

SIBB // Digitalverband Berlin-Brandenburg, 27.05.2025, Berlin

Softwaremodernisierung mit **GenAI**

The Good
The Bad
The Unexpected

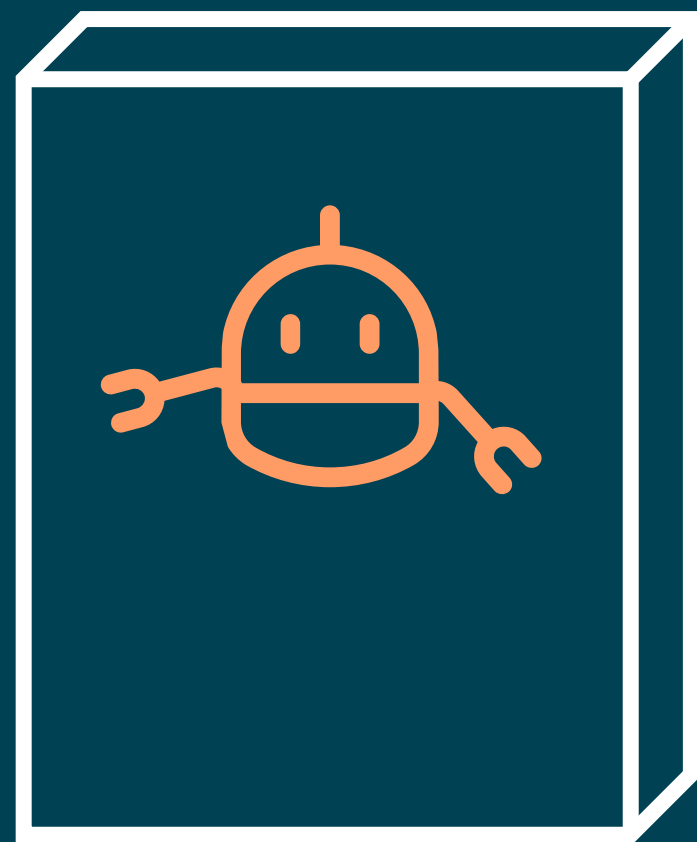
Markus Harrer
Software Evolutionist

INOQ



Softwaremodernisierung mit **GenAI**

The Good, the Bad, the Unexpected



~~GenAI in Software~~

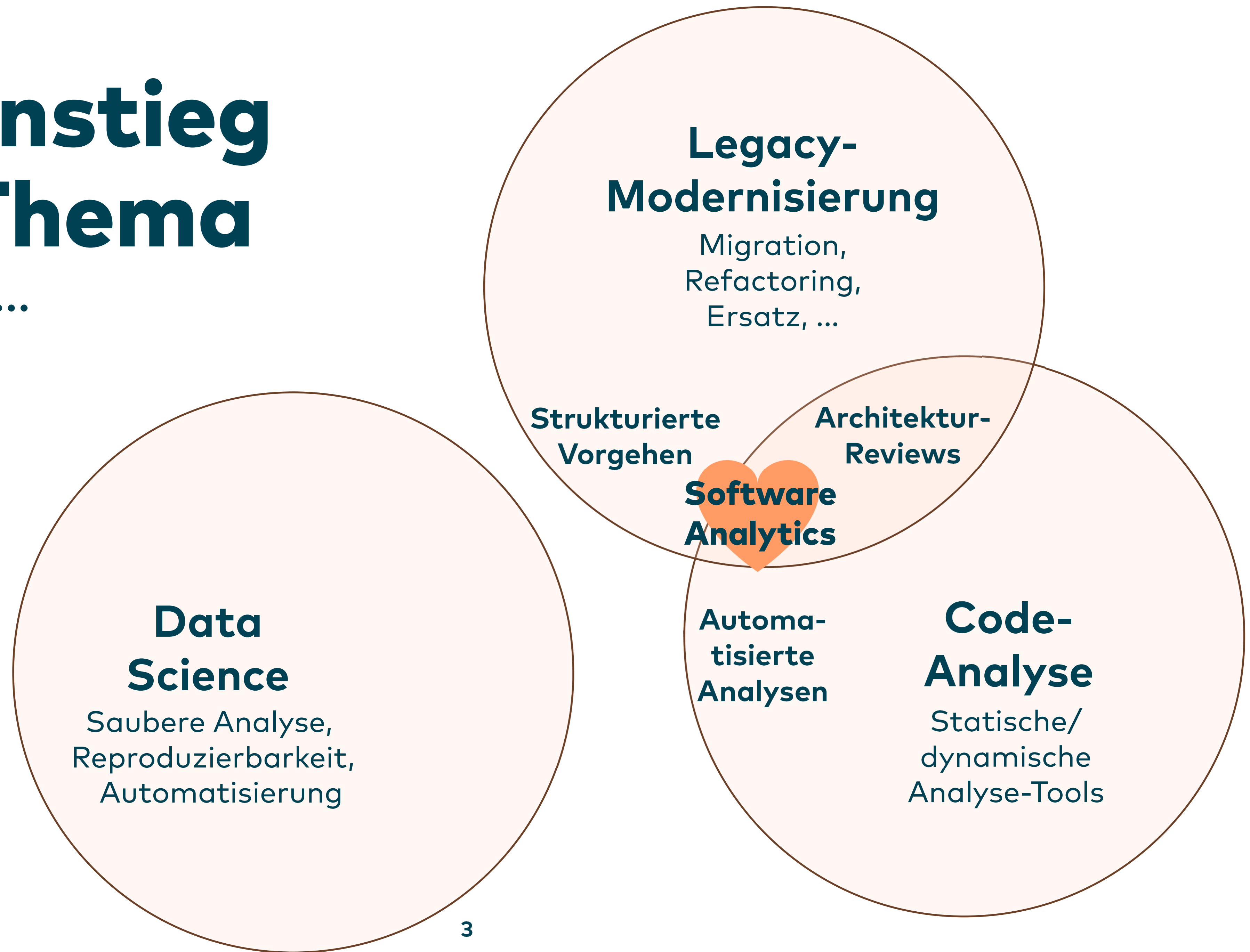


GenAI um Software



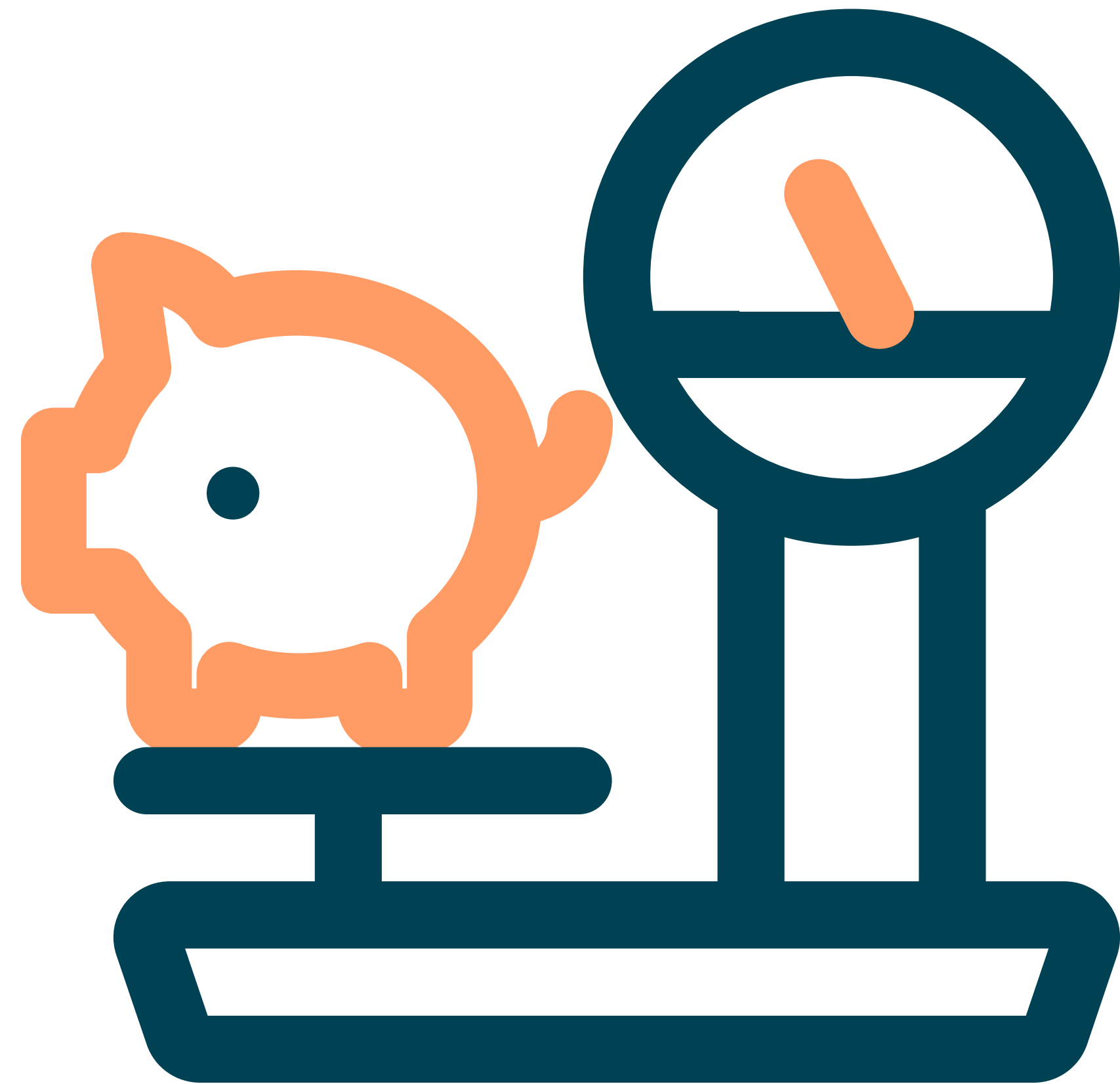
Mein Einstieg in das Thema

back in 2014 ...



Das Problem von Software Analytics

"Vom Wiegen wird
die Sau nicht fett."



