

Architecture Inception Canvas

Der kollaborative Kickstart für eine effektive
Software-Architekturarbeit



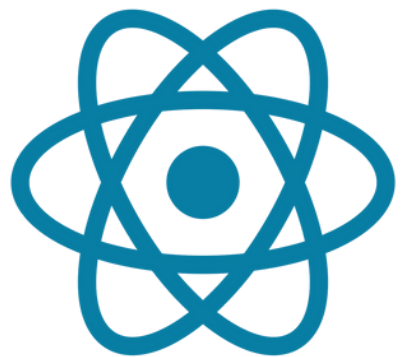
29. Oktober 2025

Bern



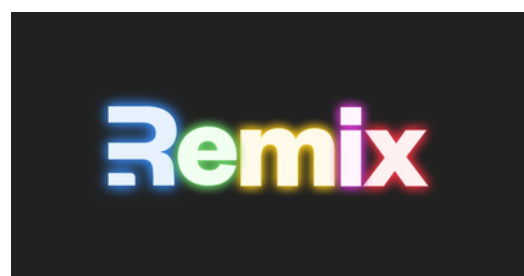
Patrick Roos





`</>` htmx

vaadin^{NX}`}``>`



NEXT.js



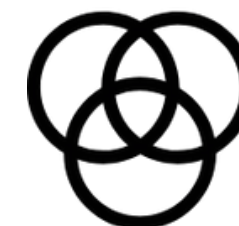
NuxtJS



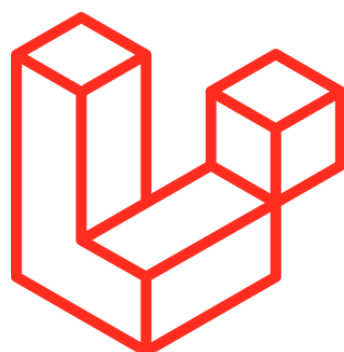
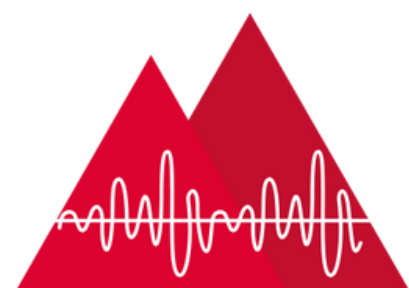
astro



Alpine.js



SOLID



SVELTEKIT

TURBOREPO

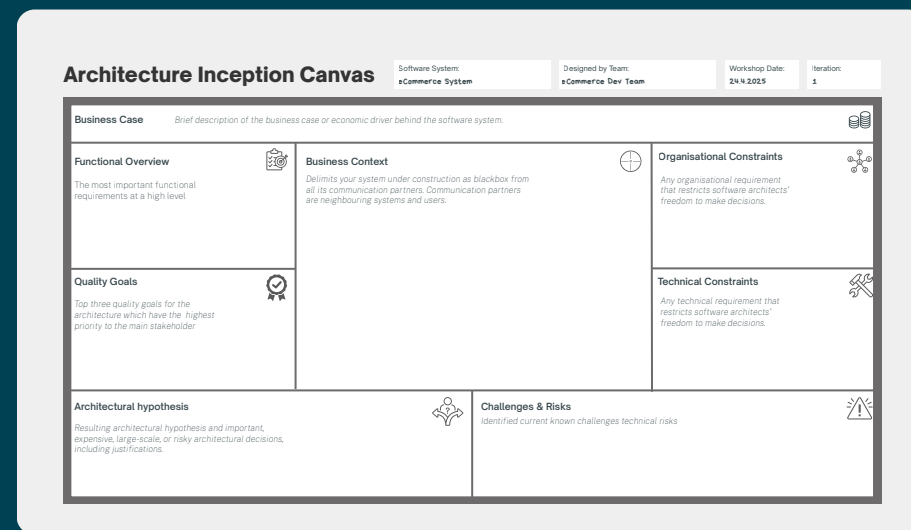
stencil



Quik

Architecture Inception Canvas

Der kollaborative Kickstart für eine effektive
Software-Architekturarbeit



The image shows a template for the Architecture Inception Canvas. It is a structured form with several sections for collaborative work. At the top, it has fields for 'Software System' (eCommerce System), 'Designed by Team' (eCommerce Dev Team), 'Workshop Date' (24.4.2025), and 'Iteration' (1). The main body is divided into six sections: 'Business Case' (Brief description of the business case or economic driver behind the software system), 'Functional Overview' (The most important functional requirements at a high level), 'Business Context' (Limits your system under construction as blackbox from all its communication partners. Communication partners are neighbouring systems and users), 'Organisational Constraints' (Any organisational requirement that restricts software architects' freedom to make decisions), 'Quality Goals' (Top three quality goals for the architecture which have the highest priority to the main stakeholder), and 'Technical Constraints' (Any technical requirement that restricts software architects' freedom to make decisions). At the bottom, there are two sections: 'Architectural hypothesis' (Resulting architectural hypothesis and important, expensive, large-scale, or risky architectural decisions, including justifications) and 'Challenges & Risks' (Identified current known challenges technical risks).



29. Oktober 2025

Bern

Patrick Roos

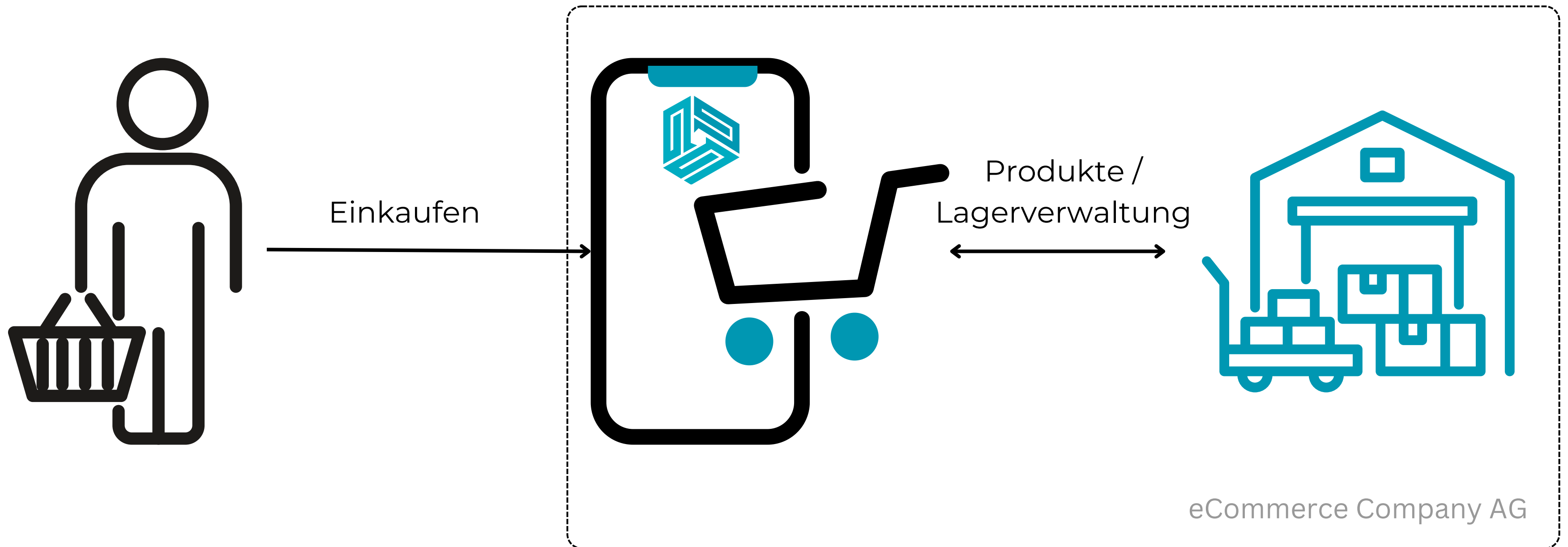
Patrick Roos

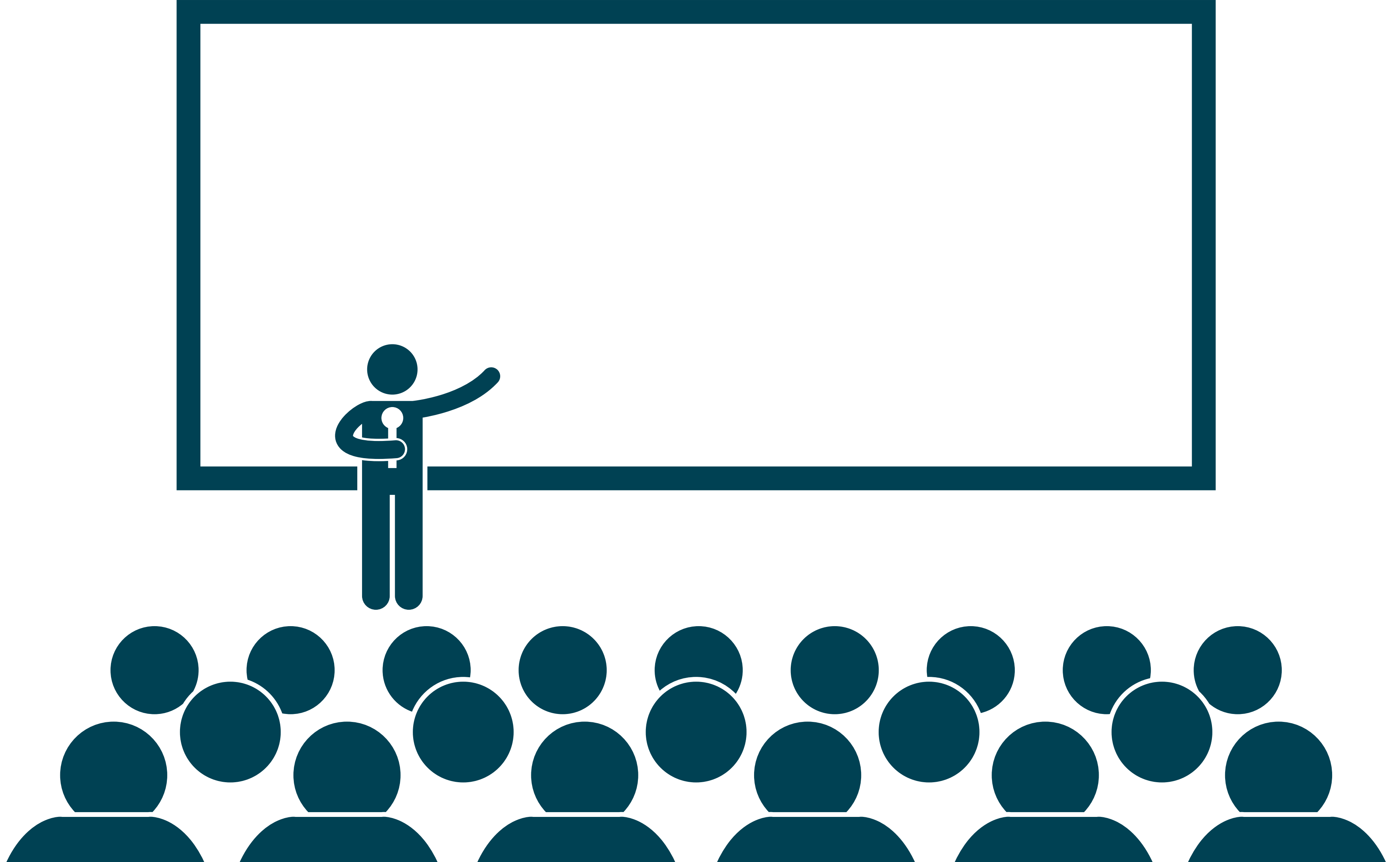


- Tech-Lead, Software Architect, Software Engineer bei Fellow GmbH
- Aktuelle Schwerpunkte
 - Hands-on Unterstützung bei der Entwicklung von SaaS Produkten und Modernisierungsinitiativen
- Seitenprojekte
 - Dozent Hochschule Luzern CAS Software Architecture
 - Blog: workingsoftware.dev
 - Architecture Inception Canvas, Frontend Architecture Map
 - Co-Host Heise Software ArchitekTOUR Podcast
- LinkedIn | X | GitHub



Case eCommerce Company

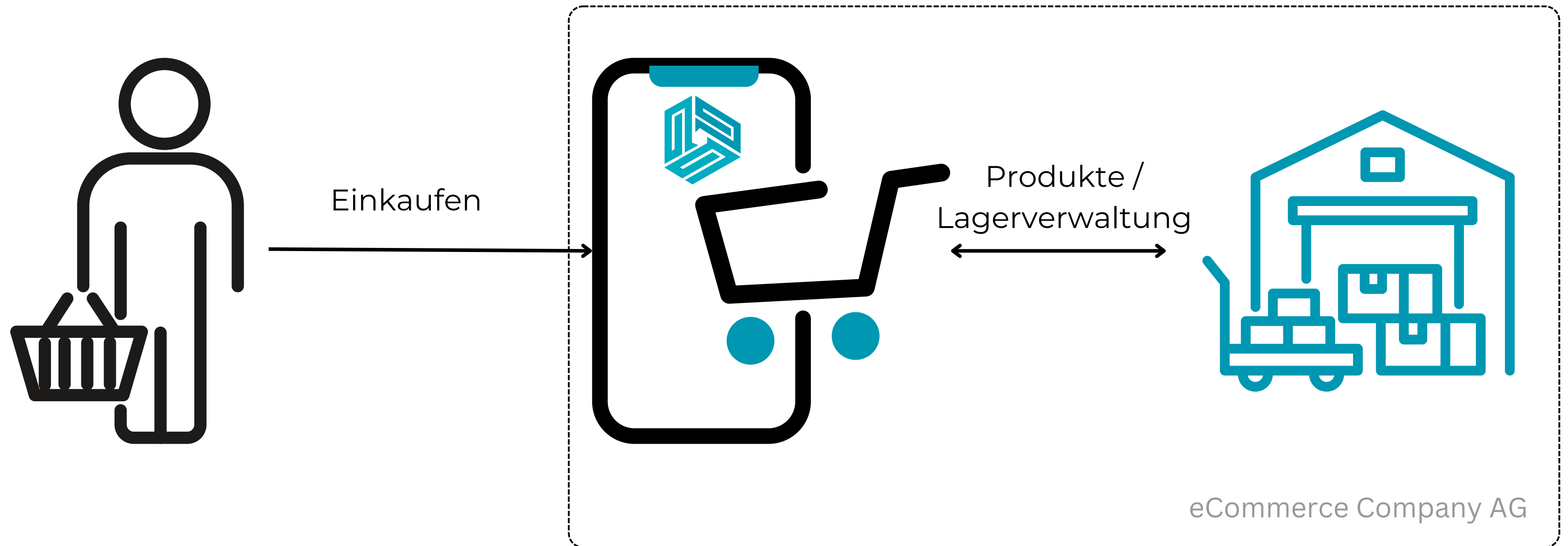




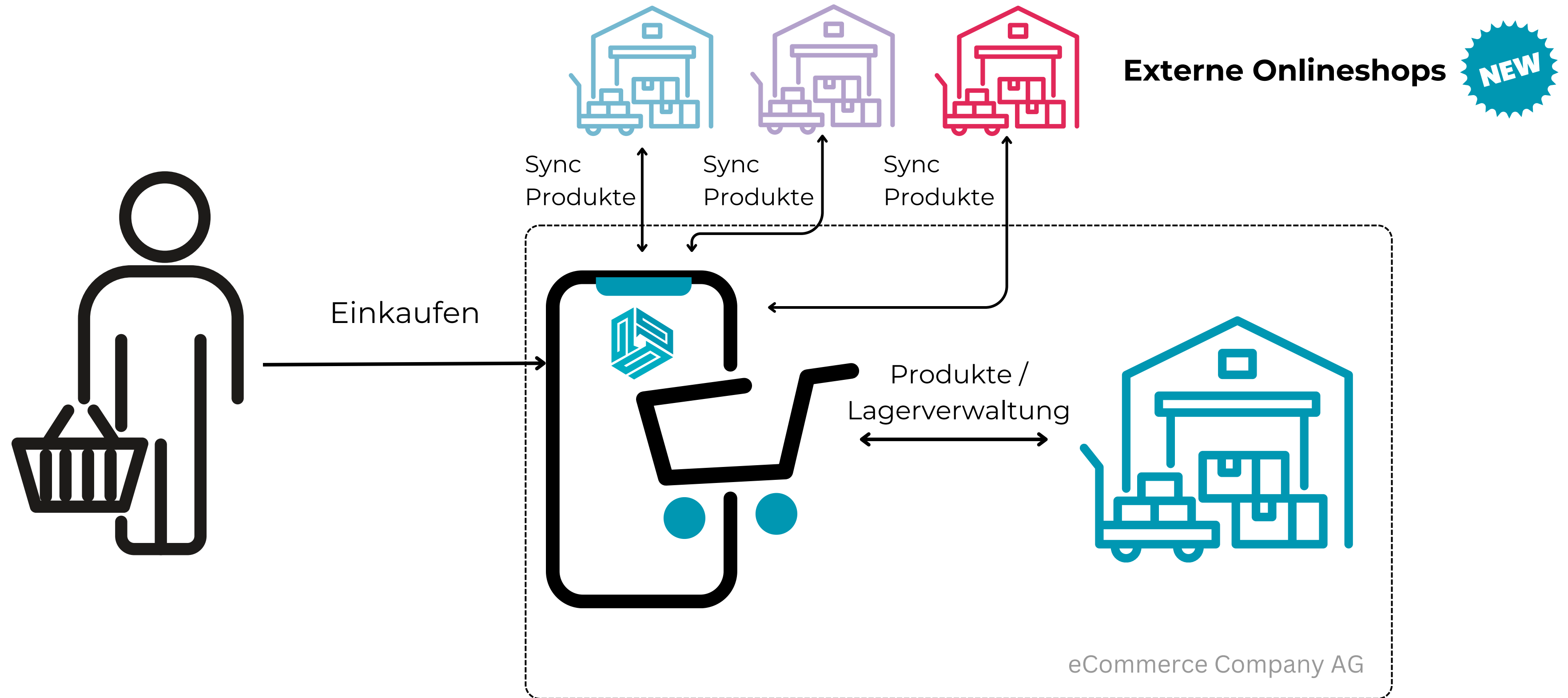
Neue Initiative...



