

The Microservices journey from a startup perspective

Susanne Kaiser
CTO
@suksr

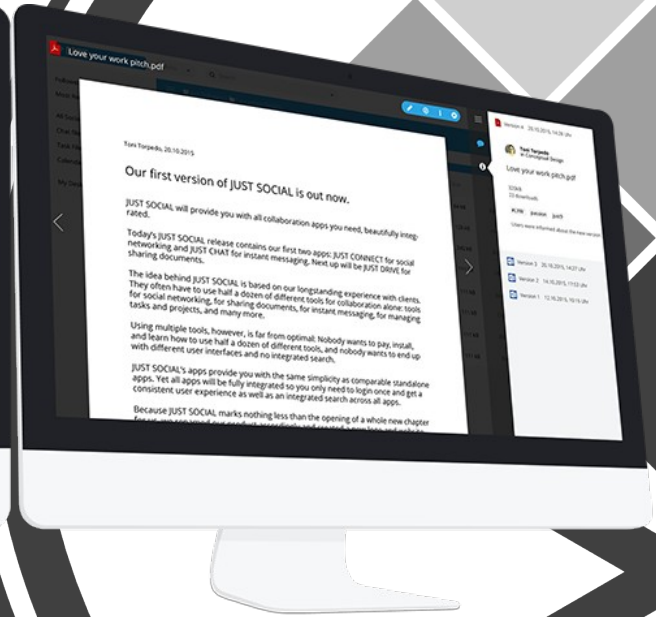
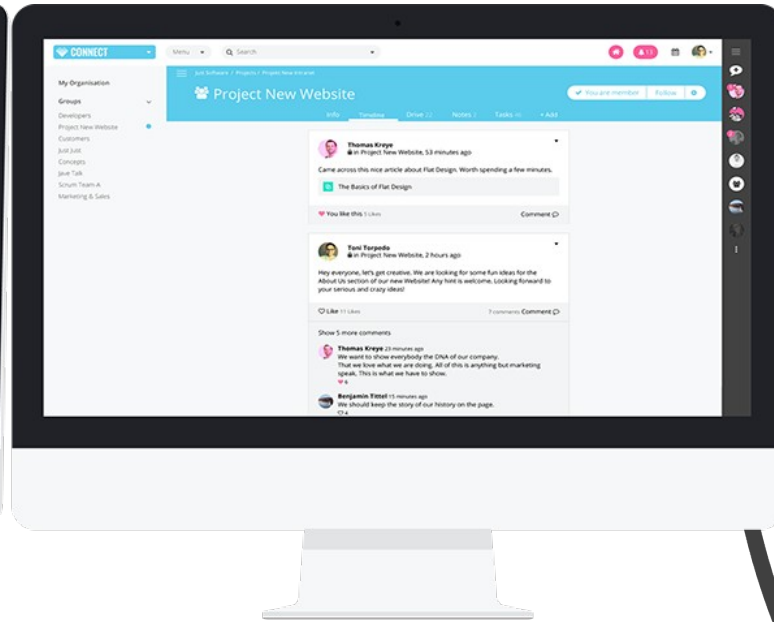
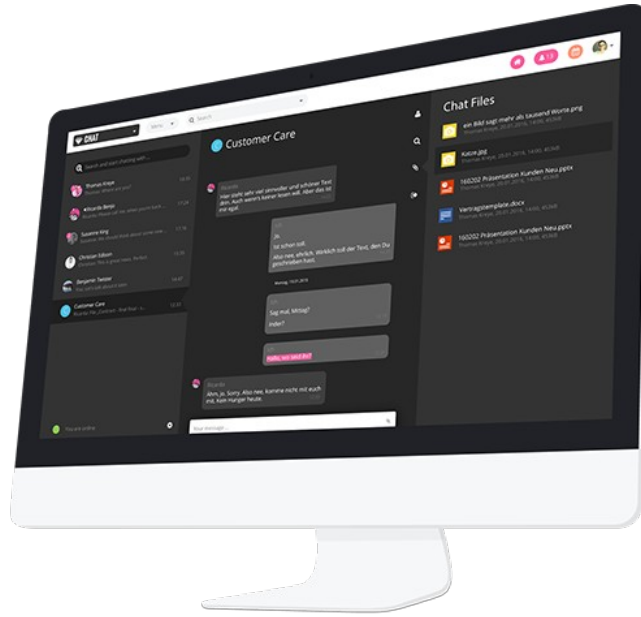
Just Software
@JustSocialApps



Each journey is different

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

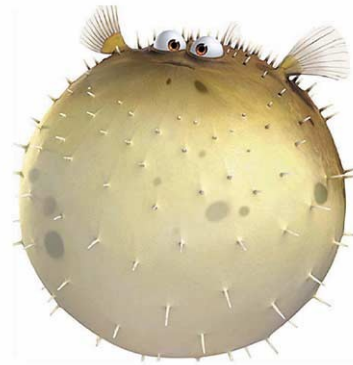
Adrian Cockcroft, AWS VP Cloud Architect,
former Netflix Chief Cloud Architect



 **JUST SOCIAL**

The beginning ... A monolith in every aspect

One team

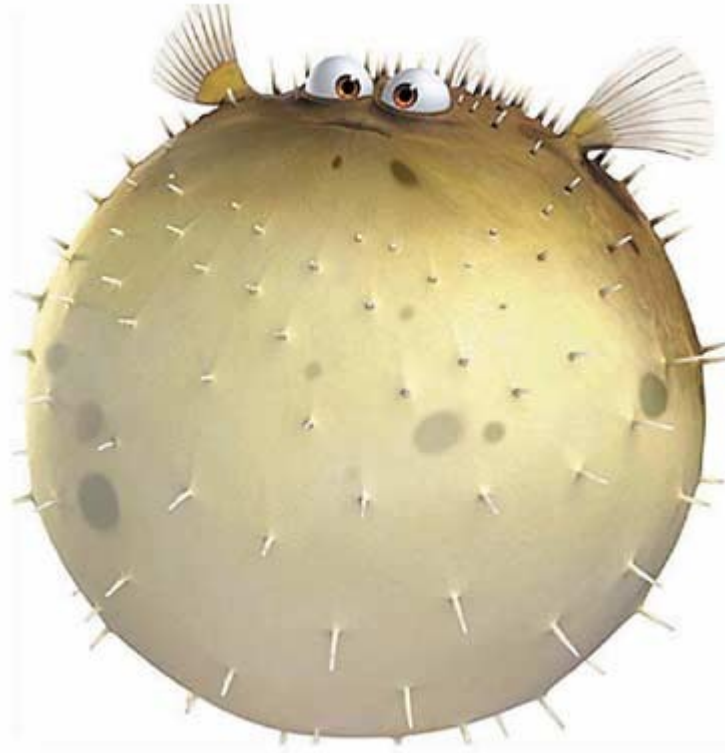


One
technology
stack

One collaboration product

After an evolving while ...

