

Growing Your Search Team

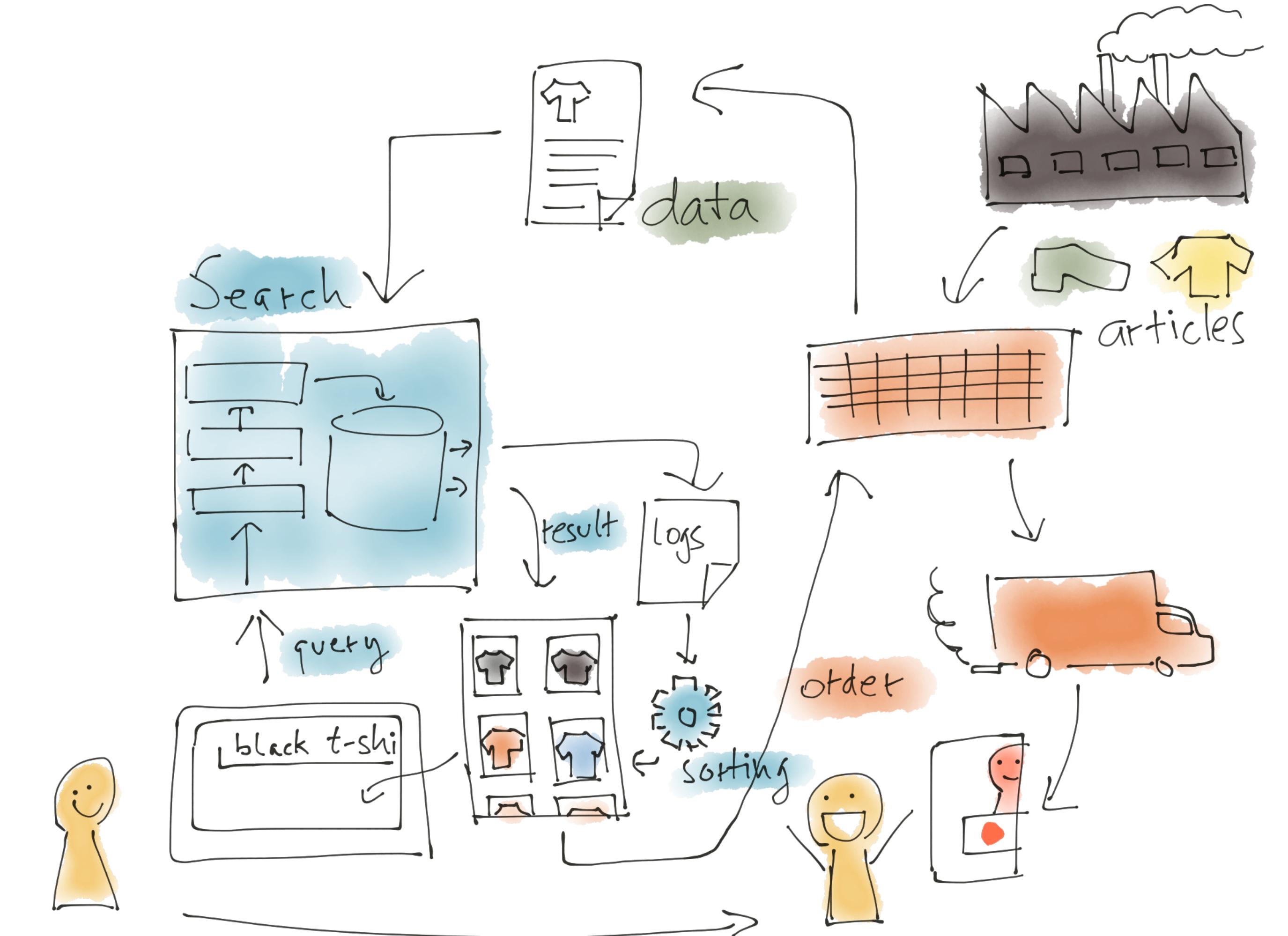
Mikio L. Braun

Mikio L. Braun, MICES 2022, June 15, 2022



Search Systems Are Complex

- Intake
- Enrichment
- Search Engine
- Ranking
- Query Understanding
- Auto-Suggest
- Synonyms, Spell Checking, ...



