OCR-D Implementierungsprojekt

Integration von Kitodo und OCR-D zur produktiven Massendigitalisierung (2021–2023)

OCR On-Demand für DFG-Viewer und Kitodo.Presentation

Stefan Weil & Christos Sidiropoulos (Universitätsbibliothek Mannheim)

26. Mai 2023

Gefördert durch
Deutsche Forschungsgemeinschaft

Einführung

dfg-viewer.de – Referenzimplementierung auf der Grundlage von Kitodo.Presentation

UB UNIVERSITÄTSBIBLIOTHEK MANNHEIM

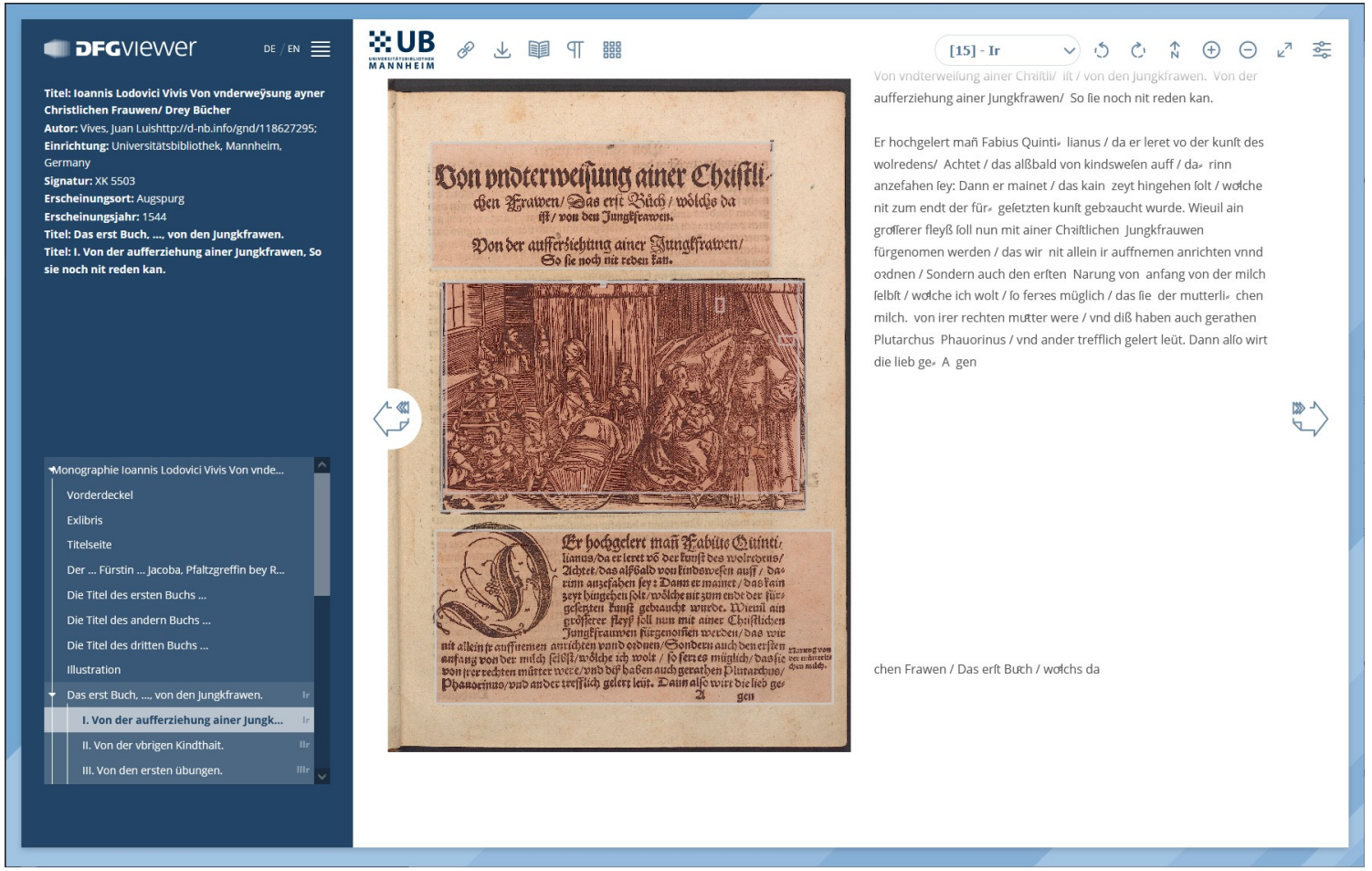
Homepage DFG Viewer

- Entwicklung und Betrieb durch die SLUB Dresden
- Förderung durch die Deutsche Forschungs-gemeinschaft (DFG)

