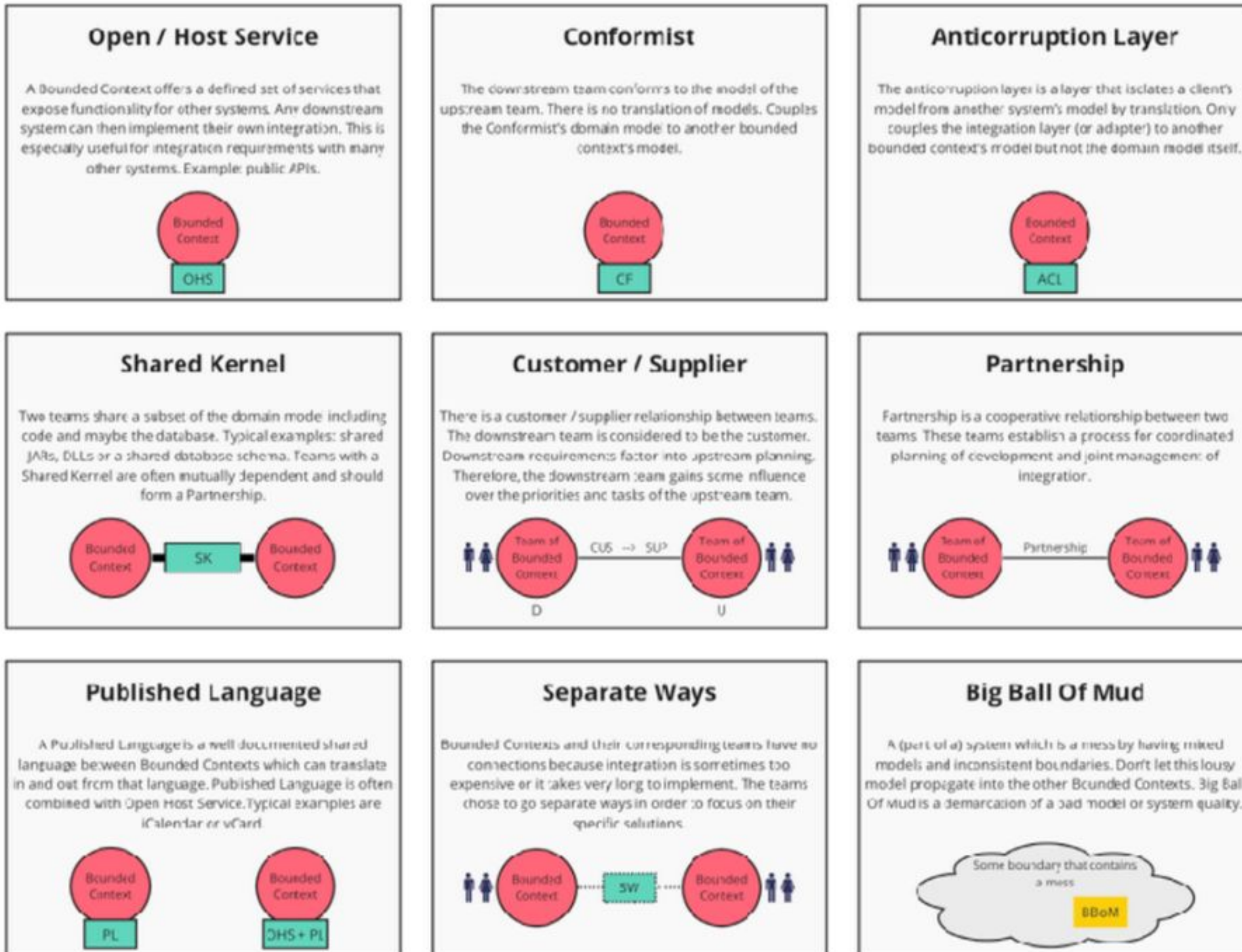
The context map uses patterns to describe the contact between bounded contexts and teams

- 🌀 Partnership
- 🌀 Shared Kernel
- 🌀 Customer / Supplier
- 🌀 Conformist
- 🌀 Anticorruption Layer
- 🌀 Separate Ways
- 🌀 Open / Host Service
- 🌀 Published Language
- 🌀 Big Ball Of Mud

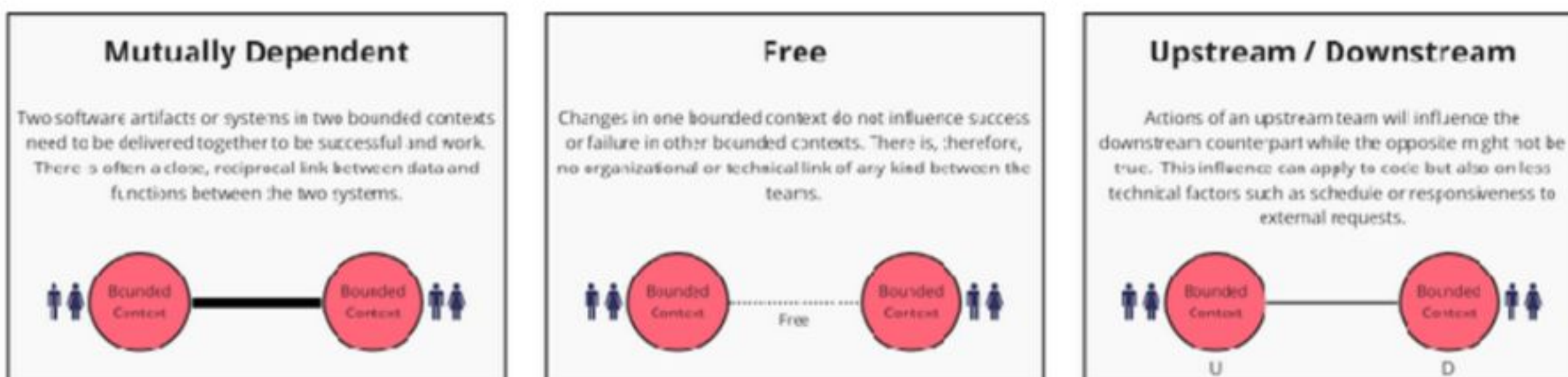
These patterns address a diverse variety of perspectives

Context Map Cheat Sheet

Context Map Patterns



Team Relationships



Check out DDD Crew on GitHub

- Cheat Sheet for all of the patterns and Team Relationships
- Context Mapping Starter Kit for Miro (as a downloadable Board Backup)
- Creative Commons

<https://github.com/ddd-crew/context-mapping>



259



433



78



72



25



I'll just mention a few of the patterns here which we will later pick up for the combination with Team Topologies.



259



433



78



72

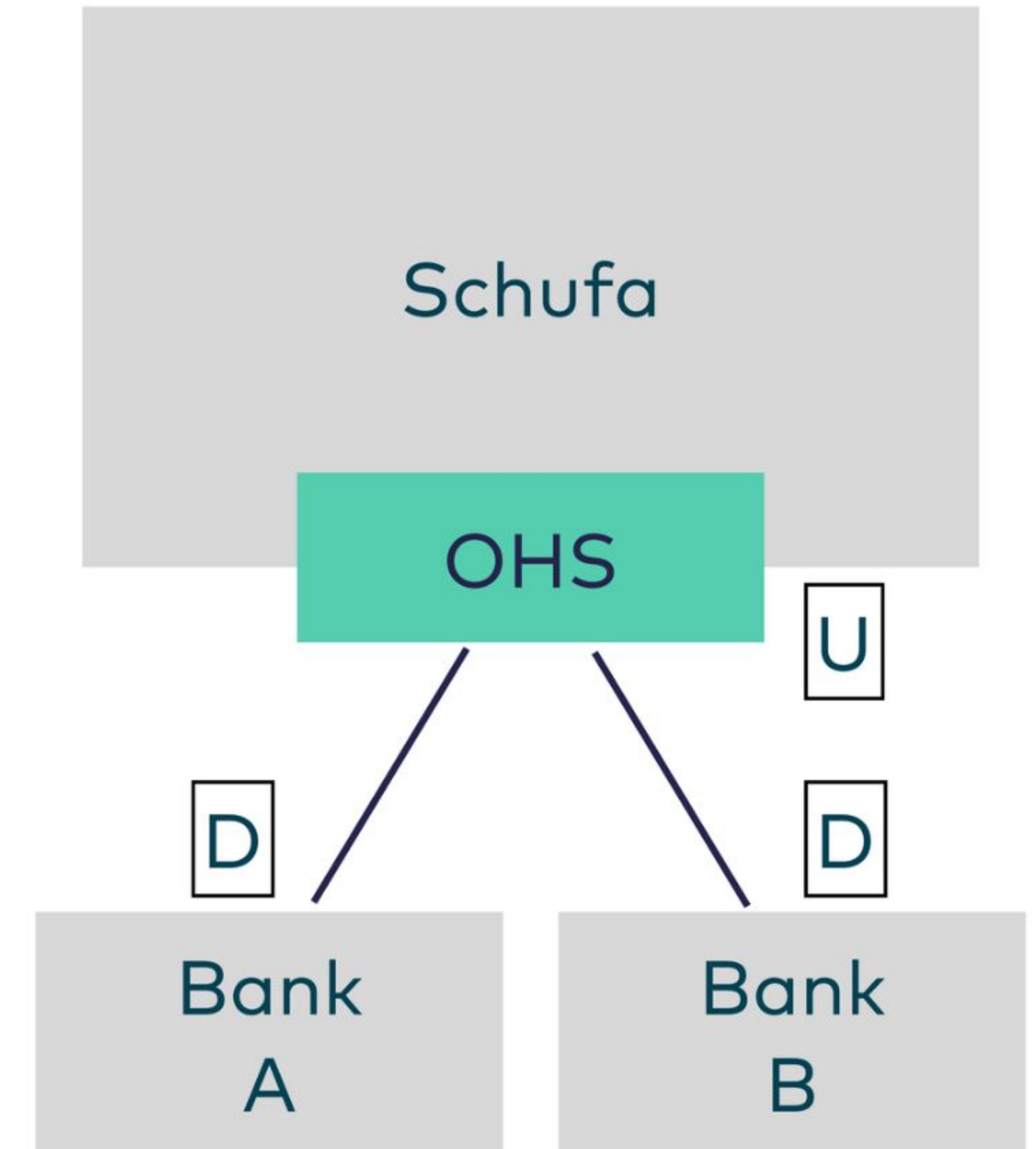


25

Open-host Service

The Open-host Service is a public API

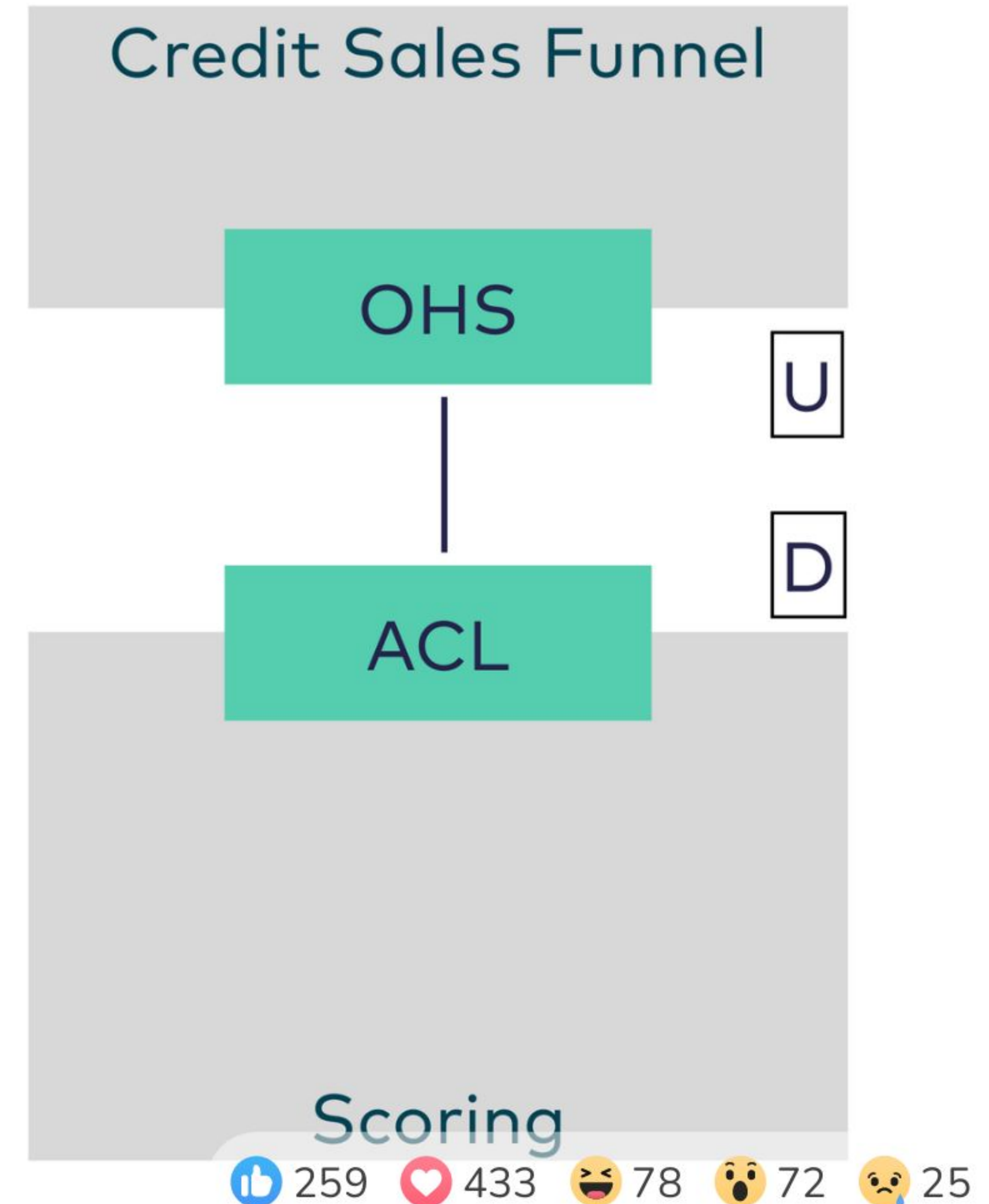
- One API for several consumers
- No point-to-point API
- Has a common, general purpose model and functionality
- The team providing the Open-host Service is an upstream team



