

# QUIZ-WARS

## WBA2KrennSteffens

Meilenstein 3 - Resumeé, Überblick Dienstgeber,  
Überblick Dienstnehmer, Deployment, Vorführung

Technology  
Arts Sciences  
TH Köln

# Resumeé

## Umgesetzte Inhalte

- Konzeption eines verteilten Systems samt Anwendungslogik
- Implementierung Dienstgebers
- Implementierung Dienstnehmers inkl. Präsentationslogik (Abb.1)
- Dokumentation aller Artefakte im Github Wiki

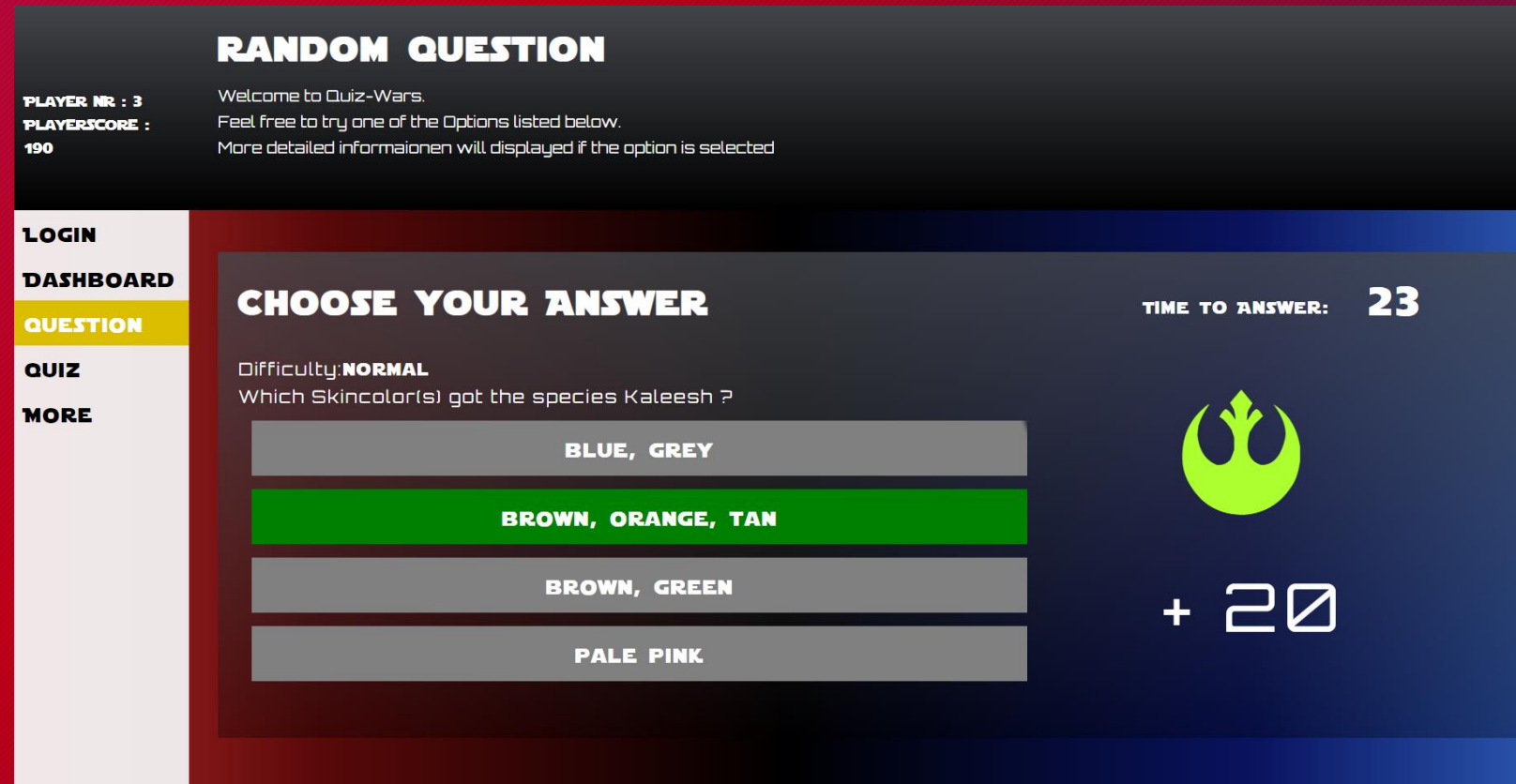


Abbildung 1. Frage-View Dienstnehmer

# Dienstgeber

Allgemein, Ressourcen, Zusammenfassung



# Dienstgeber - Allgemein

## Funktionalität

- JSON-Daten werden über API bezogen und verarbeitet
- **Anwendungslogik:** Daten werden mittels ‚Frage-Template‘ dynamisch bereitgestellt (Abb.2)
- Quizfragen und gesamte Quizze können abgerufen werden
- Enthält persistente Datenerhaltung des Systems
- Bereitgestellte Daten können in Kategorien unterschieden werden

```
1 [
2     "Which eyecolor got C-3PO ?",
3     [
4         "brown",
5         "blue",
6         "yellow",
7         "orange"
8     ],
9     2,
10    "easy"
11 ]
```



Abbildung 2 | Dynamisch erstellte Frage

# Dienstgeber - Ressourcen

## /question

- Abrufen der generierten Fragen