DFG-Viewer: Beispiel mit Volltext

Der DFG-Viewer

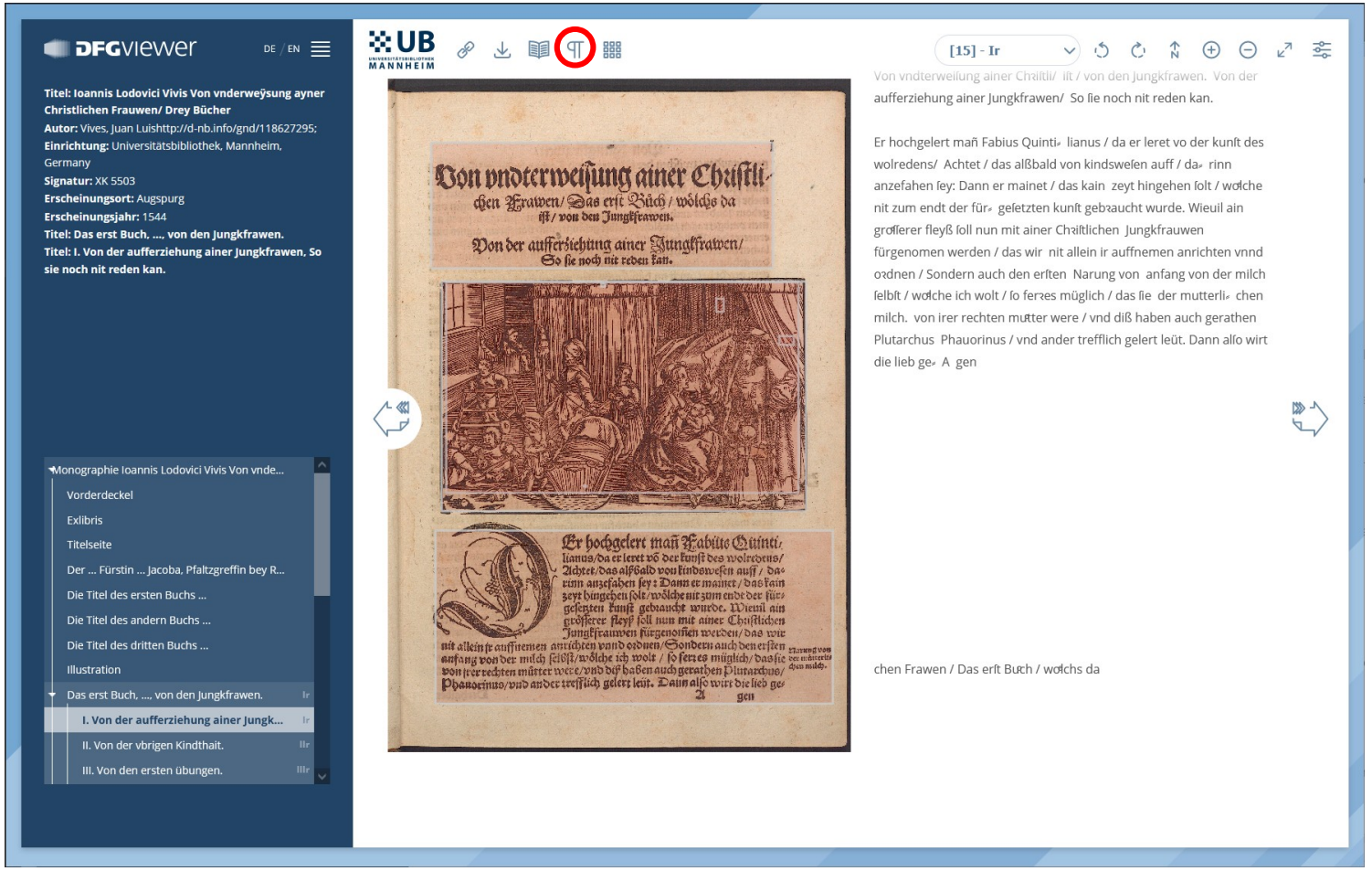
- … ist in der Lage, die (meisten) Digitalisate aus Archiven und Bibliotheken darzustellen.



