



Introduction to

Context Mapping









Hands On DOMAIN-DRIVEN DESIGN by example

Michael Plöd

Get my DDD book cheaper



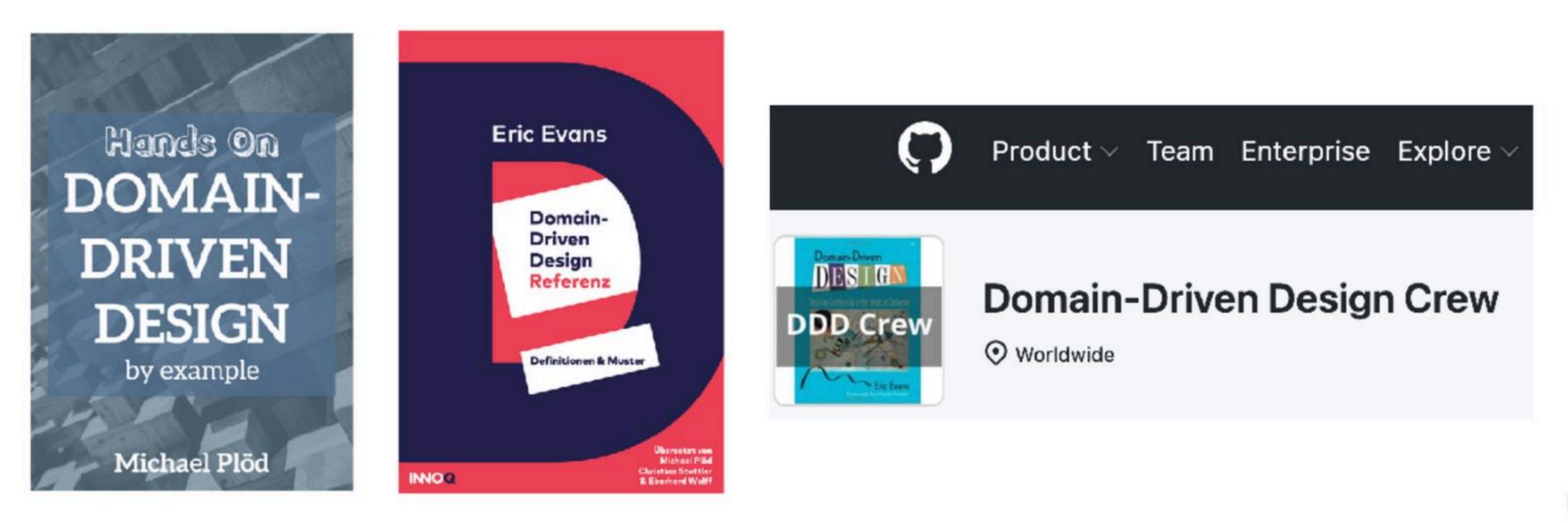
Book Voucher: 7.99 instead of (min) 9.99 http://leanpub.com/ddd-by-example/c/speakerdeck

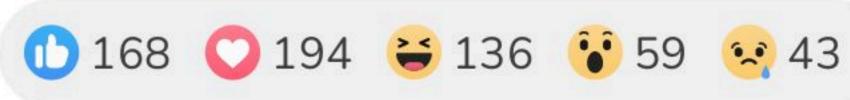


Michael Plöd

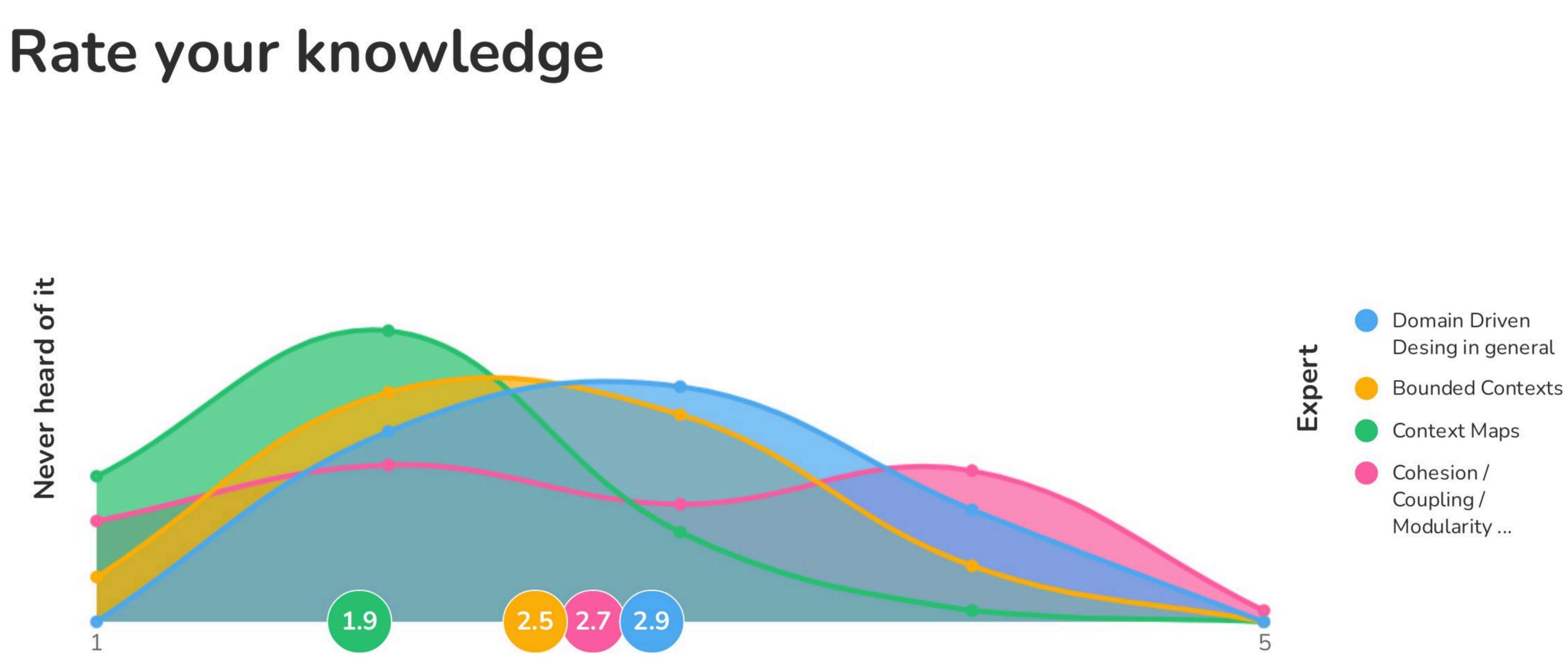
Fellow at INNOQ

- Current consulting and training activities:
- **Domain-Driven Design**
- **Team Topologies**
- Software Architecture (and often environment) transformation











"Good fences make good neighbors" **Robert Frost**















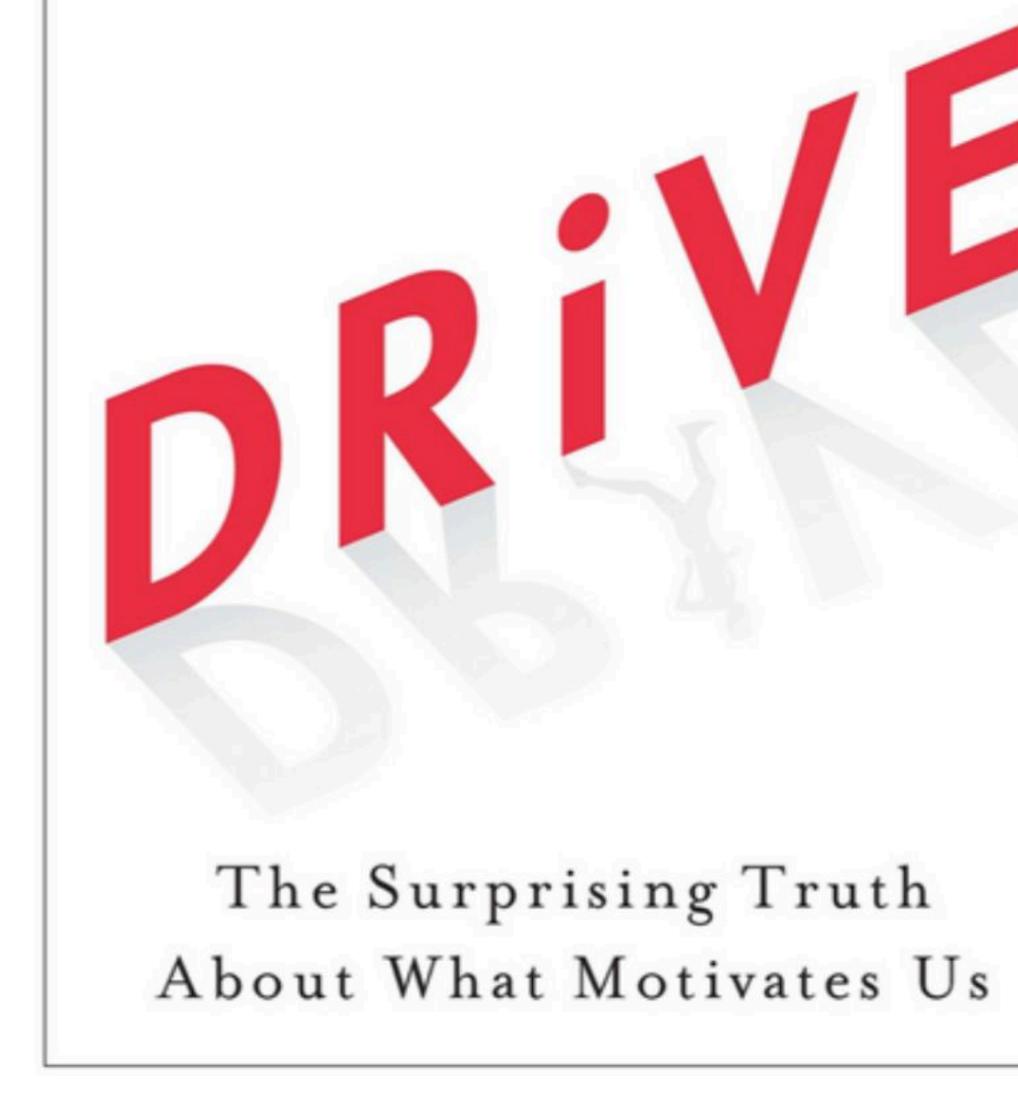


NEW YORK TIMES BESTSELLER

"Provocative and fascinating." — MALCOLM GLADWELL

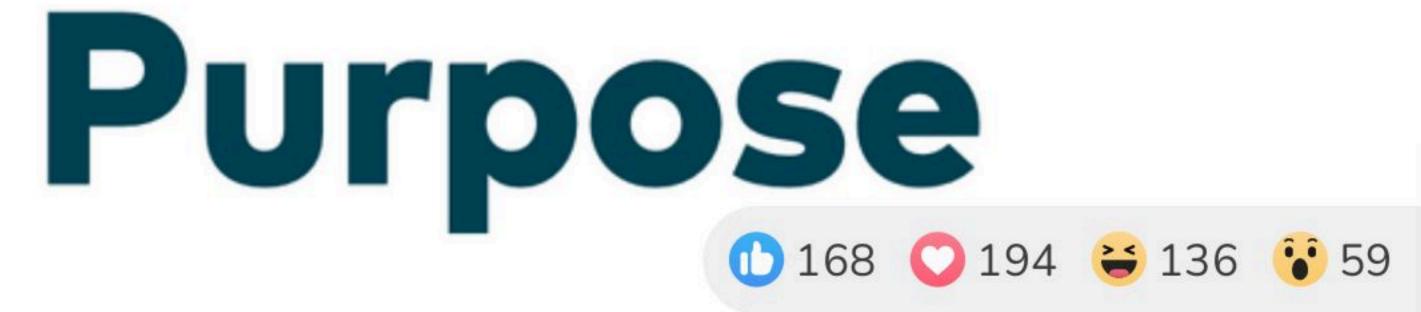
Daniel H. Pink

author of A Whole New Mind





Nastery





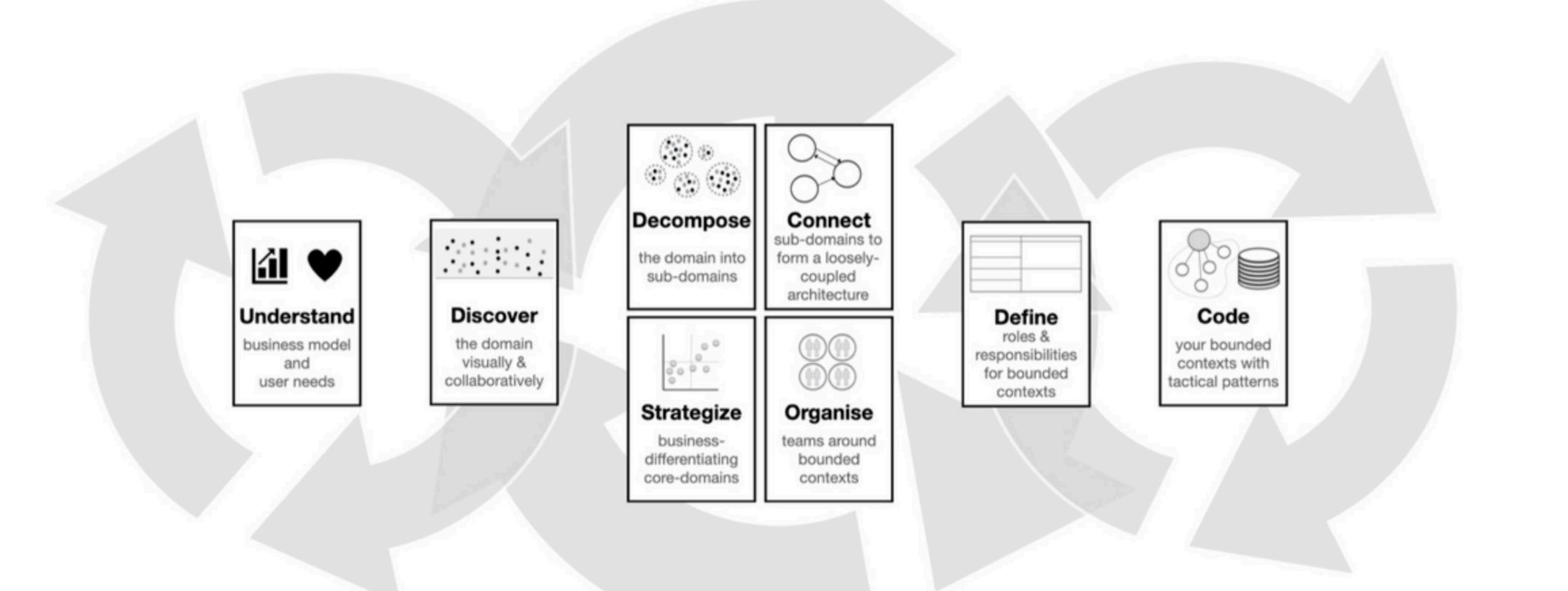


We need good boundaries in which teams can achieve Autonomy - Mastery - Purpose



Domain-Driven Design Starter Modelling Process

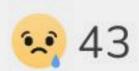
A starter process for beginners, not a rigid best-practice. DDD is continuous, evolutionary, and iterative design.



https://github.com/ddd-crew/ddd-starter-modelling-proce 🗘 168 🔘 194 😝 136 😯 59 😣 43

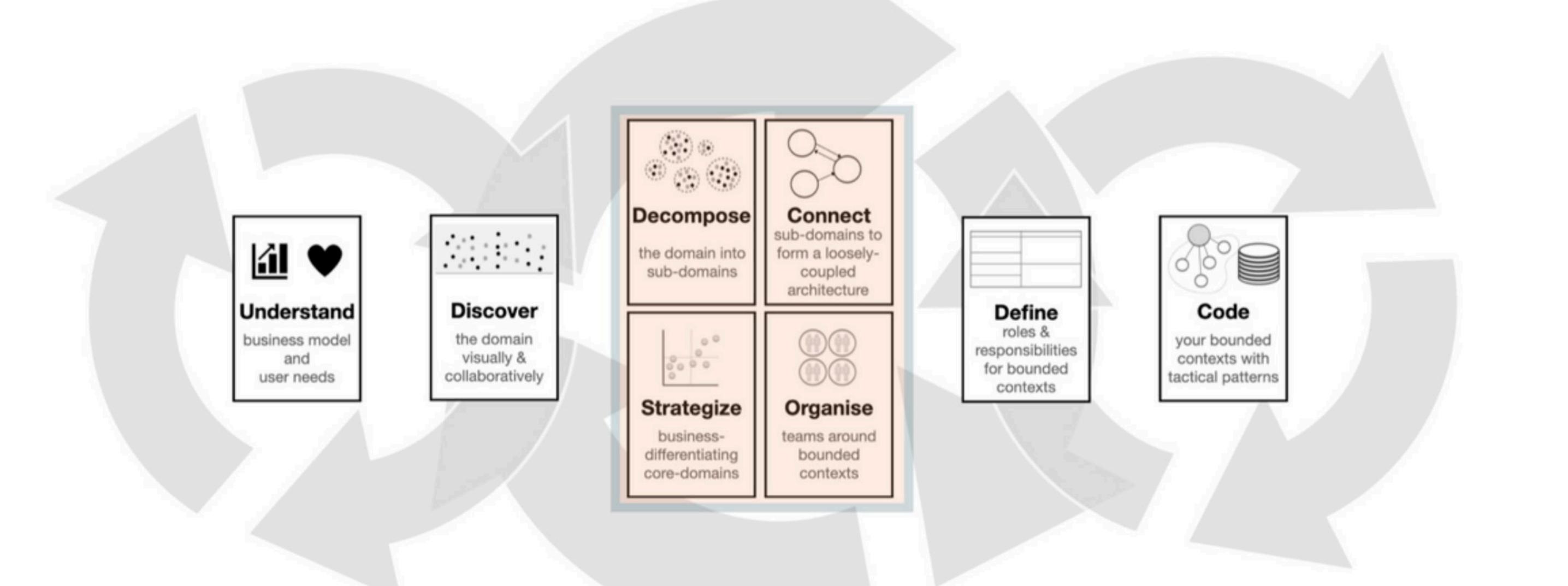






Domain-Driven Design Starter Modelling Process

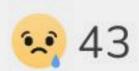
A starter process for beginners, not a rigid best-practice. DDD is continuous, evolutionary, and iterative design.



https://github.com/ddd-crew/ddd-starter-modelling-proce 🗘 168 🗘 194 😝 136 😯 59 😣 43

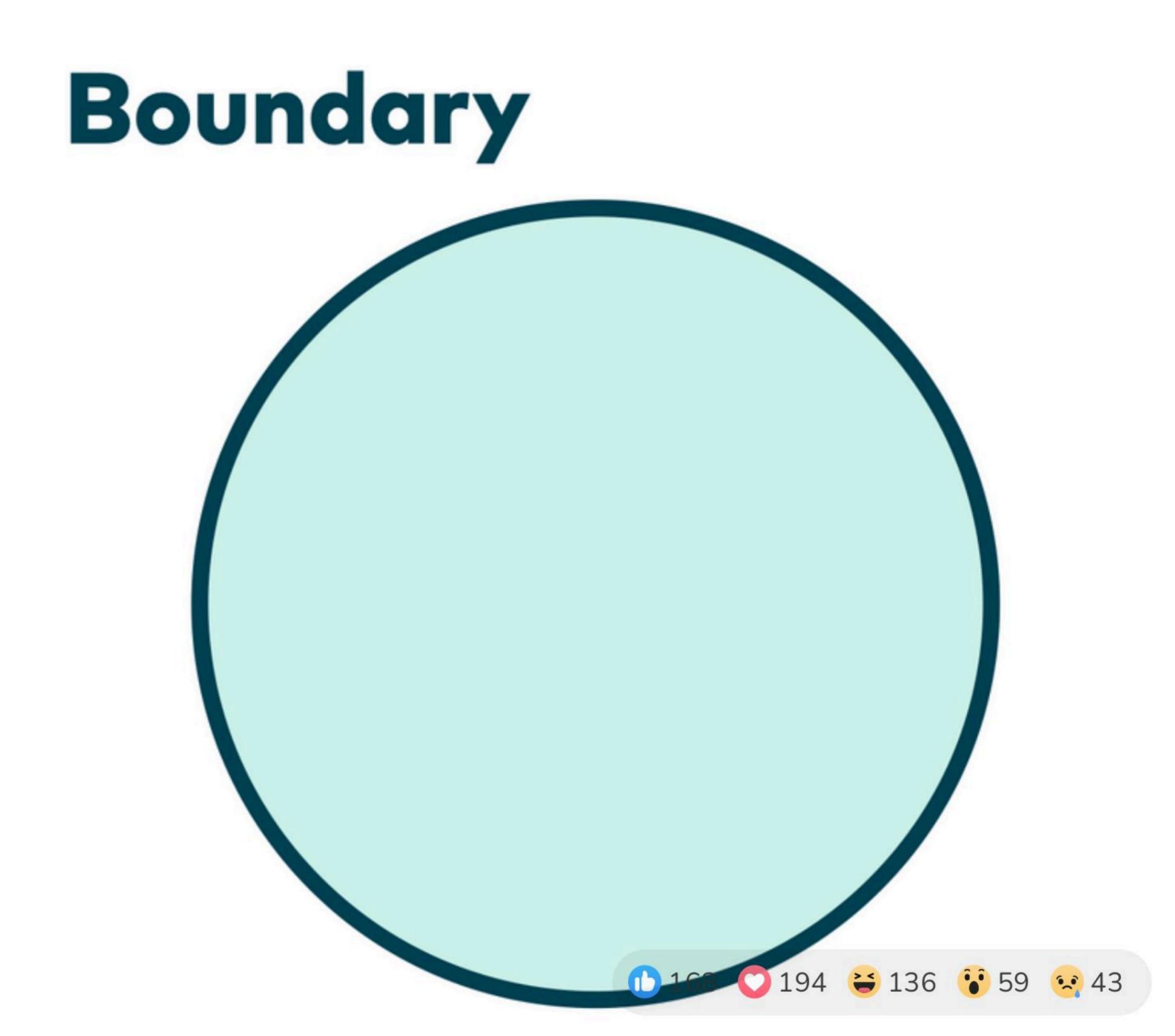






A Bounded Context is a boundary for a model expressed in a consistent (ubiquitous) language tailored around a specific purpose **Bounded Context**





Boundary for a model

Business Rules

Decisions

Policies



Language

Terminology

Definitions

Meaning



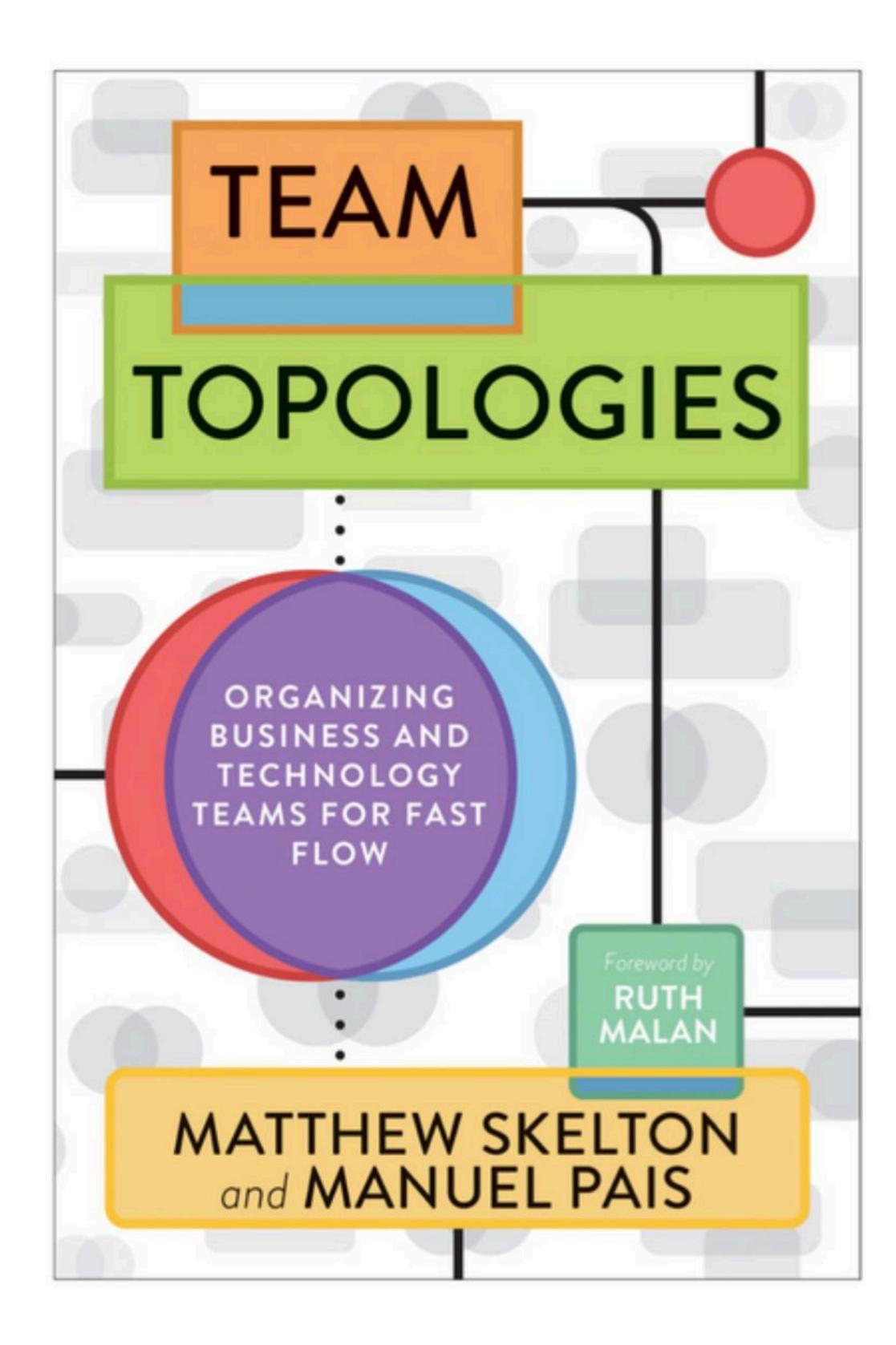


Language

Rules

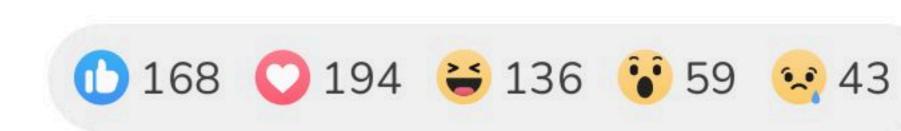
Specific Model





The Bounded Context is a

team first boundary





"Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure." Melvin Conway

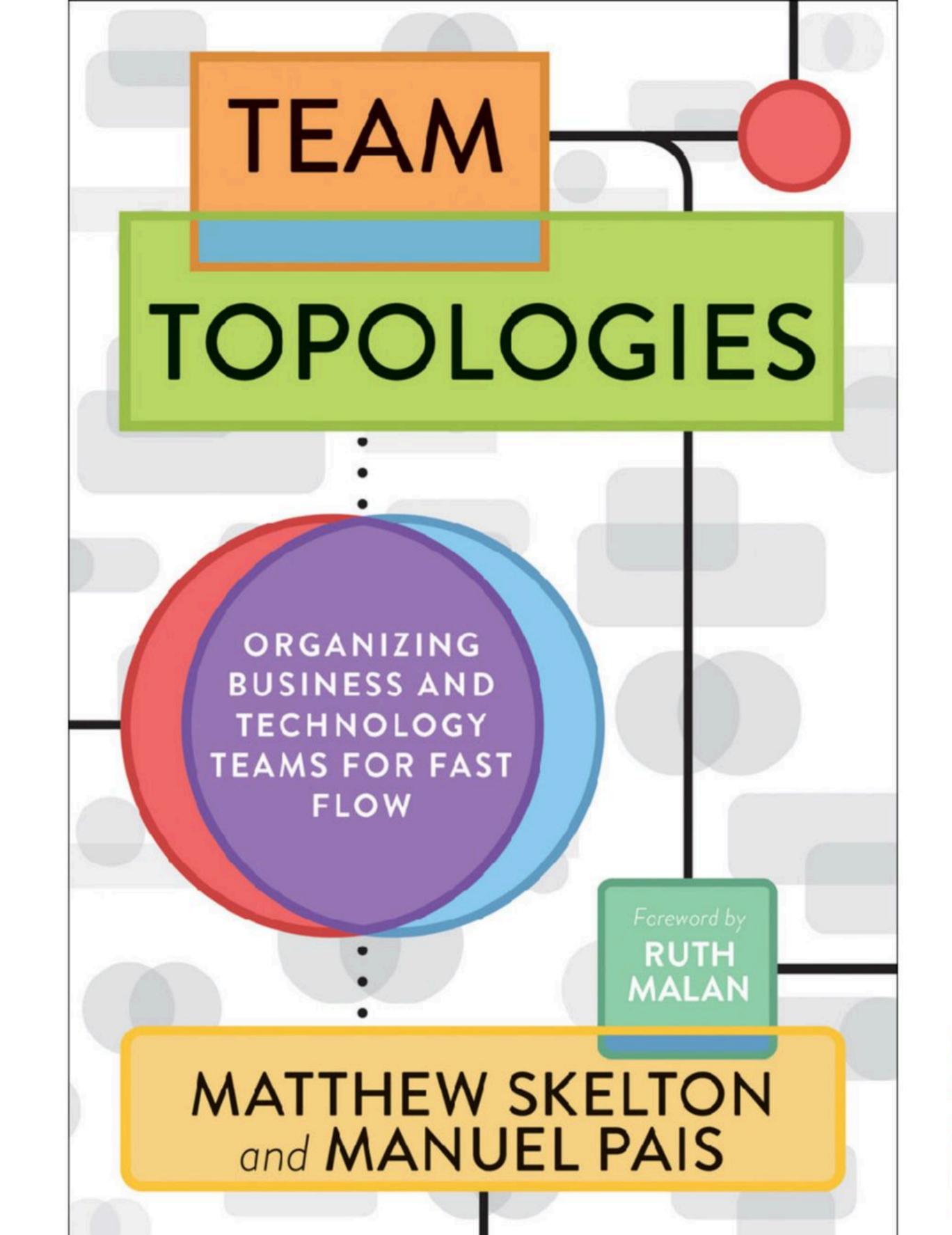










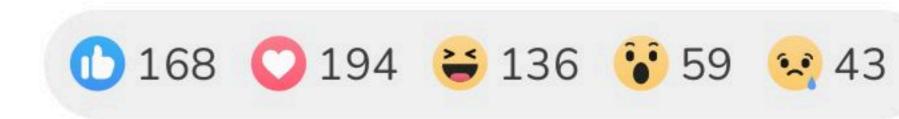


"An architect should be thinking:

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?"