Anticorruption Layer

The Anticorruption Layer translates one model to another one

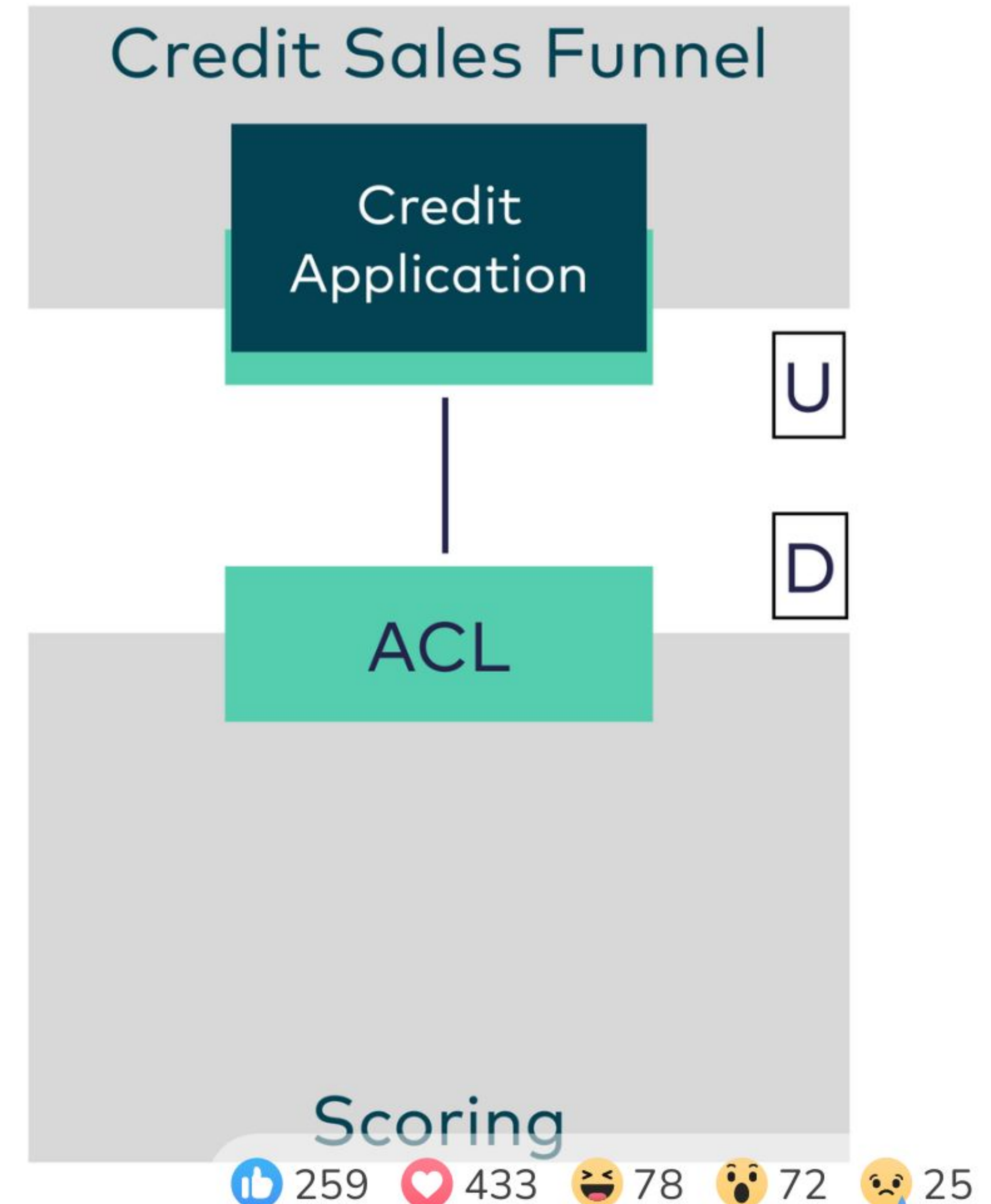
- Transforms an external model from another team / bounded context / system to another internal one
- Reduces the amount of coupling to a single layer
- The team implementing an Anticorruption Layer is always downstream



Anticorruption Layer

The Anticorruption Layer translates one model to another one

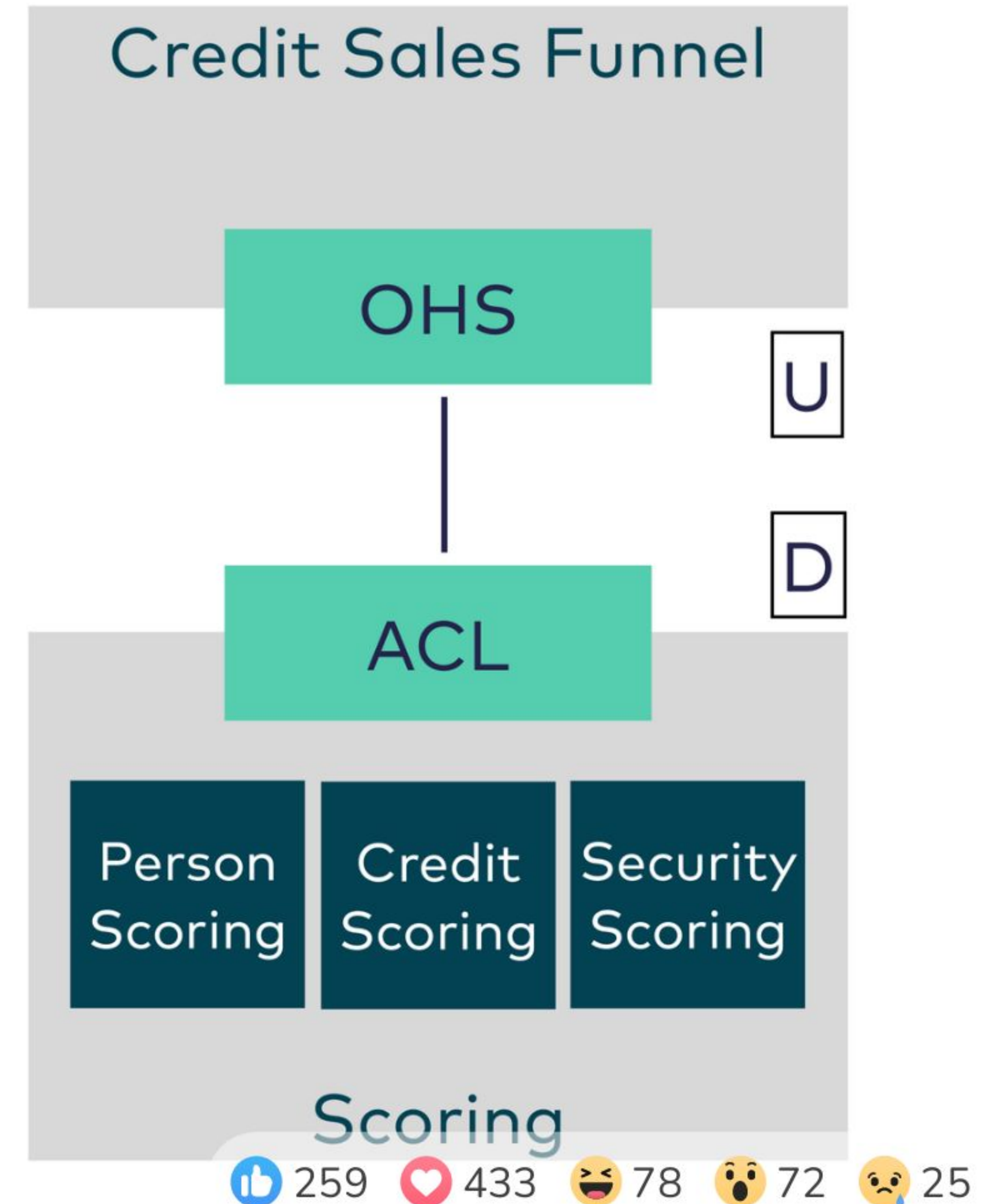
- Transforms an external model from another team / bounded context / system to another internal one
- Reduces the amount of coupling to a single layer
- The team implementing an Anticorruption Layer is always downstream



Anticorruption Layer

The Anticorruption Layer translates one model to another one

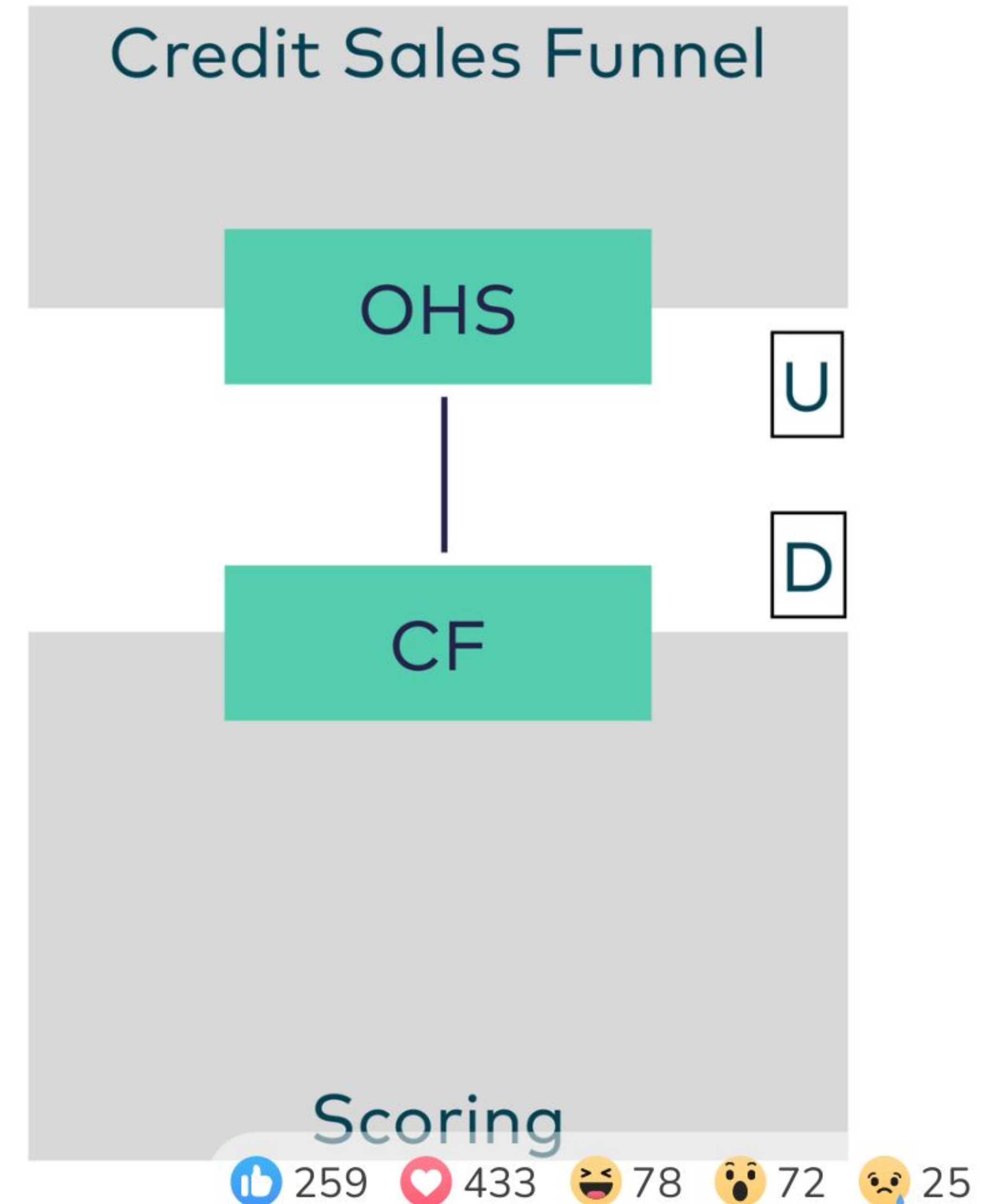
- Transforms an external model from another team / bounded context / system to another internal one
- Reduces the amount of coupling to a single layer
- The team implementing an Anticorruption Layer is always downstream



Conformist

The Conformist slavishly adheres to the upstream model

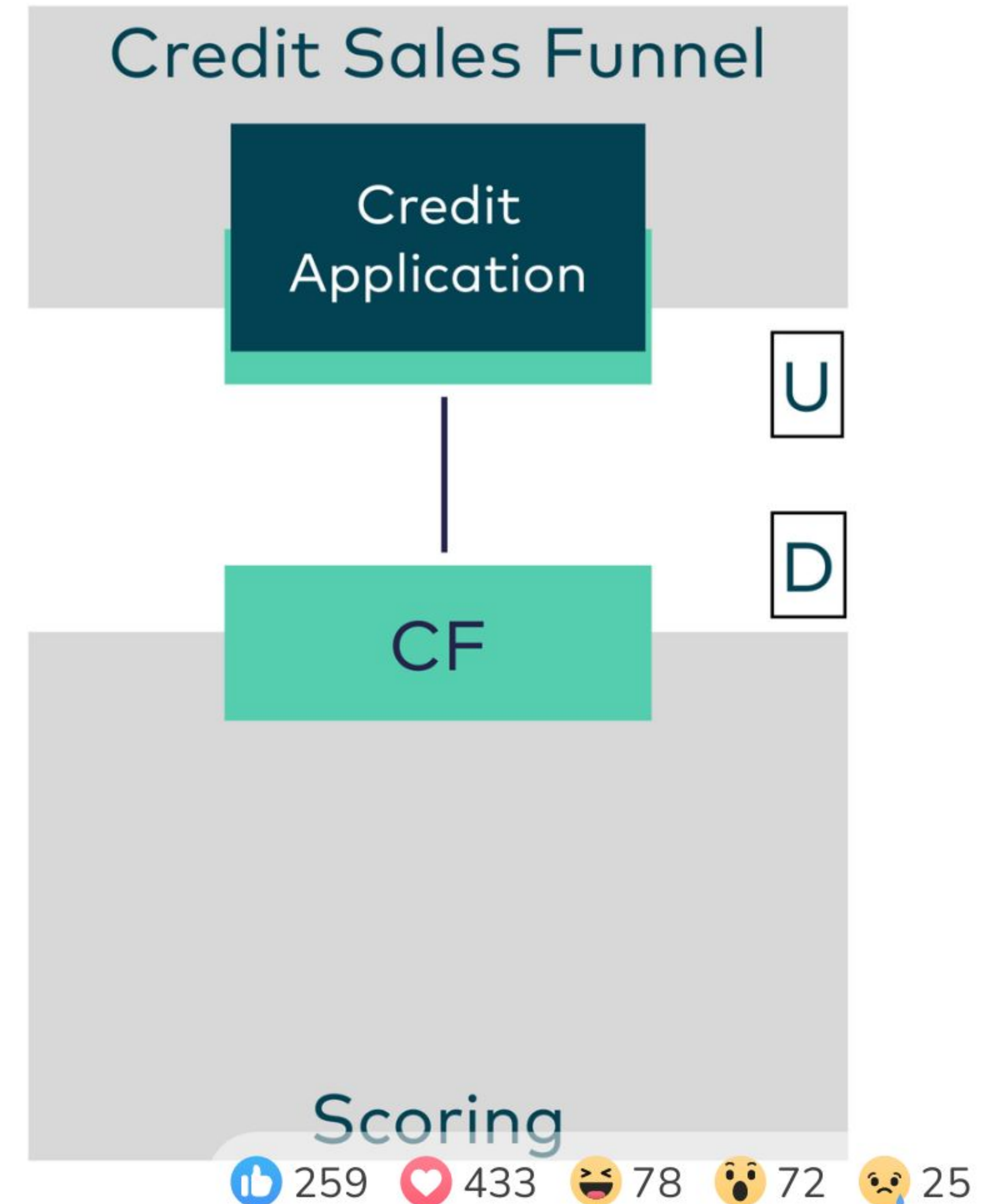
- There is no model-to-model transformation
- Motivation: Simplicity, contracts, force or delight (for the upstream model)
- The team implementing a Conformist is always downstream



Conformist

The Conformist slavishly adheres to the upstream model

- There is no model-to-model transformation
- Motivation: Simplicity, contracts, force or delight (for the upstream model)
- The team implementing a Conformist is always downstream



Conformist

The Conformist slavishly adheres to the upstream model

- There is no model-to-model transformation
- Motivation: Simplicity, contracts, force or delight (for the upstream model)
- The team implementing a Conformist is always downstream

It looks like you have been logged out

You will be directed to the Login screen in 7 seconds.