DFG-Viewer: Beispiel mit Volltext

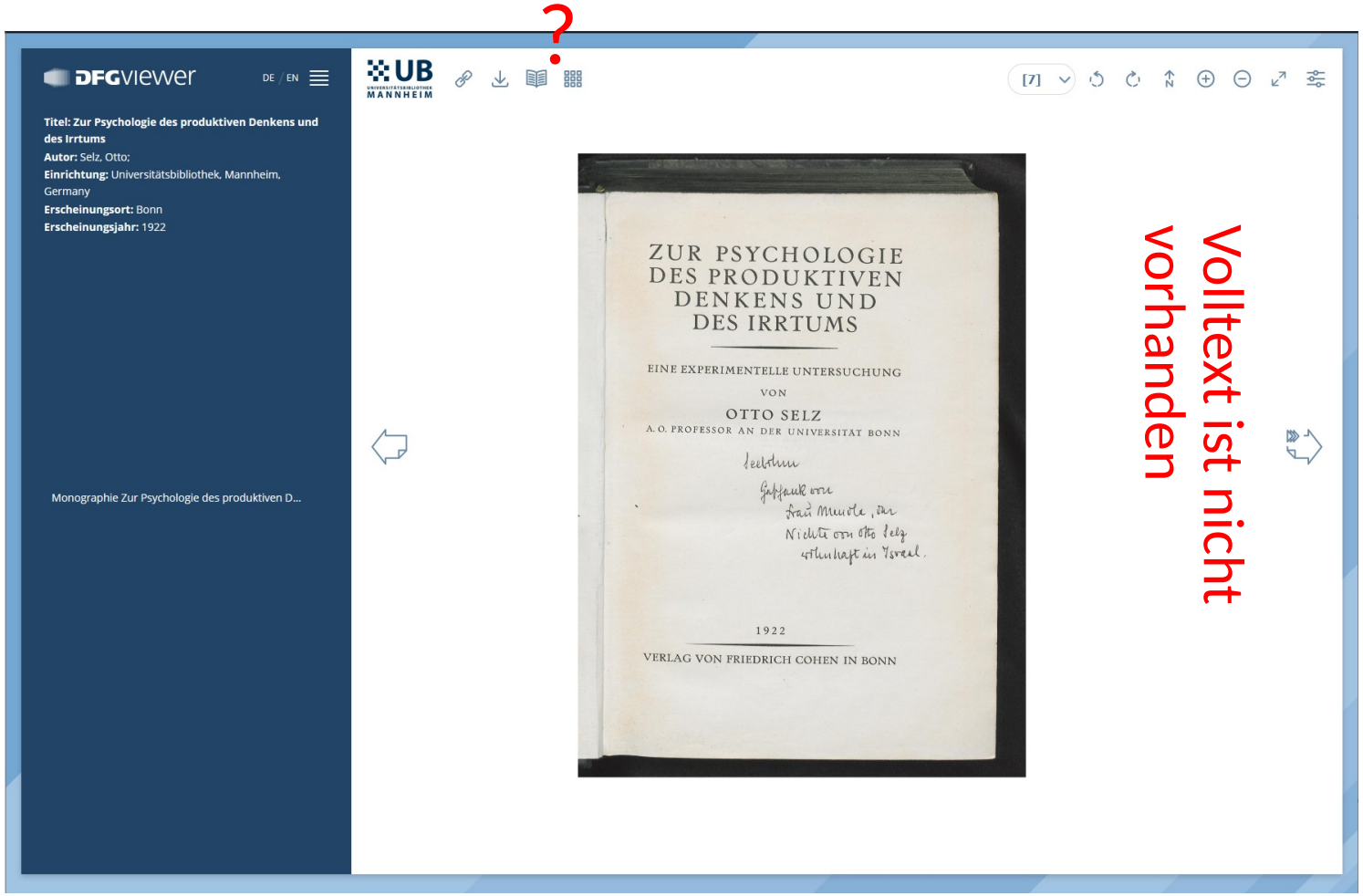
Der DFG-Viewer

- … ist in der Lage, die (meisten) Digitalisate aus Archiven und Bibliotheken darzustellen.
- Dabei können optional auch die vorhandenen Volltexte angezeigt werden (im Bild rot markiert).



DFG-Viewer: Beispiel ohne Volltext

- Wenn keine Volltexte verfügbar sind, wird das entsprechende Symbol nicht angezeigt.



