WWU
MÜNSTER

A SOLID Distributed Architecture for Sciebo Research Data Services

CS3 2020 Workshop,
Copenhagen, DK

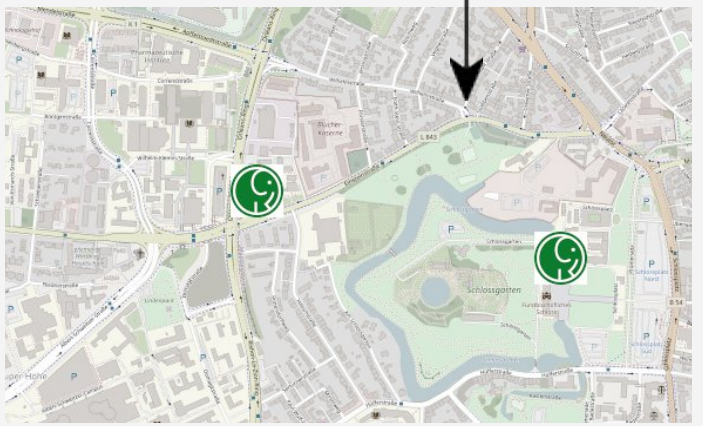
Peter Heiss,
Jens Stegmann,
Holger Angenent

wissen.leben

Westfälische Wilhelms-Universität Münster
University of Münster

Sciebo

- Owncloud based Sync&Share service for more than 100k users in North-Rhine Westphalia, Germany
- Migrating from three independent server sites in NRW to two in Münster
- Migration to OCIS planned for later this year



Additional Projects

- Sciebo Plus -> Integration in mail service
- ScienceMesh
- Sciebo RDS -> Sciebo Research Data Services

Sciebo RDS: Mission

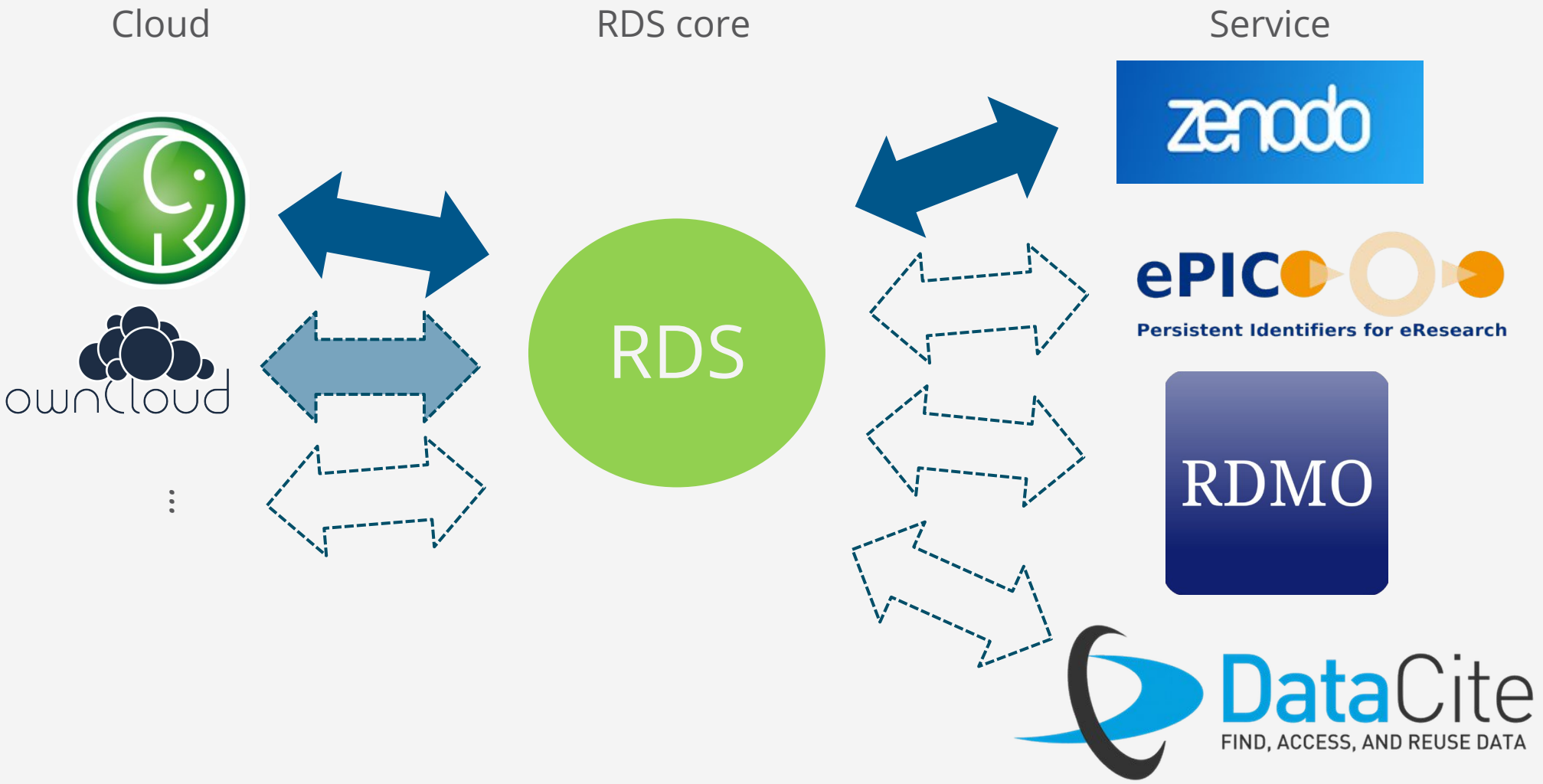
- We connect existing research data services (RDS)
  - Integration where the scientists are  

Make it easy for

 Users = UX, Design Thinking, User Stories

 Admins = Simple configuration, logging

</> Developers = Modular Architecture, CI / CD, documentation ...

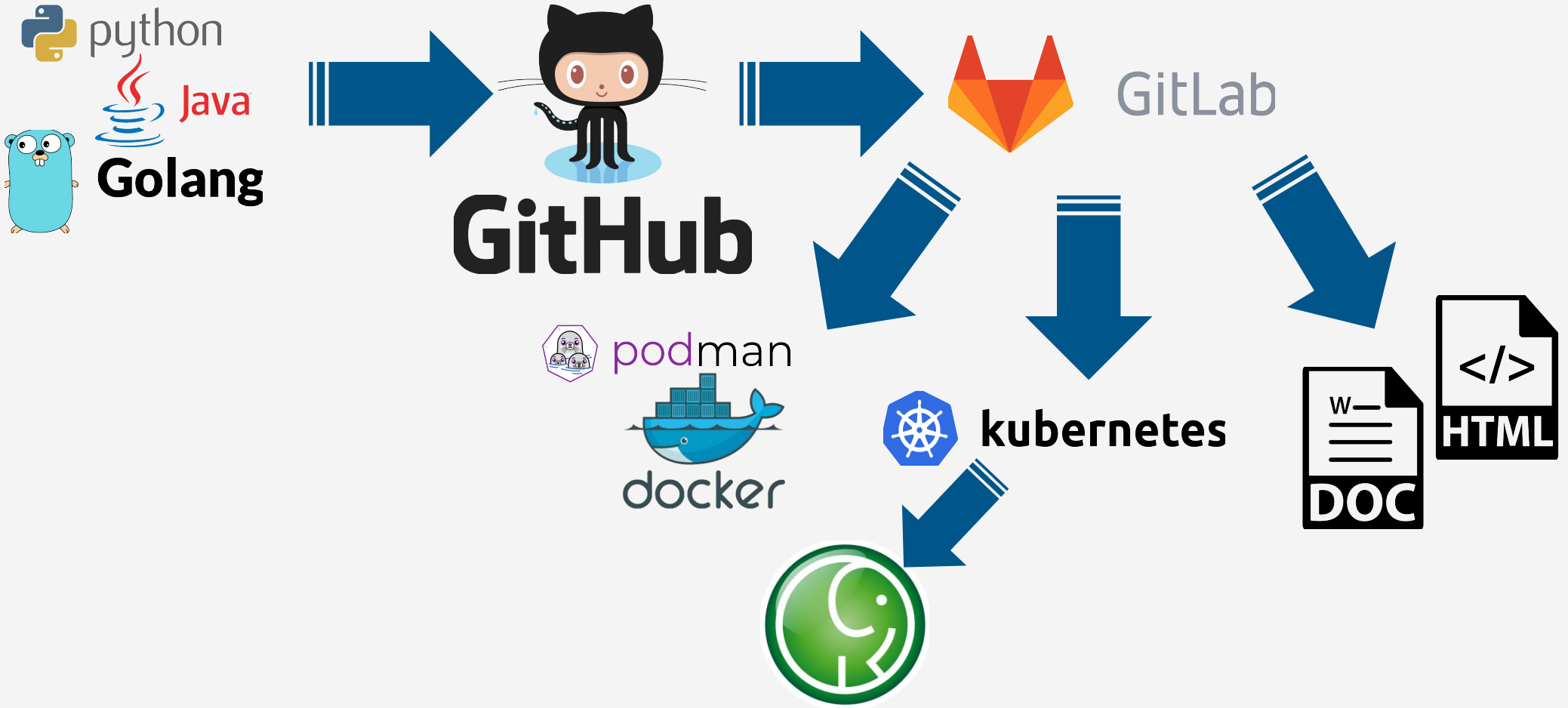

Cloud
RDS core
Service
ownCloud
RDS
zenodo
ePIC
Persistent Identifiers for eResearch
RDMO
DataCite
FIND, ACCESS, AND REUSE DATA

1. Iteration - first workflow ✅

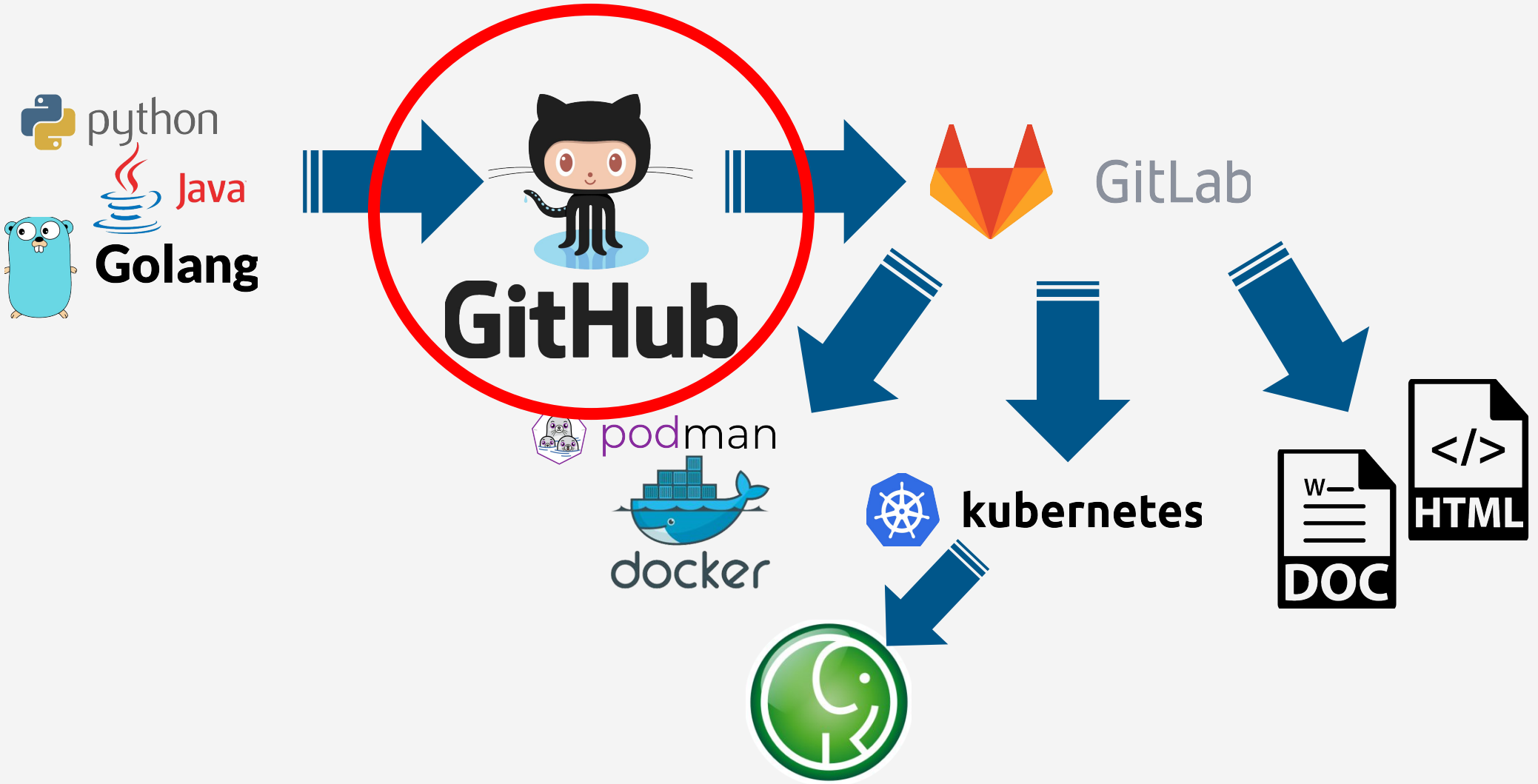
2. Iteration - extended workflow - WIP

What about you?

www.research-data-services.org


python
Java
Golang
GitHub
GitLab
podman
docker
kubernetes
W
DOC
</>
HTML

Pull requests from third parties



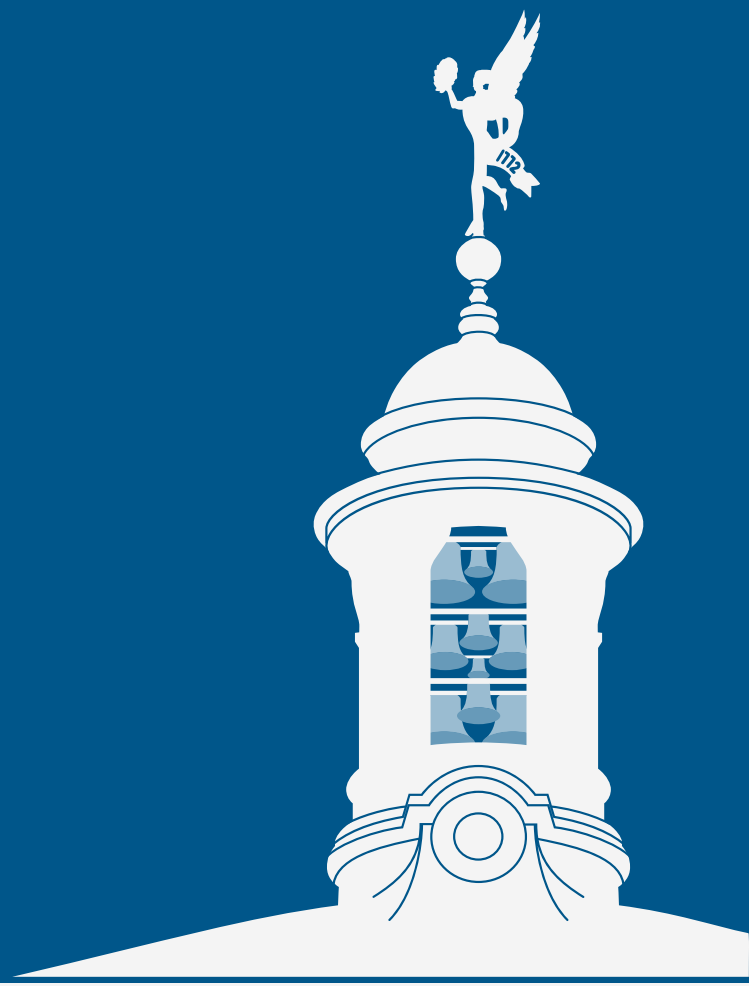
funded by DFG

Partners WWU MÜNSTER + Universität Bielefeld

UNIVERSITÄT DUISBURG ESSEN

WWU
MÜNSTER

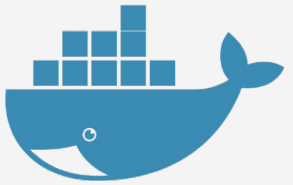
Thank you for your attention.