Why Grow A Team

- Be faster?
- Have a more robust platform?
- Work on more things in parallel?
- Because you can?

... but how?

How Not To Do It

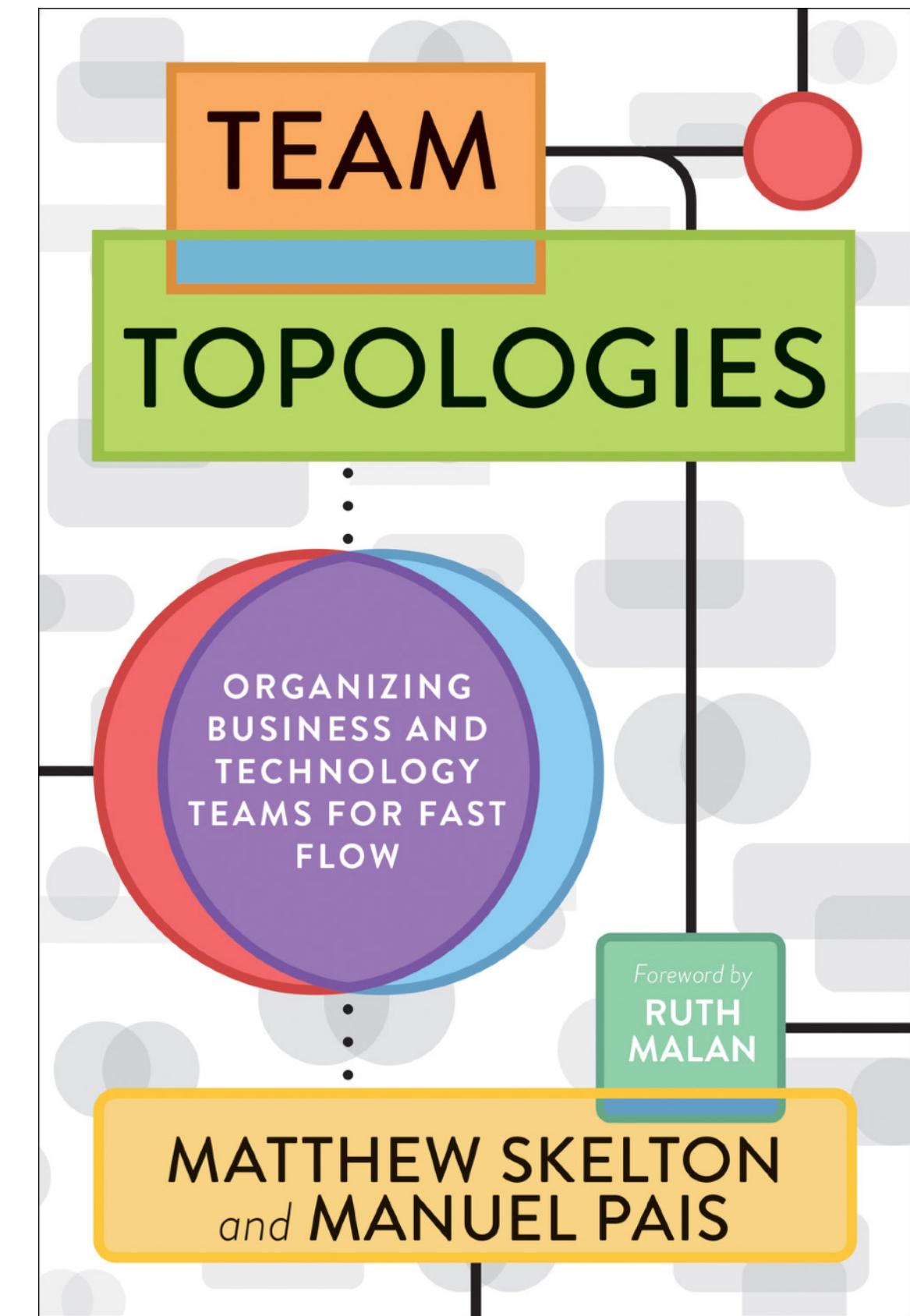
- Query Understanding Team
- Spell Checking Team
- Search Engine Team
- Intake Team
- ...