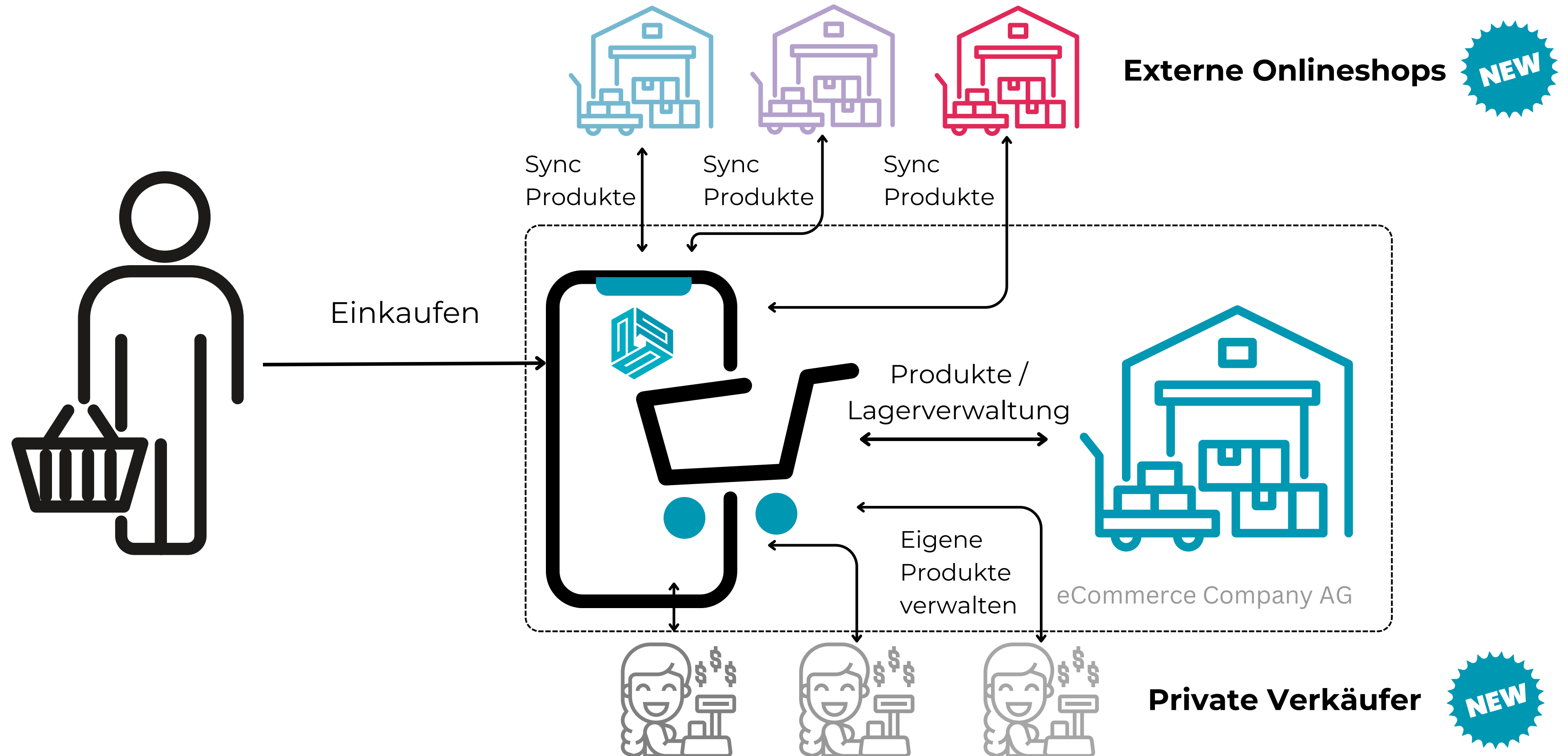
Wir erweitern unsere Software...

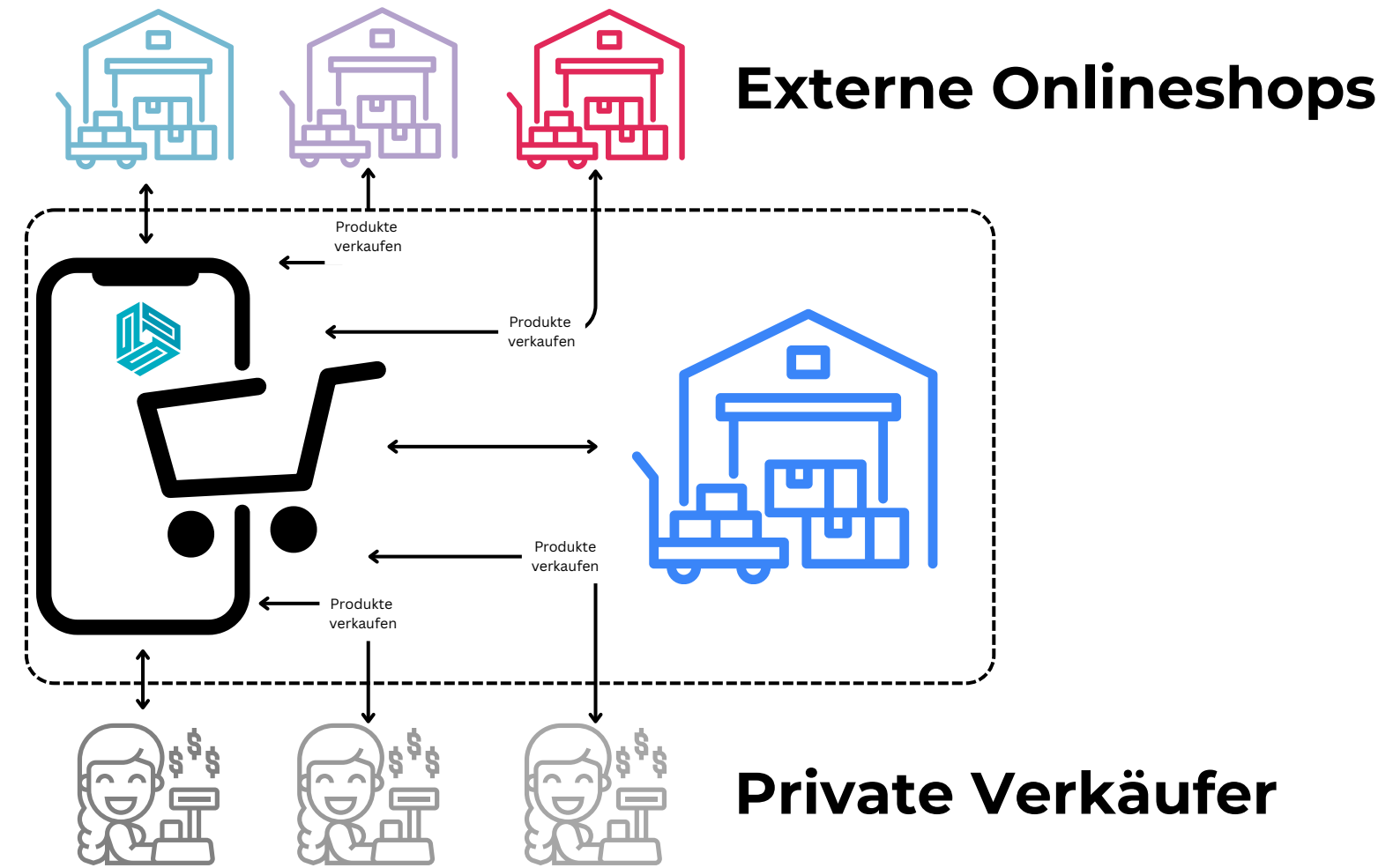


Wir erweitern unsere Software..

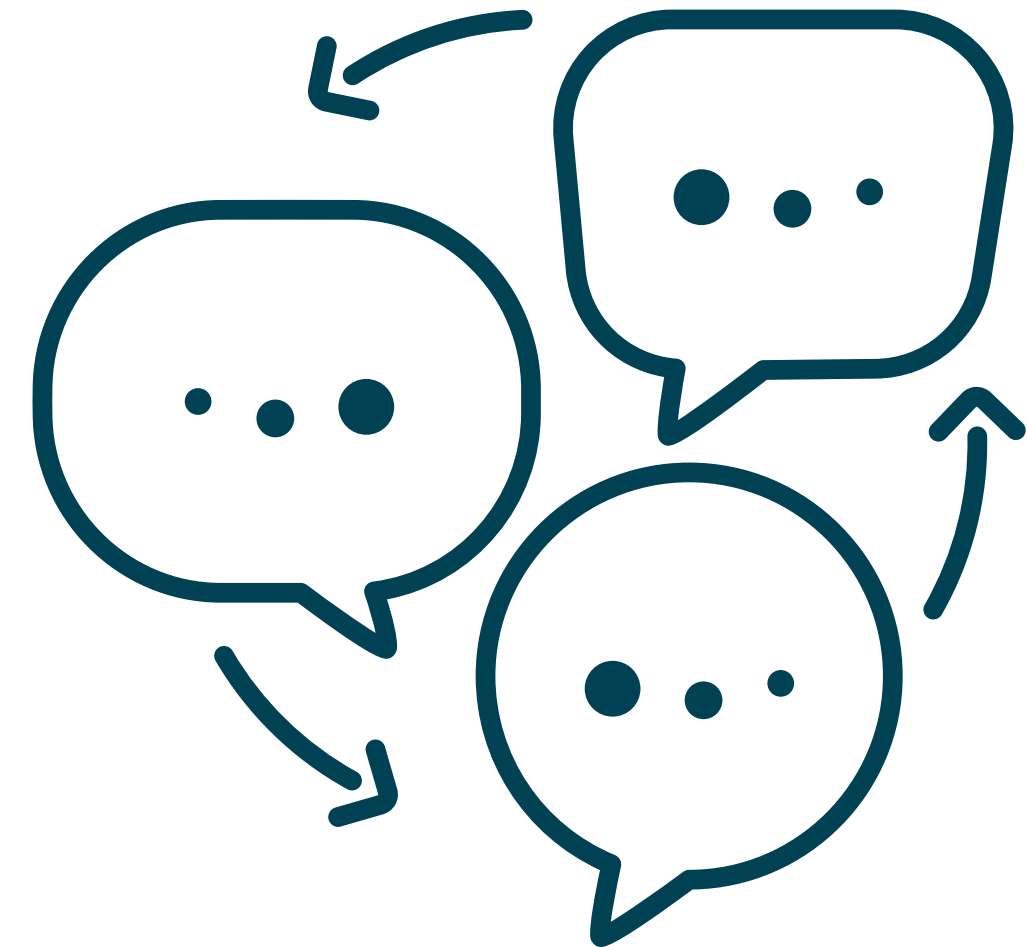


Wir erweitern unsere Software..





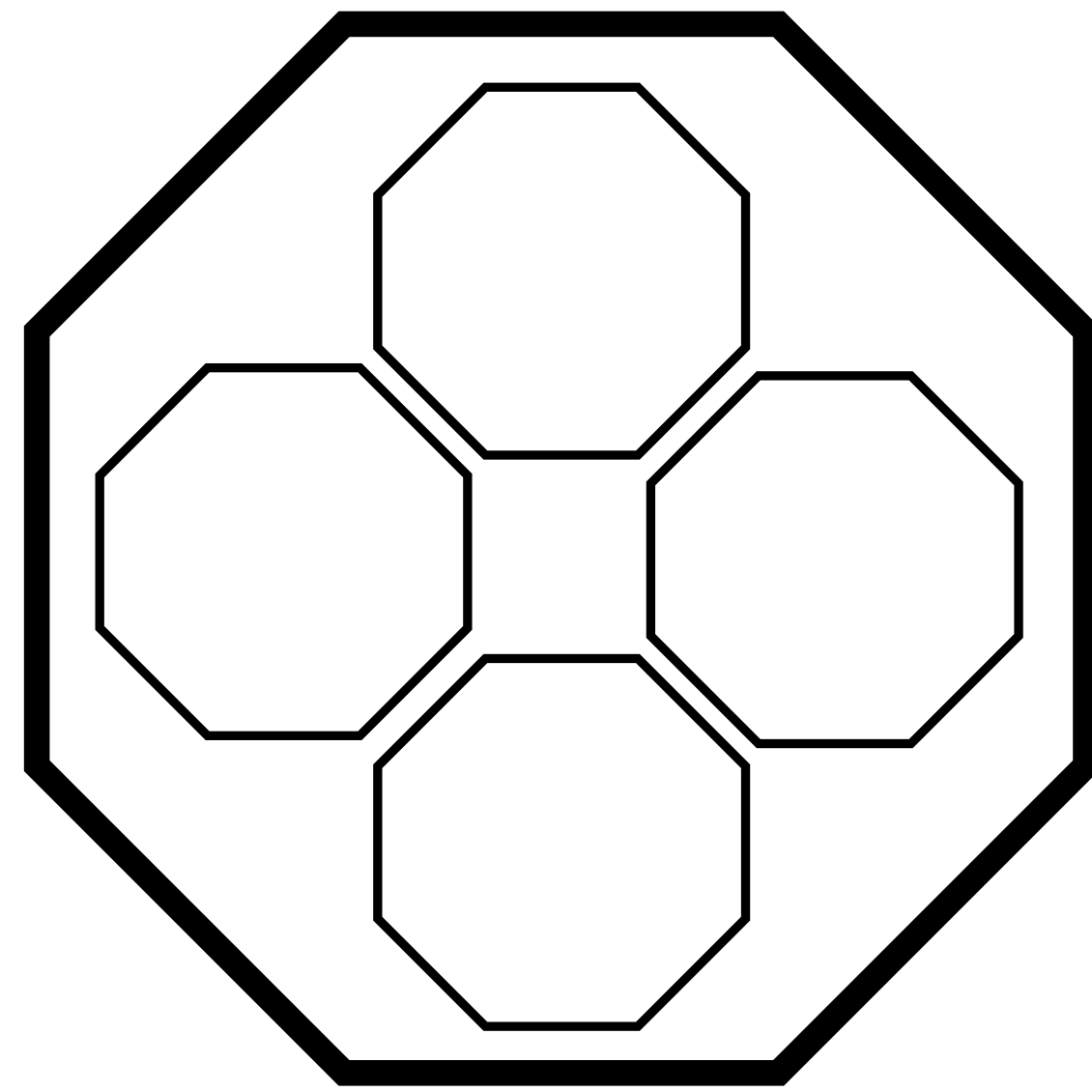
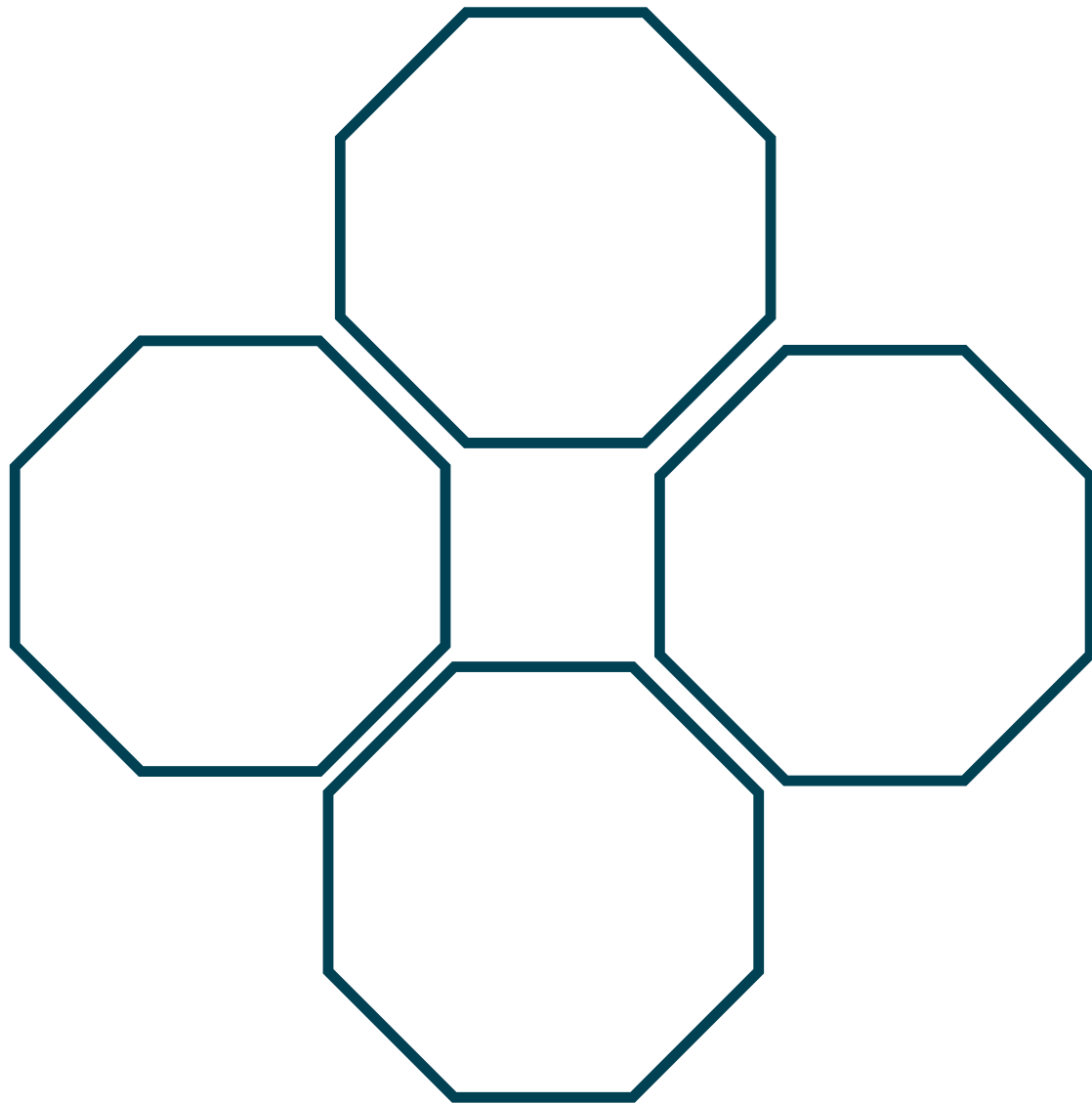
Die ersten Diskussionen in den Dev-Teams starten...



Welche Programmiersprache setzen wir ein?

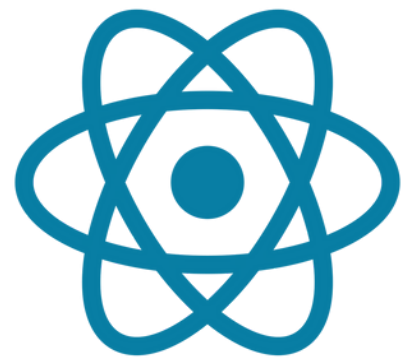


Machen wir jetzt **Microservices**
oder verfolgen wir einen
modulithischen Ansatz?

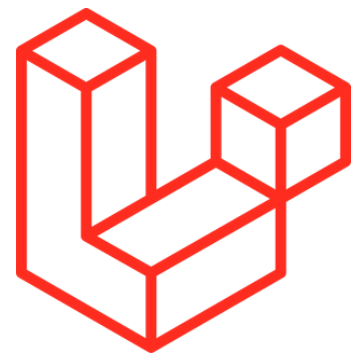


Welches **Frontend Framework** setzen wir ein?

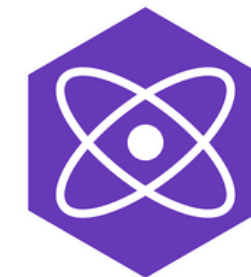
`</> htmx`



 **Alpine.js**



 **quik**



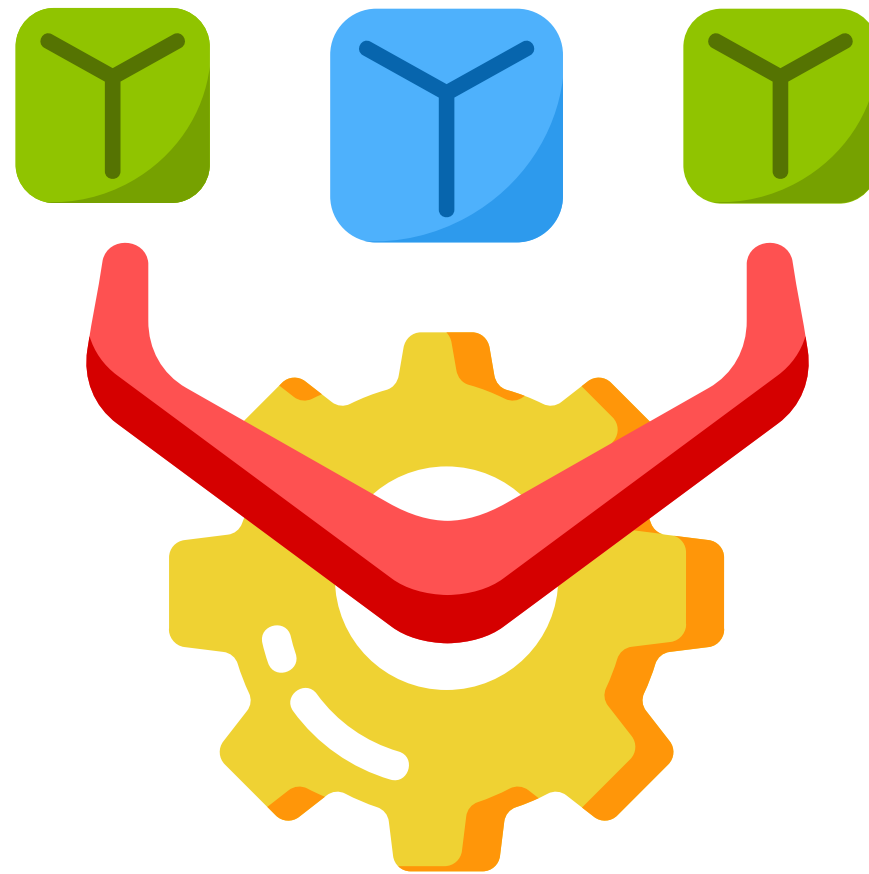
 **SOLID**

ember[®]

Setzen wir ein
Meta-Framework ein?
Wenn ja, welches?