Peter Heiss (peter.heiss@uni-muenster.de),
Jens Stegmann (jstegman@uni-muenster.de),
Holger Angenent (holger.angenent@uni-muenster.de)

wissen.leben

Web: www.research-data-services.org

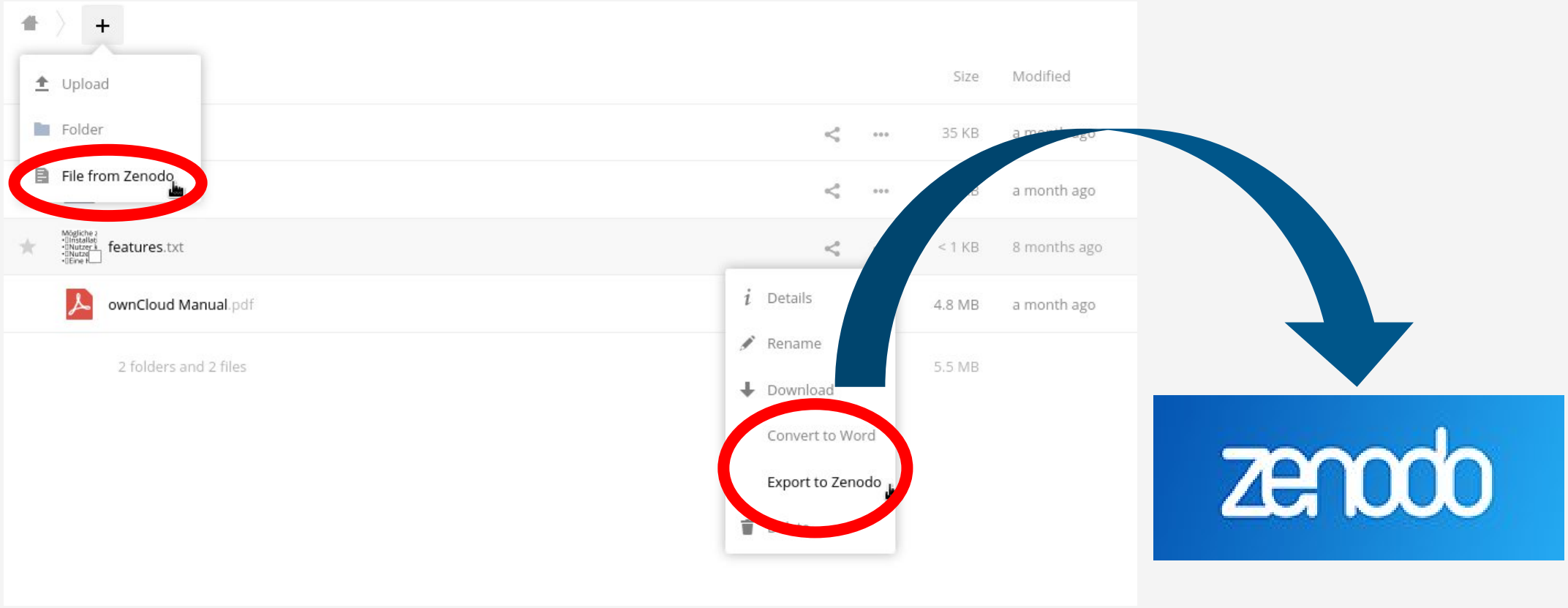


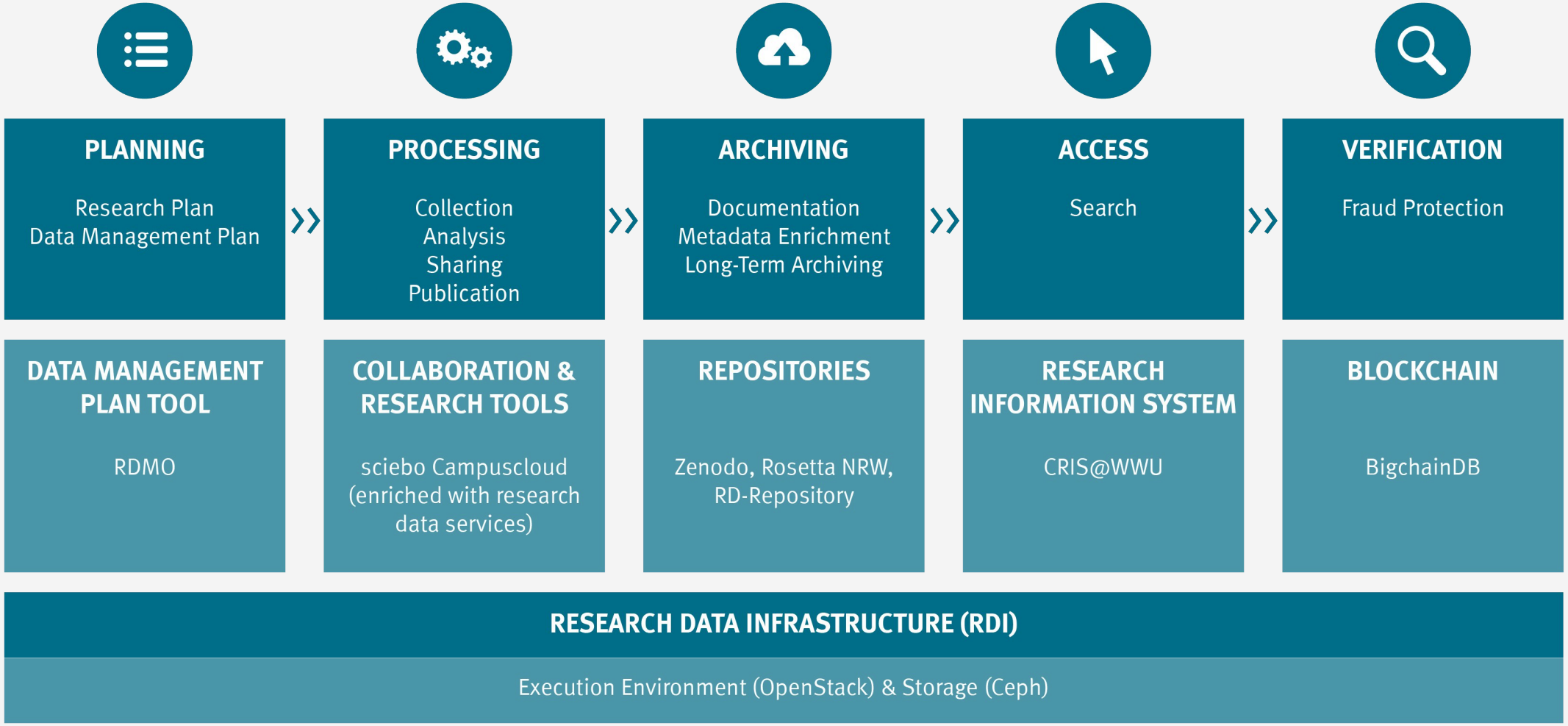
Architecture Overview

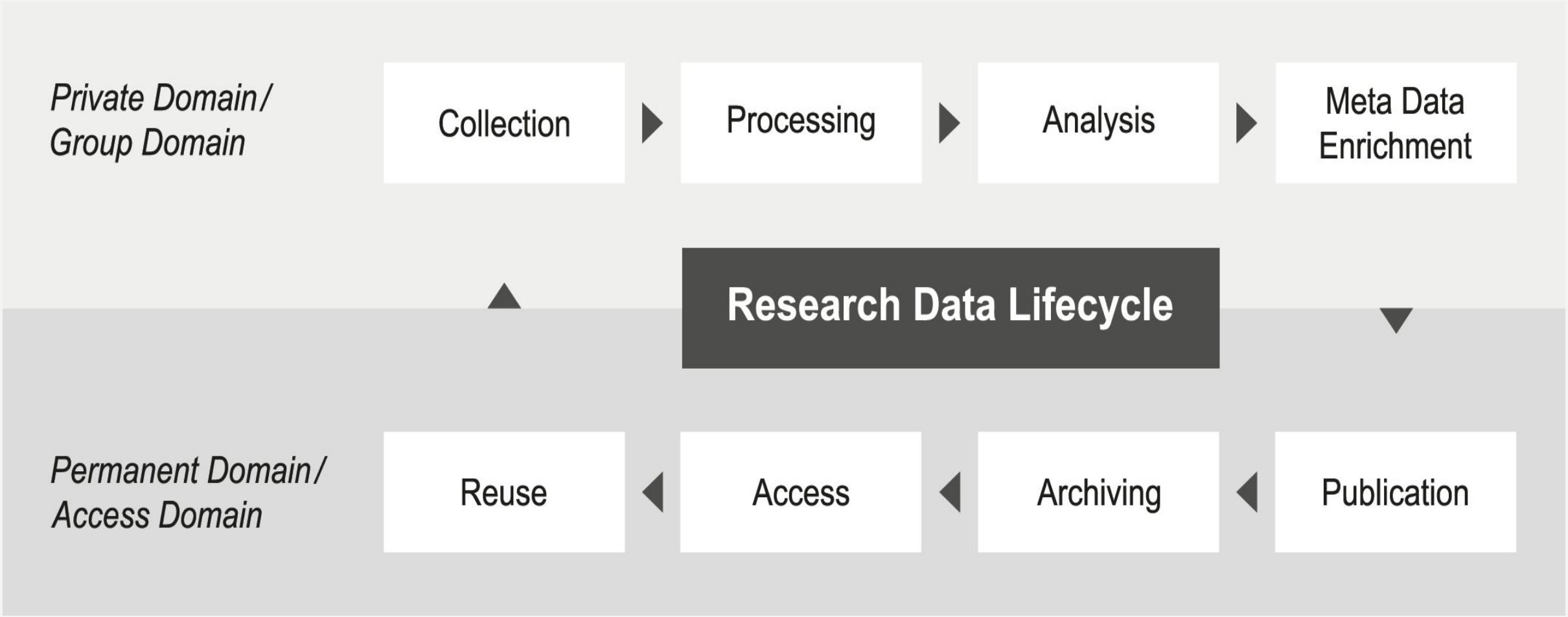
- Microservices, SOLID Principles
- Clean Architecture (Martin, 2017)

Technology Stack

- Owncloud, Open API v3, OAuth2, Docker, Kubernetes, Python, Flask, Gitlab, Github

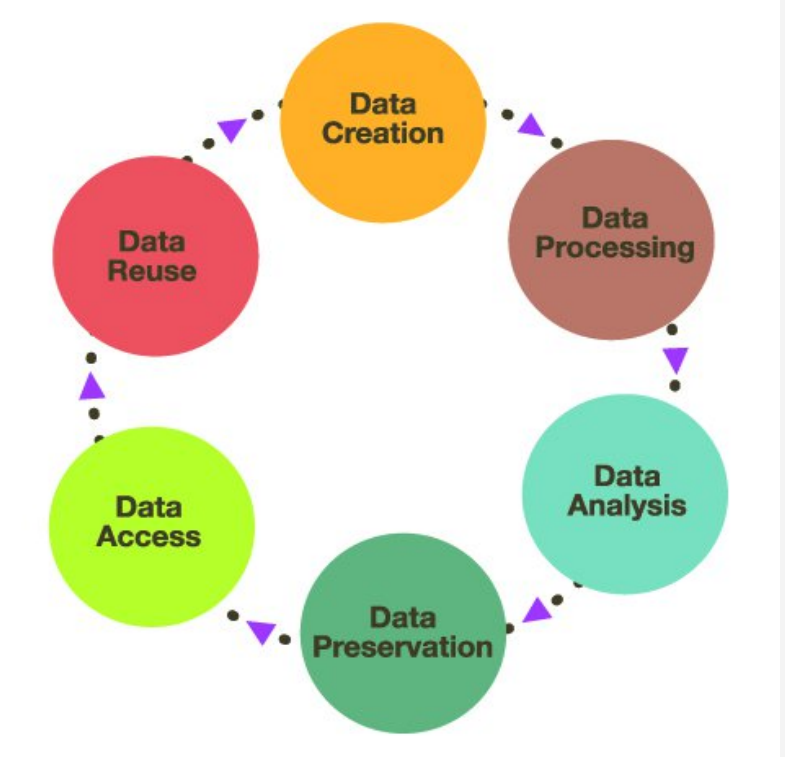

Upload
Folder
File from Zenodo
Size
Modified
35 KB
a month ago
features.txt
< 1 KB
8 months ago
ownCloud Manual.pdf
4.8 MB
a month ago
2 folders and 2 files
5.5 MB
Details
Rename
Download
Convert to Word
Export to Zenodo
zenodo


PLANNING
Research Plan
Data Management Plan
PROCESSING
Collection
Analysis
Sharing
Publication
ARCHIVING
Documentation
Metadata Enrichment
Long-Term Archiving
ACCESS
Search
VERIFICATION
Fraud Protection
DATA MANAGEMENT PLAN TOOL
RDMO
COLLABORATION & RESEARCH TOOLS
sciebo Campuscloud (enriched with research data services)
REPOSITORIES
Zenodo, Rosetta NRW, RD-Repository
RESEARCH INFORMATION SYSTEM
CRIS@WWU
BLOCKCHAIN
BigchainDB
RESEARCH DATA INFRASTRUCTURE (RDI)
Execution Environment (OpenStack) & Storage (Ceph)


Private Domain/
Group Domain
Collection
Processing
Analysis
Meta Data
Enrichment
Research Data Lifecycle
Permanent Domain/
Access Domain
Reuse
Access
Archiving
Publication

Research Data Management (RDM)

- Process concerned with all methods and policies to be applied in order to reach the goal of long-term (re-)usability of research data.
- Stages: Phases in Research Data Lifecycle
- ideal result: self-describing research data
- Data Management Plan (DMP): primary documentation, should be written at the start of a project / consortium

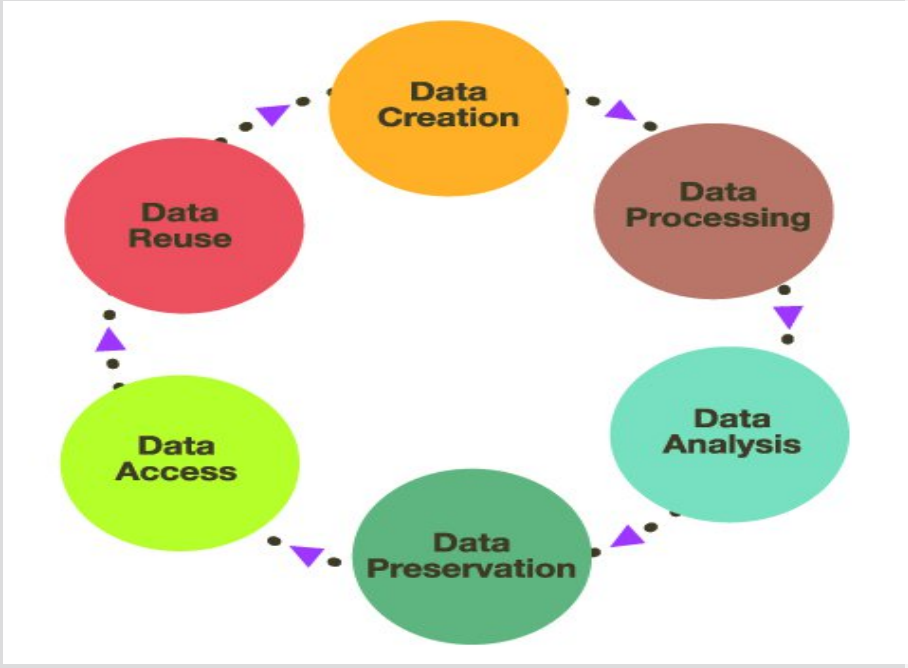


https://blogs.ntu.edu.sg/lib-datamanagement/data-lifecycle/

Research Data Lifecycle

- steps to be taken at different stages
- possible differences wrt :
  - research data types
  - academic disciplines