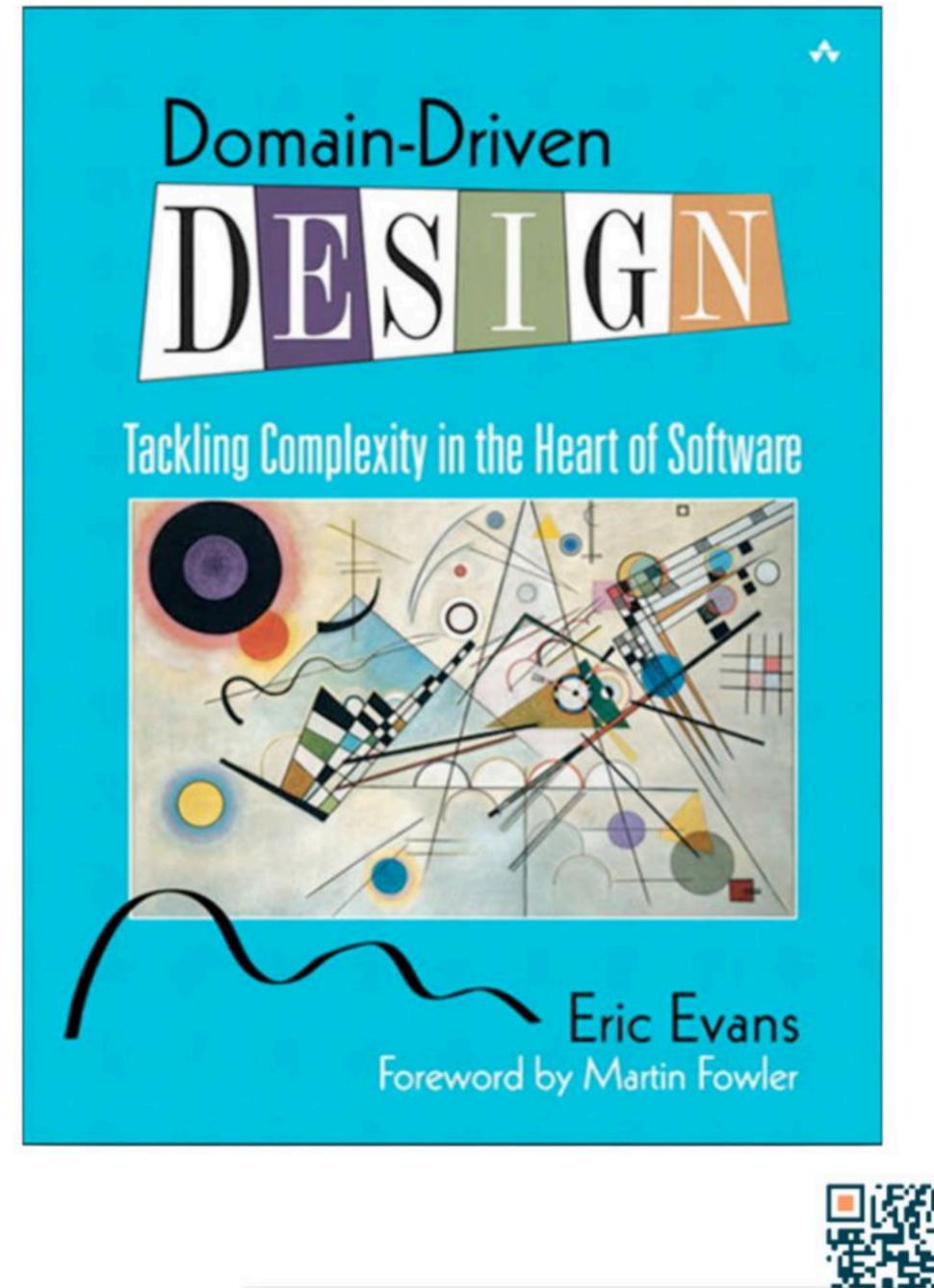




Strategic Domain Driven Design has a technique to visualize relationships between Bounded Contexts and teams: CONTEXT MAPS



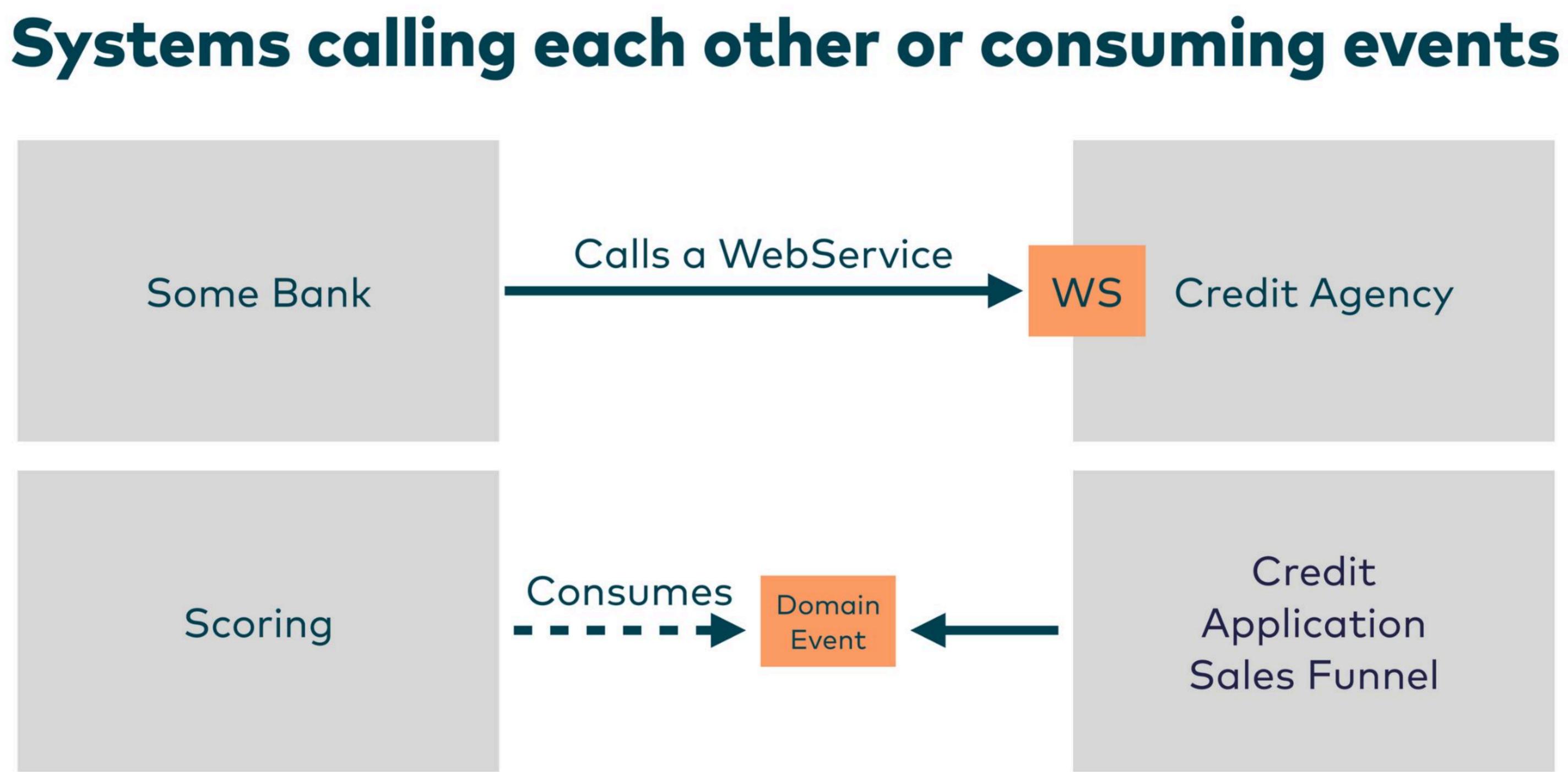


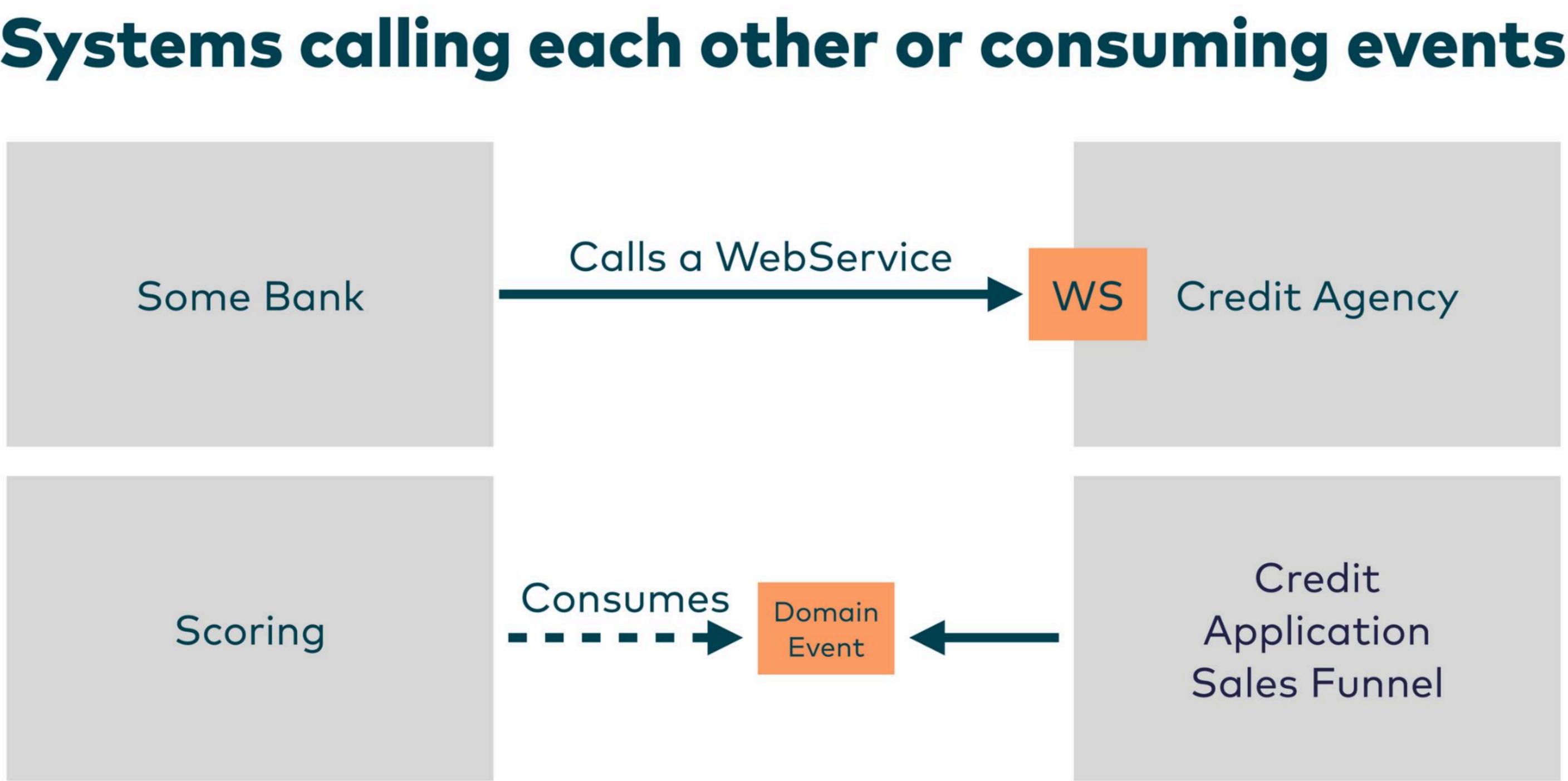


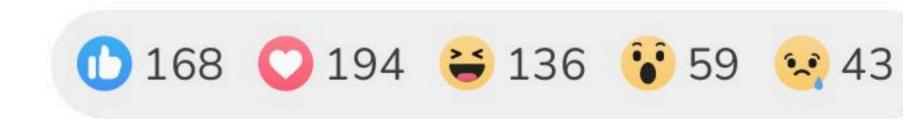


Which kinds of relationships between systems and teams do context maps address?



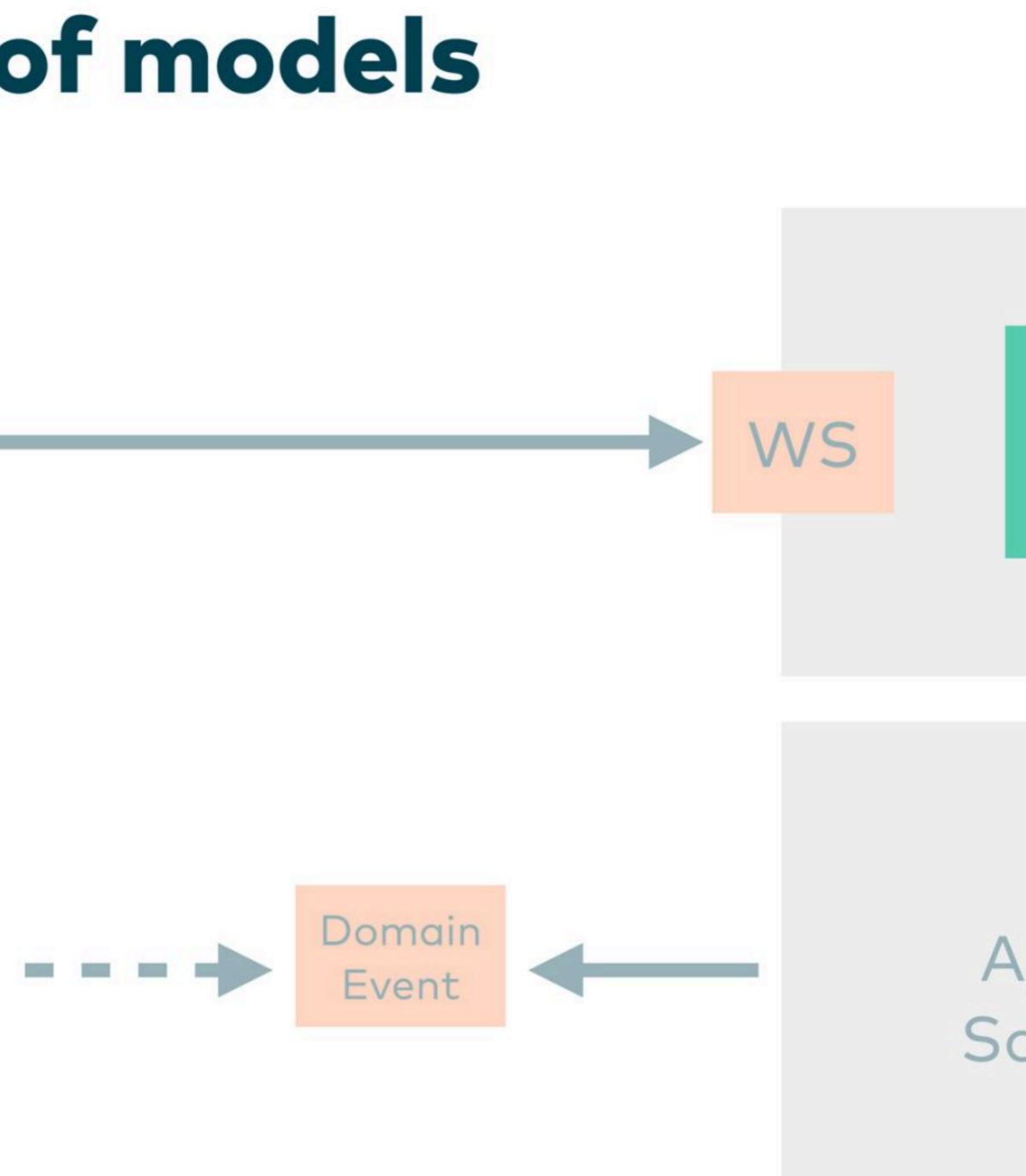


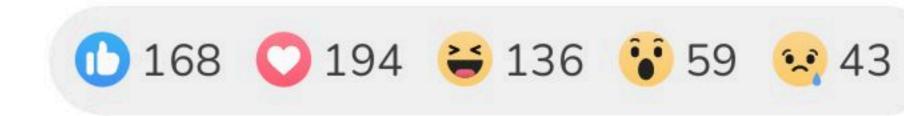




Some Bank

Scoring



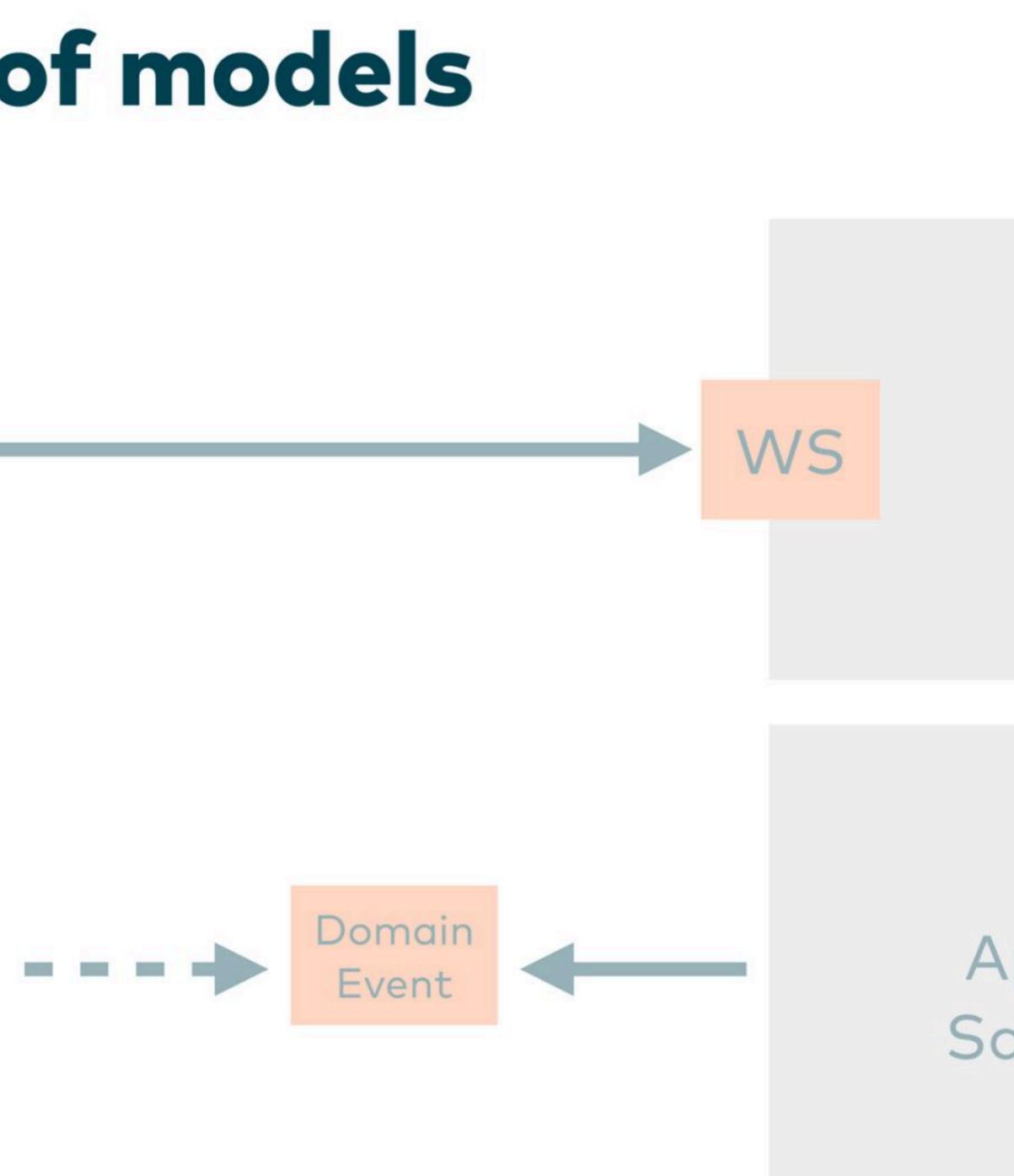


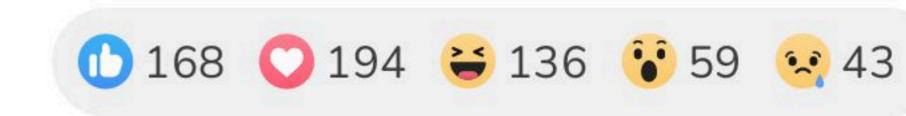
Credit Rating Result

Credit Application Sales Funnel







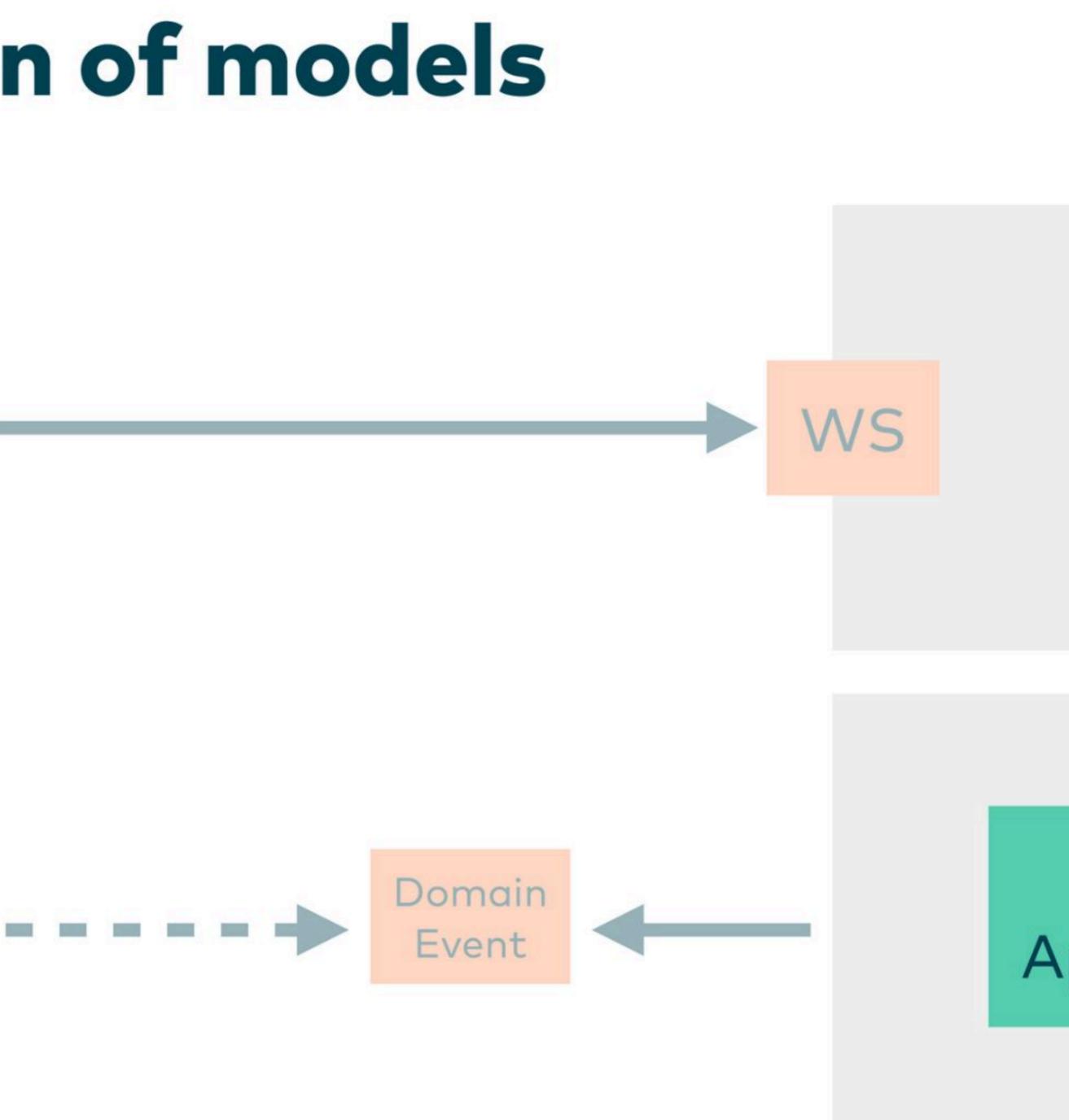


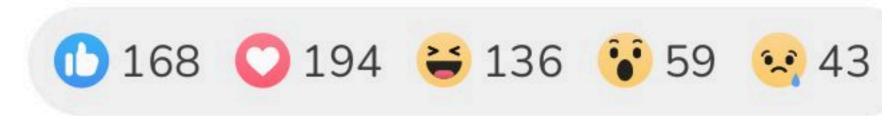
Schufa

Credit Application Sales Funnel



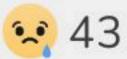






Schufa

Credit Application











Credit Application

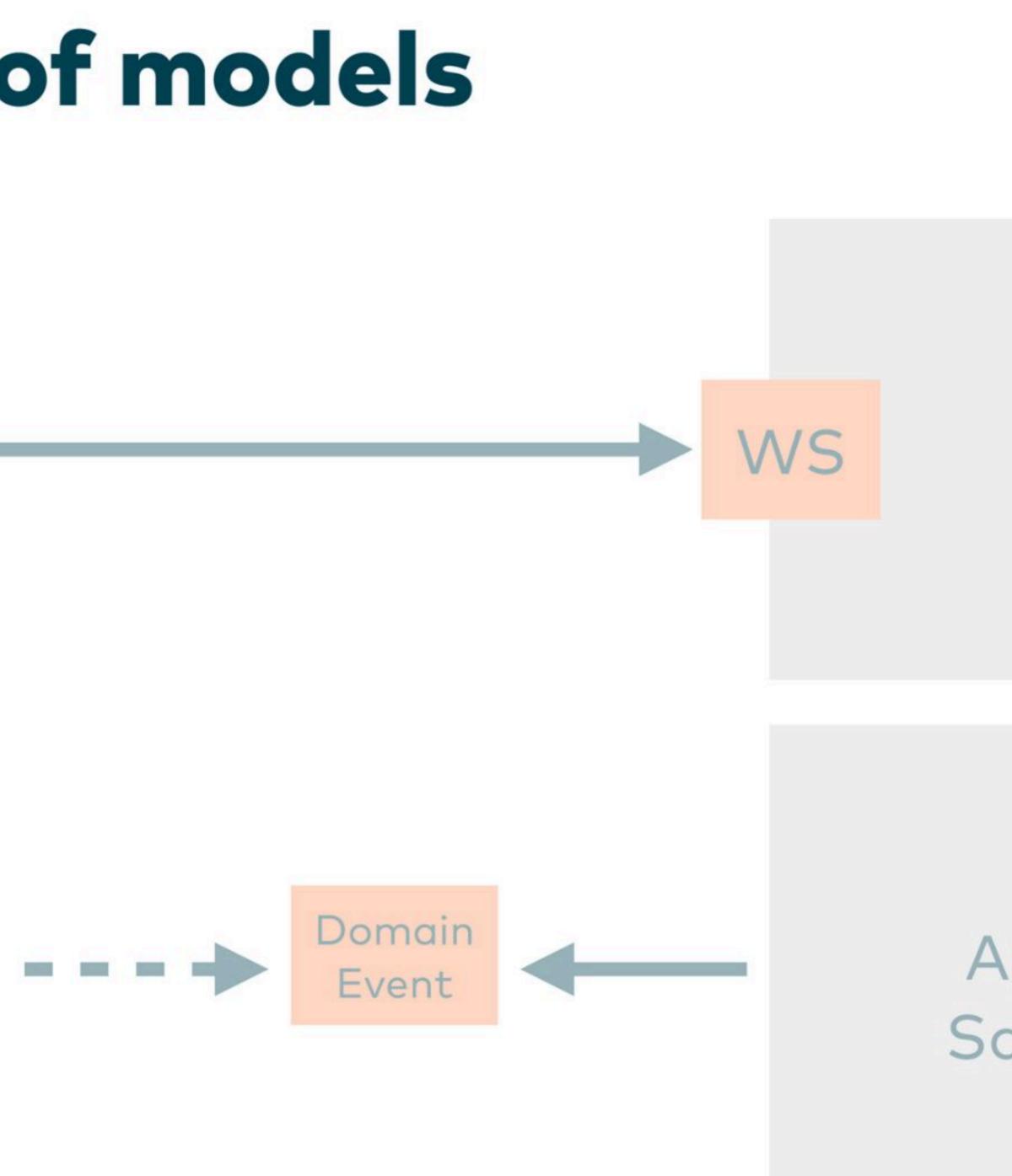
Schufa

Credit Application Sales Funnel





Credit Application





Schufa

Credit Application Sales Funnel

Teams communicating with each other

Some Bank

Ok no problem, just send us the documentation.

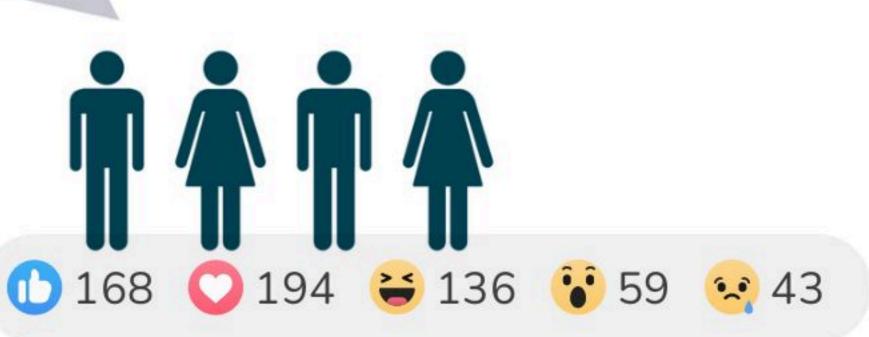


WS

We will adjust the interface and the underlying model with the next release.



Credit Agency



Teams communicating with each other

We have neither the budget, nor the capacity to implement that change.



Credit Agency

We will deprecate the domain event and replace it with a new one.