Machen wir **Micro-Frontends?**



Looking for love in all the wrong frameworks



Hype Driven Development

Life on the Bandwagon

Diese Fragestellungen sind
wichtig.

Aber **beim Start** eines
Software Vorhabens vorerst
nicht relevant.



Architecture Inception Canvas

Software System:

Designed by Team:

Workshop Date:

Iteration:

Business Case

Brief description of the business case or economic driver behind the software system.



Functional Overview



The most important functional requirements at a high level

Business Context



Delimits your system under construction as blackbox from all its communication partners. Communication partners are neighbouring systems and users.

Organisational Constraints



Any organisational requirement that restricts software architects' freedom to make decisions.

Quality Goals



Top three quality goals for the architecture which have the highest priority to the main stakeholder

Technical Constraints



Any technical requirement that restricts software architects' freedom to make decisions.

Was soll die Software tun?

Architectural hypothesis



Resulting identified uncertainties and important, expensive, large-scale, or risky architectural decisions, including justifications.

Wie können wir das erreichen?

Challenges & Risks



Identified current known challenges technical risks

Welche technischen Risiken gibt es?

Architecture Inception Canvas

Software System:

Designed by Team:

Workshop Date:

Iteration:

Business Case

Brief description of the business case or economic driver behind the software system.



Functional Overview



The most important functional requirements at a high level

Business Context



Delimits your system under construction as blackbox from all its communication partners. Communication partners are neighbouring systems and users.

Organisational Constraints



Any organisational requirement that restricts software architects' freedom to make decisions.

Quality Goals



Top three quality goals for the architecture which have the highest priority to the main stakeholder

Technical Constraints



Any technical requirement that restricts software architects' freedom to make decisions.

Architectural hypothesis



Resulting architectural hypothesis and important, expensive, large-scale, or risky architectural decisions, including justifications.

Challenges & Risks



Identified current known challenges technical risks

Software System:
Next Gen Commerce System

Designed by Team:
eCommerce Dev Team

Workshop Date:
29.10.2025

Iteration:
1

omic driver behind the software system.

Context

ur system under construction as blackbox from
unication partners. Communication partners
uring systems and users.



Organisational Constraints

Any organisational requirement
that restricts software architects'
freedom to make decisions.

Technical Constraints

Any technical requirement that
restricts software architects'
freedom to make decisions.

Always Start With

WHY?

START

HOW GREAT LEADERS INSPIRE
EVERYONE TO TAKE ACTION

WITH

SIMON SINEK

THE
GLOBAL
BESTSELLER

WHY

Architecture Inception Canvas

Software System:

Designed by Team:

Workshop Date:

Iteration:

Business Case

Brief description of the business case or economic driver behind the software system.



Functional Overview



The most important functional requirements at a high level

Business Context



Delimits your system under construction as blackbox from all its communication partners. Communication partners are neighbouring systems and users.

Organisational Constraints



Any organisational requirement that restricts software architects' freedom to make decisions.

Quality Goals



Top three quality goals for the architecture which have the highest priority to the main stakeholder

Technical Constraints



Any technical requirement that restricts software architects' freedom to make decisions.

Architectural hypothesis



Resulting architectural hypothesis and important, expensive, large-scale, or risky architectural decisions, including justifications.

Challenges & Risks



Identified current known challenges technical risks

Architecture Inception Canvas

Software Initiative:

Business Case

Brief description of the business case or economic driver behind the software system.

Functional Overview



The most important functional requirements at a high level

Business Context

Separate your system under construction as a black box from its communication partners. Communication partners are other systems and users.

Quality Goals



The quality goals of the system are defined by the business case and the functional requirements.

O'REILLY®

ÜBERSETZUNG DER
3. AUFLAGE

ASH MAURYA

RUNNING LEAN

Geschäftsmodelle
schnell und iterativ entwickeln



THE LEAN SERIES
HERAUSGEBER: ERIC RIES

Übersetzung von Jens Olaf Koch

Lean Canvas

Designed For:

Designed by:

Date:

Version:

Problem

What are the main problems the business needs to solve?

Describe the top 1–3 biggest problems your customers face.



Solution

Describe a solution for each problem



Value Propositions

What value do you deliver to your customers?

Provide a simple, clear message that explains why your solution is unique and worth paying attention to.



Unfair Advantage

Something that makes it difficult for others to copy the solution.



Customer Segments

List the target and user groups. For whom do we create value? Who are the most important customers?



Alternatives

How have these problems been solved so far?



Metrics

Select the key metrics you'll use to measure success



Channels

Through which channels do your customer segments want to be reached?



Small concept

X for Y analogy. Is there a simple analogy?

(e.g., YouTube = Flickr for video)



Cost Structure

List the fixed and variable costs.



Revenue Stream

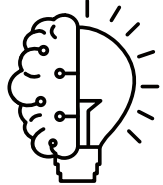
List the revenue streams.



Lean Canvas

Designed For:

Designed by:

Problem  <i>What are the main problems the business needs to solve?</i> <i>Describe the top 1–3 biggest problems your customers face.</i>	Solution  <i>Describe a solution for each problem.</i>	Value Propositions  <i>What value do you deliver to your customers?</i> <i>Provide a simple, clear message that explains why your solution is unique and worth paying attention to.</i>	Unfair Advantage  <i>Something that makes it difficult for others to copy the solution.</i>
Alternatives  <i>How have these problems been solved so far?</i>	Metrics  <i>Select the key metrics you'll use to measure success</i>	Small concept  <i>X for Y analogy. Is there a simple analogy?</i> <i>(e.g., YouTube = Flickr for video)</i>	Channels  <i>Through which channels do your customer segments want to be reached?</i>
Cost Structure  <i>List the fixed and variable costs.</i>			Revenue Stream <i>List the revenue streams.</i> <i>Drawn by Patrick Roos, Source of Information: Running Lean by Ash Maurya (2013)</i>

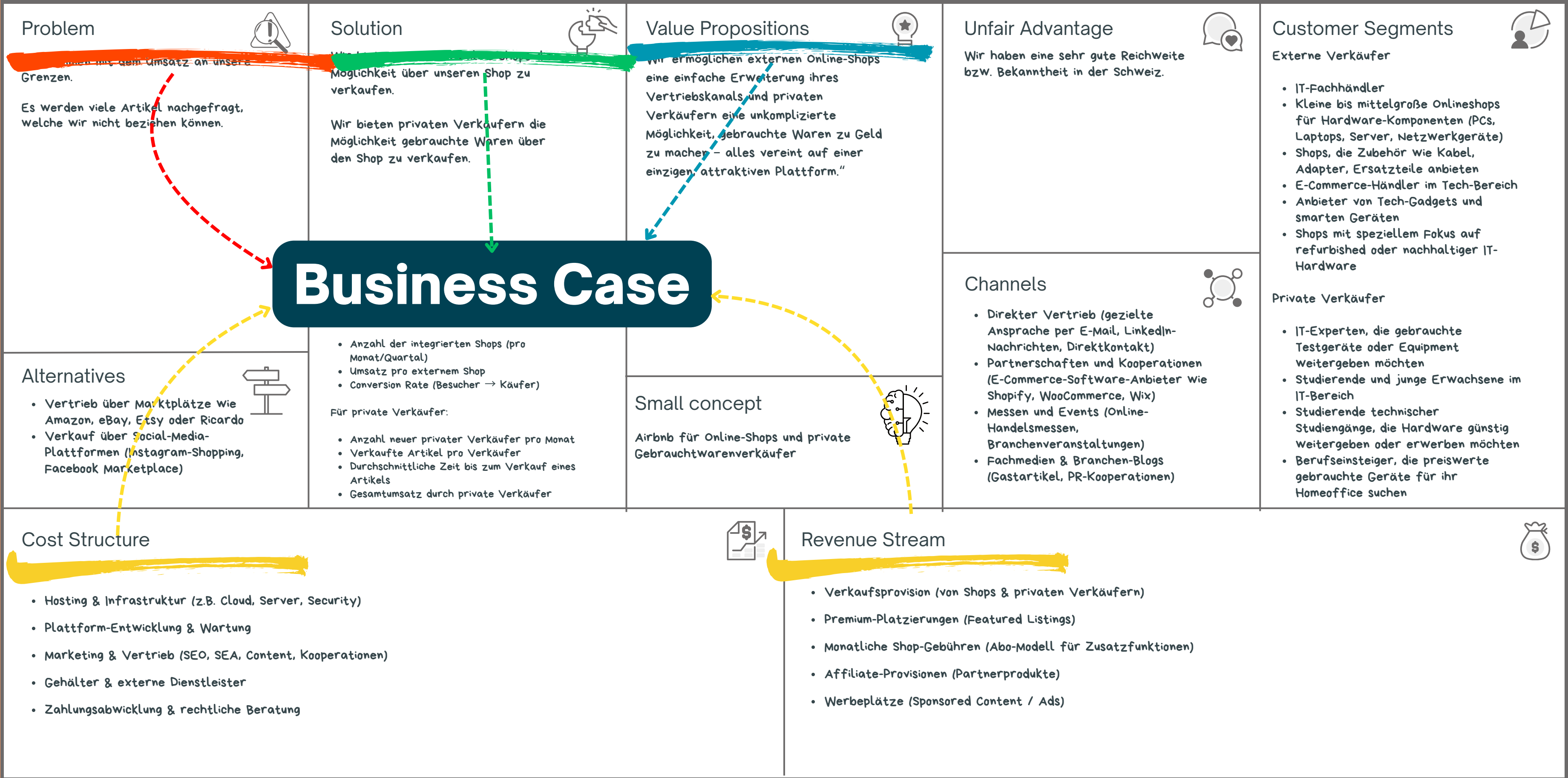
Lean Canvas

Designed For:
eCommerce Company

Designed by:
Product Management

Date:
29.10.2025

Version:
1



Architecture Inception Canvas

Software Initiative:

Designed by:

Business Case

Erschliessung neuer Märkte und Erweiterung des eigenen Produktportfolios in dem man externe Shops & private Verkäufer auf dem eigenen, reichweite starken, Shop zulässt.

Functional Overview

The most important functional requirements at a high level



Business Context

Separate your system under construction as a black box from all its communication partners. Communication partners are neighbouring external systems and users.

Quality Goals

The three most important quality goals for the architecture, which have the highest priority for the most important stakeholder



Architecture Inception Canvas

Software System:

Designed by Team:

Workshop Date:

Iteration:

Business Case

Brief description of the business case or economic driver behind the software system.



Functional Overview



The most important functional requirements at a high level

Business Context



Delimits your system under construction as blackbox from all its communication partners. Communication partners are neighbouring systems and users.

Organisational Constraints



Any organisational requirement that restricts software architects' freedom to make decisions.

Quality Goals



Top three quality goals for the architecture which have the highest priority to the main stakeholder

Technical Constraints



Any technical requirement that restricts software architects' freedom to make decisions.

Architectural hypothesis



Resulting architectural hypothesis and important, expensive, large-scale, or risky architectural decisions, including justifications.

Challenges & Risks



Identified current known challenges technical risks

Architecture Inception Canvas

Software Initi

Business Case *Erschliessung neuer Märkte in dem man externe Verkäufer & private Ve*

Functional Overview

The most important functional requirements at a high level



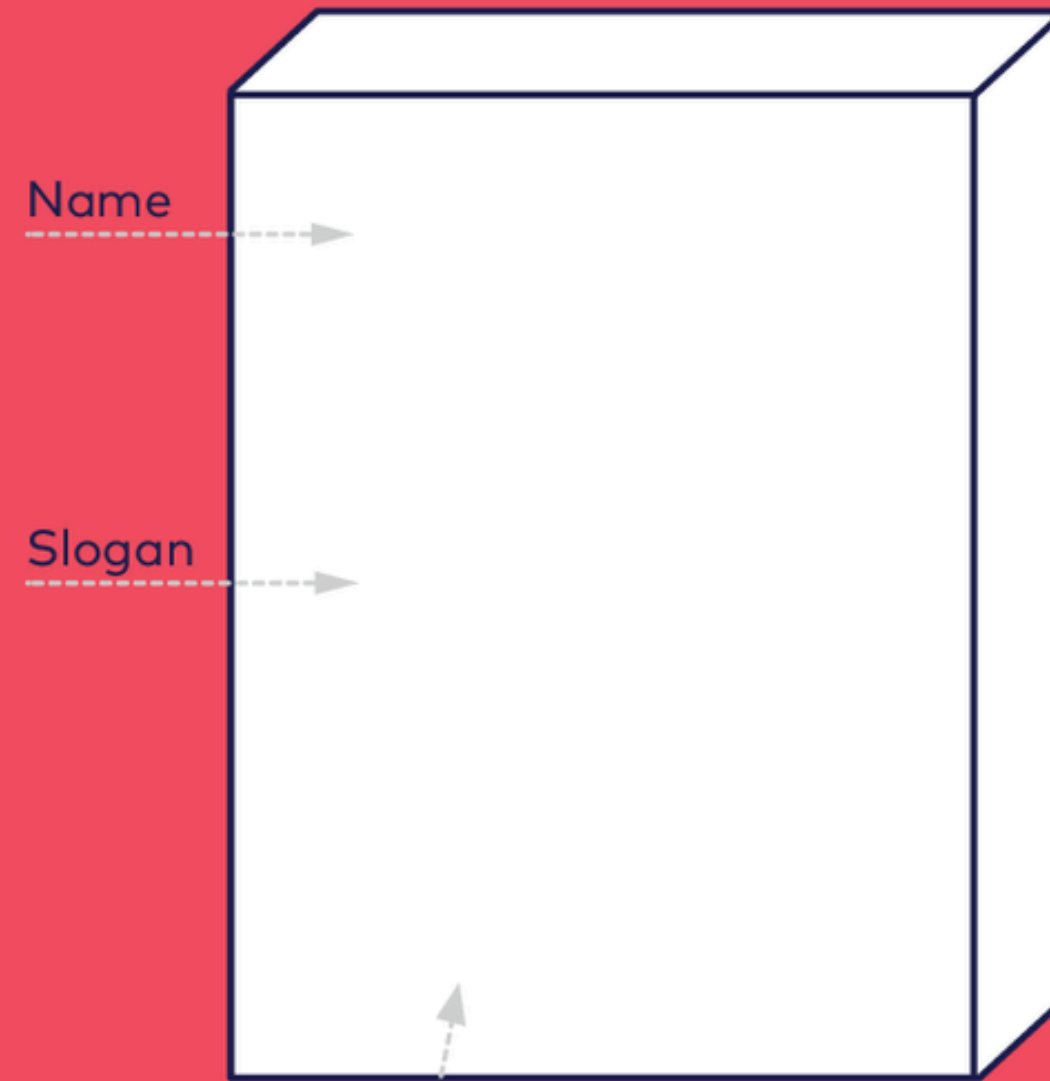
Business Context

*Separate your system under constru
communication partners. Communic
systems and users.*



Deine Produktbox

Designe die Verkaufspackung für
Deine Software!



Alleinstellungsmerkmale

cards42.org/#produktbox

INNOQ

cards42.org

Deine Produktbox

Designe die Verkaufspackung für
Deine Software!



Architecture Inception C



Business Case *Erschliessung neue Märkte*

Functional Overview



- Import, Verwaltung und Verkauf von Produkten von externen Onlineshops
- Einstellen und Verkauf von gebrauchten Waren durch Privat-Personen

Quality Goals



Architecture Inception Canvas

Software System:

Designed by Team:

Workshop Date:

Iteration:

Business Case

Brief description of the business case or economic driver behind the software system.



Functional Overview



The most important functional requirements at a high level

Business Context

Delimits your system under construction as blackbox from all its communication partners. Communication partners are neighbouring systems and users.



Organisational Constraints



Any organisational requirement that restricts software architects' freedom to make decisions.

Quality Goals



Top three quality goals for the architecture which have the highest priority to the main stakeholder

Technical Constraints



Any technical requirement that restricts software architects' freedom to make decisions.

Architectural hypothesis

Resulting architectural hypothesis and important, expensive, large-scale, or risky architectural decisions, including justifications.



Challenges & Risks

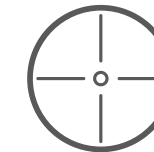
Identified current known challenges technical risks



ional
vel



Business Context



Separate your system under construction as a black box from all its communication partners. Communication partners are neighbouring external systems and users.