Productivity suffered



New features
released slowly

Usability and UX suffered

Separate Collaboration Apps



JUST CONNECT
Social Network



JUST CHAT
Real-time collaboration



JUST WIKI
Knowledge Base



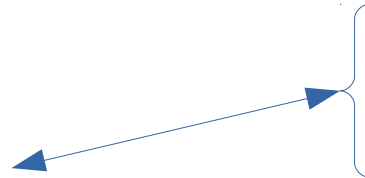
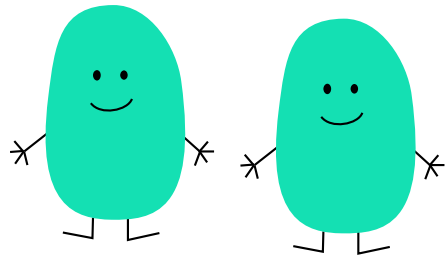
JUST DRIVE
Document Sharing



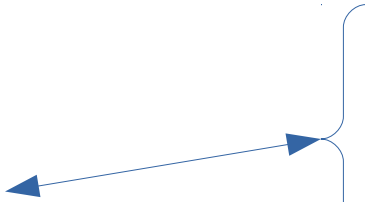
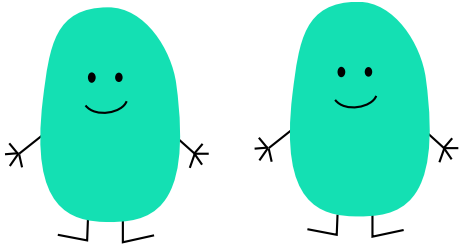
JUST TASKS
Task Management

Small, autonomous teams

with well-defined responsibilities



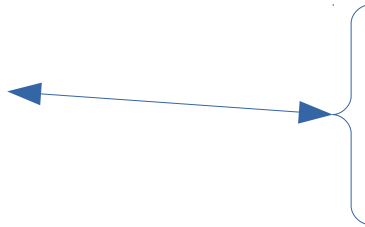
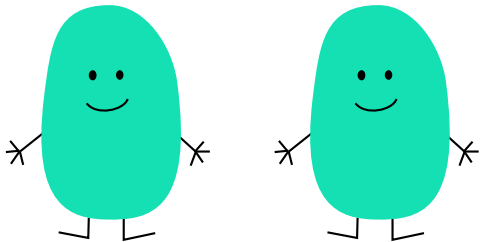
JUST CONNECT
Social Network



JUST CHAT
Real-time collaboration



JUST WIKI
Knowledge Base

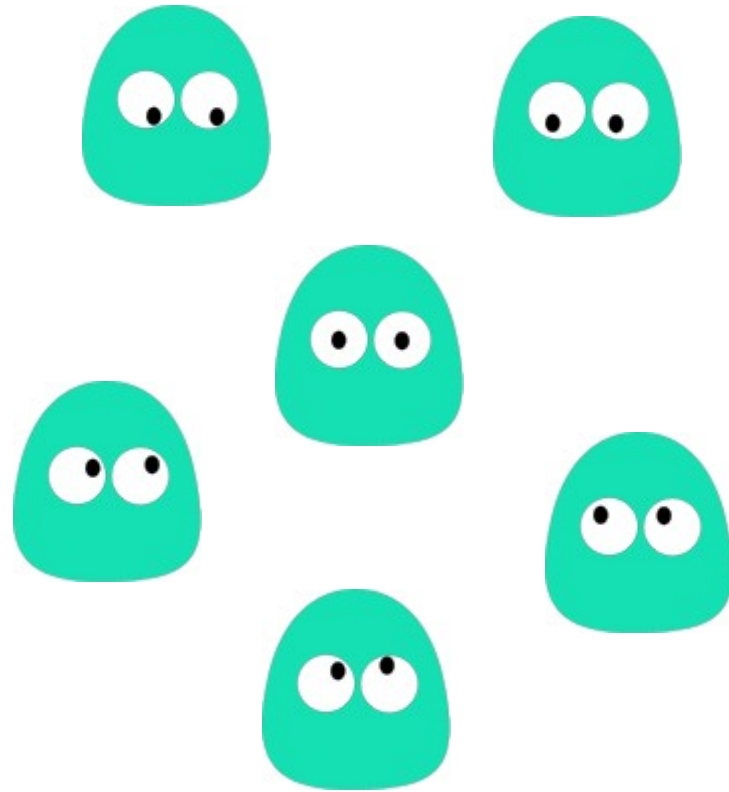


JUST DRIVE
Document Sharing



JUST TASKS
Task Management

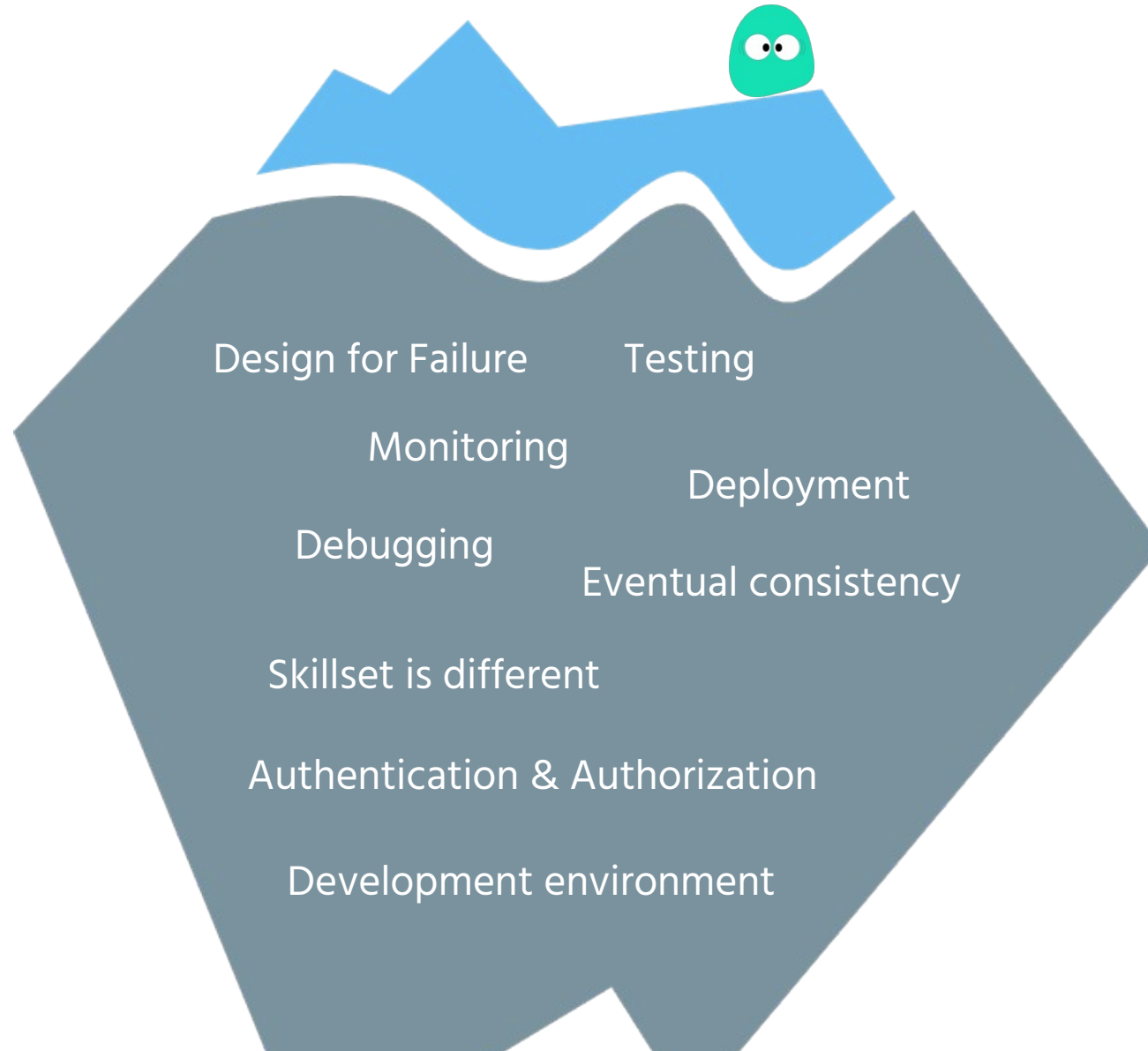
In the long run ...



