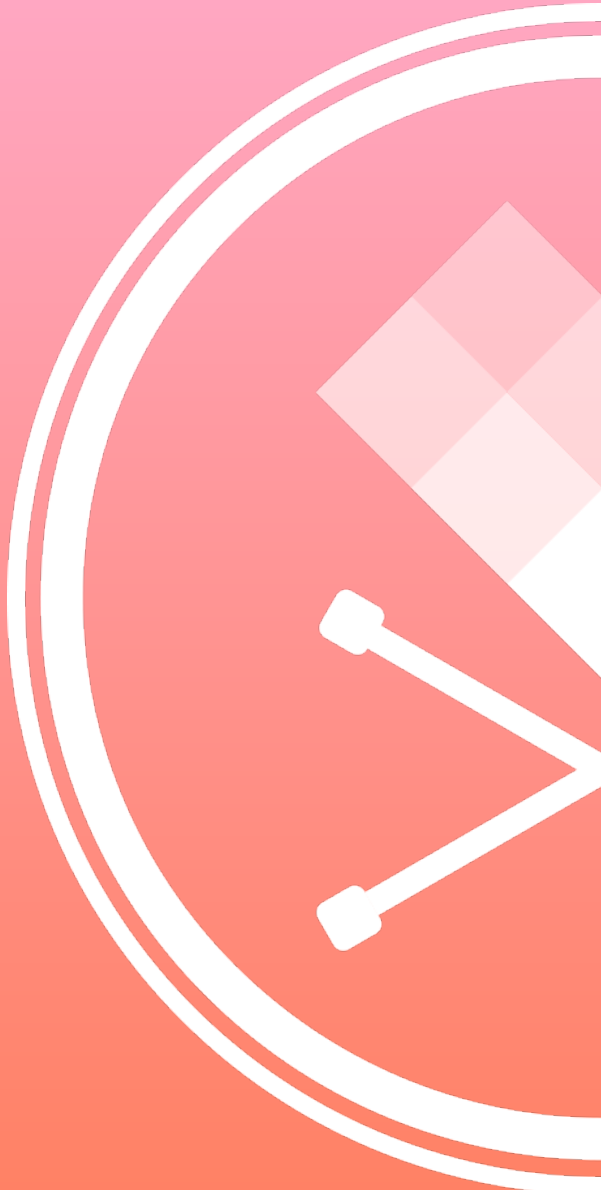
goto; conference


**Click 'Rate Session' to rate session and ask questions.**

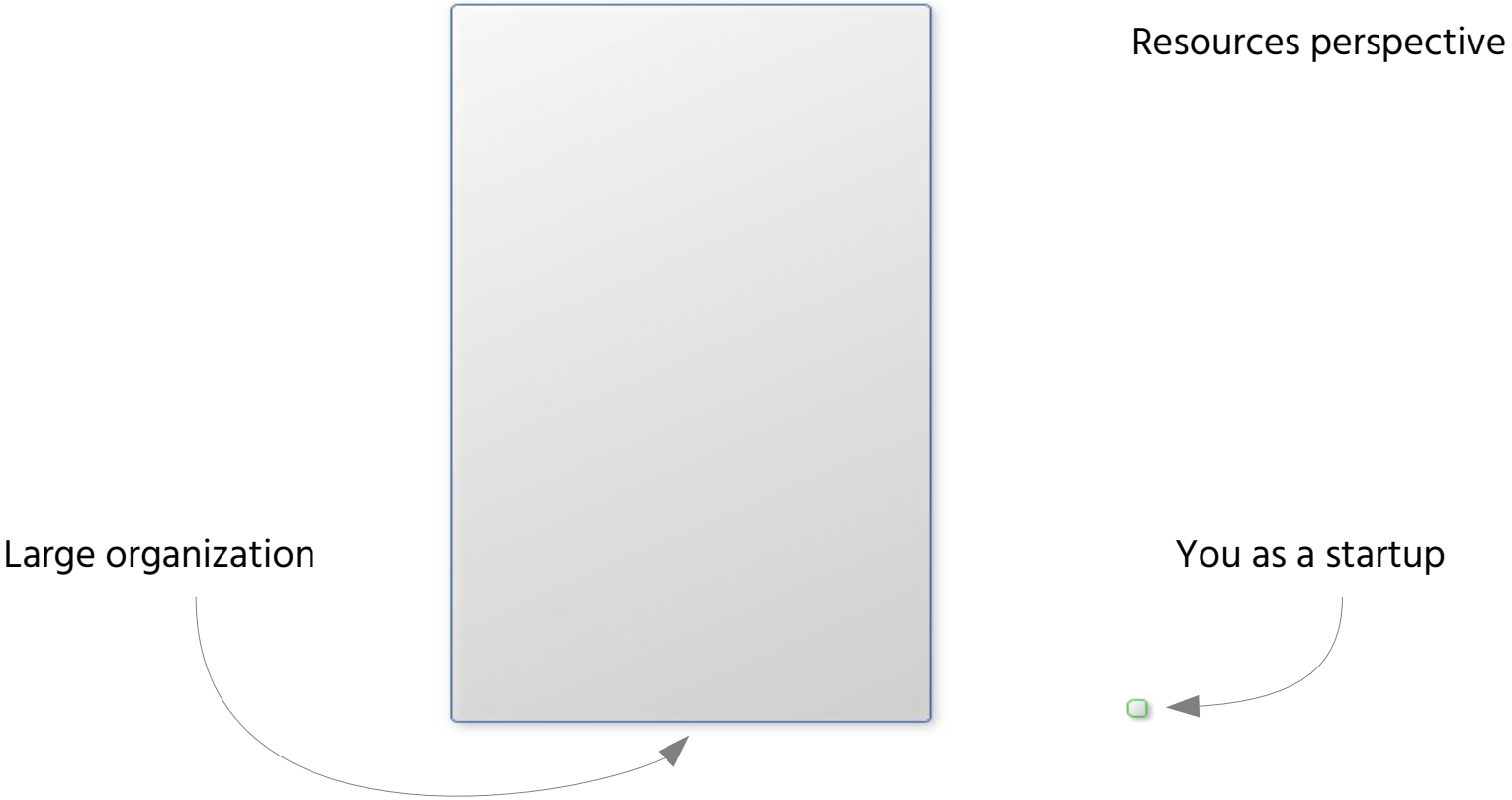
# The Microservices journey from a startup perspective

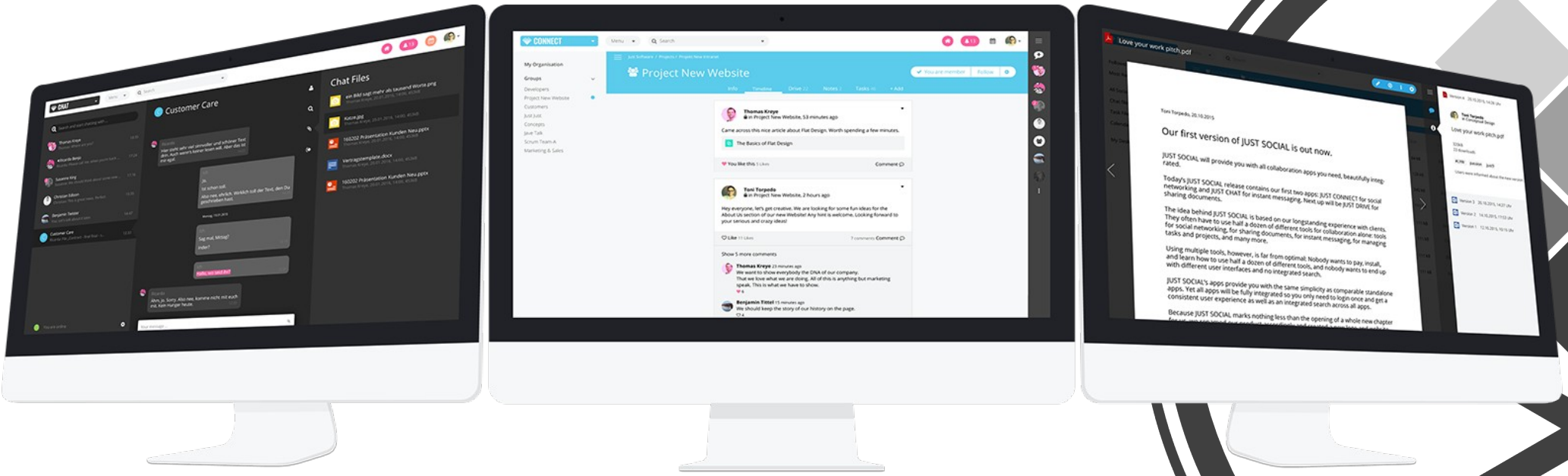
Susanne Kaiser
CTO
@suksr

Just Software
@JustSocialApps



# Each journey is different

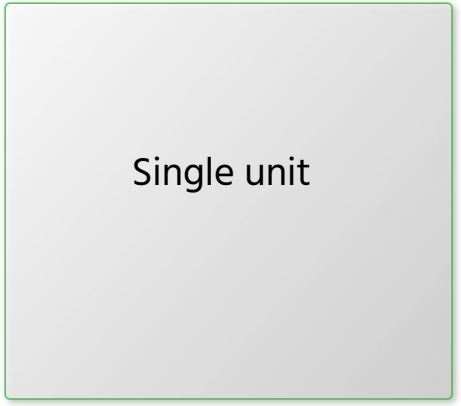



CHAT
Customer Care
Chat Files
CONNECT
Project New Website
Thomas Kreye
Toni Torpedo
Benjamin Timel
Love your work pitch.pdf
Our first version of JUST SOCIAL is out now.