Anything you do becomes a
multi-team project 

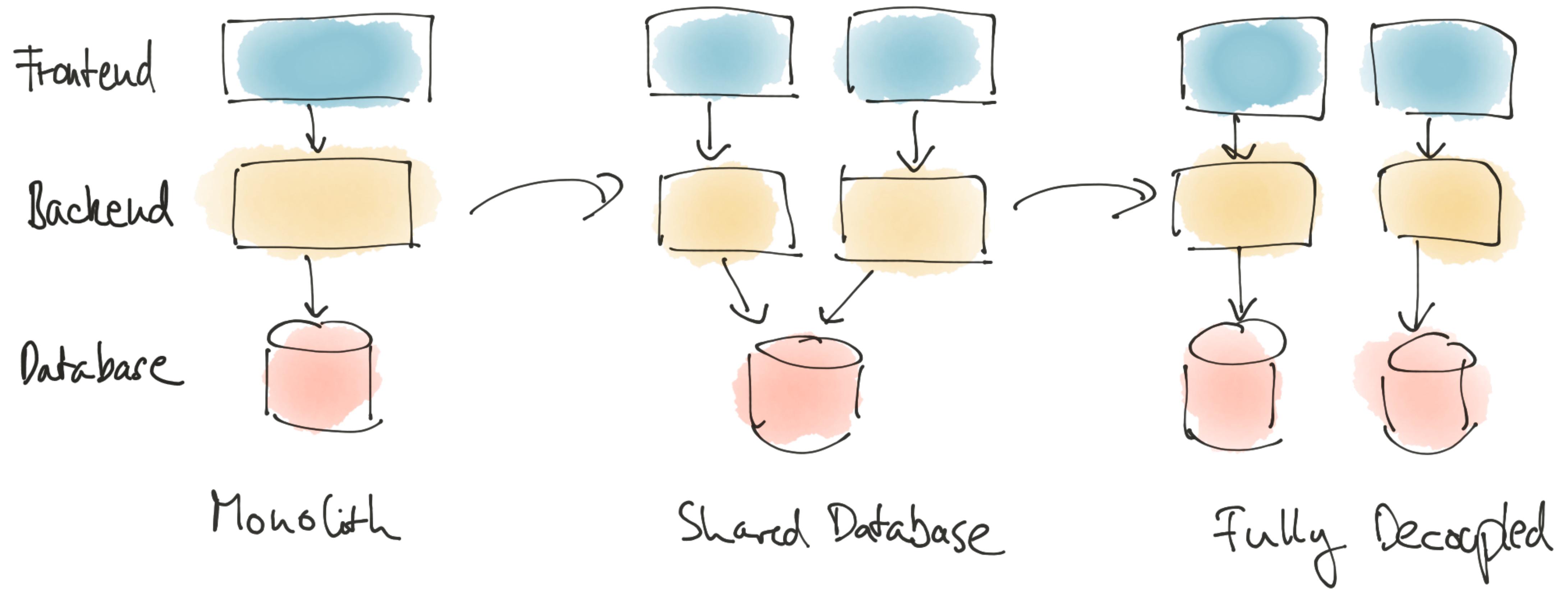
Team Topologies: Optimizing Teams For “Flow”

- Most of the work in a team should be possible independently.
- “Stream-aligned team”
- More than feature teams