Looks easy at first glance ...



Microservices come with complexity



Challenges of transformation

Transformation
takes longer than
anticipated



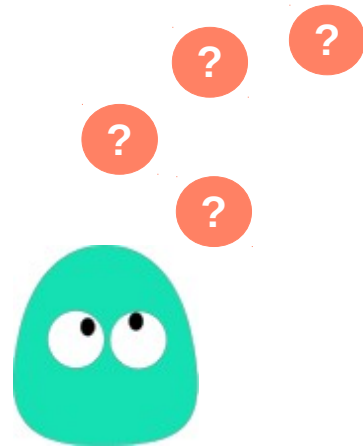
You still have to
take care of your
existing system

Core functionality
is hard to untangle

All involved parties need to
agree and to be on board

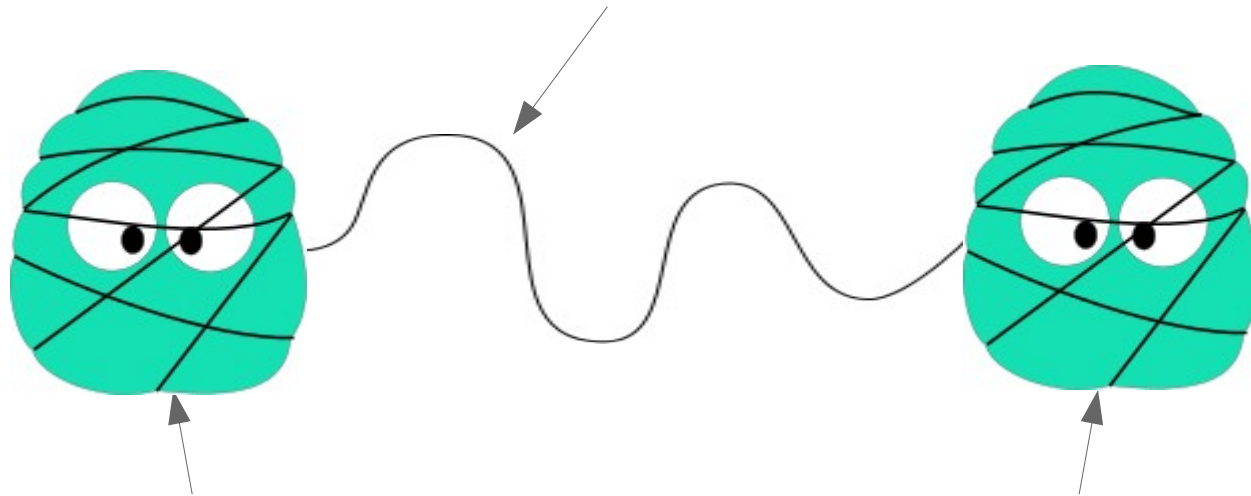
Don't partition without
a good reason.

How to start?