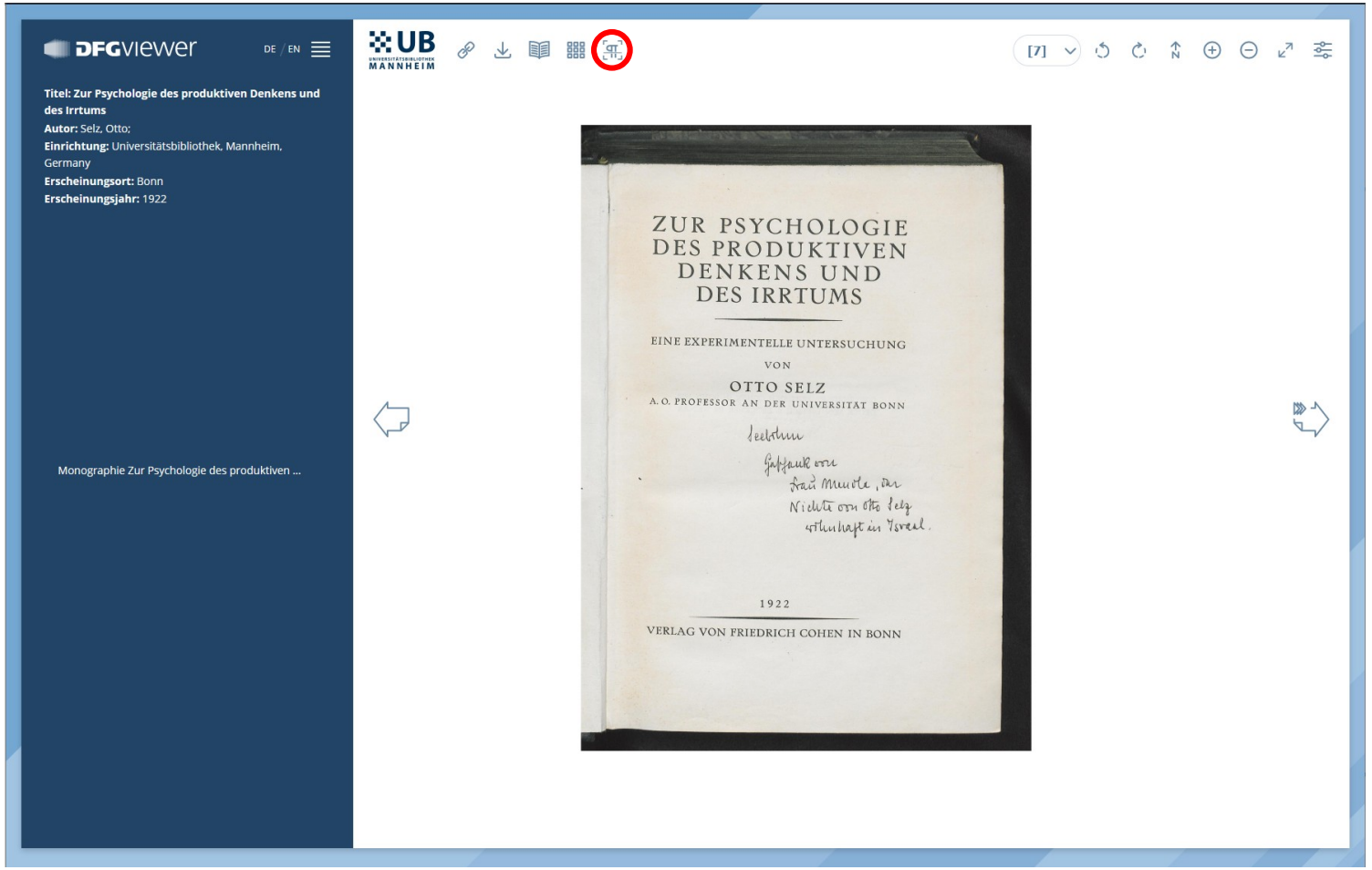
OCR On-Demand

DFG-Viewer: OCR On-Demand – Idee

- Idee: Nutzende können OCR für Werke ohne bzw. mit unbrauchbaren Volltexten anfordern.
- Der Volltext der aktuellen Seite steht nach wenigen Sekunden zur Verfügung, der Volltext des gesamten Werkes nach kurzer Zeit (abhängig vom Umfang).
- Dazu können verschiedene Optionen zur Volltextgenerierung angeboten werden.