# JUST SOCIAL

# The beginning … A monolith in every aspect

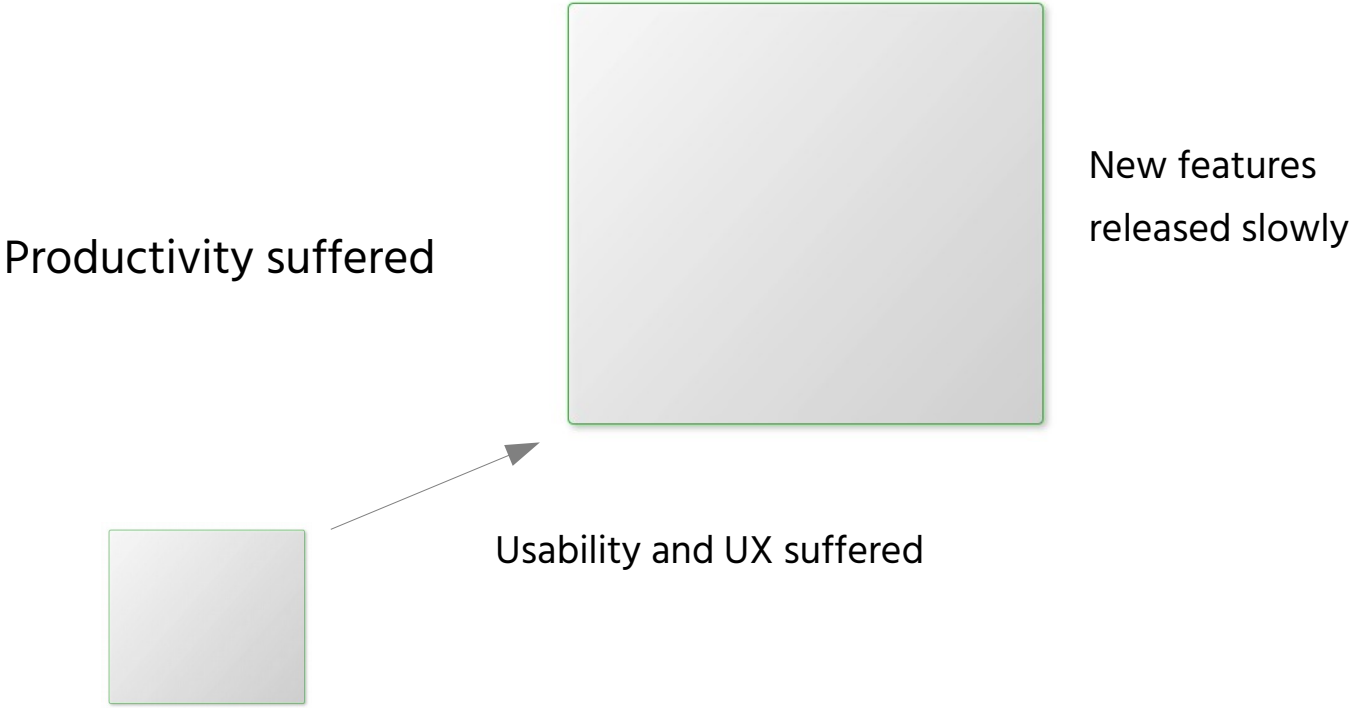
One team




One technology stack

One collaboration product

# After an evolving while …



# Separate Collaboration Apps

JUST PAGE
Social Network

JUST CONNECT
Real-time collaboration

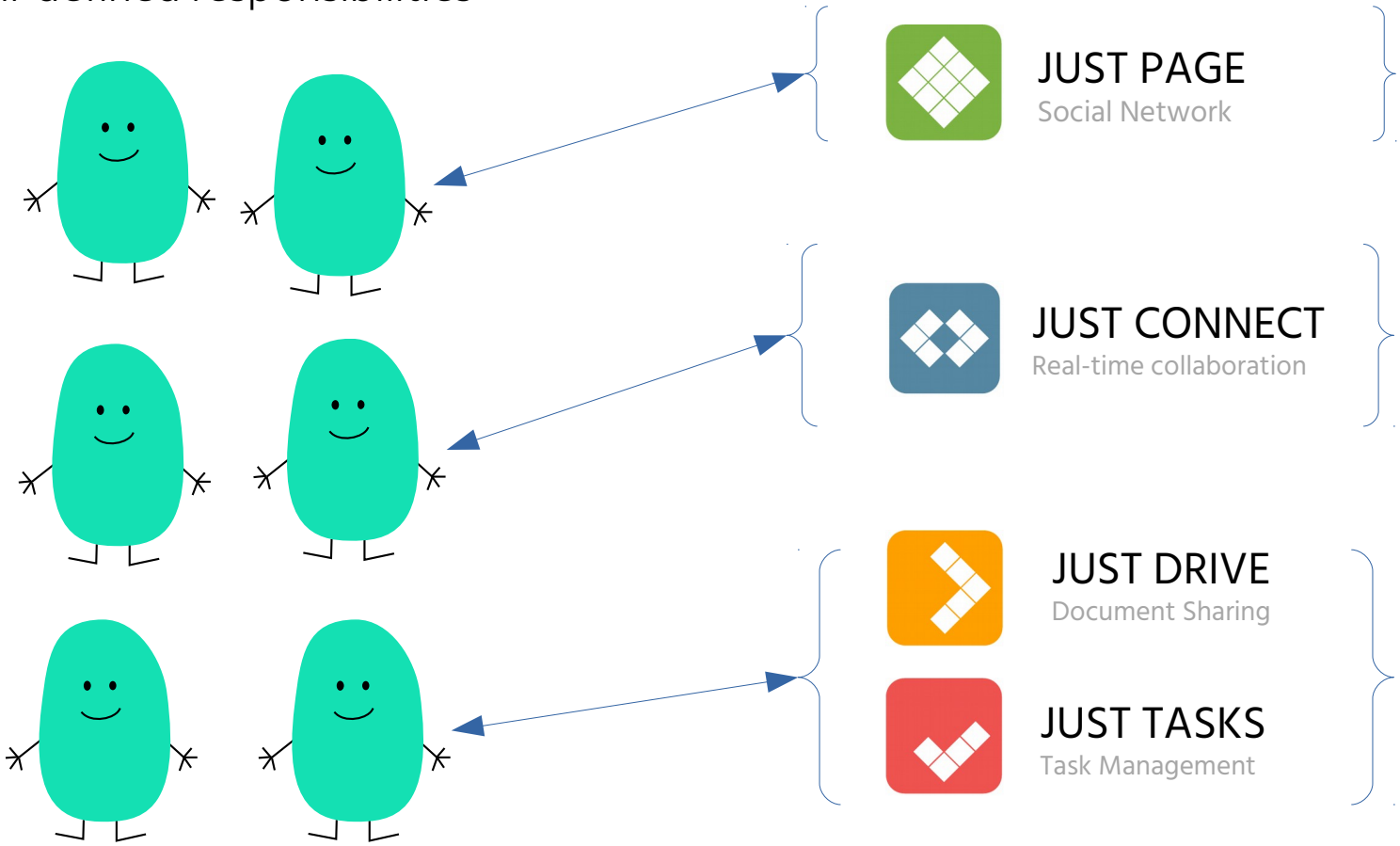
JUST TASKS
Task Management

JUST DRIVE
Document Sharing

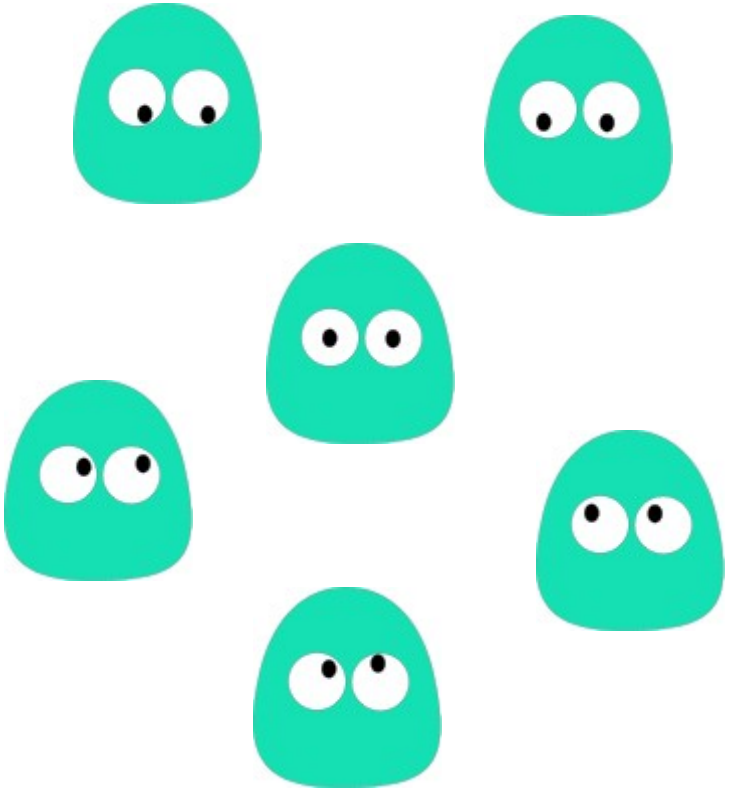
JUST SOCIAL

# Small, autonomous teams

with well-defined responsibilities

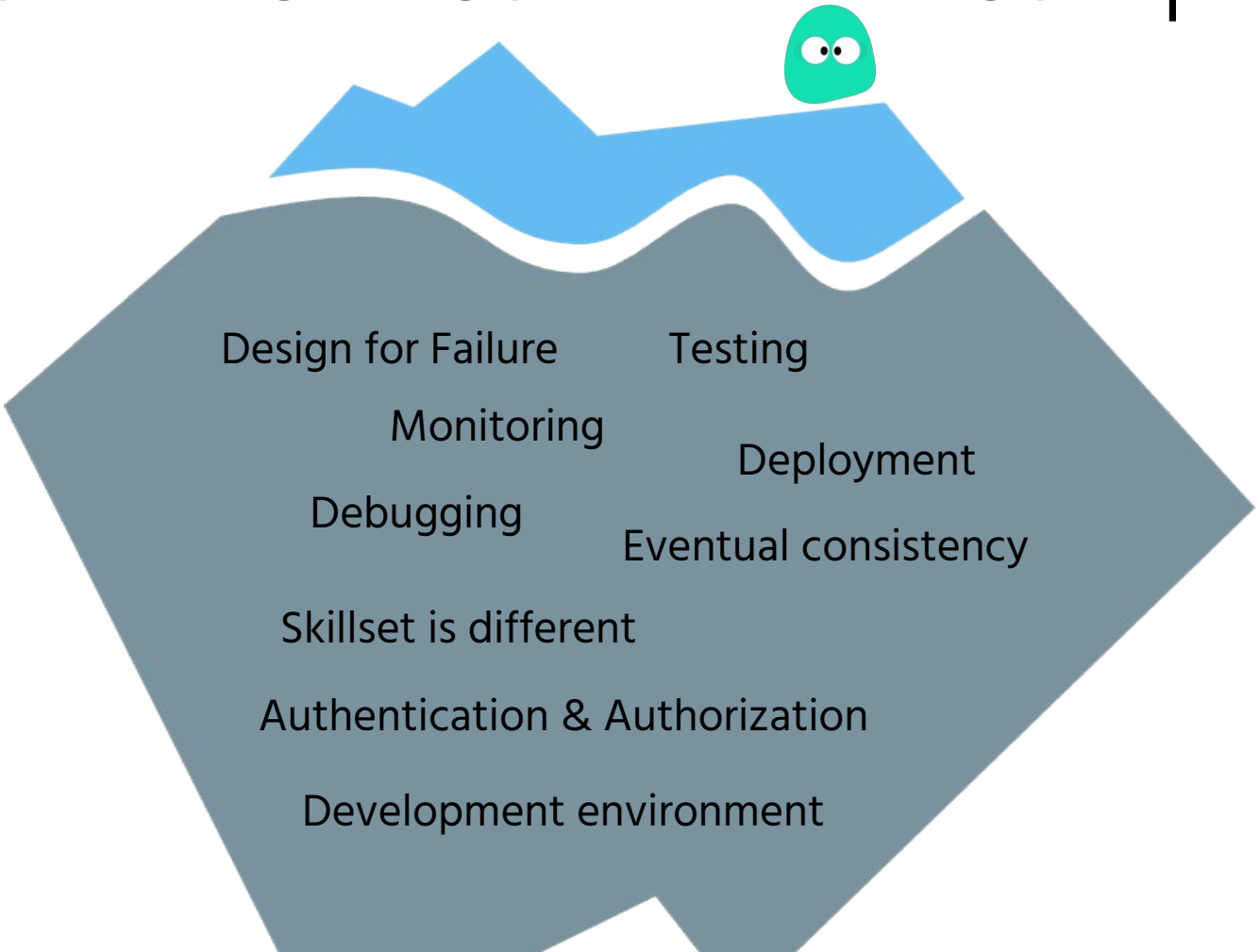


# In the long run …



# Looks easy at first glance …

# Microservices come with complexity



# Challenges of transformation

Transformation takes longer than anticipated

You still have to take care of your existing system

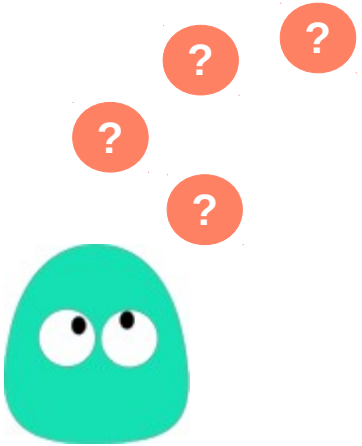
Core functionality is hard to untangle

All involved parties need to agree and to be on board

# 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

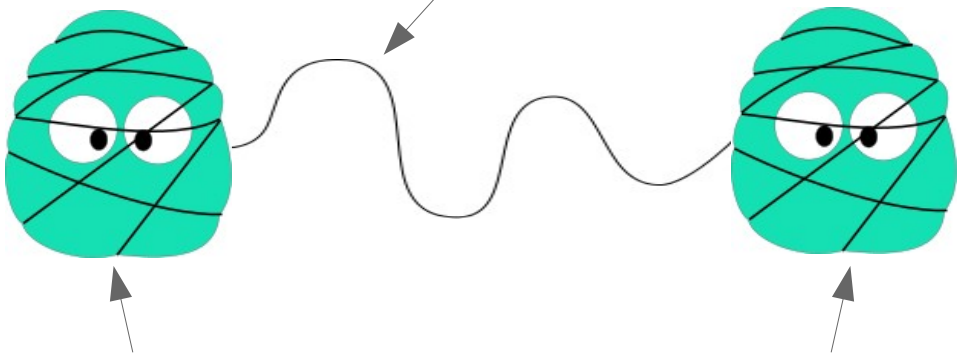
- Identify candidates
- Decompose candidates
- Establish Microservices ecosystem

# Transformation process

- Identify candidates

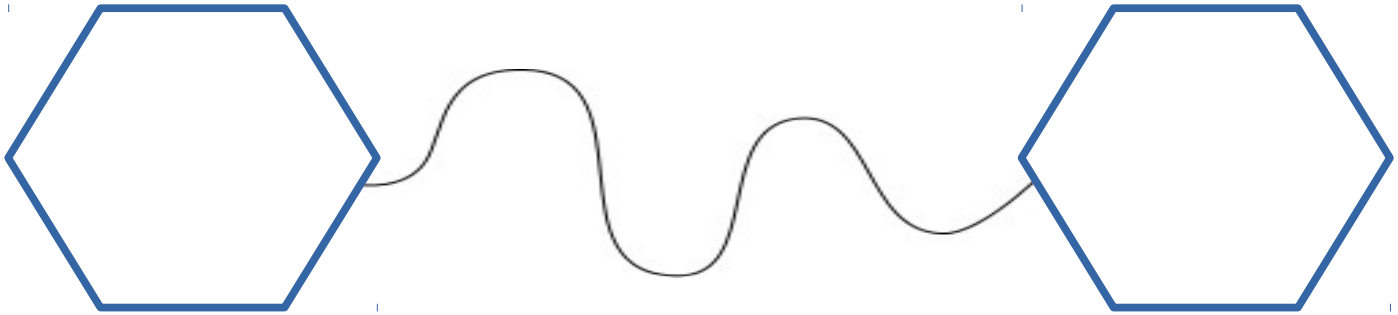
# Key concepts of modelling Microservices