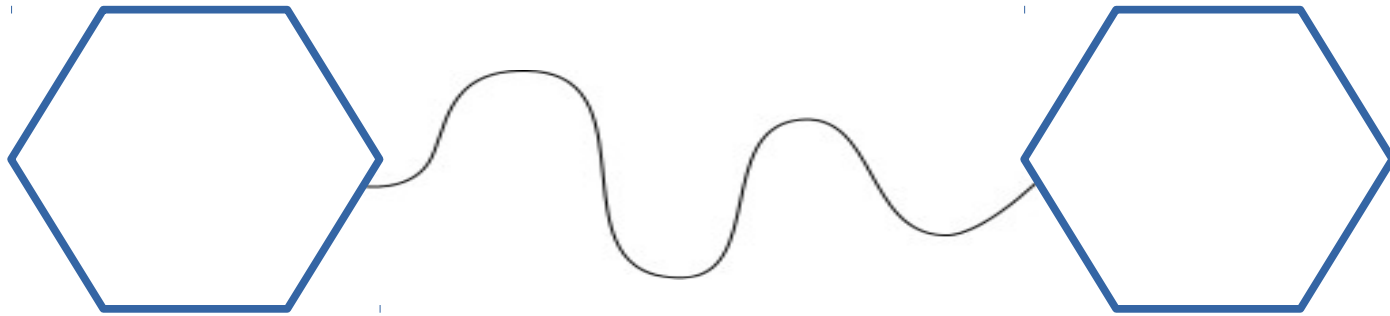
Key concepts of modelling Microservices

Loose coupling between services



High cohesion within a service

Identify Bounded Contexts



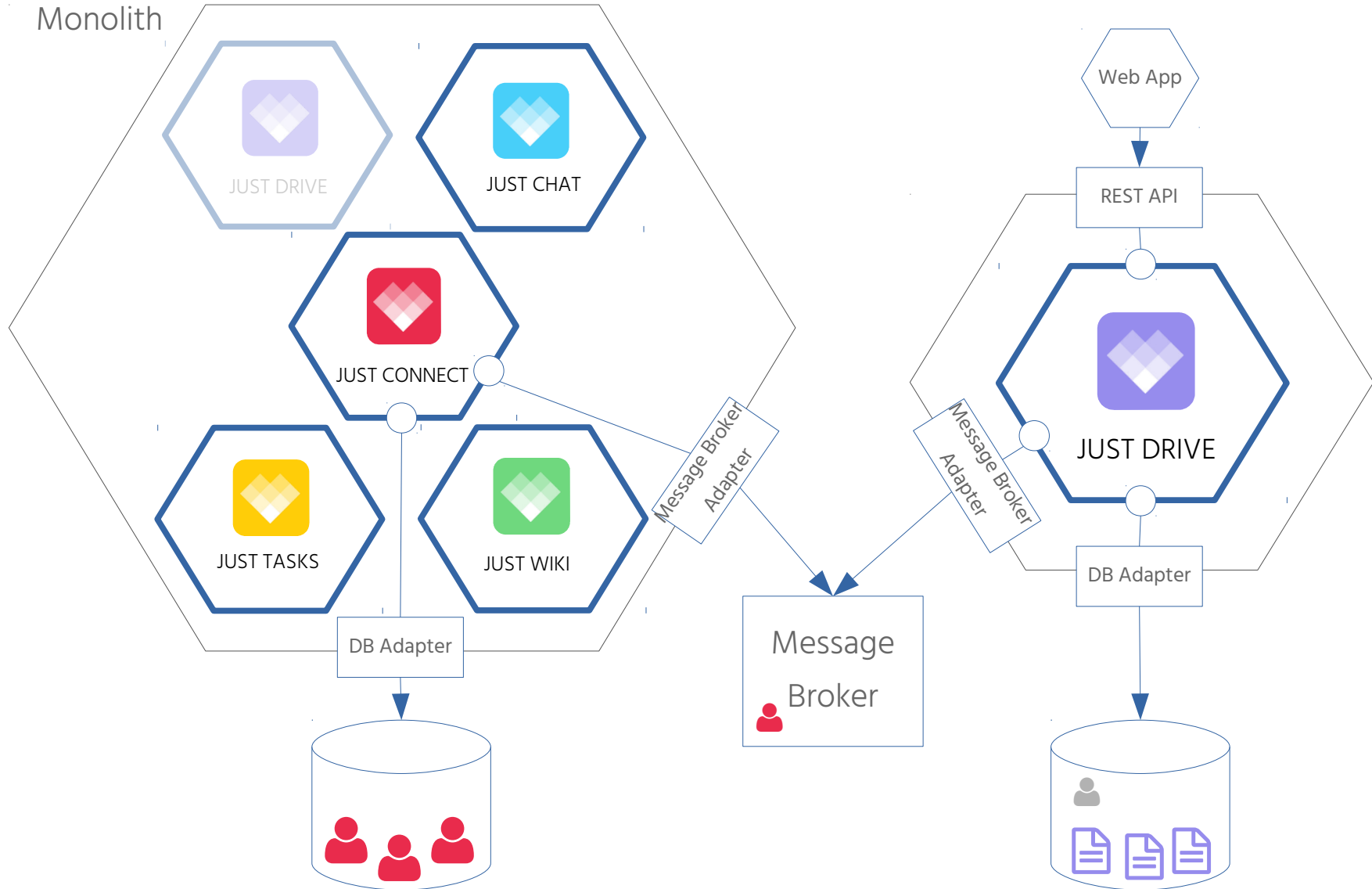
“A specific responsibility [..] enforced
with explicit boundaries”

<http://blog.sapiensworks.com/post/2012/04/17/DDD-The-Bounded-Context-Explained.aspx>