Login again

Credit Sales Funnel

OHS

U

CF

D

Credit
Application

Scoring

👍 259 ❤️ 433 😄 78 😱 72 😞 25

Which comments apply best to this Context Map?



Shared Kernel

Shared Kernel is a subset of a domain model that two teams share

- „Physically“ shared artifact between two teams
- Examples: shared JARs or database
- High degree of coupling requires a high amount of coordination between the involved teams
- Shared Kernel is no Anti-Pattern but use with caution

It looks like you have been logged out

You will be directed to the Login screen in 4 seconds.

Login again

Credit Sales Funnel

Credit
Application

Scoring



259



433



78



72



25

Shared Kernel

Shared Kernel is a subset of a domain model that two teams share

- „Physically“ shared artifact between two teams
- Examples: shared JARs or database
- High degree of coupling requires a high amount of coordination between the involved teams
- Shared Kernel is no Anti-Pattern but use with caution

It looks like you have been logged out

You will be directed to the Login screen in 3 seconds.

Login again

Credit Sales Funnel

SK

Scoring



259



433



78



72



25

Partnership

Partnership is about cooperative relationships between teams

- Establishes a process for coordinated planning of development and management of integration
- Not technical at all, Partnership is plain organizational
- Recommended for teams which depend on a Shared Kernel

It looks like you have been logged out

You will be directed to the Login screen in 2 seconds.

Login again

Ok, let's coordinate our efforts

We want to adjust something



Scoring

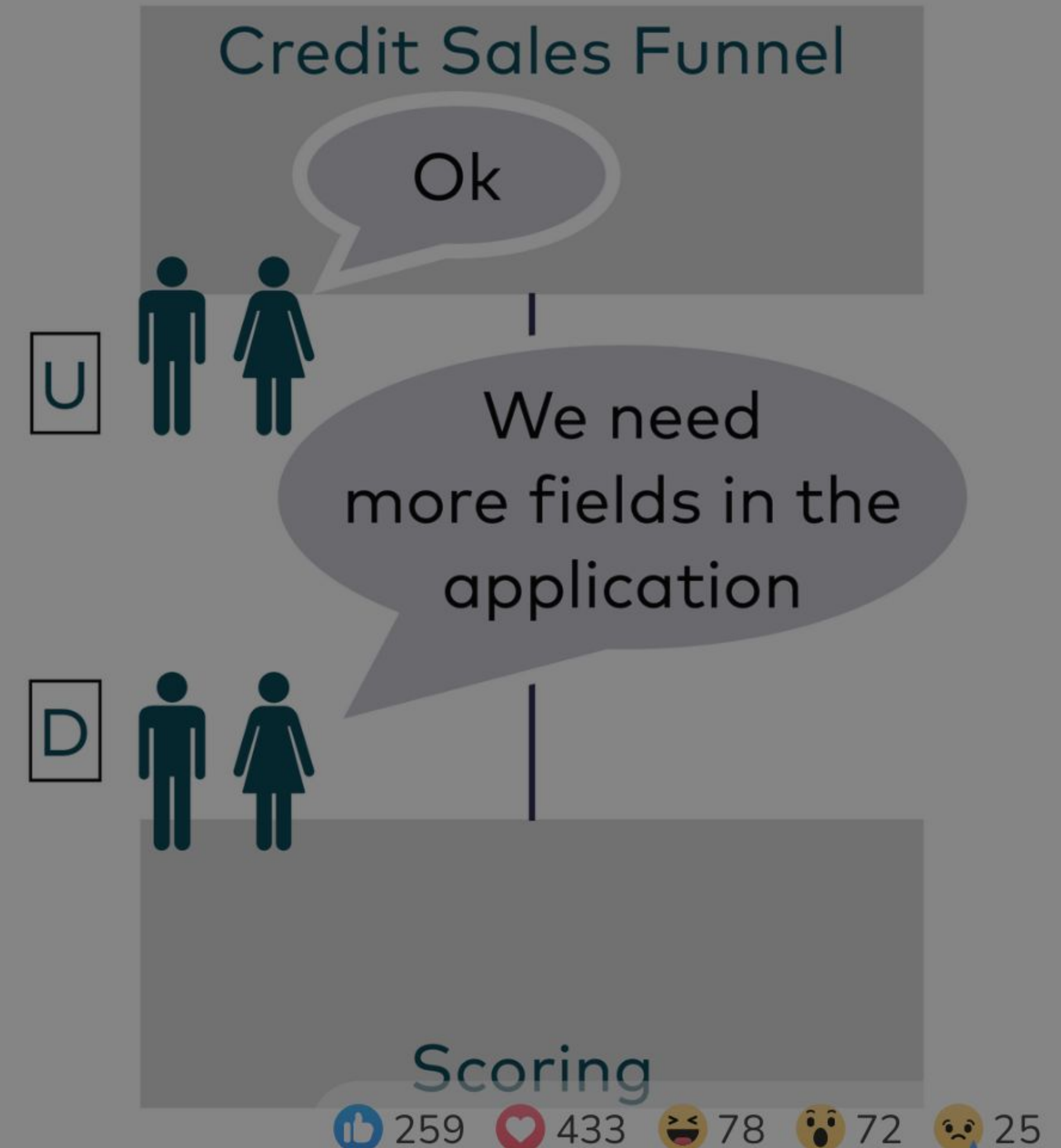
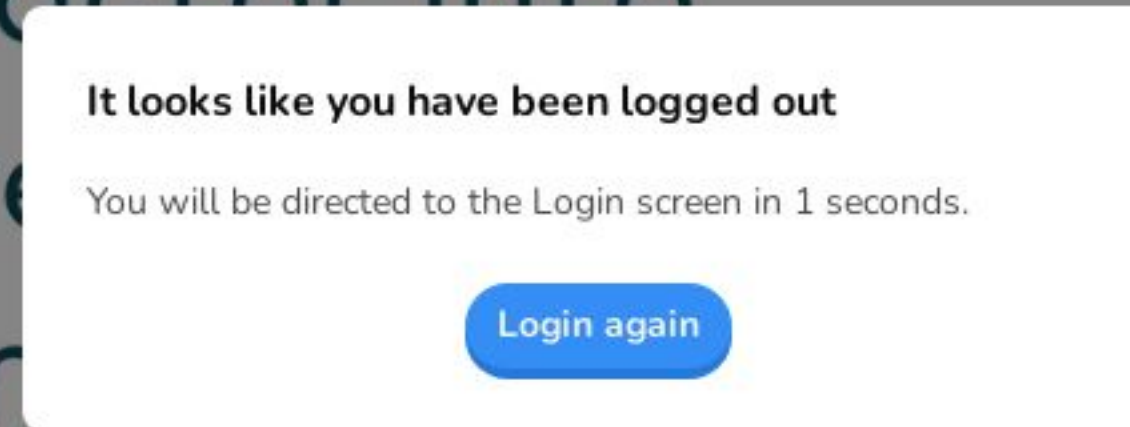


Credit Sales Funnel

Customer-Supplier

A Customer-Supplier development gives the downstream team some influence

- Downstream requirements factor into upstream planning. Therefore downstream team gains some influence over the priorities and tasks of the upstream team
- Customer-Supplier is organizational
- Mind „vetoing customer“ and customer against an OHS as anti-patterns



Customer-Supplier

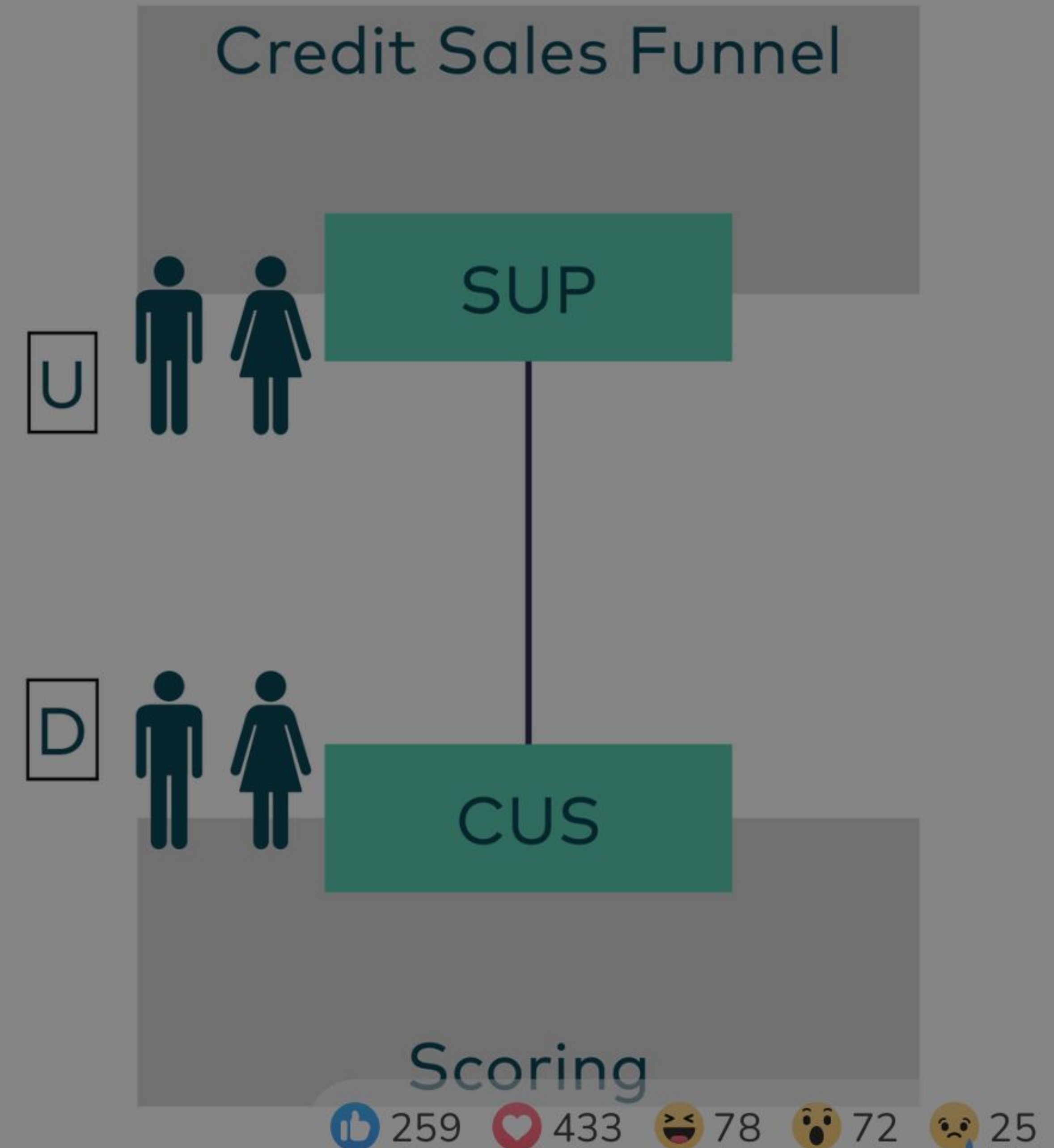
A Customer-Supplier development gives the downstream team some influence

- Downstream requirements factor into upstream planning. Therefore downstream team gains some influence over the priorities and tasks of the upstream team
- Customer-Supplier is organizational
- Mind „vetoing customer“ and customer against an OHS as anti-patterns

It looks like you have been logged out

You will be directed to the Login screen in 0 seconds.