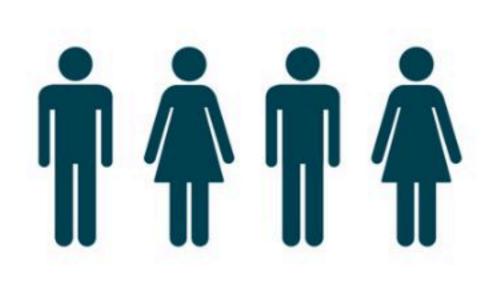
Credit Application Sales Funnel

168 \bigcirc 194 \rightleftharpoons 136 \bigcirc 59 \backsim 43

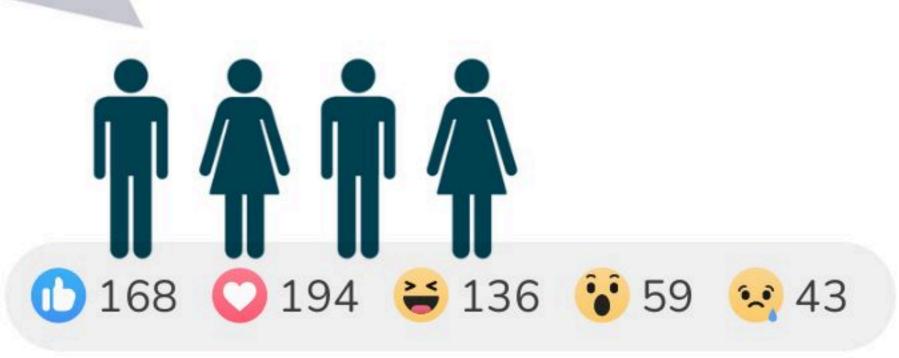


Actions of one team have an impact on others

Some Bank



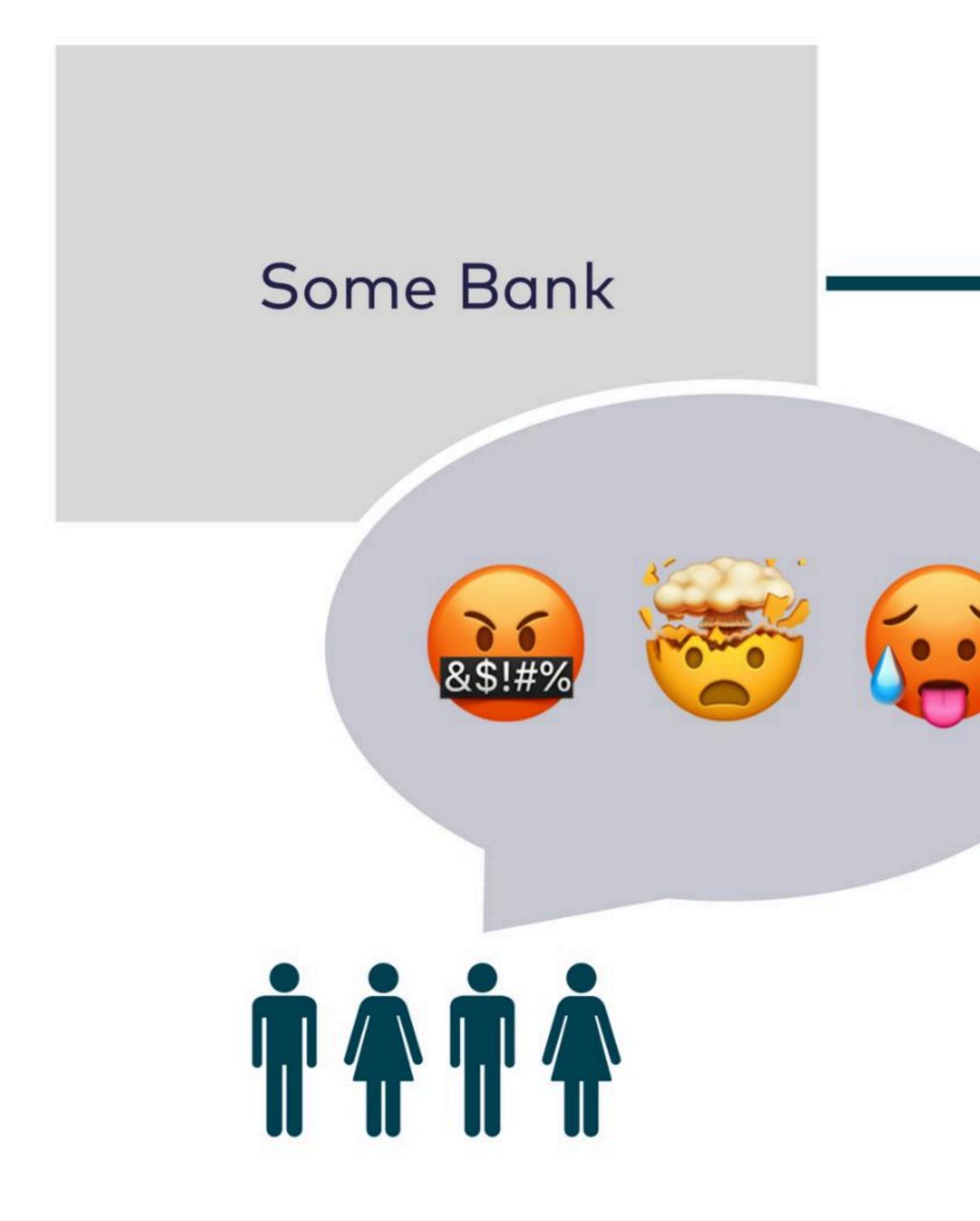
We will replace the WebService with **RESTful Resources next** week.





REST Credit Agency

Actions of one team have an impact on others



We will replace the WebService with **RESTful Resources next** week.





REST Credit Agency

168 🔘 194 😆 136 😯 59 😣 43

Dependencies between teams

Mutually Dependent

Team Dependencies

- 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

Mutually Dependent

Team Dependencies



Free

- 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.
- Changes in one bounded context do not influence success or failure in other bounded contexts.
- There is, therefore, no organizational or technical link of any kind between the teams.







Dependencies between teams

Mut Depe

Team Dependencies

Fr

Upstre

ually ndent	 Two software artifacts or syster bounded contexts need to be detto be successful and work. There is often a close, reciprocadata and functions between the
ee	 Changes in one bounded contex success or failure in other bound There is, therefore, no organizate link of any kind between the tec
eam/ stream	 An upstream context will influe counterpart while the opposite This might apply to code but als factors such as schedule or resp external requests.



ems in two lelivered together

al link between ne two systems.

xt do not influence ded contexts.

itional or technical ams.

ence the downstream might not be true.

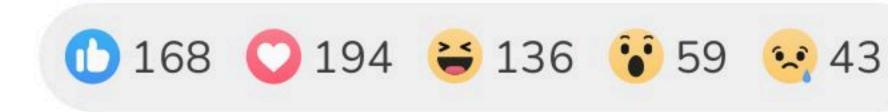
lso on less technical ponsiveness to



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

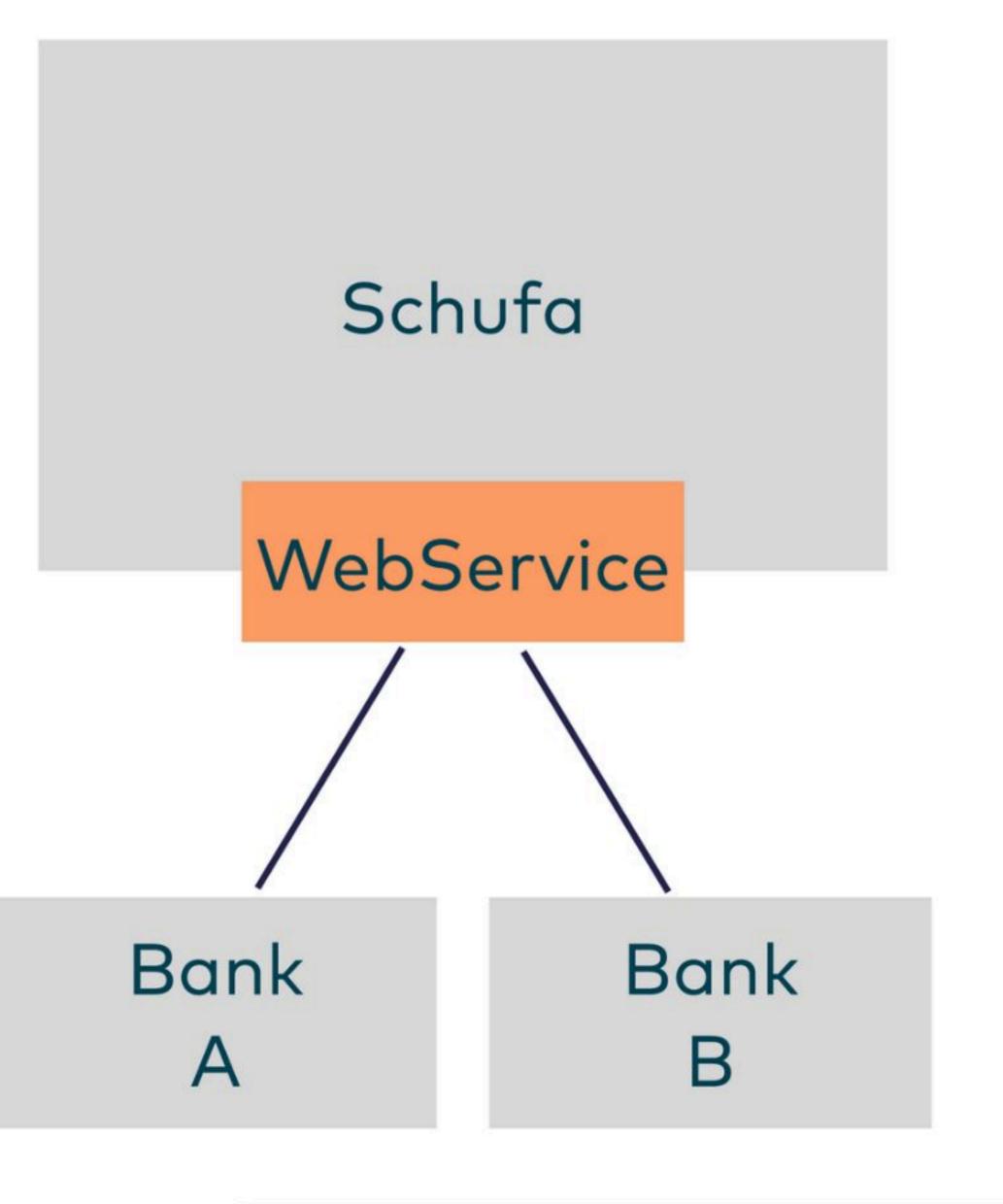


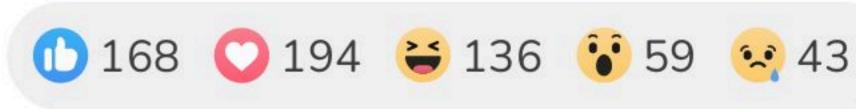
Open-host Service

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





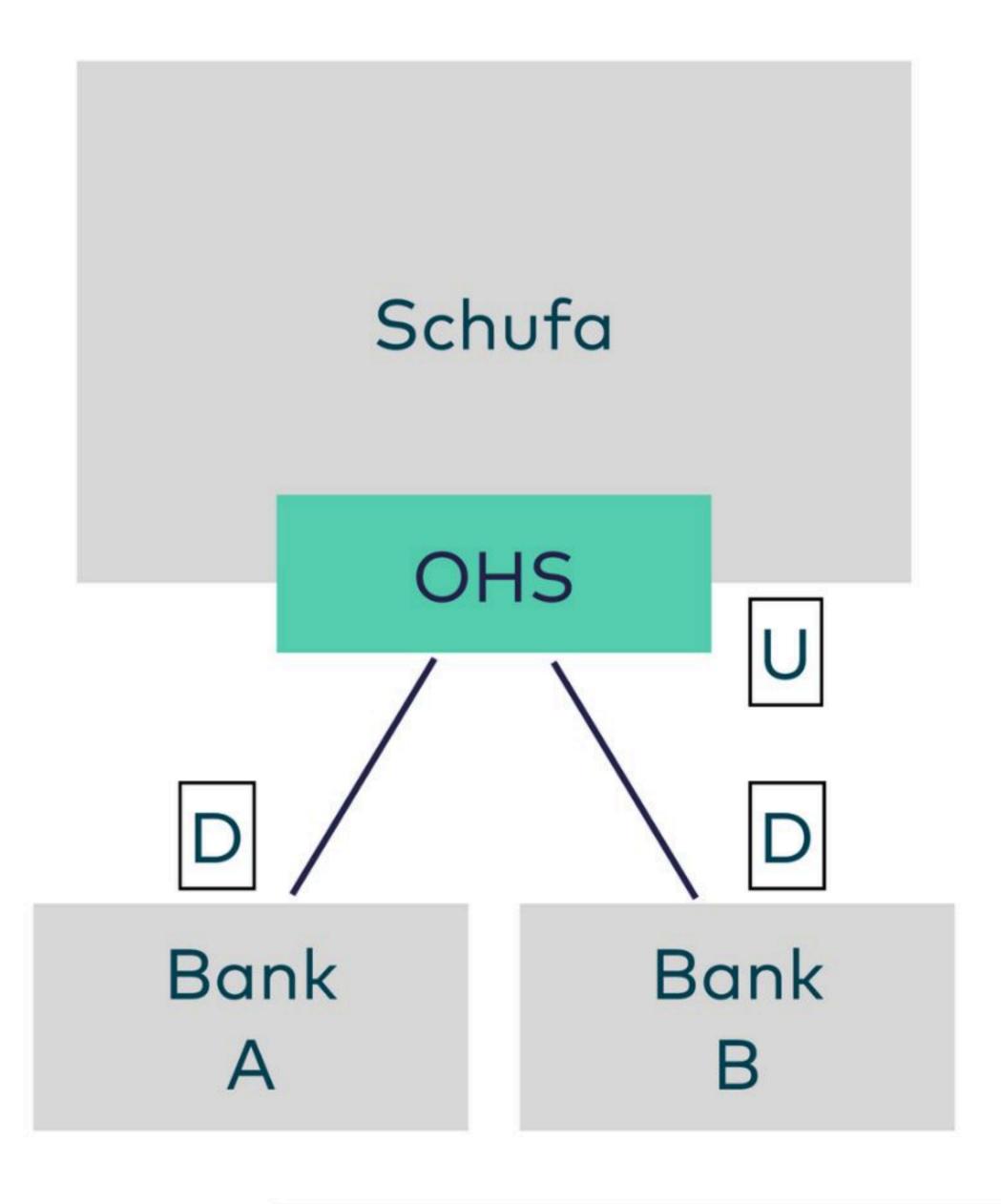


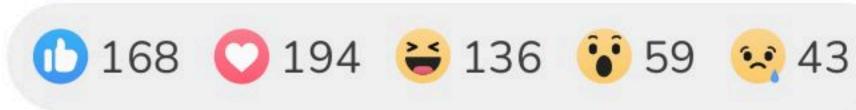
Open-host Service

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





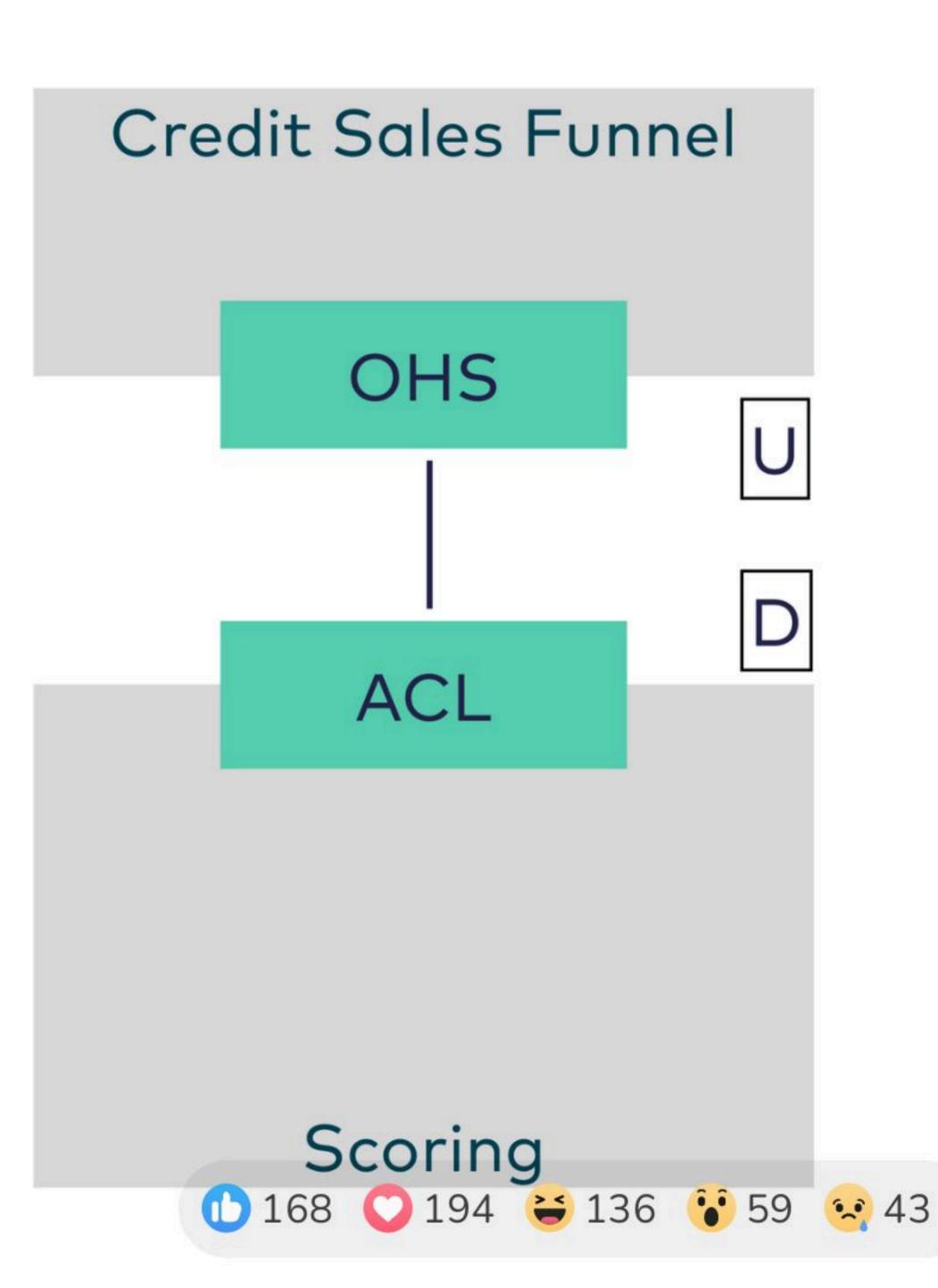


Anticorruption Layer

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



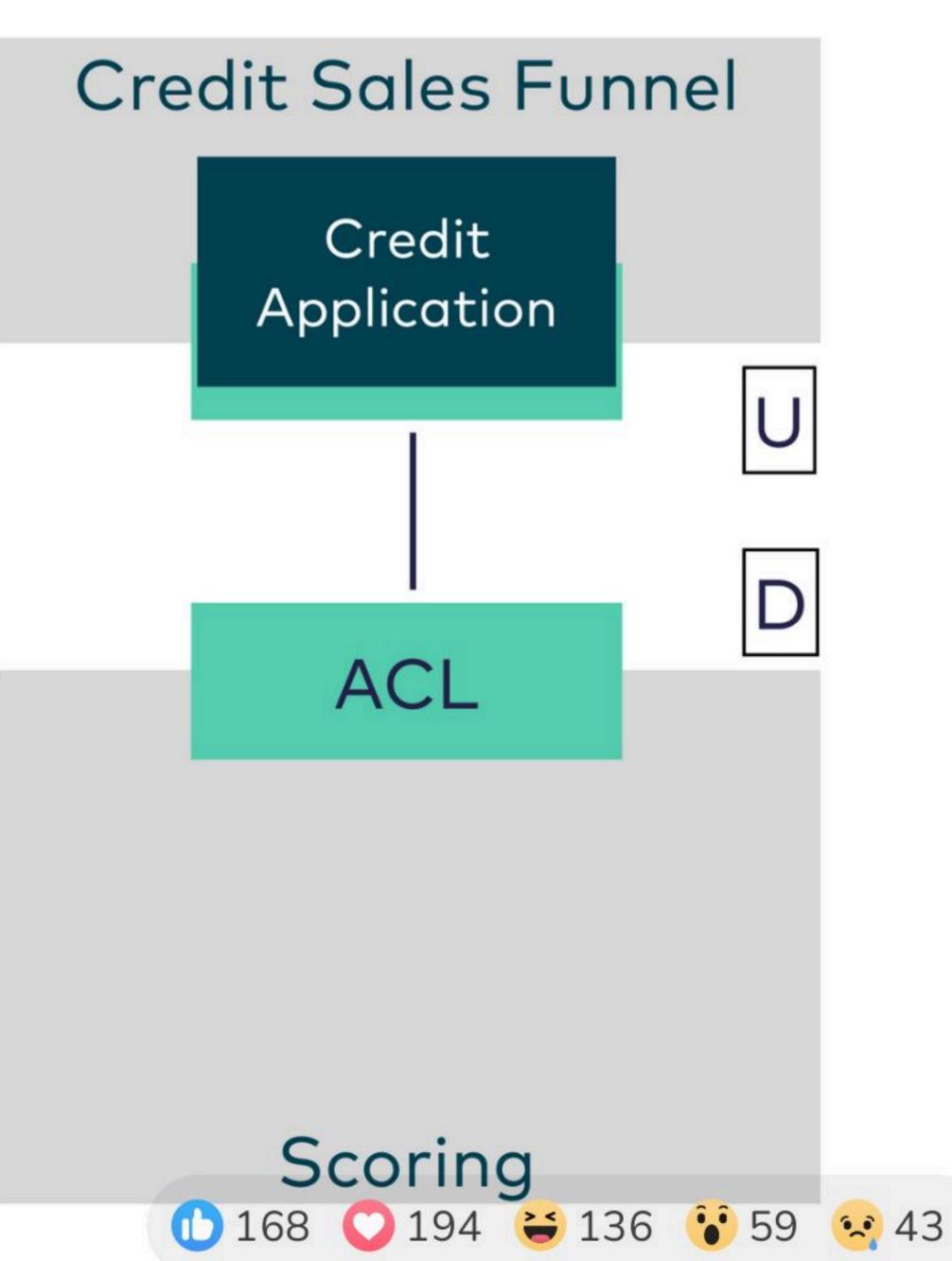


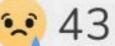
Anticorruption Layer

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





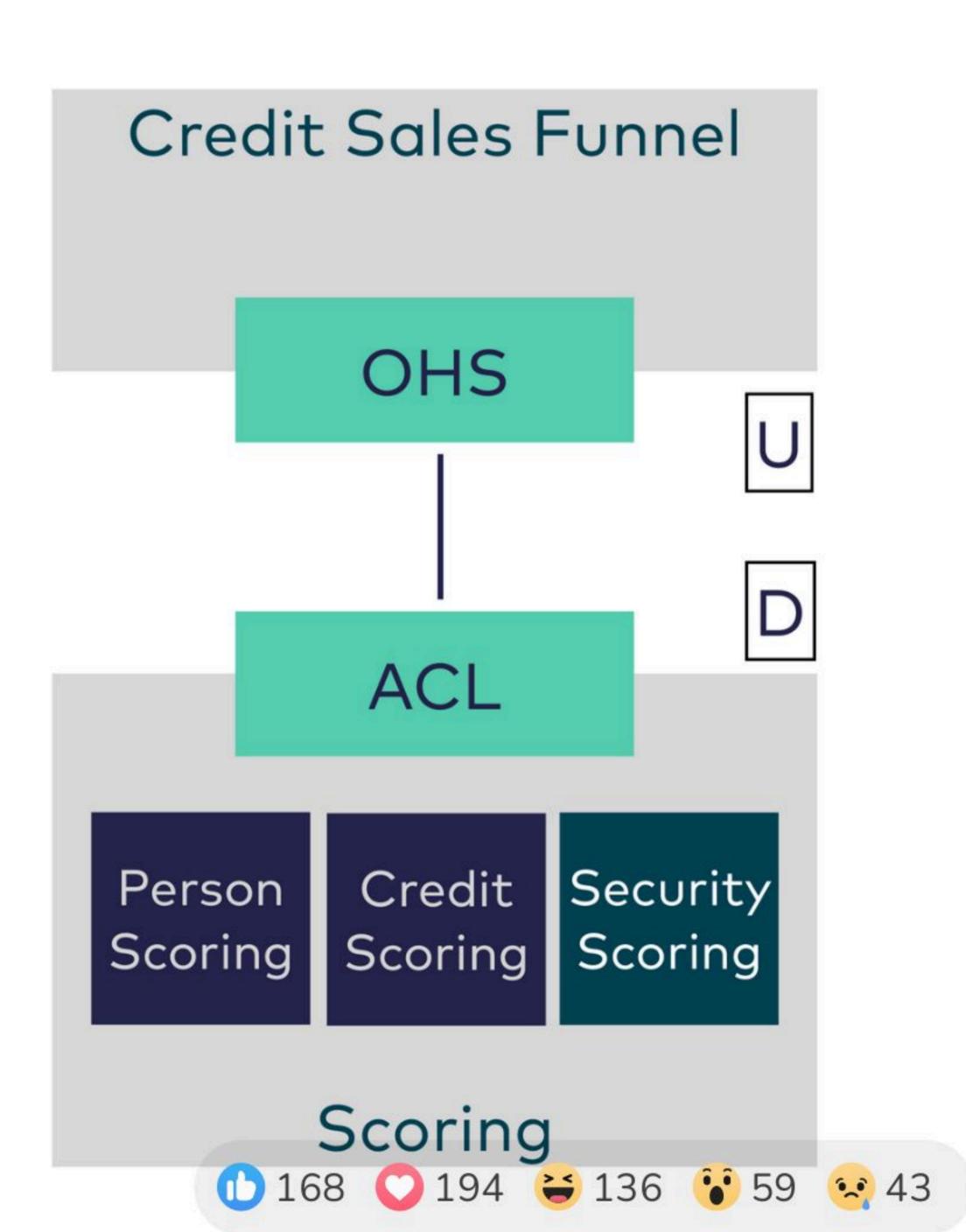


Anticorruption Layer

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

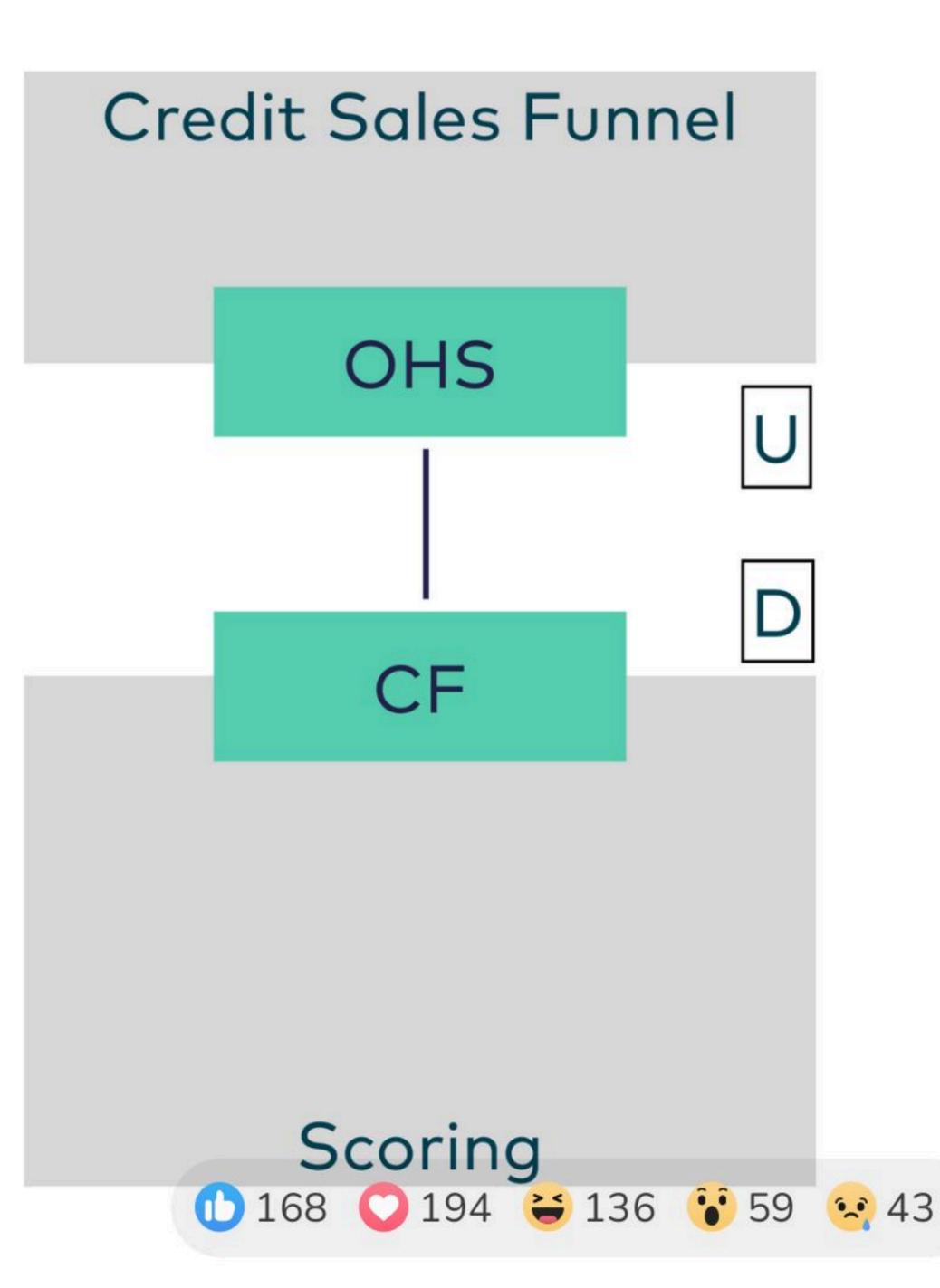






The Conformist slavishly adheres to the upstream model

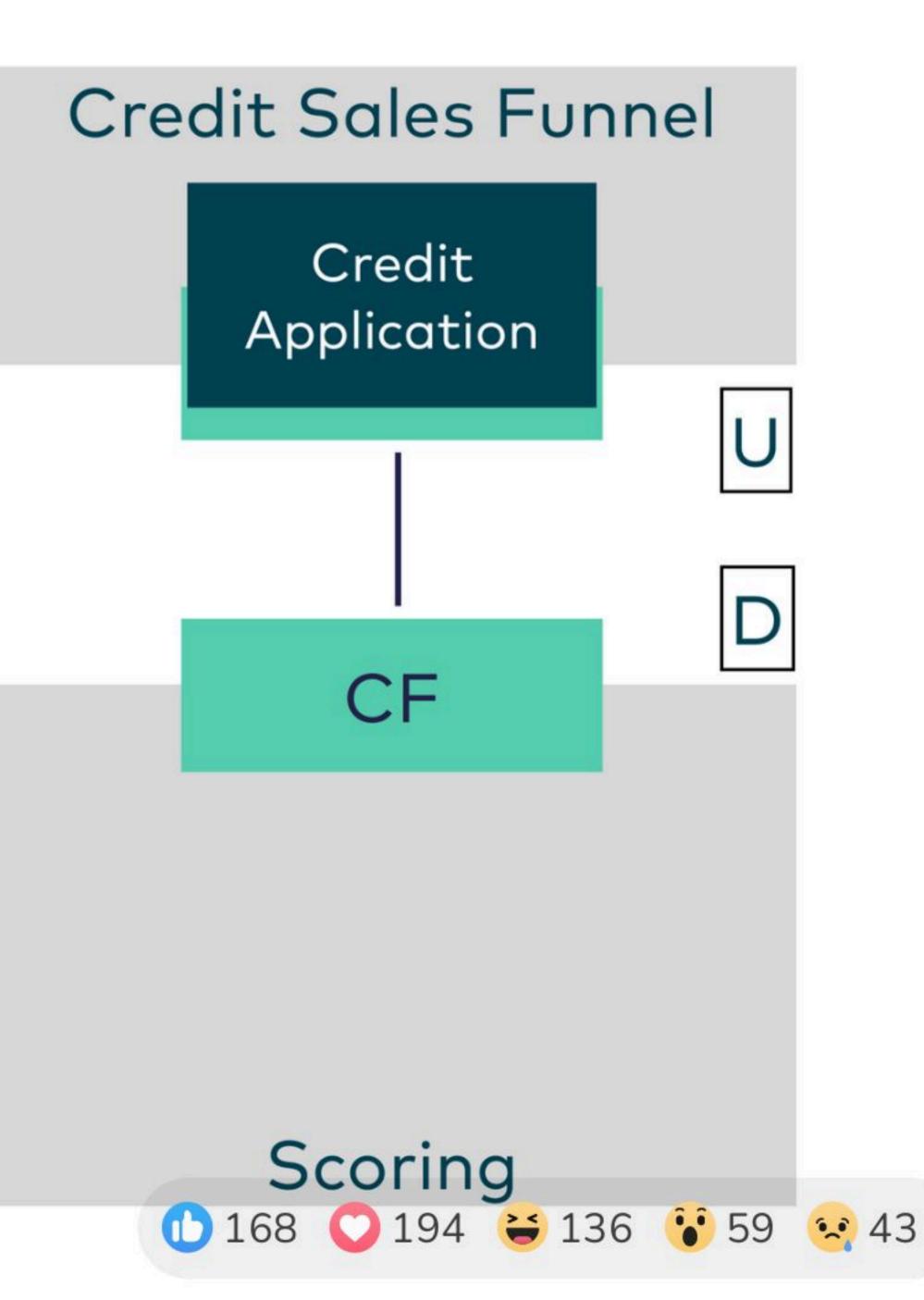
- There is no model-to-model transformation
- Motivation: Simplicity, contracts, force or delight (for the upstream model)
- The team implementing a Conformist is always downstream