- Query-Parameter gibt Aufschluss auf Kategorie
- CRUD-Operationen der Communityinhalte

## /quiz

- Abrufen von Quizen bestehend aus der API-Data
- Abrufen von Quizen bestehend aus der Community-Data

## /users

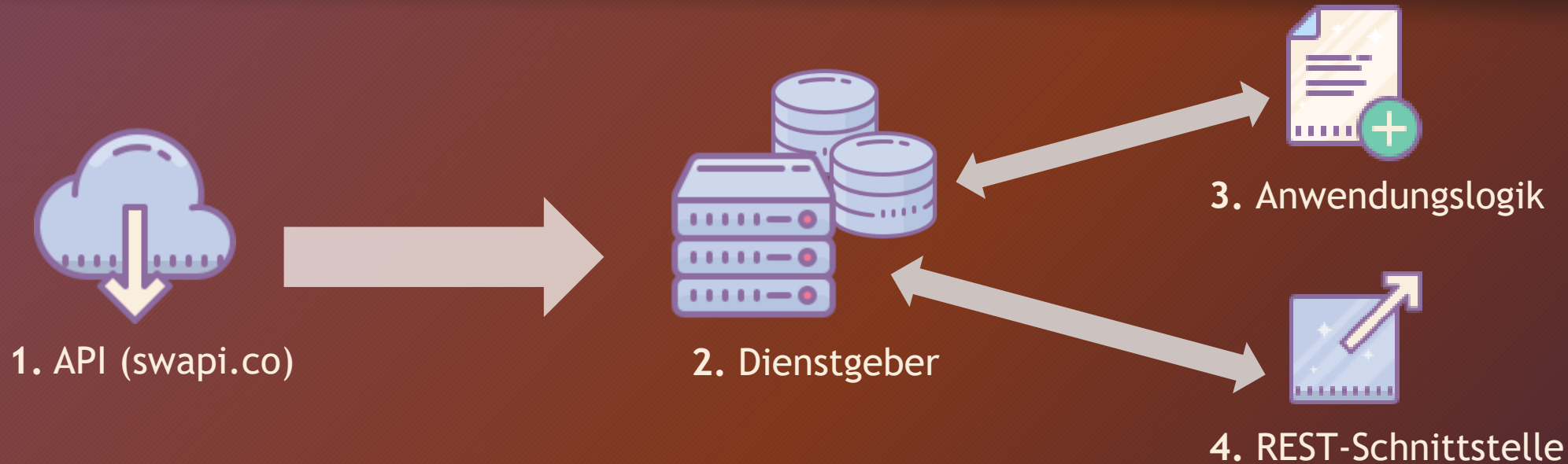
- Schreib und Leseoperationen der Userdaten

## /topics

- Übergeben aller Kategorien



# Dienstgeber - Zusammenfassung



**1** Daten werden von der API bezogen

**2** Dienstgeber stellt Fragen und Quizze, sowie Community-Inhalte bereit.

**3** Community-Inhalte werden persistent in der Datenerhaltung gespeichert

**4** Inhalte werden über die REST-Schnittstelle zur Verfügung gestellt

# Dienstnehmer

Allgemein, Ressourcen, Zusammenfassung



# Dienstnehmer - Allgemein

## Funktionalität

- Interaktion mit Dienstgeber
- Fragen und Quizze
- Fragen enthalten einen Schwierigkeitsgrad und Countdown-Timer
- Bietet GUI für unerfahrene Nutzer
- **Anwendungslogik:**
  - Verwalten der User und Userdaten.
  - Schwierigkeit der Frage beeinflusst Punktzahl, sowie Fragezeit

## CHOOSE YOUR ANSWER

Difficulty: **HARD**

Which average Lifespan got the species Yoda's species ?