Login again

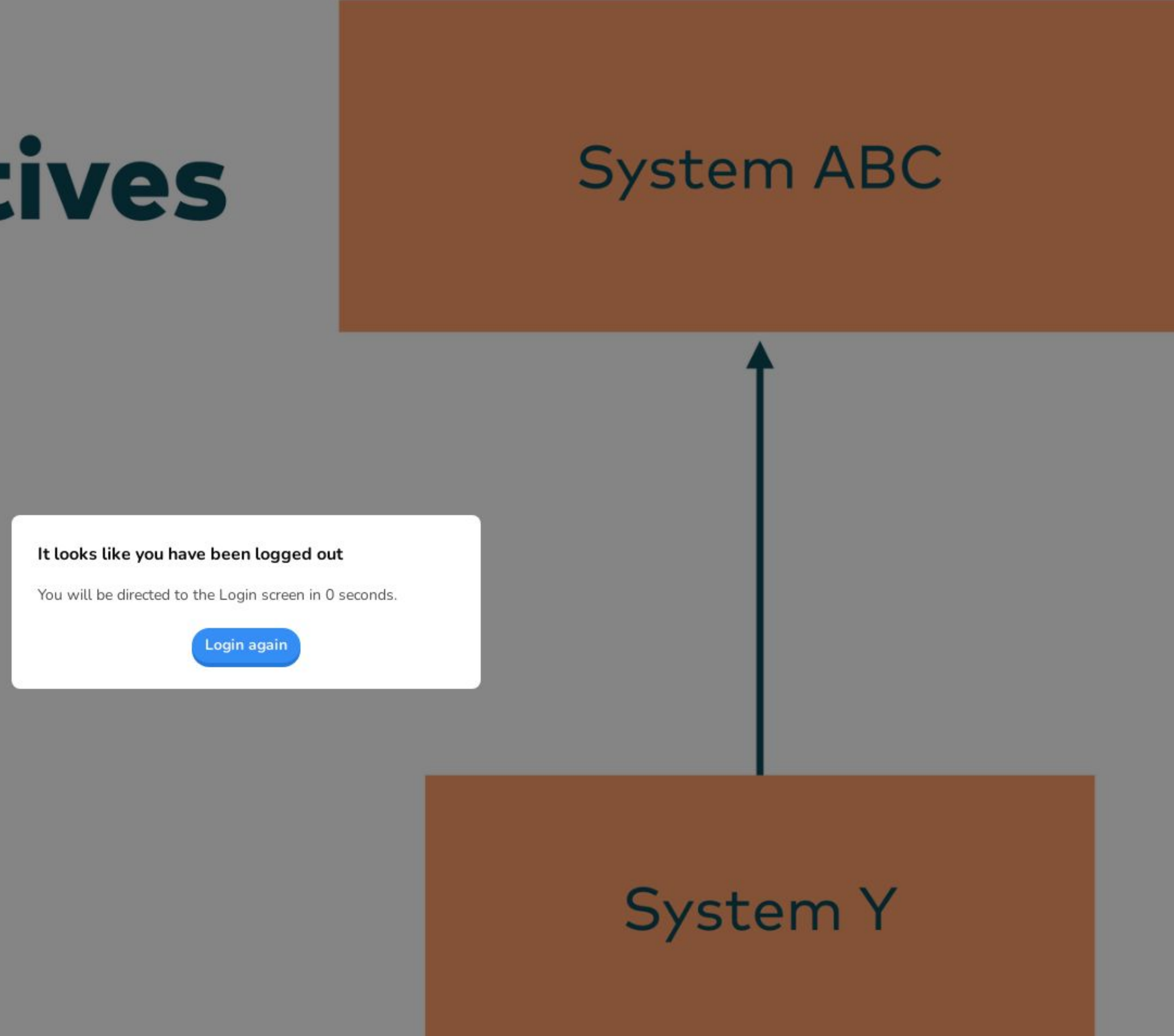


Which statements are valid for the supplier?



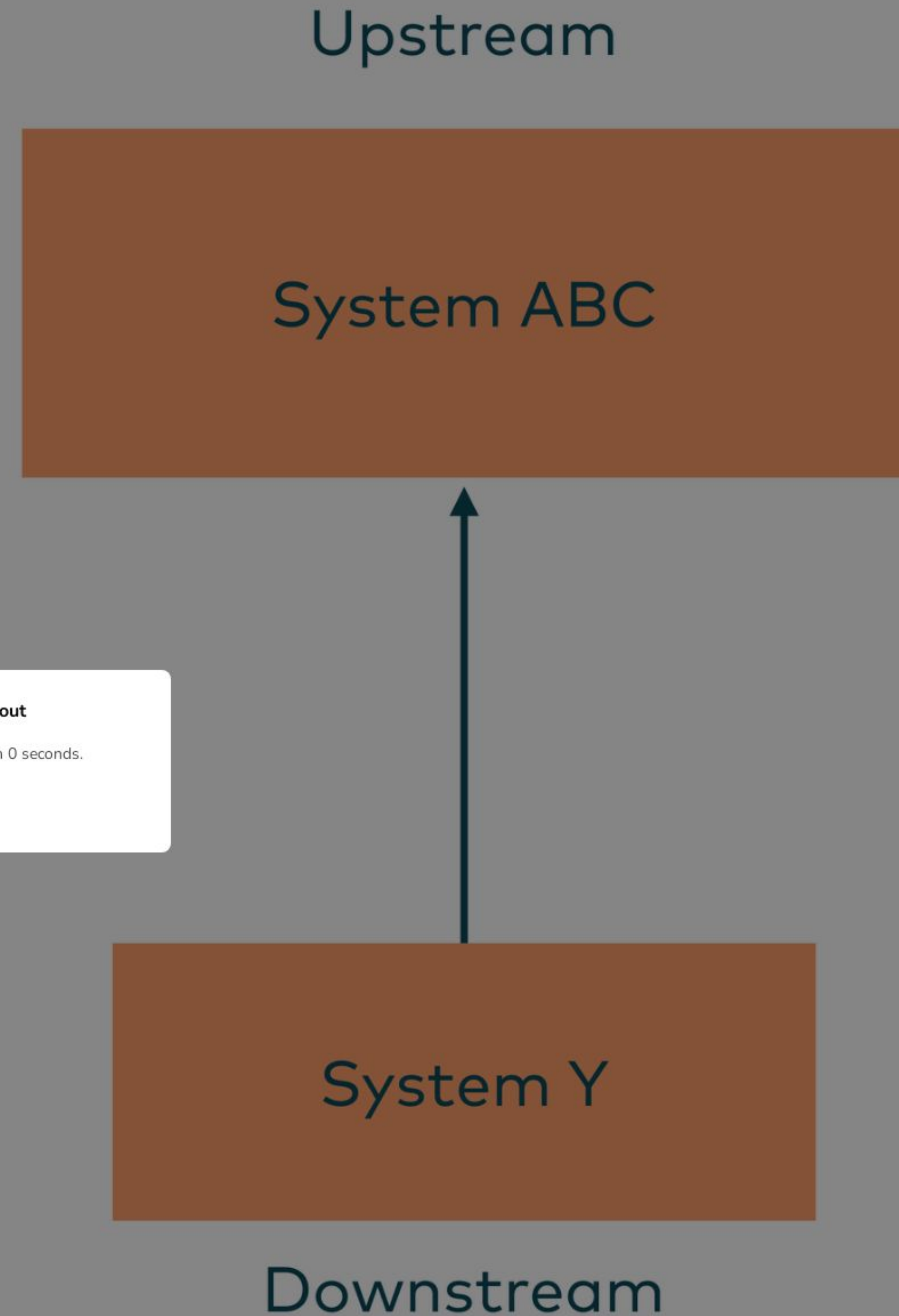
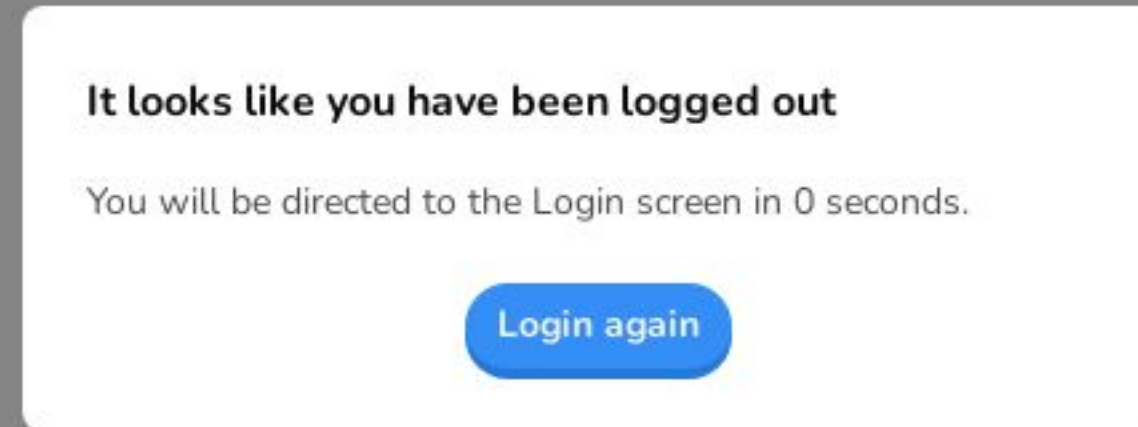
You can visualize different perspectives

- Call Relationship



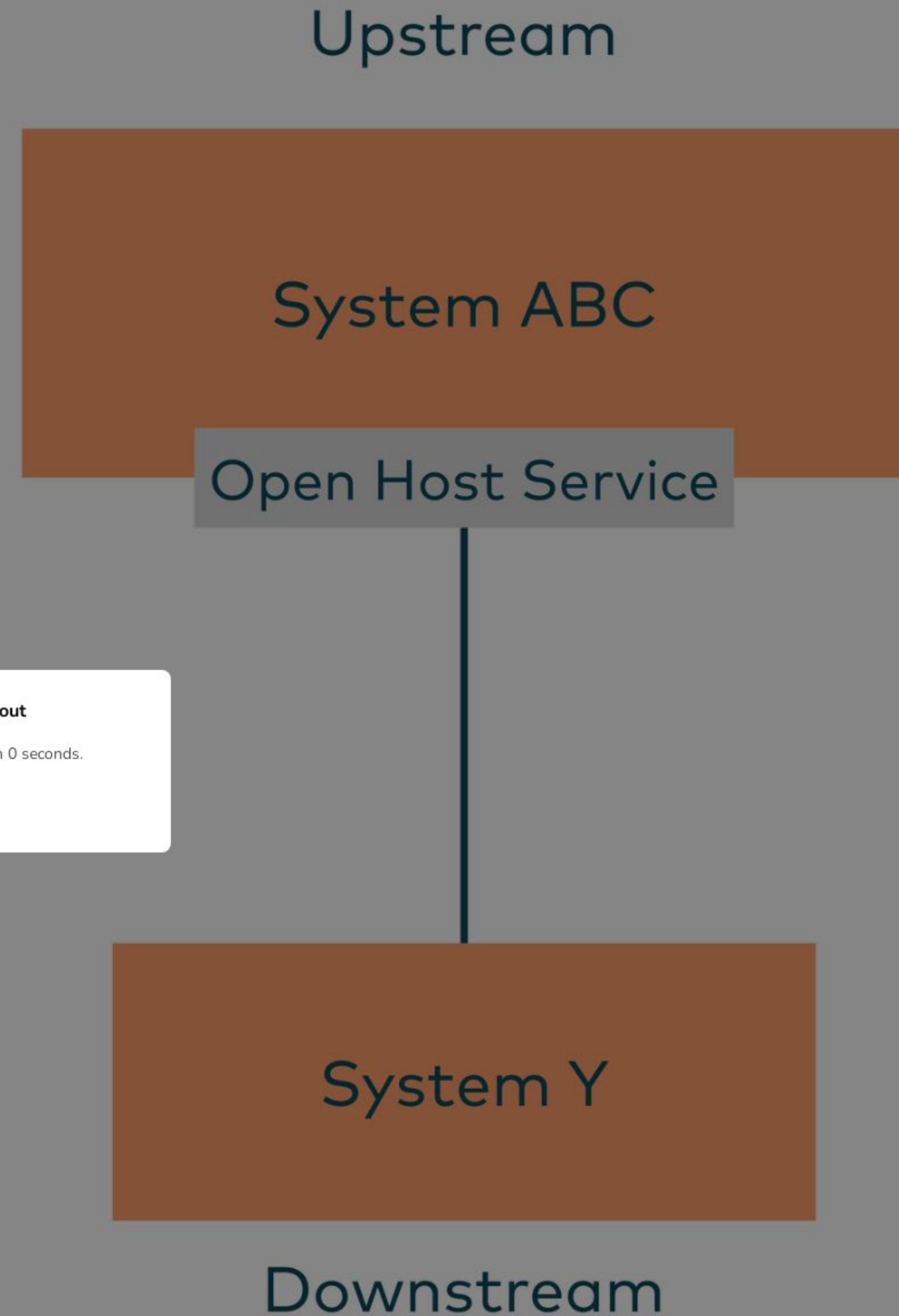
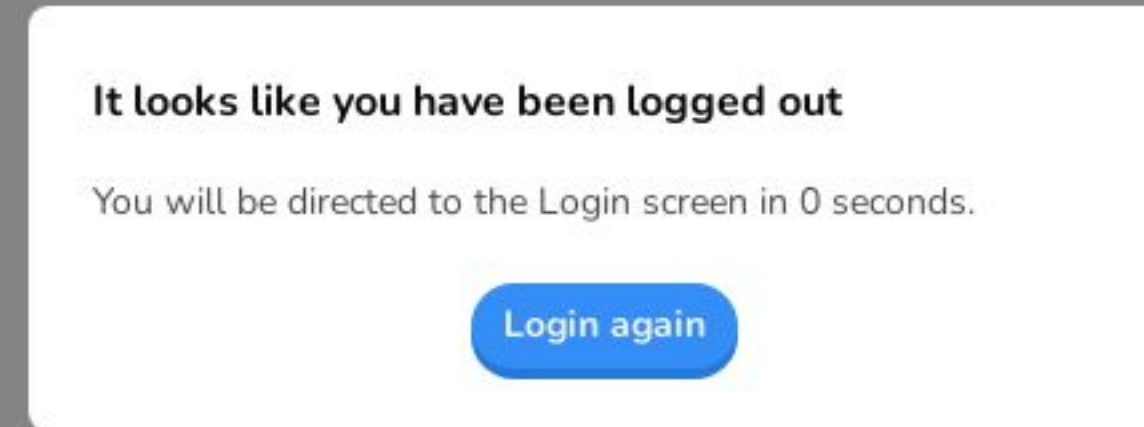
You can visualize different perspectives