The Conformist slavishly adheres to the upstream model

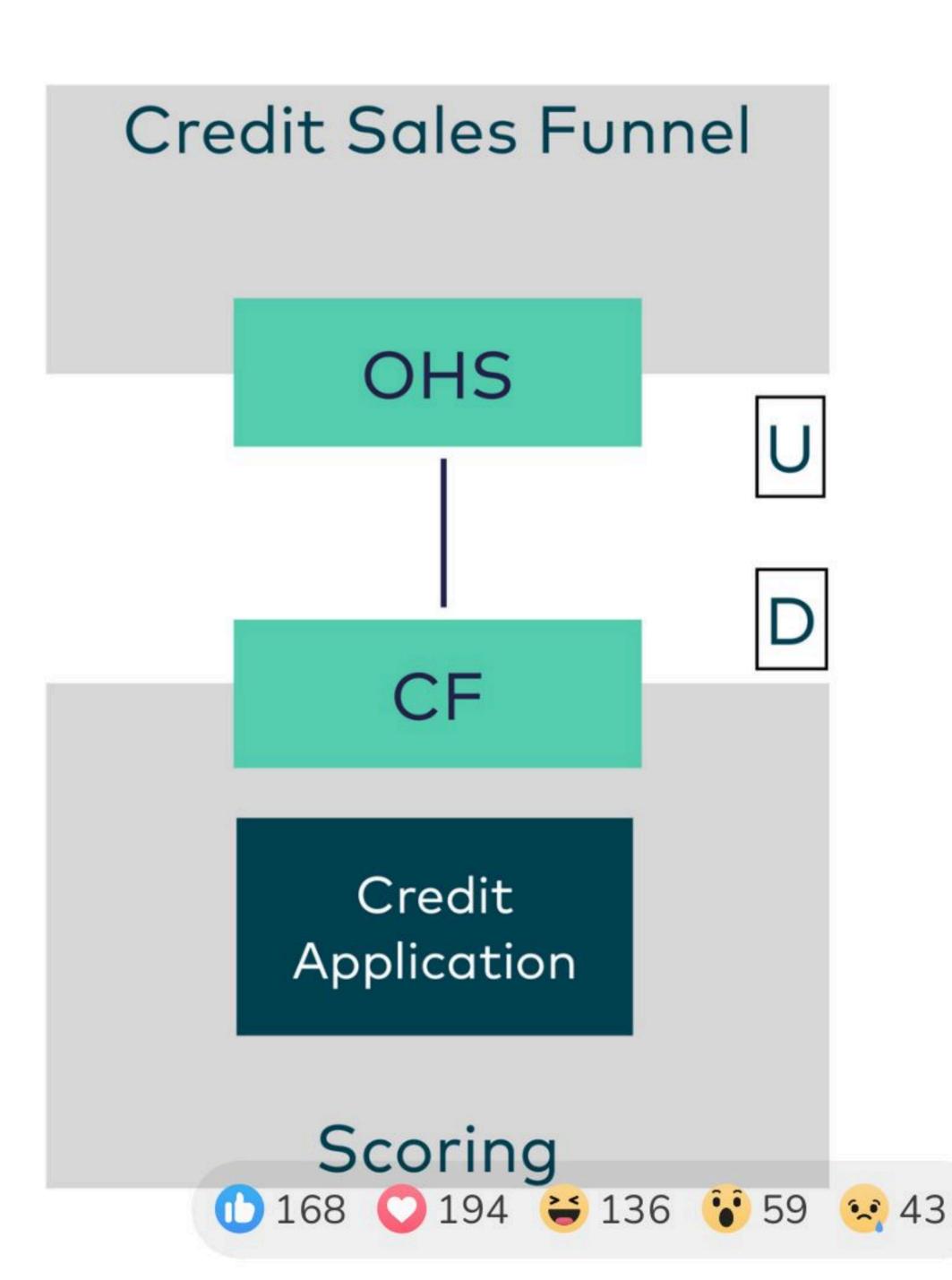
- There is no model-to-model transformation
- Motivation: Simplicity, contracts, force or delight (for the upstream model)
- The team implementing a Conformist is always downstream

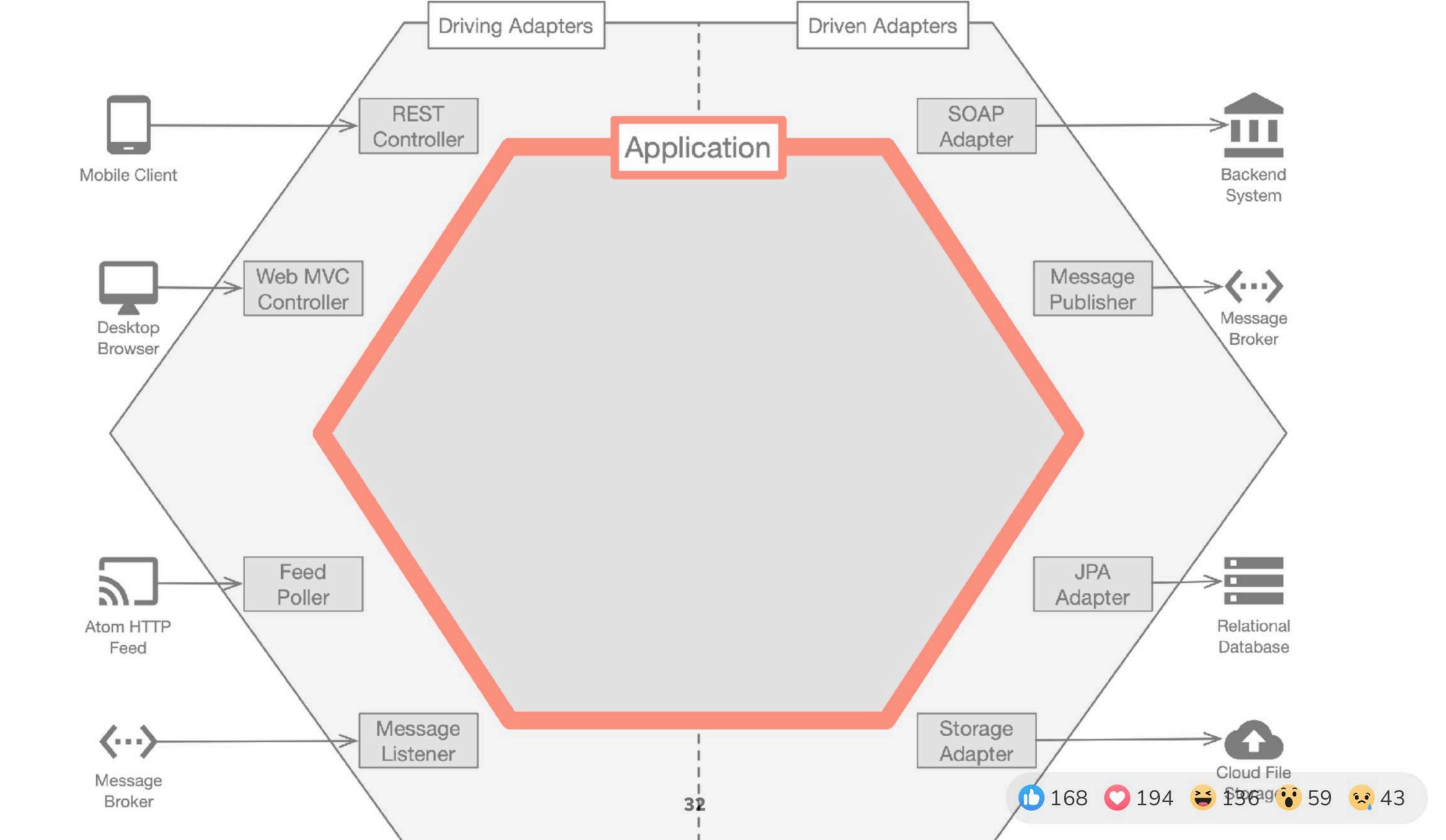


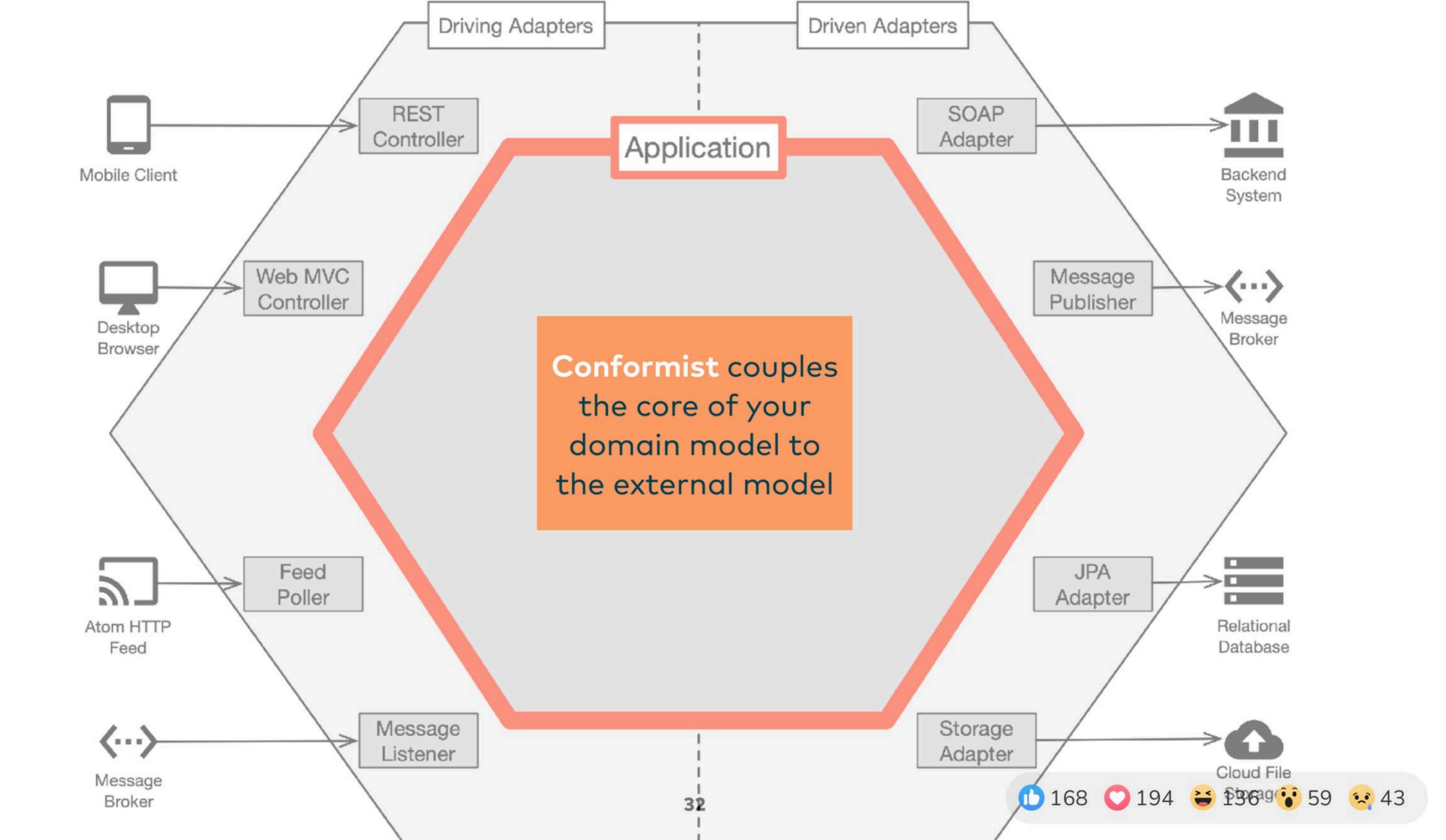


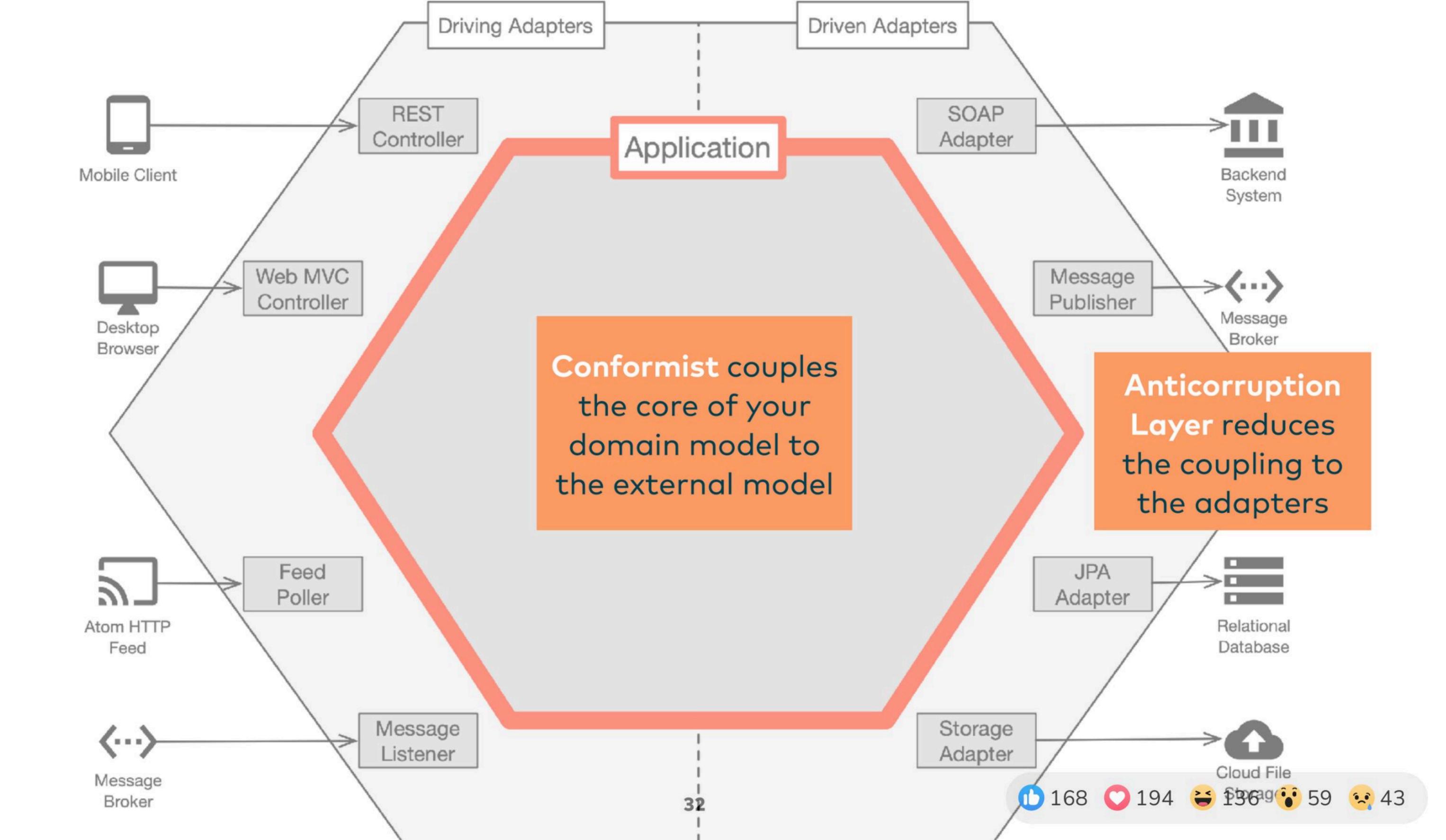
The Conformist slavishly adheres to the upstream model

- 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









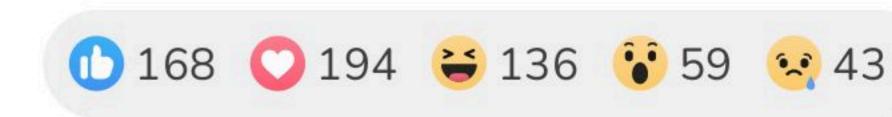
Heuristics for choosing a Conformist

The degree of coupling and also connascence of a Conformist is higher compared to an Anticorruption Layer but there are a few situations in which a Conformist may still be the better choice.

- results.

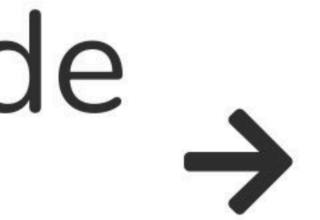
One Bounded Context provides computations, that are highly regulated by legal authorities (collateral value of a real estate for example)

• One team / Bounded Context is considered to be a specialist in aggregations or computations and we don't want others to alter their





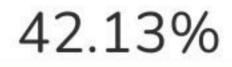
Scan this QR code to join



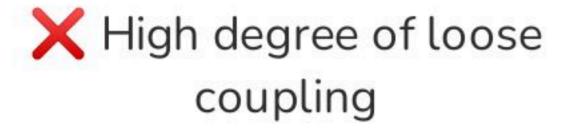




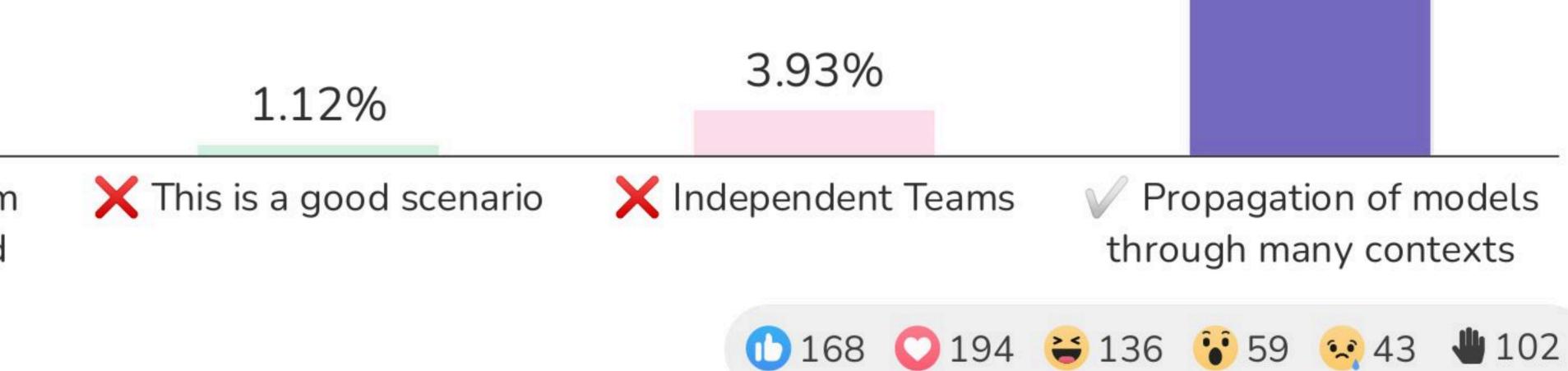
Which comments apply best to this Context Map?







✓ Lot's of cross-team collaboration needed



46.63%

Shared Kerne

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



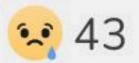




Credit Sales Funnel

Credit Application

Scoring 168 💟 194 🐸 136 😯 59 😣 43

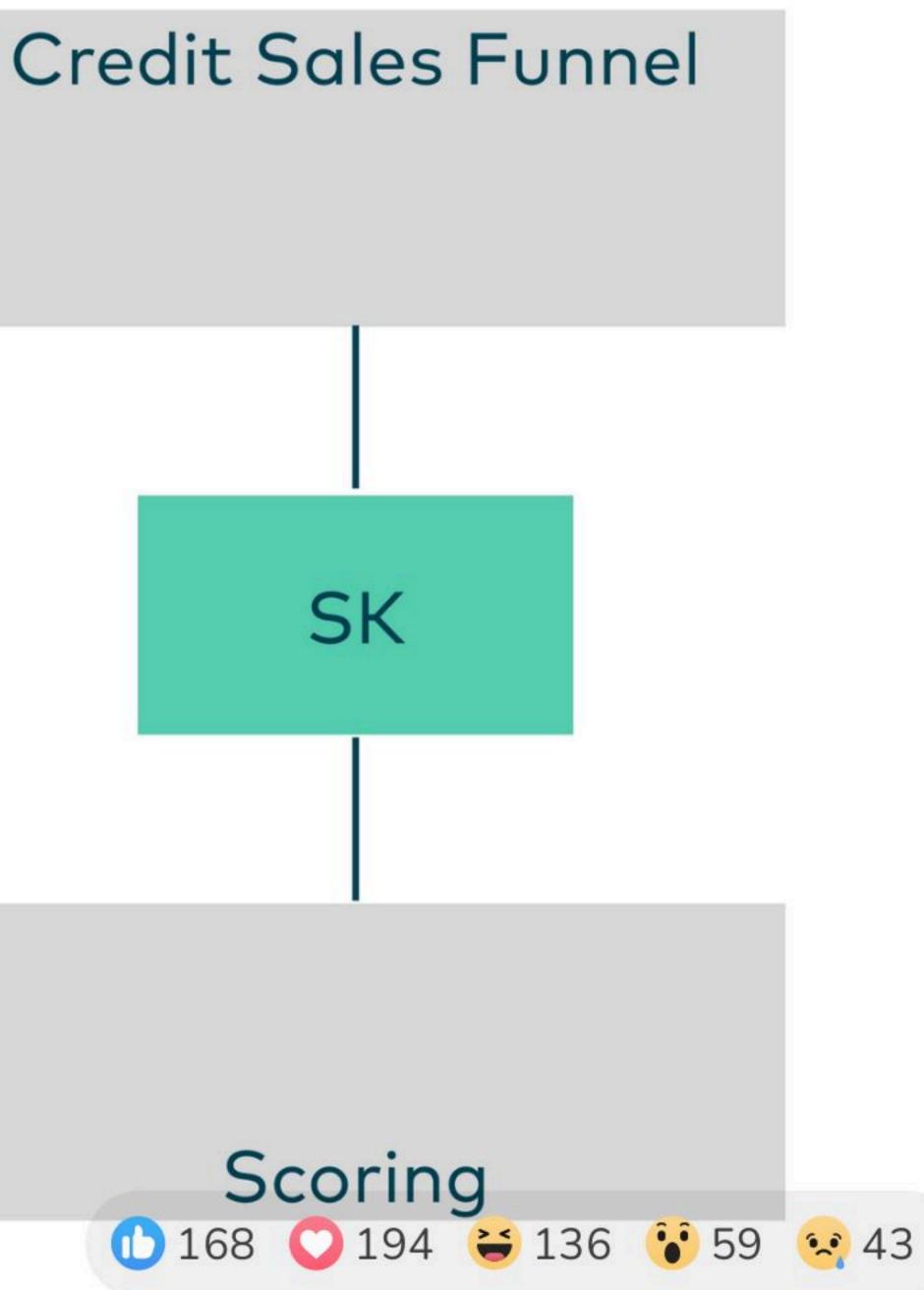


Shared Kerne

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





Heuristics for Shared Kernels

A Shared Kernel introduces a high degree of coupling between the teams and their software and is, therefore, often considered not to be a good option. However, there are situations in which a Shared Kernel may be a good idea:

One team is responsible for two or more bounded contexts which have an overlap in terms of language

Strictly avoid a Shared Kernel when two teams are in a competitive situation or where a lot of backdoor politics are being played

When two teams have a Shared Kernel, they should form a Partnership...



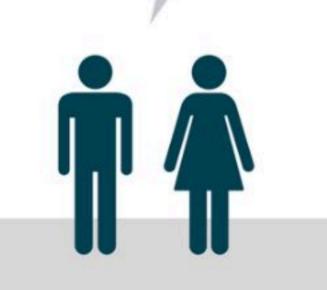
Partnership

Partnership is about cooperative relationships between teams

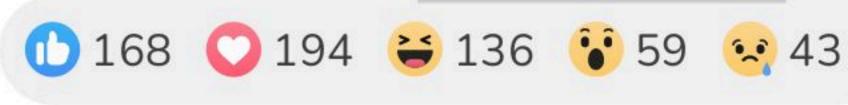
- Establishes a process for coordinated planning of development and joint management of integration
- Not technical at all, Partnership is plain organizational
- Recommended for teams which depend on a Shared Kernel

Ok, let's coordinate our efforts





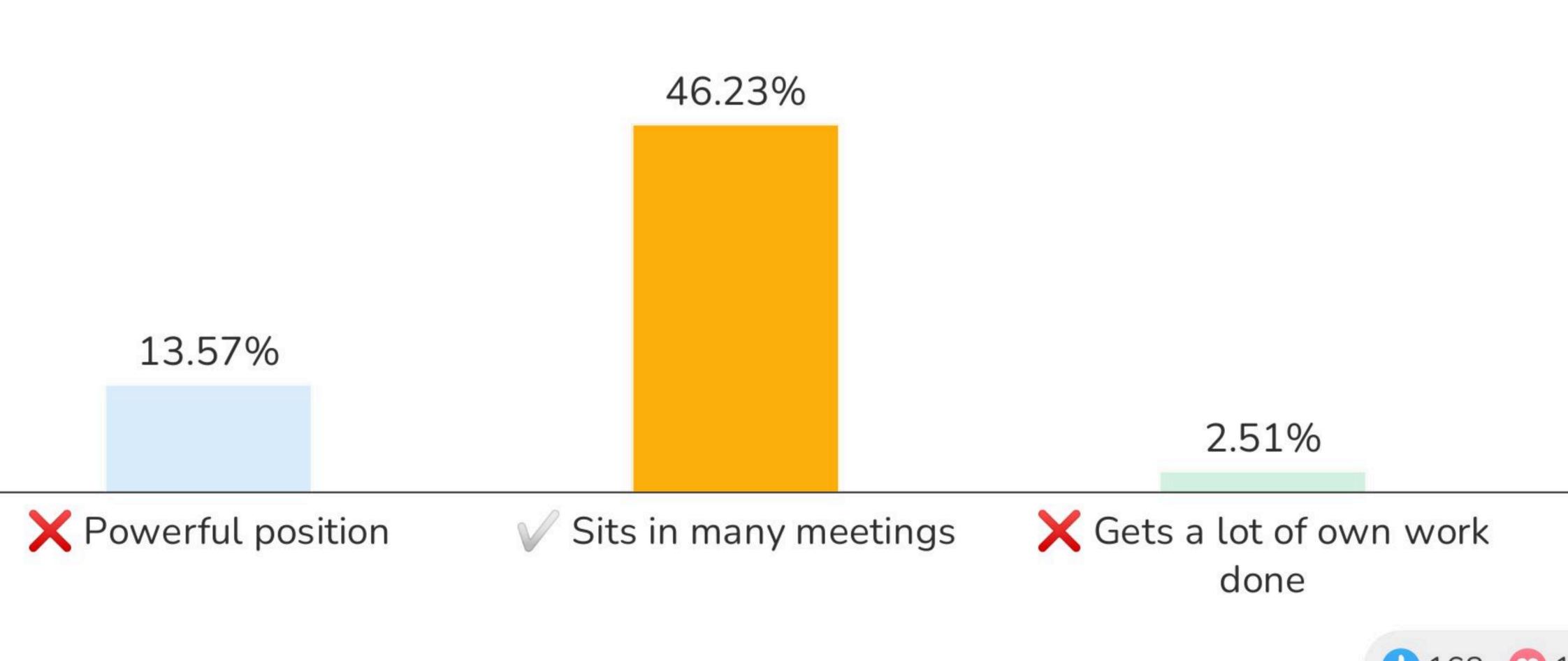
Scoring



We want to adjust something

Credit Sales Funnel

Which comments apply best regarding to the team in the middle



37.69%

Can't drive its own agenda



Customer-Supplier

A Customer-Supplier development gives the downstream team some influence

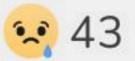
- Downstream requirements factor into upstream planning. Therefore, the downstream team gains some influence over the priorities and tasks of the upstream team
- Customer-Supplier is organizational
- Mind "vetoing customer" and customer against an OHS as anti-patterns





Credit Sales Funnel We need more fields in the application

Scoring 168 💟 194 🐸 136 당 59 😣 43

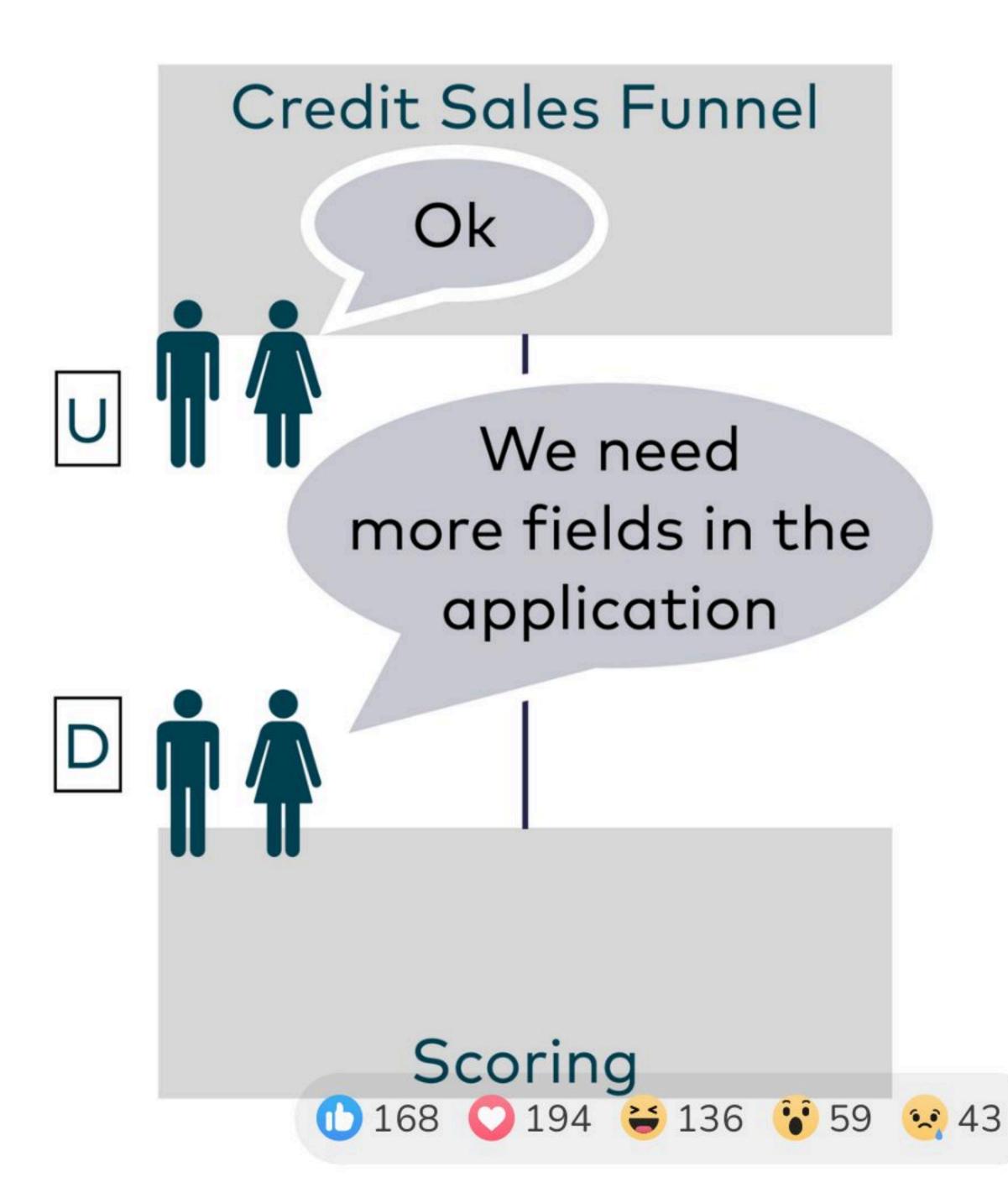


Customer-Supplier

A Customer-Supplier development gives the downstream team some influence

- Downstream requirements factor into upstream planning. Therefore, the 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, the 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