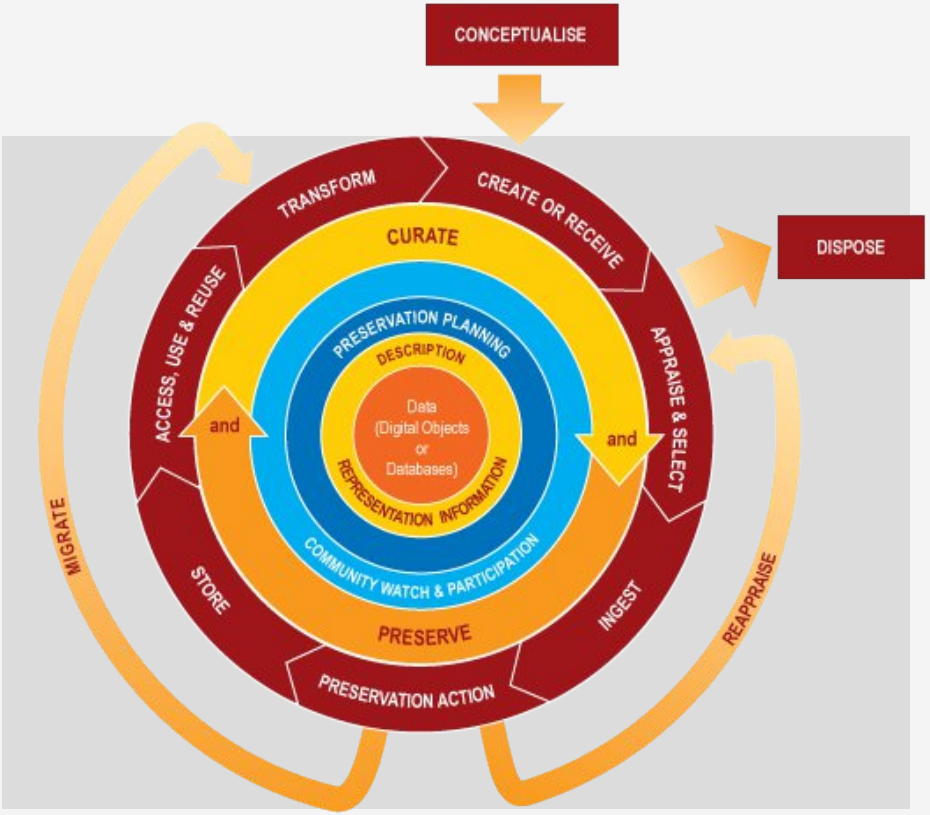
-> specialized RDSes



https://blogs.ntu.edu.sg/lib-datamanagement/data-lifecycle/

DCC Curation Lifecycle Model

- more fine-grained model, but still idealized
- users can enter at any stage
- enables to identify granular functionality wrt the curation and preservation of RD:
  - actions required or not required
  - roles and responsibilites
  - documentation of processes + policies
- userful for planning respective activities



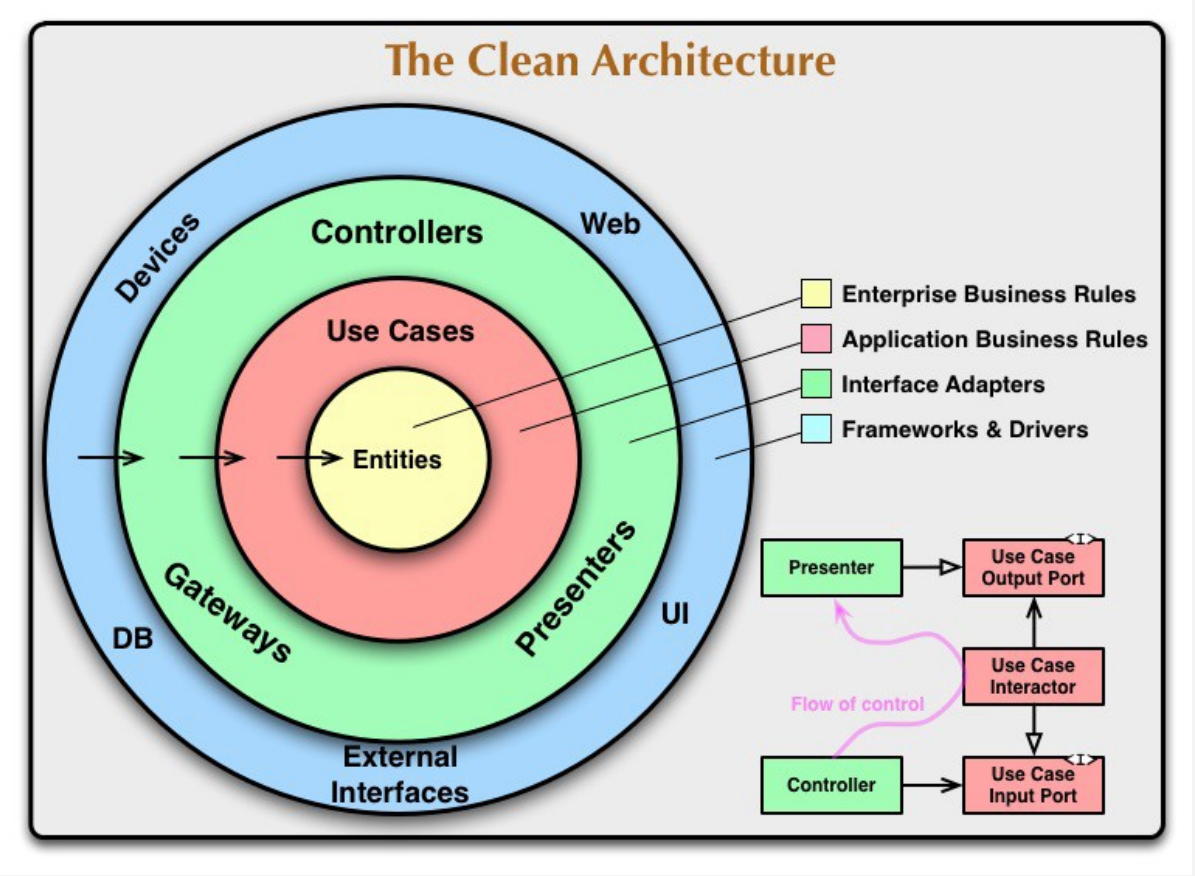
http://www.dcc.ac.uk/resources/curation-lifecycle-model

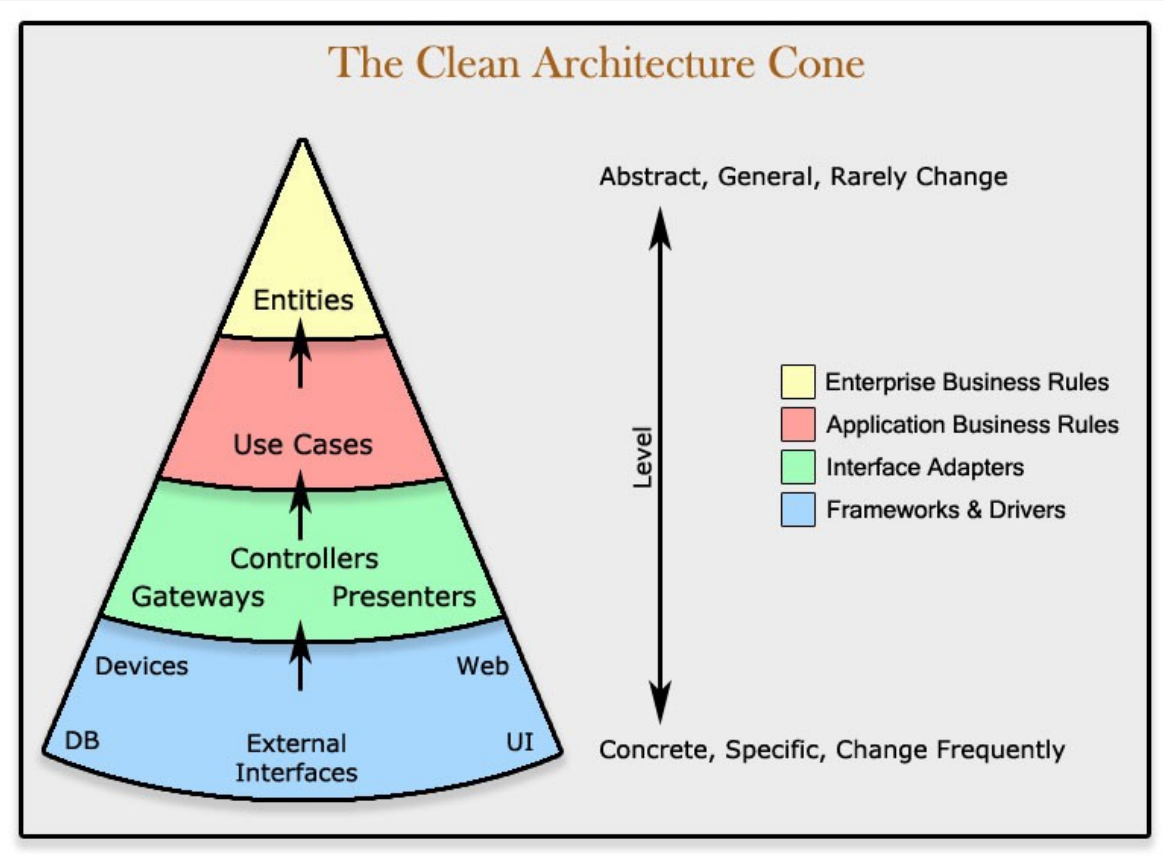
SOLID Principles

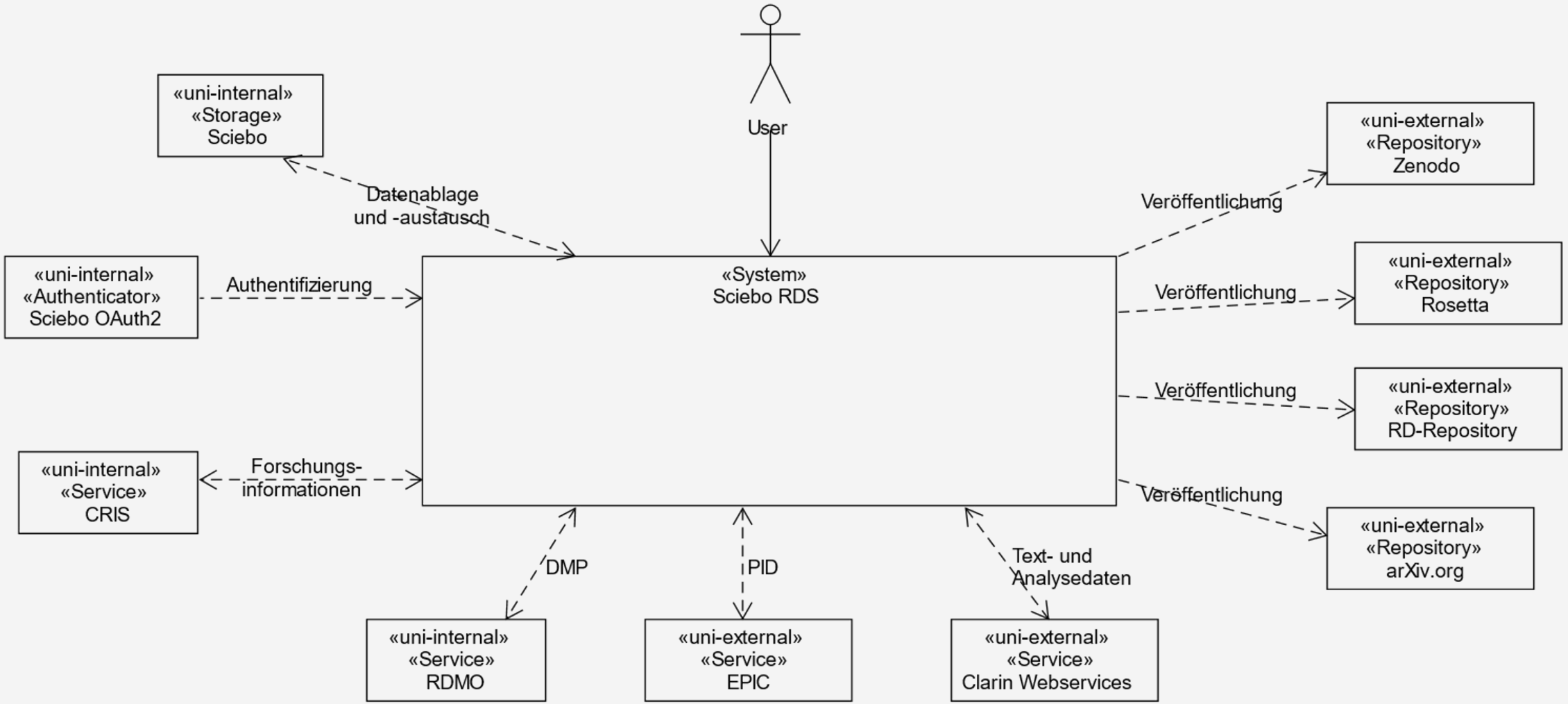
In order to simplify and clarify the design, implementation and maintenance

- **Single Responsibility:** A class should only have a single responsibility.
- **Open-Closed:** Software entities should be open for extension, but closed for modification.
- **Liskov Substitution:** Objects in a program should be replaceable with instances of their subtypes without altering the correctness of that program.
- **Interface Segregation:** Many client-specific interfaces are better than one general-purpose interface.
- **Dependency Inversion:** One should depend upon abstractions, not concretions.