<https://teamtopologies.com>

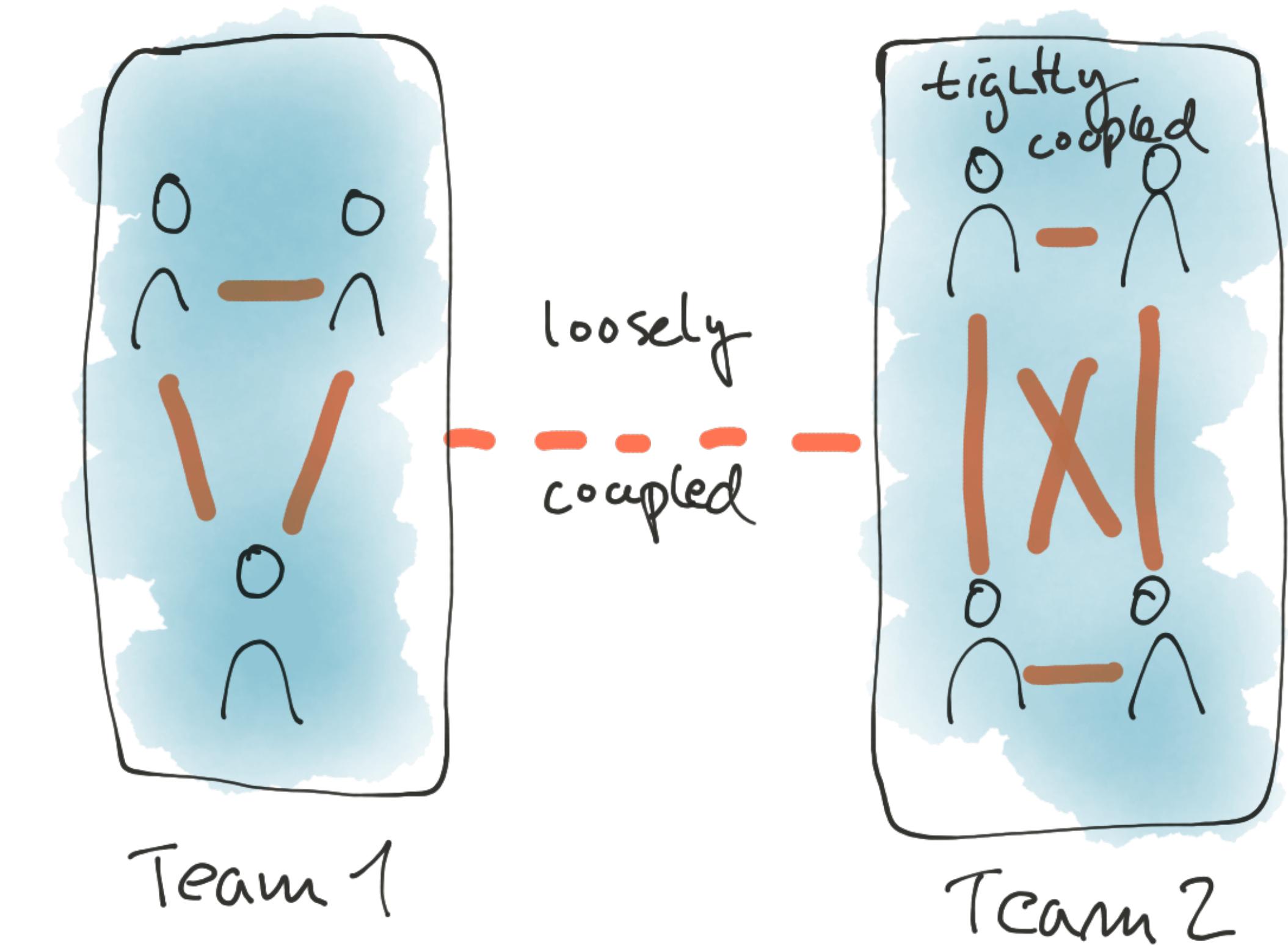


Decoupling Teams

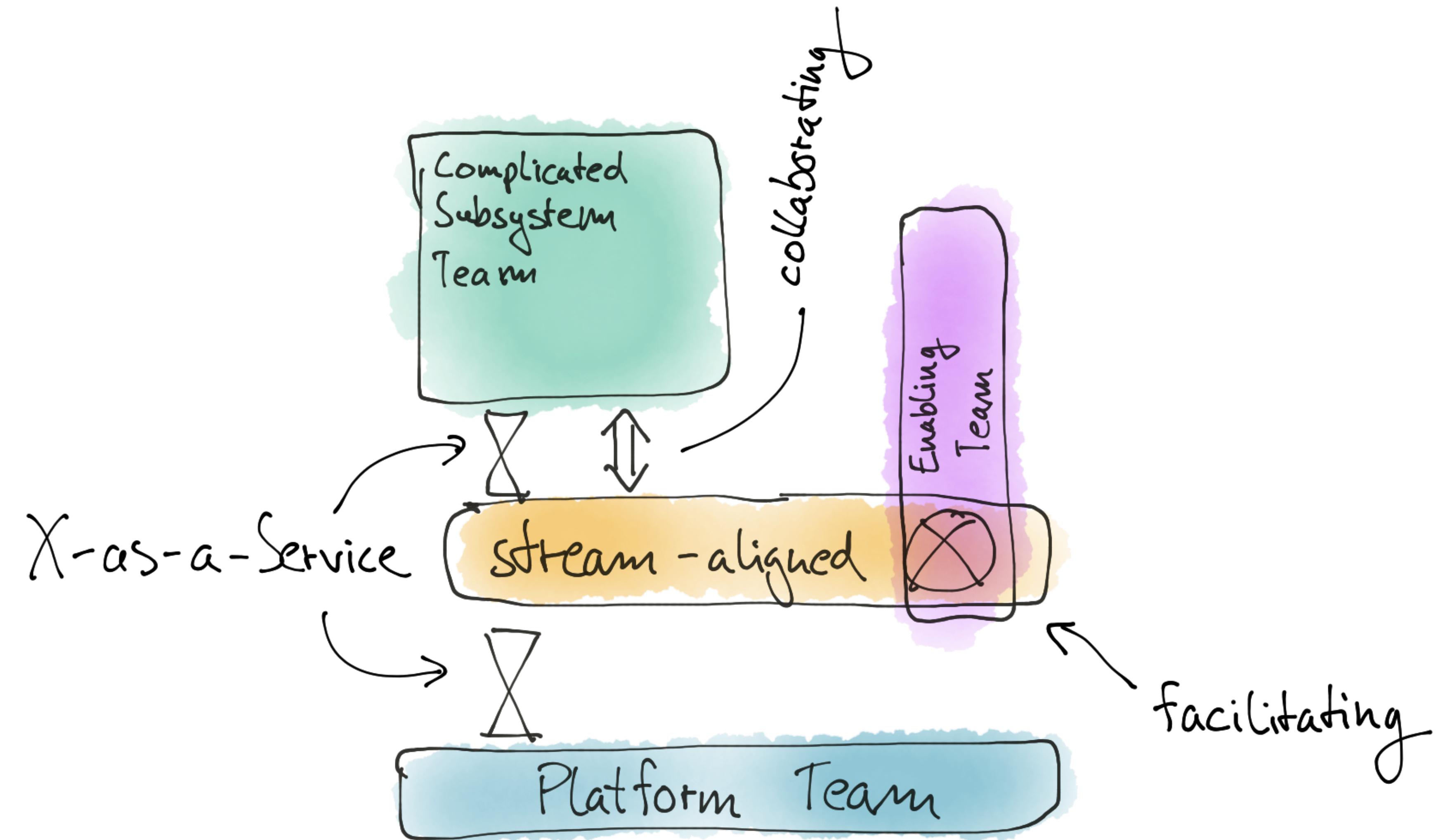


Cognitive Load and Communication Patterns

- Cognitive Load 😤
 - Technologies 💻⚙️
 - Number of people (🍕🍕 rule)
 - Communication Patterns



Team Topologies and Interaction Patterns



The Four Team Types

Stream-aligned Team

- Steady flow of feature delivery
- No hands-off to other teams
- Quickly react to feedback

Enabling Team

- “Internal consultants”
- Help teams with technology/topics they need to learn
- Tracks state-of-the-art in their area of expertise before other teams need it

Platform Team

- Creates self-service products other teams can use
- Focus on reliability and usability
- Proactively looks out for the needs of other teams

Complicated Subsystem Team

- “Internal agency”
- Reduce cognitive load for stream-aligned teams.
- Correctly prioritizes and delivers upcoming work based on the needs of other teams

Fracture Planes

- Business Domain Bounded Context
- Regulatory Compliance
- Change Cadence
- Team Location
- Risk
- Performance Isolation
- Technology
- User Personas

Team Topologies for Search

- No one size fits all.
- It really depends on your “stream of work.”