- Call Relationship
- Team Relationship - Level 1



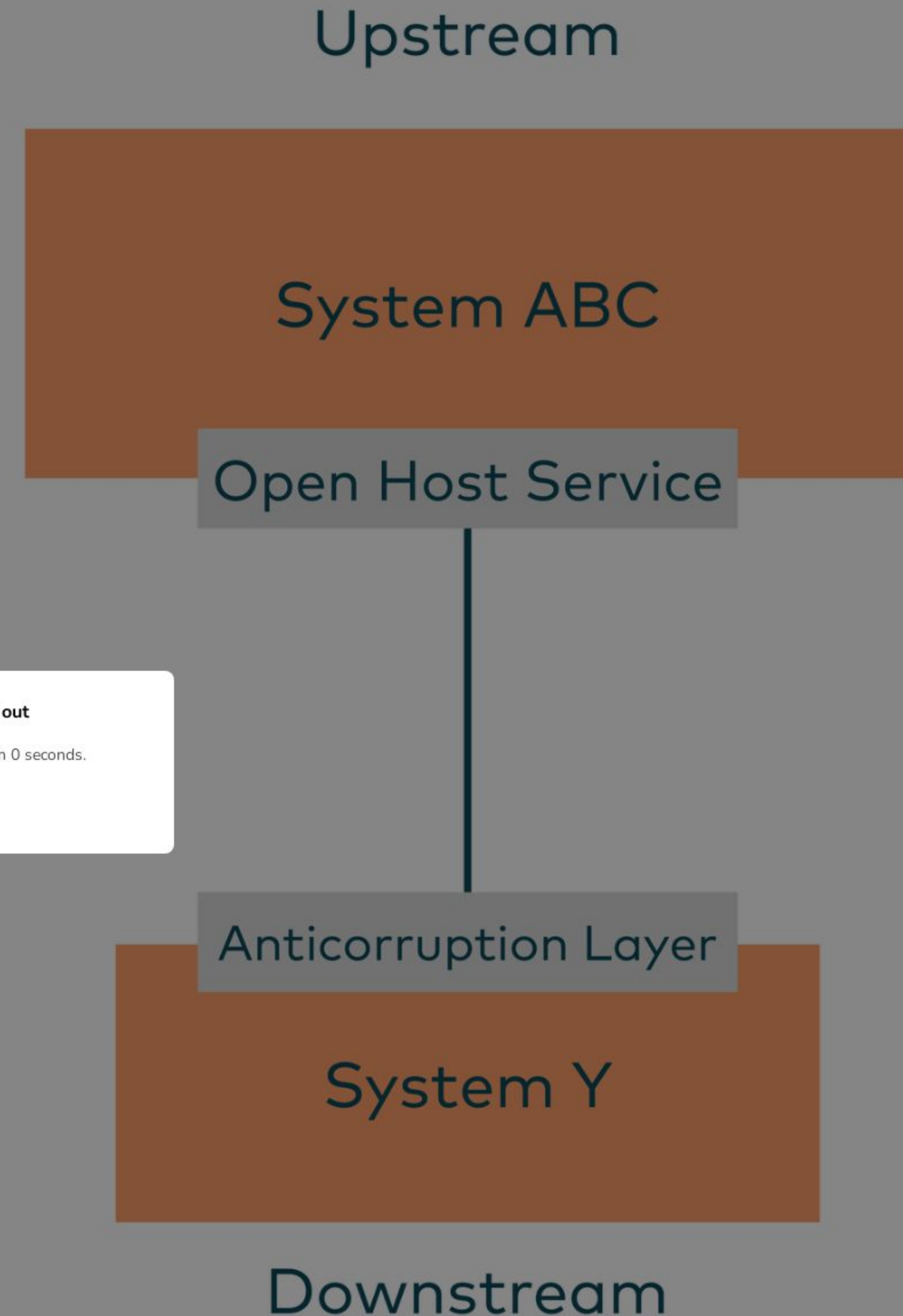
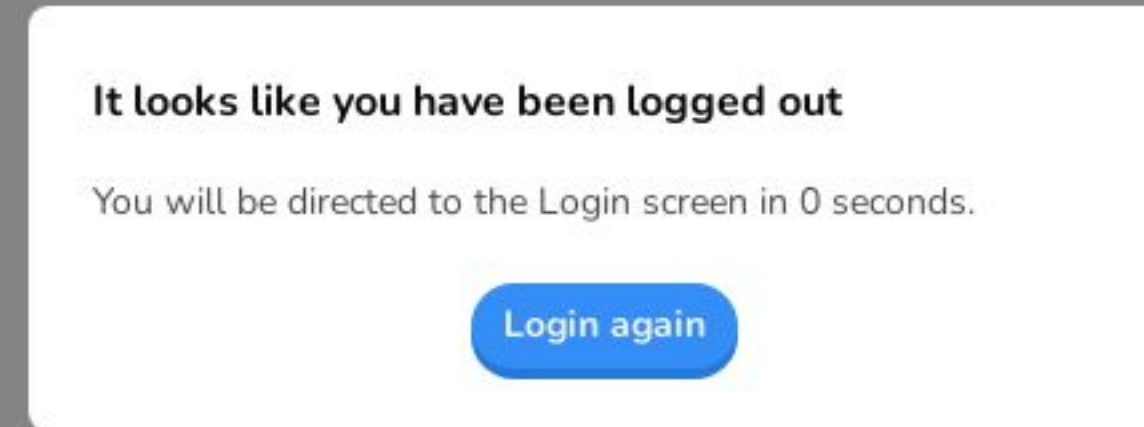
You can visualize different perspectives

- Call Relationship
- Team Relationship - Level 1
- API Level



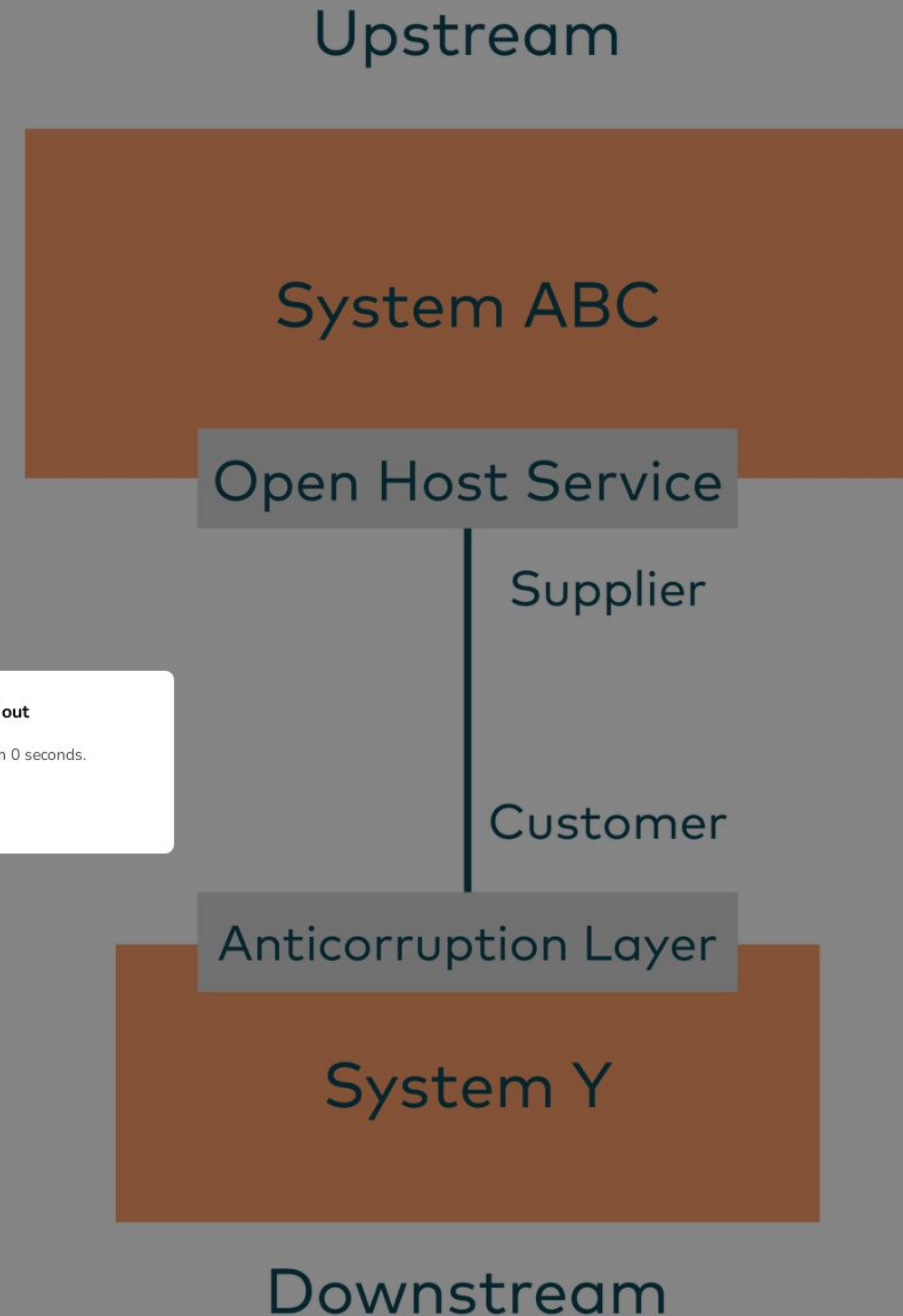
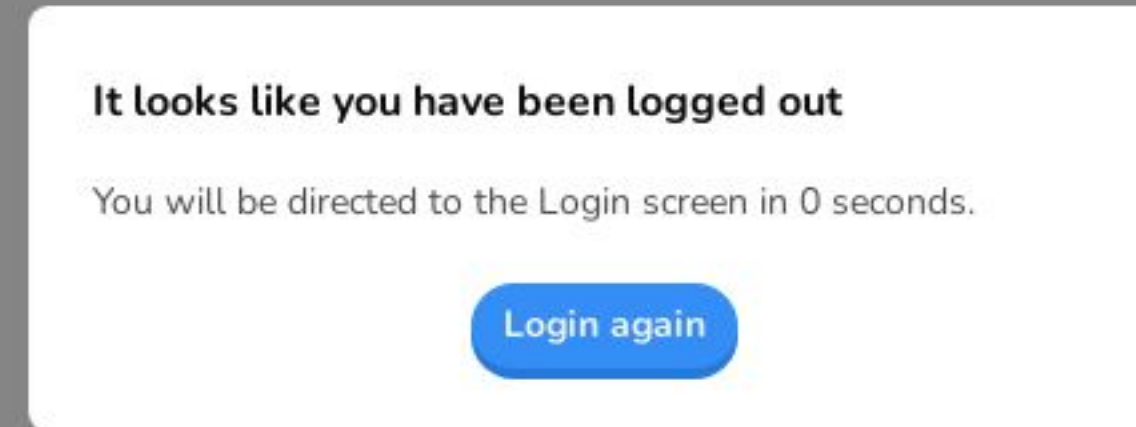
You can visualize different perspectives

- Call Relationship
- Team Relationship - Level 1
- API Level
- Model Propagation



You can visualize different perspectives

- Call Relationship
- Team Relationship - Level 1
- API Level
- Model Propagation
- Team Relationship - Level 2



The patterns address various aspects

	Team Relationships	Model Propagation	API / „technical“
Open-host Service	(✓)		✓
Anticorruption Layer		✓	
Conformist		✓	
Shared Kernel		✓	(✓)
Partnership	✓		
Customer-Supplier	✓		
Separate Ways	✓	(✓)	
Published Language	✓	(✓)	
Big Ball Of Mud			✓

It looks like you have been logged out

You will be directed to the Login screen in 0 seconds.

Login again

Team
Classifications Teams
Enabling
Team
Boundaries
„first“
Team
Topologies

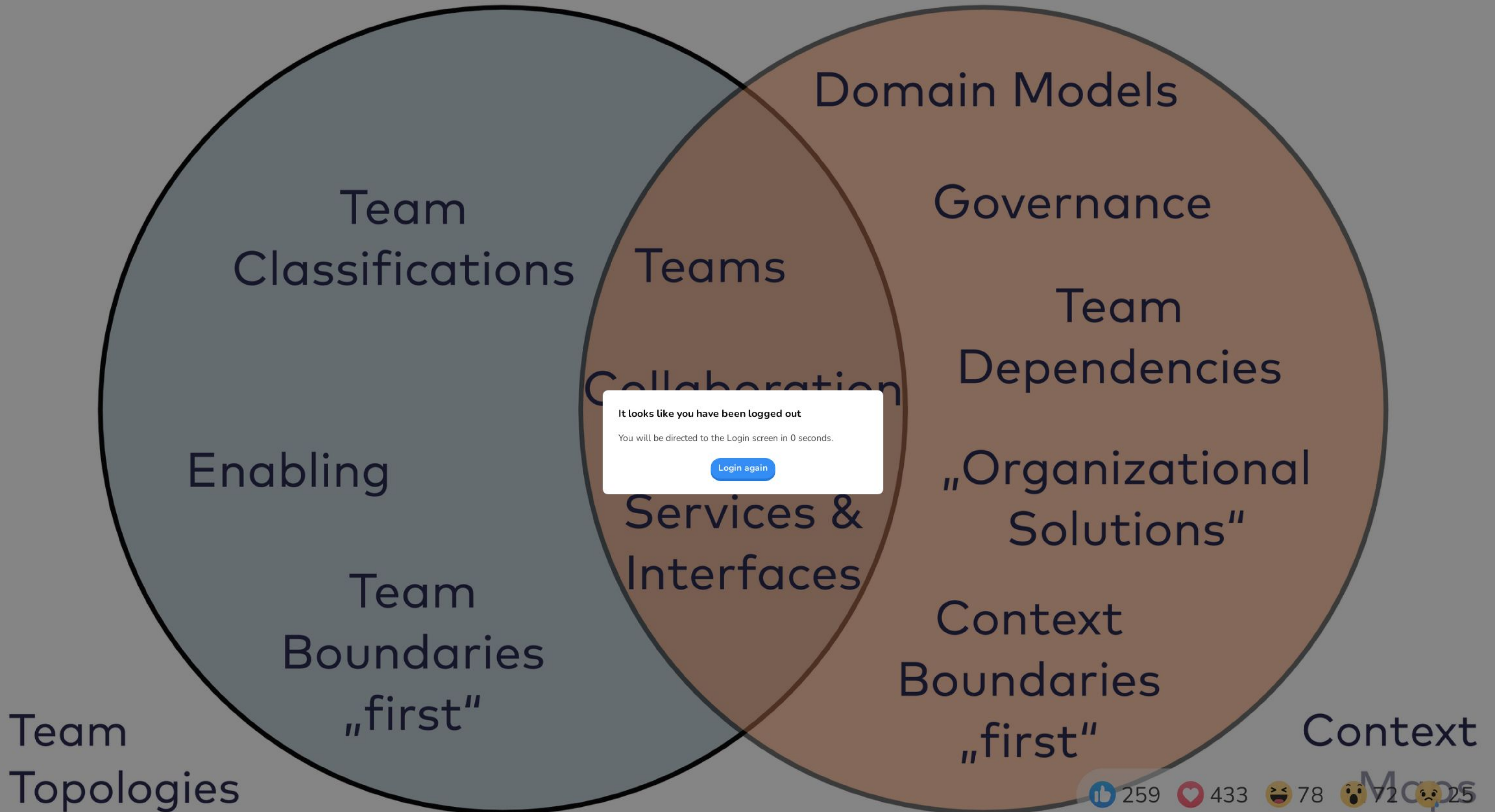
Collaboration
Services &
Interfaces

It looks like you have been logged out

You will be directed to the Login screen in 0 seconds.

Login again

 259  433  78  72  25



It looks like you have been logged out

You will be directed to the Login screen in 0 seconds.

Login again



259



433



78



72



25

YES

It looks like you have been logged out

You will be directed to the Login screen in 0 seconds.