Loose coupling between services



High cohesion within a service

# Identify Bounded Contexts



Well defined business function

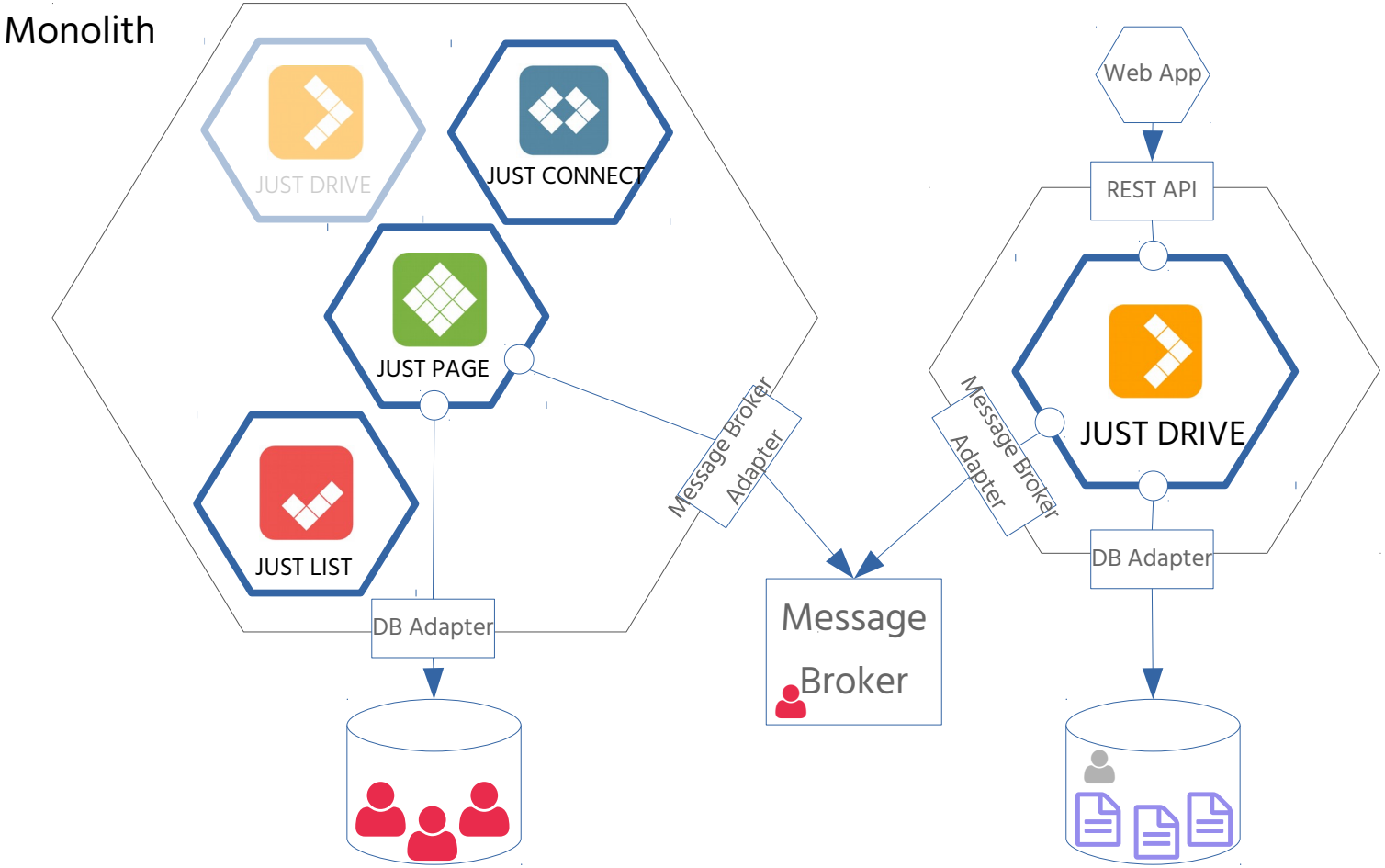
# Bounded Contexts = Collaboration Apps

Monolith

# Transformation process

Decompose candidates

# First approach as a co-existing service

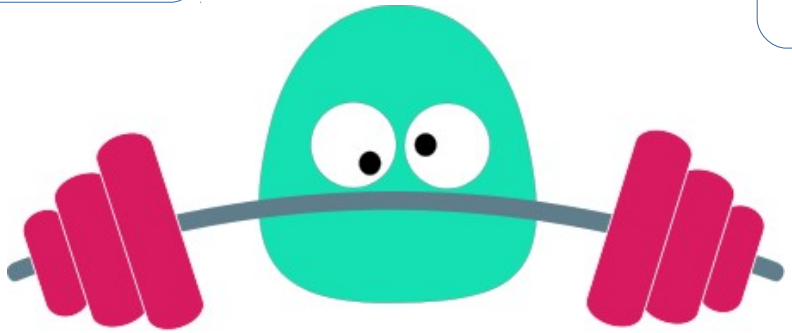


# Heavy undertake if you do all at once

New UI

Maintain & run current system

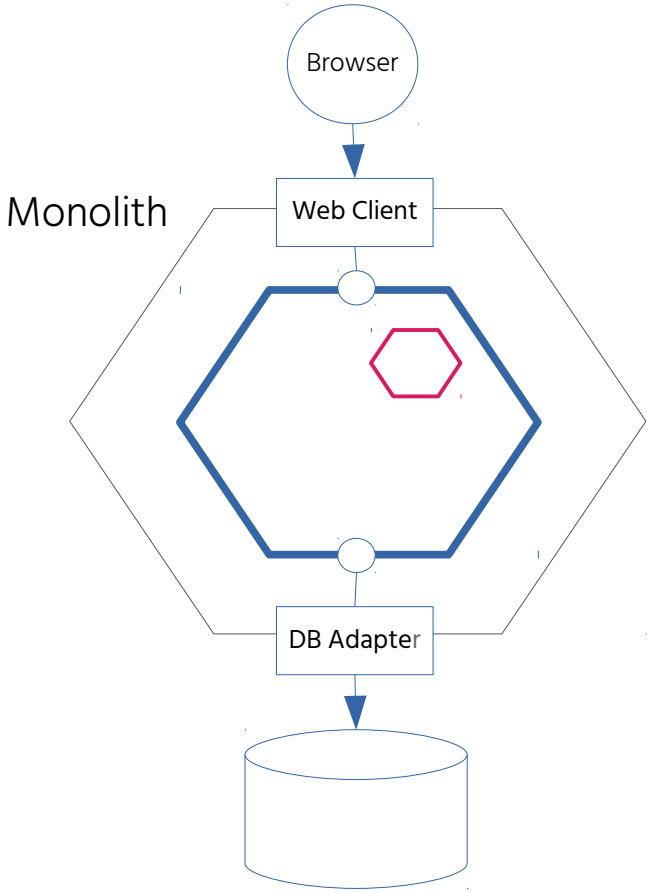
