



Nachhaltigkeit vs. Digitale Edition

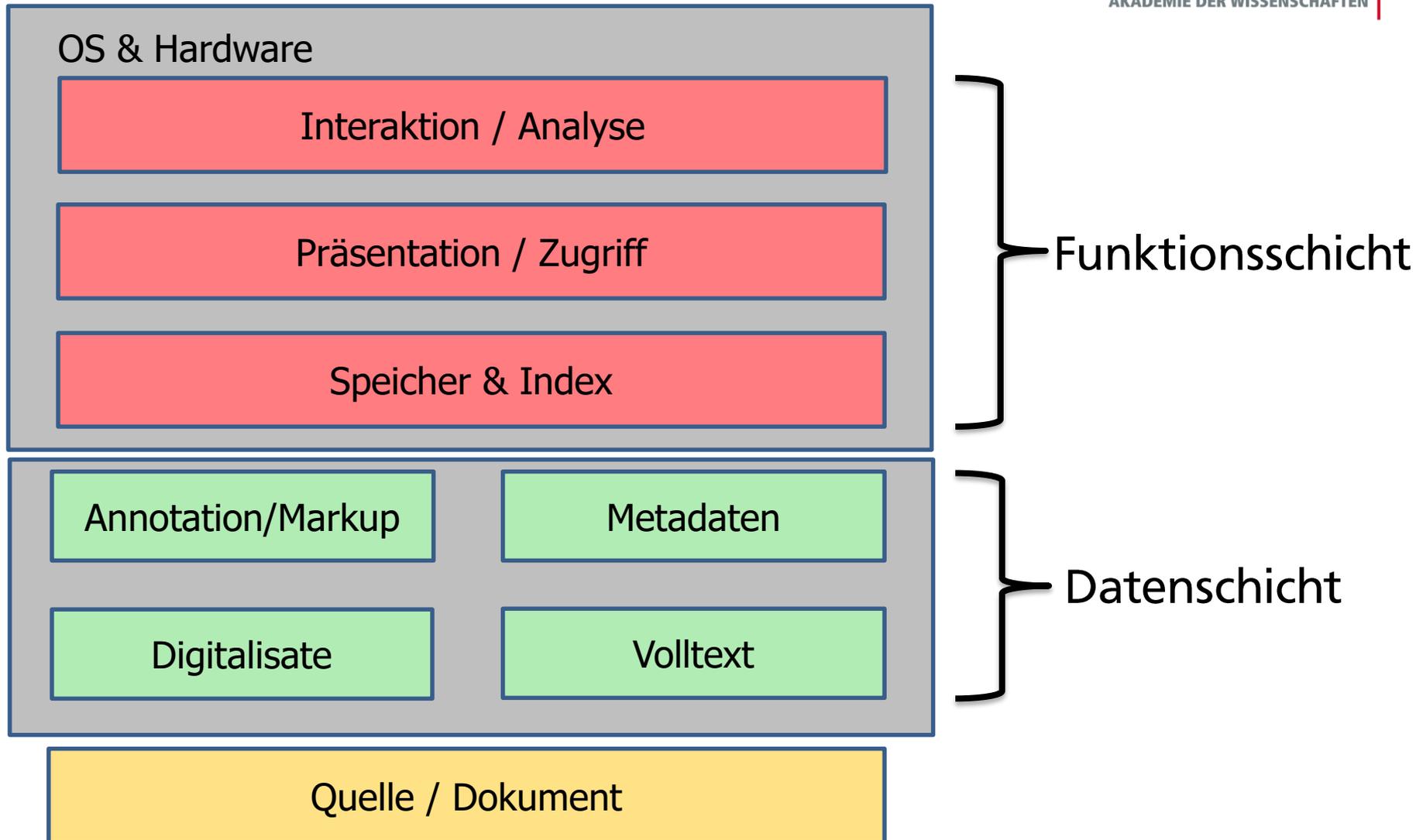
Möglichkeiten und Grenzen

Alexander Czmiel
czmiel@bbaw.de
@alexczmiel

Berlin-Brandenburgische Akademie der Wissenschaften
TELOTA



Digitale Editionen Layer Model





Digitale Editionen

=

Forschungssoftware

- Research Software Engineer-Community
- Software Sustainability Institute
- AG DH-RSE des DHd-Verbands
- DFG
- Helmholtz-Gemeinschaft
- DLR
- Ad-hoc-Arbeitsgruppe der Allianz der deutschen Wissenschaftsorganisationen

- Richtlinien, Zitation von Software, Software Carpentry, Nachhaltigkeit, Ausbildung von RSE, akademische Wahrnehmung uvm.

- Unterschiede zur klassischen Softwareentwicklung
 - Entwicklungsinteresse → Wissenschaftliche Inhalte möglichst adäquat zugänglich machen (Lesbar und bearbeitbar)
 - Zielgruppe
 - Rahmenbedingungen (Finanzierung)
 - Nutzungsanalyse

Dinge für die Ewigkeit...

...können aber nach Projektende keinen Support mehr leisten

- Digitale Editionen so entwickeln, dass sie mit geringen weiteren Investition von Ressourcen möglichst lange ihren Nutzen erfüllen.

Forschungsergebnisse

Open Access



Digitale
Editionen

Forschungssoftware

Open Source

Forschungsdaten

Open Data

→ alles nach den FAIR-Prinzipien
(Findable, Accessible/Adressable, Interoperable, Reusable)

- Technical preservation (techno-centric)
- Emulation (data-centric)
- Migration (functionality-centric)
- Cultivation (process-centric)
- Hibernation (knowledge-centric)
- Deprecation
- Procrastination



Software
Sustainability
Institute

<https://www.software.ac.uk/resources/approaches-software-sustainability>

- nicht alles kann bewahrt und kuratiert werden
- Visualisierungen, Rendering projektspezifischer Annotationen
- Verknüpfungen, verteilte Ressourcen
- Spezialsoftware
- Randtechnologien
- Virtual Reality, Augmented Reality
- proprietäre Datenformate
- Laufzeitumgebungen
- Betriebssysteme
- Hardware

- aktuelle Entwicklungen aktiv mitgestalten
 - mehr Austausch zwischen DH-Entwickler und RSE-Community
- planen, Synergien schaffen, bewährte und unterstützte Tools einsetzen
- Open Source Entwicklung unterstützen
 - Besonders bei Tools, auf denen unseren Anwendungen basieren

1. (Formalisierte) Dokumentation
2. Reduktion von Komplexität
3. DH-RSEs ausbilden & beschäftigen
4. Gut sortierter Werkzeugkasten
5. Technologien, Use-Cases & Projekte clustern
6. good coding practices
7. Paketierung von Modulen
8. Kurationsinfrastrukturen
9. Testen, testen, testen
- 10....und dokumentieren!



Better Software Better Research

(The Software Sustainability Institute - <https://www.software.ac.uk>)