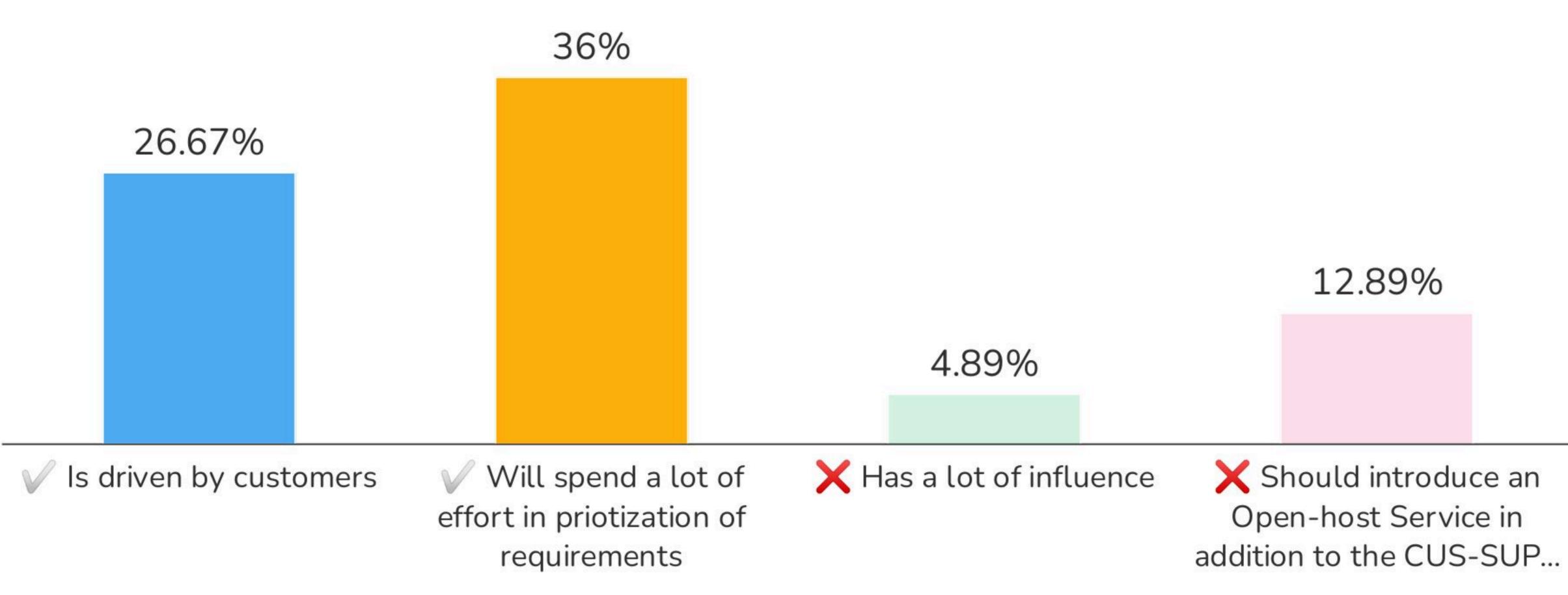


Credit Sales Funnel SUP CUS





Which statements are valid for the supplier?







19.56%

Should introduce an V **Open-host Service** instead of the CUS-SUP relationships

Separate Ways

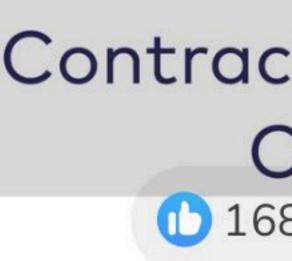
A bounded context has no connections to others

- Sometimes integration is too expensive or takes very long
- The teams choose separate ways in order to focus on specialized solutions
- Interesting pattern for minimum viable products or organizational solutions in uncertain market conditions



Credit Sales Funnel

Manual process for entering credit application data in



contract Contract Offering and Creation 168 \bigcirc 194 \rightleftharpoons 136 \bigcirc 59 \backsim 43



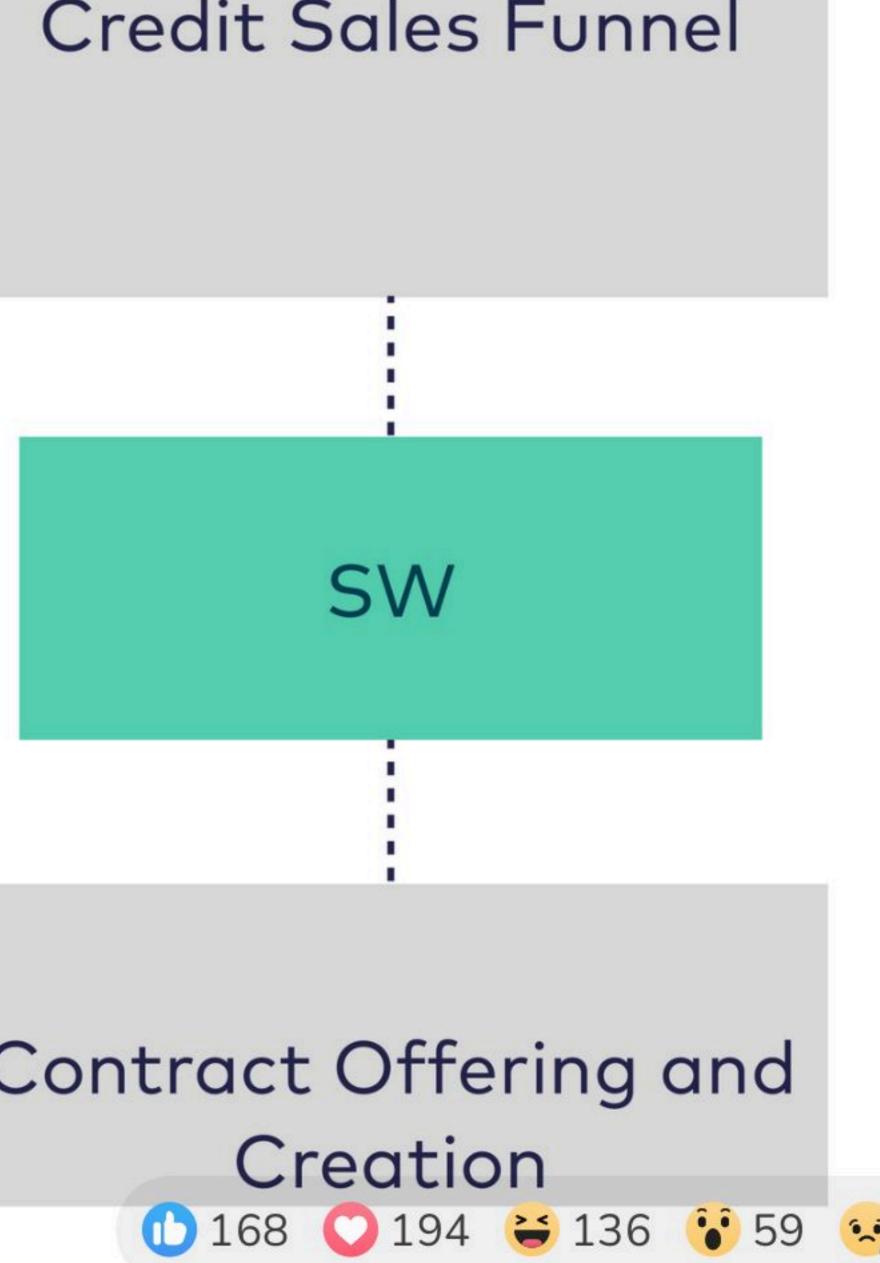
Separate Ways

A bounded context has no connections to others

- Sometimes integration is too expensive or takes very long
- The teams choose separate ways in order to focus on specialized solutions
- Interesting pattern for minimum viable products or organizational solutions in uncertain market conditions







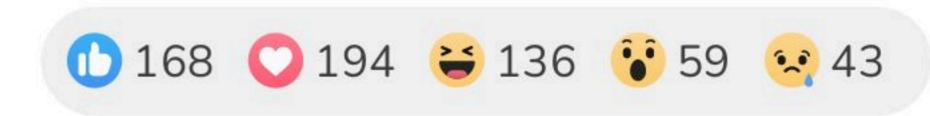


A well documented language shared between bounded contexts

- Every bounded context can translate in and out from that language
 Sometimes defined by a consortium of
- Sometimes defined by a consortium the most important stakeholders / teams
- Often combined with Open-host Service
- Examples: iCalendar, vCard, ZugFerd







Credit Sales Funnel

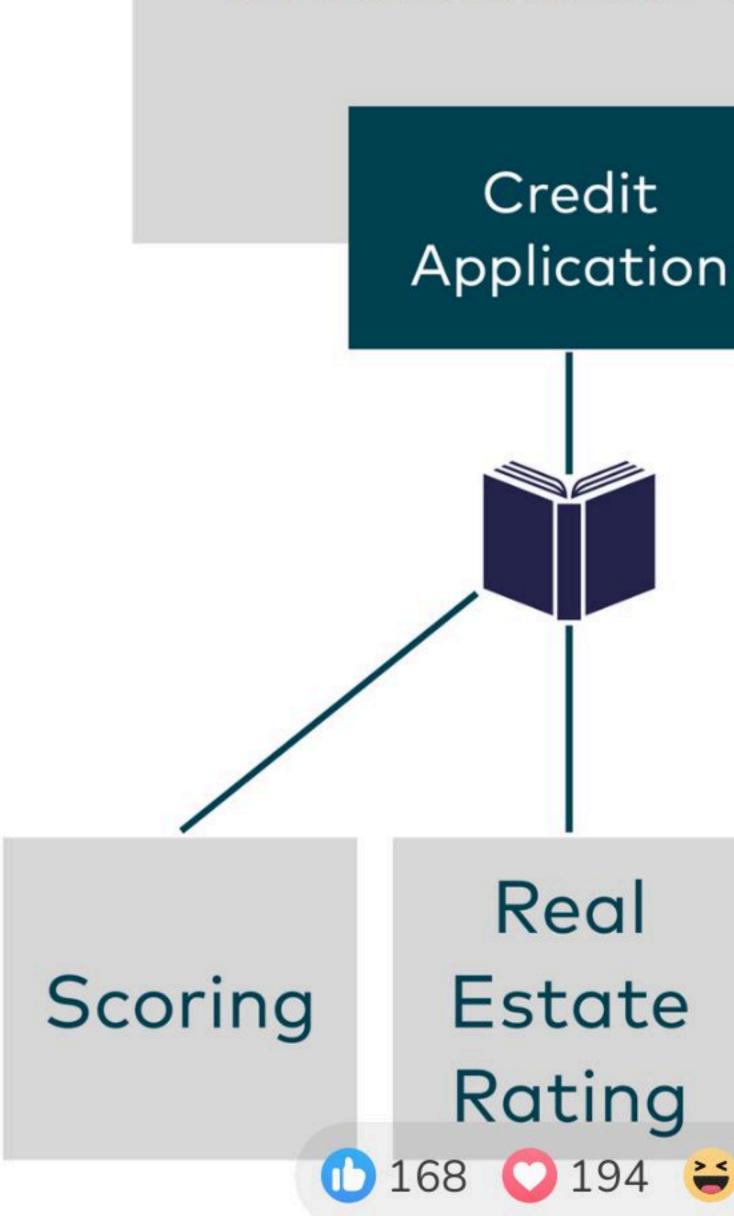
Credit Application



A well documented language shared between bounded contexts

- Every bounded context can translate in and out from that language
- Sometimes defined by a consortium of the most important stakeholders / teams
- Often combined with Open-host Service
- Examples: iCalendar, vCard, ZugFerd





Credit Sales Funnel

Estate Rating 168 💟 194 🐸 136 😯 59 😣 43

A well documented language shared between bounded contexts

- Every bounded context can translate in and out from that language
- Sometimes defined by a consortium of the most important stakeholders / teams
- Often combined with Open-host Service
- Examples: iCalendar, vCard, ZugFerd





Credit Sales Funnel

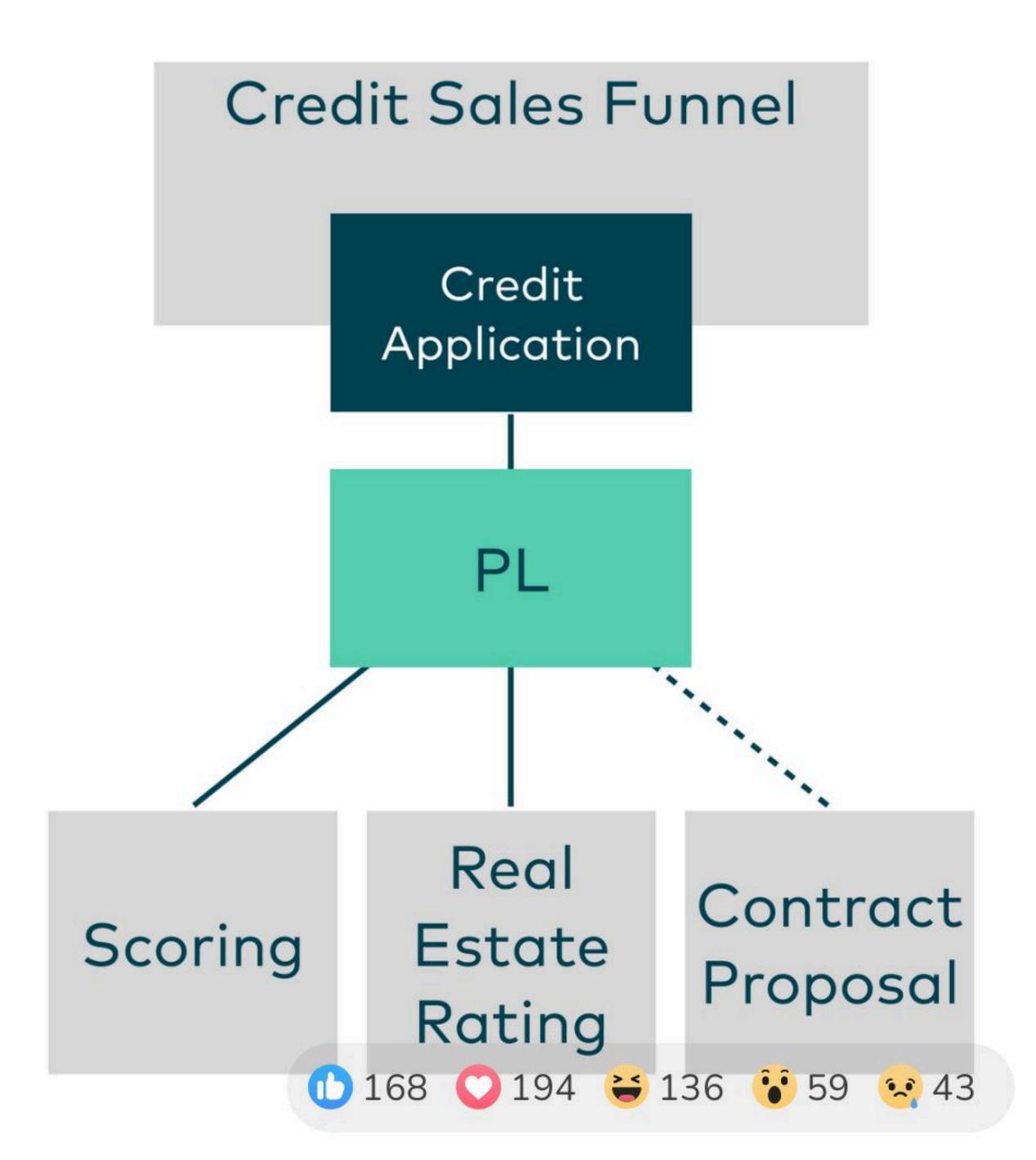
Credit Application

Real Contract Estate Proposal Rating 168 🔘 194 🐸 136 😯 59 😣 43

A well documented language shared between bounded contexts

- Every bounded context can translate in and out from that language
- Sometimes defined by a consortium of the most important stakeholders / teams
- Often combined with Open-host Service
- Examples: iCalendar, vCard, ZugFerd



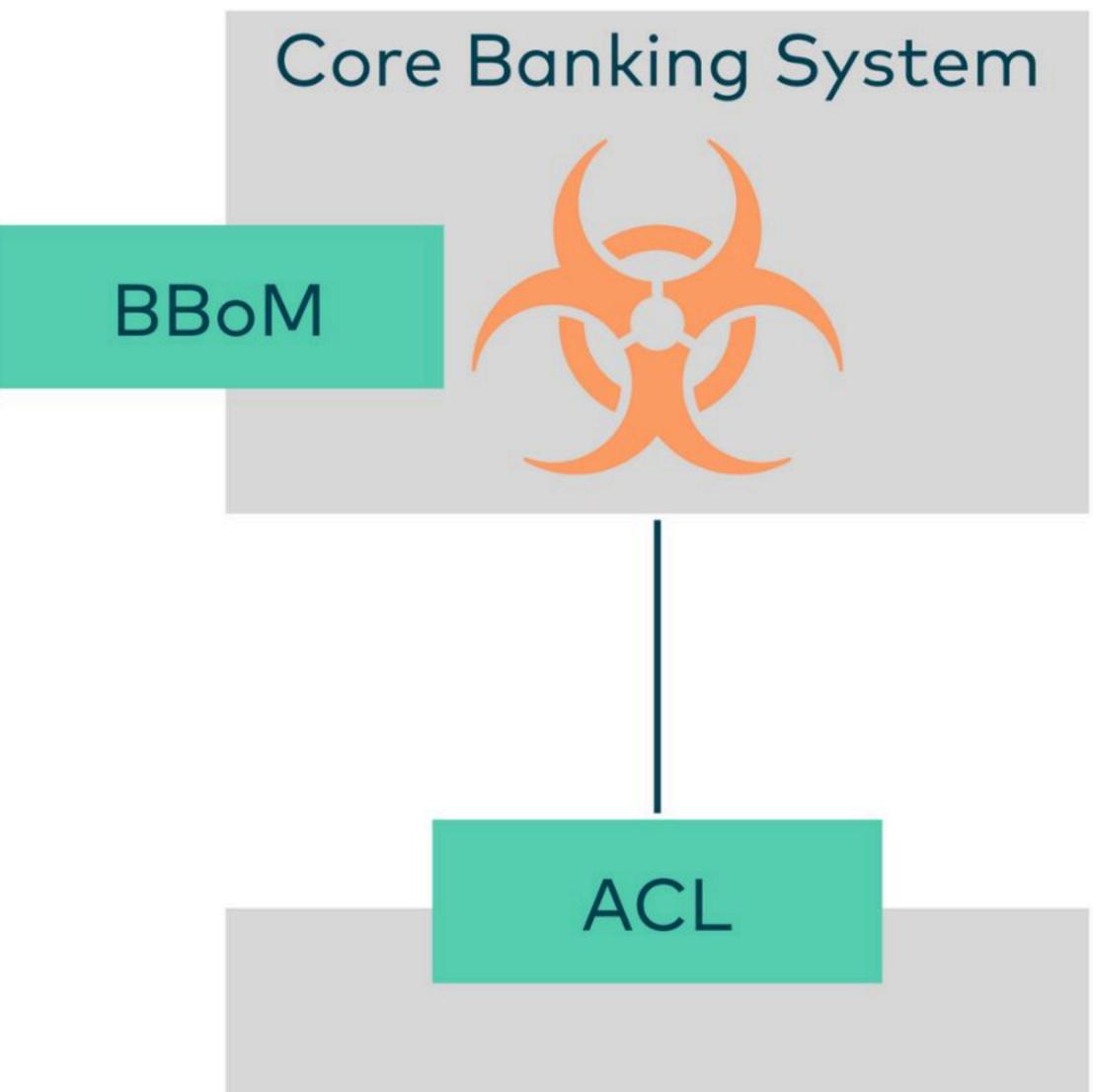


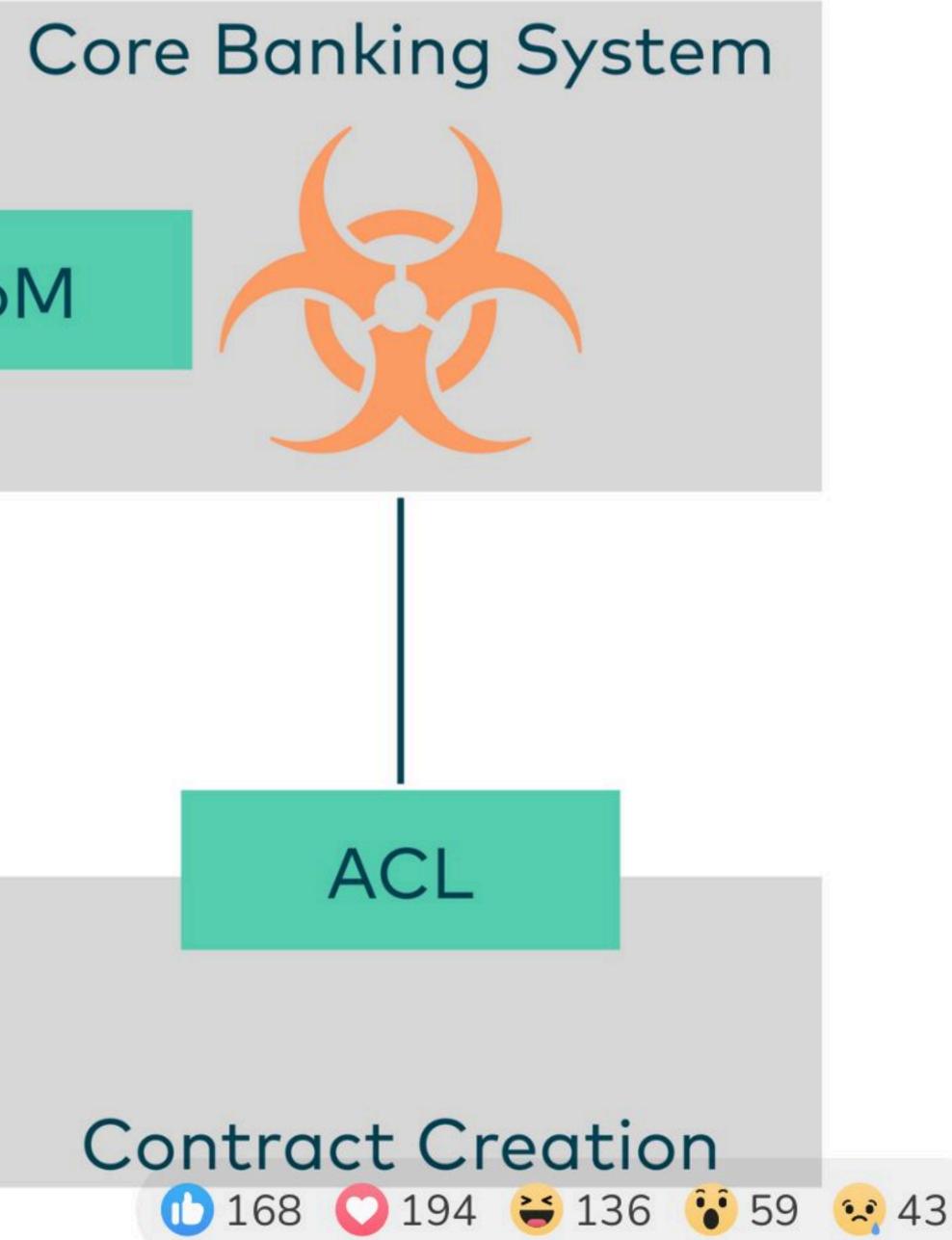
Big Ball Of Mud

A (part of a) system which is a mess by having mixed models and inconsistent boundaries

- Don't let the (lousy) model of the Big Ball Of Mud propagate into your context
- Anticorruption Layer is the pattern of choice on the downstream
- Demarcation of bad model or system quality





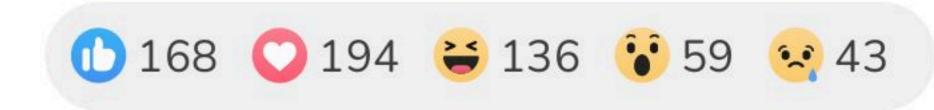


Call Relationship



System ABC







- Call Relationship
- Team Relationship Level 1

Upstream



System ABC











- Call Relationship
- Team Relationship Level 1
- API Level

Upstream



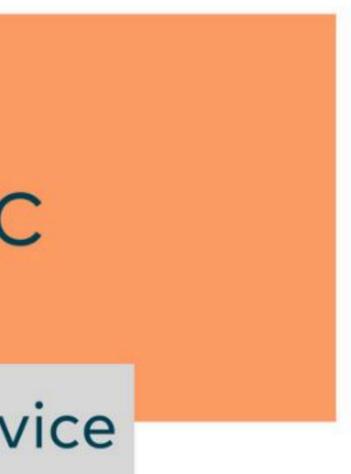
System ABC

Open Host Service











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

Upstream



System ABC

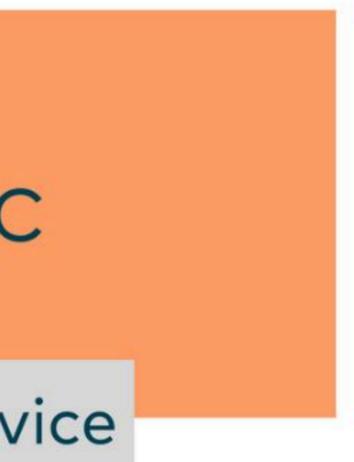
Open Host Service

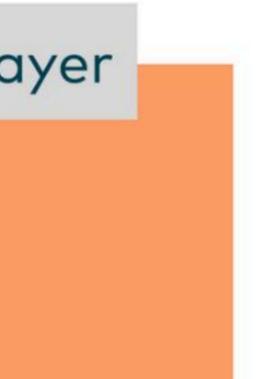
Anticorruption Layer

System Y





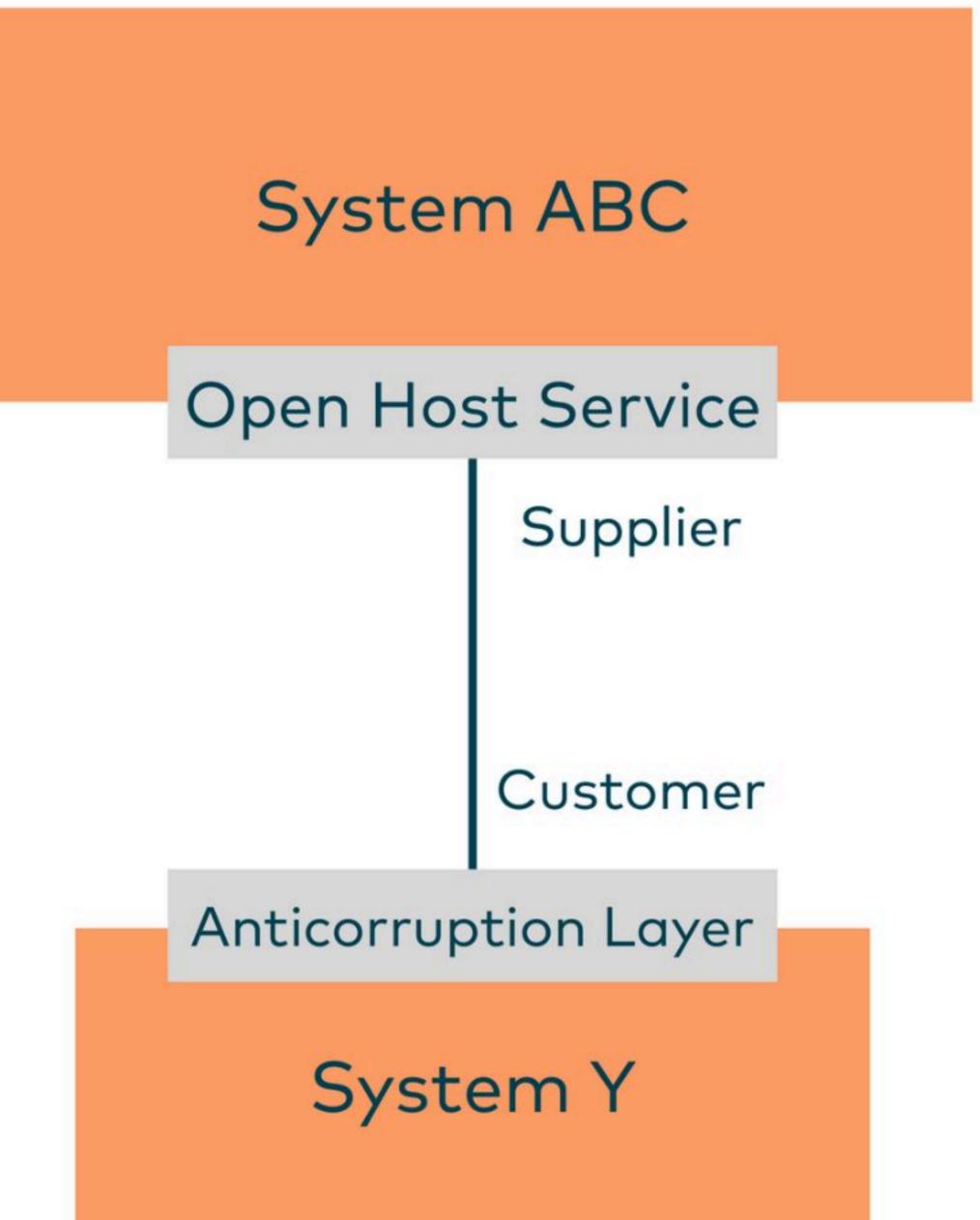






- Call Relationship
- Team Relationship Level 1
- API Level
- **Model Propagation**
- Team Relationship Level 2