Login again

you can combine Team Topologies and Context Maps



259



433



78



72



25

Which questions would you be able to answer by combining Team Topologies and Context Maps?

if services are split correctly

Heatpoints in the company

Team Responsibilities, Data Ownership, Up/Downstream relationships between teams

Find the right boundaries

Are the boundaries or fences set in s correct way

Do we have the right team for the context

Which relationship/contract is the most sustainable for long-term service evolution

- Reducing frictions and complexity between it-processes
- Align the architecture of the IT through enablement rather than enforcing

Right mix of roles per team

Check how we can align our architecture to the business we make and create the structure for it

Sometimes the QR-Codes (for books) were blocked by the emoticons

How we cut the teams

Are domains aligned with the organization

Political relations between teams and the resulting performance issues of teams

Whether we have the right boundaries

collaboration mode

impact of service interfaces on the way teams are working together

What interactions do i have between stream-aligned teams

Where are problematic interactions between my stream-aligned teams? (since we can't reduce interaction to 0)

WHO hast the Power to Change / prevent change

reduce coupling between teams

Which services should rather be in an enabling team rather than a stream-aligned team

Delivery model of an organization

We can unlock the dysfunction agile coaching approaches that do not model collaboration

Bottlenecks of development

Which teams for which contexts?

Could be a validation method for the correct team and domain setup.

Which components are the interaction between Contexts/Teams

Are the bounded contexts assigned to matching team relationships?

How to tailor teams

Which relationship my teams should have by role in topologies

Responsibility for: API Specification, Documentation, Contract Tests/ Test Data

😞 where do we lack the required skills to build teams?

How „aligned“ are stream-aligned teams?



It looks like you have been logged out

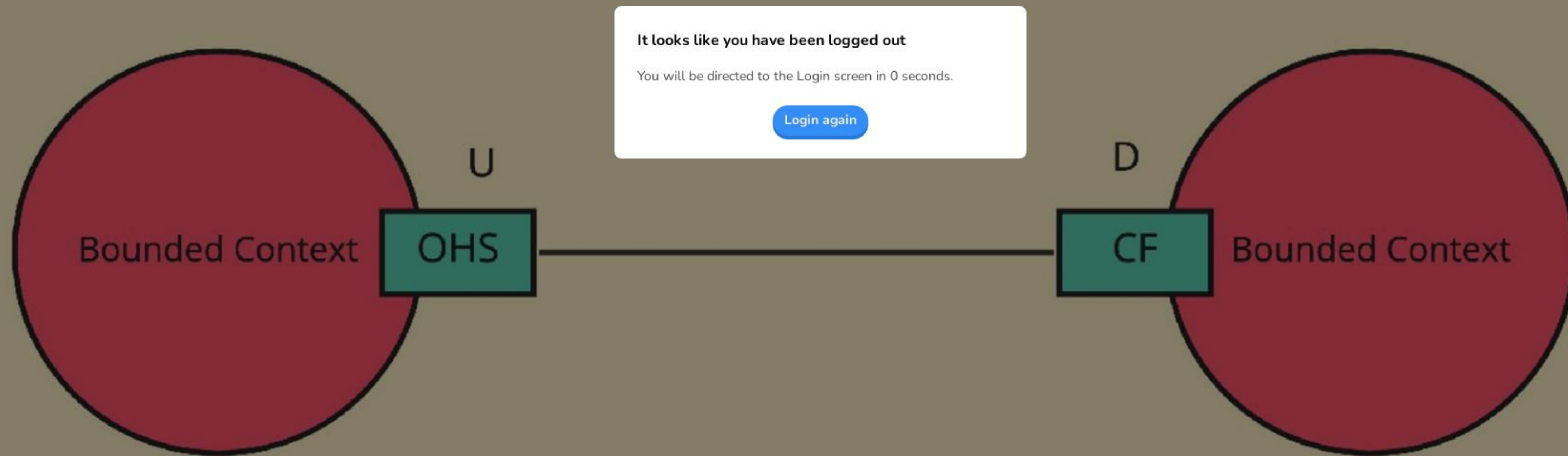
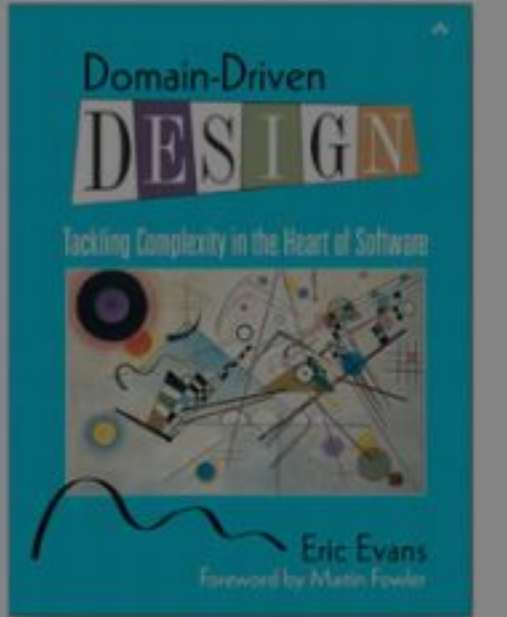
You will be directed to the Login screen in 0 seconds.

Login again

Stream-aligned team

How „aligned“ are stream-aligned teams?

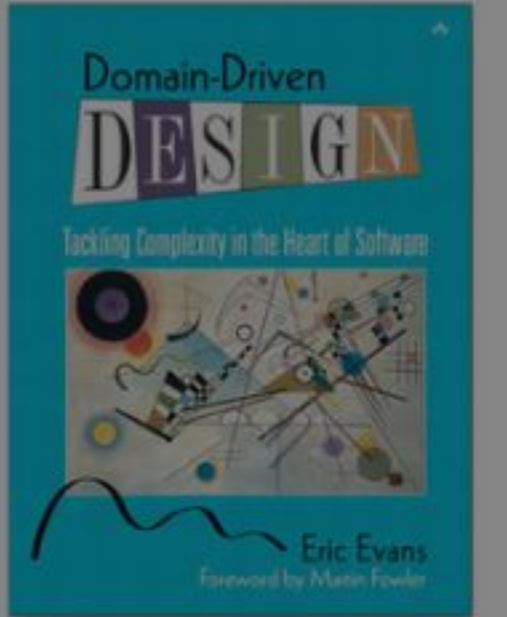
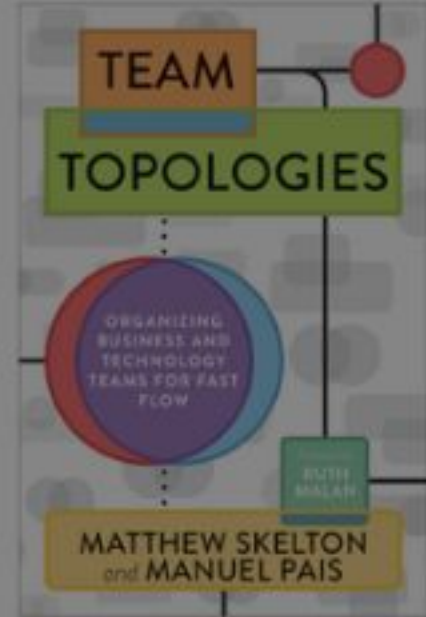
Example: not so aligned



Stream-aligned team

How „aligned“ are stream-aligned teams?

Example: aligned



It looks like you have been logged out

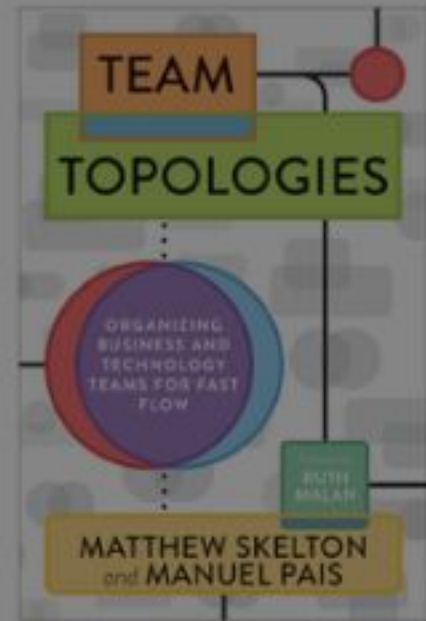
You will be directed to the Login screen in 0 seconds.

Login again

Bounded Context

Stream-aligned team

How „complicated“ is the responsibility of a complicated subsystem team?



It looks like you have been logged out

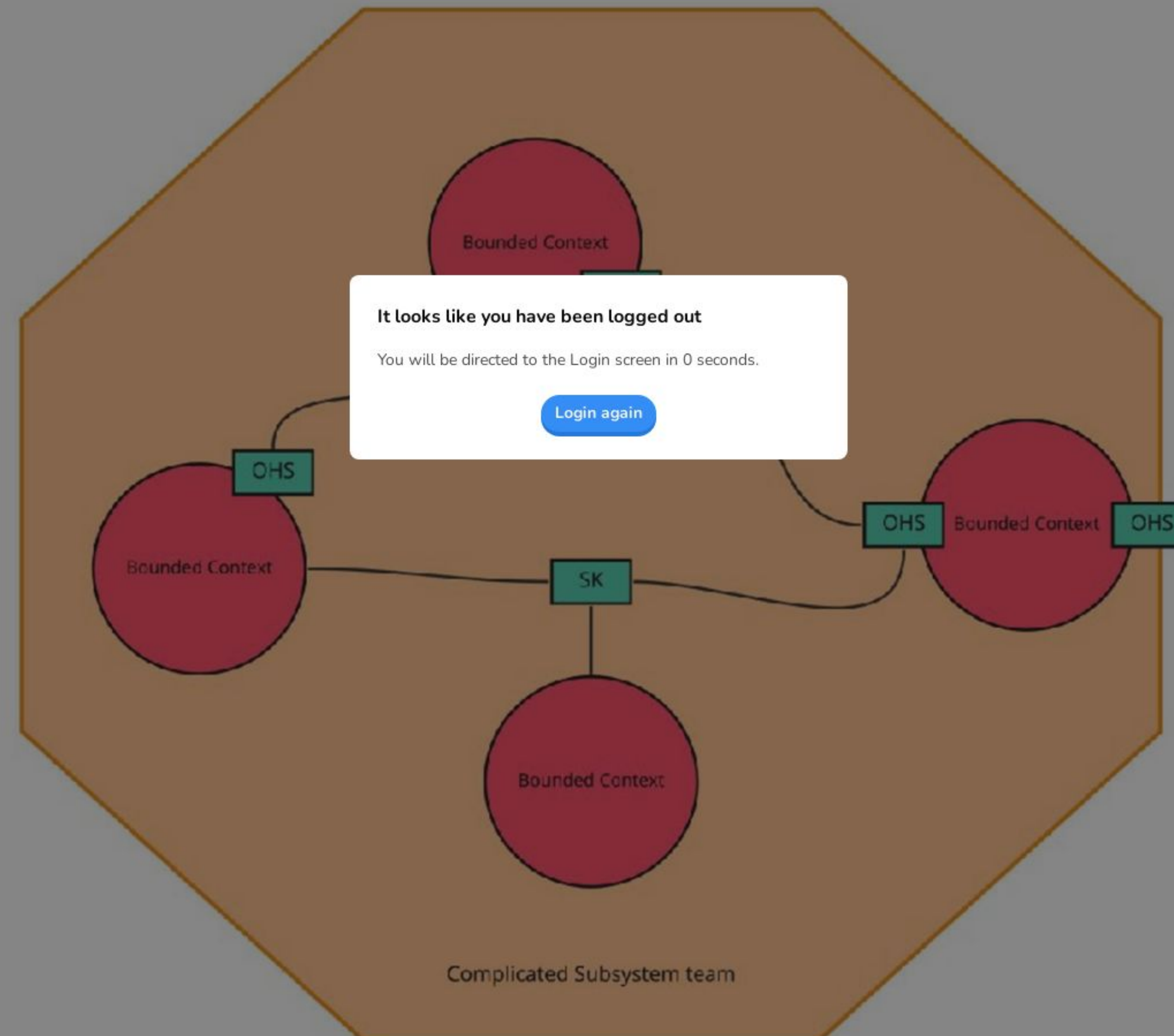
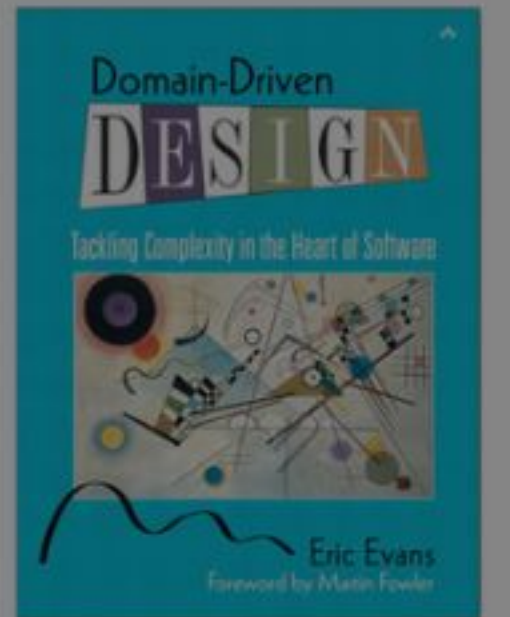
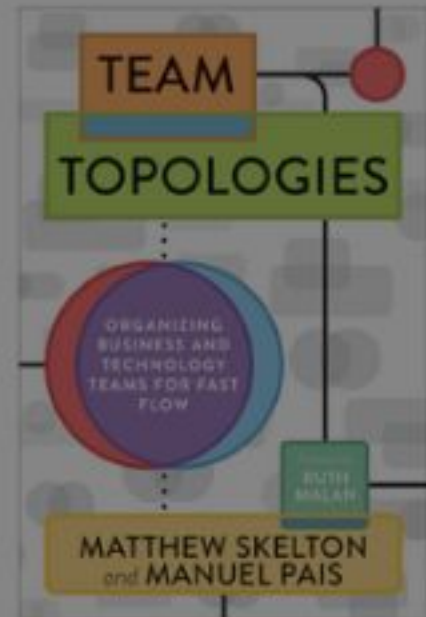
You will be directed to the Login screen in 0 seconds.

Login again

Complicated Subsystem team

👍 259 ❤️ 433 😂 78 😱 72 😞 25

How „complicated“ is the responsibility of a complicated subsystem team?



Learning

with context maps v to the boundaries of teams in order to see how they are mapped to their internal responsibilities / software boundaries and if this suits the type of team (stream-aligned, ...)

It looks like you have been logged out

You will be directed to the Login screen in 0 seconds.