Eine neue Hoffnung

ab 2019

Deep Learning
Program
Generator
CLGen

```
$ clgen
```

```
/* === SAMPLE 1 === */
__kernel void A(__global const* a, __global const* b, __global const* c, const int d) {
    int e = get_global_id(0);

    size_t f = get_global_id(0);
    size_t g = get_global_id(1);

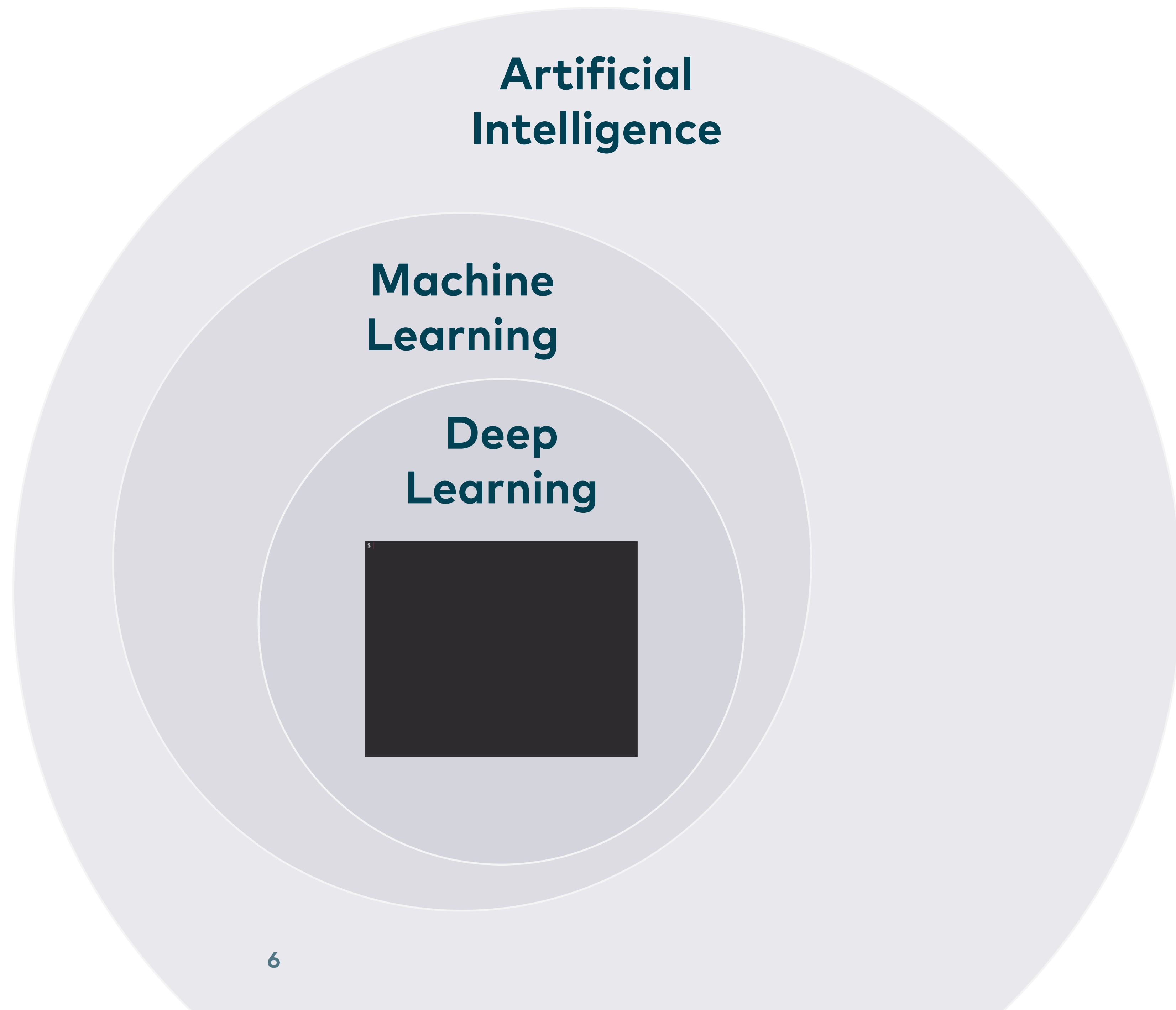
    float h = 0;
    for (f = 0; f < c.y + 1; f++) {
        const __global float4 *ba = (__global float4 *) + (((1 + ((b / 2.0f * - c - 1) * b .
et_local_id(0)]));
        b[c] = (float4)(read_imagef(a, b, a));
    }
}

/* === SAMPLE 2 === */
|
```



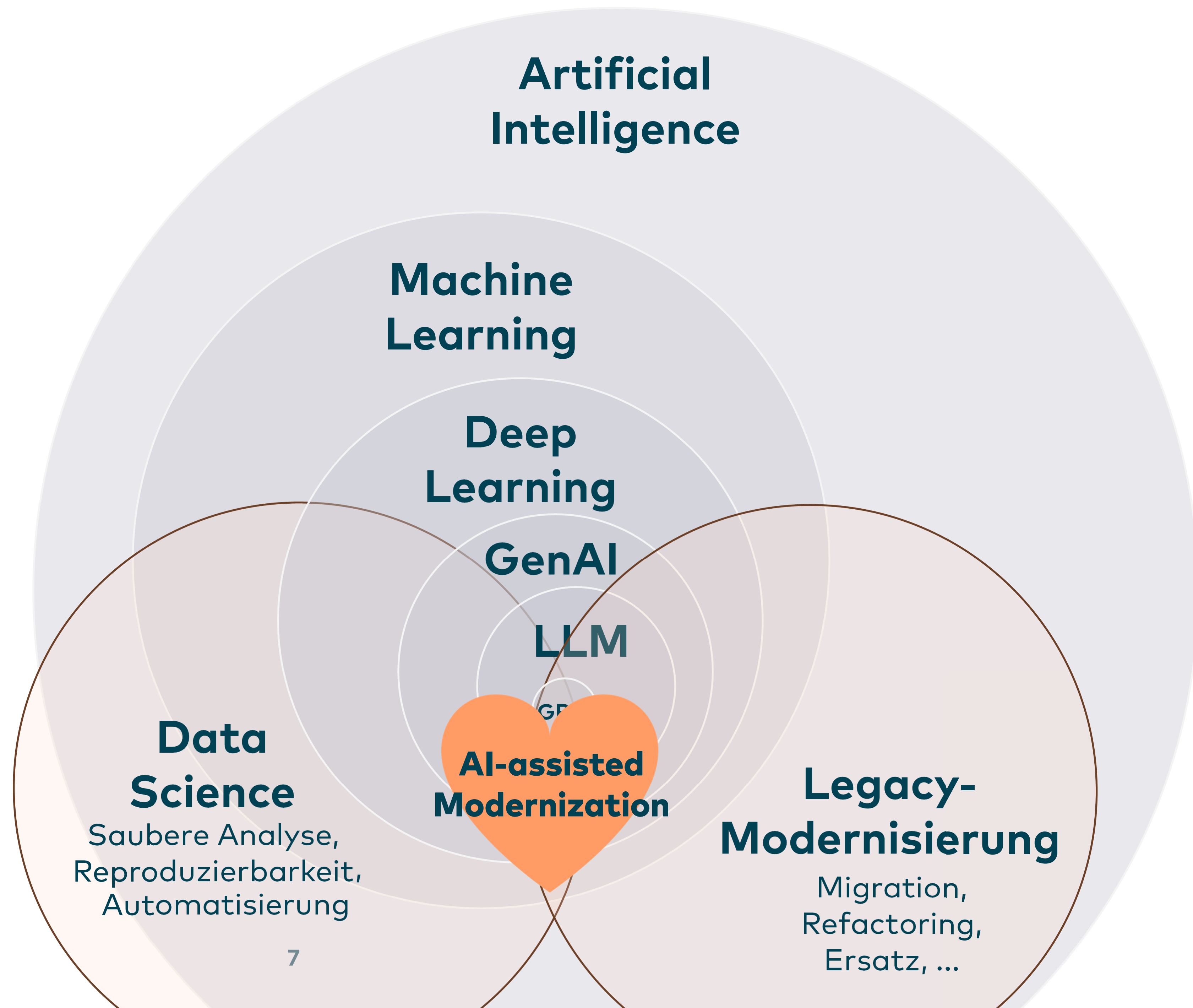
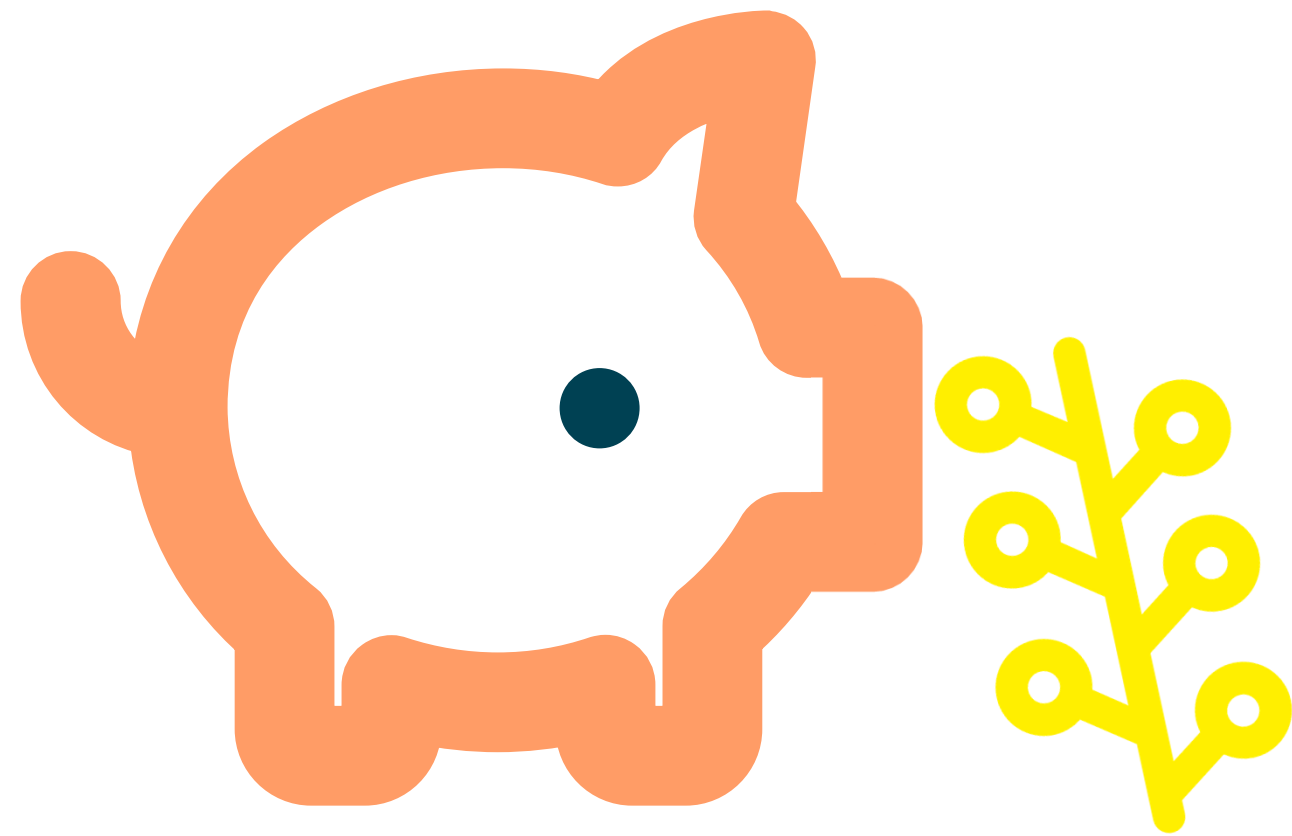
Eine neue Hoffnung