Upstream











		2	T
R	el	a	It

Open-host Service

Anticorruption Layer

Conformist

Shared Kernel

Partnership

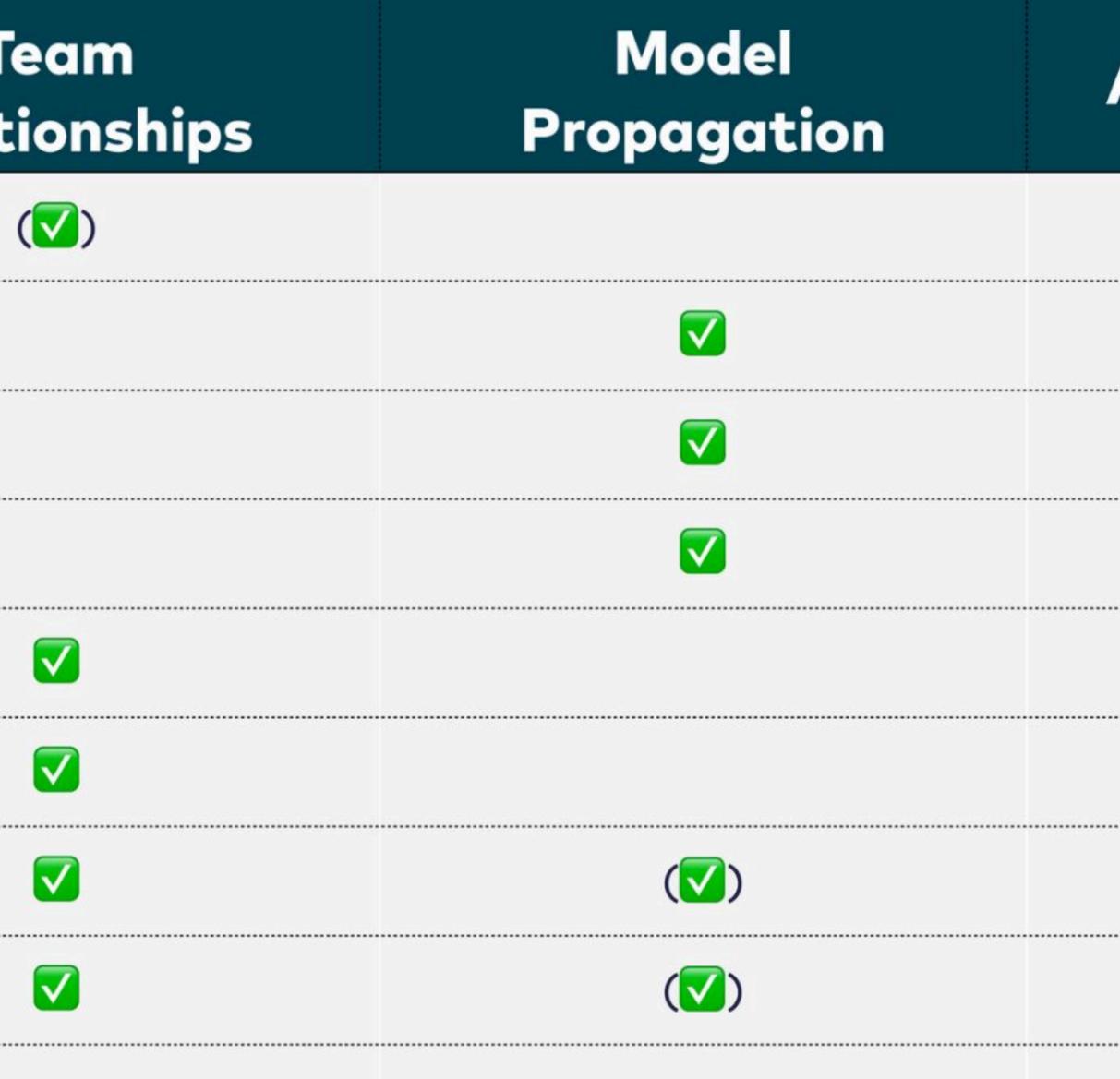
Customer-Supplier

Separate Ways

Published Language

Big Ball Of Mud







API / "technical"