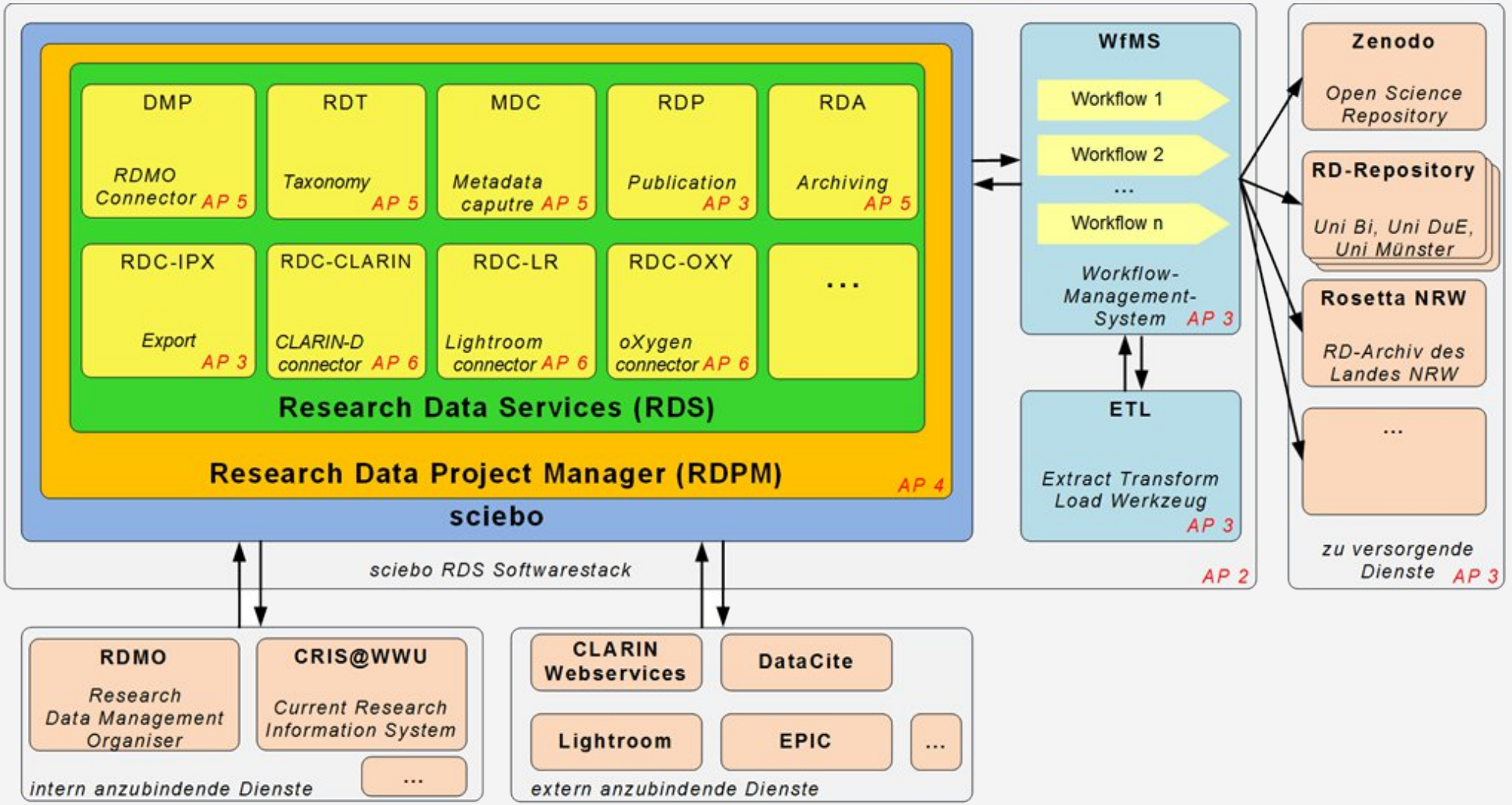
Clean Architecture

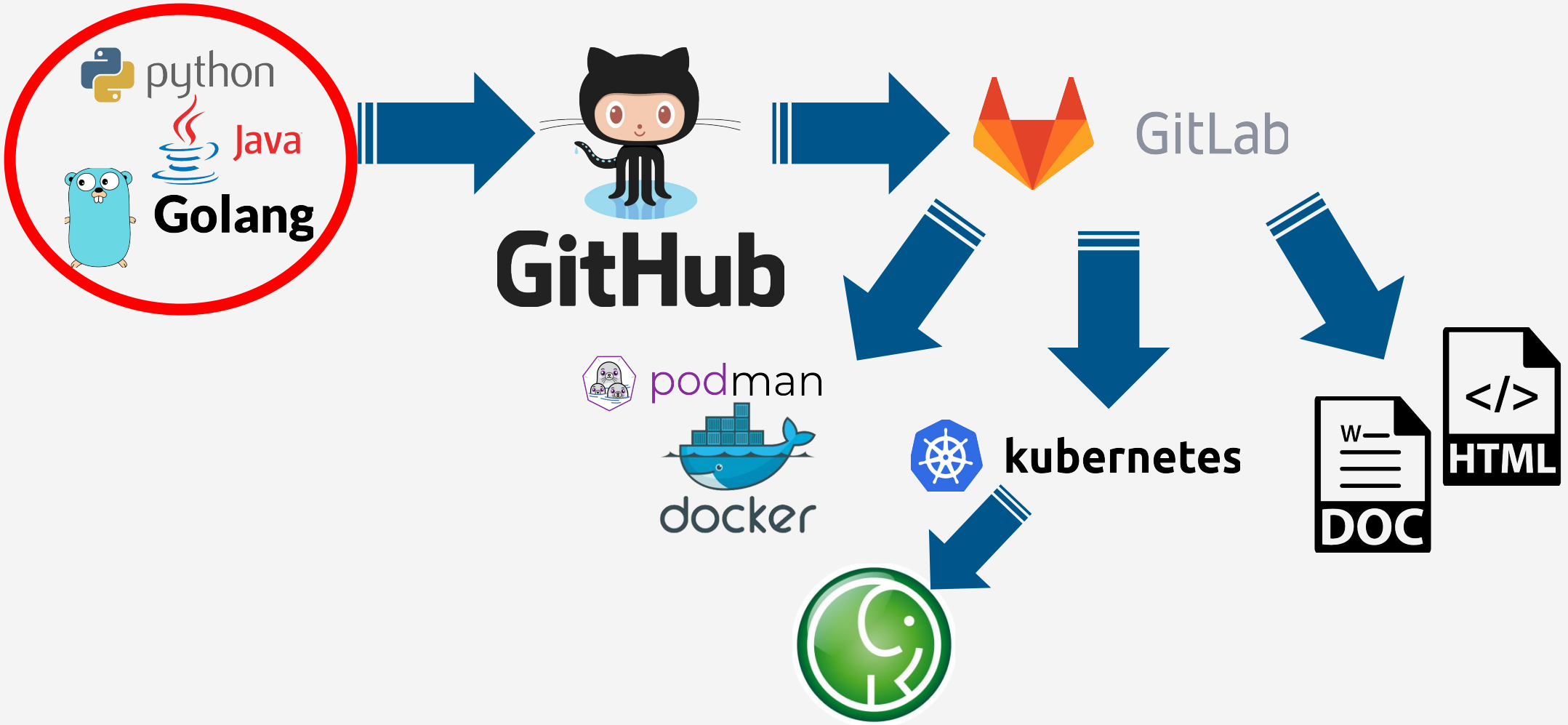


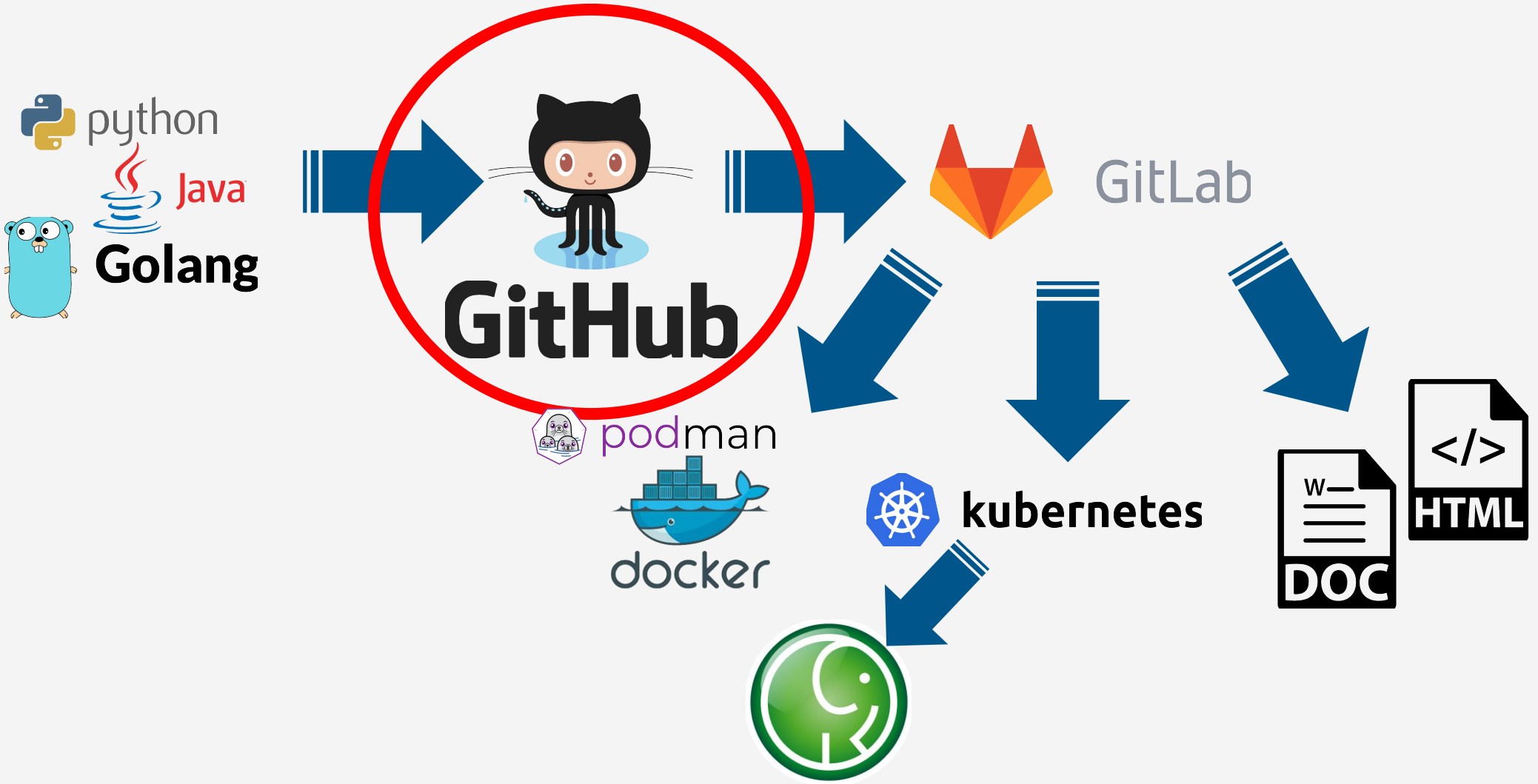


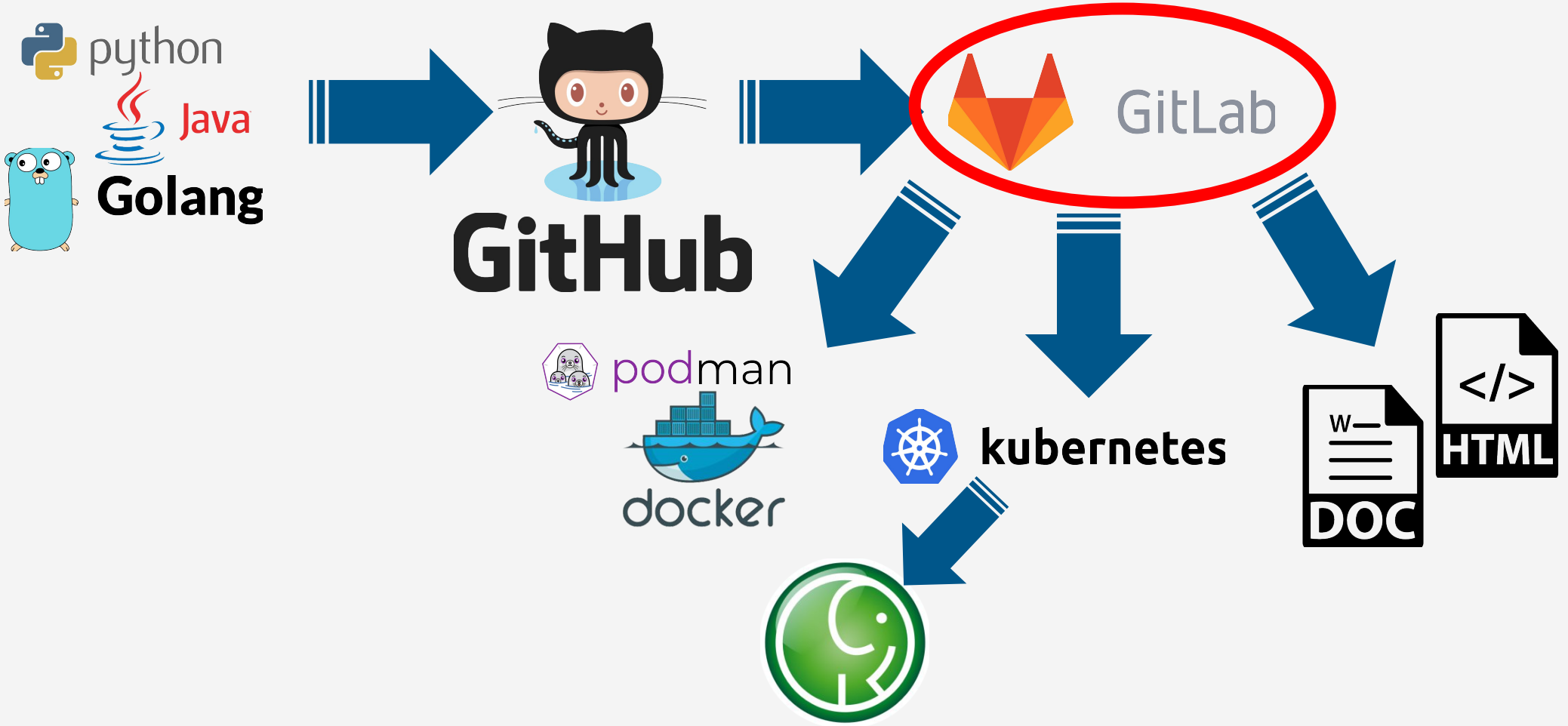

«uni-internal»
«Storage»
Sciebo
User
«uni-external»
«Repository»
Zenodo
Datenablage
und -austausch
Veröffentlichung
«uni-internal»
«Authenticator»
Sciebo OAuth2
Authentifizierung
«System»
Sciebo RDS
Veröffentlichung
«uni-external»
«Repository»
Rosetta
Veröffentlichung
«uni-external»
«Repository»
RD-Repository
«uni-internal»
«Service»
CRIS
Forschungs-
informationen
Veröffentlichung
«uni-external»
«Repository»
arXiv.org
DMP
PID
Text- und
Analysedaten
«uni-internal»
«Service»
RDMO
«uni-external»
«Service»
EPIC
«uni-external»
«Service»
Clarin Webservices