Bounded Contexts = Collaboration Apps



First approach as a co-existing service

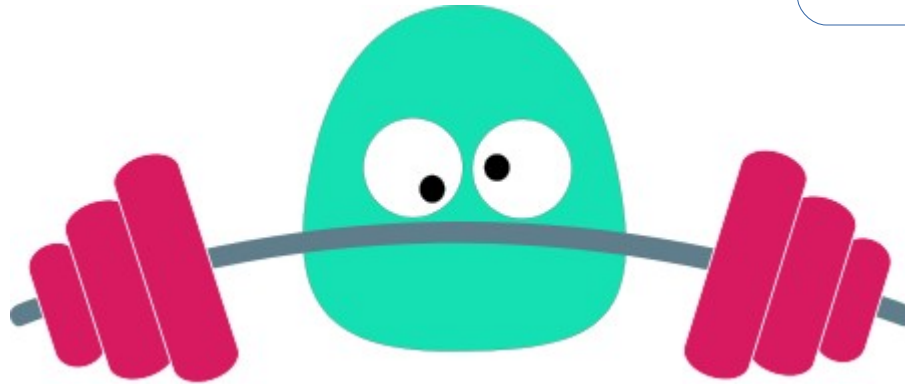


Heavy undertake if you do all at once

New UI

Maintain & run
current system

New data structure



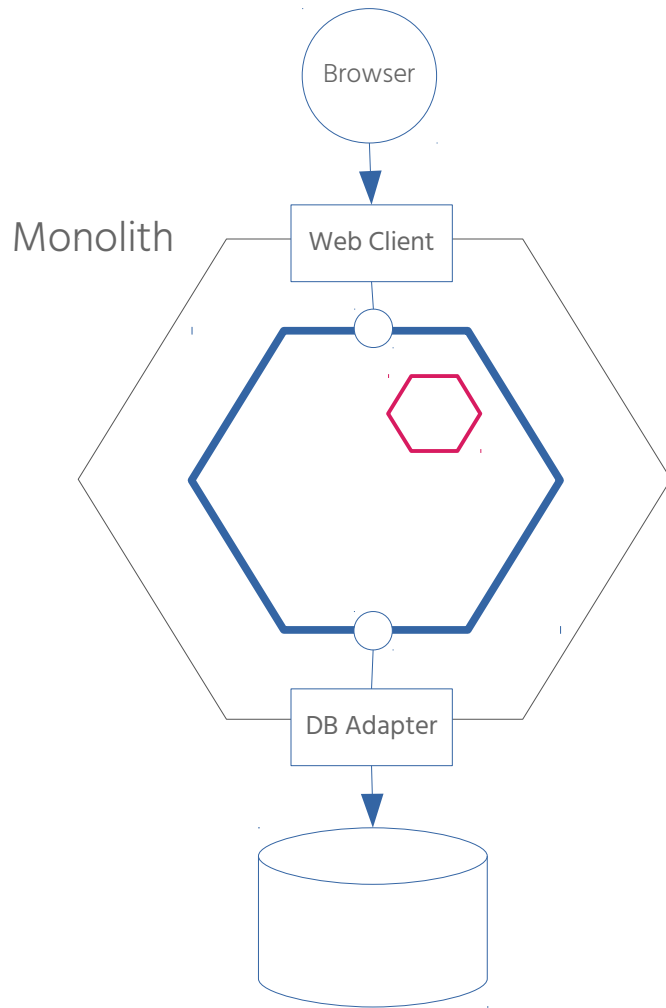
New technology

Extend skillset

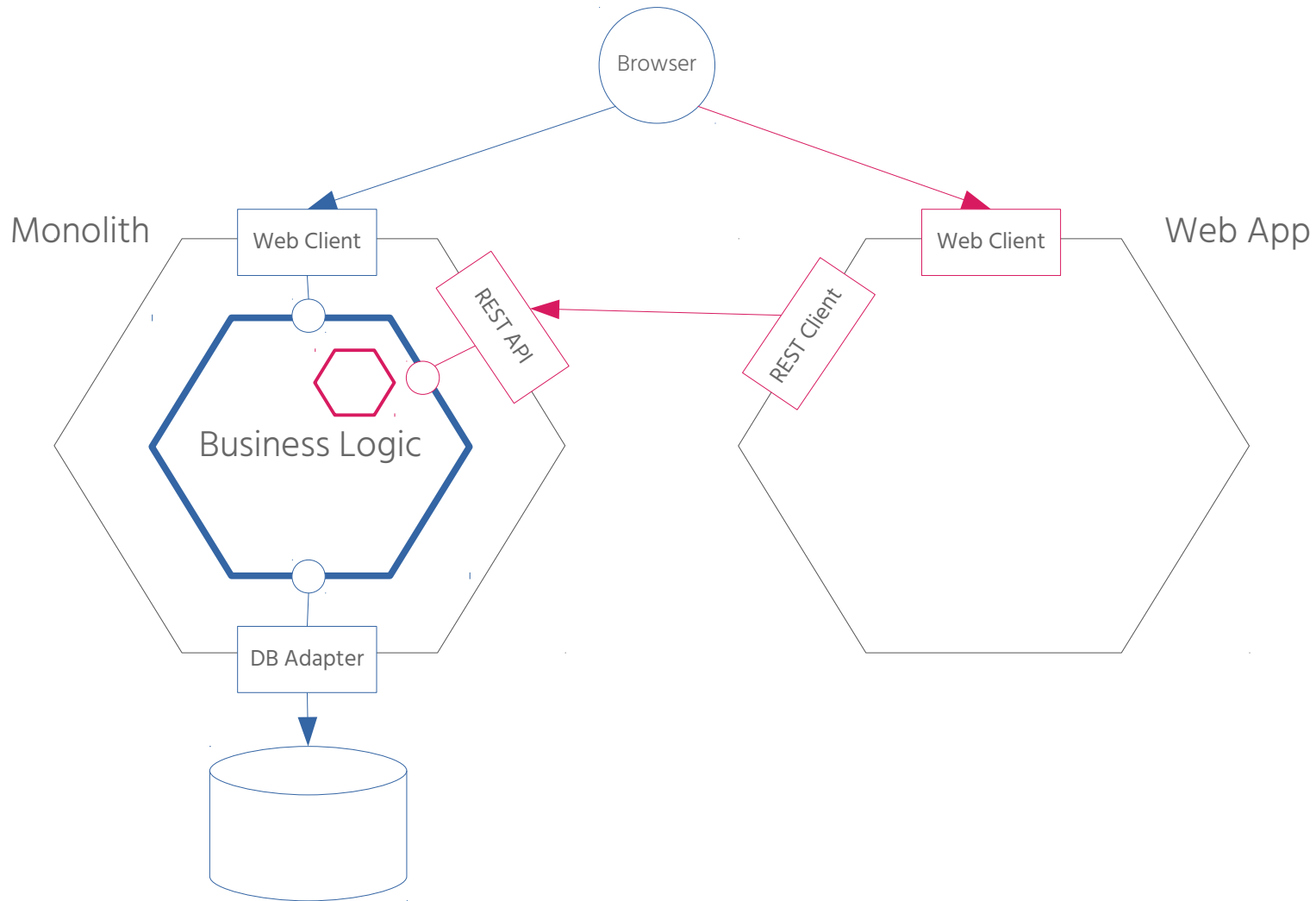