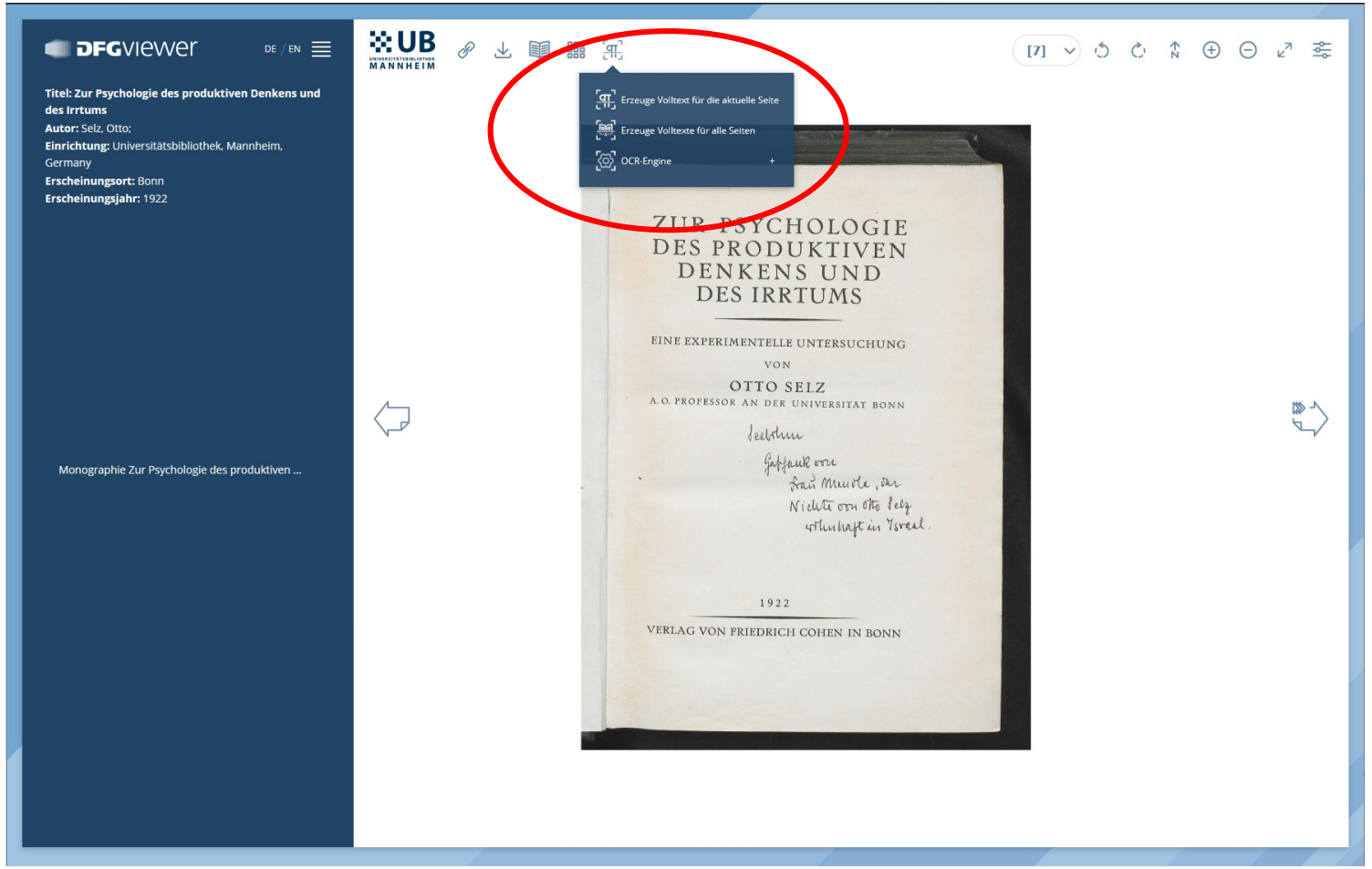
DFG-Viewer mit OCR On-Demand: Anwendung

- Wenn keine Volltexte verfügbar sind, können diese spontan erzeugt werden.
- Dazu wird ein neues Bedienelement eingeblendet (im Bild rot markiert).