What are the things you want to focus on in the next 12 months?

Typical Topics That Can Become A Stream of Work

- Full text search
- Auto suggest
- Ranking
- Faceting/Filtering
- Item selection
- Size
- Availability
- ...

Checklist

- Can you technically put everything in one team?
- Can you measure impact independently?
(e.g. ranking:
 - sorting/ranking deeply intertwined with search engine
 - Impact may be measured, but other activities might distort impact)
- (e.g. filtering:
 - needs to work with front end as well as intake)

Support Structures

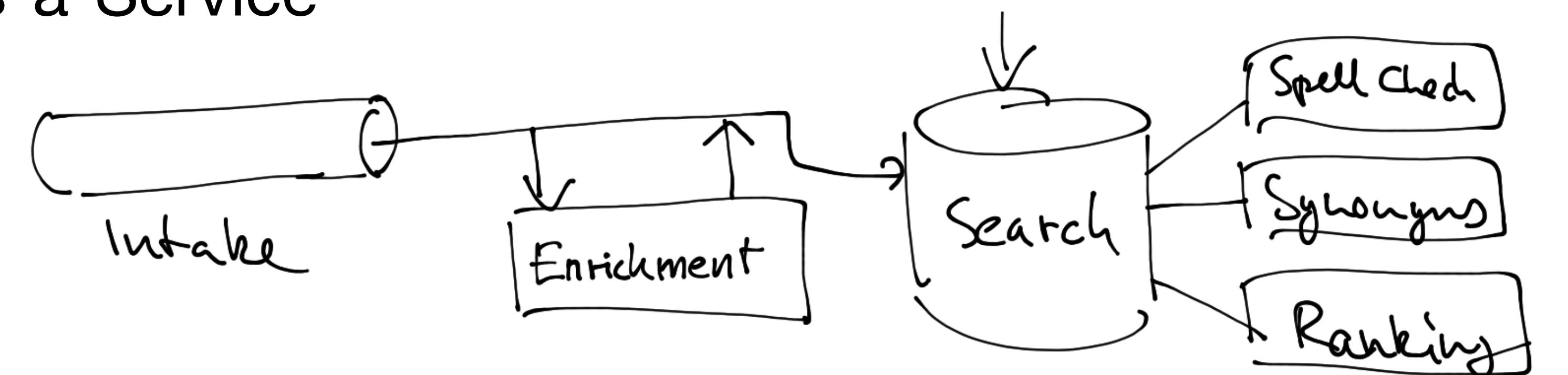
- Candidates for Platform/Complex Subsystem:
 - Frontend
 - Search Engine
 - Intake
- Candidates for Enabling:
 - Data Science, search relevance optimization

Or alternatively, have teams manage/learn that themselves!

Also: building platforms is hard!

Search Engine Platform

- Self-service search engine (incl. deployment, scaling, intake, ...)
- Make everything configurable (intake, fields, analyzers, sorting, ...)
- Platform team is there to make it robust, improve usability, keep people from shooting themselves in the foot.
- Much more than “Elastic-as-a-Service”



Replacing the Search Engine with Microservices

- Solr/Elastic is so complicated, let's pull *query processing/ranking/spell checking* out.
- You gain more *flexibility* deploying parts independently.
- You'll need to literally *reinvent the wheel* for the preprocessing.
- Esp. in query processing, mistakes are *hard to fix* later.

How to Change

- **Change interaction patterns** before making technical changes
- Close collaboration => lesser collaboration => X-as-a-service
- Never ending process
- Look out for **triggers**:
 - Software has grown too large for one team
 - Speed is becoming slower
 - Multiple business services rely on a large set of underlying services

Summary

- Search is particularly complex to scale because the technology is complex and intertwined.
- Team topologies is a great framework for decoupling teams and have clear focus for teams.
- Identify main streams of work.
- Think of proper supporting teams (platform, enabling, complicated subsystem).
- Never ending process.