The Microservices journey from a startup perspective

Susanne Kaiser
CTO
@suksr

Just Software
@JustSocialApps

Each journey is different

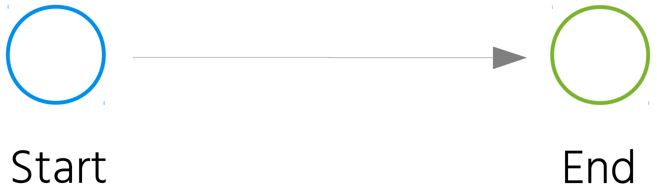
“People try to copy Netflix,
but they can only copy what they see.
They copy the results, not the process.”

Adrian Cockcroft, AWS VP Cloud Archtitect,
former Netflix Chief Cloud Architect

Our Transformation Process

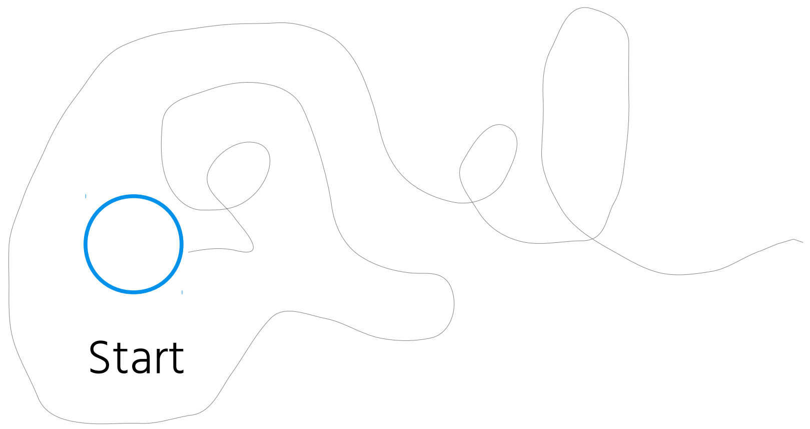
- Identify candidates
- Decompose candidates
- Establish Microservices ecosystem

Transformation process

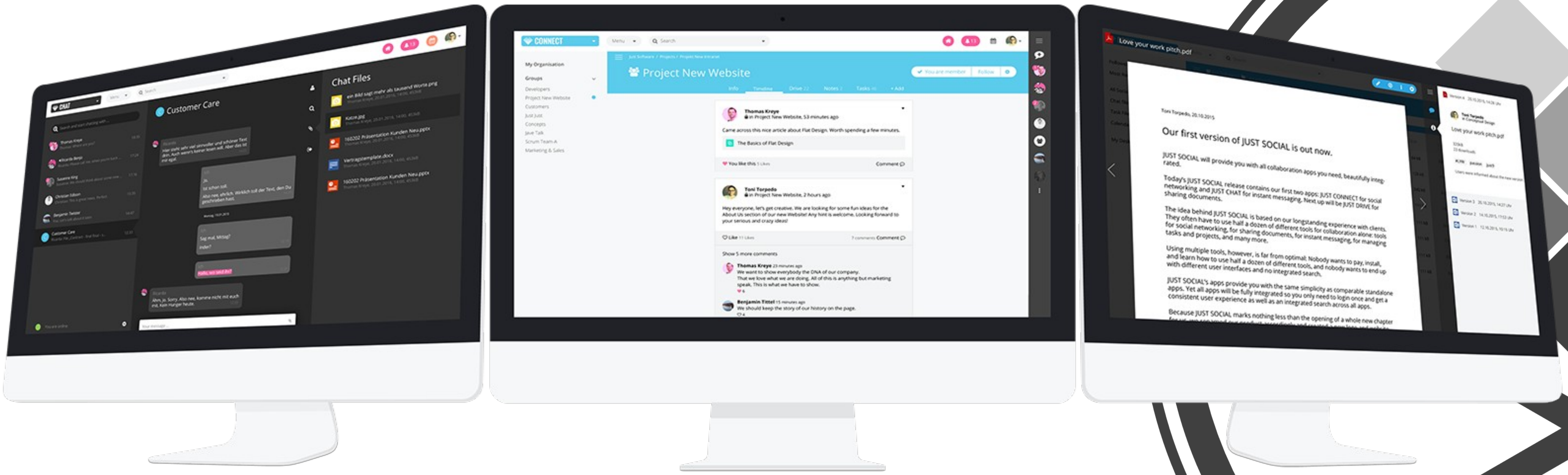


Transformation process

Theory (straightforward)



Reality (evolutionary)



Thomas Kreye
in Project New Website, 53 minutes ago
Came across this nice article about Flat Design. Worth spending a few minutes.
The Basics of Flat Design
You like this
Comment

Toni Torpedo
in Project New Website, 2 hours ago
Hey everyone, let's get creative. We are looking for some fun ideas for the About Us section of our new Website! Any hint is welcome. Looking forward to your serious and crazy ideas!
Like
Comment
Show 5 more comments
Thomas Kreye 23 minutes ago
We want to show everybody the DNA of our company.
That we love what we are doing. All of this is anything but marketing speak. This is what we have to show.
Benjamin Tittel 15 minutes ago
We should keep the story of our history on the page.

Toni Torpedo, 20.10.2015

Our first version of JUST SOCIAL is out now.

JUST SOCIAL will provide you with all collaboration apps you need, beautifully integ-rated.

Today's JUST SOCIAL release contains our first two apps: JUST CONNECT for social networking and JUST CHAT for instant messaging. Next up will be JUST DRIVE for sharing documents.

The idea behind JUST SOCIAL is based on our longstanding experience with clients. They often have to use half a dozen of different tools for collaboration alone: tools for social networking, for sharing documents, for instant messaging, for managing tasks and projects, and many more.

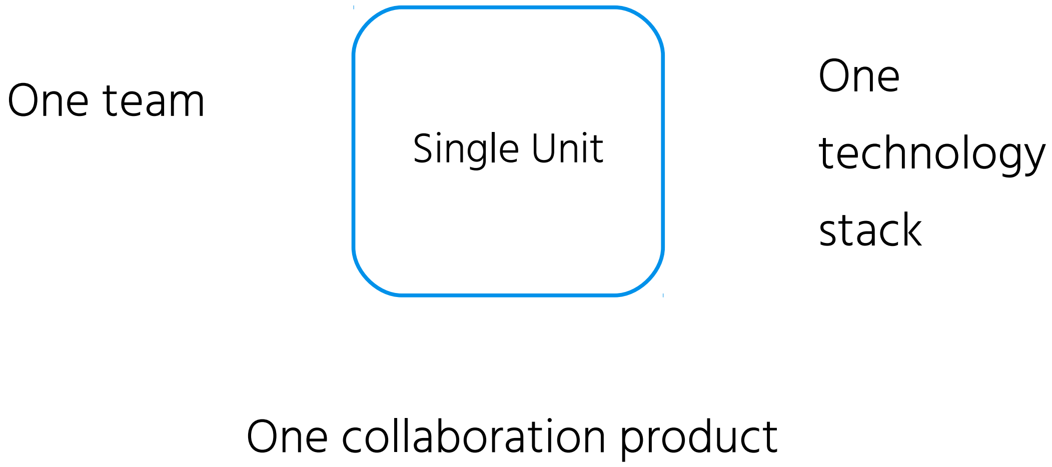
Using multiple tools, however, is far from optimal: Nobody wants to pay, install, and learn how to use half a dozen of different tools, and nobody wants to end up with different user interfaces and no integrated search.

JUST SOCIAL's apps provide you with the same simplicity as comparable standalone apps. Yet all apps will be fully integrated so you only need to login once and get a consistent user experience as well as an integrated search across all apps.