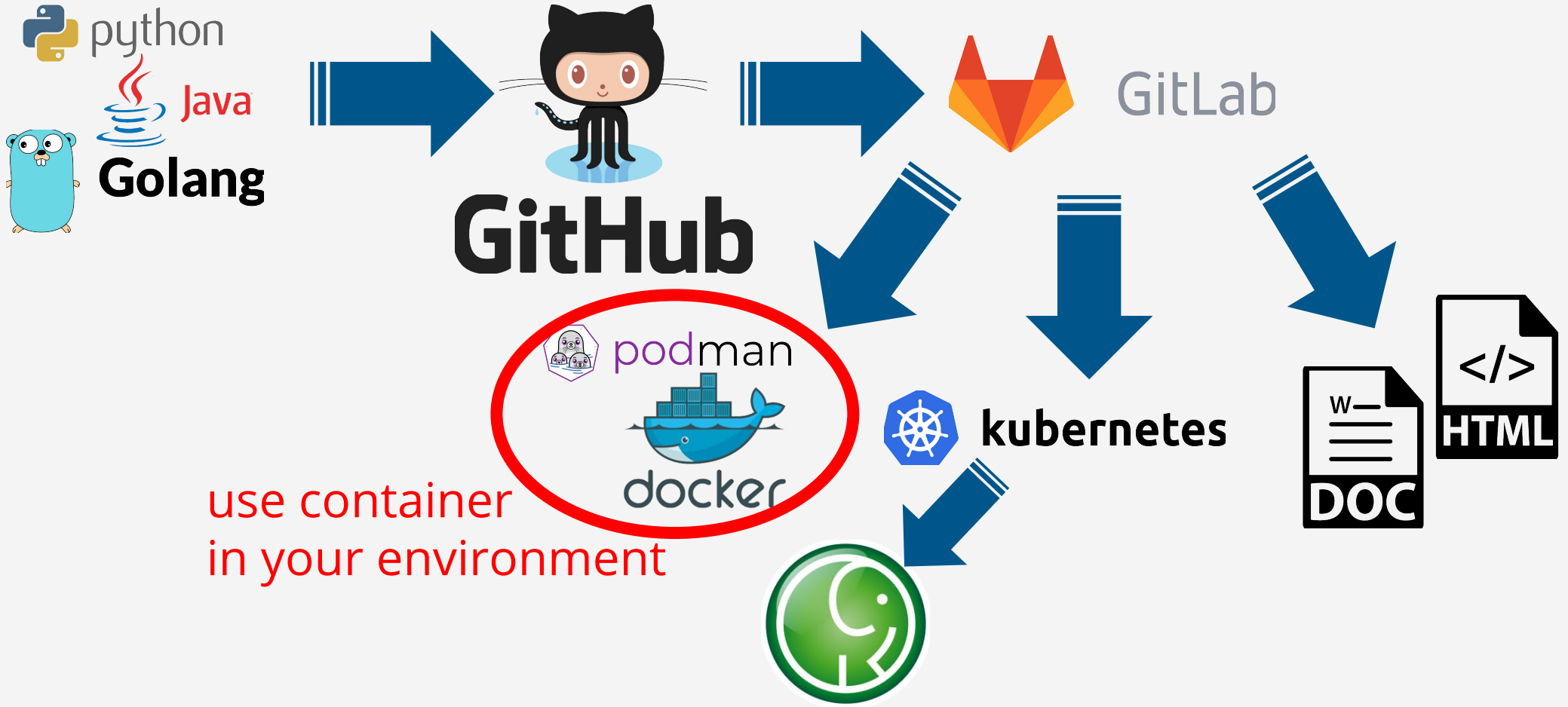
Architecture: Sciebo RDS

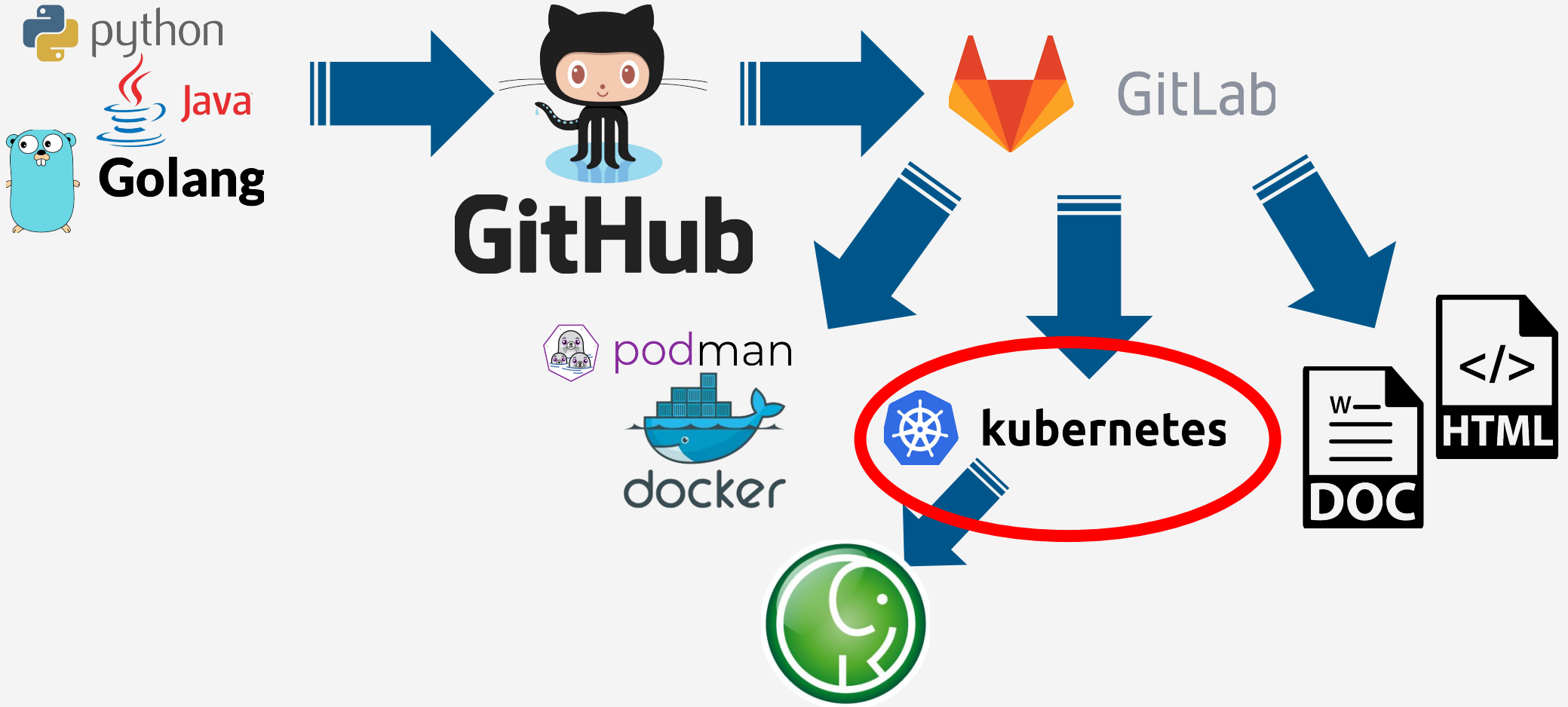


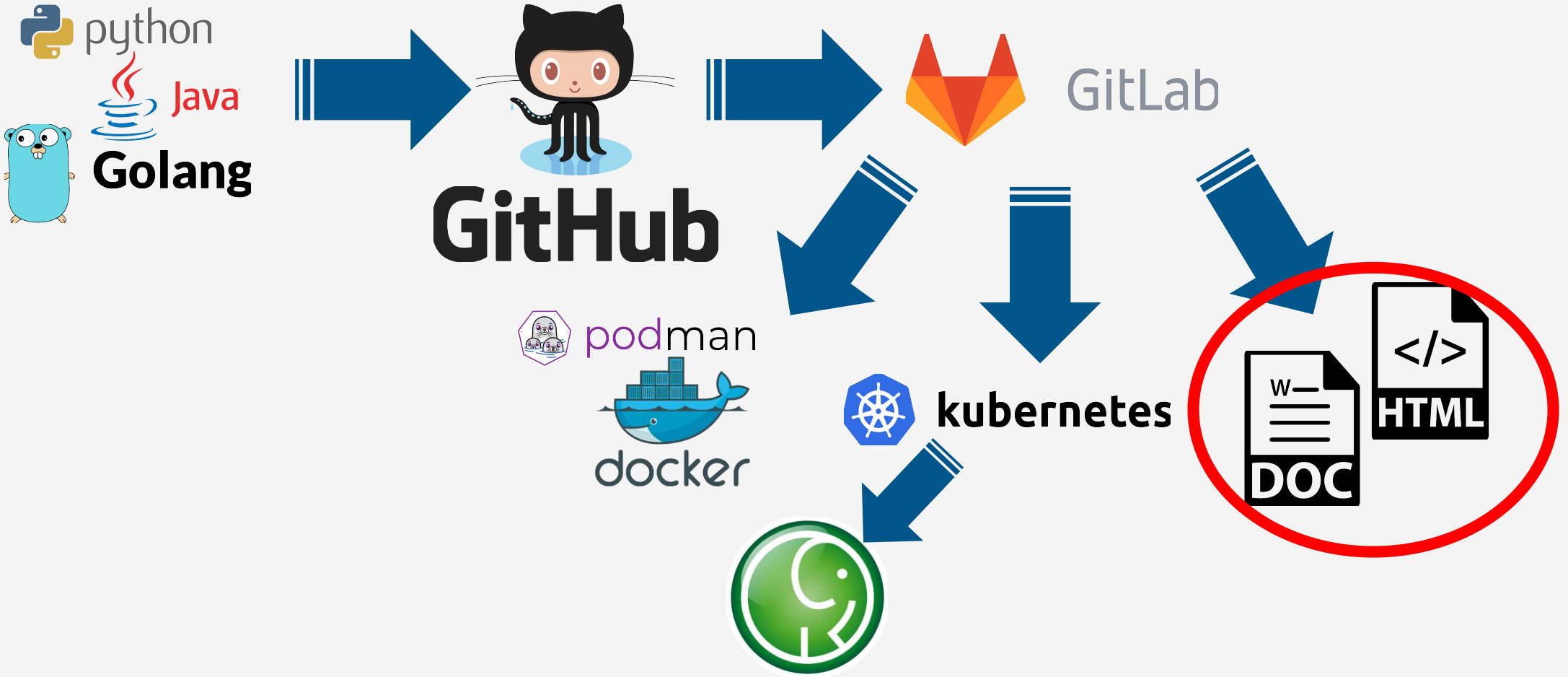

python
Java
Golang
GitHub
GitLab
podman
docker
kubernetes
W
DOC
</>
HTML

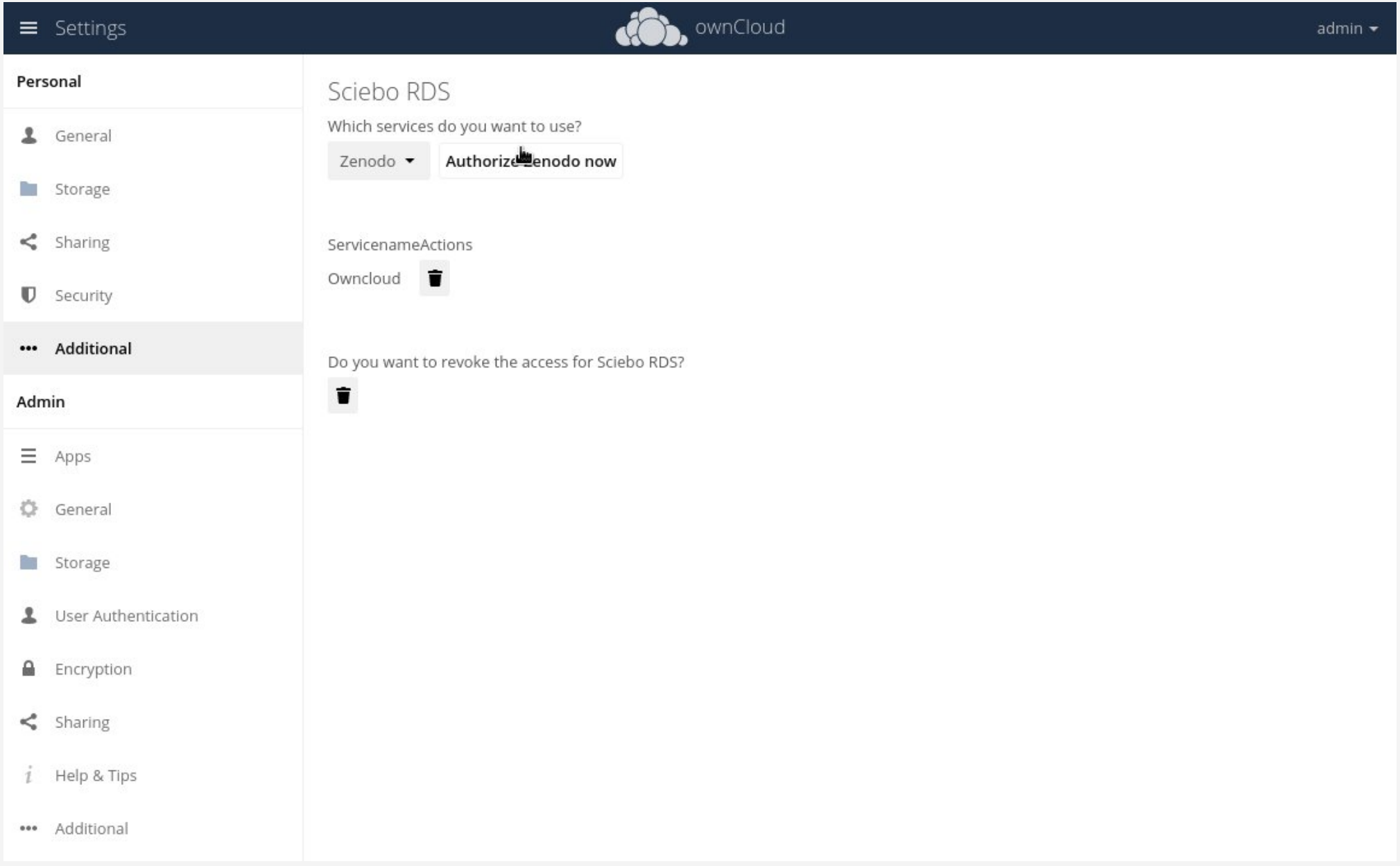

python
Java
Golang
GitHub
GitLab
podman
docker
kubernetes
</>
HTML
W
DOC

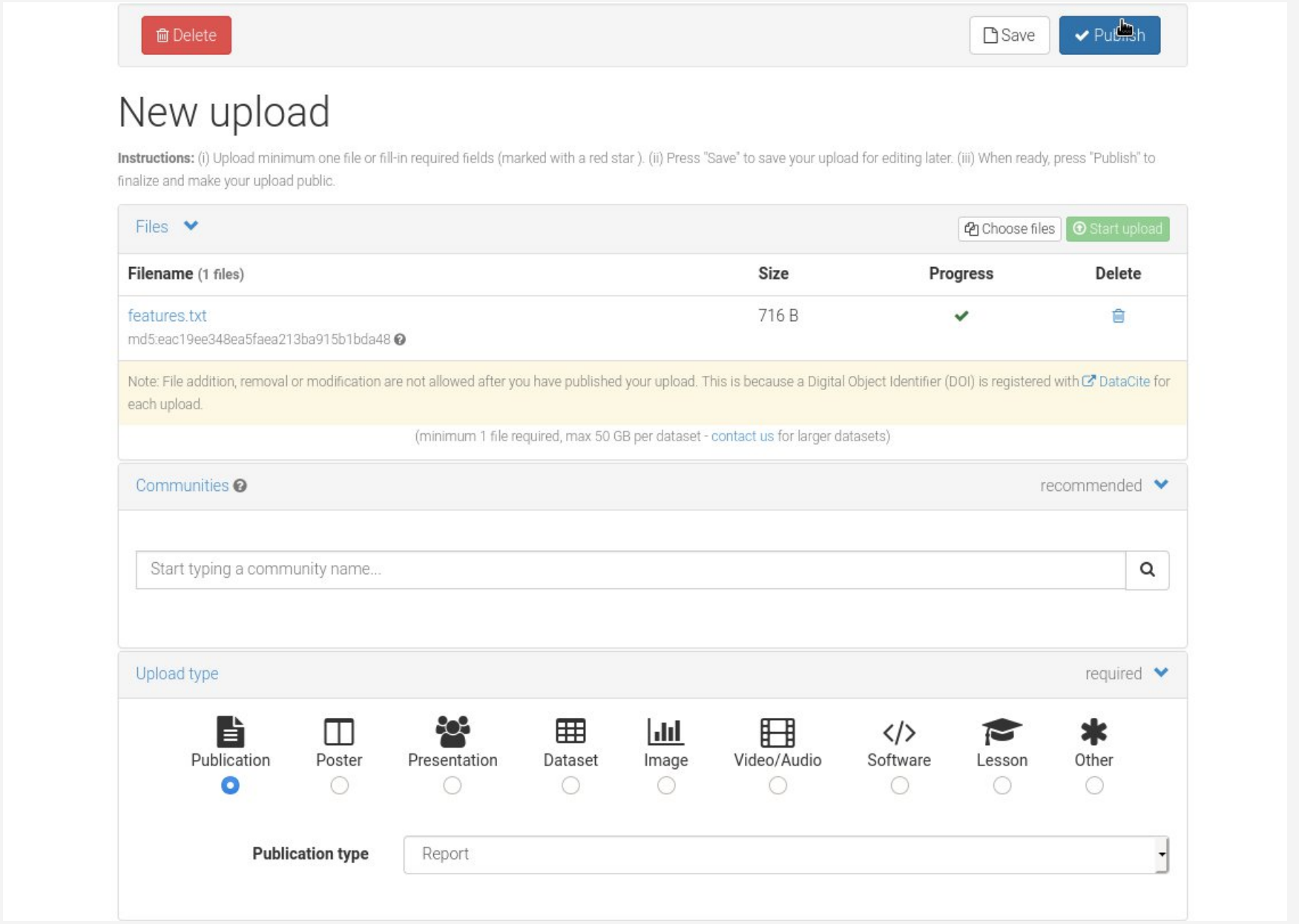

python
Java
Golang
GitHub
GitLab
podman
docker
kubernetes
W
DOC
</>
HTML


python
Java
Golang
GitHub
GitLab
podman
docker
kubernetes
use container
in your environment
W
DOC
</>
HTML


python
Java
Golang
GitHub
GitLab
podman
docker
kubernetes
W
DOC
</>
HTML


python
Java
Golang
GitHub
GitLab
podman
docker
kubernetes
W
DOC
</>
HTML


Settings
ownCloud
admin
Personal
General
Storage
Sharing
Security
Additional
Admin
Apps
General
Storage
User Authentication
Encryption
Sharing
Help & Tips
Additional
Sciebo RDS
Which services do you want to use?
Zenodo
Authorize Zenodo now
ServicenameActions
Owncloud
Do you want to revoke the access for Sciebo RDS?