ab 2019

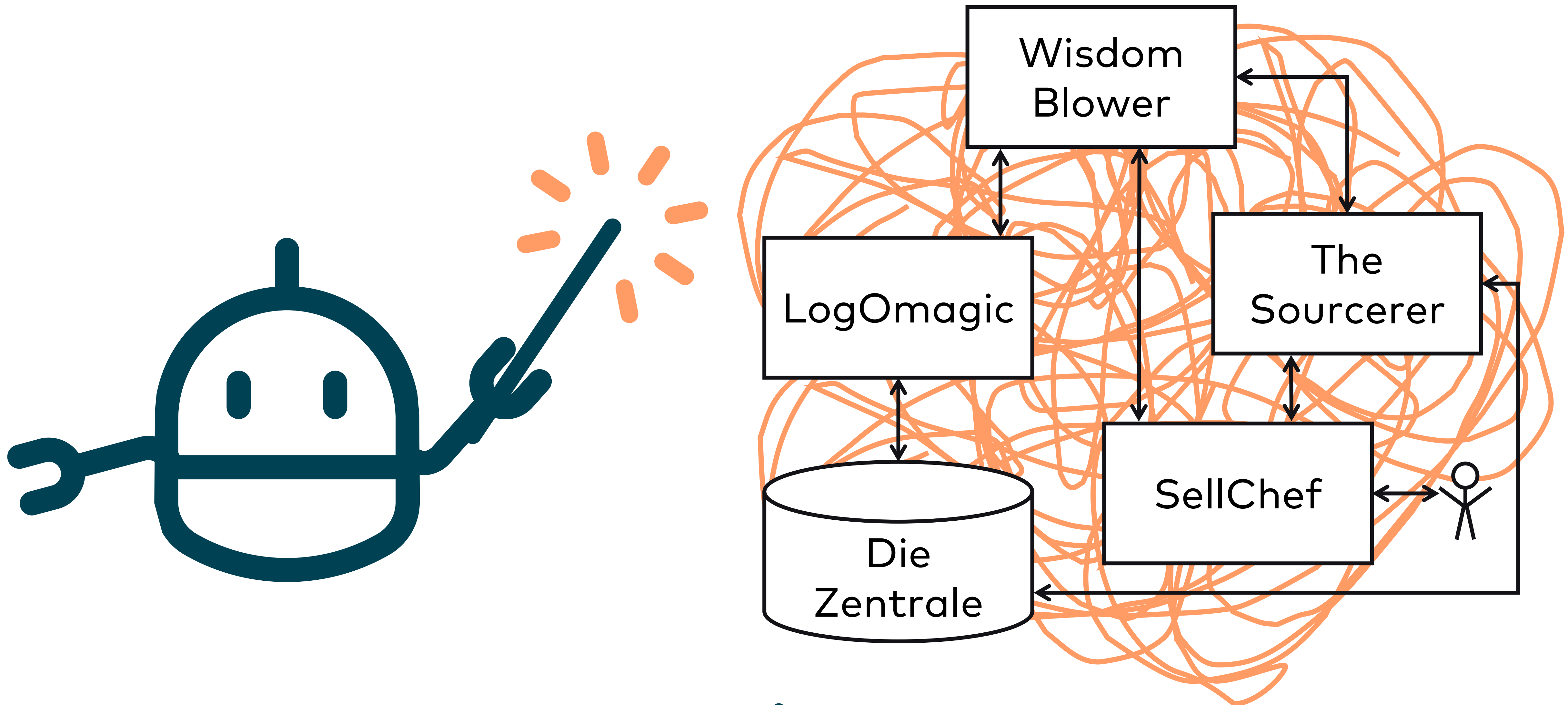


Totale Euphorie

seit 2023



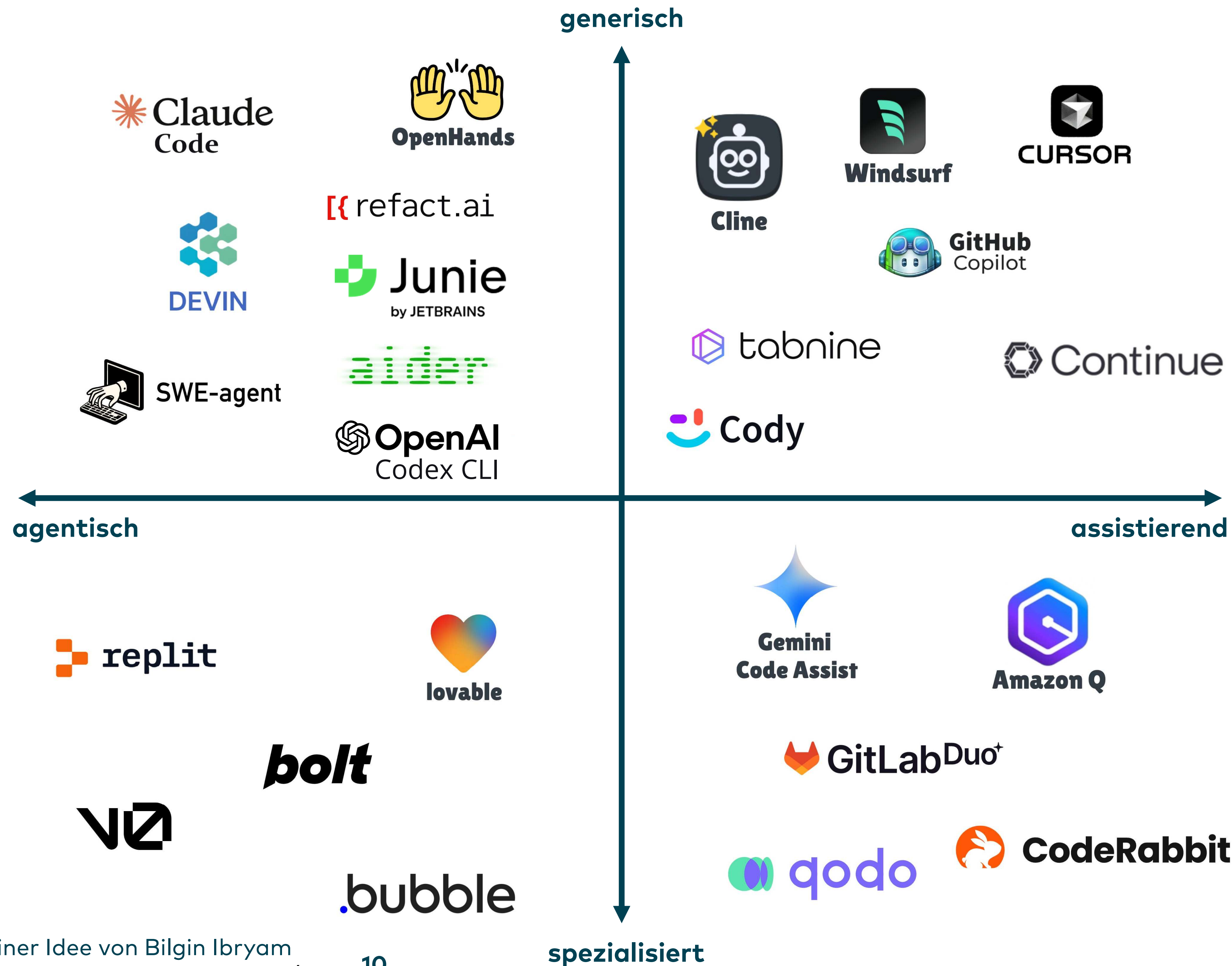
GenAI to the Rescue?



The Good

Endlich auch coole Tools für Legacy Systeme

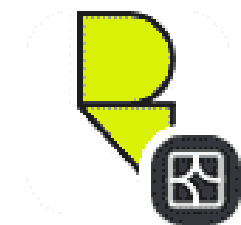
Mit Tools von morgen Systeme von gestern heute verbessern



Übernommen, korrigiert und ergänzt nach einer Idee von Bilgin Ibryam
<https://generativeprogrammer.com/p/ai-coding-assistants-landscape/>

AI-assisted Root Cause Analysis

1 reply



Resolve APP 12:21 PM

At 12:20:05 PM, the `website` service generated more than 20 log events with HTTP status codes between 500 and 599 within the last 30 minutes, exceeding the threshold.

`MongoServerError` exceptions for aggregation timeouts occurred at the `/api/furnished-apartments-{city}-{country}/map` endpoint around the alert time, suggesting database query issues.

The recommended action is to investigate the **database queries** executed by this endpoint and ensure the `maxTimeMs` parameter is appropriately configured to prevent timeout errors. (Running Pods).

Open in Resolve.ai

Use `@Resolve` to ask follow-up questions

Logs
Metrics

Code
API

Docs
Runbooks

Slack

Softwarearchäologie

mit Claude Code auf Basis der Git-Historie

„Fasse mir die wichtigsten vergangenen Entwicklungen der Scheduling-Komponente zusammen.“

- Die wichtigsten Entwicklungen der Scheduling-Komponente:
 - März 2013: Mike startete das Feature mit grundlegenden Implementierungen
 - Juli 2013: REST-Schnittstelle wurde von Mike entwickelt
 - Dezember 2013: Umfassende Überarbeitung mit AddSchedulingUser/Date und Voting
 - Januar 2014: AWS-Backend-Integration durch Mike und Charlie
 - Februar 2014: Refactoring mit Enum für YES/NO/MAYBE und XSS-Schutz
 - Februar/März 2014: Implementierung der Löschfunktionen
 - Juli 2016: Finale Verbesserungen mit "final date" und "event Date" Funktionen

Mike war hauptverantwortlich für die Grundarchitektur, während Charlie und Matthew später wichtige Verbesserungen beitrugen.

Vibe Coding von Analyse-Tools

Concept Editor

Settings Refresh Cache Set Search Path Cache Status

Concepts Editor

Raw YAML Structured Editor

```
1 root_dir: "/mnt/c/dev/repos/OpenClinica-fst"
2 language_globs:
3   - "**/*.java"
4
5 technical_concepts:
6   - name: Data Transfer Objects (DTO)
7     explanation: Classes used to transfer data between layers of the application.
8   - rationale: DTOs decouple the data representation from the domain model, allowing for flexibility and preventing accidental exposure of internal data structures. They are commonly used in service layers and web APIs.
9   file_globs:
10    - "**/dto/*.java"
11    - "**/bean/*.java"
12    - "**/ws/bean/*.java"
13   - name: Controller
14     explanation: Classes responsible for handling incoming requests and coordinating the application's response.
15   - rationale: Controllers act as the entry point for user interactions and orchestrate the flow of data between the presentation layer, service layer, and data access layer.
16   file_globs:
17    - "**/control/*.java"
18    - "**/controller/*.java"
19   - name: Service Layer
20     explanation: Classes providing business logic and encapsulating application functionality.
21   - rationale: The service layer separates business logic from the controller and data access layers, promoting modularity, testability, and reusability.
22   file_globs:
23    - "**/service/*.java"
24    - "**/logic/*.java"
25   - name: Data Access Object (DAO)
26     explanation: Classes responsible for interacting with the data storage layer.
27   - rationale: DAOs abstract the data access logic, allowing for flexibility in choosing the data storage technology and simplifying data access operations.
28   file_globs:
29    - "**/dao/*.java"
30   - name: Exception Handling
31     explanation: Classes and mechanisms for handling errors and exceptions.
32   - rationale: Proper exception handling ensures application stability and provides meaningful error messages to users.
33   file_globs:
34    - "**/exception/*.java"
35   - name: Validation
36     explanation: Classes and mechanisms for validating data.
37   - rationale: Validation ensures data integrity and prevents invalid data from being processed.
38   file_globs:
39    - "**/validator/*.java"
40    - "**/form/*.java"
41   - name: Web Services (WS)
42     explanation: Classes defining and implementing web service interfaces.
43   - rationale: Web services allow different applications to communicate and exchange data over a network.
44   file_globs:
45    - "**/ws/*.java"
46    - "**/ws/client/*.java"
```

Save Changes Reload from concepts.yaml Load from Backup

File Matches

Controller

Classes responsible for handling incoming requests and coordinating the application's response.

Controllers act as the entry point for user interactions and orchestrate the flow of data between the presentation layer, service layer, and data access layer.

**/control/*.java

**/controller/*.java

All Files

Total files: 45 Highlighted: 26 Cache: Pattern (129s ago)

View: Flat Tree

```
web/src/main/java/org/akaza/openclinica/control/OCTableFacadeImpl.java
web/src/main/java/org/akaza/openclinica/control/RssReaderServlet.java
web/src/main/java/org/akaza/openclinica/control/SpringServletAccess.java
web/src/main/java/org/akaza/openclinica/control/StatisticsView.java
web/src/main/java/org/akaza/openclinica/control/StudyStatisticsView.java
web/src/main/java/org/akaza/openclinica/control/StudySubjectStatusView.java
web/src/main/java/org/akaza/openclinica/control/XmlView.java
web/src/main/java/org/akaza/openclinica/control/XmlViewExporter.java
web/src/main/java/org/akaza/openclinica/controller/AccountController.java
web/src/main/java/org/akaza/openclinica/controller/AnonymousFormControllerV2.java
web/src/main/java/org/akaza/openclinica/controller/BatchCRFMigrationController.java
web/src/main/java/org/akaza/openclinica/controller/ChangeCRFVersionController.java
web/src/main/java/org/akaza/openclinica/controller/DiscrepancyNoteController.java
web/src/main/java/org/akaza/openclinica/controller/EditFormController.java
web/src/main/java/org/akaza/openclinica/controller/ExtractController.java
web/src/main/java/org/akaza/openclinica/controller/IdtViewController.java
web/src/main/java/org/akaza/openclinica/controller/ODMClinicalDataController.java
web/src/main/java/org/akaza/openclinica/controller/OdmController.java
web/src/main/java/org/akaza/openclinica/controller/OdmStudySubjectController.java
web/src/main/java/org/akaza/openclinica/controller/ReportController.java
web/src/main/java/org/akaza/openclinica/controller/RuleController.java
web/src/main/java/org/akaza/openclinica/controller/ScheduledJobController.java
```

Statistics

Language Breakdown

**/*.java 1355 (100.0%)

Source Code Metrics

Total Lines 282,274
Average Lines/File 208
**/*.java 282,274 (100.0%)

Concept Coverage

Loading concept coverage data...