New data structure



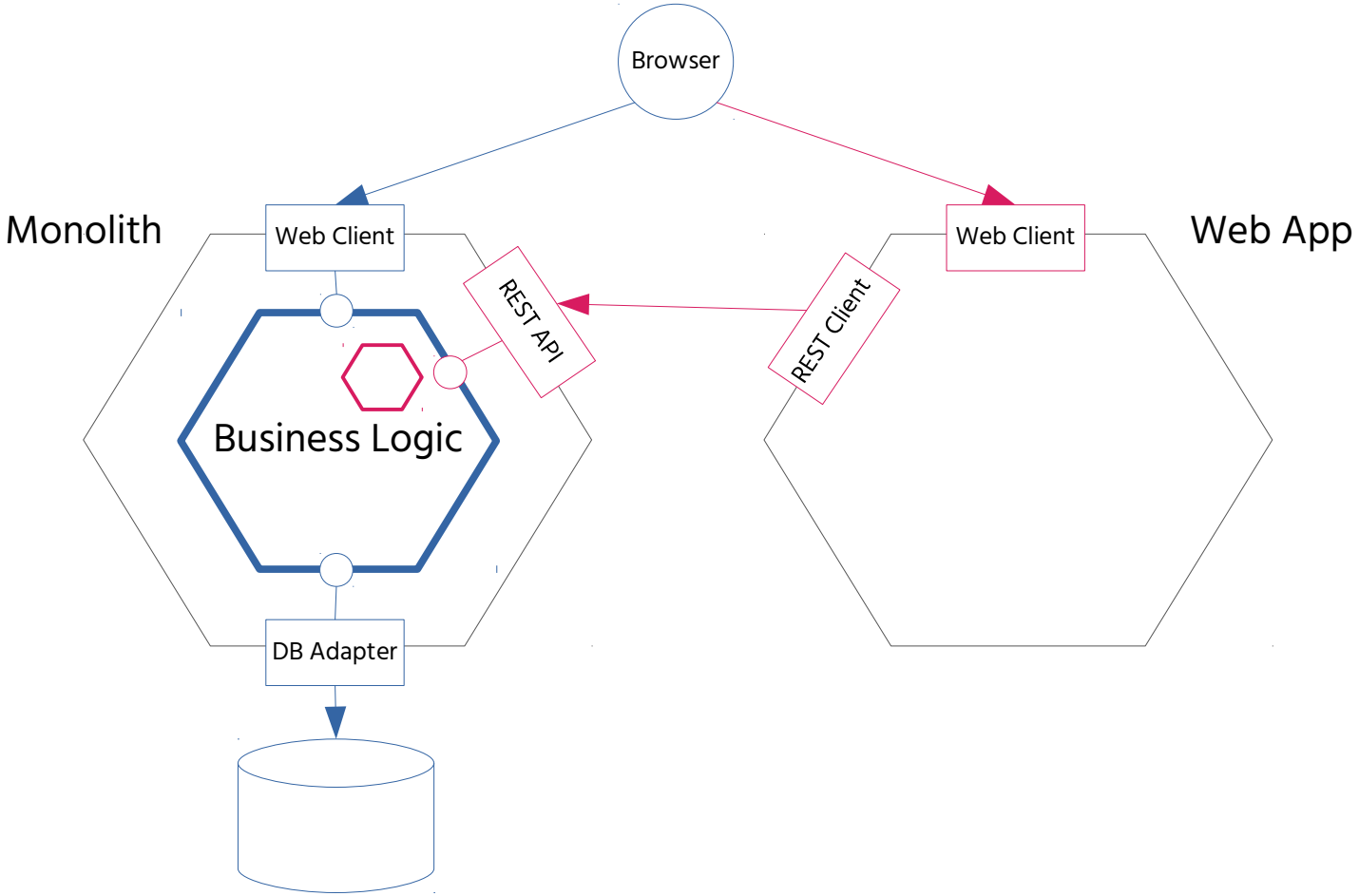
More features

# Split in steps – e.g. top down

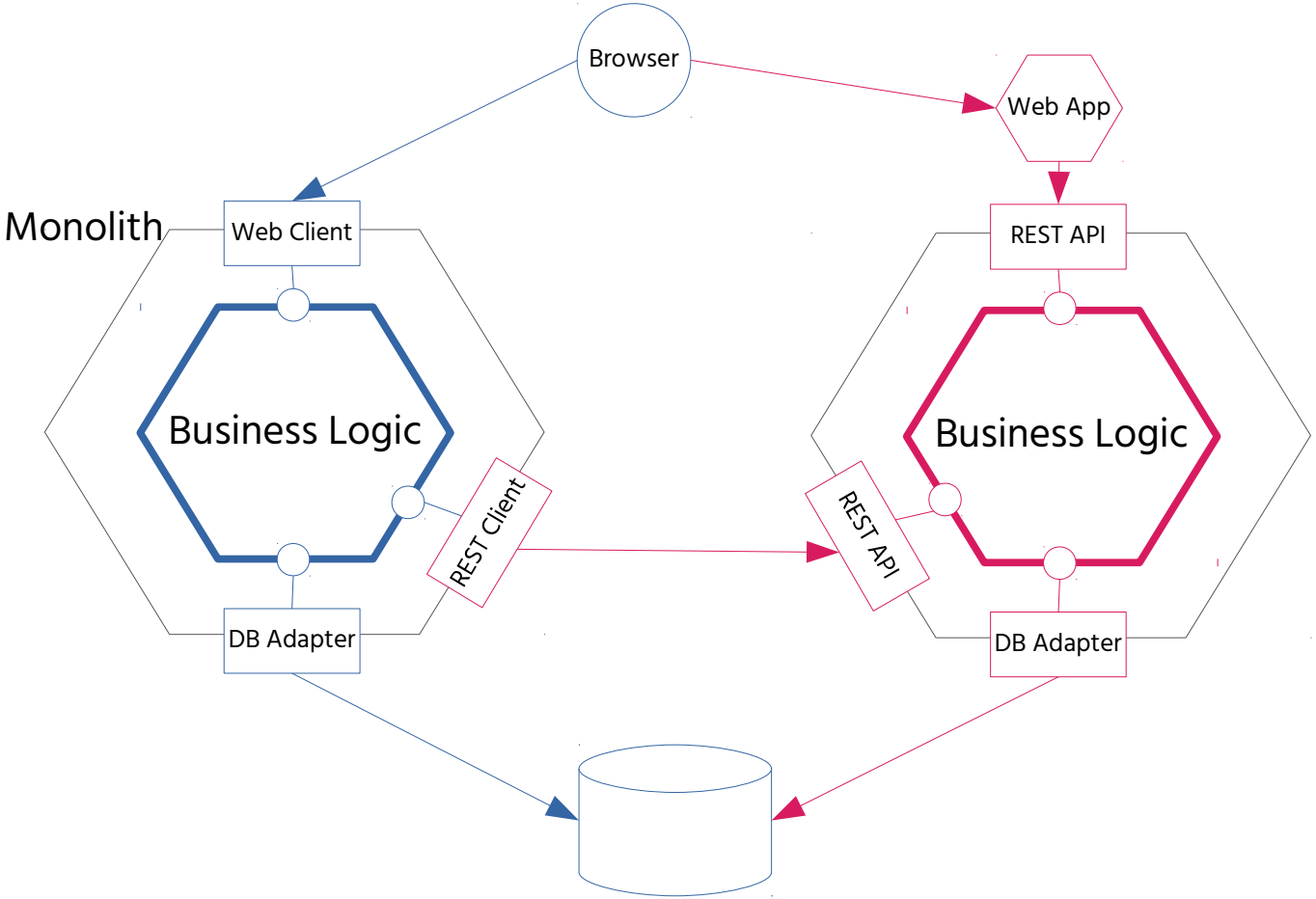
# Split in steps – e.g. top down



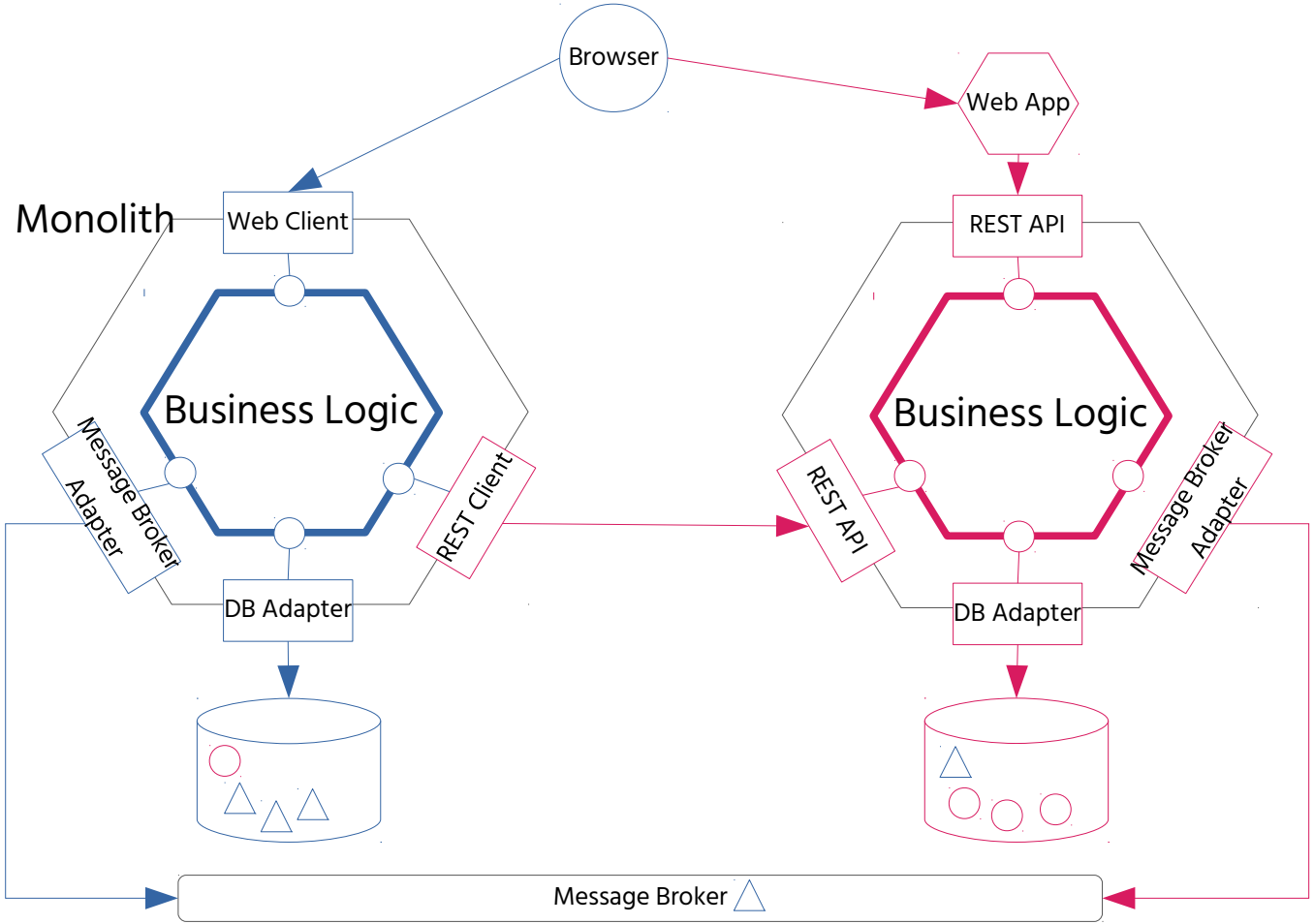
# Split in steps – Step 1) Extracting Web App



# Split in steps – Step 2) Extracting Business Logic



# Split in steps – Step 3) Extracting Data Storage



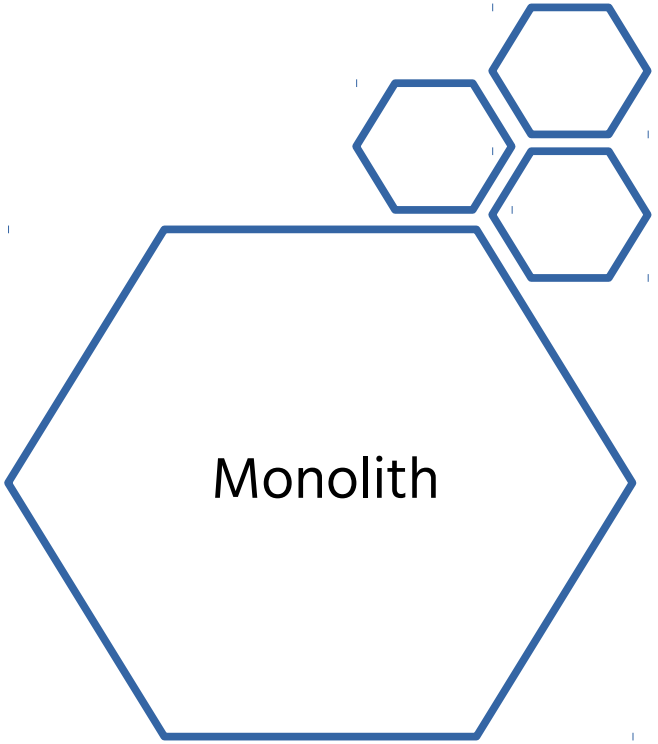
# Which one first?