**900**

**50**

**100**

**500**



Gerhard Janson <https://pixabay.com/de/photos/figuren-star-wars-freigestellt-2775971/>



# Dienstnehmer - Ressourcen

## /question

- Abrufen der generierten Fragen
- Query-Parameter verweist auf Kategorie
- CRUD-Operationen der Communityinhalte

## /quiz

- Abrufen von Quizen bestehend aus API-Data
- Abrufen von Quizen bestehend aus Community-Data

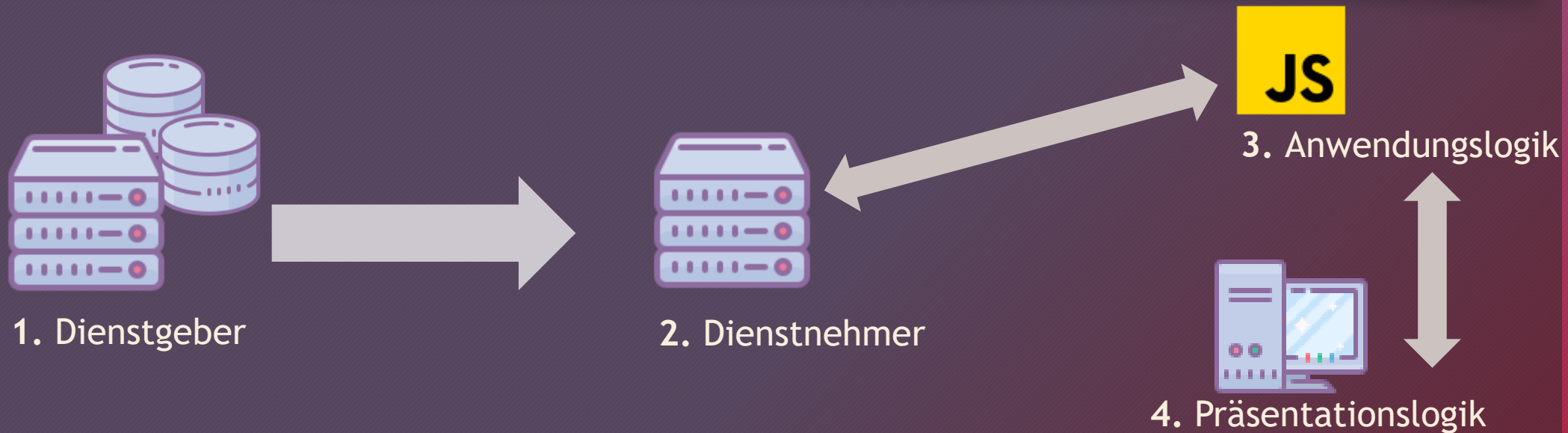
## /users

- Schreib und Leseoperationen der Userdaten

## /more

- Übergeben aller Kategorien

# Dienstnehmer - Zusammenfassung



**1** Dienstgeber stellt Daten bereit.

**2** Dienstnehmer verarbeitet bezogene Daten.

**3** Inhalte werden mittels EJS dargestellt

**4** Inhalte werden durch den Browser zur Verfügung gestellt



# Fazit

Prozessdokumentation, Live-Demonstration

# Fazit - Prozessdokumentation

## Erkenntnisse

- Sammeln von weitreichenden Erkenntnissen in den Bereichen
  - Verteilte Systeme
  - Javascript / Embedded Javascript / NodeJS / JSON

## Zeitmangel

- Viele Funktionen noch denkbar

## Ausblick

- Erfahrungen und Know-How für kommende Projekte



# Vorführung

Technology  
Arts Sciences  
TH Köln

**QUIZ-WARS**  
**MAY THE KNOWLEDGE BE WITH YOU**

- Demonstration des Systems
- Klären der offenen Fragen