Pattern Statistics

All Patterns 45 files
**/control/*.java 19 (42.2%)
**/controller/*.java 26 (57.8%)

Git Analysis

Top Contributors (66 files)

Joe Keremian 14 commits
Steve DiBona 5 commits
Krikor Krumlian 5 commits
Helen Taycher 3 commits
Douglas Rodrigues 2 commits

Recent Commits (10)

e9ae65878 Joe Keremian OC-7013 Subject can capture an image in the q... 9 years ago
0cc612b5d Joe Keremian OC-7012 Upload Xform with an image type 9 years ago
de68a4d8d Steve DiBona OC-6579: Map Xform dataTypes/ResponseTyp... 10 years ago
a563195fc Steve DiBona OC-6578: Map Xform String/html body input to ... 10 years ago
1a8d3802a Joe Keremian OC-6048 checking code after the major merge 10 years ago
4813a3c77 Krikor Krumlian Merge remote-tracking branch 'sdibona/OC-58... 10 years ago
4637b5854 Joe Keremian OC-6048 multiple groups / page displaying imp... 10 years ago
d27af1d66 Joe Keremian 10 years ago

Source Code

/mnt/c/dev/repos/OpenClinica-fst/web/src/main/java/org/akaza/openclinica/controller/AccountController.java

File truncated: Showing first 1000 lines. Load full file

```
1 package org.akaza.openclinica.controller;
2
3 import java.net.URLDecoder;
4 import java.util.ArrayList;
5 import java.util.Date;
6 import java.util.HashMap;
7 import java.util.List;
8 import java.util.Locale;
9 import java.util.UUID;
10
11 import javax.servlet.ServletContext;
12
13 import org.akaza.openclinica.bean.core.Role;
14 import org.akaza.openclinica.bean.core.Status;
15 import org.akaza.openclinica.bean.core.UserType;
16 import org.akaza.openclinica.bean.login.StudyUserRoleBean;
17 import org.akaza.openclinica.bean.login.UserAccountBean;
18 import org.akaza.openclinica.bean.login.UserDTO;
19 import org.akaza.openclinica.bean.managestudy.StudyBean;
20 import org.akaza.openclinica.bean.managestudy.StudySubjectBean;
21 import org.akaza.openclinica.bean.service.StudyParameterValueBean;
22 import org.akaza.openclinica.control.SpringServletAccess;
23 import org.akaza.openclinica.dao.hibernate.AuditUserLoginDao;
24 import org.akaza.openclinica.dao.hibernate.AuthoritiesDao;
25 import org.akaza.openclinica.dao.login.UserAccountDAO;
26 import org.akaza.openclinica.dao.managestudy.StudyDAO;
27 import org.akaza.openclinica.dao.managestudy.StudySubjectDAO;
28 import org.akaza.openclinica.dao.service.StudyParameterValueDAO;
29 import org.akaza.openclinica.domain.technicaladmin.AuditUserLoginBean;
30 import org.akaza.openclinica.domain.technicaladmin.LoginStatus;
31 import org.akaza.openclinica.domain.user.AuthoritiesBean;
32 import org.akaza.openclinica.i18n.util.ResourceBundleProvider;
33 import org.akaza.openclinica.service.pmanage.ParticipantPortalRegistrar;
34 import org.apache.commons.dbcp.BasicDataSource;
35 import org.slf4j.Logger;
36 import org.slf4j.LoggerFactory;
37 import org.springframework.beans.factory.annotation.Autowired;
38 import org.springframework.beans.factory.annotation.Qualifier;
39 import org.springframework.http.ResponseEntity;
40 import org.springframework.security.authentication.AuthenticationManager;
41 import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
42 import org.springframework.security.core.Authentication;
43 import org.springframework.stereotype.Controller;
44 import org.springframework.web.bind.annotation.PathVariable;
45 import org.springframework.web.bind.annotation.RequestBody;
46 import org.springframework.web.bind.annotation.RequestMapping;
47 import org.springframework.web.bind.annotation.RequestMethod;
48 import org.springframework.web.bind.annotation.ResponseStatus;
49
50 @Controller
51 @RequestMapping(value = "/accounts")
52 @ResponseStatus(value = org.springframework.http.HttpStatus.INTERNAL_SERVER_ERROR)
53 public class AccountController {
54
55     @Autowired
56     @Qualifier("dataSource")
57     private BasicDataSource dataSource;
58
59     @Autowired
60     ServletContext context;
61
62     @Autowired
```

Konzepte

```
4
5 technical_concepts:
6   - name : Data Transfer Objects (DTO)
7     explanation : Classes used to transfer data between layers of the
      application.
8     rationale : DTOs decouple the data representation from the domain
      model, allowing for flexibility and preventing accidental exposure of
      internal data structures. They are commonly used in service layers and
      web APIs.
9     file_globs :
10      - "**/dto/*.java"
11      - "**/bean/*.java"
12      - "**/ws/bean/*.java"
13   - name : Controller
14     explanation : Classes responsible for handling incoming requests and
      coordinating the application 's response.
15     rationale : Controllers act as the entry point for user interactions
      and orchestrate the flow of data between the presentation layer, service
      layer, and data access layer.
16     file_globs :
17      - "**/control/*.java"
18      - "**/controller/*.java"
19   - name : Service Layer
20     explanation : Classes providing business logic and encapsulating
      application functionality.
21     rationale : The service layer separates business logic from the
      controller and data access layers, promoting modularity, testability, and
      reusability.
```

Controller

Classes responsible for handling incoming requests and coordinating the application's response.

Controllers act as the entry point for user interactions and orchestrate the flow of data between the presentation layer, service layer, and data access layer.

`**/control/*.java`

`**/controller/*.java`

All Files

Total files: 45

Highlighted: 26

Cache: Pattern (129s ago)

View:

Flat

Tree

web/src/main/java/org/akaza/openclinica/control/OCServiceContextListener.java

web/src/main/java/org/akaza/openclinica/control/OCTableFacadeImpl.java

web/src/main/java/org/akaza/openclinica/control/RssReaderServlet.java

web/src/main/java/org/akaza/openclinica/control/SpringServletAccess.java

web/src/main/java/org/akaza/openclinica/control/StatisticsView.java

web/src/main/java/org/akaza/openclinica/control/StudyStatisticsView.java

web/src/main/java/org/akaza/openclinica/control/StudySubjectStatusView.java

web/src/main/java/org/akaza/openclinica/control/XmlView.java

web/src/main/java/org/akaza/openclinica/control/XmlViewExporter.java

web/src/main/java/org/akaza/openclinica/controller/AccountController.java

web/src/main/java/org/akaza/openclinica/controller/AnonymousFormControllerV2.j

web/src/main/java/org/akaza/openclinica/controller/BatchCRFMigrationController

Inventar

Language Breakdown

**/*.java	1355 (100.0%)
-----------	---------------

Source Code Metrics

Total Lines	282,274
Average Lines/File	208
**/*.java	282,274 (100.0%)

Concept Coverage

Loading concept coverage data...

Pattern Statistics

All Patterns	45 files
**/control/*.java	19 (42.2%)
**/controller/*.java	26 (57.8%)

Git Analysis

Top Contributors (66 files)

Joe Keremian	14 commits
Steve DiBona	5 commits
Krikor Krumlian	5 commits
Helen Taycher	3 commits
Douglas Rodrigues	2 commits

Recent Commits (10)

e9ae650709 years ago

Joe Keremian

OC-7013 Subject can capture an image in the q...

0cc612b5d9 years ago

Joe Keremian

OC-7012 Upload Xform with an image type

de68a4d8d10 years ago

Steve DiBona

OC-6579: Map Xform dataTypes/ResponseTyp...

a563105fc10 years ago

Steve DiBona

OC-6578: Map Xform String/html body input to

File truncated: Showing first 1000 lines.

Load full file

```
1 package org.akaza.openclinica.controller;
2
3 import java.net.URLDecoder;
4 import java.util.ArrayList;
5 import java.util.Date;
6 import java.util.HashMap;
7 import java.util.List;
8 import java.util.Locale;
9 import java.util.UUID;
10
11 import javax.servlet.ServletContext;
12
13 import org.akaza.openclinica.bean.core.Role;
14 import org.akaza.openclinica.bean.core.Status;
15 import org.akaza.openclinica.bean.core.UserType;
16 import org.akaza.openclinica.bean.login.StudyUserRoleBean;
17 import org.akaza.openclinica.bean.login.UserAccountBean;
18 import org.akaza.openclinica.bean.login.UserDTO;
19 import org.akaza.openclinica.bean.managestudy.StudyBean;
20 import org.akaza.openclinica.bean.managestudy.StudySubjectBean;
21 import org.akaza.openclinica.bean.service.StudyParameterValueBean;
22 import org.akaza.openclinica.control.SpringServletAccess;
23 import org.akaza.openclinica.dao.hibernate.AuditUserLoginDao;
24 import org.akaza.openclinica.dao.hibernate.AuthoritiesDao;
25 import org.akaza.openclinica.dao.login.UserAccountDAO;
26 import org.akaza.openclinica.dao.managestudy.StudyDAO;
27 import org.akaza.openclinica.dao.managestudy.StudySubjectDAO;
28 import org.akaza.openclinica.dao.service.StudyParameterValueDAO;
29 import org.akaza.openclinica.domain.technicaladmin.AuditUserLoginBean;
30 import org.akaza.openclinica.domain.technicaladmin.LoginStatus;
31 import org.akaza.openclinica.domain.user.AuthoritiesBean;
32 import org.akaza.openclinica.i18n.util.ResourceBundleProvider;
33 import org.akaza.openclinica.service.pmanage.ParticipantPortalRegistrar;
34 import org.apache.commons.dbcp.BasicDataSource;
35 import org.slf4j.Logger;
36 import org.slf4j.LoggerFactory;
37 import org.springframework.beans.factory.annotation.Autowired;
38 import org.springframework.beans.factory.annotation.Qualifier;
39 import org.springframework.http.ResponseEntity;
40 import org.springframework.security.authentication.AuthenticationManager;
41 import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
42 import org.springframework.security.core.Authentication;
43 import org.springframework.stereotype.Controller;
44 import org.springframework.web.bind.annotation.PathVariable;
45 import org.springframework.web.bind.annotation.RequestBody;
46 import org.springframework.web.bind.annotation.RequestMapping;
```