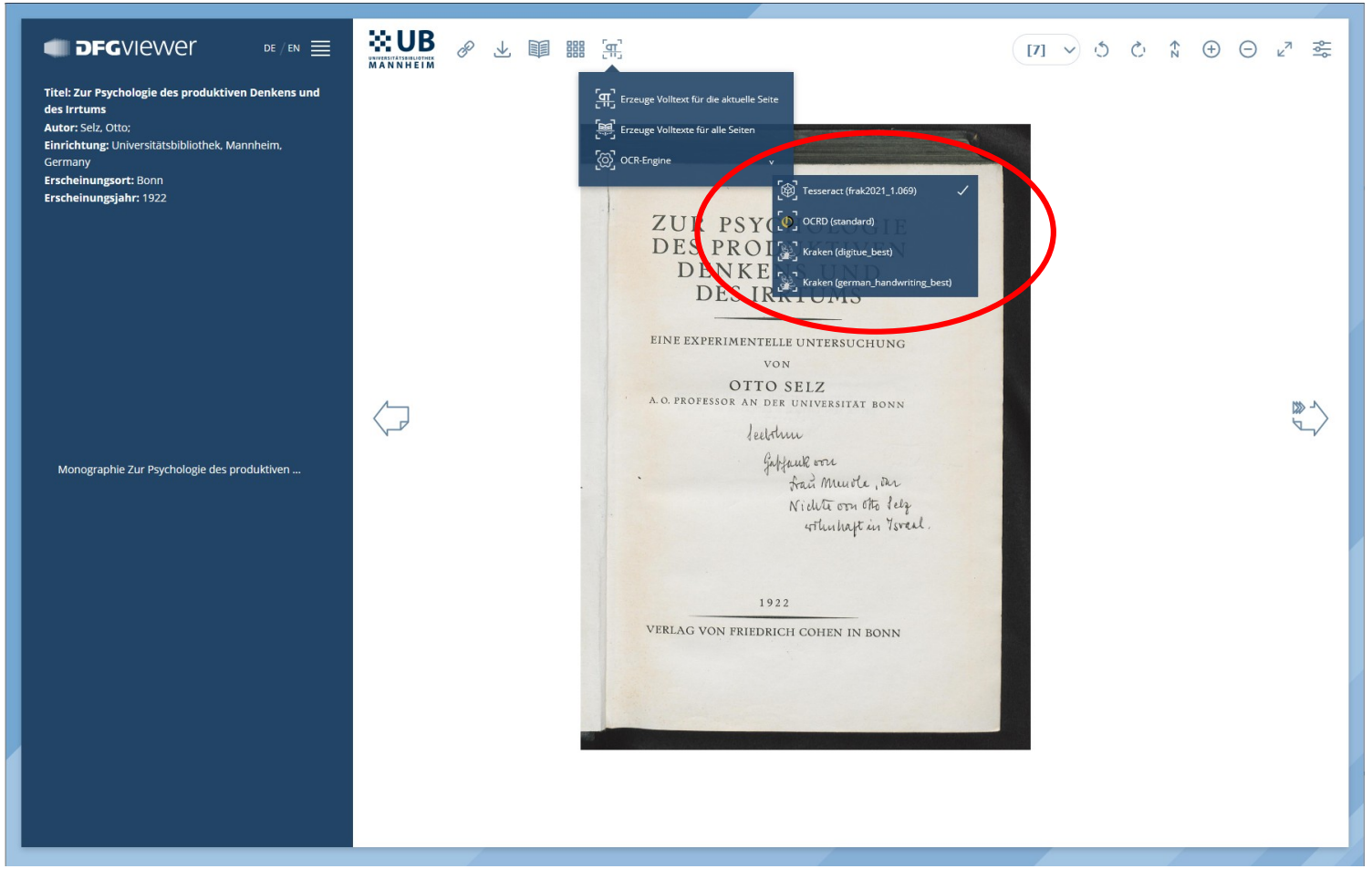
DFG-Viewer mit OCR On-Demand: Anwendung

- Das neue Bedienelement ermöglicht die Volltext-erstellung für die jeweilige Seite oder das gesamte Werk.



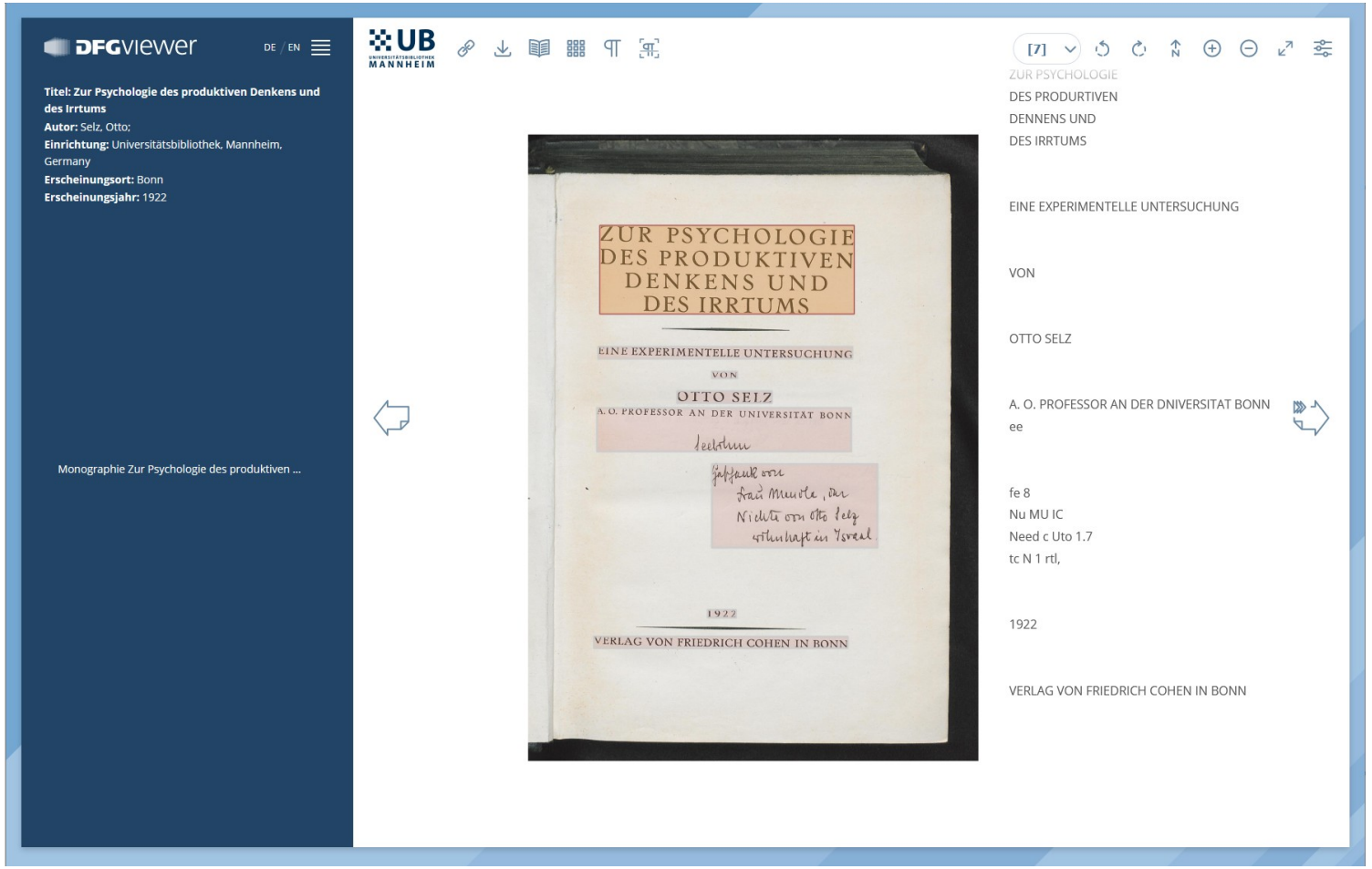
DFG-Viewer mit OCR On-Demand: Anwendung

- Das neue Bedienelement ermöglicht die Volltext-erstellung für die jeweilige Seite oder das gesamte Werk.
- Optional können verschiedene Techniken zur Volltexterstellung ausgewählt werden.