Easy to extract

Changing frequently

Different resource requirements

# Stop feeding the monolith

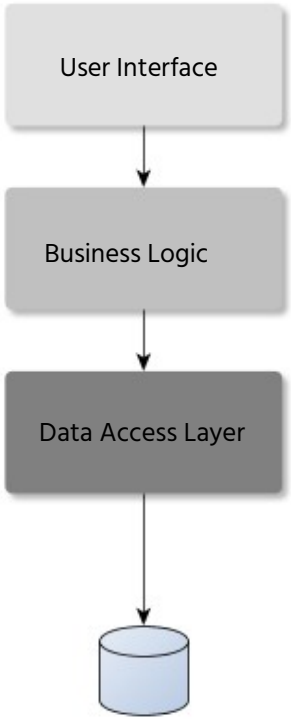


# Transformation process

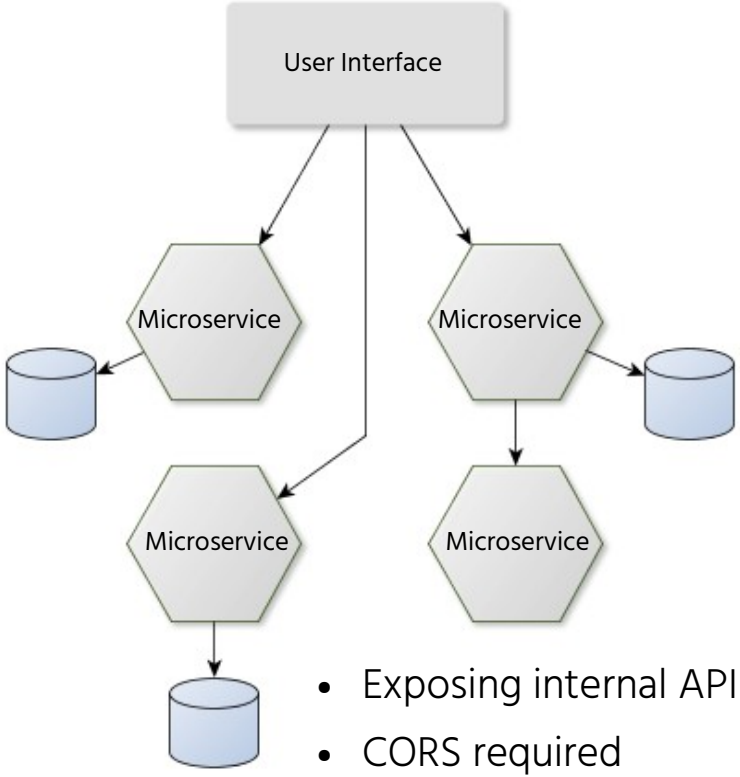
- Establish Microservices ecosystem

# Direct access over public internet problematic

## Monolith

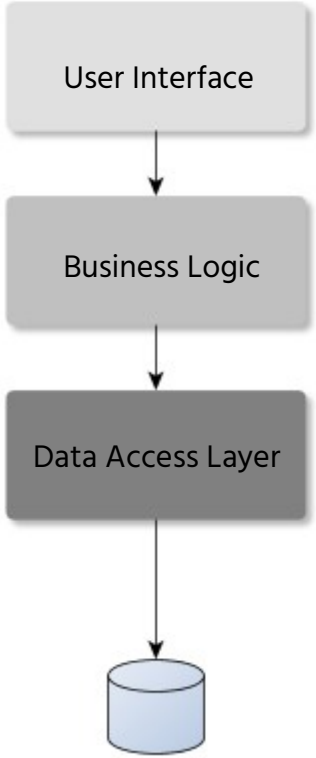


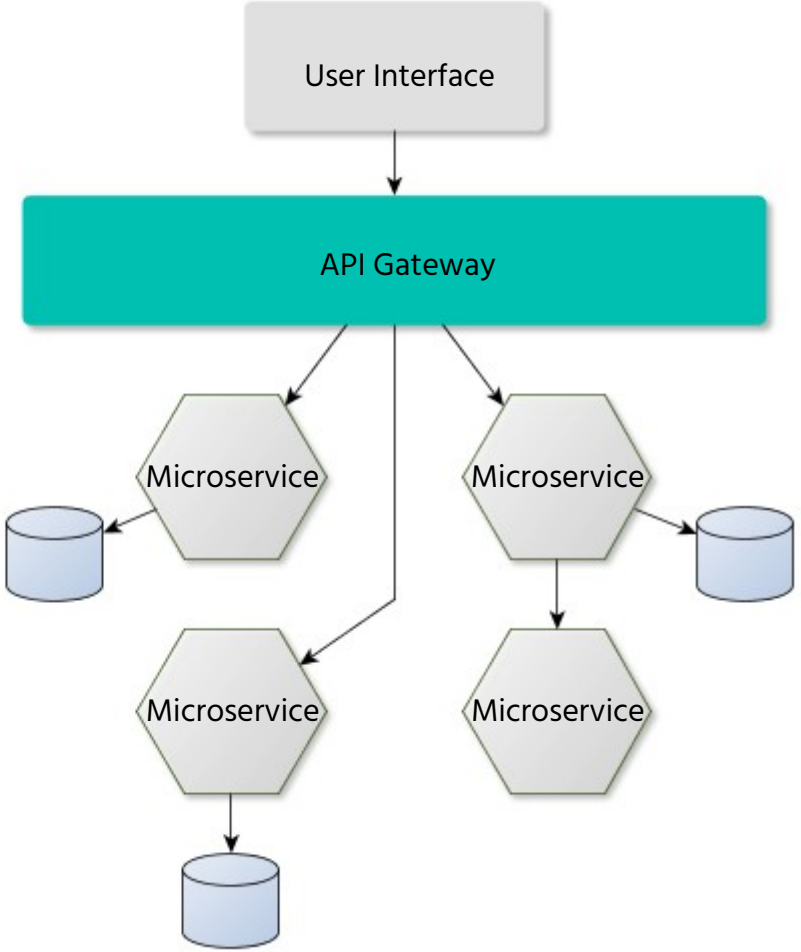

## Microservices




- Exposing internal API
- CORS required
- Multiple roundtrips
- Different clients have different needs