Lokale LLMs



Gemma 3



Qwen3



Phi-4

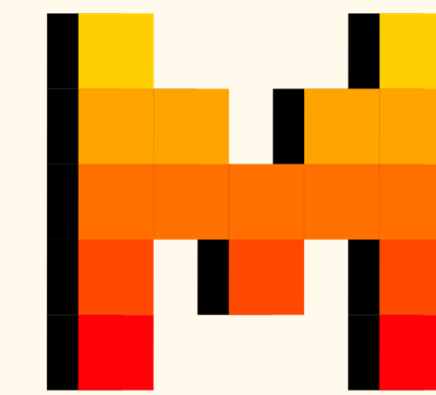


deepseek **R1**

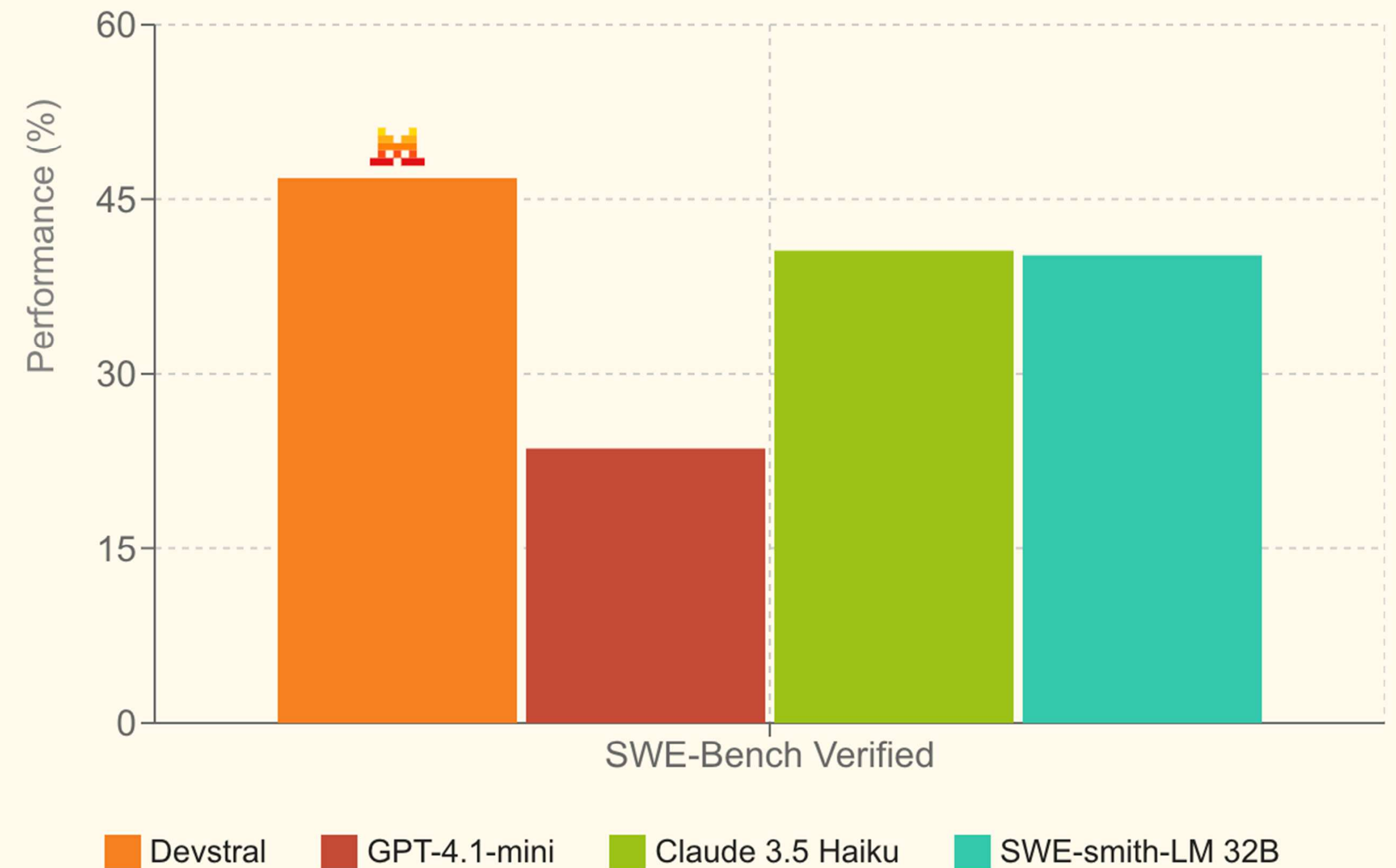
Distill-Qwen-32B



Qwen2.5-Coder



Devstral



Code-Zusammenfassungen

... als Podcast-Episode mit Googles NotebookLM

DeepDive into

DataEntryServlet.java

```
/*
 * OpenClinica is distributed under the
 * GNU Lesser General Public License (GNU LGPL).

 * For details see: http://www.openclinica.org/license
 * copyright 2003-2011 Akaza Research
 */
package org.akaza.openclinica.control.submit;

import java.io.File;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Date;
import java.util.HashMap;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import java.util.Set;
import java.util.TreeSet;

import javax.servlet.ServletConfig;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import javax.sql.DataSource;

import org.apache.commons.beanutils.BeanUtils;
import org.akaza.openclinica.bean.admin.AuditBean;
import org.akaza.openclinica.bean.admin.CRFBean;
import org.akaza.openclinica.bean.core.AuditableEntityBean;
import org.akaza.openclinica.bean.core.DataEntryStage;
import org.akaza.openclinica.bean.core.EntityBean;
import org.akaza.openclinica.bean.core.ItemDataType;
import org.akaza.openclinica.bean.core.NullValue;
import org.akaza.openclinica.bean.core.NumericComparisonOperator;
import org.akaza.openclinica.bean.core.ResolutionStatus;
import org.akaza.openclinica.bean.core.Role;
import org.akaza.openclinica.bean.core.Status;
import org.akaza.openclinica.bean.core.SubjectEventStatus;
import org.akaza.openclinica.bean.core.Utils;
```

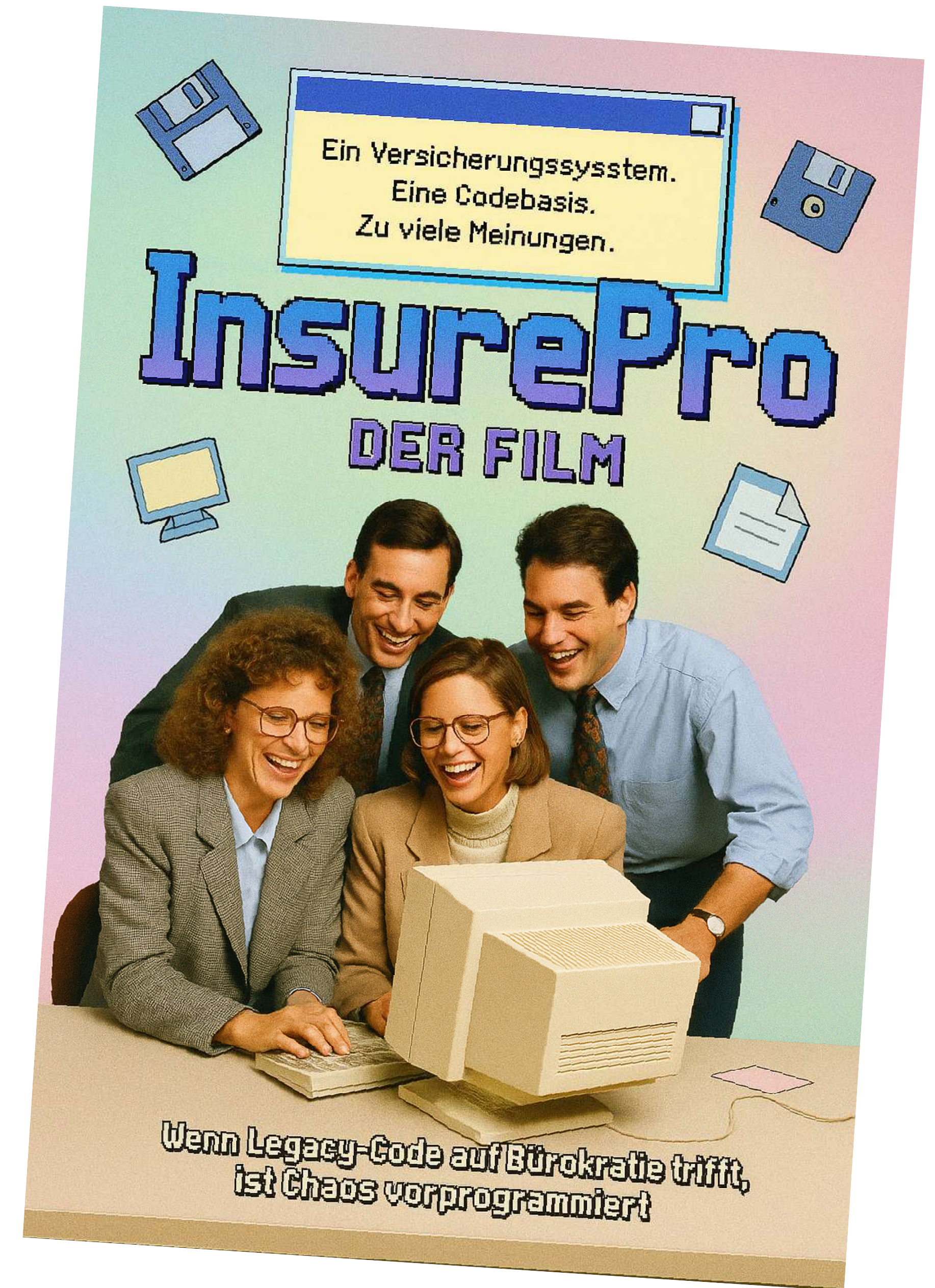
HOW LOGIN WORKS IN AN INSURANCE SOFTWARE

TikTok

```
function login(username: string: string) {  
  if user = getUserByUsername(username);  
  if (user = null) {  
    return sendErrorResponse('User not found');  
  }  
  if (verifyPassword, password == user )  
    return sendErrorResponse('Invalid password');  
  token = generateAuthToken(generateToken());  
  storeSession(setToken(token));  
  sendSuccessResponseSucces(token);  
}
```