Delete
Save
Publish
New upload
Instructions: (i) Upload minimum one file or fill-in required fields (marked with a red star). (ii) Press "Save" to save your upload for editing later. (iii) When ready, press "Publish" to finalize and make your upload public.
Files
Choose files
Start upload
Filename (1 files)
Size
Progress
Delete
features.txt
md5:eac19ee348ea5faea213ba915b1bda48
716 B
Note: File addition, removal or modification are not allowed after you have published your upload. This is because a Digital Object Identifier (DOI) is registered with DataCite for each upload.
(minimum 1 file required, max 50 GB per dataset - contact us for larger datasets)
Communities
recommended
Start typing a community name...
Upload type
required
Publication
Poster
Presentation
Dataset
Image
Video/Audio
Software
Lesson
Other
Publication type
Report

Why?

Enabling the Users

- most wanted: requirements in terms of structured RDM functionalities, support, integration etc
  - cafeteria model, support of the FAIR principles (findable, accessible, interoperable, reusable), agile requirements engineering (user stories, personas, design thinking)
- usability: focus on user experience (UX), RDM should be made as easy and appealing as possible
  - comply to best practices for user interface design !?

Why?

Enabling the Developers

- extensible: integration of new services
- configurable: wrt. local needs
- reusable: making use of the RDS modules in other contexts

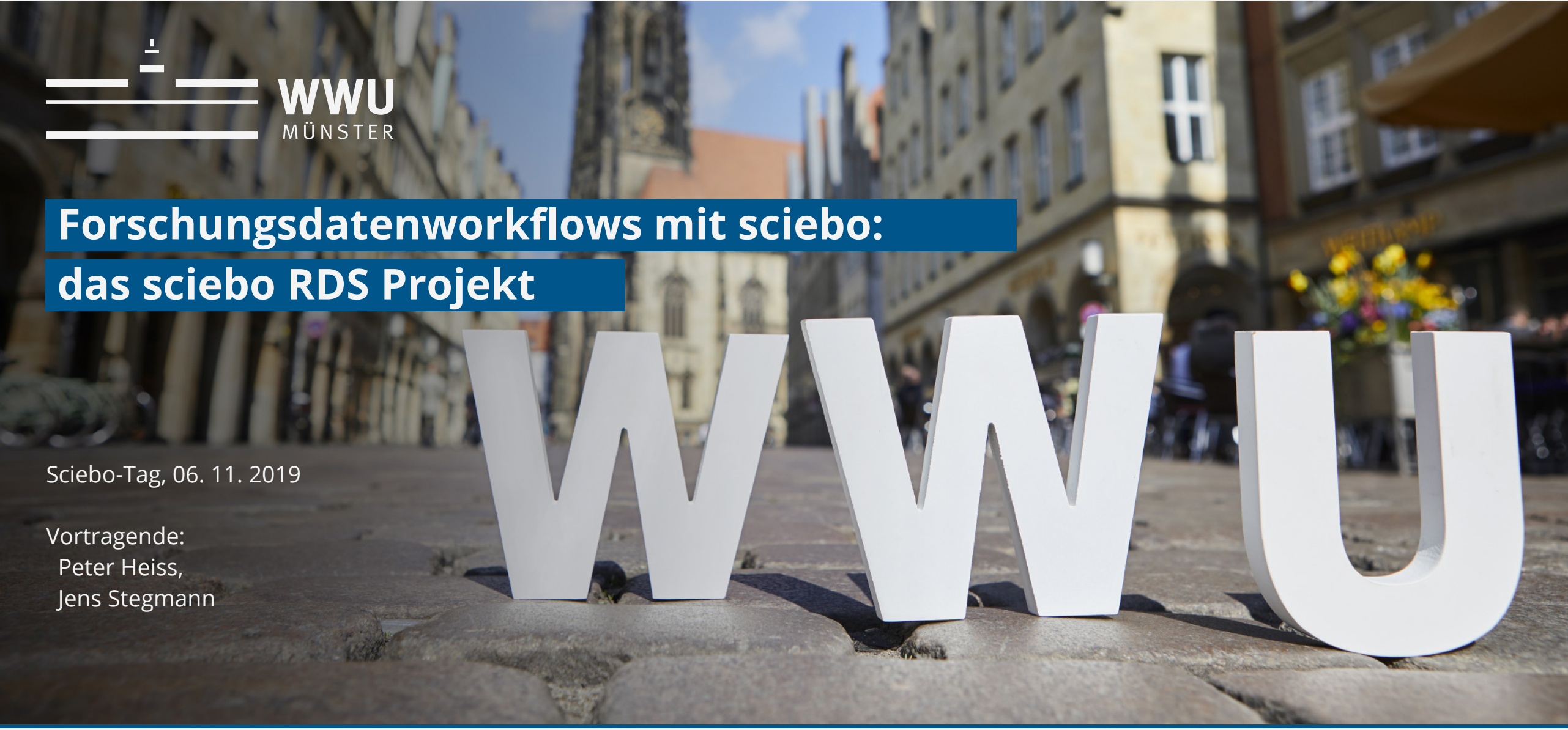
Benefits

- Integration into existing applications (Owncloud/Sciebo)
- Integration of existing services (e.g., Zenodo) through established protocols (e.g., OAuth2)
- Reusability of developed modules is possible (RDS Microservices in other contexts)

Horizontal scalability via Kubernetes

The talk will introduce the applied architectural features and discuss aspects of their implementation in depth, i.e., we intend to show how specific parts of a scalable ecosystem based on microservices and containers on top of Kubernetes can integrate and function for this purpose. Concerning the interfaces of the microservices under discussion, we showcase the usage of the OpenAPI v3 specification, which additionally fosters reusability.

Sciebo RDS is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 403637381.



Forschungsdatenworkflows mit sciebo: das sciebo RDS Projekt

Sciebo-Tag, 06. 11. 2019

Vortragende:
Peter Heiss,
Jens Stegmann

wissen.leben

Universität Münster
Universität Duisburg-Essen

Steckbrief: sciebo RDS

- **sciebo Research Data Services** – Forschungsdatenmanagementdienste und -werkzeuge für Wissenschaftler (kurz: sciebo RDS)
- **DFG-Projekt** von **Uni Münster** (ULB, ZIV) und der **Uni Duisburg-Essen** (Prof. Stieglitz, Professur für Professionelle Kommunikation in elektronischen Medien / Social Media)
- **Laufzeit 36 Monate, Beginn 04/2019**
  4 x TV-L E 13 (1,0 DuE; 1,5 ZIV; 1,5 ULB)
- **Kooperation mit Uni Bielefeld** (ohne Projektförderung)
- Webadresse: www.research-data-services.de

Warum sciebo RDS?

Was die Wissenschaftler wollen ...

- **Tools und Dienste müssen auf die Arbeitsabläufe der Forscher abgestimmt sein**, die oft fachspezifisch (und manchmal sogar projektspezifisch) sind.
- **Die Forscher widersetzen sich vorgeschriebenen, starren Systemen**.
- **Die Forscher favorisieren ein "Cafeteria"-Modell**, bei dem sie aus einer Reihe von Dienstleistungen wählen können.
- **Tools und Dienste müssen einfach zu bedienen sein**.

Warum sciebo RDS?

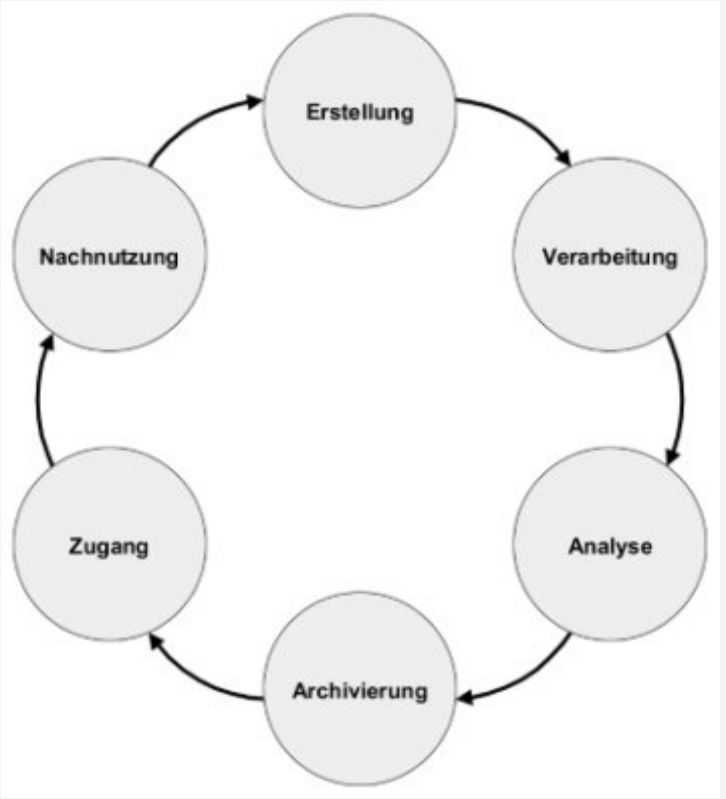
Was die Wissenschaftler wollen …

- **Die Forscher wollen die Kontrolle darüber haben, was mit ihren Daten passiert**, wer Zugang zu ihnen hat, und unter welchen Bedingungen. Folglich wollen sie sicher sein, dass derjenige, der mit ihren Daten umgeht (Rechenzentrum, Bibliothek, etc.), ihre Interessen respektiert.
- Die Forscher erwarten, dass **Werkzeuge** und Dienstleistungen ihre tägliche Forschungsarbeit **unterstützen,** institutionelle Anforderungen müssen diesem Interesse untergeordnet sein.
- Die **Vorteile der Unterstützung müssen greifbar sein** – nicht in drei Jahren, sondern jetzt.
- Der **Support muss lokal, praxisnah** und bei Bedarf – **sofort** – **verfügbar sein**.

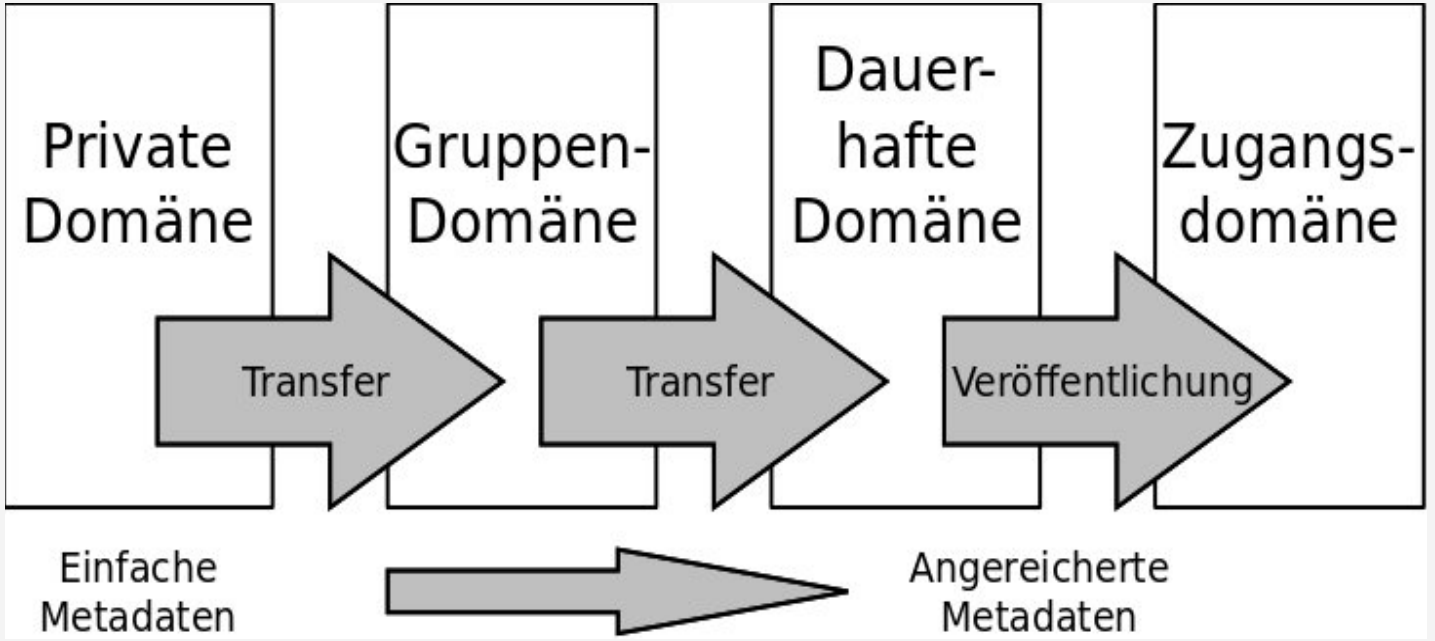
https://web.archive.org/web/20110409170938/http://www.surffoundation.nl/nl/publicaties/Documents/What_researchers_want.pdf

Forschungsdatenmanagement (FDM)

- Prozess, der **alle Methoden und Verfahren** umfasst, die **zur sicheren langfristigen Nutzbarkeit von Foschungsdaten** angewendet werden:
  - Generierung
  - Bearbeitung
  - Anreicherung
  - Archivierung
  - Veröffentlichung
- **selbstbeschreibende Forschungsdaten** als Ergebnis
- **Datenmanagementplan** (DMP) beschreibt Methoden und Verfahren, bereits zu Projektbeginn empfohlen