Organisational C

*Any organisational
that limits the softw
freedom of decision*

quality goals for
ve the highest
tant stakeholder



Technical Const

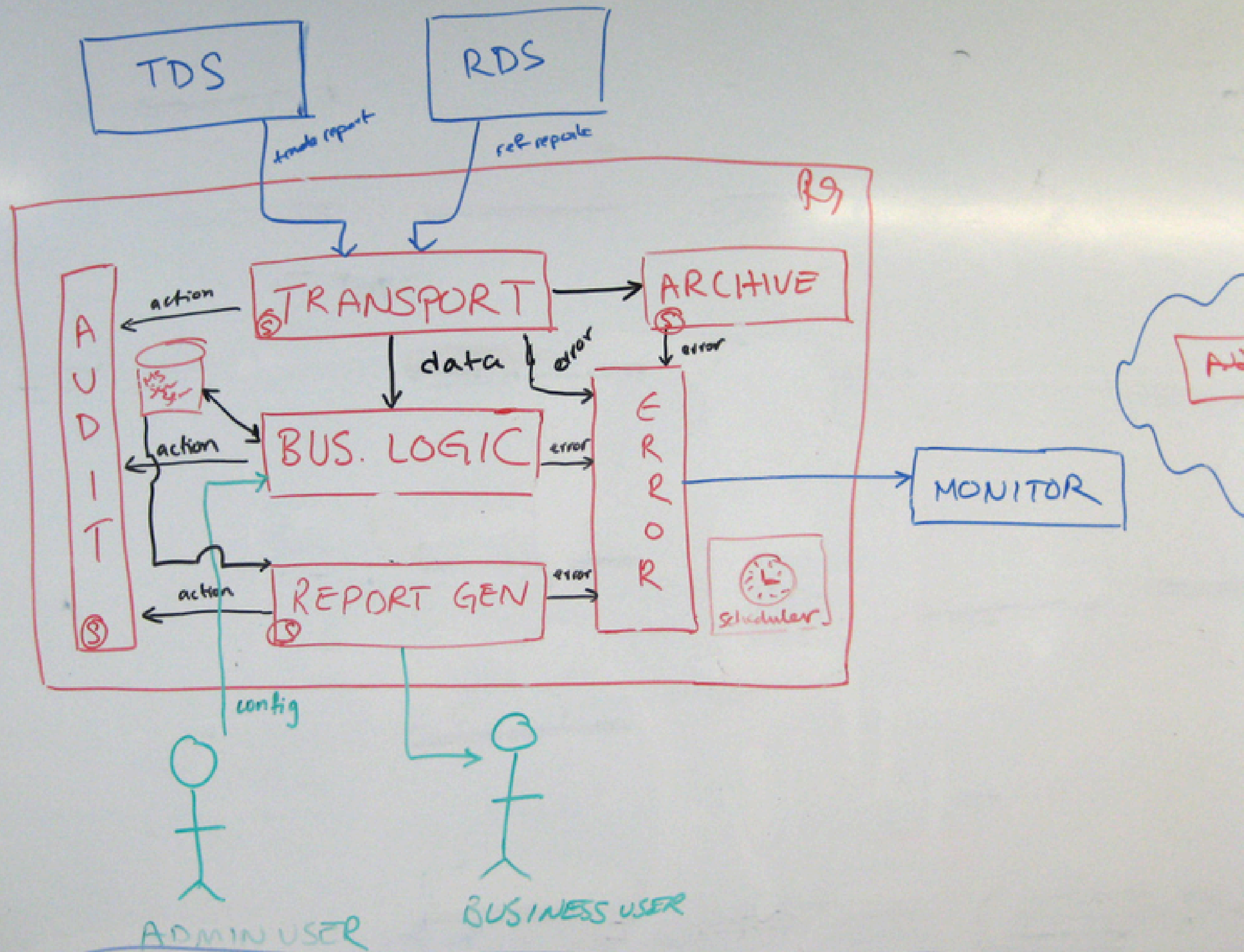
*Any technical requi
restricts the softwa
freedom of decision*

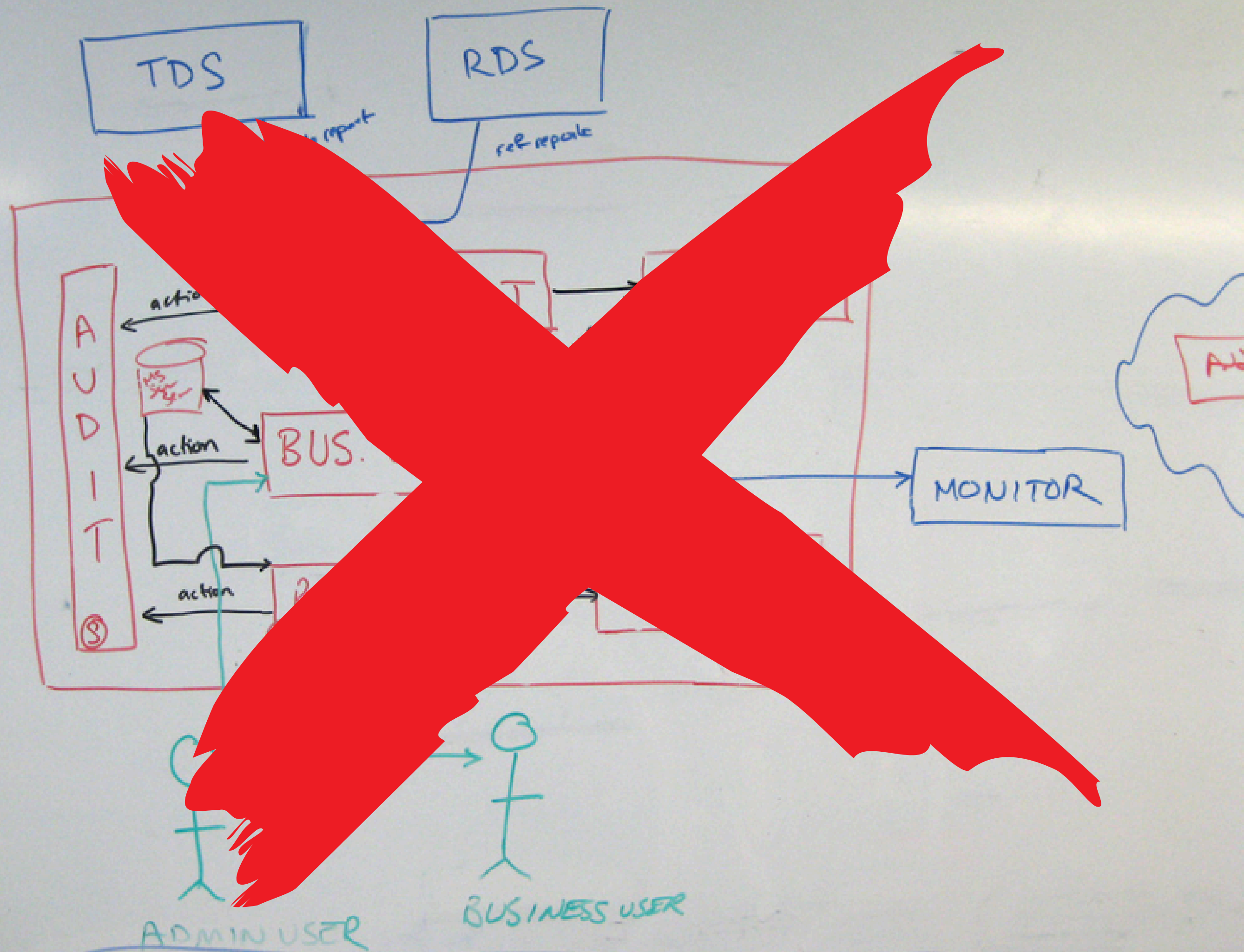
ses



Technical Challenges & Risks

Identified current known challenges technical risks





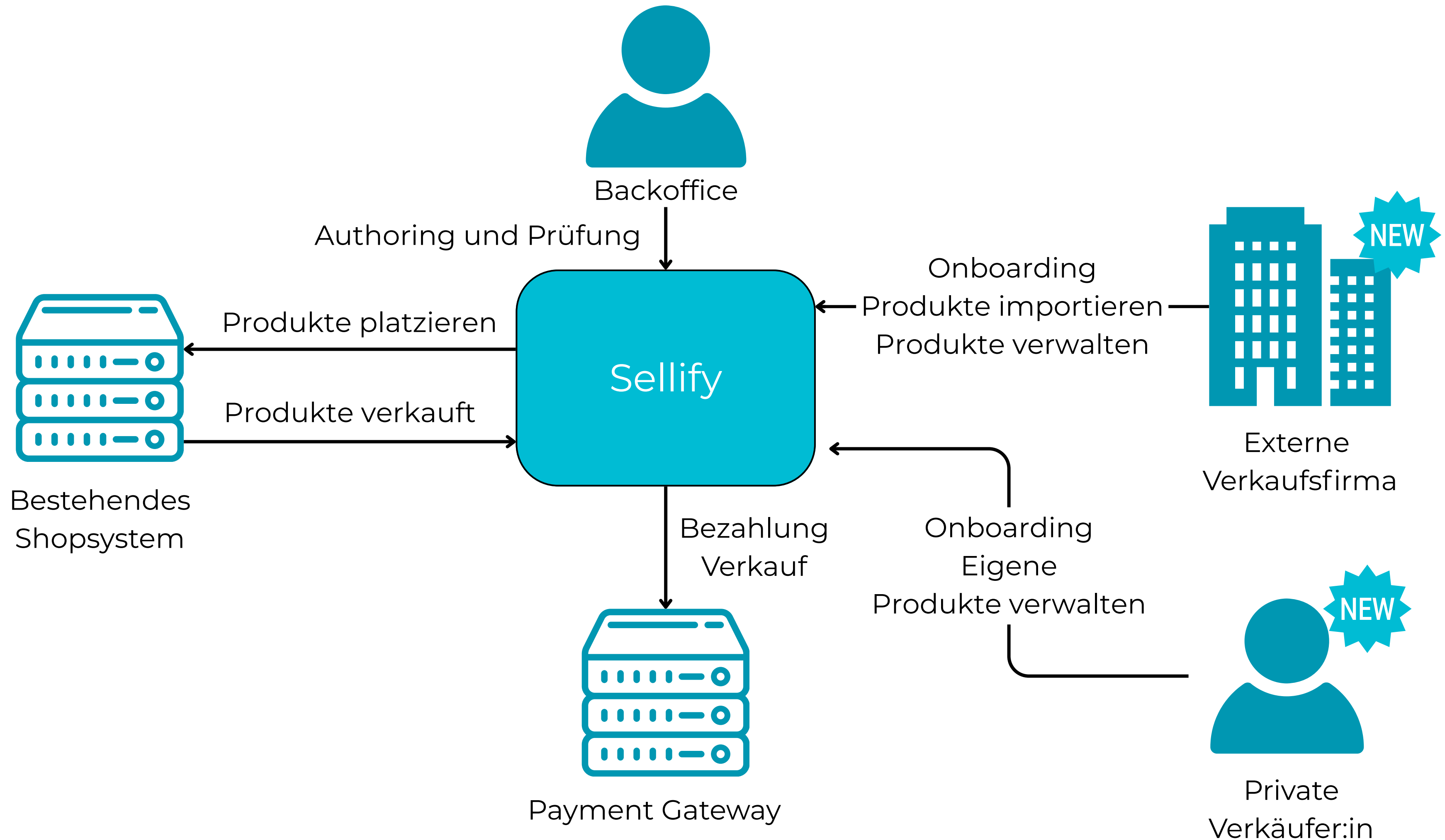
Business Context

Scope: Einzelnes Software System

Ziel: Jede Person innerhalb und ausserhalb des Entwicklungsteams (auch ohne technischen Background) versteht die Einbettung des neuen Software Systems in die Systemlandschaft.

Technik um...

- ...Klarheit über die Systemgrenzen zu erhalten
- ...Verständnis und Orientierung schaffen
- ...Identifikation erster wichtiger Schnittstellen zu ermöglichen
- ...Strategischer Entscheidungen zu unterstützen
- ...fachliche und technische Risiken zu erkennen