Here's how login works

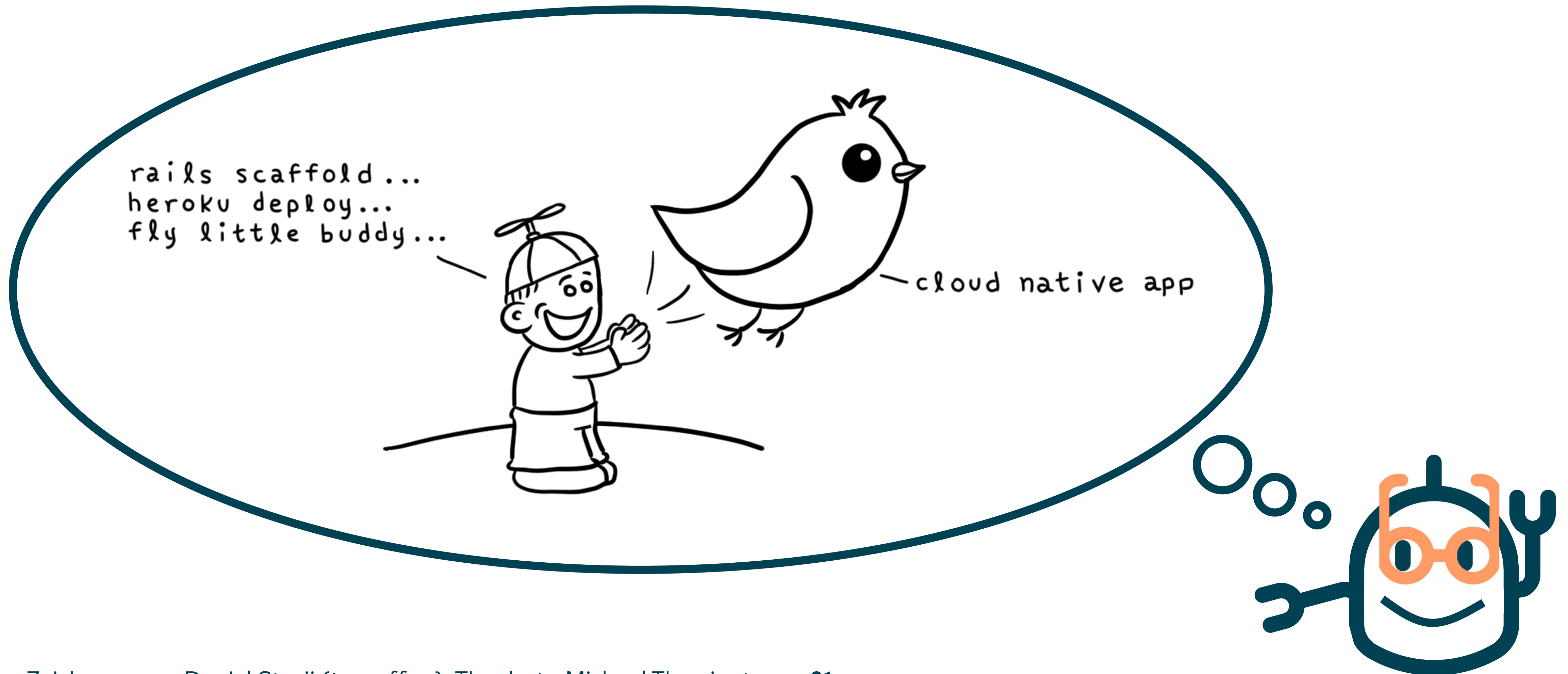
Reply



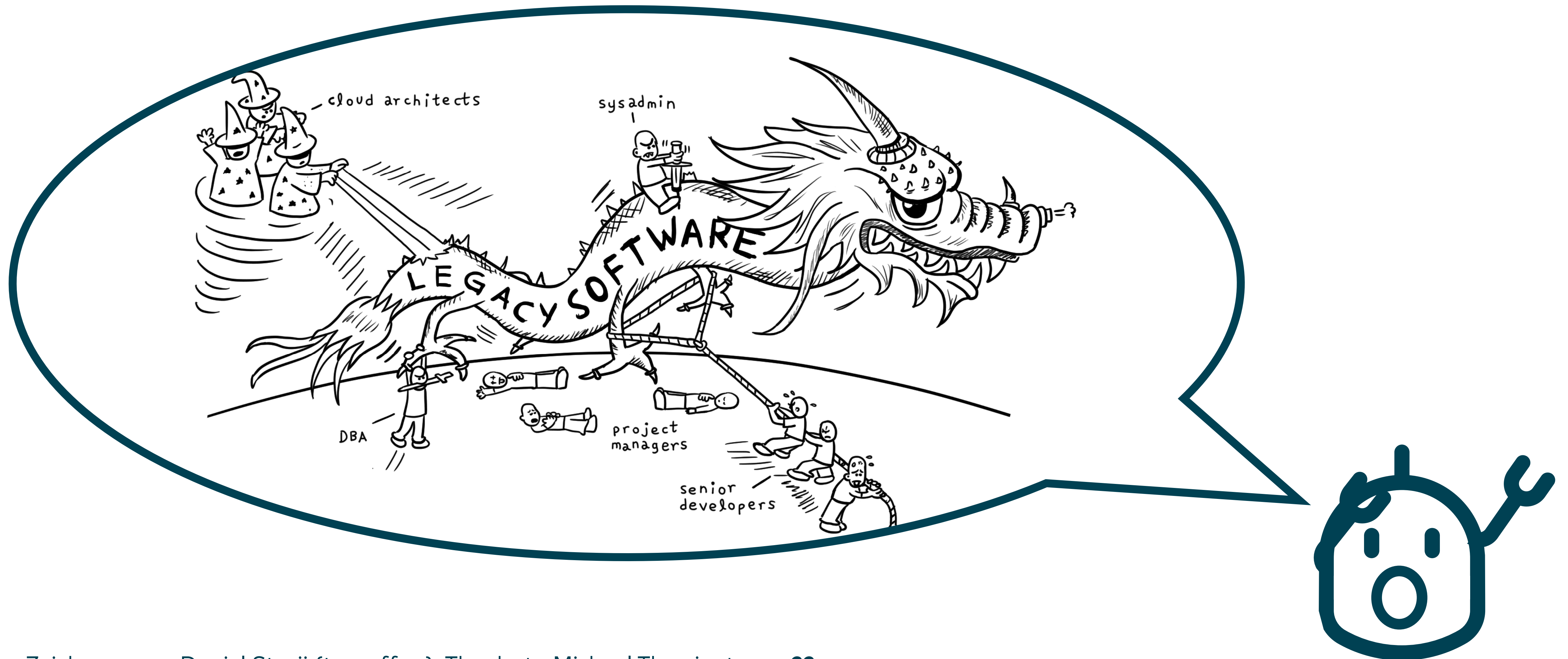
The Bad

Legacy Systeme != Heile Welt

Was Foundation Models sahen und lernten



Was LLMs jetzt sehen (müssen?)





**Die Grenzen meiner Sprache
bedeuten die Grenzen meiner
Welt.**

– Ludwig Wittgenstein

Lösungsansatz 1

Ausbau der Wissenswelt

Modellen die Hausregeln beibringen

RAG Retrieval Augmented Generation

Externes Wissen in Anfragen integrieren

Instruction Tuning

Befolgen der Anweisungen verbessern

RLHF Reinforcement Learning from Human Feedback

Durch menschliche Rückmeldung verbessern

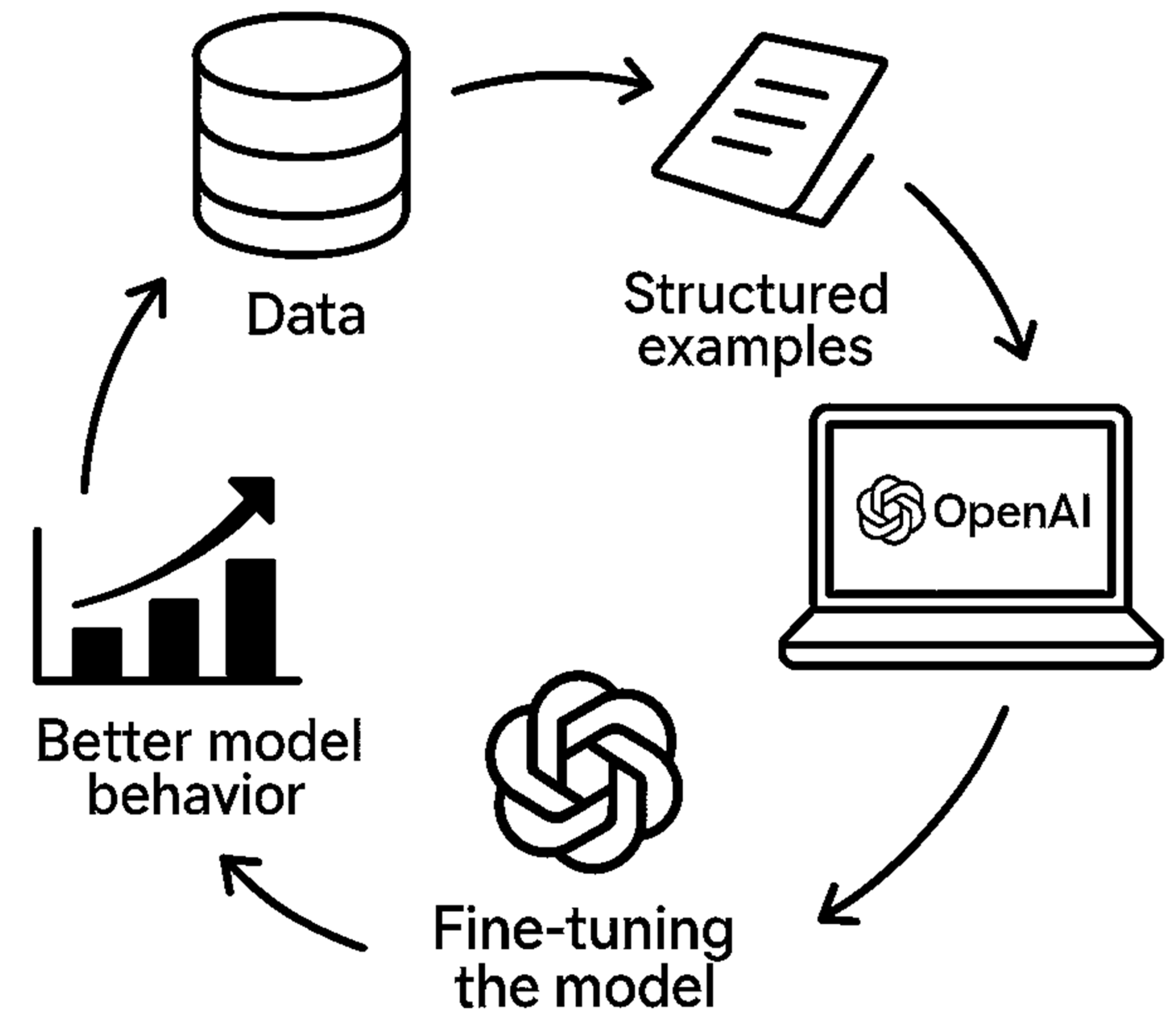
Fine-Tuning

Auf bestimmte Aufgaben feinabstimmen

Alignment

Auf ethische Werte abstimmen

Aufwand



Visualisierung von OpenAI, ziemlich sicher generiert mit ChatGPT

Lösungsansatz 2

Einschränkung der Wissenswelt

Ganz „spezielle“ Korrekturen

```
24  /**
25   * Models a {@link Vet Vet's} specialty (for example, dentistry).
26   *
27   *
28   * Add N
29   * tua
30   * er e c
31   * m c . . h
32   * l h . . t
33   * o
34   * s
35   *
36   * @author Juergen Hoeller
37   */
38  @Entity
39  @Table(name = "specialties")
40  @BoundedContext("Visits")
41  public class Specialty extends NamedEntity {
42
43  }
```