Split in steps – e.g. top down



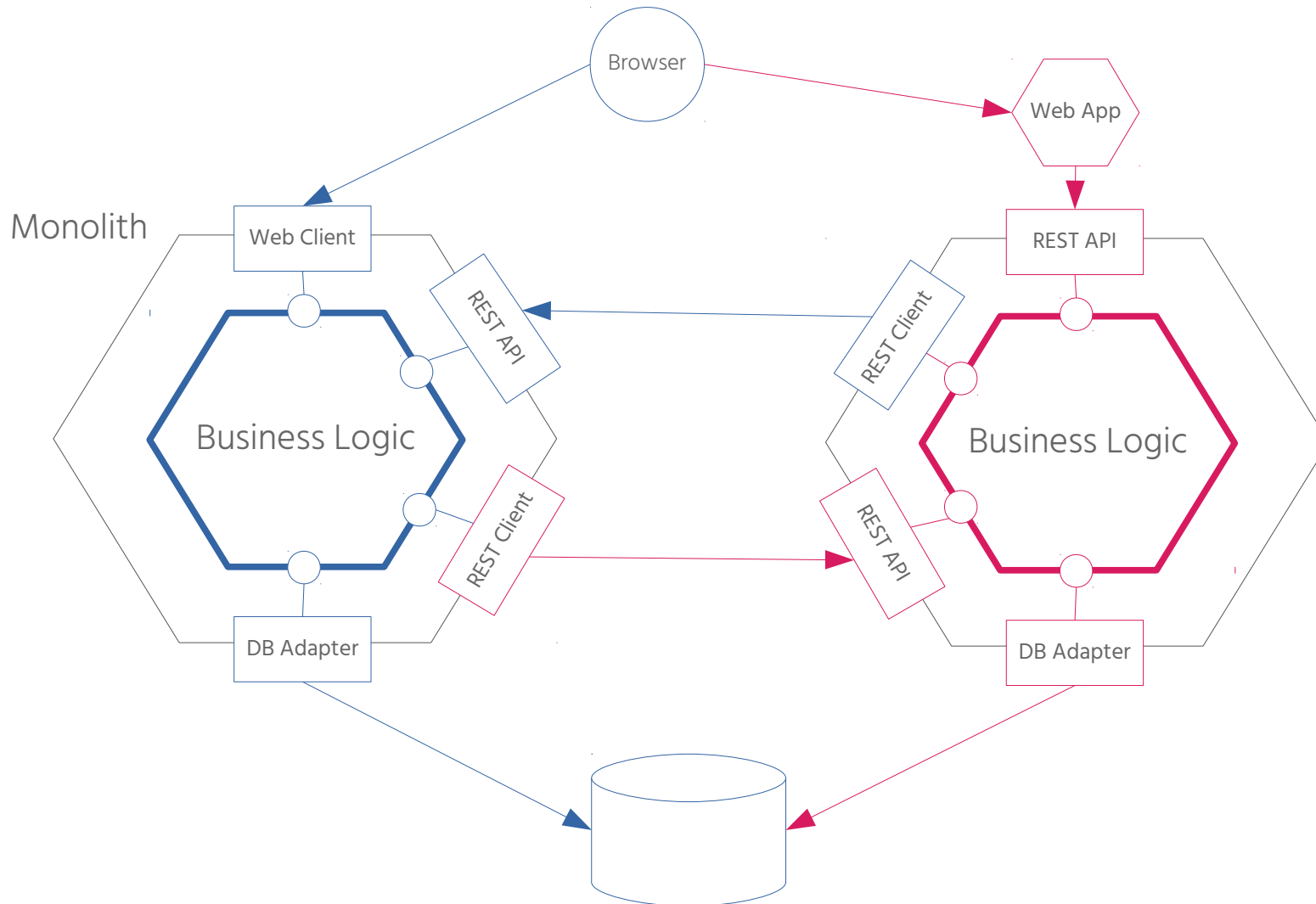
Split in steps – e.g. top down



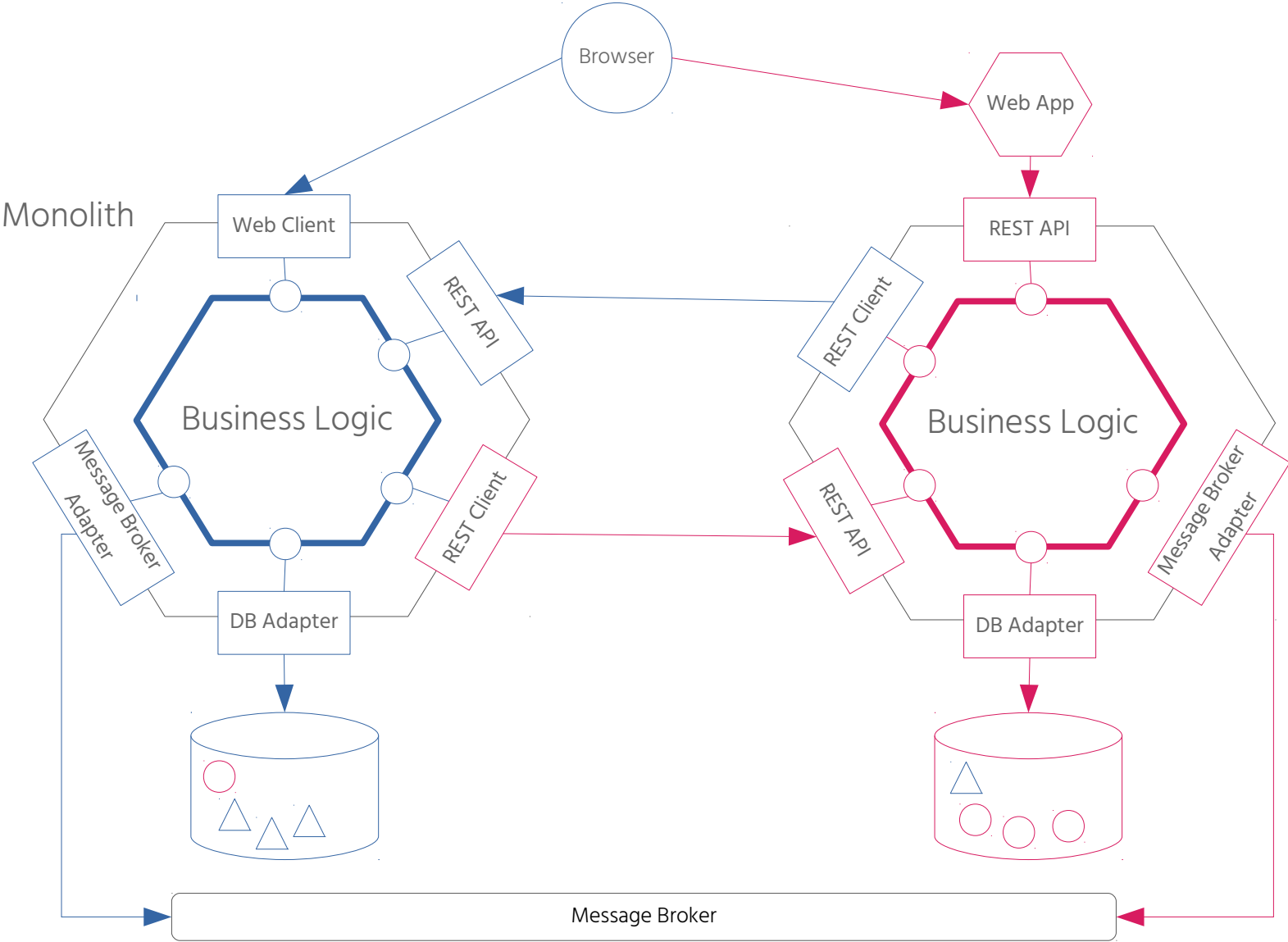
Split in steps – Step 1) Extracting Web App



Split in steps – Step 2) Extracting Business Logic



Split in steps – Step 3) Extracting Data Storage



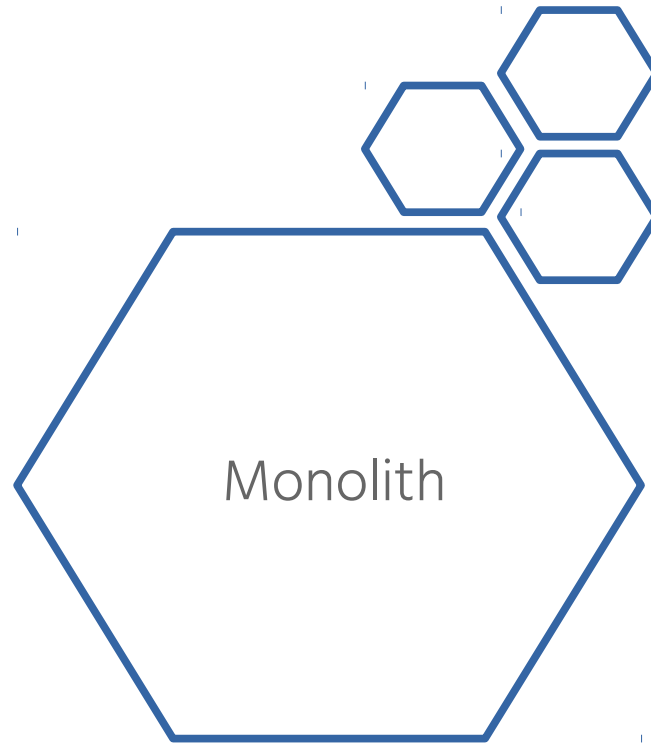
Which one first?