C4 Model

Visualisations

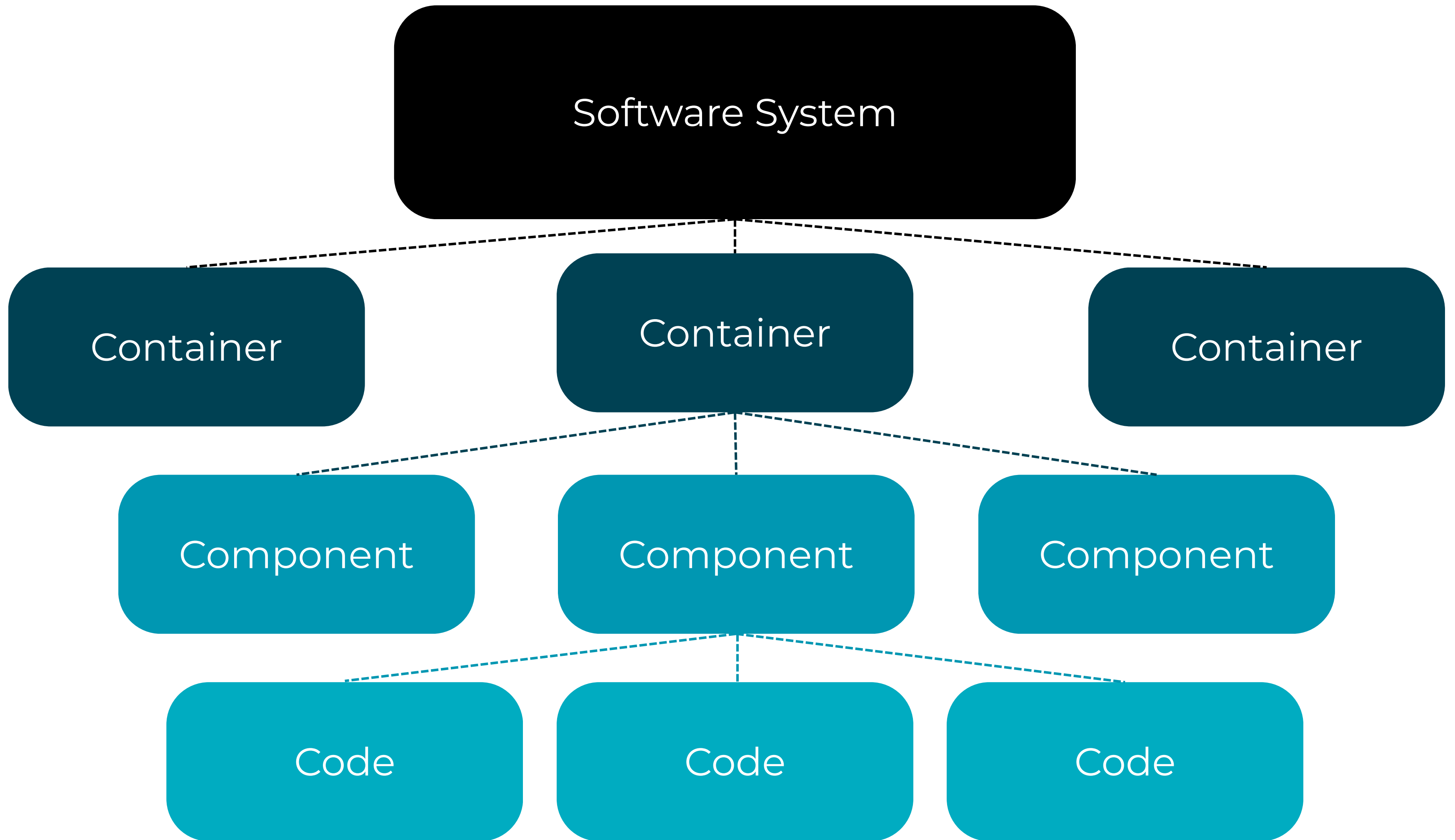
=

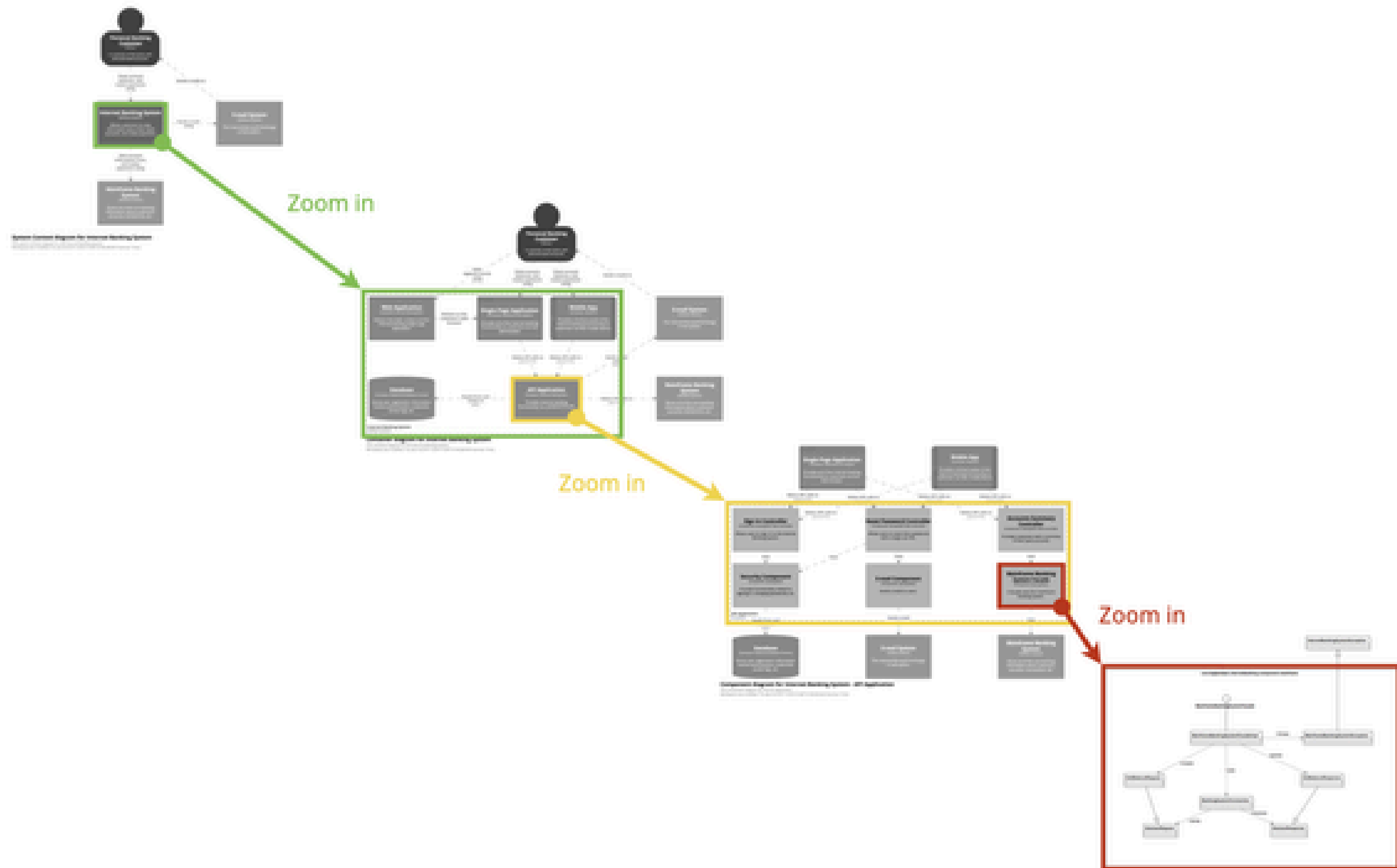
C4 Model

Abstractions + Visualisations

=

C4 Model



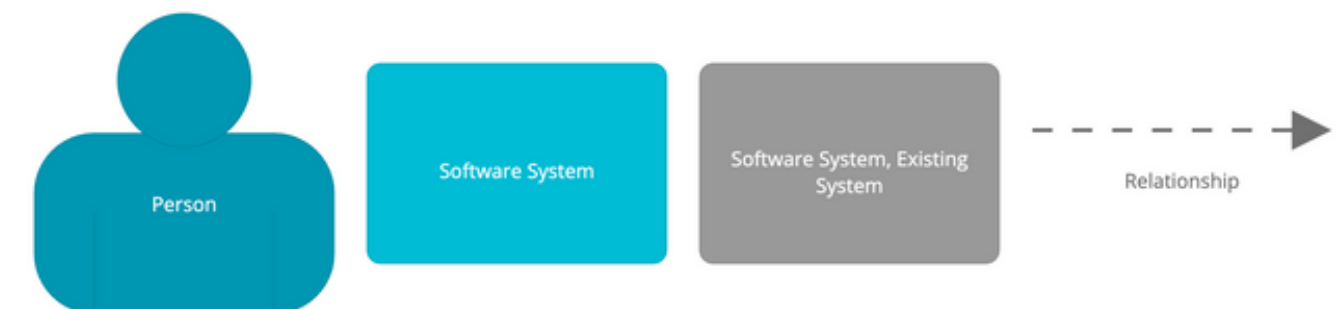
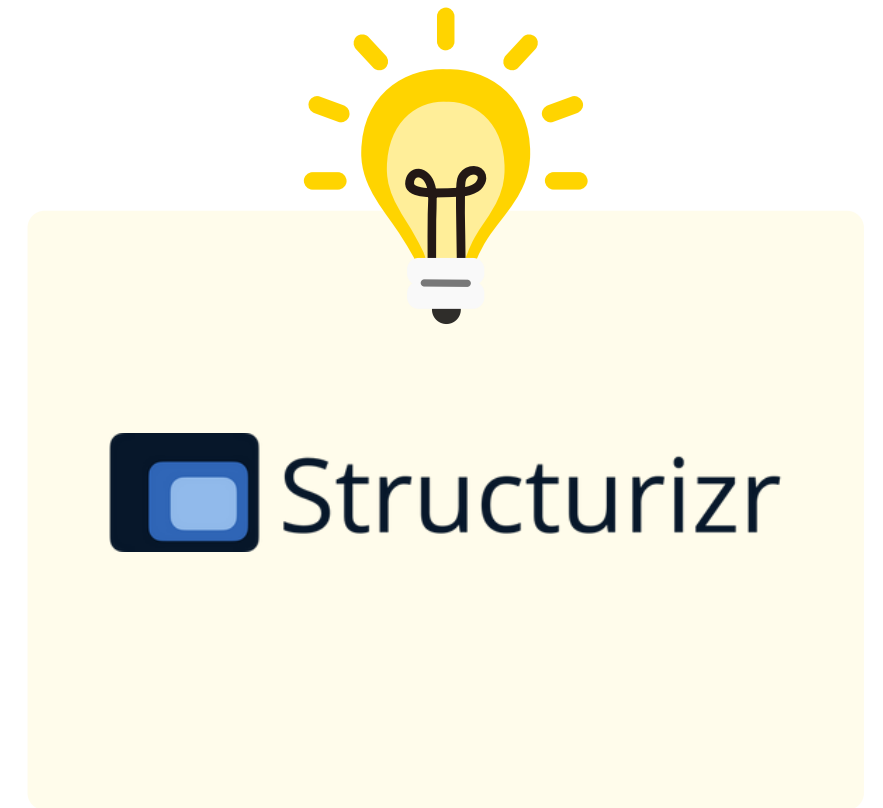
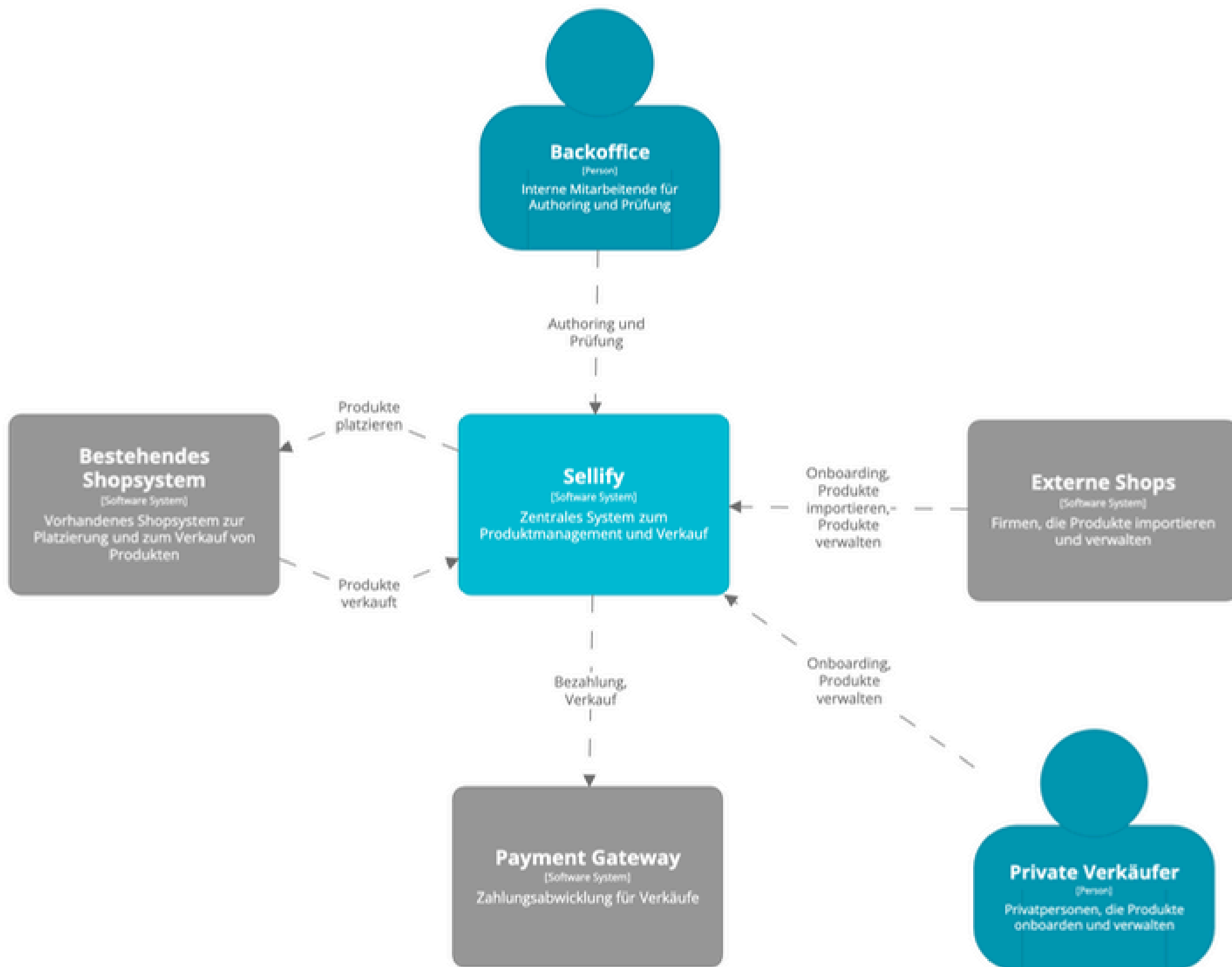


Level 1
Context

Level 2
Containers

Level 3
Components

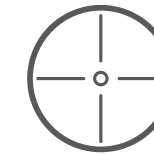
Level 4
Code



Brief description of the business case or economic driver behind the software system.



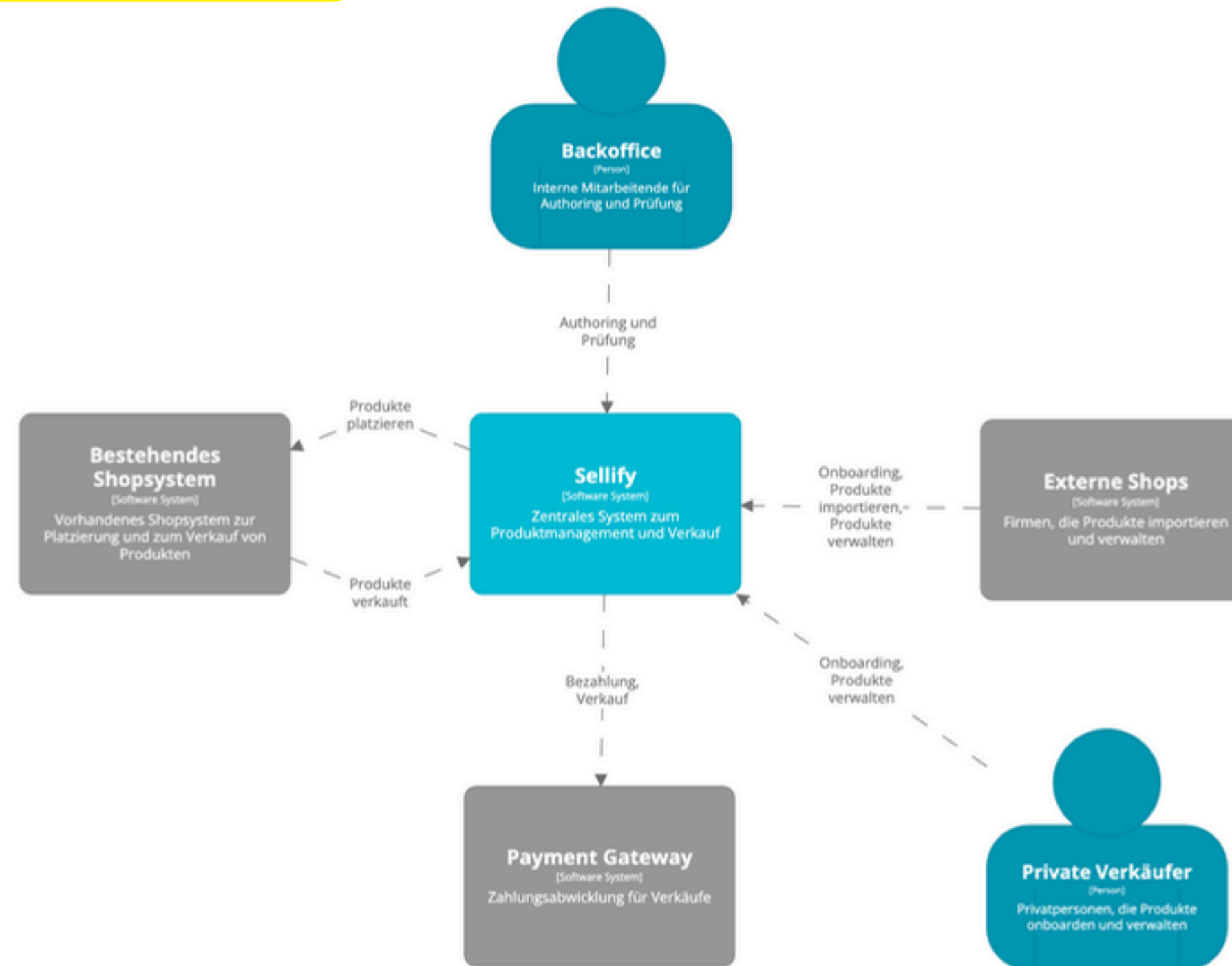
Business Context



Organisational Constraints

Any organisational constraints that limits the software system's freedom of decision

Organisational
level



quality goals for
the highest
stakeholder

Technical Constraints

Any technical requirements that restricts the software system's freedom of decision

Technical Challenges & Risks



Technical Challenges & Risks

Identified current known challenges technical risks

Architecture Inception Canvas

Software System:

Designed by Team:

Workshop Date:

Iteration:

Business Case

Brief description of the business case or economic driver behind the software system.



Functional Overview



The most important functional requirements at a high level

Business Context



Delimits your system under construction as blackbox from all its communication partners. Communication partners are neighbouring systems and users.

Organisational Constraints



Any organisational requirement that restricts software architects' freedom to make decisions.

Quality Goals



Top three quality goals for the architecture which have the highest priority to the main stakeholder

Technical Constraints



Any technical requirement that restricts software architects' freedom to make decisions.

Architectural hypothesis



Resulting architectural hypothesis and important, expensive, large-scale, or risky architectural decisions, including justifications.

Challenges & Risks



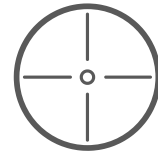
Identified current known challenges technical risks

conomic driver behind the software system.



Context

our system under construction as a black box from all its
tion partners. Communication partners are neighbouring external
d users.



Organisational Constraints



Any organisational requirement
that limits the software architects'
freedom of decision.

Technical Constraints



Any technical requirement that
restricts the software architects'
freedom of decision.



Technical Challenges & Risks

Identified current/known challenges/technical risks



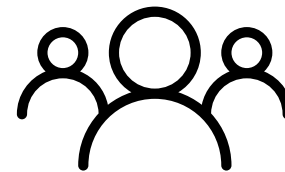
«Die Entwicklungskosten für das MVP dürfen CHF 300'000 nicht überschreiten.»

«Nutzung einer bestimmten Cloud-Plattform (z. B. Azure) vorgeschrieben.»

«Es dürfen nur Technologien eingesetzt werden, für welche intern qualifizierte Mitarbeiter vorhanden sind.»

«Erste produktive Version (MVP) muss innerhalb von 3 Monaten veröffentlicht werden.»

Software Initiative



«Lizenzen für Open-Source-Software müssen zwingend intern geprüft und genehmigt werden.»

«Es muss zwingend die bestehende CI/CD-Infrastruktur genutzt werden.»

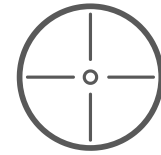
«Wichtige Architekturentscheide müssen durch das Architekturboard genehmigt werden.»

economic driver behind the software system.



Context

Our system under construction as a black box from all its communication partners. Communication partners are neighbouring external systems and users.



Organisational Constraints



- Architektur-Entscheidungen müssen durch das Architekturboard genehmigt werden.
- Erster fachlicher Durchstich sollte nach drei Monaten für erste ausgewählte Testkunden sichtbar sein
- Die Entwicklungskosten für das MVP dürfen nicht 300k überschreiten.

Technical Constraints



- Hosting auf unserer Azure Plattform
- Bestehendes React UI Komponenten-Framework soll wiederverwendet werden, um ein einheitliches Branding zu gewährleisten
- Es muss die bestehende Build- und Deployment Umgebung verwendet werden.



Technical Challenges & Risks

Identified current known challenges technical risks



Architecture Inception Canvas

Software System:

Designed by Team:

Workshop Date:

Iteration:

Business Case

Brief description of the business case or economic driver behind the software system.



Functional Overview



The most important functional requirements at a high level

Business Context



Delimits your system under construction as blackbox from all its communication partners. Communication partners are neighbouring systems and users.

Organisational Constraints



Any organisational requirement that restricts software architects' freedom to make decisions.

Quality Goals



Top three quality goals for the architecture which have the highest priority to the main stakeholder

Technical Constraints



Any technical requirement that restricts software architects' freedom to make decisions.

Architectural hypothesis



Resulting architectural hypothesis and important, expensive, large-scale, or risky architectural decisions, including justifications.

Challenges & Risks



Identified current known challenges technical risks

requirements at a high level

systems and users.

Quality Goals

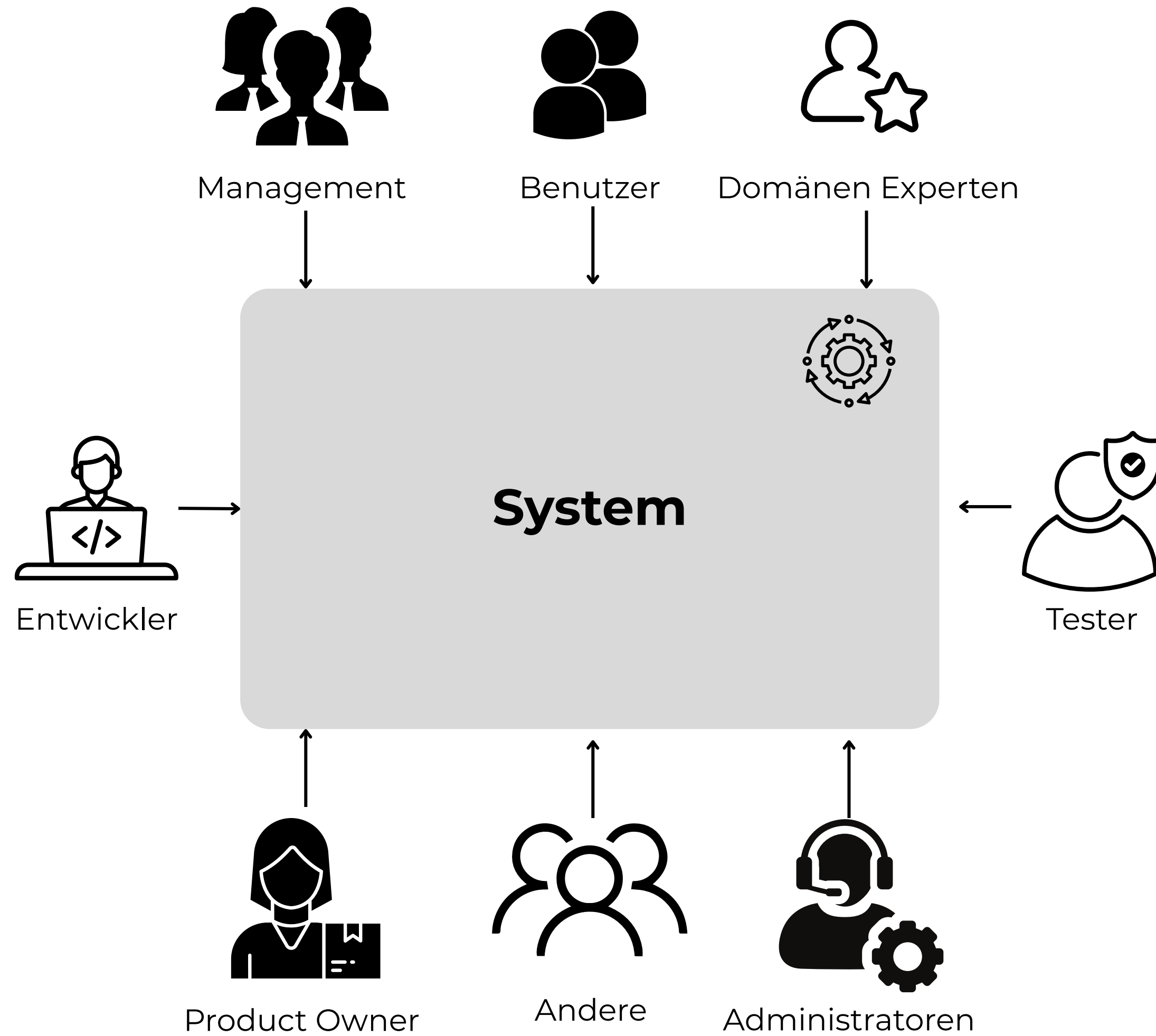


The three most important quality goals for the architecture, which have the highest priority for the most important stakeholder