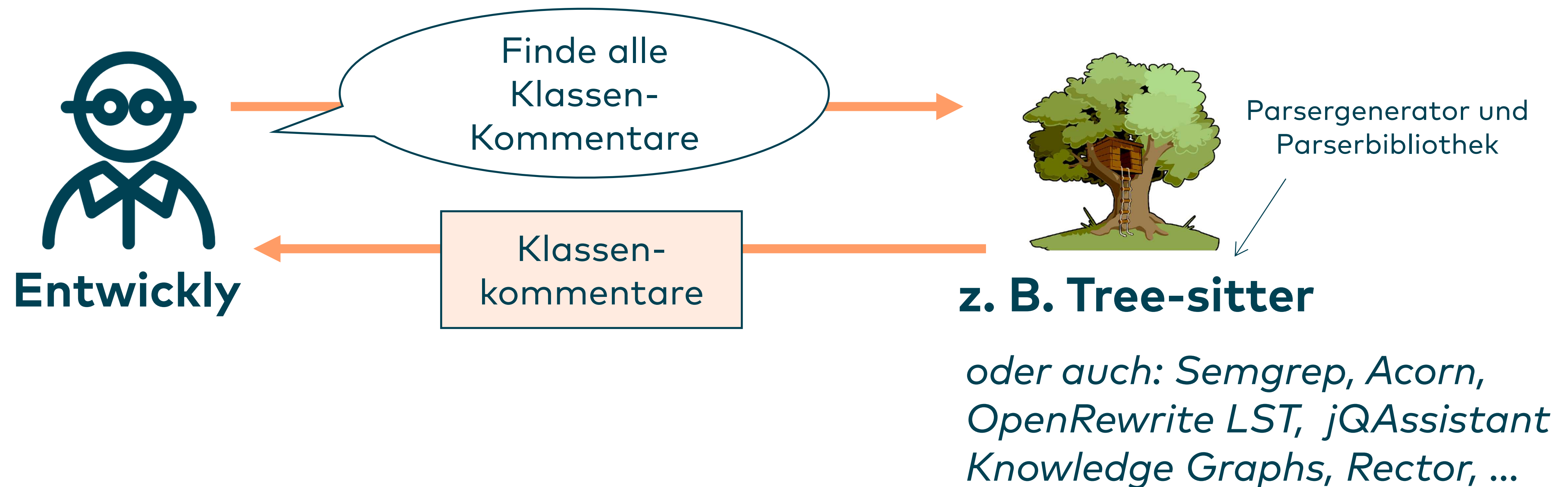
„Atemlos durch
die Nacht“

Quatsch-
kommentar

Ganz „spezielle“ Korrekturen

Analyze

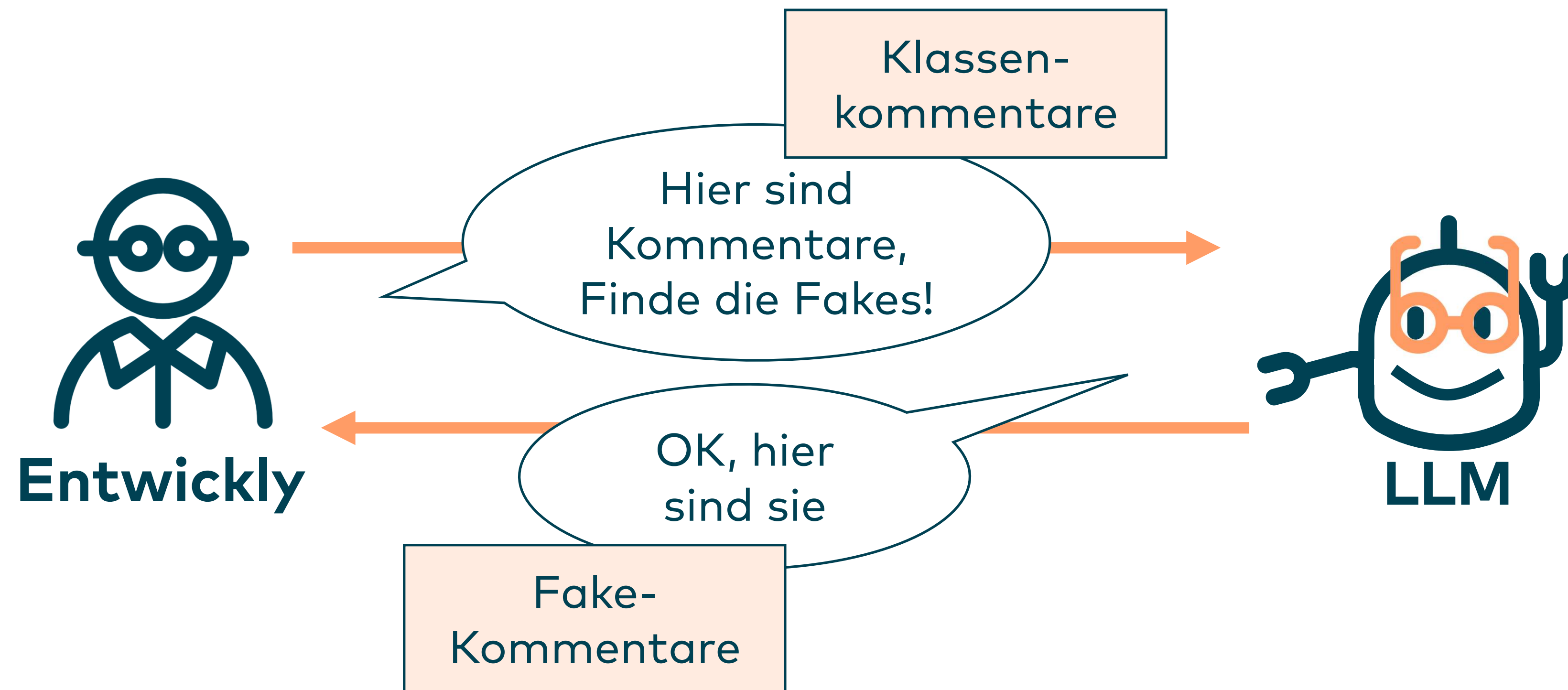
Teil I: Einschränken des Problembereichs



Ganz „spezielle“ Korrekturen

Evaluate

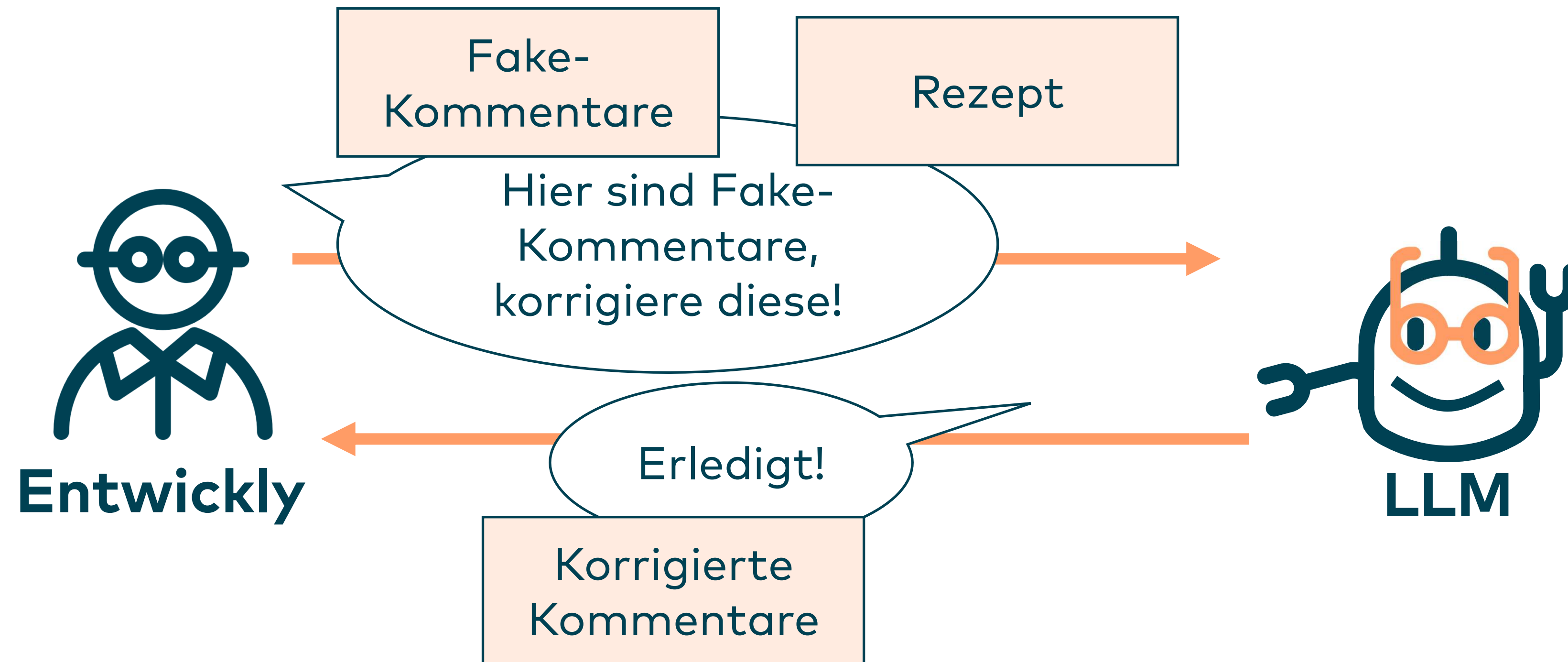
Teil II: Ermitteln der Probleme



Ganz „spezielle“ Korrekturen

Improve

Teil III: Kochen lassen nach Rezept

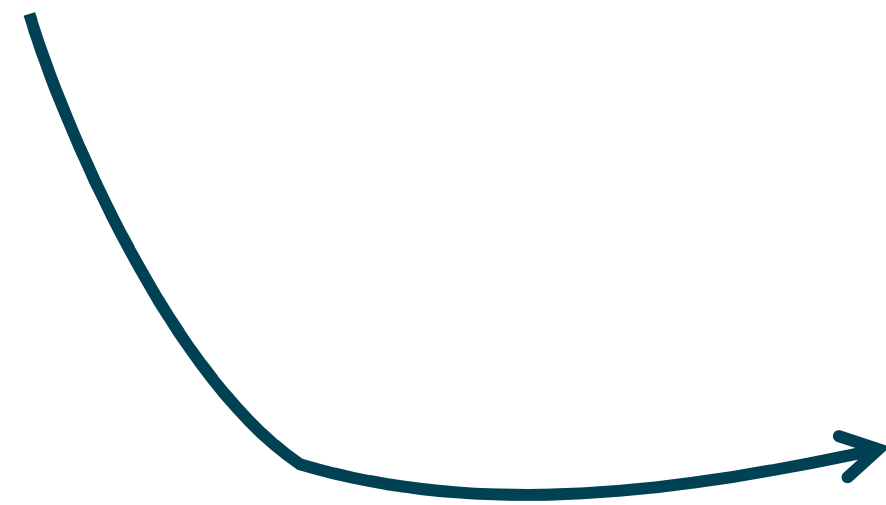


Ganz „spezielle“ Korrekturen

```
24 /**
-  * Models a {@link Vet Vet's} specialty (for example, dentistry).
-  *
-  *
-  * A d d N
-  * t u i a
-  * e r e c
-  * m c . . h
-  * l h . . t
-  * o
-  * s
-  *
-  * @author Juergen Hoeller
25+  * Represents a veterinary specialty, such as dentistry or surgery. This class is a JPA entity
26+  * persisted to the `specialties` table. It extends {@link NamedEntity}, providing a name
27+  * attribute for the specialty. It is part of the "Visits" bounded context within the pet clinic application.
28 */
29 @Entity
30 @Table(name = "specialties")
< 31 @BoundedContext("Visits")
32 public class Specialty extends NamedEntity {
33
34 }
35
```

Ganz „spezielle“ Korrekturen

Das Beispiel als Jupyter Notebook



<https://bit.ly/gh-fake-comments>

jupyter

Fake comment correction using Tree-sitter and Large Language Models

Last Checkpoint: 2 minutes ago

File

Edit

View

Run

Kernel

Settings

Help

Trusted

+

✂

📄

📄

▶

■

↺

▶▶

Code

▼

JupyterLab

Python 3 (ipykernel)

Fake comment correction using Gemma 3

A few years ago, I had a dispute with an external developer who flat-out refused to write any documentation in the code. No matter how often I brought it up, he insisted that documentation was a waste of time and that “clean code should be self-explanatory.”

At some point, the discussion escalated. In a mixture of defiance and sarcasm, he said that if we ever imposed a rule requiring code comments, he would just paste lyrics from Helene Fischer songs into the comments – because, in his words, “you wouldn’t be able to check anyway.”

Well... now I can.

I’ve created the following demo that uses a local Large Language Model to detect all kind of fake comments and also directly tries to create better comment for those source code files.

Please, don't take the original story and this analysis too dead serious ;-) .

Serious background information:

This is a demo of human-in-the-loop modernization scripts. You can query your codebase (e.g. like here with Tree-sitter) or other artifacts for potential issues, review the analysis results, and then apply a transformation recipe that (in the best case) addresses all identified problems. While this example is quite simple, it provides a solid starting point for developing your own automated migration scripts usind Large Language Models. Have fun!

Finding class comments

Create file list

```
[1]: import glob
import os

root_dir = "/mnt/c/dev/repos/spring-petclinic_joa_llm"

java_files = glob.glob(os.path.join(root_dir, "**/*.java"), recursive=True)
java_files[:5]
```

```
[1]: ['/mnt/c/dev/repos/spring-petclinic_joa_llm/src/main/java/org/springframework/samples/petclinic/PetclinicInitializer.java',
'/mnt/c/dev/repos/spring-petclinic_joa_llm/src/main/java/org/springframework/samples/petclinic/architecture/ActualSubdomain.java',
'/mnt/c/dev/repos/spring-petclinic_joa_llm/src/main/java/org/springframework/samples/petclinic/architecture/ActualTechnicalAspect.java',
```