# API-Gateway provides simplified access for client

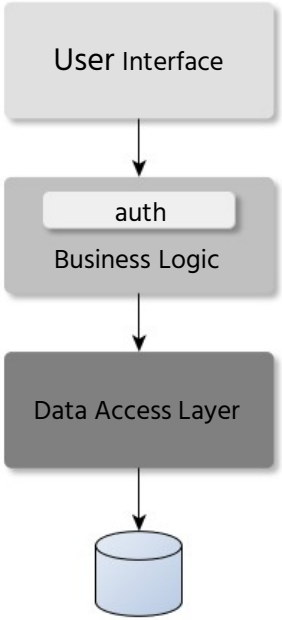


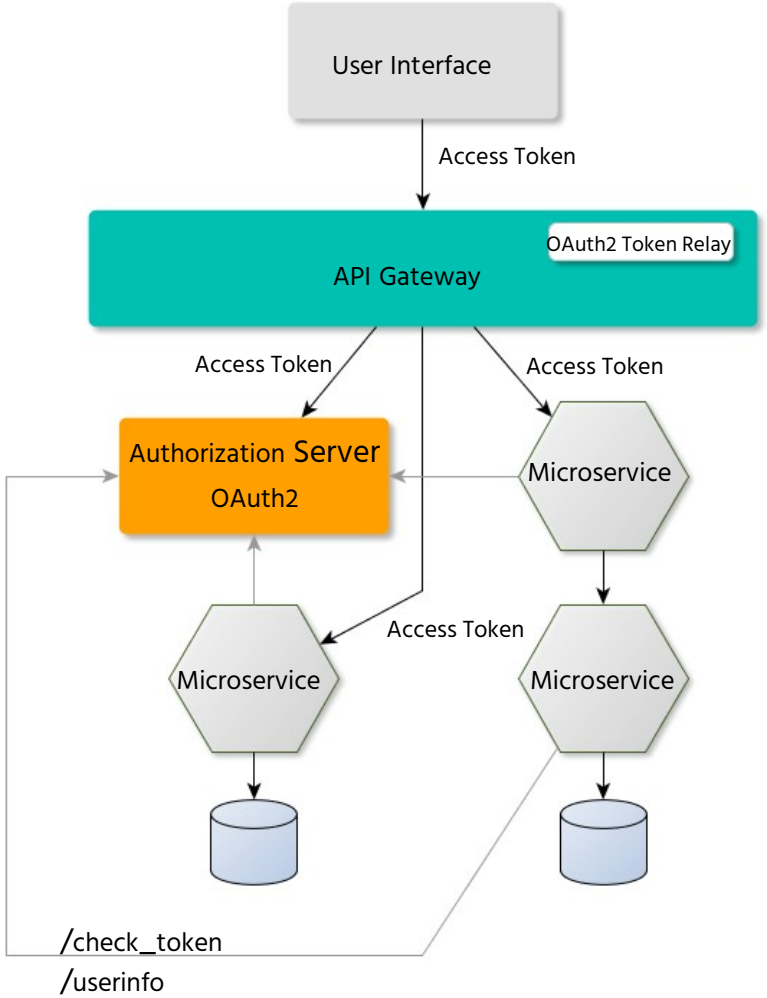



# Microservices ecosystem with …

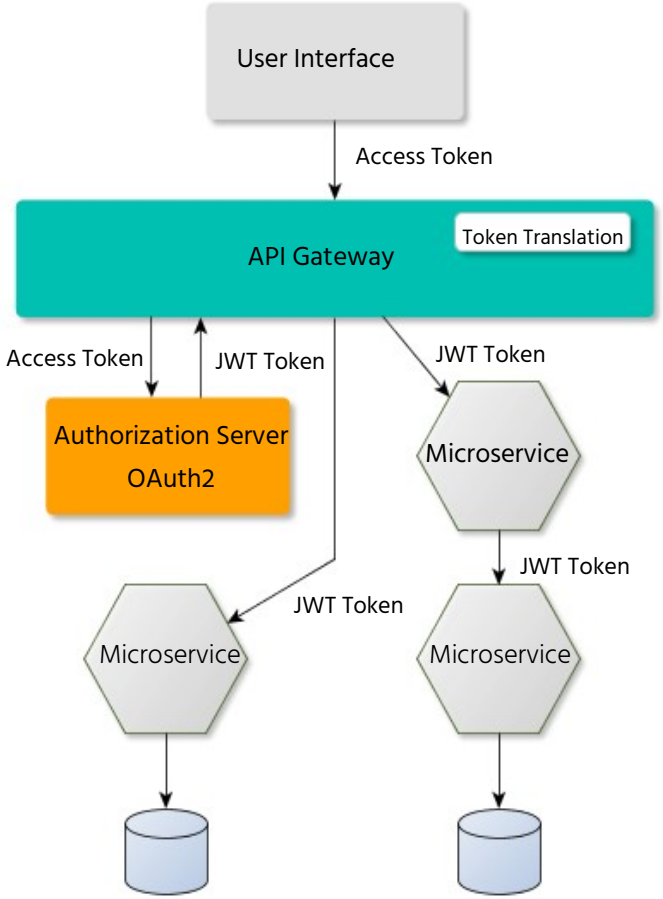

Spring Cloud & Netflix OSS

# Security: Auth-Server with API-Gateway as Token Relay

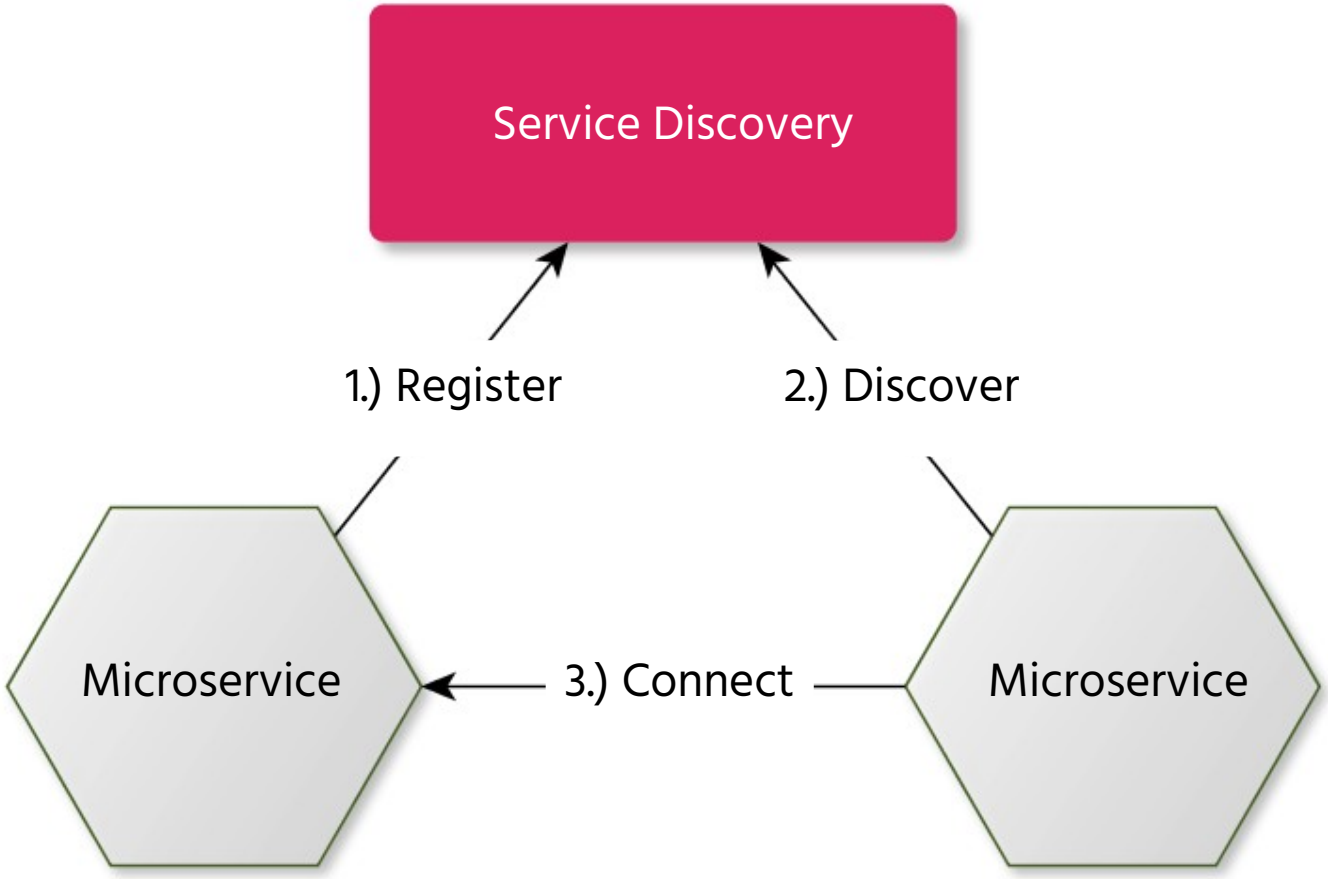


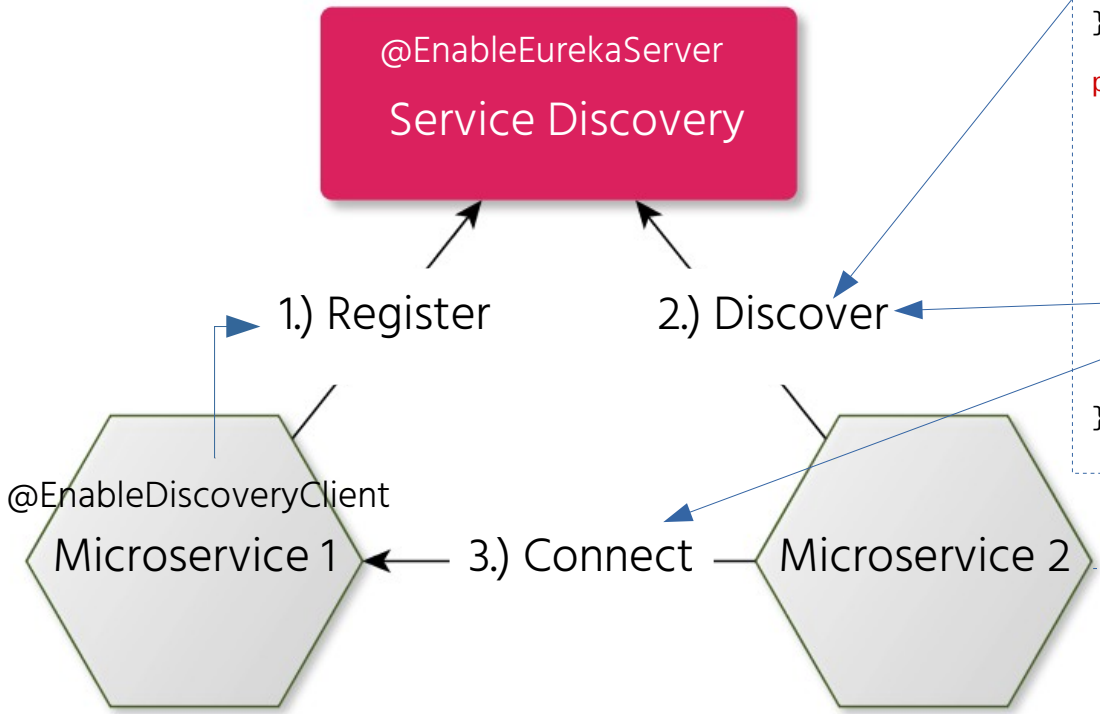
# Security: Auth-Server with API-Gateway as Token Translation



# Service-Discovery



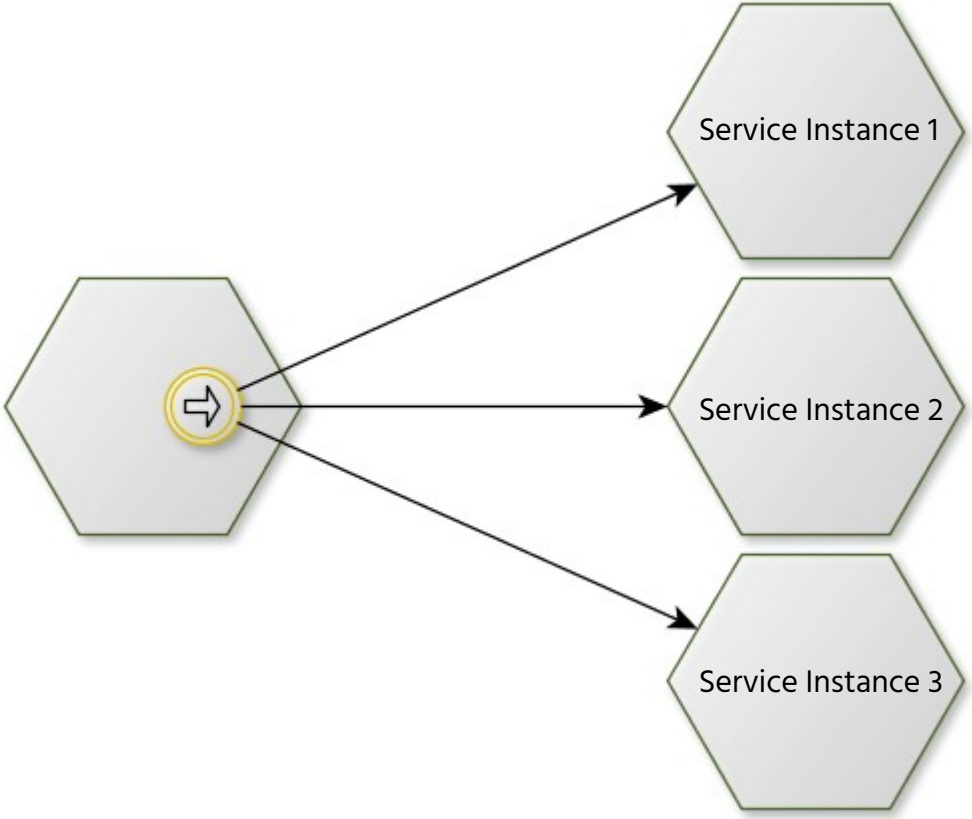
# Service-Discovery w/ Spring Cloud & Eureka