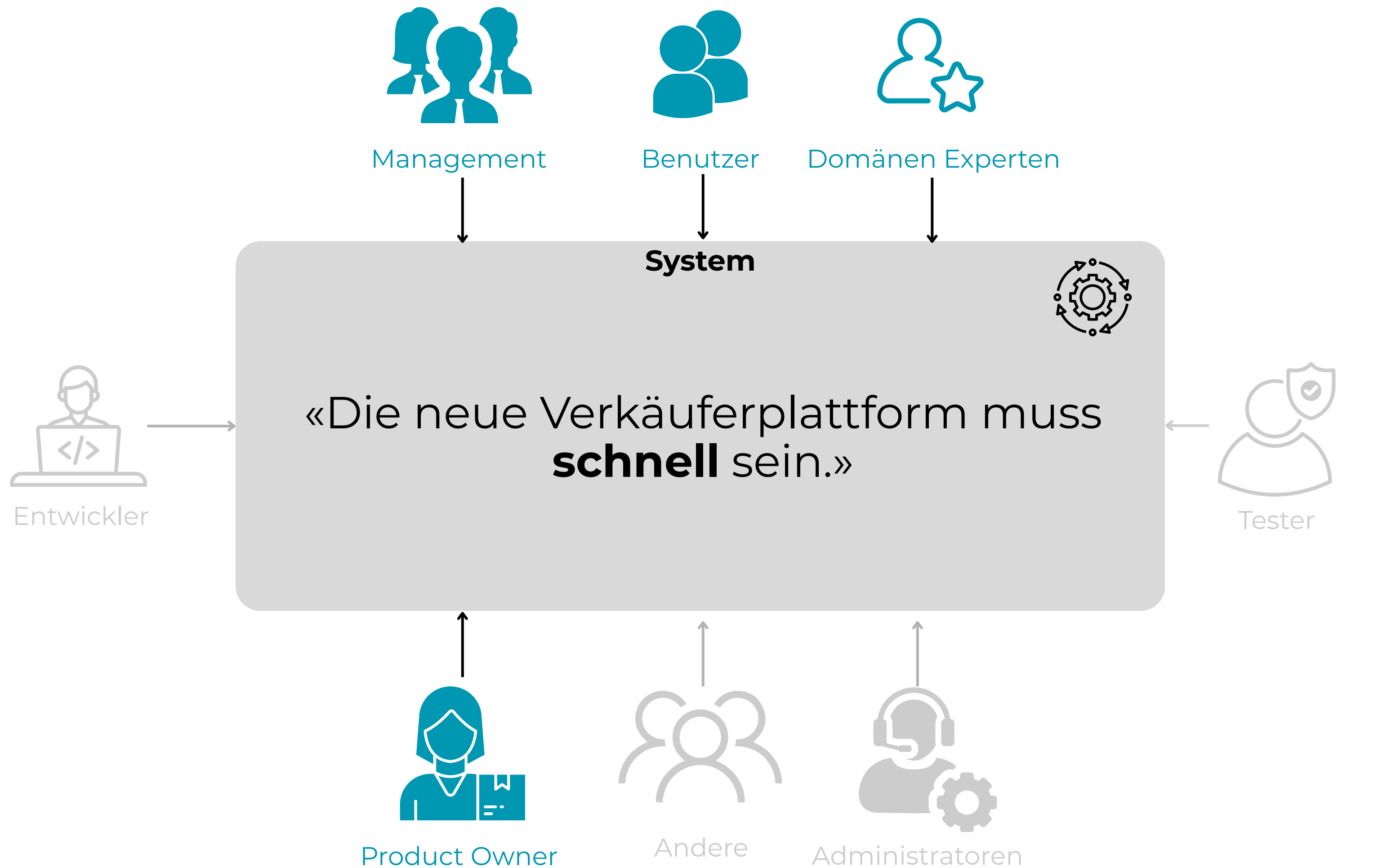
Architectural hypotheses



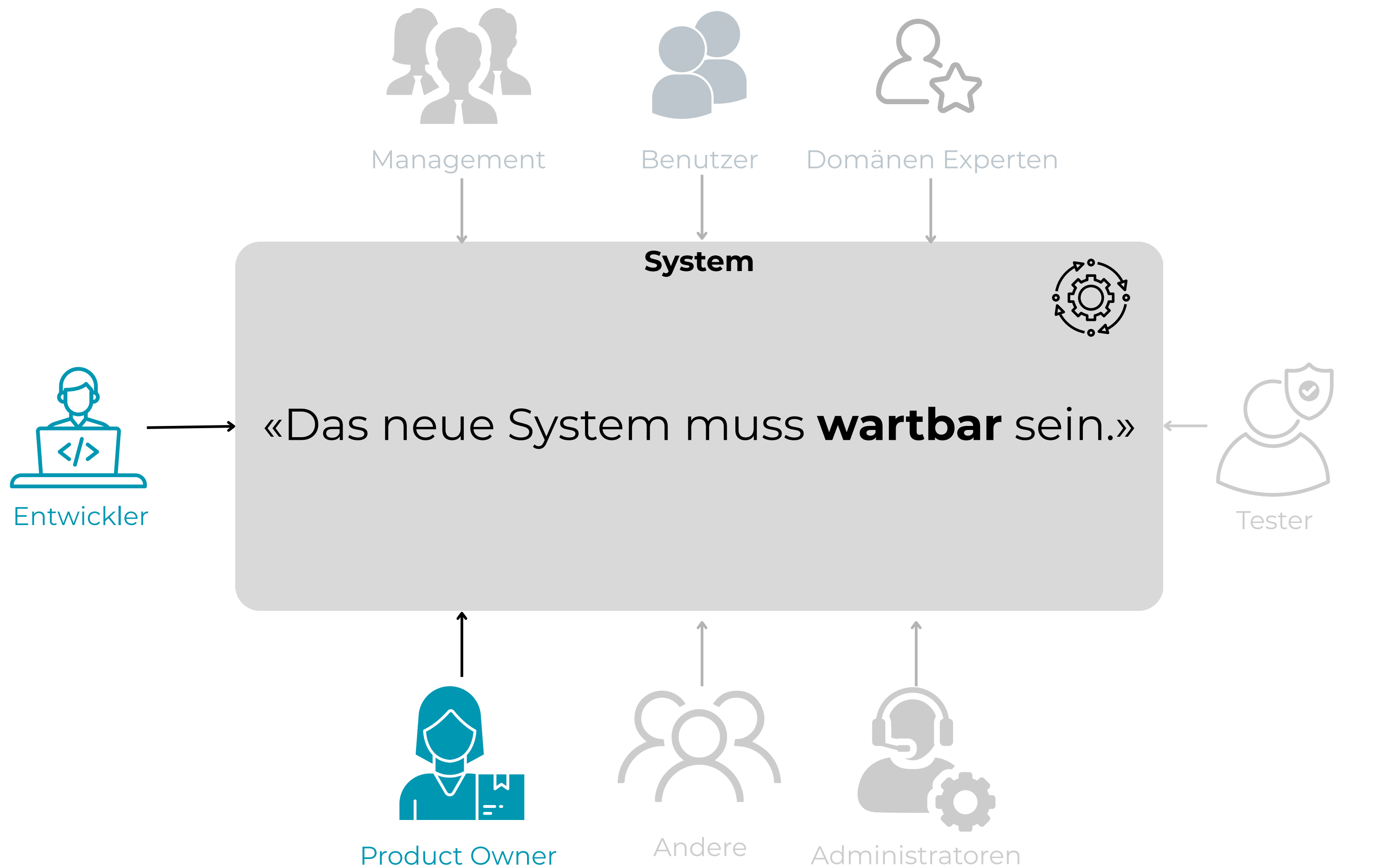
Resulting architectural hypotheses and important



→ Erwartungen



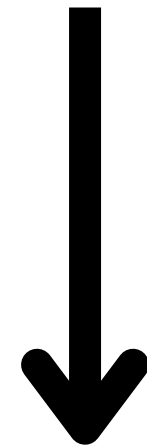
Was heisst «schnell»?



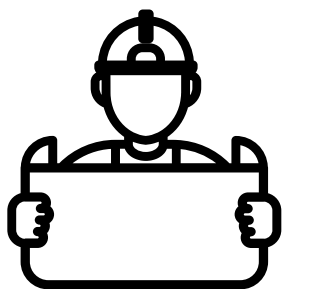
→ Erwartungen

Was heisst «wartbar»?

Implizite Erwartungen



**Software Architektur
Arbeit!**



Explizite Qualitätsanforderungen

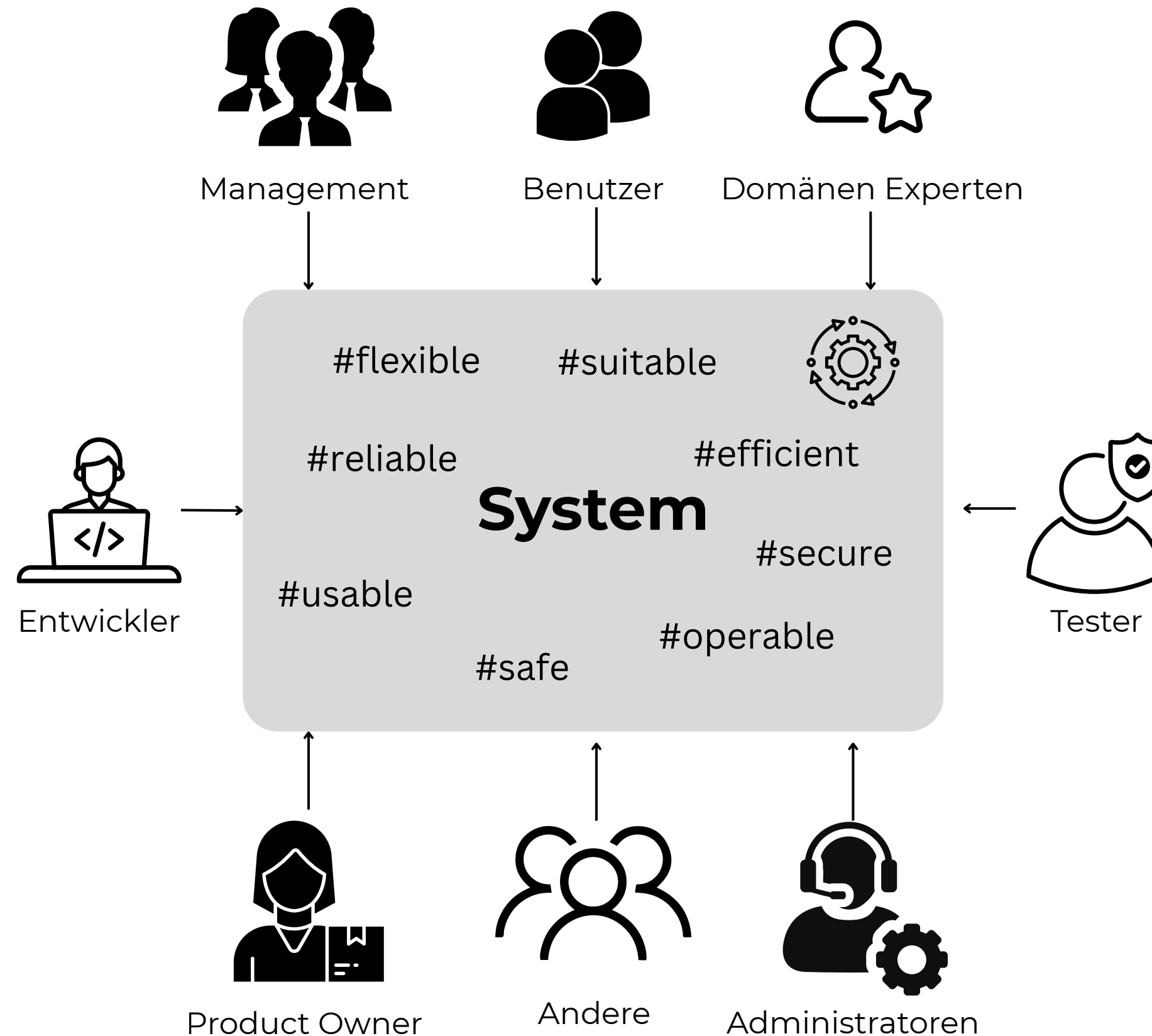
Q42 Quality Model



arc42 Quality Model

161 quality attributes, explained.

104 examples of quality requirements.



Quality Goals



1. Kompatibilität (Compatibility):

Sellify-Funktionalitäten und -Lasten dürfen die Performance, Verfügbarkeit und Benutzbarkeit des eigenen Onlineshops für eigene Produkte nicht beeinträchtigen.

2. Benutzbarkeit (Usability):

Verkäufer:innen können ihre Angebote ohne Einarbeitungszeit erstellen.

3. Flexibilität (Flexibility):

Etablierte Online-Shop Systeme können konfigurativ ihre Produkte nach Sellify synchronisieren.

Architectural hypotheses



Architecture Inception Canvas

Software System:

Designed by Team:

Workshop Date:

Iteration:

Business Case

Brief description of the business case or economic driver behind the software system.



Functional Overview



The most important functional requirements at a high level

Business Context



Delimits your system under construction as blackbox from all its communication partners. Communication partners are neighbouring systems and users.

Organisational Constraints



Any organisational requirement that restricts software architects' freedom to make decisions.

Quality Goals



Top three quality goals for the architecture which have the highest priority to the main stakeholder

Technical Constraints



Any technical requirement that restricts software architects' freedom to make decisions.

Architectural hypothesis

Resulting architectural hypothesis and important, expensive, large-scale, or risky architectural decisions, including justifications.



Challenges & Risks

Identified current known challenges technical risks



Quality Goals



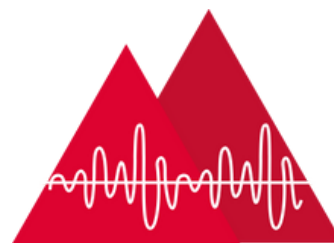
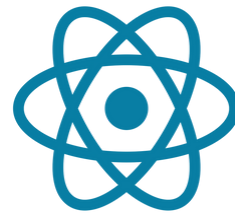
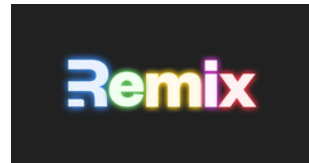
The three most important quality goals for the architecture, which have the highest priority for the most important stakeholder

Architectural hypotheses

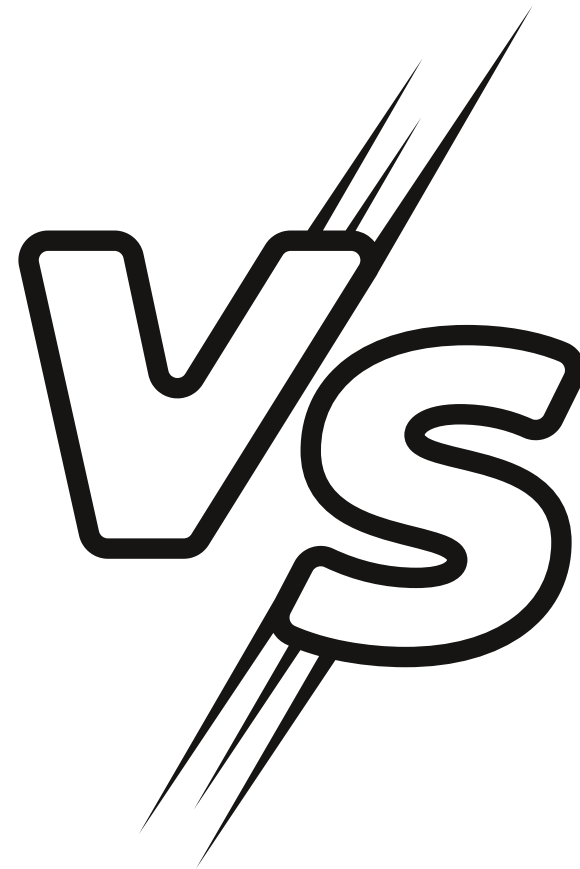


Resulting architectural hypotheses and important, expensive, large-scale or risky architectural decisions, including justifications.





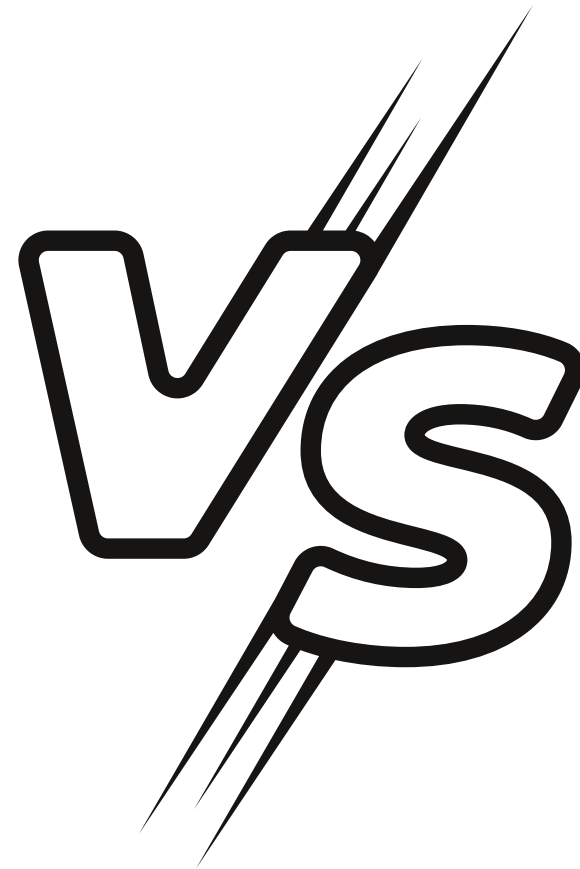
Makro



Mikro

Makro

Betreffen das „große Bild“ - die Struktur, Abgrenzung und Integration ganzer Systeme.

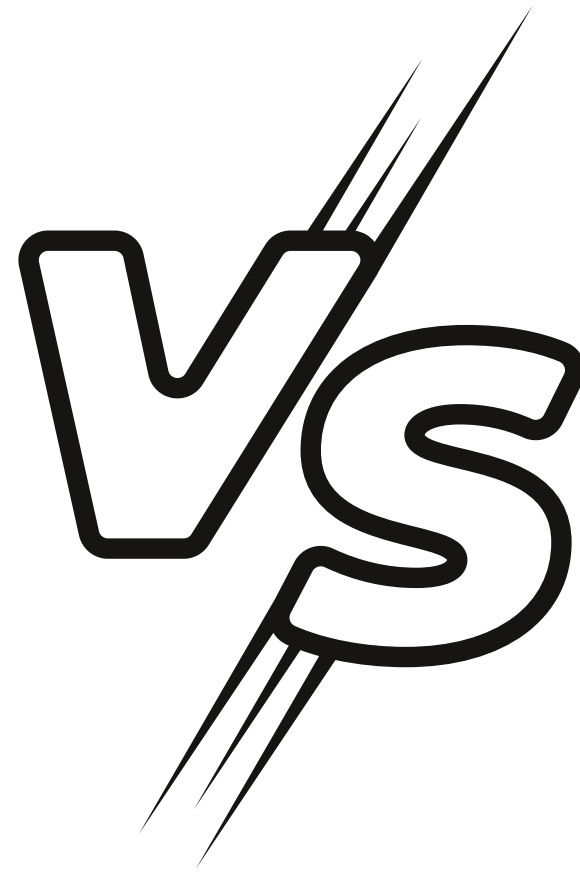


Detaillierte, komponenten-interne Entscheidungen zur konkreten technischen Umsetzung.

Mikro

Makro

Welche
Integrationsstrategie
(z.B. REST-API,
GraphQL, Events)
passt zu den
Anforderungen des
bestehenden
Systems?



Welche State
Management Library
wird im Admin-Teil
des Web-Frontend
genutzt?