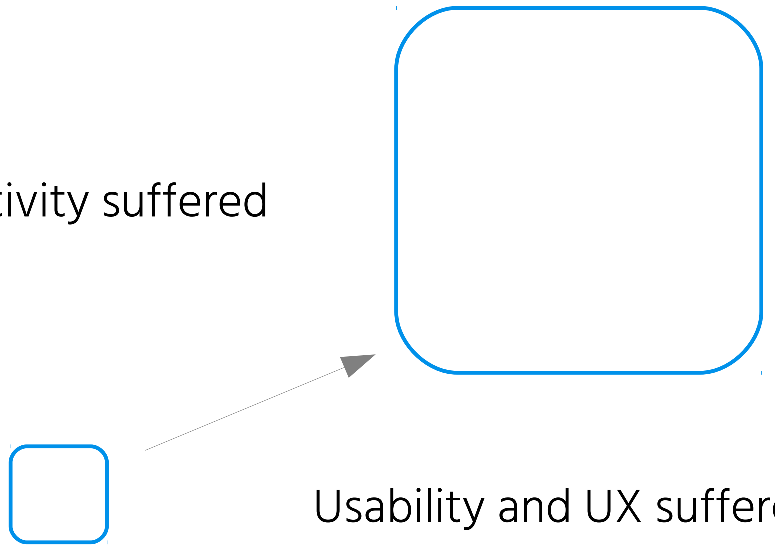
Because JUST SOCIAL marks nothing less than the opening of a whole new chapter