••••••	
8 🔘 194 🐸 136 당 59 😒	43

Some of the patterns map to team dependencies

Mutually Dependent

Team Relationships

Partnership

Shared Kernel







Some of the patterns map to team dependencies

Team Relationships



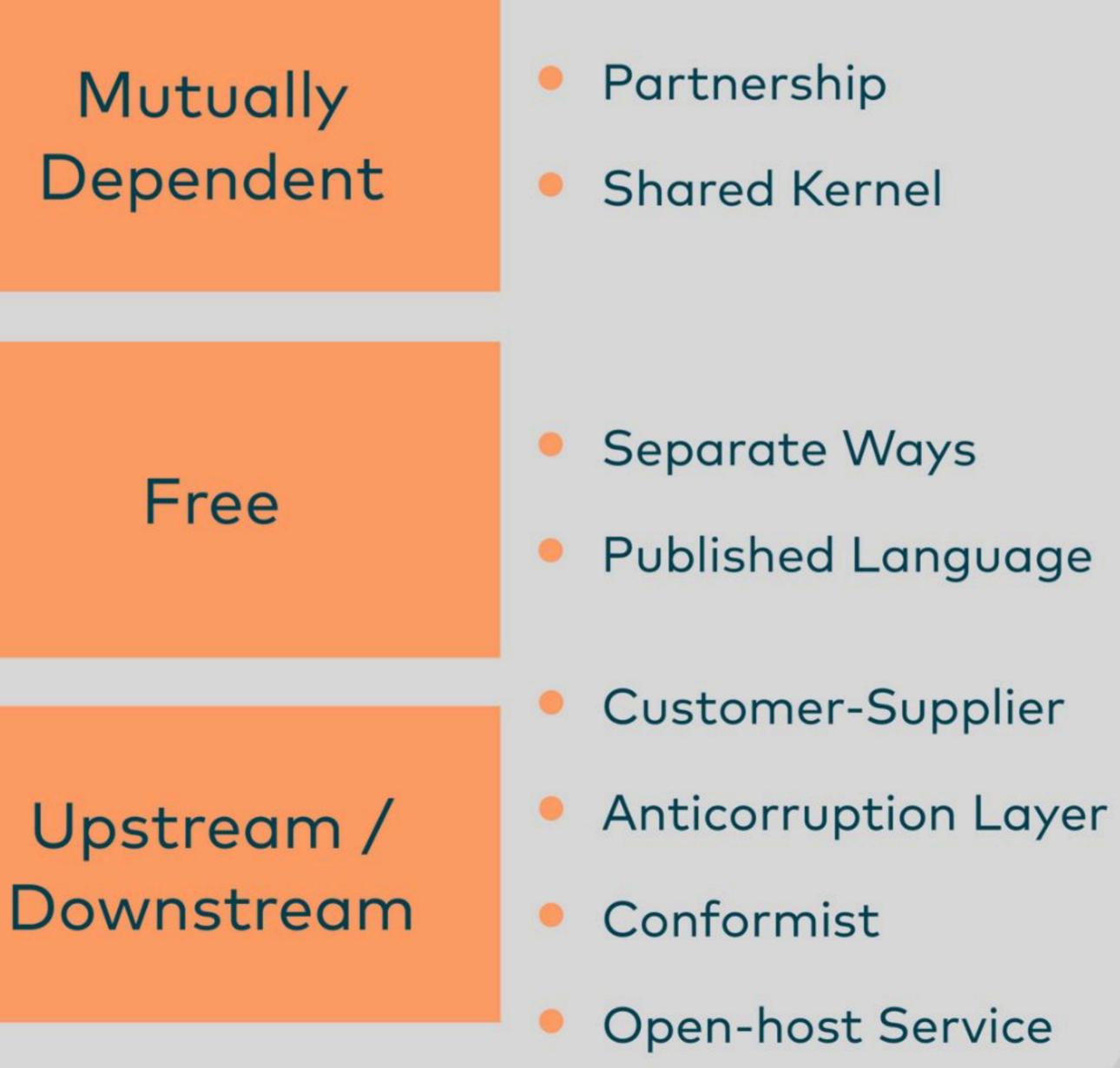






Some of the patterns map to team dependencies

Team Relationships









Mind team communication

Team munication Cor





Customer / Supplier







Separate Ways







Mind team communication

Team mmunication Sor



Shared Kernel



Customer / Supplier





Anticorruption Layer

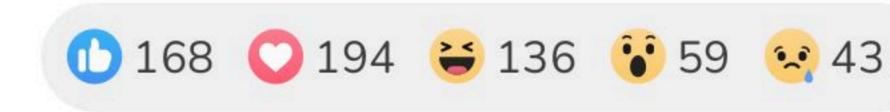


Separate Ways



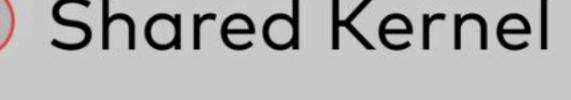
Open / Host Service

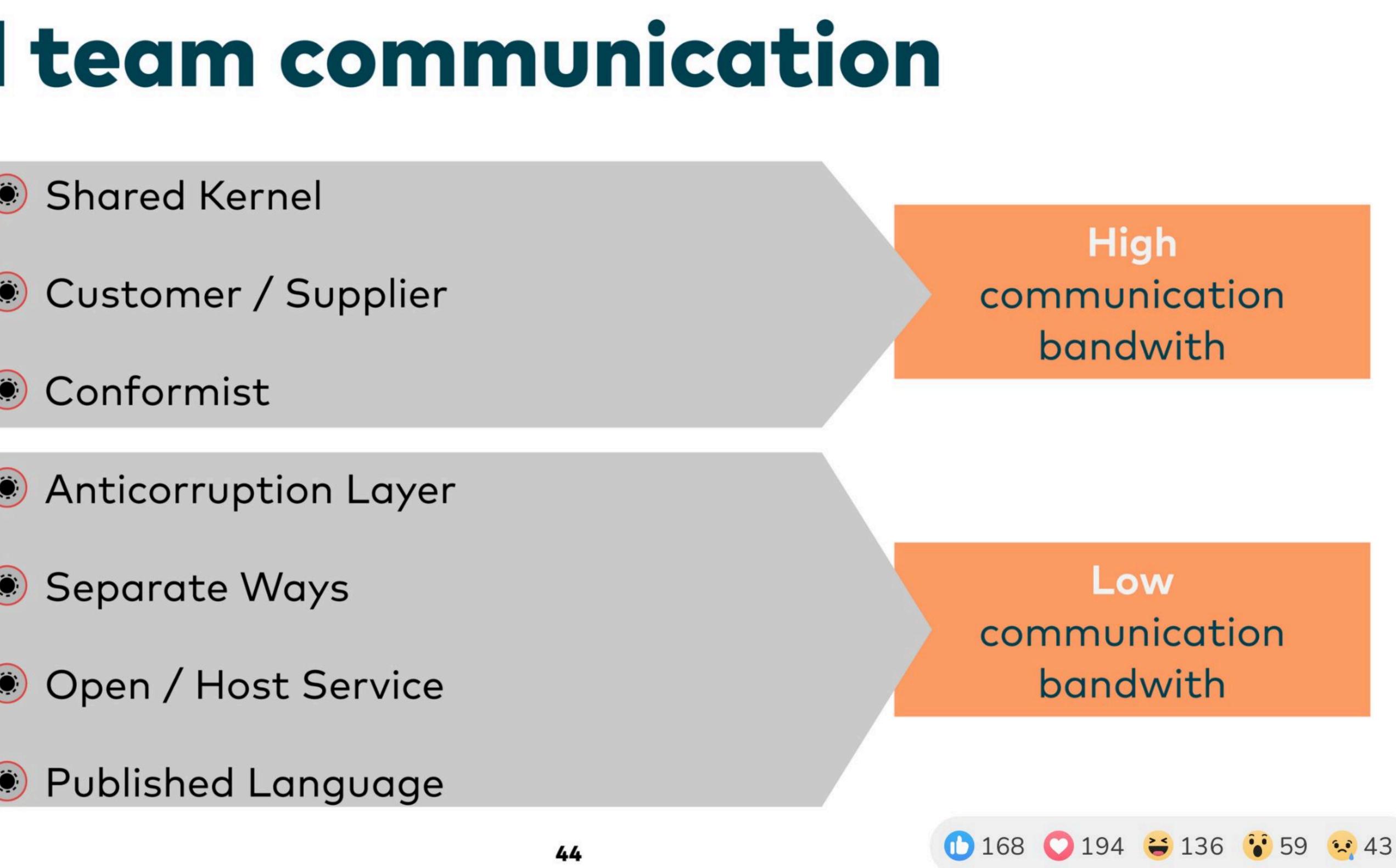




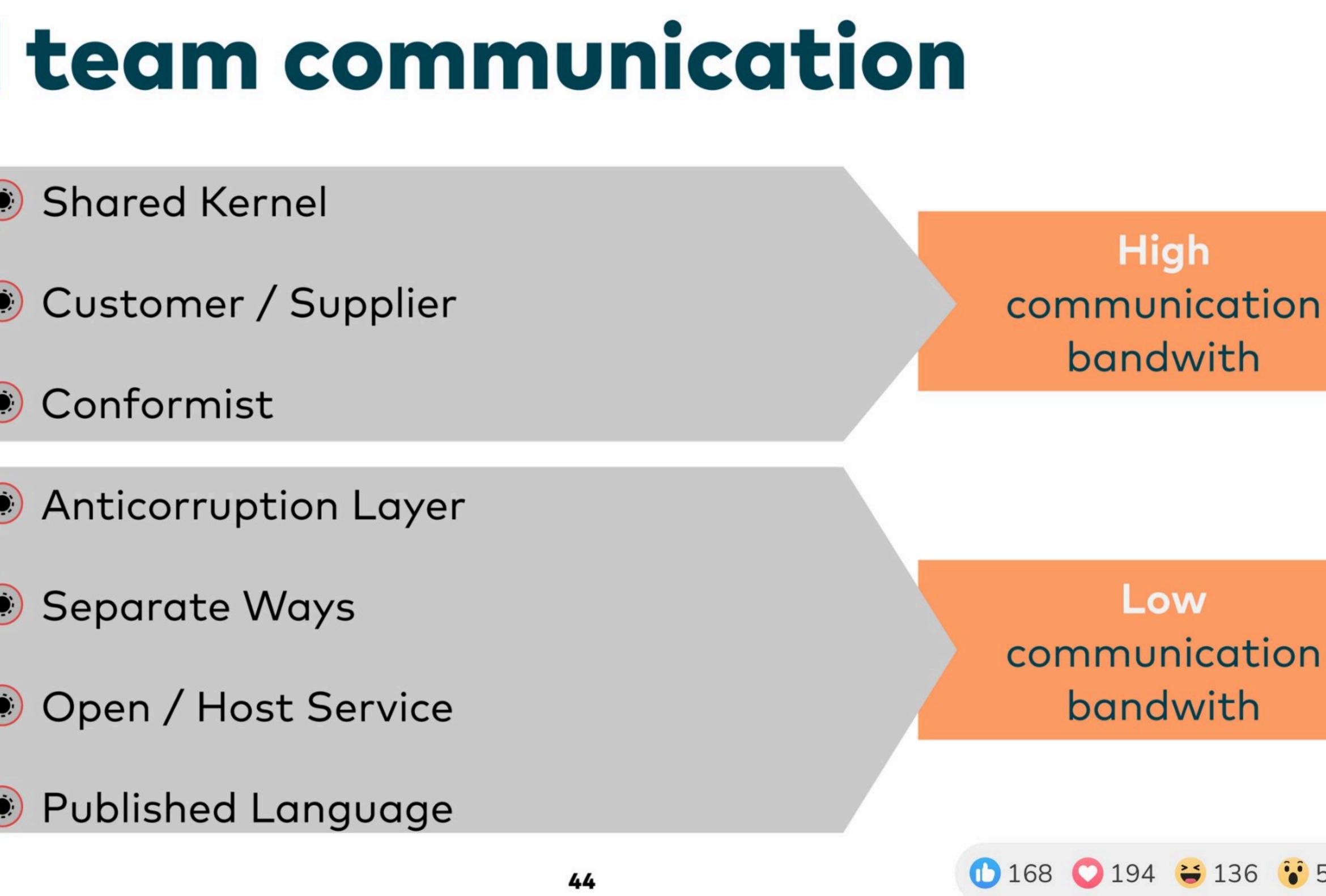
High communication bandwith

Team mmunication Cor

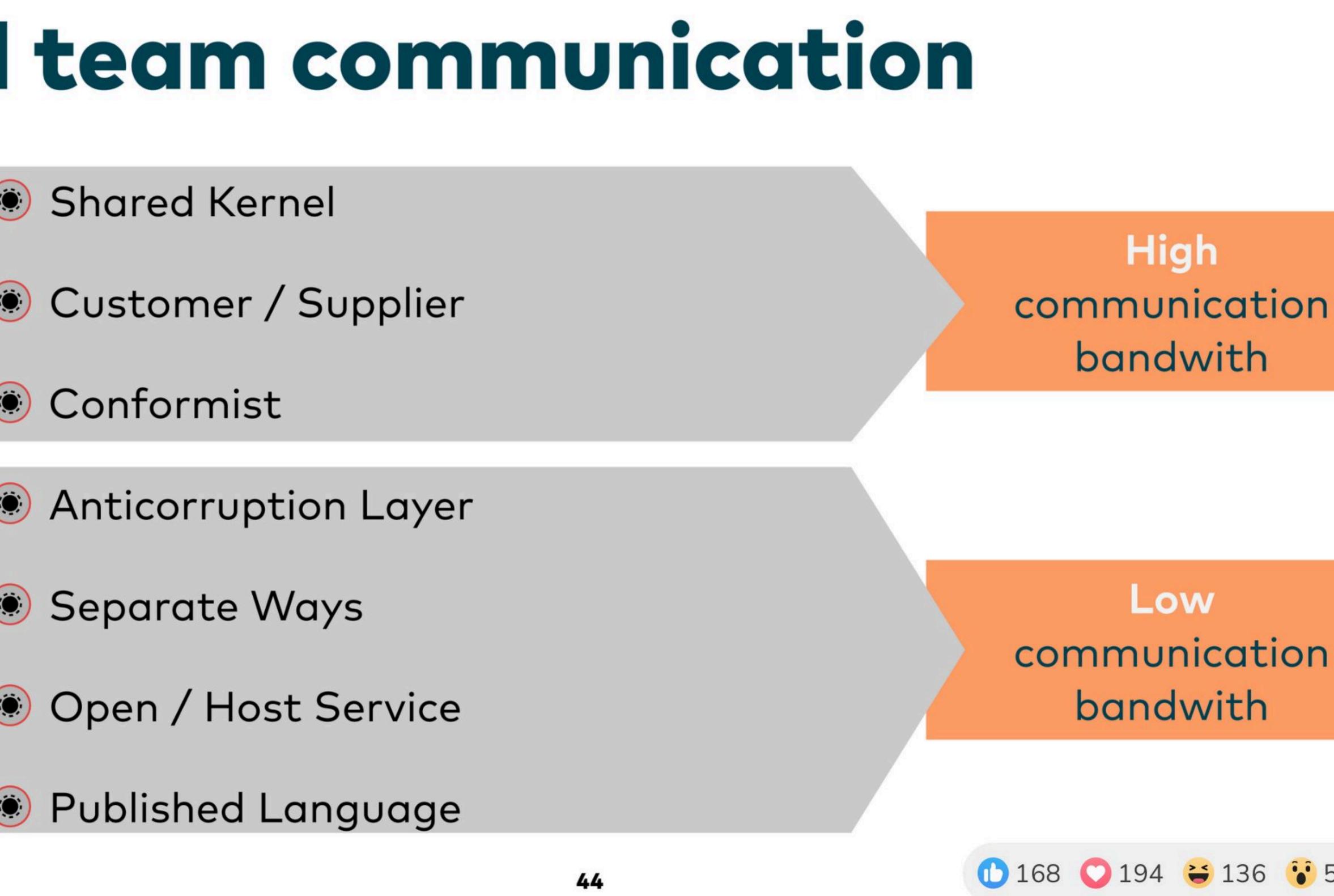


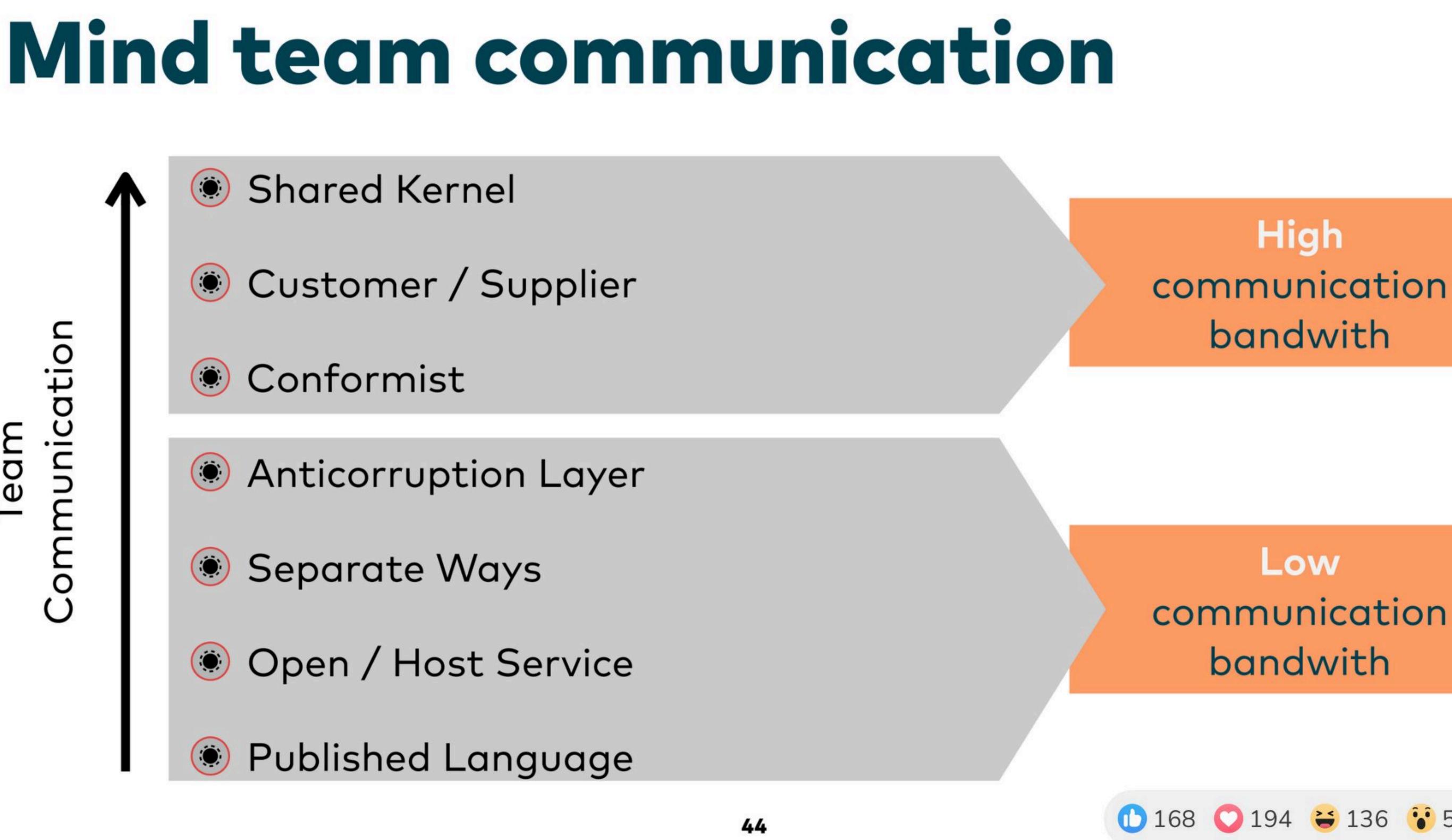






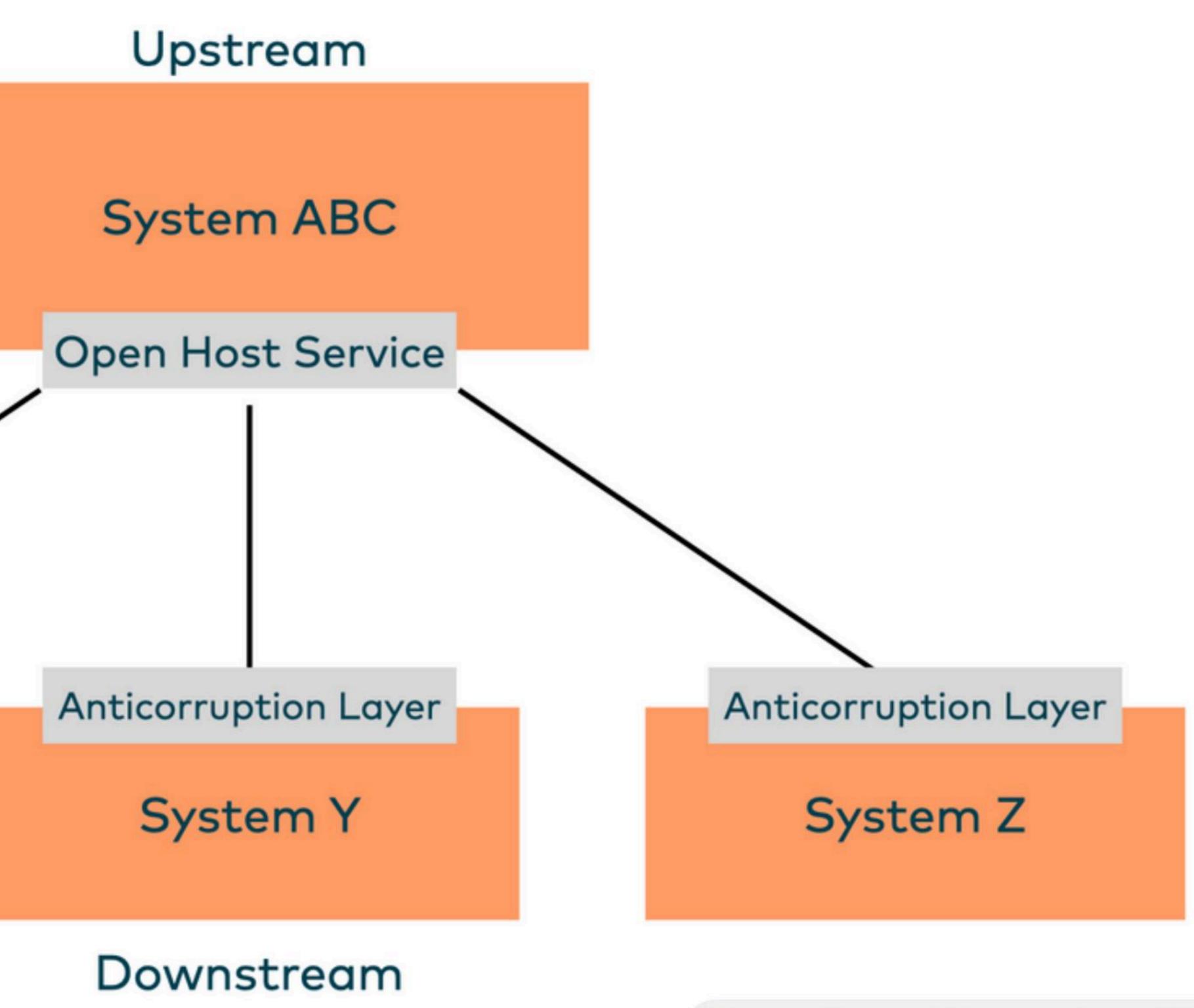






Anticorruption Layer

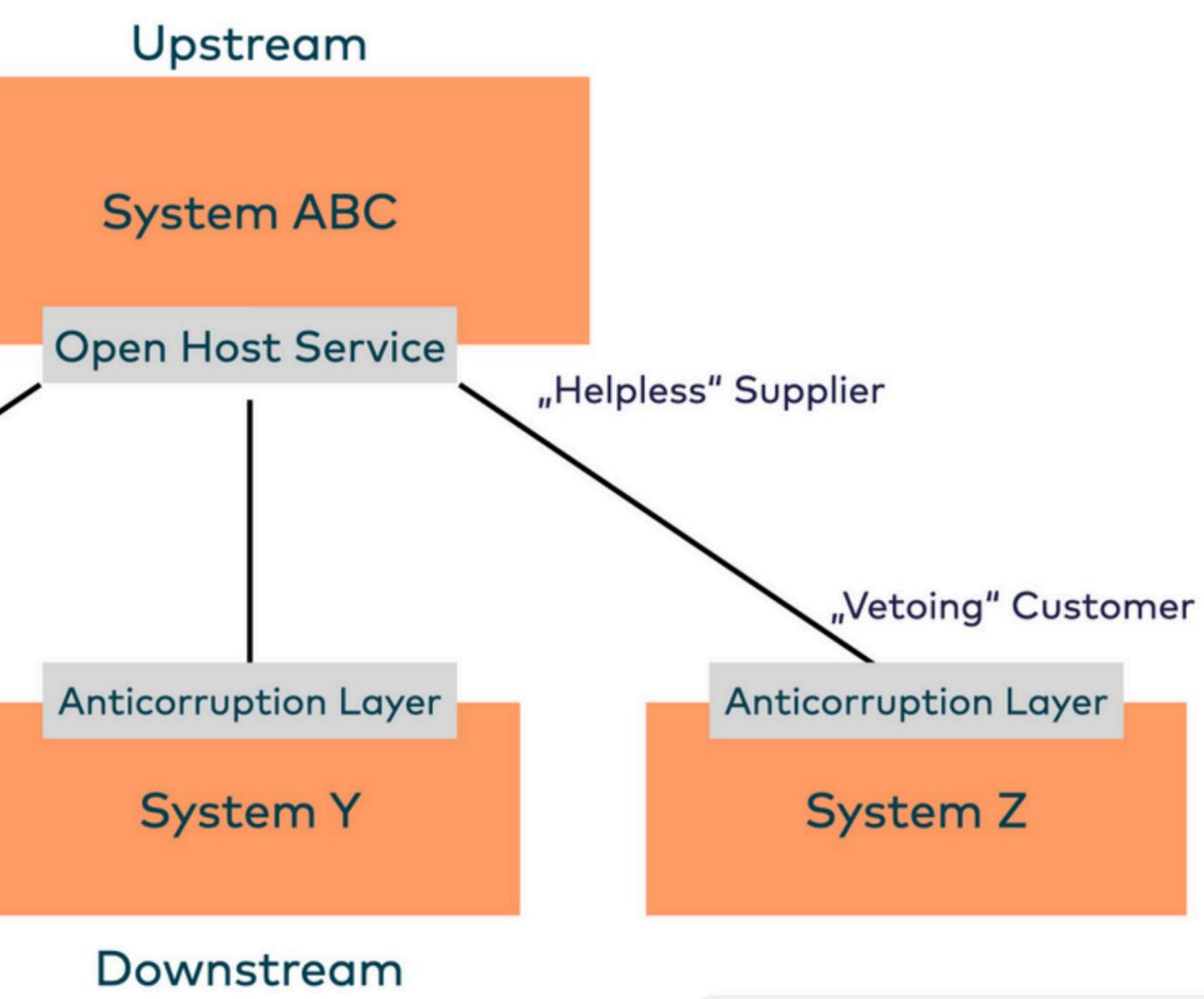




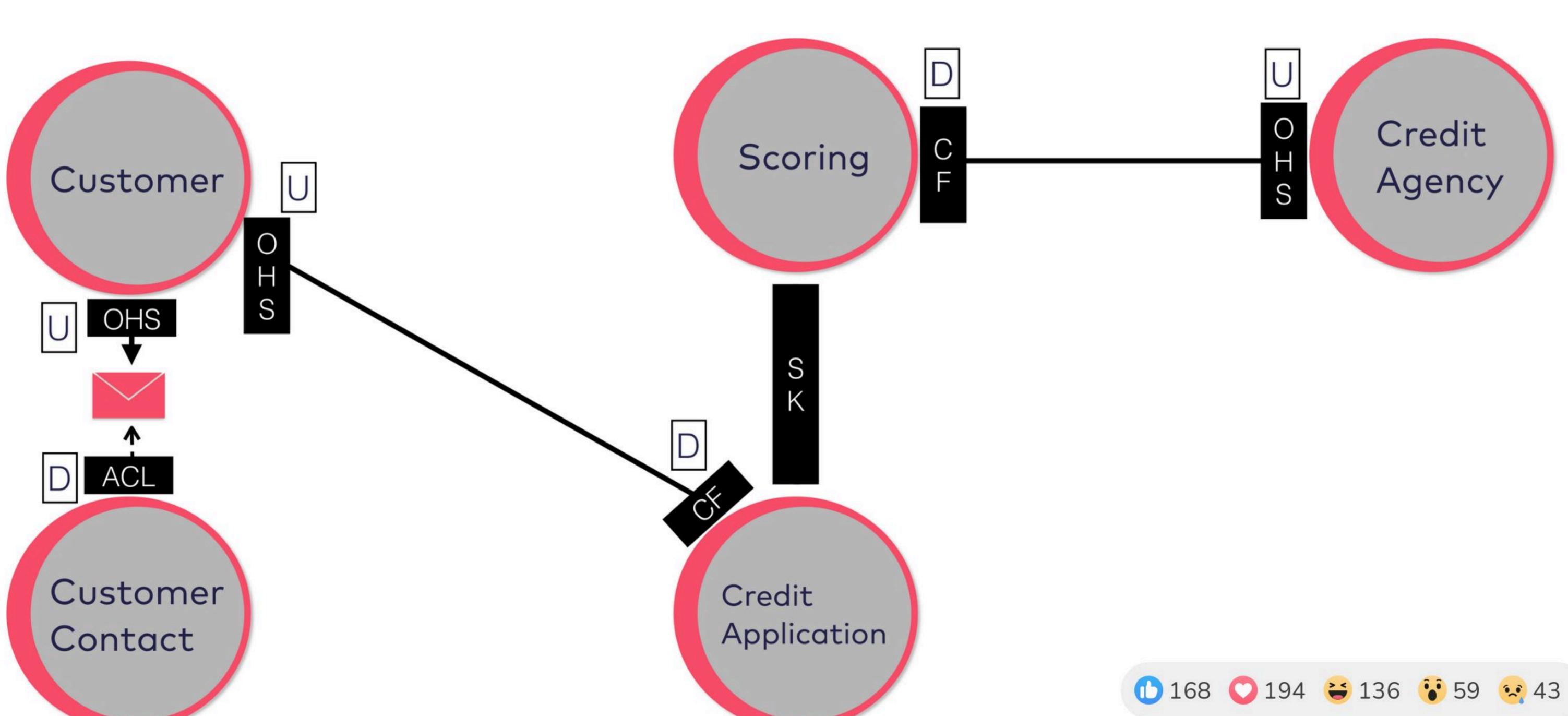


Anticorruption Layer

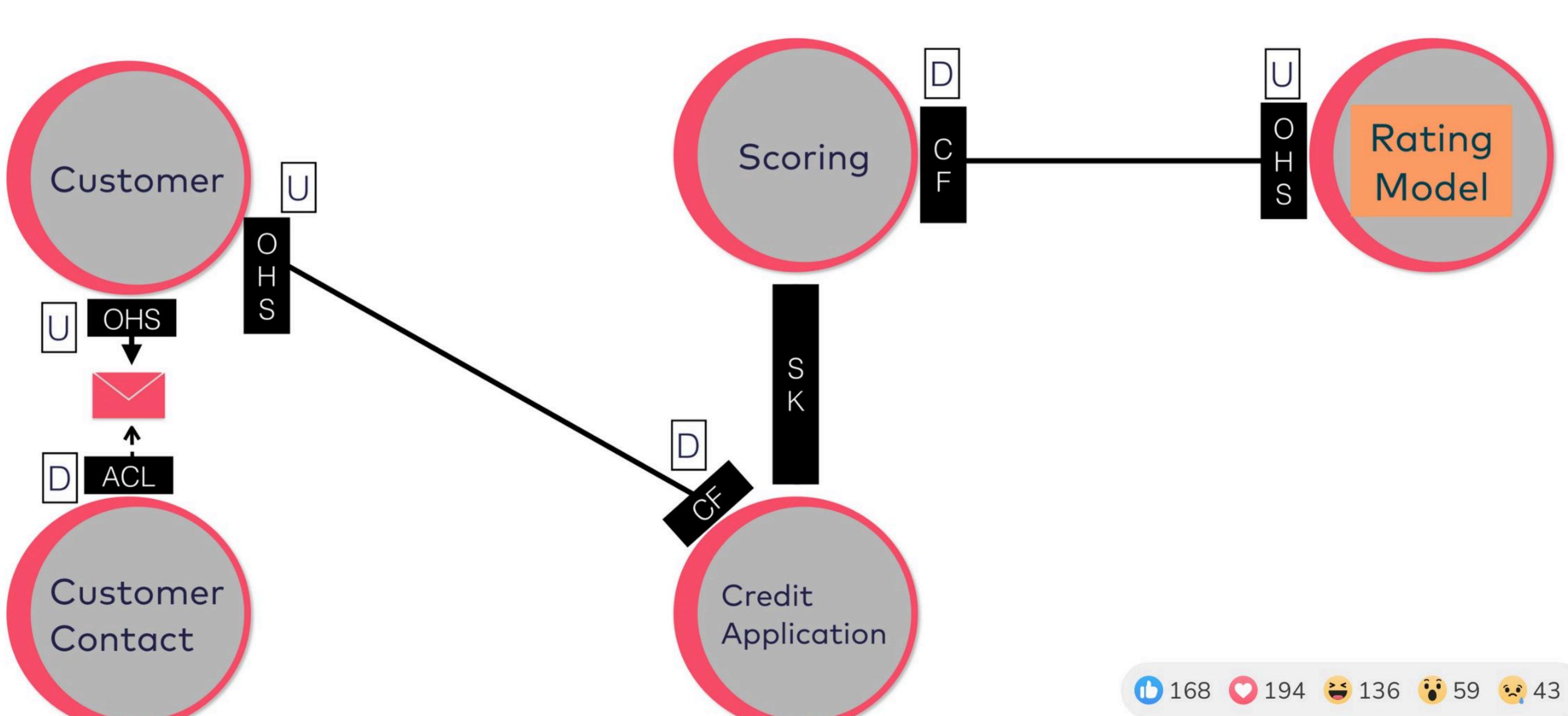




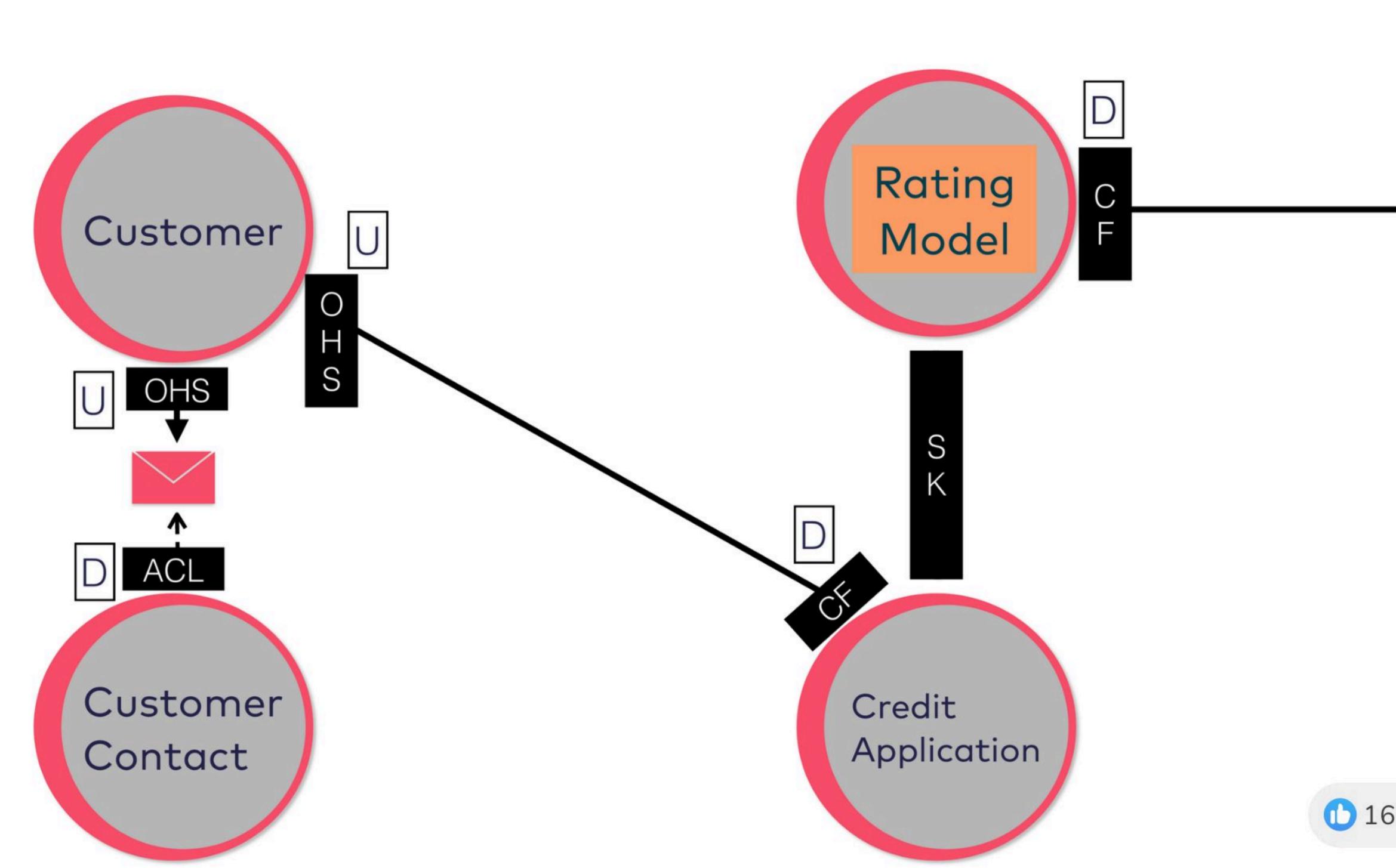






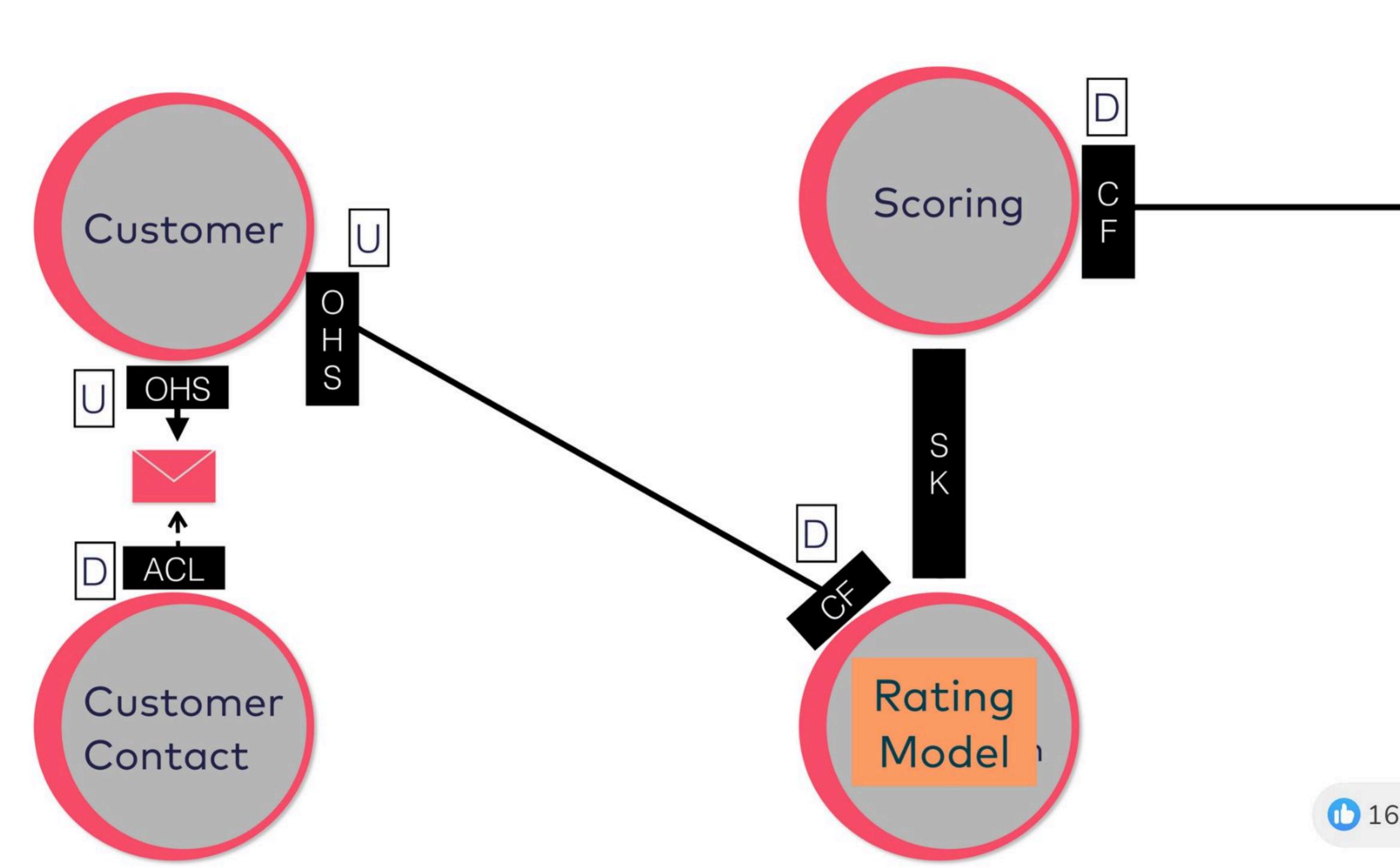






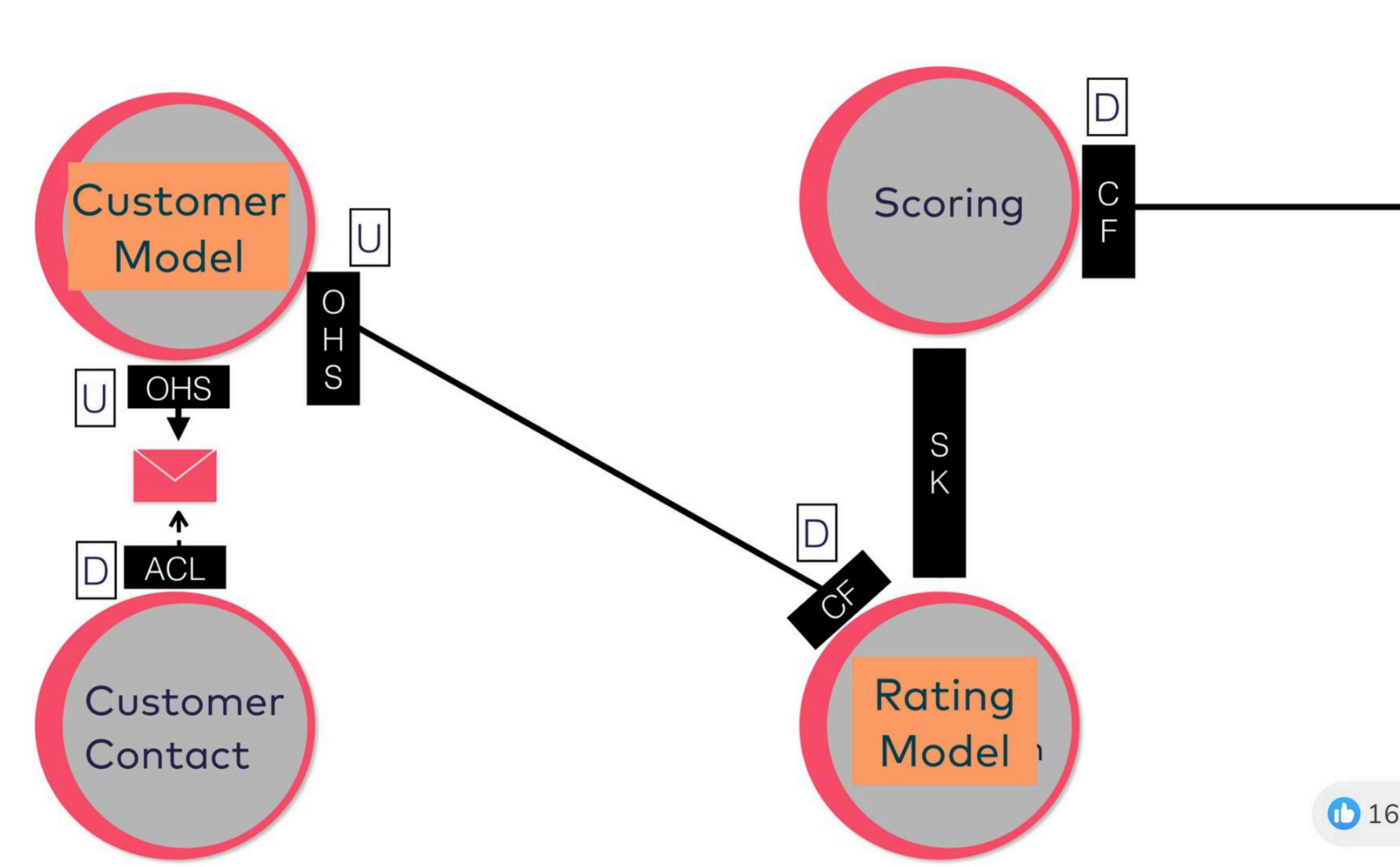






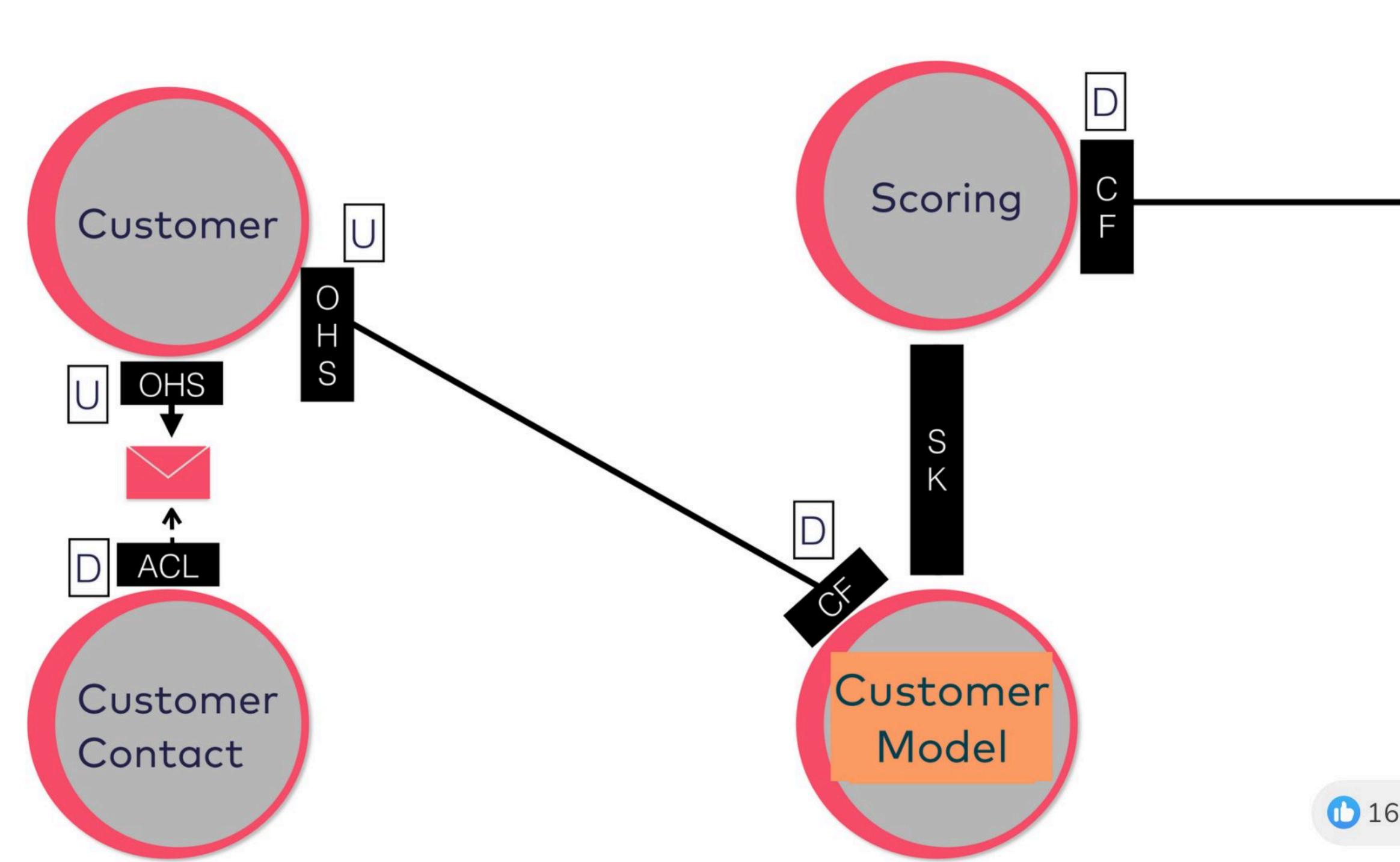






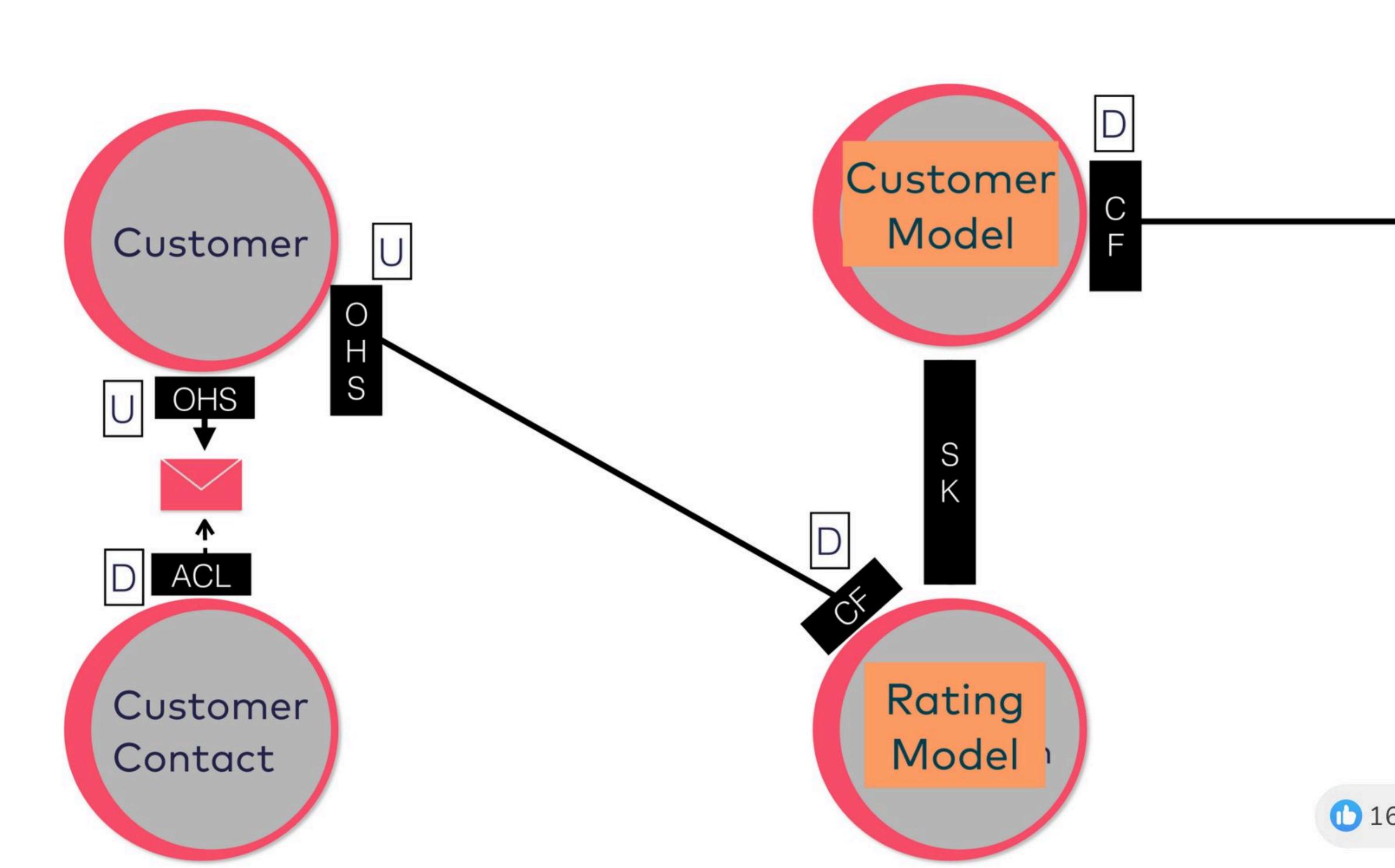






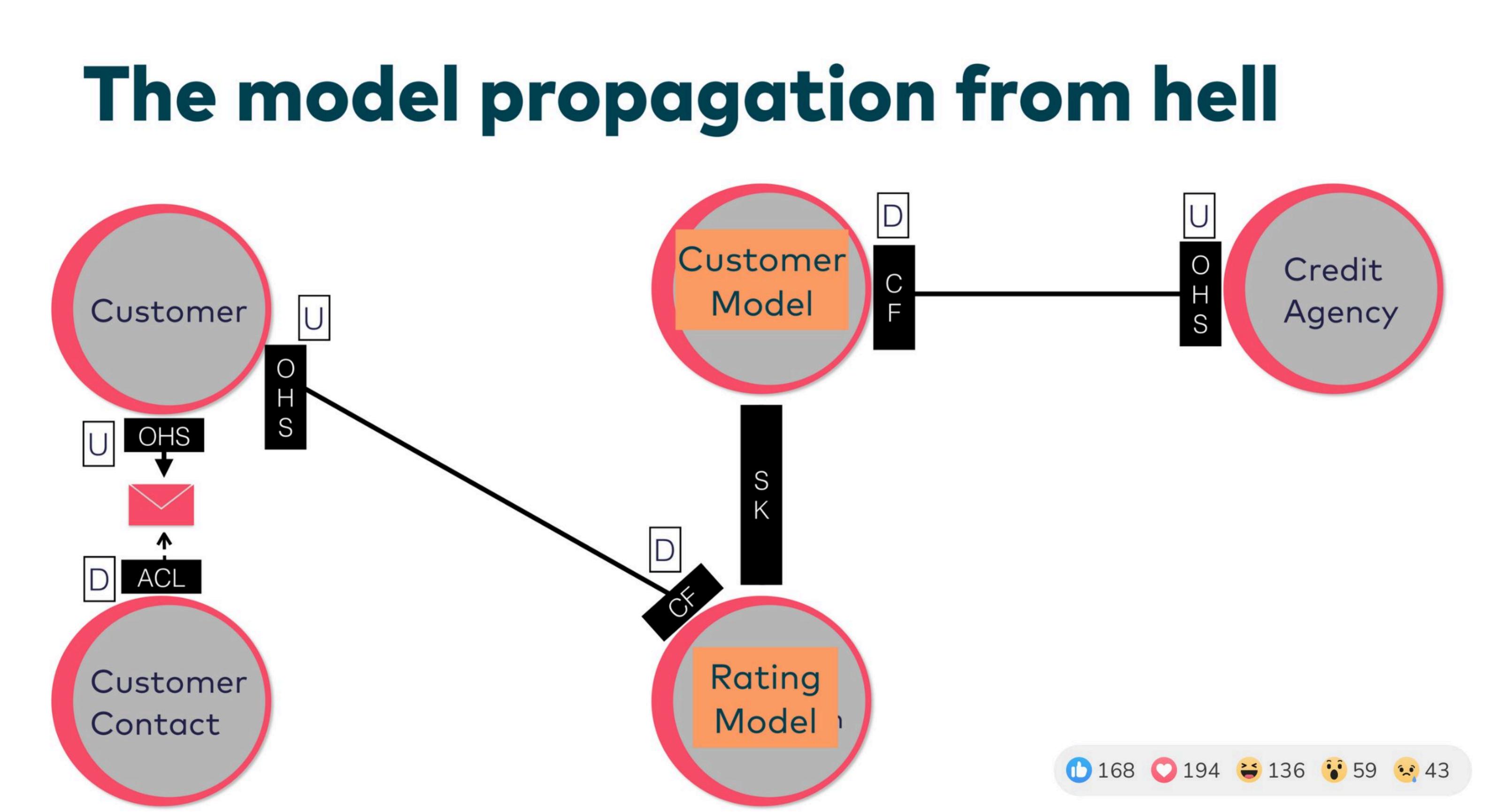












Domain categories can help as well

(Sub)c

Category

ore
domain

- Most important subdomains
- The heart of an organization's business
- Differentiation against competitors









Domain categories can help as well

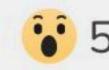
Core (Sub)domain

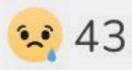
Category

Supporting Subdomain

- Most important subdomains
- The heart of an organization's business
- Differentiation against competitors
- Vital for the operating of core (sub)domains
- Lack of strategic relevance with regards to competition







Domain categories can help as well

(Sub)d

Category

Suppo Subdo

Gen Subdo

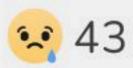
ore Jomain	 Most important subdomains The heart of an organization's b Differentiation against competi
orting omain	 Vital for the operating of core (s Lack of strategic relevance with competition
neric omain	 Needed functionality, which is new problem. "We need some solution of problem.