Mikro

«Treffe **Makro-Entscheidungen** erst dann, wenn sie tatsächlich getroffen werden müssen - aber nicht später. »



Last Responsible Moment

Letzter vernünftiger Moment



Hoch



Risiko

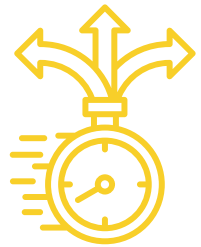


Niedrig

Letzter vernünftiger Moment



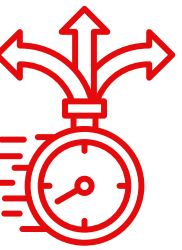
Letzter vernünftiger Moment



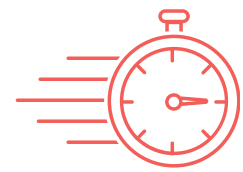
Niedrig

Dringlichkeit

Hoch



Letzter vernünftiger Moment



(externe) Abhängigkeiten



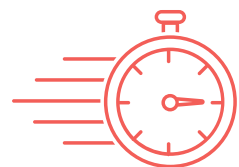
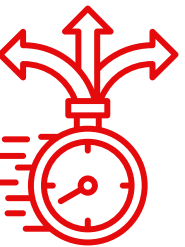
Negative Seiteneffekte
durch's Offenlassen



Niedrig

Dringlichkeit

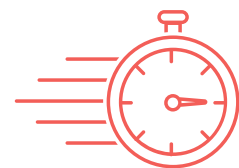
Hoch



Gefahr von de-facto
Entscheidungen



Klarheit für Entscheidungen
wichtig

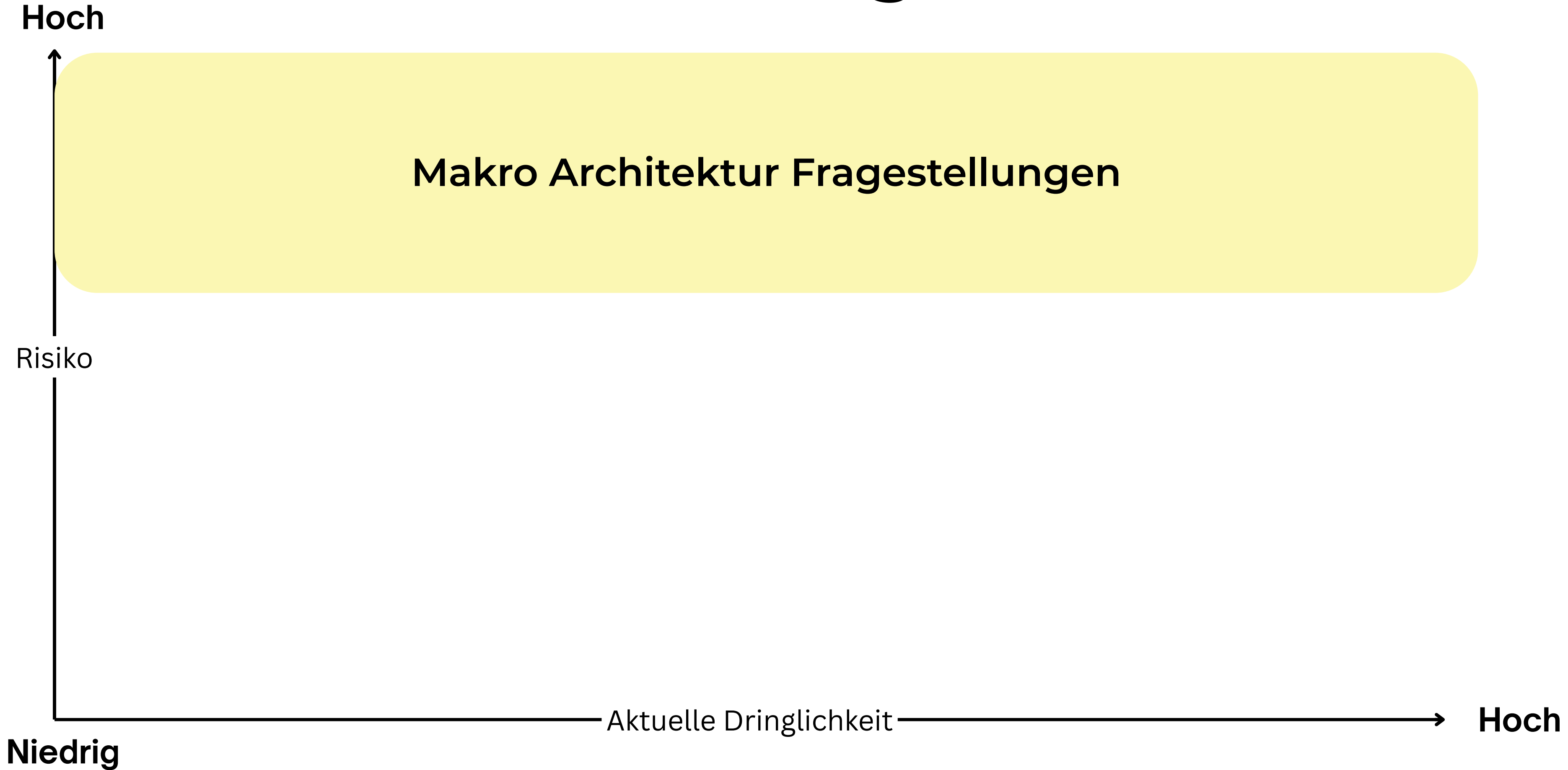


Termindruck

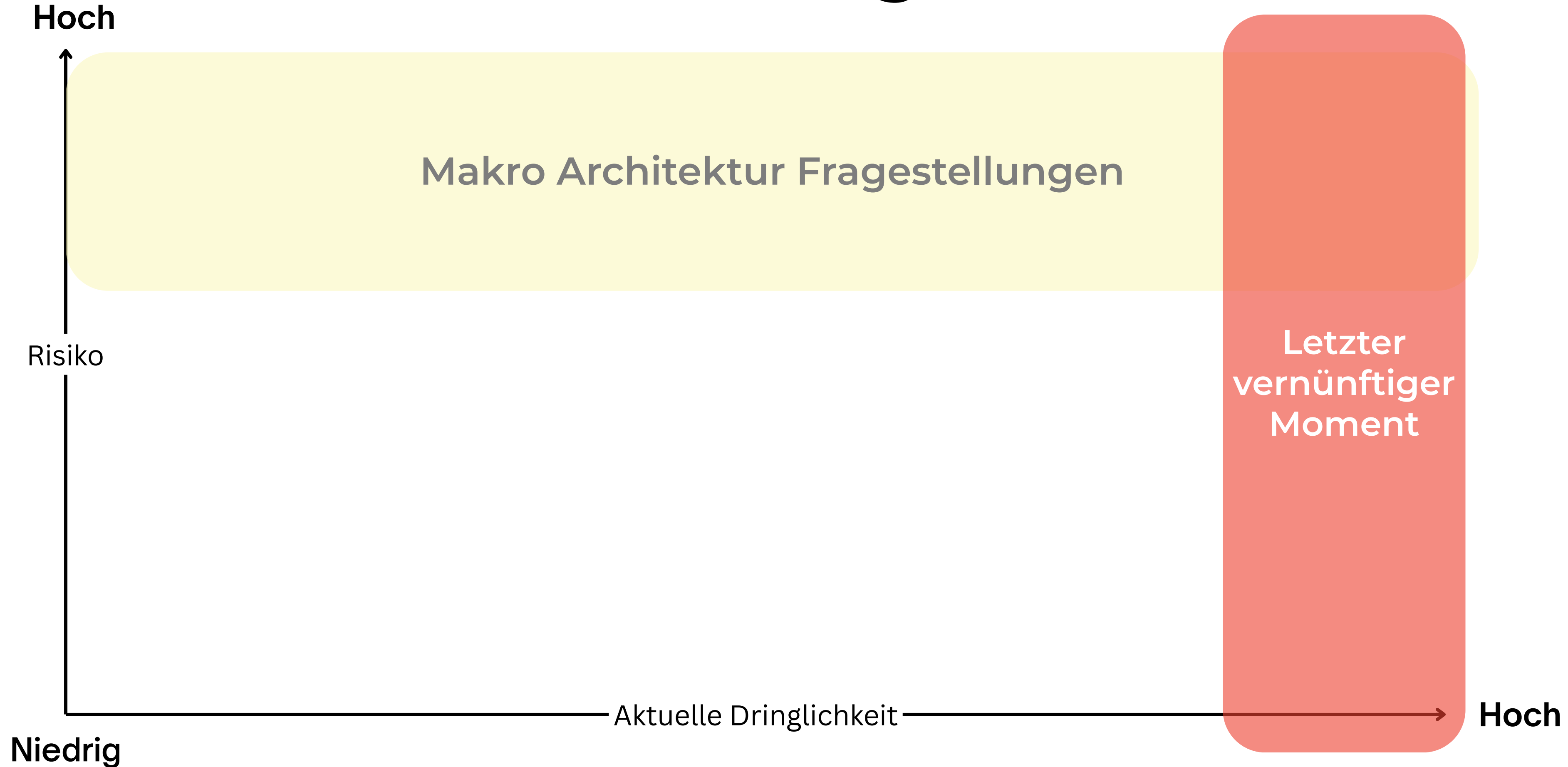
Letzter vernünftiger Moment



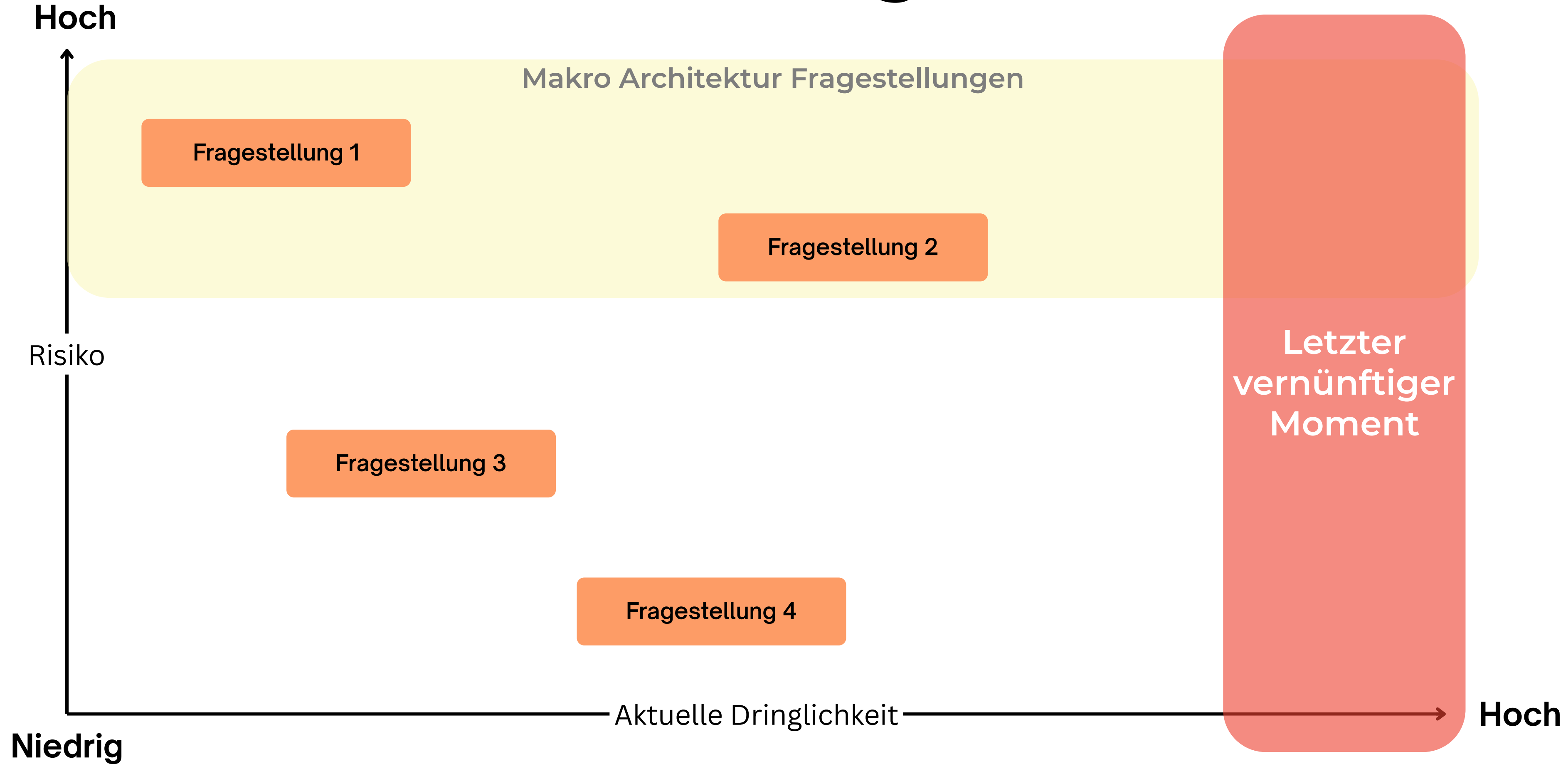
Letzter vernünftiger Moment



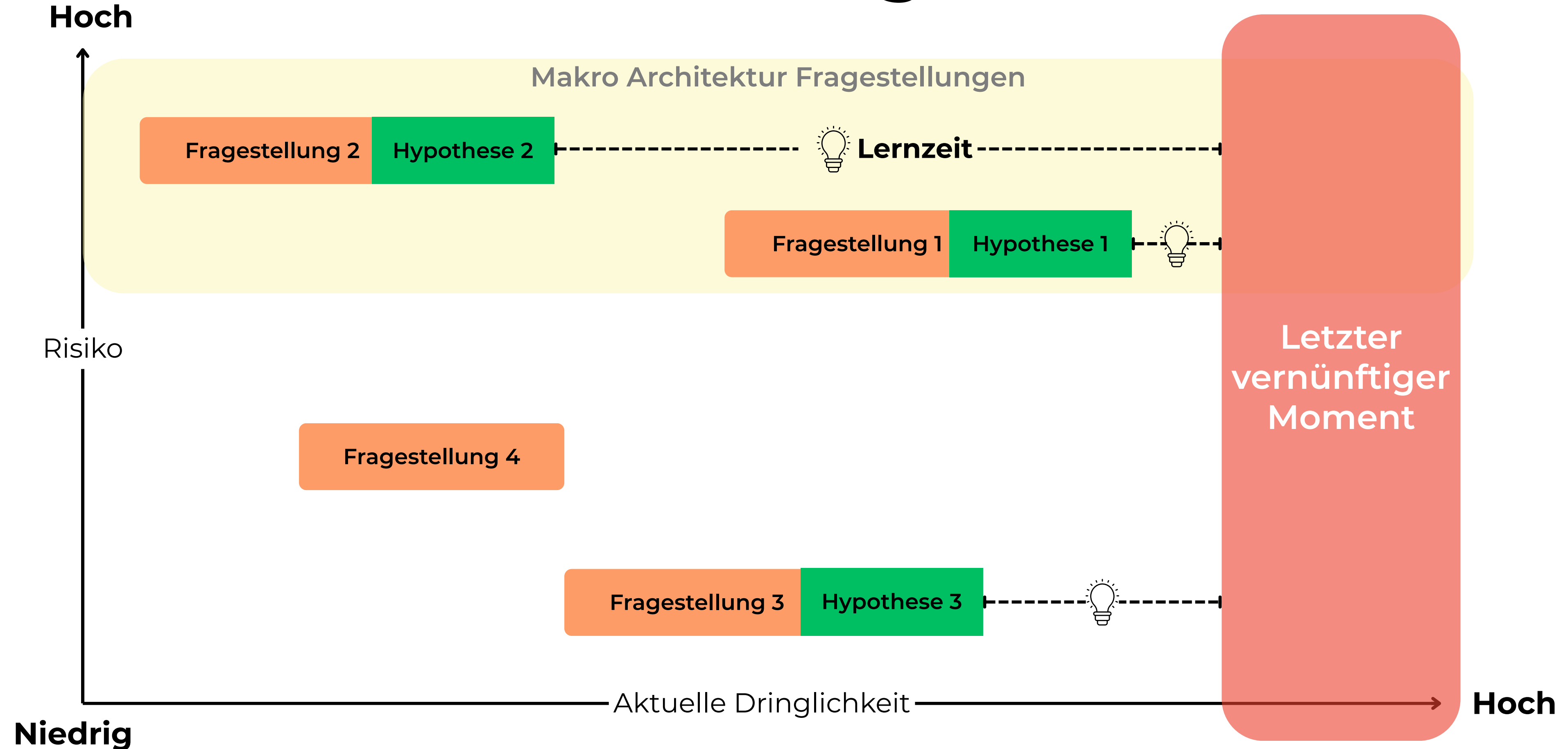
Letzter vernünftiger Moment



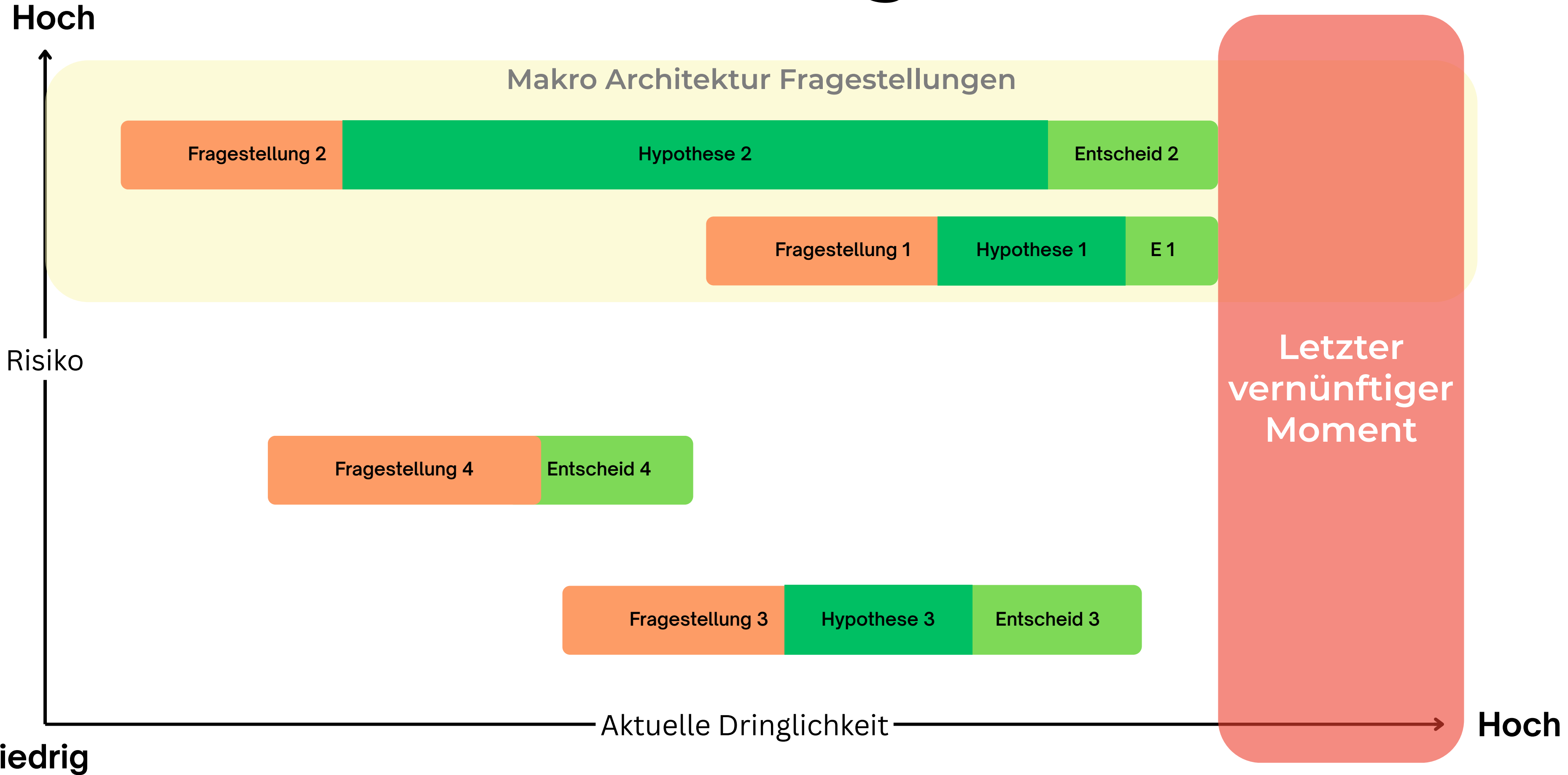
Letzter vernünftiger Moment



Letzter vernünftiger Moment



Letzter vernünftiger Moment



Warum «Hypothese»?

- ✓ Unsicherheit und Unwissenheit akzeptieren
- ✓ Reduzierung von Risiken
- ✓ Flexibilität und Anpassbarkeit erhöhen
- ✓ Verbesserte Kommunikation
- ✓ Fundierte evidenzbasierte Entscheidungen

the architecture, which have the highest priority for the most important stakeholder

Architectural hypotheses

1. Unabhängig deploybares und betreibbares Sellify System mittels eigener Persistenz. (#1 Kompatibilität)
2. Alle Datenübertragungen zum eigenen Shopsystem erfolgen über asynchrone Kanäle (#1 Kompatibilität). 
3. Export- und Importprozesse laufen in dedizierten Worker-Services, getrennt vom UI-Request Flow. (#1 Kompatibilität)
4. Client-Seitige Validierungen von Pflichtfelder und Formate, Triggering von API im Hintergrund - keine Benutzeraktion wartet direkt auf das Backend. (#2 Benutzerbarkeit)
5. Adapterbasierte Importarchitektur mit deklarativer Konfiguration (#3 Flexibilität)

Architecture Decision Records

Decision record template by Michael Nygard

This is the template in [Documenting architecture decisions - Michael Nygard](#). You can use [adr-tools](#) for managing the ADR files.