JUST SOCIAL

The beginning … A monolith in every aspect



After an evolving while …



Productivity suffered

New features
released slowly

Usability and UX suffered

Separate Collaboration Apps

JUST PAGE
Social Network

JUST CONNECT
Real-time collaboration

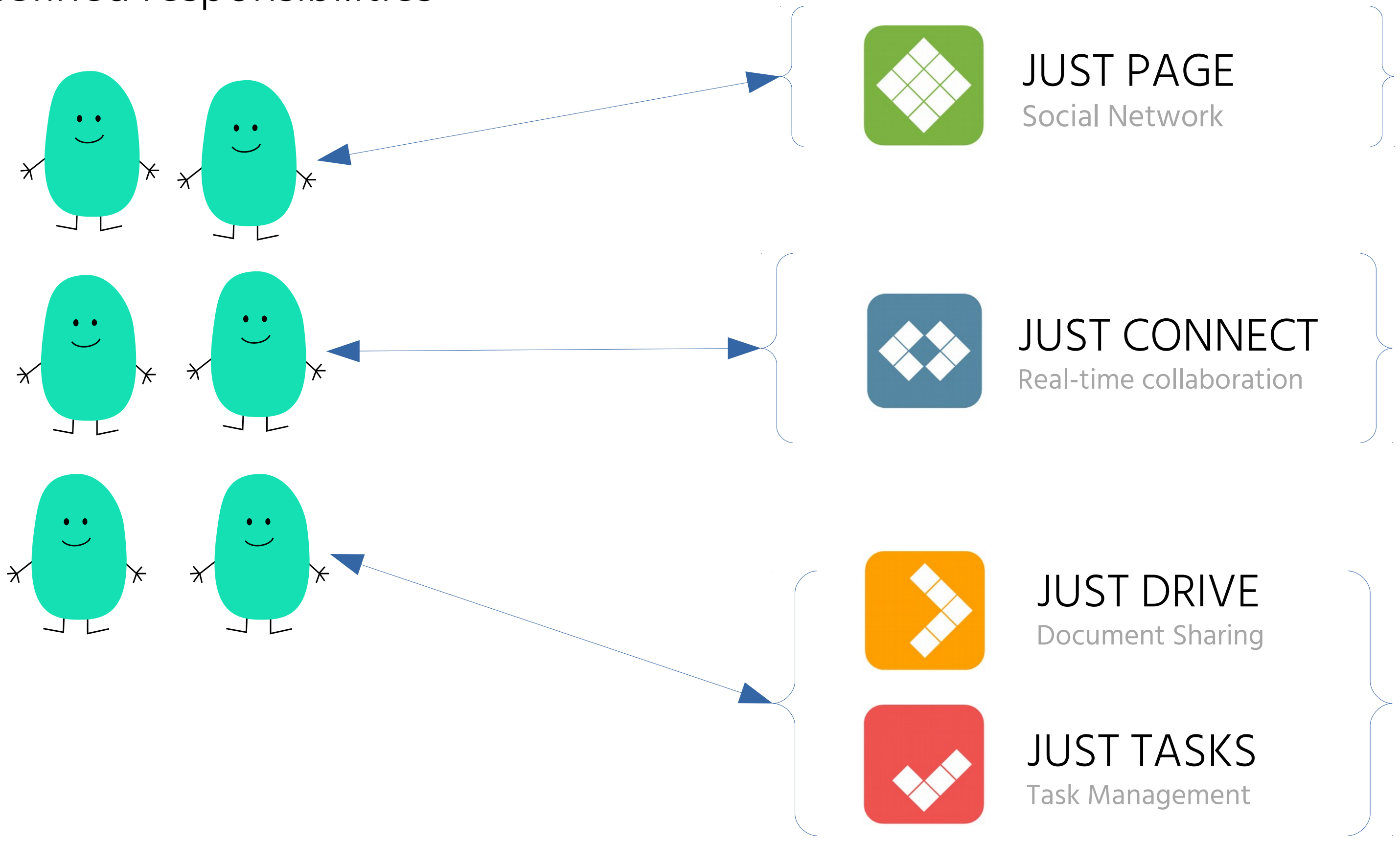
JUST TASKS
Task Management

JUST DRIVE
Document Sharing

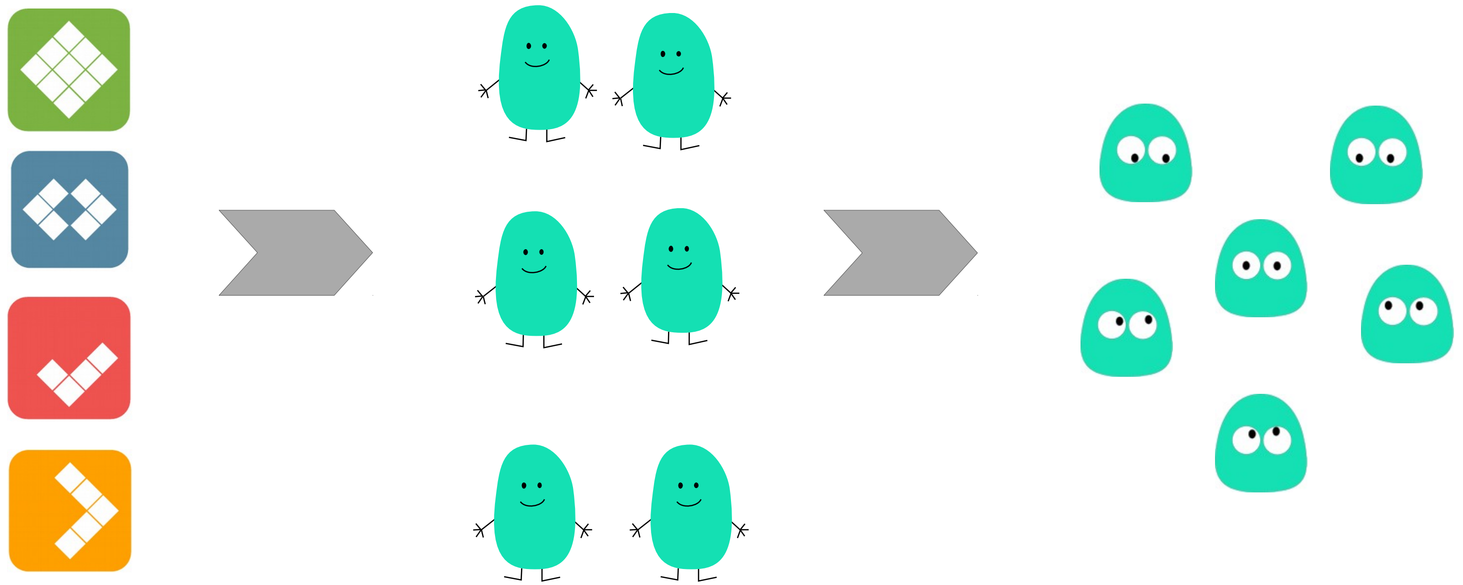
JUST SOCIAL

Small, autonomous teams

with well-defined responsibilities



In the long run …

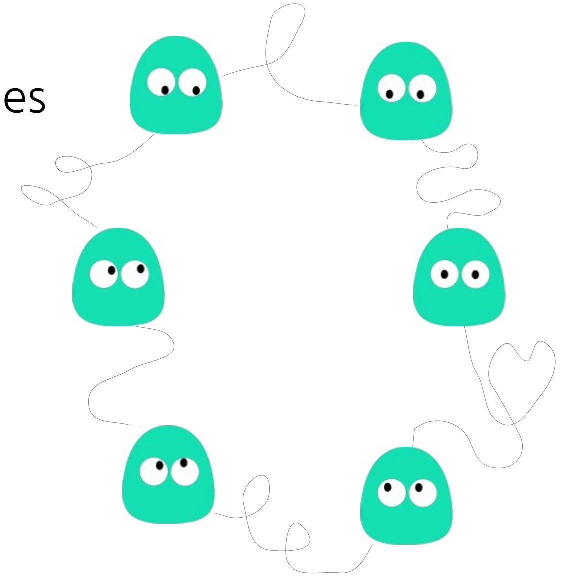


Microservices come with complexities

Multiple independent services

Slow, unreliable network

Partitioned data



Operational complexity

Communication complexity

Complexity of eventual consistency

Challenges of transformation

Different skills & tools required

You still have to take care of your existing system

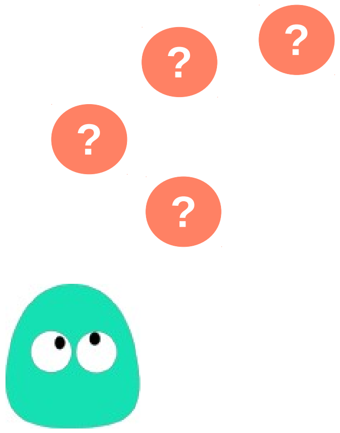
Core functionality is hard to untangle

Transformation takes longer than anticipated

Our Motivation

- Product and organizational/culture driven
- Enabling autonomous teams
  with well-defined responsibilities
- Develop and deploy independently
  to release changes quickly

How to start?