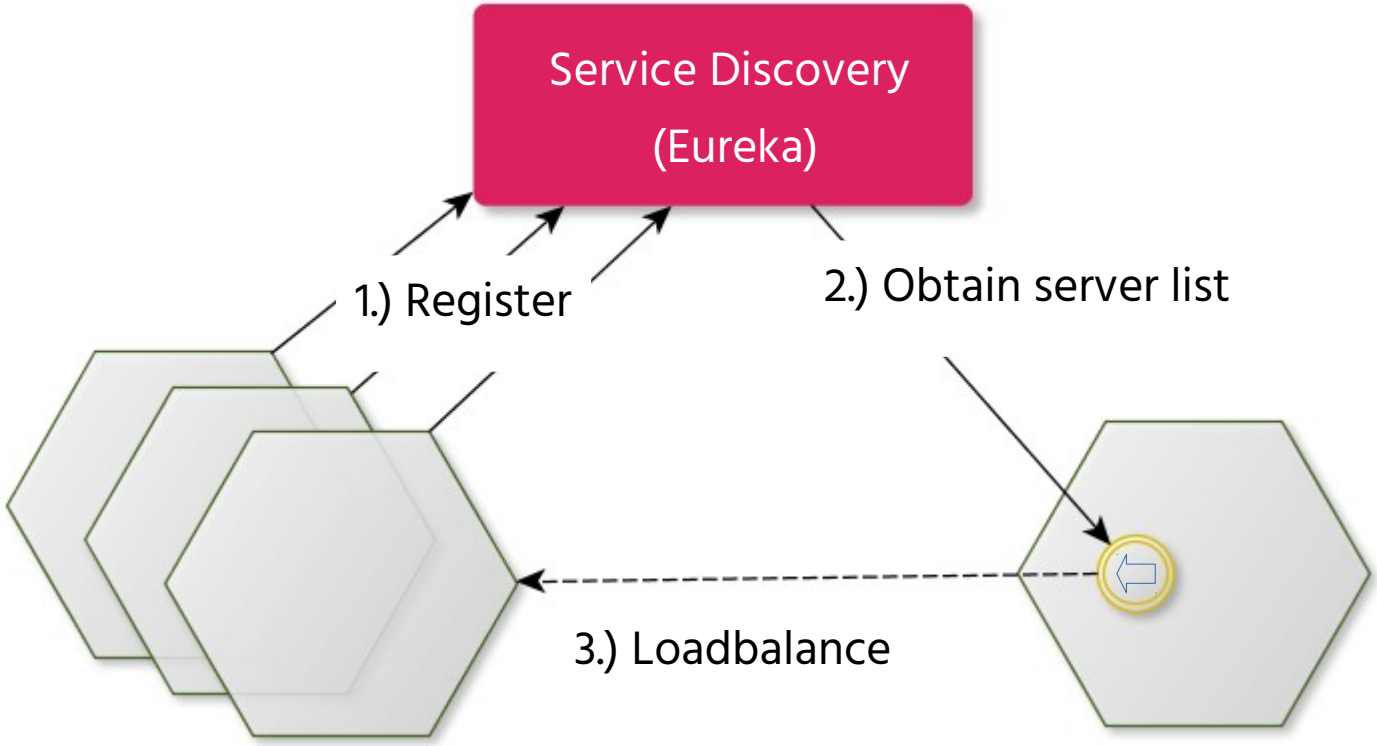

```java
@Configuration
public class MyConfiguration {
    @LoadBalanced
    @Bean
    RestTemplate restTemplate() {
        return new RestTemplate();
    }
}

public class MicroserviceTwo {
    @Autowired
    private RestTemplate restTemplate;
    public String doOtherStuff() {
        String results =
            restTemplate
                .getForObject(
                    "http://microservice-one/stuff",
                    String.class);
        return results;
    }
}
```

service id

# Client-side Loadbalancing



# Dynamic client-side Loadbalancing w/ Ribbon



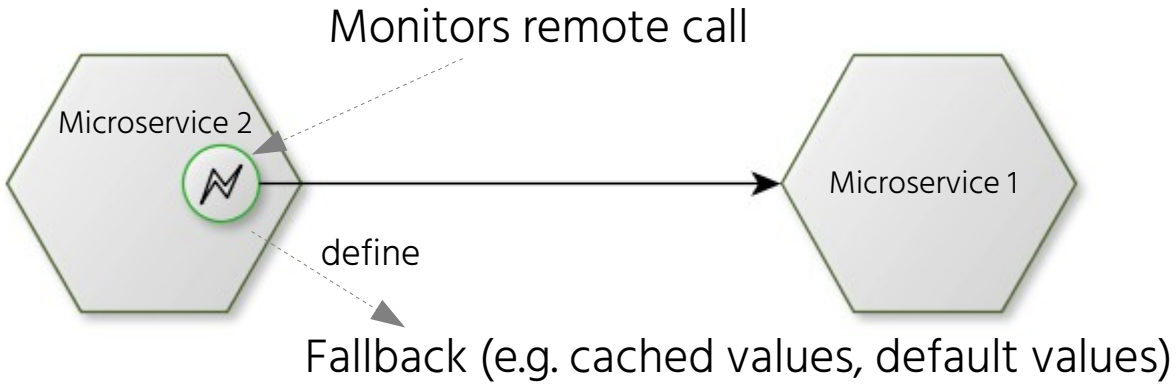


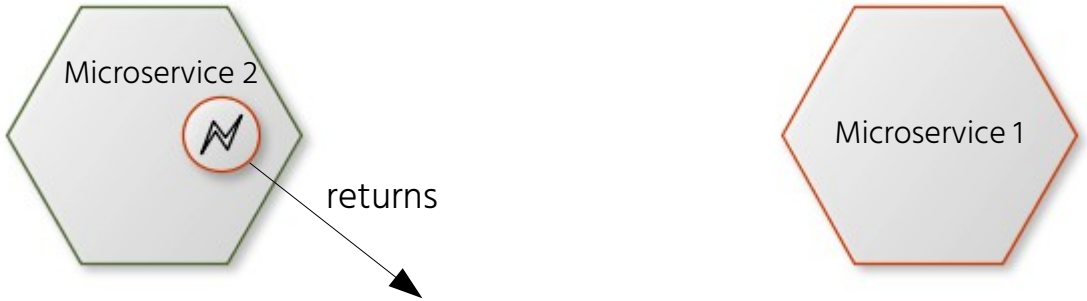
Load Balancer Ribbon

# Design for Failure

- Timeout-Handling
- Provide fallbacks
- Circuit Breakers

# Design for Failure w/ Hystrix

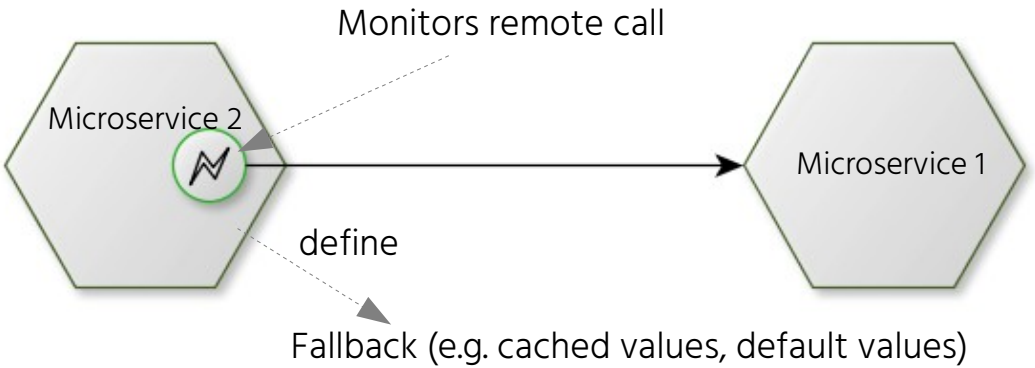




Circuit Breaker

# Design for Failure w/ Hystrix



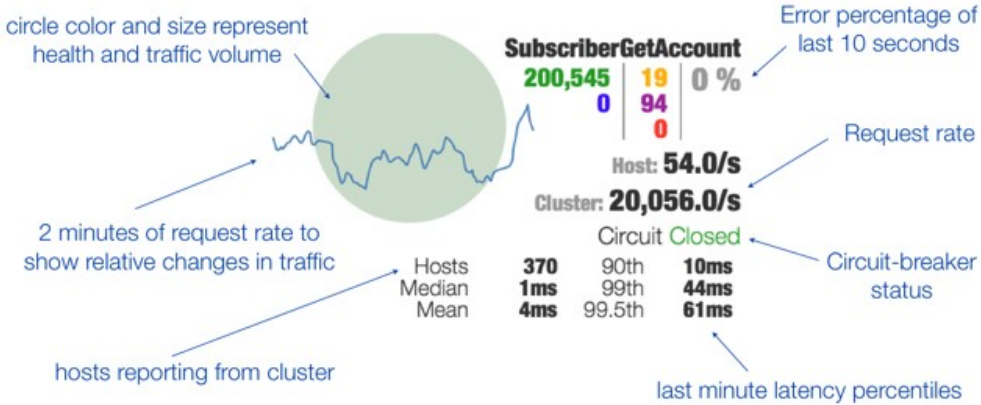

```
public class MicroserviceTwo {
    @Autowired
    private RestTemplate restTemplate;

    @HystrixCommand(fallbackMethod = "defaultValues")
    public String doOtherStuff() {
        String results =
            restTemplate
                .getForObject(
                    "http://microservice-one/stuff",
                    String.class);
        return results;
    }

    public String defaultValues() {
        ...
    }
}
```

```
@EnableCircuitBreaker
@Configuration
public class MyConfiguration {
    @LoadBalanced
    @Bean
    RestTemplate restTemplate() {
        return new RestTemplate();
    }
}
```

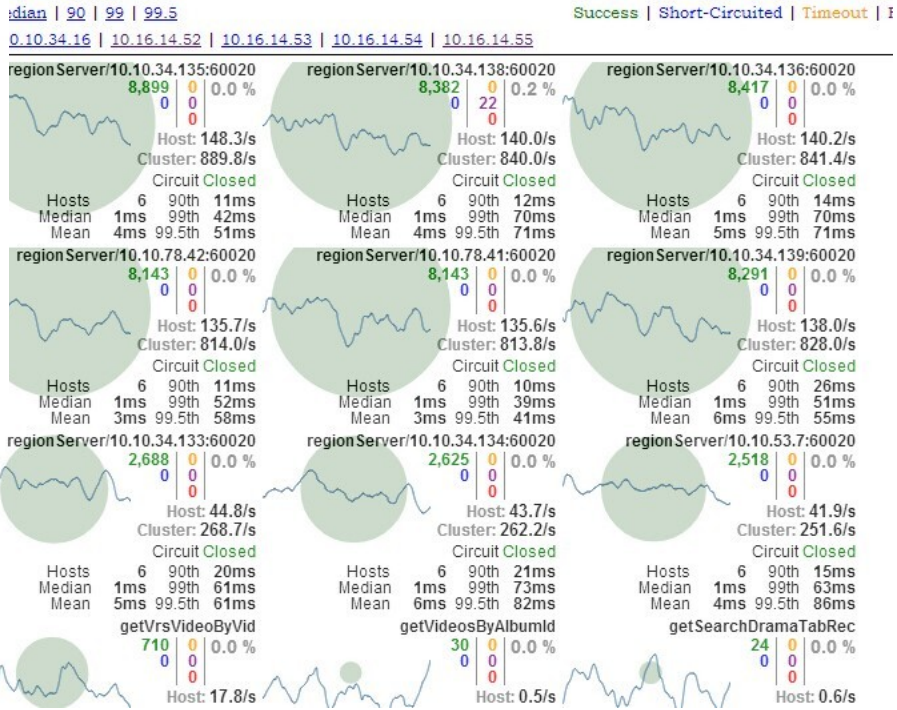
Timeout Setting (e.g. startup params)

```
-Dsun.net.client.defaultConnectTimeout=TimeoutInMiliSec
-Dsun.net.client.defaultReadTimeout=TimeoutInMiliSec
```