Easy to extract

Changing
frequently

Different resource
requirements

Stop feeding the monolith

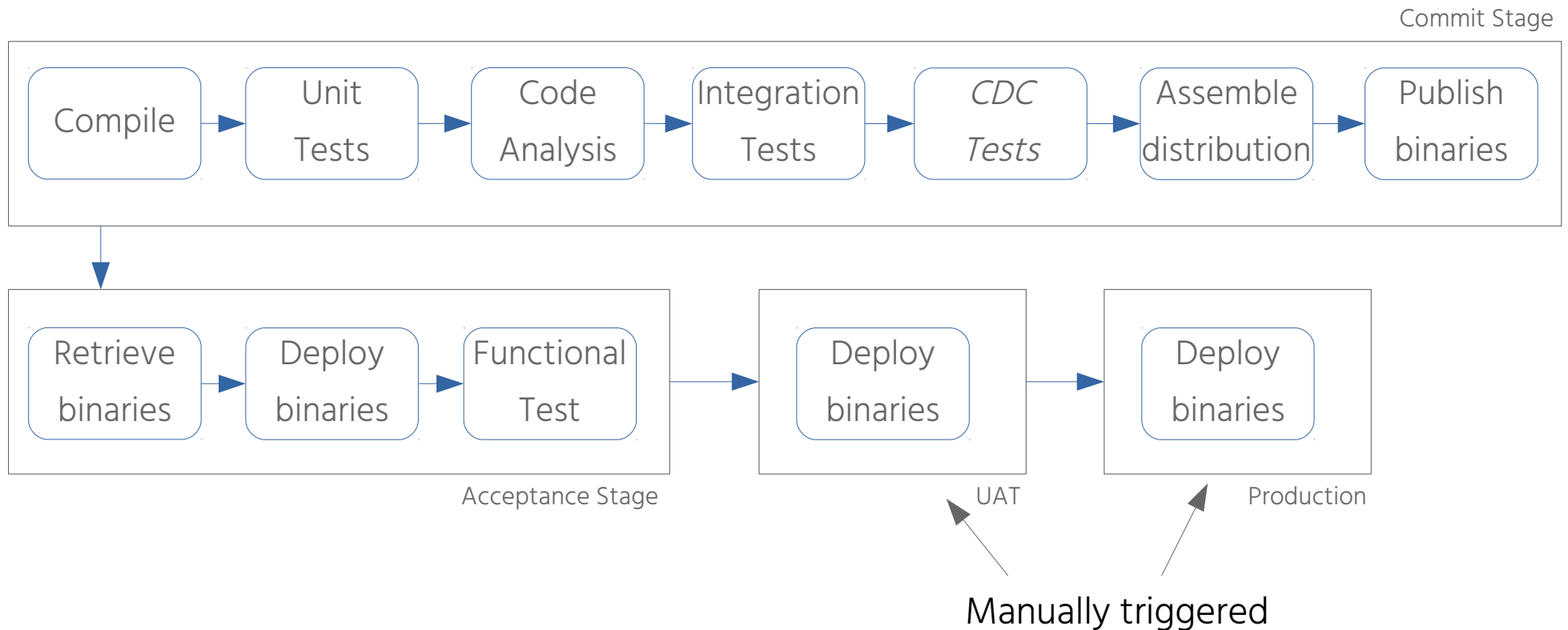


Get your CI/CD pipeline ready

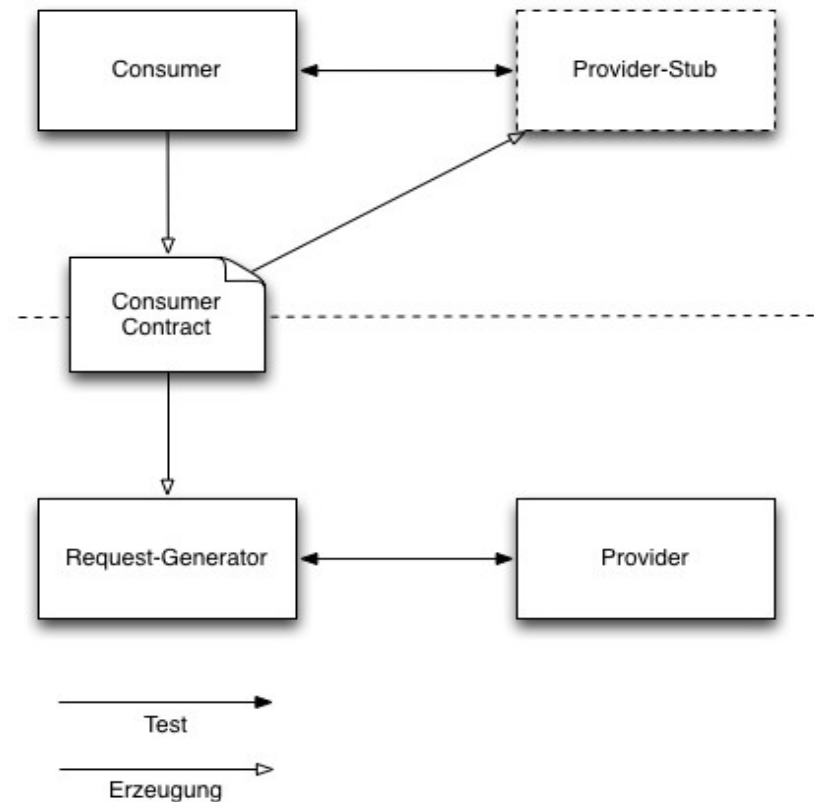


Jenkins

+



Testing - Consumer Driven Contract Tests with Pact



Source: <https://jaxenter.de/microservices-consumer-driven-contract-testing-mit-pact-20574>