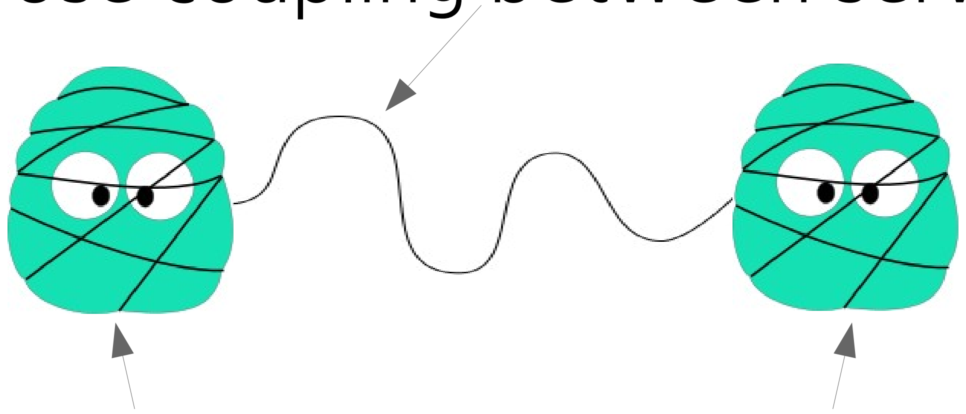


Transformation process

⬡ Identify candidates

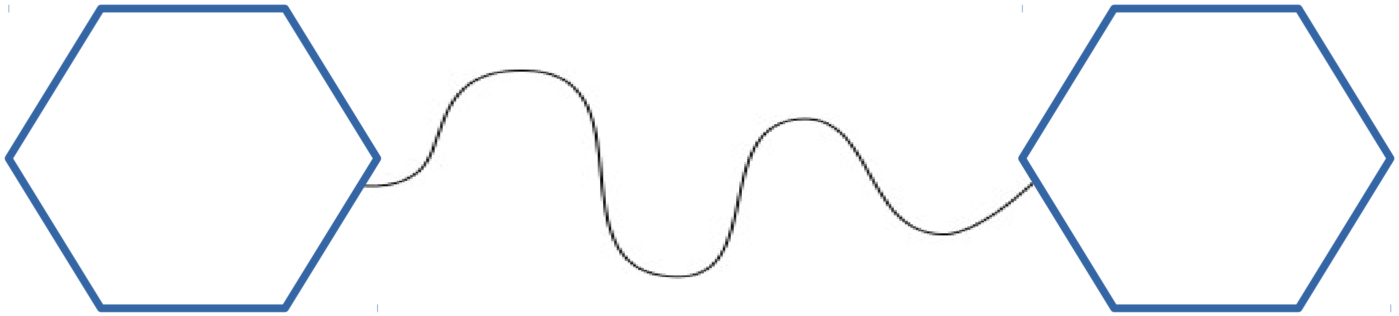
Key concepts of modelling Microservices

Loose coupling between services



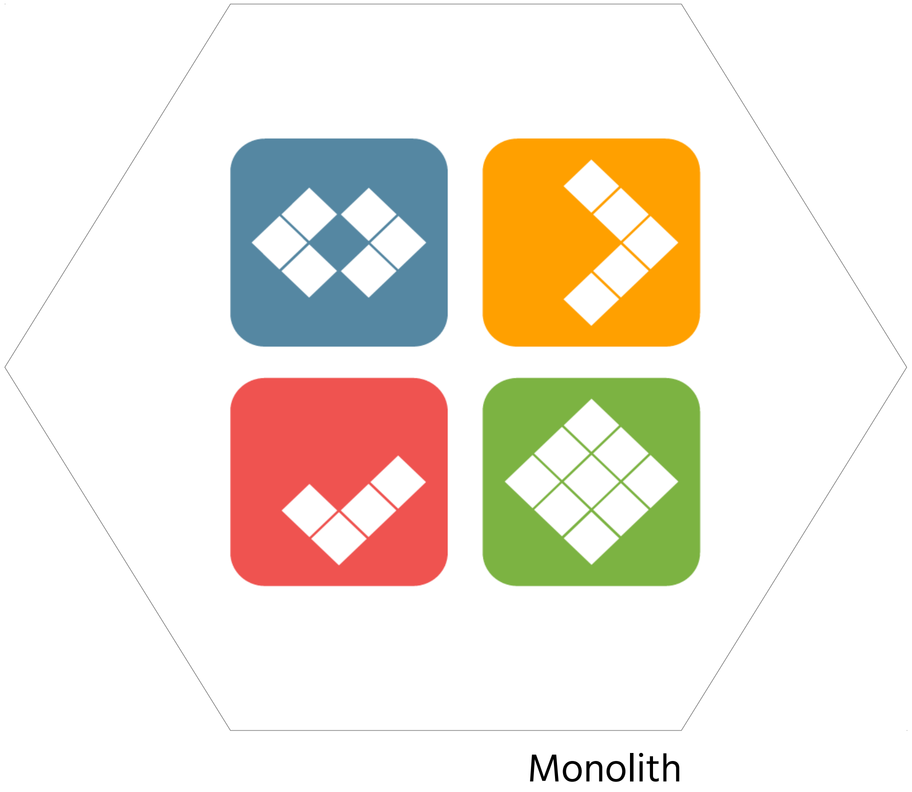
High cohesion within a service

Identify Bounded Contexts



Well defined business function

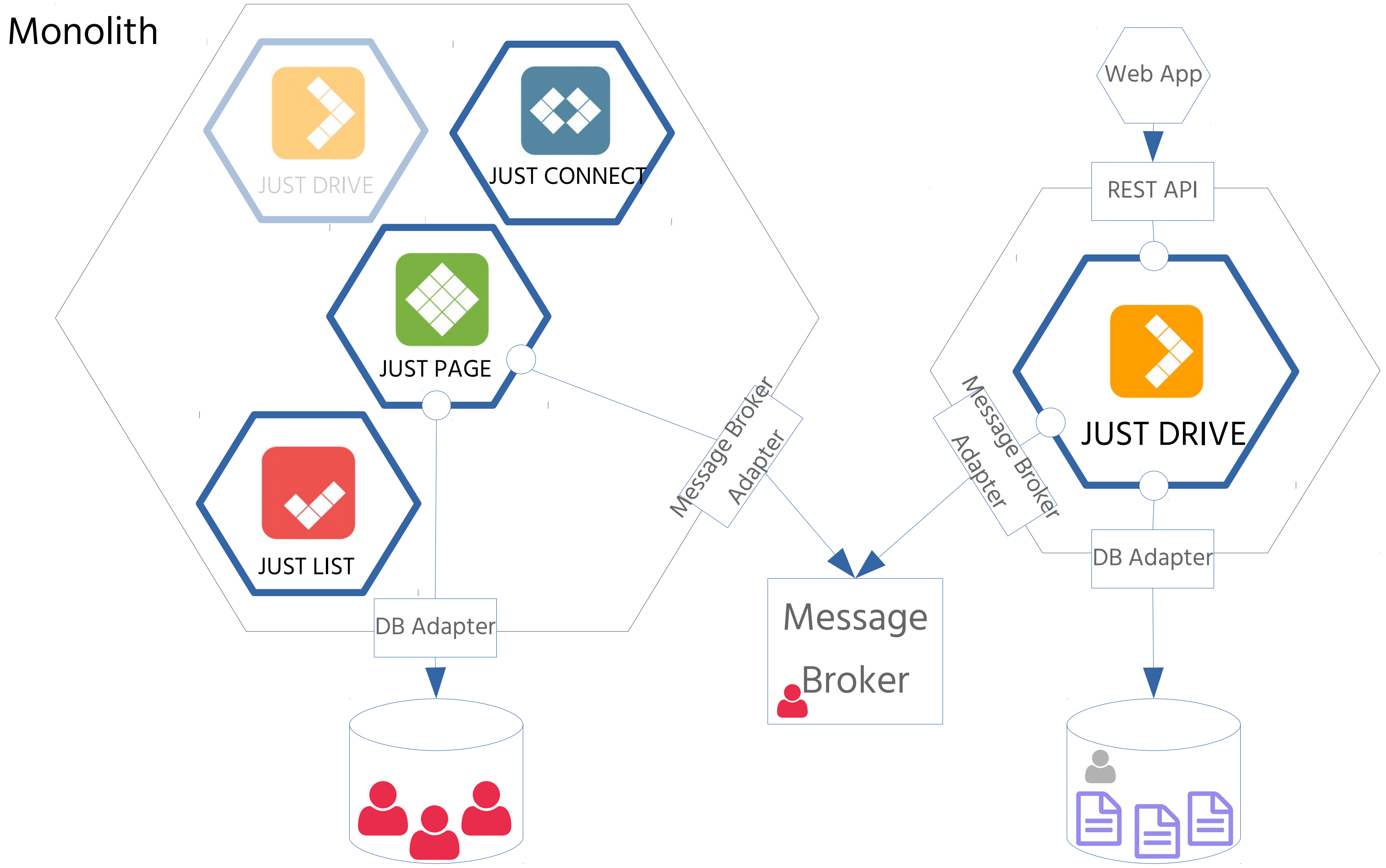
Bounded Contexts = Collaboration Apps



Transformation process

Decompose candidates