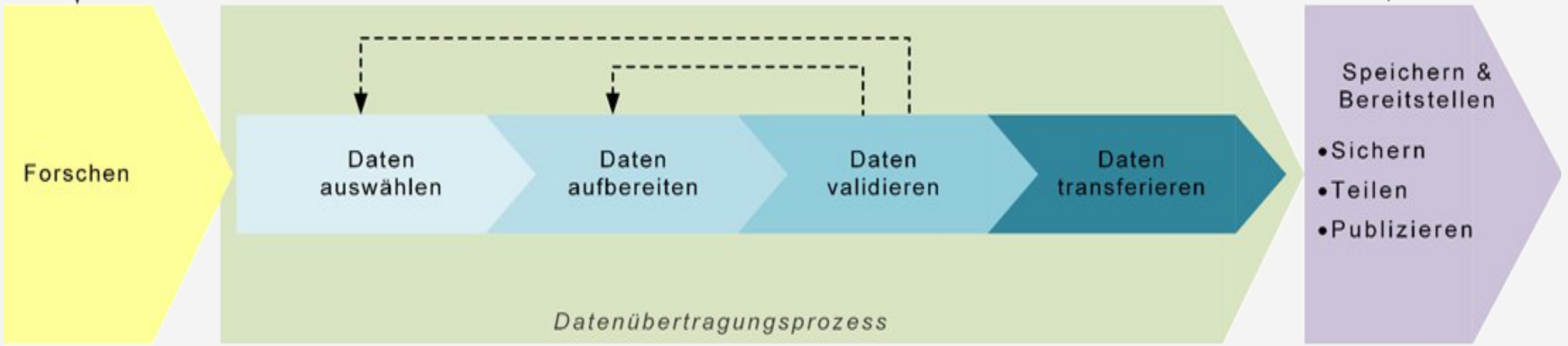
Curation Domain Model



http://www.forschungsdaten.org/index.php/Curation_Domain_Model

Private- bzw.
Gruppen-Domäne

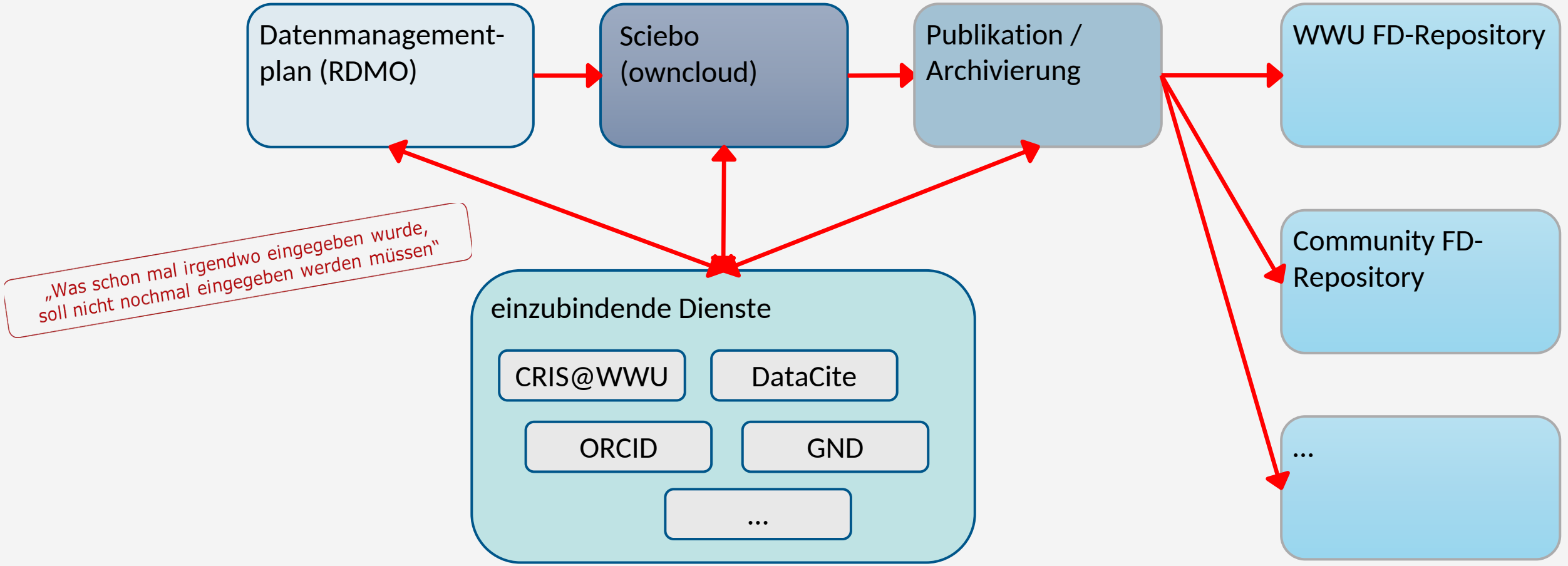
Dauerhafte- und
Zugangs-Domäne



Workflows: Metadaten

- Information für Archivierungs- u. Publikationsworkflows aus u.a.
  - Datenmanagementplan (RDMO)
  - Forschungsinformationssystem (CRIS)
- Unterschiedliche Repositorien: abweichende Anforderungen
- Metadaten müssen vom Nutzer eingegeben bzw. ergänzt werden
  - Templates für etablierte Metadatenstandards
  - Eigene Erfassungsschemata ermöglichen
  - Eingabemasken regelbasiert automatisch erzeugen,
  - Zulässige Auswahlmöglichkeiten anbieten
  - Nur das abfragen, was nötig ist

Schematische Übersicht: Der perfekte Workflow



sciebo Research Data Services

Kernaspekte: Die sciebo RDS …

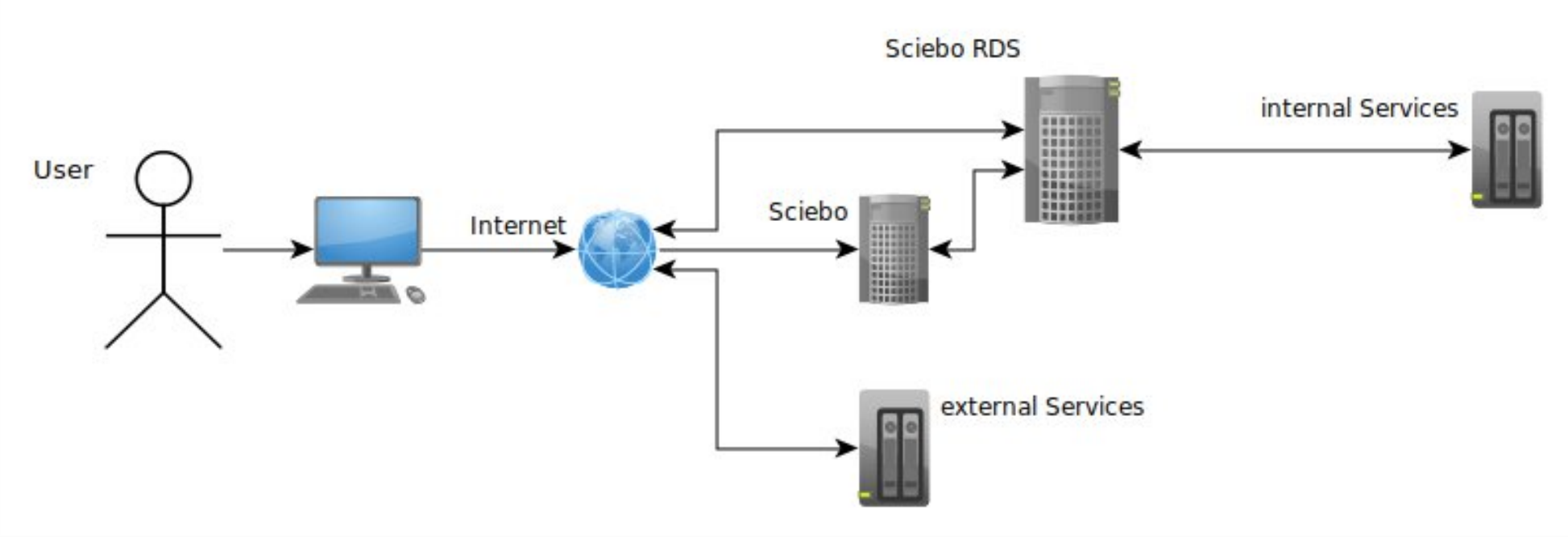
- integrieren externe Forschungsdaten-Services,
- bieten Brückenfunktionalitäten,
- adaptieren externe Expertenwerkzeuge,
- bieten grundlegende FDM-Funktionalitäten.

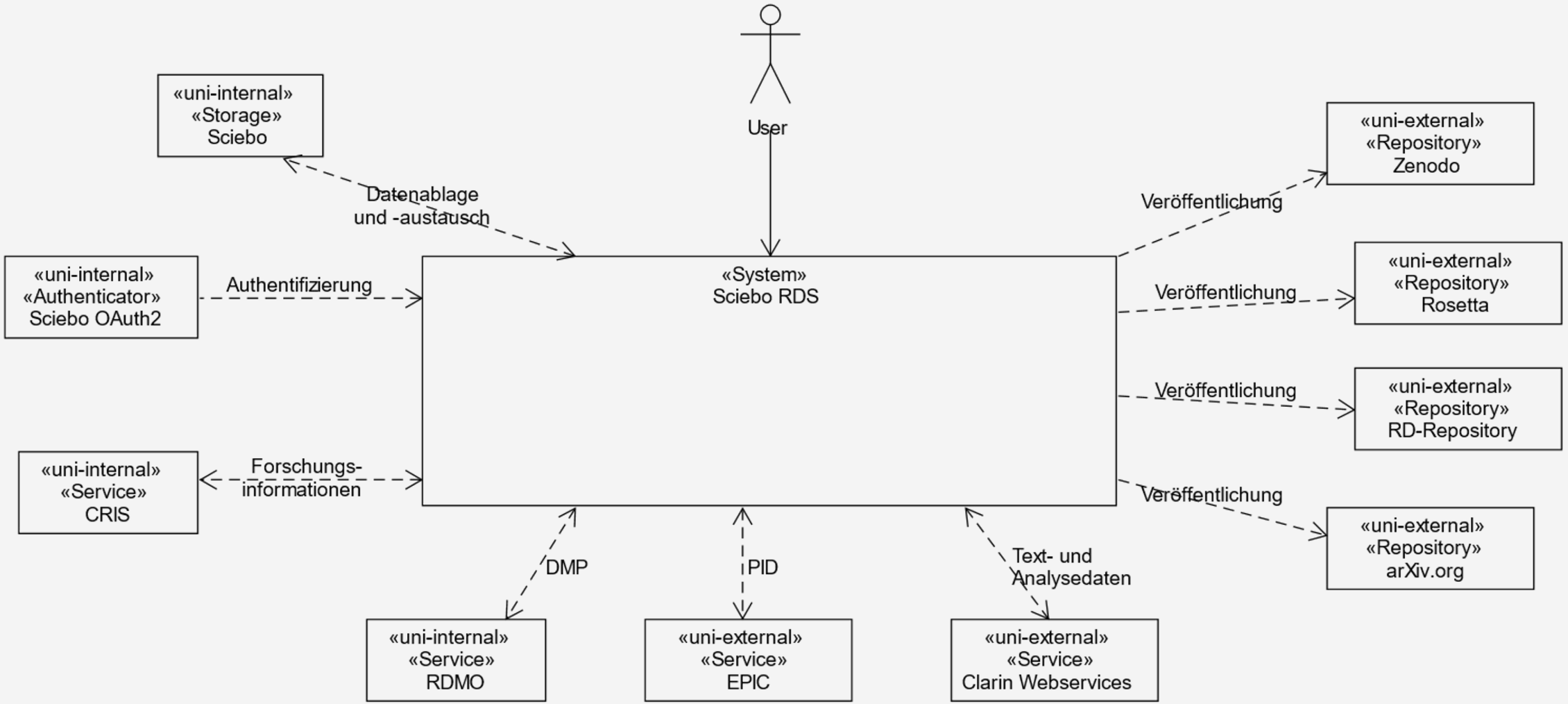
Lösungsansatz:

- Modulare Architektur + nachhaltige Infrastruktur,
- Free Open Source Software,
- Interoperabilität: Integration in die Arbeitsumgebung der Wissenschaftler,
- User Experience als strategischer Faktor.

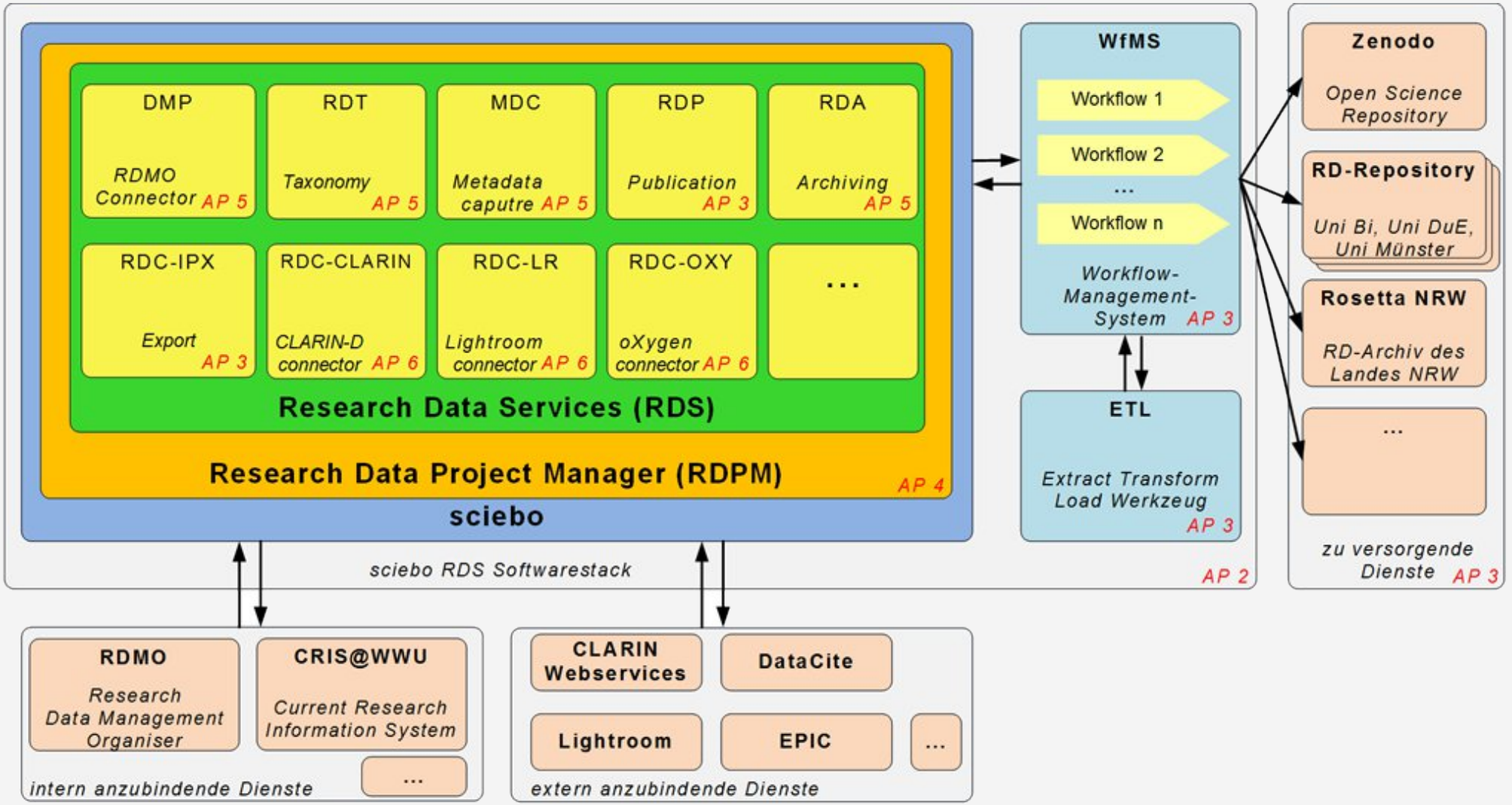
***Wichtig**: Es soll keine weitere Software zur Akkumulierung, sondern zur Aggregierung von Daten und Informationen entstehen.*

Wo befindet sich Sciebo RDS?