Deployment in a polyglot environment

 Cloud



Service Instance per Host



Each service instance runs in complete isolation

 On-Premises



Multiple Service Instances per Host/Process



Efficient resource utilization



debian



redhat



ANSIBLE

Design for Failure

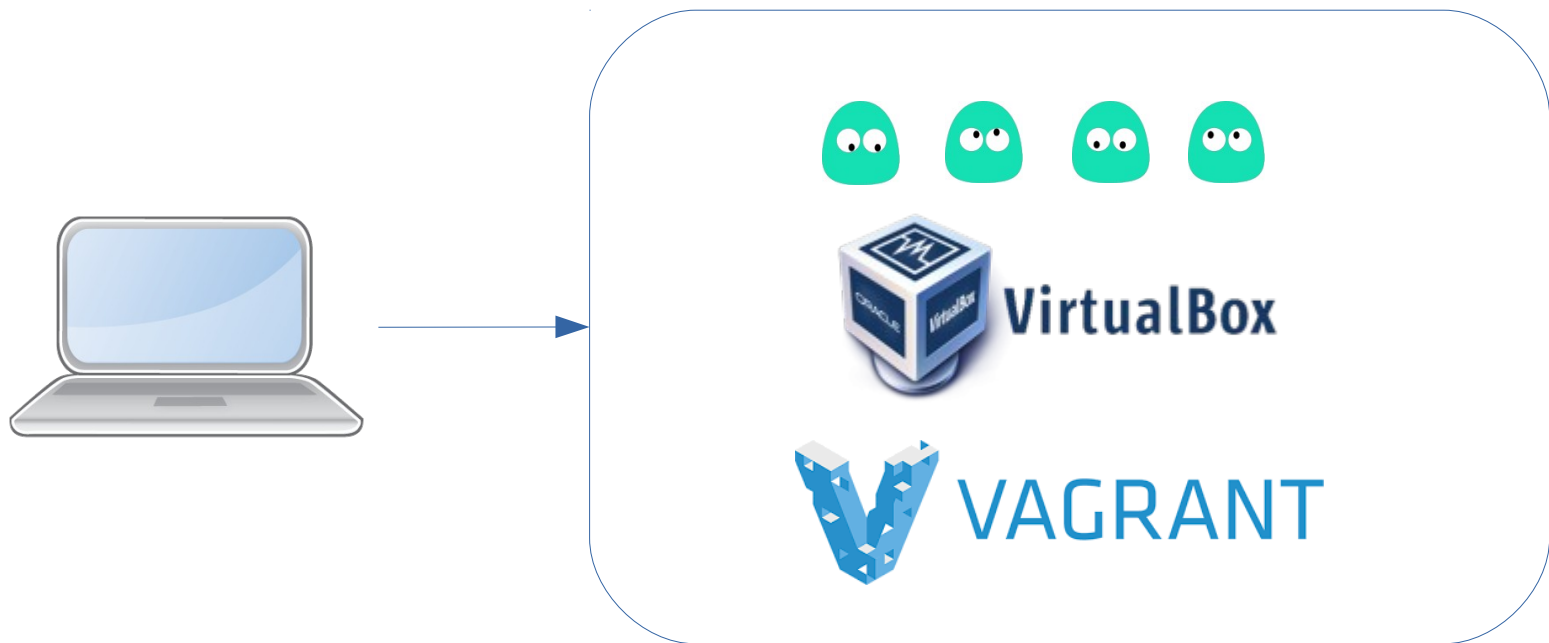
 Timeout-Handling

 Provide fallbacks

 Limit number of outstanding requests

 Circuit Breakers

Development environment



Full access to the entire system

Summary



Don't partition without a good reason



Identify your bounded contexts



Split in manageable steps

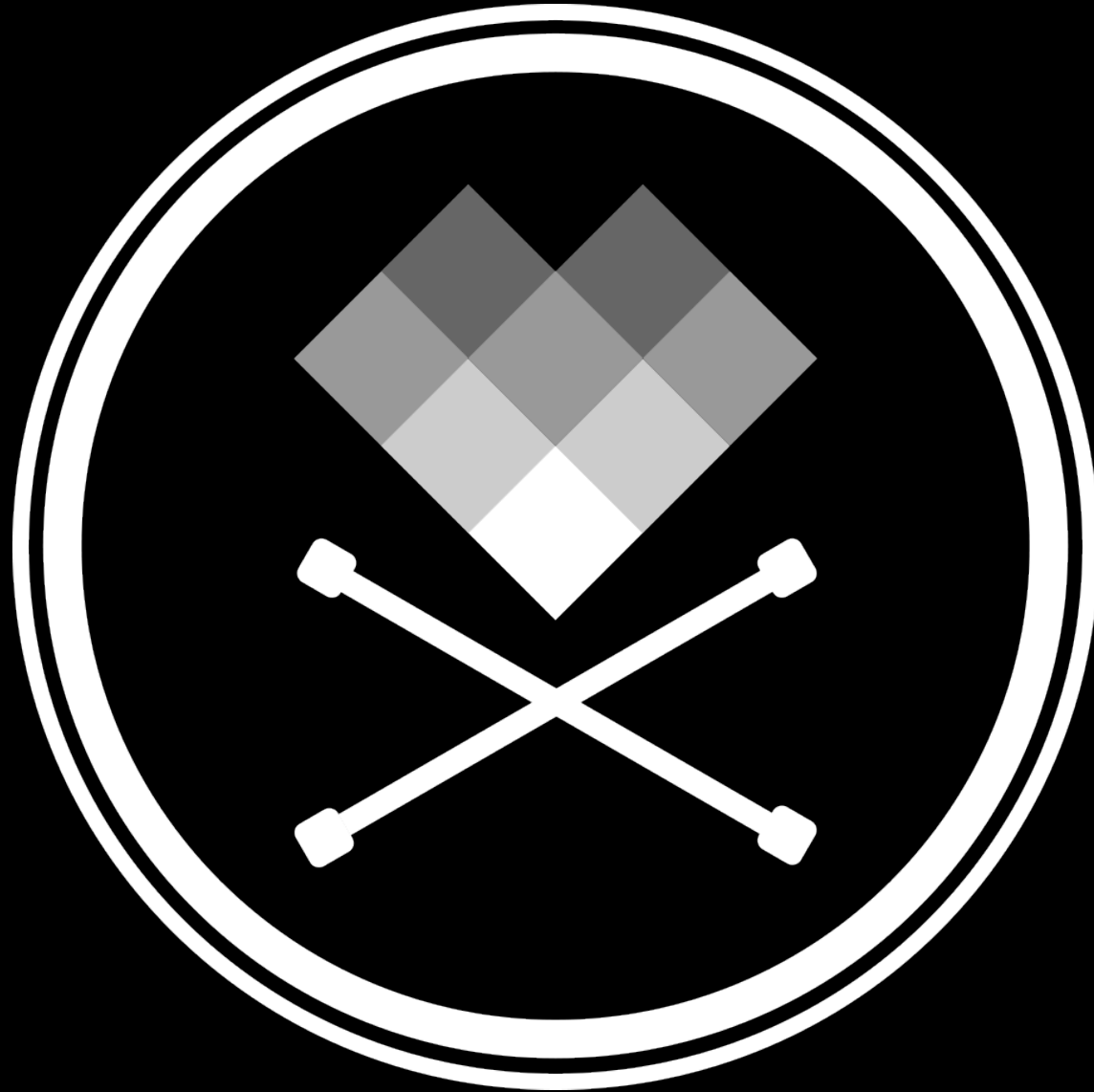


Get your build and delivery pipeline ready



It's not only a technical but also a cultural shift

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



... AND W/ MICROSERVICES !



THANK YOU!

Susanne Kaiser
CTO
@suksr

Just Software
@JustSocialApps