Login again



259



433



78



72



25

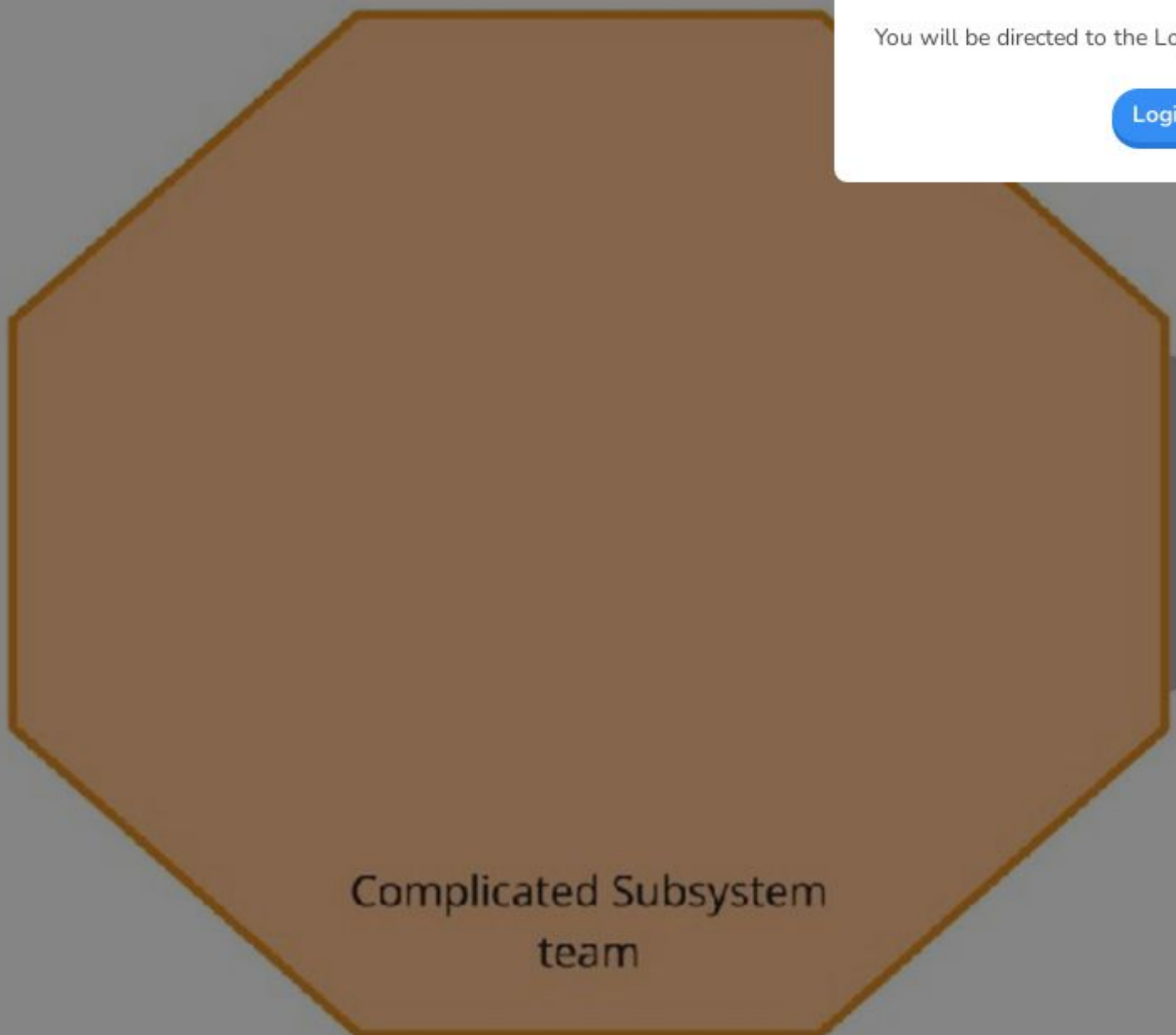
A Complicated Subsystem Team providing a service (X-aa-S) to a Stream-aligned team



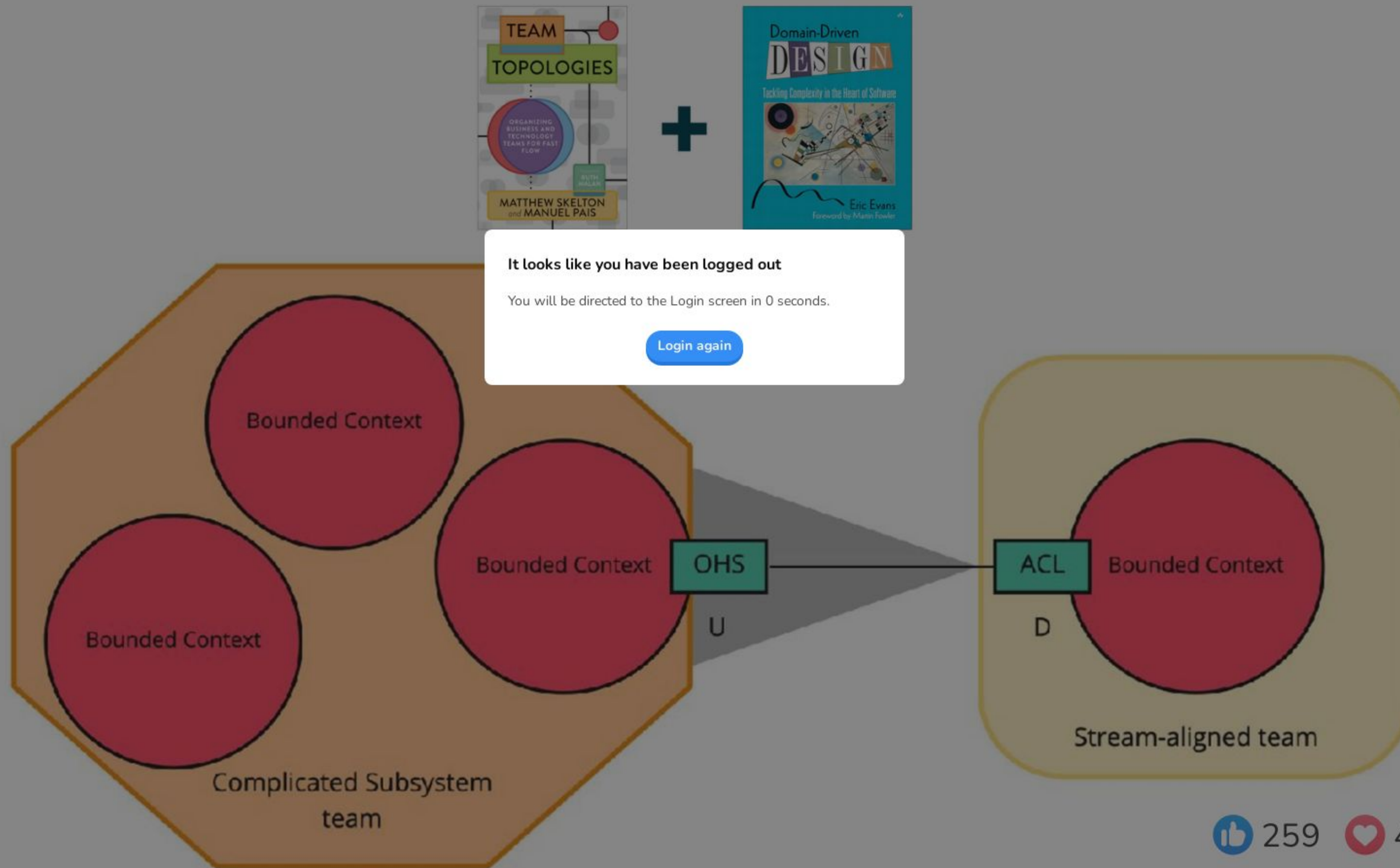
It looks like you have been logged out

You will be directed to the Login screen in 0 seconds.

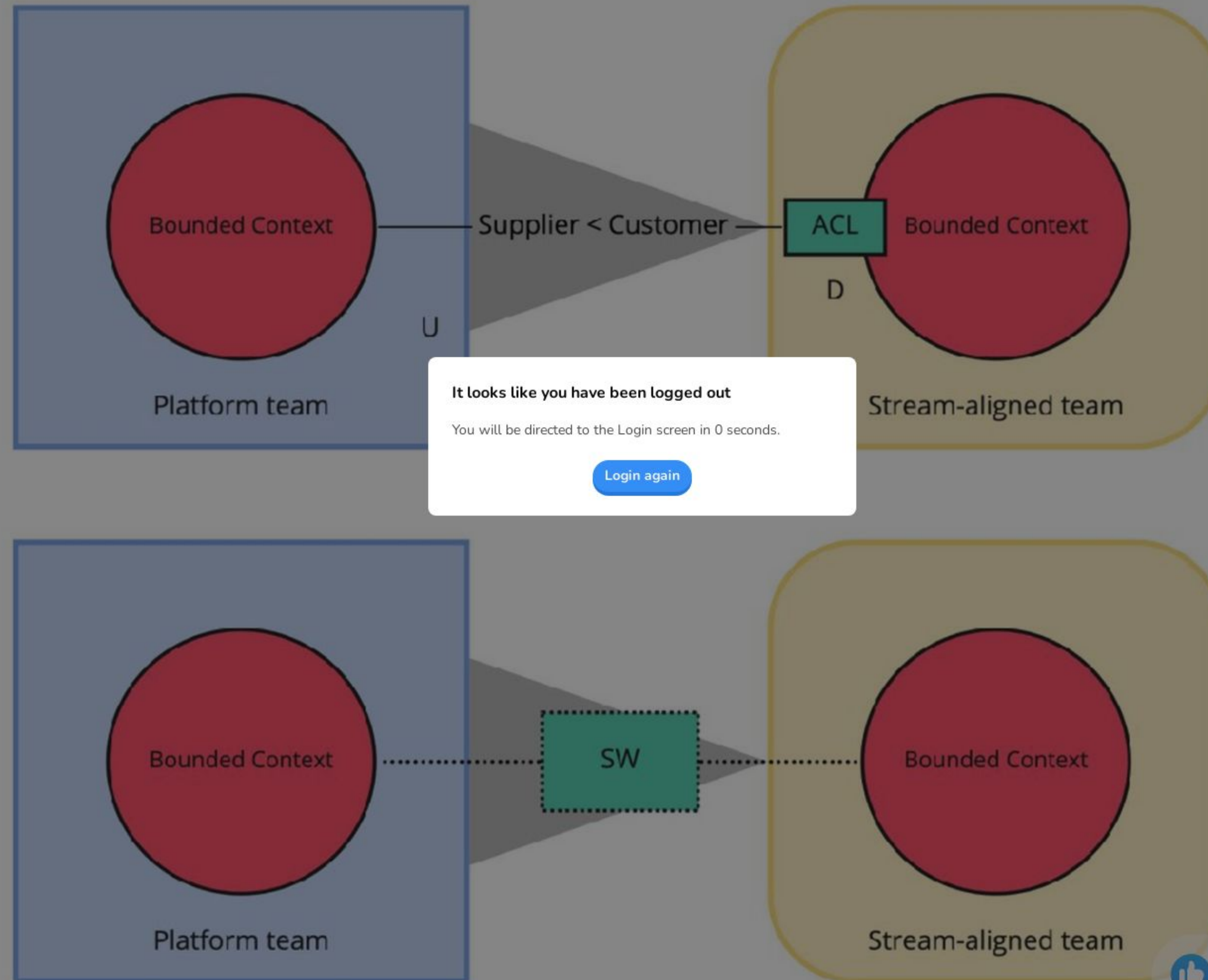
Login again



Let's dig deeper into this relationship with DDD's Context Maps



Other examples



It looks like you have been logged out

You will be directed to the Login screen in 0 seconds.