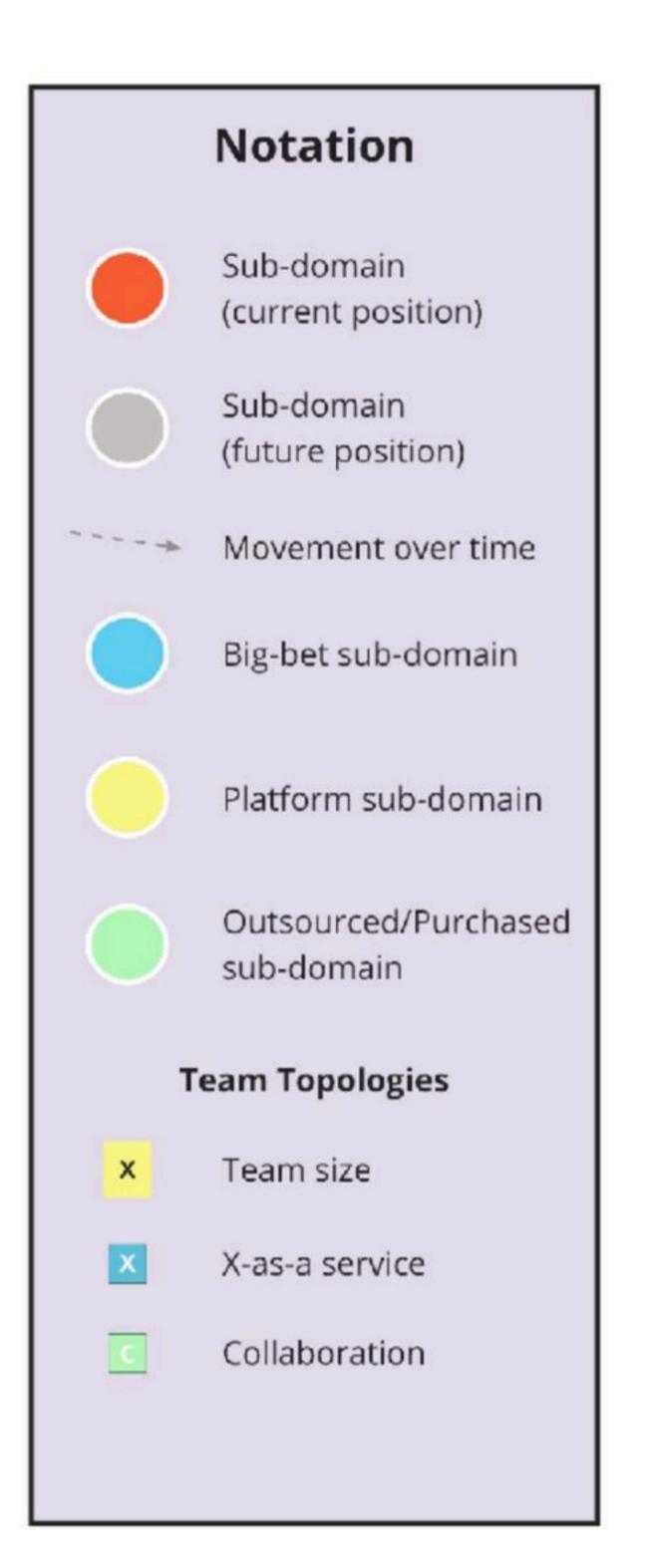


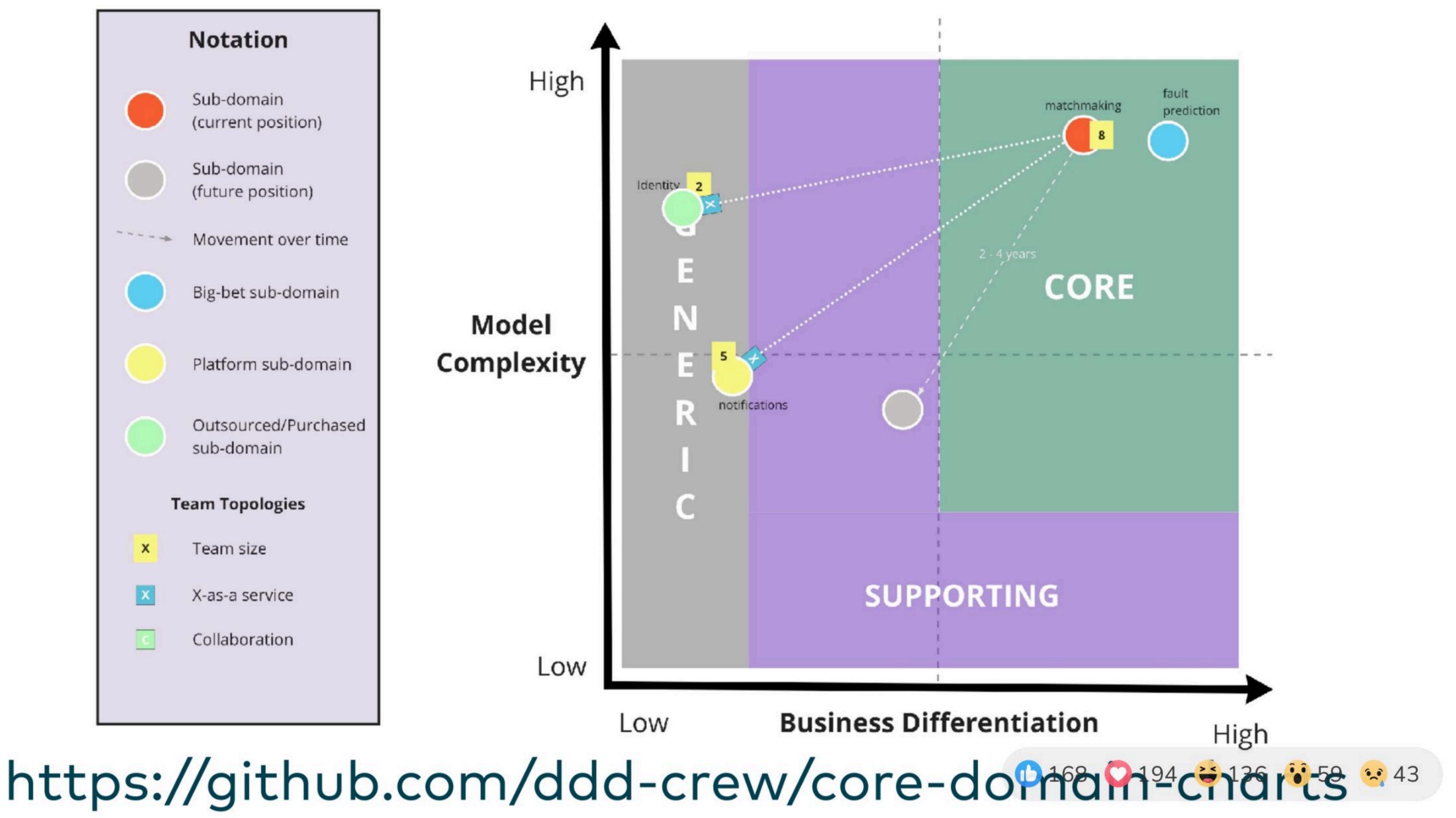
- ousiness itors
- (sub)domains
- regards to

- not critical at all
- olem x"





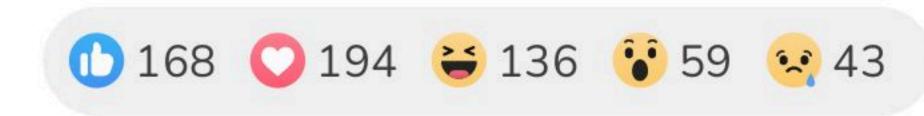




Subdomain categories & Context Maps

Ask those questions

- Should a generic or supporting subdomain be a customer for a supplier in a core domain?
- Should a core domain conform to the model of a generic or supporting subdomain?
- How do we deal with a Big Ball Of Mud in a core domain? Conform to it in the other categories?
- Do we want Partnerships or mutually dependent teams between core and the other subdomains?

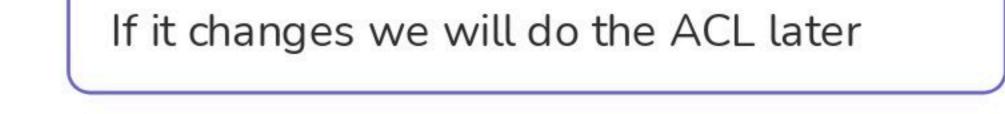




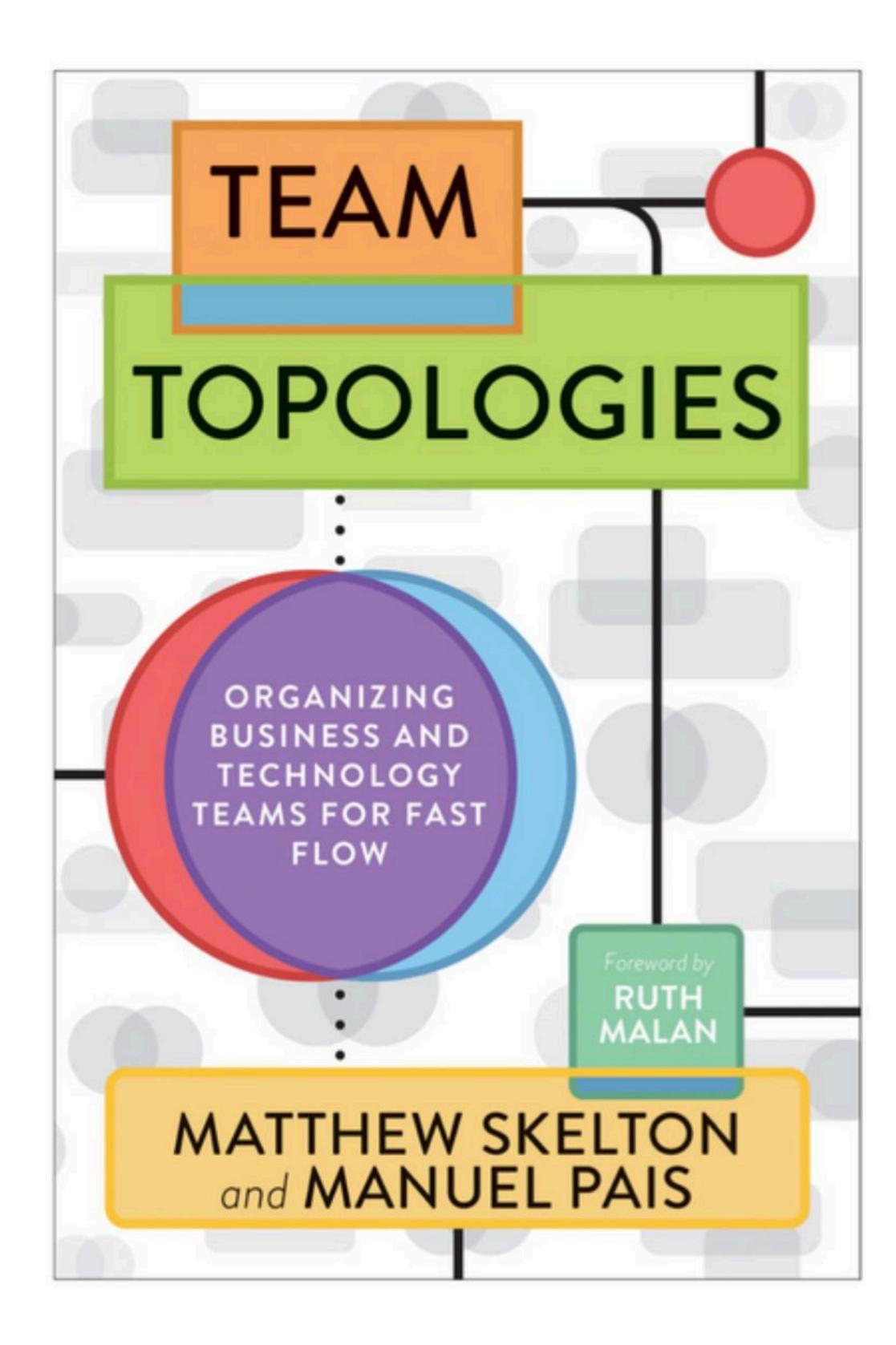
What are your thoughts on a core domain conforming to an external system?

Bad idea	Very dangerous	Hell
Bad Idea	Ew	Bad Idea
Bad idea	Risky	A major risk, which might deemed acceptable, but should be considered.
Risky	That is no core domain then.	Sounds like a bad idea
Not a good idea	Need to implement acl	
Been there it's hell	Other domains bleeding into your own core domain.	Weak position to be in. Bad idea
Risky	Mostly bad	Bad idea
Core should define its own modem	You're not in charge of your business	stupid idea
Grts hard to maintain over time.		
Lot of risk	A lot of changes can come in any time, cannot plan, can get blocked. Not a good	Depends, but generally bad
	idea :D	It depends
It depends	This can make a company fail	Terrible idea
Bye bye core domain! Bad idea!		

Bye bye core domain! Bad idea! StupidMcstupidface	Dependance from the outside World where you have no influence	Totally ok for a young business to prove there is a market
No good idea	Too risky, an ACL would be better. This is bad idea!	Bad Idea, better use acl
Then you don't have a core domain	Depends on external system	There's a risk that our core domain may not be innovative as we think.
If the external system is specialized and well: might be feasible	bad idea, domain should be independent only on itself	It depends but an ACL is recommended
Very bad idea	Hi Mom	You want to take control of core domains
Bad idea	8666	I think, if the external system matches our needs we can use it.







Team Interaction Modes

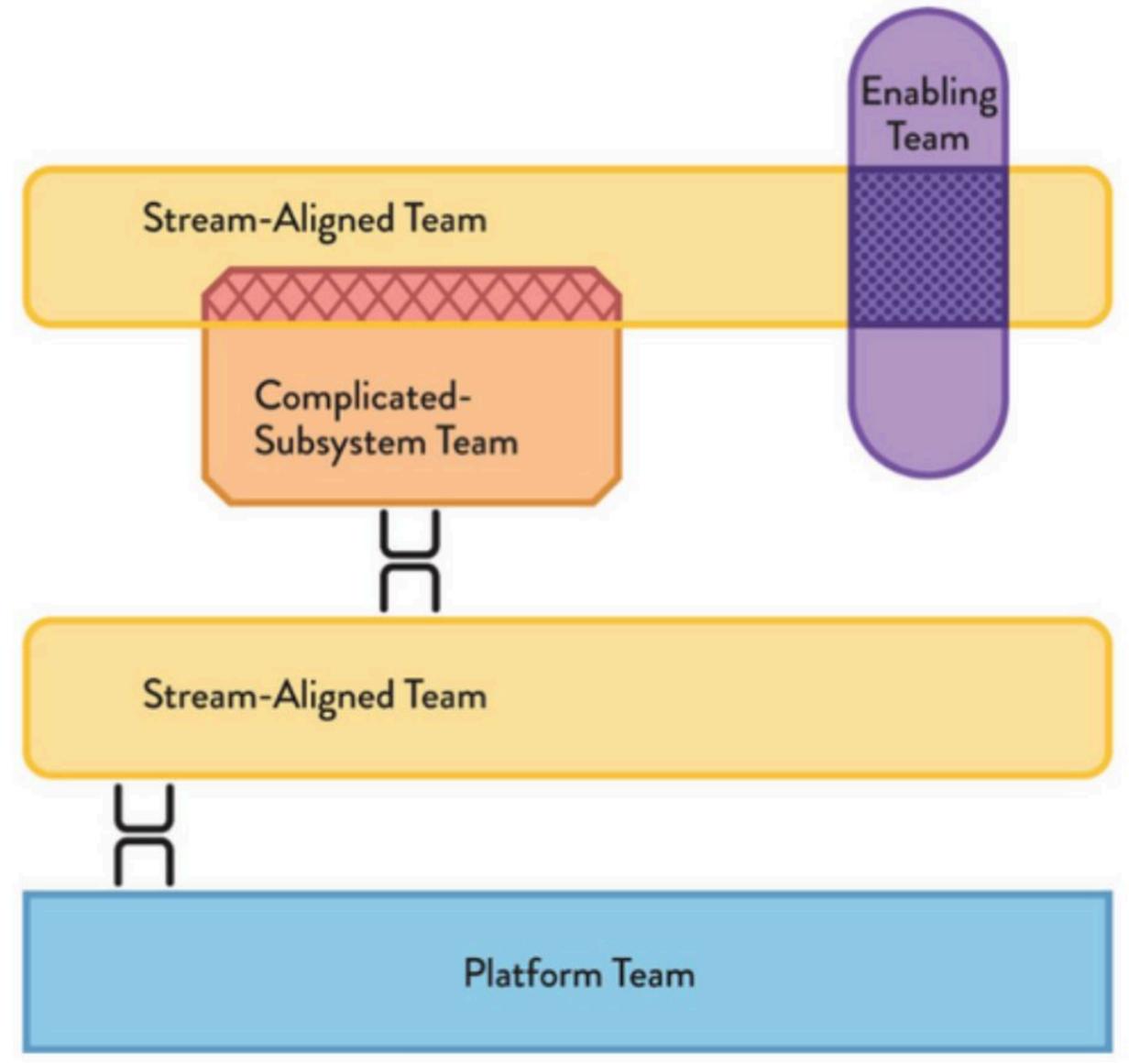
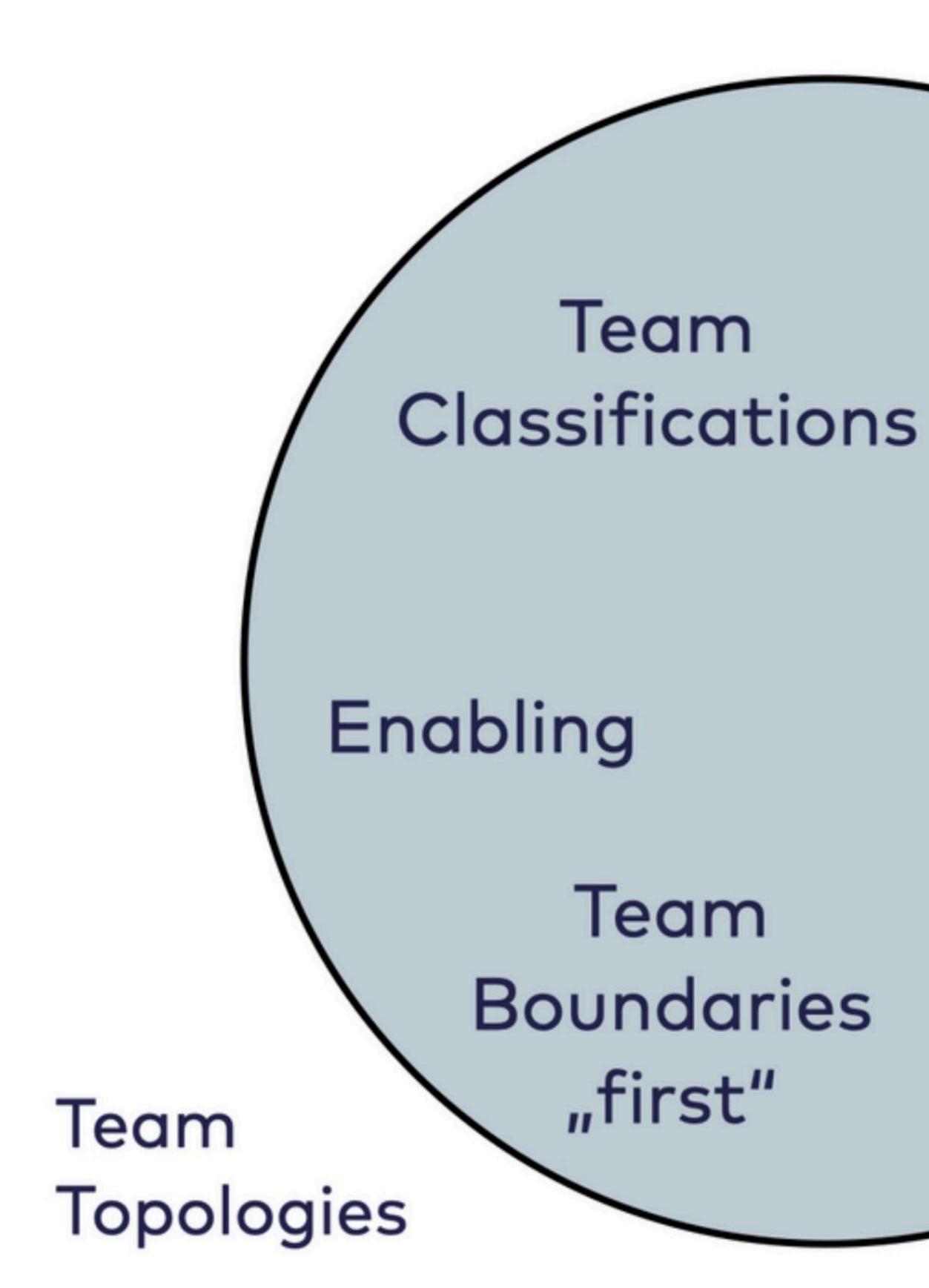


Image taken from the Palen Poplato to the start of the st



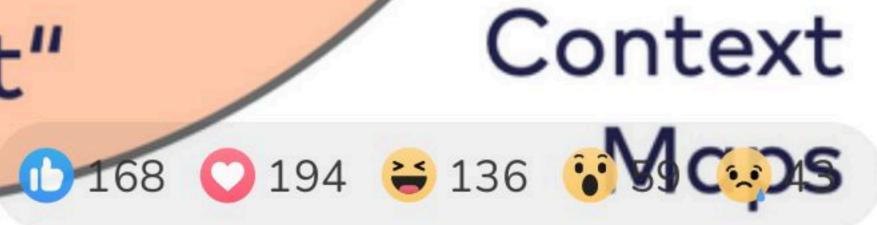


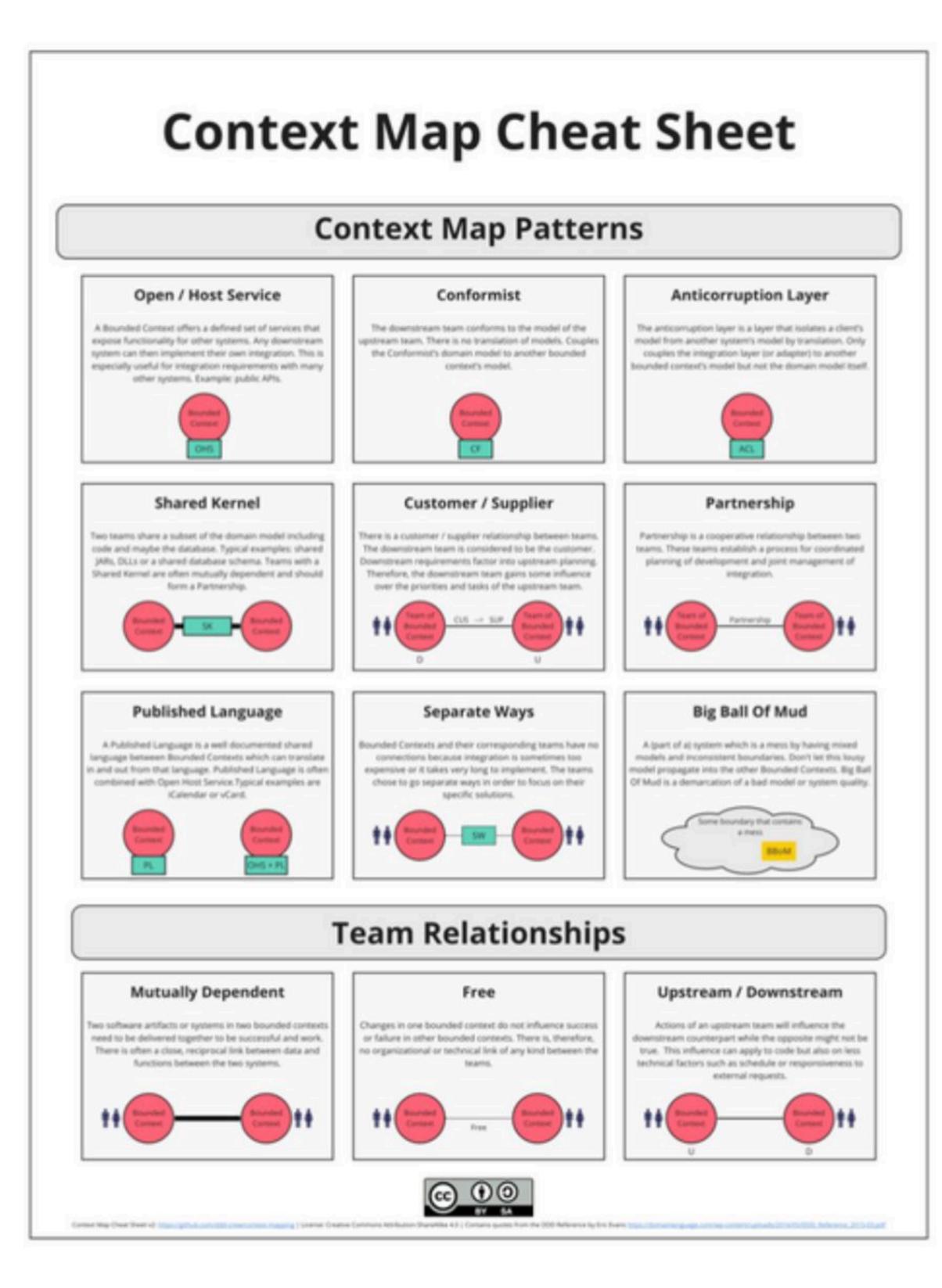




Domain Models Governance Teams Team Collabor-Dependencies ation "Organizational Services & Solutions" Interfaces Context Boundaries "first"





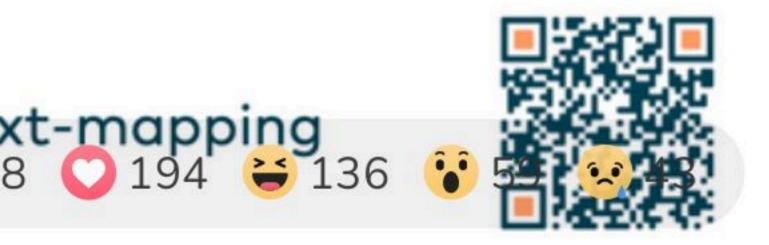


Check out DDD **Crew on GitHub**

- 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 168 194 136





Hands On DOMAIN-DRIVEN DESIGN by example

Michael Plöd

Get my DDD book cheaper



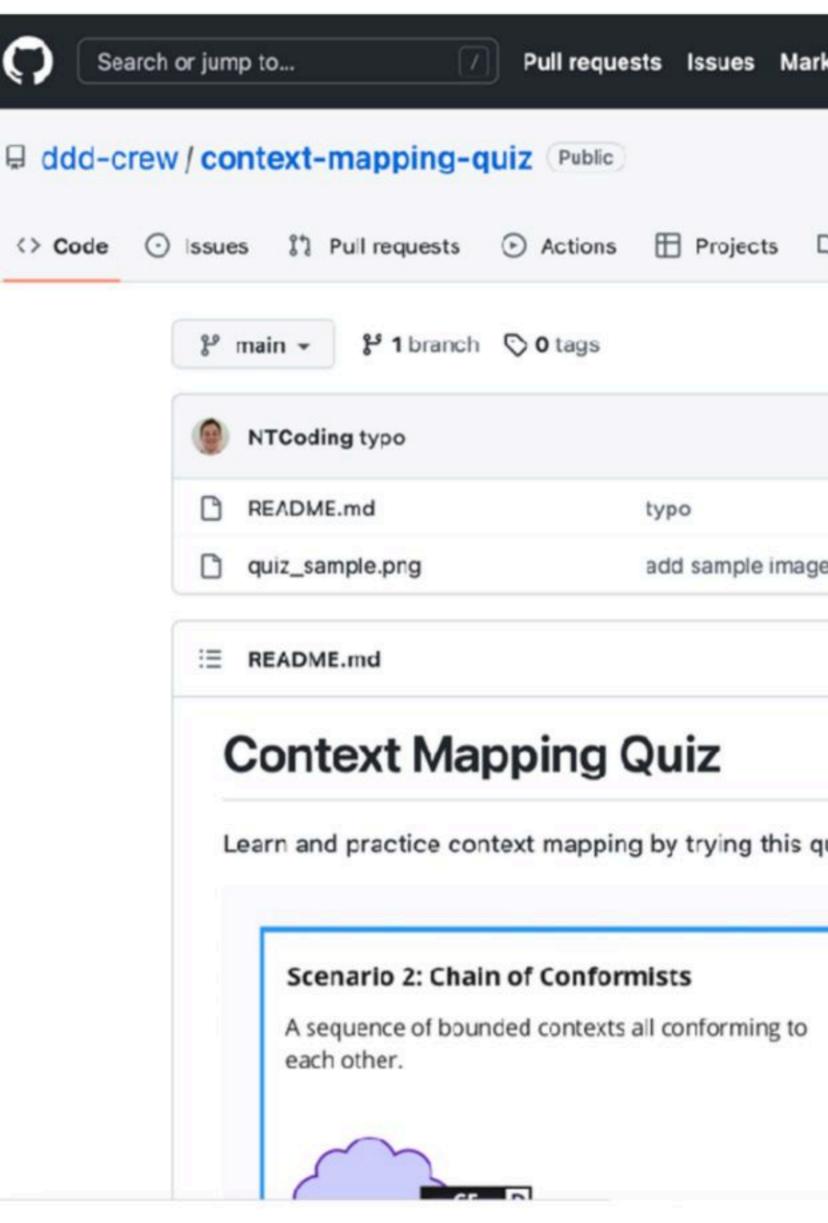
Book Voucher: 7.99 instead of (min) 9.99 http://leanpub.com/ddd-by-example/c/speakerdeck



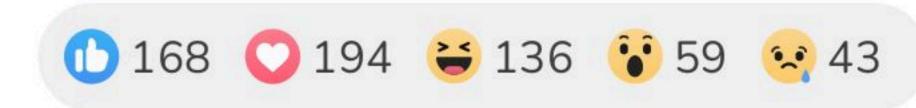


- Miro based quiz game on various scenarios
- Creative Commons





https://github.com/ddd-crew/context-mapping-guiz



ketplace	Explore
卭 Wiki	③ Security // Insights
	Go to file Add file - Code -
	f356265 on 2 Sep 2021 🛈 5 commits
	10 months ago
e	10 months ago
	Ø
uiz alone	or with your colleagues.
	What are your thoughts on this relationship? Are there any relationships that are better suited here and why?

Thanks! Guestions?



Michael Plöd michael.ploed@innoq.com

Follow me on Twitter: @bitboss

innoQ Deutschland GmbH

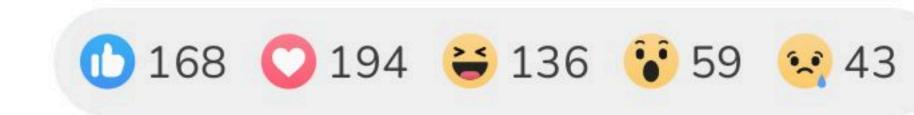
Krischerstr. 100 40789 Monheim +49 2173 3366-0 Ohlauer Str. 43 10999 Berlin

Ludwigstr. 180E 63067 Offenbach



Kreuzstr. 16 80331 München Hermannstrasse 13 20095 Hamburg

Erftstr. 15-17 50672 Köln





Königstorgraben 11 90402 Nürnberg