First approach as a co-existing service



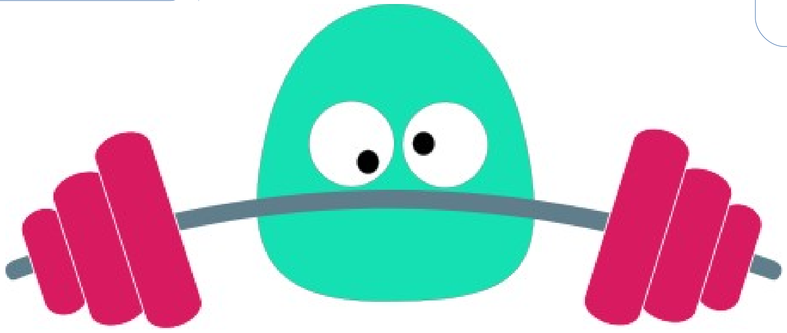
Hard work if you do all at once

New UI

Maintain & run current system

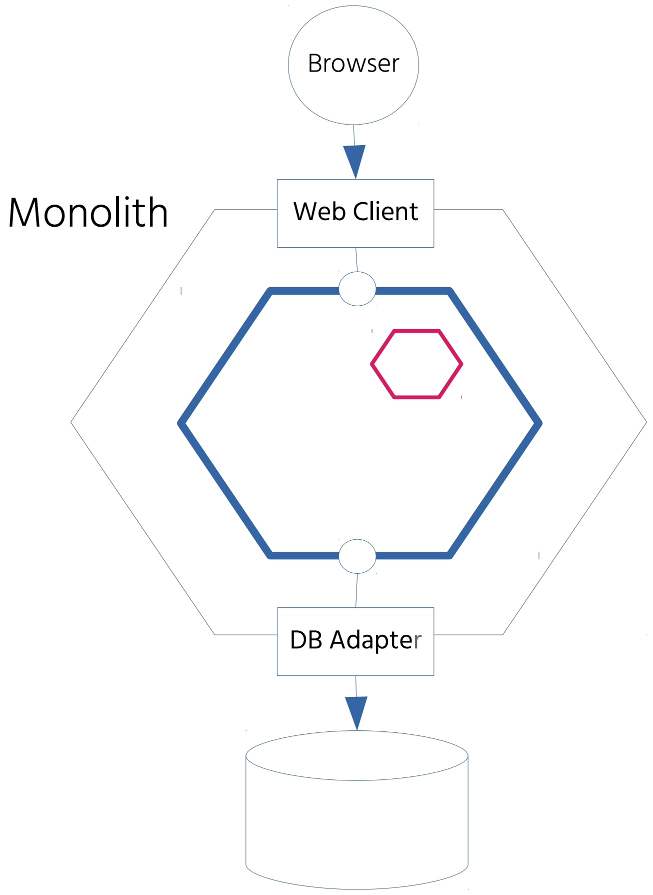
New data structure

More features

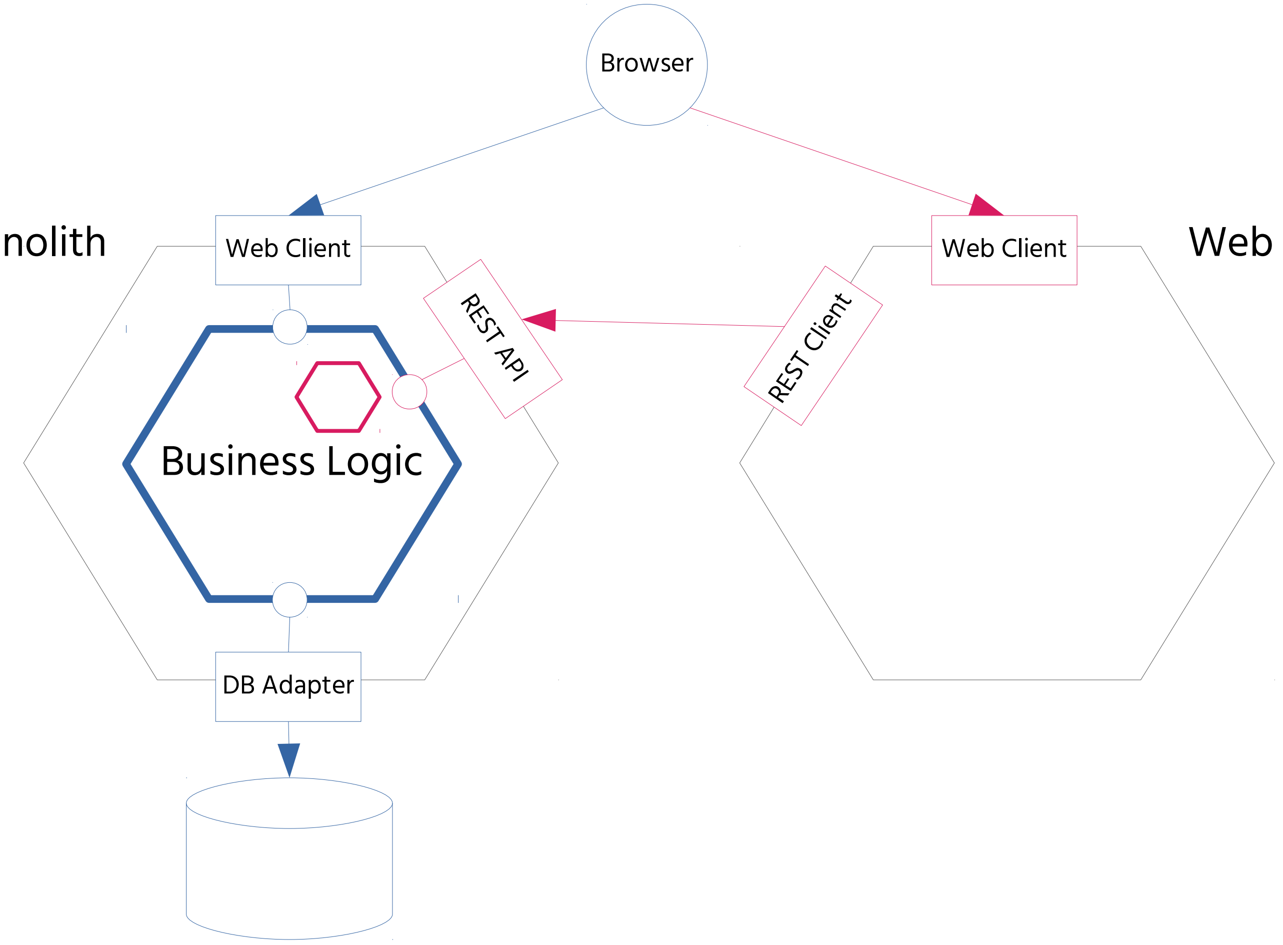


Split in steps – e.g. top down

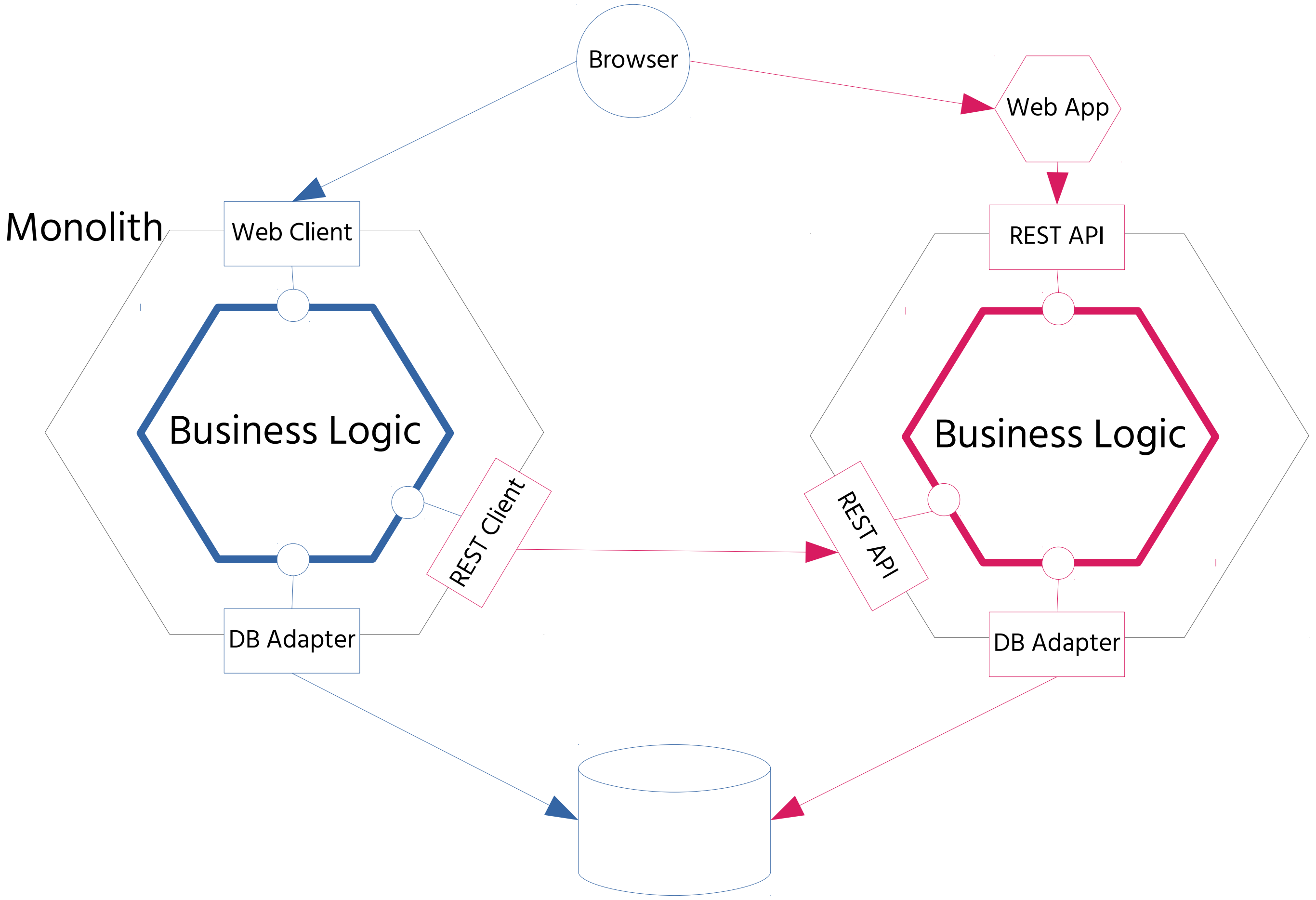
Split in steps – e.g. top down



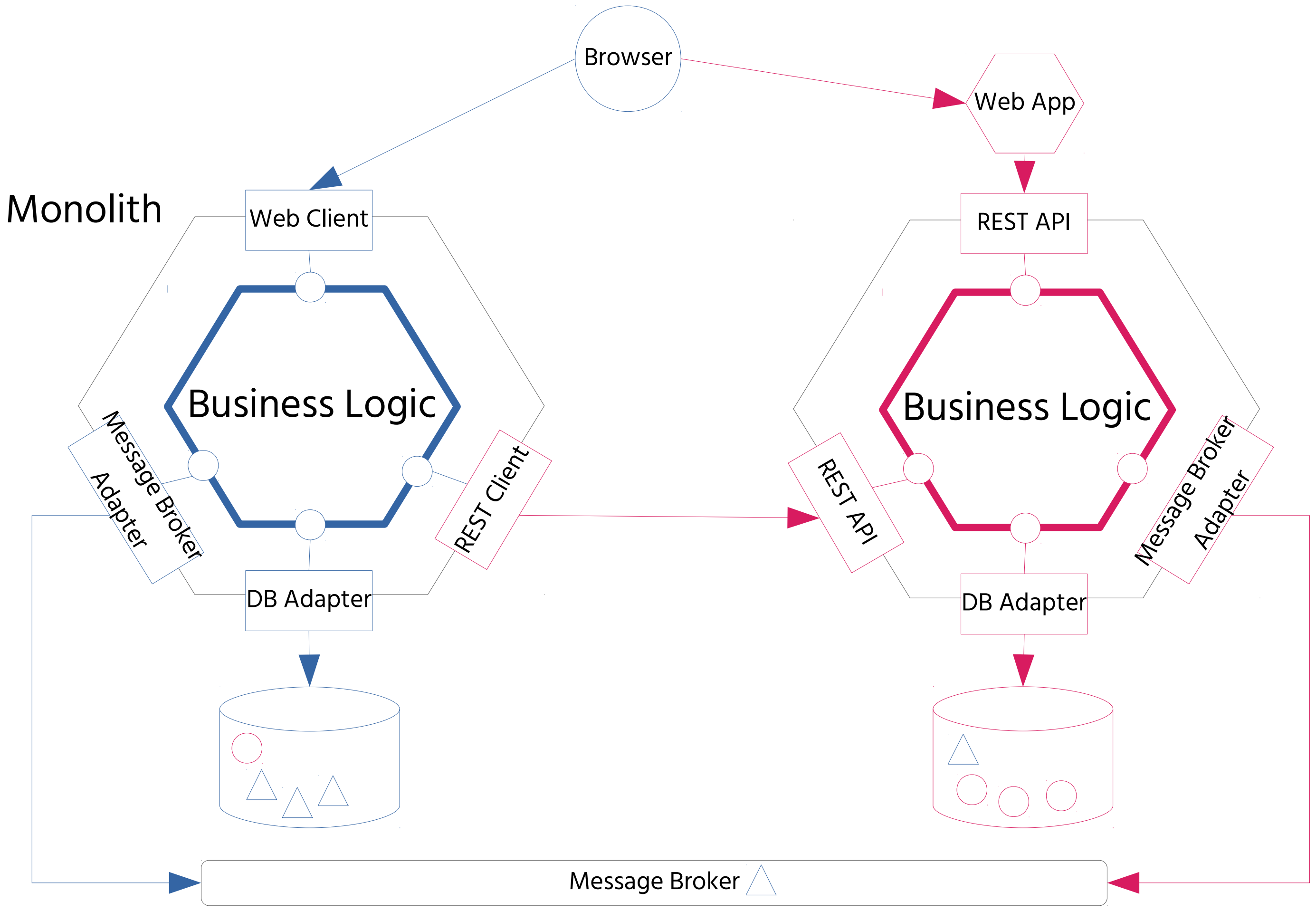
Split in steps – Step 1) Extracting Web App