«uni-internal»
«Storage»
Sciebo
User
«uni-external»
«Repository»
Zenodo
Datenablage
und -austausch
Veröffentlichung
«uni-internal»
«Authenticator»
Sciebo OAuth2
Authentifizierung
«System»
Sciebo RDS
Veröffentlichung
«uni-external»
«Repository»
Rosetta
Veröffentlichung
«uni-external»
«Repository»
RD-Repository
«uni-internal»
«Service»
CRIS
Forschungs-
informationen
Veröffentlichung
«uni-external»
«Repository»
arXiv.org
DMP
PID
Text- und
Analysedaten
«uni-internal»
«Service»
RDMO
«uni-external»
«Service»
EPIC
«uni-external»
«Service»
Clarin Webservices

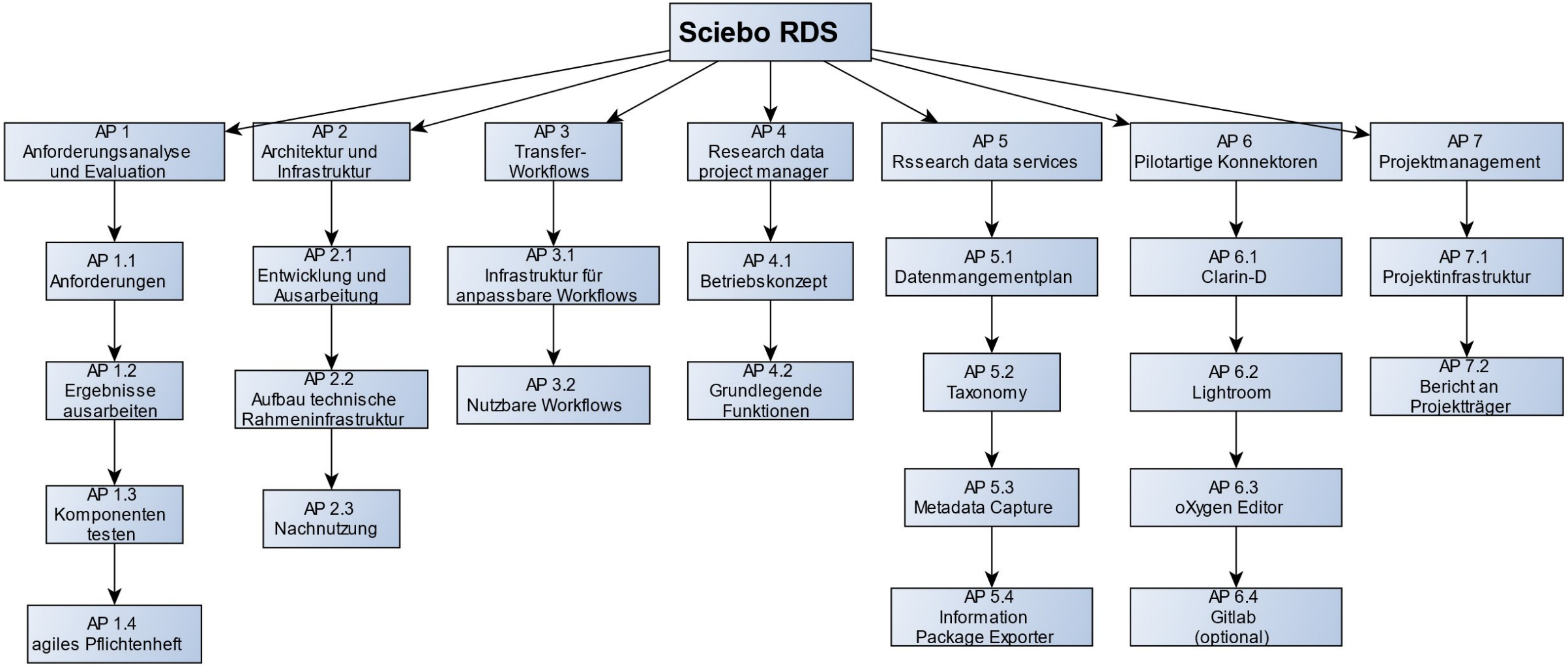
Architektur: sciebo RDS



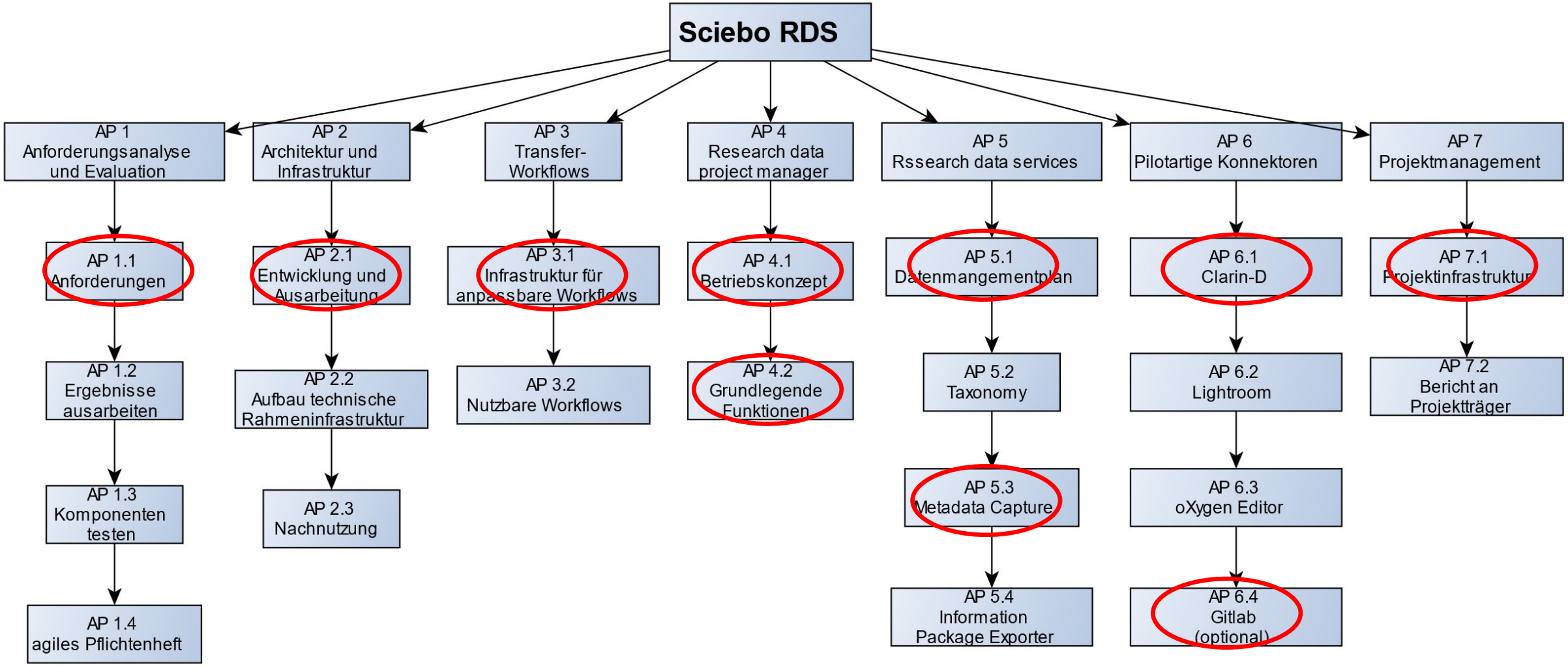
Wie soll die Software anderen verfügbar gemacht werden?

- Sämtliche Software wird Open Source gestellt
- DevOps und CI/CD durch Gitlab-Runner
- Serversoftware wird …
  - … mittels Containerisierung (Docker) im Gitlab-eigenen Container-Repositorium bereitgestellt
  - … durch Docker-Compose lauffähig sein
  - … durch Helm in einem Kubernetes-Cluster lauffähig sein
- Ausführliche Dokumentation durch Arc42, OpenAPI v3 und Readthedocs

Projektstrukturplan



Projektstrukturplan



Übersicht

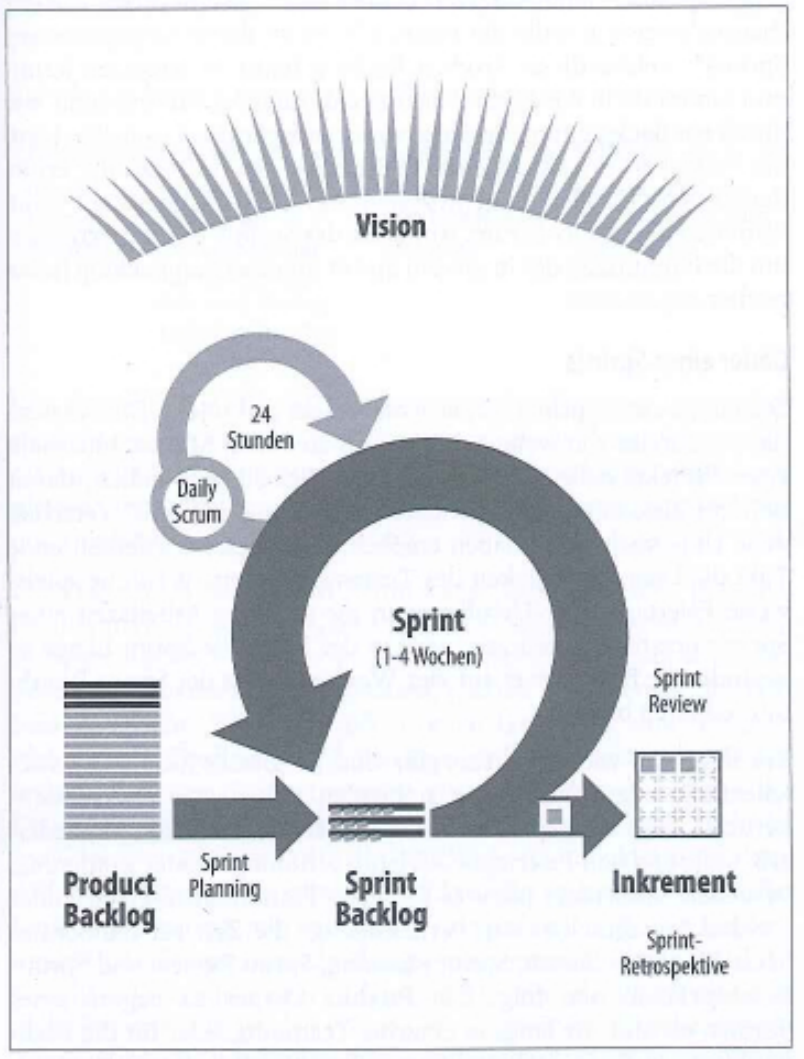
- Identifikation von (technischen) Randbedingungen und Lösungsstrategien
  - Recherche und Evaluation von Technologien zur Problemlösung
  - Einsatz von Microservices (Containerisierung (Docker)) und verteilte Systeme (Kubernetes)
- *Proof of Concept* eines ersten Architekturentwurfs erstellt
  - Erstellung einer DevOps-Pipeline für automatisches Erzeugen der Microservices
- Betrieb einer Testinstanz und Evaluation von RDMO für Datenmanagementpläne

Übersicht

- Einführung von agilem Projektmanagement
  - Umsetzung mittels Confluence und Jira
- Erzeugung von User Stories auf Basis …
  - des Projektantrags
  - von Interviews mit Forschenden (Bielefeld, Duisburg, Münster)

Agiler Entwicklungsansatz

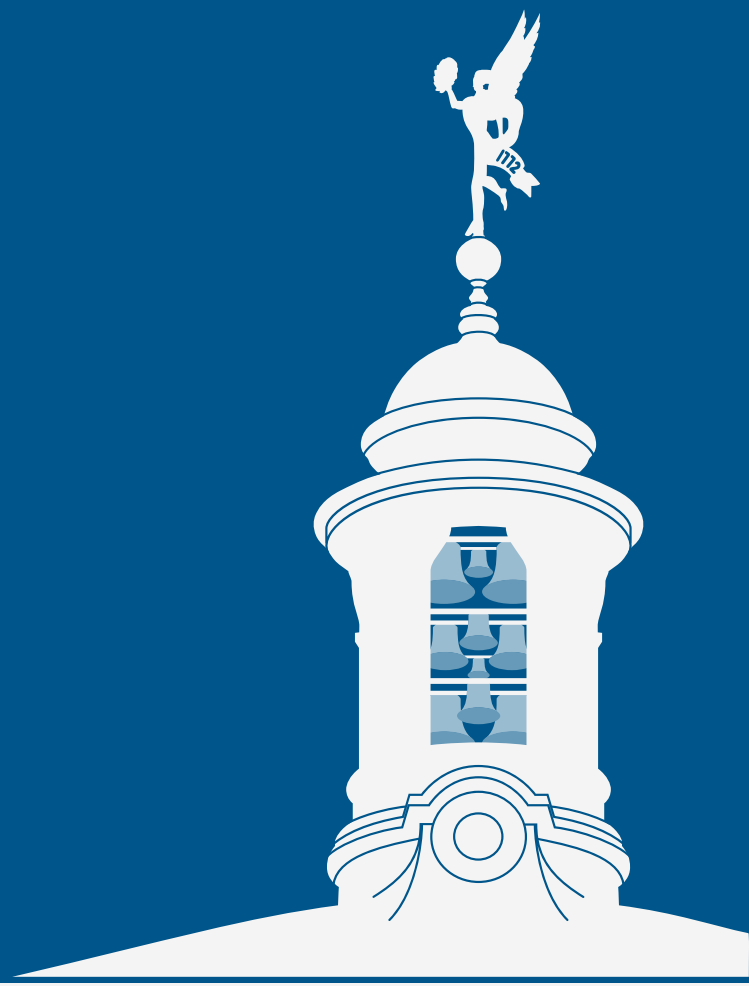
- Scrum-Framework, ergänzt um Elemente von XP und Kanban
- iterativ-inkrementeller Ansatz
- Anforderungen im Product Backlog dynamisch verwaltet
  - User Stories: "Als *Rolle* möchte ich *Funktion*, da *Nutzen*."
  - Anforderungsanalyse fortlaufend ≠ Wasserfallmodell
  - Ergänzungen und Änderungen jederzeit möglich
  - Feature Requests, Bug Reports usw. schnell berücksichtigen



Sind Fragen entstanden oder noch offen geblieben?

WWU
MÜNSTER

Vielen Dank für die Aufmerksamkeit



Jens Stegmann (jstegman@uni-muenster.de),
Peter Heiss (peter.heiss@uni-muenster.de)

wissen.leben

Web: www.research-data-services.de

Use Cases: Nutzergruppen und Szenarien

- WWU Münster: Center for Digital Humanities
  - Virtual Desktop Digital Humanities
- WWU Münster: Exzellenzcluster Religion und Politik
  - Online Editionen
- Uni Duisburg-Essen: DFG-GK „User-Centred Social Media“
  - Anforderungsanalyse und Evaluation
- Uni Bielefeld: Geistes- u. Sozialwissenschaften
  - Interdisziplinäre Anforderungen, Geistes- u. Sozialwissenschaften

AP 1: Anforderungsanalyse und Evaluation aus der Nutzerperspektive

Metaanalyse bestehender Studien und Literatur

Abgleich der Ergebnisse und Präzisierung spezifischer Bedarfe mit Vertretern verschiedener Disziplinen (Interviews, Befragung)

Ableitung von Implikationen zur Verbesserung der Komponenten

Wissenschaftliche Auswertung und Dokumentation der Projektergebnisse

Begleitung des Piloteinsatzes von FDM-Komponenten und Evaluation von Nutzerakzeptanz und Mehrwerten für FDM-Prozesse

Peter Heiss, Jens Stegmann, Holger Angenent

Forschungsfrage

How can existing research data services be improved to enhance the use of research data management steps?

Anforderungen verschiedener Disziplinen zur Verbesserung der Nutzung von Forschungsdatenmanagementservices

Mechanismen zum Befolgen von FDM Richtlinien