DFG-Viewer mit OCR On-Demand: Anwendung

- Nach erfolgreicher Volltexterzeugung wird dieser angezeigt.
- Dauer (abhängig von der ausgewählten OCR Technik und dem Seitenaufbau):
  - 1 s bis 14 s pro Seite
  - 39 min für 694 Seiten (parallelisierbar)



Praktische Umsetzung

OCR-Engines

OCR-Engine = Technik zur Erzeugung von Volltexten

Ziel:

- Die Auswahl der angebotenen OCR-Engines obliegt der Einrichtung.
- Das Hinzufügen neuer OCR-Engines soll ohne großen Aufwand oder Eingriffe in den Programmcode möglich sein.

OCR-Engines: Umsetzung

- Alle benötigten Dateien befinden sich in der Verzeichnisstruktur von Kitodo.Presentation.
- Hier befinden sich die **OCR-Engines**, bei denen es sich um einfache **Shell-Skripte** handelt, und eine **JSON-Datei**, in der festgelegt werden kann, welche der OCR-Engines aktiv sein sollen sowie weitere Informationen zu den OCR-Engines.

OCR-Engines: Umsetzung

- Alle benötigten Dateien befinden sich in der Verzeichnisstruktur von Kitodo.Presentation.
- Hier befinden sich die **OCR-Engines**, bei denen es sich um einfache **Shell-Skripte** handelt, und eine **JSON-Datei**, in der festgelegt werden kann, welche der OCR-Engines aktiv sein sollen sowie weitere Informationen zu den OCR-Engines.
- Beispiel:

```
# ls -l
FullTextGenerationScripts
kraken-basic.sh
kraken-digitue_best.sh
kraken-
german_handwriting_best.sh
ocrd-basic.sh
ocrEngines.json
tesseract-basic.sh
```

Shell-Skripte = OCR-Engines

JSON-Datei mit Auflistung der aktiven OCR-Engines

OCR-Engines: Beispiel

```
{
 "ocrEngines": [
    {
        "name": "Tesseract",
        "de": "Tesseract (frak2021_1.069)",
        "en": "Tesseract (frak2021_1.069)",
        "class": "tesseract",
        "data": "tesseract-basic"
    },
    {
        "name": "Kraken (digitue_best)",
        "de": "Kraken (digitue_best)",
        "en": "Kraken (digitue_best)",
        "class": "kraken",
        "data": "kraken-digitue_best"
    },
    {
        "name": "Kraken
(german_handwriting_best)",
        "de": "Kraken
(german_handwriting_best)",
        "en": "Kraken
(german_handwriting_best)",
        "class": "kraken",
        "data": "kraken-
german_handwriting_best"
    }
 ]
}
```

OCR-Engines: Beispiel

```
{
 "ocrEngines": [
    {
        "name": "Tesseract",
        "de": "Tesseract (frak2021_1.069)",
        "en": "Tesseract (frak2021_1.069)",
        "class": "tesseract",
        "data": "tesseract-basic"
    },
    {
        "name": "Kraken (digitue_best)",
        "de": "Kraken (digitue_best)",
        "en": "Kraken (digitue_best)",
        "class": "kraken",
        "data": "kraken-digitue_best"
    },
    {
        "name": "Kraken
(german_handwriting_best)",
        "de": "Kraken
(german_handwriting_best)",
        "en": "Kraken
(german_handwriting_best)",
        "class": "kraken",
        "data": "kraken-
german_handwriting_best"
    }
 ]
}
```

Eine der aktiven OCR-Engines mit zusätzlichen Informationen

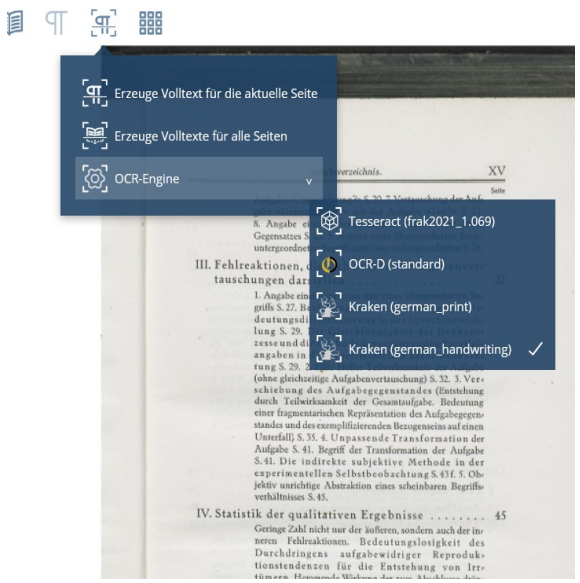
Name und optionale Übersetzungen der Engine