Split in steps – Step 2) Extracting Business Logic



Split in steps – Step 3) Extracting Data Storage



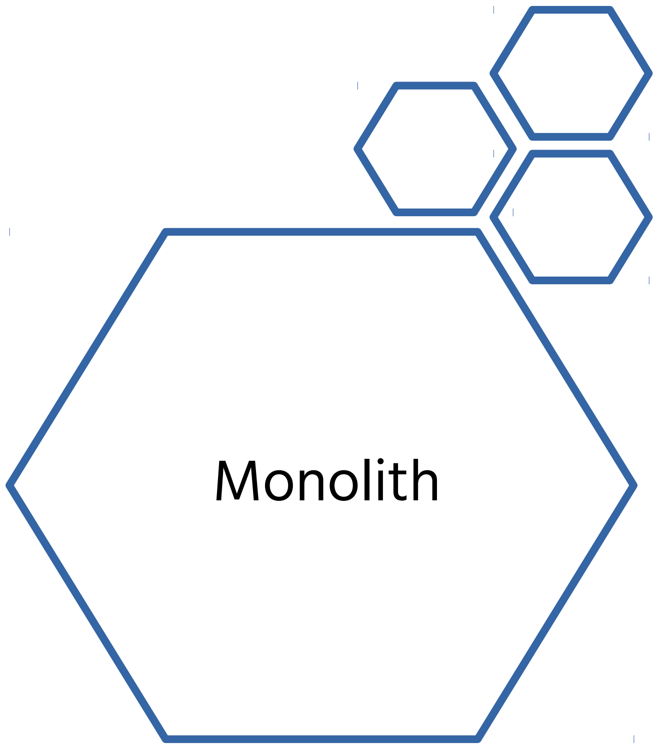
Which one first?

Easy to extract

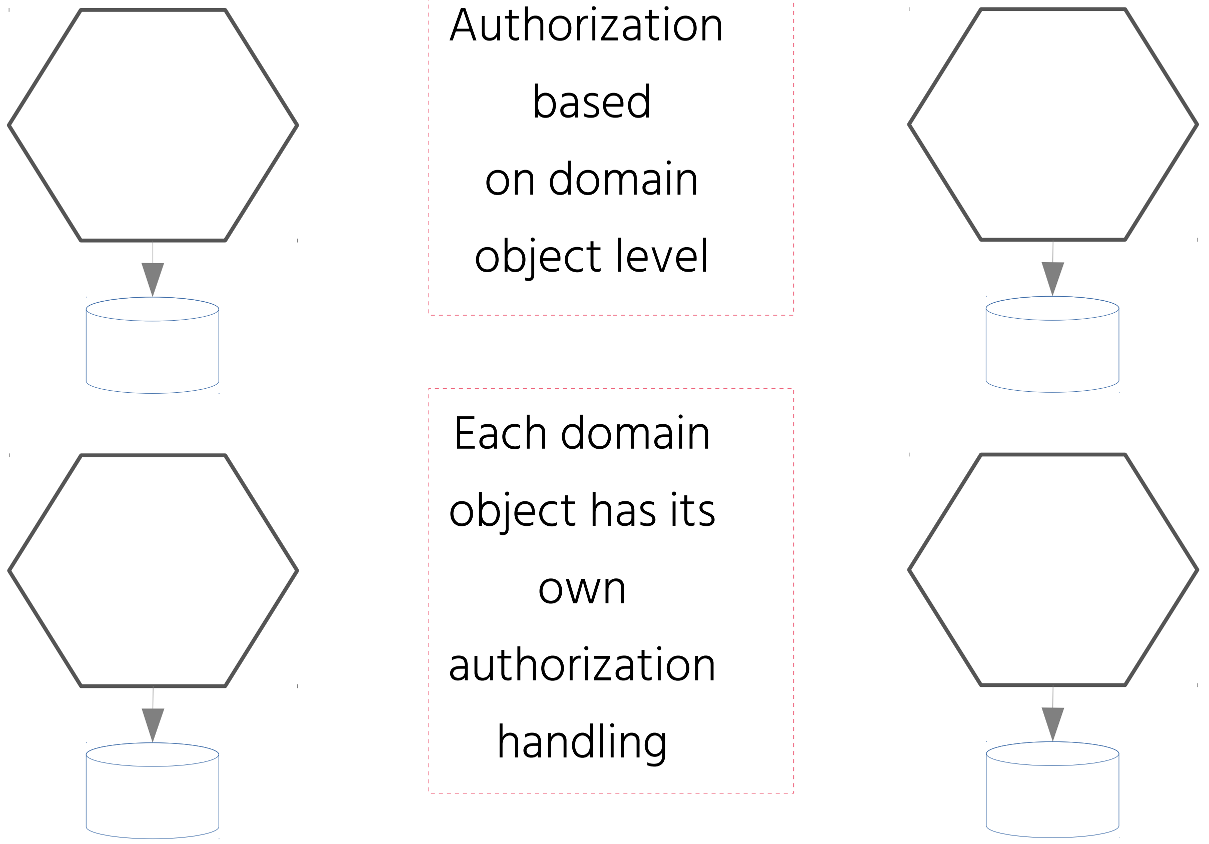
Changing frequently

Different resource requirements

Stop feeding the monolith



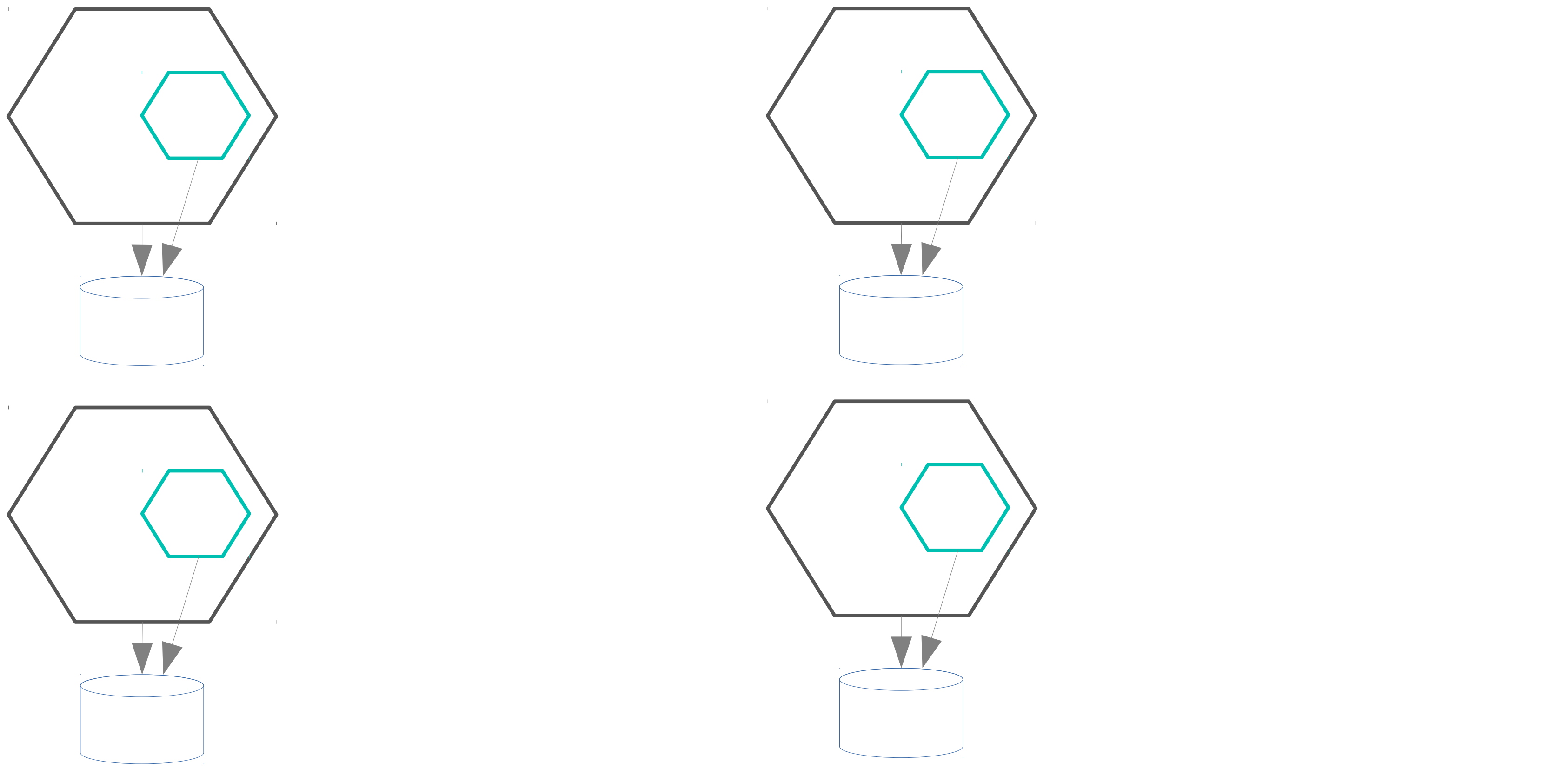
How to handle Authz? Our authz context ...