In each ADR file, write these sections:

Title

Status

What is the status, such as **proposed**, accepted, rejected, deprecated, superseded, etc.?

Context

What is the issue that we're seeing that is motivating this decision or change?

Decision

What is the change that we're proposing and/or doing?

Consequences

What becomes easier or more difficult to do because of this change?

Architecture Inception Canvas

Software System:

Designed by Team:

Workshop Date:

Iteration:

Business Case

Brief description of the business case or economic driver behind the software system.



Functional Overview



The most important functional requirements at a high level

Business Context



Delimits your system under construction as blackbox from all its communication partners. Communication partners are neighbouring systems and users.

Organisational Constraints



Any organisational requirement that restricts software architects' freedom to make decisions.

Quality Goals



Top three quality goals for the architecture which have the highest priority to the main stakeholder

Technical Constraints



Any technical requirement that restricts software architects' freedom to make decisions.

Architectural hypothesis



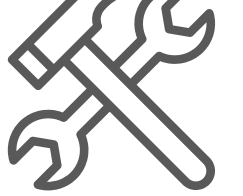
Resulting architectural hypothesis and important, expensive, large-scale, or risky architectural decisions, including justifications.

Challenges & Risks

Identified current known challenges technical risks



Technical Constraints



Any technical requirement that restricts the software architects' freedom of decision.

Technical Challenges & Risks

Identified current known challenges technical risks



Risk Storming

Step 1: Draw some software architecture diagrams

Draw one or more diagrams to show what you're planning to build or change, ideally at different levels of abstraction (e.g. using the [C4 model](#)).



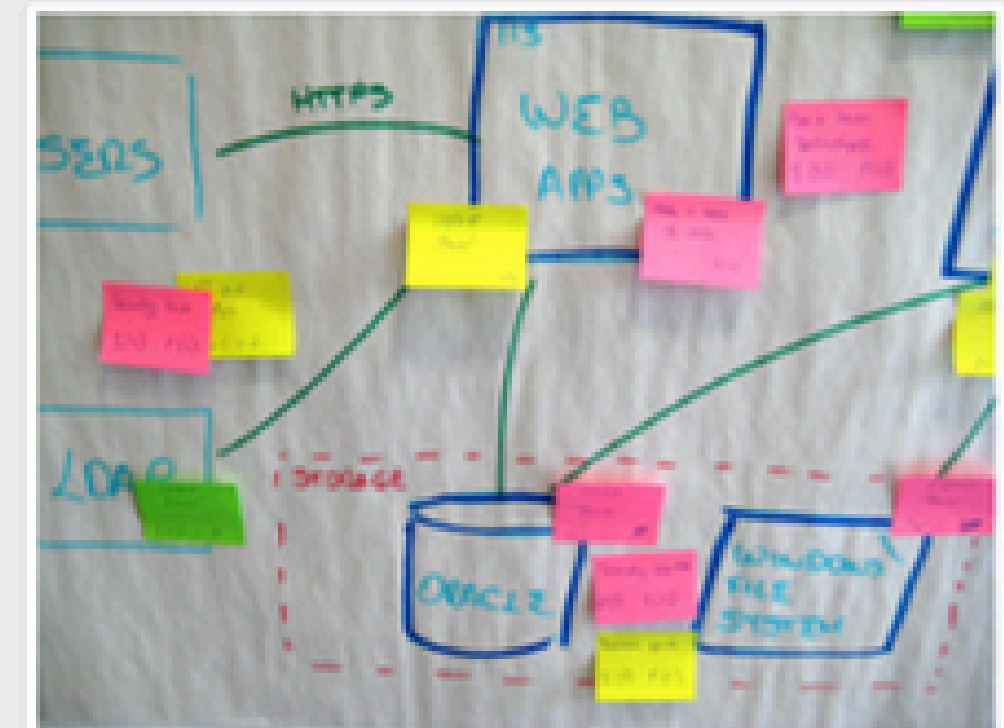
Step 2: Identify the risks individually

Gather people in front of the diagrams, and ask them to identify what they personally perceive to be risky. Write a summary of each risk on a separate sticky note, colour coded to represent low, medium, and high priority risks. Timebox this exercise (e.g. 10 minutes), and do it in silence.



Step 3: Converge the risks on the diagrams

Ask everybody to place their sticky notes onto the diagrams, sticking them in close proximity to the area where the risk has been identified.



Step 4: Review and summarise the risks

Review and summarise the output, especially focussing on risks that only one person identified, or risks where multiple people disagree on the priority.

Any technical requirement that restricts the software architects' freedom of decision.



Technical Challenges & Risks

1. Risiko von Inkonsistenzen oder Divergenz der Produktdaten (#1)
2. Verzögerte Rückmeldung der Datenübertragung (#2)
3. Ressourcensteuerung bei Lastspitzen (#3)
4. Komplexität bei Fehlerfeedback, Divergenz Frontend-Validierung und Backend-Validierung (#4)
5. Hoher Initialaufwand für Adapterkonzept, Grenzen deklarativer Konfiguration (#5)

Architecture Inception Canvas

Software System:
Next Gen Webshop

Designed by Team:
Next Gen Product Team

Workshop Date:
28.10.2025

Iteration:
1

Business Case

Erschliessung neuer Märkte und Erweiterung des eigenen Produktportfolios in dem man externe Shops & private Verkäufer auf dem eigenen, reichweite starken, Shop zulässt.



Functional Overview



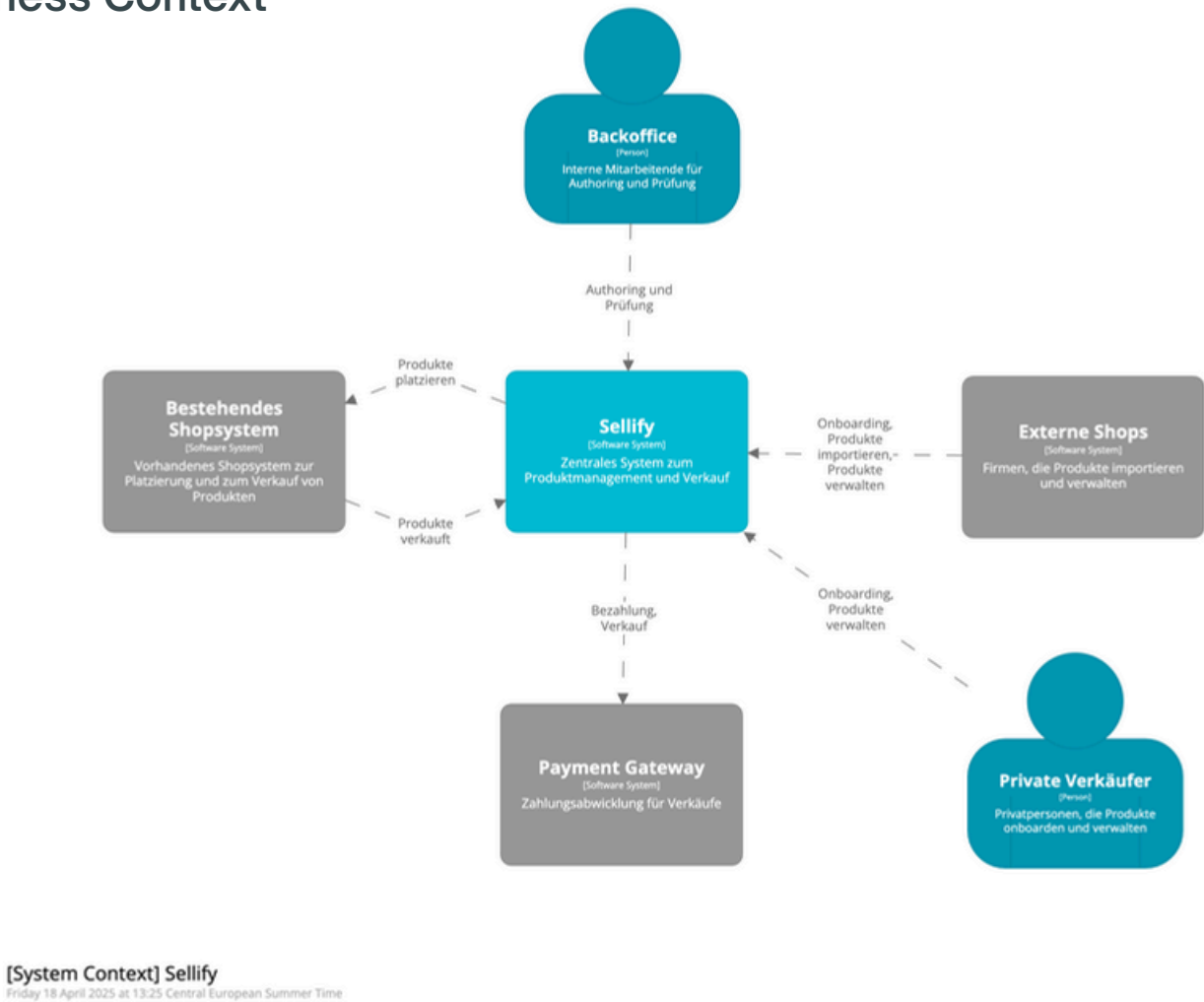
- Externe Produkte ausserhalb des eigenen Produktportfolios einstellen und verwalten
- Direktverkauf über externe bestehenden Online-Shops
- Verkauf von gebrauchten Waren

Quality Goals



1. Kompatibilität (Compatibility / Co-existence)
Sellify-Funktionalitäten und -Lasten dürfen die Performance, Verfügbarkeit und Benutzbarkeit des eigenen Onlineshops für eigene Produkte nicht beeinträchtigen.
2. Benutzbarkeit (Usability):
Verkäufer:innen können ihre Angebote intuitiv und ohne Einarbeitungszeit erstellen.
3. Flexibilität (Flexibility):
Neue externe Online-Shop Systeme können mit wenig Aufwand ihre Produkte synchronisieren.

Business Context



Organisational Constraints



- Architektur-Entscheidungen müssen durch das Architekturboard genehmigt werden.
- Erster fachlicher Durchstich sollte nach drei Monaten für erste ausgewählte Testkunden sichtbar sein
- Die Entwicklungskosten für das MVP dürfen nicht 300k überschreiten.

Technical Constraints



- Hosting auf unserer Azure Plattform
- Bestehendes React UI Komponenten-Framework soll wiederverwendet werden, um ein einheitliches Branding zu gewährleisten
- Es muss die bestehende CI/CD Umgebung verwendet werden.

Architectural hypotheses



- Unabhängig deploybares und betreibbares Sellify System mittels eigener Persistenz. (#Q1)
- Alle Datenübertragungen zum eigenen Shopsystem erfolgen über asynchrone Kanäle (#Q1).
- Export- und Importprozesse laufen in dedizierten Worker-Services, getrennt vom UI-Request Flow. (#Q1)
- Client-Seitige Validierungen von Pflichtfeldern und Formaten, Triggering von API im Hintergrund - keine Benutzeraktion wartet direkt auf das Backend. (#Q2)
- Adapterbasierte Importarchitektur mit deklarativer Konfiguration (#Q3)

Technical Challenges & Risks



- Risiko von Inkonsistenzen oder Divergenz der Produktdaten (#A1)
- Verzögerte Rückmeldung der Datenübertragung (#A2)
- Ressourcensteuerung bei Lastspitzen (#A3)
- Komplexität bei Fehlerfeedback, Divergenz Frontend-Validierung und Backend-Validierung (#A4)
- Hoher Initialaufwand für Adapterkonzept, Grenzen deklarativer Konfiguration (#A5)

Wie geht's weiter?



Nicht nur **Inception** der
Architektur, sondern auch der
Architektur-Dokumentation.

1. Introduction & Goals

Fundamental requirements, esp. quality goals

2. Constraints

Regulations and external constraints

3. Context & Scope

External systems & interfaces

4. Solution Strategy

Core ideas and solution approaches

5. Building Block View

Structure of source code, modularization (hierarchical)

6. Runtime View

Important runtime scenarios

7. Deployment View

Hardware, infrastructure & deployment

8. Crosscutting Concepts

Cross-cutting topics, often very technical and detailed

9. Architectural Decisions

Important decisions (not described elsewhere)

10. Quality Requirements

Quality tree, quality scenarios

11. Risks & Technical Debt

Known problems and risks

12. Glossary

Important and specific terms ("ubiquitous language")

1. Introduction & Goals

Fundamental requirements, esp. quality goals



2. Constraints

Regulations and external constraints



3. Context & Scope

External systems & interfaces



4. Solution Strategy

Core ideas and solution approaches



5. Building Block View

Structure of source code, modularization (hierarchical)

7. Deployment View

Hardware, infrastructure & deployment

8. Crosscutting Concepts

Cross-cutting topics, often very technical and detailed

9. Architectural Decisions

Important decisions (not described elsewhere)



10. Quality Requirements

Quality tree, quality scenarios

11. Risks & Technical Debt

Known problems and risks



6. Runtime View

Important runtime scenarios

12. Glossary

Important and specific terms ("ubiquitous language")

Architecture Inception Canvas

Software System:

Designed by Team:

Workshop Date:



Business Case

Brief description of the business case or economic driver behind the software system.

Functional Overview



The most important functional requirements at a high level

1.Introduction & Goals
--> 1.1. Requirements Overview

Quality Goals



The three most important quality goals for the architecture, which have the highest priority for the most important stakeholder

1.Introduction & Goals
--> 1.2. Quality Goals

Business Context



Separate your system under construction as a black box from all its communication partners. Communication partners are neighbouring external systems and users.

3. Context & Scope
--> 3.1. Business Context

Organisational Constraints



Any organisational requirement that restricts software architects' freedom to make decisions.

2. Constraints

Technical Constraints



Any technical requirement that restricts software architects' freedom to make decisions.

2. Constraints

Architectural hypothesis



Resulting architectural hypothesis and important, expensive, large-scale, or risky architectural decisions, including justifications.

9. Architectural Decisions

Technical Challenges & Risks



Identified current known challenges technical risks

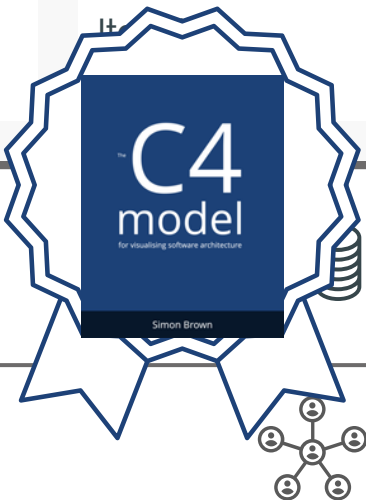
11. Risks and Technical Debt

Architecture Inception Canvas

Software System:

Designed by Team:

Workshop Date:



Business Case

Brief description of the business case or economic driver behind the software system.

Functional Overview



The most important functional requirements at a high level

Business Context



Separate your system under construction as a black box from all its communication partners. Communication partners are neighbouring external systems and users.

Organisational Constraints



Any organisational requirement that restricts software architects' freedom to make decisions.

Quality Goals



The three most important quality goals for the architecture, which have the highest priority for the most important stakeholder

1. System Context Diagram

Technical Constraints



Any technical requirement that restricts software architects' freedom to make decisions.

Architectural hypothesis



Resulting architectural hypothesis and important, expensive, large-scale, or risky architectural decisions, including justifications.

Technical Challenges & Risks



Identified current known challenges technical risks

bitmuggler	Embed Video	e3ffa93 · 2 months ago	109 Commits
.github/workflows	Replace kramdown-asciidoc with docToolchain	last year	
documentation	Improvements	2 months ago	
.adr-dir	Example ADRs included	2 years ago	
.gitignore	Improvements	2 months ago	
README.adoc	Embed Video	2 months ago	
adr-template.md	adr-template adjustments described	2 months ago	
banner.png	Banner included	2 months ago	
docToolchainConfig.groovy	use exportStructurizr from docToolchain	2 years ago	
dtcw	Improvements	2 months ago	

README

Software Architecture Documentation Starter with arc42 and C4 Model

SOFTWARE ARCHITECTURE DOCUMENTATION

STARTER



This example shows how to use [arc42](#) in combination with the [C4 model](#) with the Documentation as Code technique.

It shows how to use the techniques described in [The Ultimate Guide to Software Architecture Documentation](#).

About

This example shows how you can use arc42 in combination with the C4 model and the Documentation as Code technique.

[bitmuggler.github.io/arc42-c4-softw...](#)

- arc42
- c4
- documentation-as-code
- c4-model
- docsascode

- Readme
- Activity
- 141 stars
- 3 watching
- 25 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

- ### Contributors 2
- bitmuggler Patrick Roos
 - rdmueller Ralf D. Müller

Deployments 35

- github-pages 2 months ago

[+ 34 deployments](#)



arc42, C4 Model & Documentation-as Code – Starter Project

```
1 /*
2  * workspace "Big Bank plc" "This is an example workspace to illustrate the key features of Structurizr"
3  *
4  *
5  *
6  *
7  *
8  *
9  *
10  *
11  *
12  *
13  *
14  *
15  *
16  *
17  *
18  *
19  *
20  *
21  *
22  *
23  *
24  *
25  *
26  *
27  *
28  *
29  *
30  *
31  *
32  *
33  *
34  *
35  *
36  *
37  *
38  *
39  *
40  *
41  *
42  *
43  *
44  *
45  *
46  *
47  *
48  *
49  *
50  *
51  *
52  *
53  *
54  *
55  *
56  *
57  *
58  *
59  *
60  *
61  *
62  *
63  *
64  *
65  *
66  *
67  *
68  *
69  *
70  *
71  *
72  *
73  *
74  *
75  *
76  *
77  *
78  *
79  *
80  *
81  *
82  *
83  *
84  *
85  *
86  *
87  *
88  *
89  *
90  *
91  *
92  *
93  *
94  *
95  *
96  *
97  *
98  *
99  *
100  */
101
102 model {
103   system "Bank" {
104     include "SystemContext"
105     include "SystemLandscape"
106     include "InternetBankingSystem"
107     include "Customer"
108     include "Mainframe"
109     include "Email"
110   }
111 }
112
113 views {
114   show as Structurizr Diagram
115   systemlandscape "SystemLandscape" {
116     include "SystemContext"
117     include "InternetBankingSystem"
118     include "Customer"
119     include "Mainframe"
120     include "Email"
121   }
122   show as Structurizr Diagram
123   systemcontext "SystemContext" {
124     include "InternetBankingSystem"
125     include "Customer"
126     include "Mainframe"
127     include "Email"
128   }
129   show as Structurizr Diagram
130   internetbankingsystem "InternetBankingSystem" {
131     include "Customer"
132     include "Mainframe"
133     include "Email"
134   }
135   show as Structurizr Diagram
136   customer "Customer" {
137     include "Mainframe"
138     include "Email"
139   }
140   show as Structurizr Diagram
141   mainframe "Mainframe" {
142     include "Email"
143   }
144   show as Structurizr Diagram
145   email "Email"
146 }
```

Watch on YouTube

.....



30 FUNDAMENTAL TECHNIQUES FOR SOFTWARE ARCHITECTS



PATRICK ROOS
TECH LEAD
SOFTWARE ARCHITECT
SOFTWARE ENGINEER



30 Fundamental Techniques for Software Architects

Discover essential techniques for software architects to design modern systems, align with business goals, and manage stakeholders effectively. Learn more in this post!

 workingsoftware.dev / Jun 2

Vielen Dank!

Alle Slides und
weiterführenden
Informationen



Patrick Roos



29. Oktober 2025

Bern