# Monitoring w/ Hystrix Dashboard & Turbine

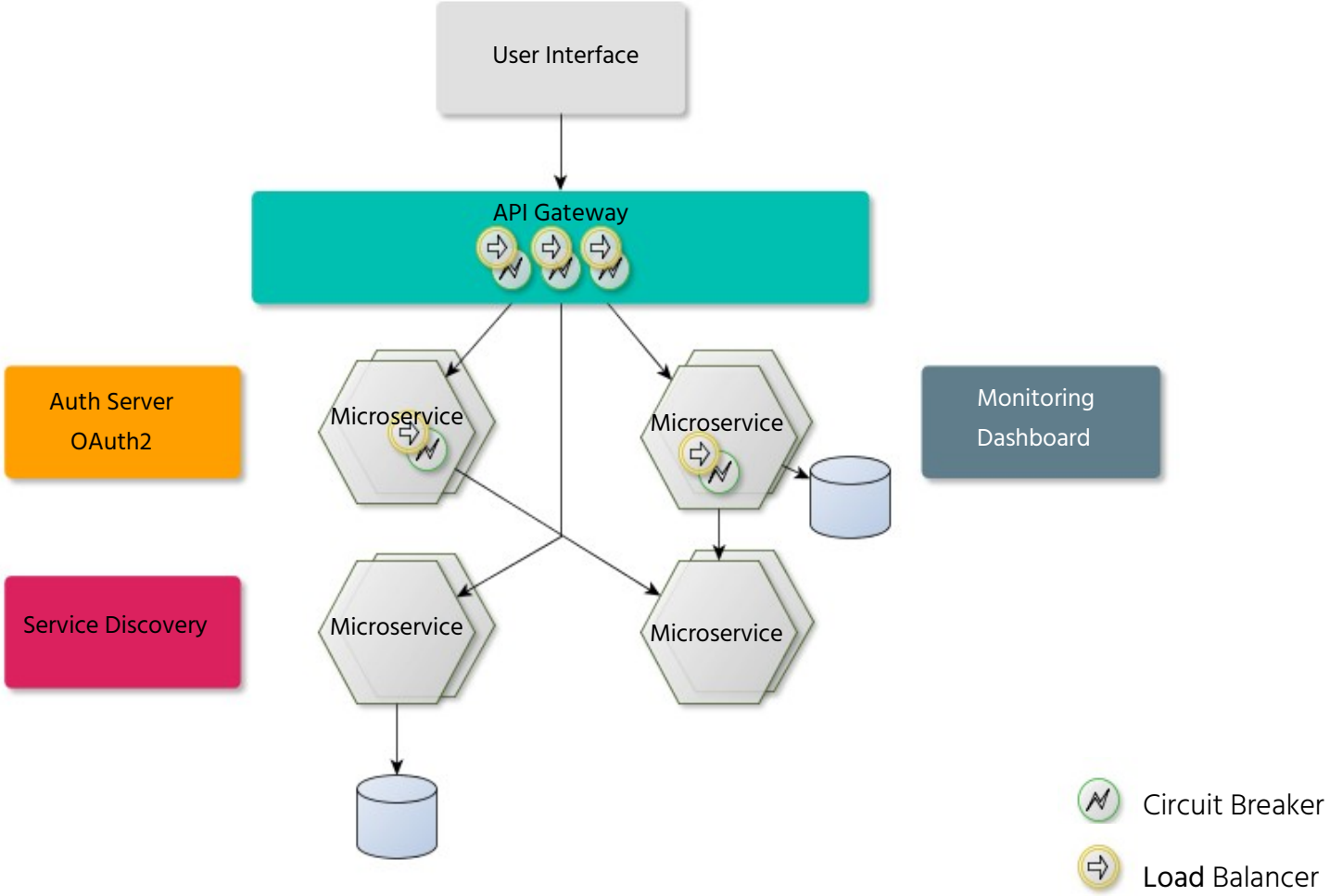




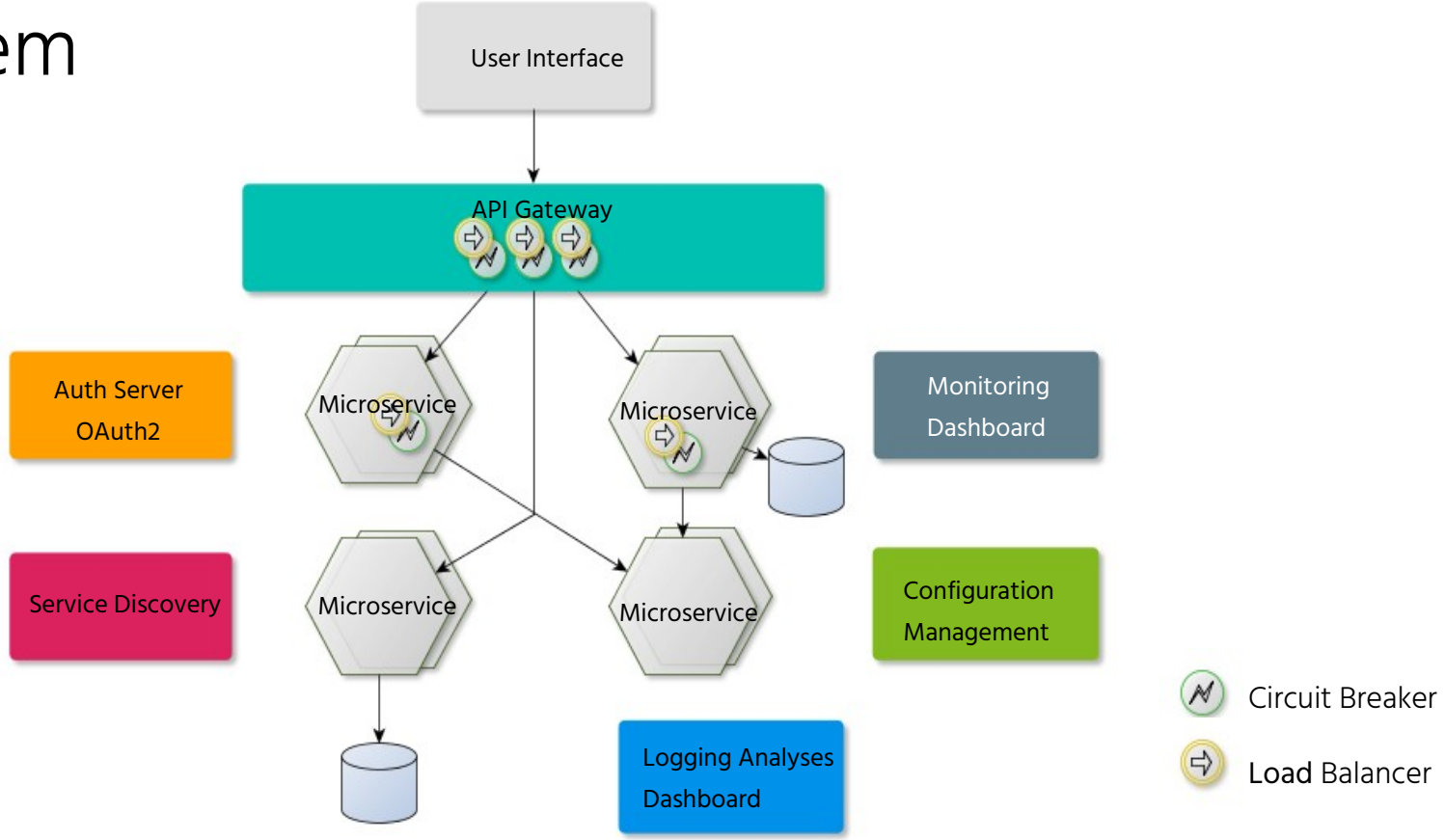



Source: http://www.programering.com/a/MDN3gzNwATE.html

# Current ecosystem so far …



# A lot to cover to establish a Microservices ecosystem




Build process, CI/CD-pipeline, testing, development environment

# Lessons learned

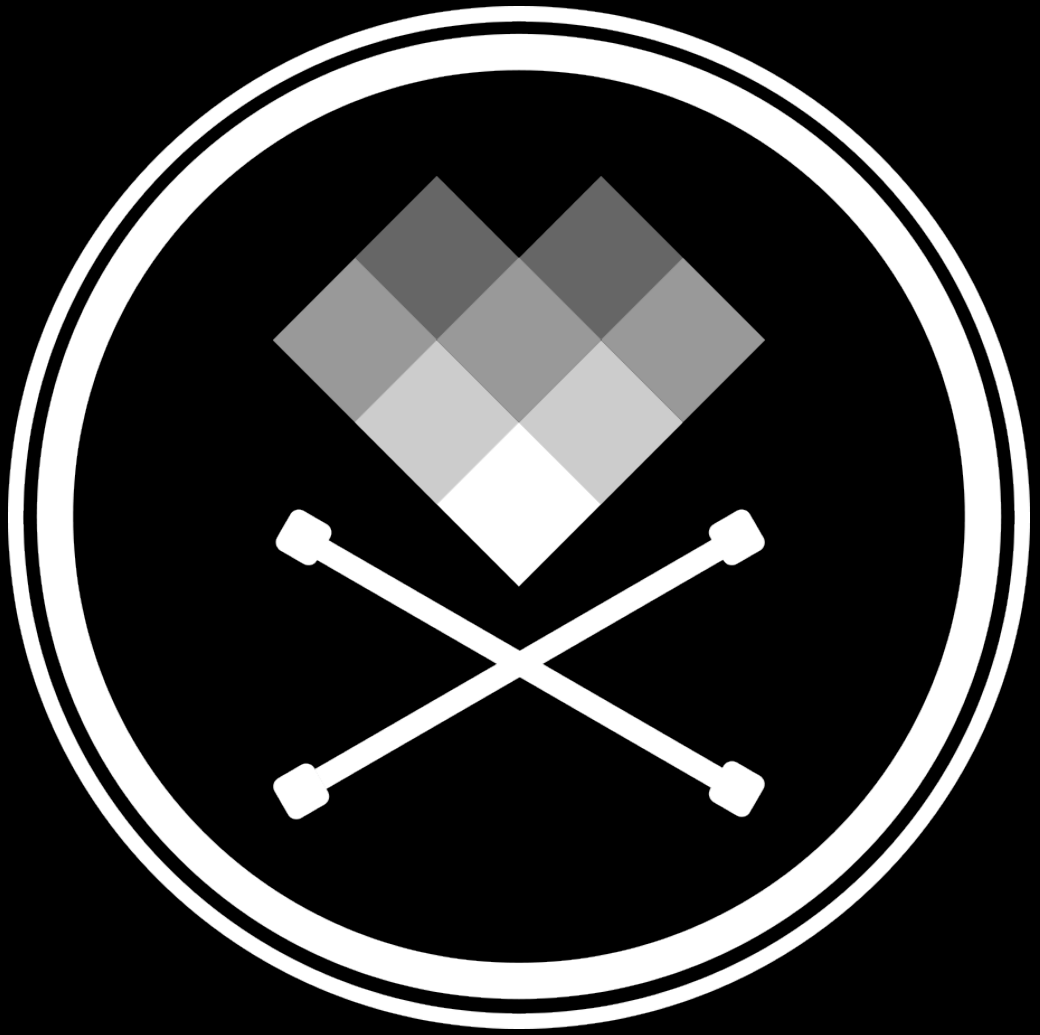
- Starting with decomposing big chunks frustrates
- Establishing Microservices ecosystem takes time and requires different skills & tools
- No explicit infrastructure team slows down the process
- A holistic picture of target architecture helps to stay focussed
- It takes far longer than originally anticipated

# Summary

- Start small & split in manageable steps
- Establish teams for Microservices & infrastructure
- Define a target architecture
- Transformation can be handled even with 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

goto; conference


Please

# Remember to rate this session

Thank you!

follow us @gotochgo