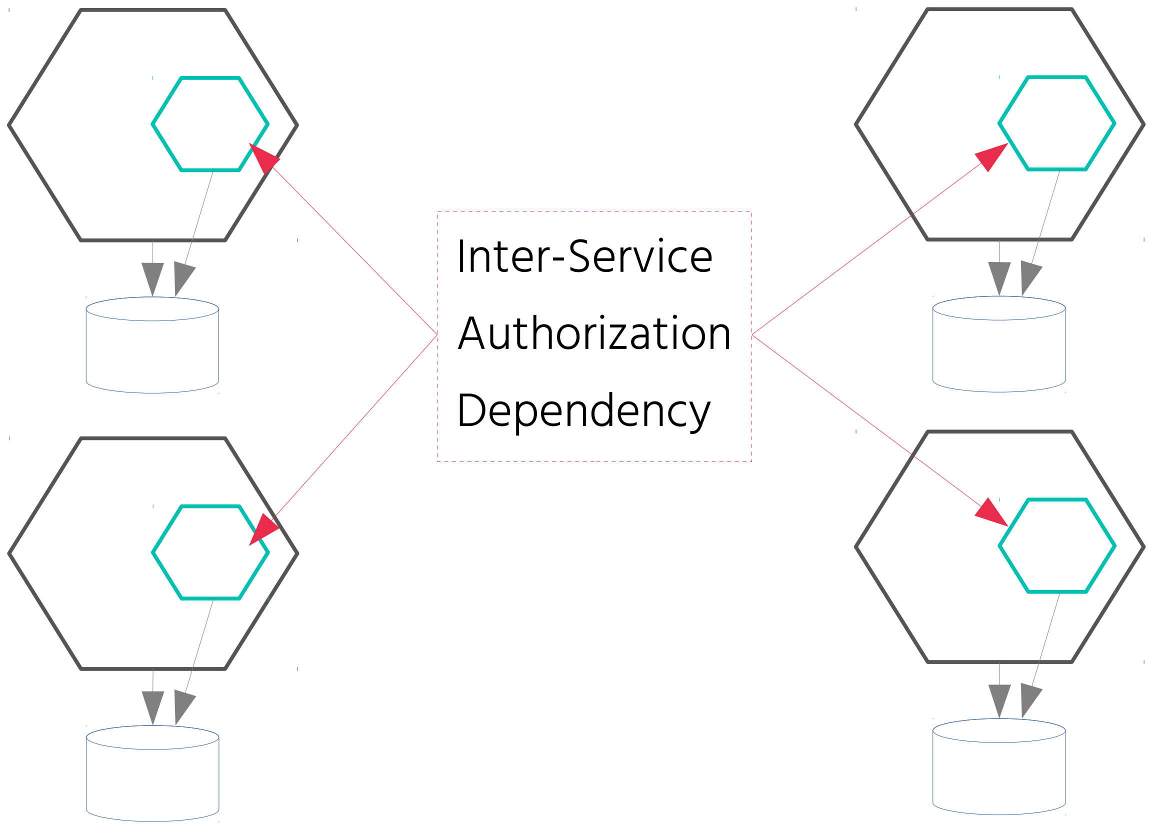
Authorization based on domain object level

Each domain object has its own authorization handling

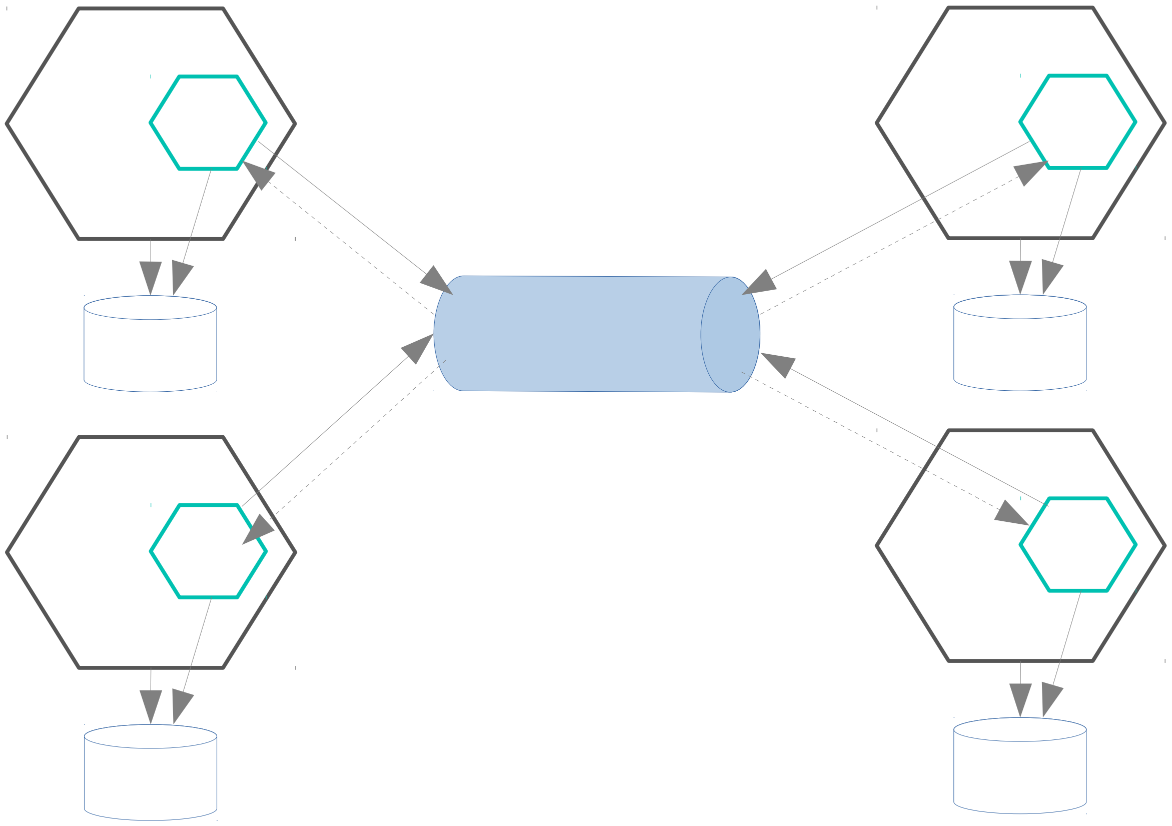
How to handle Authz? We started with…



How to handle Authz? But we missed a point ...