Login again



259



433



78

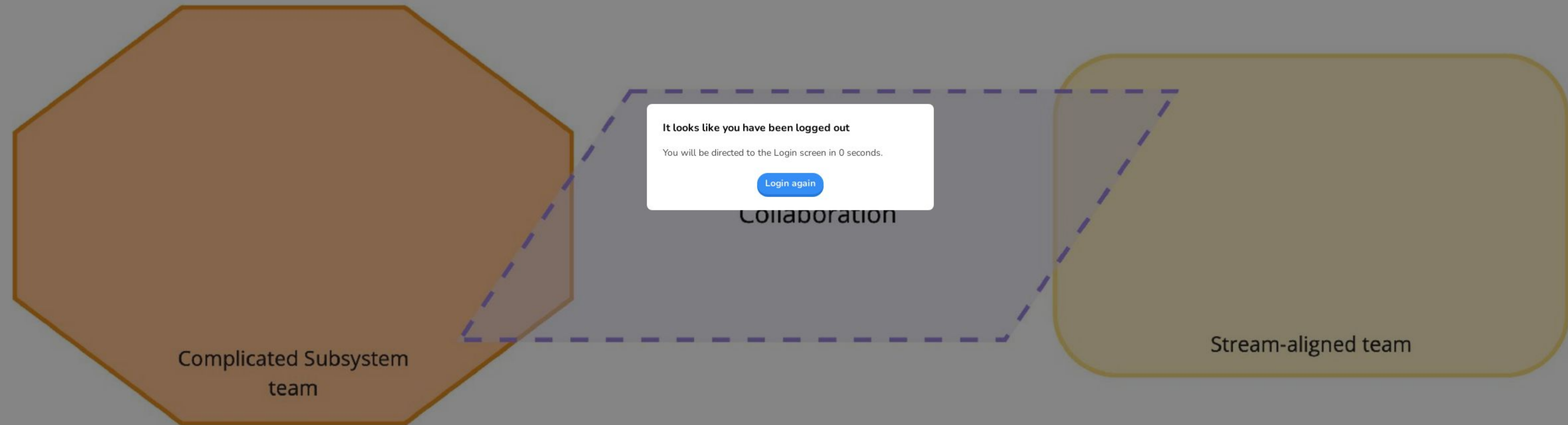


72

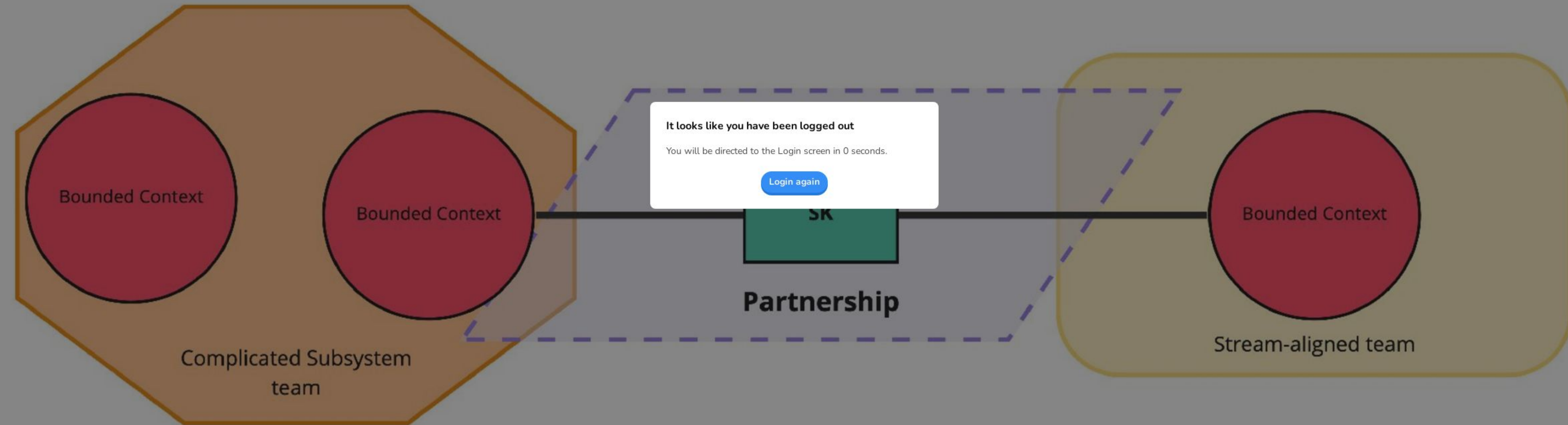


25

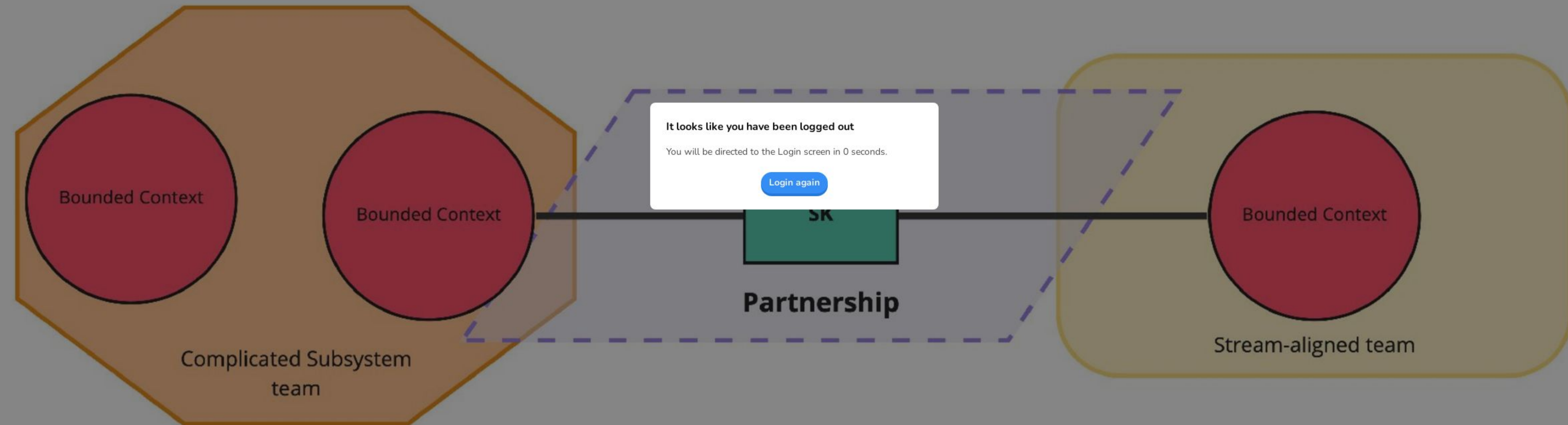
How does a Team Topologies collaboration look like in detail?



Let's drill down into the collaboration and detect something really ugly



YOU DON'T WANT THIS!



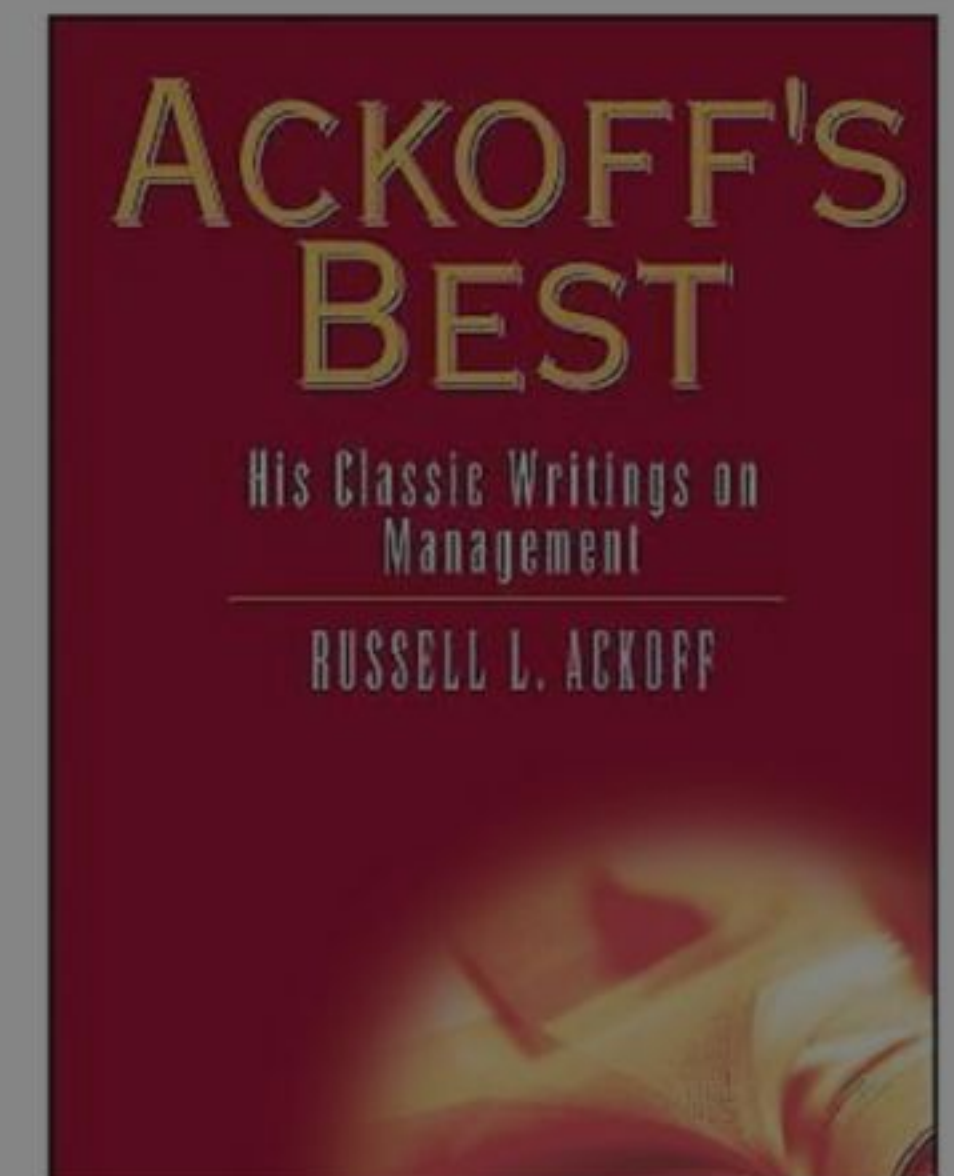
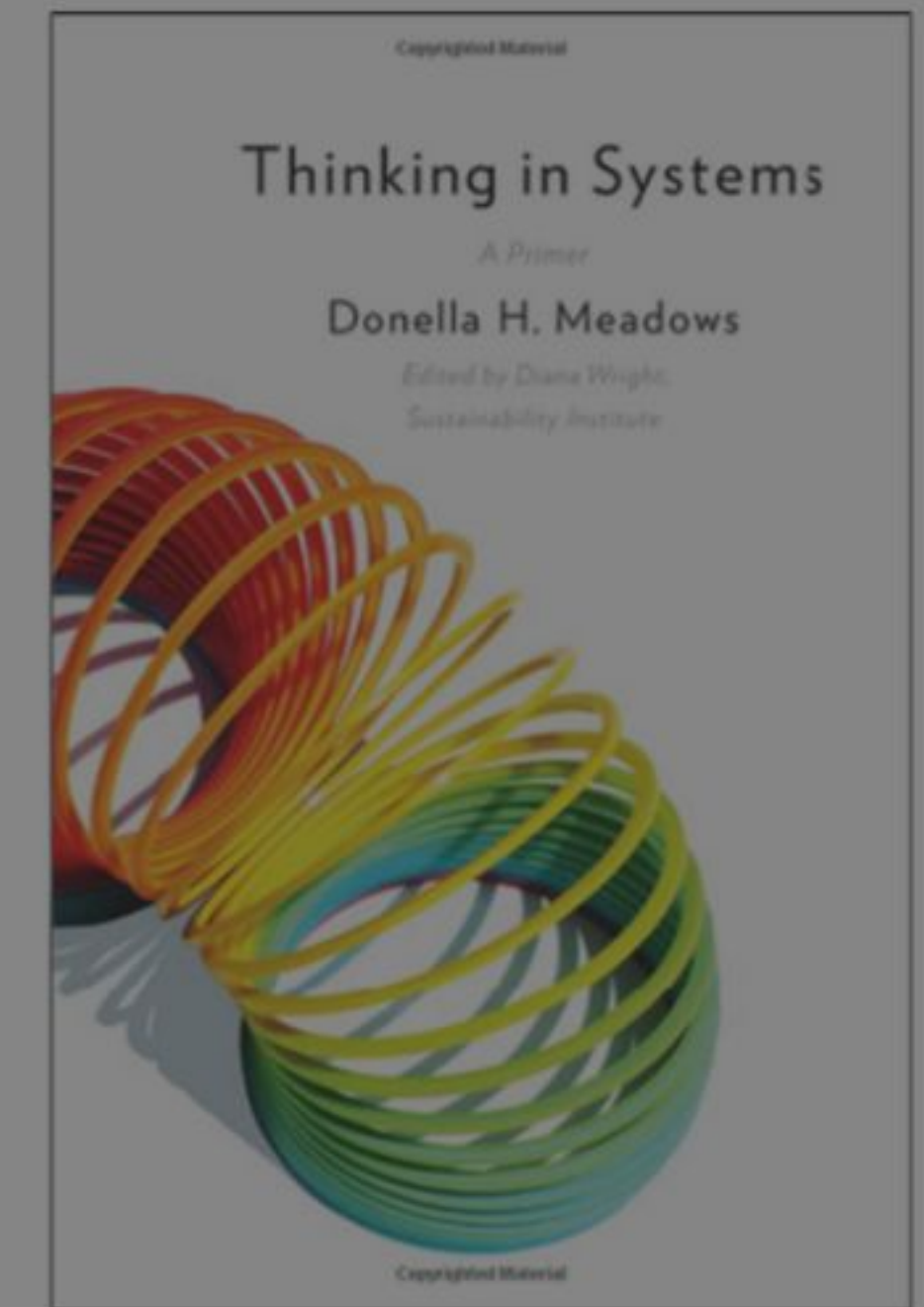
From a **Systems Thinking** perspective which aims at understanding a system as a whole **combining Team Topologies with Context Maps** makes sense

- Team Topologies give us a great starting point by focussing on teams and their relationships
- Context Maps allow us to dig deeper into the interactions of those relationships and add another perspective with their focus on Bounded Contexts
- Combining both allows us to really understand a system as a whole

It looks like you have been logged out

You will be directed to the Login screen in 0 seconds.

Login again



259



433



78

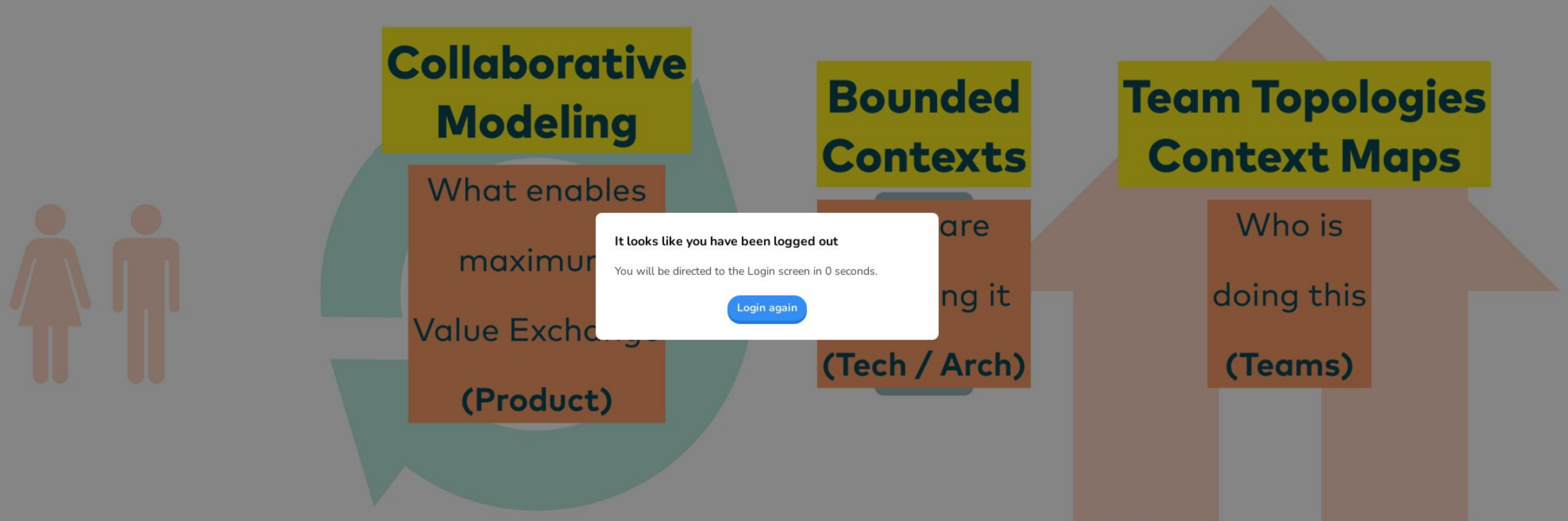


72



25

Value Exchange with Customer



Thanks!



Michael Plöd

Mastodon / Twitter: [@bitboss@mastodon.social](https://mastodon.social/@bitboss)

LinkedIn: <https://www.linkedin.com/in/michael-ploed/>

It looks like you have been logged out

You will be directed to the Login screen in 0 seconds.

Login again



Book Voucher: 7.99 instead of (min) 9.99
<http://leanpub.com/ddd-by-example/c/speakerdeck>

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim
+49 2173 3366-0

Ohlauer Str. 43
10999 Berlin

Ludwigstr. 180E
63067 Offenbach

Kreuzstr. 16
80331 München

Hermannstrasse 13
20095 Hamburg

Erftr. 15-17
50672 Köln

Königstorgraben 11
90402 Nürnberg

259 433 78 72 25