Bezeichnung des Shell-Skriptes

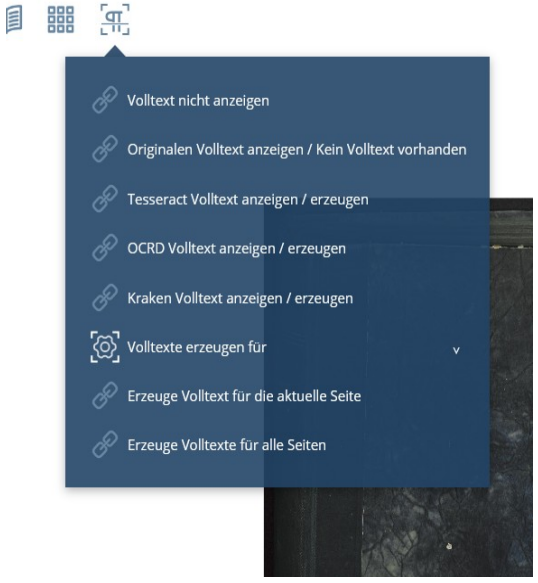
User Interface

User Interface: Variationen

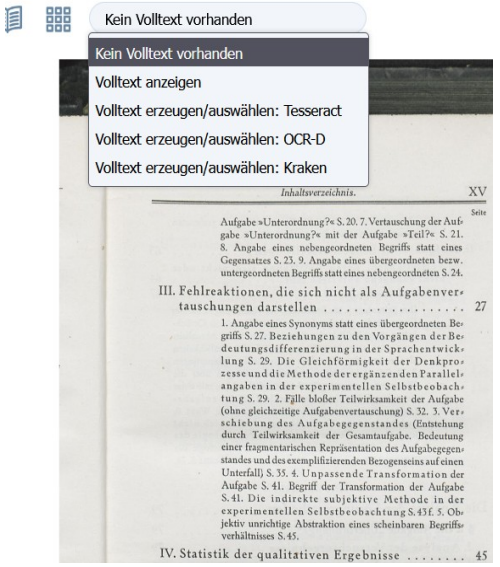
Aktueller Stand



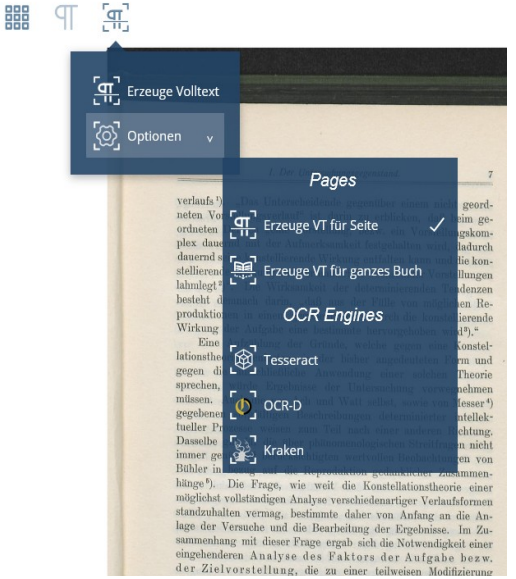
Option 1



Option 2



Option 3



Überblick

Laufende Arbeiten & Ausblick

- Restrukturierung der Oberfläche
- Indizierung verarbeiteter Werke:
  - Bereitstellung der Werke mit neu erzeugten Volltexten in „Sammlungen“ nach den besitzenden Einrichtungen
  - Suche in den Metadaten der Werke mit neu erzeugten Volltexten
  - Suche in den Volltexten der Werke mit neu erzeugten Volltexten
- OAI-Schnittstelle für Abruf neuer OCR-Ergebnisse durch besitzende Einrichtungen

Wunschliste

- Maschinelle Übersetzung der Volltexte
- Text to speech – Audioausgabe der Volltexte (Barrierefreiheit)
- Feedback-Möglichkeit zu den OCR-Resultaten
- Korrigierbare Volltexte (auch für Nachtraining verwendbar)

Weitere Informationen

- Prototyp:
  → *https://dfg-viewer.bib.uni-mannheim.de/*
- Installationsanleitung:
  → *https://github.com/UB-Mannheim/kitodo-presentation/wiki*
- Docker Konfiguration:
  → *https://github.com/UB-Mannheim/kitodo-presentation-docker*
- DFG-Viewer mit OCR-On-Demand:
  → *https://github.com/UB-Mannheim/dfg-viewer/tree/6.x-ocr*
- Kitodo.Presentation mit OCR-On-Demand:
  → *https://github.com/UB-Mannheim/kitodo-presentation/tree/4.x-ocr*
- Projektplan:
  → *https://github.com/orgs/UB-Mannheim/projects/2*

Vielen Dank!

Stefan Weil stefan.weil@uni-mannheim.de
Christos Sidiropoulos christos.sidiropoulos@uni-mannheim.de