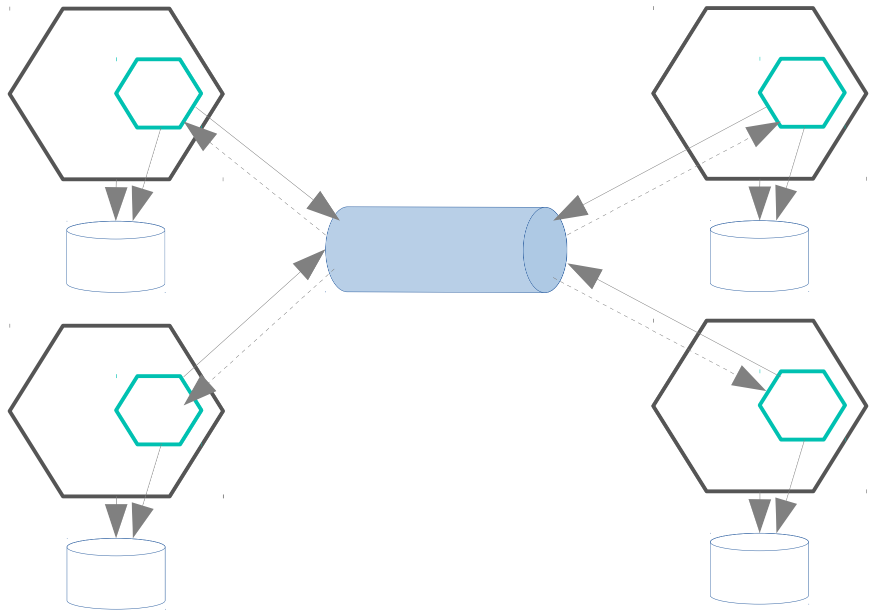
How to handle Authz? Leading to ...



How to handle Authz? Not decentralized!

+ Fast in-process calls for read access
+ No single point of failure
- Every MS has to implement authz logic
- For a change every MS has to be updated
- Duplication of all global authz data
- Verbose communication
- Tight coupling between services

=> Authorization is a cross-cutting concern

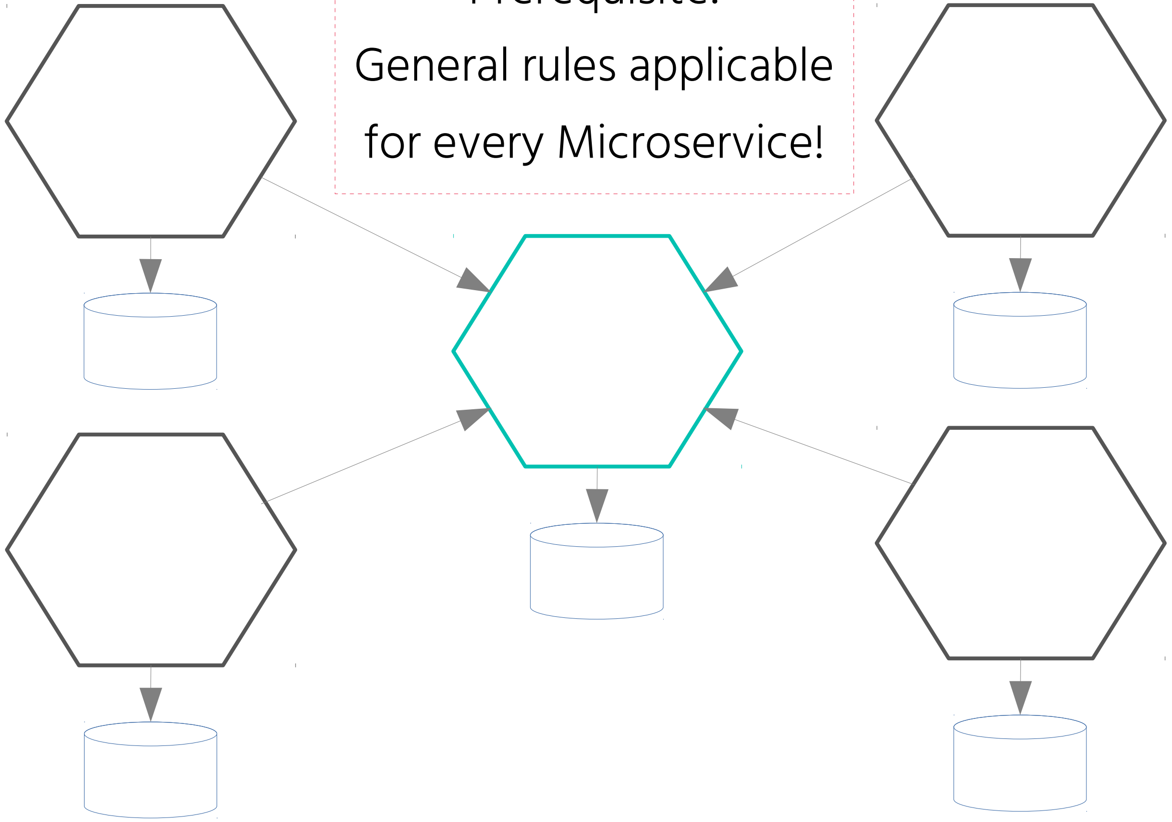


Decentralized

How to handle Authz? Centralized!

Prerequisite:

General rules applicable for every Microservice!



How to handle Authz?

+ One authz logic implementation

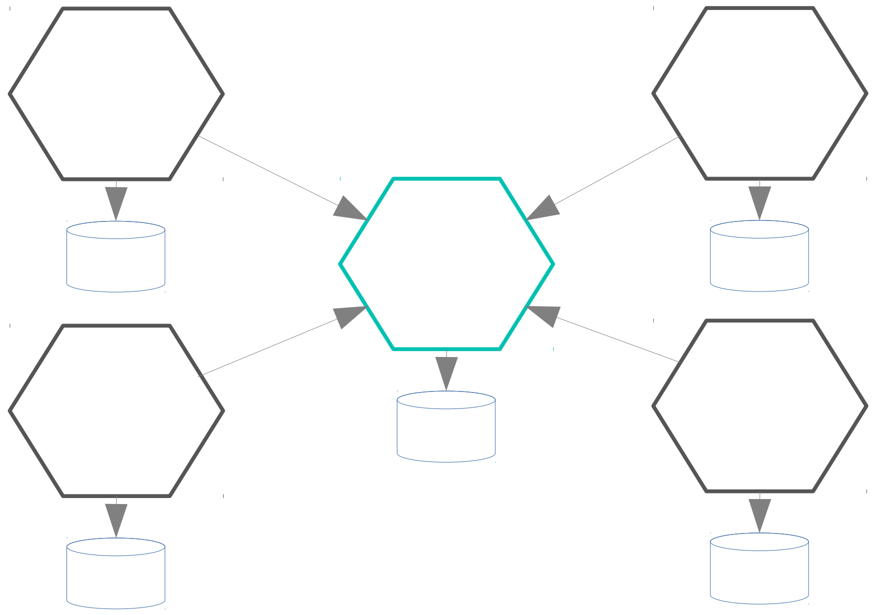
+ Change at one place

+ Explicit data sovereignty

+ No duplicated data

- Communication over network

- Single Point of Failure



Centralized

Whenever you encounter

communication and **implementation overhead**

leading to **high coupling** between the services

the seam might be wrong.

Transformation process

Establish Microservices ecosystem

Microservices ecosystem

- CI/CD Pipeline
- Monitoring
- Log tracing
- Testing (incl. API)
- Central Configuration
- Design for Failure
- API-Gateway
- Service Discovery
- Load Balancing
- Dev Sandbox

Microservices ecosystem Tool examples f. Java

- Jenkins & ANSIBLE
- Prometheus & Grafana
- Spring Cloud Sleuth & Zipkin
- Spring Cloud Config
- Pact (CDC-Testing)

- NETFLIX OSS Hystrix
- NETFLIX OSS Zuul
- NETFLIX OSS Eureka
- NETFLIX OSS Ribbon
- VAGRANT

Lessons learned

- Establishing Microservices ecosystem takes time and requires different skills & tools
- No explicit infrastructure team slows down the process
- Starting with decomposing big chunks frustrates
- Evaluate communication flow to identify wrong seams
- It takes far longer than originally anticipated

By **starting small** and decomposing in
manageable steps and taking care
of your **ecosystem from the beginning**
the transformation process can be handled
with even **limited resources**.

MADE IN ST. PAULI* W/ LOVE SWEAT & TEARS

*) Quarter of Hamburg, famous for its soccer club & entertainment district :)

... AND W/ MICROSERVISES !

THANK YOU!

Susanne Kaiser
CTO
@suksr

Just Software
@JustSocialApps