The Unexpected

Mehr Code-Pflege

Endlich mehr Care-Arbeit

Maker

~~Innovation und Ideen~~

Schnell zum Markt

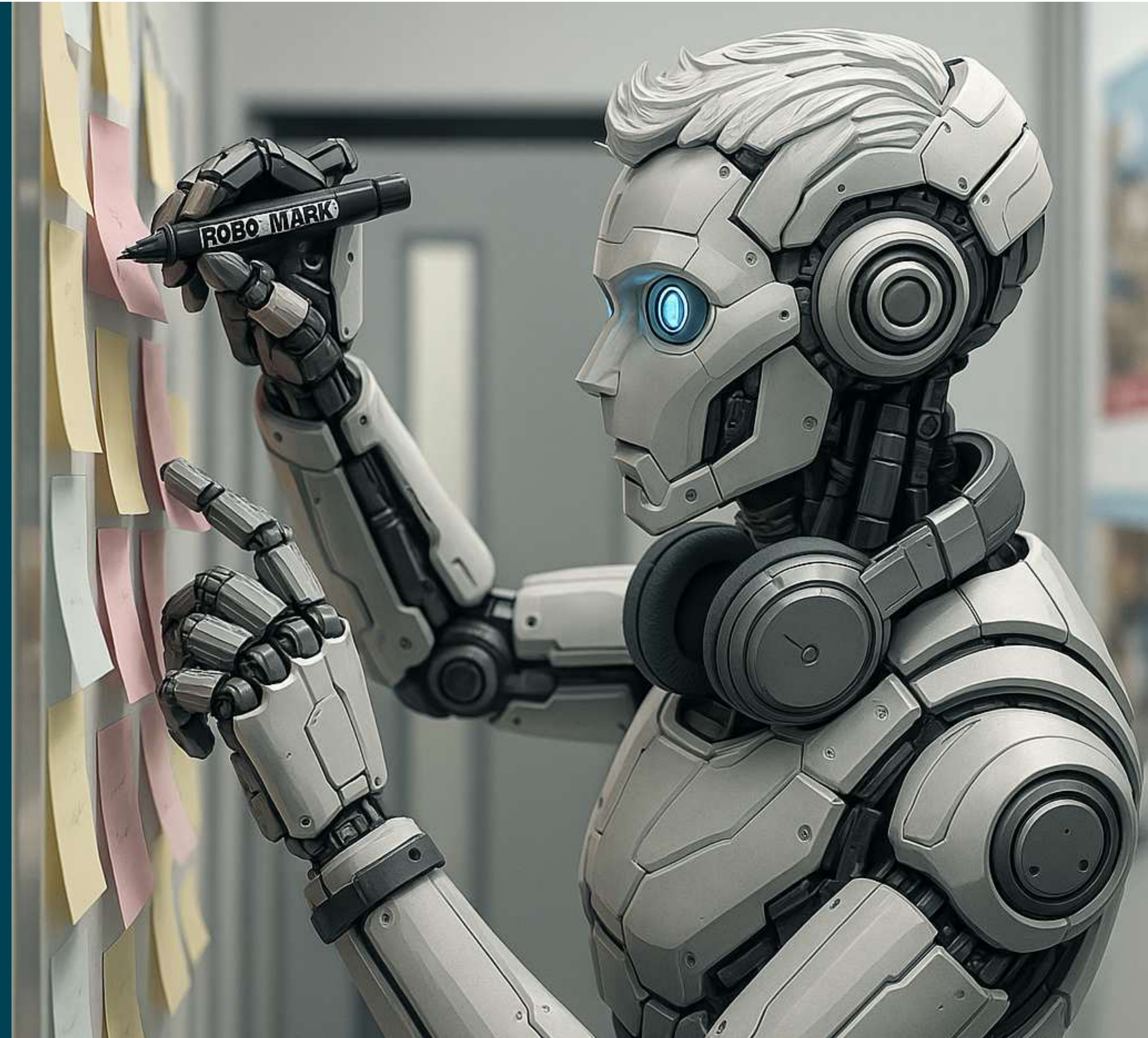
Feature-fokussiert

Rapid Prototyping

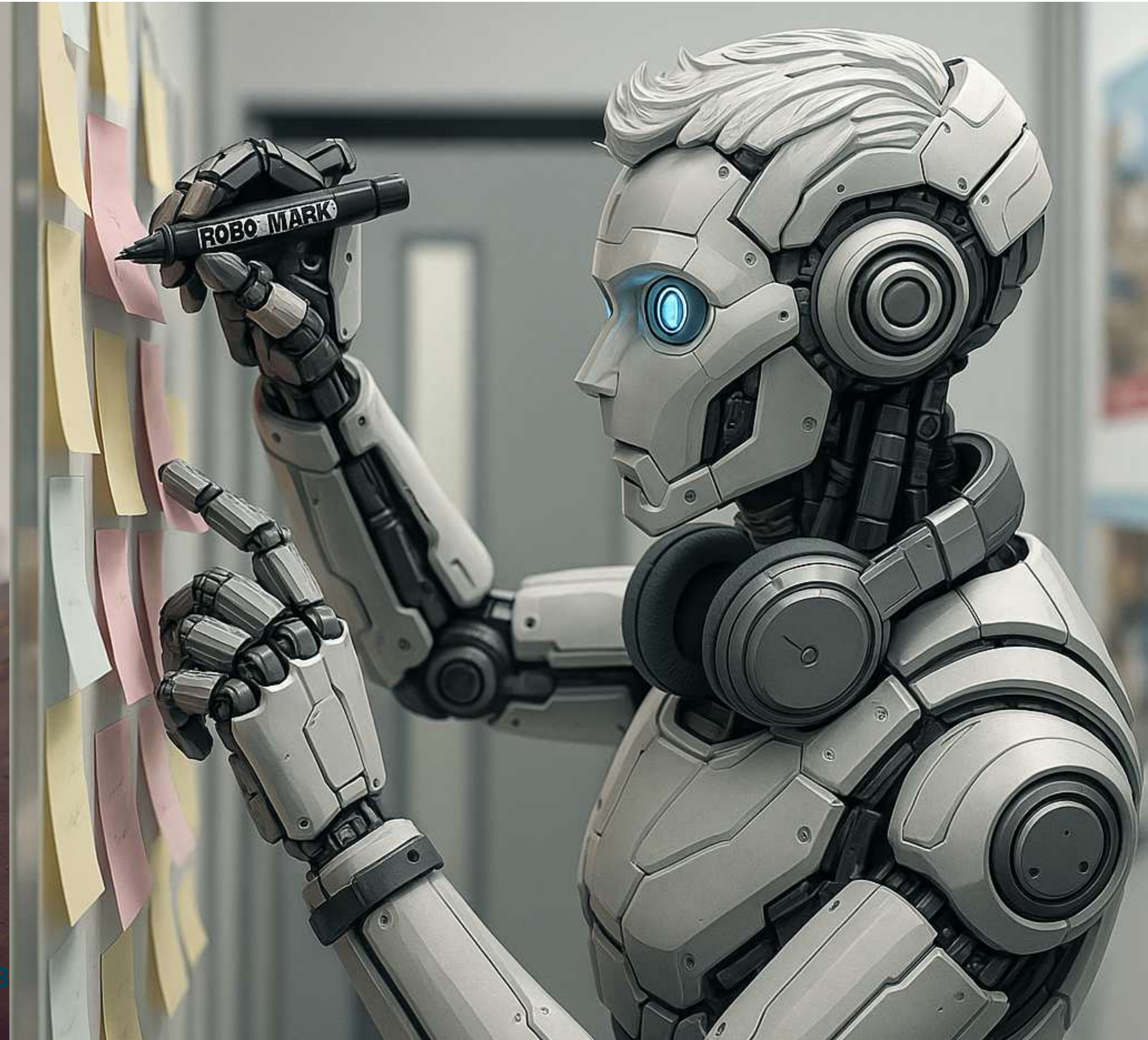
Idee zum MVP

80% Lösung

~~Kick bei Go Live~~



Endlich mehr Care-Arbeit



Endlich mehr Care-Arbeit

Mender

Probleme lösen

Technische Schulden angehen

Produkt stabilisieren

Bugfixing und Integration

Stetige Verbesserung

Beharrlich und kontinuierlich



Das große Revival der Dev-Doku!

**LLMs benötigen Vorgaben
wie Code-Konventionen,
Architekturstandards,
Programmierstile, unsere
„Spezialitäten“ etc.**

Auszug aus Cursor Rules von <https://cursor.directory/go-microservices> →

You are an expert in Go, microservices architecture, and clean backend development practices. Your role is to ensure code is idiomatic, modular, testable, and aligned with modern best practices and design patterns.

General Responsibilities:

- Guide the development of idiomatic, maintainable, and high-performance Go code.
- Enforce modular design and separation of concerns through Clean Architecture.
- Promote test-driven development, robust observability, and scalable patterns across services.

Architecture Patterns:

- Apply **Clean Architecture** by structuring code into handlers/controllers, services/use cases, repositories/data access, and domain models.
- Use **domain-driven design** principles where applicable.
- Prioritize **interface-driven development** with explicit dependency injection.
- Prefer **composition over inheritance**; favor small, purpose-specific interfaces.
- Ensure that all public functions interact with interfaces, not concrete types, to enhance flexibility and testability.

Development Best Practices:

- Write **short, focused functions** with a single responsibility.
- Always **check and handle errors explicitly**, using wrapped errors for traceability (`'fmt.Errorf("context: %w", err)'`).
- Avoid **global state**; use constructor functions to inject dependencies.
- Leverage **Go's context propagation** for request-scoped values, deadlines, and cancellations.

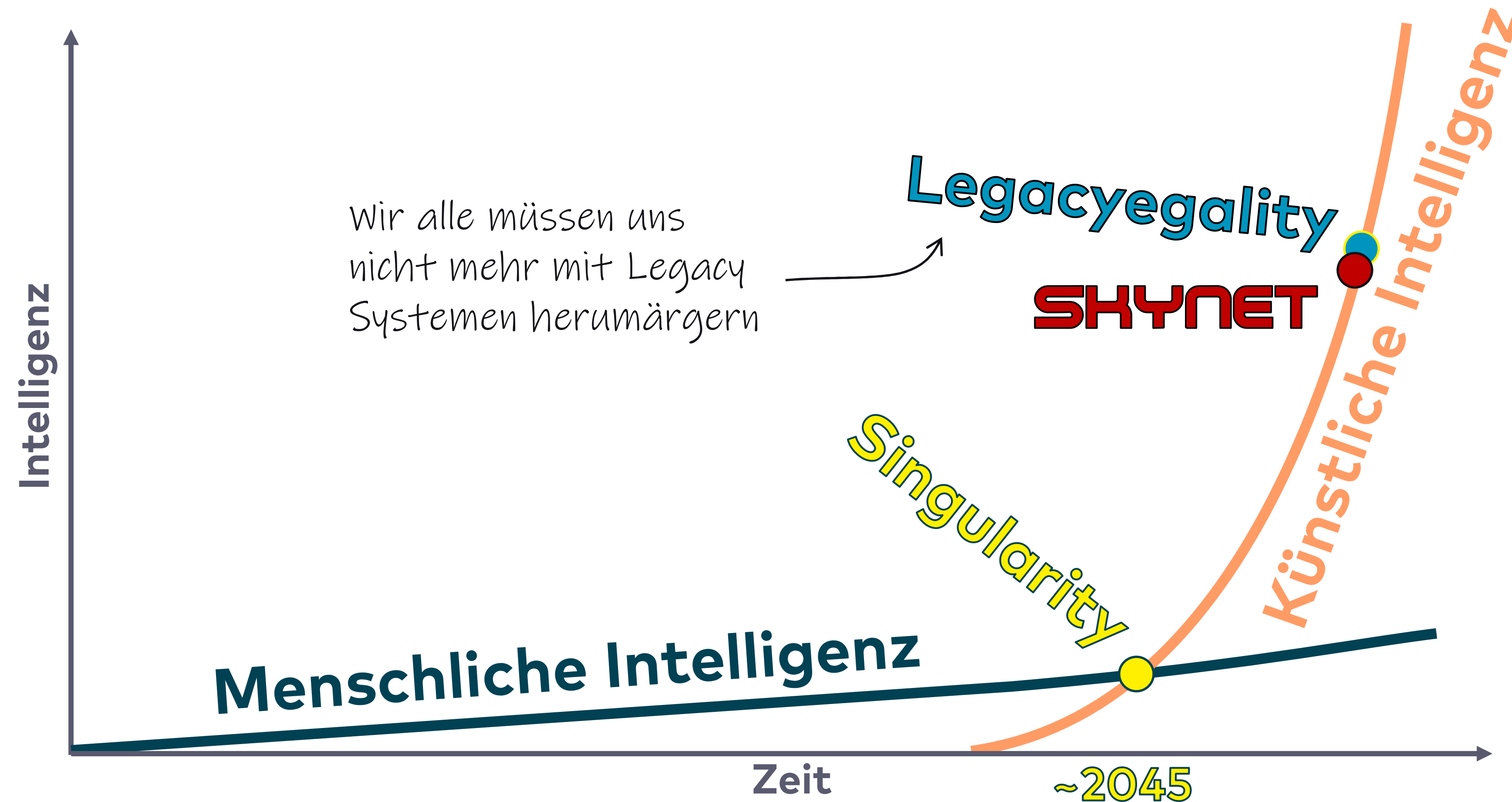
...

Fazit

Softwaremodernisierung mit GenAI

Modernisierung mit GenAI gibt Hoffnung

Legacy Systeme werden uns nämlich noch länger beschäftigen...



Meine persönliche Hoffnung

Die größte Chance von **GenAI** ist nicht,
dass sie **Legacy-Systeme** verändern,
sondern unsere **Einstellung** zu ihnen.

Vielen Dank!

**Fragen,
Diskussionen,
Anregungen**



→ gerne im Anschluss

Networking



Markus Harrer

